Introduction to linear models

Olga Dethlefsen

Table of contents

Preface

Linear models allows us to answer questions such as:

- is there a relationship between exposure and outcome, e.g. body weight and plasma volume?
- how strong is the relationship between the two variables?
- what will be a predicted value of the outcome given a new set of exposure values?
- how accurately can we predict outcome?
- which variables are associated with the response, e.g. is it body weight and height that can explain the plasma volume or is it just the body weight?

Learning outcomes

- to understand what a linear model is and be familiar with the terminology
- to be able to state linear model in the general vector-matrix notation
- to be able to use the general vector-matrix notation to numerically estimate model parameters
- to be able to use `lm()` function for model fitting, parameter estimation, hypothesis testing and prediction

Do you see a mistake or a typo? I would be grateful if you let me know via olga.dethlefsen@nbis.se

This repository contains teaching and learning materials prepared and used during "Introduction to biostatistics and machine learning" course, organized by NBIS, National Bioinformatics Infrastructure Sweden. The course is open for PhD students, postdoctoral researcher and other employees within Swedish universities. The materials are geared towards life scientists wanting to be able to understand and use basic statistical and machine learning methods. More about the course https://nbisweden.github.io/workshop-mlbiostatistics/

1 Introduction to linear models

1.1 Why linear models?

With linear models we can answer questions such as:

- is there a relationship between exposure and outcome, e.g. body weight and plasma volume?
- how strong is the relationship between the two variables?
- what will be a predicted value of the outcome given a new set of exposure values?
- how accurately can we predict outcome?
- which variables are associated with the response, e.g. is it body weight and height that can explain the plasma volume or is it just the body weight?

1.2 Statistical vs. deterministic relationship

Relationships in probability and statistics can generally be one of three things: deterministic, random, or statistical:

- a **deterministic** relationship involves **an exact relationship** between two variables, for instance Fahrenheit and Celsius degrees is defined by an equation $Fahrenheit = \frac{9}{5} \cdot Celcius + 32$
- there is **no relationship** between variables in the **random relationship**, for instance number of succulents Olga buys and time of the year as Olga keeps buying succulents whenever she feels like it throughout the entire year
- **a statistical relationship** is a **mixture of deterministic and random relationship**, e.g. the savings that Olga has left in the bank account depend on Olga's

monthly salary income (deterministic part) and the money spent on buying succulents (random part)

1.3 What linear models are and are not

- A linear model is one in which the parameters appear linearly in the deterministic part of the model
- e.g. **simple linear regression** through the origin is a simple linear model of the form

  $$Y_i = \beta x + \epsilon$$

  often used to express a relationship of **one numerical variable to another**, e.g. the calories burnt and the kilometers cycled
- linear models can become quite advanced by including **many variables**, e.g. the calories burnt could be a function of the kilometers cycled, road incline and status of a bike, or the **transformation of the variables**, e.g. a function of kilometers cycled squared

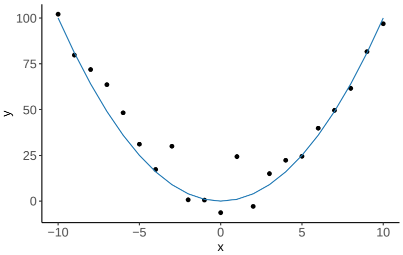


Figure 1.2: Example of a linear model: $y_i = x_i^2 + e_i$ showing that linear models can capture more than a straight line relationship

More examples where model parameters appear linearly:

- $Y_i = \alpha + \beta x_i + \gamma y_i + \epsilon_i$
- $Y_i = \alpha + \beta x_i^2 \epsilon$
- $Y_i = \alpha + \beta x_i^2 + \gamma x_i^3 + \epsilon$
- $Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 y_i + \beta_4 \sqrt{y_i} + \beta_5 x_i y_i + \epsilon$

and an example on a non-linear model where parameter β appears in the exponent of x_i

- $Y_i = \alpha + x_i^{\beta} + \epsilon$

1.4 Terminology

There are many terms and notations used interchangeably:

- y is being called:
  - response

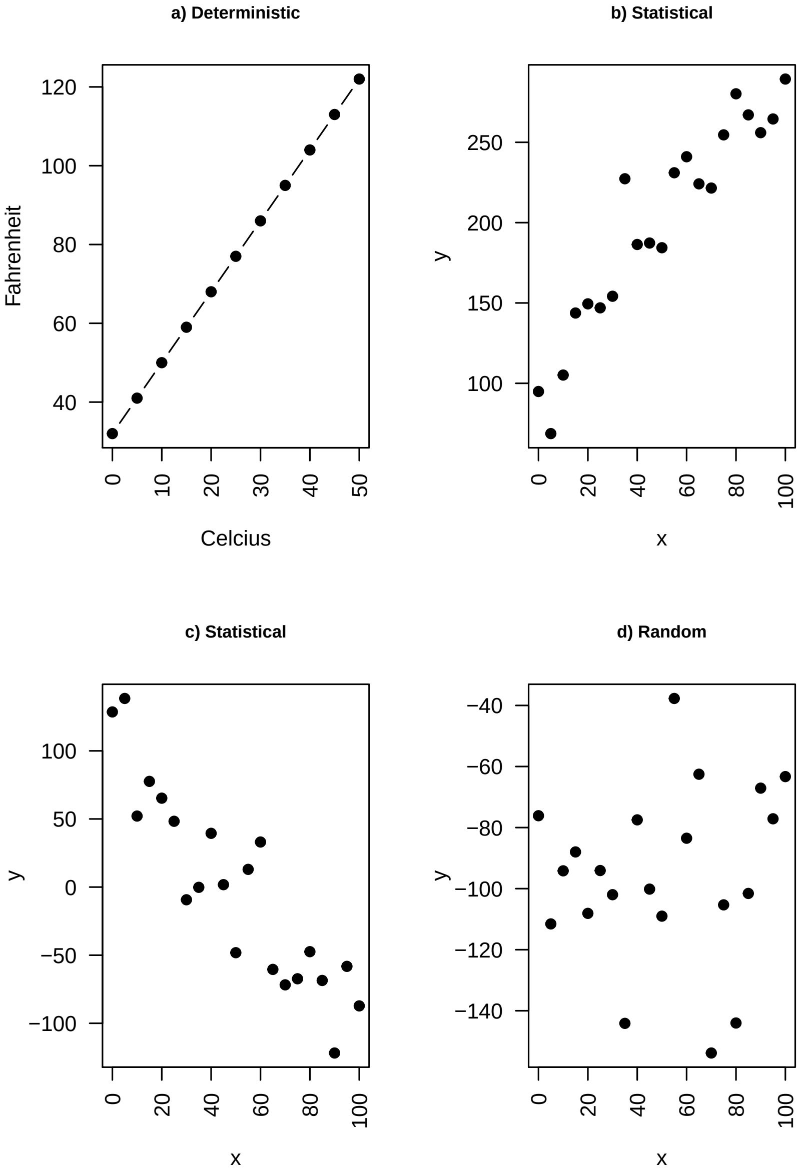


Figure 1.1: Deterministic vs. statistical relationship: a) deterministic: equation exactly describes the relationship between the two variables e.g. Ferenheit and Celcius relationship, b) statistical relationship between x and y is not perfect (increasing relationship), c) statistical relationship between x and y is not perfect (decreasing relationship), d) random signal

    - outcome
    - dependent variable

- x is being called:

    - exposure
    - explanatory variable
    - dependent variable
    - predictor
    - covariate

1.5 Simple linear regression

- It is used to check the association between **the numerical outcome and one numerical explanatory variable**
- In practice, we are finding the best-fitting straight line to describe the relationship between the outcome and exposure

Example 1.1 (Weight and plasma volume)**.** Let's look at the example data containing body weight (kg) and plasma volume (liters) for eight healthy men to see what the best-fitting straight line is.

Example data:

```r
weight <- c(58, 70, 74, 63.5, 62.0, 70.5, 71.0, 66.0) # body weight (kg)
plasma <- c(2.75, 2.86, 3.37, 2.76, 2.62, 3.49, 3.05, 3.12) # plasma volume (liters)
```

The equation for the red line is:

$$Y_i = 0.086 + 0.044 \cdot x_i \quad for\ i = 1 \ldots 8$$

and in general:

$$Y_i = \alpha + \beta \cdot x_i \quad for\ i = 1 \ldots n$$

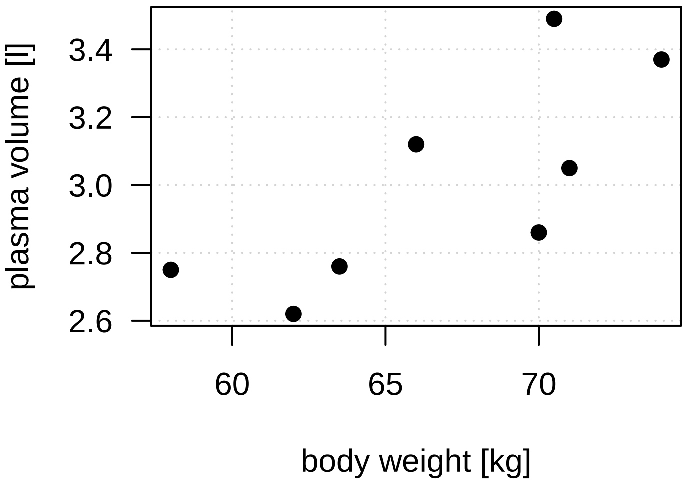


Figure 1.3: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*.

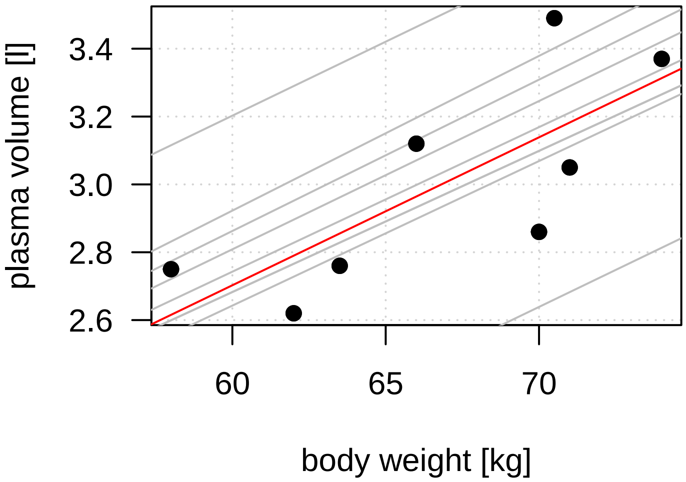


Figure 1.4: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regression gives the equation of the straight line (red) that best describes how the outcome changes (increase or decreases) with a change of exposure variable

- In other words, by finding the best-fitting straight line we are **building a statistical model** to represent the relationship between plasma volume (Y) and explanatory body weight variable (x)
- If we were to use our model $Y_i = 0.086 + 0.044 \cdot x_i$ to find plasma volume given a weight of 58 kg (our first observation, $i = 1$), we would notice that we would get $Y = 0.086 + 0.044 \cdot 58 = 2.638$, not exactly 2.75 as we have for our first man in our dataset that we started with, i.e. $2.75 - 2.638 = 0.112 \neq 0$.
- We thus add to the above equation an **error term** to account for this and now we can write our **simple regression model** more formally as:

$$Y_i = \alpha + \beta \cdot x_i + \epsilon_i \tag{1.1}$$

where:

- x: is called: exposure variable, explanatory variable, dependent variable, predictor, covariate
- y: is called: response, outcome, dependent variable
- α and β are **model coefficients**
- and ϵ_i is an **error terms**

1.6 Least squares

- in the above **"body weight - plasma volume"** example, the values of α and β have just appeared
- in practice, α and β values are unknown and we use data to **estimate these coefficients**, noting the estimates with a **hat**, $\hat{\alpha}$ and $\hat{\beta}$
- **least squares** is one of the methods of parameters estimation, i.e. finding $\hat{\alpha}$ and $\hat{\beta}$

Let $\hat{y_i} = \hat{\alpha} + \hat{\beta}x_i$ be the prediction y_i based on the i-th value of x:

- Then $\epsilon_i = y_i - \hat{y_i}$ represents the i-th **residual**, i.e. the difference between the i-th observed response value and

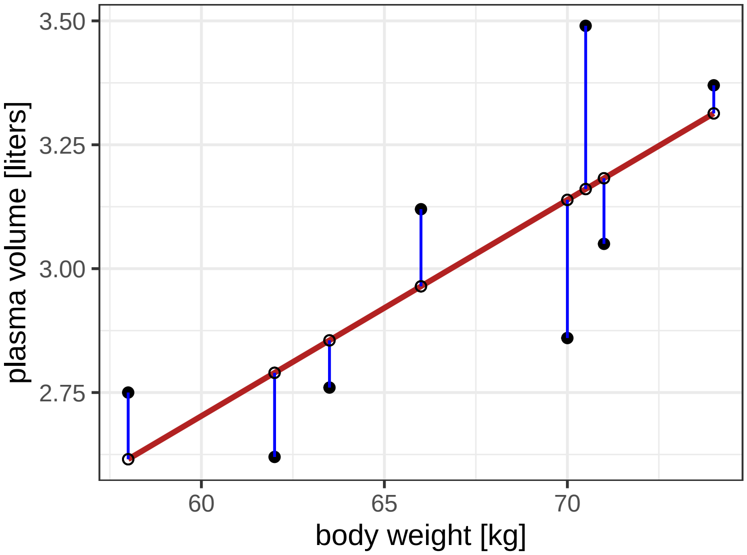


Figure 1.5: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regrssion gives the equation of the straight line (red) that best describes how the outcome changes with a change of exposure variable. Blue lines represent error terms, the vertical distances to the regression line

the i-th response value that is predicted by the linear model

- RSS, the **residual sum of squares** is defined as:

$$RSS = \epsilon_1^2 + \epsilon_2^2 + \cdots + \epsilon_n^2$$

  or equivalently as:

$$RSS = (y_1 - \hat{\alpha} - \hat{\beta}x_1)^2 + (y_2 - \hat{\alpha} - \hat{\beta}x_2)^2 + ... + (y_n - \hat{\alpha} - \hat{\beta}x_n)^2$$

- the least squares approach chooses $\hat{\alpha}$ and $\hat{\beta}$ **to minimize the RSS**. With some calculus, a good video explanation for the interested ones is here, we get Theorem 1.1

Theorem 1.1 (Least squares estimates for a simple linear regression)**.**

$$\hat{\beta} = \frac{S_{xy}}{S_{xx}}$$

$$\hat{\alpha} = \bar{y} - \frac{S_{xy}}{S_{xx}} \cdot \bar{x}$$

where:

- $\bar{x}$*: mean value of* x
- $\bar{y}$*: mean value of* y
- S_{xx}*: sum of squares of* X *defined as* $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$
- S_{yy}*: sum of squares of* Y *defined as* $S_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2$
- S_{xy}*: sum of products of* X *and* Y *defined as* $S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$

We can further re-write the above sum of squares to obtain

- sum of squares of X,

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n})$$

- sum of products of X and Y

$$S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}$$

Example 1.2 (Least squares)**.** Let's try least squares method to find coefficient estimates in the **"body weight and plasma volume example"**

```r
# initial data
weight <- c(58, 70, 74, 63.5, 62.0, 70.5, 71.0, 66.0) # body weight (kg)
plasma <- c(2.75, 2.86, 3.37, 2.76, 2.62, 3.49, 3.05, 3.12) # plasma volume (liters)

# rename variables for convenience
x <- weight
y <- plasma

# mean values of x and y
x.bar <- mean(x)
y.bar <- mean(y)

# Sum of squares
Sxx <-  sum((x - x.bar)^2)
Sxy <- sum((x-x.bar)*(y-y.bar))

# Coefficient estimates
beta.hat <- Sxy / Sxx
alpha.hat <- y.bar - Sxy/Sxx*x.bar

# Print estimated coefficients alpha and beta
print(alpha.hat)
```

```
[1] 0.08572428
```

```r
print(beta.hat)
```

```
[1] 0.04361534
```

In R we can use `lm()`, the built-in function, to fit a linear regression model and we can replace the above code with one line

```
lm(plasma ~ weight)
```

```

Call:
lm(formula = plasma ~ weight)

Coefficients:
(Intercept)       weight
    0.08572      0.04362
```

1.7 Intercept and Slope

- Linear regression gives us estimates of model coefficient $Y_i = \alpha + \beta x_i + \epsilon_i$
- α is known as the **intercept**
- β is known as the **slope**

1.8 Hypothesis testing

Is there a relationship between the response and the predictor?

