Introduction to biostatistics and machine learning

Olga Dethlefsen, Eva Freyhult, Bengt Sennblad, Payam Emami

2020-11-14

Contents

Preface

This "bookdown" book 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 in need of biostatistical skills within Swedish universities. The materials are geared towards life scientists wanting to be able to understand and use basic statistical and machine learning methods. It may also suits those already applying biostatistical methods but who have never gotten a chance to reflect on the basic statistical concepts, such as the commonly misinterpreted p-value.

More about the course https://nbisweden.github.io/workshop-mlbiostatistics/

Part I

Preliminary Mathematics

Chapter 1

Mathematical notations

Aims

- to recapitulate the basic notations and conventions used in mathematics and statistics

Learning outcomes

- to recognize natural numbers, integrals and real numbers
- to understand the differences between variables and constants
- to use symbols, especially Sigma and product notations, to represent mathematical operations

1.1 Numbers

- **Natural numbers, N**: numbers such as 0, 1, 3, …
- **Integers, Z**: include negative numbers …, -2, -1, 0, 1, 2
- **Rational numbers**: numbers that can be expressed as a ratio two integers, i.e. in a form $\frac{a}{b}$, where a and b are integers, and $b \neq 0$
- **Real numbers, R**: include both rational and irrational numbers
- **Reciprocal** of any number is found by diving 1 by the number, e.g. reciprocal of 5 is $\frac{1}{5}$
- **Absolute value** of a number can be viewed as its distance from zero, e.g. the absolute value of 6 is 6, written as $|6| = 6$ and absolute value of -5 is 5, written as $|-5| = 5$
- **Factorial** of a non-negative integer number n is denoted by $n!$ and it is a product of all positive integers less than or equal to n, e.g. $4! = 4{\cdot}3{\cdot}2{\cdot}1 = 24$

1.2 Variables, constants and letters

Mathematics gives us a precise language to communicate different concepts and ideas. To be able to use it it is essential to learn symbols and understand how they are used to represent physical quantities as well as understand the rules and conventions that have been developed to manipulate them.

- **variables**: things that can vary, e.g. temperature and time
- **constants**: fixed and unchanging quantities used in certain calculations, e.g. 3.14159
- in principle one could freely choose letters and symbols to represent variables and constants, but it is helpful and choose letters and symbols that have meaning in a particular context. Hence, we
- x, y, z, the end of the alphabet is reserved for variables
- a, b, c, the beginning of the alphabet is used to represent constants
- π, ω and Greek letters below are used frequently used to represent common constant, e.g. $\pi = 3.14159$

Table 1.1: Uppercase and lowercase letters of the Greek alphabet

Letter	Upper case	Lower case	Letter	Upper case	Lower case
Alpha	A	α	Nu	N	ν
Beta	B	β	Xi	Ξ	ξ
Gamma	Γ	γ	Omicron	O	o
Delta	Δ	δ	Pi	Π	π
Epsilon	E	ϵ	Rho	P	ρ
Zeta	Z	ζ	Sigma	Σ	σ
Eta	H	η	Tau	T	τ
Theta	Θ	θ	Upsilon	Y	υ
Iota	i	ι	Phi	Φ	ϕ
Kappa	K	κ	Chi	Γ	γ
Lambda	Γ	γ	Psi	Ψ	ψ
Mu	M	μ	Omega	Ω	ω

1.3 A precise language

- Mathematics is a precise language meaning that a special attention has to be paid to the exact position of any symbol in relation to other.
- Given two symbols x and y, xy and x^y and x_y can mean different things
- xy stands for multiplication, x^y for superscript and x_y for subscript

1.4 Using symbols

If the letters x and y represent two numbers, then:

- their **sum** is written as $x + y$
- subtracting y from x is $x - y$, known also as **difference**
- to multiply x and y we written as $x \cdot y$ or also with the multiplication signed omitted as xy. The quantity is known as **product of x and y**
- multiplication is **associative**, e.g. when we multiply three numbers together, $x \cdot y \cdot z$, the order of multiplication does not matter, so $x \cdot y \cdot z$ is the same as $z \cdot x \cdot y$ or $ycdotz \cdot x$
- division is denoted by $\frac{x}{y}$ and mans that x is divided by y. In this expression x, on the top, is called **numerator** and y, on the bottom, is called **denominator**
- division by 1 leaves any number unchanged, e.g. $\frac{x}{1} = x$ and division by 0 is not allowed

Equal sign

- the equal sign = is used in **equations**, e.g. $x - 5 = 0$ or $5x = 1$
- the equal sign = can be also used in **formulae**. Physical quantities are related through a formula in many fields, e.g. the formula $A = \pi r^2$ relates circle area A to its radius r and the formula $s = \frac{d}{t}$ defines speed as distance d divided by time t
- the equal sign = is also used in identities, expressions true for all values of the variable, e.g. $(x - 1)(x - 1) = (x^2 - 1)$
- opposite to the equal sign is "is not equal to" sign $\neq$, e.g. we can write $1 + 2 \neq 4$

Sigma and Product notation

- the Σ notation, read as **Sigma notation**, provides a convenient way of writing longs sums, e.g. the sum of $x_1 + x_2 + x_3 + ... + x_{20}$ is written as $\sum_{i=1}^{i=20} x_i$
- the Π notation, read as **Product notation**, provides a convenient way of writing longs products, e.g. $x_1 \cdot x_2 \cdot x_3 \cdot ... \cdot x_{20}$ is written as $\prod_{i=1}^{i=20} x_i$

1.5 Inequalities

Given any two real numbers a and b there are three mutually exclusive possibilities:

- $a > b$, meaning that a is greater than b
- $a < b$, meaning that a is less than b
- $a = b$, meaning that a is equal to b

Strict and weak

- inequality in $a > b$ and $a < b$ is **strict**
- as oppose to **weak** inequality denoted as $a \geq b$ or $a \leq b$

Some useful relations are:

- if $a > b$ and $b > c$ then $a > c$
- if $a > b$ then $a + c > b$ for any c
- if $a > b$ then $ac > bc$ for any positive c
- if $a > b$ then $ac < bc$ for any negative c

1.6 Indices and powers

- **Indices**, also known as **powers** are convenient when we multiply a number by itself several times
- e.g. $5 \cdot 5 \cdot 5$ is written as 5^3 and $4 \cdot 4 \cdot 4 \cdot 4 \cdot 4$ is written as 4^5
- in the expression x^y, x is called the *base* and y is called *index* or *power*

The laws of indices state:

- $a^m \cdot a^n = a^{m+n}$
- $\frac{a^m}{a^n} = a^{m-n}$
- $(a^m)^n = a^{m \cdot n}$

Rules derived from the laws of indices:

- $a^0 = 1$
- $a^1 = a$

Negative and fractional indices:

- $a^{-m} = \frac{1}{a^m}$ e.g. $5^{-2} = \frac{1}{5^2} = \frac{1}{25}$ for negative indices
- e.g. $4^{\frac{1}{2}} = \sqrt{4}$ or $8^{\frac{1}{3}} = \sqrt[3]{8}$ for fractional indices

---

1.7 Exercises: notations

Exercise 1.1. Classify numbers as natural, integers or real. If reall, specify if they are rational or irrational.

a) $\frac{1}{3}$
b) 2
c) $\sqrt{4}$
d) 2.3
e) π
f) $\sqrt{5}$
g) -7
h) 0
i) 0.25

Exercise 1.2. Classify below descriptors as variables or constants. Do you know the letters or symbols commonly used to represent these?

a) speed of light in vacuum
b) mass of an apple
c) volume of an apple
d) concentration of vitamin C in an apple
e) distance from Stockholm central station to Uppsala central station
f) time on the train to travel between the above stations
g) electron charge

Exercise 1.3. Write out explicitly what is meant by the following:

a) $\sum_{i=1}^{i=6} k_i$

b) $\prod_{i=1}^{i=6} k_i$

c) $\sum_{i=1}^{i=6} i^k$

d) $\prod_{i=1}^{i=3} i^k$

e) $\sum_{i=1}^{n} i$

f) $\sum_{i=1}^{i=4} (i+1)^k$

g) $\prod_{i=1}^{i=4} (k+1)^i$

h) $\prod_{i=0}^{n} i$

Exercise 1.4. Use Sigma or Product notation to represent the long sums and products below:

a) $1+2+3+4+5+6$
b) $2^2+3^2+4^2+5^2$
c) $4 \cdot 5 \cdot 6 \cdot 7 \cdot 8$
d) $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+...+\frac{1}{n}$
e) $2-2^2+2^3-2^4+...+2^n$
f) $3+6+9+12+\cdots+60$
g) $3x+6x^2+9x^3+12x^4+...+60x^{20}$
h) $3x \cdot 6x^2 \cdot 9x^3 \cdot 12x^4 \cdot ... \cdot 60x^{20}$

Answers to selected exercises (notations)

Exr. 1.1

a) real, rational
b) natural and integers, integers include natural numbers
c) $\sqrt{4} = 2$ so it is a natural number and/subset of integers
d) real number, rational as it could be written as $\frac{23}{10}$
e) real number, irrational as it cannot be explained by a simple fraction
f) real number, irrational as it cannot be explained by a simple fraction
g) integer, non a natural number as these do not include negative numbers
h) natural number, although there is some argument about it as some define natural numbers as positive integers starting from 1, 2 etc. while others include 0.
i) real, rational number, could be written as $\frac{25}{100}$

Exr. 1.2

a) constant, speed of light in vacuum is a constant, denoted c with $c = 299792458\frac{m}{s}$
b) variable, mass of an apple is a variable, different for different apple sizes, for instance 138 grams, denoted as $m = 100g$
c) variable, like mass volume can be different from apple to apple, denoted as V, e.g. $V = 200cm^3$
d) variable, like volume and mass can vary, denoted as ρ_i and defined as $\rho_i = \frac{m}{V}$. So given 6.3 milligrams of vitamin C in our example apple, we have $\rho_i = \frac{0.0063}{2}\frac{g}{cm^3} = 0.0000315\frac{g}{cm^3}$ concentration of vitamin D
e) constant, the distance between Stockholm and Uppsala is fixed; it could be a variable though if we were to consider an experiment on a very long time scale; distance is often denoted in physics as d
f) variable, time on the train to travel between the stations varies, often denoted as t with speed being calculated as $s = \frac{d}{t}$
g) constant, electron charge is $e = 1.60217663 \cdot 10^{-19}C$

Exr. 1.3

a) $\sum_{i=1}^{i=6} k_i = k_1 + k_2 + k_3 + k_4 + k_5 + k_6$

b) $\prod_{i=1}^{i=6} k_i = k_1 \cdot k_2 \cdot k_3 \cdot k_4 \cdot k_5 \cdot k_6$

c) $\sum_{i=1}^{i=3} i^k = 1^k + 2^k + 3^k$

d) $\prod_{i=1}^{i=3} i^k = 1^k \cdot 2^k \cdot 3^k$

e) $\sum_{i=1}^{n} i = 1 + 2 + 3 + ... + n$ we are using dots (...) to represent all the number until n. Here, thanks to Gauss we can also write $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, i.e. Gauss formula for sum of first n natural numbers

Exr. 1.4

a) $1 + 2 + 3 + 4 + 5 + 6 = \sum_{k=1}^{6} k$

b) $2^2 + 3^2 + 4^2 + 5^2 = \sum_{x=2}^{5} x^2$

c) $4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 = \prod_{x=4}^{8} x$

d) $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + ... + \frac{1}{n} = \sum_{k=1}^{n} \frac{1}{k}$

Chapter 2

Sets

Aims

- to introduce sets and basic operations on sets

Learning outcomes

- to be able to explain what a set is
- to be able to construct new sets from given sets using the basic set operations
- to be able to use Venn diagrams to shows all possible logical relations between two and three sets

2.1 Definitions

- **set**: a well-defined collection of distinct objects, e.g. $S = \{2, 4, 6\}$
- **elements**: the objects that make up the set are also known as **elements** of the set
- if x is an element of S, we say that x belongs to S and write $x \in S$ and if x is not an element of S we say that x does not belong to S and write $x \notin S$
- a set may contain **finitely** many or **infinitely** many elements

- **subset,** $\subseteq$: if every element of set A is also in B, then A is said to be a subset of B, written as $A \subseteq B$ and pronounced A is contained in B, e.g. $A \subseteq B$, when $A = \{2, 4, 6\}$ and $\$ = B = \{2, 4, 6, 8, 10\}$. Every set is a subset if itself.
- **superset**: for our outs A and B we can also say that B is a **superset** of A and write $B \supset A$

- **cardinality**: the number of elements within a set S, denoted as $|S|$

- **empty set,** $\emptyset$: is a unique set with no members, denoted by $E = \emptyset$ or $E = \{\}$. The empty set is a subset of very set.

2.2 Basic set operations

- **union of two sets,** $\cup$: two sets can be "added" together, the union of A and B, written as $A \cup B$, e.g. $\{1,2\} \cup \{2,3\} = \{1,2,3\}$ or $\{1,2,3\} \cup \{1,4,5,6\} = \{1,2,3,4,5,6\}$
- **intersection of two sets,** $\cap$: a new set can be constructed by taking members of two sets that are "in common", written as $A \cap B$, e.g. $\{1,2,3,4,5,6\} \cap \{2,3,7\} = \{2,3\}$ or $\{1,2,3\} \cap \{7\} = \{\emptyset\}$

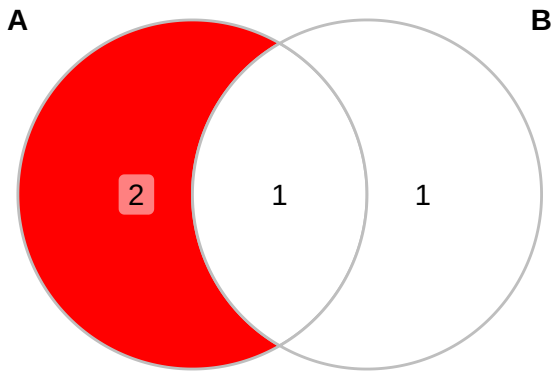
- **complement of a set,** A', A^c: are the elements not in A
- **difference of two sets,** : two sets can be "substracted", denoted by $A \ B$, by taking all elements that are members of A but are not members of B, e.g. $\{1,2,3,4\} \ \{1,3\} = \{2,4\}$. This is also in other words a relative complement of A with respect to B.

- **partition of a set**: a partition of a set S is a set of nonempty subset of S, such that every element x in S is in exactly one of these subsets. That is, the subset are pairwise *disjoint*, meaning no two sets of the partition contain elements in common, and the union of all the subset of the partition is S, e.g. Set $\{1,2,3\}$ has five partitions: i) $\{1\},\{2\},\{3\}$, ii) $\{1,2\},\{3\}$, iii) $\{1,3\},\{2\}$, iv) $\{1\},\{2,3\}$ and v) $\{1,2,3\}$

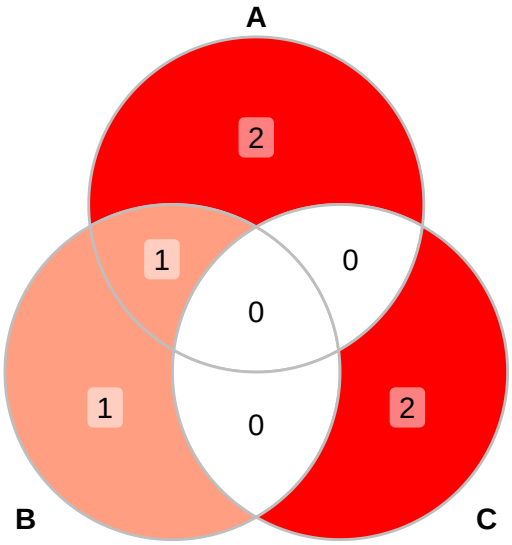
2.3 Venn diagrams

Venn diagram is a diagram that shows all possible logical relations between a finite collection of different sets. A Venn diagrams shows elements as points in the plane, and sets as regions inside closed curves. A Venn diagram consists of multiple overlapping closed curves, usually circles, each representing a set.

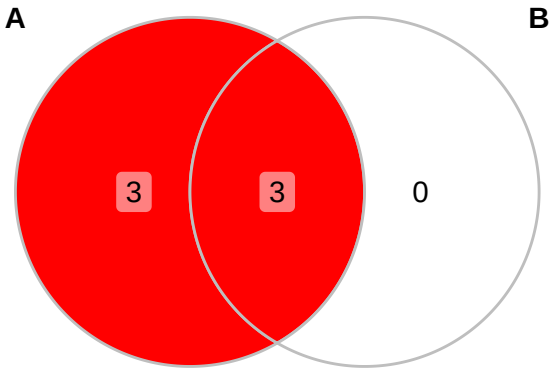
E.g. given $A = \{1,2,5\}$ and $B = \{1,6\}$ Venn diagram of A and B:



And given $A = \{1, 2, 5\}$, $B = \{1, 6\}$ and $C = \{4, 7\}$ Venn diagram of A, B and C:



And given $A = \{1, 2, 3, 4, 5, 6\}$ and $B = \{2, 4, 6\}$ Venn diagram of A and B:



2.4 Exercises: sets

Exercise 2.1. Given set $S = \{1, 2, 3, 4, 5, 6\}$:

a) what is the subset T of S consisting of its even elements?
b) what is the complement T^c?
c) what is the subset U of S containing of the prime numbers in S?
d) what is the intersection $T \cap U$?
e) what is the union of $T \cup U$?
f) what is the set difference $U \quad T$?

Exercise 2.2. Given set

$A = \{cat, elephant, dog, turtle, goldfish, hamster, parrot, tiger, guineapig, lion\}$

a) what is the subset D of A consiting of domesticated animals?
b) what is the subset C of A consiting of Felidae (cat) family?
c) what is the interection of D and C?
d) what is the complement of D, D^c?
e) what is the union of D and C?
f) what is the set difference of A C?
g) can you draw Venn diagram showing relationship between D and C?

Answers to selected exercises (sets)

Exr. 2.1

a) $T = \{2, 4, 6\}$
b) $T^c = \{1, 3, 5\}$, i.e. T^c contains all the elements of S not in T
c) $U = \{2, 3, 5\}$, the primes in S
d) $T \cap U = \{2\}$, common elements of T and U, i.e. the even and prime numbers
e) $T \cup U = \{2, 3, 4, 5, 6\}$
f) $U \quad T = \{3, 5\}$, consisting of the elements of U that are not in T

Chapter 3

Functions

Aims

- to revisit the concept of a function

Learning outcomes

- to be able to explain what function, function domain and function range are
- to be able to identify input, output, argument, independent variable, dependent variable
- to be able to evaluate function for a given value and plot the function

3.1 Definitions

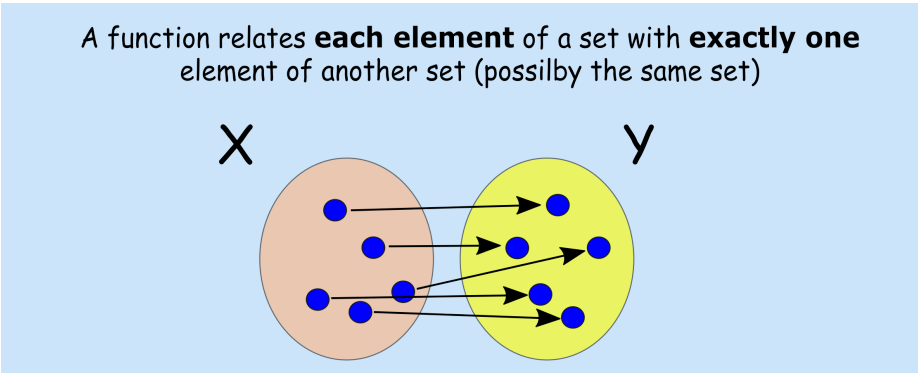


Figure 3.1: Formal function defition

- A **function**, $f(x)$, can be viewed as a rule that relates input x to an output $f(x)$
- In order for a rule to be a function it must produce a single output for any given input
- Input x is also known as **argument** of the function
- **Domain of a function**: the set of all values that the function "maps"
- **Range**: the set of all values that the function maps into

Many names are used interchangeably

Functions have been around for a while and there are many alternative names and writing conventions are being used. Common terms worth knowing:

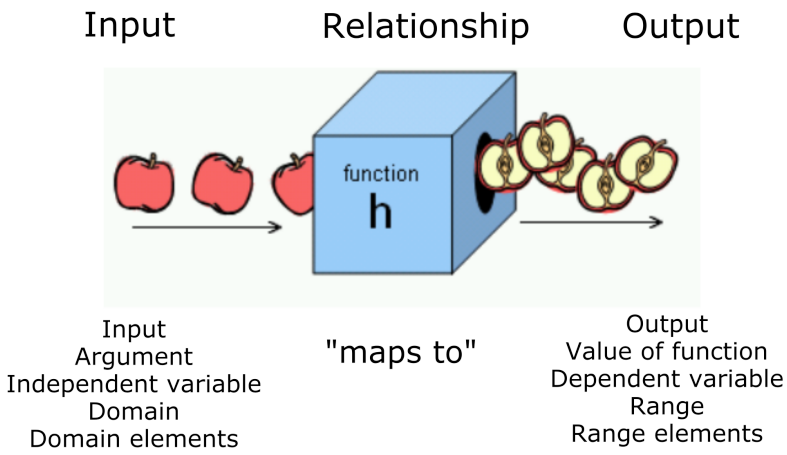


Figure 3.2: Common function terms

3.2 Evaluating function

To evaluate a function is to replace (substitute) its variable with a given number or expression. E.g. given a rule (function) that maps temperature measurements from Celsius to Fahrenheit scale:

$$f(x) = 1.8x + 32$$

where x is temperature measurements in Celsius and $f(x)$ is the associated value in Fahrenheit, we can find for a given temperature in Celsius corresponding temperature in Fahrenheit. Let's say we measure 10 Celsius degrees one autumn day in Uppsala and we want to share this information with a friend in USA. We can find the equivalent temperature in Fahrenheit by evaluating our function at

10, $f(10)$, giving us

$$f(10) = 1.8 \cdot 10 + 32 = 50$$

3.3 Plotting function

Function graphs are a convenient way of showing functions, by looking at the graph it is easier to notice function's properties, e.g. for which input values functions yields positive outcomes or whether the function is increasing or decreasing. To graph a function, one can start by evaluating function at different values for the argument x obtaining $f(x)$, plotting the points by plotting the pairs $(x, f(x))$ and connecting the dots. E.g. evaluating our above temperature rule at -20, -10, 0, 10, 20, 30 Celsius degrees results in:

x (Celsius degrees)	evaluates	f(x) (Farenheit degress)
-20	$f(-20) = 1.8 \cdot (-20) + 32$	-4
-10	$f(-20) = 1.8 \cdot (-10) + 32$	14
0	$f(-20) = 1.8 \cdot (0) + 32$	32
10	$f(-20) = 1.8 \cdot (10) + 32$	50
20	$f(-20) = 1.8 \cdot (20) + 32$	68
20	$f(-20) = 1.8 \cdot (30) + 32$	86

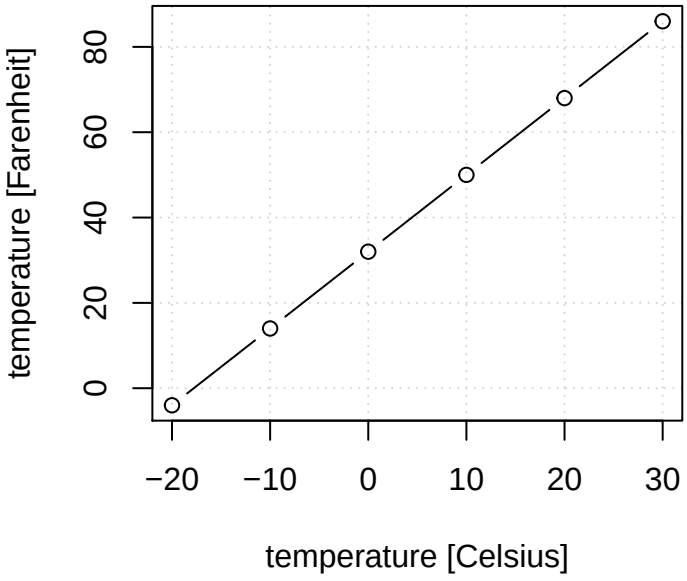


Figure 3.3: Graph of f(x) for the temeprature rule

3.4 Standard classes of functions

Algebraic function: functions that can be expressed as the solution of a polynomial equation with integer coefficients, e.g.

- constant function $f(x) = a$
- identity function $f(x) = x$
- linear function $f(x) = ax + b$
- quadratic function $f(x) = a + bx + cx^2$
- cubic function $fx() = a + bx + cx^2 + dx^3$

Transcedental functions: functions that are not algebraic, e.g.

