Data Science in Medicine: Lab 1

Summarising and Visualising Data using R

```
# Author: Areti Manataki
# Last updated: 16th September 2020
# Description: This file is used as part of Lab 1 in the Data Science in Medici
ne course.
# Additional files needed: DataScienceClass.csv

# Instructions for students:
# To run a command, place your cursor on any part of it and click Ctrl+Enter (o
r Commd+Enter)
# To write a comment, include "#" at the beginning of the corresponding line.
```

Part 1: Basics

```
5+3
```

```
## [1] 8
```

```
# variables
age <- 28 + 10 # the result of the addition is stored in the variable age
age
```

```
## [1] 38
```

```
# functions
sqrt(225)
```

```
## [1] 15
```

```
# help with functions
?sqrt
```

Import data

```
## import data (csv format)
datasciClass <- read.csv("DataScienceClass.csv", header = TRUE, sep = ",")
```

```
# get a feel for the data
head(datasciClass) #print top part of the data
```

	Grades <int>	Degree <fctr>	Hours.of.sleep <dbl>	Gender <fctr>
1	69	Psychology	8.0	Female
2	70	Informatics	6.4	Male
3	86	Mathematics	8.3	Female
4	42	Medicine	6.2	Male
5	54	Informatics	6.0	Male
6	79	Medicine	7.4	Female

6 rows

```
names(datasciClass) #column names
```

```
## [1] "Grades"         "Degree"         "Hours.of.sleep" "Gender"
```

```
str(datasciClass) #data structure
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ Grades        : int  69 70 86 42 54 79 69 35 43 58 ...
##  $ Degree        : Factor w/ 4 levels "Informatics",..: 4 1 2 3 1 3 3 2 2 1
...
##  $ Hours.of.sleep: num  8 6.4 8.3 6.2 6 7.4 9 6.1 6.3 6.7 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 2 1 2 2 1 1 2 2 1
...
```

```
# get the entire dataset (not recommended - for demonstration purposes here)
datasciClass
```

Grades <int>	Degree <fctr>	Hours.of.sleep <dbl>	Gender <fctr>
69	Psychology	8.0	Female
70	Informatics	6.4	Male
86	Mathematics	8.3	Female
42	Medicine	6.2	Male
54	Informatics	6.0	Male
79	Medicine	7.4	Female

Grades	Degree	Hours.of.sleep	Gender
<int>	<fctr>	<dbl>	<fctr>
69	Medicine	9.0	Female
35	Mathematics	6.1	Male
43	Mathematics	6.3	Male
58	Informatics	6.7	Female

1-10 of 30 rows Previous **1** 2 3 Next

```
# get the Grades column within datasciClass
datasciClass$Grades
```

```
##  [1] 69 70 86 42 54 79 69 35 43 58 95 54 68 40 38 86 84 75 66 69 57 69 63
## [24] 75 71 58 59 57 55 67
```

Part 2: Summary statistics

```
# getting overall summary
summary(datasciClass)
```

```
##      Grades              Degree  Hours.of.sleep     Gender
##  Min.   :35.00   Informatics:6   Min.   :6.000   Female:18
##  1st Qu.:55.50   Mathematics:8   1st Qu.:6.550   Male  :12
##  Median :66.50   Medicine   :9   Median :7.250
##  Mean   :63.70   Psychology :7   Mean   :7.433
##  3rd Qu.:70.75                  3rd Qu.:8.200
##  Max.   :95.00                  Max.   :9.200
```

```
mean(datasciClass$Grades)
```

```
## [1] 63.7
```

```
median(datasciClass$Grades)
```

```
## [1] 66.5
```

```
max(datasciClass$Grades)
```

```
## [1] 95
```

```
min(datasciClass$Grades)
```

```
## [1] 35
```

```
range(datasciClass$Grades)
```

```
## [1] 35 95
```

```
max(datasciClass$Grades) - min(datasciClass$Grades)
```

```
## [1] 60
```

```
var(datasciClass$Grades)
```

```
## [1] 224.2172
```

```
sd(datasciClass$Grades)
```

```
## [1] 14.97389
```

```
table(datasciClass$Degree)
```

```
##
## Informatics Mathematics    Medicine  Psychology
##           6           8           9           7
```

```
# summarising by group
by(datasciClass$Grades, datasciClass$Degree, mean)
```

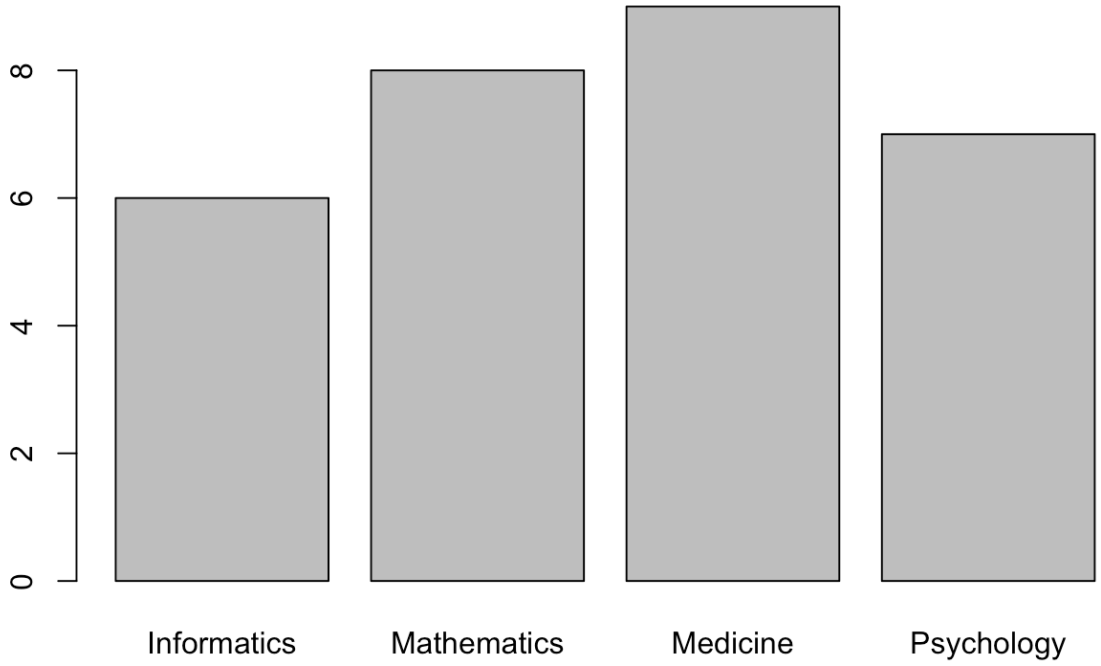
```
## datasciClass$Degree: Informatics
## [1] 71.5
## ------------------------------------------------------------
## datasciClass$Degree: Mathematics
## [1] 62.625
## ------------------------------------------------------------
## datasciClass$Degree: Medicine
## [1] 62.55556
## ------------------------------------------------------------
## datasciClass$Degree: Psychology
## [1] 59.71429
```

```
by(datasciClass, datasciClass$Degree, summary)
```

