DSCI401_HW2_LexBrunett

Lex Miguel Brunett Chavez

2023-10-03

R Markdown Homework 2

Homework 2 should be submitted as an R Markdown file with links to Google colab notes where necessary. Homework should be turned in on Sakai, all the code and other copies are in my github repository of the class.

```
Teams <- read.csv("https://raw.githubusercontent.com/gjm112/DSCI401/main/data/Teams.csv")
Violations <- read.csv("C:/Users/gatit/Downloads/Violations.csv")

if (!require(dplyr)) {
  install.packages("dplyr")
  library(dplyr)
}
```

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

```
if (!require(ggplot2)) {
  install.packages("ggplot2")
  library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

Using the Teams data frame in the Lahman package:

1. (10 points) 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.

We have to filter the dataframe to obtain only the years from 2000 through 2009.

```r
# library dplyr
if (!require(dplyr)) {
  install.packages("dplyr")
  library(dplyr)
}

# dataframe filtration
first_Question_Teams <- Teams%>%filter(yearID >= 2000 & yearID <= 2009) %>%select(yearID, W, L)

# printing the result
print(first_Question_Teams)
```

```
##     yearID   W   L
## 1     2000  82  80
## 2     2000  85  77
## 3     2000  95  67
## 4     2000  74  88
## 5     2000  85  77
## 6     2000  95  67
## 7     2000  65  97
## 8     2000  85  77
## 9     2000  90  72
## 10    2000  82  80
## 11    2000  79  83
## 12    2000  79  82
## 13    2000  72  90
## 14    2000  77  85
## 15    2000  86  76
## 16    2000  73  89
## 17    2000  69  93
## 18    2000  67  95
## 19    2000  87  74
## 20    2000  94  68
## 21    2000  91  70
## 22    2000  65  97
## 23    2000  69  93
## 24    2000  76  86
## 25    2000  91  71
## 26    2000  97  65
## 27    2000  95  67
## 28    2000  69  92
## 29    2000  71  91
## 30    2000  83  79
## 31    2001  75  87
## 32    2001  92  70
## 33    2001  88  74
## 34    2001  63  98
## 35    2001  82  79
## 36    2001  83  79
## 37    2001  88  74
## 38    2001  66  96
## 39    2001  91  71
## 40    2001  73  89
```

```
## 41    2001  66  96
## 42    2001  76  86
## 43    2001  93  69
## 44    2001  65  97
## 45    2001  86  76
## 46    2001  68  94
## 47    2001  85  77
## 48    2001  68  94
## 49    2001  95  65
## 50    2001  82  80
## 51    2001 102  60
## 52    2001  86  76
## 53    2001  62 100
## 54    2001  79  83
## 55    2001 116  46
## 56    2001  90  72
## 57    2001  93  69
## 58    2001  62 100
## 59    2001  73  89
## 60    2001  80  82
## 61    2002  99  63
## 62    2002  98  64
## 63    2002 101  59
## 64    2002  67  95
## 65    2002  93  69
## 66    2002  81  81
## 67    2002  67  95
## 68    2002  78  84
## 69    2002  74  88
## 70    2002  73  89
## 71    2002  55 106
## 72    2002  79  83
## 73    2002  84  78
## 74    2002  62 100
## 75    2002  92  70
## 76    2002  56 106
## 77    2002  94  67
## 78    2002  83  79
## 79    2002 103  58
## 80    2002  75  86
## 81    2002 103  59
## 82    2002  80  81
## 83    2002  72  89
## 84    2002  66  96
## 85    2002  93  69
## 86    2002  95  66
## 87    2002  97  65
## 88    2002  55 106
## 89    2002  72  90
## 90    2002  78  84
## 91    2003  77  85
## 92    2003  84  78
## 93    2003 101  61
## 94    2003  71  91
```

