Assignments 2 and 3

Statistical Thinking 2020

Due 6pm Sunday 8 November 2020.

Background and general instructions

This document contains the instructions for Assignment 2 and Assignment 3, and for students enrolled in ETC2420 and ETC5242.

There are three Parts to this document.

- Part A relates to Assignment **2** for **all students** enrolled in either ETC2420 or ETC5242.
- Part B relates to Assignment **3** for **all students** enrolled in either ETC2420 or ETC5242.
- Part C relates to Assignment **3** but **only** for students enrolled in **ETC5242**.

Both assignments are worth 5% of the final mark, and are group assignments.

A2A3scrip.R

An **R** script file, named **A2A3script.R** is available on the unit Moodle site. This script file contains all of the code chunks shown in this document.

Instructions for completion and submission of Assignment 2

All students enrolled in ETC2420 or ETC5242 are expected to complete Assignment 2.

The Assignment 2 marks will be determined from answers given on the Moodle Quiz activity labelled as **Assignment 2 Submission**. Once you have completed the questions in Part A of this document you can attempt the Moodle Quiz submission.

Note that due to a limitation of the Moodle quiz feature, **ALL students in each group must individually complete the Assignment 2 submission on Moodle**. The final mark for the group will be the average of the individual Moodle quiz Assignment 2 submissions.

Students do not need to have completed Parts B or C shown in this document in order to complete their submission for Assignment 2.

Instructions for completion and submission of Assignment 3

All students enrolled in ETC2420 or ETC5242 are expected to complete Assignment 3.

- Students enrolled in **ETC2420** to complete Questions from **Part B only** of this document.

- Students enrolled in **ETC5242** to complete Questions from **both Part B and Part C** of this document.

Each group's mark for Assignment 3 will be determined relative to the proportion of marks available for the relevant unit code.

Write your answers in an RMarkdown (.Rmd) file so that it compiles to produce the desired text and calculated results. Take care to write complete sentences, with appropriate punctuation and grammar, and explain any new symbols, code or output produced.

Organise your submission document by question, with individual question numbers indicated through the use of sub-sub-section headings (three hashtags) in your document (e.g. ### Question 10) to ensure adequate separation between questions. Insert your response to (or code chunk for) each numbered question in a space immediately following the identifying question number.

The questions themselves should not be restated in your submission document, however your answer must clearly identify the information it is attempting to provide.

Write in concise but complete sentences. Define any abbreviations used unless already defined in the question.

Remember to ensure your plots have suitable titles and axis labels. It may be helpful to adjust certain *theme*() settings, e.g. axis.title.x, base_size, etc.

The Group Number, and all group member names and ID numbers must be stated in the YAML section of the Rmarkdown file and appear on the accompanying .pdf produced from the .Rmarkdown file.

Show and evaluate all code chunks in your submission file using the 'echo=TRUE' and 'eval=TRUE' chunk options, and format the output so that it does not run off the page when printed. This can be achieved by setting the **global** *knitr* options as shown in the code chunk below, to be included in the first code chunk of your .Rmd file:

```
knitr::opts_chunk$set(echo = TRUE, eval = TRUE, warning = FALSE,
    message = FALSE, error = FALSE, tidy.opts = list(width.cutoff = 60),
    tidy = TRUE)

options(digits = 3)
```

As shown above, please also set the diplayed number of significant digits equal to three, however when reporting coefficients no more than three nonzero decimal places are required. (For example, if a coefficient is given in **R** as 23.12345, write 23.123 in your report, or if a coefficient is given in **R** as 0.0056789, write 0.00568 in your report.)

Once all tasks are completed, one member of each group will need to upload **two (2)** separate files to the relevant unit code labelled **Assignment 3 Submission** upload link on the unit Moodle page. These files will have names such as:

1. **GroupNumber_A3.Rmd**
2. **GroupNumber_A3.pdf** (obtained by first rendering to .html and then printing to .pdf)

Only one submission of Assignment 3 is required for each group.