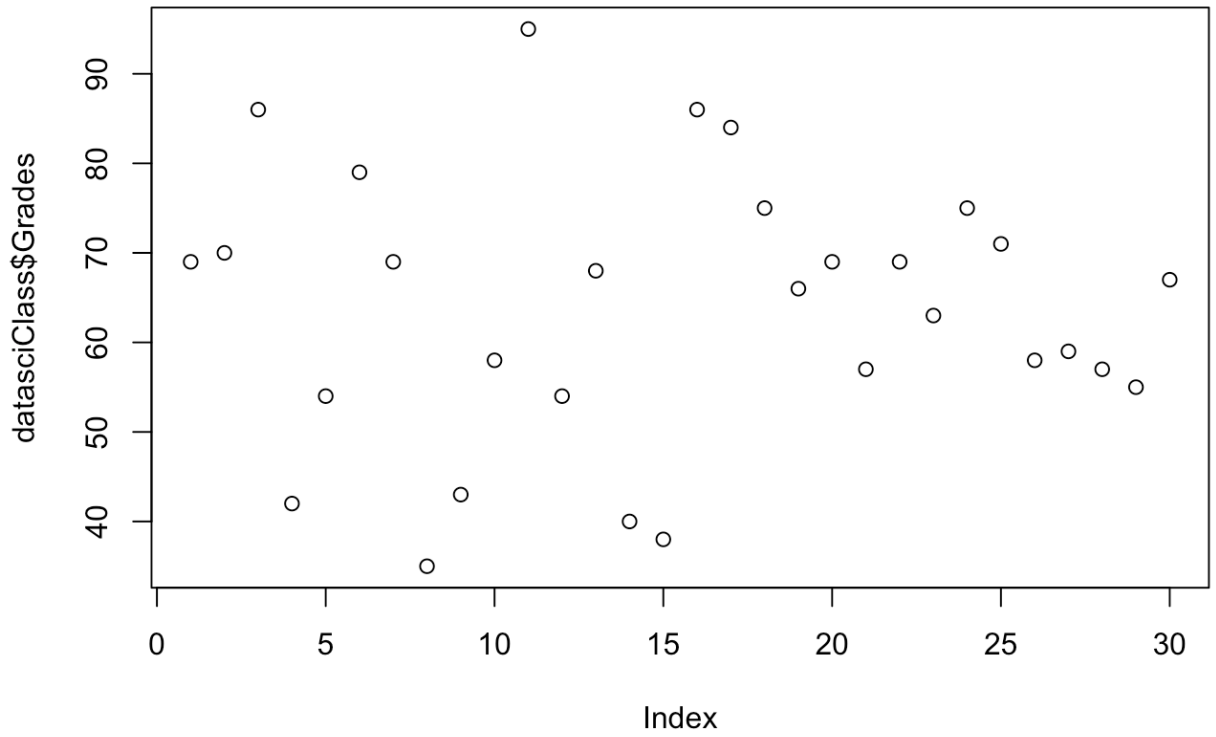
```
## datasciClass$Degree: Informatics
##      Grades             Degree  Hours.of.sleep     Gender
##  Min.   :54.0   Informatics:6   Min.   :6.000   Female:2
##  1st Qu.:60.0   Mathematics:0   1st Qu.:6.475   Male  :4
##  Median :68.0   Medicine   :0   Median :6.800
##  Mean   :71.5   Psychology :0   Mean   :6.967
##  3rd Qu.:82.0                   3rd Qu.:7.125
##  Max.   :95.0                   Max.   :8.600
## ------------------------------------------------------------
## datasciClass$Degree: Mathematics
##      Grades             Degree  Hours.of.sleep     Gender
##  Min.   :35.00   Informatics:0   Min.   :6.100   Female:5
##  1st Qu.:52.00   Mathematics:8   1st Qu.:7.050   Male  :3
##  Median :68.00   Medicine   :0   Median :8.100
##  Mean   :62.62   Psychology :0   Mean   :7.675
##  3rd Qu.:72.00                   3rd Qu.:8.225
##  Max.   :86.00                   Max.   :9.000
## ------------------------------------------------------------
## datasciClass$Degree: Medicine
##      Grades             Degree  Hours.of.sleep     Gender
##  Min.   :40.00   Informatics:0   Min.   :6.100   Female:6
##  1st Qu.:58.00   Mathematics:0   1st Qu.:6.500   Male  :3
##  Median :68.00   Medicine   :9   Median :7.000
##  Mean   :62.56   Psychology :0   Mean   :7.144
##  3rd Qu.:69.00                   3rd Qu.:7.400
##  Max.   :79.00                   Max.   :9.000
## ------------------------------------------------------------
## datasciClass$Degree: Psychology
##      Grades             Degree  Hours.of.sleep     Gender
##  Min.   :38.00   Informatics:0   Min.   :6.000   Female:5
##  1st Qu.:55.50   Mathematics:0   1st Qu.:7.350   Male  :2
##  Median :57.00   Medicine   :0   Median :8.000
##  Mean   :59.71   Psychology :7   Mean   :7.929
##  3rd Qu.:64.00                   3rd Qu.:8.800
##  Max.   :84.00                   Max.   :9.200
```