```
## 95    2003  95  67
## 96    2003  86  76
## 97    2003  88  74
## 98    2003  69  93
## 99    2003  68  94
## 100   2003  74  88
## 101   2003  43 119
## 102   2003  91  71
## 103   2003  87  75
## 104   2003  83  79
## 105   2003  85  77
## 106   2003  68  94
## 107   2003  90  72
## 108   2003  83  79
## 109   2003 101  61
## 110   2003  66  95
## 111   2003  96  66
## 112   2003  86  76
## 113   2003  75  87
## 114   2003  64  98
## 115   2003  93  69
## 116   2003 100  61
## 117   2003  85  77
## 118   2003  63  99
## 119   2003  71  91
## 120   2003  86  76
## 121   2004  92  70
## 122   2004  51 111
## 123   2004  96  66
## 124   2004  78  84
## 125   2004  98  64
## 126   2004  83  79
## 127   2004  89  73
## 128   2004  76  86
## 129   2004  80  82
## 130   2004  68  94
## 131   2004  72  90
## 132   2004  83  79
## 133   2004  92  70
## 134   2004  58 104
## 135   2004  93  69
## 136   2004  67  94
## 137   2004  92  70
## 138   2004  67  95
## 139   2004 101  61
## 140   2004  71  91
## 141   2004  91  71
## 142   2004  86  76
## 143   2004  72  89
## 144   2004  87  75
## 145   2004  63  99
## 146   2004  91  71
## 147   2004 105  57
## 148   2004  70  91
```

```
## 149   2004  89  73
## 150   2004  67  94
## 151   2005  77  85
## 152   2005  90  72
## 153   2005  74  88
## 154   2005  95  67
## 155   2005  99  63
## 156   2005  79  83
## 157   2005  73  89
## 158   2005  93  69
## 159   2005  67  95
## 160   2005  71  91
## 161   2005  83  79
## 162   2005  89  73
## 163   2005  56 106
## 164   2005  95  67
## 165   2005  71  91
## 166   2005  81  81
## 167   2005  83  79
## 168   2005  95  67
## 169   2005  83  79
## 170   2005  88  74
## 171   2005  88  74
## 172   2005  67  95
## 173   2005  82  80
## 174   2005  69  93
## 175   2005  75  87
## 176   2005 100  62
## 177   2005  67  95
## 178   2005  79  83
## 179   2005  80  82
## 180   2005  81  81
## 181   2006  76  86
## 182   2006  79  83
## 183   2006  70  92
## 184   2006  86  76
## 185   2006  90  72
## 186   2006  66  96
## 187   2006  80  82
## 188   2006  78  84
## 189   2006  76  86
## 190   2006  95  67
## 191   2006  78  84
## 192   2006  82  80
## 193   2006  62 100
## 194   2006  89  73
## 195   2006  88  74
## 196   2006  75  87
## 197   2006  96  66
## 198   2006  97  65
## 199   2006  97  65
## 200   2006  93  69
## 201   2006  85  77
## 202   2006  67  95
```

```
## 203   2006  88  74
## 204   2006  78  84
## 205   2006  76  85
## 206   2006  83  78
## 207   2006  61 101
## 208   2006  80  82
## 209   2006  87  75
## 210   2006  71  91
## 211   2007  90  72
## 212   2007  84  78
## 213   2007  69  93
## 214   2007  96  66
## 215   2007  72  90
## 216   2007  85  77
## 217   2007  72  90
## 218   2007  96  66
## 219   2007  90  73
## 220   2007  88  74
## 221   2007  71  91
## 222   2007  73  89
## 223   2007  69  93
## 224   2007  94  68
## 225   2007  82  80
## 226   2007  83  79
## 227   2007  79  83
## 228   2007  94  68
## 229   2007  88  74
## 230   2007  76  86
## 231   2007  89  73
## 232   2007  68  94
## 233   2007  89  74
## 234   2007  88  74
## 235   2007  71  91
## 236   2007  78  84
## 237   2007  66  96
## 238   2007  75  87
## 239   2007  83  79
## 240   2007  73  89
## 241   2008  82  80
## 242   2008  72  90
## 243   2008  68  93
## 244   2008  95  67
## 245   2008  89  74
## 246   2008  97  64
## 247   2008  74  88
## 248   2008  81  81
## 249   2008  74  88
## 250   2008  74  88
## 251   2008  84  77
## 252   2008  86  75
## 253   2008  75  87
## 254   2008 100  62
## 255   2008  84  78
## 256   2008  90  72
```

