Assignment 2

YOURNAME

2023-10-30

[NOTE]

An important part of developing your R programming skills is learning to write clear, well-formatted code that is easy for another person to follow. For homework assignments, think of the person grading the code as the audience to whom you are trying to communicate. The easier it is to follow your code, the more likely it will be that you receive partial credit for answers that have minor errors.

Back to Assignment 2

This assignment involves a variety of different datasets that we have seen in previous classes. For simplicity (and to ensure consistency across computer operating systems), I have saved all of the relevant data in an .RData file, which is loaded as follows:

```
load("Assignment2data.RData")

ls()
```

```
## [1] "City_poll"          "City_poll_codebook" "Soccer_fouls"
## [4] "STAR_g3"
```

STAR (13 pt)

You will analyze part of the data from the Tennessee Student Teacher Achievement Ratio study (`STAR_g3` dataset). Project STAR was a large, block-randomized experiment that evaluated the effect of class size (the number of students per classroom) during the elementary grades on students' academic performance.

1. Use `filter` to find the data for all of the black female students born in July of 1979:

```
# R code here
#load tidyverse library
library(tidyverse)

#filter the data
Fstud<- STAR_g3%>%
  filter(gender=="FEMALE", race=="BLACK",
         birthmonth=="JULY", birthyear==1979)
#Data overview
head(Fstud)
```

```
##   stdntid gender  race birthmonth birthday birthyear cmpstype g1schid
## 1   10412 FEMALE BLACK       JULY       28      1979     AIDE  244708
## 2   10422 FEMALE BLACK       JULY       28      1979    SMALL  244708
## 3   10435 FEMALE BLACK       JULY        7      1979     AIDE  205492
## 4   11278 FEMALE BLACK       JULY       10      1979     AIDE  244776
## 5   11293 FEMALE BLACK       JULY       29      1979     <NA>  244806
## 6   11297 FEMALE BLACK       JULY       28      1979  REGULAR  244736
##     g1surban g3treadss g3tmathss g3tlangss g3tlistss
## 1 INNER CITY        NA        NA        NA        NA
## 2 INNER CITY        NA        NA        NA        NA
## 3   SUBURBAN        NA        NA        NA        NA
## 4 INNER CITY        NA        NA        NA        NA
## 5 INNER CITY       593       601       620       601
## 6 INNER CITY        NA        NA        NA        NA
```

2. Use `select` to create a dataset containing the school and student id variables for all students, plus the three variables containing information about students' birthdates. Print out the first 10 rows in this dataset.

```
# R code here
#2.
Birthstud<- STAR_g3%>%
  select(cmpstype, stdntid, starts_with("birth"))

#First 10 rows
head(Birthstud,10)
```

```
##    cmpstype stdntid birthmonth birthday birthyear
## 1   REGULAR   10003        MAY       28      1980
## 2      AIDE   10015     AUGUST       25      1980
## 3     SMALL   10022    OCTOBER       27      1980
## 4     SMALL   10026    JANUARY       15      1979
## 5   REGULAR   10032   FEBRUARY       19      1979
## 6      <NA>   10033      MARCH        6      1980
## 7   REGULAR   10039   NOVEMBER       12      1979
## 8   REGULAR   10041   NOVEMBER       22      1979
## 9   REGULAR   10042       JULY       10      1979
## 10    SMALL   10043       JULY       25      1979
```

3. Use `mutate` and `paste()` to create variable containing each student's birth date in the form of a character string (e.g., `"JANUARY 12, 1981"`). Print out the first 10 rows in this dataset.

```
# R code here
#Add Birthday date to Birthstud dataset
Birthstud<- Birthstud %>%
  mutate(birthdate= paste(birthmonth," ", birthday,", ", birthyear))

#first 10 rows
head(Birthstud, 10)
```

```
##    cmpstype stdntid birthmonth birthday birthyear            birthdate
## 1   REGULAR   10003        MAY       28      1980      MAY   28 ,  1980
## 2      AIDE   10015     AUGUST       25      1980   AUGUST   25 ,  1980
## 3     SMALL   10022    OCTOBER       27      1980  OCTOBER   27 ,  1980
## 4     SMALL   10026    JANUARY       15      1979  JANUARY   15 ,  1979
## 5   REGULAR   10032   FEBRUARY       19      1979 FEBRUARY   19 ,  1979
## 6      <NA>   10033      MARCH        6      1980     MARCH   6 ,  1980
## 7   REGULAR   10039   NOVEMBER       12      1979 NOVEMBER   12 ,  1979
## 8   REGULAR   10041   NOVEMBER       22      1979 NOVEMBER   22 ,  1979
## 9   REGULAR   10042       JULY       10      1979     JULY   10 ,  1979
## 10    SMALL   10043       JULY       25      1979     JULY   25 ,  1979
```