Note that students are also not required to complete Part A as part of their submission of Assignment 3, except for the inclusion of those parts of the **R** code required in order to complete the questions from Assignment 3.

Required R packages

There are several **R** packages required for these assignments, as listed below. Students unfamiliar with any of these packages, or the functions used, are recommended to review the relevant package or function documentation.

```
library(tidyverse)
library(bayess)
library(broom)
library(car)
library(GGally)
library(meifly)
```

Additional packages may be included to facilitate the formatting of Assignment 3 submissions, if desired.

The dataset for both Assignment 2 and Assignment 3

This assignment uses the *caterpillar* dataset from the **bayess** package.

This data set is concerned with y, representing the (natural) logarithm of the average number of nests of caterpillars per tree in an area of 500 square meters. In Part B, we seek to "explain" the value of y using the single regressor variable $x1$, the altitude (in meters) of each given geographical area. In Part C, all available regression covariate variables, $x1$ through $x8$, as defined in the **R** help(caterpillar) file), are considered.

Part A: Simple linear regression

- Read the **bayess::caterpillar** dataset into **R**, and save it as a tibble named *cat*.

```
data(caterpillar)
cat <- as_tibble(caterpillar)
```

- Review the sample distribution of the response variable y, and review the *caterpillar* data help to familiarise yourself with the other variables in *cat*.

1. [10 marks] Produce a table of sample summary statistics relating to y, containing (at least) the mean, median, standard deviation, standard error, interquartile range, first quartile, third quartile, minimum, maximum. Comment on any important features of this distribution.
2. [5 marks] Use the *lm*() function to find the ordinary least squares fit of the simple linear model given by
$$y_i = \beta_0 + \beta_1 \; x_{1,i} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots, n,$$
and report your fitted regression line.

- Store the output from the *lm*() function in an **R** object named **model1**, for use in subsequent question parts.
- Create and review the objects named **tidy1**, **glance1** and **augment1** as per the code chunk below.

```
tidy1 <- tidy(model1)
glance1 <- glance(model1)
augment1 <- augment(model1)
```

3. [10 marks] Report the CLT-based 95% confidence intervals for each of the regression coefficients of **model1**[1], β_0 and β_1. Provide an appropriate explanation of what these intervals represent.

4. [10 marks] Report the R-squared value produced by the fit of **model1**, and interpret the numerical value produced from **model1**.

5. [10 marks] Why does a large *leverage* value, or a large *Cook's D* value, cause concern?

6. [10 marks] Let $(x_{1,i}, y_i)$ denote the pair of values of $x1$ and y contained in the i^{th} row of the tibble named *cat*. Identify the points, $(x_{1,i}, y_i)$, for which *either* the leverage exceeds the threshold value, *or* Cook's D exceeds the threshold value, *or both*. Comment on whether this information suggests any action should be taken.

7. [15 marks] Describe in detail three different (types of) plots involving the residuals of **model1** that would be suitable to produce and use these plots to determine whether the model is adequate.

8. [15 marks] Review the code chunk shown below. Then describe what the contents of the **R** object *df* represent, and explain how to these values may be used to produce approximate 95% confidence intervals for each of β_0 and β_1.

```
R <- 1000
n <- nrow(cat)

df <- tibble(b0 = rep(0, R), b1 = rep(0, R))

set.seed(2020)
for (j in (1:R)) {
    temp <- cat %>% slice_sample(n = n, replace = TRUE)
    temp1 <- lm(formula = y ~ x1, data = temp)
    tidytemp1 <- tidy(temp1)
    df[j, ] <- t(tidytemp1$estimate)
}
```

9. [15 marks] Assuming that **model1** is suitable for modelling y, **explain** how to implement a *lower one-sided* permutation-based (randomisation) test for the null hypothesis H_0: $\beta_1 = 0$, where the test has a 5% significance level. (You are not required to actually implement this test.[2])

