Fantasy Football Analytics

Isaac T. Petersen

To our daughter, Maisie.

Table of contents

Preface

This is a book in progress—it is incomplete. I will continue to add to and update it as I am able.

Open Access

This is an open-access book. This means that it is freely available for anyone to access.

License

Figure 1 Creative Commons License

The online version of this book is licensed under the Creative Commons Attribution License[1]. In short, you can use my work as long as you cite it.

[1] https://creativecommons.org/licenses/by/4.0/

Citation

The APA-style citation for the book is:

Petersen, I. T. (2024). *Fantasy football analytics*. Version 0.0.1. University of Iowa Libraries. https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook. [INSERT DOI LINK]

The BibTeX citation for the book is:

```
@book{petersenFantasyFootballAnalytics,
  title = {Fantasy football analytics},
  author = {Petersen, Isaac T.},
  year = {2024},
  publisher = {{University of Iowa Libraries}},
  note = {Version 0.0.1},
  doi = {INSERT},
  isbn = {INSERT},
 url = {https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook}
}
```

Accessibility

I strive to follow principles of accessibility[2] (archived at https://perma.cc/8XJ9-Q6QJ) to make the book content accessible to people with visual impairments and physical disabilities. If there are additional ways I can make the content more accessible, please let me know.

How to Contribute

This is an open-access textbook. My goal is to share data analysis strategies for free! Anyone is welcome to contribute to the project. If you would like to contribute, feel free to open an issue[3] or create a pull request[4] on

[2] https://bookdown.org/yihui/rmarkdown-cookbook/html-accessibility.html
[3] https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook/issues
[4] https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook/pulls

GitHub. The GitHub repository for the book is located here: https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook. If you have data or analysis examples that are you willing to share and include in the book, feel free to contact me.

Acknowledgments

I thank Dr. Benjamin Motz, who provided consultation and many helpful resources based on his fantasy football statistics class. I also thank key members of FantasyFootballAnalytics.net[5], including Val Pinskiy, Andrew Tungate, Dennis Andersen, and Adam Peterson, who helped develop and provide fantasy football-related resources and who helped sharpen my thinking about the topic. I also thank Professor Patrick Carroll, who taught me the value of statistics for answering important questions.

[5] http://fantasyfootballanalytics.net

1

Introduction

1.1 About this Book

How can we use information to make predictions about uncertain events? This book is about empiricism (basing theories on observed data) and judgment, prediction, and decision making in the context of uncertainty. The book provides an introduction to modern analytical techniques used to make informed predictions, test theories, and draw conclusions from a given dataset.

This book was originally written for a undergraduate-level course entitled, "Fantasy Football: Predictive Analytics and Empiricism". The chapters provide an overview of topics that each could have its own class and textbook, such as causal inference[1], factor analysis[2], cluster analysis[3], principal component analysis[4], machine learning[5], cognitive biases[6], modern portfolio theory[7], data visualization[8], simulation[9], etc. The book gives readers an overview of the breadth of the approaches to prediction and empiricism. As a consequence, the book does not cover any one technique or approach in great depth.

1.2 What is Fantasy Football?

Fantasy football is an online game where participants assemble (i.e., "draft") imaginary teams composed of real-life National Football League (NFL) players. In this game, participants compete against their opponents (e.g.,

[1] causal-inference.qmd
[2] factor-analysis.qmd
[3] cluster-analysis.qmd
[4] pca.qmd
[5] machine-learning.qmd
[6] cognitive-bias.qmd
[7] modern-portfolio-theory.qmd
[8] data-visualization.qmd
[9] simulation.qmd

friends/coworkers/classmates), accumulating points based on players' actual statistical performances in games. The goal is to outscore one's opponent each week to win matches and ultimately claim victory in the league.

1.3 Why Focus on Fantasy Football?

I was fortunate to have an excellent instructor who taught me the value of learning statistics to answer interesting and important questions. That is, I do not find statistics intrinsically interesting; rather, I find them interesting because of what they allow me to do. Many students find statistics intimidating in part because of how it is typically taught—with examples like dice rolls and coin flips that are (seemingly irrelevant and) boring to students. My contention is that applied examples are a more effective lens to teach many concepts in psychology and data analysis. It can be more engaging and relatable to learn statistics in the applied context of sports, a domain that is more intuitive to many. Many people play fantasy sports. This book involves applying statistics to a particular domain (football). People actually want to learn statistical principles and methods when they can apply them to interesting questions (e.g., sports). In my opinion [and supported by evidence; Motz (2013)], this is a much more effective way of engaging people and teaching statistics than in the context of abstract coin flips and dice rolls. Fantasy football relies heavily on prediction—trying to predict which players will perform best and selecting them accordingly. In this way, fantasy football provides a plethora of decision making opportunities in the face of uncertainty, and a wealth of data for analyzing these decisions. However, unlike many other applied domains in psychology, fantasy football (1) allows a person to see the accuracy of their predictions on a timely basis and (2) provides a safe environment for friendly competition. Thus, it provides a unique domain to evaluate—and improve—the accuracy of various prediction models.

1.4 Educational Value

Skills in statistics, statistical programming, and data analysis are highly valuable. This book includes practical and conceptual tools that build a foundation for critical thinking. The book aims to help readers evaluate theory in the light of evidence (and vice versa) and to refine decision making in the context of uncertainty. Readers will learn about the ways that psychological science (and

related disciplines) poses questions, formulates hypotheses, designs studies to test those questions, and interprets the findings, collectively with the aim to answer questions, improve decision making, and solve problems.

Of course, this is not a traditional psychology textbook. However, the book incorporates important psychological concepts, such as cognitive biases in judgment and prediction, etc. In the modern world of big data, research and society need people who know how to make sense of the information around us. Psychology is in a prime position to teach applied statistics to a wide variety of students, most of whom will not have careers as psychologists. Psychology can teach the importance of statistics given humans' cognitive biases. It can also teach about how these biases can influence how people interpret statistics. This book will teach readers the applications of statistics (prediction) and research methods (empiricism) to answer questions they find interesting, while applying scientific and psychological rigor.

1.5 Learning Objectives

This book aims to help readers accomplish the following learning objectives:

- Apply empirical inference and appreciate the value it provides over speculative supposition.
- Ask educated questions when confronted with decisions in the face of uncertainty.
- Understand human decision making, including common heuristics and cognitive biases and how to mitigate them analytically.
- Engage in critical thinking about causality, including devising plausible alternative explanations for observed effects.
- Understand causal inference including confounding, causal pathways, and counterfactuals.
- Think empirically about human behavior and performance.
- Describe the strengths and weaknesses of humans versus computers in prediction scenarios.
- Apply basic skills in statistical programming using `R` to manipulate and summarize datasets and to conduct data analysis.
- Critically evaluate the strengths and limitations of different statistical models and methodologies used in predicting uncertain events, enhancing their understanding of statistical inference and model selection.
- Use various analytical techniques for predicting the outcome of uncertain events, and for uncovering latent causes of patterns in observed data.
- Interpret findings from various statistical approaches and evaluate the accuracy of predictions.

- Engage in iterative problem-solving processes, refining analytical approaches based on feedback and outcomes, and adapting strategies accordingly.
- Communicate statistical findings and analyses in both written and oral formats, demonstrating proficiency in presenting complex information to diverse audiences.
- Make sense of big data.
- Use practical analytical skills that can be applied in future research and job settings.

1.6 Disclosures

I am the Owner of Fantasy Football Analytics, LLC, which operates https://fantasyfootballanalytics.net.

1.7 Disclaimer

"*This material probably won't win you fantasy football championships. You could take what we learn and apply it to fantasy football and you might become 5 percent more likely to win. Or… Consider the broader relevance of this. You could learn data analysis and figure out ways to apply it to other systems. And you could be making a six-figure salary within the next five years.*" – Benjamin Motz, Ph.D.

2

Intro to Football and Fantasy

This chapter provides a brief primer on (American) football and fantasy football. If you are already familiar with fantasy football, feel free to skip this chapter.

2.1 Football

Football is the most widely watched sport in the United States.[1]

2.1.1 The Objective

The goal in football is for a team to score more points than their opponent. A game lasts 60 minutes, and it is separated into four 15-minute quarters. The team with the most points when the time runs out wins.

2.1.2 The Roster

2.1.2.1 Overview

Each team has 11 players on the field at a time. The particular players who are on the field will depend on the situation, but usually includes one of the three subsets of players:

1. Offense
2. Defense
3. Special Teams

[1] https://news.gallup.com/poll/610046/football-retains-dominant-position-favorite-sport.aspx (archived at https://perma.cc/X2UG-RAAK); https://www.statista.com/statistics/1430289/most-watched-sports-leagues-usa/ (archived at https://perma.cc/JNU6-S96A)

An example formation is depicted in Figure Figure 2.1.

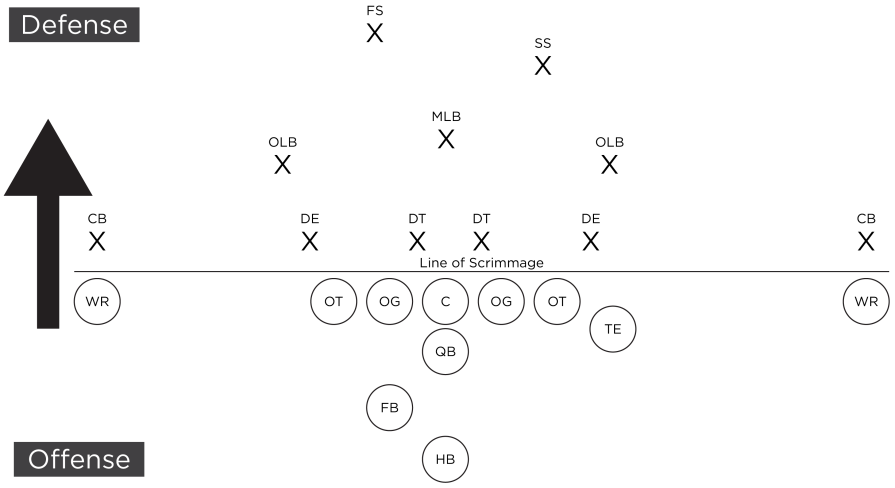

Figure 2.1 An Example Football Formation for the Offense and Defense. The solid line indicates the line of scrimmage. The arrow indicates the direction the offense tries to advance the ball.

2.1.2.2 Offense

The offense is on the field when the team has the ball.

Players on offense include:

- Quarterback (QB)
- Running Back (RB)
  - Halfback (HB) or Tailback (TB)
  - Fullback (FB)
- Wide Receiver (WR)
- Tight End (TE)
- Offensive Linemen (OL), part of the "Offensive Line"
  - Center (C)
  - Offensive Guard (OG)
  - Offensive Tackle (OT)

The quarterback is the most important player on the offense. They help lead the team down the field. The quarterback receives the ball from the Center at the beginning of the play, and they can either hand the ball off (typically to a Running Back or Fullback), pass the ball (typically to a Wide Receiver or

Tight End), or run the ball. Quarterbacks tend to have a strong arm for throwing the ball far and accurately. Some quarterbacks are fast and are considered "dual threats" to pass or run.

Running Backs take a hand-off from the Quarterback to execute a running play (i.e., a rush). They may also catch short passes from the Quarterback or help protect (i.e., block for) the Quarterback from the defensive players who are trying to tackle the Quarterback. Halfbacks and Tailbacks tend to be quick and agile. Fullbacks tend to be strong and powerful.

Wide Receivers catch passes from the Quarterback to execute a passing play. On running plays, they provide protection for the player running the ball (e.g., the Running Back) so the ball carrier can get as far as possible without being tackled. Wide receivers tend to be tall, fast, have good hands (can catch the ball well), and can jump high.

Tight Ends block for running and passing plays, and they catch passes from the Quarterback. Tight ends tend to be strong and have good hands.

Offensive Linemen block for running and passing plays. On passing plays, they provide protection for the Quarterback so the Quarterback has time to pass the ball without being tackled. On running plays, they provide protection for the player running the ball (e.g., the Running Back) so the ball carrier can get as far as possible without being tackled. Offensive Linemen tend to be large so they can provide adequate protection for the Quarterback and Running Back.

2.1.2.3 Defense

The defense is on the field when the team does not have the ball (i.e., when the opposing team has the ball).

Players on defense include:

- Defensive Linemen (DL), part of the "Defensive Line"
  - Defensive End (DE)
  - Defensive Tackle (DT)
- Linebacker (LB)
  - Middle (or Inside) Linebacker (MLB)
  - Outside Linebacker (OLB)
- Defensive Back (DB), part of the "Secondary"
  - Cornerback (CB)
  - Safety (S)
    - Free Safety (FS)
    - Strong Safety (SS)

The players on the defense attempt to tackle the offensive players for as short of gains as possible and attempt to prevent completed passes.

On passing plays, Defensive Linemen try to apply pressure to the Quarterback and try to tackle the Quarterback behind the line of scrimmage before the Quarterback can throw the ball (i.e., a sack). On rushing plays, Defensive Linemen try to tackle the ball carrier to prevent the ball carrier from advancing the ball (i.e., gaining yards). Defensive Linemen tend to be large yet quick so they can apply pressure to the Quarterback.

Linebackers are versatile in that, on a given play, they may attempt to a) "blitz" to sack the Quarterback, b) stop the Running Back, or c) prevent a completed pass. Linebackers tend to be strong yet agile.

Defensive Backs are specialist pass defenders. The main role of Cornerbacks is to cover the Wide Receivers. Safeties serve as the last line of defense for longer passes. Defensive Backs tend to be quick and agile.

2.1.2.4 Special Teams

The special teams involves specialist players who are on the field during all kicking plays including kickoffs, field goals, and punts.

Players on special teams include:

- Kicker (K)
- Punter (P)
- Holder
- Long Snapper
- Punt Returner
- Kick Returner
- and other players intended to block for or to tackle the ball carrier

On a field goal attempt, the Long Snapper snaps the ball to the Holder, who holds the ball for the Kicker. The Kicker attempts field goals and, during kickoffs, kicks the ball to the opposing team. During kickoffs, the Kick Returner catches the kicked ball and returns it for as many yards as possible. During a punt play, the Long Snapper snaps the ball to the Punter who kicks (i.e., punts) the ball to the opposing team. The Punt Returner catches the punted ball and returns it for as many yards as possible.

2.1.3 The Field

The football field is rectangular and is 120 yards long and 53 1/3 yards wide (109.73 m x 48.77 m).[2] At each end of the 120-yard field is a team's end zone.

[2]One yard is equal to three feet. A yard is just smaller than a meter (0.9144 meters).

Each end zone is 10 yards long (9.14 m). Thus, the distance from one end zone to the other end zone is 100 yards (91.44 m). Behind each end zone is a field goal post. A diagram of a football field is depicted in Figure Figure 2.2.

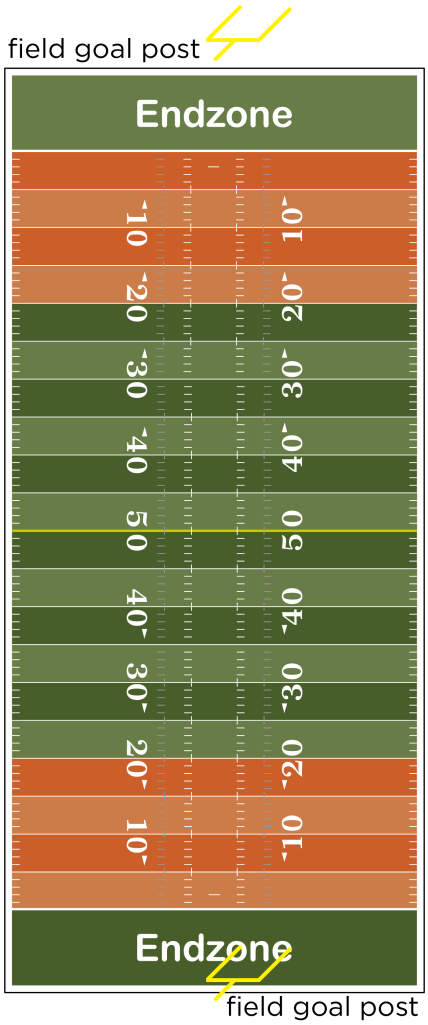

Figure 2.2 A Diagram of a Football Field. The yard markers depict the distance from the nearest end zone. The orange shaded area is called the "red zone", where chances of scoring points are highest. The original figure was modified to depict field goal posts. (Figure retrieved from https://commons.wikimedia.org/wiki/File:American_football_field.svg)

2.1.4 The Gameplay

At the beginning of the game, there is a coin flip to determine which teams receives the ball first and which team takes which side of the field. During the kickoff, the kicking team kicks the ball to the receiving team, who has the option to return the kick. The offense starts their possession at the 25 yard line—if there is no return (i.e., a touchback)—or wherever the kick returner is tackled or goes out of bounds.

The team with the ball (i.e., the offense) has four opportunities ("downs") to advance the ball (i.e., gain) 10 yards. A team can advance the ball either by running it or by throwing (i.e., passing) and catching it. At the end of a rushing play, the ball advances to wherever the ball carrier is tackled or goes out of bounds (i.e., wherever the player is "down"). At the end of a passing play, if the thrown ball is caught (i.e., a completed pass), the ball advances to wherever the ball carrier is tackled or goes out of bounds. If the thrown ball is not caught in bounds before the ball hits the ground (i.e., an incomplete pass), the ball does not advance. Wherever the ball is advanced to dictates where the next play begins. The yard position on the field where the next play takes place from is known as the "line of scrimmage". Neither team can cross the line the line of scrimmage until the next play begins. To begin the play, the ball is placed on the line of scrimmage and the Center gives (or "snaps") the ball to the Quarterback.

If the team advances the ball 10 or more yards within four downs, the team receives a "first down" and is awarded a new set of downs—four more downs to advance the ball 10 more yards. If the team advances the ball all the way to the other team's end zone, they score a touchdown. If the team fails to advance the ball 10 or more yards within four downs, the team loses the ball, and the other team takes possession at that spot on the field. There are risks of giving the other team the ball with a short distance to score. Thus, on fourth down, instead of trying to advance the ball for a first down, a team may choose to kick a field goal—to get points—or to punt.

A field goal involves a kicker kicking the ball with an intent to kick the ball through the field goal posts ("uprights"). To score points by making a field goal, the kicked ball must go between the uprights (extended vertically) and over the cross bar.

Punting involves a punter kicking the ball to the other team with an intent to give their opponent worse field position, thus making it harder for the other team to score. The punting team tries to pin the opponent as close as possible to the opponent's end zone (i.e., as far as possible from the own team's end zone), so they have a longer distance to go to score a touchdown.

