A Practitioner's Guide through the People Analytics Lifecycle

With Applications in R

Craig Starbuck

2022-03-11

Chapter 1

Foreword

Chapter 2

Introduction

Twenty years ago, I was the least likely person to write this book. My first statistics course in college was dreadful. On day one, my professor entered our large lecture hall of about 100 students and shared some grim stats: “Based on historical data, half of you won’t make it to the midterm and of those who do, half won’t receive a passing grade in the end.” This was both discouraging and motivating. Stats was a required course for my major so failure wasn’t an option; I had to pass. The course was challenging, and I attended weekly study sessions with classmates and studied a lot independently to learn the material. I saw no applications for statistics to anything I planned to do with my degree, so the course was reduced to memorization of equations; it was not enjoyable. I passed the course with a B, and I was determined to never open another stats book.

You may be wondering what changed to motivate authoring a book involving this insufferable subject. The short answer is that I discovered the very important applications to a discipline I truly love, people analytics. The practical applications were altogether absent from my undergraduate statistics course. As I began to think about complex and nuanced challenges in social science contexts, it became clear that I would not only need to reengage with stats; I would need to develop an authentic appreciation for the discipline. Over the past decade, I have taken the journey of ‘relearning’ statistics and developing a deep understanding of how statistical methodologies can be applied to various organizational problem statements.

My purpose in writing this book is to help make this content – which may unfortunately be intimidating to many – both accessible and exciting. In addition to my role in people analytics, I have taught a graduate-level business analytics course for Finance and MBA students for many years and have developed several teaching strategies through this experience that I will apply in this book. Beyond these instructional methods, this book makes a unique contribution in curating what I consider to be the most salient topics for people analytics practitioners. There are many texts available for deeper treatments of individual subjects covered in this book but as of this writing, I have found none that organize within a single text both theoretical and applied instruction spanning the whole of the people analytics lifecycle.

Thus, this book represents my earnest attempt to provide a concise – yet adequately comprehensive – treatment of the concepts and methods I've found to be most important for people analytics. My hope is that this book will ignite within you the same passion for analytics I have discovered over the past decade.

Craig Starbuck December 2021

Craig Starbuck, PhD is the CEO and Co-Founder of OrgAcuity, a tech company with a mission to democratize access to people analytics. Craig has built and led people analytics teams at companies such as Robinhood, Mastercard, Equifax, TD Ameritrade, and Scottrade, and he also spent a decade in various data engineering and analytics positions in the banking and health care industries. He is a Member of the Society for Industrial and Organizational Psychology (SIOP) and has a passion for transforming people data into information and insights that help organizations enhance the experience and wellbeing of employees.

Chapter 3

Project Planning

Nothing in this book will increase the value of an analysis no one needs. Analyses should always have a strong value proposition – a clear expectation of how an analysis will support a General Manager, People Partner, Salesperson, or other member of the organization. Curiosity is not a business reason, and this chapter will cover a set of guiding principles as well as a project management framework that will help ensure analytics are anchored in well-defined problem statements with meaningful ROI.

In addition, this book will cut through the fluff and teach you how to do stuff that matters. Knowledge of concepts is futile without an understanding of how to apply them to people analytics use cases. Whether you are a people leader, individual contributor, or aspiring analytics practitioner, this book is for you. This book will serve as a guide through the analytics lifecycle, curating the key concepts and applications germane to common questions and hypotheses within people analytics.

Guiding Principles

Among the many principles guiding how analytics teams operate, there are three that I have found to be universally applicable and critical to the success of an analytics capability.

Pro Employee Thinking

"With Great Power Comes Great Responsibility."

'Pro employee' thinking is addressed first and for good reason. People analytics has the power to improve the lives of people in meaningful ways. Whether we are shedding light on an area of the business struggling with work-life balance or identifying developmental areas of which a group of leaders may be unaware, people analytics ideally improves employee well-being and effectively, the success of the business. It is important to embrace a ‘pro employee’ philosophy, as newfound knowledge could also have damaging repercussions if shared with the wrong people or if findings are disseminated without proper instruction on how to interpret and act.

One way to error on the side of caution when considering whether to disseminate insights is to ask the following: “With this knowledge, could the recipient act in a manner that is inconsistent with our ‘pro employee’ philosophy?” If the answer to this question is not a clear “no”, discuss with your HR, legal, and privacy partners and together, determine how best to proceed. The decision may be to not share the findings with the intended audience at all or to develop a proper communication and training plan to ensure there is consistency in how recipients interpret the insights and act in response. Employment Law and Data Privacy Counsel are our friends, and it is important to build strong relationships with these critical partners.

Quality

“Garbage In, Garbage Out.”

Never compromise quality for greater velocity. If quality falls to the bottom of the priority list, all other efforts are pointless. It is unlikely that requestors of data and analytics will ever ask us to take longer to prepare the information. The onus is on us as analytics professionals to level set on a reasonable timeline for analyses based on many factors that can materially impact the quality of analyses and insights. A single instance of compromised quality can have lasting damage on the reputation of the analytics function and cause consumers of insights to view all findings as suspect. Be sure quality is consistently a top value and guard your team’s reputation at all costs. If stakeholders lose trust, there will likely be additional data requests for validation; this is wasteful to both you and your user community and detracts from the bigger story that needs to be conveyed.

To be clear, by ‘quality’ I am referring to results, which is dependent on data integrity in the source systems, proper data preparation steps, and many other factors. Most of the analyst’s time is spent on data preparation (data collection, cleaning and organizing, building training sets, mining for patterns, refining algorithms, etc.). If tight controls do not exist within the source application to support data integrity, data preparation efforts can only go so far in delivering reliable and valid findings. It is often the analysts who identify data integrity issues due to the nature of their work; therefore, close relationships should be formed with source application owners to put into place validation rules to proactively prevent the entry of erroneous data or at the very least, exception/audit reports to identify and address the issues soon after the fact.

Prioritization

“If everything is a priority, nothing is a priority.”

If there is not a supply-demand gap on the analytics team, the team likely isn’t asking enough questions. A backlog of projects can lend support for business cases requesting incremental funding to accelerate and expand the impact of people analytics. It is crucial to be relentless about prioritizing strategically important projects with ‘measurable’ impact over merely interesting questions that few care to answer. According to the Pareto Principle, 80% of outcomes (or outputs) result from 20% of causes (or inputs). In analytics, it is important to be laser focused on identifying the 20% of inputs that will result in disproportionate value creation for stakeholders. There are some general customer-oriented questions I have found to be helpful for the intake process to optimize the allocation of time and resources:

1. Does this support a company or departmental objective? If not, why should this be prioritized over something else?
2. Who is the executive sponsor? Really important projects will have an executive-level sponsor.
3. What quantitative and/or qualitative data can be provided as a rationale for this request? Is there data to support doing this, or is the problem statement rooted merely in theories and anecdotes?
4. Will this mitigate risk or enable opportunities?
5. What actions can or will be taken as a result of this analysis?
6. What is the *scale* of impact (# of impacted people)?
7. What is the *depth* of impact (minimum –> significant)?
8. Is this a dependency or blocker for another important deliverable?
9. What is the impact of not doing (or delaying) this?
10. What is the request date? Is there flexibility in this date and/or scope of the request (e.g., what does MVP look like)?

These questions can be weighted and scored as well to support a more automated and algorithmic approach to prioritization.

Tools

This book uses freely available software for statistics, modeling, and data visualization. All code has been written such that it is fully reproducible should you choose to follow along on your machine (and I highly recommend you do).

R

While there are many commercial-grade analytics toolsets, R is open-sourced statistical and data visualization software that can be downloaded free of charge. It is incredibly powerful, and there is a package (or at least the ability to easily create one) for every conceivable statistical technique and data visualization. It is also widely used in highly regulated environments. As of this writing, R Markdown – the dynamic document creator in which I am writing this book – allows for coding in 56 different languages! Therefore, the debate around whether to use Python, Julia, or something else is now moot; we need not sacrifice the advantages of other languages by choosing one.

Please note that while R basics are covered, this is not a book on how to code. It is assumed that you already understand programming fundamentals. If this is not the case, an introductory programming course is highly recommended; this is one of the best investments you can make for a successful career in analytics. The ability to write code is now table stakes for anyone in an analytics-oriented field, as this is the best way to develop reproducible analyses. Coding is to analytics professionals what typing was for Baby Boomers decades ago; a lack of coding proficiency is a major limiting factor on one's potential in this field.

The goal of the code provided in this book is not to represent the most performant, succinct, or productionalizable approaches. The code herein is intended only to facilitate understanding and demonstrate how concepts can be implemented in people analytics settings. Programming expertise is important for optimizing these approaches for production applications.

4D Framework

In practical analytics settings, we generally operate with respect to five primary constraints: timeliness, client expectation, accuracy, reliability, and cost (Bartlett, 2013). Adherence to a lightweight framework over hastily rushing into an analysis full of assumptions generally lends to better outcomes that respect these constraints. A framework ensures (a) the problem statement is understood and well-defined; (b) relevant literature and prior research are reviewed; (c) the measurement strategy is sound; (d) the analysis approach is suitable for the hypotheses being tested; and (e) results and conclusions are valid and communicated in a way that resonates with the target audience. This chapter will outline a recommended framework as well as other important considerations that should be reviewed early in the project.

It is important to develop a clear understanding of the key elements of research. Scientific research is the systematic, controlled, empirical, and critical investigation of natural phenomena guided by theory and hypotheses about the presumed relations among such phenomena (Kerlinger & Lee, 2000). In other words, research is an organized and systematic way of finding answers to questions. If you are in the business of analytics, I encourage you to think of yourself as a research scientist – regardless of whether you are wearing a lab coat or have plans to publish.

As we will discover when exploring the laws of probability in a later chapter, there is a 1 in 20 chance of finding a significant result when none exists. Therefore, it is important to remain disciplined and methodical to protect against backward research wherein the researcher mines data for interesting relationships or differences and then develops hypotheses which they know the data support. There have been many examples of bad research over the years, which often presents in the form of p-hacking or data dredging: the act of finding data to confirm what the researcher wants to prove. This can occur by running an exhaustive number of experiments to find one that supports the hypothesis, or by using only a subset of data which features the expected patterning.

Academics at elite research institutions are often under immense pressure to publish in top-tier journals which have a track record of accepting new groundbreaking research over replication studies or unsupported hypotheses, and incentives have unfortunately influenced some to compromise integrity. As my PhD advisor told me many years ago, an unsupported hypothesis – while initially disappointing given the exhaustive literature review that precedes its development – is a meaningful empirical contribution given theory suggests the opposite should be true.

If you participated in a science fair as a child, you are likely already familiar with the scientific method. The scientific method is the standard scheme of organized and systematic inquiry, and this duly applies to people analytics practitioners in the promotion of robust analyses and recommendations.

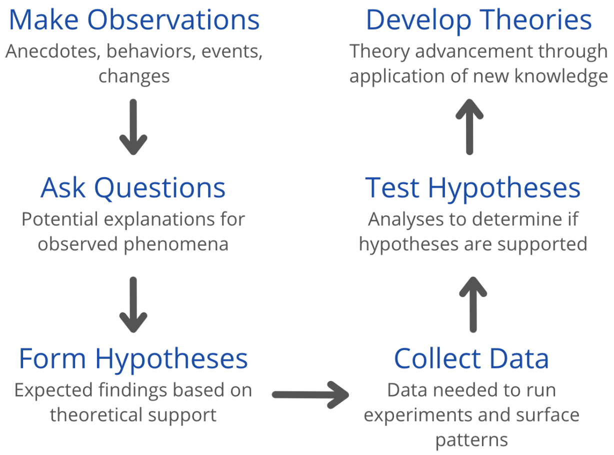

Make Observations
Anecdotes, behaviors, events, changes
Ask Questions
Potential explanations for observed phenomena
Form Hypotheses
Expected findings based on theoretical support
Collect Data
Data needed to run experiments and surface patterns
Test Hypotheses
Analyses to determine if hypotheses are supported
Develop Theories
Theory advancement through application of new knowledge

Figure 3.1: The Scientific Method

Over the years, I have adapted the scientific method into a curtailed four-dimensional framework which is intended to elevate the rigor applied to the end-to-end analytical process. The four dimensions are (a) Discover, (b) Design, (c) Develop, and (d) Deliver, and this book will be organized around these. A comprehensive checklist with questions and considerations for each phase of the analytics lifecycle can be found in the Appendix.

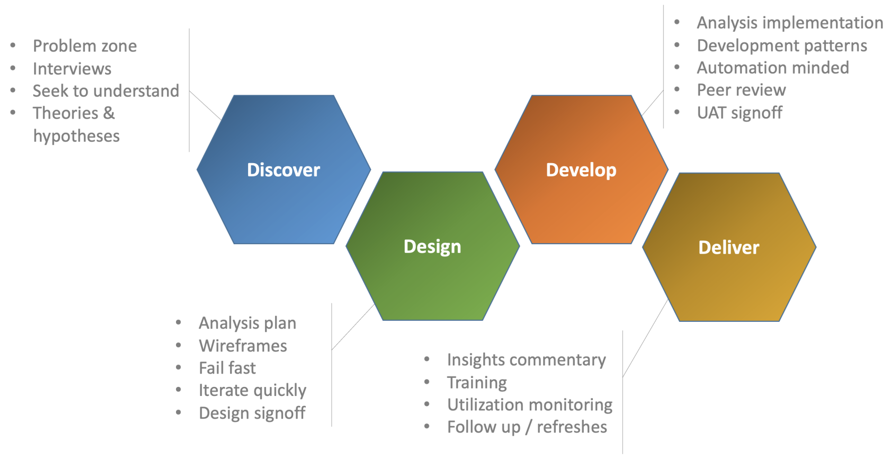


Figure 3.2: 4D Framework

Discover

You are likely familiar with the following adage: “An ounce of prevention is worth a pound of cure.” Such is the case with respect to planning in an analytics context. During the Discover phase, it is important to remain in the problem zone; seek to understand your clients’ needs through active listening and questions. This is not the time for solutioning or committing to any specific deliverables or timelines. If the client’s needs are ambiguous, proceeding without clarity is unlikely to result in a favorable outcome.

It is generally helpful to think about analytics solutions like a Product Owner thinks about the initial and subsequent releases of a commercial product. A **Minimum Viable Product (MVP)** is a version of the solution with the minimum number of features to be useful to early customers who can provide feedback for future enhancements. It is important to clarify that the MVP version of solutions has both a limited number of users and features and that this is antithetical to building solutions that seek to address every imaginable question for every possible user. Breaking down large projects into small sets of features that are easier to communicate and adopt provides space for agility and real-time adjustments to the product roadmap per user feedback.

Design

Perhaps the most important initial question to answer in the design phase is: "Does anything already exist that addresses part, or all, of the client's objectives?" If an existing solution will suffice, or a previous analysis can be easily refreshed with recent data, it may be possible to allocate time and resources elsewhere. If related or complimentary analyses have already been performed, they may accelerate new analyses.

The end-user experience is of paramount importance during the Design phase, as solutions should have a consistent look and feel regardless of who developed them. Defining and implementing design guidelines will ensure consistency across analytics projects, as well as within large projects in which multiple analysts are collaborating on various components of the solution.

Develop

While development patterns can vary widely across analytics teams, establishing a set of standards can pay dividends in the form of greater efficiency and reliability over time. Pattern-based development ensures analysts who were not involved in a particular project can access the code and easily and quickly understand each step of the analysis: data extraction –> wrangling –> cleaning –> analysis –> visualization.

This is one of the many reasons tools like Excel will not be covered in this book. Software like R and Python allows analysts to organize and annotate steps of the analytical process in a manner that is both logical and reproducible. In case it bears repeating, learning to code is likely the best investment one can make in the pursuit of a career in analytics.

Deliver

The Deliver phase can take many forms depending on the solution being released. If the solution is designed for a large user base, a series of recorded trainings may be in order so that there is a helpful reference for those unable to attend the live sessions or new joiners in the future. It is important to monitor success measures, which could be insights aligned to research hypotheses, dashboard utilization metrics, progress following data-informed interventions, or any number of others defined within the Discover phase.

Chapter 4

Introduction to R

This chapter covers how to install R, R Studio, and required packages for replicating examples in this book. This chapter also covers R basics such as objects, data structures, and data types that are fundamental to working in R. In addition, many common functions will be covered in this chapter, and many more will be introduced thoughout later chapters.

Getting Started

Run and test your code frequently. Writing a significant number of lines before testing will make debugging your code far more difficult and time-intensive than it needs to be.

Installing R

R can be compiled and ran on a variety of platforms including UNIX, Windows, and MacOS. R can be downloaded here: https://www.r-project.org/

When installing R, you will need to select a CRAN mirror. The **Comprehensive R Archive Network (CRAN)** is a network of servers around the world that store identical, current versions of code and documentation for R. You should select the CRAN mirror nearest you to minimize network load.

Installing R Studio

R Studio is an **Integrated Development Environment (IDE)** for R. R Studio provides a console with syntax editing that is helpful for debugging code as well as tools for plotting, history, and workspace management. Both open source and commercial editions are available, but the open source option is sufficient for replicating everything in this book.

R Studio can be downloaded here: https://www.rstudio.com/products/rstudio/download/#download

Installing Packages

Libraries from several R packages will be utilized in this book. The line of code below can be executed within R or R Studio to install all at once:

```r
install.packages(c("tidyverse", "corrplot", "psych", "moments", "ggpubr", "GGally", "sc
```

```
##
##   There is a binary version available but the source version is later:
##          binary source needs_compilation
## corrplot   0.89   0.92             FALSE
##
##
## The downloaded binary packages are in
##  /var/folders/b1/0nnhbsx55hvfb1b3n83x0qrw0000gn/T//Rtmpj6kjAH/downloaded_packages
```

Case Sensitivity

It's important to note that everything in R is case-sensitive. When working with functions, be sure to match the case when typing the function name. For example, `Mean()` is not the same as `mean()`; since `mean()` is the correct case for the function, capitalized characters will result in an error when executing the function.

Help

Documentation for functions is available via the `?` command. This can be inserted prior to any function to access the function's documentation. For example, `?mean` will display the documentation, including its required and optional arguments, for the `mean()` function.

Objects

Objects underpin just about everything we do in R. An object is a container for various types of data. Objects can take many forms, ranging from simple objects holding a single number or character to complex structures that support advanced visualizations. The assignment character `<-` is used to assign values to an object.

Let's use the assignment operator to assign a number and character to separate objects. Note that non-numeric values must be enveloped in either single ticks `''` or double quotes `""`:

```r
obj_1 <- 1
obj_1
```

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

```
obj_2 <- ’a’
obj_2
```

```
## [1] "a"
```

Several functions are available for returning the type of data in an object:

```
typeof(obj_2)
```

```
## [1] "character"
```

```
class(obj_2)
```

```
## [1] "character"
```

Comments

The `#` symbol is used for commenting/annotating code. This is a best practice to aid in quickly and easily deciphering the role of each line or block of code. Without comments, troubleshooting large scripts can be a more challenging task than necessary.

```
# Assign a new number to the object named obj_1
obj_1 <- 2
# Display value in obj_1
obj_1
```

```
## [1] 2
```

```
# Assign a new character to the object named obj_2
obj_2 <- ’b’
# Display value in obj_2
obj_2
```

```
## [1] "b"
```

In-line code comments can also be added where needed to reduce the number of lines in a script:

```
# Assign a new number to the object named obj_1
obj_1 <- 3
obj_1 # Display value in obj_1
```

```
## [1] 3
```

```
# Assign a new character to the object named obj_2
obj_2 <- 'c'
obj_2 # Display value in obj_2
```

```
## [1] "c"
```

Vectors

A **vector** is the most basic data object in R. Vectors contain a collection of data elements of the same data type. For example, a vector may contain a series of numbers, set of characters, collection of dates, or logical `TRUE` or `FALSE` results.

In this example, we introduce the combine function `c()`, which allows us to fill an object with more than one value:

```
# Create and fill a numeric vector named vect_num
vect_num <- c(2,4,6,8,10)
vect_num
```

```
## [1]  2  4  6  8 10
```

```
# Create and fill a character vector named vect_char
vect_char <- c('a','b','c')
vect_char
```

```
## [1] "a" "b" "c"
```

We can use the `as.Date()` function to convert character strings containing dates to an actual date data type. By default, anything within single ticks or double quotes is treated as a character, so we must make an **explicit type conversion** from characters to dates. Remember that R is case-sensitive. Therefore, `as.date()` is not a valid function; the `D` must be capitalized.

```
# Create and fill a date vector named vect_dt
vect_dt <- c(as.Date("2021-01-01"), as.Date("2022-01-01"))
vect_dt
```

```
## [1] "2021-01-01" "2022-01-01"
```

We can use the sequence generation function `seq()` to fill values between a start and end point using a specified interval. For example, we can fill `vect_dt` with the first day of each month between `2021-01-01` and `2022-01-01` via the `seq()` function and its `by = "months"` argument:

```
# Create and fill a date vector named vect_dt
vect_dt <- seq(as.Date("2021-01-01"), as.Date("2022-01-01"), by = 'months')
vect_dt
```

```
##  [1] "2021-01-01" "2021-02-01" "2021-03-01" "2021-04-01" "2021-05-01"
##  [6] "2021-06-01" "2021-07-01" "2021-08-01" "2021-09-01" "2021-10-01"
## [11] "2021-11-01" "2021-12-01" "2022-01-01"
```

We can also use the `:` operator to fill integers between a starting and ending number:

```
# Create and fill a numeric vector with values between 1 and 10
vect_num <- 1:10
vect_num
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

We can access a particular element of a vector using its index. An **index** represents the position in a set of elements. In R, the first element of a vector has an index of 1, and the final element of a vector has an index equal to the vector's length. The index is specified using square brackets, such as `[5]` for the fifth element of a vector.

```
# Return the value in position 5 of vect_num
vect_num[5]
```

```
## [1] 5
```

When applied to a vector, the `length()` function returns the number of elements in the vector, and this can be used to dynamically return the last value of vectors.

```
# Return the last element of vect_num
vect_num[length(vect_num)]
```

```
## [1] 10
```

Vectorized Operations

Vectorized operations (or vectorization) underpin mathematical operations in R and greatly simplify computation. For example, if we need to perform a mathematical operation to each data element in a numeric vector, we do not need to specify each and every element explicitly. We can simply apply the operation at the vector level, and the operation will be applied to each of the vector's individual elements.

```
# Create a numeric vector named x and fill with values between 1 and 10
x <- 1:10
```

```
# Add 2 to each element of x
x_plus2 <- x+2
x_plus2
```

```
##  [1]  3  4  5  6  7  8  9 10 11 12
```

```
# Multiply each element of x by 2
x_times2 <- x*2
x_times2
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
# Square each element of x
x_sq <- x^2
x_sq
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

Many built-in arithmetic functions are available and compatible with vectors:

```
# Aggregate sum of x elements
sum(x)
```

```
## [1] 55
```

```
# Count of x elements
length(x)
```

```
## [1] 10
```

```
# Square root of x elements
sqrt(x)
```

```
##  [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
##  [9] 3.000000 3.162278
```

```
# Natural logarithm of x elements
log(x)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
##  [8] 2.0794415 2.1972246 2.3025851
```

```
# Exponential of x elements
exp(x)
```

```
##  [1]     2.718282     7.389056    20.085537    54.598150   148.413159
##  [6]   403.428793  1096.633158  2980.957987  8103.083928 22026.465795
```

We can also perform various operations on multiple vectors. Vectorization will result in an implied ordering about the elements, in that the specified operation will be applied to the first elements of the vectors and then the second, then third, etc.

```
# Create vectors x1 and x2 and fill with integers
x1 <- 1:10
x2 <- 11:20

# Store sum of vectors to new x3 vector
x3 <- x1 + x2
x3
```

```
##  [1] 12 14 16 18 20 22 24 26 28 30
```

Using vectorization, we can also evaluate whether a specified condition is true or false for each element in a vector:

```
# Evaluate whether each element of x is less than 6, and store results to a logical vector
logical_rslts <- x<6
logical_rslts
```

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

Matrices

A **matrix** is similar to a vector in that it represents a collection of data elements of the same data type; however, the elements of a matrix are *arranged into a fixed number of rows and columns*.

We can create a matrix using the `matrix()` function. Per `?matrix`, the `nrow` and `ncol` arguments can be used to organize like data elements into a specified number of rows and columns.

```
# Create and fill matrix with numbers
mtrx_num <- matrix(data = 1:10, nrow = 5, ncol = 2)
mtrx_num
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

We can leverage shorthand for function calls. As long as the argument values are in the correct order by the documentation, the argument names are not required. Per `?matrix`, the first argument is `data`, followed by `nrow` and then `ncol`. Therefore, we can achieve the same result using the following:

```
# Create and fill matrix with numbers
mtrx_num <- matrix(1:10, 5, 2)
mtrx_num
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

Several functions are available to quickly retrieve the number of rows and columns in a rectangular object like a matrix:

```
# Return the number of rows in mtrx_num
nrow(mtrx_num)
```

```
## [1] 5
```

```
# Return the number of columns in mtrx_num
ncol(mtrx_num)
```

```
## [1] 2
```

```
# Return the number of columns and rows in mtrx_num
dim(mtrx_num)
```

```
## [1] 5 2
```

Factors

A **factor** is a data structure containing a finite number of categorical values. Each categorical value of a factor is known as a **level**, and the levels can be either ordered or unordered. This data structure is a requirement for several statistical models we will cover in later chapters.

We can create a factor using the `factor()` function:

```
# Create and fill factor with unordered categories
education <- factor(c("undergraduate", "post-graduate", "graduate"))
education
```

```
## [1] undergraduate post-graduate graduate
## Levels: graduate post-graduate undergraduate
```

Since education has an inherent ordering, we can use the `ordered` and `levels` arguments of the `factor()` function to order the categories:

```
# Create and fill factor with unordered categories
education <- factor(education, ordered = TRUE, levels = c("undergraduate", "graduate", "post-grad
education
```

```
## [1] undergraduate post-graduate graduate
## Levels: undergraduate < graduate < post-graduate
```

The ordering of factors is critical to a correct interpretation of statistical model output as we will cover later.

Data Frames

A *data frame* is similar to a matrix in that it organizes elements within rows and columns but unlike matrices, data frames can store multiple types of data. Data frames are often the most appropriate data structures for the data required in people analytics.

A data frame can be created using the `data.frame()` function:

```
# Create three vectors containing integers (x), characters (y), and dates (z)
x <- 1:10
y <- c(’a’,’b’,’c’,’d’,’e’,’f’,’g’,’h’,’i’,’j’)
z <- seq(as.Date("2021-01-01"), as.Date("2021-10-01"), by = ’months’)

# Create a data frame with 3 columns (vectors x, y, and z) and 10 rows
df <- data.frame(x, y, z)
df
```

```
##     x y          z
## 1   1 a 2021-01-01
## 2   2 b 2021-02-01
## 3   3 c 2021-03-01
## 4   4 d 2021-04-01
## 5   5 e 2021-05-01
## 6   6 f 2021-06-01
## 7   7 g 2021-07-01
## 8   8 h 2021-08-01
## 9   9 i 2021-09-01
## 10 10 j 2021-10-01
```

The data type of each vector in a data frame object can be viewed using the `str()` function:

```
# Describe the structure of df
str(df)
```

```
## ’data.frame’:    10 obs. of  3 variables:
##  $ x: int  1 2 3 4 5 6 7 8 9 10
##  $ y: chr  "a" "b" "c" "d" ...
##  $ z: Date, format: "2021-01-01" "2021-02-01" ...
```

Lists

Lists are versatile objects that can contain elements with different types and lengths. The elements of a list can be vectors, matrices, data frames, functions, or even other lists.

A list can be created using the `list()` function:

```
# Store vectors x, y, and z as well as df to a list
lst <- list(x, y, z, df)
str(lst)
```

```
## List of 4
##  $ : int [1:10] 1 2 3 4 5 6 7 8 9 10
##  $ : chr [1:10] "a" "b" "c" "d" ...
##  $ : Date[1:10], format: "2021-01-01" "2021-02-01" ...
##  $ :'data.frame':    10 obs. of  3 variables:
##   ..$ x: int [1:10] 1 2 3 4 5 6 7 8 9 10
##   ..$ y: chr [1:10] "a" "b" "c" "d" ...
##   ..$ z: Date[1:10], format: "2021-01-01" "2021-02-01" ...
```

Unlike vectors, accessing elements of a list requires double brackets, such as `[[3]]` for the third element.

```
# Return data from the third element of lst
lst[[3]]
```

```
##  [1] "2021-01-01" "2021-02-01" "2021-03-01" "2021-04-01" "2021-05-01"
##  [6] "2021-06-01" "2021-07-01" "2021-08-01" "2021-09-01" "2021-10-01"
```

Loops

In many cases, the need arises to perform an operation a variable number of times. Loops are available to avoid the cumbersome task of writing the same statement many times. The two most common types of loops are `while` and `for` loops.

