DSCI 401 HW 2

Isabel Heard

09/22/2023

Contents

1

Using the Teams data frame in the Lahman package:

```
#install.packages('Lahman')
library(Lahman)
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

A

Create a data frame that is a subset of the Teams data frame that contains only the years from 2000 through 2009 and the variables yearID, W, and L.

```r
#Import data
data(Teams)
#head(Teams)
#summary(Teams)

#Create a subset of the data
df.sub <- subset(Teams, yearID >= 2000 & yearID <= 2009, select = c("yearID", "W", "L"))
summary(df.sub)
```

```
##      yearID           W               L
##  Min.   :2000   Min.   : 43.00   Min.   : 46.00
##  1st Qu.:2002   1st Qu.: 72.00   1st Qu.: 72.00
##  Median :2004   Median : 82.00   Median : 80.00
##  Mean   :2004   Mean   : 80.95   Mean   : 80.95
##  3rd Qu.:2007   3rd Qu.: 90.00   3rd Qu.: 90.00
##  Max.   :2009   Max.   :116.00   Max.   :119.00
```

B

How many years did the Chicago Cubs (teamID is "CHN") hit at least 200 HRs in a season and what was the median number of wins in those seasons.

```r
#Filter on teamID, group by yearID, calculate median number of wins on seasons where HRs > 200
cubs <- Teams %>%
  filter(teamID == "CHN") %>%
  group_by(yearID) %>%
  summarize(total_HRs = sum(HR), median_wins = median(W)) %>%
  filter(total_HRs >= 200)

#Count the number of years
num_years_with_200_HRs <- nrow(cubs)
cat("Number of years with at least 200 HRs:", num_years_with_200_HRs, "\n")
```

```
## Number of years with at least 200 HRs: 7
```

```r
#Median number of wins in those seasons
cat("Median number of wins in those seasons:", median(cubs$median_wins), "\n")
```

```
## Median number of wins in those seasons: 84
```

C

Create a factor called election that divides the yearID into 4-year blocks that correspond to U.S. presidential terms. The first presidential term started in 1788. They each last 4 years and are still on the schedule set in 1788. During which term were the most home runs been hit?

```r
#Find the start years of each term
start_years <- seq(1788, max(Teams$yearID), by = 4)

#Add a small offset to the start_years to ensure uniquenes
start_years <- start_years + 0.001

#Create labels for the presidential terms
term_labels <- paste("Term", 1:length(start_years))

# Create the "election" factor variable
```

```
Teams$election <- cut(Teams$yearID, breaks = c(1788, start_years), labels = term_labels, right = FALSE)

#Find the term with the most home runs
term_with_most_home_runs <- aggregate(HR ~ election, Teams, sum)
term_with_most_home_runs <- term_with_most_home_runs[which.max(term_with_most_home_runs$HR), ]

cat("The most home runs were hit during term", term_with_most_home_runs$election, "- 2000-2004 \n")
```

