Temp Name

Data Science and Machine Learning - Spring 2025 - Ccs-25-dvml-2-02

2. Semester Project

media/aau_logo_en.pdf

Department of Computer Science
Selma Lagerløfs Vej 300
Aalborg Ø, 9220
http://www.es.aau.dk

Title:

TEMP NAME

Project:

2. Semester project

Period:

Februar 2025 - Juni 2025

Group:

cs-25-dvml-2-02

Members:

Christian Filtenborg Brogaard
Jonathan Skovbjerg Karoff
Marcus Lundgaard Terndrup
Puk Thejlmann Kalstrup
Sofia Jean Sabbah

Associated professors:

TEMP

Pages: ?????
Handed in ?????

Abstract:

□

Contents

1 Introduction

Regression is a tool in statistics used to understand the relationship between one dependent variable and one or more independent variables. For regression to be both accurate and reliable, a set of assumptions needs to be met. These assumptions are independence of errors, linearity, homoscedasticity, normality of errors, multicollinearity, and correct model specifications. If these assumptions are not met, it can introduce bias and inaccuracies in the regression model.

2 Problem Analysis

2.1 P

2.2 Assumptions

Regression models are tools for understanding relationships between variables and predicting . However, for this to work, certain assumptions need to be satisfied. This section will explain the previously mentioned key assumptions that are needed to ensure reliable results.

The assumption independence of errors states that the errors from the model are not correlated with each other but are independent. This means that one error cannot be used to predict the next one.

The assumption of linearity states that the relationship between the independent variables and the parameters, also known as the coefficients, is linear. This does not mean that the regression model itself has to be linear. For example, in polynomial regression, the relationship between the independent variables and the coefficients is still linear, but the independent variables can be transformed using powers.

Homoscedasticity is the assumption of constant variance across errors for all levels of the independent variables.

The assumption normality of errors states that the errors between the model and the observed values, also called residuals, are normally distributed. If this is not met, it may result in a biased model and a worse model fit.

Multicollinearity occurs when two or more independent variables in a regression model are highly correlated with each other. This means that changes in one independent variable are associated with changes in another, making it difficult to determine the individual effect of each independent variable on the dependent variable.

Correct model specifications assumes that the provided dependent variables for the models are the correct ones. If one is missing or the model is overfitting, this may result in incorrect coefficients and introduce errors into the model.

Data does not always adhere to these assumptions, so the results may be inaccurate. If such a case occurs, there are different ways to accommodate the situation. As mentioned previously, we will focus on the use of bootstrapping to achieve reliable results, while violating certain assumptions.

To understand how the classical method of constructing regression models and the method of using Monte Carlo Bootstrapping works, we need to dive deeper

into the background. Here, we need to understand ways to not only create a model, but how to test it as well. These topics will be explained in the following section.

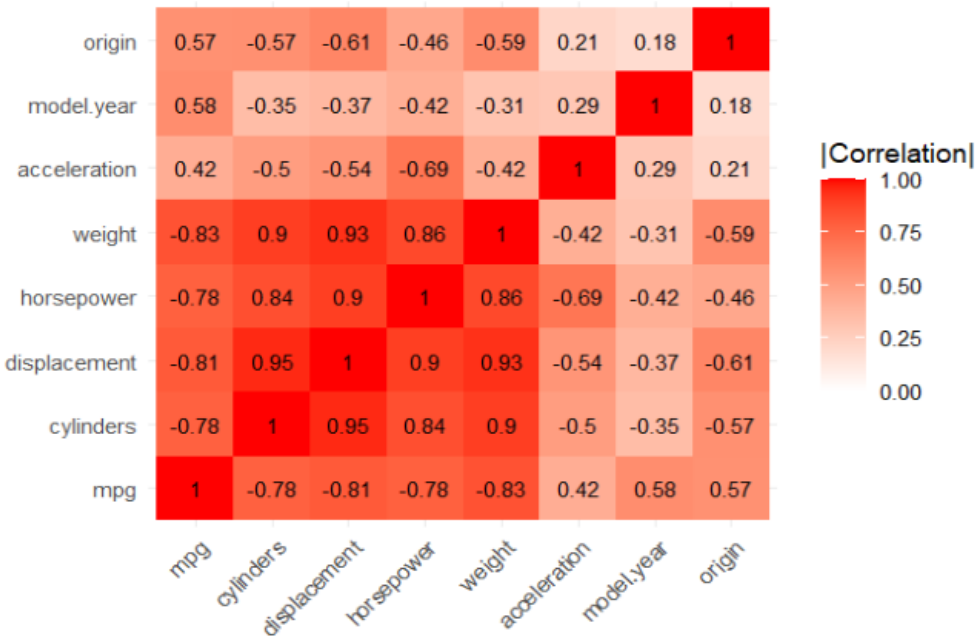

Figure 1: Heatmap made from the MPG dataset

3 MPG

An example of a data set containing violations in assumptions, specifically the assumptions of homscedasticity and no multicollinearity, is the "Miles Per Gallon" data set. Through the "Miles Per Gallon" it's possible to model the violations in the assumptions, as seen in Figure 1 and Figure 2.

Multicollinearity can be seen in Figure 1. Multicollinearit occurs when independent variables are highly correlated with each other, and as a result, it becomes more difficult to isolate the individual effect of each variable in a regression model. In this model it is shown through the numbers, where the high numbers suggest a strong linear relationship through the variables. So between 'displacement' and ' cylinders', the value is 0.95, which is close to 1 and therefore shows there is multicollinearity.

There is an example of hetereoscedasticity in the 'mpg' and 'horsepower' scatterplot in Figure 2, where there is not constant variance of the residuals. This is seen because there is a big spread between the variables, especially when horsepower decreases.

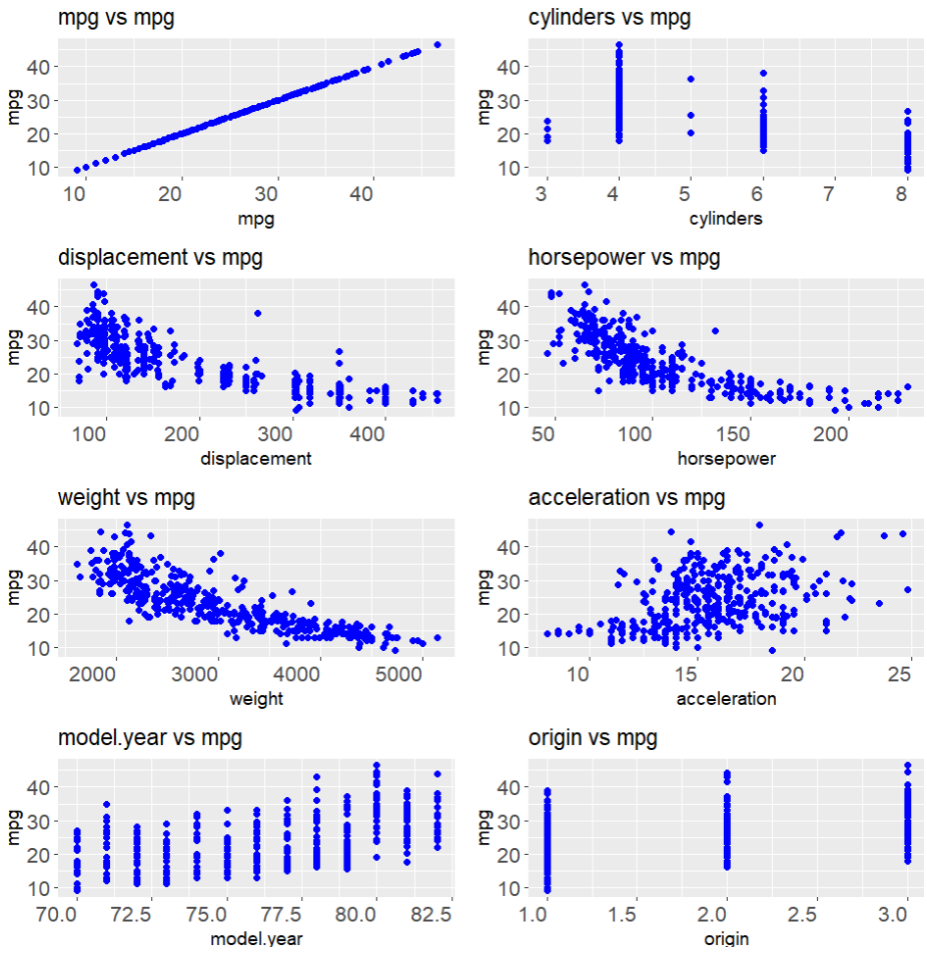

Figure 2: Scatterplots made from the MPG dataset

4 Background

Statistic is a field in mathematics, that gives tools to analyze and understand data. Statistics comes in two branches, these are descriptive statistics and inferential statistics. Descriptive statistics is used to describe the general tendencies in some data, such as finding the mean and variance, but no predictions or conclusion are made beyond the data itself. In contrast, inferential statistic is the branch of statistics that makes predictions and generalizations of a population based on a sample, this includes methods such as hypothesis testing and regressions.

To address the challenges, when violations in the assumptions of modeling regressions through classical means occur, a fundamental statistical understanding is needed. This section introduces the foundational concept of statistics that's

required to understand regression models and resampling methods.

4.1 Probability space

To understand how testing models work, we first need to start at the basics. First off is probability space.

The **sample space**, S, is the set of all possible outcomes.
If a sample space contains a finite number of possibilities or an unending sequence with as many elements as there are whole numbers, it is called a **discrete sample space**.
Example: When rolling a standard six-sided die form the discrete sample space, the possible outcomes are $S = 1, 2, 3, 4, 5, 6$.