```
## 257   2008  88  75
## 258   2008  89  73
## 259   2008  89  73
## 260   2008  75  86
## 261   2008  92  70
## 262   2008  67  95
## 263   2008  63  99
## 264   2008  61 101
## 265   2008  72  90
## 266   2008  86  76
## 267   2008  97  65
## 268   2008  79  83
## 269   2008  86  76
## 270   2008  59 102
## 271   2009  70  92
## 272   2009  86  76
## 273   2009  64  98
## 274   2009  95  67
## 275   2009  79  83
## 276   2009  83  78
## 277   2009  78  84
## 278   2009  65  97
## 279   2009  92  70
## 280   2009  86  77
## 281   2009  87  75
## 282   2009  74  88
## 283   2009  65  97
## 284   2009  97  65
## 285   2009  95  67
## 286   2009  80  82
## 287   2009  87  76
## 288   2009 103  59
## 289   2009  70  92
## 290   2009  75  87
## 291   2009  93  69
## 292   2009  62  99
## 293   2009  75  87
## 294   2009  85  77
## 295   2009  88  74
## 296   2009  91  71
## 297   2009  84  78
## 298   2009  87  75
## 299   2009  75  87
## 300   2009  59 103
```

2. (10 points) 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.

```
# filtering values for team
ChicagoCubs <- Teams %>%filter(teamID == "CHN" & HR >= 200) %>%select(yearID, HR, W)
```

```r
# counting years
years <- nrow(ChicagoCubs)

# printing the result
print(years)
```

```
## [1] 7
```

```r
median_W <- median(ChicagoCubs$W)

# printing the median
print(median_W)
```

```
## [1] 84
```

The chicago Cubs hit at least 200 HRs in a season during **7** years, in the other hand the median number of the wins in those seasons were **84** wins.

3. (10 points) 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
# pbtaining the max value
Top_Year <- Teams$yearID[which.max(Teams$HR)]

start_year <- 1788

Teams$election <- (Teams$yearID - start_year) %/% 4


library(dplyr)
election_totals <- Teams %>%group_by(election) %>%summarise(total_HR = sum(HR))


term_number <- election_totals %>%filter(total_HR == max(total_HR)) %>%pull(election)

print(term_number)
```

```
## [1] 57
```

```r
print(Top_Year)
```

```
## [1] 2019
```

The most Homeruns hit occured during the **57** st term, specifically in **2019** .

