Development of an AI Assistant to Learn Linear Regression Diagnostics

Bachelor's Thesis

Course of Study: Biomathematics
Hochschule Koblenz - University of Applied Sciences
RheinAhrCampus, Remagen

Presented by **Philipp Leubner**

Matriculation Number: 554854
Born on April 23, 2002, in Neuwied

First Examiner:	Dr. Denise Welsch
Second Examiner:	Dr. Daniel Friemert
Conducted at:	Hochschule Koblenz, RheinAhrCampus

Neuwied, September 20, 2024

Preliminaries Related to the Thesis

Usage of Quotations

At the beginning, it is explained how quotations are used in this thesis, and examples are provided. If no source is cited after sentences or paragraphs, no source was used.

Direct Quotations

Direct quotes are verbatim copies from another reference. These quotes will be enclosed in quotation marks and followed by the original source, for example: "Direct quotes can look like this" [Source 0]. In this thesis, sources will be cited within square brackets. No special font is applied.

Indirect Quotations

Quotes that are not copied verbatim from an other reference but instead are paraphrased with own words are considered as indirect quotes [Source 1].

If a source is cited at the end of a paragraph, it refers to the entire preceding paragraph. A new paragraph begins if there is a new heading or indentation. A formula, table or figure does not count as a new paragraph as long as there is no indentation after the formula. If only short passages within a paragraph refer to a source, the source will be cited directly after the sentence [Source 2]. This explanation of the use of quotations is an example of indirect citation [Source 3].

Usage of Abbreviations

When an abbreviation is used for the first time, the term is written out in full length. The abbreviation follows in parentheses. After that, only the abbrevation is used. It looks like this: This is the usage of abbreviations (abbr.). After the first time, only abbr. will be used.

Usage of Technical Terms

The first time that a technical term important for this section is mentioned, it is printed in *cursive*. After that, every following occurrance will be printed using the default font. Programming languages, packages, datasets, variables, code or prompts will be printed using `typewriter font`.

Usage of Artificial Intelligence (AI)

In this thesis, AI-generated output from ChatGPT was examined, which is why the thesis also contains AI-generated output such as text or images. In addition to this use of AI, ChatGPT was also used as a translation and grammar aid.

Abstract

In the learning process, it is common to rely on human assistance. However, having a learning method that can be used independently, especially when human help isn't available due to time limits, is also valuable. Multiple linear regression is a fundamental concept in machine learning, so there is a need for a method that enables learners to understand this topic on their own. To address this, the AI assistant OLSAI is being developed, which is based on ChatGPT and aims to provide learners with the necessary output to learn the topic. The AI assistant delivers the interpretation in natural language of the regression diagnostics. This output is grounded in a dataset, as learners often struggle with applying methods to real-world data. To develop the assistant, an initial analysis was conducted to determine what items should be included. A simple prompt was then created to generate output from ChatGPT, which was checked for statistical accuracy. While the output was statistically correct, it was incomplete and not very learner-friendly. Through prompt engineering, various prompts were optimized to produce a controlled output that is both statistically complete and learner-friendly. These optimized prompts were then used to develop the OLSAI tool in `Python`, using OpenAI's Application Programming Interface (API). The AI tool now provides a complete output for learners based on a real dataset. However, further development is needed before the tool can be made available to learners. This thesis serves as the foundation for the continued development of the OLSAI AI tool.

Contents

List of Figures V

List of Tables VII

Abbrevations VIII

1. Narratives and Objectives 1

1.1. Narrative and Learning Scope 1

1.2. OLSAI - Ordinary Least Squares Artificial Intelligence 2

1.3. Objectives of this Thesis 2

2. The Datasets 3

2.1. `Cacao.csv` 3

2.2. `Electricity1955.csv` 4

3. Results 6

3.1. Own Regression Diagnostics 6

3.1.1. Structure of the Analysis 6

3.1.2. Explanatory Data Analysis of the Dataset `cacao.csv` 7

3.1.3. Regression Diagnostics of the Dataset `cacao.csv` 13

3.2. Quality Analysis: ChatGPT Before 23

3.2.1. ChatGPT Before: Explanatory Data Analysis 24

3.2.2. ChatGPT Before: Model Assumptions, Regression Diagnostics 24

3.2.3. ChatGPT Before: Structure of the Analysis 25

3.2.4. ChatGPT Before: Summary of the Quality Analysis 26

3.3. Quality Analysis: ChatGPT After 27

3.3.1. Approaches for Prompting 27

3.3.2. Utilized Prompts 28

3.3.3. Comparison to Own Analysis 32

3.3.4. Evaluation of Learner-Friendliness 34

3.3.5. Perspective of a Learner 39

3.4. Reproduction with other Datasets . . . 41
3.5. The OLSAI Assistant in `Python` . . . 44
3.5.1. The Description of the Code . . . 44
3.5.2. The Key Elements of the Assistants API . . . 45
3.5.3. The Process of Progamming . . . 46
3.5.4. The HTML Output of the Code . . . 49

4. Technical Background **53**
4.1. What is ChatGPT? . . . 53
4.2. The Capabilities of ChatGPT . . . 53
4.2.1. Overview of Capabilities . . . 54
4.2.2. What is Advanced Data Analysis? . . . 54
4.3. What is Prompt Engineering? . . . 55
4.4. What is an Application Programming Interface? . . . 55
4.5. What is GitHub? . . . 56

5. The Learning Content **57**
5.1. What is Multiple Linear Regression? . . . 57
5.1.1. Assumptions of the Model . . . 58
5.2. What are Regression Diagnostics? . . . 59
5.2.1. Outliers . . . 60
5.2.2. High-Leverage Points . . . 60
5.2.3. Non-Linearity . . . 61
5.2.4. Heteroscedasticity . . . 62
5.2.5. Correlation of Error Terms . . . 63
5.2.6. Normality of Residuals . . . 64
5.2.7. Collinearity of Predictors . . . 64

6. Conclusion **66**
6.1. Summary of the Thesis . . . 66
6.2. Limitations and Issues . . . 67
6.3. Outlook . . . 68

Bibliography **69**

Attachment **73**

A. Additional Tables **73**

B. Additional Graphics **74**

List of Figures

3.1. Flow Chart of the Structure of the Analysis 7
3.2. Histogram of `stem_diameter` 9
3.3. Boxplot of `stem_diameter` 10
3.4. ECDF Plot of `stem_diameter` 11
3.5. Q-Q Plot of `stem_diameter` 12
3.6. Scatterplot of Studentized Residuals 14
3.7. Leverage-Plot 15
3.8. Cook's Distance 16
3.9. Residuals Plot 17
3.10. Scale-Location Plot 18
3.11. Studentized Residuals over Time 19
3.12. Q-Q Plot of Residuals 20
3.13. Correlation Matrix 21
3.14. Field of Study of Participants 34
3.15. Average Rating for Understanding by the Learner 35
3.16. Median of Rating for Understanding by the Learner 36
3.17. Average of Rating for Regression Diagnostics by the Learner 37
3.18. Median of Rating for Regression Diagnostics by the Learner 37
3.19. Answers of Normality of Residuals 38
3.20. Answers of Heteroscedasticity 38
3.21. Outlier Detection of `cacao.csv` Before Outlier Removal 42
3.22. Outlier Detection of `cacao.csv` After Outlier Removal 42
3.23. Residuals vs. Predicted Values of `Electricity1955.csv` by ChatGPT 43
3.24. Objects of Assistants [20] 45
3.25. Screenshot of Descriptive Statistics in the Output 50
3.26. Screenshot of Model Representation in the Output 51
3.27. Screenshot of Plot in the Output 52

5.1. Acceptance and Rejection Regions of the Durbin-Watson Test [9]. . . . 63

B.1. The GitHub Repository Statsomat/OLSAI 74
B.2. The Folder `Code` 75
B.3. The Folder `python_code` 76
B.4. The Folder `Datasets` 77
B.5. The Issues Tab 77
B.6. Histograms of Each Variable 78
B.7. Boxplots for Each Variable 79
B.8. ECDF Plots for Each Variable 80
B.9. QQ Plots for Each Variable 81
B.10. Questionnaire (1) 82
B.11. Questionnaire (2) 83
B.12. Questionnaire (3) 84
B.13. Questionnaire (5) 85
B.14. Questionnaire (6) 86

List of Tables

2.1. Variables of the Dataset `cacao.csv` 3
2.2. Variables of the Dataset `Electricity1955.csv` 4

3.1. Descriptive Statistics of the Dataset `cacao.csv` 8
3.2. High-Leverages of Observations of the Dataset `cacao.csv` 15
3.3. Cook's Distance of Observations of the Dataset `cacao.csv` 16
3.4. Variance Inflation Factor 22
3.5. Criteria of Quality Analysis 24
3.6. Comparison of Own Analysis and ChatGPT's Analysis 32

5.1. Summary of Diagnostic Tools 59

A.1. Diagnostic Tools 73

Abbrevations

Q-Q quantile-quantile 7

ECDF empirical cumulative distribution function 7

EDA explanatory data analysis 6

VIF variance inflation factor 20

ML machine learning 1

AI Artificial Intelligence 3

API Application Programming Interface I

ADA Advanced Data Analysis 2

LLM Large Language Model 53

1. Narratives and Objectives

1.1. Narrative and Learning Scope

Due to digitalization, machine learning (ML) and AI are playing an increasingly significant role [22]. Those fields bring many new opportunities in areas such as language processing, image analysis and even medical diagnostics. Therefore, there is a great demand for professionals in fields like Data Science, Big Data and Advanced Analytics, which cannot be met in Germany [8]. It is important to support the path to become a professional in this field. A student, working professional, or interested individual who wants to learn more about the data will be referenced as a learner. Learners usually start by acquiring the mathematical foundations of calculus, linear algebra and statistics to understand the processes of ML. Linear regression is often one of the foundational models taught in statistics and ML. It serves as the basis for more complex regression models, such as multiple linear regression. These models aim to model the relationships between dependent and independent variables. To ensure the accuracy, validity and reliability of these models, it is crucial to verify several underlying assumptions. Failing to meet these assumptions can lead to biased estimates and incorrect conclusions. Therefore, proper diagnostic checks and validations are essential in the modeling process using so-called *regression diagnostics*.

To learn the regression diagnostics of (multiple) linear regression, various resources can be used. Topic-specific books or online videos can be utilized. Additionally, experts can teach the topic to learners. However, books are often expensive, it is difficult to ask questions using videos and experts are usually not very flexible with their time. Nowadays, there are many AI-powered chatbots that can answer questions and explain topics. One of the most well-known chatbots is ChatGPT from OpenAI [15]. The chatbot is cost-effective, can be used at any time and is therefore an ideal alternative to books, videos, or consulting experts. Moreover, the chatbot can also respond to the learner's questions. Now the following scenario will be looked at. The learner wants to use ChatGPT to learn more about the regression diagnostics of (multiple) linear regression. A common problem for learners is applying the theory to a real dataset. For this reason, the learner uploads a dataset to get an explanation of regression diagnostics, using ChatGPT's Advanced Data Analysis (ADA) tool. However, the current chatbots are not perfect, as they provide objectively incorrect results, which is also called *hallucinations*. Therefore, the output must be controlled through specific inputs to prevent errors from occurring.

1.2. OLSAI - Ordinary Least Squares Artificial Intelligence

OLSAI is the name for an AI application to be built for learning a core data science method called multiple linear regression, which relies on the analysis of the ordinary least squares (OLS) in a dataset. Similarly to a human tutor, OLSAI looks at the dataset of the learner, analyses it and responds by offering a set of personalised instructions. The learner is encouraged to apply the instructions for his own dataset. The instructions are partly generated by AI models. OLSAI aims to replace non-available academic guidance and generate real-time support [30].

1.3. Objectives of this Thesis

This thesis aims to develop the first verison of the OLSAI assistant, a tool designed to assist learners with regression diagnostics. To achieve this goal, several key steps must be taken. The OLSAI assistant is built upon ChatGPT using ChatGPT's API, so it's crucial to ensure that the generated outputs are both statistically accurate and learner-friendly. Due to the hallucinations of the chatbot, the objective is to develop a controlled output from the AI to guarantee statistically accurate results. The process begins by introducing and explaining the datasets that will be used for analysis. Afterwards, an independent analysis of the `cacao.csv` dataset will be performed by myself, providing a solid foundation for the outputs generated by ChatGPT. This analysis will contain the most important statistical methods. Next, an optimized prompt will be used to generate outputs that resemble the independent analysis. These outputs will be evaluated by potential learners and improvements will be made based on their feedback. The objective is to produce outputs that effectively help learners grasp the concepts of regression diagnostics. Finally, once this groundwork is completed, the OLSAI assistant will be programmed using ChatGPT's API. This will ensure that the assistant meets the needs of learners and provides them with accurate and helpful guidance.

2. The Datasets

2.1. `Cacao.csv`

The dataset `Harvest_fruit_arthropod_count_cacao2011-2012.xlsx` contains data collected from an experiment in smallholder cacao plantations in Indonesia. The experiment involved excluding ants, birds and bats from the plantations to investigate their influence. The dataset `cacao.csv` is a subset of the full dataset, focusing only on the data related to ant exclusion. Both datasets include information on the growth, yield, and reproductive status of the cacao trees.

Variable	**Unit**
`ant_exlcusion`	-
`stem_diameter`	cm
`height`	cm
`canopy`	-
`dw_health`	cm
`dw_infect`	cm
`dw_total`	cm
`fw_pulb`	g
`fw_seeds`	g
`fw_total`	g
`ab_fl_op`	-
`ab_fl_cl`	-
`ab_fl`	-

Table 2.1.: Variables of the Dataset `cacao.csv`

The 13 different variables are listed in Table 2.1. The variable `ant_exlcusion` is a binary indicator that shows whether ants were excluded and removed from the cacao tree, with 1 indicating exclusion and 0 indicating no exclusion. The characteristics of the cacao tree are described by its height (`height`) and average stem diameter (`stem_diameter`). The canopy cover, derived from hemispherical photography, is represented by the variable canopy. The harvested yield is divided into three variables: `dw_health` represents the yield unaffected by pests and diseases, while dw_infect represents the damaged yield. The total yield, including both healthy and damaged parts, is shown as `fw_total`. The fresh weight of the fruit pulp is recorded as `fw_pulb`, and the fresh weight of the cacao beans is recorded as `fw_seeds`. The combined weight of both the beans and pulp is also included in `fw_total`. The variable `ab_fl_op` represents the total number of open flowers, `ab_fl_cl` counts the flower buds, and `ab_fl` captures the overall number of flowers, combining both open flowers and buds, across all trees in a treatment [12].

2.2. Electricity1955.csv

The dataset `Electricity1955.csv` describes the cost function data for 159 US electricity producers in 1955 and provides an overview of the cost structure in electricity generation.

Variable	Unit
cost	USD
output	mKWH
labor	-
laborshare	-
capital	-
capitalshare	-
fuel	USD
fuelshare	-

Table 2.2.: Variables of the Dataset `Electricity1955.csv`

The total cost of production is captured by the `cost` variable, while `output` refers to the total electricity generated. The variable `labor` indicates the wage rate and `laborshare` shows the proportion of costs attributed to labor. The variable `capital` represents the price index for capital goods, with `capitalshare` indicating the share of costs due to capital. Finally, `fuel` refers to the price of fuel used and `fuelshare` reflects the portion of the total cost that is due to fuel expenses. Together, these variables outline the financial dynamics of electricity generation in the year 1955 [13].

3. Results

3.1. Own Regression Diagnostics

In my own analysis, the focus will be on the most important aspects that should be included in the AI output. This analysis is intended to be the basis upon which the AI output should be oriented. In addition to a learner-friendly structure, an explanatory data analysis (EDA) and regression diagnostics will be conducted. The AI output will provide further explanations of the methods used. For statistical modeling and tests, the `Python` module `statsmodels` was used [23]. The analysis itself is also used to verify the statistical accuracy of the output, as it can be compared with the output.

3.1.1. Structure of the Analysis

In order to create the OLSAI assistant in `Python`, a basis for the analysis and the report is needed. Various elements need to be defined, such as the structure, the methods used or visual representations. It must be defined how the output should look like. Without a certain structure of the report, the analysis becomes confusing and would be of little help to the learner. This section presents my own analysis of the `cacao.csv` dataset described in Chapter 2, which will form the basis for the aimed output of ChatGPT and will be used to evaluate the quality of that output.

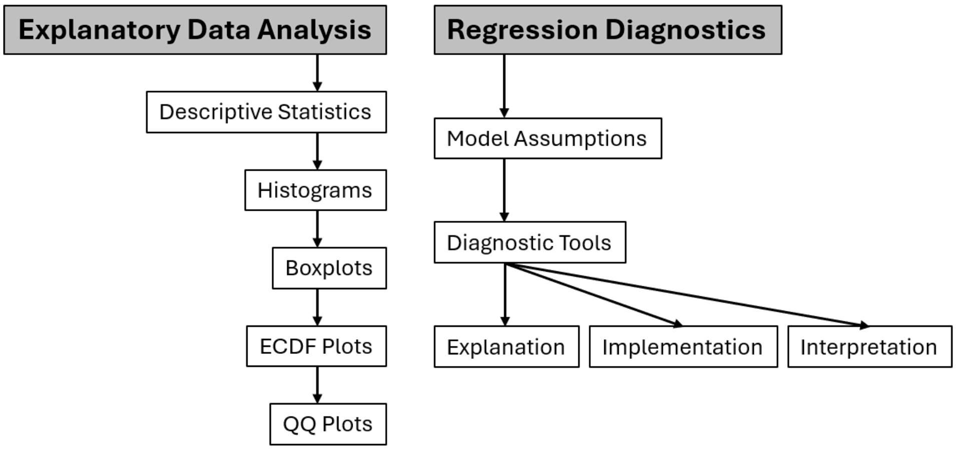


Figure 3.1.: Flow Chart of the Structure of the Analysis

The analysis will be split into two parts. The first part contains a brief *EDA* including *descriptive statistics*, *histograms*, *boxplots*, *empirical cumulative distribution function (ECDF)* plots and *quantile-quantile (Q-Q)* plots [25]. Exemplary tables and graphs are shown for illustration purposes, using the variable `stem_diameter` as an example. Not all variables are included due to clarity. Plots for every other variable can be found in the attachment Figures B.6 to B.9. The second part consists of the actual regression diagnostics. First, the model should be constructed and its assumptions should be briefly explained. Without this knowledge, the learner will struggle to understand the regression diagnostics. Next, each used diagnostic tool should be briefly described to clarify its purpose. Finally, each regression diagnostic should be explained, implemented and interpreted to demonstrate the practical application of the theory. My own analysis serves as a proof of concept for the AI assistant and includes the enumeration of used methods, their results and their interpretation. The AI assistant will additionally explain the methods used in more detail, present (model) equations and clarify model assumptions, as outlined in Chapter 5. My own analysis can be found in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `Code` (see Figure B.2) as `CacaoOwnAnalysis.ipynb`. The code for the analyses and the AI assistant is stored in the GitHub repository (Statsomat/OLSAI), making it accessible to the public [14].

3.1.2. Explanatory Data Analysis of the Dataset `cacao.csv`

The analysis should start with a short EDA, allowing the learner to become familiar with the variables in the dataset. Short summaries of the sample and measurements are provided to describe the main characteristics of the dataset using descriptive statistics, which include

- count: number of observations for each variable
- mean: average value of each variable
- standard deviation: variation of the values
- minimum, maximum: smallest and largest value in the dataset for each variable
- 25%-quartile: value below which 25% of the data fall
- 50%-quartile (median): value below which 50% of the data fall
- 75%-quartile: value below which 75% of the data fall

These *summary statistics* are calculated for each variable in the dataset and help to understand the spread and distribution of the data by displaying measures of central tendency and variability.

Summary Statistics	`stem_diameter`	`height`	`canopy`
Count	120	120	120
Mean	27.09	293	0.33
Standard Deviation	5.31	34.61	0.17
Minimum	15.74	223.75	0.02
25%-Quartile	23.38	270.38	0.18
50%-Quartile	26.71	287.88	0.34
75%-Quartile	30.37	313.19	0.45
Maximum	46.6	399.5	0.71

Table 3.1.: Descriptive Statistics of the Dataset `cacao.csv`

Table 3.1 shows exemplary descriptive statistics for the variables `stem_diameter`, `height` and `canopy`. These values can be represented in various plots. The plots used in the analysis are histograms, boxplots, ECDF plots and Q-Q plots in that specific order to keep the report organized.

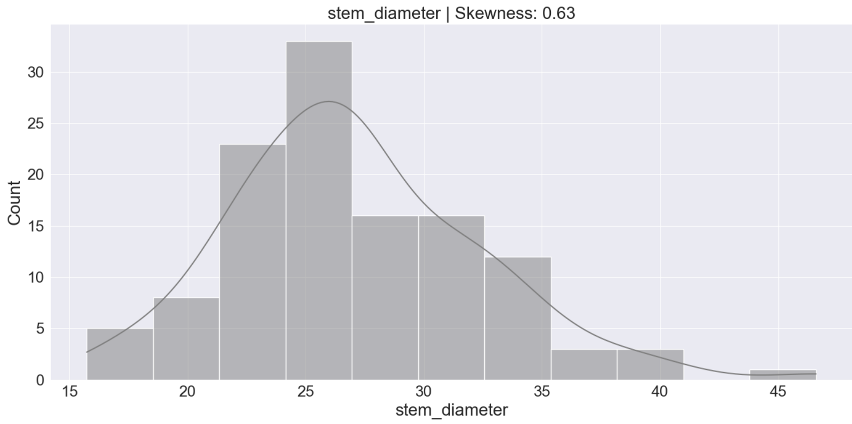


Figure 3.2.: Histogram of `stem_diameter`

Figure 3.2 presents a histogram for the variable `stem_diameter`. The x-axis displays the intervals into which the data is divided, with each bin covering a specific range of values. These intervals range from 15 to 45, which are approximately the minimum and maximum values of `stem_diameter`, as referenced in Table 3.1. The y-axis shows the count of observations that fall within each bin, with values ranging from 0 to 30. A gray line is overlaid on the histogram, representing a smoothed curve to help visualize the distribution of the data. At the top of the histogram, the variable name is shown along with its corresponding skewness value. Skewness describes the asymmetry of a distribution around its mean: a positive skewness indicates a longer tail on the right side, a negative skewness indicates a longer tail on the left and a skewness of zero indicates a symmetrical distribution. In this case, the distribution of `stem_diameter` has a skewness of 0.63, indicating a slight positive skew.

