Midterm Exam

[YOUR NAME]

Due Date: 2023-03-10

Overview

This is your midterm exam. It consists of eight questions plus an additional extra credit question. In addition, there is an additional extra credit opportunity if you respond to a short survey about this course. The survey is not part of Vanderbilt's official teaching evaluations. I use it to help me improve the course in the second half of the semester, and respond to your specific needs. The survey is anonymous.

Grading

Each question is worth 5 points, including the extra credit question and the survey. It is due by 11:59PM on Friday, March 10th. Five points will be deducted for each day late it is received. Submissions received after midnight on Sunday, March 12th will not be graded.

Please upload **two** versions of this midterm. The first is a PDF of the **knitted** output, just like your problem sets. The second is this .Rmd file in its raw form. An additional 5 points will be deducted for failing to submit in the correct format.

Resources

You are permitted to rely on any course resources from the first part of the Spring 2023 semester. These include all lecture slides, recordings, problem sets, answer keys, homeworks, and lecture notes, as well as any and all posts to Campuswire.

Campuswire access will be restricted during the week of the midterm. You are only permitted to post clarifying questions about the exam, and these should only be made visible to the instructor and TAs. The graders, TAs, and the Professor will remove questions that ask for help on the contents of the exam.

Honor

Unlike the problem sets, you are **prohibited** from working on this midterm together. You must digitally sign your name below, confirming that you did not collaborate on this exam with any of your classmates, share work, or otherwise discuss its contents.

Question 0: Independent Work Statement

Please sign your name in the space provided by typing out your full name in place of the underline:

"I, _____, am aware of the serious nature of plagiarism and affirm that I did not collaborate with other students while completing this midterm."

Codebook

Questions 1 through 4 use the `sc_debt.Rds` dataset, the codebook for which is reproduced below:

Name	Description
unitid	Unit ID
instnm	Institution Name

Name	Description
stabbr	State Abbreviation
grad_debt_mdn	Median Debt of Graduates
control	Control Public or Private
region	Census Region
preddeg	Predominant Degree Offered: Associates or Bachelors
openadmp	Open Admissions Policy: 1= Yes, 2=No,3=No 1st time students
adm_rate	Admissions Rate: proportion of applications accepted
ccbasic	Type of institution– see here (https://data.ed.gov/dataset/9dc70e6b-8426-4d71-b9d5-70ce6094a3f4/resource/658b5b83-ac9f-4e41-913e-9ba9411d7967/download/collegescorecarddatadictionary_01192021.xlsx)
selective	Institution admits fewer than 10 % of applicants, 1=Yes, 0=No
research_u	Institution is a research university 1=Yes, 0=No
sat_avg	Average SAT Scores
md_earn_wne_p6	Average Earnings of Recent Graduates
ugds	Number of undergraduates

Question 1: 5 points

Require `tidyverse` and load the `sc_debt.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/2_Intro_to_R/data/sc_debt.Rds?raw=true) dataset from `GitHub` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/2_Intro_to_R/data/sc_debt.Rds). **[1 point]**

Do schools with open admissions policies (i.e., schools that do not require SAT or ACT scores for admission) have students with higher future earnings? How big is the difference? **[2 points]**

Before you calculate the answer to this question, **wrangle** the admissions policy variable to be `"open"` if the school has an open admissions policy, and `"not"` otherwise. **[2 points] HINT**: see slide 39 in the `Data_Wrangling_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/3_Data_Wrangling/code/Data_Wrangling_slides.html#117) for help.

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
debt <- read_rds('https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/2_Intro_to_R/data/sc_debt.Rds?raw=true')

debt %>%
  drop_na(openadmp,md_earn_wne_p6) %>%
  mutate(open = ifelse(openadmp == 1,'open','not')) %>%
  group_by(open) %>%
  summarise(future_earn = mean(md_earn_wne_p6,na.rm=T)) %>%
  spread(open,future_earn) %>%
  mutate(diff = not - open)
```

```
## # A tibble: 1 × 3
##      not   open  diff
##    <dbl>  <dbl> <dbl>
## 1 36049. 26219. 9830.
```

Schools with open admissions policies have lower future earnings than those which require SAT or ACT scores. The difference is $9,829.56.

Question 2: 5 points

How confident are you in the conclusion drawn in question 1? Use 100 bootstrapped simulations with `size` set to the number of rows in the data to express your confidence.**[3 points]** How large is the average difference across these bootstraps? **[2 points] HINT**: see slide 30 in the `Multivariate_Analysis_part3_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/4_Uni_Multivariate/code/Multivariate_Analysis_part3_slides.html#82).

```
set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  bsRes <- debt %>%
    drop_na(openadmp,md_earn_wne_p6) %>%
    sample_n(size = nrow(debt),replace = T) %>%
    group_by(openadmp) %>%
    summarise(earn = mean(md_earn_wne_p6,na.rm=T)) %>%
    ungroup() %>%
    mutate(bsInd = i) %>%
    bind_rows(bsRes)
}

bsRes %>%
  mutate(openadmp = ifelse(openadmp == 1,'open','not')) %>%
  spread(openadmp,earn) %>%
  mutate(diff = not - open) %>%
  summarise(conf = mean(diff > 0,na.rm=T),
            diff = mean(diff,na.rm=T))
```

```
## # A tibble: 1 × 2
##    conf  diff
##   <dbl> <dbl>
## 1     1 9857.
```

I am more than 99.99% confident that schools that don’t have open admissions policies produce graduates who make more money than schools that do. Across 100 bootstraps, the average difference is $9,857.

Question 3: 5 points

Let's look at missingness in our data. How many schools don't report their admissions policy? How many don't report the median earnings of future graduates? **[1 point]**

Now investigate whether the missing data is associated with certain types of schools. Do schools that don't report future earnings have higher average SAT scores than the schools that do report future earnings?* **[2 points] HINT:** Create a new variable that is `"reported"` if the school reports future earnings, and `"missing"` otherwise. Again, slide 39 in the `Data_Wrangling_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/3_Data_Wrangling/code/Data_Wrangling_slides.html#117) is helpful.

Finally, use bootstrapping with 100 simulations and `size` set to the number of rows in the data to express your confidence in your conclusion. **[2 points]**

```
summary(debt %>% select(openadmp,md_earn_wne_p6))
```

```
##     openadmp     md_earn_wne_p6
##  Min.   :1.000   Min.   : 10600
##  1st Qu.:1.000   1st Qu.: 26100
##  Median :2.000   Median : 31500
##  Mean   :1.642   Mean   : 33028
##  3rd Qu.:2.000   3rd Qu.: 37400
##  Max.   :2.000   Max.   :120400
##  NA's   :64      NA's   :240
```

```
debt %>%
  mutate(missingEarn = ifelse(is.na(md_earn_wne_p6),'missing','reported')) %>%
  group_by(missingEarn) %>%
  summarise(mean(sat_avg,na.rm=T))
```

```
## # A tibble: 2 × 2
##   missingEarn `mean(sat_avg, na.rm = T)`
##   <chr>                            <dbl>
## 1 missing                          1154.
## 2 reported                         1140.
```

```
set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  bsRes <- debt %>%
    sample_n(size = nrow(debt),replace = T) %>%
  mutate(missingEarn = ifelse(is.na(md_earn_wne_p6),'missing','reported')) %>%
  group_by(missingEarn) %>%
  summarise(sat = mean(sat_avg,na.rm=T)) %>%
    mutate(bsInd = i) %>%
    bind_rows(bsRes)
}

bsRes %>%
  spread(missingEarn,sat) %>%
  mutate(diff = missing - reported) %>%
  summarise(conf = mean(diff > 0),
            avgDiff = mean(diff,na.rm=T))
```

```
## # A tibble: 1 × 2
##    conf avgDiff
##   <dbl>   <dbl>
## 1  0.72    14.1
```

64 schools don't report their admissions policy. 240 schools don't report the future earnings of their recent graduates. Schools that don't report future earnings have an average SAT score of 1153.7, while schools that do report have an average SAT score of 1140.3. However, based on bootstrap analysis, I am only 66% confident in this conclusion.

Question 4: 5 points

Now let's re-analyze the relationship between admissions policy and future earnings via regression.**[1 point]** Does this analysis change your answers to Questions 1 and 2 from above? **[2 points] HINT**: see slide 36 in the `Regression_part1_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/5_Regression/code/Regression_part1_slides.html#92).

Then calculate the RMSE using 100-fold cross validation with a 50-50 split, and interpret the result.**[2 points] HINT**: see slide 44 in the `Regression_part2_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/5_Regression/code/Regression_part2_slides.html#132).