Part B: Multiple Linear Regression

For Assignment 3 you will attempt to find a better model than **model1** for y, using some or all of the available regressors in the *caterpillar* dataset. The biggest model we will consider, referred to as the *full model* and labelled as **modelf**, is given by

$$y_i = \beta_0 + \sum_{j=1}^{8} \beta_j \, x_{j,i} + \varepsilon_i, \quad \text{for } i = 1, 2, \ldots, n.$$

Together with **modelf**, we will consider all of the $2^{(8-1)} - 1$ models that include an intercept term and *at least* one additional linear term involving one of the eight available regressors in $\{x1,\ x2,\ x2,\ x4,\ x5,\ x6,\ x7,\ x8\}$.

[1] For these confidence intervals to be approximately valid, we are effectively assuming that **model1** is appropriate for this data, but we will ignore this fact for the current purpose.

[2] Note there was a small typo in the statement of the alternative hypothesis on slide 31 of the Week 9 slides, which has since been corrected. As stated, the description of the permutation test procedure relates to an *upper* one-sided test, with H_1: $\beta_k > 0$.

- Use the *ggscatmat*() function from the **GGally** package to generate a scatterplot matrix (with extended information) for the complete set of regressors $x1$ through $x8$ and the response variable y, as shown in the code chunk below.

```
ggscatmat(cat, columns = c(1:9)) + theme(geom.text.size = 7,
    strip.text.x = element_text(size = 12)) + theme_bw(base_size = 7)
```

10. [10 marks] Describe the information shown in the *ggscatmat*() output

    i) in the lower triangular part of the "matrix",

    ii) in the upper triangular part of the matrix, and
    iii) down the main diagonal.

11. [5 marks] Use the **car::vif** function to calculate the *variance inflation factors* (VIFs) associated with the explanatory variables in **modelf**.

12. [10 marks] Explain why it is desirable to identify and remove from consideration any regressor with VIF greater than 10.

- Use the code chunk below to fit all viable multiple linear regression models under consideration.[3] This collection of all fitted models is referred to as an *ensemble*.

```
quiet <- function(x) {
    sink(tempfile())
    on.exit(sink())
    invisible(force(x))
}

all_mod <- quiet(fitall(y = cat$y, x = cat[, -c(6, 9)], method = "lm"))
summary(all_mod)
```

13. [5 marks] How many model fits does the above *all_mod* object contain? Explain how you determined this number. Then save this number in an **R** object named **nmod**.

- Run the code below to produce a panel of visualisations [4] of these model performance measures.[5] In each panel, a single point is shown indicating the relevant performance measure for each model in the ensemble of (viable) models. While not explicitly reporting the model with the highest measure, the plots shown do give information about the similarity of these values across the ensemble of models, and give an impression of the improvements that can be achieved through the inclusion (or removal) of additional regressors.

[3] The output to screen that the *fitall*() function normally produces is suppressed here by the "quiet" function: see this R help post.

[4] The summary output produced from the above code chunk includes the log-Likelihood, AIC, BIC R-squared, and adjusted R-squared values. It is useful to visualise these values according to the number of regressors (including the intercept).

[5] The summary output produced from the above code chunk includes the log-Likelihood, AIC, BIC R-squared, and adjusted R-squared values. It is useful to visualise these values according to the number of regressors (including the intercept). Note that *negAIC* and *negBIC* are plotted, rather than AIC and BIC, respectively. These values are simply the *negative* of AIC and BIC values, respectively. They reveal the same information as the AIC and BIC measures, but are *positively oriented*.

```
all_mod_s <- all_mod %>% map_df(glance) %>% mutate(model = nmod) %>%
    mutate(negBIC = -1 * BIC, negAIC = -1 * AIC)

label <- NULL
for (i in nmod) {
    l <- as.character(summary(all_mod[[i]])$call)[2]
    label <- c(label, substr(l, 5, str_length(l)))
}

all_mod_s_long <- all_mod_s %>% gather(fit_stat, val, adj.r.squared,
    negAIC, negBIC, logLik, r.squared) %>% group_by(fit_stat,
    df) %>% mutate(rank = min_rank(desc(val)))

p1 <- ggplot(all_mod_s_long, aes(df, val)) + geom_point() + geom_line(data = filter(all_mod_s_long,
    rank == 1)) + facet_wrap(~fit_stat, ncol = 5, scales = "free_y") +
    xlab("Number of regressors (including intercept)") + ylab("Values") +
    theme_grey(base_size = 10)

p1
```

