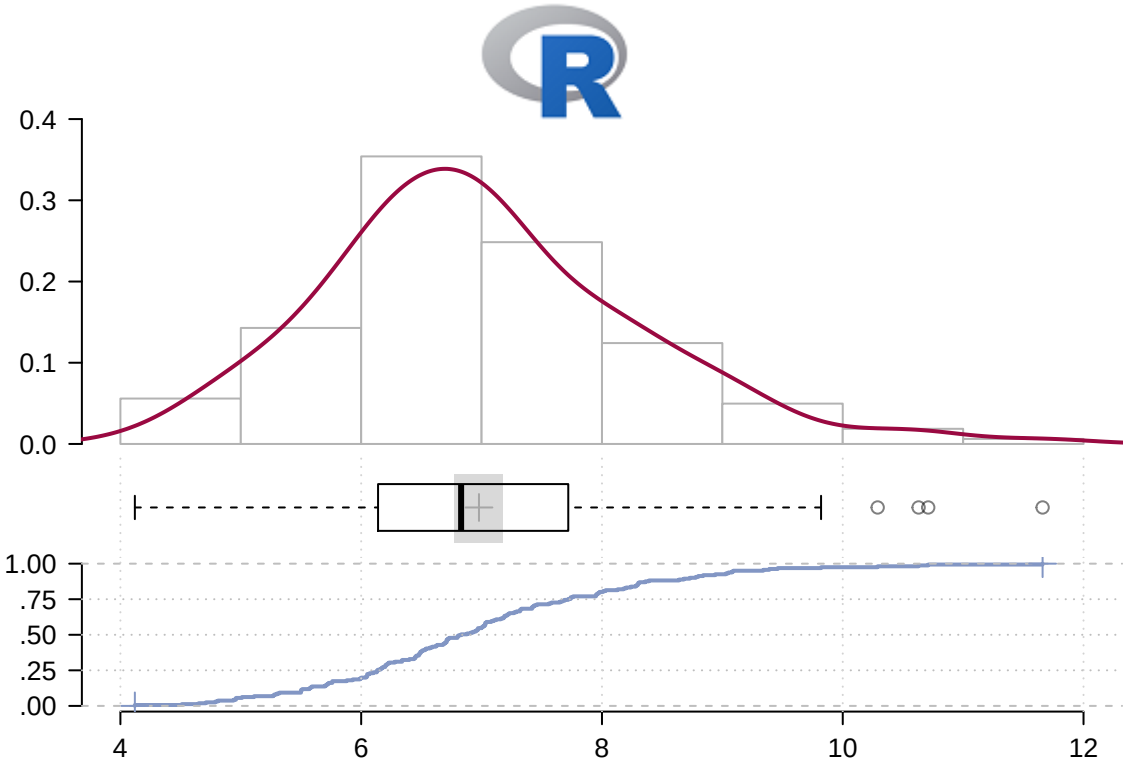
Applied Statistics using R

FACTSHEET

UNIT 1



Medical Statistics Team

University of Aberdeen

Copyright Notice

Contents

Chapter 1

Read Example data file

To illustrate the different features of R, we will first open an existing data file. If you are not familiar with reading CSV data into the R environment, check the R manual.

The dataset is called `cardiacdata.csv`. It is a comma-separated value file which we will open from RStudio environment. Please go to MyAberdeen and on the 'Materials and Activities' page, you will find the folder R datasets. The folder contains several data files (all are in .csv format); you might like to download them all for use later in the course. Note that you can also open the SPSS datasets or Excel datasets from RStudio environment.

The data contained within `cardiacdata.csv` comes from a cohort study looking at risk factors for cardiovascular disease. Numerous measurements were made on subjects at a baseline examination and a later follow-up examination. Subjects were then followed up for 10 years for mortality. (Note that some of the variables have also been categorised into two groups to enable you to perform relevant statistical tests later in the course).

The variables are as follows:

Variable	Explanation
`patno`	Unique patient ID
`age`	Age (years)
`sex`	Sex (1=Female, 2=Male)
`systolic`	Systolic blood pressure (mmHg)
`diastolic`	Diastolic blood pressure (mmHg)
`tchol`	Total cholesterol (mmol/l)
`hdlchol`	HDL cholesterol (mmol/l)
`triglyceride`	Triglyceride (mmol/l)
`socialclass`	Social class (1=I, 2=II, 3=IIIM, 4=IIIN, 5=IV, 6=V, 7=VI)
`soccl2`	Social class (2 categories) (1=High, 2=Low)
`bmi`	Body mass index (kg/m2)
`alcohol`	Alcohol intake in previous week (units)
`leisact`	Leisure activity level (0=No activity, 1=At most light, 2=At most moderate, 3=At most strenuous)
`activgr`	Leisure activity (2 categories) (1=No or light activity, 2=Moderate or strenuous activity)

Variable	Explanation
education	Highest level education attained (1=University, 2=Post school training, 3=Secondary, 4=Primary)
diabetic	Diabetic status (1=Normal, 2=Impaired glucose tolerant, 3=Diabetic)
smoking	Smoking status (1=Current, 2=Ex, 3=Never)
smokegr	Smoking status (2 categories) (1=Yes, 2=No)
death	Died during follow-up (0=Still alive, 1=Died)
causedeath	Cause of death (0=Still alive, 1=Definite MI, 2=Possible MI, 3=Sudden death, 4=Definite stroke, 5=Possible stroke, 6=Other CV, 7=Other non-CV)
systolic10	Follow-up systolic blood pressure
bmi10	Follow-up body mass index (kg/m2)
smokegr10	Follow-up smoking status (2 categories) (1=Yes, 2=No)
activgr10	Follow-up leisure activity (2 categories) (1=No or light activity, 2=Moderate or strenuous activity)
family	Family history of myocardial infarction (2 categories) (0 = no, 1 =yes)
maritalstatus	never married, 2= married, 3=divorced

1.1 Steps

- **Step 1**: Set your working directory to the folder where the data file is located:

```
setwd("C:/R/Example")
```

For example, the above code indicates that the data file `cardiacdata.csv` is in the folder `C:/R/Example/`.

There are several options to set the working directory from the RStudio environment. For example, if you have written an R script, you can set the working directory on the same folder using the following option from RStudio menu: `Session > Set Working Directory > To Source File Location`. There are other options to choose the directory.

Check the current working directory:

```
getwd()
```

- **Step 2**: For the CSV file, use the function `read.csv` to read the file `cardiacdata.csv` and assign it to an R object `DF` (or any valid R object names). Note the command expects that the first row contains the variable names. You can also use the general function `read.table` indicating the separation of variables as `,`.

```
DF <- read.csv("cardiacdata.csv")
```

Note that by default `read.csv` function considers that the missing data are coded as `NA`. If possible, do not code the missing data using numeric value although R can consider any other values or texts to assign as missing. The R object `DF` is called a **data.frame** object.

The data.frame is like an Excel spreadsheet with rows and columns. The observations are in rows while variables are in columns.

Check the help file for `read.csv` for further details.

You can also use RStudio environment directly to read the data onto the R environment. For example, in `RStudio` the steps are: File > Import Dataset > Select an appropriate file type.