```
debt_analysis <- debt %>% drop_na(md_earn_wne_p6,openadmp) %>%
  mutate(openadmp = ifelse(openadmp == 1,'open','not'))

m <- lm(md_earn_wne_p6 ~ openadmp,debt_analysis %>%
          mutate(md_earn_wne_p6 = log(md_earn_wne_p6)))
summary(m)
```

```
##
## Call:
## lm(formula = md_earn_wne_p6 ~ openadmp, data = debt_analysis %>%
##     mutate(md_earn_wne_p6 = log(md_earn_wne_p6)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00506 -0.11959 -0.00062  0.11326  1.23631
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.462263   0.005985 1748.15   <2e-16 ***
## openadmpopen -0.307392   0.010032  -30.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2284 on 2260 degrees of freedom
## Multiple R-squared:  0.2935, Adjusted R-squared:  0.2932
## F-statistic: 938.8 on 1 and 2260 DF,  p-value: < 2.2e-16
```

```
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(debt_analysis),size = round(nrow(debt_analysis)*.5),replace = F)
  train <- debt_analysis %>% slice(inds)
  test <- debt_analysis %>% slice(-inds)

  m <- lm(md_earn_wne_p6 ~ openadmp,train)
  cvRes <- test %>%
    mutate(preds = predict(m,newdata = test)) %>%
    summarise(rmse = sqrt(mean((md_earn_wne_p6 - preds)^2))) %>%
    mutate(cvInd = i) %>%
    bind_rows(cvRes)
}

cvRes %>%
  summarise(mean(rmse))
```

```
## # A tibble: 1 × 1
##   `mean(rmse)`
##          <dbl>
## 1        8150.
```

No the regression does not change my substantive conclusion from Questions 1 and 2 above. Schools with an open admissions policy produce grads who make, on average, $9,829.60 less than schools without open admissions. However, my model's errors are, on average $8,242.

Question 5: 5 points

Now open a different dataset: the `game_summary.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/game_summary.Rds?raw=true) dataset from GitHub (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/game_summary.Rds). The codebook for this dataset is copied below.

Name	Description
idGame	Unique game id
yearSeason	Which season? NBA uses ending year so 2016-17 = 2017
dateGame	Date of the game
idTeam	Unique team id
nameTeam	Team Name
locationGame	Game location, H=Home, A=Away
tov	Total turnovers
pts	Total points
treb	Total rebounds
pctFG	Field Goal Percentage
teamrest	How many days since last game for team
pctFT	Free throw percentage
isWin	Won? TRUE or FALSE
ft_80	Team scored more than 80 percent of free throws

We are fundamentally interested in predicting the number of points a team scores as a function of three variables: turnovers, the location of the game, and how much rest the team has had prior to playing in this game.

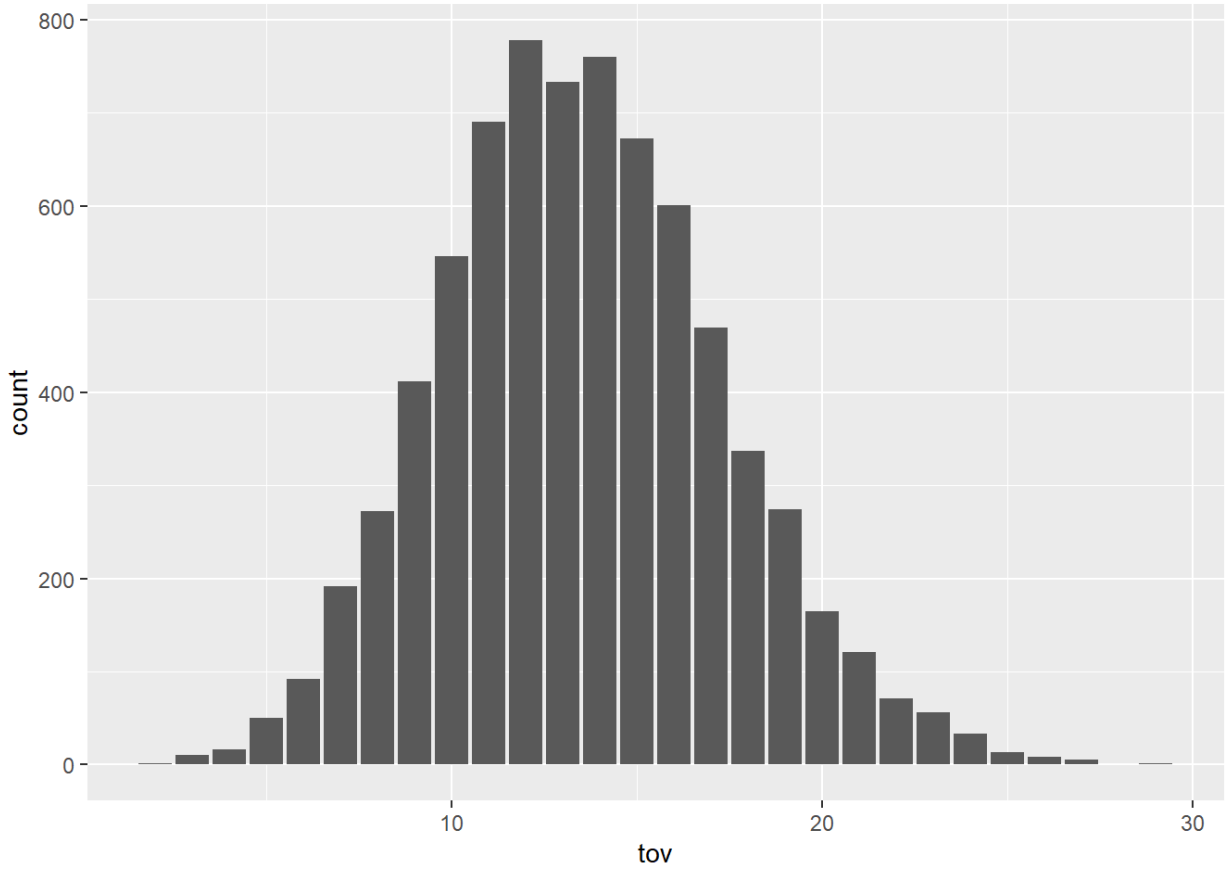
To start, answer the following questions:

1. Which variable is the outcome / dependent / Y variable? Which are the X variables? **[2 points]**
2. Is there any missingness in these data? **[1 point]**
3. Create univariate visualizations of all three predictors and the outcome. Based on your visual analysis, do you see any reason to wrangle the data? Make sure to label your plots! **[2 points]**

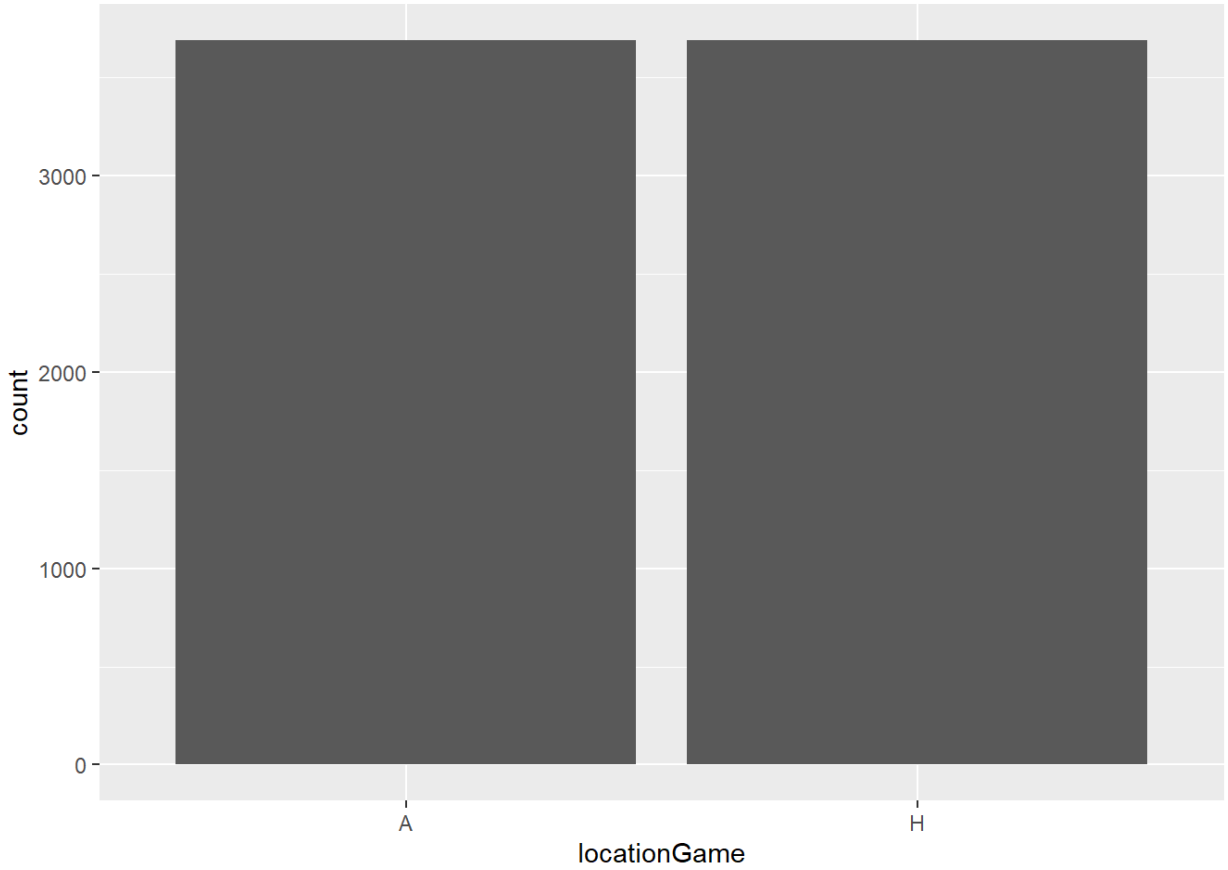
```
gms <- read_rds('https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/gam
e_summary.Rds?raw=true')

