Sports Analytics

Aaron Nielsen, Department of Statistics, Colorado State University

2022-07-12

Contents

About

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

- STAT 351 (Sports Statistics and Analytics 1)
- STAT 451 (Sports Statistics and Analytics 2

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

- Levi Kipp
- Ellie Martinez
- Isaac Moorman

Chapter 1

Exploratory Data Analysis

1.1 Getting Started With R

1.1.1 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!

1.1.2 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
library(readr)
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>
```

1.2 Descriptive Statistics

1.2.1 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?

1.2.2 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.

1.2.2.1 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.

```
# 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.

Definition 1.15. The ***population standard deviation***, σ, is the standard deviation for an entire population.

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, 0.1)
```

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

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

```
##    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
```

1.2.2.2 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 $x_1, x_2, \ldots, x_n$ 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.

```
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
```

```
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.

```
table(broncos2014)
```

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

1.3 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.

```
# install.packages('tidyverse')

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Let's load the file "NFL_2021_Team_Passing.csv" which contains NFL Team Passing Statistics, 2021

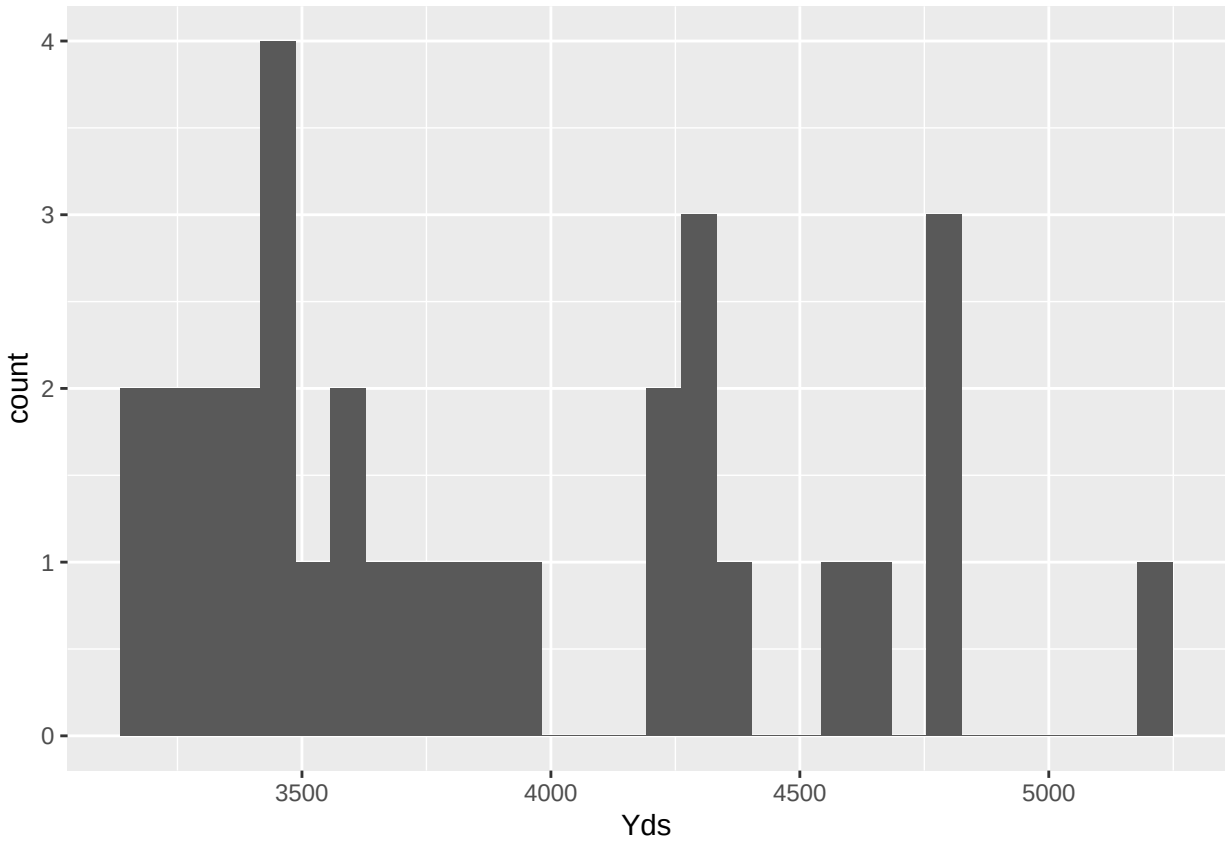
```
library(readr)
NFL_2021_Team_Passing <- read_csv("data/NFL_2021_Team_Passing.csv")
```

1.3.1 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()
```

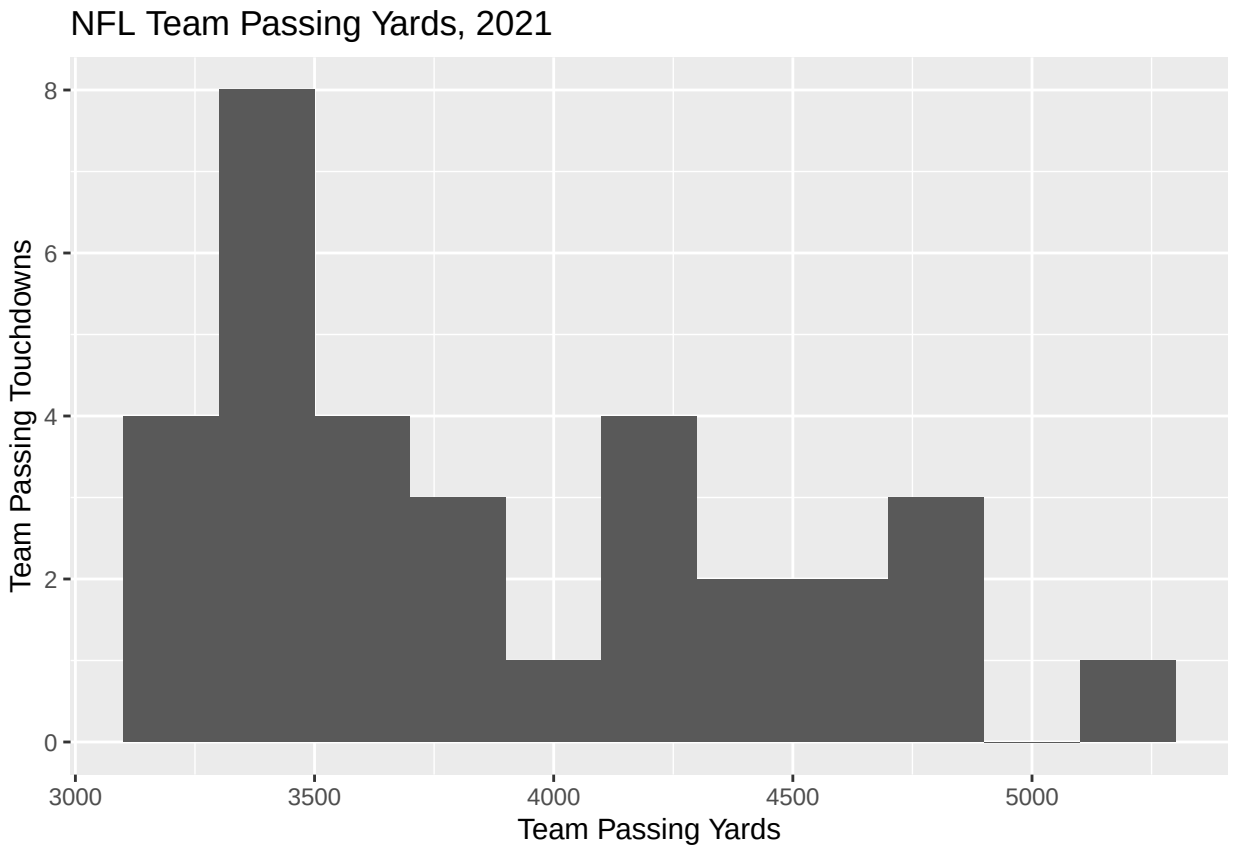


Notice how %>% is used to **pipe** the dataset into ggplot. This is using the pipe function from the **dplyr** package.

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

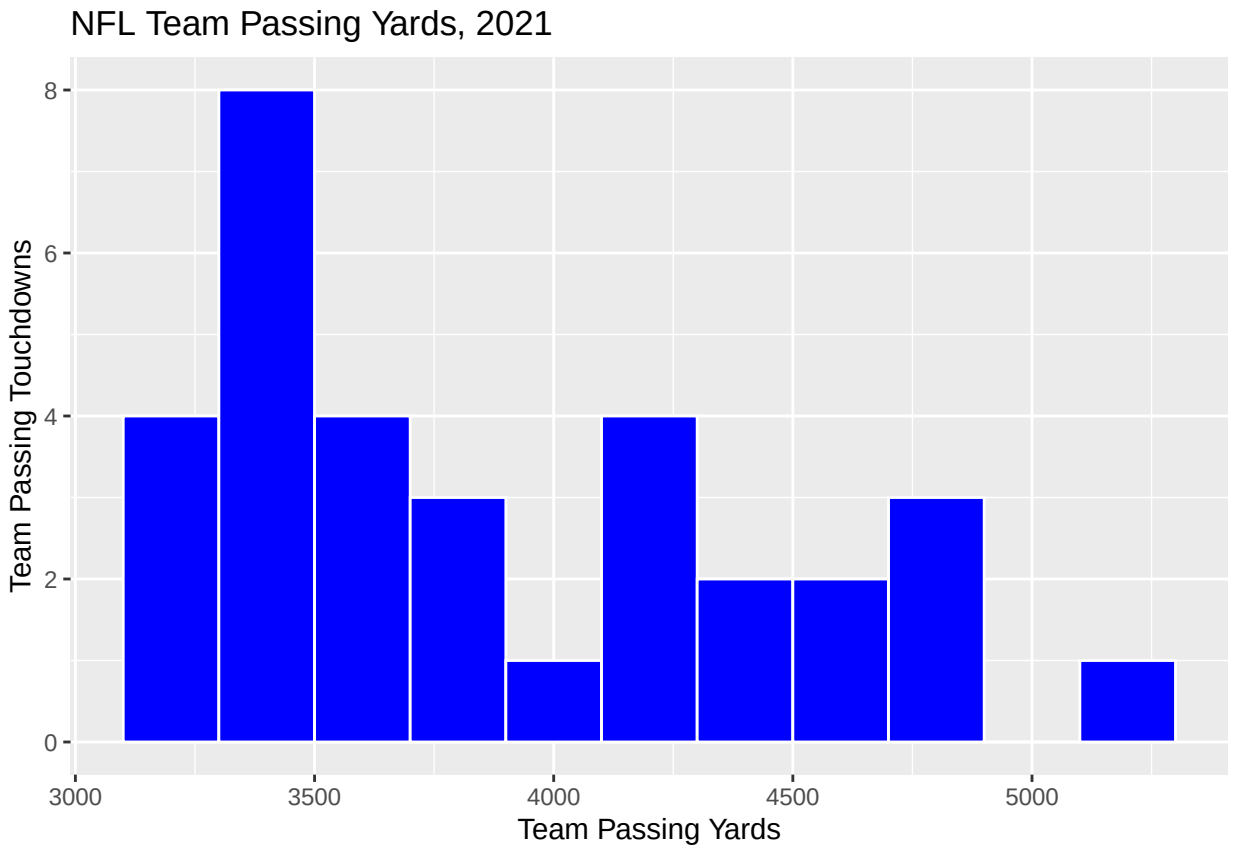
All good visualizations have good labels. Let's improve the axis labels and give the figure a title.

```
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")
```

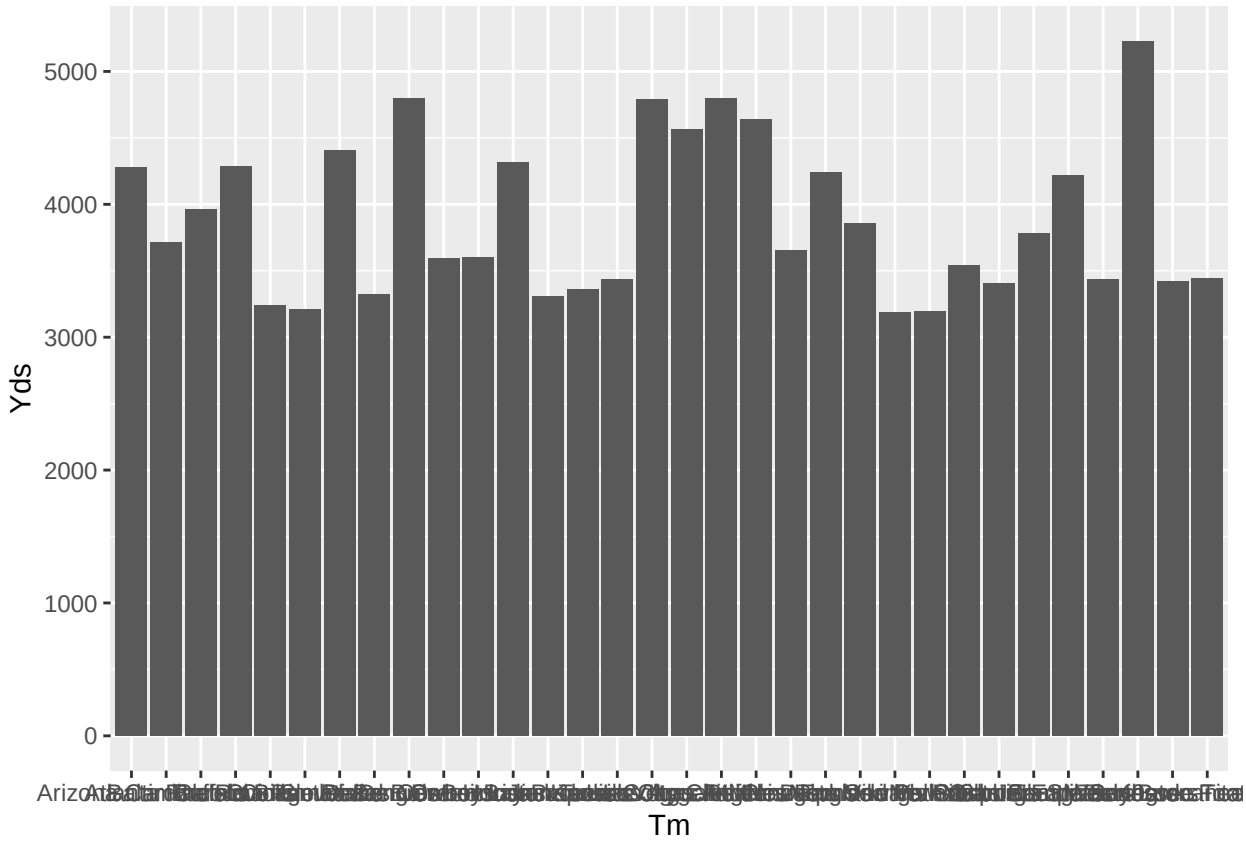


1.3.2 Bar Plots

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

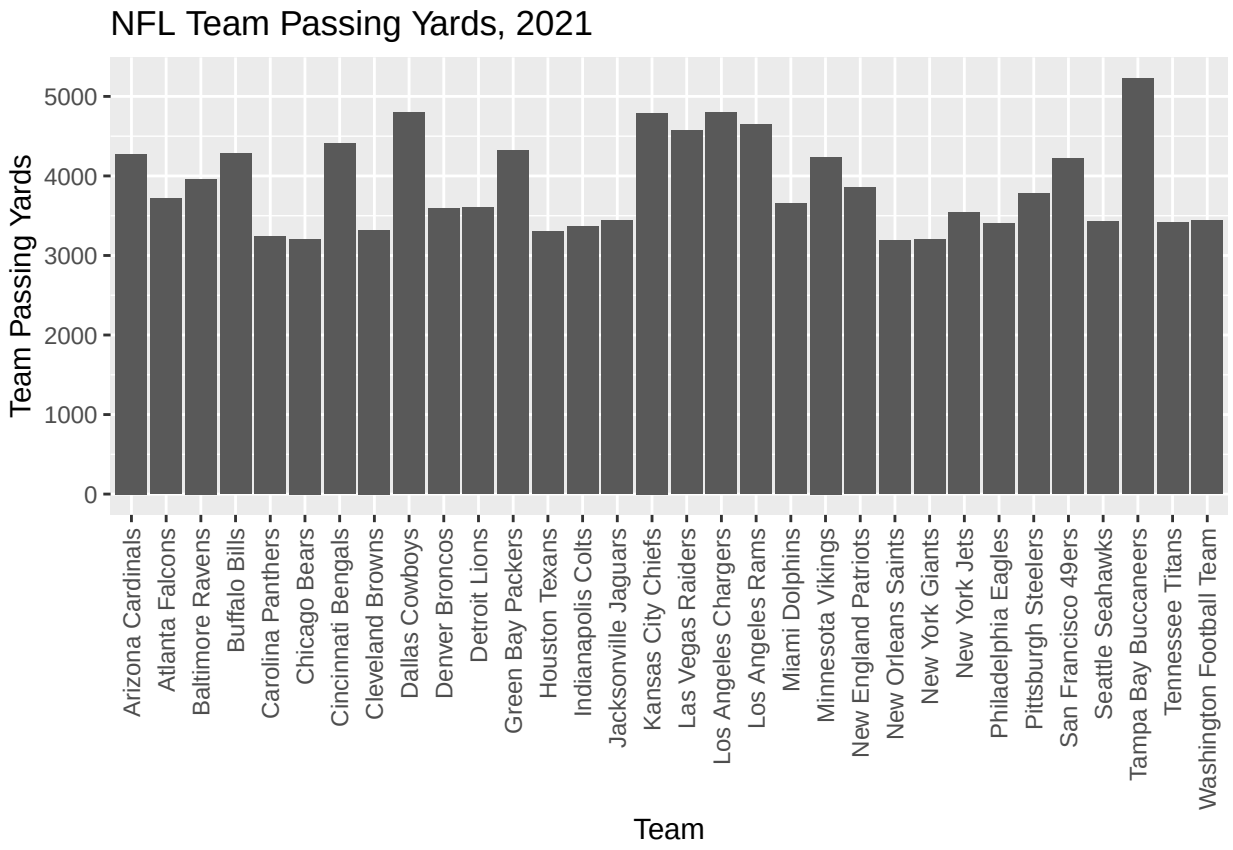
Example 1.8. Create a bar plot with teams on the horizontal axis and passing touchdowns on the vertical axis.

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



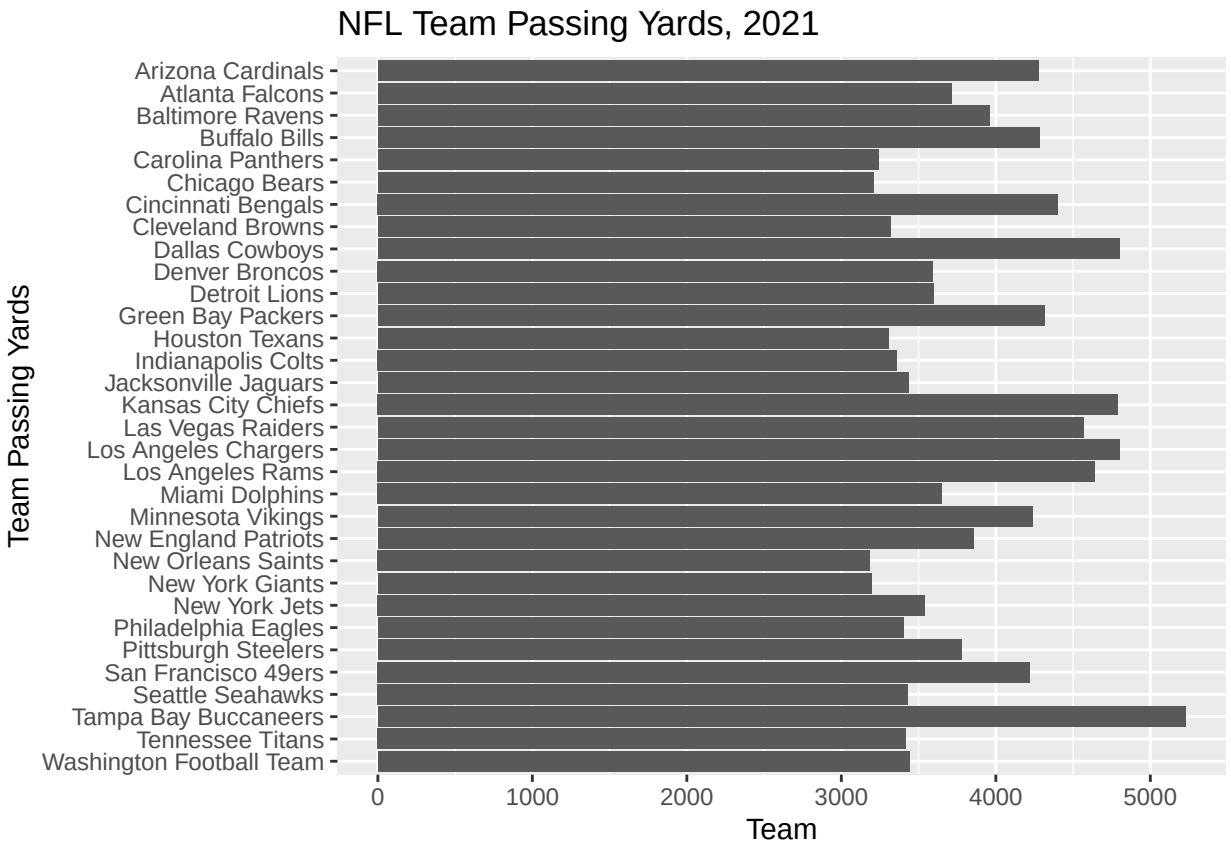
The team labels are a complete mess. Let's fix this and make some adjustments to the axis labels and figure title.

```
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 = 90,
    vjust = 0.5, hjust = 1))
```



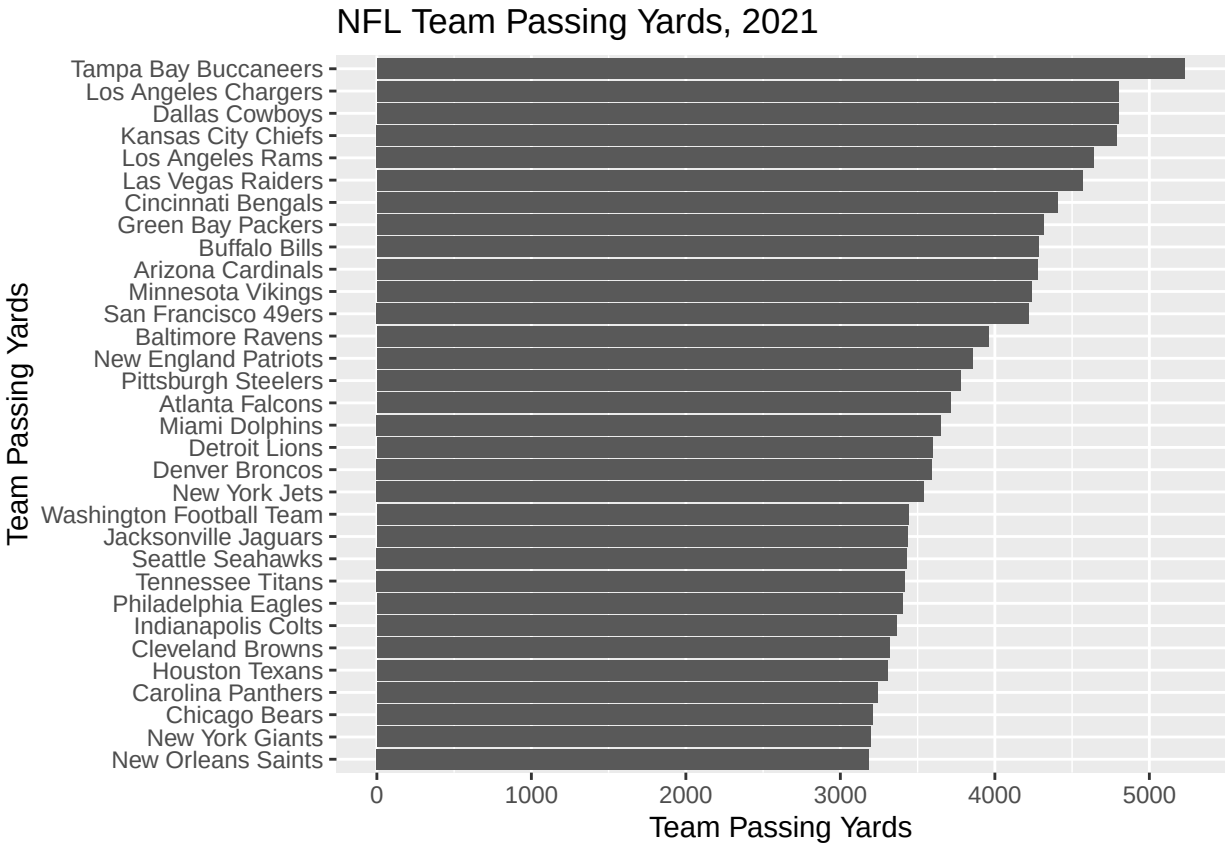
We can flip this graph if we like as well. Note that when we flip the graph, our labels get in reverse ordering, so this can be fixed using **fct_rev()** which is part of the **forcats** package.

```
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.