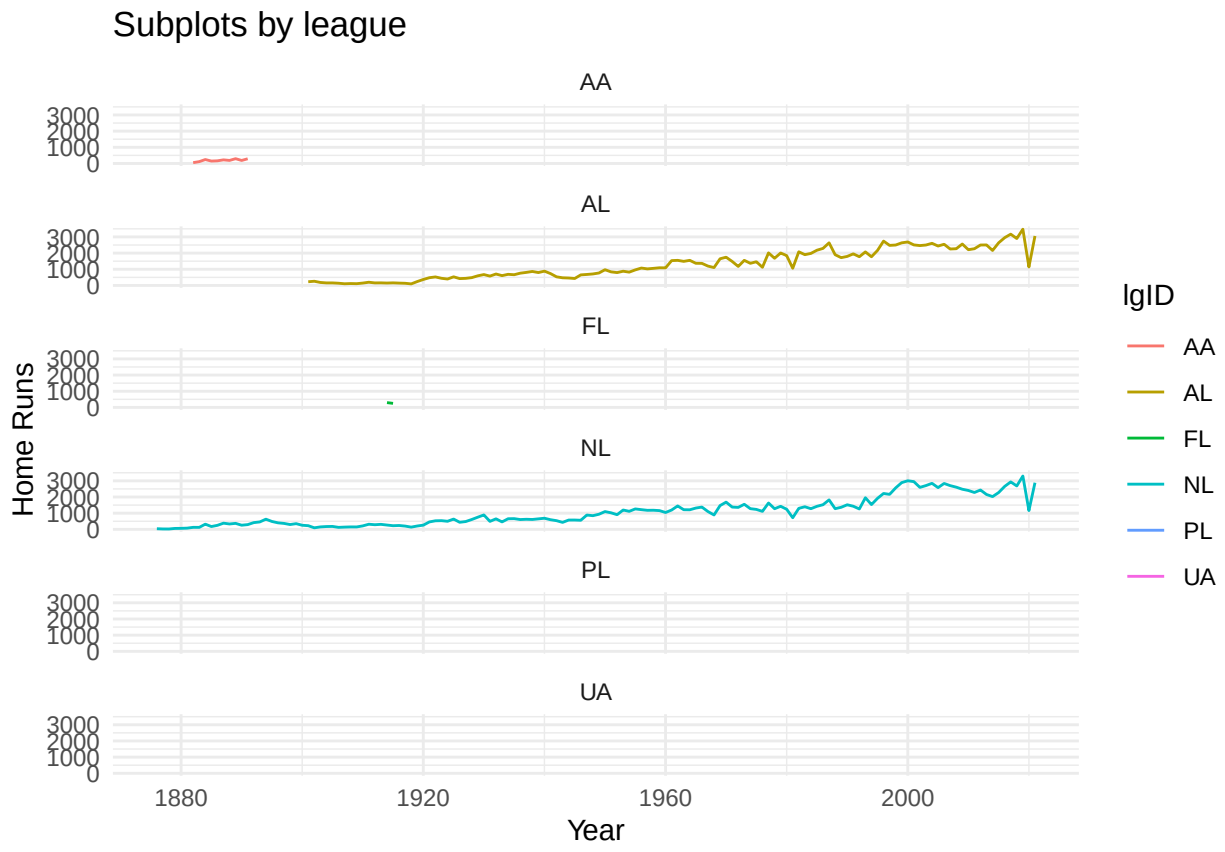
4. (10 points) Make a line plot of total home runs per season and stratify by league. Remove observations where league is missing

```
library(ggplot2)

pivot_data <- aggregate(HR ~ yearID + lgID, data = Teams, FUN = sum)

ggplot(pivot_data, aes(x = yearID, y = HR, group = lgID, color = lgID)) +
  geom_line() +
  facet_wrap(~ lgID, ncol = 1) +
  labs(x = "Year", y = "Home Runs", title = "Subplots by league") +
  theme_minimal()
```

