Thomas RIEBENBAUER

Statistical Trend Analysis for Development Projects in Semiconductor Industry

MASTER THESIS

written to obtain the academic degree of a Master of Science (MSc)

Master programme Technical Mathematics: Operations Research and Statistics

Graz University of Technology

Supervisor:
Univ.-Prof. Dipl.-Ing. Dr.techn. Ernst STADLOBER

Institute of Statistics

Graz, November 2013

This work was written in collaboration with ams AG [1].

[1] http://www.ams.com

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....	.....
Datum / date	(Unterschrift / signature)

Acknowledgment

First I want to thank my supervising professor Ernst Stadlober for his professional and personal support throughout the whole work on my thesis, which was very valuable to me.

Furthermore, I want to thank my collaboration partners at ams, namely Richard Forsyth (Engineering Director R&D Industrial and Medical Business Unit) and Christian Siller (Controlling), especially for their curious and critical view on my work. Many presentations and meetings – also with other employees of ams – helped me to sharpen the quality of my master thesis. At the same time a special thank-you goes also to Birgit Sponer for her time and dedication for discussions about her work, that deepened my understanding for the topic.

I also would like to thank my family, in particular my parents, for their love and support (not to mention the financial support) throughout the whole years of my study. Moreover, I owe many professional and personal coffee breaks and support to my study colleagues and friends during long days at the library.

Last but not least I want to thank my love Maria for her unconditional encouragement in every moment.

Graz, December 4, 2013 Thomas Riebenbauer

Abstract

Project cost and duration are two very important aspects in project planning. Based on data of past projects it is of interest how project cost and duration develop over the years. This thesis analyzes this and associated questions for projects of the semiconductor industry using mainly multivariate linear regression. The main issue is that one third of the underlying dataset are running projects, for which no final cost and duration is available. As estimations at project start turn out to be biased, the idea is to generate models based on closed projects to apply on running projects. By examining project structure in detail it is possible to use additional project information to improve the model predictions significantly. Using the resulting predictions, 240 different trends are analyzed: Over the years durations mainly decrease and duration estimations improve. Project costs are more or less constant, while estimation errors increase.
As the analysis is performed with the statistical software package **R**, another focus is on usage of the graphical package `ggplot2`. Most graphics are produced with this package and main functionalities are coded in presented functions.
This master thesis is a continuation of Sponers master thesis (2009) and is written in collaboration with ams AG, Unterpremstätten.

Kurzfassung

Projektkosten und -dauer sind sehr wichtige Faktoren in der Projektplanung. Basierend auf Daten von vergangenen Projekten, ist die Entwicklung dieser Faktoren im Jahresverlauf von Interesse. Diese Arbeit analysiert diese und ähnliche Fragestellungen für Projekte aus der Halbleiter Industrie mittels multipler linearer Regression. Laufende Projekte, die ein Drittel des Datensatzes ausmachen, spielen eine zentrale Rolle, da für diese keine tatsächlichen Projektkosten und -dauern vorhanden sind. Schätzungen zu Projektstart stellen sich als verzerrt heraus, weshalb Modelle basierend auf abgeschlossenen Projekten erstellt werden, um diese dann auf laufende Projekte anzuwenden. Zusätzlich können die Modellschätzungen durch tiefgreifende Analysen der Projektstrukturen signifikant verbessert werden. Darauf aufbauend werden 240 verschiedene Trendanalysen durchgeführt: Die Projektdauer sinkt im Zeitverlauf vorwiegend und die zugehörigen Schätzungen verbessern sich. Projektkosten bleiben durchschnittlich konstant, während die Schätzfehler steigen.
Die Analysen werden mit dem Statistik-Programm **R** durchgeführt, im Zuge dessen auch Wert auf die Verwendung des Grafikpaketes `ggplot2` gelegt wurde. Der Großteil der Grafiken wurde mit diesem Paket erzeugt und die wichtigsten Funktionalitäten in Funktionen kodiert.
Diese Masterarbeit ist eine Weiterführung der Masterarbeit von Sponer (2009) und wurde in Zusammenarbeit mit der Firma ams AG, Unterpremstätten verfasst.

Contents

List of Figures

List of Tables

Acronyms

s_{mad} median absolute standard deviation

`Inf` infinity

`NA` not available

`NaN` not a number

`lm` linear model

AIC Akaike Information Criterion

ANOVA Analysis of Variance

BIC Bayesian Information Criterion

CI confidence interval

CSV comma-separated values

df degree of freedom

EDA Exploratory Data Analysis

iid independent and identically distributed

ind independent

IQR interquartile range

LOWESS locally weighted scatterplot smoothing

LVCF Last Value Carried Forward

ME million Euro

ML Maximum Likelihood

PC Principal Component

PCA Principal Component Analyis

PI prediction interval

Q–Q plot quantile–quantile plot

SLR simple linear regression

SSE Error Sum of Squares

SSR Regression Sum of Squares

SST Total Sum of Squares

WS workstatement

Chapter 1.

Introduction

Project cost and duration are two very important aspects in project planning. Based on data of past projects it is of interest how project cost and duration develop over the years.

- Do projects tend to cost more or less?
- Do projects tend do last longer or shorter?
- What influences these factors and how?
- How can project cost and duration be estimated at project start?

Answering these questions aid on getting deeper insight in the structure of projects, which can give crucial information on project planning.

This thesis uses methods of statistical analysis, especially multiple regression analysis, to give insight into product development projects of the semiconductor industry. The analysis is performed with the statistical software package **R** (see appendix B, **R**-Packages and Functions). Below research objectives are given, that are analyzed in detail based on project data covering several years.

The collaboration partner for this thesis is ams AG, an Austrian company with headquarters in Unterpremstätten that develops and delivers analog semiconductors. The company ams has about 1300 employees in over 20 countries worldwide. The main market fields of ams are consumer, industrial, medical, mobile communications and automotive markets (AMS AG [1]).

Research Objectives

Beside many side questions of interest the main research objectives are as follows:

1. **Is the database sufficient?**
   The given dataset covers a wide range of product development projects. At the same answers are needed for the question, which part of data is sufficient to give significant results and which parts cover not enough data.

2. **How do project cost and duration develop over the years?**
   Is there a trend of project cost and duration or are these factors stable over time? If there is a trend: Where does it come from? Special interest is on the last few years of the data.
3. **Did estimations of project cost/duration get better over the years?**
   At project start duration and cost are estimated. The question is how good these estimations are and how their quality develops over the years.
4. **How to get faster and cheaper?**
   When analyzing cost and duration of a project it is of interest what these factors are influenced by. Also the question arises how these influences can be described.

In 2009 Sponer [21] gave answers on similar questions. This thesis can be seen as a continuation of her work. Due to comparability, here a similar thesis structure is chosen.

Thesis Structure

The central part of this thesis is to analyze the given dataset according to the research questions stated above. Chapter 2 describes the whole practical analysis process. After the database and its structure is examined, a closer look on the variables is given by using exploratory data analysis (EDA). Based on this an overview of the analysis approach is given. First models based on data at project start are generated. These models are improved by further analysis. As a main part of the analysis the trend analysis and its results are presented. Finalizing the practical part, models of the previous thesis by Sponer [21] are applied to the new data.

The practical analysis is supported by theory in chapter 3 Theoretical Fundamentals. The theory about linear regression models is extended by methods on diagnosing the models. A focus is also laid on methods in selecting "the best model" from a set of possible models.

The final chapter 4 Conclusions firstly provides a short summary of the main aspects of the analysis. After implications and interpretations are given, possible future prospects are discussed.

Appendix A Common Statistical Distributions states the most common statistical distributions within the context of this thesis. As the analysis is done with the statistical package **R**, appendix B **R**-Packages and Functions briefly describes **R** and states the most important packages used. Finally appendix C Self Written **R** Functions lists the source code of some by the author self written functions, that may be of interest to the reader.

Chapter 2.

Practical Analysis

2.1. Database

The data is based on development projects starting from the year 2000 up to the beginning of 2013. It can be seen as a snapshot of project status at the date of data extraction. For this analysis a raw dataset of development projects with all corresponding variables was generated.

Thus for a single project many different variables are given. For the purpose of this analysis these variables are sorted out by relevance, to have only the needed variables left. The same is done with the set of projects. These steps as well as other critical steps to form the final database are briefly described in section 2.1.1, Data Preparation.

For confidential reasons the raw data as well as the final data used to analyze are not provided within this thesis. Instead characteristic numbers of the data can be found in section 2.1.2, Variables, where the variables used for analysis are described.

To apply statistical analysis different models are used to describe the data. These models underly certain assumptions on the data. Some of them can be checked for example by hypothesis tests, others have to be assumed. Hypothesis tests are mentioned where needed and general assumptions are stated in the Assumptions section 2.1.4. For general assumptions about the models please see chapter 3, Theoretical Fundamentals.

2.1.1. Data Preparation

As the basis of the database consists of user input, the first step before starting to analyze the data is to prepare it.

According to Stadlober [23] the three steps of critical data inspection are to check

- integrity,
- consistency and plausibility,
- actuality and utility.

Applying these steps to the raw data results in the following tasks:

Data Format: The raw data is given in a comma-separated values (CSV) format. For reading a CSV file it is important to be careful about special characters. Especially the separation sign, the sign for the decimal point and, if used, the quotation sign need to be identified.

If quotation signs are used, each opened quotation also has to have a closing quotation sign within the respective cell, to keep different entries separated. If the quotation sign is used within a data entry, it has to be replaced by an alternative sign, so that multiple entries are not get treated as one.

After these modification steps the data can be read into R by the `read.csv2` function (**R**-package `utils`, which is part of **R**).

Missing Values: In the raw database missing values are marked by different identifiers. So missing values have to be identified and denoted by a unique identifier. Here the **R** built in identifier `NA` is used, which stands for "not available".

Where possible missing values are filled with correct values. For getting the missing information other databases and user information are used.
Where correct values were not available missing value imputation methods were used partly. Especially the principle of the Last Value Carried Forward (LVCF) technique (see Todorov and Templ [28]) was used, where logically applicable.

Data Validation: The nature of data based on user input is that errors may occur. For that reason the data has to be validated and corrected where possible.

The consistency and plausibility of the data is checked by logical relations. For example outliers are rechecked, variable sums are validated, where the sum is given or dates with a certain order are compared.

Extract Information of Interest: Some variables in the data are not of interest for this analysis and some variables contain redundant information. This unnecessary information is sorted out to have only *basic variables* left.

The raw database contains general development projects. Here only product development projects are of interest. Hence product development projects are filtered out by certain criteria to define the fundamental of the final dataset.

Data Rearrangement: To meet the needs of **R** and its functions for statistical analysis as well as the research questions, the data was rearranged. This includes

- extracting coded information,
- combining variables to new variables,
- creating new variables for categorizations,
- and harmonizing variable units.

The raw dataset consists of several CSV files. Thus within **R** they are combined to a single dataset containing all information of interest.

Data Anonymization: As the data includes confidential information, variables are partly anonymized for this publication.

The anonymization does not affect the statistical analysis itself.

2.1.2. Variables

For the analysis *basic variables* are used. These variables are chosen so that they contain no overlapping information as far as possible and on the same time reflect the core information of a product development project. The basic variables used for analysis are named and briefly described in table 2.1. Table 2.2 states ranges and units for each variable.

Table 2.1.: Variable names and descriptions

Name	Description
`dur_est` & `dur_act`	estimated & actual project duration
`finance`	financial project type
`bu`	business unit
`proj_start`	project start
`status`	project status
`cost_est` & `cost_act`	estimated and actual total engineering cost (project cost)
`eng_h`	engineering hours
`purch_c`	purchased cost
`mat_c`	material cost
`nre`	net revenue engineering
`chip_asp`	single chip average sales price
`chip_c`	single chip cost

Continued on next page

Table 2.1 – continued from previous page

Name	Description
`tech`	size of the technology
`die_size`	size of a single die (die size)
`sort_t` and `test_t`	sort and test time
`reuse`	effort for chip design and layout (reuse)
`yield`	yield
`pin_ct`	number of chip pins (pin count)

Remark 2.1

- The variable `reuse` $\in [0\%, 100\%]$ describes the effort of chip design and layout in the following way:
  - `reuse` $= 0\%$ means that everything has to be built newly.
  - `reuse` $= 100\%$ means that the chip design of an already developed chip can be completely reused.

  Practically `reuse` ranges within 10% to 90% most of the time.
- A project has a project start (the date is indicated by `proj_start`), at which variable values are estimated to get an impression of the project. The estimations of project cost and project duration are marked by the extension `_est`.
- The extension `_act` for project cost and duration indicates the final or actual value[2]. Actual values are collected on project closure.
- For describing project duration the time gap between project milestones named M2 and M8 is chosen. This information provides the most accurate and most comparable measurement of project duration within the given data. It shall be noted that this duration does not reflect the complete project duration, but the essential part of product development.
- The horizontal lines in table 2.1 as well as in table 2.2 mark, beside head and foot line, variable categories in the following order:
  - basic project information
  - project cost information
  - sale information and
  - technical chip information

[2]This meaning changes for projects that are still running, as no final values are known yet (see remark 2.2).

Table 2.2.: Variable ranges and units

Name	Range	Unit
dur_est & dur_act	[0.08, 8.89]	years
finance	{F1, F2}	*categorical*
bu	{BU1, BU2, BU3, BU4}	*categorical*
proj_start	[2000-03-24, 2013-01-07]	date
status	{running, closed}	*categorical*
cost_est & cost_act	[0.005, 4.616]	million Euro
eng_h	[0, 29155]	hours
purch_c	[0.00, 1.31]	million Euro
mat_c	[0.00, 1.15]	million Euro
nre	[0.00, 1.77]	million Euro
chip_asp	[0.00, 23.30]	Euro
chip_c	[0.00, 9.89]	Euro
tech	{0.13, 0.35, 0.6, 0.8}	μm
die_size	[0.26, 150.51]	mm^2
sort_t	[0.00, 75.00]	seconds
test_t	[0.00, 51.50]	seconds
reuse	[5, 100]	%
yield	[50.0, 99.7]	%
pin_ct	[3, 484]	pins

As remarked the variable name extensions `_est` and `_act` mark estimated and actual values of the two response variables duration and cost. To mark the same for most of the other variables the following extensions are used:

`_f`: the estimated value at project start. It is the variable value of the *first* workstatement[3].

`_l` the estimated value at project end/closure[4]. It is the variable value of the *last* workstatement.

Remark 2.2

- The two main variables describing cost and duration will also be called *response variables* and the other variables *predictor variables*. This is because duration and cost will be modeled by the other variables.

[3] defined in section 2.1.3

[4] This meaning changes for running projects, as they did not end yet (see remark 2.2).

In other words the *predictor variables* predict values for the response variables according to the model. The *response variables* respond on changes of the predictor variables according to the model. For details please see chapter 3, Theoretical Fundamentals.

- The variables `finance`, `bu`, `proj_start` and `status` do not have any extensions as they are fixed for a certain project. The `status` of a project is also fixed for a certain project and has no extension, as it reflects the project status at the time of database creation.

  Projects will also be called *running* or *closed* projects referring to the value of `status`.

- For running projects the meaning of the extensions `_act` and `_l` changes as they did not end yet. Thus these extensions indicate the last available information (e. g. the information of the last available workstatement).

2.1.3. Structure

Workstatement (WS) A project state is summarized in a so called *workstatement (WS)*. A WS consists of all project variables and its values at the state of WS creation. Every project has a *first* WS at project start. Every project with `status` closed has also a *last* WS at project end.

A project can also have more than two WS, but they are not created on a regular basis. Thus they can not be used to compare projects directly. Despite that the other WS will be used in chapter 2.4 to refine models.

The final dataset after data preparation (described in section 2.1.1) consists of 479 **product projects**. To recognize a certain project a unique id $\in \{1, \ldots, 479\}$ is used for identification. Within this 479 product projects also 89 **atypical projects** are included. Atypical projects are not of interest for this analysis for different reasons, e. g. canceled projects, split projects, merged projects. This sort out was done in accordance with ams AG to retrieve the projects of interest.

In the following the 390 left **typical projects** will be called "**all projects**", as they are all projects of interest. These 390 projects are also divided into subsets by WS availability and status:

There are 130 typical projects that have `status` running. All 130 **running projects** have all WS available (i. e. no missing WS). Beside the running projects there are 227 **closed projects with all WS available** (i. e. no WS is missing). The remaining 33 projects are **closed projects with missing first WS**. Missing first WS means that the WS at project start is not available. Hence these projects can not be used when analyzing variable values at project start, but when analyzing values at project end.

This described structure of the dataset is shown in fig. 2.1, which also points out the distinction of running and closed projects by the use of colors.

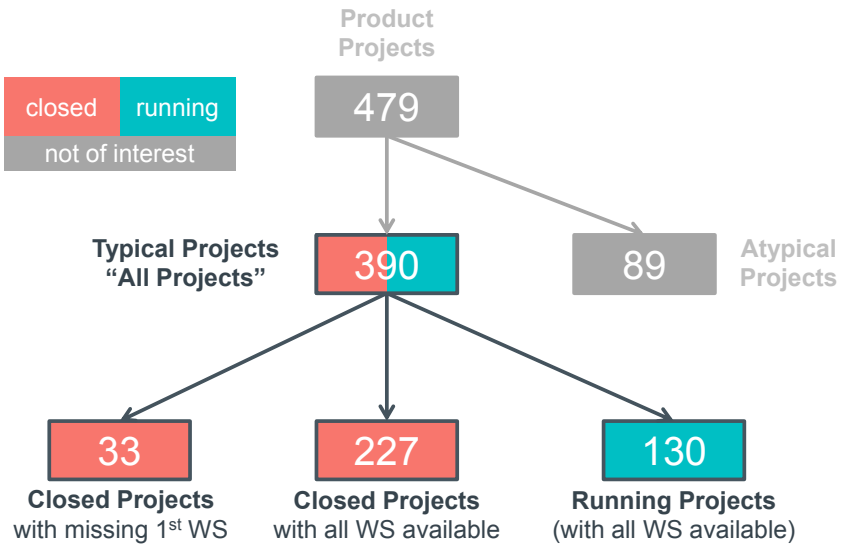


Figure 2.1.: Database structure

The specified structure of the projects is of importance as it yields different subsets of the data that are used context-based. The given names are used to refer to the specific datasets.

Furthermore also unions of the defined data subsets are used, as presented in fig. 2.2. When analyzing known actual values at project end, **all closed projects** are used (union of closed projects with missing first WS and closed projects with all WS available). The union of closed projects with all WS available and running projects gives **projects with all WS available**, that are of interest when analyzing the first WS.

Remark 2.3

- Obviously it is of importance to distinguish which database is used for analysis and on which database graphics are based on. Thus within this thesis the used database will be clearly stated in connection with the analysis and in figure captions.
  To refer to the databases also short names will be used (see table 2.3).

- There are 6 different subsets of interest of the original dataset (including itself). On different points of the statistical analysis different datasets are important. It also occurs that looking at different datasets for the same analysis is crucial. To not go beyond the scope of this written thesis only the most interesting views on the datasets are presented.

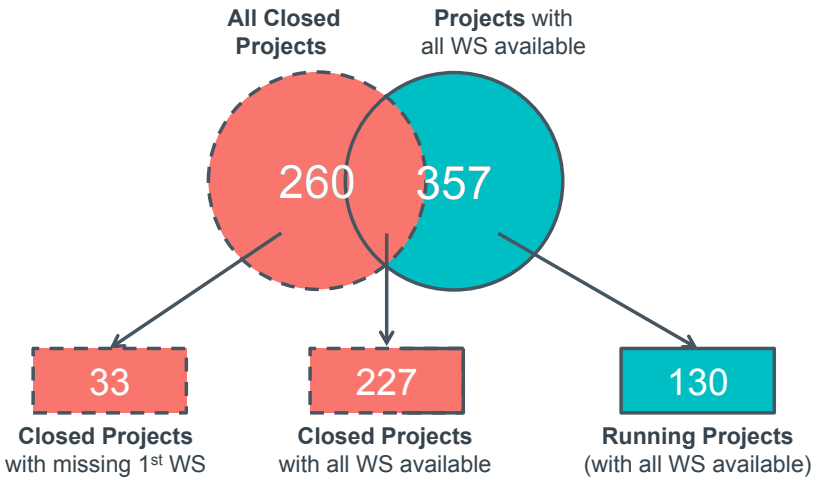


Figure 2.2.: Database structure details

Table 2.3.: Dataset names: short and long names

Short Name	Long Name
`all_pr`	all projects
`cl_pr_miss_f_WS`	closed projects with missing first WS
`cl_pr_all_WS`	closed projects with all WS available
`run_pr`	running projects (with all WS available)
`all_cl_pr`	all closed projects
`pr_all_WS`	projects with all WS available

Running and Closed Projects

As already indicated in fig. 2.1 and 2.2, the project status is of main interest.

Remark 2.4 The used colors to indicate closed (red) and running (blue) projects (see fig. 2.3) will be used throughout this thesis, unless specified different.

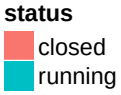


Figure 2.3.: Colors to indicate `status`

The importance of project status can be seen when looking at the distribution of the number of projects across project start year (see fig. 2.4). In 2012, the year of data generation, started 28 projects. 26 of them are still running and 2 projects are

already closed. When going back in time the portion of closed projects is increasing and the portion of running projects is decreasing. That means when looking at the last few years (e. g. 2010-2012), most projects are running projects (approximately 83%). This means that running projects cover an important part of all projects, especially because recent years are of interest regarding the research objectives.

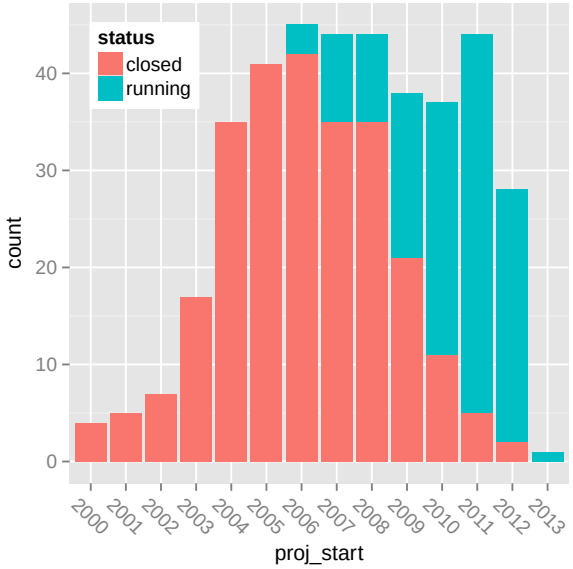


Figure 2.4.: Histogram of project start, stacked by project status. One bin covers one year. (data: all projects)

The main difference between closed and running projects is as follows:

closed: Estimations at project start are known (excluding the 33 projects with missing first WS) as well as actual variable values at project end.

running: Estimations at project start are known, but actual variable values are unknown.

Significant Dates

There are two significant dates for the database.

Today (= 2012-10-29) means the date of data generation. It is the date at which time a snapshot of all projects is taken. This means that running projects were running projects and closed projects were already closed on the date *today*. So the database reflects the state of all projects on this date.

Data Collection Start (= 2006-10-25) is the date, when the data was started to be collected. This data is an approximation based on coherences within the data.

On this data the database was introduced and all projects that started from that date on are collected in the database. Additionally all projects that were running on that date were also added to the data, which is important to consider when analyzing cost and duration trend.

The importance of these two dates is also pointed out in section 2.3, Modeling Approach.

2.1.4. Assumptions

Statistical models always underly certain assumptions that need to be checked. These assumptions can be found in chapter 3, Theoretical Fundamentals and have to be inspected for each model separately.

Additionally some basic assumptions about the dataset have to be assumed:

- **Trend information is hold by basic variables**
  To analyze project duration and cost trend over years, models based on closed projects are used to describe final duration and cost of running projects. These models are based on basic variables of the first WS (excluding project start). This means that the information of project start and project end is not used for these models.

  Hence the assumption is that the basic variables of closed projects (except project start) hold information about duration and cost trend. Thus the trend of duration and cost is an effect of changes in basic variable values.

  It was demonstrated that adding project start or project end to the predictor variables causes an unnatural bias (see section 2.5, Model Improvements).

- **Comparability of projects over time**
  As the models to predict final duration and cost of running projects are based on closed projects it is assumed that the sets of running projects and closed projects are comparable. That means that a running and a closed project having similar structure at project start have similar project duration and cost.

- **Variable meanings changing over time**
  Due to renewals over time some variable changed in their meaning. In accordance with ams AG certain steps were performed to harmonize these variables over time. The assumption is that after harmonization all variables have a meaning consistent over time.

  Examples for variable harmonizations are as follows:

  `eng_h` The engineering hours got split up to have a record of the different kinds of engineers working on a project. Here the sum of the single variables is used to have the full engineering hours for all projects.

bu The structuring of the business units changed over time. Some business units were split and some were merged. Here a categorization is used that covers the meaning of all business unit categorizations.

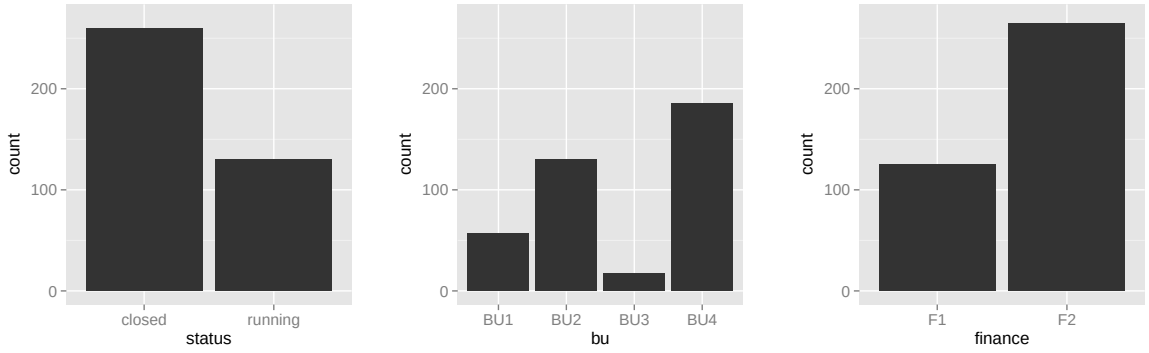
2.2. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was introduced by the statistician John Tukey in the 1970s. It is a method of graphical and numerical procedures to get a deeper insight into the data and its structures (see STADLOBER [23]).

In practical statistical analysis EDA often plays an important role in nearly every part of analysis steps. Here for example the concept of EDA is used to observe the structure of the data as given in section 2.1.3.

2.2.1. Univariate Variables

Categorical Variables




```
closed running
   260     130
```

```
BU1 BU2 BU3 BU4
 57 130  17 186
```

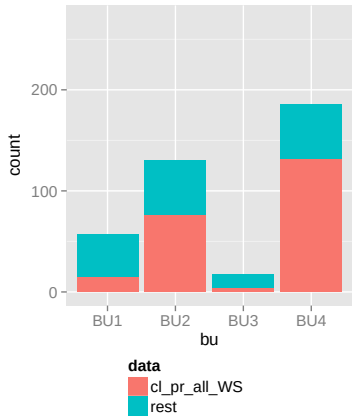
```
 F1  F2
125 265
```

Figure 2.5.: Bar chart of `status`, `bu` and `finance`. Underneath each graphic the absolute number of projects is stated accordingly. (data: all projects)

`status, bu, finance`: In section 2.1.3, Structure the distribution of project status over the project start was already pointed out. Here (fig. 2.5) it can be seen that there are twice as much closed projects than running projects. It is likely the same when looking at `finance`: more than twice as much projects have financial project type F2 compared to F1.

Looking at `bu` it turns especially out that there are few projects (about 4% of all projects) with `BU3`, which might result in difficulties analyzing this business unit

separately. The models for cost and duration of running projects will be based on the set of closed projects with all WS available. Looking at this dataset (see fig. 2.6) there are just 4 projects of `BU3` (about 2% of `cl_pr_all_WS`) and 15 projects of `BU1` (about 7% of `cl_pr_all_WS`). To be able to use the business unit category without loosing information within the levels of `BU1` and `BU3` in section 2.2.2, Reducing Category Levels these two levels will be merged with other levels.



	BU1	BU2	BU3	BU4	Sum
cl_pr_all_WS	15	76	4	132	227
rest	42	54	13	54	163
Sum	57	130	17	186	390

Figure 2.6.: Bar chart of `bu` stacked by data. (data: all projects)

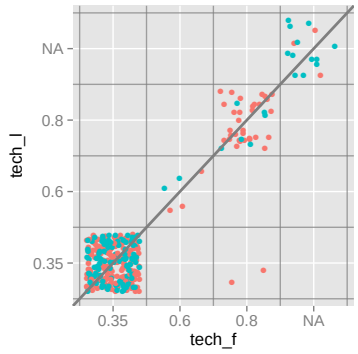
`tech`: Although the variable `tech` is not a categorical variable, it is presented here, as it has only 4 distinct values. Looking at the projects with all WS available in fig. 2.7 it can be seen that there is a very small group of projects with a technology size of 0.6. Thus also projects with `tech` 0.6 are candidates for merging it with projects of another value (see 2.2.2, Reducing Category Levels).

Remark 2.5 Looking at the 33 closed projects with missing first WS (they are not shown in fig. 2.7) is not of big interest, as `tech_l` is only used in section 2.7, Model Application of Previous Thesis. Nevertheless it shall be noticed that the left `tech`-value of 0.13 is used by one project of this set of projects only. The project with `tech`-value 0.13 gets merged with the projects with `tech`-values of 0.35 based on the same procedure which will be shown in 2.2.2, Reducing Category Levels.

Quantitative Variables

To get an overview of the quantitative variables, table 2.4 gives the values of basic characteristic numbers. Table 2.5 shows the Pearson correlation[5] r of estimated and actual resp. last available estimated variable values.

[5]Measures the linear relationship between two variables (see Stadlober [22])



	Sum	296	5	41	15	357
	NA	-	-	-	15	15
tech_l	0.8	-	-	39	-	39
	0.6	-	5	-	-	5
	0.35	296	-	2	-	298
		0.35	0.6	0.8	NA	Sum
			tech_f			

Figure 2.7.: Categorized scatterplot of `tech_f` against `tech_l` with added jitter (data: projects with all WS available)

Table 2.4.: Basic characteristic numbers of quantitative variables (data: projects with all WS available)

	Min	$q_{0.25}$	Median	Mean	$q_{0.75}$	Max	#NA
dur_est	0.08	0.58	0.78	0.89	1.06	3.42	19
dur_act	0.12	0.83	1.22	1.54	1.90	6.43	9
cost_est	0.00	0.15	0.26	0.35	0.48	2.71	0
cost_act	0.00	0.15	0.30	0.45	0.61	3.02	0
eng_h_f	0.00	880.00	1762.00	2462.02	3400.00	14300.00	0
eng_h_l	0.00	1148.00	2162.00	3437.08	4347.00	23701.00	0
purch_c_f	0.00	0.00	0.00	0.03	0.00	1.14	0
purch_c_l	0.00	0.00	0.00	0.05	0.01	1.16	0
mat_c_f	0.00	0.04	0.06	0.08	0.09	1.03	0
mat_c_l	0.00	0.04	0.06	0.08	0.09	1.15	0
nre_f	0.00	0.00	0.00	0.09	0.04	1.69	0
nre_l	0.00	0.00	0.00	0.10	0.05	1.77	0
chip_asp_f	0.00	0.25	0.53	1.36	1.28	23.30	0
chip_asp_l	0.00	0.23	0.50	1.31	1.13	20.97	0
chip_c_f	0.00	0.09	0.19	0.43	0.45	9.02	0
chip_c_l	0.00	0.09	0.19	0.42	0.42	9.89	0
die_size_f	0.32	1.70	4.76	9.62	10.23	131.50	0
die_size_l	0.26	1.71	4.88	10.10	10.56	150.51	0
sort_t_f	0.00	1.00	1.60	2.89	3.50	75.00	29
sort_t_l	0.00	0.75	2.00	2.76	3.92	20.00	23
test_t_f	0.00	1.00	1.80	2.75	3.64	30.00	45
test_t_l	0.00	1.12	2.20	3.31	4.40	22.00	40
reuse_f	10.00	50.00	70.00	65.78	85.00	100.00	31

Continued on next page

Table 2.4 – continued from previous page

	min	$q_{0.25}$	median	mean	$q_{0.75}$	max	#NA
reuse_l	10.00	50.00	70.00	65.38	85.00	100.00	31
yield_f	55.00	93.00	95.00	94.03	97.03	99.70	6
yield_l	58.91	90.00	95.00	91.91	96.03	99.70	0
pin_ct_f	3.00	10.00	19.00	31.58	33.50	484.00	78
pin_ct_l	3.00	9.00	19.00	29.24	32.00	314.00	66

Table 2.5.: Pearson correlation r of estimated and actual/last available estimated variable values (data: projects with all WS available)

		#NA	r
dur_est	dur_act	19	0.60
eng_h_f	eng_h_l	0	0.81
purch_c_f	purch_c_l	0	0.85
mat_c_f	mat_c_l	0	0.84
cost_est	cost_act	0	0.84
nre_f	nre_l	0	0.88
chip_asp_f	chip_asp_l	0	0.90
chip_c_f	chip_c_l	0	0.77
yield_f	yield_l	6	0.44
die_size_f	die_size_l	0	0.99
sort_t_f	sort_t_l	29	0.46
test_t_f	test_t_l	45	0.71
pin_ct_f	pin_ct_l	78	0.77
reuse_f	reuse_l	31	0.98

Remark 2.6

- Tables 2.4 and 2.5 show correlations within the dataset of projects with all WS available. This set contains running as well as closed projects. This yields into an issue to keep in mind, as actual values are compared to estimated values. This mixture is chosen as the intention of the tables is to give an overview and because running projects as 1/3 of all projects are important to look at.

- Differences in minimum and maximum of variable values comparing table 2.4 of this section and table 2.2 of section 2.1.2 result as in section 2.1.2 where all projects are used as database.

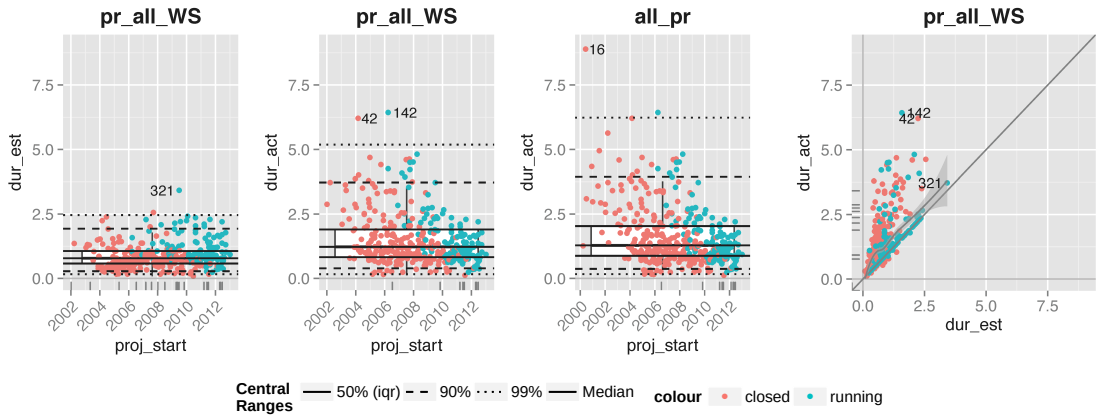


Figure 2.8.: Scatterplots of project start against duration and scatterplot of `dur_est` against `dur_act`. The first 3 plots contain boxplots for duration additional to the central ranges. On the last plot a 45° line is plotted for orientation. Also a LOESS-smooth[6] with 95% confidence interval (CI) is shown. Datasets are indicated by plot titles.

duration: Looking at project duration (see fig. 2.8), it turns out that the estimated duration is more evenly distributed over the start year than the actual duration. The actual duration has a distribution with higher standard deviation and higher median. Especially the last graphic in fig. 2.8 shows that the estimated duration tends to underestimate the actual duration. The Pearson-correlation on the third plot is 0.6 (see table 2.5).

The influence of the 33 closed projects with missing first WS (compare second to third plot in fig. 2.8) results mainly in outliers, as the upper bound of the 99% range increases. There are some outliers on each plot. Especially remarkable is project 16 on the third plot, wich is an already closed project started in 2000 that lasted nearly for 9 years.

cost: The variables representing project cost (see fig. 2.9) show a similar structure to duration variables presented above. The LOESS-smooth on the last plots indicates that there is also some underestimation of estimated cost to actual cost. Compared to duration, here we have a better approximation, which can also be seen by the Pearson correlation (cost: 0.84, duration: 0.6, see table 2.5).
There are some outliers regarding project cost. The most conspicuous are marked on each plot. Remarkable is project 272, as it has the highest estimated cost and

[6]LOESS smooth is strongly related to the LOWESS smooth (see section 2.2.3, Correlations. The **R** function is `loess` of the basic package `stats`.

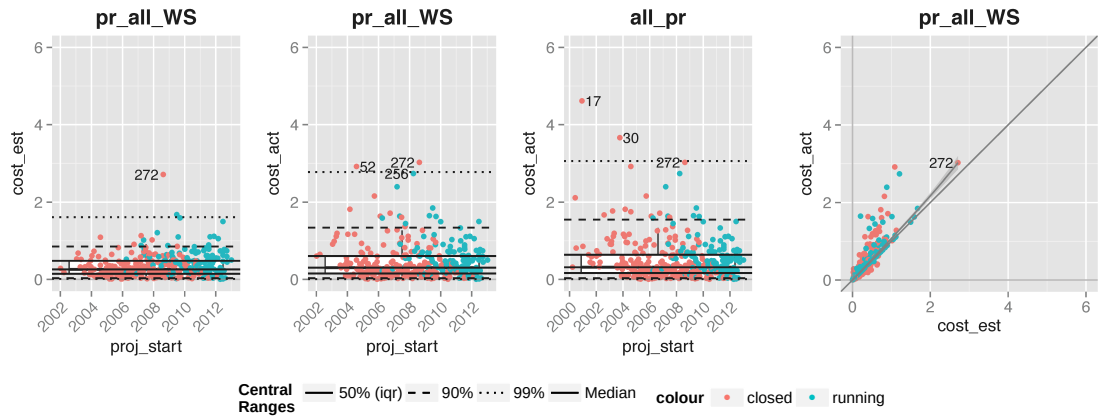


Figure 2.9.: Scatterplots of project start against cost and scatterplot of `cost_est` against `cost_act`. Datasets are indicated by plot titles.

the highest actual cost of all projects with all WS available on one hand. On the other hand project 272 acts as an outlier, as the estimated value is a rather good approximation of actual cost.

Remark 2.7

- Figures 2.8 and 2.9 show closed and running projects on all plots included. Especially regarding actual values it is important to keep in mind that "actual values" for running projects (i. e. variable has extension `_act`) reflect the last available estimations.

  Nevertheless these two figures give an overview of cost and duration distribution and the connection of estimations and actual values.

- On all plots of fig. 2.8 some short lines can be seen on the plot bottom or left plotting region. These lines represent missing values.

  For example look at the first plot of fig. 2.8: The short line at `proj_start` 2002 means that there is a project for which the estimated duration `dur_est` is not known, i. e. is a missing value.

  These *rug lines*[7] will have the same meaning on all remaining plots.

2.2.2. Reducing Category Levels

For modeling cost and duration 19 variables come into account as predictor variables. When using a categorical variable like `bu`, some problems may arise:

[7] produced by the `ggplot2` function `geom_rug`

- Some **R** functions used for regression (e.g. `regsubsets`, **R**-package `leaps`, see Lumley [15]) do not accept categorical predictor variables.
- Interaction of n predictor variables with a single categorical variable of m different levels produces $n \cdot m$ new predictor variables.

A possible solution to prevent these problems and the problem that some categories contain too less projects is to reduce the category levels. Here it was possible to reduce each categorical variable to two levels. Thus numerical values (e. g. 0 and 1) can be used instead of the remaining two levels. A variable of this type is called *dummy variable*.

Here Principal Component Analyis (PCA) is used to reduce category levels. This method takes a dataset of several variables and produces by orthogonal transformation a set of linear uncorrelated variables – the Principal Component (PC) variables. This set is generated in a sorting such that each PC has maximum variance (e.g. holds maximum information about the dataset) given the independence to previously generated PC. For more detailed information see section 3.5, Principal Component Analysis (PCA).
Based on the first few PC variables it will be decided which categories are *similar*, so that they can be merged.

PCA is performed on two categorical variables:

- **`tech`:** Here `tech` is treated as a categorical variable, as it has only four distinct values. Value 0.6 and also 0.13 are analyzed on merging with values 0.35 and 0.8.
- **`bu`:** Each of `BU1` and `BU3` shall be merged with one of `BU2` and `BU4`.

Remark 2.8

- The merging showed in this section is used in section 2.4, Modeling Cost and Duration only. The variable `bu` will be used in the original and the merged variant. To distinguish between these variants the variable name `bu_m` is used to refer to the variable with merged levels and `bu` shall refer to the original coding.
- PCA is performed on the set of projects with all WS available (`pr_all_WS`), as then all variables of the first and the last WS can be included.
- The used PCA function of **R** (see later) takes only numerical variables to perform the PCA. Thus all non-numerical variables are disregarded.
- When using PCA to analyze the category levels of `tech` also `tech_f` is removed from the dataset.
  The variable used to analyze size of technology is `tech_l`. The analysis was also performed with `tech_f`, but results were the same. This is not surprising as values of `tech_f` and `tech_l` are the same, except for 2 projects.

- The following details about the dataset refer to the analysis of `tech`, but are basically the same for the analysis of `bu`.

The used **R** function is `princomp` from the basic package `stats`, which is part of **R**. The function requires a dataset that does not include missing values, but only numerical variables. Omitting all missing values of the dataset `pr_all_WS` with 357 projects results in a dataset with 209 projects left (about 59%). To be able to use all projects, it would be necessary to leave out all variables with missing values (11 out of 29). To have a balance between the number of projects and the number of variables used, PCA will be performed on three different subsets of the projects with all WS available:

- All variables (except categorical and date variables) and missing values omitted:
  $\Rightarrow$ 209 projects (= 59% of 357)
  $\Rightarrow$ 209 projects $\cdot$ 29 variables = 6061 data points
  reference name: data1

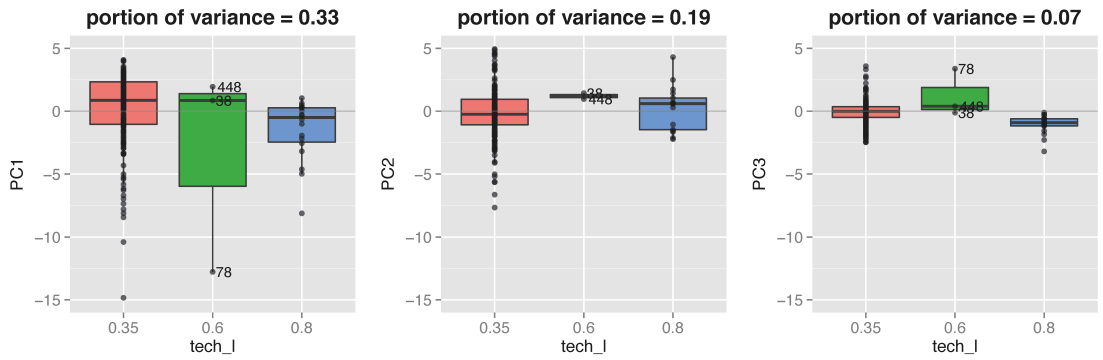
- Excluding variables `pin_ct`, `test_t` and `reuse` (each with `_f` and `_l`), that hold the most missing values (see section 2.2.4, Missing Values) and omitting missing values of the rest:
  $\Rightarrow$ 307 projects (= 86% of 357)
  $\Rightarrow$ 307 projects $\cdot$ 23 variables = 7061 data points
  reference name: data2

- Excluding all variables with missing values (see section 2.2.4, Missing Values) and omitting missing values of the rest:
  $\Rightarrow$ 357 projects (= 100% of 357)
  $\Rightarrow$ 357 projects $\cdot$ 18 variables = 6426 data points
  reference name: data3

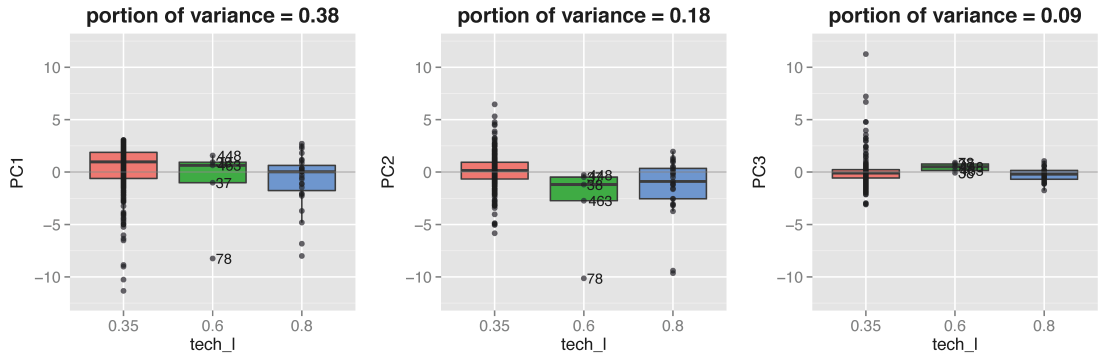
`tech`

Most important to look at is the first PC for each dataset (see fig. 2.10). Especially for data1 the data points with `tech`-value of 0.6 correspond better to 0.35 than to 0.8. Projects 38 and 448 are within the interquartile range (IQR) of 0.35 `tech` projects. Project 78 is an outlier that better fits to `tech` of 0.35. On the other plots there is no such clear tendency to `tech` of either 0.35 or 0.8.