```
library(forcats)
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()
```

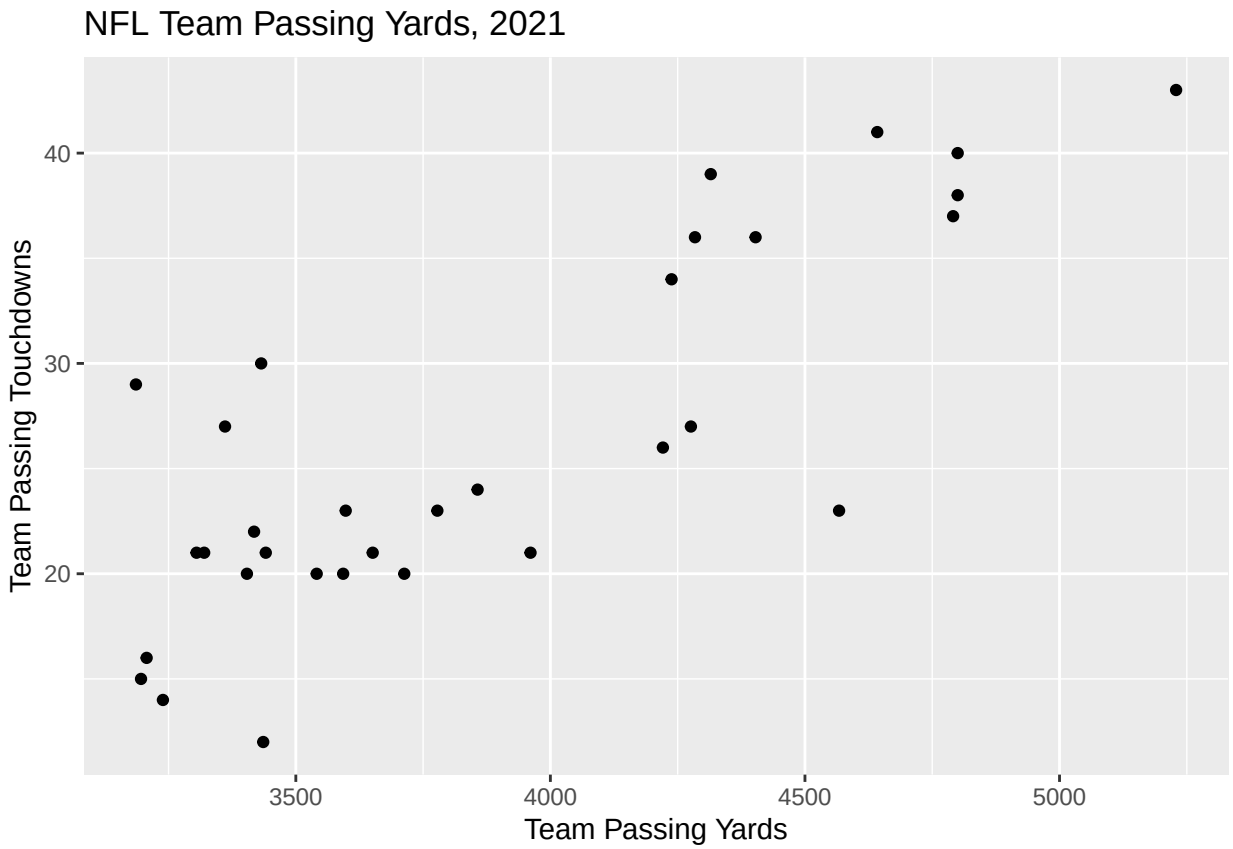


1.3.3 Scatter Plots

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

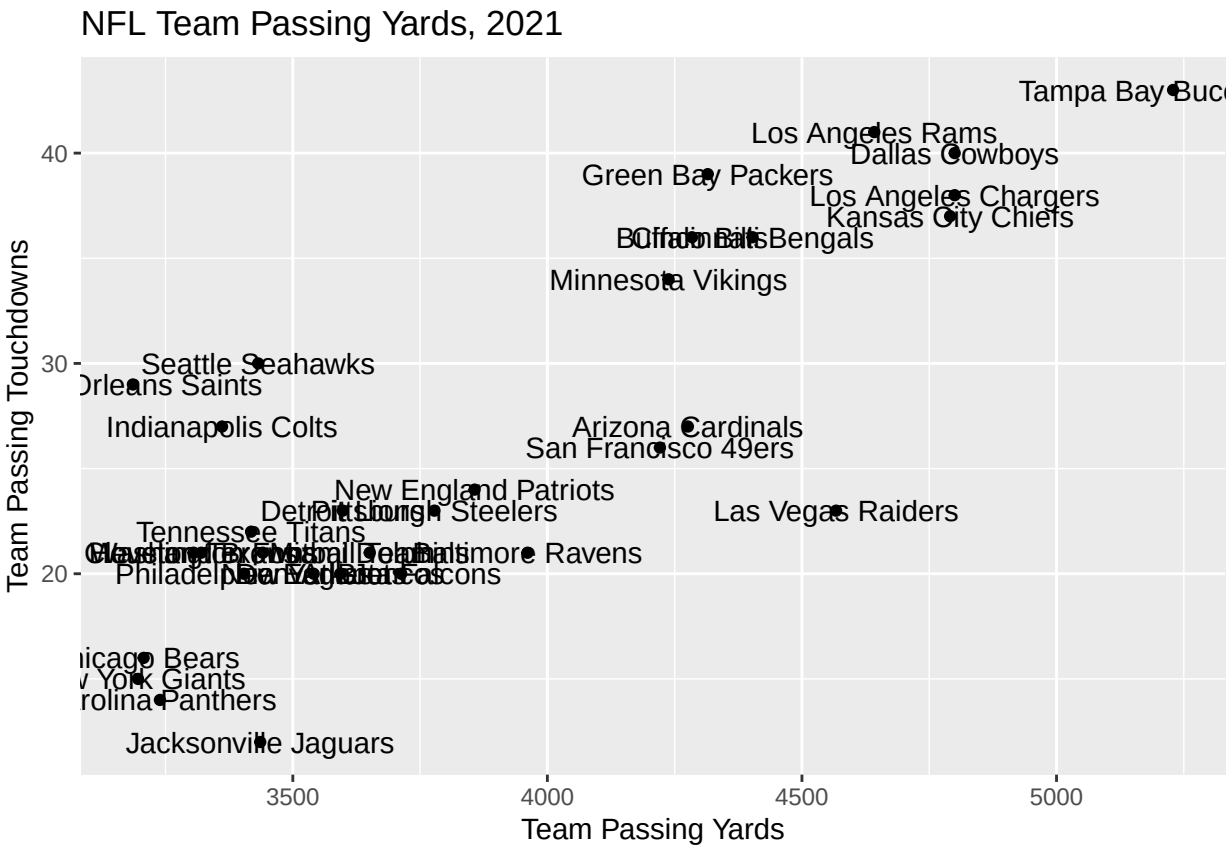
Example 1.9. Create a scatterplot of Team Passing Yards and Team Passing Touchdowns from the NFL 2021 dataset.

```
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.

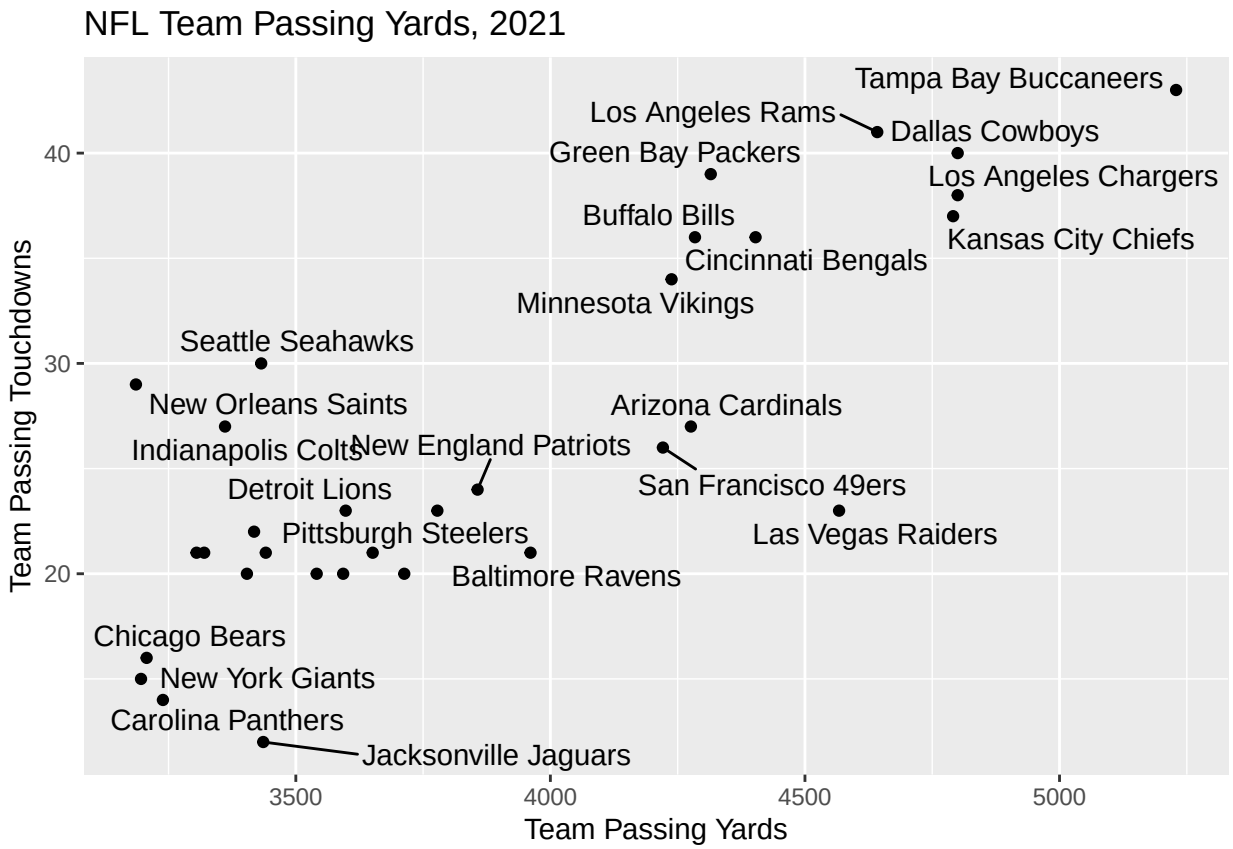
```
NFL_2021_Team_Passing %>%
    ggplot(aes(x = Yds, y = TD, label = Tm)) + geom_point() + labs(x = "Team
    Passing Yards",
    y = "Team Passing Touchdowns", title = "NFL Team Passing Yards, 2021") +
    geom_text()
```



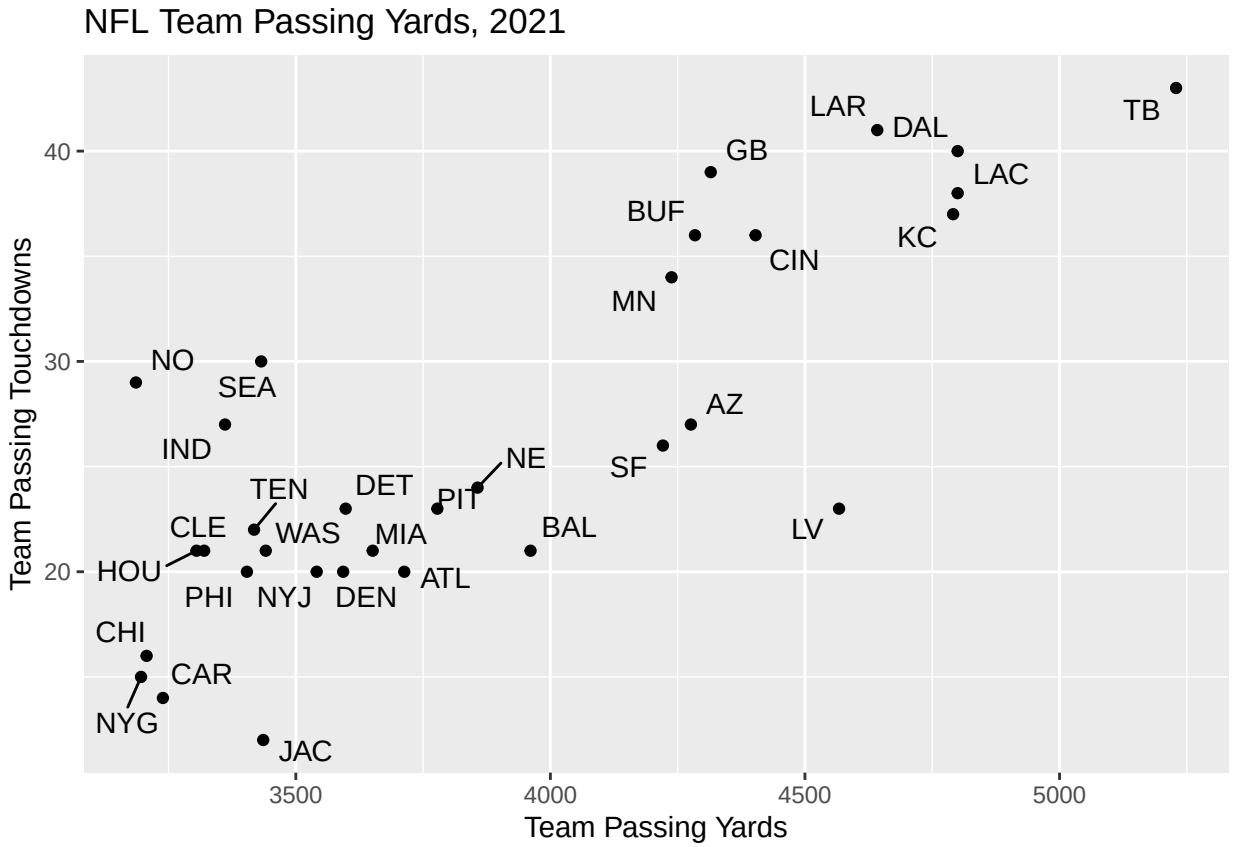
Many sports leagues have around 30 teams, so a clean scatterplot with labels can be tricky to make. Here are some options below.

```
# 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() + labs(x = "Team
    Passing Yards",
    y = "Team Passing Touchdowns", title = "NFL Team Passing Yards, 2021") +
    geom_text_repel(box.padding = 0.3)
```


NFL Team Passing Yards, 2021
Team Passing Touchdowns
Team Passing Yards
40
30
20
3500
4000
4500
5000
TB
LAR
DAL
GB
LAC
BUF
KC
CIN
MN
NO
SEA
AZ
IND
NE
SF
TEN
DET
PIT
CLE
WAS
MIA
BAL
LV
HOU
PHI
NYJ
DEN
ATL
CHI
CAR
NYG
JAC

1.4 Baseball

1.4.1 Basic Baseball 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

1.4.2 Basic Baseball Hitting 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

1.4.3 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.

1.4.4 Advanced Baseball 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

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

1.4.5 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

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

1.4.6 Calculating Advanced Hitting Statistics

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

```
rox21 <- read_csv("data/rockies_hitting2021.csv")
head(rox21)
```

```
## # A tibble: 6 x 15
##   Name      PA    AB     R     H  `2B`  `3B`    HR   RBI    BB   IBB    SO   HBP
##   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Elias~   371   338    52    83    18     1    18    44    30     1    60     2
## 2 C.J. ~   547   470    70   132    31     1    28    92    60     3   117    13
## 3 Brend~   415   387    49   110    21     3    15    51    19     0    84     7
## 4 Trevo~   595   526    88   132    34     5    24    75    53     2   139    11
## 5 Ryan ~   596   528    80   134    32     1    23    86    59     2   147     4
## 6 Raime~   533   487    69   133    26     2     6    50    40     2    70     1
## # ... with 2 more variables: SF <dbl>, oWAR <dbl>
```

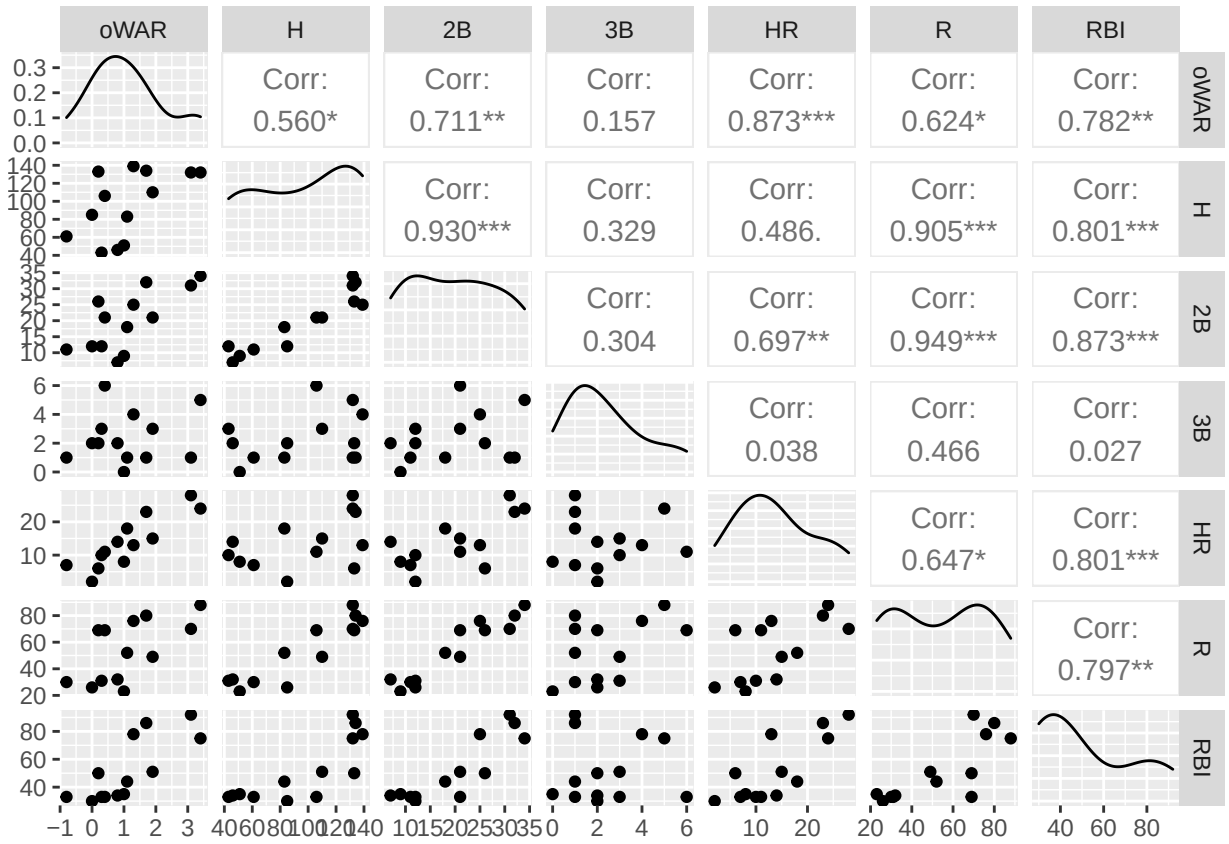
```
rox21 <- rox21 %>%
    mutate(AVG = round(H/AB, 3)) %>%
    mutate(OBA = round((H + BB + HBP)/(AB + BB + HBP + SF), 3)) %>%
    mutate(SLG = round(((H - `2B` - `3B` - HR) + 2 * `2B` + 3 * `3B` + 4 *
    HR)/AB,
        3)) %>%
    mutate(OPS = round(SLG + OBA, 3)) %>%
    mutate(wOBA = round((0.692 * (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),
        3)) %>%
    mutate(ISO = round(SLG - AVG, 3))
rox21 %>%
    select(Name, PA, AVG, OBA, SLG, OPS, wOBA, ISO)
```

```
## # A tibble: 33 x 8
##    Name                PA   AVG   OBA   SLG   OPS  wOBA   ISO
##    <chr>            <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Elias Diaz         371 0.246 0.31  0.464 0.774 0.33  0.218
```

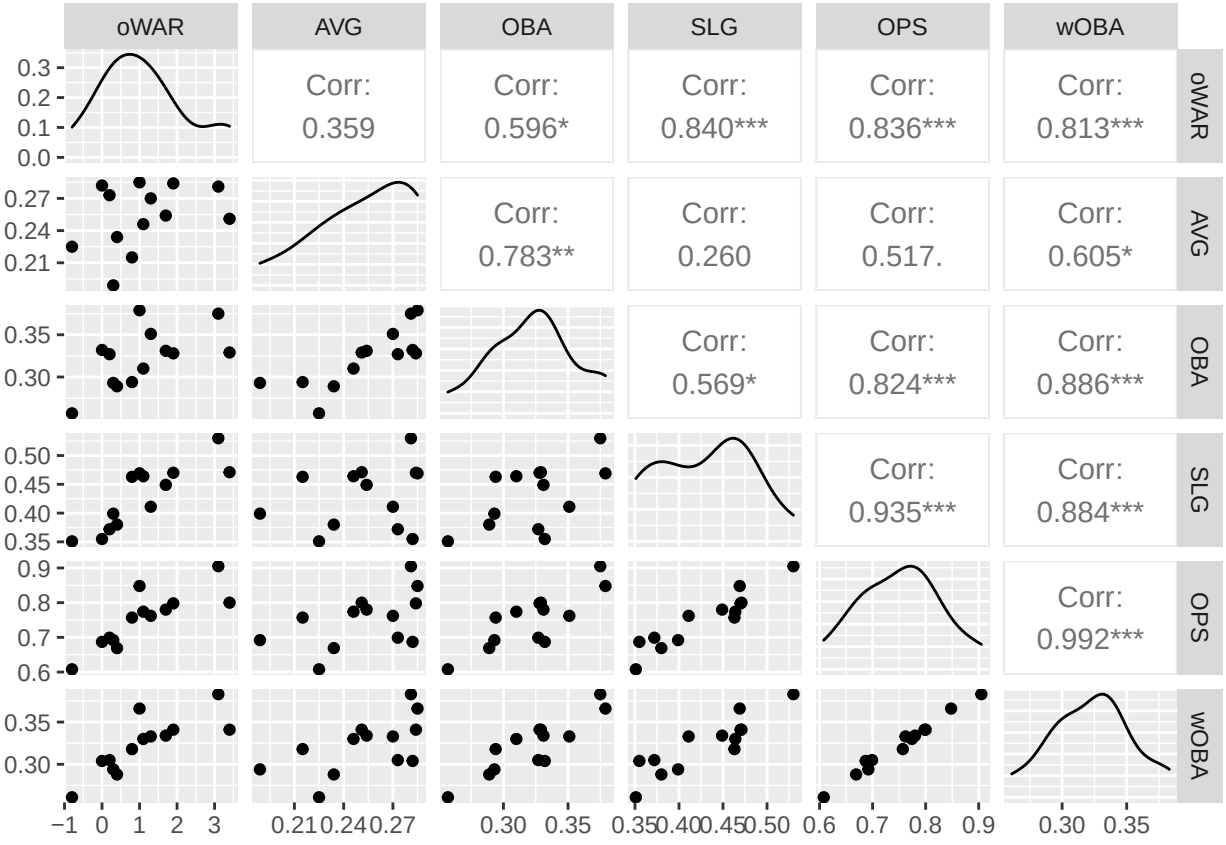
```
##  2 C.J. Cron           547 0.281 0.375 0.53  0.905 0.383 0.249
##  3 Brendan Rodgers     415 0.284 0.328 0.47  0.798 0.341 0.186
##  4 Trevor Story        595 0.251 0.329 0.471 0.8   0.341 0.22
##  5 Ryan McMahon*       596 0.254 0.331 0.449 0.78  0.334 0.195
##  6 Raimel Tapia*       533 0.273 0.327 0.372 0.699 0.305 0.099
##  7 Garrett Hampson     494 0.234 0.289 0.38  0.669 0.288 0.146
##  8 Charlie Blackmon*   582 0.27  0.351 0.411 0.762 0.333 0.141
##  9 Yonathan Daza       331 0.282 0.332 0.355 0.687 0.304 0.073
## 10 Joshua Fuentes      284 0.225 0.257 0.351 0.608 0.261 0.126
## # ... with 23 more rows
```

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()
```




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


oWAR
AVG
OBA
SLG
OPS
wOBA
Corr: 0.359
Corr: 0.596*
Corr: 0.840***
Corr: 0.836***
Corr: 0.813***
Corr: 0.783**
Corr: 0.260
Corr: 0.517.
Corr: 0.605*
Corr: 0.569*
Corr: 0.824***
Corr: 0.886***
Corr: 0.935***
Corr: 0.884***
Corr: 0.992***

1.5 Football

1.6 Basketball

Link to YouTube video describing basketball rules

1.6.1 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.

1.6.2 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

1.6.3 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
```

```
## # A tibble: 2 x 9
##   team     FG   FGA THREEP    FT   FTA   ORB   DRB   TOV
##   <chr> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 CSU      14    48      5    10    14     2    31     5
## 2 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.

The first is **Effective Field Goal Percentage**, commonly abbreviated eFG%. The formula is as follows:

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

Secondly, **Turnover Percentage** (TOV%) is calculated as:

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

Next, **Rebounding Percentage** (ORB%) is computed as:

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

Finally, the **Free Throw Factor** is found using:

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

Note: You do not have to know these formulas for the test. They are just used for this example.

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.