Part 3: Visualising data

```
plot(datasciClass$Degree)
```

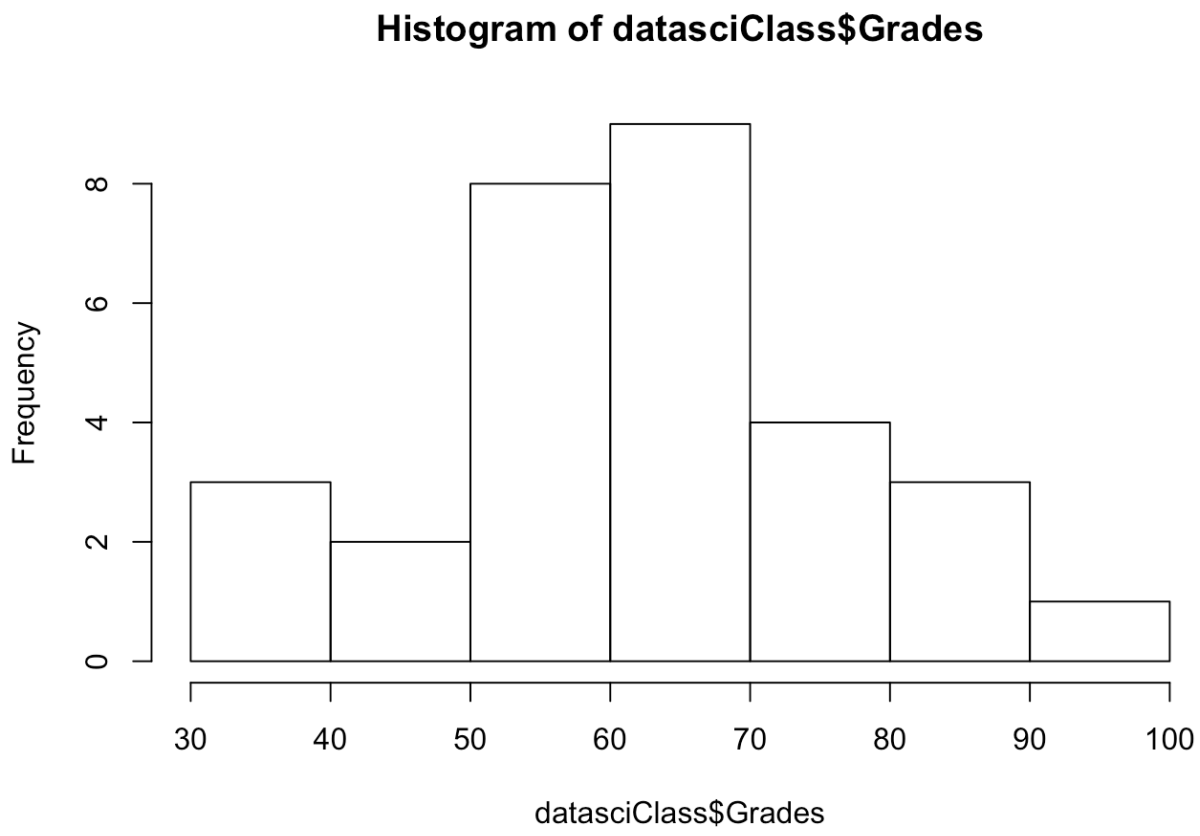



```
plot(datasciClass$Grades)
```

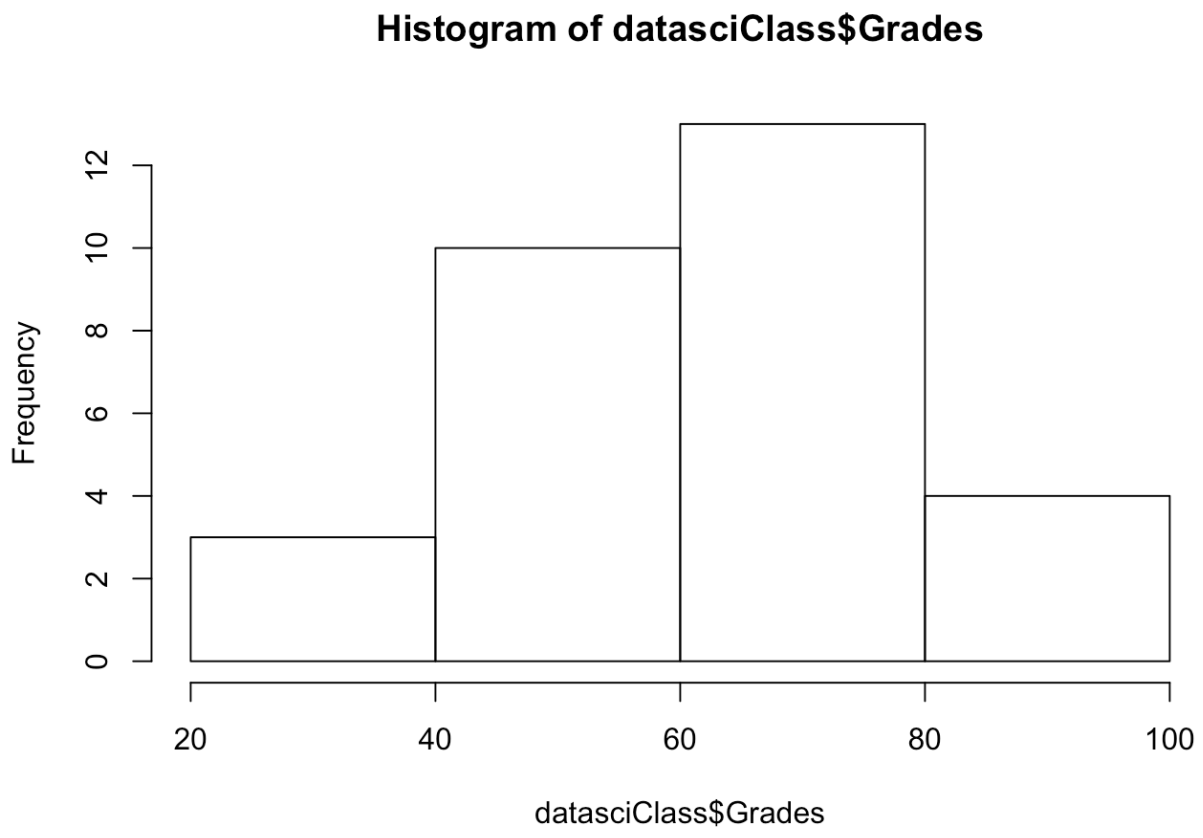


Histograms

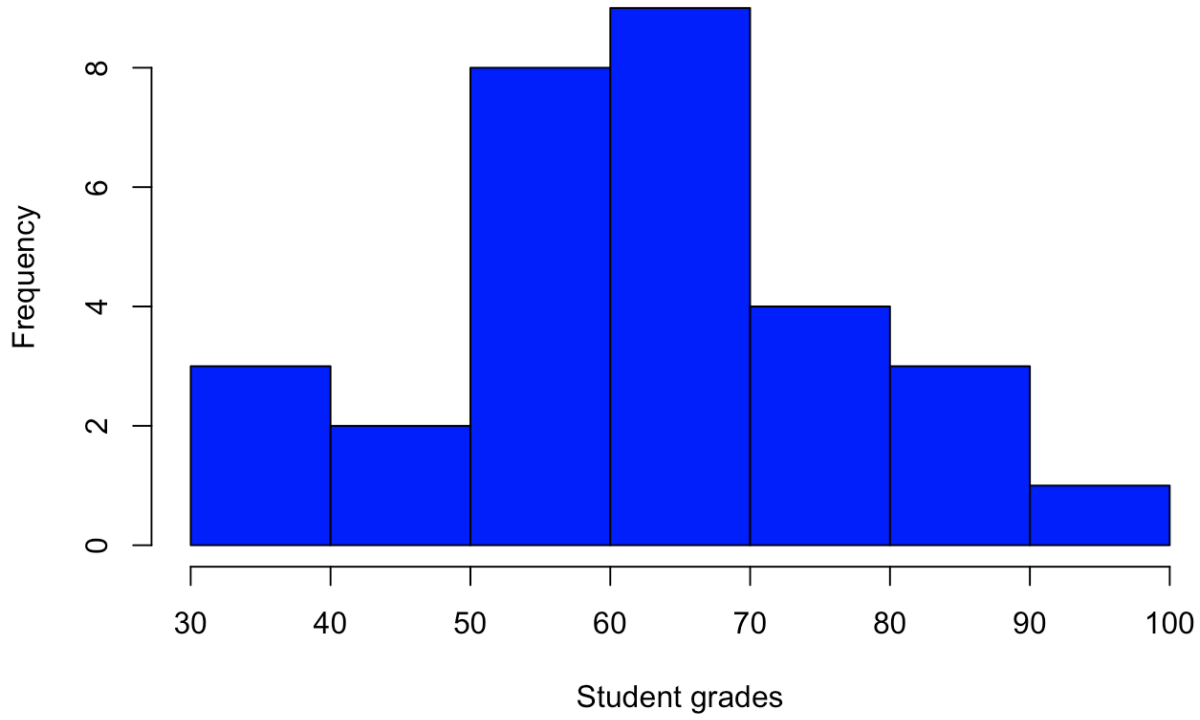
```
hist(datasciClass$Grades)
```



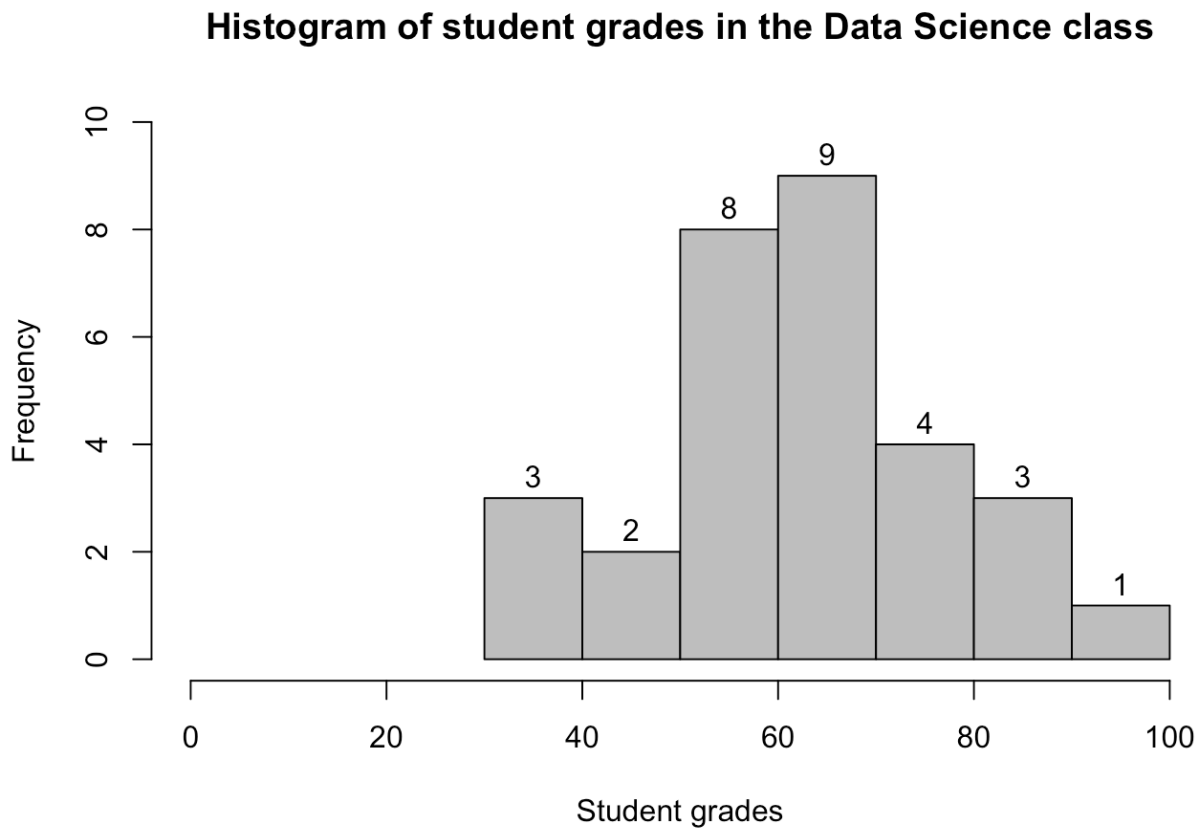

```
hist(datasciClass$Grades, breaks = 4) # set the number of bins
```




```
hist(datasciClass$Grades,
     main = "Histogram of student grades in the Data Science class", # set the
title of the plot
     xlab = "Student grades", # set the x-axis label
     ylab = "Frequency", # set the y-axis label
     col = "blue" # change the colour of the plot
     )
```