- exponential function $f(x) = e^x$
- logarithimic function $f(x) = log(x)$
- trigonometric function $f(x) = -3sin(2x)$

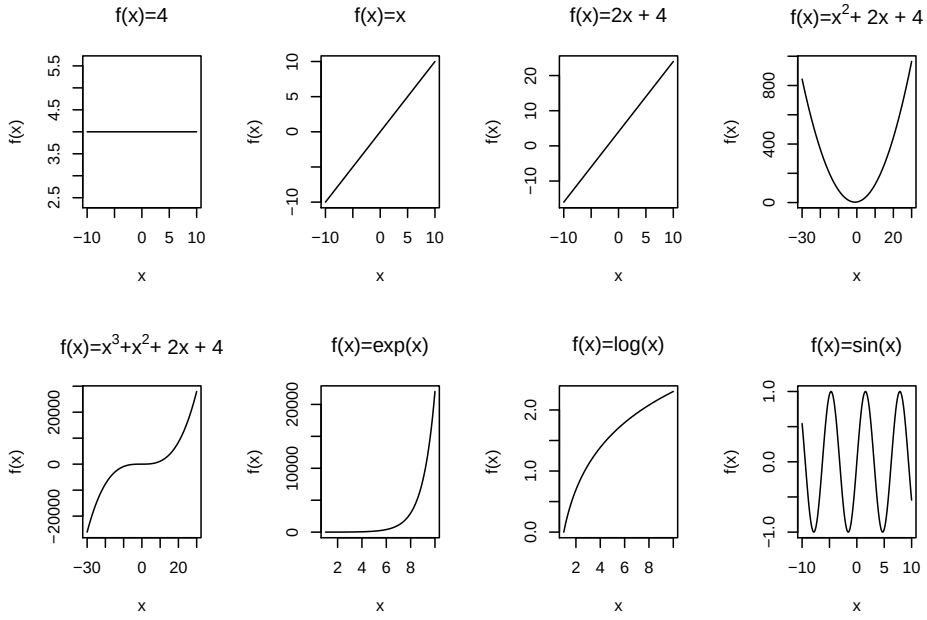


Figure 3.4: Examples of the standard classess of functions

3.5 Piecewise functions

A function can be in pieces, i.e. we can create functions that behave differently based on the input x value. They are useful to describe situations in w which a rule changes as the input value crosses certain "boundaries". E.g. a function value could be fixed in a given range and equal to the input value (identify function) for input values outside this range

$$f(x) = \begin{cases} 2 & \text{if } x \leq 1 \\ x & \text{if } x > 1 \end{cases} \tag{3.1}$$

The function can be split in many pieces, e.g. the personal training fee in SEK may depend whether the personal trainer is hired for an hour, two hours or three or more hours:

$$f(h) = \begin{cases} 500 & \text{if } h \leq 1 \\ 750 & \text{if } 1 < h \leq 2 \\ 500 + 250 \cdot h & \text{if } h > 2 \end{cases} \tag{3.2}$$

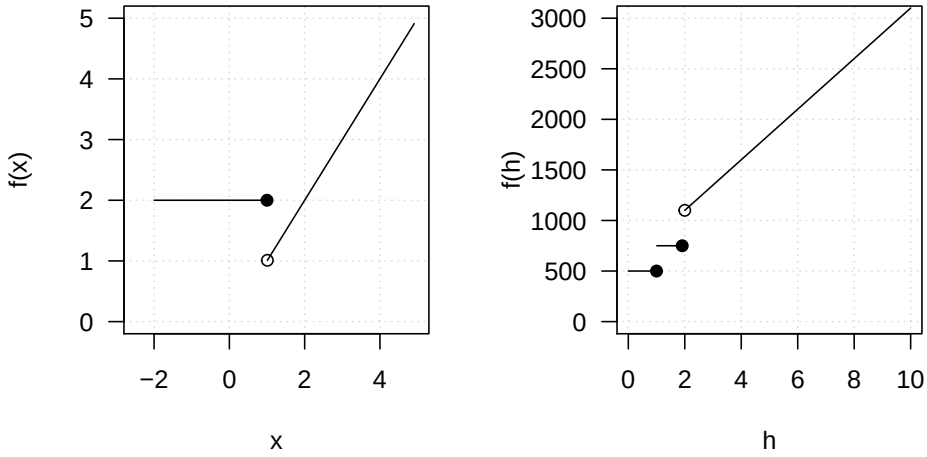


Figure 3.5: Examples of piece-wise functions

3.6 Exercises: functions

Exercise 3.1. Given the function for the personal trainer costs:

$$f(h) = \begin{cases} 500 & \text{if } h \leq 1 \\ 750 & \text{if } 1 < h \leq 2 \\ 500 + 250 \cdot h & \text{if } h > 2 \end{cases} \tag{3.3}$$

How much would you pay

a) for a 4-hours session? Evaluate function f(h) for value 4.
b) for a 2-hour session? Evalue function f(h) for value 2.

Exercise 3.2. A museum charges 50 SEK per person for a guided tour with a group of 1 to 9 people or a fixed 500 SEK fee for a group of 10 or more people. Write a function relating the number of people n to the cost C.

Exercise 3.3. Given function

$$f(x) = \begin{cases} x^2 & \text{if } x \leq 1 \\ 3 & \text{if } 1 < x \leq 2 \\ x & \text{if } x > 2 \end{cases} \tag{3.4}$$

a) sketch a graph of a function for $x \in (-4, 4)$, i.e.. for x between -4 and 4
b) evaluate function at f(1)
c) evaluate function at f(4)

Answers to selected exercises (functions)

Exr. 3.1

a) $f(4) = 500 + 250 \cdot 4 = 1500$
b) $f(2) = 750$ as $h \leq 2$ means less or equal to 2, that is including 2

Chapter 4

Differentiation

Aims

- introduce the concept of differentiation and rules of differentiation

Learning outcomes

- to be able to explain differentiation in terms of rate of change
- to be able to find derivatives in simple cases

4.1 Rate of change

- We are often interested in the rate at which some variable is changing, e.g. we may be interested in the rate at which the temperature is changing or the rate of water levels increasing
- Rapid or unusual changes may indicate that we are dealing with unusual situations, e.g. global warming or a flood
- Rates of change can be positive, negative or zero corresponding to a variable increasing, decreasing and non-changing

The function $f(x) = x^4 - 4x^3 - x^2 - e^{-x}$ changes at different rates for different values of x, e.g.

- between $x \in (-10, -9)$ the $f(x)$ is increasing at slightly higher pace than $x \in (5, 6)$
- between $x \in (-7, -5)$ the $f(x)$ is decreasing and
- between $x \in (0, 1)$ the $f(x)$ is not changing
- to be able to talk more precisely about the rate of change than just saying "large and positive" or "small and negative" change we need to quantify the changes, i.e. assign the rate of change an exact value
- **Differentiation** is a technique for calculating the rate of change of any function

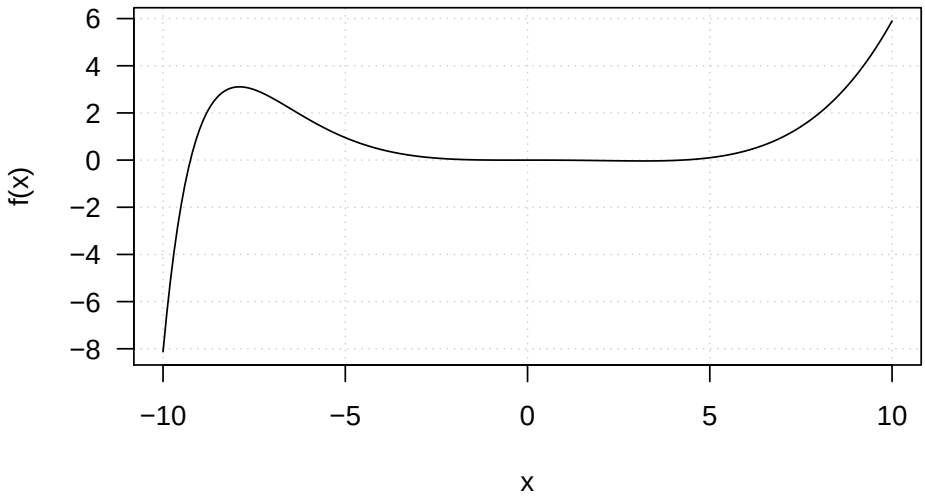


Figure 4.1: The function $f(x)$ changes at different rates for different values of x

4.2 Average rate of change across an interval

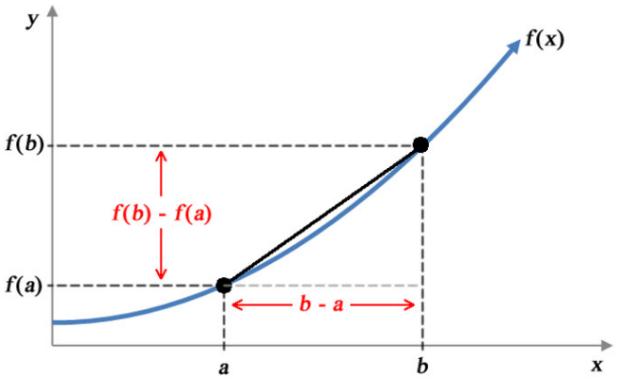


Figure 4.2: The average rate of change of $f(x)$ with respect to x over $[a, b]$ is equal to the slope of the secant line (in black)

To dive further into calculating the rate of change let's look at Figure 4.2 and define the *average rate of change* of a function across an interval. Figure 4.2 shows a function $f(x)$ with two possible argument values a and b marked and their corresponding function values $f(a)$ and $f(b)$.

Consider that x is increasing from a to b. The change in x is $b - a$, i.e. as x increases from a to b the function $f(x)$ increase from $f(a)$ to $f(b)$. The change in $f(x)$ is $f(b) - f(a)$ and the average rate of change of y across the $[a, b]$ interval

is:

$$\frac{change\ in\ y}{change\ in\ x} = \frac{f(b) - f(a)}{b - a} \tag{4.1}$$

E.g. let's take a quadratic function $f(x) = x^2$ and calculate the average rate of change across the interval $[1, 4]$.

- The change in x is $4-1$ and the change in $f(x)$ is $f(4)-f(1) = 4^2-1^2 = 16-1 = 15$. So the average rate of change is $\frac{15}{3} = 5$. What does this mean? It means that across the interval $[1, 4]$ on average the $f(x)$ value increases by 5 for every 1 unit increase in x.
- If we were to look at the average rate of change across the interval $[-2, 0]$ we would get $\frac{f(0)-f(-2)}{0-(-2)} = \frac{0-(-2)^2}{2} = \frac{-4}{2} = -2$. Here, over the $[-2, 0]$ on average the $f(x)$ value decreases by 2 for every 1 unit increase in x.
- Looking at the graph of $f(x) = x^2$ verifies our calculations

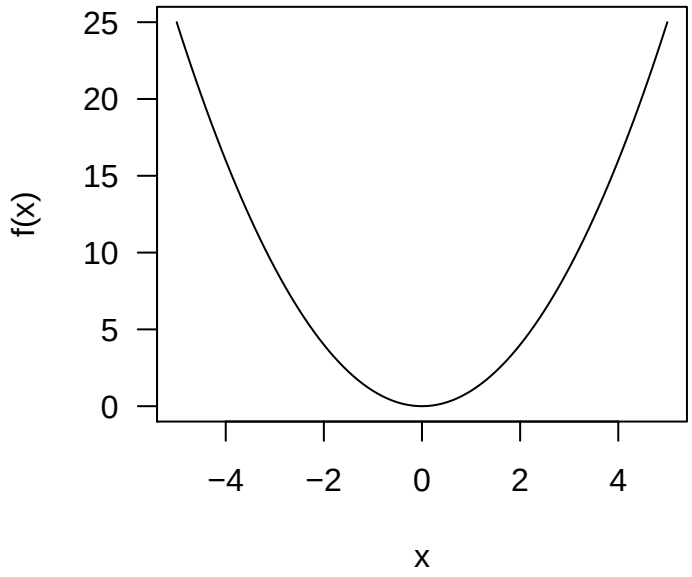


Figure 4.3: Example function $f(x) = x^2$

4.3 Rate of change at a point

- We often need to know the rate of change of a function at a point, and not simply an average rate of change across an interval.
- Figure 4.4, similar to Figure 4.2, shows, instead of two points a and b, point a and a second point defined in terms of its distance from the first

point a. Thus, the two points are now a and $a + h$ and the distance between the two points is equal to h.

- Now we can write that:

$$\frac{change\ in\ y}{change\ in\ x} = \frac{f(a+h) - f(a)}{a + h - a} = \frac{f(a+h) - f(a)}{h}$$

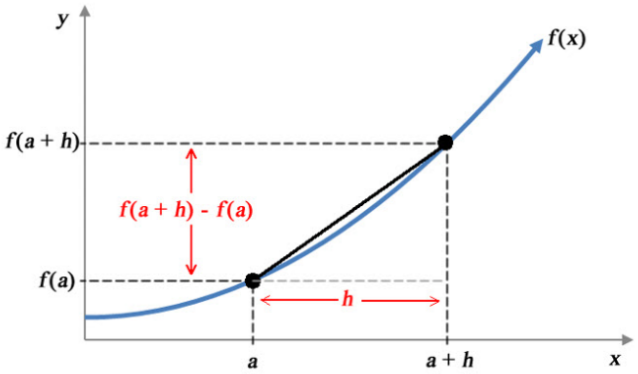


Figure 4.4: The average rate of change of $f(x)$ with respect to x over $[a, b]$ is equal to the slope of the secant line (in black)

Further:

- if we assume that the second point $a+h$ is really close to a, meaning that h approaches 0, denoted as $h \to 0$, we can find the rate of change at the point a
- the distance between the two points a and $a + h$ is getting smaller and so is the difference of the function values $f(a + h) - f(a)$. We denote these small differences as δx and δy, pronounced "delta x" and "delta y", respectively.
- the term δ reads as "delta" and represents a small change

We can thus continue and write that a **rate of change of a function at a point** is given by

$$\frac{small\ change\ in\ y}{small\ change\ in\ x} = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h} \tag{4.2}$$

E.g. let's look at the linear function $f(x) = 2x + 3$. We can find the rate of change at any point of x by:

$$\frac{small\ change\ in\ y}{small\ change\ in\ x} = \frac{f(x+h) - f(x)}{x + h - x} = \lim_{h \to 0} \frac{2(x+h) + 3 - (2x+3)}{x + h - x} = \lim_{h \to 0} \frac{2h}{h} = 2$$

It means that the function value $f(x)$ increases by 2 for every small increase, h, in x. Here, this increase is the same for all the values of x, i.e. it does not

depend on x. **The change in function value** $f(x)$ **can depend** on the value of x, for instance if we look at the quadratic $f(x) = x^2$ function, we get:

$$\frac{\textit{small change in } y}{\textit{small change in } x} = \frac{f(x+h) - f(x)}{x + h - x} = \lim_{h \to 0} \frac{x^2 + 2xh + h^2 - x^2}{h} = \lim_{h \to 0} \frac{2xh + h^2}{h} = 2x + h$$

This means that:

- the rate of change for the function $f(x)$ at a point x is $2x$
- the $f(x)$ value increases by $2x$ for every small increase, h, in x
- the rate of change along a quadratic function is changing constantly according to the value of x we are looking at, it is a function of x
- and finally that the rate of change does not give us any information about the rate of change globally.

4.4 Terminology and notation

- **differentiation** is the process of finding the rate of change of a given function
- the function is said to be **differentiated**
- the rate of change of a function is also known as the **derivative** of the function
- given a function $f(x)$ we say that we differentiate function in respect to x and write:

$$\lim_{h \to 0} \frac{\delta y}{\delta x} = \frac{dy}{dx}$$

or use the "prime"

$$f'(x)$$

4.5 Table of derivatives

- in practice, there is no need to compute $\lim_{h \to 0} \frac{\delta y}{\delta x}$ every time when we want to find a derivative of a function
- instead, we can use patterns of the common functions and their derivatives

Table 4.1: Common functions and their derivatives, k denotes a constant

Function $f(x)$	Derivative $f'(x)$
k	0
x	1
kx	k
x^n	nx^{n-1}

Function $f(x)$	Derivative $f'(x)$
kx^n	knx^{n-1}
e^x	e^x
e^{kx}	ke^{kx}
$\ln(x)$	$\frac{1}{x}$
$\ln(kx)$	$\frac{1}{x}$

We can use the Table 4.1 to find derivatives of some of the functions e.g.

- $f(x) = 3x$, $f'(x) = 3$
- $f(x) = 2x^4$, $f'(x) = 2 * 4x^{4-1} = 8x^3$
- $f(x) = e^{2x}$, $f'(x) = 2e^{2x}$
- $f(x) = \ln(4x)$, $f'(x) = \frac{1}{x}$

4.6 Exercises (differentiation)

Exercise 4.1. # Find derivatives of the functions

a) $f(x) = 2$
b) $f(x) = 2x + 1$
c) $f(x) = 5x^2$
d) $f(x) = 4x^3 + x^2$
e) $f(x) = \sqrt(x)$
f) $f(x) = \ln(2x)$
g) $f(x) = e^x$
h) $f(x) = \frac{9}{x^2} + ln(4x)$
i) $f(x) = 4x - 6x^6$
j) $f(x) = \frac{3}{x^2}$

Answers to selected exercises (differentiation)

Exr. 4.1

a) $f(x) = 2$, $f'(x) = 0$
b) $f(x) = 2x + 1$, $f'(x) = 2$
c) $f(x) = 5x^2$, $f'(x) = 10x$
d) $f(x) = 4x^3 + x^2$, $f'(x) = 12x^2 + 2x$
e) $f(x) = \sqrt(x) = x^{\frac{1}{2}}$, $f'(x) = \frac{1}{2}x^{\frac{1}{2}-1} = \frac{1}{2}x^{-\frac{1}{2}}$
f) $f(x) = \ln(2x)$, $f'(x) = \frac{1}{x}$
g) $f(x) = e^x$, $f'(x) = e^x$

Chapter 5

Integration

Aims

- to introduce the concept of integration

Learning outcomes

- to be able to explain what integration is
- to be able to explain the relationship between differentiation and integration
- to be able to integrate simple functions
- to to able to use integration to calculate the area under the curve in simple cases

5.1 Reverse to differentiation

- when a function $f(x)$ is known we can differentiate it to obtain the derivative $f'(x)$
- the reverse process is to obtain $f(x)$ from the derivative
- this process is called **integration**
- apart from simple reversing differentiation integration comes very useful in finding **areas under curves**, i.e. the area above the x-axis and below the graph of $f(x)$, assuming that $f(x)$ is positive
- the symbol for integration is $\int$ and is known as "integral sign"

E.g. let's take a function $f(x) = x^2$. Suppose we only have a derivative, which is $f'(x) = 2x$ and we would like to find the function given this derivative. Formally we write:

$$\int 2x dx = x^2 + c$$

where:

- the term $2x$ within the integral is called the **integrand**
- the term dx indicates the name of the variable involved, here x
- c is **constant of integration**

5.2 What is constant of integration?

- Integration reverses the process of differentiation, here, given our example function $f(x) = x^2$ that we pretended we do not know, we started with the derivative $f'(x) = 2x$ and via integration we obtained back the very function
$$\int 2x dx = x^2$$
- However, many function can result in the very same derivative since the derivative of a constant is 0 e.g. a derivatives of $f(x) = x^2$, $f(x) = x^2 + 10$ and $f(x) = x^2 + \frac{1}{2}$ all equal to $f'(x) = 2x$
- We have to take this into account when we are integrating, i.e. reverting differentiation. As we have no way of knowing what the original function constant is, we add it in form of c, i.e. unknown constant, called the constant of integration.

5.3 Table of integrals

Similar to differentiation, in practice we can use tables of integrals to be able to find integrals in simple cases

Table 5.1: Common functions and their integrals, k denotes a constant

Function $f(x)$	Integral $\int f(x)dx$
$constant\ k$	$kx + c$
x	$\frac{x^2}{2} + c$
kx	$k\frac{x^2}{2} + c$
x^n	$\frac{x^{n+1}}{n+1} + c\ \ if\ n \neq -1$
kx^n	$k\frac{x^{n+1}}{n+1} + c$
e^x	$e^x + c$
e^{kx}	$\frac{e^{kx}}{k} + c$
$\frac{1}{x}$	$\ln(x) + c$

E.g.

- $\int 4x^3 dx = \frac{4x^{3+1}}{3+1} = x^4 + c$
- $\int (x^2 + x)dx = \frac{x^3}{3} + \frac{x^2}{2} + c$ (note: we can evaluate integrals separately and add them as integration as differentiation is linear)

5.4 Definite integrals

- the above examples of integrals are **indefinite integrals**, the result of finding an indefinite integral is usually a function plus a constant of integration
- we have also **definite integrals**, so called because the result is a definite answer, usually a number, with no constant of integration
- definite integrals are often used to areas bounded by curves or, as we will cover later on, estimating probabilities
- we write:

$$\int_a^b f(x)dx$$

where:

- $\int_a^b f(x)dx$ is called the definite integral of $f(x)$ from a to b
- the numbers a and b are known as lower and upper limits of the integral

E.g. let's look at the function $f(x) = x$ plotted below and calculate a definite integral from 0 to 2.

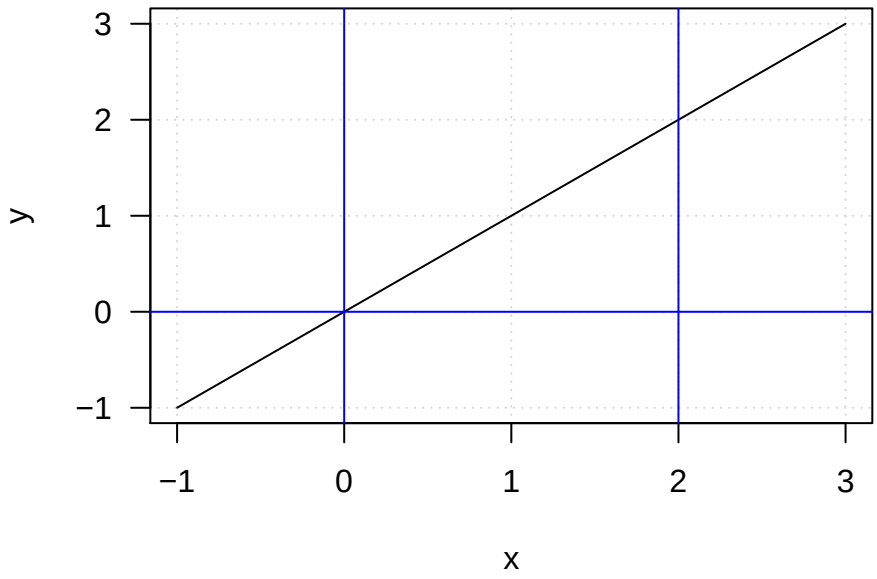


Figure 5.1: Graph of function $f(x) = x$

We write

$$\int_0^2 f(x)dx = \int_0^2 xdx = \Big[\frac{1}{2}x^2\Big]_0^2 = \frac{1}{2}(2)^2 - \frac{1}{2}(0)^2 = 2$$

so first find the integral and then we evaluate it at upper limit and subtracting the evaluation at the lower limit. Here, the result it 2. What would be the result if you tried to calculate the triangle area on the above plot, area defined by the blue vertical lines drawn at 0 and 2 and horizontal x-axis? The formula for the triangle area is $Area = \frac{1}{2} \cdot base \cdot height$ so here $Area = \frac{1}{2} \cdot 2 \cdot 2 = 2$ the same result as achieved with integration.

5.5 Exercises (integration)

Exercise 5.1. Integrate:

a) $\int 2 \cdot dx$
b) $\int 2x \cdot dx$
c) $\int (x^4 + x^2 + 1) \cdot dx$
d) $\int e^x \cdot dx$
e) $\int e^{2x} \cdot dx$
f) $\int \frac{2}{x} \cdot dx$
g) $\int_2^4 2x \cdot dx$
h) $\int_0^4 (x^2 + 1) dx$
i) $\int (x^4 + \frac{2}{x} + e^{2x}) dx$
j) $\int_0^4 (x^4 + 1) dx$

Answers to selected exercises (integration)

Exr. 5.1

a) $\int 2 \cdot dx = 2x + c$
b) $\int 2x \cdot dx = \frac{2x^2}{2} = x^2 + c$
c) $\int (x^4 + x^2 + 1) \cdot dx = \frac{x^5}{5} + \frac{x^3}{3} + x + c$
d) $\int e^x \cdot dx = e^x + c$
e) $\int e^{2x} \cdot dx = \frac{1}{2} e^{2x}$
f) $\int \frac{2}{x} \cdot dx = \int 2 \cdot \frac{1}{x} \cdot dx = 2 \ln x + c$
g) $\int_2^4 2x \cdot dx = \Big[x^2\Big]_2^4 = 16 - 4 = 12$
h) $\int_0^4 (x^2 + 1) dx = \Big[\frac{x^3}{3} + x\Big]_0^4 = \frac{4^3}{3} + 4 - 0 = \frac{64}{3} + 4 = \frac{76}{3}$

Chapter 6

Vectors

Aims

- to introduce vectors and basic vectors operations

Learning outcomes

- to be able to write n-dimensional vectors using vector notations
- to be able to perform addition and scalar multiplication
- to be able to check if two vectors are orthogonal

A large number of statistical models use vectors and matrices, both for compact representations, and for the calculations, e.g. parameter estimates.