4. Use `summarise` to calculate the average reading scores (`g3treadss`) for males and females. (HINT: use `[]` .)

```
# R code here
#calculate the average reading scores by gender
#calculate the average reading scores by gender
STAR_g3[, c("g3treadss", "gender")]%>%
  drop_na(gender)%>%
  group_by(gender)%>%
  summarise(avg_g3treadss= mean(g3treadss, na.rm = T))
```

```
## # A tibble: 2 × 2
##   gender avg_g3treadss
##   <chr>          <dbl>
## 1 FEMALE          624.
## 2 MALE            616.
```

5. Use `n_distinct()` inside of `summarize` to calculate the number of unique schools in the study.

```
# R code here
#number of unique schools
STAR_g3 %>%
  summarise(number_of_unique_schools = n_distinct(g1schid))
```

```
##   number_of_unique_schools
## 1                       64
```

6. Use `arrange` to sort the dataset of black females born in July of 1979 from highest to lowest math score.

```
# R code here
#arrange student in a descending order using math score
Fstud<- Fstud%>%
  arrange(desc(g3tmathss))
head(Fstud$g3tmathss,10)
```

```
##  [1] 626 616 604 601 570 568 558 558 554 554
```

7. Building from your answer to Q4, use `filter` and `summarize` to calculate the average reading scores (`g3treadss`) for males and females from urban schools.

```r
# R code here
#average reading scores (`g3treadss`) for males and females from urban schools
STAR_g3[, c("g3treadss", "gender","g1surban")]%>%
  drop_na(gender)%>%
  filter(g1surban=="URBAN")%>%
  group_by(gender)%>%
  summarise(avg_g3treadss= mean(g3treadss, na.rm = T))
```

```
## # A tibble: 2 × 2
##   gender avg_g3treadss
##   <chr>          <dbl>
## 1 FEMALE          621.
## 2 MALE            613.
```

8. Use `group_by` to calculate the average test scores (including reading, math, language, and listening) for males and females from urban schools. The result should be a `dataframe` with a row for males and a row for females and separate columns containing average test scores for each of the four domains.

```r
# R code here
STAR_g3[, c("g3tmathss","g3tlangss","g3tlangss" ,
            "g3treadss", "gender","g1surban")]%>%
  drop_na(gender)%>%
  filter(g1surban=="URBAN")%>%
  group_by(gender)%>%
  summarise(avg_g3tmathss= mean(g3tmathss, na.rm = T),
            avg_g3tlangss= mean(g3tlangss, na.rm = T),
            avg_g3tlangss= mean(g3tlangss, na.rm = T),
            avg_g3treadss= mean(g3treadss, na.rm = T),
  )%>%
  as.data.frame()
```

```
##   gender avg_g3tmathss avg_g3tlangss avg_g3treadss
## 1 FEMALE      618.2377      637.8607      621.0410
## 2   MALE      616.9907      627.0841      613.1121
```

9. The `n()` function calculates the number of observations in a dataset. It is useful inside of `summarize`, particularly when combined with `group_by`, for calculating the number of observations within levels of a grouping variable. Use these functions to calculate the number of students in the study born in each year from 1977 to 1981.

```r
# R code here
#number of students in the study born in each year from 1977 to 1981

students_by_birth_year <- STAR_g3%>%
  filter(birthyear >= 1977 & birthyear <= 1981) %>%  # Filtering data for the specifi
ed years
  group_by(birthyear) %>%
  summarise(number_of_students = n())
students_by_birth_year
```

```
## # A tibble: 5 × 2
##   birthyear number_of_students
##       <int>              <int>
## 1      1977                 10
## 2      1978                203
## 3      1979               1857
## 4      1980               3684
## 5      1981                 10
```

10. Create a dataset that reports the number of schools and the number of students by urbanicity. Use `g1surban` to define urbanicity, but combine `INNER CITY` and `URBAN` schools into a single category.

```
# R code here
urbanicity_data <- STAR_g3 %>%
  mutate(g1surban = ifelse(g1surban %in% c("INNER CITY", "URBAN"),
                                 "URBAN", as.character(g1surban))) %>%
  group_by(g1surban) %>%
  summarise(number_of_schools = n_distinct(g1schid),
            number_of_students = n())

urbanicity_data
```

```
## # A tibble: 3 × 3
##   g1surban number_of_schools number_of_students
##   <chr>                <int>              <int>
## 1 RURAL                   34               2957
## 2 SUBURBAN                15               1403
## 3 URBAN                   15               1428
```

11. Create a dataset that reports the proportion of white students, proportion of black students, and proportion of students from any other races within each school. Use the result to identify the ten schools with the largest percentage of black students. (Print out the result.) Also, answer: Are these schools in urban, suburban, or rural areas?

```
# R code here
school_proportions <- STAR_g3 %>%
  group_by(g1schid, race) %>%
  summarise(count = n())%>%
  mutate(prop = count / sum(count),
         count=NULL)%>%
  pivot_wider(names_from = race, values_from =prop, values_fill = 0)
```

```
## `summarise()` has grouped output by 'g1schid'. You can override using the
## `.groups` argument.
```

```
school_proportions%>%
  arrange(desc(BLACK))%>%
  select(g1schid,BLACK)%>%
  head(10)
```

```
## # A tibble: 10 × 2
## # Groups:   g1schid [10]
##    g1schid BLACK
##      <int> <dbl>
##  1  205490 1
##  2  244736 1
##  3  244746 1
##  4  244780 1
##  5  244806 1
##  6  244774 0.992
##  7  244727 0.990
##  8  244708 0.984
##  9  244723 0.984
## 10  244776 0.976
```

