Math 3330: Regression Notes

Kelly Ramsay

2024-06-10

Table of contents

Preface

Welcome to the MATH 3330 Notes! Please install R and R studio (Or you can use VSCode if you're comfortable there.)

In order to make the most of these notes, do all of the exercises in the order they appear.

There will likely be typos and some errors, please let me know if you encounter any.

1 Introduction

1.1 What is the course about?

1.1.1 The main question

The whole course is concerned with the following problem: Suppose that X and Y are some attributes of a population. What is the relationship between X and Y. How can we use X to predict Y, or how can we use X to explain Y?

For example, questions of this form include:

- How is location, square feet, parking available related to the price of an Airbnb?
- How is hours played and age related to win rate in League of Legends?
- How are creatine and protein consumption related to deadlift 1RM?
- How is treatment (A or B) related to pain levels of patients?

All of these can be answered with regression!

Exercise 1.1. What is X and what is Y here?

Solution 1.1.

- X: location, square feet, parking available Y: price of an Airbnb
- X: hours played and age Y: win rate in League of Legends
- X: creatine and protein consumption Y: deadlift 1RM
- X: treatment (A or B) Y: pain levels of patients

We suppose at the population level, **on average** that $Y = f(X)$. By on average, we mean that each person may not have exactly $Y = f(X)$,, but if we we average out Y for many people, we will have that the average is approximately $f(X)$. (This will be made more formal later).

For instance, consider the pain level question in the above example. Suppose that $f(A) = 2$ and $f(B) = 5$. Then, if we average the pain level of many patients who take treatment B, it should be close to 5.

Obviously, we cannot observe the whole population, and so we will assume that we have observed X and Y for a set of n individuals. Specifically, we observe some outcome $Y_1, \dots, Y_n$, which is a real number and some attributes (categorical or numeric) about the n individuals, denoted by $X_1, \ldots, X_n$. Note that here X_i can be vectors or single numbers.

1.1.2 Using our data, how can we determine f?

Other, related questions:

- What is the form of f? Is it linear?
- How can we estimate f, say with $\hat{f}$? What is the best $\hat{f}$? What is the error of $\hat{f}$ on average?
- How can we tell if our model is good? i.e. how does $\hat{f}$ fit the data?
- How can we tell which X values are important? How can we tell if X is related to Y at all?
- What is the effect of correlation of X values?

These are all questions we will answer in this course.

Statistical modelling starts as follows:

1. Question about a population, e.g., "How are hours played and age related to win rate in League of Legends?"
2. Data: $(Y_1, X_1), \ldots, (Y_n, X_n)$
3. Explore data with graphs and summary stats
4. Use exploratory data analysis to posit a model for the population.

Note that step 4 is necessary! Letting f be anything is too general and won't work well, so we need to use the data to give us a hint at the form of f! For instance, we might suppose that f is a linear function! That is, $f \in \{g(X) = X\beta \colon \beta \in \mathbb{R}^d\}$.

Next, we proceed with the following steps:

5. Estimation: How to get an estimate $\hat{\beta}$ of β?
6. Inference: What is the error of $\hat{\beta}$? Is f degenerate? I.e., is $\beta = 0$?
7. Fit: Does our fitted line match up with the data? What about the normality assumption? Do the errors appear normal?
8. Prediction: Predict any values if necessary.

1.1.3 Comparison with means example

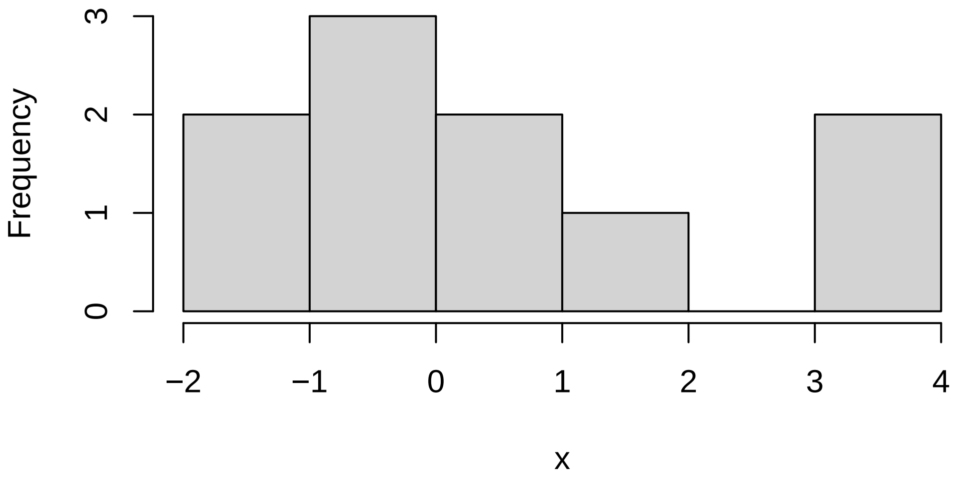
Let's compare to what we learned in previous statistics courses about two sample testing with the above steps in mind. Below we have different hours of extra sleep for two different treatments. Let's see if the sleep for groups 1 and 2 differ.

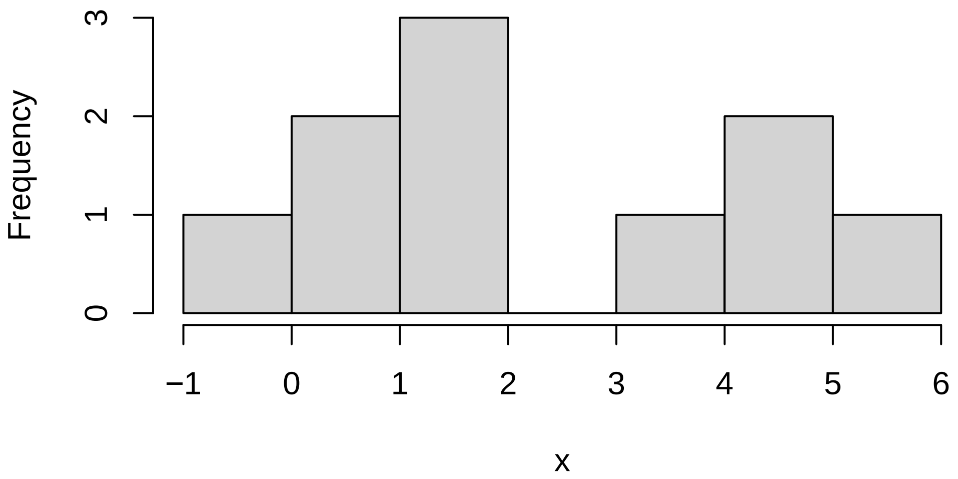
1. Do the counts for A and B differ?

```
# 2.
data('sleep')
head(sleep)
```

```
  extra group ID
1   0.7     1  1
2  -1.6     1  2
3  -0.2     1  3
4  -1.2     1  4
5  -0.1     1  5
6   3.4     1  6
```

```
# 3.
aggregate(extra ~ group, data = sleep, FUN = function(x){hist(x,main=names(x))})
```





```
Warning in format.data.frame(if (omit) x[seq_len(n0), , drop = FALSE] else x, :
corrupt data frame: columns will be truncated or padded with NAs
```

```
  group                   extra
1     1    -2, -1, 0, 1, 2, 3, 4
2     2 -1, 0, 1, 2, 3, 4, 5, 6
```

```
summary_stats = aggregate(extra ~ group, data = sleep,  FUN = summary)
aggregate(extra ~ group, data = sleep, FUN = length)
```

```
  group extra
1     1    10
2     2    10
```

We will assume that the extra hours are normal from the histograms.

Recall then that the pooled standard deviation is $\hat{\sigma}_p = \sqrt{((n_x - 1)\hat{\sigma}_x^2 + (n_y - 1)\hat{\sigma}_y^2)\big/(n_x + n_y - 2)}$ and the test statistic is:

$$T = \frac{\bar{X} - \bar{Y}}{\hat{\sigma}_p \times \sqrt{1/n_x + 1/n_y}}.$$

In addition, we have that $T \sim t_{n_x + n_y - 2}$.

```
# 5 and 6 - here these steps are the same, since we are only doing inference
t.test(sleep$extra[sleep$group==1],sleep$extra[sleep$group==2])
```

```

    Welch Two Sample t-test

data:  sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y 
     0.75      2.33 
```

```
# 7 - we checked normality earlier, 8 is not applicable
```

Here, we fail to reject the null hypothesis, and there is not enough evidence to suggest that there is a different between the groups. Notice that the p-value is 0.08, which is moderately low, so there is some evidence of a difference between the groups.

1.2 Important course information and preparation tasks

1.2.1 Prerequisite review

If you have forgotten, you should review the following concepts:

- Sample vs. population, estimates vs. parameters, hypothesis testing and confidence intervals
- Normal theory, random variables, conditional variance and expectation.
- CLT, LLN
- Linear algebra: Matrix operations, inverse, transpose etc.

1.2.2 Software

Download RStudio/R. You can use python, but I'll use R in class. If you are not familiar with R, please follow this tutorial here.

1.2.3 Outline

The course will proceed as follows:

- Review
- Core linear regression concepts
- Special Cases
- Advanced

1.2.4 Homework tasks:

- Download and install RStudio and R Software
- Think of a relationship you would want to model, what is X? what is Y?
- Review prerequisites as stated above

2 Review material

2.1 Review of random variables

Recall that

Definition 2.1. A random variable X is a function which maps outcomes $\omega \in \Omega$ to the real numbers, i.e., $X\colon \Omega \to \mathbb{R}$.

> Note
>
> Note that the notation $f\colon A \to B$ means that f is a function whose domain is A and range is B. That is, f takes a value from A and outputs some value in B.

Generally, we will just write X, and ignore the fact that X is a function.

We can categorize a random variable X as follows: - If $X\colon \Omega \to S$ where S is countable, then X is a *discrete random variable* - We say X is a *continuous random variable* if $\Pr(X = r) = 0$ for all $r \in \mathbb{R}$. - Otherwise, X is a *mixed random variable* (which we won't worry about in this course)

2.1.1 Discrete Random Variables

If $X\colon \Omega \to S$ where S is countable, then X is a discrete random variable. S can be finite, but can also be any infinite subset of the integers $\mathbb{Z}$. The distribution of X is given by its PMF, denoted by $f(x)$. For any $x \in S$, $f(x) = \Pr(X = x)$. (Note that '$\in$' means the word "in".) We must have that: - $\sum_{x \in S} f(x) = 1$, (This notation means summing over all the elements in S.) - $\forall x \in S$, $0 \leq f(x) \leq 1$. (This notation means for all x in S, $0 \leq f(x) \leq 1$.)

Examples: Binomial random variables, Poisson random variables and Geometric random variables are all discrete random variables.

Exercise 2.1. What is the PMF of a Binomial random variable? Can two different random variables have the same PMF? Why or why not?

Solution 2.1. First: $\Pr(X = x)\binom{n}{x}p^x(1-p)^{n-x}$ Second: Yes. Two random variables can be different random variables, but have the same distribution.

2.1.2 Continuous Random Variables

We say X is a *continuous random variable* if $\Pr(X = r) = 0$ for all $r \in \mathbb{R}$. If $X \colon \Omega \to S$ and X is a continuous random variable, then S is typically the real numbers, denoted by $\mathbb{R}$, but can be any uncountable subset of $\mathbb{R}$. The distribution of X is given by the PDF $f(x)$. For any interval $(a, b) \subset\in S$, $\Pr(X \in (a, b)) = \int_a^b f(x)dx$. We must have that: - $\int_{-\infty}^{\infty} f(x)dx = 1$, - $\forall x \in \mathbb{R}$, $f(x) \geq 0$.

Examples: Normal random variables, Chi- squared random variables, t random variables, Cauchy random variables, F random variables are all continuous random variables. Generally, we will focus on continuous random variables.

Exercise 2.2. What is the PMF of a Normal random variable?

Solution 2.2. $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1(x-\mu)^2/2\sigma^2}$

2.1.3 Properties of Random Variables

Let X, X_1, X_2 be random variables.

Recall the important quantities $\mathrm{E}X$, $\mathrm{var}X$, $\mathrm{cov}(X_1, X_2)$, $\mathrm{corr}(X_1, X_2)$. Recall expectation: ::: {#def-2} The expectation of a random variable X is

$$\mathrm{E}X = \sum_{x \in S} x \Pr\left(X = x\right),$$

if X is discrete and is

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

if X is continuous. ::: This is the "average'' value of the random variable. Note that it is possible for it to be impossible for $X = \mathrm{E}X$. Try to come up with an example of this!

Definition 2.2. The variance of a random variable X is

$$\mathrm{Var}\left[X\right] = \mathrm{E}\left[|X - \mathrm{E}\left[X\right]|^2\right] = \sum_{x \in S} (x - \mathrm{E}\left[X\right])^2 \Pr\left(X = x\right),$$

if X is discrete and is

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

if X is continuous.

The variance describes the variation of X about its mean. In other words, it describes on "average' ', how far is X from its mean.

Definition 2.3. The covariance between two random variables X and Y is

$$\mathrm{cov}\left[X, Y\right] = \mathrm{E}\left[(X - \mathrm{E}\left[X\right])(Y - \mathrm{E}\left[Y\right])\right].$$

The covariance describes the unnormalised linear association between X and Y.

Definition 2.4. The correlation between two random variables X and Y is

$$\mathrm{corr}\left[X, Y\right] = \mathrm{cov}\left[X, Y\right] / \sqrt{\mathrm{Var}\left[X\right] \mathrm{Var}\left[Y\right]}.$$

The correlation describes the normalized linear association between X and Y.

Next, recall that for a random variable X, its cumulative distribution function (CDF) is given by $F_X(x) = \Pr\left(X \leq x\right)$. The joint CDF of X and Y is given by $F_{XY}(x, y) = \Pr\left(X \leq x, Y \leq y\right)$.

Lastly, for a vector of d random variables $\mathbf{X} = (X_1, \ldots, X_d)$, let its CDF by $F_{\mathbf{X}}(\mathbf{x}) = \Pr\left(X_1 \leq x_1, \ldots X_d \leq x_d\right)$, where here $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x} = (x_1, \ldots, x_d)$.

We next present the concept of independence of random variables. Let $F_{XY}(x, y)$ be the joint CDF of X and Y and let F_X and F_Y be the univariate CDFs of X and Y, respectively. For two random variables X and Y, we say that X and Y are independent if $F_{XY}(x, y) = F_X(x)F_Y(y)$. More generally, two vectors of random variables $\mathbf{X}$ and $\mathbf{Y}$ are independent if $F_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) = F_{\mathbf{X}}(\mathbf{x})F_{\mathbf{Y}}(\mathbf{y})$, where A set of random variables $\{X_i\}_{i=1}^n$ are mutually independent if for any two subsets mutually exclusive subsets of $\{X_i\}_{i=1}^n$ are also independent. Note that we write $X \perp Y$ if X is independent of Y.

We have that:

Theorem 2.1.

- $X_1 \perp X_2 \implies E\left[X_1 X_2\right] = E\left[X_1\right] E\left[X_2\right]$
- $X_1 \perp X_2 \implies corr\left[X_1, X_2\right] = 0$
- $corr\left[X_1, X_2\right] = 0$ *does not imply* $X_1 \perp X_2$

Exercise 2.3. Prove Theorem 2.1 .

Let $X, X_1, X_2, \ldots X_n$ be random variables. Recall the linearity of expectation property:

Theorem 2.2. *For* $a, b \in \mathbb{R}$, *it holds that* $\mathrm{E}aX + b = a\mathrm{E}X + b$.

Exercise 2.4. Prove Theorem 2.2 .

As a corollary of Theorem 2.2 , we have that - $\mathrm{E}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i \mathrm{E}\left[X_i\right]$ - $\mathrm{Var}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i^2 \mathrm{Var}\left[X_i\right] + \sum_{i \neq j} a_i a_j \mathrm{cov}\left[X_i, X_j\right]$ - $\mathrm{Var}\left[aX_1 + bX_2 + c\right] = a^2 \mathrm{var} X_1 + b^2 \mathrm{Var}\left[X_2\right] + 2ab\mathrm{cov}\left[X_1, X_2\right]$

Exercise 2.5. What happens to $\mathrm{Var}\left[aX_1 + bX_2 + c\right]$ when $\{X_i\}_{i=1}^n$ are mutually independent?

Exercise 2.6. Let $X_1, X_2, \dots X_n$ be iid random variables with mean μ and variance σ^2. What is the mean and variance of

$$\bar{X} = \sum_{i=1}^{n} X_i / n?.$$

2.1.4 Useful properties of normal and related random variables

Let

- $\mathcal{N}(\mu, \sigma^2)$ represent the normal distribution with mean μ and variance σ^2.
- χ_k^2 be the Chi-squared distribution with k degrees of freedom
- t_n be the student-t distribution with n degrees of freedom
- $F_{m,n}$ be the F distribution with m numerator degrees of freedom and n denominator degrees of freedom

We have the following results:

Theorem 2.3. *Suppose that $X \sim \mathcal{N}(\mu, \sigma^2)$, then - $Z = \frac{X-\mu}{\sigma} \sim \mathcal{N}(0,1)$ - $Z^2 \sim \chi_1^2$.*

Let $[n] = \{1, \dots, n\}$. We also have that

Theorem 2.4.

- *If for $i \in [n]$ $Y_i \sim \chi_{k_i}^2$ and $Y_i \perp Y_j$ for $i \neq j$ then $\sum_{i=1}^n Y_i \sim \chi_{k_1 + \dots + k_n}^2$.*
- *If $Y \sim \chi_k^2$ and $Y \perp Z$, then $Z/\sqrt{Y/k} \sim t_k$.*
- *If $Y_1 \sim \chi_{k_1}^2$, $Y_2 \sim \chi_{k_2}^2$ and $Y_1 \perp Y_2$ then $\frac{Y_1/k_1}{Y_2/k_2} \sim F_{k_1,k_2}$.*

Define

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

Theorem 2.5. *Suppose that $X_1, X_2, \dots X_n \sim \mathcal{N}(\mu, \sigma^2)$ and are independent, then $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0,1)$, $\bar{X} \perp \hat{\sigma}^2$, $(n-1)\hat{\sigma}^2/\sigma^2 \sim \chi_{n-1}^2$ and $\frac{\bar{X}-\mu}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$.*

2.1.5 Central Limit Theorem

CLT: If $X_1, X_2, \dots X_n$ are i.i.d. with mean μ and variance $\sigma^2 < \infty$, then $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \overset{\mathrm{d}}{\rightarrow} \mathcal{N}(0,1)$ as $n \rightarrow \infty$.

We have that in general, for large n, regardless of the distribution of the random variables, the sample mean is approximately normally distributed.

2.1.6 Homework stop 1

Review your material and complete the above exercises before continuing to the next section.

2.2 Review of introductory statistics

The followings are some concepts that you have learned from prerequisites, and/or we have reviewed in the last two lectures.

- Sample vs. Population
- Observation vs. Random variable
- Statistic vs. Parameter
- Estimate vs. Estimator
- Estimator is a random variable and estimate is a number calculated from data
- Mean and variance of random variable
- Relationships between Normal, t , χ^2, F etc.

2.2.1 Basic premise of statistics

The whole purpose of statistics is to learn something about a population using only a sample of units from that population. A **sample** is a smaller, typically randomly selected, subset of a population. A **population** is a collection of units which we would like to know something about. For example, we may collect a sample of hamburgers from McDonald's if we want to learn something about the population of McDonald's hamburgers.

In general, at least for this course, we assume that we have access to a sample of units from a given population. Furthermore, we assume that that sample is a **random sample**. Specifically, we assume that these units in the sample are realizations of random variables. In addition, we also assume that these random variables are mutually independent. For example, we could assume that our sample $X_1, \dots, X_n$ is Normally distributed with some fixed mean μ and fixed variance σ^2. In this case, μ and σ^2 are unknown **parameters** of the population. A parameter of a population is some quantity that is a function of the distribution of our given sample. For instance, $\mathrm{E}[X_i] = \mu$. Generally, we are concerned with unknown population parameters, which are parts of the distribution that are unknown, and can only ever be estimated. For example, we may know our data is normal, but not know the mean parameter. In that case, we need to use an **estimate** of the parameter. We use a function of the data, typically called the estimator, say T, which produces the estimate, given by T computed at the sample we observed: $T(X_1, \ldots, X_n)$.

For example, to estimate μ, we typically use the sample mean. Here, the estimate is given by $\bar{X} = \sum_{i=1}^{n} X_i/n$. To be specific, the estimate is the value of $\bar{X}$ and the estimator T is the function that maps n real numbers to their mean. In general, estimates are used to give our 'best guess' at population parameters.

2.2.2 Confidence intervals:

Recall from the previous section that our estimate of a parameter is only that, an estimate. In other words, it is not exactly equal to the population parameter. For instance, if we drew a different sample our estimate would change. A confidence interval is used to acknowledge this phenomenon in the reporting of our statistics. Its used to give a range of estimates that we might have obtained from any "regular" sample we might observe. It is ultimately used to quantify the error (sometimes called uncertainty) in our estimate.

Confidence intervals consist of a level, usually denoted by $(1 - \alpha)100\%$ and two end points. For example, you have learned confidence intervals for the population mean. When we say $(-1, 1)$ is 95% confidence interval for the population mean, what does this mean? Colloquially, it means that we expect the sample mean to be somewhere within $(-1, 1)$ with high confidence. Note that confidence intervals are computed from the data, which means also that for each new sample, we would get a different confidence interval. However, the population parameter never changes. Therefore, the interval is what is varying from sample to sample. This impacts the interpretation of a confidence interval.

Continuing our example, we have that the interval $(-1, 1)$ can be interpreted as: "if we drew many more samples, 95% of the **intervals** will contain the population parameter." We **do not** say that the parameter has a 95% chance of falling in (-1,1), since the parameter is not random, the interval end points are.

For example, we have the formula for a confidence interval for the population mean is given by: $\bar{X} \pm 1.96\hat{\sigma}$. Notice that it is based only on the data. Therefore, it will change if we drew a new sample.

To summarize this section, a confidence interval is used to quantify the uncertainty in our reported estimates. By uncertainty, we specifically mean the uncertainty resulting from the fact that we have only a sample of the population, and our estimate varies depending on the sample.

2.2.3 Hypothesis tests:

Hypothesis tests are used to determine whether an effect is spurious or a real property of the population. A spurious effect is one that is specific to the sample we observed, and is not a real property of the population. For example, if the heights of males and female students are measured, and we observe that the sample mean of both male and females are equal, then this would be a spurious effect. We know that the population heights of males and females are substantially different. If we drew a new sample, we would likely observe something that mirrors the population reality (provided it is large enough).

Formally, a hypothesis test compares two competing beliefs about a population parameter, called the null and alternative hypothesis. For instance, we may wish to test whether the population heights of men is greater than women, vs. the heights being less than or equal to that of men.

We write this as follows: $H_0 \colon \mu_{men} \leq \mu_{women}$ vs. $H_a \colon \mu_{men} > \mu_{women}$.

The null hypothesis is usually chosen to be one such that if we make a mistake, the error is most serious. However, it is usually clear from the context.

In general, we compute a test statistic and its distribution **under the null hypothesis.** Then we compute how likely it was to see the observed test statistic we saw, if the null hypothesis was true. This is likelihood is given by the **p-value**. If it was sufficiently unlikely (in other words, the p-value is less than the threshold α), then we reject the null hypothesis. Otherwise, we fail to reject the null hypothesis. If we fail to reject the null hypothesis then either the null hypothesis is true, it is not true, but there was not enough data collected to show the effect.

There are two types of errors we can make in a hypothesis test: Type 1 and Type 2 error. Type one error occurs when we reject the null hypothesis when it is true. Type two error occurs when we fail to reject the null hypothesis when the alternative is true.

Let's do an example.

Exercise 2.7. In a study about online dating, you are interested in determining the average age of individuals who use online dating platforms. You want to know whether the average age of online daters is significantly different from 30. You have a dataset of 40 ages of people using online dating platforms.

How would you answer this question?

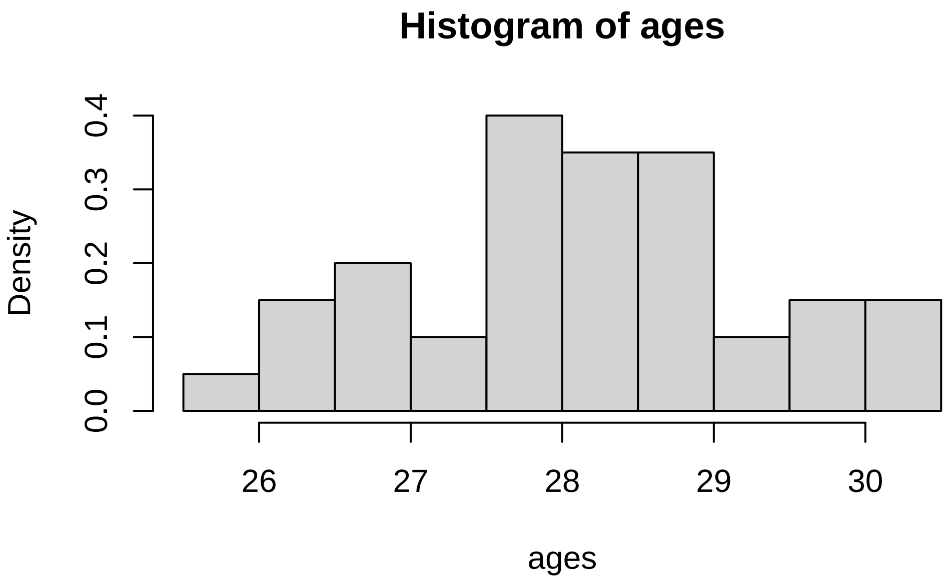
$$H_0 \colon \mu = 30 \qquad vs. \qquad H_1 \colon \mu \neq 30.$$

First, we can explore the data:

```r
getwd()
```

```
[1] "C:/Users/12RAM/OneDrive - York University/Teaching/Courses/Math 3330 Regression/Math 33
```

```r
ages=read.csv('C:\\Users\\12RAM\\OneDrive - York University\\Teaching\\Courses\\Math 3330 Reg



hist(ages,freq=F)
```



Now, assume that $X_1, \ldots, X_{40} \sim \mathcal{N}(\mu, \sigma^2)$, and independent. (We can justify normality with the histogram, or we could also invoke the CLT to get normality of the sample mean (not the data itself).) Therefore, we can do a one sample t-test. Recall that, under the null hypothesis, we have $\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$. This means that if $\left|\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}}\right| \geq t_{n-1,1-\alpha/2}$, then we reject the null hypothesis! Here, $t_{n-1,1-p}$ is the $(1-p)$th quantile of the t_{n-1} distribution. For large n and $p = 0.025$, this is roughly equal to 2.

Now, recall that

$$H_0 : \mu = 30 \qquad vs. \qquad H_1 : \mu \neq 30$$

We have that $\left|\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}}\right| = 66.234$. Using R, we get that the p-value is $< 2.2 \times 10^{-16}$.

```r
# Calculate the mean of the 'ages' data and assign it to xbar
xbar = mean(ages)
xbar  # Print the mean
```

```
[1] 28.16378
```

```r
# Calculate the variance of the 'ages' data and assign it to ssq
ssq = var(ages)
ssq  # Print the variance
```

```
[1] 1.277377
```

```r
# Calculate the length (number of observations) of the 'ages' data and assign it to n
n = length(ages)
n  # Print the number of observations
```

```
[1] 40
```

```r
# Set the significance level
alpha = 0.05

# Perform a two-sided t-test to check if the mean of 'ages' is significantly different from 3
# t.test() is the function for performing t-tests in R
test = t.test(ages, mu = 30, alternative = 'two.sided')
test  # Print the test result
```

```

    One Sample t-test

data:  ages
t = -10.275, df = 39, p-value = 1.179e-12
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 27.80232 28.52524
sample estimates:
mean of x
 28.16378
```

Here the p-value measures how much evidence there is against the null hypothesis. If the p-value is very small, then this constitutes strong evidence against the null hypothesis. If the p-value is small, but closer to 0.05, then there is evidence against the null. If it is larger, but still small, say 0.1, then this is weak evidence against the null hypothesis. It is not helpful to throw it away if it is above 0.05, therefore we should not just take $\alpha = 0.05$. Choosing α depends on how serious a type 1 error is. If it is not that serious, we can take α larger. If it is very serious, we can take α smaller.

In this example, there is very strong evidence against the null hypothesis.

Note

Note also that we can use the confidence interval method with

$$\bar{X} \pm t_{n-1,1-\alpha/2}\sqrt{\hat{\sigma}^2/n}.$$

```
# Alternative method to calculate the confidence interval
# ci will store the confidence interval values
ci = xbar + c(-1, 1) * qt(1 - alpha / 2, n - 1) * sqrt(ssq / n)
ci  # Print the confidence interval
```

```
[1] 27.80232 28.52524
```

Note

Moving beyond the one-sample testing problem, we might be interested in other population parameters, say $\theta \in \Theta$. Think Lecture 1: $\mathrm{E}[Y|X] = \beta_0 + X\beta_1$, we might want to estimate $\mathrm{E}[Y|X]$, which amounts to $\beta_0, \beta_1 \in \mathbb{R}$! In general, we may estimate θ by $\hat{\theta}$. Then we may compute the variance and distribution of $\hat{\theta}$. From there, we can make confidence intervals and conduct hypothesis tests etc.

Let's do another example:

Exercise 2.8. In a study about online dating, you are interested in determining if the average age of those who identify as men who use online dating platforms differs from those who identify as women. You have a dataset of 20 ages of each group using online dating platforms.

What is the population parameter of interest here? It is $\Delta = \mu_1 - \mu_2$, the difference in means between the two populations. Now, suppose that $X_1, \ldots, X_{20} \sim \mathcal{N}(\mu_1, \sigma^2)$ and $Y_1, \ldots, Y_{20} \sim \mathcal{N}(\mu_2, \sigma^2)$, and are mutually independent. (We could also invoke the CLT instead of assuming normality.) We can estimate those parameters with **estimates.** For instance, $\bar{X}$, $\bar{Y}$,

$$\hat{\sigma}^2 = \frac{(n_1 - 1)\hat{\sigma}_1^2 + (n_2 - 1)\hat{\sigma}_2^2}{n_1 + n_2 - 2}.$$

Exercise 2.9. Suppose that $X_1, \ldots, X_{20} \sim \mathcal{N}(\mu_1, \sigma^2)$ and $Y_1, \ldots, Y_{20} \sim \mathcal{N}(\mu_2, \sigma^2)$, and are mutually independent. Compute $\operatorname{Var}\left[\bar{X} - \bar{Y}\right]$.

Solution 2.3. Using independence of $\bar{X}$ and $\bar{Y}$ and the result of the Exercise 2.6 , we have that

$$\operatorname{Var}\left[\bar{X} - \bar{Y}\right] = \operatorname{Var}\left[\bar{X}\right] + \operatorname{Var}\left[\bar{Y}\right] = \sigma_1^2/n_1 + \sigma_1^2/n_2.$$

First, we write down the null and alternative hypothesis:

$$H_0 \colon \Delta = 0 \qquad vs.. \qquad H_1 \colon \Delta \neq 0.$$

Here, we can do a two sample t-test.

Recall that the pooled variance is given by:

$$\hat{\sigma}_p^2 = \frac{(n_1 - 1)\hat{\sigma}_1^2 + (n_2 - 1)\hat{\sigma}_2}{(n_1 + n_2 - 2)}$$

We previously said that a multiple of a one sample standard deviation follows a Chi-squared distribution. It follows that $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 \sim \chi^2_{n_1 - 1}$ and $(n_2 - 1)\hat{\sigma}_2^2/\sigma^2 \sim \chi^2_{n_2 - 1}$. Using the theory from here, specifically, $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 + (n_2 - 1)\hat{\sigma}_2^2/\sigma^2$ is a sum of independent Chi-squared random variables, and so we have $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 + (n_2 - 1)\hat{\sigma}_2^2/\sigma^2 \sim \chi^2_{n_1 + n_2 - 2}$.

Again, using the theory from here, under the null hypothesis, we have that

$$\frac{\bar{X} - \bar{Y}}{\hat{\sigma}_p/\sqrt{1/n_1 + 1/n_2}} = \frac{(\bar{X} - \bar{Y})/\sigma\sqrt{1/n_1 + 1/n_2}}{\hat{\sigma}_p/\sigma\sqrt{1/n_1 + 1/n_2}} \sim t_{n_1 + n_2 - 2}.$$

This follows from 3 facts, first, letting $Z = (\bar{X} - \bar{Y})/\sqrt{\operatorname{Var}\left[\bar{X} - \bar{Y}\right]}$, note that $Z \sim \mathcal{N}(0, 1)$. We have that

$$Z = (\bar{X} - \bar{Y})/\sqrt{\operatorname{Var}\left[\bar{X} - \bar{Y}\right]} = (\bar{X} - \bar{Y})/\sigma\sqrt{1/n_1 + 1/n_2}.$$

Next, we said earlier that $\bar{X}$ is independent of $\hat{\sigma}_1$ and $\bar{Y}$ is independent of $\hat{\sigma}_2$. Now, recall that if two random variables are independent, then any function of them is also independent. In other words, if X and Y are independent, then for real functions f and g, we have that $g(X)$ is independent of $f(Y)$. It follows that $\bar{X}$ is independent of $\hat{\sigma}_2$ and $\bar{Y}$ is independent of $\hat{\sigma}_1$. It follows that $\bar{X} - \bar{Y}$ is independent of $\hat{\sigma}_p$. Then,

$$\frac{(\bar{X} - \bar{Y})/\sigma\sqrt{1/n_1 + 1/n_2}}{\hat{\sigma}_p/\sigma\sqrt{1/n_1 + 1/n_2}}$$

is a ratio of a standard normal random variable and the square root of a Chi-squared random variable, divided by its degrees of freedom. Further, the numerator and denominator are independent. Therefore, the above quantity follows a t distribution with $n_1 + n_2 - 2$ degrees of freedom.

This means that if $\left|\frac{\bar{X}-\bar{Y}}{\hat{\sigma}_p/\sqrt{1/n_1+1/n_2}}\right| \geq t_{n_1+n_2-2,1-\alpha/2}$, then we reject the null hypothesis.

Let;s execute the test in R:

```
# Normally, I will give you a dataset. Here I generate the data
set.seed(440)
female_ages=rnorm(20,28,4)
male_ages=rnorm(20,32,4)



# Check for equal variance
var(female_ages)
```

```
[1] 15.72805
```

```
var(male_ages)
```

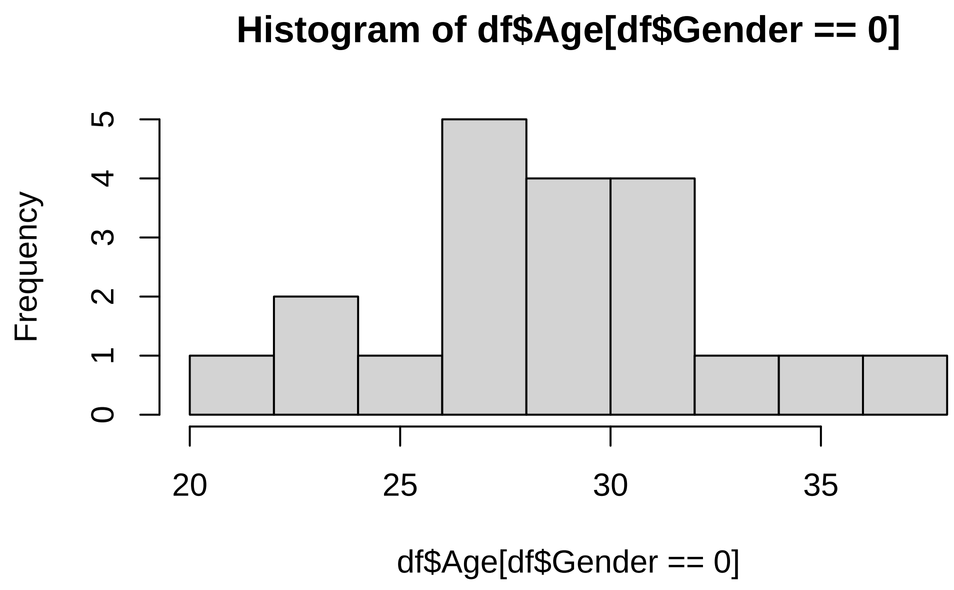
```
[1] 26.22371
```

```
## Putting the data in a dataframe
cbind("Age"=c(female_ages,male_ages),"Gender"=rep(c(0,1),each=20))
```

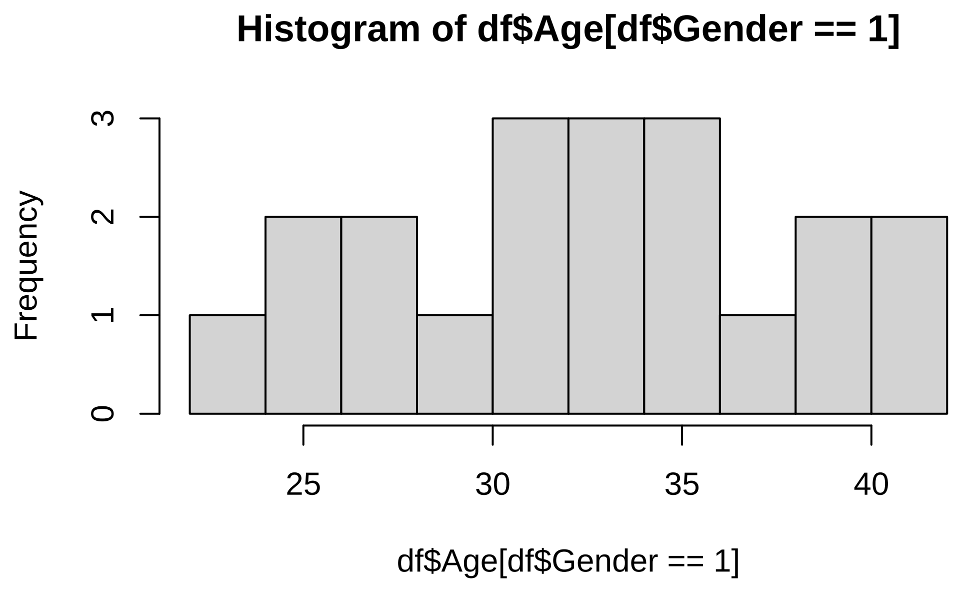
```
           Age Gender
 [1,] 37.19809      0
 [2,] 20.69693      0
 [3,] 27.80284      0
 [4,] 27.69463      0
 [5,] 29.53143      0
 [6,] 29.46190      0
 [7,] 30.41164      0
 [8,] 33.27790      0
 [9,] 22.65974      0
[10,] 30.73540      0
[11,] 34.08564      0
[12,] 27.58077      0
[13,] 23.26108      0
[14,] 30.94523      0
```

```
[15,] 31.52404      0
[16,] 29.13246      0
[17,] 26.95470      0
[18,] 24.80749      0
[19,] 28.60051      0
[20,] 26.76294      0
[21,] 25.94775      1
[22,] 40.16080      1
[23,] 25.58905      1
[24,] 32.16780      1
[25,] 29.87934      1
[26,] 35.46593      1
[27,] 35.71651      1
[28,] 37.76510      1
[29,] 27.23068      1
[30,] 33.41994      1
[31,] 40.43822      1
[32,] 31.04841      1
[33,] 32.66165      1
[34,] 38.28678      1
[35,] 34.72411      1
[36,] 39.57994      1
[37,] 26.85585      1
[38,] 31.87533      1
[39,] 23.71793      1
[40,] 30.54803      1
```

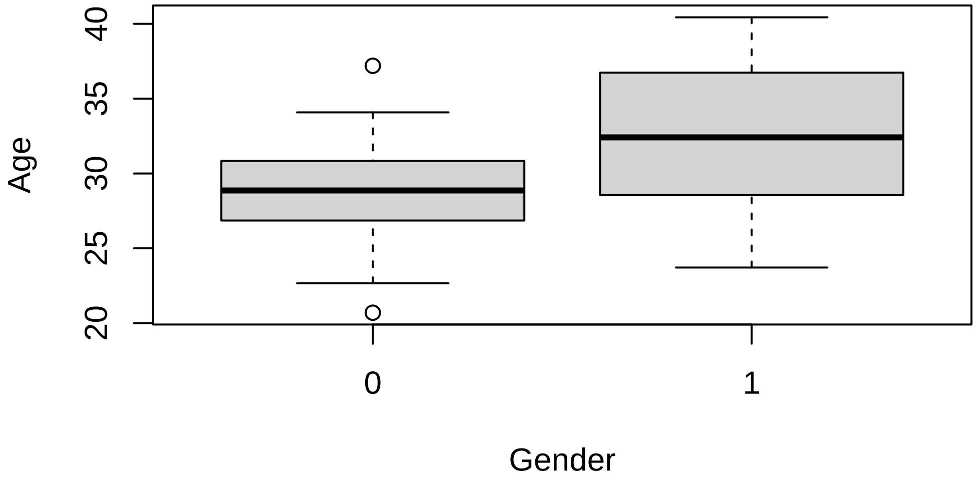
```r
df=data.frame(cbind("Age"=c(female_ages,male_ages),"Gender"=rep(c(0,1),each=20)))

#exploring the data
#hist(x) creates a histogram of the vector x

hist(df$Age[df$Gender==0])
```



```r
hist(df$Age[df$Gender==1])
```



```r
#boxplot creates boxplots of Age against gender
boxplot(Age~Gender, df)
```



```r
test=t.test(Age~Gender,data=df,var.equal=TRUE)
test
```

```

	Two Sample t-test

data:  Age by Gender
t = -2.7603, df = 38, p-value = 0.008841
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -6.929630 -1.065749
sample estimates:
mean in group 0 mean in group 1 
       28.65627        32.65396 
```

```r
#Interpret the P value, and CI, what are we going to say to a stakeholder?

#e.g.
test$estimate
```

```
mean in group 0 mean in group 1
       28.65627        32.65396
```

Note

Note also that we can use the confidence interval method, meaning that if if 0 is in the interval:

$$\hat{\Delta} \pm t_{n_1+n_2-2,1-\alpha/2}\sqrt{\hat{\sigma}_1^2/n_1 + \hat{\sigma}_2^2/n_2},$$

then we fail to reject the null hypothesis.

2.2.4 Homework stop 2

Exercise 2.10. IBM Human Resources (HR) department is evaluating job applicants from York University.
They are interested to know if the 2020 ITEC graduating class has an average GPA higher than 6 (i.e. average GPA higher than “B’’). They collected the GPA of 25 ITEC students graduated in 2020.

4.92	4.79	6.76	5.64	6.12	7.37	6.45	6.31	6.68
6.30	4.91	6.95	5.87	6.18	6.60	6.71	6.69	5.62
6.40	5.51	6.44	6.13	8.55	7.94	4.78	-	-

Tip

Use chatGPT to convert the above table to an R vector, so you don’t have to waste time!

- For the one sample testing problem, i.e., you have a sample of n normal random variables, with unknown mean and variance and you want to test whether $H_0: \mu = 0$ vs. $H_0: \mu \neq 0$, show that $\frac{\bar{X}}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$ under the null hypothesis.
- What is the distribution of each of the following: $\bar{X}, \bar{Y}, \hat{\sigma}$ under the assumption of normal data with unknown mean and variance?

Compare and contrast the following concepts. That is, define them and explain the difference between them.

- Sample vs. Population
- Observation vs. Random variable
- Statistic vs. Parameter
- Estimate vs. Estimator

2.3 Review of matrices and linear algebra

Recall that

Definition 2.5. An $(n \times m)$ matrix A takes the form

$$\begin{aligned} A &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \\ &= ((\ a_{ij}\ )) \qquad i = 1, \dots, n, \quad j = 1, \dots, m \end{aligned}$$

and a_{ij} is the element in the i^{th} row and j^{th} column of the matrix A

We also define the following:

- An $(n \times 1)$ matrix is also known as a n dimensional column vector. Note: in this course, a vector means a column vector.
- A $(1 \times m)$ matrix is also known as a m dimensional row vector
- The n dimensional one vector, 1_n, (sometimes the subscript n is suppressed when the dimension is obvious), is an n dimensional column vector with all entries being 1.
- The $(n \times n)$ identity matrix, I_n, is the $(n \times n)$ matrix with diagonal entries set equal to 1 and the off diagonal entries set equal to 0

Throughout this section, we will use the following matrices to demonstrate the numerical calculations:

$$U = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & -2 \end{pmatrix}, \ V = \begin{pmatrix} 2 & 4 \\ 1 & -2 \\ -1 & 0 \end{pmatrix}, \ k = 4$$

2.3.1 Matrix properties

First, we define the transpose of a matrix:

Definition 2.6. Let $A = ((\ a_{ij}\ ))$ for $i = 1, \dots, n$ and $j = 1, \dots, m$, is an $(n \times m)$ matrix. Then $A^\top = A$ transpose $= ((\ a_{ji}\ ))$ for $j = 1, \dots, m$ and $i = 1, \dots, n$, and $A^\top$ is an $(m \times n)$ matrix.

When we transpose a matrix A, the rows of A becomes the columns of $A^\top$ and the columns of A becomes the rows of $A^\top$.

Example 2.1. Using our example matrices, we have that

$$U^\top = \begin{pmatrix} 1 & -1 \\ 2 & 4 \\ 3 & -2 \end{pmatrix}, \; V^\top = \begin{pmatrix} 2 & 1 & -1 \\ 4 & -2 & 0 \end{pmatrix}$$

Definition 2.7. Let $A = ((\ a_{ij}\ ))$ and $B = ((\ b_{ij}\ ))$ be two $(n \times m)$ matrices. Then

$$A \pm B = ((\ a_{ij} \pm\ b_{ij}\ )).$$

Addition and subtraction of matrices required the matrices to have the same dimension.

Example 2.2. Using our example matrices, we have that: $U + V$ is undefined because they are not of the same dimension, and

$$U + V^\top = \begin{pmatrix} 1+2 & 2+1 & 3+(-1) \\ (-1)+4 & 4+(-2) & (-2)+0 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 2 & -2 \end{pmatrix}$$

Definition 2.8. Let $A = ((\ a_{ij}\ ))$ for $i = 1, \dots, n$ and $j = 1, \dots, m$, is an $(n \times m)$ matrix and k is a constant. Then

$$kA = ((\ ka_{ij}\ )) = Ak,$$

i.e. each element of the matrix A is multiplied by k.

Example 2.3. Using our example matrices, we have that:

$$kU^\top = 4 \begin{pmatrix} 1 & -1 \\ 2 & 4 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 4(1) & 4(-1) \\ 4(2) & 4(4) \\ 4(3) & 4(-2) \end{pmatrix} = \begin{pmatrix} 4 & -4 \\ 8 & 8 \\ 12 & -2 \end{pmatrix}$$

Definition 2.9. Let A and B be two matrices. Then A multiplied by B, AB, is defined only if (number of columns of A) = (number of rows of B).
The product is a ((number of rows of A) × (number of columns of B)) matrix.

More precisely, let $A = ((\ a_{ij}\ ))$ be an $(n \times m)$ matrix and $B = ((\ b_{ij}\ ))$ be an $(m \times p)$ matrix. Then $C = AB = ((\ c_{ij}\ ))$ is an $(n \times p)$ matrix with

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj}$$

ℹ Note

In matrix algebra, AB is not necessarily equal to BA.