6.1 Vectors

- A vector is an ordered set of number
- These numbers, e.g. in vector $\mathbf{x}$ can be expressed as a row $\mathbf{x} = [6 \quad 0 \quad 5 \ldots 1]$
- or as a column $\mathbf{x} = \begin{bmatrix} 6 \\ 0 \\ 5 \\ \vdots \\ 1 \end{bmatrix}$
- the number of elements in a vector is referred to as its **dimension** and we often use n to express n-dimensional vector, where n can be any natural number
- here, we denote vectors using small bold font $\mathbf{x}$, other notations may include an arrow $\vec{x}$ or overline $\overline{x}$
- also **parentheses** are used interchangeably with **square bracket**,

e.g. $\mathbf{x} = [6 \quad 0 \quad 5 \ldots 1]$ can be written as $\mathbf{x} = (6 \quad 0 \quad 5 \ldots 1)$ or $\begin{pmatrix} 6 \\ 0 \\ 5 \\ \vdots \\ 1 \end{pmatrix}$

6.2 Operations on vectors

Given two vectors of the same dimension: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$

Addition: we add two vectors, element by element $\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \\ \vdots \\ x_n + y_n \end{bmatrix}$

Scalar multiplication: we can multiple vector by a numerical value, scalar, denoted as γ:

$$\gamma \cdot \mathbf{x} = \begin{bmatrix} \gamma \cdot x_1 \\ \gamma \cdot x_2 \\ \gamma \cdot x_3 \\ \vdots \\ \gamma \cdot x_n \end{bmatrix}$$

Difference $\mathbf{x} - \mathbf{y}$ can be written as $\mathbf{x} + (-1) \cdot \mathbf{y}$, thus we multiply second vector with -1 and then add two vectors

Linear combination of vectors: the vector $\gamma \cdot \mathbf{x} + \delta \cdot \mathbf{y}$ is said to be a linear combination of $\mathbf{x}$ and $\mathbf{y}$:

$$\gamma \cdot \mathbf{x} + \delta \cdot \mathbf{y} = \begin{bmatrix} \gamma \cdot x_1 + \delta \cdot y_1 \\ \gamma \cdot x_2 + \delta \cdot y_2 \\ \gamma \cdot x_3 + \delta \cdot y_3 \\ \vdots \\ \gamma \cdot x_n + \delta \cdot y_n \end{bmatrix}$$

Inner product of vectors is given by:

$$\mathbf{x} \cdot \mathbf{y} = x_1 \cdot y_1 + x_2 \cdot y_2 + \ldots x_n \cdot y_n = \sum_{i=1}^{n} x_i \cdot y_i$$

Orthogonality of vectors: two vectors are said to be orthogonal if their inner

product is zero

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i \cdot y_i = 0$$

6.3 Null and unit vector

- a **null vector** is a vector whose elements are all 0; the difference between any vector and itself yields a null vector
- a **unit vector** is a vector whose elements are all 1

Exercise 6.1. Based on vector definitions and operations:

a) find the vector $\mathbf{x} + \mathbf{y}$ when $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}$

b) find the vector $2\mathbf{x} - \mathbf{y}$ when $\mathbf{x} = \begin{bmatrix} -2 \\ 3 \\ 5 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ -4 \\ 7 \end{bmatrix}$

c) are $\mathbf{u}$ and $\mathbf{v}$ vectors orthogonal when when $\mathbf{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$?

d) are $\mathbf{u}$ and $\mathbf{v}$ vectors orthogonal when when $\mathbf{u} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$?

e) find the value n such that the vectors $\mathbf{u} = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} n \\ 1 \\ 8 \end{bmatrix}$ are orthogonal.

Answers to selected exercises (vectors and matrices)

Exr. 6.1

a)

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+0 \\ 2+3 \\ 5+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 6 \end{bmatrix}$$

b)

$$2\mathbf{x} - \mathbf{y} = \begin{bmatrix} 2 \cdot (-2) \\ 2 \cdot 3 \\ 2 \cdot 5 \end{bmatrix} + \begin{bmatrix} (-1) \cdot 0 \\ (-1) \cdot (-4) \\ (-1) \cdot 7 \end{bmatrix} = \begin{bmatrix} -4 \\ 6 \\ 10 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \\ -7 \end{bmatrix} = \begin{bmatrix} -4+0 \\ 6+4 \\ 10-7 \end{bmatrix} = \begin{bmatrix} -4 \\ 10 \\ 3 \end{bmatrix}$$

c) Yes, to check orthogonality we need to calculate the inner product of two vectors and see if it is equal to 0, here $\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{2} u_i \cdot v_i = 1 \cdot 2 + 2 \cdot (-1) = 2 - 2 = 0$

d) No, since the inner product does not equal to 0

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{2} u_i \cdot v_i = 3 \cdot 7 + (-1) \cdot 5 = 21 - 5 = 16 \neq 0$$

Chapter 7

Matrices

Aims

- to introduce matrix and basic matrices operations

Learning outcomes

- to be able to write matrices using matrix notations
- to be able to perform simple matrix operations such as adding and multiplication
- to be able to find the reverse of the 2-dimensional matrix

7.1 Matrix

A matrix is a rectangular array of numbers e.g.

$$\mathbf{A} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & x_{1m3} & \dots & x_{mn} \end{bmatrix}$$

where:

- the notional subscripts in the typical element x_{ij} refers to its row and column location in the array, e.g. x_{12} refers to element in the first row and second column
- we say that matrix has m rows and n columns and the **dimension** of a matrix is defined as $m \times n$
- a matrix can be viewed as a set of column vectors or a set of row vectors
- a vector can be viewed as a matrix with only one column or with only one row

7.2 Special matrices

- A matrix with the same number of rows as columns, $m = n$, is said to be a **square matrix**
- A matrix that is not squared, $m \neq n$ is called **rectangular matrix**
- A **null matrix** is composed of all 0
- An **identity matrix**, denoted as $\mathbf{I}$ or $\mathbf{I_n}$, is a square matrix with 1's on the main diagonal and all other elements equal to 0, e.g. a three-dimensional identity matrix is

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- A square matrix is said to be **symmetric** if $x_{ij} = x_{ji}$ e.g.

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 2 \\ 4 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

- A **diagonal matrix** is a square matrix whose non-diagonal entries are all zero, that is $x_{ij} = 0$ for $i \neq j$, e.g.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- An **upper-triangular matrix** is a square matrix in which all entries below the diagonal are 0, that is $x_{ij} = 0$ for $i < j$ e.g.

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 4 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix}$$

- A **lower-triangular matrix** is a square matrix in which all entries above the digonal are 0, that is hat is $x_{ij} = 0$ for $i > j$ e.g.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

7.3 Matrix operations

- matrix $\mathbf{A} = \mathbf{B}$ if both matrices have exactly the same dimension and if each element of $\mathbf{A}$ equals to the corresponding element of e.g. $\mathbf{A} = \mathbf{B}$ if $\mathbf{A} = \begin{bmatrix} 1 & 3 & 4 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 3 & 4 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix}$

- for any matrix $\mathbf{A}$ the **transpose**, denoted by $\mathbf{A}^\top$ or $\mathbf{A}'$, is obtained by interchanging rows and columns, e.g. given matrix $\mathbf{A} = \begin{bmatrix} 1 & 3 & 4 \\ 0 & 2 & 5 \\ 0 & 0 & 3 \end{bmatrix}$ we have $\mathbf{A}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 2 & 0 \\ 4 & 5 & 3 \end{bmatrix}$. The transpose of a transpose of a matrix yield the original matrix, $\left(\mathbf{A}^\top\right)^\top = \mathbf{A}$
- we can **add** two matrices if they have the same dimension, e.g.

$$\mathbf{A} + \mathbf{B} = \mathbf{A} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} x_{11} + y_{11} & x_{12} + y_{12} \\ x_{21} + y_{21} & x_{22} + y_{22} \end{bmatrix}$$

- we can **multiply** a matrix by **a scalar** δ e.g.

$$\delta \cdot \mathbf{A} = \begin{bmatrix} \delta \cdot x_{11} & \delta \cdot x_{12} \\ \delta \cdot x_{21} & \delta \cdot x_{22} \end{bmatrix}$$

- we can **multiply two matrices** if they are **conformable**, i.e. first matrix has the same number of columns as the number of rows in the second matrix. We then can write:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \times \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ y_{31} & y_{32} \end{bmatrix} = \begin{bmatrix} x_{11} \cdot y_{11} + x_{12} \cdot y_{21} + x_{13} \cdot y_{31} & x_{11} \cdot y_{12} + x_{12} \cdot y_{22} + x_{13} \cdot y_{32} \\ x_{21} \cdot y_{11} + x_{22} \cdot y_{21} + x_{23} \cdot y_{31} & x_{21} \cdot y_{12} + x_{22} \cdot y_{22} + x_{23} \cdot y_{32} \end{bmatrix}$$

7.4 Inverse of a matrix

For a square matrix $\mathbf{A}$ there may exist a matrix $\mathbf{B}$ such that $\mathbf{A} \cdot \mathbf{B} = \mathbf{I}$. An **inverse**, if it exists, is denoted as $\mathbf{A}^{-1}$ and we can rewrite the definition as

$$\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{I}$$

where $\mathbf{I}$ is an identify matrix (equivalent to 1). There is no division for matrices, instead we can use inverse to multiply the matrix by an inverse, similar to when instead of dividing the number a by b we multiply a by reciprocal of $b = \frac{1}{b}$

For a 2-dimensional matrix we can follow the below formula for obtaining the inverse

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}^{-1} = \frac{1}{x_{11} \cdot x_{22} - x_{12} \cdot x_{21}} \cdot \begin{bmatrix} x_{22} & -x_{12} \\ -x_{21} & x_{11} \end{bmatrix}$$

7.5 Orthogonal matrix

- A matrix $\mathbf{A}$ for which $\mathbf{A}^\top = \mathbf{A^{-1}}$ is true is said to be **orthogonal**

Exercise 7.1. Given matrices

$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

a) what is the dimension of matrix $\mathbf{A}$?
b) what is $\mathbf{A}^\top$?
c) which of the matrices is i) an identity matrix ii) a square matrix, iii) null matrix, iv) diagonal matrix, v) a triangular matrix,?
d) calculate $\mathbf{A} + \mathbf{B}$?
e) calculate $\mathbf{A} \cdot \mathbf{C}$?
f) calculate $\mathbf{B}^\top$
g) calculate $\mathbf{A}^{-1}$
h) calculate $(\mathbf{A} + \mathbf{B})^{-1}$

Answers to selected exercises (matrices)

Exr. 7.1

a) 2×2

b) $\mathbf{A}^\top = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

c) i) identity matrix: $\mathbf{B}$, ii) a square matrix: $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, iii) null matrix: none, iv) diagonal matrix: $\mathbf{B}$ (identity matrix is diagonal) and $\mathbf{C}$, v) triangular $\mathbf{B}$ and $\mathbf{C}$ as both identify matrix $\mathbf{B}$ and diagonal matrix $\mathbf{C}$ is triangular, both lower and upper triangular

d) $\mathbf{A} + \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 5 \end{bmatrix}$

e) $\mathbf{A} \cdot \mathbf{C} = \begin{bmatrix} 1 \cdot 1 + 2 \cdot 0 & 1 \cdot 0 + 2 \cdot 2 \\ 3 \cdot 1 + 4 \cdot 0 & 3 \cdot 0 + 4 \cdot 2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 3 & 8 \end{bmatrix}$

f) $\mathbf{B}^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

g)

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \frac{1}{1 \cdot 4 - 2 \cdot 3} \cdot \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = -\frac{1}{2} \cdot \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

Part II

Linear Models

Chapter 8

Introduction to linear models

Aims

- to introduce concept of linear models using simple linear regression

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

8.1 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)

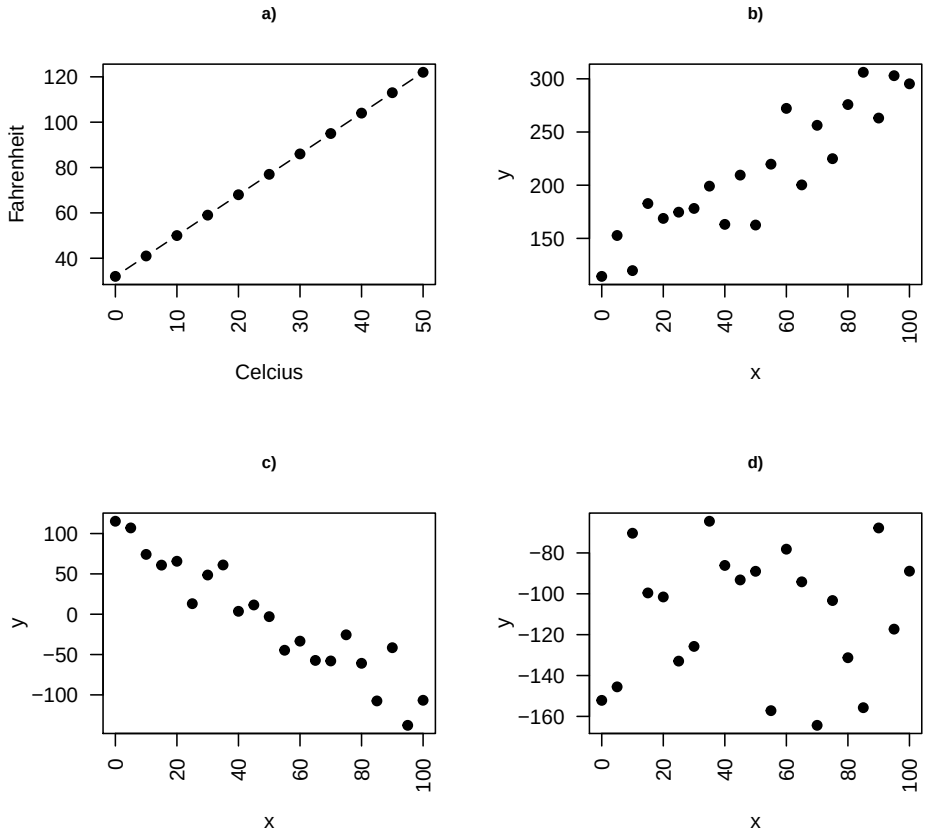


Figure 8.1: Deterministic vs. statistical relationship: a) deterministic: equation exactly describes the relationship between the two variables e.g. Ferenheit and Celcius relationship ; b) statistical relationship between x and y is not perfect (increasing relationship), c) statistical relationship between x and y is not perfect (decreasing relationship), d) random signal

8.2 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 more variables, e.g. the calories burnt could be a function of both the kilometers cycled and status of bike, or the transformation of the variables

More examples where model parameters appear linearly:

- $Y_i = \alpha + \beta x_i + \gamma x_i + \epsilon_i$
- $Y_i = \alpha + \beta x_i^2 \epsilon$
- $Y_i = \alpha + \beta x_i^2 + \gamma x_i^3 + \epsilon$

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

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

8.3 Terminology

There are many terms and notations used interchangeably:

- y is being called:
  - response
  - outcome
  - dependent variable
- x is being called:
  - exposure
  - explanatory variable
  - dependent variable
  - predictor
  - covariate

8.4 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?

8.5 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

- For example, 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:

```
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)
```

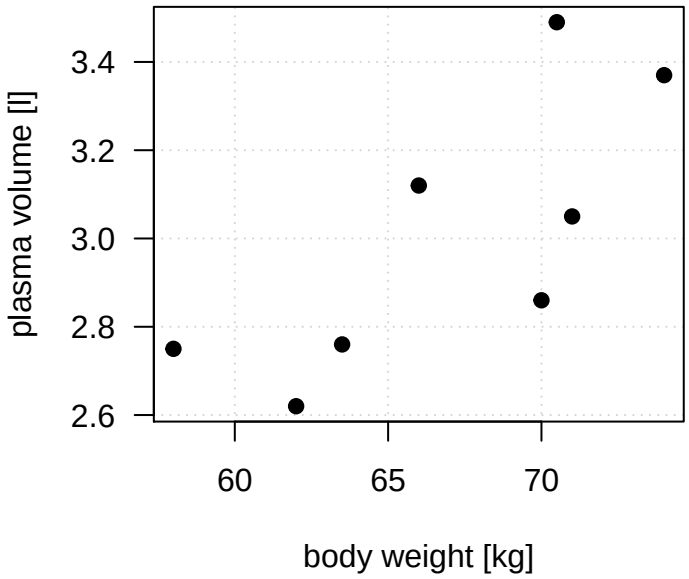


Figure 8.2: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*.

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:

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

- 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 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:

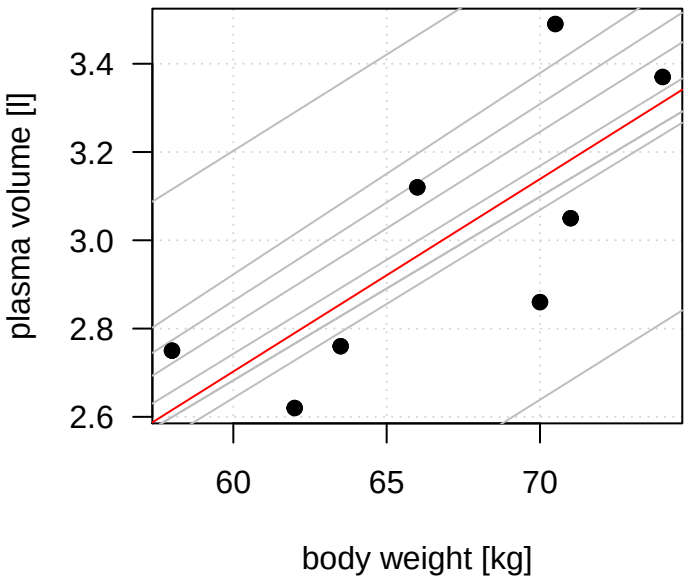


Figure 8.3: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regression gives the equation of the straight line (red) that best describes how the outcome changes (increase or decreases) with a change of exposure variable

$$Y_i = \alpha + \beta \cdot x_i + \epsilon_i \tag{8.1}$$

where:

- we call α and β **model coefficients**
- and ϵ_i **error terms**

8.6 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 $\hat{\beta}$

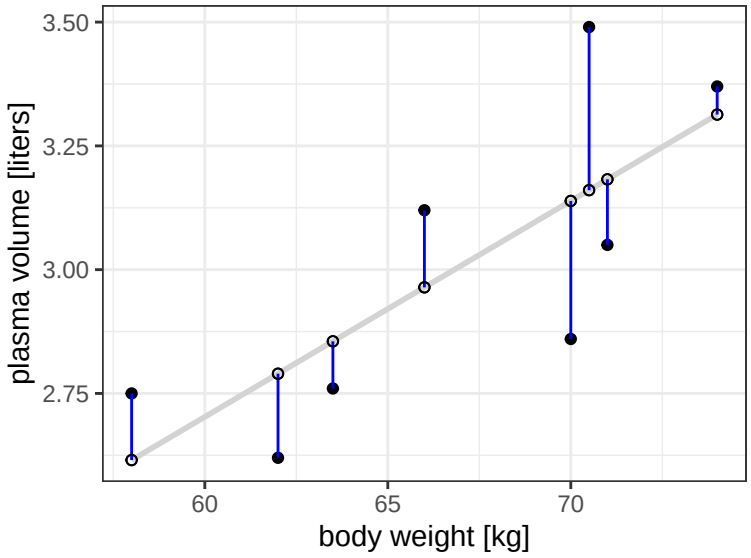


Figure 8.4: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regrssion gives the equation of the straight line (red) that best describes how the outcome changes with a change of exposure variable. Blue lines represent error terms, the vertical distances to the regression line

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 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 we get Theorem 8.1

Theorem 8.1 (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 (Least squares)

Let's try least squares method to find coefficient estimates in our body weight and plasma volume example

```
# 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

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
```

8.7 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**

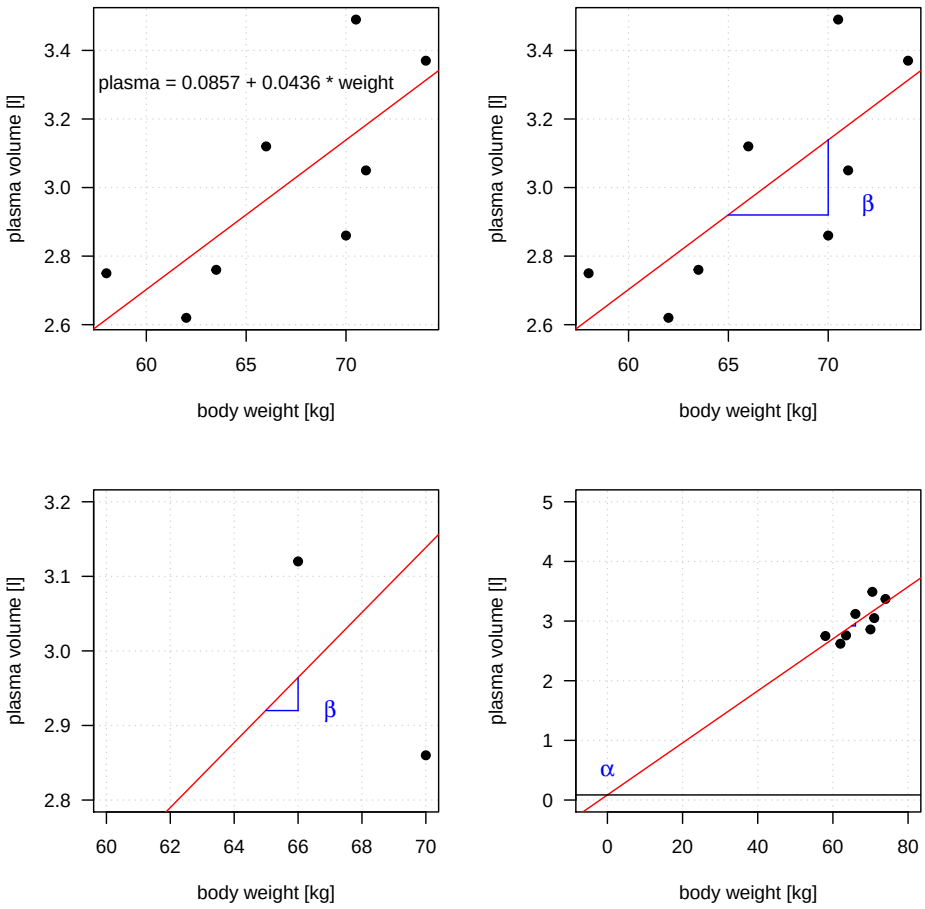


Figure 8.5: Scatter plot of the data shows that high plasma volume tends to be associated with high weight and *vice verca*. Linear regression gives the equation of the straight line that best describes how the outcome changes (increase or decreases) with a change of exposure variable (in red)

8.8 Hypothesis testing

- 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 measure 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 building confidence and prediction intervals

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

Under the null hypothesis:

- $H_0 : \beta = 0$ we have: $\frac{\hat{\beta} - \beta}{e.s.e(\hat{\beta})} \sim t(n-p)$, where
- n is number of observations
- p is number of model parameters
- $\frac{\hat{\beta} - \beta}{e.s.e(\hat{\beta})}$ is called the t-statistics
- that follows Student's t distribution with $n - p$ degrees of freedom

Example (Hypothesis testing)