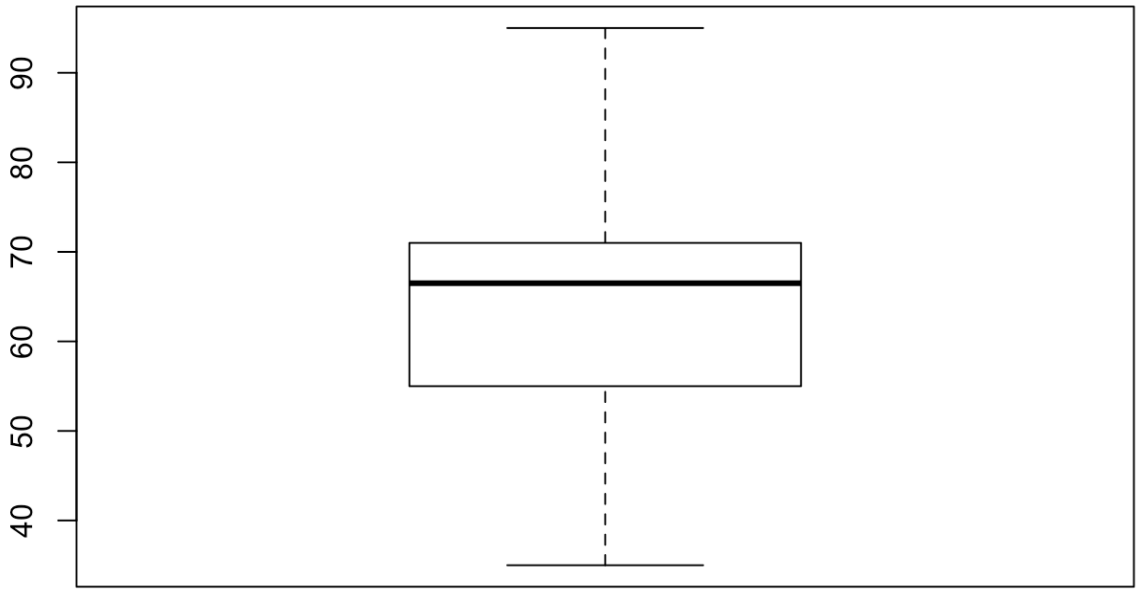



```
hist(datasciClass$Grades,
     main = "Histogram of student grades in the Data Science class",
     xlab = "Student grades",
     ylab = "Frequency",
     col = "grey",
     xlim = c(0, 100), # change the scale of the x-axis
     ylim = c(0, 10), # change the scale of the y-axis
     labels = TRUE # add frequency labels to each bar
     )
```