If a sample space contains an infinite number of possibilities equal to the number of points on a line segment, it is called a **continuous sample space**.
Example: Measuring the heights of people in a population. This is a continuous sample space, because height can take any real value within a given range.
An **event** is a subset, $A \subseteq S$, of the sample space. The event is the amount that contains all possible events. An example of a discrete event could be rolling a die and getting an uneven number, this would be the event $A = 1, 3, 5$.
For the continuous event, it could be that a person is between 160 cm and 170 cm tall.

The probability of an event A, $P(A)$, is the sum of the weights of all sample points in A. The probability of the whole sample space is 1, $P(S) = 1$. The probability of any event being between 0 and 1,$0 < P(A) < 1$. The probability of the empty set being 0, $P() = 0$.

If A and B are mutually exclusive, $A \cap B =$, then
$P(A \cup B) = P(A) + P(B)$,

where A and B never occur at the same time, so their union is equal to the two events added together.

We have the probability of union,
$P(A \cup B) = P(A) + P(B) - P(A \cap B)$,
here, the union of the two events is A added to B, but minus their common event, since it otherwise would be added twice.
Two events A and B are independent, if
$P(A|B) = P(A)$.
The equivalent definition to this is:
Two events A and B are independent if and only if
$P(A \cap B) = P(A)P(B)$. This says that the probability of both event A and B happening, is equal to the product of the two events.

4.2 Random variables

A random variable is defined as a function that associates a real number with each element in the sample space. We use capital letters to denote a random variabel, for example X, and then the corresponding small letter, in this case x, for one of its values. As an example we roll a dice 3 times, which gives us a sample space of the different combinations. Each point in the sample space gets a numerical value assigned between 0 and 3.For example, if the random variabel X assumes the number of 5's rolled, then worst case is zero 5's rolled, and best case is three 5's rolled. These values are random quantities assumed by the random variabel X, and they are written like this: $X(5,1,2) = 1$ and $X(3,6,1) = 0$.

A random variable X can be discrete, which means that its set of possible outcomes is countable. The dice example is a discrete random variable, because you can count how many times 5 is rolled. The outcomes of some statistical experiments may be neither finite nor countable. For example when something is measured such as temperature or speed where the set of possible values is an entire interval of numbers, it is not discrete. The random variable X then takes values on a continues scale, which therefore is called a continuous random variable.

4.2.1 Discrete random variable

A discrete random variable can take each of its values with a certain probability. Frequently, it is convenient to represent all the probabilities of a random variable X by a formula. Let X be a discrete random variable which can take the values $x_1, x_2, ...$ Then the distribution of X is given by the probability function:

$$f(x_i) = P(X = x_i), \quad i = 1, 2, ...$$

For a discrete random variable this function is also called the probability mass function, where following holds for each possible outcome x:

- $P(X = x) = f(x)$.
- $f(x) \geq 0$,
- $\sum_x f(x) = 1$.

In addition to the probability mass function f, the discrete random variable X also has a cumulative distribution function $F(x)$ given by:

$$F(x) = P(X \leq x) = \sum_{x_i \leq x} f(x_i), \quad x \in \mathbf{R}.$$

This helps decide the probability that the random variable assumes a value equal to or smaller than x. Its sums up the probability density functions values.

The mean of a discrete variable X, with a distribution function $f(x_i)$ is given by:

$$\mu = E(X) = \sum_i x_i P(X = x_i) = \sum_i x_i f(x_i).$$

The mean is typically the expected value. It is a weighed average of the possible values of X. The values are weighed by its probability in the sample space.

In addition to the mean, we should also mention the variance. The variance is the mean squared distance between the values of the variable and the mean value. It is given by:

$$\sigma^2 = E\left[(X - \mu)^2\right] = \sum_i (x_i - \mu)^2 f(x_i).$$

The variance indicates whether the values of X are far from the mean values or close. A high variance means that the values of X have a high probability of being far from the mean values and vice versa. Along with the variance, the standard deviation is also often used. It is given by the square root of the variance:

$$\sigma = +\sqrt{\sigma^2}.$$

The advantage of the standard deviation over the variance is that it is measured in the same units as X.

4.2.2 Continuous random variable

Contrary to a discrete random variable, a continuous random variable can take values that are not countable. A continuous random variable can take infinetly many possible values within a certain range or interval. For a continuous random variable X the distribution is given by the probability density function f, which satisfies:

- $f(x)$ is defined for all x in $\mathbf{R}$,
- $f(x) \geq 0$ for all x in $\mathbf{R}$,
- $\int_{-\infty}^{\infty} f(x)\, dx = 1$.

Condition 3. ensures that $P(-\infty < X < \infty) = 1$, which means that the probability of the random variable X being between $-\infty$ and ∞ is 100%. Furthermore the probability of X assuming a specific value a is zero, in other words: $P(X = a) = 0$. That means that the values of the density function should not be interpreted as a probability of a given outcome. Instead the probability of X is found by integrating over the probability density function. So, the probability that a continuous random variable X lies between the values a and b is:

$$P(a < X < b) = \int_a^b f(x)\, dx.$$

A continuous random variable X also has a distribution function $F(x)$, that also predicts whether X assumes a value equal to or smaller than x. For a continuous random variable it is again given by integrating over the probability density function in the interval from $-\infty$ to x:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(y)\ dy.$$

That also means $P(a < X < b)$ can be calculated by $F(b) - F(a)$.

For a continuous random variabel X the mean, variance and standard deviation the same interpretation applies. Just given by different formulars, which are:

$$\mu = E(X) = \int_{-\infty}^{\infty} xf(x)\,dx$$

and

$$\sigma^2 = E\left[(X-\mu)^2\right] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx,$$

(The standard deviation is still given by til square root of the variance).

4.3 Estimator and estimates

If we are interested in certain parameters of a population distribution, we can look at a sample. From this, we can make a **point estimate**.
Examples of this are,
$\bar{x}$ is a point estimate of μ
s is a point estimate of σ

This is often supplemented with a **confidence interval**
This is an interval around the point estimate, where we are confident that the population parameter is located.

For μ, we have different ways of estimating it. We can use the sample mean $\bar{X}$, or the average X_T of the sample upper and lower quartiles. But in this case, we have to look out for **bias**. If the distribution of a population is skewed, then X_T is biased. The result of this is, that in the long run, this estimator will systematically over or under estimate the value of μ. This is written as,
$E(X_T) \neq \mu$.
It is generally preferred that the estimator is **unbiased**. In this case, $\bar{X}$ is an unbiased estimate of the population mean μ.

The standard error of $\bar{X}$ is $\frac{\sigma}{\sqrt{n}}$. Here, the standard error decreases, when the sample size increases. If an estimator has this property, it is called **consistent**. If we compare, the estimator X_T is also consistent, but has a greater variance than $\bar{X}$.
It is generally preferred that the estimator has the smallest possible variance,

and in that case it is called efficient. So $\bar{X}$ is an efficient estimator.
When estimating a parameter, the symbol ˆ is used above it. For μ, $\hat{\mu} = \bar{X}$.
We can calculate $\bar{X}$ using the following formula,

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

For the variance σ, we can estimate it by using the formula for S^2,

$$S^2 = \frac{1}{n-1}\sum_{i=1}(X_i - \bar{X})$$

4.4 Probability distribution

Data can come in various distributions depending on different parameters such as degrees of freedom. The distribution is the shape of the data and it will have an effect on statistical models. Therefore it is important to have an understanding of distributions.

4.4.1 Normal distribution

In the world of statistics, the most common distribution is the normal distribution. It is constructed as a bell shape. The normal distribution is a continuous distribution, with this density function:

$$n(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)}{2\sigma^2}}$$

The distribution is dependent on the mean(μ) and the standard deviation(σ), where changes to the mean will result in a change in the positioning of the normal distribution. Whereas a change in the standard deviation will change the spread of the curve. The normal distribution also always contains an area under the curve that is equal to one. This is to ensure that the normal distribution correctly models probability.

There is a special variation of the normal distribution, called the standard normal distribution, where the mean is zero and the standard deviation is one. The standard normal distribution can be seen in Figure 3, this distribution is widely used in statistics as it makes math behind computing statistical inference easier. All variations of a normal distribution can be standardized by a transformation of the distribution, using the Z-score formula.

$$Z = \frac{X - \mu}{\sigma}$$

Z in the Z-score represents the amount of standard deviations a given X value, deviates from the mean.

