Intro to R Programming: Class Test 1

Please log in using your own credentials. During the test you are not allowed to talk to or otherwise communicate with other students (email, instant messaging, etc.), or access the internet/course material. The only material you can use is the R reference manual provided as well as the R help within RStudio.

Only at the end of the test when specified by the invigilators, please log on to Moodle and upload your script by clicking on the link *Upload your class test script*. Please upload one R script only including your code and comments. You do not need to upload R output. Please do not upload Word documents or Zip files. The name of the file submitted **must be** your student ID number.

Please make sure you regularly save your R script, just in case RStudio crashes. For all parts in the test give the R code which can be used to answer the questions below.

You have to attempt all questions within the time of one hour

Task 1

The data file `houseprices.csv` contains data on property sales in and around Glasgow between July 1st and December 31st 2014. It contains the following columns:

Day	Day of the month of the transaction
Month	Month of the transaction (integer)
Address	Address of the property
Lon	Longitude of the property (degrees)
Lat	Latitude of the property (degrees)
Price	Price

Crown copyright material reproduced with the permission of Registers of Scotalnd

1. **[1 mark]** Read the data file `houseprices.csv` correctly into a data frame called `houseprices`.
2. **[1 mark]** What is the average house price in August 2014?
3. **[2 marks]** Create a dataset called `houseprices.summer` including the transactions occurred between July 15th and August 15th. How many transactions occurred in that period?
4. **[1 mark]** Which house sold for the lowest price?
5. **[1 mark]** Transform the column `Lon` to include the longitude of the properties expressed in radians, i.e. divide the longitude by 180° and multiply by π. Repeat the same process for `Lat`.
6. **[4 marks]** Create a new variable `Dist2University` which contains the distance to the University in kilometres. Consider two locations with longitudes λ_1 and λ_2 and latitudes ϕ_1 and ϕ_2 expressed in radians. Define

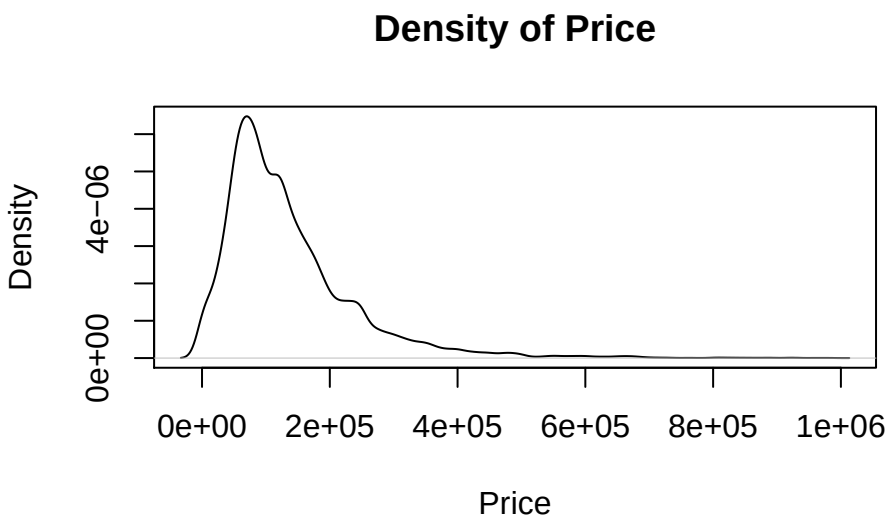
$$\begin{aligned}
\Delta\lambda &= \lambda_2 - \lambda_1 \\
\Delta\phi &= \phi_2 - \phi_1 \\
\alpha &= \sin\left(\frac{\Delta\phi}{2}\right)^2 + \cos(\phi_1)\cos(\phi_2)\sin\left(\frac{\Delta\lambda}{2}\right)^2 \\
d &= 12742\tan^{-1}\left(\frac{\sqrt{\alpha}}{\sqrt{1-\alpha}}\right),
\end{aligned}$$

then d gives the distance in kilometres between the two locations. The longitude and the latitude of the University are (in degrees):

$$\lambda = -4.2886^\circ, \qquad \phi = 55.8711^\circ$$

Hint: $tan^{-1}(\frac{a}{b})$ can be calculated in R using the function `atan2(a,b)`.