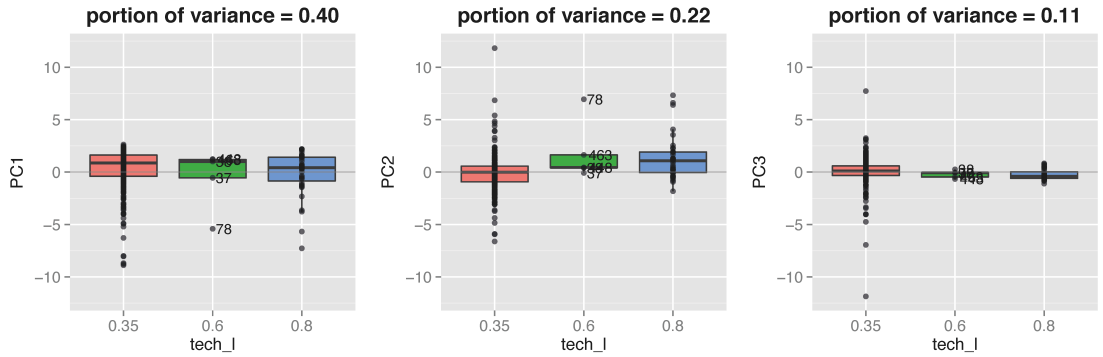
Result: Projects with `tech`-value of 0.6 are added to the set of 0.35 `tech` projects.



(a) data1, cumulative portion = 0.59

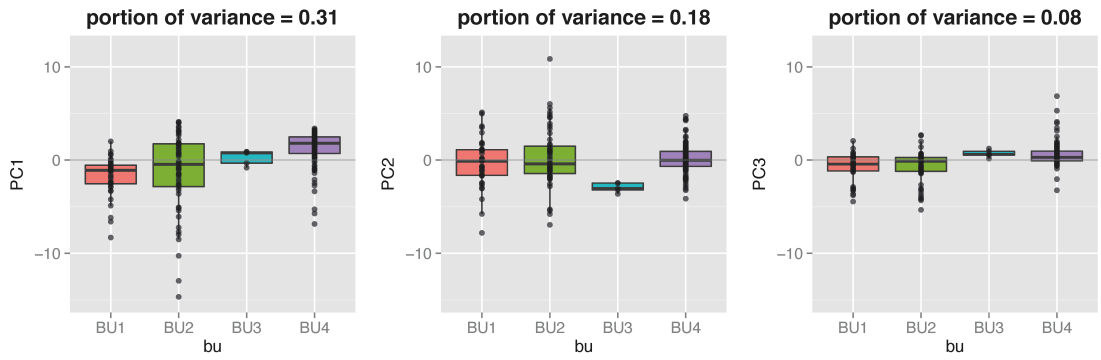


(b) data2, cumulative portion = 0.65

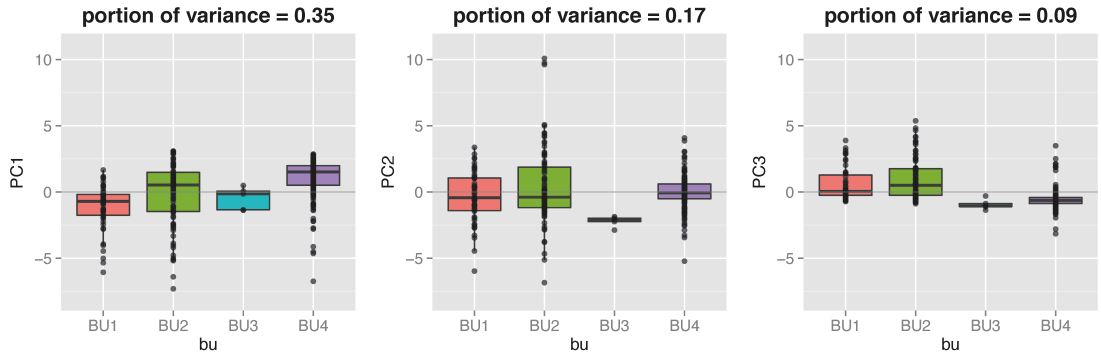


(c) data3, cumulative portion = 0.73

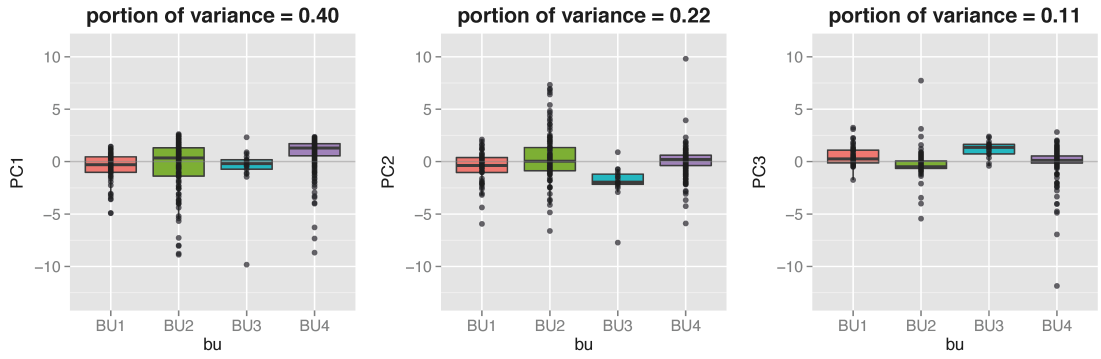
Figure 2.10.: Boxplot series of first three PC for each `tech_l`-value (data: projects with all WS available)



(a) data1, cumulative portion = 0.57



(b) data2, cumulative portion = 0.61



(c) data3, cumulative portion = 0.73

Figure 2.11.: Boxplot series of first three PC for each `bu`-value (data: projects with all WS available)

bu

BU1 projects fit nearly on all boxplot series (see fig. 2.11) best with BU2 projects. Thus these projects can be treated as similar and are put together labeled as BU2.

BU3: Looking at the first PC on all analyzed datasets BU3 projects fit the distribution of BU2 projects, but there is also no contradiction to fit with BU4 projects. On the plots of the second PC no tendency can be seen, as BU3 project values do not intersect with the IQR of neither BU2 nor BU4 projects. Regarding the third PC the boxplot series of data1 and data2 show a very good congruence of BU3 and BU4.

Result: BU3 projects are merged with BU4 projects as new bu category level BU4. BU1 projects and BU2 projects are merged to new bu category level BU2.

Remark 2.9

- The **R** function princomp gives for the described datasets always more than the used three PC (generally at most the number of variables of the dataset). In common it is not unique how many PC to choose to best represent the data and there exist different criteria to aid on deciding (see section 3.5, Principal Component Analysis (PCA)).
  Here the scree plots indicated 2 to 3 variables to choose. To be consistent it was decided to use 3 PC in all cases. On the same time already those three PC gave a satisfactory answer on the question of which category levels to merge.
- PCA for tech value 0.13 is not presented here, as it is not relevant for the first WS and works the same as shown above. The result was to merge 0.13 with 0.35 tech projects.

2.2.3. Correlations

The pairwise correlation matrix ellipses (see fig. 2.12) using Pearson correlation give a good oversight of how variables are correlated. The function used to generate the plot is plotcorr from the ellipse package (see Murdoch and Chow [17]). The ellipses represent the correlation of two variables by picturing the shape of a bivariate normal distribution of the respective correlation (see appendix A, Common Statistical Distributions about multivariate normal distribution).

Remark 2.10

- Here closed projects with all WS available are used as dataset as this is the data used to build the models of actual duration and cost. Therefore all relevant variables (except categorical variables) of interest for these models

are included. This covers variables of the first WS as well as actual duration and cost.

- To analyze correlations in this section project 272 is excluded from the data, as it is an extreme outlier that will also be excluded in further analysis.

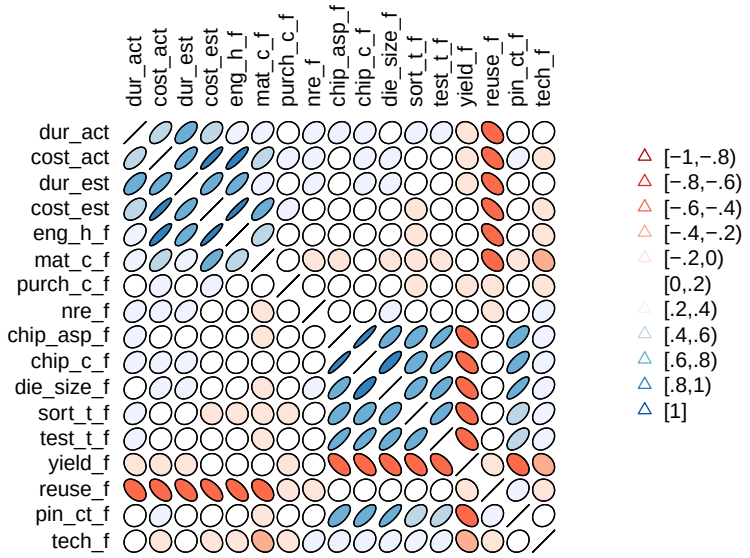


Figure 2.12.: Pairwise correlation matrix ellipses using Pearson correlation (data: closed projects with all WS available, excluding project 272)

Two blocks of high correlations (most correlations greater 0.6) can be located:

1. `dur_act`, `cost_act`, `dur_est`, `cost_est`, `eng_h_f`, `mat_c_f` and `reuse_f`
2. `chip_asp_f`, `chip_c_f`, `die_size_f`, `sort_t_f`, `test_t_f`, `yield_f` and `pin_ct_f`

The first block holds the response variables `dur_act` and `cost_act` as well as the respective variables of the first WS. The other variables are of interest, as they are candidates on predicting project duration and cost.

The second block corresponds to a set of technical chip variables and two sales variables. The only left out technical variables are `reuse` and `tech`. Here the size of technology can have only two different values, as the others are already merged with them. Reuse has a special role, as it is member of the first block that holds cost and duration information. This is logical as reuse is not a pure technical variable, but gives information on how much work of past projects can be reused. Thus the conclusion is that technical chip information correlates. At the same time sale information has a connection to technical information.

As the first block is of main interest in fig. 2.13 a scatterplot matrix of the corresponding variables is provided (using **R** function `pairs` of the basic package

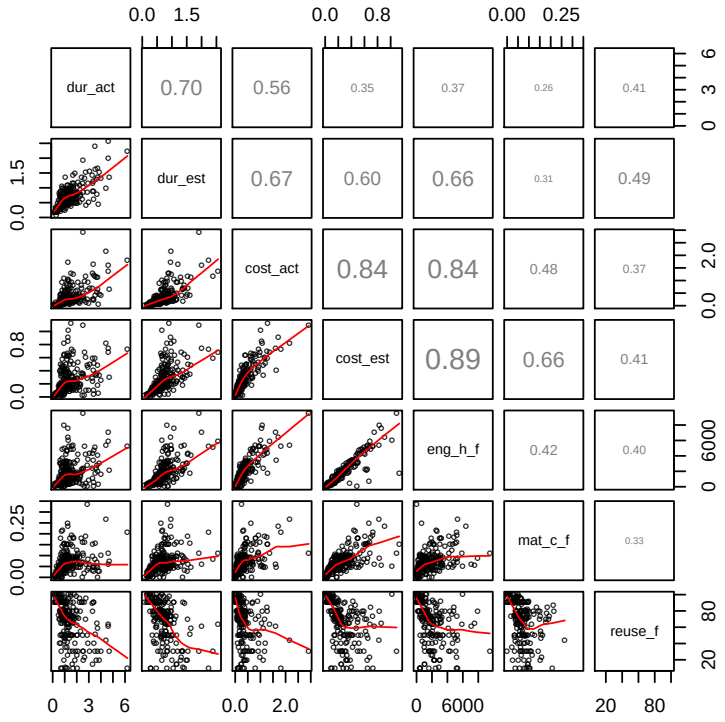
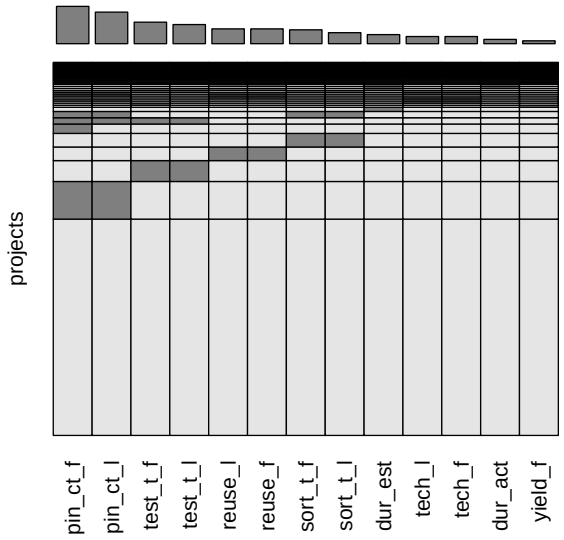


Figure 2.13.: Scatterplot matrix with LOWESS smooth on the lower panel and absolute Pearson correlation on the upper panel (data: closed projects with all WS available, excluding project 272)

`graphics`). The upper triangle shows the correlations of the corresponding variables with size relative to the correlation. In the lower triangle the scatterplots are shown with scatterplot smoothing LOWESS (locally weighted scatterplot smoothing) (see Cleveland [3]). The scatterplot matrix shows that most correlations seem to be based on a linear relationship with increasing variance. This indicates a transformation of the response variable (see Friedl [10], chapter 3). Remarkable is the correlation of 0.89 between `cost_est` and `eng_h_f`. This is interpretable such that engineering hours are the main part of project cost. Also of interest is the last row showing scatterplots using reuse. Here the relationship does not seem to be linear and the correlations are not that high (≤ 0.5) compared to the others. Especially high values of reuse tend to lead to lower cost and lower duration.

Remark 2.11 The variable correlations and characteristics within category levels are not shown here as this would go beyond the scope of this work. Nevertheless in the remaining various dependencies will be considered. The `regsubsets`-function will be used, which is a function for model selection.

2.2.4. Missing Values




```
combinations
  count percent cumsum_pct
1   207   57.98      57.98
2    36   10.08      68.06
3    20    5.60      73.66
4    13    3.64      77.30
5    13    3.64      80.94
...
```

Figure 2.14.: Aggregation plot of variables that contain any missing value. Columns indicate values of the denoted variable. Light gray areas represent available values and dark grey areas indicate missing values. The size of the areas are relative to the percentage of all variable values. Rows show connections of missing and/or non-missing value blocks. The **R**-output table on the right indicates the size of row blocks in reverse order. (data: projects with all WS available)

The data is based on user generated data, so missing values are normal. As mentioned in section 2.1.1, Data Preparation missing values were already important on preparing the data for analysis. The multivariate structure of missing values plays a main role, as it can be derived on which parts of the data to put effort on reconstruction. Also for further analysis the multivariate structure of missing values should be considered.

Multivariate structure of missing values here means how missing values are distributed over a dataset. That is how many missing values appear in a single variable and how missing values are related for different variables. In this thesis the **R**-package `VIM` (see Templ et al. [26]) is used, which provides a variety of graphical methods to visualize missing values structure.

When analyzing the data different visualization methods for missing values were used. The method presented within this thesis is the *aggregation plot*. For further methods and information see Templ et al. [25] and Templ and Filzmoser [27]. The missing value structure of projects with all WS available (see fig. 2.14) shows that only 207 projects (e. g. 58%) of the projects have no missing values.

Not using `pin_ct` and `test_t` results in $207 + 36 + 20 = 263$ (e. g. 74%) projects without missing values.

2.3. Modeling Approach

Using the results of EDA it can be decided on which approach to choose for modeling cost and duration trend. The upcoming section 2.3.1, Motivation states briefly why the performed approach was chosen. On the same time this leads to the importance of the dates *today* and *data collection start*. An overview of the analysis methods used is given in section 2.3.2, Methods Overview.

2.3.1. Motivation

When analyzing project trend it is of interest to look at the trend of raw data, as shown in fig. 2.15.

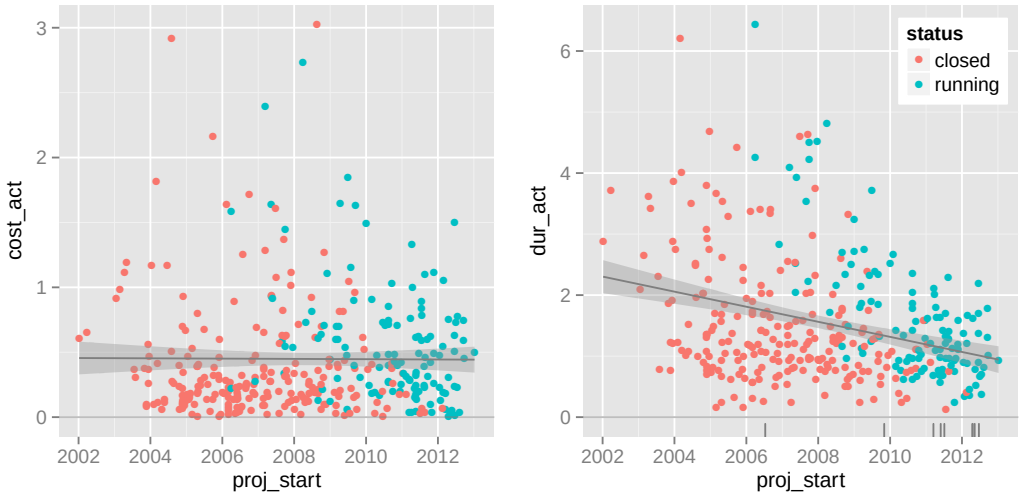


Figure 2.15.: Raw trend of actual cost and duration by project start year. The trend is represented as simple linear regression fit (data: projects with all WS available)

The simple linear regression fit with 95% confidence interval (CI) of actual cost and duration trend[8] indicates that there is no trend of project cost (the horizontal line is within the 95% CI) and a decrasing trend of project duration by project start. A problem about this point of view is that there are also running projects

[8]This and all other linear regression fits in graphics are generated automatically (unless otherwise noted). On this automatic generation there is no verification of model assumptions. Thus the fits cannot be treated as reliable models, but as an indicator to get an idea of a possible linear relationship.

included. As already specififed, actual values of running projects indicate the last available estimation. This may cause bias in the trend, as two not comparable meanings of actual values are mixed. Thus the question is where actual values for running projects come from.

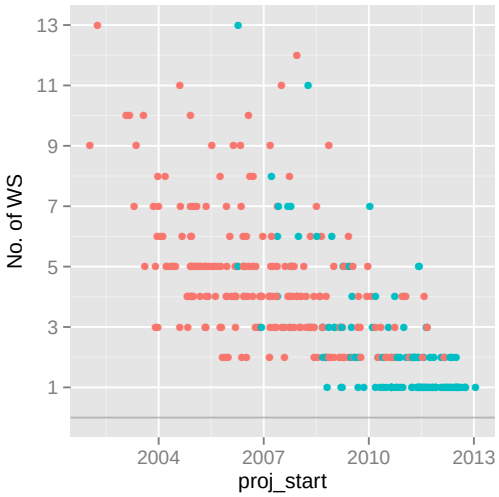


Figure 2.16.: Number of WS for each project by project start (data: projects with all WS available)

An answer to this question can be given by looking at the number of workstatements for each project, which is shown in fig. 2.16. It can be seen that most running projects (64 of 130 = 49%) have only one WS. In other words, for about half of the running projects the last available information is from the first WS.
It is also notable that the number of WS per project goes up to 13. This means that there may be some information lost, when just looking at the first and last WS. Part of this additional information will be used later to improve the models found (see 2.4, Modeling Cost and Duration).

For about half of running projects the last available information is from the first WS. So the question is how reliable this cost/duration information is. To answer this question, it can be looked at the comparison of estimated and actual values for closed projects (see fig. 2.17). Obviously, estimations for cost and duration tend to underestimate actual values. Cost estimations are relatively good for smaller projects (lower cost), but for higher cost (except for project 272) the trend is to underestimate more and more. In other words the estimated values have a systematical error in estimating actual values. Thus the approach is to generate unbiased models for fitting the actual values of running projects. To obtain these models the closed projects are used as database. After that the models can be applied on the running projects to get an unbiased fit with random error for actual cost and duration values.

As running projects are used to analyze trend, another question is why running

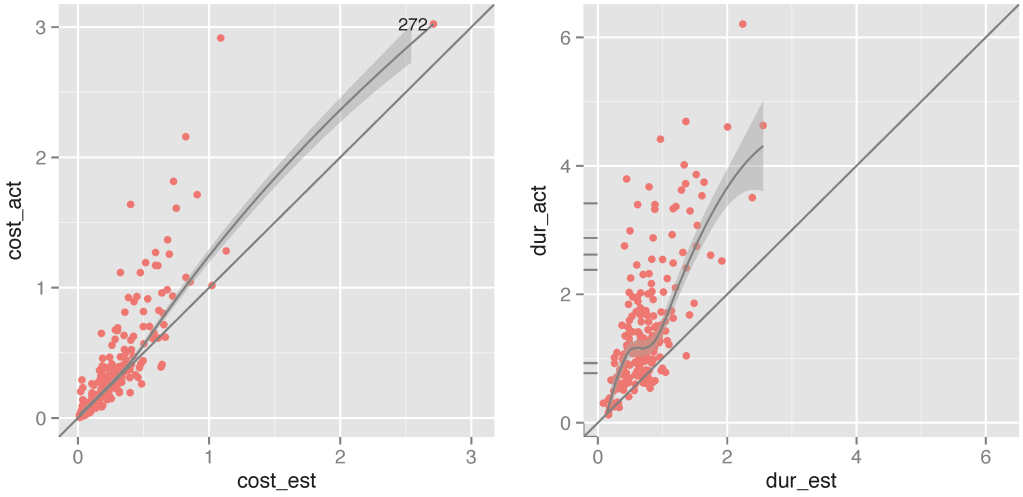


Figure 2.17.: Scatterplot of actual cost/duration against estimated cost /duration. LOESS smooth with 95% CI is added (data: closed projects with all WS available)

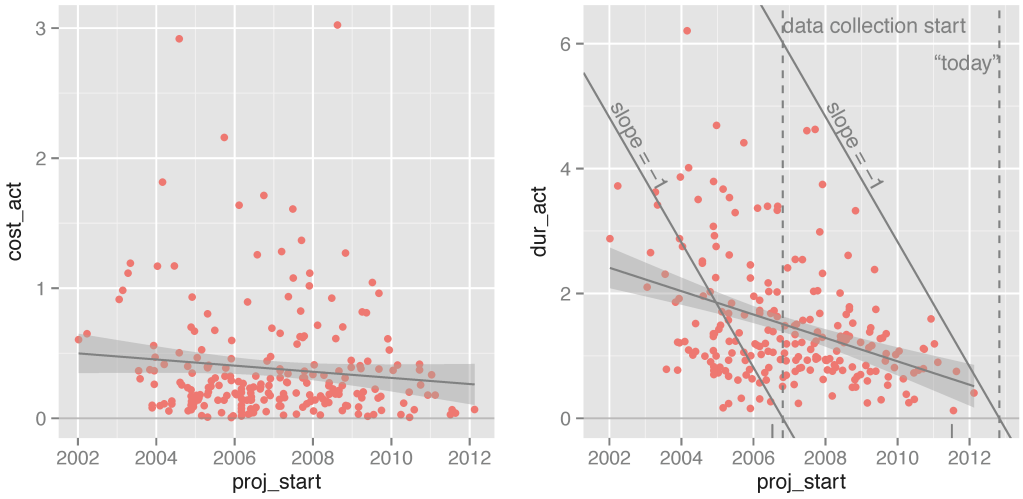


Figure 2.18.: Raw trend of actual cost/duration by project start year. On the right plot two significant dates are added. The trends are represented by simple linear regression fits (data: closed projects with all WS available)

projects are needed at all. Hence only closed projects may be used to analyze cost/duration trend (see fig. 2.18). For both, duration and cost, a more or less negative trend is observable. Additionally on both plots the data has the shape of a cone. The reason for this is made visible on the duration plot, as it depends on the two significant dates *today* and *data collection start* (see section 2.1.3, Structure). As closed projects are already finished, they can only last until *today*. Thus it is logical that the mean project duration tends to increase when going back in time from *today*. Due to the meaning of *data collection start* also projects before that date tend to last longer then before. The same conclusions apply to project cost, as `dur_act` and `cost_act` of closed projects with all WS available are correlated ($r = 0.54$ and 0.56 excluding outlier project 272).

Because of that using only closed projects leads to a biased trend. As already mentioned, for analyzing trend closed and running projects will be used. For running projects a model based on the closed projects will be generated, to get an unbiased fit for project cost and duration. This overcomes also the problem corresponding to the date *today*. The approach to prevent the bias based on the *data collection start* is simply to regard only projects started after tis date.

Remark 2.12 The above arguments are also the reason for not using project start as a predictor variable for modeling actual cost and duration.

2.3.2. Methods Overview

Here the methods used to model cost and duration trend are described and presented in fig. 2.19. The choice of methods is based on the structure of the data, as described in the previous section 2.3.1, Motivation.

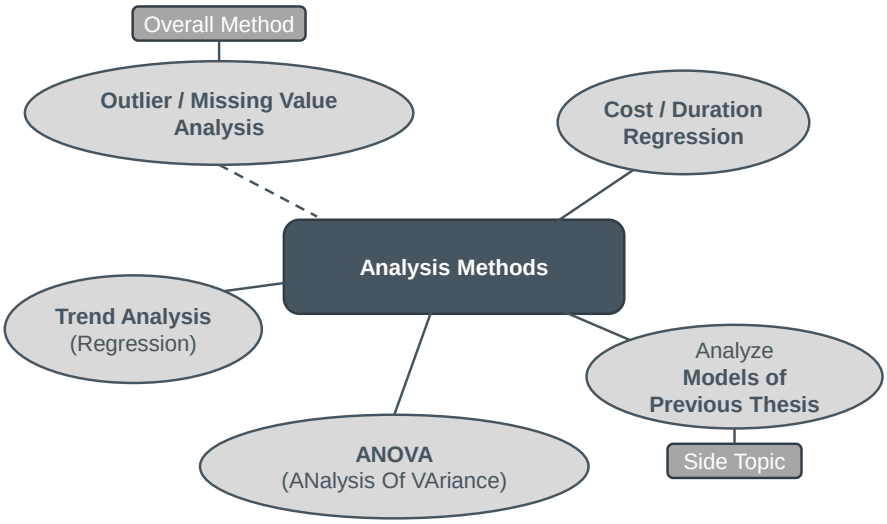


Figure 2.19.: Overview of analysis methods

Cost/Duration Regression
To get unbiased estimations for project cost and duration of running projects, closed projects are used. Based on closed projects with all WS available multiple linear regression models are generated.

Analyze Models of Previous Thesis (side topic)
As a side topic to the main research objectives it is of interest to analyze the models of the previous thesis and apply them to the new dataset (see Sponer [21]).

Analysis of Variance (ANOVA)
The first step in analyzing trend is to perform an ANOVA on project cost and duration by project start year.
As the application of this analysis did not result in significant results of interest, it is not presented within this thesis.

Trend Analysis (Regression)
To get a detailed analysis of the trends a regression analysis of project cost and duration with project start as predictor is performed. On the same time different subsets of the data are analyzed to find out which project categories are responsible for any trend found.

Outlier/Missing Value Analysis (overall method)
Due to the structure of the data, missing values and outliers play an important role in all parts of the analysis. So this fact has always be taken into account.

Remark 2.13 The intention of ANOVA was to identify yearly differences in cost and duration. For cost/duration ANOVA compares these characteristics by each year and shows significant differences, if found. Here no significant results of interest were found.

2.4. Modeling Cost and Duration

Actual cost and duration are modeled for closed projects with all WS available based on the information of the first WS. Hence only variables with extension `_f` or `_est` may be used. The purpose is to apply these models to running projects to get an unbiased prediction for actual values.

There are $m = 16$ variables used as possible predictors for modeling. These predictor variables are all variables of the first WS, excluding estimated cost as it is a linear combination of the other cost variables. Of course also the variables `proj_start` (would cause bias, as explained in section 2.3, Modeling Approach) and status (only closed projects are used) are not useful as predictor variables.

The upcoming section 2.4.1 shows an example and practical details about the process of model selection. The corresponding theory is covered by section 3.4. Sections 2.4.2 and 2.4.3 present details about the actual modeling process for actual project cost and duration based on the first WS. The results are summarized in section 2.4.5.

2.4.1. Model Selection Process

The core function used for model selection is the `regsubsets` function of the `leaps` package (see LUMLEY [15]). Basically this function takes a set of variables (one response variable and a set of predictor variables) and returns the "best models" for 1 to $p_{\max}$ predictor variables (not counting the intercept), whereas $p_{\max}$ can be specified.

When selecting variables with the `regsubsets` function different aspects have to be considered. The two main points are as follows:

1. **Predictor variables have to be numerical**
   Thus numerical substitutions for categorical variables based on section 2.2.2, Reducing Category Levels are applied.

2. **`NA`, `NaN` or `Inf` values lead to errors**
   `NaN` (not a number, e. g. 0/0) and `Inf` (infinity) values do not occur in the datasets used. To avoid errors because of not available (`NA`) values, all data entries (i. e. projects) containing `NA` values have to be omitted. How many and which projects are excluded from the dataset depends on the used variables and the structure of missing values (see section 2.2.4, Missing Values).
   Here it was chosen to start the `regsubsets` model selection with all variables. If variables containing `NA` values can be identified as not influential, they are excluded from the set of possible predictors to start the model selection newly. If variables with missing values stay significant, the models found are compared to models without these variables holding `NA` values.

For more detailed information about the usage of the `regsubsets` function, please see the **R** help manual.

Remark 2.14 Generally there is no overall "best model" for modeling a response variable based on multiple predictor variables. Different criteria exist to help on deciding which model to chose. The `regsubsets` function can be performed using different methods, on which the model selection depends. Unless otherwise specified the exhaustive method is used. This method compares all models given a fixed number $p - 1$ of predictors (excluding the intercept). Within this set of models it is possible to decide for the best model, which then is returned by the function. For many variables this leads to intensive calculations as m^{p-1} models have to be compared.

When fitting a multiple regression model with m basic variables the question arises if transformed variables or interactions have significant influence. The approach within this thesis is as follows:

Predictor Transformation

Typical transformation functions include the square function, the logarithm and the exponential function. Here the natural logarithm (will be denoted as `log`) and the square function are used. Not using the exponential function results as together with the square function this resulted in linear dependencies. The mode chosen is to apply the `regsubsets` function on the set of basic variables together with the transformed variables.
Transformations are applied to variables, where reasonable (e. g. not to categorical variables).

Interactions

m basic predictor variables and two transformations for each variable result in $3m$ predictors. Counting all predictor variables and their interactions gives approximately $(3m)^2 = 9m^2$ ($= 2304$ for $m = 16$) possible predictor variables. As this leads to extensive computing time and to keep models simple, models are generated on the basic predictor variables and its transformations only. This gives models without interactions. To not loose the potential of interactions, these models are analyzed on significance of interactions of the left significant predictor variables within the model.

Model Variety

It is preferable that different models are taken into account, as there is no single best model. Here the approach is to generate two models for project cost and duration each. For each response variable two models are generated based on predictors with and without transformations.

Remark 2.15

- To refer to transformed variables the extensions `_sq` and `_log` shall be used for square root and logarithmic transformations respectively.

- Here in practice instead of the pure logarithmic transformation $\log(x)$ the shifted version $\log(1+x)$ is used. The reason is that some variables hold also the value zero. This also results in the property of non-negative transformed variables, because all original variables are nonnegative (see section 2.1.2, Variables)[9].

[9] $x \geq 0 \Rightarrow \log(1+x) \geq 0$, because the logarithm $\log(x)$ is a monotone increasing function with root at $x = 1$

Response Transformation

To overcome problems with violations of linear regression assumptions – especially non constant error variance (heteroscedasticity) and deviations from normal distribution – a transformation of the response variable may be helpful. A common method is the Box-Cox transformation (see Friedl [9, 10] and Stadlober [23]). For theoretical details about the Box-Cox Transformation see section 3.3, Model Diagnostics.

For modeling actual project cost and duration response transformations will be needed. The step of finding out which Box-Cox transformation to use and applying the transformation is subsequent to the model selection. Applying the transformation results in the need of a newly performed model selection, as other variables may become significant. To not go beyond the scope of this thesis the process of model selection will be shown with already transformed responses.

Foregoing analysis showed that the response transformation $f(y) = y^{1/3}$ is appropriate. The following example of a model for actual cost with estimated cost as the only predictor shows briefly how to obtain such a transformation:

Example 2.1 The simple model `cost_act ~ cost_est` is taken as an example:

```
> summary(cost_act ~ cost_est, data = cl_pr_all_WS)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7664 -0.0988 -0.0338  0.0370  1.4013
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.0139     0.0220   -0.63     0.53
cost_est      1.4033     0.0566   24.80   <2e-16 ***
---
Residual standard error: 0.228 on 225 degrees of freedom
Multiple R-squared: 0.732,      Adjusted R-squared: 0.731
F-statistic:  615 on 1 and 225 DF,  p-value: <2e-16
```

The residual plots presented in fig. 2.20 show, that the residual variance depends on the fitted values (i. e. the variance is not constant). On the same time the normal quantile–quantile plot (Q–Q plot) as well as the histogram show that the residuals are not normally distributed. This is underlined by the Shapiro-Wilk test[10] with a p-value $< 2.2 \cdot 10^{-16}$.

On the same time by the residual plots in fig. 2.20 project 272 can be identified as outlier. On the three plots on top only the three outliers with highest absolute (standardized) residuals are marked. Nevertheless the rightmost point on the

[10] The Shapiro-Wilk test tests for normal distribution (see section 3.3, Model Diagnostics).

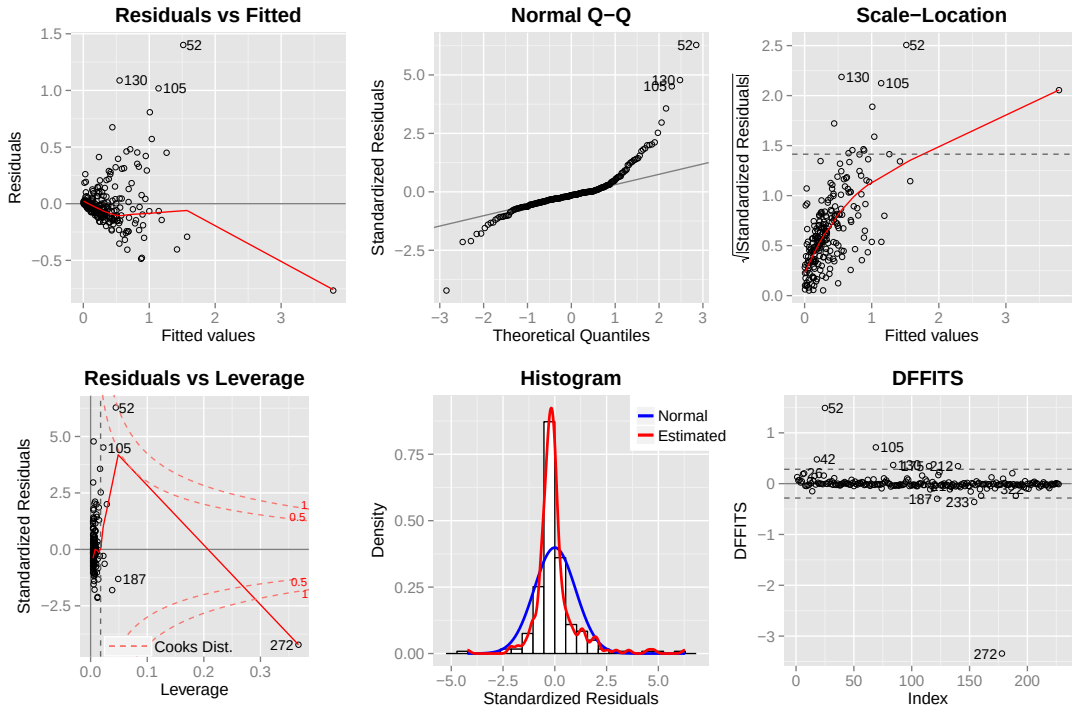


Figure 2.20.: Residual plot of `cost_act ~ cost_est` (data: closed projects with all WS available)

"Residual vs Fitted" and the "Scale-Location" plots are confirmed to represent project 272.

The Box-Cox transformation for the simple model `cost_act ~ cost_est` (see fig. 2.21) gives for the transformation parameter λ a 95% CI of about $[0.26, 0.41]$. The maximum log-likelihood estimation is about 0.33. Thus $\lambda = 1/3$ is chosen.

In addition to the transformation outlier project 272 is excluded giving the following model:

```
> summary(lm(cost_act^(1/3) ~ cost_est, data = subset(df_predfc,
+   !(id == 272)))

Residuals:
     Min       1Q   Median       3Q      Max
-0.3320 -0.0685 -0.0062  0.0538  0.4162
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.4014     0.0119    33.7   <2e-16 ***
cost_est      0.8989     0.0345    26.1   <2e-16 ***
---
Residual standard error: 0.111 on 224 degrees of freedom
Multiple R-squared: 0.752,      Adjusted R-squared: 0.751
F-statistic:  680 on 1 and 224 DF,  p-value: <2e-16
```

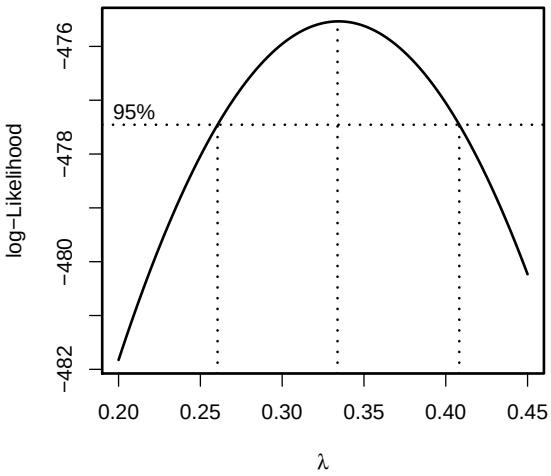


Figure 2.21.: Box-Cox transformation plot for the model `cost_act ~ cost_est` (data: closed projects with all WS available)

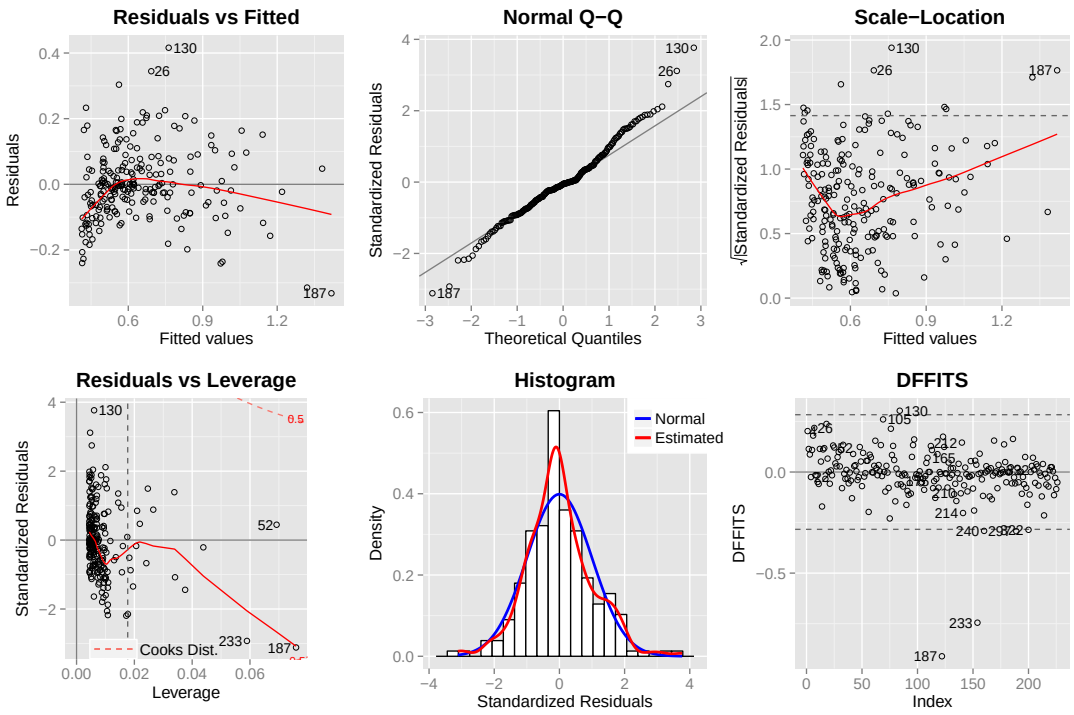


Figure 2.22.: Residual plot of `cost_act^(1/3) ~ cost_est` (data: closed projects with all WS available)

Now the distribution of the residuals looks much more like normal distribution (see fig. 2.22), which also results in a better Shapiro-Wilk p-value of 0.0052. The variance structure of the residuals is better, but still not satisfactory. The resulting model can of course be improved, as there are also some outliers to be looked at more in detail.

The purpose of this example is to show an example for a Box-Cox transformation and the origin of the transformation parameter $\lambda = 1/3$.

Remark 2.16

- The **R**-print of linear model (`lm`) summary in example 2.1 and following prints include a coding of significance levels for the p-values:
  `0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`
  (The level code denotes that the p-value is within the range of its surrounding numbers). The significance levels are used to get a quick overview of coefficient significance.

- As seen in example 2.1 project 272 is a cost outlier. In fact this project is an outlier regarding project cost, material cost and single chip cost. Because of that project 272 will be excluded from the data used to model actual project cost from the beginning.

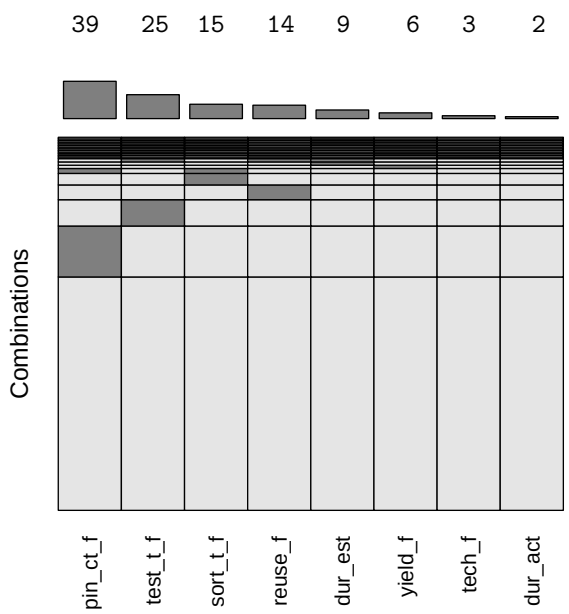
- The residual plots presented within this thesis (e. g. fig. 2.20 and fig. 2.22) are produced by the self written function `GGplotLm`. For more information see appendix C, Self Written **R** Functions.

Missing Values

As stated missing values are important for the model selection process. In section 2.2.4, Missing Values an aggregation plot for the missing values for projects with all WS available can be found. Here it is needed to know the missing value structure for the closed projects with all WS available only (see fig. 2.23). When referring in this section to variables that hold missing values, this figure will be the source of information.