```
attach(basketball_data)
eFG <- round((FG + 0.5 * THREEP)/FGA, 3) * 100
TOVPCT <- round(TOV/(FGA + 0.44 * FTA + TOV), 3) * 100
ORBPCT <- round(c(ORB[1]/(ORB[1] + DRB[2]), ORB[2]/(ORB[2] + DRB[1])), 3) * 100
FTFACTOR <- round(FT/FGA, 3) * 100
```

```
basketball_data %>%
    add_column(eFG, TOVPCT, ORBPCT, FTFACTOR)
```

```
## # A tibble: 2 x 13
##   team     FG   FGA THREEP    FT   FTA   ORB   DRB   TOV   eFG TOVPCT ORBPCT
##   <chr> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1 CSU      14    48      5    10    14     2    31     5  34.4    8.5    6.2
## 2 WYO      15    60      4     4     4    14    30    12  28.3   16.3   31.1
## # ... with 1 more variable: FTFACTOR <dbl>
```

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.

1.6.4 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.

```
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, ])
print(four_factors_teams)
```

```
##                    Team POSS.Off POSS.Def PACE.Off PACE.Def   ORtg   DRtg
## 1   Cleveland Cavaliers  8221.96  8229.72 2.083619 2.085585 110.57 109.53
## 2        Denver Nuggets  8232.20  8179.24 2.070473 2.057153 109.57 108.80
## 3 Golden State Warriors  8287.92  8457.08 2.100335 2.143203 112.26 104.22
##   F1.Off F2.Off F3.Off F4.Off F1.Def F2.Def F3.Def F4.Def
## 1  54.70  13.70  20.06  21.41  53.98  13.43  77.27  16.58
## 2  53.62  14.90  25.66  19.77  53.88  13.83  77.45  17.35
## 3  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.

1.6.5 BasketballAnalyzeR Shot Charts

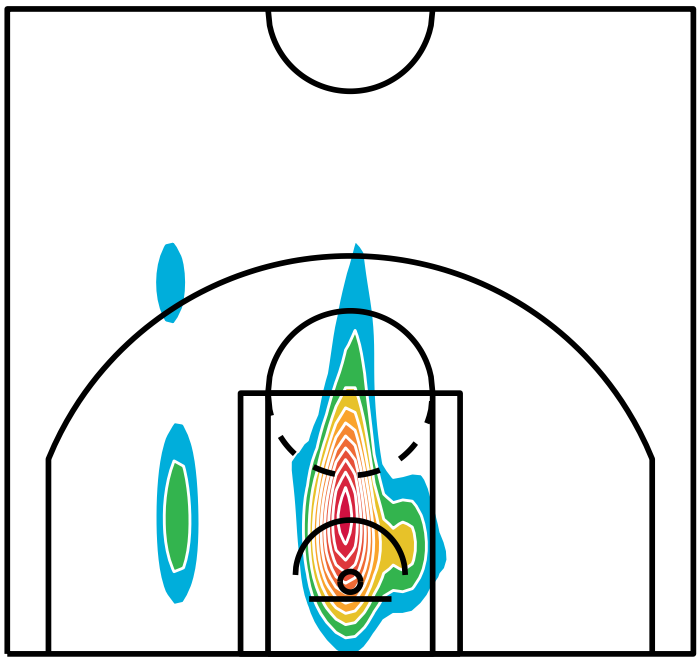
The `BasketballAnalyzeR` package includes shot location data for all players for the 2017-18 NBA season and has a function called `shotchart` that allows for the plotting of shot data.

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.

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

jokic_data <- subset(PbP, player == "Nikola Jokic")
jokic_data$xx <- jokic_data$original_x/10  #transformation
jokic_data$yy <- jokic_data$original_y/10 - 41.75  #transformation
```

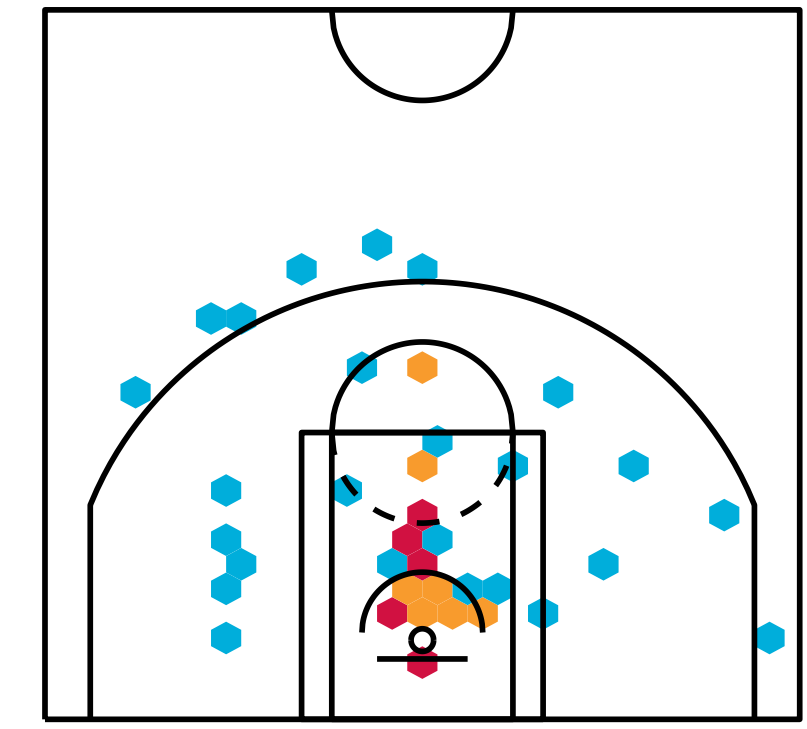
`BasketballAnalyzeR` supports three types of density visualizations within `shotchart`, one being `density-polygons`. Since `shotchart` is a `ggplot` object, a chart title can be added using `ggtitle`.

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



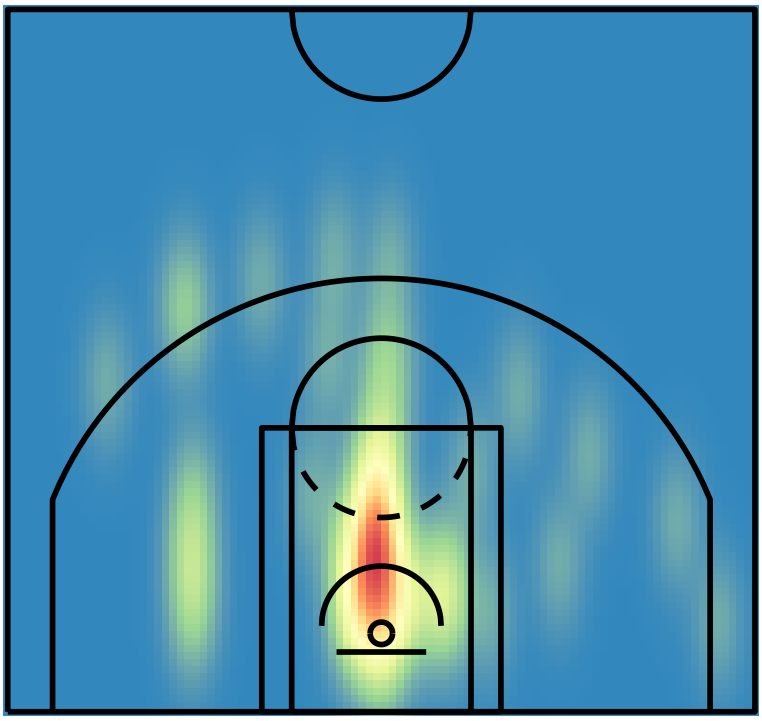
It seems most shots attempts from Jokic were in the paint; this is hardly surprising, since he plays the center position. Here's the same chart with `density-hexbin`:

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



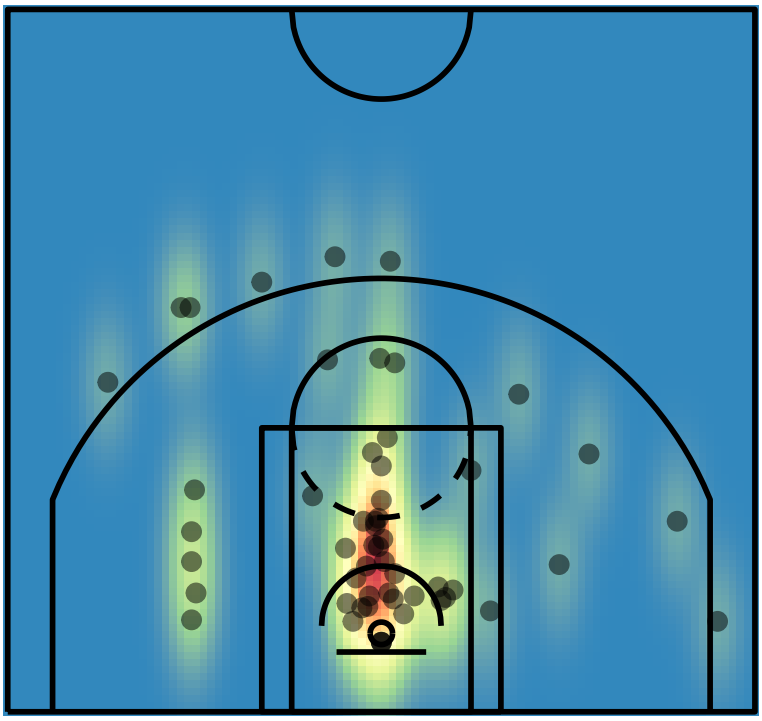
The same chart with `density-raster`:

```
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.

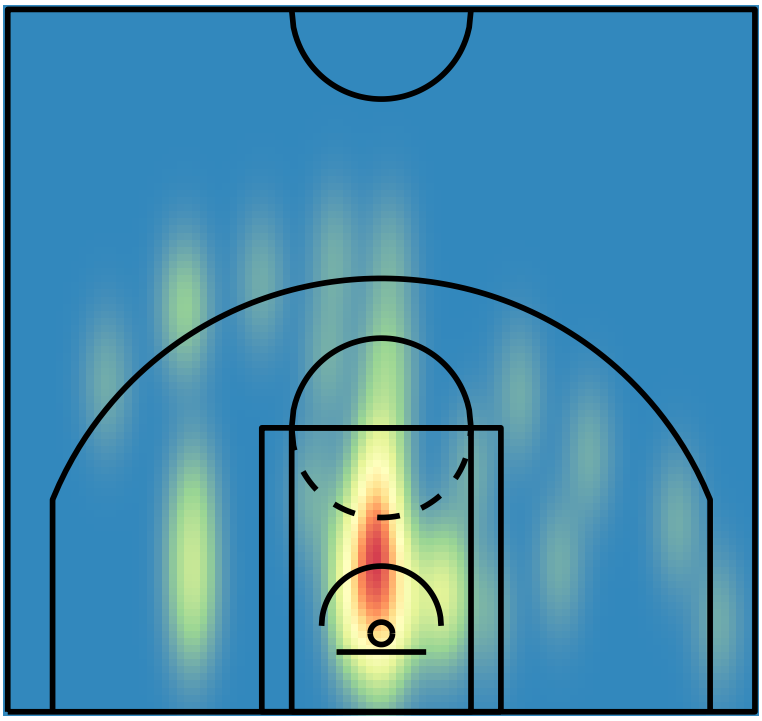
```
shotchart(data = jokic_data, x = "xx", y = "yy", type = "density-raster", scatter
= TRUE) +
    ggtitle("Nikola Jokic Shot Data, 2017-18")
```



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.

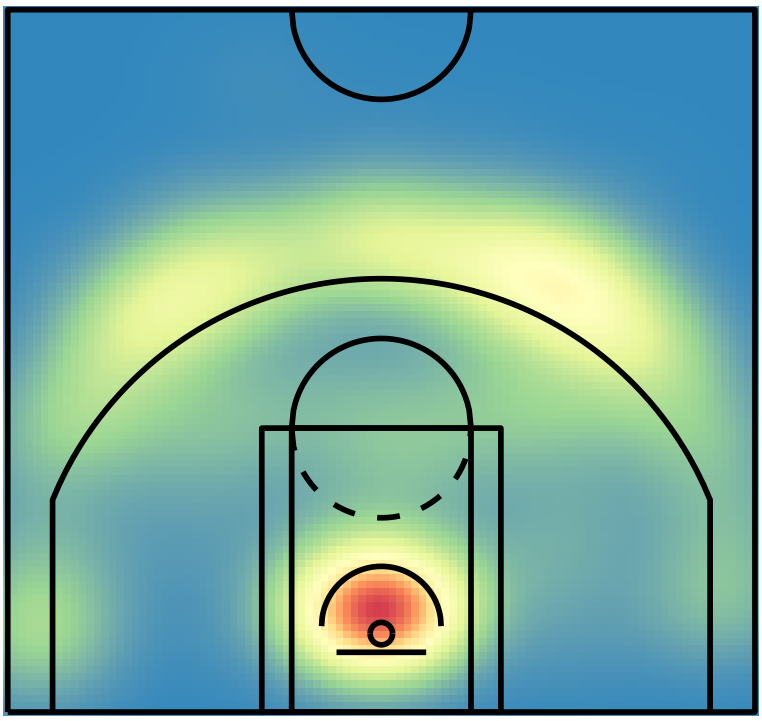
```
curry_data <- subset(PbP, player == "Stephen Curry")
curry_data$xx <- curry_data$original_x/10  #transformation
curry_data$yy <- curry_data$original_y/10 - 41.75  #transformation

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

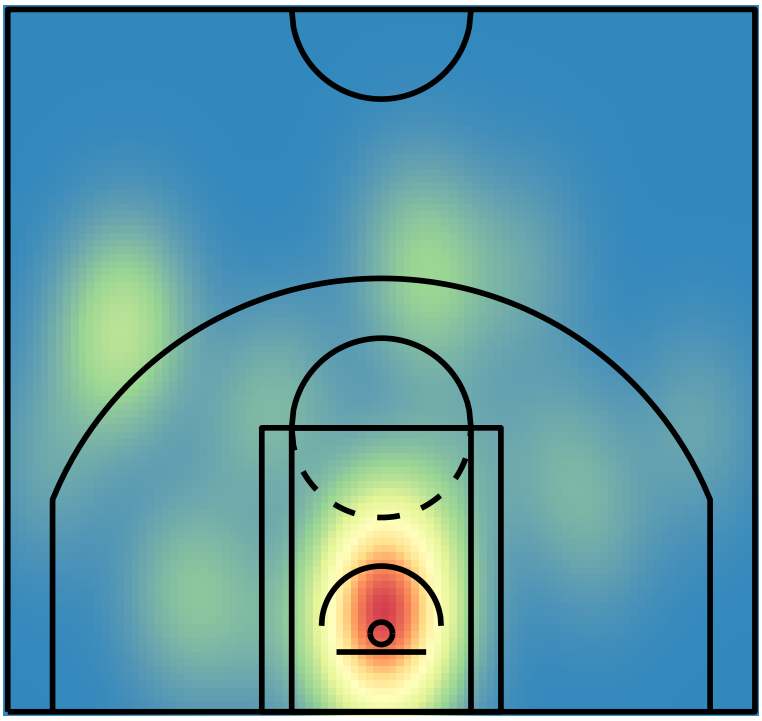


```
shotchart(data = curry_data, x = "xx", y = "yy", type = "density-raster") +
ggtitle("Steph Curry Shot Data, 2017-18")
```

Steph Curry Shot Data, 2017−18



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

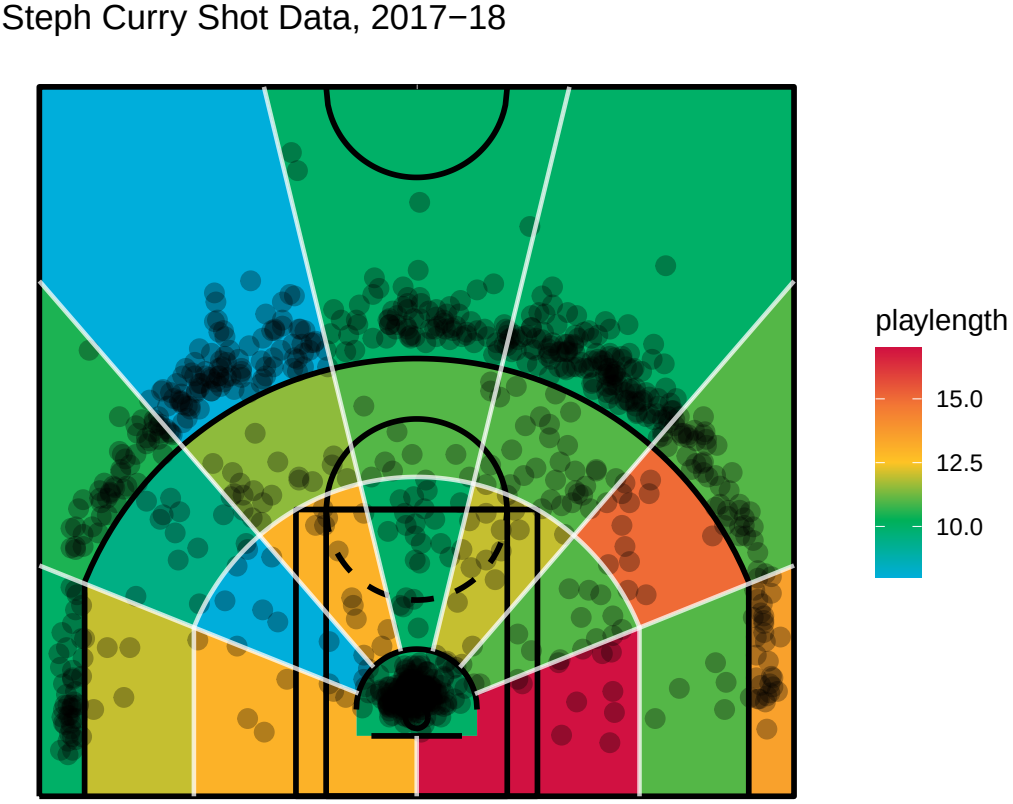


Q: Of the three players (Jokic, Curry, James), which took the highest percentage of their three-point shots from the right side of the court (when facing the basket)?

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 = 0.3) + 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)

1.7 Hockey

Link to YouTube video describing hockey rules

1.7.1 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

1.7.2 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/

1.7.3 Actual vs. Expected Goals

Example 1.12. 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.

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

head(nhl_2022_data)
```

```
## # A tibble: 6 x 107
##   team...1 season name  team...4 position   situation games_played
##   <chr>     <dbl> <chr> <chr>    <chr>      <chr>            <dbl>
## 1 WPG        2021 WPG   WPG      Team Level other               82
## 2 WPG        2021 WPG   WPG      Team Level all                 82
## 3 WPG        2021 WPG   WPG      Team Level 5on5                82
## 4 WPG        2021 WPG   WPG      Team Level 4on5                82
## 5 WPG        2021 WPG   WPG      Team Level 5on4                82
## 6 CBJ        2021 CBJ   CBJ      Team Level other               82
## # ... with 100 more variables: xGoalsPercentage <dbl>, corsiPercentage <dbl>,
## #   fenwickPercentage <dbl>, iceTime <dbl>, xOnGoalFor <dbl>, xGoalsFor <dbl>,
## #   xReboundsFor <dbl>, xFreezeFor <dbl>, xPlayStoppedFor <dbl>,
## #   xPlayContinuedInZoneFor <dbl>, xPlayContinuedOutsideZoneFor <dbl>,
## #   flurryAdjustedxGoalsFor <dbl>, scoreVenueAdjustedxGoalsFor <dbl>,
## #   flurryScoreVenueAdjustedxGoalsFor <dbl>, shotsOnGoalFor <dbl>,
## #   missedShotsFor <dbl>, blockedShotAttemptsFor <dbl>, ...
```

We can create nice looking tables using the "kableExtra'' package. Let's look at the first eight rows and a samll 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)] %>%
    kbl() %>%
    kable_styling()
```

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?

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

nrow(nhl_data_all)
```