- the calculated $\hat{\alpha}$ and $\hat{\beta}$ are **estimates of the population values** of the intercept and slope and are therefore subject to **sampling variation**
- their precision is measured by their **estimated standard errors**, `e.s.e`$(\hat{\alpha})$ and `e.s.e`$(\hat{\beta})$
- these estimated standard errors are used in **hypothesis testing** and in constructing **confidence and prediction intervals**

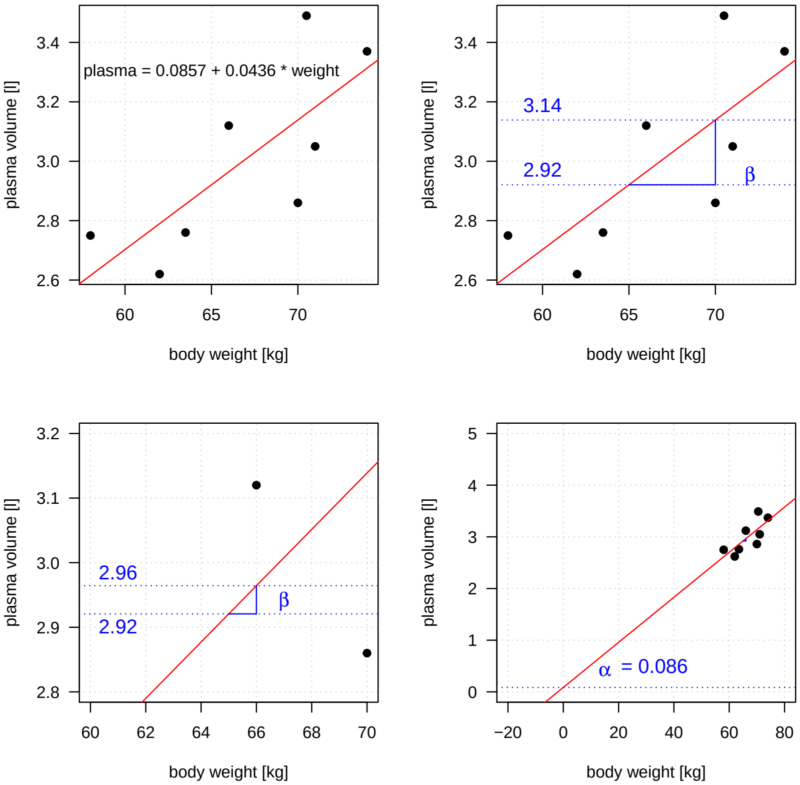


Figure 1.6: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regression gives the equation of the straight line that best describes how the outcome changes (increase or decreases) with a change of exposure variable (in red)

The most common hypothesis test involves testing the `null hypothesis` of:

- H_0 : There is no relationship between X and Y
- versus the `alternative hypothesis` H_a : there is some relationship between X and Y

Mathematically, this corresponds to testing:

- $H_0 : \beta = 0$
- versus $H_a : \beta \neq 0$
- since if $\beta = 0$ then the model $Y_i = \alpha + \beta x_i + \epsilon_i$ reduces to $Y = \alpha + \epsilon_i$

t-statistics

$$\frac{\hat{\beta} - \beta}{e.s.e(\hat{\beta})} \sim t(n-p)$$

Student's t distribution

Under the null hypothesis $H_0 : \beta = 0$

- n is number of observations
- p is number of model parameters
- $\frac{\hat{\beta}-\beta}{e.s.e(\hat{\beta})}$ is the ratio of the departure of the estimated value of a parameter, $\hat{\beta}$, from its hypothesized value, β, to its standard error, called `t-statistics`
- the `t-statistics` follows Student's t distribution with $n - p$ degrees of freedom

Example 1.3 (Hypothesis testing)**.** Let's look again at our example data. This time we will not only fit the linear regression model but also look a bit more closely at the `R` `summary` of the model

```
weight <- c(58, 70, 74, 63.5, 62.0, 70.5, 71.0, 66.0) # body weight (kg)
plasma <- c(2.75, 2.86, 3.37, 2.76, 2.62, 3.49, 3.05, 3.12) # plasma volume (liters)

model <- lm(plasma ~ weight)
print(summary(model))
```

```
Call:
lm(formula = plasma ~ weight)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27880 -0.14178 -0.01928  0.13986  0.32939 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.08572    1.02400   0.084   0.9360  
weight       0.04362    0.01527   2.857   0.0289 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2188 on 6 degrees of freedom
Multiple R-squared:  0.5763,    Adjusted R-squared:  0.5057 
F-statistic:  8.16 on 1 and 6 DF,  p-value: 0.02893
```

- Under `Estimate` we see estimates of our model coefficients, $\hat{\alpha}$ (intercept) and $\hat{\beta}$ (slope, here weight), followed by their estimated standard errors, `Std. Errors`
- If we were to test if there is an **association between weight and plasma volume** we would write under the null hypothesis $H_0 : \beta = 0$

$$\frac{\hat{\beta} - \beta}{e.s.e(\hat{\beta})} = \frac{0.04362 - 0}{0.01527} = 2.856582$$

- and we would **compare** `t-statistics` to `Student's t distribution` with $n - p = 8 - 2 = 6$ degrees of freedom (as we have 8 observations and two model parameters, α and β)
- we can use **Student's t distribution table** or **R code** to obtain the associated P-value

```r
2*pt(2.856582, df=6, lower=F)
```

```
[1] 0.02893095
```

- here the observed t-statistics is large and therefore yields a small P-value, meaning that **there is sufficient evidence to reject null hypothesis in favor of the alternative** and conclude that there is a significant association between weight and plasma volume

1.9 Vector-matrix notations

While in simple linear regression it is feasible to arrive at the parameters estimates using calculus in more realistic settings of **multiple regression**, with more than one explanatory variable in the model, it is **more efficient to use vectors and matrices to define the regression model**.

Let's **rewrite** our simple linear regression model $Y_i = \alpha + \beta_i + \epsilon_i \quad i = 1, \dots n$ **into vector-matrix notation** in **6 steps**.

1. First we rename our α to β_0 and β to β_1 as it is easier to keep tracking the number of model parameters this way
2. Then we notice that we actually have n equations such as:

$$y_1 = \beta_0 + \beta_1 x_1 + \epsilon_1$$

$$y_2 = \beta_0 + \beta_1 x_2 + \epsilon_2$$

$$y_3 = \beta_0 + \beta_1 x_3 + \epsilon_3$$

$$\dots$$

$$y_n = \beta_0 + \beta_1 x_n + \epsilon_n$$

3. We group all Y_i and ϵ_i into column vectors: $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ and $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$
4. We stack two parameters β_0 and β_1 into another column vector:
$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$
5. We append a vector of ones with the single predictor for each i and create a matrix with two columns called **design matrix**
$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$
6. We write our linear model in a vector-matrix notations as:
$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

Definition 1.1 (vector matrix form of the linear model)**.** The vector-matrix representation of a linear model with $p - 1$ predictors can be written as

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where:

- $\mathbf{Y}$ is $n \times 1$ vector of observations
- $\mathbf{X}$ is $n \times p$ **design matrix**
- β is $p \times 1$ vector of parameters

- ϵ is $n \times 1$ vector of vector of random errors, indepedent and identically distributed (i.i.d) N(0, σ^2)

In full, the above vectors and matrix have the form:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & ... & x_{1,p-1} \\ 1 & x_{2,1} & ... & x_{2,p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & ... & x_{n,p-1} \end{bmatrix}$$

Theorem 1.2 (Least squares in vector-matrix notation)**.** *The least squares estimates for a linear regression of the form:*

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

is given by:

$$\hat{} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Example 1.4 (vector-matrix-notation)**.** Following the above definition we can write the **"weight - plasma volume model"** as:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where:

$$\mathbf{Y} = \begin{bmatrix} 2.75 \\ 2.86 \\ 3.37 \\ 2.76 \\ 2.62 \\ 3.49 \\ 3.05 \\ 3.12 \end{bmatrix}$$

$$\beta = \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} \; \epsilon = \begin{bmatrix}\epsilon_1\\ \epsilon_2\\ \vdots\\ \epsilon_8\end{bmatrix} \; \mathbf{X} = \begin{bmatrix}1 & 58.0\\ 1 & 70.0\\ 1 & 74.0\\ 1 & 63.5\\ 1 & 62.0\\ 1 & 70.5\\ 1 & 71.0\\ 1 & 66.0\end{bmatrix}$$

and we can estimate model parameters using ˆ $= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

We can do it by hand or in `R` as follows:

```
n <- length(plasma) # no. of observation
Y <- as.matrix(plasma, ncol=1)
X <- cbind(rep(1, length=n), weight)
X <- as.matrix(X)

# print Y and X to double-check that the format is according to the definition
print(Y)
```

```
     [,1]
[1,] 2.75
[2,] 2.86
[3,] 3.37
[4,] 2.76
[5,] 2.62
[6,] 3.49
[7,] 3.05
[8,] 3.12
```

```
print(X)
```

```
       weight
[1,] 1   58.0
[2,] 1   70.0
[3,] 1   74.0
[4,] 1   63.5
[5,] 1   62.0
```

```
[6,] 1   70.5
[7,] 1   71.0
[8,] 1   66.0
```

```
# least squares estimate
# solve() finds inverse of matrix
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y
print(beta.hat)
```

```
             [,1]
       0.08572428
weight 0.04361534
```

1.10 Confidence intervals and prediction intervals

- when we estimate coefficients we can also find their **confidence intervals**, typically 95% confidence intervals, i.e. a range of values that contain the true unknown value of the parameter
- we can also use linear regression models to predict the response value given a new observation and find **prediction intervals**. Here, we look at any specific value of x_i, and find an interval around the predicted value y_i' for x_i such that there is a 95% probability that the real value of y (in the population) corresponding to x_i is within this interval

::: {exm-prediction-and-intervals} ## prediction and intervals

Let's:

- find confidence intervals for our coefficient estimates
- predict plasma volume for a men weighting 60 kg
- find prediction interval
- plot original data, fitted regression model, predicted observation and prediction interval

```r
# fit regression model
model <- lm(plasma ~ weight)
print(summary(model))
```

```
Call:
lm(formula = plasma ~ weight)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27880 -0.14178 -0.01928  0.13986  0.32939

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.08572    1.02400   0.084   0.9360
weight       0.04362    0.01527   2.857   0.0289 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2188 on 6 degrees of freedom
Multiple R-squared:  0.5763,    Adjusted R-squared:  0.5057
F-statistic:  8.16 on 1 and 6 DF,  p-value: 0.02893
```

```r
# find confidence intervals for the model coefficients
confint(model)
```

```
                   2.5 %     97.5 %
(Intercept) -2.419908594 2.59135716
weight       0.006255005 0.08097567
```

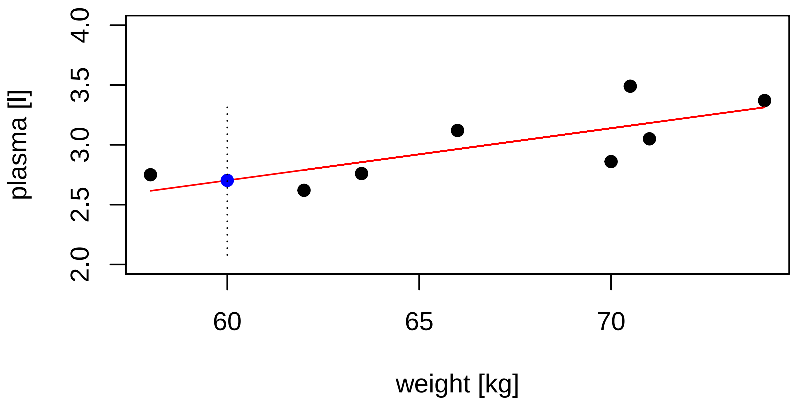
```r
# predict plasma volume for a new observation of 60 kg
# we have to create data frame with a variable name matching the one used to build the model
new.obs <- data.frame(weight = 60)
predict(model, newdata = new.obs)
```

```
       1
2.702645
```

```r
# find prediction intervals
prediction.interval <- predict(model, newdata = new.obs,  interval = "prediction")
print(prediction.interval)
```

```
       fit      lwr      upr
1 2.702645 2.079373 3.325916
```

```r
# plot the original data, fitted regression and predicted value
plot(weight, plasma, pch=19, xlab="weight [kg]", ylab="plasma [l]", ylim=c(2,4))
lines(weight, model$fitted.values, col="red") # fitted model in red
points(new.obs, predict(model, newdata = new.obs), pch=19, col="blue") # predicted value at
segments(60, prediction.interval[2], 60, prediction.interval[3], lty = 3) # add prediction i
```



Exercises (introduction to linear models)

Data for exercises are on Canvas under Files -> data_exercises –> linear-models

Exercise 1.1 (Protein levels in pregnancy)**.** The researchers were interested whether protein levels in expectant mothers are changing throughout the pregnancy. Observations have been taken on 19 healthy women and each woman was at different stage of pregnancy (gestation).

Assuming linear model:

- $Y_i = \alpha + \beta x_i + \epsilon_i$, where Y_i corresponds to protein levels in i-th observation

and taking summary statistics:

- $\sum_{i=1}^{n} x_i = 456$
- $\sum_{i=1}^{n} x_i^2 = 12164$
- $\sum_{i=1}^{n} x_i y_i = 369.87$
- $\sum_{i=1}^{n} y_i = 14.25$
- $\sum_{i=1}^{n} y_i^2 = 11.55$

a) find the least square estimates of $\hat{\alpha}$ and $\hat{\beta}$
b) knowing that e.s.e$(\hat{\beta}) = 0.003295$

can we:

- i) reject the null hypothesis that the is no relationship between protein level and gestation, i.e. perform a hypothesis test to test $H_0 : \beta = 0$;

- ii) can we reject the null hypothesis that $\beta = 0.02$, i.e. perform a hypothesis test to test $H_0 : \beta = 0.02$

c) write down the linear model in the vector-matrix notation and identify response, parameter, design and error matrices
d) read in "protein.csv" data into R, set Y as protein (response) and calculate using matrix functions the least squares estimates of model coefficients
e) use `lm()` function in R to check your calculations
f) use the fitted model in R to predict the value of protein levels at week 20. Try plotting the data, fitted linear model and the predicted value to assess whether your prediction is to be expected.

