STAT346: Statistical Data Science I

Final: Thursday, Dec 15, 2022, 02:00–03:15 p.m.

Instructions

1. This exam covers course materials (Chapters 11–18).
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 3:20 p.m. After that, there will be a deduction for the late submission (2 points per 1 minute). Still you have to submit your paper by 3:30 p.m.

Problem Set #0 (5 Points)

Run the following code, and show the result.

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

Problem Set #1 (20 Points)

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

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

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

(a) [5 point] We use data from 1961 to 2001, 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.

```
fit = Teams %>% filter(___________________) %>%
   mutate(BB = BB/G,
          singles = (H-X2B-X3B-HR)/G,
          doubles = X2B/G,
          triples = X3B/G,
          HR = HR/G,
          R = R/G) %>%
   lm(___________________________________)
tidy(fit, conf.int = T)
```

(b) [10 points] Suppose we obtained the average number of team plate appearances per game as 38.74. Then we compute the per-plate-appearance rates for players on data from 1998-2001. Then we filter out players with less than 200 plate appearances per year and compute player-specific predicted runs in the object R_hat. Then make a ggplot of R_hat versus R. Interpret the prediction result using R and R_hat for the first player abreubo01.

```
pa_per_game = 38.74
players = Batting %>%
  filter(yearID %in% 1998:2001) %>%
  group_by(playerID) %>%
  mutate(PA = BB+AB) %>%
  summarize(G=sum(PA)/pa_per_game,
    BB = sum(BB)/G,   singles = sum(H-X2B-X3B-HR)/G,
    doubles = sum(X2B)/G,    triples = sum(X3B)/G,
    HR = sum(HR)/G,   AVG = sum(H)/sum(AB),
    PA = sum(PA), R = sum(R)/G) %>%
  filter(__________) %>%
  select(-G) %>%
  mutate(______________________)
head(players)
ggplot(__________________) +
  geom_point()
```

(c) [5 points] Select three variables yearID, playerID and salary in the data Salaries. Then join players for the year 2002 with this data for the year 2002 by playerID. Here we keep the rows in players data. Show the first 6 observations.

Problem Set #2 (40 Points)

In every problems (a)–(e), *set the seed number as 2022* at the beginning.

(a) [5 points] Generate the data consisting of zeroes and ones where we obtain 1 with probability 0.3. The size of the data is 10^5 and we treat this data as our population. Save this population data into the object name `vote`. Check that the true proportion p is 0.295.

(b) [5 points] Take a random sample of size 100 (without replacement) from the `vote`, and calculate the sample proportion.

(c) [10 points] Run a Monte Carlo simulation to confirm that a 90% confidence interval includes p about 90% of the time. For this, repeat (1000 times) the random sampling from the population data `vote` where the sample size is `100` and calculate the sample mean and check if the true proportion is included or not in the approximate 90% confidence interval based on the sample data.

```r
set.seed(2022)
n = 100 #sample size
B = 1000 #iteration number
inside = replicate(B, {
  x = ____________________  #random sample from vote
  x_hat = ____________  # compute the average
  se_hat = ____________________ # compute the standard error
  ________________________________________ # use `between` function
})
mean(inside)
```

(d) [10 points] Use a Monte Carlo simulation to learn the distribution of the sample mean of size 100. Obtain 90% confidence interval with the iteration number as 10^4.

```r
set.seed(2022)
B = 10^4
vote_p = replicate(B, {
  X = _____________ # random sample from vote
  ___________ # compute the average
})
________________________ # compute the 90% confidence interval
```

(e) [10 points] Set the seed number as 2022 and obtain 90% confidence interval for the true proportion using bootstrap. Use the bootstrap iteration number as 10^4. Compare this with 90% confidence interval based on the CLT.

Problem Set #3 (15 Points)

We consider `UCBAdmission` data set, which contains data on applicants to graduate school at Berkeley for the six largest departments in 1973 classified by admission and gender.

```
data(UCBAdmissions)
dat = as.data.frame(UCBAdmissions)[c("Dept","Admit","Gender","Freq")]
```

(a) [5 points] If we think of an observation as a `Dept`, then this data `dat` is not tidy. Use the `pivot_wider` function to wrangle into tidy shape: one row for each major with variable names as `Admitted_Male`, `Rejected_Male`, `Admitted_Female`, and `Rejected_Female`. Save this data set as an object `dat1` (as shown below).

```
## # A tibble: 6 x 5
##   Dept  Admitted_Male Rejected_Male Admitted_Female Rejected_Female
##   <fct>         <dbl>         <dbl>           <dbl>           <dbl>
## 1 A               512           313              89              19
## 2 B               353           207              17               8
## 3 C               120           205             202             391
## 4 D               138           279             131             244
## 5 E                53           138              94             299
## 6 F                22           351              24             317
```