12. Let X_{ij} denote a score for unit i in school j. The *group-mean centered* variable $\tilde{X}_{ij}$ is calculated by subtracting the mean for all units in a given school j, $\bar{X}_j = \sum_{i=1}^{n_j} X_{ij}$, from the raw score:

$$\tilde{X}_{ij} = X_{ij} - \bar{X}_j.$$

Calculate the group-mean centered test scores for each of the four test score variables (i.e., 4 variables), **centering by school**. Use the `ungroup()` function so that the resulting dataset does not include any grouping variables.

Print out the first 10 rows of the dataset, selecting the new centered variables.

```
# R code here
cen_scores <- STAR_g3 %>%
  group_by(g1schid) %>%
  drop_na(starts_with("g3"))%>%
  mutate(
    g3treadss_centered = g3treadss - mean(g3treadss, na.rm = T),
    g3tmathss_centered = g3tmathss - mean(g3tmathss, na.rm = T),
    g3tlangss_centered = g3tlangss - mean(g3tlangss, na.rm = T),
    g3tlistss_centered = g3tlistss - mean(g3tlistss, na.rm = T)
  ) %>%
  ungroup()

# Print out the first 10 rows with the new centered variables
head(cen_scores[, c("g3treadss_centered", "g3tmathss_centered", "g3tlangss_centered",
"g3tlistss_centered")], 10)
```

```
## # A tibble: 10 × 4
##    g3treadss_centered g3tmathss_centered g3tlangss_centered g3tlistss_centered
##                 <dbl>              <dbl>              <dbl>              <dbl>
##  1             -14.2               -9.33             -27                 20.4 
##  2             -50.0              -62.6              -65.4              -40.7 
##  3             -54.8              -18.1              -24.3               37.0 
##  4             -21.3              -56.3              -27.8                3.70
##  5             -43.0                8.34             -17.7              -19.8 
##  6               7.71              15.9               27.8              -17.1 
##  7              17.1               22.1               21.1               35.0 
##  8              21.5               24.0               30                  6.82
##  9             -47.9              -53.1              -51.1              -21.4 
## 10             -49.2              -24.2              -55.2                5.36
```

13. Calculate the correlations among the four group-mean centered test score variables (these are equivalent to the partial correlations, after controlling for the students' schools). How do these partial correlations compare to the corresponding bi-variate correlations among the raw test score variables?

```
# R code here
#partial correlations
partial_corr<- cor(cen_scores %>% select(ends_with("centered")), method = "pearson")
partial_corr
```

```
##                    g3treadss_centered g3tmathss_centered g3tlangss_centered
## g3treadss_centered          1.0000000          0.7146886          0.7842844
## g3tmathss_centered          0.7146886          1.0000000          0.7349754
## g3tlangss_centered          0.7842844          0.7349754          1.0000000
## g3tlistss_centered          0.6167166          0.6096576          0.5434882
##                    g3tlistss_centered
## g3treadss_centered          0.6167166
## g3tmathss_centered          0.6096576
## g3tlangss_centered          0.5434882
## g3tlistss_centered          1.0000000
```

```
bivariate_corr <- STAR_g3 %>% select(starts_with("g3t")) %>% drop_na()%>%
                                       cor(method = "pearson")

#bivariate correlations
bivariate_corr
```

```
##           g3treadss g3tmathss g3tlangss g3tlistss
## g3treadss 1.0000000 0.7443226 0.7988588 0.6535104
## g3tmathss 0.7443226 1.0000000 0.7549543 0.6473654
## g3tlangss 0.7988588 0.7549543 1.0000000 0.5741999
## g3tlistss 0.6535104 0.6473654 0.5741999 1.0000000
```

Partial and bivariate correlations, you'll notice that the partial correlations are generally slightly lower than the corresponding bivariate correlations.

