DSA/ISE-5103

Guidelines for Assignment Reports

R Markdown

- R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, PDFs, html documents, Word files and more (i.e. a cool way to present your reports). For more about it, use this link.
- Headers in Markdown start #, and code starts with "‘. For a summary of the most important commands of R Markdown, search on internet for **R Markdown Cheat Sheet**.
- Compile R Markdown files with the Knit button right above the code editor.

About assignments for DSA/ISE 5103

- **Style matters**. Make it look professional. Be succinct and straightforward. Only show code and results that are valuable to answer the questions or drive your conclusions.
- All assignment submissions need to be in **PDF format** and you have to **attach your R or Rmd code**.
- Include problem numbers in **both** your R/Rmd file and your PDF.
- Your submitted code has to have the following structure: visible problem number, all the R packages used in the assignment at the top of your R script, and properly commented code. Failing to follow this structure will result in grade deduction.
- Read the submission notes at the top of your assignments and follow them.
- Plots should be clear and complete - export from RStudio to include in a word document or compile a document using RMarkdown. That will ensure that the plots have appropriate rendering & resolution in your submission.
- Some assignment reports have a limit of pages. If you exceed them, some points will be deducted.
- Remember that **R is just a tool**. If the primary goal is to demonstrate your knowledge of statistics or dimension reduction, etc., then make sure you highlight what is important and provide your **analysis**.

About this document

This document is for guiding you how to hand in your homework with R Markdown. Some important remarks:

- You will find some useful commands especially in the code chucks. **Read the comments in the code chunks!** They are in the Rmd file that created this PDF.
- This version can be compiled only as a PDF, but the commands in the code chunks work also for Word and HTML.
- If you want to directly create PDFs, you need to download TeX. You can follow this link for instructions. You can also create PDFs by converting from Word to PDF.

Example 1: Using code chuncks the right way

Embedding results in plain text

In many times, the answers for questions are straightforward. You can make the calculations in a chunk and show the results embeded in your plain text paragraphs. Use *include = FALSE* at the top of the code chunk.

Now that you calculated these values, just call it by name of the variable and it will be printed in your report once you compile it. Here an example:

"The computed mean for the attribute hp is 146.69 and its standard deviation is 68.56. From this we can conclude that ..."

Showing only results

If you want to show ONLY the results as R output, do the following (this will not show the R code to make them happen):

```
##    0%   25%   50%   75%  100%
##  52.0  96.5 123.0 180.0 335.0
```

Showing only R code

If you want to show the R code because you think it's important, but not to evaluate it (i.e. nothing happens when running this code chunk), then use *eval = FALSE*:

```
# Set eval to false to avoid evaluation the code chuck
scale_0to1 <- function(x){(x - min(x)) / (max(x) - min(x))} # creating a function to scale from 0 to 1
```

Showing R code along with results

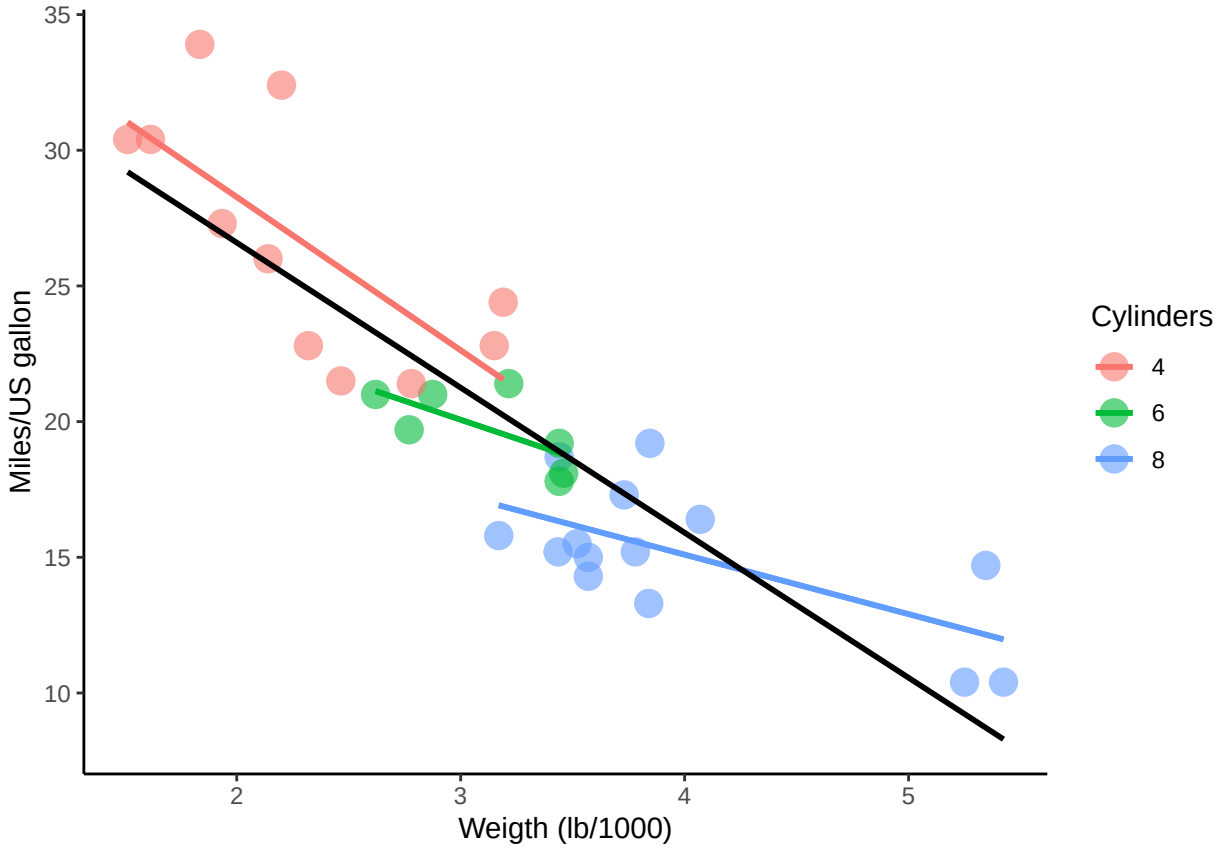
If you consider that some R code and results are needed to answer the question, set echo to TRUE and show it (and do not forget to include some analysis of the results):

```
# By default your code and results will appear
glimpse(mtcars) # command similar to str
```