In this analysis, the histograms are followed by boxplots, which summarize the dataset in five different points and visualize the distribution of the data, similar to histograms.

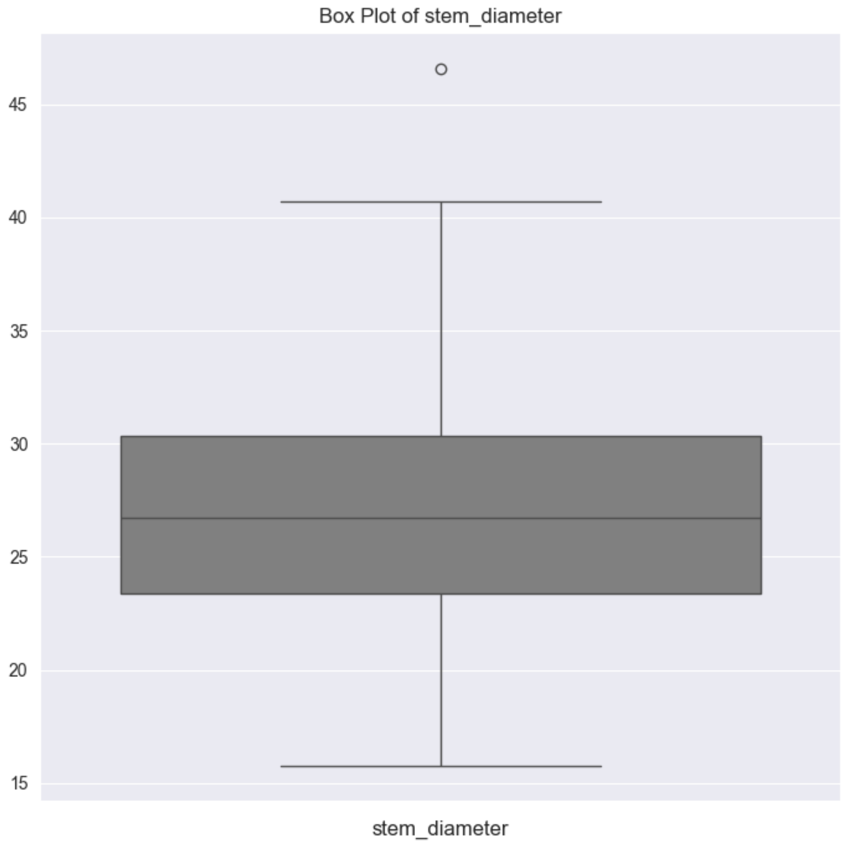


Figure 3.3.: Boxplot of `stem_diameter`

Boxplots display the median as a central line within the box, while the box itself represents the interquartile range (IQR), which spans from the 25% quartile to the 75% quartile. The exact values for these quartiles can be found in Table 3.1. In this case, the median is 27.09 and the box extends from 23.38 to 30.37. The whiskers of the boxplot stretch from the edges of the box to the smallest and largest values within 1.5 times the IQR. However, the whiskers stop if there are no data points beyond this range. Any data points outside of this range are considered outliers and are represented as dots.

After boxplots, ECDF plots are presented, visualizing the distribution of the dataset and the accumulation of observations over their range.

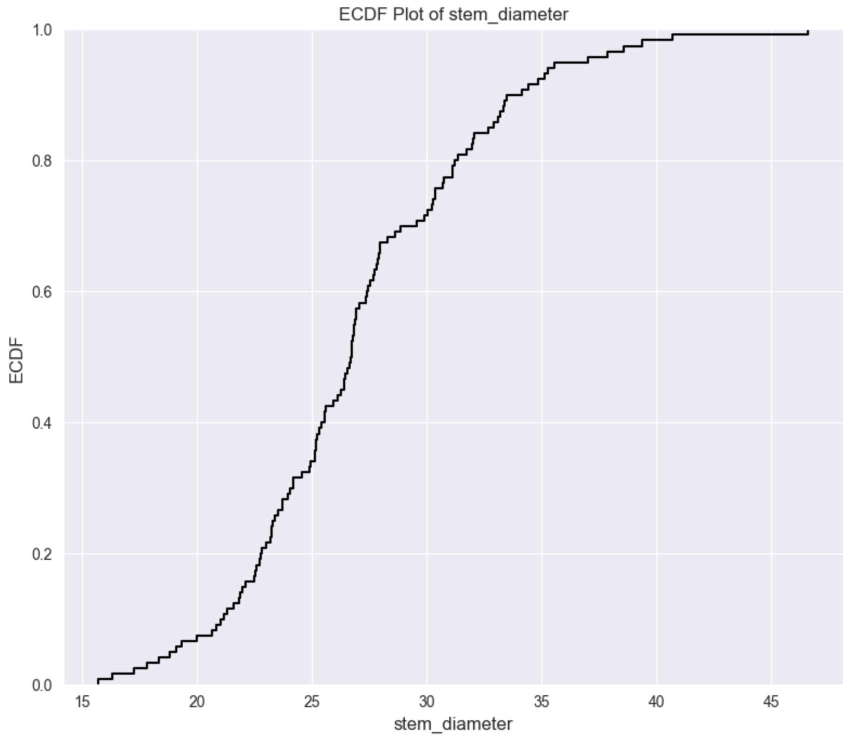


Figure 3.4.: ECDF Plot of `stem_diameter`

Figure 3.4 displays the ECDF plot for the variable `stem_diameter`. The x-axis represents the values of `stem_diameter`, ranging from approximately 15 to 45, which are the variable's minimum and maximum values. The y-axis indicates the cumulative proportion of observations, ranging from 0 to 1. In the plot, each step corresponds to an observation in the dataset. The height of each step on the y-axis reflects the proportion of data points that are less than or equal to the corresponding value on the x-axis. A large rise in the plot indicates a large number of observations within a particular range, while a smooth, gradual increase suggests a more uniform distribution of values. For the variable `stem_diameter`, the plot shows a relatively uniform distribution, as indicated by the steady increase without sudden jumps or flat sections. The median of the distribution can be identified at a y-value of 0.5. Additionally, the symmetry of the plot around the median suggests only a slight skewness, which aligns with the skewness value of 0.63 observed in Figure 3.2. This indicates a relatively balanced distribution of the data.

The last plot of the EDA should be the Q-Q plot, which compares the distribution of a dataset with a theoretical distribution. Here, the normal distribution is used, but any theoretical distribution can be applied.

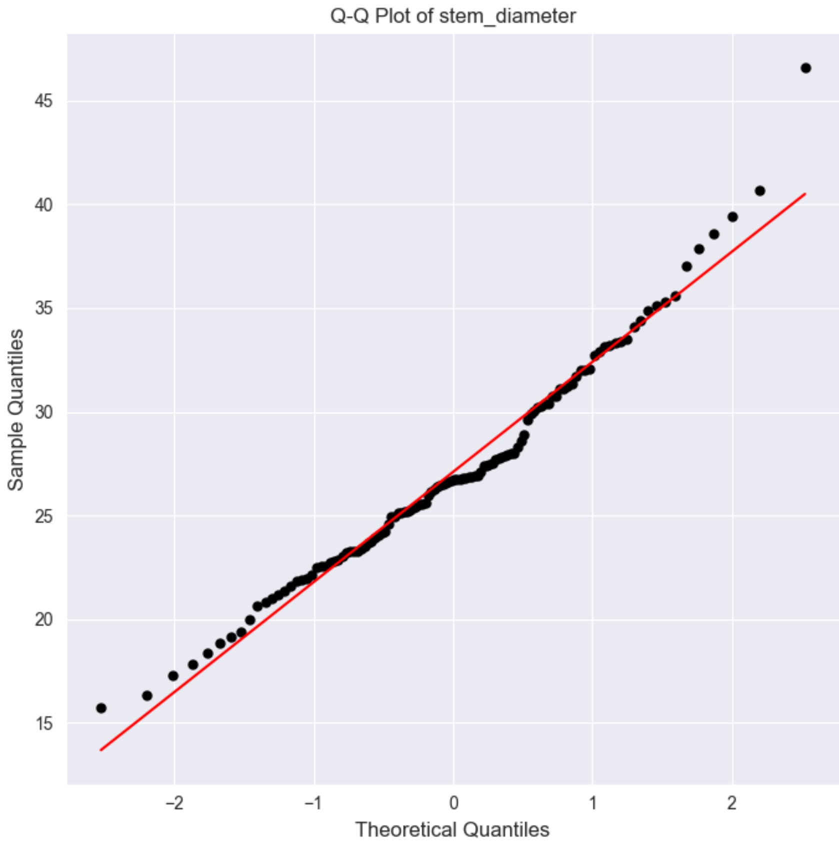


Figure 3.5.: Q-Q Plot of `stem_diameter`

The x-axis represents the quantiles of a theoretical distribution, while the y-axis represents the quantiles of the observed data for `stem_diameter`. The bisecting red line indicates where the points would lie if the data perfectly followed a normal distribution. Most of the data points are close to this red line, suggesting that `stem_diameter` approximately follows a normal distribution. However, there are deviations in the left and right tails, indicating some differences from the normal distribution. Despite these deviations, the normal distribution serves as a reasonable approximation for this variable. Overall, Q-Q plots are useful for assessing how closely a variable's distribution aligns with a theoretical distribution. The histograms, boxplots, ECDF plots and Q-Q plots for every other variable can be seen in Figure B.6 - B.9 in the attachment.

3.1.3. Regression Diagnostics of the Dataset `cacao.csv`

The goal of this analysis is to help the learner become familiar with regression diagnostics, so they can understand the different options available and how to interpret them. Ideally, the learner will be able to apply these techniques to other datasets. The preceeding plots help to visualize the distribution of variables, giving a general overview of the dataset. Before examining regression diagnostics, it is important to first build the model so that these diagnostics can be analyzed. The priority is not to find the best model for the dataset, but to understand the regression diagnostics. Therefore, `stem_diameter` was used as the dependent variable for the model, with all remaining variables as independent variables. The remaining variables include `ant_exclusion`, `height`, `canopy`, `dw_healthy`, `dw_infect`, `dw_total`, `fw_pulb`, `fw_seeds`, `fw_total`, `ab_fl_op`, `ab_fl_cl` and `ab_fl`. The model only considers independent variables, without including any interactions. After the EDA, the assumptions explained in Chapter 5.1 are checked using the regression diagnostics described in Chapter 5.2 to assess the validity of the model.

Outliers

Outliers are data points that significantly deviate from the predictions made by a model. One effective method for detecting outliers is through the use of *studentized residuals* [9]. These residuals are calculated by adjusting the original residuals with an estimate of their standard deviation. A common rule of thumb is to consider values beyond $\pm$ 3 as outliers. This means that any studentized residual less than -3 or greater than 3 is typically flagged as an outlier. After calculating the studentized residuals, the value of observation 27 in order of appearance of the dataset is 3.15, which is larger than 3 [26].

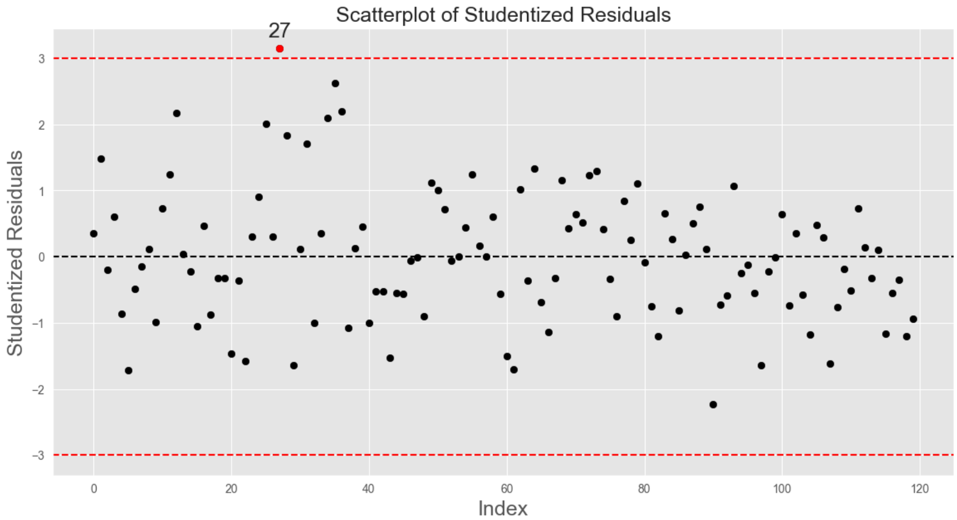


Figure 3.6.: Scatterplot of Studentized Residuals

Figure 3.6 illustrates the studentized residuals plotted against the indices of the observations. The x-axis represents the indices, ranging from 0 to approximately 120, matching the count shown in Table 3.1. The y-axis displays the studentized residuals, which range from around -3 to 3. A black dashed line indicates the zero-line, while red dashed lines mark the cutoff thresholds at $\pm$ 3. Any points outside this range are highlighted in red, indicating outliers. In this dataset, observation 27 is identified as an outlier.

High-Leverage Points

High-leverage points are observations that have a high influence on the impact regression coefficients and therefore on the model. To identify observations that have a high influence on the regression model, both *leverage* and *Cook's distance* are calculated and shown in a plot. Leverage indicates how much an observation influences the fitted values [9]. Cook's distance measures the overall impact of an observation on the regression model [7]. The threshold for high-leverage points is calculated as $\frac{2p}{n}$, which equals 0.217 for this model [9]. Table 3.2 shows only the indices and leverages of the observations whose leverages surpass this threshold.

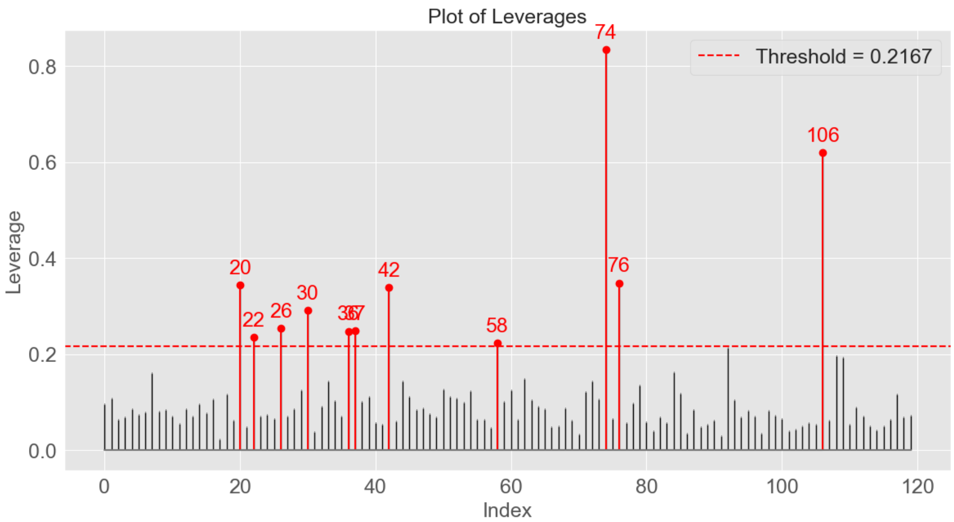


Figure 3.7.: Leverage-Plot

Index	20	22	26	30	36	37	42	58	74	76	106
Leverage	0.34	0.24	0.25	0.29	0.25	0.25	0.34	0.22	0.83	0.35	0.62

Table 3.2.: High-Leverages of Observations of the Dataset `cacao.csv`

Figure 3.7 shows the leverages plotted against the indeces of the corresponding observations. The x-axis shows the indeces of the observations, while the y-axis represents the value for the leverage of those observations. In the plot, the red dashed line indicates the threshold of 0.2167. Therefore, values above that line are considered as high-leverage points. These points are portrayed as a red line with a red dot at the end, while black lines indicate that the corresponding observation is not crossing the cut-off value.

The threshold for Cook's Distance is calculated as $\frac{4}{n}$, which equals 0.03 for this model [9]. Table 3.3 displays the indices and Cook's Distance values for observations where the Cook's Distance surpasses the specified threshold.

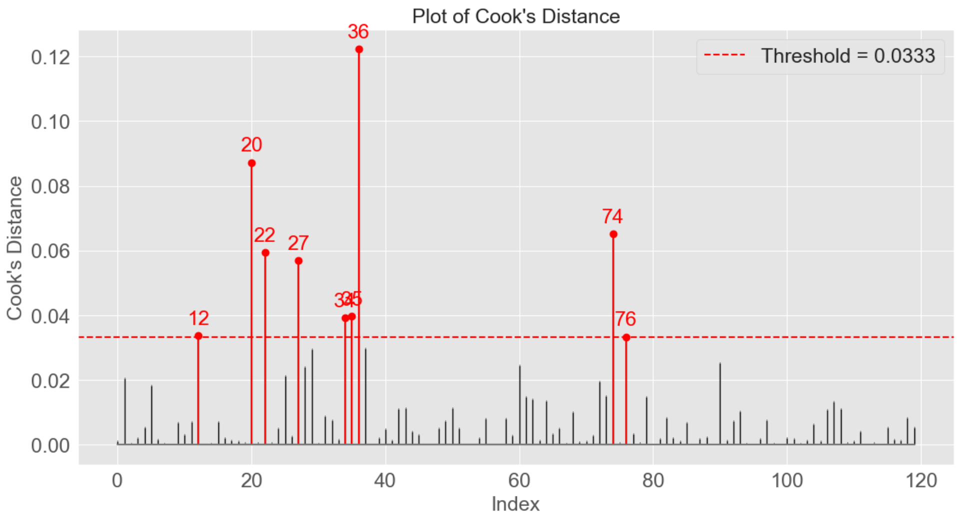


Figure 3.8.: Cook's Distance

Index	12	20	22	27	34	35	36	74	76
Cook's Distance	0.09	0.34	0.24	0.07	0.10	0.07	0.25	0.83	0.35

Table 3.3.: Cook's Distance of Observations of the Dataset `cacao.csv`

Figure 3.8 shows Cook's Distance plotted against the indeces of the corresponding observations. The x-axis shows the indeces of the observations, whereas the y-axis displays the value of Cook's Distance of those observations. The red dashed line indicates the threshold of 0.0333. Therefore, values above that line are considered as high-influence points. These points are portrayed as a red line with a red dot at the end, while black lines indicate that this observation is not crossing the cut-off value.

Looking at the result of the calculation of the leverage and Cook's Distance, the observations 20, 22, 36, 74 and 76 are particularly influential on the regression model, since those are the observations with both high-leverage and high Cook's Distance.

Non-Linearity

Linearity presents one of the key assumptions of the multiple linear regression model. Linearity means, that there is a linear relationship between our dependent variable `stem_diameter` and the independent variables. This assumption will be checked using the *Rainbow test* and a *residual plot*. The Rainbow test checks for the linearity assumption in regression models by comparing the fit of subsets of the data [28], providing a p-value of 0.25. Therefore the null hypothesis of the relationship between the variables being linear can not be rejected at the 0.05 significance level.

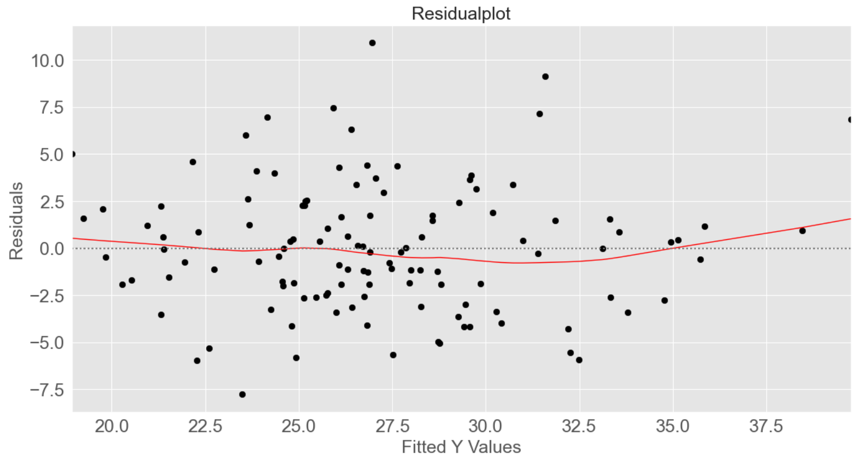


Figure 3.9.: Residuals Plot

Figure 3.9 illustrates residuals plotted represented by the x-axis against the predicted values made by the model displayed by the y-axis. The red line is a smoothed line that helps visualizing potential patterns in the residuals. Upon closer examination of this figure, randomly distributed residuals become clear. If the linearity assumption is met, the residuals should be randomly distributed. Otherwise, the linearity assumption is violated.

Heteroscedasticity

The *homoscedasticity* is the assumption in the model that the variance of residuals is constant. If this assumption is violated, which is called *heteroscedasticity*, the standard errors of the coefficients can be biased. This can lead to incorrect interpretations of statistial tests and confidence intervals. In order to check this assumption, the *Breusch-Pangan test* and a *scale-location plot* is used . The Breusch-Pagan test checks for heteroscedasticity in regression models [5], while a scale-location plot visualizes the spread of residuals to assess their homoscedasticity.

The Breusch-Pangan test provides a p-value of 0.07. Using the significance level of 0.05, the p-value being greater than 0.05 indicates that the null hypothesis of a homoscedastic variance can not be rejected. However, if a 10% significance level is used, the null hypothesis can be rejected. This indicates that there is a certain degree of heteroscedasticity present. The scale-location plot provides a visual aid for this result.

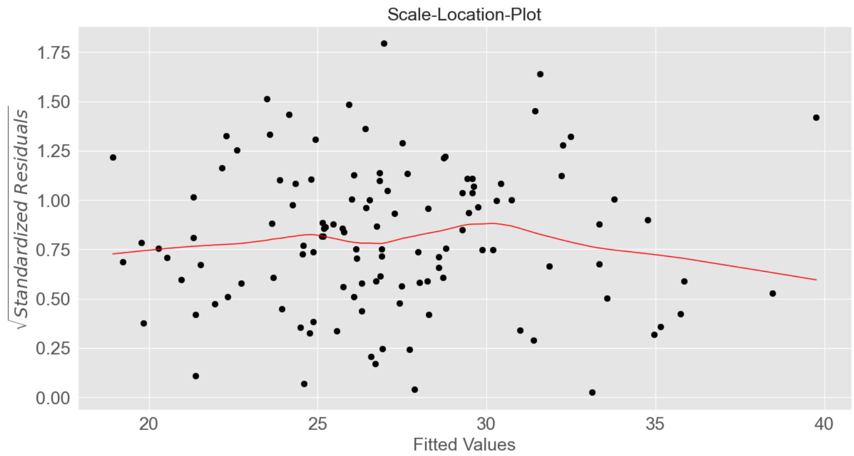


Figure 3.10.: Scale-Location Plot

Figure 3.10 shows the scale-location plot of the model. The x-axis represents the fitted values from the regression model, while the y-axis shows quare root of the standardized residuals. Standardized residuals are the residuals divided by their estimated standard deviation and therefore normalized. The red line is a smooth curve fitted through the points, helping to visualize any patterns or trends in the spread of the residuals. Ideally, the points should be randomly distributed to indicate constant variance of the residuals, illustrated by a horizontal red line. The points should not follow a certain shape for example like an U-shape or a V-shape. Here, it seems that the variance of the residuals slightly increases around the fitted values from 25 to 30 which only suggest some degree of heteroscedasticity. According to the Breusch-Pagan test and the scale-location plot, a slight violation of homoscedasticity can be assumed.