14. *[Bonus +1]* Calculate the **pooled, within-school standard deviations** for each of the four test score variables, where the pooled within-school sample variance (the squared standard deviation) is defined

as

$$S_{within}^2 = \frac{1}{N-G} \sum_{j=1}^{G} \sum_{i=1}^{n_j} \left(X_{ij} - \bar{X}_j\right)^2,$$

where N is the total number of observations and G is the number of schools. (Note that S_{within}^2 is very similar to the sample variance of the group-mean centered test scores, but with degrees of freedom $N - G$ instead of $N - 1$. Consequently, one way to calculate S_{within}^2 is to calculate the sample variance of $\tilde{X}_{ij}$ and then multiply by $(N-1)/(N-G)$.)

```
# R code here
pooled_within_school_sd <- STAR_g3 %>%
  group_by(g1schid) %>%
  summarise(
    sd_g3treadss = sd(g3treadss, na.rm = T),
    sd_g3tmathss = sd(g3tmathss, na.rm = T),
    sd_g3tlangss = sd(g3tlangss, na.rm = T),
    sd_g3tlistss = sd(g3tlistss, na.rm = T)
  )

# Print out the result
pooled_within_school_sd
```

```
## # A tibble: 64 × 5
##    g1schid sd_g3treadss sd_g3tmathss sd_g3tlangss sd_g3tlistss
##      <int>        <dbl>        <dbl>        <dbl>        <dbl>
##  1  112038         31.7         31.2         32.8         32.2
##  2  123056         29.1         27.1         29.8         27.0
##  3  128076         29.7         33.4         29.3         30.2
##  4  128079         39.9         38.0         32.1         30.1
##  5  130085         29.8         39.7         28.9         35.1
##  6  159171         42.2         41.7         36.3         30.9
##  7  161176         37.5         33.9         37.4         27.9
##  8  161183         33.6         36.0         29.9         27.2
##  9  162184         47.7         46.7         47.9         31.1
## 10  164198         36.5         40.9         31.6         31.0
## # ℹ 54 more rows
```

15. *[Bonus +1]* Calculate the **group-mean centered and scaled** scores for each of the four test score variables, where the group-mean centered and scaled score is defined as the group-mean centered variables, scaled by the within-school standard deviation:

$$\hat{X}_{ij} = \frac{X_{ij} - \bar{X}_j}{S_{within}}$$

```
# R code here
centered_and_scaled_scores <- STAR_g3 %>%
  drop_na(starts_with("g3t"))%>%
  group_by(g1schid) %>%
  group_by(g1schid) %>%
  mutate(
    centered_and_scaled_g3treadss = (g3treadss - mean(g3treadss, na.rm = T)) / sd(g3t
readss, na.rm = T),
    centered_and_scaled_g3tmathss = (g3tmathss - mean(g3tmathss, na.rm = T)) / sd(g3t
mathss, na.rm = T),
    centered_and_scaled_g3tlangss = (g3tlangss - mean(g3tlangss, na.rm = T)) / sd(g3t
langss, na.rm = T),
    centered_and_scaled_g3tlistss = (g3tlistss - mean(g3tlistss, na.rm = T)) / sd(g3t
listss, na.rm = T)
  )


head(centered_and_scaled_scores)
```

```
## # A tibble: 6 × 17
## # Groups:   g1schid [6]
##   stdntid gender race  birthmonth birthday birthyear cmpstype g1schid g1surban
##     <int> <chr>  <chr> <chr>         <int>     <int> <chr>      <int> <chr>
## 1   10003 MALE   WHITE MAY              28      1980 REGULAR   257899 RURAL
## 2   10022 FEMALE BLACK OCTOBER          27      1980 SMALL     244801 SUBURBAN
## 3   10032 MALE   BLACK FEBRUARY         19      1979 REGULAR   244776 INNER CITY
## 4   10039 FEMALE WHITE NOVEMBER         12      1979 REGULAR   164198 RURAL
## 5   10042 FEMALE WHITE JULY             10      1979 REGULAR   203457 RURAL
## 6   10055 FEMALE WHITE DECEMBER          3      1979 AIDE      161176 RURAL
## # ℹ 8 more variables: g3treadss <int>, g3tmathss <int>, g3tlangss <int>,
## #   g3tlistss <int>, centered_and_scaled_g3treadss <dbl>,
## #   centered_and_scaled_g3tmathss <dbl>, centered_and_scaled_g3tlangss <dbl>,
## #   centered_and_scaled_g3tlistss <dbl>
```

16. *[Bonus +1]* Using the group-mean centered and scaled math scores, calculate the number of students from urban, suburban, and rural schools whose math test score is more than 2.5 standard deviations above or below the average test scores of students in their school.

```
# R code here
threshold <- 2.5

outliers <- centered_and_scaled_scores %>%
  filter(g1surban %in% c("URBAN","SUBURBAN", "RURAL"))%>%
  group_by(g1surban) %>%
  filter(
    centered_and_scaled_g3tmathss > threshold |
      centered_and_scaled_g3tmathss < -threshold
  ) %>%
  summarise(number_of_outliers = n())

outliers
```

```
## # A tibble: 3 × 2
##   g1surban number_of_outliers
##   <chr>                 <int>
## 1 RURAL                    21
## 2 SUBURBAN                 11
## 3 URBAN                     2
```

Visualization

Each question involves creating a visual representation of a dataset. In developing your visualizations, you should follow the principles of good statistical graphics by ensuring that:

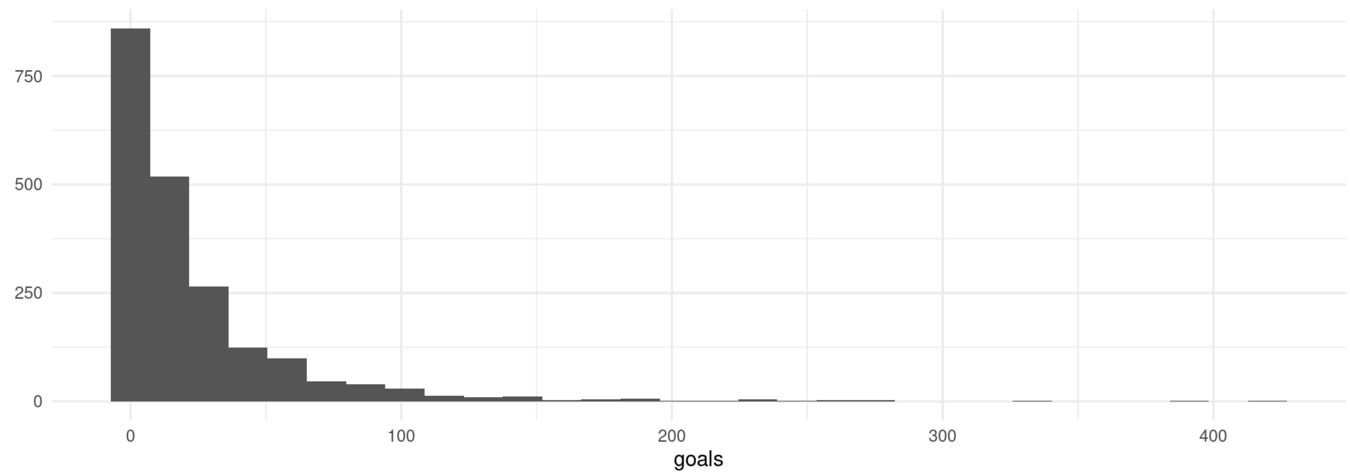
- axes, legends, and titles always have clear and sensible labels;
- the axes have sensible ranges and scales;
- relevant aspects of the data are not concealed (e.g., overlapping density plots are drawn so that each is clearly visible);
- it is easy to make comparisons between relevant quantities (e.g., categories to be compared should be close to each other);

In compiling the Rmarkdown file for this assignment, you might find it useful to change the default settings for the size of figures that are generated in a code chunk. For example, the following code displays a histogram of the number of goals scored by each player in the soccer fouls dataset. In the curly brackets at the beginning of the code chunk, I have set the `fig.width` and `fig.height` options to control the size of the resulting figure. Try changing the settings yourself and see what happens.

```
library(ggplot2)
qplot(goals, data = Soccer_fouls) + theme_minimal()
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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



Soccer data (5 pt)

This question deals with the data on soccer players and the number of yellow and red cards they receive, which is saved in the `soccer_fouls` data frame.

1. Calculate the average number of foul cards (of any color: `yellowCards`, `redCards`, and `yellowReds`) that each player receives **per game** (e.g., Bastian Schweinsteiger of Bayern-Muenchen had 93 cards in 611 games, or 0.1522095 cards per game).

```
average_cards_per_game <- Soccer_fouls%>%
  mutate(total_cards = yellowCards + redCards + yellowReds,
         cards_per_game= total_cards/games) #%>%

head(average_cards_per_game[,c("player","cards_per_game")],10)
```

```
##                    player cards_per_game
## 1            Aaron Hughes     0.02905199
## 2              Aaron Hunt     0.12797619
## 3            Aaron Lennon     0.02669903
## 4            Aaron Ramsey     0.12307692
## 5  Abdelhamid El-Kaoutari     0.11290323
## 6             Abdón Prats     0.16666667
## 7            Abdou Dampha     0.20454545
## 8            Abdou Traoré     0.12371134
## 9           Abdoul Camara     0.02112676
## 10      Abdoulaye Diallo     0.00000000
```

2. Create a density plot of the distribution of this quantity, with separate densities displayed for each league/Country.