2.4.2. Modeling Actual Project Cost based on First Workstatement

As stated in remark 2.16, for modeling actual project cost project 272 is excluded from the closed projects with all WS available.




```
combinations
   count percent cumsum_pct
1    142   62.56      62.56
2     31   13.66      76.22
3     16    7.05      83.27
4      9    3.96      87.23
5      7    3.08      90.31
6      3    1.32      91.63
7      2    0.88      92.51
...
```

Figure 2.23.: Aggregation plot of missing values with **R**-output table (data: closed projects with all WS available)

The generation of the two types of cost models, with and without transformed predictor variables, is presented in a summarized form. Each single step of decisions cannot be presented, but the main steps are demonstrated or at least remarked.

Without Transformed Predictors

The model selection process starts with all variables available. As mentioned the `regsubsets` function requires a dataset without `NA` values. Thus all projects containing `NA` values have to be omitted. The resulting project numbers are as follows:

```
           nrow pct[11]
  original  226 100  (Original data set including NA values)
NA omitted  142  63  (Original dataset without projects that hold NA values)
difference   84  37  (The projects that are omitted because of NA values)
```

As only 63% of the projects are left when using all variables, it is of interest to identify variables holding `NA` values that are not significant.

Omit non-significant variables holding `NA` values: Based on the output of the `regsubsets` function (see fig. 2.24) applied on the dataset with omitted `NA` values it can be seen that `sort_t_f`, `test_t_f` and `pin_ct_f` are variables with missing values that are not considered in any model. These variables can now be

[11] `nrow` = number of rows, `pct` = percent

excluded from the subset regression to use a bigger dataset. By the same method also `tech_f` could be identified as non significant variable.

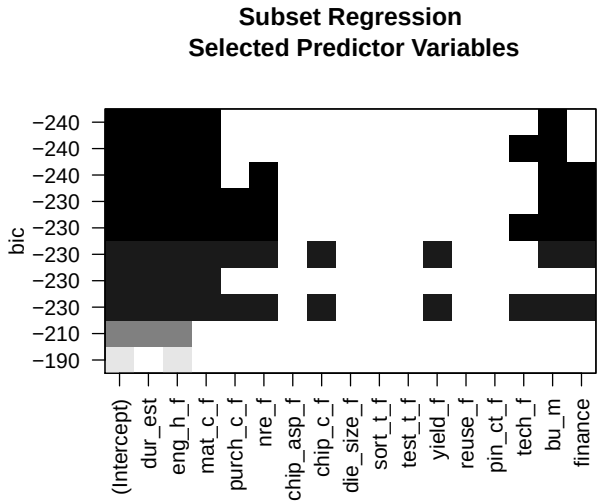


Figure 2.24.: Selected predictor variables of subset regression[12] (data: closed projects with all WS available)

Remark 2.17 As excluding only one variable already causes a bigger dataset it may follow that a variable with `NA` values that was not significant, is now considered in a model. Because of that the exclusion of variables with `NA` values is always verified stepwise. This means that only one variable is excluded at a time and then it is again checked if the other variables can still be excluded. The variable to exclude is always chosen as the one which holds the most missing values.

Looking at the subset regression of the remaining dataset (see fig. 2.25) there is no variable with `NA` values not considered in any model.

But not all of the 8 models have to be useful models, as to many variables may result in overfitting. The model selection criteria can help on deciding which number of predictors give a good model (see fig. 2.26). Based on the criteria and on a deeper look into interesting models, the set of 4 predictors can be chosen as a good model. For $p - 1 = 4$ BIC reaches its minimum and for the other criteria there is not much improvement for $p - 1 > 4$. The predictors of the models with up to 4 predictors are listed in the following (they can also be readout of fig. 2.25):

[12] A row of this plot represents one model and is marked by its BIC-value. The squares on the plot have two meanings. On one hand non-white colors correspond to the BIC-value. On the other hand any non-white color means that the respective variable is a predictor in the model.

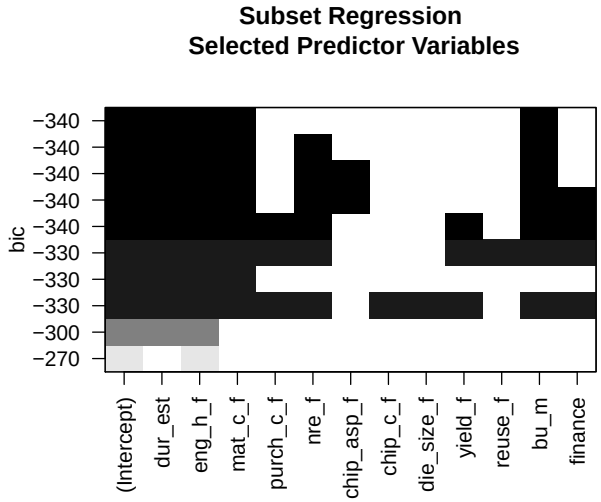


Figure 2.25.: Selected predictor variables of subset regression (data: closed projects with all WS available)

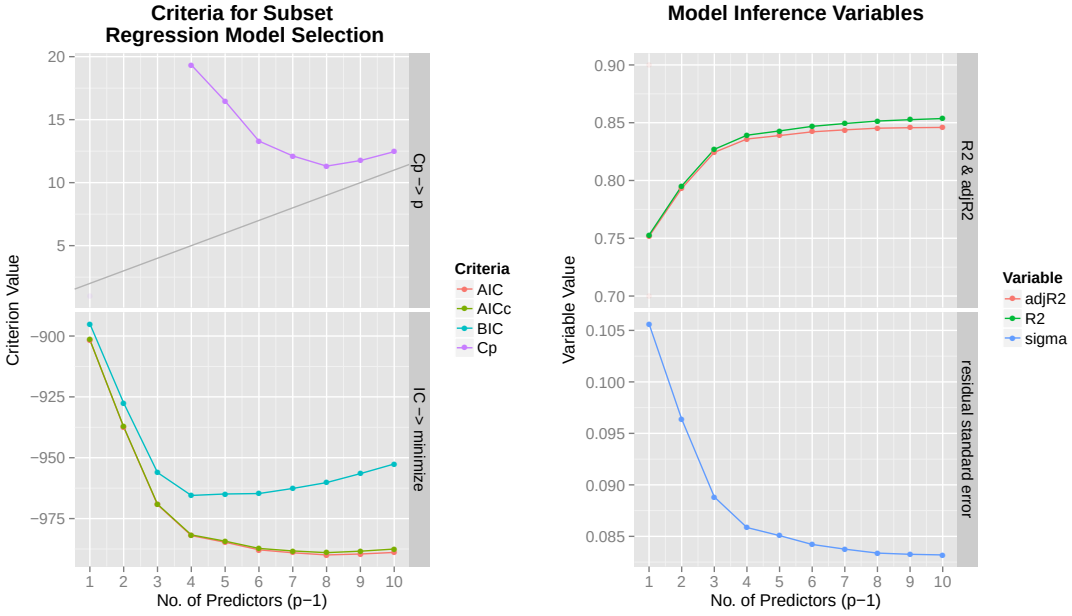


Figure 2.26.: Model Selection Criteria[13] (data: closed projects with all WS available)

```
Pred. Mod. 1 : eng_h_f
Pred. Mod. 2 : eng_h_f + dur_est
Pred. Mod. 3 : eng_h_f + dur_est + mat_c_f
Pred. Mod. 4 : eng_h_f + dur_est + mat_c_f + bu_m
```

Obviously `reuse_f` and `yield_f` do not occur in any of the models. Thus these predictors holding `NA` values can be omitted from the set of possible predictors. The only leftover predictor holding `NA` values (9) is `dur_est`. It can be shown, that omitting `dur_est` does not improve the models.

Specify model to work with: On the remaining variables a subset regression is performed again (see fig. 2.27). Mallows C_p criterion supports 6 predictors and on the same time all other criteria do not contradict this. The model with 6 predictors looks as follows:

```
> lm_c1 <- lm((cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f +
+   purch_c_f + nre_f + bu_m, data = cl_pr_all_WS)
> summary(lm_c1)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2320 -0.0576 -0.0158  0.0473  0.3671
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.28e-01   3.11e-02   13.77  < 2e-16 ***
dur_est      1.38e-01   2.37e-02    5.82  2.1e-08 ***
eng_h_f      7.94e-05   5.19e-06   15.31  < 2e-16 ***
mat_c_f      8.74e-01   1.39e-01    6.30  1.8e-09 ***
purch_c_f    4.06e-01   8.50e-02    4.78  3.3e-06 ***
nre_f       -1.13e-01   5.46e-02   -2.07     0.04 *
bu_m        -5.96e-02   1.48e-02   -4.02  8.0e-05 ***
---
Residual standard error: 0.0897 on 210 degrees of freedom
  (9 observations deleted due to missingness)
Multiple R-squared: 0.838,      Adjusted R-squared: 0.833
F-statistic:  181 on 6 and 210 DF,  p-value: <2e-16
```

`nre_f` is with a p-value of 0.04 not strongly significant, but it is kept in the model as this may change by further adjustments. It is also notable that due to the usage of `dur_est` 9 projects with missing values are deleted. To verify the model, the residual plot can be used (see fig. 2.28). The distribution of the residuals (see normal Q-Q plot and histogram) seems to be close to normality, but may be improved (Shapiro-Wilk test p-value = 0.0015). Especially project 130 is an outlier. The assumption of constant variance (see residuals vs. fitted and scale-location plots) is also not yet optimal. Beside the outlier project 130, project 187

[13] This plot summarizes the main model selection criteria in a single graphic. For each number of predictors (which is p-1, as the intercept is not counted) the value of each criterion is denoted. The values are connected to a line for each criterion.

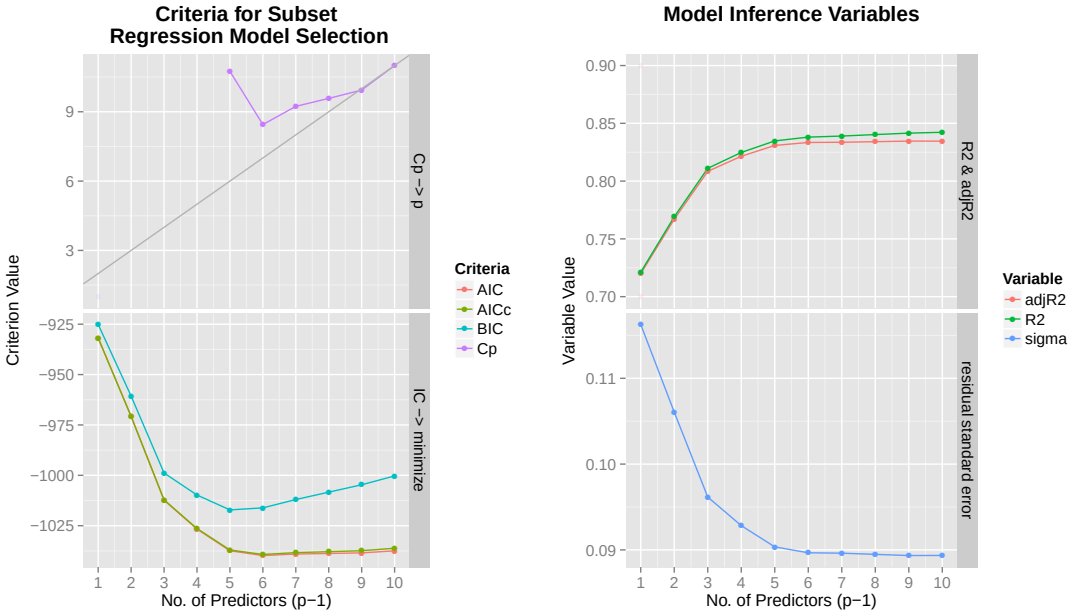


Figure 2.27.: Model Selection Criteria (data: closed projects with all WS available)

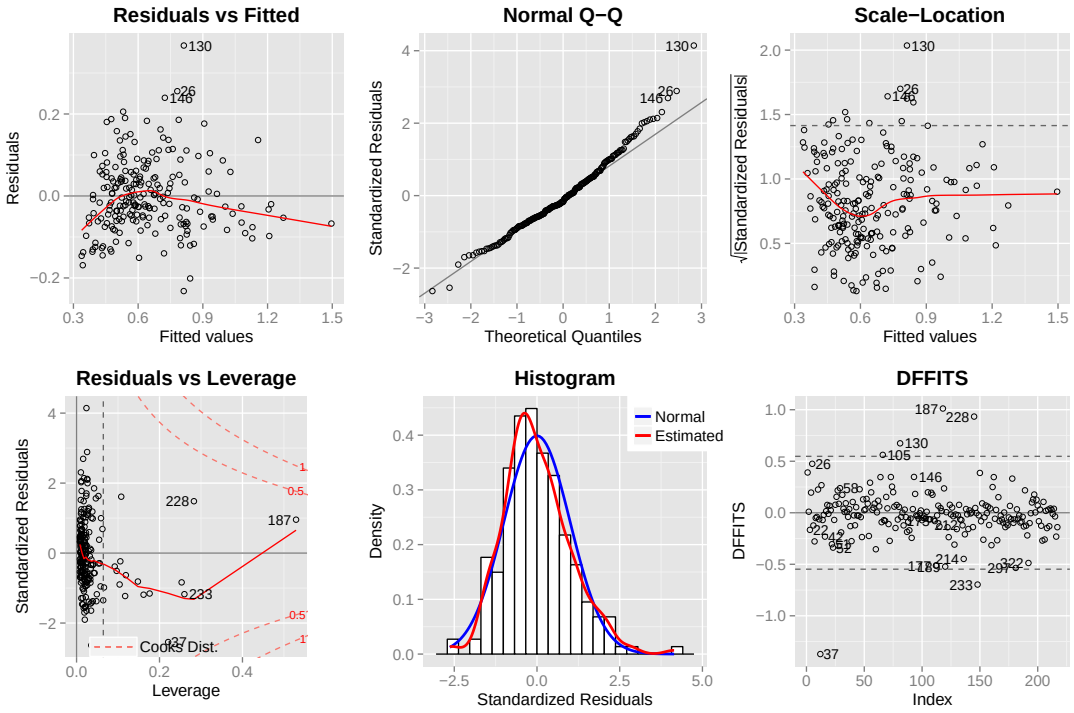


Figure 2.28.: Residual plot `lm_c1` (data: closed projects with all WS available)

seems to be influential and project 37 seems to have a high influence on the fitted values.

Further model analysis is briefly discussed in the following:

- **Outlier Elimination:** Found outliers have to be analyzed and eliminated if necessary. This process is done stepwise when outliers occur.
- **Model Confirmation:** After each step that changes the database, the model is confirmed. Here this means that it is reviewed if the same or other variables are significant.
- **Interactions:** The resulting model is analyzed on interactions using Analysis of Variance. First the previously found model is compared to the same model including all possible interactions of two variables:

```
Analysis of Variance Table

Model 1: (cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f + purch_c_f +
    nre_f + bu_m + dur_est:eng_h_f + dur_est:mat_c_f + dur_est:purch_c_f +
    dur_est:nre_f + dur_est:bu_m + eng_h_f:mat_c_f + eng_h_f:purch_c_f +
    eng_h_f:nre_f + eng_h_f:bu_m + mat_c_f:purch_c_f + mat_c_f:nre_f +
    mat_c_f:bu_m + purch_c_f:nre_f + purch_c_f:bu_m + nre_f:bu_m
Model 2: (cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f + purch_c_f +
    nre_f + bu_m
  Res.Df    RSS  Df  Sum of Sq      F Pr(>F)
1    184 1.0232
2    199 1.2415 -15    -0.2183 2.6174 0.0013 **
```

  The p-value of 0.0013 indicates not all interactions can be treated as non-significant. When adding the most significant interaction `dur_est:mat_c_f` to the model without interactions, the ANOVA results in:

```
Analysis of Variance Table

Model 1: (cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f + purch_c_f +
    nre_f + bu_m + dur_est:eng_h_f + dur_est:mat_c_f + dur_est:purch_c_f +
    dur_est:nre_f + dur_est:bu_m + eng_h_f:mat_c_f + eng_h_f:purch_c_f +
    eng_h_f:nre_f + eng_h_f:bu_m + mat_c_f:purch_c_f + mat_c_f:nre_f +
    mat_c_f:bu_m + purch_c_f:nre_f + purch_c_f:bu_m + nre_f:bu_m
Model 2: (cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f + purch_c_f +
    nre_f + bu_m + dur_est:mat_c_f
  Res.Df    RSS  Df  Sum of Sq      F Pr(>F)
1    184 1.0232
2    198 1.0826 -14    -0.0594 0.7637 0.7073
```

  Now the p-value is 0.7073, so all other interactions can be disregarded.

- **Categorical Variables:** The previous procedure involves interactions with numerical variables only. Thus on the next step interactions with the categorical variables (before category reduction, see section 2.2.2, Reducing Category Levels) is performed.

If a categorical variable is left in the model, now it is replaced by the corresponding full categorical variable. In this case `bu_m` is replaced by `bu`.

Specifying the Final Model: The result of the above described steps is the final model `lm_c` for actual project cost:

```
> excl_c <- c(26, 37, 39, 42, 105, 130, 146, 177, 187, 228, 272, 290)
> lm_c <- lm((cost_act)^(1/3) ~ dur_est + eng_h_f + mat_c_f +  bu +
+   dur_est:mat_c_f, data = subset(cl_pr_all_WS, !(id %in% excl_c)))
> summary(lm_c)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17133 -0.05261 -0.00549  0.04750  0.21642
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.12e-01   3.08e-02   10.15  < 2e-16 ***
dur_est          2.69e-01   3.12e-02    8.62  2.2e-15 ***
eng_h_f          7.60e-05   4.66e-06   16.30  < 2e-16 ***
mat_c_f          2.68e+00   3.11e-01    8.61  2.3e-15 ***
buBU2           -5.09e-02   2.56e-02   -1.99  0.04830 *
buBU3           -1.59e-01   5.21e-02   -3.06  0.00252 **
buBU4           -1.01e-01   2.55e-02   -3.94  0.00011 ***
dur_est:mat_c_f -2.20e+00   3.71e-01   -5.94  1.3e-08 ***
---
Residual standard error: 0.0751 on 198 degrees of freedom
  (9 observations deleted due to missingness)
Multiple R-squared: 0.868,      Adjusted R-squared: 0.864
```

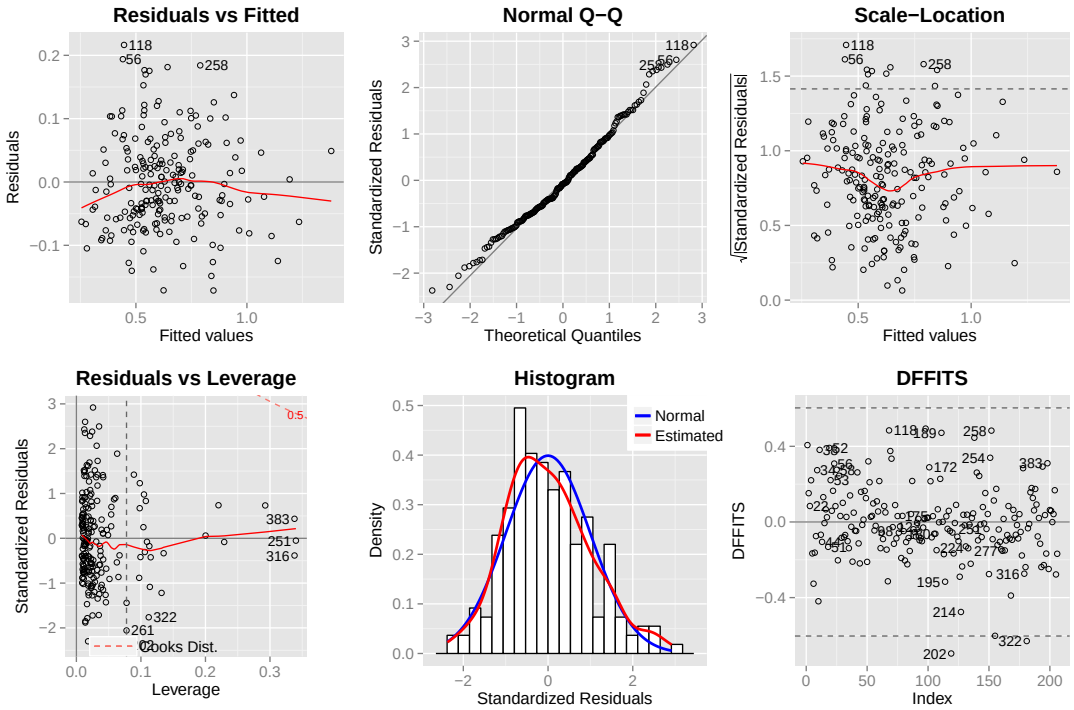
Remark 2.18

- The projects saved to `excl_c` are the identified outliers.
- The model name `lm_c` represents "linear model cost".

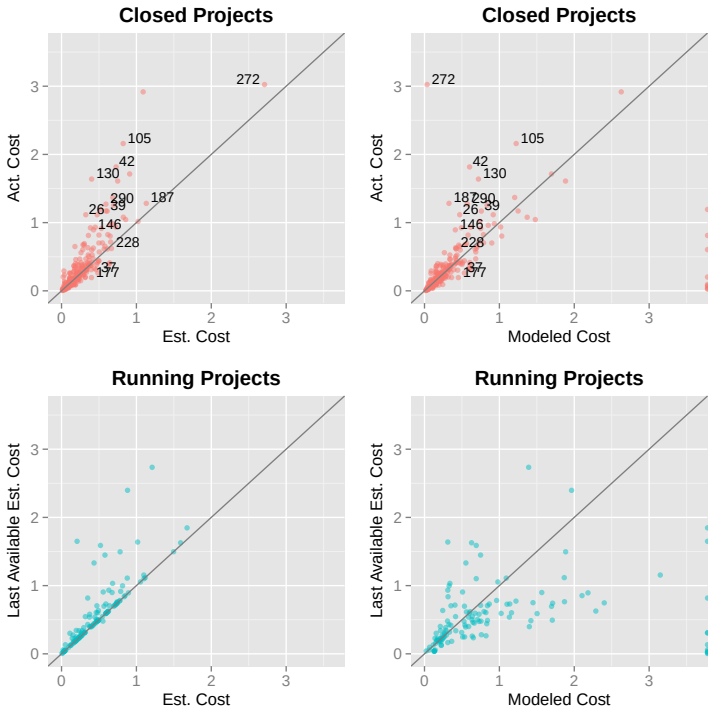
The residual plot of the final model `lm_c` (see fig. 2.29a) still shows some outliers. Especially the Q-Q plot shows some outliers. In addition the histogram makes clear that there is a skewness of residual distribution. On the same time this skewness is very slight and the Q-Q plot shows a linear and relatively symmetric structure, in particular when compared to the last residual plot. Cooks distance and leverage give no real influential points as well as the other plots.

As seen before cost is difficult to describe accurately, therefore this model can be regarded to give good predictions and can be used as an unbiased model for actual project cost.

[14] The points on the far right hand side of each plot (only the half point is visible) represent projects for which the model cannot fit values due to missingness of values for any predictor variable.



(a) Residual plot



(b) Model plot[14]

Figure 2.29.: Final cost model `lm_c` without transformed predictors (data: closed projects with all WS available)

Apply to Running Projects: Fig. 2.29b compares the estimated cost to modeled cost (fitted values of `lm_c`) on the two upper plots. It can be clearly seen that the modeled costs are an unbiased predictor for the actual cost. The two plots on the bottom show how estimated and modeled cost relate to the last available estimation. The last available estimated cost is larger then the estimated cost at project start for many projects. On the same time modeled cost tend to give higher values than the last available estimation, i. e. the model says that the cost will increase in the future.

Remark 2.19 In the 4^{th} plot of fig. 2.29b the model indicates for some running projects that the last available estimation is too high. On the other hand in fig. 2.17 in section 2.3, Modeling Approach it turned out that estimations (at project start) tend to underestimate actual values. Thus the model `lm_c` may result in too low predictions for some running projects. This issue is addressed in section 2.5, Model Improvements.

With Transformed Predictors

The same procedure as described without transformed predictors is now applied to variables and their transformations. Hence just the main steps and differences to the above model selection are described.

Here, instead of the 16 basic predictor variables, $m = 42$ predictor variables are used. As the calculation time for exhaustive subset regression is exponential in the number of predictors $p - 1$ with base m, this leads to considerable calculation times. For 16 predictors this was easily possible to handle. For 42 predictors the calculation times quickly get extensive for often repeated calculations, as table 2.6 shows[15],[16].

Table 2.6.: Calculation times

$\max(p-1)$	calculation time
8	3 sec.
9	9 sec.
10	21 sec.
11	42 sec.
12	89 sec.
13	140 sec.

As a solution it was chosen to use a two step procedure:

[15] calculations were run on a laptop with the following specifications: Windows 7 Enterprise, 4GB RAM, Intel i5 CPU with 2.67GHz

[16] Also Kleinbaum et al. [14] suggest to use exhaustive search (all-subset-selection) only for $m < 40$ predictor variables (see section 3.4, Model Selection).

1. Exhaustive `regsubsets` with $\max(p-1) = 8$. Predictors that are element of at least 4 models are forced to be included in the models of the next step.

2. Exhaustive `regsubsets` with $\max(p-1) = 15$ and forced in predictors of the first step.

Remark 2.20 The numbers of the above two steps and the numbers in table 2.6 do only apply for a special case. Concrete numbers are taken here for illustration purpose.

Omit non-significant variables holding `NA` values: Applying the `regsubsets` model selection to the full dataset and up to $\max(p-1) = 8$ predictors (see fig. 2.30) indicates that more predictors could give better models. The following variables are part of at least 4 of these models:
`eng_h_f, bu_m, dur_est_log, mat_c_f_log`

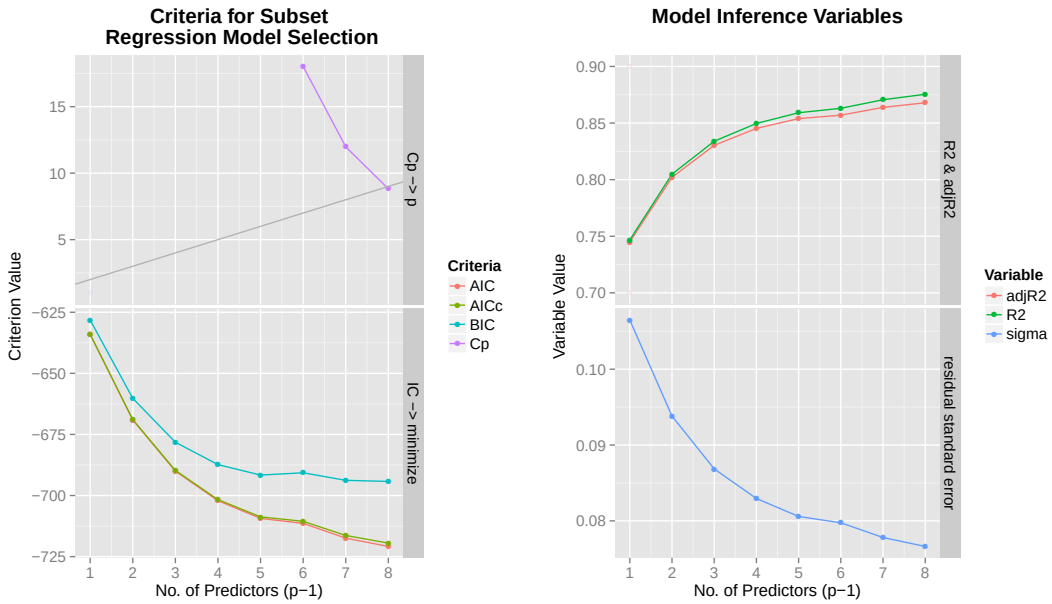


Figure 2.30.: Model Selection Criteria (data: closed projects with all WS available)

These variables are forced to be in the models of the next `regusbsets` step with up to 15 models. The corresponding plot of the selected predictor variables shows this clearly (see the left most columns of fig. 2.31). Here it can also be seen that `pin_ct_f` and its transformations – the variable(s) with most missing values – are not predictors of any model. By the same procedure, which is performed stepwise, also `test_t_f`, `sort_t_f`, `reuse_f` and `tech_f` (with its transformations) could be identified as not being significant. The predictor `yield_f` could not be excluded by this procedure, which results in 14 projects that can not be considered.

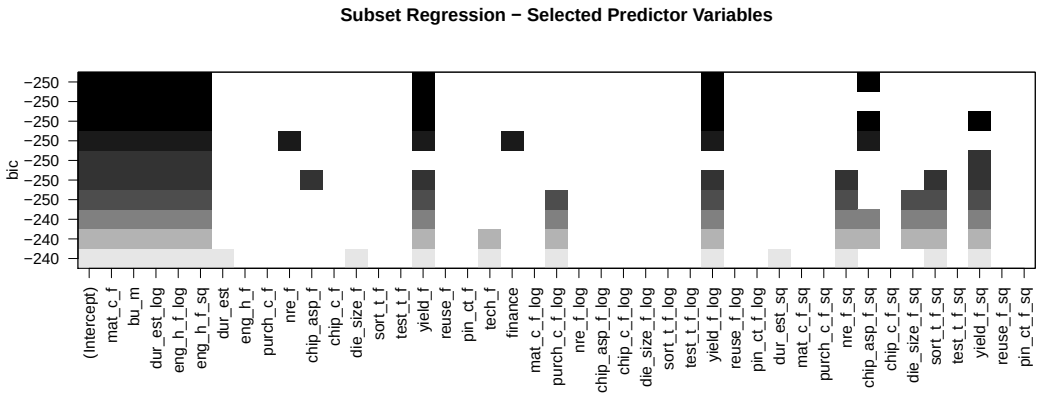


Figure 2.31.: Selected predictor variables of subset regression (data: closed projects with all WS available)

Specify model to work with: On the last step of omitting non-significant variables holding `NA` values a model with $p - 1 = 14$ predictors was chosen:

```
> lm_ct1 <- lm(formula = (cost_act)^(1/3) ~ ..., data = cl_pr_all_WS)
> summary(lm_ct1)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2132 -0.0524 -0.0058  0.0390  0.3670
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.32e+02   3.65e+01   -3.61  0.00039 ***
eng_h_f      6.27e-05   6.32e-06    9.93  < 2e-16 ***
mat_c_f     -1.40e+02   5.32e+01   -2.62  0.00939 **
purch_c_f    4.43e-01   7.64e-02    5.80  2.6e-08 ***
chip_c_f    -7.24e-02   2.14e-02   -3.38  0.00089 ***
die_size_f   4.36e-03   1.12e-03    3.91  0.00013 ***
yield_f     -1.11e+00   3.16e-01   -3.52  0.00054 ***
bu_m        -5.99e-02   1.44e-02   -4.15  5.0e-05 ***
dur_est_log  1.72e-01   4.32e-02    3.98  9.6e-05 ***
eng_h_f_log  3.91e-02   8.64e-03    4.53  1.0e-05 ***
mat_c_f_log  1.42e+02   5.38e+01    2.65  0.00874 **
yield_f_log  4.54e+01   1.27e+01    3.59  0.00041 ***
mat_c_f_sq   5.43e+01   2.11e+01    2.58  0.01064 *
nre_f_sq    -1.88e-01   7.60e-02   -2.47  0.01446 *
yield_f_sq   3.37e-03   9.78e-04    3.44  0.00070 ***
---
Residual standard error: 0.0786 on 197 degrees of freedom
  (14 observations deleted due to missingness)
Multiple R-squared: 0.88,       Adjusted R-squared: 0.872
F-statistic:  104 on 14 and 197 DF,  p-value: <2e-16
```

14 variables in the model `lm_ct1` are a noticeable number of predictors. According to model selection criteria (not shown here) also a model of 6 predictors could have

been chosen without much worse criteria values. The larger model was preferred to work with, as a wider range of interactions can be considered.

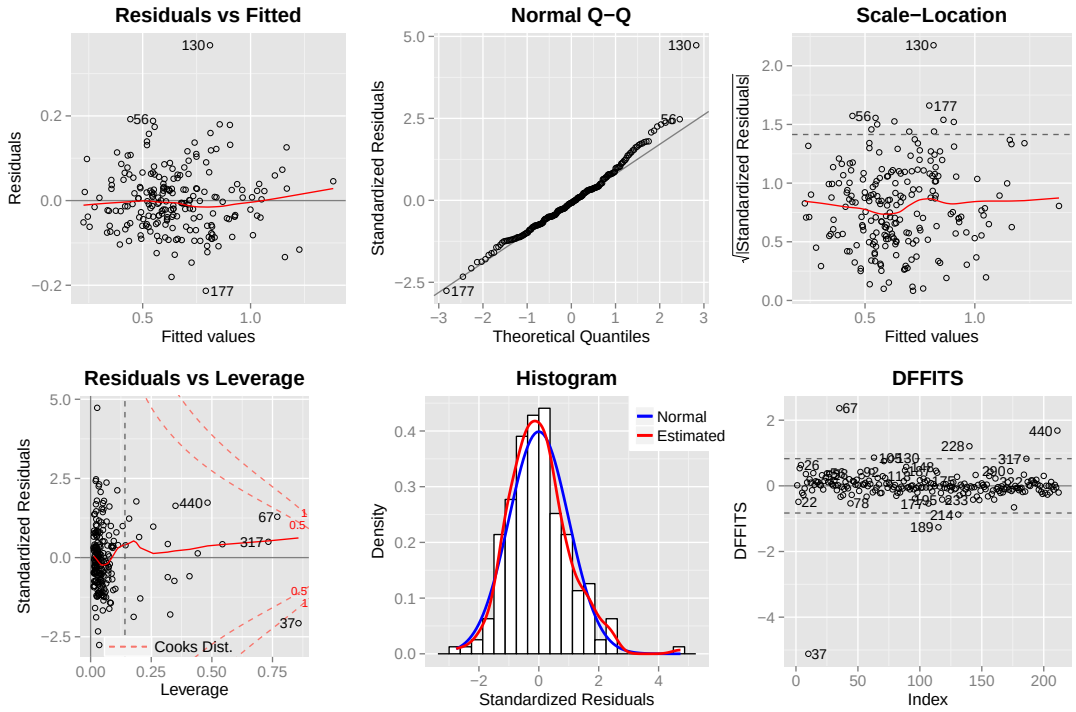


Figure 2.32.: Residual plot of model `lm_ct1` (data: closed projects with all WS available)

The residual plot of `lm_ct1` (see fig. 2.32) already shows a relatively constant variance and a relatively good adaption to normal distribution (except for outlier 130). On the first sight two outliers (37 and 130) are obvious, which requires more detailed analysis.

Further model analysis: Basically the same steps as for the model without transformed predictor variables are performed. The main changes are as follows

- **Interactions:** By Analysis of Variance (ANOVA) all interactions not including `dur_est_log` could be identified as non-significant with a p-value of 0.43. To obtain the significant interactions a `regsubsets` model selection is applied on all original predictors and their interactions with `dur_est_log` (see fig. 2.33). A bend in AIC, AIC_c and BIC can be seen for $p-1=6$ predictors. On the same time better criteria values can be obtained by more predictors, even for BIC that prefers simpler models. As for untransformed predictors a relatively simple model was finally chosen, here the more complicated model with $p-1=15$ predictors is chosen (minimizes BIC).

- **Categorical Variables:** The only left predictor, that is based on a categorical variable is `bu_m` in interaction with `dur_est_log`. As `bu_m` is replaced

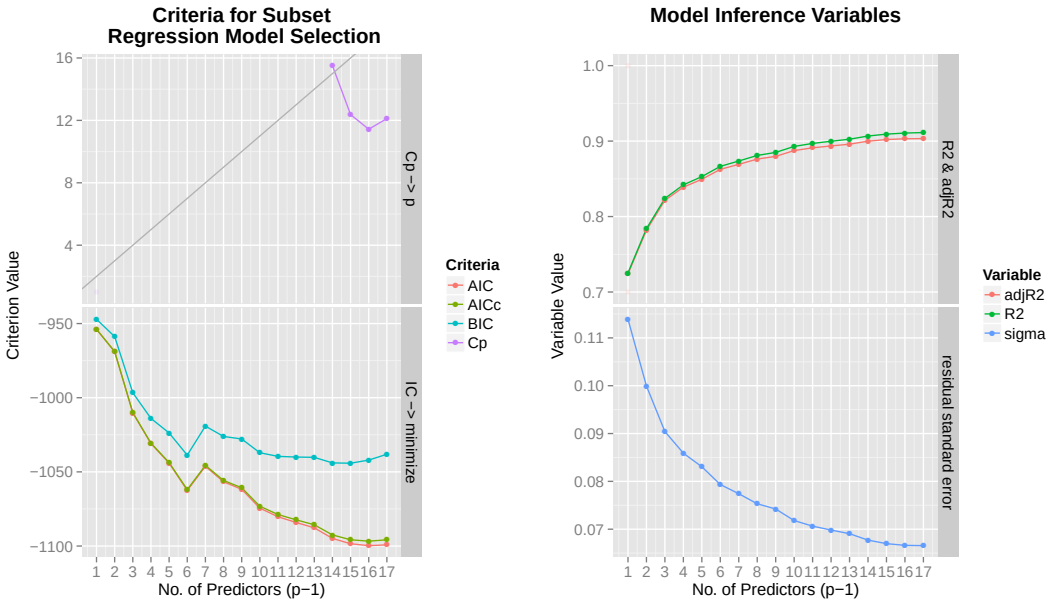


Figure 2.33.: Model Selection Criteria (data: closed projects with all WS available)

by `bu`, only the interaction with BU4 is left with a significant difference to BU1[17].

Specifying the Final Model: The result of the above described steps is the final model for actual project cost[18]:

```
> excl_ct <- c(37, 67, 130, 189, 272, 317, 440)
> lm_ct <- lm(formula = (cost_act)^(1/3) ~ ...,
+   data = subset(cl_pr_all_WS, !(id %in% excl_c)))
> summary(lm_ct)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17983 -0.04793  0.00108  0.04006  0.19049
Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       1.45e-01   3.09e-02    4.68  5.4e-06 ***
eng_h_f                           5.24e-05   9.14e-06    5.73  3.8e-08 ***
mat_c_f                          -5.08e+01   9.18e+00   -5.53  1.0e-07 ***
purch_c_f                         4.71e-01   6.90e-02    6.82  1.1e-10 ***
chip_c_f                         -1.02e-01   3.24e-02   -3.14   0.0019 **
die_size_f                        5.95e-03   1.29e-03    4.62  6.9e-06 ***
log(1 + dur_est)                 -3.80e+02   8.06e+01   -4.72  4.6e-06 ***
log(1 + mat_c_f)                  5.78e+01   9.96e+00    5.80  2.7e-08 ***
log(1 + dur_est):log(1 + mat_c_f) -9.89e+00   1.48e+00   -6.67  2.7e-10 ***
```

[17] When using categorical variables in the `lm` function, for all category levels the difference to the first level is regarded.

[18] The function `I(·)` that can be seen for some variables, is a fuinction of the `base` package denoting to use the input object “as is”.

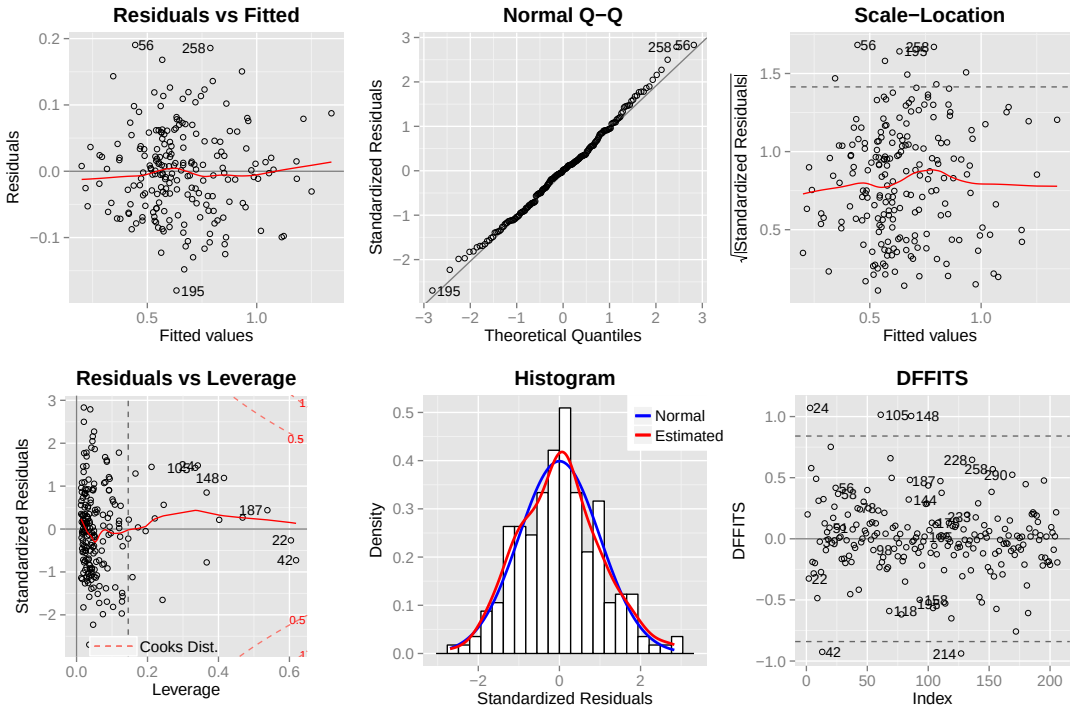
```
log(1 + dur_est):yield_f             -3.16e+00   6.81e-01   -4.64  6.6e-06 ***
log(1 + dur_est):I(bu == "BU4")TRUE -9.64e-02   2.12e-02   -4.54  9.9e-06 ***
log(1 + dur_est):log(1 + eng_h_f)    9.57e-02   3.10e-02    3.09   0.0023 **
log(1 + dur_est):log(1 + yield_f)    1.30e+02   2.77e+01    4.70  5.0e-06 ***
log(1 + dur_est):I(mat_c_f^2)        3.61e+01   6.34e+00    5.68  4.9e-08 ***
log(1 + dur_est):I(yield_f^2)        9.48e-03   2.08e-03    4.56  9.1e-06 ***
---
Residual standard error: 0.068 on 191 degrees of freedom
  (14 observations deleted due to missingness)
Multiple R-squared: 0.906,      Adjusted R-squared: 0.899
F-statistic:  132 on 14 and 191 DF,  p-value: <2e-16
```

Remark 2.21 The model name `lm_ct` stands for "linear model cost with transformed predictors".

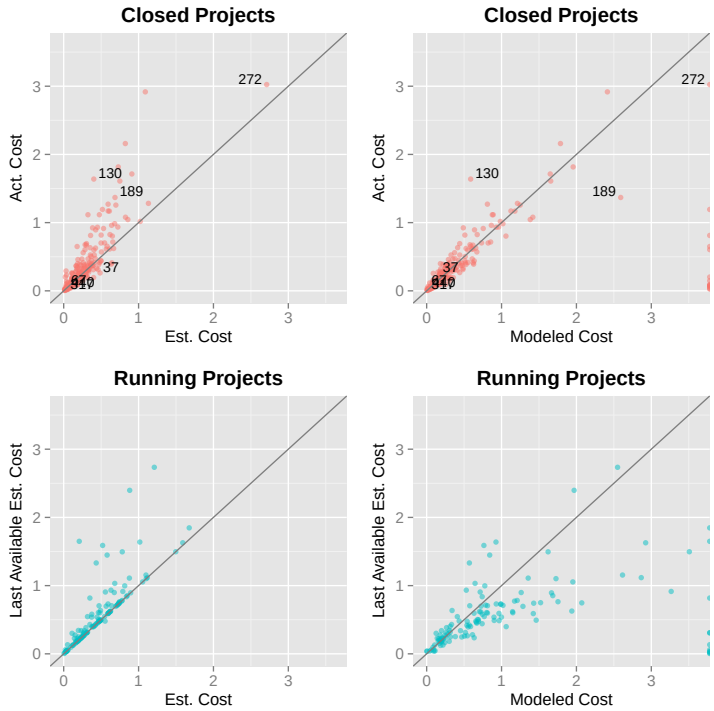
The residual plot of the final model `lm_ct` (see fig. 2.34a) shows no contradiction to constant variance. On the same time the adaption of the distribution to the normal distribution seems to be very good, as the Q–Q plot and the histogram show. This is confirmed by a Shapiro-Wilk test p-value of 0.6. Some projects still excess levels for influential points, but the excess is very small so that they are treated as inconspicuous. These observations show that there is no contradiction to the assumptions of the cost model `lm_ct` and thus can be used to predict project cost.