Example 2.4. Using our example matrices, we have that:

$$\begin{aligned} UV &= \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & -2 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 1 & -2 \\ -1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1(2)+2(1)+3(-1) & 1(4)+2(-2)+3(0) \\ (-1)(2)+4(1)+(-2)(-1) & (-1)(4)+4(-2)+(-2)(0) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 4 & -12 \end{pmatrix} \end{aligned}$$

Assume all the matrix multiplication works. Let I_n be an $(n \times n)$ identity matrix. Then

$$AI_n = A, \quad \textit{and} \quad I_n B = B.$$

Definition 2.10. Let A be an $(n \times n)$ matrix. The inverse of A, A^{-1}, if exists satisfies

$$AA^{-1} = A^{-1}A = I_n$$

and if A^{-1} does not exist, then A is a singular matrix.

> **!** Important
>
> From your linear algebra course, a prerequisite, you have learned the condition(s) for the existence of an inverse, (https://mathworld.wolfram.com/InvertibleMatrixTheorem.html)[The Invertible Matrix Theorem] and you have learned how to obtain an inverse. You should review them. Specifically, you should know how to obtain inverse of any diagonal matrix and any (2×2) non-singular matrix, i.e.,
>
> $$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Example 2.5. Using our example matrices, let

$$W = UV = \begin{pmatrix} 1 & 0 \\ 4 & -12 \end{pmatrix}$$

Then

$$W^{-1} = \frac{1}{1(-12) - 0(4)} \begin{pmatrix} -12 & 0 \\ -4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1/3 & -1/12 \end{pmatrix}.$$

You can verify that $WW^{-1} = W^{-1}W = I_2$.

2.3.2 Important identities

Lastly, we introduce some important identities:

$$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad and \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Then

$$X^\top X = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix}, \quad \text{and} \quad X^\top y = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}.$$

Also $\bar{y} = \frac{1}{n} 1^\top y$ and $\sum_{i=1}^n y_i = n\bar{y}$ \ Finally $\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - n\bar{y}^2$

These are useful identities that we will use throughout this course.

Lastly, we recall an important application of matrices. An application of matrices: Suppose that we want to solve for x_1, x_2, x_3 where they satisfy the following set of linearequations:

$$\begin{aligned} 2x_1 + 3x_2 - 4x_3 &= 0 \\ -x_1 + 4x_2 &= -1 \\ 5x_1 + x_2 - 2x_3 &= 4 \end{aligned}$$

We can set it up in matrix form as follows:

$$\begin{pmatrix} 2 & 3 & -4 \\ -1 & 4 & 0 \\ 5 & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix}$$

Or it can be presented as $Ax = b$. If A is not a singular matrix, then $x = A^{-1}b$. Since $det(A) = 62$, it is not a singular matrix. Solving the above equation this using $x = A^{-1}b$ yields that $x = (1, 0, 0.5)^\top$.

Keep this in mind, we will see it return in the next chapter.

2.3.3 Homework stop 3

Exercise 2.11. Let

$$W = \begin{pmatrix} 3 & 2 \\ -4 & 6 \end{pmatrix}$$

and $x = (2, 1)^\top$. Compute W^{-1}, $xx^\top$ and $x^\top W$. Verify that $WW^{-1} = W^{-1}W = I_2$.

- Prove each of the important identities.

- Verify $X^\top A = (A^\top X)^\top$.
- What is the rank of a matrix? Is a matrix's rank related to whether or not a matrix is invertible? Why?
- Define a positive definite matrix. When is $X^\top X$ positive definite?

2.4 Review Random Vectors

2.4.1 Definition of random vectors

Definition 2.11. Let $Y_1, \cdots, Y_n$ be random variables. Then

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

is an n-dimensional random vector.

Similar to a random variable, a random vector also comes with a probability mass function (if all the Y_i are discrete) or a probability density function (if all the Y_i are continuous), or a "mixture'' distribution (if some Y_i are discrete and others are continuous). In general, a random vector is drawn from a multivariate distribution, defined by the PMF or PDF. Just as before, the PMF and PDF range is non-negative, the PMF sums to 1 over all outcomes, and the PDF integrates to 1 over $\mathbb{R}^n$. One discrete multivariate distribution you have learned in 1131 is the Multinomial distribution. We will learn about the multivariate normal distribution soon.

2.4.2 Expected Value and Covariance

Definition 2.12. Let Y be an n-dimensional random vector, then the mean (expected value) of Y is defined as

$$\mathrm{E}(Y) = \begin{pmatrix} \mathrm{E}(Y_1) \\ \vdots \\ \mathrm{E}(Y_n) \end{pmatrix} = \mu$$

and the covariance of Y is defined as

$$\mathrm{cov}\left[Y\right] = \mathrm{E}[(Y-\mu)(Y-\mu)^\top] = ((\mathrm{cov}\left[Y_i, Y_j\right]\ )) = \Sigma.$$

Sometimes $\mathrm{cov}\left[Y\right]$ is written as $\mathrm{Var}\left[Y\right]$.

The following are some facts about Σ:

Σ is an $n \times n$ matrix with the diagonal elements being the variances, $\mathrm{Var}\left[Y_i\right]$ for $i = 1, \ldots, n$, and the off-diagonal elements being the covariances, $\mathrm{cov}\left[(\right] Y_i, Y_j)$ for $i, j = 1, \ldots, n$ and $i \neq j$. Σ is a symmetric, non-negative definite matrix. In this course, we further restrict it to be a positive definite matrix. Σ is referred to as the **covariance matrix**.

2.4.3 Properties of expected value and covariance

Let $Y \in \mathbb{R}^d$ be a random vector with $A \in \mathbb{R}^d$ and $B \in \mathbb{R}^{n \times d}$ be matrices. It holds that

- $\mathrm{E}(A + BY) = A + B\mathrm{E}(Y)$
- $\mathrm{cov}\left[A + BY\right] = B\mathrm{cov}\left[Y\right]B^\top$.

Exercise 2.12. Let $Y = (Y_1, ..., Y_n)^ $ be a random vector, where Y_i are i.i.d. random variables with mean μ and variance σ^2. What are the mean and covariance of Y? Use properties of random vectors to compute the mean and variance of the sample mean.

Solution 2.4. First, $\mathrm{E}(Y) = \mu 1$ and $\mathrm{cov}\left[Y\right] = \sigma^2 I$.Note that $\bar{Y} = (Y_1 + \cdots + Y_n)/n = \frac{1}{n}1^\top Y$. Now, we have

$$\mathrm{E}(\bar{Y}) = \mathrm{E}\left(\frac{1}{n}1^\top Y\right) = \frac{1}{n}\left(1^\top \mathrm{E}(Y)\right) = \frac{1}{n}(n\mu) = \mu$$

and,

$$\begin{aligned}
\mathrm{cov}\left[\bar{Y}\right] &= \mathrm{cov}\left[\frac{1}{n}1^\top Y\right] \\
&= \left(\frac{1}{n}\right)^2 \left(1^\top \mathrm{cov}\left[Y\right] 1\right) \\
&= \left(\frac{1}{n}\right)^2 (n\sigma^2) = \frac{\sigma^2}{n}.
\end{aligned}$$

2.4.4 Multivariate normal distribution

We say that a random vector $X \sim \mathcal{N}_d(\mu, \Sigma)$ follows a multivariate normal distribution if X has PDF:

$$\phi(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{d/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)'\Sigma^{-1}(\mathbf{x} - \mu)\right\}.$$

If $X \sim \mathcal{N}_d(\mu, \Sigma)$ and $c \in \mathbb{R}^d$, $A \in \mathbb{R}^{m \times d}$ then:

- $AX \sim \mathcal{N}(A\mu, A\Sigma A^\top)$.
- $c^\top X \sim \mathcal{N}(c^\top \mu, c^\top \Sigma c)$.

- Any conditional distribution for a subset of the variables conditional on another subset of variables is a multivariate distribution.

Using random vectors is a simple way of deriving lots of equations for this course. Working with vectors also allows those who are "geometrically gifted" to view the whole regression concepts geometrically! If not, not to worry!

2.4.5 Homework stop 4

Exercise 2.13. For a (full-rank) matrix $X \in \mathbb{R}^{n \times p}$ with $n > p$, and random vector $Y \in \mathbb{R}^{n \times 1}$ with mean μ and covariance Σ, compute the following:

- Expected value and covariance of $(X^\top X)^{-1} X^\top Y$
- Expected value of $Y^\top Y$
- Expected value and covariance of $X^\top X$
- Expected value and covariance of $X(X^\top X)^{-1} X^\top Y$

3 Linear Regression

3.1 Basics of linear regression

By the end of this section, you should be able to say what the linear and normal linear regression models are. As well as what it means to assume either of these models.

3.1.1 The linear regression model

Consider the following example.

Example 3.1. It is difficult to accurately determine a person's body fat percentage without immersing them in water. However, we can easily obtain the weight of a person. A researcher would like to know if weight and body fat percentage are related? If so, for a given weight, can the person's body fat percentage be predicted? If so, how accurate is the prediction? This researcher collected the following data:

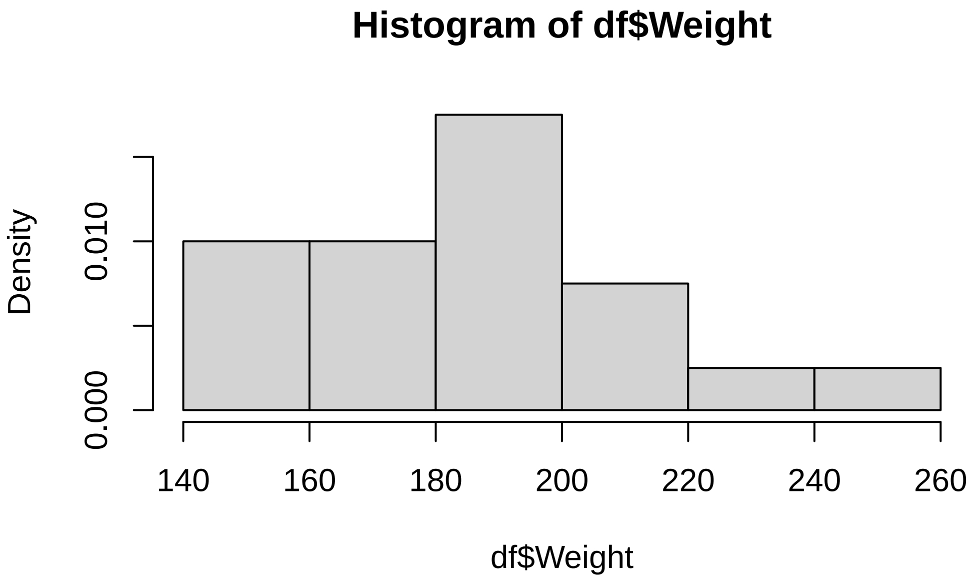
Individual	1	2	3	4	5	6	7	8	9	10
Weight (lb)	175	181	200	159	196	192	205	173	187	188
Body Fat (%)	6	21	15	6	22	31	32	21	25	30

Individual	11	12	13	14	15	16	17	18	19	20
Weight (lb)	188	240	175	168	246	160	215	159	146	219
Body Fat (%)	10	20	22	9	38	10	27	12	10	28

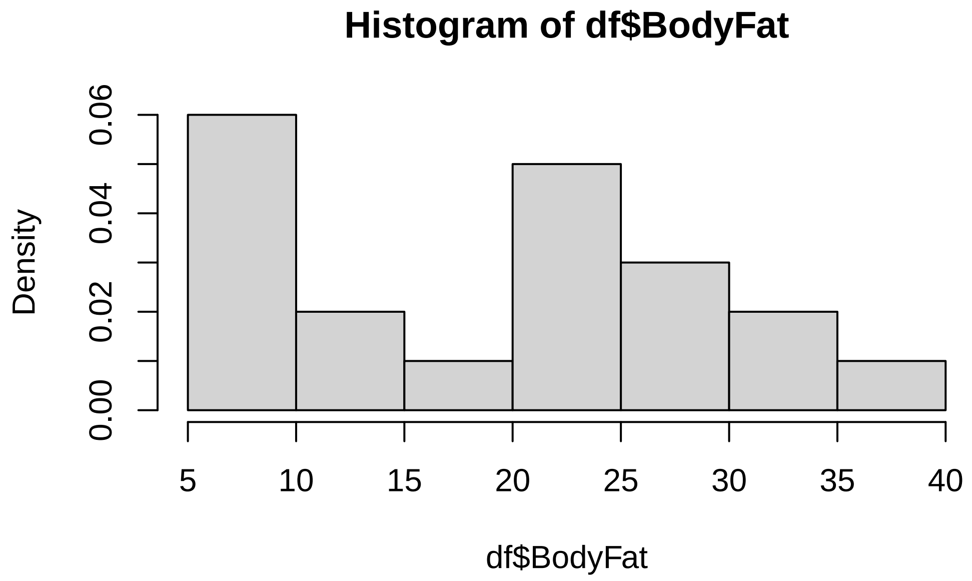
How can we (as statisticians / data scientists) answer the questions raised by the researcher?

The first thing we might do is explore the data:

```
##################### Exploratory analysis

# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))

# make some histograms
hist(df$Weight,freq=F)
```



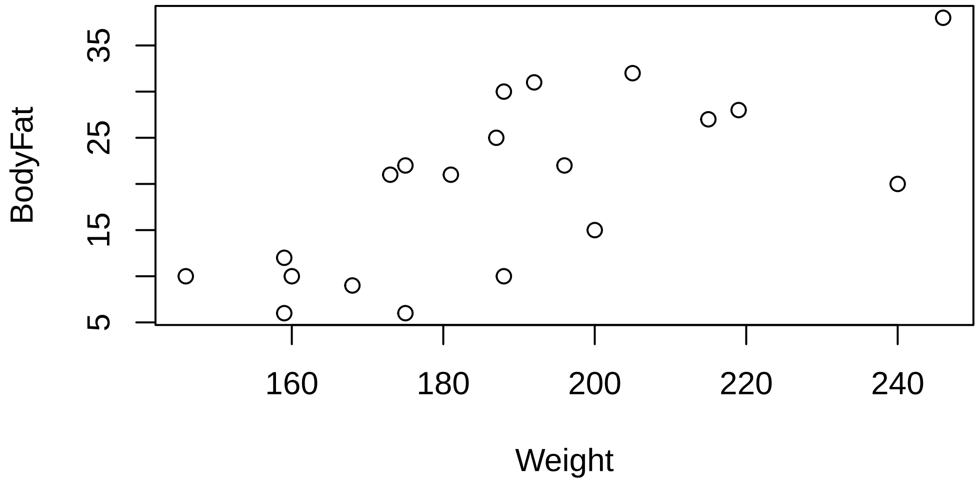
```
hist(df$BodyFat,freq=F)
```



```
# print summary statistics
summary(df)
```

```
     Weight         BodyFat     
 Min.   :146.0   Min.   : 6.00  
 1st Qu.:171.8   1st Qu.:10.00  
 Median :187.5   Median :21.00  
 Mean   :188.6   Mean   :19.75  
 3rd Qu.:201.2   3rd Qu.:27.25  
 Max.   :246.0   Max.   :38.00  
```

```
# There seems to be some relationship here
plot(df)
```



```
# Here is the correlation matrix, notice it is high!
cor(df)
```

```
           Weight   BodyFat
Weight  1.0000000 0.6966328
BodyFat 0.6966328 1.0000000
```

We have observed that there is a relatively strong linear relationship between these two variables. What next? We might ask, what is this relationship precisely?

In particular, note that we have observed a sample of vectors $(Y_1, X_1), \ldots, (Y_1, X_n)$. Now, we want to say something about the relationship between X and Y in general. One way to do that is to suppose at the **population** level that

$$\mathrm{E}\left[Y|X\right] = f(X).$$

That is, on average, Y is equal to $f(X)$. One way to do that is to assume that $Y|X = f(X) + \epsilon$, where ϵ is a random variable that satisfies $\mathrm{E}\left[\epsilon\right] = 0$. This assumption means that, for each Y_i, given X_i, we have that $Y_i = f(X_i) + \epsilon_i$. Note that we do not observe ϵ_i, but we can assume it exists. We can read this as Y_i is equal to $f(X_i)$, plus some random, individual error ϵ_i. The next step is to use the data to determine f.

Using the data analysis steps from the Introduction we can write out the first few steps:

- Question about a population: "How can we use weight to determine body fat percentage?' '
- Data: $(Y_1, X_1), \ldots, (Y_{20}, X_{20})$, (Y_i, X_i) are the body fat percentage and weight of individual $i \in [20]$.

We have explored the data with graphs and summary statistics. Now, we have posited the model $Y|X = f(X) + \epsilon$. Letting f be any function is too general. In fact, we can use the data to learn more about what f might be. Recall that earlier, we saw the scatter plot, where it looked like there was a linear relationship, (with some error), between Y and X. (We can draw a straight line through the middle of the data.)

Let's make some assumptions that make the statistical analysis easier:

1. Assume that $\forall i \in [20]$, it holds that

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

   This means that we assume that f is a line.

2. Next, we assume $\forall i \in [20]$, $\mathrm{E}\left[\epsilon_i\right] = 0$ and $\mathrm{Var}\left[\epsilon_i\right] = \sigma^2$. That is, the random error have the same mean and variance for each individual. In addition, the random errors average to 0.

3. We also assume that the individuals' Body fat percentage, weights and random errors are independent, that is, $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [20]$.

This is the **simple linear regression model**. That is, the simple linear regression model is the set of assumptions 1-3 given above.

It is often also assumed:

4. $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$,

but not always. Including the normality assumption is known as the **simple normal linear regression model.**

In general, a model is a set of assumptions about a population. The particular set of assumptions 1-3 is the simple linear regression model.

The following is some terminology used in regression analysis:

- Here, Y_i is the **response variable**, also known as the dependent variable, or the outcome variable.
- Here, X_i is the **covariate**, also known as the explanatory variable, or the independent variable.

Given a “question about a population” which involves regression, you should immediately identify the response variable and the covariates.

Now, how can we interpret this model? That is, what does it mean to assume this model?

First, observe that we assume that $\mathrm{E}[Y|X]$ is a line. This means there is a linear relationship between the average body fat percentage and weight.

Next, observe that for any individual, their actual body fat percentage is given by $Y = \mathrm{E}[Y|X] + \epsilon_i = \beta_0 + \beta_1 X_i + \epsilon_i$. Therefore, their body fat percentage will not fall exactly on the line $\beta_0 + \beta_1 X_i$. Rather, it will fall above or below the line, depending on ϵ_i. Furthermore, if we assume that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, then we know from the properties of the Normal distribution that this random error will not exceed 2σ with high probability. Therefore, most of the time, an individual's body fat percentage will fall within 2σ of the line.

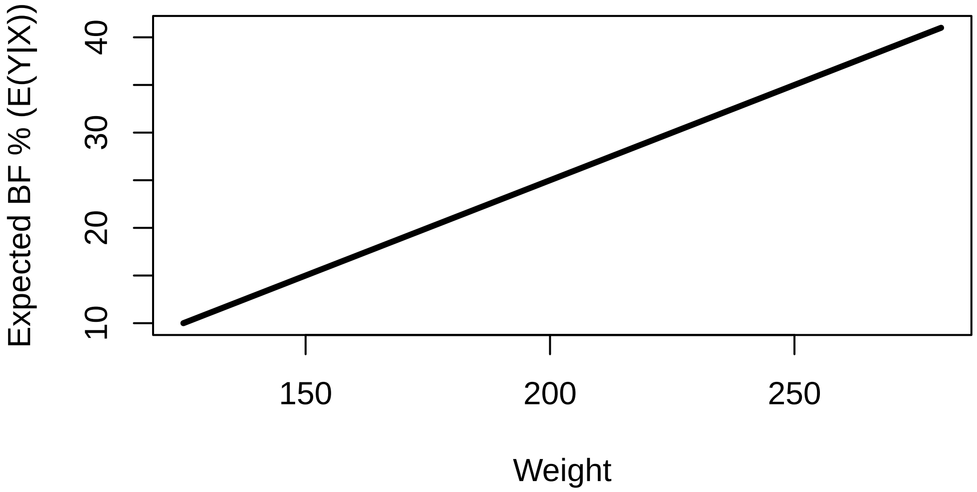
Third, notice that this quantity, 2σ, does not depend on X. That is, for any weight, we still expect an individual's body fat percentage to be within 2σ of the line, regardless of the value of weight.

Fourth, if we knew β_0, β_1, then given someone's weight, we could try to predict their body fat percentage given their weight. That is, we could calculate the expected body fat $\mathrm{E}[Y|X]$. There would still be their individual random error ϵ, so we would not be able to predict it exactly. However, if σ^2 isn't too big, then we could produce an accurate prediction.

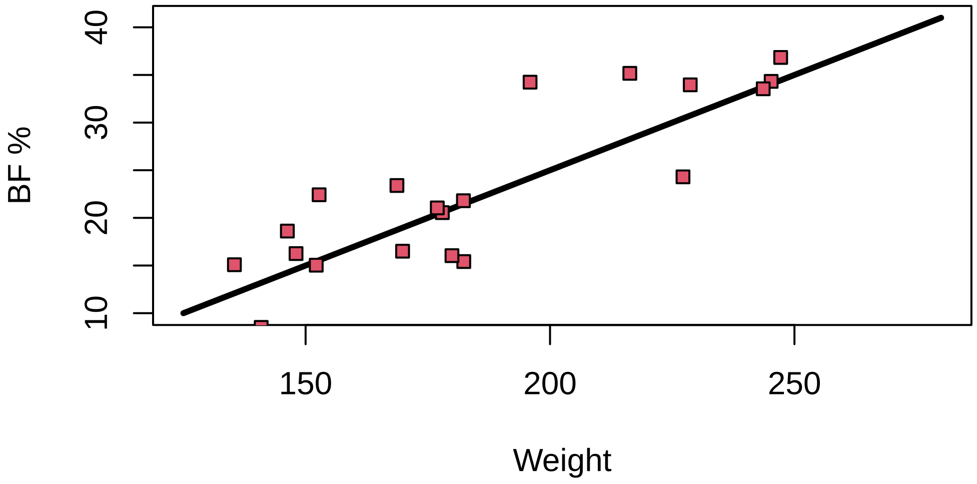
Therefore, if the model assumptions are correct, we assume there exists some line, around which the body fat percentages are scattered uniformly.

Next, we will simulate data from the normal simple linear regression model to gain a better understanding of this model. Suppose that $\beta_0 = -15$, $\beta_1 = .2$ and $\sigma = 5$. Then we would observe the following.

```
############################# Simulation
set.seed(3252)

# Suppose that beta_0=-15 and beta_1=0.2 and sigma=5,
# then we would have that the mean function E(Y|X) is given by the following line:
curve(-15+.2*x,125,280,lwd=3,xlab="Weight",ylab="Expected BF % (E(Y|X))")
```



```
# Next, let's simulate some body weights from the uniform distribution
Weight2=runif(20,135,250)

# Then, we can simulate the population body fat percentages according to the model as follows

# Simulating 20 values of the random error,
epsilons=rnorm(n=20,mean=0,sd=5)

# Computing the simulated Body fat percentages:
Bfs=-15+.2*Weight2+epsilons

# Plot the simulated values, and the mean function
curve(-15+.2*x,125,280,lwd=3,xlab="Weight",ylab="BF %")
points(Weight2,Bfs,pch=22,bg=2)
```



Notice how the data are scatted around the line uniformly? This is what data from a simple linear regression model looks like. Try changing the value in `set.seed()` and re-running the code. Notice how the data changes, but it is always scatted around the line uniformly? This is what we expect to see if the data follow a simple linear regression model.

Notice how the data simulated from our model appears similar to the body fat percentage and weights data we observed? That means this model (set of assumptions) is a good fit for our data.

Caution

In this model, and in regression in general, the response Y is not exactly equal to some function of X given by $f(X)$. The model assumes that **on average** $Y = f(X)$. Therefore, knowing someones "X" value will not exactly give us their Y value, but it would give us a good guess at it. The error ϵ is used to model the fact that someones "X" value will not exactly give us their Y value. Notice above how the actual points are scatted around the line, and not exactly equal to it! This is due to the errors ϵ.

3.1.2 The multiple linear regression model

But what about matrices? Why did we study matrices then? We can write the regression model in terms of matrices and vectors, to make it more compact.

Now, recall

$$Y_i|X_i = \beta_0 + \beta_1 X_i + \epsilon_i,$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. It is more convenient mathematically to let $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$,

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} = \left[1_n \mid (X_1, \dots, X_n)^\top\right],$$

$\beta = (\beta_0, \beta_1)^\top$ and $\ = (\epsilon_1, \dots, \epsilon_n)^\top$. Then we can write

$$\mathbf{Y}|\mathbf{X} = \mathbf{X}\beta + \ .$$

Often, we overload the notation Y, and use Y instead of $\mathbf{Y}$, and X instead of $\mathbf{X}$.

This form allows us to go beyond one explanatory variable very easily! Just add one column to X and one entry to β for each new variable. Observe the following model:

$$Y_i|(X_{i1}, \dots, X_{ik}) = \beta_0 + \beta_1 X_{i1} + \dots + \beta_k X_{ik} + \epsilon_i,$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$. This is known as the **multiple linear regression model** (MLR), or just the linear regression model for short. We can write this model in the same for as above: Let

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{1k} \\ \vdots & \vdots & \vdots \\ 1 & X_{n1} \dots & X_{nk} \end{bmatrix},$$

and $\beta = (\beta_0, \dots, \beta_k)^\top$. Then we can write the MLR as

$$\mathbf{Y}|\mathbf{X} = \mathbf{X}\beta + \ ,$$

where $\mathrm{E}\left[(\right]\epsilon) = 0$ and $\mathrm{Var}\left[\epsilon\right] = \sigma^2 I$. Notice how compact this is! As in the simple case, there is also the **normal MLR**, which further assumes that $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

We can then study the mathematical properties of

$$Y|X = X\beta + \epsilon$$

for general but fixed k, under the normal or vanilla MLR, which will cover many models.

3.1.3 Homework stop 1

Exercise 3.1. Try adjusting the parameters β_0, β_1, σ in the simulation, what happens to the data? What happens to the line?

Exercise 3.2. Is β an estimate or a population parameter? Why?

Exercise 3.3. Come up with another possible form of f that is not linear. Adjust the simulation to include this form of f.

Exercise 3.4. Write down the assumptions of the MLR and the normal MLR. What is the difference between the two models?

3.2 Least Squares

Now that we have settled on a model for the population, the next step is to use the data to estimate the model parameters. In particular, we need to estimate β. That will allow us to estimated $\mathrm{E}[Y|X]$ for any value of X.

Recall that we want to study the **population** model:

$$Y|X = X\beta + \epsilon.$$

3.2.1 Notation

For the model $Y|X = X\beta + \epsilon$, we have

- $Y \in \mathbb{R}^n$ is the response variable (a continuous random variable).
- $X \in \mathbb{R}^{n \times p}$ is the covariate matrix (Note that the first column is often 1_n).
- $X_i \in \mathbb{R}^p$ is the i^{th} observed explanatory variable ($i = 1, \dots, n$) (not a random variable, in the sense that we condition on it).
- $\beta \in \mathbb{R}^{p \times 1}$ is the coefficient vector .
- $\epsilon \in \mathbb{R}^n$ is the random error (continuous random variable) .

We may also refer to the actual observed values (versus the abstract mathematical concept of a random variable) as follows:

- $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ is the observed response variable (fixed/observed)
- x_{ij} is the i^{th} observation of the j^{th} explanatory variable (fixed/observed) Data:

Observation	Observed data point
1	$(y_1, x_{11}, x_{12}, \dots x_{1p})$
2	$(y_2, x_{21}, x_{22}, \dots x_{2p})$
$\vdots$	$\vdots$
n	$(y_n, x_{n1}, x_{n2}, \dots x_{np})$

We posit that

$$Y|X = X\beta + \epsilon,$$

where we assume that

- $\forall i \in [n]$, $\mathrm{E}[\epsilon_i] = 0$.
- $\forall i \in [n]$, $\mathrm{Var}[\epsilon_i] = \sigma^2$ (constant variance and is also known as homogeneity.)
- We also would assume that $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.
- $\beta \in \mathbb{R}^{p \times 1}$ is the unknown, population coefficient vector.
- $X \in \mathbb{R}^{n \times p}$ is a covariate matrix.

Let's talk about β. How do we interpret β? Suppose we know β. Then:

Note that

$$\mathrm{E}[Y_i|X_i] = \mathrm{E}[\beta^\top X_i + \epsilon] = \beta^\top X_i = \beta_1 X_{1,1} + \ldots + \beta_p X_{i,p}$$

What does each β_j mean? Suppose that X_j is a continuous covariate.

We can interpret (β_j) as follows:

Holding $X_{i,1}, \ldots, X_{i,j-1}, X_{i,j+1}, \ldots, X_{i,p}$ constant, a one unit increase in $X_{i,j}$ causes, on average, a β_j unit increase in Y_i.

From another angle, we have that $\partial \mathrm{E}[Y] / \partial X = \beta$, therefore, the rate of change with respect to the j^{th} covariate is β_j.

Caution

The "on average" and "holding other covariates constant" are very important components of the interpretation. First, the on average acknowledges the random error ϵ. In other words, a one unit increase in $X_{i,j}$ will not certainly increase Y_i, but it will on average. Next, the "holding other covariates constant" is used to mention how correlations between covariates are handled by the model. Some of the covariates in the model may be correlated, so increases in a given covariate may often be associated with changes in another covariate. This is not accounted for in the coefficient vectors β. That is why we must specify "holding other covariates constant".
For instance, if a model includes a terms for years of education attained and income, we know that as the number of years of education increase we expect to see a rise in income levels. As a result, to interpret the effect of coefficient on income, we must "hold years of education constant", comparing what is expected with income changes but education does not.

Caution

For now, we can assume that all of the covariates X_j are continuous variables. Later in the course, there may be categorical covariates. In this case, the β_j corresponding to the categorical covariates have a different interpretation. We will return to this later.

Recall Example 3.1. We assume $\forall i \in [20]$, it holds that

$$Y_i|X_i = \beta^\top X_i + \epsilon_i,$$

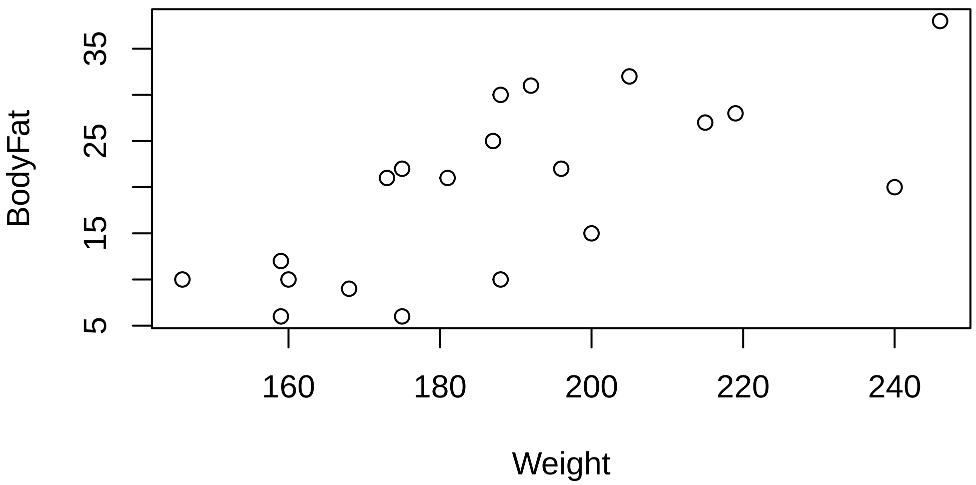
with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [20]$. A one unit increase in weight causes, on average, a β_2 unit increase in body fat percentage. Since β_1 is the intercept, it has a special interpretation. β_1 is the average value of Y_i given $X_i = 0$. It is also helpful to note that $\text{cov}(Y) = \sigma^2 I$.

3.2.2 Least squares estimation

Okay, but we don't know β! Just like we estimate the population mean with the sample mean, we need to estimate β. We would like an estimate $\hat{\beta}$, so that we can predict body fat percentage from weight. What is our best guess at β, given the data? One way to answer this, is through the method of **least squares**.

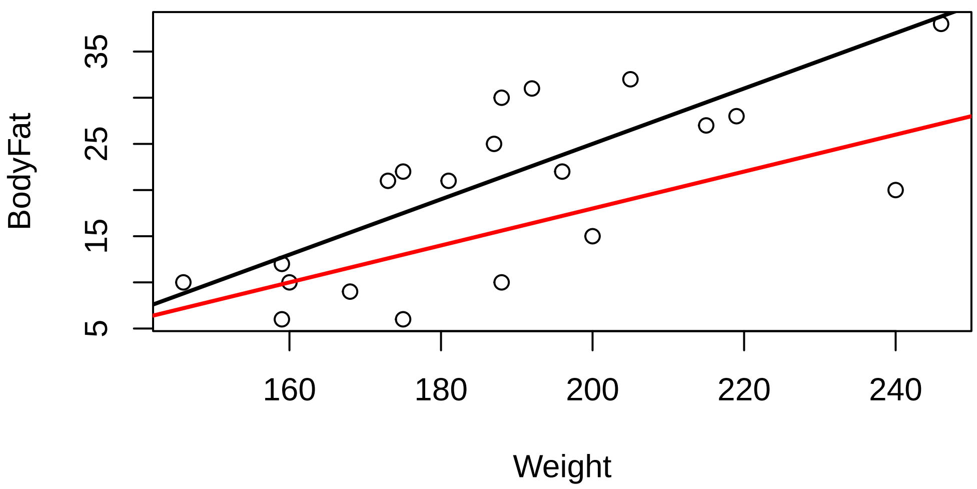
Returning to our example, recall that:

```
plot(df)
```



For example, suppose we want to determine if β is more likely to be $(-35, 0.3)^\top$ or $(-22, 0.2)^\top$. How can we say which line is a better to our data? One way is to graph them on top of the data and determine which one looks better. Let's plot these lines.

```r
plot(df)
# plot Y=-35+0.3X
abline(-35,0.3,lwd=2)
# plot Y=-25+0.2X
abline(-22,0.2,col='red',lwd=2)
```



Its not clear which one fits the data better. Even if it was clear, obviously, we cannot plot all possible lines. So how can we determine which line fits the data the “best”?

To do this, we have to define what “best” means quantitatively. For instance, one might ask which line minimizes the sum of the squared distances of the observed data points to the line? This line is then said to be the “best” line. Mathematically, given a proposed value of β, say $\beta_0 \in \mathbb{R}^p$, the signed distance to the hyperplane $X\beta_0$ is $\epsilon_0 = Y - X\beta_0$. The squared distances to the hyperplane $X\beta_0$ is then $\epsilon_0^\top \epsilon_0 = (Y - X\beta_0)^\top (Y - X\beta_0)$. We can then formulate this as a math problem: Which $\beta_0 \in \mathbb{R}^p$ minimizes $\epsilon_0^\top \epsilon_0$? i.e., $\hat{\beta} = \operatorname{argmin}_{\beta_0 \in \mathbb{R}^p} \epsilon_0^\top \epsilon_0$. It is more convenient to just write

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} (Y - X\beta)^\top (Y - X\beta).$$

In this framework, the “best” estimate is given by

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} (Y - X\beta)^\top (Y - X\beta).$$

Note best is in the sense of minimizing the average squared distance to the hyperplane/line. We could also define best in terms of some other metric, such as average absolute distance to the hyperplane/line. For now, we will stick with this metric.

The next step is to solve:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} (Y - X\beta)^\top (Y - X\beta).$$

How do we minimize a function???

RECALL in calculus, to find the minimum of a function we:

1. Obtain the first two derivatives of the function.
2. Set the first derivative to zero and solve for the critical value.
3. Use the second derivative to verify the critical value minimized the function.

Goal: Compute $\hat{\beta}$ – Minimize $g(\beta) = (Y - X\beta)^\top (Y - X\beta)$. (It may be useful to review taking derivatives with respect to vectors here.

Step 1a:

$$\begin{aligned}
\frac{\partial g}{\partial \beta} &= \frac{\partial g}{\partial \beta}(Y - X\beta)^\top (Y - X\beta) \\
&= \frac{\partial g}{\partial \beta}\left[Y^\top Y - 2(X\beta)^\top Y + (X\beta)^\top X\beta\right] && \text{(Transpose and distribute)} \\
&= -2\frac{\partial g}{\partial \beta}\beta^\top X^\top Y + \frac{\partial g}{\partial \beta}\beta^\top X^\top X\beta && ((AB)^\top = B^\top A^\top) \\
&= -2X^\top Y + 2X^\top X\beta && (\tfrac{\partial}{\partial x}x^\top A x = 2A \text{ if } A \text{ symmetric, } \tfrac{\partial}{\partial x}x^\top a = a) \\
&= -2X^\top (Y - X\beta).
\end{aligned}$$

Step 1b: (Do this for homework)

$$\frac{\partial^2 g}{\partial \beta \partial \beta^\top} = 2X^\top X.$$

Step 2: We now need $X^\top X$ to be invertible, so we will assume that X is full rank and $n \geq p$.

$$\begin{aligned}
-2X^\top (Y - X\beta) = 0 & \\
&\implies X^\top Y = X^\top X\beta \\
&\implies \beta = (X^\top X)^{-1} X^\top Y.
\end{aligned}$$

Step 3:

Recall that **if the Hessian matrix is positive definite at a critical point, then that critical point is a local minimum.** Since we have assumed X is full rank, this implies that $X^\top X$ is positive definite.

To summarize, the steps have proceeded as follows:

- Step 1a: $\frac{\partial g}{\partial \beta} = -2X^\top(Y - X\beta)$
- Step 1b: $\frac{\partial^2 g}{\partial \beta \partial \beta^\top} = 2X^\top X$ (Do this for homework)
- Step 2: $-2X^\top(Y - X\beta) = 0 \implies X^\top Y = X^\top X\beta \implies \beta = (X^\top X)^{-1}X^\top Y$
- Step 3: $2X^\top X$ is positive definite, and so

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

The estimate $\hat{\beta}$ is known as the **least squares estimate** of the regression coefficients.

Definition 3.1. The **least squares estimate** of the regression coefficients is

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

3.2.3 Example

Example 3.2. In the body weight example Example 3.1, write down X, Y and compute $\hat{\beta}$. Interpret $\hat{\beta}$.

First, we have that

$$Y = (6, 21, 15, 6, 22, 31, 32, 21, 25, 30, 10, 20, 22, 9, 38, 10, 27, 12, 10, 28)^\top$$

$$X = \left[1_{20} \mid (175, 181, 200, 159, 196, 192, 205, 173, 187, 188, 188, 240, 175, 168, 246, 160, 215, 159, 146, 219)^\top\right]$$

> Note
>
> For matrices A, B which have the same number of rows, $C = [A|B]$ is horizontal concatenation of A and B. This notation indicates that the matrix C is formed by placing A an B side by side, joining them horizontally. Therefore, X is the matrix whose first column is made up of ones, and second column is made up of the body weights.

Let's use R to compute $\hat{\beta}$.

```r
#Define X and Y
X=cbind(rep(1,nrow(df)), df$Weight)
Y=df$BodyFat

# cast to column vec
Y=matrix(Y,ncol=1)

#X'X
X_p_X=t(X)%*%X
```

```
#X'X inverse
X_p_X_inverse=solve(X_p_X)


#LS
beta_hat= X_p_X_inverse%*%t(X)%*%Y
beta_hat
```

```
            [,1]
[1,] -27.3762623
[2,]   0.2498741
```

```
# We can also use Rs lm() function to do this:
# This code is essential for the course.
# The first argument is the formula
model=lm(BodyFat ~ Weight, data=df)

#The summary function prints the model output.

summary(model)
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```
# The least squares estimates are given in the Estimate column of the summary.
```

The `lm()` function is used to fit multiple linear regression models in R. The basic usage involves specifying a formula and a data frame. The syntax is given by `lm(formula, data, ...)`.

The data argument should be the dataframe which contains your data. The formula argument is used to specify the model to be fitted. It provides a symbolic description of the model, indicating the response variable and the predictors/covariates, as well as the relationships between them. The left-hand side should be the name of your response variable, as it is named in your dataframe. To see the names of your variables use the `names()` function, e.g., `names(df)`. The right-hand side contains the covariates you want to include in your model. For instance, above, the formula is given by `BodyFat ~ Weight` . Note that `BodyFat` is the response and `Weight` is the covariate.

We now list some important properties of the least squares estimator.

Exercise 3.5. Compute $\mathrm{E}\left[\hat{\beta}\right]$ and $\mathrm{cov}(\hat{\beta})$.

Solution 3.1. It holds that $\mathrm{E}\left[\hat{\beta}\right] = \beta$ and $\mathrm{cov}(\hat{\beta}) = \sigma^2 I$.

Recall that an estimator is **unbiased** if its expectation equals the population parameter it is trying to estimate. After completing Exercise 3.5 you will see that $\hat{\beta}$ is unbiased for the parameter β.

The least squares estimator is also the "best linear unbiased estimator", or the BLUE. This is known as the Gauss–Markov theorem. This means that under the assumptions of the linear regression model, over any unbiased estimator of β we can construct, which is a linear combination of $Y_1, \dots, Y_n$, the estimator $\hat{\beta}$ has the smallest variance (and therefore, the smallest mean squared error. Recall that for an estimator $\hat{\alpha}$, the mean squared error is given by $\mathrm{E}\left[||\beta - \hat{\alpha}||^2\right]$.)

The Gauss–Markov theorem does not require the random error to be normally distributed. If we are willing to assume that $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, then $\hat{\beta}$ is also the maximum likelihood estimator and the "uniformly minimum-variance unbiased estimator", or UMVUE. This means that $\hat{\beta}$ has lower variance than any other unbiased estimator, no matter what the true value of β is.

One might ask, how can we use $\hat{\beta}$ to predict body fat percentage given weight? The estimate $\hat{\beta}$ gives us a best guess at the coefficients. Therefore, our best guess at someones body fat is given by

$$Best\ Guess = -27.3762623 + 0.2498741 \times Weight.$$

For instance, for someone who is 170 pounds, we would guess that their body fat percentage is $-27.3762623 + 0.2498741 \times 170 = 15.1023347$.

3.2.4 Homework stop 2

Exercise 3.6. Why do we need $\hat{\beta}$, why not use β?.

Exercise 3.7. Is $\hat{\beta}$ an estimate or a population parameter? What about β?

Exercise 3.8. Compute, X, Y and $\hat{\beta}$ in the following real data example:

It is challenging to assess a student's understanding of a subject without administering an exam. However, we can easily record the number of hours a student studies. A researcher would like to know if the number of hours studied and exam scores are related. This researcher collected the following data:

Student	Hours Studied	Exam Score (%)
1	5	55
2	8	65
3	12	78
4	6	58
5	10	72
6	9	68
7	15	85
8	7	60
9	11	74
10	13	80
11	14	82
12	20	90
13	5	55
14	6	59
15	18	88
16	7	62
17	16	86
18	4	50
19	3	45
20	19	89

To help you, here is some R code the dataset:

```r
# Data
study_data <- data.frame(
  Student = 1:20,
  Hours_Studied = c(5, 8, 12, 6, 10, 9, 15, 7, 11, 13, 14, 20, 5, 6, 18, 7, 16, 4, 3, 19),
  Exam_Score = c(55, 65, 78, 58, 72, 68, 85, 60, 74, 80, 82, 90, 55, 59, 88, 62, 86, 50, 45,
)
```

3.3 Least squares inference

Recall we **estimate** the parameter β using least squares:

Recall that $\hat{\beta} = (X^\top X)^{-1} X^\top Y$. We can predict a new weight $Y_{new}|X = x$ with $\hat{y}_{new} = x^\top \hat{\beta}$. We may be interested in the following questions: How good is $\hat{y}_{new}$ as a prediction, on average? How will new observations vary about the line? For example, given a specific weight, how will does body fat percentage vary around the regression line? How does $\hat{\beta}$ vary around β? Is there strong evidence that Y has a relationship with X? Is X adding information about Y at all?

To answer these questions, we need to look at the variation of our estimates and our data.

3.3.1 Important quantities: Residuals and fitted values

We now introduce some very important quantities: We call the estimated values given our observed X the fitted values: $\hat{Y} = X\hat{\beta}$. The fitted values are what our model would estimate the vector Y to be. We call $\hat{\epsilon} = Y - \hat{Y}$ is the **residual vector**. The ith entry of $\hat{\epsilon}$, say $\hat{\epsilon}_i$, is the ith **residual**. The residuals are the signed distances from the response variable to the estimated regression hyperplane. The **sum of squared error** or **sum of squared residuals** (SSE) is given by $\hat{\epsilon}^\top \hat{\epsilon} = \sum_{i=1}^n \hat{\epsilon}_i^2$. Note that since we estimated β using the least squares method, $\hat{\epsilon}^\top \hat{\epsilon}$ is minimized (with respect to varying β).

Example 3.3. Recall Example 3.1. What is the residual of individual 3? How can we interpret this value?

```
residuals=Y-X%*%beta_hat; residuals
```

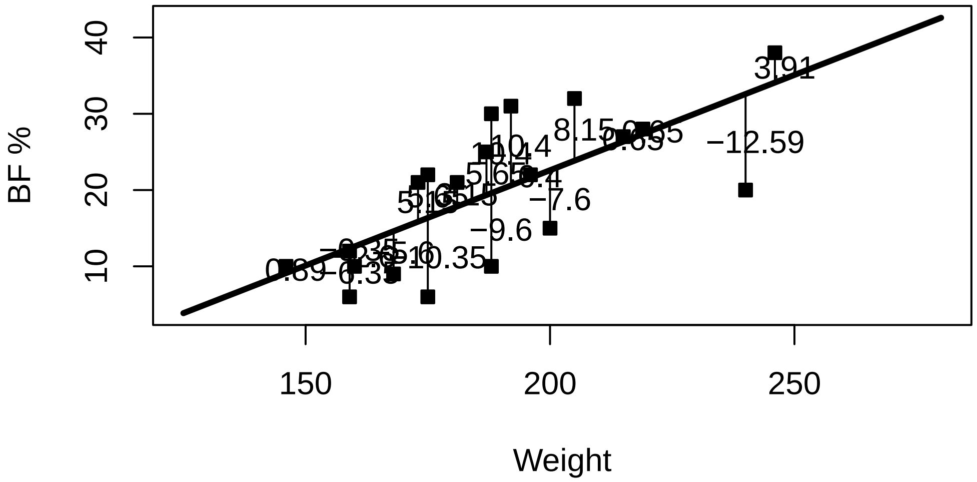
```
             [,1]
 [1,] -10.3517117
 [2,]   3.1490434
 [3,]  -7.5985652
 [4,]  -6.3537255
 [5,]   0.4009314
 [6,]  10.4004279
 [7,]   8.1520641
 [8,]   5.1480365
 [9,]   5.6497986
```

```
[10,]  10.3999245
[11,]  -9.6000755
[12,] -12.5935307
[13,]   5.6482883
[14,]  -5.6025928
[15,]   3.9072245
[16,]  -2.6035997
[17,]   0.6533228
[18,]  -0.3537255
[19,]   0.8946383
[20,]   0.6538262
```

```
# This means that individual 3's body fat is 7.5 percentage points lower than the fitted lin

residuals[3]
```

```
[1] -7.598565
```

```
# We can go further and and plot all of the residuals
curve(beta_hat[1]+beta_hat[2]*x,125,280,lwd=3,xlab="Weight",ylab="BF %")
points(Weight,BodyFat,pch=22,bg=1)
Yvals=cbind(BodyFat,model$fitted.values)
Xvals=cbind(Weight,Weight)
for(i in 1:nrow(Yvals)){
  lines(Xvals[i,],Yvals[i,])
  text(Xvals[i,1]+2,mean(Yvals[i,]),round(residuals[i],2))
}
```



```
# Then, the population body fat percentages, given weights will look like this:
#
# Bfs=-15+.2*Weight+rnorm(20,0,sd=5)
```

3.3.2 Variation decomposition

Variance decomposition is a fundamental concept that explains how the total variation in the response variable can be partitioned into different sources. This decomposition is crucial for evaluating the performance of the regression model and understanding the contributions of various factors.