Correlation of Error Terms

In the linear regression model, the error terms should not be correlated. This assumption can be checked by using the *Durbin-Watson test* and a plot of studentized residuals over time. The Durbin-Watson test checks for autocorrelation in residuals [9]. The plot of studentized residuals over time assesses changes in residuals to identify patterns or trends.

The Durbin-Watson test provides a value of 1.44. The value is not within the recommended range, but it is very close. Therefore, it can be assumed that the error terms are slightly correlated.

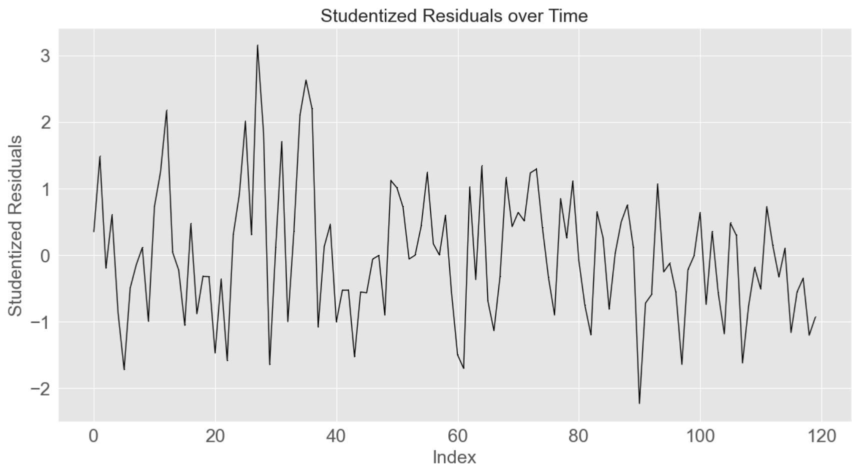


Figure 3.11.: Studentized Residuals over Time

This plot shows the studentized residuals of the plot (y-axis) and their indeces (x-axis). If the residuals appear to be randomly scattered around the zero line, it suggests that the residuals are independent. This plot shows a slightly decreasing pattern, which is why the correlation of error terms can be suggested. Both the Durbin-Watson test and the graphical aid provide the same results.

Normality of Residuals

The next assumption to check is the assumption of normality of residuals, which can be verified by using the *Shapiro-Wilk test* and the Q-Q plot. The Shapiro-Wilk test checks for normality [24]. The Q-Q plot visually compares the distribution of the data to a theoretical distribution [9]. In this case the normal distribution is used as the theoretical distribution.

The Shapiro-Wilk test provides a p-value of 0.2. Using the significance level of 0.05 the p-value being greater than 0.05 indicates that the null hypothesis of normally distributed residuals can not be rejected. This result can be verified by the Q-Q plot.

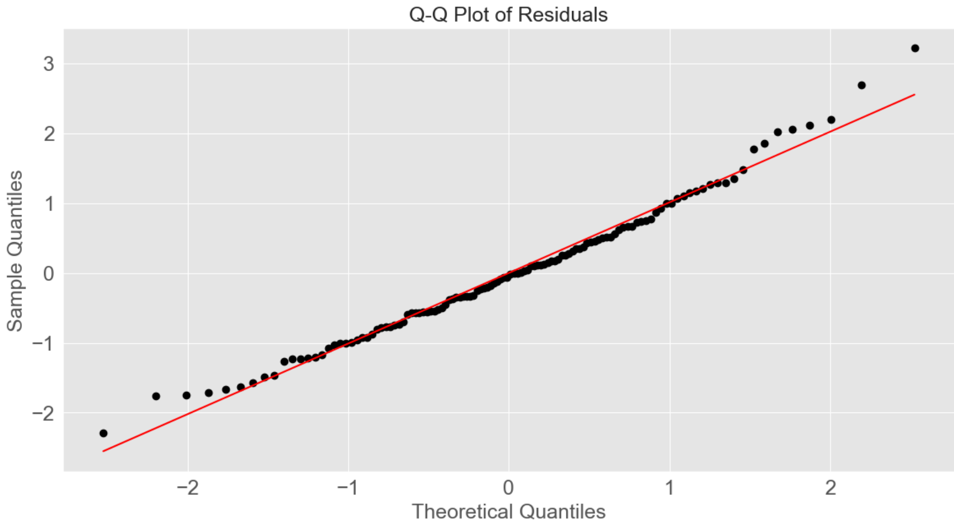


Figure 3.12.: Q-Q Plot of Residuals

The x-axis represents the theoretical quantiles of the normal distribution. The y-axis shows the observed quantile from the data. If the normality assumption is not violated the points should be aligned at the bisector, which is the red line in the plot. Most points remain close to the red line, but looking at the the beginning and the end of the red line, the points deviate as a tail from the red line. Nevertheless, the assumption of normality can be considered as not violated.

Collinearity of Predictors

In multiple linear regression, the independent variables should be linearly independent. In order to identify collinearity of variables, a correlation matrix and the *variance inflation factor (VIF)* can be used. The *VIF* is a measurement used to detect multicollinearity in a regression analysis [9]. A correlation matrix shows the correlation between multiple variables. High VIF values, meaning values greater than 10, indicate high multicollinearity between the independent variables.

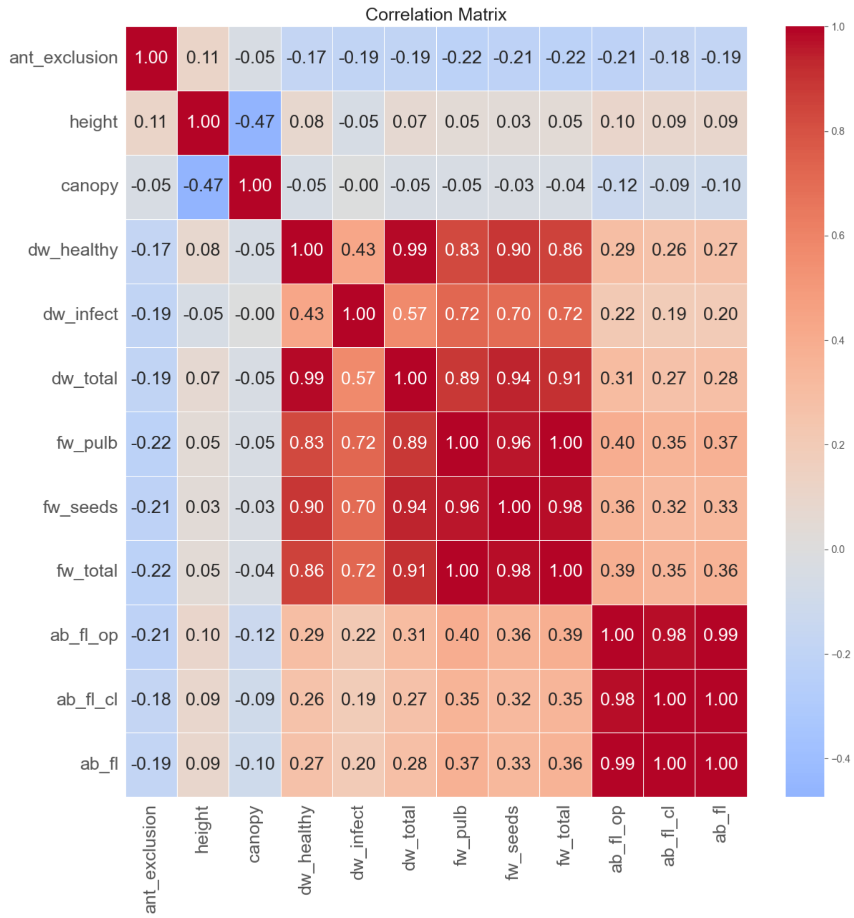


Figure 3.13.: Correlation Matrix

Figure 3.13 shows the correlation matrix of the dataset. The x-axis and y-axis both list the same set of variables, showing pairwise correlations between these variables. The heatmap uses a color gradient from blue to red. Red represents a high positive correlation (close to 1). Blue represents a high negative correlation (close to -1). The diagonal cells (correlation of a variable with itself) have a correlation coefficient of 1, indicated by dark red.

Variable	Variance Inflation Factor
fw_total	33773.31
dw_total	21920.80
fw_pulb	18933.44
dw_healthy	17354.99
ab_fl	9626.65
ab_fl_cl	4816.95
fw_seeds	2183.53
ab_fl_op	962.11
dw_infect	450.03
height	10.01

Table 3.4.: Variance Inflation Factor

By examining the VIF in Table 3.4 and the correlation matrix, it is possible to identify variables with high correlation. For example, `fw_total` has a very high VIF (33773.31) and is highly correlated with `dw_healty`, `dw_infect`, `fw_pulp` and `fw_seeds`. However, the correlation with other variables such as `ab_fl_op`, `ab_fl_cl`, or `ab_fl` is relatively high (greater than 0.3), which explains the high VIF.

Summary

Looking at the previous regression diagnostics, the following points can be summarized: In this dataset, observation 27 is considered an outlier according to the studentized residuals. Observations 20, 22, 36, 74 and 76 have both high leverage and high Cook's Distance, making them particularly influential on the regression model. The Rainbow test and the residuals plot indicate that the linearity assumption is met. According to the Breusch-Pagan test and the scale-location plot, a slight violation of homoscedasticity is assumed. However, the assumption of uncorrelated error terms is violated, as indicated by the Durbin-Watson test and the plot of studentized residuals over time. The normality of residuals can be assumed based on the Shapiro-Wilk test and the Q-Q plot. Finally, many variables are autocorrelated according to the correlation matrix and the VIF.

3.2. Quality Analysis: ChatGPT Before

The foundation has now been established to generate the output. The structure of the analysis has been demonstrated in the previous chapter. In summary, the analysis should start with a brief EDA and be followed by regression diagnostics. The methods should be briefly explained and applied to the dataset. Additionally, the results should be interpreted to help the learner understand the methods. The learner now wants to learn about regression diagnostics. Since they have a dataset, they upload it to ChatGPT and request that regression diagnostics should be performed. It raises the question whether the output is understandable, complete and correct so that it can assist the learner. As this output has not been optimized in any way, it will be analyzed in this section. The output of ChatGPT can be found in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `Code` (see Figure B.2) as `CacaoChatGPT_Before.ipynb` [14]. Therefore, this section is named "Quality Analysis: ChatGPT Before". The section "Quality Analysis: ChatGPT After" will include an analysis of the adjusted prompt and the resulting optimized output. The learner starts by uploading the dataset `cacao.csv` and typing the following prompt:

```
"Perform regression diagnostics to check the assumptions
of multiple linear regression and explain the methods
to me as a learner.
Take stem_diameter as dependent variable and
all the rest as independent variables"
```
(3.1)

Different criteria and questions will be used to analyze this output, primarily based on statistical correctness and learner-friendliness. The criteria contain topics like the structure of the output, the background on the topic and the evaluation of the regression diagnostics and are listed Table 3.5.

Criteria	Question
1)	Does it contain a brief explanatory data analysis at the beginning?
2)	Does it explain the assumptions of (classical) multiple linear regression (resp. violations of the assumptions)?
3)	Does it address all assumptions and topics to be checked?
4a)	Is there an introduction for the diagnostic tool?
4b)	Are there several diagnostic methods included (e.g. at least a plot and at least a statistical test) per category?
4c) i.	Is the method correct?
4c) ii.	Is the method explained to the learner (e.g. axes and lines for a plot, statistical test function, interpretation. . .)?
4c) iii.	Are there references?
5)	Does it contain a summary of the regression diagnostics?
6)	Does the output contain mathematical equations?
7)	Is the output well-structured and learner friendly?
8)	Does it generate reproducible `Python` code for each generated output?

Table 3.5.: Criteria of Quality Analysis

3.2.1. ChatGPT Before: Explanatory Data Analysis

The analysis should start with a brief exploratory data analysis to help the learner understand the structure of the data. ChatGPT provides no EDA, making it difficult for the learner to get a general overview of the data. Plots like histograms or boxplots are missing, which could be very helpful for the learner to understand the variables, their values and their distribution.

3.2.2. ChatGPT Before: Model Assumptions, Regression Diagnostics

After the brief EDA, the output should continue with an explanation of the regression model, its assumptions and the regression diagnostics. Another difficulty for the learner is to know the assumptions of multiple linear regression in order to check for any violations, which is the next problem of ChatGPT's output. It doesn't explain any of the assumptions specified in Chapter 5.1 explicitly. Without knowing the exact assumptions on which the regression diagnostics are based, it will be confusing for the learner to understand the diagnostics and why they should be used.

Even though the output lists different regression diagnostics, some are still missing. ChatGPT mentions linearity, homoscedasticity, normality of residuals and multicollinearity. However, outliers and high-leverage points were omitted as diagnostics. Outliers and high-leverage points are important to identify in regression analysis for several reasons. Outliers can distort parameter estimates and reduce model accuracy, whereas high-leverage points can disproportionately influence coefficients. However, it is positive that the statistical methods used by ChatGPT were all correctly implemented and applied. With regard to the statistical accuracy of the methods for the remaining diagnostics, there are no concerns.

Nevertheless, the remaining regression diagnostics will be evaluated. Each diagnostic catergory should have an introduction. After that the learner should be presented with the options to test a specific model assumption and be given a brief explanation. It should be clear how the applied diagnostic method is interpreted. Several diagnostic methods should be included for each category, such as at least one plot and at least one statistical test, with appropriate literature references. Afterwards, the methods should be applied and interpreted. Plots should be described and explained. Additionally, the values of the statistical tests and the test decisions should be presented to the learner. ChatGPT's output provides only a short introduction for each diagnostic tool. Combined with the missing model assumptions, however, it is not sufficient to give the learner a good overview of a specific diagnostic category. The learner may have difficulties understanding why these diagnostic tools should be applied. The statistical methods used are only briefly mentioned. They are neither explained in detail nor are specific thresholds provided in advance. Another major criticism is that only the normality of the residuals was tested using both a plot and a statistical test. Plots are generally not described and there is no detailed explanation of how they can be interpreted. Several statistical methods can be important for the learner to provide a comprehensive understanding of the diagnostic tools. The understanding of the topic can be deepened through various literature sources. Since literature references are missing, the learner has little opportunity to look up and study specific topics in more detail.

3.2.3. ChatGPT Before: Structure of the Analysis

One of the biggest problems for the learner-friendliness of the output is the structure. It starts by reading in the data, which is the only non-reproducible `Python` code. The learner has to change the filepath themselves. The remaining `Python` code provided in the output is reproducible if the filepath has been adjusted. Then ChatGPT lists the steps of the regression diagnostics to check the assumptions and briefly describes what is being checked in each diagnostic, but the assumptions have not been stated explicitly, as already analyzed in Chapter 3.2.2. Afterwards, ChatGPT lists each step again and briefly describes which statistical method is used. Finally, ChatGPT repeats the individual steps and statistical methods once more before providing the code for the statistical tests and plots all at once. Consequently, ChatGPT interprets each step individually and lists it again before briefly summarizing the results.

The problem with the structure thus can be divided into two problems. The first problem concerns the repetition of the diagnostic categories. Those categories are listed multiple times and content is often repeated. It would be more beneficial for a learner if all information for each diagnostic categories were provided in a single section. The learner would have all the information in one place and wouldn't need to search through the report. The second problem is the output of the code and the plots. In ChatGPT's output, the entire code and the plots are provided all at once. This makes it difficult for the learner to understand which code section belongs to which diagnostic category. Each diagnostic category should be in its own section and not be repeated. In each section, the diagnostic tool, the statistical methods, the interpretation and the code would be explained, instead of repeating the steps and addressing only one of the mentioned contents at a time.

3.2.4. ChatGPT Before: Summary of the Quality Analysis

Overall, it can be said that a learner does not receive a learner-friendly output from the mentioned prompt 3.1. It starts with the learner receiving little information about the dataset due to the lack of an EDA, making it hard to understand. Additionally, the learner does not get an overview of the assumptions of a linear model, making it difficult to understand the regression diagnostics. Furthermore, certain regression diagnostics are missing. There are not enough statistical methods provided for the learner to study them effectively. However, the methods used are statistically correct. Regression diagnostics and methods are not described enough. The learner is not sufficiently informed about their interpretation. Additionally, the structure is not learner-friendly, characterized by frequent repetition and disorganization. The prompt needs to be significantly adjusted to generate a learner-friendly output.

3.3. Quality Analysis: ChatGPT After

3.3.1. Approaches for Prompting

The analysis in Chapter 3.2 has shown the areas where the analysis of ChatGPT needs to be improved. By adjusting the prompt, an attempt will be made to achieve a output similar to the original analysis. The ChatGPT analysis will also include additional model equations, model assumptions and more detailed explanations of the methods used. Various approaches to prompting were tried and different observations were made.

In the first attempt, a single prompt was used to achieve the desired result. However, the problem was that the output was too long and ChatGPT stopped responding after a certain length. Therefore, the prompt was initially divided into three sections: Explanatory Data Analysis, Preparation of the Regression Model and Regression Diagnostics. Each of these topics was then addressed using a separate prompt. However, the same problems occurred in each of the three separate prompts. Each prompt needs to cover all the necessary topics and is therefore quite extensive. This creates a challenge in covering all the necessary information while ensuring the output is easy to understand and learner-friendly. The main issues with the output were the inaccurate presentation of the results and the lack of a learner-friendly structure. Both were related to the complexity of the prompt. ChatGPT often explained each individual step and repeated information multiple times, which led to an unnecessarily long output. Additionally, due to the long prompt, there were often no outputs of plots or test results. ChatGPT still made statements about the plots, suggesting that they were calculated but not displayed. Without the visualizations, the learner cannot understand or verify the statements.

Therefore, it was decided to further divide the prompt for the three topics. The prompt was shorter and contained less information, but there were more step-by-step prompts used to achieve the results. This had positive effects as a result. On one hand, shorter input resulted in more detailed outputs. The responses given were less frequently written in bullet points. The provided sentences were more precise and comprehensive. Additionally, during the period where the prompt was tested, there were no missing displays of test results or plots. There were hardly any repetitions by ChatGPT, which had only been partially avoided with the previous prompt. The structure guidelines were followed more precisely. For these reasons, a separate prompt was used for each subtopic (e.g., histograms, model assumptions, or outliers).

A complete report for the dataset `cacao.csv` was generated using this type of prompt. By exporting the chats from the ChatGPT account, it was possible to extract the individual outputs of the prompts. The output was then manually combined with the code that ChatGPT used for the analysis, as seen in the original chat. The symbols "\(", "\)", "\[" and "\]" need to be changed to "$" to ensure the correct display of mathematical formulas. The report has been made available to potential learners as a Jupyter Notebook. The Jupyter Notebook can also be exported as a PDF or HTML file. This process was made because, while the chat history can be shared with others, any visualizations from the analyses conducted by ChatGPT are not displayed. Without the visualizations, a significant part of the report is missing. The process is planned to be automated in the future.

3.3.2. Utilized Prompts

Before starting the prompt for the analysis, overall instructions for ChatGPT are defined by using the following prompt:

```
"Adress me as a learner.
I do not have any previous experience in Data Science.
Explain in a simple way.
I will give you a dataset, questions and instructions"
```
(3.2)

The prompt (3.2) signals, that the user wants to be addressed as someone who is new to data science with no prior experience. Concepts, processes or instructions should be explained in a simple way without assuming any prior knowledge of the learner. The output of ChatGPT focuses on breaking down complex topics into basic steps by using simple language to ensure that the learner can follow the topic.

The first part, the explanatory data analysis, starts by providing an explanation of descriptive statistics and the corresponding table for the dataset.

```
"Explain exactly what descriptive statistics are."
```
(3.3)

```
"Provide the table of descriptive statistics of this dataset
without using ace_tools. Do not explain the values."
```
(3.4)

```
"Explain the summary statistics"
```
(3.5)

By seperating this prompt into three parts (prompt (3.3), prompt (3.4) and prompt (3.5)), the output of ChatGPT becomes clearer and more detailed. After prompt (3.3), the dataset is uploaded to ChatGPT. It is stated that the `ace_tools` library should not be used. Otherwise, ChatGPT will automatically use this library for descriptive statistics without it affecting the output. By specifying not to explain values, output that describes specific values in the table is prevented and thereby avoiding unnecessarily lengthy and confusing analysis. For the different plots a certain scheme was used to generate homogeneous output.

```
"I want to know more about [Plot]
1.  What are [Plot]?
2.  What are the components of [Plot]
3.  How do I interpret [Plot]
4.  Provide [Plot] for every variable of the dataset.
Use [Function] to display the [Plot] in a grid format."
```
(3.6)

In the prompt (3.6), the field `[Plot]` serves as a placeholder for the desired plot in the exploratory data analysis, while `[Function]` is a placeholder for a specific function to create the plot. In this field functions of different libraries can be placed for personalizing the appearance of the plot. This prompt ensure that the learner gets a detailed explanation of the plot, including information on description and interpretation, in an learner-friendly output. After getting all plots, explanations and interpretations for the EDA, preparations for regression diagnostics must be done, including model equations, model assumptions and model construction.