1.2 Recode factor variables

Since the categorical variables in this data are recorded as integer, the data.frame `DF` will recognise those variables as integer variables. Hence we should convert those variables to nominal or ordinal factor (categorical variables) using the function `as.factor` or recode the variables using the function `car::recode` from the `library(car)`. The `levels` argument keeps the order of the levels identical to the orginal (otherwise the levels will be set alphabetically).

```
library(car)


DF$patno <- as.factor(DF$patno)

DF$sex <- car::recode(DF$sex, recodes = " '1'='Female'; '2'='Male' ",
                as.factor = TRUE, levels = c('Female','Male'))

DF$socialclass <- as.ordered(DF$socialclass)

DF$soccl2 <- car::recode(DF$soccl2, recodes = " '1'='High'; '2'='Low' ",
                as.factor = TRUE, levels = c('High','Low'))

DF$leisact <- car::recode(DF$leisact,
                recodes = " '0'='No'; '1'='Light';
                            '2'='Moderate'; '3'='Strenuous' ",
                as.factor = TRUE,
                levels = c('No','Light','Moderate','Strenuous'))

DF$activgr <- car::recode(DF$activgr,
                recodes = " '1'='No_Light'; '2'='Moderate_Strenuous' ",
                as.factor = TRUE, levels = c('No_Light','Moderate_Strenuous'))

DF$education <- car::recode(DF$education,
                recodes = " '1'='University'; '2'='Post_school';
                            '3'='Secondary'; '4'='Primary' ",
                as.factor = TRUE,
                levels = c('University','Post_school',
                           'Secondary', 'Primary'))

DF$diabetic <- car::recode(DF$diabetic,
               recodes = " '1'='Normal'; '2'='Imp_Glu_tol'; '3'='Diabetic' ",
               as.factor = TRUE,
```

```
                levels = c('Normal','Imp_Glu_tol','Diabetic'))


DF$smoking <- car::recode(DF$smoking,
                recodes = " '1'='Current'; '2'='Ex'; '3'='Never' ",
                as.factor = TRUE, levels = c('Current','Ex','Never'))


DF$smokegr <- car::recode(DF$smokegr, recodes = " '1'='Yes'; '2'='No' ",
                as.factor = TRUE, levels = c('Yes','No'))


DF$death <- car::recode(DF$death, recodes = " '0'='Alive'; '1'='Died' ",
                        as.factor = TRUE, levels = c('Alive','Died'))


DF$causedeath <- car::recode(DF$causedeath,
                recodes = " '0'='Alive'; '1'='Def_MI';
                            '2'='Pos_MI'; '3'='Sudden_death';
                            '4'='Def_Stroke'; '5'='Pos_Stroke';
                            '6'='Other_CV'; '7'='Other_NonCV' ",
                 as.factor = TRUE,
                 levels = c('Alive','Def_MI',
                            'Pos_MI','Sudden_death',
                            'Def_Stroke','Pos_Stroke',
                            'Other_CV', 'Other_NonCV'))


DF$smokegr10 <- car::recode(DF$smokegr10,
                  recodes = " '1'='Yes'; '2'='No' ",
                  as.factor = TRUE, levels = c('Yes','No'))

DF$activgr10 <- car::recode(DF$activgr10,
                  recodes = " '1'='No_Light'; '2'='Moderate_Strenuous' ",
                  as.factor = TRUE, levels = c('No_Light','Moderate_Strenuous'))

DF$family <- car::recode(DF$family,
                  recodes = " '0'='No'; '1'='Yes' ",
                  as.factor = TRUE, levels = c('No','Yes'))

DF$maritalstatus <- car::recode(DF$maritalstatus,
                  recodes = " '1'='Never_married'; '2'='Married'; '3'='Divorced' ",
                  as.factor = TRUE, levels = c('Never_married','Married','Divorced'))
```

Chapter 2

Data Types

2.1 What is a data type?

Data type describes what type of variable you have. They could be measured e.g. body weight or a category type variable e.g. gender.

2.2 Why is data type important?

Knowing the data type is critical to determine the best way to describe the variable and which will be the appropriate statistical test when you wish to compare groups in relation to a particular variable.

2.3 What are the different data types?

Here were use common terminology and provide some examples of different data types.

2.4 Numerical data

Numerical data can be one of two types:

- Continuous
  - A numerical data type, usually measured and decimals allowed
  - Examples include weight in kg, blood pressure in mmHg
- Discrete
  - A numerical data type with integer values
  - Examples include age in whole years, number of GP visits within one month, number of children in a household

2.5 Categorical data

Categorical data can be one of three types:

- Binary
  - A categorical variable which can only take one of two values
  - Examples include gender (male/female), disease (yes/no), status (alive/dead)
- Nominal
  - A categorical variable with more than two categories and no natural ordering
  - Examples include eye colour (blue/green/brown/other), marital status (single/married/widowed/other)
- Ordinal
  - A categorical variable that is ordered and has more than two categories
  - Examples include pain severity (mild/moderate/severe), age group (18, 19-30, 31-50, 51 years)

2.6 Summarising Data types

Please see other factsheets on how to summarise numerical and categorical variables.

2.7 Using R

2.7.1 Read the data

Read the data `cardiacdata.csv` into the R environment and assigned it the name `DF`. Set your working directory and put the data in the working directory or give the correct path.

```
DF <- read.csv('cardiacdata.csv')
```

2.7.2 Check the data.frame structure

Use the function `str` to identify the types of all variables. Check all variables carefully and confirm that you have defined all variables correctly.

```
str(DF)
```

```
'data.frame':   163 obs. of  26 variables:
 $ patno        : Factor w/ 163 levels "0049B","0052H",..: 6 54 65 69 101 115 116 2 3 4 ..
 $ age          : num  71 68.2 62.9 69.9 65 ...
 $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 1 1 2 ...
 $ systolic     : int  142 140 156 154 187 123 147 140 131 128 ...
 $ diastolic    : int  63 78 82 102 131 63 66 67 73 62 ...
 $ tchol        : num  7.03 5.32 9.34 7.19 8.84 6.17 6.2 6.96 7.02 7.21 ...
 $ hdlchol      : num  1.4 0.88 0.92 1.31 1.83 1.58 1.65 1.68 1.57 1.08 ...
 $ triglyceride : num  0.81 3.4 4.67 2.53 1.76 0.73 1.11 0.69 1.29 1.16 ...
 $ socialclass  : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 2 4 2 3 2 3 3 2 1 2 ...
 $ soccl2       : Factor w/ 2 levels "High","Low": 1 2 1 2 1 2 2 1 1 1 ...
 $ bmi          : num  24.7 26 26.6 26.2 26.1 ...
 $ alcohol      : int  9 2 7 24 14 8 0 2 0 0 ...
 $ leisact      : Factor w/ 4 levels "No","Light","Moderate",..: 3 2 3 3 3 2 3 3 3 4 ...
 $ activgr      : Factor w/ 2 levels "No_Light","Moderate_Strenuous": 2 1 2 2 2 1 2 2 2 2
 $ education    : Factor w/ 4 levels "University","Post_school",..: 1 3 1 3 3 3 3 1 1 1 ..
 $ diabetic     : Factor w/ 3 levels "Normal","Imp_Glu_tol",..: 1 1 3 1 1 1 1 1 1 1 ...
 $ smoking      : Factor w/ 3 levels "Current","Ex",..: NA NA NA NA NA NA NA 1 1 1 ...
 $ smokegr      : Factor w/ 2 levels "Yes","No": NA NA NA NA NA NA NA 1 1 1 ...
 $ death        : Factor w/ 2 levels "Alive","Died": 1 2 2 1 1 1 2 2 1 2 ...
 $ causedeath   : Factor w/ 8 levels "Alive","Def_MI",..: 1 8 3 1 1 1 8 2 5 3 ...
 $ systolic10   : int  142 NA NA 159 189 122 NA NA 132 NA ...
 $ bmi10        : num  20 NA NA 25.1 25 ...
 $ smokegr10    : Factor w/ 2 levels "Yes","No": NA NA NA NA NA NA NA NA 2 NA ...
 $ activgr10    : Factor w/ 2 levels "No_Light","Moderate_Strenuous": 2 NA NA 2 2 1 NA NA
 $ family       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ maritalstatus: Factor w/ 3 levels "Never_married",..: 2 3 1 2 2 2 2 3 1 1 ...
```

Chapter 3

Summarise Single Numeric Variable

3.1 Data type

3.1.1 Continuous:

- A numerical data type, usually measured and decimals allowed
- Examples include weight in kg, blood pressure in mmHg

3.1.2 Discrete

- A numerical data type with integer values
- Examples include age in whole years, number of GP visits within one month, number of children in a household

3.2 Summary statistics

Definition of appropriate summary statistics

The quantities you may need are mean, standard deviation, median and interquartile range with the choice dependent on the distribution of the data.