Exercise 1.2 (Glucose levels in potatoes)**.** The glucose level in potatoes depends on their storage time and the relationship is somehow curvilinear as shown below. As we believe that the quadratic function might describe the relationship, assume linear model in form $Y_i = \alpha + \beta x_i + \gamma x_i^2 + \epsilon_i \quad i = 1, \ldots, n$ where $n = 14$ and

a) write down the model in vector-matrix notation
b) load data from "potatoes.csv" and use least squares estimates to obtain estimates of model coefficients
c) use `lm()` function to verify your calculations
d) perform a hypothesis test to test $H_0 : \gamma = 0$; and comment whether there is a significant quadratic relationship
e) predict glucose concentration at storage time 4 and 16 weeks. Plot the data, the fitted model and the predicted values

```
data.potatoes <- read.csv("data/lm/potatoes.csv")
plot(data.potatoes$Weeks, data.potatoes$Glucose, pch=19, xlab="Storage time [weeks]", ylab="
```

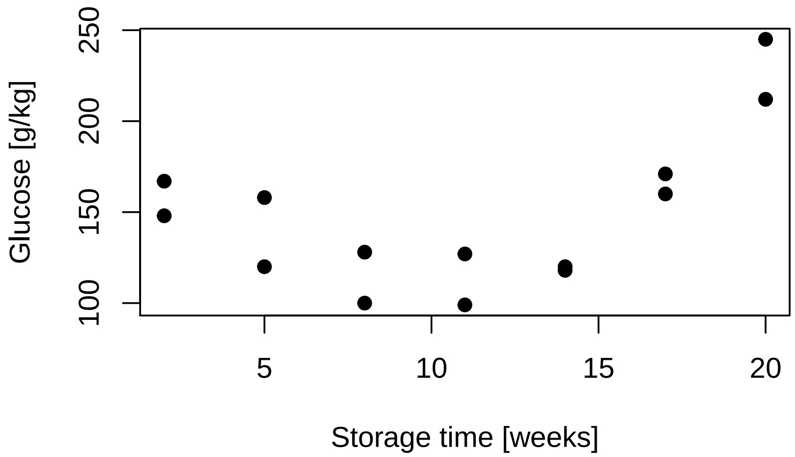


Figure 1.7: Sugar in potatoes: relationship between storage time and glucose content

Exercise 1.3 (Linear models form)**.** Which of the following models are linear models and why?

a) $Y_i = \alpha + \beta x_i + \epsilon_i$
b) $Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i$
c) $Y_i = \alpha + \beta x_i + \gamma x_i^2 + \epsilon_i$
d) $Y_i = \alpha + \gamma x_i^{\beta} + \epsilon_i$

Answers to selected exercises

Solution. Exercise 1.1

a)

- $S_{xx} = \sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n} = 12164 - \frac{456^2}{19} = 1220$
- $S_{xy} = \sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n} = 369.87 - \frac{(456 \cdot 14.25)}{19} = 27.87$
- $\hat{\beta} = \frac{S_{xy}}{S_{xx}} = 27.87/1220 = 0.02284$

- $\hat{\alpha} = \bar{y} - \frac{S_{xy}}{S_{xx}} \cdot \bar{x} = \frac{14.25}{19} - \frac{27.87}{1220} \cdot \frac{456}{19} = 0.20174$

b) i.

We can calculate test statistics following:

- $\frac{\hat{\beta}-\beta}{e.s.e(\hat{\beta})} \sim t(n-p) = \frac{0.02284-0}{0.003295} = 6.934$ where the value follows Student's t distribution with $n-p = 19-2 = 17$ degrees of freedom. We can now estimate the a p-value using Student's t distribution table or use R function

```
2*pt(6.934, df=17, lower=F)
```

```
[1] 2.414315e-06
```

As p-value « 0.001 there is sufficient evidence to reject H_0 in favor of H_1, thus we can conclude that there is a significant relationship between protein levels and gestation

b) ii.

Similarly, we can test $H_0 : \beta = 0.02$, i.e. $\frac{\hat{\beta}-\beta}{e.s.e(\hat{\beta})} \sim t(n-p) = \frac{0.02284-0.02}{0.20174} = 0.01407753$. Now the test statistics is small

```
2*pt(0.01407753, df=17, lower=F)
```

```
[1] 0.988932
```

p-value is large and hence there is no sufficient evidence to reject H_0 and we can conclude that $\beta = 0.02$

c) We can rewrite the linear model in vector-matrix formation as $\mathbf{Y} = \ \mathbf{X} + \ $ where:

response $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{19} \end{bmatrix}$

parameters $\beta = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$

design matrix $\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{19} \end{bmatrix}$

errors $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{19} \end{bmatrix}$

d) The least squares estimates in vector-matrix notation is $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ and we can calculate this in R

```r
# read in data
data.protein <- read.csv("data/lm/protein.csv")

# print out top observations
head(data.protein)
```

```
  Protein Gestation
1    0.38        11
2    0.58        12
3    0.51        13
4    0.38        15
5    0.58        17
6    0.67        18
```

```r
# define Y and X matrices given the data
n <- nrow(data.protein) # nu. of observations
Y <-  as.matrix(data.protein$Protein, ncol=1) # response
X <-  as.matrix(cbind(rep(1, length=n), data.protein$Gestation)) # design matrix
head(X) # double check that the design matrix looks like it should
```

```
     [,1] [,2]
[1,]    1   11
[2,]    1   12
[3,]    1   13
[4,]    1   15
[5,]    1   17
[6,]    1   18
```

```
# least squares estimate
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y # beta.hat is a matrix that contains our alpha and be
print(beta.hat)
```

```
           [,1]
[1,] 0.20173770
[2,] 0.02284426
```

e) We use `lm()` function to check our calculations

```
# fit linear regression model and print model summary
protein <- data.protein$Protein # our Y
gestation <- data.protein$Gestation # our X

model <- lm(protein ~ gestation)
print(summary(model))
```

```

Call:
lm(formula = protein ~ gestation)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.16853 -0.08720 -0.01009  0.08578  0.20422 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.201738   0.083363   2.420    0.027 *  
gestation   0.022844   0.003295   6.934 2.42e-06 ***
---
```

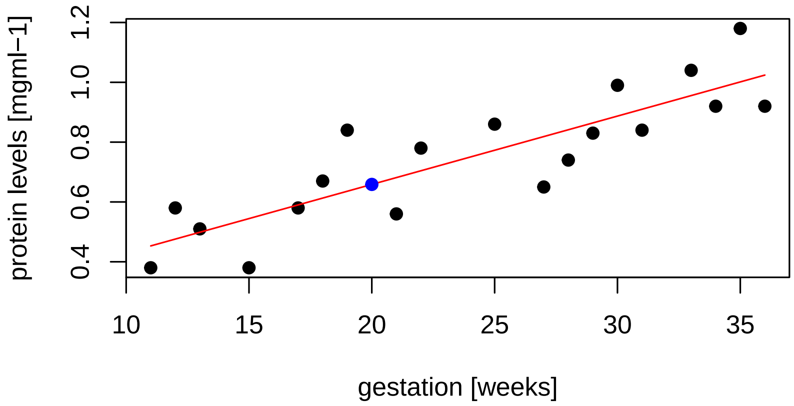
```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1151 on 17 degrees of freedom
Multiple R-squared:  0.7388,    Adjusted R-squared:  0.7234
F-statistic: 48.08 on 1 and 17 DF,  p-value: 2.416e-06
```

f)

```
new.obs <- data.frame(gestation = 20)
y.pred <- predict(model, newdata = new.obs)

# we can visualize the data, fitted linear model (red), and the predicted value (blue)
plot(gestation, protein, pch=19, xlab="gestation [weeks]", ylab="protein levels [mgml-1]")
lines(gestation, model$fitted.values, col="red")
points(new.obs, y.pred, col="blue", pch=19, cex = 1)
```



Solution. Exercise 1.2

a) We can rewrite the linear model in vector-matrix formation as

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where: response $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{14} \end{bmatrix}$

parameters $\beta = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$

design matrix $\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{14} & x_{14}^2 \end{bmatrix}$

errors $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{14} \end{bmatrix}$

b) load data to from “potatoes.csv” and use least squares estimates for obtain estimates of model coefficients

```r
data.potatoes <- read.csv("data/lm/potatoes.csv")

# define matrices
n <- nrow(data.potatoes)
Y <-  data.potatoes$Glucose
X1 <- data.potatoes$Weeks
X2 <- (data.potatoes$Weeks)^2
X <- cbind(rep(1, length(n)), X1, X2)
X <- as.matrix(X)

# least squares estimate
# beta here refers to the matrix of model coefficients incl. alpha, beta and gamma
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y
print(beta.hat)
```

```
         [,1]
   200.169312
X1 -19.443122
X2   1.030423
```

c) we use `lm()` function to verify our calculations:

```r
model <- lm(Y ~ X1 + X2)
print(summary(model))
```

```
Call:
lm(formula = Y ~ X1 + X2)

Residuals:
    Min      1Q  Median      3Q     Max 
-17.405 -11.250  -8.071  12.911  29.286 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 200.1693    15.0527  13.298 4.02e-08 ***
X1          -19.4431     3.1780  -6.118 7.54e-05 ***
X2            1.0304     0.1406   7.329 1.49e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.4 on 11 degrees of freedom
Multiple R-squared:  0.8694,    Adjusted R-squared:  0.8457 
F-statistic: 36.61 on 2 and 11 DF,  p-value: 1.373e-05
```

d) perform a hypothesis test to test $H_0 : \gamma = 0$; and comment whether we there is a significant quadratic term

- $\frac{\hat{\gamma}-\gamma}{e.s.e(\hat{\gamma})} \sim t(n-p) = \frac{1.030423-0}{0.1406} = 7.328755$ where the value follows Student's t distribution with $n-p = 19-2 = 17$ degrees of freedom. We can now estimate the a p-value using Student's t distribution table or use a function in R

```
2*pt(7.328755, df=14-3, lower=F)
```

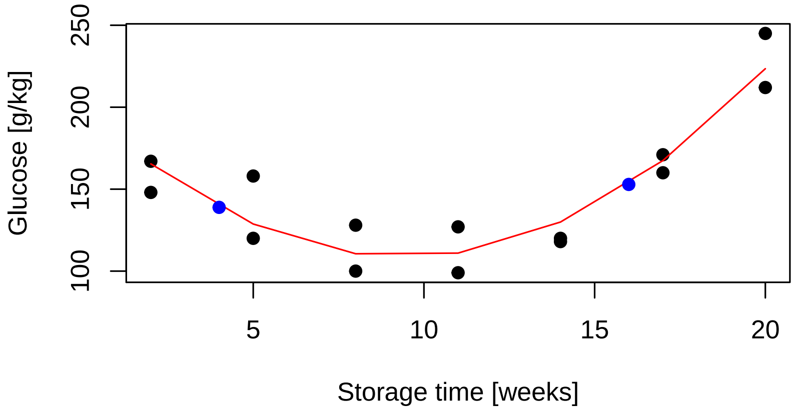
```
[1] 1.487682e-05
```

As p-value « 0.001 there is sufficient evidence to reject H_0 in favor of H_1, thus we can conclude that there is a significant quadratic relationship between glucose and storage time

e) predict glucose concentration at storage time 4 and 16 weeks

```
new.obs <- data.frame(X1 = c(4, 16), X2 = c(4^2, 16^2))
pred.y <- predict(model, newdata = new.obs)

plot(data.potatoes$Weeks, data.potatoes$Glucose, xlab="Storage time [weeks]", ylab="Glucose
lines(data.potatoes$Weeks, model$fitted.values, col="red")
points(new.obs[,1], pred.y, pch=19, col="blue")
```



2 Regression coefficients

Aims

- to clarify the interpretation of the fitted linear models

Learning outcomes

- to use `lm()` function to fit multiple linear regression model
- to be able to interpret the output of the model
- to be able to use `lm()` function to check for association between variables, group effects and interaction terms

2.1 Interpreting and using linear regression models

- In previous section we have seen how to find estimates of model coefficients, using theorems and vector-matrix notations.
- Now, we will focus on what model coefficient values tell us and how to interpret them
- And we will look at the common cases of using linear regression models
- We will do this via analyzing some examples

2.2 Example: Plasma volume

```
# data
weight <- c(58, 70, 74, 63.5, 62.0, 70.5, 71.0, 66.0) # body weight (kg)
plasma <- c(2.75, 2.86, 3.37, 2.76, 2.62, 3.49, 3.05, 3.12) # plasma volume (liters)

# fit regression model
```

```
model <- lm(plasma ~ weight)

# plot the original data and fitted regression line
plot(weight, plasma, pch=19, xlab="weight [kg]", ylab="plasma [l]")
lines(weight, model$fitted.values, col="red") # fitted model in red
grid()

# print model summary
print(summary(model))
##
## Call:
## lm(formula = plasma ~ weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27880 -0.14178 -0.01928  0.13986  0.32939
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.08572    1.02400   0.084   0.9360
## weight       0.04362    0.01527   2.857   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2188 on 6 degrees of freedom
## Multiple R-squared:  0.5763, Adjusted R-squared:  0.5057
## F-statistic:  8.16 on 1 and 6 DF,  p-value: 0.02893
```

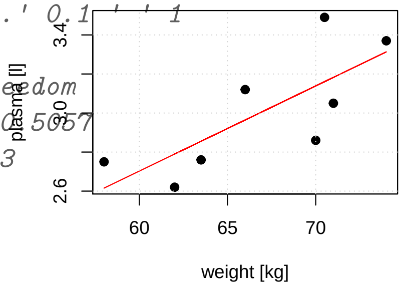


Figure 2.1: Scatter plot showing plasma volume for each weight

Model:

- $Y_i = \alpha + \beta x_i + \epsilon_i$ where x_i corresponds to $weight_i$

Slope

- The value of slope tells us how and by much the outcome changes with a unit change in x
- If we go up in weight 1 kg what would be our expected change in plasma volume[1]?
- And if we go up in weight 10 kg what would be our expected change in plasma volume[2]?

Intercept

- the **intercept**, often labeled the **constant**, is the value of Y when $x_i = 0$
- in models where x_i can be equal 0, the intercept is simply the expected mean value of response
- in models where x_i cannot be equal 0, like in our plasma example no weight makes no sense for healthy men, the intercept has no intrinsic meaning
- the intercept is thus quite often ignored in linear models, as it is the value of slope that dictates the association between exposure and outcome

- [1]: If we go up in weight 1 kg we would expect our plasma volume to increase by 0.04 liter since $\hat{\beta} = 0.04$
- [2] If we go up in weight 10 kg we would expect our plasma volume to increase by $0.04 \cdot 10 = 0.4$ liter

2.3 Example: Galapagos Islands

Researchers were interested in biological diversity on the Galapagos islands. They have collected data on number of plant species (Species) and number of endemic species on 30 islands as well as some descriptors of the islands such as area [km^2], elevation [m], distance to nearest island [km], distance to Santa Cruz [km] and the area of the adjacent island [km^2].

The preview of data is here:

```r
# data is available via faraway package
if(!require(faraway)){
    install.packages("faraway")
    library(faraway)
}

head(gala, 10) %>%
  kable() %>%
  kable_paper("hover")
```

Table 2.1: Preview of the Galapagos Islands data

	Species	Endemics	Area	Elevation	Nearest	Scruz	Adjacent
Baltra	58	23	25.09	346	0.6	0.6	1.84
Bartolome	31	21	1.24	109	0.6	26.3	572.33
Caldwell	3	3	0.21	114	2.8	58.7	0.78
Champion	25	9	0.10	46	1.9	47.4	0.18
Coamano	2	1	0.05	77	1.9	1.9	903.82
Daphne.Major	18	11	0.34	119	8.0	8.0	1.84
Daphne.Minor	24	0	0.08	93	6.0	12.0	0.34
Darwin	10	7	2.33	168	34.1	290.2	2.85
Eden	8	4	0.03	71	0.4	0.4	17.95
Enderby	2	2	0.18	112	2.6	50.2	0.10

And we can fit a linear regression model to model number of *Species* given the remaining variables. Let's keep aside for now that number of *Species* is actually a count variable, not a continuous numerical variable, we just want to estimate the number of *Species* for now.