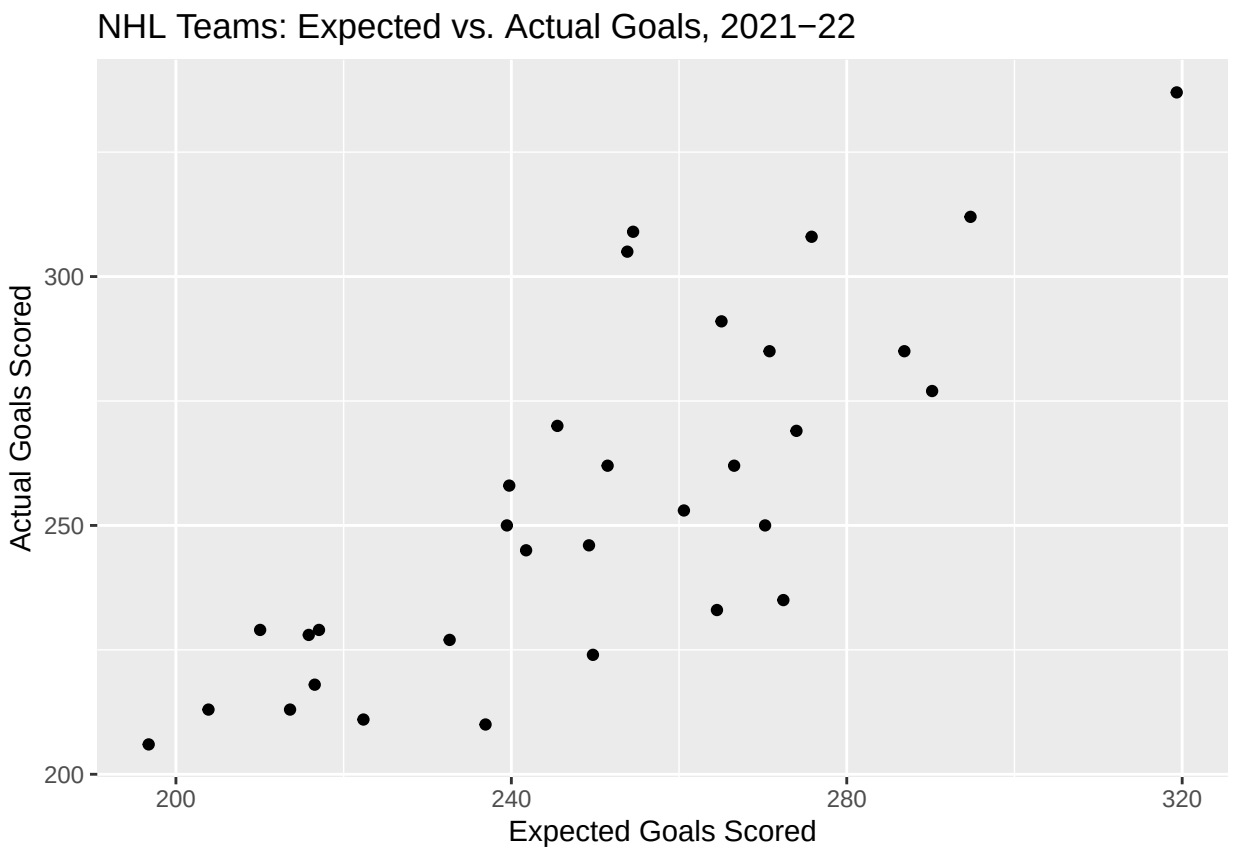
name	situation	games_played	xGoalsPercentage	corsiPercentage
WPG	other	82	0.49	0.50
WPG	all	82	0.49	0.50
WPG	5on5	82	0.49	0.49
WPG	4on5	82	0.16	0.14
WPG	5on4	82	0.86	0.86
CBJ	other	82	0.52	0.49
CBJ	all	82	0.45	0.48
CBJ	5on5	82	0.45	0.48

```
## [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!)

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

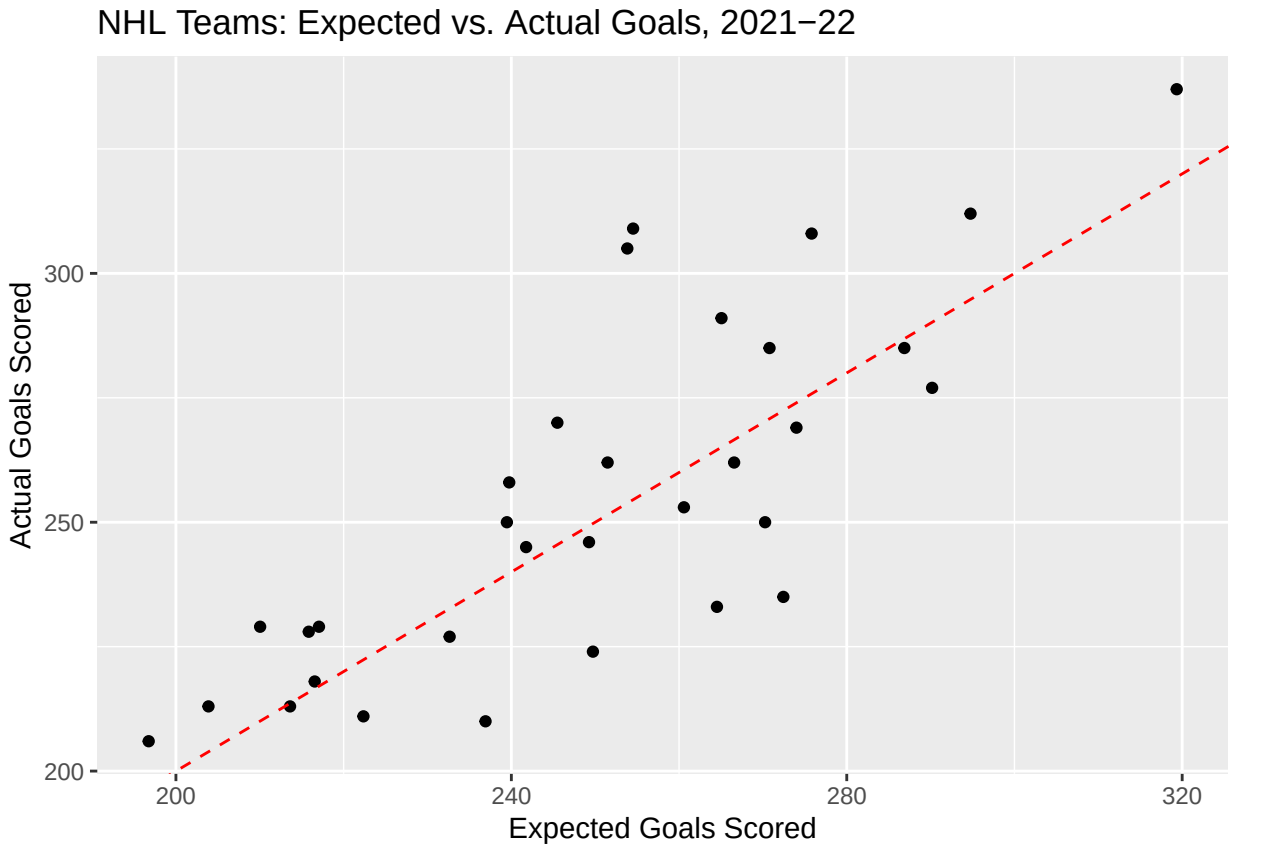


As expected, there is a general positive correlation between expected and actual goals ($r \approx 0.8$).

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)) + labs(x =
"Expected Goals Scored",
    y = "Actual Goals Scored", title = "NHL Teams: Expected vs. Actual Goals,
    2021-22") +
    geom_point() + geom_abline(intercept = 0, slope = 1, color = "red", linetype
    = "dashed")
```



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

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

A: If the data point is below the line, it means the expected goals were greater than the actual goals; if the data point is above the line, it means the actual goals were greater than the expected goals.

Q: 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?

A: We would expect that as sample size increases, the result would become closer to expectation. So, actual goals would be most likely closer to expected goals over a span of five seasons.

1.7.4 Goalie Statistics

Example 1.13. 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")
head(goalie_data)
```

```
##     X           Player Team GP       TOI Shots.Against Saves Goals.Against
## 1  87 Charlie Lindgren  STL  5 246.41667           118   113             5
## 2  34   Louis Domingue  PIT  2 118.75000            83    79             4
## 3  63   Spencer Martin  VAN  6 378.35000           218   207            11
## 4  37   Michael Houser  BUF  2 120.00000            75    71             4
## 5 100   Daniil Tarasov  CBJ  4 174.85000           111   104             7
## 6  41    Garret Sparks  L.A  2  97.21667            47    44             3
##     SV.  GAA GSAA xG.Against HD.Shots.Against HD.Saves HD.Goals.Against HDSV.
## 1 0.958 1.22 5.97      10.21               29       27                2 0.931
## 2 0.952 2.02 3.72       7.16               25       21                4 0.840
## 3 0.950 1.74 9.27      21.11               62       57                5 0.919
## 4 0.947 2.00 2.97       5.52               14       12                2 0.857
## 5 0.937 2.40 3.32       9.07               31       25                6 0.806
## 6 0.936 1.85 1.37       3.42               12        9                3 0.750
##   HDGAA HDGSAA Rush.Attempts.Against Rebound.Attempts.Against
## 1  0.49   3.32                     2                       13
## 2  2.02   0.58                     7                       13
## 3  0.79   6.37                     8                       27
## 4  1.00   0.57                     2                       10
## 5  2.06  -0.32                     3                       18
## 6  1.85  -0.80                     0                        5
##   Avg..Shot.Distance Avg..Goal.Distance
## 1              36.58              16.40
## 2              36.06               9.00
## 3              33.36              25.09
## 4              37.05              19.50
## 5              37.50              19.71
## 6              32.32              17.33
```

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)) %>%
```

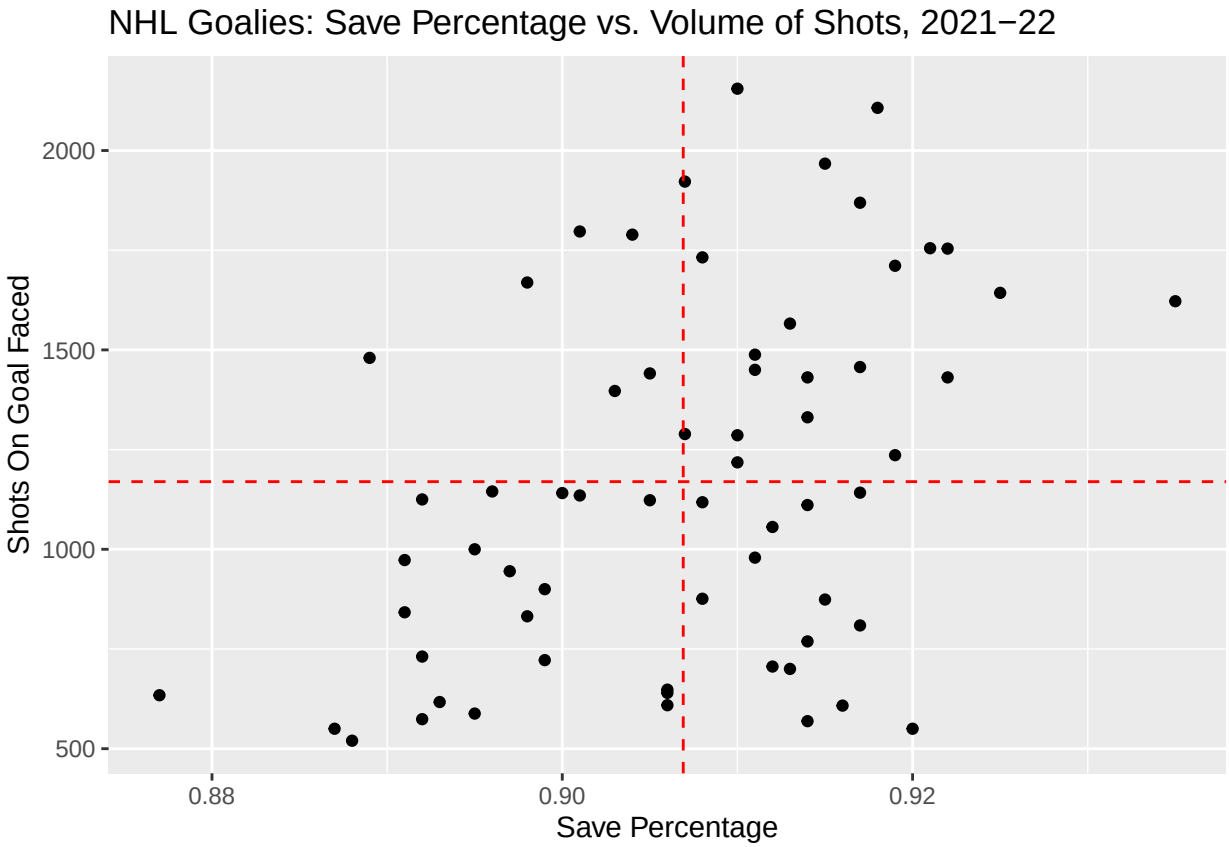
```
    select(Player, Team, GAA)
```

```
##            Player Team  GAA
## 1 Igor Shesterkin  NYR 2.07
```

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

```
##        Player Team      TOI
## 1 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.



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

A: The second (upper left) quadrant.

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

A: The first (upper right) quadrant.

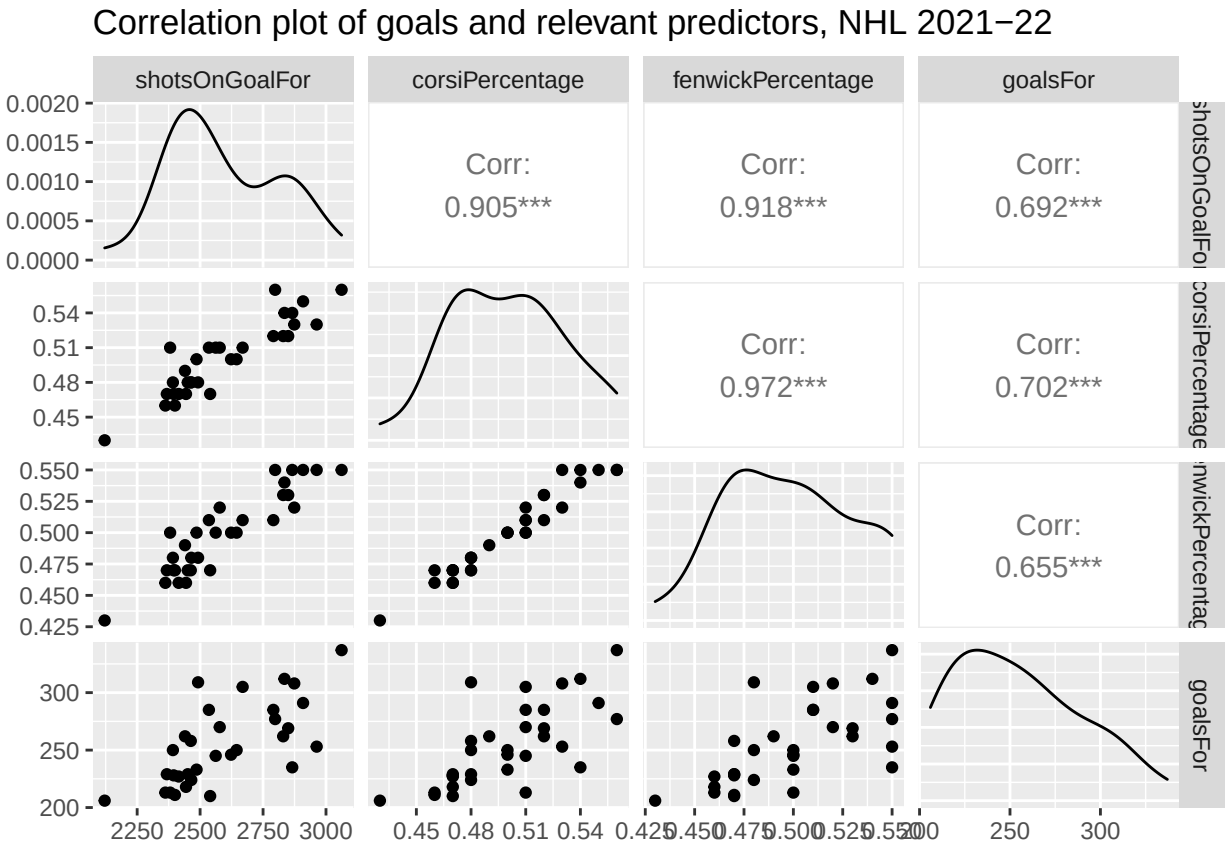
1.7.5 Correlation Plots

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 1.14. 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.

```
library("GGally")

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



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

A: CORSI percentage, r = .702.

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?

1.8 Volleyball

1.8.1 Basic Volleyball Statistics

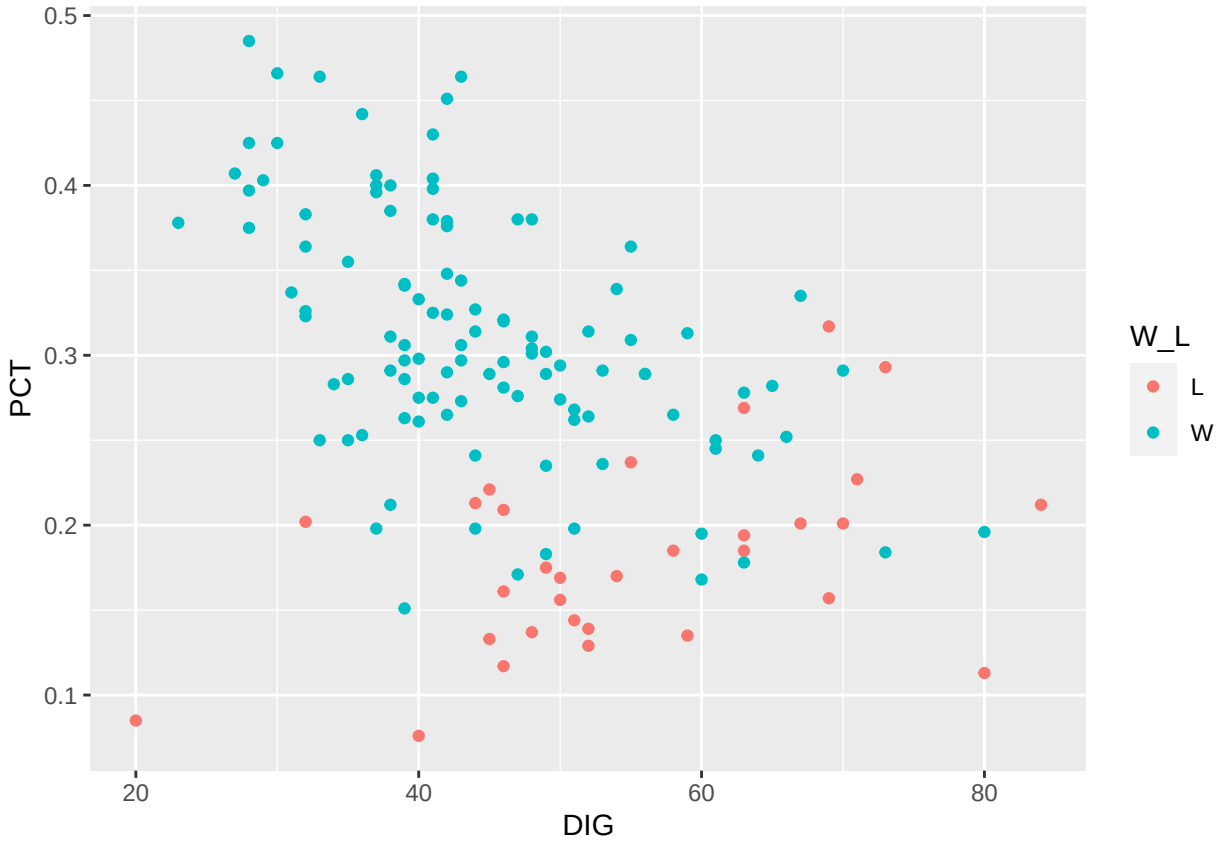
To begin, let's go over some basic volleyball statistics. The following definitions come from www.rookieroad.com

- 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.

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

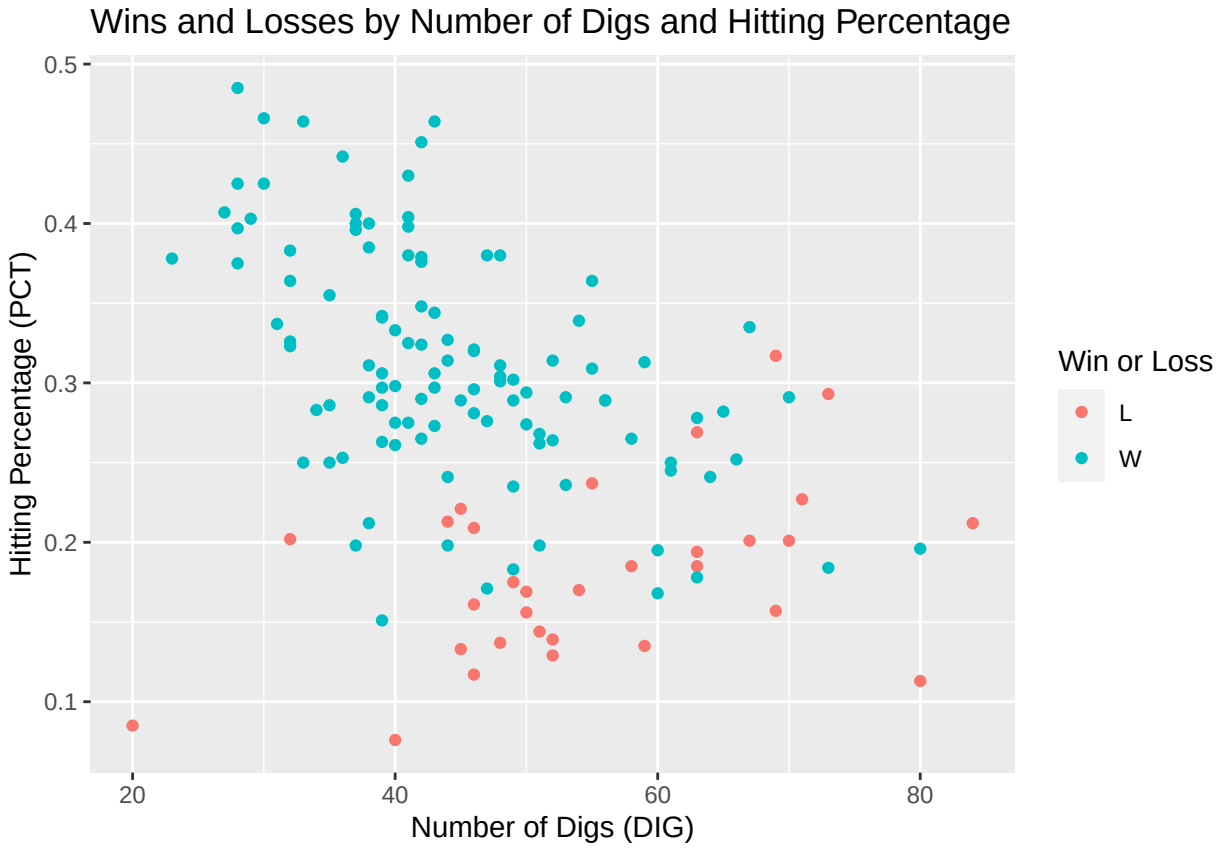
Let's look at a scatter plot of hitting percentage and the number of digs. While no conclusions can be drawn from such a plot, it can give us some insight into relationships worthy of further analysis. Before creating the plot using the code below, think about what you might expect the outcome to be.

```
# 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")
```

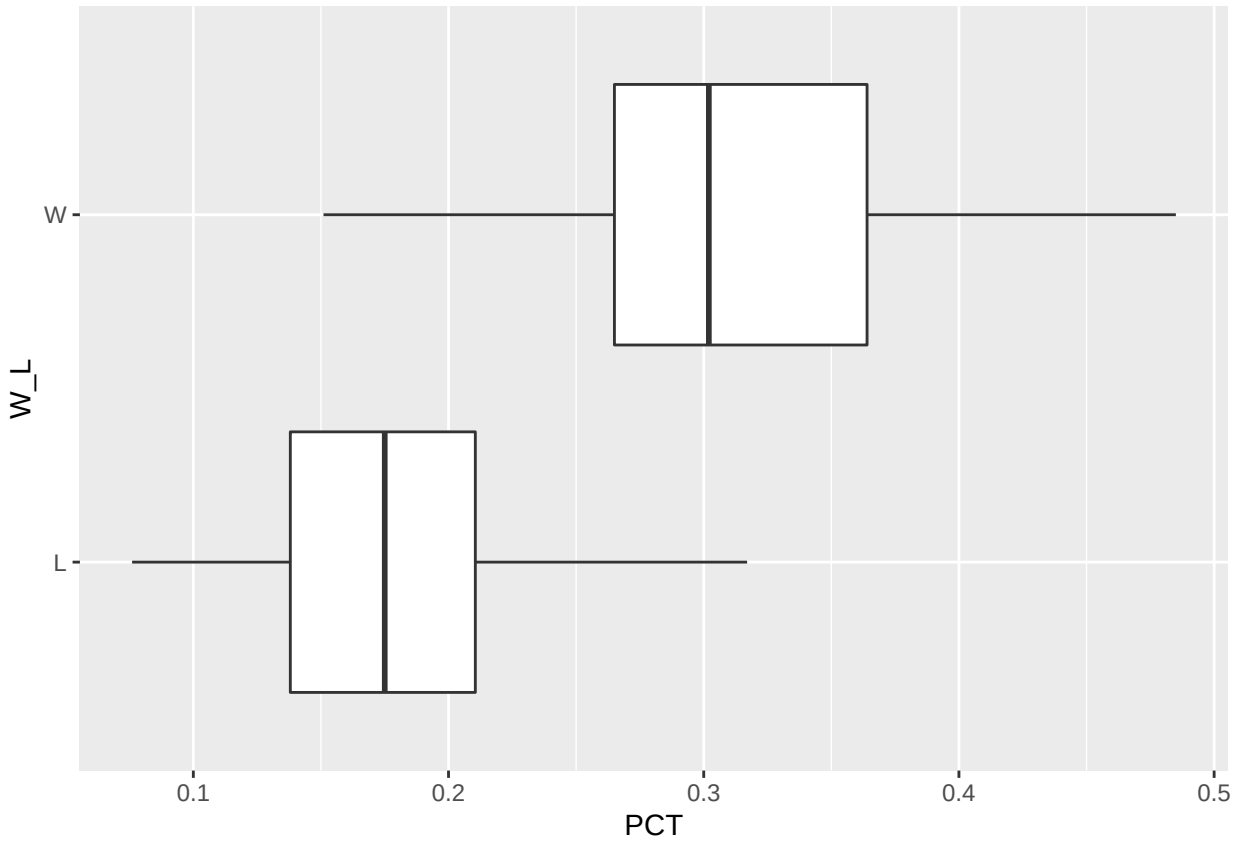


What can we learn from this visual? Well, we can see that there is a weak linear relationship between the number of digs and hitting percentage. To an extent, hitting percentage decreases as the number of digs increases. Why is this the case? Maybe if a team has a really high hitting percentage, this 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.

