STAT346: Statistical Data Science I

Final: Thursday, Dec 16, 2021, 05:00–06:15 p.m.

Instructions

1. This exam covers material from **Introduction to Data Science**, Chapter 10–16.
2. You may use any books or online resources you want during this examination, but you may not communicate with any person other than your examiner or your TAs.
3. You are required to use the RStudio IDE for this exam.
4. You should work on the provided exam template. When you finalize your exam, you should submit your paper in pdf as well as its .rmd source file. They should have the following name:
   - `stat346_final_yourID.pdf`
   - `stat346_final_yourID.rmd`
5. You should submit your paper no later than 6:20 p.m. After that, there will be a deduction for the late submission (0.5 point per 1 minute). Still you have to sumit your paper by 6:30 p.m.

Problem Set #0 (1 Point)

Run the following code, and show the result.

```
rm(list = ls())
ls()
```

Problem Set #1 (9 Points)

Consider `heights` data set from the `dslabs` package:

```
library(tidyverse)
library(dslabs)
data(heights)
```

(a) [3 points] Compute the sample mean and the sample standard deviation of height of female and male using `group_by` function.

(b) [3 points] First, we make an object `x` which contains a vector of female height.

```
x = heights %>% filter(sex == "Female") %>% pull(height)
```

Define the empirical distribution function (edf). Then provide the chance that a female (selected at random from our data values) is taller than 64.5.

(c) [3 points] Now we consider Monte Carlo simulation where the iteration number is 1000 and set the seed number as 100. Suppose we pick 500 females (whose height distribution is normal with the previous sample mean and sample standard deviation obtained in (a)) at random, then how rare is a height more than 80 inches in a group of 500 females?

Problem Set #2 (16 Points)

For this problem, we use actual polls from the 2016 election. For (a)-(c), we will use all the national polls that ended within one week before the election.

```
library(dslabs)
library(lubridate)
data("polls_us_election_2016")
polls = polls_us_election_2016 %>% filter(enddate >= "2016-10-31" & state == "U.S.")
```

(a) [3 points] For the *second* poll, obtain the sample size `N` and estimated *Trump* proportion `x_hat`. Assume there are only two candidates and let p be the true proportion for Trump. Then construct a 99% confidence interval for the election night proportion p.

(b) [3 points] Use `dplyr` to add a 99% confidence interval as two columns, call them `lower` and `upper`, to the object `polls`. Then use `select` to show the `x_hat`, `lower`, `upper`, and `grade` variables. Save this table as an object `trump`.

The final tally for the popular vote was Clinton 48.2% and Trump 46.1%. We added a column, call it `hit`, to the previous table (object `trump`) stating if the confidence interval included the true proportion $p = 0.461$ (then the value becomes `Include`) or not (then the value becomes `Fail`) using the following code:

```
p = 0.461
trump2 = trump %>%
   mutate(hit = case_when(lower <= p & upper >= p ~ "Include", TRUE ~ "Fail")) %>%
   select(lower, upper, hit, grade)
trump2 %>% head()
```

(c) [7 points] Consider the variable `grade`.

-(c1) [1 pt] Check if `grade` has any missing values.

-(c2) [2 pts] If `grade` has any missing values, we use only observations whose grade is not missing. Now, create a new variable `grade2` where $A+$,A,$A-$ are coded as A, and $B+$,B,$B-$ are coded as B and all the other values are coded as *Others*.

-(c3) [1 pt] Construct two by two table between `hit` and `grade2`.

-(c4) [1 pt] Generate the R-code to perform the Chi-squared test to test the association between `hit` and `grade2` and show the results.

-(c5) [2 pts] Using the Chi-squared test, fill in the blank in the following conclusion:

Since the calculated p-value = _____ is _____ than $\alpha = 0.05$, we _____ the null hypothesis H_0: `hit` variable is not associated with `grade2` variable.

(d) [3 points] Consider `polls_us_election_2016` data. Consider the variable `startdate`, and find the proportion of September 2016.

Problem Set #3 (9 Points)

We use the `Teams`, `Batting` and `Salaries` data by calling `library(Lahman)`:

```
library(Lahman)
library(broom)
data(Teams)
data(Batting)
data(Salaries)
```

For (a)-(b), use the `Teams` data.

(a) [3 point] We use data from 1961 to 2015, and use `BB`, `singles`, `doubles`, `triples`, and `HR` (per game) as explanatory variables to predict `R` (run per game). Please complete the following program first, and provide the fitted regression formula.

```
fit3 = Teams %>% ____________________ %>%
   mutate(BB = BB/G,
          singles = (H-X2B-X3B-HR)/G,
          doubles = X2B/G,
          triples = X3B/G,
          HR = HR/G,
          R = R/G) %>%
   lm(_________________________________)
tidy(fit3, conf.int = T)
```

(b) [3 points] To see how well our fitted regression model actually predicts run, we predict the number of runs for each team in 2016 using the function `predict`. Then calculate the correlation between the predicted run (per game) and the actual number of runs (per game) in 2016. Complete the program.

```
Teams %>% ______________________ %>%
   mutate(BB = BB/G,
          singles = (H-X2B-X3B-HR)/G,
          doubles = X2B/G,
          triples = X3B/G,
          HR = HR/G,
          R = R/G) %>%
   mutate(R_hat = ________________________) %>%
   ___________________________
```

(c) [3points] Join `Batting` for the year 2015 and `Salaries` for the year 2015 by `playerID`. Here we keep the rows in `Salaries` data. Show the first 6 observations.

Problem Set #4 (5 Points)

Load the `admissions` data set, which contains admission information for men and women across six majors and keep only the admitted percentage column:

```
library(dslabs)
data(admissions)
dat <- admissions %>% dplyr::select(-applicants)
```

(a) [2 points] If we think of an observation as a major, `dat` is not tidy. Use the `spread` function to wrangle into tidy shape: one row for each major with two variables (men admitted percentage and women admitted percentage).

(b) [3 points] Now we want to wrangle the `admissions` data. Use the `gather` function to create a `tmp` data frame having 4 columns–`major` and `gender` and two columns containing the type of observation *admitted* or *applicants* and values for such observations. Call the new columns `key` and `value`.