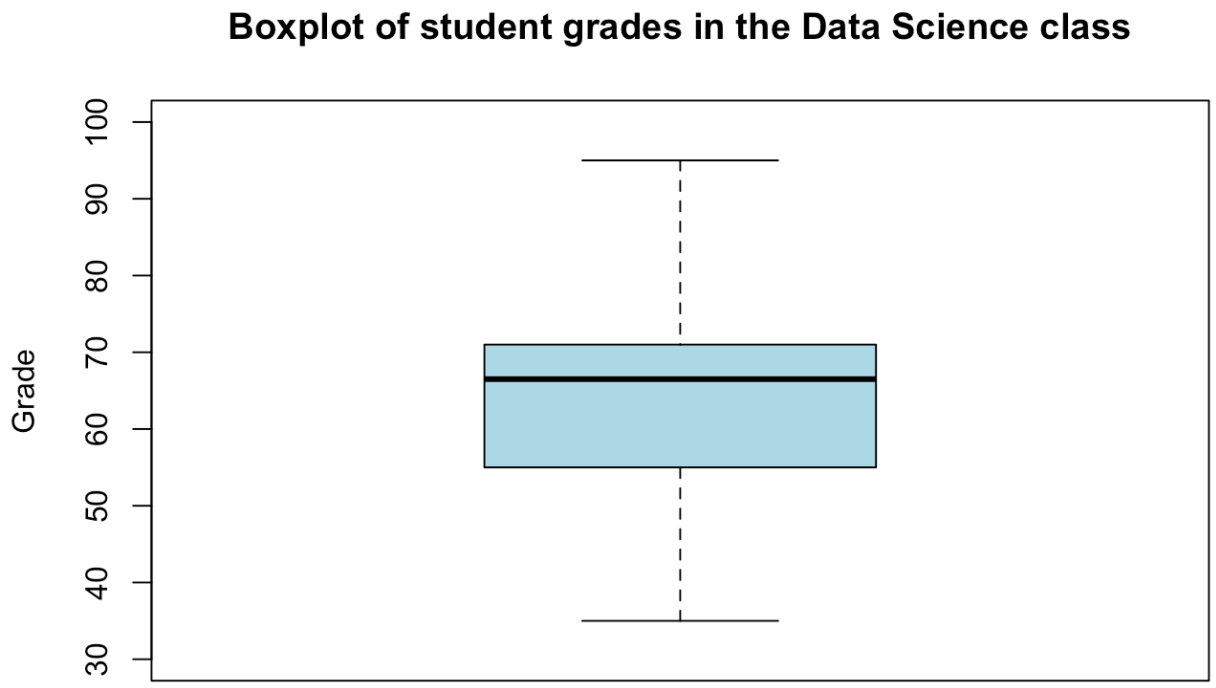


Boxplots

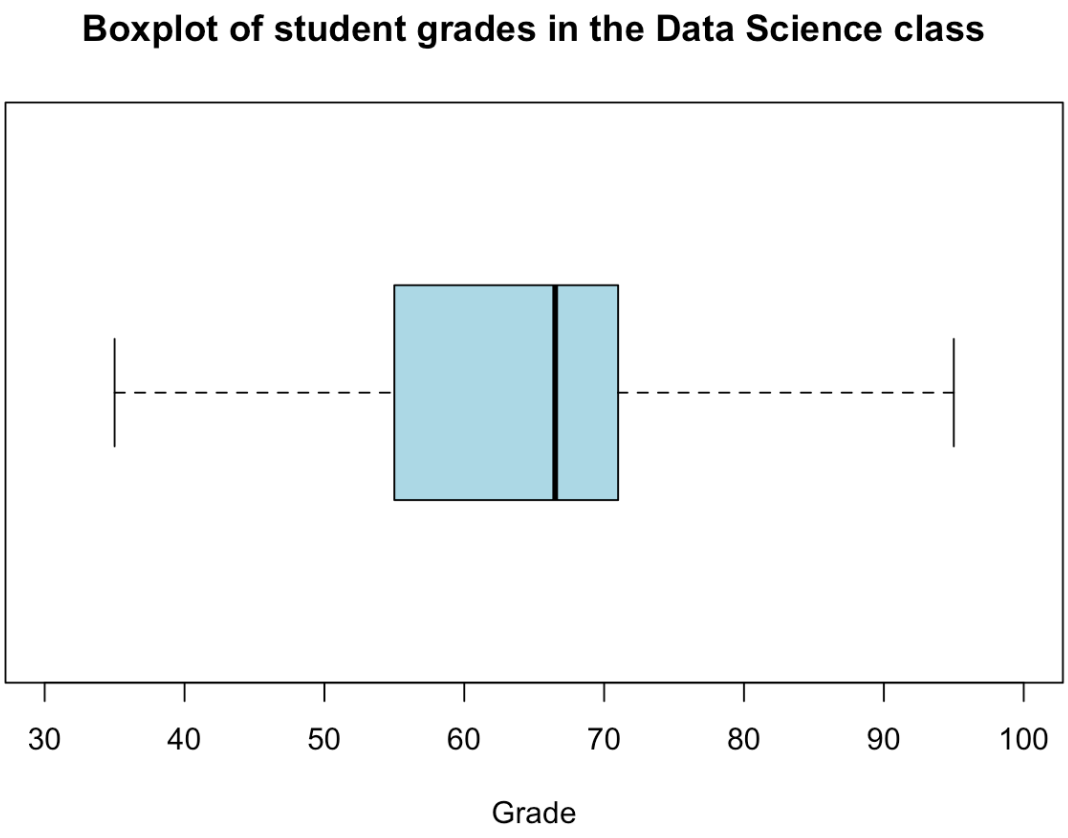
```
boxplot(datasciClass$Grades)
```



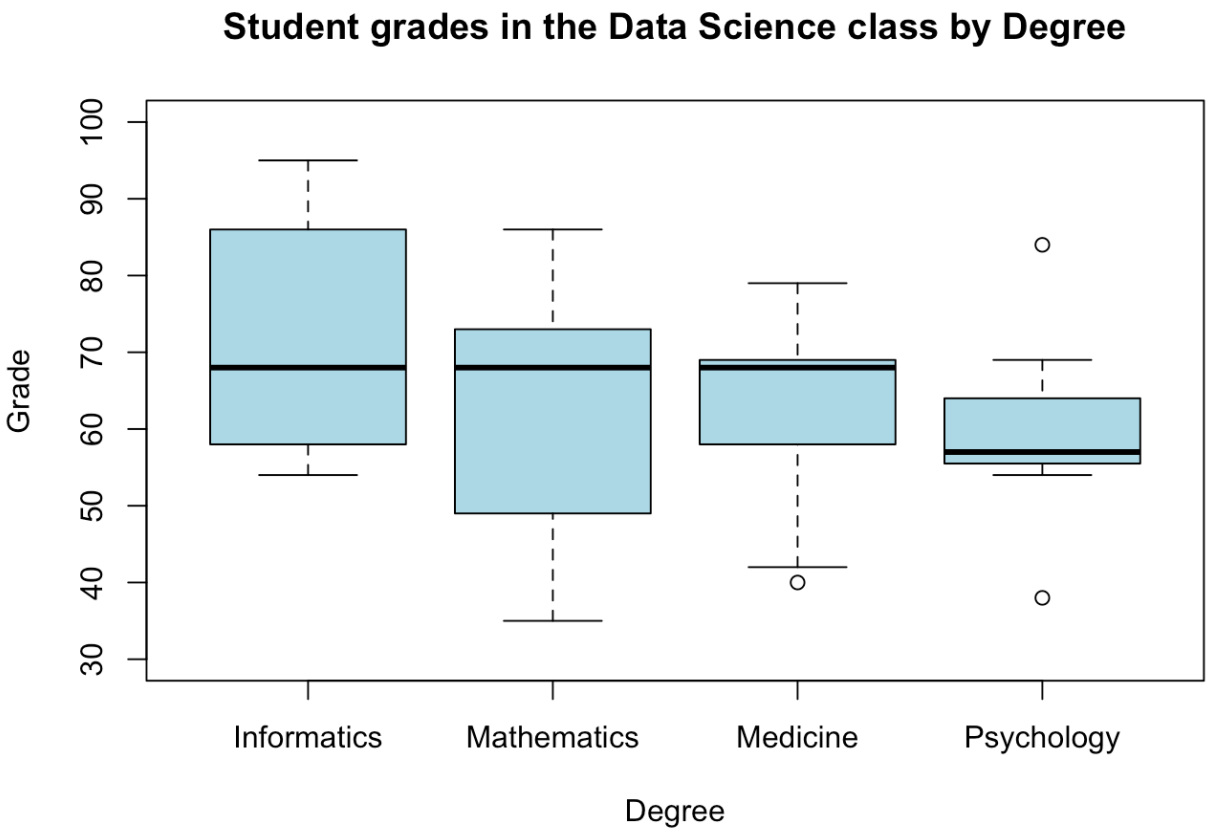

```
boxplot(datasciClass$Grades,
        main = "Boxplot of student grades in the Data Science class",
        ylab = "Grade",
        col="lightblue",
        ylim = c(30, 100)
        )
```




```
boxplot(datasciClass$Grades,
        main = "Boxplot of student grades in the Data Science class",
        xlab = "Grade",
        col="lightblue",
        ylim = c(30, 100),
        horizontal = TRUE # display the plot horizontally
        )
```




```
# boxplot by group
boxplot(datasciClass$Grades~datasciClass$Degree,
        main = "Student grades in the Data Science class by Degree",
        xlab = "Degree",
        ylab = "Grade",
        col="lightblue",
        ylim = c(30, 100)
        )
```