"What is multiple linear regression? Use x_{ij} for
```
independent variables.  Explain the ranges of i and j.
Explain what observations and predictions are.
Explain the assumptions of a (classical) linear regression model
in detail and simple, including mathematical equations.
Do not provide additional considerations or methods
for checking the assumptions.
Summarize the assumptions in mathematical form."
```
(3.7)

After applying prompt (3.7), the learner is given a detailed introduction to multiple linear regression, where the model and its assumptions are explained. Technical terms like observations and predictions relating to regression models are presented. An output of additional considerations or methods for checking the assumptions is prevented, as these are to be defined by the user. It was already shown in Chapter 3.2 that ChatGPT had some inaccuracies in the regression diagnostics.

```
"Build an OLS regression model
using [Variable] as the dependent variable
and all [Variable] as independent variables.
Do not display the regression model summary or parameters."
```
(3.8)

Now, the preparations for regression diagnostics are finished. The model on which the regression diagnostics will be examined has to be built using prompt (3.8). The field `[Variable]` is a placeholder for the dependent and independent variables, chosen by the learner. The output of regression model summaries or parameters is suppressed, since it would cause unnecessary information. The goal is to understand the regression diagnostics and not to find the best model for the dataset.

```
"Explain [Diagnostic] to me.
Then, explain [Method] to me and provide mathematical equations
I want to understand the basic idea of [Method].
Afterwards, tell me if [Diagnostic] is violated by using
[Method] with [Threshold] as threshold and [Plot].
Explain and interpret the plot.  Explain how to read the plot.  (*)
Additional infos on the plot:
- use [Adjustments]"
```
(3.9)

Prompt (3.9) shows the schematic structure for every diagnostic tool. The field `[Diagnostic]` refers to the diagnostic tool, while `[Method]` and `[Plot]` are the corresponding methods described in Chapter 5. The additional field `[Threshold]` can be used, if a certain threshold for the method is known, whereas `[Adjustments]` is a placeholder that represents graphical adjustments for the plot, for example functions of different libraries, displaying thresholds or annotations. For the diagnostic tools outliers and high-leverage points, which directly follow one another, certain adjustments need to be made.

$$\texttt{"Should regression diagnostics be repeated after removing potential outliers?"} \tag{3.10}$$

$$\texttt{"Do not answer the question whether regression diagnostics should be repeated after removing high-leverage points."} \tag{3.11}$$

For the prompt of outliers sentence (3.10) is added after `(*)` in prompt (3.9). Similarly, for high-leverage points sentence (3.11) is added after `(*)`. This ensures that the learner is informed that regression diagnostics should be repeated after outliers have been removed. If prompt (3.11) is omitted, the learner will also be told that high-leverage points should be removed, which is not always the best solution.

$$\texttt{"Summarize the results of outliers, high-leverage points, non-linearity, heteroscedasticity, correlation of error terms, normality of residuals and collinearity of predictors."} \tag{3.12}$$

$$\texttt{"Provide code to install all necessary libraries. Provide your used code of the whole conversation in one .py script for me to copy"} \tag{3.13}$$

$$\texttt{"Explain statistical tests, p-values and sample size considerations in a simple and short way."} \tag{3.14}$$

At the end, prompt (3.12) can be used to generate a summary of the results. This provides the learner with an overview of the analyses that have been conducted. Additionally, prompt (3.13) can be used to request the code as a .py file, which can then be downloaded or copied directly from the chat. Finally, through prompt (3.14), the learner receives additional information about statistical tests and sample size, which can potentially be used to question the uploaded dataset in terms of its sample size. Literature was not included in the output because, due to lack of access, the content could not be verified. If the content is accurate, literature recommendations can also be requested from ChatGPT.

3.3.3. Comparison to Own Analysis

The analysis from ChatGPT was created using a customized prompt so that all the criteria from Table 3.5 are now met. The output of ChatGPT can be found in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `Code` (see Figure B.2) as `CacaoChatGPT_After.ipynb` [14]. This prompt has improved the heavily criticized structure from Chapter 3.2 and now resembles the learner-friendly structure of my own analysis. As already described in Chapter 3.1 my own analysis only contains the mentioning of methods, their results and their interpretation, which formed the basis for the prompt and the output of ChatGPT. Additionally, the AI's output contains a detailed explanation of the methods, model equations and assumptions. To evaluate the output, it is compared with my own analysis to demonstrate that the output can offer significant value to a learner and, with some prior preparation, can serve as an alternative to human assistance.

Content	Own Analysis	ChatGPT's Analysis
Is an introduction to the EDA present?	No	Yes
Is a short EDA included?	Yes	Yes
Is there an introduction to multiple linear regression?	No	Yes
Is there an explanation of multiple linear regression model assumptions?	No	Yes
Are all Diagnostics included?	Yes	Yes
Is the method correct?	Yes	Yes
Is the explanation of the methods present?	Yes, but short	Yes
Is the interpretation of the methods present and correct?	Yes	Yes
Is the summary of results present and correct?	Yes	Yes
Are there additional information?	No	Yes

Table 3.6.: Comparison of Own Analysis and ChatGPT's Analysis

In contrast to my own analysis, as previously explained, ChatGPT begins with an explanation of descriptive statistics to prepare the learner for the output, specifically the table of descriptive statistics. It explains what can be observed from a dataset using descriptive statistics and clarifies the summary statistics. This explanation of the measurements and the table of descriptive statistics is also included in my own analysis, as it is necessary for interpretation. Similiar, for each of the different plots, ChatGPT provides an introduction. These introductions are missing in my own analysis. It explains what the plot represents, what components it consists of and how it should be interpreted. One difference in the presentation of the histograms is that ChatGPT does not display the skewness value in the diagram, although this can be adjusted in the prompt. The other plots also differ primarily in their visual representation. Additionally, the style of the `seaborn` library used by both analyses can be changed in the prompt. The style or, for example, the color of the plots are usually just personal preferences that can be adjusted as desired. Therefore, the current difference in presentation does not present an issue with ChatGPT's output, as the interpretation and meaning of the generated plots remain unchanged.

Moving on to the topic of multiple linear regression, in ChatGPT's analysis, in addition to the listing of the regression diagnostics, which is also present in my own analysis, further important theoretical information is provided. The topic of multiple linear regression is explained in detail, including the presentation of mathematical formulas and their explanations. The model equation and model assumptions are covered. These points are missing in my own analysis.

Nevertheless, both analyses contain every diagnostic tool needed with matching methods. For every diagnostics at least one statistical test and one plot is used, ensuring different ways to portray the diagnostic tool to the learner. The plots show no differences apart from minor graphical features and the results of the statistical tests are the same. An advantage of ChatGPT's output is that the methods are explained in more detail. It dives deeper into how the methods work. The explanations aim to provide the learner with a general overview of the methods, enabling them to better understand and interpret the results. My own analysis does not include these explanations, it only presents the interpretation of the provided results. However, when it comes to interpretation, both analyses are quite similar and don't differ much. Both analyses also provide a summary of the results. In ChatGPT's output, additional information is provided at the end regarding the relationship between statistical tests and sample size, which could lead the learner to question the sample size of the dataset and potentially choose a different dataset if necessary.

In summary, it can be said that both analyses correctly apply and interpret the methods. In this regard, there is no reason to prefer one analysis over the other. However, the AI-generated output offers more advantages, provided that the correct prompt is used. The challenge is to find the right prompt to generate learner-friendly output. This prompt was provided in Chapter 3.3.2. This allows to utilize the great strength of the AI-generated output. ChatGPT can quickly make precise and, in the case of multiple linear regression, statistically accurate statements, while also providing comprehensive and learner-friendly introductions to the topics. Regarding the methods used, neither analysis is better than the other. However, the learner-friendly explanation of the topics and methods gives ChatGPT's output an advantage.

3.3.4. Evaluation of Learner-Friendliness

The output has been checked for statistical accuracy and completeness so far. Efforts were also made to achieve learner-friendliness, but this has not been verified yet. For this purpose, a survey was created, which was answered by a targeted group of students. All respondents had a background in mathematics and were familiar with certain statistical fundamentals. For example, concepts like distributions, variance, or hypothesis testing were not unfamiliar terms. The participants were presented with the output of ChatGPT using the adjusted prompt and were asked to answer questions about it. The questions primarily focused on the participants' understanding of the topic. The whole questionnaire can be viewed in the attachment Figures B.10 to B.14.

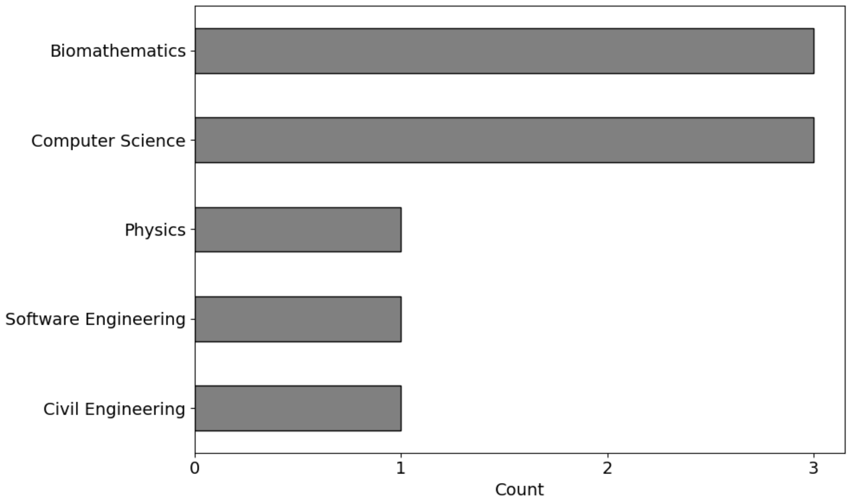


Figure 3.14.: Field of Study of Participants

In total, there were nine participants, coming from the fields of Biomathematics, Computer Science, Physics, Software Engineering and Civil Engineering.

The survey was devided into three parts. Each question could be rated on a scale from one to five, with one representing 'Strongly disagree', two for 'Disagree', three for 'Neutral', four for 'Agree' and five for 'Strongly agree'. The first part focused on evaluating the EDA. Participants were asked whether they understood the various plots (histograms, boxplots, ECDF plots, and Q-Q plots). Additionally, they answered whether the EDA helped them gain an overview of the distribution of the variables of the dataset. Therefore, the first part consists of five different questions. The second part focused on the preparations for the regression diagnostics, specifically the multiple linear regression model to which the diagnostics would be applied. Participants indicated whether they understood the principles, including the mathematical formulas for the multiple linear regression model, as well as the model assumptions. Therefore, the second part consists of two different questions. In the third part, participants answered questions about the regression diagnostics. They were asked if they understood the diagnostics, could follow the methods and found the plots and the basic idea of the statistical tests understandable. They also indicated whether enough information was provided to understand the plots and statistical tests. Therefore, the third part consists of six different questions for each of the seven diagnostics.

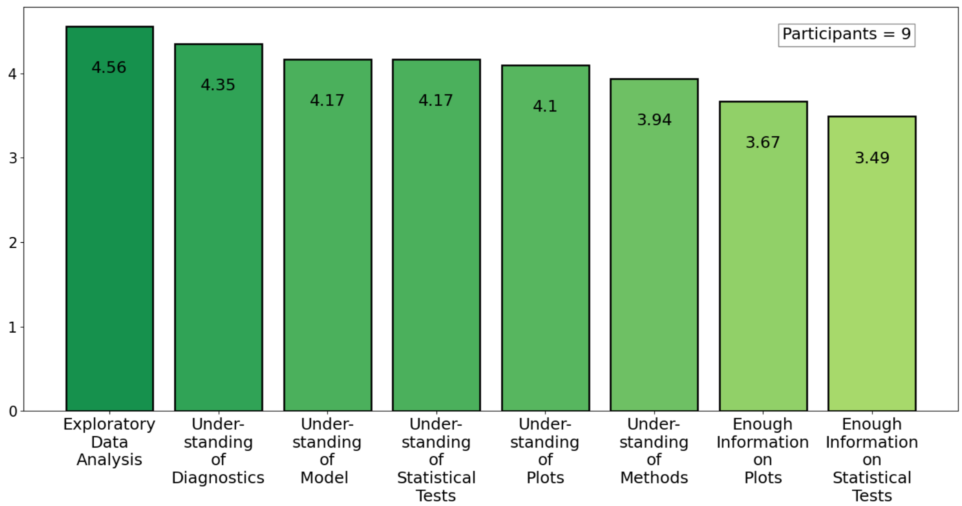


Figure 3.15.: Average Rating for Understanding by the Learner

Figure 3.15 presents the average ratings for various questions and provides an overview of the general understanding of the EDA, the model, and all diagnostics. It is important to note that the evaluation of the questions in the questionnaire, regarding whether the participants need more information, was recoded for representation in the graphs. Accordingly, the answer 'Strongly Disagree' was coded as a five, 'Disagree' as a four, 'Neutral' as a three, 'Agree' as a two and 'Strongly Agree' as a one. The question represents whether the participants received enough information in the graph. The bars are displayed in a color scale ranging from red, which corresponds to a rating of one, to green, which corresponds to a rating of five. The color transitions gradiently from red to green. Overall, it can be seen that participants rated the provided information on the plots and statistical tests, as well as their general understanding of the methods, the lowest. However, this rating is by no means to be considered poor. The average rating for these questions was still between three and four, meaning the responses were generally 'Neutral' or 'Agree.' As a result, the understanding of the methods and the provided information was still seen as sufficient. Fortunately, the other questions regarding understanding were rated on average with at least four, meaning 'Agree'. The participants indicated that they generally understood the topics.

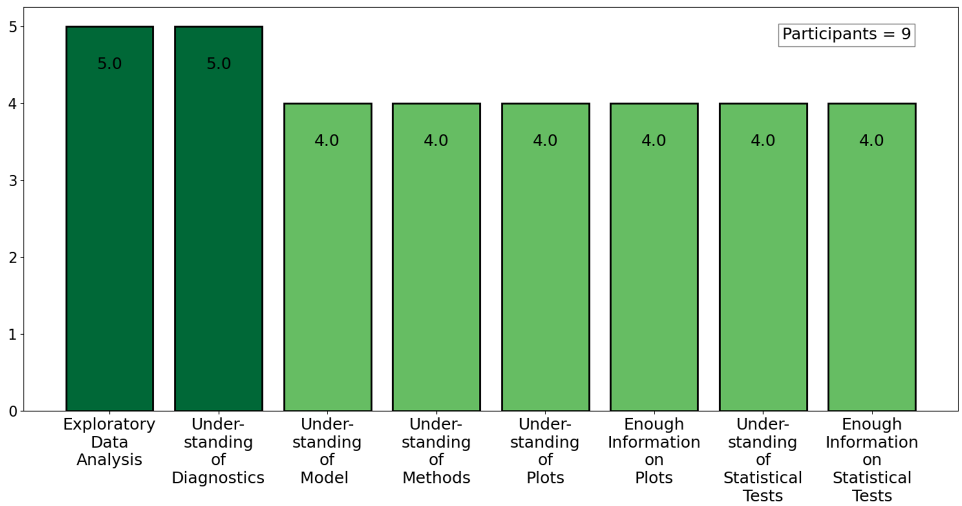


Figure 3.16.: Median of Rating for Understanding by the Learner

Due to a skewed discrete distribution, location parameters, such as the mean, can be distorted. This representation from Figure 3.16 provides further insight into the distribution of the responses by providing the median. Since the median represents the middle of the responses, it is very positive to note that, for each grouping of questions, half of the responses were four or higher. Therefore, it can be said that the participants generally had a good understanding of the output. Since the primary focus is on learning regression diagnostics, it is interesting to know which aspects the participants understood the best and which they understood the least.

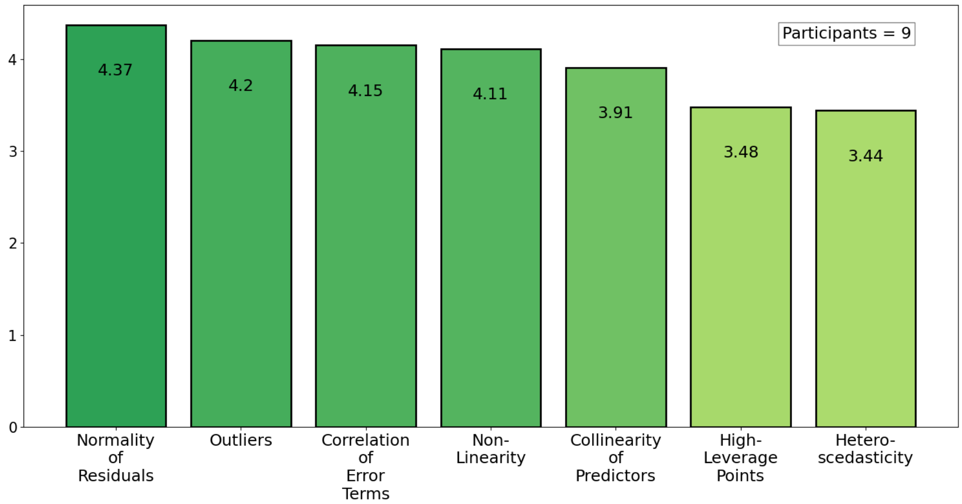


Figure 3.17.: Average of Rating for Regression Diagnostics by the Learner

Figure 3.17 shows that participants understood the normality of residuals the best and heteroscedasticity the least. However, even the average rating for heteroscedasticity was still above three.

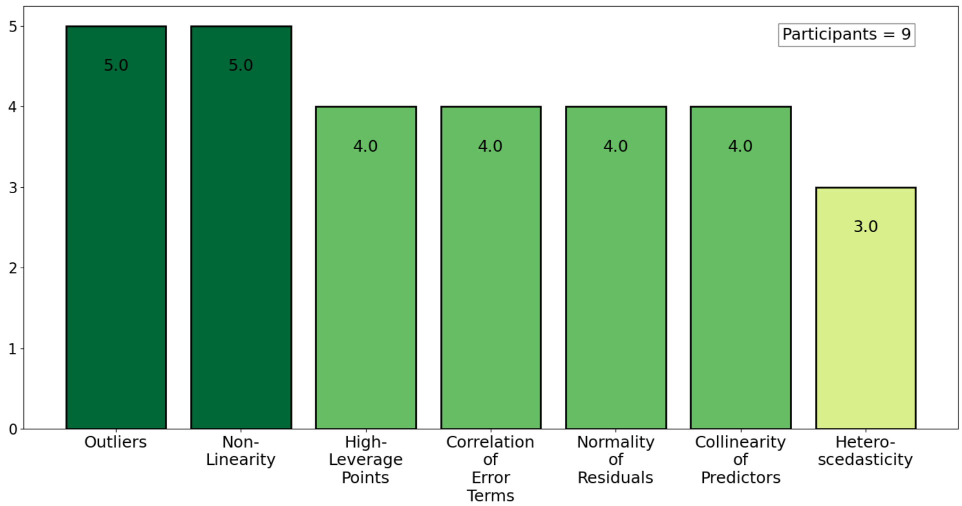


Figure 3.18.: Median of Rating for Regression Diagnostics by the Learner

Again, due to the skewed discrete distribution, it is important to also consider the median. Here, the median reveals that at least half of the ratings were three or higher. However, a rating of three indicates the answer 'Neutral', which does not mean that the topic was poorly understood. For the diagnostics of outliers, non-linearity, and normality of residuals, at least half of the ratings were five, which strongly indicates a learner-friendly output.

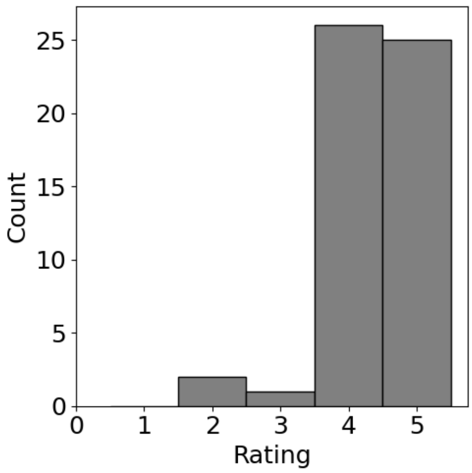


Figure 3.19.: Answers of Normality of Residuals

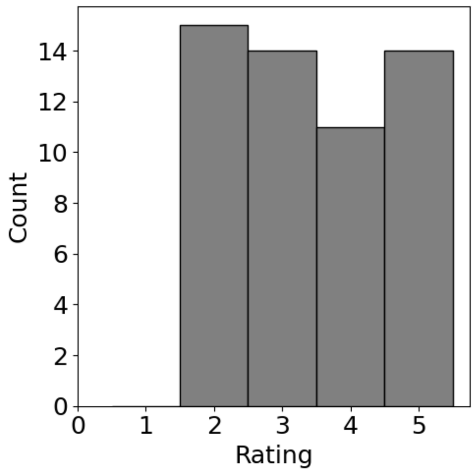


Figure 3.20.: Answers of Heteroscedasticity

Figure 3.19 shows the total number of ratings given for normality of residuals, while Figure 3.20 shows the same information for heteroscedasticity. For each diagnostic tool, there were six questions that, as previously described, could be rated on a scale from one to five. Since there were a total of nine participants, a total of 54 responses were provided for each diagnostic tool. The number of each rating in the 54 responses was counted and represented in histograms. The topic of normality of residuals was understood the best, as reflected in the histogram. There were hardly any questions rated with a two. A rating of two would indicate disagreement. In contrast, for heteroscedasticity, this was more frequently the case. More than 25% of the responses were rated with a two, but those ratings primarily concerned the understanding of the plot and the information provided about the plot, indicating that the topic of heteroscedasticity was generally still understood, aside from the plot. Because the plot was often not fully understood, this also impacted the ratings regarding whether sufficient information about the plot was provided and the overall understanding of the methods.

3.3.5. Perspective of a Learner

The output generated by an AI tool like OLSAI is intended to help learners understand specific topics like regression diagnostics. The questionnaire includes opinions on the output of an AI tool. However, a more critical survey regarding AI tools like ChatGPT, in which participants are allowed free speech, would be interesting. Therefore, two survey participants were asked for their perspectives. The participants, or learners, were asked four different questions regarding their perspectives on the use of AI tools. Since the responses reflect the participants' personal opinions, no assessments of their statements will be made. Instead, their answers will be summarized. The answers and concerns can be used to continually improve AI tools and provide a glimpse into the acceptance of AI tools like OLSAI.

Question 1:
"Were you aware of the complexity of the topic beforehand? Did you expect that ChatGPT could provide such a comprehensive insight into the subject? What do you expect from an AI tool that is supposed to explain a specific topic like "Regression Diagnostics" to you?"

Answer Participant 1:
"Yes and no, I already knew that there are many assumptions in linear regression, but I wasn't aware of the extent of diagnostic methods to check them. I wouldn't have expected ChatGPT to handle this so well, especially in terms of formulas and plots. I expect it to explain all aspects of the topic (in this case, "Regression Diagnostics") in detail and to require only minimal prior knowledge."