Fitted Model

- $Y_i = \beta_0 + \beta_1 Area_i + \beta_2 Elevation_i + \beta_3 Nearest_i + \beta_4 Scruz_i + \beta_5 Adjacent_i + \epsilon_i$

```
# fit multiple linear regression and print model summary
model1 <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
print(summary(model1))
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

Using the model compare two islands in terms of number of species

- if the second island has an elevation 1 m higher than the first one?[1]
- if the second island has an elevation 100 m higher than the first one?[2]
- if the second island is 100 km closer to Santa Cruz?[3]
- overall, is there a relationship between the response Y (Species) and predictors?[4]

- [1] the second island will have 0.32 species more than the first one, $\hat{\beta_2} = 0.319465 \approx 0.32$
- [2] the second island will have $0.32 \cdot 100 = 32$ more species than the first one
- [3] the second island would have $-0.24 \cdot 100 = -24$ less species than the first if there was enough evidence to reject the null hypothesis of $\beta_4 = 0$; It is not appropriate to try to interpret non-significant coefficients.
- [4] we have seen before that in the case of simple linear regression it was enough to test the null hypothesis of $H_0 : \beta = 0$ versus $H_0 : \beta \neq 0$ to answers the question whether there is an overall relationship between response and predictor. In case of multiple regression, with many predictors, we need to test the null hypothesis of

  $$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

  versus the alternative

  $$H_a : at\ least\ one\ \beta_j\ is\ non-zero$$

> This hypothesis test is performed by computing **F-statistics** reported in the model summary and calculated as $F = \frac{(TSS-RSS)/p}{RSS/(n-p-1)}$ where $TSS = \sum(y_i - \bar{y})^2$ and $RSS = \sum(y_i - \hat{y_i})^2$. Here, the $F - statsitics = 15.7$ and the associated $p - value < 0.05$ so there is enough evidence to reject the null hypothesis in favor of the alternative and conclude that there is an overall significant relationship between response (Species) and predictors.

Not so easy: alternative model

Consider an alternative model where we only use elevation to model the number of species

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

We fit the model in `R` and look at the model summary

```
model2 <- lm(Species ~ Elevation, data = gala)
print(summary(model2))
##
## Call:
## lm(formula = Species ~ Elevation, data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -218.319  -30.721  -14.690    4.634  259.180
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.33511   19.20529   0.590     0.56
## Elevation    0.20079    0.03465   5.795 3.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.66 on 28 degrees of freedom
## Multiple R-squared:  0.5454, Adjusted R-squared:  0.5291
## F-statistic: 33.59 on 1 and 28 DF,  p-value: 3.177e-06
```

Using the alternative model compare again two islands in terms of number of species

- if the second island has an elevation 1 m higher than the first one?[1]
- if the second island has an elevation 100 m higher than the first one?[2]

- [1] the second island will have 0.20 species more than the first one
- [2] the second island will have $0.20 \cdot 100 = 20$ more species

Specific interpretation

- Obviously there is difference between 32 and 20 times more species given the same elevation difference as obtained by the multiple regression (first model) and simple regression (alternative model).
- Our interpretations need to be more specific and we say that **a unit increase in x with other predictors held constant will produce a change equal to $\hat{\beta}$ in the response y**
- It is of course often quite unrealistic to be able to control other variables and keep them constant and for our alternative model, a change in evaluation is most likely associated with other variables, even though they are not included in the model.
- Further, our explanation contains **no notation of causation**, even though the two models are showing a strong association between elevation and number of species.
- We will learn later how to choose the best model by assessing its fit and including only relevant variable (feature selection), for now we focus on learning how to interpret the coefficients given a fitted model.

2.4 Example: Height and gender

Data are available containing the weight [lbs] and height [inches] of 10000 men and women

```r
# read in data
htwtgen <- read.csv("data/lm/heights_weights_genders.csv")
head(htwtgen)
```

```
##   Gender   Height   Weight
## 1   Male 73.84702 241.8936
## 2   Male 68.78190 162.3105
## 3   Male 74.11011 212.7409
## 4   Male 71.73098 220.0425
## 5   Male 69.88180 206.3498
## 6   Male 67.25302 152.2122

# boxplot for females and males
boxplot(htwtgen$Height ~ htwtgen$Gender,
        xlab="", ylab="Height", col="lightblue")
```

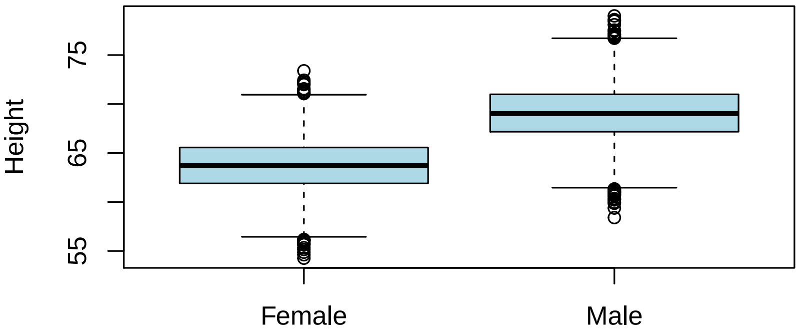


Figure 2.2: Box plot for 10 000 heigth measurments stratifed by gender

- We want to **compare the average height of men and women**.
- We can do that using linear regression and including gender as **binary variable**

Model

$$Y_i = \alpha + \beta I_{x_i} + \epsilon_i$$

where

$$I_{x_i} = \begin{cases} 1 & \text{if } x_i = 1 \\ 0 & \text{if } x_i = 0 \end{cases} \tag{2.1}$$

for some coding, e.g. we choose to set "Female=1" and "Male=0" or vice versa.

In R we write:

```
# Note: check that Gender is indeed non-numeric
print(class(htwtgen$Gender))
## [1] "character"

# fit linear regression and print model summary
model1 <- lm(Height ~ Gender, data = htwtgen)
print(summary(model1))
##
## Call:
## lm(formula = Height ~ Gender, data = htwtgen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6194  -1.8374   0.0088   1.9185   9.9724
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 63.70877    0.03933  1619.8   <2e-16 ***
## GenderMale   5.31757    0.05562    95.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.781 on 9998 degrees of freedom
## Multiple R-squared:  0.4776, Adjusted R-squared:  0.4775
## F-statistic:  9140 on 1 and 9998 DF,  p-value: < 2.2e-16
```

Estimates

$$\hat{\alpha} = 63.71$$

$$\hat{\beta} = 5.32$$

- the `lm()` function chooses automatically one of the category as baseline, here Females
- model summary prints the output of the model with the baseline category "hidden"
- i.e. notice the only label we have is "GenderMale"

- meaning that we ended-up having a model coded as below:

$$I_{x_i} = \begin{cases} 1 & \text{if} \quad person_i \; is \; male \\ 0 & \text{if} \quad person_i \; is \; female \end{cases} \tag{2.2}$$

- Consequently, if observation i is male then the expected value of height is:

$$E(Height_i|Male) = 63.71 + 5.32 = 69.03$$

- and if observation i is female then the expected value of height is:

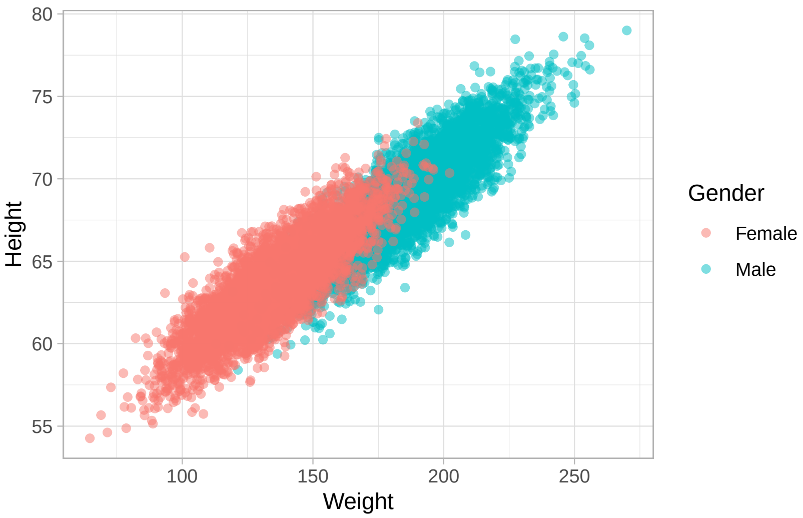
$$E(Height_i|Male) = 63.71$$

2.5 Example: Height, weight and gender I

- So as expected, there is a difference in average height between men and women.
- Can we also observe a significant relationship between weight and height?
- And if so, does this relationship depend on gender?

```
#|label: fig-htwtgen-plot
#|fig-cap: Scatter plot showing height measurments given weight, stratified by gender
#|fig-cap-location: margin
#|collapse: true
#|code-fold: false
#|fig-width: 5
#|fig-heigth: 5

# plot the data separately for Male and Female
ggplot(data=htwtgen, aes(x = Weight, y=Height, col = Gender)) +
  geom_point(alpha = 0.5) +
  theme_light()
```



- From the plot we can see that height increases with weight.
- On average, men are taller than women.
- On average, men weight more than women.
- The relationship between height and weight appears to be the same for males and females, i.e. height increases with weight for both men and women.

To assess the relationship we use a model containing height and gender.

Model

$$Y_i = \alpha + \beta I_{x_i} + \gamma x_{2,i} + \epsilon_i$$

where

$$I_{x_i} = \begin{cases} 1 & \text{if} \quad person_i \; is \; male \\ 0 & \text{if} \quad person_i \; is \; female \end{cases} \tag{2.3}$$

and $x_{2,i}$ is the weight of person i

In R we write:

```
# fit linear model and print model summary
model2 <- lm(Height ~ Gender + Weight, data = htwtgen)
print(summary(model2))
##
```

```
## Call:
## lm(formula = Height ~ Gender + Weight, data = htwtgen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4956 -0.9583  0.0126  0.9867  5.8358
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.0306678  0.1025161  458.76   <2e-16 ***
## GenderMale  -0.9628643  0.0474947  -20.27   <2e-16 ***
## Weight       0.1227594  0.0007396  165.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.435 on 9997 degrees of freedom
## Multiple R-squared:  0.8609, Adjusted R-squared:  0.8609
## F-statistic: 3.093e+04 on 2 and 9997 DF,  p-value: < 2.2e-16
```

Model together with estimates

$$Y_i = \alpha + \beta I_{x_i} + \gamma x_{2,i} + \epsilon_i$$

where

$$I_{x_i} = \begin{cases} 1 & \text{if} \quad person_i \; is \; male \\ 0 & \text{if} \quad person_i \; is \; female \end{cases} \tag{2.4}$$

and $x_{2,i}$ is the weight of person i

Estimates

$$\hat{\alpha} = 47.031$$

$$\hat{\beta} = -0.963$$

$$\hat{\gamma} = 0.123$$

- Using our estimates, for a male of with an example weight of 161.4 we would predict a height of:

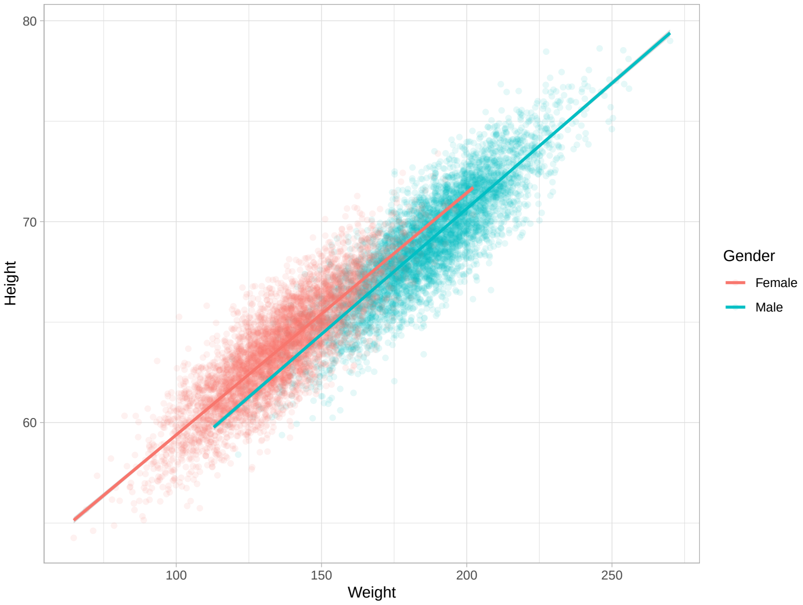
$$E(Height_i|Male, Weight = 161.4) = 47.031 — 0.963 + (0.123 \cdot 161.4) = 65.9$$

- and for a female of weight 161.4 we would predict a height of

$$E(Height_i|Female, Weight = 161.4) = 47.031 + (0.123 \cdot 161.4) = 66.9$$

In `R` we can plot our data and the fitted moded to verify our calculations:

```
# plot the data separately for men and women
# using ggplot() and geom_smooth()
ggplot(data=htwtgen, aes(x = Weight, y=Height, col = Gender)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method=lm) +
  theme_light() +
  guides(color=guide_legend(override.aes=list(fill=NA)))
```



2.6 Example: Heigth, weight and gender II

- The fitted lines in the above example are **parallel**, the **slope is modeled to be the same for men and women**, and the intercept denotes the group differences
- It is also possible to allow **both intercept and slope being fitted separately for each group**
- This is done when we except that the relationships are different in different groups, e.g. increasing in one group and decreasing in the other.
- And we then talk about including **interaction effect**, as the two lines may interact (cross).

Model

$$Y_{i,j} = \alpha_i + \beta_i x_{ij} + \epsilon_{i,j}$$

where:

- $Y_{i,j}$ is the height of person j of gender i
- x_{ij} is the weight of person j of gender i
- $i = 1$ corresponds to men in our example (keeping the same coding as above)
- $i = 2$ corresponds to women

In `R` we define the interaction term with `*`:

```
# fit linear model with interaction
model3 <- lm(Height ~ Gender * Weight, data = htwtgen)
print(summary(model3))
##
## Call:
## lm(formula = Height ~ Gender * Weight, data = htwtgen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4698 -0.9568  0.0092  0.9818  5.7544
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         47.347783   0.146325 323.579  < 2e-16 ***
## GenderMale        -1.683668   0.242119  -6.954 3.78e-12 ***
## Weight             0.120425   0.001067 112.903  < 2e-16 ***
## GenderMale:Weight  0.004493   0.001480   3.036   0.0024 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.435 on 9996 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.861
## F-statistic: 2.064e+04 on 3 and 9996 DF,  p-value: < 2.2e-16
```

Now, based on the regression output we would expect:

- for a men of weight x, a height of:

$$E(height|male\ and\ weight = x) = 47.34778 - 1.68367 + 0.12043x + 0.00449x = 45.7 + 0.125x$$

- for a women of weight x, a height of

  $E(height|female\ and\ weight = x) = 47.34778 + 0.12043x$

Estimates

$$\hat{\alpha_1} = 45.7$$

$$\hat{\beta_1} = 0.125$$

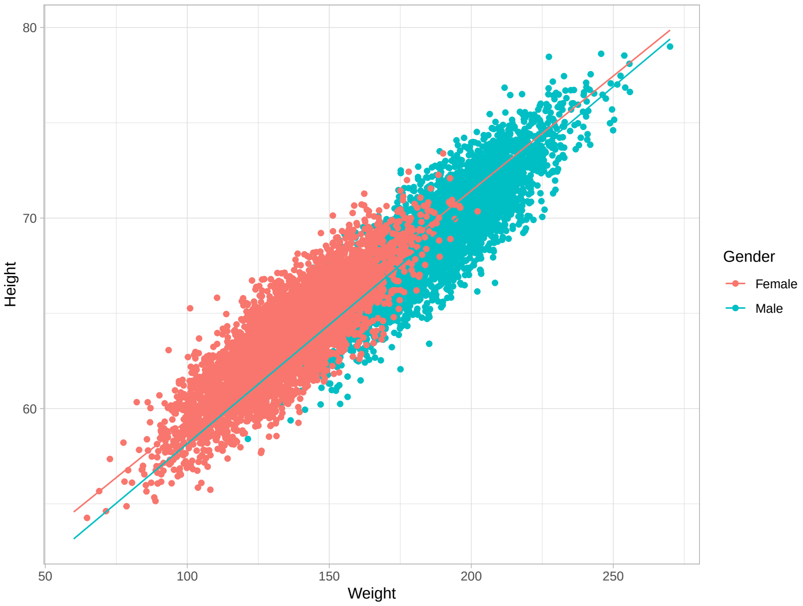
$$\hat{\alpha_2} = 47.34778$$

$$\hat{\beta_2} = 0.12043$$

- We can see from the regression output that the interaction term, “GenderMale:Weight, is significant
- and therefore the relationship between weight and height is different for men and women.
- We can plot the fitted model and see that the lines are no longer parallel.
- We will see clearer example of the interactions in the exercises.