```
## The most home runs were hit during term 55 - 2000-2004
```

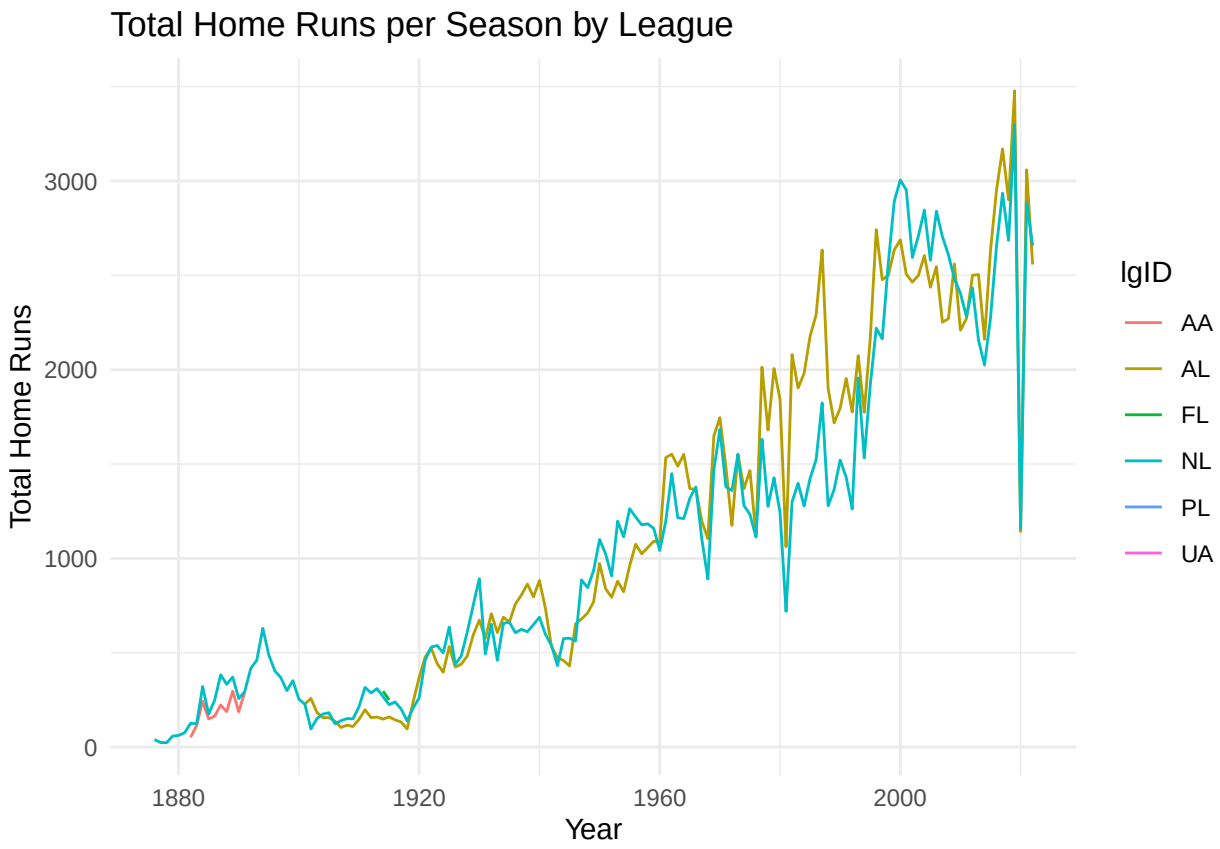
D

Make a line plot of total home runs per season and stratify by league. Remove observations where league is missing.

```
library(ggplot2)
#Remove missing values
df <- subset(Teams, !(lgID == "NA"))