Answer Participant 2:
"I wasn't aware of the complexity and I'm actually very impressed with how ChatGPT explained the topic. Although I sometimes needed more information, I believe that you can ask for it, and it will be explained afterward."

Question 2:
"After seeing what ChatGPT can generate as output for this topic, can you imagine using AI tools in the future to help you learn certain subjects?"

Answer Participant 1:
"Yes, although I still prefer learning with real people. In videos, more time is dedicated to complex parts. I think ChatGPT lacks the empathy to know that a student might struggle more in one area than in another. It seems to explain everything quite thoroughly, but also covers things that might be unnecessary or trivial, giving the same level of explanation to complex topics. This is based on my own experience, not specifically on the material presented."

Answer Participant 2:
"Absolutely. I already use AI tools to help with my study notes and lectures. I can definitely see myself using the AI tool you showed me in the future. It really helped me understand the topic."

Question 3:
"What concerns do you have about using only an AI tool as a learning aid?"

Answer Participant 1:
"Too quick acceptance of the AI's output. I believe it could be a problem if students stop questioning things themselves and only rely on the AI. This approach might lead to a lack of learning effect, where students only memorize what the AI suggests. I think that learning exclusively with AI tools could also quickly result in not engaging with the topic independently."

Answer Participant 2:
"Based on my experiences, I have no concerns. ChatGPT explains things very well, and the ability to access both new and old sources sounds great."

Question 4:
"Where do you see the limitations of AI tools for learners? Could a lack of personal and human communication be a problem from your perspective?"

Answer Participant 1:
"Yes (see 2.), and even more so in personalized learning. I believe the AI won't recognize exactly where the understanding problems lie and will continue to explain things in detail rather than precisely addressing the student's specific

issue, unless the student can clearly articulate their problem. Additionally, I can imagine that the AI might try to convey too much information at once, which could compromise fundamental understanding. I didn't observe these problems with the AI tool presented, but it must be considered that I was already familiar with the topic and wasn't encountering many things for the first time. I think that learning with AI also depends on the individual."

Answer Participant 2:
"I don't think communication could be a problem, but that varies from person to person. I see fewer limitations because ChatGPT doesn't base its responses solely on sources."

In summary, it can be said that the overall attitude towards the use of AI tools is generally positive. ChatGPT and other AI tools are often already being used for learning. Participants can envision using them in the future. However, there are some concerns about using AI tools, such as the potential lack of learning effect or the absence of human interaction and these concerns vary from individual to individual.

3.4. Reproduction with other Datasets

The prompt mentioned in Chapter 3.3.2 created a correct and learner-friendly output for the dataset `cacao.csv`. Both the EDA and the explanation of multiple linear regression and regression diagnostics were helpful for the learner. To check if the prompt is reliable, it is important to use the same prompt on a different dataset. The explanations should be the same and the results should be just as clear as they were for the dataset `cacaco.csv`. Two datasets were used to show the reproducibility. The first one is the `cacao.csv` dataset, where the observations of the identified outlier in Chapter 3.1 is omitted. The second dataset used to verify the reproducibility is `Electricity1955.csv`, describing the cost function data for 159 US electricity producers in 1955.

First, the dataset `cacao.csv` will be analyzed without the identified outlier using the optimized prompt, as it generally makes sense to repeat the regression diagnostics after removing outliers. Fortunately, there are no content differences compared to the first analysis with the adjusted prompt through ChatGPT. The explanations of the theory include no changes and are complete. When reapplying the prompt to a different dataset, it can happen that some graphic elements, which were not specifically defined, might be randomly adjusted by ChatGPT. These graphic changes are not errors, as they are simply a matter of personal preference and do not affect the overall message of the plot. If the learner, for example, wants the points in the plot to be black, this can be specified as additional information in the prompt.

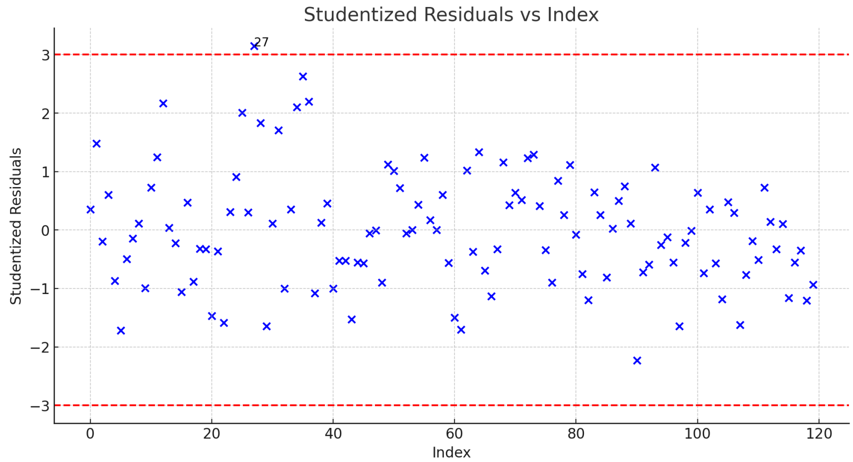


Figure 3.21.: Outlier Detection of `cacao.csv` Before Outlier Removal

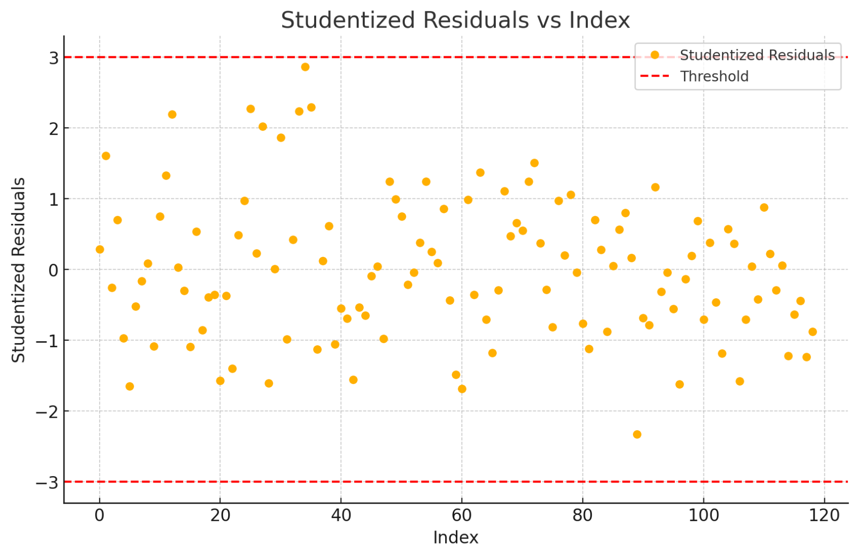


Figure 3.22.: Outlier Detection of `cacao.csv` After Outlier Removal

Figure 3.22 represents the plot of studentized residuals against the indices of the observations before the outlier was removed, whereas Figure 3.21 shows the same plot with the outlier being removed. Both figures were generated by ChatGPT. ChatGPT uses a different way to display the points. Instead of blue crosses, it used orange dots. However, the explanation of the plot remains the same and the interpretation is still correct and relevant to the plot. After the outlier was removed, the model was created, and when checking for outliers again, none were found. The pattern of correctly and fully applied methods, along with accurate explanations and interpretations of the methods and results, continues consistently throughout the entire output for the dataset `cacao.csv` without outliers.

Since only one observation was removed from this dataset, it was expected that the results for the output would remain the same. Therefore, the new dataset `Electricity1955.csv` will be used to apply the prompt to a different dataset with different values. Fortunately, there is once again nothing to criticize in the output for the new dataset, which strongly suggests the reproducibility of the output using the prompt. The theoretical explanations remained consistent and no differences were observed. The plots also do not change in content, except for the small graphical features already described. All the required plots were correctly created and displayed.

Figure 3.23.: Residuals vs. Predicted Values of `Electricity1955.csv` by ChatGPT

Naturally, the new dataset produces different results. Additionally, a different linear model was used for this dataset. The formula

```
log(cost/fuel) = log(output) + log(labor/fuel) + log(capital/fuel)
```

was used for the linear regression model. The new dataset leads to different results for the diagnostics. For example, the linearity assumption is violated in the dataset `Electricity1955.csv`, unlike in the previous dataset `cacao.csv`. The result is shown in the Figure 3.23. Additionally, ChatGPT calculated the p-value of the Rainbow test, which is $2.42 \cdot 10^{-20}$. Correctly, ChatGPT detected a violation of the linearity assumption based on the visible pattern in the plot and the p-value. Overall, ChatGPT correctly interprets the given graphics and test results for the new dataset across all other diagnostics as well, presenting the interpretations in a coherent and logical manner.

In summary, the prompt is reproducible across different datasets. ChatGPT consistently provides the same information, generates all required plots and test results and accurately answers the questions posed in the prompt each time. The test results are consistently interpreted correctly. Although the visual aspects of the plots may vary slightly between instances, the content remains consistent and the plots can always be correctly explained and interpreted. However, it must be noted that in the future, many more different datasets need to be tested to ensure the reproducibility of the prompt with certainty.

3.5. The OLSAI Assistant in `Python`

3.5.1. The Description of the Code

The previous chapters laid the groundwork for the OLSAI report. It has been demonstrated what structure the output should have, which prompts can be used, and the user-friendliness was analyzed. Based on these results, the AI assistant can now be developed. OpenAI offers an API that allows the use of ChatGPT without needing to visit the website directly. This API can be used to build various AI assistants. The documentation provides detailed explanations on how to use the API with `curl`, `node.js`, and `Python` [20]. For this project, `Python` was chosen due to its frequent use in the field of Data Science. Additionally, when analyzing datasets, ChatGPT uses `Python` and executes the `Python` code within a secure code execution environment. The `Python` script serves as the basis for the AI assistant. The goal of this code is to automate the generation of the OLSAI report without needing to manually enter the prompt on the website. Instead of receiving the output as a chat log between the learner and ChatGPT, the learner will get the results as a formatted HTML report. In the future, the code will be expanded, improved and integrated into a website. The complete `Python` code can be viewed in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `python_code` (see Figure B.3), which is located in the `Code` folder (see Figure B.2), as `olsai_in_python.py` [14].

3.5.2. The Key Elements of the Assistants API

To fully understand how the AI assistant is built and how the code for the output is structured, it's essential to first explain how the assistant works.

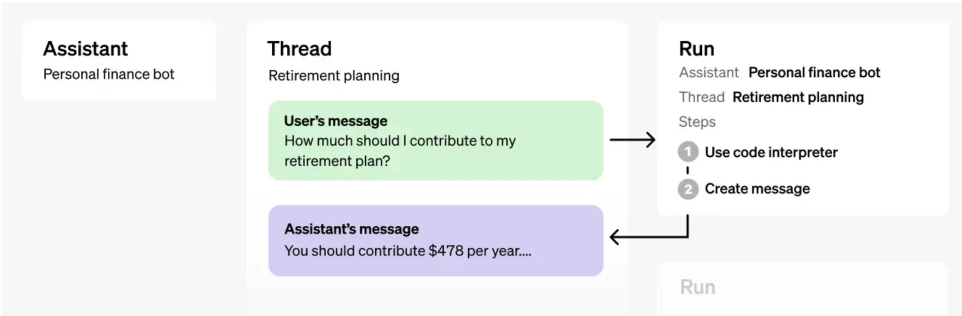


Figure 3.24.: Objects of Assistants [20]

The AI assistant is fundamentally composed of three interconnected different components: the *Assistant*, the *Thread* and the *Run*. Each of these components plays a crucial role for the assistant's function. The Assistant is the entity that is responsible for gathering information using OpenAI's models and tools and is capable of performing different tasks. It can be customized with specific settings and instructions to fit different needs, which will be explained in more detail later. The *Thread* is the object in which the messages and the conversation between the assistant and the user are stored. With such a stored conversation, the assistant can refer back to previously answered questions and use those answers in new responses. In order for the assistant to respond to the user's message, the *Run* object is necessary. During a run, the assistant calls on various models and tools to respond appropriately to the message. This object ensures that the assistant selects the right tools to provide an appropriate response. All of these three components work together to make sure that the AI assistant provides appropriate information and responses. After the user inputs a message, it is processed within the Run object. The assistant then attempts to use various tools to generate an appropriate response, which is eventually stored in the Thread [20].

3.5.3. The Process of Progamming

The complete code for the AI assitant can be viewed in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `python_code` (see Figure B.3), which is located in the `Code` folder (see Figure B.2), as `olsai_in_python.py` [14]. The most important code snippets are also presented here. To begin, all the necessary `Python` packages must be installed and loaded. The following packages were used in the code:

- `os`: Interact with operating system to retrieve OpenAI API key
- `openai`: Building the assistant using OpenAI's package
- `requests`: Making HTTP requests to download the used dataset
- `pathlib`: Handle filesystem paths
- `markdown2`: Convert Markdown text into HTML
- `tabulate`: Format data in tables
- `re`: Enable text manipulation

First, the client has to be created to easily interact with the API. For this, the API key, which can be generated in ones OpenAI account, needs to be loaded (3.15).

```
openai.api_key = os.environ.get("OPENAI_API_KEY")        (3.15)
client = openai.OpenAI()                                  (3.16)
```

There are different methods for reading this key. In the `Python` script, it was decided to store the API key in the Windows environment as `OPENAI_API_KEY` and load it from there using the `os` package. To use the code, the API key must be saved as `OPENAI_API_KEY` in the Windows environment. Using the key, the client can then be created (3.16). Next, the dataset for the analysis needs to be provided. The previously analyzed dataset, `cacao.csv`, was used and downloaded using the `requests` package. Finally, the dataset is uploaded to the client, for which the `pathlib` package was also used. Each file uploaded to the client receives an ID to reference it.

After the client was created, the assistant and the thread still need to be set up. When creating the assistant, various parameters can be specified (3.17).

```
olsai = client.beta.assistants.create(
```

```
name, instructions, tools, tool_resource, model, temperature)
```
(3.17)

For one, the name of the assistant and the instructions must be provided. The instructions guide the assistant on how it should behave. The name of the assistant is OLSAI and it has been given the instruction "You are the best teacher who can explain things the most clearly. You know everything about exploratory data analysis, multiple linear regression, and regression diagnostics. Adress the user as a student, who has no knowledge about data science". Additionally, the tools that the assistant should use must be specified. The assistant should use the *Code Interpreter*, which is designed to write and execute `Python` code and process files. The attached file is referenced using the previously generated file ID in `tool_resources`. The used ChatGPT model also needs to be selected. The used model is the latest model, `gpt-4o-2024-08-06`. Optionally, the temperature can be set as well. The temperature is a value between zero and two that determines how the output should behave. Higher values make the output more random, while lower values lead to a more focused response. A temperature of 0.1 was chosen for the assistant to obtain the most focused and consistent responses, making processing of the messages for converting easier. However, it was not tested to determine which temperature yields the best results, as this could become time-consuming and costly. Each assistant receives its own ID, which can be used to address it. Once OLSAI has been created, this ID will be used to access the assistant instead.

In addition to the client and the assistant, the thread is also needed to store the messages and will be created (3.18).

```
thread = client.beta.threads.create()
```
(3.18)

This can also be easily created and receives its own ID. With the available IDs of the assistant and the thread, it can be precisely defined which assistant and which thread should be used for further methods, such as the Run.

To perform the run, messages must first be added to the thread so that a response can be generated. When creating the message, the thread ID, role, and content must be specified (3.19).

```
message = client.beta.threads.message.create(
    thread_id, role, content, attachments)
```
(3.19)

The role is `"user"` and refers to the learner. In addition, the file ID must be specified under attachments in the message and the use of the `code_interpreter` tool must be specified. The content includes the prompt, which was already discussed in Chapter 3.3.2. This process remains the same for each prompt. Only prompt (3.4) was slightly modified. It was additionally specified that the table of descriptive statistics should be output as a Markdown table and without additional sentences to avoid issues when converting to HTML.

Now, after creating the message, the run is started.

```
run = client.beta.threads.runs.create_and_poll(
        thread_id, assistant_id)                                    (3.20)
response = client.beta.threads.messages.list(thread_id=thread.id)
                                                                    (3.21)
```

The thread ID and assistant ID has to be specified in the run (3.20). After the run is completed, the latest messages from the thread can be listed and are saved as the response (3.21). From this list, the specific responses to the prompts can then be extracted by specifying the element of the list in `data[]`.

```
only_text = response.data[0].content[0].text.value                  (3.22)
```

```
image_id = response.data[0].content[0].image_file.file_id           (3.23)
first_text = response.data[0].content[1].text.value                 (3.24)
second_text = response.data[1].content[0].text.value                (3.25)
```

As long as no output of images is expected, which is predictable based on the used prompt, the first element of the list can be used as the response to the prompt. From this element, the text value must then be filtered out and saved as a new variable, so that the responses can later be used to create the HTML file (3.22). If the used prompt is expected to produce an image, the elements must be filtered differently, as the output is no longer just a single text. By examining the output, it was found that the response to the prompt is stored in the second-to-last and last elements of the output. The first element of the response contains both the latest message and the last image (3.23, 3.24). For images, instead of the text, the file ID of the image must be identified, which should not be confused with the file ID of the dataset. The text written before the image can be extracted from the second-to-last message (3.25). When creating the plots for high-leverage points, this is the only time two plots are generated. In this case, both file IDs can be directly extracted from the last message. The image can then be downloaded using the file ID of the image. The second element of the response contains the text before the images in the output and must be extracted. Each response is then saved as a new variable to create the HTML file later.

By using this approach, the responses and images are extracted for each prompt within the `Python` script. The text outputs are not properly formatted yet. Mathematical formulas in the output are not correctly displayed. They are represented by \[, \], \(and \) in ChatGPT. However, since Markdown is to be used for formatting, a backslash is added before each of these symbols. Additionally, the Markdown table needs to be formatted correctly so that it can be converted to HTML by using a function that converts the Markdown tables of the output to HTML. Once all outputs are correctly formatted in Markdown, the texts can be converted to HTML using the `markdown2` package. The formatted variables are then simply inserted into an HTML script and saved. This way, the user of the `Python` script receives an HTML file containing the AI-generated output for the `cacao.csv` dataset.

3.5.4. The HTML Output of the Code

After the code was executed, the user receives an output in HTML format for the dataset `cacao.csv`. Messages were created with the prompts for this purpose, and a run was performed. The results were then filtered, formatted into HTML format accordingly, and the HTML document was created with them. For illustration purposes, example excerpts of the output are shown in this chapter. The complete output can be viewed in the GitHub repository (Statsomat/OLSAI) (see Figure B.1) under the folder `python_code` (see Figure B.3), which is located in the `Code` folder (see Figure B.2), as `ai_output.html` [14].

What are Descriptive Statistics?

Descriptive statistics are a set of brief descriptive coefficients that summarize a given data set, which can be either a representation of the entire population or a sample of a population. These statistics are broken down into measures of central tendency and measures of variability (spread).

1. **Measures of Central Tendency**: These are used to describe the center of a data set. The most common measures are:
   - **Mean**: The average of all data points.
   - **Median**: The middle value when the data points are arranged in ascending order.
   - **Mode**: The most frequently occurring value in the data set.
2. **Measures of Variability (Spread)**: These describe the spread or dispersion within a data set. Common measures include:
   - **Range**: The difference between the highest and lowest values.
   - **Variance**: The average of the squared differences from the mean.
   - **Standard Deviation**: The square root of the variance, representing the average distance of each data point from the mean.
   - **Interquartile Range (IQR)**: The range within which the central 50% of the data points lie, calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1).
3. **Other Descriptive Statistics**:
   - **Skewness**: A measure of the asymmetry of the distribution of values.
   - **Kurtosis**: A measure of the "tailedness" of the distribution.

Descriptive statistics provide simple summaries about the sample and the measures. They form the basis of virtually every quantitative analysis of data.

The Table of Descriptive Statistics

	ant_exclusion	stem_diameter	height	canopy	dw_healthy	dw_infect	dw_total	fw_pulb	fw_seeds	fw_total	ab_fl_op	ab_fl_cl	ab_fl
count	120	120	120	120	120	120	120	120	120	120	120	120	120
mean	0.5	27.0966	293	0.329775	1054.78	149.55	1203.53	16239.7	5404.45	21617.6	1550.13	3910.92	5452.61
std	0.502096	5.30599	34.6056	0.173708	750.21	146.953	824.982	10796.7	3541.42	14236.2	904.836	2230.83	3106.13
min	0	15.7375	223.75	0.015	0	0	0	0	0	0	241	741	1114
25%	0	23.375	270.375	0.18125	447.25	46.75	573.25	8083.5	2824.25	11107.2	905.25	2277.5	3178.25
50%	0.5	26.7125	287.875	0.344167	939.5	116	1135	14767	4983	19292	1360	3440.5	4798
75%	1	30.3656	313.188	0.449167	1536	216.5	1717.5	21869.2	7544.5	29008.2	2018.75	5053.25	7006.75
max	1	46.6	399.5	0.708333	3045	920	3500	60787	17025	77812	4369	12469	16501

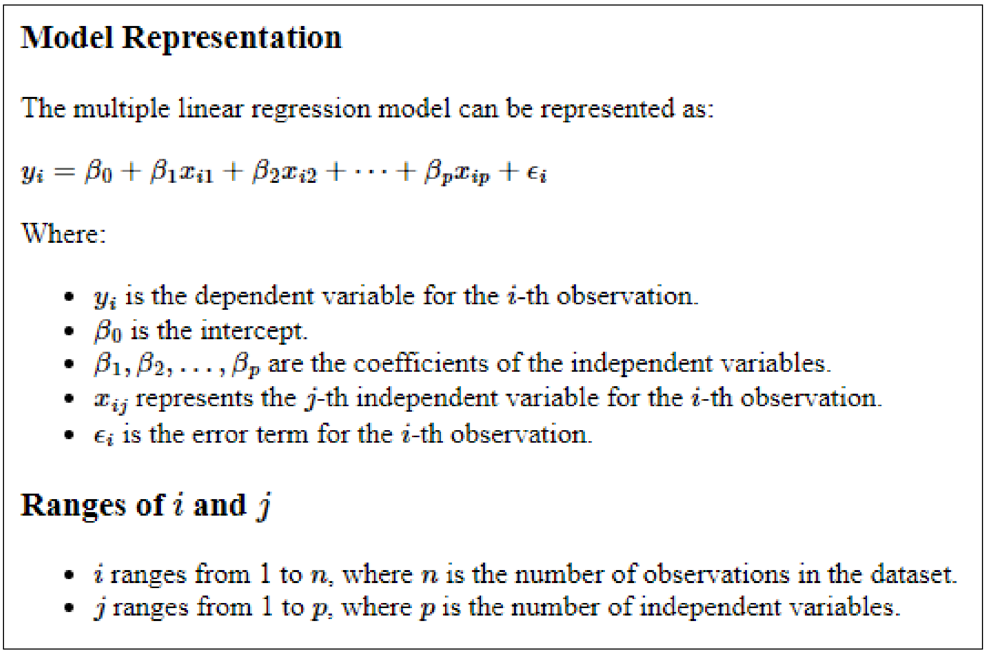
The summary statistics provided in the table give us a snapshot of the dataset's characteristics. Here's a breakdown of what each row represents:

1. **Count**: This indicates the number of observations in each column. In this dataset, each column has 120 observations.
2. **Mean**: The average value of each column. It provides a central value for the data. For example, the mean of the `stem_diameter` is approximately 27.10.
3. **Standard Deviation (std)**: This measures the amount of variation or dispersion in the data. A higher standard deviation indicates more spread out data. For instance, the `fw_pulb` has a standard deviation of about 10796.7, indicating a wide range of values.
4. **Minimum (min)**: The smallest value in each column. For example, the minimum `height` is 223.75.
5. **25th Percentile (25%)**: Also known as the first quartile, it indicates that 25% of the data points are below this value. For example, 25% of the `canopy` values are below 0.18125.
6. **50th Percentile (50%)**: Also known as the median, it divides the data into two equal halves. For example, the median `dw_healthy` is 939.5.
7. **75th Percentile (75%)**: Also known as the third quartile, it indicates that 75% of the data points are below this value. For example, 75% of the `fw_seeds` values are below 7544.5.
8. **Maximum (max)**: The largest value in each column. For example, the maximum `ab_fl` is 16501.