There are multiple ways that ball possession can switch from the offense to the other team. After scoring a touchdown, field goal, or safety, there is a kickoff, in which the scoring team kicks the ball to the opponent. Another

way that the ball switches possession to the other team is if the team commits a turnover. The defense can force a turnover by an interception, fumble recovery, or turnover on downs. A turnover due to an interception occurs when a defensive player catches the quarterback's pass. A turnover due to a fumble recovery occurs when an offensive player, who had possession of the ball, loses the ball before being down or scoring a touchdown and the ball is recovered by the opponent. A turnover on downs occurs when the team attempts on fourth down to achieve the remainder of the needed 10 yards to go but fails.

Other football-related situations include tackles for loss and sacks. A tackle for loss occurs when a ball carrier is tackled behind the line of scrimmage. A sack occurs when a Quarterback is tackled with the ball behind the line of scrimmage. A pass defended occurs when a defensive player knocks down the ball in the air so that the indended receiver cannot catch the ball.

2.1.5 The Scoring

The goal of the team with the ball (i.e., the offense) is to score points. It can do this by either advancing the ball into the other team's end zone (6 points) or by kicking a field goal (3 points). Advancing the ball in the other team's end zone is called a touchdown. After a touchdown, the offense chooses to attempt either a point-after-touchdown (PAT) or a two-point conversion. A PAT is a short kick attempt from the 15-yard line (i.e., 15 yards away from the end zone) that, if it goes through the goal posts ("uprights") and over the cross bar, is worth 1 point. A two-point conversion is a single-scoring opportunity from the 3-yard line (i.e., 3 yards away from the end zone). If the offense scores (i.e., advances the ball into the end zone) from the 3-yard line, the team is awarded 2 points.

A team can kick a field goal from any distance as long as the kick goes through the goal posts. The current record for the longest field goal is 66 yards (by Justin Tucker in 2021).

A safety occurs when the offense is tackled with the ball in their own end zone. When a safety occurs, the opposing team (i.e., defense) is awarded two points and the ball.

2.1.6 Glossary of Terms

- running play ("run") or rushing play (or "rush")—the attempt by an offensive player, typically the Running Back or Quarterback, to advance the ball "on the ground" by running it—not by passing it forward
- passing play (or "pass")—the attempt by an offensive player, typically the Quarterback, to advance the ball by throwing it forward to an offensive player

- passing attempt—the attempt to advance the ball by passing it (i.e., a thrown pass)
- rushing attempt—the attempt to advance the ball by running it
- passing completion—a thrown pass that is succesfully caught by an offensive player
- passing incompletion—a thrown pass that is not caught by an offensive player
- passing yards—the distance (in yards) the player advanced the ball by throwing it
- rushing yards—the distance (in yards) the player advanced the ball by running it
- receving yards—the distance (in yards) the player advanced the ball by catching thrown passes and then running with it further upfield
- kick/punt return yards—the distance (in yards) the player advanced the ball by returning kicks or punts
- turnover return yards—the distance (in yards) the player advanced the ball by returning turnovers
- reception—a pass that is caught by the offensive player
- touchdown—advancing the ball into the opponent's end zone either by a) throwing a completed pass that ends up in the end zone, b) running it into the end zone, c) catching it in the end zone, or d) catching it and then running it into the end zone
- passing touchdown—advancing the ball into the opponent's end zone either by throwing a completed pass that ends up in the end zone
- rushing touchdown—advancing the ball into the opponent's end zone either by running it into the end zone
- receiving touchdown—advancing the ball into the opponent's end zone either by catching it in the end zone or by catching it and then running it into the end zone
- kick/punt return touchdown—advancing the ball into the opponent's end zone when returning a kick or punt
- turnover return touchdown—advancing the ball into the opponent's end zone when returning a turnover (i.e., interception or fumble)
- two-point conversion—a single-scoring opportunity from the 3-yard line (i.e., 3 yards away from the end zone) that is an option given to a team that scores a touchdown; if the offense scores (i.e., advances the ball into the end zone) from the 3-yard line, the team is awarded 2 points
- block—when the defense/special teams blocks a kick or field goal by hitting the ball just after it is kicked to prevent the ball from going far
- kickoff—the kicking team kicks the ball to the receiving team, who has the option to return the kick
- field goal—a kicker kicks the ball with an intent to kick the ball through the field goal posts ("uprights"). To score points by making a field goal, the kicked ball must go between the uprights (extended vertically) and over the cross bar. If the field goal attempt is successful, the team gains 3 points.

- point after touchdown (PAT)—a short kick attempt from the 15-yard line (i.e., 15 yards away from the end zone) that, if it goes through the goal posts ("uprights") and over the cross bar, is worth 1 point
- extra point returned—if the defense/special teams returns the ball into the opponent's end zone during a point after touchdown (PAT) attempt, it is worth 2 points
- punt—a punter kicks the ball to the other team with an intent to give their opponent worse field position, thus making it harder for the other team to score
- fumble lost—when an offensive player, who had possession of the ball, loses the ball before being down or scoring a touchdown and the ball is recovered by the opponent
- fumble forced—when a defensive player knocks the ball out of the hands of an offensive player, who had possession of the ball
- fumble recovery—when a defensive player recovers a fumble by the opponent
- interception—when a defensive player catches a pass from an offensive player
- tackle—when a player brings down the ball carrier
- tackle solo—when a player is the main tackler (i.e., the primary player to bring down the ball carrier)
- tackle assist—when a player is one of two or more players who, together, bring down the ball carrier
- tackle for loss—when an offensive player is tackled with the ball behind the line of scrimmage
- sack—when a Quarterback is tackled with the ball behind the line of scrimmage
- pass defended—when a defensive player knocks down the ball in the air so that the indended receiver cannot catch the ball
- safety—when the offense is tackled with the ball in their own end zone

2.2 Fantasy Football

2.2.1 Overview of Fantasy Football

Fantasy football is one of the most widely played games in the history of games. It is estimated that around 62 million people play fantasy sports[3], of whom around 29 million play fantasy football.[4] As noted in the Introduction[5],

[3] https://thefsga.org/industry-demographics/ (archived at https://perma.cc/9PB8-ZDJJ)

[4] https://www.statista.com/topics/10895/fantasy-sports-in-the-us/ (archived at https://perma.cc/8YSN-UUNT)

[5] intro.qmd

fantasy football is an online game where participants assemble (i.e., "draft") imaginary teams composed of real-life National Football League (NFL) players.[6] The participants are in charge of managing and making strategic decisions for their imaginary team to have the best possible team that will score the most points. Thus, the participants are called "managers". Managers make decisions such as selecting which players to draft, selecting which players to play (i.e., "start") on a weekly basis, identifying players to pick up from the remaining pool of available players (i.e., waiver wire), and making trades with other teams. Fantasy football relies heavily on prediction—trying to predict which players will perform best and selecting them accordingly.

2.2.2 The Fantasy League

A fantasy football "league" is composed of various imaginary (i.e., "fantasy") teams—and their associated manager. In the fantasy league, the managers' fantasy teams play against each other. A fantasy league is commonly composed of 8, 10, or 12 fantasy teams, but leagues can have more or fewer teams.

2.2.3 The Roster of a Fantasy Team

On a given roster, a manager has a "starting lineup" and a "bench". Each week, the manager decides which players on their roster to put in the starting lineup, and which to keep on the bench. In many leagues, a starting lineup is composed of offensive players, a kicker, and defense/special teams:

Offensive players:

Table 2.1 Offensive Players in the Starting Lineup

Position	Typical Number of Players in Starting Lineup
Quarterback (QB)	1
Running Back (RB)	2
Wide Receiver (WR)	2
Tight End (TE)	1
Flex Position	1

A "flex position" is a flexible position that can involve a player from various positions: e.g., a Running Back, Wide Receiver, or Tight End.

Kickers:

[6] Fantasy leagues are also available for baseball[7], basketball[8], and many other sports.

- one Kicker (K)

Defense/Special Teams:

- one Team Defense (DST/D/DEF) or multiple Individual Defensive Players (IDP)

2.2.4 Scoring

2.2.4.1 Scoring Overview

In the game of fantasy football, managers accumulate points on a weekly basis based on players' actual statistical performances in NFL games. Managers receive points for only those players who are on their starting lineup (not players on their bench). A manager's goal is to outscore their opponent each week to win matches and ultimately claim victory in the league. Scoring settings can differ from league to league.

Below are common scoring settings for fantasy leagues.

2.2.4.2 Offensive Players

Table 2.2 Common Scoring Settings for Offensive Players

Statistical category	Points
Rushing or receiving TD	6
Returning a kick or punt for a TD	6
Returning or recovering a fumble for a TD	6
Passing TD	4
Passing INT	−2
Fumble lost	−2
Rushing, passing, or receiving 2-point conversion	2
Rushing or receiving yards	1 point per 10 yards
Passing yards	1 point per 25 yards

Note: "TD" = touchdown; "INT" = interception

Other common (but not necessarily standard) statistical categories include:

- receptions (called "point per reception" [PPR] leagues)

- return yards
- passing attempts
- rushing attempts

2.2.4.3 Kickers

Table 2.3 Common Scoring Settings for Kickers

Statistical category	Points
FG made: 50+ yards	5
FG made: 40–49 yards	4
FG made: 39 yards or less	3
Rushing, passing, or receiving 2-point conversion	2
Point after touchdown attempt made	1
Point after touchdown attempt missed	−1
Missed FG: 0–39 yards	−2
Missed FG: 40–49 yards	−1

Note: "FG" = field goal

2.2.4.4 Team Defense/Special Teams

Table 2.4 Common Scoring Settings for Team Defense/Special Teams

Statistical category	Points
Defensive or special teams TD	3
Interception	2
Fumble recovery	2
Blocked punt, PAT, or FG	2
Safety	2
Sack	1

Note: "TD" = touchdown; "PAT" = point after touchdown; "FG" = field goal

2.2.4.5 Individual Defensive Players

Table 2.5 Common Scoring Settings for Individual Defensive Players

Statistical category	Points
Tackle solo	1
Tackle assist	0.5
Tackle for loss	1
Sack	2
Interception	4
Fumble forced	2
Fumble recovery	2
TD	6
Safety	2
Pass defended	1
Blocked kick	2
Extra point returned	2

Note: "TD" = touchdown

Other common (but not necessarily standard) statistical categories include:

- turnover return yards

2.2.4.6 Common Scoring Abbreviations

- "TD" = touchdown
- "INT" = interception
- "yds" = yards
- "ATT" = attempts
- "2-pt conversion" = two-point conversion
- "FG" = field goal
- "PAT" = point after touchdown (i.e., extra point/point after attempt)

3

Getting Started with R for Data Analysis

The book uses R for statistical analyses (http://www.r-project.org). R is a free software environment; you can download it at no charge here: https://cran.r-project.org.

3.1 Initial Setup

To get started, follow the following steps:

1. Install R: https://cran.r-project.org
2. Install RStudio Desktop: https://posit.co/download/rstudio-desktop
3. After installing RStudio, open RStudio and run the following code in the console to install several key R packages:

```
install.packages(c("petersenlab","tidyverse","psych"))
```

> **Note 1: If you are in Dr. Petersen's class**
>
> If you are in Dr. Petersen's class, also perform the following steps:
>
> 1. Set up a free account on GitHub.com[a].
> 2. Download GitHub Desktop: https://desktop.github.com
>
> [a] https://github.com

3.2 Installing Packages

You can install R packages using the following syntax:

```
install.packages("INSERT_PACKAGE_NAME_HERE")
```

For instance, you can use the following code to install the `nflreadr` package:

```
install.packages("nflreadr")
```

3.3 Load Packages

```
library("ffanalytics")
```

```
Note: the ffanalytics package locally caches ADP & ECR data scrapes. Cached scrapes
older than 8 hours are dropped (upon checking)
  - See ?clear_ffanalytics_cache() for how to manually clear the cache
  - Use list_ffanalytics_cache() to see what is currently cached
```

```
library("nflreadr")
library("nflfastR")
```

```
Attaching package: 'nflfastR'

The following objects are masked from 'package:nflreadr':

    load_pbp, load_player_stats
```

```
library("nfl4th")
library("nflplotR")
library("progressr")
library("lubridate")
```

```
Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr   1.1.4     v readr   2.1.5
v forcats 1.0.0     v stringr 1.5.1
v ggplot2 3.5.1     v tibble  3.2.1
v purrr   1.0.2     v tidyr   1.3.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

3.4 Download Football Data

3.4.1 Players

```
nfl_players <- progressr::with_progress(
  nflreadr::load_players())
```

3.4.2 Teams

```
nfl_teams <- progressr::with_progress(
  nflreadr::load_teams(current = TRUE))
```

3.4.3 Player Info

3.4.4 Rosters

A Data Dictionary for rosters is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_rosters.html

```
nfl_rosters <- progressr::with_progress(
  nflreadr::load_rosters(seasons = TRUE))

nfl_rosters_weekly <- progressr::with_progress(
  nflreadr::load_rosters_weekly(seasons = TRUE))
```

3.4.5 Game Schedules

A Data Dictionary for game schedules data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_schedules.html

```
nfl_schedules <- progressr::with_progress(
  nflreadr::load_schedules(seasons = TRUE))
```

3.4.6 The Combine

A Data Dictionary for data from the combine is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_combine.html

```
nfl_combine <- progressr::with_progress(
  nflreadr::load_combine(seasons = TRUE))
```

3.4.7 Draft Picks

A Data Dictionary for draft picks data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_draft_picks.html

```
nfl_draftPicks <- progressr::with_progress(
  nflreadr::load_draft_picks(seasons = TRUE))
```

3.4.8 Depth Charts

A Data Dictionary for data from weekly depth charts is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_depth_charts.html

```
nfl_depthCharts <- progressr::with_progress(
  nflreadr::load_depth_charts(seasons = TRUE))
```

3.4.9 Play-By-Play Data

To download play-by-play data from prior weeks and seasons, we can use the `load_pbp()` function of the `nflreadr` package. We add a progress bar using the `with_progress()` function from the `progressr` package because it takes a while to run. A Data Dictionary for the play-by-play data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_pbp.html

> **Note 2: Downloading play-by-play data**
>
> Note: the following code takes a while to run.

```
nfl_pbp <- progressr::with_progress(
  nflreadr::load_pbp(seasons = TRUE))
```

3.4.10 4th Down Data

```
nfl_4thdown <- nfl4th::load_4th_pbp(seasons = 2014:2023)
```

3.4.11 Participation

A Data Dictionary for the participation data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_participation.html

```
nfl_participation <- progressr::with_progress(
  nflreadr::load_participation(
    seasons = TRUE,
    include_pbp = TRUE))
```

3.4.12 Historical Weekly Actual Player Statistics

We can download historical week-by-week actual player statistics using the `load_player_stats()` function from the `nflreadr` package. A Data Dictionary for statistics for offensive players is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_player_stats.html. A Data Dictionary for statistics for defensive players is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_player_stats_def.html.

```
nfl_actualStats_offense_weekly <- progressr::with_progress(
  nflreadr::load_player_stats(
    seasons = TRUE,
    stat_type = "offense"))

nfl_actualStats_defense_weekly <- progressr::with_progress(
  nflreadr::load_player_stats(
    seasons = TRUE,
    stat_type = "defense"))

nfl_actualStats_kicking_weekly <- progressr::with_progress(
  nflreadr::load_player_stats(
    seasons = TRUE,
    stat_type = "kicking"))
```

3.4.13 Injuries

A Data Dictionary for injury data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_injuries.html

```
nfl_injuries <- progressr::with_progress(
  nflreadr::load_injuries(seasons = TRUE))
```

3.4.14 Snap Counts

A Data Dictionary for snap counts data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_snap_counts.html

```
nfl_snapCounts <- progressr::with_progress(
  nflreadr::load_snap_counts(seasons = TRUE))
```

3.4.15 ESPN QBR

A Data Dictionary for ESPN QBR data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_espn_qbr.html

```
nfl_espnQBR_seasonal <- progressr::with_progress(
  nflreadr::load_espn_qbr(
    seasons = TRUE,
    summary_type = c("season")))
```

```
nfl_espnQBR_weekly <- progressr::with_progress(
  nflreadr::load_espn_qbr(
    seasons = TRUE,
    summary_type = c("weekly")))

nfl_espnQBR_weekly$game_week <- as.character(nfl_espnQBR_weekly$game_week)

nfl_espnQBR <- bind_rows(
  nfl_espnQBR_seasonal,
  nfl_espnQBR_weekly
)
```

3.4.16 NFL Next Gen Stats

A Data Dictionary for NFL Next Gen Stats data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_nextgen_stats.html

```
nfl_nextGenStats_pass_weekly <- progressr::with_progress(
  nflreadr::load_nextgen_stats(
    seasons = TRUE,
    stat_type = c("passing")))

nfl_nextGenStats_rush_weekly <- progressr::with_progress(
  nflreadr::load_nextgen_stats(
    seasons = TRUE,
    stat_type = c("rushing")))

nfl_nextGenStats_rec_weekly <- progressr::with_progress(
  nflreadr::load_nextgen_stats(
    seasons = TRUE,
    stat_type = c("receiving")))

nfl_nextGenStats_weekly <- bind_rows(
  nfl_nextGenStats_pass_weekly,
  nfl_nextGenStats_rush_weekly,
  nfl_nextGenStats_rec_weekly
)
```

3.4.17 Advanced Stats from PFR

A Data Dictionary for PFR passing data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_pfr_passing.html

```
nfl_advancedStatsPFR_pass_seasonal <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("pass"),
    summary_level = c("season")))

nfl_advancedStatsPFR_pass_weekly <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("pass"),
    summary_level = c("week")))

nfl_advancedStatsPFR_rush_seasonal <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("rush"),
    summary_level = c("season")))

nfl_advancedStatsPFR_rush_weekly <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("rush"),
    summary_level = c("week")))

nfl_advancedStatsPFR_rec_seasonal <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("rec"),
    summary_level = c("season")))

nfl_advancedStatsPFR_rec_weekly <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("rec"),
    summary_level = c("week")))

nfl_advancedStatsPFR_def_seasonal <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
    seasons = TRUE,
    stat_type = c("def"),
    summary_level = c("season")))

nfl_advancedStatsPFR_def_weekly <- progressr::with_progress(
  nflreadr::load_pfr_advstats(
```

```
    seasons = TRUE,
    stat_type = c("def"),
    summary_level = c("week")))

nfl_advancedStatsPFR <- bind_rows(
  nfl_advancedStatsPFR_pass_seasonal,
  nfl_advancedStatsPFR_pass_weekly,
  nfl_advancedStatsPFR_rush_seasonal,
  nfl_advancedStatsPFR_rush_weekly,
  nfl_advancedStatsPFR_rec_seasonal,
  nfl_advancedStatsPFR_rec_weekly,
  nfl_advancedStatsPFR_def_seasonal,
  nfl_advancedStatsPFR_def_weekly,
)
```

3.4.18 Player Contracts

A Data Dictionary for player contracts data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_contracts.html