3.3 Mean

Mean: sum of all observations divided by the number of observations

- Mean or Average or Arithmetic Mean (AM)

$$\bar{x} = (x_1 + x_2 + ... + x_n)/n$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

3.4 Median

Median: central value of the distribution – if you order the observations from smallest to largest then the median value is the value for which 50% of the observations fall below.

3.5 Variance

$$Var(x) = s_x^2 = \frac{1}{n}[(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2]$$

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

3.6 Standard Deviation

Standard deviation: the square root of the 'sum of squared distances of each observation from the mean, divided by n-1

$$s_x = \sqrt{s_x^2}$$

- Two Definitions of Variance
  - **Population Variance**
  - **Sample Variance**

Population Variance

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

Sample Variance

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

3.7 Inter-quartile range

Interquartile range: given by the 25th and 75th percentile and always presented alongside a median

3.8 Using summary statistic

When do I use each summary statistic?

- Normally distributed data you present the mean and standard deviation
- Skewed data present the median and interquartile range

Why

this is because skewed data may have unusually high or low values and the median and interquartile range are less affected by outliers and thus will give a more accurate summary of the data.

3.9 Using R

3.10 Summary statistics in R

Read the data as a data.frame in the R environment and depending on the distribution select the appropriate summary statistics from the output.

Although there are several functions to obtain individual summary statistics for a numerical variable, we will use an R package `DescTools` which provides an array of summary statistics as well as summary plot.

Here we will compute summary statistics of the numeric variable `tchol`.

3.10.1 Load `library(DescTools)`

Install the library `DescTools` using the command: `install.package(DescTools)`

You have to install it once. Once installed, load the library for every R session.

```
library(DescTools)
```

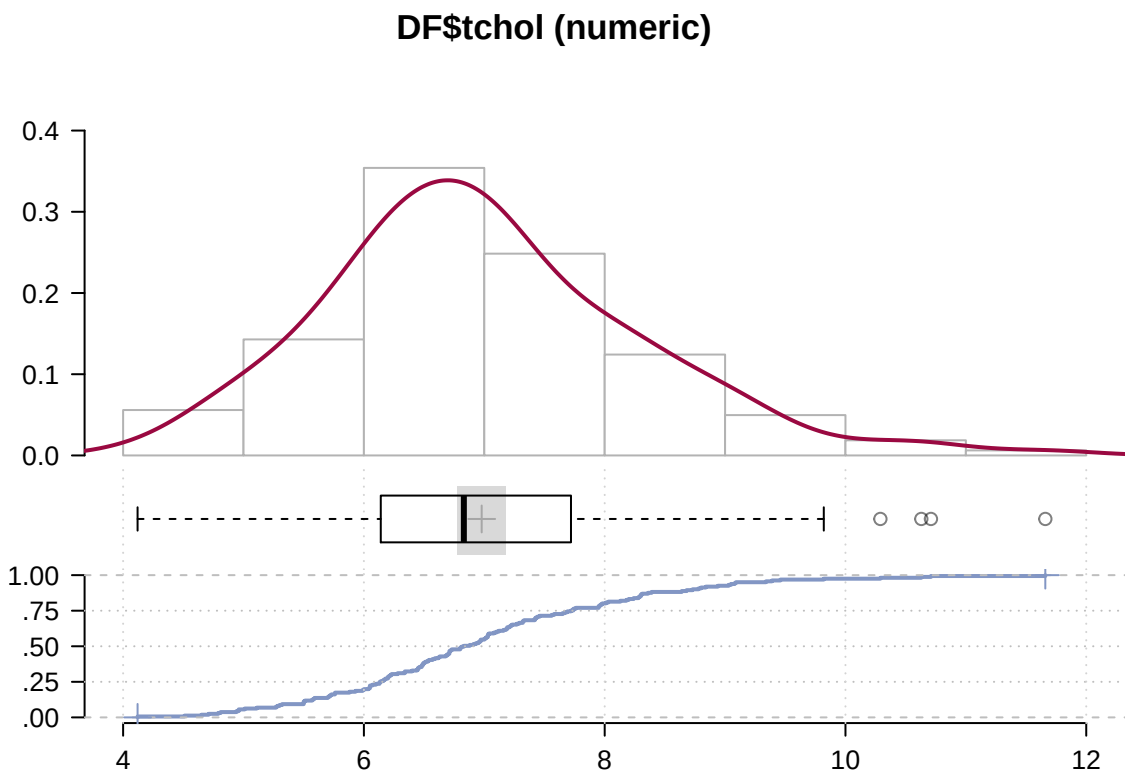
3.10.2 Summary statistics & plots

Call the function `Desc` with argument `plotit = TRUE` to compute summary statistics as well as plots. Note sometime to clarify that the function belongs to the `DescTools` package, the function is called as `DescTools::Desc`. Call the function with argument `plotit = FALSE` to compute summary statistics only.

The function displays several summary statistics and all may not be equally useful and informative.

```
DescTools::Desc(DF$tchol, plotit = TRUE)
```

```
------------------------------------------------------------------------------
DF$tchol (numeric)

  length       n     NAs  unique      0s  mean  meanCI
     163     161       2     137       0  6.98    6.78
           98.8%   1.2%           0.0%          7.18

     .05     .10     .25  median    .75   .90     .95
    4.96    5.50    6.14    6.83   7.72  8.73    9.09

   range      sd  vcoef     mad     IQR  skew    kurt
    7.54    1.30   0.19    1.13    1.58  0.64    0.74

lowest : 4.12, 4.51, 4.65, 4.71, 4.79
highest: 9.82, 10.29, 10.63, 10.71, 11.66
```



