Computational Probability and Statistics

Bradley Warner Brianna Hitt Ken Horton

2024-10-29

Table of contents

Preface

This book is based on the notes we created for our students as part of a one semester course on probability and statistics. We developed these notes from three primary resources. The most important is the OpenIntro Introductory Statistics with Randomization and Simulation (ISRS) (Diez, Barr, and Çetinkaya-Rundel 2014) book. In parts, we have used their notes and homework problems. However, in most cases we have altered their work to fit our needs. The second most important book for our work is Introduction to Probability and Statistics Using R (Kerns 2010). Finally, we have used some examples, code, and ideas from the first edition of Prium's book, Foundations and Applications of Statistics: An Introduction Using R (R. J. Pruim 2011).

In a 2024 reorganization of our inference block, we revised our inference case study and added a chapter on sampling distributions. The materials for the case study utilized the OpenIntro ISRS (Diez, Barr, and Çetinkaya-Rundel 2014) and Introduction to Modern Statistics (2e) (Çetinkaya-Rundel and Hardin 2024). We have altered their work to fit our needs, primarily by piecing together information from their inference block. Additionally, the new sampling distributions chapter borrows heavily from the OpenIntro Statistics (Diez, Çetinkaya-Rundel, and Barr 2019) and from the sampling distributions lessons by Skew the Script (Skew the Script 2024). We have used their materials with minor modifications to transform the lesson activities into a book chapter.

Who is this book for?

We designed this book for the study of statistics that maximizes computational ideas while minimizing algebraic symbol manipulation. Although we do discuss traditional small-sample, normal-based inference and some of the classical probability distributions, we rely heavily on ideas such as simulation, permutations, and the bootstrap. This means that students with a background in differential and integral calculus will be successful with this book.

This book makes extensive using of the `R` programming language. In particular we focus both on the **tidyverse** and **mosaic** packages. We include a significant amount of code in our notes and frequently demonstrate multiple ways of completing a task. We have used this book for junior and sophomore college students.

Book structure and how to use it

This book is divided into four parts. Each part begins with a case study that introduces many of the main ideas of each part. Each chapter is designed to be a standalone 50 minute lesson. Within each chapter, we give exercises that can be worked in class and we provide learning objectives.

This book assumes students have access to `R`. Finally, we keep the number of homework problems to a reasonable level and assign all problems.

The four parts of the book are:

1. Descriptive Statistical Modeling: This part introduces the student to data collection methods, summary statistics, visual summaries, and exploratory data analysis.
2. Probability Modeling: We discuss the foundational ideas of probability, counting methods, and common distributions. We use both calculus and simulation to find moments and probabilities. We introduce basic ideas of multivariate probability. We include method of moments and maximum likelihood estimators.
3. Inferential Statistical Modeling: We discuss many of the basic inference ideas found in a traditional introductory statistics class but we add ideas of bootstrap and permutation methods.
4. Predictive Statistical Modeling: The final part introduces prediction methods, mainly in the form of linear regression. This part also includes inference for regression.

The learning outcomes for this course are to use computational and mathematical statistical/probabilistic concepts for:

a. Developing probabilistic models.

b. Developing statistical models for description, inference, and prediction.

c. Advancing practical and theoretical analytic experience and skills.

Prerequisites

To take this course, students are expected to have completed calculus up through and including integral calculus. We do have multivariate ideas in the course, but they are easily taught and don't require previous exposure to calculus III (multivariable calculus). We don't assume the students have any programming experience and, thus, we include a great deal of code. We have historically supplemented the course with Data Camp courses. We have also used Posit Cloud to help students get started in `R` without the burden of loading and maintaining software.

Packages

These notes make use of the following packages in `R`: **knitr** (Xie 2024), **rmarkdown** (Allaire et al. 2024), **mosaic** (R. Pruim, Kaplan, and Horton 2024), **tidyverse** (Wickham 2023), **ISLR** (James et al. 2021), **vcd** (Meyer et al. 2023), **ggplot2** (Wickham et al. 2024), **MASS** (Ripley 2024), **openintro** (Çetinkaya-Rundel et al. 2024), **broom** (Robinson, Hayes, and Couch 2024), **infer** (Bray et al. 2024), **kableExtra** (Zhu 2024), and **DT** (Xie, Cheng, and Tan 2024).