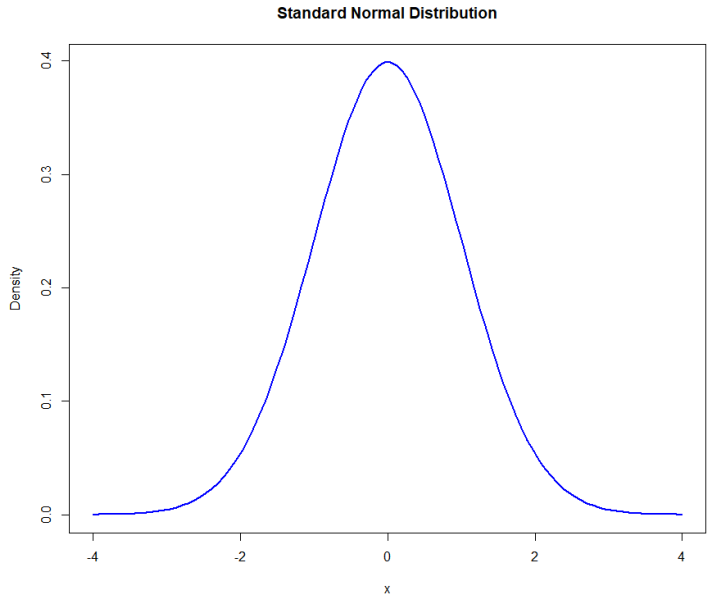

Figure 3: The standard normal distribution

4.4.2 The central limit theorem

A very effective theorem in statistics is the central limit theorem. This theorem states that if a random sample $\overline{X}$, with the size n, is taken from a population with a mean and a finite variance, then as n goes towards infinity, the distribution will resemble a normal distribution. If used with the Z-score formula, the distribution will resemble a standard normal distribution. The formula for the Z-score, when in conjunction with the central limit theorem, looks like this:

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}}$$

Where $\overline{X}$ is a random sample of size n and μ is the mean of the true population. The standard error i represented by $\sigma/\sqrt{n}$, where σ is the standard deviation and n is the sample size. Usually the standard deviation is unknown, for these situations it's possible to use the estimator S^2. This estimates the variance of the population from the variance of the sample, by this formula:

$$s^2 = \sum_{i=1}^{n} \frac{(x_i - \overline{x})}{n-1}$$

The square root of the variance is the standard deviation, therefore the square root of the estimator S^2 would be the estimated standard deviation. The

problem with using the estimator S^2, is that with small samples the variance is small and therefore it contains a lot of bias. In this situation the t-distribution would be used instead of the normal distribution, because the t-distribution takes the bias into account the bias of the standard deviation. It does this by having thicker tails, meaning that the probability of more extreme values are higher.

4.4.3 The t-distribution

The t-distribution is shaped as the standard normal distribution, in a bell shape and symmetrical around the mean of zero, the difference is that the t-distribution contains more variance. This comes from the fact that the t-distribution is dependent on the degrees of freedom. When the degrees of freedom surpasses 30, the rule of thumb is that the distribution will resemble a normal distribution. So before 30 degrees of freedom, the distribution contains more variance. The t-distribution will come to resemble the standard normal distribution, when in surpasses 30 degrees of freedom, this makes sense, since the two distributions have the same formula:

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}}$$

The only difference is the estimated standard deviation S. As seen in Figure 4, the t-distribution approaches the normal distribution as the degrees of freedom increases. The t-distribution with a smaller amount of degrees of freedom, will have more mass further out from the center, but as the degrees of freedom increases and approaches 30, then the mass will shift towards the center and shaping itself as the standard normal distribution.

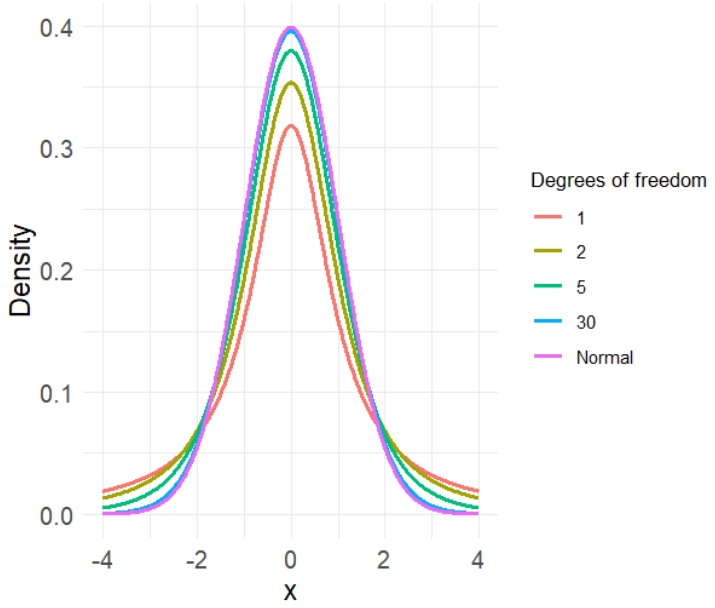

Figure 4: The t-distribution as it approaches the standard normal distribution

4.5 Statistical methods

This section will present statistical methods to evaluate the reliability and significance of a regression model. The focus will be on confidence intervals and hypothesis testing, these are two methods that can access the model uncertainty and determine whether observed effects are random.

4.5.1 Confidence intervals

The confidence interval is a good tool to use, when trying to estimate a parameter of a population. Its used to create an interval, where the parameter has a probability to lie inside of. This probability is called the confidence level and it's a chosen value, usually the chosen confidence level is either 95% or 99%. The confidence interval will become bigger with a smaller confidence level. A good confidence interval is small with a large confidence level, this will usually occur when the sample size is large. The chosen confidence level relates to an α-value, where as an example the chosen confidence level is 95%, then the α-value would be 5% or normally written as 0.05. The α-value will sometimes be needed to find the critical value, that is used to calculate the margin of error, as an example it's used when trying to find the critical value of the confidence interval, when working with a t-distribution.
To set up a confidence interval, the margin of error needs to be computed and

then that will be both added and subtracted from the point estimate. This will give the values of the outer bounds of the interval. The margin of error is calculated from this formula:

$$\text{Margin of Error} = \text{Critical Value} \pm \text{Standard Error}$$

The standard error will change depending on which parameter that the confidence interval is estimating, but the general formula for the standard error is:

$$\frac{\sigma}{\sqrt{n}}$$

An example of computing a confidence interval of the mean while working with a standard normal distribution, then the formula for the confidence interval would be this:

$$P(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1 - \alpha$$

Where $1 - \alpha$ is the confidence level. As it's the mean that is being estimated, then instead of Z-score, then μ must be isolated and that is done by multiplying $\frac{\sigma}{\sqrt{n}}$ and subtracting $\bar{X}$ on all sides, then multiplying all side by -1 to remove the minus sign. So the formula for a confidence interval of the mean will look like this:

$$P(\bar{X} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}) = 1 - \alpha$$

This formula will give the upper and lower bounds of the confidence interval.

The interpretation of a confidence interval
To interpret a confidence interval, it would be incorrect to interpret the confidence level of some value x, as the probability of the true parameter being inside of the interval. The reason behind this is that the computed interval is static, so either the value x is inside the interval or it's not. So the correct way of interpreting the confidence interval is by taking multiple samples and computing the confidence interval for all samples, then the value x would reside inside 95% of the confidence intervals. **Kilde for fortolkningen af kofidense intervaller:**
http : *//www.drhuang.com/science/mathematics/book/probability*$_a$*nd*$_s$*tatistics*$_f$*or*$_e$*ngineering*$_a$*nd*$_t$*he*$_s$*cienc*

4.5.2 Hypothesis testing

A hypothesis test is used to test an assumption about a population. This is done from a sample of the population, as the information about the population is

usually hard to come by. A hypothesis test is set up, by having a null hypothesis and an alternate hypothesis.

$$H_0 = \text{Null hypothesis}$$

$$H_a = \text{Alternate hypothesis}$$

When working with hypothesis testing, the hypothesis H_0 is usually represented as the status quo, where as the hypothesis H_a is represented as the opposition. It is also important to note that there is only two outcomes of a hypothesis test, either H_0 is rejected in favor of H_a or H_0 is failed to be rejected. Therefore in no situation can H_0 be stated to be an absolute truth, as there might be other samples where H_0 will be rejected. Therefore in a hypothesis test H_0 needs to be the thing that can be rejected and if H_0 gets rejected, then H_a will become the new status quo until proven otherwise.
In a hypothesis test H_0 will be the assumption that a parameter for two populations is the same, where as H_a can be either one of three assumptions, depending on the intention of the hypothesis test.

$$H_0 : \theta = \theta_0$$

1. $H_a : \theta \neq \theta_0$
2. $H_a : \theta < \theta_0$
3. $H_a : \theta > \theta_0$

When the direction of the rejection is not important and also is unknown, then (1) will be the case. This scenario sets up a two-tailed-test, where the hypothesis test is used to reject H_0 if H_a is either significantly larger or smaller than H_0, this means that the critical area is on both sides of the difference of θ and θ_0. Either (2) or (3) will set up a one-tailed-test, where depending on what is important, either the hypothesis test is used to determine if H_a is significantly bigger or smaller than H_0. This means that the critical area only spans one side of the difference between θ and θ_0.

Error in hypothesis testing
When making a hypothesis test there is four different possible outcomes. The results are separated by correct decisions and errors. There exist two types of hypothesis errors, called type 1 error and type 2 error. The type 1 error occurs when H_0 is mistakenly rejected and H_0 is true. Type 2 error is the opposite, where H_a is rejected and H_a is true. The types of outcomes occurring from a hypothesis test can be seen in Table 1

It is possible to compute the probability of a type 1 error occurring, this value is the same as the significance level α. To calculate the probability of a type 2 error occurring, the H_a needs to be defined, more specifically the mean of the sample is needed. Depending on the which parameter is known, different formulas are taking into use. As an example where the standard deviation is

	H_0 is true	H_0 is false
Does not reject H_0	Correct decision	Type 2 error
Reject H_0	Type 1 error	Correct decision

Table 1: Outcomes of a hypothesis test

known, its a normal distribution and its a one tailed test, then the formula for the Z-score is used, but $\overline{X}$ is changed with $\overline{x}_{crit}$ and μ is changed with μ_1:

$$Z = \frac{\overline{x}_{crit} - \mu_1}{\sigma/\sqrt{n}}$$

The value of $\overline{x}_{crit}$ is the value that separates whether H_0 is rejected or not and μ_1 is the value of the alternative hypothesis. The value of the calculated Z-score is used in a table of areas under the normal curve. This value will be used as the probability of a type 2 error occurring.

Understanding the relationship between variables is crucial when working with data, as it forms the basis for drawing meaningful conclusions. Regression models are a fundamental statistical tool used to model and analyze these relationship. Regressions models are not always robust and can include bias in their conclusions, this is a result of modeling a regression that includes breaks on one or more assumptions. This section will specifically focus on the assumption of constant variance of errors also called homoscedasticity and the assumption of no multicollinearity.

