Sports Analytics

Aaron Nielsen, Department of Statistics, Colorado State University

2022-11-07

Contents

About

This book serves as the course textbook for the following courses at Colorado State University:

- STAT 381 (Sports Statistics and Analytics I)
- STAT 382 (Sports Statistics and Analytics II)

CSU students contributed to the creation of this book. Many thanks to the following student collaborators:

- Levi Kipp
- Ellie Martinez
- Isaac Moorman

Preliminaries

Getting Started With R

Installing R

For this class, you will be using R Studio to complete statistical analyses on your computer.

To begin using R Studio, you will need to install “R” first and then install “R Studio” on your computer.

Step 1: Download R
(a) Visit https://www.r-project.org/
(b) Click **CRAN** under **Download** (c) Select any of the mirrors
(d) Click the appropriate link for your type of system (Mac, Windows, Linux)
(e) Download R on this next page.
(For Windows, this will say **install R for the first time**. For Mac, this will be under **Latest release** and will be something like **R-4.1.0.pkg** – the numbers may differ depending on the most recent version)
(f) Install R on your computer

Step 2: Download R Studio
(a) Visit https://www.rstudio.com/products/rstudio/download/#download
(b) Click to download
(c) Install R Studio on your computer

Step 3: Verify R Studio is working
(a) Open R Studio
(b) Let's enter a small dataset and calculate the average to make sure everything is working correctly.
(c) In the console, type in the following dataset of Sammy Sosa's season home run totals from 1998–2002:

```
sosa.HR <- c(66,63,50,64,49)
```

(d) In the console, calculate the average season home run total for Sammy Sosa between 1998–2002:

```
mean(sosa.HR)
```

```
## [1] 58.4
```

(e) Did you find Slammin' Sammy's average home run total from 1998–2002 was 58.4? If so, you should be set up correctly!

Some R Basics

For the following examples, let's consider Peyton Manning's career with the Denver Broncos. In his four seasons with the Broncos, Manning's passing yard totals were: 4659, 5477, 4727, 2249. Let's enter this data into R. To enter a vector of data, use the **c()** function.

```
peyton <- c(4659, 5477, 4727, 2249)
```

To look at the data you just put in the variable *peyton*, type *peyton* into the console and press enter.

```
peyton
```

```
## [1] 4659 5477 4727 2249
```

Some basic function for calculating summary statistics include **summary**, **mean()**, **median()**, **var()**, and **sd()**.

```
summary(peyton)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2249    4056    4693    4278    4914    5477
```

```
mean(peyton)
```

```
## [1] 4278
```

```
sd(peyton)
```

```
## [1] 1402.522
```

R allows you to install additional packages (collections of functions) that aren't offered in the base version of R. To install a package, use **install.packages()** and to load a package, use **library()**.

One package that we will use frequently is **tidyverse**. This package includes several other packages and functions such as **ggplot** (plotting function), **dplyr** (data manipulation package), and **stringr** (string manipulation package).

```
install.packages("tidyverse")
library("tidyverse")
```

You will also need to know how to load datasets from files. For this class, we will typically provide data files is .csv format.

Here is how to load a file:

```
# load readr package and load example dataset
NFL_2021_Team_Passing <- read_csv("data/NFL_2021_Team_Passing.csv")

# we can look at the header (first few entries) using "head()"
head(NFL_2021_Team_Passing)
```

```
## # A tibble: 6 x 25
##      Rk Tm             G   Cmp   Att `Cmp%`   Yds    TD `TD%`   Int `Int%`   Lng
##   <dbl> <chr>      <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1     1 Tampa Bay~    17   492   731   67.3  5229    43   5.9    12    1.6    62
## 2     2 Los Angel~    17   443   674   65.7  4800    38   5.6    15    2.2    72
## 3     3 Dallas Co~    17   444   647   68.6  4800    40   6.2    11    1.7    73
## 4     4 Kansas Ci~    17   448   675   66.4  4791    37   5.5    13    1.9    75
## 5     5 Los Angel~    17   406   607   66.9  4642    41   6.8    18    3      79
## 6     6 Las Vegas~    17   429   628   68.3  4567    23   3.7    14    2.2    61
## # ... with 13 more variables: `Y/A` <dbl>, `AY/A` <dbl>, `Y/C` <dbl>,
## #   `Y/G` <dbl>, Rate <dbl>, Sk <dbl>, SKYds <dbl>, `Sk%` <dbl>, `NY/A` <dbl>,
## #   `ANY/A` <dbl>, `4QC` <dbl>, GWD <dbl>, EXP <dbl>
```

Which Sports?

Which sports will we be analyzing in this class?

Challenges

What kind of challenges could we encounter dealing with sports data?

Ethical Concerns

What kinds of ethical concerns should we consider when dealing with sports data?

Chapter 1

Exploratory Data Analysis

1.1 Descriptive Statistics

1.1.1 Definitions

Definition 1.1. A ***population*** is a well-defined complete collection of objects.

Definition 1.2. A ***sample*** is a subset of the population.

Example 1.1. Suppose we are interested in studying Peyton's Manning's season passing yards totals. How could you define the population and what is one possible sample?

Definition 1.3. ***Quantitative data*** is numeric data or numbers. It can be broken into two further categories: discrete and continuous data.

Definition 1.4. ***Discrete data*** is quantitative data with a finite or countably infinite number of values.

Definition 1.5. ***Continuous data*** is quantitative data with an uncountably infinite number of values or data taken from an interval.

Example 1.2. What are possible discrete and continuous data associated with Peyton Manning?

Definition 1.6. ***Qualitative data*** refers to names, categories, or descriptions. It can also be broken down into two further categories, nominal data and ordinal data.

Definition 1.7. ***Nominal data*** is qualitative data with no natural ordering.

Definition 1.8. ***Ordinal data*** is qualitative data with a natural ordering.

Example 1.3. What are possible nominal and ordinal data associated with Peyton Manning?

1.1.2 Descriptive Statistics

While we will learn about some descriptive statistics that are unique to specific sports, there are some descriptive statistics that are frequently used in many applications.

1.1.2.1 Quantitative Data

There are different descriptive statistics depending on the type of data you are analyzing. We will begin by looking at descriptive statistics for quantitative data.

To begin, let $x_1, x_2, \ldots, x_n$ represent a numerical dataset with a sample of size n, where x_i is the i^{th} value in the dataset.

Definition 1.9. The ***sum*** of the data values is given by: $\sum_{i=1}^{n} x_i = x_1 + x_2 + \ldots + x_n$

Definition 1.10. The ***sample mean*** (or sample average), $\bar{x}$, of the numerical dataset is given by $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Definition 1.11. The ***population mean*** (or population average), μ, is the mean value for the entire population.

The mean can be thought of as a measure of center or more generally, a measure of location.

Example 1.4. Recall that Peyton Manning's season passing yards total while with the Broncos were: 4659, 5477, 4727, 2249. Calculate the sample mean of these values.

```r
# Calculate the sample of Peyton Manning's passing yards season totals with Colts
peyton.broncos <- c(4659, 5477, 4727, 2249)
mean(peyton.broncos)
```

```
## [1] 4278
```

In sports statistics, we often have to choose between using a descriptive statistic that summarizes a quantity versus a descriptive statistic that summarizes a rate. For instance, in basketball, we can compare two players based on how many points they score in a game (total quantity) or we can compare two players based on how many points per minute played (rate statistic). Many applications in sports analytics focus more on rate statistics rather than quantity statistics. Why?

We can measure the spread or variability of a dataset using *variance* and *standard devatiation*.

Definition 1.12. The ***sample variance***, s^2, of the numerical dataset is a measure of spread and is given by $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

Definition 1.13. The ***sample standard deviation***, s, of the numerical dataset is a measure of spread and is given by $s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Definition 1.14. The ***population variance***, σ^2, is the variance for an entire population and is given by $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$, where N is the population size.

Definition 1.15. The ***population standard deviation***, σ, is the standard deviation for an entire population and is given by $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

We often prefer to work with standard deviations as a measure of spread as opposed to variance because standard deviations are given in our original units.

```
# Calculate the variance and standard deviation of Peyton Manning's passing
# yards season totals with Broncos
var(peyton.broncos) # units: yards^2
```

```
## [1] 1967068
```

```
sd(peyton.broncos) # units: yards
```

```
## [1] 1402.522
```

Definition 1.16. The **sample median**, $\tilde{x}$, of a numerical dataset is the middle value when the data are ordered from smallest to largest. In other words, let $x_1, x_2, \ldots, x_n$ be the (unordered) dataset and let $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$ be the same dataset but ordered from smallest to largest. If n is odd, then $\tilde{x} = x_{(n+1)/2}$ and if n is even, then $\tilde{x} = \frac{1}{2} \cdot \left[x_{(\frac{n}{2})} + x_{(\frac{n+1}{2})}\right]$.

Example 1.5. Calculate the sample median of Peyton Manning's season passing yards total while with the Colts (3739, 4135, 4413, 4131, 4200, 4267, 4557, 3747, 4397, 4040, 4002, 4500, 4700).

Like sample mean, sample median is a measure of center. It gives you an idea of where the "middle" of your dataset is.

We can calculate sample mean and sample median in R as follows:

```
# Calculate the median of Peyton Manning's passing yards season totals with
# Broncos and Colts
peyton.colts <- c(3739, 4135, 4413, 4131, 4200, 4267, 4557, 3747, 4397, 4040,
                  4002, 4500, 4700)
median(peyton.broncos)
```

```
## [1] 4693
```

```
median(peyton.colts)
```

```
## [1] 4200
```

Definition 1.17. A ***percentile*** is a measure of relative standing. The p^{th} percentile is the number where at least p% of the data values are less than or equal to this number.

Definition 1.18. A ***quantile*** is a measure of relative standing and are the cut points for breaking a distribution of values into equal sized bins.

Definition 1.19. A ***quartile*** is a measure of relative standing and are the cut points for breaking a distribution of values into four equal parts.

```
# Calculate the 10th and 90th percentile of Peyton Manning's passing yards
# season totals with Colts
quantile(peyton.colts,0.10)
```

```
##  10%
## 3798
```

```
quantile(peyton.colts,0.90)
```

```
##    90%
## 4545.6
```

```
quantile(peyton.colts,c(0.1,0.9))
```

```
##    10%    90%
## 3798.0 4545.6
```

Special percentiles:
1. 25th percentile = 1st quartile = Q_1
2. 50th percentile = 2nd quartile = $Q_2 = \tilde{x}$
3. 75th percentile = 3rd quartile = Q_3

Definition 1.20. ***Range*** is a measure of spread, measures the full width of a dataset, and is given by: $Range = Max - Min$.

Definition 1.21. ***Interquartile range*** is a measure of spread, measures the width of the middle 50% of a dataset, and is given by: $IQR = Q_3 - Q_1$.

Definition 1.22. A ***five number summary*** describes the center, spread, and edges of a dataset and is given by: $(Min, Q_1, Q_2, Q_3, max)$.

```
summary(peyton.colts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3739    4040    4200    4218    4413    4700
```

```
quantile(peyton.colts,c(0,0.25,0.5,0.75,1))
```

```
##   0%  25%  50%  75% 100%
## 3739 4040 4200 4413 4700
```

1.1.2.2 Qualitative Data

In sports statistics, we also encounter qualitative (categorical) data which is names or labels which has its own descriptive statistics.

To begin, let $x_1, x_2, \dots, x_n$ represent a categorical dataset with a sample of size n, where x_i is the i^{th} value in the dataset.

Definition 1.23. The ***proportion*** of sampled data that fall into a category is given by: $p = \frac{\text{\# in category}}{\text{\# total}}$

’ ’Proportion” and “Probability” are often used interchangeably. Both have a minimum value of 0 and a maximum value of 1.

Definition 1.24. The ***percentage*** of sampled data that fall into a category is given by: $P\% = 100 \cdot p = 100 \cdot \frac{\text{\# in category}}{\text{\# total}}$

Percentages in this context can have a minimum value of 0% and a maximum value of 100%.

Example 1.6. In 2014, Peyton Manning started as quarterback for the Denver Broncos. The result of the Broncos’ 16-game season was:
Win, Win, Loss, Win, Win, Win, Win, Loss, Win, Loss, Win, Win, Win, Win, Loss, Win

Calculate the proportion and percentage of Broncos’ winning games in 2014.

```r
broncos2014 <-
  c("Win", "Win", "Loss", "Win", "Win", "Win", "Win", "Loss",
    "Win", "Loss", "Win", "Win", "Win", "Win", "Loss", "Win")
broncos.prop <- sum(broncos2014 == "Win")/length(broncos2014); broncos.prop
```

```
## [1] 0.75
```

```r
broncos.perc <- 100*broncos.prop; broncos.perc
```

```
## [1] 75
```

We can also build a frequency table that summarizes the categories and their occurrences using **table()** in R. Note that **table()** works for quantitative and qualitative data.

```r
table(broncos2014)
```

```
## broncos2014
## Loss  Win
##    4   12
```

1.2 Visualizations

Conveying information visually is also an important part in providing a description of a dataset.

R provides some basic plotting functions such as **plot**, **hist**, and **barplot**. These plotting functions are simple and not always very clean looking.

In this class, we will use analogous plotting functions in **ggplot2** that are much improved plotting functions.

If you have already installed the **tidyverse** package, it should have also installed the **ggplot2** package.

```
# Load the tidyverse package (which includes ggplot2)
library(tidyverse)
```

Let's load the file "NFL_2021_Team_Passing.csv" which contains NFL Team Passing Statistics, 2021 from https://www.pro-football-reference.com/years/2021/index.htm#passing

```
library(readr)
NFL_2021_Team_Passing <- read_csv("data/NFL_2021_Team_Passing.csv")
head(NFL_2021_Team_Passing)
```

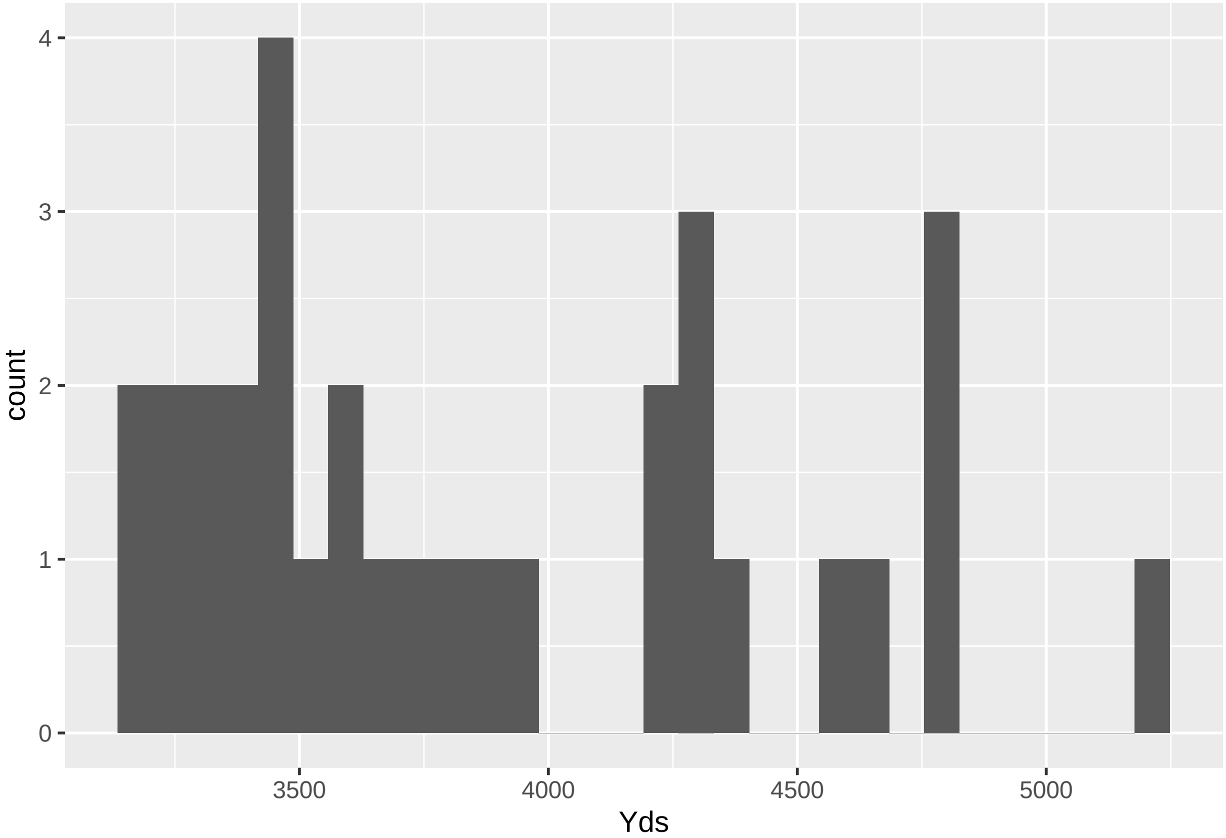
```
## # A tibble: 6 x 25
##      Rk Tm             G   Cmp   Att `Cmp%`   Yds    TD `TD%`   Int `Int%`   Lng
##   <dbl> <chr>      <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1     1 Tampa Bay~    17   492   731   67.3  5229    43   5.9    12    1.6    62
## 2     2 Los Angel~    17   443   674   65.7  4800    38   5.6    15    2.2    72
## 3     3 Dallas Co~    17   444   647   68.6  4800    40   6.2    11    1.7    73
## 4     4 Kansas Ci~    17   448   675   66.4  4791    37   5.5    13    1.9    75
## 5     5 Los Angel~    17   406   607   66.9  4642    41   6.8    18    3      79
## 6     6 Las Vegas~    17   429   628   68.3  4567    23   3.7    14    2.2    61
## # ... with 13 more variables: `Y/A` <dbl>, `AY/A` <dbl>, `Y/C` <dbl>,
## #   `Y/G` <dbl>, Rate <dbl>, Sk <dbl>, SKYds <dbl>, `Sk%` <dbl>, `NY/A` <dbl>,
## #   `ANY/A` <dbl>, `4QC` <dbl>, GWD <dbl>, EXP <dbl>
```

1.2.1 Histograms

Histograms are one of the most common and basic ways to visualize a dataset's distribution of values. To make a histogram, you will use **ggplot** and **geom_histogram**.

Example 1.7. Create a histogram of the NFL Team Passing Yards in 2021.

```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Yds)) +
  geom_histogram()
```

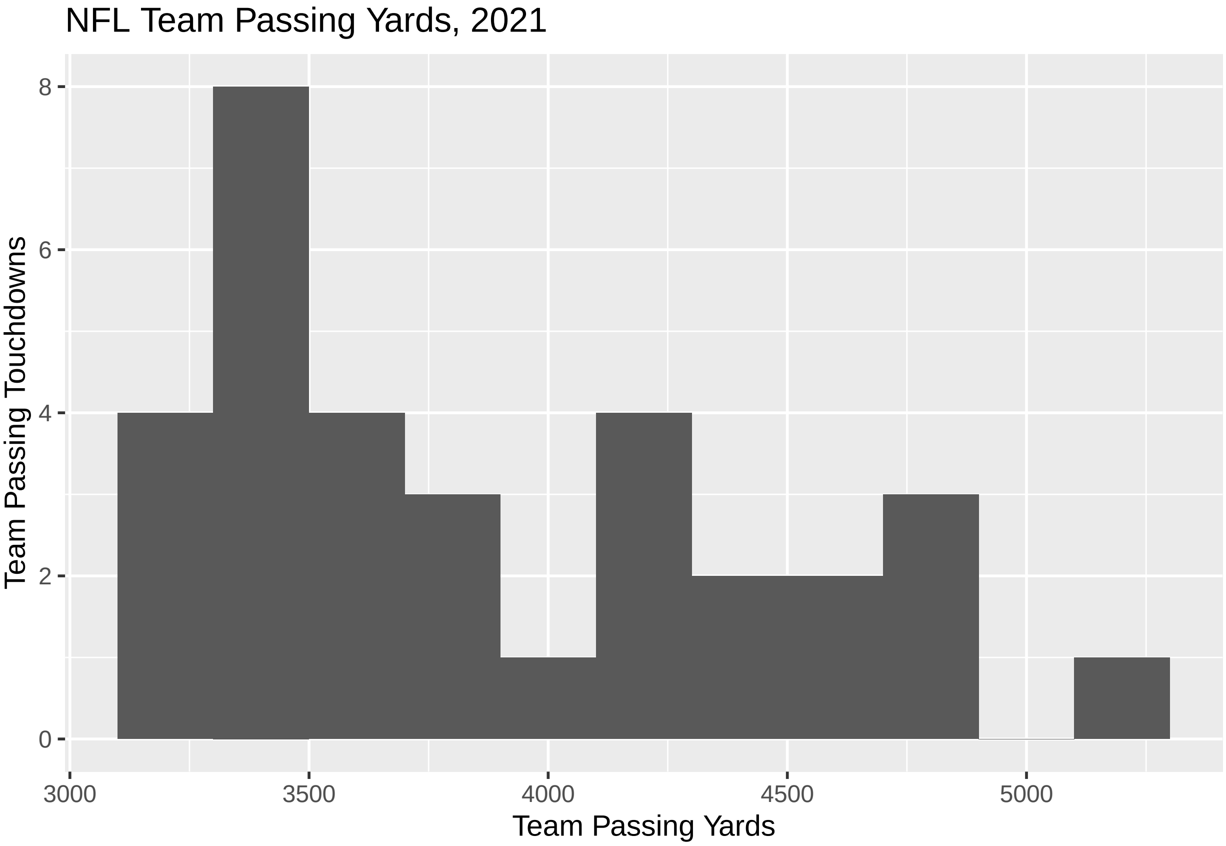


Notice how %>% is used to **pipe** the dataset into ggplot. This is using the pipe function from the **dplyr** package.

By default, **geom_histogram** uses 30 bins but this is customizable. Let's make the bins have a width of 200.

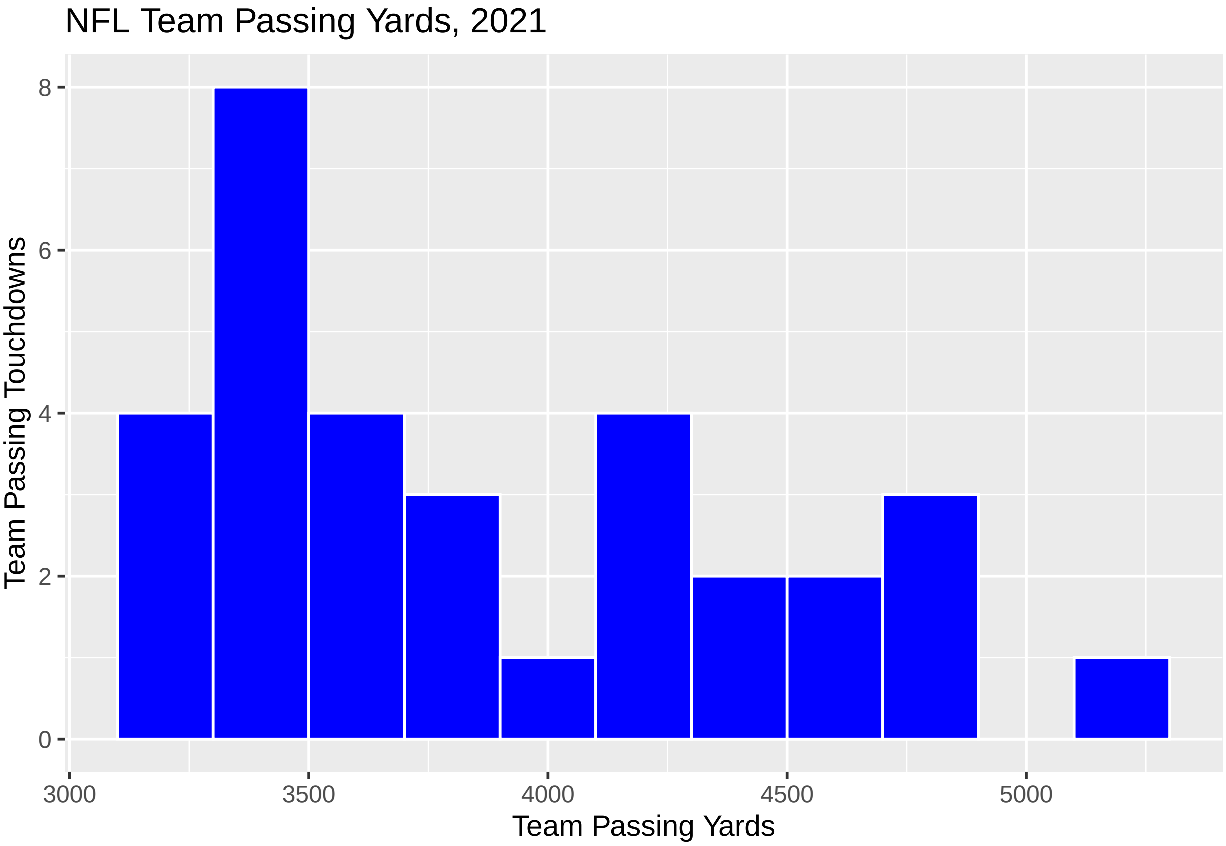
All good visualizations have good labels. Let's improve the axis labels and give the figure a title.

```
NFL_2021_Team_Passing %>% ggplot(aes(x=Yds)) +
  geom_histogram(binwidth = 200) +
  labs(x="Team Passing Yards",
       y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021")
```



We also have numerous options to change the appearance of plots when using **ggplot**. Let's change the bins color to *blue* and change the bin borders to *white*.

```
NFL_2021_Team_Passing %>% ggplot(aes(x=Yds)) +
  geom_histogram(color = "white", fill = "blue", binwidth = 200) +
  labs(x="Team Passing Yards",
       y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021")
```

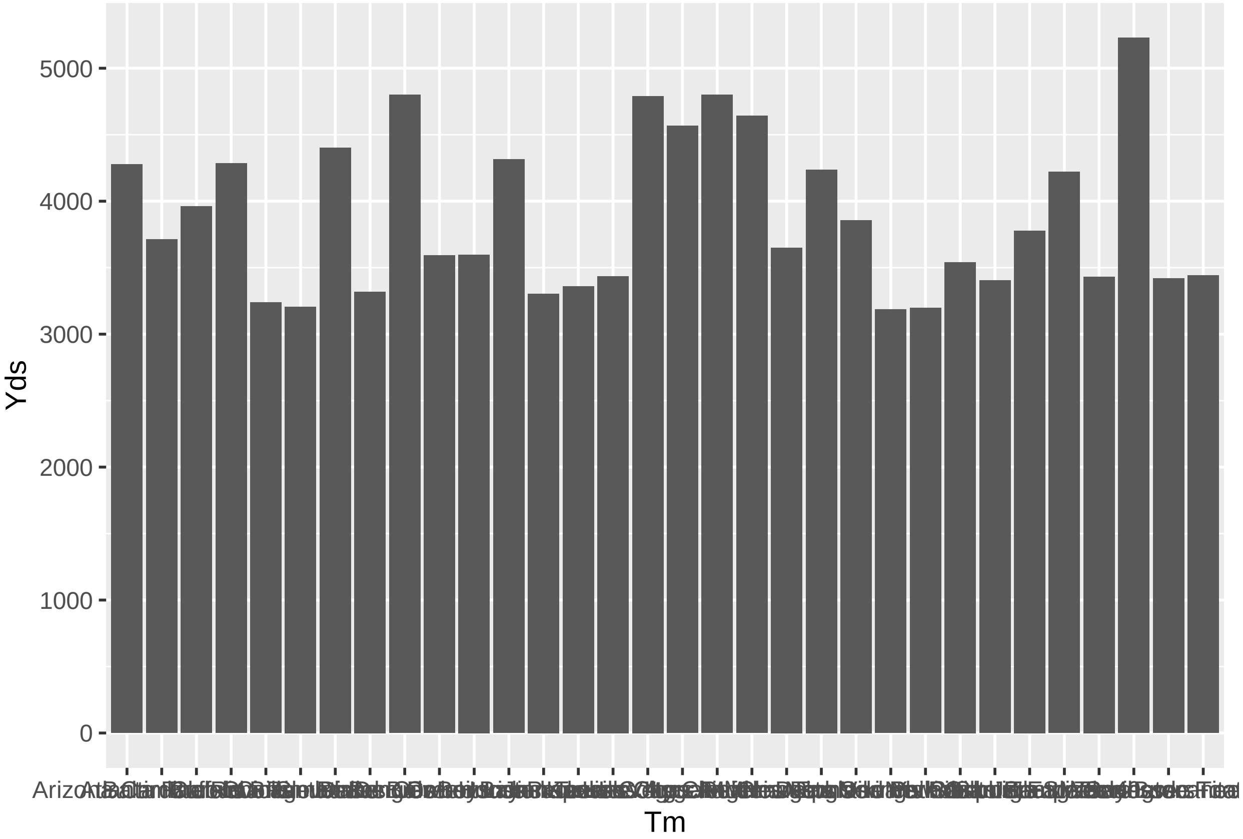


1.2.2 Bar Plots

We can also create bar plots using ggplot using the **geom__bar** function.

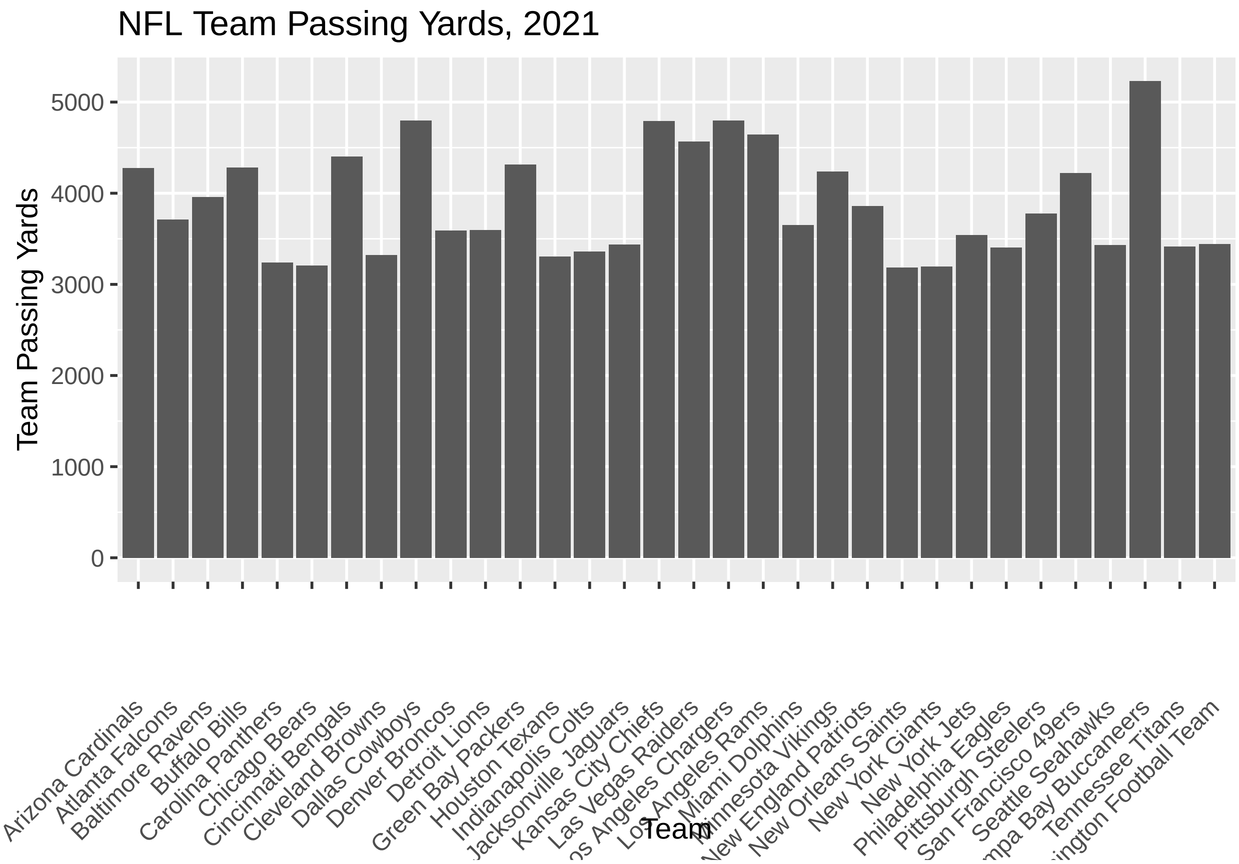
Example 1.8. Create a bar plot with teams on the horizontal axis and passing touchdowns on the vertical axis.

```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Tm,y=Yds)) +
  geom_bar(stat="identity")
```



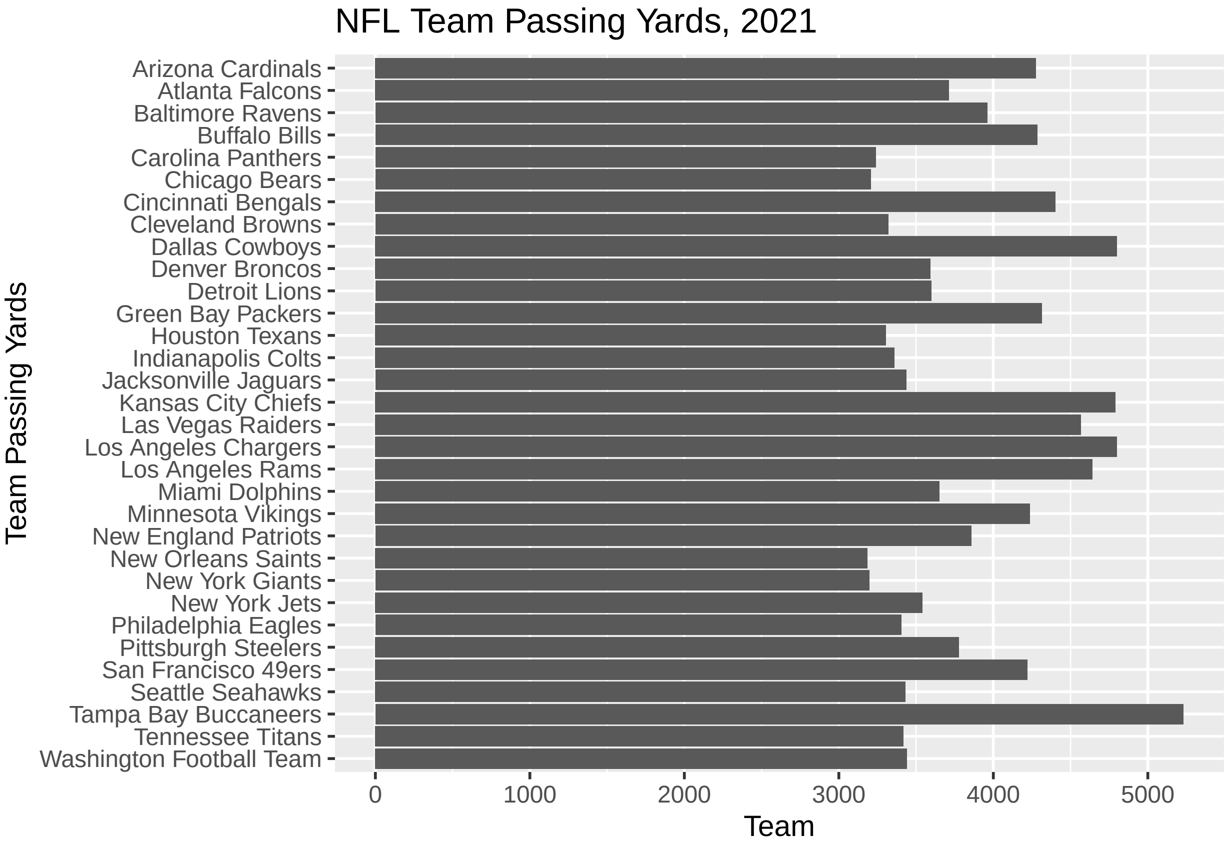
The team labels are a complete mess. Let's fix this and make some adjustments to the axis labels and figure title.

```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Tm,y=Yds)) +
  geom_bar(stat="identity") +
  labs(x="Team", y="Team Passing Yards",
       title="NFL Team Passing Yards, 2021") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



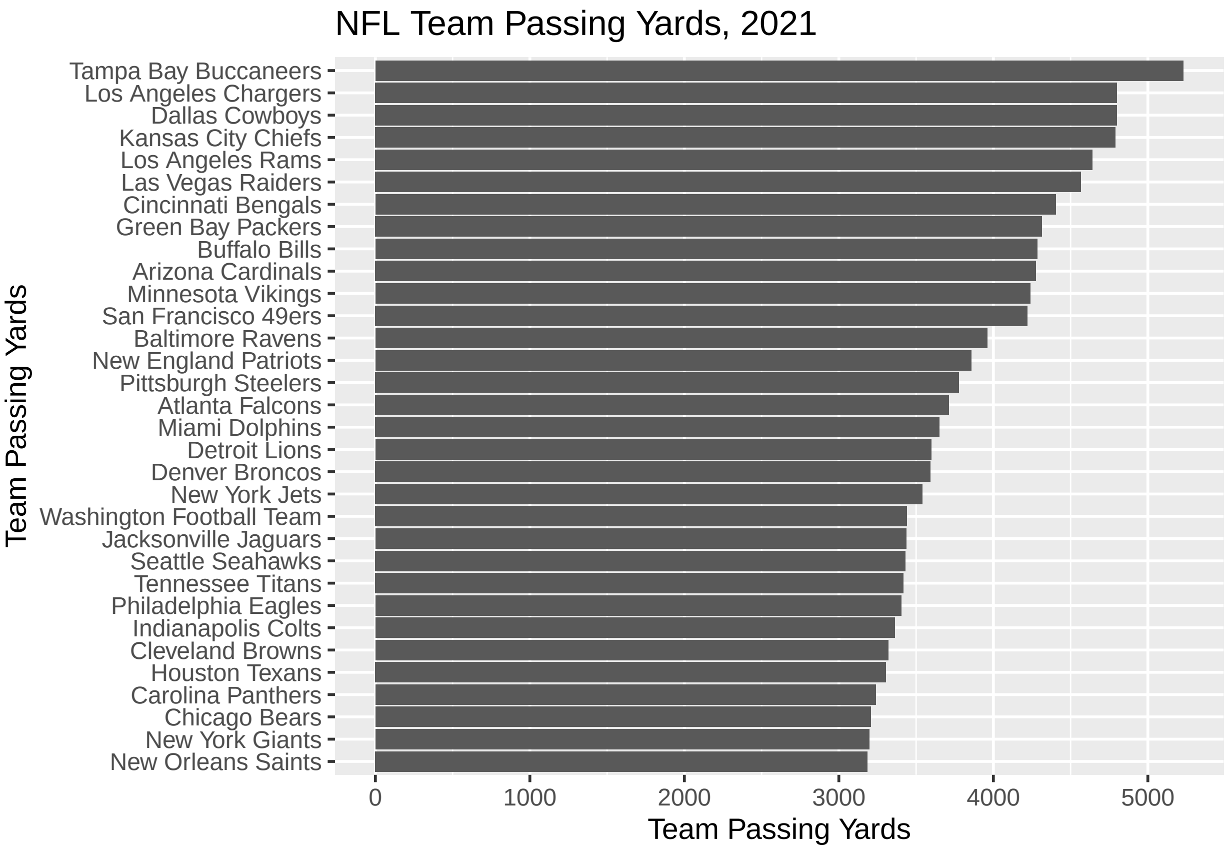
We can flip this graph if we like as well. Note that when we flip the graph, our labels get in reverse ordering, so this can be fixed using **fct_rev()** which is part of the **forcats** package.

```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=fct_rev(Tm),y=Yds)) +
  geom_bar(stat="identity") +
  labs(x="Team Passing Yards",y="Team",title="NFL Team Passing Yards, 2021") +
  coord_flip()
```



We can also order the teams from most team passing touchdowns to least using the `forcats` package.

```
NFL_2021_Team_Passing %>%
  mutate(Tm = fct_reorder(Tm,Yds)) %>%
  ggplot(aes(x=Tm,y=Yds)) +
  geom_bar(stat="identity") +
  labs(x="Team Passing Yards",
       y="Team Passing Yards",
       title="NFL Team Passing Yards, 2021") +
  coord_flip()
```

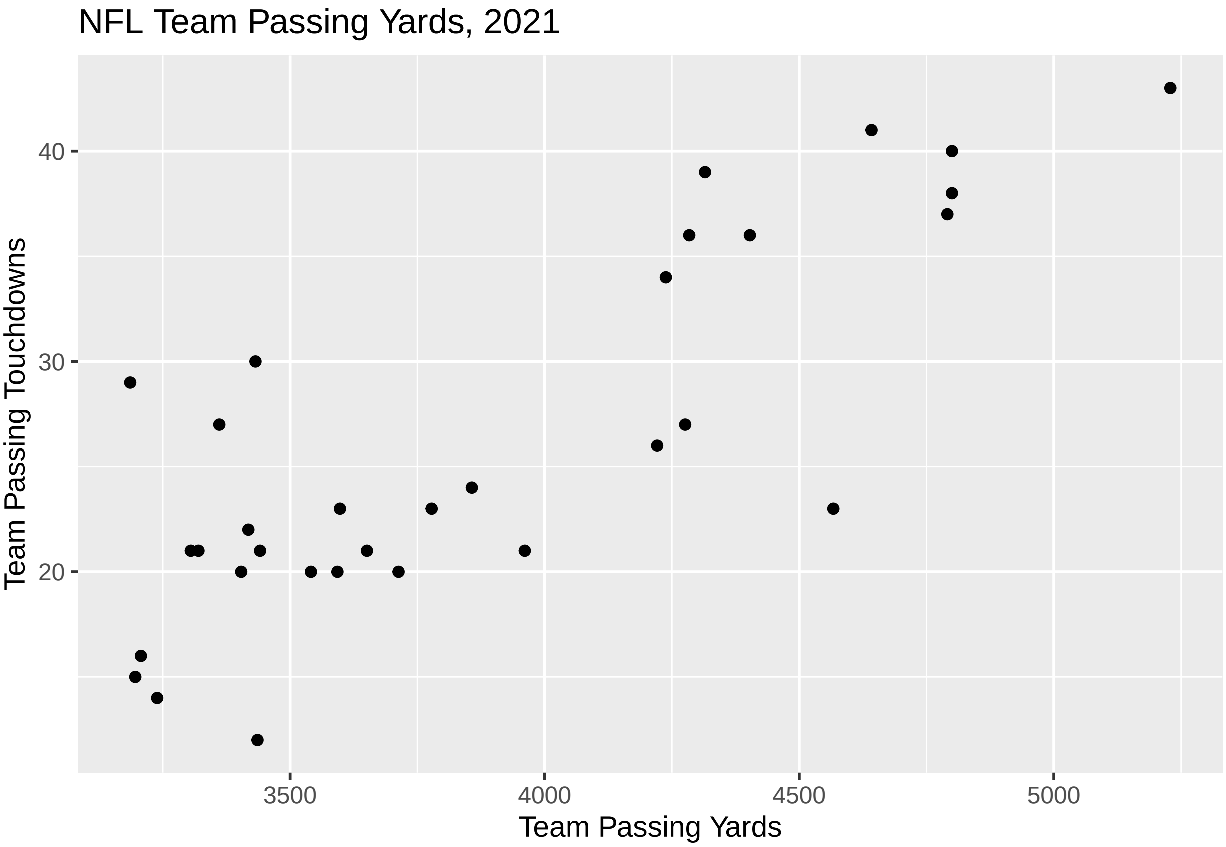


1.2.3 Scatter Plots

Another common and useful visualization is a scatterplot which shows the relationship between two numeric variable. In ggplot, you use **geom_point()**.

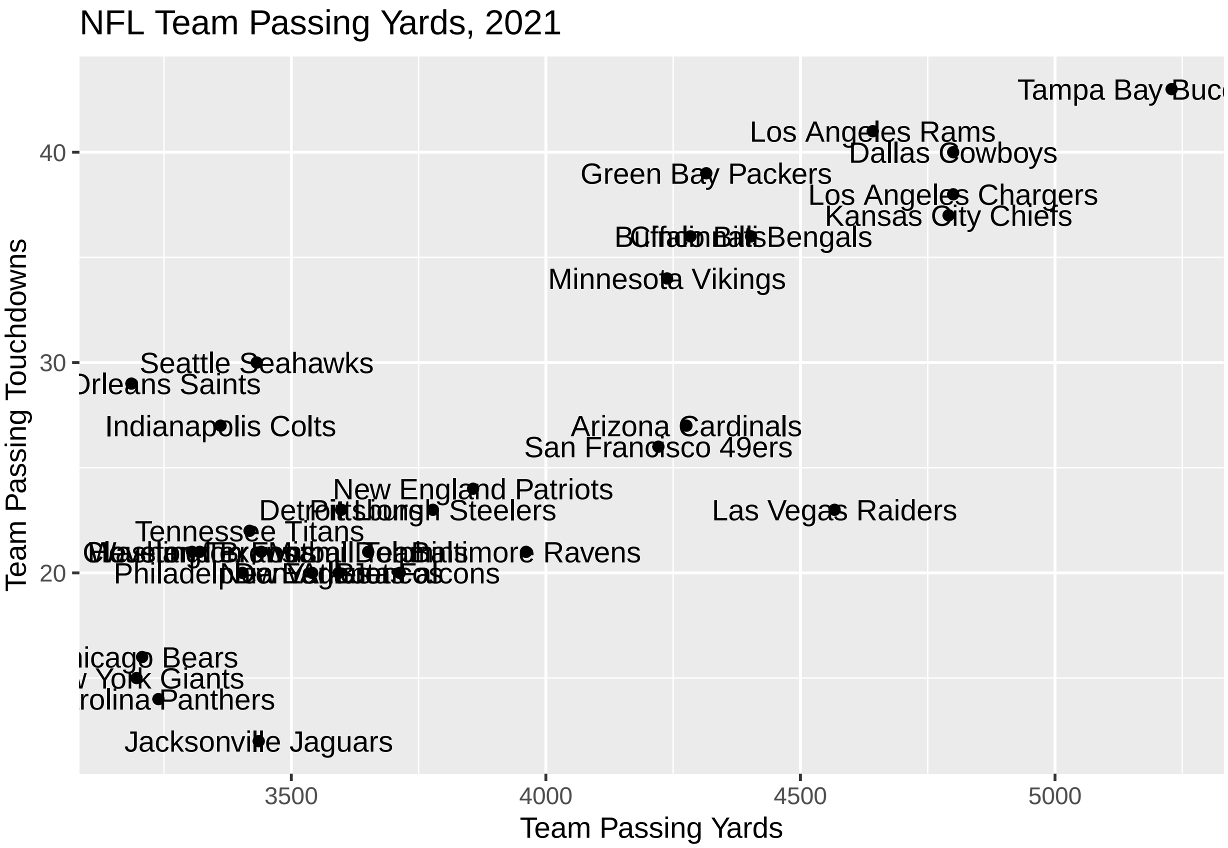
Example 1.9. Create a scatterplot of Team Passing Yards and Team Passing Touchdowns from the NFL 2021 dataset.