summary(gms %>% select(tov,locationGame,teamrest,pts))
```

```
##       tov        locationGame          teamrest           pts       
##  Min.   : 2.00   Length:7380        Min.   :  0.000   Min.   : 64.0  
##  1st Qu.:11.00   Class :character   1st Qu.:  1.000   1st Qu.: 99.0  
##  Median :13.00   Mode  :character   Median :  1.000   Median :108.0  
##  Mean   :13.53                      Mean   :  2.579   Mean   :107.7  
##  3rd Qu.:16.00                      3rd Qu.:  1.000   3rd Qu.:116.0  
##  Max.   :29.00                      Max.   :120.000   Max.   :168.0
```

```
gms %>%
  ggplot(aes(x = tov)) +
  geom_bar()
```

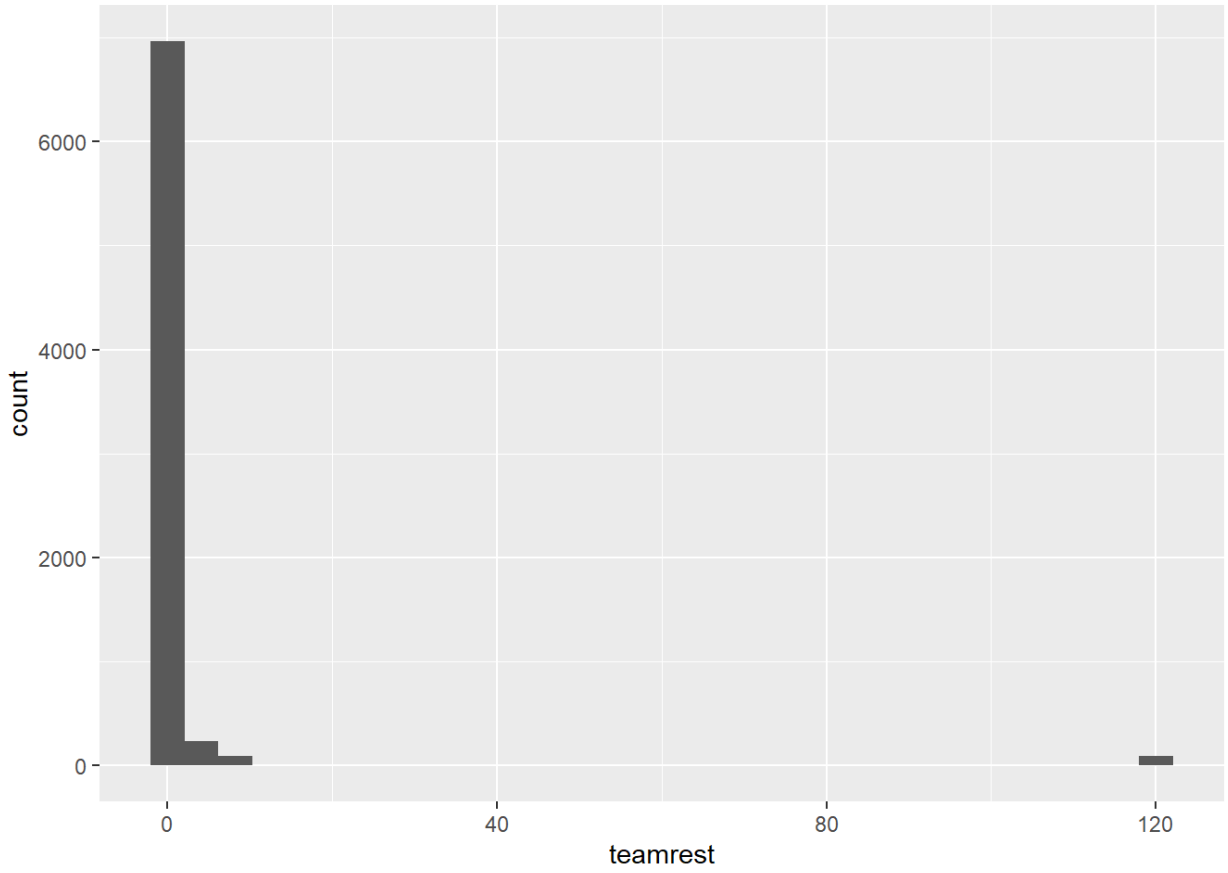



```
gms %>%
  ggplot(aes(x = locationGame)) +
  geom_bar()
```



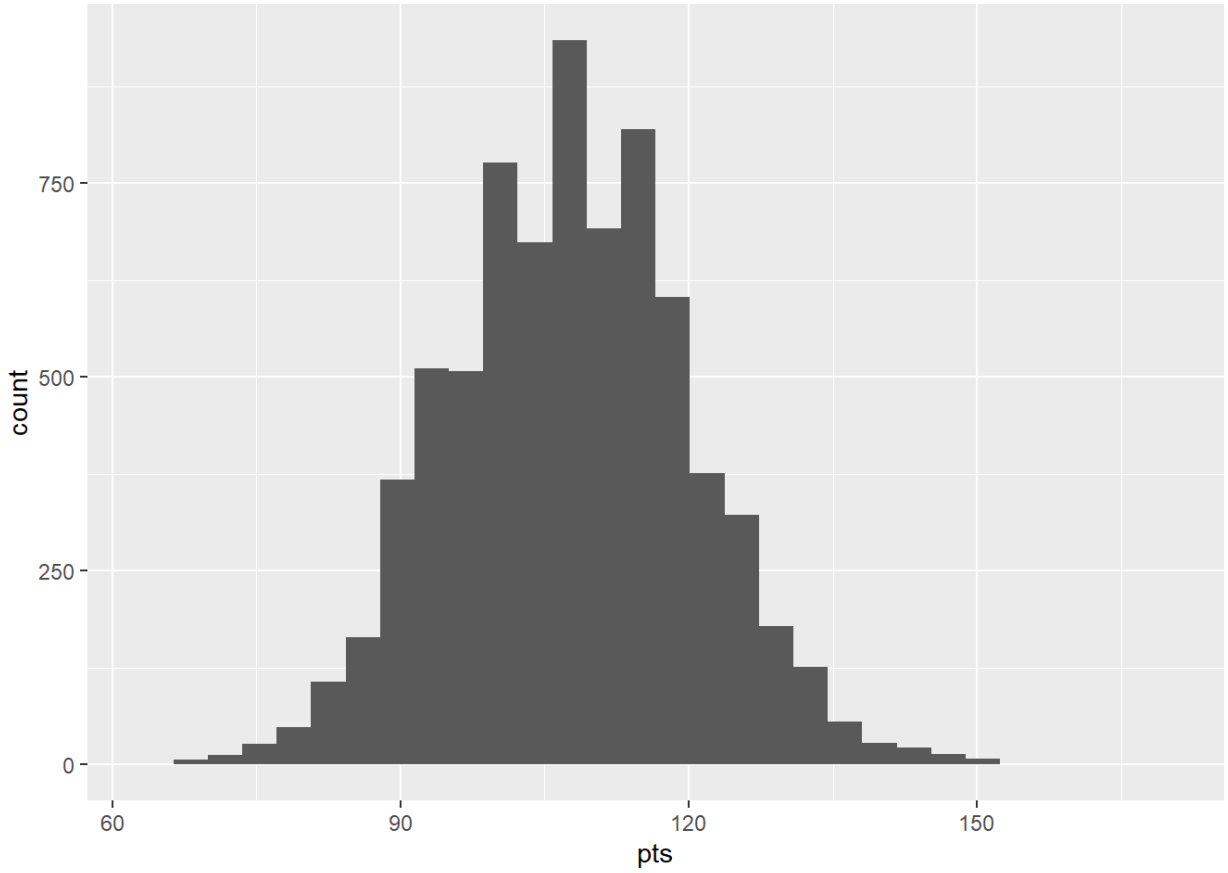
```
gms %>%
  ggplot(aes(x = teamrest)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
gms %>%
  ggplot(aes(x = pts)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



