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

Group Member 1
Group Member 2
Group Member 3

Due: 11:59PM February 3, 2019

Overview / Instructions

- **Homework assignments can be done in a group consisting of up to three members**. Please find your group members as soon as possible and register your group on our canvas site.
- **All work submitted should be completed in the R markdown format.** You can find a cheat sheet for R Markdown here. For those who have never used it before we urge you to start this homework as soon as possible.
- **Submit a zip file containing the (1) Rmd file, (2) a compiled PDF or HTML version, and (3) all necessary data files. Note: Please only upload ONE submission per HW team**. You can directly edit this file to add your answers. If you intend to work on the problems separately within your group, compile your answers into one Rmd file before submitting. We encourage that you at least attempt each problem by yourself before working with your teammates. 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 datasets are available in the `Data` folder on Canvas. Make sure to document your code with comments 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.
- A few good submissions will be used as sample solutions. When those are released, make sure to compare your answers and understand the solutions.

Question 0

Review the code and concepts covered during lecture.

Simple Regression

Question 1

This exercise is designed to help you understand the linear model using simulations. In this exercise, we will generate (x_i, y_i) pairs so that all linear model assumptions are met.

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

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

```r
# Generates a vector of size 40 with equally spaced values between 0 and 1, inclusive
x <- seq(0, 1, length = 40)
```

Q1.1 Generate data

Create a corresponding output vector for y according to the equation given above. Use `set.seed(1)`. Then, create a scatterplot with (x, y) pairs. Base R plotting is acceptable, but if you can, please attempt to use `ggplot2` to create the plot. Make sure to have clear labels and sensible titles on your plots.

```
# TODO
```

Q1.2 Understand the model

i. Find the LS estimates of β_0 and β_1, using the `lm()` function. What are the true values of β_0 and β_1? Do the estimates seem to be good?
ii. What is your RSE for this linear model fit? Is it close to $\sigma = 2$?
iii. What is the 95% confidence interval for β_1? Does this confidence interval capture the true β_1?
iv. Overlay the LS estimates and the true lines of the mean function onto a copy of the scatterplot you made above.

```
# TODO
```

Q1.3 Model diagnoses

i. Provide residual plot of `x = fitted y`, `y = residuals`.
ii. Provide a QQ-Normal plot of the residuals
iii. Comment on how well the model assumptions are met for the sample you used.

```
# TODO
```

Q1.4 Understand sampling distribution and confidence intervals

This part aims to help you understand the notion of sampling statistics and 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 guide you in setting up the simulation. Note: this code is written clearly but suboptimally; see the appendix for a more optimal R-like way to do this simulation.

```
# Inializing variables. Note b_1, upper_ci, lower_ci are vectors
x <- seq(0, 1, length = 40)
n_sim <- 100              # number of simulations
b1 <- 0                   # n_sim many LS estimates of beta_1 (=1.2). Initialize to 0 for now
upper_ci <- 0             # upper bound for beta_1. Initialize to 0 for now.
lower_ci <- 0             # lower bound for beta_1. Initialize to 0 for now.
t_star <- qt(0.975, 38)   # Food for thought: why 38 instead of 40? What is t_star?

# Perform the simulation
for (i in 1:n_sim){
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  lse <- lm(y ~ x)
  lse_output <- summary(lse)$coefficients
  se <- lse_output[2, 2]
  b1[i] <- lse_output[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)

# remove unecessary variables from our workspace
rm(se, b1, upper_ci, lower_ci, x, n_sim, b1, t_star, lse, lse_out)
```

i. Summarize the LS estimates of β_1 (stored in `results$b1`). Does the sampling distribution agree with theory?

ii. How many times do your 95% confidence intervals cover the true β_1? Display your confidence intervals graphically.

```
# TODO
```

Question 2

This question is about Major League Baseball (MLB) and payrolls. Guiding questions: how do salaries paid to players affect team wins? How could we model win propensity?

We have put together a dataset consisting of the winning records and the payroll data of all 30 MLB teams from 1998 to 2014. There are 54 variables in the dataset, including:

- `payroll`: total team payroll (in $billions) over the 17-year period
- `avgwin`: the aggregated win percentage over the 17-year period
- winning percentage and payroll (in $millions) for each team broken down for each year.

The data is stored as `MLPayData_Total.csv` on Canvas.

```
# TODO
# salary <- ... read in data
```

Q2.1 Exploratory questions

For each of the following questions, there is a `dplyr` solution that you should try to answer with.

i. Which 5 teams spent the most money in total between years 2000 and 2004, inclusive?

ii. Between 1999 and 2000, inclusive, which team(s) "improved" the most? That is, had the biggest percentage gain in wins?

iii. Using `ggplot`, pick a single year, and plot the number of games won vs. `payroll` for that year (`payroll` on x-axis). You may use any 'geom' that makes sense, such as a scatterpoint or a label with the point's corresponding team name.

```
# TODO
```

Q2.2

For a given year, is `payroll` a significant variable in predicting the winning percentage of that year? Choose a single year and run a regression to examine this. You may try this for a few different years. You can do this programmatically (i.e. for every year) if you are interested, but it is not required.

```
# TODO
```

Q2.3

With this aggregated information, use regression to analyze total payroll and overall winning percentage. Run appropriate model(s) to answer the following questions:

i. In this analysis, do the Boston Red Sox perform reasonably well given their total payroll? [Use a 95% interval.]

ii. In view of their winning percentage, how much payroll should the Oakland A's have spent? [Use a 95% interval.]

```
# TODO
```

Multiple Regression

Question 3:

This question utilizes the `Auto` dataset from ISLR. The original dataset contains 408 observations about cars. It is similar the CARS dataset that we use in our lectures. To get the data, first install the package ISLR. The `Auto` dataset should be loaded automatically. We'll use this dataset to go practice the methods learnt so far.

You can access the necessary data with the following code:

```
# Read in the Auto dataset
auto_data <- ISLR::Auto
```

Get familiar with this dataset first. Tip: you can use the command `?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.

```
# TODO
```

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` to your linear model. Is `year` still a significant variable at the .05 level? Give a precise interpretation of the `year`'s 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. Create a model with interaction by fitting `lm(mpg ~ year * horsepower)`. Is the interaction effect significant at .05 level? Explain the year effect (if any).

```
# TODO
```

Q3.3

Remember that the same variable can play different roles! Take a quick look at the variable `cylinders`, and try to use this variable in the following analyses wisely. We all agree that a larger number of cylinders 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()` and `Anova`.

iii. What are the fundamental differences between treating `cylinders` as a continuous and categorical variable in your models?

```r
# TODO
```

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 the following car: A red car built in the US in 1983 that is 180 inches long, has 8 cylinders, displaces 350 cu. inches, weighs 4000 pounds, and has a horsepower of 260. Also give a 95% CI for your prediction.

```r
# TODO
# cars <- ISLR::Auto
```

Appendix

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

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

```r
# 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)))
```