Apply to Running Projects: When applying the model `lm_ct` to running projects one large outlier was identified (not shown in plot). Project 321 resulted in a predicted cost of about 1356 million Euro (ME), whereas the last estimated cost are 1.8 ME. On the same time the maximum of actual cost of all closed projects is 3.02ME with a median of 0.24ME. Searching for reasons for this clear outlier it turned out that project 321 is an outlier for estimated duration (see fig. 2.8 in section 2.2, Exploratory Data Analysis (EDA)) and also for material cost, which are both important variables in model `lm_ct`. This reveals that a model, especially complicated models with many predictors, can be sensitive to outliers in the predictor variables, which has to be kept in mind in further steps of the analysis.

Comparing the estimated cost to the modeled cost in fig. 2.34b for the model clearly a much better adaption to the actual cost can be observed. Looking at the running projects, for estimated cost the last available cost estimation is larger for most of the running projects. In contrast the model tends to raise the last available cost estimation. Also here shall be stated that the issue of running projects for which the model predicts lower cost than the last available estimation, will be treated in the upcoming section 2.5, Model Improvements.



(a) Residual plot



(b) Model plot

Figure 2.34.: Final cost model `lm_ct` with transformed predictors (data: closed projects with all WS available)

2.4.3. Modeling Actual Project Duration based on First Workstatement

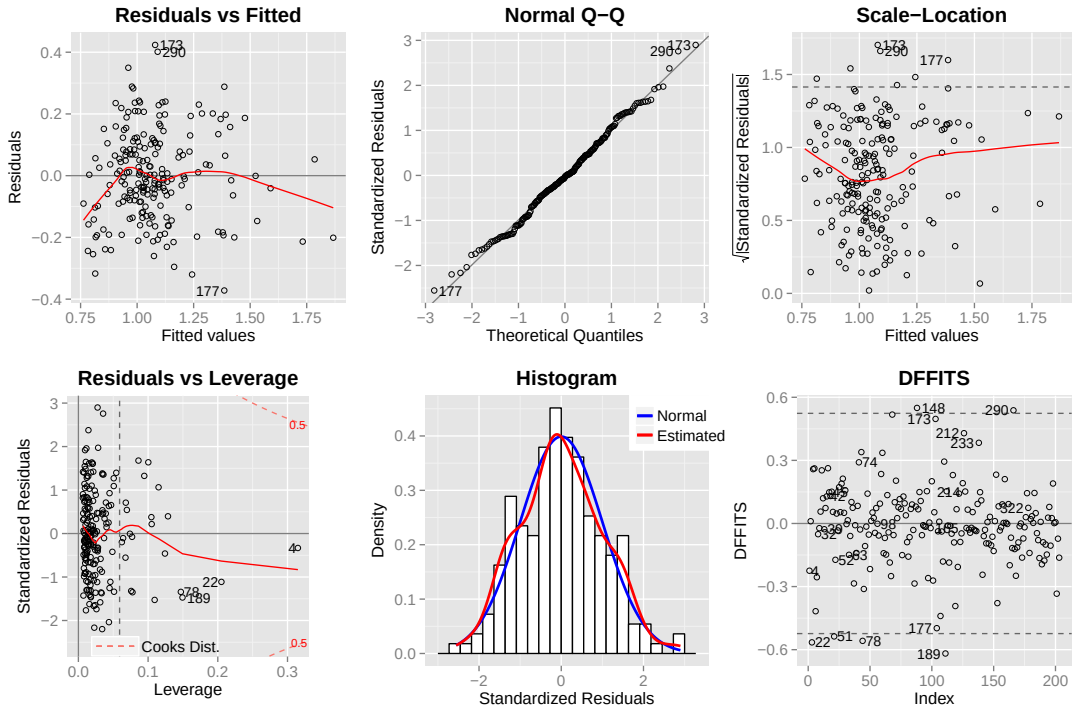
The decision process in modeling actual project duration based on the first WS is very similar as for modeling project cost. Thus this section contains just the results and some interpretation. For more information about the process of model selection see the practical section 2.4.1, Model Selection Process or the theoretical section 3.4, Model Selection.

Without Transformed Predictors

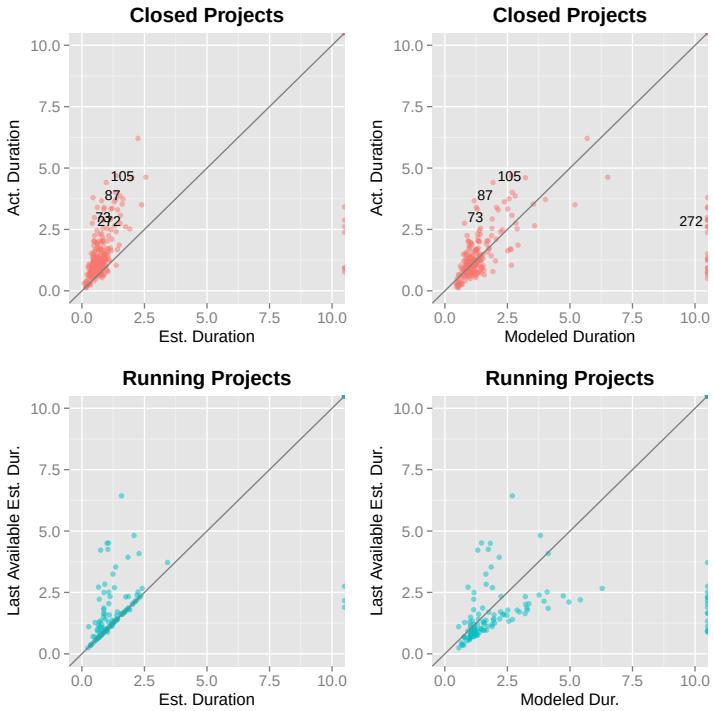
This final linear model for predicting actual duration contains only 5 basic predictors and no interactions.

```
> excl_d <- c(73, 87, 105, 272)
> lm_d <- lm(formula = (dur_act)^(1/3) ~ dur_est + eng_h_f + reuse_f +
    mat_c_f + chip_c_f, data = subset(cl_pr_all_WS, !(id %in% excl_d)))
> summary(lm_d)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3720 -0.0976 -0.0044  0.0982  0.4236
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.22e-01   5.25e-02   15.67  < 2e-16 ***
dur_est      4.25e-01   3.92e-02   10.86  < 2e-16 ***
eng_h_f     -3.17e-05   9.09e-06   -3.48  0.00061 ***
reuse_f     -1.22e-03   4.94e-04   -2.47  0.01430 *
mat_c_f      7.75e-01   2.73e-01    2.84  0.00505 **
chip_c_f     6.21e-02   1.91e-02    3.25  0.00138 **
---
Residual standard error: 0.148 on 197 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.6,       Adjusted R-squared: 0.59
F-statistic: 59.1 on 5 and 197 DF,  p-value: <2e-16
```

The residual plots (see fig. 2.35a) are inconspicuous and a Shapiro-Wilk test p-value of 0.55 confirms this for the normality assumption. Also here the plot of the fitted model values for closed and for running projects (see fig. 2.35b) shows that model predictions are unbiased.



(a) Residual plot



(b) Model plot

Figure 2.35.: Final duration model `lm_d` without transformed predictors (data: closed projects with all WS available)

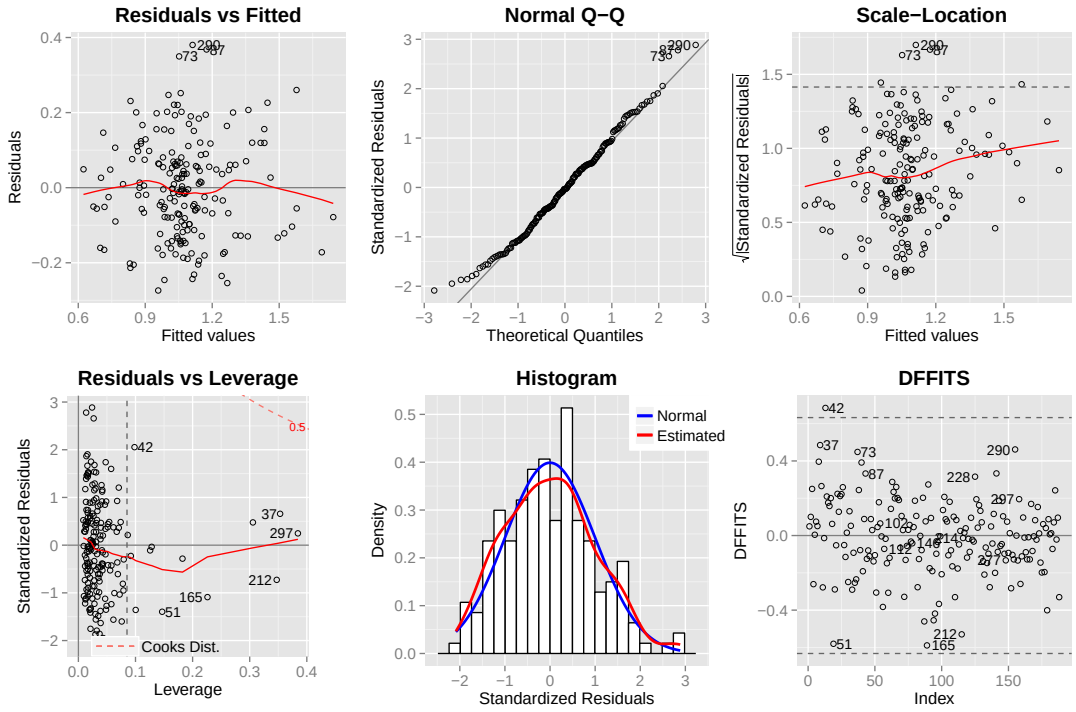
With Transformed Predictors

As for the project cost models, also for project duration the model with transformed predictors is more complicated than the model without. Thus this model with 7 predictors and several interactions is also more sensitive.

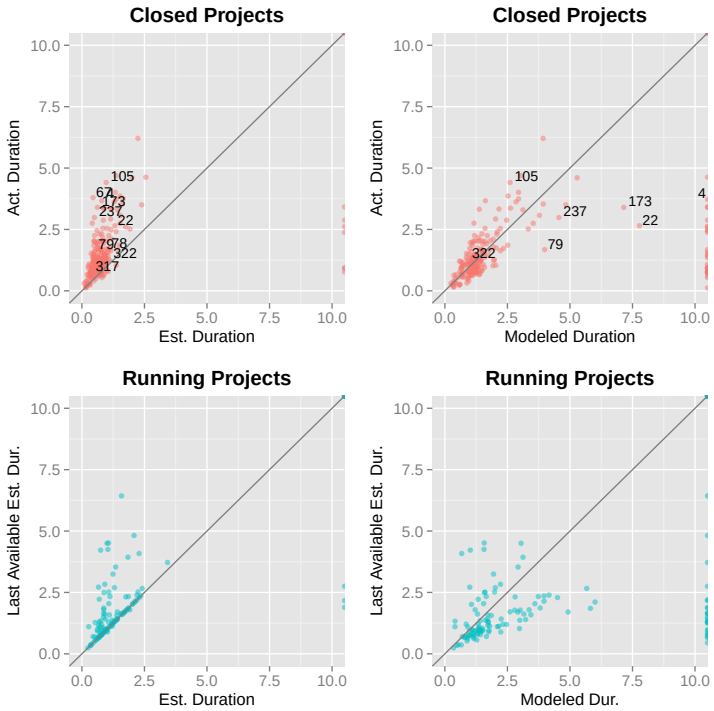
```
> excl_dt <- c(4, 22, 67, 78, 79, 105, 173, 237, 317, 322)
> lm_dt <- lm(formula = (dur_act)^(1/3) ~ ..., data = subset(cl_pr_all_WS,
    !(id %in% c(excl_dt))))
> summary(lm_t_df)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2738 -0.0971 -0.0043  0.0772  0.3802
Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                          0.70629    0.05339   13.23  < 2e-16 ***
die_size_f                           0.02855    0.00548    5.21  5.1e-07 ***
log(1 + dur_est)                     0.59762    0.07712    7.75  6.5e-13 ***
log(1 + mat_c_f)                     3.27510    0.73070    4.48  1.3e-05 ***
finance:log(1 + die_size_f)         -0.09134    0.01584   -5.77  3.4e-08 ***
log(1 + dur_est):log(1 + test_t_f)   0.16501    0.03711    4.45  1.5e-05 ***
log(1 + dur_est):I(nre_f^2)         -1.03043    0.30576   -3.37  0.00092 ***
log(1 + mat_c_f):log(1 + die_size_f) -1.31477    0.35630   -3.69  0.00030 ***
---
Residual standard error: 0.133 on 180 degrees of freedom
  (29 observations deleted due to missingness)
Multiple R-squared: 0.671,      Adjusted R-squared: 0.658
F-statistic: 52.5 on 7 and 180 DF,  p-value: <2e-16
```

Looking at the residual plots of the final model (see fig. 2.36a) the most remarkable topic is the residual distribution. The histogram shows a slight more widespread distribution for the estimated density compared to normal density. This can also be seen on the Q–Q plot as a deviation on the lower tails. On the upper tails 3 slight outliers can be seen that are not conspicuous when looking at the other plots. Also the Shapiro-Wilk test gives with a p-value of 0.12 no contradiction to the normal distribution assumption. Thus the model can be assumed to be valid.

The model outliers `excl_dt` can clearly be recognized on fig. 2.36b as outliers. Apart from this point the comparison of the model to actual and estimated values looks similar to the model without transformed predictors.



(a) Residual plot



(b) Model plot

Figure 2.36.: Final duration model `lm_dt` with transformed predictors (data: closed projects with all WS available)

2.4.4. Influential Variables

This section briefly examines the question of variables with highest influence on actual project cost and duration. The approach chosen is to perform a subset variable selection with forward search on all variables of interest (see section 3.4, Model Selection for details). This is done for variable values at project start (first WS) and project end (final WS) separately. The used data includes all closed projects with all WS available for analyzing project start and all closed projects for project end.

Actual Project Cost

- **most influential variables of *first WS*** (at project start)
  1. + engineering hours
  2. + estimated duration
  3. + material cost
  4. − business unit
  5. − nre + average chip sales price
  6. + purchased cost − reuse

- **most influential variables of *final WS*** (at project end)
  (excluding cost variables `eng_h`, `purch_c`, `mat_c` and `nre`)
  1. − reuse
  2. + pin count
  3. − business unit
  4. − average chip sales price
  5. − size of the technology

Actual Project Duration

- **most influential variables of *first WS*** (at project start)
  1. − reuse
  2. + material cost
  3. + die size − finance
  4. + test time + size of technology − business unit

- **most influential variables of *final WS*** (at project end)
  1. − reuse
  2. + engineering hours
  3. − finance
  4. + test time
  5. − purchased cost − pin count + die size

The above list gives for each response-WS combination the most influential variables. Additionally for each variable the sign of influence is given:

- "+ *var*" means that an increase of *var* increases the response (if all other variables are constant).
- "− *var*" says that an increase of *var* results in a decrease of the response (if all other variables are constant).

The variables are ordered based on the order they are added to the model by forward search. Forward search adds in each step a variable to the model that gives the most significant *additional* information.

For each response-WS combination at least two different data subsets are analyzed by subset regression. The data subsets differ on the number of projects due to missing values. Because of analyzing multiple models, the order of the variable importance is not always clear. Thus some variables are put together on an order level.

Remark 2.22

- **Order:** As stated above the order of the influential variables is not absolute. The intention is to give an idea about the significant information provided by each variable on describing actual cost and duration.
- **Database choice:** The same analysis could be done including running projects, as model fits give an unbiased prediction for actual project cost and duration. Here it was decided to use closed projects only, as the relationship of cost and duration of running projects to the other variables is the same. This is due to the relationship given by the models.
- **Connection to models:** This analysis corresponds closely to the analyzed models summarized in the upcoming section 2.4.5, which can be seen by the similarity of predictor variables to influential variables stated in this section.
- **Excluded variables:** The variables of estimated cost/duration and actual cost/duration are excluded from influence analysis, as they have logically high influence on actual cost/duration. Only estimated duration is admitted for analyzing influence on actual cost.
  On analyzing influence of variables at project end for actual cost the listed cost variables are excluded, as they are the main factors of a linear combination that exactly gives the actual cost. To get an idea of the influence of these cost variables on actual cost, please be referred to the corrlation matrix in fig. 2.37.
- **Signs of influence for categorical variables:** Here numerical versions of categorical variables `bu` and `finance` with reduced levels are used (see section 2.2.2, Reducing Category Levels). This gives the following relationship:
  - BU1 or BU2 $\Rightarrow$ `bu` $= 1$
    BU3 or BU4 $\Rightarrow$ `bu` $= 2$

– F1 ⇒ `finance` = 1
  F2 ⇒ `finance` = 2

For example take the negative influence of variable business unit on actual duration. This means that the actual duration of a project p_1 of business unit BU1 or BU2 (`bu` = 1) is higher than the actual duration of a project p_2 of business unit BU3 or BU4 (`bu` = 2). This is only a valid conclusion under the assumption that projects p_1 and p_2 have the same values for all other variables.

- **Influence of average chip sales price on cost:** It is notable that the average chip sales price shows a positive influence on cost regarding the first WS and a negative for the final WS. This may yield due to excluding cost variables. A model not containing for example engineering hours or material cost changes the way how additional influence is described.

  This fact has to be considered on interpreting the influential variables of final WS on actual cost.

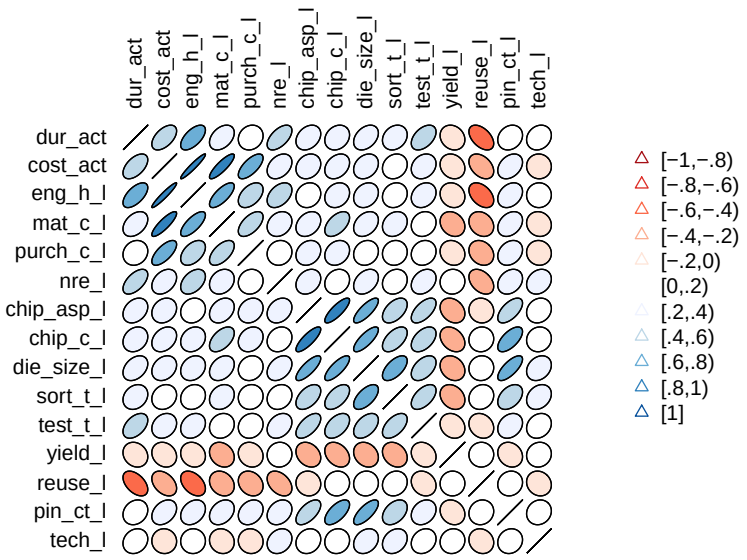


Figure 2.37.: Pairwise correlation matrix ellipses for variables of the last WS using Pearson correlation (data: closed projects with all WS available)

This analysis also corresponds closely to the analyzed correlations in section 2.2.3, Correlations. Variables highly correlated with actual project cost or duration are listed as influential variables above. High correlations can also be a reason for variables not listed here. For example fig. 2.12 on page 24 gives for `eng_h_f` and `dur_est` a high correlation between 0.6 and 0.8, but engineering hours is not listed among the most influential variables of the first WS. On the same time fig. 2.12

reveals that `eng_h_f` is significantly correlated to `reuse_f` and `mat_c_f`, which are both listed as influential variables. Because of that the engineering hours seem to give no significant additional information in describing actual duration. As the correlation matrix helps to interpret the most influential variables, fig. 2.37 gives a graphical view of the correlation matrix for variables of the last WS.

It concludes that variables not listed above do not necessarily have to be non-influential on actual cost or duration.

2.4.5. Summary

This section summarizes the results about the final models of the previous two sections. First each model pair is compared to get an idea about their differences. At the end of this section the results are also interpreted.

Remark 2.23 To reference to the fitted values of the previously identified models, following terms shall be used:

- **`cost_lm`:** fitted values of `lm_c`
  (cost model without transformed predictors)
- **`cost_lm_t`:** fitted values of `lm_ct`
  (cost model with transformed predictors)
- **`dur_lm`:** fitted values of `lm_d`
  (duration model without transformed predictors)
- **`dur_lm_t`:** fitted values of `lm_dt`
  (duration model with transformed predictors)

Comparing Model Pairs

Fig. 2.38 directly compares the predicted values of the models for project duration and cost. The graphics show scatterplots of the difference of model predictions with transformed predictors to the model predictions without transformed predictors against the actual values. Regarding duration for short lasting projects `lm_d` predicts higher duration than `lm_dt`. For longer lasting projects the situation changes (The curve down at `dur_act` from 4 to 6 is a result of project 42). The respective plot of project cost shows that the two models are very similar across the actual cost range regarding their mean. Also here the curve down is a result of an outlier (project 52).

Table 2.7 gives basic characteristics on model differences. For both the distribution looks symmetric with center at zero. For project duration 50% of the differences lie in the range of $[-0.14, 0.21]$ years, which is equal to $[-51, 77]$ days. The respective

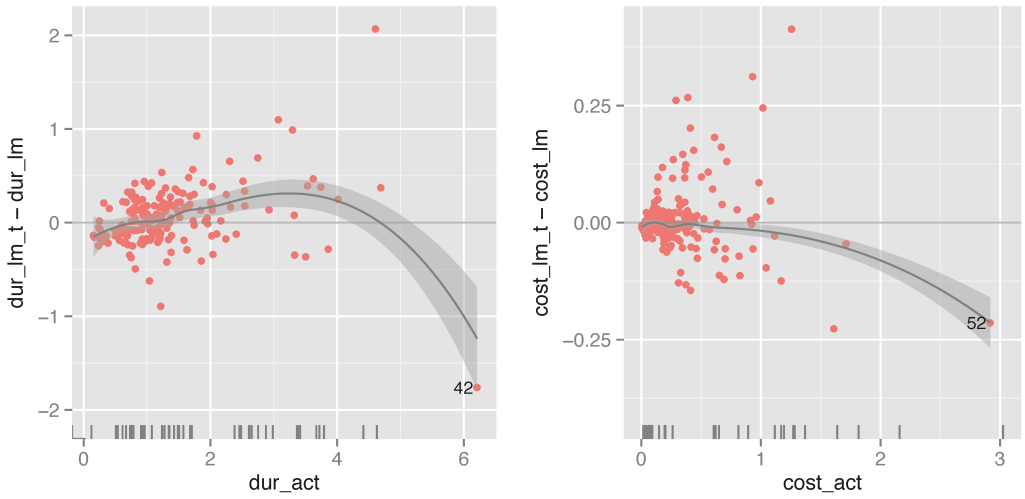


Figure 2.38.: Comparison of model predictions with and without transformed predictors with LOESS smooth. Two outliers regarding actual values are marked (data: closed projects with all WS available)

Table 2.7.: Basic characteristic numbers of model prediction differences (data: closed projects with all WS available)

	Min	$q_{0.25}$	**Median**	**Mean**	$q_{0.75}$	**Max**	**#NA**
`dur_lm_t − dur_lm`	-1.76	-0.14	0.05	0.06	0.21	2.07	49.00
`cost_lm_t − cost_lm`	-0.23	-0.02	0.00	0.01	0.02	0.41	30.00

50% range of project cost is $[-0.02, 0.02]$ ME. The outcome of this is that the model predictions can differ within certain ranges, but on average the difference is about zero.

Model Summaries

The following list summarizes the model formulas and lists closed projects excluded from the analysis, as they were identified as outliers:

- **lm_d** (duration model without transformed predictors)

$$\begin{aligned}\hat{\mathrm{m}}(\texttt{dur\_act}) \approx\ & [0.822 + 0.425\ \texttt{dur\_est} \\ & + 3.17 \cdot 10^{-5}\ \texttt{eng\_h\_f} \\ & - 1.22 \cdot 10^{-3}\ \texttt{reuse\_f} \\ & + 0.775\ \texttt{mat\_c\_f} \\ & + 0.0621\ \texttt{chip\_c\_f}]^3\end{aligned}$$

  closed projects excluded from the model:
  73, 87, 105, 272

- **lm_dt** (duration model with transformed predictors)

$$\begin{aligned}\hat{\mathrm{m}}(\texttt{dur\_act}) \approx\ & [0.70629 + 0.02855\ \texttt{die\_size\_f} \\ & + 0.59762\ \texttt{log(1+dur\_est)} \\ & + 3.27510\ \texttt{log(1+mat\_c\_f)} \\ & - 0.09134\ \texttt{finance} \cdot \texttt{log(1+die\_size\_f)} \\ & + 0.16501\ \texttt{log(1+dur\_est)} \cdot \texttt{log(1+test\_t\_f)} \\ & - 1.03043\ \texttt{log(1+dur\_est)} \cdot \texttt{(nre\_f)}^2 \\ & - 1.31477\ \texttt{log(1+mat\_c\_f)} \cdot \texttt{log(1+die\_size\_f)}]^3\end{aligned}$$

  closed projects excluded from the model:
  4, 22, 67, 78, 79, 105, 173, 237, 317, 322

- **lm_c** (cost model without transformed predictors)

$$\begin{aligned}\hat{\mathrm{m}}(\texttt{cost\_act}) \approx\ & [0.3125 + 0.2690\ \texttt{dur\_est} \\ & + 7.596 \cdot 10^{-5}\ \texttt{eng\_h\_f} \\ & + 2.677\ \texttt{mat\_c\_f} \\ & - 0.05087\ \texttt{buBU2} \\ & - 0.1595\ \texttt{buBU3} \\ & - 0.1006\ \texttt{buBU4} \\ & - 2.202\ \texttt{dur\_est} \cdot \texttt{mat\_c\_f}\ ]^3\end{aligned}$$

  closed projects excluded from the model:
  26, 37, 39, 42, 105, 130, 146, 177, 187, 228, 272, 290

- **lm_ct** (cost model with transformed predictors)

$$
\begin{aligned}
\hat{\mathrm{m}}(\texttt{cost\_act}) \approx\ & [0.1446 + 5.241 \cdot 10^{-5}\ \texttt{eng\_h\_f} \\
& - 50.77\ \texttt{mat\_c\_f} \\
& + 0.4706\ \texttt{purch\_c\_f} \\
& - 0.1019\ \texttt{chip\_c\_f} \\
& + 5.955 \cdot 10^{-3}\ \texttt{die\_size\_f} \\
& - 380.0\ \texttt{log(1+dur\_est)} \\
& + 57.77\ \texttt{log(1+mat\_c\_f)} \\
& - 9.892\ \texttt{log(1+dur\_est)} \cdot \texttt{log(1+mat\_c\_f)} \\
& - 3.159\ \texttt{log(1+dur\_est)} \cdot \texttt{yield\_f} \\
& - 0.09644\ \texttt{log(1+dur\_est)} \cdot \texttt{buBU4} \\
& + 0.09570\ \texttt{log(1+dur\_est)} \cdot \texttt{log(1+eng\_h\_f)} \\
& + 130.2\ \texttt{log(1+dur\_est)} \cdot \texttt{log(1+yield\_f)} \\
& + 36.05\ \texttt{log(1+dur\_est)} \cdot \texttt{(mat\_c\_f)}^2 \\
& + 9.480 \cdot 10^{-3}\ \texttt{log(1+dur\_est)} \cdot \texttt{(yield\_f)}^2\ ]^3
\end{aligned}
$$

  closed projects excluded from the model:
  37, 67, 130, 189, 272, 317, 440

Remark 2.24

- The function $\hat{\mathrm{m}}(\cdot)$ denotes the estimated median. As in section 3.3, Model Diagnostics stated, back-transformation of the fitted values to the original scale results in the median.

- Model `lm_dt` contains the factor `finance`, which has the levels `F1` and `F2`. When calculating the formula the value 1 has to be inserted for `finance`, if the corresponding project has financial project type `F1` and the value 2 has to be inserted for financial project type `F2`. This is the encoding as it is done by **R**.

- `lm_c` and `lm_ct` partly hold factors `buBU2`, `buBU3` and `buBU4`. These variables are *dummy variables* that take values of 0 or 1. The variable takes value 1 if the underlying project is of the corresponding business unit and value 0 otherwise.

  For example if an estimation of the median for a new project of business unit BU3 is wanted to be calculated, than the formula of the `lm_c` model is used with `BU2` $= 0$, `BU3` $= 1$ and `BU4` $= 0$.

- ATTENTION: The above used function `log(·)` denotes the natural logarithm and NOT the logarithm with base 10, as sometimes denoted.

Table 2.8.: Summary of model characteristics

	AIC	**AIC**$_c$	**BIC**	$\boldsymbol{R^2_{\text{adj}}}$	$\boldsymbol{R^2}$	$\boldsymbol{s}$
`lm_d`	-769.25	-768.82	-749.37	0.59	0.60	0.15
`lm_dt`	-749.63	-748.83	-723.74	0.66	0.67	0.13
`lm_c`	-1058.76	-1058.03	-1032.13	0.86	0.87	0.08
`lm_ct`	-1093.42	-1090.89	-1043.50	0.90	0.91	0.07

Table 2.8 summarizes some characteristics for each model. For explanations about the characteristics please see chapter 3, Theoretical Fundamentals and especially section 3.4, Model Selection.

Table 2.9.: Summary of characteristics for the simple models

	AIC	**AIC**$_c$	**BIC**	$\boldsymbol{R^2_{\text{adj}}}$	$\boldsymbol{R^2}$	$\boldsymbol{s}$
`lm_d_simple`	-774.88	-774.83	-768.19	0.52	0.52	0.16
`lm_c_simple`	-992.62	-992.57	-985.89	0.77	0.77	0.10

Additionally in table 2.9 the same characteristics are given for two simple models. `lm_d_simple` uses only `dur_est` to predict $(\texttt{dur\_est})^{1/3}$ (analogously for `lm_c_simple`). Thus the scale the values of table 2.9 can be used to directly compare to the values of 2.8.

Remark 2.25

- The s values in tables 2.8 and 2.9 are on the transformed scale of cost respective duration to the power of 1/3. This is important to consider on interpreting these values.
- On comparing the model selection criteria it should also be considered that the underlying datasets differ slightly due to outlier elimination.

Interpretation

Also here the R^2 and R^2_{adj} values show that duration is more difficult to predict than cost. In other words the portion of variance described by the model is larger for cost than for duration.

Looking at each model pair in table 2.8 it is observable that the adjusted coefficient of determination (see section 3.4, Model Selection) R^2_{adj} increases 7% resp. 4% for the duration resp. cost model, when comparing model without and with transformed predictors. So the more complicated models have a higher portion of described variance or information. Regarding the cost models also the other model

selection criteria AIC, AIC_c and BIC prefer the model with transformed predictors. In contrast for duration the model selection criteria prefer the model without transformed predictors.

By looking at the model selection criteria of the simple models in table 2.9 it is obvious that the models using more predictors are favored by every model selection criterion. The improvements in R^2_{adj}, which is associated to the portion of described variance, range from 7% to 14%.

2.5. Model Improvements

The models discussed in the previous section already describe a noticeable part of response variance. On the same time it was observed that the models predict lower response values compared to the last available estimations for some running projects. This is of course possible, but still suspicious. For this reason this section has a closer look on duration and cost WS estimations starting from the second WS. This information was not used so far for modeling.

First an introductory section briefly describes the approaches for improving the models already found. The consecutive two sections focus on analyzing how to improve the models for actual project cost and duration by using the information of estimations from all available WS. Finally the possibly found improvements are evaluated.

2.5.1. Improvement Ideas and Approaches

There were mainly two ideas on improving the models found:

1. Use the information of **estimated project end** to improve model predictions and to included direct information about time.
2. Use the information of **cost and duration estimations of all available WS** to improve the model predictions.

Remark to idea 1: This idea is based on two observations:

(i) As shown in section 2.3, Modeling Approach using the information about project start in modeling actual project cost/duration results in a bias. This bias is due to the structure of the data: Recent started and already finished projects tend to cost less and last shorter than projects started longer ago. The idea is that this problem may be avoided by using the information of project end.

(ii) In section 2.1.4, Assumptions the assumption of *comparability of projects over time* is described. When using project end as predictor variable for actual project cost/duration, this assumption is not needed any more.

First it was checked if the models of the previous section are valid over the time range. By conducting the analysis over various bisections of the data by the time range the validity could be verified. Hence no contradiction to the assumption stated in (ii) is given by using the original models.

Nevertheless (i) suggests a model improvement by using the information of project start. Here it is important to observe that using the information of estimated project end in conjunction with estimated project duration makes the information of project start available (project end − project duration = project start). This leads to an undesirable bias as already stated. This effect was verified by analyzing models that include the estimated project end as predictor.

To sum up, the realization of idea 1 would cause a bias and is not needed to fulfill the assumptions.

Remark to idea 2: It was already mentioned that using the information of all WS is associated with difficulties (see section 2.1.3 about the database structure). The main issue is that there is no comparable point of time for WS creation. Thus the information of the last available WS (or any other WS than the first) has no comparable time horizon.

On the same time the assumption that WS estimations approximate the actual value is obvious. This assumption shall be verified and analyzed on possible methods of using this information to improve the predictions for actual project cost/duration. To look at the different WS in a structured way it is chosen to compare WS of the same number. This means that all first WS are compared, all second WS are compared and so on. The first approach chosen is exploratory and described in the next two sections.

Remark 2.26 The approach of comparing WS of the same number does not result in comparing last available WS. The concept of last available estimations is only applicable to running projects, but for running projects actual duration and cost are not known. Still this chosen approach makes sense, as a strategy for improving model fits can be generated. The next step is to simulate closed projects as being running projects and analyze the improved predictions, as it will be done in section 2.5.4, Improvements Evaluation.

2.5.2. Duration Model Improvements

The dataset used here is based on the closed projects with all WS available. It is extended as for all available WS the estimations of cost and duration are considered.

Remark 2.27 On analyzing how the last available estimation can aid on improving the model predictions, only WS excluding the final WS can be regarded. The reason is that the final WS already contains the actual duration, which is assumed to be not known. This assumption is especially true for running projects. It follows that for analyzing the j-th WS only closed projects come into account, that were still running at the j-th WS.

To analyze the behavior of duration estimations of each WS, these estimations are compared to the actual duration. This means that for each WS the difference of estimated duration to the actual duration is regarded. As an example here the analysis of all WS up to the third WS is presented.

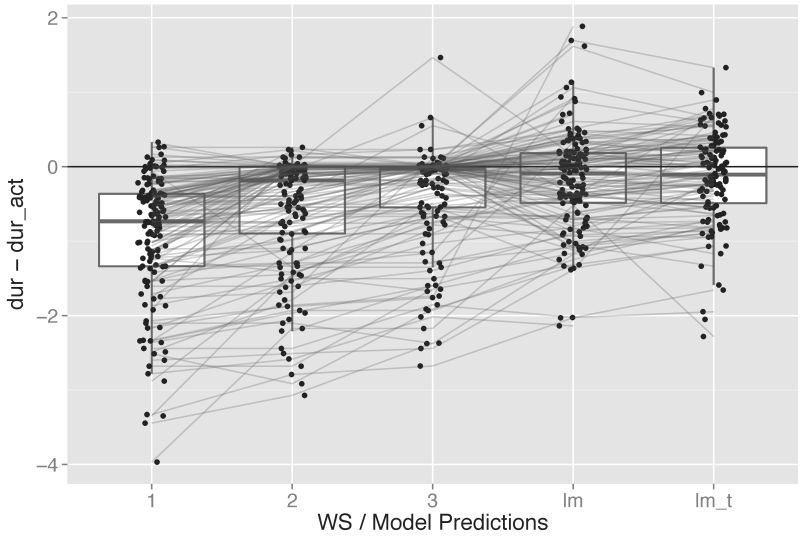


Figure 2.39.: Boxplot series of duration deviation. The first three boxplots represent the difference of the estimated duration at that WS to the actual duration. The last two boxplots represent the difference of the respective model prediction to the actual duration (lm_t and lm correspond to `lm_dt` and `lm_d` respectively). Each grey line connects a single project (data: closed projects with all WS available, that were still running at the 3rd WS)

Interpretation: Fig. 2.39 shows for WS 1, 2, and 3 the differences of the corresponding duration estimations to the actual duration as boxplots. The boxplots are overlaid by the jittered[19] observations. These boxplots are compared to the boxplots of the difference of the two duration model estimations of `lm_d` and `lm_dt` to the actual duration. To get an idea of how project estimations behave, for each project the corresponding points are connected by straight lines. Thus points on

[19]This is a method used by the `ggplot2` package to avoid overplotting. The points are *jittered* around their real values.

the zero line represent projects for which the actual duration was estimated exactly. Points above and underneath the zero line represent overestimations and underestimations, respectively. Fig. 2.39 makes clear that WS estimations tend to underestimate. On the same time estimations get better for increasing WS and the median seems to approximate zero. The models are relatively symmetric around zero, although the median is slightly negative. The aim is to use the last available WS estimation to decrease the model variations, while keeping the median at zero. A zero median or mean is necessary to have an unbiased prediction.

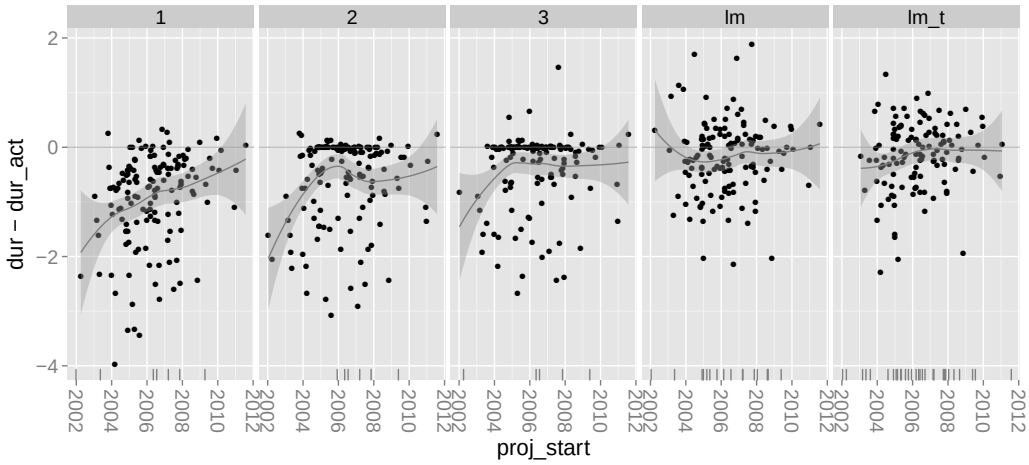


Figure 2.40.: Scatterplot series of duration deviation and LOESS smooth with 95% CI. (data: closed projects with all WS available, that were still running at the 3rd WS)

The impact of the skewness of WS estimations deviations can be seen more clearly on the corresponding scatterplot of the differences against project start, as it is shown in fig. 2.40. An accumulation of points around zero is forming for increasing WS. On the other hand the smoothing lines show that the actual duration is still systematically underestimated by WS estimations, in contrast to the model fits.

Transformation: Having a closer look on the connection of the third WS and the model `lm_d` in fig. 2.39 yields the idea that it is desirable to use as transformation the maximum of WS estimation and model prediction[20]: The idea is that the maximum improves the predictions/estimations for underestimated values and keeps some overestimations that ensure unbiasedness.

The result is shown in fig. 2.41, that plots the difference of maximum of third WS estimation and `lm_d` model fit to actual duration[21]. Here only third WS and model `lm_d` are shown. Obviously the maximum for the third WS is centered at

[20] It is equivalent to take the maximum of the difference to actual values and the maximum of WS estimations and model predictions themselves, as $\min(x-c, y-c) = \min(x, y) - c \; \forall x, y, c \in \mathbb{R}$.

[21] For both plots of fig. 2.41 the right version representing `lm_d` is the same as in fig. 2.39 and fig. 2.40, because the maximum of model prediction with itself is regarded.

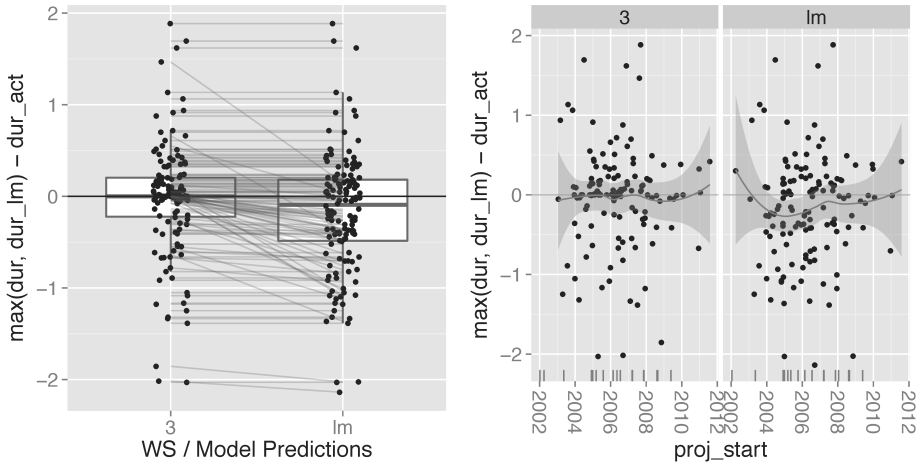


Figure 2.41.: Boxplot and scatterplot series of improved duration deviation with LOESS smooth and 95% CI for the scatterplots. The deviation is improved for the 3rd WS as the maximum of the 3rd WS estimation and the model prediction of `lm_c` is considered (data: closed projects with all WS available, that were still running at the 3rd WS)

zero and looks symmetric with a smaller IQR than the model predictions. Also the scatterplot shows for the third WS maximum an unbiased estimation across time.

Remark 2.28

- This idea of a transformation to improve model predictions will be evaluated in section 2.5.4, Improvements Evaluation.
- This section showed the analysis with focus on the third WS. In practice all WS were analyzed and validated.

2.5.3. Cost Model Improvements

Interpretation: Looking at the same situation for project cost the boxplot series in fig. 2.42 shows some differences to project duration. The distribution of project cost estimations has larger variance compared to the IQR. Also the zero line is within the IQR already for the first WS.

Transformation: The first idea might be to use the maximum of last available estimation and model predictions, as done for duration. This approach led to a

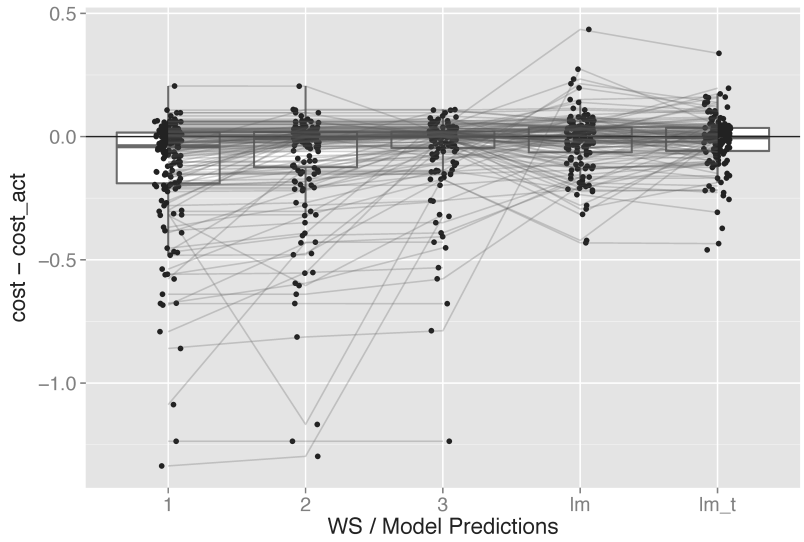


Figure 2.42.: Boxplot Series of cost deviation for WS 1 to 3 and both models (data: closed projects with all WS available, that were still running at the 3rd WS)

skewed overestimation, thus a bias. The transformation chosen instead is

$$\texttt{cost\_transf}(x, y) \coloneqq \begin{cases} (x+y)/2 & \text{if } x > y \\ y & \text{if } x \leq y \end{cases}$$

$x =$ estimation of the corresponding WS

$y =$ model prediction

This transformation is chosen based on the following steps:

- The cost models give already very good approximations of actual cost, thus the maximum caused an overestimation.
- **Idea:** Use average $(x + y)/2$.
  Result: The result gives a distribution which median fits the actual cost very good, but is still skewed caused by a set of clear underestimations of the regarded WS estimations.
- **Idea:** Use average only if WS estimation is larger than the model fit and the model fit otherwise.
  Result: Median still fits the actual cost very well and additionally the distribution looks symmetric.

The result can be seen in fig. 2.43 that applies the cost transformation function `cost_transf` to the third WS and compares it to the model fits of `lm_c`. Obviously the distribution becomes symmetric and variance decreases by applying the transformation. The transformation is also unbiased over time as the right plot shows.