Let's look again at our example data. This time we will not only fit the linear regression model but 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.
- If we were to test if there is an association between weight and plasma volume we would write under $H_0 : \beta = 0$ and $\frac{\hat{\beta}-\beta}{e.s.e(\hat{\beta})} = \frac{0.04362-0}{0.01527} = 2.856582$
- and we would compare t-statistics to Student's t distribution with $n-p = 8-2 = 6$ degrees of freedom (we have two model parameters, α and β)
- we can use Student's t distribution table or R code to obtain the p-value

```
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 an significant association between weight and plasma volume

8.9 Vector-matrix notations

While in simple linear regression it is feasible to arrive at the parameters estimates using calculus in more realistic settings with multiple regression (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 notations.

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

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

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

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

$$\dots$$

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

- we can group all Y_i and ϵ_i into column vectors: $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ and $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$
- we stack two parameters β_0 and β_1 into another column vector:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

- we then append a vector of ones with the single predictor for each i and create a matrix with two columns: design matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

Now we can write our linear model in a vector-matrix notations as:

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

Definition: vector matrix form of the linear model

The vector-matrix representation of a linear model with $p-1$ predictors can be written as

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

where:

- $\mathbf{Y}$ is $n \times 1$ vector of observations
- β is $p \times 1$ vector of parameters
- $\mathbf{X}$ is $n \times p$ design matrix
- 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 8.2 (Least squares in vector-matrix notation)**.** *The least squares estimates for a linear regression of the form:*

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

is given by:

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

Example: vector-matrix notation

Following the above definition we can write our weight - plasma volume model as:

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

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

8.10 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

Earlier we said that we use estimated standard error in hypothesis testing and in finding the intervals but we have not yet said how to calculate e.s.e. Using vector-matrix notation we can now write that:

$$\frac{(\mathbf{b}\hat{\beta} - \mathbf{b}^T\beta)}{\sqrt{\frac{RSS}{n-p}\mathbf{b^T}(\mathbf{X^T X})^{-1}\mathbf{b}}}$$

where:

- the denominator would yield e.s.e(β_1) if $\mathbf{b^T} = (0 \quad 1)$ and a model $Y_i = \beta_0 + \beta_1 x + \epsilon_i$
- a confidence interval estimate for β_1 could be estimated via:

$$\mathbf{b^T}\hat{\beta} \pm t(n-p; \frac{1+c}{2})\sqrt{\frac{RSS}{n-p}(\mathbf{b^T}(\mathbf{X^T X})^{-1}\mathbf{b})}$$

- and a prediction interval with confidence c is

$$\mathbf{b^T}\hat{\beta} \pm t(n-p; \frac{1+c}{2})\sqrt{(\frac{RSS}{n-p}(1 + \mathbf{b^T}(\mathbf{X^T X})^{-1}\mathbf{b})}$$

We will not go further into these calculations here but use R functions to obtain these

- just remember that the prediction interval is always **wider** than the confidence interval
- note $(1 +)$ in the prediction interval equation

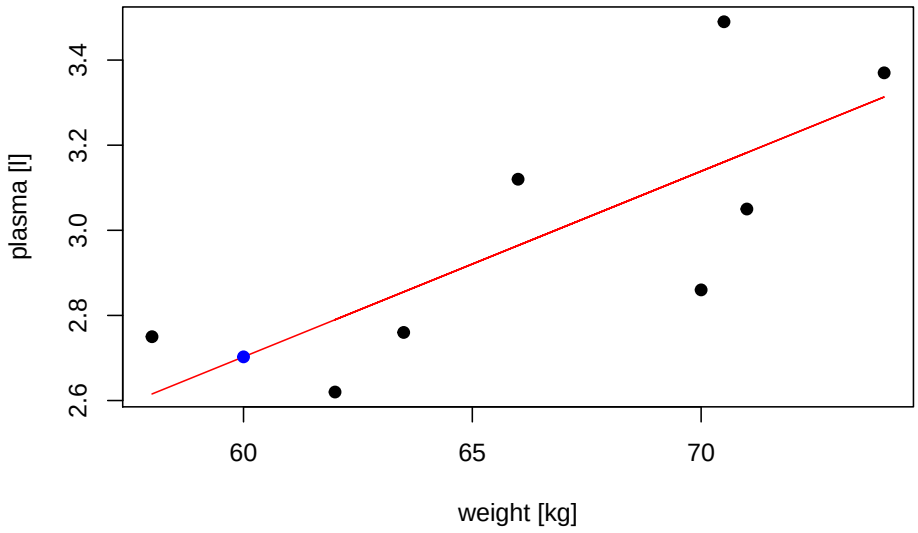
Example: prediction and intervals

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

```
# 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

# find confidence intervals for the model coefficients
confint(model)
##                     2.5 %     97.5 %
## (Intercept) -2.419908594 2.59135716
## weight       0.006255005 0.08097567

# 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
predict(model, newdata = new.obs,  interval = "prediction")
##        fit      lwr      upr
## 1 2.702645 2.079373 3.325916

# plot the original data, fitted regression and predicted value
plot(weight, plasma, pch=19, xlab="weight [kg]", ylab="plasma [l]")
lines(weight, model$fitted.values, col="red") # fitted model in red
points(new.obs, predict(model, newdata = new.obs), pch=19, col="blue") # predicted val
```



8.11 Exercises: linear models I

Data for exercises

- Link 1
- Alternative Link 2

Exercise 8.1. 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) $Y_i = \alpha + \gamma x_i^\beta + \epsilon_i$

Exercise 8.2. 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 statisitcs:

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

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

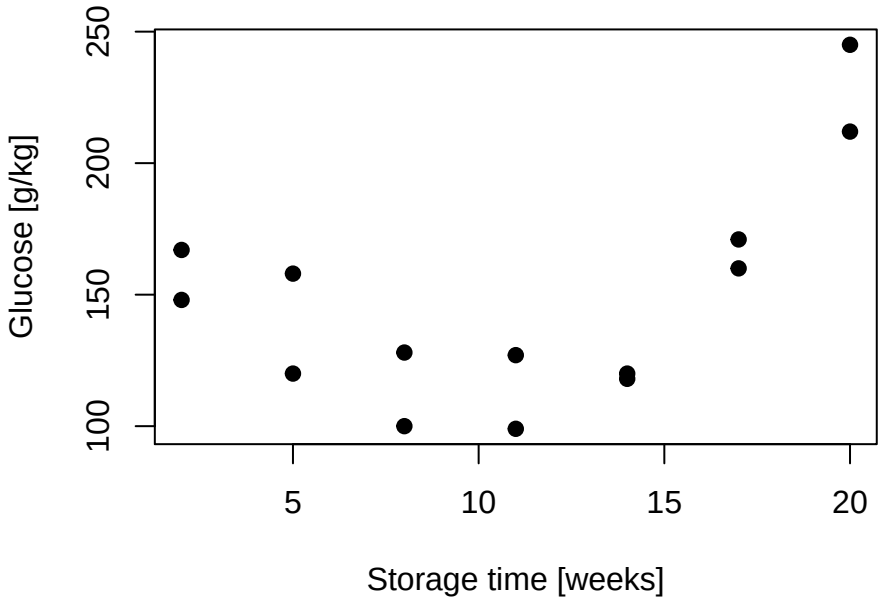


Figure 8.6: Sugar in potatoes: relationship between storage time and glucose content

Answers to selected exercises (linear models)

Exr. 8.2

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

```
# 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
```

```
# 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
print(beta.hat)
##             [,1]
## [1,] 0.20173770
## [2,] 0.02284426
```

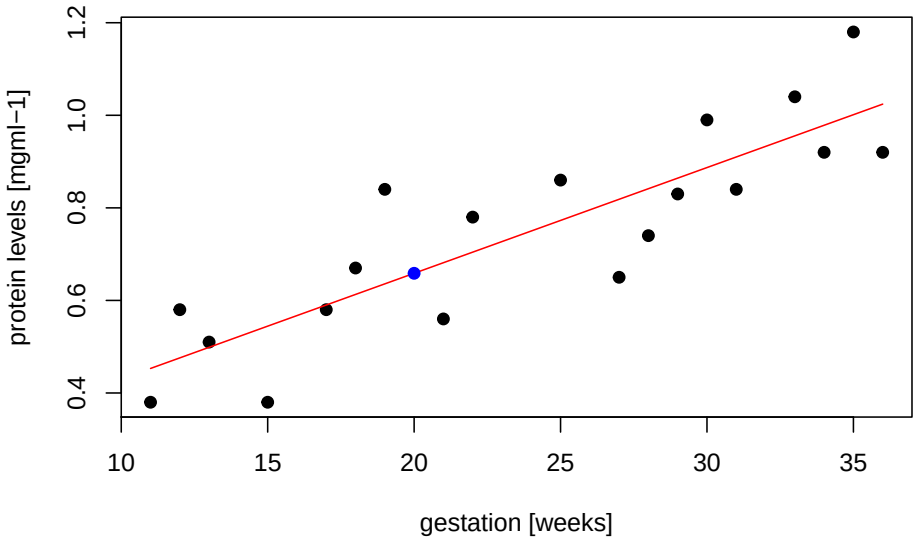
e) We use `lm()` function to check our calculations

```
# 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)
```



Exr. 8.3

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

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

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

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

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

```
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:

```
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

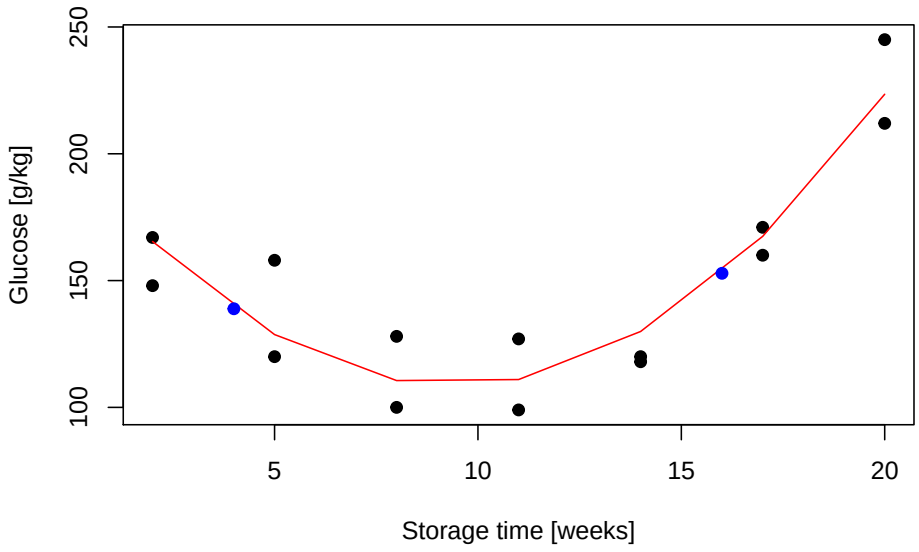
- $\frac{\hat{\gamma}-\gamma}{e.s.e(\hat{\gamma})} \sim t(n-p) = \frac{1.030423-0}{0.1406} = 7.328755$ where the value follows Student's t distribution with $n-p = 19-2 = 17$ degrees of freedom. We can now estimate the a p-value using Student's t distribution table or use a function in R

```
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 [g/kg
lines(data.potatoes$Weeks, model$fitted.values, col="red")
points(new.obs[,1], pred.y, pch=19, col="blue")
```



Chapter 9

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

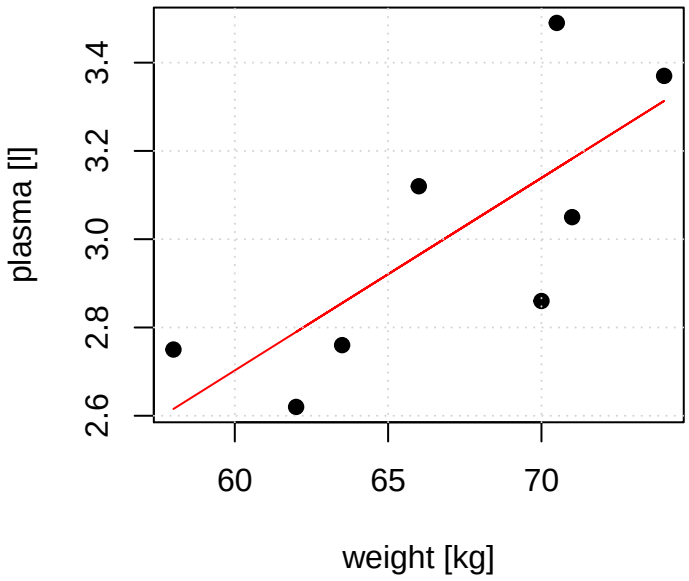
9.1 Interpreting and using linear regression models

- 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

9.2 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)
```

```r
# 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()
```




```r
# 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
```

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? _ Answer: it would increase by 0.04 liter
- And if we go up in weight 10 kg what would be our expected change in plasma volume?
- Answer: it would increase by $0.04 \cdot 10 = 0.4$ liter

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

9.3 Example: Galapagos Islands

Researchers were interested in biological diversity on the Galapagos islands. They've 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:

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

head(gala)
##              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
```

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.

```
# 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
```

Model

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

Slope

- Compare two islands, where the second island has an elevation one meter higher than the first one, what can we say about the number of species according to the model?
- Answer: the second island will have 0.32 species more than the first one
- How about if the difference is 100 m?
- Answer: now the second island will have $0.32 \cdot 100 = 32$ more species

Not so easy

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

```r
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
```

Model

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

Slope

- Compare two islands, where the second island has an elevation one meter higher than the first one, what can we say about the number of species according to the model?
- Answer: the second island will have 0.20 species more than the first one
- How about if the difference is 100 m?
- Answer: now the second island will have $0.20 \cdot 100 = 20$ more species

Specific interpretation

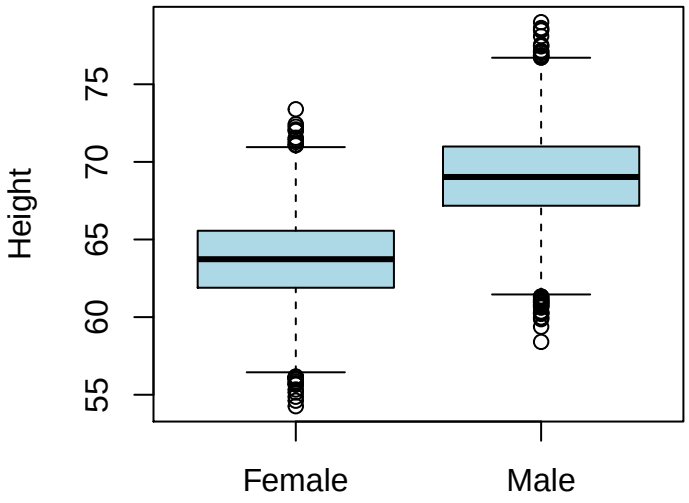
- obviously there is difference between 32 and 20 species given the same elevation difference and using the two different models
- our interpretations need to be more specific and we say
- **a unit increase in x with other predictors held constant will produce a change equal to $\hat{\beta}$ in the response y**
- it is of course often quite unrealistic to be able to control other variables and keep them constant and for our simple regression, model 2, 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 thought the two models are showing a strong association between elevation and number of species

- we will learn later how to assess models and select variables (feature selection), here, we continue focusing on learning how to interpret the coefficients given a model

9.4 Example: Height and gender

- Data are available containing the weight [lbs] and height [inches] of 10000 men and women
- We want to compare the average height of men and women
- We can do that using linear regression and including gender as **binary variable**

```
# 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")
```



Model

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

where

$$I_{x_i} = \begin{cases} 1 & \text{if } x_i = 1 \\ 0 & \text{if } x_i = 0 \end{cases} \tag{9.1}$$

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

```
# 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 choose one of the category as baseline, here Females
- model summary prints the output of the model with the baseline category "hidden"
- notice the only label we have is "GenderMale"
- meaning that we ended-up having a model coded as below:

$$I_{x_i} = \begin{cases} 1 & \text{if } \quad person_i \text{ is } male \\ 0 & \text{if } \quad person_i \text{ is } female \end{cases} \tag{9.2}$$

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

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

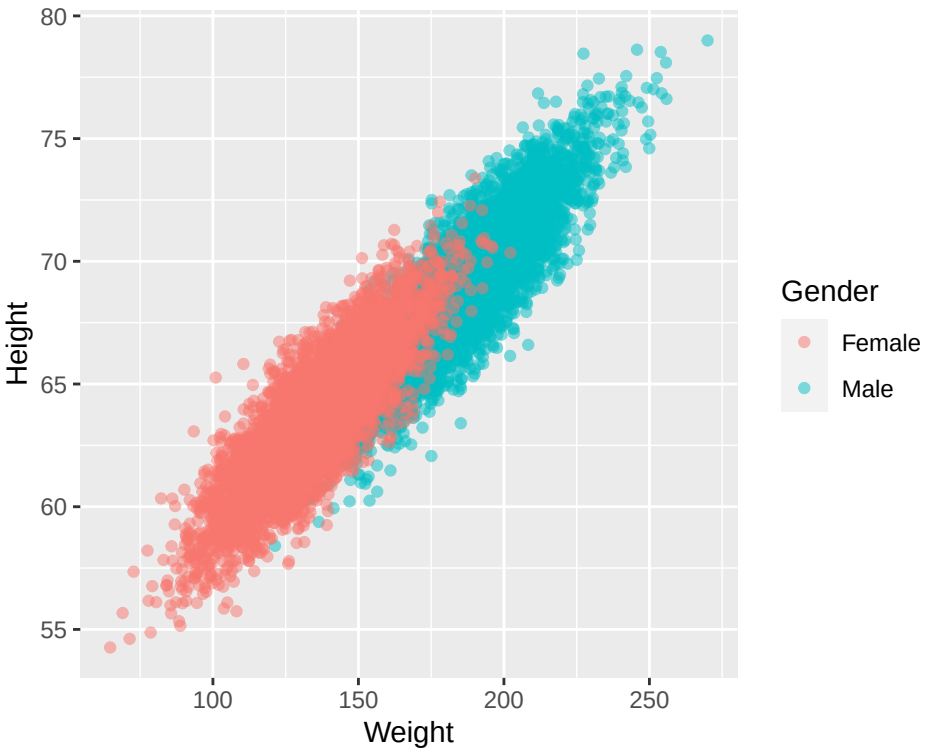
- and if observation i is female then the expected value of height is:

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

9.5 Example: Heigth, weight and gender I

- So there is a difference in height between the gender, as expected
- Is there a relationship between weight and height?
- If so, does this relationship depend on gender?

```
library(ggplot2)

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



- From the plot we can see that height increases with weight
- Males have higher heights than females
- Males have higher weights than females
- The relationship between height and weight appears to be the same for males and females, i.e. height increass with weight for both men and women

Model

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

where

$$I_{x_i} = \begin{cases} 1 & \text{if} \quad person_i \; is \; male \\ 0 & \text{if} \quad person_i \; is \; female \end{cases} \tag{9.3}$$

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

```
# 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
```

Estimates

$$\hat{\alpha} = 47.031$$
$$\hat{\beta} = -0.963$$
$$\hat{\gamma} = 0.123$$

- therefore, for a male of weight 161.4 we would predict a height of:

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

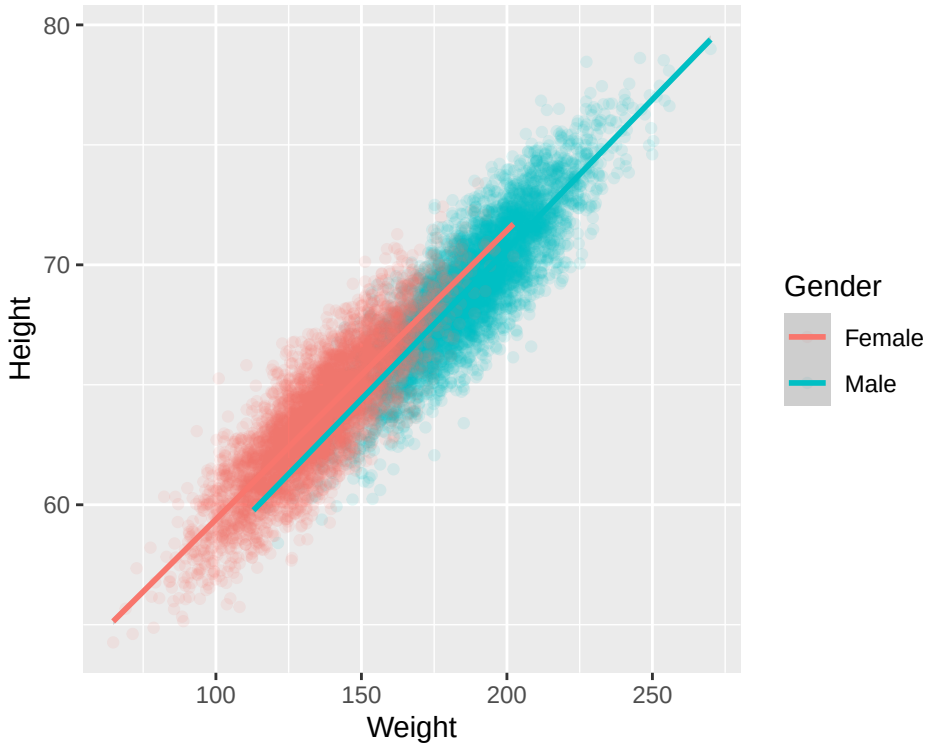
- and for a female of weight 161.4 we would predict a height of

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

- as much as our calculations are correct, the above example also shows the need of considering the data when interpreting coefficients. Our expected female height at weight 161.4 is 66.9, larger than the men height, 65.9. However, our data show that the the majority of our values for female lie between 100 and 175 and between 150 and 250 for males, i.e. in different ranges

- we should **not predict outside the data range**
- finally, we can plot our data and the fitted model

```
# plot the data separately for Male and Female
# using ggplot() and geom_smooth()
ggplot(data=htwtgen, aes(x = Weight, y=Height, col = Gender)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method=lm)
```



9.6 Example: Heigth, weight and gender II

- The fitted lines in the above example are parallel, the slope is modeled to be the same for male and females, 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
- And we then talk about including **interaction effect**

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 males in our example (keeping the same coding as above)
- $i = 2$ corresponds to females

```
# 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 male 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 female of weight x, a height of

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

Estimates

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

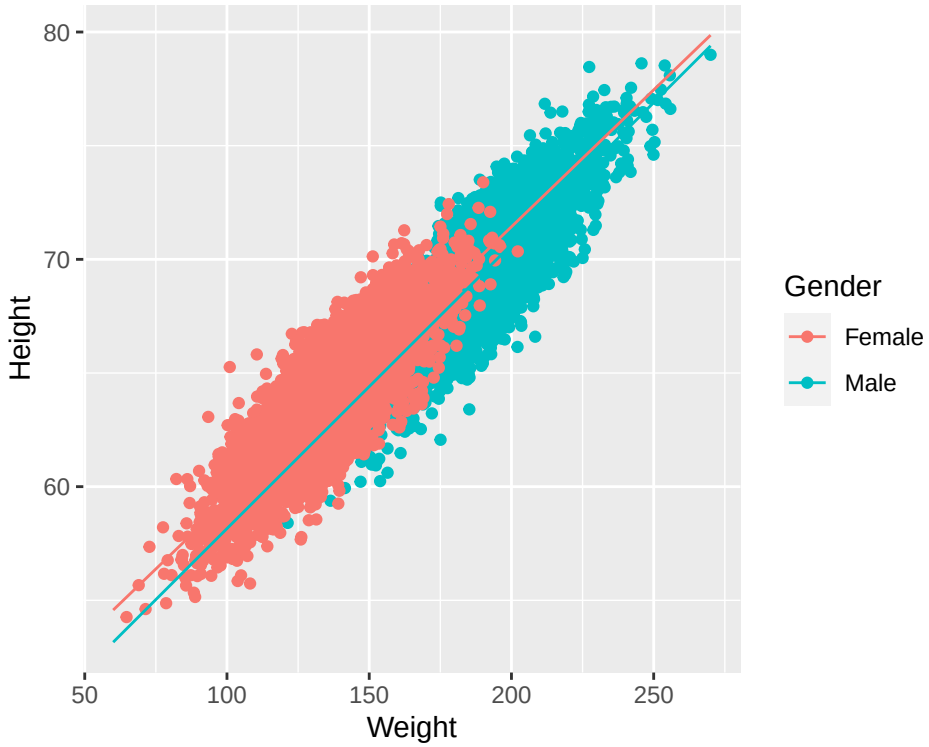
$$\hat{\beta_1} = 0.125$$

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

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

- we can see from the regression output that the interaction term, "Gender-Male:Weight, is significant
- and therefore the relationship between weight and height is different in males and females
- we can plot the fitted model now and see that the lines are no longer parallel
- we will see clearer example of interactions in exercises

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

ggPredict(model3)
```



9.7 Exercises: linear models II

Data for exercises

- Link 1
- Alternative Link 2

Exercise 9.1. Given the "height-weight-gender" data:

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 males and females give weight of 120 lbs?
c) can you use predict() function to check your calculations?

Exercise 9.2. 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 kilo-calorie 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

```
# 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 9.3. A clinical trial

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 chaining differently under different treatment? Hint: here we have three categories which can be seen as expalnding the models with tow categories by an additional one: one category will be treated as baseline

```
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
```

Answers to selected exercises (linear models II)

Exr. 9.1

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:

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

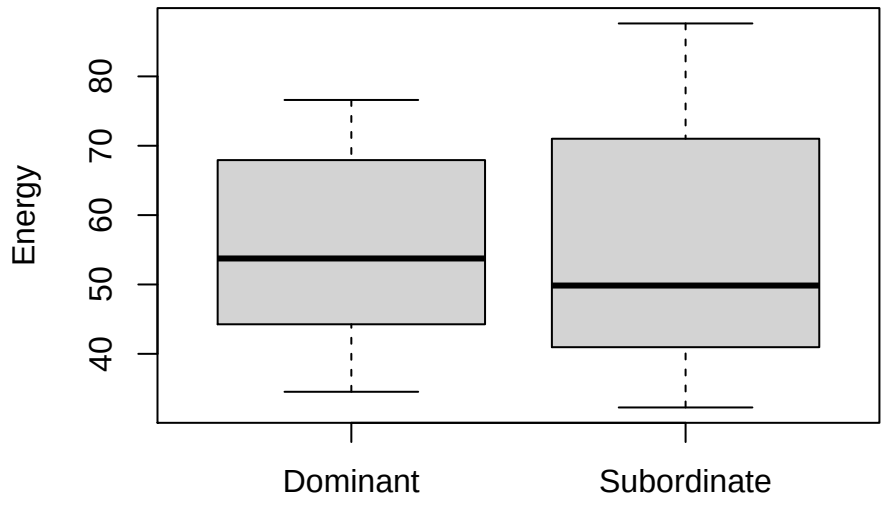
$$E(height|female\ and\ weight = x) = 47.34778 + 0.12043x$$

c)

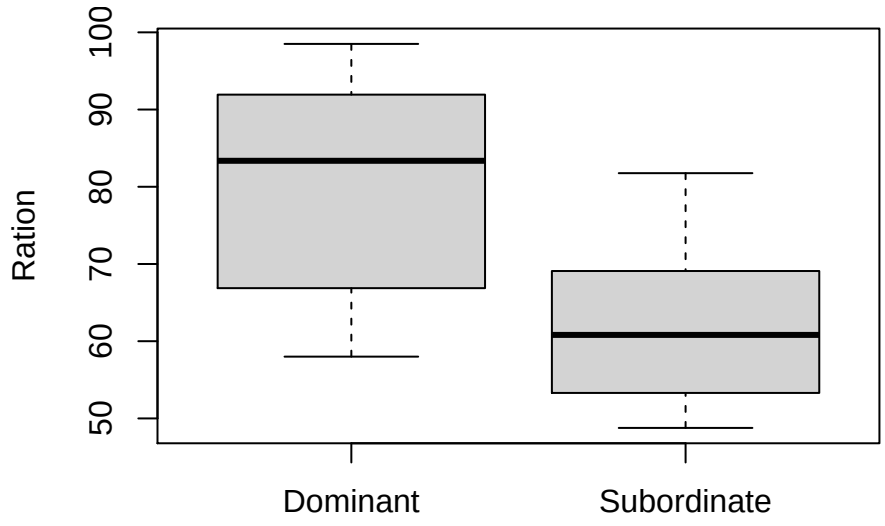
```
# for men
new.obs <- data.frame(Weight=120, Gender="Male")
predict(model3, newdata = new.obs)
##        1
## 60.65427

# for female
new.obs <- data.frame(Weight=120, Gender="Female")
predict(model3, newdata = new.obs)
##        1
## 61.79882
```

Exr. 9.2

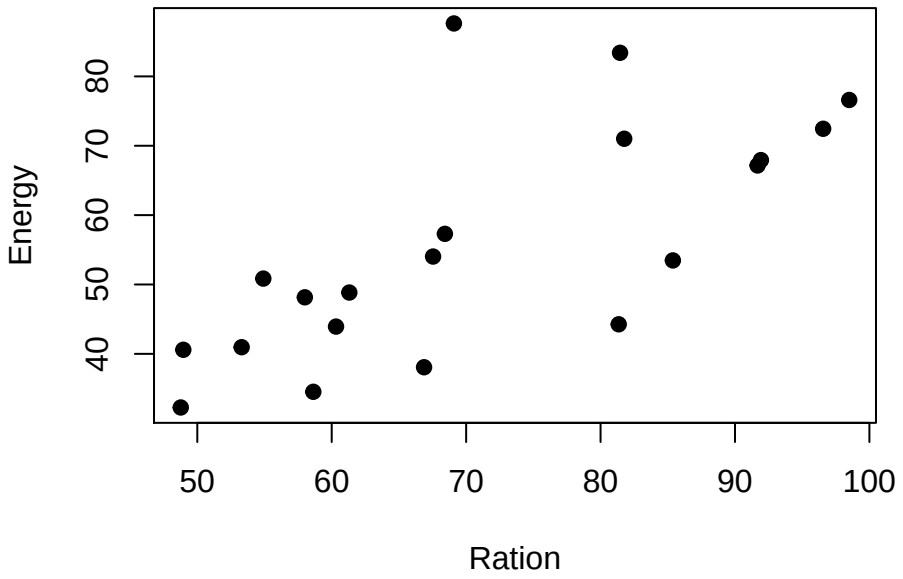
```
# read in data and show preview
trout <- read.csv("data/lm/trout.csv")

# 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")
```



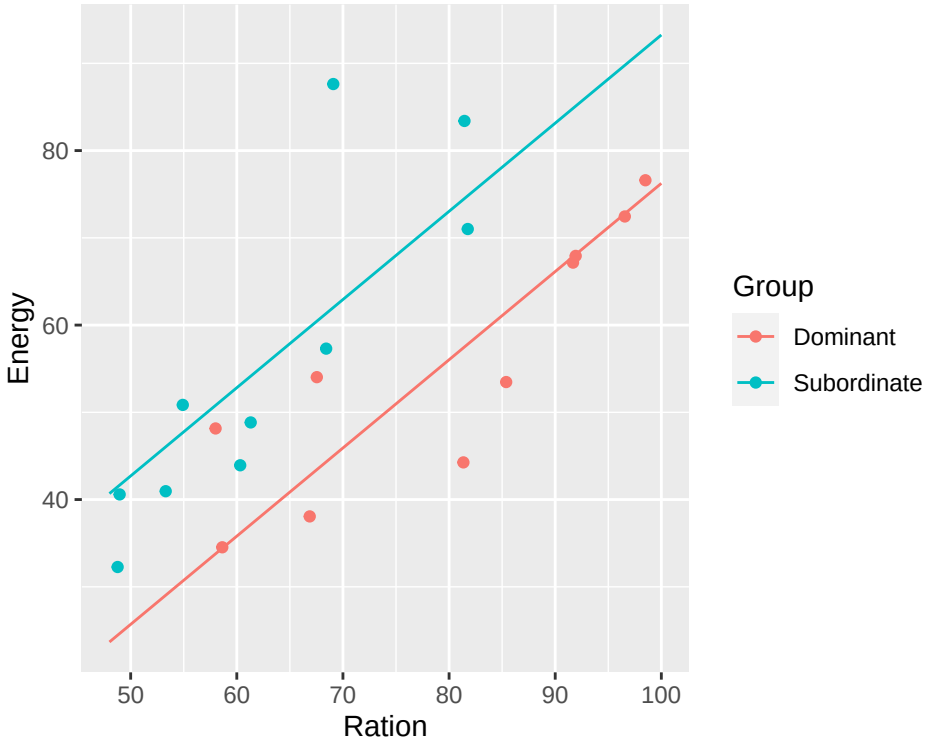

```
# scatter plot of Ration vs. Energy
plot(trout$Ration, trout$Energy, pch=19, xlab="Ration", ylab="Energy")
```



- 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 boxplots we see that the ration obtained may be different in two groups

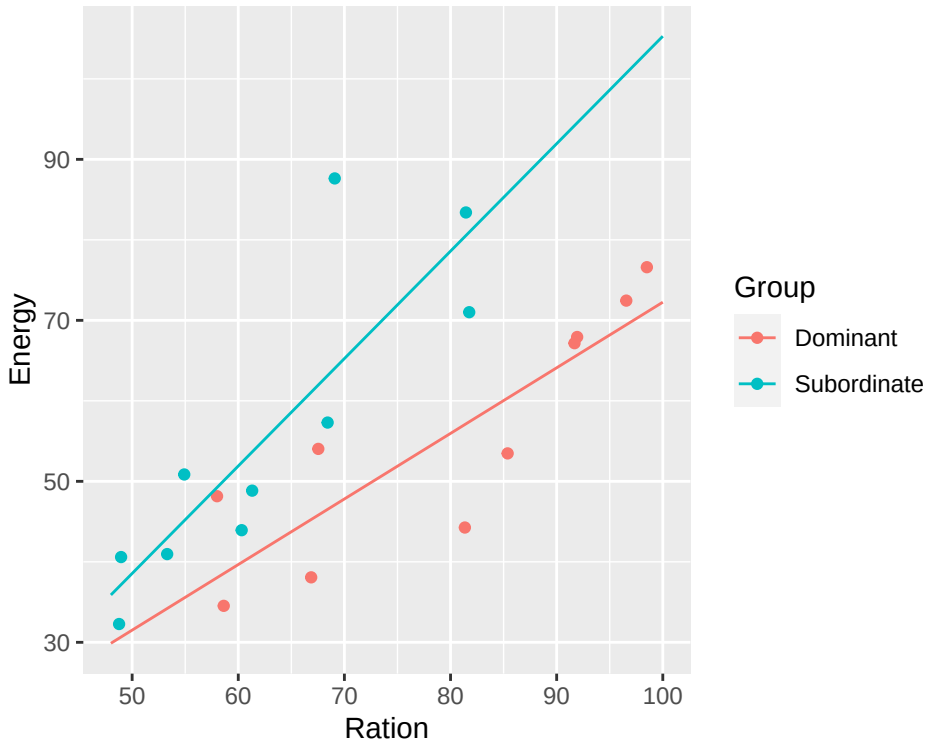
```
# Is there a relationship between ration obtained and energy expenditure
model1 <- lm(Energy ~ Ration, data = trout)
print(summary(model1))
##
## Call:
## lm(formula = Energy ~ Ration, data = trout)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.704  -4.703  -0.578   2.432  33.506
##
## 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 expend
```

```r
# Is there a relationship between ration obtained and energy expenditure different for
# 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)
```




```
# and whether there is 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)
```



Based on the regression output and plots we can say:

- there is 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

Chapter 10

Model summary & assumptions

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

10.1 Assessing model fit

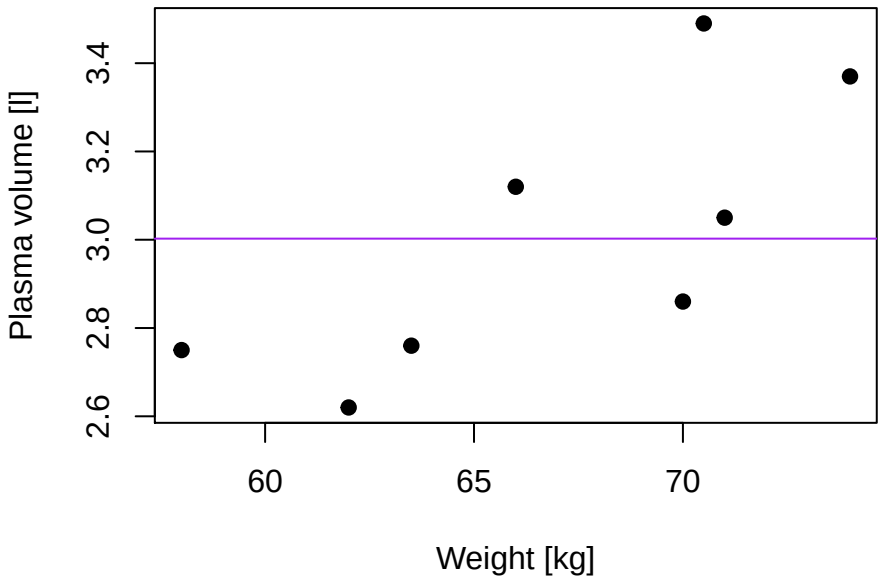
- earlier we learned how to estimate parameters in a liner model using least squares
- now we will consider how to assess the goodness of fit of a model
- we do that by calculating the amount of variability in the response that is explained by the model

10.2 R^2: summary of the fitted model

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

  $$Y_i = \beta_0 + \epsilon_i$$

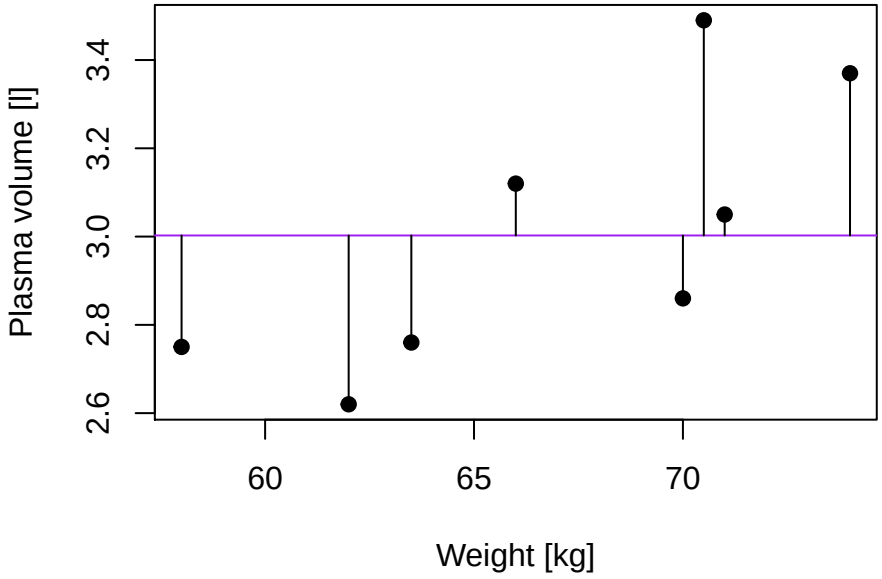
  that corresponds to a line that run through the data but lies parallel to the horizontal axis
- in our plasma volume example that would correspond the mean value of plasma volume being predicted for any value of weight (in purple)



- 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

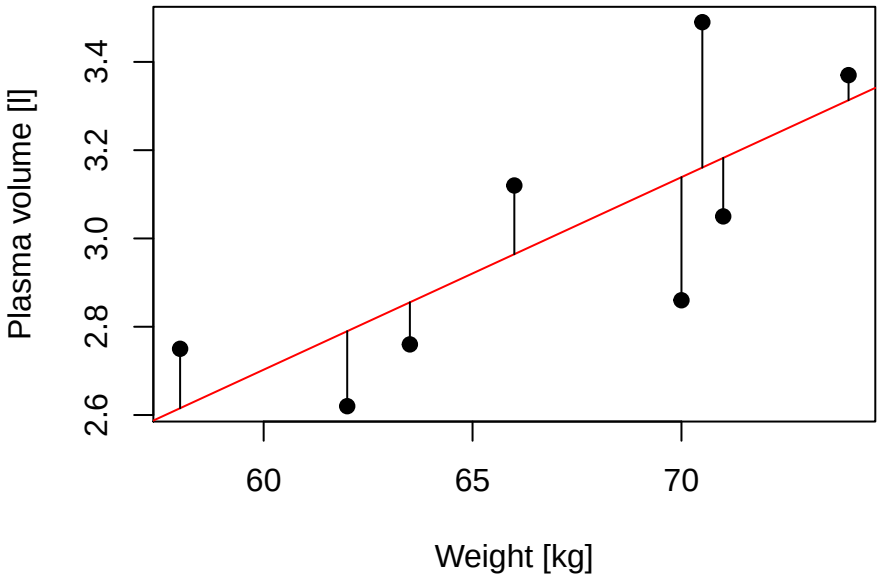
$$Y_i = \beta_0 + \beta_1 x + \epsilon_i$$

we have earlier defined

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

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

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



Definition 10.1. 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 he 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

10.3 R^2 and correlation coefficient

Theorem 10.1. *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*

10.4 $R^2(adj)$

- 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 10.2. *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
```

10.5 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

10.6 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

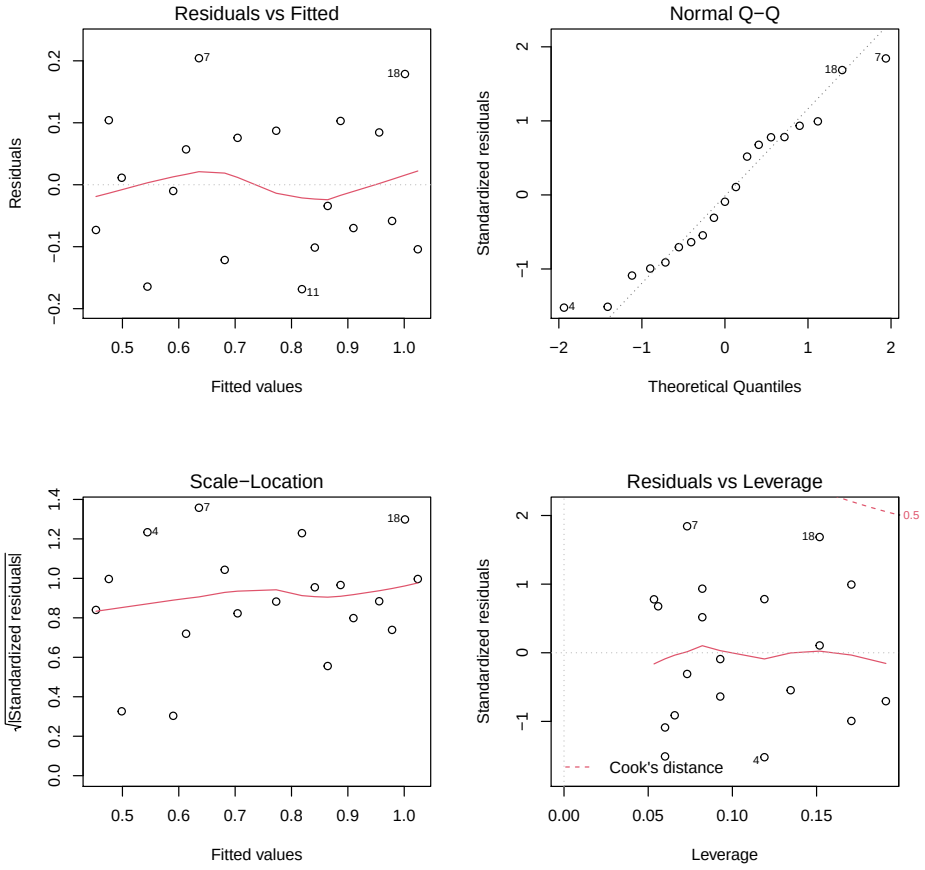
4. Plots of $\hat{\epsilon_i}$ in the order of the observations were collected

- 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

```
# 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 scatted)
- points lie reasonably well along the line in the normal probability plot, hence normality appears to be met

Examples of assumptions not being met

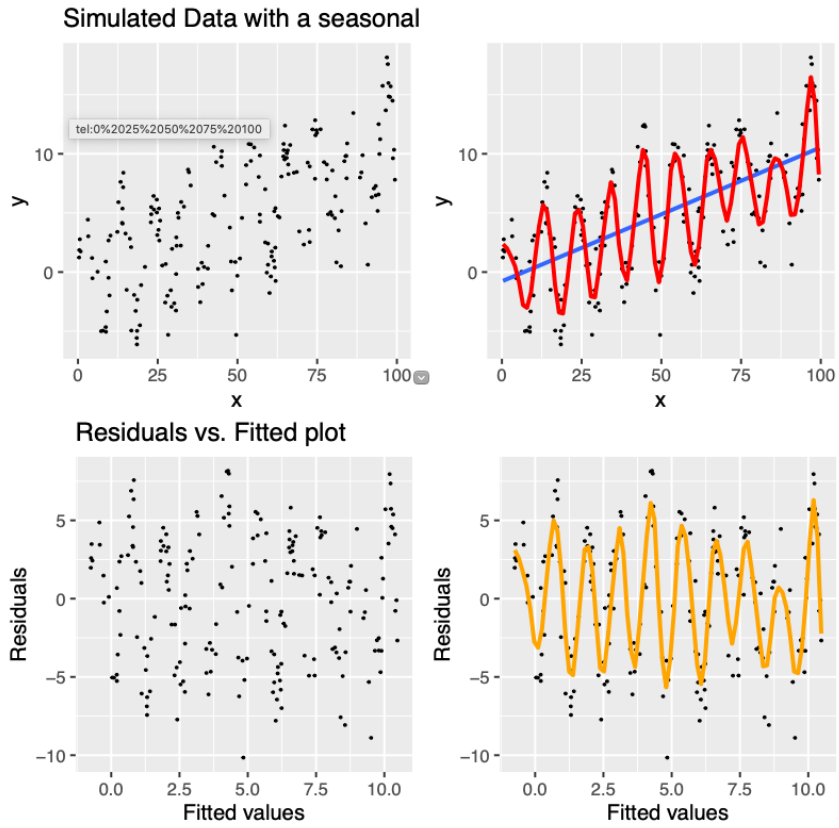


Figure 10.1: Example of data with a typical seasonal variation (up and down) coupled wtih a linear trend. The blue line gives the linear regression fit to the data, which clearly is not adequate. In comparison, if we used a non-parametric fit, we will get the red line as the fitted relationship. The residual plot retains pattern, given by orange line, indicating that the linear model is not appropriate in this case.

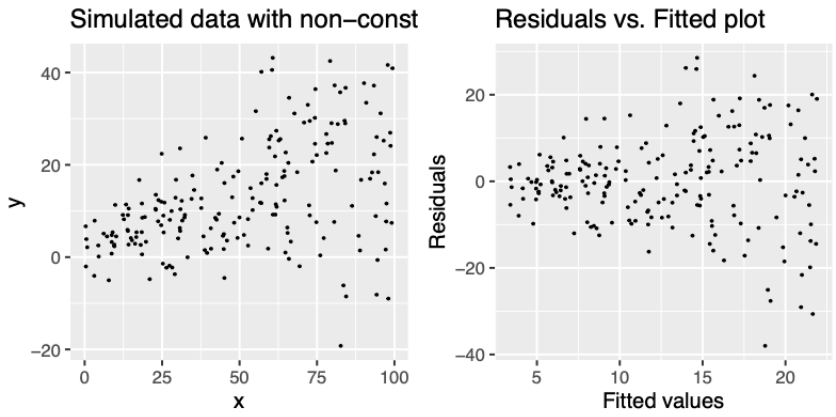


Figure 10.2: Example of non-constant variance

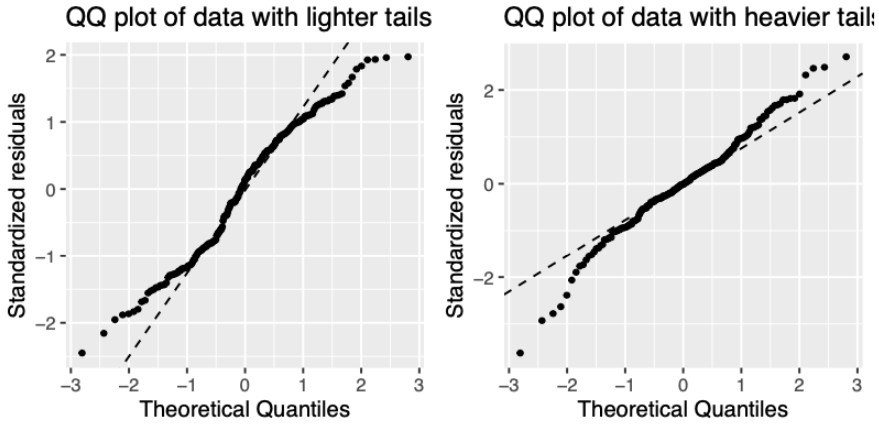


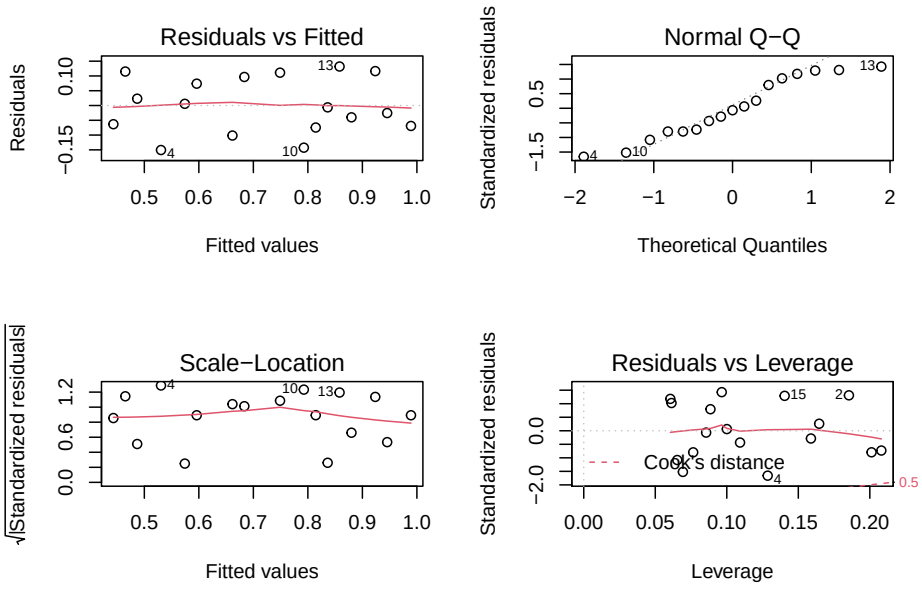
Figure 10.3: Example of residulas deviating from QQ plot, i.e. not following normal distribution. The residuals can deviate in both upper and lower tail. On the left tails are lighter meaning that they have smaller values that what would be expected, on the right there are heavier tails with values larger than expected