These statistics help us understand the distribution, central tendency, and variability of the data, providing insights into the dataset's overall structure.

Figure 3.25.: Screenshot of Descriptive Statistics in the Output

Figure 3.25 shows a screenshot of the first part of the output. In order to make this display possible, various aspects had to be considered during its creation. The headings "What are Descriptive Statistics" and "The Table of Descriptive Statistics" are headings defined in the HTML script. The rest of the text forms the output from ChatGPT, or rather the responses from the executed run. It can be seen that the extracted Markdown text of the run had to be formatted into HTML format so that, for example, lists or formatting of texts are displayed correctly. Without the prior formatting from Markdown to HTML, it would not be possible to display the text in this form. It can also be seen that the table, which was originally output in Markdown format, had to be formatted. The delimiters of the Markdown table had to be replaced with the correct HTML table tags to ensure proper display. Without this formatting, the display as shown in Figure 3.25 would not be possible.



Model Representation

The multiple linear regression model can be represented as:

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$

Where:

- y_i is the dependent variable for the i-th observation.
- β_0 is the intercept.
- $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients of the independent variables.
- x_{ij} represents the j-th independent variable for the i-th observation.
- ϵ_i is the error term for the i-th observation.

Ranges of i and j

- i ranges from 1 to n, where n is the number of observations in the dataset.
- j ranges from 1 to p, where p is the number of independent variables.

Figure 3.26.: Screenshot of Model Representation in the Output

Another aspect that had to be considered to ensure the correct display of the HTML output is the mathematical formulas, as shown in Figure 3.26. As already mentioned the Markdown text includes the characters \[, \], \(and \) for mathematical formulas. A backslash had to be placed before individual characters for the correct formatting. Now the formatting is correct, but the proper display is only achieved through the use of MathJax, a `JavaScript` display engine for mathematics [6]. This ultimately makes it possible to correctly display mathematical formulas, such as the model equation.

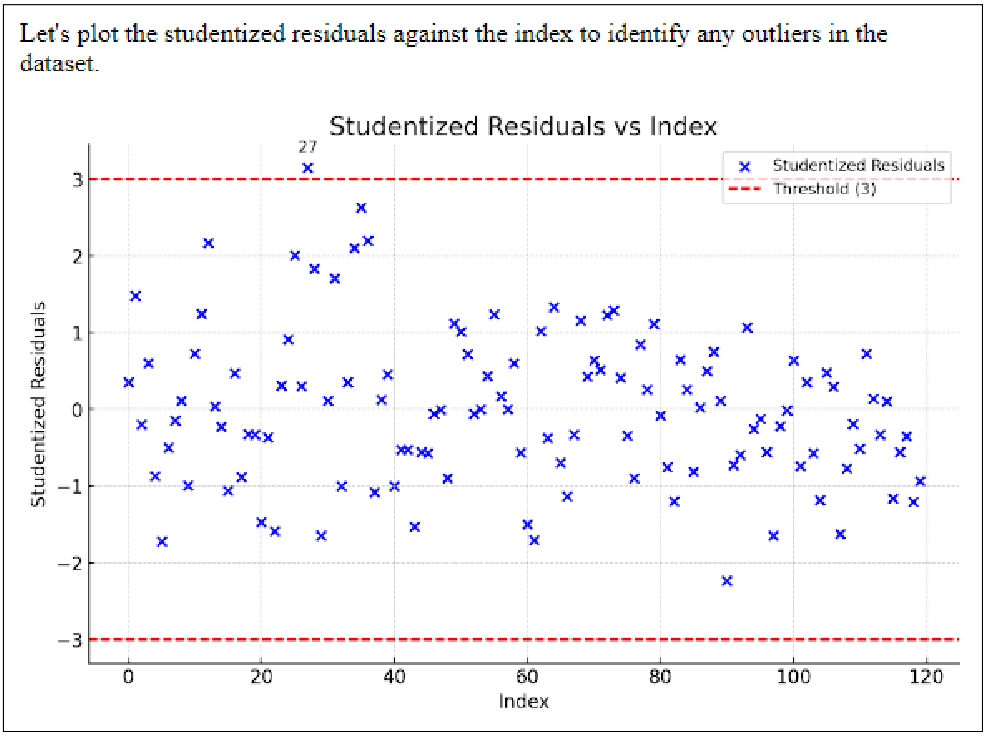


Figure 3.27.: Screenshot of Plot in the Output

Finally, the last component to complete the output is still missing. The created plots had to be extracted and inserted. Using the file ID, the plot could be extracted and downloaded from the run. These downloaded plots were saved as PNG files and can be used in the HTML document. The screenshot in Figure 3.27 shows how the plots are finally integrated into the completed HTML document.

Overall, it becomes clear that several aspects had to be considered in the HTML document generated by the `Python` code. Certain headings had to be predefined to follow a structured flow throughout the report. Additionally, the texts had to be properly formatted to ensure that the output, as shown in the screenshots, was presented clearly and without display errors. Specific formatting is required, especially for mathematical formulas. Images are integrated into the HTML output after extraction and downloading, ensuring correct display. This way, the output presents texts, formulas, and images correctly. Finally, it was now possible to write a `Python` code that automatically generates the output for the regression diagnostics. The statistical methods in the output are correctly and fully implemented, so the output could potentially help learners. This is supported by a learner-friendly structure. The foundation for OLSAI has been established with this.

4. Technical Background

4.1. What is ChatGPT?

ChatGPT is a Large Language Model (LLM) developed by OpenAI that interacts with users through conversation. LLM are deep neural networks that can understand natural language and generate new texts. These models are based on the Transformer architecture invented by Google which enables the ability to train these models on massive amounts of data [29].

GPT in ChatGPT stands for "Generative Pre-trained Transformer," the architecture on which ChatGPT is based [2]. The model was trained using various text data, including publicly available texts such as websites, books, and articles. Furthermore, OpenAI has licensed datasets from third parties, which are also used for training. Additionally, OpenAI collects user data during interactions to improve the model, but the company has taken steps to protect user privacy. Through this large amount of data, the model was able to learn how words are related in specific contexts and how typical sequences of the training data are arranged. ChatGPT can use the knowledge it has learned to determine which response fits the question or input of the user. Users enter a message, which is called a prompt, into ChatGPT and receive a response based on their message. The input can include various types of data, such as text, images, or even datasets. In contrast, the responses from ChatGPT can include text, lists, numbers, or even code snippets, for example. The user then receives a response in natural language from ChatGPT [18].

4.2. The Capabilities of ChatGPT

Interacting with ChatGPT allows the user to use many different capabilities of the model. With the various possibilities, ChatGPT can handle a wide range of tasks.

4.2.1. Overview of Capabilities

The latest model, GPT-4o, was trained with data up until October 2023 [19]. Therefore, ChatGPT does not have more recent information from the internet. However, ChatGPT can search the internet using Bing through web browsing to provide more recent information in response to a user's request. Additionally, it is possible for users to upload images, which can then be analyzed by the GPT Vision model. The information from the image can be used to provide answers and establish further connections. Furthermore, images can also be generated using the DALL-E tool. The user specifies what should be depicted in the image, and the model attempts to create a picture based on these instructions. The user also has the option to upload various documents, such as Word, PowerPoint, PDF, or TXT files, from which ChatGPT can extract and analyze the text. This allows for a wide range of questions regarding the content of the documents. ChatGPT also offers the ability to read the answers aloud. In the mobile app, it is possible to speak with ChatGPT instead of just typing messages [16].

4.2.2. What is Advanced Data Analysis?

Advanced Data Analysis (ADA), formerly known as Code Interpeter is another tool provided by ChatGPT, which requires the user to have a paid plan. This allows the user to upload datasets, which ChatGPT can then analyze. ChatGPT can create various tables and charts based on these datasets. Either ChatGPT selects the charts that best fit the data, or they are specified by the user in the prompt. In addition to static charts, there are also interactive charts available. Currently, only bar, pie, scatter, and line charts are supported. However, the data must be in specific file formats. Currently, only Excel, comma-separated values (CSV), PDF, and JSON files are supported. Additionally, the file size must not exceed 512 MB, with CSV files or spreadsheets being limited to 50 MB. To make this tool possible, the GPT-4 model was trained to perform various data analysis tasks. Through questions and their related code, ChatGPT can generate new code for new datasets. The code is executed in a secure code execution environment. In this environment, a wide range of Python libraries are loaded, which are needed for various data analyses. The ability to understand the user's question about the dataset, perform data analyses in Python and interpret the results makes this tool a powerful resource in data analysis [17].

4.3. What is Prompt Engineering?

Prompt engineering deals with the question of how to craft prompts to achieve the best results. The entered prompt significantly influences the results of a LLM and therefore has to be crafted carefully to achieve the desired outcomes. OpenAI therefore suggests various strategies for building prompts to achieve better results. The first strategy is to clearly specify what the user wants from the model. For example, it can be helpful to indicate the desired format of the output or whether the answer should be brief or detailed. Another strategy involves providing references that ChatGPT should rely on. This allows ChatGPT to extract answers from the reference text or include citations from it. A potential issue is that the request might be too complex. It is therefore recommended to break down complex inputs into simpler tasks. By dividing the request, the error rate is reduced, leading to better results. It can also be beneficial, for certain tasks, to instruct the model to complete one task before proceeding to the next. This approach ensures that tasks requiring more time yield better results, as the model doesn't prematurely conclude a task to move on. Despite the large amount of training data, the model still has some weaknesses. It can be helpful to compensate these weaknesses by using the output from other tools. OpenAI's Code Interpreter can handle mathematical operations and code, but for other tasks, external tools may be beneficial. Specifying these tools in the prompt can lead to better results. When using different strategies and prompts, it can be helpful to check the model's performance, which can also be specified in the prompt. This ensures that the prompt not only performs better in a few cases but also yields better results overall [21].

All of these strategies can be useful in finding the prompt that delivers the desired outcome. The prompt must include all necessary information for ChatGPT to produce a good result. This process of prompt engineering can be quite lengthy due to the variety of strategies, which do not guarantee success but offer suggestions. To achieve a result that satisfies the user, an experimental search for the optimal prompt, in combination with the mentioned strategies, is recommended [21].

4.4. What is an Application Programming Interface?

An Application Programming Interface (API) is a collection of definitions and protocols that allow two different software components to interact and communicate with each other. The software that wants to use the API is called the client. The client sends requests to the server, which then provides the response to the request. To use the API, the client must obtain an API key from the provider. Once the client is set up with this API key, it can utilize various functions to communicate with the server and receive responses to its requests. [1]

4.5. What is GitHub?

GitHub is used to collaborate with other programmers. It offers a cloud-based platform to store and discuss code. The code or other files are stored in a repository, from which they can be discussed and managed. GitHub is based on the open-source software Git, a version control system that allows changes in files to be tracked. The files on GitHub are stored in a Git repository, which is why Git can track changes [11].

5. The Learning Content

5.1. What is Multiple Linear Regression?

The famous statistician Georg Edward Pelham Box once said: "Essentially, all models are wrong, but some are useful" [4]. In statistics, there are various approaches to modeling in order to represent a specific relationship within the given data. However, not every model is useful for every dataset. Some models might fit the data well, whereas others might distort the interpretation and results a lot. A very famous model for modeling a linear relationship between variables is (multiple) linear regression, which is the basis for many other regression models [9].

The multiple linear regression tries to explain the *dependent* variable y_i with one or more *independent* variables $x_{i1}, \ldots, x_{ik}$, where i is the i^{th} observation $(i = 1, \ldots, n)$ and k is the number of independent variables. The model can generally be represented by the formula:

$$y_i = \beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} + \cdots + \beta_k \cdot x_{ik} + \varepsilon_i \tag{5.1}$$

The $\beta_1, ..., \beta_k$ are called *regression coefficients* and β_0 is the *intercept*. The random *error term* ε has to be added, since the relationship between the independent and the dependent variables is not exact. Therefore, y is a random variable. Additionally, this formula can also be expressed in matrix notation. The *design matrix* $\mathbf{X}$ consists of the values of the dependent variables x_{ik}, whereas the vector $\mathbf{y}$ includes the values of the dependent variable y_i. The vector β includes the intercept and the different regression coefficients, the vector ε contains the different error terms.

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix} : \text{matrix of independent variables}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} : \text{ vector of the dependent variable}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_n \end{pmatrix} : \text{ vector of intercept and regression coefficients}$$

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix} : \text{ vector of error terms}$$

The formula (5.1) can now be represented in matrix notation using those matrices and vectors:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The intercept and the different regression coefficients are unknown and therefore need to be estimated [9].

The multiple linear regression model serves two main purposes. The first purpose is prediction, where the model is used to predict values based on the independent variables. The second purpose is inference, where the model helps to understand the relationships between variables, test hypotheses and draw conclusions about the underlying population [9].

5.1.1. Assumptions of the Model

In order to get efficient estimators, the linear regression model must follow certain assumptions, which are:

$$E(\boldsymbol{\varepsilon}) = 0 \tag{5.2}$$

$$Cov(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I} \tag{5.3}$$

$$rank(\mathbf{X}) = k + 1 = p \tag{5.4}$$

$$\boldsymbol{\varepsilon} \sim N(0, \sigma^2\mathbf{I}) \tag{5.5}$$

where **I** is the identity matrix. If those assumptions are met, it can be assumed that the model fits the data well. If not, many statistical tests can be biased and wrong conclusion could be made. Therefore, they should be verified using so called regression diagnostics [9].

5.2. What are Regression Diagnostics?

Regression diagnostics are methods to check for the assumptions of a regression model. They contain different statistical methods and graphical tools to identify any violation of the assumptions numerated in (5.1.1). However, there are various methods that can be used for regression diagnostics. In the development of OLSAI, specific methods were focused on.

Diagnostic Tool / Category	**Statistical Method**	**Graphic**
Outliers	Studentized Residuals	Studentized Residuals vs. Index
High-Leverage	Cook's Distance	Leverage vs. Index
Non-Linearity	Rainbow Test	Studentized Residuals vs. $\hat{y}_i$
Heteroscedasticity	Breusch-Pangan Test	Square Root of Studentized Residuals vs. $\hat{y}_i$ or x_{ij}
Correlation of Error Terms	Durbin-Watson Test	Residuals vs. Index
Non-Normality of Residuals	Shapiro-Wilk Test	QQ-Plot Histogram of Residuals
Collinearity of Predictors	Variance Inflation Factor	Correlation Matrix

Table 5.1.: Summary of Diagnostic Tools

The table 5.1 lists chosen statistical methods and graphical tools for each diagnostic category, which will be explained in the following sections. An extended table with thresholds can be found in the attachment Table A.1.

5.2.1. Outliers

Fahrmeir offers an approach to identify outliers, which will be discussed in this section. As he states, there is no accurate definition for outliers, but it explains outliers as observations that differ greatly from the expected value of the model. Such points can cause distortion of the regression coefficients and therefore lead to wrong results and interpretation [9].

One attempt to detect outliers is to look for high residuals. For this context studentized residuals are used:

$$r_i^* = \frac{\hat{\varepsilon}_{(i)}}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}} \sim t_{n-p-1} \tag{5.6}$$

with h_{ii} as the leverage, which will be analyzed in the next section, and $\hat{\sigma}_{(i)}$ as the estimated variance without the i^{th} observation. Generally (i) means, that the i^{th} observations has been left out of the estimation of the parameters of the model. Since the studentized resiudals are t-distributed specific cut-off values can be calculated to identify high residuals. Values that are considered as outliers are

$$r_i^* > t_{1-\frac{\alpha}{2},n-p-1} \qquad \text{or} \qquad r_i^* < t_{\frac{\alpha}{2},n-p-1}$$

A rule of thumb for the cut-off values is ± 3 [26]. With this method, the outliers can be displayed graphically by plotting the studentized residuals against their indices. Values smaller than -3 or bigger than 3 are considered as outliers [9].

5.2.2. High-Leverage Points

High-leverage points and methods for identification are also explained by Fahrmeir and will be displayed here. Observations with a high leverage h_{ii} can have significant impact on the estimated regression coefficients $\hat{\beta}$ and therefore on the estimated $\hat{y}$. Leverages h_{ii} are defined as the diagonal elements of the *prediction matrix* H with

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \quad \text{with} \quad \frac{1}{n} \leq h_{ii} \leq 1$$

where observations with $h_{ii} > \frac{2p}{n}$ are highly noticeable, but not all high-leverage points are problematic. Here, $\frac{p}{n}$ is the mean leverage value, which is the reason for this rule being two times bigger than $\overline{h}$ [27].

$$\overline{h} = \frac{\sum_{i=1}^{n} h_{ii}}{n} = \frac{k+1}{n} = \frac{p}{n}$$

Those leverages can be plotted against their indeces in order to get a graphical assistance method. A statistical method to identify high-leverage points is Cook's distance which is defined as

$$D_i = \frac{(\hat{y}_{(i)} - \hat{y})'(\hat{y}_{(i)} - \hat{y})}{p \cdot \hat{\sigma}^2}$$

where $\frac{4}{n}$ is used as a cut-off value [7]. But there does not exist a fixed cut-off value, so as a rule of thumb observations with $D_i > 1$ can be considered as highly noticeable [9].

5.2.3. Non-Linearity

One assumption made in the model is that the independent variable has a linear relationship with the dependent variables. Even though the relationship of the variables might be non-linear, there could be a linear relationship within certain subsets. This can be examined with the Rainbow test which now will be explained [28].

The concept of the test is that a good linear fit can be achieved over subsets of the data, even if the true relationship is non-linear [3]. If the linear fit of the subset is significantly better than the fit of the whole dataset, the null hypothesis would be rejected. This result would suggest that the full model would not follow a linear relationship.

$$H_0 : \text{Linear relationship between variables}$$
$$H_1 : \text{No linear relationship between variables}$$

Under the null hypothesis the test statistic F is F_{n-n_1,n_1-p} distributed with

$$F = \frac{SSE_N}{n-n_1} / \frac{SSE_D}{n_1 - p}$$
$$SSE_N = y'[I - X(X'X)^{-1}X' - D + DX(X'DX)^{-1}X'D]y$$
$$SSE_D = y'(D - DX(X'DX)^{-1})y$$

where $\mathbf{D}$ is a diagonal matrix which consists of zeros everywhere except for the middle

n_1 $(p < n_1 < n)$ diagonal elements, where the value is one. This matrix represents the selection of the subset of $\mathbf{X}$ [28].

A graphical method to check for non-linearity is to plot the studentized residuals (5.6) against the estimated values $\hat{y}_i$. The residuals should be randomly distributed around the zero line if the assumption is not violated [10].

5.2.4. Heteroscedasticity

The assumption 5.5 implies that the variance of the error terms is constant, which is called homoscedasticity. If the variance of the error terms is not constant, it is referred to as heteroscedasticity. A statistical test for heteroscedacity was brought by Breusch and Pangan [5].

The variance of the error terms can be described by the multiplicative model

$$\sigma_i^2 = \sigma^2 \cdot h(\alpha_0 + \alpha_1 z_{i1} + \cdots + \alpha_q z_{iq})$$

where $z_1, \ldots, z_q$ are dependent variables that might have an impact on the variance. The function h is independent from i. This leads to the hypotheses

$$\begin{aligned} &H_0 : \alpha_0 = \cdots = \alpha_q \text{ (homoscedastic variances)} \\ &H_1 : \exists j : \alpha_j \neq 0 \text{ (heteroscedastic variances)} \end{aligned}$$

This test needs another regression which is used to build the test statistic. The regression uses the error terms and the maximum likelihood estimate of the variance:

$$g_i = \frac{\hat{\varepsilon}_i^2}{\hat{\sigma}_{ML}^2}$$

Therefore, the test statistic is as follows:

$$T = \frac{1}{2} \sum_{i=1}^{n} (\hat{g}_i - \overline{g})^2 \sim \chi_q^2$$

Since T is χ_q^2 distributed, the null hypothesis can be rejected if T is bigger than $\chi_{q,1-\alpha}^2$ [9].

Besides the statistical test a graphical method for the assessment of homoscedasticity is recommended. In order to identify homoscedasticity residual plots are used. The (squareroot of) studentized residuals (5.6) are plotted against the estimated values $\hat{y}_i$ or the independent variables x_{ij}. This plot is called a scale-location plot. The residuals should be randomly scattered around zero with constant variance when homoscedasticity is met [9].