Polynomial Regression

This section is based on Linear regression is a model that estimates the relationship between a dependent variable, y, and one or more independent variables, x.
A reasonable relationship between the two in simple regression is the linear relationship:

$$Y = \beta_0 + \beta_1 x.$$

Where β_0 is the intercept, and β_1 is the slope.
In a lot of cases, there will be more independent variables, so the relationship for multiple regression will look like this:

$$Y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n.$$

Where n is the number of independent variables, and β_2 further shapes the curvature and complexity of the curve. Linear models use the method of least squares of the residuals to estimate parameters, in order to find the best fitting line for the data.
In simple linear regression, the random error ϵ is included:

$$Y = \beta_0 + \beta_1 x + \epsilon.$$

It is assumed that ϵ is distributed with $\epsilon = 0$ and $\epsilon) = \sigma^2$, and it has consistent variance, which is usually called the *homogeneous variance assumption*. The random error ϵ adds randomness to account for the natural variability in real data, making the model more realistic.

Polynomial regression is a form of linear regression, but the relationship between x and y is an nth-degree polynomial. It fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, meaning the model predicts the expected value of y given x.

That is why it is used when the relationship between the independent variable and the dependent variable is better represented by a curve rather than a straight line, since it can show the nonlinear patterns in the data. In polynomial regression, as mentioned before, there are six assumptions, that always need to be met.

4.6 The method of least squares

To find connections in data, it is necessary to estimate coefficients, β_0 and β_1, in linear models. A widely used method to estimate the coefficients, is the least squares method that was previously mentioned.
Least squares considers the deviation of Y_i for its expected value, where the observations are (X_i, Y_i). This method also requires, that we consider the sum of the n squared deviations. This is denoted by the criterion Q,

$$Q = \sum_{i=1}^{n} (Y_1 - \beta_0 - \beta_1 X_i)^2.$$

According to the method of least squares, the estimations of β_0 and β_1 that minimize Q for the given sample observations $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$, are called b_0 and b_1.
If an analytical approach is used, the values b_0 and b_1 that minimize Q for any particular set of sample data are given by these simultaneous equations:

$$\sum Y_i = nb_0 + b_1 \sum X_i.$$

$$\sum X_i Y_i = b_0 \sum X_i + b_1 \sum X_1^2.$$

These equations are called *normalequations*, where b_0 and b_1 are the *pointestimators* of β_0 and β_1. It is possible to calculate these normal equations simultaneously for b_0 and b_1 through these expressions,

$$b_1 = \frac{\sum (X_1 - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}.$$

$$b_0 = \frac{1}{n}(\sum Y_i - b_1 \sum X_i) = \bar{Y} - b_1 \bar{X}.$$

Here, $\bar{X}$ and $\bar{Y}$ are the means of X_i and Y_i.
The normal equations can also be derived by differentiating with respect to β_0 and β_1:

$$\frac{\partial Q}{\partial \beta_0} = -2 \sum (Y_i - \beta_0 - \beta_1 X_i).$$

$$\frac{\partial Q}{\partial \beta_1} = -2 \sum X_i (Y_i - \beta_0 - \beta_1 X_i).$$

By setting these derivatives equal to zero, we can find b_0 and b_1,

$$-2 \sum (Y_i - b_0 - b_1 X_1) = 0.$$

$$-2 \sum X_i (Y_i - b_0 - b_1 X_1) = 0.$$

This can be simplified,

$$\sum_{i=1}^{n} (Y_i - b_0 - b_1 Xi) = 0.$$

$$\sum_{i=1}^{n} X_1(Y_i - b_0 - b_1 Xi) = 0.$$

And it can be expanded, so the normal equations are obtained,

$$\sum Y_1 - nb_0 - b_1 \sum X_1 = 0.$$

$$\sum X_1 Y_1 - b_0 \sum X_1 - b_1 \sum X_i^2 = 0.$$

Solving this for b_0 and b_1 will lead to values, that minimize Q, and these are the estimates for β_0 and β_1. When rearranging terms, we get the normal equations The estimates b_0 and b_1 obtain the minimum when checking the second partial derivatives.

4.7 Linear models

Equations for linear models can be written in matrix terms, where the normal error regression model for simple linear regression is.

$$Y_i = \beta_0 + \beta_1 X_i 0 \epsilon_i.$$

Which implies that,

$$Y_1 = \beta_0 + \beta_1 X_1 + \epsilon_1$$
$$Y_2 = \beta_0 + \beta_2 X_2 + \epsilon_2$$
$$\vdots$$
$$Y_n = \beta_0 + \beta_n X_n + \epsilon_n.$$

The observations vector Y is,

$$Y_{n\times 1} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}.$$

The X matrix is,

$$X_{n\times 2} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}$$

The β vector is,

$$\beta_{2\times 1} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

And the ϵ vector is,

$$\epsilon_{n\times 1} = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

This can be written in matrix terms with a dot product,

$$Y_{n\times 1} = X_{n\times 2} \cdot \beta_{2\times 1} + \epsilon_{n\times 1},$$

where,
Y is a vector of response
β is a vector of parameters
X is a matrix of constants, called det design matrix
ϵ is a vector of independent normal random variables with expectation

This can be shown in columns,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

If the dependent variable Y has more than one independent variable in a linear model, the equation looks like this,

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p - 1 X_{i,p-1} + \epsilon_i.$$

In matrix terms it is,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & ... & X_{1,p-1} \\ 1 & X_{21} & ... & X_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n1} & ... & X_{n,p-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p - 1 \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

The Y and ϵ vectors are the same as in the simple linear regression matrix. The β vector has additional parameters, and the X matrix now has a column of n observations for each $p - 1X$ variables.

4.8 Polynomial regression

Polynomial regression models the relationship like this,

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_{p-1} x^{p-1} + \epsilon.$$

The coefficients can still be found through the method of least squares,

$$Q = \sum_{i=1}^{n} (Y_i - (\beta_0 + \beta_1 X_i + \beta_2 x_i^2 + ... + \beta_{p-1} X_i^{p-1}))^2.$$

This can be written in matrix terms as,

$$Q = (Y - X\beta)'(Y - X\beta).$$

Where the design matrix for polynomial regression is,

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & ... & x_1^{p-1} \\ 1 & x_2 & x_2^2 & ... & x_2^{p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & ... & x_n^{p-1} \end{bmatrix}.$$

We can expand the expression from before, so it looks like this,

$$Q = Y'Y - Y'X\beta - \beta'X'Y + \beta'X'X\beta.$$

It is possible to find the value of β that minimizes Q by differentiating with respect to β_0 and $beta_1$,

$$\frac{\partial}{\partial \beta}(Q) = \begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \end{bmatrix}.$$

$$\frac{\partial}{\partial \beta}(Q) = -2X'Y + 2X'X\beta.$$

Then minimum is found by calculating, where the gradient is 0.

$$-2X'Y + 2X'X\beta = 0,$$

$$2X'X\beta = 2X'Y,$$

$$(X'X)^{-1}X'X\beta = (X'X)^{-1}X'Y,$$

$$\beta = (X'X)^{-1}X'Y.$$

For this to be calculated, the matrix $X'X$ has to be invertible, so the columns of X have to be linearly independant. This means, that the assumption of no multicollinearity has to be met.
So,

$$X'Xa = 0$$

$$a'X'Xa = 0$$

$$(Xa)'(Xa) = 0$$

This shows, that $X'Xa = 0$ if and only if $(Xa)'(Xa) = 0$, so $Xa = 0$.

If it is supposed, that there exists a non-trivial solution for $Xa = 0$, there also exists a non-trivial solution for $X'Xa = 0$. It is therefore only invertible when the null space of X is 0, or if the columns are linearly independent. This is called no multicollinearity, since there is independence between the independent variables.
To ensure that the solution gives a minimum, the **Hessian** is calculated,

$$\frac{\partial^2 Q}{\partial \beta' \beta} = 2X'X$$

If it is positive definite, the function Q has a global minimum, which can be shown with,

$$a'(X'X)a = (Xa)'(Xa)$$

As shown in these can only be calculated coefficients, if $Xa \neq 0$. This means, that $X'X$ is positive definite in all relevant situations, which is when the coefficients can be calculated.
The solution to the least squares method results in only containing $X'X$ and $X'Y$. The matrices are,

$$X = \begin{bmatrix} 1 & X_1 & X_1^2 \\ 1 & X_2 & X_2^2 \\ \vdots & \vdots & \vdots \\ 1 & X_n & X_n^2 \end{bmatrix}$$

$$X'X = \begin{bmatrix} n & \sum X_i & \sum X_i^2 & ... & \sum X_i^{p-1} \\ \sum X_i & \sum X_i^2 & \sum X_i^3 & ... & \sum X_i^p \\ \sum X_i^2 & \sum X_i^3 & \sum X_i^4 & ... & \sum X_i^{p+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum X_i^{p-1} & \sum X_i^p & \sum X_i^{p+1} & ... & \sum X_i^{2p-2} \end{bmatrix}$$

$$X'Y = \begin{bmatrix} \sum Y_i \\ \sum X_i Y_i \\ \sum X_i^2 Y_i \\ \vdots \\ \sum X_i^{p-1} Y_i \end{bmatrix}$$

These are the necessary matrices to use the least squares method.

4.9 Assumptions

When making a regression its important to understand, that the regression has assumptions that needs to be fulfilled if the statistical conclusions are to be correct. If these assumptions are not upheld, it will create bias that will skew the results of the model.