The residuals describe one type of variation of the response values. We can also consider the total variation of the response. The total variation of the response, or the **sum of squares total/total sum of squares** (SST) is given by $SST = (n-1)\hat{\sigma}_y = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 = (Y - \bar{Y}1)^\top (Y - \bar{Y}1)$. It can be shown that the SST can be decomposed as follows:

$$SST = (Y-\bar{Y}1)^\top(Y-\bar{Y}1) = (Y-\hat{Y})^\top(Y-\hat{Y}) + (\hat{Y}-\bar{Y})^\top(\hat{Y}-\bar{Y}) = \hat{\epsilon}^\top\hat{\epsilon} + (\hat{Y}-\bar{Y})^\top(\hat{Y}-\bar{Y}).$$

That is, $SST = SSE + SSModel$ where

- $SSModel$, OR SSM measures the total variations of the response explained by the covariates X via the model based on $\hat{\beta}$.

- SSE measures the total variations of the response unexplained by the covariates X via the model based on $\hat{\beta}$.
- Note there are sometimes other names for SSE and $SSModel$, such as $SSRegression$, $SSwithin$ and $SSbetween$, etc.

So, we have that the total variation in the response can be broken down into that which is explained by the X values, and that which is unexplained.

An interesting observation is given as follows: The first column of the X matrix is given by 1_n, which implies that

$$\bar{Y}1 = X \begin{bmatrix} \bar{Y} \\ 0 \end{bmatrix}.$$

This means that if we let $\hat{\beta}_* = (\bar{Y}, 0, \dots, 0)^\top$, then $(Y - \bar{Y}1)$ would be the signed distances to (or the residuals of) the regression hyperplane corresponding to $\hat{\beta}_*$. Since $\hat{\beta}$ minimizes the sum of squared residuals, we must have that the hyperplane corresponding to $\hat{\beta}$ has a smaller sum of squared residuals than the regression hyperplane corresponding to $\hat{\beta}_*$. Therefore, we must have that $\hat{\epsilon}^\top \hat{\epsilon} \leq (Y - \bar{Y}1)^\top (Y - \bar{Y}1)$.

Each of these terms in the decomposition is associated with a certain number of **degrees of freedom**.

- Total: $dfT = n - 1$.
- Model: $dfM = \#$ **non-zero** $\beta - 1$.
- Error: $dfE = n - \#$ **non-zero** β.

Intuitively, since the SSE is the variance unexplained by the model/covariates, the SSE is related to the error variance σ^2. In fact, to estimate σ^2, we use

$$\hat{\sigma}^2 = MSE = \frac{SSE}{dfE}.$$

The null model is defined as $Y|X = \beta_0 + \epsilon$. This is the model where the last $p - 1$ terms in the true vector β are 0. This model says that Y does not depend on X. In the null model, we only need to estimate the mean, so $df = n - 1$. Therefore, under the null model,

$$\begin{aligned} \hat{\sigma}^2 = (n-1)^{-1} SST = \hat{\sigma}_Y^2 \\ &= (n-1)^{-1} \sum_{i=1}^{n} (Y_i - \bar{Y})^2 = (n-1)^{-1} (Y - \bar{Y}1)^\top (Y - \bar{Y}1). \end{aligned}$$

Therefore, in the null model, the estimate of σ^2 via the MSE is just the usual estimate of the variance of the response. This is intuitive!

The following table can be used to summarize the variation in the response:

Source	SS	df	MS
Model	SSM	dfM	$MSModel = SSM/dfM$
Residual	SSE	dfE	$MSE = SSE/dfE$
Total	SST	dfT	

> ℹ Note
>
> It is very important to be able to interpret these terms! The derivation is also important. However, we can use a machine to compute anything for us, so memorizing the formula is not helpful.

3.3.3 Coefficients of determination

A model is a good model if it can explain a fair amount of the variation in the response. (You can think that the model explains "changes" in the response.) In other words, $SSModel$ should be as close to $SSTotal$ as possible; or equivalently, $SSError$ should be as close to 0 as possible. Now, "close" is a relative term, and so we need another value to reference to. This is where the R^2 comes in:

$$R^2 = \frac{SSModel}{SST},$$

and is the proportion of variation explained by the model. It is clear that $0 \leq R^2 \leq 1$, and so rescaling the data will not affect R^2 (like it would affect the sum of squares terms SST, SSE, SSM). If R^2 is close to 1, it is large – "close to 1" is a subjective/area dependent. Generally, the larger the R^2, the better the model!

To compare different models, we could potentially add different covariates and see if R^2 improves. However, every time you add any variable, R^2 will always increase. Therefore, it is common to use the adjusted coefficient of determination:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p}.$$

Thus, the (adjusted) coefficient of determination can be used as a measure of how well the regression model fits the data (how much variance is explained). It could also be used to compare models.

3.3.4 The F test

The coefficients of determination are summary statistics which give an idea of the fit of the model. We would also like a significance test that tells us whether the covariates explain Y, or what we observed was simply due to sampling variation.

If $\beta = (\beta_1, \ldots, \beta_p)^\top$ then let $\tilde{\beta} = (\beta_2, \ldots, \beta_p)^\top$. That is $\tilde{\beta}$ is the regression coefficients without the intercept term. Similarly, let $\tilde{\hat{\beta}} = (\hat{\beta}_2, \ldots, \hat{\beta}_p)^\top$. Now, we want to avoid the situation where $\tilde{\beta} = 0$ but $\tilde{\hat{\beta}} \neq 0$ due to sampling variation.

To do this, we perform a significance test:

$$H_0 : \tilde{\beta} = 0 \qquad vs \qquad H_1 : \tilde{\beta} \neq 0.$$

First, we need the normality assumption to perform significance test: Assume $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. With this assumption, the model is then known as the **Normal Multiple Linear Regression Model**. It is important to note that the least squares method does not require this assumption, and this assumption is required only for the significance test to be valid. To test the hypothesis stated above, we use the overall F test and the observed test statistic is $F_{obs} = MSModel/MSE$. Why?

With the extra normality assumption, we have the following holds:

- $Y|X$ is normally distributed.
- We have that $SSM/\sigma^2 \sim \chi^2_{dfM}$ and $SSE/\sigma^2 \sim \chi^2_{dfE}$.
- Furthermore, $SSM \perp SSE$.

Recall that the ratio of two independent χ^2 distributions divided by their respective degrees of freedom follows an F distribution. Therefore, we have that $F_{obs} \sim F_{dfM,dfE}$. The corresponding p-value is $\Pr(W > F_{obs})$ where $W \sim F_{dfM;dfE}$. We can alternatively reject the null hypothesis if $F_{obs} > F_{dfM;dfE,1-alpha}$, where $F_{obs} > F_{dfM;dfE,1-alpha}$ is the $1-\alpha$ quantile of the $F_{dfM,dfE}$ distribution.

We can now present the complete ANOVA table

Source	**SS**	**df**	**MS**	**F**	**p-value**
Model	SSM	dfM	$MSModel = \frac{SSR}{dfM}$	$F = \frac{MSModel}{MSE}$	$\Pr(W > F_{obs})$
Residual	SSE	dfE	$MSE = \frac{SSE}{dfE}$		
Total	SST	dfT			

Example 3.4. In Example 3.1, compute and interpret the coefficients of determination. Compute and interpret the ANOVA table. Test whether the regression model is significant. (This means perform the F test.)

```
# recall
head(df)
```

```
  Weight BodyFat
1    175       6
2    181      21
3    200      15
4    159       6
5    196      22
6    192      31
```

```r
# The F test results are given in the summary
summary(model)
```

```
Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,    Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```r
# The ANOVA table is given below

# First define the null model object using lm()
# This line fits a model with only the intercept term
null_model=lm(BodyFat~1,data=df)

# This line gets the ANOVA table
anova(null_model,model)
```

```
Analysis of Variance Table
```

```
Model 1: BodyFat ~ 1
Model 2: BodyFat ~ Weight
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)    
1     19 1737.75                                  
2     18  894.42  1    843.33 16.972 0.0006434 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# We can also do this by hand:


# Store sample size
n=nrow(df)
p=2

# Compute sum of squares
SST=t(Y-mean(Y)*rep(1,n))%*%(Y-mean(Y)*rep(1,n))
Yhat=X%*%beta_hat
res=Y-Yhat
SSE=t(res)%*%res
SSM=SST-SSE

dfe=n-p
dfm=p-1
MSM=SSM/dfm


MSE=SSE/dfe

Fv=MSM/MSE

p.val=1-pf(Fv,dfm,dfe)

# ANOVA Table:
ANOVA_Table=rbind(c(SSM,dfe,MSM,Fv,p.val),c(SSE,dfe,MSE,NA,NA),c(SST,n-1,NA,NA,NA))
rownames(ANOVA_Table)=c("Model","Error","Total")
colnames(ANOVA_Table)=c("SS","df","MS","F","p-value")
ANOVA_Table
```

```
             SS df        MS        F      p-value
Model  843.3252 18 843.32521 16.97164 0.0006434484
Error  894.4248 18  49.69027       NA           NA
```

```
Total 1737.7500 19        NA       NA           NA
```

3.3.5 Homework stop 3

Exercise 3.9. In the following real data example: **Compute and interpret** the coefficient of determination, the adjusted coefficient of determination and perform the F test for model significance. Including printing the ANOVA table, the null and alternative hypothesis, an interpretation of the p-value and the conclusion of the test.

It is challenging to assess a student's understanding of a subject without administering an exam. However, we can easily record the number of hours a student studies. A researcher would like to know if the number of hours studied and exam scores are related. This researcher collected the following data:

Student	Hours Studied	Exam Score (%)
1	5	55
2	8	65
3	12	78
4	6	58
5	10	72
6	9	68
7	15	85
8	7	60
9	11	74
10	13	80
11	14	82
12	20	90
13	5	55
14	6	59
15	18	88
16	7	62
17	16	86
18	4	50
19	3	45
20	19	89

To help you, here is some R code the dataset:

```
# Data
study_data <- data.frame(
  Student = 1:20,
  Hours_Studied = c(5, 8, 12, 6, 10, 9, 15, 7, 11, 13, 14, 20, 5, 6, 18, 7, 16, 4, 3, 19),
  Exam_Score = c(55, 65, 78, 58, 72, 68, 85, 60, 74, 80, 82, 90, 55, 59, 88, 62, 86, 50, 45,
)
```

Exercise 3.10. Write down the interpretations of: SSE, MSE, R^2, $\bar{R}^2$, SSM.

Exercise 3.11. What is the interpretation of the p-value in the ANOVA table?

Exercise 3.12. What extra assumption is needed to perform the F-test?

3.3.6 Significance of one variable

So far, we have learned that the least squares method yields the following estimate of $\hat{\beta} = (X^\top X)^{-1}X^\top Y$ with $\mathrm{E}\left[\hat{\beta}\right] = \beta$ and $\mathrm{cov}(\hat{\beta}) = (X^\top X)^{-1}\sigma^2$. Moreover, we use MSE to estimate σ^2. Next, we learned that we can summarize the SS, df, and MS in an ANOVA table. We used the F test and the coefficient of determination to evaluate the quality of the model, i.e., to see the amount of information X provides about Y.

When the model is a significant model, then, at least one of the individual explanatory variables is useful in explaining the response. We may be interested in whether a specific covariate, or set of covariates is useful in explaining the response variable. We now learn how we can test for the significance of each individual explanatory variable separately and how we can test for the significance of a subset of explanatory variables. Note that these tests also require that the random error is normally distributed.

To test for significance and compute confidence intervals of a single variate, we have to compute the distribution of $\hat{\beta}_j$. We first compute the mean and variance of $\hat{\beta}_j$. First, given that $\mathrm{E}(\hat{\beta}) = \beta$, we have $\mathrm{E}(\hat{\beta}_j) = \beta_j$. Next, $\mathrm{Var}\left[\hat{\beta}_j\right]$ is the $(j,j)^{th}$ entry of $\mathrm{cov}(\hat{\beta})$. In addition, we have derived that $\mathrm{cov}(\hat{\beta}) = (X^\top X)^{-1}\sigma^2$.

Now, recall that if Z is multivariate normal, i.e., $Z \sim \mathcal{N}(\mu, \Sigma)$, then $b+AZ \sim \mathcal{N}(b+A\mu, A\Sigma A^\top)$, i.e., $b+AZ$ is also multivariate normal. Therefore, since we have assumed that $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I)$ and that $Y|X = X\beta + \epsilon$, it follows that $Y|X \sim \mathcal{N}_n(X\beta, \sigma^2 I)$. Next, we may recall that $\hat{\beta} = (X^\top X)^{-1}XY$. Let $A = (X^\top X)^{-1}X$. Then $\hat{\beta} = AY$. It follows that $\hat{\beta}$ is also multivariate normal! Putting everything together, we have that $\hat{\beta} \sim \mathcal{N}_p(\beta, (X^\top X)^{-1}\sigma^2)$.

Theorem 3.1. *Under the assumptions of the* ***normal linear regression model*** *it holds that* $\hat{\beta} \sim \mathcal{N}_p(\beta, (X^\top X)^{-1}\sigma^2)$.

Now that we have the distribution of $\hat{\beta}$, we can use it to compute the confidence intervals for β_js.

Recall from introductory statistics (MATH 1131) that you learned that if we want to compute a confidence interval for the sample mean and the sample variance was unknown, we had to estimate the variance. Similarly, here, the variance of $\hat{\beta}_j$ contains σ, an unknown parameter. Recall that, we estimate σ^2 by MSE, and so we can estimate the variance of $\hat{\beta}_j$ by $\hat{\text{Var}}\left[\hat{\beta}_j\right] = (X^\top X)^{-1}_{j,j} MSE$.

It can be shown that $\hat{\beta} \perp MSE$. Therefore, we have that

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{\widehat{var}(\hat{\beta}_j)}} \sim t_{dfE}.$$

Now that we know the distribution of $\hat{\beta}_j$, we can perform significance testing and compute confidence intervals.

If we want to test

$$H_0 \colon \beta_j = \beta_j^0 \qquad vs \qquad \beta_j \neq \beta_j^0$$

we can do the following.

The observed test statistic is $TS = \frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}}$. Note that, under the null hypothesis, we have that $\frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}} \sim t_{dfE}$. Thus, the corresponding p-value is obtained based on the t_{dfE} distribution. Specifically, we can compute the p-value $\Pr\left(-|TS| < Z\right) + \Pr\left(|TS| > Z\right) = 2 * \Pr\left(|TS| > Z\right)$, where $Z \sim t_{dfE}$.

The test proceeds as follows:

1. State the hypotheses

$$H_0 \colon \beta_j = \beta_j^0 \qquad vs \qquad H_1 : \ \beta_j \neq \beta_j^0.$$

2. Compute the test statistic $\frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}}$ and the p-value.
3. Interpret the p-value, and use it to decide whether you reject the null hypothesis.

Often, one may choose a threshold α, and reject the null hypothesis if the p-value falls below that threshold. Other times, we use the p-value as a description of evidence against the null. If it is larger than 0.05, but still small, then that still constitutes some evidence against the null hypothesis.

Let's now discuss one-sided hypotheses. First, consider:

$$H_0 \colon \beta_j \leq \beta_j^0 \qquad vs \qquad H_1 : \ \beta_j > \beta_j^0$$

Then, if the alternative hypothesis is true, we expect TS to be positive. The p-value is given by $\Pr(TS > Z)$, where $Z \sim t_{dfE}$. Notice that the p-value is measuring how extremely positive TS is. Using the threshold method, we can also check if $TS > t_{dfE,1-\alpha}$. Next, if we want to test

$$H_0 \colon \beta_j \geq \beta_j^0 \qquad vs \qquad H_1 : \ \beta_j < \beta_j^0,$$

then if the alternative hypothesis is true, we expect TS to be negative. The p-value is given by $\Pr(TS < Z)$, where $Z \sim t_{dfE}$. Notice that the p-value is measuring how extremely negative TS is. Using the threshold method, we can also check if $TS < t_{dfE,\alpha}$.

> Note
>
> We use $t_{k,p}$ to denote the pth quantile of the t distribution with k degrees of freedom. For $p = 0.025$ and large k, this is approximately equal to 2.

In Example 3.1, test if the coefficient for weight is not equal to 1. Next, test if the coefficient for weight is greater than 1. Lastly, test if the coefficient for weight is not equal to 0.

First, we have that

$$H_0 \colon \beta_1 = 15 \qquad vs \qquad H_1 : \ \beta_1 \neq 15.$$

Now, let's execute the test:

```
#changing matrix to scalar
MSE=c(MSE)
hvar_beta=solve(t(X)%*%X)*MSE


TS=beta_hat[2]/sqrt(hvar_beta[2,2])

# not equal
# pt(x,df) is the CDF of a t distributed RV with df degrees of freedom at x.
p_val=2*(1-pt(abs(TS),dfe))
p_val
```

```
[1] 0.0006434484
```

```
# We can also use the model object to test if it is not equal to 0:
# The test statistic and the pvalue are given in the t value and Pr(>|t|)  columns, respecti
summary(model)
```

```
Call:
```

```
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,    Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

Based on the concepts that you have learned in 1131, and what we have reviewed in previous lectures, it also follows from the above analysis that a $(1-\alpha)100$ confidence interval for β_j is

$$\hat{\beta}_j \pm t_{dfE,\alpha/2}\sqrt{\widehat{var}(\hat{\beta}_j)}.$$

:::{#exm-3-4-2} In Example 3.1, compute a 99% and a 95% confidence interval for the coefficient for weight. Which one is longer? Why? Interpret these intervals. :::

```
# By hand
beta_hat[2]+c(-1,1)*qt(0.975,dfe)*sqrt(hvar_beta[2,2])
```

```
[1] 0.1224448 0.3773035
```

```
beta_hat[2]+c(-1,1)*qt(0.995,dfe)*sqrt(hvar_beta[2,2])
```

```
[1] 0.07528522 0.42446306
```

```
# Auto software/using lm:
confint(model,level=0.95)
```

```
                  2.5 %     97.5 %
(Intercept) -51.6365090 -3.1160157
Weight        0.1224448  0.3773035
```

```r
confint(model,level=0.99)
```

```
                   0.5 %    99.5 %
(Intercept) -60.61484736 5.8623227
Weight        0.07528522 0.4244631
```

If we took many samples of size 20 and computed a 95% (99%) confidence interval for each sample, then 95% (99%) of then would contain the true coefficient for the weight variable. We can conclude that with 95% (99%) confidence, the true coefficient for weight likely falls within (0.12, 0.38) ((0.08,0.42)).

> Caution
>
> The key to understanding a confidence interval is to realize that the end points of the interval depend on the sample, and are therefore, random. On the other hand, the population parameter is not random, it is fixed. Therefore, if we drew a different sample, the interval would move, and there is a $(1-100\alpha)\%$ chance that that interval catches the population parameter. Most of the time it will contain the parameter, but not always.

Recall that the point of computing a confidence interval is to report the uncertainty in our estimate that resulted from drawing a sample. We expect the true parameter to be somewhere in that range, and our best guess at the parameter is given by the center of the interval.

3.3.7 Inference for the mean response and prediction intervals

We may wish to estimate the average response at a specific set of the covariates x. Given x, the theoretical mean response is $x^\top\beta$. Given x, we can estimate the mean response as $x^\top\hat{\beta}$. For instance, what is the average body fat percentage at 160 pounds? How accurate is our estimate? We can use a confidence interval to answer this question.

Note that the expectation and variance of the estimate of the mean response are given by $\mathrm{E}\left[x^\top\hat{\beta}\right] = x^\top\beta$ and $\mathrm{Var}\left[x^\top\hat{\beta}\right] = x^\top(X^\top X)^{-1}x\sigma^2$. Again, we must estimate σ and we can write $\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right] = x^\top(X^\top X)^{-1}xMSE$.

Exercise 3.13. Under the assumptions of the normal linear regression model, show that for a fixed covariate vector $x \in \mathbb{R}^p$, $x^\top\hat{\beta}$ has a multivariate normal distribution and find it's mean and variance. Argue that $\frac{x^\top\hat{\beta}-x^\top\beta}{\sqrt{\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right]}} \sim t_{dfE}$.

It can be shown that a $(1-\alpha)100\%$ confidence interval for the mean response $\mathrm{E}[Y|X=x]$ is

$$x^\top \hat{\beta} \pm t_{dfE,\alpha/2}\sqrt{\hat{\mathrm{Var}}\left[x^\top \hat{\beta}\right]}.$$

Similarly, if we want to test

$$H_0\colon \mathrm{E}[Y|X=x]=\mu_0 \qquad vs \qquad \mathrm{E}[Y|X=x]\neq \mu_0$$

we can do the following:

The observed test statistic is $TS(x,\mu_0)=\frac{x^\top\hat{\beta}-\mu_0}{\sqrt{\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right]}}$. Observe that under the null hypothesis, we have that $TS(x,\mu_0)\sim t_{dfE}$. Therefore, the p-value is given by $2*\Pr(|TS(x,\mu_0)|>Z)$.

Similar to the previous section, we can also perform one-sided tests:

- Right-sided test ($H_1\colon x^\top\beta>\mu_0$): p-value $\Pr(TS(x,\mu_0)>Z)$.
- Left-sided test ($H_1\colon x^\top\beta<\mu_0$): p-value $\Pr(TS(x,\mu_0)<Z)$.

We may also wish to predict what the response will be, given a new set of covariates. On top of that, we may again wish to quantify how much error there is in our prediction. For instance, what is the predicted body fat percentage of someone who is 160 pounds? Note that this differs from the previous section. In the previous section, we were interested in the average body fat percentage of someone who is 160 pounds. Here, we are interested in predicting the body fat percentage of a single, specific person, and not the average of the whole population.

Specifically, suppose that we have a subject whose covariates are given by z, but we do not know the value of the subjects response, which we can denote by Y_{new}. Then the true response is $(Y_{new}|Z=z) \ = z^\top\beta+\epsilon_{new}$.

Suppose we want to predict Y_{new} and give an idea of how much error is in our prediction. The predicted response is known, and is given by $\mathrm{E}[Y_{new}|Z=z]=z^\top\hat{\beta}$. We have $\mathrm{Var}[Y_{new}|Z=z]=\mathrm{Var}\left[z\hat{\beta}\right]+\mathrm{Var}[\epsilon_{new}]=z^\top(X^\top X)^{-1}z\sigma^2+\sigma^2$. Therefore, the variation in a new response is the variation in our estimate of β plus the inherent population variation, σ^2. We have that this can be estimated with: $\hat{\mathrm{Var}}[Y_{new}|Z=z]=z^\top(X^\top X)^{-1}zMSE+MSE$.

Exercise 3.14. Under the assumptions of the normal linear regression model, show that for a fixed covariate vector $z\in\mathbb{R}^p$, $Y_{new}|Z=z$ has a multivariate normal distribution and find it's mean and variance. Argue that given $Z=z$,

$$\frac{Y_{new}-z^\top\beta}{\sqrt{\hat{\mathrm{Var}}[Y_{new}]}}\sim t_{dfE}.$$

Therefore, the $(1-\alpha)100\%$ prediction interval for Y_{new} is given by:

$$z\hat{\beta} \pm t_{dfE,\alpha/2}\sqrt{z^\top(X^\top X)^{-1}zMSE + MSE}.$$

Note that the prediction interval is wider than that of the mean response interval for the same covariate vector z. That is because it is more difficult to predict the response for a specific person than it is to estimate a mean of a population. Furthermore, the interpretation of a prediction interval is different. A $(1-\alpha)100\%$ prediction interval can be interpreted it as follows. Given a $(1-\alpha)100\%$ prediction interval for $Y_{new}|Z = z$, say (a, b), we say that the probability Y_{new} is in (a, b) is $(1-\alpha)100\%$. Note that this differs substantially from a confidence interval!

Example 3.5. In Example 3.1, execute the following: What is a 95% confidence interval for the mean of someone who weighs 165 pounds? What is a 95% confidence interval for predicted BF% of someone who weighs 165 pounds? Interpret these intervals.

```r
# Intervals are given as follows:

z <- data.frame(Weight=165)
predict(model, newdata = z, interval = 'confidence')
```

```
       fit      lwr      upr
1 13.85297 9.379675 18.32627
```

```r
predict(model, newdata = z, interval = 'prediction')
```

```
       fit       lwr      upr
1 13.85297 -1.617547 29.32349
```

We are 95% confident the mean body fat of a person who weighs 165 pounds is in 13.8529704, 9.3796749, 18.3262658. There is a 95% probability that the body fat of a person who weights 165 pounds is in 13.8529704, -1.6175473, 29.323488 . Note that the prediction interval is wider!

3.3.8 Homework stop 4

Exercise 3.15. What is the difference between a prediction interval and an interval for the mean response ?

Exercise 3.16. Code the confidence intervals for the mean response and prediction interval without using the predict function.

Exercise 3.17. Do the chapter 3 practice problems from the problem list.

3.3.9 Partial testing

We may be interested in executing the following hypothesis test:

$$H_0\colon (\beta_1, \dots, \beta_k) = 0 \qquad vs \qquad (\beta_1, \dots, \beta_k) \neq 0.$$

This amounts to testing whether the subset of variables $(\beta_1, \dots, \beta_k)$ adds anything to the model beyond $(\beta_{k+1}, \dots, \beta_p)$. For example, you may be interested in whether location related covariates affect the price of Airbnb. The overall idea is to compare the reduced (null) model with $p-k$ covariates to the complete (saturated, full) model (which contains all covariates).

Let's first review the F-test. We learned about the F test, which compares the following models:

$$Y|X = \beta^\top X + \epsilon \qquad vs \qquad Y|X = \beta_1 + \epsilon.$$

Here, the complete model is given by $Y|X = \beta^\top X + \epsilon$ and the reduced model is given by $Y|X = \beta_1 + \epsilon$. Recall that the test statistic is given by

$$\frac{SSM/dfM}{SSE/dfE} = \frac{(SST - SSE)/(dfT - dfE)}{SSE/dfE},$$

where the degrees of freedom are in terms of the full model (not the null model). We could then rewrite this test statistic as

$$\frac{SSM_C/dfM_C}{SSE_C/dfE_C} = \frac{(SST_C - SSE_C)/(dfT_C - dfE_C)}{SSE_C/dfE_C},$$

where C stands for the complete model. (All that has changed is the notation, we added a C subscript.)

Now, note that $SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$ has nothing to do with what covariates are in the model. In other words, SST is always the same, not matter what covariates are in the model. Therefore, $SST_C = SST_R = SST$, where SST_R stands for the "sum of squares total" in the reduced model. In our example of the F test, the least squares estimate of β_1 in the reduced model is $\hat{\beta}_1 = \bar{Y}$ and the associated residual vector is given by $\hat{\epsilon} = Y - \bar{Y}1_n$. But wait, observe that in this case, we have that $\hat{\epsilon}^\top \hat{\epsilon} = SST$! Therefore, putting everything together, in this example, we have that $SSM_C = SST_C - SSE_C = SSE_R - SSE_C$. That is, the model sum of squares for the complete model is the difference between the sum-squared error in the reduced model and the sum-squared error in the complete model. We can then rewrite the test statistic as

$$\frac{(SSE_R - SSE_C)/(dfT_C - dfE_C)}{SSE_C/dfE_C}.$$

The difference $SSE_R - SSE_C$ can be interpreted as the extra information gained from adding the covariates into the model OR total explained variations lost by going from the full model to the reduced model.

This idea can be generalized to develop a general method for testing hypotheses of the type:

$$H_0 \colon (\beta_2, \dots, \beta_k) = 0 \qquad vs \qquad (\beta_2, \dots, \beta_k) \neq 0.$$

We complete the test as follows. Given a full model (which contains $\beta_1, \dots, \beta_p$) and reduced model (which contains $\{\beta_1, \beta_{k+1}, \dots, \beta_p\}$), define:

- $SSE_R - SSE_C = SSdrop$
- $dfE_R - dfE_C = dfdrop$
- $MSdrop = SSdrop/dfdrop$

Then the test statistic and p-value are given by: $TS = MSdrop/MSE_C$ and $\Pr\left(F_{dfdrop,dfE_C} \geq TS\right)$, respectively.

We can interpret $SSE_R - SSE_C$ as the extra info gained from adding the extra covariates into the model OR total explained variations lost by going from the full model to the reduced model. In addition, $dfE_R - dfE_C = k - 1$, or the number of covariates dropped from the full model to obtain the reduced model.

ℹ Note

If you take $k = 1$, then this is equivalent to the t-test!

3.3.10 Partial coefficient of determination

We can define the **partial coefficient of determination** as follows:

$$\begin{aligned} R^2(X_1, \dots, X_{k-1} | X_k, \dots, X_p) &= (SSE_R - SSE_C)/SSE_R \\ &= SSdrop/SSE_R. \end{aligned}$$

You might also see the partial correlation coefficient:

$$R(X_1, \dots, X_{k-1} | X_k, \dots, X_p) = \sqrt{R^2(X_1, \dots, X_{k-1} | X_k, \dots, X_p)}.$$

This quantity is the extra proportion of variation explained from adding the covariates $X_1, \dots, X_{k-1}$ to the model which already contains $X_k, \dots, X_p$.

Example 3.6. A researcher ran an experiment to see if YouTube, Facebook and newspaper ads would improve sales. Run the partial F test to see how online advertising affects sales. Compute and interpret the following quantities:

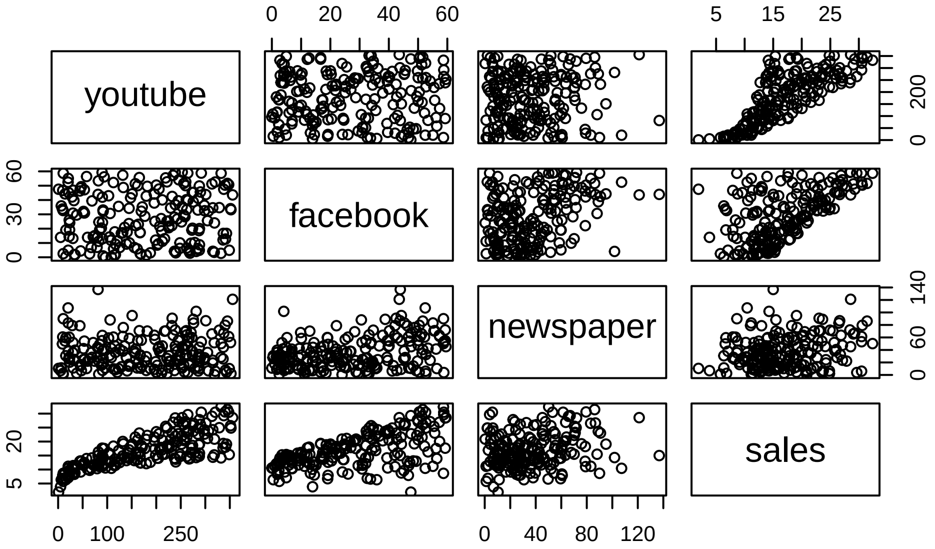
- $SSE_R - SSE_C = SSdrop$

- $dfE_R - dfE_C = dfdrop$
- $MSdrop = SSdrop/dfdrop$
- Test stat: $TS = MSdrop/MSE_C$
- p-value: $\Pr(F_{dfdrop,dfE_C} \geq TS)$
- Partial coefficient of determination

```
# install.packages('datarium')
data("marketing", package = "datarium")
#printing out first few rows
head(marketing, 4)
```

```
  youtube facebook newspaper sales
1  276.12    45.36     83.04 26.52
2   53.40    47.16     54.12 12.48
3   20.64    55.08     83.16 11.16
4  181.80    49.56     70.20 22.20
```

```
plot(marketing)
```



```
#setting n to be a variable (sample size)
n=nrow(marketing)
```

```r
# Estimation: How to get an estimate $\hat\beta$ of $\beta$?
# lm( sales~   , data= marketing)
full_model<- lm(sales ~ youtube+facebook+newspaper, data = marketing)
summary(full_model)
```

```

Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5932  -1.0690   0.2902   1.4272   3.3951 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,	Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
summ=summary(full_model)

full_model$coefficients
```

```
 (Intercept)      youtube     facebook    newspaper 
 3.526667243  0.045764645  0.188530017 -0.001037493 
```

```r
MSE=var(full_model$residuals); MSE
```

```
[1] 4.029288
```

```
MSE=summ$sigma^2

SSE_C=sum(summ$residuals^2)



# Inference: What is the error of $\hat\beta$? Is $f$ degenerate? I.e., is $\beta=0$?

#regular ANOVA
summary(full_model)
```

```
Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5932  -1.0690   0.2902   1.4272   3.3951 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,	Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
#confidence intervals for beta coefficients
confint.lm(full_model)
```

```
                  2.5 %     97.5 %
(Intercept)  2.78851474 4.26481975
youtube      0.04301371 0.04851558
facebook     0.17154745 0.20551259
newspaper   -0.01261595 0.01054097
```

```
#Partial F Test
model_red=lm(sales ~ newspaper, data = marketing)
sum_reduced=summary(model_red)
MSER=sum_reduced$sigma^2
SSE_R=sum(sum_reduced$residuals^2)

SSdrop=SSE_R-SSE_C

MSEdrop=SSdrop/2
Fstat=MSEdrop/MSE

1-pf(Fstat,2,196)
```

```
[1] 0
```

```
part_test=anova(model_red,full_model); part_test
```

```
Analysis of Variance Table

Model 1: sales ~ newspaper
Model 2: sales ~ youtube + facebook + newspaper
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    198 7394.1
2    196  801.8  2    6592.3 805.71 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
partial_c_det=SSdrop/SSE_R



SSER=sum(model_red$residuals*model_red$residuals); SSER
```

```
[1] 7394.119
```

```
dfer=model_red$df.residual; dfer
```

```
[1] 198
```

```
SSEC=sum(full_model$residuals*full_model$residuals); SSEC
```

```
[1] 801.8284
```

```
dfeC=full_model$df.residual; dfeC
```

```
[1] 196
```

```
SSdrop=SSER-SSEC; SSdrop
```

```
[1] 6592.29
```

```
dfdrop=dfer-dfeC

MSdrop=SSdrop/dfdrop; MSdrop
```

```
[1] 3296.145
```

```
R_online=SSdrop/SSER; R_online
```

```
[1] 0.8915586
```

```
part_test$F
```

```
[1]       NA 805.7141
```

```
part_test$`Pr(>F)`
```

```
[1]           NA 2.812622e-95
```

```
# Prediction: Predict any values if necessary.
# What if we have a 300$ budget and we only can pick one advertising method?
new_data=marketing[1:3,1:3]
new_data[1:3,]=diag(300,3)
predict(full_model,new_data)
```

```
        1         2         3
17.256061 60.085672  3.215419
```

```r
# It's best to put our money in FB... meta?

# What about intervals?

predict(full_model,new_data, interval = 'confidence')
```

```
        fit         lwr       upr
1 17.256061 16.56191879 17.950203
2 60.085672 55.25061022 64.920734
3  3.215419 -0.09445737  6.525296
```

It's a good time to stop and do another example to review the topics covered so far.

Example 3.7. In the dataset `mtcars` we have the following variables:

- mpg: Miles/(US) gallon
- cyl: Number of cylinders
- disp: Displacement (cu.in.)
- hp: Gross horsepower
- drat: Rear axle ratio
- wt: Weight (1000 lbs)
- qsec: 1/4 mile time
- vs: V/S
- am: Transmission (0 = automatic, 1 = manual)
- gear: Number of forward gears
- carb: Number of carburetors

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). Overall, we would like to investigate the relationship between `mpg` and the following variables: `cyl, disp, hp, drat, wt, qsec, gear, carb.` Let's investigate the following questions:

1. Assume the normal MLR model. Store the covariate matrix and response in a variable. Fit a normal MLR model to the data. – That is use `lm()` to fit the model.
2. What are the least squares estimates? What is the MSE?
3. Generate the ANOVA table. Is the model significant?
4. Test if `drat` contributes anything to the model, adjusting for the other covariates. Test if `drat` is related to `mpg`, without adjusting for the other covariates.
5. Test if the subset of variables `gear, carb` contribute to the model jointly, adjusting for the remaining covariates. What is the partial coefficient of determination? Interpret the partial coefficient of determination. Test if the subset of variables `gear, carb` contribute to the model jointly, without adjusting for the remaining covariates.

6. Compute a confidence interval for the mean mpg of cars with the following set of covariate values rmtcars[1,-1]*1.1. Compute a prediction interval for thempg‘ of a car with the above set of covariate values.
7. Compute a confidence interval for the coefficient for disp.
8. Compute and interpret the coefficient of determination.

```
data("mtcars")
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
dim(mtcars)
```

```
[1] 32 11
```

```
# 1.
# response~all variables minus the two variables we will not include
model=lm(mpg~.-vs-am,data=mtcars)
summ=summary(model)
summ
```

```
Call:
lm(formula = mpg ~ . - vs - am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0230 -1.6874 -0.4109  0.9640  5.4400 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.88964   17.81996   1.004   0.3259
cyl         -0.41460    0.95765  -0.433   0.6691
disp         0.01293    0.01758   0.736   0.4694
hp          -0.02085    0.02072  -1.006   0.3248
```

```
drat         1.10110    1.59806   0.689   0.4977  
wt          -3.92065    1.86174  -2.106   0.0463 *
qsec         0.54146    0.62122   0.872   0.3924  
gear         1.23321    1.40238   0.879   0.3883  
carb        -0.25510    0.81563  -0.313   0.7573  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.622 on 23 degrees of freedom
Multiple R-squared:  0.8596,	Adjusted R-squared:  0.8107 
F-statistic:  17.6 on 8 and 23 DF,  p-value: 4.226e-08
```

```
X=model.matrix(model)
Y=mtcars$mpg
X[1:5,]
```

```
                  (Intercept) cyl disp  hp drat    wt  qsec gear carb
Mazda RX4                   1   6  160 110 3.90 2.620 16.46    4    4
Mazda RX4 Wag               1   6  160 110 3.90 2.875 17.02    4    4
Datsun 710                  1   4  108  93 3.85 2.320 18.61    4    1
Hornet 4 Drive              1   6  258 110 3.08 3.215 19.44    3    1
Hornet Sportabout           1   8  360 175 3.15 3.440 17.02    3    2
```

```
Y
```

```
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```

```
# 2.
LSE=coef(model)
LSE
```

```
(Intercept)         cyl        disp          hp        drat          wt 
17.88963741 -0.41459575  0.01293240 -0.02084886  1.10109551 -3.92064847 
       qsec        gear        carb 
 0.54145693  1.23321026 -0.25509911 
```

```
MSE=summ$sigma^2
MSE
```

```
[1] 6.874941
```

```r
# 3.
null_model=lm(mpg~1,data=mtcars)
anova(null_model,model)
```

```
Analysis of Variance Table

Model 1: mpg ~ 1
Model 2: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     31 1126.05
2     23  158.12  8    967.92 17.599 4.226e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 4.
# Notice the p value is 0.5 , not sign.
summ$coefficients['drat',]
```

```
  Estimate Std. Error    t value   Pr(>|t|)
 1.1010955  1.5980601  0.6890201  0.4977032
```

```r
drat=lm(mpg~drat,,data=mtcars)
# Notice the p value is 1.78e-05 , sig! explain this difference!
summary(drat)
```

```


Call:
lm(formula = mpg ~ drat, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0775 -2.6803 -0.2095  2.2976  9.0225

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.525      5.477  -1.374     0.18
drat           7.678      1.507   5.096 1.78e-05 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.485 on 30 degrees of freedom
Multiple R-squared:  0.464, Adjusted R-squared:  0.4461
F-statistic: 25.97 on 1 and 30 DF,  p-value: 1.776e-05
```

```
# 5.
red_model=lm(mpg~.-vs-am-gear-carb,data=mtcars)
anova(red_model,model)
```

```
Analysis of Variance Table

Model 1: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am - gear - carb
Model 2: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     25 163.48
2     23 158.12  2    5.3532 0.3893 0.6819
```

```
ob=anova(red_model,model)
ob$`Sum of Sq`[2]/ob$RSS[1]
```

```
[1] 0.0327457
```

```
# 3.7% of the variation in mpg is explained from adding the covariate gear and carb to the mo

# 6.
new_ob=c(6.6,176,121,4.29,2.882,18.106,0,1.1,4.4,4.4)
new_ob=matrix(new_ob,nrow=1,ncol=length(new_ob))
colnames(new_ob)=names(mtcars[1,-1])
new_ob=data.frame(new_ob)
predict(model,new_ob, interval = 'confidence')
```

```
       fit      lwr      upr
1 22.43839 18.49468 26.38211
```

```
predict(model,new_ob, interval = 'prediction')
```

```
       fit     lwr      upr
1 22.43839 15.7322 29.14459
```

```
# 7.
confint(model)
```

```
                  2.5 %      97.5 %
(Intercept) -18.97375462 54.75302945
cyl          -2.39565252  1.56646102
disp         -0.02343129  0.04929609
hp           -0.06371601  0.02201829
drat         -2.20474377  4.40693480
wt           -7.77195651 -0.06934042
qsec         -0.74362628  1.82654014
gear         -1.66782660  4.13424711
carb         -1.94235037  1.43215215
```

```
#8.
summ$r.squared
```

```
[1] 0.8595764
```

```
# 85% of the variation in mpg is explained by cyl, disp, hp, drat, wt, qsec, gear and carb
```

Exercise 3.18. Interpret all of the above quantites.

3.4 Checking model assumptions

We learned how to test significance of one or multiple variables, compute confidence intervals for the estimated coefficients, mean response, and predicted response. All the methods rely on the assumptions! Recall that we assume 1. The relationship is linear $Y|X = X\beta + \epsilon$, 2. $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ 3. $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.

We now briefly discuss how to use the data to check if these assumptions are appropriate. We will cover this in more detail in the next chapter.

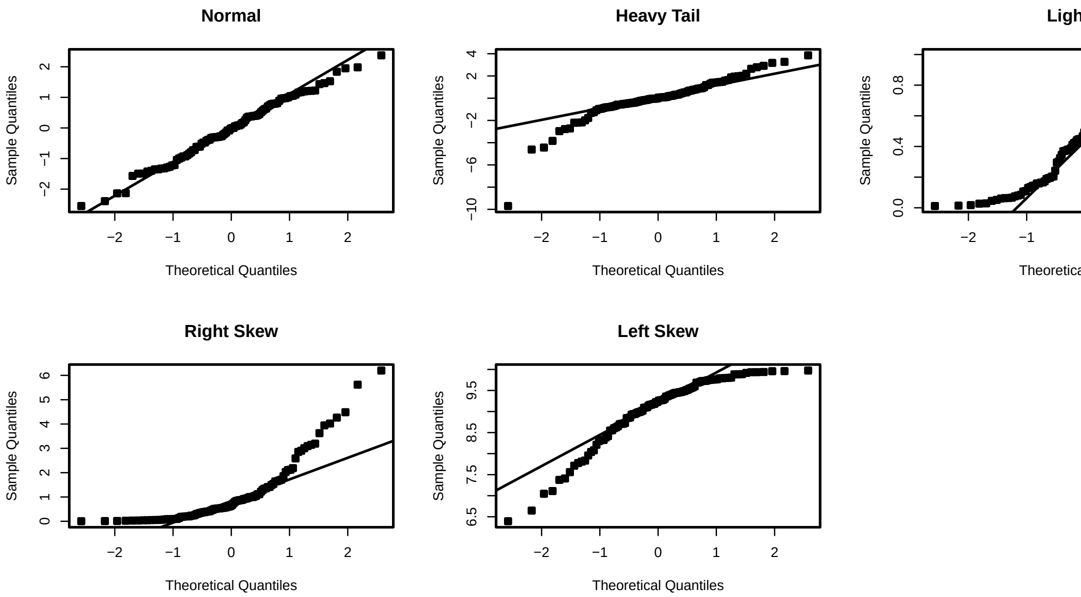
3.4.1 Checking normality

We do not know ϵ, however, we do know $\hat{\epsilon}$, which is our best proxy for the true random error vector ϵ. To check if the true random error vector is normally distributed we can use histograms and quantile-quantile plots. More specifically, if the histogram of the residuals looks more or less bell-shaped, with tails similar to the normal PDF, then the assumption of normality is valid.

Recall that a qq-plot compares the quantiles of the sample to the quantiles of the theoretical normal distribution. The x-axis represents the theoretical quantiles. The y-axis represents the sample quantiles. If the sample follows a normal distribution, the points in the qq-plot will approximately lie on a line.

Interpretation:

- Straight Line: If the points lie on or near the straight line, the sample appears normal.
- Heavy Tails: Points deviating upwards or downwards at the ends suggest the sample has heavier or lighter tails than the normal distribution.
- S-Shape: Points forming an S-shape indicate the sample has lighter tails and a heavier center than the normal distribution.



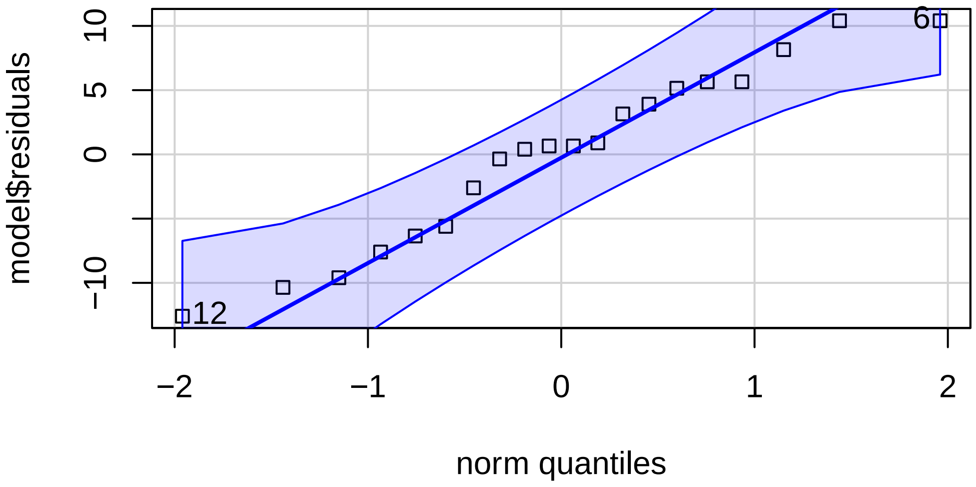
See below for an example:
Note that you will always have some deviation at the ends of the line in the qq-plot.