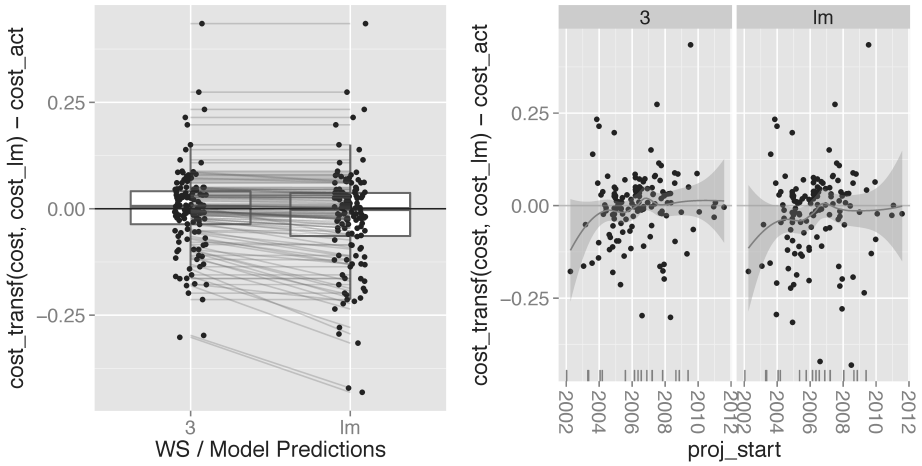


Figure 2.43.: Boxplot and scatterplot series of improved cost deviation. The scatterplots also show a LOESS smooth and 95% CI (data: closed projects with all WS available, that were still running at the 3rd WS).

2.5.4. Improvements Evaluation

The idea of how to evaluate the transformations of duration and cost is to apply the transformations to simulated running projects. This simulation can be done by choosing a date d in the past and replicating the project states at date d. An appropriate date d has to be far enough in the past, so that enough today closed projects were running. On the same time d has to be close enough to today, such that enough today closed projects were already running. Here d = 2007-01-01 is chosen with 123 projects running at date d, remaining of 227 today closed projects that have all WS available. This dataset of simulated running projects will be referred to as `data_eval`.

For each of the four models (2 models for actual cost and actual duration each) two types of evaluation models are generated, as listed in table 2.10.

Table 2.10.: Evaluation model types

	Response	**Predictor**
Type 1	actual duration/cost	original model fits
Type 2	actual duration/cost	transformed model fits

Remark 2.29

- The same analysis as shown underneath can also be performed for another date d. The analysis of other dates d gave similar results, so d = 2007-01-01 is shown as an example.

- The models of section 2.4, Modeling Cost and Duration were evaluated on a transformed scale using the response transformation $f(y) = y^{1/3}$. Thus the model assumptions are also analyzed on this transformed scale. To account for the assumptions, the evaluation models are also analyzed on the transformed scale. This implies that also the predictor has to be transformed by $f(\cdot)$.
- The improvement transformations were done on the original scale of project cost and duration. The reason is that this approach makes the improvements easier to interpret.
- To refer to the improvement transformations the extension `_mod` will be used.

The following aspects are of main interest on evaluating the model improvements:

Information described by the model: The degree of information can be measured by the coefficient of determination R^2 or the estimated residual standard error s.

Model assumptions: To draw conclusion from the model fits it is necessary to verify the model assumptions first. In place of the whole model diagnostics here the p-value of the Shapiro-Wilk test will be presented only. The other assumptions were verified during the analysis process.

Unbiasedness of predictions: The analyzed models are simple linear regression models and thus have the following form:

$$\mathrm{E}(\boldsymbol{Y}^{1/3}) = \beta_0 + \beta_1 \boldsymbol{x}^{1/3}$$

(see section 3.1, Simple Linear Regression) with $\boldsymbol{x}^{1/3}$ being an original or improved predictor (on the transformed scale). Thus

$$\mathrm{median}(\boldsymbol{Y}) = \left(\beta_0 + \beta_1 \boldsymbol{x}^{1/3}\right)^3$$

is the median of $\boldsymbol{Y}$. Here it shall be verified that $\boldsymbol{x}$ can be used as an unbiased predictor itself, which is equivalent to the hypothesis

$$\beta_0 = 0 \text{ and } \beta_1 = 1.$$

Duration Models

Evaluating the duration model (see table 2.11) shows that the improved model predictions describe for the test database an about 4% higher portion of information than the original predictions. Accordingly the residual standard error s is decreaed by the improvement transformation. On the same time the p-values

Table 2.11.: Evaluation of duration model improvements (data: `data_eval`)

Model $\mathrm{E}(\texttt{dur\_act}^{1/3}) =$	R^2	s	p-**value** $\beta_0 = 0$	p-**value** $\beta_1 = 1$	p-**value Shapiro-Wilk**
$\beta_0 + \beta_1$ `dur_lm`$^{1/3}$	0.60	0.155	0.37	0.53	0.77
$\beta_0 + \beta_1$ `dur_lm_mod`$^{1/3}$	0.64	0.147	0.72	0.75	0.79
$\beta_0 + \beta_1$ `dur_lm_t`$^{1/3}$	0.66	0.146	0.68	0.83	0.22
$\beta_0 + \beta_1$ `dur_lm_t_mod`$^{1/3}$	0.71	0.134	0.79	0.81	0.13

give no contradiction to model assumptions and unbiased predictions (all p-values $\gg 0.05$).

Decision: Concluding the improved duration predictions in fact improve the predictions and can be used as unbiased predictions for actual project duration.

Remark 2.30 ATTENTION: The values of R^2 and s can not be compared directly between the models with and without transformed predictors, as the number of observations differs due to different number of missing values and outliers. At the same time comparison gives ideas of the quality.

Cost Models

Table 2.12.: Evaluation of cost model improvements (data: `data_eval`)

Model $\mathrm{E}(\texttt{cost\_act}^{1/3}) =$	R^2	s	p-**value** $\beta_0 = 0$	p-**value** $\beta_1 = 1$	p-**value Shapiro-Wilk**
$\beta_0 + \beta_1$ `cost_lm`$^{1/3}$	0.87	0.075	0.46	0.22	0.03
$\beta_0 + \beta_1$ `cost_lm_mod`$^{1/3}$	0.92	0.061	0.02	0.02	0.82
$\beta_0 + \beta_1$ `cost_lm_t`$^{1/3}$	0.92	0.064	0.67	0.36	0.92
$\beta_0 + \beta_1$ `cost_lm_t_mod`$^{1/3}$	0.94	0.057	0.07	0.07	0.69

For project cost the improved predictions also give higher degrees of determination, whereas the increase is about 5% for `cost_lm` and about 2% for `cost_lm_t` (see table 2.12). In contrast to the original predictions, the improved ones have potential risk of being biased predictions as the p-values of the corresponding hypothesis tests are with 0.02 and 0.07 relatively low.

Decision: The decision is to use the improvement transformation only for the simpler cost model `lm_c` without transformations. As already analyzed project cost is more difficult to predict accurately, than project duration. Model `cost_lm_mod` has a large Shapiro-Wilk test p-value and a very high coefficient of determination of 0.92. On the same time here a test dataset is regarded, whereas the relatively

low p-values for testing on β_0 and β_1 are not considered as contradiction to a good model.
For the model `lm_ct` with transformations the original predictions are used, as they are unbiased for the test dataset `data_eval` and the improvement transformation gives for the test dataset only 2% higher portion of described variance.

Remark 2.31 In the upcoming sections it will not be differentiated explicitly between original and improved model predictions. For all but the predictions of `lm_ct`, the improved predictions are meant by referring to model predictions.

2.6. Trend Analysis

The intention of this section is to analyze trends of project cost and duration by time. The underlying question is if these factors increased or decreased during the last years. To give answers it is crucial to have predictions for actual values of running projects that represent reality as good as possible.

Up to here the models for unbiased predictions of actual values for running projects (based on closed projects) are derived and analyzed in detail. Hence the models give reliable and unbiased estimations of actual project cost and duration. By applying the models to running projects the trend analysis can be performed.

First section 2.6.1, Dataset and Analysis Mode introduces the data and gives details on the analysis mode. The results are presented in section 2.6.2, Results and are illustrated by examples in section 2.6.3, Example. The last two sections also give interpretations on some results. For an overall interpretation please see chapter 4, Conclusions.

2.6.1. Dataset and Analysis Mode

Dataset

As indicated in section 2.3, Modeling Approach projects started before the date *data collection start* (= 2006-10-25) tend to last longer, because of the dataset structure (see section 2.1.3, Structure). Hence the trend analysis will be based on all projects that started after the *data collection start* date. This dataset shall be denoted as `data_trend` and covers about 6 years of time (from *data collection start* = 2006-10-25 to *today* = 2012-10-29).

`data_trend` consists of

- 116 closed projects and
- 128 running projects

$\Big\} = 244$ projects

including 3 closed projects with missing first WS.

On analyzing trend, project cost of different projects are compared over the years. To make project costs comparable an inflation adjustment is necessary. Here as reference year to adjust the cost for 2012 is chosen. The used inflation rate i is an average of inflation rates from 2000 to 2017[22] of 35 advanced economies worldwide (International Monetary Fund [11]).

Let $s :=$ project start year, $d :=$ (predicted) project duration and $c :=$ project cost of a single project. The inflation adjusted cost c_{adj} is calculated as

$$\begin{aligned} c_{\text{adj}} &= \sum_{j=0}^{d-1} \frac{c}{d} \, (1+i)^{2012-(s+j)} \\ &\overset{i \geq 0}{=} c \, \frac{(1+i)^{2012-s}}{d} \, \frac{(\frac{1}{1+i})^d - 1}{(\frac{1}{1+i}) - 1} \end{aligned}$$

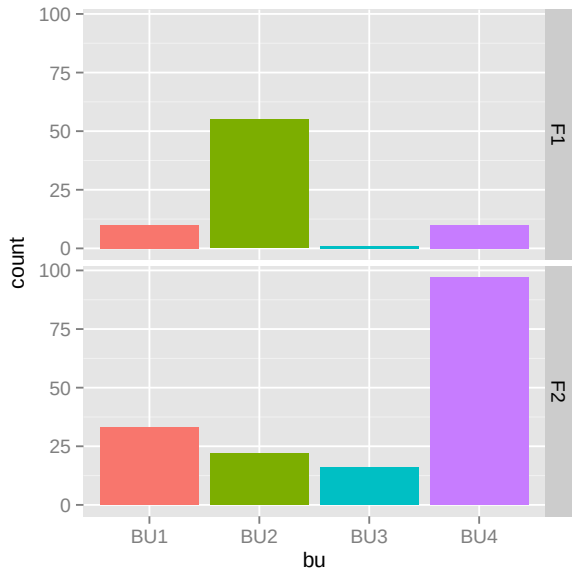
and is based on the idea to equally distributed project cost over all years while running.

The trend analysis will also analyze trends for different data categories based on business units and financial project types. Hence the distribution of projects across these categories is of interest, as shown in fig. 2.44. Especially BU3 and all but two combinations of `bu` and `finance` (F1 & BU2 and F2 & BU4) have few projects (at most 33). For that reason it might be difficult to identify trends for these categories. Especially slight trends might not be significant.

Remark 2.32

- Also here missing values are important. The distribution of missing values is already discussed in section 2.2.4, Missing Values. Here also model outliers play an important role. Approximates of number of missing values will be supplied with the results.
- For the rest of this section all cost variables shall refer to inflation adjusted costs.
- The 3 closed projects with missing first WS can not be used for analyzing estimation errors, as estimations at project start are needed.

[22] 2000 is the year of the first project start. 2017 is the predicted year of the last project closure. Inflation rates from 2012 to 2017 are estimations.



	BU1	BU2	BU3	BU4	Sum
F1	10	55	1	10	76
F2	33	22	16	97	168
Sum	43	77	17	107	244

Figure 2.44.: Histogram of business units for both financial project types (data: `data_trend`)

Analysis Mode

As the trend analysis is based on regression models that include some uncertainty, value is set to diversify the analysis. Diversity shall be given by regarding different response variables. The responses shall be denoted by four different *response modes*, as they are defined in table 2.13.

Table 2.13.: Definition of response modes

	Origin of Response for	
Mode	**Closed Projects**	**Running Projects**
`act_lm`	actual values	`_lm` predictions
`act_lm_t`	actual values	`_lm_t` predictions
`lm`	`_lm` predictions	
`lm_t`	`_lm_t` predictions	

These response modes are used in conjunction with four different response types:

- **Absolute** cost/duration
- Cost/duration **estimation error**

By *absolute* cost/duration the actual cost/duration and its predictions themselves are referred to. The *estimation error* measures the error done by estimating actual values at project start and is defined as follows:

Let $x_{\text{act}} :=$ actual value at project end and $x_{\text{est}} :=$ estimation at project start. The formula for the estimation error $x_{\text{est.err.}}$ is

$$x_{\text{est.err.}} := \frac{x_{\text{act}} - x_{\text{est}}}{x_{\text{est}}}$$

and has following properties:

- $x_{\text{est.err.}} \in [-1, \infty)$, because $x_{\text{act}} \geq 0$
- $x_{\text{est.err.}} = c \Leftrightarrow x_{\text{act}} = (1 + c)x_{\text{est}}$, with special cases
  - $x_{\text{est.err.}} = -1 \Leftrightarrow x_{\text{act}} = 0$
  - $x_{\text{est.err.}} = 0 \Leftrightarrow x_{\text{act}} = x_{\text{est}}$

The interpretation of an estimation error of $x_{\text{est.err.}} = c$ would be that the actual value x_{act} is $(1+c)100\%$ of the estimated value x_{est}. In other words, the estimation under-/overestimated the actual value by $(1 + c)100\%$

Remark 2.33

- The analysis is performed on different subsets of `data_trend`, based on business units `bu` and financial project types `finance`:
  - all projects
  - both levels of `finance` separateley (2 variants)
  - each level of `bu` separateley (4 variants)
  - each combination of `finance` and `bu` (8 variants)
- These variants for responses give 4 (response modes) times 4 (response types) = 16 possibilities. Each response possibility is analyzed on each of the $1 + 2 + 4 + 8 = 15$ datasets. Altogether 240 different trend analysis are to be performed.
- To refer to the response types *absolute* and *estimation error* the extensions `_abs` and `_est_err` will be used, respectively. The extension will be attached to `cost` or `dur` to refer to cost and duration.

2.6.2. Results

The results of the trend analyses are presented in tables 2.14 and 2.16 and are accompanied by tables 2.15 and 2.17 giving characteristic numbers (pages 82 to 85). The list below describes the content of these tables in detail (everything refers to tables 2.14 and 2.16, unless stated different):

Cell content (except top row and two rightmost columns): Each cell refers to a trend analysis of a response-data combination. Response type and mode are stated on the table margins and in the table captions. Non significant results[23] are denoted by "–". If a significant trend occurs three characteristics are specified:

Trend direction

↑ upward trend (variable increases over time)
↓ downward trend (variable decreases over time)
The color of the cell text marks the same information: Red text denotes an upward trend and green text a downward trend.

Average slope As for nearly all models a predictor transformation applies, average slopes are given. The average slope is calculated by the simple formula (last fit − first fit)/(time range) and can be interpreted as the yearly increase / decrease of the mean response. As the trend direction is given separately, only the absolute average slope is given.

Significance Beside the trend direction and the average slope it is important to know how significant the trend is, which is coded by significance stars as for regression models (see remark 2.16):
`0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`
The significance stars code the p-value of the hypothesis test that the response is constant over time. Thus the p-value can be interpreted as the probability that there is actually no trend.

Grey colored cells Some cells have a grey background color. This indicates that slope and significance is influenced by a small set of projects (typically one or two projects). Especially for a small set of projects the trend may be influenced by a single project (see example 2.2).

As treatment of these influential projects a conservative approach is chosen: If a small set of projects causes a (more) significant trend, these projects are excluded. If a trend is prevented, the projects are kept. If any of these cases occurs, the corresponding cell is highlighted by a grey background color. On the same time these influential projects are listed afterwards for completeness reasons.

Columns $n_{\mathbf{cl}}$ and $n_{\mathbf{run}}$

$n_{\text{cl}} :=$ number of *closed* projects in the corresponding data subset
$n_{\text{run}} :=$ number of *running* projects in the corresponding data subset

Top row ($\text{pct}_{\mathbf{pr}}$)

$\text{pct}_{\text{pr}} :=$ percentage of projects without `NA` values for the corresponding response[24]

[23] Here results are defined as significant, if the slope p-value ≤ 0.1.
[24] pct_{pr} is nearly the same for response types absolute and estimation error.

An important information is given by the number of projects contained in each dataset. As each response contains missing values for certain projects, in general the numbers n_{cl} and n_{run} do not give the exact number of analyzed projects (projects with available response). But multiplied with pct_{pr} ($n_{\text{cl}} \cdot \text{pct}_{\text{pr}}$ and $n_{\text{run}} \cdot \text{pct}_{\text{pr}}$) these numbers give a better approximation of the number of projects analyzed. The exact numbers are not provided, as the intention is to summarize the results.

Characteristics given by tables 2.15 and 2.17

For each dataset the median $\tilde{x}$ and median absolute standard deviation s_{mad} of the corresponding response is given. The median is a robust[25] estimator of the expected value and s_{mad} is a robust estimator of standard deviation (see Stadlober [23]). A very rough rule of thumb is that about 95% of the data lies in the range of $\tilde{x} \pm 2s_{\text{mad}}$[26].

Remark 2.34

- The most important information about the table contents is summarized at the bottom of both tables.
- The data subset of F1 & BU3 holds only one project. This project has missing values for all responses, thus this category can not be analyzed. It follows that the results for the categories F2 & BU3 and BU3 are exactly the same.

Example 2.2

This example shows how a small set of projects (here two) can influence trend significance. Here the cost estimation error `cost_est_err` of response mode `lm_t` for the projects of F2 & BU1 is analyzed. Table 2.16 gives as result "–", which means that no significant trend was found. On the same time the cell has a grey background, thus the non-significance depends on a small set of projects. Table 2.16 also says that approximately $5 \cdot 0.85 \approx 4$ closed $28 \cdot 0.85 \approx 23$ running projects are analyzed (actually here are no missing values).

The left plot of fig. 2.45 shows the regression line of the non-significant result together with 95% confidence interval (CI) and prediction interval (PI). As the plot title states, the average slope is 4.6% with a p-value of 0.327 (non-significant). The non-significance can be seen as the 95% CI includes a horizontal line. On the same plot two running projects (211 and 459) are labeled. These projects can be identified as influential: 211 pulls the left tail of the regression line up and 459 pulls the right tail down. Hence excluding 211 and 459 may result in an increasing line. Actually it really results in a significant increasing slope, as the right plot of fig. 2.45

[25] Robust against outliers

[26] For $X \sim N(\mu, \sigma^2)$ normally distributed, 95% of a random sample lies within the range of $\mu \pm 1.96\sigma$.

Figure 2.45.: Trend analysis of cost estimation error of F2 & BU1. Response mode: `lm_t`. The left model is generated with data including projects 211 and 459, whereas the right models data does not include these projects. (data: `data_trend`)

shows. The new slope is 11% with a significant p-value of 0.006 (corresponds to significance code "**").

The list of influential points (page 86) states these two projects for cost estimation error, category F2 & BU1 with response mode `lm_t`. As shown above, excluding these two projects results in a significant increase.

Table 2.14.: Results of duration trend analysis (for further explanations see the legend on the bottom)

		act_lm	act_lm_t	lm	lm_t	n_{cl}	n_{run}
	pct_{pr}	90%	87%	85%	79%		
absolute	**all pr.**	–	–	–	–	116	128
	F1	–	–	–	–	31	45
	F2	–	–	–	–	85	83
	BU1	–	–	–	–	5	38
	BU2	↓ 44d .	–	↓ 30d .	–	40	37
	BU3	↓ 64d *	–	↓ 44d *	↓ 54d ***	4	13
	BU4	–	–	–	–	67	40
	F1, BU1	–	–	–	–	0	10
	F1, BU2	↓ 63d .	–	↓ 45d .	–	27	28
	F1, BU3	–	–	–	–	0	1
	F1, BU4	–	–	–	–	4	6
	F2, BU1	↓ 33d *	–	↓ 82d *	–	5	28
	F2, BU2	–	–	–	–	13	9
	F2, BU3	↓ 64d *	–	↓ 44d *	↓ 54d ***	4	12
	F2, BU4	–	–	–	–	63	34
estimation error	**all pr.**	–	–	↓ 5% **	↓ 4% *	113	128
	F1	–	–	–	–	30	45
	F2	–	–	↓ 6% **	↓ 4% *	83	83
	BU1	↓ 13% **	–	↓ 10% **	–	5	38
	BU2	–	–	–	–	39	37
	BU3	–	–	–	–	4	13
	BU4	–	–	–	↓ 4% *	65	40
	F1, BU1	–	–	–	–	0	10
	F1, BU2	–	–	–	–	26	28
	F1, BU3	–	–	–	–	0	1
	F1, BU4	–	–	↓ 25% *	–	4	6
	F2, BU1	↓ 13% ***	–	↓ 10% **	–	5	28
	F2, BU2	–	–	–	–	13	9
	F2, BU3	–	–	–	–	4	12
	F2, BU4	↑ 6% .	–	–	–	61	34

Signif. codes: `0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`
↓ falling trend, ↑ increasing trend, numbers: average abs. slope, d=days
pct_{pr}: percent of projects without `NA` values, n_{cl}/n_{run}: # running resp. closed projects

Table 2.15.: Duration trend analysis: median $\tilde{x}$ and median absolute standard deviation s_{mad}

		act_lm		act_lm_t		lm		lm_t	
		$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}
absolute	**all pr.**	1.23	0.63	1.31	0.72	1.26	0.43	1.37	0.56
	F1	1.35	0.86	1.38	0.93	1.31	0.74	1.46	0.74
	F2	1.19	0.56	1.22	0.65	1.25	0.39	1.30	0.52
	BU1	2.61	1.45	2.92	1.50	2.70	1.58	2.92	1.50
	BU2	1.21	0.66	1.35	0.86	1.21	0.36	1.38	0.50
	BU3	1.12	0.12	1.13	0.26	1.14	0.16	1.31	0.31
	BU4	1.09	0.41	1.12	0.45	1.20	0.33	1.20	0.38
	F1, BU1	3.96	1.48	4.55	1.78	3.96	1.48	4.55	1.78
	F1, BU2	1.28	0.57	1.38	0.91	1.21	0.30	1.43	0.57
	F1, BU3	NA	NA	NA	NA	NA	NA	NA	NA
	F1, BU4	1.24	0.46	1.21	0.09	1.56	0.50	1.14	0.11
	F2, BU1	2.22	1.36	2.72	1.38	2.22	1.37	2.72	1.34
	F2, BU2	1.12	0.57	1.12	0.66	1.21	0.56	1.17	0.47
	F2, BU3	1.12	0.12	1.13	0.26	1.14	0.16	1.31	0.31
	F2, BU4	1.04	0.39	1.09	0.44	1.19	0.31	1.21	0.39
estimation error	**all pr.**	0.59	0.42	0.65	0.51	0.65	0.24	0.69	0.35
	F1	0.74	0.56	0.77	0.65	0.71	0.36	0.79	0.33
	F2	0.57	0.38	0.63	0.48	0.62	0.21	0.68	0.34
	BU1	0.74	0.40	0.93	0.39	0.71	0.36	0.90	0.44
	BU2	0.56	0.47	0.59	0.52	0.59	0.29	0.67	0.36
	BU3	0.50	0.39	0.81	0.77	0.50	0.29	0.78	0.59
	BU4	0.56	0.42	0.53	0.49	0.66	0.20	0.61	0.31
	F1, BU1	1.23	0.56	1.60	0.77	1.23	0.56	1.60	0.77
	F1, BU2	0.57	0.43	0.60	0.52	0.57	0.26	0.76	0.28
	F1, BU3	NA	NA	NA	NA	NA	NA	NA	NA
	F1, BU4	0.93	0.31	0.52	0.77	0.84	0.43	0.27	0.41
	F2, BU1	0.65	0.29	0.84	0.38	0.63	0.22	0.78	0.37
	F2, BU2	0.56	0.50	0.54	0.39	0.65	0.47	0.61	0.34
	F2, BU3	0.50	0.39	0.81	0.77	0.50	0.29	0.78	0.59
	F2, BU4	0.54	0.37	0.53	0.47	0.64	0.19	0.62	0.30

Table 2.16.: Results of cost trend analysis (for further explanations see the legend on the bottom)

		act_lm	act_lm_t	lm	lm_t	n_{cl}	n_{run}
	pct_{pr}	90%	87%	85%	79%		
absolute	**all pr.**	–	–	–	–	116	128
	F1	–	–	–	–	31	45
	F2	–	↑ 34kE .	–	–	85	83
	BU1	–	–	–	–	5	38
	BU2	–	–	↓ 45kE .	–	40	37
	BU3	–	–	–	–	4	13
	BU4	–	–	–	–	67	40
	F1, BU1	–	–	–	–	0	10
	F1, BU2	–	–	↓ 70kE *	↓ 82kE .	27	28
	F1, BU3	–	–	–	–	0	1
	F1, BU4	–	–	–	–	4	6
	F2, BU1	–	–	–	–	5	28
	F2, BU2	–	–	–	–	13	9
	F2, BU3	–	–	–	–	4	12
	F2, BU4	–	–	–	–	63	34
estimation error	**all pr.**	↑ 6% **	↑ 6% **	↑ 3% .	↑ 5% **	113	128
	F1	–	–	–	–	30	45
	F2	↑ 6% **	↑ 5% **	–	↑ 5% **	83	83
	BU1	↑ 14% *	↑ 10% *	↑ 12% *	↑ 8% .	5	38
	BU2	–	–	–	–	39	37
	BU3	–	–	–	–	4	13
	BU4	–	↑ 3% .	–	–	65	40
	F1, BU1	–	↑ 23% *	–	↑ 23% *	0	10
	F1, BU2	–	–	–	–	26	28
	F1, BU3	–	–	–	–	0	1
	F1, BU4	–	–	–	–	4	6
	F2, BU1	↑ 14% *	–	–	–	5	28
	F2, BU2	–	–	↓ 4% .	–	13	9
	F2, BU3	–	–	–	–	4	12
	F2, BU4	–	↑ 3% .	–	–	61	34

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
↓ falling trend, ↑ increasing trend, numbers: average abs. slope, kE=10^3 Euro
pct_{pr}: percent of projects without NA values, n_{cl}/n_{run}: # running resp. closed projects

Table 2.17.: Cost trend analysis: median $\tilde{x}$ and median absolute standard deviation s_{mad}

		act_lm		act_lm_t		lm		lm_t	
		$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}	$\tilde{x}$	s_{mad}
absolute	**all pr.**	0.39	0.37	0.39	0.39	0.39	0.35	0.40	0.34
	F1	0.45	0.48	0.42	0.46	0.43	0.40	0.46	0.43
	F2	0.39	0.33	0.37	0.31	0.39	0.33	0.38	0.31
	BU1	0.86	0.80	0.81	0.74	0.86	0.80	0.81	0.74
	BU2	0.35	0.41	0.40	0.44	0.34	0.32	0.38	0.40
	BU3	0.69	0.30	0.80	0.44	0.65	0.22	0.84	0.34
	BU4	0.30	0.18	0.28	0.17	0.29	0.18	0.27	0.16
	F1, BU1	0.64	0.17	0.60	0.43	0.64	0.17	0.60	0.43
	F1, BU2	0.32	0.36	0.35	0.41	0.31	0.27	0.35	0.35
	F1, BU3	NA	NA	NA	NA	NA	NA	NA	NA
	F1, BU4	0.50	0.27	0.55	0.33	0.39	0.31	0.68	0.15
	F2, BU1	0.88	0.88	0.90	0.77	0.88	0.85	0.90	0.77
	F2, BU2	0.41	0.47	0.41	0.42	0.44	0.37	0.44	0.32
	F2, BU3	0.69	0.30	0.80	0.44	0.65	0.22	0.84	0.34
	F2, BU4	0.29	0.27	0.27	0.15	0.28	0.18	0.26	0.14
estimation error	**all pr.**	0.16	0.42	0.18	0.43	0.19	0.38	0.19	0.39
	F1	0.55	0.86	0.50	0.73	0.48	0.59	0.48	0.52
	F2	0.09	0.29	0.11	0.32	0.10	0.26	0.08	0.28
	BU1	1.31	0.72	0.86	0.46	1.31	0.68	0.81	0.38
	BU2	0.30	0.49	0.35	0.51	0.39	0.30	0.35	0.28
	BU3	0.09	0.22	0.32	0.31	0.09	0.18	0.33	0.12
	BU4	−0.01	0.21	−0.02	0.22	0.00	0.16	−0.04	0.11
	F1, BU1	1.55	0.24	0.94	0.65	1.55	0.24	0.94	0.56
	F1, BU2	0.50	0.63	0.50	0.71	0.47	0.41	0.51	0.50
	F1, BU3	NA	NA	NA	NA	NA	NA	NA	NA
	F1, BU4	−0.13	0.32	0.03	0.43	−0.04	0.37	0.13	0.36
	F2, BU1	1.06	0.62	0.84	0.35	0.98	0.63	0.75	0.34
	F2, BU2	0.14	0.22	0.14	0.24	0.27	0.12	0.27	0.10
	F2, BU3	0.09	0.22	0.32	0.31	0.09	0.18	0.33	0.12
	F2, BU4	0.00	0.19	−0.02	0.20	0.01	0.15	−0.05	0.10

In the following a list of influential points is given (correspond to grey colored cells of tables 2.14 and 2.16) as well as the consequence on trend analysis when including or excluding these projects. Afterwards a separate list gives details on the projects.

- **Duration absolute**
  - all projects
    `lm` excluding 467: significant decrease
  - BU3 (same result for F2 & BU3)
    `act_lm` excluding 251: more significant decrease
    `act_lm_t` excluding 251: significant decrease
- **Duration estimation error**
  - F1
    `act_lm` excluding 195: significant decrease
    `act_lm_t` excluding 195: significant decrease
  - BU1
    `act_lm` including 222, 223, 244, 296: more significant decrease
    `act_lm_t` including 222, 223, 244, 296: significant decrease
    `lm` including 222, 223, 244, 296: more significant decrease
    `lm_t` including 222, 223, 244, 296: significant decrease
  - BU2
    `act_lm_t` excluding 195: significant decrease
  - BU3 (same result for F2 & BU3)
    `act_lm` excluding 425: significant decrease
    `lm` excluding 425: significant decrease
    `lm_t` excluding 425: significant decrease
  - F1 & BU2
    `act_lm` excluding 195: significant decrease
  - F2 & BU1
    `act_lm` including 222, 223, 244, 296: more significant decrease
    `act_lm_t` including 222, 223, 244, 296: significant decrease
    `lm` including 222, 223, 244, 296: more significant decrease
    `lm_t` including 222, 223, 244, 296: significant decrease

- **Cost absolute**
  - BU1
    `act_lm_t` excluding 211, 463: significant increase
    `lm` excluding 211, 463: significant increase
    `lm_t` excluding 211, 463: significant increase
  - F1 & BU4
    `lm` excluding 328: significant increase
  - F2 & BU1
    `act_lm_t` excluding 211, 463: significant increase
    `lm_t` excluding 211, 463: significant increase
  - F2 & BU4
    `lm_t` excluding 399: significant decrease
- **Cost estimation error**
  - BU1
    `lm` excluding 459: more significant increase
    `lm_t` excluding 211, 459: more significant increase
  - BU2
    `act_lm` excluding 418, 446, 464, 477: significant decrease
  - F1 & BU2
    `act_lm` excluding 418, 446, 464, 477: significant decrease
  - F2 & BU1
    `lm` excluding 211, 459: significant increase
    `lm_t` excluding 211, 459: significant increase

The following list gives some details on the influential points mentioned in the list above:

- **Duration absolute**
  467: running, started in 2012, duration about 5.5 (large)
  251: closed, started in 2008 (first project of this dataset), duration about 1 (small)
- **Duration estimation error**
  195: closed, started before 2008, estimation error about -.5 (small)
  222, 223, 244, 296: running, started between 2007 and 2009, large predicted cost estimation errors (see fig. 2.48 in section 2.6.3, Example)
  425: running, started in 211, estimation error about 3 (large)

- **Cost absolute**
  211: running, started 2207, cost about 3.5 (large)
  463: running, started 2012, cost about 0.1 (small)
  328: running, started 2009 (middle), cost about 1.2 (here large) Influential because it increases variance significantly, due to few projects
  399: running, started 211, cost about 1.75 (large)

- **Cost estimation error**
  211: running, started 2007, large estimation error
  459: running, started 2012, small estimation error
  418, 446, 464, 477: running projects, stared after 2011, large predicted estimation errors

2.6.3. Example

This example shall give a deeper understanding of the results and shows how some of the analyzed trends look like. Here also example 2.2 shall be mentioned, as it gives some details about the interpretation of results and also shows how a small set of projects can have major impact on significance.

Lets look at duration estimation error with response mode `act_lm`. The results are listed in to bottom part of table 2.14. For all project the cell gives "–", which means that no significant trend is found. The analyzed dataset consists of 113 closed and 128 running projects, whereas about $113 \cdot 0.9 \approx 102$ closed and about $128 \cdot 0.9 \approx 116$ running projects have non-missing response[27].

All projects: This set is visualized by scatterplots in fig. 2.46. The right plot shows a regression smooth with 95% CI and labels potential otuliers. A regression smooth gives a first idea on how the trend may look like. In contrast to the regression smooth, here by fitted regression an analyzed model[28] is meant. The regression smooth may suggest a slight decreasing trend, also because the 95% CI does not seem to include a horizontal line[29]. The right plot in fig. 2.46 shows the fitted regression line, which demonstrates that there is actually no significant trend. One reason for the misleading regression smooth is given by the clear outliers 233 and 237.

Characteristics: The characteristics table 2.15 says that the median estimation error is 0.59 with a median absolute standard deviation of 0.42. This means:

- On average the actual duration is about 159% of the estimated duration.

[27] The actual numbers are 108 closed and 106 running projects.
[28] Verified model assumptions and analyzed outliers.
[29] A horizontal line means that the estimation error is constant over time.

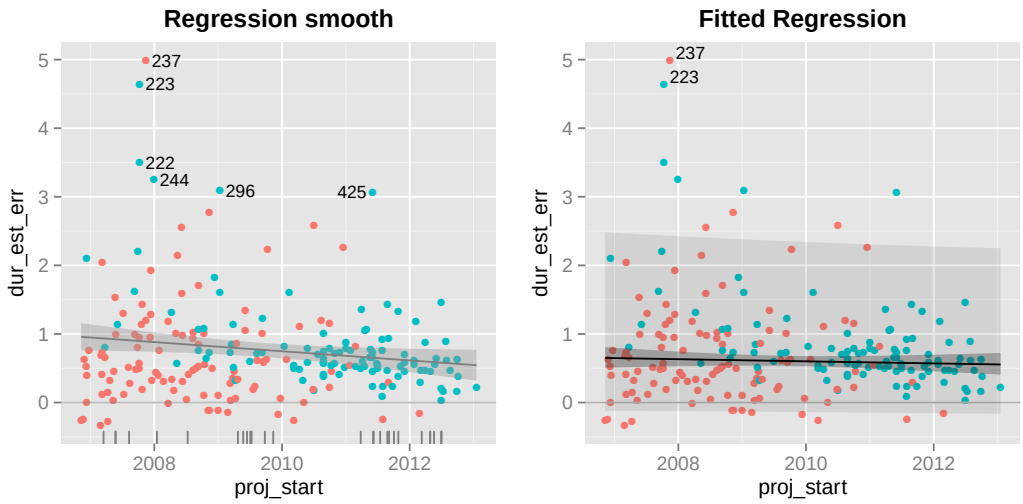


Figure 2.46.: Trend regression smooth and fitted regression line with 95% CI and PI for cost estimation error of all projects. Response mode: `act_lm`. On the right plot general outliers are labeled. The left plot labels projects excluded from regression (data: `data_trend`)

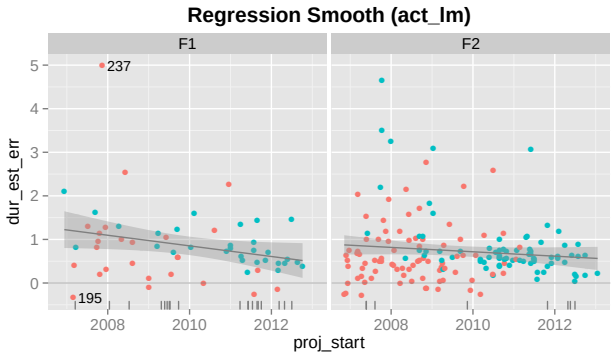
- Roughly with a chance 0.95 the duration estimation error of a random project is within $0.59 \pm 2 \cdot 0.42 = [-0.25, 1.43]$. This means that the actual duration is from 75% to 243% of the estimated duration.

Those values are also visible by regression line and PI on the fitted regression plot (see fig. 2.46).

Subsets: No significant trend for all projects does not necessarily mean that there is also no trend for subsets of these projects. The table of duration trend results 2.14 shows that there are significant trends for BU1, F2 & BU1 and F2 & BU4. These trends are visualized in fig. 2.47 as scatterplots with regression smooths for all 15 subsets. The labeled points represent excluded outliers or influential points (see list of influential points on page 86).

F2 & BU1: As example the project category F2 & BU1 shall be looked at in detail. The result table 2.14 gives "↓ 13% ***". This means that a significant decreasing trend of about 13% per year is found. The significance code *** says that the corresponding p-value is element of $[0, 0.001]$. The p-value can be interpreted as the probability of being wrong in stating that there is a trend.

Looking at fig. 2.47c only, the plot of F2 & BU1 shows a regression smooth with a clear decreasing trend. But this conclusion may be erroneous, as there are 4 running projects that rise the tail of the left regression line. Analyzing this dataset in detail shows that projects 222, 223, 244 and 296 may be kept in the model without violating assumptions, but still are influential points. By choosing a cautious approach, the projects were excluded for giving the results. On the same time the



(a) Scatterplot with regression smooth by financial project types

(b) Scatterplot with regression smooth by business units

(c) Scatterplot with regression smooth by combinations of financial project types and business units.

Figure 2.47.: Trend regression smooths for cost estimation error of all subsets. Response mode: `act_lm` (data: `data_trend`)

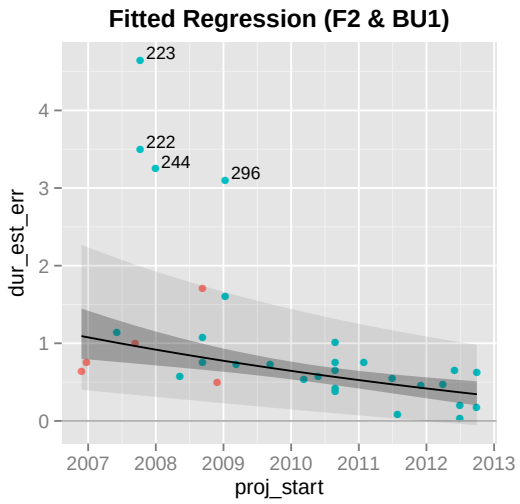


Figure 2.48.: Trend regression line with 95% CI and PI for cost estimation error of F2 & BU1. Labeled points are projects excluded from regression. Response mode: `act_lm` (data: `data_trend`)

grey background in the result table 2.14 and the corresponding comments state the issue of these 4 projects. The average 13% decreasing trend in duration estimation error with significance *** is shown as fitted regression in fig. 2.48.

2.7. Model Application of Previous Thesis

Concluding to the practical analysis part of this thesis, this section applies the models of the previous thesis (see SPONER [21]) to the actual dataset. The models were analyzed and here the results are briefly presented.

2.7.1. Data Relations

The definition of the dataset `data_comp`, that will be used to apply the models of SPONER, follows the idea that the data should be as comparable as possible to the dataset used by SPONER. At the same time full comparability can not be achieved.

First the dataset used by SPONER is briefly described:

- 109 projects
- Data from years 2002 to 2007
- Typical projects that passed milestone M8[30]

[30]Within this thesis defined as closed projects.

- Variables describing the following project characteristics:
  cost, duration, die size, pin count, reuse, size of the technology (`tech`) and business unit
- Variables represent final/actual values

Thus the `data_comp` is based on all closed projects and uses only variables of the final WS (extensions `_act` and `_1`). From the 260 closed projects all 4 projects of business unit BU3 are excluded, because this business unit did not exist like that within the dataset of SPONER.

Remark 2.35

- As the data `data_comp` holds only variables of the final WS, the models analyzed within this section cannot be compared to the models based on the first WS (see section 2.4, Modeling Cost and Duration).
- For the models of this section the unit of project cost is changed to Euro and the unit of duration is changed to days. This is done so, to use the same units as SPONER.
  The plots of the models use again the original units of ME and years for cost and duration respectively.
- Here the original values of `bu` and `tech` are used (before reducing category levels, see section 2.2.2).
- SPONER uses instead of reuse, the variable `negreuse` which is defined as $100 - \texttt{reuse}$.

An issue of comparability of the datasets is that category levels changed in some way over time or were modified. In the following it is described how comparability is achieved for business unit and size of the technology levels as possible:

Business Units (`bu`)

SPONER denoted the levels of `bu` by A, B, C and D. She describes that business unit A and C (correspond to BU1 and BU2) are merged to give a new business unit A.

Since 2007 business units B and D were merged to a business unit that is here named BU4. For the projects of BU4 it cannot be determined with reasonable effort, whether they are rather business unit B or D. As SPONER differs between B and D, here it is decided to treat the new BU4 (containing B and D) the same as B and disregard business unit D. The decision is also based on the fact, that models for business unit D resulted in constant models.

Table 2.18.: Correspondence of business unit levels between `data_comp` and the data of SPONER [21]

	notation	number of projects
`data_comp`	BU1 & BU2	19 + 92 = 111
SPONER	A (& C)	37 + 4 = 41
`data_comp`	BU4	145
SPONER	B (& D)	28 + 40 = 68

Thus the correspondence of business unit levels as stated in table 2.18 is given. The numbers of projects presented also show that the number of projects in each category of the dataset `data_comp` is more than doubled, compared to the data available for SPONER. Of course these numbers are based on the category modifications stated above.

Remark 2.36

- For analyzing cost SPONER developed a model for the union of business units A and B
- The models analyzing duration treated each business unit separately.

Size of the technology (`tech`)

The levels of the size of technology are the same for both datasets. But SPONER merged different levels:

- 0.6 was merged with 0.8, union denoted by 0.6
- 0.13 was merged with 0.35, union denoted by 0.35

This redefinition of the levels of tech is performed also on `data_comp` to be able to compare the datasets.

2.7.2. Model Analysis

Here the models of SPONER are analyzed by performing the following rough steps:

1. Apply model to new dataset `data_comp`
2. Omit outliers giving a comparable model to SPONER
3. Checking for response transformation
4. Checking new variables for significance

Generally speaking, the results are similar to the results of SPONER. Due to more data the coefficient estimates change a bit and the R^2_{adj} decrease. The model formulas could be approved as still being valid, although different response transformations turned out to be more appropriate. On the same time non of the variables that were used by SPONER became additionally significant.

The following list gives the model formulas. Additionally it shows R^2_{adj} values for the original model of SPONER and for the model applied to `data_comp` with old and new response transformation:

Cost model for business units A & B ≙ BU1, BU2 & BU4

orig. formula: `log(cost_act) ~ die_size_l + (100-reuse_l) + tech_l`

	SPONER	0.4524
R^2_{adj}	`data_comp` $\log(\cdot)$ response transformation	0.3399
	`data_comp` $(\cdot)^{1/5}$ response transformation	0.3520

Duration model for business unit A ≙ BU1 & BU2

orig. formula: `dur_act ~ die_size_l + (100-reuse_l)`

	SPONER	0.5034
R^2_{adj}	`data_comp` no response transformation	0.4547
	`data_comp` $(\cdot)^{1/2}$ response transformation	0.4757

Duration model for business unit B ≙ BU4

orig. formula: `dur_act ~ (100-reuse_l)`

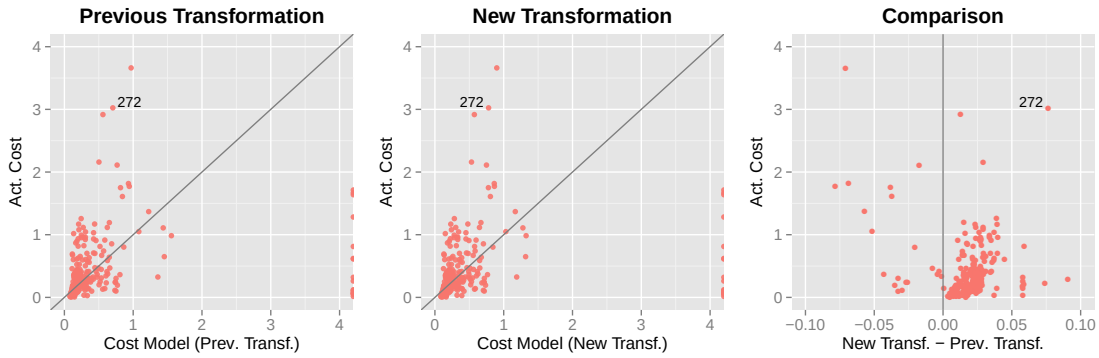
	SPONER	0.5102
R^2_{adj}	`data_comp` no response transformation	0.1599
	`data_comp` $(\cdot)^{1/2}$ response transformation	0.1957

The resulting models are plotted in fig. 2.49, that shows scatterplots of actual values against fitted values for previous and new response transformation as well as against their differences. Each two plots on the right side show now obvious difference. Looking at the plot of the fit differences reveals that the differences are very small. Table 2.19 gives characteristic values for these differences.

