MT5763: Software for Data Analysis
Group project

Rachel Sippy and Lindesay Scott-Hayward

08 November 2022

Housekeeping

- **PLEASE** read the coursework instructions carefully *before* you start (probably more than once!).
- This group project comprises **30%** of your overall module mark.
- This is a **group** project. The submitted coursework should reflect the work of you as a group. Suspected cases of copying *across* groups will be taken very seriously, so **please** adhere to the University's guidelines on good academic practice. If you have any uncertainties or questions about this, please contact us.
- You are free to divide the work as you see fit, but ultimately **all** tasks need to be reviewed and discussed by **everyone** in the group. Each person should be able to explain in detail how the tasks in the assignment were answered.
- Each member of the group will be awarded the **same** mark. It's imperative that everyone contributes their fair share to the project. If this is not the case, then the project will be further assessed through individual Q & A sessions.
- We recommend you attempt every part of the assignment; even if you do not complete everything, marks are likely to be awarded for incomplete tasks / code. Remember, we cannot allocate marks to a blank sheet of paper, so help us to help you.
- **All** of the tasks / analysis should be completed in R or SAS as instructed. There should be **no** manual manipulation of files / datasets.

Submission

- You are required to upload to Moodle a **single** file - a compiled / knitted R Markdown **group** report, either as a PDF or HTML. Name the file `MT5763_<GroupName>`, where `<GroupName>` is the name of your group e.g. `MT5763_Fife.pdf` or `MT5763_Fife.html`.
- **Each** group member has to submit the *same* / *identical* report (for redundancy and auditing purposes).
- The "deliverables" section for each task will instruct you about what to include in your report. **Please** read these sections carefully.
- Deadline is **Friday, 9th December 2022, 23:59 (UK time)**. **PLEASE** do not leave it to the last minute to upload your work.
- The School has a lateness policy. The standard policy is an initial penalty of 15% of the maximum available mark, then a further 5% per 8-hour period, or part thereof.

Technical advice

SAS code

You *can* run SAS code chunks within R Markdown (see here for details). However, this may not be straightforward to set up (especially for those running SAS through a virtual machine). Instead, you can include SAS code in your report by enclosing it within three backticks. For example:

````
```
proc export data=work.plantdata
    dbms=csv
    outfile=reffile;
run;
```
````

will render as:

```
proc export data=work.plantdata
    dbms=csv
    outfile=reffile;
run;
```

Although this does *not* produce a fully reproducible document, it's good enough for the purpose of this group project.

Figures / images

Sometimes you may want to include a figure / image which was generated externally (e.g. by some SAS code, screenshot). You can use the following `knitr` function:

```
knitr::include_graphics("path/to/your/image")
```

Marking guidance

- Task 1: Shiny app (R) - 40%
- Task 2: Bootstrap (SAS) - 25%
- Task 3: Jackknife (SAS) - 35%

You will be assessed on the following criteria:

- Successfully answering the tasks and adhering to the specifications set out in each problem.
- Providing the *deliverables* asked for.
- Clear documentation of what you have done, together with the relevant analysis and interpretation of the results.
- Code that is readable, logical, reproducible, tidy and appropriately commented.
- Appropriate use of version control.

Assignment

Version control

- Create the following **private** GitHub repositories (**ONE** per group - up to you to decide who will host what):
  - `MT5763_<GroupName>`: For the group report, where `<GroupName>` is the name of your group.
  - `MT5763_Shiny`: For the Shiny app task.
- Make sure to **only** version control files that need to be tracked.
- Make sure you **commit** changes often and include a succinct commit message that clearly describes what you changed and why. You will **not** be penalised for committing changes to fix mistakes in your code - this is one of the reasons why version control is used!
- *Before* submitting your coursework invite us as a collaborator to your repos so that we are able to access them. Instructions can be found here. Our usernames are `rsippy` and `lindesaysh`.
- Include a link to your GitHub repositories in your report.

Task 1: Shiny app (R)

- Create a Shiny app that displays web-scraped or API-accessed real-time data that is refreshed every hour[1] or when the user presses a "Refresh" button. The user should be able to download the currently displayed data by pressing a "Download" button. You are free to add other ways for the user to interact with the app.
- The data could be **anything** (within reason) that your group decides upon. For example:
  - Number of Covid-19 cases / deaths.
  - Tweets from your favourite celebrity.
  - Sports scores.
  - News feeds.
  - Satellite imagery.
  - Stock markets.
  - etc.
- The app needs to be user-friendly.
- Include a `Readme.md` and a `DESCRIPTION` file.
- Modularise your code as much as possible (use functions and `global.R` where possible).
- Publish the app on https://www.shinyapps.io/

Deliverables

- Provide a link to the GitHub repository for the app and a URL of the published app (hosted on https://www.shinyapps.io/)[2].

[1] *Hint*: Check out the `reactiveTimer` function.
[2] This URL can also be included in your GitHub repository homepage via the `Readme.md` file.

- Provide a description of the app - what it does and how users can interact with it. Include screenshot(s) to help with your explanation or embed the app within your document using the `knitr::include_app("URL of hosted app")` function.
- Do **NOT** include your Shiny app code in the report.

Task 2: Bootstrap (SAS)

- Create a faster equivalent of the bootstrapping macro `regBoot.sas`. It only needs to work for one covariate.
- Use the seals data (`seals.csv`) to perform a regression of testosterone level (in $\mu g/l$) on length (in cm). This is a fictional dataset of male hormone levels in seals of different lengths.
- State and visualise the 95% confidence intervals for the estimates of each parameter (intercept and slope). Provide a histogram for the distribution of each bootstrapped parameter.
- Compare `regBoot.sas` to your modified version to determine the speed-up.
- Compare the boostrapped parameter estimates and their 95% confidence intervals to those obtained using the built-in SAS procedure.

Deliverables

- Show your code and visualised results.
- Discuss and interpret your comparative analysis.

Task 3: Jackknife (SAS)

Jackknife is another computer intensive method used for variance and bias estimation (it pre-dates the bootstrap method). It was developed by John Tukey in the 1950s from an idea by Maurice Quenouille. Whilst in bootstrapping, the observed data (temporarily treated as a "population"), is sampled with replacement (to simulate the sampling process), in the jackknife method, each sample consists of all *but* one of the observations (i.e. sampling *without* replacement). More details can be found here.

Thus, n plausible samples are generated from the observed data (which itself is of size n), each having sequentially a single datum removed, i.e. the first jackknife sample is of length $n-1$ with the first data point removed, the second jackknife sample is also of length $n-1$ with only the second datum removed, etc.

The collection of jackknife samples can then be used to compute the statistic of interest, albeit not necessarily in the straightforward manner of the bootstrap. For example, the jackknife estimate of the standard error (SE) of the mean is given by:

$$SE = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\theta_i - \bar{x})^2}$$

where:

- n is the sample size
- $\bar{x}$ is the mean of the original sample
- θ_i is the mean of the i^{th} jackknife sample

Write and implement code (modifying code already given to you in the lecture notes e.g. the two sample randomisation test), to obtain a jackknife estimate for the standard error of the mean for seal body length, using the seals data set (`seals.csv`).

Deliverables

- Include the code used to perform the simulations / calculations, and the results obtained.
- Compare the jackknife estimate to the analytical estimate for the standard error of the mean and discuss.