10.7 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
- 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 $\hat{y_{j(i)}}$ 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)
```



10.8 Exercises: linear models III

Data for exercises

- Link 1
- Alternative Link 2

Exercise 10.1. 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")
```

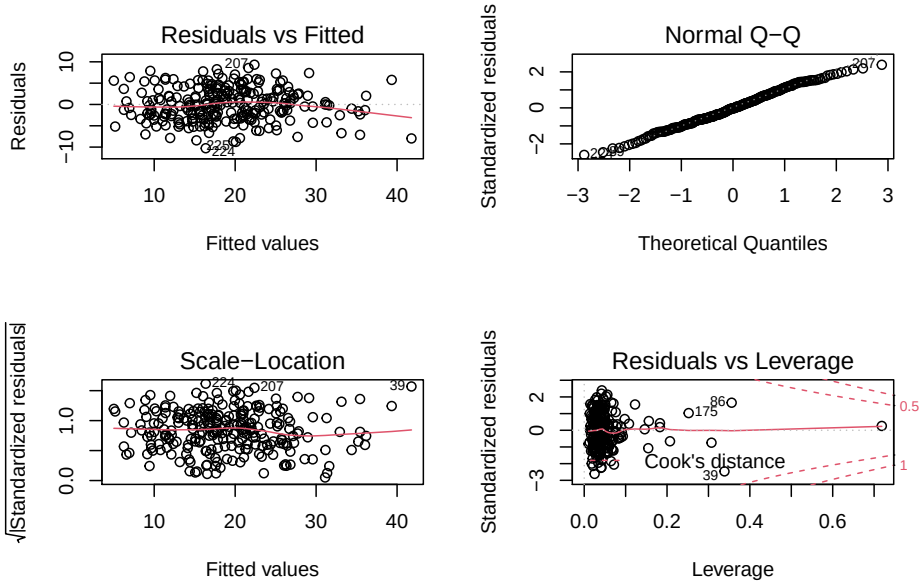
Answers to selected exercises (linear models III)

Exr. 10.1

```
# 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 + bice

# 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 +

# 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 + for
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 +
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
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 = fa
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
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.

Chapter 11

Generalized linear models

Aims

- to briefly introduce GLMs via examples of modeling binary and count response

Learning outcomes

- to understand the limits of linear regression and the application of GLMs
- to be able to use glm() function to fit and interpret logistic and Poisson regression

11.1 Why Generalized Linear Models (GLMs)

- GLMs extend linear model framework to outcome variables that do not follow normal distribution
- They are most frequently used to model binary, categorical or count data
- In the Galapagos Island example we have tried to model Species using linear model
- It kind of worked but the predicted counts were not counts (natural numbers) but rational numbers instead that make no sense when taking about count data
- Similarly, fitting a regression line to binary data yields predicted values that could take any value, including < 0
- not to mention that it is hard to argue that the values of 0 and 1s are normally distributed

11.2 Warm-up

- go to the form https://forms.gle/wKcZns85D9AN86KD6
- there is a link to a short video.

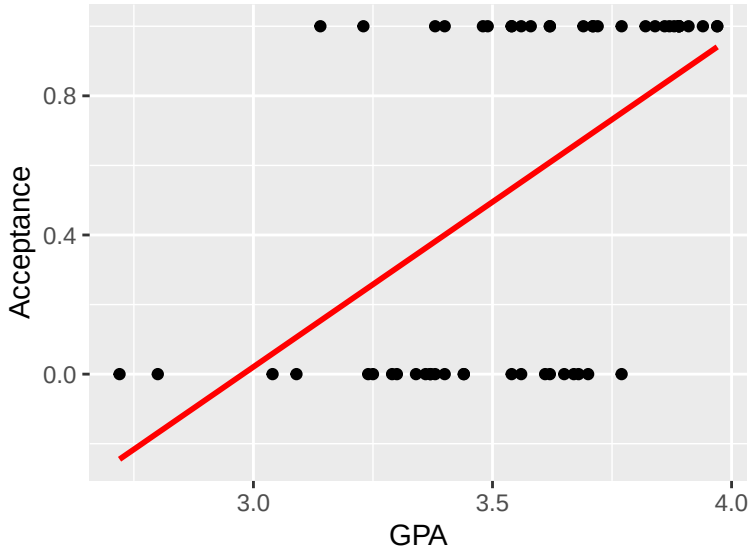


Figure 11.1: Example of fitting linear model to binary data, to model the acceptance to medical school, coded as 1 (Yes) and 0 (No) using GPA school scores. Linear model does not fit the data well in this case

- list to the video, what do you hear to begin with? Answer the question in the form and give us a little bit insignificant information about yourself (it is anonymous).

11.3 Logisitc regression

- Yanny or Laurel auditory illusion appeared online in May 2018. You could find lots of information about it, together with some plausible explanations why some people hear Yanny and some year Laurel
- One of the explanation is that with age we lose the ability to hear certain sounds
- To see if there is evidence for that, someone has already collected some data for 53 people including their age and gender

```
# Read in and preview data
yl <- read.csv("data/lm/yanny-laurel.csv")
head(yl)
##     hear age gender
## 1  Yanny  40 Female
## 2  Yanny  48   Male
## 3  Yanny  32 Female
## 4 Laurel  47 Female
## 5 Laurel  60   Male
```

```
## 6  Yanny  11 Female

# Recode Laurel to 0 and Yanny as 1 in new variable (what)
yl$word <- 0
yl$word[yl$hear=="Laurel"] <- 1

# Make some exploratory plots
par(mfrow=c(1,2))
plot(yl$age, yl$word, pch=19, xlab="age", ylab="", las=1)
boxplot(yl$age~yl$hear, xlab="", ylab="age", col="lightblue")
```

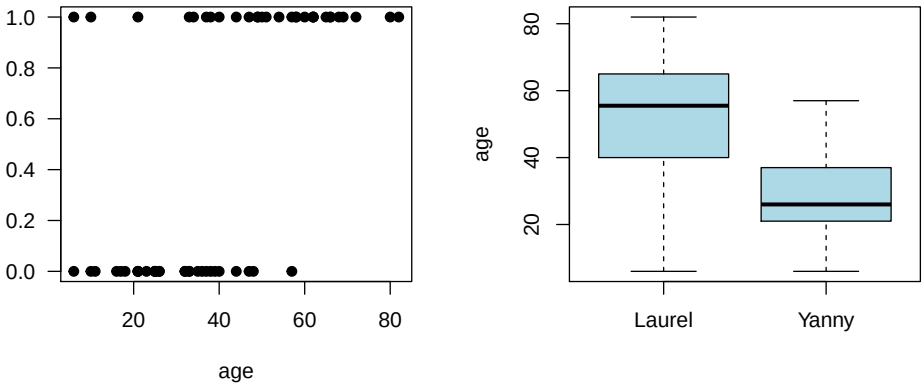


Figure 11.2: Yanny and Laurel auditory illusion data, Yanny (1), Luarel (0)

- Since the response variable takes only two values (Yanny or Laurel) we use GLM model
- to fit **logistic regression** model for the **probability of hearing Yanny**
- we let $p_i = P(Y_i = 1)$ denote the probability of hearing Yanny
- we further assume that $Y_i \sim Bin(1, p_i)$ distribution with

$$log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_i$$

this is equivalent to:

$$p_i = \frac{exp(\beta_0 + \beta_1 x_i)}{1 + exp(\beta_0 + \beta_1 x_i)}$$

- **link function** $log(\frac{p_i}{1-p_i})$ provides the link between the distribution of Y_i and the linear predictor η_i
- **GLM model** can be written as $g(\mu_i) = \eta_i = \mathbf{X}\beta$
- we use `glm()` function in R to fit GLM models

```r
# fit logistic regression model
logmodel.1 <- glm(word ~ age, family = binomial(link="logit"), data = yl)

# print model summary
print(summary(logmodel.1))
```

```
##
## Call:
## glm(formula = word ~ age, family = binomial(link = "logit"),
##     data = yl)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.86068  -0.71414  -0.04733   0.64434   2.47887
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.56159    0.95790  -3.718 0.000201 ***
## age          0.08943    0.02297   3.893 9.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.178  on 59  degrees of freedom
## Residual deviance: 57.967  on 58  degrees of freedom
## AIC: 61.967
##
## Number of Fisher Scoring iterations: 4
```

```r
# plot
ggPredict(logmodel.1)
```

```r
# to get predictions use predict() functions
# if no new observations is specified predictions are returned for the values of explo
# we specify response to return prediction on the probability scale
predict(logmodel.1, type="response")
```

```
##         1         2         3         4         5         6         7
## 0.50394192 0.67507724 0.33187793 0.65516152 0.85868941 0.07057982 0.87903410
##         8         9        10        11        12        13        14
## 0.06493370 0.35199768 0.69437930 0.91222027 0.15663558 0.78037334 0.43719992
##        15        16        17        18        19        20        21
## 0.39379165 0.59230535 0.20986467 0.45931517 0.69437930 0.22507939 0.48159183
##        22        23        24        25        26        27        28
## 0.20986467 0.71302217 0.10615359 0.97322447 0.65516152 0.11494328 0.20986467
```

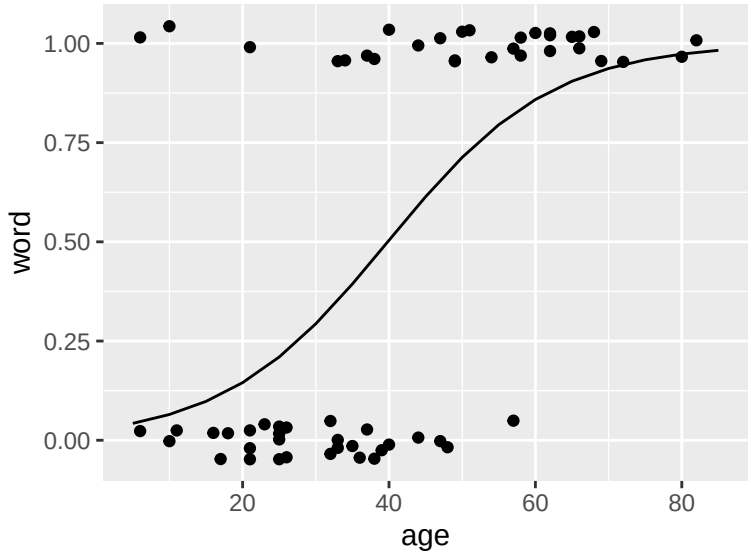


Figure 11.3: Fitted logistic model to the Yanny and Laurel data

```
##          29          30          31          32          33          34          35
## 0.12435950 0.90478991 0.87903410 0.93146108 0.15663558 0.18173908 0.33187793
##          36          37          38          39          40          41          42
## 0.59230535 0.94673070 0.06493370 0.82290366 0.91222027 0.82290366 0.92552645
##          43          44          45          46          47          48          49
## 0.83556313 0.04630975 0.50394192 0.37265683 0.97751124 0.45931517 0.41533155
##          50          51          52          53          54          55          56
## 0.35199768 0.04630975 0.83556313 0.15663558 0.20986467 0.22507939 0.15663558
##          57          58          59          60
## 0.73096842 0.35199768 0.43719992 0.87903410
```

- The regression equation for the fitted model is:

$$log(\frac{\hat{p_i}}{1-\hat{p_i}}) = -3.56 + 0.09x_i$$

- we see from the output that $\hat{\beta_0} = -3.56$ and $\hat{\beta_1} = 0.09$
- these estimates are arrived at via maximum likelihood estimation, something that is out of scope here
- but similarly to linear models, we can test the null hypothesis $H_0 : \beta_1 = 0$ by comparing, $z = \frac{\hat{\beta_1}}{e.s.e(\hat{\beta_1})} = 3.89$ with a standard normal distribution, **Wald test**, and the associated value is small meaning that there is enough evidence to reject the null, meaning that age is significantly associated with the probability with hearing Laurel and Yanny
- the same conclusion can be reached if we compare the **residual deviance**

Deviance

- we use saturated and residual deviance to assess model, instead of R^2 or $R^2(adj)$
- for a GLM model that fits the data well the approximate deviance D is

$$\chi^2(m-p)$$

  where m is the number of parameters in the saturated model (full model) and p is the number of parameters in the model of interest
- for our above model we have $83.178 - 57.967 = 25.21$ which is larger than 95th percentile of $\chi^2(59-58)$

```
qchisq(df=1, p=0.95)
## [1] 3.841459
```

- i.e. $25.21 >> 3.84$ and again we can conclude that age is a significant term in the model

Odds ratios

- In logistic regression we often interpret the model coefficients by taking $e^{\hat{\beta}}$
- and we talk about **odd ratios**
- e.g. we can say, given our above model, $e^{0.08943} = 1.093551$ that for each unit increase in age the odds of hearing Laurel get multiplied by 1.09

Other covariates

- Finally, we can use the same logic as in multiple regression to expand by models by additional variables, numerical, binary or categorical
- E.g. we can test whether there is a gender effect when hearing Yanny or Laurel

```
# fit logistic regression including age and gender
logmodel.2 <- glm(word ~ age + gender, family = binomial(link="logit"), data = yl)

# print model summary
print(summary(logmodel.2))
##
## Call:
## glm(formula = word ~ age + gender, family = binomial(link = "logit"),
##     data = yl)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.81723  -0.72585  -0.06218   0.67360   2.44755
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -3.72679    1.07333  -3.472 0.000516 ***
## age          0.09061    0.02337   3.877 0.000106 ***
## genderMale   0.23919    0.65938   0.363 0.716789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.178  on 59  degrees of freedom
## Residual deviance: 57.835  on 57  degrees of freedom
## AIC: 63.835
##
## Number of Fisher Scoring iterations: 5

# plot model
ggPredict(logmodel.2)
```

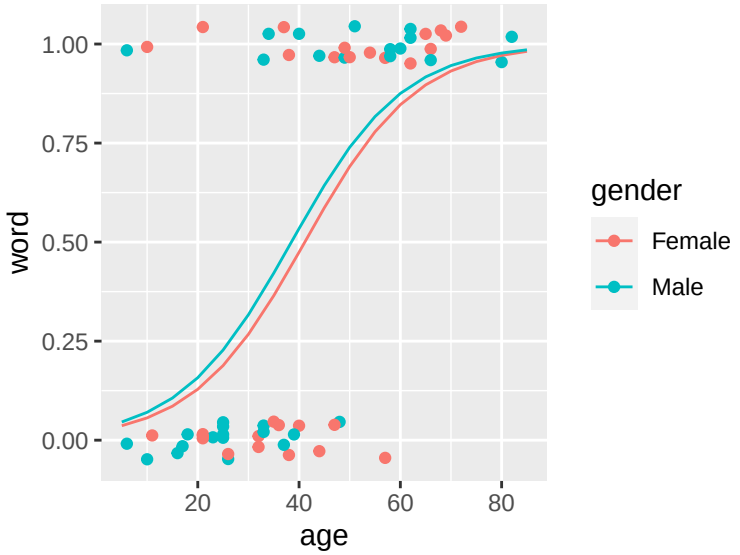


Figure 11.4: Yanny Laurel data modelled with logistic regression given age and gender. Regression lines in males and femals are very alike and the model suggest no gender effect

11.4 Poisson regression

- GLMs can be also applied to count data
- e.g. hospital admissions due to respiratory disease or number of bird nests in a certain habitat

- here, we commonly assume that data follow the Poisson distribution $Y_i \sim Pois(\mu_i)$
- and the corresponding model is

$$E(Y_i) = \mu_i = \eta_i e^{\mathbf{x_i}^T \beta}$$

with a log link $\ln \mu_i = \ln \eta_i + \mathbf{x_i}^T \beta$

Data set Suppose we wish to model Y_i the number of cancer cases in the i-th intermediate geographical location (IG) in Glasgow. We have collected data for 271 regions, a small areas that contain between 2500 and 6000 people. Together with cancer occurrence with have data:

- Y_all: number of cases of all types of cancer in te IG in 2013
- E_all: expected number of cases of all types of cancer for the IG based on the population size and demographics of the IG in 2013
- pm10: air pollution
- smoke: percentage of people in an area that smoke
- ethic: percentage of people who are non-white
- logpice: natural log of average house price
- easting and northing: co-ordinates of the central point of the IG divided by 10000

We can model the **rate of occurrence of cancer** using the very same `glm` function:¨ - now we use **poisson family distribution** to model counts - and we will include an **offset term** to model as we are modeling the rate of occurrence of the cancer that has to be adjusted by different number of people living in different regions

```
# Read in and preview data
cancer <- read.csv("data/lm/cancer.csv")
head(cancer)
##          IG Y_all     E_all pm10 smoke ethnic log.price  easting northing
## 1 S02000260   133 106.17907 17.8  21.9   5.58  11.59910 26.16245 66.96574
## 2 S02000261    38  62.43131 18.6  21.8   7.91  11.84940 26.29271 67.00278
## 3 S02000262    97 120.00694 18.6  20.8   9.58  11.74106 26.21429 67.04280
## 4 S02000263    80 109.10245 17.0  14.0  10.39  12.30138 25.45705 67.05938
## 5 S02000264   181 149.77821 18.6  15.2   5.67  11.88449 26.12484 67.09280
## 6 S02000265    77  82.31156 17.0  14.6   5.61  11.82004 25.37644 67.09826

# fit Poisson regression
epid1 <- glm(Y_all ~ pm10 + smoke + ethnic + log.price + easting + northing + offset(l
             family = poisson,
             data = cancer)

print(summary(epid1))
##
## Call:
```

```
## glm(formula = Y_all ~ pm10 + smoke + ethnic + log.price + easting +
##     northing + offset(log(E_all)), family = poisson, data = cancer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2011  -0.9338  -0.1763   0.8959   3.8416
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.8592657  0.8029040  -1.070 0.284531
## pm10         0.0500269  0.0066724   7.498 6.50e-14 ***
## smoke        0.0033516  0.0009463   3.542 0.000397 ***
## ethnic      -0.0049388  0.0006354  -7.773 7.66e-15 ***
## log.price   -0.1034461  0.0169943  -6.087 1.15e-09 ***
## easting     -0.0331305  0.0103698  -3.195 0.001399 **
## northing     0.0300213  0.0111013   2.704 0.006845 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 972.94  on 270  degrees of freedom
## Residual deviance: 565.18  on 264  degrees of freedom
## AIC: 2356.2
##
## Number of Fisher Scoring iterations: 4
```

Hypothesis testing, model fit and predictions

- follows stay the same as for logistic regression

Rate ratio

- similarly to logistic regression it common to look at the e^{β}
- for instance we are interested in the effect of air pollution on health, we could look at the pm10 coefficient
- coefficient is positive, 0.0500269, indicating that cancer incidence rate increase with increased air poluttion
- the rate ratio allows us to quantify by how much, here by a factor of $e^{0.0500269} = 1.05$

11.5 Exercises (GLMs)

Data for exercises

- Link 1
- Alternative Link 2

Exercise 11.1. Our own Yanny or Laurel dataset

Exploratory data analysis:

a) load the data that we have collected earlier on "yanny-laurel-us.cvs"
b) plot the data to explore some basic relationships between every pair of variables
c) are they any outing values, e.g. someone claiming to be too young or too skinny? Remove these observations.

Logistic regression:

d) fit a logistic linear regression model to check whether age is associated with the probability of hearing Laurel? What are the odds of hearing Laurel in our group when we get one year older?
e) are any other variables associated with hearing different words? Height? Weight? BMI? Commuting by bike?
f) if someone is 40 years of old, with 162cm, 60kg and cycling to work, what is the most likely world he/she hears?

Poisson regression:

f) fit a Poisson regression to model number of Facebook friends (no need to use offset here)
g) can you explain the number of Facebook friends with any of the variables collected?
h) if someone is 40 years of old, with 162cm, 60kg and cycling to work, how many Facebook friends he/she has?

Exercise 11.2. Additional practice

More practice with bigger more realistic data set. We have not analyzed the data ourselves yet. Let us know what you find. Anything goes.

What might affect the chance of getting a heart disease? One of the earliest studies addressing this issue started in 1960 in 3154 healthy men in the San Francisco area. At the start of the study all were free of heart disease. Eight years later the study recorded whether these men now suffered from heart disease (chd), along with many other variables that might be related. The data is available from faraway package:

- using logistic regression, can you discover anything interesting about the probability of developing heart disease
- using Poisson regression, can you comment about number of cigarettes smoked?

```
library(faraway)
data(wcgs, package="faraway")
```

```
head(wcgs)
##      age height weight sdp dbp chol behave cigs dibep chd  typechd timechd
## 2001  49     73    150 110  76  225     A2   25     B  no     none    1664
## 2002  42     70    160 154  84  177     A2   20     B  no     none    3071
## 2003  42     69    160 110  78  181     B3    0     A  no     none    3071
## 2004  41     68    152 124  78  132     B4   20     A  no     none    3064
## 2005  59     70    150 144  86  255     B3   20     A yes infdeath    1885
## 2006  44     72    204 150  90  182     B4    0     A  no     none    3102
##        arcus
## 2001  absent
## 2002 present
## 2003  absent
## 2004  absent
## 2005 present
## 2006  absent
```

Part III

Misc

Chapter 12

Classification with knn and decision trees

Aims

- to introduce classification with knn and decision trees

Learning outcomes

- to understand the concepts of splitting data into training, validation and test set
- to be able to calculate overall and class specific classification rates
- to use knn() function to select run the optimal value of k and build knn classifier
- to use rpart() function to fit and optimize a decision tree
- to use knn and a decision tree for prediction

12.1 Classification

- Classification methods are prediction models and algorithms use to classify or categorize objects based on their measurements
- They belong under **supervised learning** as we usually start off with **labeled** data, i.e. observations with measurements for which we know the label (class) of
- If we have a pair $\{\mathbf{x_i}, g_i\}$ for each observation i, with $g_i \in \{1, \dots, G\}$ being the class label, where G is the number of different classes and $\mathbf{x_i}$ a set of exploratory variables, that can be continuous, categorical or a mix of both, then we want to find a **classification rule** $f(.)$ (model) such that

$$f(\mathbf{x_i}) = g_i$$

12.2 Evaluating Classification Model Performance

- Once we have a classification model we need some way of evaluating how well it works and how it compares to other models
- There are few measures being used that involve looking at the truth (labels) and comparing it to what was predicted by the model
- Common measures include: correct (overall) classification rate, Missclassification rate, class specific rates, cross classification tables, sensitivity and specificity and ROC curves

Correct (miss)classification rate

- the simplest way to evaluate in which we count for all the n predictions how many times we got the classification right

$$Correct\ Classifcation\ Rate = \frac{\sum_{i=1}^{n} 1[f(x_i) = g_i]}{n}$$

  where 1[] is an indicator function equal to 1 if the statement in the bracket is true and 0 otherwise
- Missclassification Rate = 1 - Correct Classification Rate

Class specific rates and cross classification table

$$CCR\ for\ class\ j = \frac{number\ of\ observations\ in\ class\ j\ that\ were\ correctly\ classified}{number\ of\ observations\ in\ class\ j} = \sum_{i:g_i=j} \frac{1[f(\mathbf{x_i}) \neq$$

Example

```
# Example data
true.clas <- c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2)
pred.class <-  c(1, 1, 2, 1, 1, 2, 1, 1, 2, 2)

# correct classification rate
n <- length(true.clas)
ccr <- sum(true.clas == pred.class)/n
print(ccr)
## [1] 0.6

# cross classification table
tab.pred <- table(true.clas, pred.class)
print(tab.pred)
##           pred.class
## true.clas 1 2
##         1 4 2
##         2 2 2
```