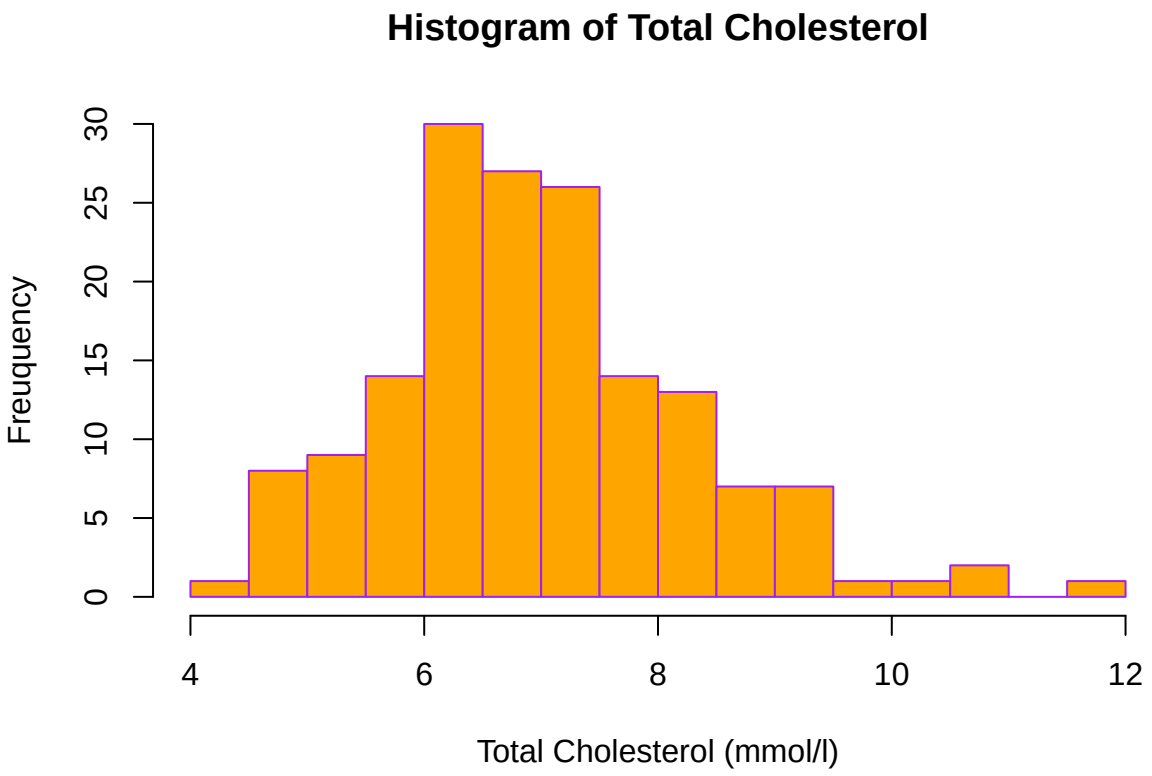
You can obtain individual summary statistics and plots for the variable using individual functions from the standard `base` R (you do not need to install any package) that we discussed in the R

manual section. Following example provides a few examples; these do not provide all outputs as above.

```
# Summary statistics

sum(!is.na(DF$tchol))

sum(is.na(DF$tchol))

min(DF$tchol, na.rm = TRUE)

max(DF$tchol, na.rm = TRUE)

range(DF$tchol, na.rm = TRUE)

mean(DF$tchol, na.rm = TRUE)

median(DF$tchol, na.rm = TRUE)

IQR(DF$tchol, na.rm = TRUE)

var(DF$tchol, na.rm = TRUE)

sd(DF$tchol, na.rm = TRUE)
```

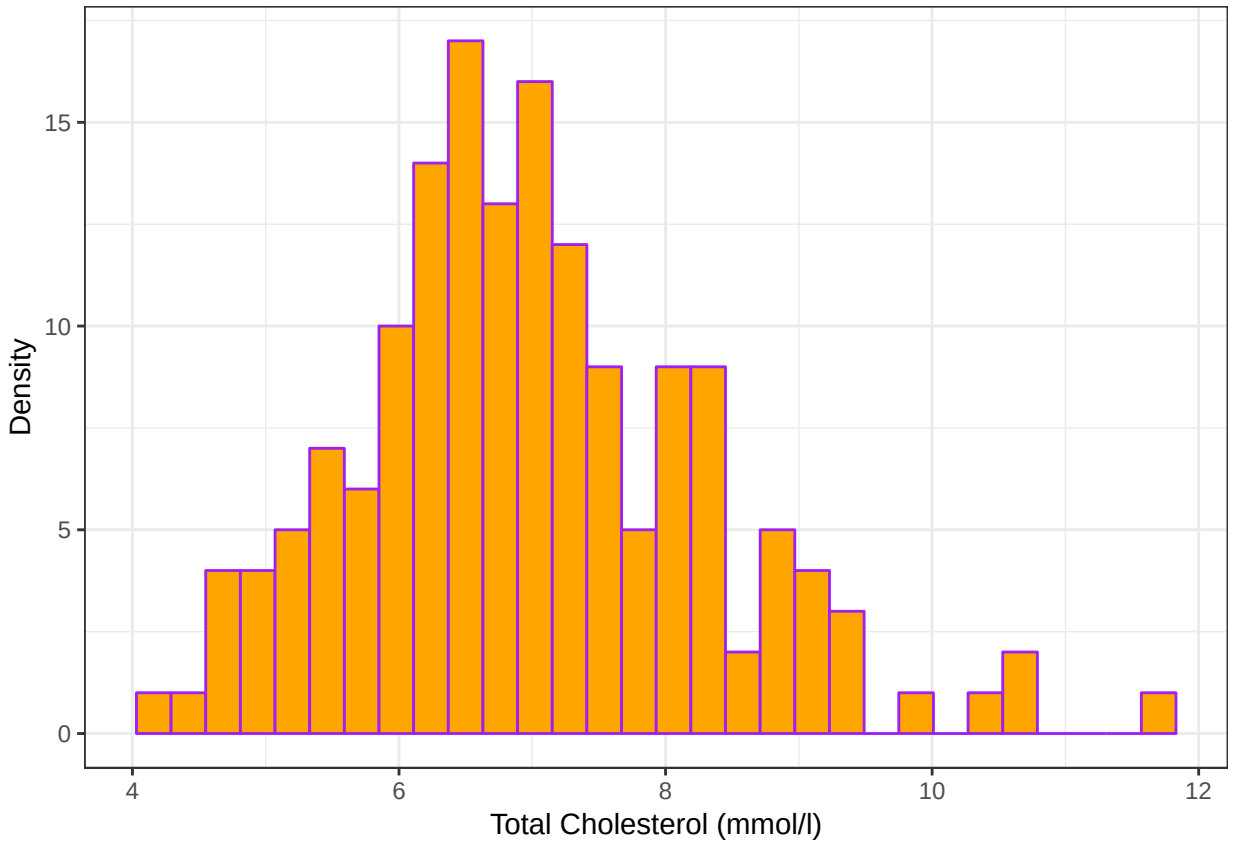
To plot a simple histogram using `base` R, run the following script.

```
# Histogram

hist(x = DF$tchol, breaks = 20, freq = TRUE,
     main = 'Histogram of Total Cholesterol',
     xlab = 'Total Cholesterol (mmol/l)',
     ylab = 'Freuquency',
     axes = TRUE, col = 'orange',
     lty = 1, border = 'purple')
```



Or, to plot a histogram using `library(ggplot2)` R, run the following script.

```
# Histogram

g <- ggplot(data = DF, mapping = aes(tchol))
g <- g + geom_histogram(fill = 'orange', colour = 'purple')
g <- g + labs(x = 'Total Cholesterol (mmol/l)',
              y = 'Density')
g <- g + theme_bw()

g
```



3.10.3 Summary of outputs

The variable total cholestrol consists of 161 observations with two missing values. There are 131 unique values. The estimate of mean (standard deviation) of total cholesteorl is 6.98 (1.30) mmol/l.

Note that if the variable is not normally distributed, appropriate statistics to quote will be median (Q1, Q3).

Chapter 4

Summarise Single Categorical Variable

4.1 Summary statistics for categorical data

The number and percentage in each category are appropriate.

4.2 Using R

4.3 Summary statistics in R

Read the data as a data.frame in the R environment.

Considering that the R package `DescTools` is loaded in the environment (if not loaded, run `library(DescTools)`), we can compute the summary statistics for the categorical variable.

Here we will compute summary statistics of the categorical variable `socialclass`.