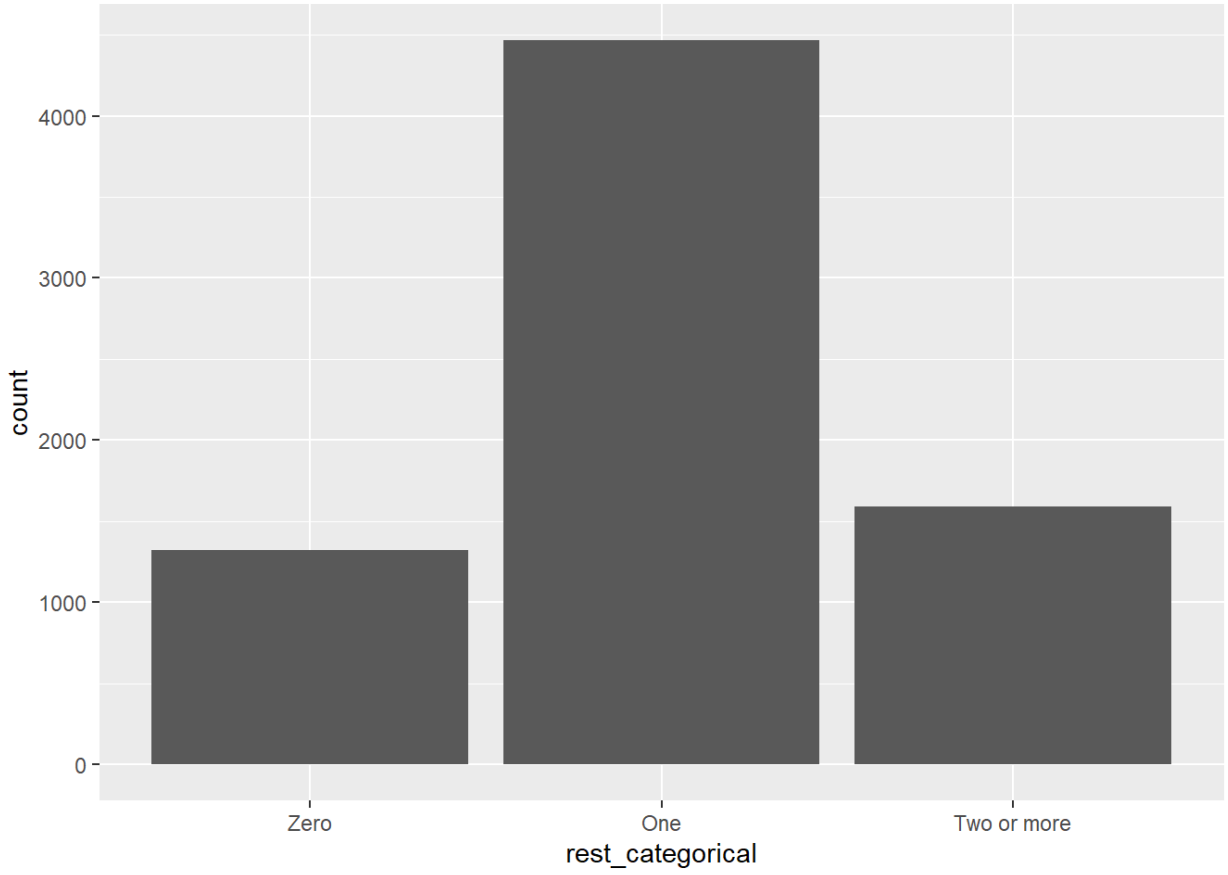
1. The outcome variable is the points (`pts`). The X variables are the turnovers (`tov`), the location of the game (`locationGame`), and the amount of rest the team has (`teamrest`).
2. There is no missingness across any of the four variables of interest.
3. The only variable that indicates a need to wrangle is the `teamrest` variable, which is highly skewed.

Question 6: 5 points

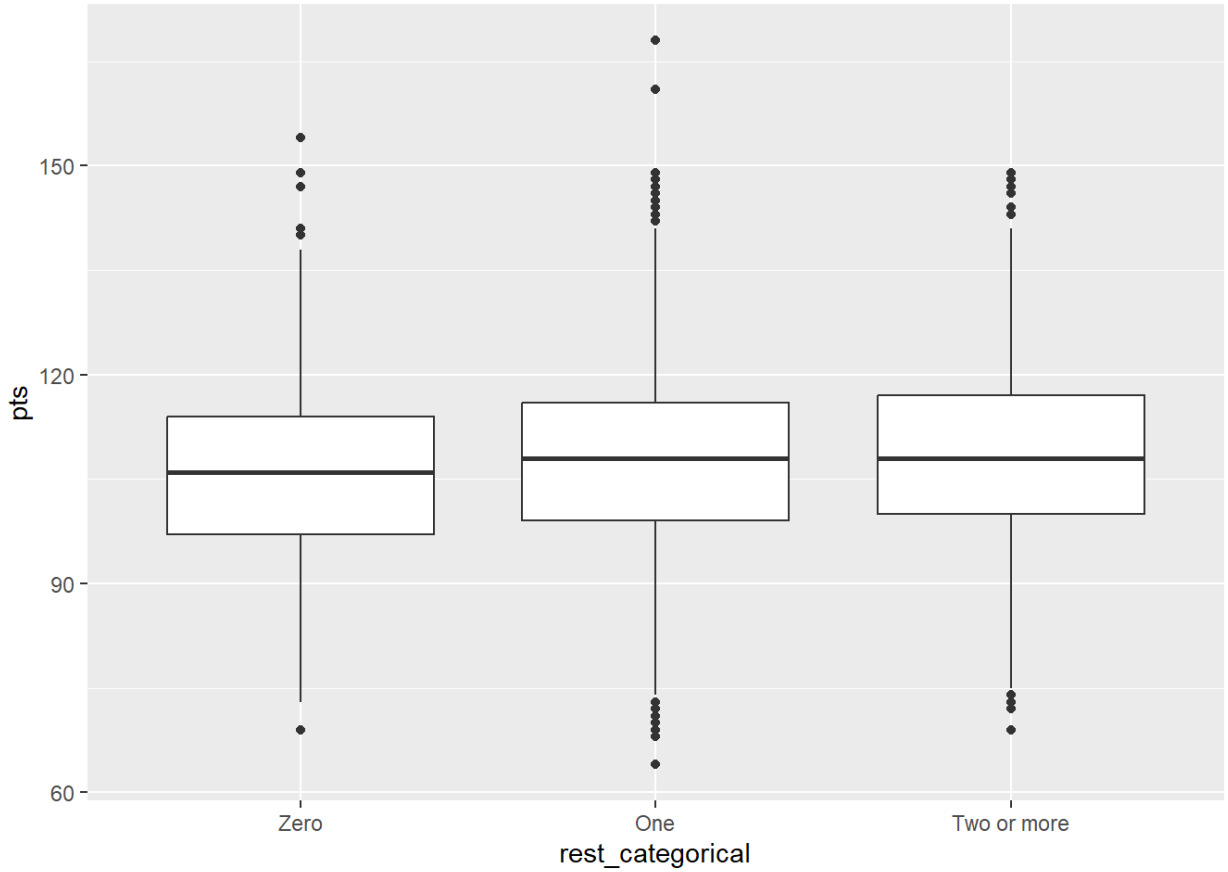
Let's start with the amount of rest the team has. Create a new version of this variable that is categorical, with one category for zero days of rest, one category for one day of rest, and one category for two or more days of rest.**[2 points] HINT**: see slide 39 in the `Data_Wrangling_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/3_Data_Wrangling/code/Data_Wrangling_slides.html#117) for help.

Now create a multivariate visualization comparing this new variable to the total number of points.**[2 points]** Based on the multivariate visualization, do teams with more rest appear to score more points than teams with less rest?**[1 point]**

```
gms <- gms %>%
  mutate(rest_categorical = factor(ifelse(teamrest == 0,'Zero',
                                   ifelse(teamrest == 1,'One','Two or more')),levels = c('Zero','One','T
wo or more')),
         rest_categorical_unordered = factor(ifelse(teamrest == 0,'Zero',
                                   ifelse(teamrest == 1,'One','Two or more'))))

gms %>%
  ggplot(aes(x = rest_categorical)) +
  geom_bar()
```




```
gms %>%
  ggplot(aes(x = rest_categorical,y = pts)) +
  geom_boxplot()
```



Based on the multivariate visualization, there appears to be a small positive relationship between the amount of points scored and the amount of rest.

Question 7: 5 points

Now run a regression using these same variables. Do teams with more rest tend to score more points than teams with less rest?**[1 point]** Evaluate your model via visual inspection of the univariate and multivariate errors.**[2 points]** Is this a good regression model? **[1 point] HINT**: see slides 25 through 28 in the `Regression_part2_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/5_Regression/code/Regression_part2_slides.html#65).