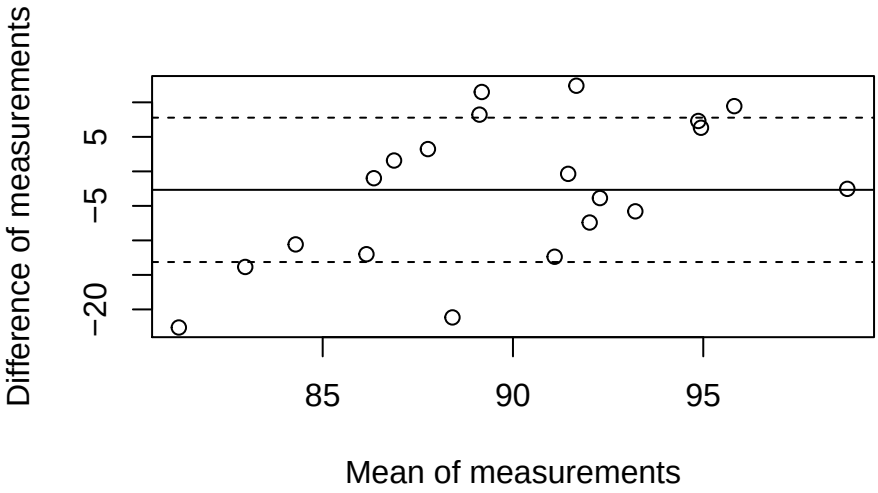
7. **[1 mark]** What was the average price of properties which are within 1km of the University?
8. **[2 marks]** Plot the density of the price of properties which cost less than one million. Include the title `Density of Price` and the label for the x-axis `Price`. Your plot should like like the following:



Task 2

The file `hearth.txt` contains measurements of the transmitral volumetric flow (`MF`) by Doppler echocardiography and left ventricular stroke volume (`SV`) by cross-section echocardiography in 21 patients without aortic valve disease.

1. **[1 mark]** Read the data correctly into R and store it in the data frame `hearth`.
2. **[1 mark]** Remove all rows containing missing values from `heart`.
3. **[2 marks]** Add the column `difference` including `MF - SV` and the column `mean` including (`MF`+ `SV`)/2 to the data frame `hearth`.
4. **[2 marks]** Suppose we are interested in informally assessing whether there is a systematic difference between the two techniques. This is often done using what is called the Bland-Altman plot, which is a scatter plot of `mean` against `difference`. Produce such a plot and label the x-axis `Mean of measurements` and the y-axis `Difference of measurements`.
5. **[2 marks]** Add a solid horizontal line at the average value of `difference` and a pair of dashed lines one standard deviation above and below the solid horizontal line. Your plot should look similar to the one below.

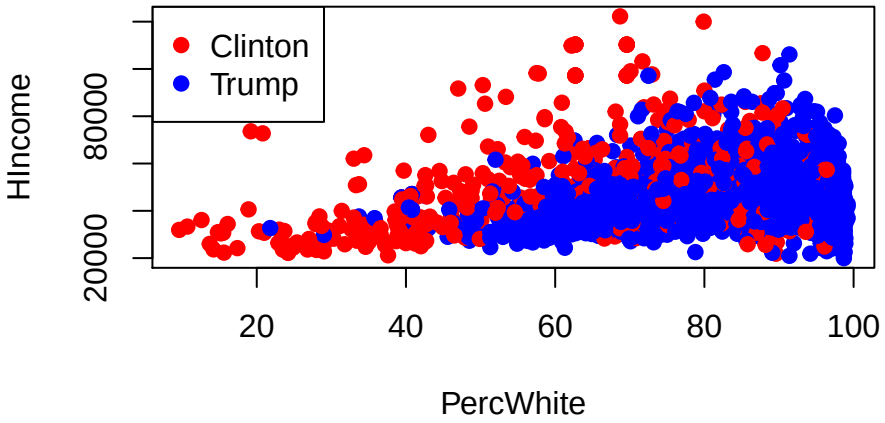


Task 3

The file `potus.txt` contains the results of the last presidential election for each county in the United States. The dataset contains the following columns:

County	Name of the county
State	Name of the state
VotesTrump	Number of votes cast for Donald Trump
VotesClinton	Number of votes cast for Hillary Clinton
PercTrump	Vote share (percent) for Donald Trump
PercClinton	Vote share (percent) for Hillary Clinton
PercWhite	Percentage of the population which is Caucasian
HIncome	Median household income.