Example 3.8. In examples Example 3.1 and Example 3.6, check that the normality assumption is valid.

```
# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))
model= lm(BodyFat ~Weight, data = df)
summary(model)
```

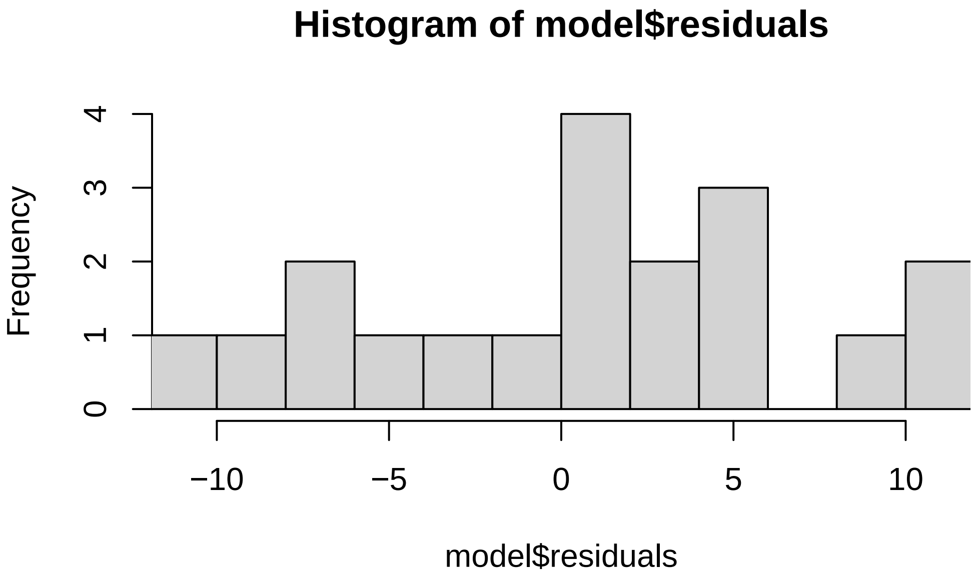
```
Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```
car::qqPlot(model$residuals,pch=22)
```



```
[1] 12  6
```

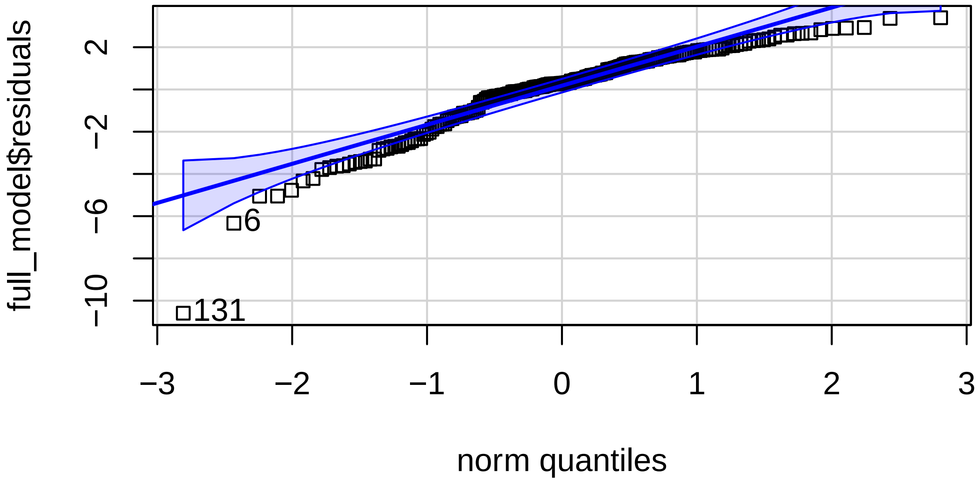
```
hist(model$residuals,breaks=10,xlim=c(-11,11))
```



```
# This appears okay!

# Let's do the next example
# install.packages('datarium')
data("marketing", package = "datarium")


# lm( sales~   , data= marketing)
full_model<- lm(sales ~ youtube+facebook+newspaper, data = marketing)
summary(full_model)
```

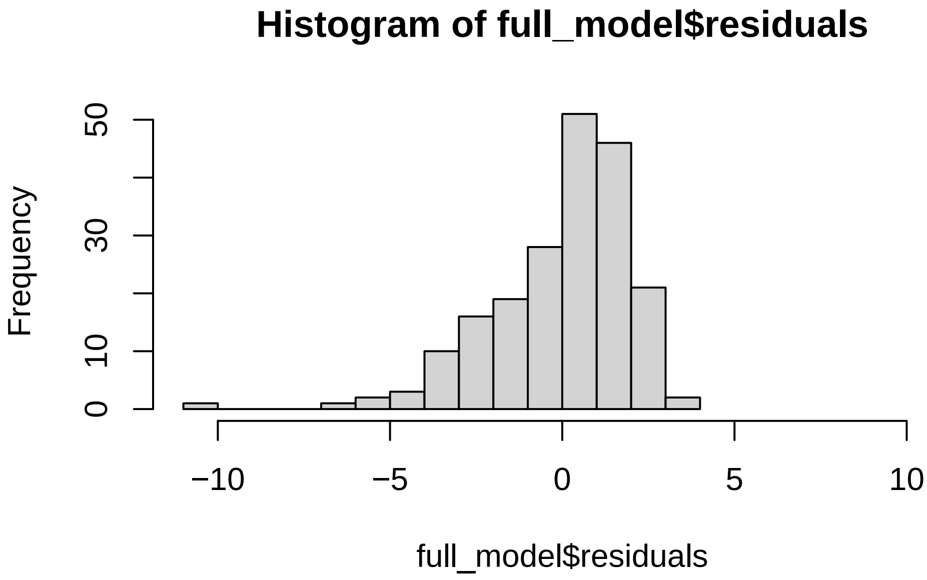
```
Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5932  -1.0690   0.2902   1.4272   3.3951 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
# Not great.
car::qqPlot(full_model$residuals,pch=22)
```



```
[1] 131   6
```

```
hist(full_model$residuals,breaks=10,xlim=c(-11,11))
```



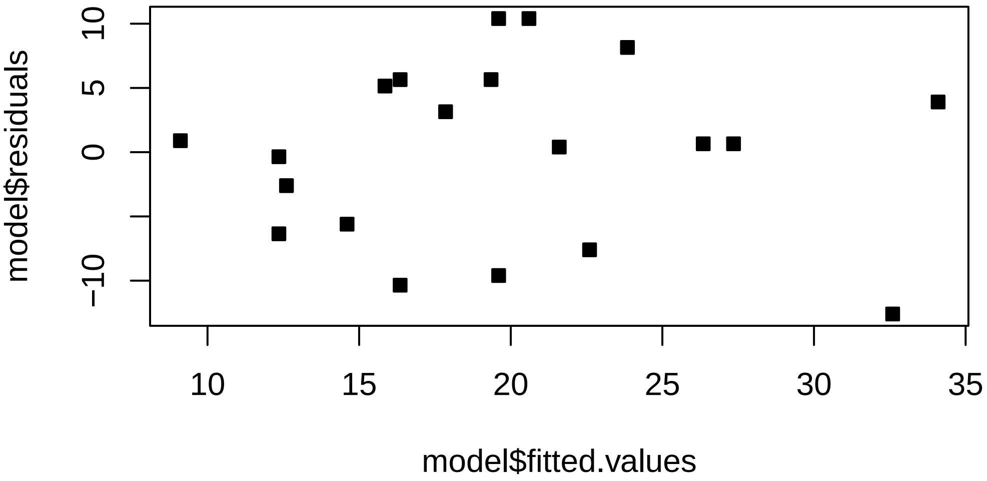
3.4.2 Checking the other assumptions

To check the remaining assumptions (constant variance, independence of residuals, zero mean and linear relationship), we can use some other diagnostic plots.

One plot is that of the fitted values $\widehat{Y}$ (x-axis) against the residuals $\hat{\epsilon}$ (y-axis). If the error depends on $\hat{y}$, then the identically distributed assumption on the errors is probably not valid. If the assumptions are valid, we should observe on the plots that at all levels of the response, the mean of the residuals is 0 and the variance remains the same. Thus, we should see a horizontal band centered at 0 containing the observations.

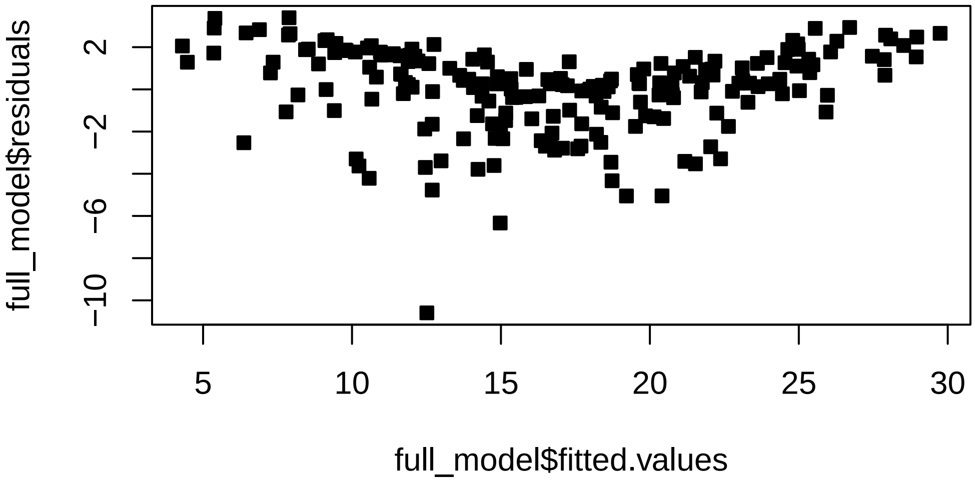
This appears to be the case in the body fat example:

```
plot(model$fitted.values,model$residuals,pch=22,bg=1)
```



Observe that in the marketing example, the residuals admit a pattern. This usually indicates either a non-linear relationship with the covariates, or an important covariate is missing. In this case, we would say the assumption of identically distributed errors is violated.

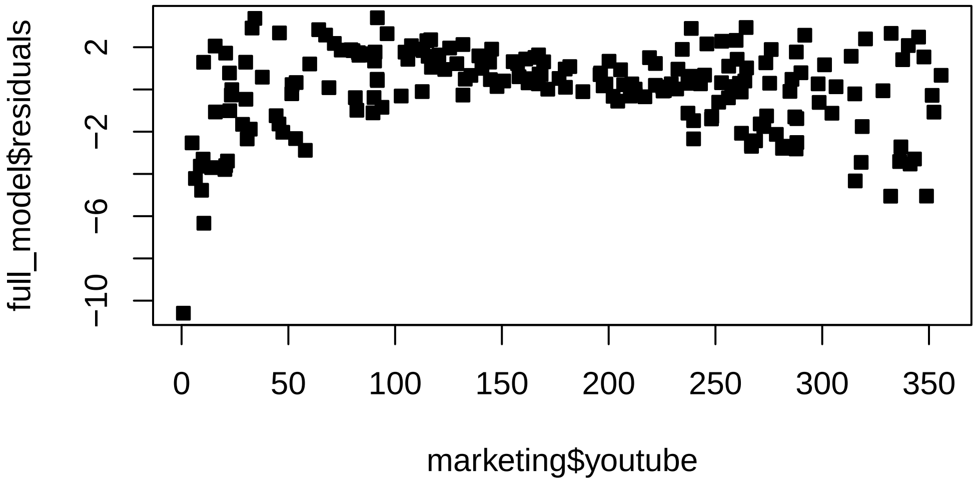
```
plot(full_model$fitted.values,full_model$residuals,pch=22,bg=1)
```



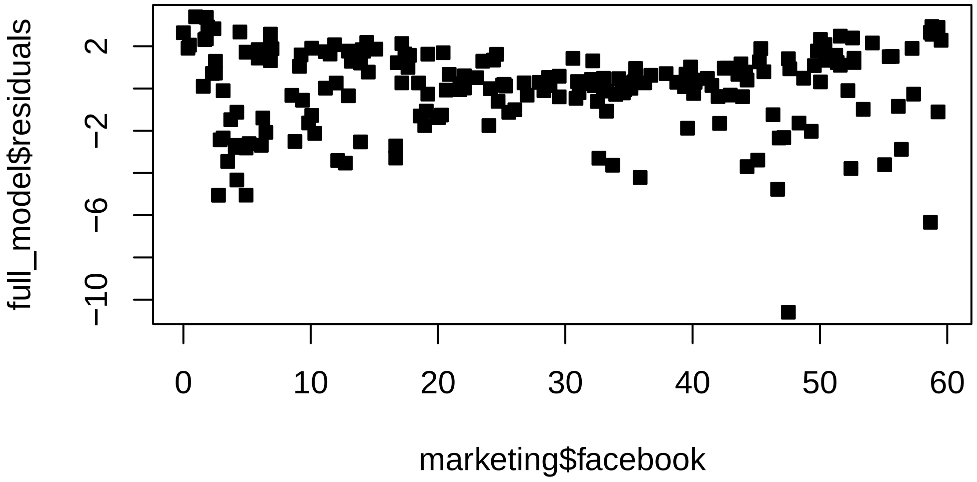
Plotting the residuals against the covariates can reveal dependence between the errors. For instance, if time is a covariate, you can plot the residuals over time to see if they have any relationship with time. If there appears to be dependence among the residuals, then the assumptions of the model are violated. That is, in these plots we should also see a horizontal band centered at 0 containing the observations. If not, then the residuals have a relationship with the given covariate.

Be VERY careful about the scale of your plot, as it can affect your interpretation. Zooming out or in too much can make everything look fine. In addition, the y-axis not being centered at 0 can cause you to misinterpret the plot.

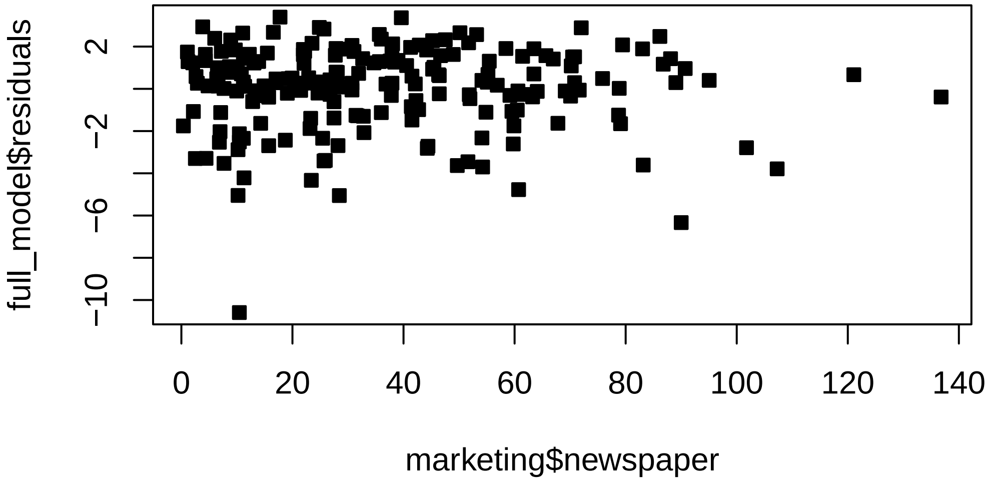
```
plot(marketing$youtube,full_model$residuals,pch=22,bg=1)
```



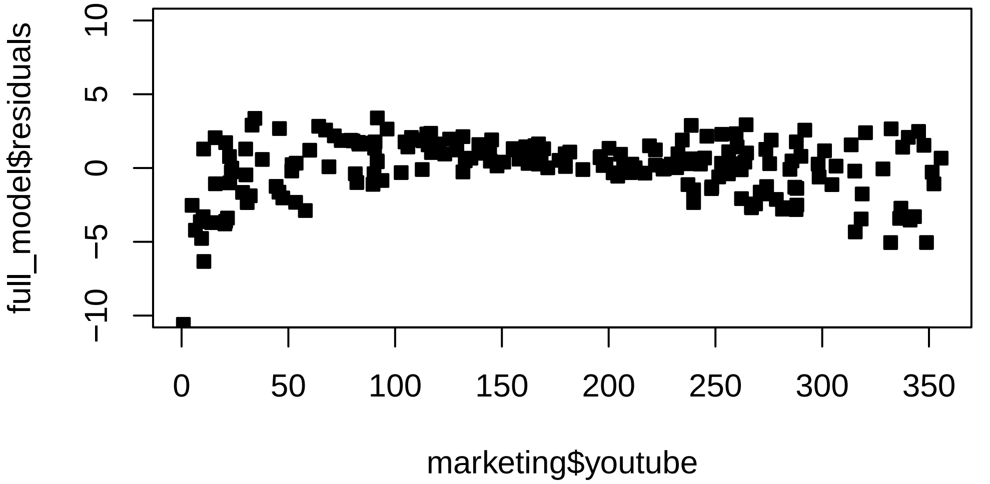
```
plot(marketing$facebook,full_model$residuals,pch=22,bg=1)
```



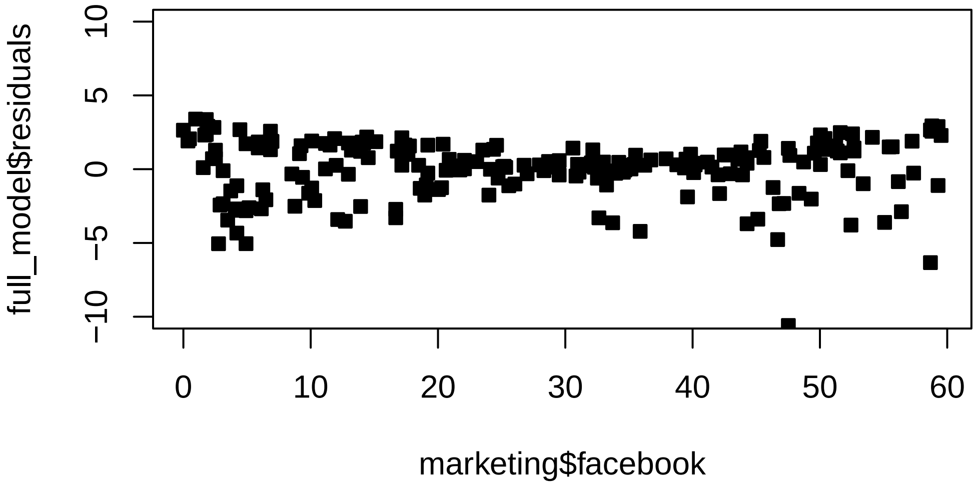
```r
plot(marketing$newspaper,full_model$residuals,pch=22,bg=1)
```



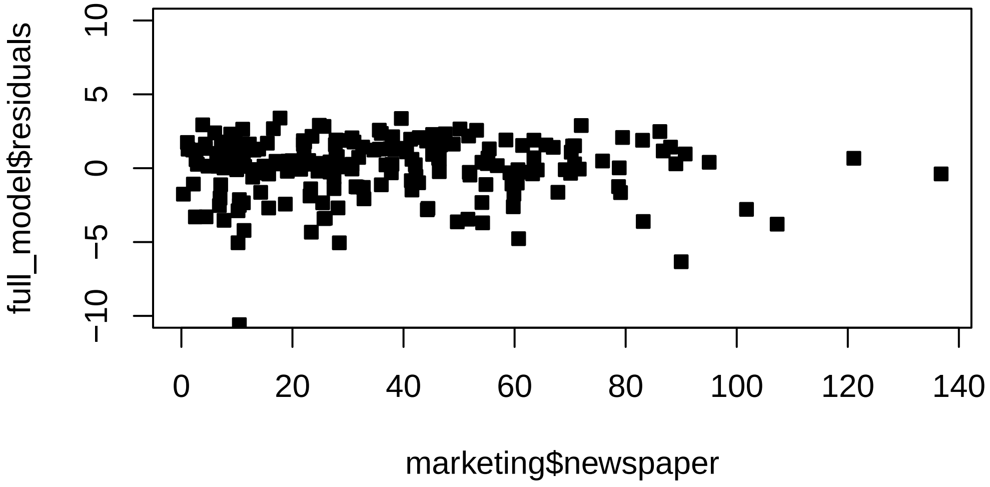
```r
plot(marketing$youtube,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```



```
plot(marketing$facebook,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```



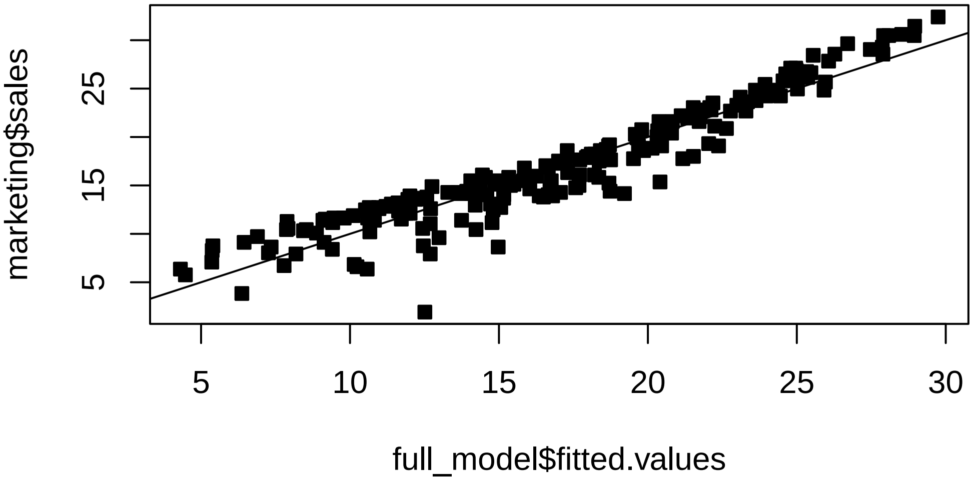
```
plot(marketing$newspaper,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```



Notice how the newspaper plot changes with the new axis limits. It appears that the variance of the error is changing with the value of the facebook and youtube budgets.

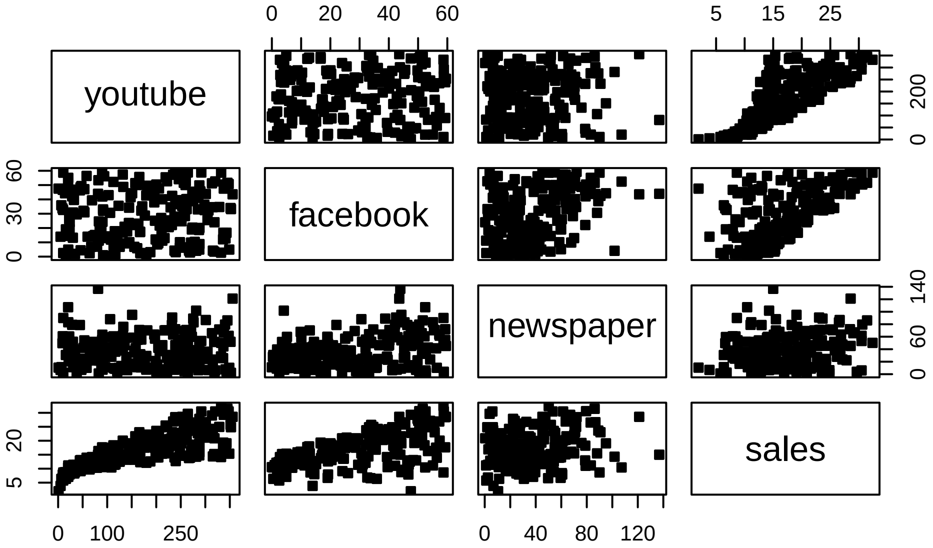
Another plot is that of the fitted values against the residuals. This gives an idea of the overall fit of the model. We should observe the points scatters around the line $y = x$.

```
plot(full_model$fitted.values,marketing$sales,pch=22,bg=1)
abline(0,1)
```



Notice that the line is slightly curved above the line at the ends. This means that at high and low values, the actual sales are empirically greater than as predicted by the model. Let's plot the actual data.

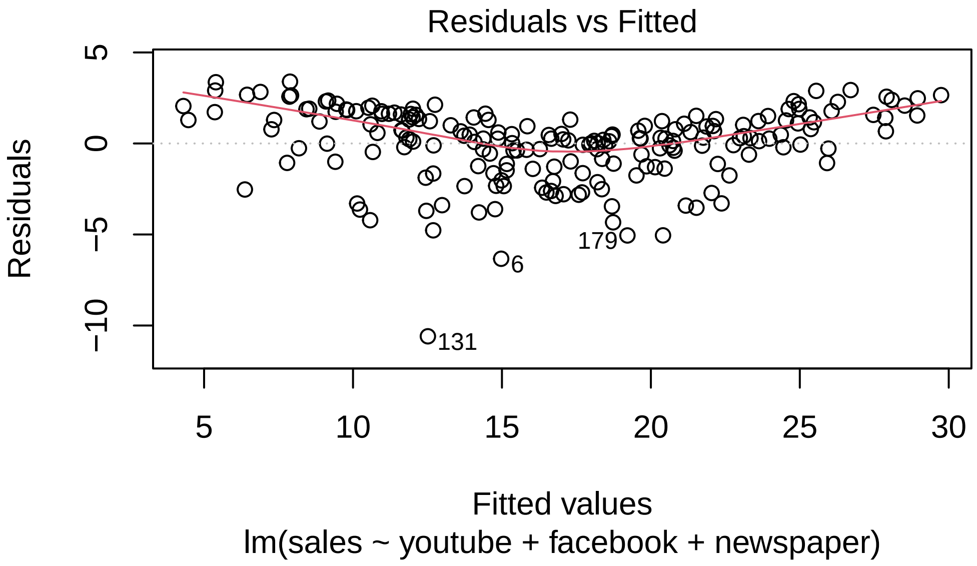
```
plot(marketing,pch=22,bg=1)
```

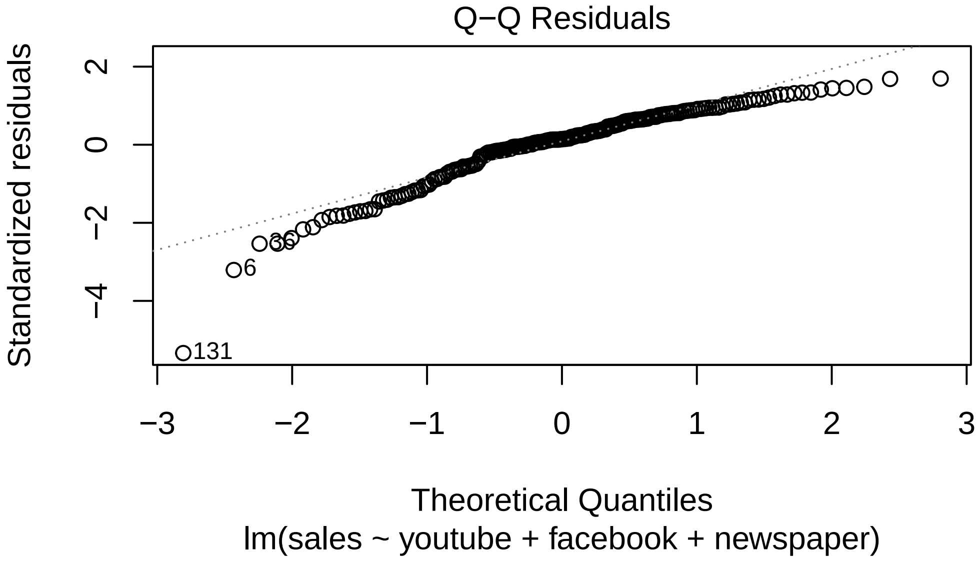


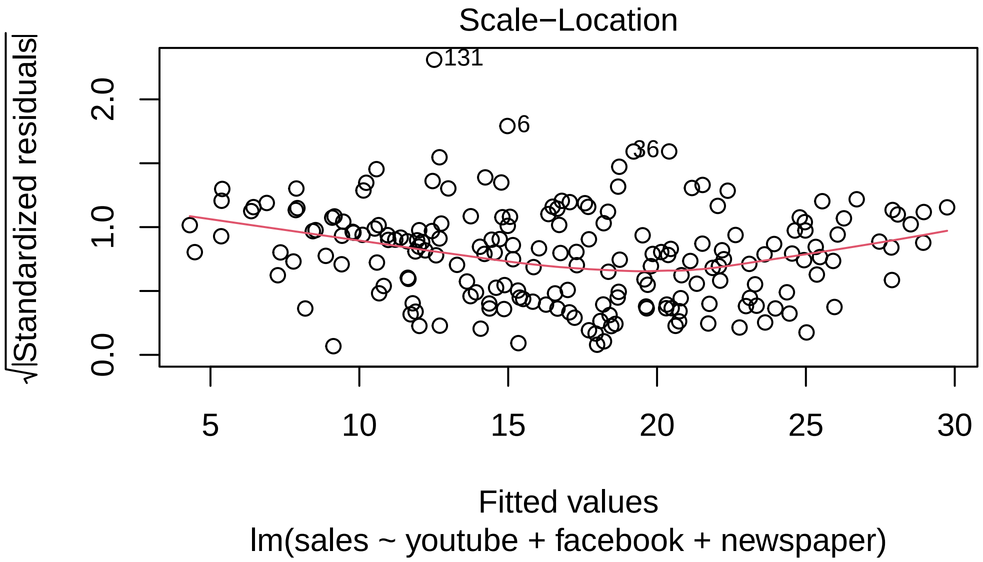
In this case, Youtube and facebook spending seems to have a nonlinear relationship with sales. We will see how to remedy this in later chapters.

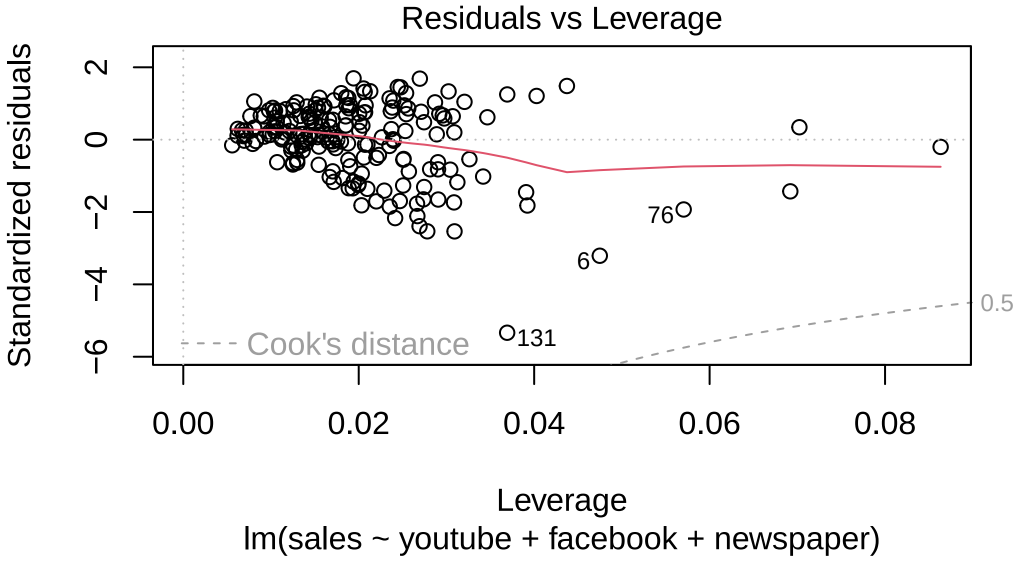
As a final note, observe that we can put the `model` object in the `plot()` function to obtain the diagnostic plots.

```
plot(full_model)
```


Residuals vs Fitted
Residuals
5
0
−5
−10
179
6
131
5
10
15
20
25
30
Fitted values
lm(sales ~ youtube + facebook + newspaper)


Q−Q Residuals
Standardized residuals
2
0
−2
−4
36
6
131
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(sales ~ youtube + facebook + newspaper)





We will learn in later chapters how to check the assumptions more thoroughly and how to remedy violations of the assumptions.

3.4.3 Homework stop 5

Complete the assigned textbook problems for Chapter 4.

Exercise 3.19. List the assumptions for the normal MLR model and the MLR model. Write down how you would check each assumption.

3.5 Simple linear regression

A special case of the multiple linear regression is **simple linear regression**. A simple linear regression model is a regression model with **one explanatory variable**: $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$.

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

3.5.1 Estimated Coefficients

In this case, following some matrix manipulations (verify this for homework), we have

$$X^\top X = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix}, \; X^\top y = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}.$$

Now, recall if

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

From MATH 1131 (or simple algebraic manipulation), we know

$$\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} = \sum_{i=1}^n x_i y_i - n^{-1} \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)$$

$$\sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2 = \sum_{i=1}^n x_i^2 - n^{-1} \left(\sum_{i=1}^n x_i \right)^2.$$

Therefore

$$
\begin{aligned}
(X^\top X)^{-1} &= \frac{1}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix} \\
&= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix}.
\end{aligned}
$$

To summarize:

$$
\begin{aligned}
X^\top X &= \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \\
X^\top y &= \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix} \\
(X^\top X)^{-1} &= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}
\end{aligned}
$$

Now, we have that

$$
\begin{aligned}
\hat{\beta} &= \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = (X^\top X)^{-1} X^\top y \\
&= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i \\ -\sum_{i=1}^n x_i \sum_{i=1}^n y_i + n\sum_{i=1}^n x_i y_i \end{pmatrix}.
\end{aligned}
$$

Now,

$$
\hat{\beta}_1 = \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \left(-\sum_{i=1}^n x_i \sum_{i=1}^n y_i + n\sum_{i=1}^n x_i y_i \right).
$$

Exercise 3.20. Let's show that

$$
\hat{\beta}_1 = \frac{n\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}.
$$

Does this look familiar? We see that,

$$
\hat{\beta}_1 = \hat{\text{cov}}(X, Y) \frac{\hat{\sigma}_y}{\hat{\sigma}_x},
$$

where $\hat{\text{cov}}(X, Y)$ is the estimated correlation between X and Y. Let's interpret this:

1. If $\hat{\text{cov}}(X, Y) \approx 0$ then $\hat{\beta}_1 \approx 0$ - low correlation implies an estimated slope close to 0.
2. The estimated coefficient $\hat{\beta}_1$ is the product of the estimated correlation between X and Y and the ratio of the estimated standard deviation of Y to that of X.

Now, looking at the intercept term, we have

$$\hat{\beta}_0 = \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \left(\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i \right).$$

Exercise 3.21. Show that

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

Observe that the intercept is the mean of Y minus the mean of X times the estimated slope. In essence, it tells us that the intercept $(\hat{\beta}_0)$ represents the value of (X) when (X) is at its mean value $((\bar{X}))$ and that $(\bar{X})$ is adjusted by subtracting the contribution of $(\hat{\beta}_1 \bar{X})$.

This adjustment ensures that the regression line passes through the point $((\bar{X}, \bar{X}))$, which is the point of averages for the data.

3.5.2 Inference in SLR

We can also simplify the values used for inference in the SLR model. Recall that $\operatorname{Var}\left[\hat{\beta}\right] = (X^\top X)^{-1}\sigma^2$, and so we have

$$\begin{aligned}
\operatorname{Var}\left[\hat{\beta}_0\right] &= \frac{\sum_{i=1}^n x_i^2}{n\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 = \frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2 + n\bar{x}^2}{n\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 \\
&= \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right]\sigma^2 \\
\operatorname{Var}\left[\hat{\beta}_1\right] &= \frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 \\
\operatorname{cov}(\hat{\beta}_0, \hat{\beta}_1) &= -\frac{\bar{x}}{\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2.
\end{aligned}$$

We know from previous sections that a $(1-\alpha)100$ confidence interval of β_i, where $i = 0, 1$, is

$$\hat{\beta}_i \pm t_{df_E, \alpha/2}\sqrt{\widehat{var}(\hat{\beta}_i)}.$$

Similarly, let β_i^0 be a hypothesized value of β_i, for $i = 0, 1$. If we want to test whether $\beta_i = \beta_i^0$, then the observed test statistic is given by

$$\frac{\hat{\beta}_i - \beta_i^0}{\sqrt{\widehat{var}(\hat{\beta}_i)}},$$

and the corresponding p-value is obtained via the t_{dfE} distribution as usual.

Note

Similarly, inference for the mean response and predictions can be obtained. We can also simplify the $ANOVA$ table, R^2, etc. For instance, the R^2 is the square of the sample correlation coefficient between X and Y.

3.5.3 Inference for the correlation coefficient

If we are interested in doing a hypothesis test, or constructing confidence intervals for the correlation between two variables, say X and Y, we can use the simple linear regression model.

We have already derived the relationship between the estimated correlation coefficient and the estimated slope of the simple linear regression model. More specifically, if the estimated correlation coefficient is 0, then the estimated slope of the simple linear regression is 0. One can show that the same relationship holds at the population level: $\beta_1 = \rho\sigma_y/\sigma_x$, where $\rho = \text{corr}\,[X, Y]$.

Now, suppose that we want to test if $H_0 : \rho = 0$ versus $H_a : \rho \neq 0$. Using the fact that $\beta_1 = \rho\sigma_y/\sigma_x$, the above test is equivalent to the statement $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 \neq 0$. Therefore, we can just test if the slope parameter in the model $Y|X = \beta_0 + \beta_1 X + \epsilon$ is 0.

Letting $\hat{\rho} = \widehat{\text{corr}}(X, Y)$ The observed test statistic is then:

$$\frac{\hat{\beta}_1}{\sqrt{\widehat{var}(\hat{\beta}_1)}} = \frac{\hat{\rho}\sqrt{n-2}}{\sqrt{1-\hat{\rho}^2}},$$

and the corresponding p-value is obtained based on the t_{dfE} distribution.

However, when the hypothesized value for ρ is non-zero, the problem becomes very complicated. The exact distribution of $\hat{\rho}$ is extremely difficult to obtain under the null hypothesis. The following procedure gives an approximation of the distribution of a function of $\hat{\rho}$ under the null hypothesis. In particular, Fisher suggested the transformation for $\rho \in (0, 1)$,

$$\theta = \frac{1}{2}\log\frac{1+\rho}{1-\rho}.$$

Then

$$\hat{\theta} = \frac{1}{2}\log\frac{1+\hat{\rho}}{1-\hat{\rho}},$$

is an estimate of θ, where $\hat{\theta}$ is approximately distributed as normal with mean θ and variance $\frac{1}{n-3}$. Hence, an approximate $(1-\alpha)100$ confidence interval of θ is

$$\hat{\theta} \pm z_{\alpha/2}\sqrt{\frac{1}{n-3}},$$

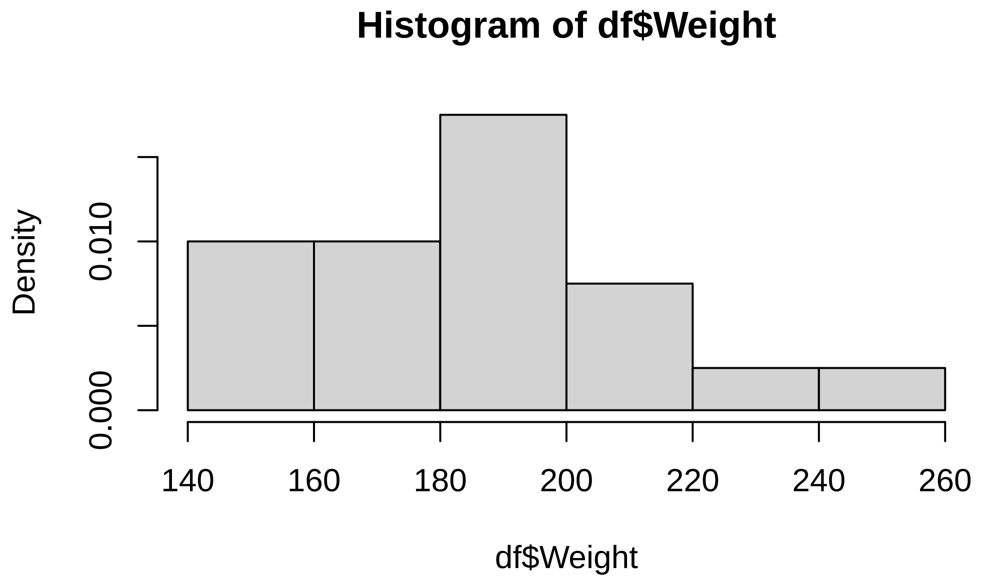
and the corresponding confidence interval of ρ can be obtained by the inverse transformation. Similarly, if the hypothesized value of ρ is ρ_0, then the hypothesized value of θ is $\theta_0 = \frac{1}{2} \log \frac{1+\rho_0}{1-\rho_0}$. The observed test statistic can be obtained and the corresponding p-value can be obtained based on the standard normal distribution.

Example 3.9. In Example 3.1 test if the correlation between body fat and weight is 0. Next, test if the correlation is greater than 1/2. Construct a 95% CI for ρ.

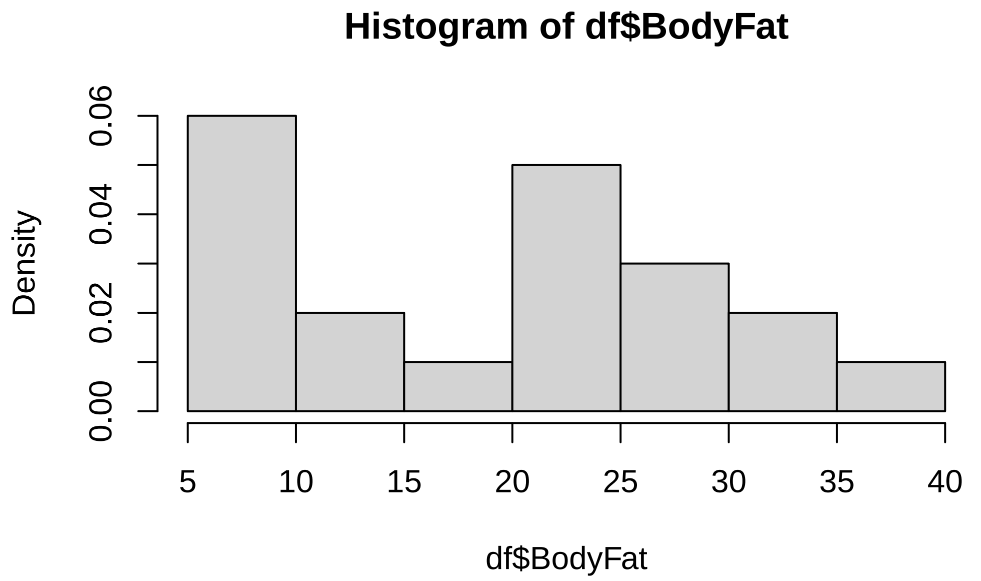
```
############################ Exploratory
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 ,
         173 , 187 , 188 , 188 , 240 , 175 , 168 ,
         246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 ,
           30 , 10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))


cor(df)
```

```
           Weight   BodyFat
Weight  1.0000000 0.6966328
BodyFat 0.6966328 1.0000000
```

```
hist(df$Weight,freq=F)
```



```
hist(df$BodyFat,freq=F)
```



```
summary(df)
```

```
     Weight          BodyFat     
 Min.   :146.0   Min.   : 6.00  
 1st Qu.:171.8   1st Qu.:10.00  
 Median :187.5   Median :21.00  
 Mean   :188.6   Mean   :19.75  
 3rd Qu.:201.2   3rd Qu.:27.25  
 Max.   :246.0   Max.   :38.00  
```

```
cor(df)[1,2]
```

```
[1] 0.6966328
```

```
X=cbind(rep(1,nrow(df)), df$Weight)
Y=df$BodyFat

beta_hat= solve(t(X)%*%X)%*%t(X)%*%Y
beta_hat
```

```
            [,1]
[1,] -27.3762623
[2,]   0.2498741
```

```
model=lm(BodyFat~ Weight,df)
model
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Coefficients:
(Intercept)       Weight  
   -27.3763       0.2499  
```

```
summary(model)
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```
cor(df)[1,2]^2
```

```
[1] 0.4852972
```

```
cor(df)[1,2]
```

```
[1] 0.6966328
```

```
a=function(x){
  (exp(2*x)-1)/(exp(2*x)+1)
}
a(1.36)
```

```
[1] 0.8763931
```

3.5.4 Homework stop 6

- Complete the Chapter 2 questions in the textbook.

Exercise 3.22. For

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

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

- Compute $\hat{\beta}$, Var $\left[\hat{\beta}_1\right]$, Var $\left[\hat{\beta}_0\right]$, cov $[(]\,\hat{\beta}_0, \hat{\beta}_1)$
- Show $\hat{\beta}_1 = r\frac{\hat{\sigma}_y}{\hat{\sigma}_x}$

3.6 Additional concepts & examples

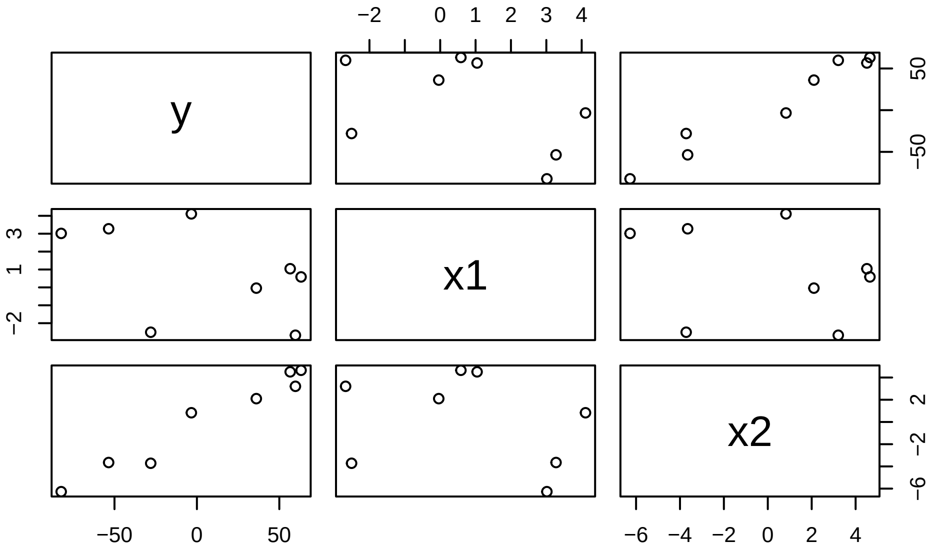
Here we touch on a few important examples and notes about the MLR.

3.6.1 Beware scatter plots in MLR

Sometimes, scatter plots are misleading for determining the relationship between Y and a collection of p covariates. In the following example, it appears that $X1$ and Y do not have a relationship, when in fact they do. Generally, this phenomena goes away with higher sample sizes.

```
# Scatter diagram beware?
# x1=c(2,3,4,1,5,6,7,8)
# x2=c(2,3,4,1,5,6,7,8)
# x=c(2,3,4,1,5,6,7,8)

# x1=1:8
# x2=c(2,1:6,4)
# y=8-5*x1+12*x2+rnorm(8,0,2)

set.seed(445)



n=8
x1=rnorm(n,5,5)
x2=rnorm(n,3,5)
y=8-5*x1+12*x2+rnorm(n,0,2)

df=data.frame(cbind(y,x1,x2))
```

```
plot(df)
```



Next, we do an example from the textbook, which uses the NFL data. Specifically, we try to evaluate the relationship between number of wins and several explanatory variables.

Example 3.10. Using the following NFL data, complete 3.1-3.4, 4.1 and 4.2 in the textbook.