Let's use a `while` loop to square integers between 1 and 5:

```
# Initialize variable
i <- 1

# Using a 'while' loop, square the values 1 through 5 and print results to the screen
# 'i' is a variable that takes on a value between 1 and 5 for the respective loop
while (i < 6) {

  print(i^2)
  i <- i + 1 # increment i by 1

}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
```

With a `while` loop, we needed to initialize the variable `i` as well as increment it by `1` within the loop. With a `for` loop, we can accomplish the same goal with less code:

```
# Using a ’for’ loop, square the values 1 through 5 and print results to the screen
for (i in 1:5) {

  print(i^2)

}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
```

User-Defined Functions (UDFs)

Though many built-in functions are available, R provides the flexibility to create our own. In fact, many functions used in this book are actually functions of functions.

Functions can be an effective alternative to loops. For example, here is a basic function that achieves the same goal as our `while` and `for` loop examples (i.e., squaring integers 1 through 5):

```
# Create a function named square.val() with one argument (x) that squares given x valu
square.val <- function(x) {

  x^2
}

# Pass integers 1 through 5 into the new square.val() function and display results
square.val(1:5)
```

```
## [1]  1  4  9 16 25
```

While many projects warrant UDFs and/or loops, we do not actually need either to square a set of integers. As you gain experience writing R code, you will naturally discover ways to write more performant and terse code:

```r
# Square integers
(1:5)^2
```

```
## [1]  1  4  9 16 25
```

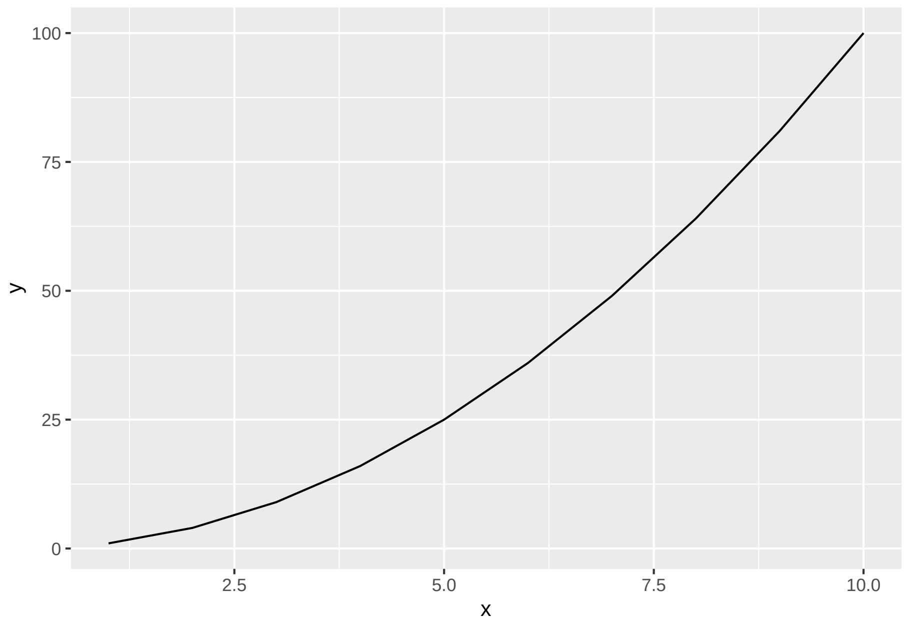
Graphics

While base R has native plotting capabilities, we will use the more flexibile and sophisticated visualization capabilities within the `ggplot2` library in this book. We can load the `ggplot2` library using the `library()` function:

```r
# Load library for data viz
library(ggplot2)
```

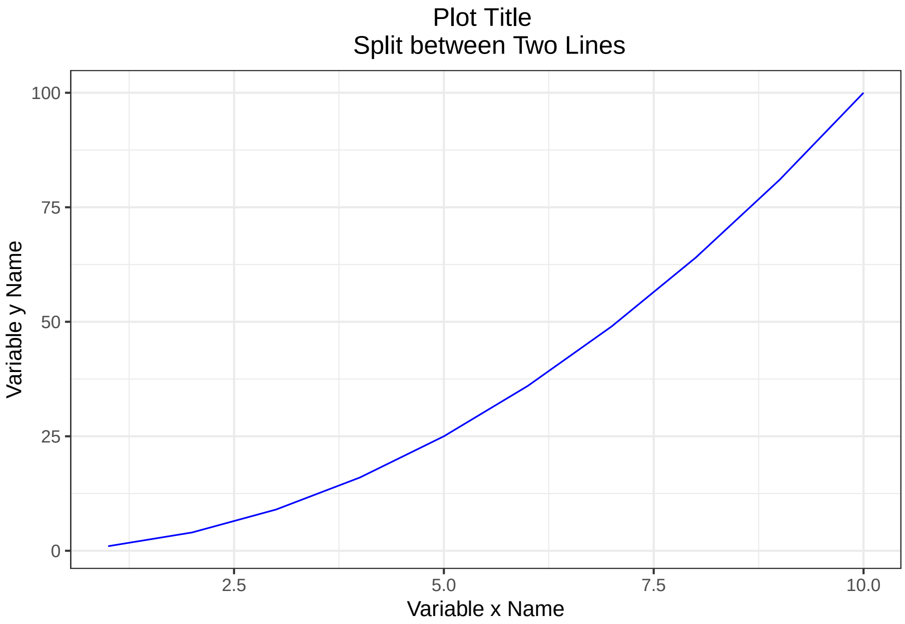
The `ggplot2` library contains many types of visualizations. For example, we can build a line chart to show how the values of vector `x` relate to values of vector `y` in a data frame named `data`:

```r
# Create data frame containing two related vectors
x <- 1:10
y <- (1:10)^2
data <- as.data.frame(cbind(x, y))

# Produce line chart
ggplot(data, aes(x = x, y = y)) +
  geom_line()
```



Furthermore, we can use `ggplot` parameters and themes to adjust the aesthetics of visuals:

```r
# Produce line chart
ggplot(data, aes(x = x, y = y)) +
  geom_line(size = .4, colour = "blue") + # Reduce line thickness and change color to
  theme_bw() + # Remove the default grey background
  ggtitle("Plot Title \n Split between Two Lines") + # Assign a title
  xlab("Variable x Name") + # Label the x-axis
  ylab("Variable y Name") + # Label the y-axis
  theme(plot.title = element_text(hjust = 0.5)) # Center plot title
```



Exercises

1. What is the difference between a factor and character vector?
2. What is vectorization?
3. How do data frames differ from matrices?
4. Executing the `Sum()` function will achieve the same result as `sum()`. A. True B. False
5. Executing `seq(1,10)` returns the same results as `1:10`. A. True B. False
6. The following command is sufficient for creating a vector recognized by R as having three dates: `dates <- c("2022-01-01", "2022-01-02", "2022-01-03")`. A. True B. False
7. How are `while` and `for` loops different?
8. If vectors `x1` and `x2` each hold 100 integers, how can we add each element of one to the respective element of the other using a single line of code?
9. List elements can be referenced by index using double brackets (e.g., `my_lst[[5]]` for the 5th element of my_lst). A. True B. False
10. What are some examples in which a user-defined function (UDF) is needed?

Chapter 5

Measurement & Sampling

This chapter will survey the type and function of study variables, measurement scales, sampling and nonsampling error, as well as probability and nonprobability sampling methods.

Variable Types

The framing of variables in research hypotheses guides the treatment of each in our analyses. In this section, we will discuss the function of *independent*, *dependent*, *control*, *moderating*, and *mediating* variables.

Independent Variables (IV)

An **Independent Variable (IV)** is a variable which is assumed to have a direct effect on another variable. IVs are sometimes referred to as *predictors*, *factors*, *features*, *antecedents*, or *explanatory* variables.

For example, we may wish to examine whether an inclusion training program given to a random sample of leaders has a positive effect on team-level belonging. In a true experimental design, participation in the inclusion training would be the only difference between the treatment (teams whose leaders who participate in the training) and the control (teams whose leaders who do not participate in the training). In this case, inclusion training participation is the IV (the variable we are manipulating).

IVs are also present in non-experimental designs. For example, we may survey employees and ask them to rate their leader's inclusiveness and provide self-reported belonging ratings. In this context, leader inclusiveness (rather than an inclusion training) is the IV. If we find that average team-level belonging scores tend to be higher when leader inclusiveness scores are higher, this *may* indicate that leader inclusion has some degree of influence on team-level belonging. Of course, there could be alternative explanations for any observed differences in team-level belonging, which is why experimental designs tend to provide stronger evidence for an IV's effect.

Dependent Variables (DV)

A **Dependent Variable (DV)** is a variable that is *dependent* on the IV. DVs are also referred to as *outcome*, *response*, or *criterion* variables.

In our leader inclusion example, team-level belonging is the DV since this variable is assumed to depend on the level of team leaders' inclusiveness. It's important to note that regardless of a study's results, it is the positioning of the variables in the study's hypotheses (rooted in theory) that determines the type of variable. If we hypothesize that leader inclusion training has a positive effect on team-level belonging, but the study finds no such effect, the inclusion intervention is still the IV and team-level belonging the DV.

Control Variables (CV)

A **Control Variable (CV)** is a variable that is held constant in research. The unchanging state of a CV allows us to understand the extent to which the IV has a unique and independent effect on the DV.

In an experimental context, control variables represent a researcher's attempt to control for alternative explanations so that the IV's main effect on the DV can be isolated. For example, in our leader inclusion example, to be confident that the reason for any observed differences in team-level belonging can be attributed to leader inclusion training rather than other factors that theoretically may explain differences. For example, we should ensure that the two groups are similar with respect to characteristics such as gender and ethnicity, since underrepresented groups (URGs) may have different experiences independent of any training programs. In this case, gender and ethnicity are CVs.

While we control for the effects of alternative explanations by way of the research design in an experimental context, we will discuss ways to control for these statistically in Chapter @ref(lm) for correlational designs. CVs are equally important in experimental and non-experimental contexts.

Moderating Variables

A **Moderating Variable** influences the strength of the effect of an IV on a DV. Moderating variables are often referred to as *interactions* or *interaction terms* in models.

Moderating variables may augment (strengthen) or attenuate (weaken) the effect one variable has on another. Interactions are widely understood in the context of medications; one drug may independently have a positive effect on one's health but when combined with another medication, the interaction can behave very differently – and may even be lethal! In our inclusive leadership example, we may find that the strength of the training's effect on team-level belonging varies based on a leader's span of control (SoC). It stands to reason that leaders with a lower SoC may find it easier to cultivate a climate of inclusivity and belonging, while leaders with a higher SoC may have more scope/projects and team dynamics to manage and may find it more difficult to consistently apply strategies covered during the training.

Interactions between variables are vital to our understanding of nuance and complexity in the dynamics influencing outcomes in organizational settings. Chapter @ref(lm) will cover how to test for significant interactions.

Mediating Variables

A **Mediating Variable** explains the *how* or *why* of an IV's effect on a DV. It may be helpful to think of mediating variables as a "go-between" – a part of the causal pathway of an effect.

In the case of inclusive leadership training, effects on belonging are likely not the result of the training itself. More likely, the training raised awareness and helped leaders develop strategies to help cultivate team inclusivity. One strategy endorsed during the training may be participative decision making, and the implementation of this strategy may explain **why** the training has a positive effect on team-level belonging. There may be multiple mediators of any observed effect of the intervention on team-level belonging depending on training outcomes.

Mediating variables may fully or partially mediate the relationship between an IV and DV. **Full mediation** indicates that the mediator fully explains the effect; in other words, without the mediator in the model, there is no relationship between an IV and DV. **Partial mediation** indicates that the mediator partially explains the effect; that is, there is still a relationship between an IV and DV without the mediator in the model. Partial mediation indicates that there are additional explanations for *how* or *why* observed effects exist, such as the implementation of additional team inclusion strategies influencing team belonging. In Chapter @ref(lm), we will discuss how to test for both full and partial mediation.

Translating research hypotheses into conceptual models of hypothesized relationships is helpful in visually representing the function of each variable in the study. Figure @ref(fig:concept-mdl) illustrates how each type of variable is depicted using our inclusive leadership example:

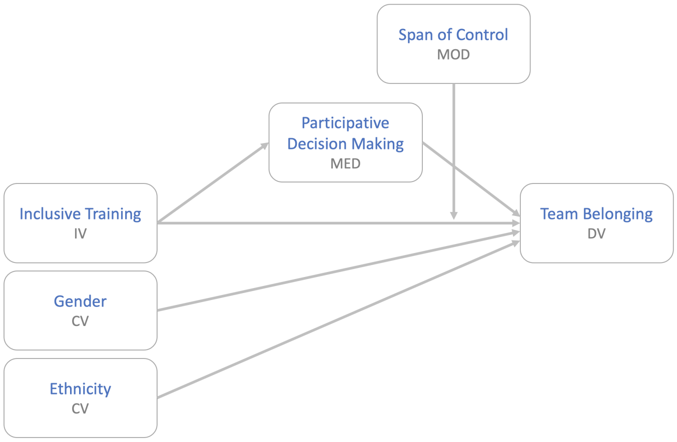


Figure 5.1: Conceptual Model of Hypothesized Relationships among Variables

Measurement Scales

Measurement scales are used to categorize and quantify variables. There are two major categorizations – *discrete* and *continuous* – and these, together with the research hypotheses, help determine appropriate types of analyses to perform.

Discrete Variables

Discrete variables are also known as *categorical* or *qualitative* variables. Categorical variables have a finite or countable number of values associated with them, and these can be further categorized as either *nominal* or *ordinal*.

Nominal

A **nominal** variable is one with two or more categories for which there is no intrinsic ordering to the categories. Examples of nominal variables include office locations, departments, and teams. A **dichotomous variable** is a type of nominal variable which has only two unordered categories. Examples of dichotomous variables include people leader vs. individual contributor, active vs. inactive status, and remote worker vs. non-remote worker.

Ordinal

An **ordinal** variable is similar to a nominal variable with one important difference: ordinal variables have *ordered* categories. Examples of ordinal variables include education levels, job levels, and survey variables measured on Likert-type scales.

Continuous Variables

Continuous variables are also known as *quantitative* variables. Continuous variables can assume any real value in some interval, and these can be further categorized as either *interval* or *ratio* variables.

Interval

Variables measured on an **interval** scale have a natural order and a quantifiable difference between values but no absolute zero value. Examples include SAT scores, IQ scores, and temperature measured in Fahrenheit or Celsius (but not Kelvin). In these examples, 0 is either not an option (i.e., SAT and IQ) or does not represent the absence of something (e.g., 0 degrees is a temperature, albeit a cold one!).

Ratio

Variables measured on a **ratio** scale have the same properties as data measured on an interval scale with one important difference: ratio data have an absolute zero value. Examples include compensation, revenue, and sales; a zero in these contexts is possible and would indicate a true absence of something.

Sampling

The goal of research is to understand a population based on data from a subset of population members. In practice, it is often not feasible to collect data from every member of a population, so we instead calculate **sample statistics** to estimate **population parameters**.

Another important concept is the **sampling frame**. While the population represents the entire group of interest, the sampling frame represents the subset of the population to which the researcher has access. In an ideal setting, the population and sampling frame are the same, but they are often different in practice. For example, a professor may be interested in understanding student sentiment about a new school policy but only has access to collect data from students in the courses she teaches. In this case, the entire student body is the population but the students she has access to (those in the courses she teaches) represent the sampling frame. The sample is the subset of the sampling frame that ultimately participates in the research (e.g,. those who complete a survey or participate in a focus group).

Sampling & Nonsampling Error

Sampling and nonsampling errors are general categorizations of biases and error in research (Albright & Winston, 2016).

Sampling error is the inevitable result of basing inferences on a random sample rather than the entire population. The two main contributors to sampling error are the size of the sample and variation in the underlying population. The risk of sampling error decreases as the sample size approaches the population size; however, it is usually not feasible to gain information from the entire population, so sampling error is generally a concern.

There are many types of **nonsampling error** that can invalidate results beyond the sampling procedure, and we will focus on several that are particularly germane to people analytics: *nonresponse bias*, *nontruthful responses*, *measurement error*, and *voluntary response bias*.

Nonresponse Bias

Surveys are a staple in the set of data sources germane to people analytics. While survey data provide unique attitudinal and perceptive signals that can be highly predictive of future behavior and events, surveys tend to be far more susceptible to nonsampling error than other data sources.

As discussed in the context of sampling error, we usually do not have access to information on entire populations of interest, so we must consider the possibility that those for whom we are missing data may have common qualities, perceptions, or opinions that differ from those for whom we do have data. This is known as **nonresponse bias**. For example, if we administer an employee experience survey to the entire organization and receive a 60% response rate, the reality is that we do not know how the 40% of nonrespondents would have responded. It is possible that nonrespondents represent highly disengaged employees, in which case their responses may have materially influenced results and conclusions in an unfavorable direction. It is also possible that the nonrespondents were busy, away on vacation, cynical to the confidentiality language in the communications, or any number of other reasons which may or may not have resulted in significantly different feedback from that of respondents.

Nonresponse bias is not limited to surveys. For example, self-reported demographics such as gender and ethnicity may not be disclosed by all employees in the HR information system (HRIS). This can bias segmentations based on these categorical dimensions. While there are strategies to address this, such as visual ID or applying models trained to infer missing values (which may be necessary to fulfill EEOC reporting requirements), there may still be error in the imputed values.

Nontruthful Responses

While high response rates may reduce nonresponse bias, this isn't always something to celebrate. Organizations that incentivize participation in surveys often do so at the risk of people responding in socially desirable ways and providing **nontruthful responses** to achieve some defined target. For example, if an employee has an unhealthy relationship with his manager but does not trust that managers will not have access to individual-level responses, the employee may decide to indicate on the survey that everything is highly favorable to help the team win the month of casual days leadership promised. This can of course skew and invalidate results.

While survey participation should be strongly encouraged since higher response rates can mitigate the risk of certain types of bias and increase confidence that the survey is representative of the collective organization's sentiments, incentivizing participation can be dangerous.

Measurement Error

Measurement error relates to errors stemming from confusing questions, survey fatigue, and low-quality scales used to measure multidimensional psychological constructs. The field of **psychometrics** is a vast scientific discipline concerned with the development of assessment tools, measurement instruments, and formalized models to understand latent psychological constructs such as engagement, belonging, purpose, and wellbeing using observable indicators.

The reduction of measurement error is a principal concern to psychometricians. For a deeper treatment of the survey scale development process, see DeVellis (2012). While an exhaustive treatment of psychometrics is beyond the scope of this book, *reliability* and *validity* are two broad sets of methods designed to increase the robustness of psychological instrumentation which will be reviewed in this section.

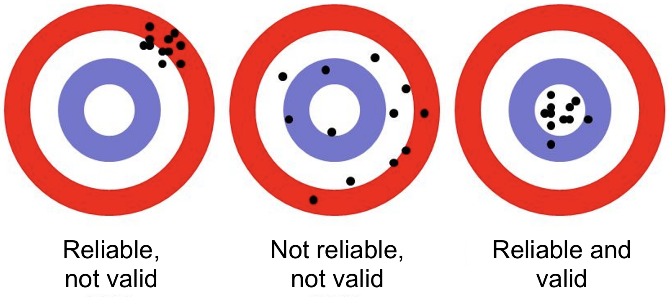


Figure 5.2: Reliability and Validity

Reliability describes the quality of measurement (i.e., the consistency or repeatability of measures). Types of reliability include:

- **Inter-Rater or Inter-Observer Reliability**: the degree to which different raters/observers give consistent estimates of the same phenomenon.
- **Test-Retest Reliability**: the consistency of a measure from one time to another.
- **Parallel-Forms Reliability**: the consistency of the results of two tests constructed in the same way from the same content domain.

- **Internal Consistency Reliability**: the consistency of results across items within a test.

Validity describes how well a concept was translated into a functioning and operating reality (operationalization). There are four main types of validity: (a) *face validity*, (b) *content validity*, (c) *construct validity*, and (d) *criterion-related validity*:

Face validity

Face validity is an assessment of how valid a measure appears on the surface. In other words, face validity represents whether the measurement approach “on its face” is a good translation of the construct. This is the least scientific method of validity and should never be accepted on its own merits.

Content Validity

Content validity is a somewhat subjective assessment of whether a measure covers the full content domain. Content validity relies on people’s perceptions to measure constructs that would otherwise be difficult to measure.

For example, a panel of experts may gather to discuss the various dimensions of a theoretical construct. The psychometrician may then use this information to develop survey items that tap these dimensions to achieve a comprehensive measure of the construct.

Construct Validity

In social science, constructs are often measured using a collection of related indicators that together, tap the various dimensions of the theoretical idea. Constructs may manifest in a set of behaviors, which provide evidence for their existence. **Construct validity** represents the degree to which a collection of indicators and behaviors – the **operationalization** of the concept – truly represents theoretical constructs.

Psychological safety, a belief that a context is safe for interpersonal risk-taking (Edmondson, 1999), has no direct measure. However, there are indicators and behaviors that are helpful in understanding the extent to which an environment is psychologically safe. We may ask employees whether they are able to bring up problems to decision makers or whether it is safe to take risks on their team. Based on the theoretical conception of psychological safety, these would be helpful (though not exhaustive) indicators of the construct in an organizational setting.

There are two types of construct validity:

- **Convergent validity**: the degree to which the operationalization is similar to (converges on) other operationalizations to which it theoretically should be similar.

- **Discriminant validity**: the degree to which the operationalization is not similar to (diverges from) other operationalizations to which it theoretically should not be similar.

A **nomological network** is central to providing evidence for a measure's construct validity. The nomological network is an idea developed by Cronbach and Meehl (1955) to represent the constructs of interest, their observable manifestations, as well as the interrelationships among them. The term "nomological" is derived from the Greek word meaning "lawful"; therefore, the nomological network aims to make clear what a construct means so that laws (nomologicals) can be applied. Simply put, the nomological network attempts to link the conceptual and theoretical realm to the observable one to provide a practical methodology for assessing a measure's construct validity.

If psychological safety theory suggests the construct should be positively related to leader openness and negatively associated with employee withholding (silence), we can use validated measures of openness and withholding to test for these theoretical relationships with psychological safety and substantiate construct validity.

Criterion-Related Validity

Criterion-related validity, sometimes referred to as *instrumental validity*, describes how well scores from one measure are adequate estimates of performance on an outcome measure (or *criterion*).

There are two types of criterion-related validity:

- **Predictive validity**: the operationalization's ability to predict something it should theoretically be able to predict.
- **Concurrent validity**: the operationalization's ability to distinguish between groups that it should theoretically be able to distinguish between.

If psychological safety should positively influence employee voice, there would be support for predictive validity if we find that employees who report more favorable perceptions of psychological safety are more willing to speak up. If we administer a new scale to measure psychological safety and *at the same time* (concurrently) administer an existing, validated measure of the same construct, highly correlated results would lend support for the new measure's concurrent validity.

Sampling Methods

It is important to understand the centrality of **randomness** in sampling. Randomization protects against subjective biases, self-validates the data, and is the key ingredient that defines the representative means of extracting information

(Kahneman, 2011). Sample data that are not representative of the population of interest can lend to anomalies – mere coincidences. While non-random data can be leveraged for directionally accurate insights, randomness is required to make inferences about a broader population with a reasonable degree of confidence.

Let's consider an example from Kahneman (2011) in which six babies are born in sequence at a hospital. The gender of these babies is of course random and independent; the gender of one does not influence the gender of another. Consider the three possible sequences of girls (G) and boys (B) below:

- BBBGGG
- GGGGGG
- BGBBGB

Are these sequences equally likely? Though it may initially be counter-intuitive, since the events are independent and the outcomes (B and G) are (approximately) equally likely, any possible sequence of births is as likely as any other.

Sample size (colloquially referred to as the n-count) is also important as this can have a material influence on the representativeness of sample data – and consequently, the veracity of results and conclusions based on them. As we will explore in Chapter @ref(inf-stats), as the sample size increases, so too does our confidence that estimates based on sample data reflect population parameters.

To illustrate the effects of sample sizes, let's consider a hypothetical study in which the promotion rate in an organization is found to be lowest in divisions that are primarily software engineers, low diversity, small, and geographically dispersed. Which of these characteristics might offer an explanation? Let's consider that this study also found that the divisions with highest promotion rates have identical characteristics: software engineers, low diversity, small, and geographically dispersed. Small samples yield extreme results more often than large samples. Small samples neither cause nor prevent outcomes; they merely allow the incidence of the outcome to be much higher (or much lower) than it is in the larger population (Kahneman, 2011).

Non-Probability Sampling

Non-probability sampling can help us gain insight into the *possible*. However, we cannot make inferences based on data collected through non-probability sampling methods since the sample is unlikely to be representative of the population.

Convenience (Accidental) Sampling

Convenience sampling is the most common type of nonprobabilistic sampling. This sampling method involves taking samples that are conveniently located around a specific location (physical or virtual).

For example, if we were to study employee sentiment about new benefit plans by polling employees walking through the lobby of a particular office building one morning, this would represent convenience sampling. Aside from the risk of employees sharing socially desirable responses in such a setting and invalidating the results, a major shortcoming of this approach is that we are only capturing the sentiment of those who happen to walk into one particular building during one limited window of time. This would not capture the voice of employees who are working remotely, working in another office location, on PTO, taking a sick day, attending an offsite conference or meeting, or stuck in traffic and running late.

Quota Sampling

Quota sampling is a nonprobabilistic sampling method in which researchers for a sample that is representative of a larger population. With quota sampling, researchers assign quotas to a group of people to create subgroups of individuals that reflect the characteristics of the population. This is nonprobabilistic since researchers choose the sample rather than randomly selecting it.

For example, if the characteristics of the employee population are known, the researcher polling employees in the office lobby about benefit plans could collect some additional information (e.g., department, job, tenure) to achieve a commensurate proportion of each in the sample. If 30% of employees in the larger workforce are in the Engineering department, the researcher could assign a quota – such as 3 in every 10 participants – to *choose* a sample in which 30% of employees come from the Engineering department.

Purposive (Judgmental) Sampling

The main goal of **purposive sampling** is to construct a sample by focusing on particular characteristics of a population that are of interest to the researcher. Purposive sampling is often used in qualitative or mixed methods research contexts in which a smaller sample is sufficient. Since it is a nonprobabilistic sampling method, purposive sampling is highly prone to researcher bias.

For example, the People Team may be interested in understanding what is top-of-mind for employees leading up to an engagement survey in order to design a survey with relevant items. The team may choose people to participate in focus groups to surface qualitative themes – not for the purpose of generalizing findings but to guide survey item selection efforts.

Probability Sampling

Probability sampling can help us gain insight into the *probable.* The main difference between non-probability and probability sampling is that non-probability sampling does not involve random selection and probability sampling does. Probability sampling is intended to facilitate inferences since data collected through random selection methods is more likely to be representative of the population and protects against over-indexing on participants with similar qualities in the sample.

Simple Random Sampling

Simple random sampling is a method in which each member of the population has the same probability of being selected for a sample. An example of simple random sampling is randomly selecting a specified number or percent of employees from the workforce to participate in a survey without regard for tenure, department, level, or other characteristics.

We can use the sample() function in R to randomly select from a vector of elements:

```r
# Load library for data wrangling
library(dplyr)

# Read employee demographics data
demographics <- read.csv("/Users/craig.starbuck/Library/Mobile Documents/com~apple~Clou

# Set seed for reproducible results
set.seed(1234)

# Sample 10 employees randomly
n = 10
sample(demographics$employee_id, n, replace = F)
```

```
##  [1] 2004 1623 3693 1934 5496 3948 3146 4175 3774 3374
```

The `replace = F` argument in the sample() function indicates that we want to sample *without* replacement (i.e., we do not want an employee to be selected twice in the sample). For example, if we draw multiple names from a hat without replacement, we will not put names back into the hat once they are drawn; each has a chance of being selected only once.

While sampling *with* replacement would not make sense for applications such as pulse survey participation, as we would not want a given employee to take a survey multiple times, `replace = T` can be applied if the application requires it (e.g., a lottery in which employees can win more than once).

Stratified Random Sampling

Stratified random sampling is a sampling method in which the population is first divided into *strata*. Then, a simple random sample is taken from each *stratum* – a homogeneous subset of the population with similar characteristics with regard to the variable of interest. The combined results constitute the sample.