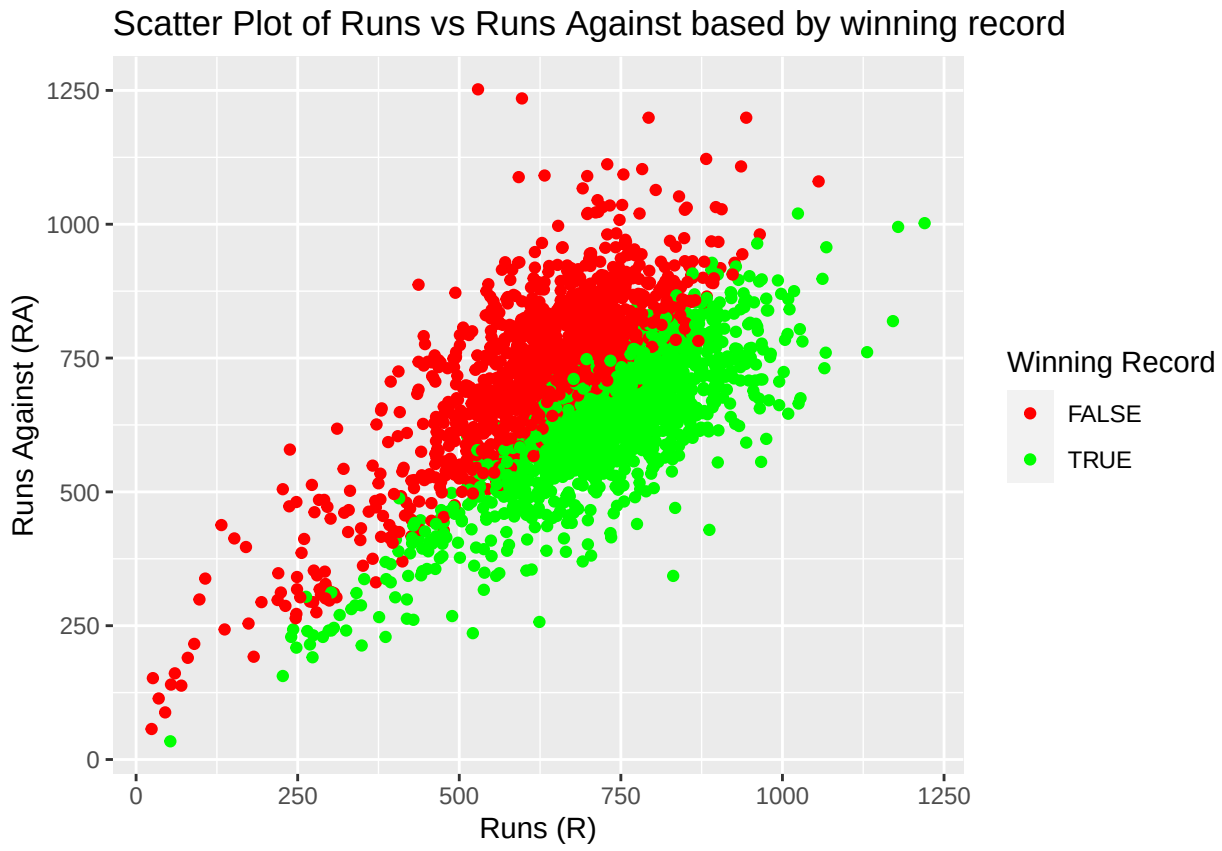
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



5. (10 points) 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.

```
Teams$winning_record <- Teams$W > Teams$L


ggplot(Teams, aes(x = R, y = RA, color = winning_record)) +
  geom_point() +
  labs(
    title = "Scatter Plot of Runs vs Runs Against based by winning record",
    x = "Runs (R)",
    y = "Runs Against (RA)",
    color = "Winning Record"
  ) +
  scale_color_manual(values = c("FALSE" = "red", "TRUE" = "green"))
```



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

6. (10 points) What proportion of inspections in each boron were given a grade of A? (Missing values should be counted as not and A grade.)

```
quantity_violations <- numeric()
quantity_inspections <- numeric()

Violations$GRADE[is.na(Violations$GRADE)] <- 0


Inspection <- Violations %>%
  filter(GRADE == "A") %>%
  select(BORO, GRADE)

inspection_set <- unique(Inspection$BORO)
boro_set <- unique(Violations$BORO)[-length(unique(Violations$BORO))]


total_inspection <- nrow(Violations)

for (x in 1:length(inspection_set)) {
  one_Inspection <- Inspection %>%
    filter(BORO == inspection_set[x]) %>%
    select(GRADE)
  r <- nrow(one_Inspection)
  quantity_inspections <- c(quantity_inspections, r)
}


for (x in 1:length(boro_set)) {
  proportion <- round((100 * quantity_inspections[x] / total_inspection), 2)
  cat(proportion, "%", boro_set[x], "\n")
}
```

```
## 8.05 % Queens
## 13.08 % Manhattan
## 8.89 % Brooklyn
## 2.96 % Bronx
## 1.4 % 0
```