```
nfl_playerContracts <- progressr::with_progress(
  nflreadr::load_contracts())
```

3.4.19 FTN Charting Data

A Data Dictionary for FTN Charting data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_ftn_charting.html

```
nfl_ftnCharting <- progressr::with_progress(
  nflreadr::load_ftn_charting(seasons = TRUE))
```

3.4.20 Fantasy Player IDs

A Data Dictionary for fantasy player ID data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_ff_playerids.html

```
nfl_playerIDs <- progressr::with_progress(
  nflreadr::load_ff_playerids())
```

3.4.21 FantasyPros Rankings

A Data Dictionary for FantasyPros ranking data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_ff_rankings.html

```
#nfl_rankings <- progressr::with_progress( # currently throws error
#  nflreadr::load_ff_rankings(type = "all"))

nfl_rankings_draft <- progressr::with_progress(
  nflreadr::load_ff_rankings(type = "draft"))

nfl_rankings_weekly <- progressr::with_progress(
  nflreadr::load_ff_rankings(type = "week"))

nfl_rankings <- bind_rows(
  nfl_rankings_draft,
  nfl_rankings_weekly
)
```

3.4.22 Expected Fantasy Points

A Data Dictionary for expected fantasy points data is located at the following link: https://nflreadr.nflverse.com/articles/dictionary_ff_opportunity.html

```
nfl_expectedFantasyPoints_weekly <- progressr::with_progress(
  nflreadr::load_ff_opportunity(
    seasons = TRUE,
    stat_type = "weekly",
    model_version = "latest"
  ))

nfl_expectedFantasyPoints_pass <- progressr::with_progress(
  nflreadr::load_ff_opportunity(
    seasons = TRUE,
    stat_type = "pbp_pass",
    model_version = "latest"
  ))

nfl_expectedFantasyPoints_rush <- progressr::with_progress(
  nflreadr::load_ff_opportunity(
    seasons = TRUE,
    stat_type = "pbp_rush",
    model_version = "latest"
  ))
```

```
nfl_expectedFantasyPoints_weekly$season <- as.integer(nfl_expectedFantasyPoints_weekly$season)

nfl_expectedFantasyPoints_offense <- bind_rows(
  nfl_expectedFantasyPoints_pass,
  nfl_expectedFantasyPoints_rush
)
```

3.5 Data Dictionary

Data Dictionaries are metadata that describe the meaning of the variables in a datset. You can find Data Dictionaries for the various NFL datasets at the following link: https://nflreadr.nflverse.com/articles/index.html.

3.6 Create a Data Frame

Here is an example of creating a data frame:

```
players <- data.frame(
  ID = 1:12,
  name = c(
    "Ken Cussion",
    "Ben Sacked",
    "Chuck Downfield",
    "Ron Ingback",
    "Rhonda Ball",
    "Hugo Long",
    "Lionel Scrimmage",
    "Drew Blood",
    "Chase Emdown",
    "Justin Time",
    "Spike D'Ball",
    "Isac Ulooz"),
  position = c("QB","QB","QB","RB","RB","WR","WR","WR","WR","TE","TE","LB"),
  age = c(40, 30, 24, 20, 18, 23, 27, 32, 26, 23, NA, 37)
  )
```

```
fantasyPoints <- data.frame(
  ID = c(2, 7, 13, 14),
  fantasyPoints = c(250, 170, 65, 15)
)
```

3.7 Variable Names

To see the names of variables in a data frame, use the following syntax:

```
names(nfl_players)
```

```
 [1] "status"                    "display_name"
 [3] "first_name"                "last_name"
 [5] "esb_id"                    "gsis_id"
 [7] "suffix"                    "birth_date"
 [9] "college_name"              "position_group"
[11] "position"                  "jersey_number"
[13] "height"                    "weight"
[15] "years_of_experience"       "team_abbr"
[17] "team_seq"                  "current_team_id"
[19] "football_name"             "entry_year"
[21] "rookie_year"               "draft_club"
[23] "college_conference"        "status_description_abbr"
[25] "status_short_description" "gsis_it_id"
[27] "short_name"                "smart_id"
[29] "headshot"                  "draft_number"
[31] "uniform_number"            "draft_round"
[33] "season"
```

```
names(players)
```

```
[1] "ID"       "name"     "position" "age"
```

```
names(fantasyPoints)
```

```
[1] "ID"           "fantasyPoints"
```

3.8 Logical Operators

3.8.1 Is Equal To: ==

```
players$position == "RB"
```

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

3.8.2 Is Not Equal To: !=

```
players$position != "RB"
```

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

3.8.3 Is Greater Than: >

```
players$age > 30
```

```
 [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE    NA  TRUE
```

3.8.4 Is Less Than: <

```
players$age < 30
```

```
 [1] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE    NA FALSE
```

3.8.5 Is Greater Than or Equal To: >=

```
players$age >= 30
```

```
 [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE    NA  TRUE
```

3.8.6 Is Less Than or Equal To: `<=`

```
players$age <= 30
```

```
 [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE    NA FALSE
```

3.8.7 Is In a Value of Another Vector: `%in%`

```
players$position %in% c("RB","WR")
```

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

3.8.8 Is Not In a Value of Another Vector: `!(%in%)`

```
!(players$position %in% c("RB","WR"))
```

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

3.8.9 Is Missing: `is.na()`

```
is.na(players$age)
```

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

3.8.10 Is Not Missing: `!is.na()`

```
!is.na(players$age)
```

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

3.8.11 And: `&`

```
players$position == "WR" & players$age > 26
```

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

3.8.12 Or: |

```
players$position == "WR" | players$age > 23
```

```
 [1]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE    NA  TRUE
```

3.9 Subset

To subset a data frame, use brackets to specify the subset of rows and columns to keep, where the value/vector before the comma specifies the rows to keep, and the value/vector after the comma specifies the columns to keep:

```
dataframe[rowsToKeep, columnsToKeep]
```

You can subset by using any of the following:

- numeric indices of the rows/columns to keep (or drop)
- names of the rows/columns to keep (or drop)
- values of `TRUE` and `FALSE` corresponding to which rows/columns to keep

3.9.1 One Variable

To subset one variable, use the following syntax:

```
players$name
```

```
 [1] "Ken Cussion"     "Ben Sacked"      "Chuck Downfield"  "Ron Ingback"
 [5] "Rhonda Ball"     "Hugo Long"       "Lionel Scrimmage" "Drew Blood"
 [9] "Chase Emdown"    "Justin Time"     "Spike D'Ball"     "Isac Ulooz"
```

or:

```
players[,"name"]
```

```
 [1] "Ken Cussion"     "Ben Sacked"      "Chuck Downfield"  "Ron Ingback"
 [5] "Rhonda Ball"     "Hugo Long"       "Lionel Scrimmage" "Drew Blood"
 [9] "Chase Emdown"    "Justin Time"     "Spike D'Ball"     "Isac Ulooz"
```

3.9.2 Particular Rows of One Variable

To subset one variable, use the following syntax:

```
players$name[which(players$position == "RB")]
```

```
[1] "Ron Ingback" "Rhonda Ball"
```

or:

```
players[which(players$position == "RB"), "name"]
```

```
[1] "Ron Ingback" "Rhonda Ball"
```

3.9.3 Particular Columns (Variables)

To subset particular columns/variables, use the following syntax:

3.9.3.1 Base R

```
subsetVars <- c("name","age")

players[,c(2,4)]
```

```
               name age
1       Ken Cussion  40
2        Ben Sacked  30
3   Chuck Downfield  24
4       Ron Ingback  20
5       Rhonda Ball  18
6         Hugo Long  23
7  Lionel Scrimmage  27
8        Drew Blood  32
```

```
9      Chase Emdown  26
10      Justin Time  23
11     Spike D'Ball  NA
12       Isac Ulooz  37
```

```
players[,c("name","age")]
```

```
               name age
1       Ken Cussion  40
2        Ben Sacked  30
3   Chuck Downfield  24
4       Ron Ingback  20
5       Rhonda Ball  18
6         Hugo Long  23
7  Lionel Scrimmage  27
8        Drew Blood  32
9      Chase Emdown  26
10      Justin Time  23
11     Spike D'Ball  NA
12       Isac Ulooz  37
```

```
players[,subsetVars]
```

```
               name age
1       Ken Cussion  40
2        Ben Sacked  30
3   Chuck Downfield  24
4       Ron Ingback  20
5       Rhonda Ball  18
6         Hugo Long  23
7  Lionel Scrimmage  27
8        Drew Blood  32
9      Chase Emdown  26
10      Justin Time  23
11     Spike D'Ball  NA
12       Isac Ulooz  37
```

Or, to drop columns:

```
dropVars <- c("name","age")

players[,-c(2,4)]
```

```
   ID position
1   1       QB
2   2       QB
3   3       QB
4   4       RB
5   5       RB
6   6       WR
7   7       WR
8   8       WR
9   9       WR
10 10       TE
11 11       TE
12 12       LB
```

```
players[,!(names(players) %in% c("name","age"))]
```

```
   ID position
1   1       QB
2   2       QB
3   3       QB
4   4       RB
5   5       RB
6   6       WR
7   7       WR
8   8       WR
9   9       WR
10 10       TE
11 11       TE
12 12       LB
```

```
players[,!(names(players) %in% dropVars)]
```

```
   ID position
1   1       QB
2   2       QB
3   3       QB
4   4       RB
5   5       RB
6   6       WR
7   7       WR
8   8       WR
9   9       WR
10 10       TE
11 11       TE
12 12       LB
```

3.9.3.2 Tidyverse

```
players %>%
  select(name, age)
```

```
               name age
1       Ken Cussion  40
2        Ben Sacked  30
3   Chuck Downfield  24
4       Ron Ingback  20
5       Rhonda Ball  18
6         Hugo Long  23
7  Lionel Scrimmage  27
8        Drew Blood  32
9      Chase Emdown  26
10      Justin Time  23
11     Spike D'Ball  NA
12       Isac Ulooz  37
```

```
players %>%
  select(name:age)
```

```
               name position age
1       Ken Cussion       QB  40
2        Ben Sacked       QB  30
3   Chuck Downfield       QB  24
4       Ron Ingback       RB  20
5       Rhonda Ball       RB  18
6         Hugo Long       WR  23
7  Lionel Scrimmage       WR  27
8        Drew Blood       WR  32
9      Chase Emdown       WR  26
10      Justin Time       TE  23
11     Spike D'Ball       TE  NA
12       Isac Ulooz       LB  37
```

```
players %>%
  select(all_of(subsetVars))
```

```
               name age
1       Ken Cussion  40
2        Ben Sacked  30
```

```
3   Chuck Downfield  24
4       Ron Ingback  20
5       Rhonda Ball  18
6         Hugo Long  23
7  Lionel Scrimmage  27
8        Drew Blood  32
9      Chase Emdown  26
10      Justin Time  23
11     Spike D'Ball  NA
12       Isac Ulooz  37
```

Or, to drop columns:

```
players %>%
  select(-name, -age)
```

```
   ID position
1   1       QB
2   2       QB
3   3       QB
4   4       RB
5   5       RB
6   6       WR
7   7       WR
8   8       WR
9   9       WR
10 10       TE
11 11       TE
12 12       LB
```

```
players %>%
  select(-c(name:age))
```

```
   ID
1   1
2   2
3   3
4   4
5   5
6   6
7   7
8   8
9   9
10 10
```

```
11 11
12 12
```

```
players %>%
  select(-all_of(dropVars))
```

```
   ID position
1   1       QB
2   2       QB
3   3       QB
4   4       RB
5   5       RB
6   6       WR
7   7       WR
8   8       WR
9   9       WR
10 10       TE
11 11       TE
12 12       LB
```

3.9.4 Particular Rows

To subset particular rows, use the following syntax:

3.9.4.1 Base R

```
subsetRows <- c(4,5)

players[c(4,5),]
```

```
  ID        name position age
4  4 Ron Ingback       RB  20
5  5 Rhonda Ball       RB  18
```

```
players[subsetRows,]
```

```
  ID        name position age
4  4 Ron Ingback       RB  20
5  5 Rhonda Ball       RB  18
```

```
players[which(players$position == "RB"),]
```

```
  ID        name position age
4  4 Ron Ingback       RB  20
5  5 Rhonda Ball       RB  18
```

3.9.4.2 Tidyverse

```
players %>%
  filter(position == "WR")
```

```
  ID             name position age
1  6        Hugo Long       WR  23
2  7 Lionel Scrimmage       WR  27
3  8       Drew Blood       WR  32
4  9     Chase Emdown       WR  26
```

```
players %>%
  filter(position == "WR", age <= 26)
```

```
  ID         name position age
1  6    Hugo Long       WR  23
2  9 Chase Emdown       WR  26
```

```
players %>%
  filter(position == "WR" | age >= 26)
```

```
  ID             name position age
1  1      Ken Cussion       QB  40
2  2       Ben Sacked       QB  30
3  6        Hugo Long       WR  23
4  7 Lionel Scrimmage       WR  27
5  8       Drew Blood       WR  32
6  9     Chase Emdown       WR  26
7 12       Isac Ulooz       LB  37
```

3.9.5 Particular Rows and Columns

To subset particular rows and columns, use the following syntax:

3.9.5.1 Base R

```
players[c(4,5), c(2,4)]
```

```
         name age
4 Ron Ingback  20
5 Rhonda Ball  18
```

```
players[subsetRows, subsetVars]
```

```
         name age
4 Ron Ingback  20
5 Rhonda Ball  18
```

```
players[which(players$position == "RB"), subsetVars]
```

```
         name age
4 Ron Ingback  20
5 Rhonda Ball  18
```

3.9.5.2 Tidyverse

```
players %>%
  filter(position == "RB") %>%
  select(all_of(subsetVars))
```

```
         name age
1 Ron Ingback  20
2 Rhonda Ball  18
```

3.10 View Data

3.10.1 All Data

To view data, use the following syntax:

```
View(players)
```

3.10.2 First 6 Rows/Elements

To view only the first six rows (if a data frame) or elements (if a vector), use the following syntax:

```
head(nfl_players)
```

```
-- nflverse players -------------------------------------------------------

i Data updated: 2024-03-01 01:18:40 UTC

# A tibble: 6 x 33
  status display_name    first_name last_name esb_id  gsis_id suffix birth_date
  <chr>  <chr>           <chr>      <chr>     <chr>   <chr>   <chr>  <chr>
1 RET    'Omar Ellison    'Omar      Ellison   ELL711~ 00-000~ <NA>   <NA>
2 ACT    A'Shawn Robinson A'Shawn    Robinson  ROB367~ 00-003~ <NA>   1995-03-21
3 ACT    A.J. Arcuri      A.J.       Arcuri    ARC716~ 00-003~ <NA>   <NA>
4 RES    A.J. Bouye       Arlandus   Bouye     BOU651~ 00-003~ <NA>   1991-08-16
5 ACT    A.J. Brown       Arthur     Brown     BRO413~ 00-003~ <NA>   1997-06-30
6 ACT    A.J. Cann        Aaron      Cann      CAN364~ 00-003~ <NA>   1991-10-03
# i 25 more variables: college_name <chr>, position_group <chr>,
#   position <chr>, jersey_number <int>, height <dbl>, weight <int>,
#   years_of_experience <chr>, team_abbr <chr>, team_seq <int>,
#   current_team_id <chr>, football_name <chr>, entry_year <int>,
#   rookie_year <int>, draft_club <chr>, college_conference <chr>,
#   status_description_abbr <chr>, status_short_description <chr>,
#   gsis_it_id <int>, short_name <chr>, smart_id <chr>, headshot <chr>, ...
```

```
head(nfl_players$display_name)
```

```
[1] "'Omar Ellison"    "A'Shawn Robinson" "A.J. Arcuri"      "A.J. Bouye"
[5] "A.J. Brown"       "A.J. Cann"
```

3.11 Data Characteristics

3.11.1 Data Structure

```
str(nfl_players)
```

```
nflvrs_d [20,039 x 33] (S3: nflverse_data/tbl_df/tbl/data.table/data.frame)
 $ status                   : chr [1:20039] "RET" "ACT" "ACT" "RES" ...
 $ display_name             : chr [1:20039] "'Omar Ellison" "A'Shawn Robinson" "A.J. Arcuri" "A.J. Bouye" ..
 $ first_name               : chr [1:20039] "'Omar" "A'Shawn" "A.J." "Arlandus" ...
 $ last_name                : chr [1:20039] "Ellison" "Robinson" "Arcuri" "Bouye" ...
 $ esb_id                   : chr [1:20039] "ELL711319" "ROB367960" "ARC716900" "BOU651714" ...
 $ gsis_id                  : chr [1:20039] "00-0004866" "00-0032889" "00-0037845" "00-0030228" ...
 $ suffix                   : chr [1:20039] NA NA NA NA ...
 $ birth_date               : chr [1:20039] NA "1995-03-21" NA "1991-08-16" ...
 $ college_name             : chr [1:20039] NA "Alabama" "Michigan State" "Central Florida" ...
 $ position_group           : chr [1:20039] "WR" "DL" "OL" "DB" ...
 $ position                 : chr [1:20039] "WR" "DT" "T" "CB" ...
 $ jersey_number            : int [1:20039] 84 91 61 24 11 60 6 81 63 20 ...
 $ height                   : num [1:20039] 73 76 79 72 72 75 76 69 76 72 ...
 $ weight                   : int [1:20039] 200 330 320 191 226 325 220 190 280 183 ...
 $ years_of_experience      : chr [1:20039] "2" "8" "2" "8" ...
 $ team_abbr                : chr [1:20039] "LAC" "NYG" "LA" "CAR" ...
 $ team_seq                 : int [1:20039] NA 1 NA 1 1 1 1 NA NA NA ...
 $ current_team_id          : chr [1:20039] "4400" "3410" "2510" "0750" ...
 $ football_name            : chr [1:20039] NA "A'Shawn" "A.J." "A.J." ...
 $ entry_year               : int [1:20039] NA 2016 2022 2013 2019 2015 2019 NA NA NA ...
 $ rookie_year              : int [1:20039] NA 2016 2022 2013 2019 2015 2019 NA NA NA ...
 $ draft_club               : chr [1:20039] NA "DET" "LA" NA ...
 $ college_conference       : chr [1:20039] NA "Southeastern Conference" "Big Ten Conference" "American A
 $ status_description_abbr : chr [1:20039] NA "A01" "A01" "R01" ...
 $ status_short_description: chr [1:20039] NA "Active" "Active" "R/Injured" ...
 $ gsis_it_id               : int [1:20039] NA 43335 54726 40688 47834 42410 48335 NA NA NA ...
 $ short_name               : chr [1:20039] NA "A.Robinson" "A.Arcuri" "A.Bouye" ...
 $ smart_id                 : chr [1:20039] "3200454c-4c71-1319-728e-d49d3d236f8f" "3200524f-4236-7960-bf20
 $ headshot                 : chr [1:20039] NA "https://static.www.nfl.com/image/private/f_auto,q_auto/leag
 $ draft_number             : int [1:20039] NA 46 261 NA 51 67 NA NA NA NA ...
 $ uniform_number           : chr [1:20039] NA "91" "61" "24" ...
 $ draft_round              : chr [1:20039] NA NA NA NA ...
 $ season                   : int [1:20039] NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, "nflverse_type")= chr "players"
 - attr(*, "nflverse_timestamp")= POSIXct[1:1], format: "2024-03-01 01:18:40"
```

3.11.2 Data Dimensions

Number of rows and columns:

```
dim(nfl_players)
```

```
[1] 20039    33
```

3.11.3 Number of Elements

```
length(nfl_players$display_name)
```

```
[1] 20039
```

3.11.4 Number of Missing Elements

```
length(nfl_players$college_name[which(is.na(nfl_players$college_name))])
```

```
[1] 12127
```

3.11.5 Number of Non-Missing Elements

```
length(nfl_players$college_name[which(!is.na(nfl_players$college_name))])
```

```
[1] 7912
```

```
length(na.omit(nfl_players$college_name))
```

```
[1] 7912
```

3.12 Create New Variables

To create a new variable, use the following syntax:

```
players$newVar <- NA
```

Here is an example of creating a new variable:

```
players$newVar <- 1:nrow(players)
```

3.13 Recode Variables

Here is an example of recoding a variable:

```
players$oldVar1 <- NA
players$oldVar1[which(players$position == "QB")] <- "quarterback"
players$oldVar1[which(players$position == "RB")] <- "running back"
players$oldVar1[which(players$position == "WR")] <- "wide receiver"
players$oldVar1[which(players$position == "TE")] <- "tight end"

players$oldVar2 <- NA
players$oldVar2[which(players$age < 30)] <- "young"
players$oldVar2[which(players$age >= 30)] <- "old"
```

Recode multiple variables:

```
players %>%
  mutate(across(c(
    oldVar1:oldVar2),
    ~ case_match(
      .,
      c("quarterback","old","running back") ~ 0,
      c("wide receiver","tight end","young") ~ 1)))
```

```
   ID             name position age oldVar1 oldVar2
1   1      Ken Cussion       QB  40       0       0
2   2       Ben Sacked       QB  30       0       0
3   3  Chuck Downfield       QB  24       0       1
4   4      Ron Ingback       RB  20       0       1
5   5      Rhonda Ball       RB  18       0       1
6   6        Hugo Long       WR  23       1       1
7   7 Lionel Scrimmage       WR  27       1       1
8   8       Drew Blood       WR  32       1       0
```

```
9   9     Chase Emdown       WR  26       1       1
10 10      Justin Time       TE  23       1       1
11 11     Spike D'Ball       TE  NA       1      NA
12 12       Isac Ulooz       LB  37      NA       0
```

3.14 Rename Variables

```
players <- players %>%
  rename(
    newVar1 = oldVar1,
    newVar2 = oldVar2)
```

Using a vector of variable names:

```
varNamesFrom <- c("oldVar1","oldVar2")
varNamesTo <- c("newVar1","newVar2")

players <- players %>%
  rename_with(~ varNamesTo, all_of(varNamesFrom))
```

3.15 Convert the Types of Variables

One variable:

```
players$factorVar <- factor(players$ID)
players$numericVar <- as.numeric(players$age)
players$integerVar <- as.integer(players$newVar1)
```

```
Warning: NAs introduced by coercion
```

```
players$characterVar <- as.character(players$newVar2)
```

Multiple variables:

```
players %>%
  mutate(across(c(
    ID,
    age),
    as.numeric))
```

```
   ID          name position age     newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old        1         40
2   2      Ben Sacked       QB  30   quarterback     old        2         30
3   3  Chuck Downfield       QB  24   quarterback   young        3         24
4   4     Ron Ingback       RB  20  running back   young        4         20
5   5     Rhonda Ball       RB  18  running back   young        5         18
6   6       Hugo Long       WR  23 wide receiver   young        6         23
7   7 Lionel Scrimmage       WR  27 wide receiver   young        7         27
8   8      Drew Blood       WR  32 wide receiver     old        8         32
9   9    Chase Emdown       WR  26 wide receiver   young        9         26
10 10     Justin Time       TE  23     tight end   young       10         23
11 11    Spike D'Ball       TE  NA     tight end    <NA>       11         NA
12 12      Isac Ulooz       LB  37          <NA>     old       12         37
   integerVar characterVar
1          NA          old
2          NA          old
3          NA        young
4          NA        young
5          NA        young
6          NA        young
7          NA        young
8          NA          old
9          NA        young
10         NA        young
11         NA         <NA>
12         NA          old
```

```
players %>%
  mutate(across(
    age:newVar1,
    as.character))
```

```
   ID          name position  age     newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB   40   quarterback     old        1         40
2   2      Ben Sacked       QB   30   quarterback     old        2         30
3   3  Chuck Downfield       QB   24   quarterback   young        3         24
4   4     Ron Ingback       RB   20  running back   young        4         20
5   5     Rhonda Ball       RB   18  running back   young        5         18
```

```
6   6       Hugo Long      WR   23 wide receiver   young        6        23
7   7 Lionel Scrimmage      WR   27 wide receiver   young        7        27
8   8      Drew Blood       WR   32 wide receiver     old        8        32
9   9    Chase Emdown       WR   26 wide receiver   young        9        26
10 10     Justin Time       TE   23     tight end   young       10        23
11 11    Spike D'Ball       TE <NA>     tight end    <NA>       11        NA
12 12      Isac Ulooz       LB   37          <NA>     old       12        37
   integerVar characterVar
1          NA          old
2          NA          old
3          NA        young
4          NA        young
5          NA        young
6          NA        young
7          NA        young
8          NA          old
9          NA        young
10         NA        young
11         NA         <NA>
12         NA          old
```

```
players %>%
  mutate(across(where(is.factor), as.character))
```

```
  ID          name position age      newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old        1        40
2   2      Ben Sacked       QB  30   quarterback     old        2        30
3   3  Chuck Downfield      QB  24   quarterback   young        3        24
4   4     Ron Ingback       RB  20  running back   young        4        20
5   5     Rhonda Ball       RB  18  running back   young        5        18
6   6       Hugo Long      WR  23 wide receiver   young        6        23
7   7 Lionel Scrimmage      WR  27 wide receiver   young        7        27
8   8      Drew Blood       WR  32 wide receiver     old        8        32
9   9    Chase Emdown       WR  26 wide receiver   young        9        26
10 10     Justin Time       TE  23     tight end   young       10        23
11 11    Spike D'Ball       TE  NA     tight end    <NA>       11        NA
12 12      Isac Ulooz       LB  37          <NA>     old       12        37
   integerVar characterVar
1          NA          old
2          NA          old
3          NA        young
4          NA        young
5          NA        young
6          NA        young
7          NA        young
```

```
8           NA          old
9           NA        young
10          NA        young
11          NA         <NA>
12          NA          old
```

3.16 Merging/Joins

3.16.1 Overview

Merging (also called joining) merges two data objects using a shared set of variables called "keys." The keys are the variable(s) that uniquely identify each row (i.e., they account for the levels of nesting). In some data objects, the key might be the player's identification number (e.g., `player_id`). However, some data objects have multiple keys. For instance, in long form data objects, each participant may have multiple rows corresponding to multiple seasons. In this case, the keys may be `player_id` and `season`. If a participant has multiple rows corresponding to seasons and games/weeks, the keys are `player_id`, `season`, and `week`. In general, each row should have a value on each of the keys; there should be no missingness in the keys.

To merge two objects, the key(s) that will be used to match the records must be present in both objects. The keys are used to merge the variables in object 1 (`x`) with the variables in object 2 (`y`). Different merge types select different rows to merge.

Note: if the two objects include variables with the same name (apart from the keys), `R` will not know how you want each to appear in the merged object. So, it will add a suffix (e.g., `.x`, `.y`) to each common variable to indicate which object (i.e., object `x` or object `y`) the variable came from, where object `x` is the first object—i.e., the object to which object `y` (the second object) is merged. In general, apart from the keys, you should not include variables with the same name in two objects to be merged. To prevent this, either remove or rename the shared variable in one of the objects, or include the shared variable as a key. However, as described above, you should include it as a key ***only*** if it uniquely identifies each row in terms of levels of nesting.

3.16.2 Data Before Merging

Here are the data in the `players` object:

```
players
```

```
   ID          name position age      newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old        1         40
2   2      Ben Sacked       QB  30   quarterback     old        2         30
3   3  Chuck Downfield       QB  24   quarterback   young        3         24
4   4     Ron Ingback       RB  20  running back   young        4         20
5   5     Rhonda Ball       RB  18  running back   young        5         18
6   6       Hugo Long       WR  23 wide receiver   young        6         23
7   7 Lionel Scrimmage       WR  27 wide receiver   young        7         27
8   8      Drew Blood       WR  32 wide receiver     old        8         32
9   9    Chase Emdown       WR  26 wide receiver   young        9         26
10 10     Justin Time       TE  23     tight end   young       10         23
11 11    Spike D'Ball       TE  NA     tight end    <NA>       11         NA
12 12      Isac Ulooz       LB  37          <NA>     old       12         37
   integerVar characterVar
1          NA          old
2          NA          old
3          NA        young
4          NA        young
5          NA        young
6          NA        young
7          NA        young
8          NA          old
9          NA        young
10         NA        young
11         NA         <NA>
12         NA          old
```

```
dim(players)
```

```
[1] 12 10
```

The data are structured in ID form. That is, every row in the dataset is uniquely identified by the variable, `ID`.

Here are the data in the `fantasyPoints` object:

```
fantasyPoints
```

```
  ID fantasyPoints
1  2           250
2  7           170
3 13            65
4 14            15
```

```
dim(fantasyPoints)
```

```
[1] 4 2
```

3.16.3 Types of Joins

3.16.3.1 Visual Overview of Join Types

Below is a visual that depicts various types of merges/joins. Object `x` is the circle labeled as `x`. Object `y` is the circle labeled as `y`. The area of overlap in the Venn diagram indicates the rows on the keys that are shared between the two objects (e.g., the same `player_id`, `season`, and `week`). The non-overlapping area indicates the rows on the keys that are unique to each object. The shaded blue area indicates which rows (on the keys) are kept in the merged object from each of the two objects, when using each of the merge types. For instance, a left outer join keeps the shared rows and the rows that are unique to object `x`, but it drops the rows that are unique to object `y`.

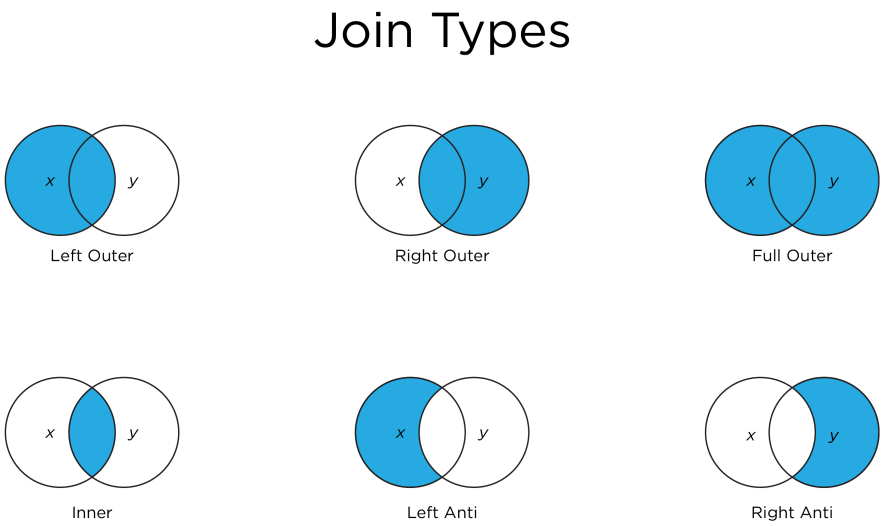

Figure 3.1 Types of merges/joins

3.16.3.2 Full Outer Join

A full outer join includes all rows in x **or** y. It returns columns from x and y. Here is how to merge two data frames using a full outer join (i.e., "full join"):

```
fullJoinData <- full_join(
  players,
  fantasyPoints,
  by = "ID")

fullJoinData
```

```
   ID          name position age      newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old         1         40
2   2      Ben Sacked       QB  30   quarterback     old         2         30
3   3  Chuck Downfield       QB  24   quarterback   young         3         24
4   4     Ron Ingback       RB  20  running back   young         4         20
5   5     Rhonda Ball       RB  18  running back   young         5         18
6   6       Hugo Long       WR  23 wide receiver   young         6         23
7   7 Lionel Scrimmage       WR  27 wide receiver   young         7         27
8   8      Drew Blood       WR  32 wide receiver     old         8         32
9   9    Chase Emdown       WR  26 wide receiver   young         9         26
10 10     Justin Time       TE  23     tight end   young        10         23
11 11    Spike D'Ball       TE  NA     tight end    <NA>        11         NA
12 12      Isac Ulooz       LB  37          <NA>     old        12         37
13 13            <NA>     <NA>  NA          <NA>    <NA>      <NA>         NA
14 14            <NA>     <NA>  NA          <NA>    <NA>      <NA>         NA
   integerVar characterVar fantasyPoints
1          NA          old            NA
2          NA          old           250
3          NA        young            NA
4          NA        young            NA
5          NA        young            NA
6          NA        young            NA
7          NA        young           170
8          NA          old            NA
9          NA        young            NA
10         NA        young            NA
11         NA         <NA>            NA
12         NA          old            NA
13         NA         <NA>            65
14         NA         <NA>            15
```

```
dim(fullJoinData)
```

```
[1] 14 11
```

3.16.3.3 Left Outer Join

A left outer join includes all rows in x. It returns columns from x and y. Here is how to merge two data frames using a left outer join ("left join"):

```
leftJoinData <- left_join(
  players,
  fantasyPoints,
  by = "ID")

leftJoinData
```

```
   ID          name position age     newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old         1         40
2   2      Ben Sacked       QB  30   quarterback     old         2         30
3   3  Chuck Downfield       QB  24   quarterback   young         3         24
4   4     Ron Ingback       RB  20  running back   young         4         20
5   5     Rhonda Ball       RB  18  running back   young         5         18
6   6       Hugo Long       WR  23 wide receiver   young         6         23
7   7 Lionel Scrimmage       WR  27 wide receiver   young         7         27
8   8      Drew Blood       WR  32 wide receiver     old         8         32
9   9    Chase Emdown       WR  26 wide receiver   young         9         26
10 10     Justin Time       TE  23     tight end   young        10         23
11 11    Spike D'Ball       TE  NA     tight end    <NA>        11         NA
12 12      Isac Ulooz       LB  37         <NA>     old        12         37
   integerVar characterVar fantasyPoints
1          NA          old            NA
2          NA          old           250
3          NA        young            NA
4          NA        young            NA
5          NA        young            NA
6          NA        young            NA
7          NA        young           170
8          NA          old            NA
9          NA        young            NA
10         NA        young            NA
11         NA         <NA>            NA
12         NA          old            NA
```

```
dim(leftJoinData)
```

```
[1] 12 11
```

3.16.3.4 Right Outer Join

A right outer join includes all rows in y. It returns columns from x and y. Here is how to merge two data frames using a right outer join ("right join"):

```
rightJoinData <- right_join(
  players,
  fantasyPoints,
  by = "ID")

rightJoinData
```

```
  ID          name position age      newVar1 newVar2 factorVar numericVar
1  2      Ben Sacked       QB  30   quarterback     old        2         30
2  7 Lionel Scrimmage       WR  27 wide receiver   young        7         27
3 13            <NA>     <NA>  NA          <NA>    <NA>      <NA>         NA
4 14            <NA>     <NA>  NA          <NA>    <NA>      <NA>         NA
  integerVar characterVar fantasyPoints
1         NA          old           250
2         NA        young           170
3         NA         <NA>            65
4         NA         <NA>            15
```

```
dim(rightJoinData)
```

```
[1]  4 11
```

3.16.3.5 Inner Join

An inner join includes all rows that are in **both** x **and** y. An inner join will return one row of x for each matching row of y, and can duplicate values of records on either side (left or right) if x and y have more than one matching record. It returns columns from x and y. Here is how to merge two data frames using an inner join:

```
innerJoinData <- inner_join(
  players,
  fantasyPoints,
  by = "ID")

innerJoinData
```

```
  ID          name position age      newVar1 newVar2 factorVar numericVar
```

```
1  2      Ben Sacked      QB  30   quarterback     old        2        30
2  7 Lionel Scrimmage      WR  27 wide receiver   young       7        27
  integerVar characterVar fantasyPoints
1         NA          old           250
2         NA        young           170
```

```
dim(innerJoinData)
```

```
[1]  2 11
```

3.16.3.6 Semi Join

A semi join is a filter. A left semi join returns all rows from x **with** a match in y. That is, it filters out records from x that are not in y. Unlike an inner join, a left semi join will never duplicate rows of x, and it includes columns from only x (not from y). Here is how to merge two data frames using a left semi join:

```
semiJoinData <- semi_join(
  players,
  fantasyPoints,
  by = "ID")

semiJoinData
```

```
  ID          name position age      newVar1 newVar2 factorVar numericVar
1  2      Ben Sacked      QB  30   quarterback     old        2        30
2  7 Lionel Scrimmage      WR  27 wide receiver   young       7        27
  integerVar characterVar
1         NA          old
2         NA        young
```

```
dim(semiJoinData)
```

```
[1]  2 10
```

3.16.3.7 Anti Join

An anti join is a filter. A left anti join returns all rows from x **without** a match in y. That is, it filters out records from x that are in y. It returns columns from only x (not from y). Here is how to merge two data frames using a left anti join:

```
antiJoinData <- anti_join(
  players,
  fantasyPoints,
  by = "ID")

antiJoinData
```

```
   ID          name position age      newVar1 newVar2 factorVar numericVar
1   1     Ken Cussion       QB  40   quarterback     old         1         40
2   3 Chuck Downfield       QB  24   quarterback   young         3         24
3   4     Ron Ingback       RB  20  running back   young         4         20
4   5     Rhonda Ball       RB  18  running back   young         5         18
5   6       Hugo Long       WR  23 wide receiver   young         6         23
6   8      Drew Blood       WR  32 wide receiver     old         8         32
7   9    Chase Emdown       WR  26 wide receiver   young         9         26
8  10     Justin Time       TE  23     tight end   young        10         23
9  11    Spike D'Ball       TE  NA     tight end    <NA>        11         NA
10 12      Isac Ulooz       LB  37          <NA>     old        12         37
   integerVar characterVar
1          NA          old
2          NA        young
3          NA        young
4          NA        young
5          NA        young
6          NA          old
7          NA        young
8          NA        young
9          NA         <NA>
10         NA          old
```

```
dim(antiJoinData)
```

```
[1] 10 10
```

3.16.3.8 Cross Join

A cross join combines each row in x with each row in y.

