IMT 573: Module 6 Lab

Conditional Probability

Ali Qazi

Due: July 31, 2022

Collaborators: List collaborators here. Akeel Qazi, Anthony Mercado

Objectives

Conditional probability is a concept core to modeling data. In this lab exercise, we will work on framing questions in terms of conditional probabilities and computing probabilities to answer those questions. As you work through these questions you will be given opportunities to practice your data manipulation skills, as well as visualization skills. I encourage you to all explain the data anlaysis in written explanations.

Instructions

Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Cloud.

1. Open the `06_lab_condprob.Rmd` and save a copy to your local directory. Supply your solutions to the assignment by editing `06_lab_condprob.Rmd`.
2. First, replace the "YOUR NAME HERE" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and I encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit`. When the PDF report is generated rename the knitted PDF file to `lab6_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup

In this lab you will need, at minimum, the following R packages.

```
# Load standard libraries
library(tidyverse)
library(nycflights13)
```

If a baseball team scores X runs, what is the probability it will win the game?

This is the question we will explore in this lab (adapted from Decision Science News, 2014). We will use a dataset of baseball game statistics from 2010-2013.

Baseball is a played between two teams who take turns batting and fielding. A run is scored when a player advances around the bases and returns to home plate. More information about the dataset can be found at http://www.retrosheet.org/.

Data files can be found data folder on RStudio Cloud. Load them into one data.frame in R as shown below. Comment this code to demonstrate you understand how it works.

Note: More information about the dataset can be found at http://www.retrosheet.org/

```
# Data can be obtained from http://www.retrosheet.org/
# Data do not have column names on them. You can obtain
# column names from http://www.dangoldstein.com/flash/bball/cnames.txt

# Read in the column names
colNames <- read.csv("cnames.txt", header=TRUE)

# Create an empty object to store the data
baseballData <- NULL
for (year in seq(2010,2013,by=1)){ # Loop through years to get all data
  mypath <- paste('GL',year,'.TXT',sep='') # Create the path name for the file
  # cat(mypath,'\n') # Tell me what file I am working on
  # Read in the file and bind to data with correct column names
  baseballData <- rbind(baseballData,read.csv(mypath,
  col.names=colNames$Name))
  baseballData <- tbl_df(baseballData)
}
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
baseballData
```

```
## # A tibble: 9,716 x 161
##        Date Numberofgame Day   Visitor VisitorLeague VisitorGameNum Home
##       <int>        <int> <chr> <chr>   <chr>                  <int> <chr>
##  1 20100405            0 Mon   MIN     AL                         1 ANA
##  2 20100405            0 Mon   CLE     AL                         1 CHA
##  3 20100405            0 Mon   DET     AL                         1 KCA
##  4 20100405            0 Mon   SEA     AL                         1 OAK
##  5 20100405            0 Mon   TOR     AL                         1 TEX
##  6 20100405            0 Mon   SDN     NL                         1 ARI
##  7 20100405            0 Mon   CHN     NL                         1 ATL
##  8 20100405            0 Mon   SLN     NL                         1 CIN
##  9 20100405            0 Mon   SFN     NL                         1 HOU
```

```
## 10 20100405            0 Mon   COL     NL                          1 MIL
## # ... with 9,706 more rows, and 154 more variables: HomeLeague <chr>,
## #   HomeGameNum <int>, VisitorScore <int>, HomeScore <int>, Outs <int>,
## #   DayorNight <chr>, Completion <chr>, Forfeit <lgl>, Protest <chr>,
## #   ParkID <chr>, Attendance <int>, DurationMinutes <int>,
## #   VisitingLineScores <chr>, HomeLineScores <chr>, Vat.bats <int>,
## #   Vhits <int>, Vdoubles <int>, Vtriples <int>, Vhomeruns <int>, VRBI <int>,
## #   Vsacrificehits <int>, Vsacrificeflies <int>, Vhit.by.pitch <int>, ...
```

Select the following relevant columns and create a new local data.frame to store the data you will use for your analysis.