```
################# NFL example ################
# This gives you the data sets used in the textbook
# install.packages('MPV')
df=MPV::table.b1
# Note for more information, run ?MPV::table.b1

head(df)
```

```
   y   x1   x2   x3   x4 x5  x6   x7   x8   x9
1 10 2113 1985 38.9 64.7  4 868 59.7 2205 1917
2 11 2003 2855 38.8 61.3  3 615 55.0 2096 1575
3 11 2957 1737 40.1 60.0 14 914 65.6 1847 2175
4 13 2285 2905 41.6 45.3 -4 957 61.4 1903 2476
5 10 2971 1666 39.2 53.8 15 836 66.1 1457 1866
6 11 2309 2927 39.7 74.1  8 786 61.0 1848 2339
```

```
# names too long
names(df)
```

```
 [1] "y"  "x1" "x2" "x3" "x4" "x5" "x6" "x7" "x8" "x9"
```

```
# rename to make it easier
names(df)=c("Wins","RushY","PassY","PuntA","FGP","TurnD","PenY","PerR","ORY","OPY")
names(df)
```

```
 [1] "Wins"  "RushY" "PassY" "PuntA" "FGP"   "TurnD" "PenY"  "PerR"  "ORY"  
[10] "OPY"  
```

```
# Wins~ beta_1+beta_2Passing yrds+beta_3per_rush+beta_4ORY+epsilon
# summary(df)
# plot(df)
# run the model
regression_model=lm(  Wins ~ PassY+PerR+ORY  ,data= df )
summary(regression_model)
```

```

Call:
lm(formula = Wins ~ PassY + PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0370 -0.7129 -0.2043  1.1101  3.7049 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.808372   7.900859  -0.229 0.820899    
PassY        0.003598   0.000695   5.177 2.66e-05 ***
PerR         0.193960   0.088233   2.198 0.037815 *  
ORY         -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.706 on 24 degrees of freedom
Multiple R-squared:  0.7863,	Adjusted R-squared:  0.7596 
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

```
# model=lm(Wins~PassY+PerR+ORY,data=df)
# get the confidence intervals.
confint(regression_model)
```

```
                    2.5 %       97.5 %
(Intercept) -18.114944410 14.498200293
PassY         0.002163664  0.005032477
PerR          0.011855322  0.376065098
ORY          -0.007451027 -0.002179961
```

What conclusions can you make from this output? - All variables seem important! For isntance, we sdee that for every 1% increase in percentage rushing, there is a 0.193960 increase in number of wins, on average, holding passing yards and opponent rushing yards constant.

```
#### CI
# mean response of z'\beta , z=(2000,60,1900)'
new_data=data.frame(  matrix(c(2000,60,1900),ncol=3) )
names(new_data)
```

```
[1] "X1" "X2" "X3"
```

```
names(new_data)=c( 'PassY','PerR','ORY' )

predict(regression_model, new_data  ,   interval = 'confidence')
```

```
       fit      lwr      upr
1 7.875942 7.072672 8.679213
```

```
predict(regression_model, new_data  ,   interval = 'predict')
```

```
       fit      lwr     upr
1 7.875942 4.263986 11.4879
```

```
## ANOVA

regression_model_reduced=lm(  Wins ~ 1  ,data= df )
summary(regression_model_reduced)
```

```
Call:
lm(formula = Wins ~ 1, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9643 -2.9643 -0.4643  3.0357  6.0357 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.9643     0.6576   10.59 4.09e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.48 on 27 degrees of freedom
```

```
anova(regression_model_reduced,regression_model)
```

```
Analysis of Variance Table

Model 1: Wins ~ 1
Model 2: Wins ~ PassY + PerR + ORY
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     27 326.96                                  
2     24  69.87  3    257.09 29.437 3.273e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# subset test

regression_model_reduced=lm(  Wins ~ PassY ,data= df )
anova(regression_model_reduced,regression_model)
```

```
Analysis of Variance Table

Model 1: Wins ~ PassY
Model 2: Wins ~ PassY + PerR + ORY
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     26 250.77                                  
2     24  69.87  2     180.9 31.069 2.189e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# summary(df)

summary(regression_model)
```

```

Call:
lm(formula = Wins ~ PassY + PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0370 -0.7129 -0.2043  1.1101  3.7049 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.808372   7.900859  -0.229 0.820899    
PassY        0.003598   0.000695   5.177 2.66e-05 ***
PerR         0.193960   0.088233   2.198 0.037815 *  
ORY         -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.706 on 24 degrees of freedom
Multiple R-squared:  0.7863,	Adjusted R-squared:  0.7596 
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

```
# anova(regression_model)
summ=summary(regression_model)
summ$r.squared
```

```
[1] 0.7863069
```

```
summ$adj.r.squared
```

```
[1] 0.7595953
```

```
regression_model2=lm(Wins~PassY+ORY,data=df)


SSER=sum(regression_model2$residuals*regression_model2$residuals); SSER
```

```
[1] 83.9382
```

```
dfer=regression_model2$df.residual; dfer
```

```
[1] 25
```

```
SSEC=sum(regression_model$residuals*regression_model$residuals); SSEC
```

```
[1] 69.87
```

```
dfeC=regression_model$df.residual; dfeC
```

```
[1] 24
```

```
SSdrop=SSER-SSEC; SSdrop
```

```
[1] 14.06819
```

```
dfdrop=dfer-dfeC

MSdrop=SSdrop/dfdrop; MSdrop
```

```
[1] 14.06819
```

```
R_prp=SSdrop/SSER; R_prp
```

```
[1] 0.1676018
```

```
MSdrop
```

```
[1] 14.06819
```

```
1-pf(MSdrop,dfdrop,dfeC)
```

```
[1] 0.000986662
```

```
cor(regression_model$fitted.values , df$Wins)^2
```

```
[1] 0.7863069
```

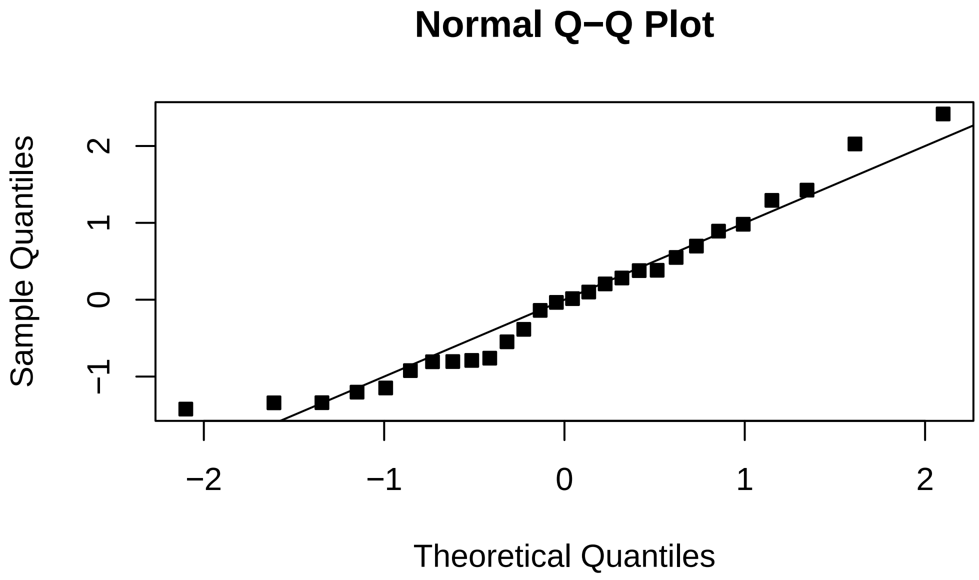
```
confint(regression_model2)
```

```
                   2.5 %       97.5 %
(Intercept)  9.321778092 20.103571885
PassY        0.001654121  0.004568143
ORY         -0.008797465 -0.004819085
```

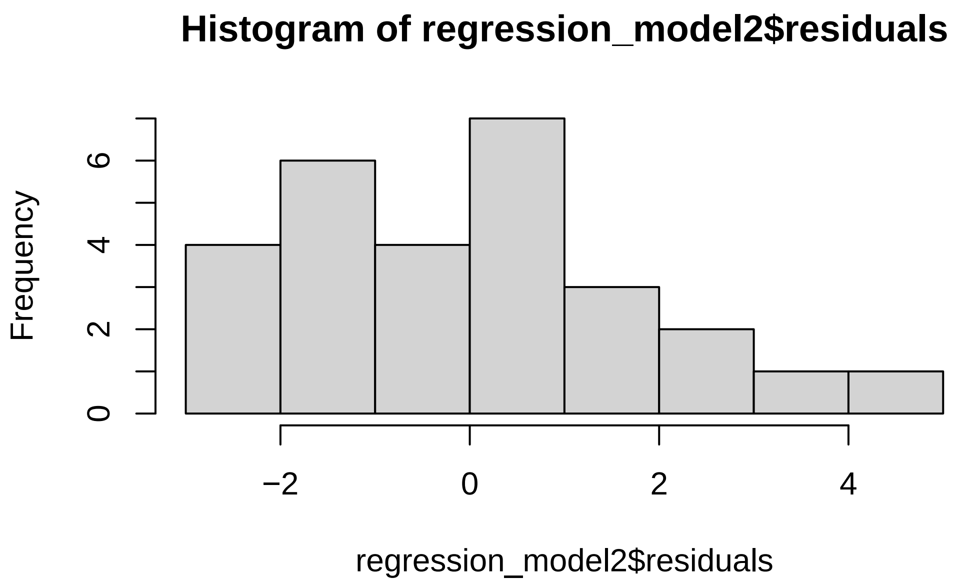
```
new_data=df[1,c(3,8,9)]
new_data[1,]=c(2300 , 56 , 2100)
predict(regression_model2,new_data,interval = 'confidence')
```

```
     fit      lwr      upr
1 7.5709 6.814662 8.327138
```

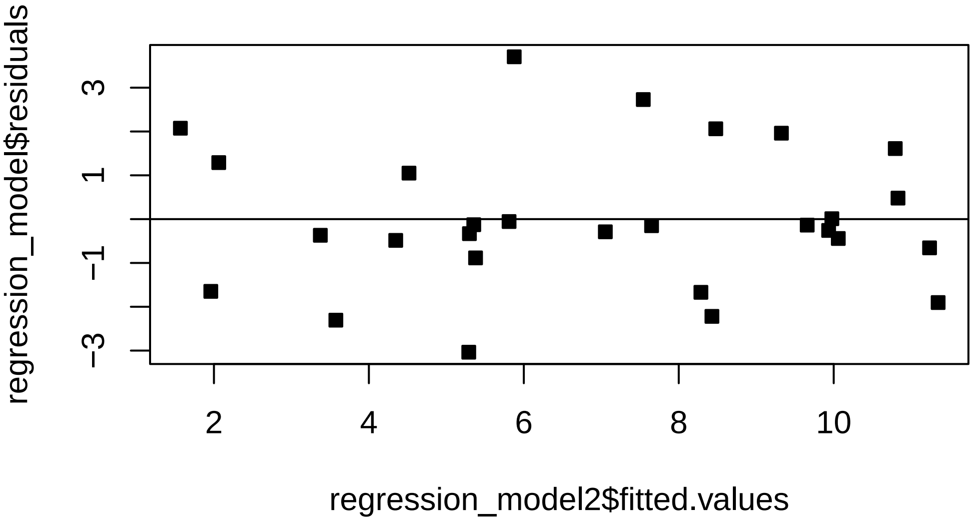
```
#########  check the fit #######
MSE=summ$sigma^2
qqnorm(regression_model2$residuals/summ$sigma,pch=22,bg=1)
abline(0,1)
```



```
hist(regression_model2$residuals,breaks=6)
```



```
plot(regression_model2$fitted.values,regression_model$residuals,pch=22,bg=1)
abline(h=0)
```

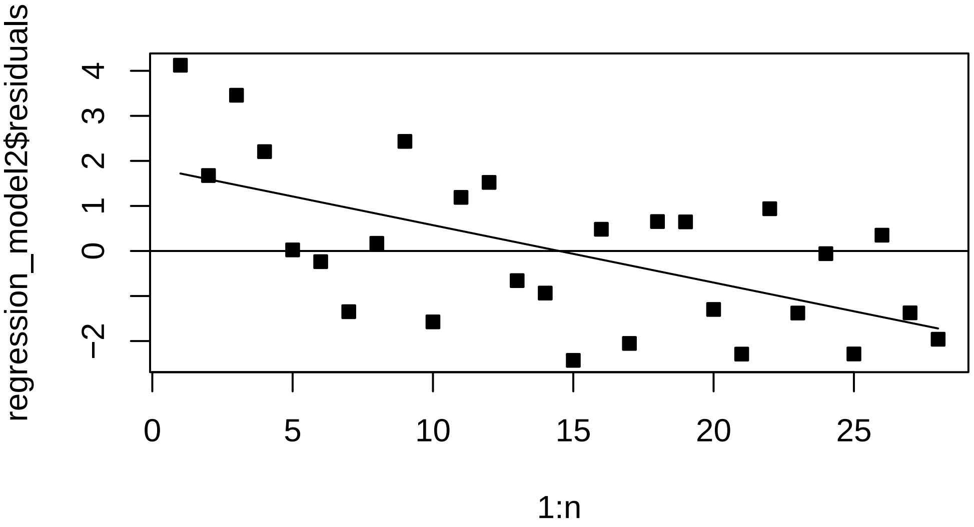


```
n=nrow(df)
plot(1:n,regression_model2$residuals,pch=22,bg=1)
abline(h=0)

time=(1:n)
res=lm(regression_model2$residuals~time)
summary(res)
```

```
Call:
lm(formula = regression_model2$residuals ~ time)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.36425 -1.04520 -0.07845  1.16457  2.40353 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.8479     0.5610   3.294 0.002852 ** 
time         -0.1274     0.0338  -3.771 0.000848 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.445 on 26 degrees of freedom
Multiple R-squared:  0.3535,    Adjusted R-squared:  0.3286
F-statistic: 14.22 on 1 and 26 DF,  p-value: 0.0008481
```

```
lines(time,res$fitted.values)
```



```
regression_model3=lm(Wins~PerR+ORY,data=df)
summ3=summary(regression_model3)
summ3
```

```
Call:
lm(formula = Wins ~ PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7985 -1.5166 -0.5792  1.9927  4.5248 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)  
(Intercept) 17.944319   9.862484   1.819  0.08084 .
```

```
PerR         0.048371   0.119219   0.406  0.68839
ORY         -0.006537   0.001758  -3.719  0.00102 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.432 on 25 degrees of freedom
Multiple R-squared:  0.5477,    Adjusted R-squared:  0.5115
F-statistic: 15.13 on 2 and 25 DF,  p-value: 4.935e-05
```

```
summ3$r.squared
```

```
[1] 0.5476628
```

```
summ3$adj.r.squared
```

```
[1] 0.5114759
```

```
confint(regression_model)
```

```
                    2.5 %       97.5 %
(Intercept) -18.114944410 14.498200293
PassY         0.002163664  0.005032477
PerR          0.011855322  0.376065098
ORY          -0.007451027 -0.002179961
```

```
confint(regression_model3)
```

```
                  2.5 %       97.5 %
(Intercept) -2.36784828 38.256485319
PerR        -0.19716429  0.293906022
ORY         -0.01015637 -0.002916818
```

```
new_data3=df[1,c(8,9)]
new_data3[1,]=c(56 , 2100)
predict(regression_model3,new_data3,interval = 'confidence')
```

```
       fit      lwr      upr
1 6.926243 5.828643 8.023842
```

```
predict(regression_model2,new_data,interval = 'confidence')
```

```
     fit      lwr      upr
1 7.5709 6.814662 8.327138
```

Be careful about extrapolating beyond the region containing the original observations. It is very possible that a model that fits well in the region of the original data will perform poorly outside that region. It is easy to inadvertently extrapolate, since the levels of the regressors jointly define a region containing the data which is impossible to visualize in its entirety beyond 2 dimensions. Ideally, we want to make inferences which lie inside the convex hull of the regressors.

We can use the diagonal of the hat matrix $H = X(X^\top X)^{-1}X^\top$. In general, the point that has the largest value of h_{ii}, say h_{max}, will lie on the boundary of the convex hull in a region of the x-space where the density of the observations is relatively low. Points that lie in the set $\{x^\top (X^\top X)^{-1}x \leq h_{max}\}$ enclose the convex hull. Thus, for a value we are interested in predicting, say y, we can check if we are extrapolating with $y^\top (X^\top X)^{-1}y \leq h_{max}$.

A serious problem that may dramatically impact the usefulness of a regression model is multicollinearity, or near - linear dependence among the regression variables. Multicollinearity implies near - linear dependence among the regressors. The regressors are the columns of the X matrix, so clearly an exact linear dependence would result in a singular $X^\top X$. This will impact our ability to estimate β.

We can check for this dependence with the **variance inflation factor** (VIF). The variance inflation factor can be written as $(1 - R_j^2)^{-1}$, where R_j^2 is the coefficient of determination obtained from regressing X_j on the other regressor variables. If VIF is large, say > 3, then you will likely need to make some changes to your regression model.

Sometimes, you may observe that regression coefficients have the a sign that is unexpected, or contradicts nature. This is likely due to one of the following:

- The range of some of the regressors is too small – if the range of some of the regressors is too small, then the variance of $\hat{\beta}$ is high.
- Important regressors have not been included in the model.
- Multicollinearity is present.
- Computational errors have been made.

We close this Chapter with the following statement. Recall the modelling overview from Chapter 1:

- Posit the model: What is the linear regression model – what are all the assumptions of the linear regression model?
- Estimation: How can we estimate parameters of the linear regression model?

- Inference: How can we compute confidence intervals and run hypothesis tests associated with the linear regression model?
- Fit: Does our fitted line match up with the data? What about the normality assumption? Do the errors appear normal? Do the errors seem independent? Is the variance constant? How much variability is explained by our model?
- Prediction: How can we predict a new Y? What is the error of this prediction

If you have learned the concepts of this chapter, you should be able to complete all of these steps! In the following chapters, we will discuss different problems that can arise in regression modelling and how to remedy them.

3.6.2 Homework questions

Exercise 3.23. Show $\text{Var}\left[\hat{Y}|X\right] = \sigma^2 H$.

Exercise 3.24. Check for multicolinearity in our past examples.

Exercise 3.25. Complete the problem sets from Chapter's 2, 3 and 4!

4 Residual analysis

Recall that we want to study the normal MLR:

$$Y|X = X\beta + \epsilon,$$

where - $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$. - $\beta \in \mathbb{R}^{p \times 1}$ is the unknown, population coefficient vector - $X \in \mathbb{R}^{n \times p}$ is a covariate matrix

We assume that: - The relationship is linear $Y|X = X\beta + \epsilon$, - $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. - $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.

We have seen some methods for checking if these are appropriate, we will dive deeper now.

Recall that the residuals are defined as:

$$\hat{\epsilon} = Y - \hat{Y} = Y - X\hat{\beta}.$$

Given that a residual may be viewed as the deviation between the data and the fit, it is also a measure of the variability in the response variable not explained by the regression model. It is also convenient to think of the residuals as the realized or observed values of the model errors. Thus, it's reasonable to conclude that departures from the assumptions on the errors should show up in the residuals. Analysis of the residuals is an effective way to discover several types of model inadequacies.

4.1 Properties of residuals

The following are some properties of the residual vector. First, the sample mean of the residuals is zero: $\sum_{i=1}^n \hat{\epsilon}_i / n = \hat{\epsilon} 1_n \cdot 1/n = 0$. We also have that $\mathrm{E}[\hat{\epsilon}] = 0$. Next, the sample variance of the residual vector is approximately the MSE: $\frac{1}{n-1}\sum_{i=1}^n \hat{\epsilon}_i^2 = \frac{n-p}{n-1} MSE$. Lastly, unlike the random error ϵ_i, the residuals **are not** independent. Sometimes we say that they are "approximately independent" if $p << n$, which we will touch on later.

4.2 Types of residuals

We will refer to $\hat{\epsilon}_i$ as simply the residuals, or ordinary residuals when we need to be extra clear.

The standardized residual is given by

$$d_i = \hat{\epsilon}_i/\sqrt{MSE}.$$

This is an approximate Z-score for the residuals, since the residuals have 0 mean, the MSE is approximately the variance of the random error and the residuals approximate the random error. We say that large d_i (> 3) indicates an outlier, though, we may want to use a more robust measure of the variance. We will generally prefer to use a different type of residual, which we now present.

We now introduce the hat matrix: $H = X(X^\top X)^{-1}X^\top$. Note that H is symmetric and idempotent. The hat matrix appears often in regression analysis, and you should remember this quantity. It is called the hat matrix because $\hat{Y} = HY$.

Note that the eigenvalues of H, and any idempotent matrix A are either 0 or 1:

$$\lambda x = Ax = A^2 x = A\lambda x = \lambda^2 x,$$

which implies that $\lambda \in \{0, 1\}$.

Exercise 4.1. Verify that H is symmetric and idempotent and that $\hat{Y} = HY$, where one recalls that a matrix A is idempotent if $AA = A$.

Now, note that:

$$\hat{\epsilon} = (I - H)Y = (I - H)\epsilon.$$

Exercise 4.2. Verify that $\hat{\epsilon} = (I - H)Y = (I - H)\epsilon.$.

Using this identity, we have that $\text{Var}\left[\hat{\epsilon}\right] = (I - H)\epsilon(I - H)^\top = (I - H)\sigma^2$.

The fact that H is symmetric and idempotent implies that its diagonal elements are between 0 and 1. It follows that the elements on the diagonal of $(I - H)$ are also between 0 and 1. Therefore, the MSE overestimates the variance of the residuals: the variance of residual i is given by $(1 - h_{ii})\sigma^2 < \sigma^2 \approx MSE$. Here, h_{ii} denotes the ith diagonal element of the matrix H.

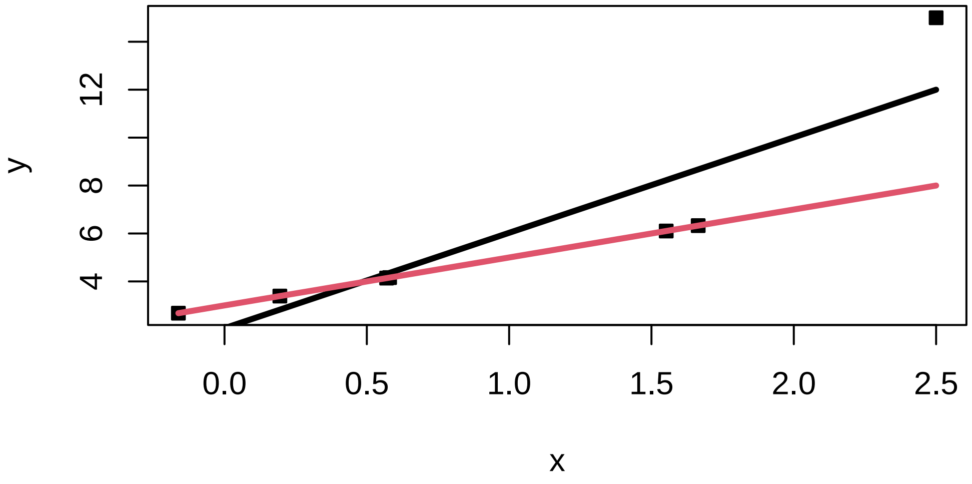
Furthermore, h_{ii} is a measure of the location of the ith point in x-space, $\text{Var}\left[\hat{\epsilon}_i\right]$ depends on where X_i lies. Points near the center of the x-space have larger variance than residuals at more remote locations. What do you think about this?

Now, intuitively, violations of model assumptions would be more likely to occur at remote points. However, the variance of the ordinary residuals is lower at these points. Therefore, these violations may be hard to detect from inspection of the ordinary residuals because their residuals will often be smaller.

Therefore, we will call points that are outlying in the x-space **leverage points**. We will refer to **influence points** as points that are not only remote in terms of the x-space, but also the observed response for that point is not consistent with the response that would be predicted for that point, using only the other data points.

In the example below, observe that the right-most point is both a leverage and influential point.

```
##########
# Simulate some data
set.seed(330)
x=c(rnorm(6),2.5)
y=x*2+3
y[7]=y[7]+7

# Plot the data and fitted lines
plot(x,y,pch=22,bg=1)
a=lm(y~x)
curve(a$coefficients[1]+x*a$coefficients[2],add=T,lwd=3)
curve(x*2+3,add=T,col=2,lwd=3)
```



```
a$coefficients
```

```
(Intercept)           x 
   2.048937    3.979977 
```

Let $\hat{Y}_n^*$ be the estimate of Y_n based on the other data and let $\delta_n = Y_n - \hat{Y}_n^*$. Note that one can show that $\hat{Y}_n = \hat{Y}_n^* + h_{nn}\delta_n$. Next, we know that if X_n is outlying, i.e., $||X_n||$ is large, then $h_{nn} \approx 1$. This implies that $\hat{Y}_n \approx Y_n$, which means that the regression line is dragged to pass through (X_n, Y_n).

To detect these types of outlying points, it makes sense to then define the **studentized residuals**:

$$r_i = \frac{\hat{\epsilon}_i}{\sqrt{MSE(1-h_{ii})}}.$$

The studentized residuals in the simple linear regression model reduce to

$$r_i = \frac{\hat{\epsilon}_i}{\sqrt{MSE}\left[1 - \left(\frac{1}{n} + \frac{\left(X_i - \bar{X}\right)^2}{\sum(X-\bar{X})^2}\right)\right]}.$$

Observe that as X_i grows large, we have that $\frac{\left(X_i - \bar{X}\right)^2}{\sum(X-\bar{X})^2} \to 1$, which implies that $\left[1 - \left(\frac{1}{n} + \frac{\left(X_i - \bar{X}\right)^2}{\sum(X-\bar{X})^2}\right)\right] \to 0$ and $r_i \to \infty$. On the other hand, as $\hat{\epsilon}_i$ grows large, we have that r_i grows large. Therefore, the studentized residual will be large for observations with large ordinary residuals, and for leverage observations.

Earlier, we presented δ_i, the difference between the response of the ith observation and the predicted response based on the observations with the ith points removed. These are known as the **PRESS residuals**. This seems hard computationally, but one can show that

$$\delta_i = \frac{\hat{\epsilon}_i}{1 - h_{ii}}.$$

Note that when h_{ii} is large, this indicates a highly influential point. Observe that a large PRESS residual δ_i, but small ordinary residual $\hat{\epsilon}_i$, indicates that the model fit without (X_i, Y_i) predicts Y_i poorly.

Exercise 4.3. Show that standardizing the PRESS residual, that is, dividing the PRESS residual by its standard deviation, results in $\hat{\epsilon}_i / \sqrt{\sigma^2(1 - h_{ii})}$. Compare this to the studentized residual.

Lastly, if we believe that (X_i, Y_i) is outlying, then we can also leave (X_i, Y_i) out in the MSE calculation. This results in the **R-studentized residuals**:

$$\tilde{r}_i = \frac{\hat{\epsilon}_i}{\sqrt{\widetilde{MSE}_i(1 - h_{ii})}},$$

where $\widetilde{MSE}_i$ is the mean squared error computed from the regression model with (X_i, Y_i) excluded:

$$\widetilde{MSE}_i = \frac{(n - p + 1)MSE - \hat{\epsilon}_i^2 / (1 - h_{ii})}{n - p}.$$

4.3 Revisiting checking model assumptions

Recall from Checking model assumptions that we plot the residuals to check various assumptions. In this case, we can now use our upgraded residuals to make these plots. In general, any of the residuals that incorporate the values h_{ii} are acceptable. We will generally use the studentized residuals.

Recall that we may want to plot:

- QQplot of the studentized residuals
- Histogram of the studentized residuals
- Plot of studentized residuals against the fitted Values
- Studentized residuals against the covariates
- Studentized residuals against covariates that are not currently in the model

- Studentized residuals against time in some contexts

Example 4.1. Here, this data contains delivery times, the number of products in the delivery and the distance of the delivery. Perform a residual analysis on the model which regresses delivery times against the number of products in the delivery and the distance of the delivery. Compute all the different types of residuals.

```
########## Delivery Time



# Load and inspect the data
data(delivery, package="robustbase")
df=delivery
n=nrow(df)
head(df)
```

```
  n.prod distance delTime
1      7      560   16.68
2      3      220   11.50
3      3      340   12.03
4      4       80   14.88
5      6      150   13.75
6      7      330   18.11
```

```
# Fit the model
model=lm(delTime~.,data=df)
s=summary(model); s
```

```


Call:
lm(formula = delTime ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7880 -0.6629  0.4364  1.1566  7.4197 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.341231   1.096730   2.135 0.044170 *  
n.prod      1.615907   0.170735   9.464 3.25e-09 ***
distance    0.014385   0.003613   3.981 0.000631 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.259 on 22 degrees of freedom
Multiple R-squared:  0.9596,	Adjusted R-squared:  0.9559 
F-statistic: 261.2 on 2 and 22 DF,  p-value: 4.687e-16
```

```
# X matrix
X=model.matrix(model)

# Hat matrix
hat=X%*%solve(t(X)%*%X)%*%t(X)

# Compute h_ii
hii=diag(hat)
hii
```

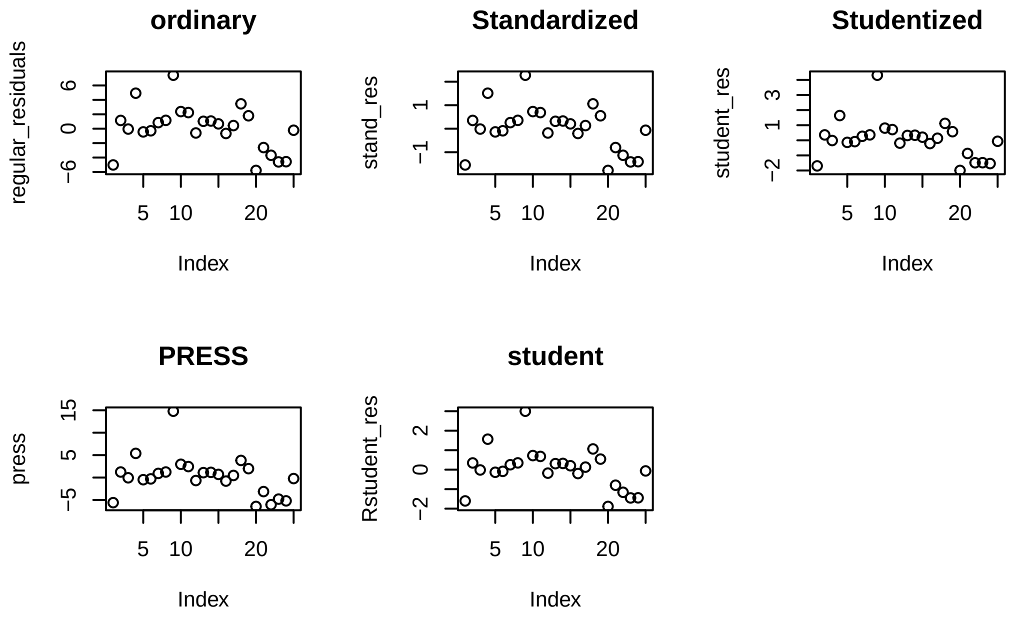
```
         1          2          3          4          5          6          7 
0.10180178 0.07070164 0.09873476 0.08537479 0.07501050 0.04286693 0.08179867 
         8          9         10         11         12         13         14 
0.06372559 0.49829216 0.19629595 0.08613260 0.11365570 0.06112463 0.07824332 
        15         16         17         18         19         20         21 
0.04111077 0.16594043 0.05943202 0.09626046 0.09644857 0.10168486 0.16527689 
        22         23         24         25 
0.39157522 0.04126005 0.12060826 0.06664345 
```

```
max(hii)
```

```
[1] 0.4982922
```

```
# Notice 9 is large

###### ordinary residuals
regular_residuals=model$residuals
# or


# standardized residuals
stand_res=model$residuals/s$sigma

# studentized residuals
```

```
student_res=rstudent(model)


#PRESS residuals
press=model$residuals/(1-hii)

# Get the MSE_is
MSE_i=((n-2)*(s$sigma)^2-regular_residuals^2/(1-hii))/(n-3)

#r studentized residuals
Rstudent_res=model$residuals/sqrt(MSE_i)

# Plot them all and compare
par(mfrow=c(2,3))
plot(regular_residuals,main="ordinary")
plot(stand_res,main="Standardized")
plot(student_res,main="Studentized")
plot(press,main="PRESS")
plot(Rstudent_res,main="student")

# Notice 9 is much more outlying in the last 3 graphs.

# Reset plotting
par(mfrow=c(1,1))
```



```
# 9 is largest
which.max(student_res)
```

```
9
9
```

```
# Notice the standardized is half as large as the studentized.
student_res[9]
```

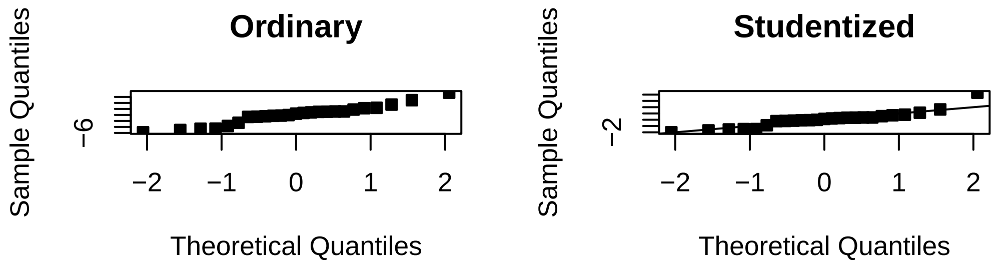
```
      9
4.31078
```

```
stand_res[9]
```

```
       9
2.276351
```

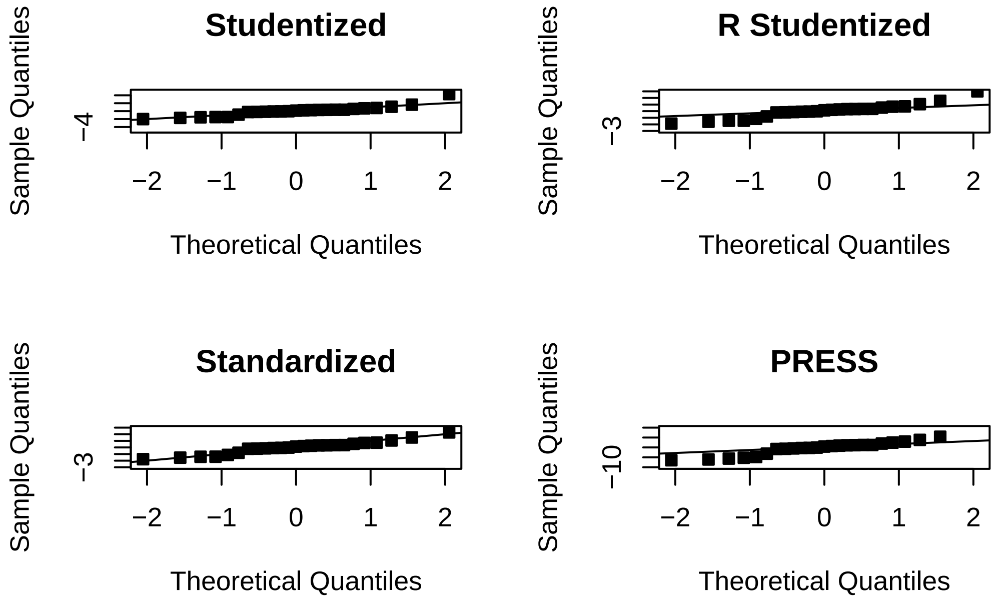
```
par(mfrow=c(2,2))

# Notice the difference !!!
```

```
qqnorm(regular_residuals,pch=22,bg=1,main="Ordinary")

qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)

# Compare all

par(mfrow=c(2,2))
```

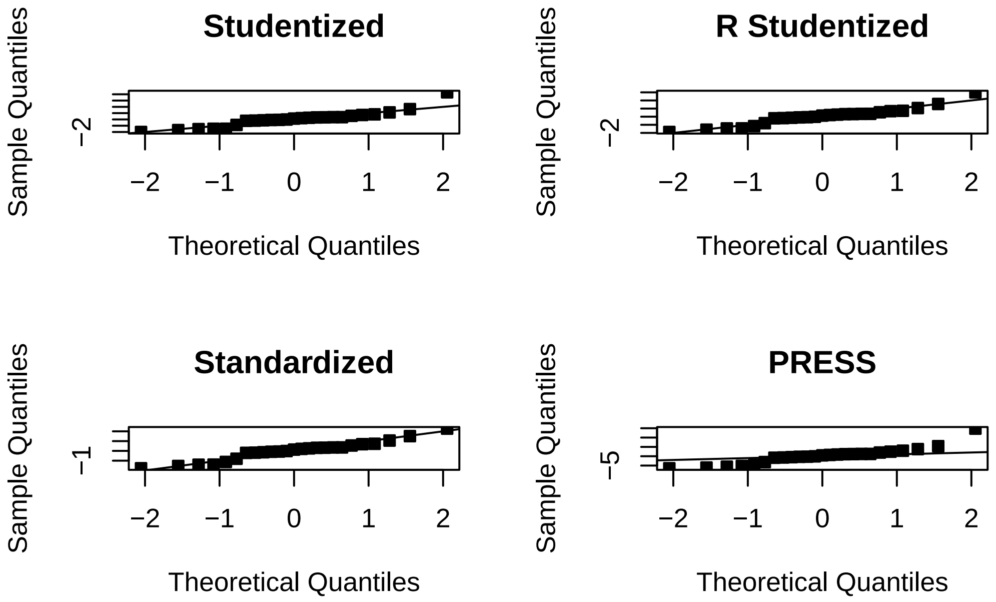


```
qqnorm(student_res,pch=22,bg=1,ylim=c(-5,5),main="Studentized")
abline(0,1)
# hist(student_res)

qqnorm(Rstudent_res,pch=22,bg=1,ylim=c(-3,3),main="R Studentized")
qqline(Rstudent_res,pch=22,bg=1,ylim=c(-10,10))
# abline(0,1)

qqnorm(stand_res,pch=22,bg=1,ylim=c(-3,3),main="Standardized")
abline(0,1)

qqnorm(press,pch=22,bg=1,ylim=c(-10,10),main="PRESS")
qqline(press,pch=22,bg=1,ylim=c(-10,10))
```

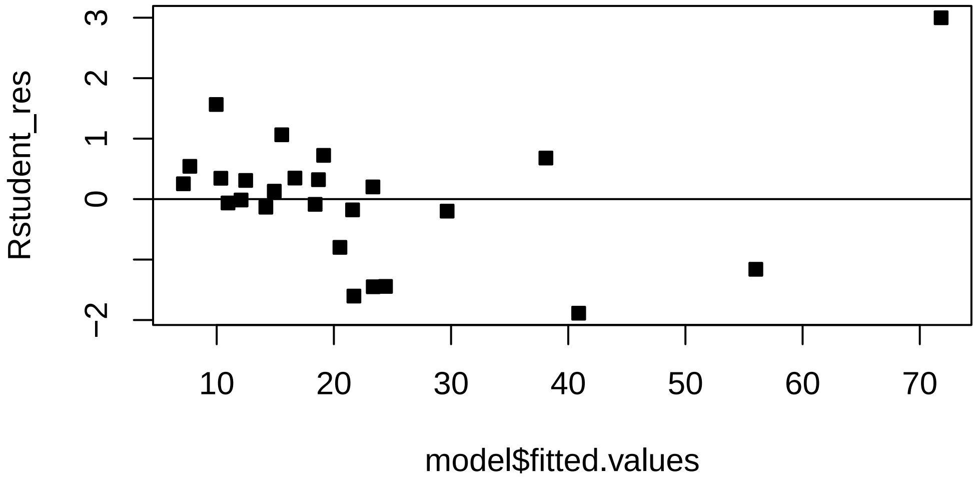
```
#careful of the scale!

par(mfrow=c(3,2))
qqline(model$residuals,pch=22,bg=1,main="Ordinary")
```



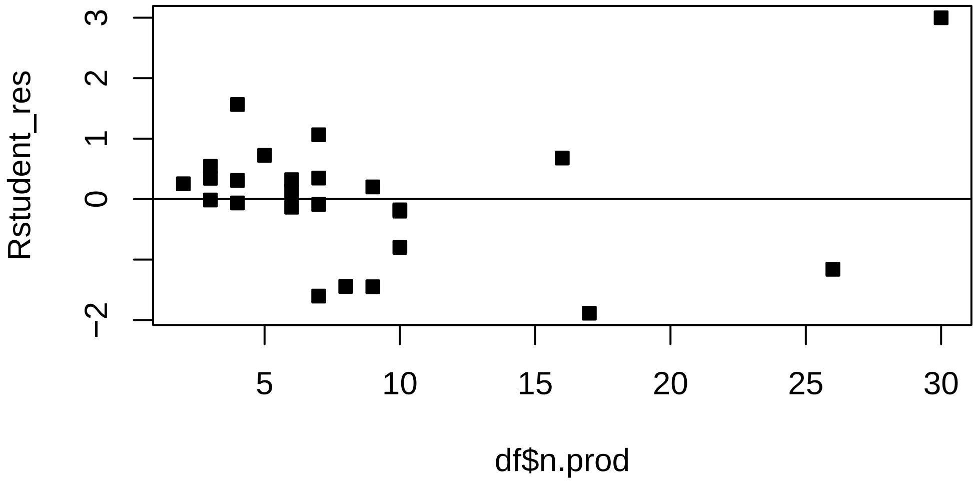
```
par(mfrow=c(2,2))
qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)
qqnorm(Rstudent_res,pch=22,bg=1,main="R Studentized")
abline(0,1)
qqnorm(stand_res,pch=22,bg=1,main="Standardized")
abline(0,1)
qqnorm(press,pch=22,bg=1,main="PRESS")
abline(0,1)
```



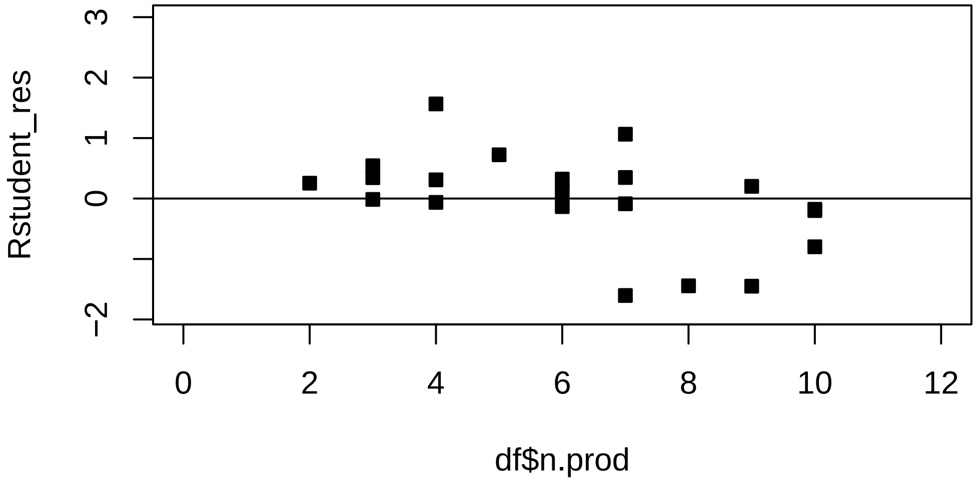
```
# Now we plot the fitted values against the R studentized residuals
par(mfrow=c(1,1),pch=22)
plot(model$fitted.values,Rstudent_res,bg=1)
abline(h=0)
```



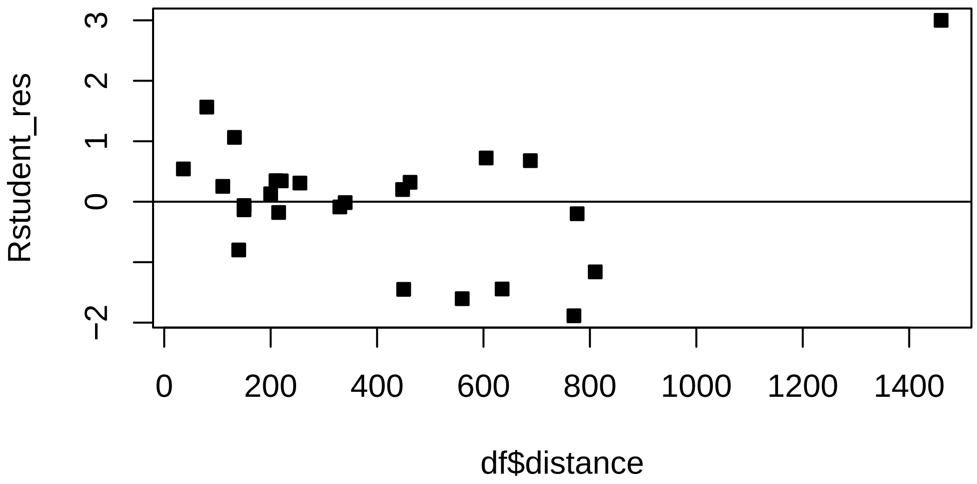
```
# Now we plot the number of products against the R studentized residuals
# There is one moderately large delivery!
plot(df$n.prod,Rstudent_res,bg=1)
abline(h=0)
```



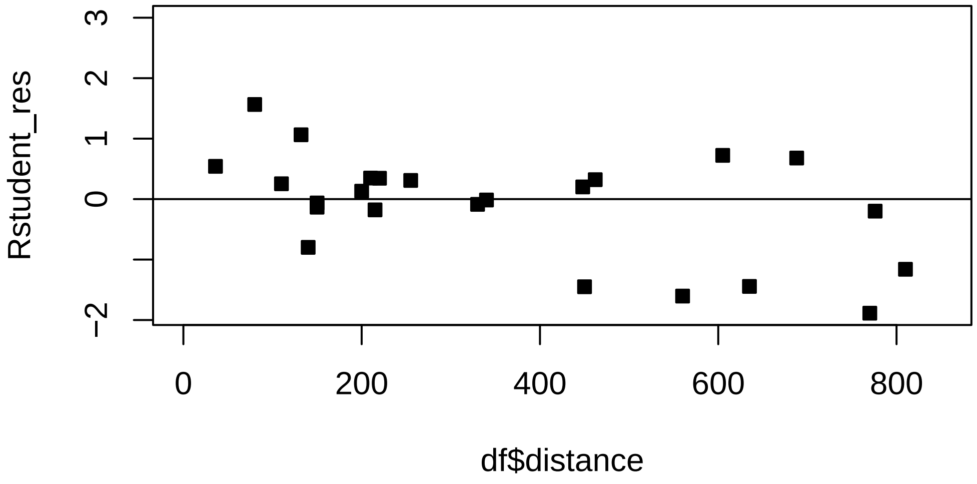
```
# Care for the scale
plot(df$n.prod,Rstudent_res,bg=1,xlim=c(0,12))
abline(h=0)
```