```
crossJoinData <- cross_join(
  players,
  fantasyPoints)

crossJoinData
```

	ID.x	name	position	age	newVar1	newVar2	factorVar
1	1	Ken Cussion	QB	40	quarterback	old	1
2	1	Ken Cussion	QB	40	quarterback	old	1
3	1	Ken Cussion	QB	40	quarterback	old	1
4	1	Ken Cussion	QB	40	quarterback	old	1
5	2	Ben Sacked	QB	30	quarterback	old	2
6	2	Ben Sacked	QB	30	quarterback	old	2
7	2	Ben Sacked	QB	30	quarterback	old	2
8	2	Ben Sacked	QB	30	quarterback	old	2
9	3	Chuck Downfield	QB	24	quarterback	young	3
10	3	Chuck Downfield	QB	24	quarterback	young	3
11	3	Chuck Downfield	QB	24	quarterback	young	3
12	3	Chuck Downfield	QB	24	quarterback	young	3
13	4	Ron Ingback	RB	20	running back	young	4
14	4	Ron Ingback	RB	20	running back	young	4
15	4	Ron Ingback	RB	20	running back	young	4
16	4	Ron Ingback	RB	20	running back	young	4
17	5	Rhonda Ball	RB	18	running back	young	5
18	5	Rhonda Ball	RB	18	running back	young	5
19	5	Rhonda Ball	RB	18	running back	young	5
20	5	Rhonda Ball	RB	18	running back	young	5
21	6	Hugo Long	WR	23	wide receiver	young	6
22	6	Hugo Long	WR	23	wide receiver	young	6
23	6	Hugo Long	WR	23	wide receiver	young	6
24	6	Hugo Long	WR	23	wide receiver	young	6
25	7	Lionel Scrimmage	WR	27	wide receiver	young	7
26	7	Lionel Scrimmage	WR	27	wide receiver	young	7
27	7	Lionel Scrimmage	WR	27	wide receiver	young	7
28	7	Lionel Scrimmage	WR	27	wide receiver	young	7
29	8	Drew Blood	WR	32	wide receiver	old	8
30	8	Drew Blood	WR	32	wide receiver	old	8
31	8	Drew Blood	WR	32	wide receiver	old	8
32	8	Drew Blood	WR	32	wide receiver	old	8
33	9	Chase Emdown	WR	26	wide receiver	young	9
34	9	Chase Emdown	WR	26	wide receiver	young	9
35	9	Chase Emdown	WR	26	wide receiver	young	9
36	9	Chase Emdown	WR	26	wide receiver	young	9
37	10	Justin Time	TE	23	tight end	young	10
38	10	Justin Time	TE	23	tight end	young	10
39	10	Justin Time	TE	23	tight end	young	10
40	10	Justin Time	TE	23	tight end	young	10
41	11	Spike D'Ball	TE	NA	tight end	<NA>	11
42	11	Spike D'Ball	TE	NA	tight end	<NA>	11
43	11	Spike D'Ball	TE	NA	tight end	<NA>	11
44	11	Spike D'Ball	TE	NA	tight end	<NA>	11

```
45   12       Isac Ulooz       LB  37          <NA>     old        12
46   12       Isac Ulooz       LB  37          <NA>     old        12
47   12       Isac Ulooz       LB  37          <NA>     old        12
48   12       Isac Ulooz       LB  37          <NA>     old        12
   numericVar integerVar characterVar ID.y fantasyPoints
1          40         NA          old    2           250
2          40         NA          old    7           170
3          40         NA          old   13            65
4          40         NA          old   14            15
5          30         NA          old    2           250
6          30         NA          old    7           170
7          30         NA          old   13            65
8          30         NA          old   14            15
9          24         NA        young    2           250
10         24         NA        young    7           170
11         24         NA        young   13            65
12         24         NA        young   14            15
13         20         NA        young    2           250
14         20         NA        young    7           170
15         20         NA        young   13            65
16         20         NA        young   14            15
17         18         NA        young    2           250
18         18         NA        young    7           170
19         18         NA        young   13            65
20         18         NA        young   14            15
21         23         NA        young    2           250
22         23         NA        young    7           170
23         23         NA        young   13            65
24         23         NA        young   14            15
25         27         NA        young    2           250
26         27         NA        young    7           170
27         27         NA        young   13            65
28         27         NA        young   14            15
29         32         NA          old    2           250
30         32         NA          old    7           170
31         32         NA          old   13            65
32         32         NA          old   14            15
33         26         NA        young    2           250
34         26         NA        young    7           170
35         26         NA        young   13            65
36         26         NA        young   14            15
37         23         NA        young    2           250
38         23         NA        young    7           170
39         23         NA        young   13            65
40         23         NA        young   14            15
```

```
41          NA          NA          <NA>    2           250
42          NA          NA          <NA>    7           170
43          NA          NA          <NA>   13            65
44          NA          NA          <NA>   14            15
45          37          NA           old    2           250
46          37          NA           old    7           170
47          37          NA           old   13            65
48          37          NA           old   14            15
```

```
dim(crossJoinData)
```

```
[1] 48 12
```

3.17 Transform Data from Long to Wide

Here are the data in the `nfl_actualStats_offense_weekly` object. The data are structured in player-season-week form. That is, every row in the dataset is uniquely identified by the variables, `player_id`, `season`, and `week`.

Original data:

```
dataLong <- nfl_actualStats_offense_weekly %>%
  select(player_id, player_display_name, season, week, fantasy_points)

dim(dataLong)
```

```
[1] 129739      5
```

```
names(dataLong)
```

```
[1] "player_id"          "player_display_name" "season"
[4] "week"               "fantasy_points"
```

Below, we widen the data widened by two variables (`season` and `week`), using `tidyverse`, so that the data are now in player form (where each row is uniquely identified by the `player_id` variable):

```
dataWide <- dataLong %>%
  pivot_wider(
    names_from = c(season, week),
    names_glue = "{.value}_{season}_week{week}",
    values_from = fantasy_points)

dim(dataWide)
```

```
[1] 4021  530
```

```
names(dataWide)
```

```
  [1] "player_id"                  "player_display_name"
  [3] "fantasy_points_1999_week1"  "fantasy_points_1999_week2"
  [5] "fantasy_points_1999_week4"  "fantasy_points_1999_week7"
  [7] "fantasy_points_1999_week8"  "fantasy_points_1999_week9"
  [9] "fantasy_points_1999_week10" "fantasy_points_1999_week11"
 [11] "fantasy_points_1999_week12" "fantasy_points_1999_week13"
 [13] "fantasy_points_1999_week14" "fantasy_points_1999_week15"
 [15] "fantasy_points_1999_week16" "fantasy_points_1999_week5"
 [17] "fantasy_points_1999_week6"  "fantasy_points_1999_week17"
 [19] "fantasy_points_1999_week18" "fantasy_points_1999_week3"
 [21] "fantasy_points_1999_week19" "fantasy_points_1999_week20"
 [23] "fantasy_points_1999_week21" "fantasy_points_2000_week1"
 [25] "fantasy_points_2000_week12" "fantasy_points_2000_week14"
 [27] "fantasy_points_2000_week15" "fantasy_points_2000_week6"
 [29] "fantasy_points_2000_week10" "fantasy_points_2000_week4"
 [31] "fantasy_points_2000_week5"  "fantasy_points_2000_week7"
 [33] "fantasy_points_2000_week8"  "fantasy_points_2000_week9"
 [35] "fantasy_points_2000_week11" "fantasy_points_2000_week13"
 [37] "fantasy_points_2000_week2"  "fantasy_points_2000_week16"
 [39] "fantasy_points_2000_week17" "fantasy_points_2000_week3"
 [41] "fantasy_points_2000_week18" "fantasy_points_2000_week19"
 [43] "fantasy_points_2000_week21" "fantasy_points_2000_week20"
 [45] "fantasy_points_2001_week15" "fantasy_points_2001_week17"
 [47] "fantasy_points_2001_week1"  "fantasy_points_2001_week3"
 [49] "fantasy_points_2001_week4"  "fantasy_points_2001_week5"
 [51] "fantasy_points_2001_week6"  "fantasy_points_2001_week9"
 [53] "fantasy_points_2001_week11" "fantasy_points_2001_week12"
 [55] "fantasy_points_2001_week13" "fantasy_points_2001_week14"
 [57] "fantasy_points_2001_week16" "fantasy_points_2001_week2"
 [59] "fantasy_points_2001_week7"  "fantasy_points_2001_week8"
 [61] "fantasy_points_2001_week10" "fantasy_points_2001_week19"
 [63] "fantasy_points_2001_week18" "fantasy_points_2001_week20"
```

```
 [65] "fantasy_points_2001_week21" "fantasy_points_2002_week3"
 [67] "fantasy_points_2002_week1"  "fantasy_points_2002_week2"
 [69] "fantasy_points_2002_week4"  "fantasy_points_2002_week6"
 [71] "fantasy_points_2002_week7"  "fantasy_points_2002_week8"
 [73] "fantasy_points_2002_week9"  "fantasy_points_2002_week5"
 [75] "fantasy_points_2002_week10" "fantasy_points_2002_week11"
 [77] "fantasy_points_2002_week12" "fantasy_points_2002_week13"
 [79] "fantasy_points_2002_week14" "fantasy_points_2002_week15"
 [81] "fantasy_points_2002_week16" "fantasy_points_2002_week17"
 [83] "fantasy_points_2002_week19" "fantasy_points_2002_week20"
 [85] "fantasy_points_2002_week21" "fantasy_points_2002_week18"
 [87] "fantasy_points_2003_week2"  "fantasy_points_2003_week4"
 [89] "fantasy_points_2003_week5"  "fantasy_points_2003_week7"
 [91] "fantasy_points_2003_week8"  "fantasy_points_2003_week9"
 [93] "fantasy_points_2003_week10" "fantasy_points_2003_week11"
 [95] "fantasy_points_2003_week13" "fantasy_points_2003_week14"
 [97] "fantasy_points_2003_week15" "fantasy_points_2003_week17"
 [99] "fantasy_points_2003_week1"  "fantasy_points_2003_week3"
[101] "fantasy_points_2003_week6"  "fantasy_points_2003_week12"
[103] "fantasy_points_2003_week16" "fantasy_points_2003_week18"
[105] "fantasy_points_2003_week19" "fantasy_points_2003_week20"
[107] "fantasy_points_2003_week21" "fantasy_points_2004_week2"
[109] "fantasy_points_2004_week5"  "fantasy_points_2004_week6"
[111] "fantasy_points_2004_week10" "fantasy_points_2004_week16"
[113] "fantasy_points_2004_week17" "fantasy_points_2004_week1"
[115] "fantasy_points_2004_week3"  "fantasy_points_2004_week7"
[117] "fantasy_points_2004_week8"  "fantasy_points_2004_week9"
[119] "fantasy_points_2004_week11" "fantasy_points_2004_week12"
[121] "fantasy_points_2004_week13" "fantasy_points_2004_week14"
[123] "fantasy_points_2004_week15" "fantasy_points_2004_week4"
[125] "fantasy_points_2004_week19" "fantasy_points_2004_week20"
[127] "fantasy_points_2004_week21" "fantasy_points_2004_week18"
[129] "fantasy_points_2005_week1"  "fantasy_points_2005_week2"
[131] "fantasy_points_2005_week3"  "fantasy_points_2005_week4"
[133] "fantasy_points_2005_week6"  "fantasy_points_2005_week7"
[135] "fantasy_points_2005_week8"  "fantasy_points_2005_week10"
[137] "fantasy_points_2005_week11" "fantasy_points_2005_week13"
[139] "fantasy_points_2005_week14" "fantasy_points_2005_week15"
[141] "fantasy_points_2005_week16" "fantasy_points_2005_week17"
[143] "fantasy_points_2005_week5"  "fantasy_points_2005_week9"
[145] "fantasy_points_2005_week12" "fantasy_points_2005_week18"
[147] "fantasy_points_2005_week19" "fantasy_points_2005_week20"
[149] "fantasy_points_2005_week21" "fantasy_points_2006_week4"
[151] "fantasy_points_2006_week1"  "fantasy_points_2006_week2"
[153] "fantasy_points_2006_week3"  "fantasy_points_2006_week10"
```

```
[155] "fantasy_points_2006_week11" "fantasy_points_2006_week12"
[157] "fantasy_points_2006_week13" "fantasy_points_2006_week14"
[159] "fantasy_points_2006_week5"  "fantasy_points_2006_week6"
[161] "fantasy_points_2006_week7"  "fantasy_points_2006_week15"
[163] "fantasy_points_2006_week16" "fantasy_points_2006_week17"
[165] "fantasy_points_2006_week8"  "fantasy_points_2006_week9"
[167] "fantasy_points_2006_week18" "fantasy_points_2006_week19"
[169] "fantasy_points_2006_week20" "fantasy_points_2006_week21"
[171] "fantasy_points_2007_week1"  "fantasy_points_2007_week2"
[173] "fantasy_points_2007_week3"  "fantasy_points_2007_week5"
[175] "fantasy_points_2007_week9"  "fantasy_points_2007_week16"
[177] "fantasy_points_2007_week17" "fantasy_points_2007_week14"
[179] "fantasy_points_2007_week4"  "fantasy_points_2007_week6"
[181] "fantasy_points_2007_week7"  "fantasy_points_2007_week8"
[183] "fantasy_points_2007_week10" "fantasy_points_2007_week11"
[185] "fantasy_points_2007_week12" "fantasy_points_2007_week13"
[187] "fantasy_points_2007_week15" "fantasy_points_2007_week19"
[189] "fantasy_points_2007_week21" "fantasy_points_2007_week18"
[191] "fantasy_points_2007_week20" "fantasy_points_2008_week8"
[193] "fantasy_points_2008_week1"  "fantasy_points_2008_week2"
[195] "fantasy_points_2008_week3"  "fantasy_points_2008_week4"
[197] "fantasy_points_2008_week5"  "fantasy_points_2008_week6"
[199] "fantasy_points_2008_week7"  "fantasy_points_2008_week14"
[201] "fantasy_points_2008_week10" "fantasy_points_2008_week11"
[203] "fantasy_points_2008_week12" "fantasy_points_2008_week13"
[205] "fantasy_points_2008_week15" "fantasy_points_2008_week16"
[207] "fantasy_points_2008_week17" "fantasy_points_2008_week9"
[209] "fantasy_points_2008_week19" "fantasy_points_2008_week18"
[211] "fantasy_points_2008_week20" "fantasy_points_2008_week21"
[213] "fantasy_points_2009_week9"  "fantasy_points_2009_week11"
[215] "fantasy_points_2009_week2"  "fantasy_points_2009_week3"
[217] "fantasy_points_2009_week5"  "fantasy_points_2009_week7"
[219] "fantasy_points_2009_week12" "fantasy_points_2009_week13"
[221] "fantasy_points_2009_week14" "fantasy_points_2009_week15"
[223] "fantasy_points_2009_week16" "fantasy_points_2009_week17"
[225] "fantasy_points_2009_week1"  "fantasy_points_2009_week4"
[227] "fantasy_points_2009_week8"  "fantasy_points_2009_week6"
[229] "fantasy_points_2009_week10" "fantasy_points_2009_week18"
[231] "fantasy_points_2009_week19" "fantasy_points_2009_week20"
[233] "fantasy_points_2009_week21" "fantasy_points_2010_week2"
[235] "fantasy_points_2010_week3"  "fantasy_points_2010_week4"
[237] "fantasy_points_2010_week1"  "fantasy_points_2010_week7"
[239] "fantasy_points_2010_week17" "fantasy_points_2010_week6"
[241] "fantasy_points_2010_week8"  "fantasy_points_2010_week10"
[243] "fantasy_points_2010_week13" "fantasy_points_2010_week14"
```

```
[245] "fantasy_points_2010_week15" "fantasy_points_2010_week16"
[247] "fantasy_points_2010_week5"  "fantasy_points_2010_week20"
[249] "fantasy_points_2010_week12" "fantasy_points_2010_week11"
[251] "fantasy_points_2010_week18" "fantasy_points_2010_week19"
[253] "fantasy_points_2010_week21" "fantasy_points_2010_week9"
[255] "fantasy_points_2011_week17" "fantasy_points_2011_week13"
[257] "fantasy_points_2011_week14" "fantasy_points_2011_week16"
[259] "fantasy_points_2011_week15" "fantasy_points_2011_week1"
[261] "fantasy_points_2011_week2"  "fantasy_points_2011_week3"
[263] "fantasy_points_2011_week4"  "fantasy_points_2011_week5"
[265] "fantasy_points_2011_week6"  "fantasy_points_2011_week7"
[267] "fantasy_points_2011_week9"  "fantasy_points_2011_week10"
[269] "fantasy_points_2011_week11" "fantasy_points_2011_week12"
[271] "fantasy_points_2011_week19" "fantasy_points_2011_week8"
[273] "fantasy_points_2011_week18" "fantasy_points_2011_week20"
[275] "fantasy_points_2011_week21" "fantasy_points_2012_week12"
[277] "fantasy_points_2012_week13" "fantasy_points_2012_week2"
[279] "fantasy_points_2012_week3"  "fantasy_points_2012_week4"
[281] "fantasy_points_2012_week5"  "fantasy_points_2012_week7"
[283] "fantasy_points_2012_week8"  "fantasy_points_2012_week9"
[285] "fantasy_points_2012_week11" "fantasy_points_2012_week16"
[287] "fantasy_points_2012_week1"  "fantasy_points_2012_week6"
[289] "fantasy_points_2012_week10" "fantasy_points_2012_week14"
[291] "fantasy_points_2012_week15" "fantasy_points_2012_week17"
[293] "fantasy_points_2012_week19" "fantasy_points_2012_week20"
[295] "fantasy_points_2012_week21" "fantasy_points_2012_week18"
[297] "fantasy_points_2013_week1"  "fantasy_points_2013_week2"
[299] "fantasy_points_2013_week3"  "fantasy_points_2013_week4"
[301] "fantasy_points_2013_week5"  "fantasy_points_2013_week7"
[303] "fantasy_points_2013_week8"  "fantasy_points_2013_week9"
[305] "fantasy_points_2013_week10" "fantasy_points_2013_week11"
[307] "fantasy_points_2013_week12" "fantasy_points_2013_week13"
[309] "fantasy_points_2013_week14" "fantasy_points_2013_week15"
[311] "fantasy_points_2013_week16" "fantasy_points_2013_week17"
[313] "fantasy_points_2013_week6"  "fantasy_points_2013_week19"
[315] "fantasy_points_2013_week20" "fantasy_points_2013_week21"
[317] "fantasy_points_2013_week18" "fantasy_points_2014_week3"
[319] "fantasy_points_2014_week4"  "fantasy_points_2014_week16"
[321] "fantasy_points_2014_week17" "fantasy_points_2014_week1"
[323] "fantasy_points_2014_week2"  "fantasy_points_2014_week5"
[325] "fantasy_points_2014_week6"  "fantasy_points_2014_week7"
[327] "fantasy_points_2014_week8"  "fantasy_points_2014_week9"
[329] "fantasy_points_2014_week10" "fantasy_points_2014_week11"
[331] "fantasy_points_2014_week12" "fantasy_points_2014_week13"
[333] "fantasy_points_2014_week14" "fantasy_points_2014_week15"
```

