Notes for the BAN400 Exam

Table of contents

1 Functions **1**
1.1 Basic functions 1
1.2 Math 1
1.3 Reading data 1

2 Topics **2**
2.1 Loops and iterations 2
2.2 Plotting 2

1 Functions

1.1 Basic functions

Function	Package	Description
mean()	base	Calculates the mean of a vector of numbers
median()	base	Calculates the median of a vector of numbers
sd()	base	Calculates the standard deviation of a vector of numbers
var()	base	Calculates the variance of a vector of numbers
sum()	base	Calculates the sum of a vector of numbers
c()	base	Creates vector
length()	base	The number of elements in a vector or list
ncol()	base	Number of columns of data frame or matrix
nrow()	base	Number of rows of data frame or matrix
min()	base	The smallest value in a set
max()	base	The largest value in a set

1.2 Math

Function	Package	Description
sqrt()	base	Calculates square root of number or vector of numbers
abs()	base	Calculates absolute value of number or vector of numbers

1.3 Reading data

Function	Package	Description
read_delim()	readr	Read file with columns separated by any delimiter
read_csv()	readr	Read csv-file (comma separated values)
read_excel()	readxl	Read data from excel files

2 Topics

2.1 Loops and iterations

2.1.1 Standard for-loop

```
for(i in 1:n) {
  ... do something with i...
}
```

Note that we can iterate over any type of vector, not just numbers, and we can give the iteration variable any name we want. In the example above it is `i`.

2.1.2 While loop

Repeat until a certain condition is met. For example

```
i <- 1
while(i < 10) {
  print(i)
  i <- i + 1
}
```

2.2 Plotting

We use `ggplot2` as the standard package for plotting, and the main function is `ggplot`. We supply a data frame to the first argument and an aesthetic mapping to the second argument. We add *layers* of plotting components using the plus sign. A simple example:

```
ggplot(df, aes(x = x_variable, y = y_variable, colour = grouping_variable)) +
  geom_point()
```

Many types of layers may contain other data sets via the `data` argument and/or updated aesthetic mappings via the `mapping` argument. Data and mappings are typically inherited from the layer above if not specified in a new layer. There are many types of functions for making further adjustments to labels, titles, axes and other properties. A more complete example may look like this:

```
ggplot(df, aes(x = x_variable, y = y_variable)) +
  geom_point() +
  geom_point(mapping = aes(x = new_x_variable, y = new_y_variable, colour = grouping_var),
             data = new_df) +
  xlab("X label") +
  ylab("Y label") +
  ggtitle("Title of plot") +
  theme_minimal()
```