(b) [5 points] Now use `pivot_longer` to wrangle the above `dat1` so that we obtain the following `dat2` (head part is shown).

```
## # A tibble: 6 x 3
##   Dept  Admit_Gender     Freq
##   <fct> <chr>           <dbl>
## 1 A     Admitted_Male     512
## 2 A     Rejected_Male     313
## 3 A     Admitted_Female    89
## 4 A     Rejected_Female    19
## 5 B     Admitted_Male     353
## 6 B     Rejected_Male     207
```

(c) [5 points] Note that the above `dat2` has a variable `Admit_Gender` containing information from the two variables `Admit` and `Gender`. Use `separate` function to obtain the same data as `dat` as shown below (only the head part is shown and column types may be different from those in 'dat').

```
## # A tibble: 6 x 4
##   Dept  Admit    Gender  Freq
##   <fct> <chr>    <chr>  <dbl>
## 1 A     Admitted Male     512
## 2 A     Rejected Male     313
## 3 A     Admitted Female    89
## 4 A     Rejected Female    19
## 5 B     Admitted Male     353
## 6 B     Rejected Male     207
```

Problem Set #4 (20 Points)

For this problem, we use actual polls from the 2016 election. We will use all the national polls.

```
library(dslabs)
library(lubridate)
data("polls_us_election_2016")
polls = polls_us_election_2016 %>% filter(state == "U.S.")
```

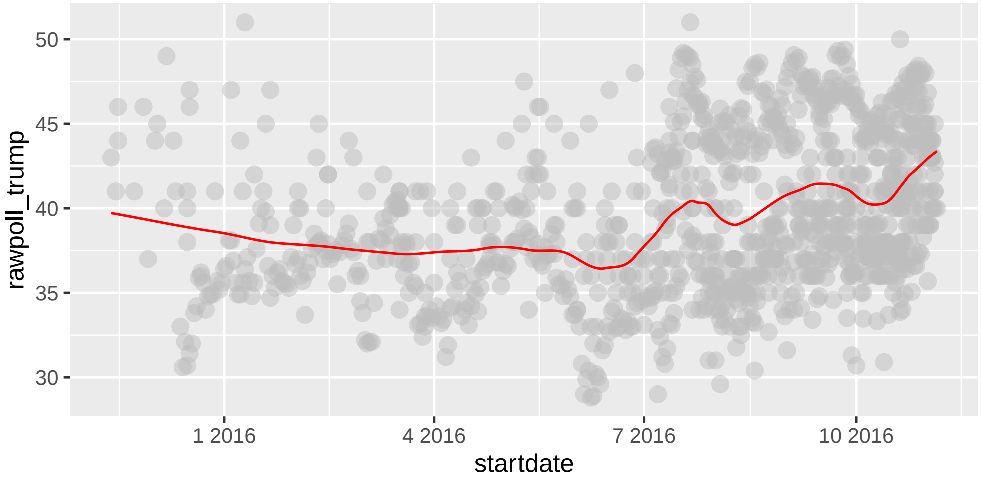
(a) [5 points] Sort `startdate` and show the first three dates. Consider the variable `enddate`, and find the proportion of October 2016.

(b) [5 points] Group all the polls by week of the year using `round_date` function (with `startdate` variable), and compute the average `rawpoll_trump` per week (saving this as `ave_T`), and draw a scatter plot using `qplot` function.

```
polls %>% mutate(________________________) %>%
  ___________________ %>%
  summarize(_____________________) %>%
  qplot(week, ave_T, data=., ylim = c(20,50))
```

(c) [10 points] Provide the loess fit for `rawpoll_trump` using `span = 0.2` and `degree = 1`. In order to use the loess function, we make a new variable `day` and fit a smooth line of `rawpoll_trump` versus `day`. Provide the following plot.

```
span = 0.2
polls$day = as.numeric(polls$startdate)
fit = loess(_____________________________________)
polls %>% mutate(smooth = fit$fitted) %>%
  ggplot(aes(_____________________)) +
  geom_point(size = 3, alpha = 0.5, color = "grey") +
  geom_line(_____________________, color = "red")
```


rawpoll_trump
startdate