```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Yds,y=TD,label=Tm)) +
  geom_point() +
  labs(x="Team Passing Yards",
       y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021")
```



We may want to include team labels on this plot, however, it can get messy very quickly with a lot of points.

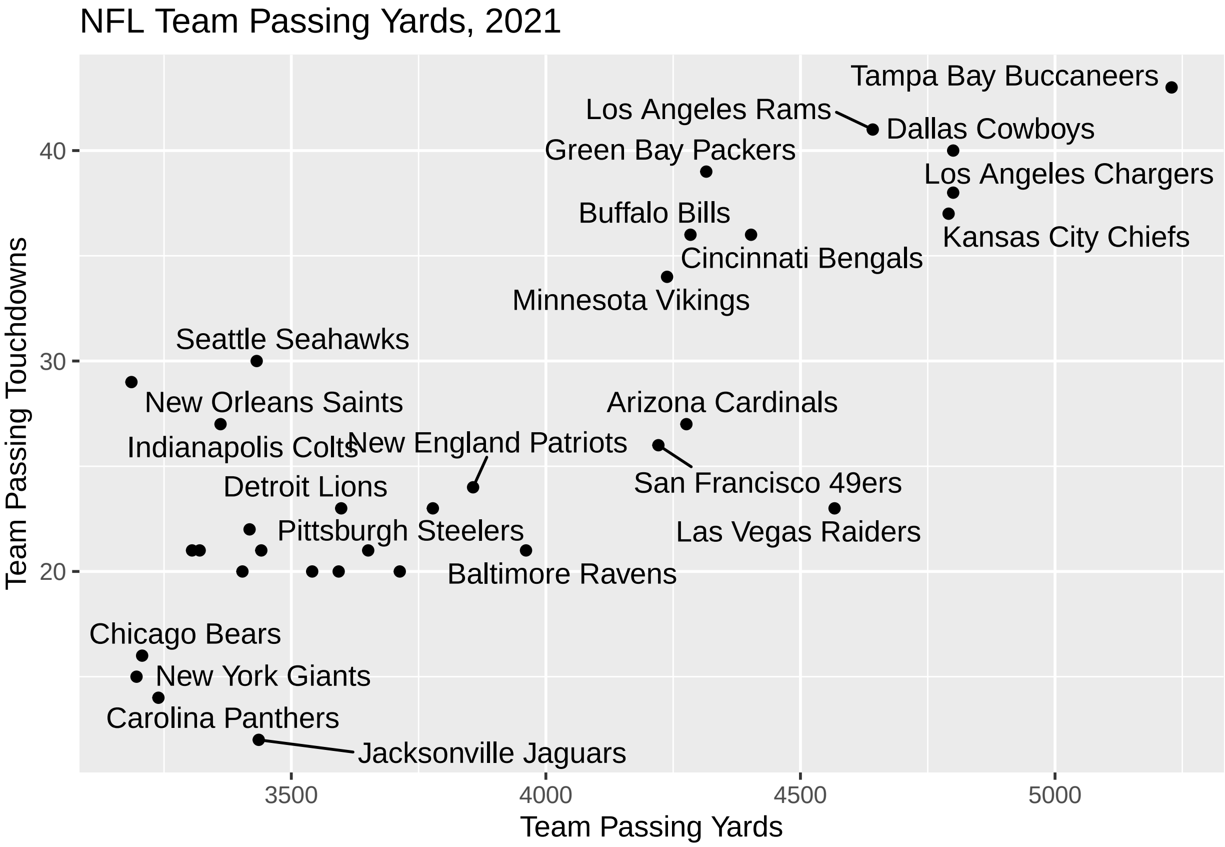
```
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Yds,y=TD,label=Tm)) +
  geom_point() +
  labs(x="Team Passing Yards",
       y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021") +
  geom_text()
```



Many sports leagues have around 30 teams, so a clean scatterplot with labels can be tricky to make. Here are some options below.

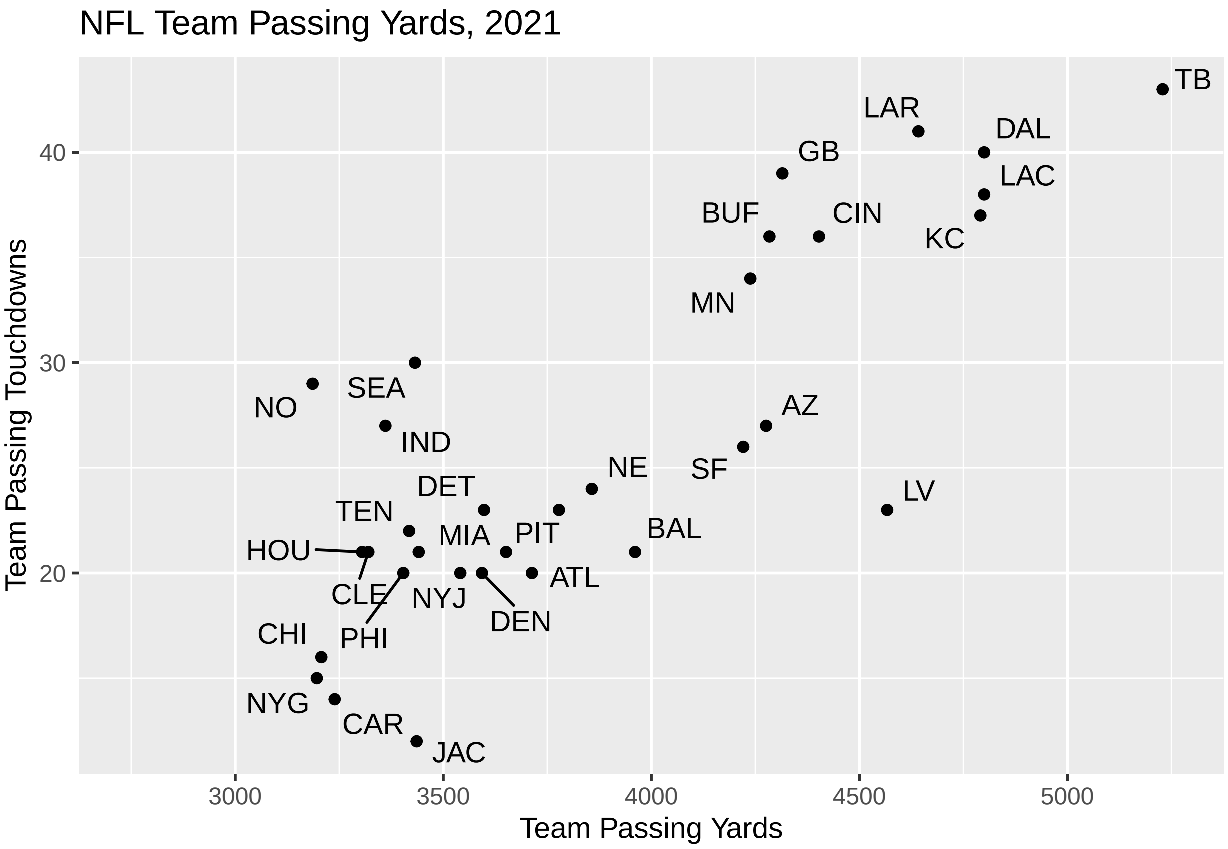
```
# install ggrepel package
library(ggrepel)
NFL_2021_Team_Passing %>%
  ggplot(aes(x=Yds,y=TD,label=Tm)) +
  geom_point() +
  labs(x="Team Passing Yards",
       y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021") +
  geom_text_repel()
```

```
## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
NFL_2021_Team_Passing$Abbr <-
  c("TB","LAC","DAL","KC","LAR","LV","CIN","GB","BUF","AZ","MN","SF",
    "BAL","NE","PIT","ATL","MIA","DET","DEN","NYJ","WAS","JAC","SEA",
    "TEN","PHI","IND","CLE","HOU","CAR","CHI","NYG","NO")

NFL_2021_Team_Passing %>%
  ggplot(aes(x=Yds,y=TD,label=Abbr)) +
  geom_point() +
  scale_x_continuous(limits=c(2750,5250)) +
  labs(x="Team Passing Yards",y="Team Passing Touchdowns",
       title="NFL Team Passing Yards, 2021") +
  geom_text_repel(box.padding = 0.3)
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



1.2.4 Kable Tables

We can build nice tables using `kableR` and `kableExtra`. Let's look at a few options.

```r
# Use a smaller dataset as an example
NFL21 <- NFL_2021_Team_Passing %>% select(2:8) %>% slice_head(n = 5)

# Default output for tabular data
NFL21
```

```
## # A tibble: 5 x 7
##   Tm                         G   Cmp   Att `Cmp%`   Yds    TD
##   <chr>                  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 Tampa Bay Buccaneers      17   492   731   67.3  5229    43
## 2 Los Angeles Chargers      17   443   674   65.7  4800    38
## 3 Dallas Cowboys            17   444   647   68.6  4800    40
## 4 Kansas City Chiefs        17   448   675   66.4  4791    37
## 5 Los Angeles Rams          17   406   607   66.9  4642    41
```

You can find additional details on customizing kable tables at https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_pdf.pdf

```r
# Output using Kable (with no additional options)
library(kableExtra)

# Output using Kable and Kable-Styling and some additional options
NFL21 %>% kable() %>% kable_styling(latex_options="hold_position")
```

Tm	G	Cmp	Att	Cmp%	Yds	TD
Tampa Bay Buccaneers	17	492	731	67.3	5229	43
Los Angeles Chargers	17	443	674	65.7	4800	38
Dallas Cowboys	17	444	647	68.6	4800	40
Kansas City Chiefs	17	448	675	66.4	4791	37
Los Angeles Rams	17	406	607	66.9	4642	41

```r
# More options
NFL21 %>% kable(booktabs=TRUE) %>% kable_styling(latex_options="hold_position")
```

Tm	G	Cmp	Att	Cmp%	Yds	TD
Tampa Bay Buccaneers	17	492	731	67.3	5229	43
Los Angeles Chargers	17	443	674	65.7	4800	38
Dallas Cowboys	17	444	647	68.6	4800	40
Kansas City Chiefs	17	448	675	66.4	4791	37
Los Angeles Rams	17	406	607	66.9	4642	41

If you are going to use kable tables frequently in a document, you can write a short function to set your default options. After I ran the function below, the function *kt* will produce a kable table with my options set.

```
# kable table global setup
kt <- function(data) {
  knitr::kable(data, digits=3,linesep='',booktabs=TRUE) %>%
kable_styling(bootstrap_options='striped', latex_options='HOLD_position',
full_width = F, position = "center")
}
```

```
NFL21 %>% kt()
```

Tm	G	Cmp	Att	Cmp%	Yds	TD
Tampa Bay Buccaneers	17	492	731	67.3	5229	43
Los Angeles Chargers	17	443	674	65.7	4800	38
Dallas Cowboys	17	444	647	68.6	4800	40
Kansas City Chiefs	17	448	675	66.4	4791	37
Los Angeles Rams	17	406	607	66.9	4642	41

1.3 Baseball

Baseball rules YouTube video: https://www.youtube.com/watch?v=skOsApsF0jQ&t=63s

1.3.1 Hitting Statistics

1.3.1.1 Basic Hitting Statistics

- **Plate Appearances** (PA): number of completed batting appearances
- **At-Bats** (AB): Batting appearances, not including bases on balls, hit by pitch, sacrifices, interference, or obstruction
- **Hits** (H): Times reached base because of a batted, fair ball without error by the defense
- **Singles** (1B): Hits on which the batter reached first base safely without the contribution of a fielding error
- **Doubles** (2B): Hits on which the batter reached second base safely without the contribution of a fielding error
- **Triples** (3B): Hits on which the batter reached third base safely without the contribution of a fielding error
- **Home Runs** (HR): Hits on which the batter successfully touched all four bases, without the contribution of a fielding error
- **Total Bases** (TB): One for each single, two for each double, three for each triple, and four for each home run
- **Hit by Pitch** (HBP): Times touched by a pitch and awarded first base as a result
- **Sacrifice Fly** (SF): Number of fly ball outs which allow another runner to advance on the basepaths or score
- **Base on Balls** (BB or Walk): Times receiving four balls and advancing to first base
- **Intentional Base on Balls** (IBB or Intentional Walk): Times receiving four balls *intentionally* and advancing to first base
- **Strikeout** (K): Number of times that strike three is taken or swung at and missed, or bunted foul
- **Runs** (R): Times reached home base legally and safely
- **Runs Batted In** (RBI): Number of runners who scored due to a batters's action, except when batter grounded into double play or reached on an error
- **Batting Average** (AVG or BA): Hits divided by at bats
- **On Base Percentage/Average** (OBP or OBA): Times reached base (H + BB + HBP) divided by at bats plus walks plus hit by pitch plus sacrifice flies (AB + BB + HBP + SF)
- **Slugging Percentage/Average** (SLG): Total bases divided by at-bats
- **On-base Plus Slugging** (OPS): On-base percentage plus slugging average

1.3.1.2 Advanced Baseball Hitting Statistics

- **Isolated Power** (ISO): Slugging percentage minus Batting average
- **On-base Plus Slugging Plus** (OPS+): OPS normalized for park effects with 100 being league average
- **Weighted On-Base Average** (wOBA): Hitting rate statistic that attempts to credit the hitter for the value of each outcome. The following formula can be updated each year based on the scoring environment. The following formula was updated for the 2021 season.

$$wOBA = \frac{0.69 \cdot (BB - IBB) + 0.719 \cdot HBP + 0.87 \cdot 1B + 1.217 \cdot 2B + 1.529 \cdot 3B + 1.94 \cdot HR}{AB + BB - IBB + SF + HBP}$$

- **Expected Weighted On-Base Average** (xwOBA): Hitting rate statistic that attempts to credit the hitter for the value of each *expected* outcome based on Statcast data.

1.3.2 Pitching Statistics

1.3.2.1 Basic Pitching Statistics

- **Innings Pitched** (IP): Number of outs recorded while pitching divided by three
- **Strikeout** (K): Number of batters who received strike three
- **Base on Balls** (BB or Walk): Times pitching four balls, allowing the batter-runner to advance to first base
- **Hits Allowed** (H): Total hits allowed
- **Wins** (W): Number of games where pitcher was pitching while his team took the lead and went on to win
- **Losses** (L): Number of games where pitcher was pitching while the opposing team took the lead, never lost the lead, and went on to win
- **Earned Runs** (ER): Number of runs that did not occur as a result of errors or passed balls
- **Earned Run Average** (ERA): Earned runs times innings in a game (usually nine) divided by innings pitched
- **Walks and Hits Per Inning Pitched** (WHIP): Walks plus hits allowed divided by innings pitched

1.3.2.2 Advanced Pitching Statistics

- **Fielding Independent Pitching** (FIP): Statistic that estimates a pitcher's run prevention independent of the performance of the defense

$$FIP = \frac{13 \cdot HR + 3 \cdot (BB + HBP) - 2 \cdot K}{IP} + FIP_{constant}$$

The $FIP_{constant}$ is generally around 3.10 and is put FIP on a scale similar to ERA.

- **Expected Fielding Independent Pitching** (xFIP): Statistic that estimates a pitcher's expected run prevention independent of the performance of the defense

$$xFIP = \frac{13 \cdot (FlyBalls \cdot LgHR/FB\%) + 3 \cdot (BB + HBP) - 2 \cdot K}{IP} + FIP_{constant}$$

1.3.3 Wins Above Replacement

- **Wins Above Replacement** (WAR): Estimated number of wins that a player has outperformed a replacement player by with the same playing time. This is one of the most crucial statistics in Sabermetrics.

More about WAR from Fangraphs: https://library.fangraphs.com/misc/war/

References:
https://www.baseball-reference.com/bullpen/Baseball_statistics
https://blogs.fangraphs.com/glossary/
https://library.fangraphs.com/fangraphs-library-glossary/

1.3.4 Calculating Hitting Statistics

Example 1.10. Using the Colorado Rockies 2021 individual hitting statistics, calculate the AVG, OBA, SLG, OPS, ISO, wOBA.

```
# load data file and look at the header
rox21 <- read_csv("data/rockies_hitting2021.csv")
rox21 %>% slice_head(n=5) %>% kt()
```

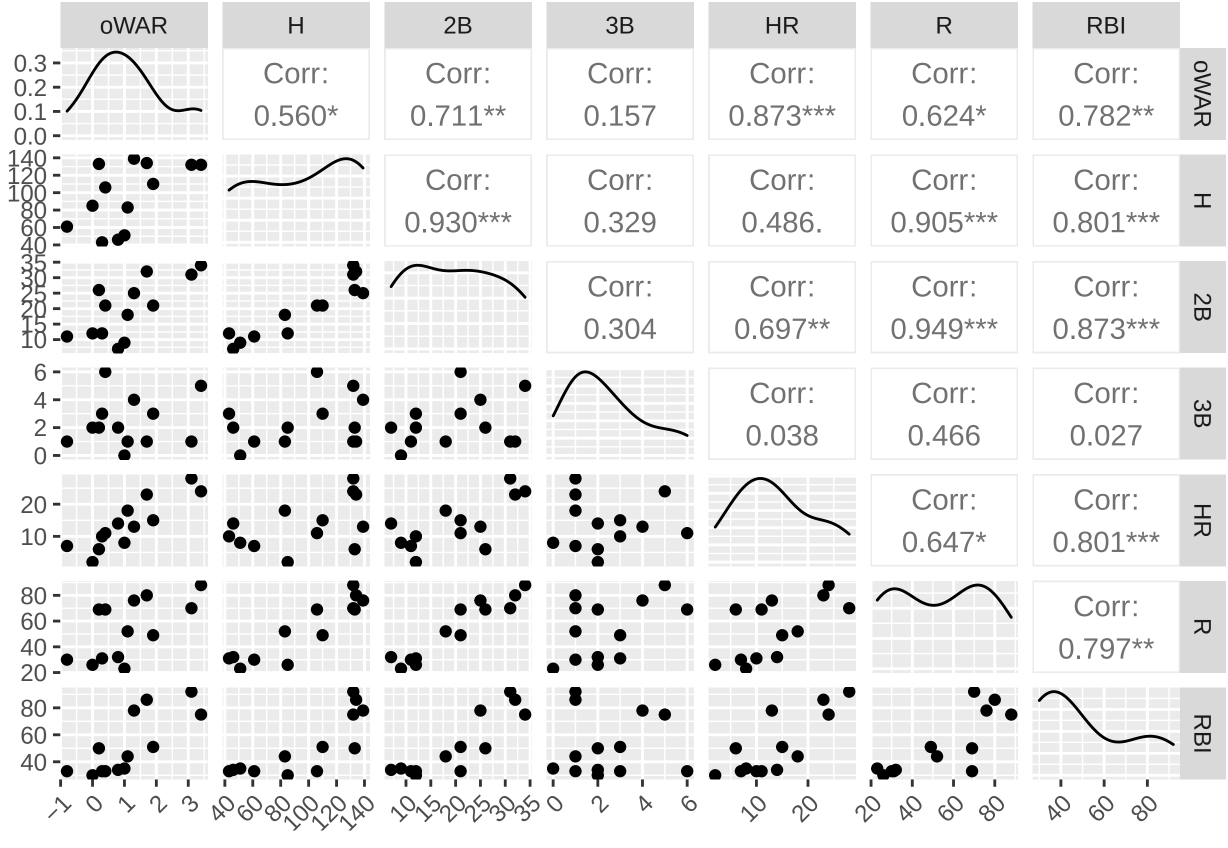
Name	PA	AB	R	H	2B	3B	HR	RBI	BB	IBB	SO	HBP	SF	oWAR
Elias Diaz	371	338	52	83	18	1	18	44	30	1	60	2	1	1.1
C.J. Cron	547	470	70	132	31	1	28	92	60	3	117	13	4	3.1
Brendan Rodgers	415	387	49	110	21	3	15	51	19	0	84	7	2	1.9
Trevor Story	595	526	88	132	34	5	24	75	53	2	139	11	5	3.4
Ryan McMahon*	596	528	80	134	32	1	23	86	59	2	147	4	5	1.7

```
# Create new variables using the mutate function
rox21 <- rox21 %>%
  mutate(AVG = H/AB,3) %>%
  mutate(OBA = (H + BB + HBP)/(AB + BB + HBP + SF)) %>%
  mutate(SLG = ((H-`2B`-`3B`-HR) + 2*`2B` + 3*`3B` + 4*HR)/AB) %>%
  mutate(OPS = SLG + OBA) %>%
  mutate(wOBA = (0.692 * (BB - IBB) + 0.722 * HBP + 0.879 * (H-`2B`-`3B`-HR) +
  1.242 * `2B` + 1.568 * `3B` + 2.007 * HR)/(AB + BB - IBB + SF + HBP)) %>%
  mutate(ISO = SLG-AVG)
rox21 %>% slice_head(n=5) %>% select(Name,PA,AVG,OBA,SLG,OPS,wOBA,ISO) %>% kt()
```

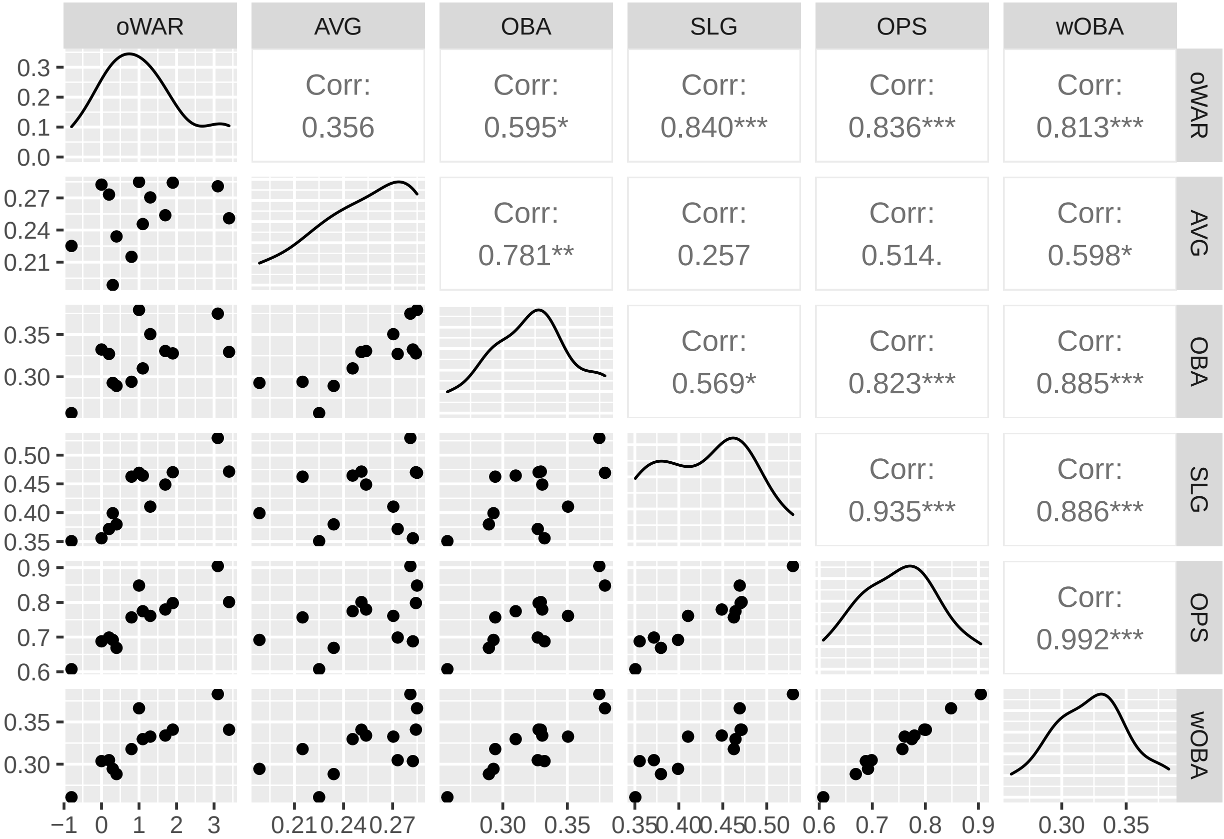
Name	PA	AVG	OBA	SLG	OPS	wOBA	ISO
Elias Diaz	371	0.246	0.310	0.464	0.774	0.330	0.219
C.J. Cron	547	0.281	0.375	0.530	0.905	0.383	0.249
Brendan Rodgers	415	0.284	0.328	0.470	0.798	0.341	0.186
Trevor Story	595	0.251	0.329	0.471	0.801	0.341	0.221
Ryan McMahon*	596	0.254	0.331	0.449	0.779	0.334	0.195

Example 1.11. oWAR is Baseball Reference's offensive WAR statistic. Note that Baseball Reference and Fangraphs use different formulas when calculating WAR though their results are typically similar. For Rockies players with at least 100 at-bats in 2021, what hitting statistics are most and least correlated to oWAR?

```
# Let's remove players with less than 100 ABs
rox21_100 <- rox21 %>% filter(AB >= 100)

# GGally package has a nice pairs plotting function
library("GGally")

# Standard hitting statistics
rox21_100 %>% select(oWAR,H,`2B`,`3B`,HR,R,RBI) %>% ggpairs() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Rate statistics
rox21_100 %>% select(oWAR,AVG,OBA,SLG,OPS,wOBA) %>% ggpairs()
```


oWAR
AVG
OBA
SLG
OPS
wOBA
Corr:
0.356
Corr:
0.595*
Corr:
0.840***
Corr:
0.836***
Corr:
0.813***
Corr:
0.781**
Corr:
0.257
Corr:
0.514.
Corr:
0.598*
Corr:
0.569*
Corr:
0.823***
Corr:
0.885***
Corr:
0.935***
Corr:
0.886***
Corr:
0.992***
0.3
0.2
0.1
0.0
0.27
0.24
0.21
0.35
0.30
0.50
0.45
0.40
0.35
0.9
0.8
0.7
0.6
0.35
0.30
−1 0 1 2 3
0.21 0.24 0.27
0.30 0.35
0.35 0.40 0.45 0.50
0.6 0.7 0.8 0.9
0.30 0.35

1.3.5 Evaluating Pitching Statistics

Earned run average (ERA) has been traditionally used to evaluate a pitcher, however, it has some flaws. First of all, it is highly dependent on the fielders playing behind the pitcher. If a pitcher's shortstop has poor range, he won't convert as many groundballs into outs as a shortstop with good range. ERA is also a noisy measurement in that can be affected easily by random luck.

To overcome some of the downsides of ERA, FIP and xFIP were developed to help reduce the variance (noise) of the measurement and to remove factors, like defense, that are not a function of the pitcher's ability.

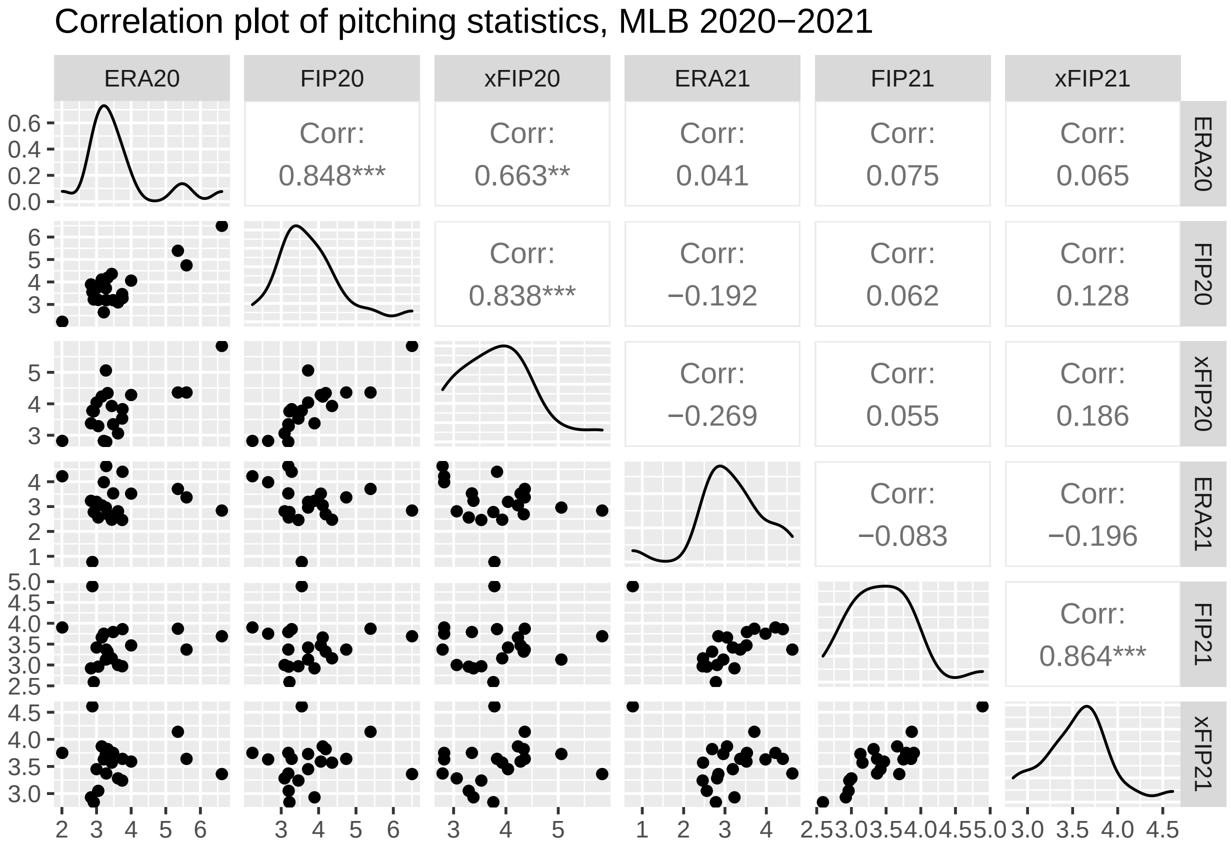
Let's look at the twenty starting pitchers that had the most innings pitched in MLB for the total of the 2020 and 2021 seasons. We want to examine the year-to-year correlation between ERA, FIP, and xFIP.

```r
pitchers2021 <- read_csv("data/MLBpitchers20-21.csv")
pitchers2021 %>% slice_head(n=5) %>% kt()
```

Player	ERA20	FIP20	xFIP20	ERA21	FIP21	xFIP21
Zack Wheeler	2.92	3.22	3.76	2.78	2.59	2.84
Adam Wainwright	3.15	4.11	4.23	3.05	3.66	3.87
Kyle Hendricks	2.88	3.55	3.78	0.77	4.89	4.61
German Marquez	3.75	3.28	3.83	4.40	3.86	3.64
Luis Castillo	3.21	2.65	2.82	3.98	3.75	3.63

Let's look at the correlations between these variables, paying close attention to what variables are most correlated with ERA20, a pitcher's ERA in 2020.

```r
pitchers2021 %>%
  select(-Player) %>%
  ggpairs(title="Correlation plot of pitching statistics, MLB 2020-2021")
```



This is a small dataset that only contains twenty pitchers (samples), but you will notice that ERA20 is more highly correlated with FIP21 and xFIP21 than ERA21. In other words, FIP and xFIP are likely better predictors of ERA success in the future rather than ERA success in the past.

1.4 Football

Link to YouTube video describing football rules

1.4.1 Basic Football Statistics

- **Yards** (Yd): Number of yards gained from the line of scrimmage (can be broken down into Offense: Rushing and Passing Yards, Defense: Rushing and Passing Yards Allowed, and Special Teams: Kick and Punt Return Yards)
- **Touchdowns** (TD): Number of times the offense carries or passes the ball successfully into the end zone of the opposing side (can be broken down into: Offensive TDs: Rushing and Passing TDs, Defensive TDs: interception or fumble recovery for a touchdown, and Special Teams TDs: kick or punt return touchdowns)
- **Sacks** (Sk): Number of times a player or a team tackles the quarterback behind the line of scrimmage before he can throw a pass
- **Interceptions** (INT): Number of times a player or a team catches an opponent's pass

1.4.2 Advanced Football Statistics

- **Passer Rating (or Passing Efficiency)**: Measure of performance of a quarterback

For NFL, the formula is as follows:

$$PasserRating_{NFL} = \left(\frac{a+b+c+d}{6}\right) \cdot 100$$

where:

$$a = \left(\frac{COMP}{ATT} - 0.3\right) \cdot 5$$

$$b = \left(\frac{YDS}{ATT} - 3\right) \cdot 0.25$$

$$c = \left(\frac{TD}{ATT}\right) \cdot 20$$

$$d = 2.375 - \left(\frac{INT}{ATT} \cdot 25\right)$$

ATT = Number of passing attempts, COMP = Number of completions, YDS = Passing yards, TD = Touchdown passes, INT = Interceptions

Note: If the result of any calculation is greater than 2.375, it is set to 2.375. If the result is a negative number, it is set to zero.

For College Football, the formula is as follows:

$$PasserRating_{NCAAF} = \frac{(8.4 \cdot YDS) + (330 \cdot TD) + (100 \cdot COMP) - (200 \cdot INT)}{ATT}$$

- **Total Quarterback Rating** (QBR): Proprietary measure of performance of a quarterback developed by ESPN in 2011. This is a more comprehensive measurement of quarterback performance that accounts for the quarterback's impact on his team's passes, rushes, turnovers, and penalties in terms of **expected points added**.
- **Expected Points Added** (EPA): Measure of how many points a player or a play is worth to a team.

References:
https://www.pro-football-reference.com/about/glossary.htm
https://en.wikipedia.org/wiki/Passer_rating

Example 1.12. Individual NFL quarterback passing statistics for the 2021 season are provided in `NFL_Ind_Passing_2021.csv`. Note that this dataset only includes quarterbacks with at least 100 passing yards in 2021. Passer efficiency is given in the `RTG` column.

```
# Data: https://www.espn.com/nfl/stats/player
QB_21 <- read_csv("data/NFL_Ind_Passing_2021.csv")
QB_21 %>% select(1,3:8,11:12,15:16) %>% slice_head(n=10) %>% kt()
```

Name	GP	CMP	ATT	CMP%	YDS	AVG	TD	INT	QBR	RTG
Tom Brady	17	485	719	67.5	5316	7.4	43	12	68.1	102.1
Justin Herbert	17	443	672	65.9	5014	7.5	38	15	65.6	97.7
Patrick Mahomes	17	436	658	66.3	4839	7.4	37	13	62.2	98.5
Josh Allen	17	409	646	63.3	4407	6.8	36	15	60.7	92.2
Derek Carr	17	428	626	68.4	4804	7.7	23	14	52.4	94.0
Ben Roethlisberger	16	390	605	64.5	3740	6.2	22	10	35.6	86.8
Trevor Lawrence	17	359	602	59.6	3641	6.0	12	17	33.5	71.9
Matthew Stafford	17	404	601	67.2	4886	8.1	41	17	63.8	102.9
Dak Prescott	16	410	596	68.8	4449	7.5	37	10	54.6	104.2
Kirk Cousins	16	372	561	66.3	4221	7.5	33	7	52.3	103.1

```
names(QB_21)
```

```
##  [1] "Name"   "Team"   "GP"     "CMP"    "ATT"    "CMP%"   "YDS"    "AVG"    "YDS/G"
## [10] "LNG"    "TD"     "INT"    "SACK"   "SYL"    "QBR"    "RTG"
```

(a) Confirm this is passer rating by creating a new variable to calculate passer rating using the provided statistics.

```
# Create intermediate variables
QB_21 <- QB_21 %>%
  mutate(a = (CMP/ATT-0.3)*5) %>%
  mutate(b = (YDS/ATT-3)*0.25) %>%
  mutate(c = (TD/ATT)*20) %>%
  mutate(d = 2.375 -(INT/ATT*25))

# Check to see if a,b,c,d are less than 0 or greater than 2.375
QB_21 %>% summarize(min(a),max(a),min(b),max(b),min(c),max(c),min(d),max(d))
```

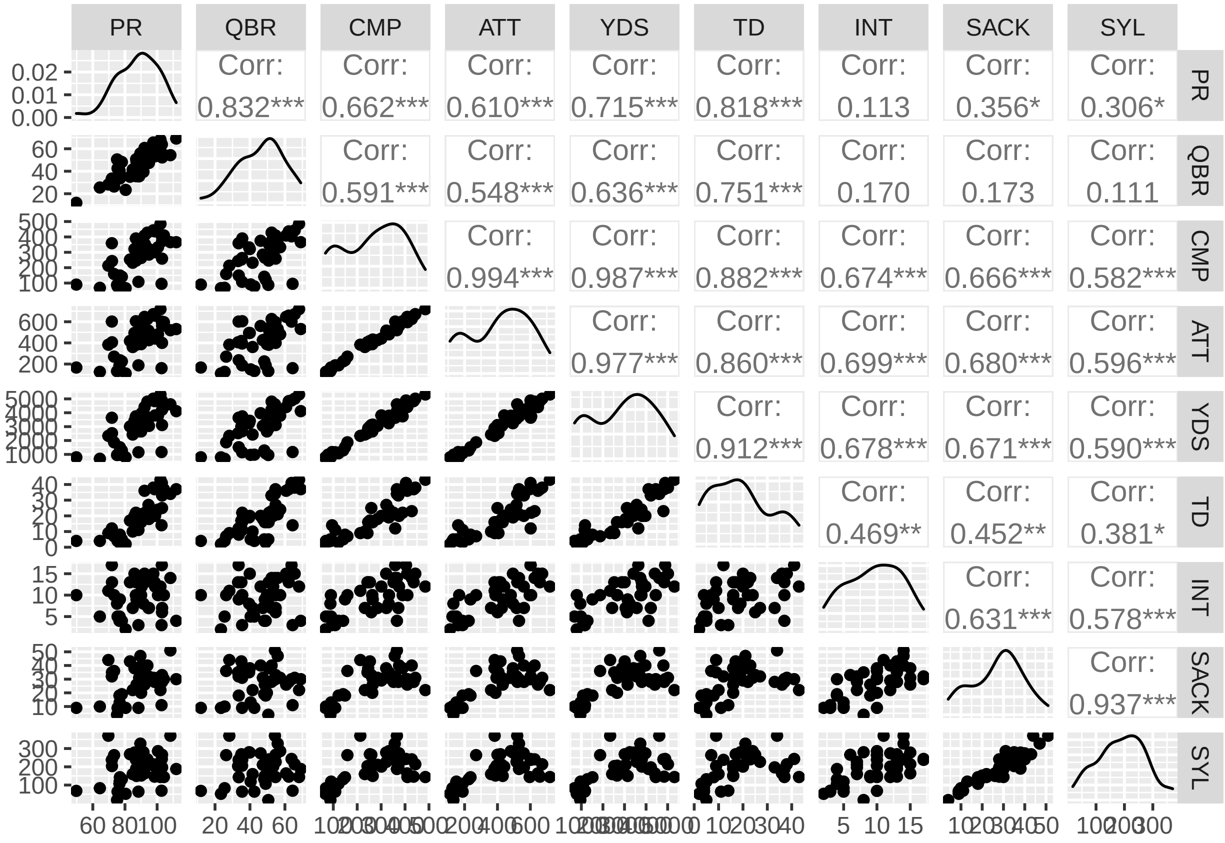
```
## # A tibble: 1 x 8
##   `min(a)` `max(a)` `min(b)` `max(b)` `min(c)` `max(c)` `min(d)` `max(d)`
##      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1     1.19     2.02    0.433     1.47    0.319     1.74    0.860     2.19
```

```
QB_21 <- QB_21 %>%
  mutate(PR = (a+b+c+d)/6*100)

QB_21 %>% select(Name,RTG,PR) %>% slice_head(n=5) %>% kt()
```

Name	RTG	PR
Tom Brady	102.1	102.083
Justin Herbert	97.7	97.656
Patrick Mahomes	98.5	98.455
Josh Allen	92.2	92.170
Derek Carr	94.0	93.963

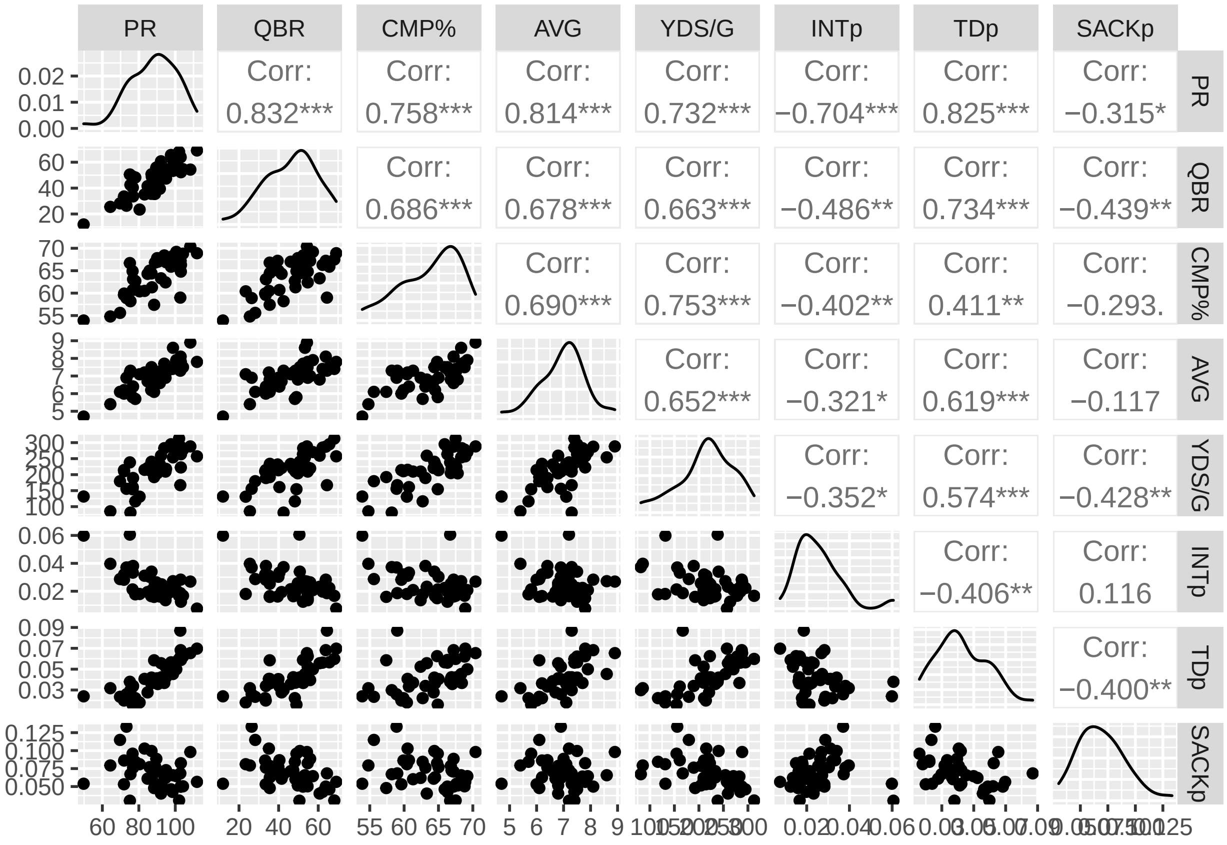
(b) What counting statistics are most correlated with passer rating and with QBR?

```
# Grab the counting statistics and create a correlation plot with PR and QBR
QB_21 %>% select(PR,QBR,CMP,ATT,YDS,TD,INT,SACK,SYL) %>% ggpairs()
```



(c) What rate statistics are most correlated with passer efficiency and with QBR? Use CMP% (completion percentage, completions per attempt), AVG (average yards per attempt), YDS/G (yards per game), and create new variables for interceptions per attempt, touchdowns per attempt, and sacks per attempt.

```
# Grab the counting statistics and create a correlation plot with PR and QBR
QB_21 <- QB_21 %>%
  mutate(INTp = INT/ATT) %>%
  mutate(TDp = TD/ATT) %>%
  mutate(SACKp = SACK/ATT)

QB_21 %>% select(PR,QBR,`CMP%`,AVG,`YDS/G`,INTp,TDp,SACKp) %>% ggpairs()
```


PR
QBR
CMP%
AVG
YDS/G
INTp
TDp
SACKp
Corr: 0.832***
Corr: 0.758***
Corr: 0.814***
Corr: 0.732***
Corr: −0.704***
Corr: 0.825***
Corr: −0.315*
Corr: 0.686***
Corr: 0.678***
Corr: 0.663***
Corr: −0.486**
Corr: 0.734***
Corr: −0.439**
Corr: 0.690***
Corr: 0.753***
Corr: −0.402**
Corr: 0.411**
Corr: −0.293.
Corr: 0.652***
Corr: −0.321*
Corr: 0.619***
Corr: −0.117
Corr: −0.352*
Corr: 0.574***
Corr: −0.428**
Corr: −0.406**
Corr: 0.116
Corr: −0.400**
0.00 0.01 0.02
20 40 60
55 60 65 70
5 6 7 8 9
100 150 200 250 300
0.02 0.04 0.06
0.03 0.05 0.07 0.09
0.050 0.075 0.100 0.125
60 80 100
20 40 60
55 60 65 70
5 6 7 8 9
100150200250300
0.02 0.04 0.06
0.030.050.070.09
0.0500.0750.1000.125

1.5 Basketball

Link to YouTube video describing basketball rules

1.5.1 Basic Basketball Statistics

- **Field Goal (FG)**: A made shot from either 2- or 3-point range. Free throws, worth 1 point, are not considered field goals. Field goal statistics often include attempts, makes, and percentage.
- **Free Throw (FT)**: After certain fouls, the clock stops and a player shoots an uncontested shot from the foul line. These free throws are worth 1 point each; like with field goals, FT statistics often include attempts, makes, and percentage.
- **Assists (AST)**: A player is credited with an assist if they pass the ball to a teammate and the teammate scores a field goal after zero or one dribbles. No more than one assist can be recorded per field goal.
- **Turnover (TO)**: A player or team can be charged with a turnover for an action or violation that ends their offensive possession before being able to attempt a field goal. For a player (especially a guard), TOs can be compared to assists using Assist:Turnover ratio.
- **Rebound (REB)**: The first player to gain control of the ball following a missed field goal is credited with a rebound. If the player is on the same team as the field goal shooter, it is an offensive rebound; otherwise, a defensive rebound.
- **Points per Possession (PPP)**: Divides a team's points by number of possessions to account for a team's pace.