```
# There is one very far delivery!
plot(df$distance,Rstudent_res,bg=1)
abline(h=0)
```



```
# Care for the scale
plot(df$distance,Rstudent_res,bg=1,xlim=c(0,850))
abline(h=0)
```



```
# What happens to the model when we remove this outlying observation (the far distance delive
df2=df[-which.max(df$distance),]


# refit the model
model=lm(delTime~.,data=df2)
s=summary(model); s
```

```
Call:
lm(formula = delTime ~ ., data = df2)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0325 -1.2331  0.0199  1.4730  4.8167 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.447238   0.952469   4.669 0.000131 ***
n.prod      1.497691   0.130207  11.502 1.58e-10 ***
distance    0.010324   0.002854   3.618 0.001614 ** 
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.43 on 21 degrees of freedom
Multiple R-squared:  0.9487,    Adjusted R-squared:  0.9438
F-statistic: 194.2 on 2 and 21 DF,  p-value: 2.859e-14
```

```
# X matrix
X=model.matrix(model)

# Hat matrix
hat=X%*%solve(t(X)%*%X)%*%t(X)

# Compute h_ii
hii=diag(hat)
hii
```

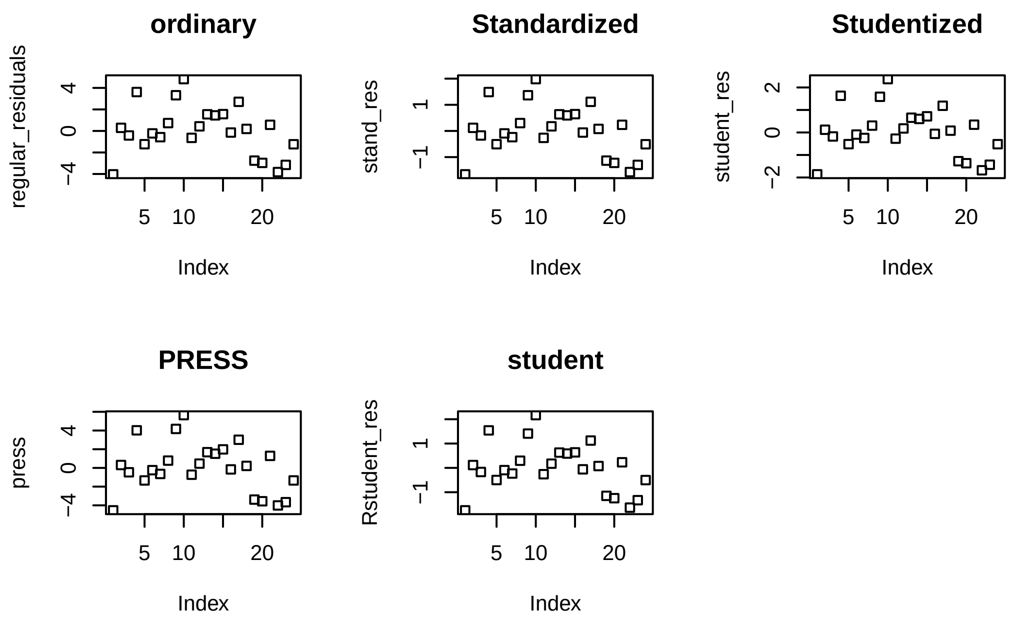
```
         1          2          3          4          5          6          7
0.11083391 0.07741039 0.09998709 0.10097319 0.08066357 0.04290146 0.10024969
         8         10         11         12         13         14         15
0.06537738 0.20438000 0.14675966 0.11367920 0.06437975 0.08033747 0.04661503
        16         17         18         19         20         21         22
0.21115081 0.06254612 0.10128434 0.11992977 0.18537865 0.16642759 0.55671434
        23         24         25
0.04687996 0.13894064 0.07620000
```

```
max(hii)
```

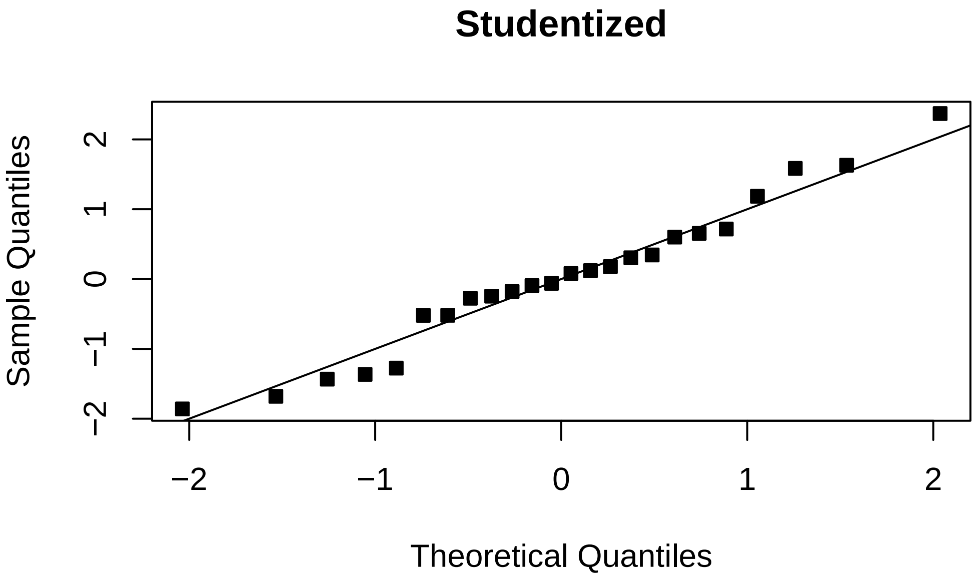
```
[1] 0.5567143
```

```
###### ordinary residuals
regular_residuals=model$residuals
# or


# standardized residuals
stand_res=model$residuals/s$sigma

# studentized residuals
student_res=rstudent(model)
```

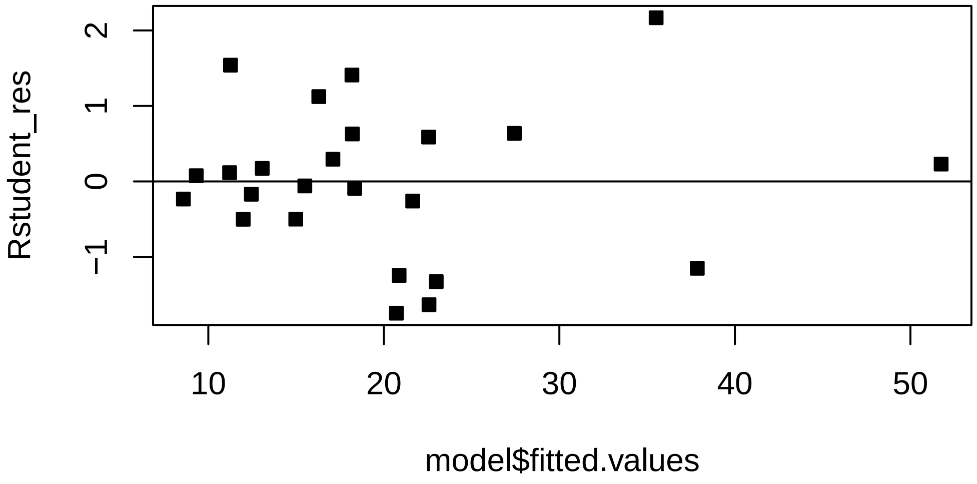
```
#PRESS residuals
press=model$residuals/(1-hii)

# Get the MSE_is
MSE_i=((n-2)*(s$sigma)^2-regular_residuals^2/(1-hii))/(n-3)

#r studentized residuals
Rstudent_res=model$residuals/sqrt(MSE_i)

# Plot them all and compare - much better
par(mfrow=c(2,3))
plot(regular_residuals,main="ordinary")
plot(stand_res,main="Standardized")
plot(student_res,main="Studentized")
plot(press,main="PRESS")
plot(Rstudent_res,main="student")


# Notice that these graphs are fine now...
par(mfrow=c(1,1))
```



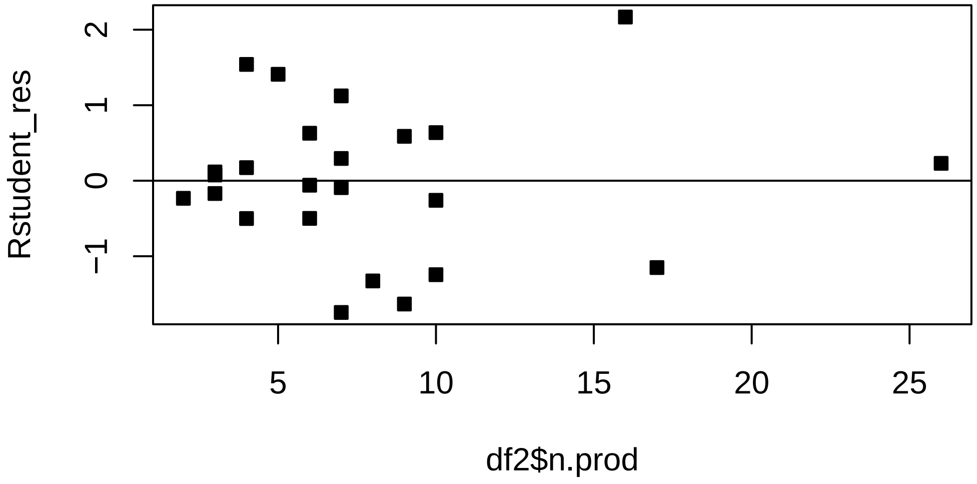
```
qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)
```



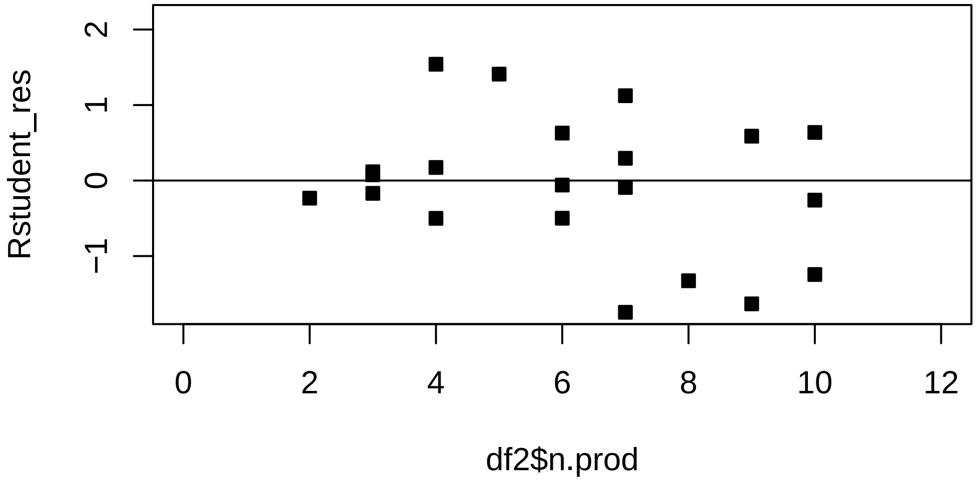
```
# Now we plot the fitted values against the R studentized residuals
par(mfrow=c(1,1),pch=22)
plot(model$fitted.values,Rstudent_res,bg=1)
abline(h=0)
```



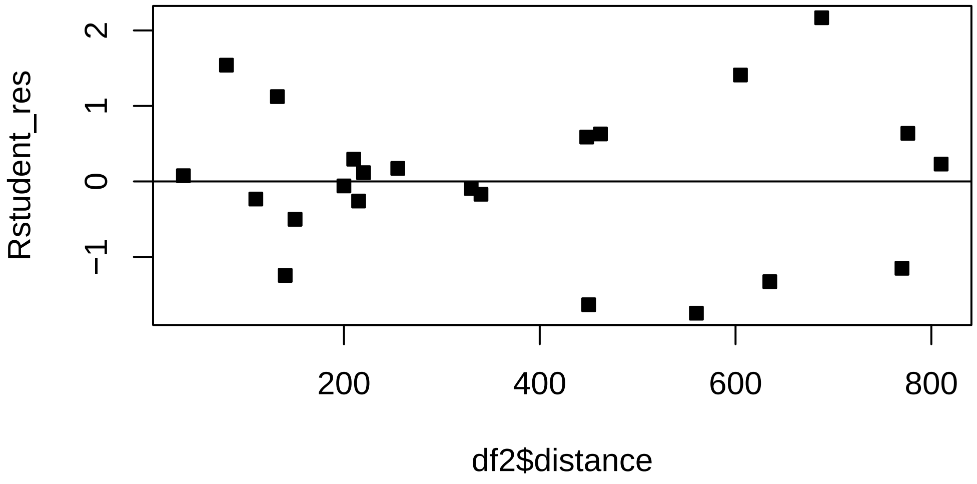
```
# Now we plot the number of products against the R studentized residuals
plot(df2$n.prod,Rstudent_res,bg=1)
abline(h=0)
```



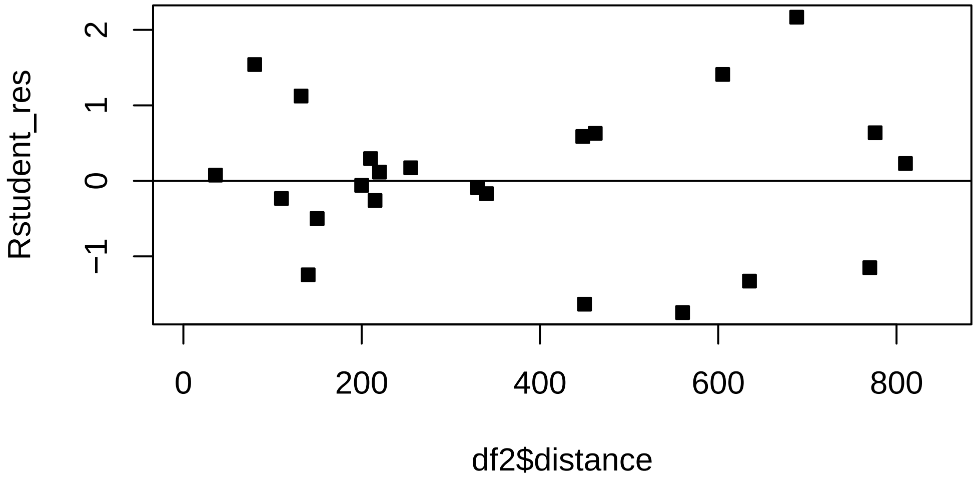
```
# Care for the scale
plot(df2$n.prod,Rstudent_res,bg=1,xlim=c(0,12))
abline(h=0)
```



```
plot(df2$distance,Rstudent_res,bg=1)
abline(h=0)
```



```
# Care for the scale
plot(df2$distance,Rstudent_res,bg=1,xlim=c(0,850))
abline(h=0)
```



Now, we have introduced different types of residuals and the appropriate graphs to examine when checking for violations of the assumptions. When we observe violations of the assumptions - what do we do? That will be the topic of the next section.

Some of these remedies include: - Transformations of the response - Transformations of certain regressors - Robust methods/outlier removal - Inclusion of new regressors

4.4 Homework stop

Do the Chapter 4 questions from the textbook.

Exercise 4.4. In the context of a regression model, do you think a point outlying in the x-space is more problematic than a point outlying in the y-space?

Exercise 4.5. Make a table describing the differences between each type of residual.

Exercise 4.6. Perform a residual analysis on the marketing data from Example Example 3.6.

Exercise 4.7. Perform a residual analysis on the data from Example Example 3.7.

5 Transformations

5.1 Variance-stabilizing transformations

Recall that we assume that $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. A common reason for a violation of this assumption is for Y to have a distribution in which the variance is related to its mean. For example, if the response Y is a Poisson random variable, i.e.,

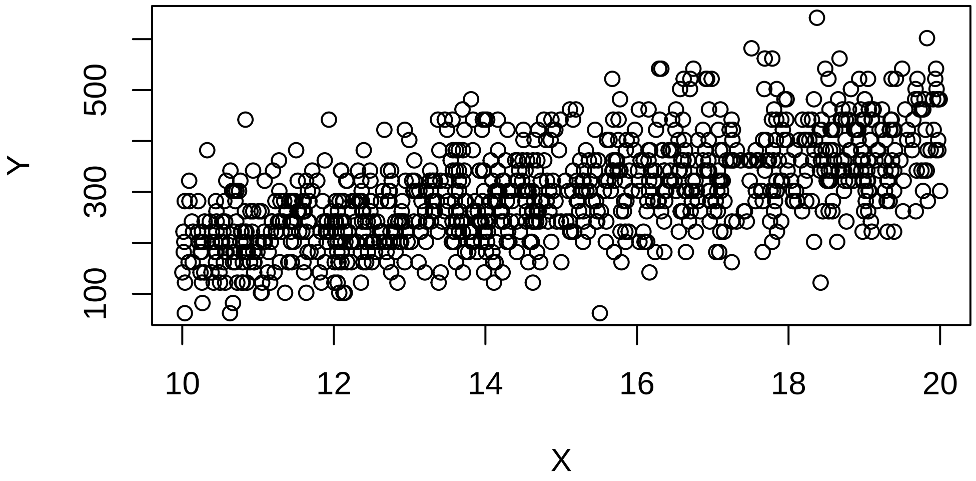
$$Y|X \sim Pois(X\beta),$$

then we have that $\mathrm{E}[Y] = X\beta$, then $\mathrm{E}[Y] = \mathrm{Var}[Y] = X\beta$. In this case, the simple linear regression assumptions are violated. In particular, the variance is not the same for each observation. Here, it happens that taking the response to be roughly $\sqrt{Y}$ fixes the problem. That is, performing the regression analysis with $\sqrt{Y}$ as the response variable instead of Y, ensures that the regression assumptions are (approximately) satisfied. This example gives rise to the idea of **transformations**. If our data do not satisfy the assumptions for the MLR or the normal MLR, we might ask if there is some transformation of either the response, some of the covariates, or both that make the data suitable for a MLR analysis. Note that the assumptions are important. For instance, if the variance is not homogeneous, the OLS estimator will still be unbiased, but they will no longer have BLUE property. That means that some other estimator will work better for such data!

Which transformation should we choose? Sometimes, we can use prior experience or theoretical considerations to guide us in selecting an appropriate transformation. Other times, we must choose it empirically, i.e., based on the data. Often, the square root and the logarithm are popular choices. If you response is between 0 and 1, and the data appear to be "football shaped", then you may like to take the $\arcsin(\sqrt{Y})$.

We now demonstrate what one of these relationships looks like in simple linear regression. We now simulate a dataset where $\sigma^2 \propto \mathrm{E}[Y|X]$, and plot X against Y. We use the Poisson example discussed previously.

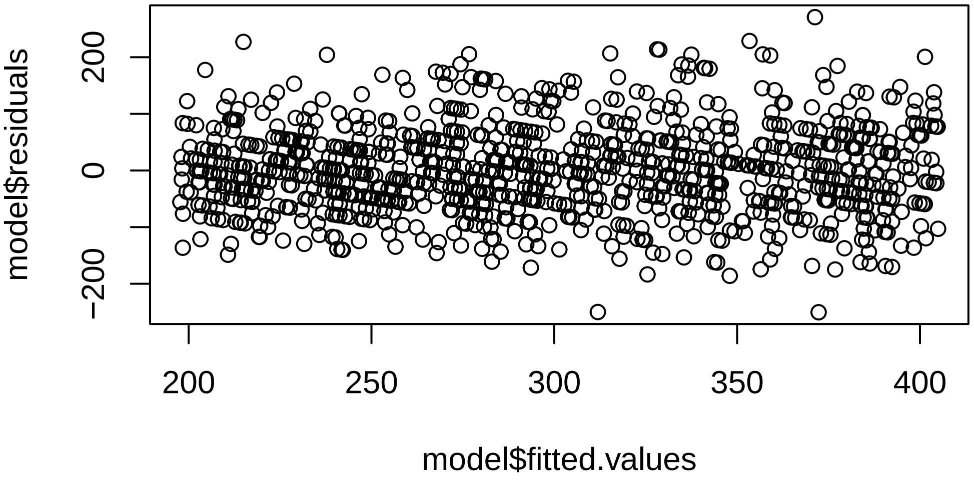
```
set.seed(2352)
n=1000
X=runif(n,5,10)*2
Y=20*rpois(n,X)+2
plot(X,Y)
```



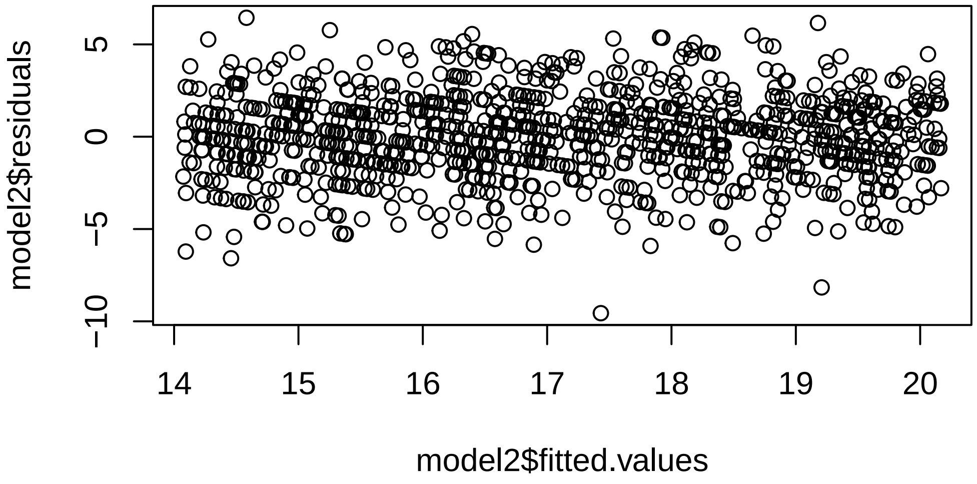
```
# Performing a regression analysis yields:
model=lm(Y~X)
# Notice the intercept is poorly estimated!
summary(model)
```

```
Call:
lm(formula = Y ~ X)

Residuals:
     Min       1Q   Median       3Q      Max
-250.289  -53.392   -2.127   48.795  270.716

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -9.5199    13.0735  -0.728    0.467
X            20.7241     0.8599  24.101   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 77.61 on 998 degrees of freedom
Multiple R-squared:  0.3679,    Adjusted R-squared:  0.3673
F-statistic: 580.9 on 1 and 998 DF,  p-value: < 2.2e-16
```

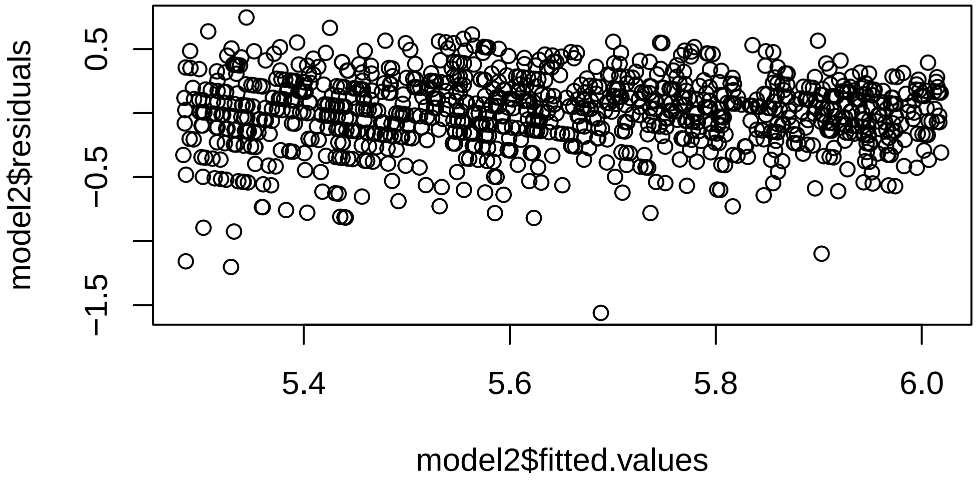
```
# Notice the fan shape in the residuals against the fitted values?
plot(model$fitted.values,model$residuals)
```



```
# Let's perform the transformations

model2=lm(sqrt(Y)~X)

plot(model2$fitted.values,model2$residuals)
```

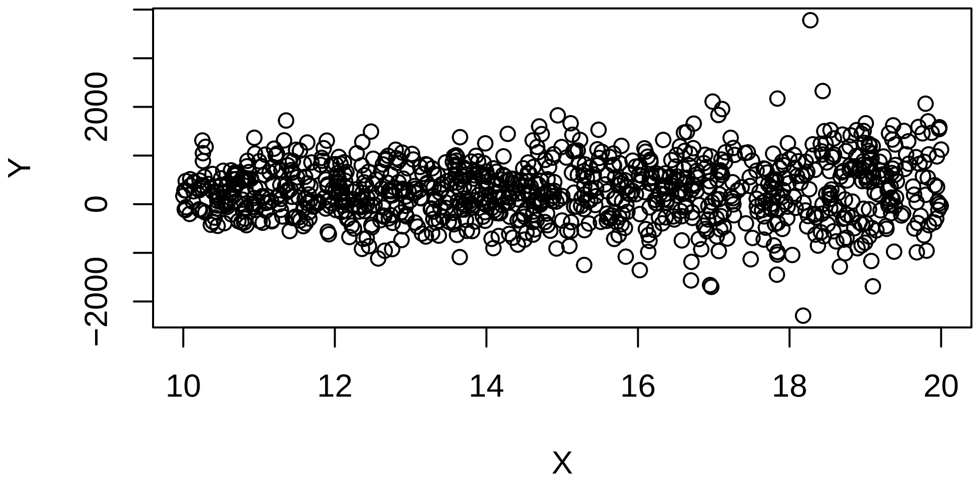


```
model2=lm(log(Y)~X)

plot(model2$fitted.values,model2$residuals)
```



However, these transformations do not always work. Suppose we have that $Y \sim \mathcal{N}(X, 4 * X^2)$. We then have that $\sigma = 2{*}X = 2{*}\mathrm{E}\left[Y|X\right]$. Notice how the spread of the points is increasing with X? This is a symptom of non-homogeneous variance. However, the proposed transformations do not work.

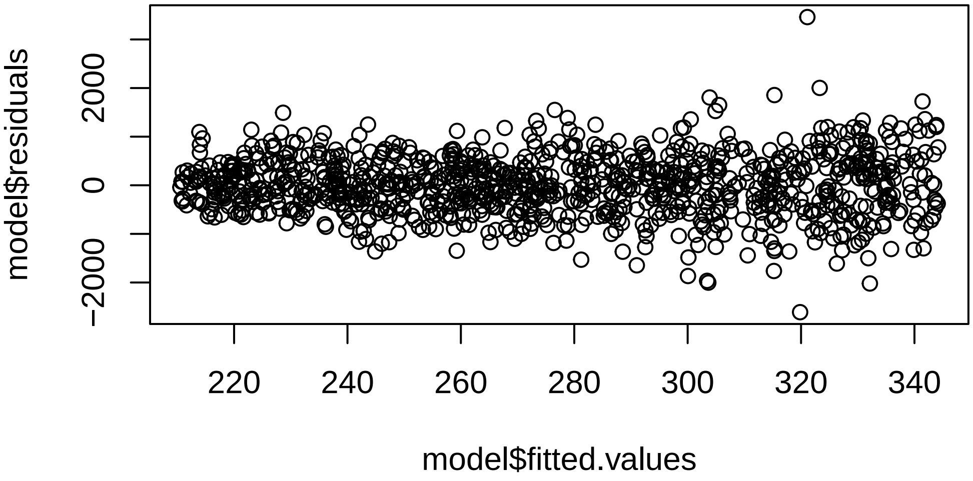
```
set.seed(2352)
# \sigma^2\propto \E{Y}
Y=20*rnorm(n,X,X*2)+2

plot(X,Y)
```



```
# Performing a regression analysis yields:
model=lm(Y~X)

# notice the intercept is poorly estimated.
summary(model)
```

```
Call:
lm(formula = Y ~ X)

Residuals:
    Min      1Q  Median      3Q     Max
-2610.8  -375.3    -3.5   384.8  3460.5

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   76.844    103.345   0.744   0.4573
X             13.367      6.797   1.966   0.0495 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 613.5 on 998 degrees of freedom
Multiple R-squared:  0.00386,   Adjusted R-squared:  0.002861
F-statistic: 3.867 on 1 and 998 DF,  p-value: 0.04953
```

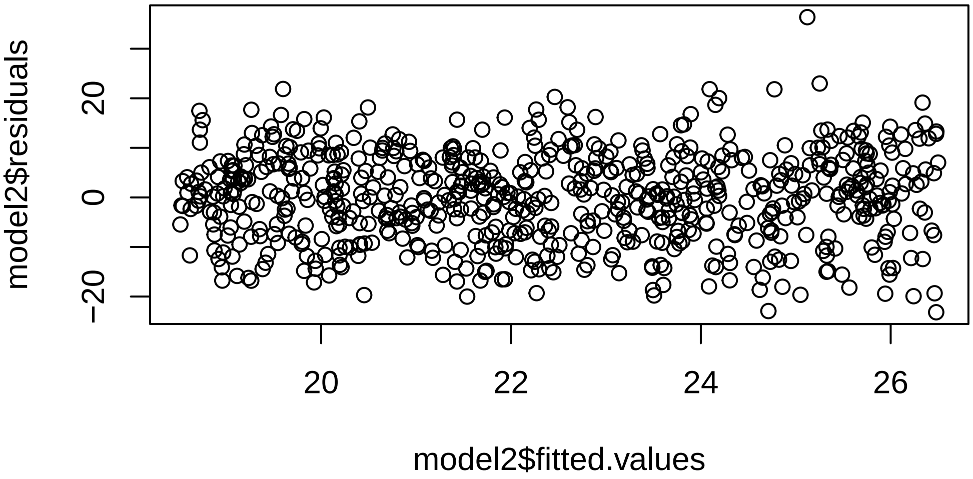
```
# Notice the fan shape in the residuals against the fitted values?
plot(model$fitted.values,model$residuals)
```



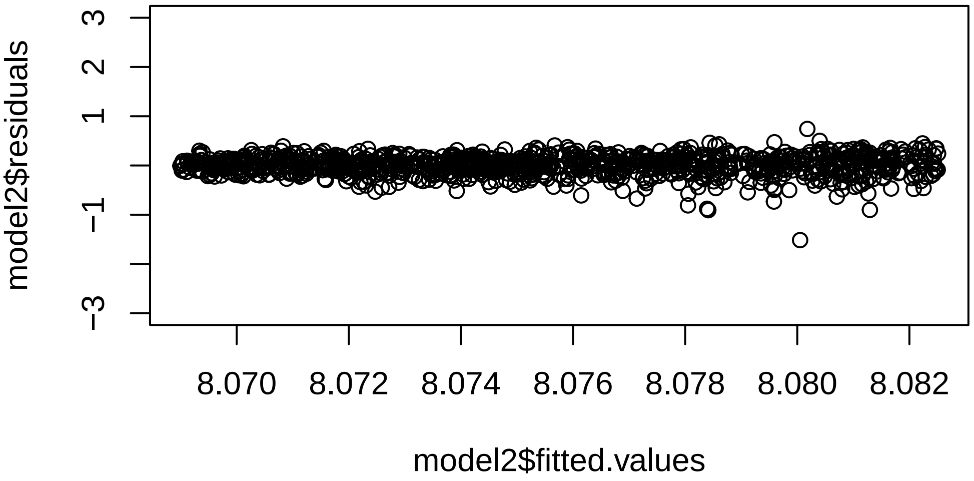
```
# Let's perform the transformation

# Performing a regression analysis yields:
model2=lm(sqrt(Y)~X)
```

```
Warning in sqrt(Y): NaNs produced
```

```
# Notice the fan shape in the residuals against the fitted values?
plot(model2$fitted.values,model2$residuals)
```



```
# Performing a regression analysis yields:
model2=lm(log(Y+3000)~X)

# Notice the fan shape in the residuals against the fitted values?
plot(model2$fitted.values,model2$residuals,ylim=c(-3,3))
```



In general, a good transformation to correct violated assumptions can improve estimates and test accuracy.

Caution

It is often necessary to convert any predicted values back to the original units. Applying the inverse transformation to predicted values gives an estimate of the median of the distribution of the (untransformed) response – instead of the mean. This implies that predictions are generally biased. Prediction and confidence intervals do not suffer this illness. They can be converted back to the original units via the inverse transformation and the interpretation will remain the same.

Let's expand on this. It is a good time to recall that in general, for a real function f, we have that $\mathrm{E}\left[f(X)\right] \neq f(\mathrm{E}\left[X\right])$. For instance, for many random variables Z, we would have that $\mathrm{E}\left[Z^2\right] \neq \mathrm{E}\left[Z\right]^2$, $\mathrm{E}\left[\log Z\right] \neq \log \mathrm{E}\left[Z\right]$ etc. .

In a transformed regression model, we fit the following model:

$$f(Y) = X\beta + \epsilon.$$

If we are interested in predicting the value of Y given z, then it seems natural to take the predictions for $f(Y)$ given z, which are given by $\beta^\top z$ and apply the inverse transformation f^{-1}. For instance, to predict $Y|Z = z$, we may compute: $f^{-1}(\beta^\top z)$. It turns out, this prediction is biased, and we should use a different method instead.

To see why it's biased, observe that the predictions from the model $f(Y) = X\beta + \epsilon$ for a new set of covariates z are given by $\widehat{f}(Y) = \widehat{\beta}^\top z \approx \mathrm{E}\left[f(Y)|Z = z\right]$. Now, we have that

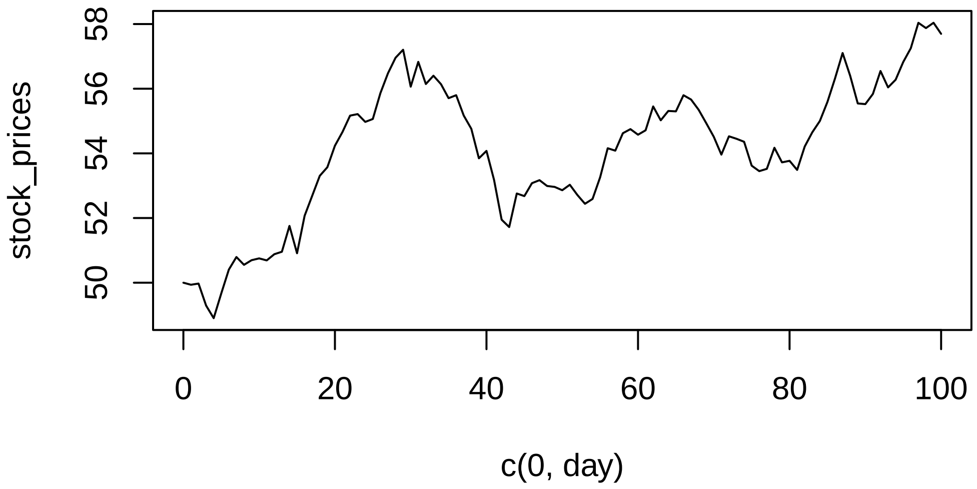
$$f^{-1}(\widehat{\beta}^\top z) \approx f^{-1}(\mathrm{E}\left[f(Y)|Z = z\right]) \neq \mathrm{E}\left[f^{-1}(f(Y))|Z = z\right] = \mathrm{E}\left[\widehat{f}(Y)|Z = z\right].$$

The solution to this problem is to adjust for the bias. For the log transform, we can multiply the resulting inverse transformed predictions by $\exp(\widehat{\sigma}^2/2)$. For the square root transformation, we add $\widehat{\sigma}^2$ to the resulting inverse transformed predictions. See (Miller 1984) for more information.

One can also use confidence and prediction intervals to predict the value of Y given z. Confidence or prediction intervals may be directly converted from one metric to another – such interval estimates are percentiles of a distribution which are unaffected by the transformation. They can be converted back to the original units via the inverse transformation and the interpretation will remain the same. Optimal intervals are intervals with the shortest average interval length for a given confidence level, under a given set of assumptions. However, it may be that the resulting intervals may not be "optimal". One way to get a prediction in the original units, is to apply the inverse transformation to the prediction interval computed from the transformed model and take the midpoint of that interval. This does not always work well - and should be checked against the original data.

Example 5.1. Let's simulate what happens when, given the day $t \in [100]$, we try to estimate the mean stock price P_t for some stock (maybe ?Gamestop?) in a model which regresses the logged rate of return against the day. Note that the logged returns at time t are given by: $L = \log\left(\frac{P_t}{P_{t-1}}\right)$.

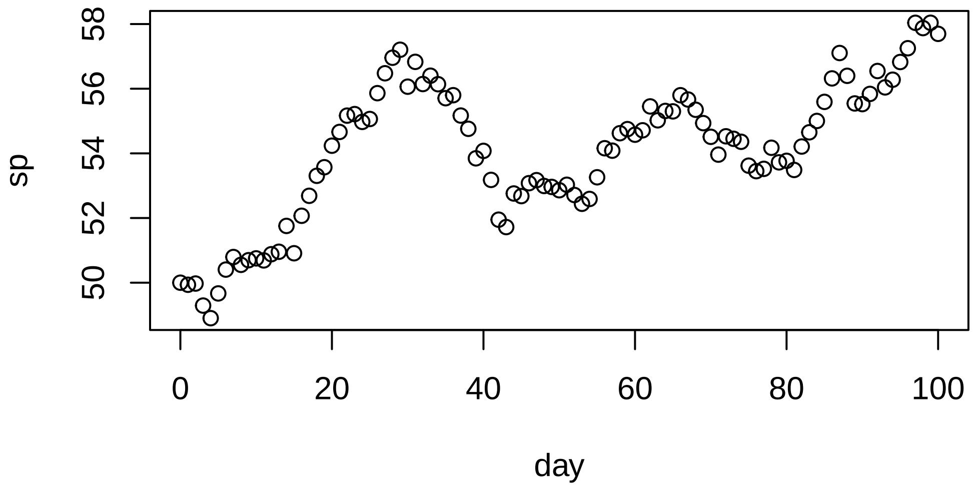
```
set.seed(2352)
# Simulate the stock prices
n=100
day=seq(1:n)
log_return=rnorm(n,0.000001+0.000005*day,0.01)
# log_return=rnorm(n,0.000001+0.000005*day,0.0005)
stock_prices=c(50,50*exp(cumsum(log_return)))
# exp(log_return)[1:10]

plot(c(0,day), stock_prices,type='l')
```



```
df=data.frame(cbind("day"=c(0,day),"sp"=stock_prices))


# Now, suppose this is our starting dataset
head(df)
```

```
  day       sp
1   0 50.00000
2   1 49.93624
3   2 49.97369
4   3 49.29229
5   4 48.90210
6   5 49.66743
```

```
# Notice that the pattern is not great... but we can regress on the transformed response
plot(df)
```



```
# Fitting the model

# Compute the log returns
df$lr=NA
df$lr[2:(n+1)]=log(df$sp[-1]/df$sp[-(n+1)])

# Sanity Check
# log_return[1:5]
# df$lr[2:6]

model=lm(lr~day,df)
summary(model)
```

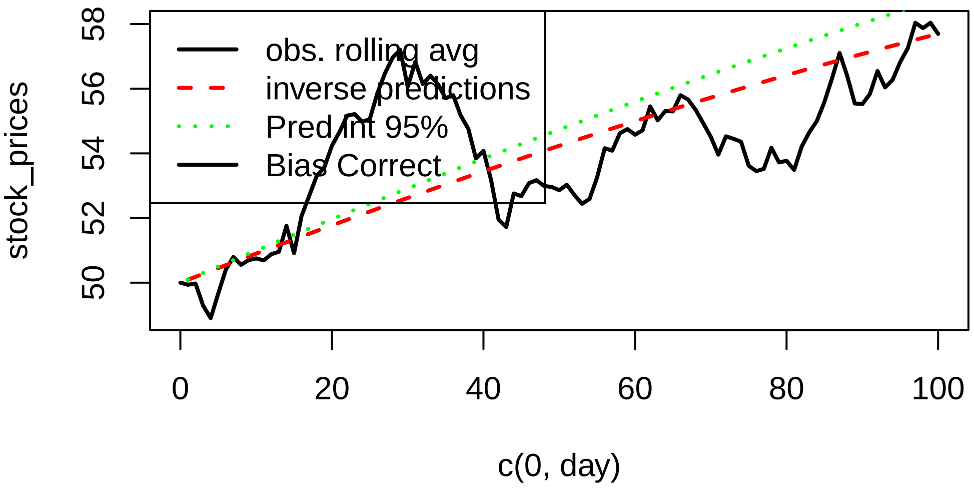
```

Call:
lm(formula = lr ~ day, data = df)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0249054 -0.0064851  0.0000518  0.0075839  0.0208044

Coefficients:
```

```
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.824e-03  1.926e-03   0.947    0.346
day         -7.771e-06  3.312e-05  -0.235    0.815

Residual standard error: 0.00956 on 98 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.0005615, Adjusted R-squared:  -0.009637
F-statistic: 0.05506 on 1 and 98 DF,  p-value: 0.815
```

```
plot(c(0,day), stock_prices,type='l',lwd=2)
lines(50*exp(cumsum(fitted.values(model))),col='red',lty=2,lwd=2)

# Intervals for the mean at each time point
intervals=predict(model,interval ='prediction')[,2:3]
```

```
Warning in predict.lm(model, interval = "prediction"): predictions on current data refer to
```

```
midpoint=ret=rep(0,n)
for(i in 1:n){
  if(i==1){
    midpoint[i]=50*exp(intervals[i,1])/2+50*exp(intervals[i,2])/2
  }
  else
    midpoint[i]=midpoint[i-1]*(exp(intervals[i,1])+exp(intervals[i,2]))/2
}
lines(midpoint,col="green",lty=3,lwd=2)

legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Bias Corr
```

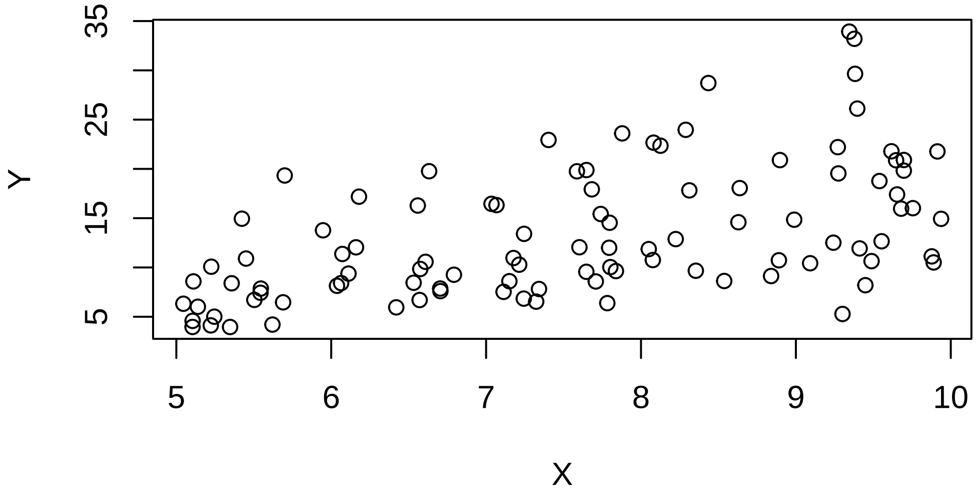


A second example...

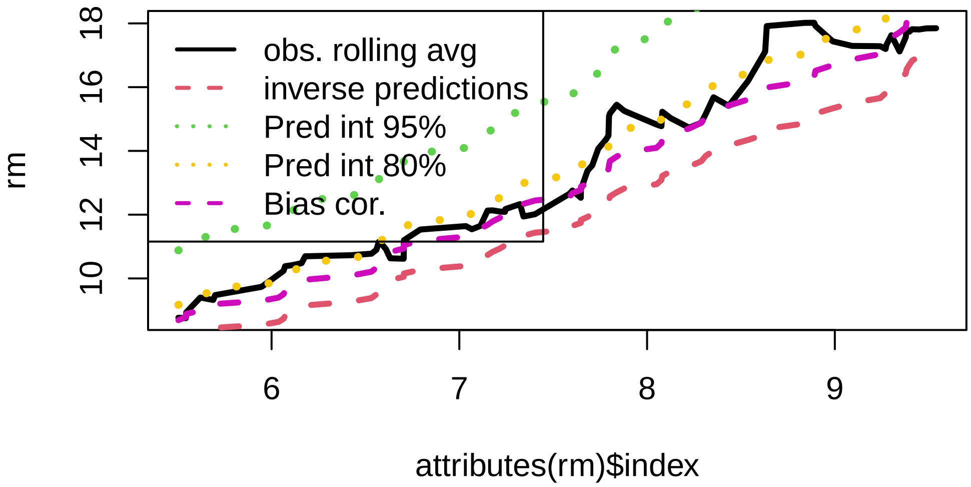
```
set.seed(2352)
# Simulate data
n=100
X=runif(n,5,10)
logs=rnorm(n,1+0.2*X,0.5)
Y=exp(logs)
plot(X,Y)


df=data.frame(cbind("X"=X,"Y"=Y))
df=df[order(X),]

# Fitting the model
model=lm(log(Y)~X,data=df)
summary(model)
```

```
Call:
lm(formula = log(Y) ~ X, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.13484 -0.31721 -0.02877  0.29765  0.85929

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.00060    0.21048   4.754 6.85e-06 ***
X            0.19333    0.02721   7.106 1.94e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4113 on 98 degrees of freedom
Multiple R-squared:  0.3401,    Adjusted R-squared:  0.3333
F-statistic:  50.5 on 1 and 98 DF,  p-value: 1.937e-10
```

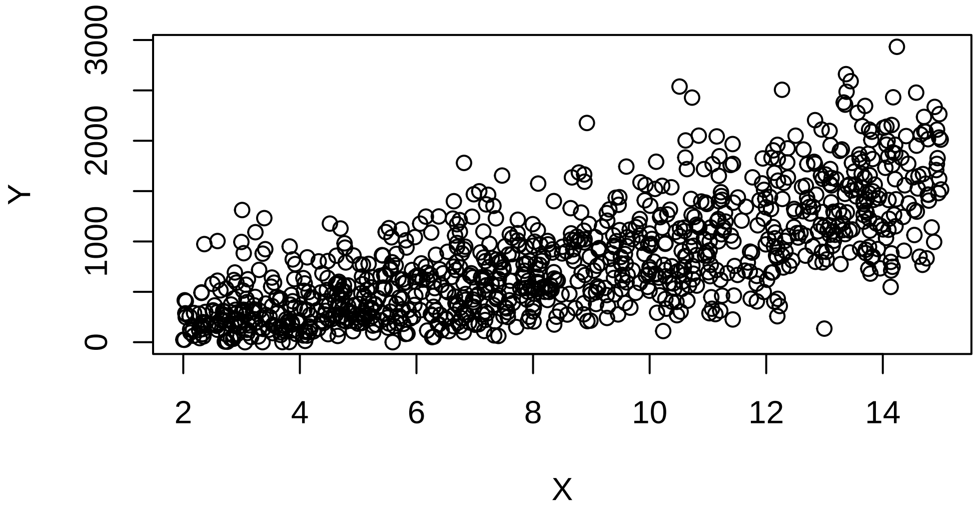
```
s=summary(model)$sigma
# Rolling average




zb=zoo::zoo(x=df$Y,df$X)
```



```
rm=zoo::rollmean(zb,25)

plot(attributes(rm)$index,rm,lty=1,lwd=3,type='l')
# plot(X,Y)

zb=zoo::zoo(x=exp(fitted.values(model)),df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=2,lty=2,lwd=3)

lines(attributes(rm)$index,rm*exp(s^2/2),col=6,lty=2,lwd=3)