- Date
- Home
- Visitor
- HomeLeague
- VisitorLeague
- HomeScore
- VisitorScore

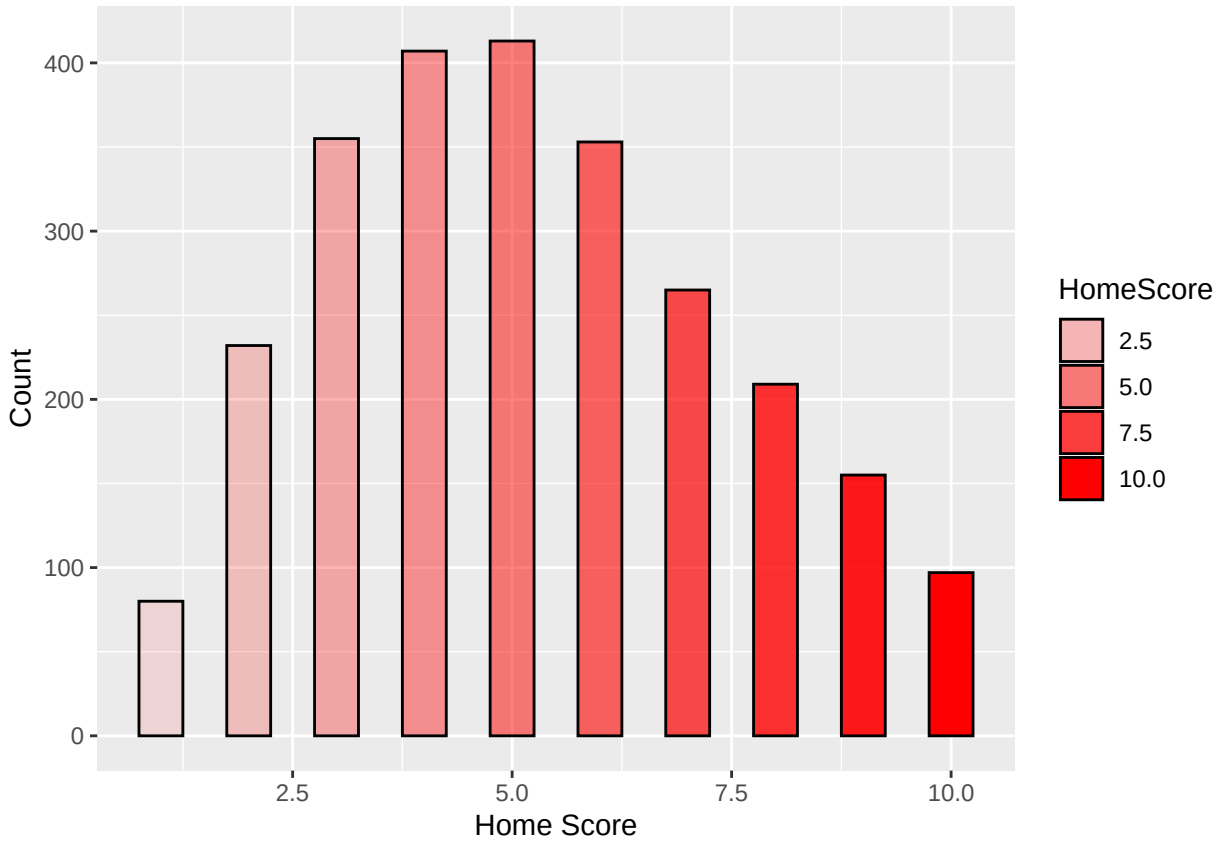
```
baseball_data <- select(baseballData, Date, Home, Visitor,VisitorLeague,
                        HomeLeague, HomeScore, VisitorScore)
```

Considering only games between two teams in the National League, compute the conditional probability of the team winning given X runs scored, for $X = 0, ..., 10$. Do this separately for Home and Visitor teams. 11

- Design a visualization that shows your results.
- Discuss what you find.

```
national_league <- filter(baseball_data, HomeLeague == "NL")
home_run <- filter(national_league, HomeScore > VisitorScore, HomeScore <= 10)
home_run <- group_by(home_run, HomeScore)
home_run <- summarise(home_run, count = n())
```