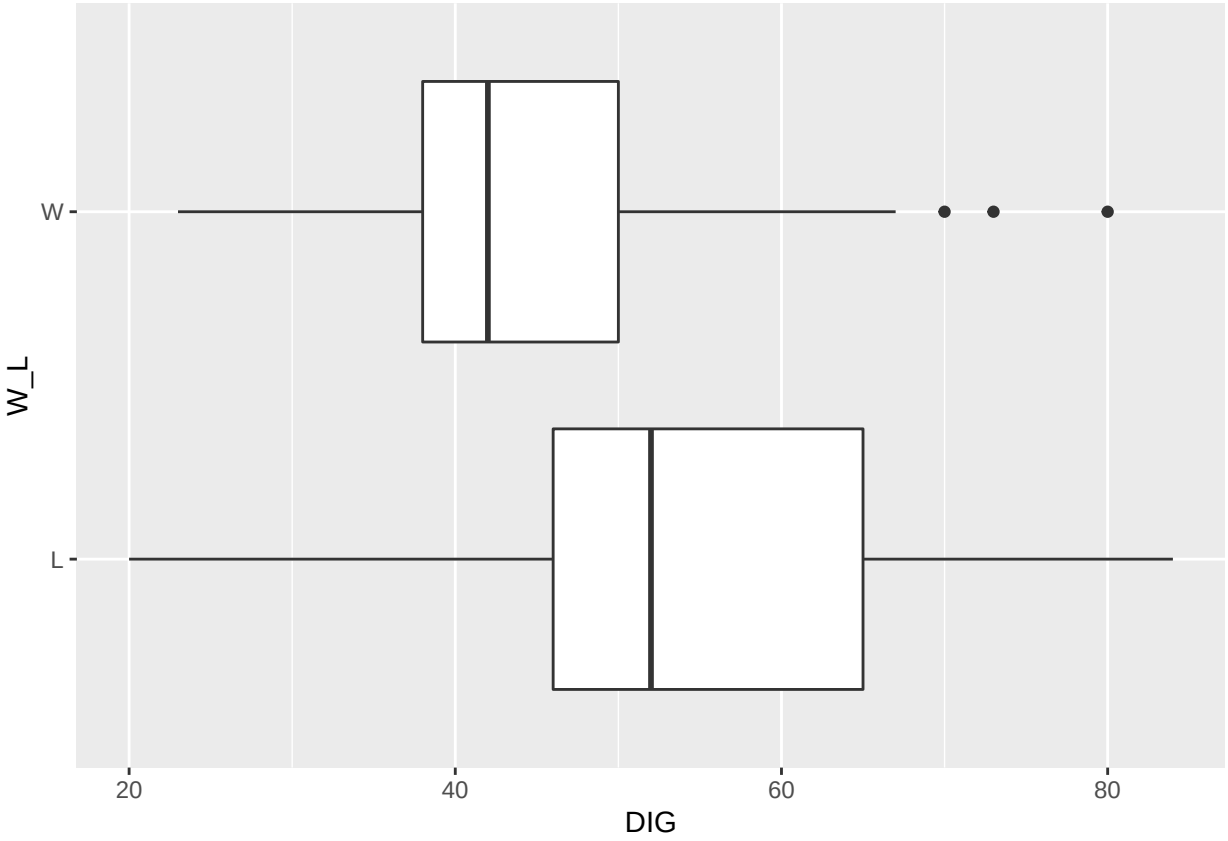
1.8.3 Box Plot

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
```

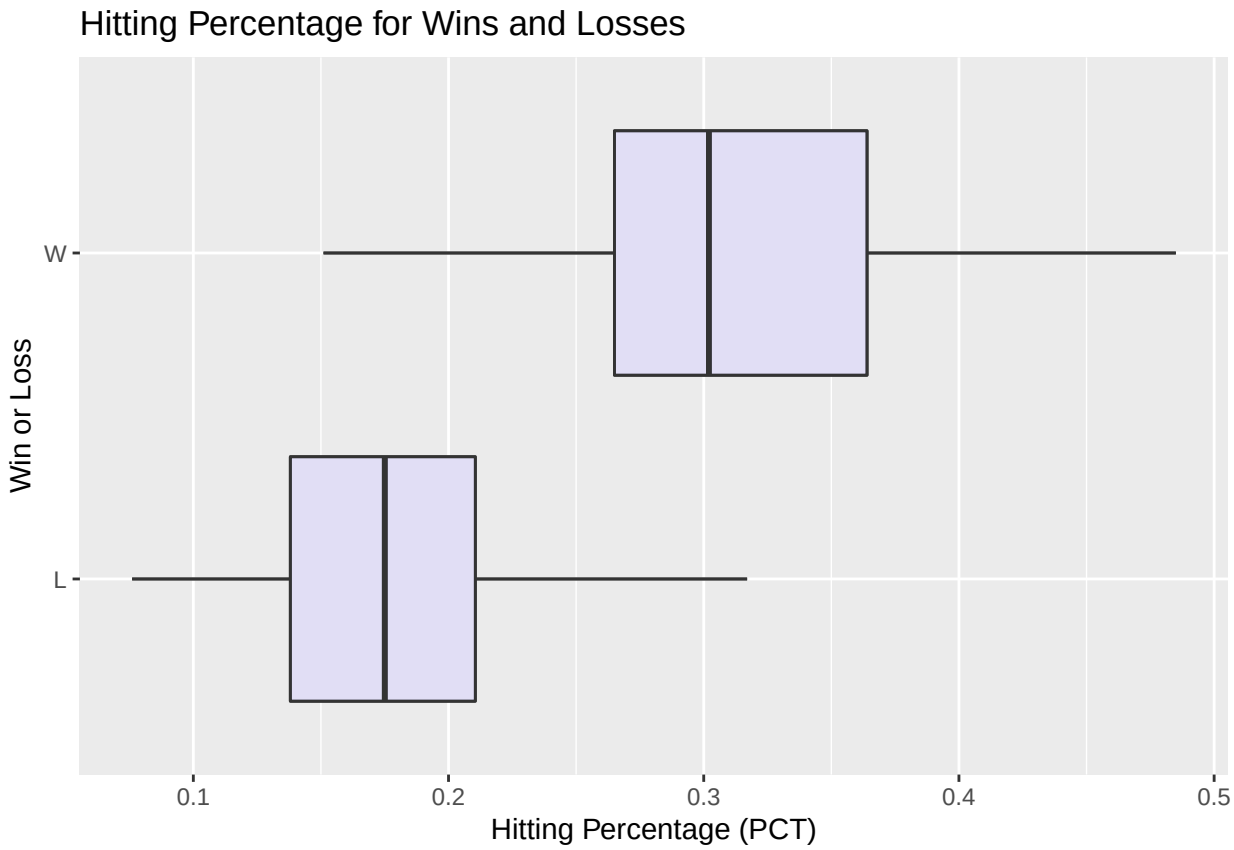



```
dig_viz <- ggplot(data = csu_vb, aes(x = DIG, y = W_L)) + geom_boxplot()
dig_viz
```

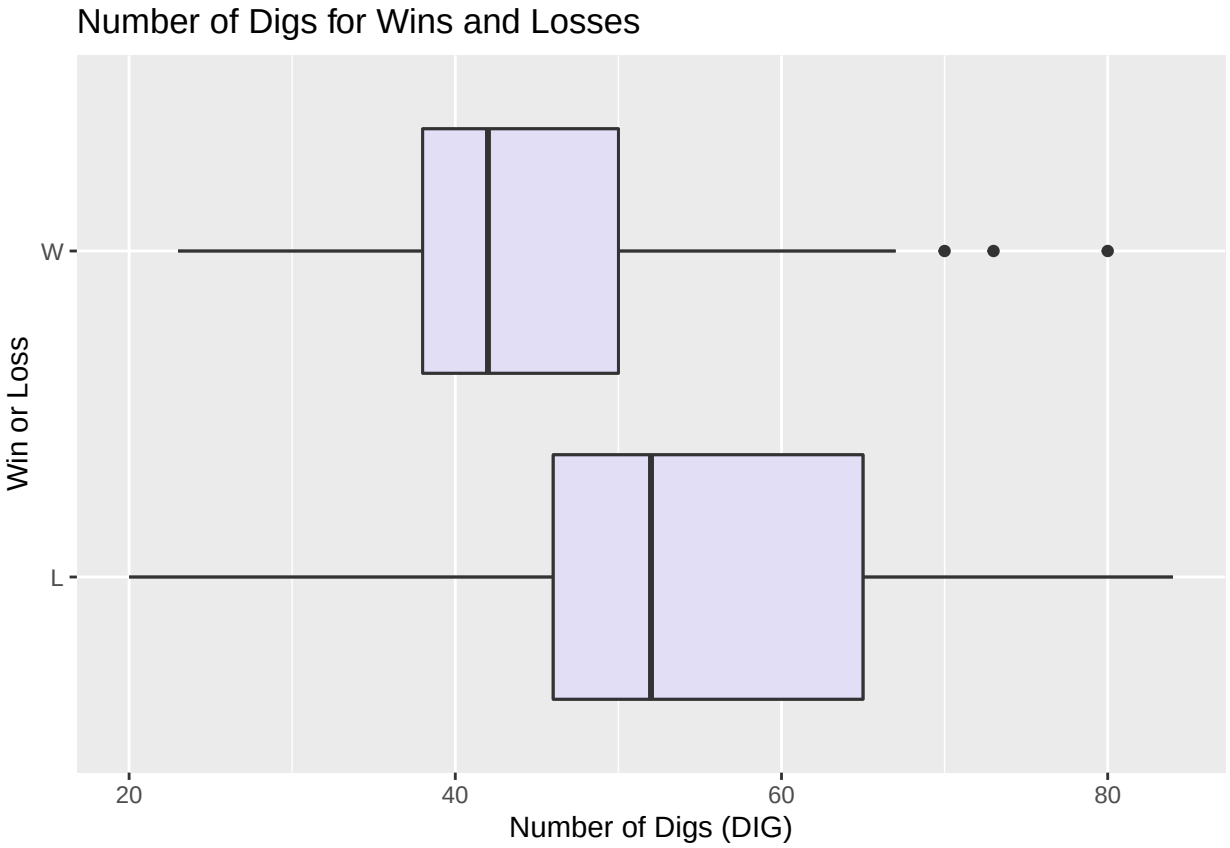


Let's modify these plots to make them more complete and visually appealing.

```
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.

1.9 Soccer

1.9.1 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.

1.9.2 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.

1.9.3 Bar Plot

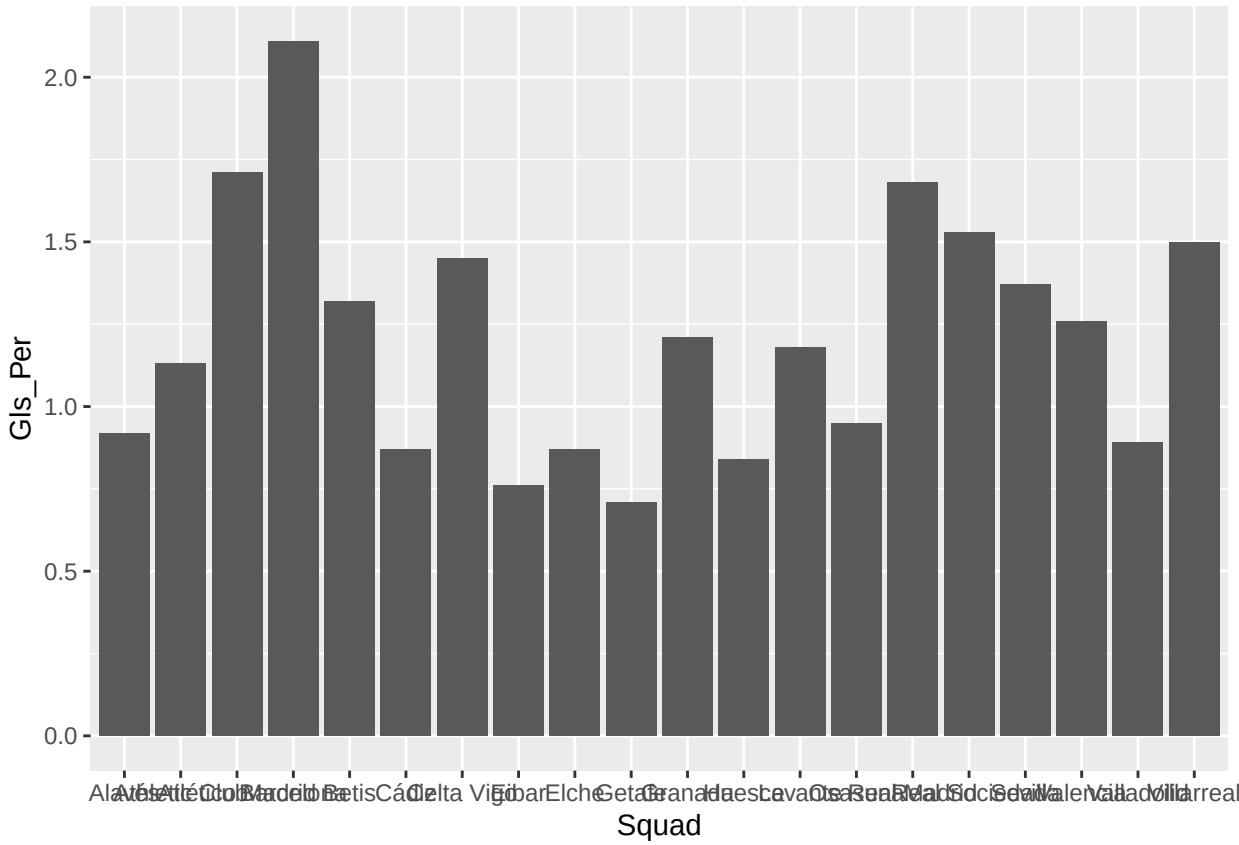
Now that we have an understanding of some basic shooting statistics, let us go through some EDA examples. For this first example, we will need to install the "worldfootballR" package.

```
library(worldfootballR)
```

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

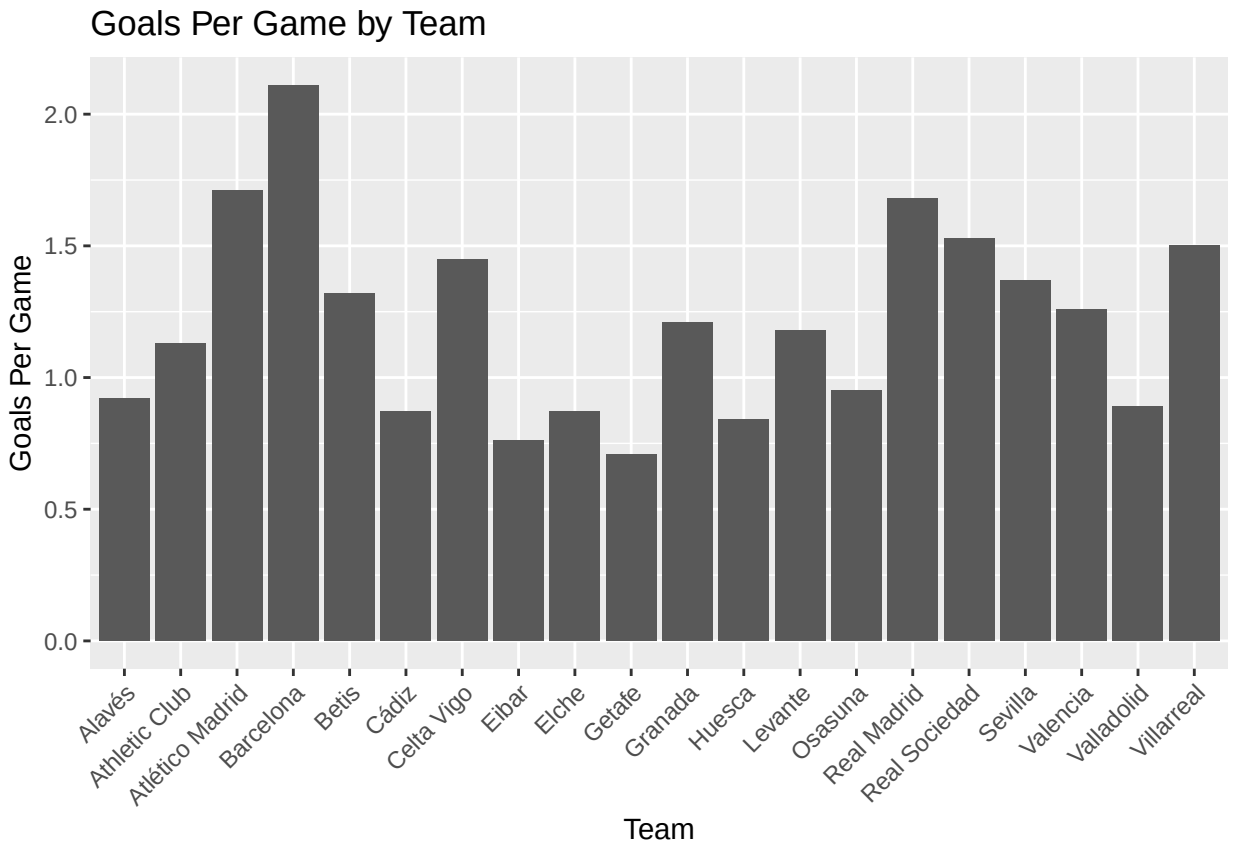
```
# Get 'Squad Standard Stats' Data
big5_2021_stats <- fb_big5_advanced_season_stats(season_end_year = 2021,
stat_type = "standard",
    team_or_player = "team")
liga_2021_stats <- big5_2021_stats[which((big5_2021_stats$Comp == "La Liga")), ]
```

```
# 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)) + geom_bar(stat = "identity")
team_goals_viz
```



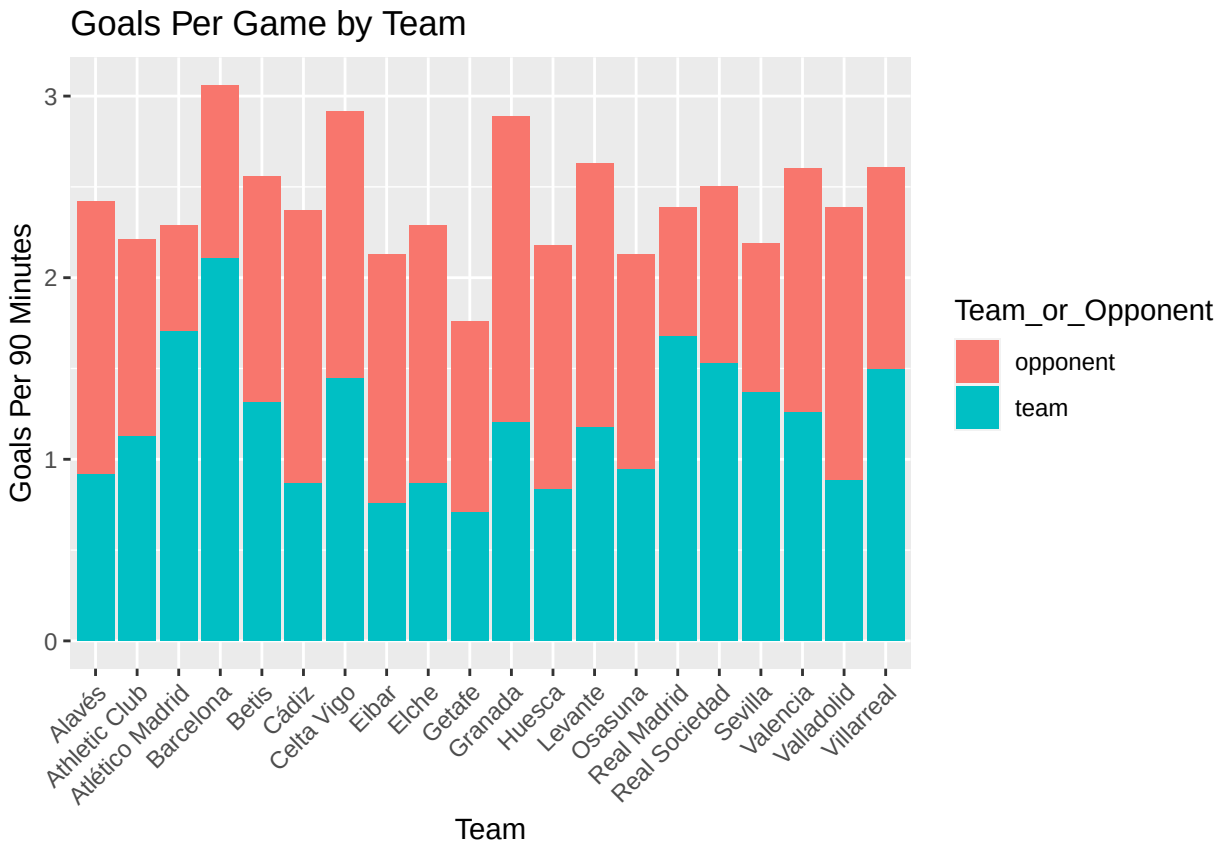
This plot is a good starting point, but still looks pretty messy. Let's add a title, change the axis titles, and rotate the axis labels so they are not overlapping over one another.

```
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
```