```
# Create a density plot
ggplot(average_cards_per_game, aes(x = cards_per_game, fill = leagueCountry)) +
  geom_density(alpha = 0.5) +
  labs(title = "Average Foul Cards per Game",
       x = "Average Foul Cards per Game",
       caption = "fig:1 Density Plot of Average Foul Cards per Game by League/Countr
y",
       y = "Density") +
  theme_minimal()
```

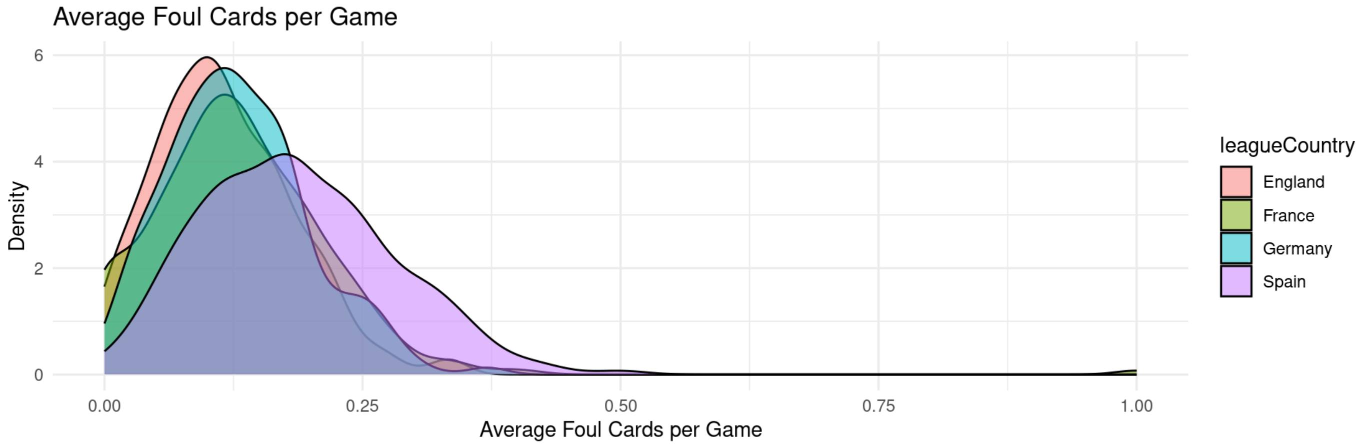


fig:1 Density Plot of Average Foul Cards per Game by League/Country

3. Check for outliers (e.g., outside of 1st and 3rd quartiles; or any operational definition of outliers from you) and revise your graph accordingly.

```
##find Q1, Q3, and interquartile range for values in column cards_per_game
Q1 <- quantile(average_cards_per_game$cards_per_game, .25)
Q3 <- quantile(average_cards_per_game$cards_per_game, .75)
IQR <- IQR(average_cards_per_game$cards_per_game)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
no_outliers <- subset(average_cards_per_game, average_cards_per_game$cards_per_game>
(Q1 - 1.5*IQR) & average_cards_per_game$cards_per_game< (Q3 + 1.5*IQR))

# Create a density plot
ggplot(no_outliers, aes(x = cards_per_game, fill = leagueCountry)) +
  geom_density(alpha = 0.5) +
  labs(title = "Average Foul Cards per Game",
       x = "Average Foul Cards per Game",
       caption = "fig:2 Density Plot of Average Foul Cards per Game by League/Country
with no outliers",
       y = "Density") +
  theme_minimal()
```

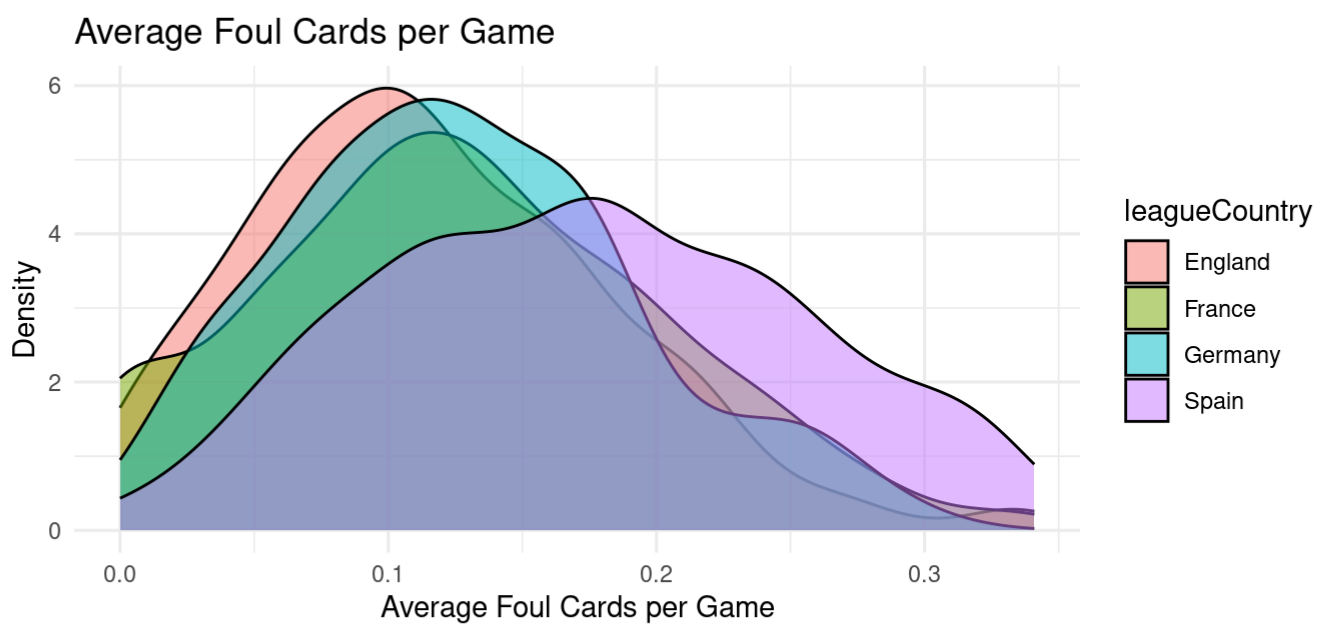


fig:2 Density Plot of Average Foul Cards per Game by League/Country with no outliers

4. Interpret the graph: Which league appears to have the highest average number of fouls per game?

Explain your interpretation here. The English league has the highest average number of fouls per game. This is indicated by the highest density at around 0.2 average foul cards per game for the English league (represented in green). The French and Spanish leagues have lower densities, indicating fewer average fouls per game.

Ways to improve public education (4 pt)

This question deals with the data from the State of the City poll conducted by the Pew Center, which is saved in the `City_poll` data frame.

Questions 18a through 18d all start with the stem “Please tell me what impact you think each of the following changes would have on improving the quality of public education in your community….” The responses were coded so that 1 = “Better,” 2 = “About the same,” 3 = “Worse,” 8 = “Don’t know,” and 9 = “Refused.”

1. Reformat these variables as factors with informative labels. (NOTE. Don't forget to re-code missing data.)

