Practice Assignment 1 Memorandum

Sean van der Merwe

2022-01-17

# Objective

Your goal with this assignment is to reproduce this document *as closely as possible* using R Markdown, but with one major exception: **your document must also show the R code that produces the R output before each set of R output**.

## Details

Begin by creating a new R Markdown file in RStudio (File -> New File -> R Markdown). Save this file with a file name that includes your student number.

Copy all the raw text and headings from this document into your new file below the ‘setup’ code chunk.

Format the raw text and headings using markdown formatting. See Help -> Cheatsheets -> R Markdown Cheat Sheet and R Markdown Reference Guide for information on how to do this.

Insert code chunks and write R code to produce the figures and tables. Do not do the figures and tables manually or using copy-paste.

Your code must begin with the following code block that creates a data set. You will then do a linear regression of *y* on *x* and also plot the regression with prediction intervals, as is done in this document.

set.seed(1234)
n <- 123
x <- rgamma(n, 12, 2) - 2
y <- -1 + 2\*x + rnorm(n)\*3/2

**NB: Do this task yourself from scratch urgently as your future assignments will use the knowledge gained from this assignment. Also note that you may be asked in class how you accomplished specific aspects of this task.**

## R Markdown

See <http://rmarkdown.rstudio.com> for a start. R Markdown can be used to create Word files like this one, PDF documents and PDF presentations via LateX, static (fixed) web pages, web presentations, web pages with interactive elements like explorable graphs, or 3D graphs that can be rotated, and live web pages with deep user interaction (like dashboards). R Markdown documents can even be generated interactively based on user inputs.

In the newer versions of R Studio there is even a visual editor that makes R Studio look more like Word with easy buttons to click, although learning the raw markdown will help you more in the long run.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

You can embed an R code chunk using the *Insert new code chunk* button, or Ctrl+Alt+i. Do this now, and include code that fits an ordinary linear model relating *y* to *x* and then produces a summary of the fit. Make sure your code also saves the model and summary in R variables.

((lm(y~x) -> m1) |> summary() -> s1)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
## Min 1Q Median 3Q Max
## -2.9713 -1.2080 -0.0706 1.0524 3.6787
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98976 0.34453 -2.873 0.00481 \*\*
## x 2.02557 0.07875 25.721 < 2e-16 \*\*\*
## ---
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.511 on 121 degrees of freedom
## Multiple R-squared: 0.8454, Adjusted R-squared: 0.8441
## F-statistic: 661.6 on 1 and 121 DF, p-value: < 2.2e-16

Next, use the *kable* command from the *knitr* package to output a neat table of the coefficients section of your summary. The last column (p-values) should be rounded to 4 decimal places, but the other columns only to 2.

s1 |> coefficients() |> knitr::kable(digits = c(2, 2, 2, 4))

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| (Intercept) | -0.99 | 0.34 | -2.87 | 0.0048 |
| x | 2.03 | 0.08 | 25.72 | 0.0000 |

Now use the *predict* function to obtain 95% prediction intervals for new *x* values (say from -1 to 11 or so), but do not show these yet. These intervals must be based on the linear model you fitted earlier.

newx <- -1:11
m1 |> predict(newdata = list(x=newx), interval = 'prediction') -> newpreds

Produce a graph, similar to the one below, showing the data, the fit, and the prediction intervals. **NB: Pick your own aesthetics!** *Every person in the class should have at least one colour and style element different to the rest.* Feel free to use colour packages such as *viridisLite* or *RColorBrewer* for interesting colour themes. ***Add your student number as text to the graph, using the second last digit as the x coordinate and the last digit as the y coordinate.*** [Bonus marks will be awarded for doing the plot nicely in both base R graphics and ggplot graphics.]

par(mar=c(6,5,1,1))
plot(x, y, type='n', main='', xlab='', ylab='y')
grid()
lines(newx, newpreds[,1], lwd=2, lty=1, col=rgb(0, 0.4, 0))
lines(newx, newpreds[,2], lwd=1, lty=2, col=rgb(0, 0.7, 0.4))
lines(newx, newpreds[,3], lwd=1, lty=2, col=rgb(0, 0.7, 0.4))
points(x, y, pch=20, col=rgb(0.6, 0, 0.6))
mtext('x', 1, 2)
legend('bottom', c('Observations ', 'Linear Fit', '95% Interval'), col = c(rgb(0.6, 0, 0.6), rgb(0, 0.4, 0), rgb(0, 0.7, 0.4)), pch=c(20, NA, NA), lwd=c(NA, 3, 3), lty = c(NA, 1, 2), xpd=NA, inset = c(0, -0.4), horiz=TRUE)
text(7, 8, '2012345678')



### Nitty gritty

Make reference to the file ‘wordreference01.docx’ provided on Blackboard in your YAML title block so that the language, margins, and heading colours of your output match this document.

Remember to use markdown to make headings and lists, not manual formatting.

Your document should have your own name and date at the start. Hint: include the date automatically by adding R code in the YAML section.

#### Vector graphics

The default knitting options produce raster (picture) graphs. These may look nice from far but up close they are far from nice. They also have unnecessarily large file sizes. Always ensure that you produce vector graphics instead. Vector graphics store graphs as sets of instructions (not pictures) so that they look good at any level of zoom.

* When knitting to Word on a Windows computer always use the *dev=‘win.metafile’* code chunk option, either locally or globally to ensure that you obtain vector graphics output.
* If you are knitting to Word on Linux you will need to use the *devEMF* package instead.
* If you are knitting to PDF or any LateX format then use *dev=‘pdf’*.
* If you are creating a web page then try *Plotly* graphics or some other web graphics tools.

### Citing

Try to cite all your sources in every assignment if at all possible. You can use markdown citations, or just manually link to the key resources you used. For example, here is a click-able link to [UFS Blackboard](https://learn.ufs.ac.za/).

Below is a code chunk that explains how to cite R itself.

citation()

##
## To cite R in publications use:
##
## R Core Team (2021). R: A language and environment for statistical
## computing. R Foundation for Statistical Computing, Vienna, Austria.
## URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
## title = {R: A Language and Environment for Statistical Computing},
## author = {{R Core Team}},
## organization = {R Foundation for Statistical Computing},
## address = {Vienna, Austria},
## year = {2021},
## url = {https://www.R-project.org/},
## }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.