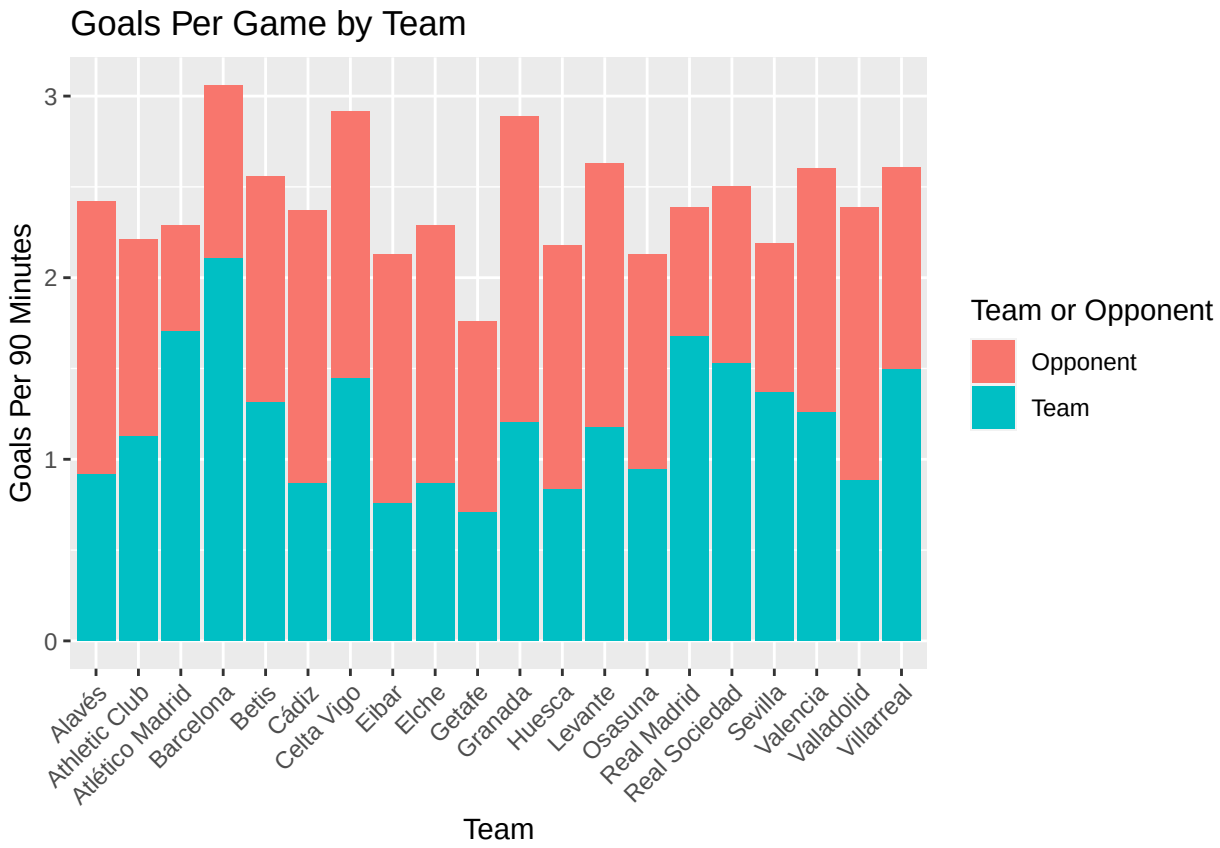
This is already looking a lot better. Now, we will add the goals scored per game *against* each team. Why is this of interest? Well, at first glance, Barcelona seems like a pretty impressive team, as they score more goals per game than any other team in the league. However, what if they also have more 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"))
```

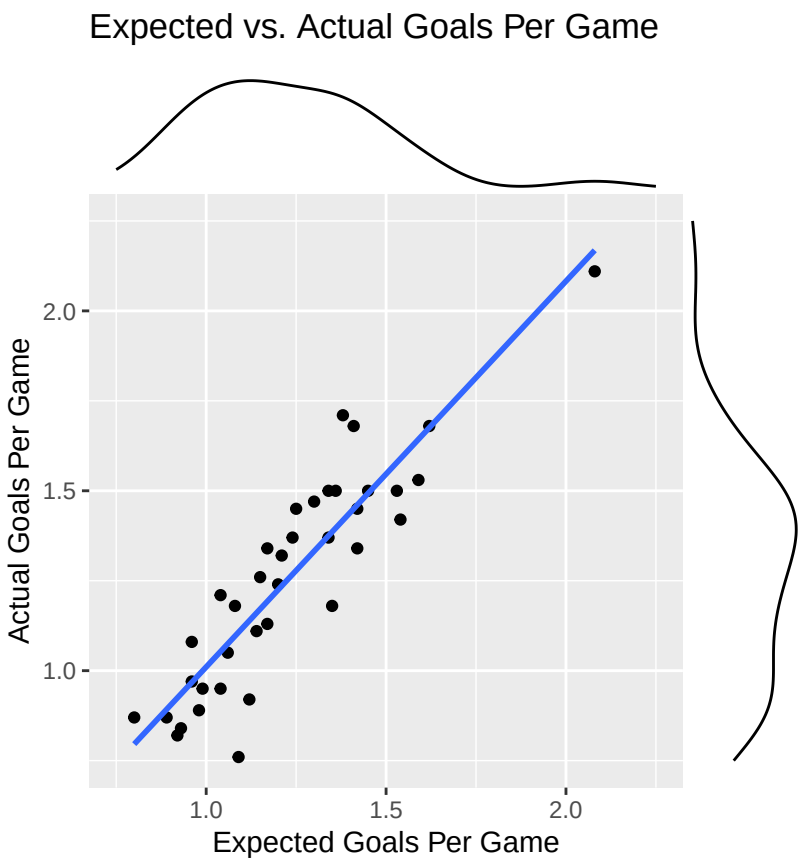


What does this graph show us? Well, we are able to see the average number of goals scored for and against each team per game. It looks like Barcelona is scoring a lot more goals than they are letting be scored against them, while other teams like Valladolid tend to have a higher proportion of goals scored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

1.9.4 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?

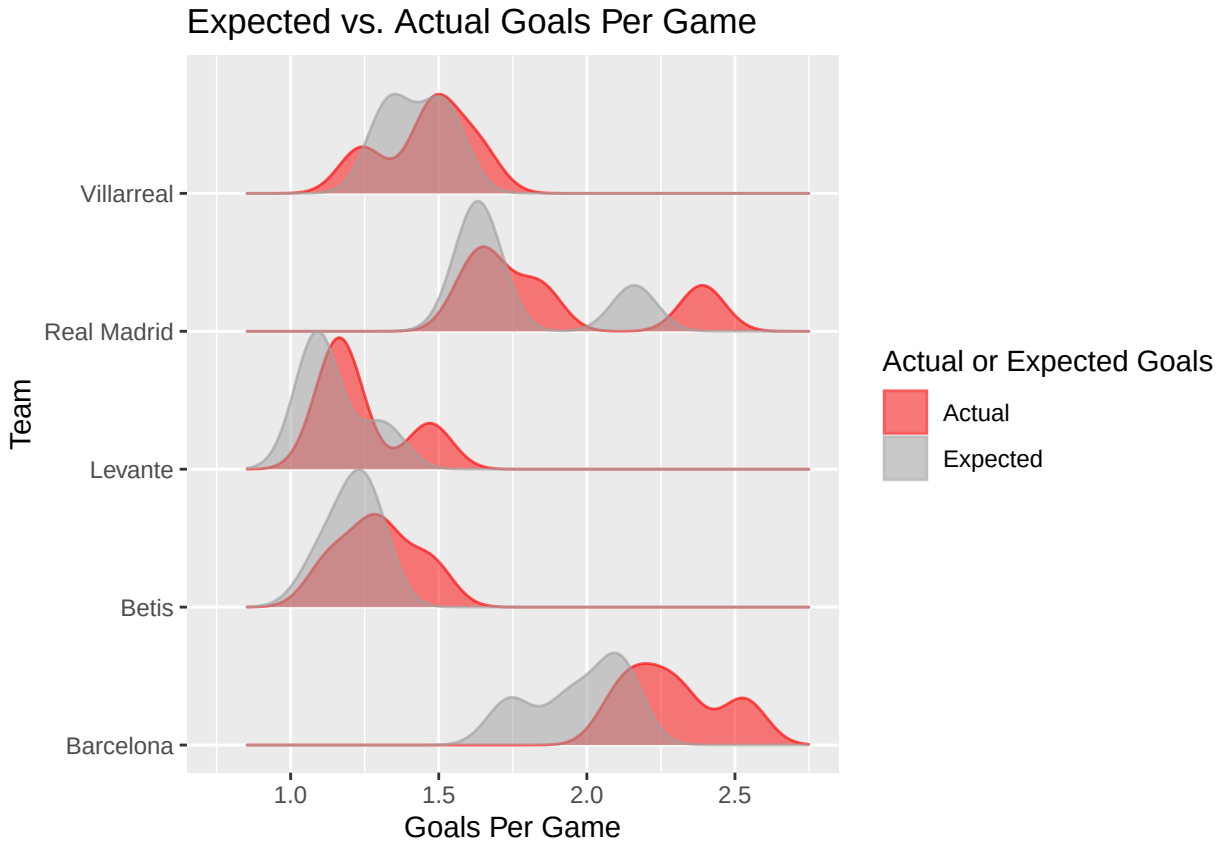
1.9.5 Density Ridges Plot

At first glance, it seems that actual goals scored per game do not differ greatly from expected goals per game. Let us look at some density plots for actual and expected goals per game for five of the top teams in LaLiga over the last four seasons. These are the top five teams as of June 21st, 2022 on www.foxsports.com.

```
library(ggridges)

# Get 'Squad Standard Stats' data for the last four seasons
top_liga_2021_stats <- read_csv("data/laliga21.csv")
top_liga_2020_stats <- read_csv("data/laliga20.csv")
top_liga_2019_stats <- read_csv("data/laliga19.csv")
top_liga_2018_stats <- read_csv("data/laliga18.csv")

top_liga_2021_stats <- top_liga_2021_stats[which(top_liga_2021_stats$Squad ==
"Real Madrid" |
    top_liga_2021_stats$Squad == "Villarreal" | top_liga_2021_stats$Squad ==
"Barcelona" |
    top_liga_2021_stats$Squad == "Levante" | top_liga_2021_stats$Squad ==
"Betis"),
```

```
    ]
top_liga_2020_stats <- top_liga_2020_stats[which(top_liga_2020_stats$Squad ==
"Real Madrid" |
    top_liga_2020_stats$Squad == "Villarreal" | top_liga_2020_stats$Squad ==
"Barcelona" |
    top_liga_2020_stats$Squad == "Levante" | top_liga_2020_stats$Squad ==
"Betis"),
    ]
top_liga_2019_stats <- top_liga_2019_stats[which(top_liga_2019_stats$Squad ==
"Real Madrid" |
    top_liga_2019_stats$Squad == "Villarreal" | top_liga_2019_stats$Squad ==
"Barcelona" |
    top_liga_2019_stats$Squad == "Levante" | top_liga_2019_stats$Squad ==
"Betis"),
    ]
top_liga_2018_stats <- top_liga_2018_stats[which(top_liga_2018_stats$Squad ==
"Real Madrid" |
    top_liga_2018_stats$Squad == "Villarreal" | top_liga_2018_stats$Squad ==
"Barcelona" |
    top_liga_2018_stats$Squad == "Levante" | top_liga_2018_stats$Squad ==
"Betis"),
    ]

# Combine all four seasons' data into one data frame
top_liga_stats <- rbind(top_liga_2018_stats, top_liga_2019_stats,
top_liga_2020_stats,
    top_liga_2021_stats)

goals_act <-
data.frame(top_liga_stats$Gls_Per[which(top_liga_stats$Team_or_Opponent ==
    "team")])
goals_act$team <- top_liga_stats$Squad[which(top_liga_stats$Team_or_Opponent ==
"team")]
goals_act$exp_or_act <- "actual"
goals_act$year <-
top_liga_stats$Season_End_Year[which(top_liga_stats$Team_or_Opponent ==
    "team")]
colnames(goals_act)[1] <- "stats"
goals_exp <-
data.frame(top_liga_stats$xG_Per[which(top_liga_stats$Team_or_Opponent ==
    "team")])
goals_exp$team <- top_liga_stats$Squad[which(top_liga_stats$Team_or_Opponent ==
"team")]
goals_exp$exp_or_act <- "expected"
goals_exp$year <-
top_liga_stats$Season_End_Year[which(top_liga_stats$Team_or_Opponent ==
    "team")]
```

```
colnames(goals_exp)[1] <- "stats"
goals <- rbind(goals_act, goals_exp)

# Plot density ridges
ggplot(data = goals) + geom_density_ridges(aes(x = stats, y = team, fill =
exp_or_act,
    color = exp_or_act), alpha = 0.5, scale = 1) + scale_x_continuous(limits =
    c(0.75,
    2.75)) + scale_y_discrete(expand = expand_scale(add = c(0.2, 1))) +
    ggtitle("Expected vs. Actual Goals Per Game") +
    xlab("Goals Per Game") + ylab("Team") + scale_fill_cyclical(name = "Actual or
    Expected Goals",
    labels = c("Actual", "Expected"), guide = "legend", values = c("#FF0000A0",
    "#A0A0A0A0")) +
    scale_color_cyclical(name = "Actual or Expected Goals", labels = c("Actual",
        "Expected"), guide = "legend", values = c("#FF0000A0", "#A0A0A0A0"))
```



Let us break down exactly what this visual is showing us. We are looking at the density of expected and actual goals per game for the top five teams in LaLiga, over the last four seasons (with the last season ending in 2021). We can see that Barcelona is typically scoring more goals than what is expected of them, as the density of actual goals is condensed around higher goal numbers than the density of expected goals. Villarreal, however, is performing just as well as what is expected of them based on expected and actual goals scored.

Chapter 2

Probability

Chapter Preview

Probability is the study of randomness. In this chapter, we will define probability, learn rules of probability, and apply these rules to sports data.

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.

Example 2.2.

2.2 Set Theory

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

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 of "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

- $P(\emptyset) = 0$
- $P(A^c) = 1 - P(A)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(A \cup B \cup C) =$
  $P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$
- $P([A \cup B]^c) = P(A^c \cap B^c)$
- $P([A \cap B]^c) = P(A^c \cup B^c)$

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 2.8. In 2001, Barry Bonds broke the single season home run record with 73 home runs. In this season, he had 664 plate appearances, 156 hits, 177 walks and 9 hit by pitches. Given that Bonds reached base (via hit, walk, or HBP), what was the probability that he got a hit?

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

Definition 2.10. Events $A_1, A_2, ..., A_n$ 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* $A_1, A_2, \ldots, A_n$ *form a partition of* Ω*, then:* $P(B) = P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \ldots P(B|A_n)P(A_n)$

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* $B_1, B_2, \ldots, B_n$ *form a partition of* Ω*, then:* $P(B_k|A) = \frac{P(B_k \cap A)}{P(A)} = \frac{P(A|B_k)P(B_k)}{P(A|B_1)P(B_1)+P(A|B_2)P(B_2)+\ldots+P(A|B_n)P(B_n)}$

Example 2.9. Over the course of a season, a hockey player scored a goal 30% of the time during a home game, and $P(\textit{player scores} \mid \textit{away game}) = .18$. 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?

Example 2.10. Injured Baserunner (Prob) A runner on first base with 2 out and nobody else on base will attempt to steal second base on the first pitch 70% of the time if he is fully healthy but only 10% of the time if he is playing through an injury. Assume that 80% of the player population is healthy. You see a randomly selected runner not attempt a steal in this situation. What is the probability that the runner is playing through an injury?

From Bayes Theorem:

Pr(Injury given No Steal) = Pr(No Steal given Injury)*Pr(Injury)/P(No Steal).

Pr(No Steal given Injury) = 1 - Pr(Steal given Injury) = 0.9.

Pr(Injury) = 1- Pr(Healthy) = 0.2.

Pr(No Steal) = Pr(No Steal given Injury)*Pr(Injury)+Pr(No Steal given Healthy)*Pr(Healthy).

Pr(No Steal) = 0.9*0.2+0.7*0.8 = 0.74.

Therefore Pr(Injury given No Steal) = 0.9*0.2/0.74 = 0.243.

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 $1! \equiv 1$.

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: $P_{n,k} = \frac{n!}{(n-k)!}$

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

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 $n_1 \cdot n_2 \cdot ... \cdot n_k$ possible k-tuples.*

2.5 Odds and Gambling

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

2.6 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 2.11.

Example 2.12.

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 2.13. 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.

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 2.14. 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 2.15. For the Rockies/Cubs four game series example, calculate $Var(X)$.

2.7 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

2.7.1 Binomial RVs

Definition 2.19. 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 2.16. 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 wins 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 2.17. 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.

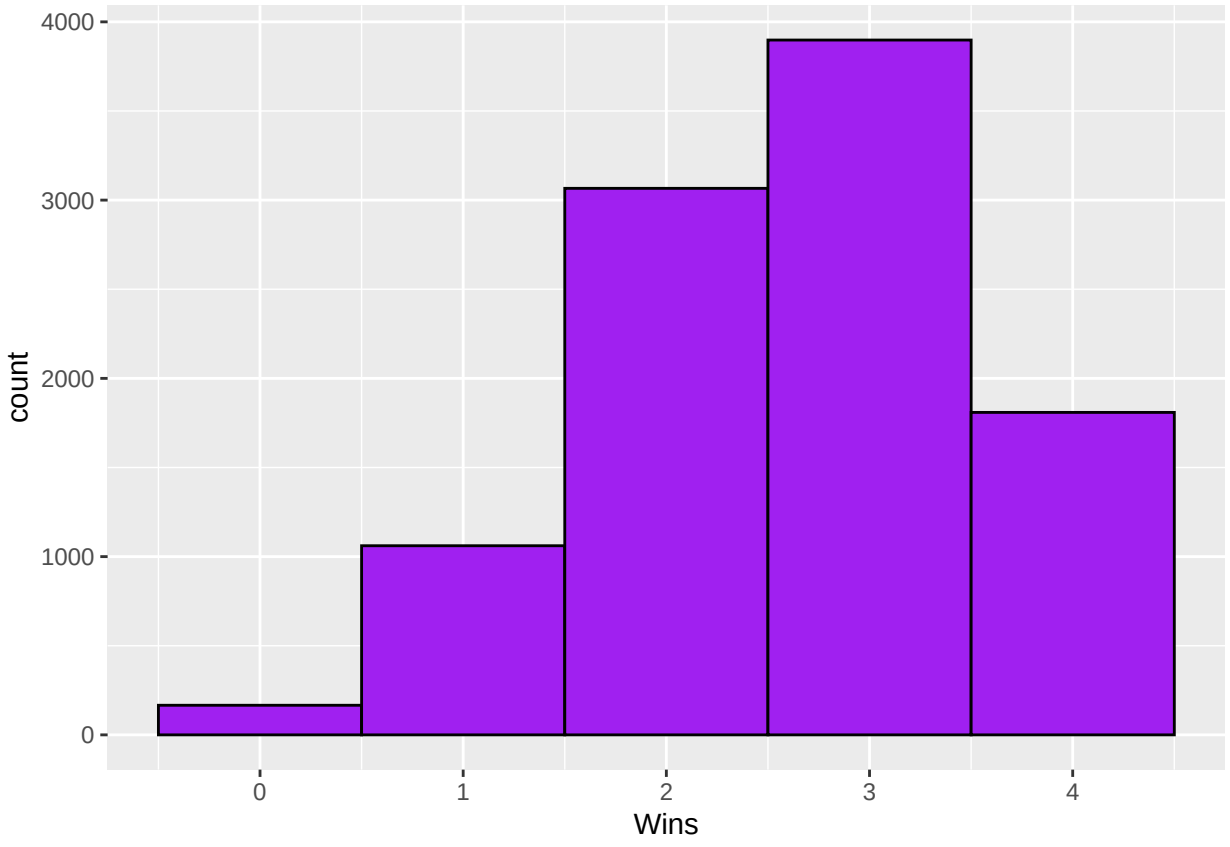
```
set.seed(2020)
rockies_wins <- rbinom(n = 10000, size = 4, prob = 0.65)
mean(rockies_wins)
```

```
## [1] 2.6123
```

```
var(rockies_wins)
```

```
## [1] 0.9110798
```

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



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

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

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

2.7.2 Geometric RVs

Definition 2.20. 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 2.18. 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
```

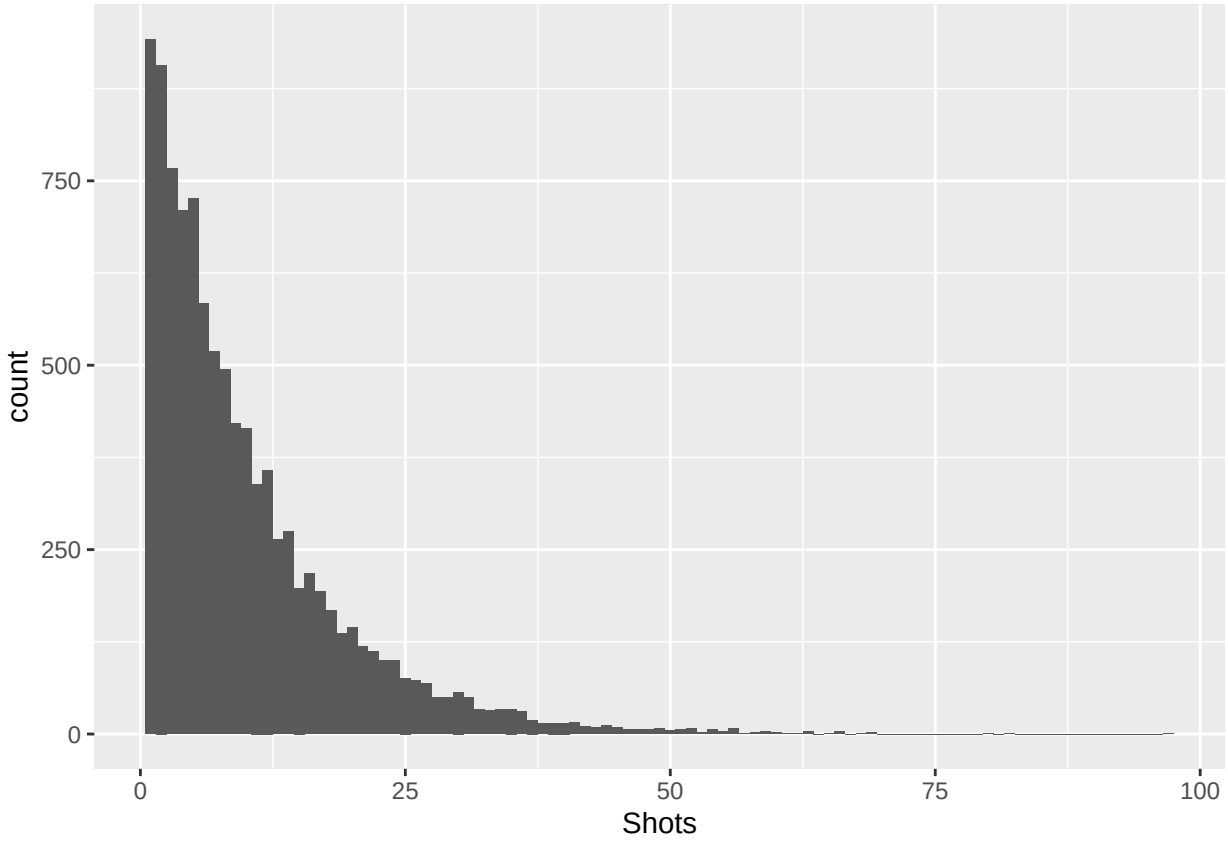
```
mean(first_success)
```

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

The mean of this sample of variables is 10.827, 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)
```



2.7.3 Poisson RVs

Definition 2.21. 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 2.19. 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.

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

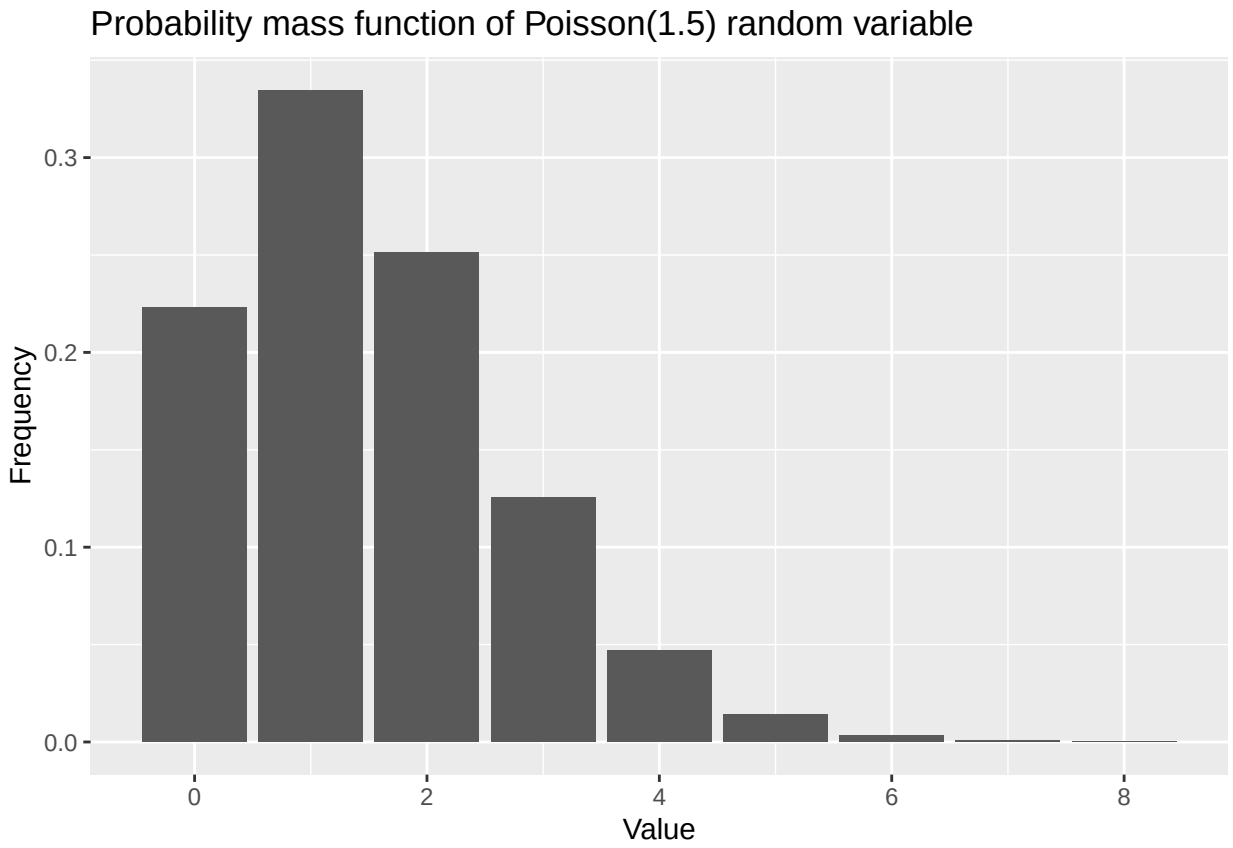
```
## [1] 0.2231302
```

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

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

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