- Next, run the code chunk below to extract the model associated with the maximum value of each of the five (positively oriented) performance measures.

```
print("Adjusted R-squared")
indexadjRsq <- c(1:nmod)[all_mod_s$adj.r.squared == max(all_mod_s$adj.r.squared)]
indexadjRsq
max_adjRsq <- all_mod[[indexadjRsq]]
max_adjRsq

print("log-Likelihood")
indexlogLik <- c(1:nmod)[all_mod_s$logLik == max(all_mod_s$logLik)]
indexlogLik
max_logLik <- all_mod[[indexlogLik]]
max_logLik

print("Negative AIC")
indexAIC <- c(1:nmod)[all_mod_s$negAIC == max(all_mod_s$negAIC)]
indexAIC
max_AIC <- all_mod[[indexAIC]]
max_AIC

print("Negative BIC")
indexBIC <- c(1:nmod)[all_mod_s$negBIC == max(all_mod_s$negBIC)]
indexBIC
max_BIC <- all_mod[[indexBIC]]
max_BIC

print("R-squared")
indexRsq <- c(1:nmod)[all_mod_s$r.squared == max(all_mod_s$r.squared)]
indexRsq
max_Rsq <- all_mod[[indexRsq]]
max_Rsq
```

14. [20 marks] Based on the information available and considering only the five identified potentially "best" multiple linear regression models in the ensemble, one each corresponding to the five performance measures, which multiple linear regression model do you prefer, and why? Describe the process used to determine your *preferred model*, and denote this by **modelp**. Incorporate in your discussion all relevant plots and summary statistics that you used to make your decision.

15. [5 marks] Report the estimated final form of your preferred model, provide CLT-based confidence intervals for all unknown regression coefficients, and report the estimated standard deviation of the residuals.

16. [10 marks] The code chunk below generates a set of $R = 1000$ values from the bootstrap sampling distribution of the regression coefficients from **modelf**. Modify and extend this code to produce 95% bootstrap confidence intervals for each of the regression coefficients of your preferred model. Be sure to keep the same basic code structure, number of bootstrap replications, and random seed value. Report the resulting confidence intervals for each coefficient and provide a suitable interpretation of these quantities.

```
modelf <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8, data = cat)
tidyf <- tidy(modelf)

R <- 1000
n <- nrow(cat)

R_coeffs <- tibble(b0 = rep(0, R), b1 = rep(0, R), b2 = rep(0,
    R), b3 = rep(0, R), b4 = rep(0, R), b5 = rep(0, R), b6 = rep(0,
    R), b7 = rep(0, R), b8 = rep(0, R))

set.seed(2020)
for (j in (1:R)) {
    temp <- cat %>% slice_sample(n = n, replace = TRUE)
    tempf <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8, data = temp)
    tidyf <- tidy(tempf)
    R_coeffs[j, ] <- t(tidyf$estimate)
}
```

Part C: Additional Questions for ETC5242 Groups

17. [20 marks] In relation to your preferred model, implement a *two-sided* permutation-based test of H_0: $\beta_1 = 0$ with a 5% significance level, again using **setseed(2020)**. Briefly explain the rationale for this test, report your results and provide a suitable justification for your conclusion.

18. [15 marks] Explain what is the *Leave One Out Cross Validation* statistic, denoted as *LOOCV*, calculate its value in association with **modelf** and with your preferred model. Provide a brief discussion comparing the two results.