```
# ggiraphExtra makes it easy to visualize fitted models
if(!require(ggiraphExtra)){
    install.packages("ggiraphExtra")
    library(ggiraphExtra)
}

ggPredict(model3) +
  theme_light() +
  guides(color=guide_legend(override.aes=list(fill=NA)))
```



3 Exercises (Regression coefficients)

Data for exercises are on Canvas under Files -> data_exercises –> linear-models

Exercise 3.1 (Height-weigth-gender). a) repeat fitting the models with a) gender, b) weight and gender and c) interaction between weight and gender

b) given the model with the interaction term, what is expected height of a man and a women given a weight of 120 lbs?

c) can you use predict() function to check your calculations?

Exercise 3.2 (Trout). When the behavior of a group of trout is studied, some fish are observed to become dominant and others to become subordinate. Dominant fish have freedom of movement whereas subordinate fish tend to congregate in the periphery of the waterway to avoid crossing the path of the dominant fish. Data on energy expenditure and ration of blood obtained were collected as part of a laboratory experiment for 20 trout. Energy and ration is measured in calories per kilocalorie per trout per day.

Use the below code to load the data to R and use linear regression models to answer:

- a) is there a relationship between ration obtained and energy expenditure
- b) is the relationship between ration obtained and energy expenditure different for each type of fish?

- Hint: it is good to start with some explanatory plots between every pair of variable

```r
# read in data and show preview
trout <- read.csv("data/lm/trout.csv")

# recode the Group variable and treat like categories (factor)
trout$Group <- factor(trout$Group, labels=c("Dominant", "Subordinate"))
```

Exercise 3.3 (Lowering blood pressure)**.** A clinical trial has been carried out to compare three drug treatments which are intended to lower blood pressure in hypertensive patients. The data contains initial values fo systolic blood pressure (bp) in mmHg for each patient and the reduction achieved during the course of the trial. For each patient, allocation to treatment (drug) was carried out randomly and conditions such as the length of the treatment and dose of the drug were standardized as far as possible.

Use linear regression to answer questions:

a) is there an association between the reduction in blood pressure and initial blood pressure
b) is reduction in blood pressure different across the treatment (in three drug groups)?
c) is reduction in blood pressure different across the treatment when accounting for initial blood pressure?
d) is reduction in blood pressure changing differently under different treatment? Hint: here we have three categories which can be seen as expanding the model with two categories by an additional one: one category will be treated as baseline

```r
blooddrug <- read.csv("data/lm/bloodrug.csv")
blooddrug$drug <- factor(blooddrug$drug)
head(blooddrug)
```

```
  initial redn drug
1     158    4    1
2     176   21    1
3     174   36    1
4     168   14    1
5     174   34    1
6     186   37    1
```

3.1 Answers to selected exercises

Solution. Exercise 3.1

a)

```
htwtgen <- read.csv("data/lm/heights_weights_genders.csv")
head(htwtgen)
```

```
  Gender   Height   Weight
1   Male 73.84702 241.8936
2   Male 68.78190 162.3105
3   Male 74.11011 212.7409
4   Male 71.73098 220.0425
5   Male 69.88180 206.3498
6   Male 67.25302 152.2122
```

```
# a)
model1 <- lm(Height ~ Gender, data = htwtgen)
model2 <- lm(Height ~ Gender + Weight, data = htwtgen)
model3 <- lm(Height ~ Gender * Weight, data = htwtgen)

# print(summary(model1))
# print(summary(model2))
# print(summary(model3))
```

b) use equations to find the height for men and women respectively:

$$E(height|male\ and\ weight = x) = 47.34778 - 1.68367 + 0.12043x + 0.00449x = 45.7 + 0.125x$$

$$E(height|female\ and\ weight = x) = 47.34778 + 0.12043x$$

c)

```
# for men
new.obs <- data.frame(Weight=120, Gender="Male")
predict(model3, newdata = new.obs)
```

```
       1 
60.65427 
```

```
# for female
new.obs <- data.frame(Weight=120, Gender="Female")
predict(model3, newdata = new.obs)
```

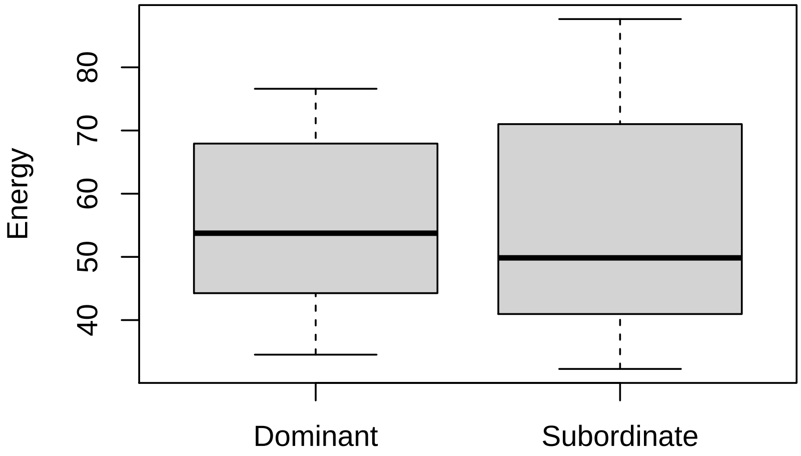
```
       1 
61.79882 
```

Solution. Exercise 3.2

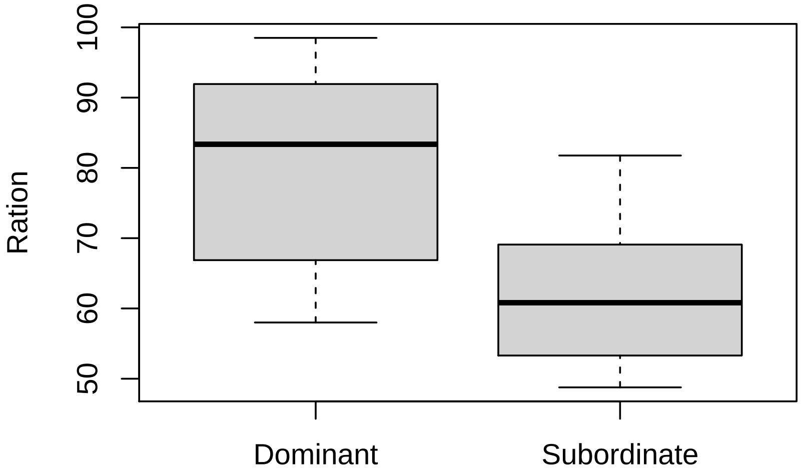
```
# read in data and show preview
trout <- read.csv("data/lm/trout.csv")

# recode the Group variable and treat like categories (factor)
trout$Group <- factor(trout$Group, labels=c("Dominant", "Subordinate"))
head(trout)
##   Energy Ration    Group
## 1  44.26  81.35 Dominant
## 2  67.16  91.68 Dominant
## 3  48.15  58.00 Dominant
## 4  34.53  58.63 Dominant
## 5  67.93  91.93 Dominant
## 6  72.45  96.56 Dominant
```

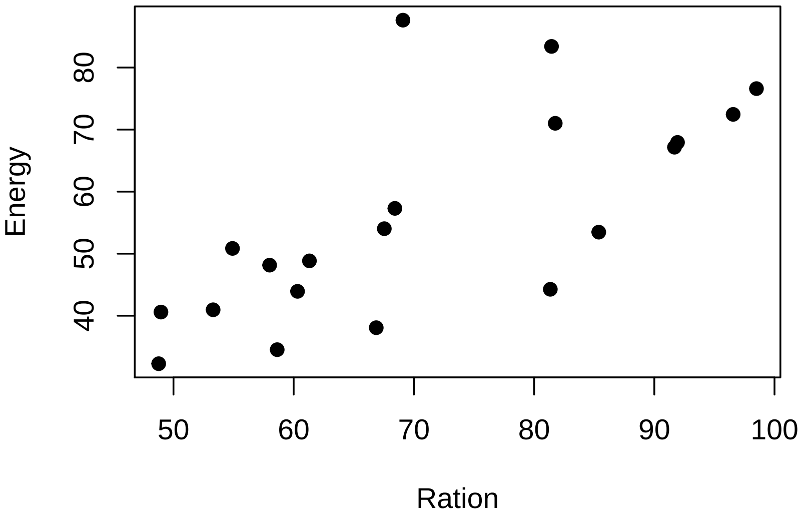
```
# plot data
# boxplots of Energy and Ration per group
boxplot(trout$Energy ~ trout$Group, xlab="", ylab="Energy")
```



```
boxplot(trout$Ration ~ trout$Group, xlab="", ylab="Ration")
```



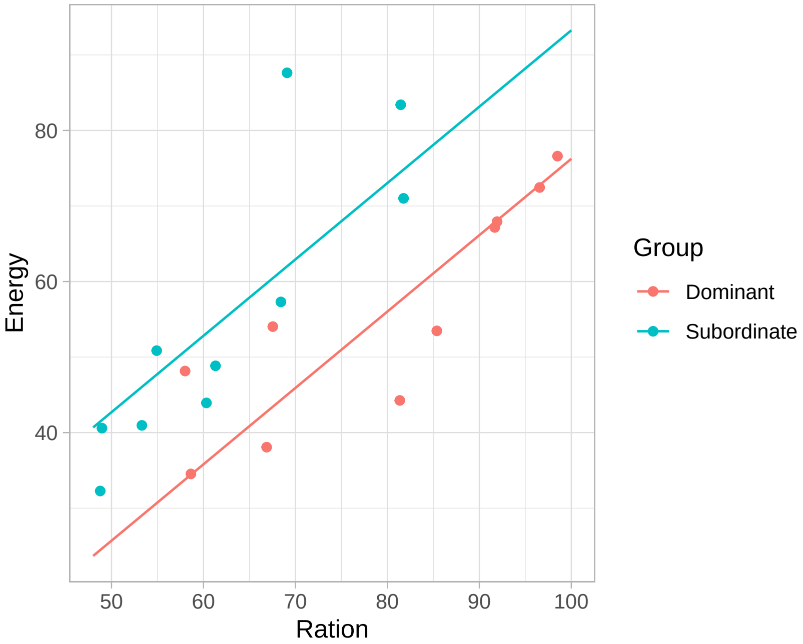
```
# scatter plot of Ration vs. Energy
plot(trout$Ration, trout$Energy, pch=19, xlab="Ration", ylab="Energy")
```



- From the exploratory plots we see that there is some sort of relationship between ratio and energy, i.e. energy increase while ration obtained increases
- From box plots we see that the ration obtained may be different in two groups

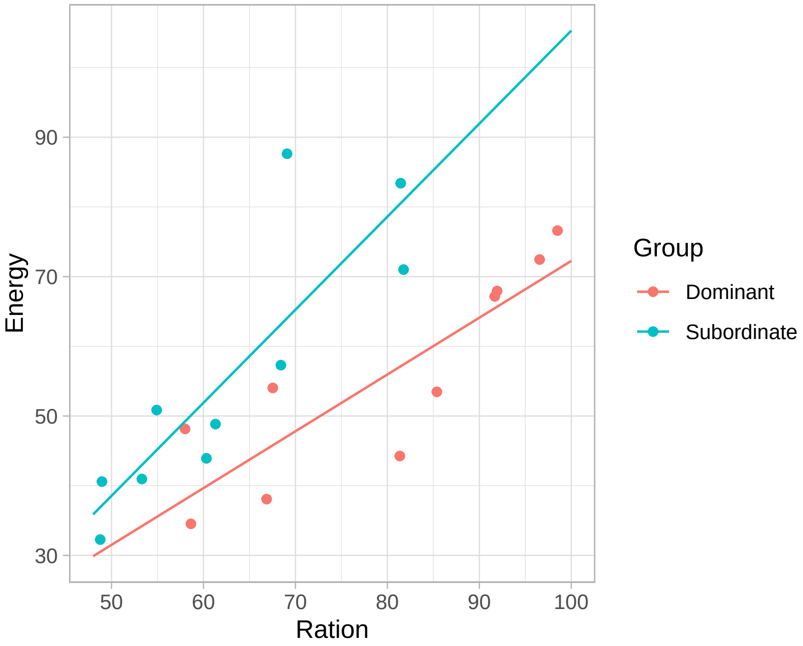
```
# Is there a relationship between ration obtained and energy expenditure
model1 <- lm(Energy ~ Ration, data = trout)
print(summary(model1))
##
## Call:
## lm(formula = Energy ~ Ration, data = trout)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.704  -4.703  -0.578   2.432  33.506
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3037    12.5156   0.344 0.734930
## Ration        0.7211     0.1716   4.203 0.000535 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.05 on 18 degrees of freedom
## Multiple R-squared:  0.4953, Adjusted R-squared:  0.4673
## F-statistic: 17.66 on 1 and 18 DF,  p-value: 0.0005348
# from the regression output we can see that yes, a unit increase in ratio increase energy e

# Is there a relationship between ration obtained and energy expenditure different for each
# we first check if there is a group effect
model2 <- lm(Energy ~ Ration + Group, data = trout)
print(summary(model2))
##
## Call:
## lm(formula = Energy ~ Ration + Group, data = trout)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.130  -5.139  -0.870   2.199  25.622
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -24.8506    13.3031  -1.868  0.07910 .
## Ration              1.0109     0.1626   6.218 9.36e-06 ***
## GroupSubordinate  17.0120     5.1075   3.331  0.00396 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.647 on 17 degrees of freedom
## Multiple R-squared:  0.6946, Adjusted R-squared:  0.6587
## F-statistic: 19.33 on 2 and 17 DF,  p-value: 4.182e-05
ggPredict(model2) +
  theme_light() +
  guides(color=guide_legend(override.aes=list(fill=NA)))
```



```
# and whether there is an interaction effect
model3 <- lm(Energy ~ Ration * Group, data = trout)
print(summary(model3))
##
## Call:
## lm(formula = Energy ~ Ration * Group, data = trout)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7951  -6.0981  -0.1554   3.9612  23.5946
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -9.2330    15.9394  -0.579 0.570483
## Ration                       0.8149     0.1968   4.141 0.000767 ***
## GroupSubordinate           -18.9558    22.6934  -0.835 0.415848
## Ration:GroupSubordinate      0.5200     0.3204   1.623 0.124148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9.214 on 16 degrees of freedom
## Multiple R-squared:  0.7378, Adjusted R-squared:  0.6886
## F-statistic:    15 on 3 and 16 DF,  p-value: 6.537e-05
ggPredict(model3) +
  theme_light() +
  guides(color=guide_legend(override.aes=list(fill=NA)))
```



Based on the regression output and plots we can say:

- there is a relationship between ration obtained and energy expenditure
- that this relationship is the same in the two groups although the energy expenditure is higher in the dominant fish

Solution. Exercise 3.3

a)

Yes. The `redn` and `initial` were significantly associated (p-value = 0.00312, linear regression).