```
ggplot(transform(data.frame(x = c(0:8)), y = dpois(x, lambda = 1.5)), aes(x, y))
+
    geom_bar(stat = "identity") + labs(x = "Value", y = "Frequency", title =
    "Probability mass function of Poisson(1.5) random variable")
```



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

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

library("kableExtra")

goals <- c(0:4, "5 or more")
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)

rapids.data %>%
    kbl() %>%
    kable_styling()
```

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

goals	actual_frequency	actual_proportion	expected_frequency	expected_proportion
0	6	0.1764706	7.6	0.2231302
1	14	0.4117647	11.4	0.3346952
2	7	0.2058824	8.5	0.2510214
3	6	0.1764706	4.3	0.1255107
4	0	0.0000000	1.6	0.0470665
5 or more	1	0.0294118	0.6	0.0185759

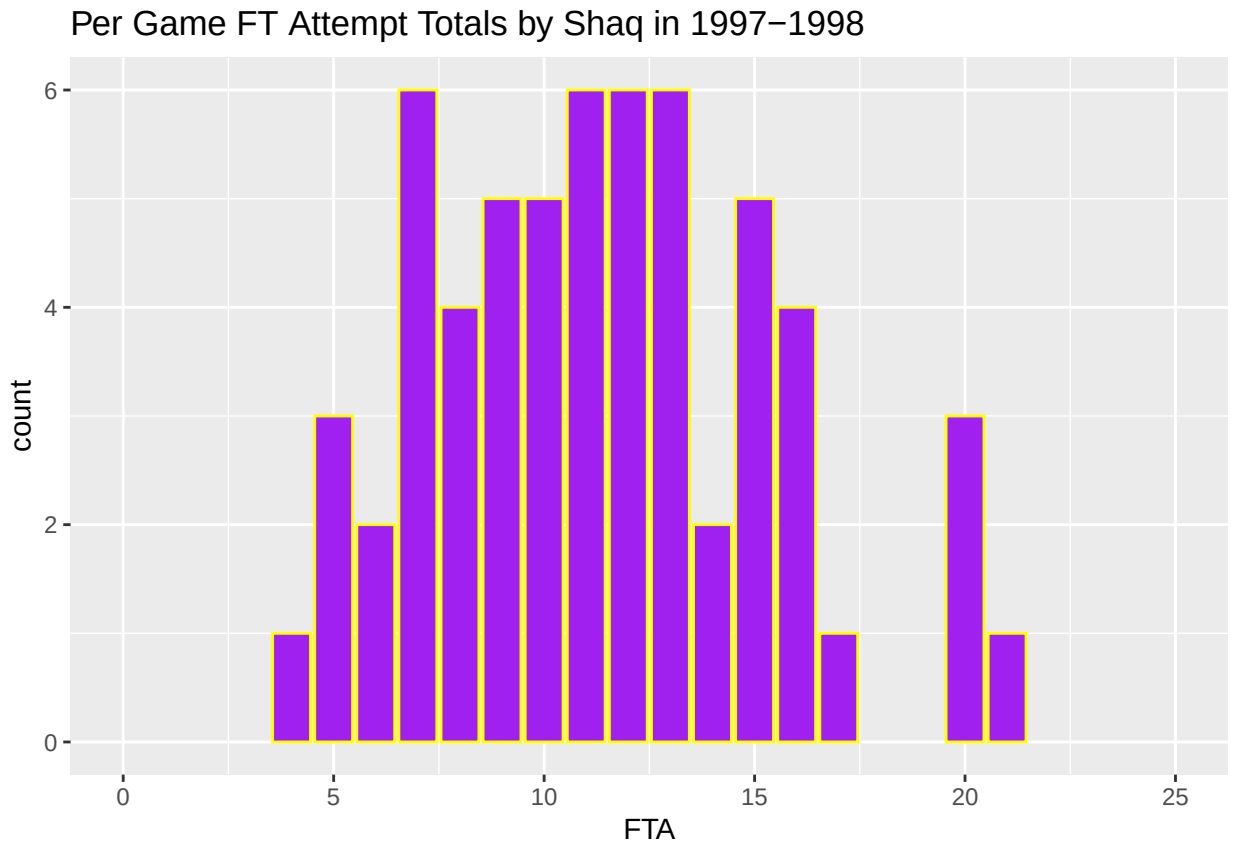
(c) Even if the true population distribution of 2021 Rapids goals was truly a Poisson(1.5) random variable, why might the actual distribution of their goals differ from the probability mass function?

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

Example 2.20. 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?

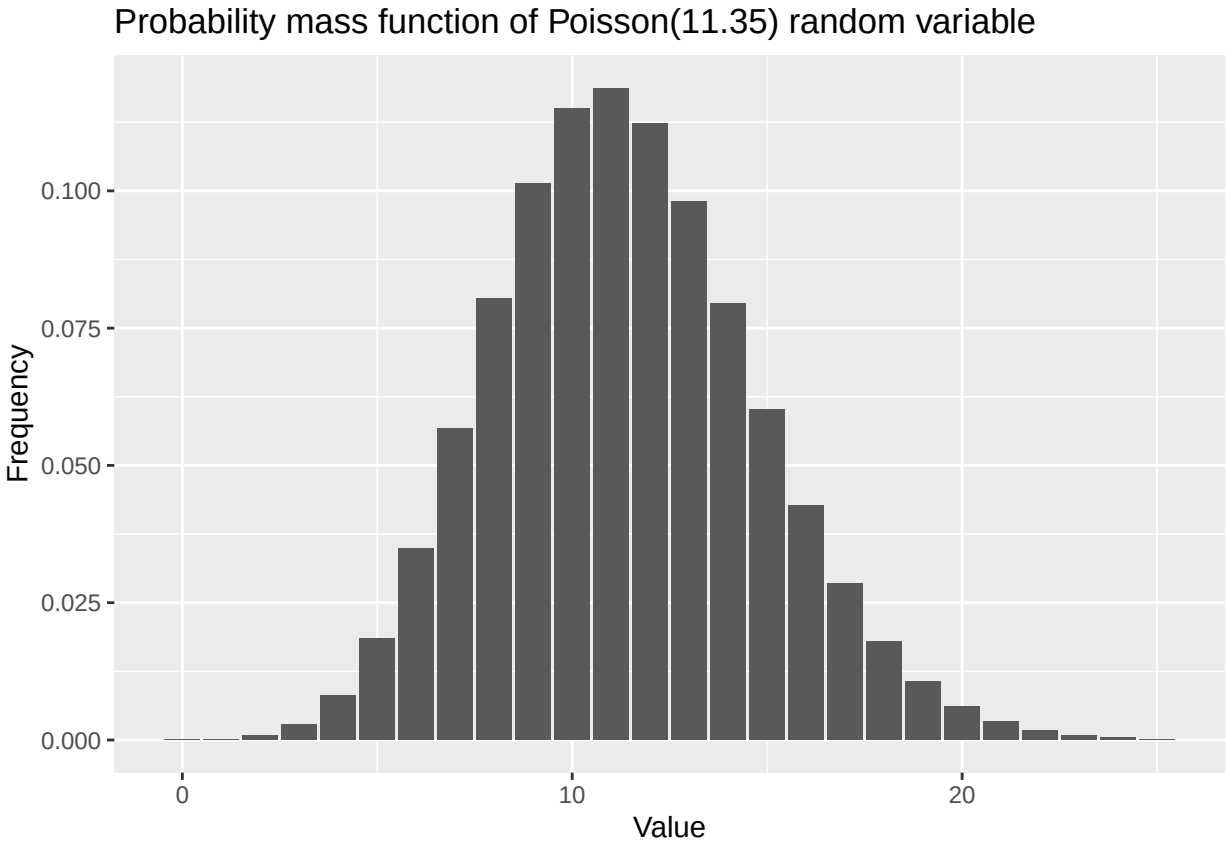
(a) Plot the data.

```
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.

```
ggplot(transform(data.frame(x = c(0:25)), y = dpois(x, lambda = 11.35)), aes(x,
y)) +
    geom_bar(stat = "identity") + labs(x = "Value", y = "Frequency", title =
    "Probability mass function of Poisson(11.35) random variable")
```



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

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

```
shaqFTA <- shaq9798 %>%
    select(FTA)
var(shaqFTA)
```

```
##          FTA
## FTA 16.33305
```

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

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

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

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

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

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

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

2.7.4 Negative Binomial RVs

Definition 2.22. 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 $k \geq 0$,

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

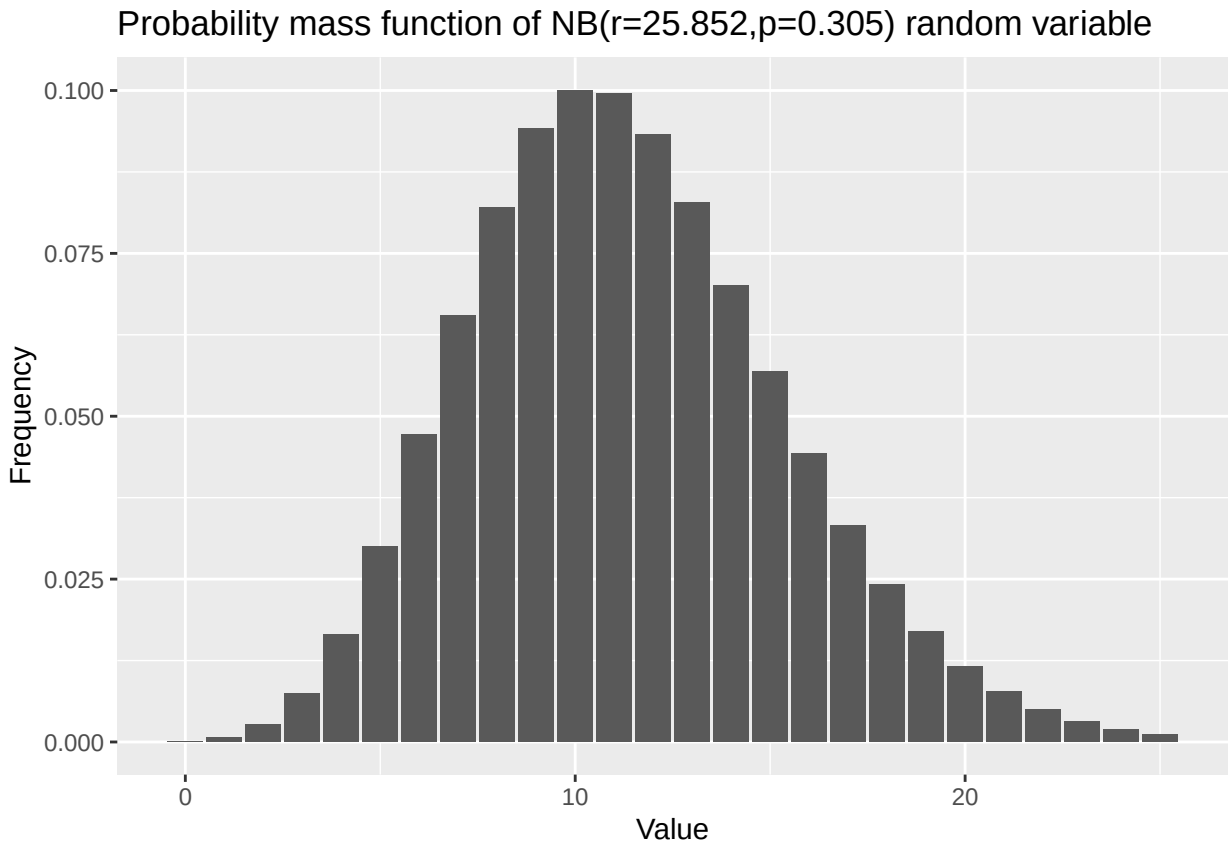
(a) Find an appropriate choice of parameters, r and p.

```
shaq.mean <- mean(shaqFTA$FTA)
shaq.var <- var(shaqFTA$FTA)
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 $prob = 1 - p$.

```
ggplot(transform(data.frame(x = c(0:25)), y = dnbinom(x, size = rhat, prob = 1 -
    phat)), 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$FTA)
shaq.var <- var(shaqFTA$FTA)
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)/nrow(shaqFTA)
shaq20
```

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

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

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

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

2.7.5 Normal RVs

Definition 2.23. 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 2.22. 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 Arenado?

(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 2.23. 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 explaination: https://www.mlb.com/glossary/advanced-stats/wins-above-replacement

We want Pr(X>Y) or Pr(X-Y>0).
Let Z = X-Y.
E[Z]=1.5 Var(Z)=5 Pr(Z>0)=1-Pr(Z $\leq$ 0)

```
# Calculate probability Z<=0
pr <- pnorm(0, 1.5, sqrt(5))
print(1 - pr)
```

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

The Probability that Player X outperforms Player Y is 0.7488.

2.8 Extra Stuff

2.8.1 Sets and Conditional Probability

100 sports fans in Colorado were polled and it was found that 64 had attended either a Denver Nuggets or Colorado Avalanche game at Ball Arena (formerly Pepsi Center). 34 people had seen only a Nuggets game, while 17 had seen both a Nuggets and an Avalanche game.

Q: How many people saw an Avalanche game but not a Nuggets game?

A: 64 - 34 - 17 = 13

Q: What is the probability that a randomly selected person in the poll had been to a Nuggets game?

A: (34 + 17) / 100 = .51

Q: What is the probability that a randomly selected person that had been to a game at Ball Arena had been to a Nuggets game?

A: (34 + 17) / 64 = .797

Q: What is the probability that a randomly selected person had been to a Nuggets game given they had been to an Avalanche game?

A: 17 / (17 + 13) = .567

2.8.2 Binomials and Multinomials

Suppose we are curious about probabilities regarding the results of a soccer team's next five games.

Wait!!! A soccer game has three possible outcomes (win, lose, draw)! We can't use the binomial distribution, since it limits us to two possible outcomes!

It depends. If we are interested in the probability that a soccer team wins 2 of their next 5 games, we can use the binomial distribution. We can create the following partition of the sample space of outcomes: (Win) and (Win^C), where the second set includes both losing and drawing.

Then, the formula would be represented as:

$\binom{5}{2}\ P(Win)^2\ P(Win^C)^{(5-2)}$

If we are interested in the probability of the team winning two of the next five games, drawing two, and losing one, we cannot use the binomial theorem. That involves three outcomes, and would be represented as a multinomial.

2.8.3 Expectation - Baseball

The expectation of a discrete random variable is a weighted average. The "weights" are the probabilities of the possible values of the variable.

Consider the following table, which shows the number of career hits by type for the all-time Major League Baseball leader in total bases, Hank Aaron.

The expected number of bases for a Hank Aaron hit is the sum of the number of bases attained for each hit multiplied by the relative frequency of the occurrence of that type of hit.

Hit_type	Number_bases	Hit_Frequency	Hit_Proportion
Single	1	2294	0.6083267
Double	2	624	0.1654734
Triple	3	98	0.0259878
Home Run	4	755	0.2002122

$1 \cdot \frac{2294}{3771} + 2 \cdot \frac{624}{3771} + 3 \cdot \frac{98}{3771} + 4 \cdot \frac{755}{3771} = 1.18181$

This is the same process that is occurring whenever we calculate the expectation of any discrete random variable. Recall the formula for expectation is $E[X] = \sum_{x \in \Omega} \; x \cdot p(x)$. Each value in the sample space is "adjusted" by the probability of that value, then the sum of all values in Ω is taken to arrive at the weighted average, or expected value, of the random variable.

2.8.4 Basketball Scenario

You are the coach of a basketball team that is down two points with one second remaining in the fourth quarter. During a timeout, you are considering the best play to call for your team. The first option is a three-point shot attempt, which you estimate has a 30% chance of succeeding. The second option is a two-point shot attempt, which has a 50% chance of making the field goal, a 30% chance of missing it and ending the game, and a 20% chance the shooter will miss but be fouled, in which case the shooter's free throw success will follow a $Bin(2, .8)$ random variable. Finally, you estimate that your team's probability of winning the game in overtime is .45.

Assume the above situations are exhaustive (i.e., the other team will not get another possession, no fouls will be called before the ball is put in play, lightning will not hit the arena and postpone the game, etc.). Which of the two plays should you call to maximize the win probability for your team?

A: The probability of winning the game with the three-point shot attempt is .3. If the two-point shot attempt is called for, there is a .5 probability of making the field goal and a $(.2)(.8)(.8) = .128$ probability that the foul is called and both free throws are made. Thus, the total probability of scoring two points and sending the game to overtime is .628. Then, the probability of winning the game in OT after tying it in regulation is $(.628)(.45) = .2828$. This is less than .3, so shooting the three-pointer is the option that maximizes the win probability, given these situational probabilities.

Q: What is the minimum estimated overtime win probability to make calling for the two-point play the better option?

A: $P(\textit{score 2 points in regulation}) \cdot P(\textit{win in OT}) > P(\textit{win in regulation})$
$.628 \cdot P(\textit{win in OT}) > .3$
$P(\textit{win in OT}) > .478$

2.8.5 Multiple Probability Distributions - Basketball

Suppose the number of points scored by a basketball player follows a Poisson(12) random variable, the number of rebounds by a Poisson(7) distribution, and assists by a Discrete Uniform(2, 11), independently of each other.

Q: What is the probability that this player records a points, rebounds, assists triple-double in a game?

A: $P(Triple\ Double) = P(Points \geq 10\ \cap\ Rebounds \geq 10\ \cap\ Assists \geq 10)$

```
ppois(9, lambda = 12, lower.tail = F)
```

```
## [1] 0.7576078
```

$P(Points \geq 10) = P(Poisson(12) \geq 10) \approx .758$

```
ppois(9, lambda = 7, lower.tail = F)
```

```
## [1] 0.1695041
```

$P(Rebounds \geq 10) = P(Poisson(7) \geq 10) \approx .170$

$P(Assists \geq 10) = P(Discrete\ Uniform(2, 11) \geq 10) = .2$

Since the events are independent, we can multiply their probabilities. The probability of the player scoring the triple-double is $(.758)(.170)(.2) = .0257$.

Q: Your friend offers you 4 to 1 that the player will not record a triple-double in their next 10 games. With the knowledge that the athlete's performance in a game is unaffected by performances in previous games, would you take the bet?

A: $P(no\ triple\ double) = 1 - .0257 = .9743$, so $P(no\ triple\ double\ in\ next\ 10\ games) = (.9743)^{10} = .771$

The odds of no triple-double are $\frac{.771}{1-.771} = 3.37$, so the bet of no triple-double at 4 to 1 odds is favorable.

answers may vary for following questions

Q: What differences do you notice between the actual results and the expected values based on the Poisson random variable?

A: There were fewer games in which the Rapids scored 4 or more goals than the model would indicate, yet the Rapids were shut out less often than the model would indicate.

Q: 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?

A: 34 is a relatively small sample size; random variables may not coincide with their expected values for finite sample sizes.

Q: What are the advantages of using the Poisson distribution to model Major League soccer goals? What are the disadvantages?

A: Poisson random variables can take on the natural numbers (including zero), which aligns with the number of goals that can be scored in a match. One disadvantage is that it is possible for a Poisson to take on values that are not realistic for the situation, such as double-digit integers or higher. Only one game in MLS history has had a team score more than seven goals in a game. However, when λ is small (such as 1.5), these extreme values are relatively unlikely.

2.8.6 Law of Total Probability - Hockey

Over the course of a season, a hockey player scored a goal 30% of the time during a home game, and $P(player\ scores \mid away\ game) = .18$. Assume all games are either home or away.

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

A: $P(score) = P(score|home)P(home) + P(score|away)P(away) = .3(.5) + .18(.5) = .24$

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

A: $P(score) = P(score|home)P(home) + P(score|away)P(away) = .3(\frac{2}{3}) + .18(\frac{1}{3}) = .26$

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

A: $P(score) = P(score|home)P(home) + P(score|away)P(away) = .3(\frac{2}{5}) + .18(\frac{3}{5}) = .228$

2.8.7 Law of Total Probability - Baseball

You work in the front office of a professional baseball club and have just learned that a certain prospect hits .200 against left-handed pitchers and .400 against right-handed pitchers (their overall batting average is unknown). The general manager of the team overhears you talking about the .400 statistic of the player and becomes very exited that they have the chance to draft a .400 hitter. What would you say to caution the GM that the player might not be a remarkable hitter?

A: We don't know the proportion of the player's at-bats that came against left-handed pitchers versus right-handed pitchers. If we want to know the player's batting average unconditional on the type of pitcher they are facing, we have to adjust $P(hit \mid left-handed\ pitcher)$ by $P(left-handed\ pitcher)$ and $P(hit \mid right-handed\ pitcher)$ by $P(right-handed\ pitcher)$ before adding them to determine $P(hit)$. For example, if 90% of the player's at-bats were against left-handed pitchers, then their overall batting average is a pedestrian .220.