1.5.2 Advanced Basketball Statistics

- **True Shooting Percentage (TS%)**: Unlike traditional shooting percentage, this statistic considers both field goals and free throws. It also gives more weight to shots that are worth more points.
- **Win Shares**: Win shares give each player points for actions that contribute to a team's success. Win shares take into account a variety of offensive and defensive statistics, but can be calculated using different methods on different platforms.
- **Value Over Replacement Player (VORP)**: This is basketball's response to baseball's WAR. VORP is a rate statistic that estimates a player's offensive output as compared to an "average" player.
- **Player Efficiency Rating (PER)**: According to its creator John Hollinger, "The PER sums up all a player's positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player's performance." This statistic rewards great offensive performance more than great defensive plays.

References:
https://www.basketball-reference.com/about/per.html
https://www.basketball-reference.com/about/ws.html
https://www.basketball-reference.com/about/glossary.html

1.5.3 Four Factors

Tibbles are a type of data frame supported by the `tidyverse` package. The following tibble contains data from a Mountain West tournament game played between the CSU and Wyoming women's basketball teams during the 2021-2022 season, which CSU won 51-38. (Here's the link to the box score on the CSU athletics website.)

```
basketball_data <- tibble(team = c("CSU","WYO"), FG = c(14,15), FGA = c(48,60),
                          THREEP = c(5,4), FT = c(10,4), FTA = c(14,4),
                          ORB = c(2,14), DRB = c(31,30), TOV = c(5,12))
basketball_data %>% kt()
```

team	FG	FGA	THREEP	FT	FTA	ORB	DRB	TOV
CSU	14	48	5	10	14	2	31	5
WYO	15	60	4	4	4	14	30	12

This tibble contains all the data needed to calculate the *Four Factors*. The Four Factors of a basketball game are statistics formulated by Dean Oliver, former Director of Quantitative Analysis for the Denver Nuggets (among other roles). These statistics are also promoted by sports data platforms like Hudl.com.

The Four Factors each have offensive and defensive versions; for this example, we'll focus on the offensive perspective.

1. ***Scoring***: **Effective Field Goal Percentage** (eFG%)

$eFG\% = \frac{FG\ +\ 0.5(3P)}{FGA}$

2. ***Crashing***: **Turnover Percentage** (TOV%)

$TOV\% = \frac{TOV}{FGA\ +\ 0.44(FTA)\ +\ TOV}$

3. ***Protecting***: **Rebounding Percentage** (ORB%)

$ORB\% = \frac{ORB}{ORB\ +\ Opponent\ DRB}$

4. ***Attacking***: **Free Throw Factor** (FT Factor)

$FT\ factor = \frac{FT}{FGA}$

Let's calculate the values of eFG%, TOV%, ORB%, and Free Throw Factor for both CSU and Wyoming and add them as new columns in the tibble using the `add_column` function.

```
basketball_data <- basketball_data %>%
  mutate(eFG = 100*(FG + .5 * THREEP)/FGA) %>%
  mutate(TOVPCT = 100*(FG + .5 * THREEP)/FGA) %>%
  mutate(ORBPCT = 100*c(ORB[1]/(ORB[1]+DRB[2]), ORB[2]/(ORB[2]+DRB[1]))) %>%
  mutate(FTFACTOR = 100*FT/FGA)

basketball_data %>% select(team,eFG, TOVPCT, ORBPCT, FTFACTOR) %>% kt()
```

team	eFG	TOVPCT	ORBPCT	FTFACTOR
CSU	34.375	34.375	6.250	20.833
WYO	28.333	28.333	31.111	6.667

While this method does produce Four Factors data, it could be difficult to scale for calculating the same statistics for a sample of several games. In the next section, we will introduce an `R` package that aids in the calculation of Four Factors.

1.5.3.1 BasketballAnalyzeR Four Factors

The authors of "Basketball Data Science With Applications in R" developed the `BasketballAnalyzeR` package to be used in conjunction with the book. `BasketballAnalyzeR` includes built-in datasets from the 2017-18 NBA season and provides many functions for analyzing and plotting basketball data. One such function is `fourfactors`, which offers a simpler way to perform a four factors analysis.

```r
library("BasketballAnalyzeR")

teams <- c("DEN", "CLE", "GSW") #Nuggets, Cavaliers, Warriors
team_data <- which(Tadd$team %in% teams)
four_factors_teams <- fourfactors(Tbox[team_data, ], Obox[team_data, ])
four_factors_teams %>% select(1,8:15) %>% kt()
```

Team	F1.Off	F2.Off	F3.Off	F4.Off	F1.Def	F2.Def	F3.Def	F4.Def
Cleveland Cavaliers	54.70	13.70	20.06	21.41	53.98	13.43	77.27	16.58
Denver Nuggets	53.62	14.90	25.66	19.77	53.88	13.83	77.45	17.35
Golden State Warriors	56.91	15.26	21.05	19.48	50.44	13.89	76.31	18.55

This is a much simpler and neater way to calculate Four Factors.

In the 2017-18 season, the Warriors and Cavaliers met in the NBA Finals, while the Nuggets just missed the playoffs. It would be expected that the two Finals teams would have higher values for the Four Factors. While this is mostly true, for which of the Four Factors did the Nuggets have the highest value?

A: Factor 3 (rebounding percentage), both offensive and defensive.

1.5.4 Shot Charts

The `BasketballAnalyzeR` package includes shot location data for all players for the 2017-18 NBA season and has a function called `shotchart` that allows for the plotting of shot data.

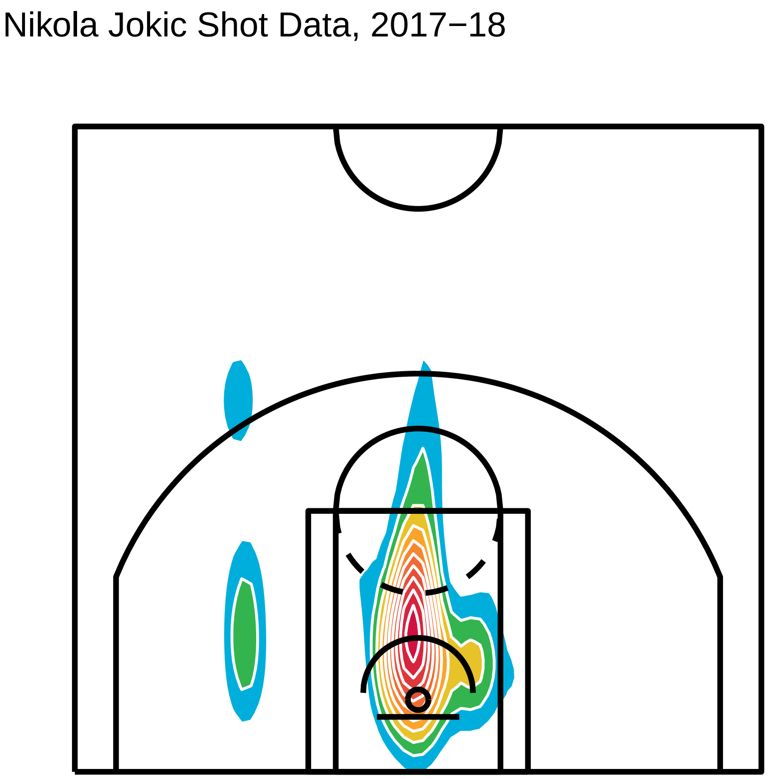
Let's plot shot location data for Nikola Jokic. First, the coordinates must be transformed so that the point (0,0) is located at the corner of the court; the original coordinates place the origin at the center of the hoop.

```r
PbP <- PbPmanipulation(PbP.BDB)

jokic_data <- subset(PbP, player == "Nikola Jokic")
```

```r
jokic_data$xx <- jokic_data$original_x/10 #transformation
jokic_data$yy <- jokic_data$original_y/10 - 41.75 #transformation
```

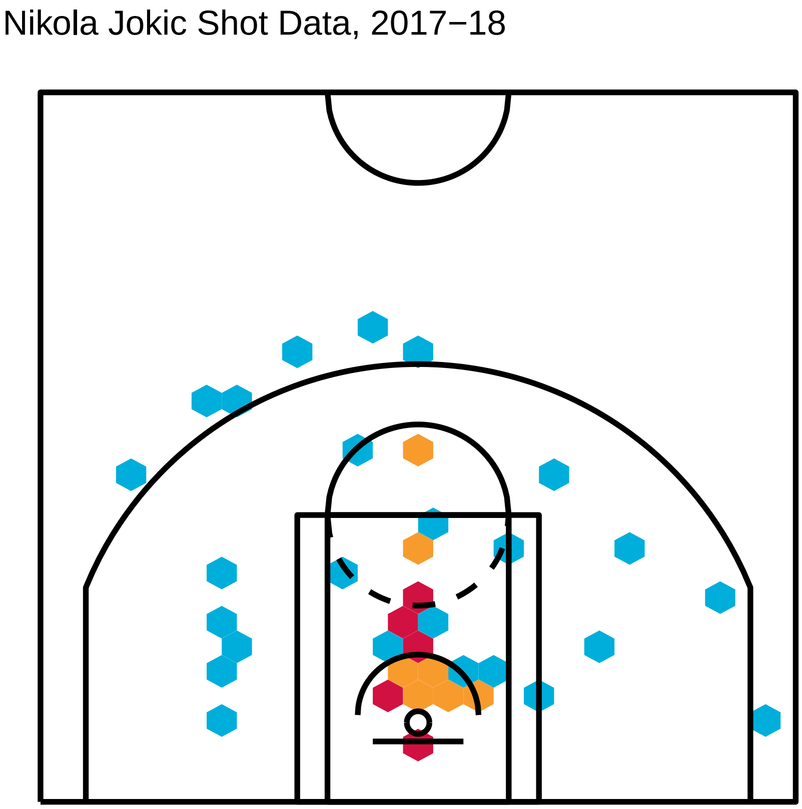
`BasketballAnalyzeR` supports three types of density visualizations within `shotchart`, one being `density-polygons`. Since `shotchart` is a `ggplot` object, a chart title can be added using `ggtitle`.

```r
shotchart(data = jokic_data, x="xx", y="yy", type="density-polygons") +
  ggtitle("Nikola Jokic Shot Data, 2017-18")
```



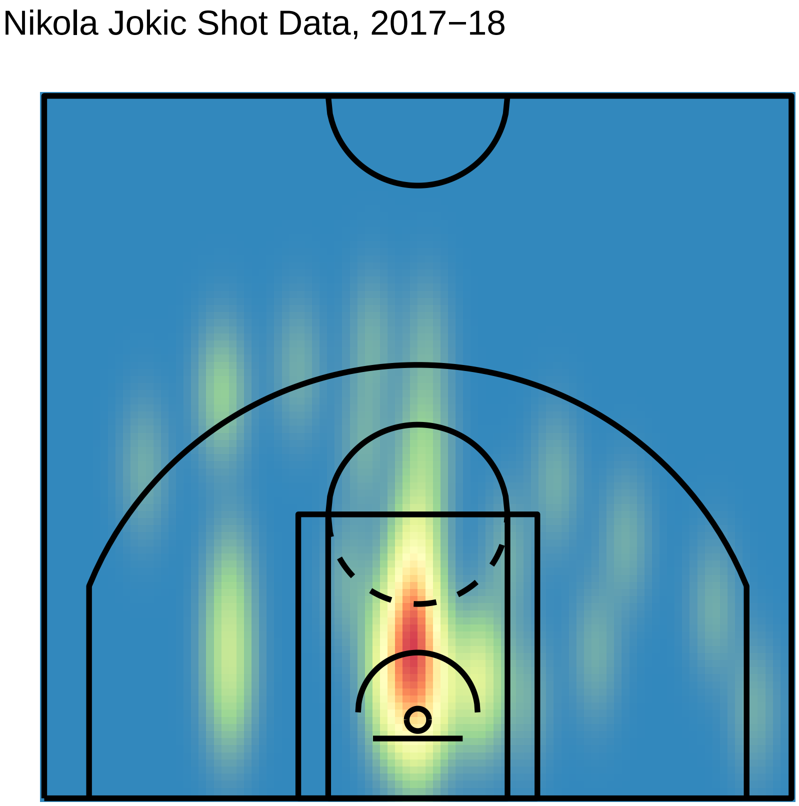
It seems most shots attempts from Jokic were in the paint; this is hardly surprising, since he plays the center position. Here's the same chart with `density-hexbin`:

```r
shotchart(data = jokic_data, x="xx", y="yy", type="density-hexbin") +
  ggtitle("Nikola Jokic Shot Data, 2017-18")
```



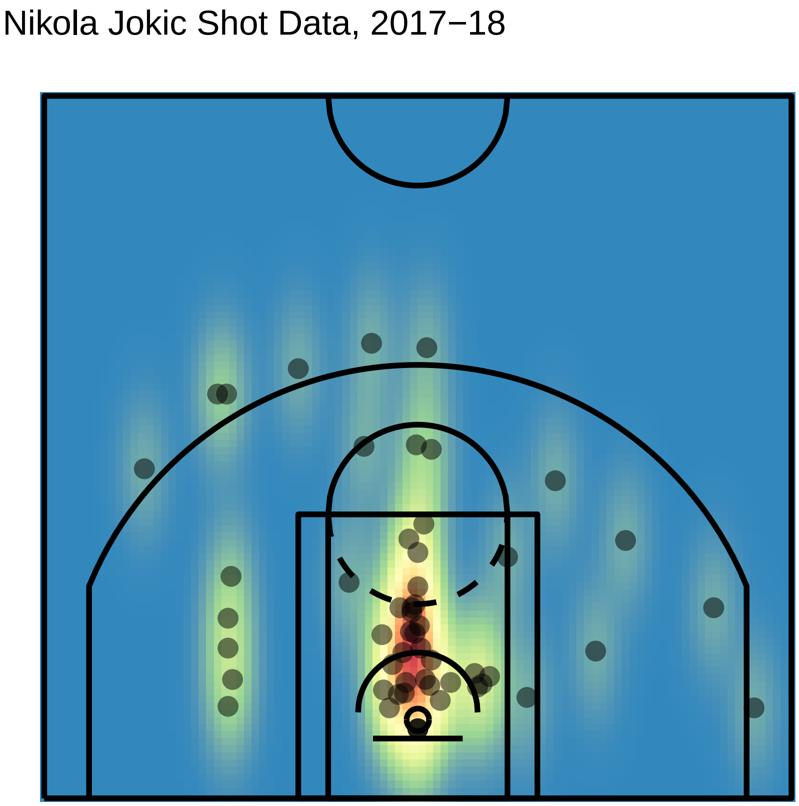
The same chart with `density-raster`:

```
shotchart(data = jokic_data, x="xx", y="yy", type="density-raster") +
  ggtitle("Nikola Jokic Shot Data, 2017-18")
```



Within the `shotchart` function, setting `scatter=TRUE` overlays the shots on the chart. Point size and transparency can also be customized.

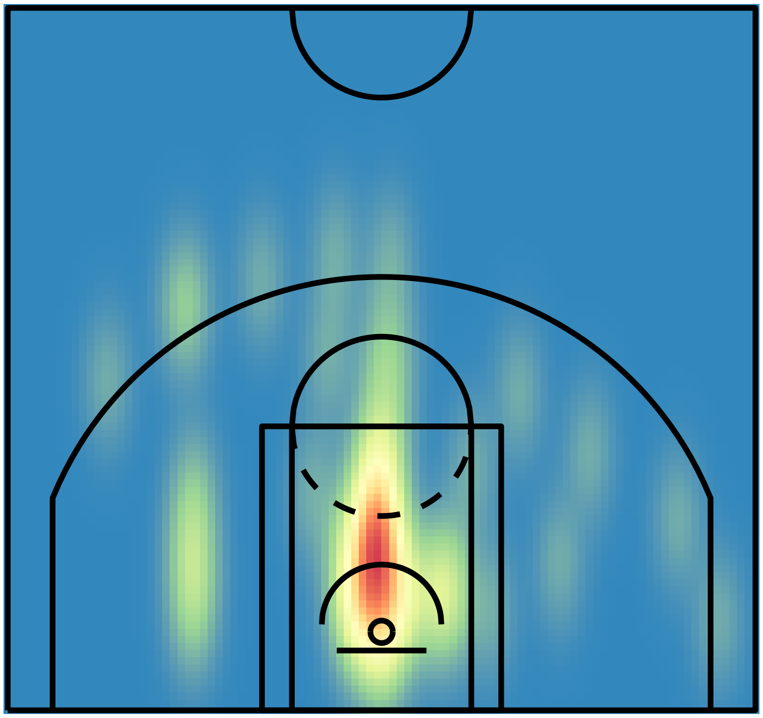
```
shotchart(data = jokic_data, x="xx", y="yy", type="density-raster", scatter=TRUE)
+
  ggtitle("Nikola Jokic Shot Data, 2017-18")
```



Let's now compare shot charts of Nikola Jokic, Steph Curry, and Lebron James. This group of players includes one member of each team for which we calculated Four Factors.

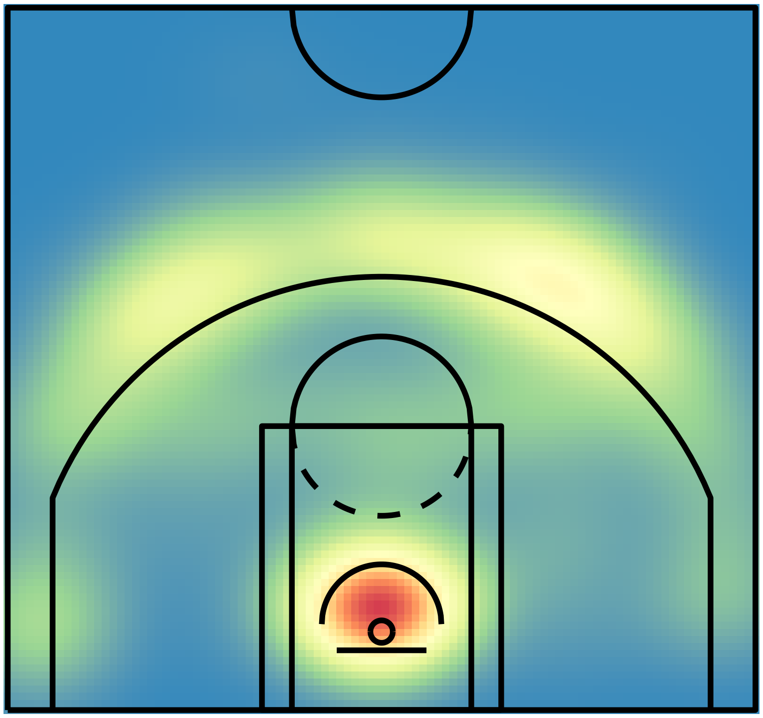
```
curry_data <- subset(PbP, player == "Stephen Curry")
curry_data$xx <- curry_data$original_x/10 #transformation
curry_data$yy <- curry_data$original_y/10 - 41.75 #transformation

james_data <- subset(PbP, player == "LeBron James")
james_data$xx <- james_data$original_x/10 #transformation
james_data$yy <- james_data$original_y/10 - 41.75 #transformation

shotchart(data = jokic_data, x="xx", y="yy", type="density-raster") +
  ggtitle("Nikola Jokic Shot Data, 2017-18")
```

Nikola Jokic Shot Data, 2017−18

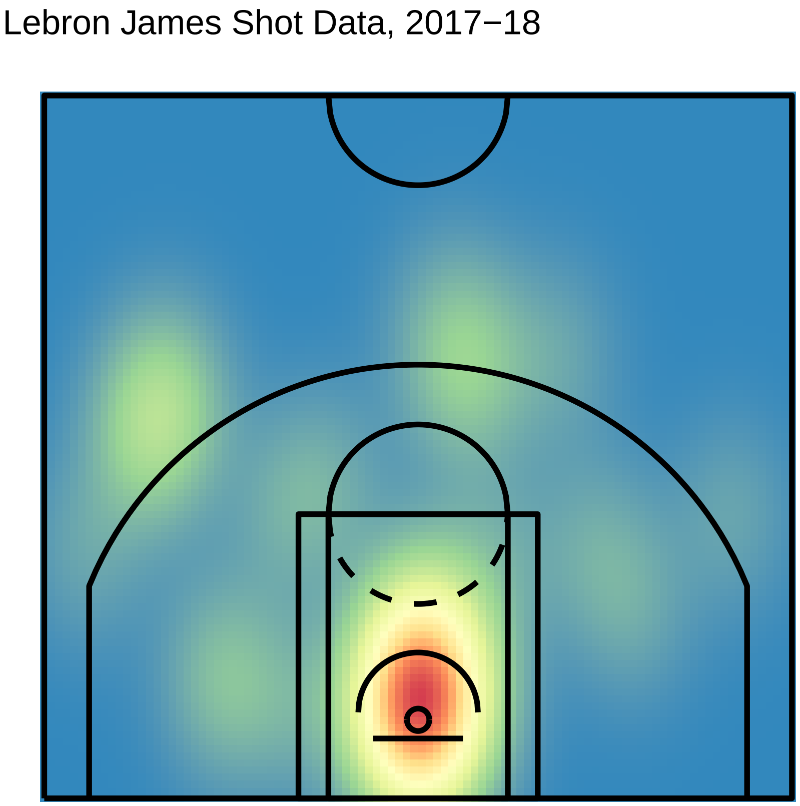


```
shotchart(data = curry_data, x="xx", y="yy", type="density-raster") +
  ggtitle("Steph Curry Shot Data, 2017-18")
```

Steph Curry Shot Data, 2017−18



```
shotchart(data = james_data, x="xx", y="yy", type="density-raster") +
  ggtitle("Lebron James Shot Data, 2017-18")
```

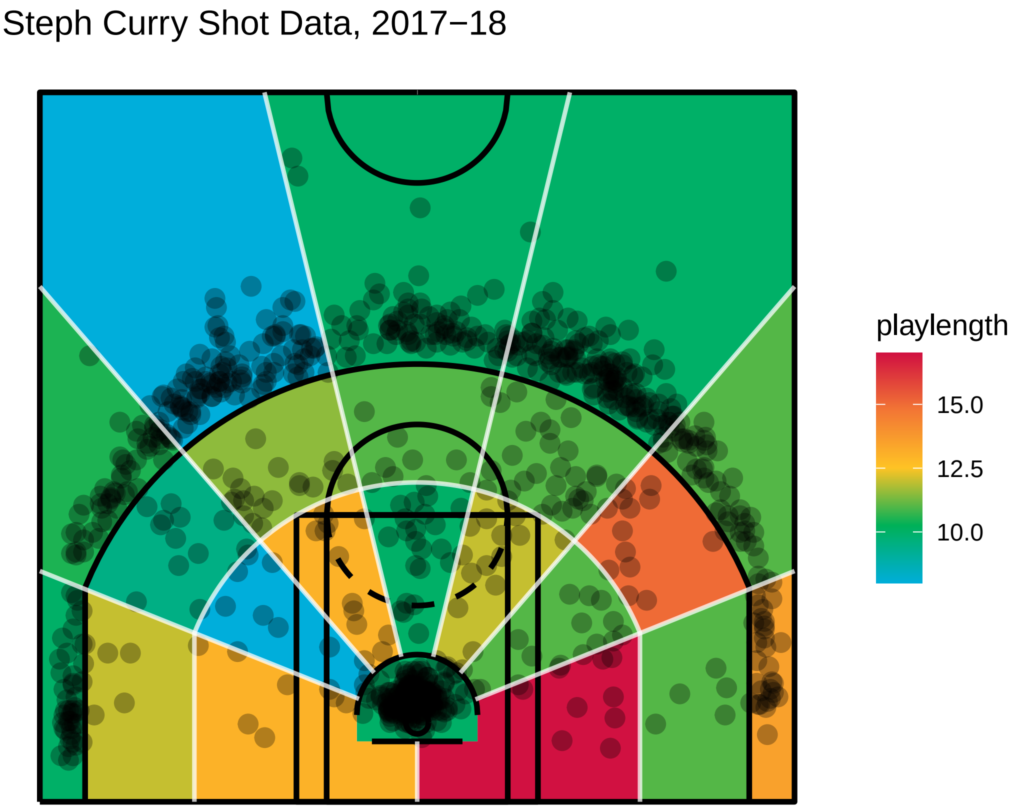


Q: Of the three players (Jokic, Curry, James), which took the highest percentage of their three-point shots from the right side of the court (when facing the basket)?

A: Nikola Jokic shot almost all of his attempts from the right side. Steph Curry took many shots from beyond the arc and tended toward the left side, while James was split between the right side and the center.

Now, let's focus on Steph Curry's shooting. The following chart splits the court into zones based on angle and distance from the basket. The color in each zone represents the average length of the play leading up to that shot among shots taken in that zone.

```
shotchart(data = curry_data, x="xx", y="yy", z="playlength",
          type="sectors", num.sect = 7, scatter=TRUE, pt.alpha=.3) +
  ggtitle("Steph Curry Shot Data, 2017-18")
```



Q: In general, did Steph Curry tend to shoot closer to the basket during plays of a longer duration or a shorter duration?

A: There is not a perfect correlation, but it seems that two-point field goals were attempted more often during longer plays, while shots taken outside the three-point arc were taken during plays of a shorter duration.

References: https://rdrr.io/cran/BasketballAnalyzeR/
Basketball Data Science (Zuccolotto and Manisera, 2020)

1.6 Hockey

Link to YouTube video describing hockey rules

1.6.1 Basic Hockey Statistics

Here are some basic statistics that are used often to describe hockey games.

- **Goals (G)**: If a team scores, the skater on the scoring team who last touched the puck is credited with a goal.
- **Assists (A)**: The players (up to two) on the scoring team who last touch the puck before the goalscorer are credited with assists, unless the opposing team has possession of the puck in between.
- **Points (PTS)**: Goals plus assists. [Not to be confused with team points awarded in the regular season standings by the many hockey leagues, including the NHL (two points for a win, one point for an overtime/shootout loss, zero points for a regulation loss)].
- **Shots On Goal (SOG)**: Shot attempts in which the puck has been shot directly on goal. Shot attempts which are blocked or miss the goal are not considered SOGs. A team's shots on goal should equal the opposing goaltender's saves plus the team's goals scored.
- **Goals Against Average (GAA)**: Of a goaltender, the number of goals allowed by that goaltender adjusted to a per-60 minute rate.
- **Penalty Minutes (PIM)**: The amount of penalty time an individual player is assigned for their infractions. PIM may be different than the amount of time the player actually spends in the penalty box.

Reference:
https://www.milehighhockey.com/pages/stats

1.6.2 Advanced Hockey Statistics

- **CORSI**: CORSI only applies to 5 on 5 ("even-strength") situations. It is calculated as the difference between shot attempts on offense (shots on goal + blocked shots + missed shots) minus shot attempts allowed on defense. CORSI can also be expressed as a percentage, with percentages over 50% indicating that the player is on ice for more offensive shots than defensive shots.
- **Expected Goals (xG)**: Expected Goals statistics give each shot an estimated probability of scoring a goal based on factors such as shot location and game situation. xG cannot be less than 0 or greater than 1 for any particular shot, and different platforms may have different methods of calculating expected goals.
- **Fenwick/Unblocked Shot Attempts (USAT)**: Similar to CORSI, but omits blocked shots from the calculation. This statistic is used in many Expected Goals calculations.

Because the flow of a hockey game is usually quite different in situations other than the normal 5 on 5, such as a power play (5 on 4) or concurrent penalties (4 on 4), many hockey databases separate data by the type of game situation. We will see this below with a dataset from MoneyPuck, but it is also present on Natural Stat Trick, QuantHockey, and hockey-reference.

References:
https://www.nhl.com/lightning/news/hockey-analytics-101-understanding-advanced-stats-and-how-theyre-measured/c-735819
https://theathletic.com/121980/2017/10/09/an-advanced-stat-primer-understanding-basic-hockey-metrics/

1.6.3 Actual vs. Expected Goals

Example 1.13. For this example, we’ll use a set of NHL data from moneypuck.com. First, let’s load the data into `R` and open the data frame.

```
url <-
"https://moneypuck.com/moneypuck/playerData/seasonSummary/2021/regular/teams.csv"
nhl_2022_data <- read_csv(url)

nhl_2022_data %>%
  slice_head(n=10) %>%
  select(3,6,8,9,10) %>% kt()
```

name	situation	xGoalsPercentage	corsiPercentage	fenwickPercentage
WPG	other	0.49	0.50	0.47
WPG	all	0.49	0.50	0.50
WPG	5on5	0.49	0.49	0.50
WPG	4on5	0.16	0.14	0.15
WPG	5on4	0.86	0.86	0.85
CBJ	other	0.52	0.49	0.49
CBJ	all	0.45	0.48	0.47
CBJ	5on5	0.45	0.48	0.47
CBJ	4on5	0.14	0.18	0.21
CBJ	5on4	0.81	0.84	0.82

We can create nice looking tables using the “kableExtra’’ package. Let’s look at the first eight rows and a small selection of columns of the data frame and format the table output using a kable table.

```
library("kableExtra")

nhl_2022_data[1:8, c(3,6:9)] %>% kt()
```

name	situation	games_played	xGoalsPercentage	corsiPercentage
WPG	other	82	0.49	0.50
WPG	all	82	0.49	0.50
WPG	5on5	82	0.49	0.49
WPG	4on5	82	0.16	0.14
WPG	5on4	82	0.86	0.86
CBJ	other	82	0.52	0.49
CBJ	all	82	0.45	0.48
CBJ	5on5	82	0.45	0.48

This dataset includes a *lot* of covariates. It also splits these data by different game situations: even-strength (5 on 5), power play (5 on 4), etc. Let's subset the data to include all game situations.

Use the `nrow` command to check the number of columns in the new data frame. Check: Is it the same as the number of teams in the league for the 2021-2022 season?

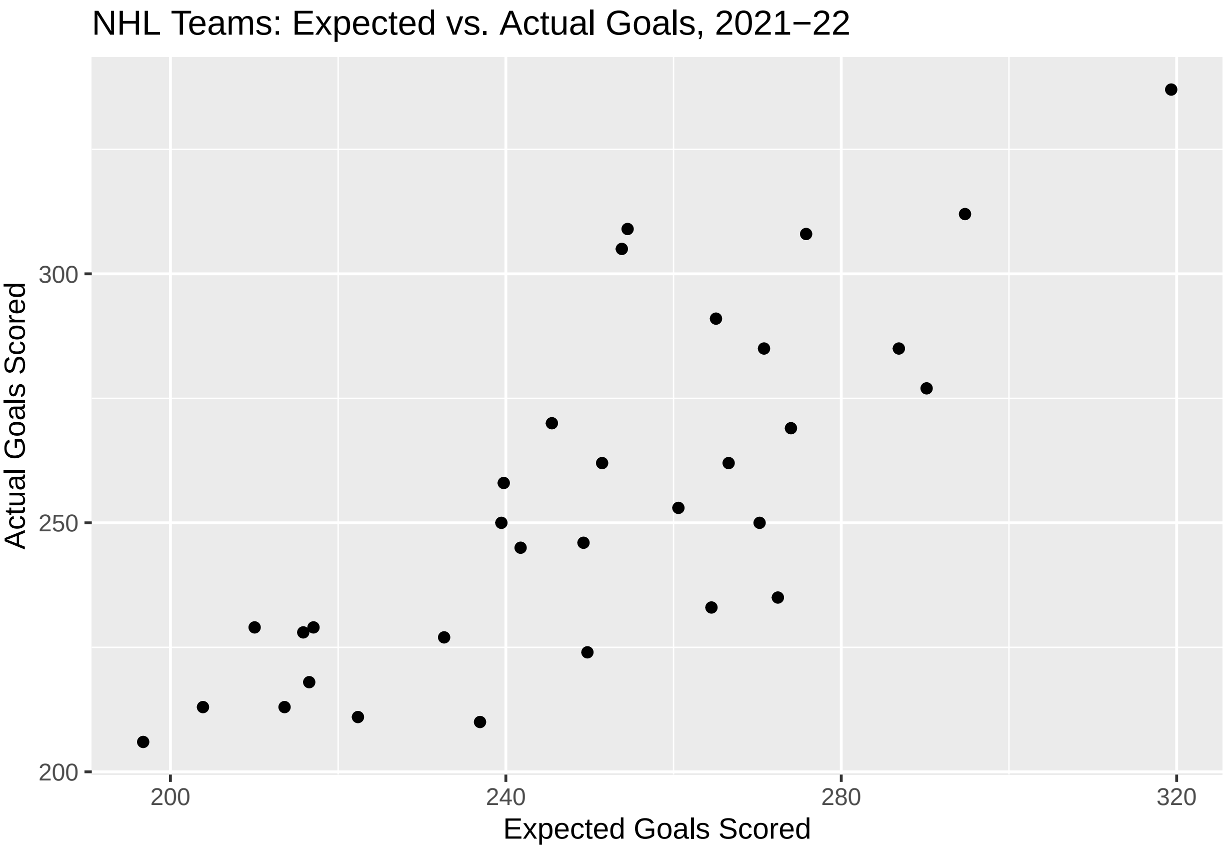
```r
nhl_data_all <- filter(nhl_2022_data, situation == "all")

nrow(nhl_data_all)
```

```
## [1] 32
```

The dataset includes an Expected Goals statistic for each team in the `xGoalsFor` column. Let's plot this quantity against the team's actual number of goals scored; this is given by the `goalsFor` column.

(Remember to always have a good title and axis labels!)

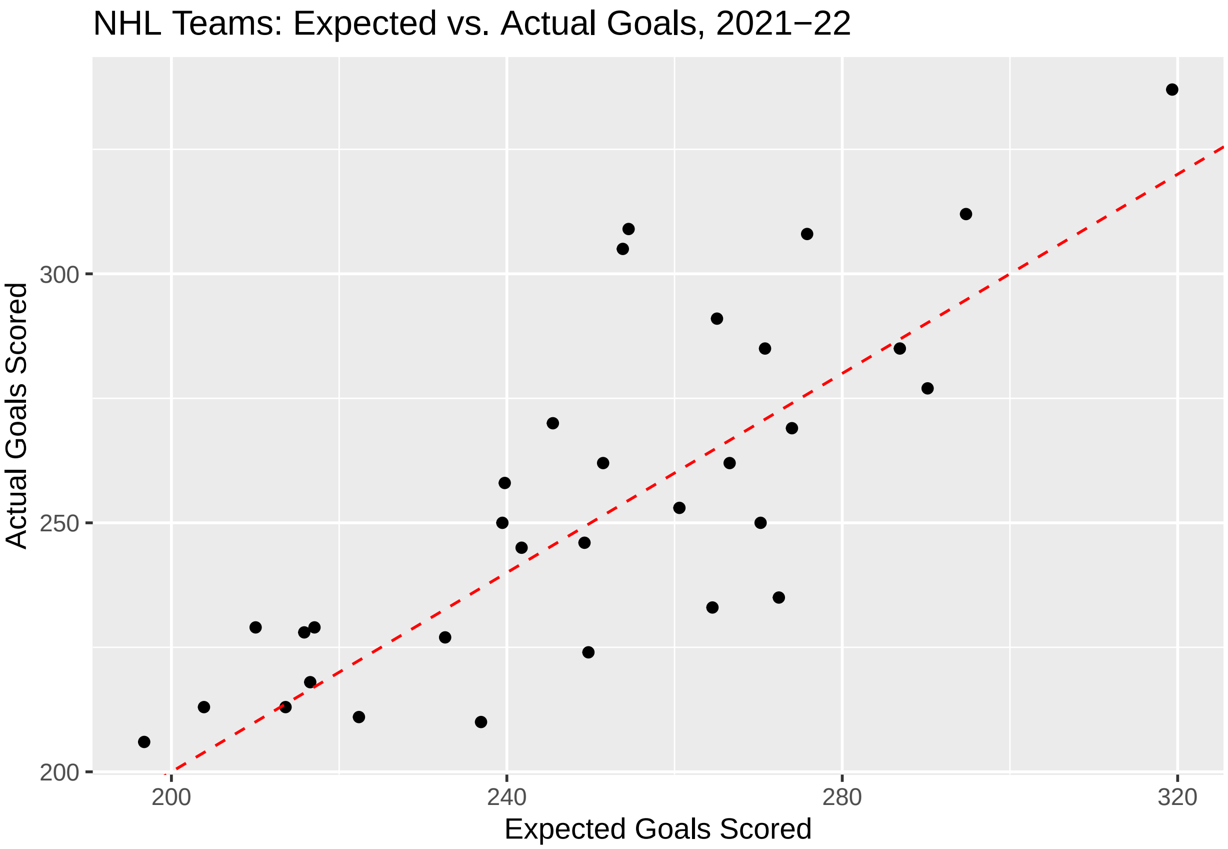
```r
ggplot(data=nhl_data_all, aes(x=xGoalsFor, y=goalsFor)) +
  geom_point() +
  ggtitle("NHL Teams: Expected vs. Actual Goals, 2021-22") +
  xlab("Expected Goals Scored") +
  ylab("Actual Goals Scored")
```



As expected, there is a general positive correlation between expected and actual goals ($r \approx 0.8$). However, there is some variability - for example, the Kings only scored 7 more actual goals than the Ducks, despite having 56.6 more expected goals.

Let's add a line to the graph using the `geom_abline` function corresponding to the line $y = x$, the line on which data points would fall if expected goals were equal to actual goals. We can also customize the line's color and type.

```
ggplot(data=nhl_data_all, aes(x=xGoalsFor, y=goalsFor)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, color="red", linetype="dashed") +
  ggtitle("NHL Teams: Expected vs. Actual Goals, 2021-22") +
  xlab("Expected Goals Scored") +
  ylab("Actual Goals Scored")
```



Note: A slope of 0 and an intercept of 1 are actually the default parameters for the function.

What does it mean for a team's data point to fall below this line? Above it?

Do you think that a team's expected goals would be more likely to be closer to its actual goals for a ten-game stretch, an entire season, or five consecutive seasons? Why?

1.6.4 Goalie Statistics

Example 1.14. For this next example, let's use goalie data from the 2021-2022 season from Natural Stat Trick.

```
goalie_data <- read.csv("data/GoalieTotals_NaturalStatTrick.csv")
goalie_data %>%
  select(2,3,4,5,6,7,8,12) %>%
  arrange(-TOI) %>%
  slice_head(n=10) %>%
  kt()
```

Player	Team	GP	TOI	Shots.Against	Saves	Goals.Against	xG.Against
Juuse Saros	NSH	67	3931.383	2107	1934	173	180.69
Connor Hellebuyck	WPG	66	3903.500	2155	1962	193	199.26
Andrei Vasilevskiy	T.B	63	3760.750	1869	1713	156	165.89
Thatcher Demko	VAN	64	3699.550	1967	1799	168	173.26
Jacob Markstrom	CGY	63	3695.833	1754	1617	137	152.26
Tristan Jarry	PIT	58	3414.717	1711	1573	138	143.92
Elvis Merzlikins	CBJ	59	3320.400	1922	1744	178	164.94
Marc-Andre Fleury	CHI, MIN	56	3284.867	1732	1573	159	148.00
Darcy Kuemper	COL	57	3258.117	1755	1617	138	154.23
John Gibson	ANA	56	3235.583	1789	1617	172	163.53

The dataset includes 119 goalies, but many of them didn't play very much. We can subset the data to include only goaltenders that faced at least 500 shots.

Which player among qualified goalies had the best goals against average? On which team did he play, and what was his GAA?

Which goalie had the most playing time? What was his team, and how much time did he spend on the ice?

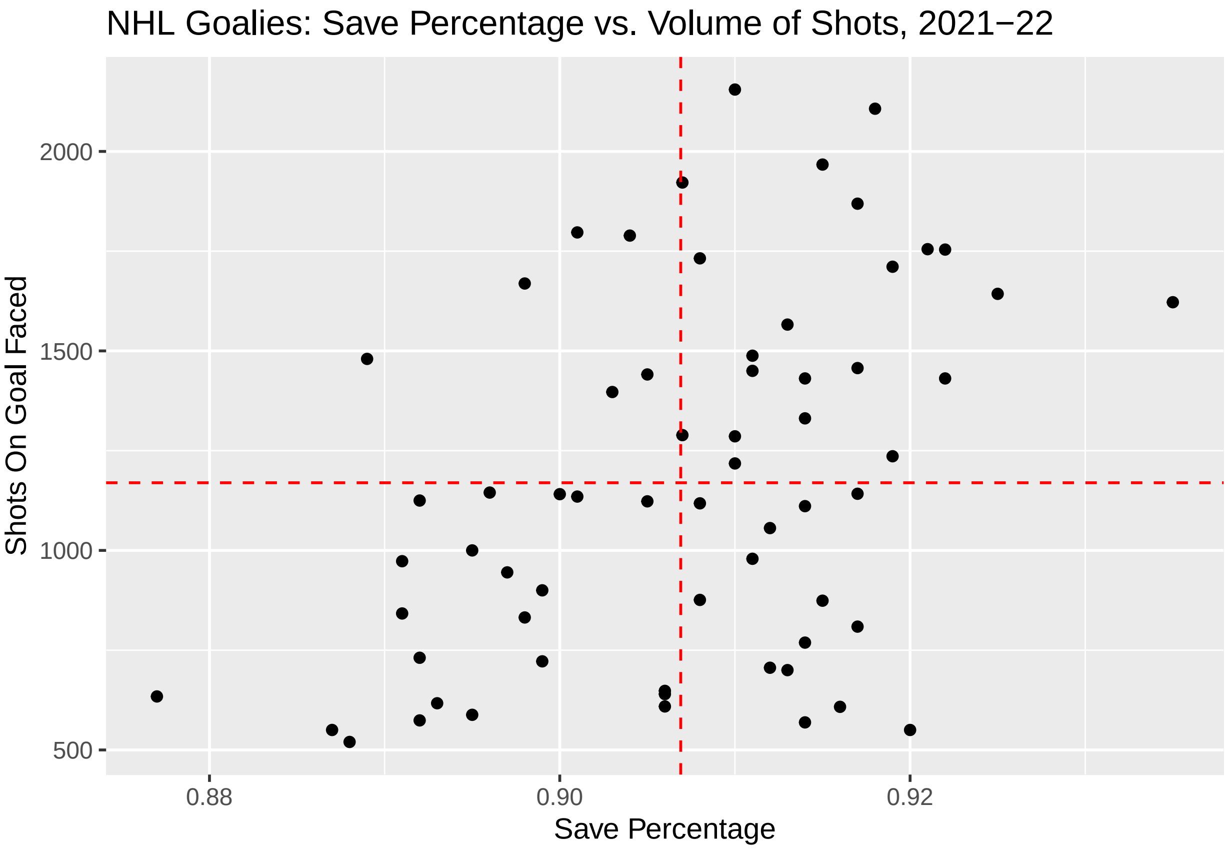
```
filtered_goalie_data <- filter(goalie_data, Shots.Against >= 500)

filtered_goalie_data %>% filter(GAA == min(GAA)) %>% select(Player, Team, GAA)
%>% kt()
```

Player	Team	GAA
Igor Shesterkin	NYR	2.07

```
filtered_goalie_data %>% filter(TOI == max(TOI)) %>% select(Player, Team, TOI)
%>% kt()
```

Player	Team	TOI
Juuse Saros	NSH	3931.383

The following plot compares save percentage to the number of shots on goal faced for the qualified goalies. The dashed horizontal line is placed at the average shots on goal faced among qualified players, and the dashed vertical line is placed at the average save percentage among qualified players.



Which quadrant of the graph represents goalies that faced a higher than average number of shots, but had a below-average save percentage?

Which quadrant represents goalies that had a high save percentage and faced a high volume of shots?