#Group the data by year and league, calculate the total home runs per season (year)
hr_per_season <- df %>% group_by(yearID, lgID) %>% summarize(total_home_runs = sum(HR))
```

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

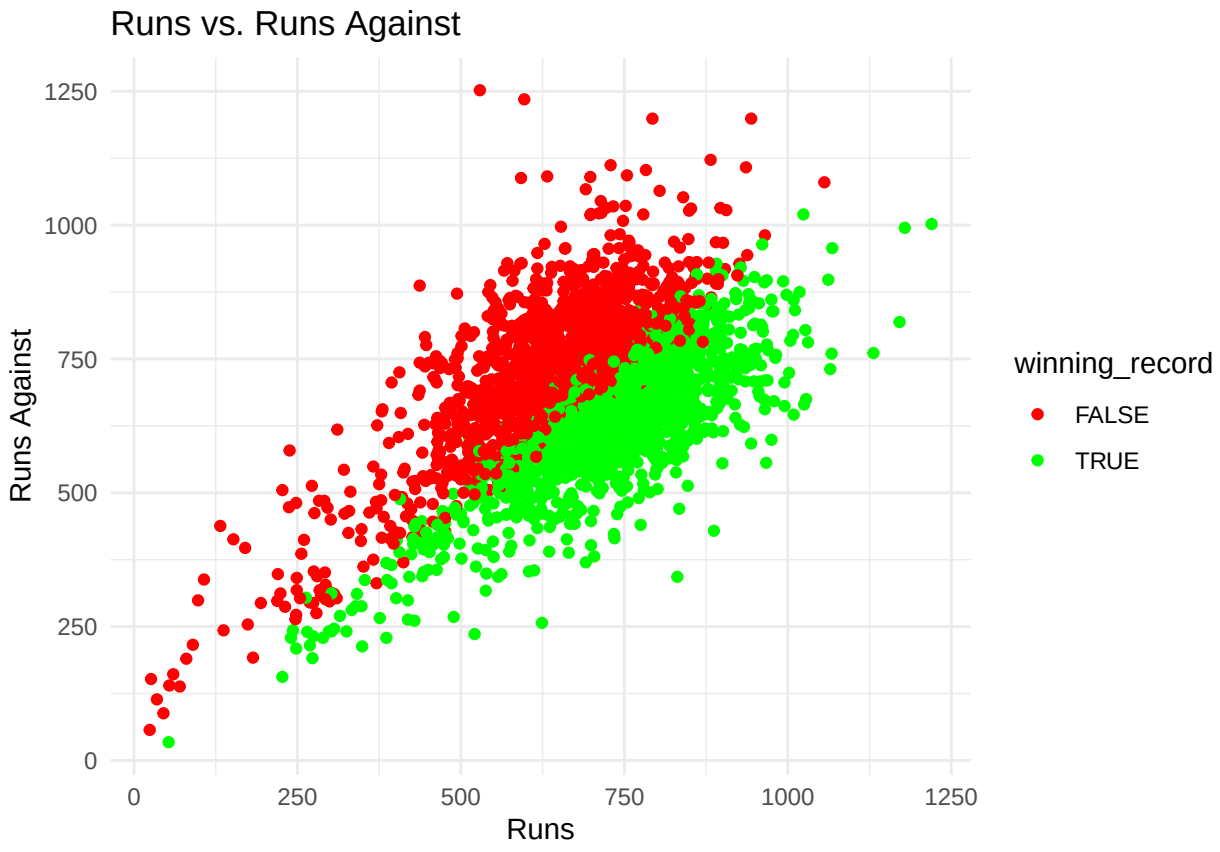
```
ggplot(hr_per_season, aes(x = yearID, y = total_home_runs, color = lgID)) +
  geom_line() + labs(x = "Year", y = "Total Home Runs", title = "Total Home Runs per Season by League") +
  theme_minimal()
```



E

Create an indicator variable called "winning record" which is defined as TRUE if the number of wins is greater than the number of losses and FALSE otherwise. Plot a scatter plot of Runs (R) vs Runs against (RA) with the color of each point showing whether that team had a winning record or not.

```
#Create "winning_record" variable
Teams <- Teams %>% mutate(winning_record = W > L)

#Scatter plot of R vs RA
ggplot(Teams, aes(x = R, y = RA, color = winning_record)) +
  geom_point() +
  scale_color_manual(values = c("TRUE" = "green", "FALSE" = "red")) +
  labs(x = "Runs", y = "Runs Against", title = "Runs vs. Runs Against") +
  theme_minimal()
```



2

The Violations data set in the mdsr package contains information regarding the outcome of health inspections of restaurants in New York City.

```
#install.packages('mdsr')
library(mdsr)

data(Violations)
#head(Violations)
#tail(Violations)
```

Write out Violations csv file to work with in python.

```
#data(Violations)
#csv_filename <- "Violations.csv"
#write.csv(Violations, file = csv_filename, row.names = FALSE)
```

A

What proportion of inspections in each boro were given a grade of A? (Missing values should be counted as not and A grade.)