Finally, calculate the RMSE using 100-fold cross validation with an 80-20 split, and interpret the result.**[1 point]**

```
m <- lm(pts ~ rest_categorical,gms)
summary(m)
```

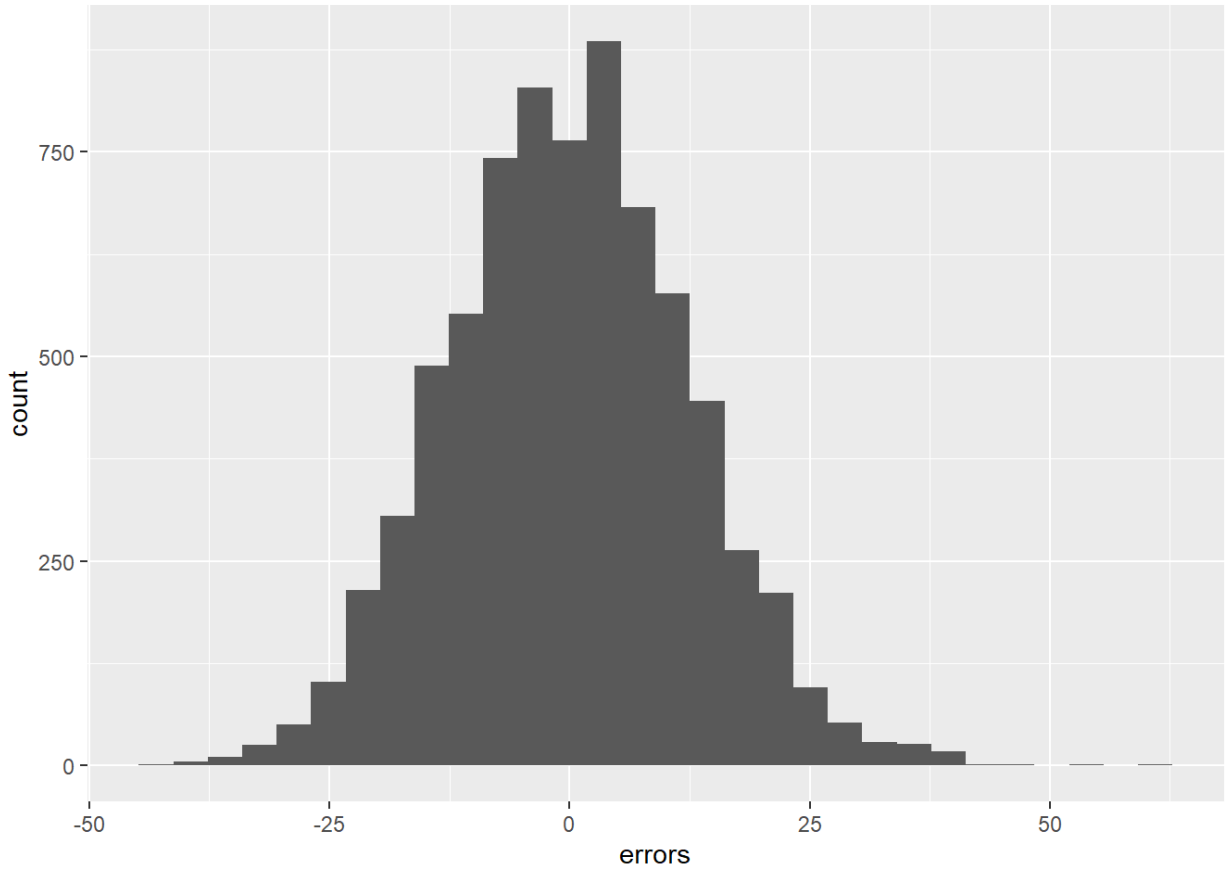
```
##
## Call:
## lm(formula = pts ~ rest_categorical, data = gms)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.871  -8.871   0.129   8.129  60.129
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 106.0877     0.3442 308.211  < 2e-16 ***
## rest_categoricalOne           1.7833     0.3919   4.551 5.43e-06 ***
## rest_categoricalTwo or more   2.5212     0.4656   5.415 6.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.52 on 7377 degrees of freedom
## Multiple R-squared:  0.004208,   Adjusted R-squared:  0.003938
## F-statistic: 15.59 on 2 and 7377 DF,  p-value: 1.761e-07
```

```
# Some students didn't reorder their categorical variable. The results should
#   look like this.
m <- lm(pts ~ rest_categorical_unordered,gms)
summary(m)
```

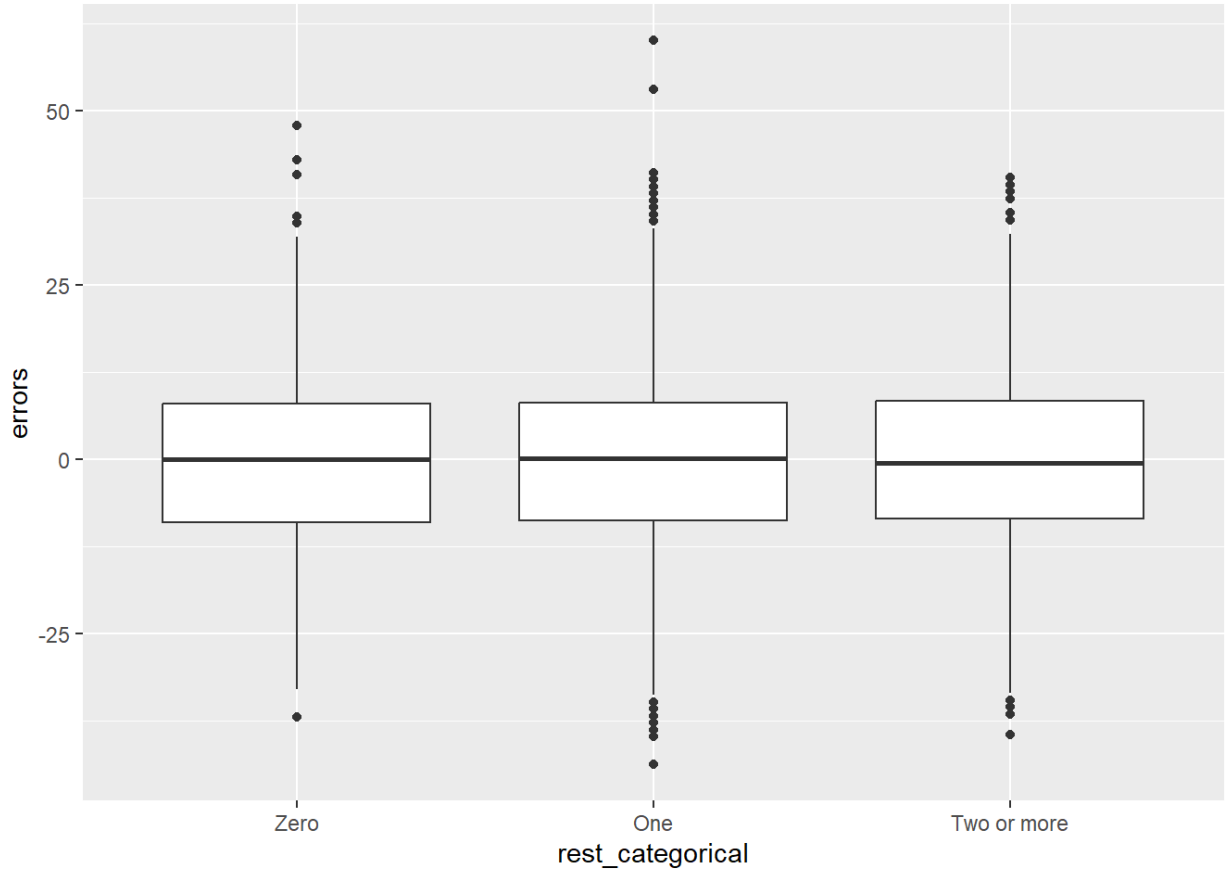
```
##
## Call:
## lm(formula = pts ~ rest_categorical_unordered, data = gms)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.871  -8.871   0.129   8.129  60.129
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        107.8710     0.1873 575.948  < 2e-16 ***
## rest_categorical_unorderedTwo or more   0.7379     0.3652   2.020   0.0434 *
## rest_categorical_unorderedZero         -1.7833     0.3919  -4.551 5.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.52 on 7377 degrees of freedom
## Multiple R-squared:  0.004208,   Adjusted R-squared:  0.003938
## F-statistic: 15.59 on 2 and 7377 DF,  p-value: 1.761e-07
```

```
gms$errors <- resid(m)

gms %>%
  ggplot(aes(x = errors)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
gms %>%
  ggplot(aes(x = rest_categorical,y = errors)) +
  geom_boxplot()
```




```
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(gms),size = round(nrow(gms)*.8),replace = F)
  train <- gms %>% slice(inds)
  test <- gms %>% slice(-inds)

  m <- lm(pts ~ rest_categorical,train)

  cvRes <- cvRes %>%
    bind_rows(test %>%
    mutate(preds = predict(m,newdata = test)) %>%
    summarise(rmse = sqrt(mean((pts - preds)^2))) %>%
    mutate(cvInd = i))
}

mean(cvRes$rmse)
```

```
## [1] 12.53943
```

According to the regression, teams with no days of rest score 106 points per game. Those with one day of rest score an additional 1.78 points, while those with two days of rest score an additional 2.52 points. I am more than 99.99% confident in both of these conclusions. Furthermore, I believe this model is good since there is no evidence of asymmetry in either the univariate or multivariate visualizations of the errors. On average, the model's predictions are off by roughly 12.5 points.

