SDA Group Submission Assignment 1

CD

For your group submission, upload one .pdf in Canvas. Please refer to the R Markdown cheat sheet and the R Markdown Submission template (to be found in Canvas) to get started with R Markdown 😀

If you follow all installation instructions given in Canvas and still have big issues with R Markdown installation, you can exceptionally submit a .pdf generated in a different way (e.g., by copying code 'manually' into a text editor), which should still include all relevant explanations, code snippets and graphics.

Always remember that there are several ways to code something, and that the useful `R` functions provided for each task are just some suggestions (that is, you can use your favorites even if they are not listed). Finally, the symbol ϟ indicates something generally worthy of attention. Have fun!

Exercise 1

a. Write a function `CTL_unif(n, m)` that

   - draws `n` samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, each consisting of `m` independent draws from $\mathcal{U}(0,1)$ (which denotes the uniform distribution on the interval $(0,1)$);
   - computes (and returns) the mean of each sample.

   Hint: you may want to look into `replicate`(see the help page by executing `?"replicate"`) to repeatedly execute function calls/evaluate expressions.

b. Set a seed of your choice (for reproducibility) and plot the following side-by-side:

   - an histogram (scaled to density) of the means returned by calling `CTL_unif(n = 500, m = 30)`, with the theoretical normal density of $\mathcal{N}\left(\frac{1}{2}, \frac{1}{12m}\right)$ (which denotes the normal distribution with mean $1/2$ and *variance* $1/(12m) = 1/(12 \cdot 30)$) superimposed as a colorful line;
   - an histogram (scaled to density) of the means returned by calling `CTL_unif(n = 500, m = 200)`, with the theoretical normal density of $\mathcal{N}\left(\frac{1}{2}, \frac{1}{12m}\right)$ (which denotes the normal distribution with mean $1/2$ and *variance* $1/(12m) = 1/(12 \cdot 200)$) superimposed as a colorful line.

   The x-axis interval should be $(0.3, 0.7)$ in both histograms, while the y-axis limits should be set in such a way that nothing is cropped. Remember to give each histogram a title (with `main = ...`) and proper x-axis label (with `xlab = ...`).

   Hint: look into the help page `?"hist"` (especially under Arguments). For graphics, you can use `par(mfrow = c(i, j))` to plot a $i \times j$ grid of graphs. To superimpose a normal density, `curve(dnorm(x, ...), ..., add = TRUE)` can be used (recall that the first argument of `curve` has to be a function of `x`, see `?"curve"`). Alternatively/more generally, `lines(...)` can be use to add a line to an existing plot. ϟ Always recall that `R` functions such as `dnorm` ask you to specify the *standard deviation* of a normal distribution, not its variance.

c. Comment on what you observe in the graphs of b.

Deliverables

- For all subtasks, to aid in correction: all relevant code
- Your code for the function in a.
- The graphical output required by b. (the .pdf should show the code that generated it)

- A written comment for c.

Useful `R` functions

`runif`, `replicate`, `apply`, `mean`, `set.seed`, `par(mfrow = c(1, 2))`, `hist`, `curve`, `dnorm`, `lines`, ...

Exercise 2

Load the built-in data set `airquality` in `R` and inspect it using

```
data("airquality")
head(airquality)
```

a. As we can see from the first few rows of the data frame, there are missing values denoted by `NA`. To check for missing values in column `Ozone`, is it a good idea to use `airquality$Ozone == NA`? Write a short comment explaining why or why *not*. Then, calculate the proportion of missing values in column `Ozone`. Finally, you should compute the average of the (available) values in this column: which extra argument passed to function `mean` ensures a numerical output?

   Hint: in `R`, a `data.frame` (such as `airquality`) can behave as a `list` or as a `matrix`, in the sense that its colums can be extracted 'by name' using the operator `$` or 'by number' using square brackets. To learn about function arguments, their defaults etc. you should always make use of the help pages.

b. Construct a new data frame called `airquality_clean` from `airquality` by:

   - removing every row in `airquality` which contains at least one missing value;
   - removing columns `Month` and `Day`. For each of the variables in `airquality_clean`, construct a numerical summary and plot a boxplot.

   Hint: `summary` can be applied to a data frame, in which case it will act on all columns 'simultaneously': this might be result in a more compact and elegant output then applying the function on each column separately. ϟ `boxplot` has a similar behavior when applied to a data frame, however it then plots the same y-axis for all variables. Thus, if the measurement unit and/or the scale of the variables in a data frame are very different, it is better to plot each variable separately. Boxplots can also be put next to each other using `par(mfrow = c(i, j))`, and a title should be added for each one using `main = ...`

c. What is the correlation between `Ozone` and `Temp`? Does it change if you consider the square root of variable `Ozone` instead (keeping `Temp` as is)? Plot the two corresponding scatterplots side by side, and discuss which one more closely indicates a linear relationship between the variables?

   Hint: use `airquality_clean` here to avoids NAs in correlations.

Deliverables

- For all subtasks, to aid in correction: all relevant code
- Your code to perform the calculations required by a. and c., as well as the transformation in b.
- The graphical outputs required by b. and c.
- Short answers to `R`-related questions asked within the exercise can also be included as `R` comments using `#` (as long as you show the relevant code and comments in the .pdf, of course). Interpretations and longer answers should rather be included in the main text.

Useful `R` functions

`head`, `is.na`, `na.omit`, `complete.cases`, `summary`, `boxplot`, `cor`, `plot`, ...