STAT 471/571/701 Modern Data Mining - HW 1

Group Member 1
Group Member 2
Group Member 3

Due: September 17, 2017

Overview / Instructions

All the works submitted should be done through r markdown format. Find RMarkdown cheat sheet here. For those who have never used it before we urge you to start this homework as soon as possible.

This is homework #1 of STAT 471/571/701. It will be *due on September 17, 2017 by 11:59 PM* on Canvas. You can directly edit this file to add your answers. **Submit a zip file containing the Rmd file, a PDF or HTML version, and all data files necessary with only 1 submission per HW team**. If you intend to work on separate problems separately, compile your answers into 1 Rmd file before submitting. Additionally, ensure that you can 'knit' or compile your Rmd file. It is also likely that you need to configure Rstudio to properly convert files to PDF. **These instructions** should be helpful.

In general, be as concise as possible while giving a fully complete answer. All necessary data is available in the `Data` folder on Canvas. Make sure to document your code so the teaching fellows can follow along. R Markdown is particularly useful because it follows a 'stream of consciousness' approach: as you write code in a code chunk, make sure to explain what you are doing outside of the chunk.

Remember that the Code of Academic Integrity strictly applies to this course. Any questions you have on the homework should be directed to Piazza. If you have questions that would reveal part of the solution, ask them in 'private to instructors' mode.

Solutions will be posted after the deadline. Make sure to compare your answers to and understand the solutions.

Question 0

Review the code and concepts covered during lecture.

EDA

Question 1: Exploratory Data Analysis with Sirius XM

This question is about estimating audience size and is designed as a tutorial on the data exploration process of data cleaning, data summary and data visualization. No formal statistical inference is necessary for this question. First time R users may want to defer or skip this question.

Background: Wharton launched a talk show called "Business Radio Powered by the Wharton School" through the Sirius Radio station in January of 2014. Within a short period of time the general reaction had been overwhelmingly positive. To find out the audience size for the show, we designed a survey and collected a data set via MTURK in May of 2014. Our goal was to estimate the audience size. There were 51.6 million Sirius Radio listeners then. One approach is to estimate the proportion of the Wharton listeners to that of the Sirius listeners, p, so that we will come up with an audience size estimate of approximately 51.6 times p.

To do so, a simple survey was launched via Amazon Mechanical Turk (MTurk) on May 24, 2014 and we set it to be run for 6 days with a target maximum sample size of 2000 as our goal. Most of the observations came in within the first two days. The main questions of interest are "Have you ever listened to Sirius Radio" and "Have you ever listened to Sirius Business Radio by Wharton?". A few demographic features used as control variables were also collected; these include Gender, Age and Household Income.

We requested that only people in United States answer the questions. Each person can only fill in the questionnaire once to avoid duplicates. Aside from these restrictions, we opened the survey to everyone in MTurk with a hope that the sample would be more randomly chosen.

The raw data is stored as `Survey_results_final.csv` on Canvas.

Q1.1

Load the data into R.

```
# radio <- ...
```

For each of the following 2 questions, there is a `dplyr` solution and a `base` R solution. Provide *both* ways of doing so.

i. We need to clean and select only the variables of interest. Select only the variables Age, Gender, Education Level, Household Income in 2013, Sirius Listener?, Wharton Listener? and Time used to finish the survey.

```
# your R code goes here
```

ii. Change the variable names to be "age", "gender", "education", "income", "sirius", "wharton", "worktime".

```
# your R code goes here
```

Q1.2

As in real world data with user input, the data is incomplete, missing values, and has incorrect responses. There is no general rule for dealing with these problems beyond "use common sense." In whatever case, explain what the problems were and how you addressed them. Do not use Excel, however tempting it might be.

Tip: reflect on the reasons for which data could be wrong or missing. How would you address each case? For this homework, if you are trying to predict missing values with regression, you are definitely overthinking. Keep it simple.

Q1.3

Write a brief report to summarize all the variables collected. Include both summary statistics (including sample size) and graphical displays such as histograms or bar charts where appropriate. Comment on what you have found from this sample. (For example - it's very interesting to think about why would one work for a job that pays only 10cents/each survey? Who are those survey workers? The answer may be interesting even if it may not directly relate to our goal.)

Q1.4 Sample property questions

i. Does this sample appear to be a random sample from the general population of the USA?
ii. Does this sample appear to be a random sample from the MTURK population?

iii. Assume that the proportion of Wharton listeners vs. that of Sirius listeners remains the same in the general population as it is in the MTURK population. Use the data to provide an estimate of the number of Wharton listeners in the USA. In order to make this estimate do you need to break down the proportion of Wharton to Sirius by gender (or by income.)? Provide some graphical or numerical evidence to support your reasoning.

Q1.5

4. Give a final estimate of the Wharton audience size in January 2014. Assume that the sample is a random sample of the MTURK population, and that the proportion of Wharton listeners vs. Sirius listeners remains the same in the general population as it is in the MTURK population. Briefly summarize your findings and how you came to that conclusion.

Simple Regression

Question 2

This exercise is designed to help you understand the linear model and see everything through simulations.

Presume that x and y are linearly related with a normal error, such that $y = 1 + 1.2x + \epsilon$. The standard deviation of the error is $\sigma = 2$.

