Big Data Analytics

Lab 2

1 Description

In this lab we are going to explore the way of visualising and summarising data in R.

2 Data set

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica. iris is a data frame with 150 cases (rows) and 5 variables (columns) named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`. Data set can be loaded in R by running:

```
library(datasets)
```

Then it is available under the name `iris` (type it and run to see what it contains). Check the help to get more information about the data set.

Previously we have seen some data types, especially different types of vectors. Here, the data set is returned as a data frame structure which is a structure that enables you to store different variables of the same data, together. If you would like to store all students information, you could have a data frame with columns such as name, date of birth, address etc, and in each row we would store student's records (more on this during the lecture). You can check if Iris data set is a data frame:

```
is.data.frame(iris)
```

You can access data frames in the same way you would access an array - like in the first lab. You can just call `iris[1, 2]` to get the second column of the first row. To retrieve the whole column or row, just leave the index empty, like `iris[, 2]` to get the whole second column. It is also possible to access a column by the name - `iris$Sepal.Length` will return column containing mpg. This one:

```
iris[1:10,]$Sepal.Length
```

takes first 10 rows and then it returns only the column `Sepal.Length`.

3 Visualisation

Before we start exploring different ways of visualising data, there is a useful R command that aggregates categorical data:

```
table(iris$Species)
```

3.1 Pie Chart

Pie charts are very useful when you would like to present some summaries of categorical data. In R you create them in the following way:

```
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(slices, labels = lbls, main="Pie Chart of Countries")
```

3.2 Bar Chart

Similar to pie charts - we use them if we have categorical data and we would like to visualise how many of observations we have in each category.

```
counts <- table(iris$Species)
barplot(counts, main="Species Distribution", xlab="Species")
```

3.3 Histogram

Useful to summarise numerical data by aggregating them into buckets and counting.

```
hist(iris$Sepal.Width, main="Histogram of sepal width", xlab="Sepal Width")
```

3.4 Scatter Plot

Scatter plot displays values for typically two variables for a set of data. By colouring the points or changing the shape it is possible to increase the number of variables.

```
plot(iris$Sepal.Length, iris$Sepal.Width,
     xlab="Sepal Length", ylab="Sepal Width", pch=19)
```

4 Summaries

We also have very easy and useful functions to summarise the data. Consider the following data (discussed in last lecture).

```
x <- c(414, 123, 72, 79, 66, 84, 169, 144, 102, 110, 162) # 11 elements
y <- c(414, 123, 72, 79, 66, 84, 169, 144, 102, 110)      # 10 elements
```

4.1 Median

Median is the centre value of a data set. It splits the data set into two halves. Check with the results from the lecture slides.

```
median(x)
median(y)
```

4.2 Mean

Mean is an arithmetic average of the data set:

```
mean(x)
mean(y)
```

4.3 Quantiles

To get the quantiles of the data set run:

```
quantile(x, type=1)
quantile(y, type=1)
```

Check different types.

4.4 Head

Probably viewing the whole data set is not sufficient. Instead it is possible to view only top `n` rows of the data set:

```
head(iris)
```

There is also `tail` function. Try it to see what it does.

4.5 Summary

Instead of checking min, max, median, mean, etc. separately for each column, it can be done all together:

```
summary(iris)
```

4.6 Standard Deviation

Standard deviation is a measure that is used to quantify the amount of dispersion of dataset values:

```
library(stats)
sd(x)
sd(y)
```

For calculating variance type `var(x)` and `var(y)`.

5 Questions

Answer the following questions!

1. Calculate square root of 31729
2. Create a new object 'b' with value 3247.5
3. Convert 'b' from previous question to Character, Integer, and Logical classes.
4. Create a vector numbers from 10 to 80 and find out its class.
5. Create a vector containing following mixed elements (212, 'a', 23.13, 'b', TRUE, FALSE) and find out its class.
6. Create an empty vector of class character having the length 50. Display the default values.
7. Assign the characters 'a', 'b', 'c', 'd', and 'e' in above vector (10 each).
8. Create a vector with some of your friend's names.
9. Get the length of above vector.
10. Sort the vector (your friends) by names.
11. Reverse direction of the above sort.
12. Create a vector using rep and seq functions containing values: 'a','a',1,2,3,4,5,7,9,11
13. Remove missing value from c(NA, 13, 22, NA, 544, NA).
14. Find the class of "iris" dataset, find the class of all the columns of 'iris', get the summary of 'iris', get the top 6 rows, view it in a spreadsheet format, get row names, get column names, get number of rows and get number of columns.
15. Get rows with Sepal.Width greater than 3.5 from 'iris' dataset.