4.9.1 Homoscedasticity

One of the assumptions of a polynomial regression is that homoscedasticity is fulfilled. Homoscedasticity is the assumption of constant error variance, where observations in a dataset would exhibit errors that have roughly the same spread across all levels of the independent variable.
If this assumption is not upheld, then this will cause the standard error to be biased and therefore not trustworthy. This problem causes further testing involving this standard error to become wrong, an example is the hypothesis test. The reason for the assumption needs to be upheld, comes from how the regression is created. The regression is created via the ordinary least square method, that requires the assumption of homoscedasticity to be upheld.
A way to display homoscedasticity is through the variance-covariance matrix. The matrix shows whether the data contains homoscedasticity or heteroscedasticity through the diagonal values. If the matrix contains all the same values through the diagonal, then the assumption of homoscedasticity is upheld, else the data contains heteroscedasticity. This is a showcase of the variance-covariance matrix with homoscedasticity:

$$Var(\varepsilon) = \sigma^2 = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix}$$

The general way of writing the variance-covariance matrix, is by this formula:

$$Var(\varepsilon) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} & \cdots & \sigma_{2n} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} & \cdots & \sigma_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \sigma_{3n} & \cdots & \sigma_{nn} \end{bmatrix}$$

Every position in the matrix is calculated, then the diagonal will tell if the data contains homoscedasticity or heteroscedasticity. The positions that are not on the diagonal should be zero else the data contains another problem, that is autocorrelation, meaning that the observations in the data set are correlated.

Source: https://openpublishing.library.umass.edu/pare/article/id/1590/

Detecting heteroscedasticity

When working with data, one method of checking for heteroscedasticity is to visualize it through plotting the residuals, but its not always possible to detect heteroscedasticity through visual media. Another approach is to calculate if the data contains heteroscedasticity. This can be done through the Breusch-Pagan test, that is specifically designed to detect heteroscedasticity. The test detects heteroscedasticity, by regressing the residuals on the independent variables and checks if the independent variables has an effect on the residual variance. If this is not the case, then there is no heteroscedasticity. This checked

through a hypothesis test, where the H_0 is that the data contains homoscedasticity and H_a is that the dataset contains heteroscedasticity. source: *https* : *//sscc.wisc.edu/sscc/pubs/RegDiag−R/homoscedasticity.html?fbclid* = *IwZXh0bgNhZW0CMTEAAR4q*
IAo8l − *UaO*1*oDWCuUfkvw*$_a$*em*$_9$*Ft*7*wkFWKUHycQ*1*GnX* − *x*0*g*

4.9.2 No multicollinearity

Perfect multicollinearity is a term used for describing a perfect linear relationship between two or more independent variables. This relationship occurs when an independent variable can be perfectly predicted from other independent variables. In mathematical terms, this could be written as a linear regression:

$$X_1 = c + \beta_1 \cdot X_2 + ... + \beta_n \cdot X_n$$

Where $X_1...X_n$ is all the independent variables that have a perfect linear relationship. The coefficients are represented by $\beta_1...\beta_n$ and they are the amount that X_1 changes when their relative independent variable changes. Lastly c is the intercept and represents the value of X_1, when all other independent variables are zero.
The regression model can feel the effects of multicollinearity even without there being perfect multicollinearity. A strong linear relationship is enough to have an effect on the model. The problem caused by multicollinearity, is that as it increases the variance of the value that the coefficients can receive also increases. Where as perfect multicollinearity will make the model unable to estimate a value of one coefficient, due to the perfect linearity between the independent variables.
Source: https://ekja.org/upload/pdf/kja-19087.pdf
Detecting multicollinearity
To check for multicollinearity in a dataset, a good approach is pearson´s correlation coefficient. This will make a table of all pairwise correlation, this means that all combinations of independent variables are checked for multicollinearity. This can be seen in **table...**. The correlation coefficient is calculated through this formula:

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Where n is the number of observations, with x and y representing the two variables tested for correlation and the pairwise correlation coefficient is denoted as r. When computing the value of r, the value will be in a range: $-1 \leq r \leq 1$. If the value of r is -1 or 1, that indicates a perfect either negative or positive correlation and if the value is 0, then there is no correlation between the variables.

When producing a synthetic dataset, there is a need for a lot of random numbers. Many programming languages have a built-in function that produces numbers that appear random, but actually are not. Computers are deterministic machines, and therefore cannot produce a number without some sort of algorithm. If the algorithm is known, it becomes possible to predict the next number; hence the numbers are not completely random. Random numbers should be independent; that is, each number should have no connection to any previously produced values. The distribution should also be uniform, meaning if you generate 1,000,000 random numbers in the range [0,1), you'd expect about 500,000 values in [0, 0.5) and about 500,000 in [0.5, 1). Earlier in history, these numbers have been produced by flipping coins or rolling dice. Today, it is possible to produce truly random numbers by using atmospheric noise. Despite its potential advantages, this method requires significant resources, making it inefficient for the intended application. Therefore, pseudo-random numbers will be used instead.

Pseudo-random numbers behave like random numbers but are deterministically generated from a seed value. While these numbers are not truly random, they are sufficiently unpredictable for many practical applications. To generate the data, we use a Pseudo-Random Number Generator, also known as a PRNG. PRNGs are algorithms that produce sequences of numbers that appear random.

Random numbers are widely used in fields such as statistics, game theory, cryptography, and simulations. These applications require numbers that behave as if they were random, yet can be reproduced when needed. This is where PRNGs come in. They allow for repeatable randomness, making them ideal for controlled experiments, testing, and security.
This chapter will explore the key concepts behind PRNGs. Before going into the mechanics of these generators, it is important to first understand what 'random' means and the characteristics that define truly random numbers.

4.10 Properties of PRNGs

The quality of a PRNG is determined by several key factors that influence its use for different applications. Some of the properties of a good PRNG are independence, a long period and reproducibility

The numbers produced by the PRNG should be statistically independent, ensuring that each generated value exhibits no correlation with previous numbers or other sequences. This implies that knowledge of previously generated numbers or sequences provides no advantage in predicting the next output.

A PRNG operates within a specific interval before its sequence begins to repeat. A high-quality PRNG has a long interval, delaying repetition and enhancing its unpredictability. Conversely, a PRNG with a shorter period becomes more predictable and less suitable for practical use.

A key feature of a PRNG is its ability to reproduce the same sequence of numbers when given a specific seed. This property is particularly useful in testing and simulation scenarios, where it is essential to generate identical sequences multiple times for consistency and reproducibility.

In addition, a PRNG must be fast and efficient to prevent it from introducing performance bottlenecks within an application. The speed of number generation directly impacts computational efficiency, especially in applications requiring a large volume of random numbers. An inefficient PRNG can significantly slow down processes, undermining the overall performance of the system. Therefore, balancing randomness and efficiency is essential for practical applications

4.11 Linear Congruential Generator

Linear Congruential Generator (LCG) is a commonly used approach to generate pseudo-random numbers. LCG generates a sequence of numbers using a linear recurrence relation, expressed as:

$$X_{n+1} = (aX_n + c) \bmod m[2].$$

X_0 is the seed value and must be in the range $0 \leq X_0 < m$
a is the multiplier,
c is the increment and
m is the modulus, which specifies the range of values. m must be greater than 0

The operation 'mod m' represents division by m, where only the remainder is retained. This ensures that the generated number remains within the range 0 to $m-1$. X_0, a and c must all be in the interval $[0, m)$. Here is an example of the first 4 numbers of a sequence given these parameters:

$a = 5$, $c = 1$, $m = 16$, and $X_0 = 7$:

$$X_1 = (5 \cdot 7 + 1) \bmod 16 = 4.$$
$$X_2 = (5 \cdot 4 + 1) \bmod 16 = 5.$$
$$X_3 = (5 \cdot 5 + 1) \bmod 16 = 10.$$
$$X_4 = (5 \cdot 10 + 1) \bmod 16 = 3.$$

This sequence has a period of 16. In an LCG, the period can be as large as m, because the remainder after division by m will always be less than or equal to

m. Consequently, choosing a large m is typically desirable, as it can potentially lead to longer periods. However, the period length is not determined solely by m; the choice of other parameters—such as the multiplier, increment, and the seed, along with their relationships, significantly impact the overall period. It is possible to select a larger m and still end up with a shorter period if the parameters are not chosen properly. Here is an example where a larger m results in a shorter period, illustrating that.

$$a = 4,\ c = 6,\ m = 20,\ X_0 = 3.$$

$$X_1 = (4 \cdot 3 + 6) = 18 \bmod 20 = 18.$$
$$X_2 = (4 \cdot 18 + 6) = 78 \bmod 20 = 78 \bmod 20 = 18.$$

Here a larger m is used, but a shorter period of 1 appears.

The selection of the optimal parameters for a LCG would be excessive for the objective of this project, it is beyond the scope of this project and will therefore not be addressed further.

4.12 Test of PRNG

As stated previously, the length of the sequence produced by the PRNG is not the only important factor. A uniform distribution and independence of each generated number is significant too. If these criteria are not met, there will be correlation between the generated numbers, which means that the randomness is of low quality. For example, consider a sequence that follows the three-times-table, with the first number being 3: (3,6,9,12,...). In that sequence, it is pretty easy to guess the upcoming number, based on the previous one. Normally the correlations are harder to spot than this example, which is why it is important to test the quality of a PRNG.

There are different tests used to evaluate PRNG quality and performance, including the Kolmogorov-Smirnov test, chi-squared test and the spectral test [2]. Passing one of these tests does not mean that it will pass others. Therefore, every test that it passes, just makes it more likely to produce a high quality sequence. One of the important tests for LCG is the spectral test. Another advantage of the spectral test in the context of this paper is that it can be visualized in two or three dimensions, which can make it easier to understand.