Acknowledgements

We have been lucky to have numerous open sources to help facilitate this work. Thank you to those who helped to provide edits including Jessica Hauschild, Justin Graham, Kris Pruitt, Matt Davis, and Skyler Royse.

File Creation Information

- File creation date: 2024-08-06
- R version 4.4.1 (2024-06-14)

References

Objectives

Descriptive Statistical Modeling

1 - Data Case Study

1. Use `R` for basic analysis and visualization.
2. Compile a PDF report using `knitr`.

2 - Data Basics

1) Differentiate between various statistical terminologies such as c*ase, observational unit, variables, data frame, tidy data, numerical variable, discrete numeric, continuous numeric, categorical variable, levels, scatterplot, associated variables*, and *independent*, and construct examples to demonstrate their proper use in context.
2) Within a given dataset, evaluate different types of variables and justify their classifications (e.g. categorical, discrete numerical, continuous numerical).
3) Given a study description, develop an appropriate research question and justify the organization of data as tidy.
4) Create and interpret scatterplots using `R` to analyze the relationship between two numerical variables by evaluating the strength and direction of the association.

3 - Overview of Data Collection Principles

1) Differentiate between various statistical terminologies such as *population, sample, anecdotal evidence, bias, simple random sample, systematic sample, representative sample, non-response bias, convenience sample, explanatory variable, response variable, observational study, cohort, experiment, randomized experiment,* and *placebo*, and construct examples to demonstrate their proper use in context.
2) Evaluate descriptions of research project to identify the population of interest, assess the generalizability of the study, determine the explanatory and response variables, classify the study as observational or experimental, and determine the type of sample used.

3) Design and justify sampling procedures for various research contexts, comparing the strengths and weaknesses of different sampling methods (simple random, systematic, convenience, etc.) and proposing improvements to minimize bias and enhance representativeness.

4 - Studies

1) Differentiate between various statistical terminologies such as *observational study, confounding variable, prospective study, retrospective study, simple random sampling, stratified sampling, strata, cluster sampling, multistage sampling, experiment, randomized experiment, control, replicate, blocking, treatment group, control group, blinded study, placebo, placebo effect,* and *double-blind*, and construct examples to demonstrate their proper use in context.
2) Evaluate study descriptions using appropriate terminology, and analyze the study design for potential biases or confounding variables.
3) Given a scenario, identify and assess flaws in reasoning, and propose robust study and sampling methodologies to address these flaws.

5 - Numerical Data

1) Differentiate between various statistical terminologies such as *scatterplot, mean, distribution, point estimate, weighted mean, histogram, data density, right skewed, left skewed, symmetric, mode, unimodal, bimodal, multimodal, variance, standard deviation, density plot, box plot, median, interquartile range, first quartile, third quartile, whiskers, outlier, robust estimate*, and *transformation*, and construct examples to demonstrate their proper use in context.
2) Using `R`, generate and interpret summary statistics for numerical variables.
3) Create and evaluate graphical summaries of numerical variables using `R`, choosing the most appropriate types of plots for different data characteristics and research questions.
4) Synthesize numerical and graphical summaries to provide interpretations and explanations of a data set.

6 - Categorical Data

1) Differentiate between various statistical terminologies such as *factor, contingency table, marginal counts, joint counts, frequency table, relative frequency table, bar plot, conditioning, segmented bar plot, mosaic plot, pie chart, side-by-side box plot,* and *density plot*, and construct examples to demonstrate their proper use in context.

2) Using `R`, generate and interpret tables for categorical variables.

3) Using `R`, generate and interpret summary statistics for numerical variables by groups.

4) Create and evaluate graphical summaries of both categorical and numerical variables using `R`, selecting the most appropriate visualization techniques for different types of data and research questions.

5) Synthesize numerical and graphical summaries to provide interpretations and explanations of a data set.

Probability Modeling

7 - Probability Case Study

1) Use `R` to simulate a probabilistic model.

2) Gain an introduction to probabilistic thinking through computational, mathematical, and data science approaches.

8 - Probability Rules

1) Differentiate between various statistical terminologies such as *sample space, outcome, event, subset, intersection, union, complement, probability, mutually exclusive, exhaustive, independent, multiplication rule, permutation,* and *combination*, and construct examples to demonstrate their proper use in context.

2) Apply basic probability properties and counting rules to calculate the probabilities of events in different scenarios. Interpret the calculated probabilities in context.