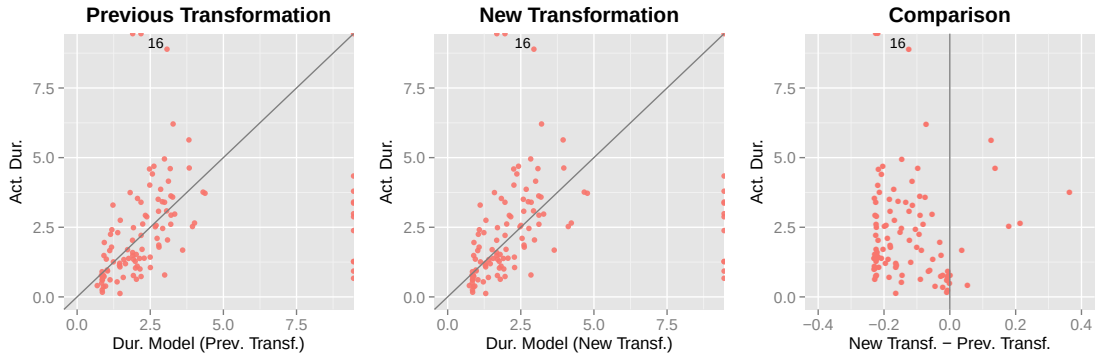
Table 2.19.: Characteristics of model comparison: Difference of fitted values new to previous transformation. Cost values are in Euro and duration values in days (data: `data_trend`)

Model	$\bar{x}$	$\tilde{x}$	s	s_{mad}
Cost (`bu` A & B ≙ BU1, BU2 & BU4)	14 303	19 262	32 497	10 775
Duration (`bu` A ≙ BU1 & BU2)	-43.6	-57.3	44.8	35.6
Duration (`bu` B ≙ BU4)	-15.7	-22.8	16.7	5.1

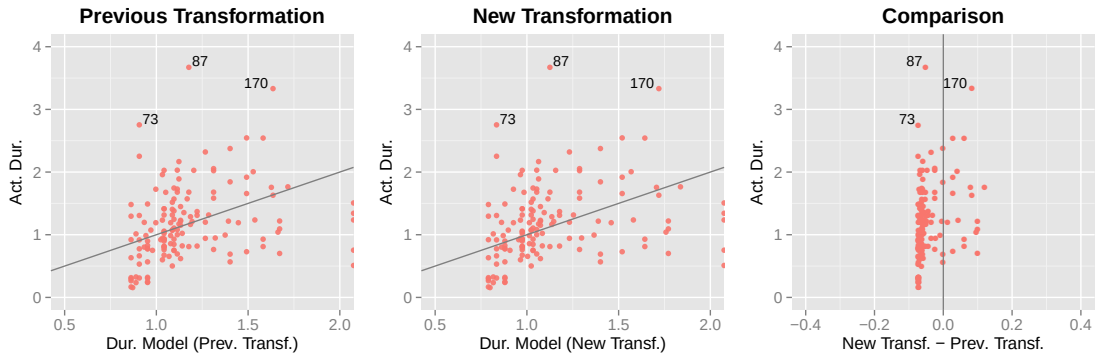
Concluding it can be said that some degree of description for the new extended data `data_trend` is lost, but the models are still valid. The models can still be applied to the data and although other transformations turned out to be more appropriate, the transformations by SPONER give very similar results.



(a) Cost model, `bu` A & B ≙ BU1, BU2 & BU4



(b) Duration model, `bu` A ≙ BU1 & BU2



(c) Duration model, `bu` B ≙ BU4

Figure 2.49.: Application of models by Sponer [21]. Scatterplots of actual values against fitted values. Labeled points are excluded model outliers. For these plots cost is in million Euros and duration in years (data: `data_trend`)

Chapter 3.

Theoretical Fundamentals

This chapter gives a brief introduction to the theoretical fundamentals of the statistical analysis presented in this thesis. The aim is to give a summary and an overview of the theoretic aspects. For further details, please be referred to the corresponding literature. As main references for this chapter we give Fahrmeir, Kneib and Lang [6], Friedl [10], Kleinbaum et al. [14] and Sachs and Hedderich [19] (in alphabetical order).

When analyzing multivariate data, there is a wide range of analysis methods. Kleinbaum et al. [14] give a rough guide to the choice of an appropriate method (see table 3.1), which is based on variable classification. The focus here is on multiple regression analysis, as most predictor variables are continuous.

Table 3.1.: Rough guide to multivariate methods

	Classification of Variables		
Method	**Dependent**	**Independent**	**General Purpose**
Multiple regression analysis	Continuous	Classically all continuous, but in practice any type(s) can be used	To describe the extent, direction, and strength of the relationship between several independent variables and a continuous dependent variable
Analysis of variance	Continuous	All nominal	To describe the relationship between a continuous dependent variable and one or more nominal independent variables

Continued on next page

Table 3.1 – continued from previous page

	Classification of Variables		
Method	**Dependent**	**Independent**	**General Purpose**
Analysis of covariance	Continuous	Mixture of nominal variables and continuous variables (the latter used as control variables)	To describe the relationship between a continuous dependent variable and one or more nominal independent variables, controlling for the effect of one or more continuous independent variables
Logistic regression analysis	Dichotomous	A mixture of various types can be used	To determine how one ore more independent variables are related to the probability of the occurrence of one of two possible outcomes
Poisson regression analysis	Discrete	A mixture of various types can be used	To determine how one ore more independent variables are related to the rate of occurrence of some outcomes

Notation

The following list of descriptions gives a rough guideline to the notation of variables used within this chapter. The guidelines apply to Latin letters and is valid unless a variable is specified differently.

Small (non bold) letters e. g. x, y
Known/fixed constant

Small bold letters e. g. $\boldsymbol{x}$, $\boldsymbol{y}$
Known/fixed vectorized constant
i. e. $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ with known constant $x_i = i$-th entry of $\boldsymbol{x}$

Capital (non bold) letters e. g. X, Y
Random variable or matrix (type will be defined on usage)

Capital bold letters e. g. $\boldsymbol{X}$, $\boldsymbol{Y}$
Vectorized random variable
i. e. $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ with random variable $Y_i = i$-th entry of $\boldsymbol{Y}$

Remark 3.1

- In general. here vectors shall denote column vectors (unless specified different). To denote row vectors, they will be transposed denoted by a superscript "T". For example $\boldsymbol{x}$ transposed is denoted by $\boldsymbol{x}^T$.
- Estimations are denoted by a hat. For example the estimation of β is written as $\hat{\beta}$.
- Greek letters describe unknown variables.

3.1. Simple Linear Regression

The simple linear regression (SLR) model is specified as follows:
Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ be a vector of independent random variables[31] and $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ known observations. The simple linear regression model is described by the relationship

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots, n \tag{3.1}$$

with $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ a random vector of independent and identically distributed (iid) $N(0, \sigma^2)$ variables. Thus $\mathrm{E}(\epsilon_i) = 0$ and $\mathrm{Var}(\epsilon_i) = \sigma^2$. β_0 (intercept), β_1 (slope) and σ^2 (error variance) are unknown constants. In terms of vector notation, equation (3.1) can be expressed by

$$\boldsymbol{Y} = \beta_0 \mathbf{1}_n + \beta_1 \boldsymbol{x} + \epsilon$$

with $\mathbf{1}_n = (1, \ldots, 1)^T$, which links simple to multiple linear regression (see section 3.2).

As a consequence it follows $\mu_i := \mathrm{E}(Y_i) = \beta_0 + \beta_1 x_i$ and $\mathrm{Var}(Y_i) = \sigma^2$. Thus $Y_i \overset{\text{ind}}{\sim} N(\beta_0 + \beta_1 x_i, \sigma^2)$ (ind = independent).

Remark 3.2

- The distribution of the Y_i's is independent, but not identical as the expected value depends on x_i.
- The term *simple* refers to the fact, that there is only one predictor x_i in the SLR model.
- The SLR model is *linear* in the parameters β_0 and β_1.
- The fact that $\mathrm{Var}(Y_i) = \sigma^2$ is constant for all $i = 1, \ldots, n$ is called *Homoscedasticity*.
- $\boldsymbol{Y}$ is called *response* and $\boldsymbol{x}$ is called *predictor* variable.

[31] The realization of $\boldsymbol{Y}$ is denoted by $\boldsymbol{y} = (y_1, \ldots, y_n)^T$. Realizations of variables using $\boldsymbol{Y}$ are given by replacing $\boldsymbol{Y}$ with $\boldsymbol{y}$.

3.1.1. Parameter Estimation

There are three unknown parameters in the SLR model: $\beta_0, \beta_1, \sigma^2$. This section focuses on the estimation of these parameters.

Least-Squares Method

First the parameters β_0, β_1 shall be estimated, such that the Error Sum of Squares (SSE) is minimized.

$$\text{SSE}(\beta_0, \beta_1) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (Y_i - (\beta_0 + \beta_1 x_i))^2$$

Minimizing the SSE with respect to β_0 and β_1 leads to their estimations $\hat{\beta}_0$ and $\hat{\beta}_1$ respectively.

$$\begin{aligned} \hat{\beta}_0 &= \overline{Y} - \hat{\beta}_1 \overline{x} \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \end{aligned}$$

The method of minimizing $\text{SSE}(\beta_o, \beta_1)$ is called *least-squares method*.

Remark 3.3

- The estimations of $\hat{\beta}_0$ and $\hat{\beta}_1$ by the least-squares method are independent of distribution assumptions.
- Based on the estimations $\hat{\beta}_0$, $\hat{\beta}_1$ of β_0, β_1 the expected value of the i-th observation Y_i can be estimated by $\hat{\mu}_i = \widehat{E}(Y_i) = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

Maximum Likelihood (ML) Method

The least-squares method does not give an estimation for σ^2. To obtain an estimation $\hat{\sigma}^2$ by the Maximum Likelihood (ML) method the distribution assumption is needed. Under (3.1) the Log-Likelihood function of the sample is

$$\begin{aligned} \log L(\beta_0, \beta_1, \sigma^2 | \boldsymbol{y}) &= \log \prod_{i=1}^{n} f_i(\beta_0, \beta_1, \sigma^2 | y_i) \\ &= -\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned} \tag{3.2}$$

with $f_i(\cdot)$ the density of the normal distribution $N(\beta_0 + \beta_1 x_i, \sigma^2)$ (see appendix A, Common Statistical Distributions).

Maximizing the Log-Likelihood yields the same estimations for β_0 and β_1 as the least-squares method and an estimation of the variance of $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1)^2 = \frac{1}{n}\,\mathrm{SSE}(\hat{\beta}_0, \hat{\beta}_1)$. This estimation is biased, therefore the unbiased estimation

$$s^2 = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1)^2 = \frac{1}{n-2}\,\mathrm{SSE}(\hat{\beta}_0, \hat{\beta}_1)$$

is used.

Remark 3.4 $n-2$ ca be interpreted as the *degree of freedom* (df) of the estimator s^2, which is the number of observations n minus the number of parameters estimated from the sample (here: $\hat{\beta}_0$ and $\hat{\beta}_1$).

3.2. Multiple Linear Regression

In multiple linear regression the number of predictors is larger then one. In general let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{p-1}$ be a set of known predictor vectors of dimension n. The multiple regression formula is

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.3}$$

with $X = (\mathbf{1}_n, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{p-1})$ a $n \times p$ matrix and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})^T$ the n-dimensional vector of coefficients. Again normal distribution is assumed: $\boldsymbol{Y} \sim N_n(X\boldsymbol{\beta}, \sigma^2 I_n)$[32]. This assumption is equivalent to $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T \sim N_n(0, \sigma^2 I)$.

The multiple regression equation (3.3) can equivalently be expressed as

$$\begin{aligned}
&\boldsymbol{Y} = \beta_0\mathbf{1}_n + \beta_1\boldsymbol{x}_1 + \ldots + \beta_{p-1}\boldsymbol{x}_{p-1} + \boldsymbol{\epsilon}, && \text{or}\\
&Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_{p-1}x_{ip-1} + \epsilon_i, \quad i = 1, \ldots, n, && \text{or}\\
&\begin{pmatrix} Y_1 \\ \vdots \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & \cdots & \cdots & x_{1p-1} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & \cdots & x_{np-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{p-1} \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \vdots \\ \epsilon_n \end{pmatrix}.
\end{aligned}$$

Remark 3.5

- Multiple linear regression is a generalization of simple linear regression.
- The distributions of $\boldsymbol{Y}$ and $\boldsymbol{\epsilon}$ follow a multivariate normal distribution. The single components are independent and normally distributed (see appendix A).

[32] I_n denotes the $n \times n$ identity matrix

- Regression formula (3.3) is *linear* in the coefficients $\beta_0, \ldots, \beta_{p-1}$, but the predictors can also be nonlinear. Lets consider for example the following regression formula:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 \log(x_{i,1}) + \beta_3 x_{i1} x_{i2}^2 + \epsilon_i, \quad i = 1, \ldots, n.$$

  By substituting $\tilde{x}_{i2} \coloneqq \log(x_{i,1})$ and $\tilde{x}_{i3} \coloneqq x_{i1} x_{i2}^2$ the regression formula can be rewritten as

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 \tilde{x}_{i,2} + \beta_3 \tilde{x}_{i,3} + \epsilon_i, \quad i = 1, \ldots, n$$

  which fits the general multiple linear regression formula.

- The expected value of $\boldsymbol{Y}$ shall be denoted by $\boldsymbol{\mu} \coloneqq \mathrm{E}(\boldsymbol{Y})$.

- Throughout the literature (e. g. Sachs and Hedderich [19]) it is also common to define the number of predictor variables $\boldsymbol{x}_i$ as p, instead of $p - 1$. The results are the same.

3.2.1. Parameter Estimation

The procedure of estimating the parameters in multiple linear regression is analogous to the procedure for SLR. Here the $p + 1$ parameters $\beta_0, \beta_1, \ldots, \beta_{p-1}$ and σ^2 have to be estimated.

The estimation method is the same as for SLR. Applying the least squares method leads to minimization of

$$\mathrm{SSE}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = (\boldsymbol{Y} - X\boldsymbol{\beta})^T (\boldsymbol{Y} - X\boldsymbol{\beta}).$$

Minimizing $\mathrm{SSE}(\boldsymbol{\beta})$ results in the following estimation for $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{Y} \tag{3.4}$$

Also the ML method from SLR can be applied analogously to the multiple linear regression model. Inserting the multiple regression model in the Log-Likelihood of the SLR model (equation (3.2)) and using the unbiased version results in the following estimation of σ:

$$s^2 = \frac{1}{n-p}\, \hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}} = \frac{1}{n-p}\, \mathrm{SSE}(\hat{\boldsymbol{\beta}}) \tag{3.5}$$

with estimated error $\hat{\boldsymbol{\epsilon}} = \boldsymbol{Y} - X\hat{\boldsymbol{\beta}}$. $\hat{\boldsymbol{\epsilon}}$ is also called the *residual vector* with *i-th residual* ϵ_i (error or the i-th observation).

Properties

In equation (3.4) it can be seen that each element of $\hat{\boldsymbol{\beta}}$ is a linear combination of the elements of $\boldsymbol{Y}$. Thus because $\boldsymbol{Y}$ is normally distributed, also $\hat{\boldsymbol{\beta}}$ is normally distributed. Mean and covariance can easily be calculated as

$$\begin{aligned}\mathrm{E}(\hat{\boldsymbol{\beta}}) &= (X^T X)^{-1} X^T \mathrm{E}(\boldsymbol{Y}) \overset{\mathrm{E}(\boldsymbol{Y})=\boldsymbol{\beta}}{=} \boldsymbol{\beta} \\ \mathrm{Cov}(\hat{\boldsymbol{\beta}}) &= (X^T X)^{-1} X^T \mathrm{Var}(\boldsymbol{Y}) X (XX^T)^{-1} \overset{\mathrm{Var}(\boldsymbol{Y})=\sigma^2 I_n}{=} \sigma^2 (X^T X)^{-1}.\end{aligned}$$

It follows

$$\hat{\boldsymbol{\beta}} \sim N_p(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1}).$$

The least-squares estimator $\hat{\boldsymbol{\beta}}$ has some preferable properties:

- $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of $\boldsymbol{\beta}$, i. e. $\mathrm{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$.
- *Gauss-Markov-Theorem*: Among all linear and unbiased predictors $\hat{\boldsymbol{\beta}}^L$, the least-squares estimator $\hat{\boldsymbol{\beta}}$ has minimal variance, i. e.

  $$\mathrm{Var}(\hat{\boldsymbol{\beta}}_i) \leq \mathrm{Var}(\hat{\boldsymbol{\beta}}_i^L) \quad \forall\, i = 1 \ldots, n \quad \forall\, \hat{\boldsymbol{\beta}}^L$$

  (see Fahrmeir, Kneib and Lang [6]).
- For the estimated expectation of the response it follows $\hat{\boldsymbol{\mu}} \coloneqq \widehat{\mathrm{E}(\boldsymbol{Y})} = X\hat{\boldsymbol{\beta}}$. The $\hat{\mu}_i$'s are also called *fitted values* or *predictions*.

  The fitted values $\hat{\mu}_i$ are also denoted by $\hat{Y}_i$, which indicates that this variable can be interpreted as an estimation of Y_i. This is among others of interest for new observations.
- By using equation (3.4) $\hat{\boldsymbol{\mu}}$ can be rewritten:

  $$\hat{\boldsymbol{\mu}} = X(X^T X)^{-1} X^T \boldsymbol{Y} = H\boldsymbol{Y}$$

  with $H = X(X^T X)^{-1} X^T$ denoting the symmetric ($H = H^T$) and idempotent ($HH = H$) *hat matrix*.

The residuals (estimated error terms) are defined analogously to SLR and can be rewritten by using the hat matrix:

$$\hat{\boldsymbol{\epsilon}} \coloneqq \boldsymbol{Y} - \hat{\boldsymbol{\mu}} = \boldsymbol{Y} - H\boldsymbol{Y} = (I_n - H)\boldsymbol{Y}.$$

Hence the residuals are as well a linear combination of $\boldsymbol{Y}$ and follow the normal distribution:

$$\begin{aligned}&\mathrm{E}(\hat{\boldsymbol{\epsilon}}) = \mathrm{E}(\boldsymbol{Y}) - \mathrm{E}(\hat{\boldsymbol{\mu}}) \overset{\hat{\boldsymbol{\mu}}=X\hat{\boldsymbol{\beta}}}{=} X\boldsymbol{\beta} - X\boldsymbol{\beta} = 0 \\ &\mathrm{Var}(\hat{\boldsymbol{\epsilon}}) = \sigma^2 (I_n - H) \\ &\hat{\boldsymbol{\epsilon}} \sim N_n(0, \sigma^2 (I_n - H)).\end{aligned}$$

Remark 3.6

- It can be shown that $(\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}})/\sigma^2 \sim \chi^2_{n-p}$. Applying equation (3.5) it follows that $((n-p)/\sigma^2)\ s^2 = \mathrm{SSE}(\hat{\boldsymbol{\beta}})/\sigma^2 \sim \chi^2_{n-p}$.
- With this result it can be shown that s^2 is an unbiased estimator of σ^2:

$$\mathrm{E}\left(\frac{n-p}{\sigma^2}\ s^2\right) = n-p \ \Rightarrow\ \frac{n-p}{\sigma^2}\ \mathrm{E}(s^2) = n-p \ \Rightarrow\ \mathrm{E}(s^2) = \sigma^2.$$

- Despite the model errors ϵ_i are iid distributed, the residuals are not, as the variance depends on the i-th observation ($\mathrm{Var}(\hat{\epsilon}_i) = \sigma^2(1-h_{ii})$, with h_{ii} the i-th diagonal element of the hat matrix H). Therefore also *standardized residuals* r_i are used, which are defined as follows:

$$r_i \coloneqq \frac{\hat{\epsilon}_i}{s\sqrt{1-h_{ii}}}.$$

  Given valid model assumptions the standardized residuals have constant variance.

- Also *studentized residuals* r_i^* are practically used:

$$r_i^* \coloneqq \frac{\hat{\epsilon}_{(i)}}{s_{(i)}\sqrt{1+\boldsymbol{x}_i^T(X_{(i)}^TX_{(i)})^{-1}\boldsymbol{x}_i}}$$

  whereas variables with index in brackets define the respective variable based on all observations excluding the i-th ($n-1$ observations).

- Sometimes the naming of variations of residuals vary:
  Studentized residuals are also called *jackknife residuals* and standardized residuals may also be named as studentized residuals. Thus care about the exact meaning has to be taken.

3.2.2. Analysis of Variance (ANOVA) Table

The Analysis of Variance (ANOVA) provides some basic estimates of variance used in regression analysis.

Remark 3.7 The ANOVA method, as described in table 3.1 (page 97), is closely related to linear regression and uses the principle of the ANOVA table as presented here. For more details see for example Kleinbaum et al. [14].

Obviously the following equations hold

$$(Y_i - \overline{Y}) = (\hat{Y}_i - \overline{Y}) + (Y_i - \hat{Y}_i) \quad \forall\, i = 1, \ldots, n$$
$$\Rightarrow \sum_{i=1}^{n}(Y_i - \overline{Y}) = \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y}) + \sum_{i=1}^{n}(Y_i - \hat{Y}_i).$$

The second equation is also valid in its squared version

$$\underbrace{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}_{=:\ \mathrm{SST}} = \underbrace{\sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2}_{=:\ \mathrm{SSR}} + \underbrace{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}_{=:\ \mathrm{SSE}} \tag{3.6}$$

which can be interpreted as a partition of the variance. This partition is summarized in table 3.2, which also defines the *Mean Error Sum of Squares* (MSE) and the *Mean Regression Sum of Squares* (MSR).

Table 3.2.: ANOVA table

Source of Variation	**Sum of Squares (SS)**	**df**	**Mean SS**
Regression	$\mathrm{SSR} \coloneqq \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2$	$p-1$	$\mathrm{MSR} \coloneqq \frac{\mathrm{SSR}}{p-1}$
Error	$\mathrm{SSE} \coloneqq \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	$n-p$	$\mathrm{MSE} \coloneqq \frac{\mathrm{SSE}}{n-p}$
Total	$\mathrm{SST} \coloneqq \sum_{i=1}^{n}(Y_i - \overline{Y})^2$	$n-1$	

The notations of the ANOVA components are summarized in the following list:

- **SST** Total Sum of Squares
- **SSR** Regression Sum of Squares
- **SSE** Error Sum of Squares

Remark 3.8 Throughout the literature different notations for the summed squares are used. The results stay the same, but care has to be taken when reading different literatures. For example the term SSR is sometimes referred to as *Residual Sum of Squares*, which equals the Error Sum of Square, as it is used here.

The coefficient of determination R^2

This subsection is mainly based on Fahrmeir, Kneib and Lang [6]. The coefficient of determination is defined as follows:

$$R^2 \coloneqq \frac{\mathrm{SSR}}{\mathrm{SST}} \overset{(3.6)}{=} \frac{\mathrm{SST} - \mathrm{SSE}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}.$$

Because of the partition of variance (equation (3.6)) it follows that $0 \leq R^2 \leq 1$. Thus two extreme cases for values of R^2 apply with following interpretations:

1. $R^2 = 1$
   $\Rightarrow$ The error sum of squares SSE $= \sum_{i=1}^{n} \hat{\epsilon}_i^2$ is zero. This means that all residues are equal to zero and the model fits the data perfectly.

2. $R^2 = 0$
   $\Rightarrow$ The regression sum of squares SSR $= \sum_{i=1}^{n} (\hat{Y}_i - \overline{Y})^2$ is zero. This means that $\hat{Y}_i = \overline{Y} \; \forall \; i = 1, \ldots, n$. In other words the predictors have no influence on the predictions.

Remark 3.9 $R^2 = 0$ means that none of the predictors has a linear influence on the response $\boldsymbol{Y}$. But there may be some kind of non-linear influence. For example the square of a predictor variable $\boldsymbol{x}_i^2$ can be influential, while the linear term $\boldsymbol{x}_i$ has no influence.

There are some more notable properties of the coefficient of determination:

- The R^2 value can also be interpreted as the squared correlation coefficient between the response $\boldsymbol{Y}$ and its prediction $\hat{\boldsymbol{Y}} = \hat{\boldsymbol{\mu}}$.
- Adding a new predictor to the multiple linear regression model never decreases the value of R^2, even if the new predictor has no influence.
- When comparing different models by the coefficient of determination, care has to be taken. The response $\boldsymbol{Y}$ has to be the same, the number of parameters $p - 1$ has to be the same and the Intercept has to be included.

3.2.3. Hypothesis Tests

When performing regression analysis, questions about certainty arise. Hypothesis tests about the regression coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})^T$ give some answers. Generally speaking, depending on the application various different variants of hypothesis tests can be considered.

The central assumption used for hypothesis tests and also confidence intervals (see section 3.2.4, Confidence and Prediction Intervals) is the distribution assumption: $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$. Fahrmeir, Kneib and Lang [6] provide a generalization of hypothesis tests on $\boldsymbol{\beta}$:

$$H_0 : C\boldsymbol{\beta} = \boldsymbol{d} \quad \text{against} \quad H_1 : C\boldsymbol{\beta} \neq \boldsymbol{d} \tag{3.7}$$

with C a $r \times p$ matrix such that $\text{rank}(C) = r \leq p$ and $\boldsymbol{d} = (d_1, \ldots, d_r)$.

Remark 3.10

- Fahrmeir, Kneib and Lang [6] also state that these tests are relatively robust against small deviations of the normal distribution assumption. Their book additionally gives a section on the asymptotic validity of these tests.
- For a more general approach that discusses hypothesis tests on linear models in a common sense, please see Stadlober [23].

The hypothesis test statistic[33] can be derived as follows:

1. Get $\text{SSE} = \hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}}$, which is the error sum of squares for the full model (SSE of the model without constraints to β).
2. Get $\text{SSE}_{H_0} \coloneqq \hat{\boldsymbol{\epsilon}}_{H_0}^T \hat{\boldsymbol{\epsilon}}_{H_0}$ with $\hat{\boldsymbol{\epsilon}}_{H_0}$ defined as the residuals of the model under H_0.
3. Define the test statistic F as:

$$F \coloneqq \frac{\frac{1}{r}\left(\text{SSE}_{H_0} - \text{SSE}\right)}{\frac{1}{n-p}\,\text{SSE}}$$

   with r equals the number of rows of C, that is the number of hypothesis equations.

Remark 3.11

- As seen in section 3.2.1, Parameter Estimation the SSE is minimized for the overall model. Thus it derives that $\text{SSE}_{H_0} \geq \text{SSE} \Rightarrow F \geq 0$.
- As the test statistic under H_0 follows a Fisher-F distribution, this test is called *F-test*.

In remark 3.6 it was stated that $\text{SSE}/\sigma^2 \sim \chi^2_{n-p}$. It can also be shown that $(\text{SSE}_{H_0} - \text{SSE})/\sigma^2 \sim \chi^2_r$. On the same time these two random variables are independent and hence F is Fisher-F distributed with r and $n-p$ degrees of freedom: $F \sim F_{r,n-p}$ (see appendix A Common Statistical Distributions). Using this result gives the following decision rule for hypothesis test (3.7):

$$\text{Reject } H_0 \text{ if } F^* > F_{1-\alpha;r,n-p} \tag{3.8}$$

with a given significance level α, $F_{1-\alpha;r,n-p}$ the $1-\alpha$ quantile of the $F_{r,n-p}$ distribution and F^* the realization of F.

[33] A random variable, on which the hypothesis test is based on.

Based on hypothesis test (3.7) three basic types of tests are derived, as suggested by Kleinbaum et al. [14]:

1. **Overall Test:** Is the overall model, i. e. the set of *all predictors*, significant for predicting the response $\boldsymbol{Y}$?
2. **Test for Addition of a Single Variable:** Does adding a *single predictor* variable provide significant additional information in predicting the response $\boldsymbol{Y}$, compared to the model without this predictor?
3. **Test for Addition of a Group of Variables:** Does adding a *set of predictors* provide significant additional information in predicting the response $\boldsymbol{Y}$?

Overall Test

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_{p-1} = 0 \quad \text{against} \quad H_1 : \exists j \in \{1, \ldots, p-1\} : \beta_j \neq 0$$

H_0 can be interpreted as "All $p-1$ independent variables considered together do not explain a significant amount of variation in $\boldsymbol{Y}$" (Kleinbaum et al. [14]). The connection to the general hypothesis test (3.7) is drawn by:

$$C = \begin{pmatrix} 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{and} \quad r = p - 1.$$

Note that the the null-hypothesis gives a model with intercept left as the only coefficient: $Y_i = \beta_0 + \epsilon_i$. Under this model the least-squares estimator of β_0 is $\overline{\boldsymbol{Y}}$, which results in $\text{SSE}_{H_0} = \text{SST}$. Applying this to the test statistic gives:

$$F = \frac{\frac{1}{p-1}\left(\text{SST} - \text{SSE}\right)}{\frac{1}{n-p}\,\text{SSE}} = \frac{\text{SSR}/(p-1)}{\text{SSE}/(n-p)} = \frac{\text{MSR}}{\text{MSE}} \sim F_{p-1,n-p}.$$

Remark 3.12

- According to Fahrmeir, Kneib and Lang [6] for this hypothesis test F can be rewritten as $F = \frac{n-p}{p-1}\,\frac{R^2}{1-R^2}$, which gives the following interpretation: For a small R^2 value F gets small and thus the Null-Hypothesis (the overall model is not significant) is more likely to be kept than for a large R^2 value (close to 1).
- This hypothesis is calculated by the **R** `summary` output of a linear regression model (produced by the `lm` function). It is shown in the bottom line as

  `F-statistics:` ⟨*value*⟩ `on` ⟨*p-1*⟩ `and` ⟨*n-p*⟩ `DF, p-value:` ⟨*value*⟩

  with parameters in angle brackets being replaced by their respective values.

Test for Addition of a Single Variable

$$H_0 : \beta_j = 0 \quad \text{against} \quad H_1 : \beta_j \neq 0, \qquad j = 1, \ldots, p-1$$

In this case H_0 states that the predictor $\boldsymbol{x}_j$ has no additional influence, given all other predictors are in the model. This test fits the general hypothesis test by

$$C = (0, \ldots, 0, 1, 0, \ldots, 0), \quad d = 0 \quad \text{and} \quad r = 1$$

with C being a row vector of all zero entries, except for an entry of 1 at the j-th position. In this special case it can be shown that the test statistic is:

$$F = \frac{\hat{\beta}_j^2}{\widehat{\text{Var}(\hat{\beta}_j)}} \sim F_{1,n-p}.$$

Remark 3.13

- This test is equivalent to the so-called *t-test*, that uses a t-distributed random variable. To be precise $T = \hat{\beta}_j / se_j$ with $se_j = (\widehat{\text{Var}(\hat{\beta}_j)})^{1/2}$ is t_{n-p} distributed. T relates to F by $F = T^2$ (see appendix A Common Statistical Distributions).
- This type of test is done as t-test for every predictor in the **R** `summary` output for linear models. In the following example

```
          Estimate Std. Error t value Pr(>|t|)
pred1   1.380e-01  2.370e-02   5.823 2.14e-08 ***
```

  the coefficient of the predictor `pred1` is $\hat{\beta}_j \approx 0.138$ with $se_j \approx 0.0237$. Thus $T = \frac{0.138}{0.0237} \approx 5.823$. The p-value is equal to $\text{P}(T_{n-p} > 5.823)$ and in this case results as $2.14 \cdot 10^{-8}$. The stars are a graphical representation of the p-value, as stated in section 2.6, Trend Analysis[34].

Test for Addition of a Group of Variabless

Let $\boldsymbol{\beta}_{\text{group}} = (\beta_{i_1}, \ldots, \beta_{i_k})^T$ be a vector of $k \leq p-1$ coefficients.

$$H_0 : \boldsymbol{\beta}_{\text{group}} = 0 \quad \text{against} \quad H_1 : \exists j \in \{1, \ldots, k\} : \beta_{i_j} \neq 0$$

This test also corresponds to a respective representation of the matrix C and vector d in the general hypothesis test (3.7). As $r = k$ the test statistic F is $F_{k,n-p}$ distributed. For more details about the calculation see Fahrmeir, Kneib and Lang [6] or Kleinbaum et al. [14].

[34] `0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`

Remark 3.14

- This test is a generalization of the overall test and the test for addition of a single variable.
- In **R** this test can be performed with the `anova` function of the basic `stats` package. Within this thesis it was for example applied implicitly when analyzing the interactions in the cost/duration models with transformed predictors (see section 2.4, Modeling Cost and Duration).

3.2.4. Confidence and Prediction Intervals

Let $\boldsymbol{x}_{\text{new}} = (1, x_{\text{new},1}, \ldots, x_{\text{new},p-1})$ be a new observation. The expected value of the corresponding response $\mathrm{E}(Y_{\text{new}}) = \boldsymbol{x}_{\text{new}}^T\boldsymbol{\beta}$ can easily be estimated by $\hat{y}_{\text{new}} = \hat{\mu}_{\text{new}} = \boldsymbol{x}_{\text{new}}^T\hat{\boldsymbol{\beta}}$. Beside the point estimator it is often of interest to calculate intervals, that represent the amount of uncertainty.

$1-\alpha$ **Confidence Interval (CI) for** $\mu_{\text{new}} = \mathrm{E}(Y_{\text{new}})$
Interval that holds the true mean response with a certainty of $(1-\alpha)100\%$.

$$\hat{y}_{\text{new}} \pm t_{n-p,1-\alpha/2}\ s\sqrt{\boldsymbol{x}_{\text{new}}^T(X^TX)^{-1}\boldsymbol{x}_{\text{new}}}$$

$1-\alpha$ **Prediction Interval (PI) for** y_{new}
Interval that holds the true response with a certainty of $(1-\alpha)100\%$.

$$\hat{y}_{\text{new}} \pm t_{n-p,1-\alpha/2}\ s\sqrt{1+\boldsymbol{x}_{\text{new}}^T(X^TX)^{-1}\boldsymbol{x}_{\text{new}}}$$

Remark 3.15

- The formula of the confidence interval is based on the distribution of $\hat{\boldsymbol{\beta}}$:

$$\begin{aligned}&\hat{\boldsymbol{\beta}} \sim N_p(\boldsymbol{\beta}, \sigma^2(X^TX)^{-1})\\ &\Rightarrow \boldsymbol{x}_{\text{new}}^T\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{x}_{\text{new}}^T\boldsymbol{\beta}, \sigma^2\boldsymbol{x}_{\text{new}}^T(X^TX)^{-1}\boldsymbol{x}_{\text{new}}).\end{aligned}$$

- The formula of the confidence interval is based on the distribution of $\hat{\boldsymbol{\epsilon}}_{\text{new}} = Y_{\text{new}} - \boldsymbol{x}_{\text{new}}^T\hat{\boldsymbol{\beta}}$:

$$\hat{\epsilon}_{\text{new}} \sim N(0, \sigma^2(1+\boldsymbol{x}_{\text{new}}^T(X^TX)^{-1}\boldsymbol{x}_{\text{new}})).$$

- As the formulas reveal, the prediction interval is wider than the confidence interval.

3.3. Model Diagnostics

When applying linear regression to model relationships, it is very important that the stated assumptions are fulfilled. If the assumptions are violated, all the conclusions of the previous chapters can not be drawn, which may result in erroneous interpretations. On the same time it is important to detect outliers, that have undesirable influence on the model. Thus this section provides an overview of some possibilities to check if the validity of the assumptions is reasonable. For further literature see for example Fahrmeir, Kneib and Lang [6] or Kleinbaum et al. [14].

The focus here is on exploratory methods to check the assumptions. The statistics program **R** provides for this purpose a residual plot for linear models. Within this thesis an analogous self-written plot is used, which is based on the graphical **R**-package `ggplot2`. As a reference fig. 3.1 is used to provide an exemplary residual plot.

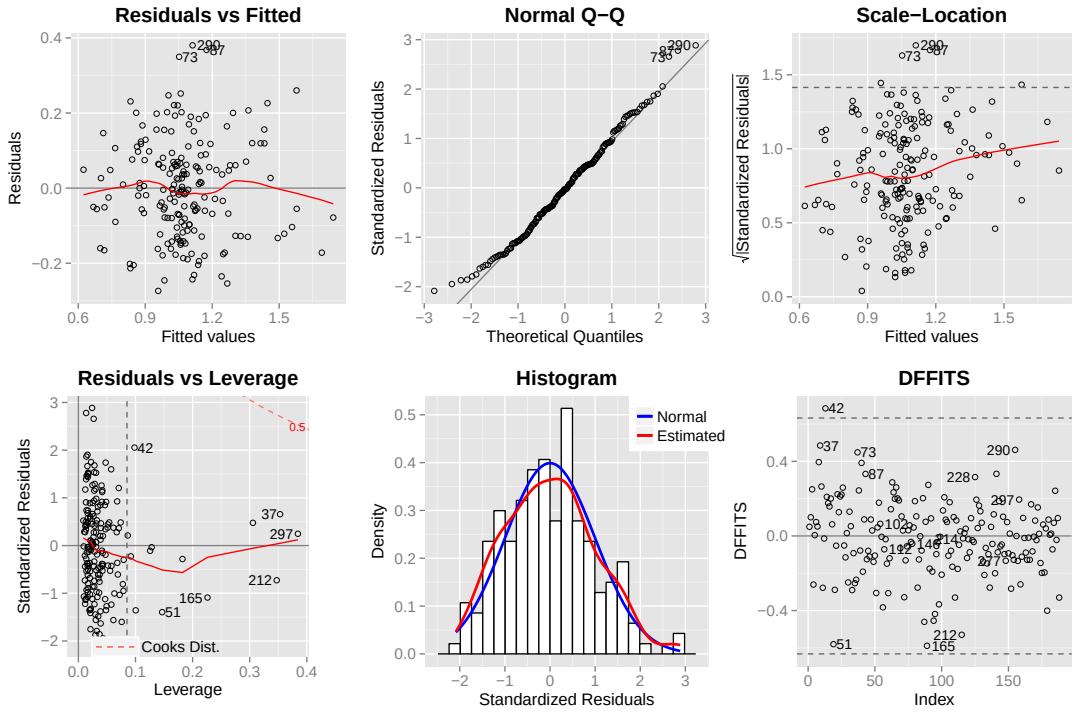


Figure 3.1.: Exemplary residual plot generated with the self-written function `GGplotLm`

Remark 3.16 This remark explains some details about the residual plot as shown in fig. 3.1, which is generated by the self-written function `GGplotLm`. The source code can be found in appendix C.

- The first four plots are basically the same as for the object-oriented plot function `plot.lm` for linear model (`lm`). For more details see the corresponding **R** help-page.
- **Red Smoothing Lines:** The red lines on the plots (except for the histogram) represent the scatterplot smoothing LOWESS (see Cleveland [3]). The main purpose is to the get an idea about the variance.
- **Numbered Points:** The numbered points on the plots are aimed to detect possible outliers.
  - On the first three plots always the three points with largest (absolute) value on the y-axis are marked. This is the same procedure as done by the standard plot procedure `plot.lm`.
  - On the fourth plot *Residuals vs Leverage* the union of the three points with highest leverage value and the three points with largest Cook's distance are labeled by their numbers.
  - The last plot *DFFITS* is based on the **R** function `influence.measures` of the default package `stats`, which calculates some standard leave-one-out deletion statistics for linear models (see e. g. Belsley, Kuh and Welsch [2] and Cook and Weisberg [4]). Generally DFBETAS for all coefficients β_i, DFFITS and COVRATIO are computed. The function also gives possible influential points, which are all marked on the plot, although the scatterplot is based on DFFITS only.
  - For definition of DFBETAS, DFFITS and COVRATIO see section 3.3.2, Influential Observations.

  Here the label numbers on the plots are project numbers.
- **Dotted Lines** The horizontal and vertical dotted lines represent additional levels for possible influential points, as defined underneath.
- **Histogram** The histogram of the standardized residuals additionally shows the estimated density as well as the standard normal density.

3.3.1. Model Assumptions

Linearity

The linearity of the regression function can be checked by plotting the residuals against the fitted values. The residuals should vary randomly around the zero line across the whole spectrum of fitted values. Additionally scatterplots of the residuals against the predictor variables $\boldsymbol{x}_i$ could be used to detect non-linearity.

Homoscedasticity

Non-constancy of error variance can be diagnosed by plotting the (standardized/studentized) residuals against the fitted values or the predictors $\boldsymbol{x}_i$. It is suggested by Fahrmeir, Kneib and Lang [6] to favor standardized or studentized residuals over standard residuals, as standard residuals do not have constant variance themselves.

The reference picture is a constant variation around zero. Fahrmeir, Kneib and Lang [6] provide tests and actions against heteroscedasticity.

Normality Assumption

A graphical tool to check the normality assumption is the normal quantile–quantile plot (Q–Q plot), which plots the sample quantiles against the theoretical quantiles[35] of the normal distribution (see e. g. Stadlober [23]). The reference for normally distributed variables is a straight line of data points.

It is possible to test for normality of the residuals – among others – by using the Shapiro-Wilk test (see Sachs and Hedderich [19] or Stadlober [23]). As the null-hypothesis says that the sample is normally distributed, a small p-value[36] indicates that the normal distribution is violated and a large p-value indicates that there is no contradiction to normal distribution.

Remedy: Response Transformation

According to Kleinbaum et al. [14] there are three primary reasons for using transformations:

1. Stabilize the response variance
2. Normalize the response
3. Linearize the regression model

An important empirical solution to compute a proper response transformation is the *Box-Cox Transformation*. The idea is to find a parameter λ such that the variance of a random variable Y gets independent of the expected value by transforming Y to $T_\lambda(Y)$ with

$$T_\lambda(Y) = \begin{cases} Y^\lambda & \lambda \neq 0 \\ \log(Y) & \lambda = 0 \end{cases}$$

[35] The p-quantile x_p of a random variable X has the property, that $\mathrm{P}(X \leq x_p) \geq p$ and $\mathrm{P}(X \geq x_p) \leq p$ for $p \in (0, 1)$

[36] Typically $p < 0.05$

This leads to a log-likelihood maximization. For more details about the theory see Stadlober [23].

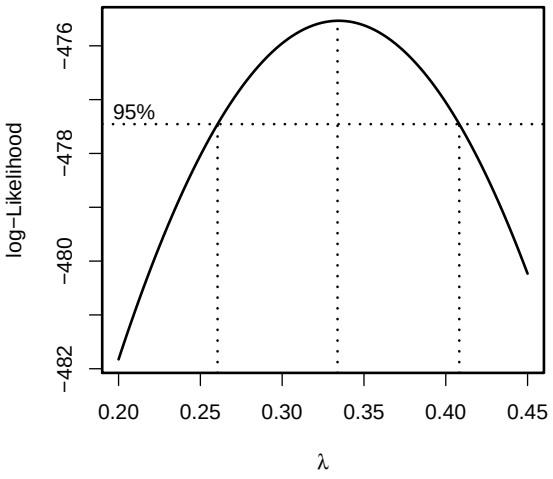


Figure 3.2.: Example of a Box-Cox transformation plot

The **R**-function `boxcox` from the `MASS` package calculates and also optionally plots the empirical solution of log-likelihood maximization (see fig. 3.2). The central vertical line marks the Box-Cox estimation $\hat{\lambda}$ for λ. The two sideways vertical lines mark a 95% confidence interval for λ. In practice a rational number within the 95% CI is chosen to use for the transformation. In the example of fig. 3.2 an appropriate choice would be $\hat{\lambda} = \frac{1}{3}$.