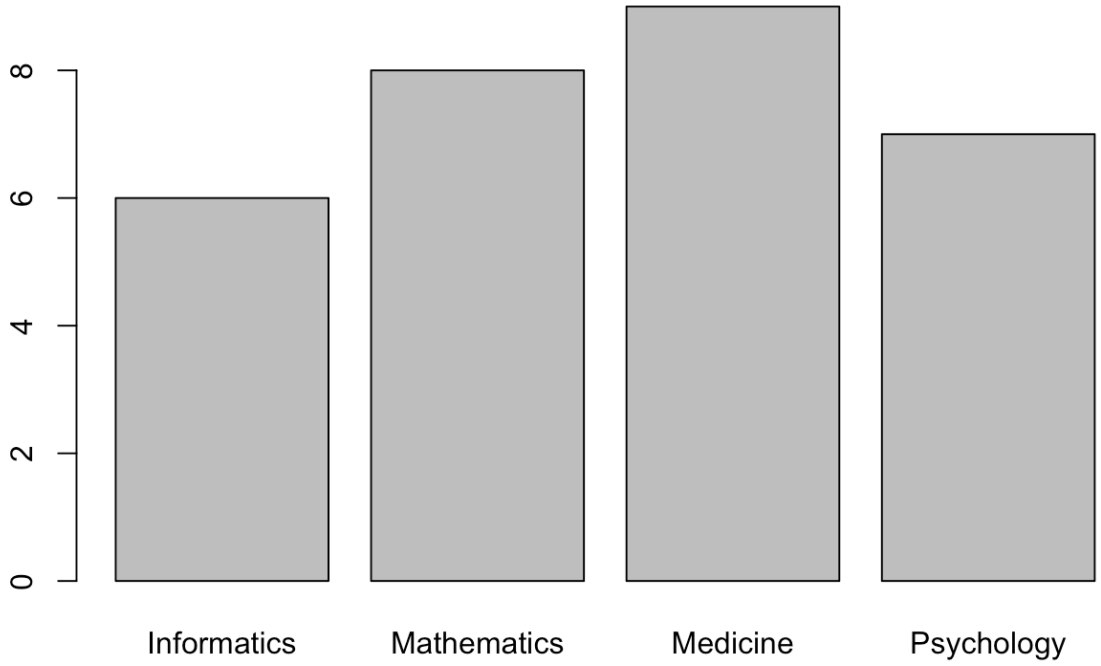


Bar chart

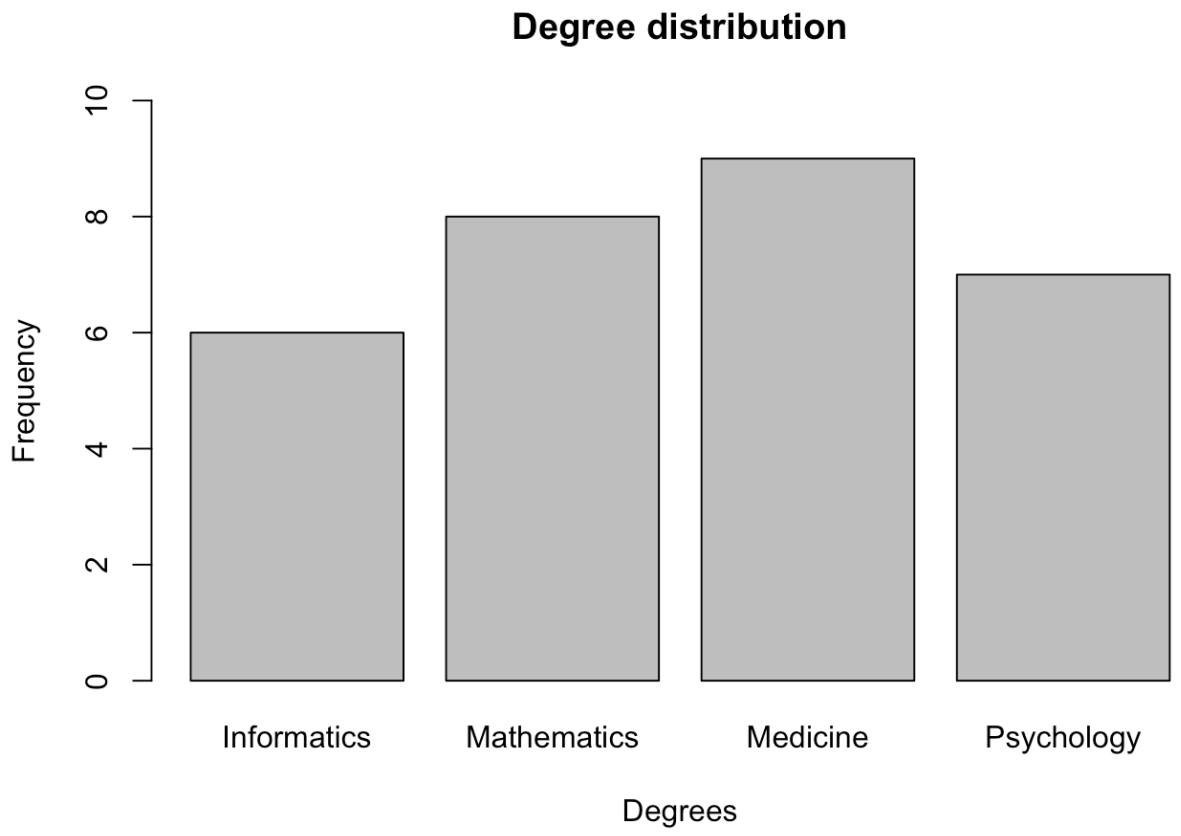
```
# first get a frequency table
freq <- table(datasciClass$Degree)
freq
```

```
##
## Informatics Mathematics    Medicine  Psychology
##           6           8           9           7
```

```
# get a bar chart
barplot(freq)
```



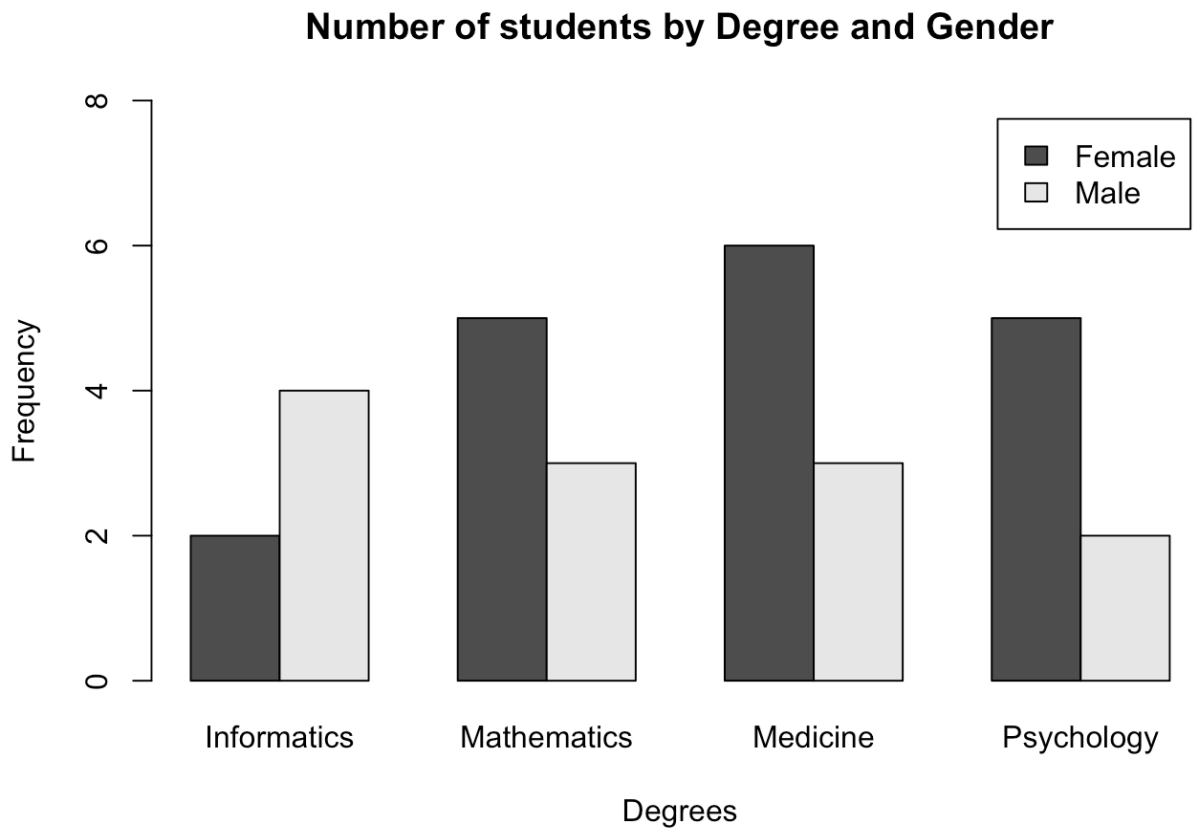

```
barplot(freq,
        main = "Degree distribution",
        xlab = "Degrees",
        ylab = "Frequency",
        ylim = c(0, 10)
        )
```




```
# get a grouped bar chart
freq2 <- table(datasciClass$Gender, datasciClass$Degree)
freq2
```

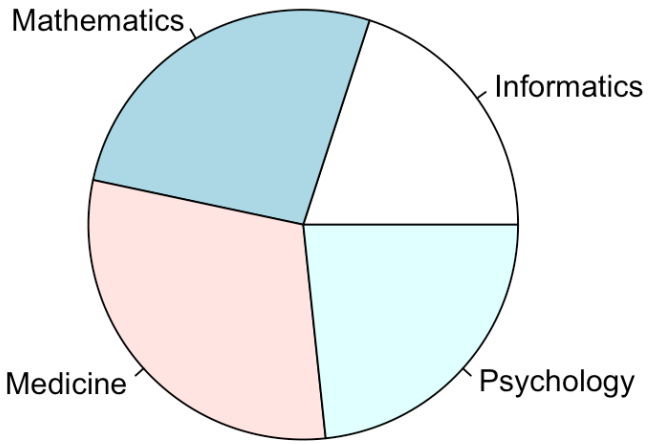
```
##
##          Informatics Mathematics Medicine Psychology
##   Female           2           5        6          5
##   Male             4           3        3          2
```

```
barplot(freq2,
        main = "Number of students by Degree and Gender",
        xlab = "Degrees",
        ylab = "Frequency",
        ylim = c(0, 8),
        beside=TRUE, # get a grouped bar chart (if FALSE, then we get a stacked
bar chart)
        legend = rownames(freq2) # get the legend
        )
```