1.6.5 Correlation Plots

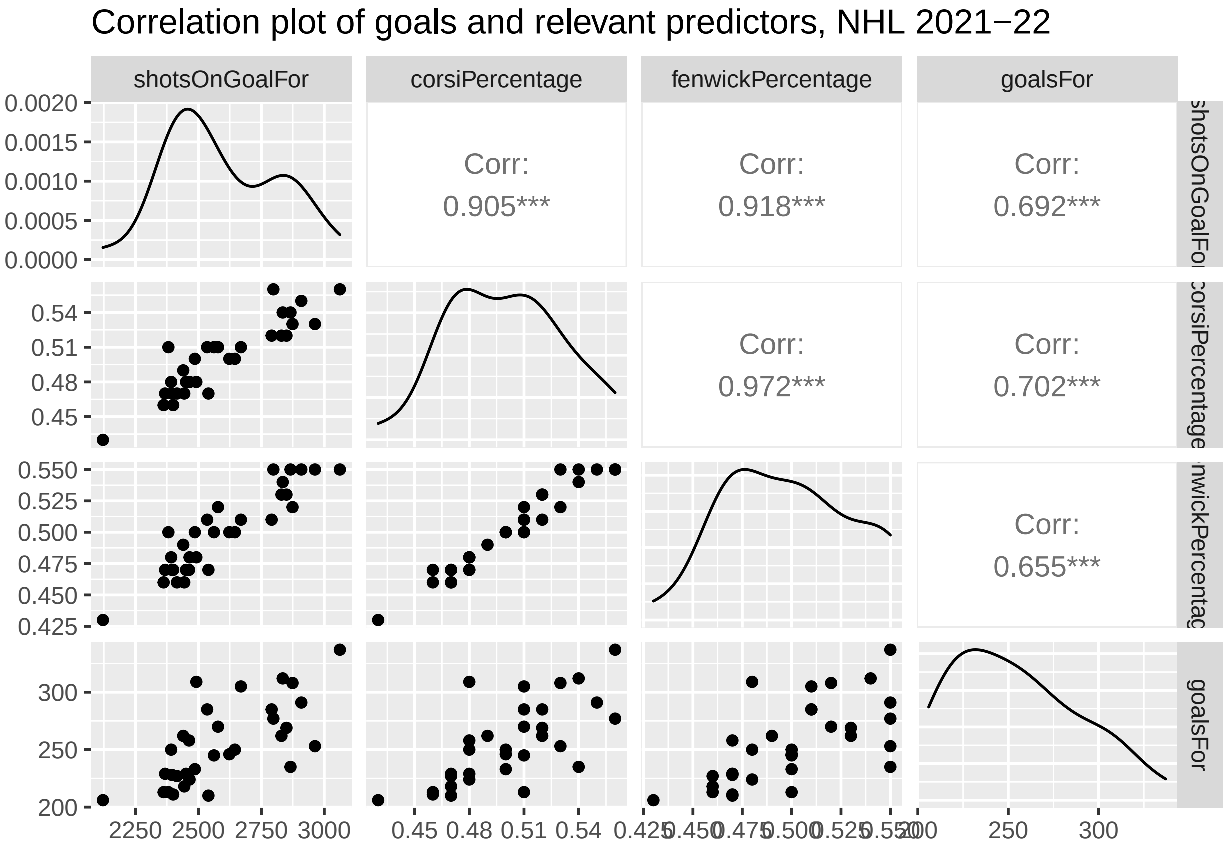
When analyzing sports data, there may be many circumstances where statisticians consider which of several variables are most highly correlated to an outcome variable of interest. In this case, it can be useful to use a correlation plot (also known as a correlation matrix or correlogram). Tidyverse and related packages provide many options for creating correlation plots.

Suppose a statistician has recently learned about some advanced hockey statistics and is interested in researching which stat has the highest correlation with goals scored. The statistician wants to compare team shots on goal, CORSI, and Fenwick to observe the association with goals scored for

NHL teams.

Example 1.15. The following plot uses the same 2021-2022 data from Moneypuck.com; it gives the pairwise scatterplots and correlation values for each of the variables, as well as smoothed plots of each individual variable along the diagonals.

```
goal_stats <- nhl_data_all %>%
  select(shotsOnGoalFor, corsiPercentage, fenwickPercentage, goalsFor)
ggpairs(goal_stats,
        title="Correlation plot of goals and relevant predictors, NHL 2021-22")
```

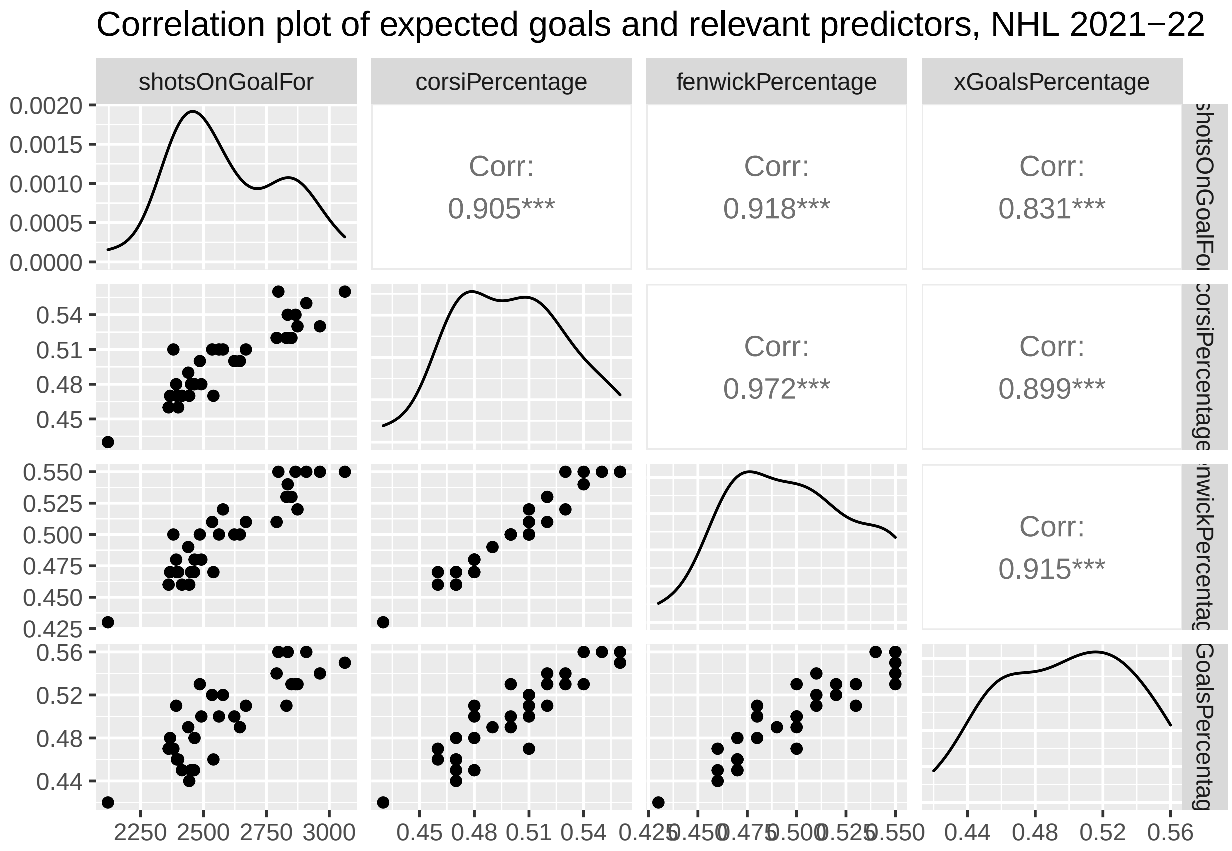


Which of the variables has the strongest correlation with goals scored?

In the article “An advanced stat primer: Understanding basic hockey metrics”, Charlie O’Connor states, “Generally speaking, Corsi is more predictive of future goal differential than Fenwick... however, Fenwick forms the basis for the most widely-used Expected Goals models.” Let’s use the same predictors in a correlation plot with Expected Goals percentage. Does Fenwick have the strongest correlation with xGoal percentage?

```
xGoal_stats <- nhl_data_all %>%
  select(shotsOnGoalFor, corsiPercentage, fenwickPercentage, xGoalsPercentage)
```

```
ggpairs(xGoal_stats,
        title="Correlation plot of expected goals and relevant predictors, NHL
        2021-22")
```



1.7 Volleyball

Volleyball rules Youtube video: https://www.youtube.com/watch?v=9g7nYQv-kPM

1.7.1 Basic Volleyball Statistics

- A **Service Ace (SA)** occurs when a player's serve touches the ground on the other team's side without being touched by a player on that side.
- A **Kill (K)** occurs when a player gets the ball over the net without it being returned by the opponent.
- An **Assist (AST)** is a pass made directly before a player makes a kill.
- **Hitting Percentage (PCT)** is the number of attempted kills (minus errors) divided by the total number of kill attempts. This helps determine how well a player or team is succeeding at their kill attempts.
- A **Dig** is a pass of a hard-driven ball from the other team.

Reference:
www.rookieroad.com

For Volleyball EDA, we will be using CSU Women's Volleyball data from the last five seasons.

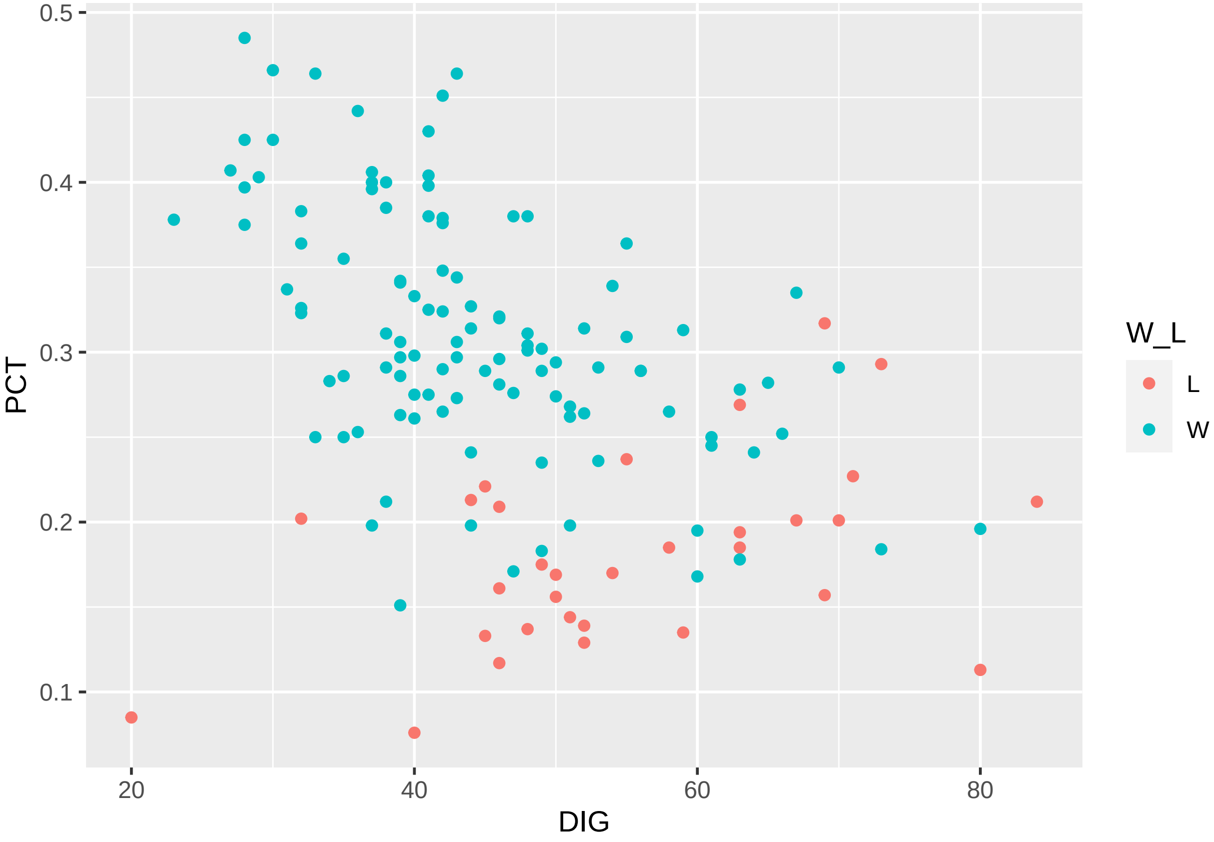
```
# Load CSU Women's Volleyball Data
csu_vb <- read_csv("data/csu_volleyball.csv")
colnames(csu_vb)[3] <- "W_L"
csu_vb %>% slice_head(n=10) %>% select(1:13) %>% kt()
```

Date	Opponent	W_L	SP	K	E	TA	PCT	AST	SA	SE	RE	DIG
8/25/17	Duke	L	5	66	28	179	0.212	64	5	13	6	84
8/26/17	Central Florida	W	4	56	18	126	0.302	52	7	10	7	49
8/29/17	Northern Colorado	W	3	39	8	77	0.403	38	5	12	4	29
9/1/17	vs TCU	W	5	62	20	149	0.282	59	6	10	7	65
9/1/17	vs UNC Asheville	W	3	41	7	80	0.425	39	8	8	5	28
9/2/17	at Florida State	W	3	48	12	95	0.379	45	6	4	1	42
9/8/17	Ball State	W	4	59	24	145	0.241	56	6	8	3	44
9/8/17	Michigan	W	3	48	8	101	0.396	46	3	6	4	37
9/10/17	Idaho State	W	3	46	11	92	0.380	46	4	3	4	48
9/15/17	UAlbany	W	3	41	7	73	0.466	36	5	5	1	30

Let's look at a scatter plot of hitting percentage and the number of digs. While no conclusions can be drawn from such a plot, it can give us some insight into relationships worthy of further analysis. Before creating the plot using the code below, think about what you might expect the outcome to be.

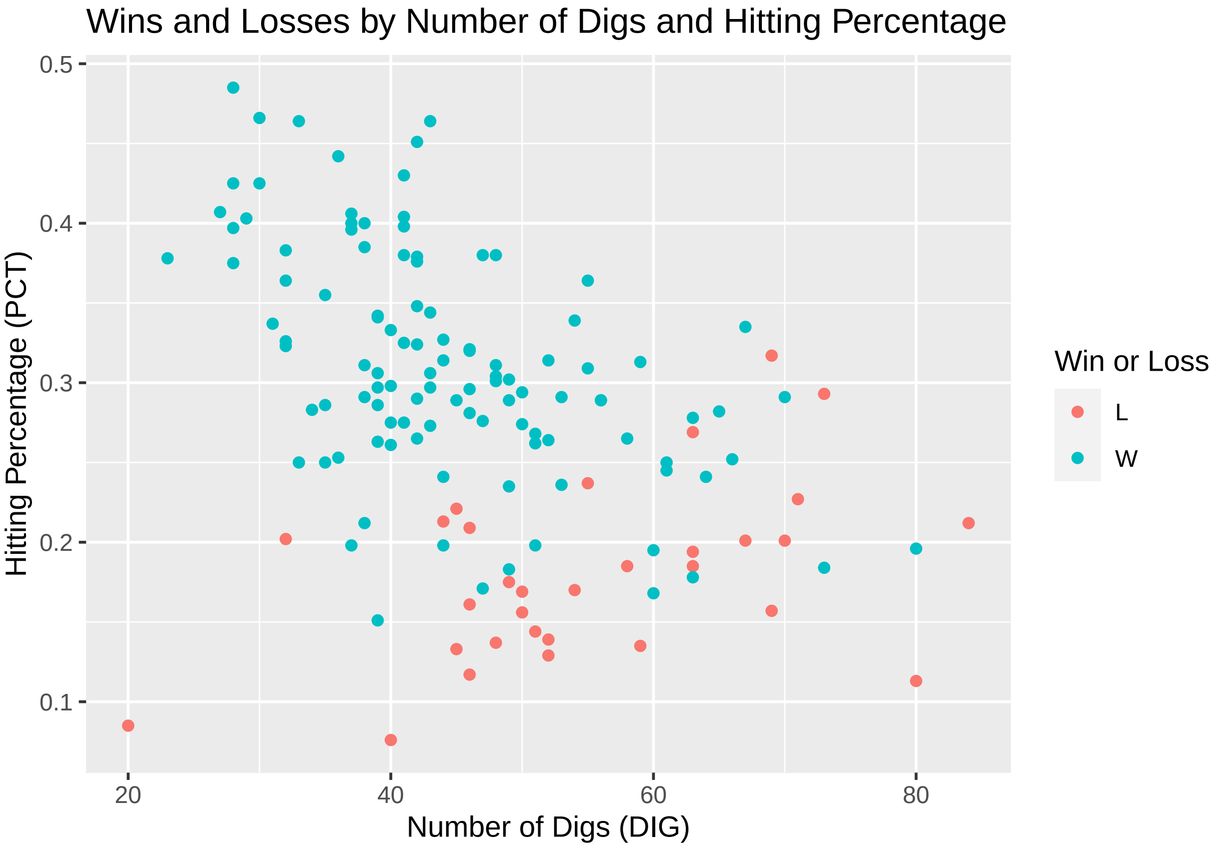
1.7.2 Scatter Plot

```
# Digs, Hitting Percentage, Win/Lose
dig_pct_viz <- ggplot(data = csu_vb, aes(x = DIG, y = PCT, color = W_L)) +
  geom_point()
dig_pct_viz
```



Let’s change the axis titles, legend title, and add a main title.

```
dig_pct_viz +
  labs(title = "Wins and Losses by Number of Digs and Hitting Percentage",
       x = "Number of Digs (DIG)", y = "Hitting Percentage (PCT)",
       color = "Win or Loss")
```

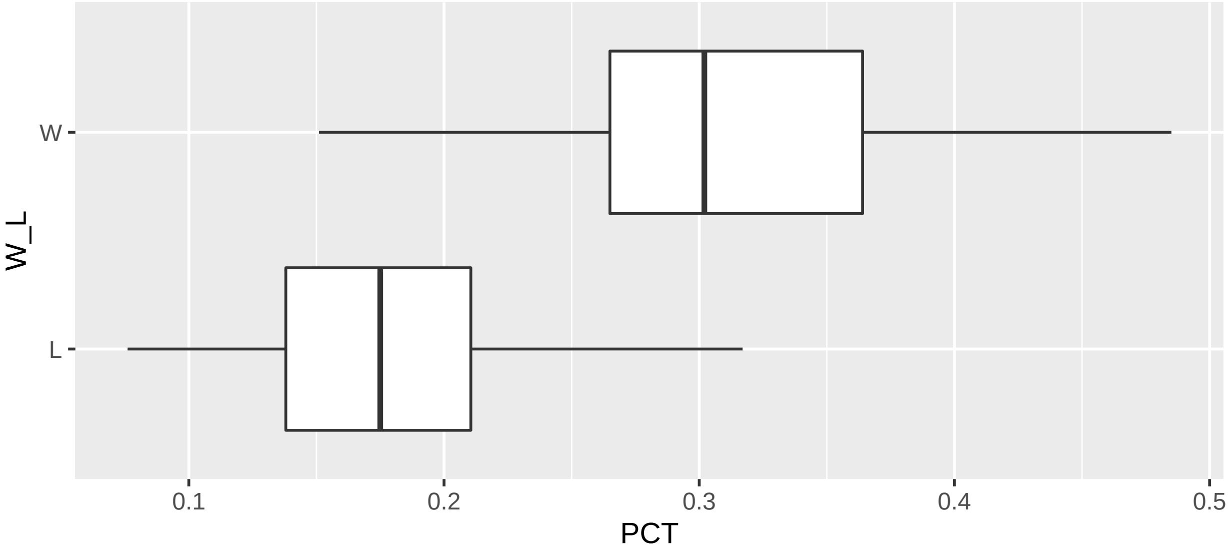


What can we learn from this visual? Well, we can see that there is a weak linear relationship between the number of digs and hitting percentage. To an extent, hitting percentage decreases as the number of digs increases. Why is this the case? Maybe if a team has a really high hitting percentage, this means that the opposing team does not have as many opportunities to attack the other team offensively, reducing the number of opportunities for digs. It also seems that while wins and losses are somewhat evenly spread across the number of digs, there is a more clear cutoff for hitting percentage. It seems that the majority of wins are associated with a hitting percentage of at least 0.2, while the majority of losses are associated with a hitting percentage of less than 0.3.

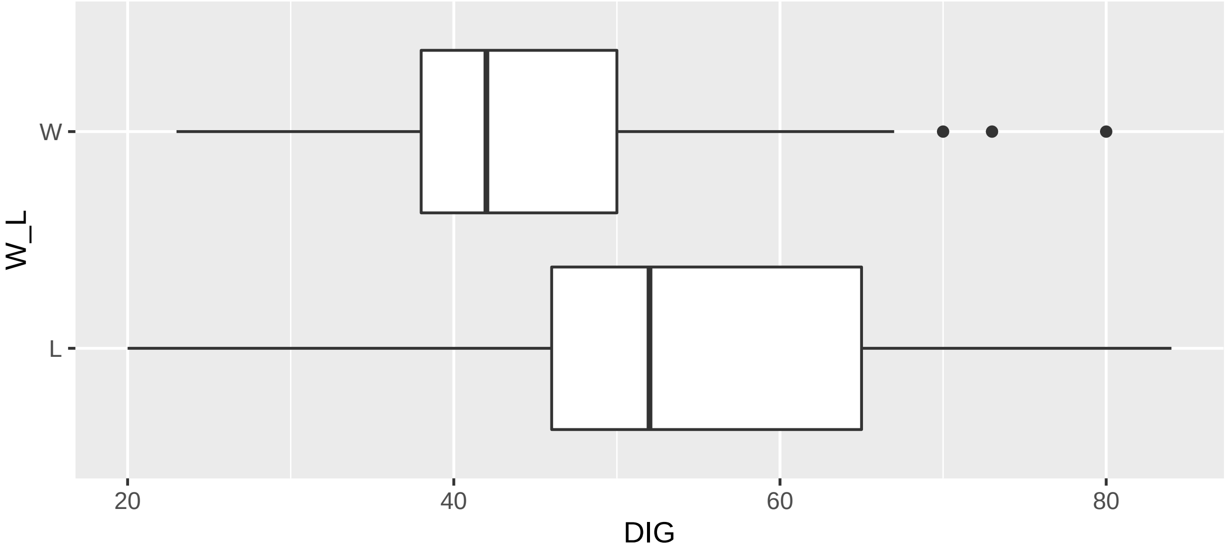
1.7.3 Box Plot

Now let's take a closer look at the distribution of hitting percentage and digs for wins and losses. To do this, we will create box plots for each statistic.

```
pct_viz <- ggplot(data = csu_vb, aes(x = PCT, y = W_L)) +
  geom_boxplot()
pct_viz
```

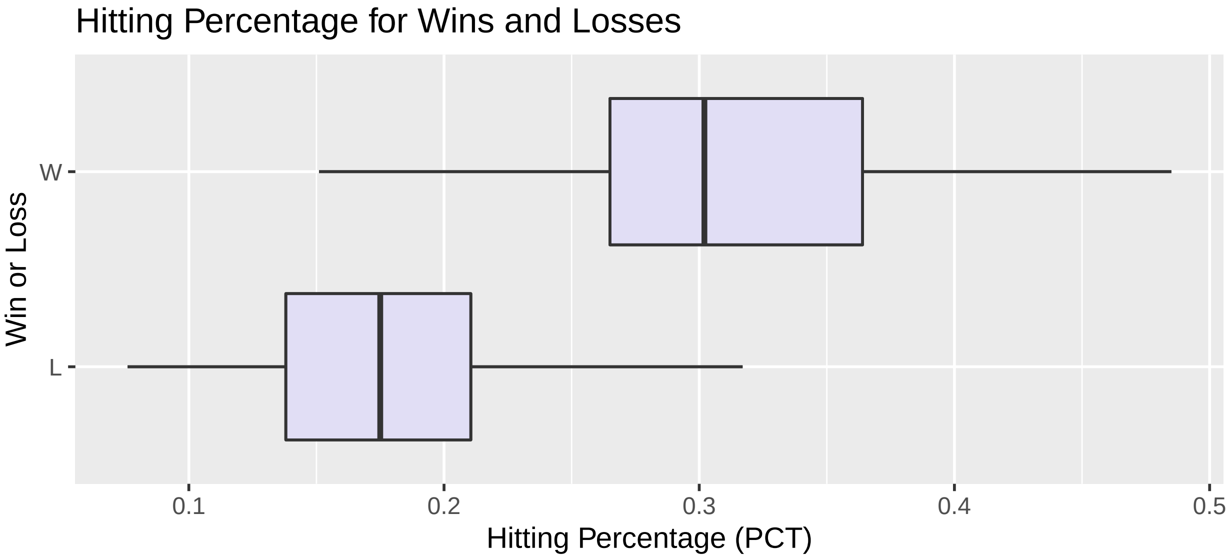


```
dig_viz <- ggplot(data = csu_vb, aes(x = DIG, y = W_L)) +
  geom_boxplot()
dig_viz
```

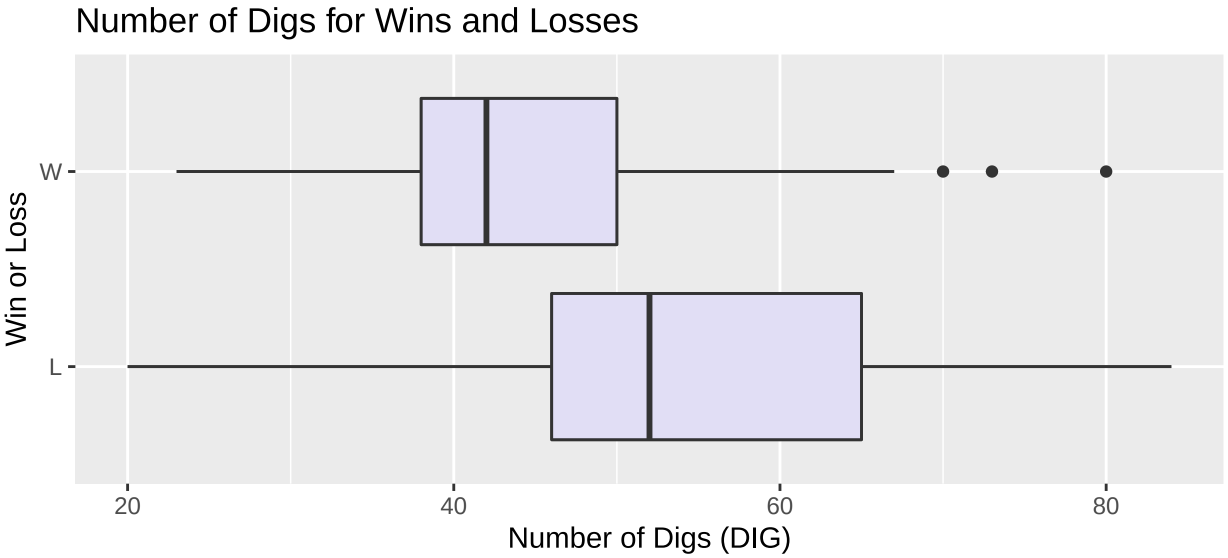


Let's modify these plots to make them more complete and visually appealing.

```
pct_viz +
  labs(title = "Hitting Percentage for Wins and Losses",
       x = "Hitting Percentage (PCT)",
       y = "Win or Loss") +
  geom_boxplot(fill = "slateblue", alpha = 0.2)
```



```
dig_viz +
  labs(title = "Number of Digs for Wins and Losses",
       x = "Number of Digs (DIG)",
       y = "Win or Loss") +
  geom_boxplot(fill = "slateblue", alpha = 0.2)
```



Box plots allow us to isolate each statistic (number of kills and hitting percentage) so we can more clearly determine the center and spread of each between wins and losses.

1.8 Soccer

Rules of Soccer YouTube video: https://www.youtube.com/watch?v=dFLaabgXhpc

1.8.1 Basic Soccer Statistics

- **Shots (SH)** represent all shots taken by a team throughout the game. This is simply an attempt by a player to shoot the ball toward the net, even if they miss or the shot is saved (Rookie Road).
- **Shots on Goal (SOG)** represent all shots that would have gone into the goal if not saved by a defender or goalkeeper (Rookie Road).
- **Assist (A)** occur when a player passes the ball to someone, and the next shot results in a goal.
- **Possession** refers to the percentage of time a team had control of the ball during a game.

1.8.2 Advanced Soccer Statistics

- **Expected Goals (xG)** "indicates how many goals a team could have expected to score based on the quantity and quality of chances that they created in a match" (Tippett 2019, 4).

These definitions come from www.rookieroad.com and "The Expected Goals Philosophy" by James Tippett.

To learn more about expected goals, check out this YouTube video:
https://www.youtube.com/watch?v=w7zPZsLGK18

1.8.3 Bar Plot

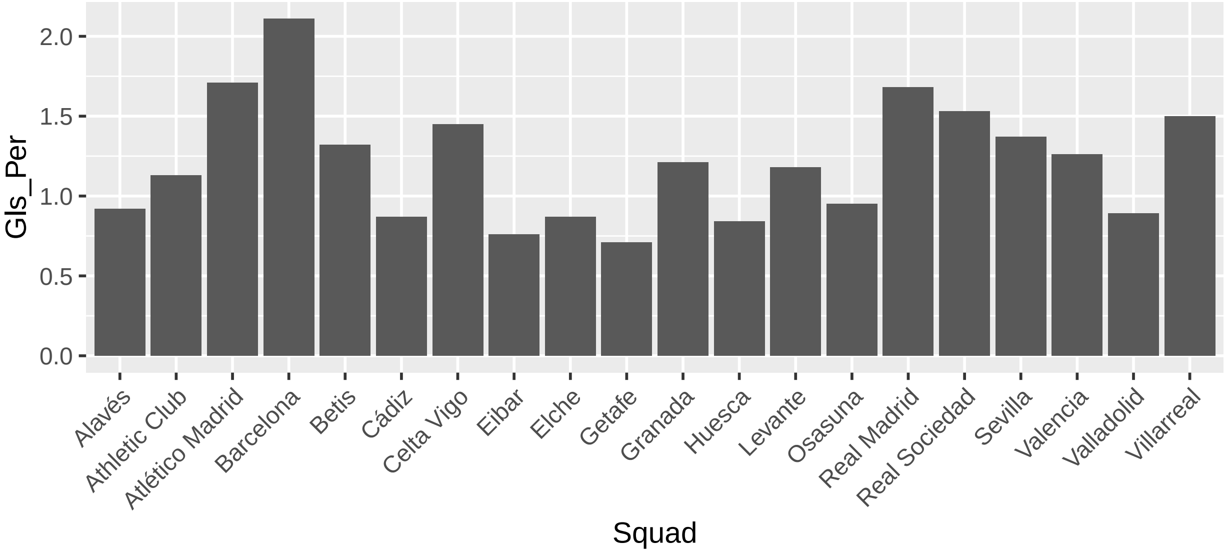
Now that we have an understanding of some basic shooting statistics, let us go through some EDA examples. For this first example, we will need to install the "worldfootballR" package.

```
library(worldfootballR)
```

Next we will look at some data specific to La Liga, which is a soccer league in the men's top professional soccer division.

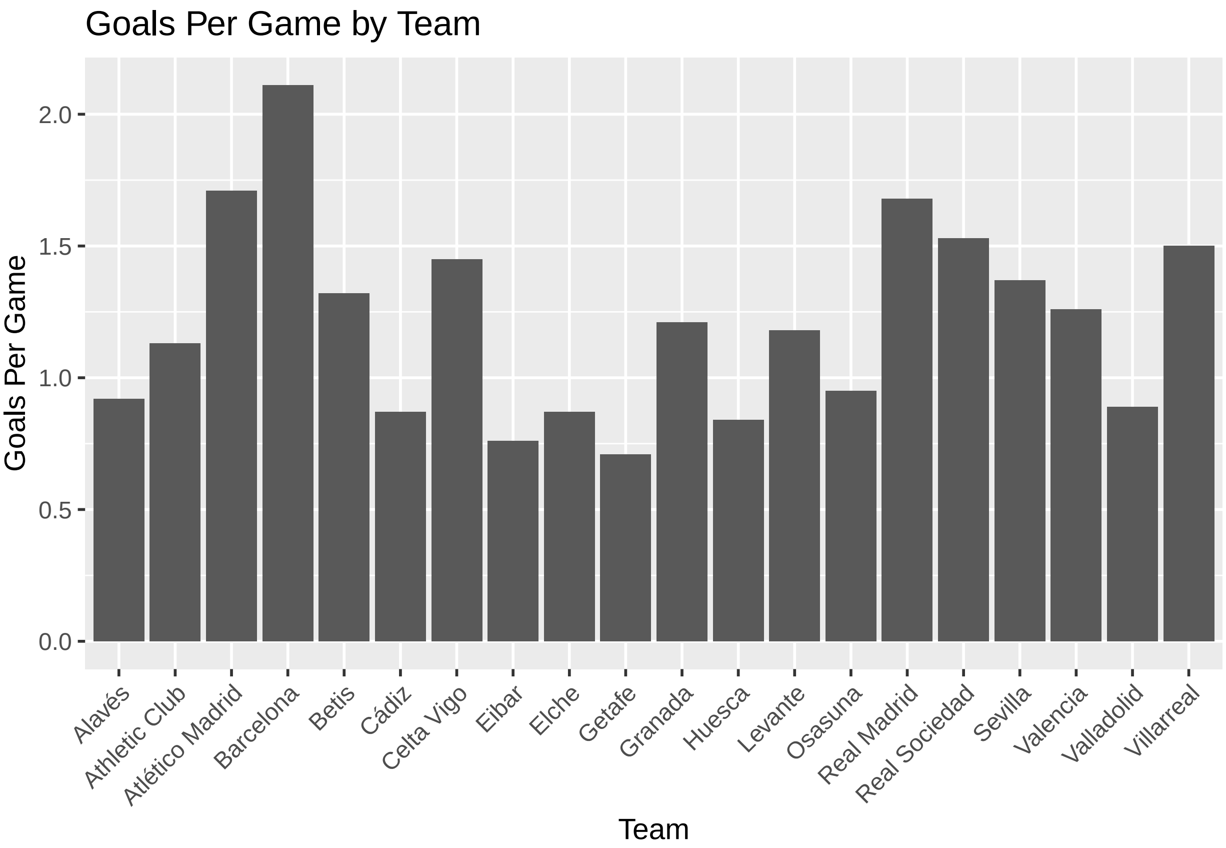
```
# Get "Squad Standard Stats" Data
big5_2021_stats <- fb_big5_advanced_season_stats(
  season_end_year = 2021, stat_type = "standard", team_or_player = "team")
liga_2021_stats <- big5_2021_stats[which((big5_2021_stats$Comp == "La Liga")),]

# look at the first ten entries and a selection of columns
liga_2021_stats %>%
  select(Squad,Team_or_Opponent,Poss,Gls,Ast,xG_Expected,xA_Expected) %>%
  slice_head(n=10) %>% kt()
```

```
# Create visual for each team's goals per game
team_goals_viz <-
  ggplot(data = liga_2021_stats[which(liga_2021_stats$Team_or_Opponent ==
  "team"),],
                       aes(x = Squad, y = Gls_Per)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_bar(stat = "identity")
team_goals_viz
```



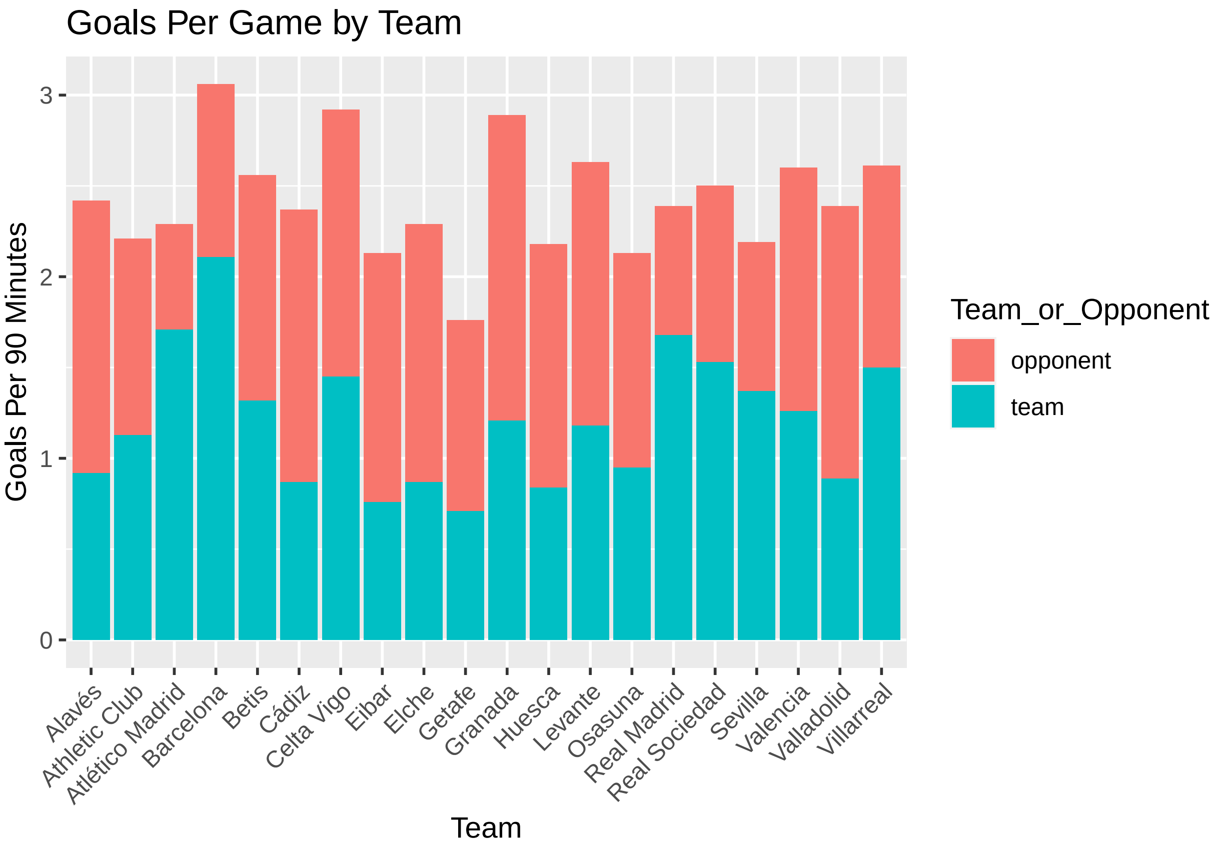
This plot is a good starting point, but still looks pretty messy. Let's add a title, change the axis titles, and rotate the axis labels so they are not overlapping over one another.

```
team_goals_viz <- team_goals_viz +
  xlab("Team") +
  ylab("Goals Per Game") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Goals Per Game by Team")
team_goals_viz
```



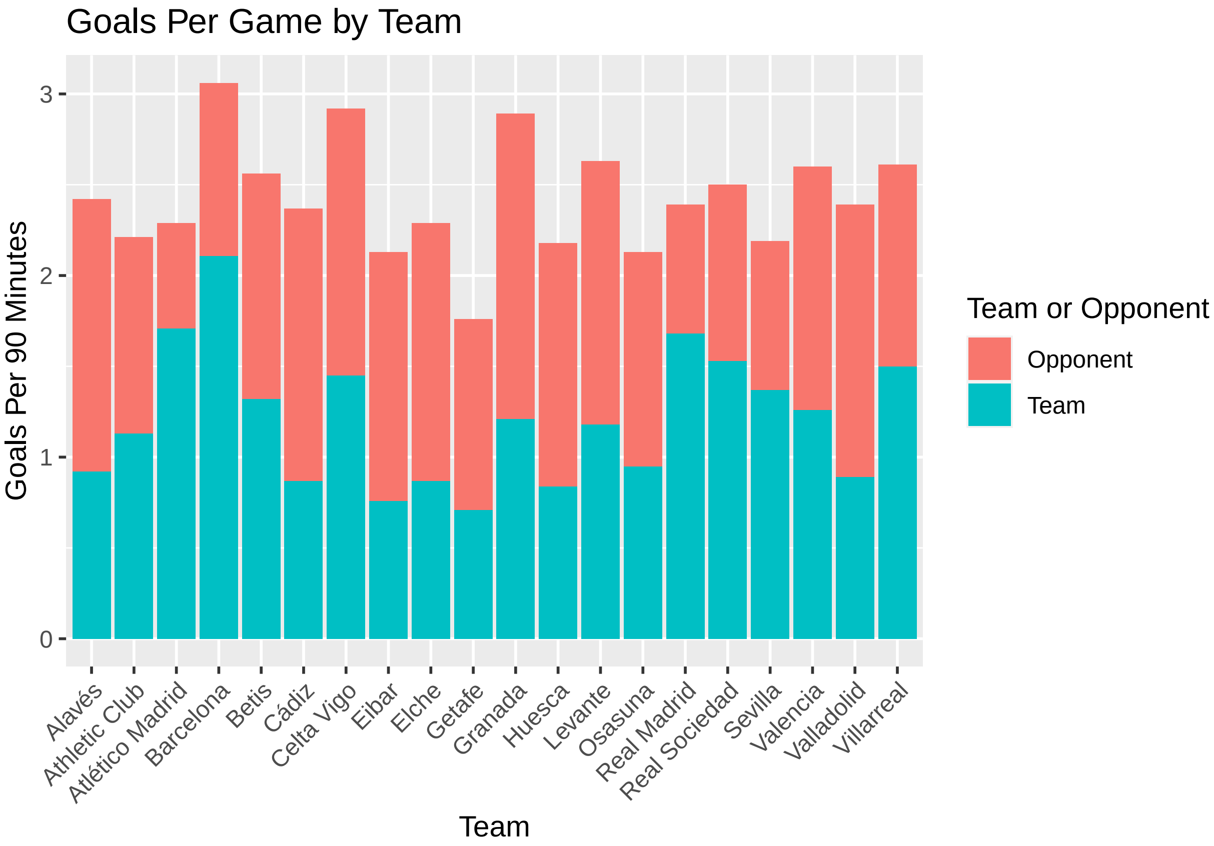
This is already looking a lot better. Now, we will add the goals scored per game *against* each team. Why is this of interest? Well, at first glance, Barcelona seems like a pretty impressive team, as they score more goals per game than any other team in the league. However, what if they also have more goals scored against them than any other team in the league? This could be important context, so we will include it in the graph below.

```
all_goals_viz <- ggplot(data = liga_2021_stats, aes(x = Squad, y = Gls_Per)) +
  geom_bar(stat = "identity", aes(fill = Team_or_Opponent), position = "stack") +
  xlab("Team") +
  ylab("Goals Per 90 Minutes") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Goals Per Game by Team")
all_goals_viz
```



This is looking pretty good, but let's clean it up just a bit by changing the legend title and labels.

```
all_goals_viz +
  scale_fill_discrete(name = "Team or Opponent", labels = c("Opponent","Team"))
```

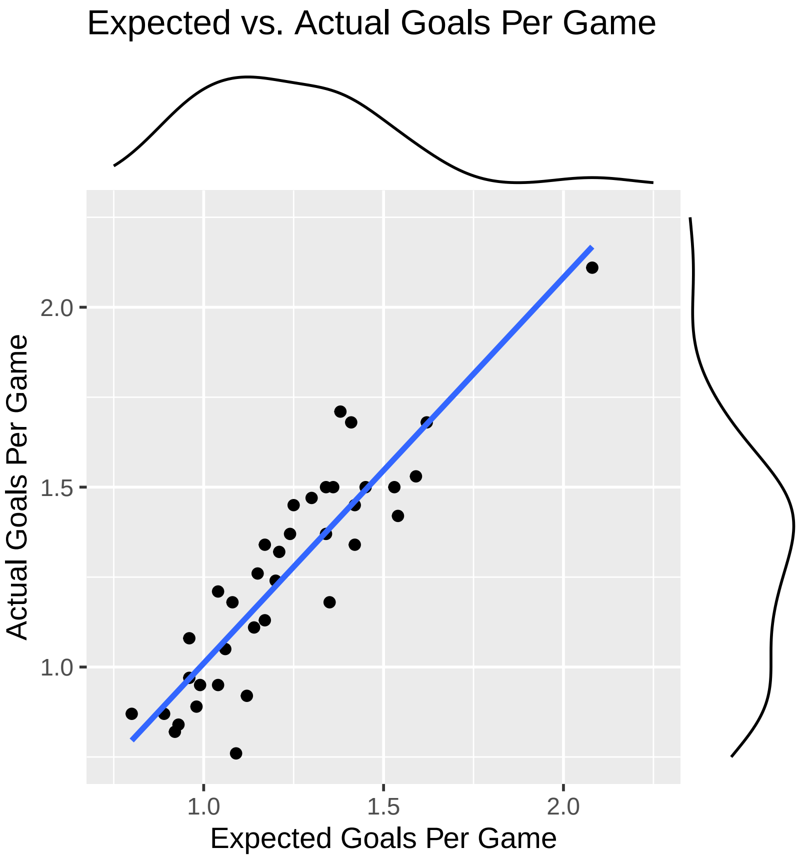


What does this graph show us? Well, we are able to see the average number of goals scored for and against each team per game. It looks like Barcelona is scoring a lot more goals than they are letting be scored against them, while other teams like Valladolid tend to have a higher proportion of goals scored for the opposing team.