```
#Group data by boro and grade, and count grade
grade_counts <- Violations %>% group_by(boro, grade) %>% summarize(count = n())
```

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

```
#Filter where grade is A
grade_A_counts <- grade_counts %>% filter(grade == "A")

#Find total inspections in each boro
total_counts <- grade_counts %>% group_by(boro) %>% summarize(total_count = sum(count))

#Calculate the proportion of A grades
result <- merge(grade_A_counts, total_counts, by = "boro") %>% mutate(proportion_A = count / total_coun
print(result)
```

```
##            boro grade count total_count proportion_A
## 1         BRONX     A 15167       43112    0.3518046
## 2      BROOKLYN     A 38153      115659    0.3298749
## 3     MANHATTAN     A 66484      194130    0.3424715
## 4       Missing     A    33          73    0.4520548
## 5        QUEENS     A 36731      111660    0.3289540
## 6 STATEN ISLAND     A  5794       15987    0.3624195
```

B

Find the top ten dba's with the most number of inspections. Then compute the average score for each of these dba's and sort by mean score. Which of these top 10 had the lowest average inspection score?

```
#Get dba by number of inspections
dba_counts <- Violations %>% group_by(dba) %>% summarize(inspection_count = n())

#Top ten dba's based on inspections
top_10_dbas <- dba_counts %>% arrange(desc(inspection_count)) %>% head(10)

#Filter the original data frame to keep only the rows for the top 10 DBAs
filtered_df <- Violations %>% filter(dba %in% top_10_dbas$dba)

#Mean score for each of the top 10
average_scores <- filtered_df %>% group_by(dba) %>% summarize(mean_score = mean(score, na.rm = TRUE)) %

#Find store with lowest inspection score
lowest_average_score_dba <- average_scores$dba[1]
cat("DBA with the lowest average inspection score:", lowest_average_score_dba, "\n")
```