Other possible issues: low sample size of player's at-bats, the fact that pro pitchers will be harder to hit against than non-pros

Chapter 3

Monte Carlo Simulation

3.1 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:

- Initialize vectors and variables
- Run a simulation and calculate the estimate of interest
- Save the estimate
- Run the simulation "n" times
- Analyze the estimates 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 3.1. Simulate 10 overtime coin tosses with and without using `set.seed()` and compare the results

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

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

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

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

```
# 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"
```

```
# 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 3.2. 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?

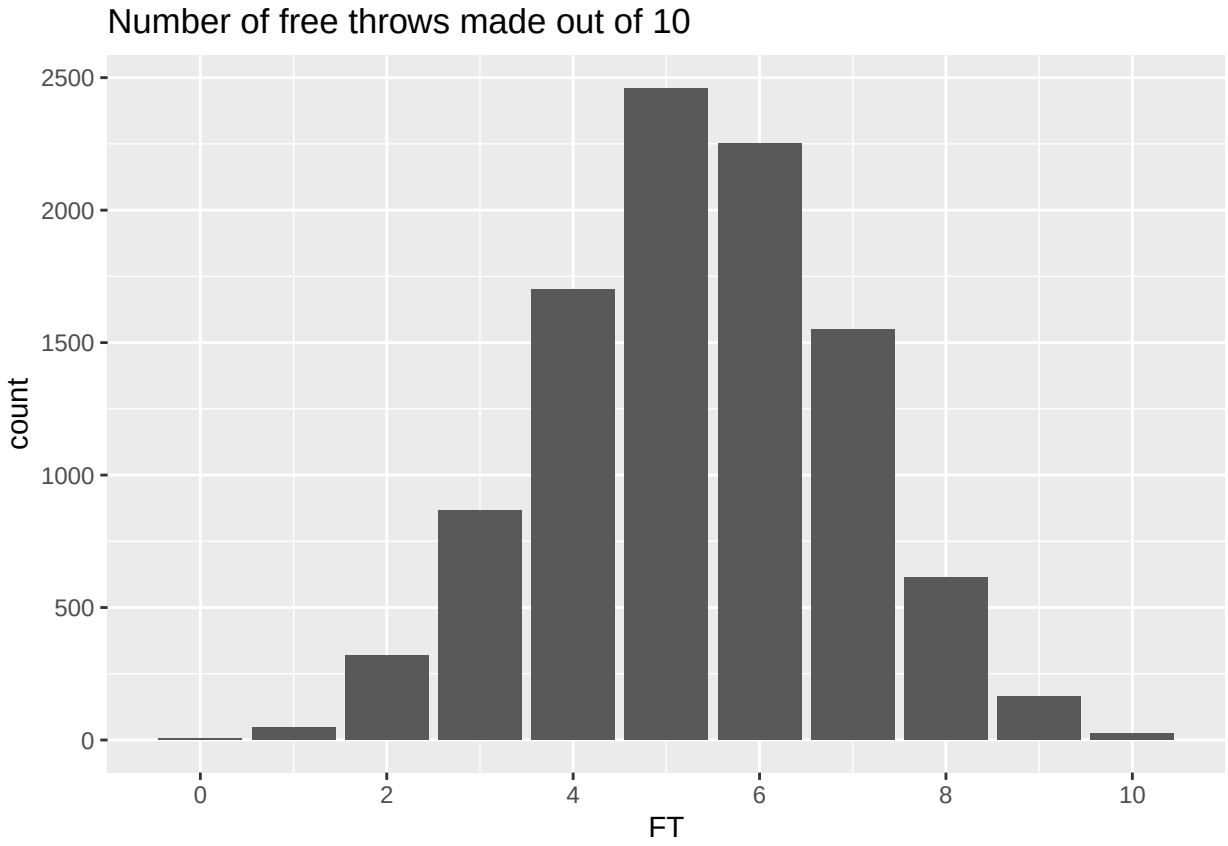
In this case, we can calculate the exact probability of interest using binomial random variable.

```
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)
}

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))
```




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

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

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

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

```
set.seed(1)

# Number of Simulations
n.sims <- 1e+05

# 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] 0.00174
```

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

Example 3.3. In 1997-1998 with the Los Angeles Lakers, Shaq attempted an average of 11.35 free throws per game with a standard deviation of 4.04. While with the Lakers, Shaq played in an average of 63.6 games per year with a standard deviation of 10.6. Create a simulation to model the season total number of free throw attempts that Shaq would have while with the Lakers.

Note: his actual season totals of free throws attempted while with the Lakers were: 479, 681, 498, 824, 972, 712, 725, 676

Let's model the number of games that Shaq played in as a Binomial random variable. There are 82 regular sesaon games, so let $n = 82$. Shaq played in an average of 63.6 games, so let $p = \frac{64.25}{82} = 0.784$. Shaq played in about 78% of the games during his career with the Lakers.

For the number of free throw attempts per game, we could model this as a Poisson random variable or a Negative Binomial random variable. As noted in the previous chapter, the variance of Shaq's FT attempts is a fair bit greater than the mean which means that it is overdispersed. Negative Binomial may be a more appropriate model than a Poisson. From last chapter, we found $\hat{r} = 25.85$ and $\hat{p} = 0.305$.

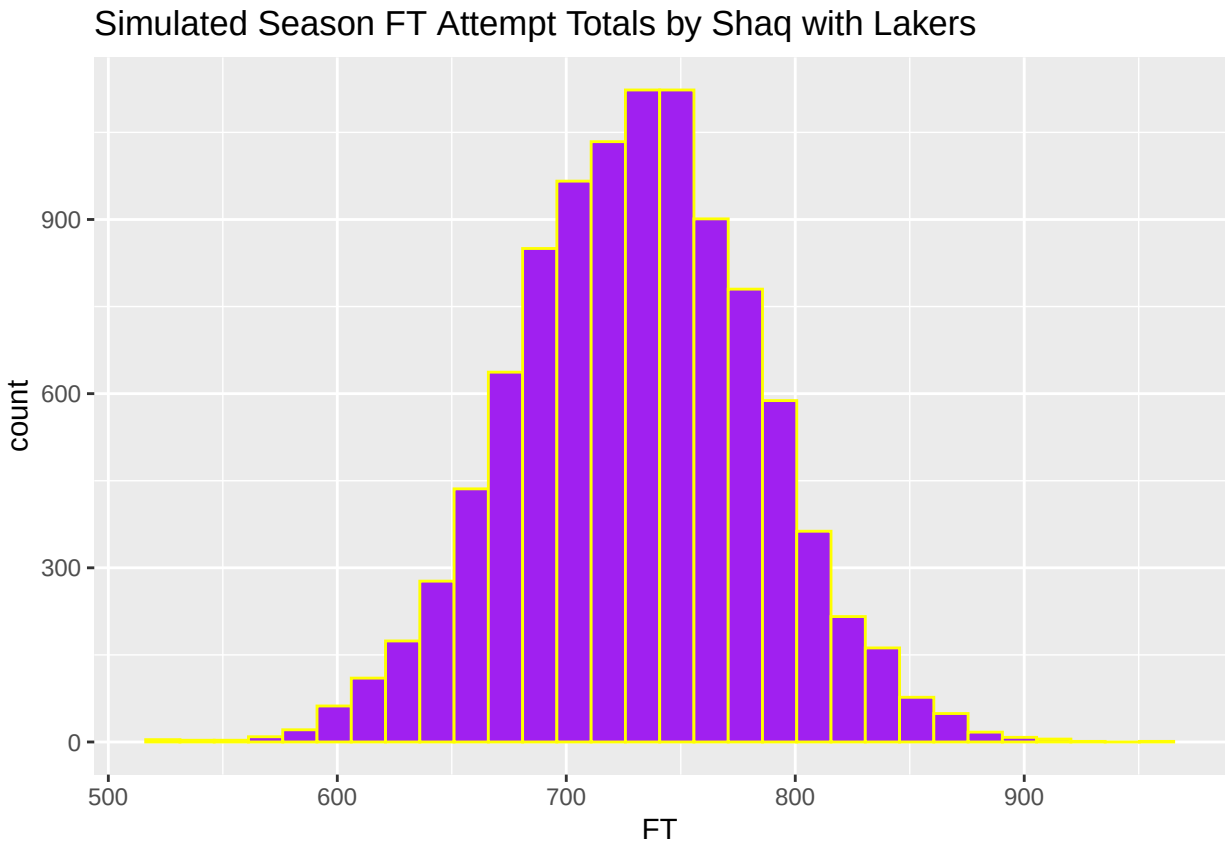
```
set.seed(2020)
n.sims <- 10000
FT <- rep(0, n.sims)

# simulate the number of games played in a season, round to a whole number
games.sim <- rbinom(n = n.sims, size = 82, prob = 0.784)

# number of games can't exceed 82 games (regular season total)
games.sim[games.sim > 82] = 82

for (i in 1:n.sims) {
    # simulate the season total FT attempts in each simulation
    temp <- rnbinom(n = games.sim[i], size = 25.85, prob = 1 - 0.305)
    FT[i] <- sum(temp)
}

# Simulated mean and SD of season totals of free throw attempts
c(mean(FT), sd(FT))
```

```
## [1] 730.47550  53.20566
```

```
# Actual mean and SD of season totals of free throw attempts
FT.actual <- c(479, 681, 498, 824, 972, 712, 725, 676)
FT.actual.mean <- mean(FT.actual)
FT.actual.var <- var(FT.actual) * 7/8  # population variance
FT.actual.sd <- sqrt(FT.actual.var)
c(FT.actual.mean, FT.actual.sd)
```

```
## [1] 695.8750 150.2393
```

```
FT %>%
    as.data.frame() %>%
    ggplot(aes(x = FT)) + geom_histogram(bins = 30, color = "yellow", fill =
    "purple") +
    ggtitle("Simulated Season FT Attempt Totals by Shaq with Lakers")
```



Notice that the mean of our simulation is somewhat close to Shaq's true season average number of free throw attempts but the variance of the simulation is far too low.

We can also simulate data using resampling. In this case, rather than simulating random variables according to a distribution, we can use our actual data as a sampling distribution.

Example 3.4. Using resampling, simulate the number of free throws Shaq would attempt while with the Lakers. Compare the mean and variance of the simulation to Shaq's actual statistics.

```
set.seed(2020)
n.sims <- 10000
FT <- rep(0, n.sims)
shaq.games <- c(51, 60, 49, 79, 74, 67, 67, 67)
shaq.FTA <- read_csv("data/shaqFT.csv", col_names = FALSE)
shaq.FTA <- shaq.FTA$X1

# sample (with replacement) from Shaq's FTA game totals
games.sim <- sample(x = shaq.games, size = n.sims, replace = T)

for (i in 1:n.sims) {
    # sample (with replacement) from Shaq's FTA game totals
    temp <- sample(x = shaq.FTA, size = games.sim[i], replace = T)
    FT[i] <- sum(temp)
}
```

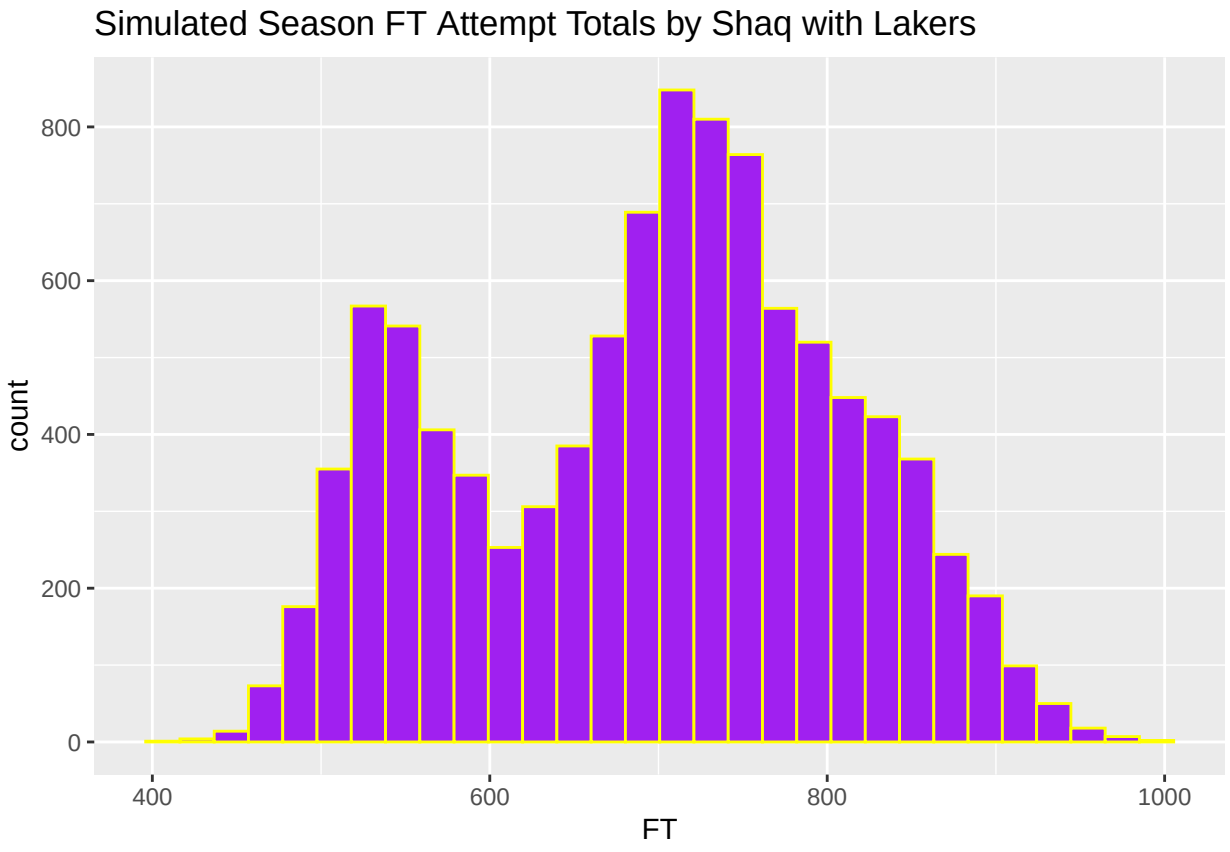
```
# Simulated mean and SD of season totals of free throw attempts
c(mean(FT), sd(FT))
```

```
## [1] 695.8883 112.6666
```

```
# Actual mean and SD of season totals of free throw attempts
FT.actual <- c(479, 681, 498, 824, 972, 712, 725, 676)
FT.actual.mean <- mean(FT.actual)
FT.actual.var <- var(FT.actual) * 7/8  # population variance
FT.actual.sd <- sqrt(FT.actual.var)
c(FT.actual.mean, FT.actual.sd)
```

```
## [1] 695.8750 150.2393
```

```
FT %>%
    as.data.frame() %>%
    ggplot(aes(x = FT)) + geom_histogram(bins = 30, color = "yellow", fill =
    "purple") +
    ggtitle("Simulated Season FT Attempt Totals by Shaq with Lakers")
```



This simulation is biased low on the variance but is better than the earlier simulation.

One reason that you may want to do complicated simulations like the example above is to make predictions for a player's future seasons.

0	1	2	3	4	5	6	7	8
5	2	8	0	7	2	4	2	2
0.156	0.062	0.250	0.000	0.219	0.062	0.125	0.062	0.062

3.2 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 3.1. 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 3.5. 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.]

```
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 = 0.818)
}

# 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
```

```
# Estimate probability of 0 FT made
sum(FT0.ind)/n.sims
```

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

3.3 A few reminders/tips for simulation, and a basic example

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)

nReps <- 10000

team_A_goals <- rpois(n = nReps, lambda = 4)
team_B_goals <- rpois(n = nReps, 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.

These tips will help you be more efficient when performing simulation tasks in `R`.

3.4 Streak Simulation - Basketball

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.

Q: 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?

A: 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.

```
set.seed(2022)

nReps <- 10000
longest_streak <- rep(NA, nReps)

for (i in 1:nReps) {
    game_results <- rbinom(size = 1, n = 82, prob = 0.25)  # 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
```

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

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

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

Chapter 4

Statistical Inference

4.1 One Sample and Two Sample t-tests and confidence intervals

Chapter 5

Correlation

Chapter 6

Linear Regression

Chapter 7

Data Scraping

```
library(dplyr)
library(rvest)
```

```
##
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(tidyverse)
library(kableExtra)
```

7.1 wnba scraping

```
wilson <-
"https://www.basketball-reference.com/wnba/players/w/wilsoa01w/gamelog/2022/"
wil_doc <- rvest::read_html(wilson)

wil_doc %>%
    rvest::html_elements(., xpath = "//*[(@id = 'div_wnba_pgl_basic')]") %>%
    rvest::html_table() -> wil
wil <- wil[[1]]
head(wil)
```

```
## # A tibble: 6 x 28
##   Rk    Date   Age   Tm    ``    Opp   ``    GS    MP    FG    FGA   `FG%` `3P`
##   <chr> <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 1     2022-~ 25-2~ LVA   "@"   PHO   W (+~ 1     28:35 5     8     .625  0
## 2 2     2022-~ 25-2~ LVA   ""    SEA   W (+~ 1     35:06 8     14    .571  1
## 3 3     2022-~ 25-2~ LVA   "@"   WAS   L (-~ 1     29:56 4     11    .364  0
## 4 4     2022-~ 25-2~ LVA   "@"   ATL   W (+~ 1     29:08 6     11    .545  0
```

```
## 5 5     2022-~ 25-2~ LVA   ""    PHO   W (+~ 1     33:45 4     8     .500  0
## 6 6     2022-~ 25-2~ LVA   ""    MIN   W (+~ 1     31:16 5     9     .556  1
## # ... with 15 more variables: `3PA` <chr>, `3P%` <chr>, FT <chr>, FTA <chr>,
## #   `FT%` <chr>, ORB <chr>, DRB <chr>, TRB <chr>, AST <chr>, STL <chr>,
## #   BLK <chr>, TOV <chr>, PF <chr>, PTS <chr>, GmSc <chr>
```

```
# wil2 <- mutate_all(wil, function(x) as.numeric(as.character(x)))
# mean(wil2['PTS'])

# wil$eFG<- (wil['FG'] + (0.5*wil['3P']))/wil['FGA'] wil$eFG
# ![Screenshot]('~/Google Drive/My Drive/Sports
# Analytics/SportsAnalyticsBook/images/scraping1')
```

Chapter 8

Principal Component Analysis

Chapter 9

Clustering

Chapter 10

Classification

Chapter 11

Decision Trees

11.1 Random Forests

11.2 Gradient Boosting

Chapter 12

Non-parametric Statistics

Chapter 13

Baseball

Chapter 14

Football

Chapter 15

Basketball

Chapter 16

Soccer

Chapter 17

Hockey

Chapter 18

Volleyball

18.1 Resources

Women's Volleyball D1 Statistics

Chapter 19

Other Sports

Chapter 20

Text solutions

20.1 Chapter 1

Example 1.1:
Population: all season passing totals of Manning's career
Sample: season passing totals of Manning's career with Broncos

Example 1.2:
Discrete: Passing yards, Passing TDs
Continuous: Passing attempt release times, Average yards per pass by game

Example 1.3:
Nominal: Pass result (completion, incomplete, interception)
Ordinal: Season injury status (no injuries, some injuries, missed full year)

Example 1.4:
$(4659 + 5477 + 4727 + 2249)/4 = 4278$

Example 1.5:
Colts ordered data: 3739, 3747, 4002, 4040, 4131, 4135, **4200**, 4267, 4397, 4413, 4500, 4557, 4700
Broncos ordered data: 2249, 4659, 4727, 5477 $\rightarrow (4659 + 4727)/2 = 4693$

Example 1.6:
Wins: 12, Games: 16, p=12/16=0.75

Chapter 21

Aaron's stuff

21.1 Notes for Chapter 2 (Probability)

Axioms of Probability:

1. $P(A) \geq 0$
2. $P(\Omega) = 1$
3. If $A_1, A_2, \ldots, A_n$ are disjoint events, then $P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$

Theorem 21.1 (Bayes theorem)**.** *Let A and B be events in Ω such that $P(B) > 0$. Then we have the following:*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

21.2 Suggested Readings

21.2.1 Moneyball

Moneyball, Chapter 2, How to Find a Ballplayer [Lewis, 2004]

Near the end of the chapter (page 40), Michael Lewis give a list of players the Oakland Athletics hoped to draft. How did these players turn out? Find the WAR for each of the players in their pre-free agency years and compare it against the Rockies draft picks in the same rounds from the same draft.

21.2.2 Future Value

Future Value, Chapter 7, How to Scout [Longenhagen and McDaniel, 2020]

If a player receives a running grade of 40, approximately what proportion of MLB players have a lower have a lower running grade?

For a given tool, about 95% of all player grades fall between what two bounds? (Consider the middle 95% of the distribution of grades.)

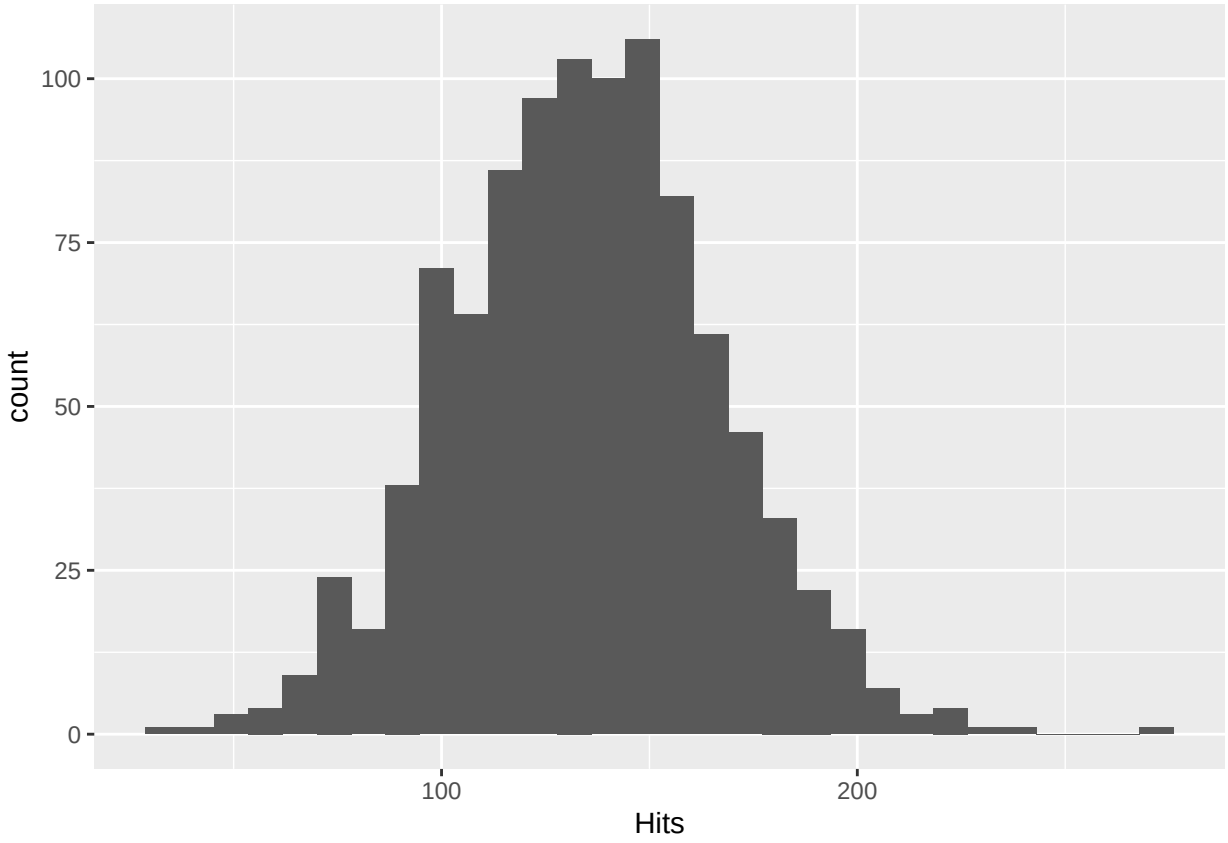
21.3 Notes for Chapter 4 (Simulation)

21.3.1 Baseball Simulation Example

```
library(tidyverse)
```

This is a baseball example for chapter 4.

```
set.seed(2022)
n.sims <- 1000
hits <- rep(0, n.sims)
avg <- 0.3
atbats.mean <- 450
atbats.sd <- 100
sim.atbats <- round(rnorm(n.sims, atbats.mean, atbats.sd))

for (i in 1:n.sims) {
    sim.hits <- rbinom(1, sim.atbats[i], avg)
    hits[i] = sim.hits
}
hits.df <- data.frame(Hits = hits)
hits.df %>%
    ggplot(aes(x = Hits)) + geom_histogram()
```



Bibliography

Michael Lewis. *Moneyball: The art of winning an unfair game*. WW Norton & Company, 2004.

Eric Longenhagen and Kiley McDaniel. *Future Value: The battle for baseball's soul and how teams will find the next superstar*. Triumph Books, 2020.