```
[335] "fantasy_points_2014_week19" "fantasy_points_2014_week20"
[337] "fantasy_points_2014_week21" "fantasy_points_2014_week18"
[339] "fantasy_points_2015_week4"  "fantasy_points_2015_week5"
[341] "fantasy_points_2015_week11" "fantasy_points_2015_week12"
[343] "fantasy_points_2015_week13" "fantasy_points_2015_week14"
[345] "fantasy_points_2015_week15" "fantasy_points_2015_week16"
[347] "fantasy_points_2015_week1"  "fantasy_points_2015_week2"
[349] "fantasy_points_2015_week3"  "fantasy_points_2015_week6"
[351] "fantasy_points_2015_week8"  "fantasy_points_2015_week9"
[353] "fantasy_points_2015_week10" "fantasy_points_2015_week17"
[355] "fantasy_points_2015_week19" "fantasy_points_2015_week20"
[357] "fantasy_points_2015_week21" "fantasy_points_2015_week7"
[359] "fantasy_points_2015_week18" "fantasy_points_2016_week5"
[361] "fantasy_points_2016_week6"  "fantasy_points_2016_week7"
[363] "fantasy_points_2016_week8"  "fantasy_points_2016_week10"
[365] "fantasy_points_2016_week11" "fantasy_points_2016_week12"
[367] "fantasy_points_2016_week13" "fantasy_points_2016_week14"
[369] "fantasy_points_2016_week15" "fantasy_points_2016_week16"
[371] "fantasy_points_2016_week17" "fantasy_points_2016_week19"
[373] "fantasy_points_2016_week20" "fantasy_points_2016_week21"
[375] "fantasy_points_2016_week1"  "fantasy_points_2016_week2"
[377] "fantasy_points_2016_week3"  "fantasy_points_2016_week4"
[379] "fantasy_points_2016_week9"  "fantasy_points_2016_week18"
[381] "fantasy_points_2017_week1"  "fantasy_points_2017_week2"
[383] "fantasy_points_2017_week3"  "fantasy_points_2017_week4"
[385] "fantasy_points_2017_week5"  "fantasy_points_2017_week6"
[387] "fantasy_points_2017_week7"  "fantasy_points_2017_week8"
[389] "fantasy_points_2017_week10" "fantasy_points_2017_week11"
[391] "fantasy_points_2017_week12" "fantasy_points_2017_week13"
[393] "fantasy_points_2017_week14" "fantasy_points_2017_week15"
[395] "fantasy_points_2017_week16" "fantasy_points_2017_week17"
[397] "fantasy_points_2017_week19" "fantasy_points_2017_week20"
[399] "fantasy_points_2017_week21" "fantasy_points_2017_week9"
[401] "fantasy_points_2017_week18" "fantasy_points_2018_week1"
[403] "fantasy_points_2018_week2"  "fantasy_points_2018_week3"
[405] "fantasy_points_2018_week4"  "fantasy_points_2018_week5"
[407] "fantasy_points_2018_week6"  "fantasy_points_2018_week7"
[409] "fantasy_points_2018_week8"  "fantasy_points_2018_week9"
[411] "fantasy_points_2018_week10" "fantasy_points_2018_week12"
[413] "fantasy_points_2018_week13" "fantasy_points_2018_week14"
[415] "fantasy_points_2018_week15" "fantasy_points_2018_week16"
[417] "fantasy_points_2018_week17" "fantasy_points_2018_week19"
[419] "fantasy_points_2018_week20" "fantasy_points_2018_week21"
[421] "fantasy_points_2018_week11" "fantasy_points_2018_week18"
[423] "fantasy_points_2019_week1"  "fantasy_points_2019_week2"
```

```
[425] "fantasy_points_2019_week3"  "fantasy_points_2019_week4"
[427] "fantasy_points_2019_week5"  "fantasy_points_2019_week6"
[429] "fantasy_points_2019_week7"  "fantasy_points_2019_week8"
[431] "fantasy_points_2019_week9"  "fantasy_points_2019_week11"
[433] "fantasy_points_2019_week12" "fantasy_points_2019_week13"
[435] "fantasy_points_2019_week14" "fantasy_points_2019_week15"
[437] "fantasy_points_2019_week16" "fantasy_points_2019_week17"
[439] "fantasy_points_2019_week18" "fantasy_points_2019_week10"
[441] "fantasy_points_2019_week19" "fantasy_points_2019_week20"
[443] "fantasy_points_2019_week21" "fantasy_points_2020_week1"
[445] "fantasy_points_2020_week2"  "fantasy_points_2020_week3"
[447] "fantasy_points_2020_week4"  "fantasy_points_2020_week5"
[449] "fantasy_points_2020_week6"  "fantasy_points_2020_week7"
[451] "fantasy_points_2020_week8"  "fantasy_points_2020_week9"
[453] "fantasy_points_2020_week10" "fantasy_points_2020_week11"
[455] "fantasy_points_2020_week12" "fantasy_points_2020_week14"
[457] "fantasy_points_2020_week15" "fantasy_points_2020_week16"
[459] "fantasy_points_2020_week17" "fantasy_points_2020_week18"
[461] "fantasy_points_2020_week19" "fantasy_points_2020_week20"
[463] "fantasy_points_2020_week21" "fantasy_points_2020_week13"
[465] "fantasy_points_2021_week1"  "fantasy_points_2021_week2"
[467] "fantasy_points_2021_week3"  "fantasy_points_2021_week4"
[469] "fantasy_points_2021_week5"  "fantasy_points_2021_week6"
[471] "fantasy_points_2021_week7"  "fantasy_points_2021_week8"
[473] "fantasy_points_2021_week10" "fantasy_points_2021_week11"
[475] "fantasy_points_2021_week12" "fantasy_points_2021_week13"
[477] "fantasy_points_2021_week14" "fantasy_points_2021_week15"
[479] "fantasy_points_2021_week16" "fantasy_points_2021_week17"
[481] "fantasy_points_2021_week18" "fantasy_points_2021_week19"
[483] "fantasy_points_2021_week20" "fantasy_points_2021_week9"
[485] "fantasy_points_2021_week21" "fantasy_points_2021_week22"
[487] "fantasy_points_2022_week1"  "fantasy_points_2022_week2"
[489] "fantasy_points_2022_week3"  "fantasy_points_2022_week4"
[491] "fantasy_points_2022_week5"  "fantasy_points_2022_week6"
[493] "fantasy_points_2022_week7"  "fantasy_points_2022_week8"
[495] "fantasy_points_2022_week9"  "fantasy_points_2022_week10"
[497] "fantasy_points_2022_week12" "fantasy_points_2022_week13"
[499] "fantasy_points_2022_week14" "fantasy_points_2022_week15"
[501] "fantasy_points_2022_week16" "fantasy_points_2022_week17"
[503] "fantasy_points_2022_week18" "fantasy_points_2022_week19"
[505] "fantasy_points_2022_week11" "fantasy_points_2022_week20"
[507] "fantasy_points_2022_week21" "fantasy_points_2022_week22"
[509] "fantasy_points_2023_week1"  "fantasy_points_2023_week4"
[511] "fantasy_points_2023_week7"  "fantasy_points_2023_week11"
[513] "fantasy_points_2023_week14" "fantasy_points_2023_week16"
```

```
[515] "fantasy_points_2023_week13" "fantasy_points_2023_week15"
[517] "fantasy_points_2023_week17" "fantasy_points_2023_week19"
[519] "fantasy_points_2023_week2"  "fantasy_points_2023_week3"
[521] "fantasy_points_2023_week5"  "fantasy_points_2023_week6"
[523] "fantasy_points_2023_week8"  "fantasy_points_2023_week12"
[525] "fantasy_points_2023_week18" "fantasy_points_2023_week10"
[527] "fantasy_points_2023_week21" "fantasy_points_2023_week22"
[529] "fantasy_points_2023_week9"  "fantasy_points_2023_week20"
```

3.18 Transform Data from Wide to Long

Original data:

```
dataWide <- nfl_actualStats_offense_weekly %>%
  select(player_id, player_display_name, season, week, recent_team, opponent_team)

dim(dataWide)
```

```
[1] 129739      6
```

```
names(dataWide)
```

```
[1] "player_id"          "player_display_name" "season"
[4] "week"               "recent_team"         "opponent_team"
```

Data in long form, transformed from wide form using `tidyverse`:

```
dataLong <- dataWide %>%
  pivot_longer(
    cols = c(recent_team, opponent_team),
    names_to = "role",
    values_to = "team")

dim(dataLong)
```

```
[1] 259478      6
```

```
names(dataLong)
```

```
[1] "player_id"          "player_display_name" "season"
[4] "week"               "role"                "team"
```

3.19 Calculations

3.19.1 Historical Actual Player Statistics

In addition to week-by-week actual player statistics, we can also compute historical actual player statistics as a function of different timeframes, including season-by-season and career statistics.

3.19.1.1 Career Statistics

First, we can compute the players' career statistics using the `calculate_player_stats()`, `calculate_player_stats_def()`, and `calculate_player_stats_kicking()` functions from the `nflfastR` package for offensive players, defensive players, and kickers, respectively.

> **Note 3: Calculating players' career statistics**
>
> Note: the following code takes a while to run.

```
nfl_actualStats_offense_career <- nflfastR::calculate_player_stats(
  nfl_pbp,
  weekly = FALSE)

nfl_actualStats_defense_career <- nflfastR::calculate_player_stats_def(
  nfl_pbp,
  weekly = FALSE)

nfl_actualStats_kicking_career <- nflfastR::calculate_player_stats_kicking(
  nfl_pbp,
  weekly = FALSE)
```

3.19.1.2 Season-by-Season Statistics

Second, we can compute the players' season-by-season statistics.

```
seasons <- unique(nfl_pbp$season)

nfl_pbp_seasonalList <- list()
nfl_actualStats_offense_seasonalList <- list()
nfl_actualStats_defense_seasonalList <- list()
nfl_actualStats_kicking_seasonalList <- list()
```

> **Note 4: Calculating players' season-by-season statistics**
>
> Note: the following code takes a while to run.

```
pb <- txtProgressBar(
  min = 0,
  max = length(seasons),
  style = 3)

for(i in 1:length(seasons)){
  # Subset play-by-play data by season
  nfl_pbp_seasonalList[[i]] <- nfl_pbp %>%
    filter(season == seasons[i])

  # Compute actual statistics by season
  nfl_actualStats_offense_seasonalList[[i]] <-
    nflfastR::calculate_player_stats(
      nfl_pbp_seasonalList[[i]],
      weekly = FALSE)

  nfl_actualStats_defense_seasonalList[[i]] <-
    nflfastR::calculate_player_stats_def(
      nfl_pbp_seasonalList[[i]],
      weekly = FALSE)

  nfl_actualStats_kicking_seasonalList[[i]] <-
    nflfastR::calculate_player_stats_kicking(
      nfl_pbp_seasonalList[[i]],
      weekly = FALSE)

  nfl_actualStats_offense_seasonalList[[i]]$season <- seasons[i]
  nfl_actualStats_defense_seasonalList[[i]]$season <- seasons[i]
  nfl_actualStats_kicking_seasonalList[[i]]$season <- seasons[i]

  print(
    paste("Completed computing projections for season: ", seasons[i], sep = ""))
```

```
  # Update the progress bar
  setTxtProgressBar(pb, i)
}

# Close the progress bar
close(pb)

nfl_actualStats_offense_seasonal <- nfl_actualStats_offense_seasonalList %>%
  bind_rows()
nfl_actualStats_defense_seasonal <- nfl_actualStats_defense_seasonalList %>%
  bind_rows()
nfl_actualStats_kicking_seasonal <- nfl_actualStats_kicking_seasonalList %>%
  bind_rows()
```

3.19.1.3 Week-by-Week Statistics

We already load players' week-by-week statistics above. Nevertheless, we could compute players' weekly statistics from the play-by-play data using the following syntax:

```
nfl_actualStats_offense_weekly <- nflfastR::calculate_player_stats(
  nfl_pbp,
  weekly = TRUE)

nfl_actualStats_defense_weekly <- nflfastR::calculate_player_stats_def(
  nfl_pbp,
  weekly = TRUE)

nfl_actualStats_kicking_weekly <- nflfastR::calculate_player_stats_kicking(
  nfl_pbp,
  weekly = TRUE)
```

3.19.2 Historical Actual Fantasy Points

Specify scoring settings:

3.19.2.1 Weekly

3.19.2.2 Seasonal

3.19.2.3 Career

3.19.3 Player Age

```
# Reshape from wide to long format
nfl_actualStats_offense_weekly_long <- nfl_actualStats_offense_weekly %>%
  pivot_longer(
    cols = c(recent_team, opponent_team),
    names_to = "role",
    values_to = "team")

# Perform separate inner join operations for the home_team and away_team
nfl_actualStats_offense_weekly_home <- inner_join(
  nfl_actualStats_offense_weekly_long,
  nfl_schedules,
  by = c("season","week","team" = "home_team")) %>%
  mutate(home_away = "home_team")

nfl_actualStats_offense_weekly_away <- inner_join(
  nfl_actualStats_offense_weekly_long,
  nfl_schedules,
  by = c("season","week","team" = "away_team")) %>%
  mutate(home_away = "away_team")

nfl_actualStats_defense_weekly_home <- inner_join(
  nfl_actualStats_defense_weekly,
  nfl_schedules,
  by = c("season","week","team" = "home_team")) %>%
  mutate(home_away = "home_team")

nfl_actualStats_defense_weekly_away <- inner_join(
  nfl_actualStats_defense_weekly,
  nfl_schedules,
  by = c("season","week","team" = "away_team")) %>%
  mutate(home_away = "away_team")

nfl_actualStats_kicking_weekly_home <- inner_join(
  nfl_actualStats_kicking_weekly,
  nfl_schedules,
  by = c("season","week","team" = "home_team")) %>%
```

```
  mutate(home_away = "home_team")

nfl_actualStats_kicking_weekly_away <- inner_join(
  nfl_actualStats_kicking_weekly,
  nfl_schedules,
  by = c("season","week","team" = "away_team")) %>%
  mutate(home_away = "away_team")

# Combine the results of the join operations
nfl_actualStats_offense_weekly_schedules_long <- bind_rows(
  nfl_actualStats_offense_weekly_home,
  nfl_actualStats_offense_weekly_away)

nfl_actualStats_defense_weekly_schedules_long <- bind_rows(
  nfl_actualStats_defense_weekly_home,
  nfl_actualStats_defense_weekly_away)

nfl_actualStats_kicking_weekly_schedules_long <- bind_rows(
  nfl_actualStats_kicking_weekly_home,
  nfl_actualStats_kicking_weekly_away)

# Reshape from long to wide
player_game_gameday_offense <- nfl_actualStats_offense_weekly_schedules_long %>%
  distinct(player_id, season, week, game_id, home_away, team, gameday) %>% #, .keep_all = TRUE
  pivot_wider(
    names_from = home_away,
    values_from = team)

player_game_gameday_defense <- nfl_actualStats_defense_weekly_schedules_long %>%
  distinct(player_id, season, week, game_id, home_away, team, gameday) %>% #, .keep_all = TRUE
  pivot_wider(
    names_from = home_away,
    values_from = team)

player_game_gameday_kicking <- nfl_actualStats_kicking_weekly_schedules_long %>%
  distinct(player_id, season, week, game_id, home_away, team, gameday) %>% #, .keep_all = TRUE
  pivot_wider(
    names_from = home_away,
    values_from = team)

# Merge player birthdate and the game date
player_game_birthdate_gameday_offense <- left_join(
  player_game_gameday_offense,
  unique(nfl_players[,c("gsis_id","birth_date")]),
```

```
  by = c("player_id" = "gsis_id")
)

player_game_birthdate_gameday_defense <- left_join(
  player_game_gameday_defense,
  unique(nfl_players[,c("gsis_id","birth_date")]),
  by = c("player_id" = "gsis_id")
)

player_game_birthdate_gameday_kicking <- left_join(
  player_game_gameday_kicking,
  unique(nfl_players[,c("gsis_id","birth_date")]),
  by = c("player_id" = "gsis_id")
)

player_game_birthdate_gameday_offense$birth_date <- ymd(player_game_birthdate_gameday_offense$birt
player_game_birthdate_gameday_offense$gameday <- ymd(player_game_birthdate_gameday_offense$gameday

player_game_birthdate_gameday_defense$birth_date <- ymd(player_game_birthdate_gameday_defense$birt
player_game_birthdate_gameday_defense$gameday <- ymd(player_game_birthdate_gameday_defense$gameday

player_game_birthdate_gameday_kicking$birth_date <- ymd(player_game_birthdate_gameday_kicking$birt
player_game_birthdate_gameday_kicking$gameday <- ymd(player_game_birthdate_gameday_kicking$gameday

# Calculate player's age for a given week as the difference between their birthdate and the game d
player_game_birthdate_gameday_offense$age <- interval(
  start = player_game_birthdate_gameday_offense$birth_date,
  end = player_game_birthdate_gameday_offense$gameday
) %>%
  time_length(unit = "years")

player_game_birthdate_gameday_defense$age <- interval(
  start = player_game_birthdate_gameday_defense$birth_date,
  end = player_game_birthdate_gameday_defense$gameday
) %>%
  time_length(unit = "years")

player_game_birthdate_gameday_kicking$age <- interval(
  start = player_game_birthdate_gameday_kicking$birth_date,
  end = player_game_birthdate_gameday_kicking$gameday
) %>%
  time_length(unit = "years")