Remark 3.17

- A model with transformed predictor $T_\lambda(Y)$ describes the mean response (i. e. $\mathrm{E}(T_\lambda(Y))$) on the transformed scale, whereas by back-transformation to the original scale the median response is described (see Friedl [9]).
- By applying the inverse transformation the fitted values, confidence interval and prediction interval can be obtained on the original scale. The regression coefficients can not be back-transformed directly.

3.3.2. Influential Observations

Influential observations are observations that have a large impact on the estimations $\hat{\boldsymbol{\beta}}$ and/or $\hat{\boldsymbol{\mu}}$. There are different measurements of influential observations. Some widely used are presented in the following.

Remark 3.18 An influential observation does not necessary mean, that this observation has to be removed from the data, as the observation does not necessarily violate the model assumptions. Anyway it has to be kept in mind, that removing outliers mean a modification of the database. In practice the reason for an observation being influential should be analyzed.

Leverage

The diagonal elements h_{ii} of the hat-matrix $H = x^T(X^TX)^{-1}X$ can be written as

$$h_{ii} = \frac{1}{n} + \frac{1}{n-1}(\boldsymbol{x}_i^1 - \overline{\boldsymbol{x}}_1)^T S^{-1}(\boldsymbol{x}_i^1 - \overline{\boldsymbol{x}}_1)$$

with $\boldsymbol{x}_i^1$ denoting the transposed i-th row of the design matrix X (excluding the intercept) and $\overline{\boldsymbol{x}}_1$ denotes the vector of means and S denotes the covariance matrix of these $\boldsymbol{x}_i^1$. Thus $h_{ii} > \frac{1}{n}$ and h_{ii} grows by growing distance of $\boldsymbol{x}_i$ to $\overline{\boldsymbol{x}}_1$. Observations are called *high-leverage* points, if they fulfill

$$h_{ii} > 2\overline{h} = \frac{2p}{n}$$

(see Friedl [10]).

Remark 3.19

- High-leverage points do not necessarily have to be influential.
- The high-leverage condition is plotted as a vertical dashed line on the *Residuals vs. Leverage* plot of the residual plots (see e. g. fig. 3.1).

Cook's-Distance

The Cook's-distance D_i measures the difference of $\hat{\boldsymbol{Y}}_{(i)}$, the estimation of the mean response with excluding the i-th observation, to the standard mean estimation $\hat{\boldsymbol{Y}}$. The definition is based on their Euclidean distance and can be rewritten in dependence of standardized residuals r_i and diagonal elements h_{ii} of the hat-matrix:

$$D_i \coloneqq \frac{(\hat{\boldsymbol{Y}}_{(i)} - \hat{\boldsymbol{Y}})^T(\hat{\boldsymbol{Y}}_{(i)} - \hat{\boldsymbol{Y}})}{ps^2} = \frac{r_i^2}{p}\left(\frac{h_{ii}}{1-h_{ii}}\right), \quad i = 1, \ldots, n.$$

Thus the Cook's distance is large, if the i-th observation is a high-leverage point or has a large residue. A rule of thumb is to consider observations with $D_i > 0.5$ as conspicuous (see Fahrmeir, Kneib and Lang [6]).

Remark 3.20 Because of the dependence of D_i on r_i and h_{ii} only, the values of the Cook's distance can be drawn as contour lines on the scatterplot of r_i against h_{ii}. As an example see the red dashed line on the bottom left plot *Residuals vs. Leverage* of fig. 3.1.

Other Measurement of Influential Single Observations

Friedl [10] gives the following condition for standardized residuals r_i, that mark potential outliers:

$$|r_i| > 2\sqrt{\mathrm{Var}(r_i)} = 2 \Rightarrow \sqrt{|r_i|} > \sqrt{2} \approx 1.41.$$

This condition is plotted on the *Scale-Location* plots of the residual plots as a horizontal dashed line (see e. g. fig. 3.1).

The following list provides some measurements of the influence of single observations on certain estimators (see Belsley, Kuh and Welsch [2], Cook and Weisberg [4] and Friedl [10]):

- **DFBETAS** Influence of i-th observation on β_j

$$\mathrm{DFBETAS}_{ij} := \frac{\hat{\beta}_j - \hat{\beta}_{j_{(i)}}}{s_{(i)}\sqrt{(X^T X)^{-1}_{jj}}}$$

- **DFFITS** Influence of i-th observation on μ_i

$$\mathrm{DFFITS}_i := \frac{\hat{\epsilon}_i}{s_{(i)}\sqrt{1-h_{ii}}}\sqrt{\frac{h_{ii}}{1-h_{ii}}}$$

- **COVRATIO** Influence of i-th observation on the covariance matrix

$$\mathrm{COVRATIO}_i := \frac{\det\left(s^2_{(i)}(X^T_{(i)}X_{(i)})^{-1}\right)}{\det\left(s^2(X^T X)^{-1}\right)}$$

Remark 3.21

- In the above formulas "det()" denotes the matrix determinant and the index in brackets (as in $s^2_{(i)}$) denotes that the parameter is computed excluding the i-th observation.
- The condition for potential influential observations based on DFFITS by Friedl [10]

$$|\mathrm{DFFITS}| > 3\sqrt{\frac{p}{n-p}}$$

  is plotted on the *DFFITS* plots of the residual plots as horizontal lines (see e. g. bottom right plot in fig. 3.1).

3.4. Model Selection

In multiple linear regression a major question is which predictor variables to choose from a given set of possible predictor variables. Therefore some criteria are needed to compare different models. Especially for a large quantity of predictor variables it is also of interest to apply methods of predictor selection, that overcome the needs of analyzing each single model[37].

In the following sections let the number of possible predictors be m and the numbers of predictors of the current model be $p \leq m$.

3.4.1. General Approaches

Kleinbaum et al. [14] propose the following general steps in selecting the best regression model:

1. **Specify the maximum model to be considered.**
   For specifying the maximum model m potential predictor variables have to be chosen.

2. **Specify a criterion for selecting a model.**
   Typically a set of criteria is used, as each criterion for itself has certain tendencies. According to Friedl [10] a popular strategy for model selection is as follows: Calculate the values of R^2_{adj}, AIC, AIC_c and BIC and compare the models that minimize AIC, AIC_c and BIC with the one maximizing R^2_{adj}.

3. **Specify a strategy for selecting variables.**
   Some strategies are described later within this section.

4. **Conduct the specific analysis.**
   It is needed to diagnose the chosen model, as described in section 3.3, Model Diagnostics.

5. **Evaluate the reliability of the model chosen.**
   This step questions how reliable the chosen model is when applying it to other samples. Kleinbaum et al. [14] briefly discuss three methods to assessing model reliability: The *follow-up-study*, the *split-sample analysis* and the *holdout sample*.

Within this thesis these steps were performed with help of the **R**-function `regsubsets` of the package `leaps` (see Lumley [15]). For more details about the `regsubsets` function and its use within this thesis see section 2.4.1, Model Selection Process. Generally speaking the `leaps` package provides many more functions on model selection procedures.

[37] m possible predictors lead to 2^m possible regression models (without considering interactions or predictor transformations)

Kleinbaum et al. [14] recommend for model selection to generate, based on previous knowledge, a relatively small set of models of interest, that can be evaluated by strategies of step 3. In practice this method is often not applicable due to the analysis structure. Alternative methods often used in practice are described in the following list:

- **All-Subset-Selection (Exhaustive Search)** This method (theoretically) compares all possible models and returns the best models in the sense of a model selection criterion. Kleinbaum et al. [14] suggest this method for $m < 40$.
- **Forward-Selection** Based on a start model this method adds on each step a single variable to the model. The new variable is chosen as the one that gives the best improvement of the model selection criterion. If no improvement is possible, the algorithm stops.
- **Backward-Selection** This method starts with the full model of all m variables. In each step a variable is eliminated from the model. Also here a variable is chosen, which gives the best improvement of the model selection criterion when eliminating it. If no improvement is possible, the algorithm stops.
- **Stepwise-Selection** On stepwise-selection in each step a variable can be added, removed or exchanged, based on the best improvement in the model selection criterion.

3.4.2. Model Selection Criteria

This section focuses on describing different model selection criteria. Within this thesis the described criteria were used simultaneously on choosing a model. The decision process was aided by the self-written function `ModelSelCrit` that visualizes the model selection criteria (see appendix C, Self Written **R** Functions). Fig. 3.3 shows an example plot.

R-squared adjusted (R^2_{adj})

The coefficient of determination is defined as

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}}$$

and can be interpreted as the portion of response variance described by the model. As already mentioned an issue about the R^2 value is, that it never decreases by

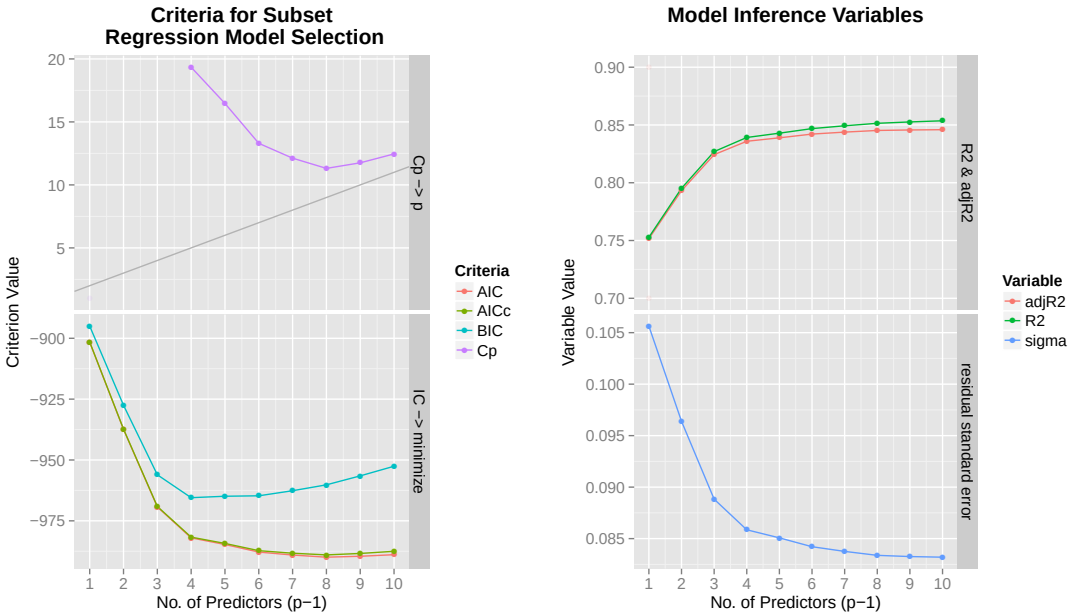


Figure 3.3.: Example of model selection criteria plot generated with the self-written function `ModelSelCrit`

adding variables. Thus maximizing the R^2 value may lead to the full model, even if it contains irrelevant predictors. Hence the adjusted R^2 value R^2_{adj} is defined:

$$R^2_{\text{adj}} := 1 - \frac{\text{SSE}/(n-p)}{\text{SST}/(n-1)}.$$

It can be shown that adding a new predictor leads to an increasing R^2_{adj} value, only if the test statistic of the corresponding hypothesis test (test for addition of a single variable) gives a value greater than one (see Friedl [10]). This leads to an increase of the R^2_{adj} value already for a p-value of about 0.3 (see Fahrmeir, Kneib and Lang [6]). As this may result in the problem of *overfitting*, care has to be taken on using the R^2_{adj} value.

Akaike Information Criterion (AIC)

A popular information criterion is the Akaike Information Criterion (AIC), that measures the balance of high adaption to complexity of the model. The AIC is defined as

$$\text{AIC} := -2 \log L(\hat{\beta}, s^2) + 2(p+1)$$

whereas $L(\hat{\beta}, s^2)$ is the likelihood function, that measures model adaption. Minimizing the AIC leads to a balance of maximizing the likelihood function and minimizing the number of parameters $p+1$. Inserting the multiple regression formula

into the AIC definiton and removing constant summands results in the following form

$$\mathrm{AIC} = n \log \frac{\mathrm{SSE}(\hat{\beta})}{n} + 2p$$

as it is also calculated by **R** (see Friedl [10]).

Corrected Akaike Information Criterion (AIC_c)

Theres is also a corrected version AIC_c of the AIC, as the AIC has the tendency to overfitting for a small number of observations n or a high number of parameters p compared to n. The corrected AIC is defined as

$$\mathrm{AIC}_c \coloneqq -2 \log L(\hat{\beta}, s^2) + 2(p+1) + 2\frac{(p+1)(p+2)}{n-p} = \mathrm{AIC} + \frac{(p+1)(p+2)}{n-p}.$$

According to Friedl [10] AIC_c minimization should be preferred upon the AIC if $n/(p+1) \leq 40$. On the same hand he recommends to use AIC_c in practice, as $\lim_{n\to\infty} \mathrm{AIC}_c = \mathrm{AIC}$.

Bayesian Information Criterion (BIC)

Schwarz [20] introduced the Bayesian Information Criterion (BIC), that is defined very similar to the AIC:

$$\mathrm{BIC} \coloneqq -2 \log L(\hat{\beta}, s^2) + (p+1) \log(n).$$

Compared to the AIC, minimizing the BIC is more likely to favor simple models.

Mallow's C_p

Let $\boldsymbol{\beta}_m$ be the coefficients of the full model and $\boldsymbol{\beta}_p$ be the coefficients of a model with $p \leq m$ parameters. Mallow's complexity parameter C_p is defined as

$$\begin{aligned} C_p &\coloneqq \frac{\mathrm{SSE}(\boldsymbol{\beta}_p)}{\mathrm{SSE}(\boldsymbol{\beta}_m)/(n-m)} - n + 2p \\ &= \frac{\mathrm{MSE}(\boldsymbol{\beta}_p)\ (n-p)}{\mathrm{MSE}(\boldsymbol{\beta}_m)} - n + 2p. \end{aligned}$$

The last formulation shows that the value of C_p is roughly equal to p, if $\mathrm{MSE}(\boldsymbol{\beta}_p)$ is roughly equal to $\mathrm{MSE}(\boldsymbol{\beta}_m)$, which means that the correct model is of size p (see Kleinbaum et al. [14]). As a simple model selection rule Mallows [16] proposes to choose the model that minimizes C_p.

3.5. Principal Component Analysis (PCA)

This section is based on Johnson and Wichern [12] and Jolliffe [13].

The idea of Principal Component Analyis (PCA) is to reduce the dimension of a dataset, while keeping as much variation/information as possible. This is obtained by transforming the variables to a new set of variables, the Principal Components (PCs). The PCs are uncorrelated and ordered such that each PC contains the maximum variance given the previous PCs (see Jolliffe [13]).

3.5.1. Definition and Representation of Principal Components

Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a set of of random variables[38] The principal components PC_i $i = 1, \ldots, p$ are linear combinations of the X_j's, which leads to following general representation by Johnson and Wichern [12]):

$$\begin{aligned}
\mathrm{PC}_1 &= \boldsymbol{a}_1^T \boldsymbol{X} = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\
\mathrm{PC}_2 &= \boldsymbol{a}_2^T \boldsymbol{X} = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\
&\vdots \\
\mathrm{PC}_p &= \boldsymbol{a}_p^T \boldsymbol{X} = a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p.
\end{aligned}$$

Let $\boldsymbol{\Sigma}$ be the covariance matrix of $\boldsymbol{X}$ then it follows

$$\begin{aligned}
\mathrm{Var}(\mathrm{PC}_i) &= \boldsymbol{a}_i^T \boldsymbol{\Sigma} \boldsymbol{a}_i & i &= 1, \ldots, p \\
\mathrm{Cov}(\mathrm{PC}_i, \mathrm{PC}_k) &= \boldsymbol{a}_i^T \boldsymbol{\Sigma} \boldsymbol{a}_k & i, k &= 1, \ldots, p.
\end{aligned}$$

Hence the PCs can be defined as follows: PC_i = linear combination $\boldsymbol{a}_i^T \boldsymbol{X}$, such that $\mathrm{Var}(\boldsymbol{a}_i^T \boldsymbol{X})$ is maximized subject to $\mathrm{Cov}(\boldsymbol{a}_i^T \boldsymbol{X}, \boldsymbol{a}_k^T \boldsymbol{X}) = 0 \; \forall \; k < i$ and $\boldsymbol{a}_i^T \boldsymbol{a}_i = 1$.

Remark 3.22

- The condition $\boldsymbol{a}_i^T \boldsymbol{a}_i = 1$ is necessary, as maximizing $\mathrm{Var}(\boldsymbol{a}_i^T \boldsymbol{X})$ is an unbounded problem. This is because $\mathrm{Var}(cY) = |c| \; \mathrm{Var}(Y) > \mathrm{Var}(Y)$ for all constants c with $|c| > 1$.
- Also other constraints to $\boldsymbol{a}_i$ are possible, e. g. $\max_j |a_{ij}| = 1$ (see Jolliffe [13]).

[38] No distribution assumption is necessary here. On the same time multivariate normality assumption leads to useful interpretations and further inferences (see Johnson and Wichern [12]).

Let $(\lambda_j, \boldsymbol{e}_j), j = 1, \ldots, p$ be the pairs of eigenvalues and eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$. Let these pairs be ordered such that $\lambda_1 > \lambda_2 > \ldots > \lambda_p$. Then it can be shown that the i-th PC is given by

$$\mathrm{PC}_i = \boldsymbol{e}_i^T \boldsymbol{X} = e_{i1}X_1 + e_{i2}X_2 + \cdots + e_{ip}X_p \quad i = 1 \ldots, p \tag{3.9}$$

with

$$\begin{aligned} \mathrm{Var}(\mathrm{PC}_i) &= \boldsymbol{e}_i^T \boldsymbol{\Sigma} \boldsymbol{e}_i = \lambda_i && i = 1, \ldots, p \\ \mathrm{Cov}(\mathrm{PC}_i, \mathrm{PC}_k) &= \boldsymbol{e}_i^T \boldsymbol{\Sigma} \boldsymbol{e}_k = 0 && i, k = 1, \ldots, p. \end{aligned}$$

Remark 3.23 Ideas for proofing the PC result (equation (3.9))

- The result can be constructed by applying the technique of Lagrangian multipliers to the constrained maximization problem stated above (see Jolliffe [13]).
- A direct proof can be performed by looking at $\max_{\boldsymbol{a}} \frac{\boldsymbol{a}^T \boldsymbol{\Sigma} \boldsymbol{a}}{\boldsymbol{a}^T \boldsymbol{a}}$ and using $\boldsymbol{e}_i^T \boldsymbol{\Sigma} \boldsymbol{e}_i = \lambda_i$ and $\boldsymbol{e}_i^T \boldsymbol{e}_i = 1$ (see Johnson and Wichern [12]).

3.5.2. General Remarks about Principal Components

Apply PCA to sampled data

The previous section derived the PC representation for random variables. In practice sampled data is available. The above result can be applied analogously to the data by using the sample covariance matrix $\boldsymbol{S}$ instead of $\boldsymbol{\Sigma}$ (see Johnson and Wichern [12]).

Choosing a Representative Number of PC

An important reason on performing PCA is to describe the information of p variables by a much smaller number of PC variables. There is no unique answer on how many PCs to choose to represent the original data without loosing much information. A visual method is to plot the ordered eigenvalues $\hat{\lambda}_i$ of the sample covariance matrix $\boldsymbol{S}$ against the index i, which is called *scree plot*[39]. Johnson and Wichern [12] suggest to look for an elbow or bend on the scree plot, which index defines the number of appropriate PCs by the corresponding index i. Fig. 3.4 shows an example scree plot that has a clear elbow at $i = 3$. Thus $i = 3$ is the suggested number of representative PCs. The reason is that the remaining eigenvalues have about the same size and are relatively small compared to previous eigenvalues.

For more details and other methods on choosing an appropriate number of principal components see Jolliffe [13].

[39] "Scree is the rock debris at the bottom of a cliff." (Johnson and Wichern [12])

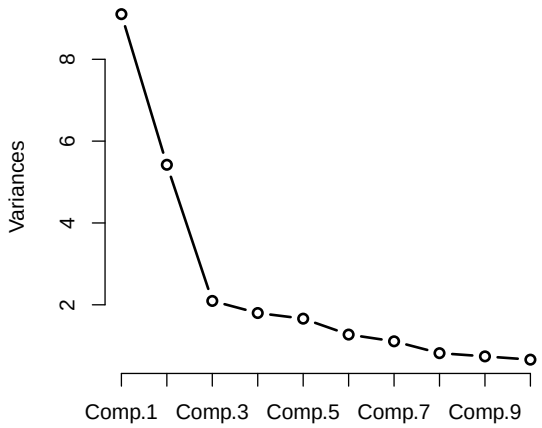


Figure 3.4.: Example scree plot

Computing the Principal Components with R

R provides with its basic package `stats` the function `princomp` that performs PCA for a given dataset. The function returns an object of class `princomp`. In the following let this object be denoted by `fit`. The principal components themselves can be accessed by the command `fit$scores`. Functions of interest to work with a `princomp` object are among others as follows (all part pf basic **R** packages):

- **`summary(fit)`**: Calculates for each PC standard deviation, absolute and cumulative portion of variance.
- **`loadings(fit)`**: Returns the composition of the PCs, i. e. the values of each $\boldsymbol{e}_i$.
- **`plot(fit)`**: A plot of a `princomp` object gives the *scree plot*, as for example fig. 3.4.
- **`biplot(fit)`**: Generates a *biplot*, that shows a scatterplot with two principal components and their loadings on the same plot.

For further details see the corresponding **R** help pages.

There is a wide range of applications and derivations of principle component analysis. For more details and further reading please be referred to Johnson and Wichern [12] and Jolliffe [13].

Chapter 4.

Conclusions

Summary

The main aspect of this thesis is the idea of using the information of closed projects to get unbiased predictions for final project cost and duration of running projects.

By examining the structure of the data using Exploratory Data Analysis (EDA) it turned out, that the estimations of project cost and duration can not be directly used on analyzing running projects. As for running projects in general only the first workstatement (WS) can be assumed to exist, the idea was to generate models based on data of the first WS. As two third of the data are closed projects with known final cost and duration, the closed projects were used to generate these models. By detailed analysis and attaching importance to model diversity, it was possible to retrieve models with a good statistical basis. Further discussion led to the idea of improving models by other available WS. Deeper analysis of the structure of closed projects showed how to obtain significant improvements of model quality by using these WS. Thus the prediction for running projects adapts depending on its progress. Using the resulting models, it was possible to analyze cost and duration trend with many details.

Beside these described steps, high effort was put on data preparation and examination of data structure. On all steps of analysis it was also important to analyze the structure of missing values, to not loose valuable information. On the way to the main purpose of analyzing trends in cost and duration, many interesting side results arose. For example the model improvements showed how project duration and cost develop by WS.

Most graphics shown within this thesis were produced using the graphical **R** package `ggplot2`. Due to the grammar of graphics (see Wilkinson [32]), `ggplot2` is based on graphics for many different purposes. Main functionalities were collected in functions, to make them accessible and to allow modifications easily. A selection of these graphical and other functions is put in the appendix to make them available to others.

The thesis is completed by a theoretical look at regression analysis with related topics as well as Principal Component Analyis (PCA).

Implications and Interpretations

In the central analysis part of cost and duration trend analysis many results were obtained. Generally the results should be seen as indicators for trends. There are four response modes, which can give differing results. Thus an overview look is recommended to interpret the results as a whole. It is also important to notice that the trends are represented by expected values of the response variable (either actual project duration or cost). Especially the examples underline this by intervals that contain the real expected value, thus the real trend, with a chance of 95% (confidence interval (CI)).

The detailed trend analysis of project categories can be used to identify the origin of trends. On the other hand it may reveal that no overall trend hides a trend in some categories.

On interpreting the trends it is also necessary to look at influential projects. As explained, especially in project categories with few projects it may happen that single projects prevent or support the model to detect significant trends. Based on knowledge if an influential project can be regarded as representative or not for the respective category, the interpretation may change.

Concluding the following list gives summarized answers on the research questions stated in chapter 1 Introduction:

1. **Is the database sufficient?**
   Generally the answer is: Yes.

   For some aspects of statistical analysis there is not enough data. Especially the trend analysis revealed that, due to data structure, some project categories hold too few projects to find possible trends.

   This has an important influence on trend analysis interpretation: No indicated significant trend may be a result of no actual trend or of too less data to be recognized as significant trend.

2. **How do project cost and duration develop over the years?**

   Duration There was no overall significant trend found. A deeper look revealed that business units BU2 and BU3 seem to have a more or less decrease in project duration. These trends are originated in the financial project types F1 and F2 respectively. Here also a decreasing trend may be present for the conjunction of F2 & BU1. It shall be remarked that only BU3 and F2 & BU3 shows decreasing trends for 3 of 4 response modes. The other trends could only be verified as significant for two response modes.

Cost BU2 shows for one response mode a tendency to lower project costs. This trend gets more significant when depending BU2 on F1. Only F2 shows slight significant increasing cost. There are two projects (211 and 463) which prevent significant trends for several categories (see grey backgrounds and the influential points list).

For more details see tables 2.14 and 2.16 as well as other results in section 2.6, Trend Analysis.

3. **Did estimations of project cost/duration get better over the years?**

   Duration Project duration estimations at project start show the tendency to have become more accurate in time. There is only one slight significant increase for F2 & BU4. Also partly decreasing trends in the estimation error of all projects can be seen. Many subcategories also indicate decreasing trends. On the same time many cases occur, where non-significance is based on influential projects.

   Cost The cost estimation error shows particularly increasing trends. Here it is interesting that for all projects as well as BU1 separately, the cases appear of (increasing) trends for all response modes. Only F2 & BU2 show a slight decreasing cost estimation error. Fur BU2 and F1 & BU2 a set of 4 projects prevents a significant decrease.

   For more details see tables 2.14 and 2.16 as well as other results in section 2.6, Trend Analysis.

4. **How to get faster and cheaper?**
   This is a question difficult to answer directly. An issue is that for example lowering engineering hours can not be regarded separately. A decrease in a variable practically also effects other variables in certain ways due to dependencies.

   Here the approach to answer this question is to identify influential variables and the sign of their influence. Thus according to the data following actions result – at least by isolated consideration – in faster and cheaper projects:

   Duration

   - Reuse the information of already developed chips (raise reuse).
   - Keep engineering hours and material cost low.
   - Focus on development of chips with low die size.
   - Lower the test time.

Cost

- Keep especially engineering hours and material cost low (correspond directly to project cost).
- Reuse as much information as possible from past projects (raise reuse).

As stated, the above recommendations are based on isolated examinations. The suggestions may be self-evident, nevertheless they represent the most significant influences as described in section 2.4.4, Influential Variables.

Future Prospects

Section 2.5, Model Improvements shows a method to use the information of consecutive WS. It provides an adaptive prediction of model cost and duration as the project progresses. These adaptations used the newly estimated cost and duration only. To further improve models and define models for each WS, further and deeper analysis is necessary. A major issue to consider is that there is no equidistant chronological distribution of WS.

Here the focus was on trend analysis of actual values. Thus all information at project start provided was used to generate models for actual values. This includes estimated duration and indirectly also estimated project cost. As the models are based on these estimations, the model predictions can not be used to replace the estimations without needing them. Statistical analysis could be used to generate models that provide predictions for cost and duration based on basic estimations of project characteristics only.

Based on project start and the date of *today*, passed project duration of running projects can be computed. Using this information for running projects and the actual duration of closed projects, all projects can be put together to apply *Survival Analysis* (see e. g. Tableman and Kim [24], who use **R**/**S** code). First steps in Survival Analysis did not reveal better models for project duration, than the ones presented in this thesis. Nevertheless deeper Survival Analysis can reveal more details about the research objectives.

Finally evaluating the results and outcomes of this thesis in the future may be valuable. Especially the predictions for running projects can be evaluated easily. Additionally, as done with the models by Sponer, the models selected here may be evaluated on relevance for the future.

Appendices

Appendix A.

Common Statistical Distributions

This appendix chapter provides some common statistical distributions with basic properties. References for this chapter are (in alphabetical order) Friedl [8], Sachs and Hedderich [19] and Stadlober [22]. Especially the distribution plots are based on the plots provided by Stadlober [22].

Normal Distribution

Notation $X \sim N(\mu, \sigma^2), \quad \mu \in \mathbb{R},\ \sigma \in \mathbb{R}_+$

Density (see fig. A.1)

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in (-\infty, \infty)$$

Characteristics $\mathrm{E}(X) = \mu, \mathrm{Var}(X) = \sigma^2$

Properties

- $X \sim N(\mu, \sigma^2) \Rightarrow \frac{X-\mu}{\sigma} \sim Z$ with $Z \sim N(0, 1)$
- Let $X_1, \ldots, X_n$ be independent with $X_i \sim N(\mu, \sigma^2)$
  $\overline{X} \coloneqq \frac{1}{n}\sum_{i=1}^n X_i,\ S^2 \coloneqq \frac{1}{n-1}\sum_{i=1}^n \left(X_i - \overline{X}\right)^2 \Rightarrow$
  - $\overline{X}$ and S^2 are independent
  - $\overline{X} \sim N(\mu, \sigma^2/n), \quad \frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0, 1)$
  - $(n-1)S^2/\sigma^2 \sim \chi^2_{n-1}$
- Let $X_1, \ldots, X_n$ be independent with $X_i \sim N(\mu_i, \sigma_i^2)$ and $c_1, \ldots, c_n$ be constants $\Rightarrow Y \coloneqq \sum_{i=1}^n c_i X_i \sim N(\mu, \sigma^2)$ with $\mu = \sum_{i=1}^n c_i\mu_i$ and $\sigma^2 = \sum_{i=1}^n c_i^2\sigma_i^2$

- **Central Limit Theorem (CLT)**
  Let $X_1, X_2, \ldots$ be independent and identically distributed (iid) random variables with $\mathrm{E}(X_i) = \mu$ and $\mathrm{Var}(X) = \sigma^2 < \infty$

$$\Rightarrow \lim_{n\to\infty} \mathrm{P}\left(\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \leq z\right) = \Phi(z), \quad \forall z \in \mathbb{R}$$

  with $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\Phi(z)$ the cumulative distribution function (cdf) of the standard Normal Distribution $N(0,1)$.

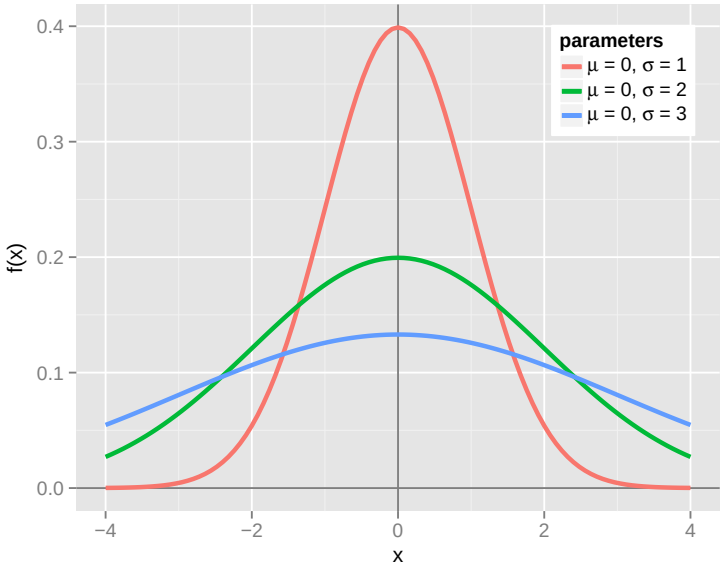


Figure A.1.: Density of $N(\mu, \sigma)$ distribution for $\mu = 0;\ \sigma = 1, 2, 3$

Remark A.1 Multivariate normal distribution
Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ a p-dimensional random vector with density

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{-p/2}|\Sigma|^{-p/2}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right), \quad x \in \mathbb{R}^p$$

with $\boldsymbol{\mu} \in \mathbb{R}^p$ and Σ a positive semidefinite $p \times p$ matrix. $|\cdot|$ denotes the matrix determinant and Σ^{-1} the matrix inverse of Σ.

Then $\boldsymbol{X}$ follows the p-dimensional *multivariate normal distribution* with expected value $\mathrm{E}(\boldsymbol{X}) = \mu$ and covariance matrix $\mathrm{Var}(\boldsymbol{X}) = \Sigma$ denoted by $\boldsymbol{X} \sim N_p(\mu, \Sigma)$.

Properties:

- Let $X_1, \ldots, X_p$ be independent and normally distributed $\Leftrightarrow \boldsymbol{X} = (X_1, \ldots, X_p)$ follows a p-dimensional multivariate normal distribution with $\mathrm{Var}(\boldsymbol{X}) = D$ a diagonal matrix.

- $\boldsymbol{X} = (X_1, \ldots, X_p) \sim N_p(\mu, \Sigma) \Rightarrow$ Each k-dimensional ($k \leq n$) subset $\boldsymbol{Y}$ of $\boldsymbol{X}$ with $\boldsymbol{Y} = (X_{i_1}, \ldots, X_{i_k})$ $\{i_1, \ldots, i_k\} \in \{1, \ldots, n\}$ is multivariate normally distributed. Especially it follows that each X_i with $i \in \{1, \ldots, n\}$ is normally distributed.

For more information on multivariate normal distribution see Fahrmeir, Kneib and Lang [6].

Chi-Squared Distribution

Notation $X \sim \chi^2_n, \quad n \in \mathbb{N}, \quad n$ ... degree of freedom

Density (see fig. A.2)

$$f_n(x) = \begin{cases} \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Characteristics $\mathrm{E}(X) = n, \mathrm{Var}(X) = 2n$

Properties

- $X_1, \ldots, X_n \overset{\text{iid}}{\sim} N(0,1) \Rightarrow \sum_{i=1}^n X_i^2 \sim \chi^2_n$
- $\chi^2_n \sim \gamma(\frac{n}{2}, \frac{1}{2})$ (χ^2_n is a special Gamma distribution)

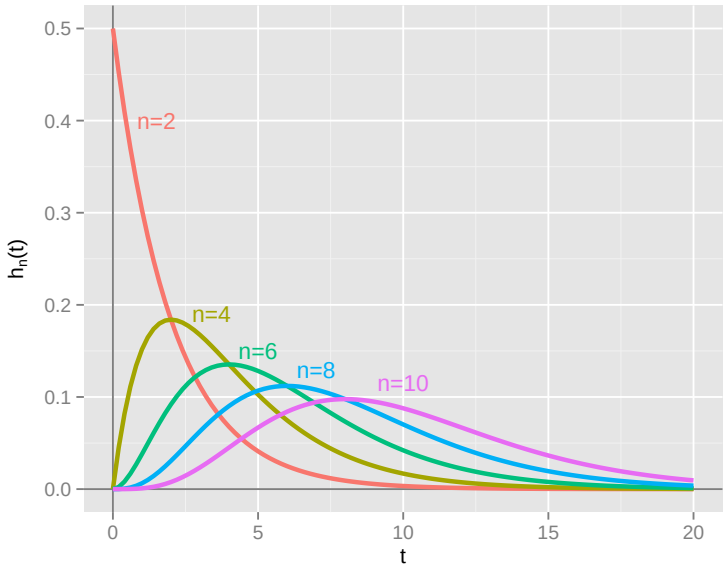


Figure A.2.: Density of χ^2_n distribution for $n = 2, 4, 6, 8$ and 10

Student-t Distribution

Notation $T \sim t_n, \quad n \in \mathbb{N}, \quad n \ \ldots$ degree of freedom

Density (see fig. A.3)

$$f_n(t) = \frac{\Gamma((n+1)/2)}{\Gamma(n/2)\sqrt{n\pi}}(1 + t^2/2)^{-(n+1)/2}, \quad x \in (-\infty, \infty)$$

Characteristics $\mathrm{E}(T) = 0$ (if $n > 1$), $\mathrm{Var}(T) = \frac{n}{n-2}$ (if $n > 2$)

Properties

- Let $X \sim N(0,1)$ and $Y \sim \chi^2_n$ be independent $\Rightarrow \frac{X}{\sqrt{Y/n}} \sim t_n$
- Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} N(\mu, \sigma^2) \Rightarrow \frac{\overline{X}-\mu}{S/\sqrt{n}} \sim t_{n-1}$
- For $n > k$ all moments $\mathrm{E}(T^j)$ with $j < k$ exist.
  For example the distribution of t_1 does not have an expected value.
- For $n \to \infty$ the t_n distribution approximates the standard normal distribution. Thus $t_\infty \sim N(0,1)$.

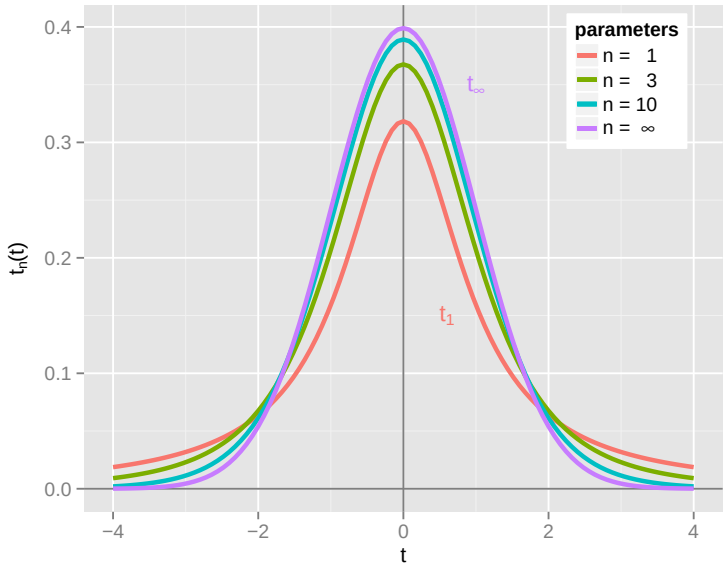


Figure A.3.: Density of t_n distribution for $n = 1, 3, 10$ and the limit case $n = \infty$ ($N(0,1)$ distribution)

Fisher-F Distribution

Notation $F \sim F_{m,n}$

Density (see fig. A.4)

$$f_{m,n}(t) = \begin{cases} \frac{\Gamma((m+n)/2)}{\Gamma(m/2)\Gamma(n/2)} \frac{m}{n} \left(\frac{m}{n} t\right)^{-(m+n)/2} & t > 0 \\ 0 & t \leq 0 \end{cases}$$

Characteristics $\mathrm{E}(F) = \frac{n}{n-2}$ (if $n > 2$), $\mathrm{Var}(F) = \frac{2n^2(m+n-2)}{m(n-2)^2(n-4}$ (if $n > 4$)

Properties

- Let $X \sim \chi^2_m$ and $Y \sim \chi^2_n$ be independent $\Rightarrow$ $\frac{X/m}{Y/n} \sim F_{m,n}$
- Let $X_1, \ldots, X_m \overset{\text{iid}}{\sim} N(\mu_X, \sigma^2_X)$ and $Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} N(\mu_Y, \sigma^2_Y)$ be independent $\Rightarrow$ $\frac{S^2_X/\sigma^2_X}{S^2_Y/\sigma^2_Y} \sim F_{m-1,n-1}$
- $F \sim F_{m,n} \Rightarrow 1/F \sim F_{n,m}$
- $T \sim t_n \Rightarrow T^2 \sim F_{1,n}$

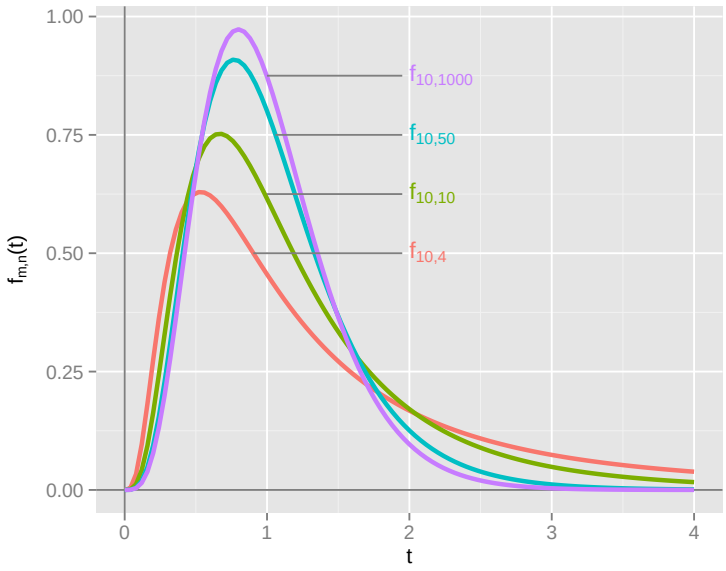


Figure A.4.: Density of $F_{m,n}$ distribution for $m = 10$; $n = 4, 10, 50, 1000$

Appendix B.

R-Packages and Functions

This appendix chapter first describes briefly the used software **R** and afterwards the main used packages and functions.

R

The used software to perform the analysis and to generate the graphics of this work is called **R**, which is a free statistical programming language and software package published under the terms of GNU General Public License (see R Development Core Team [18]). It shall also be mentioned that the author used **R** in conjunction with the editor Tinn-R (see Faria, Grosjean and Jelihovschi [7]). The **R** version used is 2.15.2.

Used Packages and Functions

R is highly extensible by numerous packages available online. A package can be interpreted as a collection of functions for a more or less specific use. All packages and their documentations are available at the Comprehensive R Archive Network (CRAN)[40]. A good overview of **R** and its applications is given by Crawley [5]. To access the help for a single function, just type `?<function-name>` into the command line interface.

The descriptions of the packages and functions are taken from the respective documentations.

[40] http://cran.r-project.org/

`ellipse`: Functions for drawing ellipses and ellipse-like confidence regions

Used Version v0.3-7

Reference Murdoch and Chow [17]

Description "This package contains various routines for drawing ellipses and ellipse-like confidence regions, implementing the plots described in Murdoch and Chow (1996), A graphical display of large correlation matrices, The American Statistician 50, 178-180. There are also routines implementing the profile plots described in Bates and Watts (1988), Nonlinear Regression Analysis and its Applications."

`plotcorr` "This function plots a correlation matrix using ellipse-shaped glyphs for each entry. The ellipse represents a level curve of the density of a bivariate normal with the matching correlation. "

`ggplot2`: An implementation of the Grammar of Graphics

Used Version v0.9.2.1

Reference Wickham [30]

Description "An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources. It also implements a sophisticated multidimensional conditioning system and a consistent interface to map data to aesthetic attributes. See the `ggplot2` website for more information, documentation and examples."

Note `ggplot2` is a graphical package and most of the plots within this thesis are generated with this package. The website for this package[41] is providing a more detailed documentation than the **R** help files do. It contains also many examples.

`leaps`: Regression subset selection

Used Version v2.9

Reference Lumley [15]

Description "Regression subset selection including exhaustive search"

`regsubsets` "Model selection by exhaustive search, forward or backward stepwise, or sequential replacement"

[41] `http://docs.ggplot2.org/`

`MASS`: Support functions and datasets for Venables and Ripley's MASS

Used Version v7.3-22

Reference Venables and Ripley [29]

Description "Functions and datasets to support Venables and Ripley, 'Modern Applied Statistics with S' (4th edition, 2002)."

`boxcox` "Computes and optionally plots profile log-likelihoods for the parameter of the Box-Cox power transformation."

`plyr`: Tools for splitting, applying and combining data

Used Version v1.7.1

Reference Wickham [31]

Description "`plyr` is a set of tools that solves a common set of problems: you need to break a big problem down into manageable pieces, operate on each pieces and then put all the pieces back together. For example, you might want to fit a model to each spatial location or time point in your study, summarise data by panels or collapse high-dimensional arrays to simpler summary statistics. The development of plyr has been generously supported by BD (Becton Dickinson)."

`ddply` "For each subset of a data frame, apply function then combine results into a data frame."

`VIM`: Visualization and Imputation of Missing Values

Used Version v3.0.2

Reference Templ et al. [26]

Description "This package introduces new tools for the visualization of missing and/or imputed values, which can be used for exploring the data and the structure of the missing and/or imputed values. Depending on this structure of the missing values, the corresponding methods may help to identify the mechanism generating the missing values and allows to explore the data including missing values. In addition, the quality of imputation can be visually explored using various univariate, bivariate, multiple and multivariate plot methods. A graphical user interface available in the separate package `VIMGUI` allows an easy handling of the implemented plot methods."

`aggr` "Calculate or plot the amount of missing/imputed values in each variable and the amount of missing/imputed values in certain combinations of variables."

Appendix C.

Self Written R Functions

This chapter presents some **R** functions written by the author. They have different purposes, inputs, outputs and modes, which are briefly described on the beginning of each function. The functions used to generate graphics are based on the **R** package `ggplot2` (see Wickham [30]).