Question 8: 5 points

Now run a combined regression where you predict the points scored as a function of the categorical days of rest variable you created in Q6, as well as the number of turnovers and the location of the game.**[1 point]** Does this combined model perform better than the simple model used in Question 7? Use 100-fold cross validation with an 80-20 split to answer this question. **[2 points]**

Finally, using this combined model, predict the number of points scored by a team playing at home with 5 turnovers and 1 day of rest, as well as the lower and upper bounds of your prediction. **[2 points] HINT**: see slide 36 in the `Regression_part2_slides.html` (https://www.jamesbisbee.com/DS1000_S2023/Lectures/5_Regression/code/Regression_part2_slides.html#103).

```
summary(mComb <- lm(pts ~ rest_categorical + tov + locationGame,gms))
```

```
##
## Call:
## lm(formula = pts ~ rest_categorical + tov + locationGame, data = gms)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.731  -8.477  -0.179   8.165  62.359
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                110.13904    0.62111 177.326  < 2e-16 ***
## rest_categoricalOne          1.35176    0.39142   3.453 0.000557 ***
## rest_categoricalTwo or more  2.18415    0.46333   4.714 2.47e-06 ***
## tov                         -0.36560    0.03783  -9.665  < 2e-16 ***
## locationGameH                2.45591    0.29114   8.436  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.38 on 7375 degrees of freedom
## Multiple R-squared:  0.0263, Adjusted R-squared:  0.02577
## F-statistic:  49.8 on 4 and 7375 DF,  p-value: < 2.2e-16
```

```
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(gms),size = round(nrow(gms)*.8),replace = F)
  train <- gms %>% slice(inds)
  test <- gms %>% slice(-inds)

  mComb <- lm(pts ~ rest_categorical + tov + locationGame,train)

  cvRes <- cvRes %>%
    bind_rows(test %>%
    mutate(predComb = predict(mComb,newdata = test)) %>%
    summarise(rmseComb = sqrt(mean((pts - predComb)^2))) %>%
    mutate(cvInd = i))
}


cvRes %>%
  summarise(mean(rmseComb))
```

```
## # A tibble: 1 × 1
##   `mean(rmseComb)`
##              <dbl>
## 1             12.4
```

```
test <- data.frame(rest_categorical = 'One',tov = 5,locationGame = "H")


summary(mComb <- lm(pts ~ rest_categorical + tov + locationGame,gms))
```

```
##
## Call:
## lm(formula = pts ~ rest_categorical + tov + locationGame, data = gms)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.731  -8.477  -0.179   8.165  62.359
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                110.13904    0.62111 177.326  < 2e-16 ***
## rest_categoricalOne          1.35176    0.39142   3.453 0.000557 ***
## rest_categoricalTwo or more  2.18415    0.46333   4.714 2.47e-06 ***
## tov                         -0.36560    0.03783  -9.665  < 2e-16 ***
## locationGameH                2.45591    0.29114   8.436  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.38 on 7375 degrees of freedom
## Multiple R-squared:  0.0263, Adjusted R-squared:  0.02577
## F-statistic:  49.8 on 4 and 7375 DF,  p-value: < 2.2e-16
```

```
predict(mComb,test)
```

```
##        1
## 112.1187
```

```
predict(mComb,test) - 12.38826
```

```
##        1
## 99.73044
```

```
predict(mComb,test) + 12.38826
```

```
##       1
## 124.507
```

According to the cross validation results, the combined model is the best to predict the number of points scored, making mistakes of 12.39 points on average, as compared to 12.52 points for the amount of rest.

A team with 5 turnovers playing at home on 1 day of rest is predicted to score 112.12 points. The lower bound is 99.73 points and the upper bound is 124.51 points.

Extra Credit #1: 5 EC points

Open a new dataset that contains Donald Trump's tweets: the `Trumptweets.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/6_Midterm/data/Trumptweets.Rds?raw=true) dataset from GitHub (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/6_Midterm/data/Trumptweets.Rds). The units of observation in this dataset (i.e., the rows) are every tweet written by Donald Trump. This dataset has the following variables:

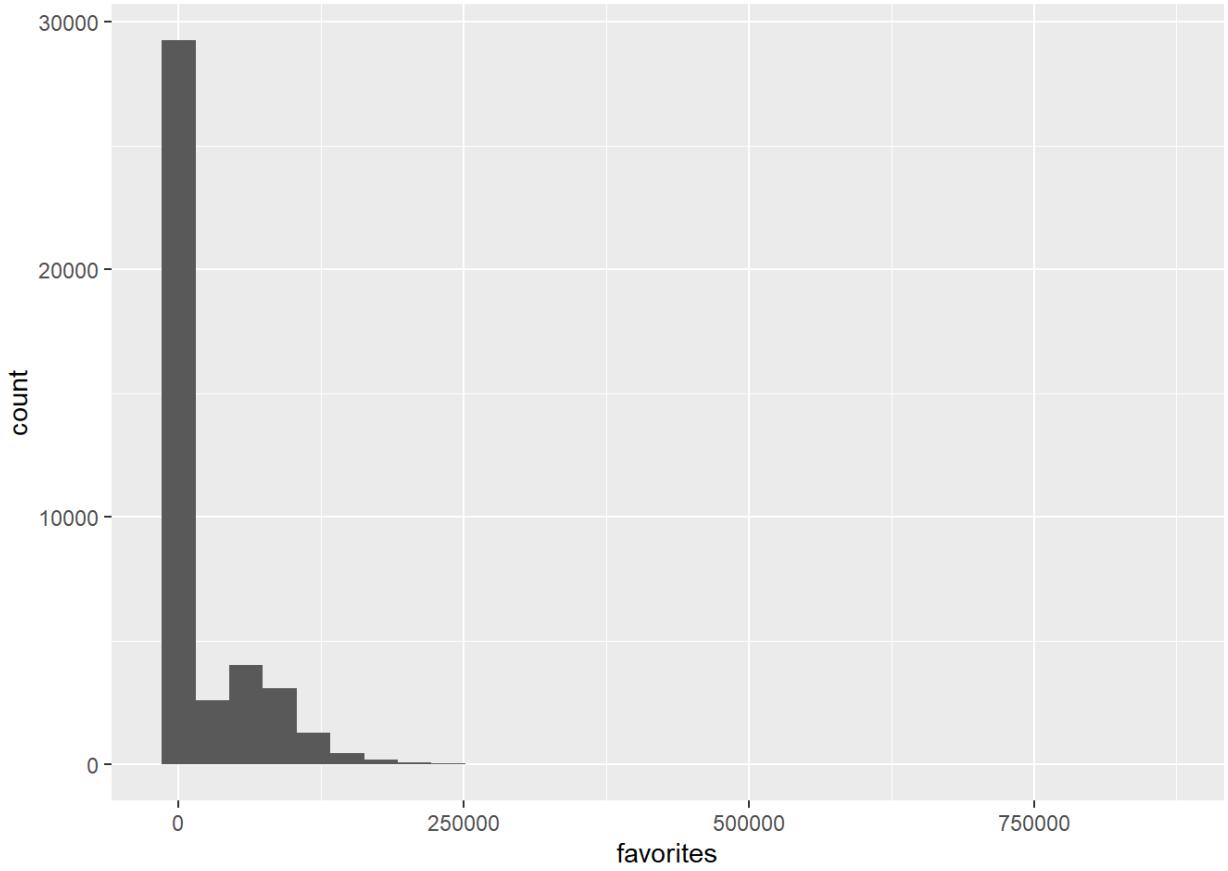
Name	Description
tweet_id	Tweet ID
content	The text of the tweet
date	The date the tweet was written
retweets	The total number of times the tweet was retweeted
favorites	The total number of times the tweet was favorited

Now answer the following research question: did Donald Trump's popularity on Twitter increase after he announced his candidacy for president on June 19, 2016? Full extra credit will be based on the following rubric:

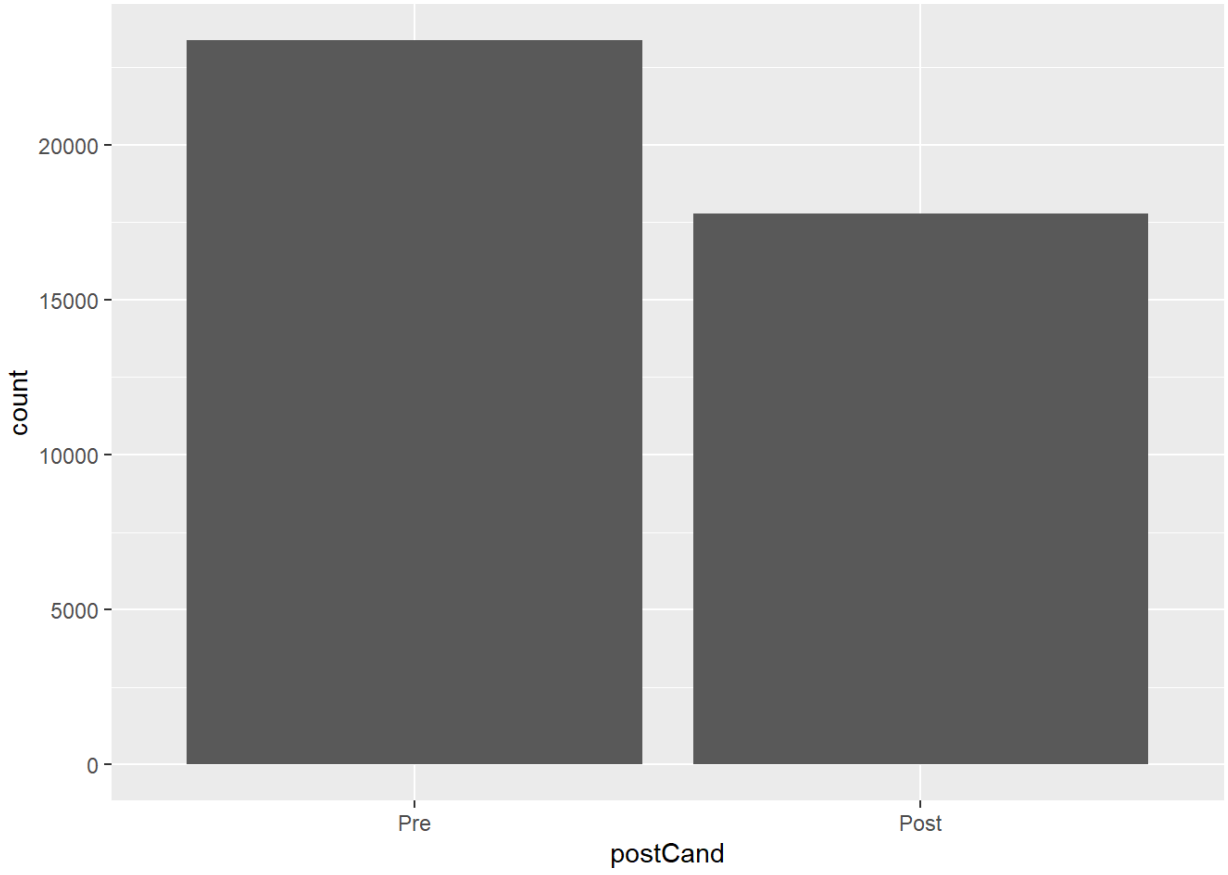
- Write out a theory and hypothesis (1 point)
- State which variables you will use, and indicate which is the outcome and which is the predictor (1 point)
- Visualize the outcome and explanatory variable (1 point) **Tip: add +1 to any variables you think require a log transformation before taking the log.**
- Run a regression and interpret the results (1 point)
- Evaluate your model with univariate and multivariate visualization of the errors, and 100-fold cross validation with an 80-20 split (1 point)

```
tweets <- read_rds('./Trumptweets.Rds')

tweets %>%
  ggplot(aes(x = favorites)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



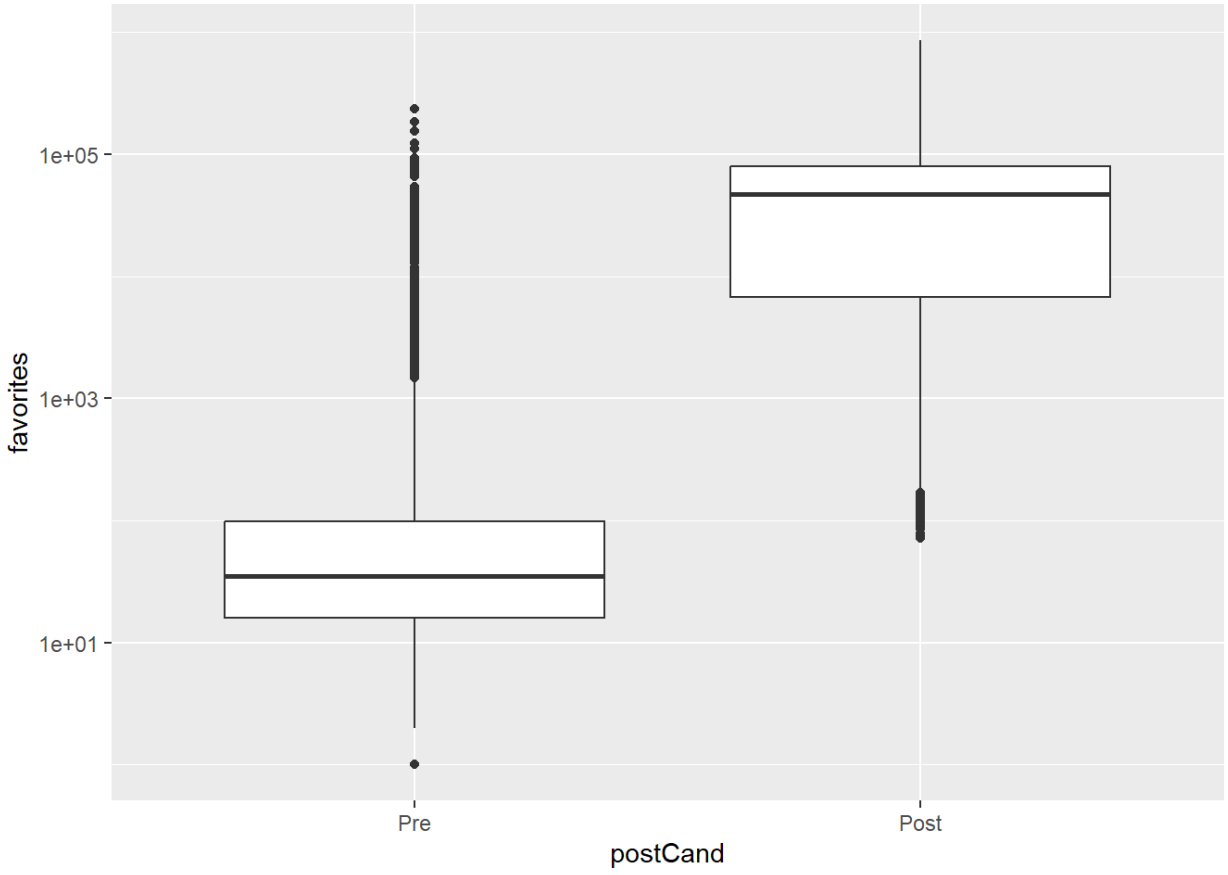

```
tweets <- tweets %>%
  mutate(postCand = factor(ifelse(date > as.Date('2015-06-19'),'Post','Pre'),levels = c('Pre','Post')))