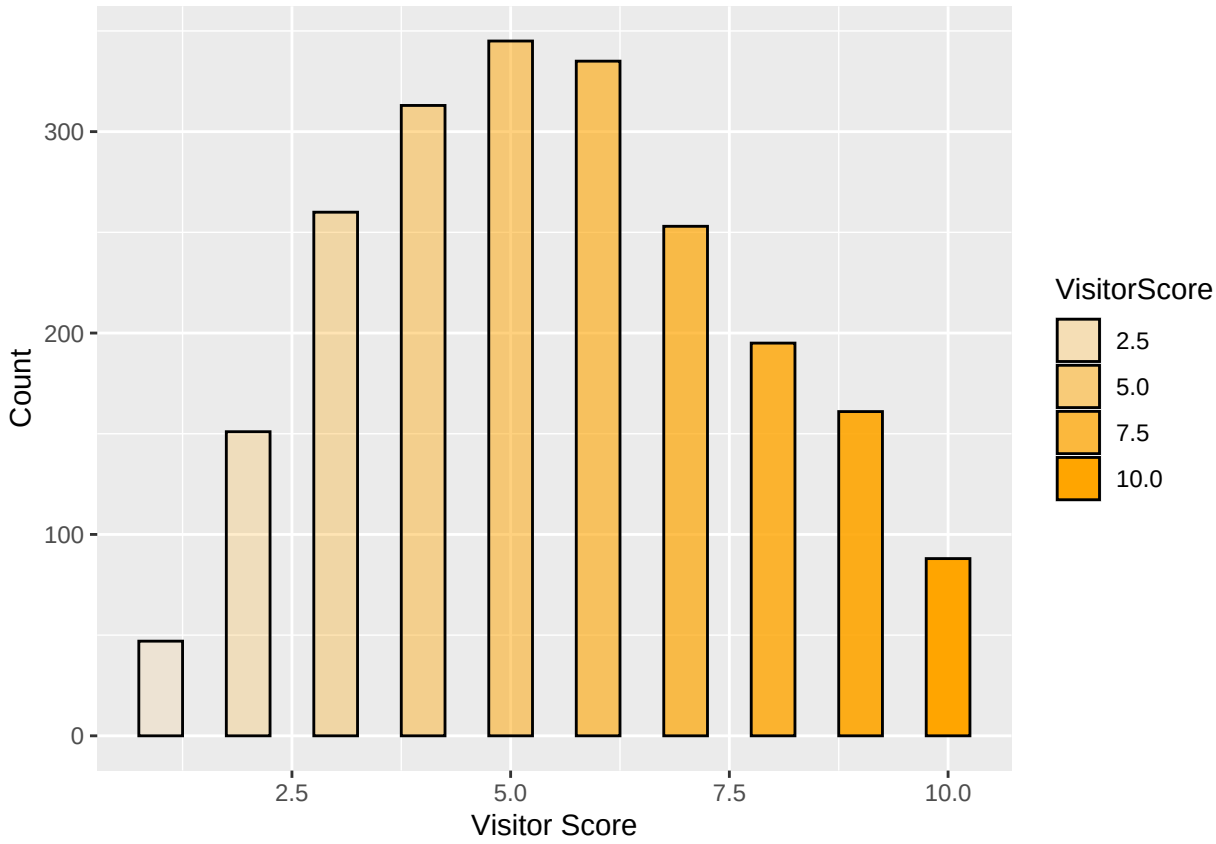
```
ggplot(data = home_run, aes( x = HomeScore, y = count)) +
  geom_bar(aes(alpha = HomeScore), colour = "black", fill = "red", width = 0.5,
  stat = "identity") + ylab("Count") + xlab("Home Score")
```



This graph counts the number of Home wins. Home team is given x runs scored.

```
national_league <- filter(baseball_data, VisitorLeague == "NL")
visitor_run <- filter(national_league, VisitorScore > HomeScore, VisitorScore <= 10)
visitor_run <- group_by(visitor_run, VisitorScore)
visitor_run <- summarise(visitor_run, count = n())
```

```
ggplot(data = visitor_run, aes( x = VisitorScore, y = count)) +
  geom_bar(aes(alpha = VisitorScore), colour = "black", fill = "orange", width = 0.5,
  stat = "identity") + ylab("Count") + xlab("Visitor Score")
```



This graph counts the number of Visitor wins. Visitor team is given x runs scored.

Conditional probability of an event occurring based on some other event. In the two visualizations we compute the conditional probability of the home or visitor team winning given X runs scored, for $X = 0, ..., 10$. From the two visualization we can see that they look very similar. The mean for the distribution is both 5 with a count around 400.

Extra Credit: Repeat the above problem, but now consider the probability of winning given the number of hits.