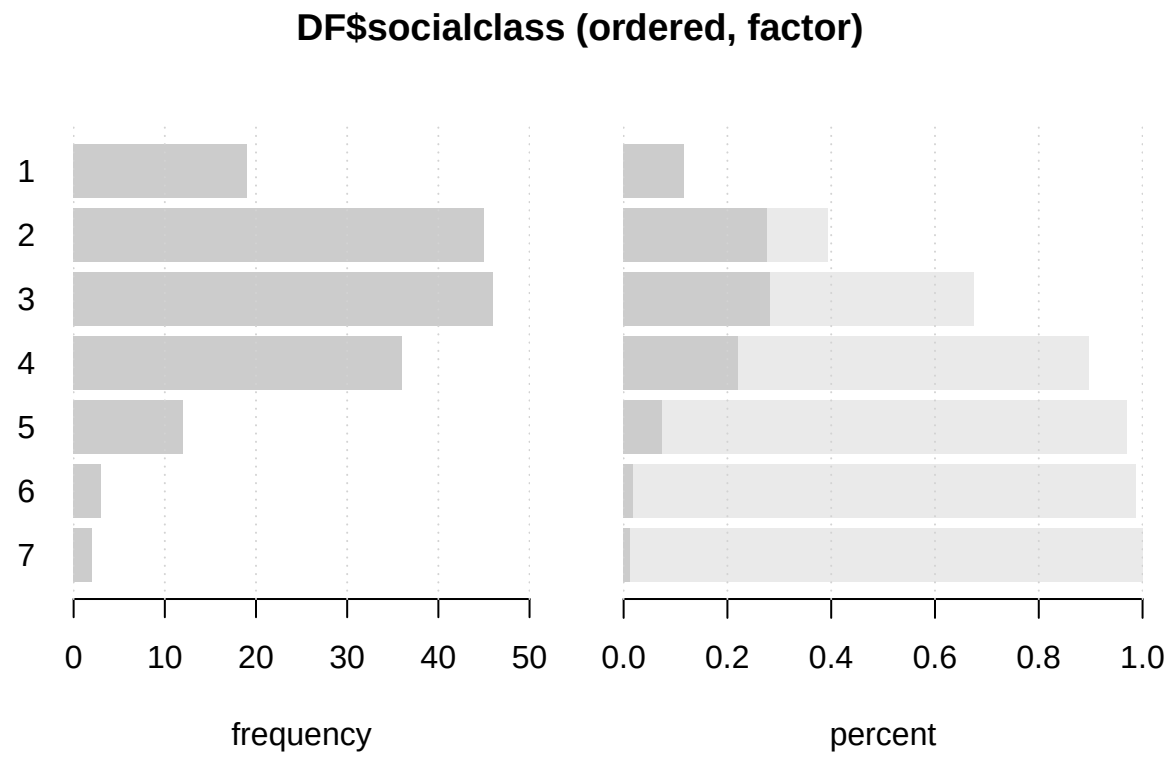
4.3.1 Summary statistics & plots

Call the function `Desc` with argument `plotit = TRUE` to compute summary statistics as well as plots. Note sometime to clarify that the function belongs to the `DescTools` package, the function is called as `DescTools::Desc`. Call the function with argument `plotit = FALSE` to compute summary statistics only.

```
DescTools::Desc(DF$socialclass, plotit = TRUE)
```

```
------------------------------------------------------------------------------
DF$socialclass (ordered, factor)

  length      n    NAs unique levels  dupes
     163    163      0      7      7      y
         100.0%   0.0%
```

	level	freq	perc	cumfreq	cumperc
1	1	19	11.7%	19	11.7%
2	2	45	27.6%	64	39.3%
3	3	46	28.2%	110	67.5%
4	4	36	22.1%	146	89.6%
5	5	12	7.4%	158	96.9%
6	6	3	1.8%	161	98.8%
7	7	2	1.2%	163	100.0%



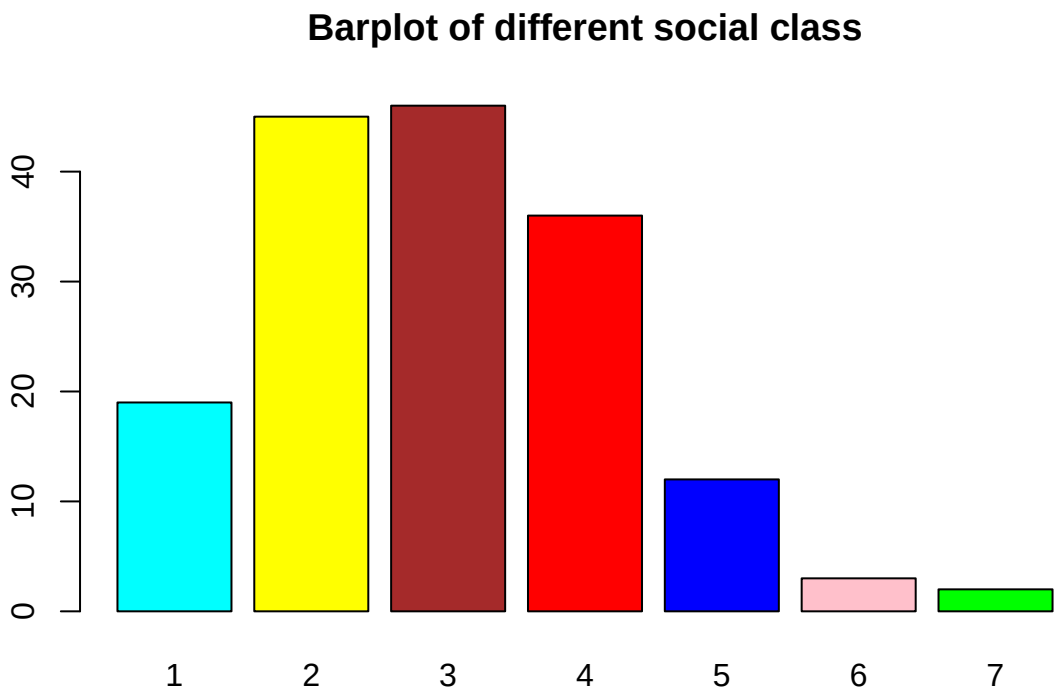
You can also obtain individual summary statistics for the variable using the `base` R function `table`. The function `prop.table` shows the overall proportion, and with `margin = 1` shows the proportion within rows and `margin = 2` shows the proportion within columns.

```
tab <- table(DF$socialclass)

# Overall proportion
prop.table(tab)

# Proportion within rows
prop.table(tab, margin = 1)
```

```
# Proportion within columns
prop.table(tab, margin = 2)
```

To plot a simple barplot using `base` R, use the function `barplot` on the table object.