1.8.4 Scatter Plot

In addition to simply knowing the average actual number of goals scored for and against each team per game, we may be interested in how this compares to the expected number of goals scored per game, as well.

```
library(ggExtra)
act_exp_viz <- ggplot(data = liga_2021_stats,
                      aes(x = xG_Per, y = Gls_Per, label = Squad)) +
  geom_point() +
  scale_x_continuous(limits = c(0.75,2.25)) +
  scale_y_continuous(limits = c(0.75,2.25)) +
  ggtitle("Expected vs. Actual Goals Per Game") +
  xlab("Expected Goals Per Game") +
  ylab("Actual Goals Per Game") +
  geom_smooth(method = "lm", se = FALSE) +
  theme(aspect.ratio = 2/2)
ggMarginal(act_exp_viz, type = "density")
```



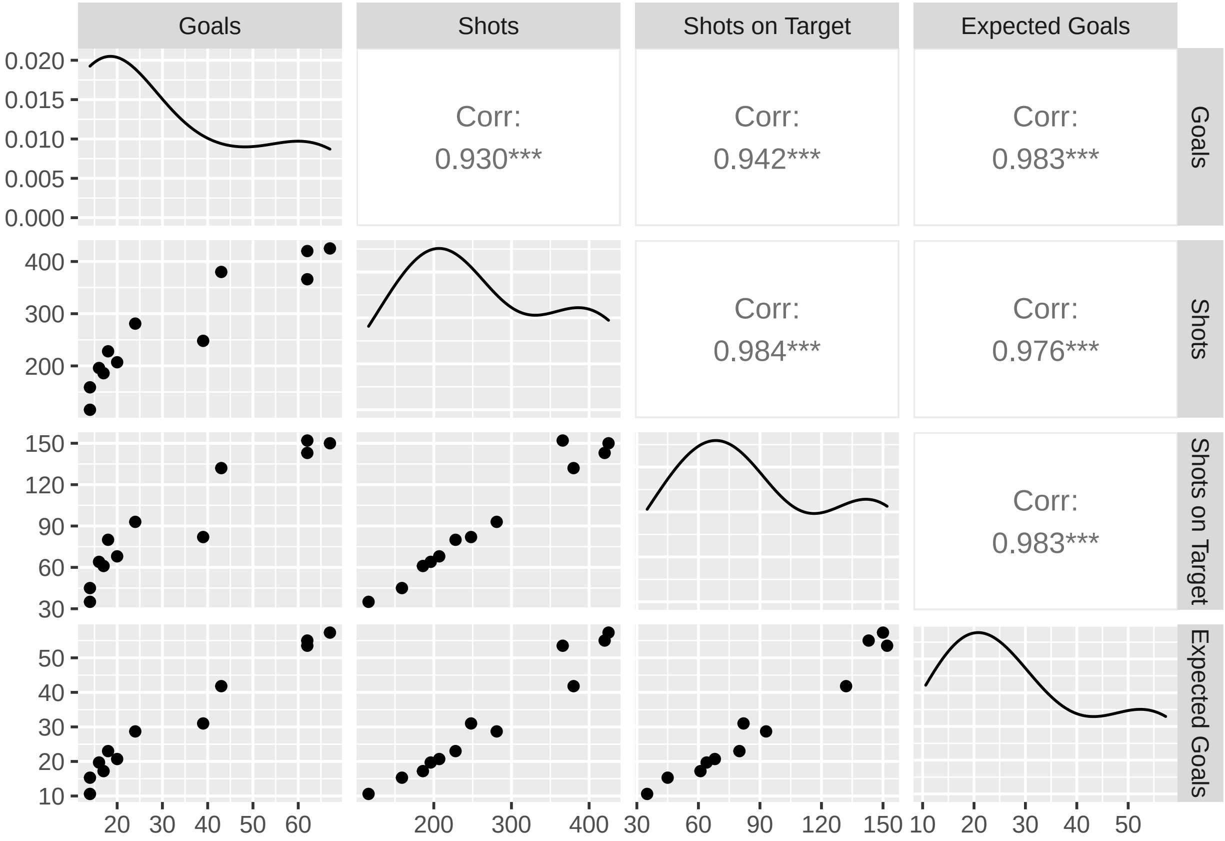
As you can see, we fit a line to the data. At first glance, it seems to have a positive slope slightly greater than 1. What does this mean in the scenario of actual and expected goals per game?

Example 1.16. Use the `worldfootballR` package to collect team shooting data from the Women's 1st Tier League (Women's Super League) in England for 2020-2021. Plot goals vs. shots, shots on target, and expected goals. Comment on the relationships.

```r
wsl_2021_shooting <- get_season_team_stats(
  country = "ENG", gender = "F", season_end_year = "2021",
  tier = "1st", stat_type = "shooting")
wsl_2021_shooting <-wsl_2021_shooting %>%
  select(Squad,Gls_Standard,Sh_Standard,SoT_Standard,xG_Expected) %>%
  rename(Goals=Gls_Standard,Shots=Sh_Standard,
         `Shots on Target`=SoT_Standard,`Expected Goals` = xG_Expected)
wsl_2021_shooting %>% kt()
```

Squad	Goals	Shots	Shots on Target	Expected Goals
Arsenal	62	366	152	53.5
Aston Villa	14	159	45	15.3
Birmingham City	14	116	35	10.6
Brighton	20	207	68	20.7
Bristol City	17	186	61	17.2
Chelsea	67	425	150	57.3
Everton	39	248	82	31.0
Manchester City	62	420	143	55.0
Manchester Utd	43	380	132	41.8
Reading	24	281	93	28.7
Tottenham	16	196	64	19.7
West Ham	18	228	80	23.0
vs Arsenal	14	169	51	15.4
vs Aston Villa	45	355	121	39.4
vs Birmingham City	43	370	126	40.0
vs Brighton	41	316	93	39.0
vs Bristol City	68	426	165	51.5
vs Chelsea	9	143	35	13.0
vs Everton	29	243	86	30.6
vs Manchester City	13	122	40	12.9
vs Manchester Utd	17	202	67	20.0
vs Reading	41	296	103	39.0
vs Tottenham	39	276	102	35.2
vs West Ham	37	294	116	37.9

```
wsl_2021_shooting <- wsl_2021_shooting %>%
  slice_head(n=12)
wsl_2021_shooting %>%
  select(Goals,Shots,`Shots on Target`,`Expected Goals`) %>%
  ggpairs()
```



Chapter 2

Probability

2.1 Definitions

Definition 2.1. An ***experiment*** is any activity or process whose outcome is subject to uncertainty.

Definition 2.2. The ***sample space*** of an experiment, denoted by Ω or $\mathcal{S}$, is the set of all possible outcomes of that experiment.

Definition 2.3. An ***event*** is any collection (subset) of outcomes contained in the sample space, Ω.

Example 2.1. Give some examples of discrete random variables in sports.

Example 2.2. Give some examples of continuous random variables in sports.

2.2 Set Theory

For the following examples, suppose that we are interested in the batting outcomes of a plate appearance in baseball.

Let A be the event that the batter gets walked, let B be the event that the batter gets a hit, let C be the event that the batter strikes out, and let D be the event that the batter makes it to first base at the end of their at bat.

We will define a handful of set operations to help us when we begin calculating the probability of different events occurring.

Definition 2.4. The ***compliment*** of an event A, denoted by A^c or A', is the set of all outcomes in Ω that are not contained in A.

Example 2.3. Draw a Venn diagram illustrating A^c and describe the event.

Definition 2.5. The ***union*** of two events A and B, denoted by $A \cup B$ and read "A or B", is the event consisting of all outcomes that are either in A or B or in both.

Example 2.4. Draw a Venn diagram illustrating $A \cup D$ and describe the event.

Definition 2.6. The ***intersection*** of two events A and B, denoted by $A \cap B$ and read "A and B", is the event consisting of all outcomes that are in both A and B.

Example 2.5. Draw a Venn diagram illustrating $A \cap D$ and describe the event.

Definition 2.7. The ***difference*** of two events A and B, denoted by $A \,/\, B$ and read "difference of A and B", is the event consisting of all outcomes that are in A but not in B.

Example 2.6. Draw a Venn diagram illustrating $D \,/\, A$ and describe the event.

Definition 2.8. Two events A and B are said to be ***disjoint*** (or ***mutually exclusive***) if $A \cap B = \emptyset$

Example 2.7. Are the events A and B disjoint? How about A and D?

2.3 Axioms, Properties, and Laws

There are some basic assumptions or "axioms" which are the foundation of the theory of probability. Andrey Kolmogorov first described these axioms in 1933.

2.3.1 Axioms of Probability

1. $P(A) \geq 0$, for any event A

2. $P(\Omega) = 1$

3. If $A_1, A_2, A_3, ...$ is a collection of disjoint events, then:
   $P(\cup_{i=1}^{\infty} A_i) = P(A_1 \cup A_2 \cup ...) = \sum_{i=1}^{\infty} P(A_i)$

Note that all probabilities are between 0 and 1, that is, for any event A, $0 \leq P(A) \leq 1$.

We can convert to percentages by multiplying probabilities by 100, however, this is a set that is only done after all calculations have been completed.

2.3.2 Properties of Probability

- $P(\emptyset) = 0$

- $P(A^c) = 1 - P(A)$

- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

- $P(A \cup B \cup C) =$
  $P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$

- $P([A \cup B]^c) = P(A^c \cap B^c)$

- $P([A \cap B]^c) = P(A^c \cup B^c)$

Example 2.8. In 2001, Barry Bonds broke the single season home run record with 73 home runs. In this season, he had 664 plate appearances (476 at-bats), 156 hits, 177 walks, 9 hit by pitches, and 2 sacrifice flies. Use this information to answer the following questions.

(a) Plate appearances that result in a walk, hit by pitch, or sacrifice fly are not counted towards a player's at-bats. Confirm that Bonds had 476 official at-bats.

(b) Suppose an at-bat is chosen at random. What is the probability that Bonds got a hit? (This is his batting average.)

For the following examples, assume that one of Bonds' plate appearances is chosen at random.

(c) What is the probability that Bonds reached base via a hit, walk, or hit by pitch? (This is his on-base average/percentage.)

(d) What is the probability that Bonds did not reach base?

(e) Calculate $P\left(HBP \cup Walk\right)$

(f) Calculate $P\left(HBP^c \cap Walk^c\right)$

2.3.3 Laws of Probability

Definition 2.9. Let A and B be two events such that $P(B) > 0$. Then the ***conditional probability*** of A given B, written $P(A|B)$, is given by: $P(A|B) = \frac{P(A \cap B)}{P(B)}$

Example 2.9. In 1998, Sammy Sosa hit 66 home runs in 722 plate appearances, the third highest single season homework total ever. During June 1998, Sosa 20 home runs in 121 plate appearances. Suppose a randomly selected plate appearance is selected. Calculate the following probabilities.

(a) $P(HR)$

(b) $P(HR \cap June)$

(c) $P(June)$

(d) $P(HR|June)$

(e) $P(June|HR)$

(f) $P(HR|June^c)$

Theorem 2.1 (Multiplication Rule)**.** *For any two events A and B, $P(A \cap B) = P(B|A) \cdot P(A)$.*

Example 2.10. Calculate the probability that a randomly selected plate appearance from Sosa's 1998 season is a home run in June.

Definition 2.10. Events $A_1, A_2, \dots, A_n$ are said to form a ***partition*** of a sample space Ω if both:
(i) $A_i \cap A_j = \emptyset$ $(i \neq j)$
(ii) $\cup_{i=1}^{n} A_i = \Omega$

Theorem 2.2 (Law of Total Probability)**.** *Suppose events $A_1, A_2, \dots, A_n$ form a partition of Ω, then: $P(B) = P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots P(B|A_n)P(A_n)$*

Example 2.11. What is one possible way to partition Sosa's plate appearances in 1998?

Theorem 2.3 (Bayes Theorem: simple version)**.** *Suppose events B and C form a partition of Ω, then: $P(B|A) = \frac{P(B \cap A)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B)+P(A|C)P(C)}$*

Theorem 2.4 (Bayes Theorem)**.** *Suppose events $B_1, B_2, \dots, B_n$ form a partition of Ω, then: $P(B_k|A) = \frac{P(B_k \cap A)}{P(A)} = \frac{P(A|B_k)P(B_k)}{P(A|B_1)P(B_1)+P(A|B_2)P(B_2)+\dots+P(A|B_n)P(B_n)}$*

Example 2.12. Given that Sosa hit a home run in 1998, what is the probability that he hit it in June?

Example 2.13. Over the course of a season, a hockey player scored a goal 30% of the time during a home game, and 18%. Assume all games are either home or away. Use this information to answer the following questions.

(a) What is the probability the player scored a goal in any game if there were an equal number of home and away games?

(b) What is the probability the player scored a goal in any game if there were twice as many home games as away games?

(c) What is the probability the player scored a goal in any game if the ratio of home games to away games is 2:3?

2.4 Combinatorics

Combinatorics is the mathematical study of counting, particularly with respect to permutations and combinations.

Definition 2.11. The ***factorial function*** ***($n!$)*** is defined for all positive integers by: $n! = n \cdot (n - 1) \cdot ... 2 \cdot 1$

Note that $0! \equiv 1$ and $1! \equiv 1$.

Example 2.14. A baseball/softball batting lineup has nine ordered players. Suppose the manager has selected the nine players to bat. How many different batting orders are there possible?

Definition 2.12. An ordered subset is called a ***permutation***. The number of permutations of size k that can be formed from the n elements in a set is given by: $P_{n,k} = \frac{n!}{(n-k)!}$

Example 2.15. In MLB, each team has a 26-person roster, of which about 13 are hitters. Assuming one of the batters is the designated hitter, how many different batting lineups of 9 players can a manager create?

Definition 2.13. An unordered subset is called a ***combination***. The number of combinations of size k that can be formed from the n elements in a set is given by: $C_{n,k} = \binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$

Example 2.16. Suppose a manager has 13 hitters to choose from to fill out a starting lineup of 9 players. Ignoring positions, how many different ways can the manager pick a 9-person lineup from 13 possible hitters?

Theorem 2.5 (Product Rule for Ordered Pairs)**.** *If the first element of an ordered pair can be selected in n_1 ways and for each of the these n_1 ways the second element of the pair can be selected in n_2 ways, then the number of pairs is $n_1 \cdot n_2$.*

Theorem 2.6 (Generalized Product Rule)**.** *Suppose a set consists of k elements (k-tuples) and that there are n_1 possible choices for the first element, n_2 possible choices for the second element, ... , and n_k possible choices for the k^{th} element, then there are $n_1 \cdot n_2 \cdot \ldots \cdot n_k$ possible k-tuples.*

Example 2.17. A baseball manager selects nine hitters and one pitcher for a starting lineup. Suppose they can choose from 13 hitters and 13 hitters. How many different possible lineups can the manager choose from?

2.5 Random Variables

Definition 2.14. Let Ω be the sample space of an experiment. A ***random variable*** is a rule that associates a number with each outcome in Ω. In other words, a random variable is a function whose domain is Ω and whose range is the set of real numbers.

Random variables are be broken down into subcategories:
1. ***Discrete random variables*** - random variables which have a sample space that is finite or countably infinite.
2. ***Continuous random variables*** - random variables which have a sample space that is uncountably infinite (such as an interval of real numbers)

Discrete and ***Continuous*** random variables use similar yet slightly different mathematical tools. Discrete random variables involve working with "sums" and continuous random variables involve working with "integrals".

Example 2.18. Give an example of a discrete random variable in hockey.

Example 2.19. Give an example of a continuous random variable in hockey.

Definition 2.15. A ***probability distribution*** is a function that gives probabilities of different possible outcomes for a given experiment.

The probability distribution for a discrete random variable, $p(x)$, is called a ***probability mass function (pmf)***.

The probability distribution for a continuous random variable, $f(x)$, is called a ***probability density function (pdf)***.

Example 2.20. Suppose the Colorado Rockies are playing a four game series against the Chicago Cubs and that the Rockies have a 65% chance of winning an individual game. Further, assume that the games are independent. The following PMF describes the outcomes (number of Rockies wins) and their probabilities.

Rockies wins, X	0.000	1.000	2.000	3.000	4.000
Probability, p(X)	0.015	0.111	0.311	0.384	0.179

What is the probability that the Rockies win zero games? What is the probability that the Rockies win at least two games? Why might the independence assumption be false?

We may be interested in describing the center or average value of our random variable. We can do this with the following definitions.

Definition 2.16. The ***expected value*** (or ***population mean*** or ***average***) of a random variable X is given by:

(i) $E[X] = \mu = \sum_{x \in \Omega} x \cdot p(x)$ (for discrete random variables)

(ii) $E[X] = \mu = \int_{x \in \Omega} x \cdot f(x) dx$ (for continuous random variables)

For this class, evaluating integrals is not essential, so we will avoid using Calculus (integrals and derivatives) when possible.

Sometimes, it makes sense to calculate the expected value of a function of a random variable. This can be easily done with a slight modification to the previous definition. Let $h(X)$ be some function of a random variable X. The expected value of $h(X)$, $E[h(X)]$, is given by:

(i) $E[h(X)] = \sum_{x \in \Omega} h(x) \cdot p(x)$ (for discrete random variables)

(ii) $E[h(X)] = \int_{x \in \Omega} h(x) \cdot f(x) dx$ (for continuous random variables)

Example 2.21. For the Rockies/Cubs four game series example, calculate $E[X]$ and $E[X^2]$.

The spread or variability associated with a random variable can be calculated using expected values as well.

Definition 2.17. The ***population variance*** of a random variable X is given by:

(i) $Var(X) = \sum_{x \in \Omega} (x-\mu)^2 \cdot p(x)$ (for discrete random variables)

(ii) $Var(X) = \int_{x \in \Omega} (x-\mu)^2 \cdot f(x) dx$ (for continuous random variables)

There is also a shortcut formula for calculating variance:

Theorem 2.7. $Var(X) = E[X^2] - (E[X])^2$

Definition 2.18. The ***population standard deviation*** of a random variable X is given by:

$SD(X) = \sigma = \sqrt{Var(X)} = \sqrt{E[X^2] - (E[X])^2}$

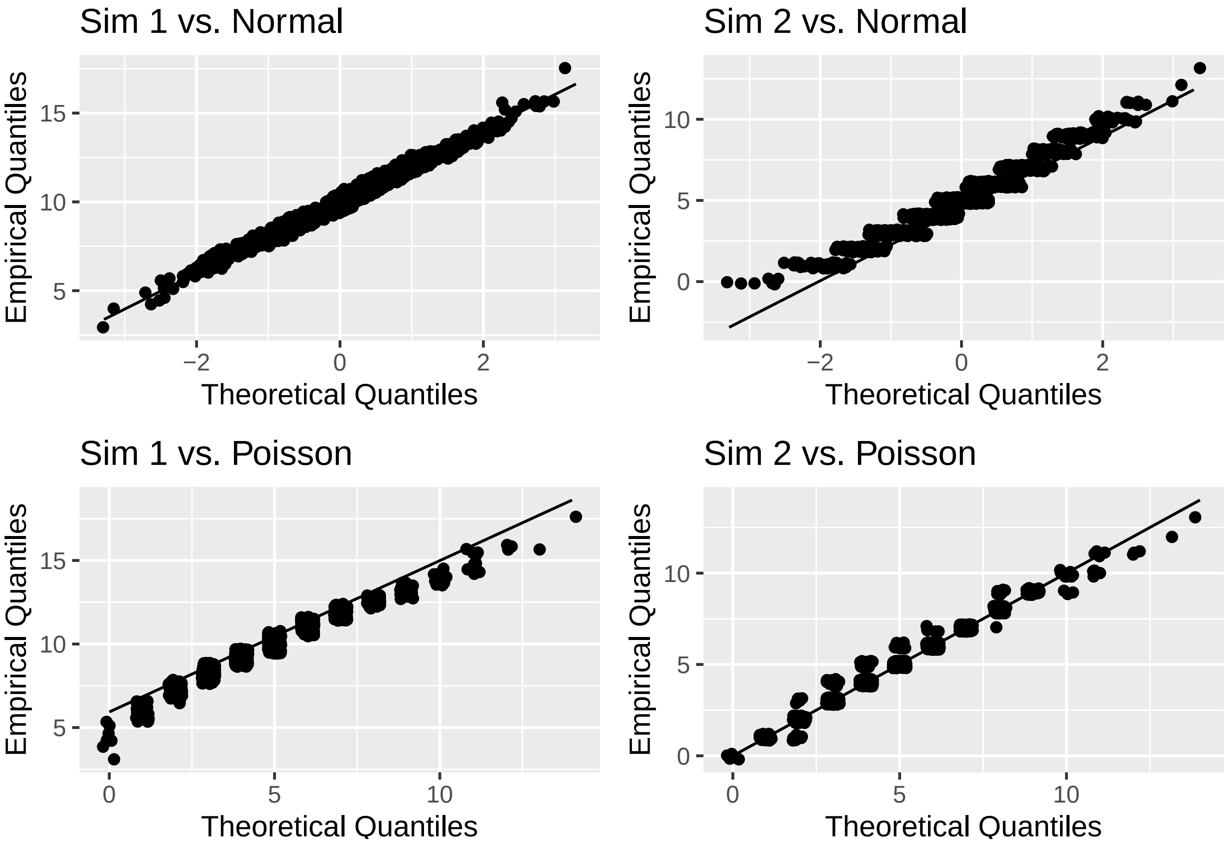
Example 2.22. For the Rockies/Cubs four game series example, calculate $Var(X)$.

Definition 2.19. A ***quantile-quantile plot*** (QQ plot) can be used to compare an empirical probability distribution against a theoretical distribution.

Example 2.23. Let's generate two simulated datasets. The first dataset will follow a normal distribution with mean 10 and variance 4 and the second dataset will follow a Poisson distribution with rate 5. We'll use QQ-plots to match each simulation with the correct distribution.

```r
set.seed(2022)
sim1.df <- data.frame(x=rnorm(n = 1000,mean = 10,sd = 2))
sim2.df <- data.frame(x=rpois(n = 1000,lambda = 5))
jitter_width <- 0.2

fig1 <- ggplot(sim1.df, aes(sample = x)) +
  stat_qq(position=position_jitter(width = jitter_width,height = jitter_width)) +
  stat_qq_line() +
  ggtitle("Sim 1 vs. Normal") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")

fig2 <- ggplot(sim2.df, aes(sample = x)) +
  stat_qq(position=position_jitter(width = jitter_width,height = jitter_width)) +
  stat_qq_line() +
  ggtitle("Sim 2 vs. Normal") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")

fig3 <- ggplot(sim1.df, aes(sample = x)) +
  stat_qq(distribution = qpois, dparams = 5,position=position_jitter(width =
  jitter_width,height = jitter_width)) +
  stat_qq_line(distribution = qpois, dparams = 5) +
  ggtitle("Sim 1 vs. Poisson") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")

fig4 <- ggplot(sim2.df, aes(sample = x)) +
  stat_qq(distribution = qpois, dparams = 5,position=position_jitter(width =
  jitter_width,height = jitter_width)) +
  stat_qq_line(distribution = qpois, dparams = 5) +
  ggtitle("Sim 2 vs. Poisson") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")
```

```
library(gridExtra)
grid.arrange(fig1,fig2,fig3,fig4,ncol=2)
```



What do you conclude from the four QQ plots?

2.6 Common Random Variables

There are several families of random variables that show up frequently in applications. Some of these random variables include: - Binomial - Geometric - Poisson - Normal

2.6.1 Binomial RVs

Definition 2.20. A ***binomial(n,p) random variable*** is a discrete random variable that counts the numbers of "successes" over a fixed number of trials, n, with each trial having an equal probability of success, p.

$P(X=k)=\binom{n}{k}p^k(1-p)^{n-k}=\frac{n!}{k!\,\cdot\,(n-k)!}p^k(1-p)^{n-k}$, where $0 \leq k \leq n, 0 \leq p \leq 1$

If $X \sim Binomial(n,p)$, then $E[X]=np$ and $Var(X)=np(1-p)$

Example 2.24. The Cubs and Rockies are playing a 4-game series. The Rockies have a 0.65 probability of winning each game, and the Cubs have a 0.35 probability. Assume each game is independent. Solve for the following quantities.

(a) The Cubs win exactly 1 game.

(b) The Rockies win exactly 2 games.

(c) The Cubs win at least 2 games.

(d) The series ends in a sweep.

(e) The expected number of wins for the Rockies.

(f) The variance and standard deviations of wins for the Rockies.

Example 2.25. Complete 10,000 simulations of the four game series between the Rockies and Cubs. For the number of Rockies wins, calculate the sample mean and sample variance and compare these to the population values. Also, plot a histogram of the sample data.

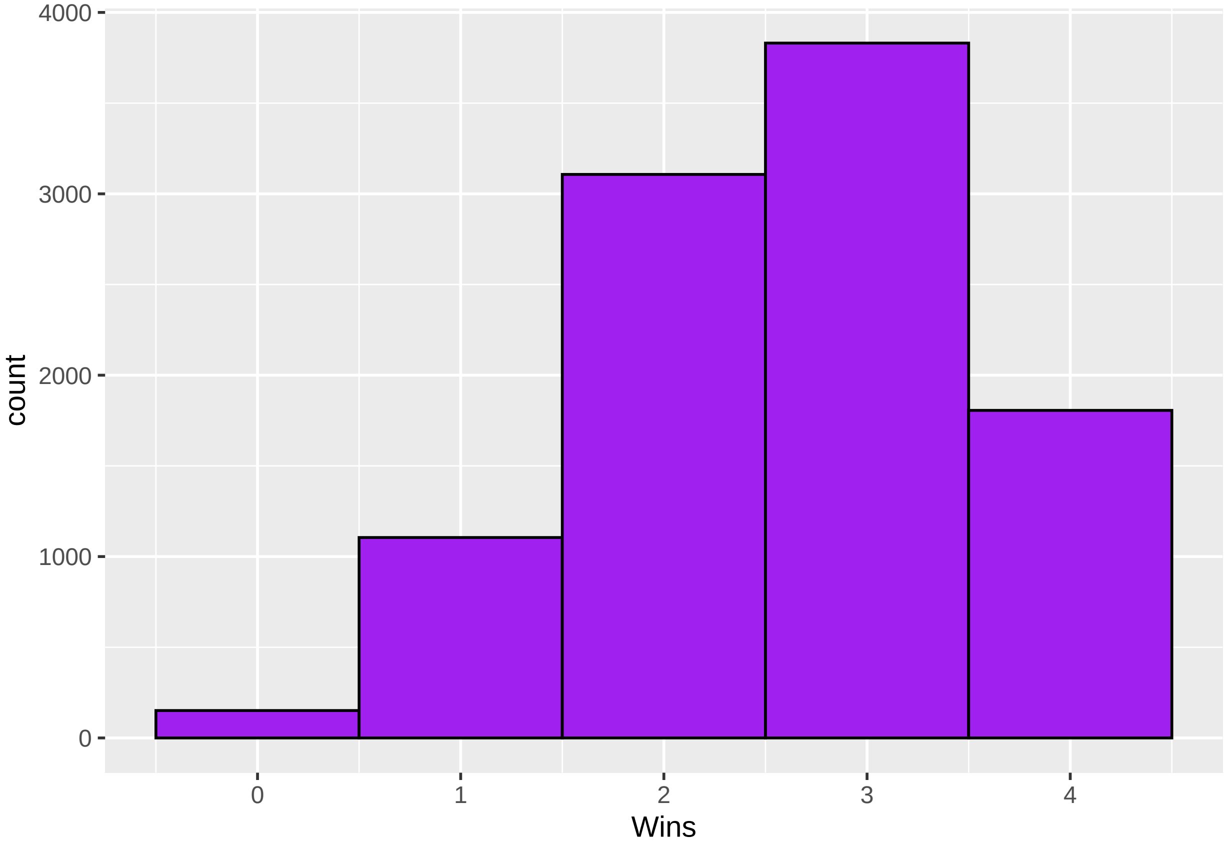
```
set.seed(2022)
rockies_wins <- rbinom(n=10000,size=4,prob=0.65)
mean(rockies_wins)
```

```
## [1] 2.6036
```

```
var(rockies_wins)
```

```
## [1] 0.9121583
```

```
rockies_wins_df <- data.frame(Wins=rockies_wins)
rockies_wins_df %>% ggplot(aes(Wins)) + geom_histogram(binwidth = 1,color =
"black", fill = "purple")
```



2.6.1.1 Binomial Coefficient Symmetry

Playoff series for a certain sports league are played as a best-of-seven series, with one team hosting four games and the opposing team hosing three. An executive for the league wishes to know the number of ways the home and away games can be assigned. (One such combination is A-A-B-B-A-B-A, the format used by the NBA and NHL for their best-of-seven series.) What is the total number of combinations?

However, instead of thinking about the number of ways to assign the games to the team that gets four home games, what if we thought about the number of ways to assign games to the team that gets three home games?

That would be $\binom{7}{3}$. We can use the `choose` command in R to find this quantity.

```
choose(7,3)
```

```
## [1] 35
```

It turns out that this binomial coefficient is also equal to 35.

Theorem: $\binom{n}{k} = \binom{n}{n-k}$

$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$

$\binom{n}{n-k} = \frac{n!}{(n-k)! \cdot (n-(n-k))!} = \frac{n!}{(n-k)! \cdot k!} = \binom{n}{k}$

2.6.2 Geometric RVs

Definition 2.21. A ***Geometric(p) random variable*** is a discrete random variable that counts the numbers of trials until a "success" occurs, where the probability of success, p, is constant across all trials.

$P(X = k) = p(1 - p)^{k-1}$, where $k \geq 1, 0 \leq p \leq 1$

If $X \sim Geometric(p)$, then $E[X] = \frac{1}{p}$ and $Var(X) = \frac{p}{1-p}$

Example 2.26. Suppose the number of shots needed by a hockey team in order to score their first goal, X, is modeled by a Geometric($\frac{1}{10}$) random variable. Use this information to answer the following questions.

(a) What is the probability that it takes exactly 3 shots to score the first goal?

(b) What is the probability that it takes less than 3 shots to score the first goal?

(c) What is the probability that it takes more than 3 shots to score the first goal?

Caution: Some references parameterize the Geometric distribution based on the number of failures before the first success, rather than the trial on which the first success occurs. This changes the PMF, mean, and variance, so be careful.

Let's simulate the number of shot attempts required to score the first goal (Geometric($p = 1/10$)) from the previous example.

```
set.seed(2020)
geometric <- rgeom(10000, 1/10)
head(geometric, 20)
```

```
##  [1]  2  2  7 55  6 11  2 11  2  5  0 50 17  2  7  0  7 19 17  1
```

Some of the values were 0, which could not happen if R was considering the number of the trial on which the first success occurred. You can add 1 to the values given by `R` to arrive at the first success distribution.

```
first_success <- geometric + 1
head(first_success, 20)
```

```
##  [1]  3  3  8 56  7 12  3 12  3  6  1 51 18  3  8  1  8 20 18  2
```

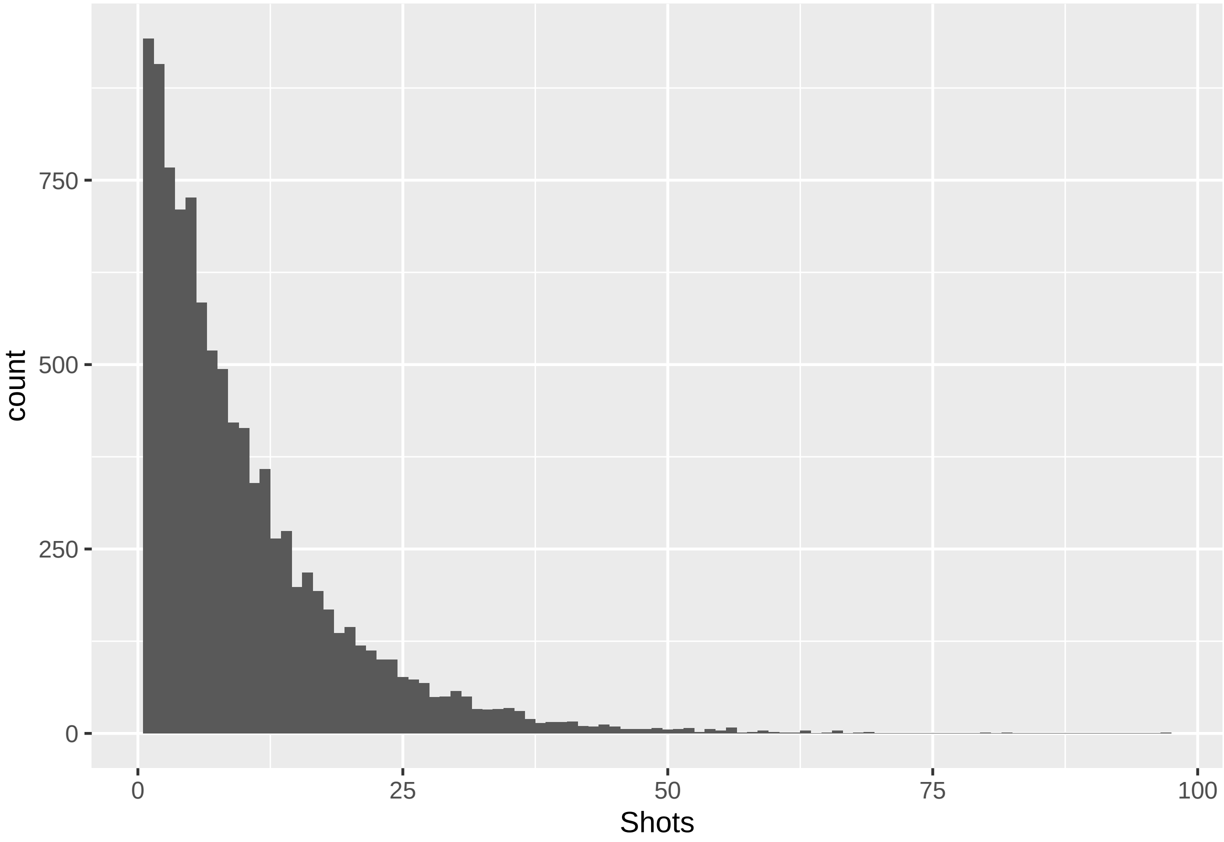
```
(mean_success <- mean(first_success))
```

```
## [1] 10.0669
```

The mean of this sample of variables is 10.0669, which is close to the expected mean of $\frac{1}{p} = 10$.

Let's plot the sample distribution of shots required to score a goal from the simulation as well.

```
first_success_df = data.frame(Shots = first_success)
first_success_df %>% ggplot(aes(x=Shots)) + geom_histogram(binwidth = 1)
```



2.6.3 Poisson RVs

Definition 2.22. A ***Poisson(λ) random variable*** is a discrete random variable that counts the numbers of “successes” for a given rate parameter, λ, for a given interval.

$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$, where $k \geq 0$,

If $X \sim Poisson(\lambda)$, then $E[X] = \lambda$ and $Var(X) = \lambda$

Example 2.27. During the 2021 Major League Soccer season, the Colorado Rapids scored 51 goals in 34 games on their way to a first-place finish in the Western Conference regular season standings.

The team scored $\frac{51}{34} = 1.5$ goals per game. Let’s model the distribution of Rapids goals using a Poisson(1.5) random variable that we’ll call Y.

(a) Which is more likely: Y taking on the value 0 or Y taking on the value 2?

We can calculate these probabilities in R using the `dpois` command.

```r
dpois(x=0, lambda=1.5)
```

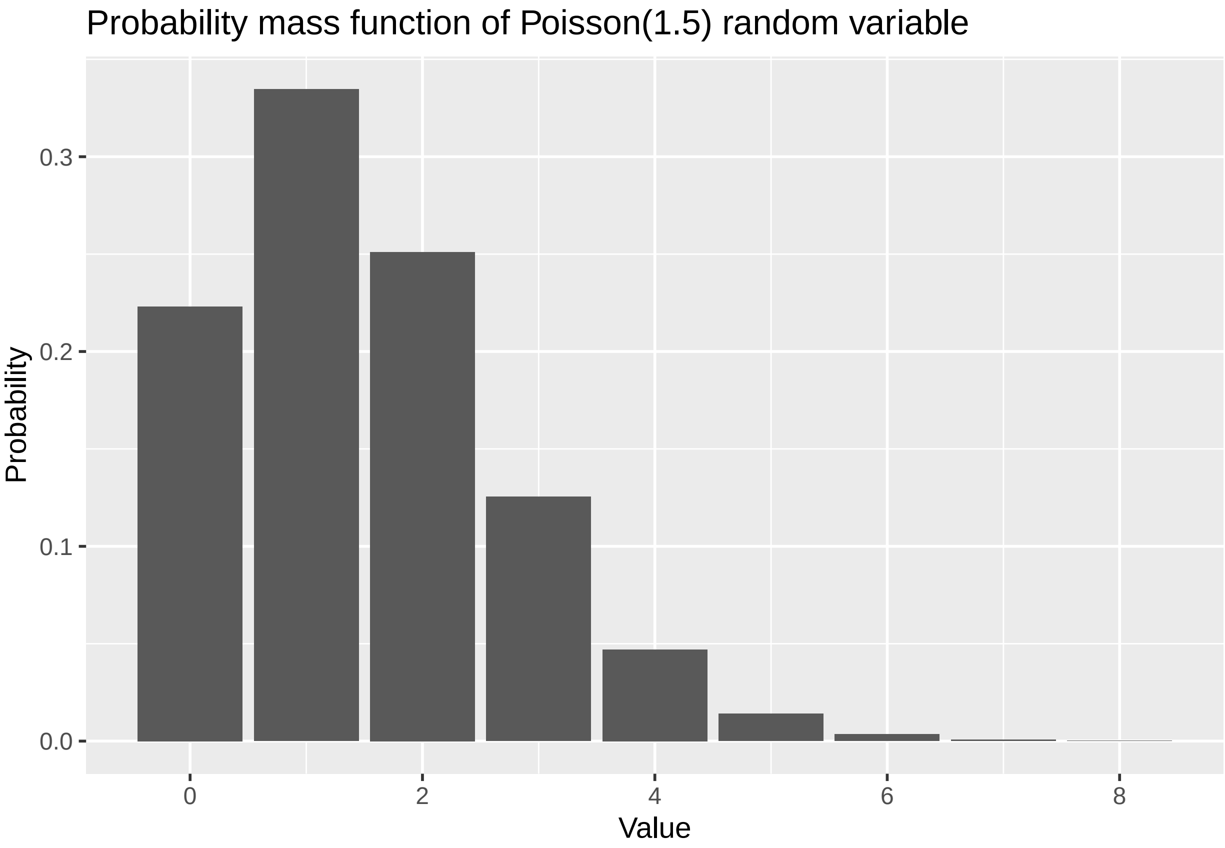
```
## [1] 0.2231302
```

```r
dpois(x=2, lambda=1.5)
```

```
## [1] 0.2510214
```

We can also plot the PMF of Y to check visually.

```
x <- 0:8
y <- dpois(x, lambda = 1.5)
poisson.pmf <- data.frame(x,y)
ggplot(transform(poisson.pmf), aes(x, y)) +
  geom_bar(stat="identity") +
  ggtitle("Probability mass function of Poisson(1.5) random variable") +
  xlab("Value") +
  ylab("Probability")
```



Let’s check whether using a Poisson distribution was appropriate by comparing it to the actual 2021 Colorado Rapids match results.

```r
# Data: https://www.espn.com/soccer/team/results/_/id/184/season/2021

library("kableExtra")
rapids21 <-
  c(0,1,1,3,3,1,3,2,1,1,2,1,2,0,1,0,3,2,2,1,1,1,2,1,0,3,0,3,1,1,2,0,1,5)

goals <- c(0:4, "5+")
actual_frequency <- c(6, 14, 7, 6, 0, 1)
actual_proportion <- actual_frequency / sum(actual_frequency)
expected_proportion <- c(dpois(0:4, lambda=1.5),
                         ppois(4, lambda=1.5, lower.tail=FALSE))
expected_frequency <- round(expected_proportion * 34, 1)

rapids.data <- data.frame(goals, actual_frequency, actual_proportion,
                          expected_frequency, expected_proportion)
names(rapids.data) = c("Goals","Actual Frequency","Actual Proportion","Expected
Frequency","Expected Proportion")

rapids.data %>% kt()
```

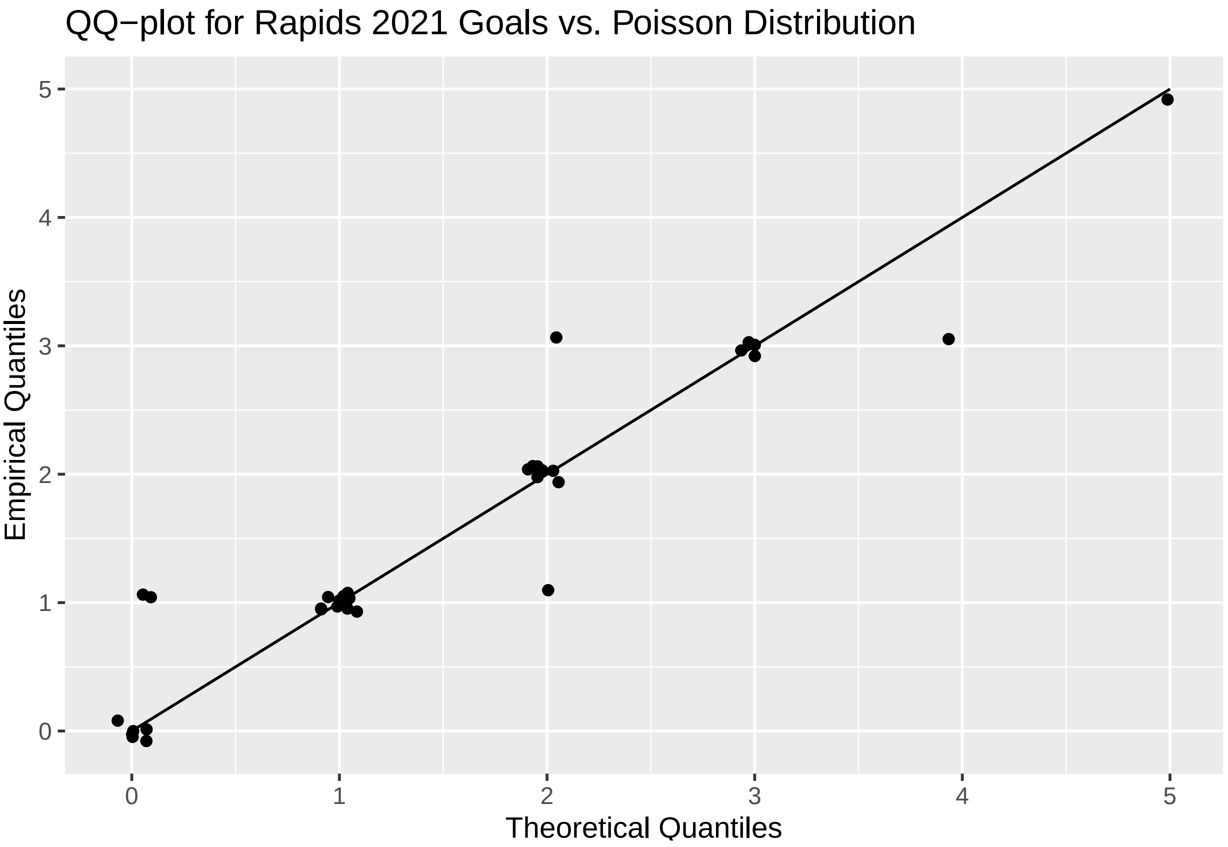
Goals	Actual Frequency	Actual Proportion	Expected Frequency	Expected Proportion
0	6	0.176	7.6	0.223
1	14	0.412	11.4	0.335
2	7	0.206	8.5	0.251
3	6	0.176	4.3	0.126
4	0	0.000	1.6	0.047
5+	1	0.029	0.6	0.019

(b) What differences do you notice between the actual results and the expected values based on the Poisson random variable?

(c) Even if the true population distribution of 2021 Rapids goals was truly a Poisson(1.5) random variable, why might the actual distribution of their goals differ from the probability mass function?

(d) Use a QQ plot to compare the probability distribution of the 2021 Rapids goals to a Poisson distribution.

```
rapids21 %>%
  as.data.frame %>%
  ggplot(aes(sample = rapids21)) +
  stat_qq(distribution = qpois,
          dparams = 1.5,position=position_jitter(width = 0.1,height = 0.1)) +
  stat_qq_line(distribution = qpois, dparams = 1.5) +
  ggtitle("QQ-plot for Rapids 2021 Goals vs. Poisson Distribution") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")
```

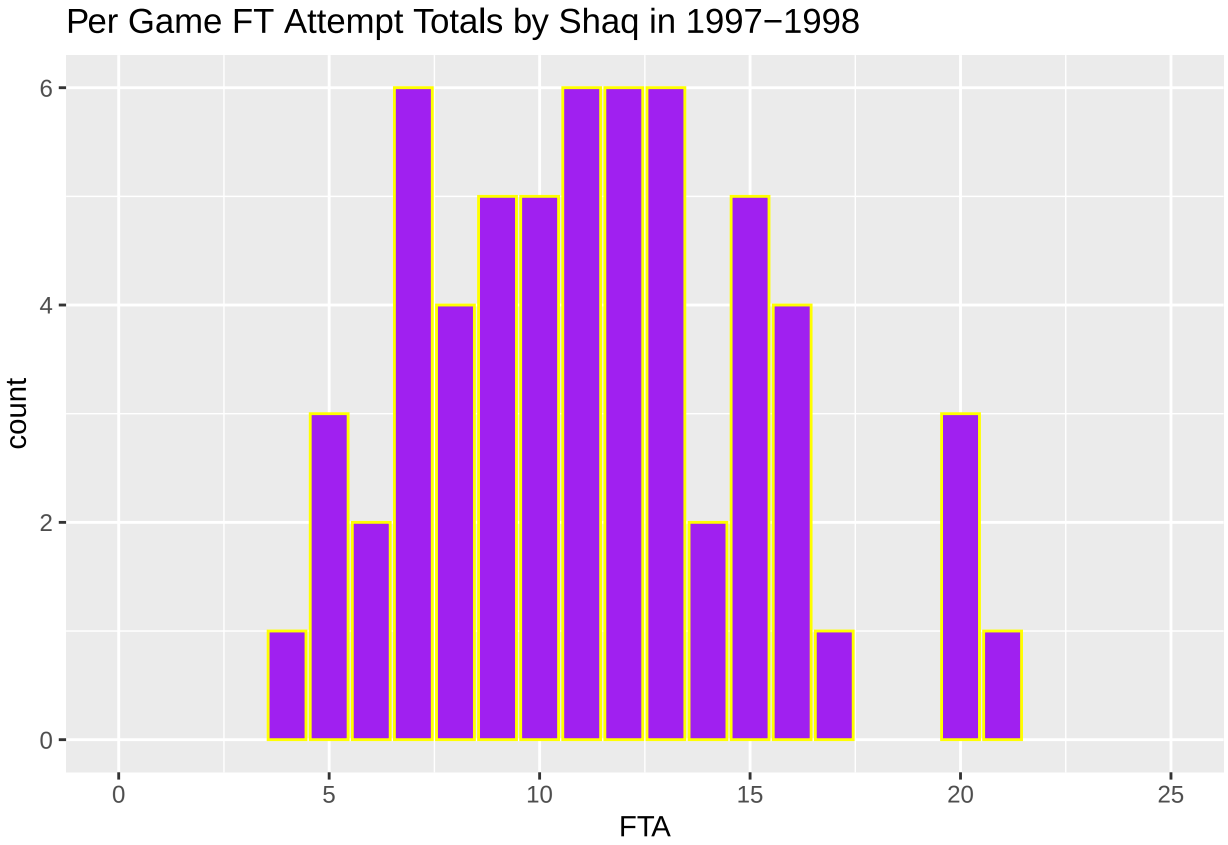


(e) What are the advantages of using the Poisson distribution to model Major League soccer goals? What are the disadvantages?