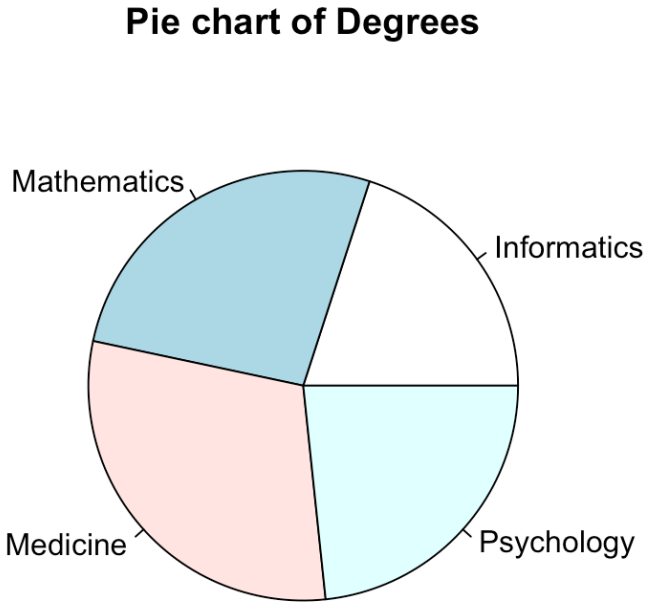


Pie chart

```
pie(freq)
```

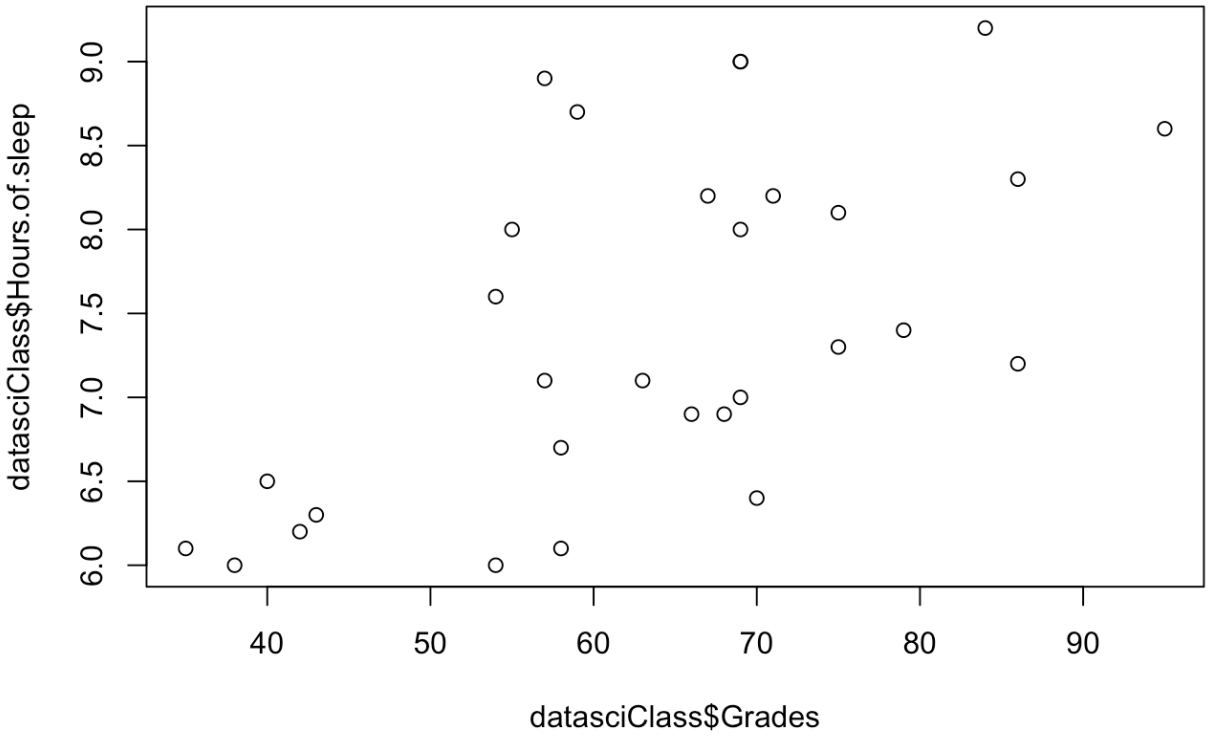



```
pie(freq, main = "Pie chart of Degrees")
```