To ensure samples do not comprise a larger proportion of employees from a particular department, education level, tenure band, generational cohort, or other variable deemed useful in explaining differences in response scores, researchers can randomly select members for each stratum based on the proportion in the respective stratum in the larger population. For example, if 30% of all employees are in the Engineering department, the researcher could *randomly* select a calculated number of employees from the Engineering department such that 30% of employees in the sample come from this department.

Let's demonstrate stratified random sampling in R by sampling $\frac{1}{2}$ of employees within each location. First, we can review counts of employees for each division using the group_by() and summarise() functions from the `dplyr` library:

```
# Return employee counts by location
demographics %>%
group_by(location) %>%
summarise(n = n())
```

```
## # A tibble: 4 x 2
##   location                           n
##   <chr>                          <int>
## 1 HQ 1 - Austin                   1733
## 2 HQ 2 - Atlanta                  1522
## 3 Operations Center - St. Louis    651
## 4 Remote                           969
```

Next, we will randomly select $\frac{1}{2}$ of employees within each location using the group_by() and sample_frac() functions from the `dplyr` library. We will store the selected employees' records to a data frame and then query it to validate that the counts are roughly $\frac{1}{2}$ the total count observed for each location.

```
# Obtain and store stratified random sample
strat_sample <- demographics %>%
                group_by(location) %>%
                sample_frac(size = .5)

# Return sample counts by location
strat_sample %>%
group_by(location) %>%
summarise(n = n())
```

```
## # A tibble: 4 x 2
##   location                           n
##   <chr>                          <int>
## 1 HQ 1 - Austin                    866
## 2 HQ 2 - Atlanta                   761
## 3 Operations Center - St. Louis    326
## 4 Remote                           484
```

Cluster Sampling

Cluster sampling is a sampling method often used in market research in which the population is first divided into clusters. Then, a simple random sample of clusters is taken. All the members of the selected clusters together constitute the sample. Unlike stratified random sampling, it is the clusters that are selected at random – not the individuals. It is hoped that each cluster by itself is representative of the population (i.e., each cluster is heterogeneous).

For example, employees could be partitioned into clusters based only on their geographic region. Since there is not further partitioning on other variables, each cluster is expected to be heterogeneous on the basis of variables other than geographic region – that is, unless geography is related to other variables (e.g., call center employees are all located in a company's Pacific Northwest region). By selecting a random set of clusters, the combination of employees across the selected clusters is expected to be representative of the population.

Let's demonstrate how to implement cluster sampling in R:

```
# Randomly assign each employee to 1 of 10 clusters (groups)
demographics$cluster <- sample(1:10, size = nrow(demographics), replace = T)

# Randomly select 5 clusters
clusters <- sample(unique(demographics$cluster), size = 5, replace = F)

# Store cluster sample
clust_sample <- demographics %>%
                filter(cluster %in% clusters)
```

Systematic Sampling

Systematic sampling involves selecting sample members from a population according to a random starting point and a fixed, periodic interval known as a sampling interval. The sampling interval is computed by taking the population size and dividing it by the desired sample size. The resulting number is the interval at which population members are selected for the sample.

For example, if there are 10,000 employees and our desired sample size is 500, the sampling interval is 20. Therefore, we would select every 20th employee for our sample. It is important that the sequence does not represent a standardized pattern that would bias the data; this process needs to be random. For example, if the employee id generated by the HRIS increases with time, we would expect employees with longer tenure to have lower employee ids while new joiners would have higher employee ids. Ordering employees by employee id prior to selection could bias the sample on the basis of variables related to tenure (e.g., aggressive periods of Engineering hiring).

Let's walk through the step-by-step process for implementing the systematic sampling procedure in R:

```
# Specify desired sample size
n = 100

# Determine population size
N = nrow(demographics)

# Compute sampling interval, rounding up to nearest whole number via the ceiling() function
si = ceiling(N/n)

# Randomly select value between 1 and the sampling interval for starting value
strt = sample(1:si, 1)

# Increment starting value by the sampling interval until the desired sample size is achieved, a
index = seq(strt, strt + si * (n - 1), si)

# Store systematic sample
syst_sample <- demographics[index, ]
```

Exercises

1. Parameters are descriptions or characteristics of a sample, while statistics are descriptions or characteristics of a population. A. True B. False

2. How does a sampling frame differ from a sample?

3. How does cluster sampling differ from stratified random sampling?

4. What is the primary benefit of probabilistic sampling methods over non-probabilistic sampling?

5. Nonresponse bias is only applicable in the context of surveys. A. True B. False

6. Which of the following variable types influences the strength of the effect one variable has on another? A. Dependent Variable B. Independent Variable C. Moderating Variable D. Mediating Variable

7. 100 randomly selected employees in the Marketing department of an organization participated in a survey on career pathing for marketing professionals. What is the sample and what is the population sampled in this case? A. Sample: 100 employees who completed the survey, Population: All employees in the organization B. Sample: 100 employees who completed the survey, Population: Marketing employees C. Sample: All Marketing professionals, Population: All employees in the organization D. Sample: All employees in the organization, Population: Employees across all companies globally

8. Zero represents the absence of something for interval scaled variables, whereas this is not the case for ratio scaled variables. A. True B. False

9. Discrete variables can have more than 2 values. A. True B. False

10. Which of the following are NOT probabilistic sampling methods? A. Simple random sampling B. Quota sampling C. Systematic sampling D. Purposive sampling

Chapter 6

Research Fundamentals

The importance of appropriate research methods and designs cannot be overstated. Just as probability sampling methods help us achieve data that are representative of the population, research methods and designs help us achieve an accurate understanding of various phenomena and ensure conclusions are justified.

Research Questions

Research questions are fundamental to all research projects. Research questions help focus the study, determine the appropriate methodology, and guide each stage of inquiry, analysis, and reporting. Some examples of research questions germane to people analytics include:

- Q_1: Why has there been an increase in attrition over the past quarter?
- Q_2: How equitable are promotion nominations across the organization?
- Q_3: Are there meaningful differences in the favorability of experiences for remote vs. non-remote employees?
- Q_4: Do new joiners have the training and resources they need to be successful?
- Q_5: What portion of team performance is attributable to leadership effectiveness?

Research Hypotheses

Research hypotheses are testable statements about the expected outcome of a research project or experiment.

- H_1: Manager satisfaction is a significant predictor of voluntary attrition.
- H_2: Promotion nomination rates are not significantly different by gender and ethnicity.
- H_3: Employee experience favorability is not significantly different between remote and non-remote workers.
- H_4: New hire training perceptions are positively associated with onboarding experience favorability.
- H_5: Leadership effectiveness perceptions explain significant variation in team performance.

Internal vs. External Validity

Internal validity refers to the extent to which confounding variables are controlled. In other words, internal validity reflects the *robustness* of the study.

For example, if a study finds a significant relationship between work location and attrition but considers no other factors or explanations, this would *not* be a robust study. Work location may emerge significant because certain roles for which attrition is higher are more concentrated in one or more geographies. It could also be the case that the company has made acquisitions in new geographies, and the acquired employees have significantly different experiences (and attrition rates) relative to non-acquired employees.

Confounding variables are critically important in the context of internal validity. A confounding variable is an extraneous variable whose presence impacts the variables being studied such that results do not reflect the actual relationships. Studies with weak internal validity often result in spurious associations that *confound* the true relationship between two variables, leading to invalid conclusions and recommendations.

External validity refers to the extent to which study conclusions will hold in other contexts (for other people, in other places, at other times). *Randomization* is fundamental to our ability to generalize and apply findings to other groups or contexts.

If we survey employees to understand sentiments about recent changes in business strategy but exclude groups for which there may be different impacts or perceptions, conclusions about the collective sentiment would be suspect at best.

Research Methods

There are three major categories of research methods: (1) *quantitative*, (2) *qualitative*, and (3) *mixed methods.*

1. **Quantitative**

- Addresses "what" questions
- Utilizes numerical data (e.g., surveys, systems)
- Primarily deductive
- Used to test hypotheses
- Involves statistical analyses
- More objective
- More generalizable

2. **Qualitative**

    - Addresses "how" and "why" questions
    - Utilizes text data (e.g., focus groups, interviews, open-ended feedback)
    - Primarily inductive
    - Used to formulate theory or hypotheses
    - Involves organizing data into categories or themes
    - More subjective
    - Less generalizable

3. **Mixed Methods**

    - Integrates the strengths of both quantitative and qualitative methods within a single study, often leading with qualitative approaches to build theory and hypotheses followed by quantitative methods to test hypotheses

Research Designs

In addition to determining whether a quantitative, qualitative, or mixed methods study is most appropriate, researchers also need to decide on the type of study within each of these three. **Research designs** are the types of inquiry within quantitative, qualitative, and mixed methods approaches that issue specific direction for the research procedures (Creswell, 2018). There are multiple taxonomies for research designs, and we will simplify to the most common types.

Within the quantitative category, there are three types of designs: (a) *experimental*, (b) *quasi-experimental*, and (c) *correlational*. As shown in Figure @ref(fig:res-designs), it is important to understand the centrality of randomization in this decision.

Experimental Research

Experimental research is concerned with casual (internal) validity. Randomized experimental designs provide the most rigor with regard to causal validity. However, in social science research contexts, true experiments often are not possible due to ethical considerations.

For example, if we were interested in understanding the causal effect leadership quality has on employee engagement, based on a hypothesis that poor leadership

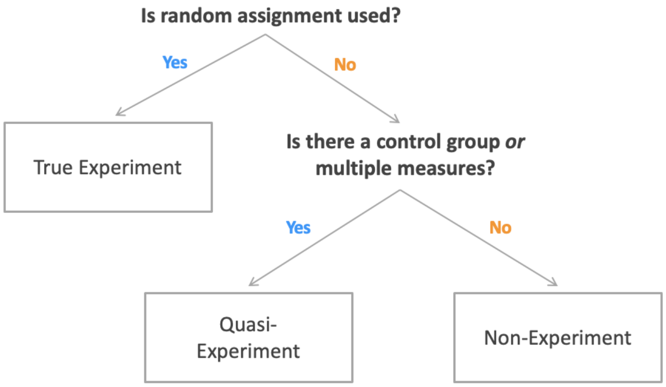

Is random assignment used?
Yes
No
True Experiment
Is there a control group *or* multiple measures?
Yes
No
Quasi-Experiment
Non-Experiment

Figure 6.1: Quantitative Research Designs

decreases employee engagement, we would need to randomly assign employees to one of two groups that are identical on the basis of all variables that could theoretically explain why employees vary in their levels of engagement. Then, we would need to manipulate the variable of interest (leadership quality) to evaluate if the group of employees subjected to poor leadership (treatment group) reports significantly different levels of engagement relative to the group of employees for whom leadership quality has not been manipulated (control group). In a practical setting, it would of course be unethical to purposefully subject employees to poor leadership with the expectation of reducing engagement – and consequently, retention, productivity, and impact to the organization.

Clinical trials are a common setting for true experiments, as isolating the effects of an experimental drug can be a matter of life or death. In a randomized clinical trial, patients are randomly assigned to an experimental group (patients who receive the drug) or control group (patients who receive a placebo). To protect against placebo effects biasing the results, patients do not know if they receive the experimental treatment or the placebo. Done correctly, these experiments have the highest level of internal validity.

Another example of an experimental design is **A/B testing**. A/B testing is often performed in the context of website optimization, in which two or more versions of the site are shown to customers to identify which version impacts key success metrics more positively. In a people analytics context, we may create two versions of a dashboard and *randomly* assign the permissioned users to each. We could then assess whether utilization rates, average usage time, repeat usage, among other success measures are significantly different between the two groups of users to inform which design is most effective.

In experimental research, it is important to consider the influence of the **Hawthorne Effect**, which refers to the tendency of some individuals to modify their behavior in response to the awareness that they are being observed. This term was coined during experiments at Western Electric's factory in the Hawthorne suburb of Chicago in the late 1920s and early 1930s. One of many studies conducted to understand how work environments effect productivity was known as the "Illumination Experiment". During this study, researchers experimented with a number of lighting levels in a warehouse in which workers made electrical relays. The researchers found that any change in the lighting – even when introducting poor lighting – led to favorable changes in output. However, these productivity gains disappeared once the attention faded (Roethlisberg & Dickson, 1939).

In a people analytics context, if we inform employees that we are going to monitor their productivity more closely over a period of time, it is likely that at least some employees will attempt to modify their behavior in order to increase productivity levels. After all, higher productivity is generally regarded as an ideal across companies and industries. In this case, manipulation of an IV to study a treatment effect, such as flexible work arrangements, would be impacted by this phenomenon; that is, observed differences in productivity may not be attributable to flexible work arrangements but merely due to employees knowing they were being observed.

Quasi-Experimental Research

Quasi-experimental research is an experiment in which participants cannot be randomly assigned.

In the case of our leadership quality example, a quasi-experiment may examine engagement differences between two groups of employees who rate their leader either favorably (Group A) or unfavorably (Group B). A key limitation of this approach is that the groups may be different in important ways beyond leader perception incongruities. For example, Group A employees may be concentrated within a single department, whereas Group B employees may span all other departments. This would indicate that the difference in leadership – and presumably engagement – is driven by factors unique to the department, making it more challenging to isolate the effects of leadership quality on engagement. Perhaps the department with unfavorable leader perceptions has seen significant attrition, or the department is largely first-time people leaders in need of coaching and support.

Another example of quasi-experiments is a pretest-posttest setting in which there are multiple measures. Random assignment could be used in pretest-posttest contexts, in which case this would be characterized as a true experiment, but often this approach is implemented without random assignment. For example, we could test the hypothesized effect of leadership quality on engagement via a pretest-posttest approach. If leaders are selected for a leadership development workshop, we could survey the leaders' teams and collect data on leader effectiveness perceptions and self-reported engagement prior to (baseline) and after the workshop. It is unlikely that leaders were selected for this workshop by a random process; more likely, there were criteria driving the selection, such as leaders who were identified as critical talent or who achieved a certain performance level. If this study finds that improvements in leadership effectiveness correlate with improvements in engagement, there would be *some* evidence in favor of improving leadership quality; however, this would not be sufficient evidence for a causal effect.

Though quasi-experiments are not as robust as true experiments, they are usually more feasible in a people analytics context. True experiments control for confounding variables by way of the research design (randomization ensures equivalent groups), while these factors must be controlled statistically in quasi-experimental contexts. In Chapter @ref(lm), we will discuss how to model relationships among multiple variables in order to study how one variable influences another while holding constant variables that may influence the outcome but are not the primary focus of the research.

Non-Experimental Research

Unlike experimental and quasi-experimental designs, **non-experimental research** does not involve the manipulation of an IV. The goal of non-experiments is not to provide evidence for causal effects, but to study measured variables as they naturally occur.

Given the potential for alternative explanations of any observed differences or relationships, non-experimental research tends to have lower internal validity than experimental and quasi-experimental designs. As we have discussed, it is often not possible or ethical to manipulate IVs or to randomly assign people to groups. In addition, the nature of research questions does not always warrant experiments. In these cases, one or three non-experimental approaches may be considered: (a) *cross-sectional*, (b) *correlational*, and (c) *observational*.

Cross-sectional research compares two or more natural groups of people. For example, we may examine differences in engagement between employees in the Engineering department relative to employees in the Product department. In this case, we would neither manipulate one's department to determine how department influences engagement, nor randomly assign employees to these departments. Department membership exists apart from the research, so these naturally occurring groups can be leveraged for comparisons. There are of course many examples of naturally occurring groups that we would not manipulate, such as gender, ethnicity, generation, education, job family, job level, location, and tenure band. When participant characteristics are used to create groups, the IV is sometimes referred to as *experimenter-selected* rather than *experimenter-manipulated* IVs.

Correlational research involves studying the statistical relationship between two variables without the manipulation of an IV. The relationship between leadership quality and engagement could be evaluated using correlational research. However, we would be unable to leverage a correlational design to test a hypothesis positing a causal effect of leadership quality on engagement. We would be limited to understanding how leadership quality and engagement covary; that is, to what extent a change in one variable is associated with a change in the other. Engagement may tend to increase as leadership quality increases, but a correlational design does not lend to understanding the direction of causal influence – if such an effect exists.

Observational research refers to studies in which the researcher gathers information without research subjects being explicitly involved in the recording of data. Collecting data from the company's HRIS could be an observational research method. For example, if we access data on terminations to determine the rate of attrition over a specified period, we would not need to interfere by asking past or present employees for this information. We would also do so without manipulating an IV, tagging people to naturally occurring or artificially created groups, or evaluating the association of attrition with another variable. The reality is that such an approach would not be too actionable, however, as this would offer no understanding of what may be influencing attrition or how attrition varies across departments, jobs, locations, or other theoretically-relevant dimensions.

Exercises

1. What type of research method and design would be best suited for a study aiming to understand the effect of stay interviews on employee attrition?
2. Why are quasi-experiments less rigorous than true experiments?
3. Why aren't experimental designs always implemented when an experimenter-manipulated IV is warranted?
4. What is the role of research questions?
5. What is the role of research hypotheses?
6. What is the difference between internal and external validity, and why are these concepts important in research?
7. What is an example of a mixed methods study?
8. What is the key difference between experimental and non-experimental research designs?
9. What are the differences between cross-sectional, correlational, and observational non-experimental designs?
10. How can the Hawthorne Effect impact the integrity of an experiment?

Chapter 7

Univariate & Bivariate Analysis

This chapter reviews essential univariate and bivariate analysis concepts that underpin the more complex statistical methods in subsequent chapters of this book.

Univariate Analysis

Descriptive statistics are rudimentary analysis techniques that help describe and summarize data in a meaningful way. Descriptive statistics do not allow us to draw any conclusions beyond the available data but are helpful in interpreting the data at hand.

There are two categories of descriptive statistics: (a) **measures of central tendency** describe the central position in a set of data; and (b) **measures of spread** describe how dispersed the data are.

Measures of Central Tendency

Mean

Perhaps the most intuitive measure of central tendency is the **mean**, which is often referred to as the average. The mean of a sample is denoted by $\bar{x}$ and is defined by:

$$\bar{x} = \frac{\displaystyle\sum_{i=1}^{n} x_i}{n}$$

The population mean is denoted by μ and is defined by:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{N}$$

The mean of a set of numeric values can be calculated using the mean() function in R:

```r
# Fill vector x with integers
x <- c(1,1,1,2,2,2,3,3,4,50)

# Calculate average of vector x
mean(x)
```

```
## [1] 6.9
```

Median

The **median** represents the midpoint in a sorted vector of numbers. For vectors with an even number of values, the median is the average of the middle two numbers; it is simply the middle number for vectors with an odd number of values. When the distribution of data is skewed, or there is an extreme value like we observe in vector x, the median is a better measure of central tendency.

The median() function in R can be used to handle the sorting and midpoint selection:

```r
# Calculate median of vector x
median(x)
```

```
## [1] 2
```

In this example, the median is only 2 while the mean is 6.9 (which is not representative of any of the values in vector x). Large deltas between mean and median values are evidence of outliers.

Mode

The **mode** is the most frequent number in a set of values.

While mean() and median() are standard functions in R, mode() returns the internal storage mode of the object rather than the statistical mode of the data. We can easily create a function to return the statistical mode(s):

```
# Create function to calculate statistical mode(s)
stat.mode <- function(x) {
  ux <- unique(x)
  tab <- tabulate(match(x, ux))
  ux[tab == max(tab)]
}

# Return mode(s) of vector x
stat.mode(x)
```

```
## [1] 1 2
```

In this case, we have a bimodal distribution since both 1 and 2 occur most frequently.

Range

The **range** is the difference between the maximum and minimum values in a set of numbers.

The range() function in R returns the minimum and maximum numbers:

```
# Return lowest and highest values of vector x
range(x)
```

```
## [1]  1 50
```

We can leverage the max() and min() functions to calculate the difference between these values:

```
# Calculate range of vector x
max(x, na.rm = TRUE) - min(x, na.rm = TRUE)
```

```
## [1] 49
```

In people analytics, there are many conventional descriptive metrics – largely counts, percentages, and ratios cut by time series (day, month, quarter, year) and categorical dimensions (department, job, location, tenure band). Here is a sample of common measures:

- Time to Fill: average days between job requisition posting and offer acceptance
- Offer Acceptance Rate: percent of offers extended to candidates that are accepted

- Pass-Through Rate: percent of candidates in a particular stage of the recruiting process who passed through to the next stage
- Progress to Goal: percent of approved positions that have been filled
- cNPS/eNPS: candidate and employee NPS (-100 to 100)
- Headcount: counts and percent of workforce across worker types (employee, intern, contingent)
- Diversity: counts and percent of workforce across gender, ethnicity, and generational cohorts
- Positions: count and percent of open, committed, and filled seats
- Hires: counts and rates
- Career Moves: counts and rates
- Turnover: counts and rates (usually terms / average headcount over the period)
- Workforce Growth: net changes over time, accounting for hires, internal transfers, and exits
- Span of Control: ratio of people leaders to individual contributors
- Layers/Tiers: average and median number of layers removed from CEO
- Engagement: average score or top-box favorability score

Measures of Spread

Variance

Variance is a measure of variability in the data. Variance is calculated using the average of squared differences – or deviations – from the mean.

Variance of a population is defined by:

$$\sigma^2 = \frac{\sum\limits_{i=1}^{n}(x_i - \mu)^2}{N}$$

Variance of a sample is defined by:

$$s^2 = \frac{\sum\limits_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

It is important to note that since differences are squared, the variance is always non-negative. In addition, we cannot compare these squared differences to the arithmetic mean since the units are different. For example, if we calculate the variance of annual compensation measured in USD, variance should be expressed as USD squared while the mean exists in the original USD unit of measurement.

In R, the sample variance can be calculated using the var() function:

```
# Load library for data wrangling
library(dplyr)

# Read employee demographics data
demographics <- read.csv("/Users/craig.starbuck/Library/Mobile Documents/com~apple~CloudDocs/Docu

# Calculate sample variance for annual compensation
var(demographics$annual_comp)
```

```
## [1] 1876686960
```

Sample statistics are the default in R. Since the population variance differs from the sample variance by a factor of $s^2(\frac{n-1}{n})$, it is simple to convert output from var() to the population variance:

```
# Store number of observations
n = length(demographics$annual_comp)

# Calculate population variance for annual compensation
var(demographics$annual_comp) * (n - 1) / n
```

```
## [1] 1876301998
```

Standard Deviation

The **standard deviation** is simply the square root of the variance.

The standard deviation of a population is defined by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}}$$

The standard deviation of a sample is defined by:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n - 1}}$$

Since a squared value can be converted back to its original units by taking its square root, the standard deviation expresses variability around the mean in the variable's original units.

In R, the sample standard deviation can be calculated using the sd() function:

```
# Calculate sample standard deviation for annual compensation
sd(demographics$annual_comp)
```

```
## [1] 43320.75
```

Since the population standard deviation differs from the sample standard deviation by a factor of $s\sqrt{\frac{n-1}{n}}$, it is simple to convert output from sd() to the population standard deviation:

```
# Calculate population standard deviation for annual compensation
sd(demographics$annual_comp) * sqrt((n - 1) / n)
```

```
## [1] 43316.3
```

Quartiles

Quartiles are a staple of exploratory data analysis (EDA). A quartile is a type of quantile that partitions data into four equally sized parts after ordering the data. Note that each partition is equally sized with respect to the number of data points – not the range of values in each. Quartiles are also related to **percentiles**. For example, Q1 is the 25th percentile – the value at or below which 25% of values lie. Percentiles are likely more familiar than quartiles, as percentiles show up in the height and weight measurements of babies, performance on standardized tests like the SAT and GRE, among other things.

The **Interquartile Range (IQR)** represents the difference between Q3 and Q1 cut point values (the middle two quartiles). The IQR is sometimes used to detect extreme values in a distribution; values less than $Q1 - 1.5 * IQR$ or greater than $Q3 + 1.5 * IQR$ are generally considered outliers.

In R, the quantile() function returns the values that bookend each quartile:

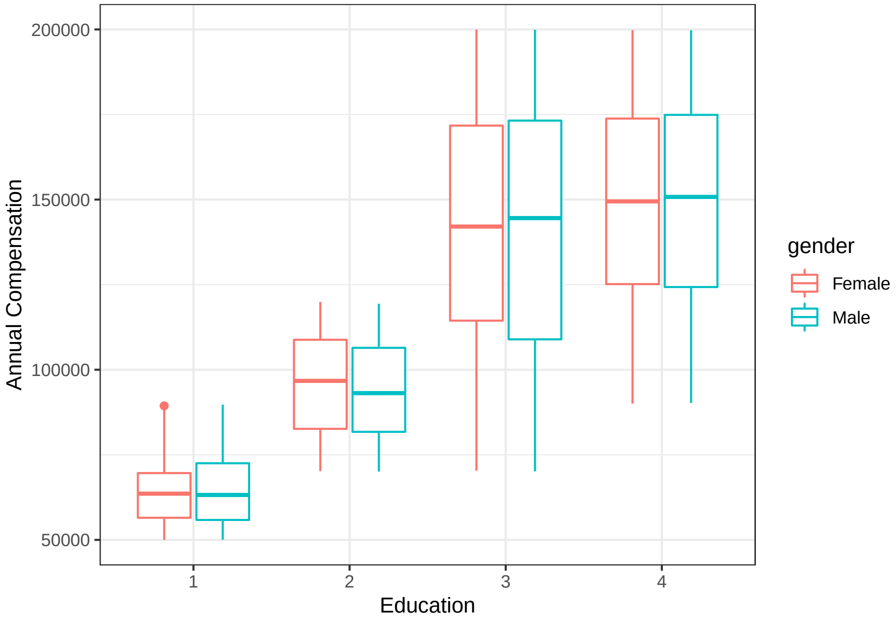
```
# Return quartiles for annual compensation
quantile(demographics$annual_comp)
```

```
##       0%      25%      50%      75%     100%
##  50016.0  87946.0 125522.0 163262.5 199968.0
```

Based on this output, we know that 25% of people in our data earn annual compensation of 87,946 USD or less, 125,522 USD is the median annual compensation, and 75% of people earn annual compensation of 163,263 USD or less.

Boxplots are a common way to visualize the distribution of data by categorical and ordinal factors. Boxplots are not usually found in presentations to stakeholders, since they are a bit more technical and often require explanation, but these are very useful to analysts for understanding data distributions during the EDA phase. In R, the ggplot2 library has robust and flexible data visualization capabilities which we will leverage throughout this book. Let's visualize the spread of annual compensation by education level and gender using the ggplot() function:

```
# Load library for data viz
library(ggplot2)

# Produce boxplots to visualize compensation distribution by education and gender
ggplot(demographics, aes(x = as.factor(education), y = annual_comp, color = gender)) +
  labs(x = "Education", y = "Annual Compensation") +
  theme_bw() +
  geom_boxplot()
```



Boxplots can be interpreted as follows: * Horizontal lines represent median compensation values. * The box in the middle of each distribution represents the IQR. * The end of the line above the IQR represents the threshold for outliers in the upper range: $Q3 + 1.5 * IQR$. * The end of the line below the IQR represents the threshold for outliers in the lower range: $Q1 - 1.5 * IQR$. * Data points represent outliers: $x > Q3 + 1.5 * IQR$ or $x < Q1 - 1.5 * IQR$.

We can also return a specific percentile value using the probs argument in the quantile() function. For example, if we want to know the 80th percentile annual compensation value, we can execute the following:

```r
# Return 80th percentile annual compensation value
quantile(demographics$annual_comp, probs = .8)
```

```
##      80%
## 170196.4
```

In addition, the summary() function returns several common descriptive statistics for an object:

```r
# Return common descriptives
summary(demographics$annual_comp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   50016   87946  125522  125369  163262  199968
```

Skewness

Skewness is a measure of the horizontal distance between the mode and mean – a representation of symmetric distortion. In most practical settings, data are not normally distributed. That is, the data are skewed either positively (right-tailed distribution) or negatively (left-tailed distribution). The coefficient of skewness is one of many ways in which we can asertain the degree of skew in the data. The skewness of sample data is defined as:

$$Sk = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{s^3}$$

A positive skewness coefficient indicates positive skew, while a negative coefficient indicates negative skew. The order of descriptive statistics can also be leveraged to ascertain the direction of skew in the data:

- Positive skewness: mode $<$ median $<$ mean
- Negative skewness: mode $>$ median $>$ mean
- Symmetrical distribution: mode $=$ median $=$ mean

Figure @ref(fig:skewness) illustrates the placement of these descriptive statistics in each of the three types of distributions:

The magnitude of skewness can be determined by measuring the distance between the mode and mean relative to the variable's scale. Alternatively, we can simply evaluate this using the coefficient of skewness:

- If skewness is between -0.5 - 0.5, the data are symmetrical.

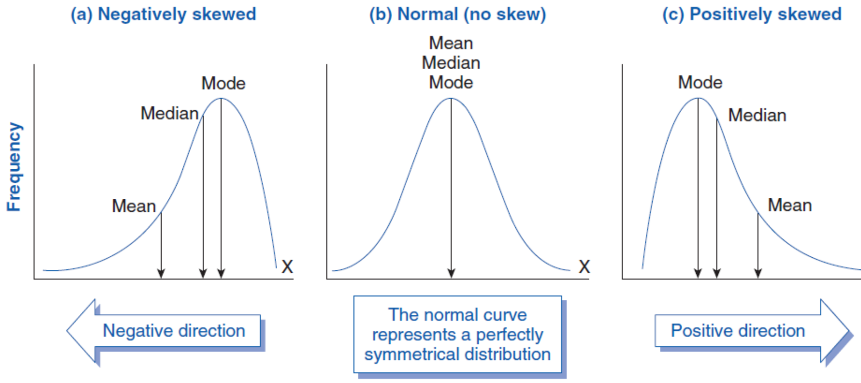


Figure 7.1: Skewness

- If skewness is between -0.5 and -1 or 0.5 and 1, the data are moderately skewed.
- If skewness is < -1 or > 1, the data are highly skewed.

Since there is not a base R function for skewness, we can leverage the moments library to calculate skewness:

```
# Load library
library(moments)

# Calculate skewness for org tenure, rounded to three significant figures
round(skewness(demographics$org_tenure), 3)
```

```
## [1] 0.562
```

Statistical Moments, after which this library was named, play an important role in specifying the appropriate probability distribution for a set of data. Moments are a set of statistical parameters used to describe the characteristics of a distribution. Skewness is the third statistical moment in the set; hence the sum of cubed differences and cubic polynomial in the denominator of the formula above. The complete set of moments comprises: (1) expected value or mean, (2) variance and standard deviation, (3) skewness, and (4) kurtosis.

We can verify that the skewness() function from the moments library returns the expected value (per the aforementioned formula) by validating against a manual calculation:

```
# Store components of skewness calculation
n = length(demographics$org_tenure)
```

```
x = demographics$org_tenure
x_bar = mean(demographics$org_tenure)
s = sd(demographics$org_tenure)

# Calculate skewness manually, rounded to three significant figures
round(1/n * (sum((x - x_bar)^3) / s^3), 3)
```

```
## [1] 0.562
```

A skewness coefficient of .562 indicates that organization tenure is moderately and positively skewed. We can visualize the data to confirm the expected right-tailed distribution:

```
# Produce histogram to visualize sample distribution
ggplot() +
  aes(demographics$org_tenure) +
  labs(x = "Organization Tenure", y = "Density") +
  geom_histogram(aes(y = ..density..), fill = "#414141") +
  geom_density(fill = "#ADD8E6", alpha = 0.6) +
  theme_bw()
```

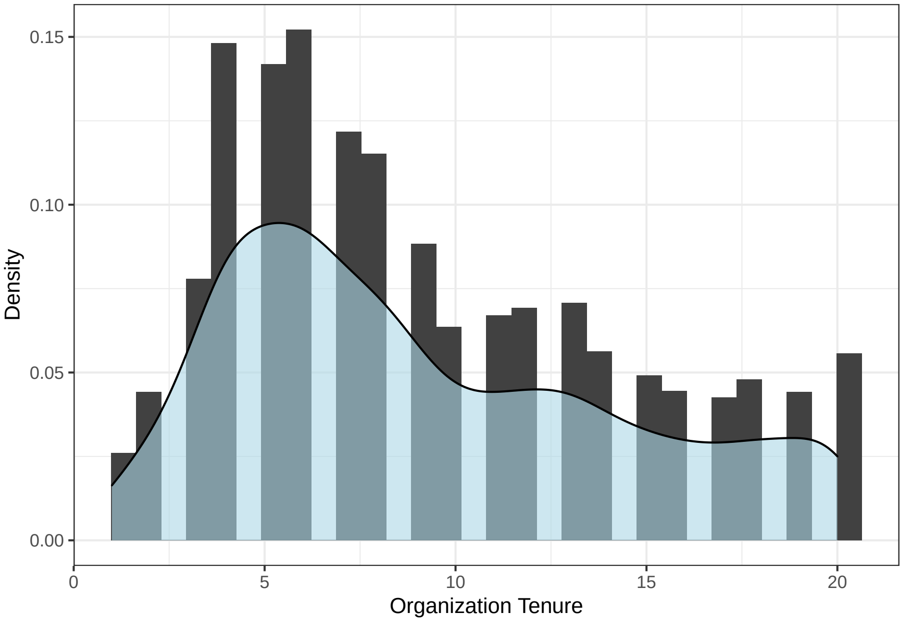


Figure 7.2: Organization Tenure Distribution

Kurtosis

While skewness provides information on the symmetry of a distribution, **kurtosis** provides information on the heaviness of a distribution's tails ("tailedness"). Kurtosis is the fourth statistical moment, defined by:

$$K = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4}$$

Note that the quartic functions characteristic of the fourth statistical moment are the only differences from the skewness formula we reviewed in the prior section (which featured cubic functions).

The terms **leptokurtic** and **platykurtic** are often used to describe distributions with light and heavy tails, respectively. "Platy-" in platykurtic is the same root as "platypus", and I've found it helpful to recall the characteristics of the flat platypus when characterizing frequency distributions as platkurtic (wide and flat) vs. its antithesis, leptokurtic (tall and skinny). The normal (or Gaussian) distribution is referred to as a **mesokurtic** distribution in the context of kurtosis.

Figure @ref(fig:kurtosis) illustrates the three kurtosis categorizations:

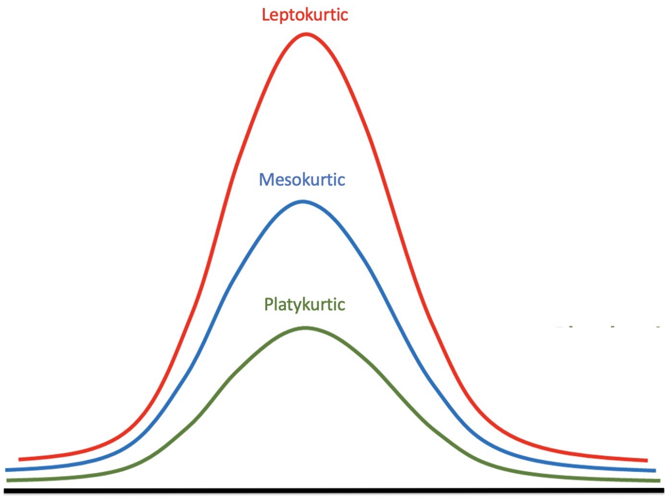


Figure 7.3: Kurtosis

Kurtosis is measured relative to a normal distribution. Normal distributions have a kurtosis coefficient of 3. Therefore, the kurtosis coefficient is greater than 3 for leptokurtic distributions and less than 3 for platykurtic distributions.

The moments library can also be used to calculate kurtosis in R:

```r
# Calculate kurtosis for org tenure, rounded to two significant figures
round(kurtosis(demographics$org_tenure), 2)
```

```
## [1] 2.26
```

We can verify that the kurtosis() function returns the expected value (per the aforementioned formula) by validating against a manual calculation:

```r
# Calculate kurtosis manually, rounded to two significant figures
round(1/n * (sum((x - x_bar)^4) / s^4), 2)
```

```
## [1] 2.26
```

As we saw in Figure @ref(fig:org-tenure-distribution), there is a long right tail about the organization tenure distribution. However, due to the moderate skew, there is no left tail. Therefore, our kurtosis coefficient is < 3 since the presence of the right tail is offset by the lack of a left tail. This is why it is important not to characterize a distribution based on a single isolated metric; we need the complete set of statistical moments to fully understand the distribution of data.

Bivariate Analysis

Covariance

While variance provides an understanding of how values for a single variable vary, **covariance** is an unstandardized measure of how two variables vary together. Values can range from $-\infty$ to $+\infty$, and these values can be used to understand the direction of the linear relationship between variables. Positive covariance values indicate that the variables vary in the same direction (e.g., tend to increase or decrease together), while negative covariance values indicate that the variables vary in opposite directions (e.g., when one increases, the other decreases, or vice versa).

Covariance of a sample is defined by:

$$cov_{x,y} = \frac{\sum\limits_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

It's important to note that while covariance aids our understanding of the direction of the relationship between two variables, we cannot use it to understand the strength of the association since it is unstandardized. Due to differences in variables' units of measurement, the strength of the relationship between two variables with large covariance could be weak, while the strength of the relationship between another pair of variables with small covariance could be strong.

In R, we can compute the covariance between a pair of numeric variables by passing the two vectors into the cov() function:

```
# Calculate sample covariance between annual compensation and age using complete observations (m
cov(demographics$annual_comp, demographics$age, use = "complete.obs")
```

```
## [1] 453179.4
```

In this example, using the default Pearson method, the covariance between annual compensation and age is 453179.1. The positive value indicates that annual compensation is generally higher for older employees and lower for younger employees.

Just as we multiplied the sample variance by $(n-1)/n$ to obtain the population variance, we can apply the same approach to convert the sample covariance returned by cov() to the population covariance:

```
# Calculate population covariance between annual compensation and age
cov(demographics$annual_comp, demographics$age, use = "complete.obs") * (n - 1) / n
```

```
## [1] 453086.4
```

The examples thus far have only examined associations between two variables at a time. However, rather than looking at isolated pairwise relationships, we can produce a covariance matrix to surface associations among many variables by passing a dataframe or matrix object into the cov() function:

```
# Generate a correlation matrix among continuous variables
cov(demographics[, c("annual_comp", "age", "org_tenure", "job_tenure")], use = "complete.obs")
```

```
##              annual_comp          age   org_tenure  job_tenure
## annual_comp 1.876687e+09 453179.39528 138551.73439 61576.65315
## age         4.531794e+05    168.38661     41.56006    17.96902
## org_tenure  1.385517e+05     41.56006     26.03270    11.04572
## job_tenure  6.157665e+04     17.96902     11.04572    13.29996
```

Using the default Pearson method, the cov() function will return sample variances for each variable down the diagonal, since covariance is not applicable in the context of a variable with itself. We can confirm by producing the variance for age:

```r
# Return sample variance for age
var(demographics$age)
```

```
## [1] 168.3866
```

As expected, the variance for age ($s^2 = 168.39$) matches the value found in the age x age cell of the covariance matrix.

Correlation

Correlation is a scaled form of covariance. While covariance provides an unstandardized measure of the direction of a relationship between variables, correlation provides a standardized measure that can be used to quantify both the direction and strength of bivariate relationships. Correlation coefficients range from -1 to 1, where -1 indicates a perfectly negative association, 1 indicates a perfectly positive association, and 0 indicates the absence of an association. **Pearson's product-moment correlation coefficient** r is defined by:

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

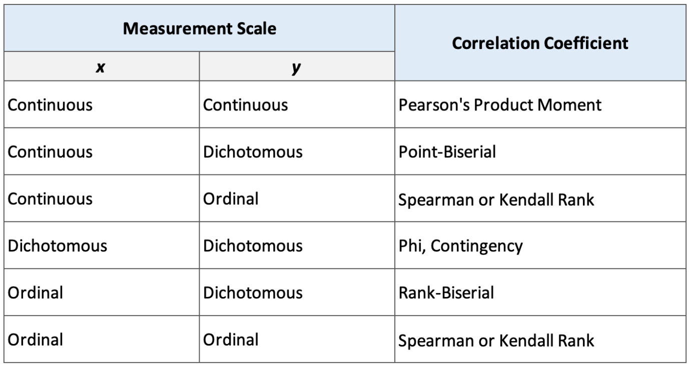
In R, Pearson's r can be calculated using the cor() function:

```r
# Calculate the correlation between annual compensation and age
cor(demographics$annual_comp, demographics$age, use = "complete.obs")
```

```
## [1] 0.8061585
```

While we already know that the relationship between annual compensation and age is positive based on the positive covariance coefficient, Pearson's r of .81 indicates that the strength of the positive association is strong ($r = 1$ is perfectly positive). Though there are no absolute rules for categorizing the strength of relationships, as thresholds often vary by domain, the following is a general rule of thumb for interpreting the strength of bivariate associations:

- Weak = Absolute value of correlation coefficients between 0 and .3
- Moderate = Absolute value of correlation coefficients between .4 and .6
- Strong = Absolute value of correlation coefficients between .7 and 1



Measurement Scale		Correlation Coefficient
x	*y*	
Continuous	Continuous	Pearson's Product Moment
Continuous	Dichotomous	Point-Biserial
Continuous	Ordinal	Spearman or Kendall Rank
Dichotomous	Dichotomous	Phi, Contingency
Ordinal	Dichotomous	Rank-Biserial
Ordinal	Ordinal	Spearman or Kendall Rank

Figure 7.4: Proper Applications of Correlation Coefficients

There are several correlation coefficients, and the measurement scale of x and y determine the appropriate type:

Pearson's r can be used when both variables are measured on continuous scales or when one is continuous and the other is dichotomous (point-biserial correlation).

When one or both variables are ordinal, we can leverage **Spearman's** ρ or **Kendall's** τ, which are both standardized nonparametric measures of the association between one or two rank-ordered variables. Let's look at Spearman's ρ, which is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

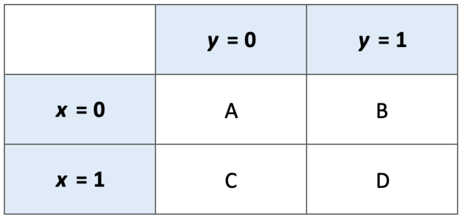
We can override the default Pearson method in the cor() function to implement a specific form of rank correlation using the `method` argument:

```r
# Calculate the correlation between job level and education level using Spearman's method
cor(demographics$job_level, demographics$education, method = "spearman", use = "complete.obs")
```

```
## [1] 0.6749776
```

The ρ coefficient of .67 indicates that the positive association we observed between job level and education level is moderate-to-strong. We could also pass `method = "kendall"` to this cor() function to implement Kendall's τ.

The **Phi Coefficient** (ϕ), sometimes referred to as the mean square contingency coefficient or Matthews correlation in ML, can be used to understand the association between two dichotomous variables. For a 2x2 table for two random variables x and y:



	y = 0	*y* = 1
x = 0	A	B
x = 1	C	D

Figure 7.5: 2x2 Table for Random Variables x and y

The ϕ coefficient is defined as:

$$\phi = \frac{(AD - BC)}{\sqrt{(A+B)(C+D)(A+C)(B+D)}}$$

To illustrate, let's examine whether there is a relationship between gender and performance after transforming performance from its ordinal form to a dichotomous variable (high vs. low performance). We can leverage the `psych` library to calculate ϕ in R:

```
# Load library for Phi Coefficient
library(psych)

# Set females to 1 and everything else to 0
demographics$gender_code <- ifelse(demographics$gender == 'Female', 1, 0)

# Set high performers (3 and above) to 1 and everything else to 0
demographics$performance_code <- ifelse(demographics$performance < 3, 0, 1)

# Create a 2x2 contingency table
contingency_tbl <- table(demographics$gender_code, demographics$performance_code)

# Calculate the Phi Coefficient between dichotomous variables
phi(contingency_tbl)
```

```
## [1] 0.01
```

ϕ is essentially 0, which means performance ratings are distributed equitably across gender categories (good news!).

A correlation matrix can be produced to surface associations among many variables by passing a dataframe or matrix object into the cor() function:

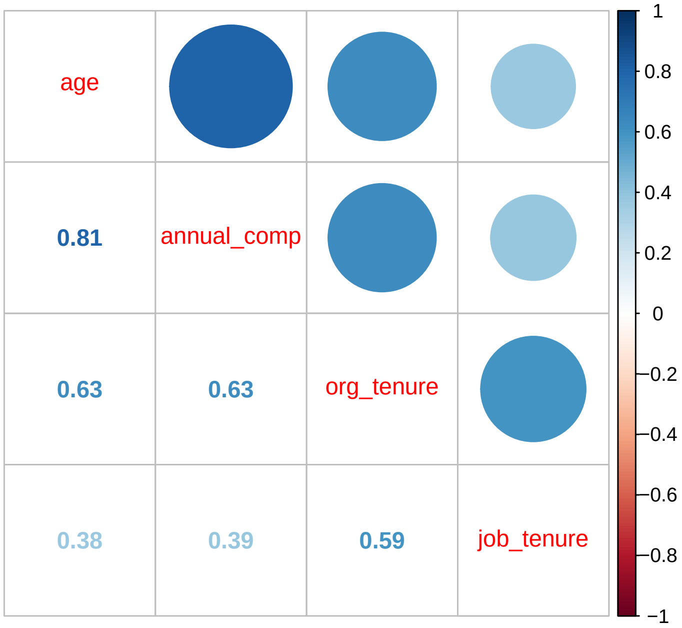
```
# Generate a correlation matrix among continuous variables
cor(demographics[, c("annual_comp", "age", "org_tenure", "job_tenure")], use = "complete.obs")
```

```
##             annual_comp       age org_tenure job_tenure
## annual_comp   1.0000000 0.8061585  0.6268397  0.3897580
## age           0.8061585 1.0000000  0.6277154  0.3797042
## org_tenure    0.6268397 0.6277154  1.0000000  0.5936211
## job_tenure    0.3897580 0.3797042  0.5936211  1.0000000
```

Based on this correlation matrix, there are several moderate and strong pairwise associations in the data. The values down the diagonal are 1 because these represent the correlation between each variable and itself. You may also notice that the information above and below the diagonal is identical and, therefore, redundant.

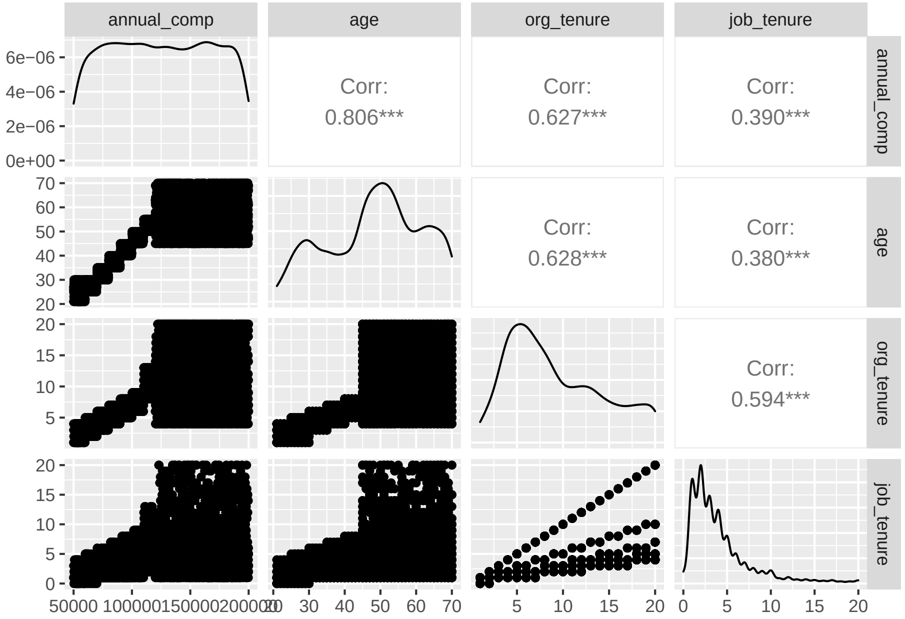
A great R library for visualizing correlation matrices is corrplot. Several arguments can be specified for various visual representations of the relationships among variables:

```
# Load library for correlation visuals
library(corrplot)

# Store correlation matrix to object M
M <- cor(demographics[, c("annual_comp", "age", "org_tenure", "job_tenure")], use = "complete.obs

# Visualize correlation matrix
corrplot.mixed(M, order = 'AOE')
```



The GGally library produces a variety of useful information, including correlation coefficients, bivariate scatterplots, and univariate distributions:

```
# Load library for correlation visuals
library(GGally)

# Visualize correlation matrix
ggpairs(demographics[, c("annual_comp", "age", "org_tenure", "job_tenure")])
```



As shown in these visuals, there are strong linear associations between variable pairs, such as age and org tenure – but only for certain values. For example, age is positively correlated with org tenure but beyond about 45 years-old, org tenure varies widely. We may find that these bivariate associations look quite different for certain business areas or jobs, assuming business units, divisions, and jobs were created at different points in the company's history. There is often a lot of noise in data at the broader company level, so understanding the nature and nuance of associations will be highly important. We will explore this further in the context of linear regression in Chapter @ref(lm).

It's important to remember that correlation is not causation. Correlations can be spurious (variables related by chance), and drawing conclusions based on bivariate associations alone – especially in the absence of sound theoretical underpinnings – can be dangerous.

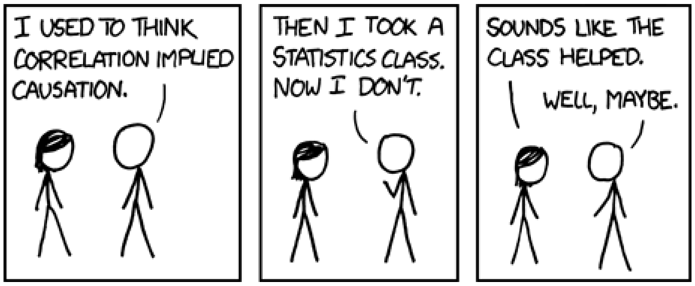
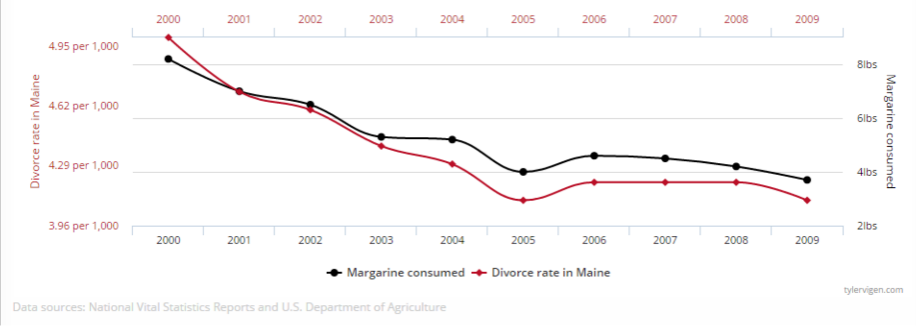

I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

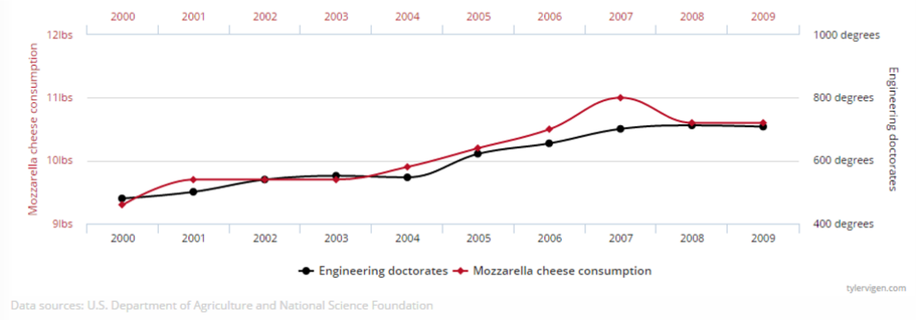
Figure 7.6: Correlation comic (Source: www.explainxkcd.com)

Here are two examples of nearly perfect correlations between variables for which there is likely no true direct association:



Data sources: National Vital Statistics Reports and U.S. Department of Agriculture

Figure 7.7: Correlation between Maine Divorce Rate and Margarine Consumption (r = .99)



Data sources: U.S. Department of Agriculture and National Science Foundation

Figure 7.8: Correlation between Mozzarella Cheese Consumption and Civil Engineering Doctorate Conferrals (r = .96)

In addition, covariance and correlation alone are not sufficient for determining whether an observed association in sample data is also present in the population. To understand the likelihood that patterns observed in sample data are also present in the larger population of interest, we need to move beyond descriptive measures.

Exercises

1. Which of the following measures of central tendency is least sensitive to extreme values (outliers)? A. Median B. Mean C. Range

2. The standard deviation represents the 'average' amount by which x values deviate (or vary) from the mean. A large standard deviation indicates

there is considerable spread in the data, whereas a small standard deviation indicates the mean is fairly representative of the data. A. True B. False

3. A positively skewed distribution has its largest allocation to the left and a negative distribution to the right. A. True B. False

4. Large covariance coefficients always indicate strong bivariate associations. A. True B. False

5. Which of the following can be found in boxplots? A. Quartiles B. Median C. Mean D. IQR E. Outliers

6. The 3rd quartile (Q3) is equivalent to the 75th percentile. A. True B. False

7. Which of the following correlation coefficients can be used when evaluating the relationship between a pair of rank-ordered variables? A. Pearson's Product Moment B. Spearman's Rank C. Phi D. Point-Biserial E. Kendall's Rank

8. Which of the following correlation coefficients can be used when evaluating the relationship between a pair of dichotomous variables? A. Pearson's Product Moment B. Spearman's Rank C. Phi D. Point-Biserial E. Kendall's Rank

9. Platykurtic distributions are flat relative to mesokurtic distributions. A. True B. False

10. When using the Pearson method, values down the diagonal of a covariance matrix represent each variable's variance. A. True B. False

Chapter 8

Inferential Statistics

The objective of **inferential statistics** is to make inferences – with some degree of confidence – about a population based on available sample data. Several related concepts are fundamental to this goal and will be covered here.

Introduction to Probability

Randomness and uncertainty exist all around us. In **probability theory**, random phenomena refer to events or experiments whose outcomes cannot be predicted with certainty (Pishro-Nik, 2014). If you've taken a course in probability, there is a good chance you have considered the case of a fair coin flip – one of the most intuitive applications of probability. In the absence of information on how the coin is flipped, we cannot be certain of the outcome. What we can be certain of is that with a large number of coin flips, the proportion of heads will become increasingly close to 50%, or $\frac{1}{2}$.

The **Law of Large Numbers (LLN)** is an important theorem for building an intuitive understanding of how probability relates to the statistical inference concepts we will cover. In the case of a fair coin flip, it is possible to observe many consecutive heads by chance. This is because small samples can lend to anomalies. However, as the number of flips increases, we will undoubtedly observe an increasing number of tails; we expect a roughly equal number of heads and tails with a large enough number of flips.

Probability Distributions

Probability distributions are statistical functions that yield the probability of obtaining possible values for a random variable. In other words, the underlying distribution describes how the values of a random variable vary. Probabilities range from 0 to 1, and the sum of all probabilities for all possible values must equal 1.

For example, let's look at how org tenure is distributed across employees. We can understand the general shape of the distribution using descriptive statistics:

```
# Load library for data wrangling
library(dplyr)

# Read employee demographics data
demographics <- read.csv("/Users/craig.starbuck/Library/Mobile Documents/com~apple~Clou

# Produce descriptive stats for org tenure
summary(demographics$org_tenure)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   8.000   9.193  13.000  20.000
```

Comparing the higher mean value of 9.2 to the median value of 8 indicates there are larger values skewing the mean upward which can be seen in the Q3 and max values.

Beyond descriptives, visuals are often helpful in understanding a variable's distribution. As shown in Figure @ref(fig:org-tenure-dist), it is clear that org tenure is positively skewed, and understanding the shape (or spread) of this distribution enables us to identify which values are most likely in order to estimate the likelihood of different results:

```
# Load library for data visualization
library(ggplot2)

# Visualize org tenure distribution
ggplot() +
  aes(demographics$org_tenure) +
  labs(x = "Org Tenure", y = "Density") +
  geom_histogram(aes(y = ..density..), fill = "#414141") +
  geom_density(fill = "#ADD8E6", alpha = 0.6) +
  theme_bw()
```

You can likely image the shape of probability distributions for many common events. If we consider the probability of employees exiting an organization, the outcome is binary. That is, employees either leave or stay; there are no options between these extremes. However, the distribution of performance scores will look quite different. Most organizations have expected – or even forced – distributions in which an average rating is awarded most frequently and low and

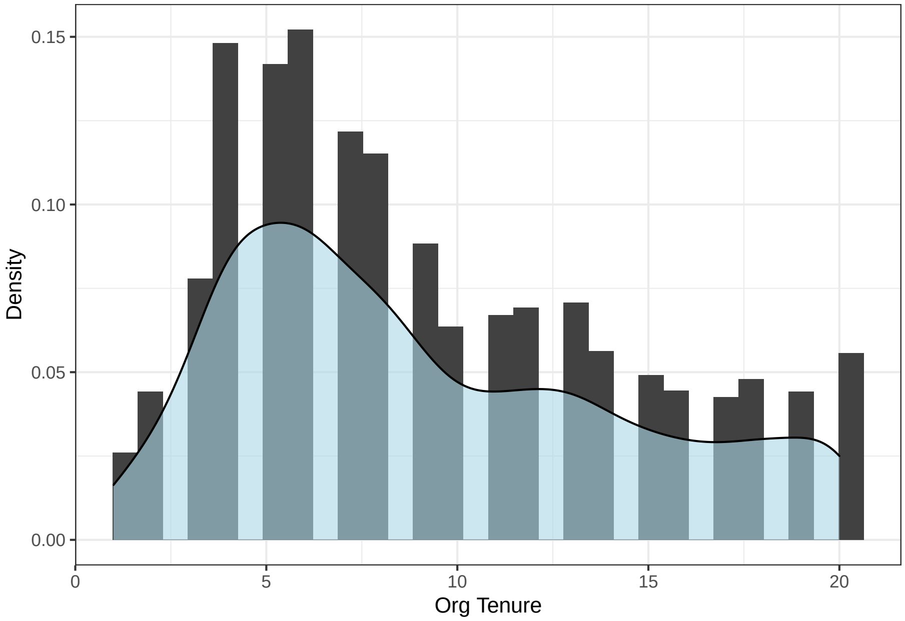


Figure 8.1: Org Tenure Distribution

high performance ratings less frequently. This would start to look more like a bell curve as the number of performance levels increases.

Just as we grouped variables into disrete and continuous categories in Chapter @ref(measure-sampl), this is also how probability distributions are categorized. If you read Chapter @ref(measure-sampl), you likely already have some a priori expectations about the characteristics of discrete and continuous distributions.

The shape of a probability distribution is defined by parameters, which represent its essential properties (e.g., measures of central tendency and spread). These probability distributions underpin the many types of statistical tests covered in this book.

Discrete Probability Distributions

Discrete probability distributions, also known as **Probability Mass Functions (PMF)**, can be leveraged to model different types of nominal and ordinal variables. Some common discrete distributions include:

- **Bernoulli**: probability of success or failure for a *single* trial with two outcomes
- **Binomial**: number of successes and failures in a *sequence* of independent trials with two outcomes (collection of Bernoulli trials)

- **Multinomial**: generalization of the binomial distribution for experiements with more than two outcomes
- **Negative Binomial (Pascal)**: a version of the binomial distribution for a *fixed* number of trials (this is positively skewed despite what the name might suggest)
- **Poisson**: probability of a given number of events occurring over a specified period
- **Geometric**: special case of the negative binomial distribution that repeats trials until a success is observed (rather than a fixed number of times)

A number of functions are available in R to simulate PMFs. The precise shape of a distribution depends on the parameters, but we will simulate and visualize these common PMFs to illustrate differences in the general shape of each. First, let's simulate the distributions by drawing 1,000 random values from each with a specified set of parameters:

```
# Set seed for reproducible random distribution
set.seed(1234)

# Simulate bernoulli distribution
bernoulli_dist <- rbinom(1000, 1, prob = .5)

# Simulate binomial distribution
# Notice the important difference relative to the Bernoulli simulation (100 trials vs.
binomial_dist <- rbinom(1000, 100, prob = .5)

# Simulate negative binomial distribution
nbinomial_dist <- rnbinom(1000, 100, prob = .5)

# Simulate multinomial distribution with varying probabilities per level
multinomial_dist <- rmultinom(1000, 4, prob = c(.4, .3, .2, .6))

# Simulate poisson distribution
poisson_dist <- rpois(1000, 10)

# Simulate geometric distribution
geometric_dist <- rgeom(1000, prob = .2)
```

Next, we will visualize each distribution:

```
# Load library for arranging visuals
library(ggpubr)

# Create user-defined function (UDF) to simplify probability distribution visualizatio
# Function arguments: (1) data = object containing random distribution values; (2) type = ’discre
dist.viz <- function(data, type, title) {

  if (type == "discrete"){

    # Discrete distribution
    viz <- ggplot() +
           aes(data) +
           labs(title = paste(title), x = "x", y = "count") +
           geom_histogram(fill = "#414141") +
           theme_bw() +
           theme(plot.title = element_text(hjust = 0.5))

  } else {

    # Continuous distribution
    viz <- ggplot() +
           aes(data) +
           labs(title = paste(title), x = "x", y = "density") +
           geom_histogram(aes(y = ..density..), fill = "#414141") +
           geom_density(fill = "#ADD8E6", alpha = 0.6) +
           theme_bw() +
           theme(plot.title = element_text(hjust = 0.5))
  }

  return(viz)
}

# Call UDF to build visualizations and store to objects
p_bernoulli <- dist.viz(data = bernoulli_dist, type = "discrete", title = "Bernoulli")
p_binomial <- dist.viz(data = binomial_dist, type = "discrete", title = "Binomial")
p_nbinomial <- dist.viz(data = nbinomial_dist, type = "discrete", title = "Negative Binomial")
p_multinomial <- dist.viz(data = multinomial_dist, type = "discrete", title = "Multinomial")
p_poisson <- dist.viz(data = poisson_dist, type = "discrete", title = "Poisson")
p_geometric <- dist.viz(data = geometric_dist, type = "discrete", title = "Geometric")

# Display distribution visualizations
ggarrange(p_bernoulli, p_binomial, p_nbinomial, p_multinomial, p_poisson, p_geometric,
          ncol = 3, nrow = 2)
```

Continuous Probability Distributions

Continuous probability distributions, also known as **Probability Density Functions (PDF)**, can be leveraged to model different types of interval and ratio variables. Some common continuous distributions include:

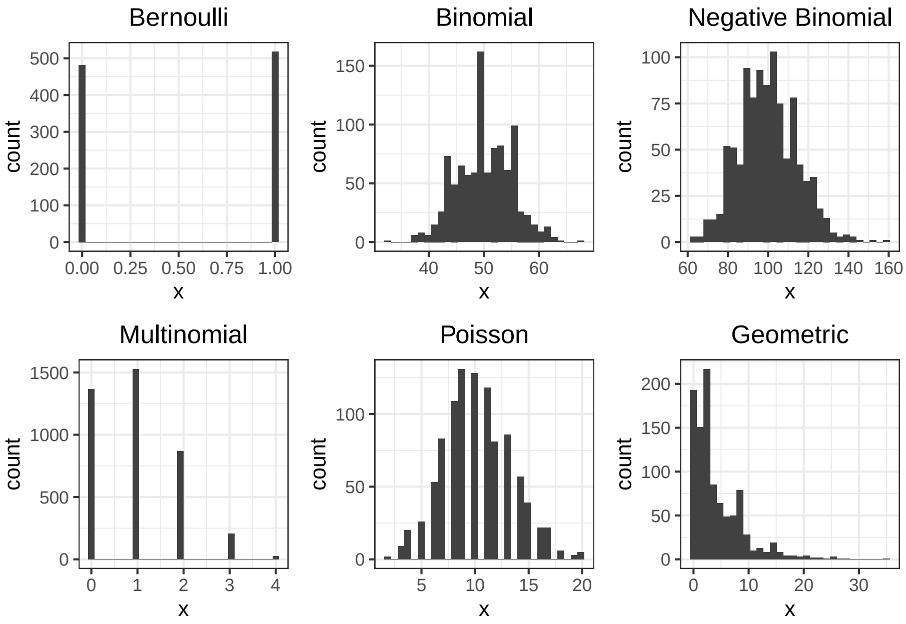


Figure 8.2: Discrete Probability Distributions

- **Normal (Gaussian)**: distribution characterized by a mean and standard deviation for which the mean, median, and mode are equal
- **Uniform**: values of a random variable with equal probabilities of occurring
- **Log-Normal**: normal distribution of log-transformed values
- **Student's T**: similar to the normal distribution but with thicker tails (approaches normal as n increases)
- **Chi-Square**: similar to the t distribution in that the shape approaches normal as n increases
- **F**: developed to examine variances from random samples taken from two independent normal populations

A number of functions are available in R to simulate PDFs. While the precise shape of a distribution depends on the parameters, we will simulate and visualize these common PDFs to illustrate differences in the general shape of each. Let's first draw 1,000 random values from each distribution with a specified set of parameters:

```
# Set seed for reproducible random distribution
set.seed(1234)

# Simulate normal distribution
```

```
normal_dist <- rnorm(1000, mean = 50, sd = 5)

# Simulate log-normal distribution
lnormal_dist <- rlnorm(1000, meanlog = 0, sdlog = 1)

# Simulate uniform distribution
uniform_dist <- runif(1000, min = 1, max = 100)

# Simulate student’s t distribution
t_dist <- rt(1000, df = 5)

# Simulate chi-square distribution
chisq_dist <- rchisq(1000, df = 5)

# Simulate F distribution
f_dist <- rf(1000, df1 = 5, df2 = 200)
```

Next, we will visualize each distribution. Since these continuous distributions are probability *density* functions, we will superimpose density plots over each:

```
# Call UDF to build visualizations and store to objects
# Note that as long as the arguments are in the order specified in the function (see our UDF defi
p_normal <- dist.viz(normal_dist, "continuous", "Normal")
p_lnormal <- dist.viz(lnormal_dist, "continuous", "Log-Normal")
p_uniform <- dist.viz(uniform_dist, "continuous", "Uniform")
p_t <- dist.viz(t_dist, "continuous", "Student’s T")
p_chisq <- dist.viz(chisq_dist, "continuous", "Chi-Square")
p_f <- dist.viz(f_dist, "continuous", "F")

# Display distribution visualizations
ggarrange(p_normal, p_lnormal, p_uniform, p_t, p_chisq, p_f,
          ncol = 3, nrow = 2)
```

The distribution of data is critically important in statistics. The accuracy of many statistical tests is based on assumptions rooted in underlying data distributions, and violating these assumptions can result in serious errors due to misaligned probability distributions. Though there are many more discrete and continuous probability distributions, we will leverage several of these common types to assess the likelihood of differences, effects, and associations in later chapters of this book.

Conditional Probability

Conditional probability reflects the probability conditioned on the occurrence of a previous event or outcome. For example, we may find that the pro-

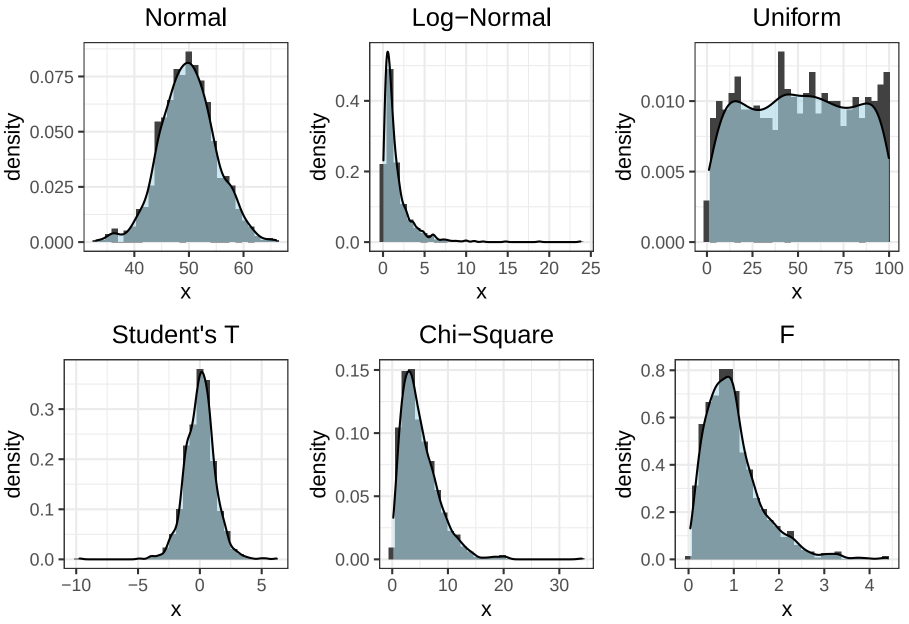


Figure 8.3: Continuous Probability Distributions

portion of heads is greater or less than $\frac{1}{2}$ with a large number of fair coin flips when the coin is consistently heads up when flipped. The outcome is, therefore, conditioned on the fixed – rather than random – positioning of the coin when flipped.

Formally, **Bayes' Theorem (alternatively, Bayes' Rule)** states that for any two events A and B wherein the probability of A is not 0 ($P(A) \neq 0$):

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

where:

- A = an event
- B = another event
- $P(A|B)$ = conditional probability that event A occurs, given event B occurs (posterior)
- $P(B|A)$ = conditional probability that event B occurs, given event A occurs (likelihood)
- $P(B)$ = normalizing constant, constraining the probability distribution to sum to 1 (evidence)
- $P(A)$ = probability event A occurs before knowing if event B occurs (prior)

Bayes' Rule allows us to predict the outcome more accurately by conditioning the probability on known factors rather than assuming all events operate under the same conditions. Bayes' Rule is pervasive in people analytics, as the probability of outcomes can vary widely when conditioned on a person's age, tenure, education, job, perceptions, relationships, and many other factors. For example, if we consider a company with 100 terminations over a 12-month period and average headcount of 1,000, the probability of attrition not conditioned on any other factor is 10%, or $\frac{1}{10}$. Aside from trending this probability over time to identify if overall attrition is becoming more or less of a concern, this isn't too helpful at the company level. However, if we condition the probability of attrition on an event – such as a recent manager exit – and find that the probability of attrition among those whose manager has left in the last six months is 70%, or $\frac{7}{10}$, this is far more actionable (and concerning).

The **Monty Hall Problem** is an excellent example of how our intuition is often at odds with the laws of conditional probability.

In the classic game show, Let's Make a Deal, Monty Hall asks contestants to choose one of three closed doors. Behind one door is a prize while the other two doors contain nothing. After the contestant selects a door, Monty opens one of the other two doors which does not contain a prize. At this point, there are two closed doors: the door the contestant selected and another for which the content remains unknown. All that is known at this point is that the prize is behind one of the two closed doors.

It is at this juncture that Monty introduces a twist by asking if the contestant would like to switch doors. Most assume that the two closed doors have an equal (50/50) chance of containing the prize, because we generally think of probabilities as independent, random events. However, this is incorrect. Contestants who switch from their original selection have a 66% chance (rather than 50%) of winning. This may be counterintuitive, because the brain wants to reduce the problem to a simple coin flip. There is a major difference between the Monty Hall problem and a coin flip; for two outcomes to have the same probability, randomness and independence are required. In the case of the Monty Hall problem, neither assumption is satisfied.

When all three doors are closed, each has the same probability of being selected. The probability of choosing the door with a prize is .33. Monty's knowledge of the door containing the prize does not impact the probability of selecting the winning door. This is because the choice is completely random given we have no information that would increase the probability of a door containing the prize. The process is no longer random when Monty uses his insider knowledge about the prize's location and opens a door he knows does not contain the prize. The probabilities change. Since Monty will never show the door containing the prize, he is careful to always open a door that has nothing behind it. If he was not constrained by the requirement to not reveal the prize's location and instead chose to open one of the remaining doors at random, the probabilities would be equal (and he may end up opening the door that contains the prize).

Seeing is believing, so let’s prove this with a simulation in R:

```r
# Set seed for reproducible simulations
set.seed(12345)

# Set number of simulations
trials = 10000

# Store switch/keep decisions
decisions = c("switch", "keep")

# Store integer for each door
doors = 1:3

# Initialize empty data frame for results
results = NULL

for (n in 1:trials){

  for (decision in decisions){

    # Select correct door
    correct_door <- sample(doors, 1, replace = T)

    # Contestant chooses a door at random
    selected_door <- sample(doors, 1, replace = T)

    # Open door that was neither selected by the contestant nor contains the prize
    # Choose one door to open if multiple remain without the prize (i.e., the contesta
    remaining_doors <- which(!doors == correct_door & !doors == selected_door)
    open_door <- sample(remaining_doors, 1, replace = T)

    # Contestant makes decision to switch doors or keep with the originally selected d
    selected_door <- ifelse(decision == "switch", which(!doors == selected_door & !doo

    # Store results in data frame
    results <- rbind(results, cbind.data.frame(
                     trial = n,
                     decision = decision,
                     result = ifelse(correct_door == selected_door, "win", "lose")))
  }
}

# Calculate percentage difference in wins for switch vs. keep decisions
switch_wins <- nrow(results[results$decision == "switch" & results$result == "win", ])
keep_wins <- nrow(results[results$decision == "keep" & results$result == "win", ]) / n
```

```
round((switch_wins - keep_wins) / keep_wins * 100, 0)
```

```
## [1] 45
```

As we can see, wins occur nearly 50% more often when contestants switch doors. This exercise hopefully demonstrates the importance of conditional probability and statistical assumptions like randomness. Also, if ever you find yourself playing Let's Make a Deal, switch doors.

Central Limit Theorem

The **Central Limit Theorem (CLT)** is a mainstay of statistics and probability and fundamental to understanding the mechanics of multivariate inferential analysis techniques we will cover later in this book. The CLT was initially coined by a French-born mathematician named Abraham De Moivre in the 1700s. While initially unpopular, it was later reintroduced and attracted new interest from theorists and academics (Daw & Pearson, 1972).

The CLT states that the average of independent random variables, when increased in number, tend to follow a normal (or Gaussian) distribution. The distribution of sample means approaches a normal distribution regardless of the shape of the population distribution from which the samples are drawn. This is important because the normal distribution has properties that can be used to test the likelihood that an observed value, difference, or relationship in a sample is also present in the population.

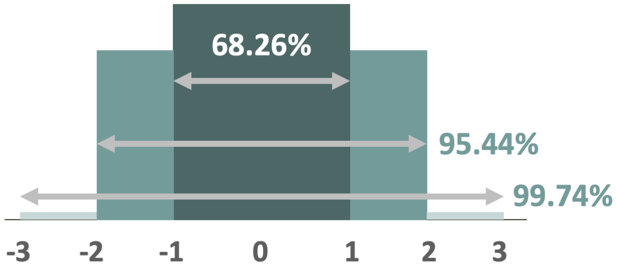


Figure 8.4: The Empirical Rule

Let's begin with an intuitive example of CLT. Imagine that we have a reliable way to measure how fun a population is on a 100-point scale, where 100 indicates maximum fun (life of the party) and 1 indicates maximum boringness. Consider that a small statistics conference is in progress at a nearby convention center, and there are 40 statisticians in attendance. In a separate room at the same convention center, there is also a group of 40 random people (non-statisticians) who are gathered to discuss some less interesting topic. Our job is to walk into one of the rooms and determine – based on the "fun" factor alone – whether we have entered the statistics conference or the other, less interesting gathering of non-statisticians.

Instinctively, we already know the statisticians will be more fun than the other group. However, let's assume we need the mean fun score and standard deviation of these two groups for this example. The group of statisticians have, on average, a fun score of 85 with a standard deviation of 2, while the group of non-statisticians are a bit less fun with a mean score of 65 and a standard deviation of 4. With a known population mean and standard deviation, the standard error (the standard deviation of the sample means) provides the ability to calculate the probability that the sample (the room of 40 people) belongs to the population of interest (fellow statisticians).

Herein lies the beauty of the CLT: roughly 68 percent of sample means will lie within one standard error of the population mean, roughly 95 percent within two standard errors of the population mean, and roughly 99 percent within three standard errors of the population mean. Therefore, any room whose members have an average fun score that is not within two standard errors of the population mean (between 81 and 89 for our statisticians) is statistically unlikely to be the group of statisticians for which we are searching. This is because in less than 5 in 100 cases could we randomly draw a 'reasonably sized' sample of statisticians with average funness so extremely different from the population average.

Because small samples lend to anomalies, we could – by chance – select a single person who happens to fall in the tails (extremely boring or extremely fun); however, as the sample size increases, it becomes more and more likely that the observed average reflects the average of the larger population. It would be virtually impossible (in less than 1 in 100 cases) to draw a random sample of statisticians from the population with average funness that is not within three standard errors of the population mean (between 79 and 91). Therefore, if we find that the room of people have an average fun score of 75, we will likely have far more fun in the other room!

Let's now see the CLT in action by simulating a random uniform population distribution from which we can draw random samples. Remember, the shape of the population distribution does not matter; we could simulate an Exponential, Gamma, Poisson, Binomial, or other distribution and observe the same behavior.

```
# Set seed for reproducible random distribution
set.seed(1234)

# Generate uniform population distribution with 1000 values ranging from 1 to 100
rand.unif <- runif(1000, min = 1, max = 100)
```

```
# Calculate population mean
mean(rand.unif)
```

```
## [1] 51.22007
```

```
# Calculate population variance
N = length(rand.unif)
var(rand.unif) * (N - 1) / N
```

```
## [1] 830.3155
```

```
# Produce histogram to visualize population distribution
ggplot() +
  aes(rand.unif) +
  labs(x = "x", y = "Density") +
  geom_histogram(aes(y = ..density..), fill = "#414141") +
  geom_density(fill = "#ADD8E6", alpha = 0.6) +
  theme_bw()
```

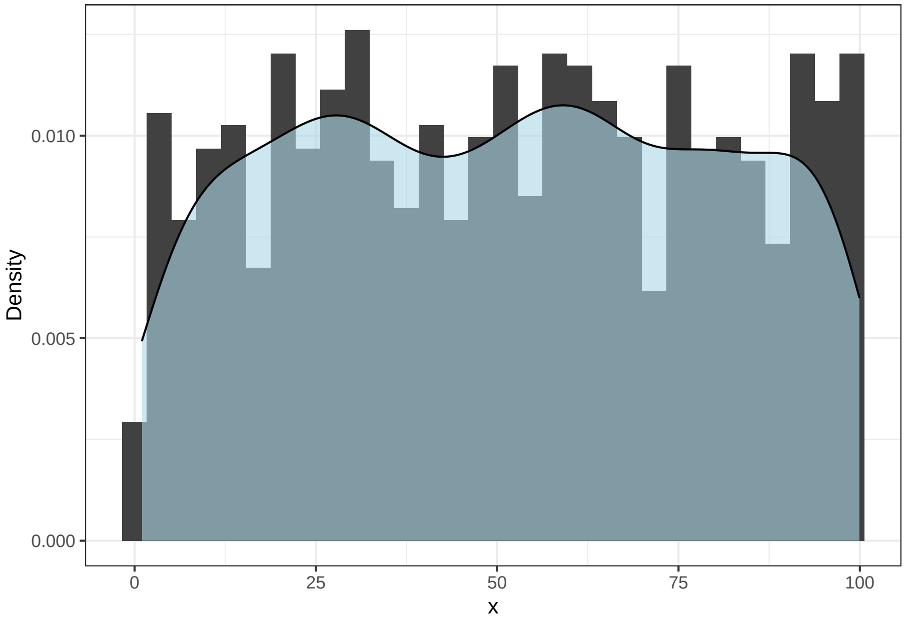


Figure 8.5: Uniform Population Distribution (N = 1000)

As expected, these randomly generated data are uniformly distributed. Next, we will draw 100 random samples of various sizes and plot the average of each.

```
# Define number of samples to draw from population distribution
samples <- 10000

# Populate vector with sample sizes
sample_n <- c(1:5,10,25,50)

# Initialize empty data frame to hold sample means
sample_means = NULL

# Set seed for reproducible random samples
set.seed(456)

# For each n, draw random samples
for (n in sample_n) {

  for (draw in 1:samples) {

      # Store sample means in data frame
      sample_means <- rbind(sample_means, cbind.data.frame(
                            n = n,
                            x_bar = mean(sample(rand.unif, n, replace = TRUE, prob = NU
  }
}

# Produce histograms to visualize distributions of sample means, grouped by n-count
sample_means %>% ggplot() +
  aes(x = x_bar, fill = n) +
  labs(x = "x-bar", y = "Density") +
  geom_histogram(aes(y = ..density..), fill = "#414141") +
  geom_density(fill = "#ADD8E6", alpha = 0.6) +
  theme_bw() +
  facet_wrap(~n)
```

Per the CLT, we can see that as n increases, the sample means become more normally distributed.

Confidence Intervals

A **Confidence Interval (CI)** is a range of values that likely contains the value of an unknown population parameter. These unknown population parameters are often μ or σ, though we will also leverage CIs in later chapters for regression coefficients, proportions, rates, and differences.

If we draw random samples from a population, we can compute a CI for each

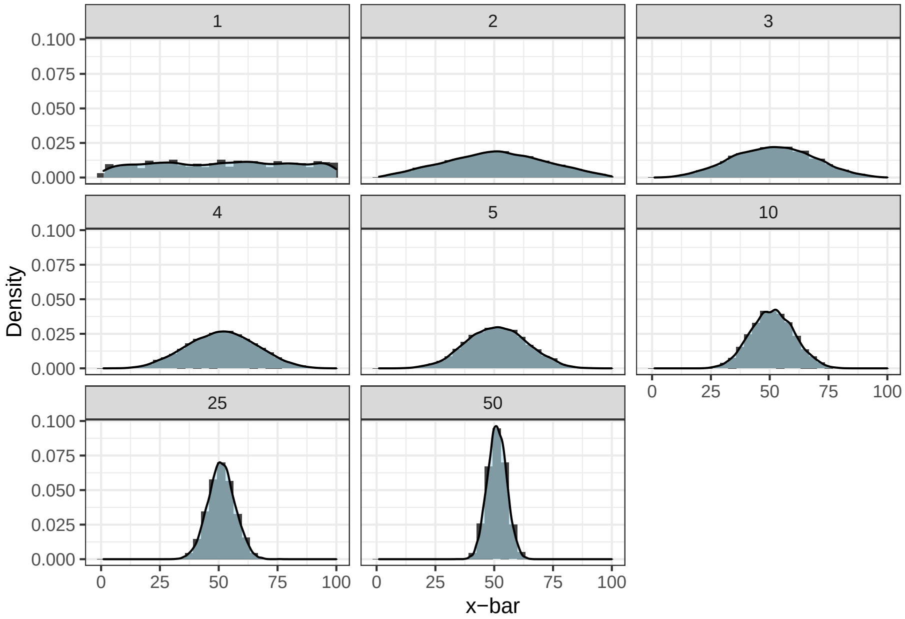


Figure 8.6: Distribution of 10,000 Sample Means of Varied Size

sample. Building on the CLT, for a given confidence level (usually 95%, though 99% or 90% are sometimes used), the specified percent of sample intervals is expected to include the estimated population parameter. For example, for a 95% CI we would expect 19 in every 20 (or 95 in every 100) intervals across the samples to include the true population parameter. This is illustrated in Figure @ref(fig:conf-int):

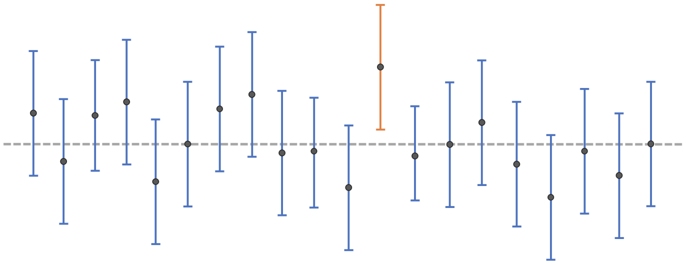


Figure 8.7: Intervals for 95 Percent Confidence Level

It is important to note that CIs should not be applied to the distribution of sample values; CIs relate to *population parameters*. A common misinterpretation of a CI is that it represents an interval within which a certain percent of sample values exist, and this is inaccurate. Because this misinterpretation is so prevalent, there is a good chance you will be tested on your understanding of CIs when applying to positions involving statistical analyses!

A related concept that is fundamental to estimating CIs is the **standard error (SE)**, which is the standard deviation of sample means. While the standard deviation is a measure of variability for a random variable, the variability captured by the SE reflects how well a sample represents the population. Since sample statistics will approach the actual population parameters as the size of the sample increases, the SE and sample size are inversely related; that is, the SE decreases as the sample size increases. The SE is defined by:

$$SE = \frac{\sigma}{\sqrt{n}}$$

Since the CLT is fundamental to inferential statistics, let's validate that our simulated distribution of sample means adheres to the properties of normally distributed data per the Empirical Rule:

```
# Store sample means with n = 50
x_bars <- sample_means[sample_means$n == 50, "x_bar"]

# Store sample size
n <- length(x_bars)

# Calculate percent of sample means within +/- 2 SEs
length(x_bars[x_bars < mean(x_bars) + 2 * sd(x_bars) & x_bars > mean(x_bars) - 2 * sd(
```

```
## [1] 95.35
```

95% of sample means are within 2 SEs, which is what we expect per the characteristics of the normal distribution.

```
# Calculate percent of sample means within +/- 3 SEs
length(x_bars[x_bars < mean(x_bars) + 3 * sd(x_bars) & x_bars > mean(x_bars) - 3 * sd(
```

```
## [1] 99.79
```

Nearly all of the sample means are within 3 SEs, indicating that it would be highly unlikely – nearly impossible even – to observe a sample mean 'from the same population' that falls outside this interval.

Now, let's illustrate the relationship between CIs and standard errors using sample data from our uniform population distribution. In our example, both μ and σ are known and our sample size n is at least 30; therefore, we can use a

Z-Test to calculate the 95% CI. A z score of 1.96 corresponds to the 95% CI for a two-tailed distribution; that is, we are looking for significantly different values in either the larger or smaller direction. The 95% CI represents the range of values we would expect to include μ in at least 95 of 100 random samples taken from the population.

The CI in this case is defined by:

$$CI = \bar{x} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Let's randomly take $n = 100$ from the population, and compute sample statistics to estimate the 95% CI:

```
# Set seed for reproducible random samples
set.seed(456)

# Sample 100 values from uniform population distribution
x <- sample(rand.unif, 100, replace = TRUE, prob = NULL)

# Calculate 95% CI
ci95_lower_bound <- mean(x) - 1.96 * (sd(x) / sqrt(100))
ci95_upper_bound <- mean(x) + 1.96 * (sd(x) / sqrt(100))
```

```
# Print lower bound for 95% CI
ci95_lower_bound
```

```
## [1] 47.90733
```

```
# Print upper bound for 95% CI
ci95_upper_bound
```

```
## [1] 58.98773
```

Our known μ is 51.2, which is covered by our 95% CI (47.9 - 59.0). Per the CLT, in less than 5% of cases would we expect to draw a random sample from the population that results in a 95% CI which does not include μ. Note that our CI narrows with larger samples since our confidence that the range includes μ increases with more data.

Next, let's look at a 99% CI. We will enter 2.576 for z:

```
# Calculate 99% CI
ci99_lower_bound <- mean(x) - 2.576 * (sd(x) / sqrt(100))
ci99_upper_bound <- mean(x) + 2.576 * (sd(x) / sqrt(100))
```

```
# Print lower bound for 99% CI
ci99_lower_bound
```

```
## [1] 46.16612
```

```
# Print upper bound for 99% CI
ci99_upper_bound
```

```
## [1] 60.72893
```

Like the 95% CI, this slightly wider 99% CI (46.2 - 60.7) also includes our μ of 51.2.

If σ is not known, and/or we have a small sample ($n < 30$), we need to use a **T-Test** to calculate the CIs. In a people analytics setting, the reality is that population parameters are often unknown. For example, if we knew how engagement scores vary in the employee population, there would be no need to survey a sample of employees and make inferences about said population.

As we will see, the T-Test underpins many statistical tests and models germane to the people analytics discipline since we are often working with small datasets, so it is important to understand the mechanics. As shown in Figure @ref(fig:t-distribution), the t distribution is increasingly wider and shorter relative to the normal distribution as the sample size decreases; this is also characteristic of the sampling distribution of means for smaller samples we observed in our CLT example. Specifically, **degrees of freedom (df)** is used to determine the shape of the probability distribution. Degrees of freedom represents the number of observations in the data that are free to vary when estimating statistical parameters, which is a function of the sample size ($n - 1$). For example, if we could choose 1 of 5 projects to work on each day between Monday and Friday, we would only be able to *choose* 4 out of the 5 days; on Friday, only 1 project would remain to be selected, so our degrees of freedom (the number of days in which we have a choice between projects) would be 4.

When estimating the CI for smaller samples, we need to leverage the wider, more platykurtic t distribution to achieve greater accuracy. Therefore, the CI for a two-tailed test in this case is defined by:

$$CI = \bar{x} \pm t_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Let's compare CIs calculated using a T-Test to those calculated using the Z-Test. While a fixed z score can be used for each CI level when $n > 30$, the t statistic varies based on both the CI level and df. Though R will determine the correct t statistic for us, let's reference the table shown in Figure @ref(fig:t-crit) to manually lookup the t statistic:

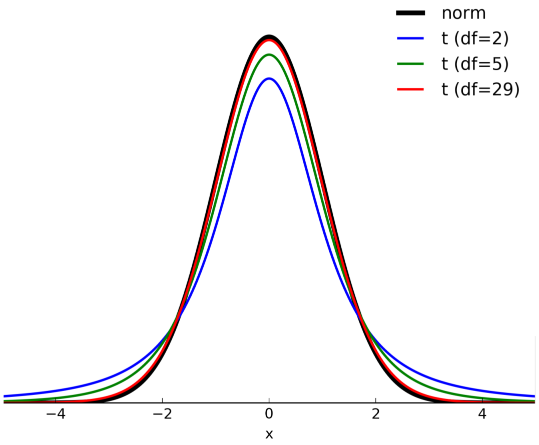


Figure 8.8: t Distribution Shape by Degrees of Freedom

For illustrative purposes, let's draw a smaller sample of $n = 25$ from our uniform population distribution and calculate the 95% CI using the t statistic from the table ($df = 24$). The t statistic for this CI and df is 2.064:

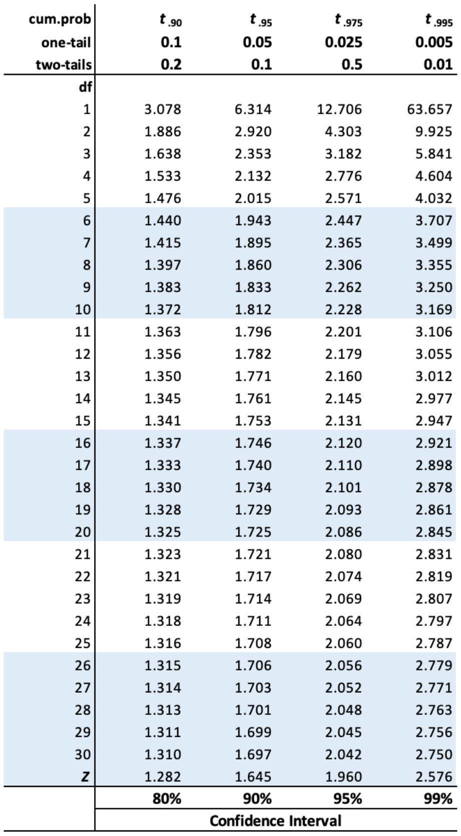
```
# Set seed for reproducible random samples
set.seed(456)

# Sample 25 values from uniform population distribution
x <- sample(rand.unif, 25, replace = TRUE, prob = NULL)

# Calculate 95% CI
ci95_lower_bound <- mean(x) - 2.064 * (sd(x) / sqrt(25))
ci95_upper_bound <- mean(x) + 2.064 * (sd(x) / sqrt(25))
```

```
# Print lower bound for 95% CI
ci95_lower_bound
```

```
## [1] 35.24305
```

```
# Print upper bound for 95% CI
ci95_upper_bound
```

```
## [1] 59.60959
```



cum.prob	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.995}$
one-tail	0.1	0.05	0.025	0.005
two-tails	0.2	0.1	0.5	0.01
df				
1	3.078	6.314	12.706	63.657
2	1.886	2.920	4.303	9.925
3	1.638	2.353	3.182	5.841
4	1.533	2.132	2.776	4.604
5	1.476	2.015	2.571	4.032
6	1.440	1.943	2.447	3.707
7	1.415	1.895	2.365	3.499
8	1.397	1.860	2.306	3.355
9	1.383	1.833	2.262	3.250
10	1.372	1.812	2.228	3.169
11	1.363	1.796	2.201	3.106
12	1.356	1.782	2.179	3.055
13	1.350	1.771	2.160	3.012
14	1.345	1.761	2.145	2.977
15	1.341	1.753	2.131	2.947
16	1.337	1.746	2.120	2.921
17	1.333	1.740	2.110	2.898
18	1.330	1.734	2.101	2.878
19	1.328	1.729	2.093	2.861
20	1.325	1.725	2.086	2.845
21	1.323	1.721	2.080	2.831
22	1.321	1.717	2.074	2.819
23	1.319	1.714	2.069	2.807
24	1.318	1.711	2.064	2.797
25	1.316	1.708	2.060	2.787
26	1.315	1.706	2.056	2.779
27	1.314	1.703	2.052	2.771
28	1.313	1.701	2.048	2.763
29	1.311	1.699	2.045	2.756
30	1.310	1.697	2.042	2.750
z	1.282	1.645	1.960	2.576
	80%	90%	95%	99%
	Confidence Interval			

Figure 8.9: Critical Values of Student's t Distribution

As expected, the 95% CI using the t statistic is much wider (35.2 - 59.6), acknowledging the increased uncertainty in estimating population parameters given the limited information in this smaller sample. To increase our confidence to the 99% level, the interval widens even further (30.9 - 63.9):

```
# Calculate 99% CI
ci99_lower_bound <- mean(x) - 2.797 * (sd(x) / sqrt(25))
ci99_upper_bound <- mean(x) + 2.797 * (sd(x) / sqrt(25))
```

```
# Print lower bound for 99% CI
ci99_lower_bound
```

```
## [1] 30.91633
```

```
# Print upper bound for 99% CI
ci99_upper_bound
```

```
## [1] 63.93631
```

Hypothesis Testing

Hypothesis testing is how we leverage CIs to test whether a significant difference or relationship exists in the data. Sir Ronald Fisher invented what is known as the null hypothesis, which states that there is no relationship/difference; disprove me if you can! The null hypothesis is defined by:

$$H_0 : \mu_A = \mu_B$$

The objective of hypothesis testing is to determine if there is sufficient evidence to reject the null hypothesis in favor of an alternative hypothesis. The null hypothesis always states that there is 'nothing' of significance. For example, if we want to test whether an intervention has an effect on an outcome in a population, the null hypothesis states that there is no effect. If we want to test whether there is a difference in average scores between two groups in a population, the null hypothesis states that there is no difference.

An alternative hypothesis may simply state that there is a difference or relationship in the population, or it may specify the expected direction (e.g., Population A has a significantly 'larger' or 'smaller' average value than Population B; Variable A is 'positively' or 'negatively' related to Variable B). Therefore, alternative hypotheses are defined by:

$$H_A : \mu_A \neq \mu_B$$

$$H_A : \mu_A < \mu_B$$

$$H_A : \mu_A > \mu_B$$

Alpha

The **alpha** level of a hypothesis test, denoted by α, represents the probability of obtaining observed results due to chance if the null hypothesis is true. In other words, α is the probability of rejecting the null hypothesis (and therefore claiming that there is a significant difference or relationship) when in fact we should have failed to reject it because there is insufficient evidence to support the alternative hypothesis.

α is often set at .05 but is sometimes set at a more rigorous .01, depending upon the context and tolerance for error. An α of .05 corresponds to a 95% CI (1 - .05), and .01 to a 99% CI (1 - .01). With non-directional alternative hypotheses, we must divide α by 2 (i.e., we could observe a significant result in either tail of the distribution), while one-tailed tests position the rejection region entirely within one tail based on what is being hypothesized.

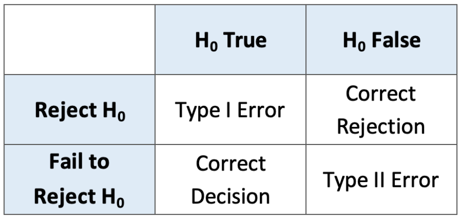
At the .05 level, we would conclude that a finding is statistically significant if the chance of observing a value at least as extreme as the one observed is less than 1 in 20 if the null hypothesis is true. Note that we observed this behavior with our simulated distribution of sample means. While we could observe more extreme values by chance with repeated attempts, in less than 1 in every 20 times would we expect a 95% CI that does not capture μ. Moreover, in less than 1 in every 100 times should we expect a sample with a 99% CI that does not capture μ.

Beta

Another key value is **Beta**, denoted by β, which relates to the power of the analysis. Simply put, power reflects our ability to find a difference or relationship if there is one. Power is calculated by 1 - β. At this point, it should be intuitive that larger samples increase our chances of observing significant results. As we observed in the T-Test example, CIs for small samples ($n < 30$) are quite wide relative to those for large samples; therefore, the power of the analysis to detect significance is limited given how extremely different values of x must be to observe non-overlapping CIs.

Type I & II Errors

A **Type I Error** is a false positive, wherein we conclude that there is a significant difference or relationship when there is not. A **Type II Error** is a false negative, wherein we fail to capture a significant finding. α represents our chance of making a Type I Error, while β represents our chance of making a Type II Error. I once had a professor explain that committing a Type I error is a shame, while committing a Type II error is a pity, and I've found this to be a helpful way to remember what each type of error represents.



	H_0 True	**H_0 False**
Reject H_0	Type I Error	Correct Rejection
Fail to Reject H_0	Correct Decision	Type II Error

Figure 8.10: Type I and II Errors

P-Values

In statistical tests, the **p-value** is referenced to determine whether the null hypothesis can be rejected. The p-value represents the probability of obtaining a result at least as extreme as the one observed if the null hypothesis is true. As a general rule, if $p < .05$, we can confidently reject the null hypothesis and conclude that the observed difference or relationship was unlikely a chance observation.

While statistical significance helps us understand the probability of observing results by chance when there is no difference or effect in the population, it does not tell us anything about the size of the difference or effect. Analysis should never be reduced to inspecting p-values; in fact, p-values have been the subject of much controversy among researchers and practitioners in recent years. Later chapters will cover how to interpret results of statistical tests to surface the story and determine if there is anything 'practically' significant among statistically significant findings.

Bonferroni Correction

One caveat when leveraging a p-value to determine statistical significance is that when multiple testing is performed – that is, multiple tests using the same sample data – the probability of a Type I error increases by a factor equivalent to the number of tests performed. It's important to note that there is not agreement among statisticians about how (or even whether) the p-value threshold for statistical significance needs to be adjusted to account for this increased risk. Nevertheless, we will cover this conservative approach for mitigating this risk.

Thus far, we have only discussed statistical significance in the context of a **per analysis error rate** – that is, the probability of committing a Type I error for a single statistical test. However, when two or more tests are being conducted on the same sample, the **familywise error rate** is an important factor in determining statistical significance. The familywise error rate reflects the fact that as we conduct more and more analyses on the same sample, the probability of a Type I error across the set (or family) of analyses increases. The familywise error rate can be calculated by:

$$\alpha_{FW} = 1 - (1 - \alpha_{PC})^C,$$

where c is equal to the number of comparisons (or statistical tests) performed, and α_{PC} is equal to the specified per analysis error rate (usually .05). For example, if $\alpha = .05$ per analysis, the probability of a Type I error with three tests on the same data increases from 5% to 14.3%: $1 - (1 - .05)^3 = .143$.

The most common method of adjusting the familywise error rate down to the specified per analysis error rate is the **Bonferroni Correction**. To implement this correction, we can simply divide α by the number of analyses performed on the dataset – such as $\alpha/3 = .017$ in the case of three analyses with $\alpha = .05$. This means that for each statistical test, we must achieve $p < .017$ to report a statistically significant result. An alternative which allows us to achieve the same number of statistically significant results is to multiply the unadjusted per analysis p-values for each statistical test by the number of tests. For example, if we run three statistical tests and receive $p = .014$, $p = .047$, and $p = .125$, we would achieve one significant result with the first method ($p < .017$) as well as with the alternative since the first statistical test satisfies the per analysis error rate ($p < .05$): $p = .014 * 3 = .042$.

Perneger (1998) is one of many who oppose the use of the Bonferroni Correction, suggesting that these "adjustments are, at best, unnecessary and, at worst, deleterious to sound statistical inference." The Bonferroni Correction is controversial among researchers because while applying the correction reduces the chance of a Type I error, it also increases the chance of a Type II error. Because this correction makes it more difficult to detect significant results, it is rare to find such a correction reported in published research, though research often involves multiple testing on the same sample. Perneger suggests that simply describing the statistical tests that were performed, and why, is sufficient for dealing with potential problems introduced by multiple testing.

Parametric vs. Nonparametric Tests

Parametric statistics assume the population is normally distributed. **Nonparametric statistics** do not assume anything about the population parameters or distribution and are, therefore, often referred to as distribution free tests. Assuming the normality assumption holds, parametric tests generally have more power than their nonparametric counterparts. This means that with a nonparametric test, we are less likely to reject the null hypothesis when it is false if the data come from normally distributed populations.

Since the mean (expected value) is the most common measure of central tendency, parametric tests usually focus on comparing the mean or variance of data. You may recall that μ and σ are sufficient to characterize a population distribution when data are situated symmetrically around the mean. However, the mean can be sensitive to outliers. If outliers are present in the data, the median may be a better way of representing the central tendency of data; in this case, nonparametric tests may be more appropriate.

In addition to normally distributed data in the population, and ensuring outliers are not materially influencing the mean, parametric tests also assume **homogeneity of variance** and **independence**. Homogeneity of variance assumes the variances across multiple groups are equal, though parametric tests are generally robust to violations of equal variances when the sample sizes are large. The assumption of independence requires observations to be randomly sampled from the population and independent of one another; that is, the value of one observation does not influence or depend on the value of another.

Spearman's correlation coefficient, which we used to evaluate the relationship between job level and education in Chapter @ref(uni-bi-stats), is a nonparametric test since ordinal data are usually not normally distributed in the population. There is a nonparametric equivalent for each parametric test, and these will be reviewed in detail in Chapter @ref(aod).

It's important to remember that the normal distribution properties under the CLT relate to the sampling distribution of means – not to the distribution of the population or to the data for one individual sample. The CLT is important for estimating population parameters, but it does not transform a population distribution from nonnormal to normal. If we know the population distribution is nonnormal (e.g., ordinal, nominal, or skewed data), nonparametric tests should be leveraged.

Exercises

1. Which of the following is an example of a null hypothesis, where μ reflects the mean of a population? A. $\mu_A = \mu_B$ B. $\mu_A \neq \mu_B$ C. $\mu_A < \mu_B$ D. $\mu_A > \mu_B$ E. None of the above; all are examples of alternative hypotheses.

2. Which of the following describes a Type I Error? A. Failing to reject the null hypothesis when it is false B. Rejecting the null hypothesis when it is true C. Reporting something as significant when nothing of significance is present (a shame) D. Failing to detect something of significance (a pity) E. Both B and C F. Both A and D

3. The primary purpose of inferential statistics is to make inferences about a population based on sample data. Inferential statistics allows these inferences to be made with defined levels of confidence that what is observed in a sample is also characteristic of the larger population. A. True B. False

4. A T-Test should be used when σ is unknown and/or $n < 30$. A. True B. False

5. Randomness is essential to probabilistic methods. A. True B. False

6. Which of the following is characteristic of the Bonferroni Correction? A. Reducing the risk of a Type I Error B. Increasing the risk of a Type II Error C. Reducing the familywise error rate

7. Tolerance for a wider interval is an important tradeoff decision when increasing the level of confidence that a range of values contains an unknown population parameter. A. True B. False

8. A *CI* represents the range of values we expect to include an unknown population parameter (often the mean) for a specified degree of confidence. A. True B. False

9. When population parameters are unknown, which of the following tests is most appropriate for testing $\mu_A = \mu_B$? A. Z-Test B. T-Test C. Bayes' Theorem D. P-Test

10. According to the Empirical Rule, 95% of normally distributed data lie within how many standard deviations of the mean? A. 1 B. 2 C. 3 D. 4

Chapter 9

Data Wrangling and Preparation

To begin a data analysis, we must first access, integrate, and clean the relevant data. These tasks account for as much as 80% of a data scientist's work.

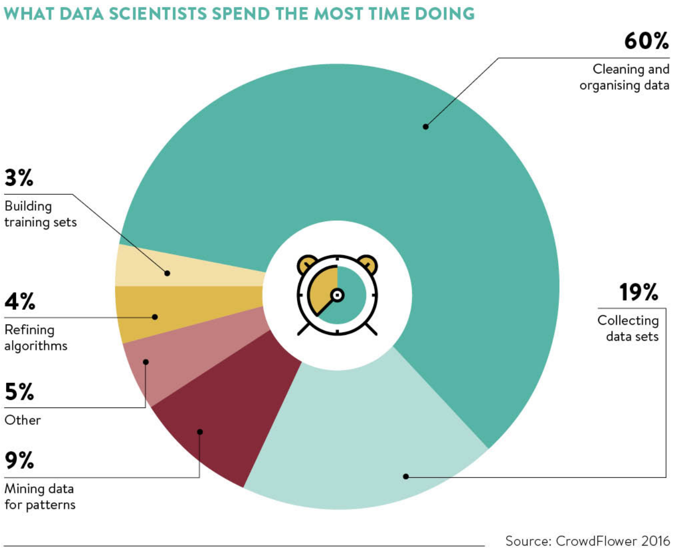


Figure 9.1: What Data Scientists Really Do

Data Management

Data are extracted directly from the source systems in which they are generated or from downstream repositories such as a *data lake*, *data warehouse*, or *data mart*.

A **data lake** stores myriad types of data – both structured and unstructured – in its native format until needed. **Structured data** refers to data that fits a predefined data model, such as hire dates formatted as `MM/DD/YYYY` or zip codes stored as a five-digit string. **Unstructured data** has no predefined data model, such as audio and video files, free-form performance review comments, emails, or digital exhaust from messenging tools like Slack; it is difficult to structure this type of data within a set of related tables.

The main difference between a data lake and data warehouse is the type of data they are designed to store. A **data warehouse (DW)** is designed to support analytics across large collections of data, such as transactional data (e.g., point-of-sale systems), point-in-time snapshots (e.g., month-end close reports), survey responses, spreadsheets, and more. As shown in Figure @ref(fig:dw-schema), data in a DW are structured and organized into schemas of related tables to enhance the performance of queries spanning sets of large and diverse data.

Figure @ref(fig:dw-schema) illustrates how worker, position, and candidate schemas may be related. For example, a candidate submits a job application to a posted requisition, which is connected to an open position Finance approved as part of the company's workforce plan; when the selected candidate is hired, they become a worker with one or many events (hire, promotion, transfer, termination) and activities (learning, performance appraisals, surveys) during their tenure with the company.

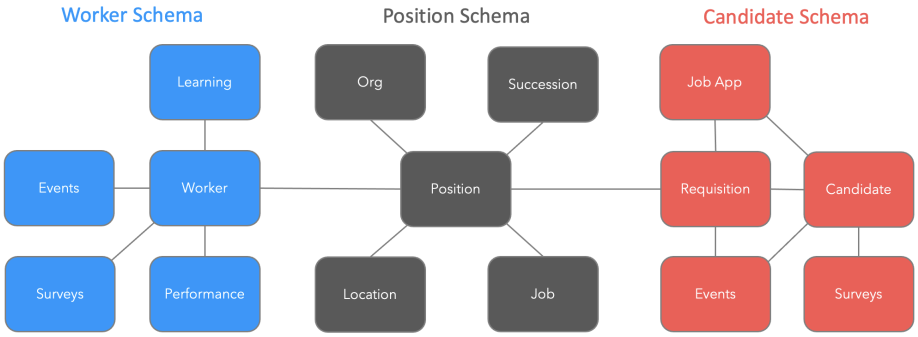
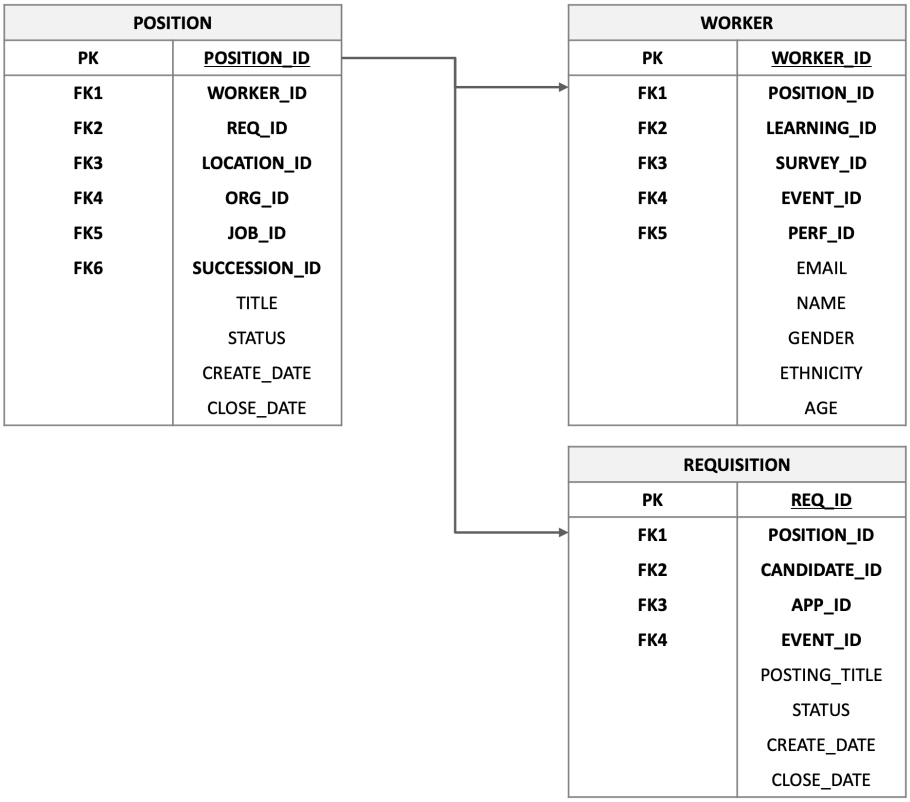


Figure 9.2: Related Tables Organized within Schemas

Tables in a DW are related using a set of keys. Each table needs a **Primary Key (PK)**, which is a unique identifier for each row in the table. A PK may be a single column or multiple columns; a multi-column PK is often referred to as a *composite key*. It is generally best to leverage non-recyclable system-generated ids for PKs, as descriptors like names tend to be unreliable. A **Foreign Key (FK)** is a column whose values correspond to the values of a PK in another table. **Referential integrity** is the logical dependency of a FK on a PK, and these constraints protect against orphaned FK values in child tables by deleting PK values from an associated parent table.

Figure @ref(fig:dw-erd) shows an **Entity Relationship Diagram (ERD)** that depicts PK/FK relationships among the Position, Worker, and Requisition tables. Notice that the PK of each related table shown in Figure @ref(fig:dw-schema) is listed as a FK.



POSITION	
PK	POSITION_ID
FK1	WORKER_ID
FK2	REQ_ID
FK3	LOCATION_ID
FK4	ORG_ID
FK5	JOB_ID
FK6	SUCCESSION_ID
	TITLE
	STATUS
	CREATE_DATE
	CLOSE_DATE

WORKER	
PK	WORKER_ID
FK1	POSITION_ID
FK2	LEARNING_ID
FK3	SURVEY_ID
FK4	EVENT_ID
FK5	PERF_ID
	EMAIL
	NAME
	GENDER
	ETHNICITY
	AGE

REQUISITION	
PK	REQ_ID
FK1	POSITION_ID
FK2	CANDIDATE_ID
FK3	APP_ID
FK4	EVENT_ID
	POSTING_TITLE
	STATUS
	CREATE_DATE
	CLOSE_DATE

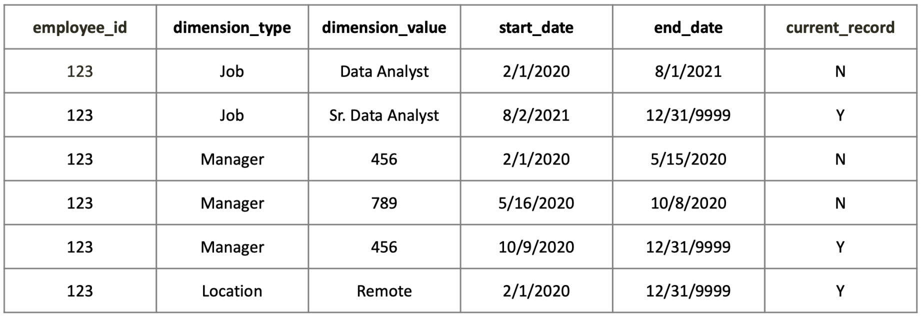
Figure 9.3: Entity Relationship Diagram (ERD)

A DW usually contains many different types of tables, and the two most common are *Type 1* and *Type 2* tables. These tables are sometimes referred to as **slowly changing dimensions (SCD)**. A **Type 1 table** is overwritten on a regular cadence (usually daily) and contains no history – only current values. For example, a Type 1 table may contain the latest known attributes for each active and terminated worker such as job, location, and manager. A Type 1 table is convenient when only current information needs to be queried. A **Type 2**

table is a table in which a new record is inserted when a change occurs for one or more specified dimensions. Jobs, managers, and locations are examples of slowly changing dimensions but unlike the Type 1 table which is truncated and reloaded with the latest information, the Type 2 table houses a *start date* and *end date* for each worker and dimension to facilitate reporting and analysis on changes to attribute values over time.

Figure @ref(fig:type-2-tbl) illustrates the design of a Type 2 SCD for an active worker's job, manager, and location changes. As the data show, worker 123 was promoted from Data Analyst to Sr. Data Analyst 1.5 years after joining, began reporting to their original manager (456) after a short stint reporting to someone else (789), and has worked remotely throughout their tenure with the company.

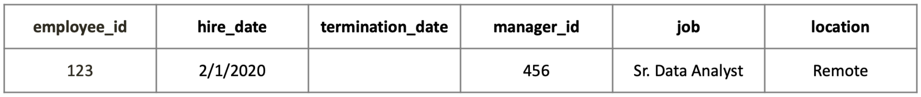
Note that `end_date = '12/31/9999'` and `current_record = 'Y'` for rows that represent *current attributes* for active workers. For inactive workers, `end_date` would be set to the worker's termination date for rows that represent the *last known attributes*, along with `current_record = 'Y'`:



employee_id	dimension_type	dimension_value	start_date	end_date	current_record
123	Job	Data Analyst	2/1/2020	8/1/2021	N
123	Job	Sr. Data Analyst	8/2/2021	12/31/9999	Y
123	Manager	456	2/1/2020	5/15/2020	N
123	Manager	789	5/16/2020	10/8/2020	N
123	Manager	456	10/9/2020	12/31/9999	Y
123	Location	Remote	2/1/2020	12/31/9999	Y

Figure 9.4: Type 2 SCD

While we could select rows where `current_record = 'Y'` to construct a view of the last known attributes for each worker in this table, each worker would have a separate row for each dimension which isn't ideal. Using a Type 1 table simplifies the task and output, as each dimension value is stored in a separate column. Figure @ref(fig:type-1-tbl) shows how the current record for worker 123 would look in a Type 1 SCD:



employee_id	hire_date	termination_date	manager_id	job	location
123	2/1/2020		456	Sr. Data Analyst	Remote

Figure 9.5: Type 1 SCD

A **data mart** is a subset of a DW designed to easily deliver specific information to a certain set of users on a particular subject or for a well-defined use case. For example, in a people analytics context a diversity data mart could be developed to better isolate and secure restricted data such as gender and ethnicity. This data may be used to support diversity descriptives and trends for a limited audience approved by Data Privacy and Employment Law counsel based on justified business needs.

SQL

Structured Query Language (SQL) is the most common language used to extract and wrangle data contained in a relational database. SQL is an essential skill for anyone working in analytics.

When working with large datasets, it is best to filter records on the database side to avoid reading superfluous records into an analytics tool such as R only to then filter data to the relevant subset. For example, if we are performing an analysis on employees in the Engineering department, we should filter to this subset on the database side rather than load data on the entire workforce into R before filtering to the relevant records. Fewer records can help enhance the performance of R scripts – especially when R is running on a local machine, such as a laptop, rather than on a more powerful server.

Basics

There are three main *clauses* in a SQL query: (a) `SELECT`, (b) `FROM`, and (c) `WHERE`. The `SELECT` and `FROM` clauses are required, though the optional `WHERE` clause is frequently utilized.

- **SELECT**: Specifies the columns to include in the output
- **FROM**: Specifies the table(s) in which the relevant data are contained
- **WHERE**: Specifies the rows to search

Despite the clauses being ordered as shown above (`SELECT` then `FROM` then `WHERE`), the `FROM` clause is actually the first to execute since we first need to identify the relevant table(s) before filtering rows and selecting columns. The `SELECT` clause is the last to execute.

Additional clauses are available for grouping and sorting data.

- **GROUP BY**: Specifies the columns by which data should be grouped when using aggregate functions
- **HAVING**: Specifies conditions for filtering rows based on aggregate functions
- **ORDER BY**: Specifies how data should be sorted in the output

When implementing aggregate functions in a `SELECT` clause, such as counting, summing, or averaging a numeric field, all other non-aggregated fields must be included in the `GROUP BY` clause.

Though it is important to execute SQL queries directly on the database to minimize the amount of data read into R, we will use the `sqldf` library within R to demonstrate the mechanics of a SQL query. The `sqldf` library allows us to write SQL to query a data frame in R in lieu of a database. While the syntax of SQL may vary by database, the core structure of queries is univeral.

First, let's load the `sqldf` library and store data to a data frame named `demographics`:

```
# Load SQL library
library(sqldf)

# Read employee demographics data
demographics <- read.csv("/Users/craig.starbuck/Library/Mobile Documents/com~apple~Clou

# Return row and column counts
dim(demographics)
```

```
## [1] 4875   22
```

Next, we will apply the `sqldf` function to our data frame to extract specific rows and columns. In addition to the `SELECT`, `FROM`, and `WHERE` clauses, we will use the `LIMIT` clause to limit the number of rows that are displayed given the data frame's size ($n = 4,875$). In a practical setting, the `LIMIT` clause is only used for efficient data profiling and troubleshooting, as we would not want to arbitrarily truncate a data set used for analysis.

A best practice in writing SQL is to capitalize the names of clauses and functions and to use separate lines and indentation to make the SQL statements more readable:

```
# Store SQL query as a character string using the paste() function
sql_string <- paste("SELECT
                      employee_id
                    FROM
                      demographics
                    WHERE
                      team = 'Social Marketing'
                    LIMIT 10")

# Execute SQL query
sqldf(sql_string)
```

```
##    employee_id
## 1         1013
## 2         1034
## 3         1035
## 4         1036
## 5         1037
## 6         1038
## 7         1139
## 8         1140
## 9         1141
## 10        1142
```

This query returned a list of employee ids for the 496 employees on the Social Marketing team.

Aggregate Functions

Next, let's take a look at the average organization tenure by job for those on the Social Marketing team:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      AVG(org_tenure)
                    FROM
                      demographics
                    WHERE
                      team = 'Social Marketing'
                    GROUP BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                        job_title AVG(org_tenure)
## 1          Community Coordinator        9.224138
## 2             Content Strategist        8.152778
## 3             Creative Assistant        9.454545
## 4                   Data Analyst        8.666667
## 5  Digital Marketing Specialist        8.228571
## 6                       Director        8.100000
## 7                        Manager        9.340000
## 8                 SEO Specialist        9.596774
## 9        Social Media Strategist       10.000000
## 10                Vice President        7.500000
```

There are 10 distinct job titles among employees on the Social Marketing team, and the average organization tenure for these ranges from 7.5 to 10.

Since there could be few employees in certain jobs, in which case average organization tenure may not be as meaningful, we can use the COUNT(*) function to count the number of rows for each group. In this case, COUNT(*) will return the number of employees in each job on the Social Marketing team. We will also utilize aliases via AS in the SELECT clause to assign different names to the output fields:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      demographics
                    WHERE
                      team = ’Social Marketing’
                    GROUP BY
                      job_title
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                     job_title employee_cnt avg_org_tenure
## 1       Community Coordinator           58       9.224138
## 2          Content Strategist           72       8.152778
## 3          Creative Assistant           44       9.454545
## 4                Data Analyst           72       8.666667
## 5  Digital Marketing Specialist           70       8.228571
## 6                    Director           10       8.100000
## 7                     Manager           50       9.340000
## 8              SEO Specialist           62       9.596774
## 9     Social Media Strategist           56      10.000000
## 10             Vice President            2       7.500000
```

The output shows that there are only 2 Vice Presidents and 10 Directors on the Social Marketing team, while other job titles are much more common.

Since relatively few employees are Vice Presidents or Directors, let’s use the HAVING clause to only show average organization tenure for Social Marketing team jobs with more than 10 employees:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      demographics
                    WHERE
                      team = 'Social Marketing'
                    GROUP BY
                      job_title
                    HAVING
                      COUNT(*) > 10
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                      job_title employee_cnt avg_org_tenure
## 1        Community Coordinator           58       9.224138
## 2           Content Strategist           72       8.152778
## 3           Creative Assistant           44       9.454545
## 4                 Data Analyst           72       8.666667
## 5 Digital Marketing Specialist           70       8.228571
## 6                      Manager           50       9.340000
## 7               SEO Specialist           62       9.596774
## 8      Social Media Strategist           56      10.000000
```

Joins

In a practical setting, the required data are rarely contained within a single table. Therefore, we must query multiple tables and join them together.

To illustrate how SQL joins work, let's partition the fields used in the previous SQL queries into three separate data frames. For many people analytics projects, employee id is the PK since this identifer for one employee should not be assigned to another employee – past, present, or future. Email or network id may also be a suitable PK. We will ensure the employee_id column is present in each of the three data frames to facilitate subsequent joins:

```
# Partition select columns into three data frames
job_df <- demographics[, c("employee_id", "job_title")]
tenure_df <- demographics[, c("employee_id", "org_tenure")]
team_df <- demographics[, c("employee_id", "team")]
```

Next, we will query our three new dataframes to return the average organization tenure for Social Marketing team jobs with more than 10 employees. For this, we will leverage an `INNER JOIN`, which will return records only for employee ids which are present in all three data frames. For example, if a record exists in `job_df` and `tenure_df` for a particular employee id, but there is no team designation for it in `team_df`, that employee id would not be included in the output.

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      job_df
                        INNER JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                        INNER JOIN
                      team_df
                        ON
                      job_df.employee_id = team_df.employee_id
                    WHERE
                      team = 'Social Marketing'
                    GROUP BY
                      job_title
                    HAVING
                      COUNT(*) > 10
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                      job_title employee_cnt avg_org_tenure
## 1        Community Coordinator           58       9.224138
## 2           Content Strategist           72       8.152778
## 3           Creative Assistant           44       9.454545
## 4                 Data Analyst           72       8.666667
## 5 Digital Marketing Specialist           70       8.228571
## 6                      Manager           50       9.340000
## 7               SEO Specialist           62       9.596774
## 8      Social Media Strategist           56      10.000000
```

Note that the `INNER JOIN` in this SQL query was structured such that both tenure_df and team_df were joined to job_df via the employee_id column. We could have instead joined job_df to team_df and achieved the same result since all employee ids exist in all three data frames in this case.

If it were possible for all employee ids to exist in job_df but not in either tenure_df or team_df, we could leverage a LEFT JOIN to ensure all records from job_df are included in the output irrespective of whether they have matches in tenure_df or team_df:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      job_df
                        LEFT JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                        LEFT JOIN
                      team_df
                        ON
                      job_df.employee_id = team_df.employee_id
                    WHERE
                      team = 'Social Marketing'
                    GROUP BY
                      job_title
                    HAVING
                      COUNT(*) > 10
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                       job_title employee_cnt avg_org_tenure
## 1        Community Coordinator           58       9.224138
## 2           Content Strategist           72       8.152778
## 3           Creative Assistant           44       9.454545
## 4                 Data Analyst           72       8.666667
## 5 Digital Marketing Specialist           70       8.228571
## 6                      Manager           50       9.340000
## 7               SEO Specialist           62       9.596774
## 8      Social Media Strategist           56      10.000000
```

Subqueries

Subqueries are nested SQL queries that allow us to pass results from an inner query to the main outer query.

For example, instead of applying the `team = 'Social Marketing` condition in the `WHERE` clause, we can use a subquery to pass a list of Social Marketing employee ids into the outer query for filtering. In this case, we would not need to include `team_df` in the join conditions of our main query:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      job_df
                        LEFT JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                    WHERE
                      job_df.employee_id IN (SELECT employee_id FROM team_df WHERE tea
                    GROUP BY
                      job_title
                    HAVING
                      COUNT(*) > 10
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                      job_title employee_cnt avg_org_tenure
## 1        Community Coordinator           58       9.224138
## 2           Content Strategist           72       8.152778
## 3           Creative Assistant           44       9.454545
## 4                 Data Analyst           72       8.666667
## 5 Digital Marketing Specialist           70       8.228571
## 6                      Manager           50       9.340000
## 7               SEO Specialist           62       9.596774
## 8      Social Media Strategist           56      10.000000
```

Derived Tables

An alternative to a subquery is creating a **derived table** in the `FROM` clause using a SQL statement. When using an `INNER JOIN` to connect `job_df` and tenure_df to the derived table containing Social Marketing employee ids, any records in job_df or tenure_df that do not relate to Social Marketing employees will be dropped. With this approach, we would not need a WHERE clause:

```
# Store SQL query as a character string
sql_string <- paste("SELECT
                      job_title,
                      COUNT(*) AS employee_cnt,
                      AVG(org_tenure) AS avg_org_tenure
                    FROM
                      job_df
                        INNER JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                        INNER JOIN
                      (SELECT employee_id FROM team_df WHERE team = 'Social Marketing') social_mk
                        ON
                      job_df.employee_id = social_mktg_ids.employee_id
                    GROUP BY
                      job_title
                    HAVING
                      COUNT(*) > 10
                    ORDER BY
                      job_title")

# Execute SQL query
sqldf(sql_string)
```

```
##                      job_title employee_cnt avg_org_tenure
## 1        Community Coordinator           58       9.224138
## 2           Content Strategist           72       8.152778
## 3           Creative Assistant           44       9.454545
## 4                 Data Analyst           72       8.666667
## 5 Digital Marketing Specialist           70       8.228571
## 6                      Manager           50       9.340000
## 7               SEO Specialist           62       9.596774
## 8      Social Media Strategist           56      10.000000
```

Window Functions

Window functions are used for performing calculations over a set of rows without collapsing the records. Unlike the aggregate functions we've covered, window functions do not collapse rows into a single value; the calculated value is returned for each of the rows over which the calculation was performed.

For example, we can assign an organization tenure rank by Social Marketing job using the `RANK()` and `OVER()` functions in the `SELECT` clause. The `PARTITION BY` argument functions like a `GROUP BY` clause but without collapsing rows, while the `ORDER BY` argument sorts the records in ascending (`ASC`) or descending (`DESC`) order for proper ranking:

```
# Store SQL query as a character string
# Limit output to 10 records since query does not collapse records
# Since employee_id is a column in multiple tables used in this query, we need to indi
sql_string <- paste("SELECT
                      job_df.employee_id,
                      job_title,
                      org_tenure,
                      RANK () OVER (PARTITION BY job_title ORDER BY org_tenure DESC) AS
                    FROM
                      job_df
                        INNER JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                        INNER JOIN
                      (SELECT employee_id FROM team_df WHERE team = ’Social Marketing’)
                        ON
                      job_df.employee_id = social_mktg_ids.employee_id
                    ORDER BY
                      job_title,
                      org_tenure_rank
                    LIMIT 10")

# Execute SQL query
sqldf(sql_string)
```

```
##    employee_id             job_title org_tenure org_tenure_rank
## 1         1236 Community Coordinator         20               1
## 2         1322 Community Coordinator         20               1
## 3         1264 Community Coordinator         19               3
## 4         1348 Community Coordinator         19               3
## 5         3759 Community Coordinator         19               3
## 6         3694 Community Coordinator         18               6
## 7         1257 Community Coordinator         17               7
## 8         3702 Community Coordinator         17               7
## 9         1344 Community Coordinator         16               9
## 10        3659 Community Coordinator         16               9
```

Notice that in the case of organization tenure ties, the `RANK()` function assigns the same rank and then adds the number of ties to that rank to determine the rank for the next highest tenure value.

We can also treat this query as a derived table, and then filter on the derived `org_tenure_rank` field to return a list of Social Marketing employees with the highest tenure for each job. The `*` symbol in the `SELECT` clause will return all columns from the specified table:

```
# Store SQL query as a character string
sql_string <- paste("SELECT tbl.*
                    FROM
                    (SELECT
                      job_df.employee_id,
                      job_title,
                      org_tenure,
                      RANK () OVER (PARTITION BY job_title ORDER BY org_tenure DESC) AS org_tenur
                    FROM
                      job_df
                        INNER JOIN
                      tenure_df
                        ON
                      job_df.employee_id = tenure_df.employee_id
                        INNER JOIN
                      (SELECT employee_id FROM team_df WHERE team = 'Social Marketing') social_mk
                        ON
                      job_df.employee_id = social_mktg_ids.employee_id
                    ORDER BY
                      job_title,
                      org_tenure_rank) tbl
                    WHERE
                      tbl.org_tenure_rank = 1")

# Execute SQL query
sqldf(sql_string)
```

```
##    employee_id                    job_title org_tenure org_tenure_rank
## 1         1236        Community Coordinator         20               1
## 2         1322        Community Coordinator         20               1
## 3         1209           Content Strategist         20               1
## 4         3786           Creative Assistant         20               1
## 5         3808           Creative Assistant         20               1
## 6         1200                 Data Analyst         19               1
## 7         1245                 Data Analyst         19               1
## 8         1400                 Data Analyst         19               1
## 9         3826                 Data Analyst         19               1
## 10        3641 Digital Marketing Specialist         20               1
## 11        3778 Digital Marketing Specialist         20                1
## 12        3475                     Director         15                1
## 13        1157                      Manager         20                1
## 14        1161                      Manager         20                1
## 15        3578                      Manager         20                1
## 16        3599                      Manager         20                1
## 17        1325               SEO Specialist         20                1
## 18        3750               SEO Specialist         20                1
## 19        3845               SEO Specialist         20                1
## 20        1368      Social Media Strategist         20                1
## 21        1013               Vice President         10                1
```

Data Screening & Cleaning

The initial data review process is sometimes referred to as **exporatory data analysis (EDA)**. EDA involves investigating patterns, completeness, anomalies, and assumptions using summary statistics and graphical representations.

A handy function in base R for initial data screening is `summary()`:

```
# Summarize df
summary(demographics)
```

```
##   employee_id     manager_id     hire_date         current_job_start_date
##  Min.   :1001   Min.   :1001   Length:4875        Length:4875
##  1st Qu.:2220   1st Qu.:1973   Class  :character  Class  :character
##  Median :3438   Median :3188   Mode   :character  Mode   :character
##  Mean   :3438   Mean   :3317
##  3rd Qu.:4656   3rd Qu.:4410
##  Max.   :5875   Max.   :5625
##                 NA's   :1
##  current_manager_start_date last_promotion_date  job_title
##  Length:4875                Length:4875         Length:4875
##  Class  :character          Class  :character   Class  :character
##  Mode   :character          Mode   :character   Mode   :character
##
##
##
##
##  business_unit        division             team             location
##  Length:4875          Length:4875          Length:4875          Length:4875
##  Class  :character    Class  :character    Class  :character    Class  :character
##  Mode   :character    Mode   :character    Mode   :character    Mode   :character
##
```

```
##
##
##
##     work_zip        home_zip      annual_comp       job_level
##  Min.   :30301   Min.   :30301   Min.   : 50016   Min.   : 1.00
##  1st Qu.:30318   1st Qu.:30318   1st Qu.: 87946   1st Qu.: 4.00
##  Median :63143   Median :63139   Median :125522   Median : 8.00
##  Mean   :56693   Mean   :56693   Mean   :125369   Mean   : 7.02
##  3rd Qu.:78722   3rd Qu.:78722   3rd Qu.:163262   3rd Qu.:10.00
##  Max.   :78759   Max.   :78759   Max.   :199968   Max.   :10.00
##  NA’s   :1311    NA’s   :1311
##       age          org_tenure       job_tenure        gender
##  Min.   :21.00   Min.   : 1.000   Min.   : 0.000   Length:4875
##  1st Qu.:39.00   1st Qu.: 5.000   1st Qu.: 2.000   Class :character
##  Median :50.00   Median : 8.000   Median : 3.000   Mode  :character
##  Mean   :48.51   Mean   : 9.193   Mean   : 4.154
##  3rd Qu.:59.00   3rd Qu.:13.000   3rd Qu.: 5.000
##  Max.   :70.00   Max.   :20.000   Max.   :20.000
##
##   ethnicity           education     performance
##  Length:4875        Min.   :1.00   Min.   :1.000
##  Class :character   1st Qu.:2.00   1st Qu.:1.000
##  Mode  :character   Median :3.00   Median :3.000
##                     Mean   :2.88   Mean   :2.495
##                     3rd Qu.:4.00   3rd Qu.:3.000
##                     Max.   :4.00   Max.   :4.000
##
```

Note that fields with `NA's` contain missing values. Also, by default `employee_id` and `manager_id` are both treated as integers in R, which is why descriptive statistics appropriate for numeric data are provided. `employee_id` and `manager_id` should actually be treated as character strings since we will not perform any arithmetic operations using these ids.

Missingness

Before considering whether and how to handle missing data, it is important to distinguish between *structural missingness* and *informative missingness* (Kuhn & Johnson, 2013).

Structural missingness relates to data that is missing for a logical reason. For example, we would not expect a new joiner with a few days of tenure to have a performance score. Likewise, we would not expect an active employee who is not a rehire to have a termination date. Therefore, we would not want to address missing values in these cases.

Informative missingness relates to missing data that is informative regarding an outcome of interest. For example, in a survey context we may find a relationship between missing values on manager effectiveness questions and unfavorability on a psychological safety scale. This may indicate that employees who are fearful of retaliation are uncomfortable providing honest feedback about their managers, while employees who feel it is safe to speak up about issues are more comfortable responding in prosocial ways.

In some cases, we have the luxury of simply removing observations with missing values and using the remaining complete cases for analysis. However, since we are often working with wide datasets containing relatively few observations in people analytics, this may not be feasible. As we will cover in later chapters, sample size considerations are fundamental to achieving adequate power in statistical testing, so case removal is only possible with larger datasets.

Data imputation refers to the methods by which missing data are replaced with substituted values when case removal is not appropriate. The most common data imputation method is replacing missing values with a descriptive statistic such as the mean or mode based on available data. For example, if most employees have an age in the system, the average, median, or most frequent age could be used in place of the cases with a missing age. To be more precise, the average, median, or most frequent age of those with *similar characteristics* may be used (e.g., similar years of experience, job, level). We would expect there to be less variability in age within a well-defined segment relative to the entire employee population, so this would likely be a more accurate substitute for an individual’s actual age.

Let’s evaluate the `demographics` data frame for missing `annual_comp` values using the logical `is.na()` function, and return values of variables relevant in determining one’s annual compensation:

```
# Force a NA in lieu of annual comp for illustrative purposes
demographics[demographics$employee_id == ’4305’, ’annual_comp’] <- NA

# Return relevant employee characteristics where annual comp is missing
demographics[is.na(demographics$annual_comp), c("employee_id", "job_title", "location"
```

```
##      employee_id job_title      location job_level
## 3305        4305   Manager HQ 1 - Austin         7
```

Next, we can impute the average value of `annual_comp` based on employees with the same values for the relevant variables:

```
# Return average annual comp for employees with similar characteristics, excluding emp
imputed_comp <- mean(demographics[demographics$job_title == ’Manager’ & demographics$lo

# Impute missing comp for relevant segment
demographics[demographics$employee_id == ’4305’, ’annual_comp’] <- imputed_comp
```

```
# Display imputed comp
demographics[demographics$employee_id == '4305', 'annual_comp']
```

```
## [1] 115338.2
```

While this approach should help in demonstrating the mechanics of imputing a missing value on a case-by-case basis, a more scalable solution is needed for data with a large number of missing values across employees with different values of these variables. There are more sophisticated methods of data imputation that involve models to estimate missing values. *Linear regression* and *K-Nearest Neighbor (KNN)* models are commonly used for this. These modeling techniques leverage a similar approach to the method outlined above in that the target values of cases with similar characteristics to those with missing values are used to aid estimation. These models will be covered in detail in later chapters.

Outliers

The treatment of outliers is a controversial topic. Appropriate methods for defining and addressing outliers are domain-specific, and there are many important considerations that should inform whether and how outliers should be treated. As discussed in Chapter @ref(uni-bi-stats), a common method of outlier detection is identifying values which fall outside the following interval:

$$I = Q1 - 1.5 * IQR; Q3 + 1.5 * IQR$$

Low Variability

Variables with **low variability** often do not provide sufficient information for identifying patterns in data. For example, if we are interested in using information on stock options to understand why employees vary in their levels of retention risk, but find that the employee stock purchase plan (ESPP) terms are identical for nearly all employees, including a stock option variable in the analysis is unlikely to provide any meaningful signal.

When working with survey data, checking for **straightlining** should be an early data screening step. Straightlining refers to a constant response across all survey items, which may be evidence that the respondent lost motivation or was not attentive and thoughtful when taking the survey. Since straight-line responses may influence results, it is often best to discard these cases – especially when the sample size is adequately large for the planned analyses without them. If the same response is given for both positively and negatively worded versions of a question (e.g., comparing “I plan to be here in a year” to “I do not plan to be here in a year”), which we expect to be inversely related, this gives added support for discarding these responses.

Fields with low variability can be easily identified using descriptive statistics from the `summary()` function. If the `Min` and `Max` values are equal, there is no variability in the field's values.

Inconsistent Categories

Inconsistent categories impact aggregation and trending by categorical dimensions. It is often necessary to create mappings based on logical rules in order to standardize dimension values across time. In the case of reorgs, a department may be disbanded, renamed, or integrated into one or multiple other departments. Therefore, when working with historical data, records may contain legacy department names that do not align with the current organizational taxonomy. Mapping from former to current departments may require logic based on manager ids, divisions, job profiles, locations, or other variables depending on the nature of reorgs over time.

Job architecture projects often introduce the need for mappings as well. Jobs and levels may completely change for all employees with a job architecture revamp, in which case trending along job and level dimensions (e.g., attrition by job or level over multiple years) is only possible with logic that clarifies how legacy jobs and levels map to those in the new career framework.

Changes to allowable values in source systems often result in inconsistent categorical data over time. For example, the education field may switch from a free-form text field in which employees can enter any value (e.g., B.S., B.A., BS, BA, Bachelor of Science, Bachelor of Arts, Bachelor's, Bachelors, Bachelor's Degree, Bachelor Degree, undergraduate degree, 4-year degree, four-year degree) to a standardized solution in which there is a set of allowable values from which employees can choose (e.g., Bachelor's Degree, Master's Degree, Doctoral Degree). This will either warrant a one-time historical cleanup upon implementing the allowable values or downstream logic to tidy up data for analytics. A best practice is to address data quality issues upstream to avoid duplicative data cleaning procedures across downstream applications.

Data Binning

Data binning refers to the process by which larger high-level groups of values are defined and constructed. As a general rule, very granular categories should be avoided – especially when there is no theoretical basis for such categories facilitating a project's objectives or deepening insights. Where the *n*-count is expected to be consistently low for a defined categorical bin, it is usually best to define a larger bin. A variable measuring highest level of educational attainment that contains 9th, 10th, 11th, and 12th grade categories may be converted into higher-level "High School Not Completed" and "High School Completed" bins.

For modeling applications, it is important to let the algorithm determine the cutpoints for numeric data in relation to the outcome. For example, if organization tenure is measured in years, arbitrarily defining bin sizes of 'Less Than 1 Year', '1-5 Years', and 'More Than 5 Years' will likely result in information loss. Any variability *within* these bins that may be useful in explaining variance in the outcome would be lost with such wide bins. In fact, machine learning (ML) models such as decision trees, which will be covered in later chapters, are great for algorithmically determining cut points for binning numeric data across descriptive, diagnostic, and predictive projects alike.

One-Hot Encoding

One-hot encoding, also known as **dummy coding**, involves transforming a categorical variable into numeric values on which statistical procedures can be performed. For EDA, dummy coding is not required, as metrics such as counts and percentages can be calculated on these dimensions for descriptive purposes. However, for statistical and ML modeling applications categorical variables must be converted into $k - 1$ variables, where k is the number of categories, using binary (1/0) coding.

Understanding how categorical data are coded is critical to a correct interpretation of output. For example, if a remote work variable exists with "Remote" or "Non-Remote" values, we could code "Remote" values as 1 and "Non-Remote" values as 0. We could then evaluate the statistical relationship of this transformed categorical variable with other numeric variables.

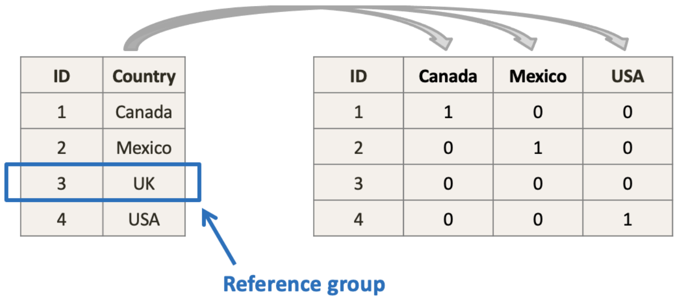
If a categorical variable has more than 2 values, we must create a separate 1/0 field for each value and omit one category for use as a reference group. As we will cover in Chapter @ref(lm), one of several assumptions in linear regression is that independent variables are not collinear; that is no pair of independent variables is highly correlated. Without an omitted category, each of the one-hot encoded fields will be perfectly correlated with the others; when the field representing category A is 1, the fields for other categories will always be 0. As illustrated in Figure @ref(fig:onehot-encoding), by omitting a category there will be cases when all fields have a 0 value (i.e., rows where the value is the omitted category), which will reduce the strength of the bivariate correlations.

For a categorical variable with only two values, the `ifelse()` function can be leveraged to assign values:

```r
# Return unique values of gender field
unique(demographics$gender)
```

```
## [1] "Male"   "Female"
```

```r
# Gender one-hot encoding
demographics$gender_ohe <- ifelse(demographics$gender == 'Female', 1, 0)

# Preview records
head(demographics[, c("employee_id", "gender_ohe")])
```



ID	Country
1	Canada
2	Mexico
3	UK
4	USA

ID	Canada	Mexico	USA
1	1	0	0
2	0	1	0
3	0	0	0
4	0	0	1

Figure 9.6: One-Hot Encoding

```
##   employee_id gender_ohe
## 1        1001          0
## 2        1002          1
## 3        1003          0
## 4        1004          0
## 5        1005          1
## 6        1006          1
```

For variables with more than 2 categories, we can leverage the `model.matrix()` function to one-hot encode them all:

```
# Load library
library(dplyr)

# Determine location with lowest n-count
locations <- demographics %>% count(location, sort = TRUE)

# Display counts by location
locations
```

```
##                          location    n
## 1                 HQ 1 - Austin 1733
## 2                HQ 2 - Atlanta 1522
## 3                        Remote  969
## 4 Operations Center - St. Louis  651
```

```
# Store location with smallest n-count
min_location <- locations[nrow(locations), "location"]
```

```
# Location one-hot encoding
# Omit location with lowest n-count
location_ohe <- model.matrix(~location-1, data = demographics[!demographics$location == min_locat
head(location_ohe)
```

```
##   locationHQ 1 - Austin locationHQ 2 - Atlanta locationRemote
## 1                     1                      0              0
## 2                     1                      0              0
## 3                     1                      0              0
## 4                     1                      0              0
## 5                     1                      0              0
## 6                     1                      0              0
```

Feature Engineering

Level one people analytics tends to utilize only the delivered fields from the HRIS (e.g., location, job profile, org tenure), but a good next step is to derive smarter variables from these fields. These can then be used to cut data differently or as inputs in models. Below are some examples of how basic data available in the HRIS can be transformed into new variables that provide different information:

- Number of jobs per unit of tenure (larger proportions tend to see greater career pathing)
- Office/remote worker (binary variable dummy coded as 1/0)
- Local/remote manager (binary variable dummy coded as 1/0)
- Hire/Rehire (binary variable dummy coded as 1/0)
- Hired/acquired (proxy for culture shock effects)
- Gender isolation (ratio of employee's gender to number of the same within immediate work group)
- Generation isolation (comparison of age bracket to most frequent generational bracket within immediate work group)
- Ethnic isolation (ratio of employee's ethnicity to number of the same within immediate work group)
- Difference between employee and manager age
- Percentage change between last two performance appraisal scores (per competency and/or overall)
- Team and department quit outbreak indicators (ratio of terms over x months relative to average headcount over x months)
- Industry experience (binary or length in years)

Exercises

1. What are the differences between data lakes, data warehouses, and data marts?
2. What is the difference between a Type 1 and Type 2 table in a DW?
3. What two clauses must always be present in a SQL query?
4. How do aggregate functions differ from window functions in SQL?
5. What is a subquery?
6. What is the difference between an INNER JOIN and LEFT JOIN?
7. Why is it dangeous to address missing values without domain knowledge of how the data are generated?
8. How can missing values be addressed when impacted records cannot be eliminated from a data set?
9. When is one-hot encoding required for categorical variables?
10. Why should variables with low to no variability be dropped?

Chapter 10

Analysis of Differences

Chapter 11

Linear Regression

It's important to draw a distinction between inferential and predictive models. Inferential models are highly interpretable and their utility is largely in understanding the nature and magnitude of the effect variables have on outcomes. Inferential models also lend to quantifying the extent to which we can generalize the observed effects to the larger population from which the sample was drawn. The objective in predictive modeling is to also to learn from patterns in historical data but for the purpose of achieving the most accurate predictions of future events – even at the expense of interpretability. To be clear, this isn't to say that predictive models cannot be interpreted – they certainly can – but I've seen relatively few applications for predictive modeling in people analytics because models generally need to be highly interpretable to support action planning.

This chapter is dedicated to inferential models to support a working understanding of how to interpret model output and communicate clear, data-driven narratives that respect the nuance and noise characteristic of people data. The following chapter will provide an overview of predictive modeling frameworks.

Regression is perhaps the most important statistical learning technique for people analytics. If you have taken a statistics course at the undergraduate or graduate levels, you have surely already encountered it. Before diving into the math to understand the mechanics of regression, let's develop an intuitive understanding.

Imagine we are sitting at a large public park in NYC on a nice fall afternoon. If asked to estimate the annual compensation of the next person to walk by, in the absence of any additional information how would you estimate this? Most would likely estimate the average annual compensation of everyone capable of walking by. Since this would include both residents and visitors, this would be a very large group of people! The obvious limitation with this approach is that among the large group of people capable of walking by, there is likely a significant range of annual compensation values. Many walking by may be children, unemployed, or retirees who earn no annual compensation, while others may be highly compensated senior executives at the pinnacle of their careers. Since the range of annual compensation could be zero to billions of dollars, estimating the average of such a large population is likely going to be highly inaccurate without more information about who may walk by.

Let's consider that we are sitting outside on a weekday afternoon. Should this influence our annual compensation estimate? It is likely that we can eliminate a large segment of those likely to walk by, as we would expect most children to be in school on a typical fall weekday afternoon. It's also unlikely that those who are employed and not on vacation will walk by on a fall weekday afternoon. Therefore, factoring in that it is a weekday should limit the size of the population which in turn may reduce the range of annual compensation values for our population of passerbys.

Let's now consider that the park is open only to invited guests for a symposium on people analytics. Though it may be difficult to believe, a relatively small subset of the population is likely interested in attending such a symposium, so this information will likely be very helpful in reducing the size of the population who could walk by, which should further reduce the range of annual compensation since we probably have a good idea of the profile of those most likely to attend. This probably also lessens (or altogether eliminates) the importance of the weekday factor in explaining why people vary in the amount of compensation they earn each year.

Furthermore, let's consider that only those who reside in NYC and Boise were invited, and that the next person to walk by resides in Boise. Most companies apply a significant cost of living multiplier to the compensation for those in an expensive region such as NYC, resulting in a significant difference in compensation relative to those residing in a much less expensive city like Boise – all else being equal. Therefore, if we can partition attendees into two groups based on their geography, this should limit the range of annual compensation significantly within each – likely making the average compensation amount in each group a more nuanced and reasonable estimate.

What if we also learn the specific zip code in which the next passerby from Boise resides? The important information is likely captured in the larger city label (NYC vs. Boise), and the compensation for the specific zip codes within each city are unlikely to vary to a significant degree. Assuming this is true, it probably would not make sense to consider both the city name and zip code since they are effectively redundant pieces of information with regard to explaining differences in annual compensation.

What if we learn that the next person to walk by will be wearing a blue shirt? Does this influence your estimate? Unless there is research to suggest shirt color and earnings are related, this information will likely not contribute any significant information to our understanding of why people vary in the amount of compensation they earn annually and should, therefore, not be considered.

You can probably think of many relevant variables that would help further narrow the range of annual compensation. These may include job, level, years of experience, education, location, among other factors. The main thing to understand is that for each group of observations with the same characteristics – such as senior analysts with a graduate degree who reside in NYC – there is a distribution of annual compensation. This distribution reflects unexplained variance. That is, we do not have information to explain why the compensation for each and every person is not the same and in social science contexts, it simply is not practical to explain 100 percent of the variance in outcomes. Two people may be similar on hundreds of factors (experience, education, skills) but one was simply a more effective negotiator when offered the same role and commanded a higher salary. It's likely we do not have data on salary negotiation ability so this information would leave us with unexplained variance in compensation. The goal is simply to identify the variables that provide the most information in helping us tighten the distribution so that estimating the average value will generally be an accurate estimate for those in the larger population with the same characteristics.

While we can generally improve our estimates with more relevant information (not shirt color or residential zip code in this case), it is important to understand that samples which are too small ($n < 30$) lend to anomalies; modeling noise in sparse data can result in models that are unlikely to generalize beyond the sample data. For example, if the only people from Boise to attend the people analytics symposium happen to be two ultra wealthy tech entrepreneurs who earn millions each year, it would not be appropriate to use this as the basis for our estimates of all future attendees from Boise. This is a phenomenon known as overfitting that will be covered later in this chapter.

This is the essence of regression modeling: find a limited number of variables which independently or jointly provide significant information that helps explain (by reducing) variance around the average value. As illustrated in this example, adding additional variables (information) can impact the importance of other variables or may offer no incremental information at all. In the subsequent sections, we will cover how to identify which variables are important and how to quantify the effect they have on the outcome.

Simple Linear Regression

Parameter Estimation

Ordinary Least Squares (OLS) is the most common method for estimating unknown parameters in a linear regression model.

Multiple Linear Regression

Moderation

Mediation

Polynomial Regression

Hierarchical Models

Exercises

Chapter 12

Generalized Linear Regression

Logistic Regression

Logistic regression is an excellent tool when the outcome is categorical. Logistic regression allows us to model the probability of different classes – a type of modeling often referred to as classification. The context for classification can be binomial for two classes (e.g., active/inactive, promoted/not promoted), multinomial for multiple unordered classes (e.g., skills, job families), or ordinal for multiple ordered classes (e.g., survey items measured on a Likert scale, performance level).

Binomial Logistic Regression

Multinomial Logistic Regression

Ordinal Logistic Regression

Proportional Odds Logistic Regression

Poisson Regression

Exercises

Chapter 13

Survival Analysis

Exercises

Chapter 14

Predictive Models

Chapter 15

Unsupervised Learning

Factor Analysis

Clustering

Exercises

Chapter 16

Network Analysis

Centrality Measures

Degree Centrality

Eigenvector Centrality

Betweenness Centrality

Closeness Centrality

Exercises

Chapter 17

Data Visualization

Design Guidelines

Visualization Types

Tables

Heatmaps

Scatterplots

Line Graphs

Single Series

Two Series

Multiple Series

Slopegraphs

Bar Charts

Vertical

Horizontal

Stacked

Waterfall Charts

Area Charts

Pie Charts

Exercises

Chapter 18

Data Storytelling

Know Your Audience

TL;DR

Telling the Story

Reference Material

Exercises

Chapter 19

Bibliography

Albright, S. C., & Winston, W. L. (2016). Business Analytics: Data Analysis & Decision Making (6th ed.). Boston, MA: Cengage.

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. Journal of Personality and Social Psychology, 51(6), 1173–1182.

Bartlett, R. (2013). A Practitioner's Guide to Business Analytics: Using Data Analysis Tools to Improve Your Organization's Decision Making and Strategy. New York, NY: McGraw-Hill.

Creswell, J. W., & Creswell, J. D. (2018). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (5th ed.). Los Angeles: Sage.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. Psychological Bulletin, 52(4), 281–302. https://doi.org/10.1037/h0040957

Daw, R. H., & Pearson, E. S. (1972). Studies in the History of Probability and Statistics. XXX. Abraham De Moivre's 1733 Derivation of the Normal Curve: A Bibliographical Note. Biometrika, 59(3), 677–680. https://doi.org/10.2307/2334818

DeVellis, R. F. (2012). Scale development: Theory and applications. Thousand Oaks, Calif: Sage.

Edmondson, A. (1999). Psychological safety and learning behavior in work teams. Administrative Science Quarterly, 44(2), 350-383.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: With Applications in R. New York: Springer.

Kahneman, D. (2011). Thinking, fast and slow. New York: Farrar, Straus and Giroux.

Kerlinger, F., & Lee, H. (2000). Foundations of behavioral research (4th ed.). Melbourne: Wadsworth.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. New York: Springer.

Perneger, T. V. (1998). What's wrong with Bonferroni adjustments. BMJ, 316(7139), 1236-1238.

Pishro-Nik, H. (2014). Introduction to Probability: Statistics and Random Processes. Blue Bell, PA: Kappa Research, LLC.

Roethlisberger, F. J., & Dickson, W. J. (1939). Management and the worker. Harvard Univ. Press.

Wheelan, C. (2013). Naked Statistics: Stripping the Dread from the Data. New York: W.W. Norton.

Chapter 20

Appendix

Exercise Solutions

Intro to R

1. What is the difference between a factor and character vector?

Factors take on a limited number of predefined categorical values, while character vectors may contain any number of values (including numbers and dates treated as characters).

2. What is vectorization?

Vectorization performs a given operation on each and every element of an object. This makes writing code more efficient and concise.

3. How do data frames differ from matrices?

While both data frames and matrices are retangular objects (values organized within rows and columns), data frames can hold multiple types of data (e.g., integers, characters, dates, logical) while matrices are limited to single data type.

4. Executing the `Sum()` function will achieve the same result as `sum()`. B. False
5. Executing `seq(1,10)` returns the same results as `1:10`. A. True

6. The following command is sufficient for creating a vector recognized by R as having three dates: `dates <- c("2022-01-01", "2022-01-02", "2022-01-03")`. B. False

7. How are `while` and `for` loops different?

The main difference between a `while` and `for` loop is that a `for` loop is used when the number of iterations is known, compared to a `while` loop where the code runs until a given condition is satisified.

8. If vectors `x1` and `x2` each hold 100 integers, how can we add each element of one to the respective element of the other using a single line of code?

Thanks to vectorization, this is as simple as executing the following code: `x1 + x2`.

9. List elements can be referenced by index using double brackets (e.g., `my_lst[[5]]` for the 5th element of my_lst). A. True

10. What are some examples in which a user-defined function (UDF) is needed?

The use cases for UDFs are many. In analytics contexts, domain-specific algorithms often need to be developed, and UDFs provide the flexibility to implement the logic at scale.

Measurement & Sampling

1. Parameters are descriptions or characteristics of a sample, while statistics are descriptions or characteristics of a population. B. False

2. How does a sampling frame differ from a sample?

   A sampling frame represents the subset of the population to which the researcher has access, while the sample is the subset of the sampling frame that ultimately participates in the research.

3. How does cluster sampling differ from stratified random sampling?

   It is the *clusters* that are selected at random with cluster sampling rather than the *individuals* with stratified random sampling.

4. What is the primary benefit of probabilistic sampling methods over non-probabilistic sampling?

   Probability sampling allows us to make inferences about a larger population based on sample data, while this is not the case for non-probability sampling methods since the sample is unlikely to be representative of the population.

5. Nonresponse bias is only applicable in the context of surveys. B. False

6. Which of the following variable types influences the strength of the effect one variable has on another? C. Moderating Variable

7. 100 randomly selected employees in the Marketing department of an organization participated in a survey on career pathing for marketing professionals. What is the sample and what is the population sampled in this case? B. Sample: 100 employees who completed the survey, Population: Marketing employees

8. Zero represents the absence of something for inveral scaled variables, whereas this is not the case for ratio scaled variables. B. False

9. Discrete variables can have more than 2 values. A. True

10. Which of the following are NOT probabilistic sampling methods? B. Quota sampling D. Purposive sampling

Research Fundamentals

1. What type of research method and design would be best suited for a study aiming to understand the effect of stay interviews on employee attrition?

A quantitative research method and quasi-experimental design would be best suited for this study. While experimental designs are more rigorous than quasi-experiments, stay interviews are unlikely to be initiated via random processes; leaders will likely identify select people as having a higher risk of attrition and initiate stay interviews accordingly.

2. Why are quasi-experiments less rigorous than true experiments?

Randomized experimental designs provide the most rigor with regard to causal validity, but random assignment is not performed in a quasi-experimental context.

3. Why aren't experimental designs always implemented when an experimenter-manipulated IV is warranted?

Randomized experimental designs provide the most rigor with regard to causal validity. However, in social science research contexts, true experiments often are not possible due to ethical considerations.

4. What is the role of research questions?

Research questions help focus the study, determine the appropriate methodology, and guide each stage of inquiry, analysis, and reporting.

5. What is the role of research hypotheses?

Research hypotheses are testable statements about the expected outcome of a research project or experiment.

6. What is the difference between internal and external validity, and why are these concepts important in research?

Internal validity reflects the robustness of the study – the extent to which confounding variables are controlled. External validity refers to the extent to which study conclusions will hold in other contexts (for other people, in other places, at other times). Confidence in conclusions about causal effects or statistical relationships, and the likelihood of such conclusions to hold true in other contexts, is rooted in a study's internal and external validity.

7. What is an example of a mixed methods study?

One example of a mixed methods study is a two-part study in which part 1 involves employee focus groups to surface themes that are top-of-mind for employees, and part 2 involves a survey to measure constructs related to the emergent themes in order to test statistical relationships with outcomes like engagement and retention.

8. What is the key difference between experimental and non-experimental research designs?

Unlike experimental designs, non-experiments do not involve the manipulation of an IV. The goal of non-experiments is not to provide evidence for causal effects, but to study measured variables as they naturally occur.

9. What are the differences between cross-sectional, correlational, and observational non-experimental designs?

Cross-sectional research compares two or more natural groups of people. Correlational research involves studying the statistical relationship between two variables. Observational research refers to studies in which the researcher gathers information without research subjects being explicitly involved in the recording of data.

10. How can the Hawthorne Effect impact the integrity of an experiment?

Observed differences in a DV may not be attributable to the manipulation of an IV but merely due to research subjects having knowledge that they are being observed and temporarily modifying behavior.

Univariate & Bivariate Analysis

1. Which of the following measures of central tendency is least sensitive to extreme values (outliers)? A. Median
2. The standard deviation represents the 'average' amount by which x values deviate (or vary) from the mean. A large standard deviation indicates there is considerable spread in the data, whereas a small standard deviation indicates the mean is fairly representative of the data. A. True
3. A positively skewed distribution has its largest allocation to the left and a negative distribution to the right. B. False
4. Large covariance coefficients always indicate strong bivariate associations. B. False
5. Which of the following can be found in boxplots? A. Quartiles B. Median D. IQR E. Outliers
6. The 3rd quartile (Q3) is equivalent to the 75th percentile. A. True
7. Which of the following correlation coefficients can be used when evaluating the relationship between a pair of rank-ordered variables? B. Spearman's Rank E. Kendall's Rank
8. Which of the following correlation coefficients can be used when evaluating the relationship between a pair of dichotomous variables? C. Phi
9. Platykurtic distributions are flat relative to mesokurtic distributions. A. True
10. When using the Pearson method, values down the diagonal of a covariance matrix represent the variance for each variable. A. True

Inferential Statistics

1. Which of the following is an example of a null hypothesis, where μ reflects the mean of a population? A. $\mu_A = \mu_B$
2. Which of the following describes a Type I Error? E. Both B and C
3. The primary purpose of inferential statistics is to make inferences about a population based on sample data. Inferential statistics allows these inferences to be made with defined levels of confidence that what is observed in a sample is also characteristic of the larger population. A. True

4. A T-Test should be used when σ is unknown and/or $n < 30$. A. True

5. Randomness is essential to probabilistic methods. A. True

6. Which of the following is characteristic of the Bonferroni Correction? A. Reducing the risk of a Type I Error B. Increasing the risk of a Type II Error C. Reducing the familywise error rate

7. Tolerance for a wider interval is an important tradeoff decision when increasing the level of confidence that a range of values contains an unknown population parameter. A. True

8. A CI represents the range of values we expect to include an unknown population parameter (often the mean) for a specified degree of confidence. A. True

9. When population parameters are unknown, which of the following tests is most appropriate for testing $\mu_A = \mu_B$? B. T-Test

10. According to the Empirical Rule, 95% of normally distributed data lie within how many standard deviations of the mean? B. 2

** Data Wrangling and Preparation**

1. What are the differences between data lakes, data warehouses, and data marts?

2. What is the difference between a Type 1 and Type 2 table in a DW?

3. What two clauses must always be present in a SQL query?

4. How do aggregate functions differ from window functions in SQL?

5. What is a subquery?

6. What is the difference between an INNER JOIN and LEFT JOIN?

7. Why is it dangeous to address missing values without domain knowledge of how the data are generated?

8. How can missing values be addressed when impacted records cannot be eliminated from a data set?

9. When is one-hot encoding required for categorical variables?

10. Why should variables with low to no variability be dropped?

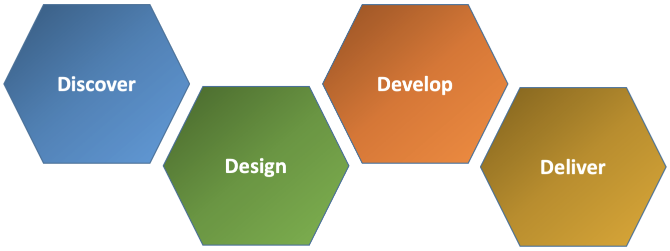


Figure 20.1: Figure 1: 4D Framework

4D Framework

1. Discover

You are likely familiar with the old adage: "An ounce of prevention is worth a pound of cure." Such is the case with respect to planning in an analytics context. During the Discover phase, it is important to remain in the problem zone; seek to understand your clients' needs through active listening and questions. This is not the time for solutioning or committing to any specific deliverables. If the client's needs are ambiguous, proceeding will likely be an exercise in futility. Outlined below is a set of general questions that should be considered during this initial phase to prevent allocating scarce time and resourcing to a project that ultimately misses the mark.

- Client

  Who is the client? A client can be a person or organization that has contracting you for consulting services, or an internal stakeholder within your organization who has need. What is important to them?

- Primary Objective

  - What is the client ultimately hoping to accomplish?
  - Is the request merely to satisfy one's curiosity, or are there actions that can realistically be taken to materially influence said objective?

- Problem Statement

  - One of my most important early steps is clearly defining the problem statement. If your understanding of the problem – after translating from the business terms in which it was initially expressed – is misaligned with the client's needs, none of the subsequent steps matter.

- Guiding Theories

  - What theoretical explanations can the client offer as potential rationalizations for the phenomena of interest?
  - Are there existing theories in the organizational literature that should guide how the problem is tackled (e.g., findings from similar research implemented in other contexts)?

- Research Questions

  To respect the nuances of the problem statement, it is important to unpack it and frame as a set of overarching questions to guide the research.

  - Q1: ...
  - Q2: ...
  - Q3: ...

- Research Hypotheses

  Once research questions are developed, what do you expect to find based on anecdotal stories or empirical findings? As a next step, these expectations should be expressed in the form of research hypotheses. Please note that these research hypotheses are different from statistical hypotheses.

  - H1: ...
  - H2: ...
  - H3: ...

  To ensure the hypotheses lend themselves to actionable analyses, it is important to consider the following: "What does success look like?" In other words, once the project is complete, against which success measures will the project's success be determined? Curiosity is not a business reason and hope is not a reasonable strategy. The following questions may prove helpful in the promotion of actionable – over merely interesting outcomes:

  - What will be done if the hypotheses are empirically supported?
  - What will be done if the hypotheses are not empirically supported?

- Assumptions

  At this point, it's helpful to consider what assumptions may be embedded in this discovery work. Are the questions and hypotheses rooted in what the client has theorized, or are these the product of an ambiguous understanding of the client's needs?

- Cadence

  - Is this analysis a one-off, or could there be a need to refresh this analysis on a regular cadence?
  - Are there dates associated with programs, actions, etc. this analysis is intended to support?

- Aggregation

  Is there a need for individual-level detail supporting the analysis? Aaggregate data should generally be the default unless a compelling justification exists and approval from legal and privacy partners is granted. One important role of analysts is to help keep the audience focused on the bigger picture and findings. Access to individual-level detail can not only introduce unnecessary legal and compliance risk but can also lead to questions and probing that can delay taking needed actions based on the results.

- Deliverable

  What is the preferred method of communicating the results of the analysis (e.g., interactive dashboard, static slide deck, document)? It is important to determine this early so that subsequent efforts can be structured to support the preferred deliverable. For example, if an interactive dashboard is preferred, does your Engineering department need to prioritize dependent tasks such as data feeds, row-level security, BI development, and production server migrations?

- Filters & Dimensions

  How does your client prefer to segment the workforce? Some common grouping dimensions are business unit, division, team, job family, location, tenure, and management level.

2. Design

Perhaps the most important initial question to answer in the design phase is: "Does anything already exist that addresses part, or all, of the client's objectives?" If the existing solution will suffice, it's possible that there is simply a communication/education gap, and you can allocate time and resources elsewhere.

The end-user experience is of paramount importance during the Design phase, as solutions should have a consistent look and feel regardless of who developed the product. To achieve this, it is important to resist siloed thinking and consider the broader set of analytics solutions the team has delivered – or is in the process of delivering.

- Data Privacy

  Are there potential concerns with the study's objective, planned actions, and/or requested data elements from an employee privacy or legal perspective? A cross-functional data governance committee can help with efficient and consistent decisioning on requests for people data and analytics.

- Data Sources & Elements

  - What data sources are required?

  - What data elements are required?

  In cases where sensitive attributes such as gender, ethnicity, age, sexual orientation, and disability status are requested, it's always best to exercise a 'safety first' mentality and consult with legal and privacy partners to ensure there is comfort with the intended use of the data. The decision on whether or not to include these sensitive data elements is often less about what the audience can view (e.g., People Partners may already have access to the information at the person level in the source system) and more anchored in what they plan to do with the information.

  Is the required data already accessible in a data warehouse or other analytics environment? If not, does it need to be? What is required to achieve this?

- Data Quality

  It is important to understand the data generative process and never make assumptions about how anomalies or missing data should be interpreted. After identifying what data sources will be required for a particular analysis, it is important to meet with source system owners and data stewards to deeply understand the business processes by which data are generated in the system(s). Are there data quality concerns that need to be explored and addressed?

- Variables

  How will the constructs be measured (e.g., survey instrument, derived attribute, calculated field)?

- Analysis Method

  What are the appropriate analysis methods based on the research hypotheses? If modeling is required, is it more important to index on accuracy or interpretability?

- Dependencies

  Are other teams required to develop this solution? What is the nature of the work each dependent team will perform? Are there required system configuration changes? Do these teams have capacity to support?

- Change Management

  Will this solution impact current processes or solutions? If so, what is the change management plan to facilitate a seamless transition and user experience?

- Sign-Off

  Generally, it is best for the client to signoff on the problem statement, analysis approach, and wire frame for the deliverable (if applicable) before providing an ETA and proceeding to the development phase. This ensures alignment on the client's needs and the perceived utility of the solution in addressing those needs.

3. Develop

- Development Patterns
  - Are there development patterns that should guide the development approach to support consistency?
  - Are there existing calculated fields that can/should be leveraged for derived data?
  - Are there best practices that should be employed to optimize performance (e.g., load time for dashboards, executing complex queries during non-peak times)?
  - Are there standard color palettes that should be applied?
- Productionalizable Code
  - How do models and data science pipelines need to be developed to facilitate a seamless migration from lower to upper environments? For example, initial exploratory data analysis (EDA) may be performed using curated data in flat files for the purpose of identifying meaningful trends, relationships, and differences, but where will this data need to be sourced in production to automate the refresh of models at a regular interval? If the data were provided from multiple source systems, what joins are required to integrate the data? What transformation logic or business rules need to be applied to reproduce the curated data?
- Unit Testing
  - What test cases will ensure the veracity of data?
  - Who will perform the testing?
- UAT Testing
  - In the spirit of agility and constant contact with the client to prevent surprises, it is generally a good idea to have the client take the solution for a test run within the UAT environment and then provide sign-off before migrating to production. If the deliverable is a deck or doc with results from a model, UAT may surface clarifying questions that can be addressed before releasing to the broader audience.

4. Deliver

The Deliver phase can take many forms depending on the solution being released. If the solution is designed for a large user base, a series of recorded trainings may be in order so that there is a helpful reference for those unable to attend the live sessions or new joiners in the future. It is important to monitor success measures, which could be insights aligned to research hypotheses, dashboard utilization metrics, or any number of others defined within the Discover phase.