Example 2.28. In 1997-1998 with the Los Angeles Lakers, Shaquille O'Neal attempted an average of 11.35 free throws per game with a standard deviation of 4.04. Is it appropriate to model Shaq's per game free throw attempts as a Poisson(11.35) random variable?

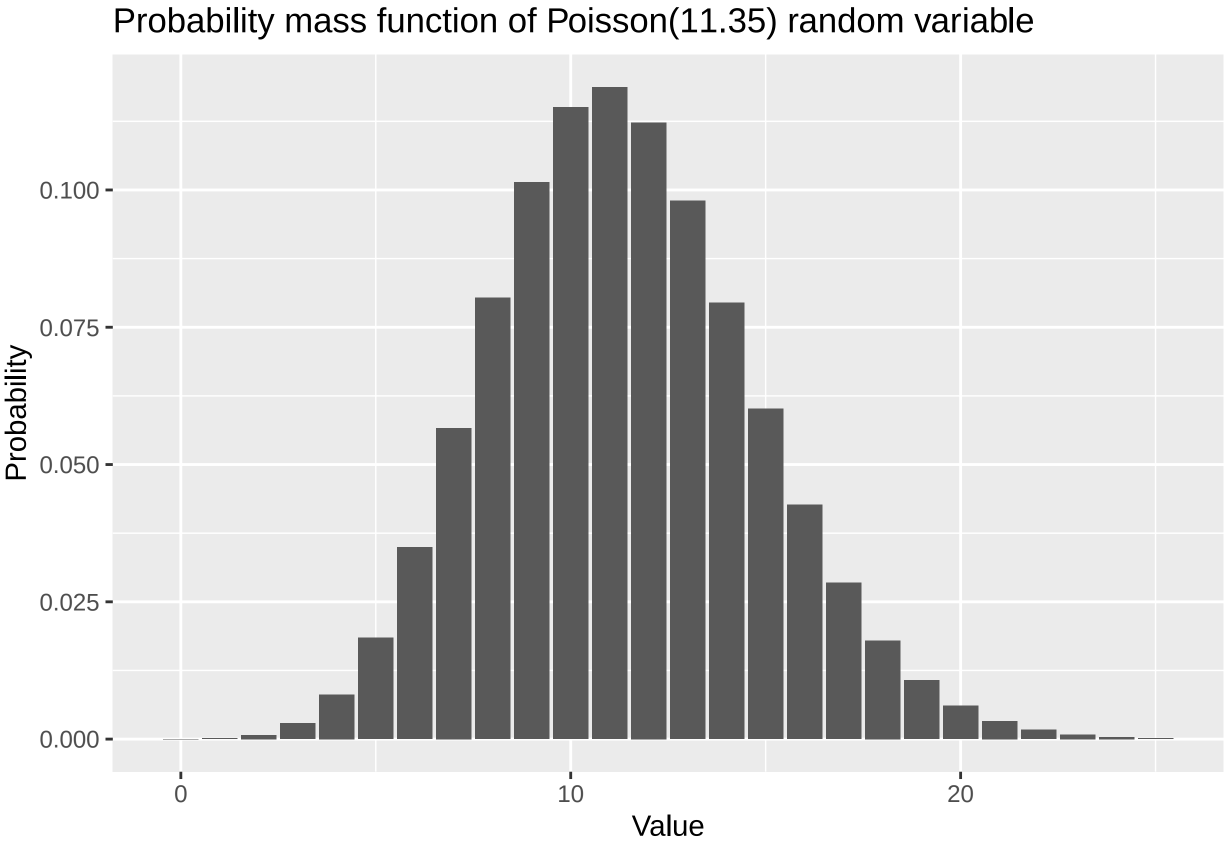
(a) Plot the data.

```
shaq9798 <- read_csv("data/shaq9798.csv")
shaq9798 %>% ggplot(aes(x=FTA))  +
  geom_bar(color = "yellow", fill = "purple") +
  ggtitle("Per Game FT Attempt Totals by Shaq in 1997-1998") +
  xlim(0,25)
```



(b) Plot the PMF of a Poisson(11.35) random variable.

```
x <- 0:25
y <- dpois(x, lambda = 11.35)
poisson.pmf <- data.frame(x,y)
ggplot(poisson.pmf, aes(x, y)) +
  geom_bar(stat="identity") +
  ggtitle("Probability mass function of Poisson(11.35) random variable") +
  xlab("Value") +
  ylab("Probability")
```



(c) What similarities and what differences do you notice?

(d) Calculate the variance of the two distributions and compare them.

```
var(shaq9798$FTA)
```

```
## [1] 16.33305
```

```
# Var(Poisson(11.35)) = 11.35
```

(e) Calculate the probability that Shaq had 20 or more free throws and compare it to $P(Poisson(11.35) \geq 20)$

```
shaqFTA <- shaq9798$FTA
shaq20 <- sum(shaqFTA >= 20)/length(shaqFTA); shaq20
```

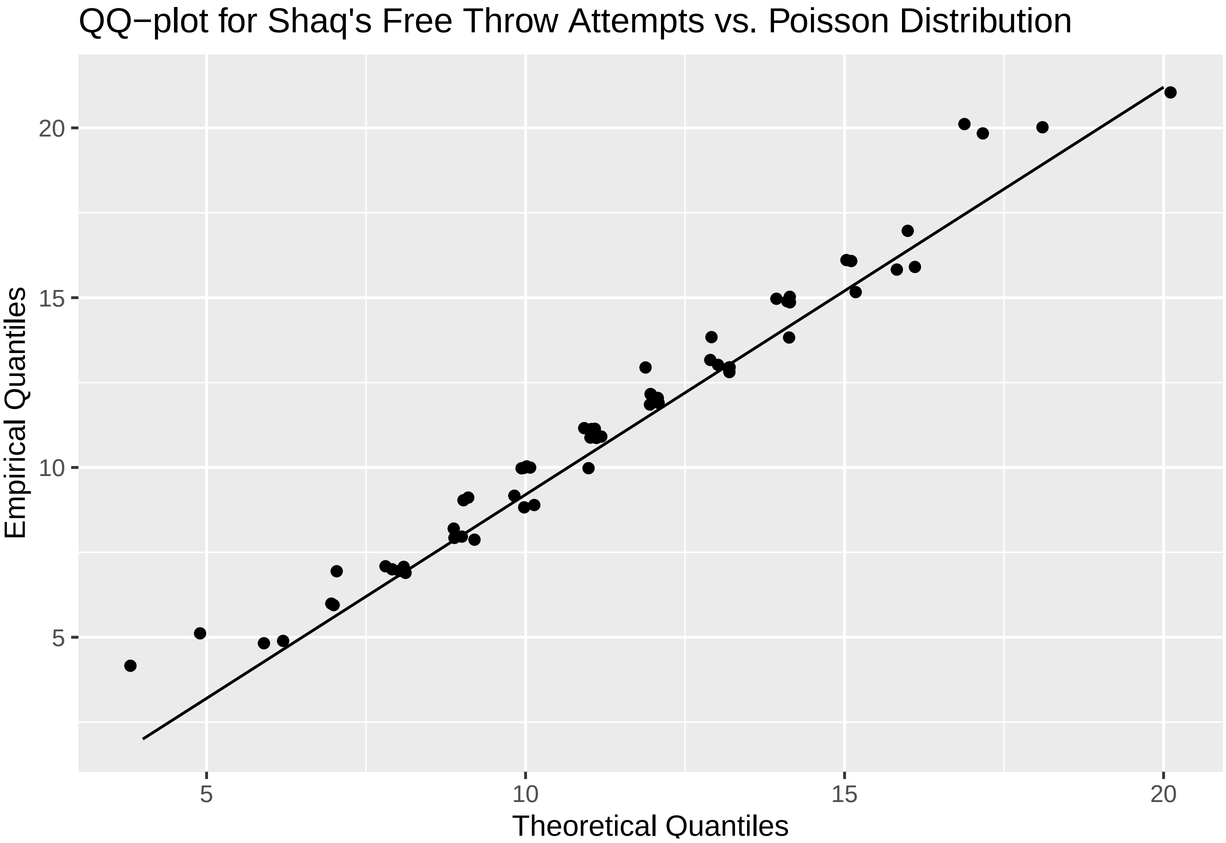
```
## [1] 0.06666667
```

```
poisson20 <- ppois(20, lambda=11.35, lower.tail=FALSE); poisson20
```

```
## [1] 0.006536079
```

(f) Create a QQ-plot of Shaq's per game free throw attempts and a Poisson distribution.

```
shaq9798 %>%
  ggplot(aes(sample = FTA)) +
  stat_qq(distribution = qpois,dparams =
  11.35,position=position_jitter(width=0.2,height=0.2)) +
  stat_qq_line(distribution = qpois, dparams = 11.35) +
  ggtitle("QQ-plot for Shaq's Free Throw Attempts vs. Poisson Distribution") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")
```



(g) Is the Poisson distribution appropriate to model Shaq's FTA per game? Explain.

2.6.4 Negative Binomial RVs

Definition 2.23. A ***Negative Binomial(r,p) random variable*** is a discrete random variable that counts the numbers of "successes" for given parameters, r and p.

$P(X = k) = \binom{k+r-1}{k}(1-p)^r p^k$, where $k = 0, 1, 2, ...$

If $X \sim NB(r, p)$, then $E[X] = \frac{rp}{1-p}$ and $Var(X) = \frac{rp}{(1-p)^2}$

The Negative Binomial distribution is often used to model count data that is "overdispersed". A property of the Poisson distribution is that the mean and variance are equal. If you are analyzing count data such that the variance is much greater than the mean (i.e., overdispersed), then the Negative Binomial distribution may be an appropriate substitute.

Given sample count data, we can estimate appropriate parameters for a Negative Binomial in many ways. One such way is to use the "method of moments" estimator.

These estimators are given by:

$\hat{p} = \frac{s^2 - \bar{x}}{s^2}$ and $\hat{r} = \frac{\bar{x}^2}{s^2 - \bar{x}}$

Example 2.29. Using Shaq's 1997-1998 data, model his per game free throw attempts as a Negative Binomial random variable.

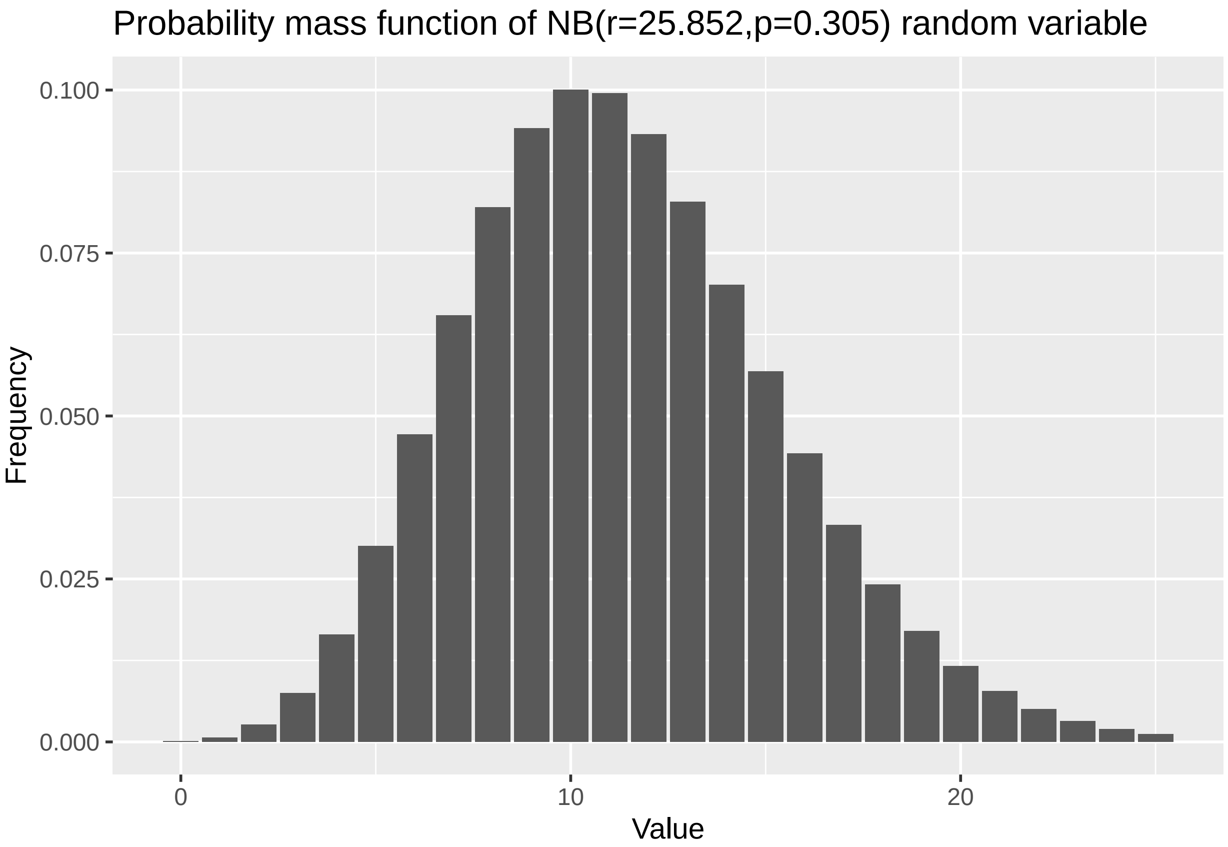
(a) Find an appropriate choice of parameters, r and p.

```
shaq.mean <- mean(shaqFTA)
shaq.var <- var(shaqFTA)
rhat <- shaq.mean^2/(shaq.var-shaq.mean)
phat <- (shaq.var-shaq.mean)/shaq.var
c(rhat,phat)
```

```
## [1] 25.85213  0.30509
```

(b) Plot the Negative Binomial distribution. Note that R uses an alternative parameterization for p. Use $prob = 1 - p$.

```
x <- 0:25
y <- dnbinom(x,size=rhat,prob=1-phat)
geom.pmf <- data.frame(x,y)
ggplot(geom.pmf, aes(x, y)) +
  geom_bar(stat="identity") +
  labs(x="Value", y="Frequency", title="Probability mass function of
  NB(r=25.852,p=0.305) random variable")
```



(c) Calculate the mean and variance of the Negative Binomial and Shaq's dataset.

```
shaq.mean <- mean(shaqFTA)
shaq.var <- var(shaqFTA)
NB.mean <- (rhat*phat)/(1-phat)
NB.var <- (rhat*phat)/(1-phat)^2
c(shaq.mean,shaq.var)
```

```
## [1] 11.35000 16.33305
```

```
c(NB.mean,NB.var)
```

```
## [1] 11.35000 16.33305
```

(d) Calculate the probability that Shaq had 20 or more free throws and compare it to $P(NB(r = 25.852, p = 0.305) \geq 20)$

```
shaq20 <- sum(shaqFTA >= 20)/length(shaqFTA); shaq20
```

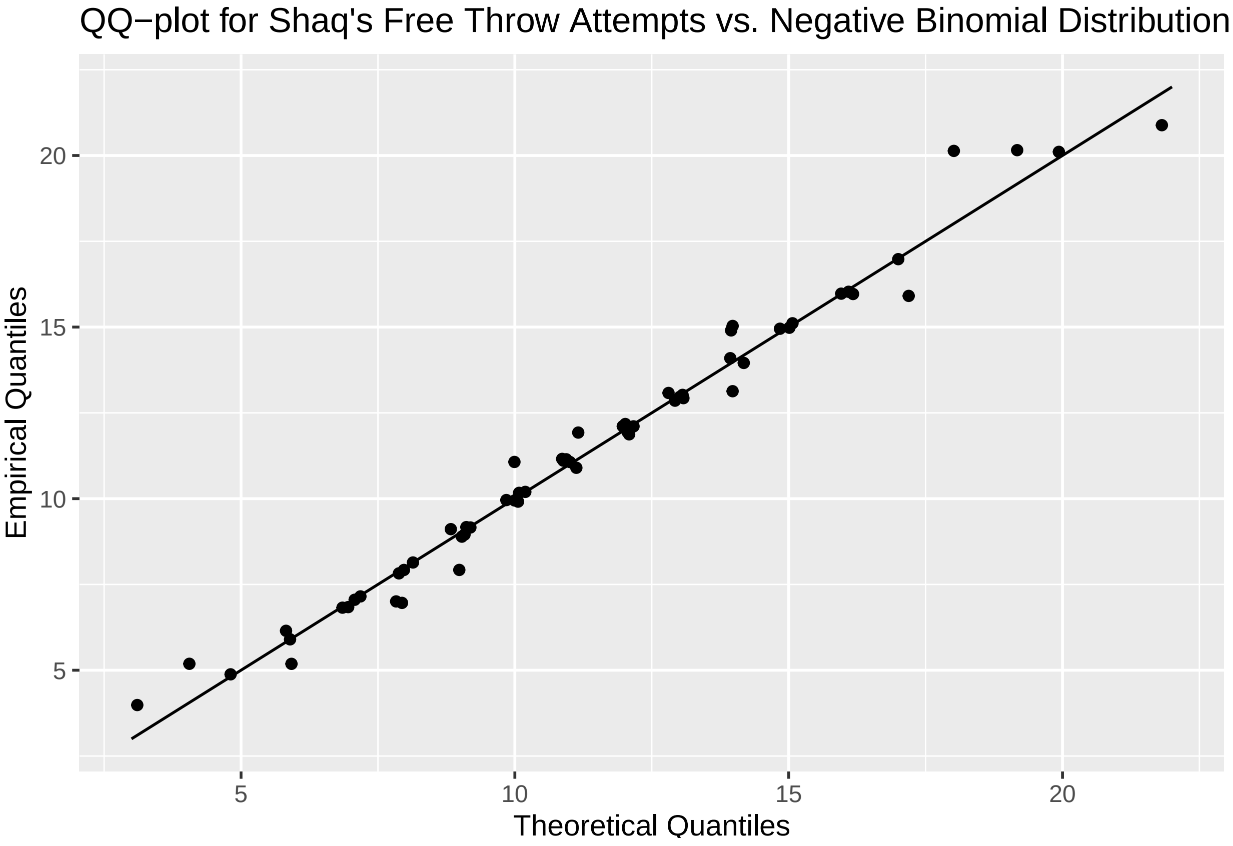
```
## [1] 0.06666667
```

```
nb20 <- pnbinom(20,size=rhat,prob=1-phat,lower.tail=FALSE); nb20
```

```
## [1] 0.0208711
```

(e) Create a QQ-plot of Shaq's per game free throw attempts and a Negative Binomial distribution.

```
shaq9798 %>%
  ggplot(aes(sample = FTA)) +
  stat_qq(distribution = qnbinom,dparams =
  list(size=25.852,mu=11.35),position=position_jitter(width=0.2,height=0.2)) +
  stat_qq_line(distribution = qnbinom, dparams = list(size=25.852,mu=11.35)) +
  ggtitle("QQ-plot for Shaq's Free Throw Attempts vs. Negative Binomial
  Distribution") +
  xlab("Theoretical Quantiles") +
  ylab("Empirical Quantiles")
```



(f) Is the Negative Binomial distribution appropriate to model Shaq's FTA per game? How does it compare to using the Poisson distribution? Explain.

2.6.5 Normal RVs

Definition 2.24. A ***Normal(μ,σ^2) random variable*** is a continuous random variable that is bell-shaped with mean μ and variance σ^2.

To calculate probabilities under the normal curve, you need either to integrate, use a table, or a computer.

Note that a normal random variable can be standardized by using: $z = \frac{x-\mu}{\sigma}$

Theorem 2.8. *For a normal(μ,σ^2) random variable, we have the following approximations:*
- About 68% of the data falls within one standard deviation of the mean (i.e., $\mu \pm \sigma$)
- About 95% of the data falls within two standard deviations of the mean (i.e., $\mu \pm 2\sigma$)
- About 99.7% of the data falls within three standard deviations of the mean (i.e., $\mu \pm 3\sigma$)

Example 2.30. The skills (or tools) of a baseball player are often rated on a scale of 20-80, where 50 is an average grade, 20 is the lowest grade, and 80 is the highest grade. The distribution of tool grades is approximately normally distributed ($\mu = 50, \sigma = 10$).

See https://blogs.fangraphs.com/scouting-explained-the-20-80-scouting-scale/ for more details. Calculate the following probabilities.

(a) Former Rockie Nolan Arenado has been graded to have game power of 70. Game power estimates a player's ability to hit home runs. Approximately what percentage of baseball players have equal or greater game power than Arenado?

(b) Mike Trout has been graded to have raw power of 55. Raw power estimates a player's ability to hit baseballs hard (i.e., hard hit rate). Approximately what percentage of baseball players have equal or less raw power than Trout?

(c) Suppose a Rockies prospect is said to be in the top 10% of all baseball players in terms of their speed. What approximate speed grade would correspond to the player?

(d) Suppose a Rockies prospect is said to be in the bottom 20% of all baseball players in terms of their hit ability. What approximate hit grade would correspond to the player?

(e) Between what two grades do approximately 95% of all players lie for a given tool?

Let's check our answers:

```
a <- 1-pnorm(q=70,mean=50,sd=10); a
```

```
## [1] 0.02275013
```

```
b <- pnorm(q=55,mean=50,sd=10); b
```

```
## [1] 0.6914625
```

```
c <- qnorm(0.1,mean=50,sd=10,lower.tail = F); c
```

```
## [1] 62.81552
```

```
d <- qnorm(0.2,mean=50,sd=10,lower.tail = T); d
```

```
## [1] 41.58379
```

```
e <- pnorm(q=70,mean=50,sd=10) - pnorm(q=30,mean=50,sd=10); e
```

```
## [1] 0.9544997
```

Example 2.31. Player X has a projected mean WAR of 3 with standard deviation of 2 and player Y has a projected mean WAR of 1.5 with a standard deviation of 3. Assume projected WAR is normally distributed. What is the probability that Player X outperforms Player Y?

Link to WAR explanation: https://www.mlb.com/glossary/advanced-stats/wins-above-replacement

We want $P(X > Y)$ or $P(X - Y > 0)$

```r
# Calculate probability Z<=0
p <- pnorm(0,1.5,sqrt(5),lower.tail = F); p
```

```
## [1] 0.7488325
```

2.7 Win Probability Models

The probability an NFL team will win can be modeled as a normal random variable using the Vegas line. This method is outlined by Wayne Winston in *Mathletics* building on previous research by Hal Stern.

2.7.1 Estimating Pregame Win Probability

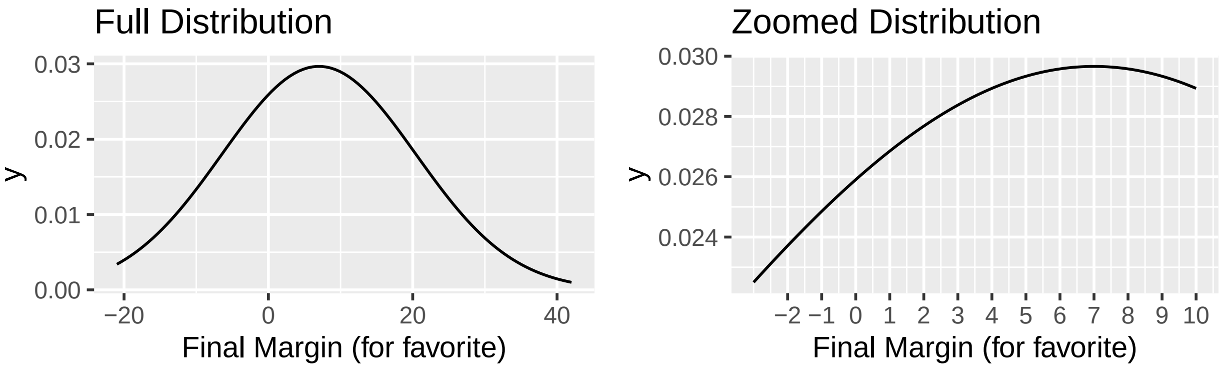
Winston estimates that the final margin of victory is approximately a normal random variable with a mean of the Vegas line and a standard deviation between 13-14. Winston and Stern estimated the standard deviation to be **13.86** based on data from the 1981, 1983, and 1984 NFL seasons.

This estimated standard deviation has since been updated to be **13.45** based on data from 1978 – 2012 NFL seasons.

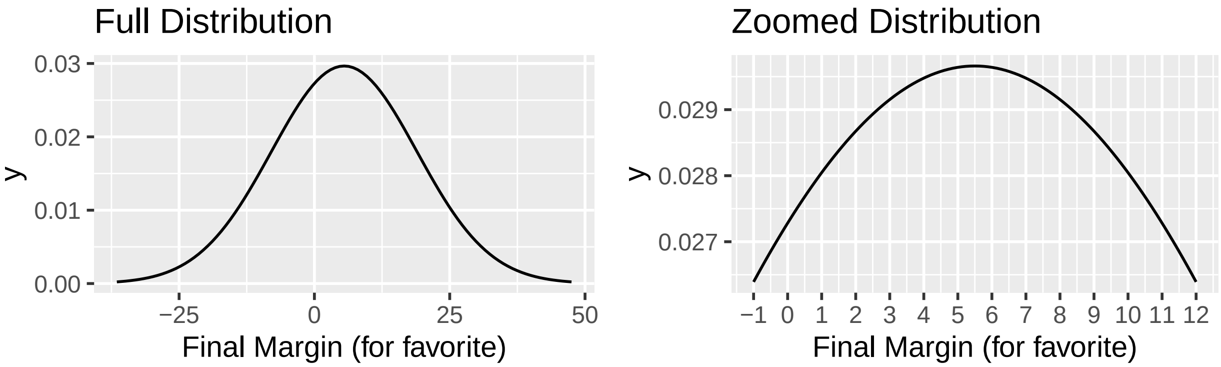
Reference: https://www.pro-football-reference.com/about/win_prob.htm

Example 2.32. Sketch the distribution of the final margin of victory for an NFL team that is favored by 7 points. Shade the area for a win in regulation (for the favorite) and a tie in regulation.

```r
library(ggplot2)
library(gridExtra)


final_margin_full <- ggplot(data.frame(x = c(-21, 42)), aes(x = x)) +
        stat_function(fun = dnorm, args = list(mean = 7, sd = 13.45)) +
  xlab("Final Margin (for favorite)") + ggtitle("Full Distribution")

final_margin_zoom <- ggplot(data.frame(x = c(-3, 10)), aes(x = x)) +
        stat_function(fun = dnorm, args = list(mean = 7, sd = 13.45)) +
  xlab("Final Margin (for favorite)") + ggtitle("Zoomed Distribution") +
  scale_x_continuous(breaks=seq(-2, 10, 1))

grid.arrange(final_margin_full, final_margin_zoom, ncol=2)
```



Example 2.33. In Super Bowl 50, the Denver Broncos were 5.5 point underdogs (+5.5) against the Carolina Panthers. Estimate the pregame win probability for the Denver Broncos.



```r
fav_line <- 5.5
full_sd <- 13.45

win_prob <- pnorm(q = 0.5, mean = fav_line, sd = full_sd, lower.tail = FALSE)

tie_prob <- pnorm(q = 0.5, mean = fav_line, sd = full_sd, lower.tail = TRUE) -
            pnorm(q = -0.5, mean = fav_line, sd = full_sd, lower.tail = TRUE)

# win probability for favorite (panthers)
win_prob <- win_prob + 1/2 * tie_prob

# win probability for underdog (broncos)
(broncos_win_prob <- 1 - win_prob)
```

```
## [1] 0.3414021
```

2.7.2 Estimating In-Game Win Probability

Winston proposed an updated method to calculate in-game win probabilities based on the time remaining in the game, current score margin, and Vegas line.

In this proposed method, the pregame mean (Vegas line) and pregame variance (13.45) are scaled down linearly as a function of time remaining in the game.

For instance, since there are four 15-minute quarters (total regulation game time is 60 minutes), then the in-game mean and standard deviation are calculated as follows.

$$\mu_{updated} = Line \cdot \frac{\text{time remaining}}{60}$$

$$\sigma_{updated} = \frac{13.45}{\sqrt{60/\text{time remaining}}}$$

Note: This model assumes perfectly neutral possession, down, distance, and field-position conditions.

Reference: https://www.pro-football-reference.com/about/win_prob.htm

Example 2.34. In Super Bowl 50, the Denver Broncos led the Carolina Panthers 10-0 after the first quarter. The Broncos were 5.5 point underdogs at the start of the game. Use Winston's method to estimate the probability of a Broncos win after one quarter.

```r
fav_margin <- -10
fav_line <- 5.5
full_sd <- 13.45
full_min <- 60
new_min <- 45

new_mean <- fav_line * new_min/full_min
new_sd <- full_sd / sqrt(full_min/new_min)

win_prob <- pnorm(q = 0.5 - fav_margin, mean = new_mean,
                  sd = new_sd, lower.tail = FALSE)

tie_prob <- pnorm(q = 0.5 - fav_margin, mean = new_mean,
                  sd = new_sd, lower.tail = TRUE) -
            pnorm(q = -0.5 - fav_margin, mean = new_mean,
                  sd = new_sd, lower.tail = TRUE)

# win probability for favorite (panthers)
win_prob <- win_prob + 1/2 * tie_prob

# win probability for underdog (broncos)
(broncos_win_prob <- 1 - win_prob)
```

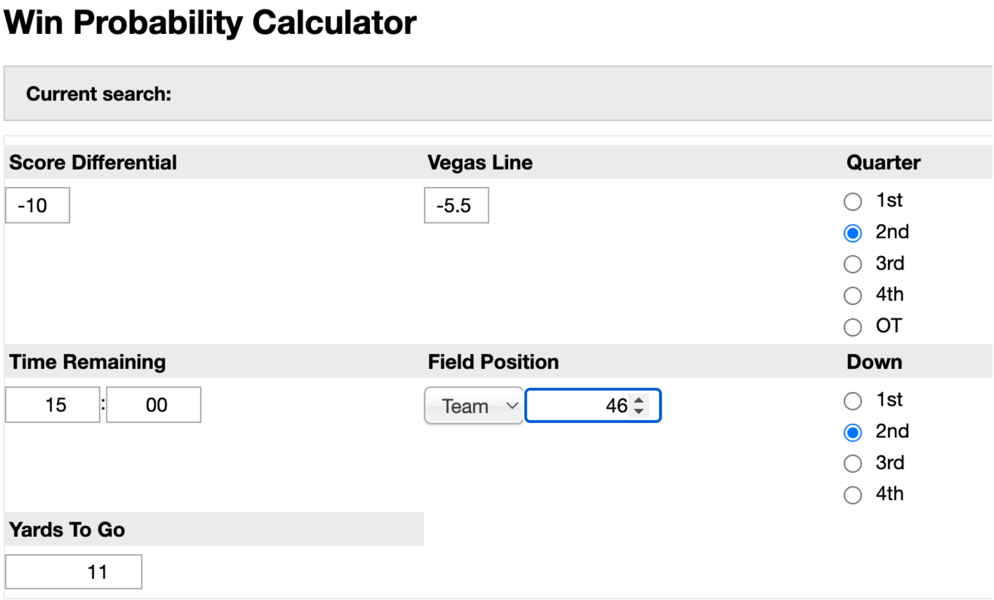
```
## [1] 0.6928385
```

The win probability models outlined above use only the Vegas line, the current score, and the time remaining in the game. More accurate estimation models will also consider possession, down, distance, and field-position conditions. Pro Football Reference offers an in-game win probability model with these additional factors.

Reference: https://www.pro-football-reference.com/boxscores/win_prob.cgi

Example 2.35. The Broncos led the Panthers 10-0 at the end of the first quarter. The Panthers had 2nd and 11 at their own 46 at the beginning of the second quarter. Calculate the Broncos win probability.

Reference: https://www.pro-football-reference.com/boxscores/201602070den.htm



Win Probability Calculator

Current search:

Score Differential: -10

Vegas Line: -5.5

Quarter: ○ 1st ● 2nd ○ 3rd ○ 4th ○ OT

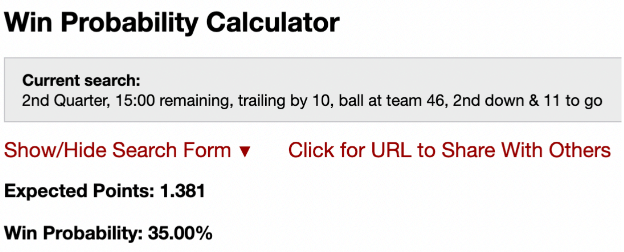
Time Remaining: 15 : 00

Field Position: Team 46

Down: ○ 1st ● 2nd ○ 3rd ○ 4th

Yards To Go: 11

Figure 2.1: Pro Football Reference Win Probability Calculator Inputs



Win Probability Calculator

Current search:
2nd Quarter, 15:00 remaining, trailing by 10, ball at team 46, 2nd down & 11 to go

Expected Points: 1.381

Win Probability: 35.00%

Figure 2.2: Pro Football Reference Win Probability Calculator Output

Class Reading: Chapters 6–7 in the *Hidden Game of Football* by Carroll, Palmer, Thorn

Chapter 3

Odds and Bets

Sports Betting in USA

In 2018, the United States Supreme Court overturned a 1992 federal law that banned commercial sports betting in most states. For more about this ruling, see this New York Times article: https://www.nytimes.com/2018/05/14/us/politics/supreme-court-sports-betting-new-jersey.html

In Colorado, you must be at least 21 years old to gamble including betting on sports games.

Gambling Addiction

Gambling addiction, compulsive gambling, or gambling disorder is a serious impulse-control disorder. Here is a link to the Mayo Clinic's discussion of compulsive gambling: https://www.mayoclinic.org/diseases-conditions/compulsive-gambling/symptoms-causes/syc-20355178

Gambling is ***not*** encouraged and can lead to numerous problems. If you choose to gamble on sports, don't bet beyond your means and seek help if you worry that you may be suffering from gambling addiction.

3.1 Odds

In the previous chapter, we quantified uncertainty using probabilities (numbers between 0 and 1) and percentages (numbers between 0 and 100). We can also quantify uncertainty using **odds**.

Definition 3.1. The ***odds*** (in favor) of an outcome A is the probability of A divided by the probability of A^C.

$$Odds = \frac{p}{1-p}$$

Note: We may also be given the **odds against** a particular outcome. In this case, we have $OddsAgainst = \frac{1-p}{p}$.

Some Common Odds:

Odds	p	q	Odds	p	q
0:1	0.000	1.000	1:0	1.000	0.000
1:1	0.500	0.500	1:1	0.500	0.500
2:1	0.667	0.333	1:2	0.333	0.667
3:1	0.750	0.250	1:3	0.250	0.750
4:1	0.800	0.200	1:4	0.200	0.800
5:1	0.833	0.167	1:5	0.167	0.833
10:1	0.909	0.091	1:10	0.091	0.909
100:1	0.990	0.010	1:100	0.010	0.990

In the following table, the odds of an outcome along with the probability of the outcome, p, and the probability of the outcome's complement, *q=1-p* are given.

Example 3.1. For the following examples, convert between probability and odds.

(a) Suppose that the probability that the Cubs win the 2025 World Series is 0.03. What are the odds in favor?

(b) Suppose that the odds that the Rockies win the World Series in 2025 is 100:1. What is the probability in favor?

(c) Suppose there is a 1% chance that CSU wins a national championship in any sport in the next decade. What are the probability and the odds against CSU winning a national championship in the next decade?

Example 3.2. Suppose that there are three finalists in an Olympic competition. It is estimated that Athlete A has a 50% chance of winning, Athlete B has a 40% chance of winning, and Player C has a 10% chance of winning. Calculate the odds against each of the athletes winning the competition.

3.2 Gambling Odds

Here's a helpful resource on calculating gambling odds:
https://www.actionnetwork.com/education/decimal-odds

Gambling odds are a bit different that odds in a probability context. Gambling odds tell the amount that the bookmaker will pay out for a winning bet. For example, if a bookmaker is offering "10:1" that the Rockies win the next World Series, the bookmaker will pay out 10x the original wager plus the original wager if the Rockies win the World Series and the bookmaker will keep the original bet if the Rockies fail to win the World Series.

Definition 3.2. The ***implied probability***, $\tilde{p}$, of a wager is the probability that corresponds to the odds of the wager.

$$\tilde{p} = \frac{Odds}{Odds + 1}$$

Definition 3.3. ***Fractional odds*** are common in horse racing and quote the net total that will be paid out to the bettor, should he or she win, relative to the stake. The numerator and denominator of fractional odds are always positive integers. For example, a \$100 wager on a "5:1" bet would result in a payout of $5 \cdot \$100 + \$100 = \$600$ if the wager is correct and a payout of \$0 if the wager is incorrect.

Definition 3.4. ***Decimal Odds*** represent the multiplier of a winning bet and is calculated by taking the inverse of the implied probability. For instance, if you bet \$100 with decimal odds of 1.8, your payout is \$180 (\$100 wager + \$80 winnings).

$$Decimal\,Odds = [\tilde{p}]^{-1}$$

3.3 Types of Bets

Definition 3.5. A ***moneyline bet*** are common bets in American sports. When the moneyline is positive, the figure tells what the winning payout would be on a $100 bet. When the moneyline is negative, the figure tells what wager is required for a winning payout of $100.

Example 3.3. For the following scenarios, assume a wager of $100. Calculate the payout for a winning bet.

(a) Fractional odds of 4/1

(b) Moneyline +400

(c) Moneyline -300

(d) Decimal Odds of 1.5

Definition 3.6. A *moneyline bet* refers to the odds of a straight-up outcome on a game without consideration of a point spread. Typically, favorites will have negative moneyline odds and underdogs will have a positive moneyline odds.

If ML is negative:

$$\tilde{p} = \frac{-ML_{NEG}}{-ML_{NEG} + 100}$$

If ML is positive:

$$\tilde{p} = \frac{100}{ML_{POS} + 100}$$

Example 3.4. Calculate the implied probabilities for the following examples.

(a) Fractional odds of 4/1

(b) Moneyline -125

(c) Moneyline +125

(d) Decimal Odds of 1.5

Example 3.5. Suppose that for an upcoming NFL game, a moneyline wager on the Broncos is -105 and a moneyline wager on the Raiders is -120.

(a) Calculate the implied probabilities of the two possible outcomes. (Note that sometime ties are an optional wager. Other times, ties will result in a push, that is, no winner.)

(b) What is the sum of the implied probabilities?

(c) Does the result from (b) violate the axioms of probability? If so, why?

Note: The goal of bookmaking is to make money, so there will be almost certainly be a house advantage. This means that the sum of the implied probabilities will be greater than 1.

Definition 3.7. The ***house advantage*** (or ***hold percentage***) is the percentage of money that sportsbooks keep for every dollar earned.

$$House\ Advantage = 100 \cdot \sum_{i=1}^{n} \tilde{p}_i - 100$$

Example 3.6. Suppose that a bookmaker is offering a moneyline wager on the Broncos at -105 and a moneyline wager on the Raiders at -120. Calculate the bookmakers expected winnings based on the following amounts of total bets.

(a) $500 wagered on the Broncos, $500 wagered on the Raiders

(b) $250 wagered on the Broncos, $750 wagered on the Raiders

(c) $750 wagered on the Broncos, $250 wagered on the Raiders

(d) $0 wagered on the Broncos, $1000 wagered on the Raiders

(e) $1000 wagered on the Broncos, $0 wagered on the Raiders

(f) What is the house advantage?

Example 3.7. A ***point spread bet*** is a bet based on the projected margin of victory that can result in a win, loss, or push.

For example, if a sportsbook offers Rockies -1.5 against the Cubs and you bet on the Rockies, if the Rockies win by 2 or more runs, you win and if the Rockies win by 1 run or lose, then you lose.

Example 3.8. Suppose a sportsbook offers Broncos +14 against the Raiders at -110 (they also offer Raiders -14 at -110) and you have $20 to wager. Calculate the payout for the following scenarios.

(a) You bet $20 on the Broncos and the final score is Broncos 21 Raiders 20. What is your payout?

(b) You bet $20 on the Raiders and the final score is Broncos 21 Raiders 20. What is your payout?

(c) You bet $10 on the Broncos and $10 on the Raiders and the final score is Broncos 21 Raiders 20. What is your payout?

Example 3.9. A ***parlay bet*** is a combination of multiple bets where the winnings from a winning bet are placed on other bets. All bets must be win for the parlay bet to pay out.

Example 3.10. Suppose that CSU is a three point favorite (-3) against CU in a women's basketball game and the over/under on total number of points is 97. Assume that the price of bets is -110.

(a) Suppose you place two bets, CSU -3 and Over 97, and the outcome of the game is CSU 60 CU 40. What is your payout?

(b) Suppose you place a parlay bet on CSU -3 and Over 97 and the outcome of the game is CSU 60 CU 40. What is your payout?

(c) Suppose you place two bets, CSU -3 and Over 97, and the outcome of the game is CSU 44 CU 40. What is your payout?

(d) Suppose you place a parlay bet on CSU -3 and Over 97 and the outcome of the game is CSU 44 CU 40. What is your payout?

Reference:
https://www.wikihow.com/Calculate-Odds

Class Reading: Chapters 44 in the *Mathletics* by Wayne Winston

Chapter 4

Monte Carlo Simulation

4.1 Basics

Monte Carlo Simulation is a collection of computer-driven, computational algorithms that use repeated random sampling to calculate estimates. The basic steps for such a simulation are as follows:

1) Set seed for replicability
2) Initialize all variables/vectors
3) Loop through “n” simulations and save simulated values
4) Analyze the simulated values from the “n” simulations

One function that will be particularly useful for simulation is `set.seed()`.

`set.seed()` allows us to replicate any simulation by giving the initial seed for the simulation. The actual number that is “seeded” is not particularly important though if you want to replicate the same simulations, you will want to re-use this number.

Example 4.1. Simulate 10 overtime coin tosses with and without using `set.seed()` and compare the results

```r
# Sample 1
sample(c("H","T"),size=10,prob=c(0.5,0.5),replace=T)
```

```
##  [1] "H" "T" "H" "H" "H" "T" "T" "T" "H" "T"
```

```r
# Sample 2
sample(c("H","T"),size=10,prob=c(0.5,0.5),replace=T)
```

```
##  [1] "T" "T" "T" "H" "H" "T" "T" "T" "H" "H"
```

```r
# Sample 3
set.seed(2020)
sample(c("H","T"),size=10,prob=c(0.5,0.5),replace=T)
```

```
##  [1] "H" "T" "H" "T" "T" "T" "T" "T" "T" "H"
```

```r
# Sample 4
set.seed(2020)
sample(c("H","T"),size=10,prob=c(0.5,0.5),replace=T)
```

```
##  [1] "H" "T" "H" "T" "T" "T" "T" "T" "T" "H"
```

Simulation can be very helpful when you want to estimate quantities that are not easily solved using analytical methods like formulas.

Example 4.2. Shaquille O'Neal has a career free throw percentage of 52.7%. Suppose that Shaq takes 10 free throw shots. What is the probability that he makes all 10 shots?