```
model1 <- lm(redn ~ initial, data = blooddrug)
summary(model1)
##
## Call:
## lm(formula = redn ~ initial, data = blooddrug)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.476 -11.705   1.558   9.197  24.392
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -72.7302    29.1879  -2.492  0.02036 *
## initial       0.5902     0.1788   3.301  0.00312 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.79 on 23 degrees of freedom
## Multiple R-squared:  0.3214, Adjusted R-squared:  0.2919
## F-statistic: 10.89 on 1 and 23 DF,  p-value: 0.003125
```

b)

No. The `drug2` and `drug3` were not significantly different from `drug1` (p-value = 0.714 and p-value = 0.628, respectively). The patients of the drug 1 group had 2.750 higher blood pressure drop (`redn`) than those of the drug 2 group. However, the difference was relatively small comparing to the standard error of the estimate, which was 7.402. The difference between drug 1 and 3 was relatively small, too.

```
model2 <- lm(redn ~ drug, data = blooddrug)
summary(model2)
##
## Call:
## lm(formula = redn ~ drug, data = blooddrug)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -32.000  -9.286   0.000  12.714  26.000 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   23.250      5.517   4.214 0.000358 ***
## drug2          2.750      7.402   0.372 0.713796    
## drug3         -3.964      8.076  -0.491 0.628379    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 15.6 on 22 degrees of freedom
## Multiple R-squared:  0.03349,    Adjusted R-squared:  -0.05437 
## F-statistic: 0.3812 on 2 and 22 DF,  p-value: 0.6875
```

c)

Yes. The `redn` of the `drug2` group was significantly higher than that of the `drug1` group after adjustment for the effects of the `initial` (P = 0.018). The reduction of the patients who got the drug 2 was much higher (13.6906) than the drug 1, comparing to the standard error of the difference (5.3534) after accounting for initial blood pressure.

```
model3 <- lm(redn ~ drug + initial, data = blooddrug)
summary(model3)
## 
## Call:
## lm(formula = redn ~ drug + initial, data = blooddrug)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -15.8114 -10.5842  -0.4959   6.2834  16.4265 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -124.8488    28.0674  -4.448 0.000223 ***
## drug2         13.6906     5.3534   2.557 0.018346 *  
## drug3         -7.2045     5.4275  -1.327 0.198625    
## initial        0.8895     0.1671   5.323 2.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.42 on 21 degrees of freedom
## Multiple R-squared:  0.5886, Adjusted R-squared:  0.5298
## F-statistic: 10.01 on 3 and 21 DF,  p-value: 0.0002666
```

4 Model diagnostics

Aims

- to introduce concepts of linear models summary and assumptions

Learning outcomes

- to able to interpret R^2 and $R^2(adj)$ values
- state the assumptions of a linear model and assess them using residual plots

4.1 Assessing model fit

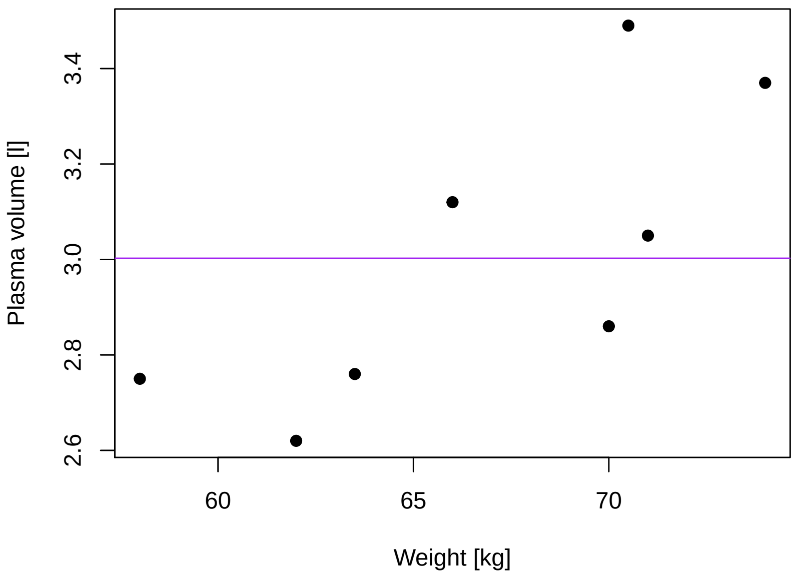
- earlier we learned how to estimate parameters in a liner model using least squares
- now we will consider how to assess the goodness of fit of a model
- we do that by calculating the amount of variability in the response that is explained by the model

4.2 R^2: summary of the fitted model

- considering a simple linear regression, the simplest model, **Model 0**, we could consider fitting is

  $$Y_i = \beta_0 + \epsilon_i$$

  that corresponds to a line that run through the data but lies parallel to the horizontal axis
- in our plasma volume example that would correspond the mean value of plasma volume being predicted for any value of weight (in purple)



- TSS, denoted **Total corrected sum-of-squares** is the residual sum-of-squares for Model 0

$$S(\hat{\beta}_0) = TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2 = S_{yy}$$

corresponding the to the sum of squared distances to the purple line

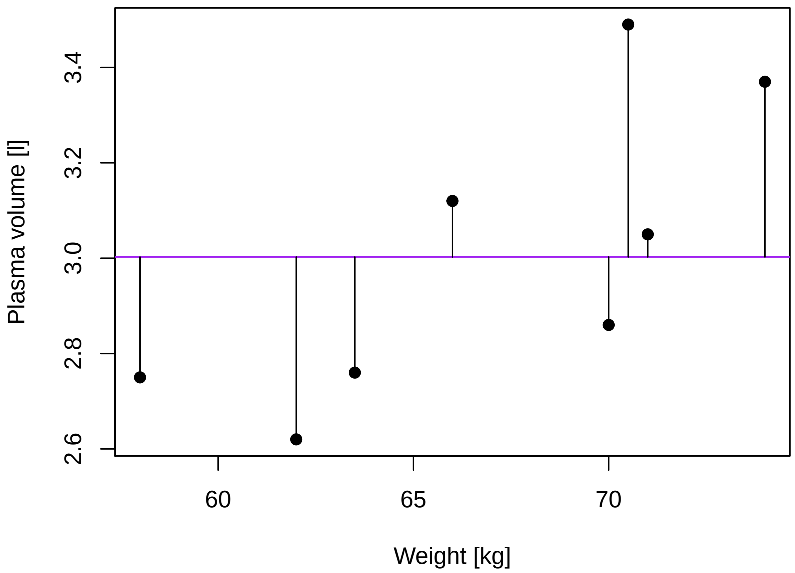
- Fitting **Model 1** of the form

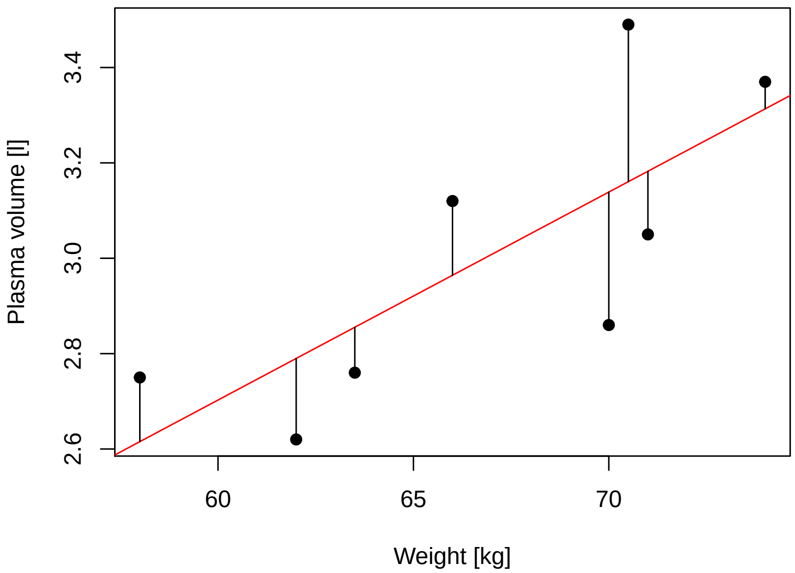
  $$Y_i = \beta_0 + \beta_1 x + \epsilon_i$$

  we have earlier defined
- **RSS**, the residual sum-of-squares as:

$$RSS = \sum_{i=1}^{n}(y_i - \{\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \dots + \hat{\beta}_p x_{pi}\}) = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

- that corresponds to the squared distances between the observed values $y_i, \dots, y_n$ to fitted values $\hat{y_1}, \dots \hat{y_n}$, i.e. distances to the red fitted line


Plasma volume [l]
2.6
2.8
3.0
3.2
3.4
60
65
70
Weight [kg]


Plasma volume [l]
2.6
2.8
3.0
3.2
3.4
60
65
70
Weight [kg]

Definition 4.1 (R^2)**.** A simple but useful measure of model fit is given by

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

where:

- RSS is the residual sum-of-squares for Model 1, the fitted model of interest
- TSS is the sum of squares of the **null model**

- R^2 quantifies how much of a drop in the residual sum-of-squares is accounted for by fitting the proposed model
- R^2 is also referred as **coefficient of determination**
- It is expressed on a scale, as a proportion (between 0 and 1) of the total variation in the data
- Values of R^2 approaching 1 indicate the model to be a good fit
- Values of R^2 less than 0.5 suggest that the model gives rather a poor fit to the data

4.3 R^2 and correlation coefficient

Theorem 4.1 (R^2)**.** *In the case of simple linear regression:*

Model 1: $Y_i = \beta_0 + \beta_1 x + \epsilon_i$

$$R^2 = r^2$$

where:

- R^2 *is the coefficient of determination*
- r^2 *is the sample correlation coefficient*

4.4 $R^2(adj)$

- in the case of multiple linear regression, where there is more than one explanatory variable in the model
- we are using the adjusted version of R^2 to assess the model fit
- as the number of explanatory variables increase, R^2 also increases
- $R^2(adj)$ takes this into account, i.e. adjusts for the fact that there is more than one explanatory variable in the model

Theorem 4.2 ($R^2(adj)$)**.** *For any multiple linear regression*

$$Y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{p-1} x_{(p-1)i} + \epsilon_i$$

$R^2(adj)$ is defined as

$$R^2(adj) = 1 - \frac{\frac{RSS}{n-p-1}}{\frac{TSS}{n-1}}$$

where

- *p is the number of independent predictors, i.e. the number of variables in the model, excluding the constant*

$R^2(adj)$ can also be calculated from R^2:

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

We can calculate the values in R and compare the results to the output of linear regression

```
htwtgen <- read.csv("data/lm/heights_weights_genders.csv")
head(htwtgen)
##   Gender   Height   Weight
## 1   Male 73.84702 241.8936
## 2   Male 68.78190 162.3105
## 3   Male 74.11011 212.7409
## 4   Male 71.73098 220.0425
## 5   Male 69.88180 206.3498
## 6   Male 67.25302 152.2122
attach(htwtgen)

## Simple linear regression
model.simple <- lm(Height ~ Weight, data=htwtgen)

# TSS
TSS <- sum((Height - mean(Height))^2)

# RSS
# residuals are returned in the model type names(model.simple)
RSS <- sum((model.simple$residuals)^2)
R2 <- 1 - (RSS/TSS)

print(R2)
## [1] 0.8551742
print(summary(model.simple))
##
## Call:
## lm(formula = Height ~ Weight, data = htwtgen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8142 -0.9907  0.0263  0.9918  5.5950
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.848e+01  7.507e-02   645.8   <2e-16 ***
## Weight      1.108e-01  4.561e-04   243.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.464 on 9998 degrees of freedom
## Multiple R-squared:  0.8552, Adjusted R-squared:  0.8552
## F-statistic: 5.904e+04 on 1 and 9998 DF,  p-value: < 2.2e-16
## Multiple regression
model.multiple <- lm(Height ~ Weight + Gender, data=htwtgen)
n <- length(Weight)
p <- 1

RSS <- sum((model.multiple$residuals)^2)
R2_adj <- 1 - (RSS/(n-p-1))/(TSS/(n-1))

print(R2_adj)
## [1] 0.8608793
print(summary(model.multiple))
##
## Call:
## lm(formula = Height ~ Weight + Gender, data = htwtgen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4956 -0.9583  0.0126  0.9867  5.8358
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.0306678  0.1025161  458.76   <2e-16 ***
## Weight       0.1227594  0.0007396  165.97   <2e-16 ***
## GenderMale  -0.9628643  0.0474947  -20.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.435 on 9997 degrees of freedom
## Multiple R-squared:  0.8609, Adjusted R-squared:  0.8609
## F-statistic: 3.093e+04 on 2 and 9997 DF,  p-value: < 2.2e-16
```

4.5 The assumptions of a linear model

- up until now we were fitting models and discussed how to assess the model fit
- before making use of a fitted model for explanation or prediction, it is wise to check that the model provides an adequate description of the data

- informally we have been using box plots and scatter plots to look at the data
- there are however formal definitions of the assumptions

Assumption A: The deterministic part of the model captures all the non-random structure in the data

- this implies that the **mean of the errors** ϵ_i is zero
- it applies only over the range of explanatory variables

Assumption B: the scale of variability of the errors is constant at all values of the explanatory variables

- practically we are looking at whether the observations are equally spread on both side of the regression line

Assumption C: the errors are independent

- broadly speaking this means that knowledge of errors attached to one observation does not give us any information about the error attached to another

Assumptions D: the errors are normally distributed

- this will allow us to describe the variation in the model's parameters estimates and therefore make inferences about the population from which our sample was taken

Assumption E: the values of the explanatory variables are recorded without error

- this one is not possible to check via examining the data, instead we have to consider the nature of the experiment

4.6 Checking assumptions

Residuals, $\hat{\epsilon_i} = y_i - \hat{y_i}$ are the **main ingredient to check model assumptions**. We use plots such as:

1. Histograms or normal probability plots of $\hat{\epsilon_i}$

- useful to check the assumption of normality

2. Plots of $\hat{\epsilon}_i$ versus the fitted values $\hat{y}_i$

- used to detect changes in error variance
- used to check if the mean of the errors is zero

3. Plots of $\hat{\epsilon}_i$ vs. an explanatory variable x_{ij}

- this helps to check that the variable x_j has a linear relationship with the response variable

4. Plots of $\hat{\epsilon}_i$ vs. an explanatory variable x_{kj} that is **not** in the model

- this helps to check whether the additional variable x_k might have a relationship with the response variable

4. Plots of $\hat{\epsilon}_i$ in the order of the observations were collected

- this is useful to check whether errors might be correlated over time

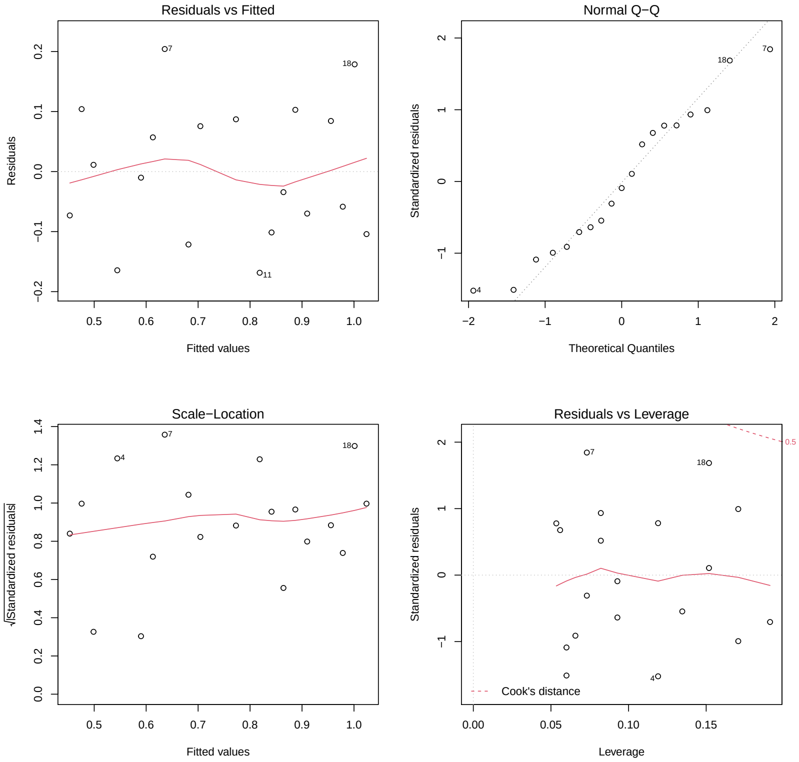
Let's look at the "good" example going back to our data of protein levels during pregnancy