```
df <- City_poll%>%
  mutate_at(vars(q18a:q18d), ~factor(.x,
                                     levels = c(1, 2, 3, 8, 9),
                                     labels = c("Better", "About the same", "Worse",
"Don't know", "Refused"),
                                     exclude = NULL))

df%>%
  select(q18a:q18d)%>%
  head()
```

```
##             q18a   q18b   q18c           q18d
## 1          Worse Better Better About the same
## 2 About the same Better Better     Don't know
## 3 About the same  Worse Better About the same
## 4         Better Better Better         Better
## 5 About the same Better Better         Better
## 6 About the same Better Better         Better
```

2. Create a bar chart or set of bar charts that display the distribution of responses to each of these questions. The figure should make it easy to compare the respondents' views about which proposals would be effective.

```
p1<-df %>%
  select(q18a, q18b, q18c, q18d) %>%
  gather(key = "Question", value = "Response") %>%
  ggplot(aes(x = Response, fill = Response)) +
  geom_bar() +
  facet_wrap(~ Question, scales = "free_y") +
  theme_minimal() +
  labs(x = "Response", y = "Count", fill = "Response",
       title = "Distribution of Responses to Questions 18a through 18d",
       caption  = "fig3: Please tell me what impact you think each of the following c
hanges would have on improving the quality of public education in your community")
```

3. Interpret the graph: Which of the proposals is most widely view as effective? Which is most widely viewed as detrimental?

```
# R code here
p1
```

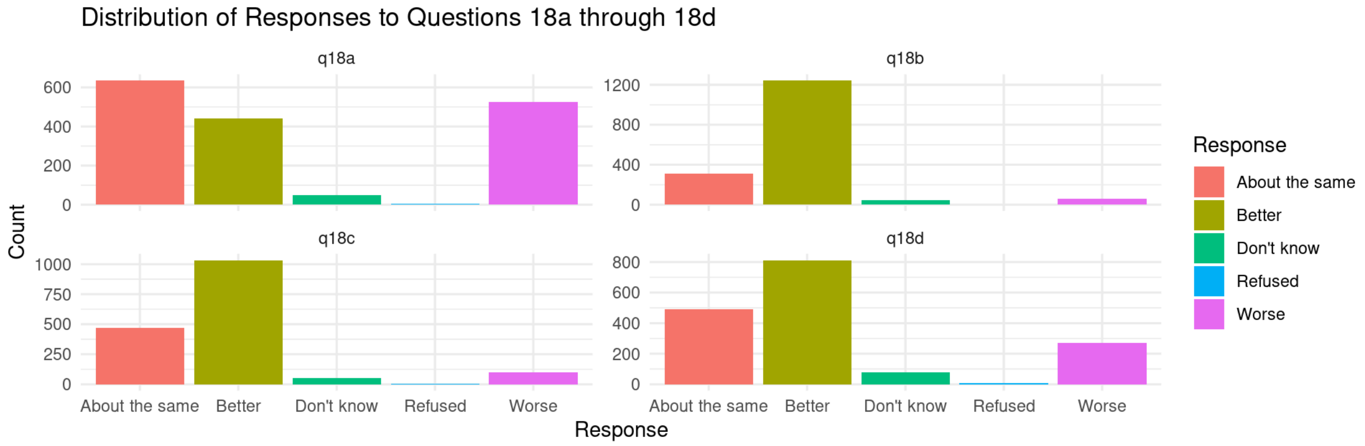


fig3: Please tell me what impact you think each of the following changes would have on improving the quality of public education in your community

Explain your interpretation here.

The proposal that is most widely viewed as effective is that students should have sames classes. Linking the pay for teachers and adminstartors to performance of the students is viewed as the most detrimental proposals.

Differing views by party (3 pt)

Continuing with the same data as in the previous question, I have created the variable `party_leaning` that captures respondents' political orientation (it is a composite of the variables `party` and `partyln`).