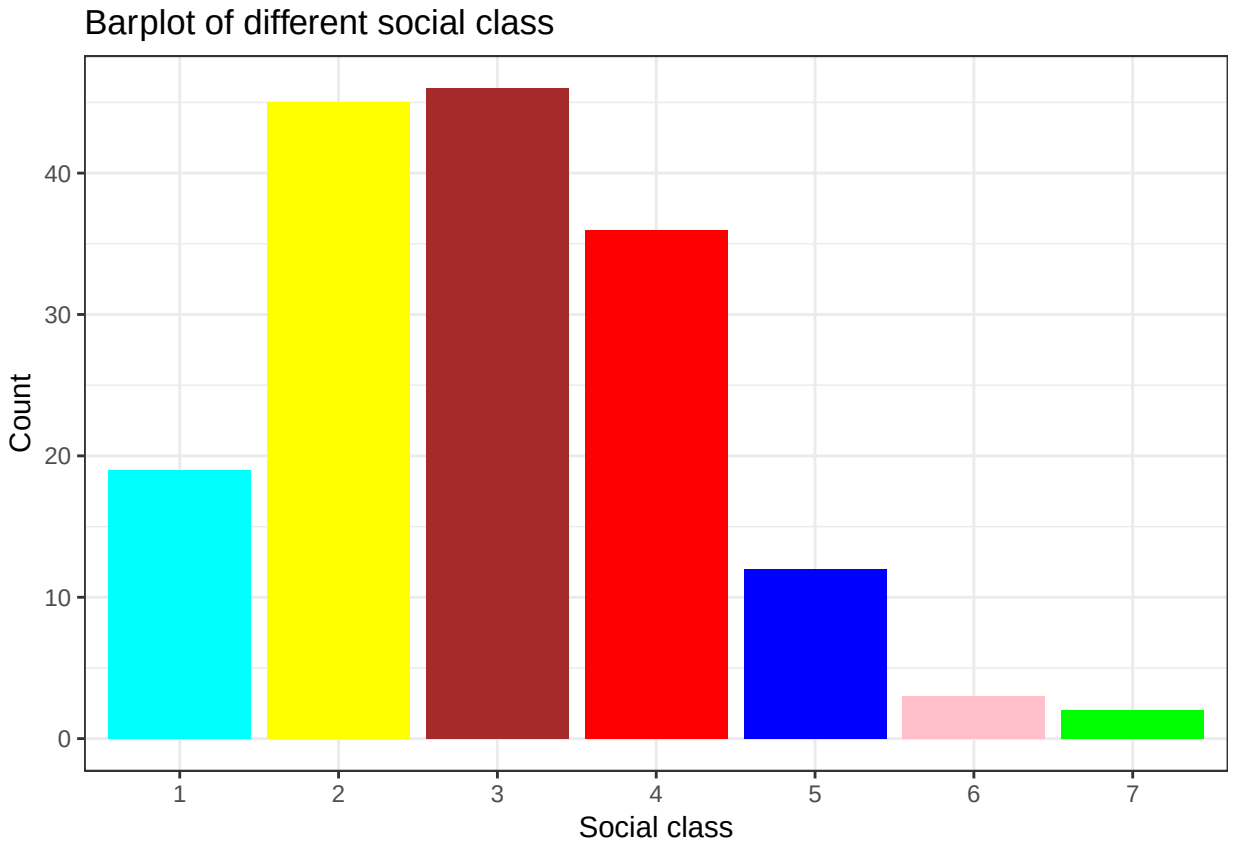
```
# Barplot

tab <- table(DF$socialclass)

barplot(tab,
    col=c('cyan', 'yellow', 'brown','red', 'blue', 'pink', 'green'),
    main='Barplot of different social class')
```



Or, to crate a barplot using `library(ggplot2)` R, run the following script.

```
# Barplot

g <- ggplot(data = DF, aes(x  =  socialclass))
g <- g + geom_bar(fill = c('cyan', 'yellow', 'brown','red', 'blue', 'pink', 'green'))
g <- g + labs(title = 'Barplot of different social class',
              x = 'Social class',
              y = 'Count')
```

```
g <- g + theme_bw()
g
```



4.3.2 Summary of outputs

There are 163 individuals with no missing data for the variable social class (`socialclass`). Of those with available data, 19 (11.7%), 45 (27.6%), 46 (28.2%) and 36 (22.1%) are in social class 1, 2, 3 and 4, respectively. The social class 7 has the smallest member i.e. 2 (1.2%).

Chapter 5

Summarise Numerical by Categorical Variable

Numerical by Categorical Variable

A reminder, for numerical variables if the data are:

- Normally distributed, then present the *mean* and *standard deviation*
- Skewed, then give the *median* and *interquartile range* i.e. *Q1* and *Q3*

5.1 Using R

5.2 Summary statistics in R

Read the data as a data.frame in the R environment and depending on the distribution select the appropriate summary statistics from the output.

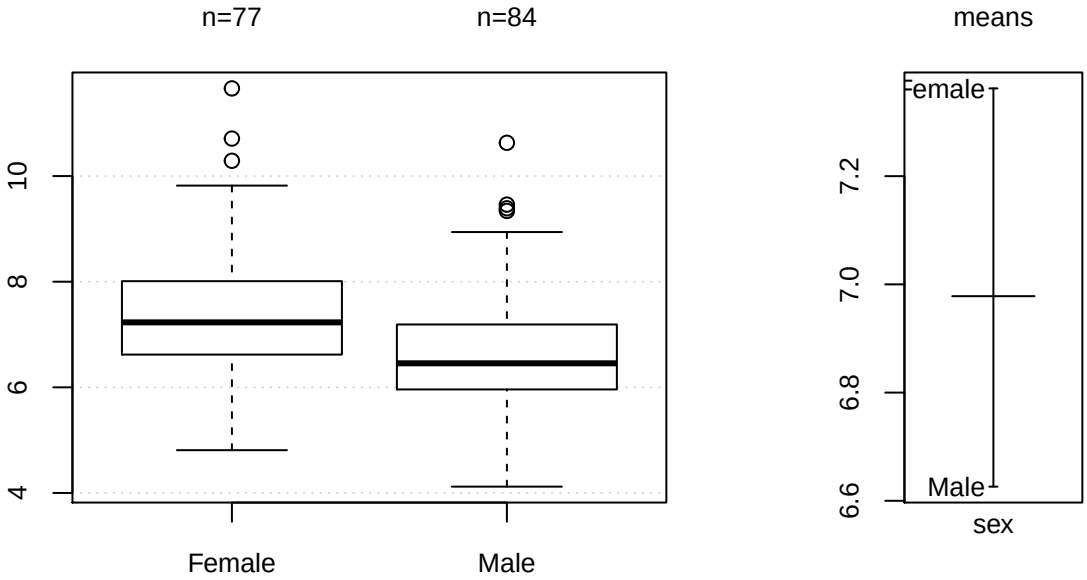
Considering that the R package `DescTools` is loaded in the environment (if not loaded, run `library(DescTools)`), we can compute the summary statistics for the categorical variable. Call the function `Desc` with argument `plotit = TRUE` to compute summary statistics as well as plots. Note sometime to clarify that the function belongs to the `DescTools` package, the function is called as `DescTools::Desc`. Call the function with argument `plotit = FALSE` to compute summary statistics only.

Here we will compute summary statistics of the continuous variable `tchol` by the categorical variable `sex`.