```r
# read in data
data.protein <- read.csv("data/lm/protein.csv")

protein <- data.protein$Protein # our Y
gestation <- data.protein$Gestation # our X

model <- lm(protein ~ gestation)

# plot diagnostic plots of the linear model
# by default plot(model) calls four diagnostics plots
# par() divides plot window in 2 x 2 grid
par(mfrow=c(2,2))
plot(model)
```



- the residual plots provides examples of a situation where the assumptions appear to be met
- the linear regression appears to describe data quite well
- there is no obvious trend of any kind in the residuals vs. fitted values (the shape is scattered)
- points lie reasonably well along the line in the normal probability plot, hence normality appears to be met

Examples of assumptions not being met

4.7 Influential observations

- Sometimes individual observations can exert a great deal of influence on the fitted model
- One routine way of checking for this is to fit the model n times, missing out each observation in turn

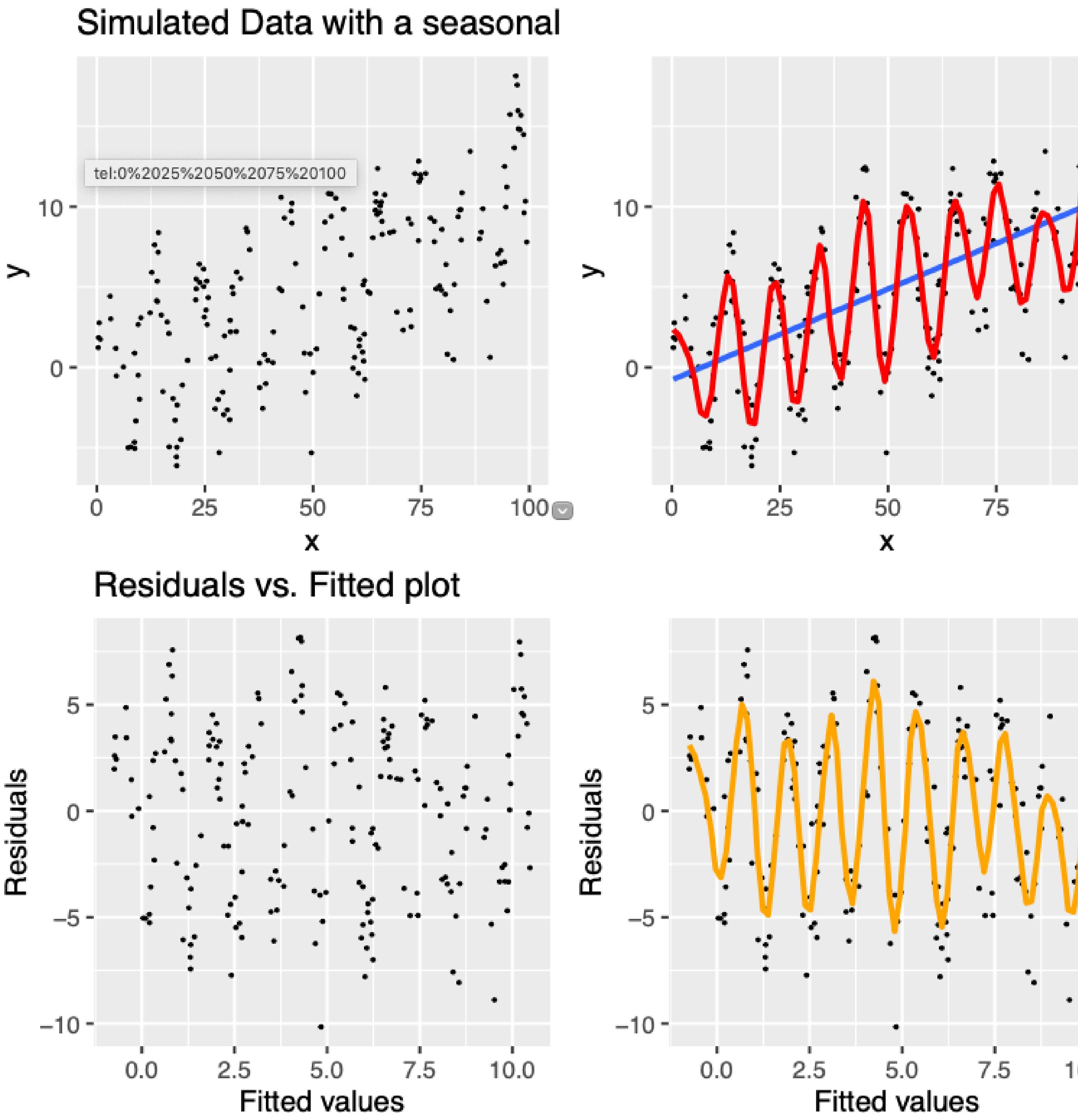


Figure 4.1: Example of data with a typical seasonal variation (up and down) coupled wtih a linear trend. The blue line gives the linear regression fit to the data, which clearly is not adequate. In comparison, if we used a non-parametric fit, we will get the red line as the fitted relationship. The residual plot retains pattern, given by orange line, indicating that the linear model is not appropriate in this case

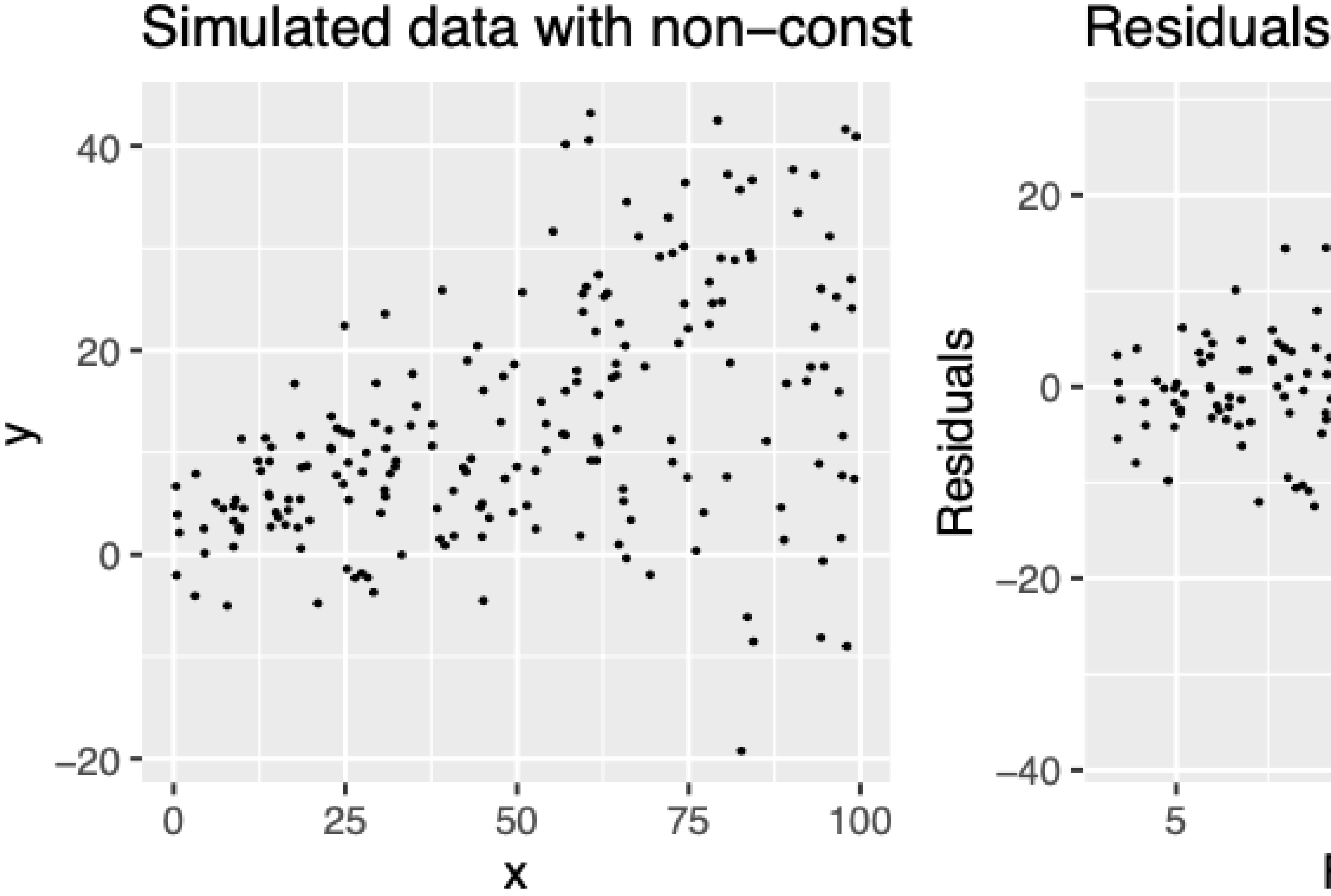


Figure 4.2: Example of non-constant variance

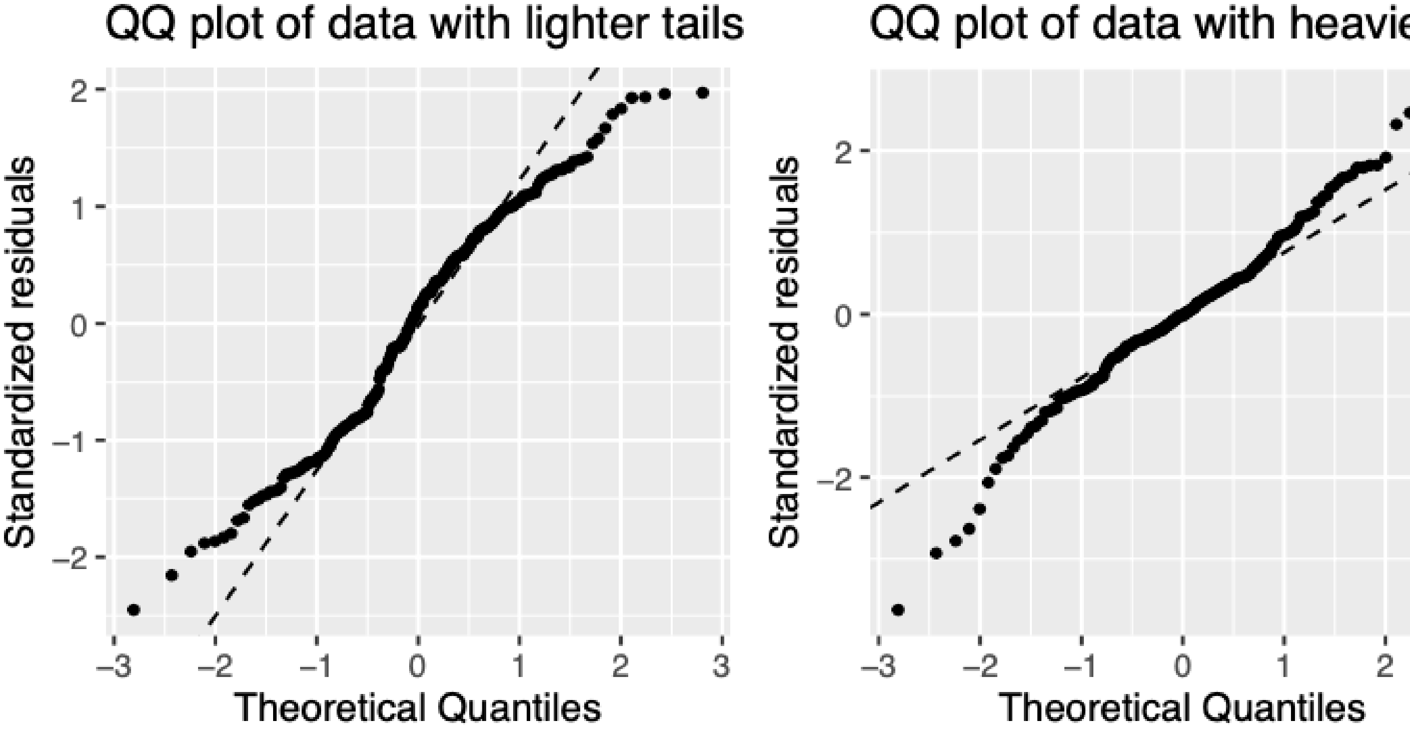


Figure 4.3: Example of residulas deviating from QQ plot, i.e. not following normal distribution. The residuals can deviate in both upper and lower tail. On the left tails are lighter meaning that they have smaller values that what would be expected, on the right there are heavier tails with values larger than expected

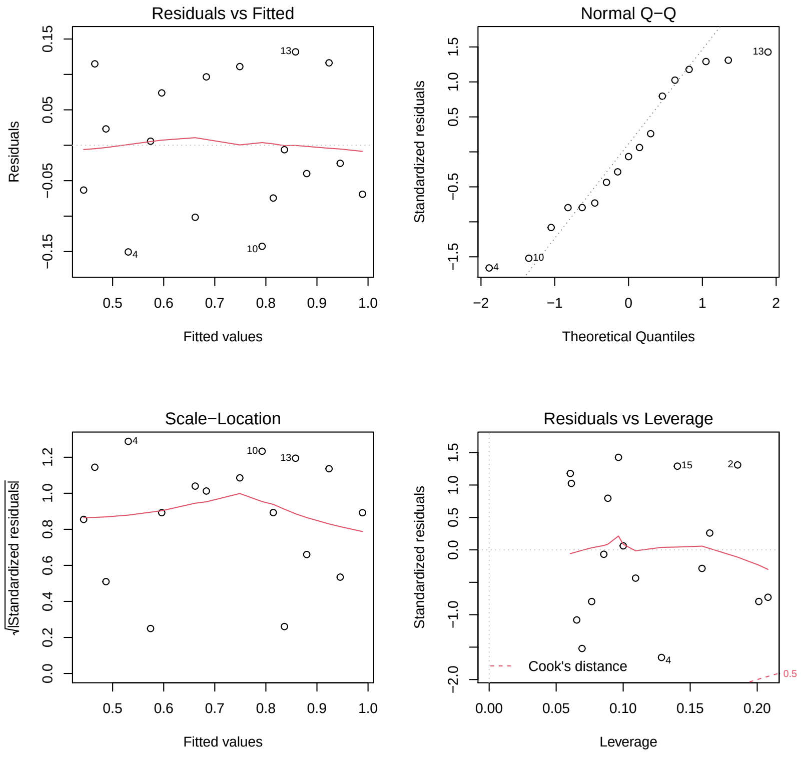
- If we removed i-th observation and compared the fitted value from the full model, say $\hat{y}_j$ to those obtained by removing this point, denoted $y_{\widehat{j(i)}}$ then
- observations with a high Cook's distance (measuring the effect of deleting a given observation) could be influential

Let's remove some observation with higher Cook's distance from protein data set, re-fit our model and compare the diagnostics plots

```
# observations to be removed (based on Residuals vs. Leverage plot)
obs <- c(18,7)