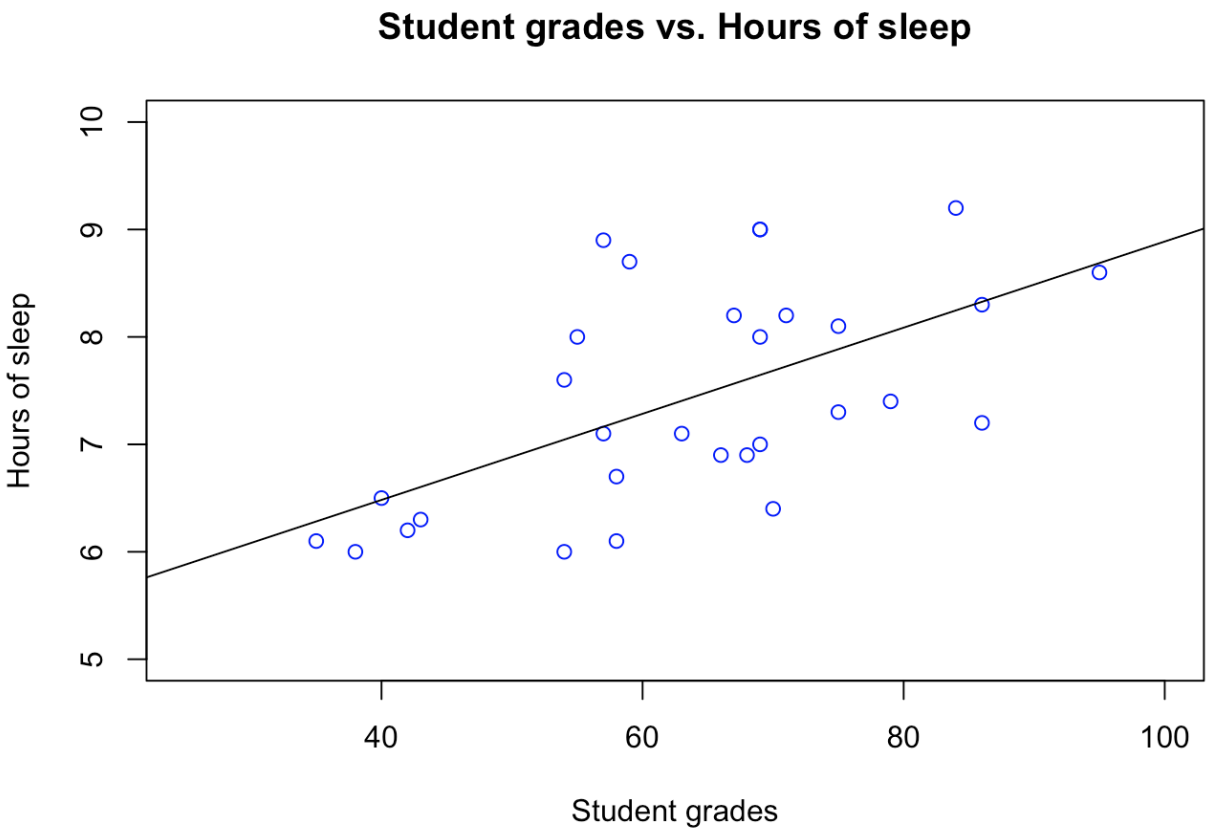


Scatterplot

```
plot(datasciClass$Grades, datasciClass$Hours.of.sleep)
```



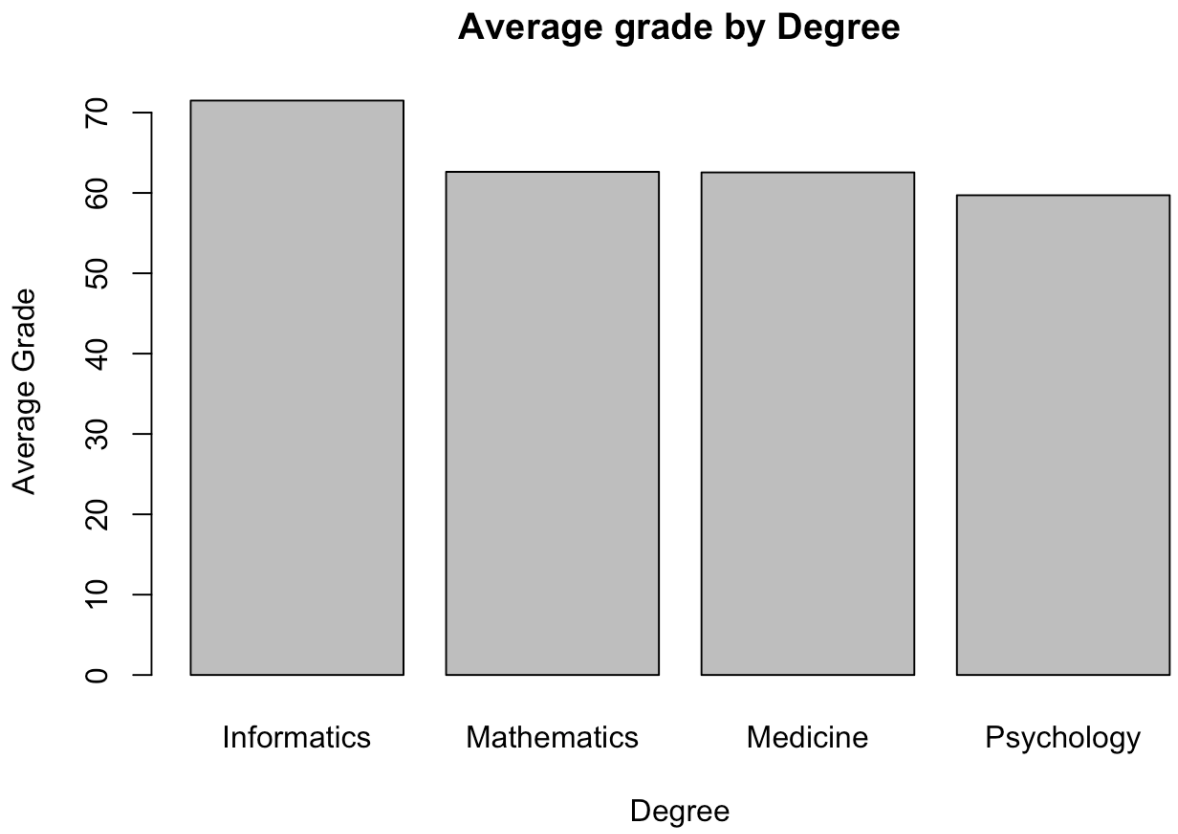

```
plot(datasciClass$Grades, datasciClass$Hours.of.sleep,
     main = "Student grades vs. Hours of sleep",
     xlab = "Student grades",
     ylab = "Hours of sleep",
     xlim = c(25, 100),
     ylim = c(5, 10),
     col = "blue"
     )

# draw a line of best fit
abline(lm(datasciClass$Hours.of.sleep ~ datasciClass$Grades))
```




```
# Note that:
# lm: generates a linear regression model of the two variables
# abline: draws trend line
```

Visualise by group

```
# visualising a variable mean per group
g <- by(datasciClass$Grades, datasciClass$Degree, mean)
barplot(g,
        main = "Average grade by Degree",
        xlab = "Degree",
        ylab = "Average Grade"
        )
```



Part 4: Manipulating data

Vectors

```
#create a numerical vector
weekly_sales <- c(200, 120, 130, 125, 220)
weekly_sales
```

```
## [1] 200 120 130 125 220
```

```
# get the length of the weekly_sales vector
length(weekly_sales)
```

```
## [1] 5
```

```
#create a character vector
friends <- c("maria", "john", "harry")


## indexing vectors
# get the 3rd element
weekly_sales[3]
```

```
## [1] 130
```

```
# get the 3rd and 5th element
weekly_sales[c(3,5)]
```

```
## [1] 130 220
```

```
# get from the 3rd up to the 5th elements
weekly_sales[3:5]
```

```
## [1] 130 125 220
```

```
# get the elements with index 3, 4, 5, 2 and 4 (in this order)
weekly_sales[c(3, 4, 5, 2, 4)]
```

```
## [1] 130 125 220 120 125
```

```
## subsetting vectors
weekly_sales[weekly_sales > 180]
```

```
## [1] 200 220
```

```
weekly_sales[weekly_sales > 180 | weekly_sales < 128 ]
```

```
## [1] 200 120 125 220
```

```
#alter elements of a vector
weekly_sales[3] <- 140

#add elements to a vector
weekly_sales[6] <- 130
```

Factors

```
gender <- c(1, 1, 2, 1)

# encode a vector as a factor
gender <- as.factor(gender)

# get the levels of a factor
levels(gender)
```

```
## [1] "1" "2"
```

```
# set the levels of a factor
levels(gender) <- c("male", "female")
```

Data Frames

```
name <- c("Tom", "Dave", "Anna", "John")
age <- c (20, 35, 28, 30)

# create a data frame by combining vectors
people <- data.frame(name, age, gender)
people
```

name <fctr>	age <dbl>	gender <fctr>
Tom	20	male
Dave	35	male
Anna	28	female
John	30	male

4 rows

```
## indexing data frames
# get the element that is on the 1st row and 2nd column
people[1,2]
```

```
## [1] 20
```

```
# get the elements that are on the 1st row and on columns 1 up to 3
people[1, 1:3]
```

	name <fctr>	age <dbl>	gender <fctr>
1	Tom	20	male

1 row

```
# get the entire 1st row
people[1,]
```

	name <fctr>	age <dbl>	gender <fctr>

	name <fctr>	age <dbl>	gender <fctr>
1	Tom	20	male

1 row

```
# get the entire 2nd column
people[,2]
```

```
## [1] 20 35 28 30
```

```
# get the entire age column
people$age
```

```
## [1] 20 35 28 30
```

```
## subsetting data frames (notice the comma!)
# get all rows that satisfy a constraint (one or more constraints)
people[people$gender=="male",]
```

	name <fctr>	age <dbl>	gender <fctr>
1	Tom	20	male
2	Dave	35	male
4	John	30	male

3 rows

```
people[(people$gender=="male") & (people$age>22),]
```

	name <fctr>	age <dbl>	gender <fctr>
2	Dave	35	male
4	John	30	male

2 rows

```
# Useful: We can use subsetting to summarise a particular group in datasciClass
infgroup <- datasciClass[datasciClass$Degree=='Informatics',]
mean(infgroup$Grades)
```

```
## [1] 71.5
```

```
## add a new column to a dataframe and instantiate it
people$city <- c("Edinburgh", "Edinburgh", "Aberdeen", "Glasgow")
```

Part 5: More basics

```
#help with functions
?sqrt
?qplot
```

```
## No documentation for 'qplot' in specified packages and libraries:
## you could try '??qplot'
```

```
??qplot

#define your own function
sum_of_two <- function(x, y){
  z <- 2*y
  x + z
}

sum_of_two(3,5)
```

```
## [1] 13
```

```
# when reassigning values to an object, the old ones cease to exist
myAge <- 23
myAge
```

```
## [1] 23
```

```
myAge <- 45
myAge
```

```
## [1] 45
```

```
# It can sometimes be useful to create a copy of an object that we want to modi
fy
people$age2 <- people$age + 10
```