```
table(City_poll$party_leaning, useNA = "always")
```

```
##
## Republican Democratic      Other Don't know    Refused       <NA>
##        588        769        102        110         87          0
```

1. Create a bar chart or set of bar charts that display the distribution of responses to each of the *education items* (Q18) by political orientation, so that it is possible to compare differences between Democratic-leaning and Republican-leaning respondents' opinions about the effectiveness of each policy. You can exclude the response categories other than "Democratic" and "Republican." (i.e., Treat thoes as `NA`)

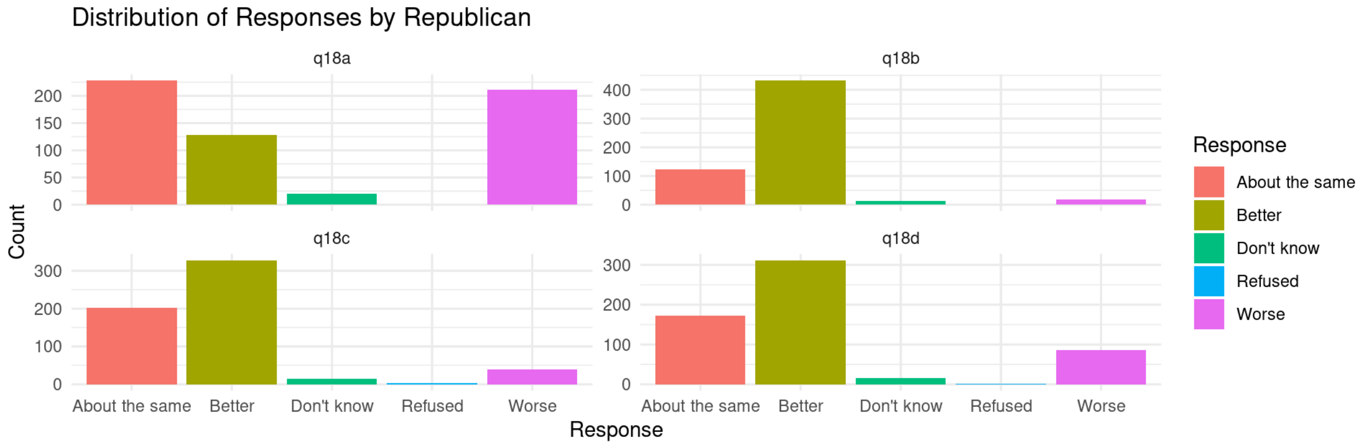


fig4: Please tell me what impact you think each of the following changes would have on improving the quality of public education in your community

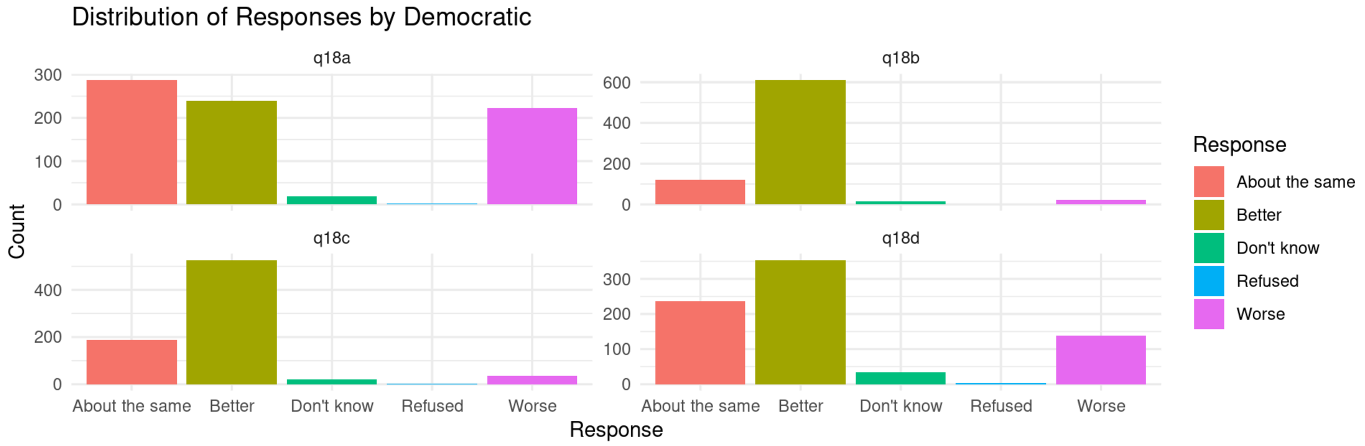


fig 5: Please tell me what impact you think each of the following changes would have on improving the quality of public education in your community

2. Interpret the graph: Which of the proposals is most widely view as effective by Republicans? by Democrats? For which of the proposals is there the greatest discrepancy between the opinions of Democrats and Republicans?

Explain your interpretation here. Both republicans and Democratic seems to agree on most of the proposals. There are no major discrepancies between the two parties. Both parties feel that having smaller classes is more effective. And are all aganist having longer school days.