In this case, we can calculate the exact probability of interest using binomial random variable.

```
dbinom(x=10,size=10,prob=0.527)
```

```
## [1] 0.001652366
```

In more complicated simulations, there may not be an easy formula to use to calculate the value of interest. In these situations, simulation can be very helpful in estimating quantities.

```
set.seed(2020)

# Number of Simulations
n.sims <- 10000

# Initialize FT variable with 10000 zeros
FT <- rep(0,n.sims)

for(i in 1:n.sims){
  # Simulate 10 free throws
  temp <- sample(x=c(0,1), size = 10, replace = T, prob = c(0.473,0.527) )

  # Count the number of free throws made and store them in FT
  FT[i] <- sum(temp)
}

prob10 <- sum(FT == 10)/n.sims; prob10
```

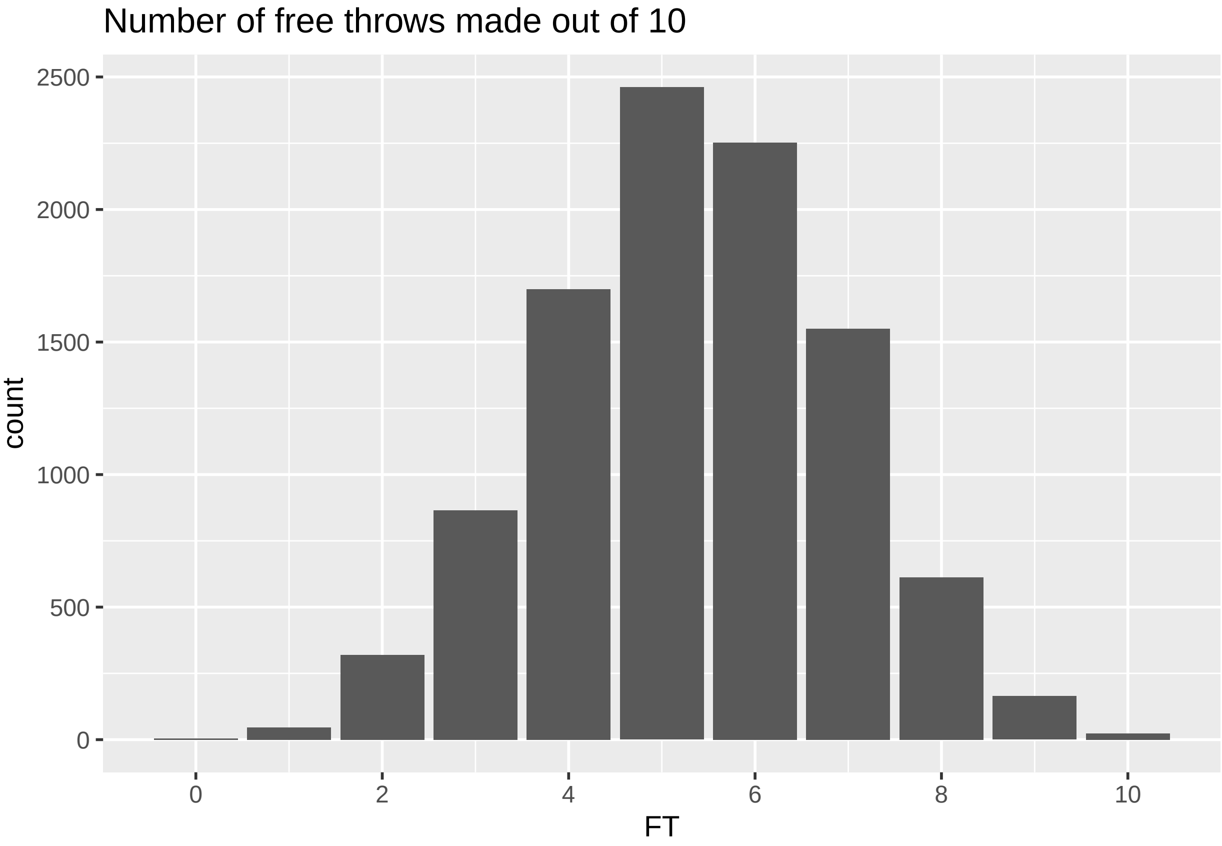
```
## [1] 0.0023
```

```
prob10 <- mean(FT == 10); prob10
```

```
## [1] 0.0023
```

The estimated probability that Shaq goes 10-for-10 in free throw attempts based on his career average is 0.0023.

```
FT %>%
  as.data.frame() %>%
  ggplot(aes(x=FT)) +
  geom_bar() +
  ggtitle("Number of free throws made out of 10") +
  scale_x_continuous(breaks = seq(0, 10, by = 2))
```



If we run the simulation again with a different seed, we will get another estimate (0.0019).

```
set.seed(1)

# Number of Simulations
n.sims <- 10000

# Initialize FT variable with 10000 zeros
FT <- rep(0,n.sims)

for(i in 1:n.sims){
  # Simulate 10 free throws
  temp <- sample(x=c(0,1), size = 10, replace = T, prob = c(0.473,0.527) )

  # Count the number of free throws made and store them in FT
  FT[i] <- sum(temp)
}

(prob10 <- mean(FT == 10))
```

```
## [1] 0.0019
```

As we increase the number of simulations, the estimate will become more accurate.

```
set.seed(1)

# Number of Simulations
n.sims <- 100000

# Initialize FT variable with 10000 zeros
FT <- rep(0,n.sims)

for(i in 1:n.sims){
  # Simulate 10 free throws
  temp <- sample(x=c(0,1), size = 10, replace = T, prob = c(0.473,0.527) )

  # Count the number of free throws made and store them in FT
  FT[i] <- sum(temp)
}

(prob10 <- mean(FT==10))
```

```
## [1] 0.00174
```

One way to simulate data is to make assumptions about the distributions of the underlying data. The random variables given in the last chapter are possible candidates.

4.2 Estimating Probabilities

We can use simulation to estimate probabilities of different events occurring. One way to do this is for each simulation to record a "1" if the event of interest occurs and a "0" if the event of interest does not occur.

Definition 4.1. The ***indicator function***, $I(A)$, is defined such that $I(A)$ is equal to 1 if A occurs and is equal to 0 if A does not occur.

For instance, suppose we roll a die and a "6" is on top. Then we have the following: $I(6) = 1, I(5) = 0, I(even) = 1, I(odd) = 0$.

One way to calculate probabilities is to use the following rule: $P(A) = E[I(A)]$. The probability that A occurs is equal to the expected value of the indicator function of A.

Example 4.3. During the 2021 WNBA season, Kahleah Copper of the Chicago Sky had a free throw percentage of 81.8%. She played a total of 32 games and the probability mass function for number of free throw attempts per game are given in the table below.

Estimate the probability that Copper did not make a free throw in a game.

Note: Copper did not make a free throw in 6 out of the 32 games for a probability of 0.1875.

FTA	nFTA	p(FTA)
0	5	0.156
1	2	0.062
2	8	0.250
3	0	0.000
4	7	0.219
5	2	0.062
6	4	0.125
7	2	0.062
8	2	0.062

```
set.seed(2020)
n.sims <- 10000
games <- 32
FTprob <- 0.818
FTA <- 0:8
nFTA <- c(5,2,8,0,7,2,4,2,2)
pFTA <- nFTA/32
FT <- rep(0, n.sims)
FT0.ind <- rep(0,n.sims)
```

```
# Simulate the number of FTA per game
FTA.sim <- sample(x = FTA,size = n.sims,replace = T,prob = pFTA)

# Simulate 10,000 games and record number of FT made
for(i in 1:n.sims){
  FT[i] <- rbinom(n=1,size = FTA.sim[i],prob = FTprob)
}

# Look at the header of the simulated data
head(FT)
```

```
## [1] 6 3 0 0 1 1
```

```
# Create indicator function for 0 FT made
FT0.ind = FT == 0
head(FT0.ind)
```

```
## [1] FALSE FALSE  TRUE  TRUE FALSE FALSE
```

```
# Calculate the probability via mean of the indicator function
mean(FT0.ind)
```

```
## [1] 0.1711
```

Example 4.4. The number of regulation goals scored in a game by Hockey Team A, X, is a Poisson(4) random variable, and the same for Hockey Team B, Y, is a Poisson(3.2) random variable.

A statistician is interested in the probability that Team A defeats Team B in regulation. This is $P(X > Y)$ which is difficult to calculate manually. However, using simulation, we can straightforwardly obtain an accurate estimation of this quantity.

There are many built-in functions in `R` that allow users to generate realizations from common probability distributions (`rnorm`, `rbinom`, `rexp`, etc.) Let's use the `rpois` function to simulate the appropriate variables, remembering to set a seed so that our results are easily replicable.

```
set.seed(2022)
n.sims <- 10000

team_A_goals <- rpois(n = n.sims, lambda = 4)
team_B_goals <- rpois(n = n.sims, lambda = 3.2)
```

Now, to find $P(X > Y)$, we can use the following line of code:

```
mean(team_A_goals > team_B_goals)
```

```
## [1] 0.5415
```

Why does this work? First, operations to vectors are executed elementwise, meaning that `R` compares `team_A_goals[1]` to `team_B_goals[1]`, then `team_A_goals[2]` to `team_B_goals[2]`, and so on. Second, logical operators are stored as zeroes (when the condition is false) and ones (when the condition is true). The mean of a vector of zeroes and ones is the proportion of ones, which is the frequency of the logical statement being true. In our simulation, it was 0.5415. The true value is 0.5427, meaning that the simulation was quite accurate.

Example 4.5. In baseball, hitting for the cycle requires a hitter to get a single, double, triple, and home run in the same game. This is a rare occurrence in professional baseball having happened only 339 times at present count.

On August 10, 2009, Colorado Rockie Troy Tulowitzki hit for the cycle against the Cubs at Coors Field in Denver going 5-for-5 with two singles, one double, one triple, and one home run. Here's a video recap: https://www.youtube.com/watch?v=sTU6ice3ga0

Simulate 100,000 games (5 at-bats per game) for Tulowitzki based on his career numbers and use them to estimate the probability that Tulowitzki hits for the cycle.

Tulowitzki's career totals are: 4804 at-bats, 878 singles, 264 doubles, 24 triples, and 225 home runs.

```r
set.seed(2022)
n.sims <- 100000
n.ab <- 5
cycle.ind <- rep(0,n.sims)

# Possible outcomes: 0 = out/walk, 1 = single, 2 = double, 3 = triple, 4 = HR
x <- 0:4
px <- c(3413,878,264,24,225)/4804
tulo <- data.frame(x,px)

for( i in 1:n.sims){
  game <- sample(x = tulo$x, prob = tulo$px, size = n.ab, replace = T)
  cycle <- (1 %in% game) & (2 %in% game) & (3 %in% game) & (4 %in% game)
  if( cycle ){
    cycle.ind[i] <- 1
  }
}

mean(cycle.ind)
```

```
## [1] 0.00024
```

4.3 Simulating Streaks

Streaks are often of interest to casual sports fans. Some especially famous streaks include Joe DiMaggio's 56-game hitting streak in 1941, Wayne Gretzky's 51 consecutive games with a point in 1983-1984, and the Chicago Cubs 108 year World Series drought.

Simulation can be helpful in quantifying the likelihood of different kinds of streaks like winning streaks or hitting streaks.

4.3.1 Winning Streak Simulation

Example 4.6. Suppose an NBA team is in the middle of a rebuild and has a 25% probability of winning each of its games in the following 82-game season. What is the probability that the team will go on at least one winning streak of four or more games over the course of the 82-game season? Use simulation to answer this question.

We can simulate a season for the team, find the longest winning streak in that season, and store it in a vector. After repeating that process 10,000 times, we can then find the proportion of the values in that vector that are greater than or equal to 4.

```r
set.seed(2022)
n.sims <- 10000
n.games <- 82
win.prob <- 0.25
longest_streak <- rep(NA, n.sims)

for (i in 1:n.sims) {
  game_results <- rbinom(size = 1, n = n.games, prob = win.prob) # 1=win, 0=loss
  streaks <- rle(game_results)
  longest_streak[i] <- max(streaks$lengths[streaks$values==1])
}

table(longest_streak)
```

```
## longest_streak
##    1    2    3    4    5    6    7    8    9
##  116 3626 4233 1480  410  105   21    7    2
```

```r
mean(longest_streak >= 4)
```

```
## [1] 0.2025
```

The team had a 4+ game winning streak in about 20% of the simulations.

4.3.2 Hitting Streak Simulation

In 1941, New York Yankee Joe DiMaggio had a 56-game hitting streak which is an all-time record in MLB. How unlikely was such an outcome?

Background videos on DiMaggio's 56 game hitting streak:

https://www.youtube.com/watch?v=Y5K49dtOKmo

https://www.youtube.com/embed/BErlc16YS8A

Example 4.7. Let's build a simulation to estimate the probability of a hitting streak of at least 56 games using DiMaggio's statistics. DiMaggio's 1941 game log is contained in `dimaggio41.csv`.

```
dimaggio <- read_csv("data/dimaggio41.csv", col_names = TRUE)

names(dimaggio)
```

```
##  [1] "Rk"   "Gtm"  "Date" "Opp"  "Rslt" "PA"   "AB"   "R"    "H"    "2B"
## [11] "3B"   "HR"   "RBI"  "BB"   "IBB"  "SO"   "HBP"  "SH"   "SF"   "ROE"
## [21] "GDP"  "SB"   "CS"   "BA"   "OBP"  "SLG"  "OPS"  "BOP"  "aLI"  "WPA"
## [31] "acLI" "cWPA" "RE24" "Pos"
```

```
nrow(dimaggio)
```

```
## [1] 140
```

```
dimaggio %>% select(1:13) %>% slice(1:10,139:140) %>% kt()
```

Rk	Gtm	Date	Opp	Rslt	PA	AB	R	H	2B	3B	HR	RBI
1	1	Apr 14	WSH	W3-0	4	4	0	2	0	1	0	1
2	2	Apr 15	PHA	L1-3	4	4	1	2	1	0	0	0
3	3	Apr 16	PHA	L7-10	5	5	1	4	2	0	1	2
4	4	Apr 17	PHA	W9-4	5	4	2	2	0	0	0	0
5	5	Apr 18	WSH	L4-7	4	4	1	1	0	0	0	1
6	6	Apr 19	WSH	W5-2	5	5	1	1	0	0	1	2
7	7	Apr 20	PHA	W19-5	6	5	4	3	0	0	1	6
8	8	Apr 21	PHA	W14-4	6	5	3	4	1	0	1	2
9	9	Apr 22	PHA	L5-6	4	3	1	0	0	0	0	0
10	10	Apr 23	BOS	W4-2	5	4	0	0	0	0	0	1
139	156	Sep 28	WSH	L0-5	4	4	0	1	1	0	0	0
NA	NA	NA	NA	90-47	622	541	122	193	43	11	30	125

DiMaggio played in 139 games, had 622 plate appearances, 541 at-bats, and 193 hits.

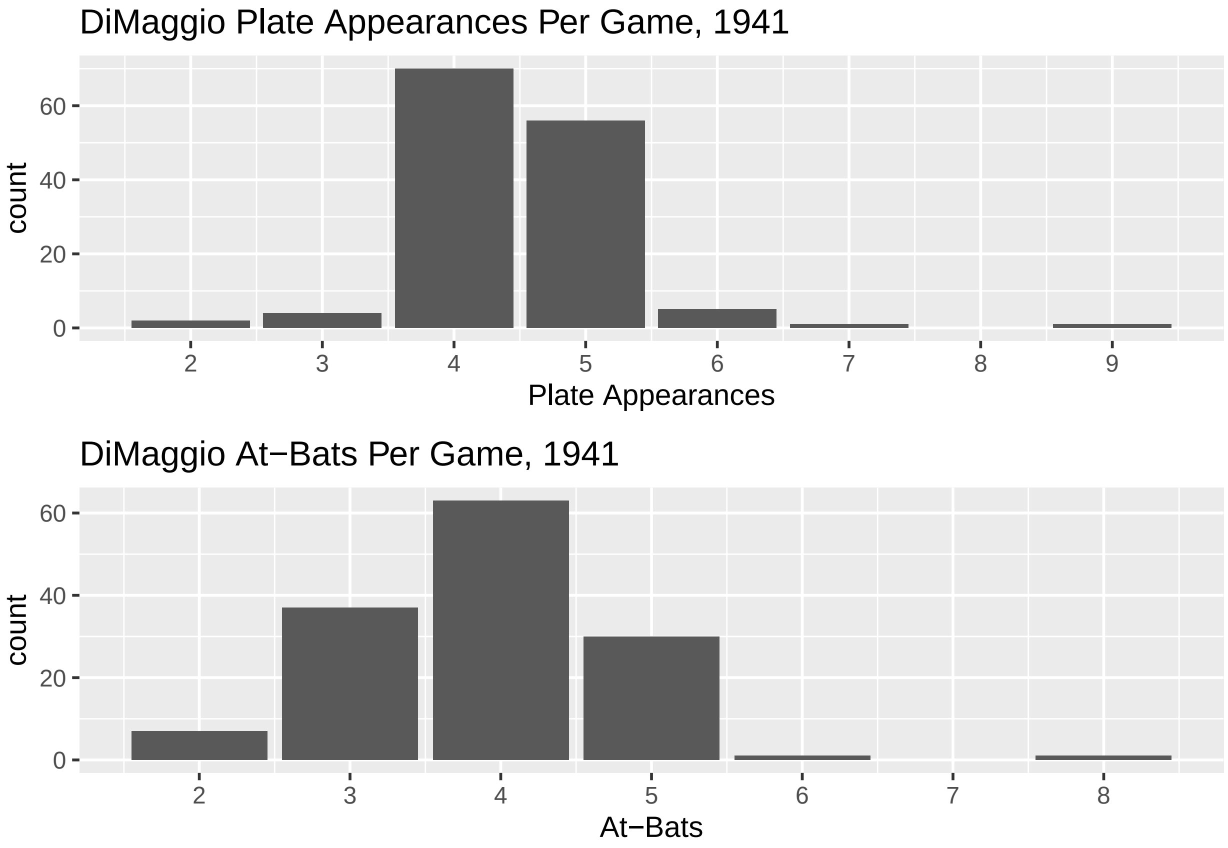
```r
# remove last row (totals)
dimaggio <- dimaggio %>% slice(1:139)

# Create indicator variable for a hit
hit.game <- ifelse(dimaggio$H > 0,1,0)

# Use rle to calculate the streak lengths
streaks <- rle(hit.game)
table(streaks) %>% kt()
```

	0	1
1	5	2
2	4	3
3	4	2
4	0	2
5	0	1
7	0	1
8	0	1
16	0	1
56	0	1

As seen above, DiMaggio had a 56-game hitting streak. An impossible feat to match?

(a) Create a histogram for DiMaggio's per game plate appearances and at bats. (Hint: for discrete values, `geom_bar()` is often a good option.)

```
library(gridExtra)

p1 <- dimaggio %>%
  ggplot(aes(x=PA)) +
  geom_bar() +
  scale_x_continuous(breaks=0:10) +
  ggtitle("DiMaggio Plate Appearances Per Game, 1941") +
  xlab("Plate Appearances")
p2 <- dimaggio %>%
  ggplot(aes(x=AB)) +
  geom_bar() +
  scale_x_continuous(breaks=0:10) +
  ggtitle("DiMaggio At-Bats Per Game, 1941") +
  xlab("At-Bats")

grid.arrange(p1, p2, ncol = 1)
```



(b) Create a frequency and percentage frequency table for plate appearances and at-bats.

```
library(janitor)
table.pa = tabyl(dimaggio,PA) %>%
  adorn_totals("row") %>%
  adorn_pct_formatting(digits = 1)
names(table.pa) = c("Plate Appearances", "Frequency", "Percent")
table.pa %>% kt()
```

Plate Appearances	Frequency	Percent
2	2	1.4%
3	4	2.9%
4	70	50.4%
5	56	40.3%
6	5	3.6%
7	1	0.7%
9	1	0.7%
Total	139	100.0%

```
table.ab = tabyl(dimaggio,AB) %>%
  adorn_totals("row") %>%
  adorn_pct_formatting(digits = 1)
names(table.ab) = c("At-Bats", "Frequency", "Percent")
table.ab %>% kt()
```

At-Bats	Frequency	Percent
2	7	5.0%
3	37	26.6%
4	63	45.3%
5	30	21.6%
6	1	0.7%
8	1	0.7%
Total	139	100.0%

(c) DiMaggio had 193 hits in 622 plate appearances over 139 games. We will simulate DiMaggio's season of 139 games 100,000 times to estimate the probability of a 56-game hitting streak.

There are many ways to do this. Let's use the *empirical probability mass function* of his per game plate appearances to simulate the number of plate appearances that he gets in his 139 games.

```
pa <- tabyl(dimaggio,PA) %>% select(1,2)
pa <- pa %>%
  as.data.frame() %>%
  mutate(Prob=n/139)
pa %>% kt()
```

PA	n	Prob
2	2	0.014
3	4	0.029
4	70	0.504
5	56	0.403
6	5	0.036
7	1	0.007
9	1	0.007

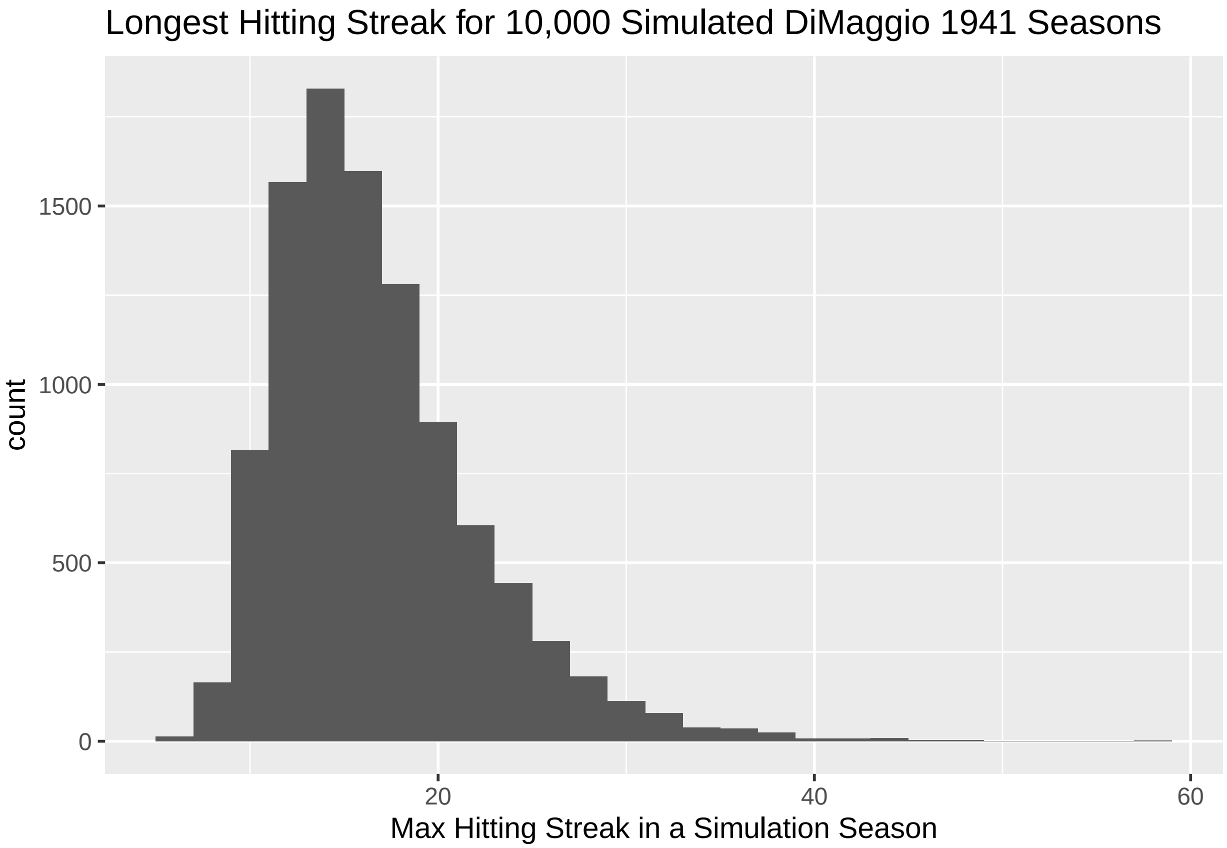
```
# One simulated season of per game plate appearances
sim.pa <- sample(x=pa$PA,prob = pa$Prob,size=139,replace=T)
sim.pa
```

```
##   [1] 4 4 4 9 4 4 4 5 2 5 5 5 4 5 4 4 4 4 4 4 5 5 4 5 9 7 4 4 5 5 5 5 5 5 6 4 5
##  [38] 4 5 4 5 5 4 4 5 4 5 7 2 5 4 5 4 4 4 4 4 5 4 4 4 5 6 4 4 5 4 4 5 4 5 6 4 5
##  [75] 4 4 4 4 5 4 5 4 5 4 5 3 5 5 3 5 4 2 4 5 4 4 4 4 5 5 4 5 5 4 5 5 4 5 5 5 4
## [112] 5 4 4 4 5 4 5 4 4 4 4 5 5 4 4 5 5 4 5 4 4 4 4 5 4 4 5 4
```

```
# DiMaggio Simulation
set.seed(2022)
n.sims <- 10000
n.games <- 139
prob.hit <- 0.310

longest.streak <- rep(0, n.sims)
sim.games <- rep(0,n.games)

for( i in 1: n.sims){
  sim.pa <- sample(x=pa$PA,prob = pa$Prob,size=n.games,replace=T)
  for( j in 1:n.games){
    sim.games[j] <- rbinom(n = 1,size = sim.pa[j],prob = prob.hit)
  }
  sim.hits <- ifelse(sim.games > 0,1,0)
  streaks <- rle(sim.hits)
  longest.streak[i] <- max(streaks$lengths[streaks$values==1])
}

# table of longest streaks during simulated seasons
table(longest.streak)
```

```
## longest.streak
##   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
##   3  11  38 127 306 510 756 811 908 920 807 790 685 595 474 421 333 272 258 186
##  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45
## 142 139 100  81  68  45  51  28  22  16  13  23  14  11   7   1   5   3   4   5
##  46  47  48  49  50  56  58  59
##   2   2   2   1   1   1   1   1
```

```
longest.streak %>%
  as.data.frame() %>%
  ggplot(aes(x=longest.streak)) +
  geom_histogram(binwidth=2) +
  ggtitle("Longest Hitting Streak for 10,000 Simulated DiMaggio 1941 Seasons") +
  xlab("Max Hitting Streak in a Simulation Season")
```



```
# statistics for 10,000 simulations
(mean(longest.streak))
```

```
## [1] 17.2849
```

```
(mean(longest.streak>=56))
```

```
## [1] 3e-04
```

Running the simulation above with `set.seed(2022)` and `n.sims=10000`, we get $P(Streak \geq 66) = 3 \cdot 10^{-4}$. There were three simulated hitting streaks of at least 56 games.

If we run the simulation again with `set.seed(2022)` but increase `n.sims=100000`, we get $P(Streak \geq 66) = \frac{10}{100000} = 10^{-4}$. In other words, we estimate the probability that DiMaggio gets a hitting streak of at least 56 games in 100000 simulated seasons is about 1-in-10000.

```
# statistics for 100,000 simulations
(mean(longest.streak))
```

```
## [1] 17.1655
```

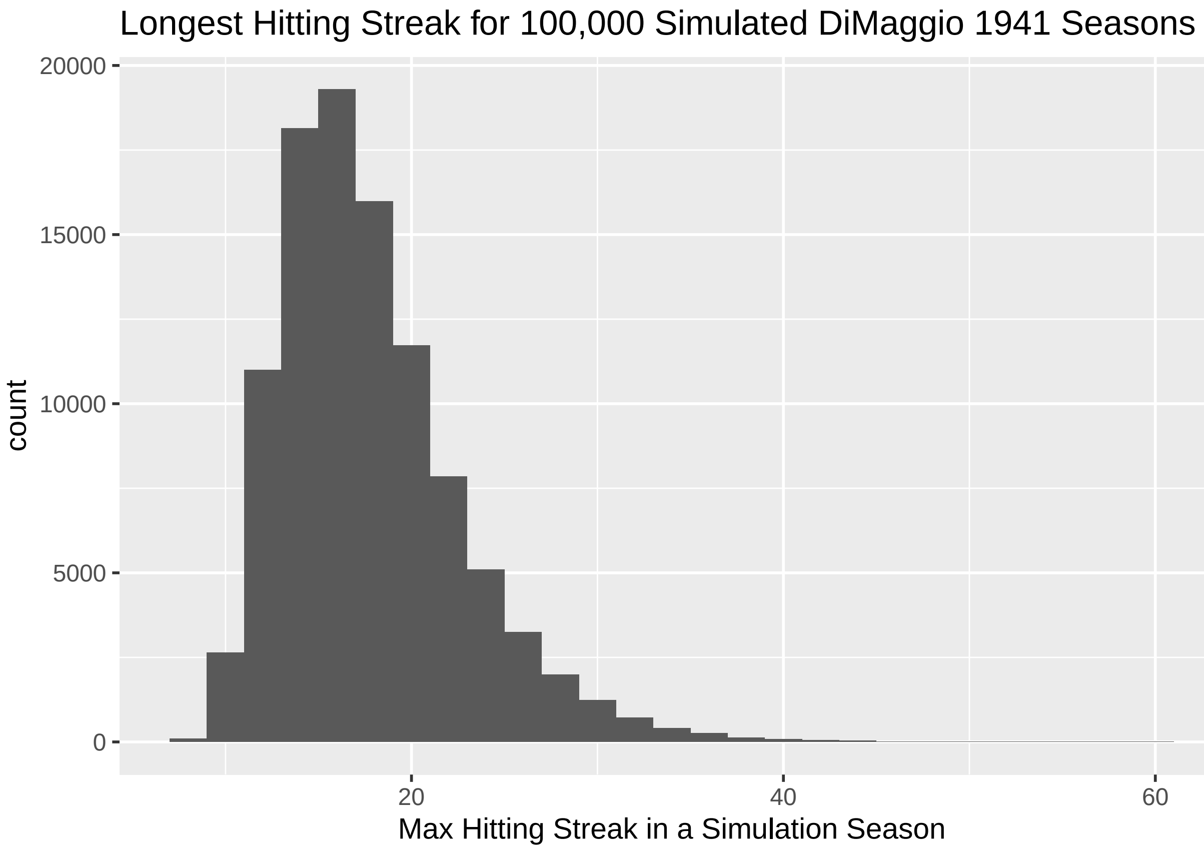
```
(mean(longest.streak>=56))
```

```
## [1] 1e-04
```

We would prefer to not use nested for loops, as they are slow. Can you find a faster simulation method?

Let's try simulating by permuting DiMaggio's games. In other words, let's randomly order DiMaggio's 1941 games and analyze the hitting streaks.

```
# Create indicator variable for a hit
hit.game <- ifelse(dimaggio$H > 0,1,0)

set.seed(2022)
n.sims <- 100000
longest.streak <- rep(0, n.sims)

for( i in 1: n.sims){
  sim.hits <- sample(x=hit.game,replace=F)
  streaks <- rle(sim.hits)
  longest.streak[i] <- max(streaks$lengths[streaks$values==1])
}

longest.streak %>%
  as.data.frame() %>%
  ggplot(aes(x=longest.streak)) +
  geom_histogram(binwidth=2) +
  ggtitle("Longest Hitting Streak for 100,000 Simulated DiMaggio 1941 Seasons") +
  xlab("Max Hitting Streak in a Simulation Season")
```



```
# statistics for 100,000 simulations
(mean(longest.streak))
```

```
## [1] 18.25769
```

```
(mean(longest.streak>=56))
```

```
## [1] 7e-05
```

Using this simulation method with reordering, there is an estimated probability of $7 \cdot 10^{-5}$ or about 1-in-14000 chance of DiMaggio hitting a 56-game hit streak.

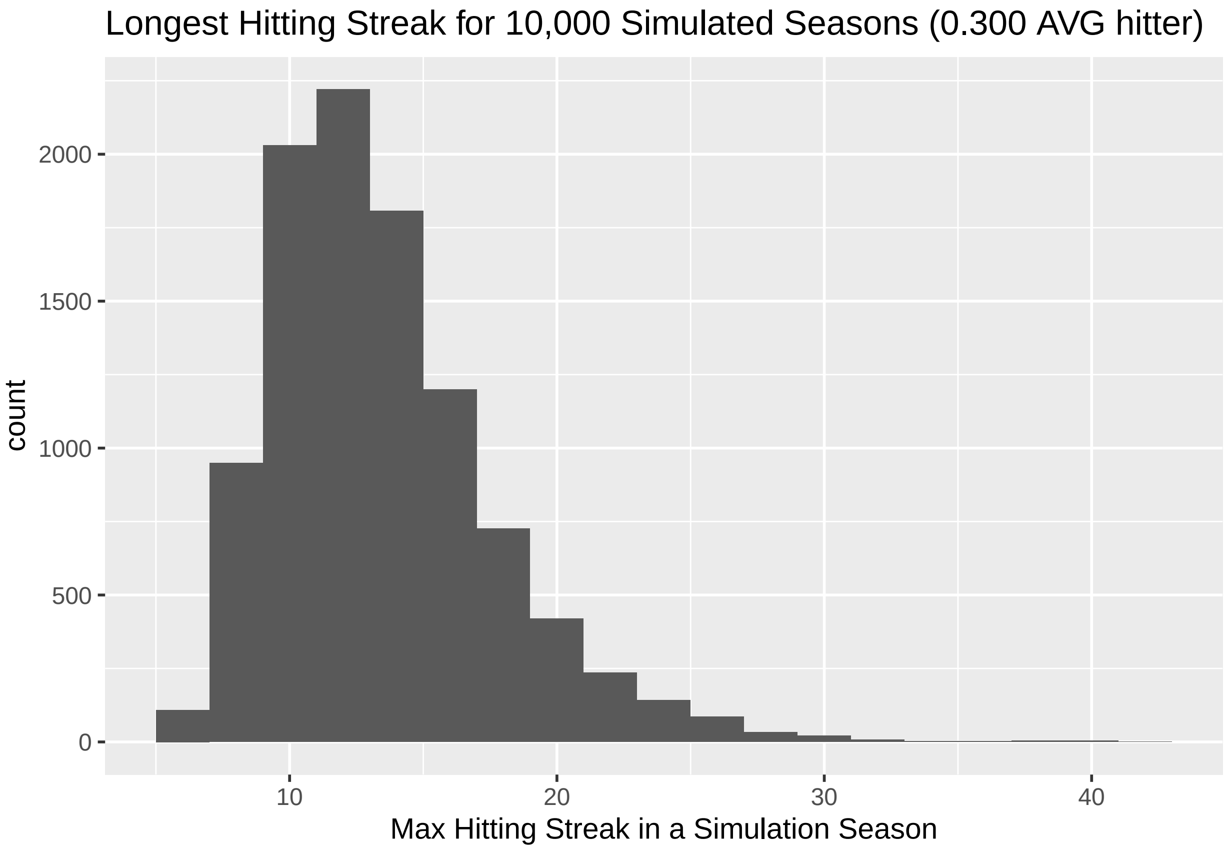
Other authors have used different simulation and mathematical methods for estimating the rarity of Dimaggio's 56 game hitting streak.

Billie et al (2010) used an at-bat rather than plate appearance simulation and estimated the likelihood as 1-in-5000.

Rothman et al (2010) estimated the likelihood as 1-in-10000.

Example 4.8. Let's consider a more general simulation for hitting streaks. Suppose a hitter with a 0.300 batting average plays 140 games and has an equal likelihood of 3, 4, or 5 at-bats in a game. Simulate 10,000 seasons for this hitter, calculate the average max hitting streak, and the overall max hitting streak. After this, repeat the process for a hitter with a 0.400 batting average.

```r
# 0.300 Hitter Hitting Streak Simulation
set.seed(2022)
n.sims <- 10000
n.games <- 140
prob.hit <- 0.300

longest.streak <- rep(0, n.sims)
sim.games <- rep(0,n.games)

for( i in 1: n.sims){
  sim.pa <- sample(x=3:5,size=n.games,replace=T)
  for( j in 1:n.games){
    sim.games[j] <- rbinom(n = 1,size = sim.pa[j],prob = prob.hit)
  }
  sim.hits <- ifelse(sim.games > 0,1,0)
  streaks <- rle(sim.hits)
  longest.streak[i] <- max(streaks$lengths[streaks$values==1])
}

longest.streak %>%
  as.data.frame() %>%
  ggplot(aes(x=longest.streak)) +
  geom_histogram(binwidth=2) +
  ggtitle("Longest Hitting Streak for 10,000 Simulated Seasons (0.300 AVG
  hitter)") +
  xlab("Max Hitting Streak in a Simulation Season")
```

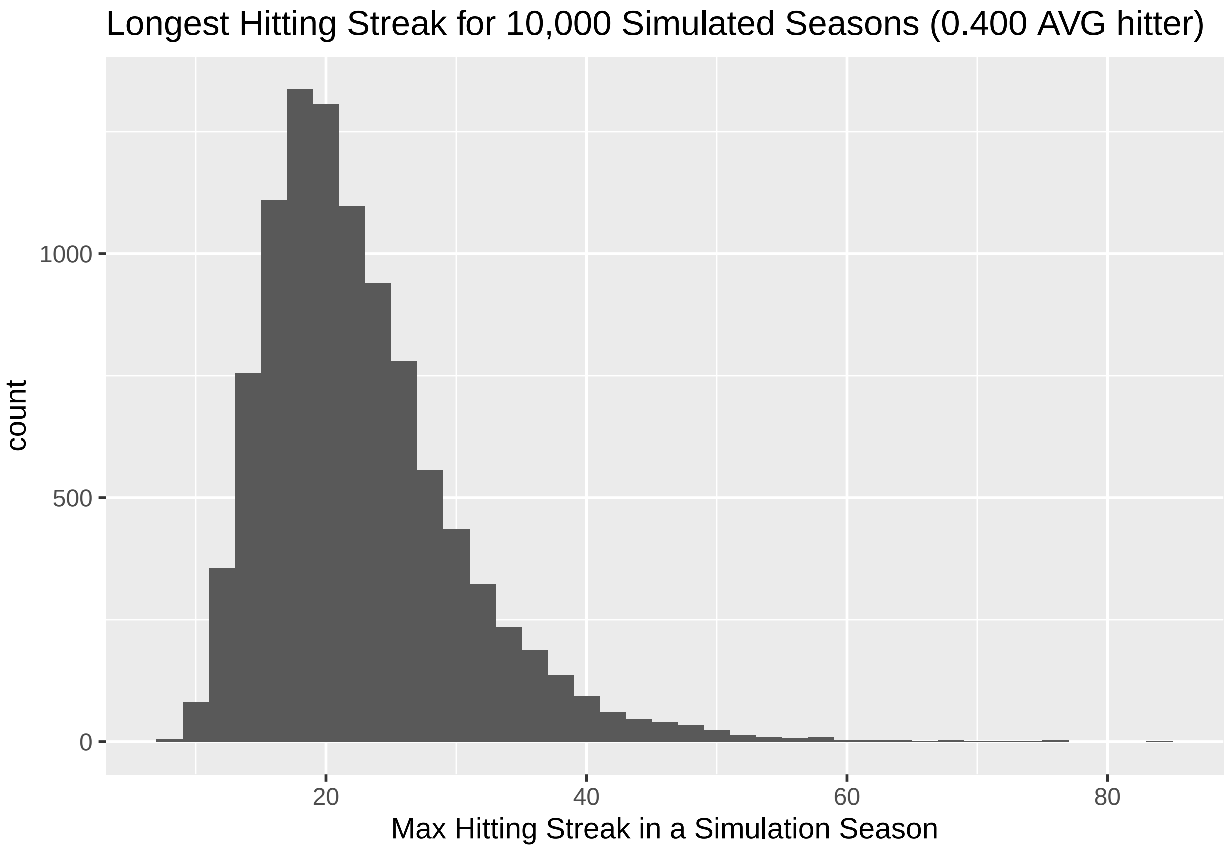


```
# statistics for 10,000 simulations (0.300 batting average)
(mean(longest.streak))
```

```
## [1] 13.9355
```

```
(max(longest.streak))
```

```
## [1] 42
```



```
# statistics for 10,000 simulations (0.400 batting average)
(mean(longest.streak))
```

```
## [1] 23.1349
```

```
(max(longest.streak))
```

```
## [1] 85
```

4.4 Gambling Simulations

Simulation can be very helpful and useful in evaluating betting systems in (sports) gambling.

4.4.1 Martingale System

One such famous betting system is called a martingale system. Under this system, the bettor makes an initial wager. If they lose, they make the same wager. If they lose again, they follow a *double or nothing* procedure. This means that the bettor will continue betting the amount they have lost until they eventually win and get back even. What is wrong with this system?

Example 4.9. Suppose you are betting on sports matches on the spread where there is no house advantage, so all bets are 1:1 and each team is equally likely to win. You begin by wagering $1 and you lose the initial bet (call it *Bet Zero*). You decide to employee the martingale system. Calculate the expected number of bets that you would have to make to break even. Also, create a histogram for the the biggest deficits in the simulations.

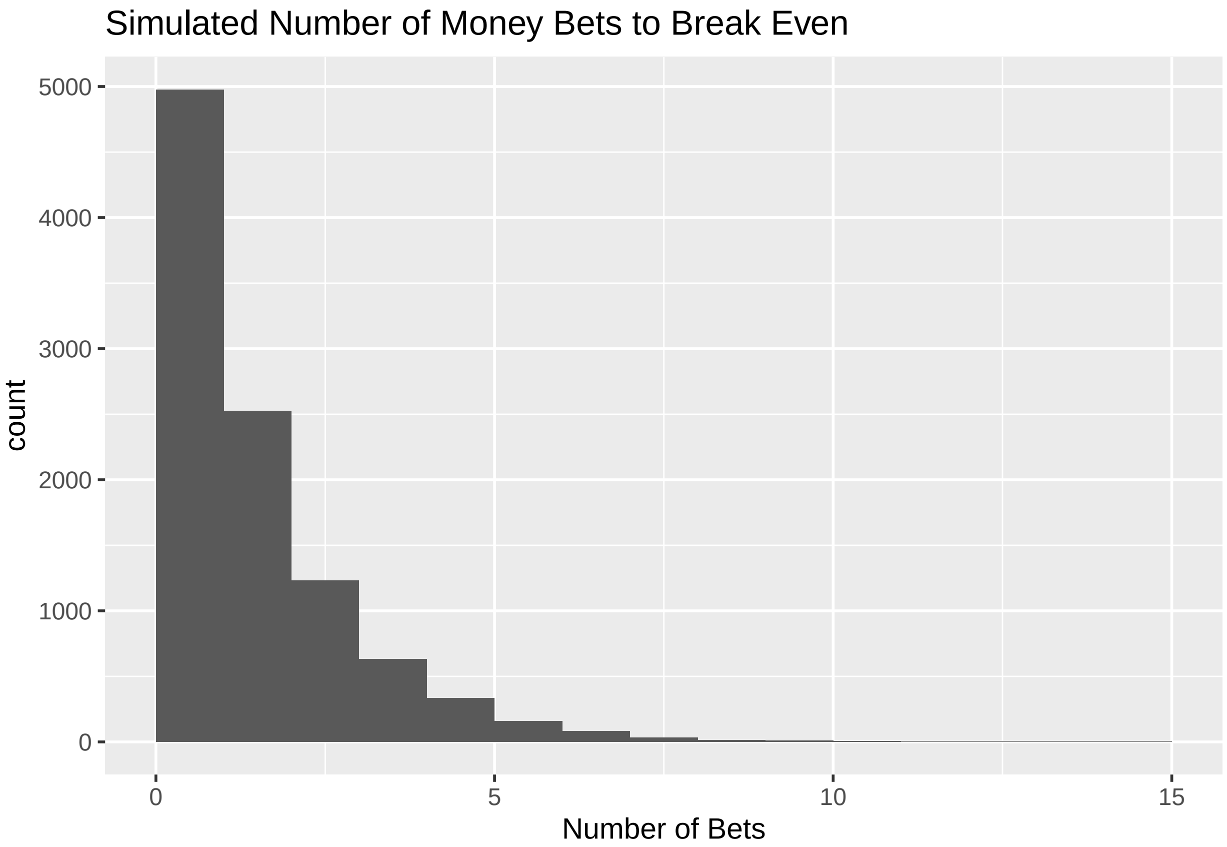
```
set.seed(2022)
n.sims <- 10000
sim.bets <- rep(NA,n.sims)
max.deficit <- rep(NA,n.sims)

for( i in 1:n.sims ){
  bets <- 1

  while( 1 ){
    if( rbernoulli(n=1,0.5) ){
      bets <- bets + 1
    } else {
      break;
    }
  }
  sim.bets[i] <- bets
}

mean(sim.bets)
```

```
## [1] 1.9996
```

```
max(sim.bets)
```

```
## [1] 12
```

```
sim.bets %>% as.data.frame() %>%
  ggplot(aes(x=sim.bets)) +
  geom_histogram(breaks=0:15) +
  ggtitle("Simulated Number of Money Bets to Break Even") +
  xlab("Number of Bets")
```

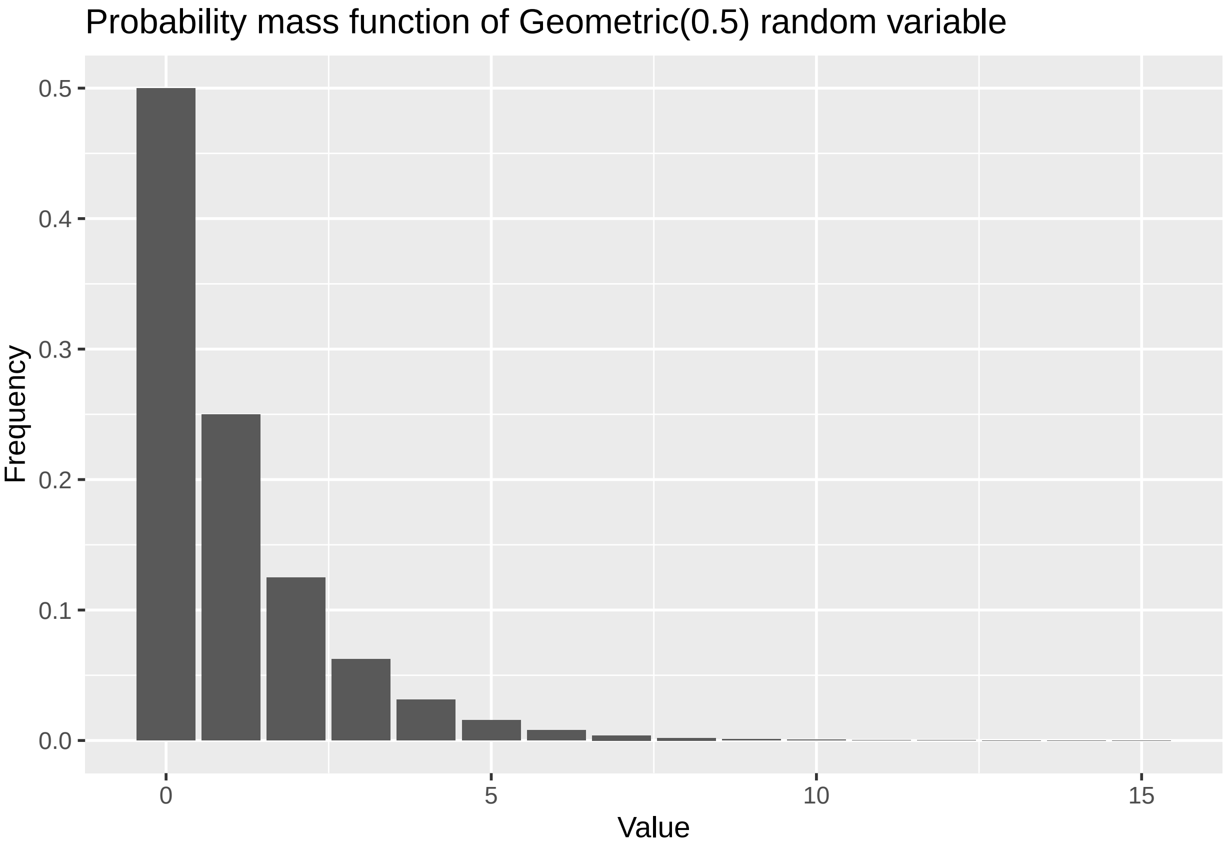


On average, two bets were needed to be placed after *Bet Zero* to break even.