The following functions are provided in alphabetical order:

GGplotFit

```
#-----------------------------------------------------------------
# Name
#   GGplotFit
# Description
#   Plots a model fit of the model 'mod' on a scatterplot of y
#   against x.
# Input (necessary only)
#   mod   the model, which fit to plot
#   x     the x variable for the scatterplot
#   y     the y variable for the scatterplot
#   data  data set to use for plotting and generateing the fitted
#         values
#   It has to be valid: length(x) == length(y) == nrow(data)
# Output
#   -
#-----------------------------------------------------------------
GGplotFit <- function(mod, x, y, data, trans=NULL, xlab=NULL,
    ylab=NULL, llab=NULL, title=NULL, colour=NULL, label_id=c())
{
  if(is.null(trans)) trans <- identity
  if(!is.null(x)){ data$x <- x }else{ stop("empty x") }
  if(!is.null(y)){ data$y <- y }else{ stop("empty y") }
  if(is.null(data)){ data <- data.frame(x=x, y=y) }
  n <- nrow(data)
  n_na_omit <- nrow(mod$model)
  data[, c("fit", "lwr_c", "upr_c")] <-
    trans(predict(mod, newdata=data, interval="confidence",
      level = 0.95))
  data[, c("fit", "lwr_p", "upr_p")] <-
    trans(predict(mod, newdata=data, interval="predict",
      level = 0.95))
  if(!is.null(colour)) data$colour <- colour
  # labels
  if(is.null(xlab)){
    name <- deparse(substitute(x))
    xlab <- GetVarName(name)
  }
  if(is.null(ylab)){
    name <- deparse(substitute(y))
    ylab <- GetVarName(name)
  }
  if(!is.null(colour)){
    if(is.null(llab)){
      name <- deparse(substitute(colour))
      llab <- GetVarName(name)
  }}
  if(is.null(title))
    title <- paste("Trend Analysis:", sQuote(ylab))
  ggplot(data, aes(x=x, y=y)) +
    (if(is.null(colour)){
      geom_point()
```

```
    }else{ geom_point(aes(colour=colour)) })+
    geom_hline(yintercept=0, colour="grey70") +
    scale_colour_discrete(name=llab) +
    geom_ribbon(aes(ymin=lwr_p, ymax=upr_p), alpha=1/10) +
    geom_ribbon(aes(ymin=lwr_c, ymax=upr_c), alpha=1/3) +
    geom_ribbon(aes(ymin=fit, ymax=fit)) +
    geom_line(aes(y=fit)) +
    labs(x=xlab, y=ylab) +
    (if(title!=""){ labs(title=title) }else{ geom_blank() }) +
    BaseTheme(base_size=base_size_) +
    GGplotLabel(data=data, label=data$id,
      subset=(data$id %in% label_id), hjust=Adjust(data$x))
}
#-----------------------------------------------------------------
```

GGplotLabel

```
#-----------------------------------------------------------------
# Name
#   GGplotLabel
# Description
#   Generates a text geom (geom_text) for adding to a
#   ggplot2-object. Is used to label points on a plot.
#   (Code partly from ggplot2 book, partly from 'qqline' function
#   and rest by the author).
# Input
#   data    data of the plot
#   label   labels for the points (default: corresponding rownames
#           of data)
#   subset  subset of the data to label (logical)
#   hjust   horizontal adjustment for the labels
#   vjust   vertical adjustment for the labels
#   print   logical, to decide if labeld points should be printed
#           out
#   plot    if plot = F: return labels only
# Output
#   -
#-----------------------------------------------------------------
GGplotLabel <- function(data, label = rownames, subset = NULL,
  hjust = -0.1, vjust = -0.1, print=F, plot=T, ...)
{
  n = nrow(data)
  if(is.function(label)){
    label <- label(data)
  }else if(is.character(label) & (length(label) == 1)){
    if(!is.null(data[, paste(label)])){
      label <- data[, paste(label)]
    }else{
```

```
      label <- rep(label, n)
    }
  }
  if(is.null(subset))
    subset <- rep(T, n)
  subset[is.na(subset)] <- FALSE
  if(sum(subset) == 0)
    return(geom_blank())
  label <- as.character(label)
  label[is.na(label)] <- ""
  if(length(subset) != n){
    warning("length of 'subset' not compatible with data dim")
    return(geom_blank())
  }
  if(length(label) != n){
    warning("length of 'label' not compatible with data dim")
    return(geom_blank())
  }
  data$label <- label
  if(length(hjust) == 1) hjust <- rep(hjust, n)
  if(length(vjust) == 1) vjust <- rep(vjust, n)
  data$hjust <- hjust
  data$vjust <- vjust
  if(print){
    print(label[subset])
  }
  if(!plot) return(data[subset, ]$label)
  geom_text(data=data[subset, ], aes(label=label, hjust=hjust,
     vjust=vjust),
    size=base_size_*0.3, show_guide=FALSE, ...)
}
#------------------------------------------------------------------
```

GGplotLm

```
#------------------------------------------------------------------
#   GGplotLm
# Description
#   Residual plots for "lm" and "aov" objects.
#   (Code partly from ggplot2 book, partly from qqline function,
#   partly from plot.lm function and rest self-written by the
#   author)
# Input
#   mod   The model to generate the residual plot for.
#   other intput variables work the same as for plot.lm.
# Output
#   Residual Plot
#   Only prints the labeled points for each plot.
```

```
#------------------------------------------------------------------
GGplotLm <- function(mod, which = c(1,2,3,4,5,6),
    cook_levels = c(0.5, 1), id_n = 3,
    caption = list("Residuals vs Fitted",  "Normal Q-Q",
    "Scale-Location", "Residuals vs Leverage", "Histogram",
    "DFFITS"), plot=T)
{
  require(ggplot2)
  require(gridExtra)
  p <- length(coef(mod))
  n <- length(mod$resid)
  if (!is.numeric(which) || any(which < 1) || any(which > 6))
    stop("'which' must be in 1:6")
  plots <- vector('list', 6)
  ifelse(length(which) <= 3, plots$nrow <- 1, plots$nrow <- 2)
  # from plot.lm
  DropInf <- function(x, h, id){
    if (any(isInf <- h >= 1)) {
      warning("Not plotting observations with leverage one:\n
        id = ", paste(id[isInf], collapse = ", "), call. = FALSE)
      x[isInf] <- NaN
    }
    x
  }
  # from qqline
  geom_qqline <- function(mod){
    probs <- c(0.25, 0.75)
    y <- fortify(mod)$.stdresid
    y <- quantile(y, probs, names = FALSE, type = 7, na.rm = TRUE)
    x <- qnorm(probs)
    slope <- diff(y)/diff(x)
    int <- y[1L] - slope * x[1L]
    geom_abline(intercept = int, slope = slope, colour = "grey50",
      size = 0.5)
  }
  # function GetCaption
  GetCaption <- function(k){
    if (length(caption) < k){ NA_character_
    }else{ as.graphicsAnnot(caption[[k]]) }
  }
  df_mod_param <- fortify(mod)
  labels_id <- rownames(df_mod_param)
  # function geom_text_id
  geom_text_id <- function(data, x, ind, i,
      subset_orig_data=rep(T,n))
  {
    subset_label <- 1:n %in% ind
    cat(sQuote(GetCaption(i)), " outliers:\n", sep="")
    hjust = Adjust(x)
    GGplotLabel(data, label = labels_id[subset_orig_data],
      subset = subset_label[subset_orig_data], hjust = hjust,
      vjust = 0.5, print=T)
  }
```

```
show_res <- sort.list(abs(df_mod_param$.resid),
  decreasing = TRUE)[1:id_n]
show_std_res <- sort.list(abs(df_mod_param$.stdresid),
  decreasing = TRUE)[1:id_n]
show_cook_lev <-  union(sort.list(abs(df_mod_param$.cooksd),
  decreasing = TRUE)[1:id_n], sort.list(abs(df_mod_param$.hat),
  decreasing = TRUE)[1:id_n])
# ------------------------------------------------------------
# 1: Residuals vs. Fitted
i <- 1
if(i %in% which){
  plots[[1]] <- ggplot(df_mod_param, aes(.fitted, .resid)) +
    geom_hline(yintercept = 0, colour = "grey50", size = 0.5) +
    geom_point(shape = 1) +
    geom_line(aes(x = lowess(.fitted,.resid)$x,
      y = lowess(.fitted,.resid)$y), colour="red") +
    labs(title = GetCaption(i),
      x = "Fitted values", y = "Residuals") +
    BaseTheme(base_size=base_size_)
  if(id_n) plots[[1]] <- plots[[1]] + geom_text_id(
    df_mod_param, df_mod_param$.fitted, show_res, i)
}
# ------------------------------------------------------------
# 2: Normal Q-Q plot of residuals
i <- 2
if(i%in% which){
  plots[[2]] <- ggplot(mod, aes(sample = .stdresid)) +
    geom_qqline(mod) +
    stat_qq(shape = 1) +
    labs(title = GetCaption(i),
      x="Theoretical Quantiles", y="Standardized Residuals") +
    BaseTheme(base_size=base_size_)
  qq <- qqnorm(df_mod_param$.stdresid, plot.it=F)
  if(id_n){
    hjust = ifelse(qq$x > mean(range(qq$x)), 1.2, -0.2)
    subset <- 1:n %in% show_std_res
    data <- df_mod_param
    data$x <- qq$x; data$y <- qq$y
    cat(sQuote(GetCaption(i)), " outliers:\n", sep="")
    print(rownames(data[subset, ]))
    plots[[2]] <- plots[[2]] +
      geom_text(data=data[subset, ], aes(x, y),
        label=labels_id[subset], hjust=hjust[subset],
        vjust=0.5, size=base_size_*0.3)
  }
}
# ------------------------------------------------------------
# 3: Scale-Location
# (Fitted Values vs. sqrt(abs(Standardized Residuals)))
i <- 3
if(i %in% which){
  lowess_ <- function(...){lowess(...)$y}
```

```
  df_p3 <- data.frame(x  = df_mod_param$.fitted,
    y = sqrt(abs(df_mod_param$.stdresid)))
  subset_orig_data <- !is.nan(df_p3$y)
  df_p3 <- df_p3[subset_orig_data, ]
  plots[[3]] <-
    ggplot(df_p3, aes(x, y)) +
    geom_point(shape = 1) +
    geom_line(aes(x = lowess(x,y)$x, y = lowess(x,y)$y),
      colour="red") +
    geom_hline(yintercept=sqrt(2), color="grey40", linetype=2) +
    labs(title = GetCaption(i),
      x = "Fitted values",
      y = expression(sqrt(abs("Standardized Residuals")))) +
    BaseTheme(base_size=base_size_)
  if(id_n) plots[[3]] <- plots[[3]] +
    geom_text_id(df_p3, df_p3$x, show_std_res, i,
      subset_orig_data=subset_orig_data)
}
# ---------------------------------------------------------------
# 4: Residuals vs. Leverage
i <- 4
if(i %in% which){
  hat <- DropInf(df_mod_param$.hat, df_mod_param$.hat,
    rownames(df_mod_param))
  df_p4 <- data.frame(
    hat = hat,
    stdresid = df_mod_param$.stdresid)
  subset_orig_data <- !is.nan(df_p4$hat)
  df_p4 <- df_p4[!is.nan(df_p4$hat), ]
  plots[[4]] <- ggplot(df_p4, aes(x = hat, y = stdresid)) +
    geom_hline(yintercept = 0, colour = "grey50", size = 0.5) +
    geom_vline(xintercept = 0, colour = "grey50", size = 0.5) +
    geom_point(shape = 1) +
    geom_line(aes(x = lowess(hat,stdresid)$x,
      y = lowess(hat,stdresid)$y), colour="red") +
    geom_vline(xintercept=2*p/n, color="grey40", linetype=2) +
    labs(title = GetCaption(i),
      x = "Leverage", y = "Standardized Residuals") +
    BaseTheme(base_size=base_size_)
  range = data.frame(
    x=ggplot_build(plots[[4]])$panel$ranges[[1]]$x.range,
    y=ggplot_build(plots[[4]])$panel$ranges[[1]]$y.range)
  plots[[4]] <- plots[[4]] +
    coord_cartesian(xlim=range$x, ylim=range$y)
  hh <- seq.int(min(min(df_p4$hat), max(df_p4$hat)/100),
    ceil(range$x[2]*100)/100, l=100)
  df_p4_cook <- data.frame(x=hh, y=p*(1-hh)/hh)
  legend = F
  for(crit in cook_levels){
    df_p4_cook$ysp <- +sqrt(crit*df_p4_cook$y)
    df_p4_cook$ysn <- -sqrt(crit*df_p4_cook$y)
    df_annotate <- subset(df_p4_cook, x == max(x[x<1]))[1, ]
    plots[[4]] <- plots[[4]] +
```

```
        # add cook distance lines
        geom_line(data=df_p4_cook, aes(x=x, y=ysp,
          color="Cooks Dist."), linetype=2) +
        geom_line(data=df_p4_cook, aes(x=x,
          y=ysn,color="Cooks Dist."), linetype=2) +
        theme(plot.margin = unit(c(1,3,1,1), "lines")) +
        # add cook distance annotation
        geom_text(data=df_annotate, aes(x=Inf, y=ysn),
          hjust=1.1, vjust=1, label=crit, color=2,
          size=base_size_*0.25)  +
        geom_text(data=df_annotate, aes(x=Inf, y=ysp),
          hjust=1.2, vjust=0, label=crit, color=2,
          size=base_size_*0.25)
      if(any(df_p4_cook$ysp < range$y[2]) |
          any(df_p4_cook$ysn > range$y[1]))
        legend = T
    }
    if(legend){
      plots[[4]] <- plots[[4]] +
        theme(legend.title=element_blank(),
        legend.justification = 'left',
        legend.position=c(0,0.05),
        legend.background = element_blank(),
        legend.key.width = unit(1*base_size_/10, "cm"))
    } else {plots[[4]] <- plots[[4]] + OmitLegend()}
    if(id_n) plots[[4]] <- plots[[4]] +
      geom_text_id(df_p4, df_p4$hat, show_cook_lev, i,
        subset_orig_data=subset_orig_data)
  }
  # ---------------------------------------------------------------
  # 5: Histogram of residuals
  i <- 5
  if(i %in% which){
    df_range <- df_mod_param[!is.nan(df_mod_param$.stdresid), ]
    range <- max(df_range$.stdresid) - min(df_range$.stdresid)
    plots[[5]] <-
    ggplot(mod, aes(x = .stdresid)) +
      geom_histogram(aes(y = ..density..), colour="black",
        fill="white", binwidth=range/20) +
      stat_function(fun=dnorm, aes(colour="Normal"), size = 1) +
      stat_density(aes(colour="Estimated"), geom="line", size = 1,
        position="identity") +
      labs(title = GetCaption(i),
        x = "Standardized Residuals", y="Density") +
      scale_colour_manual(name = "Density",
        breaks = c("Normal", "Estimated"),
        values = c("red", "blue")) +
      BaseTheme(base_size=base_size_) +
      theme(legend.position = c(0.9, 0.9),
        legend.title = element_blank())
  }
  # ---------------------------------------------------------------
  # 6: DFFITS
```

```
  i <- 6
  if(i %in% which){
    df_p6 <- as.data.frame(influence.measures(mod)$infmat)
    df_p6$index <- 1:nrow(df_p6)
    capture.output(show_infl <- df_p6$index[ rownames(df_p6) %in%
      rownames(summary(influence.measures(mod)))])
    plots[[6]] <- ggplot(df_p6, aes(index, dffit)) +
      geom_hline(yintercept = 0, colour = "grey50", size = 0.5) +
      geom_hline(yintercept = c(-3*sqrt(p/(n - p)),
        3*sqrt(p/(n - p))), color="grey40", linetype=2) +
      geom_point(shape = 1) +
      labs(title = GetCaption(i),
        x = "Index", y = "DFFITS") +
      BaseTheme(base_size=base_size_)
    if(id_n) plots[[6]] <- plots[[6]] +
      geom_text_id(df_p6, df_p6$index, show_infl, i)
  }
  # ---------------------------------------------------------------
  if(plot)
    do.call(grid.arrange, plots[c(which, 7)])
}
#----------------------------------------------------------------
```

GGscatterPlot

```
#----------------------------------------------------------------
# Name
#   GGscatterPlot
# Description
#   Scatterplot built with 'ggplot2' and various options
# Input
#   x,y (necessary): points to plot
#   all other input parameters are optional and can be used to
#   adjust the plot to personal needs
# Output
#   var:          variable names in a single string
#   coefficients: some coefficients about the data
#   labeled:      the labeled points (here together with their
#                 project number generated by 'GetNr' function)
#----------------------------------------------------------------
GGscatterPlot <- function(x, y, colour=NULL, lim_adj=T, xlim=NULL,
    ylim=NULL, line=T, xlab=NULL, ylab=NULL, llab=NULL,
    title=NULL, label=NULL, label_bool=NULL, get_nr=F, smooth=F,
    method="loess", smooth_col="grey50",
    smooth_family="symmetric", zero_vline=T, legend_pos="right",
    plot=T, facet_r=NULL, facet_c=NULL, xlab_ang=0, return_plot=F)
{
  data <- data.frame(x=x, y=y)
```

```
  if(!is.null(colour)) data$colour <- colour
  if(!is.null(facet_r)) data$facet_r <- facet_r
  if(!is.null(facet_c)) data$facet_c <- facet_c

  # for output
  # coefficients
  if(class(x) %in% c("numeric", "integer")){
    cor_test <- cor.test(x=x, y=y, na.action=na.exclude,
      method="pearson")
  }else{ cor_test <- list(estimate=NA, p.value=NA) }
  df_coef <- data.frame(n=nrow(data),
    NAs=nrow(data)-nrow(na.omit(data)),
    NAs_pct=round(100-nrow(na.omit(data))/nrow(data)*100),
    cor = round(cor_test$estimate, 2),
    cor_p_value = round(cor_test$p.value, 4))
  rownames(df_coef) <- "1"
  # point labeling
  if(is.null(label_bool) | is.null(label))
    label_bool <- label <- rep(F, length(var))
  label_bool[is.na(label_bool)] <- F
  # output: labeled
  data$label <- label; data$label_bool <- label_bool
  if(class(labeled <- label[label_bool]) == "factor")
    labeled <- as.character(labeled)
  if(get_nr) labeled <- GetNr(GetNr(as.character(labeled)))
  if(nrow(na.omit(subset(data, label_bool))) == 0)
    data$label_bool <- F
  # define plot labels
  if(is.null(xlab)){
    name <- deparse(substitute(x))
    xlab <- GetVarName(name)
  }
  if(is.null(ylab)){
    name <- deparse(substitute(y))
    ylab <- GetVarName(name)
  }
  if(!is.null(colour)){
    if(is.null(llab)){
      name <- deparse(substitute(colour))
      llab <- GetVarName(name)
  }}
  if(is.null(title)){
    title <- paste(sQuote(xlab), "&", sQuote(ylab))
  }
  if(plot){
    if(!is.null(xlim) | !is.null(ylim)) lim_adj = FALSE
    if(lim_adj) limits <- range(c(x,y), na.rm=T)
    hjust <- Adjust(data$x)
    hjust_xlab <- 0.5 + 0.5*sign(xlab_ang); vjust_xlab <- 0.5
    if(abs(xlab_ang) == 45) vjust_xlab <- 1
    df_miss_x <- data[is.na(data$x), ]
    df_miss_x[is.na(df_miss_x)] <- -Inf
    df_miss_y <- data[is.na(data$y), ]
```

```
    df_miss_y[is.na(df_miss_y)] <- -Inf
    p <- ggplot(data, aes(x, y)) +
      (if(!is.null(colour)){
        geom_point(aes(colour=colour))
      }else{ geom_point() }) +
      scale_colour_discrete(name=llab) +
      geom_hline(yintercept=0, colour="gray50", alpha=0.5) +
      (if(zero_vline){
        geom_vline(xintercept=0, colour="gray50", alpha=0.5)
      }else{ geom_blank() }) +
      BaseTheme(base_size=base_size_) +
      theme(axis.text.x =
        element_text(angle = xlab_ang, hjust = hjust_xlab,
        vjust = vjust_xlab)) +
      (if(line){
        geom_abline(intercept=0, slope=1, colour="grey50")
      }else{ geom_blank() }) +
      (if(lim_adj & (class(x) == "numeric")){
        xlim(limits) }else{ geom_blank() }) +
      (if(lim_adj){ ylim(limits) }else{ geom_blank() }) +
      (if(!is.null(xlim)){ xlim(xlim) }else{ geom_blank() }) +
      (if(!is.null(ylim)){ ylim(ylim) }else{ geom_blank() }) +
      labs(x = xlab, y = ylab) +
      (if(title %in% c("", NA, "none", FALSE))
        { geom_blank() }else{ labs(title = title) }) +
      (if(smooth){
        if(method == "loess"){
          # family="symmetric": make loess robust against outliers
          if(smooth_col != "aes"){ geom_smooth(method=method,
            family=smooth_family, colour=smooth_col)
          }else{ geom_smooth(method=method, family=smooth_family,
              aes(colour=colour)) }
        }else{
          if(smooth_col != "aes"){
            geom_smooth(method=method, colour=smooth_col)
          }else{ geom_smooth(method=method, aes(colour=colour)) }
        }
      }else{ geom_blank() }) +
      (if(any(is.na(data$x))){ geom_rug(data=df_miss_x, side="l",
        colour="grey50", alpha=1)
      }else{ geom_blank() }) +
      (if(any(is.na(data$y))){ geom_rug(data=df_miss_y, side="b",
        colour="grey50", alpha=1)
      }else{ geom_blank() }) +
      theme(legend.position = legend_pos) +
      (if(!is.null(facet_r)){
        if(is.null(facet_c)){ facet_grid(.~facet_r)
        }else{ facet_grid(facet_c~facet_r) }
      }else{
        if(!is.null(facet_c)){ facet_grid(facet_c~.)
        }else{ geom_blank() }
      }) +
      GGplotLabel(data, label = data$label,
```

```
        subset = data$label_bool, hjust=hjust, vjust=0.5, plot=T)
    if(!return_plot){
      print(p)
    }else{
      return(p)
    }
  }
  list(var=paste(xlab, "&", ylab), coefficients=df_coef,
    labeled=labeled)
}
#-----------------------------------------------------------------
```

ModelSelCrit

```
#-----------------------------------------------------------------
# Name
#   ModelSelCrit
# Description
#   Plots model selection criteriea for each model size, based on
#   a regsubsets object.
# Input
#   regs  'regsubsets' object
#   data  data used to generate regs
#   plot  should the plot be generated? (logical)
#   lm    optional 'lm' object: if given, model selection
#         critera are calclated for this model only.
# Output
#   ret   data frame giving criteria values for each model size
#   additionally the predictors of each model are printed
#-----------------------------------------------------------------
ModelSelCrit <- function(regs, data, plot=T, lm=NULL){
  if(!is.null(lm)){
    n <- length(lm$residuals)
    p <- length(lm$coef)
    # AIC = - 2*log L + k * edf
    AIC <- extractAIC(lm, k=2)[2]
    AICc <- extractAIC(lm, k=2)[2] + 2*p*(p+1)/(n-p-1)
    BIC <- extractAIC(lm, k=log(n))[2]
    sigma <- summary(lm)$sigma
    R2 <- summary(lm)$r.squared
    adjR2 <- summary(lm)$adj.r
    ret <- data.frame(AIC, AICc, BIC, adjR2, R2, sigma)
    return(round(ret, 3))
  }
  # apply regsubsets function from package leaps
  require(leaps)
  regs_sum <- summary(regs)
  # extract relevant information
```

```
predictors <- colnames(regs_sum$which)[-1]
model_matrix <- regs_sum$which[, -1]
nr_models <- dim(model_matrix)[1]
n <- nrow(data)
# coefficients to calculate
nr_pred <- AIC <- AICc <- BIC <- sigma <- R2 <- adjR2 <-
  rep(0, nr_models)
for(i in 1:nr_models)
{
  p <- length(coef(regs, i)) # = nr of pred (incl. intercept)
  nr_pred[i] <- p - 1
  predictors <- paste(names(coef(regs, i))[-1],
    sep="", collapse=" + ")
  modi <- lm(as.formula(paste("y ~ ", predictors)), data=data)
  AIC[i] <- extractAIC(modi, k=2)[2]
  AICc[i] <- extractAIC(modi, k=2)[2] + 2*p*(p+1)/(n-p-1)
  BIC[i] <- extractAIC(modi, k=log(n))[2]
  sigma[i] <- summary(modi)$sigma
  R2[i] <- summary(modi)$r.squared
  adjR2[i] <- summary(modi)$adj.r.squared
  cat("Pred. Mod.", i, ":", predictors, "\n")
}
cat("\n")
Cp <- regs_sum$cp
ret <- data.frame(AIC, AICc, BIC, Cp, adjR2, R2, sigma)
if(plot){
  require(ggplot2); require(gridExtra)
  l <- min(ret[, c("R2", "adjR2")])
  h <- max(ret[, c("R2", "adjR2")])
  l <- floor(10*l)/10; h <- ceil(10*h)/10
  # plot criteria variables
  df_plot1 <- data.frame(nr_pred = rep(nr_pred,
      times=dim(ret[, 1:4])[2]),
    crit_val = as.vector(as.matrix(ret[, 1:4])),
    crit = rep(names(ret)[1:4], each=nr_models),
    group = rep(c(rep("IC -> minimize", 3), "Cp -> p"),
      each=nr_models),
    alpha = rep(1, nr_models*4))
  df_plot1$crit_val[(df_plot1$crit == "Cp") &
    (df_plot1$crit_val > 20)] <- NA
  df_plot1 <- rbind(df_plot1, data.frame(nr_pred = min(nr_pred),
    crit_val = 1, crit = "Cp", group = "Cp -> p", alpha = 0))
  p1 <- ggplot(df_plot1, aes(x=nr_pred, y=crit_val, group=crit,
      colour=crit)) +
    BaseTheme(base_size=base_size_) +
    geom_line(data=subset(df_plot1, alpha==1)) +
    geom_point(aes(alpha=alpha)) +
    geom_abline(intercept=1, slope=1, colour="grey70") +
    labs(x="No. of Predictors (p-1)",
      y="Criterion Value",
      colour="Criteria",
      title = "Criteria for Subset\n
        Regression Model Selection") +
```

```
      scale_x_continuous(breaks=min(nr_pred):max(nr_pred)) +
      facet_grid(group~., scale="free") +
      scale_alpha(guide="none")
    # plot of standard error sigma, R2 and R2_adj
    df_plot2 <- data.frame(
      nr_pred = rep(nr_pred, times=dim(ret[, 5:7])[2]),
      val = as.vector(as.matrix(ret[, 5:7])),
      var = rep(names(ret)[5:7], each=nr_models),
      group = rep(c("R2 & adjR2", "R2 & adjR2",
        "residual standard error"), each = nr_models),
      alpha=rep(1, nr_models*3))
    df_plot2 <- rbind(df_plot2,
      data.frame(nr_pred = rep(min(nr_pred), 2),
      val = c(1,h), var = rep("adjR2", 2),
      group = rep("R2 & adjR2", 2), alpha = rep(0, 2)))
    p2 <- ggplot(df_plot2,
        aes(x=nr_pred, y=val, group=var, colour=var)) +
      BaseTheme(base_size=base_size_) +
      geom_line(data=subset(df_plot2, alpha==1)) +
      geom_point(aes(alpha=alpha)) +
      labs(x="No. of Predictors (p-1)",
        y="Variable Value",
        colour="Variable",
        title = "Model Inference Variables\n") +
      scale_x_continuous(breaks=min(nr_pred):max(nr_pred)) +
      facet_grid(group~., scale="free") +
      scale_alpha(guide="none")
    print(grid.arrange(p1, p2, nrow=1))
  }
  round(ret, 3)
}
#-----------------------------------------------------------------
```

MySummary

```
#-----------------------------------------------------------------
# Name
#   MySummary
# Description
#   Summary function (based on summary function from base package)
#   with additional characteristic variables. Generates also a
#   'summary plot' with boxplot and central ranges, if 'var' is
#   numerical.
# Input
#   var    sample to analyze
#   x      optinal x variable for the plot
#   several other parameters are given to adjust the plot to
#   personal needs
```

```
# Output
#   list containing two data frames: 'coefficients' and 'ranges'
#   summary plot
#------------------------------------------------------------------
MySummary <- function(var, x=NULL, digits=2, plot=T,
    ctrl_ranges=c(0, 0.5, 0.9, 0.99), label_bool=NULL, label=NULL,
    xlab=NULL, ylab=NULL, title=NULL, get_nr=F, ylim=NULL,
    zero_hline=T, colour=NULL, xlab_ang=0, notch=F, facet_r=NULL,
    facet_c=NULL, boxplot_fill=TRUE, legend=TRUE)
{
  if(!(class(var) %in% c("numeric", "integer")))
    return(summary(var))

  df_sum <- as.data.frame(as.list(summary(var)))
  if(sum(is.na(var)) == 0) df_sum$NAs <- 0
  names(df_sum) <-
    c("min", "q_25", "median", "mean", "q_75", "max", "NAs")
  df_sum <- round(df_sum, digits)
  n <- length(var)
  df_sum$NAs_pct <- round(df_sum$NAs/n*100); df_sum$n <- n

  # specify ranges
  if(!is.null(ctrl_ranges)){
    df_range <- data.frame(lower = numeric(), upper = numeric())
    alpha <- 1-ctrl_ranges
    quant_l <- alpha/2; quant_u <- 1-alpha/2
    quant_names <- paste(ctrl_ranges*100, "%", sep="")
    quant_names[quant_names == "50%"] <- "50% (iqr)"
    quant_names[quant_names == "0%"] <- "Median"
    for(i in 1:length(ctrl_ranges)){
      df_range[quant_names[i], ] <-
        quantile(var, c(quant_l[i], quant_u[i]), na.rm=T)
    }
  }else{ df_range <- NULL }

  # get names of var and x
  name <- deparse(substitute(var))
  name_var <- GetVarName(name)
  name_var_return <- name_var # used for return list
  if(!is.null(ylab)) name_var <- ylab
  if(is.null(xlab)){
    if(is.null(x)){
      name_x <- "Index"
    }else{
      name <- deparse(substitute(x))
      name_x <- GetVarName(name)
    }
  }else{ name_x <- xlab }
  if(is.null(title))
    title <- paste("Summary of", sQuote(name_var))

  if(is.null(label_bool) | is.null(label))
    label_bool <- label <- rep(F, length(var))
```

```
  label_bool[is.na(label_bool)] <- F
  if(plot){
    if(!is.null(ctrl_ranges)){
      df_range$label = rownames(df_range)
      df_range$lty <- c(1, 1:(nrow(df_range)-1))[1:nrow(df_range)]
    }
    df_plot <- data.frame(var=var)
    df_plot$x <- if(is.null(x)){ 1:length(var) }else{ x }
    df_plot$label <- label; df_plot$label_bool <- label_bool
    df_plot$hjust <- Adjust(df_plot$x)
    if(!is.null(facet_r)) df_plot$facet_r <- facet_r
    if(!is.null(facet_c)) df_plot$facet_c <- facet_c
    if(!is.null(colour)) df_plot$colour <- colour
    df_miss_x <- df_plot[is.na(df_plot$x), ]
    df_miss_x[is.na(df_miss_x)] <- -Inf
    df_miss_var <- df_plot[is.na(df_plot$var), ]
    df_miss_var[is.na(df_miss_var)] <- -Inf
    p <- ggplot(data=df_plot, aes(x=x, y=var))
    hjust_xlab <- 0.5 + 0.5*sign(xlab_ang); vjust_xlab <- 0.5
    if(abs(xlab_ang) == 45) vjust_xlab <- 1
    p <- p +
      (if(class(x)=="factor" & boxplot_fill == TRUE){
        list(aes(fill=x), scale_fill_discrete(name=""))
      }else{ geom_blank() }) +
      geom_boxplot(outlier.size = 0, notch=notch) +
      (if(!is.null(colour)){ geom_point(aes(colour=colour))
        }else{ geom_point(alpha=0.5) }) +
      (if(zero_hline){
        geom_hline(yintercept=0, colour="gray50", alpha=0.5,
          show_guide=F)
      }else{ geom_blank() }) +
      (if(!is.null(ctrl_ranges)){
        list(geom_hline(data=df_range, size=1*base_size_/20,
          aes(yintercept=upper, linetype=label), show_guide=T),
        geom_hline(data=df_range, size=1*base_size_/20,
          aes(yintercept=lower, linetype=label), show_guide=T),
        scale_linetype_manual(name = "Central\nRanges",
          values = setNames(df_range$lty, c(df_range$label))))
      }else{ geom_blank() }) +
      # plot missing values (if any)
      (if(any(is.na(df_plot$x))){
        geom_rug(data=df_miss_x, side="l", colour="grey50",
          alpha=1)
      }else{ geom_blank() }) +
      (if(any(is.na(df_plot$var))){
        geom_rug(data=df_miss_var, side="b", colour="grey50",
          alpha=1)
      }else{ geom_blank() }) +
      (if(!is.null(facet_r)){
        if(is.null(facet_c)){ facet_grid(.~facet_r)
        }else{ facet_grid(facet_c~facet_r) }
      }else{
        if(!is.null(facet_c)){ facet_grid(facet_c~.)
```

```
        }else{ geom_blank() }
      }) +
      BaseTheme(base_size=base_size_) +
      theme(axis.text.x = element_text(angle = xlab_ang,
        hjust = hjust_xlab, vjust = vjust_xlab)) +
      labs(x = name_x, y = name_var) +
      (if(title %in% c("", NA, "none", FALSE))
        { geom_blank() }else{ labs(title = title) }) +
      theme(legend.key.width = unit(1*base_size_/14, "cm")) +
      (if(!is.null(ylim)){ ylim(ylim) }else{ geom_blank() }) +
      GGplotLabel(df_plot, label = df_plot$label,
        subset = df_plot$label_bool, hjust=df_plot$hjust,
        vjust=0.5) +
      (if(!is.null(colour) & !is.null(ctrl_ranges)){
        guides(colour=guide_legend(override.aes=
          list(linetype=0)))
      }else{ geom_blank() }) +
      (if(legend){ geom_blank() }else{ OmitLegend() })
    print(p)
    df_range$label <- NULL
  }
  cat("Summary of", sQuote(name_var), "\n\n")
  if(class(labeled <- label[label_bool]) == "factor")
    labeled <- as.character(labeled)
  if(get_nr) labeled <- GetNr(GetNr(as.character(labeled)))
  if(!is.null(df_range)) df_range <- round(df_range, digits)
  list(var=name_var_return,
    coefficients=df_sum,
    ranges=df_range,
    labeled=labeled)
}
#------------------------------------------------------------------
```

Other Functions

The subsequent functions are either small functions, or functions needed to run the functions presented above.

Adjust

```
#------------------------------------------------------------------
# Name
#   Adjust
# Description
#   Returns a horizontal adjustmend to use for lables of a
#   variable x.
# Input
#   x: variable, which labels should be horizontal adjusted
```

```
# Output
#   Horizontal adjustment
#-----------------------------------------------------------------
Adjust <- function(x){
  if(class(x) == "factor") x <- as.numeric(x)
  x[x == Inf] <- max(x[x < Inf], na.rm=T)
  ifelse(x > mean(range(x, na.rm=T)), 1.2, -0.2)
}
#-----------------------------------------------------------------
```

AggrMissings

```
#-----------------------------------------------------------------
# Name:
#   AggrMissings
# Description
#   Provides additional information to an "aggr"-object.
#   For a number of combination blocks (bottom up) the number of
#   observations, the percentage and cumulative percentage is
#   calculated.
# Input
#   aggr          an "aggr" object (generated with function aggr
#                 {VIM}
#   subset        only used, if 'prop' is not provided
#                 defines the number of combinations returned.
#   prop          All combinations are returned, such that the
#                 last combination has a cumulative percentage
#                 greater than 'prop'.
# Output
#   n             number of observations
#   missings      number and perdentage of missings for each
#                 variable
#   combinations  number and percentage of missings for each
#                 block (bottom up)
#-----------------------------------------------------------------
AggrMissings <- function(aggr, subset=1:10, prop)
{
  require(VIM)
  n <- sum(aggr$count)
  ct <- aggr$missings$Count
  ct_mis <- data.frame(count = ct,
    percent = round(100*ct/n, 2),
    row.names=aggr$missings$Variable)
  ct_mis <- ct_mis[order(-ct_mis$count), ]
  ct_comb <- data.frame(count = aggr$count,
    percent = round(aggr$percent, 2),
    cumsum_pct = round(aggr$percent, 2))
  ct_comb <- ct_comb[order(-ct_comb$count), ]
  ct_comb$cumsum_pct <- cumsum(ct_comb$cumsum_pct)
  rownames(ct_comb) <- 1:max(as.numeric(rownames(ct_comb)))
  if(!missing(prop)){
    if(prop < 1) prop = prop * 100
```

```
    subset <- !(ct_comb$cumsum_pct > prop)
    subset[subset == FALSE][1] <- T
  }
  list(n = n, missings = ct_mis,
    combinations = ct_comb[subset, ])
}
#-----------------------------------------------------------------
```

LmRegsubsets

```
#-----------------------------------------------------------------
# Name
#   LmRegsubsets
# Description
#   Returns the i-th model of a regsubsets model and optionaly
#   produces the standard residual plot (plot.lm).
# Input
#   regs    'regsubsets' or 'summary.regsubsets' object
#   i       the index of the model to return
#   data    underlying data
#   plot    produce the residual plot? (logical)
#   sum     print the model summary? (logical)
# Output
#   ret     desired regression model
#-----------------------------------------------------------------
LmRegsubsets <- function(regs, i, data, plot=F, sum=F){
  if(class(regs) == "regsubsets") regs <- summary(regs)
  predictors <- paste(names(coef(regs$obj, i))[-1],
    sep="", collapse=" + ")
  ret <- lm(as.formula(paste("y ~ ", predictors)), data=data)
  if(plot){ par(mfrow=c(2,2)); plot(ret); par(mfrow=c(1,1)) }
  if(sum) print(summary(ret))
  return(ret)
}
```

NaOmit

```
#-----------------------------------------------------------------
# Name
#   NaOmit
# Description
#   Omits all rows of the inout data frame and prints
#   information on the number of omited rows.
# Input
#   data  data frame to omit rows with NA values
#   ret   should the result be returned? (logical), usefull if
#         only information about number of missings is wanted
# Output
#  data   data frame with omitted NA rows
#-----------------------------------------------------------------
```

```
NaOmit <- function(data, ret=T){
  nrow_orig <- nrow(data)
  data <- na.omit(data)
  nrow_omit <- nrow(data)
  info <- data.frame(nrow = c(nrow_orig, nrow_omit,
    nrow_orig-nrow_omit),
    row.names = c("  original", "NA omitted", "difference"))
  info$pct <- round(info$nrow/nrow_orig*100)
  print(info)
  if(ret) data
}
#-------------------------------------------------------------------
```

Bibliography

[1] AMS AG. Corporate Information. http://www.ams.com, October 2013. 1

[2] Belsley, D. A., Kuh, E., and Welsch, R. E. *Regression Diagnostics*. Wiley, 1980. 3.16, 3.3.2

[3] Cleveland, W. S. LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *The American Statistician 35*, 1 (1981). 2.2.3, 3.16

[4] Cook, R., and Weisberg, S. *Regression Diagnostics*. Chapman and Hall, 1982. 3.16, 3.3.2

[5] Crawley, M. J. *The R Book*, 1st ed. Wiley, 2007. B

[6] Fahrmeir, L., Kneib, T., and Lang, S. *Regression*, 2nd ed. Statistik und ihre Anwendungen. Springer, 2009. 3, 3.2.1, 3.2.2, 3.2.3, 3.10, 3.12, 3.2.3, 3.3, 3.3.1, 3.3.2, 3.4.2, A.1

[7] Faria, J. C., Grosjean, P., and Jelihovschi, E. *Tinn-R - GUI/Editor for R Language and Environment Statistical Computing*, 2008. version 2.3.7.1, http://sourceforge.net/projects/tinn-r. B

[8] Friedl, H. *Mathematische Statistik*. Institute of Statistics, Graz University of Technology, 2010. Lecture Notes. A

[9] Friedl, H. *Generalisierte Lineare Modelle*. Institute of Statistics, Graz University of Technology, 2011. Lecture Notes. 2.4.1, 3.17

[10] Friedl, H. *Regressionsanalyse*. Institute of Statistics, Graz University of Technology, 2012. Lecture Notes. 2.2.3, 2.4.1, 3, 3.3.2, 3.3.2, 3.21, 2, 3.4.2, 3.4.2, 3.4.2

[11] International Monetary Fund. World Economic Outlook Database. Tech. rep., October 2012. http://www.imf.org. 2.6.1

[12] Johnson, R. A., and Wichern, D. W. *Applied Multivariate Statistical Analysis*, 6th ed. Pearson, 2007. 3.5, 3.5.1, 38, 3.23, 3.5.2, 39, 3.5.2

[13] Jollife, I. T. *Principal Component Analysis*. Springer Series in Statistics. Springer, 1986. 3.5, 3.22, 3.23, 3.5.2

[14] Kleinbaum, D. G., Kupper, L. L., Muller, K. E., and Nizam, A. *Applied Regression Analysis and Multivariable Methods*, 3rd ed. Duxbury Press, 1998. 16, 3, 3.7, 3.2.3, 3.2.3, 3.2.3, 3.3, 3.3.1, 3.4.1, 5, 3.4.1, 3.4.2

[15] Lumley, T. (using Fortran code by Alan Miller). *Leaps: Regression Subset Selection*, 2009. **R** package version 2.9. http://CRAN.R-project.org/package=leaps. 2.2.2, 2.4.1, 3.4.1, B

[16] Mallows, C. L. Some Comments on C_p. *Technometrics 15*, 4 (1973), pp. 661–675. 3.4.2

[17] Murdoch, D., and Chow, E. D. *ellipse: Functions for drawing ellipses and ellipse-like confidence regions*, 2012. **R** package version 0.3-7, http://CRAN.R-project.org/package=ellipse. 2.2.3, B

[18] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. version 2.15.2, http://www.R-project.org. B

[19] Sachs, L., and Hedderich, J. *Angewandte Statistik. Methodensammlung mit R*. Springer, 2006. 3, 3.5, 3.3.1, A

[20] Schwarz, G. Estimating the Dimension of a Model. *The Annals of Statistics 6*, 2 (1978), 461–464. 3.4.2

[21] Sponer, B. Modellierung von Projektdaten mittels multipler Regression. Master's thesis, Graz University of Technology, 2009. 1, 1, 2.3.2, 2.7, 2.18, 2.49

[22] Stadlober, E. *Statistik*. Institute of Statistics, Graz University of Technology, 2008. Lecture Notes. 5, A

[23] Stadlober, E. *Angewandte Statistik*. Institute of Statistics, Graz University of Technology, 2011. Lecture Notes. 2.1.1, 2.2, 2.4.1, 2.6.2, 3.10, 3.3.1, 3.3.1

[24] Tableman, M., and Kim, J. S. *Survival Analysis Using S: Analysis of Time-to-Event Data*. Texts in Statistical Science. Chapman & Hall/CRC, 2004. 4

[25] Templ, M., Alfons, A., and Filzmoser, P. Exploring the multivariate structure of missing values using the r package vim. In *Abstracts of the 5th R useR Conference* (2009). see also the associated presentation. 2.2.4

[26] Templ, M., Alfons, A., Kowarik, A., and Prantner, B. *VIM: Visualization and Imputation of Missing Values*, 2012. **R** package version 3.0.2, http://CRAN.R-project.org/package=VIM. 2.2.4, B

[27] Templ, M., and Filzmoser, P. *Visualization of Missing Values using the R-Package VIM*. Department of Statistics and Probability Theory, Vienna University of Technology, 2008. Research Report. 2.2.4

[28] Todorov, V., and Templ, M. R in the statistical office: Part ii. Working, United Nations Industrial Development Organization, 2012. 2.1.1

[29] Venables, W. N., and Ripley, B. D. *Modern Applied Statistics with S*, 4th ed. Springer, New York, 2002. **R** package version 7.3-22, http://www.stats.ox.ac.uk/pub/MASS4. B

[30] Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. Springer New York, 2009. **R** package version 0.9.2.1, http://had.co.nz/ggplot2/book. B, C

[31] Wickham, H. The Split-Apply-Combine Strategy for Data Analysis. *Journal of Statistical Software 40*, 1 (2011), 1–29. plyr **R** package version 1.7.1, http://www.jstatsoft.org/v40/i01/. B

[32] Wilkinson, L., and Wills, G. *The Grammar of Graphics*. Statistics and Computing. Springer, 2005. 4

Index