The test looks at how the numbers in the sequence are distributed in different dimensions. If hyperplanes and lines occur, as seen on Figure 5 and Figure 6, generated by $X_{n+1} = 137 \cdot X_n + 187 \bmod 256$, the sequence fails the test, since the distribution is not random.

Looking at the spectral test used on a LCG with the parameters from the Park-Miller Minimal Standard $X_{n+1} = (48271 \cdot X_n) \bmod (2^{31} - 1)$[2], on Figure 7

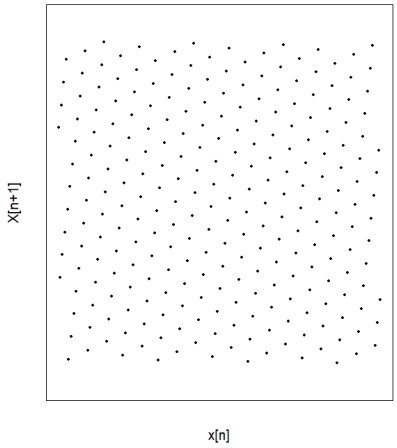

Figure 5: 2D spectral test for LCG using bad parameters: $X_{n+1} = (137 \cdot X_n + 187) \bmod 256$.

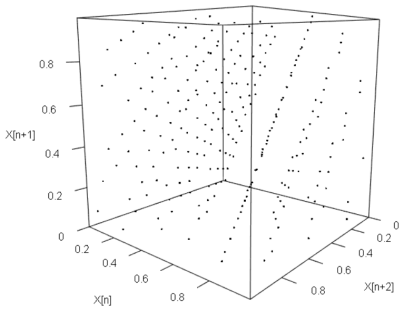

Figure 6: 3D spectral test for LCG using bad parameters: $X_{n+1} = (137 \cdot X_n + 187) \bmod 256$.

and Figure 8, the hyperplanes will not always be visible in 2d or 3d, but only in higher dimensions. Therefore a visual examination will not be enough to conclude the complete quality of a PRNG. Methods for inspecting LCGs in these dimensions do exist, but will not be mentioned in this project

In this project, the PRNG used will be the Mersenne-Twister. It has very good properties, uniformity and a period of $2^19937 - 1$ and passes the spectral test [3]. The test in 2D and 3D can be seen on figure 9 and figure 10. Another reason for this choice is that Mersenne-Twister is the built-in generator in R, which makes it convenient.

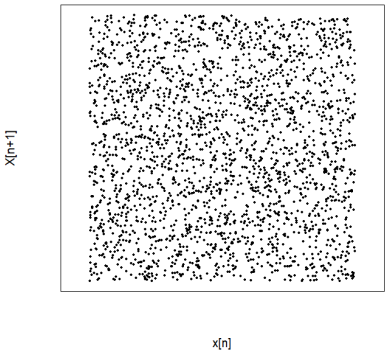

Figure 7: 2D spectral test for LCG using good parameters: $X_{n+1} = (48271 \cdot X_n) \bmod (2^{31} - 1)$.

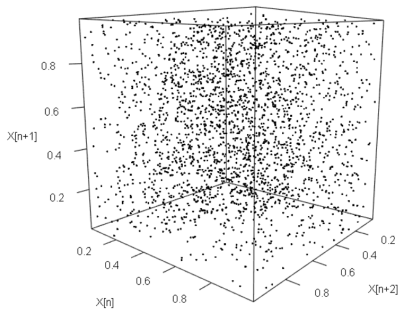

Figure 8: 3D spectral test for LCG using good parameters: $X_{n+1} = (48271 \cdot X_n) \bmod (2^{31} - 1)$.

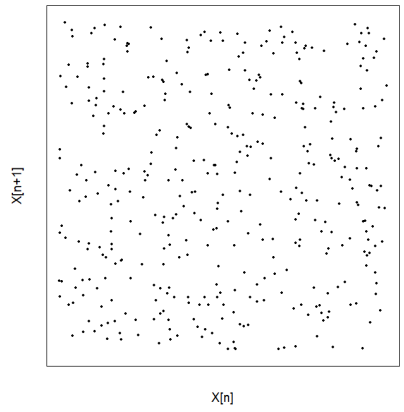

Figure 9: 2D Spectral test for the PRNG Mersenne-Twister, in R.

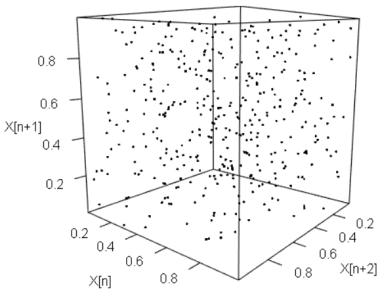

Figure 10: 3D Spectral test for the PRNG Mersenne-Twister, in R.

The following mathematical theory is based on the book, "Mathematical Statistics with Resampling and R" (Chihara & Hesterberg). The bootstrap is a procedure that uses a given sample to create a new distribution, called the bootstrap distribution. To find the bootstrap distribution, other samples are drawn from the original sample; in that way, it is a tool for resampling. The bootstrap distribution is used to approximate the sample distribution of a specific statistic θ. For most statistics, bootstrap distributions approximate the spread, bias, and shape of the actual sampling distribution. The statistic could for example be the mean. To create the bootstrap distribution of the mean, bootstrap samples are drawn from the original sample, and then the mean is calculated for each resample. In other words, the original sample is now treated as the population. If the statistic tested is the mean, then the bootstrap distribution of the mean will look approximately like the sampling distribution of the mean and have almost the same spread and shape. However, note that the mean of the bootstrap distribution will be the same as the original sample and not the population.
So, the idea of the bootstrap is that the original sample approximates the population. Therefore, resamples of the original sample would approximate the same as taking more samples from the population. That is indeed the concept of resampling.

4.13 Resampling

Resampling is a method in statistics used to estimate variability or improve model performance. In simple terms, resampling is creating new samples from existing data. There are different methods for resampling, one of them being bootstrapping. Bootstrapping is resampling with replacement. That means, when new samples are simulated, the same data point can be chosen multiple times, because when it is selected from the original sample, it is put back before simulating the next sample. That way, the number of observations n is always the same, but some data points might be chosen multiple times and others not at all.

4.14 Method

As previously mentioned, the principle of the bootstrap is to use a sample as an approximation of the population. From the samples, bootstrap samples are generated through resampling. The notation $*$ indicates a bootstrap sample. For every bootstrap sample, a statistic noted as $\hat{\theta}^*$ or $\overline{x}^*$ is calculated. The spread of the calculated $\hat{\theta}^*$ is defined as the bootstrap distribution.

Bootstrapping is also used for confidence intervals for a population parameter θ. If the values of the bootstrap distribution are known, it is possible to determine the confidence intervals by calculating the wanted percentiles. Typically a 95% confidence interval is calculated, which is done by calculating the

2.5% and 97.5% percentile. This is called the bootstrap percentile confidence interval.

4.15 Monte Carlo Principle

In the bootstrap method with n observations, there would theoretically be $\binom{2n-1}{n}$ different bootstrap samples. For large values of n, there would be a large amount of possible bootstrap samples. Therefore, it can be impractical or even impossible to generate all of them. To address this, the Monte Carlo Principle is applied, which means not every bootstrap sample has to be calculated to give precise results. Instead, a random amount of bootstrap samples is generated to approximate the distribution of the statistic of interest. Theory says that for "quick-and-dirty" resampling results 1000 samples should at least be made, but to ensure precise results, 10.000 samples or more are needed.

4.16 Parametric and nonparametric bootstrap

There are two types of bootstrapping: parametric and nonparametric. Parametric bootstrapping is when an assumption is made about the population's distribution. For example, assuming the data follows a normal distribution, then the parameters are estimated from that model, and new samples are generated by simulating data from the fitted distribution. Nonparametric bootstrapping is the opposite, when there are no assumptions made about the population's distribution. It resamples from the observed data with replacement and hereby treats the empirical distribution as an estimate for the population. In this project, the focus will be on the nonparametric bootstrap, because that is the one used.

For the nonparametric bootstrap $\hat{F}(x)$ is the empirical cumulative distribution function, given by:

$$\hat{F}(x) = \frac{1}{n}(m \leq x).$$

In addition to this, the nonparametric bootstrap also has an empirical probability mass function $\hat{f}(s)$, given by:

$$\hat{f}(s) = \frac{1}{n}(m = s).$$

In both cases m represents the observed sample, and n is the sample size. Each observation is assigned a probability of $\frac{1}{n}$ to be selected as a bootstrap sample. The cumulative distribution $\hat{F}(x)$ tells the proportion of values in the sample that are less than or equal to the value x. The probability mass function $\hat{f}(s)$ gives the frequency of the value s in the sample. The advantage of this method is that it is based on the sample, without knowing the distribution of the population. No assumptions are made, the data tells what it can, and no bias is introduced based on wrong assumptions.

To evaluate the quality of a regression model, various metrics are used. Each metric provides a score that reflects the reliability of the model's predictions.

Metrics are measures used to evaluate how well a model performs. Metrics assess the accuracy of the model's predictions compared to actual values. Common metrics like $\mathbf{R^2}$ and MBE help determine how closely predictions match real outcomes, while others like bias and variance reveal systematic errors and model stability. Metrics help model selection and improvement by providing objective feedback on performance. They're essential for comparing models, detecting underfitting or overfitting, and ensuring predictions are reliable for real-world use.