1. [**1 mark**] Read the file `potus.txt` into R and store into a data frame called `potus`.
2. [**2 marks**] What is the average median household income in counties in which Donald Trump received at least three times as many votes as Hillary Clinton?
3. [**1 mark**] How many votes have been cast for Hillary Clinton in the state of California?
4. [**1 mark**] Add a column to `potus` named `Hillary.Wins` which is `TRUE` for counties where Hillary Clinton has more votes than Donald Trump and `FALSE` otherwise.
5. [**3 marks**] Create a plot of `HIncome` against `PercWhite`. Counties in which Hillary Clinton obtained more votes than Donald Trump should be plotted in red, the others in blue.
6. [**1 mark**] Add a legend to the plot from part 5. Your final plot should look similar to the one below.



Intro to R Programming: Class Test 2

During the test you are not allowed to talk to or otherwise communicate with other students (email, instant messaging, etc.), or access the internet/course material. The only material you can use is the R reference manual provided as well as the R help within RStudio.

Only at the end of the test when specified by the invigilators, please log on to Moodle and upload your script by clicking on the link *Upload your class test script*. Please upload one R script only including your code and comments. You do not need to upload R output. Please do not upload Word documents or Zip files. Files submitted within the time of 5 minutes from the end of the test will be considered for marking. It is your own responsibility to correctly submit your script.

Please make sure you regularly save your R script, just in case RStudio crashes.

You have to attempt all questions within the time of two hours

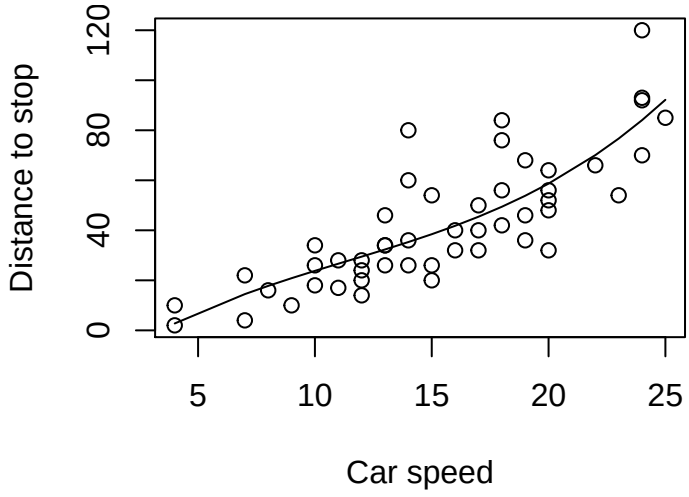
Task 1

The data set `cars` is freely available from R (just type `cars`) and includes the speed of 50 cars (`speed`) and the distances taken to stop (`dist`). In this question you will implement a cubic regression to predict the response variable `dist` variable given the covariate `speed`.

1. [**2 marks**] Produce a scatterplot of the data. Your plot should have labels `Car speed` and `Distance to stop` in the x-axis and y-axis respectively.
2. [**2 marks**] Create the design matrix

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{50} & x_{50}^2 & x_{50}^3 \end{pmatrix}$$

   where $x_1, \ldots, x_{50}$ are the values of the variable `speed`.
3. [**2 marks**] Create a matrix `XtX` which holds $X^{\mathrm{T}}X$ and a vector `Xty` which holds $X^{\mathrm{T}}y$, where $y = (y_1, \ldots, y_{50})$ are the values of `dist`.
4. [**1 mark**] Store in a vector `beta` the least-square estimates obtained by computing $\hat{\beta} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$.
5. [**1 mark**] Compute the fitted values $\hat{y} = X\hat{\beta}$ and store them in a vector `y.hat`.
6. [**1 mark**] Overlay the estimated cubic regression to the scatterplot of part 1. Your plot should look similar to the one below.



7. **[4 marks]** Write a function `least.square.cubic` which computes the parameter estimates $\hat{\beta}$ and the fitted values $\hat{y}$ of a univariate cubic regression model using the steps above. The input of the function should be a vector `x` including the covariate and the vector `y` including the response. The function should output a list with elements `parameters`, reporting $\hat{\beta}$, and `fitted`, reporting $\hat{y}$. The function should return an error if the lengths of `x` and `y` are different.