In two simulations (out of 10,000 total simulations), 12 bets were needed to be made to break even after the initial failed bet, *Bet Zero*. The first bet after *Bet Zero* was for \$1. The second failed bet was from \$2. The twelfth bet was for $\$2^{11} = \$2048$. If your total bankroll was \$1000, you wouldn't have been even able to make the twelfth bet to break even. The weakness of the martingale system is that you are only guaranteed (with probability 1) to break even if you have an infinite bankroll.

The distribution in this example should look familiar. It is the geometric distribution. Let's plot the probability mass function for a Geometric(0.5) random variable and calculate the probability that 12 or more bets are required to break even.

```
ggplot(transform(data.frame(x=c(0:15)), y=dgeom(x, prob = 0.5)), aes(x, y)) +
  geom_bar(stat="identity") +
  ggtitle("Probability mass function of Geometric(0.5) random variable") +
  labs(x="Value", y="Frequency", )
```



```
# recall that R counts the number of failures before a success,
# so we will be looking for at least 11 failures
pgeom(q=11,prob=0.5,lower.tail=F)
```

```
## [1] 0.0002441406
```

There is a 0.02% chance that we will need to make 12 or more wagers. In other words, if your bankroll is \$1000, there is a 0.02% you lose the full bankroll before you can break even.

4.4.2 Gambler's Ruin

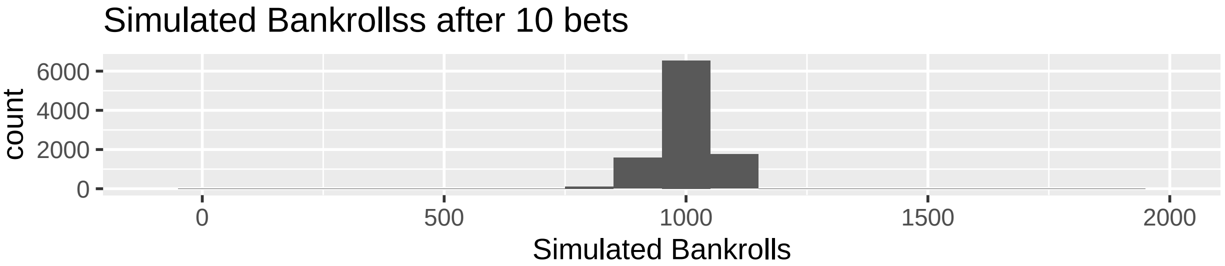
The previous example is an ideal case where there is no house advantage. Sportsbooks will always have a house advantage, so it is more practical to consider an example where there is house advantage. For example, a point spread bet at -110 for two evenly matched teams will offer the sportsbook a house advantage.

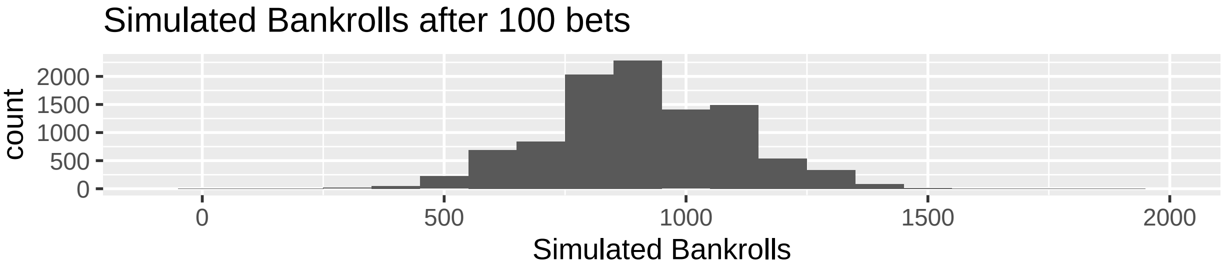
Example 4.10. Suppose you have a bankroll of $1000 and place $20 bets on point spread bets at -110. You will stop if you ever have less than $20. Assume that both teams are equally likely to beat the spread. How long will your bankroll last? Calculate your expected bankroll after 10 bets, 100 bets, and 1000 bets.

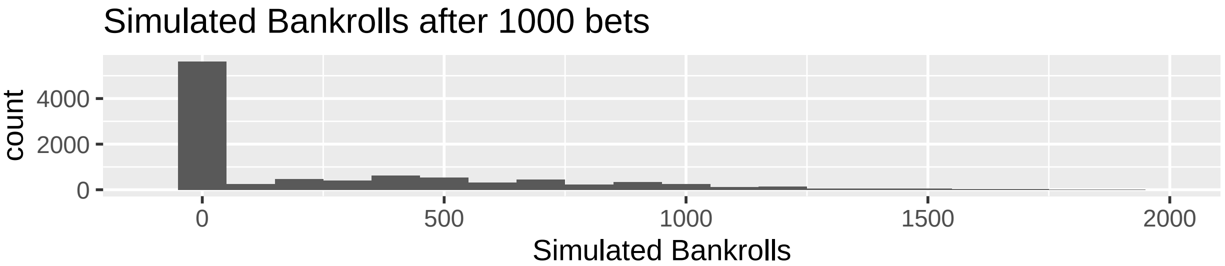
We'll use the following code and run simulations with total.bets = 10, 100, 1000, 10000.

```
set.seed(2022)
n.sims <- 10000
bankroll.end <- rep(0,n.sims)

# betting info
bankroll.start <- 1000
wager <- 20
ml <- -110
total.bets <- 10
implied.prob <- (-ml)/(-ml+100)
dec.odds <- 1/implied.prob
profit <- wager * (dec.odds - 1)

for(i in 1:n.sims){
  bets <- 0
  bankroll <- bankroll.start
  while(bankroll >= 20 && bets < total.bets){
    bets <- bets + 1
    if( rbernoulli(n=1,0.5) ){
      bankroll <- bankroll + profit
    } else
      bankroll <- bankroll - 20
  }
  bankroll.end[i] <- bankroll
}

mean.ending.bankroll <- mean(bankroll.end)
max.ending.bankroll <- max(bankroll.end)
data.frame(mean.ending.bankroll,max.ending.bankroll) %>% kt()

bankroll.end %>% as.data.frame() %>%
  ggplot(aes(x=bankroll.end)) +
  geom_histogram(binwidth=100) +
  ggtitle("Simulated Bankroll (starting at $1000) after 10 bets") +
  xlab("Bankroll after 10 bets")
```

Number of Bets	Mean Ending Bankroll	Max Ending Payroll
10	991.707	1181.818
100	909.599	1672.727
1000	274.529	2572.727


Simulated Bankrollss after 10 bets


Simulated Bankrolls after 100 bets


Simulated Bankrolls after 1000 bets

All bettors eventually go broke under this scenario due to the house advantage. This is called the concept of **Gambler's Ruin**. A gambler playing a game with negative expected value will eventually go broke, regardless of their betting system.

See Wikipedia for further details: https://en.wikipedia.org/wiki/Gambler's_ruin

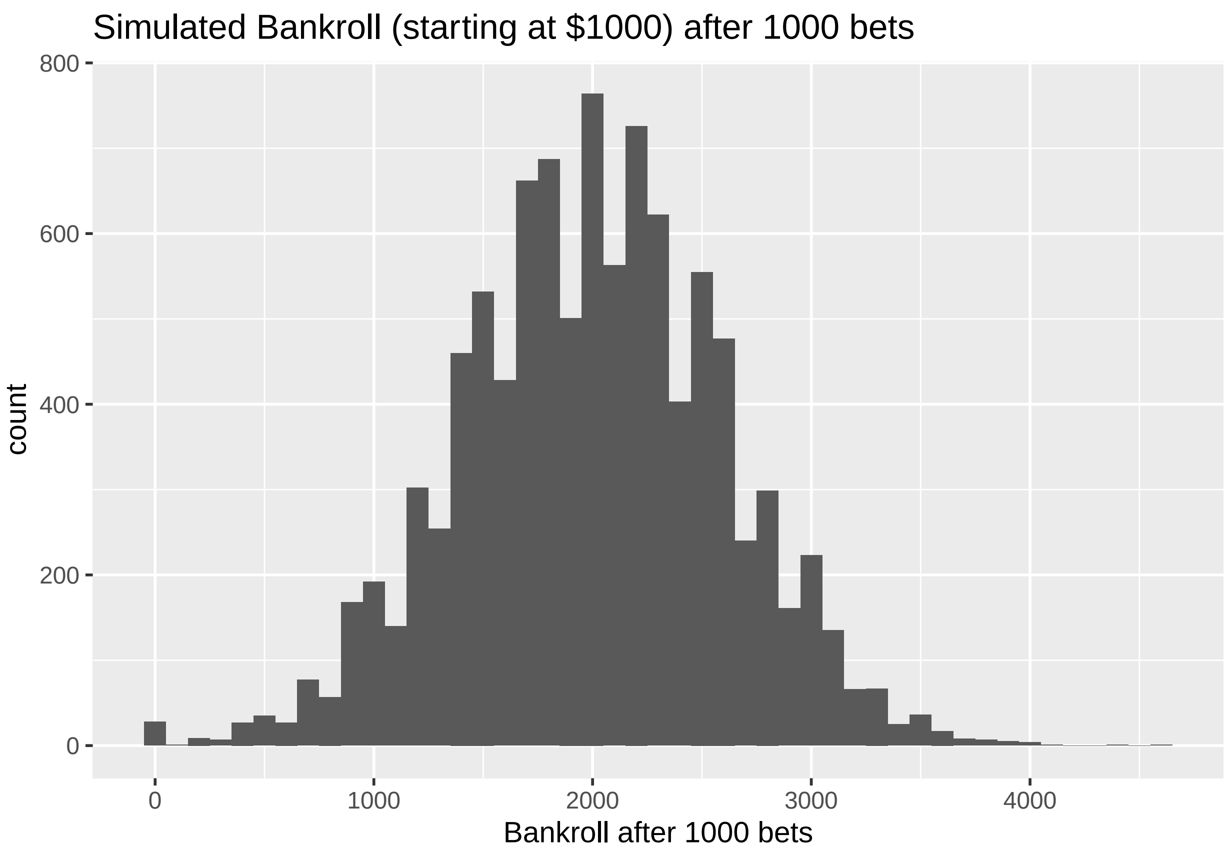
4.4.3 Simulating a Winning System

Example 4.11. Suppose your betting system allows you pick winners on the spread 55% of the time. If you are betting at -110, then your probability exceeds the implied probability of 52.4% and this system is not guaranteed to be an eventual loser. Repeat the simulation above with the new probability of winning a wager and 1,000 total bets.

```
set.seed(2022)
n.sims <- 10000
bankroll.end <- rep(0,n.sims)

# betting info
bankroll.start <- 1000
wager <- 20
ml <- -110
total.bets <- 1000
implied.prob <- (-ml)/(-ml+100)
dec.odds <- 1/implied.prob
profit <- wager * (dec.odds - 1)

for(i in 1:n.sims){
  bets <- 0
  bankroll <- bankroll.start
  while(bankroll >= 20 && bets < total.bets){
    bets <- bets + 1
    if( rbernoulli(n=1,0.55) ){
      bankroll <- bankroll + profit
    } else
      bankroll <- bankroll - 20
  }
  bankroll.end[i] <- bankroll
}

mean.bankroll <- mean(bankroll.end)
max.bankroll <- max(bankroll.end)
median.bankroll <- median(bankroll.end)
prob.zero <- mean(bankroll.end <= 20)
data.frame(prob.zero,mean.bankroll,median.bankroll,max.bankroll) %>% kt()
```

prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.003	1996.283	2000	4634.545

```
bankroll.end %>% as.data.frame() %>%
  ggplot(aes(x=bankroll.end)) +
  geom_histogram(binwidth=100) +
  ggtitle("Simulated Bankroll (starting at $1000) after 1000 bets") +
  xlab("Bankroll after 1000 bets")
```


Simulated Bankroll (starting at $1000) after 1000 bets
count
800
600
400
200
0
0
1000
2000
3000
4000
Bankroll after 1000 bets

4.4.4 Using the Kelly Growth Criterion

Example 4.12. Suppose your betting system allows you pick winners on the spread 55% of the time. If you are betting at -110, then your probability exceeds the implied probability of 52.4% and this system is not guaranteed to be an eventual loser.

The Kelly Growth Criterion is a formula for an optimal size of a bet when the expected returns are known. The formula is as follows:

$$f = \frac{p}{LoseMult} - \frac{q}{WinMult}$$

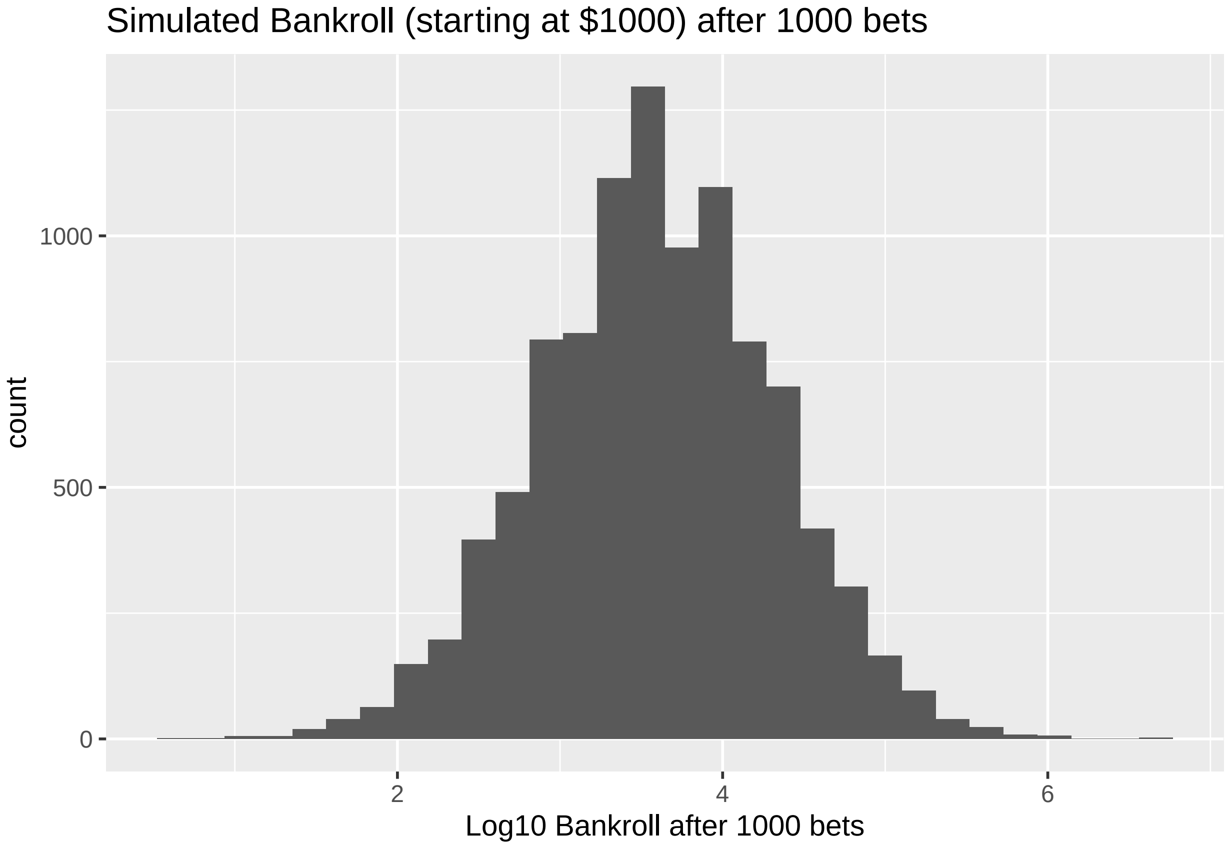
Calculate f and use this value in the previous simulation. How does the results of this simulation differ from the previous fixed bet simulation?

```r
set.seed(2022)
n.sims <- 10000
bankroll.end <- rep(0,n.sims)

# betting info
bankroll.start <- 1000
ml <- -110
total.bets <- 1000
implied.prob <- (-ml)/(-ml+100)
dec.odds <- 1/implied.prob
profit <- wager * (dec.odds - 1)
f = 0.055

for(i in 1:n.sims){
  bets <- 0
  bankroll <- bankroll.start
  while(bankroll > 0 && bets < total.bets){
    wager <- bankroll * f
    bets <- bets + 1
    if( rbernoulli(n=1,0.55) ){
      bankroll <- bankroll + wager*10/11
    } else
      bankroll <- bankroll - wager
  }
  bankroll.end[i] <- bankroll
}

mean.bankroll <- mean(bankroll.end)
max.bankroll <- max(bankroll.end)
```

```
median.bankroll <- median(bankroll.end)
prob.zero <- mean(bankroll.end <= 20)
data.frame(f,prob.zero,mean.bankroll,median.bankroll,max.bankroll) %>% kt()
```

f	prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.055	0.001	15989.77	3966.691	5697642

```
bankroll.end %>% as.data.frame() %>%
  ggplot(aes(x=log10(bankroll.end))) +
  geom_histogram(bins=30) +
  ggtitle("Simulated Bankroll (starting at $1000) after 1000 bets") +
  xlab("Log10 Bankroll after 1000 bets")
```



Example 4.13. Repeat the previous simulation for $f = 0.01, f = 0.1, f = 0.3$

f	prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.01	0	1646.572	1575.793	5886.551

f	prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.055	0.001	15989.77	3966.691	5697642

f	prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.1	0.079	261172.5	1558.97	906807883

f	prob.zero	mean.bankroll	median.bankroll	max.bankroll
0.3	0.995	84832.93	0	650931393

4.5 Simulating Censored Data

In 1994, major league baseball players went on strike and the season ended prematurely. Most teams played about 115 games out of the scheduled 162 game season. What might have happened in the remainder of the season?

There are two individual statistics are particularly of interest.

1) Tony Gwynn of the San Diego Padres was batting 0.394 when the strike occurred. The last time that a player hit 0.400 in a season was 1941 for MLB (Ted Williams, 0.406) and 1948 for the Negro League (Artie Wilson, 0.435).
2) Matt Williams of the San Francisco Giants had 43 home runs when the strike occurred. At the time, the single season home run record was 61 by Roger Maris of the New York Yankees in 1961.

We will use simulation to estimate the probability that these players would match these records.

4.5.1 Tony Gwynn 1994 Batting Average Simulation

The San Diego Padres had played 117 games when the strike ended the season in 1994. Tony Gwynn had a batting average of 0.394, 475 plate appearances, 419 at-bats, and 165 hits over 110 games at that point. Assume that Gwynn would play in all remaining games for the season.

Video: https://www.youtube.com/watch?v=irr-JblLOZM

(a) How many at-bats was Gwynn expected to have in the remainder of the season?

```
n.season = 162
n.partial = 117
n.hits = 165
(n.remain = n.season - n.partial)
```

```
## [1] 45
```

```
ab = 419
ab.remain = ab/n.partial*n.remain
ab.remain = round(ab.remain); ab.remain
```

```
## [1] 161
```

Gwynn was expected to have about 161 more at-bats in the remainder of the 45 cancelled games.

(b) Assuming that Gwynn's hitting ability stayed constant for the remainder of the season, estimate the probability that his average would exceed 0.400.

```
avg = 0.394
n.sims = 10000
sim.hits = rbinom(n=n.sims,size=ab.remain,prob=avg)
sim.hits.total = n.hits + sim.hits
sim.avg = sim.hits.total/(ab+ab.remain)
mean(sim.avg >= 0.400)
```

```
## [1] 0.3118
```

For more discussion on this issue, see: https://www.complex.com/sports/2016/07/tony-gwynn-would-have-hit-400-1994

4.5.2 Matt Williams 1994 Home Run Simulation

The San Francisco Giants had played 115 games when the strike ended the season in 1994. Matt Williams had hit 43 homeruns in 483 plate appearances in 112 games at that point. Assume that Williams would play in all remaining games for the season.

Video: https://www.youtube.com/watch?v=wVkBu---RKw

(a) How many plate appearances was Williams expected to have in the remainder of the season?

```
n.season = 162
n.partial = 115
n.hr = 43
(n.remain = n.season - n.partial)
```

```
## [1] 47
```

```
ab = 483
ab.remain = ab/n.partial*n.remain
ab.remain = round(ab.remain); ab.remain
```

```
## [1] 197
```

Williams was expected to have about 197 more plate appearances in the remainder of the 47 cancelled games. He would need to hit 18 home runs to match Maris and 19 or more to beat Maris.

(b) Estimate the probability that Matt Williams would have beaten Maris' single season home run record.

```
hr.avg = n.hr/ab
n.sims = 10000
sim.hr = rbinom(n=n.sims,size=ab.remain,prob=hr.avg)
sim.hr.total = n.hr + sim.hr
mean(sim.hr.total >= 62)
```

```
## [1] 0.4
```

For more discussion on this issue, see: https://www.mlb.com/news/matt-williams-1994-home-run-chase

Chapter 5

Inferential Statistics

5.1 Defining a Population

For team and individual statistics in sports, we often split data up by the seasons. For statistical inference, we need to define what the population is and what the sample is. It is acceptable to define a season as the population and a subset of the season as a sample, but this is usually not exactly what we want to do.

Instead, we often think of a season as a random sample from a theoretical population of all possible seasons. In the same manner, we can think of individual games are a random sample from a theoretical population of all possible games.

5.2 Statistical Inference

Definition 5.1. *Statistical Inference* is the process of inferring properties of a population by use of sample data.

For statistical inference, we need a point estimator and a measure of uncertainty. We will focus on statistical inference for population means and population proportions.

We will examine two methods of Statistical Inference:

1. Confidence Intervals
2. Hypothesis Tests

For sports data, we often want to infer properties of a player's (or team's) underlying ability on a game or season level.

5.2.1 Confidence Intervals

Definition 5.2. A **confidence interval** gives a range of plausible values for a population parameter. Confidence intervals can be one-sided or two-sided.

Different Types of Confidence Intervals

There are three different types of hypothesis tests.

1. Left-tailed confidence interval
2. Right-tailed confidence interval
3. Two-tailed confidence interval

How to Interpret a confidence interval

5.2.2 Hypothesis Tests

The goal of a hypothesis test is to test competing claims about a population parameter. In other words, we want to determine if the sample data are consistent with the null hypothesis which is our initial assumption regarding the population parameter of interest.

Definition 5.3. The *null hypothesis*, denoted H_0, is the claim that is initially assumed to be true.

Definition 5.4. The *alternative hypothesis*, denoted H_a, is the assertion contradictory to H_0 ("the opposite of H_0").

The null hypothesis will be rejected in favor of the alternative hypothesis if the sample evidence suggests that H_0 is false.

The two possible conclusions of a hypothesis test are *Reject* H_0 or *Fail to Reject* H_0.

Rules for formulating hypotheses

1. H_0 contains "="

2. H_a contains $\neq$, $>$, $<$

3. Usually, the claim we are attempting to show is more plausible is H_a

Different Types of Hypothesis Tests

There are three different types of hypothesis tests.

1. Left-tailed test

2. Right-tailed test

3. Two-tailed test

Errors in Hypothesis Testing

After collecting a sample, we will decide which hypothesis is supported by the data.

Since we don't know the true parameter value, there is a chance we will make an error.

Two Types of Errors in Hypothesis Testing

1. Type I error (α): Reject H_0 when H_0 is true.

2. Type II error (β): Fail to reject H_0 when H_0 is false.

Note: α is also called the level of significance for a test.

Six Steps for Hypothesis Testing

Different textbooks will give a different number of steps or outline on how to complete a hypothesis test. For our class, stick to the following method.

Step 1: State the hypotheses.

Step 2: Determine the level of significance.

Step 3: Compute a test statistic.

Step 4: Calculate a p-value.

Step 5: Make a statistical decision.

Step 6: Interpret the statistical decision (in the context of the problem).

5.3 Inference for One Population Parameter

Theorem 5.1. *A random sample from an infinite population with mean μ and variance σ^2 has the following properties:*

$E[\bar{X}] = \mu$

$Var(\bar{X}) = \frac{\sigma^2}{n}$

$\hat{SE}(\bar{X}) = \frac{s}{\sqrt{n}}$

5.3.1 Confidence Interval for Population Mean

Definition 5.5. The confidence interval for a population mean μ with sample mean $\bar{x}$ and sample variance s^2 is given by:

$\bar{x} \pm t_{c.v.} \frac{s}{\sqrt{n}}$, $df = n - 1$,

where $t_{c.v.}$ denotes the critical value for a t-distribution with $df = n - 1$ degrees of freedom for the required confidence level.

Example 5.1. In Michael Jordan's third season, 1986–1987, he had a career best with an average of 37.1 points per game (standard deviation: 9.92) while playing an average of 40.0 minutes per game over 82 games. Data for Jordan's 1986–1987 season are given in `jordan86-87.csv`.

Video highlights from Michael Jordan's 1986–1987 season:
https://www.youtube.com/watch?v=cWSrUhhR3DA

Let μ be the true point scoring ability of Michael Jordan during his 1986–1987 season. Create a 90% confidence interval for μ.

```
jordan86_87 <- read_csv("data/jordan86-87.csv")

jordan86_87 %>% summarize(
  Games = n(),
  `Points Mean`= mean(PTS),
  `Points SD` = sd(PTS),
  `Minutes Mean` = mean(MP),
  `Minutes SD` = sd(MP)) %>%
  kable(booktabs=T,digits = 2)
```

Games	Points Mean	Points SD	Minutes Mean	Minutes SD
82	37.09	9.92	40.01	4.44

```
library(infer)
jordan86_87 %>% t_test(response=PTS) %>%
  kable(booktabs=T,digits = 2)
```

statistic	t_df	p_value	alternative	estimate	lower_ci	upper_ci
33.84	81	0	two.sided	37.09	34.9	39.27

5.3.2 Hypothesis Test for Population Mean

Definition 5.6. The test statistic for a population mean μ with sample mean $\bar{x}$ and sample variance s^2 is given by:

$t_{test} = \frac{\bar{x}-\mu}{s/\sqrt{n}} \sim t_{n-1}$

Example 5.2. Wilt Chamberlain is considered one of the greatest of all-time in basketball and he had a career year in the 1961–1962 season. Data for Chamberlain's 1961–1962 season are given in `wilt61-62.csv`.

On March 2, 1962, Wilt Chamberlain scored 100 points in a single game. He also averaged 50.4 points per game (standard deviation: 12.0 points) while playing an average of 48.5 minutes per game over 80 games.

Background videos on Wilt Chamberlain:
https://www.youtube.com/watch?v=LxMeEzhvNRs
https://www.youtube.com/watch?v=1WsCtyLGg1w

Let μ be Wilt's true point scoring ability in 1961–1962. Test the claim that $\mu > 50$ PPG at $\alpha = 0.05$.

```
wilt61_62 <- read_csv("data/wilt61-62.csv")

wilt61_62 %>% summarize(
  Games = n(),
  `Points Mean`= mean(PTS),
  `Points SD` = sd(PTS),
  `Minutes Mean` = mean(MP),
  `Minutes SD` = sd(MP)) %>%
  kable(booktabs=T,digits = 2)
```

Games	Points Mean	Points SD	Minutes Mean	Minutes SD
80	50.36	11.99	48.52	2.5

```
wilt61_62 %>% t_test(response=PTS,mu=50,alternative = "greater") %>%
  kable(booktabs=T,digits = 2)
```

statistic	t_df	p_value	alternative	estimate	lower_ci	upper_ci
0.27	79	0.39	greater	50.36	48.13	Inf

5.3.3 Test for Difference in Means

Theorem 5.2. *The test statistic for a difference in population means $\mu_1 - \mu_2$ is given by:*

$t_{test} = \frac{\bar{x}_1 - \bar{x}_2 - D_0}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \sim t_\nu$

where $\nu = \frac{\left(s_1^2/n_1 + s_2^2/n_2\right)^2}{(s_1^2/n_1)^2/(n_1-1) + (s_2^2/n_2)^2/(n_2-1)}$

5.3.4 Confidence Interval for Difference in Means

Theorem 5.3. *The confidence interval for a difference in population means $\mu_1 - \mu_2$ is given by:*

$(\bar{x}_1 - \bar{x}_2) \pm t_{c.v.}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$

where $t_{c.v.}$ denotes the critical value for a t-distribution with $df = \nu$ degrees of freedom for the required confidence level and $\nu = \frac{\left(s_1^2/n_1 + s_2^2/n_2\right)^2}{(s_1^2/n_1)^2/(n_1-1) + (s_2^2/n_2)^2/(n_2-1)}$

Example 5.3. NBA statistics are often given as a rate statistic per 36 minutes to account for differences in playing time. Compare points per game (PPG) and points per 36 minutes (PP36) on a season level for Jordan's 1986–1987 season and Chamberlain's 1961–1962 season.

```r
wilt_pp36 <- sum(wilt61_62$PTS)/sum(wilt61_62$MP)*36
jordan_pp36 <- sum(jordan86_87$PTS)/sum(jordan86_87$MP)*36
data.frame(`Wilt PPG` = mean(wilt61_62$PTS),
           `Jordan PPG` = mean(jordan86_87$PTS),
           `Wilt PP36` = wilt_pp36,
           `Jordan PP36` = jordan_pp36)  %>%
  kable(booktabs=T,digits = 2)
```

Wilt.PPG	Jordan.PPG	Wilt.PP36	Jordan.PP36
50.36	37.09	37.36	33.37

Example 5.4. Calculate PP36 on a game-by-game basis for Jordan and Chamberlain. Create a 95% confidence interval for the difference in PP36 and make a statistical decision as to if there is a difference.

```
jordan86_87$PP36 <- jordan86_87$PTS/jordan86_87$MP*36
wilt61_62$PP36 <- wilt61_62$PTS/wilt61_62$MP*36
jordan_wilt <- data.frame(PP36=c(jordan86_87$PP36,wilt61_62$PP36),
                          Player=c(rep("MJ",82),rep("WC",80)))
jordan_wilt %>% t_test(formula=PP36~Player,
                       order=c("WC","MJ"),conf_level = 0.95) %>%
  kable(booktabs=T,digits = 2)
```

statistic	t_df	p_value	alternative	estimate	lower_ci	upper_ci
2.84	159.96	0.01	two.sided	3.87	1.18	6.56

Theorem 5.4. *A random sample of size n from an infinite population with mean π has the following properties:*

$E[\hat{p}] = \pi$

$Var(\hat{p}) = \frac{\pi(1-\pi)}{n}$

$\hat{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

5.3.5 Confidence Interval for Population Proportion

Theorem 5.5. *The confidence interval for a population proportion π with sample proportion $\hat{p}$ and sample size n is given by:*

$\hat{p} \pm z_{c.v.}\sqrt{\frac{p(1-p)}{n}}$

where $z_{c.v.}$ denotes the critical value for a z-distribution for the required confidence level.

Example 5.5. x

5.3.6 Test for Population Proportion

Theorem 5.6. *The test statistic for a population proportion π is given by:*

$z_{test} = \frac{\hat{p}-\pi}{\sqrt{\frac{\pi(1-\pi)}{n}}} \sim \mathcal{N}(0,1)$

Example 5.6. AL vs NL

5.3.7 Confidence Interval for Difference in Proportions

Theorem 5.7. *The confidence interval for a difference in population proportions* $\pi_1 - \pi_2$ *is given by:*

$(\hat{p}_1 - \hat{p}_2) \pm z_{c.v.}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$

where $z_{c.v.}$ *denotes the critical value for a z-distribution for the required confidence level.*

Example 5.7. x

5.3.8 Test for Difference in Proportions

Theorem 5.8. *The test statistic for a difference in population proportions $\pi_1 - \pi_2$ is given by:*
$z_{test} = \frac{\hat{p}_1 - \hat{p}_2 - D_0}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}} \sim \mathcal{N}(0,1)$

Example 5.8. x

5.4 Margin of Error Calcuations

5.4.1 MOE for Probabilities

(Shooting percentages, batting averages, save percentage)

5.4.2 MOE for Averages

(PPG, Digs per game, Passer Rating–using bootstrap)

5.4.3 MOE estimation using simulation

(Durant scoring, AM, page 98)

5.5 One Sample and Two Sample t-tests and confidence intervals

5.5.1 Hockey Faceoffs

(Mathletics, chapter 40 page 383)

5.5.2 Hockey Penalty Scoring

(Mathletics, chapter 40, page 13)

5.5.3 NFL Overtime Winners

(Scorecasting, page 192) (Mathletics, chapter 25, page 211)

5.5.4 Icing the Kicker

(Scorecasting, page 211)

5.5.5 Example: Football One vs. Two Point Conversion

(Mathletics, chapter 23, page 194) – discussion of “the chart”

5.5.6 Two Sample Tests

(NBA Players, AM, page 103)

(Comparing NL and AL, page 106)

5.6 Permutation Tests

5.7 Bootstrap

```
library(infer)
observed_statistic <- wilt61_62 %>%
  specify(response = PTS) %>%
  calculate(stat = "mean")
```

```
null_dist_1_sample <- wilt61_62 %>%
  specify(response = PTS) %>%
  hypothesize(null = "point", mu = 50) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

Chapter 6

Correlation

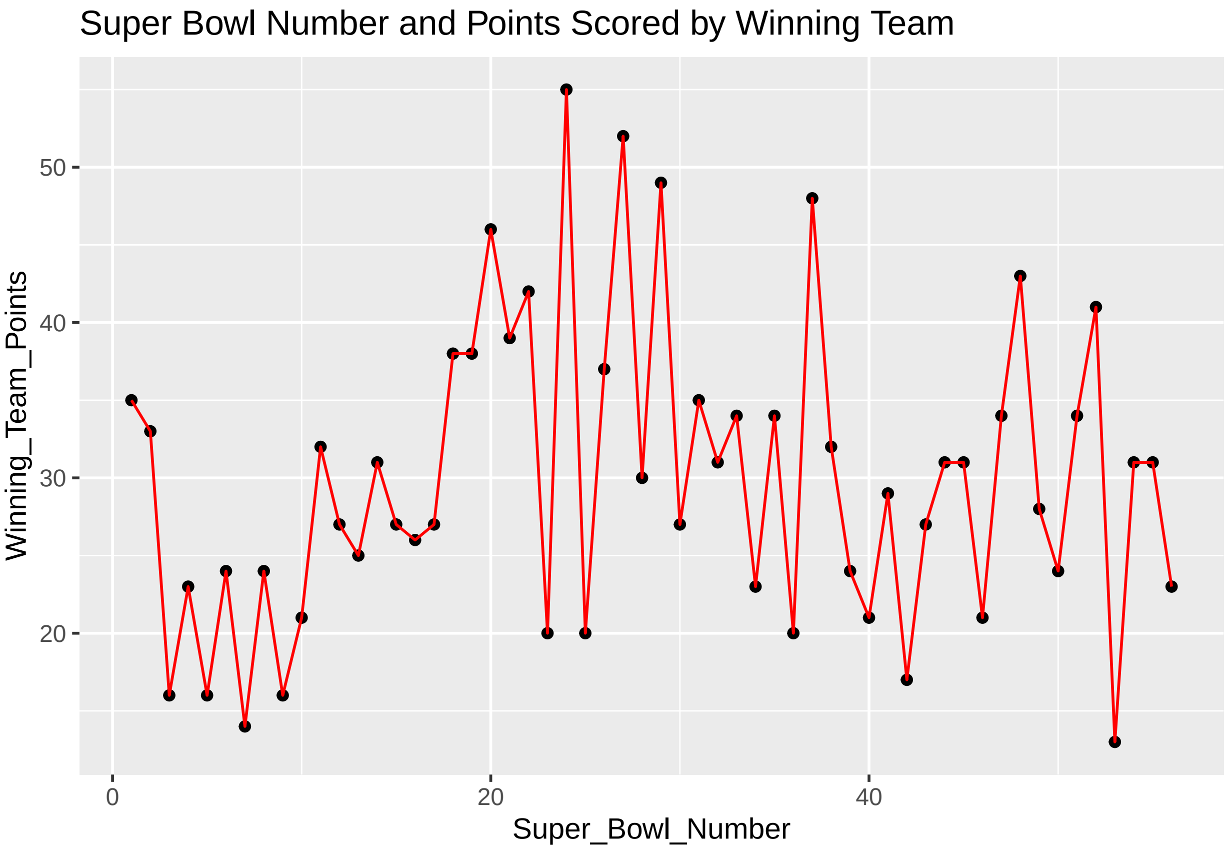
6.1 Pearson's Correlation Coefficient

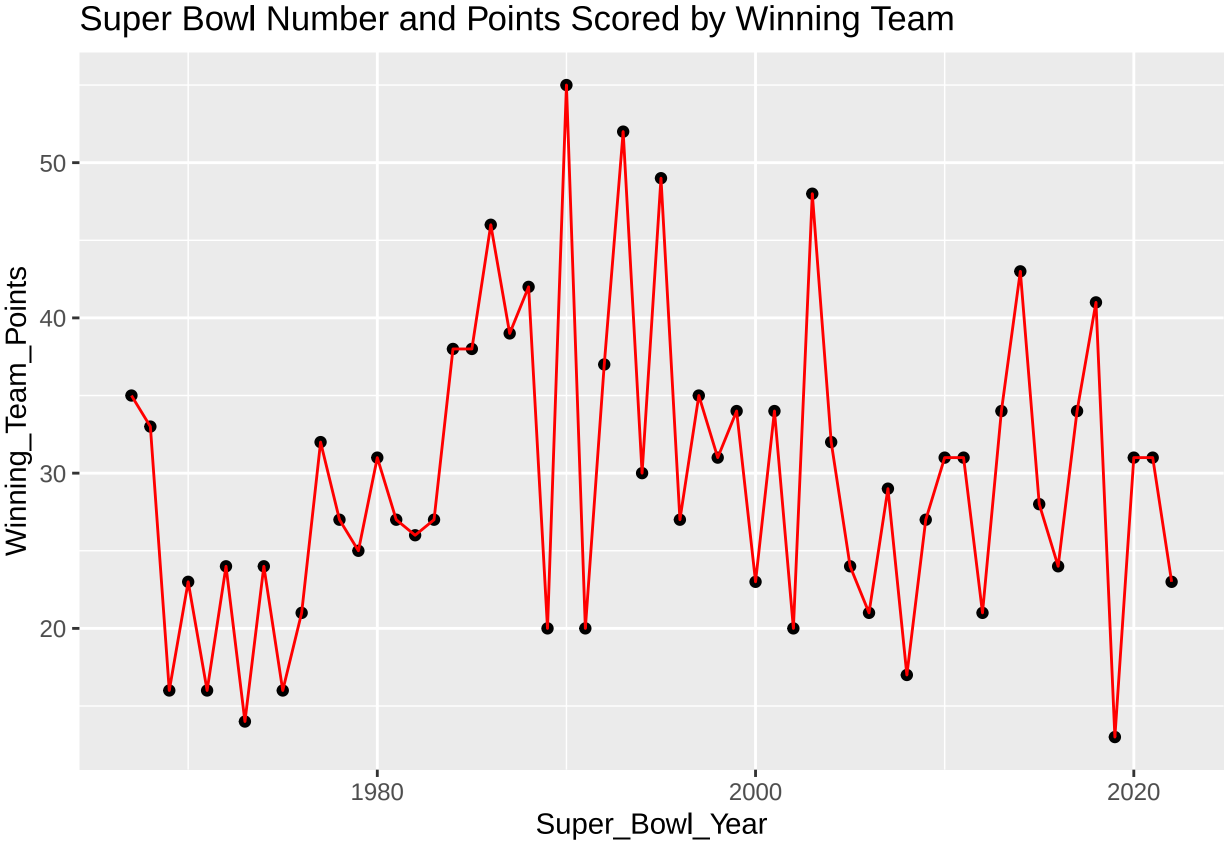
6.1.1 Partial Correlation and Confounding Variables

(AM, page 131, pitchers with more innings)

6.1.2 Properties of the Linear Correlation Coefficient

Pearson's correlation coefficient is unaffected by changes in location (adding a constant) or scale (multiplying by a constant) of data. To illustrate this concept, let's consider the correlation between the number of points scored by the winning team of the Super Bowl and 1) the number of the Super Bowl {1, 2, 3, … , 56} and 2) the year of the Super Bowl {1967, 1968, 1969, … , 2022}.


Super Bowl Number and Points Scored by Winning Team
Winning_Team_Points
50
40
30
20
0
20
40
Super_Bowl_Number


Super Bowl Number and Points Scored by Winning Team
Winning_Team_Points
50
40
30
20
1980
2000
2020
Super_Bowl_Year

```
cor(Super_Bowl_Number, Winning_Team_Points)
```

```
## [1] 0.1063479
```

```
cor(Super_Bowl_Year, Winning_Team_Points)
```

```
## [1] 0.1063479
```

The only difference between the two plots is that the x-axis is shifted. Since the change is essentially a linear (location) transformation, the correlations are identical.

Note: The correlations were equal only because there has been a one-to-one relationship between Super Bowl number and year since 1967. Major League Baseball has had years in which no World Series was held (1904 - boycott, 1994 - players' strike); thus, the same example applied to the MLB would produce slightly different correlation coefficients.

Q: The correlation coefficient is positive. What does this mean in terms of the trend of number of points scored by Super Bowl winners over time?

A: Winners of more recent Super Bowls are more likely to have scored more points than winners of older Super Bowls.

In a similar way to the example above, the correlation between a basketball player's career field goal attempts and three-pointers made and the correlation between the same player's career FGA and points scored from three-point field goals are exactly the same, due to the scale invariance property of correlation.

6.2 Rank Correlation

(Example using team data)

6.3 Autocorrelation

(AM, page 134, NFL team records, pre- and post- salary cap)

6.3.1 Hot Hand

(Scorecasting, page 215)

6.3.2 Streakiness in Sports

(Mathletics, chapter 11, page 105)

6.4 Association of Categorical Variables

(AM, page 143, Yule's Q)

(AM, page 139, winning at halftime vs winning game)

(AM, page 144, Brady TD passes vs sacks)

(AM, page 145, Nadal clay courts)

Chapter 7

Linear Regression

7.1 Simple Linear Regression

(AM, page 162, OPS vs runs scored)

7.1.1 Pythagorean Records

(Mathletics, chapter 1, page 13) (Mathletics, chapter 41, page 385)

7.1.2 Regression to the Mean

(Rushing Yards, Batting Average, Rank in Goals)

7.2 Variable Transformations

7.2.1 Draft Pick Values

(Mathletics, chapter 26, page 222)

7.3 Multiple Linear Regression

7.3.1 Four Factors and Winning

(Mathletics, chapter 28, page 250)

7.3.2 Linear Weights in Baseball

(Mathletics, chapter 3, page 18)

7.3.3 Pass Air Yards Regression

(Mathletics, chapter 27, page 230)

7.3.4 EPL Passing and Scoring

(AM, page 183, 213)

7.3.5 Hockey PPG vs TOI

(AM, page 181, 187, quadratic, exponential models)

7.4 Issues with Multiyear Data

(AM, page 191, correlated errors)

7.5 Interaction

(AM, page 222, strikeout rate vs. movement and velocity)

7.6 Confounding Variables

Example 7.1. A statistician hypothesizes that an NBA team's draft position can be used to predict the number of regular-season wins for the team for the upcoming season. To investigate the relationship, they decide to estimate the following simple linear regression model:

$Wins\ = \beta_0 + \beta_1(Draft\ Position) + \epsilon_i$

Using 2021-22 NBA data, the estimated intercept and slope for the model are:

```
##
## Call:
## lm(formula = Wins_2021_2022 ~ Draft_Position_2021, data = NBA_Data)
##
## Coefficients:
##         (Intercept)  Draft_Position_2021
##             30.2621               0.6928
```

Interpret the estimated slope in this model.

A: For every one-unit increase in draft position, the expected number of wins for the team increases by 0.6928.

After looking at the results of the model and the interpretation for the slope, one might identify that having a higher draft position is associated with more wins in the following season and conclude that teams with high draft picks should trade for lower draft picks in order to have a higher expected number of games won in the following season. What is the flaw in this reasoning?

Recognizing that the SLR model may lead to erroneous lines of logic, the statistician decides to tackle the confounding by adding the wins for the team in the previous season to the model as a covariate. The theoretical model being estimated is now as follows:

$Wins\ = \beta_0 + \beta_1(Draft\ Position) + \beta_2(Previous\ Season\ Wins) + \epsilon_i$

```
##
## Call:
## lm(formula = Wins_2021_2022 ~ Draft_Position_2021 + Wins_2020_2021_Scaled,
##     data = NBA_Data)
##
## Coefficients:
##           (Intercept)    Draft_Position_2021  Wins_2020_2021_Scaled
##               -12.731                 -1.859                  2.013
```

When controlling for an NBA team's wins in the previous season, the sign for the slope coefficient for draft position flips to a negative. This suggests that for teams with the same winning percentage in the previous season, a lower draft position is associated with more wins in the following season. The results of this model, which included an important confounding variable, aligned with the conventional wisdom that lower draft positions are helpful for a team's future success.

Data: https://en.wikipedia.org/wiki/2021_NBA_draft, https://www.nba.com/standings?GroupBy=conf&Season=2020-21&Section=overall

7.7 Logistic Regression

7.7.1 Sack and Interception Probability Models

(Mathletics, chapter 27, page 230) (Mathletics, chapter 1, page 13)

7.7.2 Soccer Goal Logistic Model

(Mathletics, chapter 39, page 354)

7.7.3 Field Goal Success

(AM, page 266)

Chapter 8

Data Scraping and Acquisition

Chapter 9

Principal Component Analysis

Chapter 10

Clustering

Chapter 11

Classification

Chapter 12

Nonparametric Methods

Chapter 13

Sports Drafts

Chapter 14

Baseball

Chapter 15

Football

Chapter 16

Basketball

Chapter 17

Hockey

Chapter 18

Soccer