# Merge with player info
```

```
player_age_offense <-  left_join(
  player_game_birthdate_gameday_offense,
  nfl_players %>% select(-birth_date, -season),
  by = c("player_id" = "gsis_id"))

player_age_defense <-  left_join(
  player_game_birthdate_gameday_defense,
  nfl_players %>% select(-birth_date, -season),
  by = c("player_id" = "gsis_id"))

player_age_kicking <-  left_join(
  player_game_birthdate_gameday_kicking,
  nfl_players %>% select(-birth_date, -season),
  by = c("player_id" = "gsis_id"))

# Add game_id to weekly stats to facilitate merging
nfl_actualStats_game_offense_weekly <- nfl_actualStats_offense_weekly %>%
  left_join(
    player_age_offense[,c("season","week","player_id","game_id")],
    by = c("season","week","player_id"))

nfl_actualStats_game_defense_weekly <- nfl_actualStats_defense_weekly %>%
  left_join(
    player_age_offense[,c("season","week","player_id","game_id")],
    by = c("season","week","player_id"))

nfl_actualStats_game_kicking_weekly <- nfl_actualStats_kicking_weekly %>%
  left_join(
    player_age_offense[,c("season","week","player_id","game_id")],
    by = c("season","week","player_id"))

# Merge with player weekly stats
player_age_stats_offense <- left_join(
  player_age_offense %>% select(-position, -position_group),
  nfl_actualStats_game_offense_weekly,
  by = c(c("season","week","player_id","game_id")))

player_age_stats_defense <- left_join(
  player_age_defense %>% select(-position, -position_group),
  nfl_actualStats_game_defense_weekly,
  by = c(c("season","week","player_id","game_id")))

player_age_stats_kicking <- left_join(
  player_age_kicking %>% select(-position, -position_group),
```

```
  nfl_actualStats_game_kicking_weekly,
  by = c(c("season","week","player_id","game_id")))

player_age_stats_offense$years_of_experience <- as.integer(player_age_stats_offense$years_of_exper
player_age_stats_defense$years_of_experience <- as.integer(player_age_stats_defense$years_of_exper
player_age_stats_kicking$years_of_experience <- as.integer(player_age_stats_kicking$years_of_exper

# Merge player info with seasonal stats
player_seasonal_offense <- left_join(
  nfl_actualStats_offense_seasonal,
  nfl_players %>% select(-position, -position_group, -season),
  by = c("player_id" = "gsis_id")
)

player_seasonal_defense <- left_join(
  nfl_actualStats_defense_seasonal,
  nfl_players %>% select(-position, -position_group, -season),
  by = c("player_id" = "gsis_id")
)

player_seasonal_kicking <- left_join(
  nfl_actualStats_kicking_seasonal,
  nfl_players %>% select(-position, -position_group, -season),
  by = c("player_id" = "gsis_id")
)

# Calculate age
season_startdate <- nfl_schedules %>%
  group_by(season) %>%
  summarise(startdate = min(gameday, na.rm = TRUE))

player_seasonal_offense <- player_seasonal_offense %>%
  left_join(
    season_startdate,
    by = "season"
  )

player_seasonal_defense <- player_seasonal_defense %>%
  left_join(
    season_startdate,
    by = "season"
  )

player_seasonal_kicking <- player_seasonal_kicking %>%
```

```
  left_join(
    season_startdate,
    by = "season"
  )

player_seasonal_offense$age <- interval(
  start = player_seasonal_offense$birth_date,
  end = player_seasonal_offense$startdate
) %>%
  time_length(unit = "years")

player_seasonal_defense$age <- interval(
  start = player_seasonal_defense$birth_date,
  end = player_seasonal_defense$startdate
) %>%
  time_length(unit = "years")

player_seasonal_kicking$age <- interval(
  start = player_seasonal_kicking$birth_date,
  end = player_seasonal_kicking$startdate
) %>%
  time_length(unit = "years")
```

3.20 Plotting

3.20.1 Rushing Yards per Carry By Player Age

```
# Prepare Data
rushing_attempts <- nfl_pbp %>%
  dplyr::filter(
    season_type == "REG") %>%
    filter(
      rush == 1,
      rush_attempt == 1,
      qb_scramble == 0,
      qb_dropback == 0,
      !is.na(rushing_yards))

rb_yardsPerCarry <- rushing_attempts %>%
  group_by(rusher_id, season) %>%
```

```
  summarise(
    ypc = mean(rushing_yards, na.rm = TRUE),
    rush_attempts = n(),
    .groups = "drop") %>%
  ungroup() %>%
  left_join(
    nfl_players %>% select(-season),
    by = c("rusher_id" = "gsis_id")
  ) %>%
  filter(
    position_group == "RB",
    rush_attempts >= 50) %>%
  left_join(
    season_startdate,
    by = "season"
  )

rb_yardsPerCarry$age <- interval(
  start = rb_yardsPerCarry$birth_date,
  end = rb_yardsPerCarry$startdate
) %>%
  time_length(unit = "years")

# Create Plot
ggplot2::ggplot(
  data = rb_yardsPerCarry,
  ggplot2::aes(
    x = age,
    y = ypc)) +
  ggplot2::geom_point() +
  ggplot2::geom_smooth() +
  ggplot2::labs(
    x = "Rushing Back Age (years)",
    y = "Rushing Yards per Carry/season",
    title = "2023 NFL Rushing Yards Per Carry per Season by Player Age",
    subtitle = "(minimum 50 rushing attempts)"
  ) +
  ggplot2::theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

Warning: Removed 865 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 865 rows containing missing values or values outside the scale range
```

(`geom_point()`).

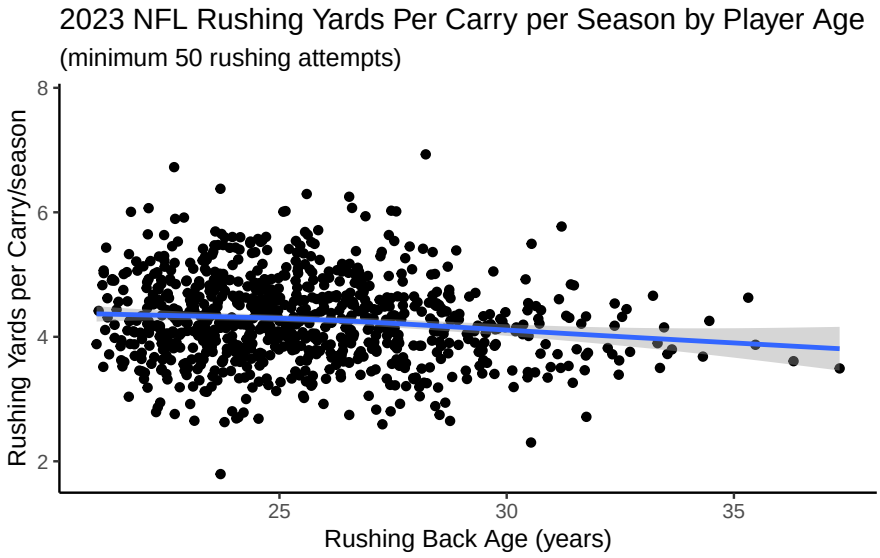

Figure 3.2 2023 NFL Rushing Yards Per Carry per Season by Player Age

```
# Subset Data
rb_seasonal <- player_seasonal_offense %>%
  filter(position_group == "RB")

# Create Plot
ggplot2::ggplot(
  data = rb_seasonal,
  ggplot2::aes(
    x = age,
    y = rushing_epa)) +
  ggplot2::geom_point() +
  ggplot2::geom_smooth() +
  ggplot2::labs(
    x = "Rushing Back Age (years)",
    y = "Rushing EPA/season",
    title = "2023 NFL Rushing Expected Points Added (EPA) per Season by Player Age"
  ) +
  ggplot2::theme_classic()
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
Warning: Removed 2415 rows containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 2415 rows containing missing values or values outside the scale range
(`geom_point()`).
```

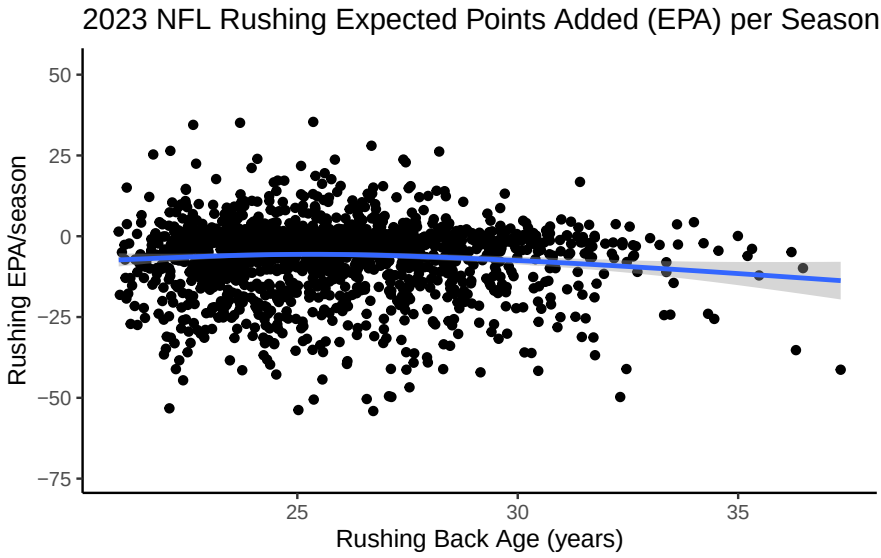

Figure 3.3 2023 NFL Rushing Expected Points Added (EPA) per Season by Player Age

3.20.2 Defensive and Offensive EPA per Play

Expected points added (EPA) per play by the team with possession.

```
pbp_regularSeason <- nfl_pbp %>%
  dplyr::filter(
    season == 2023,
    season_type == "REG") %>%
  dplyr::filter(!is.na(posteam) & (rush == 1 | pass == 1))

epa_offense <- pbp_regularSeason %>%
  dplyr::group_by(team = posteam) %>%
  dplyr::summarise(off_epa = mean(epa, na.rm = TRUE))

epa_defense <- pbp_regularSeason %>%
```

```
  dplyr::group_by(team = defteam) %>%
  dplyr::summarise(def_epa = mean(epa, na.rm = TRUE))

epa_combined <- epa_offense %>%
  dplyr::inner_join(epa_defense, by = "team")

ggplot2::ggplot(
  data = epa_combined,
  ggplot2::aes(
    x = off_epa,
    y = def_epa)) +
  nflplotR::geom_mean_lines(
    ggplot2::aes(
      x0 = off_epa ,
      y0 = def_epa)) +
  nflplotR::geom_nfl_logos(
    ggplot2::aes(
      team_abbr = team),
      width = 0.065,
      alpha = 0.7) +
  ggplot2::labs(
    x = "Offense EPA/play",
    y = "Defense EPA/play",
    title = "2023 NFL Offensive and Defensive EPA per Play"
  ) +
  ggplot2::theme_classic() +
  ggplot2::scale_y_reverse()
```

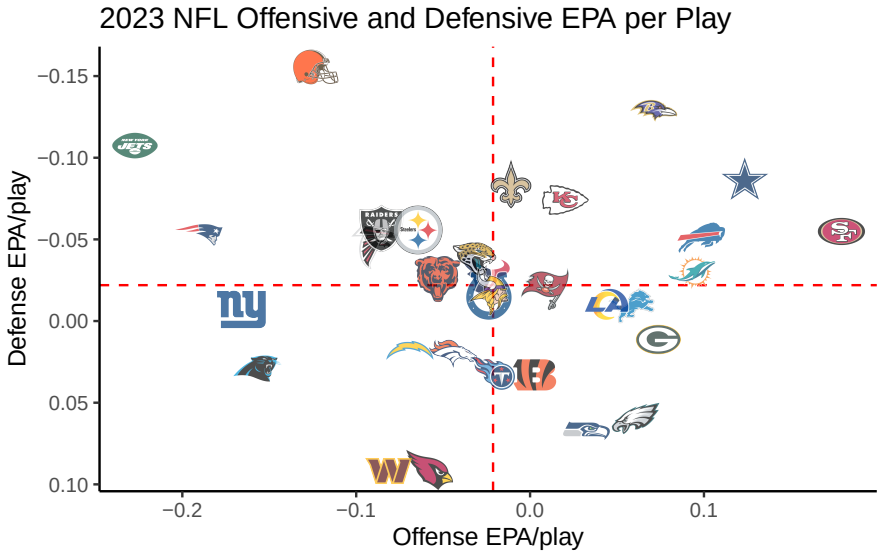

Figure 3.4 2023 NFL Offensive and Defensive EPA per Play

4

Player Evaluation

4.1 Getting Started

4.1.1 Load Packages

```
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

4.2 Overview

Evaluating players for fantasy football could be thought of as similar to the process of evaluating companies when picking stocks to buy. You want to evaluate and compare various assets so that you get the assets with the best value.

There are various domains of criteria we can consider when evaluating a football player's fantasy prospects. Potential domains to consider include:

- athletic profile

- historical performance
- health
- age and career stage
- situational factors
- matchups
- cognitive and motivational factors
- fantasy value

The discussion that follows is based on my and others' *impressions* of some of the characteristics that may be valuable to consider when evaluating players. However, the extent to which any factor is actually relevant for predicting future performance is an empirical question and should be evaluated empirically.

4.3 Athletic Profile

Factors related to a player's athletic profile include factors such as:

- body shape
  - height
  - weight
  - hand size
  - wing span (arm length)
- body function
  - agility
  - strength
  - speed
  - acceleration/explosiveness
  - jumping ability

In terms of body shape, we might consider a player's height, weight, hand size, and wing span (arm length). Height allows players to see over opponents and to reach balls higher in the air. Thus, greater height is particularly valuable for Quarterbacks and Wide Receivers. Heavier players are tougher to budge and to tackle. Greater weight is particularly valuable for Linemen, Fullbacks, and Tight Ends, but it can also be valuable—to a deree—for Quarterbacks, Running Backs, and Wide Receivers. Hand size and wing span is particularly valuable for people catching the ball; thus, a larger hand size and longer wing span are particularly valuable for Wide Receivers and Tight Ends.

In terms of body function, we can consider a player's agility, strength, speed, acceleration/explosiveness, and jumping ability. For Wide Receivers, speed, explosiveness, and jumping ability are particularly valuable. For Running Backs, agility, strength, speed, and explosiveness are particularly valuable.

Many aspects of a player's athletic profile are available from the National Football League (NFL) Combine, which is especially relevant for evaluating rookies. We demonstrate how to import data from the NFL Combine in Section Section 3.4.6. There are also calculators that integrate information about body shape and information from the NFL Combine to determine a player's relative athletic score (RAS) for their position: https://ras.football/ras-calculator/

4.4 Historical Performance

4.4.1 Overview

> "The best predictor of future behavior is past behavior." – Unknown

> "Past performance does not guarantee future results." – A common disclaimer about investments.

Factors relating to historical performance to consider could include:

- performance in college
  - draft position
- performance in the NFL
- efficiency
- consistency

It is important to consider a player's past performance. However, the extent to which historical performance may predict future performance may depend on many factors such as (a) the similarity of the prior situation to the current situation, (b) how long ago the prior situation was, and (c) the extent to which the player (or situation) has changed in the interim. For rookies, the player does not have prior seasons of performance in the NFL to draw upon. Thus, when evaluating rookies, it can be helpful to consider their performance in college or in their prior leagues. However, there are large differences between the situation in college and the situation in the NFL, so prior success in college may not portend future success in the NFL. An indicator that intends to be prognostic of future performance, and that accounts for past performance, is a player's draft position—that is, how early (or late) was a player selected in the NFL Draft. The earlier a player was selected in the NFL Draft, the greater likelihood that the player will perform well.

For players who have played in the NFL, past performance becomes more relevant because, presumably, the prior situation is more similar (than was their situation in college) to their current situation. Nevertheless, lots of things change from game to game and season to season: injuries, coaches, coaching strategies, teammates, etc. So just because a player performed well or poorly in a given game or season does not necessarily mean that they will perform similarly in subsequent games/seasons. Nevertheless, historical performance is one of the best indicators we have.

We demonstrate how to import historical player statistics in Section Section 3.4.12. We demonstrate how to calculate historical player statistics in Section Section 3.19.1. We demonstrate how to calculate historical fantasy points in Section Section 3.19.2.

4.4.2 Efficiency

In addition to how many fantasy points a player scores in terms of historical performance, we also care about efficiency and consistency. How efficient were they given the number of opportunities they had? If they were relatively more efficient, they will likely score more points than many of their peers when given more opportunities. If they were relativelly inefficient, their capacity to score fantasy points may be more dependent on touches/opportunities. Efficiency might be operationalized by indicators such as yards per passing attempt, yards per rushing attempt, yards per target, yards per reception, etc.

4.4.3 Consistency

In terms of consistency, how consistent was the player they from game to game and from season to season? For instance, we could examine the standard

deviations of players' fantasy points across games in a given season. However, the standard deviation tends to be upwardly biased as the mean increases. So, we can account for the player's mean fantasy points per game by dividing their game-to-game standard deviation of fantasy points (σ) by their mean fantasy points across games (μ). This is known as the coefficient of variation (CV):

$$CV = \frac{\sigma}{\mu}$$

Players with a lower standard deviation and a lower coefficient of variation (of fantasy points across games) are more consistent. In the example below, Player 2 might be preferable to Player 1 because Player 2 is more consistent; Player 1 is more "boom-or-bust." Despite showing a similar mean of fantasy points across weeks, Player 2 shows a smaller week-to-week standard deviation and coefficient of variation.

```
set.seed(1)

playerScoresByWeek <- data.frame(
  player1_scores = rnorm(17, mean = 20, sd = 7),
  player2_scores = rnorm(17, mean = 20, sd = 4),
  player3_scores = rnorm(17, mean = 10, sd = 4),
  player4_scores = rnorm(17, mean = 10, sd = 1)
)

consistencyData <- data.frame(t(playerScoresByWeek))

weekNames <- paste("week", 1:17, sep = "")

names(consistencyData) <- weekNames
row.names(consistencyData) <- NULL

consistencyData$mean <- rowMeans(consistencyData[,weekNames])
consistencyData$sd <- apply(consistencyData, 1, sd)
consistencyData$cv <- consistencyData$sd / consistencyData$mean

consistencyData$player <- c(1, 2, 3, 4)

consistencyData <- consistencyData %>%
  select(player, mean, sd, cv, week1:week17)

round(consistencyData, 2)
```

```
  player  mean   sd   cv week1 week2 week3 week4 week5 week6 week7 week8 week9
```

```
1    1 20.60 6.47 0.31 15.61 21.29 14.15 31.17 22.31 14.26 23.41 25.17 24.03
2    2 20.61 3.35 0.16 23.78 23.28 22.38 23.68 23.13 20.30 12.04 22.48 19.78
3    3 10.32 2.65 0.26  4.49  8.34  8.42  9.76 14.40 13.05  9.34  8.99 12.79
4    4 10.19 1.11 0.11  9.39 10.34  8.87 11.43 11.98  9.63  8.96 10.57  9.86
  week10 week11 week12 week13 week14 week15 week16 week17
1  17.86  30.58  22.73  15.65   4.50  27.87  19.69  19.89
2  19.38  14.12  18.09  21.67  25.43  19.59  21.55  19.78
3  12.23   7.24   7.17  11.46  13.07   9.55  13.52  11.59
4  12.40   9.96  10.69  10.03   9.26  10.19   8.20  11.47
```

4.5 Health

Health-related factors to consider include:

- current injury status
- injury history

It is also important to consider a player's past and current health status. In terms of a player's current health status, it is important to consider whether they are injured or are playing at less than 100% of their typical health. In terms of a player's prior health status, one can consider their injury history, including the frequency and severity of injuries and their prognosis.