Task 2

Consider a vector $y = (y_1, \ldots, y_n)$ containing a noisy time series as shown in the picture below. Our objective is to remove the noise from it. A simple way of doing so is to use a so-called triagonal filter which yields a smoothed time series $\hat{y}$.

The triagonal filter is controlled by a width parameter w, which needs to be an odd integer (the larger w the smoother the output). The triagonal filter uses the following algorithm.

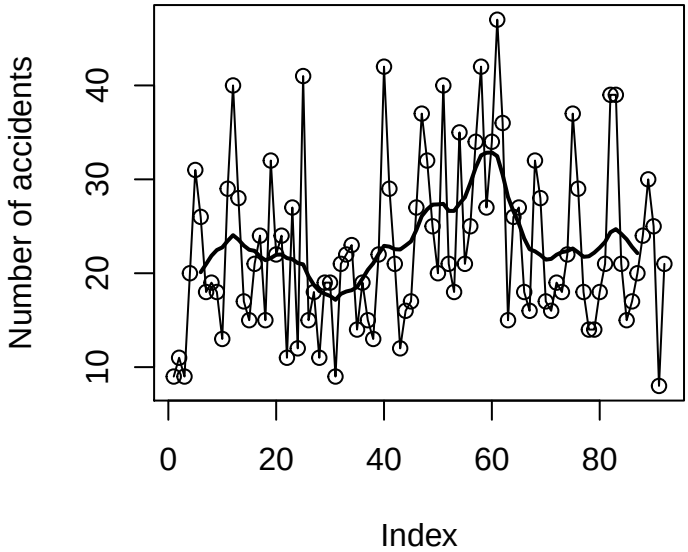
1. Compute $k = (w-1)/2$
2. For $i = k+1, \ldots, n-k$, compute

$$\hat{y}_i = \frac{y_{i-k} + 2y_{i-k+1} + \cdots + ky_{i-1} + (k+1)y_i + ky_{i+1} + \cdots + 2y_{i+k-1} + y_{i+k}}{(k+1)^2}$$

Note that this simple version of the triagonal filter does not produce smoothed values for the first k and last k observations. Thus you should set

$$\hat{y}_1 = \cdots = \hat{y}_k = \texttt{NA} \quad \text{and} \quad \hat{y}_{n-k+1} = \cdots = \hat{y}_n = \texttt{NA}$$

1. **[1 mark]** Consider the `Traffic` dataset from the `MASS` library (which you can access by typing `library(MASS)`). Store in a variable `accidents` the first 100 observations of the column `y` of `Traffic`.
2. **[2 marks]** Draw the values of `accidents` in a time series plot, with points connected by lines. Your plot should have labels `Index` and `Number of accidents` for the x-axis and y-axis respectively.
3. **[1 mark]** Set `w <- 11` and compute the coefficient `k` above.
4. **[3 marks]** Compute the smoothed time series $\hat{y}$ using the algorithm above with `w = 11`.
5. **[1 mark]** Add a line to the plot of part 2 reporting the smoothed time series $\hat{y}$ computed in part 4. Your plot should look similar to the one below.



6. **[4 marks]** Write a function `tri.filter` which takes a vector `y` and a scalar `w` as arguments and which returns the triagonal filter $\hat{y}$. $\hat{y}$ should be computed as set out above. The default value of w should be 11. Your code should check whether w is odd and return an error message if not.

Task 3

The index of dispersion is defined as the ratio of the variance σ^2 to the mean μ of a random variable, i.e. σ^2/μ. In this question you will implement a function to compute this index.

1. [**2 marks**] Define the function `disp.index` which takes as input a data frame `x` including only numeric columns and output the index of dispersion for each column of `x`. Your function cannot use `for` loops.
2. [**3 marks**] The `iris` dataset is freely available from R (just type `iris`). `iris` 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. Compute the index of dispersion of the variables sepal length, sepal width, petal length and petal width for each species of iris. You cannot manually select the species in your code. The dispersion indexes are as follows:

```
## iris$Species: setosa
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##   0.02482001   0.04191651   0.02062872   0.04514684
## ------------------------------------------------------------
## iris$Species: versicolor
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##   0.04488421   0.03554852   0.05183482   0.02949180
## ------------------------------------------------------------
## iris$Species: virginica
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##   0.06137566   0.03497111   0.05486091   0.03723231
```