# Intervals for the mean at each time point
nd=data.frame("X"=df$X)
ivs=predict(model, newdata = nd,interval = 'prediction')[,2:3]
intervals=rowMeans(exp(ivs))
zb=zoo::zoo(x=intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=3,lty=3,lwd=4)


# Intervals for the mean at each time point - notice when we lower the level the performance
nd=data.frame("X"=df$X)
ivs=predict(model, newdata = nd,interval = 'prediction', level = 0.8)[,2:3]
intervals=rowMeans(exp(ivs))
zb=zoo::zoo(x=intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=7,lty=3,lwd=4)




# Intervals for the mean at each time point using confidence intervals
# ivs=predict(model, newdata = nd,interval = 'confidence', level = 0.8)[,2:3]
# intervals=rowMeans(exp(ivs))
# zb=zoo::zoo(x=intervals,df$X)
# rm=zoo::rollmean(zb,25)
# lines(attributes(rm)$index,rm,col=6,lty=3,lwd=3)

legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Pred int 8
```



```
set.seed(2352)

# Simulate data

n=1000
X=runif(n,2,15)
sq=rnorm(n,10+2*X,7)
Y=sq^2
plot(X,Y)
```



```
df=data.frame(cbind("X"=X,"Y"=Y))
df=df[order(X),]

# Fitting the model
model=lm(sqrt(Y)~X,data=df)
summary(model)
```

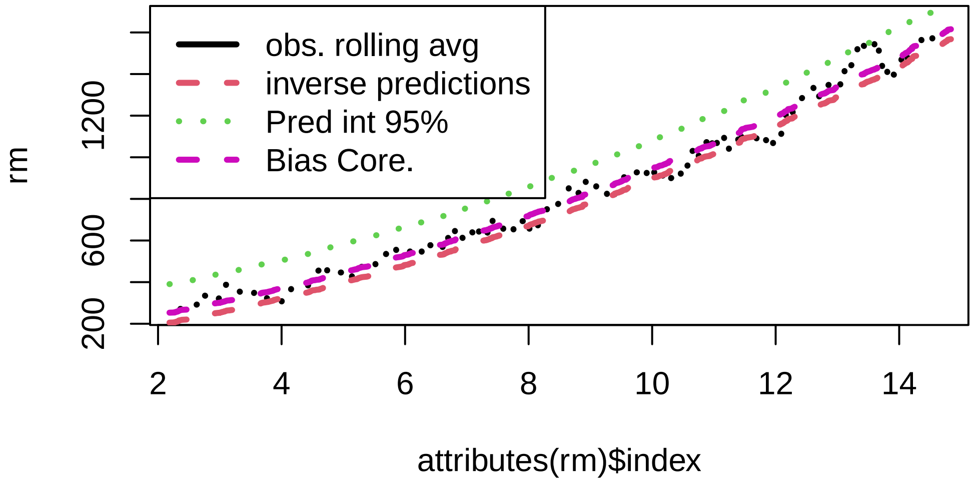
```
Call:
lm(formula = sqrt(Y) ~ X, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-24.2975  -4.5612  -0.0315   4.7083  20.2670 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.94439    0.54223   18.34   <2e-16 ***
X            1.99810    0.05897   33.88   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.919 on 998 degrees of freedom
Multiple R-squared:  0.535, Adjusted R-squared:  0.5345
F-statistic:  1148 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
s=summary(model)$sigma
# plot(X,Y)
# Rolling average



zb=zoo::zoo(x=df$Y,df$X)
rm=zoo::rollmean(zb,50)

plot(attributes(rm)$index,rm,col=1,lty=3,lwd=3,type='l')
zb=zoo::zoo(fitted.values(model)^2,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=2,lty=2,lwd=3)
lines(attributes(rm)$index,rm+s^2,col=6,lty=2,lwd=3)


# Intervals for the mean at each time point
intervals=rowMeans(predict.lm(model,interval = 'prediction')[,2:3]^2)
```

```
Warning in predict.lm(model, interval = "prediction"): predictions on current data refer to
```

```r
zb=zoo::zoo(intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=3,lty=3,lwd=3)
legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Bias Core
```



We see that the bias correction is the best performing method. However, this involves working out the bias for each transformation. For a complicated transformation, this may be quite difficult. For common transformations, this has already been completed for us.

Let's do an example with some real data. The following example is taken from the textbook:

Example 5.2. An electric utility is interested in developing a model relating peak - hour demand Y to total energy usage during the month X. This is an important planning problem because while most customers pay directly for energy usage (in kilowatt - hours), the generation system must be large enough to meet the maximum demand imposed. Data for 53 residential customers for the month of August is given below.

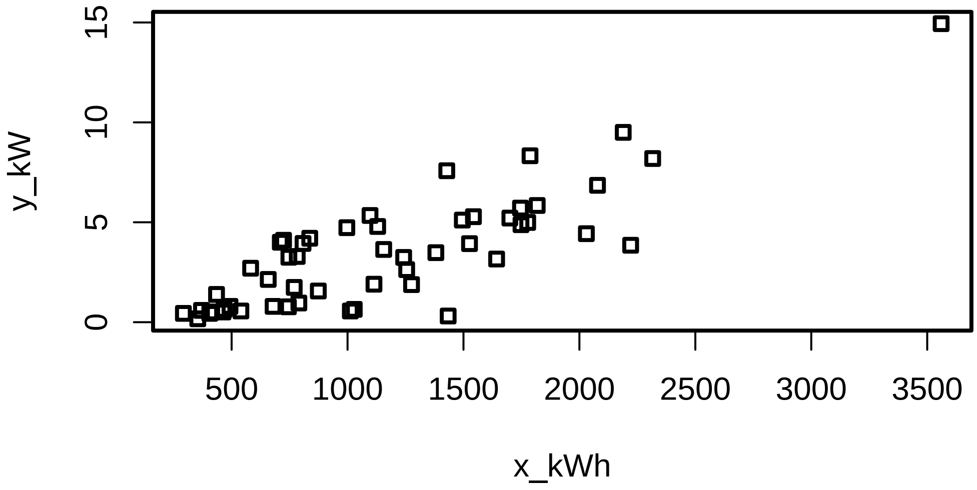
```
# Electric Utility Data

df<- data.frame(
    Customer = c(1:53),
    x_kWh = c(679, 292, 1012, 493, 582, 1156, 997, 2189, 1097, 2078, 1818, 1700, 747, 2030,
    y_kW = c(0.79, 0.44, 0.56, 0.79, 2.70, 3.64, 4.73, 9.50, 5.34, 6.85, 5.84, 5.21, 3.25, 4
  )



df
```

```
   Customer x_kWh  y_kW
```

```
1          1   679  0.79
2          2   292  0.44
3          3  1012  0.56
4          4   493  0.79
5          5   582  2.70
6          6  1156  3.64
7          7   997  4.73
8          8  2189  9.50
9          9  1097  5.34
10        10  2078  6.85
11        11  1818  5.84
12        12  1700  5.21
13        13   747  3.25
14        14  2030  4.43
15        15  1643  3.16
16        16   414  0.50
17        17   354  0.17
18        18  1276  1.88
19        19   745  0.77
20        20   435  1.39
21        21   540  0.56
22        22   874  1.56
23        23  1543  5.28
24        24  1029  0.64
25        25   710  4.00
26        26  1434  0.31
27        27   837  4.20
28        28  1748  4.88
29        29  1381  3.48
30        30  1428  7.58
31        31  1255  2.63
32        32  1777  4.99
33        33   370  0.59
34        34  2316  8.19
35        35  1130  4.79
36        36   463  0.51
37        37   770  1.74
38        38   724  4.10
39        39   808  3.94
40        40   790  0.96
41        41   783  3.29
42        42   406  0.44
43        43  1242  3.24
```

```
44        44   658  2.14
45        45  1746  5.71
46        46   468  0.64
47        47  1114  1.90
48        48   413  0.51
49        49  1787  8.33
50        50  3560 14.94
51        51  1495  5.11
52        52  2221  3.85
53        53  1526  3.93
```

```
#  changing the plot aesthetics
par(pch=22,lwd=2)


# Explore

plot(df[,2:3])
```

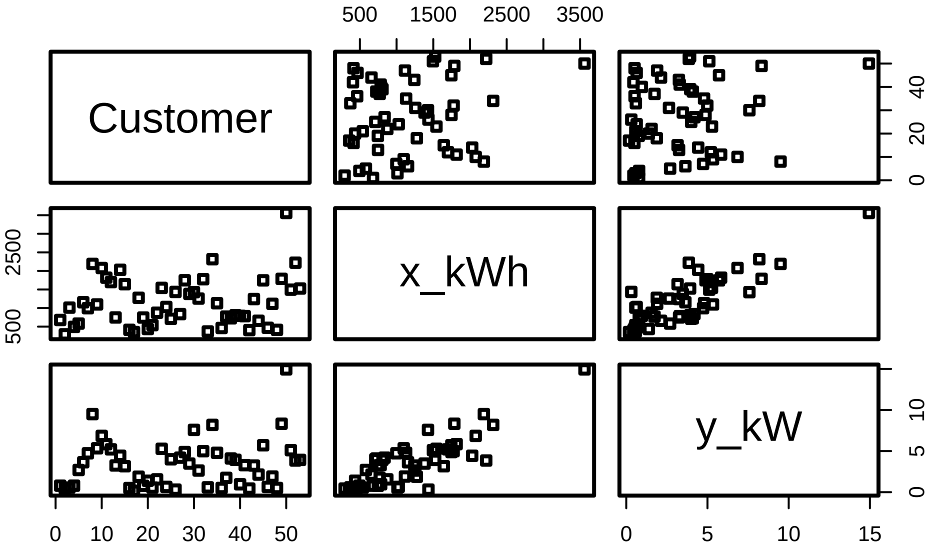


```
summary(df)
```

```
    Customer        x_kWh             y_kW
```

```
 Min.   : 1   Min.   : 292   Min.   : 0.170
 1st Qu.:14   1st Qu.: 679   1st Qu.: 0.790
 Median :27   Median :1029   Median : 3.250
 Mean   :27   Mean   :1153   Mean   : 3.413
 3rd Qu.:40   3rd Qu.:1543   3rd Qu.: 4.880
 Max.   :53   Max.   :3560   Max.   :14.940
```

```
plot(df)
```



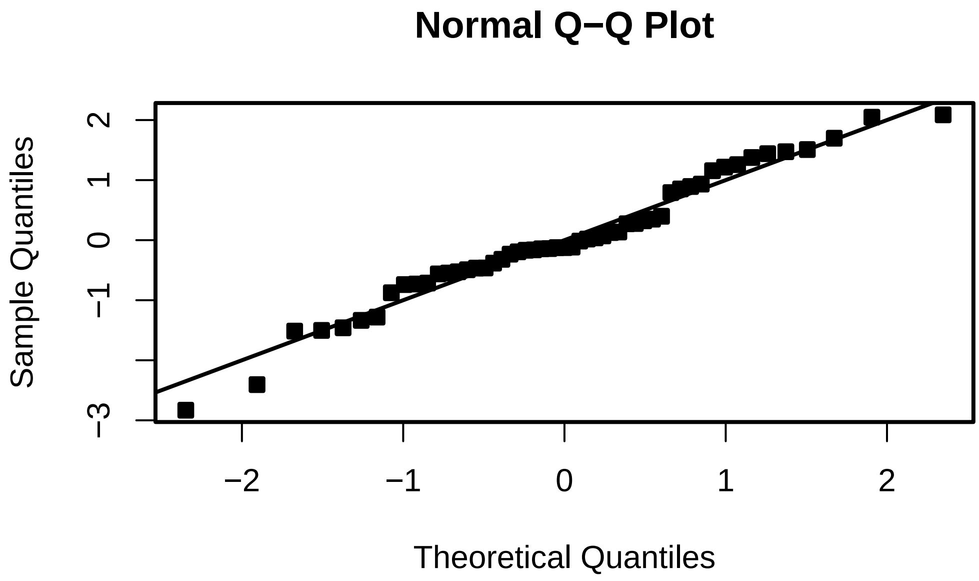
```
# Model

model=lm(y_kW~x_kWh, df); model
```

```

Call:
lm(formula = y_kW ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh  
  -0.831304     0.003683  
```

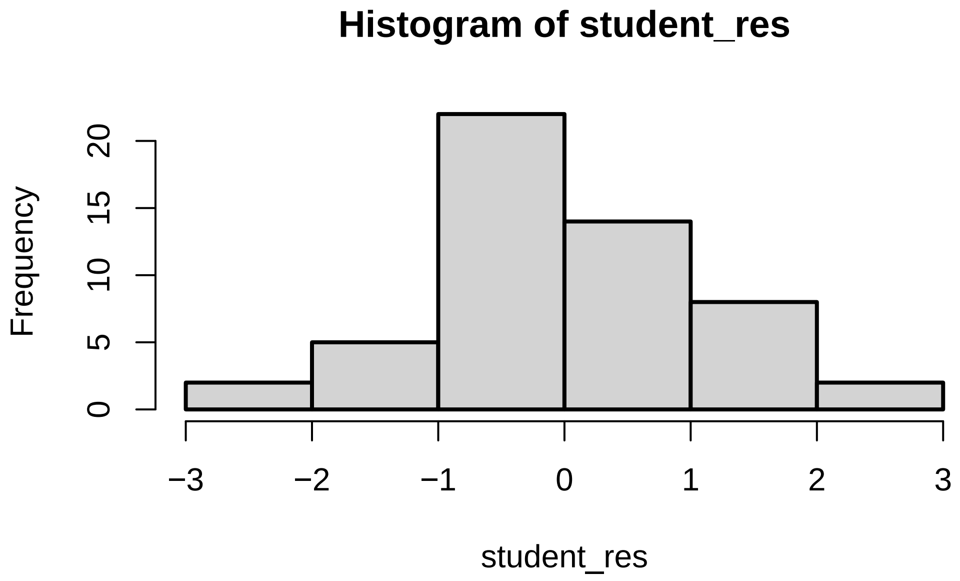
```
summm=summary(model); summ
```

```
Call:
lm(formula = y_kW ~ x_kWh, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.1399 -0.8275 -0.1934  1.2376  3.1522 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.8313037  0.4416121  -1.882   0.0655 .  
x_kWh        0.0036828  0.0003339  11.030 4.11e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.577 on 51 degrees of freedom
Multiple R-squared:  0.7046,	Adjusted R-squared:  0.6988 
F-statistic: 121.7 on 1 and 51 DF,  p-value: 4.106e-15
```

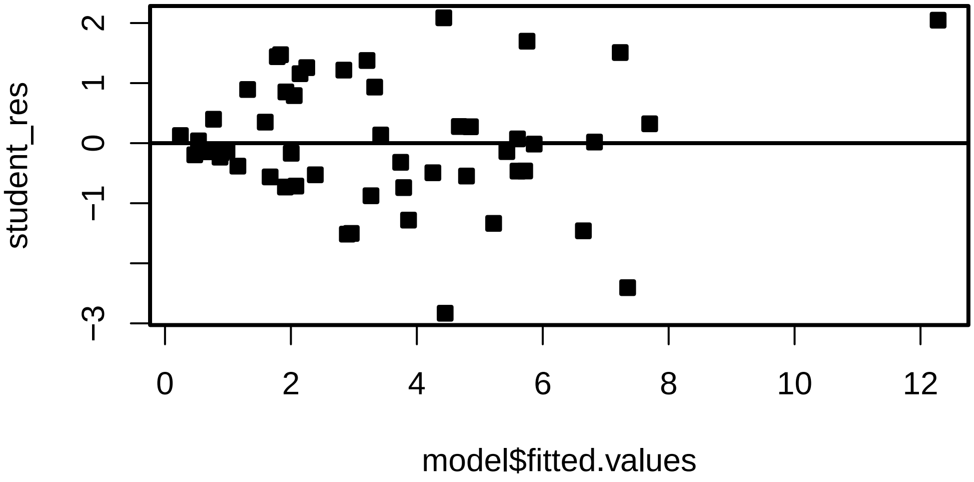
```
# Now do the residual analysis
# Studentized residuals
student_res=rstudent(model)
MSE=summ$sigma^2

qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```



```
hist(student_res,breaks=6)
```



```
plot(model$fitted.values,student_res,pch=22,bg=1)
abline(h=0)
```



We see that the residual variance increases with the mean of Y. This is easily seen by the fan shape of the residuals in the plot of the residuals against the fitted values.

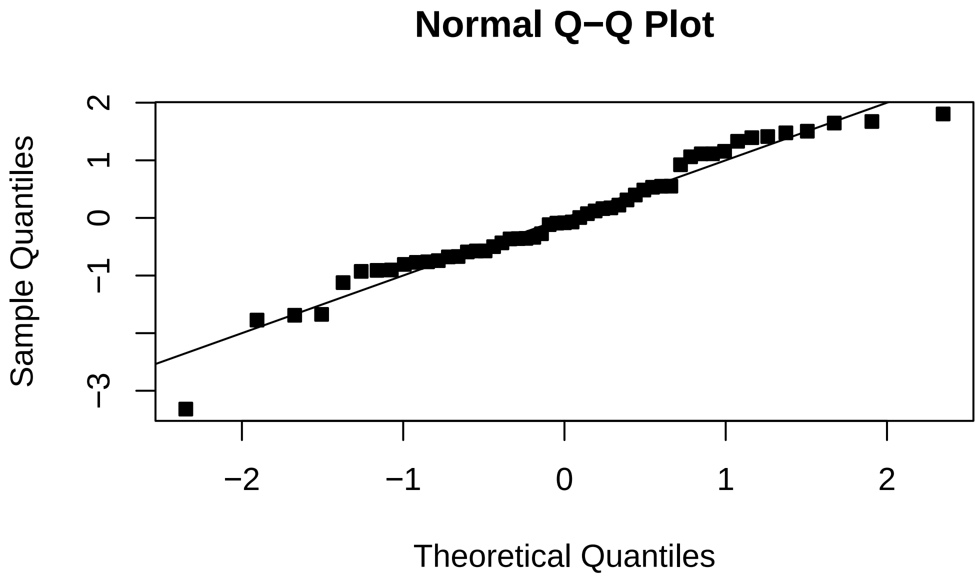
```
######## Let's try the sqrt transformation

model2=lm(sqrt(y_kW)~x_kWh, df)
model2
```

```
Call:
lm(formula = sqrt(y_kW) ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh
  0.5822259    0.0009529
```

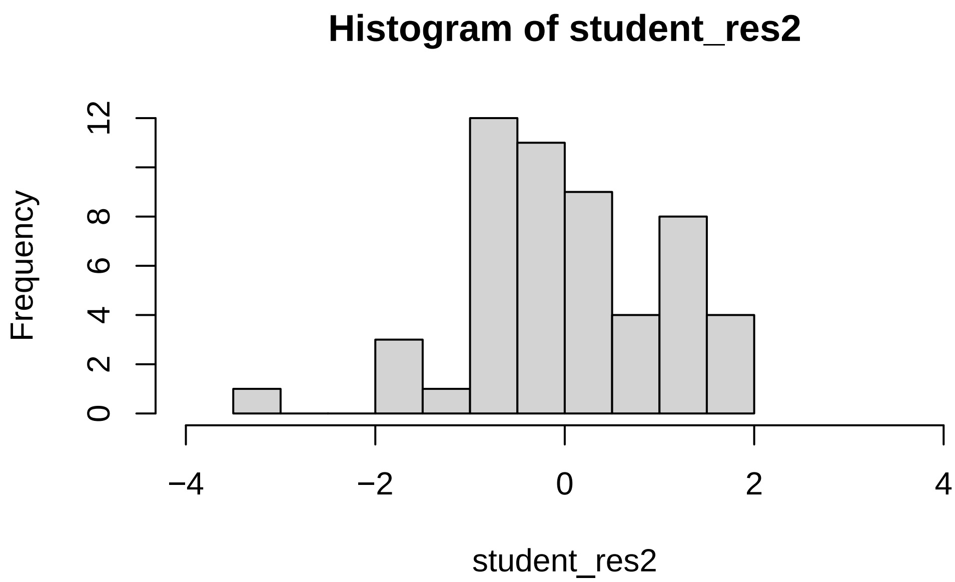
```
summ2=summary(model2); summ2
```

```
Call:
lm(formula = sqrt(y_kW) ~ x_kWh, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.39185 -0.30576 -0.03875  0.25378  0.81027 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.822e-01  1.299e-01   4.481 4.22e-05 ***
x_kWh       9.529e-04  9.824e-05   9.699 3.61e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.464 on 51 degrees of freedom
Multiple R-squared:  0.6485,    Adjusted R-squared:  0.6416 
F-statistic: 94.08 on 1 and 51 DF,  p-value: 3.614e-13
```

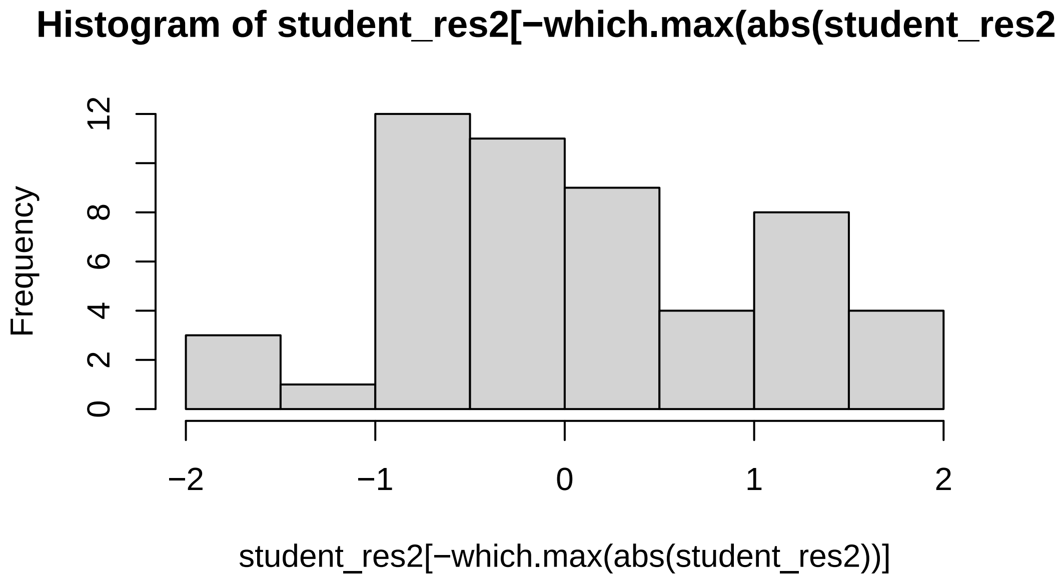
```r
student_res2=rstudent(model2)
MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



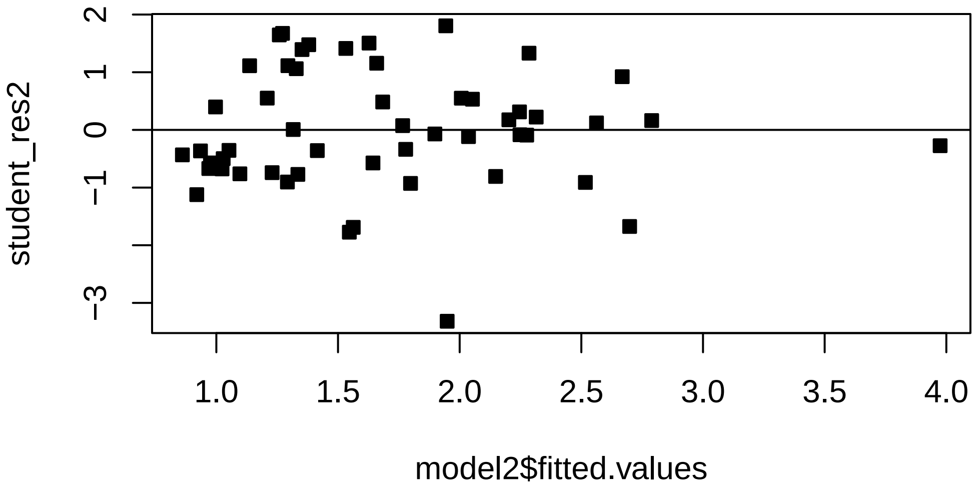
```
hist(student_res2,breaks=10,xlim=c(-4,4))
```



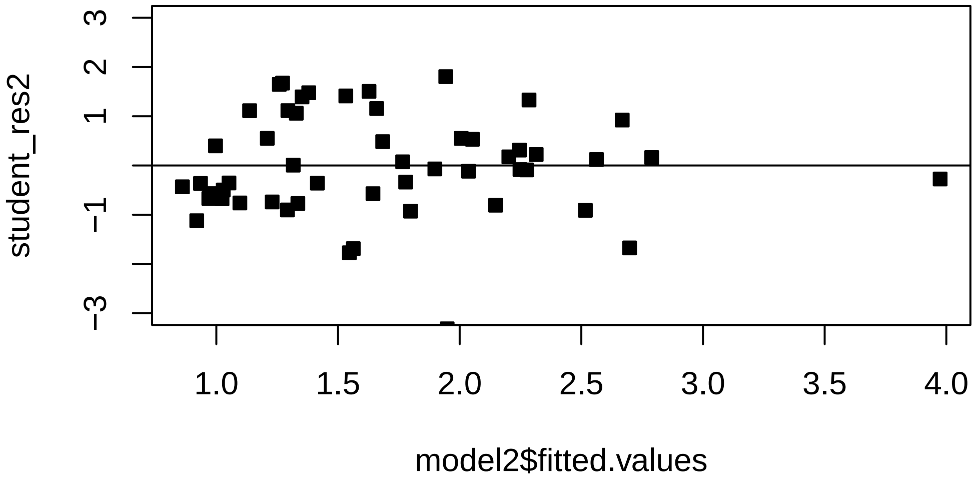
```
hist(student_res2[-which.max(abs(student_res2))],breaks=10,xlim=c(-2,2))
```



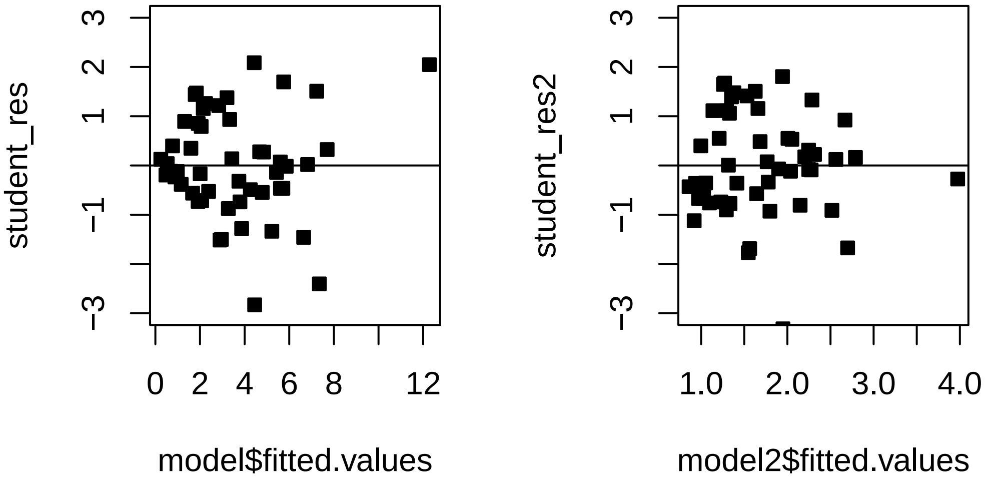
```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



```
# There is one large outlier skewing the previous plot. Let's rescale and remove.
plot(model2$fitted.values,student_res2,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```



```
# Compare!
par(mfrow=c(1,2))
plot(model$fitted.values,student_res,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
plot(model2$fitted.values,student_res2,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```



We see that the transformation has solved the problem. Note that sometimes, even though the square-root transformation may be more suitable, the analyst may opt for the logarithm transform. This is because the log transformation gives a nicer interpretation to the coefficients. In this case, that is not working well, see below:

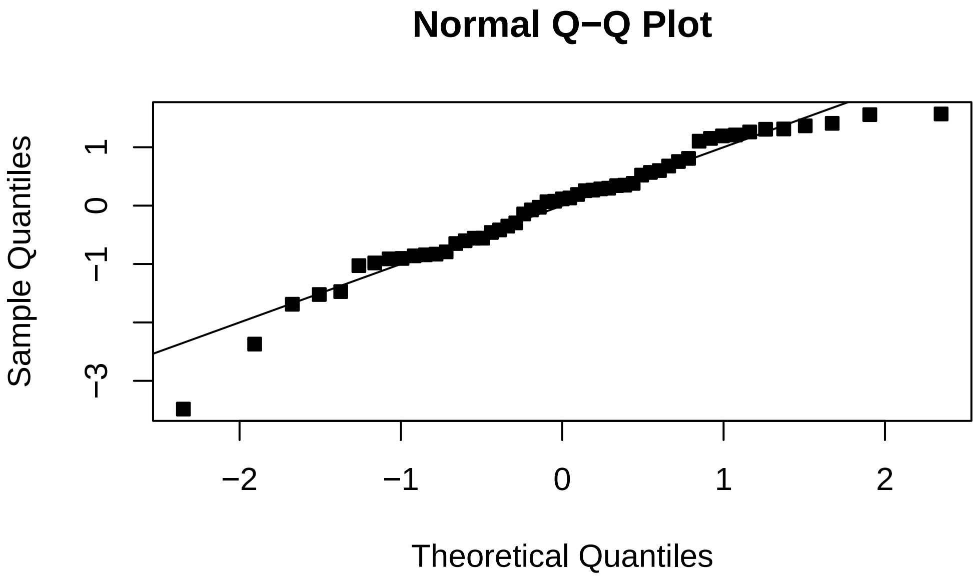
```
######## Let's try the log transformation
par(mfrow=c(1,1))


model3=lm(log(y_kW)~x_kWh, df)
model3
```

```
Call:
lm(formula = log(y_kW) ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh
  -0.558713     0.001172
```

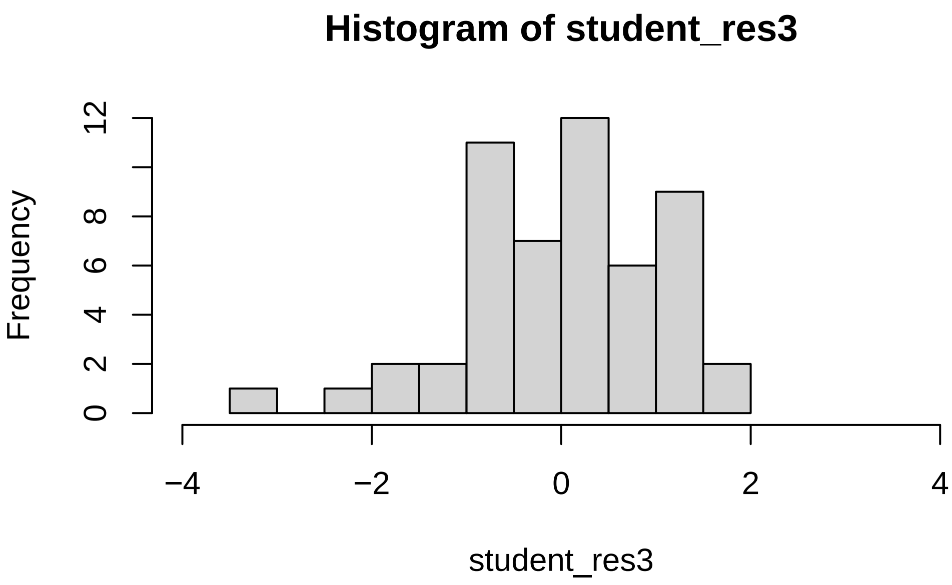
```
summ3=summary(model3); summ3
```

```
Call:
lm(formula = log(y_kW) ~ x_kWh, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.29261 -0.47256  0.08414  0.49628  1.12143 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.5587131  0.2057201  -2.716    0.009 ** 
x_kWh        0.0011716  0.0001555   7.533 7.86e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7347 on 51 degrees of freedom
Multiple R-squared:  0.5266,    Adjusted R-squared:  0.5174 
F-statistic: 56.74 on 1 and 51 DF,  p-value: 7.862e-10
```

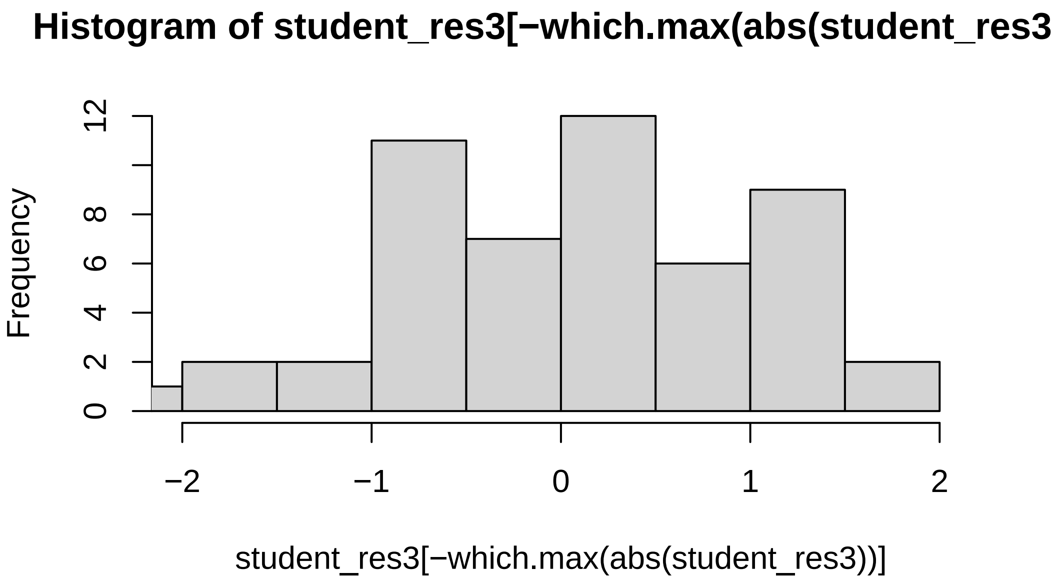
```
student_res3=rstudent(model3)
MSE3=summ3$sigma^2
qqnorm(student_res3,pch=22,bg=1)
abline(0,1)
```



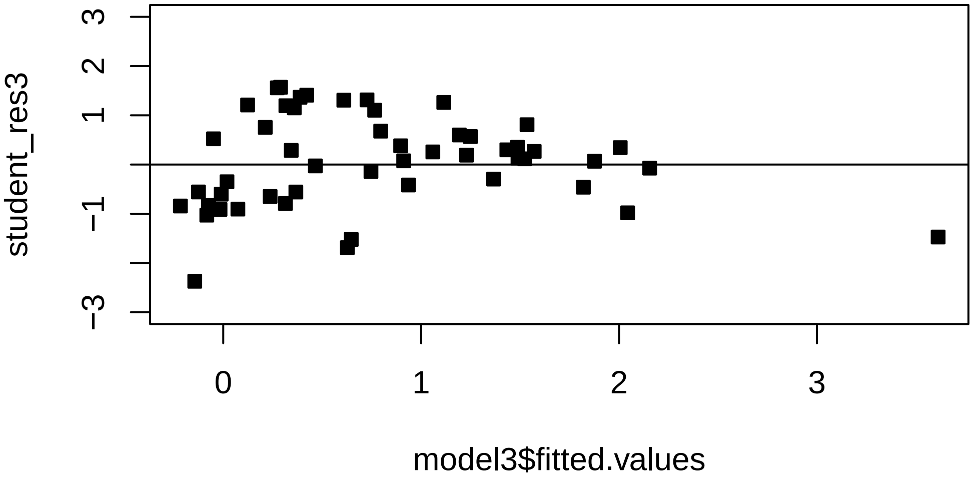
```
hist(student_res3,breaks=10,xlim=c(-4,4))
```



```
hist(student_res3[-which.max(abs(student_res3))],breaks=10,xlim=c(-2,2))
```



```
plot(model3$fitted.values,student_res3,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```



5.1.1 Linearizing the model

Moving on, we may suspect that the relationship between the regressors and the response is nonlinear, either through empirical evidence or theoretical justification. In some cases a nonlinear function can be linearized by using a suitable transformation. Such nonlinear models are called intrinsically linear. For example, consider the model $Y = \beta_0 e^{\beta_1 X}\epsilon$. Taking the log of both sides yields:

$$\log(Y) = \log(\beta_0) + \beta_1 X + \log(\epsilon).$$

Reparameterizing with $Z = \log(Y)$, $\alpha_0 = \log(\beta_0)$ and $\eta = \log(\epsilon)$, we have that

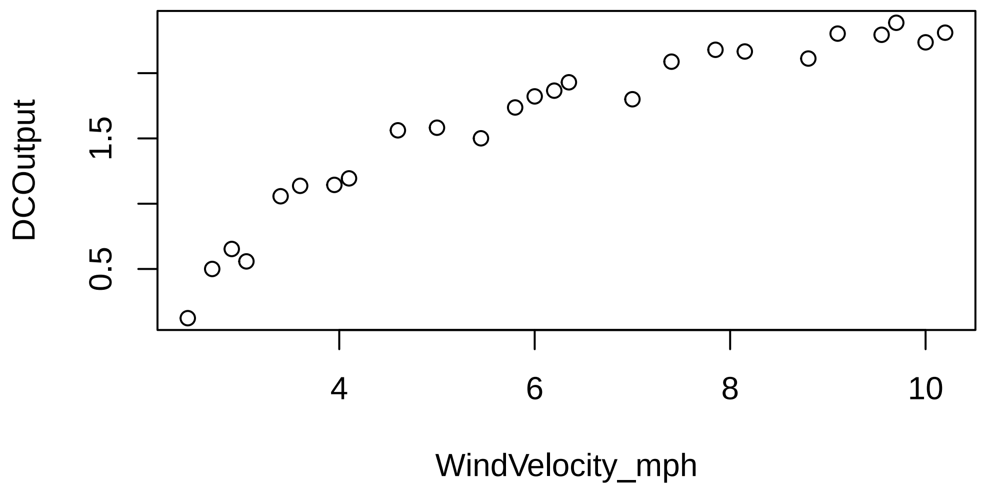
$$Z = \alpha_0 + \beta_1 X + \eta.$$

If we are willing to assume that η are symmetric about 0 with a constant variance, then we can run a linear regression with the model given above. To get estimates for the original units Y, we can transform back as previously discussed. A model is linearizable if there exists some reparameterization which places the model in the form of the MLR.

Exercise 5.1. Show the following models are linearizable - that is, find the linear reparameterization of the following models: 1. $Y = \beta_0 X_1^{\beta}$ 2. $Y = \beta_0 X^{\beta_1 X}$ 3. $Y = \beta_0 + \log X$ 4. $Y = X/(\beta_0 X - \beta_1)$

Example 5.3. A research engineer is investigating the use of a windmill to generate electricity. He has collected data on the DC output from his windmill and the corresponding wind velocity. See below. Find a well-fitting regression model for this data.

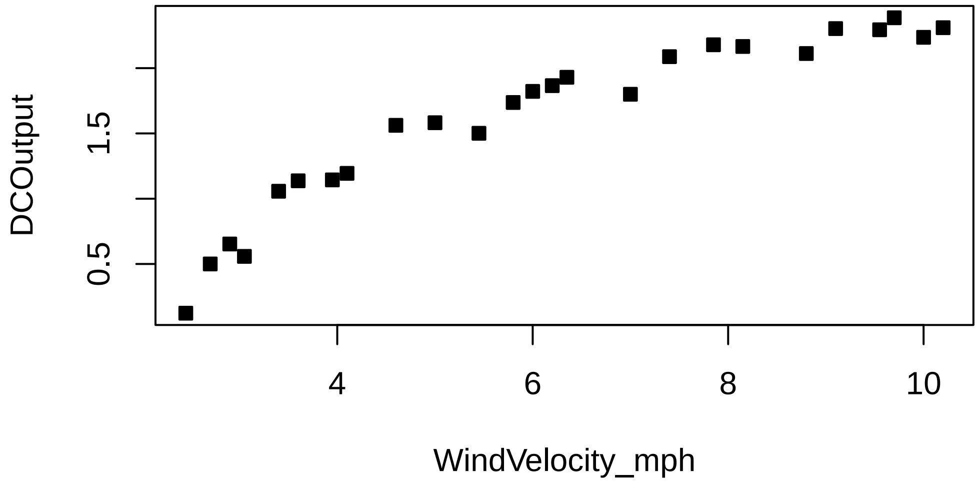
```
########### Windmill data





# Create the data frame
df_wind <- data.frame(
  WindVelocity_mph = c(5.00, 6.00, 3.40, 2.70, 10.00, 9.70, 9.55, 3.05, 8.15, 6.20,
                       2.90, 6.35, 4.60, 5.80, 7.40, 3.60, 7.85, 8.80, 7.00, 5.45,
                       9.10, 10.20, 4.10, 3.95, 2.45),
  DCOutput = c(1.582, 1.822, 1.057, 0.500, 2.236, 2.386, 2.294, 0.558, 2.166, 1.866,
               0.653, 1.930, 1.562, 1.737, 2.088, 1.137, 2.179, 2.112, 1.800, 1.501,
               2.303, 2.310, 1.194, 1.144, 0.123)
)

###############


par(mfrow=c(1,1))
plot(df_wind)
```



```
summary(df_wind)
```

```
 WindVelocity_mph    DCOutput
 Min.   : 2.450   Min.   :0.123
 1st Qu.: 3.950   1st Qu.:1.144
 Median : 6.000   Median :1.800
 Mean   : 6.132   Mean   :1.610
 3rd Qu.: 8.150   3rd Qu.:2.166
 Max.   :10.200   Max.   :2.386
```

```
plot(df_wind,pch=22,bg=1)
```



```
model=lm(DCOutput~WindVelocity_mph, df_wind)
model
```

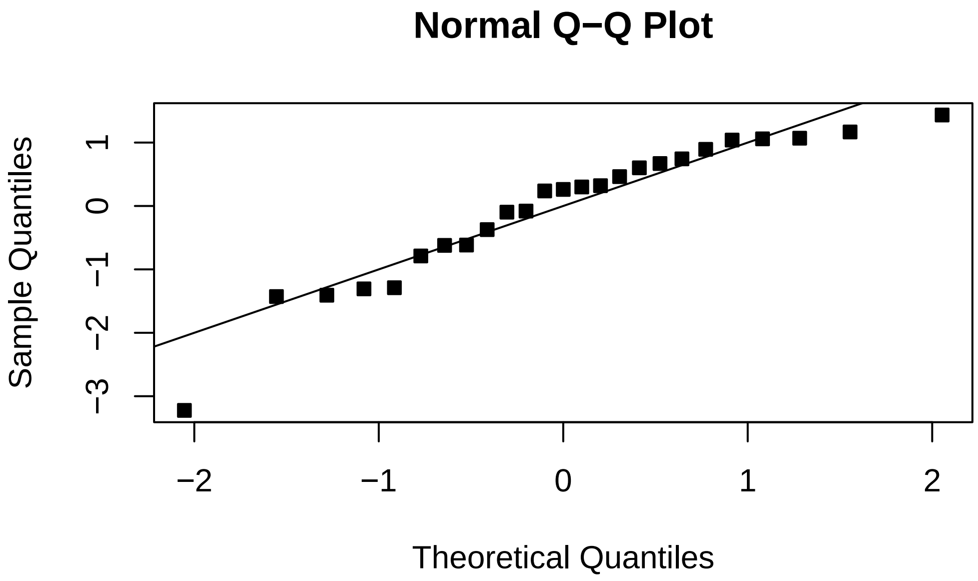
```
Call:
lm(formula = DCOutput ~ WindVelocity_mph, data = df_wind)

Coefficients:
     (Intercept)  WindVelocity_mph
          0.1309            0.2411
```

```
summ=summary(model); summ
```

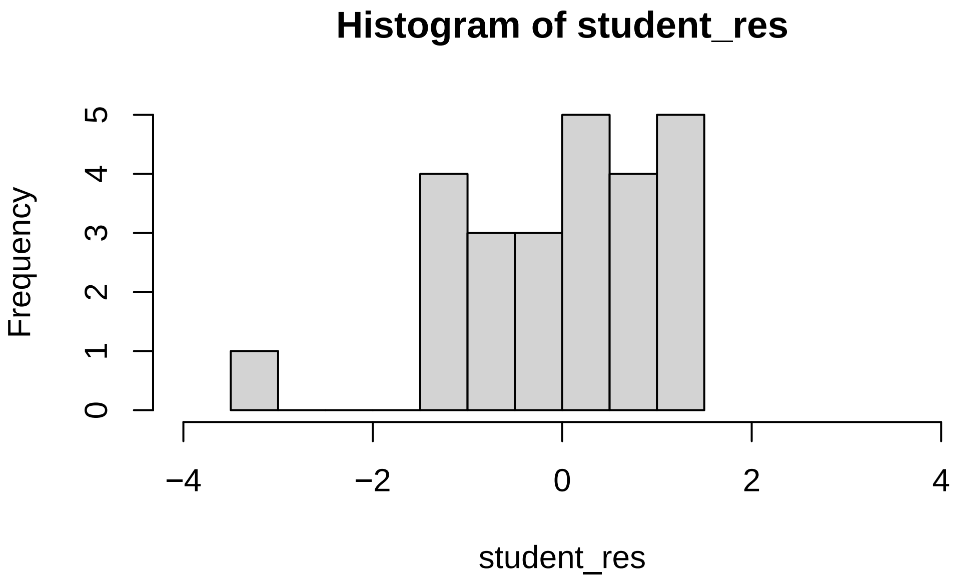
```
Call:
lm(formula = DCOutput ~ WindVelocity_mph, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59869 -0.14099  0.06059  0.17262  0.32184

Coefficients:
```

```
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.13088    0.12599   1.039     0.31
WindVelocity_mph  0.24115    0.01905  12.659 7.55e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2361 on 23 degrees of freedom
Multiple R-squared:  0.8745,    Adjusted R-squared:  0.869
F-statistic: 160.3 on 1 and 23 DF,  p-value: 7.546e-12
```

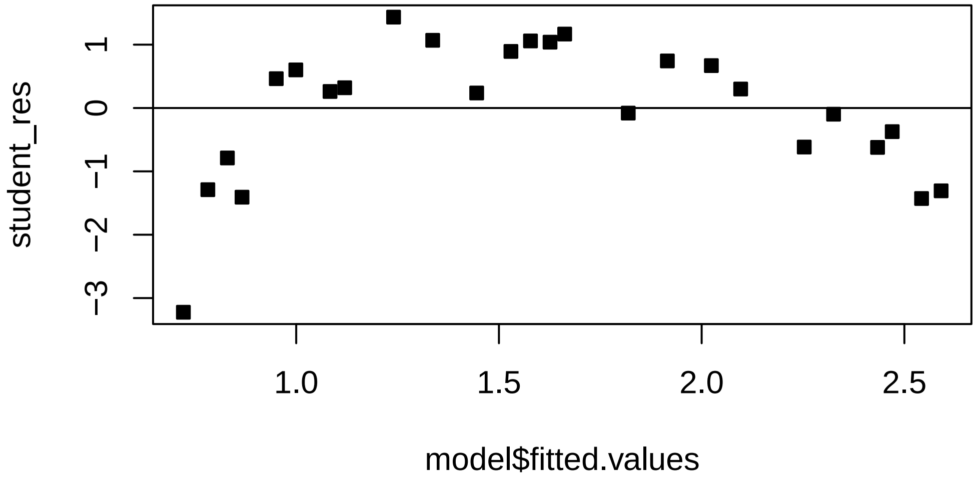
```
student_res=rstudent(model)