We demonstrate how to import injury reports in Section Section 3.4.13.

4.6 Age and Career Stage

Age and career stage-related factors include:

- age
- experience
- touches

A player's age is relevant because of important age-related changes in a player's speed, ability to recover from injury, etc. A player's experience is relevant because players develop knowledge and skills with greater experience. A player's prior touches/usage is also relevant, because it speaks to how many hits a player may have taken. For players who take more hits, it may be more likely that their bodies "break down" sooner.

4.7 Situational Factors

Situational factors one could consider include:

- team quality
- role on team
- teammates
- opportunity and usage
  - snap count
  - touches/targets
  - red zone usage

Football is a team sport. A player is embedded within a broader team context; it is important to consider the strength of their team context insofar as it may support— or detract from—a player's performance. For instance, for a Quarterback, it is important to consider how strong the pass blocking is from the Offensive Line. Will they have enough time to throw the ball, or will they be constantly under pressure to be sacked? It is also important to consider the strength of the pass catchers—the Wide Receivers and Tight Ends. For a Running Back, it is important to consider how strong the run blocking is from the Offensive Line. For a Wide Receiver, it is important to consider how strong the pass blocking is, and how strong the Quarterback is.

It is also important to consider a player's role on the team. Is the player a starter or a backup? Related to this, it is important to consider the strength of one's teammates. For a given Running Back, if a teammate is better at running the ball, this may take away from how much the player sees the field. For a given Wide Receiver, if a teammate is better at catching the ball, this may take some targets away from the player. However, the team's top defensive back is often matched up against the team's top Wide Receiver. So, if the team's top Wide Receiver is matched up against a particularly strong Defensive Back, the second- and third-best Wide Receivers may more targets than usual.

It is also important to consider a player's opportunity and usage, which are influenced by many factors, including the skill of the player, the skill of their teammates, the role of the player on the team, the coaching style, the strategy of the opposing team, game scripts, etc. In terms of the player's opportunity and usage, how many snaps do they get? How many touches and/or targets do they receive? Being on the field for more snaps and receiving more touches and/or targets means that the player has more opportunities to score fantasy points. Are they targeted in the red zone? Red zone targets are more likely to lead to touchdown scoring opportunities, which are particularly valuable in fantasy football.

4.8 Matchups

Matchup-related factors to consider include:

- strength of schedule
- weekly matchup

Another aspect to consider is how challenging their matchup(s) and strength of schedule is. For a Quarterback, it is valuable to consider how strong the oppenent's passing defense is. For a Running Back, how strong is the running defense? For a Wide Receiver, how strong is the passing defense and the Defensive Back that is likely to be assigned to guard them?

4.9 Cognitive and Motivational Factors

Other factors to consider include cognitive and motivational factors. Some coaches refer to these as the "X Factor" or "the intangibles." However, just as any other construct in psychology, we can devise ways to operationalize them.

Cognitive and motivational factors one could consider include:

- reaction time
- knowledge and intelligence
- work ethic and mental toughness
- incentives
  - contract performance incentives
  - whether they are in a contract year

A player's knowledge, intelligence, and reaction time can help them gain an upper-hand even when they may not be the fastest or strongest. A player's work ethic and mental toughness may help them be resilient and persevere in the face of challenges. Contact-related incentives may lead a player to put forth greater effort. For instance, a contract may have a performance incentive that provides a player greater compensation if they achieve a particular performance milestone (e.g., receiving yards). Another potential incentive is if a player is in what is called their "contract year" (i.e., the last year of their current contract). If a player is in the last year of their current contract, they have an incentive to perform well so they can get re-signed to a new contract.

4.10 Fantasy Value

4.10.1 Sources From Which to Evaluate Fantasy Value

There are several sources that one can draw upon to evaluate a player's fantasy value:

- expert or aggregated rankings
- layperson rankings
  - players' Average Draft Position (ADP) in other league snake drafts
  - players' Average Auction Value (AAV) in other league auction drafts
- expert or aggregated projections

4.10.1.1 Expert Fantasy Rankings

Fantasy rankings (by so-called "experts") are provided by many sources. To reduce some of the bias due to a given source, some services aggregate projections across sources, consistent with a "wisdom of the crowd" approach. FantasyPros[1] aggregates fantasy rankings across sources. Fantasy Football Analytics[2] creates fantasy rankings from projections that are aggregated across sources (see the webapp here: https://apps.fantasyfootballanalytics.net).

4.10.1.2 Layperson Fantasy Rankings: ADP and AAV

Average Draft Position (ADP) and Average Auction Value (AAV), are based on league drafts, mostly composed of everyday people. ADP is based on snake drafts, whereas AAV is based on auction drafts. Thus, ADP and AAV are consistent with a "wisdom of the crowd" approach, and I refer to them as forms of rankings by laypeople. ADP data are provided by FantasyPros[3]. AAV data are also provided by FantasyPros[4].

4.10.1.3 Projections

Projections are provided by various sources. Projections (and rankings, for that matter) are a bit of a black box. It is often unclear how they were derived

[1] https://www.fantasypros.com/nfl/rankings/consensus-cheatsheets.php
[2] https://fantasyfootballanalytics.net
[3] https://www.fantasypros.com/nfl/adp/overall.php
[4] https://www.fantasypros.com/nfl/auction-values/calculator.php

by a particular source. That is, it is unclear how much of the projection was based on statistical analysis versus conjecture.

To reduce some of the bias due to a given source, some services aggregate projections across sources, consistent with a "wisdom of the crowd" approach. Projections that are aggregated across sources are provided by Fantasy Football Analytics[5] (see the webapp here: https://apps.fantasyfootballanalytics.net) and by FantasyPros[6].

4.10.1.4 Benefits of Using Projections Rather than Rankings

It is important to keep in mind that rankings, ADP, and AAV are specific to roster and scoring settings of a particular league. For instance, in point-per-reception (PPR) leagues, players who catch lots of passes (Wide Receivers, Tight Ends, and some Running Backs) are valued more highly. As another example, Quarterbacks are valued more highly in 2-Quarterback leagues. Thus, if using rankings, ADP, or AAV, it is important to find ones from leagues that mirror—as closely as possible—your league settings.

Projected statistics (e.g., projected passing touchdowns) are agnostic to league settings and can thus be used to generate league-specific fantasy projections and rankings. Thus, projected statisitics may be more useful than rankings because they can be used to generate rankings for your particular league settings. For instance, if you know how many touchdowns, yards, and interceptions a Quarterback is a projected to throw (in addition to any other relevant categories for the player, e.g., rushing yards and touchdowns), you can calculate how many fantasy points the Quarterback is expected to gain in your league (or in any league). Thus, you can calculate ranking from projections, but you cannot reverse engineer projections from rankings.

4.10.2 Indices to Evaluate Fantasy Value

Based on the sources above (rankings, ADP, AAV, and projections), we can derive multiple indices to evaluate fantasy value. There are many potential indices that can be worthwhile to consider, including a player's:

- dropoff
- value over replacement player (VORP)
- uncertainty

[5] https://fantasyfootballanalytics.net
[6] https://www.fantasypros.com/nfl/auction-values/calculator.php

4.10.2.1 Dropoff

A player's *dropoff* is the difference between (a) the player's projected points and (b) the projected points of the next-best player at that position.

4.10.2.2 Value Over Replacement Player

Because players from some positions (e.g., Quarterbacks) tend to score more points than players from other positions (e.g., Wide Receivers), it would be inadvisable to compare players across different positions based on projected points. In order to more fairly compare players across positions, we can consider a player's value over a typical replacement player at that position (shortened to "value over replacement player"). A player's *value over a replacement player* (VORP) is the difference between (a) a player's projected fantasy points and (b) the fantasy points that you would be expected to get from a typical bench player at that position. Thus, VORP provides an index of how much added value a player provides.

4.10.2.3 Uncertainty

A player's *uncertainty* is how much variability there is in projections or rankings for a given player across sources. For instance, consider a scenario where three experts provide ratings about two players, Player A and Player B. Player A is projected to score 300, 310, and 290 points by experts 1, 2, and 3, respectively. Player B is projected to score 400, 300, and 200 points by experts 1, 2, and 3, respectively. In this case, both players are (on average) projected to score the same number of points (300).

```
exampleData <- data.frame(
  player = c(rep("A", 3), rep("B", 3)),
  expert = c(1:3, 1:3),
  projectedPoints = c(300, 310, 290, 400, 300, 200)
)

playerA_mean <- mean(exampleData$projectedPoints[which(exampleData$player == "A")])
playerB_mean <- mean(exampleData$projectedPoints[which(exampleData$player == "B")])

playerA_mean
```

```
[1] 300
```

```
playerB_mean
```

```
[1] 300
```

However, the players differ considerably in their uncertainty (i.e., the source-to-source variability in their projections), as operationalized with the standard deviation and coefficient variation of projected points across sources for a given player.

```
playerA_sd <- sd(exampleData$projectedPoints[which(exampleData$player == "A")])
playerb_sd <- sd(exampleData$projectedPoints[which(exampleData$player == "B")])

playerA_cv <- playerA_mean / playerA_sd
playerB_cv <- playerB_mean / playerb_sd

playerA_sd
```

```
[1] 10
```

```
playerb_sd
```

```
[1] 100
```

```
playerA_cv
```

```
[1] 30
```

```
playerB_cv
```

```
[1] 3
```

Here is a depiction of a density plot of projected points for a player with a low, medium, and high uncertainty:

```
playerA <- rnorm(1000000, mean = 150, sd = 5)
playerB <- rnorm(1000000, mean = 150, sd = 15)
playerC <- rnorm(1000000, mean = 150, sd = 30)

mydata <- data.frame(playerA, playerB, playerC)

mydata_long <- mydata %>%
  pivot_longer(
    cols = everything(),
    names_to = "player",
    values_to = "points"
  ) %>%
```

```
  mutate(
    name = case_match(
      player,
      "playerA" ~ "Player A",
      "playerB" ~ "Player B",
      "playerC" ~ "Player C",
    )
  )

ggplot2::ggplot(
  data = mydata_long,
  ggplot2::aes(
    x = points,
    fill = name
  )
) +
  ggplot2::geom_density(alpha = .3) +
  ggplot2::labs(
    x = "Players' Projected Points",
    title = "Density Plot of Projected Points for Three Players"
  ) +
  ggplot2::theme_classic() +
  ggplot2::theme(legend.title = element_blank())
```

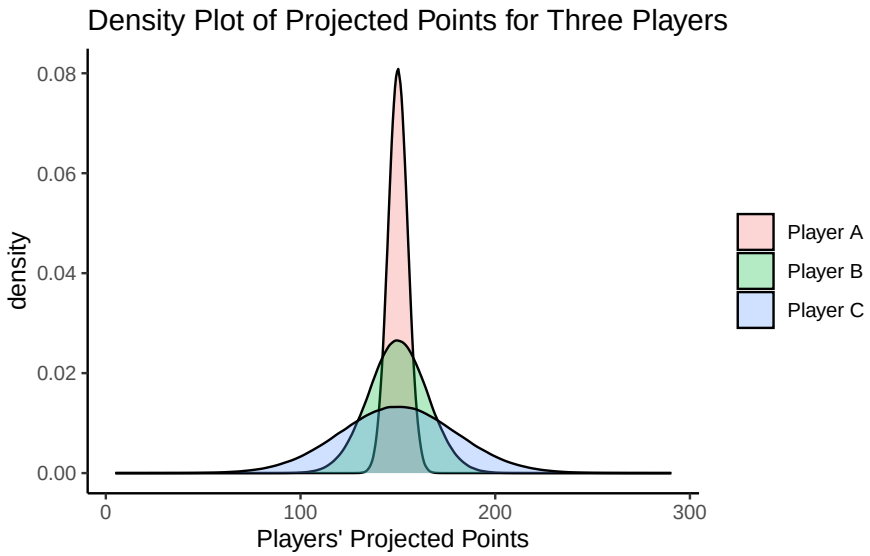

Figure 4.1 Density Plot of Projected Points for Three Players

Uncertainty is not necessarily a bad characteristic of a player's projected points. It just means we have less confidence about how the player may be expected to perform. Thus, players with greater uncertainty are risky and tend to have a higher upside (or ceiling) and a lower downside (or floor).

4.11 Putting it Altogether

After performing an evaluation of the relevant domain(s) for a given player, then one must integrate the evaluation information across domains to make a judgment about a player's overall value. When thinking about a player's value, it can be worth thinking of a player's upside and a player's downside. Player that are more consistent may show higher downside but a lower upside. Younger, less experienced players may show a higher upside but a lower downside.

The extent to which you prioritize a higher upside versus a higher downside may depend on many factors. For instance, when drafting players, you may prioritize drafting players with the highest downside (i.e., the safest players), whereas you may draft sleepers (i.e., players with higher upside) for your bench. When choosing which players to start in a given week, if you are predicted to beat a team handily, it may make sense to start the players with the highest downside. By contrast, if you are predicted to lose to a team by a good margin, it may make sense to start the players with the highest upside.

5

The Fantasy Draft

5.1 Getting Started

5.1.1 Load Packages

5.2 Types of Fantasy Drafts

There are several types of drafts in fantasy football. The most common types of drafts are snake drafts and auction drafts.

5.2.1 Snake Draft

In a snake draft, the participants (i.e., managers) are assigned a draft order. In the first round, the managers draft in that order. In the second round, the managers draft in reverse order. It continues to "snake" in this way, round after round, so that the person who has the first pick in a given round has the last pick in the next round, and whoever has the last pick in a given round has the first pick in the next round.

5.2.2 Auction Draft

In an auction draft, the managers are assigned a nomination order and there is a salary cap (e.g., $200). The first manager chooses which player to nominate. Then, the managers bid on that player like in an auction. In order to bid, the manager must raise the price by at least $1. If two managers want to obtain the same player, they may continue to raise the amount until one manager backs out and is no longer to bid by raising the price. The highest bidder wins (i.e., drafts) that player. Then, the second manager nominates a player, and the managers bid on that player. This process repeats until all teams have drafted their allotment of players.

In the auction draft, unlike the snake draft, all players are available to all teams. For instance, in the snake draft, the first 9 players drafted are unavailable to the 10th pick of the first round. So, if you have the 10th pick and want the top-ranked player, this player would not be available to you in the snake draft. However, in the auction draft, every player is available to every manager, so long as the manager is able and willing to bid enough.

5.3 Draft Strategy

5.3.1 Overview

There is no one "right" draft strategy. Sometimes it works best to "zig" when everyone else is "zagging". For instance, if you notice that everyone else is drafting Wide Receivers, this may mean that other managers are over-valuing Wide Receivers, and this could be a nice opportunity to draft a Running Back for good value.

In general, you will first want to generate the rankings you will use to select which players to prioritize. You may generate your rankings based one or more of the following:

- your evaluation of players[1]
- expert or aggregated rankings
- layperson rankings
  - players' Average Draft Position (ADP) in other league drafts (for snake drafts)
  - players' Average Auction Value (AAV) in other league drafts (for auction drafts[2])
- expert or aggregated projections
- indices derived from rankings and projections

Section Section 4.10.1 describes where to obtain aggregated rankings, aggregated projections, ADP, and AAV data.

An important concept in the draft is "dropoff", which is described in Section Section 4.10.2.1. *Dropoff* at a given position, is the difference—in terms of projected fantasy points—between (a) the best available player remaining at that position and (b) the second-best available player remaining at that position.

[1] player-evaluation.qmd
[2] sec-draftStrategyAuction

If there is a bigger dropoff at a given position, there may be greater value in drafting the top player from that position. For instance, consider the following scenario: "Quarterback A" is projected to score 325 points, and "Quarterback B" is projected to score 320 points. "Tight End A" is projected to score 230 points, and "Tight End B" is projected to score 150 points. In this example, there is a much greater dropoff for Tight Ends than there is for Quarterbacks. Thus, even though "Quarterback A" is projected to score more points than "Tight End A", "Tight End A" may be more valuable because there is still a good Quarterback available if someone else drafts "Quarterback A".

Another important concept is a player's value over a typical replacement player at that position (shortened to "value over replacement player"; VORP), which is described in Section Section 4.10.2.2.

Another important concept is a player's uncertainty, which is described in Section Section 4.10.2.3.

In both snake and auction draft formats, your goal is to draft the team whose weekly starting lineup scores the most points and thus the collection of players with the greatest VORP. For your starting lineup, it may make sense—especially with your earliest selections—when comparing two players with equivalent VORP, to prioritize players with higher consistency and lower uncertainty, because they may be considered "safer" with a higher floor. However, when drafting your bench players, it make make more sense to prioritize high risk, high reward players with greater uncertainty, because they may have a higher ceiling. The Spurs in the National Basketball Association (NBA) were well-reputed for excelling in this draft strategy. They frequently used their second-round picks to draft high-risk, high-reward players; sometimes, the secound round pick was a bust, but other times their second round picks including Manu Ginobili and Tony Parker greatly outperformed expectations and contributed to the team's strong success.

However, the draft strategies to achieve the "optimal lineup" differ between snake versus auction drafts.

5.3.2 Snake Draft

In general, your goal is to draft the team whose weekly starting lineup has the greatest VORP. Consequently, you are often looking to pick the player with the highest VORP at a given selection, while keeping in mind (a) the dropoff of players at other positions and (b) which players may be available at subsequent picks so that you do not sacrifice too much later value with a given selection. For instance, if a particular Quarterback has a slightly higher VORP than a particular Running Back, but the Quarterback is likely to be available at the manager's next pick but the Running Back is likely to be unavailable at their next pick, it might make more sense to draft the Running

Back.

5.3.3 Auction Draft

References

Motz, B. (2013). Fantasy football: A touchdown for undergraduate statistics education. *Proceedings of the Games, Learning, and Society Conference*, *9.0*, 222–228. https://doi.org/10.1184/R1/6686804.v1

Index