4.17 R-Squared

The R^2 (coefficient of determination) measures how much better the model predicts outcomes compared to simply using the mean. It is calculated using the Total Sum of Squares (TSS) and the Sum of Squared Errors (SSE) with this formula:

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

TSS explains how much the values of a dataset vary from the mean and is calculated by:

$$\sum_{j=1}^{n}(A_j - \bar{A})^2,$$

where A_j is the actual value and $\hat{A}$ is the mean of all actual values.

The SSE is how far the predictions from the model are from the actual values and is calculated by:

$$\sum_{j=1}^{n}(P_j - A_j)^2,$$

where $\hat{A}$ is the predicted value from the model, P_j is the predicted value, A_j is the actual value, and n is the number of observations [1]. The errors are squared to make all values positive, highlight larger mistakes more strongly, and stay consistent with how variance is calculated in statistics.

R^2 explains how much of the variance explained by the model. The value of R^2 typically lies in the range $0 \leq R^2 \leq 1$, where 0 means that the model explains none of the variance, a value of 1 would mean that the model explains all the variance, while 0.5 means that the model explains 50% of the variance. If the R^2-value is negative, it means that the model is performing so poorly that it would be better to simply predict the mean for all observations.

Although widely used, the R^2-score should not be relied upon as the only metric of model performance due to several shortcomings. Primarily, it evaluates the proportion of variance in the dependent variable explained by the model, but does not reflect the accuracy of individual values. As a result, a model can achieve a high R^2 despite making major errors on specific data points. In addition, R^2 is vulnerable to overfitting, especially when the model becomes overly complex and starts fitting the noise in the data instead of underlying trends. Another critical issue is that R^2 does not adjust for the number of predictors in the model. It will never decrease when more features are added, even if those features are irrelevant. For a more reliable assessment, alternative metrics should be used alongside R^2 to evaluate a regression model.

4.18 Mean Bias Error

Mean Bias Error (MBE) is a metric that measures the average difference between the actual values and the model's predictions. Unlike other metrics, MBE does not emphasize the magnitude of the error but instead indicates whether the model systematically overestimates or underestimates, revealing potential bias in the predictions.

The formula for MBE is:

$$\text{MBE} = \frac{1}{n}\sum_{j=1}^{n}(e_j).$$

Here $e_j = A_j - P_j$, again P_j is the predicted value, A_j is the actual value, and n is the number of observations [1].

When interpreting MBE, there are three different cases to consider. If the MBE ≥ 0, the model tends to over-predict the values compared to the actual values. If the MBE ≤ 0, the opposite is the case, and the model tends to under-predict the values. If the MBE ≈ 0 there is no consistent bias in either direction.

The MBE measures the bias of the model, whether it tends to predict values that are generally too high or too low compared to the actual values. This means that we cannot use MBE to estimate the size of the error or the overall quality of the model. If two actual values are 100 and the model predicts 120 and 80 the error calculated with $\hat{y} - y$ is $+20$ and -20. Because the absolute or squared values are not used, these errors will cancel each other out. Therefore, the size of the error is not computed. Instead, to understand the size of the errors, Root Mean Square Error can be used, which penalizes large deviations more heavily.

4.19 Root Mean Square Error

Root Mean Square Error (RMSE) is a metric used to evaluate regression models. It measures the average size of prediction errors. Since it squares the errors, it punishes larger errors more than smaller ones, which is useful if the objective is to emphasize the impact of larger deviations of the actual values.
The formula for RMSE:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} e_j^2}{n}}.$$

As before, here $e_j = A_j - P_j$, where P_j is the predicted value, A_j is the actual value, and n is the number of observations [1].
The idea is to find the error by subtracting the actual value from its prediction, square the error so negatives and positives do not cancel out (like in MBE) and large errors will stand out. Then the average of all squared errors is found, and the square root is taken to bring the result back to the original scale.

The RMSE is always a non-negative number, and the closer RMSE is to 0, the better the model is performing. The scale and unit of RMSE is the same as the target variable. So, if the data is about 'Miles Per Gallon', a RMSE of 2 means that the predictions of the model are 2 miles off, on average. Therefore, RMSE is most useful when the objective is comparing different models on the same dataset, or at least on datasets with the same scale, and in the same units. This also means an RMSE of 5 can be great if the values of the target variable range from 1-1000, but awful if the range is 1-10. Like the other metrics, RMSE should be used alongside other metrics to provide a more complete picture of the model's performance. RMSE does not explain why a model performs well or poorly. To understand why a model performs poorly, bias-variance decomposition can be used to split error into systematic bias and model instability.

4.20 Bias-Variance

To understand the sources of error in a model's predictions, the bias-variance decomposition framework helps explain errors in a model's predictions. Bias measures how far, on average, the model's predictions are from the true values. Variance describes how sensitive the model is to changes in the training data; it measures how much the model's predictions vary when trained on different datasets.

The ideal case is when both bias and variance are low, meaning the model is both accurate and stable. This framework is especially useful for diagnosing underfitting, which is caused by high bias, and overfitting, which is caused by high variance. By examining both bias and variance, the decomposition helps in understanding the trade-off between model complexity and generalization.

4.20.1 Variance

The variance of a model, for a single input, can be calculated using this formula:

$$\text{Var}[\hat{f}(x)] = \frac{1}{M}\sum_{j=1}^{M}(\hat{f}_j(x) - \overline{\hat{f}}(x))^2.$$

Where:

- M is the number of models or simulations,
- $\hat{f}_j(x)$ is the prediction by model j at input x,
- $\overline{\hat{f}}(x)$ is the mean prediction across all M models at input x.

This formula measures how much predictions vary at a specific input x across multiple model runs. To calculate the overall variance across the input space, the average of the variances at all test inputs is taken:

$$\text{Overall Model Variance} = \frac{1}{n}\sum_{i=1}^{n}\text{Var}[\hat{f}(x_i)].$$

This overall measure is beneficial because it captures how the model's prediction variability behaves across the entire dataset, rather than at just one point. A model with high variance is likely too complex and sensitive to the noise in the data.

4.20.2 Bias

Bias measures how far the model's average prediction is from the true value. Since bias can be positive or negative, it is typically squared to focus on its magnitude. The squared bias for a specific input x is calculated as:

$$\text{Bias}^2(x) = (\overline{\hat{f}}(x) - f(x))^2.$$

The overall squared bias across all inputs is calculated by averaging over the test set:

$$\text{Bias}^2 = \frac{1}{n}\sum_{i=1}^{n}(\bar{\hat{f}}(x_i) - f(x_i))^2.$$

High bias typically indicates that the model is too simple to capture the underlying structure of the data, resulting in systematic under- or over-prediction.

4.21 Confidence Intervals

Confidence intervals are used in model evaluation to judge the stability and reliability of a model's estimations. Unlike the previously mentioned metrics, confidence intervals do not provide a single number, but instead a range within which the true parameter value is likely to lie, given a specified level of confidence. The level of confidence is typically 95%.

By comparing overlap and width of the confidence intervals across different models, it is possible to assess which model generates the best estimates, based on consistency and reliability.

The validity and correctness of a regression model build upon OLS, relies on key assumptions to be fulfilled, specifically in context of this project, the key assumptions in focus are homoscedasticity (constant variance of errors) and no multicollinearity (Independence among independent variables). These key assumptions are essential for reliable statistical inference and parameter estimation. As established in section 4.5.1 and 4.5.2, when these assumptions are not upheld, then results can be misleading, such as confidence intervals and hypothesis test.
In the real world these assumption may not always be upheld, therefore this project will explore Monte Carlo Bootstrapping as a resampling method that can overcome violations in regression assumptions and how this method can offer more reliable statistical results.

- How will breaking the assumptions, homoscedasticity and no multicollinearity in a classical polynomial regression, effect the performance of the model and how can this be solved by using Monte Carlo Bootstrapping?

5 Comparison between Regressions

5.1 Setting up the models

To understand how assumption violations affect classical polynomial regression, we begin by generating synthetic data using a random number generator. First, four independent variables are created, each normally distributed with known standard deviations. Then, the dependent variable is generated as a function of these four independent variables. This synthetic dataset initially satisfies all the assumptions required for ordinary least squares (OLS) regression.

```r
set.seed(16)

# Generate independent variables
n <- 50
x1 <- rnorm(n, mean = 20, sd = 1)
x2 <- rnorm(n, mean = 15, sd = 4)
x3 <- rnorm(n, mean = 10, sd = 3)
x4 <- rnorm(n, mean = 5, sd = 5)

# Generate dependent variable with a polynomial relationship
y <- 3 + 0.8*x1^2 + 0.8*x2^3 + 0.0053*x3^4 + 0.02*x4^5 + rnorm(n,
    mean = 0, sd = 5)

# Create a data frame
data <- data.frame(y, x1, x2, x3, x4)
```

Listing 1: datageneration

Next, an error term is added that scales with the dependent variable. This introduces heteroscedasticity meaning the error variance is no longer constant thereby violating the assumption of homoscedasticity. This step is done deliberately to ensure that this is the only assumption being violated, so any observed effects on the model can be attributed specifically to this violation.

```r

# Add an error term to x variables that scales with the
    corresponding y value
x1new <- x1 + rnorm(n, mean = 0, sd = 0.1 * abs(y))
x2new <- x2 + rnorm(n, mean = 0, sd = 0.1 * abs(y))
x3new <- x3 + rnorm(n, mean = 0, sd = 0.1 * abs(y))
x4new <- x4 + rnorm(n, mean = 0, sd = 0.1 * abs(y))

# Create a new data frame
datanew <- data.frame(y, x1new, x2new, x3new, x4new)
```