7. (20 points) 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?

```
inspections_DBA <- table(Violations$DBA)
inspections_DBA <- head(sort(inspections_DBA, decreasing = TRUE), 10)
```

```r
df_top_10 <- Violations[Violations$DBA %in% names(inspections_DBA), ]


dba <- unique(df_top_10$DBA)
mean_list <- numeric()


for (x in 1:length(dba)) {
  one_dba <- df_top_10 %>%
    filter(DBA == dba[x]) %>%
    select(SCORE)
  one_mean <- mean(one_dba$SCORE, na.rm = TRUE)
  mean_list <- c(mean_list, one_mean)
}


dba_mean_df <- data.frame(DBA = dba, Mean_SCORE = mean_list)


dba_mean_df <- dba_mean_df[order(dba_mean_df$Mean_SCORE), ]


print(dba_mean_df)
```

```
##                                       DBA Mean_SCORE
## 4                                 STARBUCKS   11.79778
## 2                                    DUNKIN   13.49041
## 8                                 MCDONALD'S   13.72874
## 7                                   POPEYES   14.77258
## 5                                BURGER KING   15.54828
## 3                                    SUBWAY   17.23882
## 9  GOLDEN KRUST CARIBBEAN BAKERY & GRILL   19.86492
## 10                    CROWN FRIED CHICKEN   24.11565
## 6                  KENNEDY FRIED CHICKEN   24.40597
## 1                                             NaN
```

8. (20 points) 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
inspections_zipcode <- table(Violations$ZIPCODE)
inspections_zipcode <- inspections_zipcode[inspections_zipcode >= 50]


inspections_zipcode <- Violations[Violations$ZIPCODE %in% names(inspections_zipcode), ]


inspections_zipcode <- inspections_zipcode[inspections_zipcode$BORO == "Manhattan", c("DBA", "SCORE", "Z


zipcode <- unique(inspections_zipcode$ZIPCODE)
```

```r
median_list <- numeric()


for (x in 1:length(zipcode)) {
  one_zipcode <- inspections_zipcode %>%
    filter(ZIPCODE == zipcode[x]) %>%
    select(SCORE)
  one_median <- median(one_zipcode$SCORE, na.rm = TRUE)
  median_list <- c(median_list, one_median)
}


dba_median_df <- data.frame(ZIPCODE = zipcode, Median_SCORE = median_list)


dba_median_df <- dba_median_df[order(dba_median_df$Median_SCORE), ]


print(dba_median_df)
```

```
##    ZIPCODE Median_SCORE
## 14   10121           11
## 42   11371           11
## 37   10020           12
## 41   10281           12
## 45   10169           12
## 11   10028           13
## 12   10019           13
## 17   10017           13
## 27   10280           13
## 32   10112           13
## 50   10006           13
## 16   10036           14
## 31   10021           14
## 35   10007           14
## 51   10119           14
## 33   10022           15
## 38   10065           15
## 1                     16
## 21   10005           16
## 23   10001           16
## 34   10031           16
## 24   10018           17
## 30   10012           17
## 36   10038           17
## 2    10014           18
## 3    10003           18
## 6    10010           18
## 7    10016           18
## 18   10004           18
## 20   10011           18
## 22   10027           18
## 40   10023           18
```

```
## 49   10039          18
## 10   10002          19
## 25   10024          19
## 39   10034          19
## 44   10282          19
## 48   10026          19
## 5    10033          20
## 8    10044          20
## 26   10009          20
## 9    10032          21
## 15   10025          21
## 29   10013          21
## 4    10035          22
## 13   10029          22
## 47   10030          22
## 19   10128          23
## 28   10040          23
## 43   10075          23
## 46   10037          23
```

The main pattern seen between the inspections numbers and the median is that most of the zipcodes have at least more than 10 points in the score, that means that having more than 50 inspections with 10 point as median value means that most of these restaurants passed through 10 points and they complete actions to improve a achieve better score.

Please review the same homework in google colab (Python) following this link: **Google Colab notebook**

And the same file has been uploaded in my Github repository of the class please follow this link: **My GitHub Repository**