Note: we can create a sample input vector ($n = 40$) for x with the following code:

```
x <- seq(0, 1, length = 40)
```

Q2.1

Create a corresponding output vector for y according to the equation given above. Then, create a scatterplot with (x, y) pairs. Base R plotting is acceptable, but if you can, attempt to use `ggplot2` to create the plot.

Q2.2

Find the LS estimates of β_0 and β_1, using the `lm()` function.

Q2.3

Overlay the LS estimates onto a copy of the scatterplot you made above.

Q2.4

What is the 95% confidence interval for β_1? Does this confidence interval capture the true β_1?

Q2.5

What is your RSE for this linear model fit? Is it close to $\sigma = 2$?

Q2.6

This part aims to help understand the notion of sampling statistics, confidence intervals. Let's concentrate on estimating the slope only.

Generate 100 samples of size $n = 40$, and estimate the slope coefficient from each sample. We include some sample code below, which should aim you in setting up the simulation. Note: this code is written clearly but suboptimally; see the appendix for a more R-like way to do this simulation.

```r
x <- seq(0, 1, length = 40)
n_sim <- 100
b1 <- numeric(n_sim)   # nsim many LS estimates of beta1 (=1.2)
upper_ci <- numeric(n_sim)  # lower bound
lower_ci <- numeric(n_sim)  # upper bound
t_star <- qt(0.975, 38)

# Carry out the simulation
for (i in 1:n_sim){
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  lse <- lm(y ~ x)
  lse_out <- summary(lse)$coefficients
  se <- lse_out[2, 2]
  b1[i] <- lse_out[2, 1]
  upper_ci[i] <- b1[i] + t_star * se
  lower_ci[i] <- b1[i] - t_star * se
}
results <- cbind(se, b1, upper_ci, lower_ci)
rm(se, b1, upper_ci, lower_ci, x, n_sim, b1, t_star, lse, lse_out)
```

i. Summarize the LS estimates of β_1 (in the above, `sim_results$b1`). Does the sampling distribution agree with the theory?
ii. How many times do your 95% confidence intervals cover the true β_1? Display your confidence intervals graphically.

Multiple Regression

Question 3:

Auto data from ISLR. The original data contains 408 observations about cars. It has some similarity as the data CARS that we use in our lectures. To get the data, first install the package ISLR. The data Auto should be loaded automatically. We use this case to go through methods learnt so far.

You can access the necessary data with the following code:

```r
# check if you have ISLR package, if not, install it
if(!requireNamespace('ISLR')) install.packages('ISLR')
auto_data <- ISLR::Auto
```

Get familiar with this dataset first. You can use `?ISLR::Auto` to view a description of the dataset.

Q3.1

Explore the data, with particular focus on pairwise plots and summary statistics. Briefly summarize your findings and any peculiarities in the data.

Q3.2

What effect does time have on MPG?

i. Start with a simple regression of mpg vs. year and report R's `summary` output. Is year a significant variable at the .05 level? State what effect year has on mpg, if any, according to this model.
ii. Add horsepower on top of the variable year. Is year still a significant variable at the .05 level? Give a precise interpretation of the year effect found here.
iii. The two 95% CI's for the coefficient of year differ among i) and ii). How would you explain the difference to a non-statistician?
iv. Do a model with interaction by fitting `lm(mpg ~ year * horsepower)`. Is the interaction effect significant at .05 level? Explain the year effect (if any).

Q3.3

Remember that the same variable can play different roles! Take a quick look at the variable `cylinders`, try to use this variable in the following analyses wisely. We all agree that larger number of cylinder will lower mpg. However, we can interpret `cylinders` as either a continuous (numeric) variable or a categorical variable.

i. Fit a model, that treats `cylinders` as a continuous/numeric variable: `lm(mpg ~ horsepower + cylinders, ISLR::Auto)`. Is `cylinders` significant at the 0.01 level? What effect does `cylinders` play in this model?
ii. Fit a model that treats `cylinders` as a categorical/factor variable: `lm(mpg ~ horsepower + as.factor(cylinders), ISLR::Auto)`. Is `cylinders` significant at the .01 level? What is the effect of `cylinders` in this model? Use `anova(fit1, fit2)` and `Anova(fit2)to help gauge the effect. Explain the difference between`anova()andAnova‘.
iii. What are the fundamental differences between treating `cylinders` as a numeric and or a factor models?

Q3.4

Final modelling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view.

i. Describe the final model. Include diagnostic plots with particular focus on the model residuals and diagnoses.
ii. Summarize the effects found.
iii. Predict the mpg of a car that is: built in 1983, in US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower and weighs 4000 pounds. Give a 95% CI.

Appendix

This is code that is roughly equivalent to what we provide above in Question 2 (simulations).

```
simulate_lm <- function(n) {
  # note: `n` is an input but not used (don't worry about this hack)
  x <- seq(0, 1, length = 40)
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  t_star <- qt(0.975, 38)
  lse <- lm(y ~ x)
  lse_out <- summary(lse)$coefficients
  se <- lse_out[2, 2]
  b1 <- lse_out[2, 1]
```

```r
  upper_CI = b1 + t_star * se
  lower_CI = b1 - t_star * se
  return(data.frame(se, b1, upper_CI, lower_CI))
}

# this step runs the simulation 100 times,
# then matrix transposes the result so rows are observations
sim_results <- data.frame(t(sapply(X = 1:100, FUN = simulate_lm)))
```