Listing 2: datageneration

The dataset is then split into training and test sets, with 80% used for training and the remaining 20% for testing. A standard linear polynomial regression model is fitted to the training data. The mean bias error and root mean squared

error (RMSE) calculated are useing the testdata and standard error, and confidence intervals for the coefficients are calculated all this is to evaluate the model's performance.

```
1
2  # 5. Split into tranings data and testdata
3  set.seed(123)
4  trainindices <- sample(1:nrow(datanew), size = 0.8 * nrow(datanew))
5  traindata <- datanew[trainindices, ]
6  testdata <- datanew[-trainindices, ]
7
8  # 6. OLS-model
9  olsmodel <- lm(y ~ I(x1new^2) + I(x2new^3) + I(x3new^4) + I(x4new
       ^5), data = traindata)
10
11 # Get summary for OLS model to extract coefficient SE and CI
12 ols_summary <- summary(olsmodel)
13 ols_coefs <- ols_summary$coefficients
14
15 # Calculate Confidence Intervals for OLS coefficients
16 ols_confint <- confint(olsmodel)
17
18 # --- OLS R-squared ---
19 r_squared <- summary(olsmodel)$r.squared
20 adj_r_squared <- summary(olsmodel)$adj.r.squared
21
22 # OLS Prediction Errors
23 ols_preds_info <- predict(olsmodel, newdata = testdata, se.fit =
       TRUE, interval = "confidence")
24 olspreds <- ols_preds_info$fit[, "fit"]
25 olserrors <- olspreds - testdata$y
26 MBEOLS <- mean(olserrors)
27 RMSEOLS <- sqrt(mean(olserrors^2))
28 SEOLS_pred <- mean(ols_preds_info$se.fit)
29 CI_OLS_pred <- ols_preds_info$fit[, "upr"] - ols_preds_info$fit[, "
       lwr"]
30 AvgCIWidthOLS_pred <- mean(CI_OLS_pred)
```

Listing 3: datageneration

Following this, a bootstrap is performed by resampling the training data 10000 times. For each resampled dataset, a linear polynomial model is fitted. The coefficients from each model are stored in a new data frame, and predictions are made on the test data and also stored in the dataframe. These predictions are then compared to the actual test values to compute the mean bias error and RMSE and the coefficients are used the calculated the standard error and confidence intervals of the coefficients for the bootstrap models.

```
1
2  # 7. difin number of bootstraps
3  n_simulations <- 10000
4
```

```r
# 8. Bootstrap-model for coefficients
coef_function <- function(data, indices) {
  d <- data[indices,]
  fit <- lm(y ~ I(x1new^2) + I(x2new^3) + I(x3new^4) + I(x4new^5),
    data = d)
  return(coef(fit))
}

boot_results <- boot(data = traindata, statistic = coef_function, R
     = n_simulations)

bootstrap_coef_se <- apply(boot_results$t, 2, sd)
bootstrap_coef_ci <- t(apply(boot_results$t, 2, quantile, probs = c
    (0.025, 0.975)))


# Bootstrap Prediction Errors
bootstrappredictions <- matrix(NA, nrow = nrow(testdata), ncol = n_
    simulations)
for (i in 1:n_simulations) {
  indices <- sample(1:nrow(traindata), replace = TRUE)
  bootmodel <- lm(y ~ I(x1new^2) + I(x2new^3) + I(x3new^4) + I(
    x4new^5),
                  data = traindata[indices, ])
  bootstrappredictions[, i] <- predict(bootmodel, newdata =
    testdata)
}
bootstrapmeanpreds <- rowMeans(bootstrappredictions)
bootstraperrors <- bootstrapmeanpreds - testdata$y
MBEBootstrap <- mean(bootstraperrors)
RMSEBootstrap <- sqrt(mean(bootstraperrors^2))
SEBootstrap_pred <- mean(apply(bootstrappredictions, 1, sd))
CI_Bootstrap_pred <- t(apply(bootstrappredictions, 1, quantile,
    probs = c(0.025, 0.975)))
AvgCIWidthBootstrap_pred <- mean(CI_Bootstrap_pred[, 2] - CI_
    Bootstrap_pred[, 1])

# --- Bootstrap R-squared Distribution ---
bootstrap_r_squared <- numeric(n_simulations)
for (i in 1:n_simulations) {
  indices <- sample(1:nrow(traindata), replace = TRUE)
  bootmodel <- lm(y ~ I(x1new^2) + I(x2new^3) + I(x3new^4) + I(
    x4new^5),
                  data = traindata[indices, ])
  bootstrap_r_squared[i] <- summary(bootmodel)$r.squared
}
mean_r_squared_boot <- mean(bootstrap_r_squared)
ci_r_squared_boot <- quantile(bootstrap_r_squared, probs = c(0.025,
     0.975))
```

Listing 4: datageneration

5.2 Results

Comparing the results of the two models shown in Table 2, the Mean Bias Error (MBE) of the OLS model is -539, indicating that it tends to underpredict the

Table 2: Comparison of OLS and Bootstrap model

	OLS	**Bootstrap**
MBE	-539.4769	22.2755
RMSE	1921.0420	1377.4900
Std. Error		
Intercept	478.4811	540.9184
$I(x1^2)$	1.4306e-03	3.2636e-03
$I(x2^3)$	3.0470e-07	9.8684e-07
$I(x3^4)$	7.5112e-12	5.1212e-09
$I(x4^5)$	2.4766e-13	4.2884e-12
Coefficient (95% CI)		
Intercept	[2854.183 , 4796.919]	[2443.400 , 4556.982]
$I(x1^2)$	[0.0004 , 0.0062]	[0.0023 , 0.0128]
$I(x2^3)$	[-1.06e-06 , 1.80e-07]	[-2.70e-06 , 6.24e-07]
$I(x3^4)$	[3.34e-11 , 6.39e-11]	[4.33e-11 , 1.20e-08]
$I(x4^5)$	[-3.48e-12 , -2.47e-12]	[-4.23e-12 , 8.08e-12]

actual values. In contrast, the bootstrap model has an MBE of 22, suggesting a slight overprediction. The smaller absolute bias of the bootstrap model indicates improved accuracy. This is further supported by the Root Mean Square Error (RMSE), where the OLS model scores 1921 compared to the bootstrap model's 1377. This substantial difference reinforces the conclusion that the bootstrap model provides more accurate predictions.

When comparing the standard errors of the estimated coefficients, the OLS model appears to have lower standard errors. However, it is important to note that these may not be reliable due to the violation of the homoscedasticity assumption. In such cases, the standard errors from OLS can be misleading. In contrast, the bootstrap standard errors are derived from the empirical distribution of the data and are therefore more representative of the true variability. As a result, the confidence intervals produced by the OLS model may not be trustworthy, while those from the bootstrap model better reflect the uncertainty in the estimates showing a false sense of predictability. Overall, the bootstrap approach demonstrates superior predictive performance and more robust inference under assumption violations.

6 Discussion

6.1 Assumption violations and implications

This project has demonstrated how data that violates the assumptions of multicollinearity and heteroscedasticity in ordinary least squares (OLS) regression models can be identified. The effect of violating the assumption of homoscedasticity has been illustrated using synthetic data. It is important to note that in real-world scenarios, the violation of the homoscedasticity assumption is usually not the only issue. Nevertheless, isolating this assumption in a controlled setting allows for clearer analysis of its individual impact on model performance.

6.2 Bootstrapping as a Solution

The analysis shows that one way to build accurate regression models in the presence of assumption violations is through bootstrapping. Because it relies on the empirical distribution of the data, bootstrapping requires fewer assumptions than OLS and produces results that are easier to interpret. The bootstrap model exhibited improved performance, with a significantly lower Root RMSE of 1377 compared to 1921 for the OLS model, and a MBE of 22, compared to -539 for OLS. These results indicate better predictive accuracy and reduced bias.

6.3 Alternative approaches and their limitations

There are also alternative methods, such as Weighted Least Squares (WLS), which assign different weights to data points depending on the variance at each point. Another option is to transform the data using techniques like log transformation or square root transformation. Performing OLS regression on the transformed data can sometimes yield a more accurate model. However, it is important to understand that, like bootstrapping, these alternatives also have drawbacks. In the case of WLS, the weights for each data point are unknown and estimating them can be computationally expensive and introduce uncertainty into the model. It also complicates interpretation, as each data point is adjusted by an unknown weight. Log and square root transformations are simpler but do not always resolve assumption violations. In some cases, a log transformation may even worsen heteroscedasticity. Additionally, interpreting models based on transformed data becomes more complex, as the relationships are no longer between the original values but between their logarithms or square roots.

6.4 Strengths and limits of bootstrapping

This highlights why bootstrapping remains widely used despite its computational burden. Since it depends on the empirical distribution of the data, it requires fewer theoretical assumptions and produces results that are often more robust. However, bootstrapping also has limitations, such as deciding how many

resamples to use and, in the case of parametric bootstrapping, assuming that the sample distribution accurately represents the population. Furthermore, estimating parameters using the mean can make the model vulnerable to outliers, which can distort the results if not accounted properly for.

7 Conclusion

In this project, synthetic data was used to demonstrate the effete of violateing the assumption of homoscedasticity. on OLS polynomial regression and bootstrapped polynomial regression. The results show that the bootstrap model exhibits superior predictive accuracy, as evidenced by its lower RMSE. 1377 for the bootstrap vs. 1921 for OLS and smaller MBE 22 for the bootstrap vs. -539 for OLS. Furthermore, the standard errors and confidence intervals produced by the OLS model are shown to be unreliable, whereas those from the bootstrap method are more trustworthy, as they are based on the empirical variation in the data. Overall, the bootstrap approach proves to be excellent in terms of predictive power and robustness, especially in cases where the data is homoscedastic.

8 Litteratur

9 latextables

References

[1] Alexei Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. 2019.

[2] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, 3rd edition, 1997.

[3] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. 8, 1998.