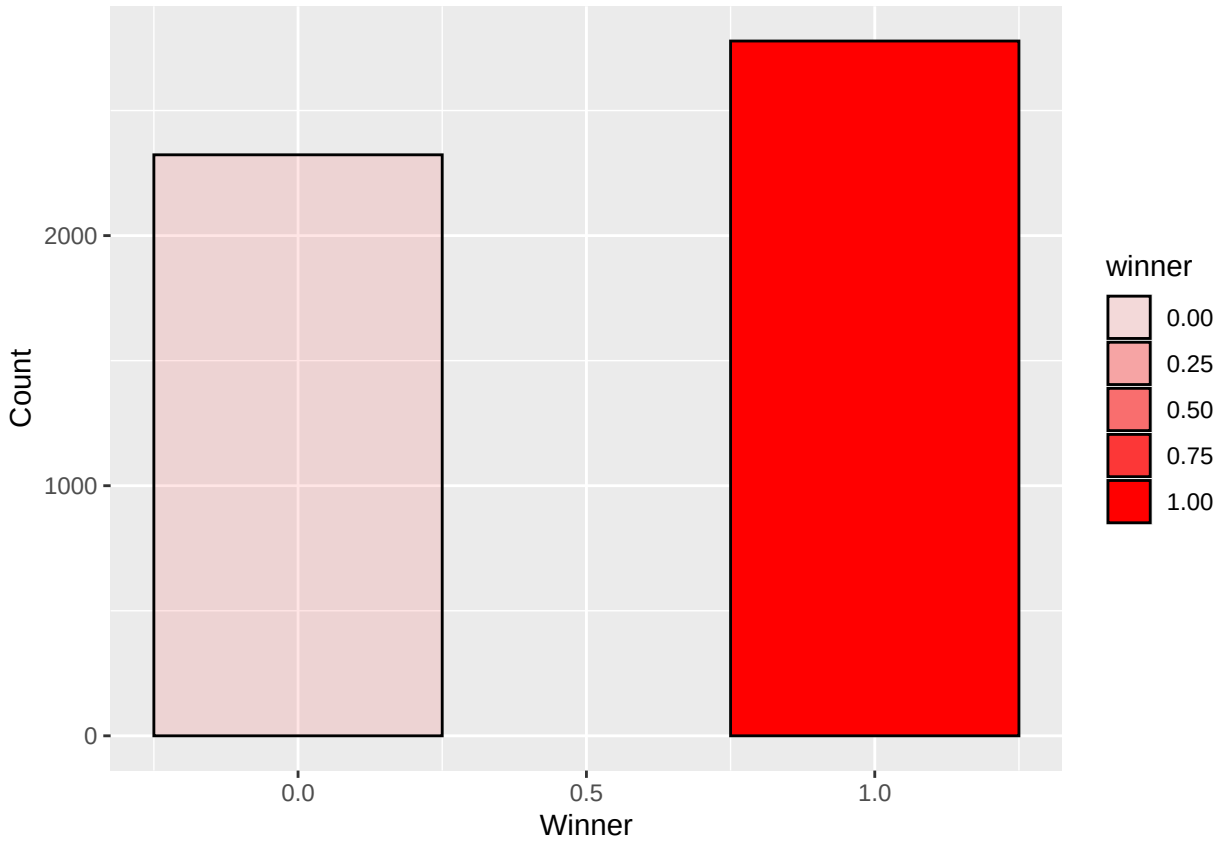
My attempt-

```
#baseball_winner <- mutate(baseball_data, winner = (ifelse)
                          #(HomeScore > VisitorScore, "Home", "Visitor"))

national_league <- filter(baseball_data, VisitorLeague == "NL")
baseball_winnercount <- national_league
baseball_winnercount <- mutate(baseball_winnercount, winner = (ifelse)
                           (HomeScore > VisitorScore, 1, 0))
baseball_winnercount <- group_by(baseball_winnercount, winner)

baseball_winnercount <- summarise(baseball_winnercount, count = n())

ggplot(data = baseball_winnercount, aes(x = winner, y = count)) +
  geom_bar(aes(alpha = winner), colour = "black", fill = "red", width = 0.5,
  stat = "identity") + ylab("Count") + xlab("Winner")
```


Count
2000
1000
0
0.0
0.5
1.0
Winner
winner
0.00
0.25
0.50
0.75
1.00