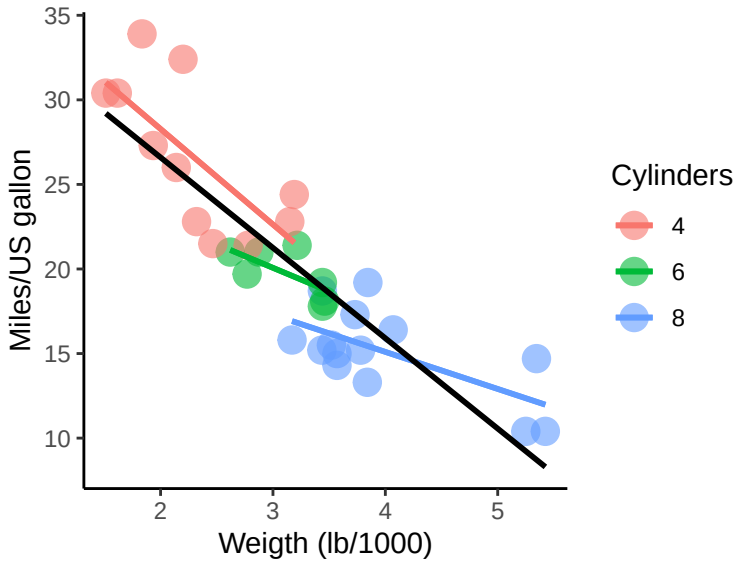
```
## Observations: 32
## Variables: 11
## $ mpg  <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19....
## $ cyl  <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4, ...
## $ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7, 1...
## $ hp   <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, ...
## $ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.9...
## $ wt   <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190, 3...
## $ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00, 2...
## $ vs   <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, ...
## $ am   <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, ...
## $ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, ...
## $ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, ...
```

Example 2: Shrinking plots to make them fit

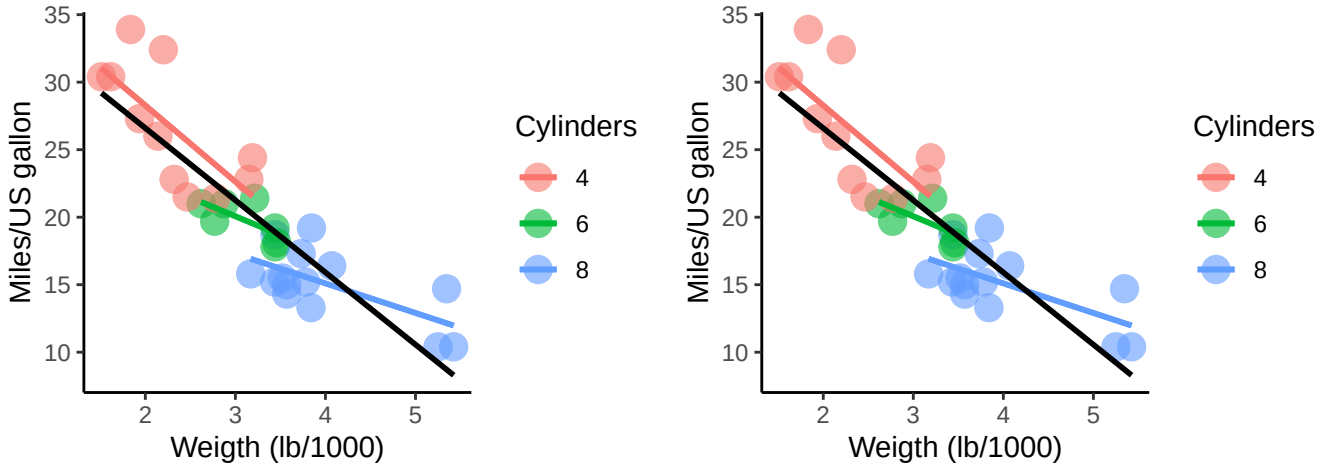
It's a common problem that your plots are going to be too big to look nice in your report. Like the following plot:



You can shrink the plot easily with the *fig* commands, and you'll have set the figure width and height.



We can even make two plots fit in the same space (as two columns) by using the right width and height.



Now that you have the perfect plot in the suitable space, **DO NOT forget to comment on it!** If you don't comment/use them for your analysis, then there is no purpose to keep them in your report.

Final remarks

- Remind, your PDF report DOES NOT have to contain unnecesary code. Save space for your analysis and results that help your analysis.
- Each piece of results or plot that you are presenting in the report should come along with some analysis.
- If you have several interesting results, use a table or something creative to show them. Search on internet table tools like this.
- Do not forget to submit your R code along with your report. If you used R Markdown, you are good to upload your .Rmd file instead of the .R file as long as it has all the code you used.

Thanks for reading.