3) Explain and illustrate the basic axioms of probability.

4) Use `R` to perform calculations and simulations for determining the probabilities of events.

9 - Conditional Probability

1) Define and differentiate between conditional probability and joint probability, and provide real-world examples to illustrate these concepts and their differences.

2) Calculate conditional probabilities from given data or scenarios using their formal definition, and interpret these probabilities in the context of practical examples.

3) Using conditional probability, determine whether two events are independent and justify your conclusion with appropriate calculations and reasoning.

4) Apply Bayes' Rule to solve problems both mathematically and through simulation usinng `R`.

10 - Random Variables

1) Differentiate between various statistical terminologies such as *random variable, discrete random variable, continuous random variable, sample space/support, probability mass function, cumulative distribution function, moment, expectation, mean,* and *variance*, and construct examples to demonstrate their proper use in context.
2) For a given discrete random variable, derive and interpret the probability mass function (pmf) and apply this function to calculate the probabilities of various events.
3) Simulate random variables for a discrete distribution using `R`.
4) Calculate and interpret the moments, such as expected value/mean and variance, of a discrete random variable.
5) Calculate and interpret the expected value/mean and variance of a linear transformation of a random variable.

11 - Continuous Random Variables

1) Differentiate between various statistical terminologies such as *probability density function (pdf)* and *cumulative distribution function (cdf)* for continuous random variables, and construct examples to demonstrate their proper use in context.
2) For a given continuous random variable, derive and interpret the probability density function (pdf) and apply this function to calculate the probabilities of various events.
3) Calculate and interpret the moments, such as the expected value/mean and variance, of a continuous random variable.

12 - Named Discrete Distributions

1) Differentiate between common discrete distributions (Uniform, Binomial, Poisson) by identifying their parameters, assumptions, and moments. Evaluate scenarios to determine the most appropriate distribution to model various types of data.
2) Apply `R` to calculate probabilities and quantiles, and simulate random variables for common discrete distributions.

13 - Named Continuous Distributions

1) Differentiate between common continuous distributions (Uniform, Exponential, Normal) by identifying their parameters, assumptions, and moments. Evaluate scenarios to determine the most appropriate distribution to model various types of data.
2) Apply `R` to calculate probabilities and quantiles, and simulate random variables for common continuous distributions.
3) State and apply the empirical rule (68-95-99.7 rule).
4) Explain the relationship between the Poisson process and the Poisson and Exponential distributions, and describe how these distributions model different aspects of the same process.
5) Apply the memory-less property in context of the Exponential distribution and use it to simplify probability calculations.

14 - Multivariate Distributions

1) Differentiate between j*oint probability mass/density functions (pmf/pdf), marginal pmfs/pdfs*, and *conditional pmfs/pdfs*, and provide real-world examples to illustrate these concepts and their differences.
2) For a given joint pmf/pdf, derive the marginal and conditional pmfs/pdfs through summation or integration or using `R`.
3) Apply joint, marginal, and conditional pmfs/pdfs to calculate probabilities of events involving multiple random variables.

15 - Multivariate Expectation

1) Given a joint pmf/pdf, calculate and interpret the expected values/means and variances of random variables and functions of random variables.
2) Differentiate between covariance and correlation, and given a joint pmf/pdf, calculate and interpret the covariance and correlation between two random variables.
3) Given a joint pmf/pdf, determine whether random variables are independent of one another and justify your conclusion with appropriate calculations and reasoning.
4) Calculate and interpret conditional expectations for given joint pmfs/pdfs.

16 - Transformations

1) Determine the distribution of a transformed discrete random variable using appropriate methods, and use it to calculate probabilities.
2) Determine the distribution of a transformed continuous random variable using appropriate methods, and use it to calculate probabilities.
3) Determine the distribution of a transformation of multivariate random variables using simulation, and use it to calculate probabilities.

17 - Estimation Methods

1) Apply the method of moments to estimate parameters or sets of parameters from given data.
2) Derive the likelihood function for a given random sample from a distribution.
3) Derive a maximum likelihood estimate of a parameter or set of parameters.
4) Calculate and interpret the bias of an estimator by analyzing its expected value relative to the true parameter.

Inferential Statistical Modeling

18 - Inferential Thinking Case Study

1) Using bootstrap methods, obtain and interpret a confidence interval for an unknown parameter, based on a random sample.
2) