

Fantasy Football Analytics

Isaac T. Petersen

To our daughter, Maisie.

Table of contents

Preface	xi
How to Contribute	xi
Open Access	xi
License	xii
Citation	xii
About the Author	xiii
Accessibility	xiv
Acknowledgments	xiv
1 Introduction	1
1.1 About this Book	1
1.2 What is Fantasy Football?	1
1.3 Why Focus on Fantasy Football?	2
1.4 Educational Value	2
1.5 Learning Objectives	3
1.6 Disclosures	4
1.7 Disclaimer	4
2 Intro to Football and Fantasy	5
2.1 Football	5
2.1.1 The Objective	5
2.1.2 The Roster	5
2.1.3 The Field	8
2.1.4 The Gameplay	10
2.1.5 The Scoring	11
	iii

2.1.6	Glossary of Terms	11
2.2	Fantasy Football	13
2.2.1	Overview of Fantasy Football	13
2.2.2	The Fantasy League	14
2.2.3	The Roster of a Fantasy Team	14
2.2.4	Scoring	15
3	Getting Started with R for Data Analysis	19
3.1	Initial Setup	19
3.2	Installing Packages	20
3.3	Load Packages	20
3.4	Download Football Data	20
3.4.1	Players	20
3.4.2	Teams	20
3.4.3	Player Info	21
3.4.4	Rosters	21
3.4.5	Game Schedules	21
3.4.6	The Combine	21
3.4.7	Draft Picks	21
3.4.8	Depth Charts	22
3.4.9	Play-By-Play Data	22
3.4.10	4th Down Data	22
3.4.11	Participation	22
3.4.12	Historical Weekly Actual Player Statistics	23
3.4.13	Injuries	23
3.4.14	Snap Counts	23
3.4.15	ESPN QBR	24
3.4.16	NFL Next Gen Stats	24
3.4.17	Advanced Stats from PFR	25
3.4.18	Player Contracts	26
3.4.19	FTN Charting Data	26

Contents v

3.4.20 Fantasy Player IDs	27
3.4.21 FantasyPros Rankings	27
3.4.22 Expected Fantasy Points	27
3.5 Data Dictionary	28
3.6 Create a Data Frame	28
3.7 Variable Names	29
3.8 Logical Operators	30
3.8.1 Is Equal To: ==	30
3.8.2 Is Not Equal To: !=	30
3.8.3 Is Greater Than: >	30
3.8.4 Is Less Than: <	30
3.8.5 Is Greater Than or Equal To: >=	31
3.8.6 Is Less Than or Equal To: <=	31
3.8.7 Is In a Value of Another Vector: %in%	31
3.8.8 Is Not In a Value of Another Vector: !(%in%)	31
3.8.9 Is Missing: is.na()	31
3.8.10 Is Not Missing: !is.na()	32
3.8.11 And: &	32
3.8.12 Or: 	32
3.9 Subset	32
3.9.1 One Variable	33
3.9.2 Particular Rows of One Variable	33
3.9.3 Particular Columns (Variables)	33
3.9.4 Particular Rows	38
3.9.5 Particular Rows and Columns	40
3.10 View Data	41
3.10.1 All Data	41
3.10.2 First 6 Rows/Elements	41
3.11 Data Characteristics	42
3.11.1 Data Structure	42
3.11.2 Data Dimensions	43

3.11.3	Number of Elements	43
3.11.4	Number of Missing Elements	43
3.11.5	Number of Non-Missing Elements	43
3.12	Create New Variables	44
3.13	Recode Variables	44
3.14	Rename Variables	45
3.15	Convert the Types of Variables	45
3.16	Merging/Joins	48
3.16.1	Overview	48
3.16.2	Data Before Merging	49
3.16.3	Types of Joins	50
3.17	Transform Data from Long to Wide	58
3.18	Transform Data from Wide to Long	65
3.19	Calculations	66
3.19.1	Historical Actual Player Statistics	66
3.19.2	Historical Actual Fantasy Points	69
3.19.3	Player Age	69
3.20	Plotting	74
3.20.1	Rushing Yards per Carry By Player Age	74
3.20.2	Defensive and Offensive EPA per Play	77
4	Player Evaluation	79
4.1	Getting Started	79
4.1.1	Load Packages	79
4.2	Overview	79
4.3	Athletic Profile	80
4.4	Historical Performance	81
4.4.1	Overview	81
4.4.2	Efficiency	82
4.4.3	Consistency	82
4.5	Health	84

<i>Contents</i>	vii
4.6 Age and Career Stage	84
4.7 Situational Factors	84
4.8 Matchups	86
4.9 Cognitive and Motivational Factors	86
4.10 Fantasy Value	87
4.10.1 Sources From Which to Evaluate Fantasy Value	87
4.10.2 Indices to Evaluate Fantasy Value	88
4.11 Putting it Altogether	92
5 The Fantasy Draft	95
5.1 Getting Started	95
5.1.1 Load Packages	95
5.2 Types of Fantasy Drafts	95
5.2.1 Snake Draft	95
5.2.2 Auction Draft	95
5.2.3 Comparison	96
5.3 Draft Strategy	96
5.3.1 Overview	96
5.3.2 Snake Draft	98
5.3.3 Auction Draft	98
6 Research Methods	99
6.1 Getting Started	99
6.1.1 Load Packages	99
6.2 Sample vs Population	99
6.3 Research Designs	100
6.3.1 Experiment	100
6.3.2 Correlational/Observational Study	101
6.3.3 Case Study	103
6.3.4 Other Features of the Research Design	104
6.4 Research Design Validity	106

6.4.1	Internal Validity	106
6.4.2	External Validity	106
6.4.3	Tradeoffs Between Internal and External Validity	106
6.4.4	Conclusion Validity	107
6.5	Mediation vs Moderation	108
6.5.1	Mediation	109
6.5.2	Moderation (i.e., Interaction)	111
6.6	Levels of Measurement	112
6.6.1	Nominal	113
6.6.2	Ordinal	114
6.6.3	Interval	114
6.6.4	Ratio	115
6.7	Psychometrics	116
6.7.1	Measurement Reliability	116
6.7.2	Measurement Validity	117
6.7.3	Reliability vs Validity	120
7	Basic Statistics	121
7.1	Getting Started	121
7.1.1	Load Packages	121
7.2	Descriptive Statistics	121
7.2.1	Center	121
7.2.2	Spread	122
7.2.3	Shape	123
7.2.4	Combination	123
7.3	Scores and Scales	124
7.3.1	Raw Scores	124
7.3.2	z Scores	124
7.4	Inferential Statistics	128
7.4.1	Null Hypothesis Significance Testing	128
7.4.2	Practical Significance	140

<i>Contents</i>	ix
7.5 Statistical Decision Tree	144
7.6 Statistical Tests	145
7.6.1 <i>t</i> -Test	145
7.6.2 Analysis of Variance	147
7.6.3 Correlation	148
7.6.4 (Multiple) Regression	148
7.6.5 Chi-Square Test	148
7.6.6 Formulating Statistical Tests in Terms of Partitioned Variance	149
7.6.7 Critical Value	149
7.6.8 Statistical Power	156
8 Correlation Analysis	181
8.1 Getting Started	181
8.1.1 Load Packages	181
8.2 Overview of Correlation	181
8.3 The Correlation Coefficient (<i>r</i>)	181
8.4 Examples	194
8.4.1 Covariance	194
8.4.2 Pearson Correlation	194
8.4.3 Spearman Correlation	194
8.4.4 Nonlinear Correlation	194
8.4.5 Correlation Matrix	194
8.4.6 Correlogram	194
8.5 Session Info	194
9 Multiple Regression	195
9.1 Getting Started	195
9.1.1 Load Packages	195
9.2 Overview of Multiple Regression	195
9.3 Components	196
9.4 Coefficient of Determination (R^2)	197

9.5 Covariates	197
9.6 Multicollinearity	198

Preface

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

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, please consider one of the following:

- open an issue¹ or create a pull request² on the book's GitHub repository³.
- buy me a coffee⁴—Support me in developing this (free!) resource for fantasy football analytics... Even a cup of coffee helps me stay awake!

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.

Open Access

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

¹<https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook/issues>

²<https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook/pulls>

³<https://github.com/isaactpetersen/Fantasy-Football-Analytics-Textbook>

⁴<https://www.buymeacoffee.com/isaactpetersen>

License



Figure 1 Creative Commons License

The online version of this book is licensed under the Creative Commons Attribution License⁵. In short, you can use my work as long as you cite it.

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}
}
```

⁵<https://creativecommons.org/licenses/by/4.0/>

About the Author

I am an Assistant Professor in the Department of Psychological and Brain Sciences at the University of Iowa. I am a licensed psychologist with expertise in child clinical psychology. Why am I writing about fantasy football and data analysis? Because fantasy football involves the intersection of two things I love: sports and statistics.

Through my training, I have learned the value of statistics for answering important questions that I find interesting. In graduate training, I came to the realization that statistics are relevant not only for psychology and science, but also for domains that I enjoy as hobbies, including sports and fantasy sports. I have played in a longstanding fantasy football league for over 20 years (since my junior year of high school) with old friends from high school. I wanted to apply what I was learning about statistics to help others improve their performance in fantasy football and to help people—including those who might not otherwise be interested—to learn statistics. So I began blogging online about the value of applying statistics to improve decision making in fantasy football. Apparently, many people were interested in learning statistics when they could apply them to a domain that they find interesting like fantasy football. My blog eventually became FantasyFootballAnalytics.net⁶, a website that uses advanced statistics to help people win their fantasy football leagues.

In terms of my R and statistics background, I have published many peer-reviewed publications⁷ that employ advanced statistical methods, have published a book on psychological assessment⁸ (Petersen, 2024b, 2024c) that includes applied examples in R, and have published the `petersenlab` R package⁹ (Petersen, 2024a) on the Comprehensive R Archive Network (CRAN). I am also a co-author of the `ffanalytics` R package¹⁰ that provides free utilities for downloading fantasy football projections and additional fantasy-relevant data, and for calculating projected points given your league settings.

⁶<http://fantasyfootballanalytics.net>

⁷<https://developmental-psychopathology.lab.uiowa.edu/publications>

⁸<https://www.routledge.com/9781032413068>

⁹<https://cran.r-project.org/web/packages/petersenlab/index.html>

¹⁰<https://github.com/FantasyFootballAnalytics/ffanalytics>

Accessibility

I strive to follow principles of accessibility¹¹ (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.

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.net¹², 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.

¹¹<https://bookdown.org/yihui/rmarkdown-cookbook/html-accessibility.html>

¹²<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 inference¹, factor analysis², cluster analysis³, principal component analysis⁴, machine learning⁵, cognitive biases⁶, modern portfolio theory⁷, data visualization⁸, simulation⁹, 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 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.

²One 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 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 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 defensed occurs when a defensive player knocks down the ball in the air so that the intended 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 successfully 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
- receiving 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 defensed—when a defensive player knocks down the ball in the air so that the intended 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 sports³, of whom around 29 million play fantasy football.⁴ As noted in the Introduction⁵,

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:

⁶Fantasy leagues are also available for baseball⁷, basketball⁸, 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
2-point conversion	
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", "ffanalytics", "nflreadr", "nflfastR", "nfl4th", "nflplotR",  
  "gsisdecoder", "progressr", "lubridate", "tidyverse", "psych"))
```

i 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")
library("nflreadr")
library("nflfastR")
library("nfl4th")
library("nflplotR")
library("progressr")
library("lubridate")
library("tidyverse")
```

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_charts(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 dataset. 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 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 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 TRUE TRUE TRUE
```

3.8.9 Is Missing: is.na()

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

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

3.8.10 Is Not Missing: !is.na()

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

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

3.8.11 And: &

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

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE 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 Uloozi"
```

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 Uloozi"
```

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 Uloozi 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 Uloozi 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 Uloozi 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 Uloozi 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 Ullooz   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 Ullooz 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 Uloozi     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)
```

```
# 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>
1 RET    'Omar Ellison  'Omar       Ellison   ELL711~ 00-000~ <NA>  <NA>
2 ACT    A'Shawn Robinson A'Shawn  Robinson  R0B367~ 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 ...
- 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 Uloozi		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)
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 Uloozi	LB	37		<NA>	old	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 Uloozi 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 Uloozi	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 Uloozi	LB	37		<NA>	old	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 .

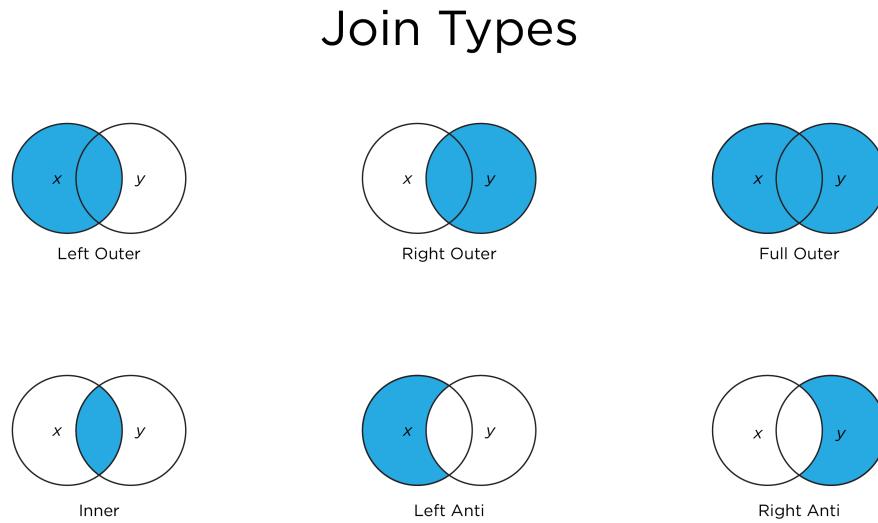


Figure 3.1 Types of merges/joins

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 Uloozi	LB	37		old	12	37
13	13		<NA>	<NA>	NA	<NA>	<NA>	NA
14	14		<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")
```

	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	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

Depending on the analysis, it may be important to restructure the data to be in long or wide form. When the data are in wide form, each player has only one row. When the data are in long form, each player has multiple rows—e.g., a row for each game. The data structure is called wide or long form because a dataset in wide form has more columns and fewer rows (i.e., it appears wider and shorter), whereas a dataset in long form has more rows and fewer columns (i.e., it appears narrower and taller).

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`. This is an example of long form, because each player has multiple rows.

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 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

Conversely, we can also restructure 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.

i 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$birth_date <- ymd(player_game_birthdate_gameday_offense$birth_date)
player_game_birthdate_gameday_offense$gameday <- ymd(player_game_birthdate_gameday_offense$gameday)

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

player_game_birthdate_gameday_kicking$birth_date <- ymd(player_game_birthdate_gameday_kicking$birth_date)
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 day
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_experience)
player_age_stats_defense$years_of_experience <- as.integer(player_age_stats_defense$years_of_experience)
player_age_stats_kicking$years_of_experience <- as.integer(player_age_stats_kicking$years_of_experience)

# 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()
```

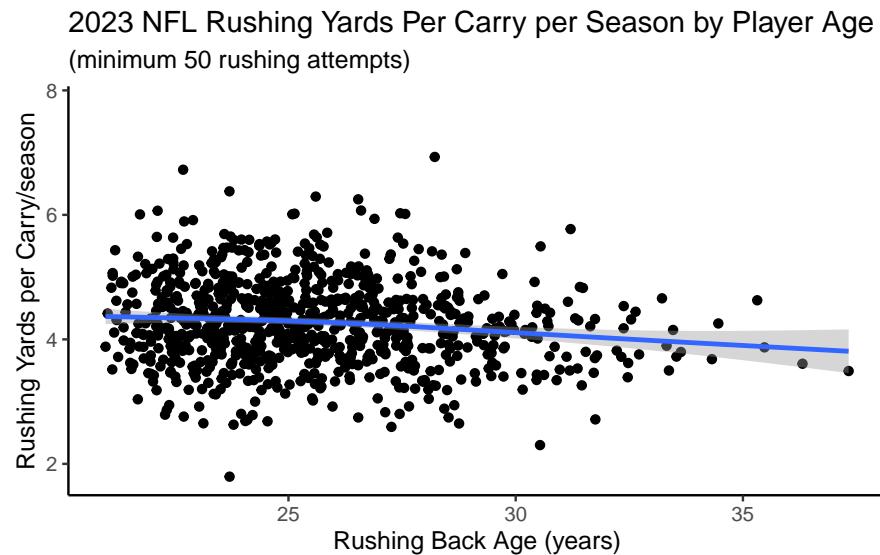


Figure 3.2 2023 NFL Rushing Yards Per Carry per Season by Player Age

```
# 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()
```

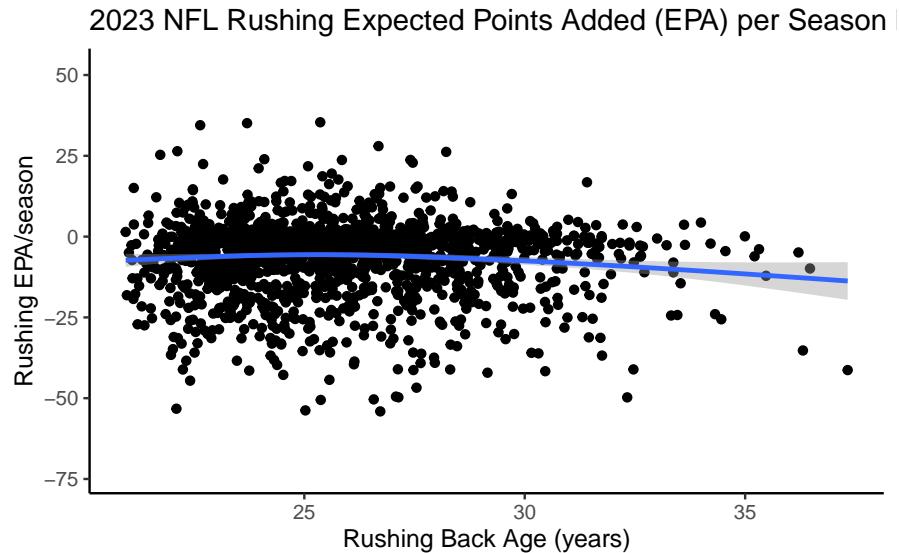


Figure 3.3 2023 NFL Rushing Expected Points Added (EPA) per Season by Player Age

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()

```

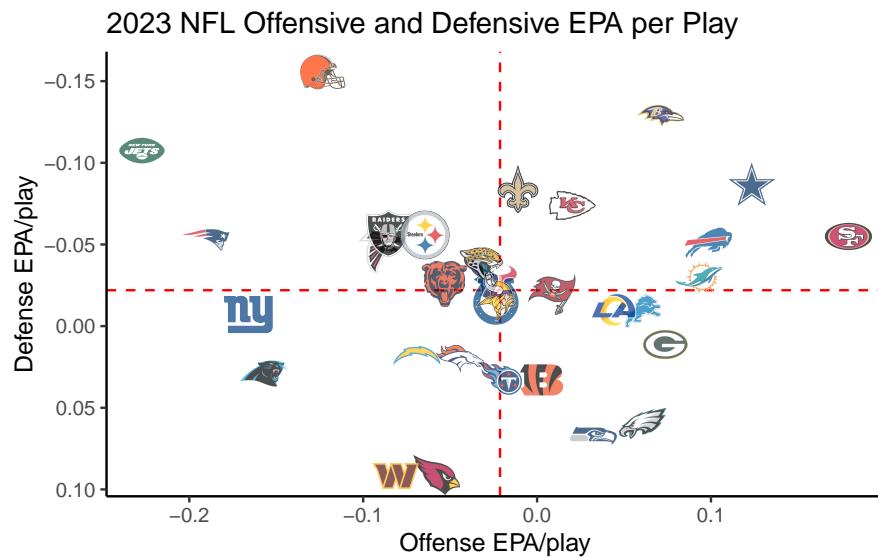


Figure 3.4 2023 NFL Offensive and Defensive EPA per Play

4

Player Evaluation

4.1 Getting Started

4.1.1 Load Packages

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

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 degree—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 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 3.4.12. We demonstrate how to calculate historical player statistics in Section 3.19.1. We demonstrate how to calculate historical fantasy points in 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 relatively 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), which is provided in Equation 4.1.

$$CV = \frac{\sigma}{\mu} \quad (4.1)$$

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	week10	week11	week12	week13	week14	week15	week16	week17
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	17.86	30.58	22.73	15.65	4.50	27.87	19.69	19.89
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	19.38	14.12	18.09	21.67	25.43	19.59	21.55	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	12.23	7.24	7.17	11.46	13.07	9.55	13.52	11.59
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	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 [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 opponent'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. Insofar as they are observable, they are measurable.

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. FantasyPros¹ aggregates fantasy rankings across sources. Fantasy Football Analytics² 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 FantasyPros³. AAV data are also provided by FantasyPros⁴.

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 Analytics⁵ (see the webapp here: <https://apps.fantasyfootballanalytics.net>) and by FantasyPros⁶.

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 statistics 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 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://apps.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_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_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
```

```
[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())
```

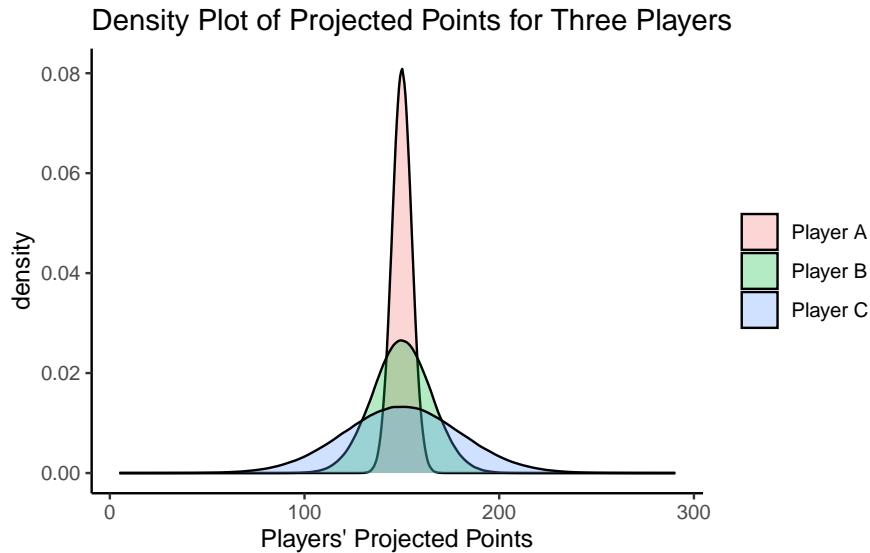


Figure 4.1 Density Plot of Projected Points for Three Players

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.

5.2.3 Comparison

Snake drafts are more common than auction drafts. Snake drafts tend to be quicker than auction drafts. However, auction drafts are more fair than snake drafts. In an auction draft, unlike a snake draft, all players are available to all teams. For instance, in a 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 players¹
- 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 drafts²)
- expert or aggregated projections
- indices derived from rankings and projections

Section 4.10.1 describes where to obtain [aggregated rankings](#), [aggregated projections](#), [ADP](#), and [AAV](#) data.

¹[player-evaluation.qmd](#)

²[sec-draftStrategyAuction](#)

An important concept in the draft is “**dropoff**”, which is described in 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. 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 4.10.2.2.

Another important concept is a player’s **uncertainty**, which is described in 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 players for your bench, it makes more sense to prioritize high-risk, high reward players with greater **uncertainty**, because they may have a higher ceiling. Players with a higher ceiling have a potential to be “sleepers”—players who are valued low (i.e., with a high **ADP** or low **AAV**) and who outperform their valuation. Note that, although players with greater **uncertainty** are high-risk, high-reward players, selecting this kind of a player for your bench (i.e., in a late round or for a small cost) is a *lower* risk selection, because you have less to lose with later/lower-cost picks. That is, even though the *player* is higher risk, selecting a higher risk player for your bench is a lower risk *decision*.

The Spurs in the National Basketball Association (NBA) were well-reputed for excelling in this draft strategy³ (archived at <https://perma.cc/X7NW-WZC6>). They frequently used their second-round picks to draft high-risk, high-reward players. Sometimes, the second round pick was a bust, but they have little to lose with a failed second round pick. Other times, their second round picks—including Willie Anderson, DeJuan Blair, Goran Dragic, Luis Scola, and Manu Ginóbili—greatly outperformed expectations. Thanks, in

³<https://harvardsportsanalysis.org/2013/11/beating-the-nba-draft-does-any-team-outperform-expectations/>

part, to this draft strategy, the team showed strong extended success for nearly three decades from 1989 through the late-2010s.

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

According to an analysis⁴ by the Harvard Sports Analysis Collective (archived at <https://perma.cc/P7RX-92UU>), the majority of the manager’s salary cap should be spent on the starting lineup, and you should spend less on bench players. This is known as the “stars and scrubs” draft strategy. Based on the analysis, the author recommended applying a 10% premium to the top players and a 10% discount to the lower-tiered players. The idea behind the approach is that a player on your bench does not contribute to the team’s points and, thus, most players drafted to your bench do not contribute much to the team’s points throughout the season. That said, bench players can be important in the case of a starter’s injury or under-performance. So, it is recommended to draft starters with lower **uncertainty** who are safer. In contrast to your starting lineup, you may look to draft players on your bench who have greater **uncertainty** for their high reward potential in a low-risk selection given the lower price.

An alternative to the “stars and scrubs” approach is to wait to draft more “high-value” players after other managers have over-paid for players.

⁴<https://harvardsportsanalysis.wordpress.com/wp-content/uploads/2012/04/fantasyfootballdraftanalysis1.pdf>

6

Research Methods

6.1 Getting Started

6.1.1 Load Packages

6.2 Sample vs Population

In research, it is important to distinguish between the sample and the target population. The target *population* is who you want your study's findings to generalize to. For instance, if we want our findings to lead to inferences we can draw regarding all current NFL players, then NFL players are our target population. However, despite our best efforts to recruit all NFL players into our study, we may not succeed in doing that. The participants (i.e., people or players) who we successfully recruit to be in our study represent our *sample*.

It is rare for the sample to include all people who are in the target population. It can be costly to recruit large samples, and many potential participants may decline to participate for a variety of reasons (insufficient time, lack of interest in the study, distrust of scientists, etc.). Thus, our goals are (a) to recruit as many people from the population as possible and (b) for the sample to be as *representative* of the population as possible.

For increasing the representativeness of the sample (with respect to the population), we might conduct a *random sample*, in which each person in the population (i.e., each NFL player) has equal likelihood of being selected. For instance, we might randomly select 250 players to recruit to the study. True random samples, though strong in aspiration, are difficult and costly to achieve. In reality, many researchers conduct convenience sampling. A convenience sample is recruited because it is convenient (i.e., less costly and time-consuming).

For instance, many studies examine college students—in part, because they are easy to recruit. If our target population is NFL players but we are unable to recruit NFL players into our study, we could easily recruit a large sample

of college students. Although the convenience sample may afford a very large sample, the college student sample may not be representative of the target population (NFL players). Thus, the findings in our study may not *generalize* to NFL players—that is, what we learn in college students may not apply in the same way among NFL players. For instance, if we learn that consumption of sports drinks (compared to drinking only water) improves running speed among college students, that may not be the case among NFL players.

6.3 Research Designs

There are three broad types of research designs:

- experiment
- correlational/observational study
- case study

6.3.1 Experiment

In an *experiment*, there are one or more things (i.e., variables) that we manipulate to see how the manipulation influences the process of interest. The variable that we manipulate is the *independent variable*. By contrast, the *dependent variable* is the variable that we evaluate to determine whether it was influenced by the manipulation (i.e., by the independent variable). Besides the independent and dependent variables, the researcher attempts to hold everything else constant through processes including standardization and random assignment. *Standardization* involves using the same procedures to assess each participant, so that scores can be fairly compared across participants (and groups). Random assignment involves randomly assigning participants to conditions of the independent variable, so the people in each condition are comparable and do not differ systematically.

For instance, we may be interested to evaluate whether players perform better (e.g., run faster) if they drink a sports drink compared when they drink only water. Our hypothesis might be that players will be expected to perform better when they drink a sports drink (compared to when they drink only water). To this this research question and hypothesis, we might conduct an experiment by randomly assigning some players during practice to receive a sports drink and some players to receive only water. In this case, our independent variable is whether the player receives a sports drink. Our dependent variable might be their 40-yard dash time during practice.

6.3.2 Correlational/Observational Study

In a correlational (aka observational) study, we do not manipulate a variable to see how the manipulation influences another variable. Instead, we examine how two variables, a predictor and an outcome variable, are associated. The hypothesized cause is called the predictor variable. The hypothesized effect is called the outcome variable. In this way, the predictor variable is similar to the independent variable, and the outcome variable is similar to the dependent variable. However, unlike the independent and dependent variables in an experiment, the predictor and outcome variables in a correlational study are not manipulated.

For instance, to use a correlational study to test the possibility that players who drink sports drinks perform better than players who drink only water, we could examine whether the players who drink sports drinks during a game score more fantasy points than players who drink only water during the game. In this case, our predictor variable is whether the players drink sports drinks during a game. Our outcome variable is the number of fantasy points the player scored.

6.3.2.1 Correlation Does Not Imply Causation

As the maxim goes, “correlation does not imply causation”—just because two variables are associated does not necessarily mean that they are causally related.

Just because X is associated with Y does not mean that X causes Y . Consider that you find an association between variables X and Y .

There are several reasons why you might observe an association between X and Y :

- X causes Y
- Y causes X
- a third variable (i.e., confound), Z , influences both X and Y
- the association between X and Y is spurious

For instance, one possibility is that the association we observed reflects our hypothesis that X causes Y , as depicted in Figure 6.1. That is, consumption of more sports drink may improve players’ performance.

However, a second possibility is that the association reflects the opposite direction of effect, where Y actually causes X , as depicted in Figure 6.2. For instance, greater performance may lead players to drink more sports drink (rather than the reverse).

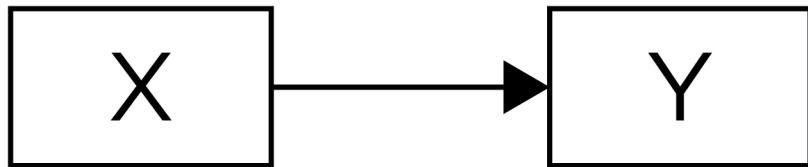


Figure 6.1 Hypothesized Causal Effect Based on an Observed Association Between X and Y , Such That X Causes Y .

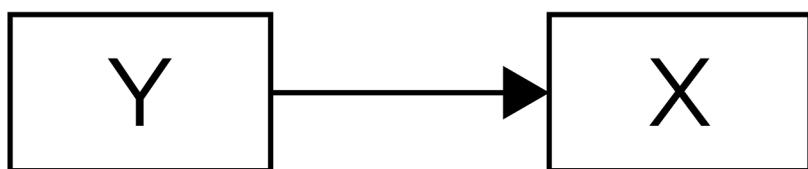


Figure 6.2 Reverse (Opposite) Direction of Effect From the Hypothesized Effect, Where Y Causes X .

A third possibility is that the association could reflect the influence of a third variable. If a third variable is a common cause of each and accounts for their association, it is a *confound*. An observed association between X and Y could reflect a confound—i.e., a cause (Z) that influences both X and Y , which explains why X and Y are correlated even though they are not causally related. A third variable confound that is a common cause of both X and Y is depicted in Figure 6.3. For instance, it may not be that sport drink consumption per se influences player performance; rather, it may be that players who are more intelligent or have more financial resources tend to drink more sports drinks and also tend to perform better. In this case, intelligence or financial resources may be a confound that influences both sports drink consumption and player performance, but sports drink consumptions—though correlated with player performance—does not influence player performance.

For another example, consider that ice cream sales are associated with shark attacks. It is unlikely that more people eating ice creams leads to shark attacks. There is a likely a third variable—heat waves—that is a confound because it influences both ice cream sales and shark attacks and explains their association.

Lastly, the association might be spurious. It might just reflect random variation (i.e., chance), and that when tested on an independent sample, what appeared as an association in the original dataset may not hold when testing the association in a new dataset.

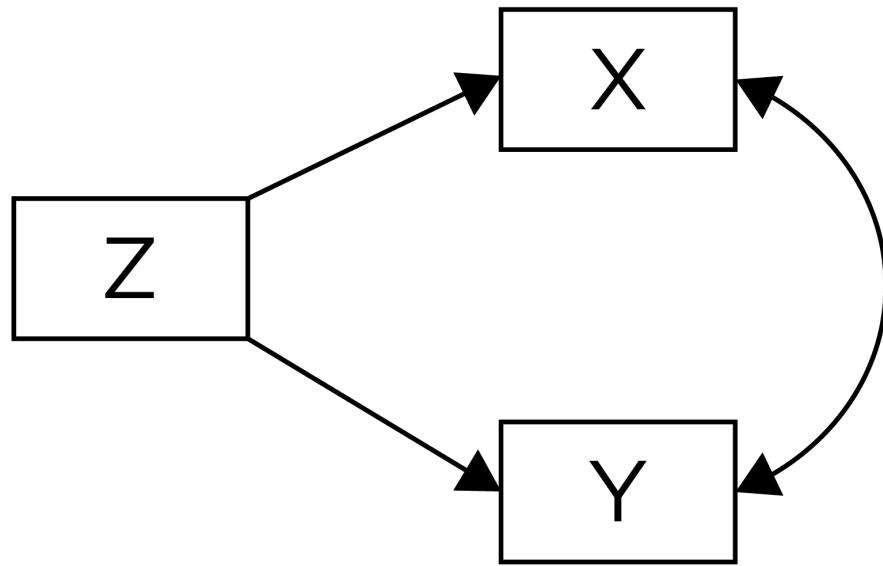


Figure 6.3 Confounded Association Between X and Y due to a Common Cause, Z .

6.3.3 Case Study

In a case study, we assess a small sample of individuals (commonly only one person or a few people), often with rich qualitative information. Themes may be coded from the qualitative information, which may help inform inferences about whether some process may have played a role in influencing the outcome of interest. The inferences are then drawn in a subjective, qualitative way. Testimonials and anecdotes are examples are case studies.

For instance, to use a case study to evaluate the possibility that players who drink sports drinks perform better than players who drink only water, we could conduct an in-depth interview with a player. In the interview, we might ask the player how they performed in games with versus without a sports drink and have them discuss whether they believe the sports drink improved their performance (and if so, how). Then, based on the player's responses, we might code the responses to extract themes and to make a qualitative judgement of whether or not the player likely performed better during games in which they had a sports drink.

6.3.4 Other Features of the Research Design

6.3.4.1 Number of Timepoints

In addition to whether the research design is an [experiment](#), [correlational/observational study](#), or a [case study](#), a research design can also have one or multiple timepoints. The differing number of timepoints allow studies to be characterized as one of the following:

- cross-sectional
- longitudinal

6.3.4.1.1 Cross-Sectional

A *cross-sectional study* is a study with one timepoint.

For instance, in a cross-sectional study evaluating whether having a sports drink improves player performance, we might assess players' drinking behavior and performance during only game 1.

Cross-sectional studies are more common than longitudinal studies because cross-sectional studies are less costly and time-consuming. They can provide a helpful starting point to test findings more rigorously in subsequent longitudinal studies.

6.3.4.1.2 Longitudinal Design

A *longitudinal study* is a study with more than one timepoint. When the same measures are assessed at each of multiple timepoints, we refer to this as a "repeated measures" design.

In a longitudinal study evaluating whether having a sports drink improves player performance, we might assess players' drinking behavior and performance during each game of the season, and possibly across multiple seasons.

Longitudinal studies are less common than cross-sectional studies because longitudinal studies are more costly and time-consuming. Nevertheless, longitudinal studies can allow us to test our hypotheses more rigorously, because they can allow us to test whether changes in the predictor/independent variable leads to changes in the outcome/dependent variable. Thus, compared to cross-sectional studies, longitudinal studies can provide greater confidence in causal inferences.

6.3.4.2 Within- or Between-Subject

A research design can also be within-subject, between-subject, or both. A study can involve both within-subject and between-subject comparisons if one predictor/independent variable is within-subject and another predictor/independent variable is between-subject.

6.3.4.2.1 Within-Subject Design

A *within-subject design* is one in which each participant (i.e., person or player) receives multiple levels of the independent variable (or predictor).

For instance, in an experiment evaluating whether having a sports drink improves player performance, we might assign players to drink the sports drink in the first half of the game and to drink only water in the second half of the game. Or we could assign some of the players to drink sports drink in the first half and water in the second half, and assign the other players to drink water in the first half and sports drink in the second half.

In a correlational study evaluating whether having a sports drink improves player performance, we might evaluate how within-person changes in sports drink consumption are associated with within-person changes in performance. That is, we could evaluate, when a given player has a sports drink (or more sports drinks), do they perform better than when the same individual has only water (or fewer sports drinks)?

Within-subject designs tend to have greater statistical power than between-subject designs. However, within-subject designs often have *carryover effects*. For instance, consider the study in which we assign players to drink only water in the first and third quarters and to drink sports drink in the second and fourth quarters (an A-B-A-B design). Drinking sports drink in the second quarter could increase how much hydration a player has throughout the rest of the game, which could lead to altered performance in the third and fourth quarters that is not due to what they drink in third and fourth quarters.

6.3.4.2.2 Between-Subject Design

A *between-subject design* is one in which each participant (i.e., person or player) receives only one level of the independent variable.

For instance, in an experiment evaluating whether having a sports drink improves player performance, we might assign some players to drink the sports drink but the other players to drink only water.

In a correlational study evaluating whether having a sports drink improves player performance, we might evaluate whether people who drink sports drinks tend to perform better than players who drink only water. Or, we could evaluate whether players who drink more sports drinks perform better than players who drink fewer sports drinks (i.e., whether the number of sports drinks consumed during a game is correlated with player performance).

6.4 Research Design Validity

Research design validity involves the accuracy of inferences from a study. There are three types of research design validity:

- internal validity
- external validity
- conclusion validity

6.4.1 Internal Validity

Internal validity is the extent to which we can be confident that the associations identified in the study are causal.

6.4.2 External Validity

External validity is the extent to which we can be confident that findings from the study play out similarly in the real world—that is, the findings generalize to the target population.

6.4.3 Tradeoffs Between Internal and External Validity

There is a tradeoff between **internal** and **external** validity—a single research design cannot have both high **internal** and high **external validity**. Each study and design has weaknesses. Some research designs are better suited for making causal inferences, whereas other designs tend to be better suited for making inferences that generalize to the real world. The research design that is best suited to making causal inferences is an **experiment** because it is the design in which the researcher has the greatest control over the variables. Thus, **experiments** tend to have higher **internal validity** than other research designs. However, by manipulating one variable and holding everything else constant,

the research takes place in a very standardized fashion that can become like studying a process in a vacuum. So, even if a process is theoretically causal in a vacuum, it may act differently in the real world when it interacts with other processes.

Correlational designs have greater capacity for [external validity](#) than [experimental designs](#) because the participants can be observed in their natural environments to evaluate how variables are related in the real world. However, the greater [external validity](#) comes at a cost of lower [internal validity](#). Correlational designs are not well-positioned to make causal inferences. [Correlational studies](#) can account for potential confounds using *covariates* or for the reverse direction of effect using longitudinal designs, but the researcher has less control over the variables than in an [experiment](#).

As the [internal validity](#) of a study's design increases, its [external validity](#) tends to decrease. The greater control we have over variables (and, therefore, have greater confidence about causal inferences), the lower the likelihood that the findings reflect what happens in the real world because it is studying things in a metaphorical vacuum. Because no single research design can have both high [internal](#) and [external](#) validity, scientific inquiry needs a combination of many different research designs so we can be more confident in our inferences—[experimental designs](#) for making causal inferences and [correlational designs](#) for making inferences that are more likely to reflect the real world.

[Case studies](#), because they have smaller sample sizes and inferences drawn in a subjective, qualitative way, tend to have lower [external validity](#) than both [experimental](#) and [correlational](#) studies. [Case studies](#) also tend to have lower [internal validity](#) because they have less control over variables, and thus fail to remove the possibility of illusory correlations, potential confounds, or the reverse direction of effect. Thus, [case studies](#) are among the weakest forms of evidence. Nevertheless, case studies can still be useful for generating hypotheses that can then be tested empirically with a larger sample in [experimental](#) or [correlational](#) studies.

6.4.4 Conclusion Validity

Conclusion validity is the extent to which a study's conclusions are reasonable about the association among variables based on the data. That is, were the correct statistical analyses performed, and are the interpretations of the findings from those analyses correct?

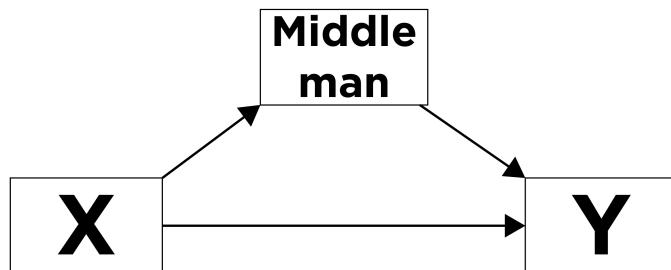
6.5 Mediation vs Moderation

Both types of effects involve (at least) three variables:

1. An independent/predictor variable, which will be labeled as x .
2. A dependent/outcome variable, which will be labeled as y .
3. The mediator or moderator variable, which will be labeled as M .

A mnemonic to help remember the difference between **mediation** and **moderation** is in Figure 6.4.

Mediation = a ‘middle man’ along the pathway



Moderation = the effect/path is ‘modified’

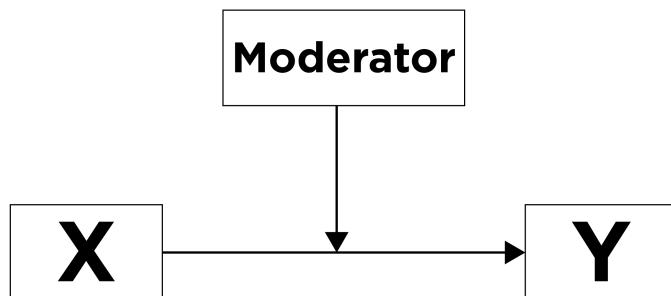


Figure 6.4 Mediation Versus Moderation Mnemonic.

6.5.1 Mediation

6.5.1.1 Overview

Mediation is a causal chain of events, where one variable (a mediator variable) at least partially explains (or accounts for) the association between two other variables (the predictor variable and the outcome variable). In mediation, a predictor (X) leads to a mediator (M), which leads to an outcome (Y). Mediation answers the question of, “**Why (or how)** does X influence Y ? A mediator (M) is a variable that helps explain the association between two other variables, and it answers the question of why/how X influences Y . That is, the mediator is the variable that helps explain how/why X is related to Y . In other words, you can think of the mediator as the mechanism that helps explain why X has an impact on Y . The association between X and Y gets smaller when accounting for M . Visually this can be written as in Figure 6.5:



Figure 6.5 Mediation.

where X is causing M , which in turn is causing Y . In other words, X leads to M , and M leads to Y .

For instance, if we determine that consuming sports drinks improves player performance, we may want to know how/why. That is, what is the mechanism that leads consumption of sports drinks to improve player performance? We might hypothesize that consumption of sports drink helps increase a player’s hydration, which in turn will improve the player’s performance. In this case, increased hydration mediates (i.e., helps explain or account for) the effect of the sports drink consumption on improved player performance.

Question: Why/how does sports drink consumption lead players to perform better?

Answer: increased hydration

As a picture, we can draw this association as in Figure 6.6:

6.5.1.2 Types of Mediation

6.5.1.2.1 Full Mediation

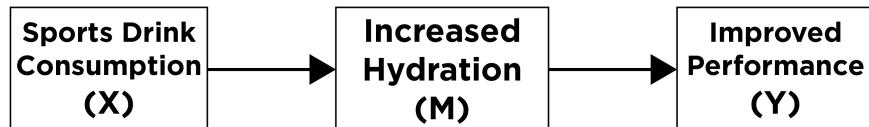


Figure 6.6 Mediation Example.

When one mechanism fully accounts for the effect of the predictor variable on the outcome variable, this is known as **full mediation**, as depicted in Figure 6.7:



Figure 6.7 Full Mediation.

6.5.1.2.2 Partial Mediation

When a single process partially—but does not fully—accounts for the effect of the predictor variable on the outcome variable; this is known as **partial mediation** and is depicted in Figure 6.8:

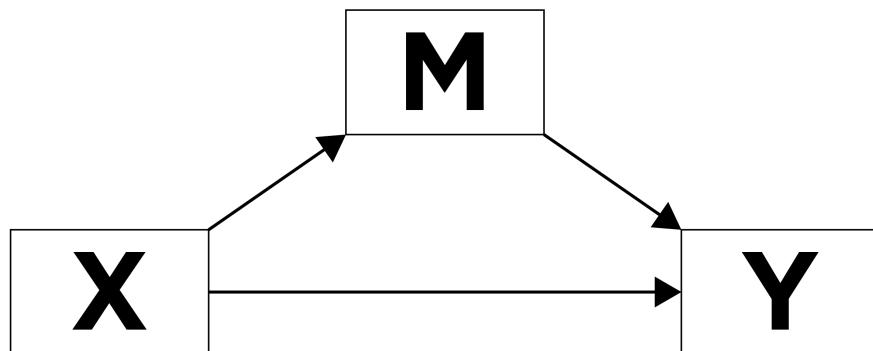


Figure 6.8 Partial Mediation.

6.5.1.2.3 Multiple Mediators

In addition, there can be multiple mediators/mechanisms that account for the effect of a predictor variable on an outcome variable, as depicted in Figure 6.9:

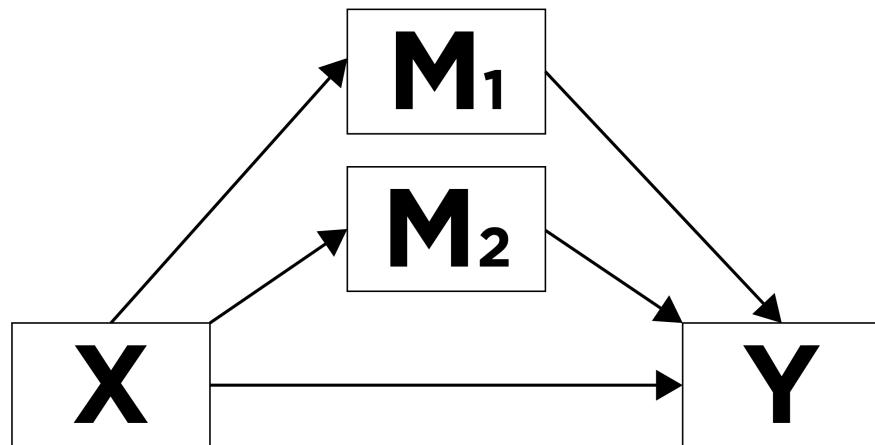


Figure 6.9 Multiple Mediators.

6.5.2 Moderation (i.e., Interaction)

6.5.2.1 Overview

Moderation (sometimes called an “interaction”), on the other hand, occurs when there is a variable or condition (M ; called a “moderator”) that changes the association between X and Y . That is, the effect of the predictor variable on the outcome variable differs at different levels of the moderator variable. In these cases, X and M work together to have an effect on Y ; here X does not have a direct effect on M . Moderation answers the question of, “**For whom** does X influence Y ? ” If X influences Y more strongly for some people or in some circumstances, we would say that there is an interaction such that the effect of X on Y depends on M , as depicted in Figure 6.10:

For example, if the effect of consuming sports drinks on player performance differs for Quarterbacks and Wide Receivers, the interaction could be depicted in Figure 6.11 and Figure 6.12:

An interaction can be identified visually by non-parallel lines at different levels of the moderator. In this example, the player’s position moderates the effect consuming sports drinks on player performance. In particular, there is a strong

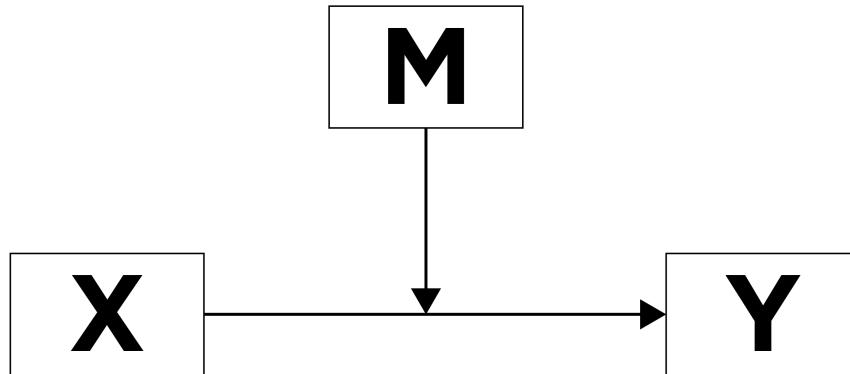


Figure 6.10 Moderation.

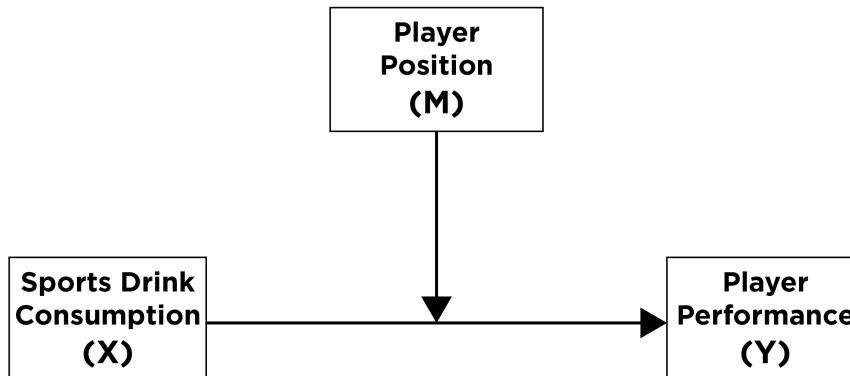


Figure 6.11 Moderation Example: Path Diagram.

positive association between consuming sports drinks and player performance for Wide Receivers (as evidenced by the upward slope of the best-fit regression line), whereas there is no association between consuming sports drinks and player performance for Quarterbacks (as evidenced by the flat line).

6.6 Levels of Measurement

It is important to know the levels of measurement of your data, because the level(s) of measurement of your data constrain the types of comparisons and

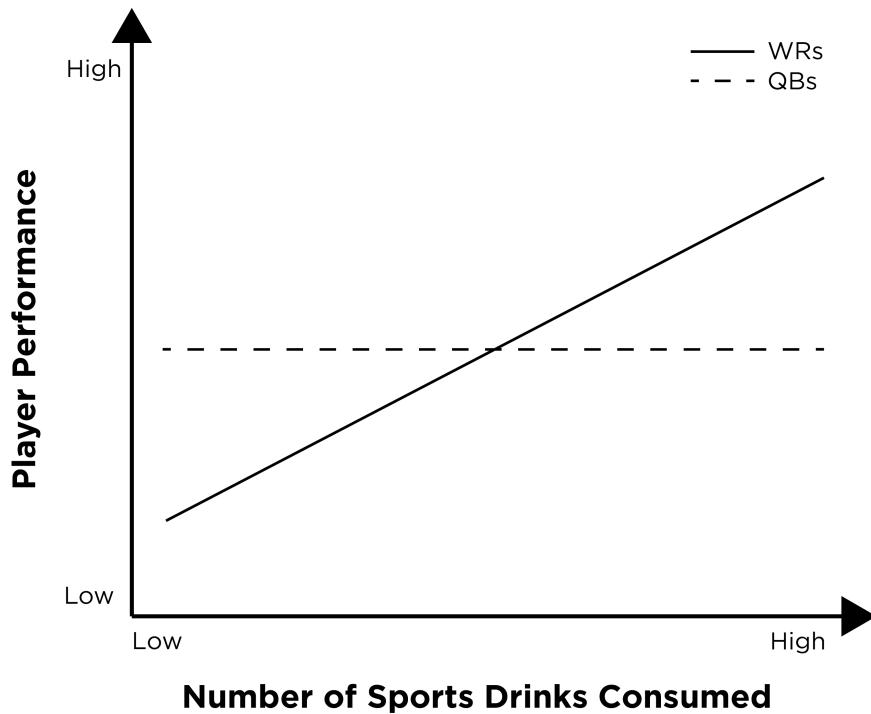


Figure 6.12 Moderation Example: Interaction Graph.

analyses that you can meaningfully perform. There are four levels of measurement that any variable can have:

- nominal
- ordinal
- interval
- ratio

Each is described below:

6.6.1 Nominal

A variable is considered nominal if it is composed of qualitative classifications. You cannot meaningfully evaluate whether one number in the variable is larger than another number in the variable because higher numbers do not reflect higher levels of the concept. Examples of nominal variables include:

- sex (e.g., 1 = male; 2 = female)

- race (e.g., 1 = American Indian; 2 = Asian; 3 = Black; 4 = Pacific Islander; 5 = White)
- ethnicity (e.g., 0 = Non-Hispanic/Latino; 1 = Hispanic/Latino)
- zip code
- jersey number

A football player's jersey number is an example of a nominal variable. A jersey number of 7 is not higher on whatever concept of interest compared to a jersey number of 6.

To examine the central tendency of a nominal variable, you can determine the mode, but you cannot calculate a mean or median.

6.6.2 Ordinal

A variable is considered ordinal if the classifications are ordered. However, ordinal variables do not have equally spaced intervals. Examples of ordinal intervals include:

- likert response scales (e.g., 1 = strongly disagree; 2 = disagree; 3 = neutral; 4 = agree; 5 = strongly agree)
- educational attainment (e.g., 1 = no formal education; 2 = elementary school; 3 = middle school; 4 = high school; 5 = college; 6 = graduate degree)
- academic grades on A–F scale (e.g., 1 = A; 2 = B; 3 = C; 4 = D; 5 = F)
- player rank (1 = 1st; 2 = 2nd; 3 = 3rd, etc.)

A football player's fantasy rank is an example of an ordinal variable. A player with a fantasy rank of 1 has a higher rank than a player with a rank of 2, but it is not known how far apart each player is—i.e., the intervals do not all reflect the same distance. For instance, the distance between the top-ranked player and the 2nd-best player might be 30 points, whereas the distance between the 2nd-best player and the 3rd-best player might be 2 points.

To examine the central tendency of ordinal data, the median and mode are most appropriate; however, the mean may be used (unlike for nominal data).

6.6.3 Interval

A variable is considered interval if the classifications are ordered (similar to ordinal data) and have equally spaced intervals (unlike ordinal data). However, interval variables do not have a meaningful zero that reflects absence. Examples of interval data include:

- temperature on the Fahrenheit or Celsius scale

- time of day

For instance, the temperature difference between 80 and 90 degrees Fahrenheit is the same as the temperature difference between 90 and 100 degrees Fahrenheit. However, 0 degrees Fahrenheit does not reflect absence of temperature/heat.

Interval data can be meaningfully added or subtracted. For instance, if a game starts at 4 pm and ends at 7 pm, you know the game lasted 3 hours ($7 - 4 = 3$). However, interval data cannot be meaningfully multiplied or divided. For instance, 100 degrees Fahrenheit is not twice as hot as 50 degrees Fahrenheit.

To examine the central tendency of interval data, you can compute the mean, median, or mode.

6.6.4 Ratio

A variable is considered ratio if the classifications are ordered (similar to ordinal data), have equally spaced intervals (like interval data), and have an absolute zero point that reflects absence of the concept. Examples of ratio data include:

- temperature on the Kelvin scale
- height
- weight
- age
- distance
- speed
- volume
- time elapsed
- income
- years of formal education
- points in football

For instance, points in football has order, equally spaced intervals, and an absolute zero—a team cannot score less than zero points, and zero points reflects absence of points (though it could be argued to be interval data because zero points does not reflect absence of skill.)

Ratio data can be meaningfully added, subtracted, multiplied, or divided. A player who weighs 350 pounds weighs twice as much as someone who weighs 175 pounds.

To examine the central tendency of ratio data, you can compute the mean, median, or mode.

6.7 Psychometrics

Below, I provide brief discussions of various aspects of measurement reliability and validity. For more information on these and other aspects of psychometrics, see Petersen (2024b) and Petersen (2024c).

6.7.1 Measurement Reliability

The *reliability* of a measure's scores deals with the *consistency* of measurement. This book focuses on the following types of reliability:

- test-retest reliability
- inter-rater reliability
- intra-rater reliability
- internal consistency
- parallel-forms reliability

For more information on these and other aspects of reliability, see <https://isaactpetersen.github.io/Principles-Psychological-Assessment/reliability.html> (Petersen, 2024b, 2024c).

6.7.1.1 Test-Retest Reliability

Test-retest reliability evaluates the consistency of scores across time. For a construct that is expected to be stable across time (e.g., hand size in adults), we would expect our measurements to be consistent across time. The consistency of scores across time can be examined in terms of relative or absolute test-retest reliability. Relative test-retest reliability—i.e., the consistency of individual differences across time—is commonly evaluated using the coefficient of stability (i.e., the Pearson correlation coefficient). Absolute test-retest reliability—i.e., the absolute consistency of people's scores across time—is commonly evaluated using the coefficient of repeatability.

6.7.1.2 Inter-Rater Reliability

Inter-rater reliability evaluates the consistency of scores across raters. For instance, if we have a strong measure for assessing college players' aptitude to succeed in the NFL, the measure should yield a similar score for a given player regardless of which (trained) rater (e.g., coach or talent scout) uses it to rate the player. The consistency of scores across raters is commonly

evaluated using the intraclass correlation coefficient (for continuous variables) and Cohen's kappa (κ ; for categorical variables).

6.7.1.3 Intra-Rater Reliability

Intra-rater reliability evaluates the consistency of scores within a given rater. If we have a strong measure for assessing college players' aptitude to succeed in the NFL, the measure should yield a similar score for a given player from the same (trained) rater (e.g., coach or talent scout) each time they rate the same player (assuming the player's aptitude has not changed). The consistency of scores within raters can be evaluated using similar approaches as those evaluating [inter-rater reliability](#).

6.7.1.4 Internal Consistency

Internal consistency evaluates the consistency of scores across items within a measure. If we develop a strong questionnaire measure to assess a college players' aptitude to succeed in the NFL, the scores should be relatively consistent across items. The consistency of scores across items within a measure is commonly evaluated using Cronbach's alpha (α) or McDonald's omega (ω).

6.7.1.5 Parallel-Forms Reliability

Parallel-forms reliability evaluates the consistency of scores across different but equivalent forms of a measure. If we develop two equivalent versions of the Wonderlic Contemporary Cognitive Ability Test (Form A and Form B) so that players sitting next to each other do not receive the same items, we would expect a player's score on Form A would be similar to their score on Form B. Parallel-forms reliability is commonly evaluated using the coefficient of equivalence (i.e., the Pearson correlation coefficient).

6.7.2 Measurement Validity

The *validity* of a measure's scores deals with the *accuracy* of measurement. This book focuses on the following types of validity:

- [face validity](#)
- [content validity](#)
- [criterion-related validity](#)
 - [concurrent \(criterion-related\) validity](#)
 - [predictive \(criterion-related\) validity](#)
- [construct validity](#)

- convergent validity
- discriminant validity
- incremental validity

For more information on these and other aspects of validity, see <https://isaactpetersen.github.io/Principles-Psychological-Assessment/validity.html> (Petersen, 2024b, 2024c).

6.7.2.1 Face Validity

Face validity evaluates the extent to which a measure “looks like” (on its face) it assesses the construct of interest. For instance, if a measure is developed to assess aptitude of Wide Receivers for the position, it would be considered to have face validity if everyday (lay) people believe that it assesses aptitude for being a successful Wide Receiver.

6.7.2.2 Content Validity

Content validity evaluates the extent to which the measure assesses the full breadth of the content, as determined by context experts. For the measure to have content validity, it should not have gaps (missing content facets) or intrusions (facets of other constructs). For instance, a strong measure for assessing a player’s aptitude to succeed in the NFL might need to include a player’s speed, strength, size, lateral quickness, etc. If the measure is missing their speed, this would be a content gap. If the measure assesses a construct-irrelevant facet (e.g., their attractiveness), this would be a content intrusion.

6.7.2.3 Criterion-Related Validity

Criterion-related validity evaluates the extent to which the measure’s scores are related to meaningful variables of interest. Criterion-related validity is commonly evaluated using a Pearson correlation or some form of regression.

There are two types of criterion-related validity:

- concurrent (criterion-related) validity
- predictive (criterion-related) validity

6.7.2.3.1 Concurrent (Criterion-Related) Validity

Concurrent criterion-related validity evaluates the extent to which the measure’s scores are related to meaningful variables of interest assessed at the

same point in time. That is, concurrent validity could evaluate whether current player statistics (e.g., passing yards) are associated with their fantasy points.

6.7.2.3.2 Predictive (Criterion-Related) Validity

Predictive criterion-related validity evaluates the extent to which the measure's scores are related to meaningful variables of interest that are assessed at a later point in time. For example, predictive validity could evaluate whether scores on the measure we developed to assess a player's aptitude to succeed in the NFL predicts later performance in the NFL.

6.7.2.4 Construct Validity

Construct validity evaluates the extent to which the measure's scores accurately assess the construct of interest. If we develop a measure with intent to assess aptitude for being a successful Running Back, and it appears to more accurately assess aptitude for being a successful Wide Receiver, then our measure has poor construct validity for assessing aptitude for being a successful Running Back. Construct validity subsumes **convergent** and discriminant validity, in addition to all of the other forms of measurement validity.

6.7.2.5 Convergent Validity

Convergent validity evaluates the extent to which the measure's scores are related to other measures of the same construct. For instance, if we develop a new measure to assess intelligence, its scores should be related to scores from other measures designed to assess intelligence (e.g., Wonderlic Contemporary Cognitive Ability Test).

6.7.2.6 Discriminant Validity

Discriminant validity evaluates the extent to which the measure's scores are unrelated to measures of the different constructs. For instance, if we develop a new measure to assess intelligence, its scores should be less strongly associated with measures of other constructs (e.g., measures of happiness).

6.7.2.7 Incremental Validity

Incremental validity evaluates the extent to which the measure's scores provide an increase in predictive accuracy compared to other information that is easily

and cheaply available. That is, in order to be useful, a strong measure should tell us something that we did not already know. For instance, if we develop a strong measure of intelligence, it should result in increased predictive accuracy (for success in the NFL) compared to when just relying on the Wonderlic Contemporary Cognitive Ability Test.

6.7.3 Reliability vs Validity

Reliability and validity are different but related. Reliability refers to the *consistency* of scores, whereas accuracy refers to the *accuracy* of scores. Validity depends on reliability. Reliability is necessary—but insufficient for—validity. That is, consistency is necessary—but insufficient for—accuracy. As depicted in Figure 6.13, a measure can be no more valid than it is reliable. A measure can be consistent but inaccurate; however, a measure cannot be accurate but inconsistent.



Figure 6.13 Reliability Versus Validity.

7

Basic Statistics

7.1 Getting Started

7.1.1 Load Packages

```
library("petersenlab")
library("pwr")
library("pwrss")
library("WebPower")
library("grid")
library("tidyverse")
```

7.2 Descriptive Statistics

Descriptive statistics are used to describe data. For instance, they may be used to describe the center, spread, or shape of the data. There are various indices of each.

7.2.1 Center

Indices to describe the *center* (central tendency) of a variable's data include:

- mean
- median
- mode

The mean of X (written as: \bar{X}) is calculated as:

$$\bar{X} = \frac{\sum X_i}{n} = \frac{X_1 + X_2 + \dots + X_n}{n} \quad (7.1)$$

That is, to compute the mean, sum all of the values and divide by the number of values (n).

The median is determined as the value at the 50th percentile (i.e., the value that is higher than 50% of the values and is lower than the other 50% of values).

The mode is the most common/frequent value.

Below is R code to estimate each:

7.2.2 Spread

Indices to describe the *spread* (variability) of a variable's data include:

- standard deviation
- variance
- range
- minimum and maximum
- interquartile range (IQR)

The (sample) variance of X (written as: s^2) is calculated as:

$$s^2 = \frac{\sum(X_i - \bar{X})^2}{n - 1} \quad (7.2)$$

The (sample) standard deviation of X (written as: s) is calculated as:

$$s = \sqrt{\frac{\sum(X_i - \bar{X})^2}{n - 1}} \quad (7.3)$$

The range is calculated of X is calculated as:

$$\text{range} = \text{maximum} - \text{minimum} \quad (7.4)$$

The interquartile range (IQR) is calculated as:

$$\text{IQR} = Q_3 - Q_1 \quad (7.5)$$

where Q_3 is the score at the third quartile (i.e., 75th percentile), and Q_1 is the score at the first quartile (i.e., 25th percentile).

Below is R code to estimate each:

7.2.3 Shape

Indices to describe the *shape* of a variable's data include:

- skewness
- kurtosis

Below is R code to estimate each:

7.2.4 Combination

To estimate multiple indices of center, spread, and shape of the data, you can use the following code:

```
#psych::describe(mydata)

#mydata %>%
#  summarise(across(
#    everything(),
#    .fns = list(
#      n = ~ length(na.omit(.)),
#      missingness = ~ mean(is.na(.)) * 100,
#      M = ~ mean(., na.rm = TRUE),
#      SD = ~ sd(., na.rm = TRUE),
#      min = ~ min(., na.rm = TRUE),
#      max = ~ max(., na.rm = TRUE),
#      range = ~ max(., na.rm = TRUE) - min(., na.rm = TRUE),
#      IQR = ~ IQR(., na.rm = TRUE),
#      median = ~ median(., na.rm = TRUE),
#      mode = ~ petersenlab::Mode(., multipleModes = "mean"),
#      skewness = ~ psych::skew(., na.rm = TRUE),
#      kurtosis = ~ psych::kurtosi(., na.rm = TRUE)),
#      .names = "{.col}.fn}")) %>%
#  pivot_longer(
#    cols = everything(),
#    names_to = c("variable","index"),
#    names_sep = "\\.")
#  %>%
#  pivot_wider(
#    names_from = index,
#    values_from = value)
```

7.3 Scores and Scales

There are many different types of scores and scales. This book focuses on [raw scores](#) and [z-scores](#). For information on other scores and scales, including percentile ranks, *T*-scores, standard scores, scaled scores, and stanine scores, see here: <https://isaactpetersen.github.io/Principles-Psychological-Assessment/scoresScales.html#scoreTransformation> (Petersen, 2024c).

7.3.1 Raw Scores

Raw scores are the original data on the original metric. Thus, raw scores are considered *unstandardized*. For example, raw scores that represent the players' age may range from 20 to 40. Raw scores depend on the construct and unit; thus raw scores may not be comparable across variables.

7.3.2 *z* Scores

z scores have a mean of zero and a standard deviation of one. *z* scores are frequently used to render scores across variables more comparable. Thus, *z* scores are considered a form of a *standardized score*.

z scores are calculated using Equation 7.6:

$$z = \frac{X - \bar{X}}{\sigma} \quad (7.6)$$

where X is the observed score, \bar{X} is the mean observed score, and σ is the standard deviation of the observed scores.

You can easily convert a variable to a *z* score using the `scale()` function:

```
scale(variable)
```

With a standard normal curve, 68% of scores fall within one standard deviation of the mean. 95% of scores fall within two standard deviations of the mean. 99.7% of scores fall within three standard deviations of the mean.

The area under a normal curve within one standard deviation of the mean is calculated below using the `pnorm()` function, which calculates the cumulative density function for a normal curve.

```
stdDeviations <- 1

pnorm(stdDeviations) - pnorm(stdDeviations * -1)

[1] 0.6826895
```

The area under a normal curve within one standard deviation of the mean is depicted in Figure 7.1.

```
x <- seq(-4, 4, length = 200)
y <- dnorm(x, mean = 0, sd = 1)
plot(x, y, type = "l",
      xlab = "z Score",
      ylab = "Normal Density")

x <- seq(stdDeviations * -1, stdDeviations, length = 100)
y <- dnorm(x, mean = 0, sd = 1)
polygon(c(stdDeviations * -1, x, stdDeviations),
        c(0, y, 0),
        col = "blue")
```

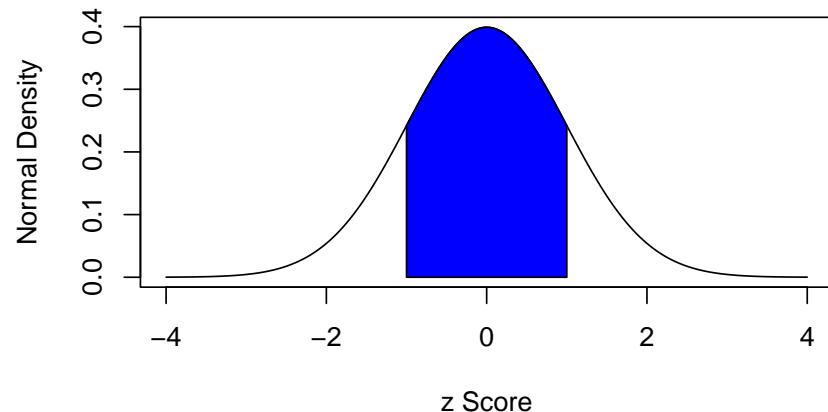


Figure 7.1 Density of Standard Normal Distribution. The blue region represents the area within one standard deviation of the mean.

The area under a normal curve within two standard deviations of the mean is calculated below:

```
stdDeviations <- 2
pnorm(stdDeviations) - pnorm(stdDeviations * -1)
[1] 0.9544997
```

The area under a normal curve within two standard deviations of the mean is depicted in Figure 7.2.

```
x <- seq(-4, 4, length = 200)
y <- dnorm(x, mean = 0, sd = 1)
plot(x, y, type = "l",
      xlab = "z Score",
      ylab = "Normal Density")

x <- seq(stdDeviations * -1, stdDeviations, length = 100)
y <- dnorm(x, mean = 0, sd = 1)
polygon(c(stdDeviations * -1, x, stdDeviations),
        c(0, y, 0),
        col = "blue")
```

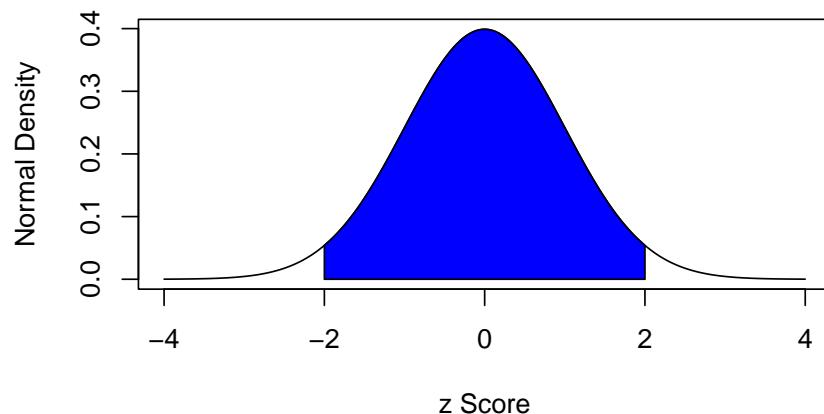


Figure 7.2 Density of Standard Normal Distribution. The blue region represents the area within two standard deviations of the mean.

The area under a normal curve within three standard deviations of the mean is calculated below:

```
stdDeviations <- 3  
  
pnorm(stdDeviations) - pnorm(stdDeviations * -1)  
  
[1] 0.9973002
```

The area under a normal curve within three standard deviations of the mean is depicted in Figure 7.3.

```
x <- seq(-4, 4, length = 200)  
y <- dnorm(x, mean = 0, sd = 1)  
plot(x, y, type = "l",  
      xlab = "z Score",  
      ylab = "Normal Density")  
  
x <- seq(stdDeviations * -1, stdDeviations, length = 100)  
y <- dnorm(x, mean = 0, sd = 1)  
polygon(c(stdDeviations * -1, x, stdDeviations),  
        c(0, y, 0),  
        col = "blue")
```

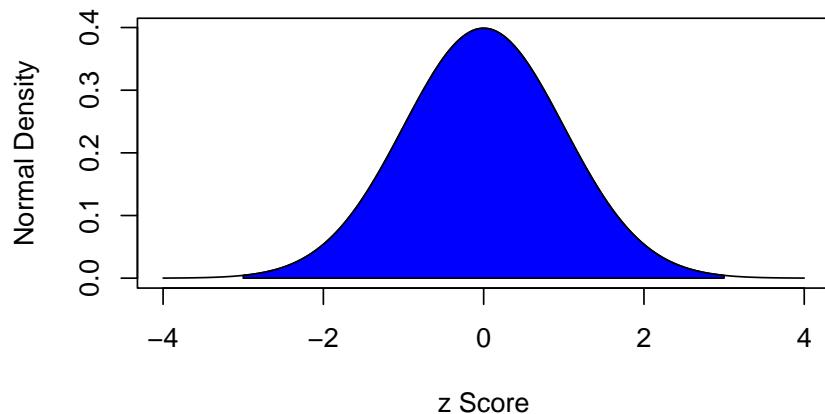


Figure 7.3 Density of Standard Normal Distribution. The blue region represents the area within three standard deviations of the mean.

If you want to determine the z score associated with a particular percentile in a normal distribution, you can use the `qnorm()` function. For instance, the z score associated with the 37th percentile is:

```
qnorm(.37)
```

```
[1] -0.3318533
```

7.4 Inferential Statistics

Inferential statistics are used to draw inferences regarding whether there is (a) a difference in level on variable across groups or (b) an association between variables. For instance, inferential statistics may be used to evaluate whether Quarterbacks tend to have longer careers compared to Running Backs. Or, they could be used to evaluate whether number of carries is associated with injury likelihood. To apply inferential statistics, we make use of the null hypothesis (H_0) and the alternative hypothesis (H_1).

7.4.1 Null Hypothesis Significance Testing

7.4.1.1 Null Hypothesis (H_0)

When testing whether there are differences in level across groups on a variable of interest, the null hypothesis (H_0) is that there is no difference in level across groups. For instance, when testing whether Quarterbacks tend to have longer careers compared to Running Backs, the null hypothesis (H_0) is that Quarterbacks do not systematically differ from Running Backs in the length of their career.

When testing whether there is an association between variables, the null hypothesis (H_0) is that there is no association between the variables. For instance, when testing whether number of carries is associated with injury likelihood, the null hypothesis (H_0) is that there is no association between number of carries and injury likelihood.

7.4.1.2 Alternative Hypothesis (H_1)

The alternative hypothesis (H_1) is the researcher's hypothesis that they want to evaluate. An alternative hypothesis (H_1) might be directional (i.e., one-sided) or non-directional (i.e., two-sided).

Directional hypotheses specify a particular direction, such as which group will have larger scores or which direction (positive or negative) two variables will be associated. Examples of directional hypotheses include:

- Quarterbacks have longer careers compared to Running Backs
- Number of carries is positively associated with injury likelihood

Non-directional hypotheses do not specify a particular direction. For instance, non-directional hypotheses may state that two groups differ but do not specify which group will have larger scores. Or, non-directional hypotheses may state that two variables are associated but do not state what the sign is of the association—i.e., positive or negative. Examples of non-directional hypotheses include:

- Quarterbacks differ in the length of their careers compared to Running Backs
- Number of carries is associated with injury likelihood

7.4.1.3 Statistical Significance

In science, statistical significance is evaluated with the p -value. The p -value does not represent the probability that you observed the result by chance. The p -value represents a conditional probability—it examines the probability of one event given another event. In particular, the p -value evaluates the likelihood that you would detect a result as at least as extreme as the one observed (in terms of the magnitude of the difference or of the association) given that the null hypothesis (H_0) is true.

This can be expressed in conditional probability notation, $Pr(A|B)$, which is the probability (likelihood) of event A occurring given that event B occurred (or given condition B).

The conditional probability notation for a left-tailed directional test (i.e., Quarterbacks have shorter careers than Running Backs; or number of carries is negatively associated with injury likelihood) is in Equation 7.7.

$$p\text{-value} = Pr(T \leq t|H_0) \quad (7.7)$$

where T is the test statistic of interest (e.g., the distribution of t -, r -, or F values, depending on the test) and t is the observed test statistic (e.g., t -, r -, or F -coefficient, depending on the test).

The conditional probability notation for a right-tailed directional test (i.e., Quarterbacks have longer careers than Running Backs; or number of carries is positively associated with injury likelihood) is in Equation 7.8.

$$p\text{-value} = Pr(T \geq t|H_0) \quad (7.8)$$

The conditional probability notation for a two-tailed non-directional test (i.e., Quarterbacks differ in the length of their careers compared to Running Backs; or number of carries is associated with injury likelihood) is in Equation 7.9.

$$p\text{-value} = 2 \times \min(Pr(T \leq t|H_0), Pr(T \geq t|H_0)) \quad (7.9)$$

where $\min(a, b)$ is the smaller number of a and b .

If the distribution of the test statistic is symmetric around zero, the p -value for the two-tailed non-directional test simplifies to Equation 7.10.

$$p\text{-value} = 2 \times Pr(T \geq |t||H_0) \quad (7.10)$$

Nevertheless, to be conservative (i.e., to avoid false positive/Type I errors), many researchers use two-tailed p -values regardless whether their hypothesis is one- or two-tailed.

For a test of group differences, the p -value evaluates the likelihood that you would observe a difference as large or larger than the one you observed between the groups if there were no systematic difference between the groups, as depicted in Figure 7.4. For instance, when evaluating whether Quarterbacks have longer careers than Running Backs, and you observed a mean difference of 0.03 years, the p -value evaluates the likelihood that you would observe a difference as larger or larger than 0.03 years between the groups if Quarterbacks do not differ from Running Backs in terms of the length of their career.

```
set.seed(52242)

nObserved <- 1000
nPopulation <- 1000000

observedGroups <- data.frame(
  score = c(rnorm(nObserved, mean = 47, sd = 3), rnorm(nObserved, mean = 52, sd = 3)),
  group = as.factor(c(rep("Group 1", nObserved), rep("Group 2", nObserved)))
)

populationGroups <- data.frame(
  score = c(rnorm(nPopulation, mean = 50, sd = 3.03), rnorm(nPopulation, mean = 50, sd = 3)),
  group = as.factor(c(rep("Group 1", nPopulation), rep("Group 2", nPopulation)))
)

ggplot2::ggplot(
  data = observedGroups,
  mapping = aes(
    x = score,
    fill = group,
```

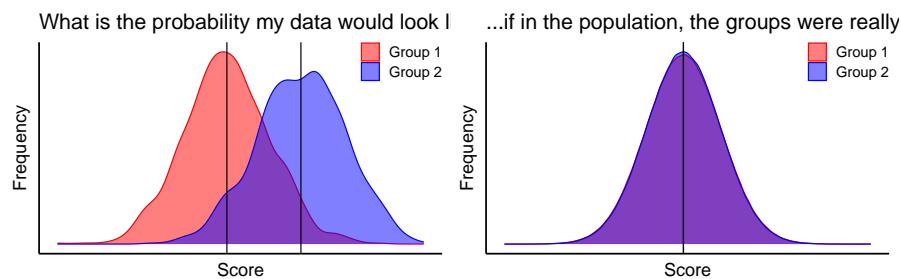
```
    color = group
)
) +
geom_density(alpha = 0.5) +
scale_color_manual(values = c("red", "blue")) +
scale_fill_manual(values = c("red","blue")) +
geom_vline(xintercept = mean(observedGroups$score[which(observedGroups$group == "Group 1")])) +
geom_vline(xintercept = mean(observedGroups$score[which(observedGroups$group == "Group 2")])) +
ggplot2::labs(
  x = "Score",
  y = "Frequency",
  title = "What is the probability my data would look like this..."
) +
ggplot2::theme_classic(
  base_size = 16) +
ggplot2::theme(
  legend.title = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  #plot.title.position = "plot"
  legend.position = "inside",
  legend.margin = margin(0, 0, 0, 0),
  legend.justification.top = "left",
  legend.justification.left = "top",
  legend.justification.bottom = "right",
  legend.justification.inside = c(1, 1),
  legend.location = "plot")

ggplot2::ggplot(
  data = populationGroups,
  mapping = aes(
    x = score,
    fill = group,
    color = group
  )
) +
geom_density(alpha = 0.5) +
scale_color_manual(values = c("red", "blue")) +
scale_fill_manual(values = c("red","blue")) +
geom_vline(xintercept = mean(populationGroups$score[which(populationGroups$group == "Group 1")])) +
geom_vline(xintercept = mean(populationGroups$score[which(populationGroups$group == "Group 2")])) +
ggplot2::labs()
```

```

x = "Score",
y = "Frequency",
title = "...if in the population, the groups were really this:"
) +
ggplot2::theme_classic(
  base_size = 16) +
ggplot2::theme(
  legend.title = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  #plot.title.position = "plot",
  legend.position = "inside",
  legend.margin = margin(0, 0, 0, 0),
  legend.justification.top = "left",
  legend.justification.left = "top",
  legend.justification.bottom = "right",
  legend.justification.inside = c(1, 1),
  legend.location = "plot")

```



- (a) What is the probability my data would look like this... (b) ...if in the population, the groups were really this?

Figure 7.4 Interpretation of p -Values When Examining The Differences Between Groups. The vertical black lines reflect the group means.

For a test of whether two variables are associated, the p -value evaluates the likelihood that you would observe an association as strong or stronger than the one you observed between the groups if there were no association between the variables, as depicted in Figure 7.5. For instance, when evaluating whether number of carries is positively associated with injury likelihood, and you observed a correlation coefficient of $r = .25$ between number of carries and injury likelihood, the p -value evaluates the likelihood that you would observe a correlation as strong or stronger than $r = .25$ between the variables if number of

carries is not associated with injury likelihood.

```
set.seed(52242)

observedCorrelation <- 0.9

correlations <- data.frame(criterion = rnorm(2000))
correlations$sample <- NA
correlations$sample[1:100] <- complement(correlations$criterion[1:100], observedCorrelation)
correlations$population <- complement(correlations$criterion, 0)

ggplot2::ggplot(
  data = correlations,
  mapping = aes(
    x = sample,
    y = criterion
  )
) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_continuous(
    limits = c(-3.5,3)
  ) +
  annotate(
    x = 0,
    y = 4,
    label = paste("italic(r) != ", 0, sep = ""),
    parse = TRUE,
    geom = "text",
    size = 7) +
  ggplot2::labs(
    x = "Predictor Variable",
    y = "Outcome Variable",
    title = "What is the probability my data would look like this..."
) +
  ggplot2::theme_classic(
    base_size = 16) +
  ggplot2::theme(
    legend.title = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank())

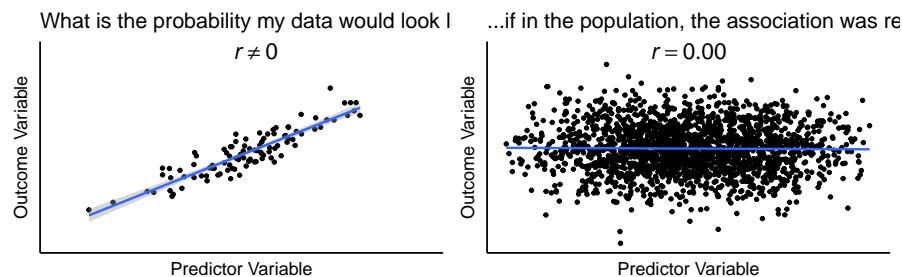
ggplot2::ggplot(
```

```

data = correlations,
mapping = aes(
  x = population,
  y = criterion
)
) +
geom_point() +
geom_smooth(
  method = "lm",
  se = FALSE) +
scale_x_continuous(
  limits = c(-2.5,2.5)
) +
annotate(
  x = 0,
  y = 4,
  label = paste("italic(r) == ''", "0.00", "", sep = ""),
  parse = TRUE,
  geom = "text",
  size = 7) +
ggplot2::labs(
  x = "Predictor Variable",
  y = "Outcome Variable",
  title = "...if in the population, the association was really this:")
) +
ggplot2::theme_classic(
  base_size = 16) +
ggplot2::theme(
  legend.title = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank())

```

Using what is called null-hypothesis significance testing (NHST), we consider an effect to be *statistically significant* if the *p*-value is less than some threshold, called the *alpha level*. In science, we typically want to be conservative because a false positive (i.e., Type I error) is considered more problematic than a false negative (i.e., Type II error). That is, we would rather say an effect does not exist when it really does than to say an effect does exist when it really does not. Thus, we typically set the alpha level to a low value, commonly .05. Then, we would consider an effect to be *statistically significant* if the *p*-value is less than .05. That is, there is a small chance (5%; or 1 in 20 times) that we would observe an effect at least as extreme as the effect observed, if the null hypothesis were true. So, you might expect around 5% of tests where the



- (a) What is the probability my data would look like this...
- (b) ...if in the population, the association was really this?

Figure 7.5 Interpretation of p -Values When Examining The Association Between Variables.

null hypothesis is true to be statistically significant just by chance. We could lower the rate of Type II (i.e., false negative) errors—i.e., we could detect more effects—if we set the alpha level to a higher value (e.g., .10); however, raising the alpha level would raise the possibility of Type I (false positive) errors.

If the p -value is less than .05, we reject the null hypothesis (H_0) that there was no difference or association. Thus, we conclude that there was a statistically significant (non-zero) difference or association. If the p -value is greater than .05, we fail to reject the null hypothesis; the difference/association was not statistically significant. Thus, we do not have confidence that there was a difference or association. However, we do not accept the null hypothesis; it could be there we did not observe an effect because we did not have adequate power to detect the effect—e.g., if the effect size was small, the data were noisy, and the sample size was small and/or unrepresentative.

There are four general possibilities of decision making outcomes when performing null-hypothesis significance testing:

1. We (correctly) reject the null hypothesis when it is in fact false ($1 - \beta$). This is a true positive. For instance, we may correctly determine that Quarterbacks have longer careers than Running Backs.
2. We (correctly) fail to reject the null hypothesis when it is in fact true ($1 - \alpha$). This is a true negative. For instance, we may correctly determine that Quarterbacks do not have longer careers than Running Backs.
3. We (incorrectly) reject the null hypothesis when it is in fact true (α). This is a false positive. When performing null hypothesis testing, a false positive is known as a Type I error. For instance, we may incorrectly determine that Quarterbacks have longer careers than

Running Backs when, in fact, Quarterbacks and Running Backs do not differ in their career length.

4. We (incorrectly) fail to reject the null hypothesis when it is in fact false (β). This is a false negative. When performing null hypothesis testing, a false negative is known as a Type II error. For instance, we may incorrectly determine that Quarterbacks and Running Backs do not differ in their career length when, in fact, Quarterbacks have longer careers than Running Backs.

A two-by-two confusion matrix for null-hypothesis significance testing is in Figure 7.6.

		Reject H_0	Fail to reject H_0
		Correct True Positive $1 - \beta$ ("power")	Type II error False Negative beta (β)
		Type I error False Positive alpha (α)	Correct True Negative $1 - \alpha$
Truth	H_0 false		
	H_0 true		

Figure 7.6 A Two-by-Two Confusion Matrix for Null-Hypothesis Significance Testing.

In statistics, *power* is the probability of detecting an effect, if, in fact, the effect exists. Otherwise said, power is the probability of rejecting the null hypothesis, if, in fact, the null hypothesis is false. Power is influenced by several variables:

- the sample size (N): the larger the N , the greater the power
 - for group comparisons, the power depends on the sample size of each group
- the **effect size**: the larger the effect, the greater the power
 - for group comparisons, larger effect sizes reflect:
 - * larger between-group variance, and
 - * smaller within-group variance (i.e., strong measurement precision, i.e., **reliability**)
- the alpha level: the researcher specifies the alpha level (though it is typically set at .05); the higher the alpha level, the greater the power; however, the higher we set the alpha level, the higher the likelihood of Type I errors (false positives)

- one- versus two-tailed tests: one-tailed tests have higher power than two-tailed tests
- **within-subject** versus **between-subject** comparisons: **within-subject designs** tend to have greater power than **between-subject designs**

A plot of statistical power is in Figure 7.7.

```
m1 <- 0 # mu H0
sd1 <- 1.5 # sigma H0
m2 <- 3.5 # mu HA
sd2 <- 1.5 # sigma HA

z_crit <- qnorm(1-(0.05/2), m1, sd1)

# set length of tails
min1 <- m1-sd1*4
max1 <- m1+sd1*4
min2 <- m2-sd2*4
max2 <- m2+sd2*4
# create x sequence
x <- seq(min(min1,min2), max(max1, max2), .01)
# generate normal dist #1
y1 <- dnorm(x, m1, sd1)
# put in data frame
df1 <- data.frame("x" = x, "y" = y1)
# generate normal dist #2
y2 <- dnorm(x, m2, sd2)
# put in data frame
df2 <- data.frame("x" = x, "y" = y2)

# Alpha polygon
y.poly <- pmin(y1,y2)
poly1 <- data.frame(x=x, y=y.poly)
poly1 <- poly1[poly1$x >= z_crit, ]
poly1<-rbind(poly1, c(z_crit, 0)) # add lower-left corner

# Beta polygon
poly2 <- df2
poly2 <- poly2[poly2$x <= z_crit,]
poly2<-rbind(poly2, c(z_crit, 0)) # add lower-left corner

# power polygon; 1-beta
poly3 <- df2
poly3 <- poly3[poly3$x >= z_crit,]
poly3 <-rbind(poly3, c(z_crit, 0)) # add lower-left corner
```

```

# combine polygons.
poly1$id <- 3 # alpha, give it the highest number to make it the top layer
poly2$id <- 2 # beta
poly3$id <- 1 # power; 1 - beta
poly <- rbind(poly1, poly2, poly3)
poly$id <- factor(poly$id, labels=c("power","beta","alpha"))

# plot with ggplot2
ggplot(poly, aes(x,y, fill=id, group=id)) +
  geom_polygon(show.legend=F, alpha=I(8/10)) +
  # add line for treatment group
  geom_line(data=df1, aes(x,y, color="H0", group=NULL, fill=NULL), linewidth=1.5, show_guide=F) +
  # add line for treatment group. These lines could be combined into one dataframe.
  geom_line(data=df2, aes(color="HA", group=NULL, fill=NULL), linewidth=1.5, show_guide=F) +
  # add vlines for z_crit
  geom_vline(xintercept = z_crit, linewidth=1, linetype="dashed") +
  # change colors
  scale_color_manual("Group",
                     values= c("HA" = "#981e0b","H0" = "black")) +
  scale_fill_manual("test", values= c("alpha" = "#0d6374","beta" = "#be805e","power"="#7ceceee")) +
  # beta arrow
  annotate("segment", x=0.1, y=0.045, xend=1.3, yend=0.01, arrow = arrow(length = unit(0.3, "cm")))
  annotate("text", label="beta", x=0, y=0.05, parse=T, size=8) +
  # alpha arrow
  annotate("segment", x=4, y=0.043, xend=3.4, yend=0.01, arrow = arrow(length = unit(0.3, "cm")),
  annotate("text", label="frac(alpha,2)", x=4.2, y=0.05, parse=T, size=8) +
  # power arrow
  annotate("segment", x=6, y=0.2, xend=4.5, yend=0.15, arrow = arrow(length = unit(0.3, "cm"))),
  annotate("text", label=expression(paste(1-beta, " (\\\"power\\\")")), x=6.1, y=0.21, parse=T, size=8) +
  # H_0 title
  annotate("text", label="H[0]", x=m1, y=0.28, parse=T, size=8) +
  # H_a title
  annotate("text", label="H[1]", x=m2, y=0.28, parse=T, size=8) +
  ggtitle("Statistical Power") +
  # remove some elements
  theme(
    panel.grid.minor = element_blank(),
    panel.grid.major = element_blank(),
    panel.background = element_blank(),
    plot.background = element_rect(fill="white"),
    panel.border = element_blank(),
    axis.line = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),

```

```
axis.ticks = element_blank(),
axis.title.x = element_blank(),
axis.title.y = element_blank(),
plot.title = element_text(size=22))
```

Statistical Power

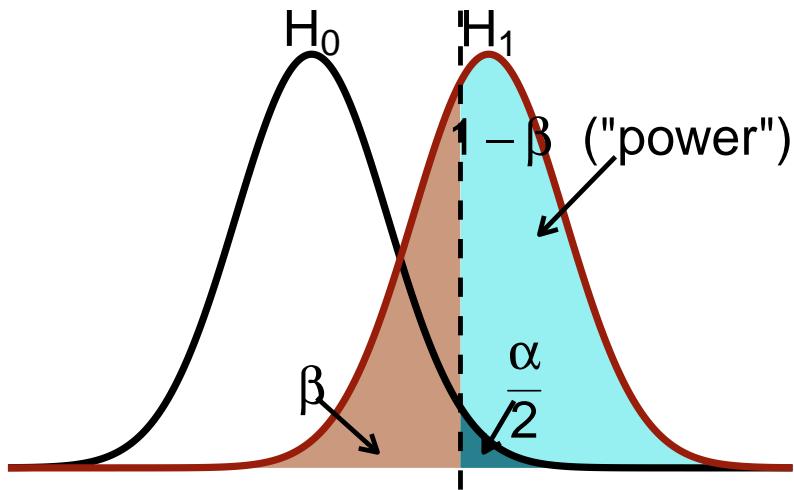


Figure 7.7 Statistical Power (Adapted from Kristoffer Magnusson: <https://rpsychologist.com/creating-a-typical-textbook-illustration-of-statistical-power-using-either-ggplot-or-base-graphics>; archived at <https://perma.cc/FG3J-85L6>). The dashed line represents the critical value or threshold.

Interactive visualizations by Kristoffer Magnusson on p -values and null-hypothesis significance testing are below:

- <https://rpsychologist.com/pvalue/> (archived at <https://perma.cc/JP9F-9ZVY>)
- <https://rpsychologist.com/d3/pdist/> (archived at <https://perma.cc/BE96-8LSJ>)
- <https://rpsychologist.com/d3/nhst/> (archived at <https://perma.cc/ZU9A-37F3>)

Twelve misconceptions about p -values (Goodman, 2008) are in Table 7.1.

Table 7.1 Twelve Misconceptions About p -Values from Goodman (2008). Goodman also provides a discussion about why each statement is false.

Misconception	Number
If $p = .05$, the null hypothesis has only a 5% chance of being true.	1
A nonsignificant difference (eg, $p > .05$) means there is no difference between groups.	2
A statistically significant finding is clinically important.	3
Studies with p -values on opposite sides of .05 are conflicting.	4
Studies with the same p -value provide the same evidence against the null hypothesis.	5
$p = .05$ means that we have observed data that would occur only 5% of the time under the null hypothesis.	6
$p = .05$ and $p < .05$ mean the same thing.	7
p -values are properly written as inequalities (e.g., " $p \leq .05$ " when $p = .015$).	8
$p = .05$ means that if you reject the null hypothesis, the probability of a Type I error is only 5%.	9
With a $p = .05$ threshold for significance, the chance of a Type I error will be 5%.	10
You should use a one-sided p -value when you don't care about a result in one direction, or a difference in that direction is impossible.	11
A scientific conclusion or treatment policy should be based on whether or not the p -value is significant.	12

That is, the p -value is not:

- the probability that the effect was due to chance
- the probability that the null hypothesis is true
- the size of the effect
- the importance of the effect
- whether the effect is true, real, or causal

Statistical significance involves the *consistency* of an effect/association/difference; it suggests that the association/difference is reliably non-zero. However, just because something is statistically significant does not mean that it is important. In addition to statistical significance, it is also important to consider practical significance.

7.4.2 Practical Significance

Practical significance deals with how large or important the effect/association/difference is. It is based on the magnitude of the effect,

called the *effect size*. Effect size can be quantified in various ways including:

- Cohen's d
- Standardized regression coefficient (beta; β)
- Correlation coefficient (r)
- Cohen's ω (omega)
- Cohen's f
- Cohen's f^2
- Coefficient of determination (R^2)
- Eta squared (η^2)
- Partial eta squared (η_p^2)

7.4.2.1 Cohen's d

Cohen's d is calculated as in Equation 7.11:

$$\begin{aligned} d &= \frac{\text{mean difference}}{\text{pooled standard deviation}} \\ &= \frac{\bar{X}_1 - \bar{X}_2}{s} \end{aligned} \quad (7.11)$$

where:

$$s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (7.12)$$

where n_1 and n_2 is the sample size of group 1 and group 2, respectively, and s_1 and s_2 is the standard deviation of group 1 and group 2, respectively.

7.4.2.2 Standardized Regression Coefficient (Beta; β)

The standardized regression coefficient (beta; β) is used in multiple regression, and is calculated as in Equation 7.13:

$$\beta_x = B_x \times \frac{s_x}{s_y} \quad (7.13)$$

where B_x is the unstandardized regression coefficient of the **predictor variable** x in predicting the **outcome variable** y , s_x is the standard deviation of x , and s_y is the standard deviation of y .

7.4.2.3 Correlation Coefficient (r)

The formula for the correlation coefficient is in Chapter 8.

7.4.2.4 Cohen's ω

Cohen's ω is used in chi-square tests, and is calculated as in Equation 7.14:

$$\omega = \sqrt{\frac{\chi^2}{N} - \frac{df}{N}} \quad (7.14)$$

where χ^2 is the chi-square statistic from the test, N is the sample size, and df is the degrees of freedom.

7.4.2.5 Cohen's f

Cohen's f is commonly used in ANOVA, and is calculated as in Equation 7.15:

$$\begin{aligned} f &= \sqrt{\frac{R^2}{1 - R^2}} \\ &= \sqrt{\frac{\eta^2}{1 - \eta^2}} \end{aligned} \quad (7.15)$$

7.4.2.6 Cohen's f^2

Cohen's f^2 is commonly used in regression, and is calculated as in Equation 7.16:

$$\begin{aligned} f^2 &= \frac{R^2}{1 - R^2} \\ &= \frac{\eta^2}{1 - \eta^2} \end{aligned} \quad (7.16)$$

To calculate the effect size of a particular predictor, you can calculate Δf^2 as in Equation 7.17:

$$\begin{aligned} \Delta f^2 &= \frac{R_{\text{model}}^2 - R_{\text{reduced}}^2}{1 - R_{\text{model}}^2} \\ &= \frac{\eta_{\text{model}}^2 - \eta_{\text{reduced}}^2}{1 - \eta_{\text{model}}^2} \end{aligned} \quad (7.17)$$

where R_{model}^2 is the R^2 of the model with the **predictor variable** of interest and R_{reduced}^2 is the R^2 of the model without the **predictor variable** of interest.

7.4.2.7 Coefficient of Determination (R^2)

The coefficient of determination (R^2) reflects the proportion of variance in the **outcome variable** that is explained by the **predictor variable(s)**. R^2 is commonly used in regression, and is calculated as in Equation 7.18:

$$\begin{aligned}
 R^2 &= 1 - \frac{\sum(Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y})^2} \\
 &= 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}} \\
 &= 1 - \frac{\text{sum of squared residuals}}{\text{total sum of squares}} \\
 &= \frac{f^2}{1 + f^2} \\
 &= \eta^2
 \end{aligned} \tag{7.18}$$

where Y_i is the observed value of the **outcome variable** for the i th observation, \hat{Y}_i is the model predicted value for the i th observation, \bar{Y} is the mean of the observed values of the **outcome variable**. The total sum of squares is an index of the total variation in the **outcome variable**.

7.4.2.8 Eta Squared (η^2) and Partial Eta Squared (η_p^2)

Like R^2 , eta squared (η^2) reflects the proportion of variance in the **dependent variable** that is explained by the **independent variable(s)**. η^2 is commonly used in ANOVA, and is calculated as in Equation 7.19:

$$\begin{aligned}
 \eta^2 &= \frac{SS_{\text{effect}}}{SS_{\text{total}}} \\
 &= 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}} \\
 &= 1 - \frac{\text{sum of squared residuals}}{\text{total sum of squares}} \\
 &= \frac{f^2}{1 + f^2} \\
 &= R^2
 \end{aligned} \tag{7.19}$$

where SS_{effect} is the sum of squares for the effect of interest and SS_{total} is the total sum of squares.

Partial eta squared (η_p^2) reflects the proportion of variance in the **dependent variable** that is explained by the **independent variable** while controlling for

the other **independent variables**. η_p^2 is commonly used in ANOVA, and is calculated as in Equation 7.20:

$$\eta_p^2 = \frac{SS_{\text{effect}}}{SS_{\text{effect}} + SS_{\text{error}}} \quad (7.20)$$

where SS_{effect} is the sum of squares for the effect of interest and $SS_{\{\text{error}\}}$ is the sum of squares for the residual error term.

7.4.2.9 Effect Size Thresholds

Effect size thresholds (McGrath & Meyer, 2006) for small, medium, and large effect sizes are in Table 7.2.

Table 7.2 Effect Size Thresholds for Small, Medium, and Large Effect Sizes.

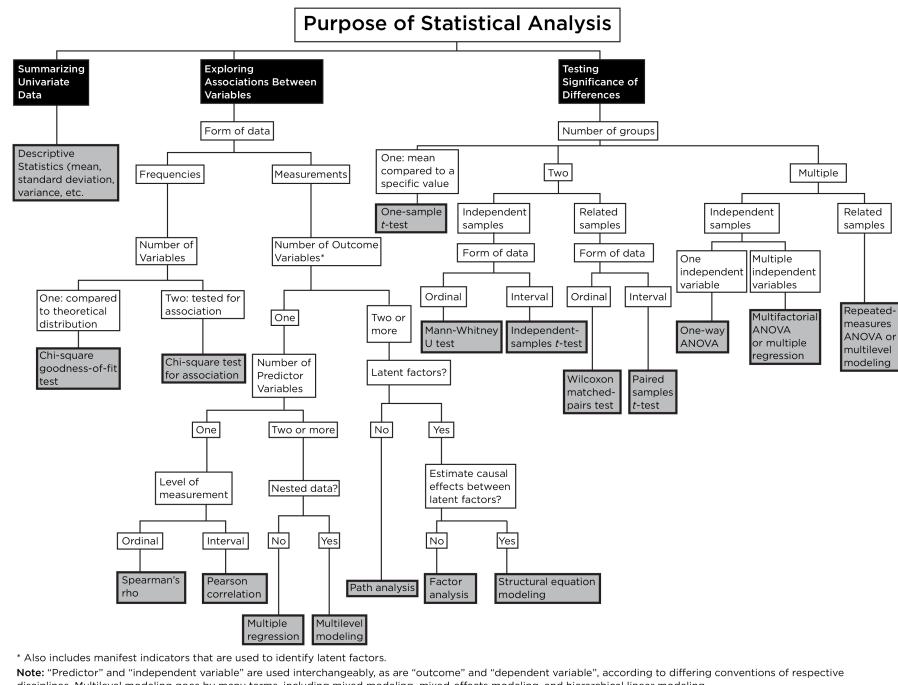
Effect Size Index	Small	Medium	Large
Cohen's d	$\geq .20 $	$\geq .50 $	$\geq .80 $
Standardized regression coefficient (beta; β)	$\geq .10 $	$\geq .24 $	$\geq .37 $
Correlation coefficient (r)	$\geq .10 $	$\geq .24 $	$\geq .37 $
Cohen's ω	$\geq .10$	$\geq .30$	$\geq .50$
Cohen's f	$\geq .10$	$\geq .25$	$\geq .40$
Cohen's f^2	$\geq .01$	$\geq .06$	$\geq .16$
Coefficient of determination (R^2)	$\geq .01$	$\geq .06$	$\geq .14$
Eta squared (η^2)	$\geq .01$	$\geq .06$	$\geq .14$
Partial eta squared (η_p^2)	$\geq .01$	$\geq .06$	$\geq .14$

7.5 Statistical Decision Tree

An example statistical decision tree is depicted in Figure 7.8.

However, many statistical tests can be re-formulated in a regression framework, as in Figure 7.9.

For an online, interactive statistical decision tree to help you decide which statistical analysis to use, see here: <https://www.statsflowchart.co.uk>



* Also includes manifest indicators that are used to identify latent factors.
 Note: "Predictor" and "independent variable" are used interchangeably, as are "outcome" and "dependent variable", according to differing conventions of respective disciplines. Multilevel modeling goes by many terms, including mixed modeling, mixed-effects modeling, and hierarchical linear modeling.

Figure 7.8 A Statistical Decision Tree For Choosing an Appropriate Statistical Procedure. Adapted from: <https://commons.wikimedia.org/wiki/File:InferentialStatisticalDecisionMakingTrees.pdf>. The original source is: Corston, R. & Colman, A. M. (2000). *A crash course in SPSS for Windows*. Wiley-Blackwell. Changes were made to the original, including the addition of several statistical tests.

7.6 Statistical Tests

7.6.1 *t*-Test

There are several *t*-tests:

- one-sample *t*-test
- two-samples *t*-test
 - independent samples *t*-test
 - paired samples *t*-test

A one-sample *t*-test is used to evaluate whether a sample mean differs systematically from a particular value. The null hypothesis is that the sample mean

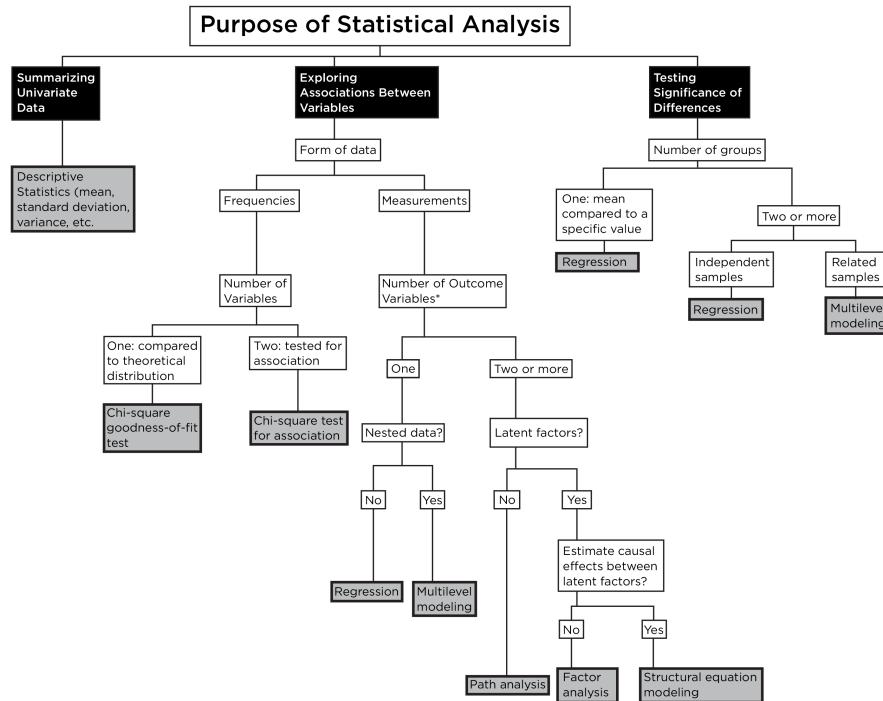


Figure 7.9 A Statistical Decision Tree For Choosing an Appropriate Statistical Procedure, Re-Formulated in a Regression Framework. Adapted from: <https://commons.wikimedia.org/wiki/File:InferentialStatisticalDecisionMakingTrees.pdf>. The original source is: Corston, R. & Colman, A. M. (2000). *A crash course in SPSS for Windows*. Wiley-Blackwell. Changes were made to the original, including re-formulating the tests in a regression framework.

does not differ systematically from the pre-specified value. The alternative hypothesis is that the sample mean differs systematically from the pre-specified value. For instance, let's say you want to test out a new draft strategy. You could participate in a mock draft and draft players using the new strategy. Then, you could use a one-sample t -test to evaluate whether your new draft strategy yields players with more projected points than the average of players' projected points for other teams.

Two-samples t -tests are used to test for differences between scores of two groups. If the two groups are independent, the independent samples t -test is used. If the two groups involve paired samples, the paired samples t -test is used. The null hypothesis is that the mean of group 1 does not differ sys-

tematically from the mean of group 2. The alternative hypothesis is that the mean of group 1 differs systematically from the mean of group 2. For instance, you could use an independent-samples t -test if you want to examine whether Quarterbacks tend to have longer careers than Running Backs. By contrast, you could use a paired samples t -test if you want to examine whether Quarterbacks tend to score more points in the second year of their contract compared to their rookie year, because the same subjects were assessed twice (i.e., a [within-subject design](#)).

7.6.2 Analysis of Variance

Analysis of variance (ANOVA) allows examining whether groups differ systematically as a function of one or more factors. There are multiple variants of ANOVA:

- one-way ANOVA
- factorial ANOVA
- repeated measures ANOVA (RM-ANOVA)
- multivariate ANOVA (MANOVA)

Like two-samples t -tests, ANOVA allows examining whether groups differ as a function of an [independent variable](#). However, unlike a t -test, ANOVA allows examining multiple [independent variables](#) and more than two groups. The null hypothesis is that the groups' mean value does not differ systematically. The alternative hypothesis is that the groups' mean value differs systematically.

A one-way ANOVA examines whether two or more groups differ as a function of an [independent variable](#). For instance, you could use a one-way ANOVA to evaluate if you want to evaluate whether multiple positions differ in their length of career. Factorial ANOVA examines whether two or more groups differ as a function of multiple [independent variables](#). For instance, you could use factorial ANOVA to evaluate whether one's length of career depends on one's position and weight. Repeated measures ANOVA examines whether scores differ across repeated measures (e.g., across time) for the same participants. For instance, you could use repeated-measures ANOVA to evaluate whether rookies score more points as the season progresses. Multivariate ANOVA examines whether multiple [dependent variables](#) differ as a function of one or more factor(s). For instance, you could use MANOVA to evaluate whether one's contract length and pay differ as a function of one's position.

7.6.3 Correlation

Correlation examines the association between a **predictor** and **outcome** variable. The null hypothesis is that the two variables are not associated. The alternative hypothesis is that the two variables are associated.

The Pearson correlation coefficient (r) is calculated as in Equation 7.21:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \quad (7.21)$$

where X is the **predictor variable** and Y is the **outcome variable**.

7.6.4 (Multiple) Regression

Regression, like correlation, examines the association between a **predictor** and **outcome** variable. However, unlike correlation, regression allows multiple **predictor variables**.

Regression with a single predictor takes the form in Equation 9.1. A regression line is depicted in Figure 9.3. Multiple regression (i.e., regression with multiple predictors) takes the form in Equation 9.2.

The null hypothesis is that the **predictor variable(s)** are not associated with the **outcome variable**. The alternative hypothesis is that the **predictor variable(s)** are associated with the **outcome variable**.

7.6.5 Chi-Square Test

There are two primary types of chi-square tests:

- chi-square goodness-of-fit test
- chi-square test for association (aka test of independence)

The chi-square goodness-of-fit test evaluates whether a set of categorical data came from a specified distribution. The null hypothesis is that the data came from the specified distribution. The alternative hypothesis is that the data did not come from the specified distribution.

The chi-square test for association evaluates whether two categorical variables are associated. The null hypothesis is that the two variables are not associated. The alternative hypothesis is that the two variables are associated.

7.6.6 Formulating Statistical Tests in Terms of Partitioned Variance

Many statistical tests can be formulated in terms of partitioned variance.

For instance, the t statistic from the independent-samples t -test and the F statistic from ANOVA can be thought of as the ratio of between-group variance to within-group variance, as in Equation 7.22:

$$t \text{ or } F = \frac{\text{between-group variance}}{\text{within-group variance}} \quad (7.22)$$

The correlation coefficient can be thought of as the ratio of shared variance (i.e., covariance) to total variance, as in Equation 7.23:

$$r = \frac{\text{shared variance}}{\text{total variance}} \quad (7.23)$$

The coefficient of determination (R^2) is the proportion of variance in the **outcome variable** that is explained by the **predictor variables**. η^2 is the proportion of variance in the **dependent variable** that is explained by the **independent variables**. The coefficient of determination and η^2 can be expressed as the ratio of variance explained in the **outcome** or **dependent** variable to the total variance in the **outcome** or **dependent** variable, as in Equation 7.24:

$$R^2 \text{ or } \eta^2 = \frac{\text{variance explained in the outcome variable}}{\text{total variance in the outcome variable}} \quad (7.24)$$

7.6.7 Critical Value

The critical value is the test value for a given test, above which the effect is considered to be **statistically significant**. The critical value for **statistical significance** for each test can be determined based on the degrees of freedom and alpha level. The degrees of freedom (df) refer to the number of values in the calculation of a test statistic that are free to vary.

```
alpha <- .05
N <- 200
nGroup1 <- 150
nGroup2 <- 150
numGroups <- 4
numLevelsFactorA <- 3
numLevelsFactorB <- 4
numMeasurements <- 4
numPredictors <- 5
numCategories <- 6
```

```
numRows <- 5
numColumns <- 2
```

7.6.7.1 One-Sample t -Test

For a one-sample t -test, the degrees of freedom is in Equation 7.25:

$$df = N - 1 \quad (7.25)$$

where N is sample size.

```
df_oneSampleTtest <- N - 1
```

One-tailed test:

```
qt(1 - alpha, df_oneSampleTtest)
```

[1] 1.652547

Two-tailed test:

```
qt(1 - alpha/2, df_oneSampleTtest)
```

[1] 1.971957

7.6.7.2 Independent-Samples t -Test

For an independent-samples t -test, the degrees of freedom is in Equation 7.26:

$$df = n_1 + n_2 - 2 \quad (7.26)$$

where n_1 is the sample size of group 1 and n_2 is the sample size of group 2.

```
df_independentSamplesTtest <- nGroup1 + nGroup2 - 2
```

One-tailed test:

```
qt(1 - alpha, df_independentSamplesTtest)
```

[1] 1.649983

Two-tailed test:

```
qt(1 - alpha/2, df_independentSamplesTtest)
```

```
[1] 1.967957
```

7.6.7.3 Paired-Samples *t*-Test

For a paired-samples *t*-test, the degrees of freedom is in Equation 7.27:

$$df = N - 1 \quad (7.27)$$

where N is sample size (i.e., the number of paired observations).

```
df_pairedSamplesTtest <- N - 1
```

One-tailed test:

```
qt(1 - alpha, df_pairedSamplesTtest)
```

```
[1] 1.652547
```

Two-tailed test:

```
qt(1 - alpha/2, df_pairedSamplesTtest)
```

```
[1] 1.971957
```

7.6.7.4 One-Way ANOVA

For a one-way ANOVA, the degrees of freedom is in Equation 7.28:

$$\begin{aligned} df_{\text{between}} &= g - 1 \\ df_{\text{within}} &= N - g \end{aligned} \quad (7.28)$$

where N is sample size and g is the number of groups.

```
df_betweenOneWayANOVA <- numGroups - 1  
df_withinOneWayANOVA <- N - numGroups
```

One-tailed test:

```
qf(1 - alpha, df_betweenOneWayANOVA, df_withinOneWayANOVA)
```

```
[1] 2.650677
```

Two-tailed test:

```
qf(1 - alpha/2, df_betweenOneWayANOVA, df_withinOneWayANOVA)
```

```
[1] 3.183378
```

7.6.7.5 Factorial ANOVA

For a factorial two-way ANOVA, the degrees of freedom is in Equation 7.29:

$$\begin{aligned} df_{\text{Factor A}} &= a - 1 \\ df_{\text{Factor B}} &= b - 1 \\ df_{\text{Interaction}} &= (a - 1)(b - 1) \\ df_{\text{error}} &= ab(N - 1) \end{aligned} \tag{7.29}$$

where N is sample size, a is the number of levels for factor A, and b is the number of levels for factor B.

```
df_factorA <- numLevelsFactorA - 1
df_factorB <- numLevelsFactorB - 1
df_interaction <- df_factorA * df_factorB
df_error <- numLevelsFactorA * numLevelsFactorB * (N - 1)
```

Factor A (one-tailed test):

```
qf(1 - alpha, df_factorA, df_error)
```

```
[1] 2.999494
```

Factor B (one-tailed test):

```
qf(1 - alpha, df_factorB, df_error)
```

```
[1] 2.608629
```

Interaction (one-tailed test):

```
qf(1 - alpha, df_interaction, df_error)
```

[1] 2.102376

Factor A (two-tailed test):

```
qf(1 - alpha/2, df_factorA, df_error)
```

[1] 3.694584

Factor B (two-tailed test):

```
qf(1 - alpha/2, df_factorB, df_error)
```

[1] 3.121587

Interaction (two-tailed test):

```
qf(1 - alpha/2, df_interaction, df_error)
```

[1] 2.413504

7.6.7.6 Repeated Measures ANOVA

For a repeated measures ANOVA, the degrees of freedom is in Equation 7.30:

$$\begin{aligned} df_1 &= T - 1 \\ df_2 &= (T - 1)(N - 1) \end{aligned} \tag{7.30}$$

where N is sample size and T is the number of measurements (i.e., the number of levels of the within-person factor: e.g., timepoints or conditions).

```
df1_RMANOVA <- numMeasurements - 1
df2_RMANOVA <- (numMeasurements - 1) * (N - 1)
```

One-tailed test:

```
qf(1 - alpha, df1_RMANOVA, df2_RMANOVA)
```

[1] 2.619828

Two-tailed test:

```
qf(1 - alpha/2, df1_RMANOVA, df2_RMANOVA)
```

```
[1] 3.138017
```

7.6.7.7 Correlation

For a correlation, the degrees of freedom is in Equation 7.31:

$$df = N - 2 \quad (7.31)$$

where N is sample size.

```
df_correlation <- N - 2
```

One-tailed test:

```
qt(1 - alpha, df_correlation)
```

```
[1] 1.652586
```

Two-tailed test:

```
qt(1 - alpha/2, df_correlation)
```

```
[1] 1.972017
```

7.6.7.8 Multiple Regression

For multiple regression, the degrees of freedom is in Equation 7.32:

$$\begin{aligned} df_1 &= p \\ df_2 &= N - p - 1 \end{aligned} \quad (7.32)$$

where N is sample size and p is the number of predictors.

```
df1_regression <- numPredictors
df2_regression <- N - numPredictors - 1
```

One-tailed test:

```
qf(1 - alpha, df1_regression, df2_regression)
```

```
[1] 2.260647
```

Two-tailed test:

```
qf(1 - alpha/2, df1_regression, df2_regression)
```

```
[1] 2.63243
```

7.6.7.9 Chi-Square Goodness-of-Fit Test

For the chi-square goodness-of-fit test, the degrees of freedom is in Equation 7.33:

$$df = c - 1 \quad (7.33)$$

where c is the number of categories.

```
df_chisquareGOF <- numCategories - 1
```

One-tailed test:

```
qchisq(1 - alpha, df_chisquareGOF)
```

```
[1] 11.0705
```

Two-tailed test:

```
qchisq(1 - alpha/2, df_chisquareGOF)
```

```
[1] 12.8325
```

7.6.7.10 Chi-Square Test for Association

For the chi-square test for association, the degrees of freedom is in Equation 7.34:

$$df = (r - 1) \times (c - 1) \quad (7.34)$$

where r is the number of rows in the contingency table and c is the number of columns in the contingency table.

```
df_chisquareAssociation <- (numRows - 1) * (numColumns - 1)
```

One-tailed test:

```
qchisq(1 - alpha, df_chisquareAssociation)
```

```
[1] 9.487729
```

Two-tailed test:

```
qchisq(1 - alpha/2, df_chisquareAssociation)
```

```
[1] 11.14329
```

7.6.8 Statistical Power

As described above, *statistical power* is the probability of detecting an effect, if, in fact, the effect exists. Statistical power for a given test can be calculated based on three factors:

- **effect size**
- sample size
- **alpha level**

Knowing any three of the following, you can calculate the fourth: statistical power, **effect size**, sample size, and **alpha level**. Below is R code for calculating power for each of various statistical tests (i.e., a *power analysis*). For free point-and-click software for calculating statistical power, see G*Power: <https://www.psychologie.hhu.de/arbeitsgruppen/allgemeine-psychologie-und-arbeitspsychologie/gpower.html>

```
power <- .8
effectSize_d <- .5
effectSize_r <- .24
effectSize_beta <- .24
effectSize_f <- .25
effectSize_fSquared <- .06
effectSize_omega <- .3
```

7.6.8.1 One-Sample *t*-Test

Solving for statistical power achieved (given **effect size**, sample size, and **alpha level**):

```
pwr::pwr.t.test(  
  n = N,  
  d = effectSize_d,  
  sig.level = alpha,  
  type = "one.sample",  
  alternative = "two.sided")
```

```
One-sample t test power calculation
```

```
  n = 200  
  d = 0.5  
  sig.level = 0.05  
  power = 0.9999998  
  alternative = two.sided
```

Solving for sample size needed (given **effect size**, power, and **alpha level**):

```
pwr::pwr.t.test(  
  power = power,  
  d = effectSize_d,  
  sig.level = alpha,  
  type = "one.sample",  
  alternative = "two.sided")
```

```
One-sample t test power calculation
```

```
  n = 33.36713  
  d = 0.5  
  sig.level = 0.05  
  power = 0.8  
  alternative = two.sided
```

Solving for the minimum detectable **effect size** (given sample size, power, and **alpha level**):

```
pwr::pwr.t.test(
  power = power,
  n = N,
  sig.level = alpha,
  type = "one.sample",
  alternative = "two.sided")
```

```
One-sample t test power calculation

  n = 200
  d = 0.1990655
  sig.level = 0.05
  power = 0.8
  alternative = two.sided
```

7.6.8.2 Independent-Samples *t*-Test

7.6.8.2.1 Balanced Group Sizes

Solving for statistical power achieved (given [effect size](#), sample size per group, and [alpha level](#)):

```
pwr::pwr.t.test(
  n = N,
  d = effectSize_d,
  sig.level = alpha,
  type = "two.sample",
  alternative = "two.sided")
```

```
Two-sample t test power calculation

  n = 200
  d = 0.5
  sig.level = 0.05
  power = 0.9987689
  alternative = two.sided
```

NOTE: *n* is number in *each* group

Solving for sample size per group needed (given [effect size](#), power, and [alpha level](#)):

```
pwr::pwr.t.test(  
  power = power,  
  d = effectSize_d,  
  sig.level = alpha,  
  type = "two.sample",  
  alternative = "two.sided")
```

Two-sample t test power calculation

```
n = 63.76561  
d = 0.5  
sig.level = 0.05  
power = 0.8  
alternative = two.sided
```

NOTE: n is number in *each* group

Solving for the minimum detectable **effect size** (given sample size per group, power, and **alpha level**):

```
pwr::pwr.t.test(  
  power = power,  
  n = N,  
  sig.level = alpha,  
  type = "two.sample",  
  alternative = "two.sided")
```

Two-sample t test power calculation

```
n = 200  
d = 0.2808267  
sig.level = 0.05  
power = 0.8  
alternative = two.sided
```

NOTE: n is number in *each* group

7.6.8.2.2 Unbalanced Group Sizes

Solving for statistical power achieved (given **effect size**, sample size per group, and **alpha level**):

```
pwr::pwr.t2n.test(  
  n1 = nGroup1,  
  n2 = nGroup2,  
  d = effectSize_d,  
  sig.level = alpha,  
  alternative = "two.sided")
```

```
t test power calculation  
  
  n1 = 150  
  n2 = 150  
  d = 0.5  
  sig.level = 0.05  
  power = 0.9907677  
  alternative = two.sided
```

Solving for sample size per group needed (given **effect size**, power, and **alpha level**):

```
pwr::pwr.t2n.test(  
  power = power,  
  n1 = nGroup1,  
  d = effectSize_d,  
  sig.level = alpha,  
  alternative = "two.sided")
```

```
t test power calculation  
  
  n1 = 150  
  n2 = 40.22483  
  d = 0.5  
  sig.level = 0.05  
  power = 0.8  
  alternative = two.sided
```

Solving for the minimum detectable **effect size** (given sample size per group, power, and **alpha level**):

```
pwr::pwr.t2n.test(  
  power = power,  
  n1 = nGroup1,
```

```
n2 = nGroup2,  
sig.level = alpha,  
alternative = "two.sided")
```

```
t test power calculation  
  
n1 = 150  
n2 = 150  
d = 0.3245459  
sig.level = 0.05  
power = 0.8  
alternative = two.sided
```

7.6.8.3 Paired-Samples *t*-Test

Solving for statistical power achieved (given **effect size**, sample size per group, and **alpha level**):

```
pwr::pwr.t.test(  
  n = N,  
  d = effectSize_d,  
  sig.level = alpha,  
  type = "paired",  
  alternative = "two.sided")
```

```
Paired t test power calculation  
  
n = 200  
d = 0.5  
sig.level = 0.05  
power = 0.9999998  
alternative = two.sided
```

NOTE: n is number of *pairs*

Solving for sample size per group needed (given **effect size**, power, and **alpha level**):

```
pwr::pwr.t.test(  
  power = power,  
  d = effectSize_d,
```

```
sig.level = alpha,
type = "paired",
alternative = "two.sided")
```

Paired t test power calculation

```
n = 33.36713
d = 0.5
sig.level = 0.05
power = 0.8
alternative = two.sided
```

NOTE: n is number of *pairs*

Solving for the minimum detectable **effect size** (given sample size per group, power, and **alpha level**):

```
pwr:::pwr.t.test(
  power = power,
  n = N,
  sig.level = alpha,
  type = "paired",
  alternative = "two.sided")
```

Paired t test power calculation

```
n = 200
d = 0.1990655
sig.level = 0.05
power = 0.8
alternative = two.sided
```

NOTE: n is number of *pairs*

7.6.8.4 One-Way ANOVA

Solving for statistical power achieved (given **effect size**, sample size per group, and **alpha level**):

```
pwr::pwr.anova.test(  
  n = N,  
  f = effectSize_f,  
  sig.level = alpha,  
  k = numGroups)
```

Balanced one-way analysis of variance power calculation

```
k = 4  
n = 200  
f = 0.25  
sig.level = 0.05  
power = 0.9999962
```

NOTE: n is number in each group

Solving for sample size per group needed (given [effect size](#), power, and [alpha level](#)):

```
pwr::pwr.anova.test(  
  power = power,  
  f = effectSize_f,  
  sig.level = alpha,  
  k = numGroups)
```

Balanced one-way analysis of variance power calculation

```
k = 4  
n = 44.59927  
f = 0.25  
sig.level = 0.05  
power = 0.8
```

NOTE: n is number in each group

Solving for the minimum detectable [effect size](#) (given sample size per group, power, and [alpha level](#)):

```
pwr::pwr.anova.test(  
  power = power,  
  n = N,  
  sig.level = alpha,  
  k = numGroups)
```

```
Balanced one-way analysis of variance power calculation

k = 4
n = 200
f = 0.117038
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

The power analysis code above assumes the groups are of equal size (i.e., a balanced design). If the design is unbalanced (i.e., there are different numbers of participants in each group), it may be necessary to conduct a power analysis via a simulation. Below is an example of evaluating the statistical power for detecting an effect unbalanced designs via simulation:

```
nSim <- 1000 # number of simulations

# Function to generate data and perform ANOVA
simulate_anova <- function(nGroup1, nGroup2, f, alpha) {
  # Means for each group
  mean1 <- 0
  mean2 <- f * sqrt((nGroup1 + nGroup2) / 2)

  # Generate data
  group1 <- rnorm(nGroup1, mean = mean1, sd = 1)
  group2 <- rnorm(nGroup2, mean = mean2, sd = 1)

  # Combine data
  data <- data.frame(
    value = c(group1, group2),
    group = factor(rep(c("Group1", "Group2"), c(nGroup1, nGroup2)))
  )

  # Perform ANOVA
  aov_result <- aov(value ~ group, data = data)
  p_value <- summary(aov_result)[[1]][["Pr(>F)"]][1]

  # Check if p-value is less than alpha
  return(p_value < alpha)
}

# Run simulations
set.seed(52242) # for reproducibility
```

```
powerSimulationOneWayAnova <- replicate(
  nSim,
  simulate_anova(
    nGroup1 = 10,
    nGroup2 = 25,
    f = effectSize_f,
    alpha = alpha))

# Estimate power
mean(powerSimulationOneWayAnova)
```

```
[1] 0.774
```

7.6.8.5 Factorial ANOVA

The power analysis code below assumes the groups are of equal size (i.e., a balanced design). If the design is unbalanced (i.e., there are different numbers of participants in each group), it may be necessary to conduct a power analysis via a simulation. See Section 7.6.8.4 for an example power analysis simulation for one-way ANOVA.

Solving for statistical power achieved (given `effect size`, sample size per group, and `alpha level`):

```
pwr::pwr.anova.test(
  n = N,
  f = effectSize_f,
  sig.level = alpha,
  k = numLevelsFactorA)
```

```
Balanced one-way analysis of variance power calculation
```

```
k = 3
n = 200
f = 0.25
sig.level = 0.05
power = 0.9999238
```

```
NOTE: n is number in each group
```

```
pwr::pwr.anova.test(
  n = N,
  f = effectSize_f,
```

```
sig.level = alpha,
k = numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 4
n = 200
f = 0.25
sig.level = 0.05
power = 0.9999962
```

NOTE: n is number in each group

```
pwr::pwr.anova.test(
  n = N,
  f = effectSize_f,
  sig.level = alpha,
  k = numLevelsFactorA + numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 7
n = 200
f = 0.25
sig.level = 0.05
power = 1
```

NOTE: n is number in each group

Solving for sample size per group needed (given [effect size](#), power, and [alpha level](#)):

```
pwr::pwr.anova.test(
  power = power,
  f = effectSize_f,
  sig.level = alpha,
  k = numLevelsFactorA)
```

Balanced one-way analysis of variance power calculation

```
k = 3
n = 52.3966
f = 0.25
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

```
pwr::pwr.anova.test(
  power = power,
  f = effectSize_f,
  sig.level = alpha,
  k = numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 4
n = 44.59927
f = 0.25
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

```
pwr::pwr.anova.test(
  power = power,
  f = effectSize_f,
  sig.level = alpha,
  k = numLevelsFactorA + numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 7
n = 32.05196
f = 0.25
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

Solving for the minimum detectable **effect size** (given sample size per group, power, and **alpha level**):

```
pwr::pwr.anova.test(  
  power = power,  
  n = N,  
  sig.level = alpha,  
  k = numLevelsFactorA)
```

Balanced one-way analysis of variance power calculation

```
k = 3  
n = 200  
f = 0.1270373  
sig.level = 0.05  
power = 0.8
```

NOTE: n is number in each group

```
pwr::pwr.anova.test(  
  power = power,  
  n = N,  
  sig.level = alpha,  
  k = numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 4  
n = 200  
f = 0.117038  
sig.level = 0.05  
power = 0.8
```

NOTE: n is number in each group

```
pwr::pwr.anova.test(  
  power = power,  
  n = N,  
  sig.level = alpha,  
  k = numLevelsFactorA + numLevelsFactorB)
```

Balanced one-way analysis of variance power calculation

```
k = 7
n = 200
f = 0.09889082
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

7.6.8.6 Repeated Measures ANOVA

Solving for statistical power achieved (given **effect size**, sample size per group, and **alpha level**):

```
WebPower::wp.rmanova(
  n = N,
  ng = numGroups,
  nm = numMeasurements,
  f = effectSize_f,
  alpha = alpha,
  type = 0)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.25	4	4	1	0.05	0.8484718

NOTE: Power analysis for between-effect test

URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(
  n = N,
  ng = numGroups,
  nm = numMeasurements,
  f = effectSize_f,
  alpha = alpha,
  type = 1)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.25	4	4	1	0.05	0.8536292

NOTE: Power analysis for within-effect test

URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(
  n = N,
  ng = numGroups,
  nm = numMeasurements,
  f = effectSize_f,
  alpha = alpha,
  type = 2)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.25	4	4	1	0.05	0.6756298

NOTE: Power analysis for interaction-effect test
URL: <http://psychstat.org/rmanova>

Solving for sample size per group needed (given **effect size**, power, and **alpha level**):

```
WebPower::wp.rmanova(
  power = power,
  ng = numGroups,
  nm = numMeasurements,
  f = effectSize_f,
  alpha = alpha,
  type = 0)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
178.3971	0.25	4	4	1	0.05	0.8

NOTE: Power analysis for between-effect test
URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(
  power = power,
  ng = numGroups,
  nm = numMeasurements,
  f = effectSize_f,
  alpha = alpha,
  type = 1)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
175.7692	0.25	4	4	1	0.05	0.8

NOTE: Power analysis for within-effect test

URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(  
  power = power,  
  ng = numGroups,  
  nm = numMeasurements,  
  f = effectSize_f,  
  alpha = alpha,  
  type = 2)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
253.2369	0.25	4	4	1	0.05	0.8

NOTE: Power analysis for interaction-effect test

URL: <http://psychstat.org/rmanova>

Solving for the minimum detectable **effect size** (given sample size per group, power, and **alpha level**):

```
WebPower::wp.rmanova(  
  power = power,  
  n = N,  
  ng = numGroups,  
  nm = numMeasurements,  
  alpha = alpha,  
  type = 0)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.2358259	4	4	1	0.05	0.8

NOTE: Power analysis for between-effect test

URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(
  power = power,
  n = N,
  ng = numGroups,
  nm = numMeasurements,
  alpha = alpha,
  type = 1)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.2342726	4	4	1	0.05	0.8

NOTE: Power analysis for within-effect test
URL: <http://psychstat.org/rmanova>

```
WebPower::wp.rmanova(
  power = power,
  n = N,
  ng = numGroups,
  nm = numMeasurements,
  alpha = alpha,
  type = 2)
```

Repeated-measures ANOVA analysis

n	f	ng	nm	nscor	alpha	power
200	0.2817486	4	4	1	0.05	0.8

NOTE: Power analysis for interaction-effect test
URL: <http://psychstat.org/rmanova>

7.6.8.7 Correlation

Solving for statistical power achieved (given **effect size**, sample size per group, and **alpha level**):

```
pwr::pwr.r.test(
  n = N,
  r = effectSize_r,
  sig.level = alpha,
  alternative = "two.sided")
```

```
approximate correlation power calculation (arctangh transformation)

n = 200
r = 0.24
sig.level = 0.05
power = 0.9310138
alternative = two.sided
```

Solving for sample size per group needed (given [effect size](#), power, and [alpha level](#)):

```
pwr:::pwr.r.test(
  power = power,
  r = effectSize_r,
  sig.level = alpha,
  alternative = "two.sided")
```

```
approximate correlation power calculation (arctangh transformation)

n = 133.1299
r = 0.24
sig.level = 0.05
power = 0.8
alternative = two.sided
```

Solving for the minimum detectable [effect size](#) (given sample size per group, power, and [alpha level](#)):

```
pwr:::pwr.r.test(
  power = power,
  n = N,
  sig.level = alpha,
  alternative = "two.sided")
```

```
approximate correlation power calculation (arctangh transformation)

n = 200
r = 0.1965767
sig.level = 0.05
power = 0.8
alternative = two.sided
```

7.6.8.8 Multiple Regression

Solving for statistical power achieved (given **effect size**, sample size, and **alpha level**):

```
pwr::pwr.f2.test(
  f2 = effectSize_fSquared,
  sig.level = alpha,
  u = numPredictors,
  v = N - numPredictors - 1)
```

Multiple regression power calculation

```
u = 5
v = 194
f2 = 0.06
sig.level = 0.05
power = 0.7548031
```

```
pwrss::pwrss.t.reg(
  n = N,
  beta1 = effectSize_beta,
  k = numPredictors,
  alpha = alpha,
  alternative = "not equal")
```

```
Linear Regression Coefficient (t Test)
H0: beta1 = beta0
HA: beta1 != beta0
-----
Statistical power = 0.936
n = 200
-----
Alternative = "not equal"
Degrees of freedom = 194
Non-centrality parameter = 3.496
Type I error rate = 0.05
Type II error rate = 0.064
```

Solving for sample size needed (given **effect size**, power, and **alpha level**)—
 $v = N - \text{numberOfPredictors} - 1$; thus, $N = v + \text{numberOfPredictors} + 1$:

```
multipleRegressionSampleSizeModel <- pwr::pwr.f2.test(  
  power = power,  
  f2 = effectSize_fSquared,  
  sig.level = alpha,  
  u = numPredictors)  
  
multipleRegressionSampleSizeModel
```

Multiple regression power calculation

```
u = 5  
v = 213.3947  
f2 = 0.06  
sig.level = 0.05  
power = 0.8  
  
vNeeded <- multipleRegressionSampleSizeModel$v  
sampleSizeNeeded <- vNeeded + numPredictors + 1  
sampleSizeNeeded
```

[1] 219.3947

```
pwrss::pwrss.t.reg(  
  power = power,  
  beta1 = effectSize_beta,  
  k = numPredictors,  
  alpha = alpha,  
  alternative = "not equal")
```

```
Linear Regression Coefficient (t Test)  
H0: beta1 = beta0  
HA: beta1 != beta0  
-----  
Statistical power = 0.8  
n = 131  
-----  
Alternative = "not equal"  
Degrees of freedom = 124.427  
Non-centrality parameter = 2.823  
Type I error rate = 0.05  
Type II error rate = 0.2
```

Solving for the minimum detectable **effect size** (given sample size, power, and **alpha level**):

```
pwr::pwr.f2.test(
  power = power,
  sig.level = alpha,
  u = numPredictors,
  v = N - numPredictors - 1)
```

Multiple regression power calculation

```
u = 5
v = 194
f2 = 0.06597765
sig.level = 0.05
power = 0.8
```

7.6.8.9 Chi-Square Goodness-of-Fit Test

Solving for statistical power achieved (given **effect size**, sample size, and **alpha level**):

```
pwr::pwr.chisq.test(
  N = N,
  w = effectSize_omega,
  df = numCategories - 1,
  sig.level = alpha)
```

Chi squared power calculation

```
w = 0.3
N = 200
df = 5
sig.level = 0.05
power = 0.9269225
```

NOTE: N is the number of observations

Solving for sample size needed (given **effect size**, power, and **alpha level**):

```
pwr::pwr.chisq.test(  
  power = power,  
  w = effectSize_omega,  
  df = numCategories - 1,  
  sig.level = alpha)
```

Chi squared power calculation

```
w = 0.3  
N = 142.529  
df = 5  
sig.level = 0.05  
power = 0.8
```

NOTE: N is the number of observations

Solving for the minimum detectable [effect size](#) (given sample size, power, and [alpha level](#)):

```
pwr::pwr.chisq.test(  
  power = power,  
  N = N,  
  df = numCategories - 1,  
  sig.level = alpha)
```

Chi squared power calculation

```
w = 0.2532543  
N = 200  
df = 5  
sig.level = 0.05  
power = 0.8
```

NOTE: N is the number of observations

7.6.8.10 Chi-Square Test for Association

Solving for statistical power achieved (given [effect size](#), sample size, and [alpha level](#)):

```
pwr::pwr.chisq.test(
  N = N,
  w = effectSize_omega,
  df = ( numRows - 1)*( numColumns - 1),
  sig.level = alpha)
```

```
Chi squared power calculation

      w = 0.3
      N = 200
      df = 4
sig.level = 0.05
power = 0.9431195
```

NOTE: N is the number of observations

Solving for sample size needed (given effect size, power, and alpha level):

```
pwr::pwr.chisq.test(
  power = power,
  w = effectSize_omega,
  df = ( numRows - 1)*( numColumns - 1),
  sig.level = alpha)
```

```
Chi squared power calculation

      w = 0.3
      N = 132.6143
      df = 4
sig.level = 0.05
power = 0.8
```

NOTE: N is the number of observations

Solving for the minimum detectable effect size (given sample size, power, and alpha level):

```
pwr::pwr.chisq.test(
  power = power,
  N = N,
  df = ( numRows - 1)*( numColumns - 1),
  sig.level = alpha)
```

```
Chi squared power calculation

w = 0.2442875
N = 200
df = 4
sig.level = 0.05
power = 0.8
```

NOTE: N is the number of observations

7.6.8.11 Multilevel Modeling

Power analysis for multilevel modeling approaches is more complicated than it is for other statistical analyses, such as [correlation](#), multiple regression, [t-tests](#), ANOVA¹, etc.

There are free web applications for calculating power in multilevel modeling:

- https://aguinis.shinyapps.io/ml_power/
- https://koumurrayama.shinyapps.io/tmethod_mlm/
- <https://webpower.psychstat.org/wiki/models/index>

7.6.8.12 Path Analysis, Factor Analysis, and Structural Equation Modeling

Power analysis for latent variable modeling approaches like structural equation modeling (SEM) is more complicated than it is for other statistical analyses, such as [correlation](#), multiple regression, [t-tests](#), ANOVA², etc.

I provide an example of power analysis in SEM using Monte Carlo simulation in R here: <https://isaactpetersen.github.io/Principles-Psychological-Assessment/sem.html#monteCarloPowerAnalysis> (Petersen, 2024c).

There are also free web applications for calculating power in SEM:

- <https://sjak.shinyapps.io/power4SEM/>
- <https://sempower.shinyapps.io/sempower/>
- <https://yilinandrewang.shinyapps.io/pwrSEM/>
- <https://webpower.psychstat.org/wiki/models/index>

¹@sec-anova

²@sec-anova

7.6.8.13 Mediation and Moderation

There are free tools for calculating power for tests of mediation and moderation:

- https://schoemanna.shinyapps.io/mc_power_med/
- <https://www.causalevaluation.org/power-analysis.html> (web application:
<https://powerupr.shinyapps.io/index/>)
- <https://webpower.psychstat.org/wiki/models/index>

8

Correlation Analysis

8.1 Getting Started

8.1.1 Load Packages

```
library("petersenlab")
library("XICOR")
```

8.2 Overview of Correlation

Correlation is a metric of the association between variables. Covariance is the association between variables and is an unstandardized metric that differs for variables with different scales. By contrast, correlation is a standarized metric that does not differ for variables with different scales. When examining the association between variables that are [interval](#) or [ratio](#) levels of measurement, Pearson correlation is used. When examining the association between variables that are [ordinal](#) in level of measurement, Spearman correlation is used. Pearson correlation is an index of the *linear* association between variables. If a nonlinear association is present, other indices like ξ [Chatterjee (2021)] and distance correlation coefficients are better suited to detect the association.

8.3 The Correlation Coefficient (r)

The formula for the correlation coefficient is in Equation 7.21.

The correlation coefficient ranges from -1.0 to $+1.0$. The correlation coefficient (r) tells you two things: (1) the direction of the association (positive or negative) and (2) the magnitude of the association. If the correlation coefficient is positive, the association is positive. If the correlation coefficient is negative, the association is negative. If the association is positive, as x increases, y increases (or conversely, as x decreases, y decreases). If the association is negative, as x increases, y decreases (or conversely, as x decreases, y increases). The smaller the absolute value of the correlation coefficient (i.e., the closer the r value is to zero), the weaker the association and the flatter the slope of the best-fit line in a scatterplot. The larger the absolute value of the correlation coefficient (i.e., the closer the absolute value of the r value is to one), the stronger the association and the steeper the slope of the best-fit line in a scatterplot. See Figure 8.1 for a range of different correlation coefficients and what some example data may look like for each direction and strength of association.

```
set.seed(52242)
correlations <- data.frame(criterion = rnorm(1000))

correlations$v1 <- complement(correlations$criterion, -1)
correlations$v2 <- complement(correlations$criterion, -.9)
correlations$v3 <- complement(correlations$criterion, -.8)
correlations$v4 <- complement(correlations$criterion, -.7)
correlations$v5 <- complement(correlations$criterion, -.6)
correlations$v6 <- complement(correlations$criterion, -.5)
correlations$v7 <- complement(correlations$criterion, -.4)
correlations$v8 <- complement(correlations$criterion, -.3)
correlations$v9 <- complement(correlations$criterion, -.2)
correlations$v10 <- complement(correlations$criterion, -.1)
correlations$v11 <- complement(correlations$criterion, 0)
correlations$v12 <- complement(correlations$criterion, .1)
correlations$v13 <- complement(correlations$criterion, .2)
correlations$v14 <- complement(correlations$criterion, .3)
correlations$v15 <- complement(correlations$criterion, .4)
correlations$v16 <- complement(correlations$criterion, .5)
correlations$v17 <- complement(correlations$criterion, .6)
correlations$v18 <- complement(correlations$criterion, .7)
correlations$v19 <- complement(correlations$criterion, .8)
correlations$v20 <- complement(correlations$criterion, .9)
correlations$v21 <- complement(correlations$criterion, 1)

par(mfrow = c(7,3), mar = c(1, 0, 1, 0))

# -1.0
plot(correlations$criterion, correlations$v1, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",
```

```
main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v1)$estimate, 2), col = "black"))

# -.9
plot(correlations$criterion, correlations$v2, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v2)$estimate, 2), col = "black"))

# -.8
plot(correlations$criterion, correlations$v3, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v3)$estimate, 2), col = "black"))

# -.7
plot(correlations$criterion, correlations$v4, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v4)$estimate, 2), col = "black"))

# -.6
plot(correlations$criterion, correlations$v5, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v5)$estimate, 2), col = "black"))

# -.5
plot(correlations$criterion, correlations$v6, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v6)$estimate, 2), col = "black"))

# -.4
plot(correlations$criterion, correlations$v7, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v7)$estimate, 2), col = "black"))

# -.3
plot(correlations$criterion, correlations$v8, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v8)$estimate, 2), col = "black"))

# -.2
plot(correlations$criterion, correlations$v9, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = ""), list(x = round(cor.test(x = correlations$criterion, y = correlations$v9)$estimate, 2), col = "black"))

# -.1
```

```
plot(correlations$criterion, correlations$v10, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v10)$estimate, 2)), col = "black")  
  
abline(lm(v10 ~ criterion, data = correlations), col = "black")  
  
# 0.0  
plot(correlations$criterion, correlations$v11, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v11)$estimate, 2)), col = "black")  
  
abline(lm(v11 ~ criterion, data = correlations), col = "black")  
  
# 0.1  
plot(correlations$criterion, correlations$v12, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v12)$estimate, 2)), col = "black")  
  
abline(lm(v12 ~ criterion, data = correlations), col = "black")  
  
# 0.2  
plot(correlations$criterion, correlations$v13, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v13)$estimate, 2)), col = "black")  
  
abline(lm(v13 ~ criterion, data = correlations), col = "black")  
  
# 0.3  
plot(correlations$criterion, correlations$v14, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v14)$estimate, 2)), col = "black")  
  
abline(lm(v14 ~ criterion, data = correlations), col = "black")  
  
# 0.4  
plot(correlations$criterion, correlations$v15, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v15)$estimate, 2)), col = "black")  
  
abline(lm(v15 ~ criterion, data = correlations), col = "black")  
  
# 0.5  
plot(correlations$criterion, correlations$v16, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v16)$estimate, 2)), col = "black")  
  
abline(lm(v16 ~ criterion, data = correlations), col = "black")  
  
# 0.6  
plot(correlations$criterion, correlations$v17, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v17)$estimate, 2)), col = "black")  
  
abline(lm(v17 ~ criterion, data = correlations), col = "black")  
  
# 0.7  
plot(correlations$criterion, correlations$v18, xaxt = "n", yaxt = "n", xlab = "" , ylab = "",  
     main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v18)$estimate, 2)), col = "black")  
  
abline(lm(v18 ~ criterion, data = correlations), col = "black")
```

```
# 0.8
plot(correlations$criterion, correlations$v19, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v19)$estimate, 2)))
abline(lm(v19 ~ criterion, data = correlations), col = "black")

# 0.9
plot(correlations$criterion, correlations$v20, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v20)$estimate, 2)))
abline(lm(v20 ~ criterion, data = correlations), col = "black")

# 1.0
plot(correlations$criterion, correlations$v21, xaxt = "n", yaxt = "n", xlab = "", ylab = "",
      main = substitute(paste(italic(r), " = ", x, sep = "")), list(x = round(cor.test(x = correlations$criterion, y = correlations$v21)$estimate, 2)))
abline(lm(v21 ~ criterion, data = correlations), col = "black")

invisible(dev.off()) #par(mfrow = c(1,1))
```

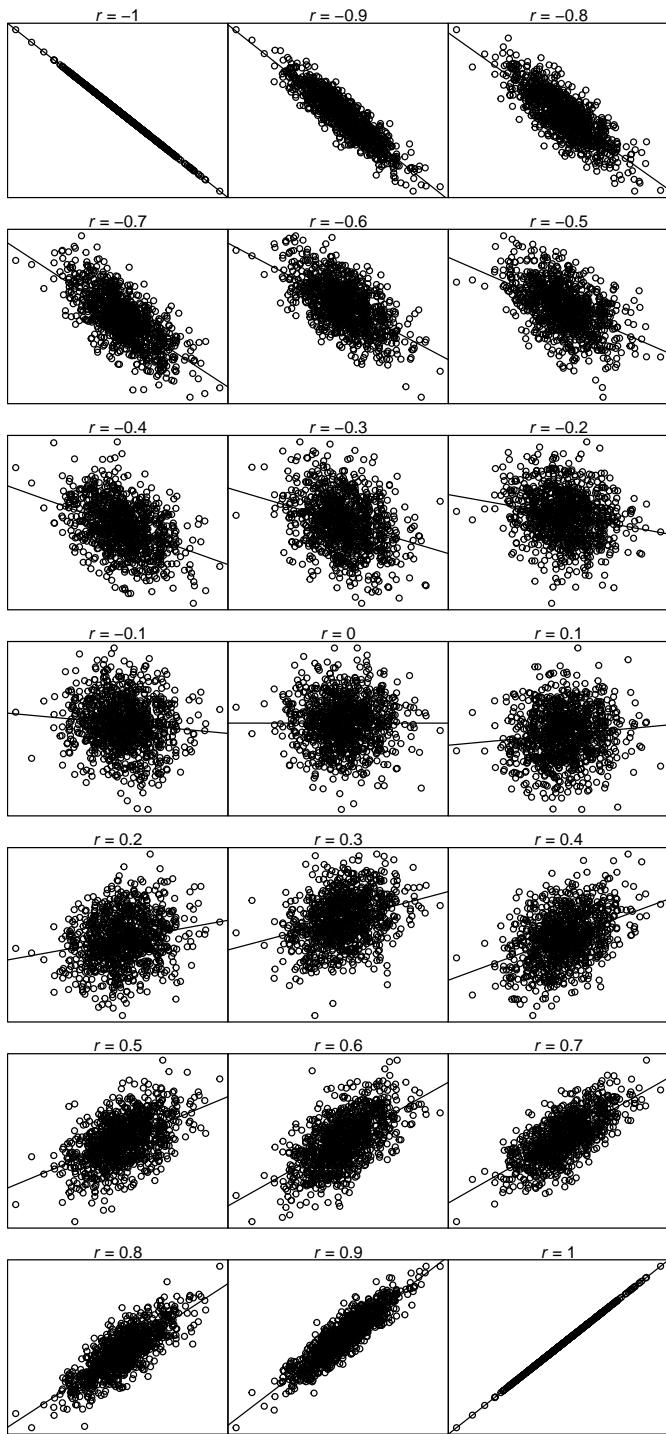


Figure 8.1 Correlation Coefficients.

See `?@fig-interpretingCorrelationCoefficients` for the interpretation of the magnitude and direction (sign) of various correlation coefficients.

```
library("patchwork")

set.seed(52242)
correlations2 <- data.frame(criterion = rnorm(15))

correlations2$v1 <- complement(correlations2$criterion, -1)
correlations2$v2 <- complement(correlations2$criterion, -.9)
correlations2$v3 <- complement(correlations2$criterion, -.8)
correlations2$v4 <- complement(correlations2$criterion, -.7)
correlations2$v5 <- complement(correlations2$criterion, -.6)
correlations2$v6 <- complement(correlations2$criterion, -.5)
correlations2$v7 <- complement(correlations2$criterion, -.4)
correlations2$v8 <- complement(correlations2$criterion, -.3)
correlations2$v9 <- complement(correlations2$criterion, -.2)
correlations2$v10 <- complement(correlations2$criterion, -.1)
correlations2$v11 <- complement(correlations2$criterion, 0)
correlations2$v12 <- complement(correlations2$criterion, .1)
correlations2$v13 <- complement(correlations2$criterion, .2)
correlations2$v14 <- complement(correlations2$criterion, .3)
correlations2$v15 <- complement(correlations2$criterion, .4)
correlations2$v16 <- complement(correlations2$criterion, .5)
correlations2$v17 <- complement(correlations2$criterion, .6)
correlations2$v18 <- complement(correlations2$criterion, .7)
correlations2$v19 <- complement(correlations2$criterion, .8)
correlations2$v20 <- complement(correlations2$criterion, .9)
correlations2$v21 <- complement(correlations2$criterion, 1)

# -1.0
p1 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v1
  )
) +
  geom_point() +
  geom_smooth(
    method = "lm",
    se = FALSE) +
  labs(
    title = "Perfect Negative Association",
    subtitle = expression(paste(italic("r"), " = ", "-1.0")))

```

```
) +
theme_classic(
  base_size = 16) +
theme(
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v1)): could not find function "gg

```
# -0.9
p2 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v2
  )
) +
  geom_point() +
  geom_smooth(
    method = "lm",
    se = FALSE) +
  labs(
    title = "Strong Negative Association",
    subtitle = expression(paste(italic("r"), " = ", "-0.9")))
) +
  theme_classic(
    base_size = 16) +
  theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v2)): could not find function "gg

```
# -0.5
p3 <- ggplot(
```

```
data = correlations2,
mapping = aes(
  x = criterion,
  y = v6
)
) +
geom_point() +
geom_smooth(
  method = "lm",
  se = FALSE) +
labs(
  title = "Moderate Negative Association",
  subtitle = expression(paste(italic("r"), " = ", "-0.5")))
) +
theme_classic(
  base_size = 16) +
theme(
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v6)): could not find function "gg

```
# -0.2
p4 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v9
)
) +
geom_point() +
geom_smooth(
  method = "lm",
  se = FALSE) +
labs(
  title = "Weak Negative Association",
  subtitle = expression(paste(italic("r"), " = ", "-0.2")))
) +
theme_classic(
  base_size = 16) +
```

```
theme(  
  axis.title.x = element_blank(),  
  axis.text.x = element_blank(),  
  axis.ticks.x = element_blank(),  
  axis.title.y = element_blank(),  
  axis.text.y = element_blank(),  
  axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v9)): could not find function "gg"

```
# 0.0  
p5 <- ggplot(  
  data = correlations2,  
  mapping = aes(  
    x = criterion,  
    y = v11  
  )  
) +  
  geom_point() +  
  geom_smooth(  
    method = "lm",  
    se = FALSE) +  
  labs(  
    title = "No Association",  
    subtitle = expression(paste(italic("r"), " = ", "0.0"))  
) +  
  theme_classic(  
    base_size = 16) +  
  theme(  
    axis.title.x = element_blank(),  
    axis.text.x = element_blank(),  
    axis.ticks.x = element_blank(),  
    axis.title.y = element_blank(),  
    axis.text.y = element_blank(),  
    axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v11)): could not find function "gg"

```
# 0.2  
p6 <- ggplot(  
  data = correlations2,  
  mapping = aes(  
    x = criterion,
```

```
    y = v13
)
) +
geom_point() +
geom_smooth(
  method = "lm",
  se = FALSE) +
labs(
  title = "Weak Positive Association",
  subtitle = expression(paste(italic("r"), " = ", "0.2")))
) +
theme_classic(
  base_size = 16) +
theme(
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  axis.ticks.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v13)): could not find function "g

```
# 0.5
p7 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v16
  )
) +
geom_point() +
geom_smooth(
  method = "lm",
  se = FALSE) +
labs(
  title = "Moderate Positive Association",
  subtitle = expression(paste(italic("r"), " = ", "0.5")))
) +
theme_classic(
  base_size = 16) +
theme(
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
```

```
axis.ticks.x = element_blank(),
axis.title.y = element_blank(),
axis.text.y = element_blank(),
axis.ticks.y = element_blank()
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v16)): could not find function "g

```
# 0.9
p8 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v20
  )
) +
  geom_point() +
  geom_smooth(
    method = "lm",
    se = FALSE) +
  labs(
    title = "Strong Positive Association",
    subtitle = expression(paste(italic("r"), " = ", "0.9")))
) +
  theme_classic(
    base_size = 16) +
  theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank())
```

Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v20)): could not find function "g

```
# 1.0
p9 <- ggplot(
  data = correlations2,
  mapping = aes(
    x = criterion,
    y = v21
  )
) +
```

```
geom_point() +
  geom_smooth(
    method = "lm",
    se = FALSE) +
  labs(
    title = "Perfect Positive Association",
    subtitle = expression(paste(italic("r"), " = ", "1.0")))
) +
  theme_classic(
    base_size = 16) +
  theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank())
```

```
Error in ggplot(data = correlations2, mapping = aes(x = criterion, y = v21)): could not find function "g
```

```
p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9
```

```
Error in eval(expr, envir, enclos): object 'p1' not found
```

Interactive visualizations by Kristoffer Magnusson on p -values and null-hypothesis significance testing are below:

- <https://rpsychologist.com/correlation/> (archived at <https://perma.cc/G8YR-VCM4>)

8.4 Examples

8.4.1 Covariance

8.4.2 Pearson Correlation

8.4.3 Spearman Correlation

8.4.4 Nonlinear Correlation

8.4.5 Correlation Matrix

8.4.6 Correlogram

8.5 Session Info

9

Multiple Regression

9.1 Getting Started

9.1.1 Load Packages

9.2 Overview of Multiple Regression

Multiple regression examines the association between multiple predictor variables and one outcome variable. It allows obtaining a more accurate estimate of the unique contribution of a given predictor, by controlling for other variables (**covariates**).

Regression with one predictor takes the form of Equation 9.1:

$$y = \beta_0 + \beta_1 x_1 + \epsilon \quad (9.1)$$

where y is the **outcome variable**, β_0 is the intercept, β_1 is the slope, x_1 is the **predictor variable**, and ϵ is the error term.

A regression line is depicted in Figure 9.3.

Regression with multiple predictors—i.e., multiple regression—takes the form of Equation 9.2:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon \quad (9.2)$$

where p is the number of **predictor variables**.

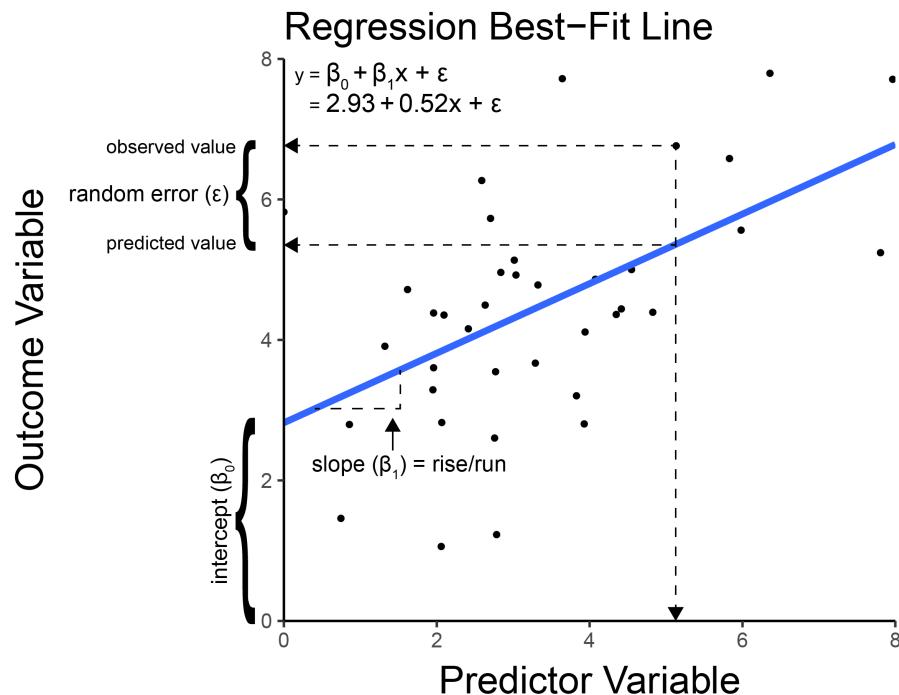


Figure 9.1 A Regression Best-Fit Line.

9.3 Components

- B = unstandardized coefficient: direction and magnitude of the estimate (original scale)
- β (beta) = standardized coefficient: direction and magnitude of the estimate (standard deviation scale)
- SE = standard error: uncertainty of unstandardized estimate

The unstandardized regression coefficient (B) is interpreted such that, for every unit change in the **predictor variable**, there is a ___ unit change in the outcome variable. For instance, when examining the association between age and fantasy points, if the unstandardized regression coefficient is 2.3, players score on average 2.3 more points for each additional year of age. (In reality, we might expect a nonlinear, inverted-U-shaped association between age and fantasy points such that players tend to reach their peak in the middle of their careers.) Unstandardized regression coefficients are tied to the metric of the raw data. Thus, a large unstandardized regression coefficient for two variables

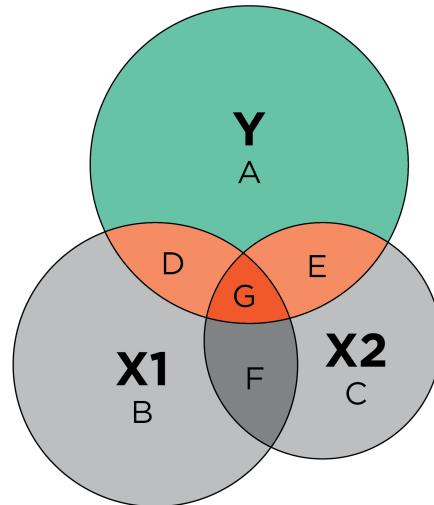
may mean completely different things. Holding the strength of the association constant, you tend to see larger unstandardized regression coefficients for variables with smaller units and smaller unstandardized regression coefficients for variables with larger units.

Standardized regression coefficients can be obtained by standardizing the variables to [z-scores](#) so they all have a mean of zero and standard deviation of one. The standardized regression coefficient (β) is interpreted such that, for every standard deviation change in the [predictor variable](#), there is a ___ standard deviation change in the outcome variable. For instance, when examining the association between age and fantasy points, if the standardized regression coefficient is 0.1, players score on average 0.1 standard deviation more points for each additional standard deviation of their year of age. Standardized regression coefficients—though not the case in all instances—tend to fall between $[-1, 1]$. Thus, standardized regression coefficients tend to be more comparable across variables and models compared to unstandardized regression coefficients. In this way, standardized regression coefficients provide a meaningful index of [effect size](#).

9.4 Coefficient of Determination (R^2)

9.5 Covariates

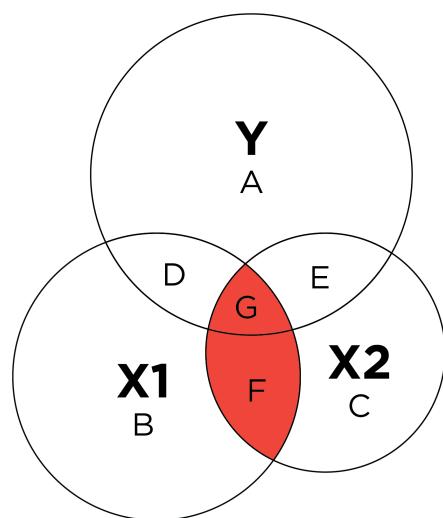
Covariates are variables that you include in the statistical model to try to control for them so you can better isolate the unique contribution of the predictor variable(s) in relation to the outcome variable. Use of covariates examines the association between the predictor variable and the outcome variable when holding people's level constant on the covariates. Inclusion of confounds as covariates allows potentially gaining a more accurate estimate of the causal effect of the predictor variable on the outcome variable. Ideally, you want to include any and all confounds as covariates. As described in Section 6.3.2.1, confounds are third variables that influence both the predictor variable and the outcome variable and explain their association. Covariates are potentially (but not necessarily) confounds. For instance, you might include the player's age as a covariate in a model that examines whether a player's 40-yard dash time at the NFL Combine predicts their fantasy points in their rookie year, but it may not be a confound.



$$R^2 = \frac{D + E + G}{A + D + E + G}$$

Figure 9.2 Conceptual Depiction of Proportion of Variance Explained (R^2) in an Outcome Variable (Y) by Multiple Predictors (X_1 and X_2) in Multiple Regression. The size of each circle represents the variable's variance. The proportion of variance in Y that is explained by the predictors is depicted by the areas in orange. The dark orange space (G) is where multiple predictors explain overlapping variance in the outcome. Overlapping variance that is explained in the outcome (G) will not be recovered in the regression coefficients when both predictors are included in the regression model. From Petersen (2024b) and Petersen (2024c).

9.6 Multicollinearity



$$\text{Multicollinearity} = F + G$$

Figure 9.3 Conceptual Depiction of Multicollinearity in Multiple Regression.
From Petersen (2024b) and Petersen (2024c).



References

- Chatterjee, S. (2021). A new coefficient of correlation. *Journal of the American Statistical Association*, 116(536), 2009–2022. <https://doi.org/10.1080/01621459.2020.1758115>
- Corston, R., & Colman, A. M. (2000). *A crash course in SPSS for windows*. Wiley-Blackwell.
- Goodman, S. (2008). A dirty dozen: Twelve p-value misconceptions. *Seminars in Hematology*, 45(3), 135–140. <https://doi.org/10.1053/j.seminhematol.2008.04.003>
- McGrath, R. E., & Meyer, G. J. (2006). When effect sizes disagree: The case of r and d. *Psychological Methods*, 11(4), 386–401. <https://doi.org/10.1037/1082-989X.11.4.386>
- 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>
- Petersen, I. T. (2024a). *petersenlab: A collection of R functions by the Petersen Lab*. <https://doi.org/10.5281/zenodo.7602890>
- Petersen, I. T. (2024c). *Principles of psychological assessment: With applied examples in R*. University of Iowa Libraries. <https://doi.org/10.25820/work.007199>
- Petersen, I. T. (2024b). *Principles of psychological assessment: With applied examples in R*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781003357421>



Index

correlation, [181](#), [182](#)

GitHub, [xi](#)