5.2.5. Correlation of Error Terms

The problem of correlation of error terms is also known as autocorrelation. In the linear regression model, the error terms should not be correlated. Fahrmeir provides methods to identify autocorrelation which will be presented in this section. The given statistical test to detect autocorrelation is the Durbin-Watson test, where a model with correlated error terms is assumed. This provides the null hypothesis of the correlation ρ being zero in order to test the alternative hypothesis of the correlation not being equal to zero.

$$H_0 : \rho = 0$$
$$H_1 : \rho \neq 0$$
$$d = \frac{\sum_{i=2}^{n}(\hat{\varepsilon}_i - \hat{\varepsilon}_{i-1})^2}{\sum_{i=2}^{n}\hat{\varepsilon}_i^2} \approx 2(1-\hat{\rho}) \text{ for large n}$$

The acceptance region for the null hypothesis is dependent from d_o, whereas the rejection region for is dependent from d_u. Those values can be looked up on a table for different sample sizes n and numbers of dependent variables p.

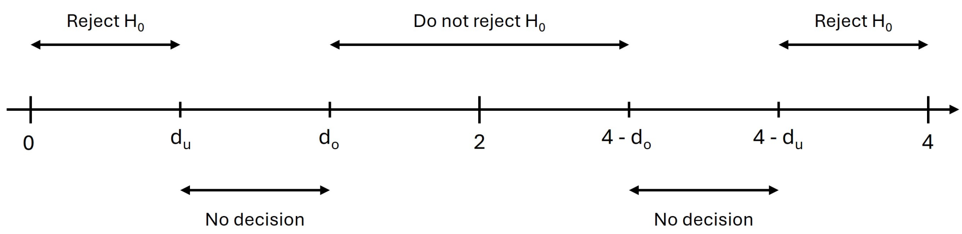


Figure 5.1.: Acceptance and Rejection Regions of the Durbin-Watson Test [9].

The scale of the of d reaches from 0 to 4 since $\hat{\rho}$ can only take values from -1 to 1. If d is between 0 and d_u or between $4 - d_u$ and 4 then the null hypothesis can be rejected. Only if the value of d is between d_u and $4 - d_o$ the null hypothesis can not be rejected. If the value d is not within any of these intervalls, it leads to no decision of the Durbin-Watson test [9].

A graphical method to check for autocorrelation is to plot the studentized residuals against the time. In this plot, no pattern should be discernible. Certain pattern can point to autocorrelation, such as a descending or increasing trend in the residuals [9].

5.2.6. Normality of Residuals

The assumption 5.5 states that the errors should be normal distributed. A statistical test that checks for normality of the residuals is the Shapiro-Wilk test which will be explained here. The test checks for the hypothesis that the data are derived from a normal distribution followed by the test statistic W:

$$H_0 : \text{Data from normal distribution}$$
$$H_1 : \text{Data not from normal distribution}$$
$$W = \frac{\left(\sum_{i=1}^{n} a_i x_i\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad \text{with} \quad \mathbf{a}' = (a_1, \ldots, a_n) = \frac{\mathbf{m}'\mathbf{V}^{-1}}{\sqrt{\mathbf{m}'\mathbf{V}^{-1}\mathbf{V}^{-1}\mathbf{m}}}$$

where $\mathbf{x}$ is a vector of ordered observations, $\mathbf{m}$ is the vector of expected values of a standard normal order statistics and $\mathbf{V}$ is the corresponding covariance matrix. If the data are derived from a normal distribution, W should be close to one. The test statistics can be evaluated by the p-value [24].

Two simple graphical ways for the assessment of a normal distribution can be used: The Q-Q plot and a histogram of residuals. For the Q-Q plot the residuals are sorted and the quantiles are computed. Then the quantiles of a normal distribution were computed and those quantiles are plotted against each other. For a normal distribution, the points should be on the bisector. If the points differ from this line, it can be assumed that the residuals are not normally distributed. An alternative option is to make a histogram of the residuals. This histogram should look like a bell curve from a normal distribution, if the residuals are normally distributed [10].

5.2.7. Collinearity of Predictors

The assumption (5.4) implies that the design matrix $\mathbf{X}$ has full rank, which means that the columns of the matrix are linearly independent. This concludes that linear dependency within $\mathbf{X}$ can lead to not uniquely estimable regression coefficients. Collinearity means, that two variables are highly correlated. An approach to check for collinearity is the variance inflation factor:

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2} \tag{5.7}$$

where j is the index of the dependent variable x_j. High correlation of x_j and the other dependent variables results in a high measure of certainty R_j^2. Variance inflation factors above 10 are considered as problematic in terms of collinearity [9].

A graphical method that could aid the visualization of collinearity is a covariance matrix with different ways to portray high correlation. One approach is to use big circles for high correlation and small circles for low correlation, whereas another way of visualizing the covariance matrix is to change the color of the element of the matrix dependent on the value of the correlation [9].

6. Conclusion

6.1. Summary of the Thesis

The main goal of this thesis was to develop an AI assistant that can be further improved in the future. The AI assistant is designed to help learners understand the topic of multiple linear regression, particularly regression diagnostics, by automatically generating an output that explains the topic using a dataset. To achieve this goal, various aspects were considered and analyzed.

In the beginning of this thesis, an analysis of a dataset was conducted by myself to create a foundation for the intended AI output. The analysis focused only on regression diagnostics, enabling an analysis of ChatGPT's output regarding the statistical methods. One major goal was to find a prompt that could generate a learner-friendly output using AI. The AI-generated output should also include descriptive statistics, model explanations, and further detailed explanations of the methods. Based on this foundation, the initial output from ChatGPT was evaluated when a simple prompt was entered. This prompt contained no prior knowledge of the topic and could therefore have come from the learner. It became evident that the output of this prompt was insufficient for a learner in many respects. On one hand, the topics were incompletely covered, and on the other, the structure was far from learner-friendly, characterized by frequent repetitions and a lack of organization. Based on these results, various approaches were tested to determine which prompt would produce the best outcome. The output needed to meet important criteria, particularly by significantly improving the negative aspects of the previous prompt. The result was a set of prompts that together produced a clear, informative, and learner-friendly output. This output was well-structured, had a logical flow and was easy to understand. The learner-friendliness was evaluated by potential learners who received the output and completed a questionnaire about it. No category was rated lower than 'Neutral', both in terms of the median and the mean. Additionally, learners were asked about their views on AI tools designed to assist with learning. Learners would use or are already using AI tools as learning aids. Therefore, it was time to develop

OLSAI, the AI tool designed to help learners understand multiple linear regression, with a primary focus on the regression diagnostics of the model. However, the development of this tool is not completed yet. This thesis has only laid the foundation for its future development. OLSAI was developed in `Python` using OpenAI's API. The API made it possible to obtain ChatGPT's output in `Python` and extract the results. These results were then used to generate the output in HTML format. In summary, by examining the effects of different prompts, a statistically accurate and learner-friendly output was created using a `Python`-programmed AI assistant, based on OpenAI's API and ChatGPT.

6.2. Limitations and Issues

However, there were some issues that need to be addressed here as well. The prompt identified during the investigation does not necessarily represent the best possible solution. It is a solution that can produce the desired output. There is certainly a more detailed prompt that could generate an equally good output. At the current stage of the thesis, the prompt found was fully sufficient for the project. Additionally, the prompt was only tested on the datasets `cacao.csv`, `cacao.csv` without outliers and `Electricity1955.csv`. To obtain more accurate results, a larger number of datasets would need to be tested, which was not possible due to the limited time available in this thesis. Further investigation, however, would be too time-consuming, as an adequate result has already been found. Additionally, pre-programmed `Python` packages were used for statistical methods, which means there is a reliance on the assumption that these contain no errors. Otherwise, the entire theory and the packages would have had to be developed from scratch, which would exceed the scope and intent of this work. There were also challenges in developing OLSAI. When creating an AI assistant, various parameters can be configured, such as the instructions or the temperature. These parameters, especially the temperature, were not further tested to determine which value works best. The main issue here is the cost of the API and the time of generating the output. Due to the complexity of the topic and the number of prompts, the creation process is time-consuming. Additionally, using the API incurs costs. Generating an output, assuming no errors occur, costs approximately one to two Euros and takes about eight minutes. Although it sounds time-consuming, manually creating the output with ChatGPT would take just as long, if not longer. Due to the complexity, even manually, a longer time would be needed to wait for the output. The creation of plots is particularly expensive, making it financially unfeasible to test various settings, as the costs would quickly escalate. Furthermore, the reproducibility of the `Python` code for the AI assistant must be discussed. So far, the reproducibility of the code has not been extensively tested. Therefore, errors may occur if the code is executed by other people. Additionally, during the development, further errors arose that have not yet been addressed in the code. For example, the various run statuses were not considered and no solution has been integrated in case the run status is not 'completed.' Also, due to the controlled output through the prompt, it is known which elements in the run represent which output. The AI assistant's output must have a certain length to determine if no error occurred during generation. If this length is not met, there is currently no integrated solution for what happens next. However, it should be emphasized that the code represents the first version of OLSAI and still needs further development to create a reliable and complete AI tool. The mentioned problems are known and need to be addressed in the further development. For additional issues, an issue was created in the repository (see Figure B.5), where these can be discussed, and comments can be made.

6.3. Outlook

Through the investigation of the prompt, evaluation by learners, and initial programming of the AI assistant, it was possible to generate an AI-produced output for a dataset. In the future, several aspects need to be considered for the further development of the OLSAI tool. The tool needs to be tested on a variety of datasets to ensure error-free usage. Additionally, different parameters must be tested to achieve an improved result. Furthermore, once development is completed, the tool must be integrated into a website so that learners can access it. Additionally, it must be ensured that the tool is not misused for merely analyzing datasets, but rather that the learning aspect remains the focus. To achieve this, OLSAI should be capable of synthesizing a dataset similar to the dataset of the learner, encouraging the learner to apply what they've learned to their own dataset.

In conclusion, the thesis has provided a strong foundation for the further development of the AI tool OLSAI. The results of the investigations have laid the groundwork. For the further development of this tool, the previously mentioned points must be taken into account. However, there is strong confidence that OLSAI will be further developed in a way that future learners will greatly benefit from it.

Bibliography

[1] Amazon Web Services. *What is an API (Application Programming Interface)?* 2024. URL: `https://aws.amazon.com/what-is/api/?nc1=h_ls` (visited on 09/02/2024).

[2] J. An, W. Ding, and C. Lin. "ChatGPT". In: *tackle the growing carbon footprint of generative AI* 615 (2023), p. 586.

[3] B. H. Baltagi. *What Is Econometrics?* Springer, 2011.

[4] G. E. Box. "Robustness in the strategy of scientific model building". In: *Robustness in statistics*. Elsevier, 1979, pp. 201–236.

[5] T. S. Breusch and A. R. Pagan. "A Simple Test for Heteroscedasticity and Random Coefficient Variation". In: *Econometrica* 47.5 (1979), pp. 1287–1294. URL: `http://www.jstor.org/stable/1911963` (visited on 07/17/2024).

[6] D. Cervone. "MathJax: a platform for mathematics on the Web". In: *Notices of the AMS* 59.2 (2012), pp. 312–316.

[7] R. D. Cook. "Detection of influential observation in linear regression". In: *Technometrics* 19.1 (1977), pp. 15–18.

[8] I. Döbel et al. "Maschinelles Lernen–Kompetenzen, Anwendungen und Forschungsbedarf". In: *Fraunhofer IAIS, Fraunhofer IMW, Fraunhofer Zentrale. Zugriff am* 21 (2018), p. 2020.

[9] L. Fahrmeir, T. Kneib, and S. Lang. *Regression*. Springer, 2009.

[10] J. J. Faraway. *Practical Regression and Anova Using R*. Chapman and Hall/CRC, 2002.

[11] GitHub. *About GitHub and Git*. 2024. URL: `https://docs.github.com/en/get-started/start-your-journey/about-github-and-git` (visited on 09/06/2024).

[12] P. Gras et al. "How ants, birds and bats affect crop yield along shade gradients in tropical cacao agroforestry". In: *Journal of Applied Ecology* 53.3 (2016), pp. 953–963.

[13] W. Greene. *Econometric Analysis*. NJ: Prentice Hall, 2003.

[14] P. Leubner and D. Welsch. *OLSAI*. Zenodo. 2024. DOI: 10.5281/zenodo.13691945. URL: https://doi.org/10.5281/zenodo.13691945.

[15] OpenAI. *ChatGPT*. 2024. URL: https://chatgpt.com/ (visited on 09/06/2024).

[16] OpenAI. *ChatGPT Capabilities Overview*. 2024. URL: https://help.openai.com/en/articles/9260256-chatgpt-capabilities-overview (visited on 09/02/2024).

[17] OpenAI. *Data analysis with ChatGPT*. 2024. URL: https://help.openai.com/en/articles/8437071-data-analysis-with-chatgpt#h_595be2b61b (visited on 09/02/2024).

[18] OpenAI. *How ChatGPT and Our Language Models are Developed*. 2024. URL: https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-language-models-are-developed#h_2df02d4917 (visited on 09/01/2024).

[19] OpenAI. *Models*. 2024. URL: https://platform.openai.com/docs/models (visited on 09/02/2024).

[20] OpenAI. *OpenAI Documentation*. 2024. URL: https://platform.openai.com/docs/ (visited on 08/29/2024).

[21] OpenAI. *Prompt Engineering*. 2024. URL: https://platform.openai.com/docs/guides/prompt-engineering (visited on 09/02/2024).

[22] A. Sarirete et al. *Artificial intelligence and machine learning research: towards digital transformation at a global scale*. 2022.

[23] S. Seabold and J. Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[24] S. Shaphiro and M. Wilk. "An analysis of variance test for normality". In: *Biometrika* 52.3 (1965), pp. 591–611.

[25] Statsomat. *Exploratory Data Analysis (Statsomat/Edapy)*. 2024. URL: https://statsomat.com/statsomat/exploratory-data-analysis-eda-with-python/ (visited on 07/30/2024).

[26] The Pennsylvania State University. *Studentized Residuals*. 2018. URL: https://online.stat.psu.edu/stat462/node/247/ (visited on 07/23/2024).

[27] The Pennsylvania State University. *Using Leverages to Help Identify Extreme X Values*. 2018. URL: https://online.stat.psu.edu/stat462/node/171/ (visited on 07/23/2024).

[28] J. M. Utts. “The rainbow test for lack of fit in regression”. In: *Communications in Statistics-Theory and Methods* 11.24 (1982), pp. 2801–2815.

[29] A. Vaswani. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* (2017).

[30] D. Welsch. *OLSAI*. Feb. 2024. DOI: `10.5281/zenodo.10663270`. URL: `https://doi.org/10.5281/zenodo.10663270`.

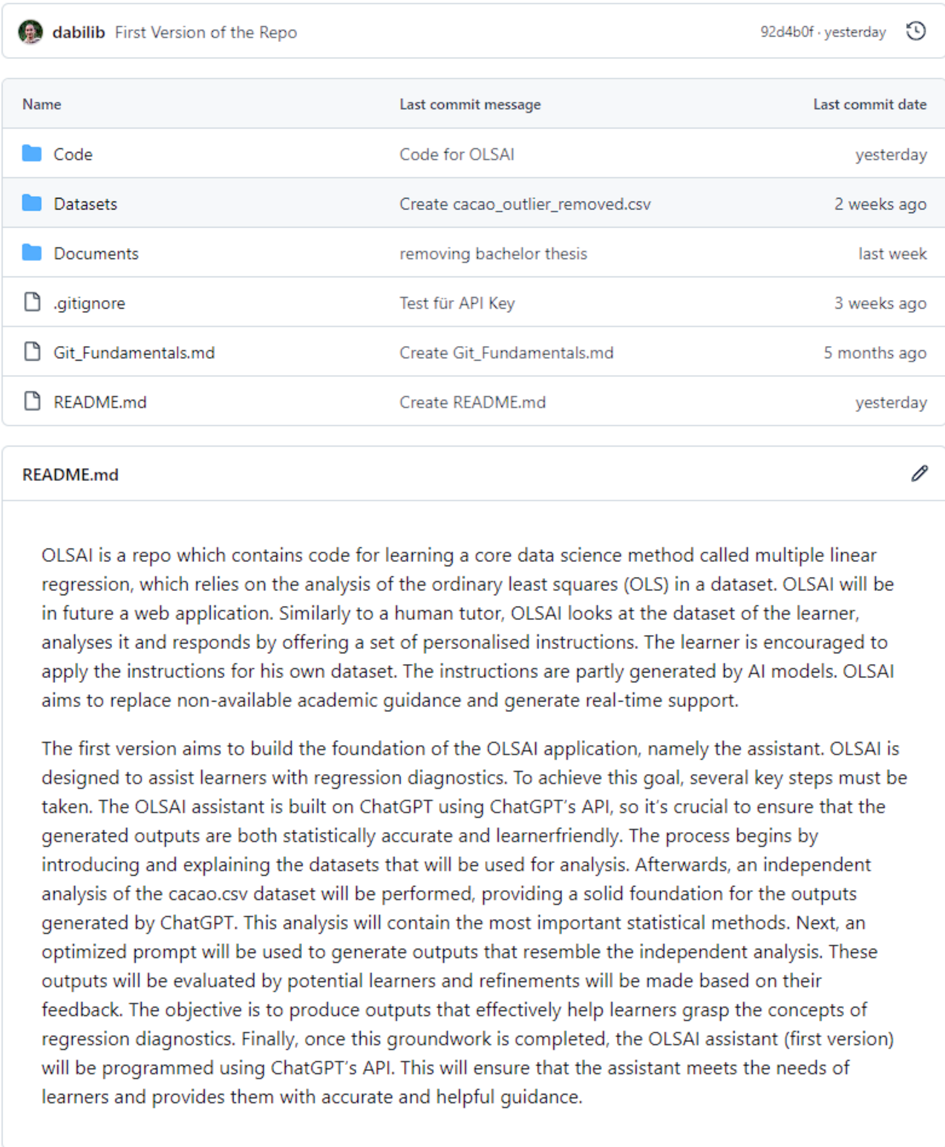
Attachment

A. Additional Tables

Diagnostic Tool	Statistical Method	Threshold	Graphic
Outliers	Studentized Residuals	$r_i^* > t_{1-\frac{\alpha}{2},n-p-1}$ $r_i^* < t_{\frac{\alpha}{2},n-p-1}$ or ± 3	Studentized Residuals vs. Index
High-Leverage	Cook's Distance	$D_i > \frac{4}{n}$	Leverage vs. Index
Non-Linearity	Rainbow Test	significance level	Studentized Residuals vs. $\hat{y}_i$
Heteroscedasticity	Breusch-Pangan Test	significance level	Square Root of Studentized Residuals vs. $\hat{y}_i$ or x_{ij}
Correlation of Error Terms	Durbin-Watson Test	Figure 5.1	Residuals vs. Index
Non-Normality of Residuals	Shapiro-Wilk Test	significance level	QQ-Plot, Histogram of Residuals
Collinearity of Predictors	Variance Inflation Factor	$\mathrm{VIF}_j > 10$	Correlation Matrix

Table A.1.: Diagnostic Tools

B. Additional Graphics



dabilib First Version of the Repo 92d4b0f · yesterday

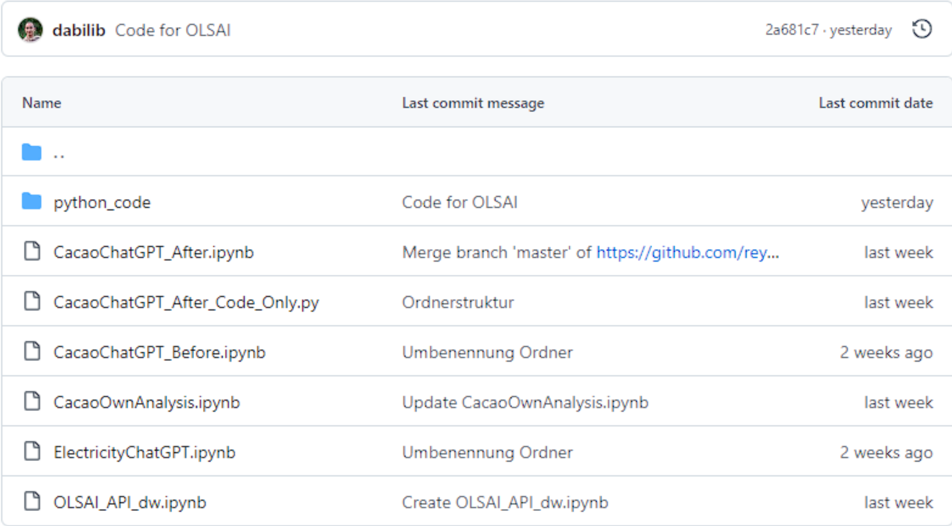
Name	Last commit message	Last commit date
Code	Code for OLSAI	yesterday
Datasets	Create cacao_outlier_removed.csv	2 weeks ago
Documents	removing bachelor thesis	last week
.gitignore	Test für API Key	3 weeks ago
Git_Fundamentals.md	Create Git_Fundamentals.md	5 months ago
README.md	Create README.md	yesterday

README.md

OLSAI is a repo which contains code for learning a core data science method called multiple linear regression, which relies on the analysis of the ordinary least squares (OLS) in a dataset. OLSAI will be in future a web application. Similarly to a human tutor, OLSAI looks at the dataset of the learner, analyses it and responds by offering a set of personalised instructions. The learner is encouraged to apply the instructions for his own dataset. The instructions are partly generated by AI models. OLSAI aims to replace non-available academic guidance and generate real-time support.

The first version aims to build the foundation of the OLSAI application, namely the assistant. OLSAI is designed to assist learners with regression diagnostics. To achieve this goal, several key steps must be taken. The OLSAI assistant is built on ChatGPT using ChatGPT's API, so it's crucial to ensure that the generated outputs are both statistically accurate and learnerfriendly. The process begins by introducing and explaining the datasets that will be used for analysis. Afterwards, an independent analysis of the cacao.csv dataset will be performed, providing a solid foundation for the outputs generated by ChatGPT. This analysis will contain the most important statistical methods. Next, an optimized prompt will be used to generate outputs that resemble the independent analysis. These outputs will be evaluated by potential learners and refinements will be made based on their feedback. The objective is to produce outputs that effectively help learners grasp the concepts of regression diagnostics. Finally, once this groundwork is completed, the OLSAI assistant (first version) will be programmed using ChatGPT's API. This will ensure that the assistant meets the needs of learners and provides them with accurate and helpful guidance.