# fit models removing observations
model.2 <- lm(protein[-obs] ~ gestation[-obs])

# plot diagnostics plot
par(mfrow=c(2,2))
plot(model.2)
```



4.8 Selecting best model

- We have learned what linear models are, how to find estimates and interpret model coefficients and how to check for the overall relationship between response and predictors. We also know how to assess model fit, check model assumptions and find potential outliers. Given a set of predictors, e.g. many genes, how do we arrive at the best model?
- As a rule of thumb, we want a model that **fits the data best and is as simple as possible**, meaning it contains only relevant predictors.
- In practice, this means, that for smaller data sets, e.g. with up to 10 predictors, one works with **manually** trying different models, including different subsets of predictors, interactions terms and/or their transformations.
- When the number of predictors is large, one can try **automated approaches of feature selection** like forward selection or stepwise regression, the last one demonstrated in the exercises below.
- Finally, as we will learn later in the course, we can use **regularization techniques** that allow including all parameters in the model but constrain (regularizes) coefficient estimates towards zero for the less relevant predictors, preventing building complex models and thus overfitting.

5 Exercises (Model diagnostics)

Data for exercises are on Canvas under Files -> data_exercises –> linear-models

Exercise 5.1 (Brozek score)**.** Researchers collected age, weight, height and 10 body circumference measurements for 252 men in an attempt to find an alternative way of calculate body fat as oppose to measuring someone weight and volume, the latter one by submerging in a water tank. Is it possible to predict body fat using easy-to-record measurements?

Use lm() function and fit a linear method to model brozek, score estimate of percent body fat

- find R^2 and $R^2(adj)$
- assess the diagnostics plots to check for model assumptions
- delete observation #86 with the highest Cook's distance and re-fit the model (model.clean)
- look at the model summary. Are all variables associated with brozek score?
- try improving the model fit by removing variables with the highest p-value first and re-fitting the model until all the variables are significantly associated with the response (p value less than 0.1); note down the $R^2(adj)$ values while doing so
- compare the output models for model.clean and final model

To access and preview the data:

```
data(fat, package = "faraway")
```

Answers to selected exercises

Solution. Exercise 5.1

```
# access and preview data
data(fat, package = "faraway")
head(fat)
##   brozek siri density age weight height adipos  free neck chest abdom   hip
## 1   12.6 12.3  1.0708  23 154.25  67.75   23.7 134.9 36.2  93.1  85.2  94.5
## 2    6.9  6.1  1.0853  22 173.25  72.25   23.4 161.3 38.5  93.6  83.0  98.7
## 3   24.6 25.3  1.0414  22 154.00  66.25   24.7 116.0 34.0  95.8  87.9  99.2
## 4   10.9 10.4  1.0751  26 184.75  72.25   24.9 164.7 37.4 101.8  86.4 101.2
## 5   27.8 28.7  1.0340  24 184.25  71.25   25.6 133.1 34.4  97.3 100.0 101.9
## 6   20.6 20.9  1.0502  24 210.25  74.75   26.5 167.0 39.0 104.5  94.4 107.8
##   thigh knee ankle biceps forearm wrist
## 1  59.0 37.3  21.9   32.0    27.4  17.1
## 2  58.7 37.3  23.4   30.5    28.9  18.2
## 3  59.6 38.9  24.0   28.8    25.2  16.6
## 4  60.1 37.3  22.8   32.4    29.4  18.2
## 5  63.2 42.2  24.0   32.2    27.7  17.7
## 6  66.0 42.0  25.6   35.7    30.6  18.8

# fit linear regression model
model.all <- lm(brozek ~ age + weight + height + neck + abdom + hip + thigh + knee + ankle +

# print model summary
print(summary(model.all))
##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + abdom +
##     hip + thigh + knee + ankle + biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2664  -2.5658  -0.0798   2.8976   9.3204
##
```

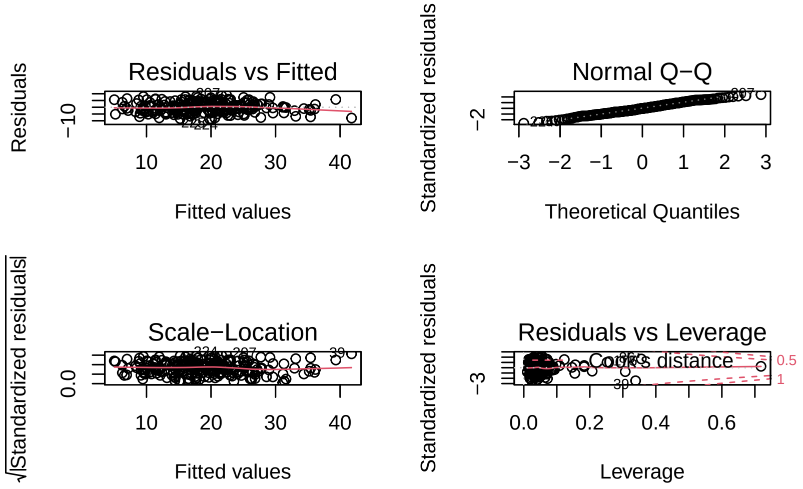
```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.063433  14.489336  -1.178  0.24011
## age           0.056520   0.029888   1.891  0.05983 .
## weight       -0.085513   0.045170  -1.893  0.05954 .
## height       -0.059703   0.086695  -0.689  0.49171
## neck         -0.439315   0.214802  -2.045  0.04193 *
## abdom         0.875779   0.070589  12.407  < 2e-16 ***
## hip          -0.192118   0.132655  -1.448  0.14885
## thigh         0.237304   0.131793   1.801  0.07303 .
## knee         -0.006595   0.222832  -0.030  0.97642
## ankle         0.164831   0.204681   0.805  0.42144
## biceps        0.149530   0.157693   0.948  0.34397
## forearm       0.424885   0.182801   2.324  0.02095 *
## wrist        -1.474317   0.494475  -2.982  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.98 on 239 degrees of freedom
## Multiple R-squared:  0.7489, Adjusted R-squared:  0.7363
## F-statistic:  59.4 on 12 and 239 DF,  p-value: < 2.2e-16

# diagnostics plots
par(mfrow=c(2,2))
plot(model.all)

# remove potentially influential observations
obs <- c(86)
fat2 <- fat[-obs, ]

# re-fit the model
model.clean <- lm(brozek ~ age + weight + height + neck + abdom + hip + thigh + knee + ankle

# diagnostics plots
par(mfrow=c(2,2))
plot(model.clean)
```



```
# model summary
print(summary(model.clean))
##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + abdom +
##     hip + thigh + knee + ankle + biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2664  -2.5658  -0.0798   2.8976   9.3204
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.063433  14.489336  -1.178  0.24011
## age           0.056520   0.029888   1.891  0.05983 .
## weight       -0.085513   0.045170  -1.893  0.05954 .
## height       -0.059703   0.086695  -0.689  0.49171
## neck         -0.439315   0.214802  -2.045  0.04193 *
## abdom         0.875779   0.070589  12.407  < 2e-16 ***
## hip          -0.192118   0.132655  -1.448  0.14885
## thigh         0.237304   0.131793   1.801  0.07303 .
## knee         -0.006595   0.222832  -0.030  0.97642
## ankle         0.164831   0.204681   0.805  0.42144
## biceps        0.149530   0.157693   0.948  0.34397
## forearm       0.424885   0.182801   2.324  0.02095 *
## wrist        -1.474317   0.494475  -2.982  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.98 on 239 degrees of freedom
## Multiple R-squared:  0.7489, Adjusted R-squared:  0.7363
## F-statistic:  59.4 on 12 and 239 DF,  p-value: < 2.2e-16

# re-fit the model (no height)
model.red1 <- lm(brozek ~ age + weight + neck + abdom + hip + thigh + knee + ankle + biceps
print(summary(model.red1))
##
## Call:
## lm(formula = brozek ~ age + weight + neck + abdom + hip + thigh +
##     knee + ankle + biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2830  -2.6162  -0.1017   2.8789   9.3713
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.66569   11.97691  -1.892  0.05963 .
## age           0.05948    0.02954   2.013  0.04521 *
## weight       -0.09829    0.04114  -2.389  0.01765 *
## neck         -0.43444    0.21445  -2.026  0.04389 *
## abdom         0.88762    0.06839  12.979  < 2e-16 ***
## hip          -0.17180    0.12919  -1.330  0.18483
## thigh         0.25327    0.12960   1.954  0.05183 .
## knee         -0.02318    0.22128  -0.105  0.91665
## ankle         0.17300    0.20411   0.848  0.39752
## biceps        0.15695    0.15715   0.999  0.31894
## forearm       0.43091    0.18239   2.363  0.01895 *
## wrist        -1.51011    0.49120  -3.074  0.00235 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.976 on 240 degrees of freedom
## Multiple R-squared:  0.7484, Adjusted R-squared:  0.7369
## F-statistic:  64.9 on 11 and 240 DF,  p-value: < 2.2e-16

# re-fit the model (no knee)
model.red2 <- lm(brozek ~ age + weight + neck + abdom + hip + thigh + ankle + biceps + forea
print(summary(model.red2))
##
## Call:
## lm(formula = brozek ~ age + weight + neck + abdom + hip + thigh +
##     ankle + biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2552  -2.5979  -0.1133   2.8693   9.3584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.08716   11.25781  -2.051  0.04137 *
## age           0.05875    0.02864   2.051  0.04134 *
## weight       -0.09965    0.03897  -2.557  0.01117 *
## neck         -0.43088    0.21131  -2.039  0.04253 *
## abdom         0.88875    0.06740  13.186  < 2e-16 ***
## hip          -0.17231    0.12884  -1.337  0.18234
## thigh         0.24942    0.12403   2.011  0.04544 *
## ankle         0.16946    0.20089   0.844  0.39974
## biceps        0.15847    0.15616   1.015  0.31123
## forearm       0.42946    0.18150   2.366  0.01876 *
## wrist        -1.51470    0.48823  -3.102  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.968 on 241 degrees of freedom
## Multiple R-squared:  0.7484, Adjusted R-squared:  0.738
## F-statistic: 71.69 on 10 and 241 DF,  p-value: < 2.2e-16

# re-fit the model (no ankle)
model.red3 <- lm(brozek ~ age + weight + neck + abdom + hip + thigh  + biceps + forearm + wr
print(summary(model.red3))
##
## Call:
```

```
## lm(formula = brozek ~ age + weight + neck + abdom + hip + thigh +
##     biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0740  -2.5615  -0.1021   2.7999   9.3199
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.61247   10.86240  -1.898   0.0589 .
## age           0.05727    0.02857   2.004   0.0461 *
## weight       -0.09141    0.03770  -2.424   0.0161 *
## neck         -0.45458    0.20931  -2.172   0.0308 *
## abdom         0.88098    0.06673  13.203   <2e-16 ***
## hip          -0.17575    0.12870  -1.366   0.1733
## thigh         0.25504    0.12378   2.061   0.0404 *
## biceps        0.15178    0.15587   0.974   0.3311
## forearm       0.42805    0.18138   2.360   0.0191 *
## wrist        -1.40948    0.47175  -2.988   0.0031 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.965 on 242 degrees of freedom
## Multiple R-squared:  0.7477, Adjusted R-squared:  0.7383
## F-statistic: 79.67 on 9 and 242 DF,  p-value: < 2.2e-16

# re-fit the model (no biceps)
model.red4 <- lm(brozek ~ age + weight + neck + abdom + hip + thigh  + forearm + wrist, data
print(summary(model.red4))
##
## Call:
## lm(formula = brozek ~ age + weight + neck + abdom + hip + thigh +
##     forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0574  -2.7411  -0.1912   2.6929   9.4977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -20.06213   10.84654  -1.850  0.06558 .
## age           0.05922    0.02850   2.078  0.03876 *
## weight       -0.08414    0.03695  -2.277  0.02366 *
## neck         -0.43189    0.20799  -2.077  0.03889 *
## abdom         0.87721    0.06661  13.170  < 2e-16 ***
## hip          -0.18641    0.12821  -1.454  0.14727
## thigh         0.28644    0.11949   2.397  0.01727 *
## forearm       0.48255    0.17251   2.797  0.00557 **
## wrist        -1.40487    0.47167  -2.978  0.00319 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.965 on 243 degrees of freedom
## Multiple R-squared:  0.7467, Adjusted R-squared:  0.7383
## F-statistic: 89.53 on 8 and 243 DF,  p-value: < 2.2e-16

# re-fit the model (no hip)
model.red5 <- lm(brozek ~ age + weight + neck + abdom  + thigh  + forearm + wrist, data = fa
print(summary(model.red5))
##
## Call:
## lm(formula = brozek ~ age + weight + neck + abdom + thigh + forearm +
##     wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0193  -2.8016  -0.1234   2.9387   9.0019
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -30.17420    8.34200  -3.617 0.000362 ***
## age           0.06149    0.02852   2.156 0.032047 *
## weight       -0.11236    0.03151  -3.565 0.000437 ***
## neck         -0.37203    0.20434  -1.821 0.069876 .
## abdom         0.85152    0.06437  13.229  < 2e-16 ***
## thigh         0.20973    0.10745   1.952 0.052099 .
## forearm       0.51824    0.17115   3.028 0.002726 **
## wrist        -1.40081    0.47274  -2.963 0.003346 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.974 on 244 degrees of freedom
## Multiple R-squared:  0.7445, Adjusted R-squared:  0.7371
## F-statistic: 101.6 on 7 and 244 DF,  p-value: < 2.2e-16

# compare model.clean and final model
print(summary(model.clean))
##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + abdom +
##     hip + thigh + knee + ankle + biceps + forearm + wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2664  -2.5658  -0.0798   2.8976   9.3204
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.063433  14.489336  -1.178  0.24011
## age           0.056520   0.029888   1.891  0.05983 .
## weight       -0.085513   0.045170  -1.893  0.05954 .
## height       -0.059703   0.086695  -0.689  0.49171
## neck         -0.439315   0.214802  -2.045  0.04193 *
## abdom         0.875779   0.070589  12.407  < 2e-16 ***
## hip          -0.192118   0.132655  -1.448  0.14885
## thigh         0.237304   0.131793   1.801  0.07303 .
## knee         -0.006595   0.222832  -0.030  0.97642
## ankle         0.164831   0.204681   0.805  0.42144
## biceps        0.149530   0.157693   0.948  0.34397
## forearm       0.424885   0.182801   2.324  0.02095 *
## wrist        -1.474317   0.494475  -2.982  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.98 on 239 degrees of freedom
## Multiple R-squared:  0.7489, Adjusted R-squared:  0.7363
## F-statistic:  59.4 on 12 and 239 DF,  p-value: < 2.2e-16
print(summary(model.red5))
##
```

```
## Call:
## lm(formula = brozek ~ age + weight + neck + abdom + thigh + forearm +
##     wrist, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0193  -2.8016  -0.1234   2.9387   9.0019
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -30.17420    8.34200  -3.617 0.000362 ***
## age           0.06149    0.02852   2.156 0.032047 *
## weight       -0.11236    0.03151  -3.565 0.000437 ***
## neck         -0.37203    0.20434  -1.821 0.069876 .
## abdom         0.85152    0.06437  13.229  < 2e-16 ***
## thigh         0.20973    0.10745   1.952 0.052099 .
## forearm       0.51824    0.17115   3.028 0.002726 **
## wrist        -1.40081    0.47274  -2.963 0.003346 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.974 on 244 degrees of freedom
## Multiple R-squared:  0.7445, Adjusted R-squared:  0.7371
## F-statistic: 101.6 on 7 and 244 DF,  p-value: < 2.2e-16
```

Note: we have just run a very simple feature selection using stepwise regression. In this method, using backward elimination, we build a model containing all the variables and remove them one by one based on defined criteria (here we have used p-values) and we stop when we have a justifiable model or when removing a predictor does not change the chosen criterion significantly.