

# Notes for the BAN400 Exam

## Table of contents

<b>1 Functions</b>	<b>1</b>
1.1 Basic functions . . . . .	1
1.2 Math . . . . .	1
1.3 Reading data . . . . .	1
<b>2 Topics</b>	<b>1</b>
2.1 Loops and iterations . . . . .	1
2.2 Plotting . . . . .	2

## 1 Functions

### 1.1 Basic functions

Function	Package	Description
c()	base	Creates vector
mean()	base	Calculates the mean of a vector of numbers
median()	base	Calculates the median of a vector of numbers
ncol()	base	Number of columns of data frame or matrix
nrow()	base	Number of rows of data frame of matrix
sd()	base	Calculates the standard deviation of a vector of numbers
var()	base	Calculates the variance of a vector of numbers

### 1.2 Math

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

### 1.3 Reading data

Function	Package	Description
read_csv()	readr	Read csv-file (comma separated values)
read_delim()	readr	Read file with columns separated by any delimiter
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()
```