```
# cross classification rate
# we divide each row by its sum (using sweep function)
tab.rate <- sweep(tab.pred, 1, apply(tab.pred, 1, sum), "/")
tab.rate <- round(tab.rate, 2)
print(tab.rate)
##           pred.class
## true.clas    1    2
##         1 0.67 0.33
##         2 0.50 0.50
```

12.3 Data splitting

- part of the issue of fitting complex models to data is that the model can be continually tweaked to adapt as well as possible
- but the results may not be generalizable to future data due to the added complexity modeling noise that is unique to a particular dataset (overfitting)
- to deal with overconfident estimation of future performance we randomly split data into training data, validation data and test data
- common split strategy are 50%/25%/25% and 33%/33%/33% for training/validation/test
- **training data**: this is data to give fit (train) the classification model, i.e. derive the classification rule
- **validation data**: this is data used to select which parameters or types of model perform best, i.e. to validate the performance of model parameters
- **test data**: this data is used to give an estimate of future prediction performance for the model and parameters chosen

12.4 Cross validation

- the could happen that despite random splitting in train/validation/test dataset one of the subsets does not represent data (i.e. gets all the difficult observation to classify)
- or that we do not have enough data in each subset after performing the split
- In **K-fold cross-validation** we split data into K roughly equal-sized parts
- We start by setting the validation data to be the first set of data and the training data to be all other sets
- We estimate the validation error rate / correct classification rate for the split
- We then repeat the process $K-1$ times, each time with a different part of the data set to be the validation data and the remainder being the training data

- We finish with K different error of correct classification rates
- In this way, every data point has its class membership predicted once
- The final reporter error rate is usually the average of K error rates

Now we know how to assess our classification models. Let's try it out on two methods, k-nearest neighbors and decision tree

12.5 k-nearest neighbours

- k-nearest neighbours (knn) is a non-parametric classification method, i.e. we do not have to assume a parametric model for the data of the classes
- there is no need to worry about the diagnostic tests for

Algorithm

- Decide on the value of k
- Calculate the distance between the query-instance (new observation) and all the training samples
- Sort the distances and determine the nearest neighbours based on the k-th minimum distance
- Gather the categories of the nearest neighbours
- Use simple majority of the categories of nearest neighbours as the prediction value of the new observation

Euclidean distance is a classic distance used with knn; other distance measures are also used incl. weighted Euclidean distance, Mahalanobis distance, Manhatan distance, maximum distance etc.

choosing k

- for problems with 2 classes, choose an odd number of k to avoid ties
- use validation data to fit the model for a series of k values
- pick the value of k which results in the best model (as assessed by the method of choice, e.g. overall classification rate)

Let's see how it works in practice on a classical iris dataset containing measurements on petals and sepals as well as species information (setosa, versicolor, virginica)

```
# library with knn() function
library(class)

# preview iris dataset
head(iris)
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

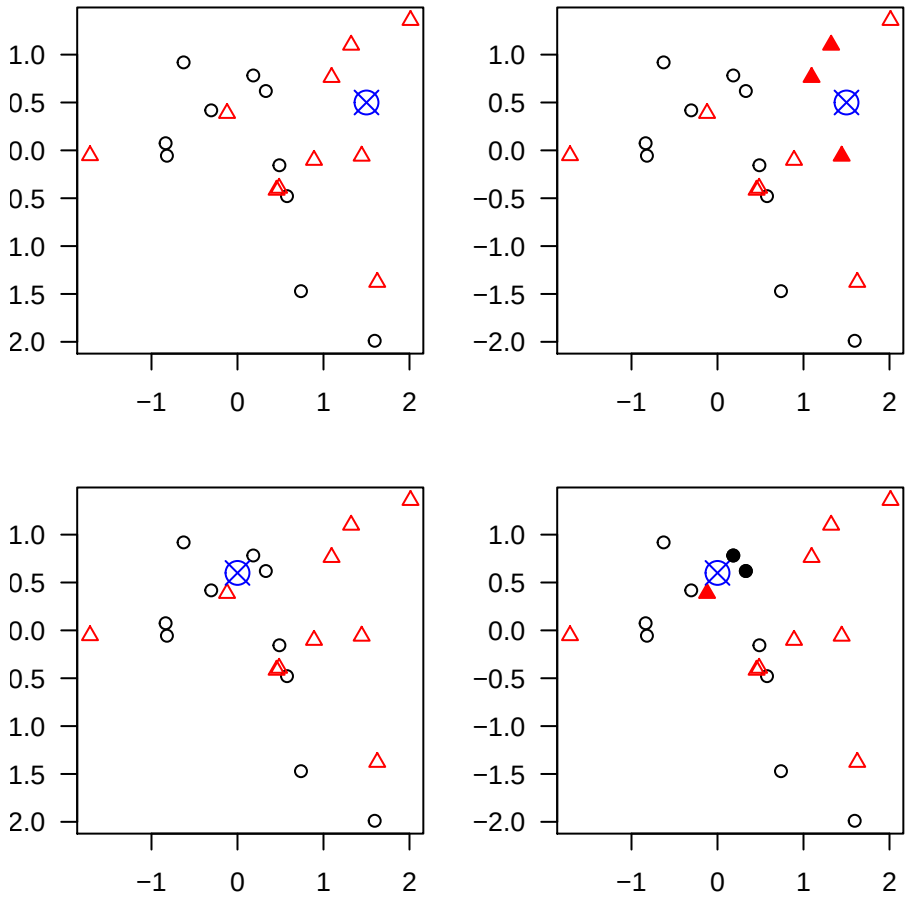
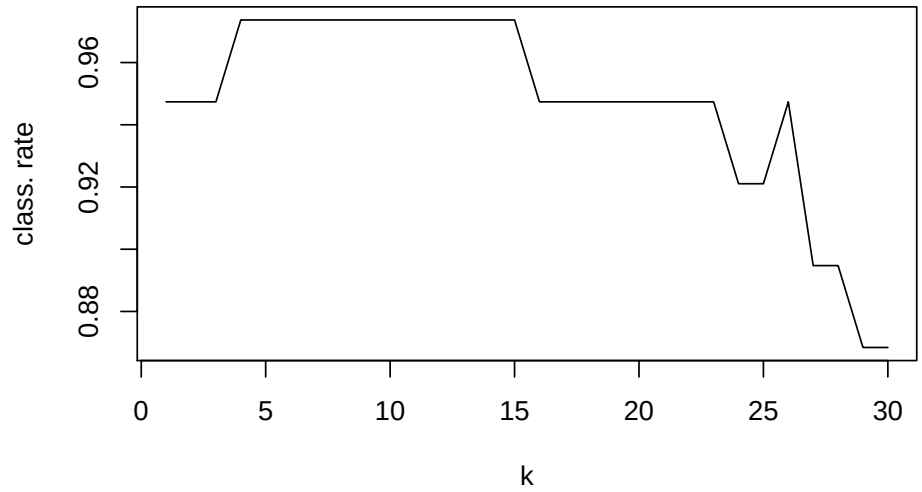


Figure 12.1: An example of k-nearest neighbours algorithm with k=3; in the top new observation (blue) is closest to three red triangales and thus classified as a red triangle; in the bottom, a new observation (blue) is closest to 2 black dots and 1 red triangle thus classified as a black dot (majority vote)

```
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

# summary statistics
summary(iris)
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##

# split data into train 50%, validation 25% and test dataset 25%
set.seed(5)
n <- nrow(iris) # no. of observations
idx.train <- sample(c(1:n), round(n/2))
idx.valid <- sample(c(1:n)[-idx.train], round(n/4))
idx.test <- setdiff(c(1:n), c(idx.train, idx.valid))

data.train <- iris[idx.train,]
data.valid <- iris[idx.valid,]
data.test <- iris[idx.test,]

dim(data.train)
## [1] 75  5
dim(data.valid)
## [1] 38  5
dim(data.test)
## [1] 37  5

# run knn with different values of k from 1 : 30
k.values <- 1:30
class.rate <- rep(0, length(k.values)) # allocate empty vector to collect correct clas
for (k in seq_along(k.values))
{
  pred.class <- knn(train = data.train[, -5], test=data.valid[, -5], cl = data.train[,
```

```
  class.rate[k] <- sum((pred.class==data.valid[,5]))/length(pred.class)
}

# for which value of k we reach the highest classification rate
which.max(class.rate)
## [1] 4

# plot classification rate as a function of k
plot(class.rate, type="l", xlab="k", ylab="class. rate")
```




```
# how would our model perform on the future data using the optimal k?
pred.class <- knn(train = data.train[, -5], data.test[, -5], data.train[,5], k=which.max(class.ra
class.rate <- sum((pred.class==data.test[,5]))/length(pred.class)
print(class.rate)
```

```
## [1] 1
```

12.6 Classification trees

- they are often used to represent knowledge and aid decision-making
- they can be easily interpretable by anyone
- similar to knn they are assumption free and can handle various data input
- they can be presented as diagrams or pseudo-code via text
- they can be used for both classification and regression
- here we will focus on classification

Terminology

- **Root node**: represents the entire population of the data set

- **Splitting**: the process of dividing a node into two or more nodes
- **Decision / internal node**: when a new node is split into further nodes
- **Leaf / terminal noel**: nodes that do not split into further nodes
- **Subtree**: a subsection of a tree
- **Branch**: a subtree that is only one side of a split from a node

To make predictions we simply travel down the tree starting from the top

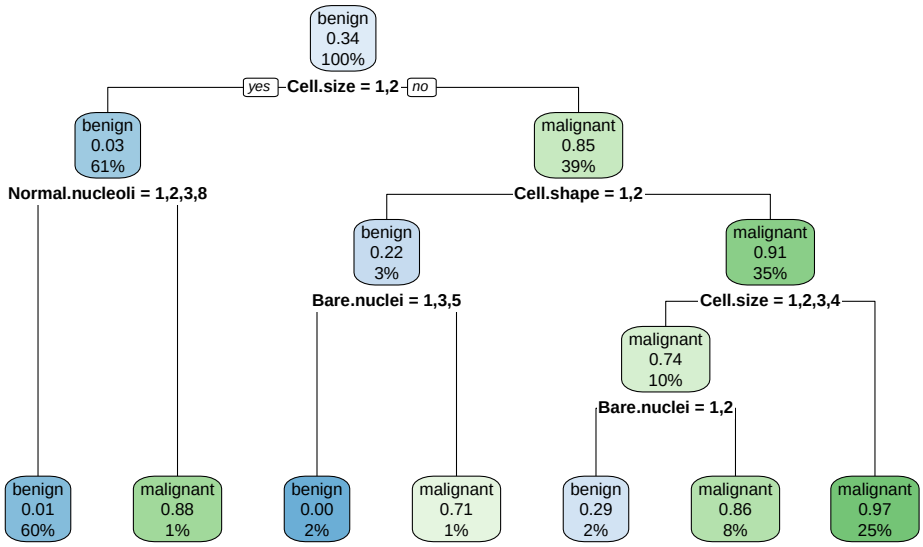


Figure 12.2: Example of the decision tree classifying tumour into bening and malignant type

Fitting trees 1. pick the variable that gives the best split (often based on the lowest Gini index) 2. partition the data based on the value of this variable 3. repeat step 1. and step 2. 4. stop splitting when no further gain can be made or some pre-set stopping rule is met Alternatively, the data is split as much as possible and the tree is **pruned**

Gini index

- measures impurity in node, an alternative way of assessing model's performance to classification rates that have been shown to result in local overfitting in decision trees
- Gini index varies between 0 and (1-1/n) where n is the number of categories in a dependent variable

$$Gini = \sum_{i=1}^{c} (p_i)^2$$

where c is the number of categories

```
## [1] 699  11
## [1] "benign"    "malignant"
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant
##
## Classification tree:
## rpart(formula = Class ~ Cl.thickness + Cell.size + Cell.shape +
##     Marg.adhesion + Epith.c.size + Bare.nuclei + Bl.cromatin +
##     Normal.nucleoli + Mitoses, data = BreastCancer)
##
## Variables actually used in tree construction:
## [1] Bare.nuclei     Cell.shape      Cell.size       Normal.nucleoli
##
## Root node error: 241/699 = 0.34478
##
## n= 699
##
##         CP nsplit rel error  xerror     xstd
## 1 0.780083      0   1.00000 1.00000 0.052142
## 2 0.053942      1   0.21992 0.26141 0.031415
## 3 0.024896      2   0.16598 0.18257 0.026644
## 4 0.012448      3   0.14108 0.16598 0.025481
## 5 0.010000      6   0.10373 0.16183 0.025180
```

Importance of the variable

- defined as the sum of goodness of split measures for each split for which it as the primary variable

```
# show variable.importance attribute
tree.1$variable.importance
##       Cell.size      Cell.shape Normal.nucleoli    Epith.c.size     Bl.cromatin
##      228.196290      195.580593      167.558093      164.615713      160.203228
##     Bare.nuclei         Mitoses    Cl.thickness   Marg.adhesion
##      154.590550        5.763756        4.301576        2.655170
```

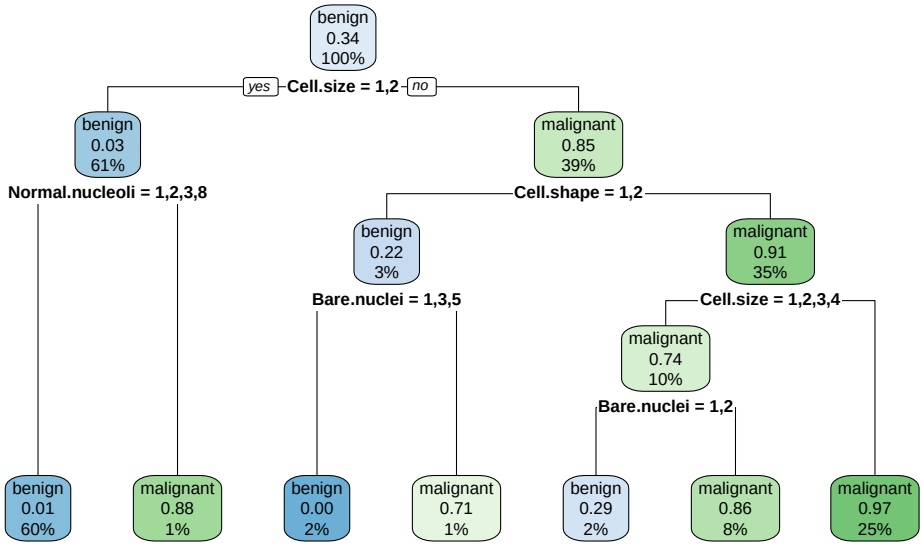


Figure 12.3: Example of the decision tree classifying tumour into bening and malignant type with rpart() default parameteres

Complexity measure of a tree

- in rpart() the complexity measure is calculated based on the size of a tree and the ability of the tree to separate the classes of the target variable
- if the next best split in growing a tree does not reduce the tree's overall complexity by a certain amount, rpart() terminates the growing process
- `cp` is the complexity parameter, set to negative amount results in a fully grown tree (maximum splits)

```
##
## Classification tree:
## rpart(formula = Class ~ Cl.thickness + Cell.size + Cell.shape +
##     Marg.adhesion + Epith.c.size + Bare.nuclei + Bl.cromatin +
##     Normal.nucleoli + Mitoses, data = BreastCancer, cp = -1)
##
## Variables actually used in tree construction:
## [1] Bare.nuclei     Bl.cromatin     Cell.shape      Cell.size
## [5] Cl.thickness    Normal.nucleoli
##
## Root node error: 241/699 = 0.34478
##
## n= 699
##
##          CP nsplit rel error  xerror     xstd
## 1  0.780083      0   1.00000 1.00000 0.052142
```

```
## 2  0.053942      1   0.21992 0.24481 0.030497
## 3  0.024896      2   0.16598 0.17842 0.026359
## 4  0.012448      3   0.14108 0.16183 0.025180
## 5  0.000000      6   0.10373 0.17012 0.025778
## 6 -1.000000     15   0.10373 0.17012 0.025778
```

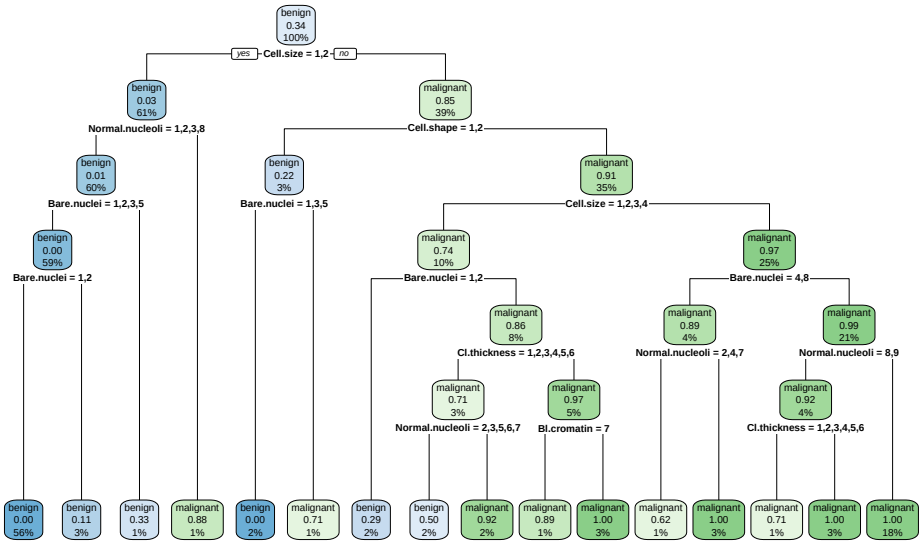


Figure 12.4: Example of the decision tree classifying tumour into bening and malignant type with rpart(), fully grown tree

Pruning a tree

- fully grown trees do not usually perform well against data not in the training set (overfitting)
- a solution to this is to reduce (prune) the tree
- typically, this is done by choosing the complexity parameter associated with the minimum possible cross-validated error
- `xerror`, in the tree view output, in our $cp = 0.14108$ in the above case

```
##
## Classification tree:
## rpart(formula = Class ~ Cl.thickness + Cell.size + Cell.shape +
##     Marg.adhesion + Epith.c.size + Bare.nuclei + Bl.cromatin +
##     Normal.nucleoli + Mitoses, data = BreastCancer, cp = -1)
##
## Variables actually used in tree construction:
## [1] Cell.size
##
## Root node error: 241/699 = 0.34478
##
```

```
## n= 699
##
##         CP nsplit rel error  xerror     xstd
## 1 0.78008      0   1.00000 1.00000 0.052142
## 2 0.14108      1   0.21992 0.24481 0.030497
##       Cell.size      Cell.shape    Epith.c.size Normal.nucleoli     Bl.cromatin
##        222.9401        174.2235        163.4894        153.5809        151.9295
##     Bare.nuclei
##        142.0211
```

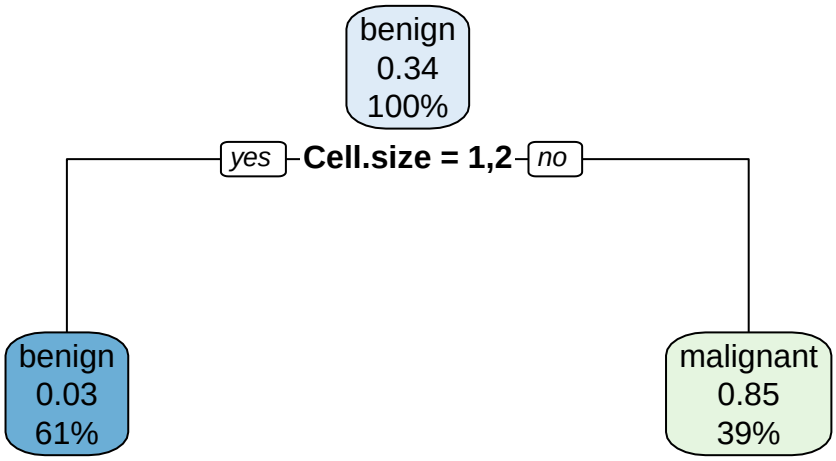


Figure 12.5: Example of the decision tree classifying tumour into bening and malignant type with rpart() pruned tree to minimize cross-validation error

Predictions of future observations

- having our best tree model we can predict the outcome of applications on a test data set and assess model performance on "unseen" data

```
# predict cancer type given tree model
cancertype <- predict(tree.2pruned, newdata = data.test, type="class")

# cross classification table
table(cancertype, data.test$Class)
##
## cancertype  benign malignant
##   benign       208         6
##   malignant     22       113
```

12.7 Exercises: classification

Exercise 12.1. knn and rpart practice

Make sure you can run and understand the above knn and rpart examples

Exercise 12.2. Comparing knn() and rpart()

Given BreastCancer data

a) build a best knn() classification model that you can to predict the cancer in BreatCancer data set
b) try improving the rpart() model, look at the documentation ?rpart.control() and try to figure out and test changing other parameters, especially `minsplit` and `minbucket`
c) compare the performance of your best knn() models with your best rpart() model on the test data
d) share which method knn or rpart performs better together with the overall classification rate on Zulip (under Day-04)

```
# Install "mlbench" package
install.packages("mlbench")
## Installing package into '/Users/olga.hrydziuszko/Desktop/bookdown-mlbiostatistics/packrat/lib/
## (as 'lib' is unspecified)
##
## The downloaded binary packages are in
##         /var/folders/hw/jx67_4vj6ljfd13xsg7xzvt83k7mrx/T//RtmpxIUSTD/downloaded_packages
library(mlbench)

# Look at the Breast Cancer data
# more about data is here: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Orig
data(BreastCancer)
dim(BreastCancer)
## [1] 699  11
levels(BreastCancer$Class)
## [1] "benign"    "malignant"
head(BreastCancer)
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
```

```
## 4            4            3            7       1    benign
## 5            1            3            1       1    benign
## 6           10            9            7       1 malignant
```

Chapter 13

ANN regression and classification

Aims

- to introduce regression and classification with ANN via examples

Learning outcomes

- to be able to use neuralnet() package for classification and regression

13.1 Exercise 1.1

13.1.1 Set-up

```
# Clean all variables and load libraries
rm(list=ls())
library(mlbench) # CancerBreast dataset
library(class) # knn
library(rpart) # decision tree
library(rpart.plot) # decision tree plots
library(neuralnet) # ann
library(NeuralNetTools) # ann tools
library(ggplot2) # plotting
library(ggiraphExtra)
```

13.1.2 Load and process the data

```
# Data input
data("BreastCancer")
```

```
head(BreastCancer) # preview data
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1           2
## 2 1002945            5         4          4             5           7
## 3 1015425            3         1          1             1           2
## 4 1016277            6         8          8             1           3
## 5 1017023            4         1          1             3           2
## 6 1017122            8        10         10             8           7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant
str(BreastCancer) # show data types
## 'data.frame':        699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ..

# data summary
summary(BreastCancer)
##        Id               Cl.thickness   Cell.size     Cell.shape  Marg.adhesion
##  Length:699         1      :145   1      :384   1      :353   1      :407
##  Class :character   5      :130   10     : 67   2      : 59   2      : 58
##  Mode  :character   3      :108   3      : 52   10     : 58   3      : 58
##                     4      : 80   2      : 45   3      : 56   10     : 55
##                     10     : 69   4      : 40   4      : 44   4      : 33
##                     2      : 50   5      : 30   5      : 34   8      : 25
##                     (Other):117   (Other): 81   (Other): 95   (Other): 63
##   Epith.c.size  Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses
##  2      :386   1      :402   2      :166   1      :443     1      :579
##  3      : 72   10     :132   3      :165   10     : 61     2      : 35
##  4      : 48   2      : 30   1      :152   3      : 44     3      : 33
##  1      : 47   5      : 30   7      : 73   2      : 36     10     : 14
##  6      : 41   3      : 28   4      : 40   8      : 24     4      : 12
```

```
##  5       : 39   (Other): 61   5       : 34   6       : 22     7       :  9
##  (Other): 66   NA's   : 16   (Other): 69   (Other): 69     (Other): 17
##        Class
##  benign   :458
##  malignant:241
##
##
##
##
##
```

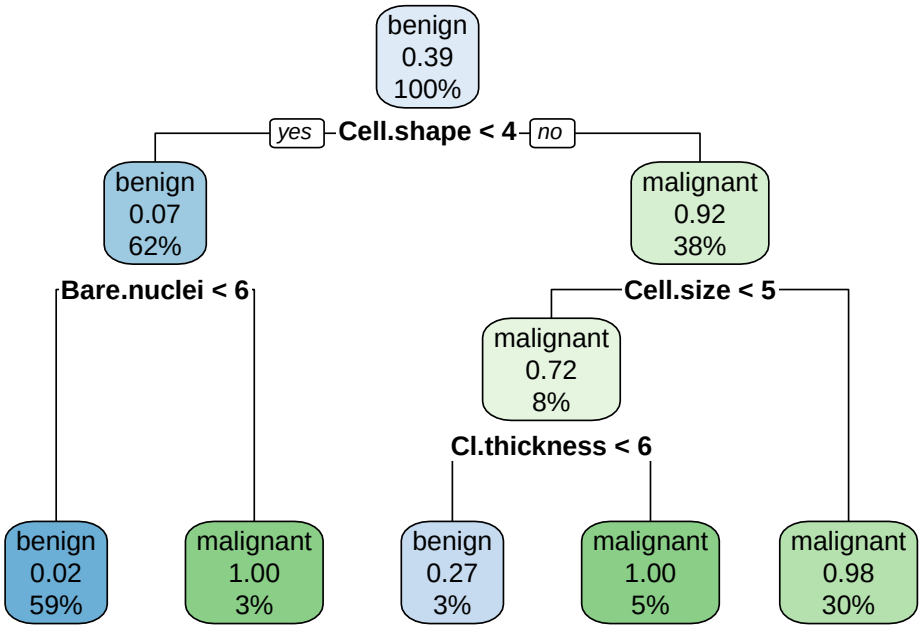
```
# convert variables to numerical values
data.breast <- BreastCancer
data.breast <- data.frame(Class = BreastCancer$Class,
                          Cell.size = as.numeric(BreastCancer$Cell.size),
                          Cell.shape = as.numeric(BreastCancer$Cell.shape),
                          Cl.thickness = as.numeric(BreastCancer$Cl.thickness),
                          Marg.adhesion = as.numeric(BreastCancer$Marg.adhesion),
                          Epith.c.size = as.numeric(BreastCancer$Epith.c.size),
                          Bare.nuclei = as.numeric(BreastCancer$Bare.nuclei),
                          Bl.cromatin = as.numeric(BreastCancer$Bare.nuclei),
                          Normal.nucleoli = as.numeric(BreastCancer$Normal.nucleoli),
                          Mitoses = as.numeric(BreastCancer$Mitoses)
                          )

str(data.breast)
## 'data.frame':        699 obs. of  10 variables:
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
##  $ Cell.size      : num  1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : num  1 4 1 8 1 10 1 2 1 1 ...
##  $ Cl.thickness   : num  5 5 3 6 4 8 1 2 2 4 ...
##  $ Marg.adhesion  : num  1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : num  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ Normal.nucleoli: num  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : num  1 1 1 1 1 1 1 1 5 1 ...

# Remove missing data
data.breast <- na.omit(data.breast)
```

