Exercises for Lecture 5

Statistical Computing with R, 2023-24

Exercise 1

Create an R Markdown file where you will write your solutions to the exercises from this session. You can follow the same instructions given in exercise 1 of coding session 4 to set your `.Rmd` file up.

Exercise 2

The moving average with lag k for a vector $x = (x_1, x_2, ..., x_n)$ can be defined as the mean of the last k values up until (and including) x_t, where $t = 1, ..., n$:

$$MA_1(x; t; k) = \frac{1}{k} \sum_{i=\max(t-k+1,1)}^{t} x_i.$$

Let `x = c(3, 4, 6, 2, 4, 3, 8, 1, 3)`.

1. Compute $MA_1(x, 5, 3)$ and $MA_1(x, 2, 3)$.
2. Write a function that computes $MA_1(x; t; k)$ for a vector of arbitrary length.
3. Write a `for( )` loop to compute $MA_1(x; t; 3)$ for $t = 1, 2, ..., n$.

Exercise 3

Alternatively, we could define the moving average only when $t \geq k$:

$$MA_2(x; t; k) = \frac{1}{k} \sum_{i=t-k+1}^{t} x_i,$$

and set $MA_2(x; t; k) =$ `NA` when $t < k$.

1. Compute $MA_2(x, 5, 3)$ and $MA_2(x, 2, 3)$. Compare the results to those of exercise 2.
2. Write a function that computes $MA_2(x; t; k)$ for a vector of arbitrary length.
3. Write a `for( )` loop to compute $MA_2(x; t; 3)$ for $t = 1, 2, ..., 9$. Compare with the results of exercise 2.

Exercise 4

Recreate in your R session the following matrix:

```
set.seed(4)
r = 100
A = matrix(NA, nrow = r, ncol = 5)
A[ , 1:2] = rpois(2*r, 6)
A[ , 3] = rgamma(r, shape = 2, scale = 1)
A[ , 4:5] = rbinom(r, 2, 0.6)
```

1. Use a `for( )` loop to compute the mean of each column
2. Compute the mean of each row, and store the results in a vector
3. Write a function that computes the column means of a given matrix. Compare its output to that of the `colMeans` function
4. Write a function that returns the mean and standard deviation of each column in a matrix. The output of such function should be a data frame containing the following variables: column, mean, standard deviation

Exercise 5

The file `irish_polls.csv`, obtained from the website of *Europe elects* (https://europeelects.eu/), contains data on political opinion polls conducted in Ireland from 2016 to 2021.

1. Download the file from github.com/mirkosignorelli/Teaching/tree/main/SCwR_course and save it in the folder where your R Markdown file is located. Import the file in R.

Data import and data cleaning

As you will see, the dataset is not ready for immediate use in R: the variables with the estimated share of votes for the different parties are characters, contain the symbol `%` when percentages are available and the string `'Not Available'` when no data are available. To proceed, we need to first fix these problems and then convert the variables to numeric.

2. Focus your attention on columns 10 to 21. Replace all the `'Not Available'` occurrences with an `NA`, and remove the `%` sign otherwise.
3. Convert the variables in columns 10 to 21 to numeric.

Data analysis and visualization

For the rest of the exercise, we will restrict our attention to the 10 most recent opinion polls available in the data frame.

4. Select the 10 most recent polls (i.e., the first 10 rows in the imported data frame)
5. Compute the average estimated share of each party. Which parties have a mean share above 6%? For the rest of the exercise, consider only those parties, plus an

additional category named "Others" whose value is 100 minus the sum of the shares of the aforementioned parties.

6. Draw a bar plot that compares the shares of different political parties. Make sure that bars are sorted by party share, and exploit the arguments of `barplot( )` to make the graph more aesthetically pleasing
7. Make a pie chart based on the same data
8. Create a waffle plot based on the same data
9. Create a single PDF file containing the plots created at points 6, 7 and 8. Each plot should be on a separate page
10. Save each of the plots created at points 6, 7 and 8 in a separate JPG file

Exercise 6

In this exercise we will use the dataset `sat.act` from the `R` package `psych`. The dataset contains the self-reported SAT Verbal, SAT Quantitative and the ACT scores of 700 subjects which were collected as part of an online personality test. The SAT and ACT are standardized ability tests mostly used in the U.S. for college admissions decisions. Different Students tend to do better on one test over the other.

1. Install the package `psych`, load the package and load the dataset `sat.act`.
2. Read the help page of the dataset (`?sat.act`) to check the range of the scores ACT, SATV and SATQ and depending on the range, choose 2 scores you want to visualize together in a plot.
3. Use `density()` to visualize the nonparametric density estimate of the first score and use `lines()`to add the density of the second score to your plot.
4. Remove the top and right borders of the graph, give your plot a title and give the two score densities different colours.

We will now add the titles of the variables to their respective density graphs with the help of the `locator()` function. Next lecture, you will learn how to add a legend to your graphs.

5. Copy and paste the code for your graph to the console, add a line with `locator(2)`, and press enter. Then go to the graph, and click on the point in the graph where you want the variable name for the density line to be. Do this for both variables. You will get the coordinates of these spots. Add the variable names to the graph with the function `text()`, and use the coordinates you find to give the labels the right positions.
6. Change the size of your text by varying the `cex` argument and change the colour of the text to the same colour as the density estimate.