5.2.1 Summary statistics & plots

```
DescTools::Desc(tchol ~ sex, data = DF, plotit = TRUE)
```

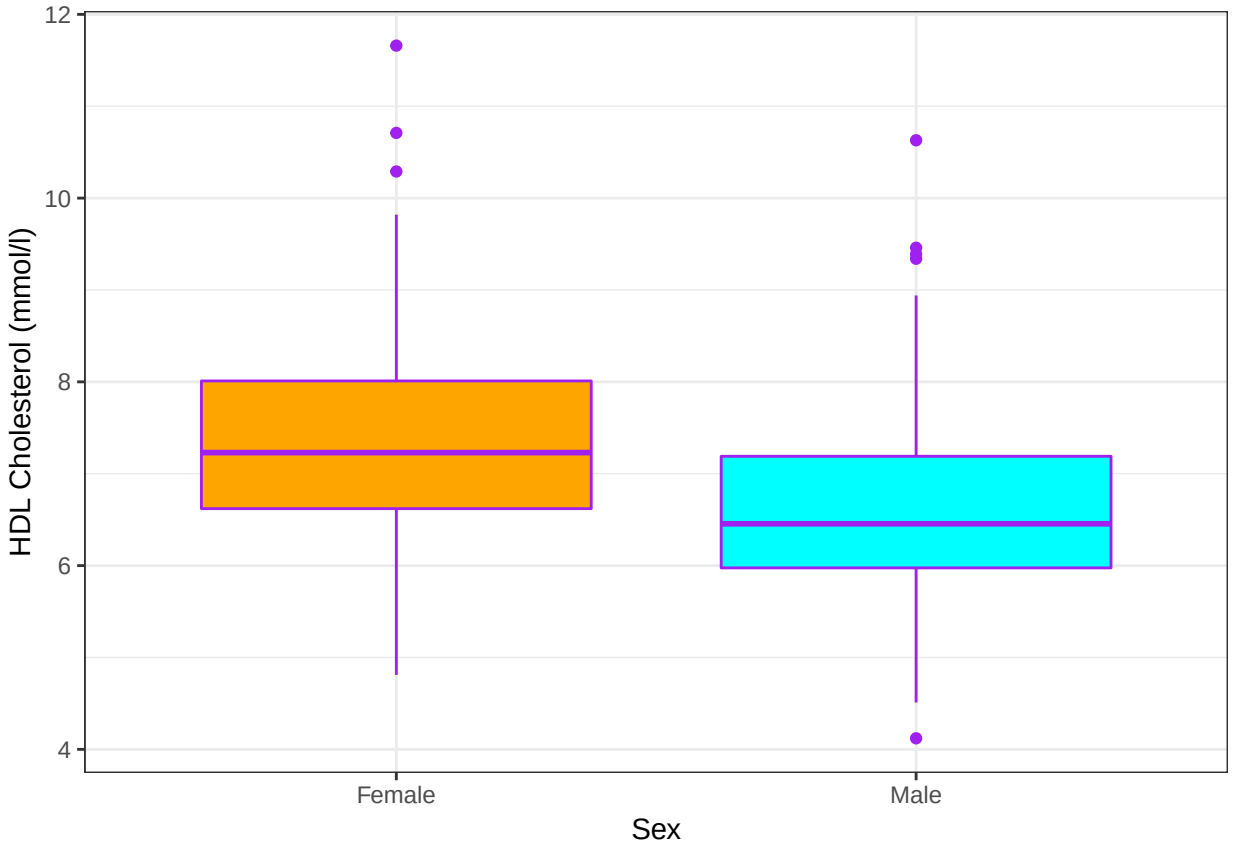
```
------------------------------------------------------------------------------
tchol ~ sex

Summary:
n pairs: 163, valid: 161 (98.8%), missings: 2 (1.2%), groups: 2


          Female     Male
mean       7.362    6.626
median     7.230    6.455
sd         1.257    1.238
IQR        1.390    1.215
n             77       84
np       47.826%  52.174%
NAs            1        1
0s             0        0

Kruskal-Wallis rank sum test:
  Kruskal-Wallis chi-squared = 15.934, df = 1, p-value = 6.561e-05
```



It is possible to use functions or a combination of functions (or write your own functions) in standard `base` R, to create the summary statistics for the numerical by categorical variables. We will, however, not discuss it here.

To create a boxplot using `library(ggplot2)` R, run the following script.

```
# Boxplot

g <- ggplot(data = DF, mapping = aes(x = sex, y = tchol))
g <- g + geom_boxplot(fill = c('orange','cyan'),
                      colour = 'purple',
                      linetype = 1, size = 0.5)
g <- g + labs(x = 'Sex', y = 'HDL Cholesterol (mmol/l)')
g <- g + theme_bw()
g
```



5.3 Summary of outputs

The estimate of mean (standard deviation) of total cholesteorl for females is 7.36 (1.26) mmol/l compared to males with mean 6.63 (1.24) mmol/l.

Chapter 6

Summarise Two Categorical Variables

6.1 Summary statistics for categorical data

To summarise a categorical variable, we need the number and percentage in each category.

6.2 Using R

6.3 Summary statistics in R

Read the data as a data.frame in the R environment.

Considering that the R package `DescTools` is loaded in the environment (if not loaded, run `library(DescTools)`), we can compute the summary statistics for the categorical variable.

Here we will compute summary statistics of the categorical variables `socialclass` and `sex`.

6.3.1 Summary statistics & plots

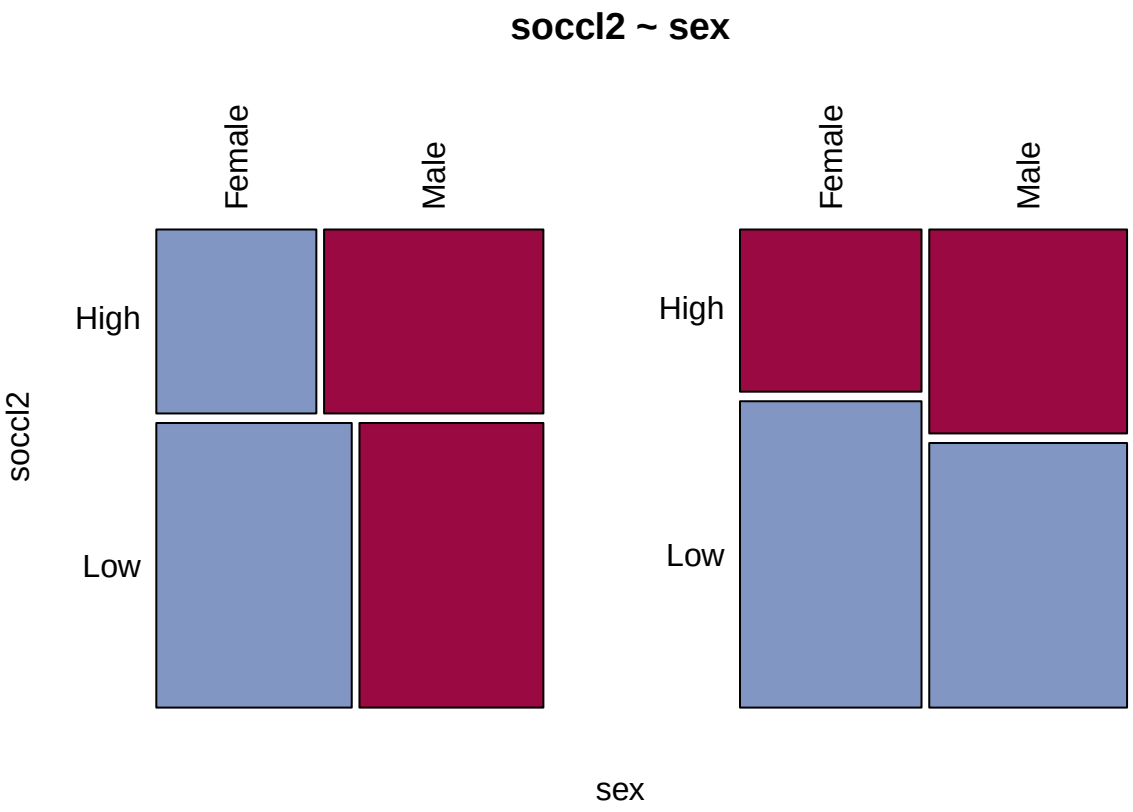
Call the function `Desc` with argument `plotit = TRUE` to compute summary statistics as well as plots. Call the function with argument `plotit = FALSE` to compute summary statistics only.