MSE=summ$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```



```
hist(student_res,breaks=10,xlim=c(-4,4))
```

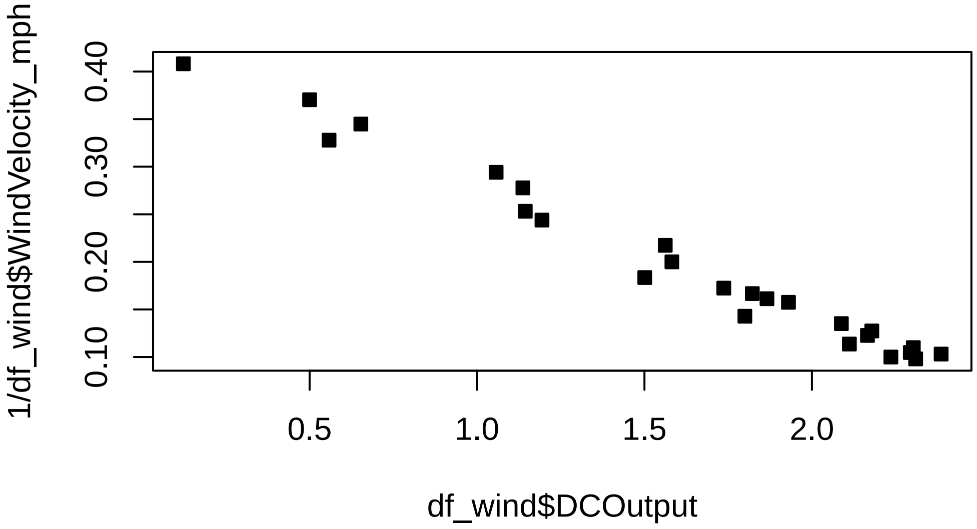


```
plot(model$fitted.values,student_res,pch=22,bg=1)
abline(h=0)
```



The fit is not good. Looking at the scatterplot, we might initially consider using a quadratic model to account for the pictured curvature. However, the scatterplot suggests that as wind speed increases, DC output approaches an upper limit of approximately 2.5. This is also consistent with the theory of windmill operation. Since the quadratic model will eventually bend downward as wind speed increases, it would not be appropriate for these data. A more reasonable model for the windmill data that incorporates an upper asymptote would be based on $1/X$.

```
plot(df_wind$DCOutput,1/df_wind$WindVelocity_mph,pch=22,bg=1)
```



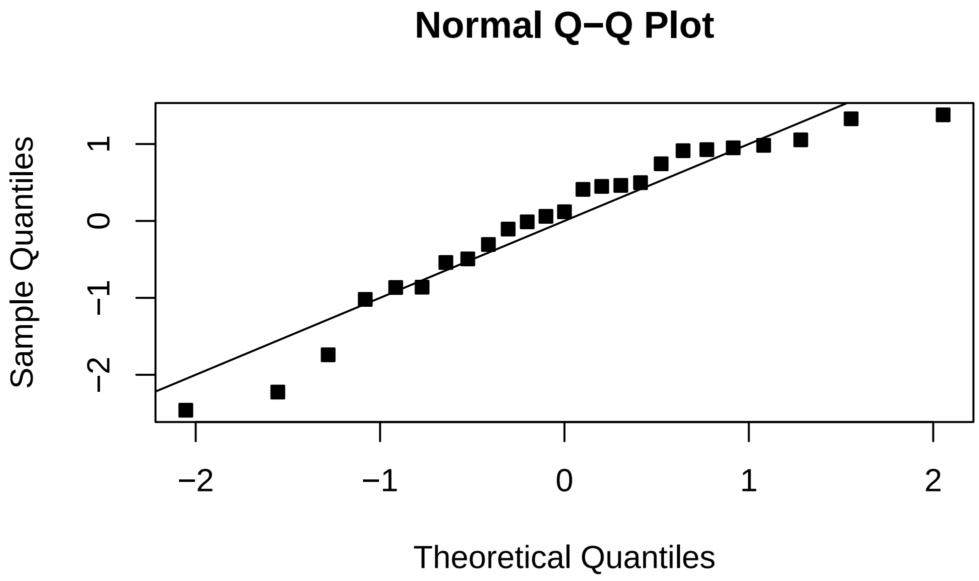
```
# plot(df$DCOutput,log(df$WindVelocity_mph))
df_wind$WindVelocity_mph_inv=1/df_wind$WindVelocity_mph
model2=lm(DCOutput~WindVelocity_mph_inv, df_wind)
model2
```

```

Call:
lm(formula = DCOutput ~ WindVelocity_mph_inv, data = df_wind)

Coefficients:
         (Intercept)  WindVelocity_mph_inv  
               2.979                -6.935  
```

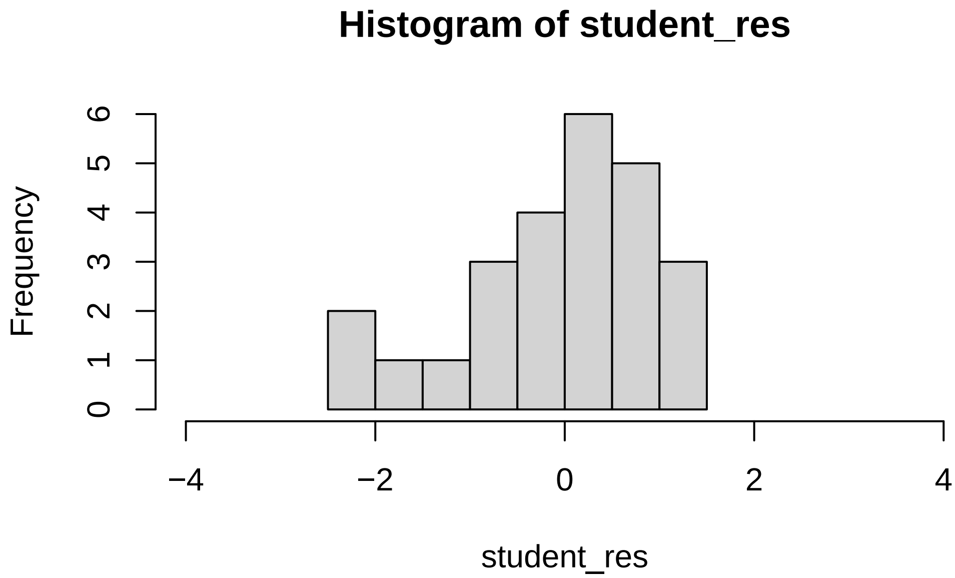
```
summ=summary(model2); summ
```

```
Call:
lm(formula = DCOutput ~ WindVelocity_mph_inv, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.20547 -0.04940  0.01100  0.08352  0.12204 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)            2.9789     0.0449   66.34   <2e-16 ***
WindVelocity_mph_inv  -6.9345     0.2064  -33.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09417 on 23 degrees of freedom
Multiple R-squared:   0.98, Adjusted R-squared:  0.9792 
F-statistic:  1128 on 1 and 23 DF,  p-value: < 2.2e-16
```

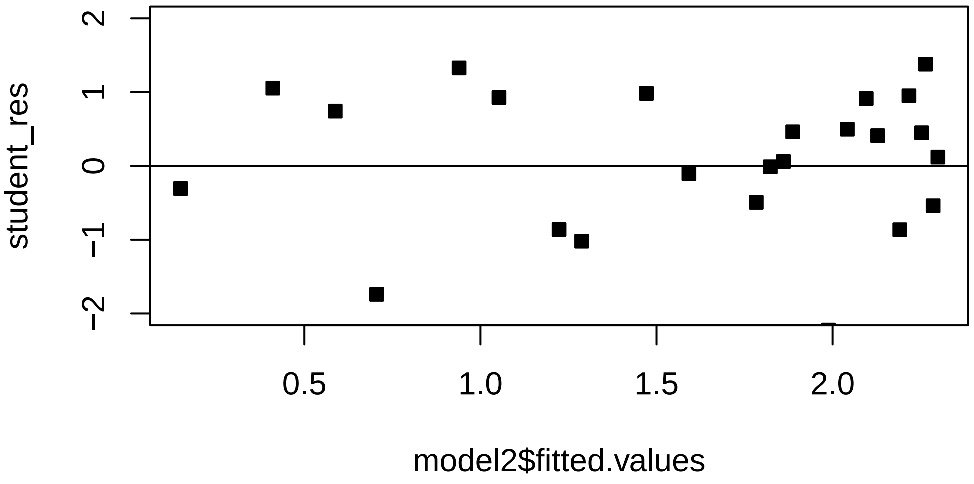
```
student_res=rstudent(model2)
MSE=summ$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```



```
hist(student_res,xlim=c(-4,4))
```



```
plot(model2$fitted.values,student_res,pch=22,bg=1,ylim=c(-2,2))
abline(h=0)
```



5.1.2 Box Cox Transformations

One technique is to use the data to estimate which transformation is best, a popular instance is the Box-Cox transformation. Consider the class of transformations: $\{y^\lambda : \lambda \in \mathbb{R}\}$. The regression model and λ can be estimated simultaneously using the method of maximum likelihood. Recall that we used the method of least squares to estimate the model parameters - maximum likelihood is an alternative estimation strategy. Think of λ like an extra model parameter, on top of β and σ that we need to estimate.

Let

$$\tilde{y} = \log^{-1}(1/n \sum_{i=1}^{n} \log y_i)$$

$$y_\lambda = \begin{cases} \dfrac{y^\lambda - 1}{\lambda \tilde{y}^{\lambda-1}} & \lambda \neq 0 \\ \tilde{y} \log y & \lambda = 0 \end{cases}.$$

We then fit the following model

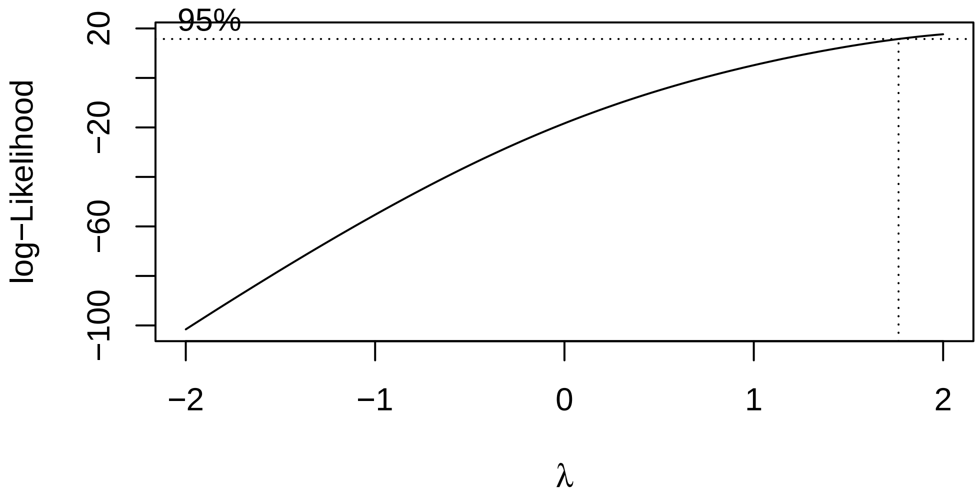
$$y_\lambda = X\beta + \epsilon.$$

Even though $y^\lambda \neq y_\lambda$, we use y^λ (or $\log y$ if $\lambda = 0$) as the final response - as it is more interpretable. It is entirely acceptable to use y_λ as the response for the final model - this model will have a scale difference and an origin shift in comparison to the model using y^λ (or $\log y$). Usually the final λ used in the model is rounded to a nice number for interpretation. A computational procedure is used for estimating λ, which we will not cover here. In general, we can compute a confidence interval for λ, and if it contains 1 then we may not need to transform.

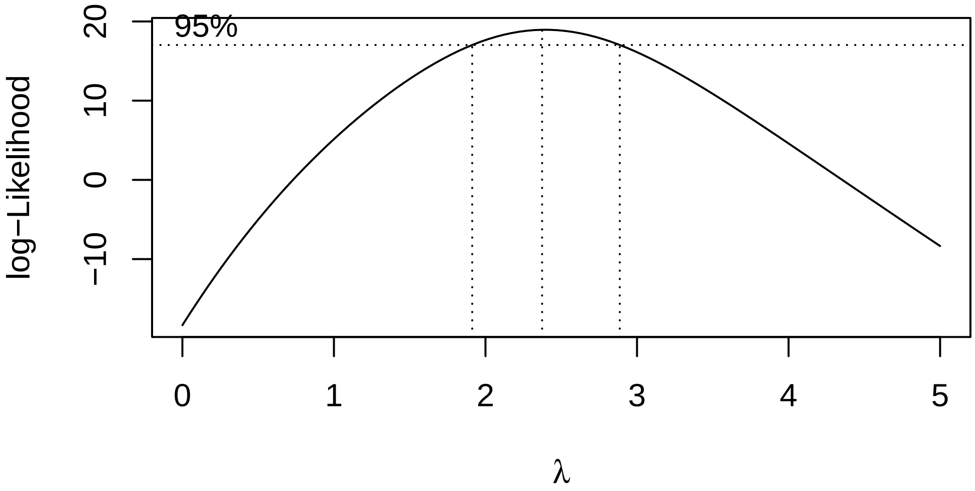
Example 5.4. Let's apply the Box-Cox transformation to the two previous examples.

The R function `boxcox` from the `MASS` package can be used to execute the Box-Cox transformation. It requires you to specify a grid of points for λ, given below by `seq(-2, 2, 1/10)`. We can set the `plotit` parameter to `TRUE` in order to see if this grid is big enough. We should see a peak or mode in the log-likelihood function that is plotted. If we don't, we can expand the grid on the side which has the largest value of the log-likelihood. Observe below that we need to include points higher than 2 on the grid, as the function is still increasing for at $\lambda = 2$:

```r
# You need the MASS package.
# install.packages('MASS')

bc=MASS::boxcox(DCOutput~WindVelocity_mph,data=df_wind,
       lambda = seq(-2, 2, 1/10),
       plotit = TRUE,
       eps = 1/50,
       xlab = expression(lambda),
       ylab = "log-Likelihood")
```



```
# bc

#Observe that

bc=MASS::boxcox(DCOutput~WindVelocity_mph,data=df_wind,
                lambda = seq(0, 5, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
```



```
# bc

#Seems like we should try lambda=2
```

The confidence interval goes from just below 2 to just below 3. Let's pick a round number, and try the transformation $\lambda = 2$.

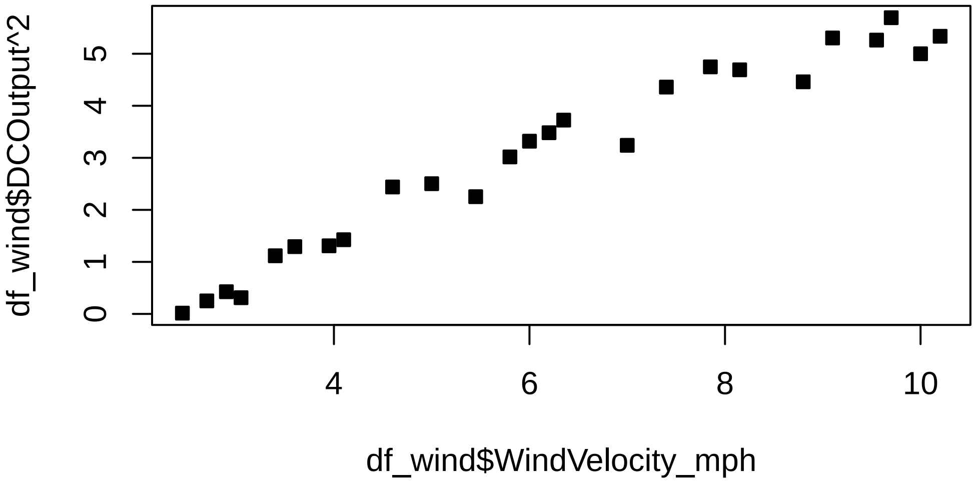
```
# plot(df$DCOutput,log(df$WindVelocity_mph))
model3=lm(DCOutput^2~WindVelocity_mph, df_wind)
model3
```

```
Call:
lm(formula = DCOutput^2 ~ WindVelocity_mph, data = df_wind)

Coefficients:
     (Intercept)  WindVelocity_mph
         -1.3585            0.7107
```

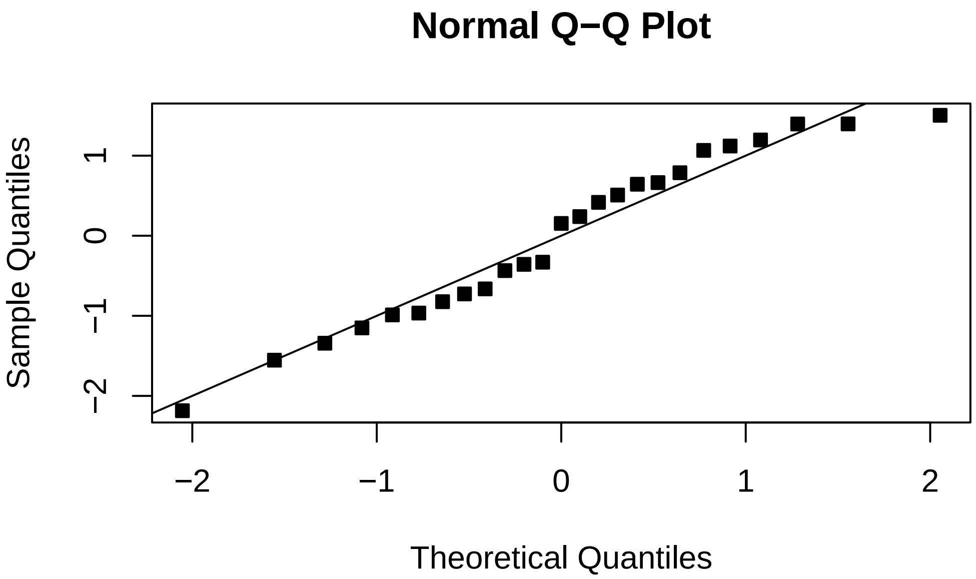
```
summ3=summary(model3); summ3
```

```
Call:
lm(formula = DCOutput^2 ~ WindVelocity_mph, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.74840 -0.31027  0.05951  0.30793  0.57072 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -1.35851    0.21239  -6.396 1.58e-06 ***
WindVelocity_mph  0.71066    0.03211  22.130  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3979 on 23 degrees of freedom
Multiple R-squared:  0.9551,    Adjusted R-squared:  0.9532 
F-statistic: 489.7 on 1 and 23 DF,  p-value: < 2.2e-16
```

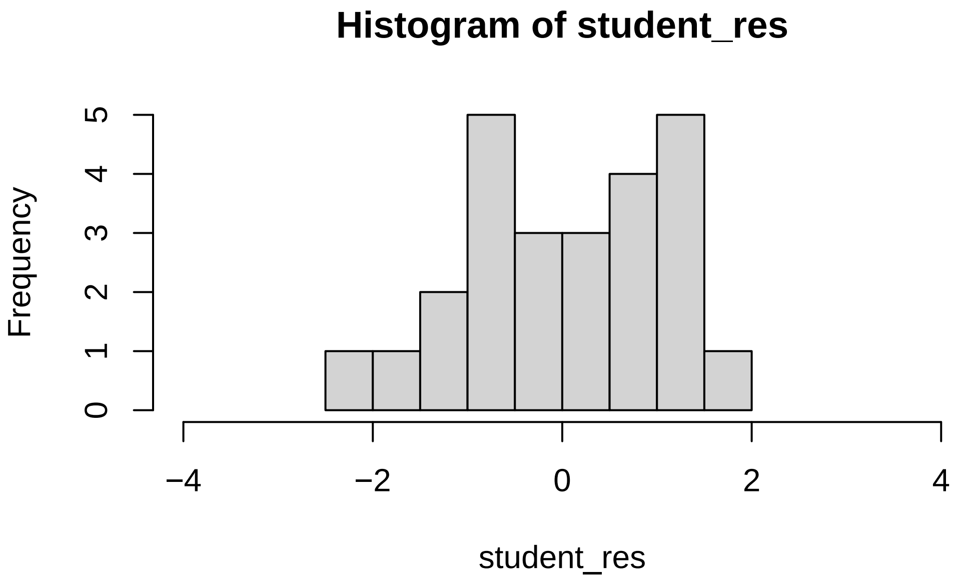
```r
plot(df_wind$WindVelocity_mph,df_wind$DCOutput^2,pch=22,bg=1)
```



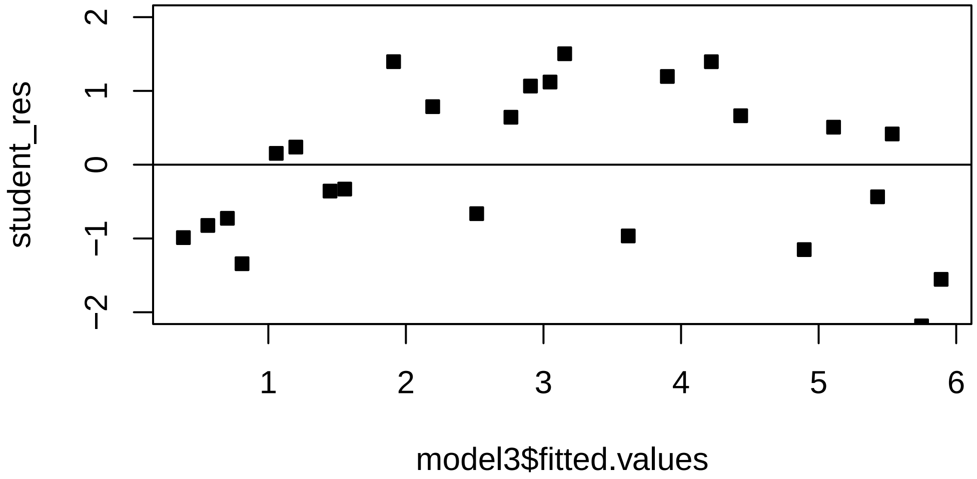
```
student_res=rstudent(model3)
MSE=summ3$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```



```
hist(student_res,xlim=c(-4,4))
```



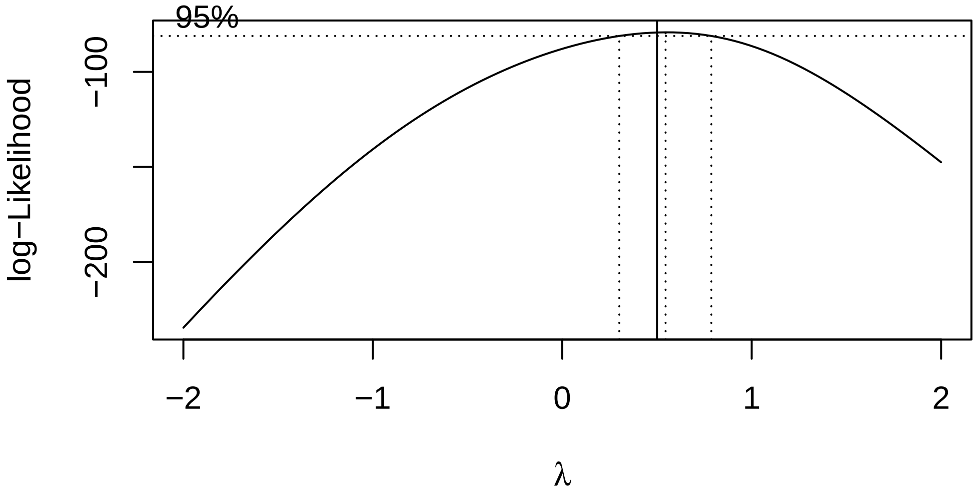
```
plot(model3$fitted.values,student_res,pch=22,bg=1,ylim=c(-2,2))
abline(h=0)
```



The fit is not bad. The R^2 is very high. There is a pattern in the QQplot and a slight pattern in the residuals plot. For knowing nothing about wind velocity, it is not bad.

The electricity data clearly points to the square root transformation - matching the analysis we did previously.

```
## Electricity

model=lm(y_kW~x_kWh, df)

bc=MASS::boxcox(y_kW~x_kWh,data=df,
                lambda = seq(-2, 2, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
abline(v=0.5)
```



```
# bc
```

5.1.3 Homework stop

Complete the assigned Chapter 5 questions.

References

Miller, Don M. 1984. "Reducing Transformation Bias in Curve Fitting." *The American Statistician* 38 (2): 124–26. http://www.jstor.org/stable/2683247.

6 Introduction to R software

6.1 Some Basics

R is a Statistical Programming language, it consists of 2 types of objects: data and functions.

```
##Data
x<-2
print(x)
```

```
[1] 2
```

```
##function
log(2)
```

```
[1] 0.6931472
```

Data is stored in variables and can take many forms. To store a value in a variable use "<-", above we set the variable x equal to 2. There are many data types in R, we will go through some of them.

```
#real numbers
num=29.333
num
```

```
[1] 29.333
```

```
#Some math
#adding and subtraction
2+3-2
```

```
[1] 3
```

```
#multiplying and dividing
num<-5*(10/25)
num
```

```
[1] 2
```

```
#Strings
word<-"hello"
word
```

```
[1] "hello"
```

```
word='hello'
```

6.2 Booleans

Booleans take on either TRUE or FALSE values, and can be very useful in R. You can set booleans to the result of a comparison of two data types, some of the syntax is below:

- <,>,<=,>= corresponds to less than, greater than, less than or equal, greater than or equal
- ==, != equals, not equals
- && , written like a&&b where a and b are booleans, it is TRUE is *both* a and b are TRUE
- || , written like a||b where a and b are booleans, it is TRUE if at least *one of* a and b are TRUE

```
#booleans can be initialize in a variety of ways, for example
#must capitalize the true or false
FALSE
```

```
[1] FALSE
```

```
F
```

```
[1] FALSE
```

```
T
```

```
[1] TRUE
```

```
myBoolean<-TRUE
myBoolean
```

```
[1] TRUE
```

```
myBoolean2<- 3<4
myBoolean2
```

```
[1] TRUE
```

```
myBoolean3<-"this"=="that"
myBoolean3
```

```
[1] FALSE
```

```
## && (and) is TRUE if BOTH input booleans are true
## || (or) is TRUE if AT LEAST one input boolean is true
myBoolean4<-myBoolean2&&myBoolean
myBoolean4
```

```
[1] TRUE
```

6.3 Vectors

Vectors in R are used frequently, they are “lists” or “arrays” of all the same data type.

```
##vectors are created with c(data,data,data)
myVector<-c(2,3,4,5,6,7,8,9,10)
myVector
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```
#a:b is a shortcut for a sequence from a to b adding 1
#you can create vectors of sequences using seq(), for more type ?seq in the console
myVector2<-2:10
myVector2
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```
as.numeric(2:10)
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```
as.double(2:10)
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```
myVector2<-rep(NA,l=20)


#These do not have to be numbers, they can be vectors, Strings, booleans...
myVector<-c(myVector,myVector)
myVector
```

```
 [1]  2  3  4  5  6  7  8  9 10  2  3  4  5  6  7  8  9 10
```

```
myVector3<-c("this","is","a","vector","of","strings")
myVector3
```

```
[1] "this"    "is"      "a"       "vector"  "of"      "strings"
```

```
#access elements with square brackets []
myVector[1]
```

```
[1] 2
```

```
#more advanced accesssing
#access elements 1 to 5
myVector[1:5]
```

```
[1] 2 3 4 5 6
```

```
#access elements 1, 4 and 6
myVector[c(1,4,6)]
```

```
[1] 2 5 7
```

```
#access elements that are greater than 2
myVector[myVector>2]
```

```
 [1]  3  4  5  6  7  8  9 10  3  4  5  6  7  8  9 10
```

```
myVector[-c(1,4,6)]
```

```
 [1]  3  4  6  8  9 10  2  3  4  5  6  7  8  9 10
```

We can perform mathematical operations and comparisons on vectors

```
x<-1:10
x
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
#adds 1 to every element
x+1
```

```
 [1]  2  3  4  5  6  7  8  9 10 11
```

```
#this works for comparisons
x<4
```

```
 [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
x[x<4]
```

```
[1] 1 2 3
```

```
#multiplies element 1 to element 1 of second vectors
x*-(1:10)
```

```
 [1]   -1   -4   -9  -16  -25  -36  -49  -64  -81 -100
```

```
#beware repetition
x-c(1,2)
```

```
 [1] 0 0 2 2 4 4 6 6 8 8
```

```
# mathematical operations on the vector apply to each element

#squares each element
x^2
```

```
 [1]   1   4   9  16  25  36  49  64  81 100
```

```
#log each element
log(x)
```

```
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
 [8] 2.0794415 2.1972246 2.3025851
```

```
#Example: Dot Product
x<-c(1,2,3)
y<-c(2,5,8)
#sum adds the elements of the vector together
sum(x*y)
```

```
[1] 36
```

6.4 Matrices

You can also use matrices in R.

```
#you can create a matrix with matrix(vector of data,nrow=number of rows,ncol=number of colum
#You can see it will fill in the data down the columns first
myMatrix<-matrix(1:9,nrow=3,ncol=3); myMatrix
```

```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
myMatrix
```

```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
#rbind and cbind add a row or column repectively to the matrix
#you can create matrices with rbind(rowvector1,rowvector2,...), or with cbind(column vector

myMatrix<-rbind(c(2,3,4),c(3,4,5),c(1,2,3))
myMatrix
```

```
     [,1] [,2] [,3]
[1,]    2    3    4
[2,]    3    4    5
[3,]    1    2    3
```

```
myMatrix2<-cbind(c(1,2,3),c(4,5,6),c(7,8,9))
myMatrix2
```

```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
myMatrix3<-cbind(myMatrix2,c(10,11,12))
myMatrix3
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```
myMatrix3<-cbind(c(10,11,12),myMatrix2)
```

We can also do Matrix math:

```
#again math functions apply to every element
myMatrix^2
```

```
     [,1] [,2] [,3]
[1,]    4    9   16
[2,]    9   16   25
[3,]    1    4    9
```

```
#multiply with '%*%
myMatrix2%*%myMatrix
```

```
     [,1] [,2] [,3]
[1,]   21   33   45
[2,]   27   42   57
[3,]   33   51   69
```

```
#we can find the inverse with 'solve()
X<-matrix(c(1,0,1,-2,3,0,1,4,2),nrow=3)
X
```

```
     [,1] [,2] [,3]
[1,]    1   -2    1
[2,]    0    3    4
[3,]    1    0    2
```

```
solve(X)
```

```
     [,1] [,2] [,3]
[1,] -1.2 -0.8  2.2
[2,] -0.8 -0.2  0.8
[3,]  0.6  0.4 -0.6
```

```
#check dimension
dim(X)
```

```
[1] 3 3
```

```
#We can also transpose with t()
t(X)
```

```
     [,1] [,2] [,3]
[1,]    1    0    1
[2,]   -2    3    0
[3,]    1    4    2
```

```
#Some times to multiply vectors we have to trun them into matrix types
myVector<-c(1,2,3)
newM<-matrix(myVector,ncol=1)
```

6.5 Functions

Functions are objects that take an input and transform it into some output, just like in mathematics. We have already seen some, such as `log()`.

They are called with this format `output<-functionName(input)`.

- The input is called *parameters*, and there can be many parameters
- parameters are usually described in the documentation
- the output is what the function *returns*
- functions can only return 1 object, but this includes a list… so it could return many objects in the form of a list object

R has many, many functions, to learn more about a function type `?functionName` and the documentation will come up.

```
#A simple function
#here the function log is called, with the parameter 2, and the output is stored in the varia
x<-log(2)
x
```

```
[1] 0.6931472
```

```
#A more complicated function
#What are the parameters?
#not rep(a,n) gives a vector of size n where all elements are a
s<-sample(x=1:10,size=4,replace=TRUE,prob=rep(1/10,10))
s
```

```
[1] 9 6 7 3
```

We have seen other people's functions but we can also make our own! Let's see an example first:

```
#recall the dot product example...
dotProd=function(a,b){
  value<-sum(a*b)
  return(value)
}
#calling our function
dotProd(x,y)
```

```
[1] 10.39721
```

What exactly does this code say?

- We stored the function in the variable `dotProd`
- to tell the compiler we are creating a function, we use the keyword function
- we specify the parameters in round brackets ()
- we put the names of the parameters in the () only, not what data type we expect them to be
- inside curly brackets, we put the code that the function will run when it is called
- `return()` ends the function, and sends back the variable in the brackets

Back to built in functions… `R` is a statistical software, what does that mean? It already includes many common statistical functions! For most common distributions there are functions for the pdf, cdf, inverse cdf as well as one to get a sample from that distribution. The syntax is in the format: `dDistName(x,parameters)`, `pDistName(x,parameters)`, `qDistName(x,parameters)` and `rDistName(x,parameters)` respectively. This will make more sense in the example below…

```
#The normal distribution, sd is the standard deviation
#pdf
dnorm(c(2,3,5),mean=0,sd=1)
```

```
[1] 5.399097e-02 4.431848e-03 1.486720e-06
```

```
#cdf
pnorm(c(2,3,5),mean=0,sd=1)
```

```
[1] 0.9772499 0.9986501 0.9999997
```

```
#inverse cdf
qnorm(c(0.2,.5,.3),mean=0,sd=1)
```

```
[1] -0.8416212  0.0000000 -0.5244005
```

```
#random sample of size 10
rnorm(10,mean=0,sd=1)
```

```
 [1] -1.078915630  1.640992605  1.288852871  0.465496854  0.346615142
 [6] -0.007280197  0.536523310  0.115918794 -0.930134742  0.231840639
```

6.6 Plotting

R is very good for plotting! There are many types of plots in R, here are some useful plotting functions, this list is not exhaustive…

- `plot(x,y,...)` produces a scatter plot.
- `abline(a=intercept,b=slope,...)`
- `curve(expr,...)` evaluates an expression along a grid to create a curve
- `hist(data)` creates a histogram

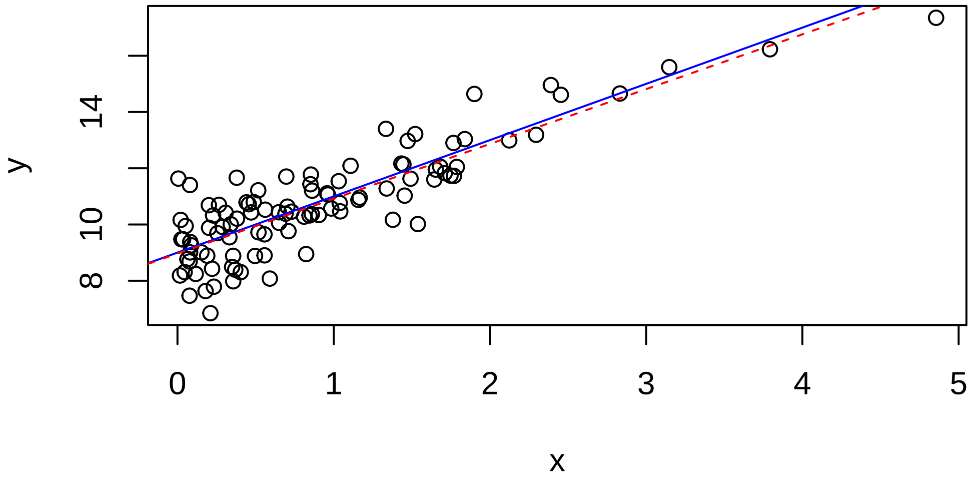
Plot functions have many parameters, some include `col` which changes the color and add which should be set to `TRUE` if the plot should be added to the existing plot. The best way to learn plots is with examples, I have included a regression example below.

```
#simulate errors
epsilon<-rnorm(100)
x<-rexp(100)
y<-9+2*x+epsilon

#scatter plot with true line
plot(x,y)
```

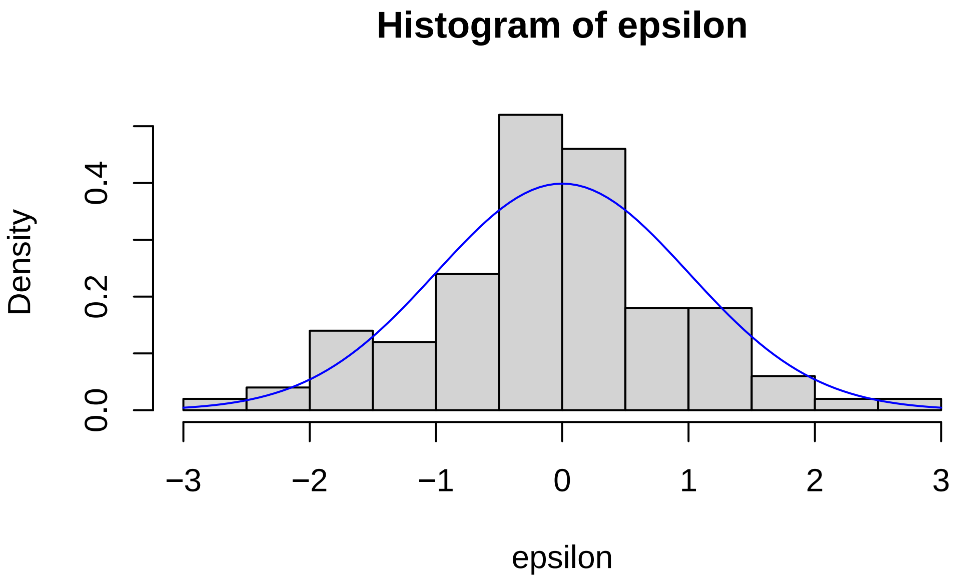
```
abline(a=9,b=2,col="blue")

#least squares line
lmm<-lm(y~x)
summary(lmm)
```

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.5326 -0.5414 -0.1160  0.5462  2.6534 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   8.9679     0.1434   62.54   <2e-16 ***
x             1.9484     0.1160   16.80   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9879 on 98 degrees of freedom
Multiple R-squared:  0.7423,	Adjusted R-squared:  0.7397 
F-statistic: 282.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
abline(lmm$coefficients[1],lmm$coefficients[2],col="red",lty=2)
```



```
#histogram of residuals
hist(epsilon,freq = F)
#x is what you want to evaluate the grid along
curve(dnorm(x),add=T,col="blue")
```



6.7 If Statements

If statements are essential in programming, and they are a form of 'Control Structure'. They take the form `if(boolean variable){some task}`.
When the computer runs through the code, it checks if the boolean value is `TRUE`, and if it is, it executes the code in the curly brackets, code in curly brackets is called a *block*. A simple example...

```
jim<-"nice"

if(jim=="nice"){
  alice="nice"
}
```

Placing an `else{some code}` after the if statement will execute the code in it's block if the code in the above if statement *was not* executed. The if and else must be in the same block so I have surrounded them in curly brackets.

```
jim<-"nice"
##same block
{
```

```
if(jim=="nice"){
  alice="nice"
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "nice"
```

```
jim<-"mean"
##same block
{
if(jim=="nice"){
  alice="nice"
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "not nice"
```

You may also use `else if(boolean){block}`, which executes it's block if the above (else) if statement(s) did not execute. See below:

```
jim<-"okay"
##same block
{
if(jim=="nice"){
  alice="nice"
}
else if(jim=="okay"){
  alice="okay"
}
  #Here if jim is not okay or nice, then we check if he is neutral.
else if(jim=="neutral"){
  alice="neutral"
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "okay"
```

Lastly you may put if statements inside of other if statements, called ‘nested ifs’.

```
jim<-"nice"
##same block

if(jim=="nice"){
  alice=sample(c("nice","not nice"),1)
  if(alice=="nice"){
    print(alice)
  }
  else{
    print(alice)
  }
}
```

```
[1] "not nice"
```

6.8 Loops

Loops execute operations within their blocks repeatedly. There are 2 types of loops you will generally use, for loops and while loops. For loops repeat the block a set number of times, while while loops repeat until a condition is satisfied. You can also nest loops, like if statements.

```
#calculate 2 to the power of ten
x<-1
#this reads for i in 1 to 10, this can be any vector that i loops through, not just a sequen
for(i in 1:10){
  x<-x*2
}

x
```

```
[1] 1024
```

```
for(i in 1:10){
  x<-x+i
}

vec=2:5

for(i in vec){
  x<-x+i
}


#calculate power of 2 less that 1000
x<-1
while(2*x<1000){
  x<-x*2
}
x
```

```
[1] 512
```

```
#nested loop
for(i in c(10,9,8,7,6,5,4,3,2,1)){
  v<-NULL
  for(j in 1:i){
    v<-c(v,"*")
  }
  print(v)
}
```

```
 [1] "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*"
[1] "*" "*" "*"
[1] "*" "*"
[1] "*"
```

You can also use the replicate function, which replicates a line of code a specified number of times. This gives a 10 by 5 matrix.

```
replicate(5,rnorm(10))
```

```
            [,1]       [,2]        [,3]        [,4]        [,5]
 [1,]  1.8318790 -1.8137875  0.02102309  0.24428684  1.38177652
 [2,]  1.0407765  1.0103691  0.90663829  0.02002666  0.57195969
 [3,]  1.2241047  2.4606678  1.63445520 -2.77294736 -0.53912585
 [4,]  1.1079609  0.7081370  0.64626458 -0.45401002 -0.44996537
 [5,]  0.1430367  0.1302675 -0.26482785  0.21581332  0.22938311
 [6,] -2.1642367 -1.5354584  0.11523552  0.96940943 -0.28116975
 [7,]  0.2416078 -0.5435865 -0.26965968  0.40618336 -1.52080350
 [8,] -0.1237735  0.2513472  0.76339807 -0.34903153  1.71814771
 [9,] -0.4793219  0.0456668  1.51237422  0.17672306  1.37363510
[10,]  1.9456363  0.2472481  0.46179049 -1.26769109 -0.01580745
```

Similar functions include `sapply()` and `apply()`. `sapply(X,FUN,...)` applies the function that the parameter `FUN` is set to to individual elements of a vector. `apply(X,MARGIN,FUN,...)` applies `FUN` to the rows or columns depending on what `MARGIN` is set to, 1 for rows and 2 for columns.

6.9 Coverage Probability Example

Here we generate 10000 samples of size 100 from the exponential distribution, with $\lambda = 2$. We calculate 10000 confidence intervals for $1/\lambda$ with \$ =\$1%, using the normal approximation:

$$\sqrt{n}(\bar{X} - 1/\lambda) \sim N(0, 1/\lambda^2)$$

and interval:

$$(\bar{X} - t_{99}(0.005) * S/\sqrt{n}, \bar{X} + t_{99}(0.005) * S/\sqrt{n})$$

We then check the proportion of intervals that contain the true value of $1/\lambda$.

```
#10000 samples, each of size 100 from the exponential distribution
x<-replicate(10000,rexp(100,rate=2))
#x is 100 by 10000, each column is a sample
dim(x)
```

```
[1]   100 10000
```

```
#calculate sample variances
S_Vector<-apply(x,2,sd);

# S_Vector

#get the t value
tval<-qt(1-0.005,99)
#calculate the means

means<-apply(x,2,mean); length(means)
```

```
[1] 10000
```

```
# lower and upper bounds
lower<-means-S_Vector*tval/10
upper<-means+S_Vector*tval/10
intervals<-rbind(lower,upper)
#example interval
intervals[,1]
```

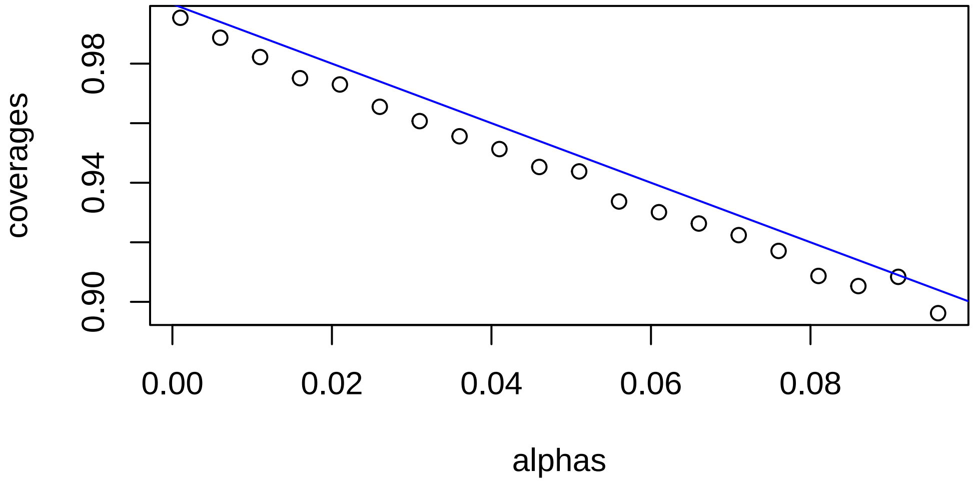
```
    lower     upper
0.3962586 0.6815397
```

```
#we now check each interval to see if it contains the mean
successes<-0
for(i in 1:ncol(intervals)){
  #if 0.5 is in the interval, add 1
  if((intervals[1,i]<0.5)&&(intervals[2,i]>0.5))
    successes<-successes+1
}
#here is the coverage probability...
coverage.prob<-successes/ncol(intervals)
coverage.prob
```

```
[1] 0.9824
```

Something more advanced...

```
#Vectorzing the function changes the way the function calculates when it is a passed a vector
#it will run the function once per element if it is vectorized instead of passing the vector
getCovProb<-Vectorize(function(alpha){
#10000 samples, each of size 100 from the exponential distribution
  x<-replicate(10000,rexp(100,rate=2))
#x is 100 by 10000, each column is a sample
  dim(x)
#calculate sample variances
  S_Vector<-apply(x,2,sd)
#get the t value
  tval<-qt(1-alpha/2,99)
#calculate the means
  means<-apply(x,2,mean)
# lower and upper bounds
  lower<-means-S_Vector*tval/10
  upper<-means+S_Vector*tval/10
  intervals<-rbind(lower,upper)
#example interval
  intervals[,1]

#we now check each interval to see if it contains the mean
successes<-0
for(i in 1:ncol(intervals)){
  #if 0.5 is in the interval, add 1
  if((intervals[1,i]<0.5)&(intervals[2,i]>0.5))
    successes<-successes+1
}
#here is the coverage probability...
coverage.prob<-successes/ncol(intervals)
return(coverage.prob)
})
#here we find the coverage probability for many alphas
alphas<-seq(from=0.001,to=0.1,by=0.005)
coverages<-getCovProb(alphas)
#adds a scatter plot
plot(alphas,coverages)
#adds a line
abline(a=1,b=-1,col="blue")
```



For more information you can visit here . It is also very easy to find tutorials on the web (Youtube is good), you could also look at the book by Lafaye, Drouilhet and Liquet (2013).