13.1.3 Data split

```
# Split into train, validation, test
# split data into train 50%, validation 25% and test dataset 25%
```

```
set.seed(1)
n <- nrow(data.breast) # no. of observations
idx.train <- sample(c(1:n), round(n/2))
idx.valid <- sample(c(1:n)[-idx.train], round(n/4))
idx.test <- setdiff(c(1:n), c(idx.train, idx.valid))

data.train <- data.breast[idx.train,]
data.valid <- data.breast[idx.valid,]
data.test <- data.breast[idx.test,]

dim(data.train)
## [1] 342  10
dim(data.valid)
## [1] 171  10
dim(data.test)
## [1] 170  10
```

```
# KNN (k=3)
knn.pred <- knn(train = data.train[, -1], data.test[, -1], data.train[,1], k=3)
knn.cr <- sum((knn.pred==data.test[,1]))/length(knn.pred) # classification rate
print(knn.cr)
## [1] 0.9588235

# Decision tree
cart.1 <- rpart(data.train$Class ~ Cl.thickness + Cell.size +  Cell.shape + Marg.adhes
rpart.plot(cart.1)
```




```
cart.pred <- predict(cart.1, newdata = data.test, type="class")
cart.cr <- sum((cart.pred==data.test[,1]))/length(cart.pred)
print(cart.cr)
## [1] 0.9352941

# Logisitc regression
logreg <- glm(data.train$Class ~ Cl.thickness + Cell.size +  Cell.shape + Marg.adhesion + Epith.c
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
print(summary(logreg))
##
## Call:
## glm(formula = data.train$Class ~ Cl.thickness + Cell.size + Cell.shape +
##     Marg.adhesion + Epith.c.size + Bare.nuclei + Bl.cromatin +
##     Normal.nucleoli + Mitoses, family = binomial(link = "logit"),
##     data = data.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.78015  -0.09873  -0.04989   0.00842   2.29108
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -10.8815     2.1472  -5.068 4.02e-07 ***
## Cl.thickness      0.4050     0.2242   1.807 0.070827 .
## Cell.size         0.1039     0.2724   0.381 0.702933
```

```
## Cell.shape        0.6366     0.3678   1.731 0.083481 .
## Marg.adhesion     0.3343     0.1951   1.714 0.086534 .
## Epith.c.size     -0.4608     0.2947  -1.564 0.117860
## Bare.nuclei       0.5100     0.1525   3.343 0.000828 ***
## Bl.cromatin           NA         NA      NA       NA
## Normal.nucleoli   0.5765     0.1944   2.966 0.003015 **
## Mitoses           2.3031     1.2862   1.791 0.073361 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 457.974  on 341  degrees of freedom
## Residual deviance:  40.942  on 333  degrees of freedom
## AIC: 58.942
##
## Number of Fisher Scoring iterations: 9
logreg.prob <- predict(logreg, newdata=data.test, type="response")
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

logreg.pred <- rep("malignant", nrow(data.test))
logreg.pred[logreg.prob <= 0.5] <- "benign"
logreg.cr <- sum((logreg.pred==data.test[,1]))/length(logreg.pred)
print(logreg.cr)
## [1] 0.9411765

print(c(knn.cr, cart.cr, logreg.cr))
## [1] 0.9588235 0.9352941 0.9411765
```

13.1.5 ANN classification: fitting model

```
# fit ANN classification model (default parameters and logistic activation function)
# notice linear.output set to FALSE
ann.c1 <- neuralnet(Class ~ Cl.thickness + Cell.size +  Cell.shape +
                      Marg.adhesion + Epith.c.size + Bare.nuclei +
                      Bl.cromatin + Normal.nucleoli + Mitoses,
                      data=data.train,
                      linear.output = FALSE,
                      act.fct = "logistic")

plotnet(ann.c1, cex_val = 0.6)
```

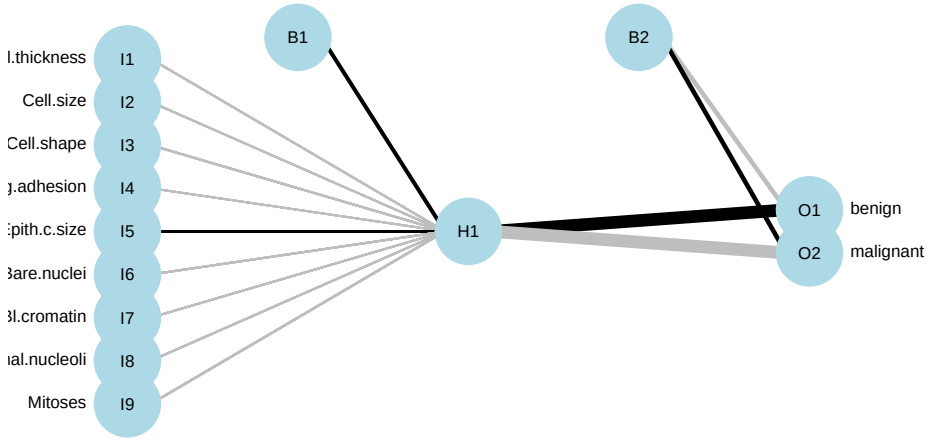


Figure 13.1: ANN model representation. The black lines show the connections between each layer and the weights on each connection; B nodes represent bias terms added in each step (bias nodes), and the hidden nodes are the ones between the input and output, here only one node

13.1.6 ANN classification: comparering to logistic regression

```
# we run prediction using compute function()
ann.c1_predictions <- neuralnet::compute(ann.c1,
                                  data.test)
ann.c1_predictions <- ann.c1_predictions$net.result

# The prediction result shows the probability of each class
# with first column corresponding to benign and second to malignant
# we know as by typing str(data.breast) we can see that our Class is ordered benign and malignant
head(ann.c1_predictions)
##           [,1]         [,2]
## 13 0.003303861 9.965274e-01
## 17 1.000000000 1.278634e-10
## 20 1.000000000 3.511953e-10
## 23 1.000000000 9.554493e-11
## 27 0.999999999 4.639298e-10
## 31 1.000000000 2.895116e-10

# So we need the extract the class with the highest prediction values as the predicted result
# e.g by using ifelse
ann.c1_pred <- max.col(ann.c1_predictions) # find out which column has the maximum value
ann.c1_pred <- ifelse(ann.c1_pred==1, "benign", "malignant") # if the first column was max, set t
ann1.cr <- sum((ann.c1_pred==data.test[,1]))/length(ann.c1_pred)
```

```
print(ann1.cr)
## [1] 0.9470588

# Compare with our previous models
print(c(knn.cr, cart.cr, logreg.cr, ann1.cr))
## [1] 0.9588235 0.9352941 0.9411765 0.9470588

# Note: we see we are not doing quite similar to logistic regression
# It is not surprising as we only have one hidden layer with only one node with logist
# and the linear regression can be viewed as a simple special case of neural network w
```

13.1.7 ANN classification: adding hidden layers

```
# Lets add some more 2 more hidden layers, with 2 and 5 nodes, via hidden parameter
# and see if our predictions improve
set.seed(1)
ann.c2 <- neuralnet(Class ~ Cl.thickness + Cell.size +  Cell.shape +
                        Marg.adhesion + Epith.c.size + Bare.nuclei +
                        Bl.cromatin + Normal.nucleoli + Mitoses,
                        data=data.train,
                        hidden = c(2,5),
                        linear.output = FALSE,
                        act.fct = "logistic")

plotnet(ann.c2, cex_val = 0.6)
```

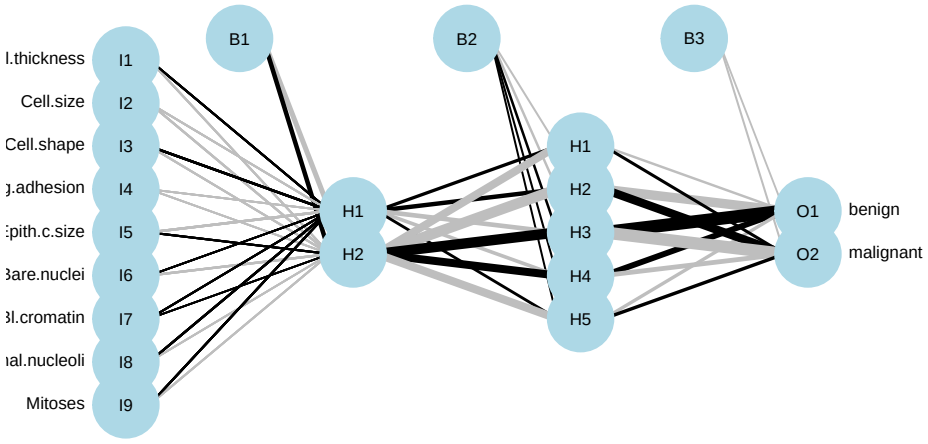


Figure 13.2: ANN model representation for classification, two hidden layers, with two and five nodes respectively

```r
ann.c2_predictions <- neuralnet::compute(ann.c2, data.test)$net.result
ann.c2_pred <- max.col(ann.c2_predictions) # find out which column has the maximum value
ann.c2_pred <- ifelse(ann.c2_pred==1, "benign", "malignant") # if the first column was max, set t
ann2.cr <- sum((ann.c2_pred==data.test[,1]))/length(ann.c2_pred)

# Compare with our previous models
print(c(knn.cr, cart.cr, logreg.cr, ann1.cr, ann2.cr))
## [1] 0.9588235 0.9352941 0.9411765 0.9470588 0.9529412

# Feel free to experiment with more layers and more nodes
```

13.1.8 ANN classification: changing activation function

```r
# For multi-class problems softmax activation function is used (equivalent to multiple logistic n
# We can change activation function via act.fct parameter and include custom functions
# e.g. softplus <- function(x) log(1 + exp(x))
# or relu <- function(x) sapply(x, function(z) max(0,z))
# here we switch to "tanh" from a default "logistic"

set.seed(101)
ann.c3 <- neuralnet(Class ~ Cl.thickness + Cell.size +  Cell.shape +
                        Marg.adhesion + Epith.c.size + Bare.nuclei +
                        Bl.cromatin + Normal.nucleoli + Mitoses,
                        data=data.train,
                        hidden = c(2,5),
                        linear.output = FALSE,
                        act.fct = "tanh")

ann.c3_predictions <- neuralnet::compute(ann.c3, data.test)$net.result
ann.c3_pred <- max.col(ann.c3_predictions) # find out which column has the maximum value
ann.c3_pred <- ifelse(ann.c3_pred==1, "benign", "malignant") # if the first column was max, set t
ann3.cr <- sum((ann.c3_pred==data.test[,1]))/length(ann.c3_pred)

# Compare with our previous models
print(c(knn.cr, cart.cr, logreg.cr, ann1.cr, ann2.cr, ann3.cr))
## [1] 0.9588235 0.9352941 0.9411765 0.9470588 0.9529412 0.7000000
```

13.1.9 ANN regression

```r
set.seed(1)

# read in data on Pima Indians Diabeses Database
```

```
data(PimaIndiansDiabetes)
head(PimaIndiansDiabetes)
##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35       0 33.6    0.627  50      pos
## 2        1      85       66      29       0 26.6    0.351  31      neg
## 3        8     183       64       0       0 23.3    0.672  32      pos
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74       0       0 25.6    0.201  30      neg
summary(PimaIndiansDiabetes)
##     pregnant         glucose         pressure         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     insulin           mass           pedigree           age         diabetes
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00   neg:500
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   pos:268
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
str(PimaIndiansDiabetes)
## 'data.frame':        768 obs. of  9 variables:
##  $ pregnant: num  6 1 8 1 0 5 3 10 2 8 ...
##  $ glucose : num  148 85 183 89 137 116 78 115 197 125 ...
##  $ pressure: num  72 66 64 66 40 74 50 0 70 96 ...
##  $ triceps : num  35 29 0 23 35 0 32 0 45 0 ...
##  $ insulin : num  0 0 0 94 168 0 88 0 543 0 ...
##  $ mass    : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ pedigree: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ age     : num  50 31 32 21 33 30 26 29 53 54 ...
##  $ diabetes: Factor w/ 2 levels "neg","pos": 2 1 2 1 2 1 2 1 2 2 ...
# pregnant: Number of times pregnant
# glucose: Plasma glucose concentration (glucose tolerance test)
# pressure: Diastolic blood pressure (mm Hg)
# triceps: Triceps skin fold thickness (mm)
# insulin: 2-Hour serum insulin (mu U/ml)
# mass: Body mass index (weight in kg/(height in m)\^2) Diabetes pedigree function
# age: Age (years)
# diabetes: Class variable (test for diabetes)

# remove missing values i.e. some mass measurements are 0
```

```r
idx.0 <- which(PimaIndiansDiabetes$mass == 0)
data.pima <- PimaIndiansDiabetes[-idx.0,]

# re-scale data
data.pima$diabetes <- as.numeric(data.pima$diabetes)-1
maxs <- apply(data.pima, 2, max)
mins <- apply(data.pima, 2, min)
data.pima <- as.data.frame(scale(data.pima, center=mins, scale=maxs - mins))
head(data.pima)
##     pregnant   glucose  pressure   triceps   insulin      mass   pedigree
## 1 0.35294118 0.7437186 0.5901639 0.3535354 0.0000000 0.3149284 0.23441503
## 2 0.05882353 0.4271357 0.5409836 0.2929293 0.0000000 0.1717791 0.11656704
## 3 0.47058824 0.9195980 0.5245902 0.0000000 0.0000000 0.1042945 0.25362938
## 4 0.05882353 0.4472362 0.5409836 0.2323232 0.1111111 0.2024540 0.03800171
## 5 0.00000000 0.6884422 0.3278689 0.3535354 0.1985816 0.5092025 0.94363792
## 6 0.29411765 0.5829146 0.6065574 0.0000000 0.0000000 0.1513292 0.05251921
##         age diabetes
## 1 0.4833333        1
## 2 0.1666667        0
## 3 0.1833333        1
## 4 0.0000000        0
## 5 0.2000000        1
## 6 0.1500000        0

# split into train and test
n <- nrow(data.pima)
idx.train <- sample(c(1:n), round(n/2))
idx.test <- setdiff(c(1:n), idx.train)
data.train <- data.pima[idx.train, ]
data.test <- data.pima[idx.test,]

# fit multiple regression model to predict mass (BMI) based on other measurements
reg <- lm(mass ~ age + triceps + factor(diabetes) + insulin + glucose + pregnant + pressure, data
summary(reg)
##
## Call:
## lm(formula = mass ~ age + triceps + factor(diabetes) + insulin +
##     glucose + pregnant + pressure, data = data.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27091 -0.09095 -0.01383  0.07434  0.49669
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         0.10099    0.03375   2.992  0.00296 **
## age                -0.02083    0.04168  -0.500  0.61746
## triceps             0.30591    0.04332   7.062 8.20e-12 ***
## factor(diabetes)1   0.09178    0.01520   6.037 3.82e-09 ***
## insulin            -0.09777    0.06031  -1.621  0.10585
## glucose             0.02333    0.04736   0.493  0.62264
## pregnant           -0.06298    0.03806  -1.655  0.09883 .
## pressure            0.18860    0.04577   4.120 4.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1219 on 370 degrees of freedom
## Multiple R-squared:  0.2906,        Adjusted R-squared:  0.2772
## F-statistic: 21.65 on 7 and 370 DF,  p-value: < 2.2e-16

# calculate sum of squared errors (SSE)
yhat.reg1 <- predict(reg, newdata = data.test)
reg.sse <- data.frame(sse.train = sum((reg$fitted.values - data.train$mass)^2)/2,
                      sse.test = sum((yhat.reg1 - data.test$mass)^2)/2)

# fit ANN regression model
# notice linear.output set to TRUE this time
ann.r1 <- neuralnet(mass ~  age + triceps  + insulin + glucose + pregnant + pressure,
                        data = data.train,
                        hidden = c(5),
                        linear.output = TRUE)

plotnet(ann.r1)
```

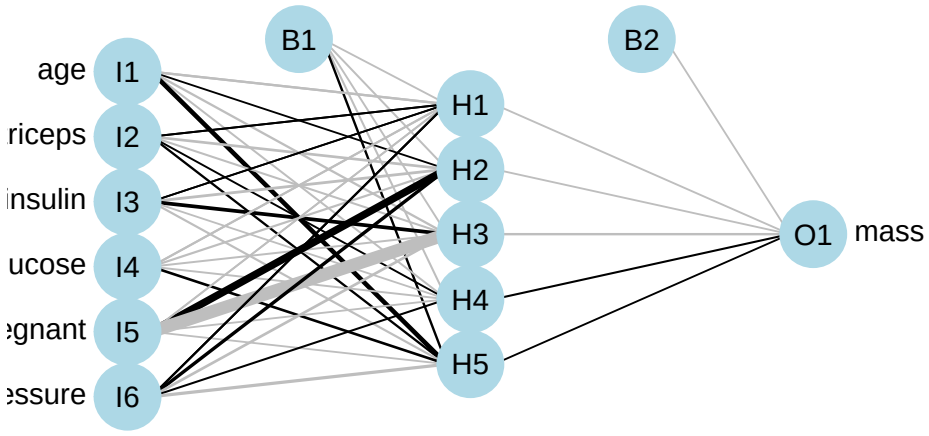


Figure 13.3: ANN network visualisation for regression

```
# SSE and RMSE errors
yhat.annr1 <- neuralnet::compute(ann.r1, data.test)$net.result
annr1.sse <- data.frame(sse.train = sum((ann.r1$net.result[[1]] - data.train$mass)^2)/2,
                        sse.test = sum((yhat.annr1 - data.train$mass)^2)/2)
## Warning in yhat.annr1 - data.train$mass: longer object length is not a multiple
## of shorter object length

# compare SSE errors for train and test between multiple regression and ANN
print(reg.sse)
##   sse.train sse.test
## 1  2.751006  3.04521
print(annr1.sse)
##   sse.train sse.test
## 1  2.109606 137.7486

# plot predicted values vs. mass values for the train and test data separately
par(mfrow=c(1,2))
plot(data.train$mass, reg$fitted.values, pch=19, col="lightblue", xlab="mass (true value)", ylab=
points(data.train$mass, ann.r1$net.result[[1]], pch=19, col="coral2")
lines(x=c(-100:100), y=c(-100:100))

plot(data.test$mass, yhat.reg1, pch=19, col="lightblue", xlab="mass (true value)", ylab="Predicti
points(data.test$mass, yhat.annr1, pch=19, col="coral2")
lines(x=c(-100:100), y=c(-100:100))
```

13.1.10 ANN rep value

Have we said at the lecture that the weights are randomly assigned at the start? How do we know that we have a best network? Above we have looked at one network and used a random seed to ensure reproducibility of the results. We could have been just unlucky with our comparisons to logigits or linear regression. In practice, one would fit many networks and choose a best one

```
# Here we will set rep to 5, it is getting computationally heavy
set.seed(1)
ann.r2 <- neuralnet(mass ~  age + triceps  + insulin + glucose + pregnant + pressure,
                        data = data.train,
                        hidden = c(5),
                        linear.output = TRUE,
                        rep = 5)

#plot(ann.r2, rep = "best")
```

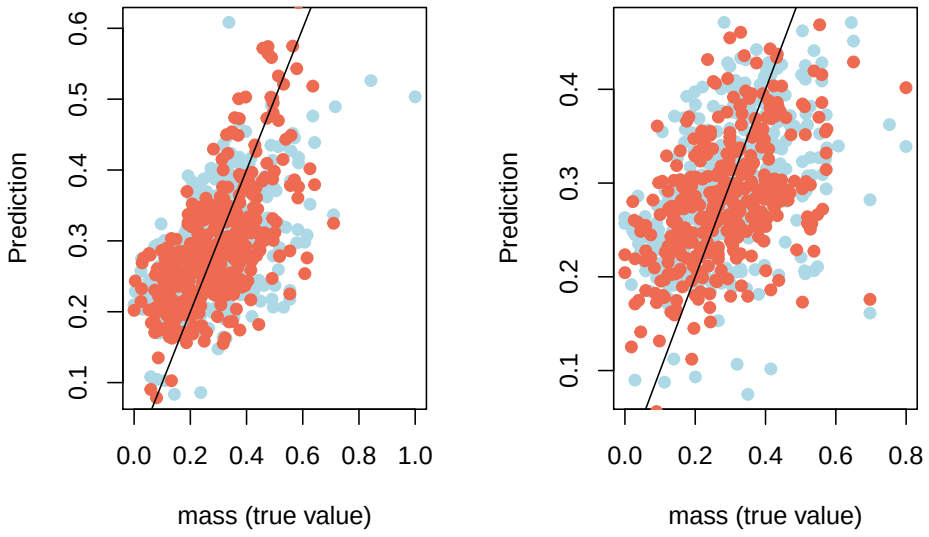


Figure 13.4: Comparison of predicted values vs. known values with linear regression (blue) and ANN (red) on the training data set (left) and on the test data set (right)

13.1.11 ANN relative importance of variables

- Weights that connect nodes in a neural network cannot be interpreted as the parameters coefficients of a standard linear regression model
- They can be thought of as analogous to them and can be used to describe relationships between variables
- i.e. weights dictate the relative influence of information that is processed in the network such that input variables that are not relevant in terms of their correlation with the response are suppressed by the weights
- A difference between neural network and a regression model is that the number of weights is excessive in the former case
- This makes neural network powerful and flexible, the price for that is easiness of integration of the model, it is not so straightforward anymore
- One method to identify the relative importance of the covariates is by Garson (1991) and is based by deconstructing the model weights (implented in NerualNetTools)

```
garson(ann.r1)
```

Feel free to experiment with different options of the neuralnet() and different datasets that we have used so far. Do note however, that neurlanet() package is great fpr getting some ideas about running ANN, the heavy duty work is done on GPUs typically via Keras, often in Python, although now Tensforlow is available for R as well.

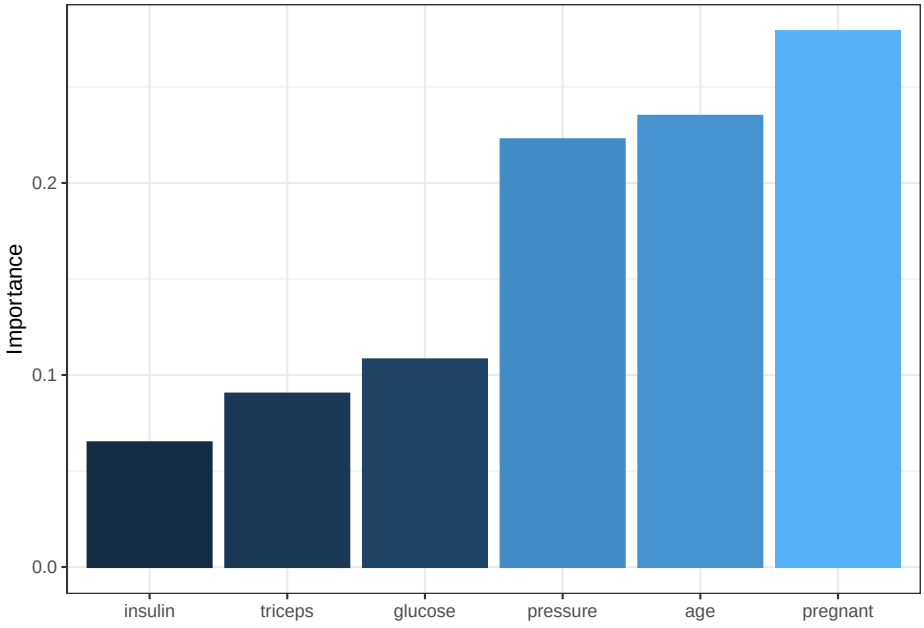


Figure 13.5: Output of the variable important call for one of the ANN regression models