tweets %>%
  ggplot(aes(x = postCand)) +
  geom_bar()
```




```
tweets %>%
  ggplot(aes(x = postCand,y = favorites)) +
  geom_boxplot() +
  scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 801 rows containing non-finite values (`stat_boxplot()`).
```

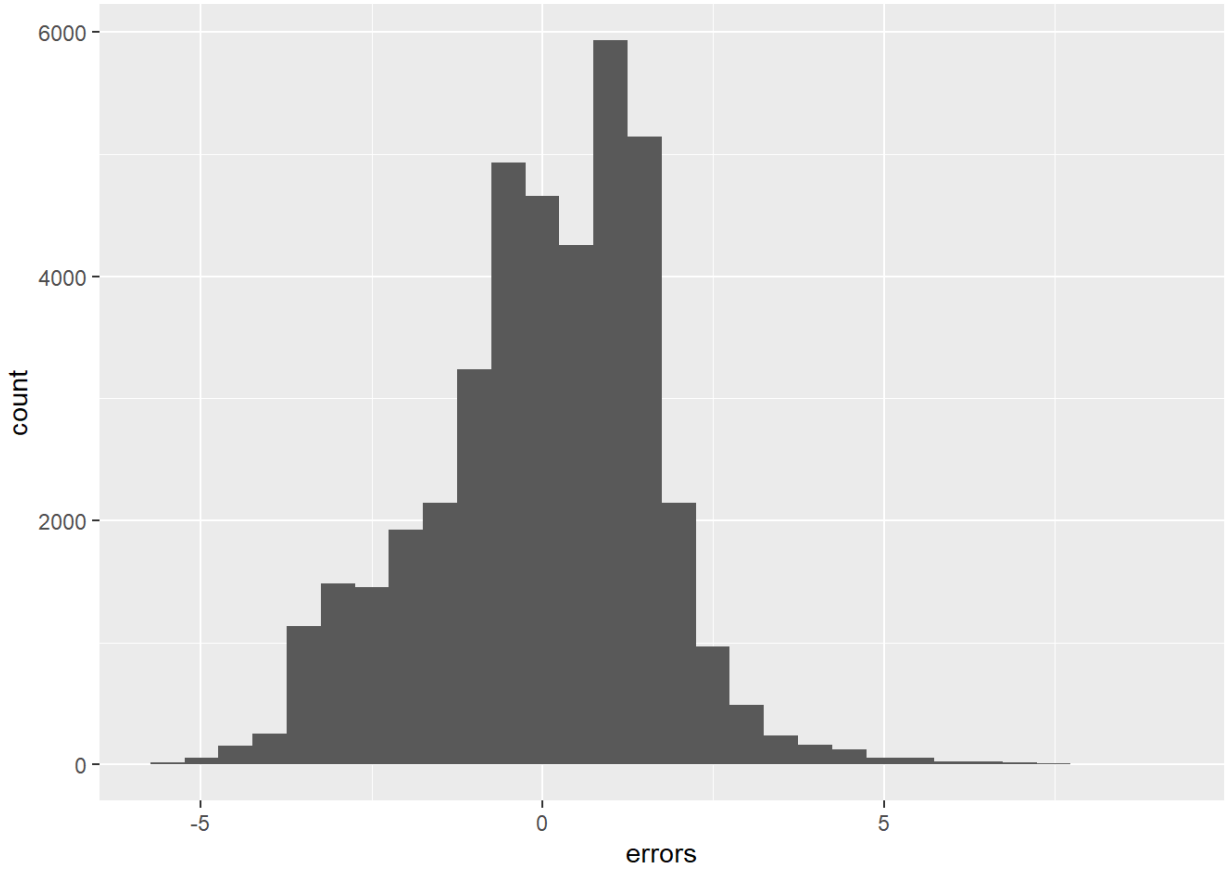



```
tweets <- tweets %>%
  mutate(log_favs = log(favorites+1)) %>%
  drop_na(log_favs,postCand)

summary(m <- lm(log_favs ~ postCand,tweets))
```

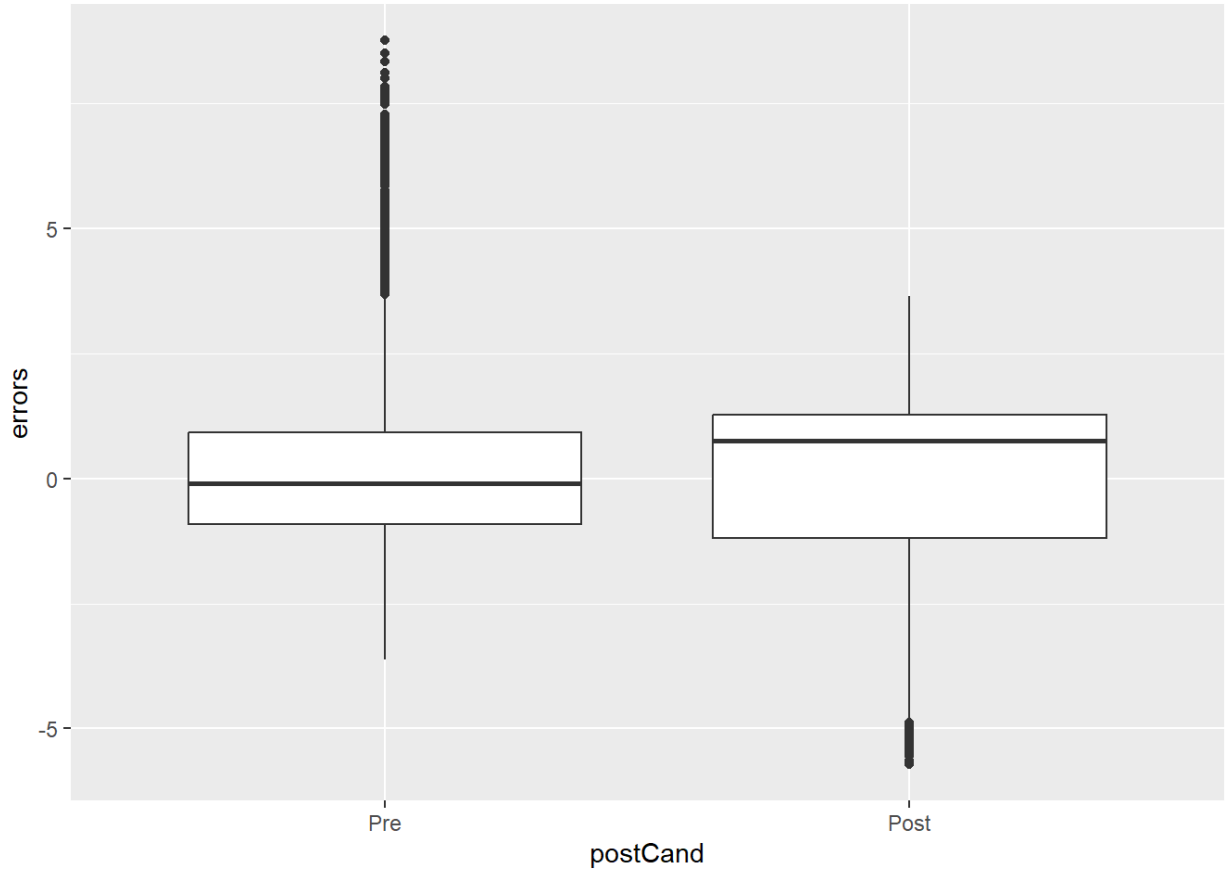
```
##
## Call:
## lm(formula = log_favs ~ postCand, data = tweets)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7212 -0.9846  0.1375  1.1885  8.7557
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.62366    0.01093   331.4   <2e-16 ***
## postCandPost  6.38797    0.01664   384.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.671 on 41120 degrees of freedom
## Multiple R-squared:  0.7819, Adjusted R-squared:  0.7819
## F-statistic: 1.475e+05 on 1 and 41120 DF,  p-value: < 2.2e-16
```

```
tweets <- tweets %>%
  mutate(preds = predict(m)) %>%
  mutate(errors = log_favs - preds)

tweets %>%
  ggplot(aes(x = errors)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
tweets %>%
  ggplot(aes(x = postCand,y = errors)) +
  geom_boxplot()
```




```
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(tweets),size = round(nrow(tweets)*.8),replace = F)
  train <- tweets %>% slice(inds)
  test <- tweets %>% slice(-inds)

  m <- lm(log_favs ~ postCand,train)

  cvRes <- cvRes %>%
    bind_rows(test %>%
    mutate(preds = predict(m,newdata = test)) %>%
    summarise(rmse = sqrt(mean((log_favs - preds)^2))))
}

mean(cvRes$rmse)
```

```
## [1] 1.670184
```

- I theorize that being a presidential candidate makes an individual more famous, and that fame leads to more popular posts on Twitter. As such, I hypothesize that Trump became more popular on Twitter after announcing his candidacy for president on June 19th, 2016.
- My outcome variable is the number of favorites his tweets received. My predictor is a binary measure of whether the tweet was posted prior to, or following, his candidacy.
- Based on the univariate visualization of the variables, I need to log the favorites since they are highly skewed.
- The regression model indicates that, prior to announcing his candidacy, Donald Trump's tweets received, on average, `exp(3.62366)` favorites. After announcing his candidacy, this increased by `exp(6.38797)`.
- The univariate visualization of the errors indicates some asymmetry, and the multivariate visualization further suggests that the model is a poor fit to the data, with many of the pre-announcement predictions under-estimating the

popularity of his posts, while many of the post-announcement predictions over-estimating his popularity. The overall RMSE is 1.67.

Extra Credit #2: 5 EC Points

Please complete this **anonymous** course evaluation. This does not influence Professor Bisbee's career or position in the university and will only be used to improve the course. You can find the anonymous survey here (https://nyu.qualtrics.com/jfe/form/SV_b7t5vqhhbalgGZ8). Upon completing the survey, you will be given a completion code, which you should paste back at the end of your midterm below.

NOTE: There is only one completion code to ensure that all responses are anonymized and can't be linked back to the midterm exams. To prevent students from sharing the code with their friends to get the 5 extra credit points without completing the survey, these 5 points are only provided if the number of midterms with the completion code *exactly equals the number of survey responses*. In other words, if there are 150 exams with the completion code, but only 50 completed surveys, **all students will forfeit their extra credit points**. The purpose of this strict rule is to disincentivize the sharing of this code either by those who would fill out the survey and then share the code, or by those who would ask to be given the code without filling out the survey.

- `D@taSc!enceForEveryone`