Figure B.1.: The GitHub Repository Statsomat/OLSAI



dabilib Code for OLSAI 2a681c7 · yesterday

Name	Last commit message	Last commit date
..		
python_code	Code for OLSAI	yesterday
CacaoChatGPT_After.ipynb	Merge branch 'master' of https://github.com/rey...	last week
CacaoChatGPT_After_Code_Only.py	Ordnerstruktur	last week
CacaoChatGPT_Before.ipynb	Umbenennung Ordner	2 weeks ago
CacaoOwnAnalysis.ipynb	Update CacaoOwnAnalysis.ipynb	last week
ElectricityChatGPT.ipynb	Umbenennung Ordner	2 weeks ago
OLSAI_API_dw.ipynb	Create OLSAI_API_dw.ipynb	last week

Figure B.2.: The Folder `Code`

dabilib Code for OLSAI 2a681c7 · yesterday

Name	Last commit message	Last commit date
..		
ai_output.html	Code for OLSAI	yesterday
image11.png	Code for OLSAI	yesterday
image16.png	Code for OLSAI	yesterday
image18.png	Code for OLSAI	yesterday
image18_2.png	Code for OLSAI	yesterday
image20.png	Code for OLSAI	yesterday
image22.png	Code for OLSAI	yesterday
image24.png	Code for OLSAI	yesterday
image26.png	Code for OLSAI	yesterday
image28.png	Code for OLSAI	yesterday
image5.png	Code for OLSAI	yesterday
image7.png	Code for OLSAI	yesterday
image9.png	Code for OLSAI	yesterday
olsai_in_python.py	Code for OLSAI	yesterday

Figure B.3.: The Folder `python_code`

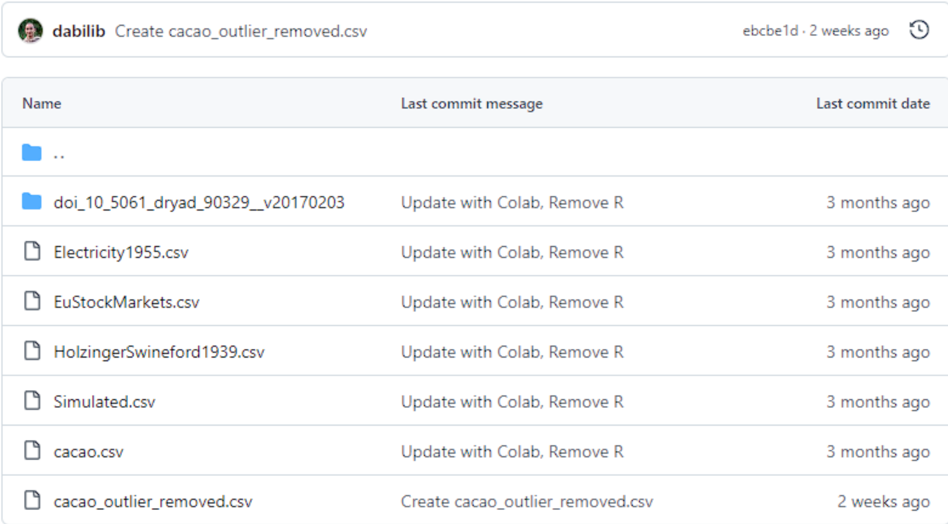


Figure B.4.: The Folder `Datasets`

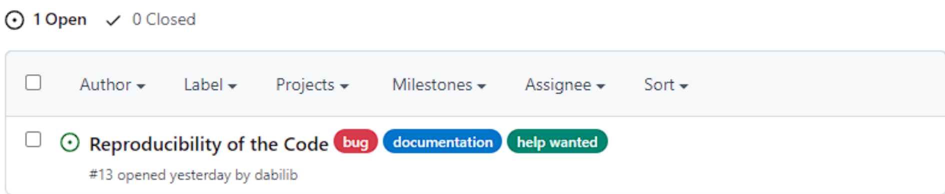


Figure B.5.: The Issues Tab

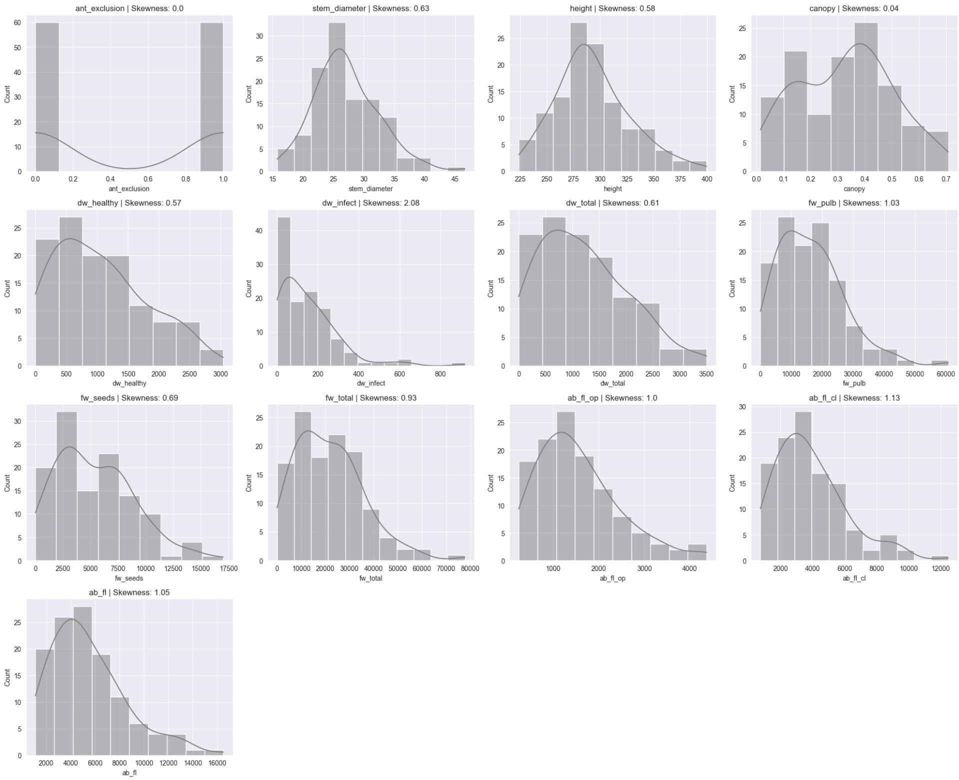


Figure B.6.: Histograms of Each Variable

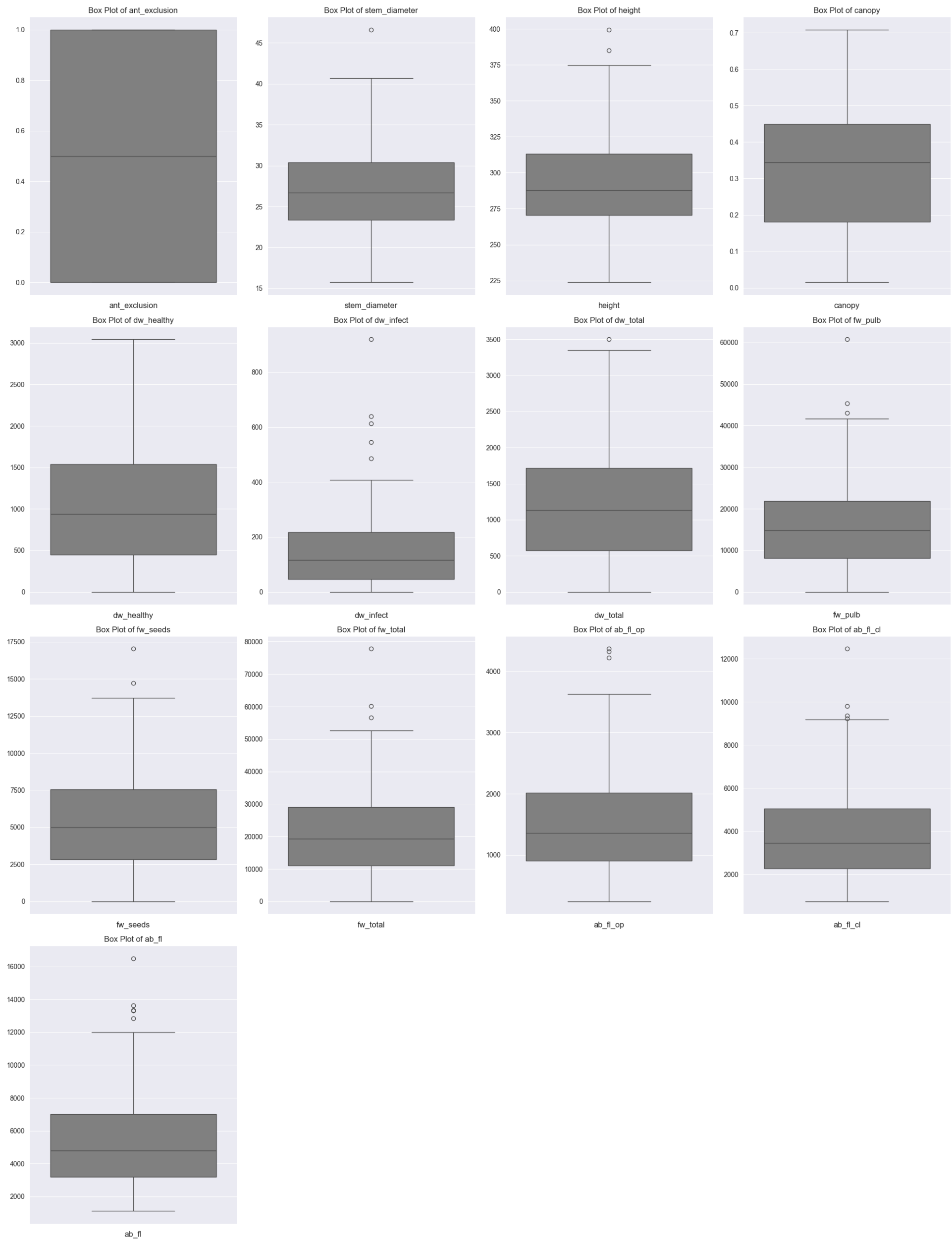


Figure B.7.: Boxplots for Each Variable

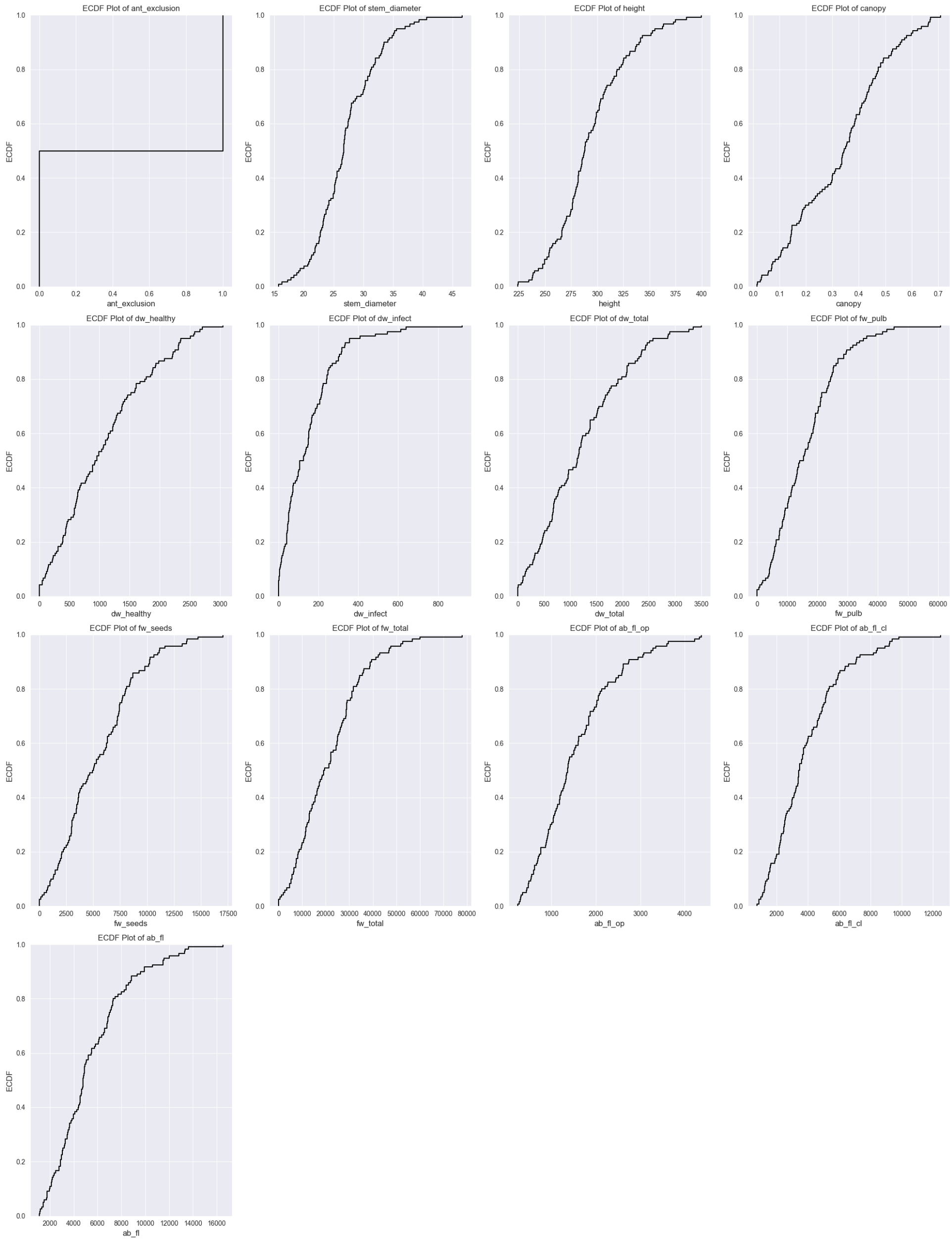


Figure B.8.: ECDF Plots for Each Variable

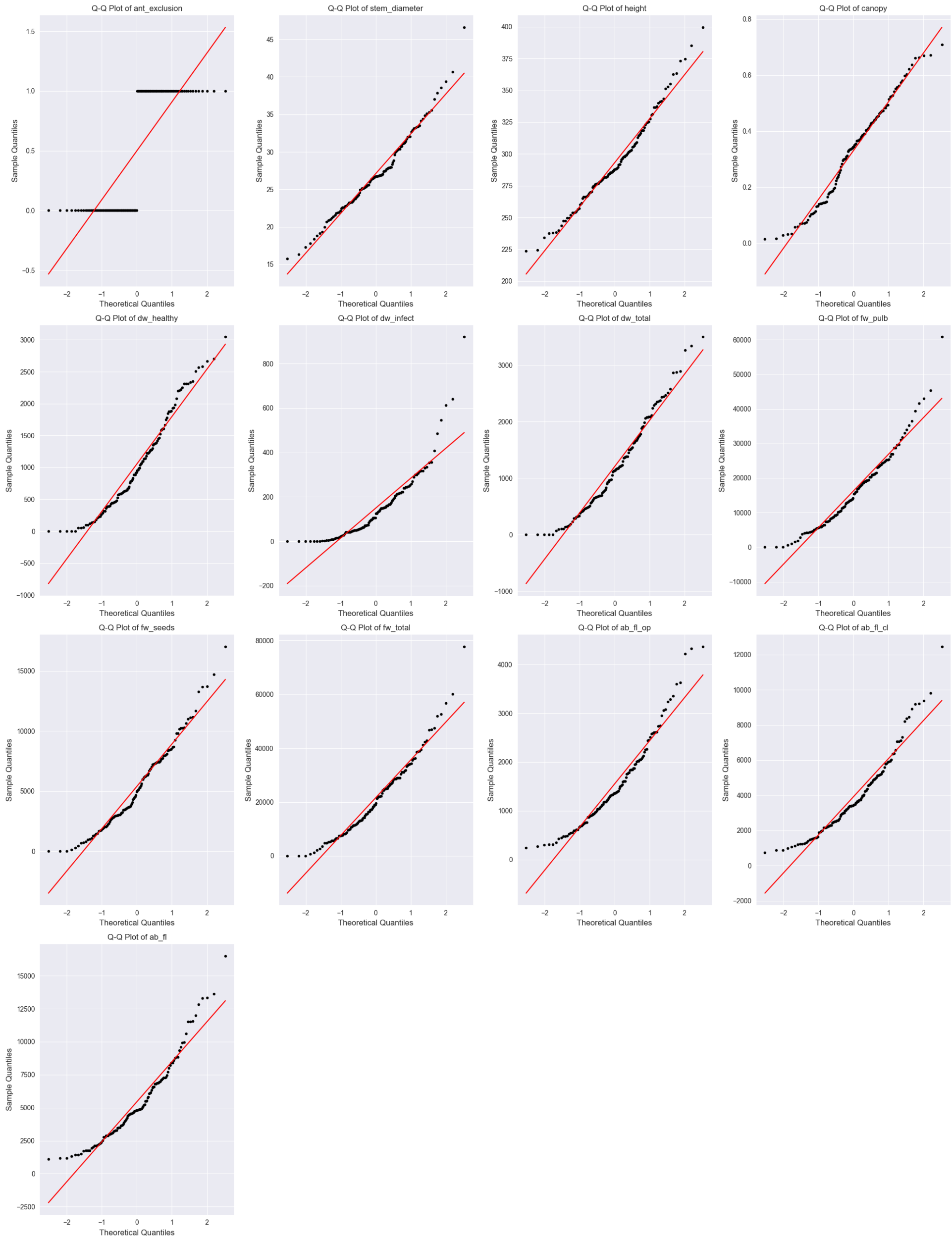


Figure B.9.: QQ Plots for Each Variable

Hochschule Koblenz Questionnaire Philipp Leubner

Questionnaire: Quality Analysis of OpenAI's ADA Tool

Imagine you are currently learning Data Science, Mathematics, or another similar field. In doing so, you come across the topic of "Multiple Linear Regression." Linear regression is an important foundation for understanding further regression models. The model has many assumptions that need to be checked. To understand these, you upload a dataset to ChatGPT and use an optimized prompt. Under this link, you will find the output from ChatGPT as a PDF, Jupyter Notebook, and HTML file (HTML file is recommended: download the HTML file and open it in your browser (do not open it from OneDrive, as it causes incorrect display of formulas)):

LINK TO THE OUTPUT EXPIRED

Please read through the output and answer the following questions. Your answers will be used to create an AI assistant optimized for this topic. This AI assistant should be able to help future students understand the topic in an applied manner, rather than just learning the theory.

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Part 1: Explanatory Data Analysis					
I understand how to read the provided histograms	☐	☐	☐	☐	☐
I understand how to read the provided boxplots	☐	☐	☐	☐	☐
I understand how to read the provided ECDF plots	☐	☐	☐	☐	☐
I understand how to read the provided QQ plots	☐	☐	☐	☐	☐
I have a general understanding about the distribution of the variables of the dataset	☐	☐	☐	☐	☐
Part 2: Preparations for Regression Diagnostics					
I understand the assumption of a linear regression model	☐	☐	☐	☐	☐
The mathematical equations are helpful for my understanding	☐	☐	☐	☐	☐

Figure B.10.: Questionnaire (1)

Hochschule Koblenz Questionnaire Philipp Leubner

Part 3: Regression Diagnostics					
Outliers					
I understand what outliers are.	☐	☐	☐	☐	☐
I understand the methods to check for outliers.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐
I need further information on the statistical test.	☐	☐	☐	☐	☐
High-Leverage Points					
I understand what high-leverage points are.	☐	☐	☐	☐	☐
I understand the methods to check for high-leverage points.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐

Figure B.11.: Questionnaire (2)

Hochschule Koblenz Questionnaire Philipp Leubner

I need further information on the statistical test.	☐	☐	☐	☐	☐
Non-Linearity					
I understand what non-linearity is.	☐	☐	☐	☐	☐
I understand the methods to check for non-linearity.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐
I need further information on the statistical test.	☐	☐	☐	☐	☐
Heteroscedasticity					
I understand what heteroscedasticity is.	☐	☐	☐	☐	☐
I understand the methods to check for heteroscedasticity.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐

Figure B.12.: Questionnaire (3)

Hochschule Koblenz | Questionnaire | Philipp Leubner

I need further information on the statistical test.	☐	☐	☐	☐	☐
Correlation of Error Terms					
I understand what correlation of error terms is.	☐	☐	☐	☐	☐
I understand the methods to check for correlation of error terms.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐
I need further information on the statistical test.	☐	☐	☐	☐	☐
Normality of Residuals					
I understand what normality of residuals is.	☐	☐	☐	☐	☐
I understand the methods to check for normality of residuals.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐

Figure B.13.: Questionnaire (5)

Hochschule Koblenz | Questionnaire | Philipp Leubner

I need further information on the statistical test.	☐	☐	☐	☐	☐
Collinearity of Predictors					
I understand what collinearity of predictors is.	☐	☐	☐	☐	☐
I understand the methods to check for collinearity of predictors.	☐	☐	☐	☐	☐
I understand the plot and the interpretation.	☐	☐	☐	☐	☐
I need further information on the plot.	☐	☐	☐	☐	☐
I understand the basic idea of the statistical test and the interpretation.	☐	☐	☐	☐	☐
I need further information on the statistical test.	☐	☐	☐	☐	☐

After you have answered the questions, please describe positive or negative aspects of the output. Please also describe what you are studying or in which field you are working.

____

Thank you all for helping me with my bachelor's thesis! ☺

Figure B.14.: Questionnaire (6)

Declaration of Authorship

I hereby declare that I have written the present work independently and only using the specified sources and tools, and that this work has not yet been used to obtain other academic credits. All verbatim and paraphrased excerpts and quotations are clearly marked and referenced. I assure that I have not used any tools whose use has been explicitly prohibited by the instructors.
By submitting the present work, I assume responsibility for the submitted overall product.
When using AI: I take responsibility for any AI-generated content that I have incorporated into my work. To the best of my knowledge and belief, I have verified the accuracy of the adopted (AI-generated) statements and content.
In the event of violations against the aforementioned declaration, the work will be graded as "failed" or "insufficient" and will be treated as an attempt to deceive in accordance with the examination regulations.

Please check the applicable option:
The work will be published internally within the university. With this I

[ ] I agree [ ] I do not agree

I agree that

- the title of my work, along with my name and the names of the supervisors, will be published in the library catalog (OPAC) and
- that the work can be viewed internally within the university as a PDF.

For this, the university receives a simple, non-transferable right of use exclusively for the purpose of publication in the library. The author's right to publish or exploit the work in other ways, e.g., through a publisher, remains unaffected. The university is not obliged to publish the work. Consent can be withdrawn at any time in writing (by letter). The library will then promptly remove the work from the OPAC or the display. Publication also depends on the later approval of the first supervisor.

Place, Date Signature