Note the outputs may include more information than required for a summary presentation.

```
DescTools::Desc(soccl2 ~ sex, data = DF, plotit = TRUE)
```

```
------------------------------------------------------------------------------
soccl2 ~ sex


Summary:
n: 163, rows: 2, columns: 2
```

```
Pearson's Chi-squared test (cont. adj):
  X-squared = 1.0073, df = 1, p-value = 0.3156
Fisher's exact test p-value = 0.2645
McNemar's chi-squared = 1.9205, df = 1, p-value = 0.1658

                    estimate lwr.ci upr.ci'

odds ratio             0.687  0.364  1.294
rel. risk (col1)       0.819  0.580  1.156
rel. risk (col2)       1.192  0.891  1.596


Phi-Coefficient        0.091
Contingency Coeff.     0.091
Cramer's V             0.091


         sex     Female   Male    Sum
soccl2

High     freq        27     37     64
         perc     16.6%  22.7%  39.3%
         p.row    42.2%  57.8%      .
         p.col    34.6%  43.5%      .

Low      freq        51     48     99
         perc     31.3%  29.4%  60.7%
         p.row    51.5%  48.5%      .
         p.col    65.4%  56.5%      .

Sum      freq        78     85    163
         perc     47.9%  52.1% 100.0%
         p.row        .      .      .
         p.col        .      .      .


----------
' 95% conf. level
```



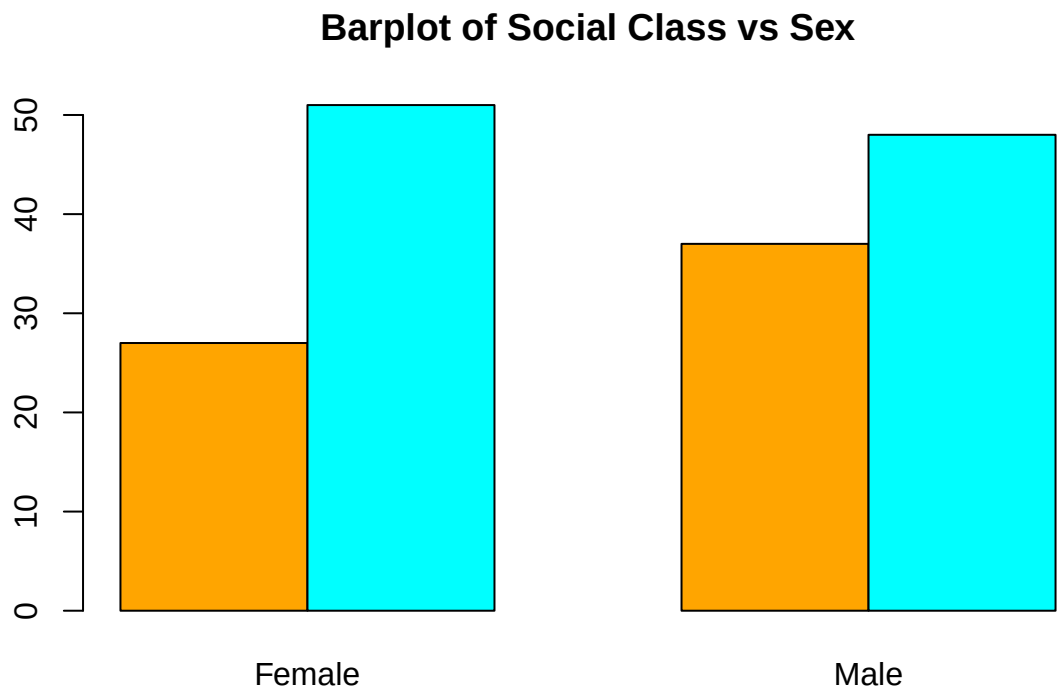
You can also obtain individual summary statistics for the variable using the `base` R function `table`. The function `prop.table` shows the overall proportion, and with `margin = 1` shows the proportion within rows and `margin = 2` shows the proportion within columns.

```
tab <- table(DF$soccl2, DF$sex)
tab

# Overall proportion
prop.table(tab)

# Proportion within rows
prop.table(tab, margin = 1)

# Proportion within columns
prop.table(tab, margin = 2)
```

To plot a simple barplot using `base` R, use the function `barplot` on the table object.

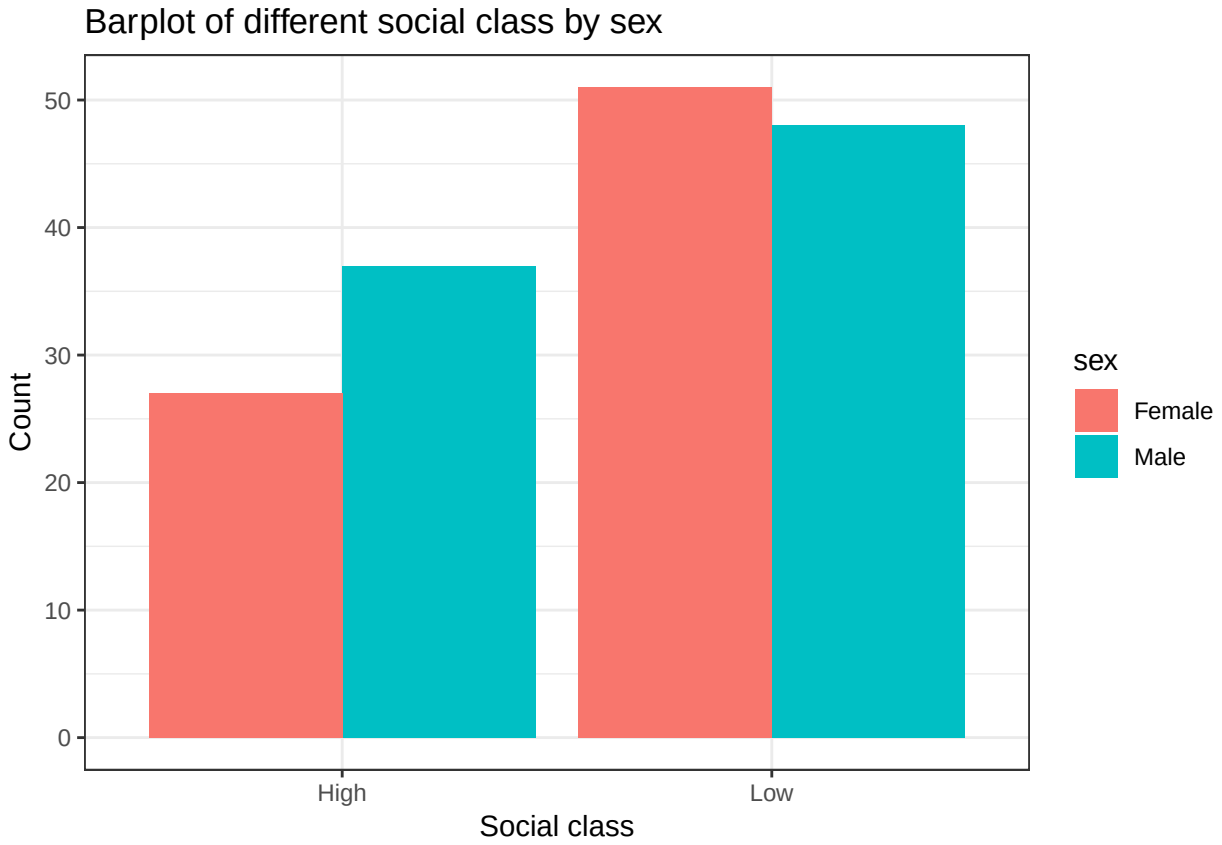
```
# Barplot

tab <- table(DF$soccl2, DF$sex)
```

```
barplot(tab, beside = TRUE,
    col=c('orange', 'cyan'),
    main='Barplot of Social Class vs Sex')
```



Or, to crate a barplot using `library(ggplot2)` R, run the following script.

```
# Barplot

g <- ggplot(data = DF, mapping = aes(x = soccl2, fill = sex))
g <- g + geom_bar(position = 'dodge')
g <- g + labs(title = 'Barplot of different social class by sex',
              x = 'Social class',
              y = 'Count')
g <- g + theme_bw()
g
```



6.3.2 Summary of outputs

The output shows that there are 78 females, of which 27 (34.6%) belong to social class 1 (High) and 51 (65.4%) belong to social class 2 (Low). Of 85 males, 37 (43.5%) belong to social class 1 (High) and 48 (56.5%) belong to social class 2 (Low).