```
## DBA with the lowest average inspection score: STARBUCKS
```

```r
#All top ten average scores
print(average_scores)
```

```
## # A tibble: 10 x 2
##    dba                                  mean_score
##    <chr>                                     <dbl>
##  1 STARBUCKS                                  11.7
##  2 DUNKIN' DONUTS                             13.6
##  3 DUNKIN' DONUTS, BASKIN ROBBINS             14.5
##  4 SUBWAY                                     14.8
##  5 MCDONALD'S                                 17.0
##  6 BURGER KING                                17.3
##  7 KENNEDY FRIED CHICKEN                      18.0
##  8 GOLDEN KRUST CARIBBEAN BAKERY & GRILL      18.3
##  9 CROWN FRIED CHICKEN                        18.6
## 10 DOMINO'S                                   18.6
```

C

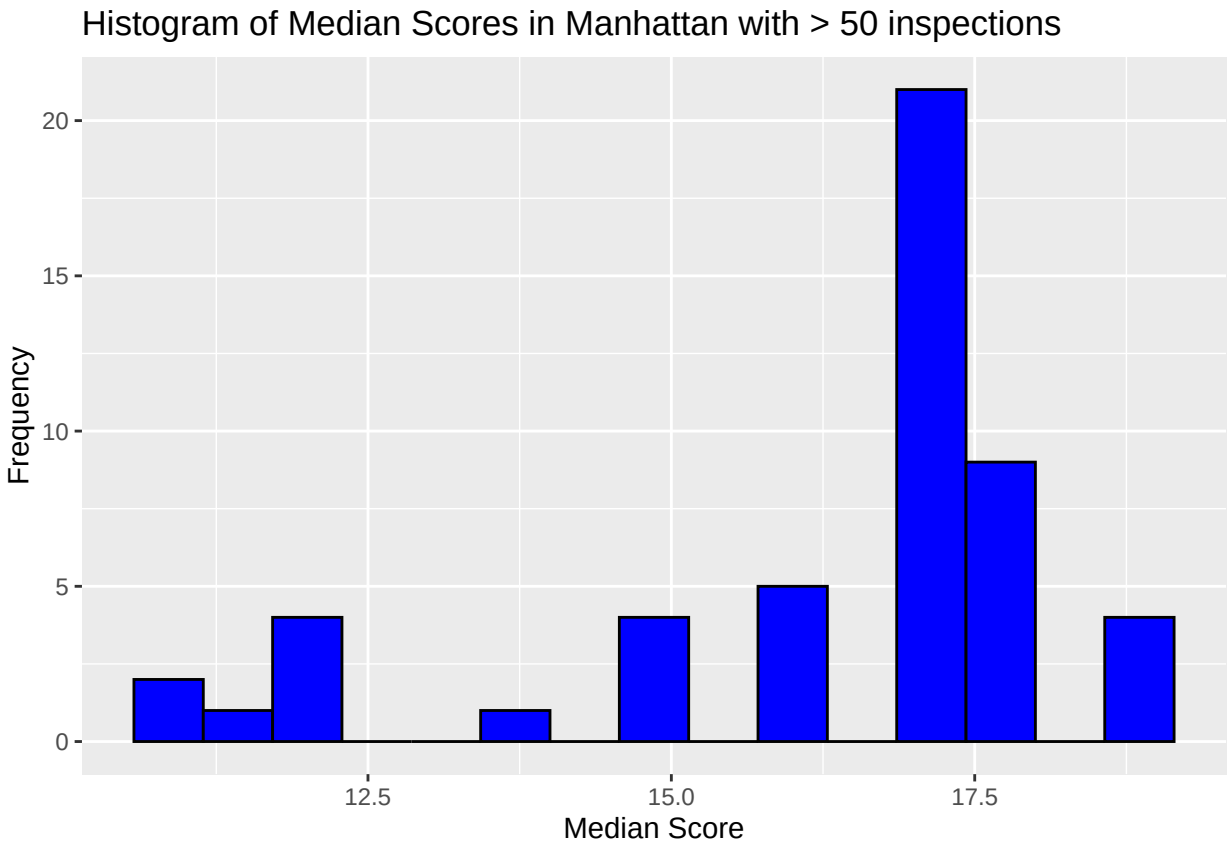
Use these data to calculate the median violation score by zip code for zip codes in Manhattan with 50 or more inspections. What pattern do you see between the number of inspections and the median score?

```r
#Filter for Manhattan
manhattan_df <- Violations %>% filter(boro == "MANHATTAN")

#Group the data by 'zip_code' and calculate the count of inspections for each zip code
zip_code_counts <- manhattan_df %>%
  group_by(zipcode) %>%
  summarize(inspection_count = n())

#Filter to keep only zip codes with 50 or more inspections
zip_codes_50_or_more <- zip_code_counts %>%
  filter(inspection_count >= 50)

# Filter the original data frame to keep only the rows for zip codes with 50 or more inspections
filtered_df <- Violations %>%
  filter(zipcode %in% zip_codes_50_or_more$zipcode)

# Calculate the median violation score by zip code
median_scores <- filtered_df %>%
  group_by(zipcode) %>%
  summarize(
    median_score = median(score, na.rm = TRUE),
    inspection_count = n())

print(median_scores)
```

```
## # A tibble: 51 x 3
##    zipcode median_score inspection_count
##      <int>        <dbl>            <int>
##  1   10001           15             8420
##  2   10002           18             9113
##  3   10003           17            13556
##  4   10004           14             2304
##  5   10005           17             1209
```

```
##  6   10006           17             977
##  7   10007           16            2356
##  8   10009           17            6131
##  9   10010           17            4658
## 10   10011           17            8790
## # ... with 41 more rows
```

```
ggplot(median_scores, aes(x = median_score)) +
  geom_histogram(bins = 15, fill = "blue", color = "black") +
  labs(x = "Median Score", y = "Frequency", title = "Histogram of Median Scores in Manhattan with > 50 i
```



The more inspections, the higher the median score.

Link to Colab

https://colab.research.google.com/drive/1uX6_pM1eR8yPIt4egT1LImYel8nvPBr_?usp=sharing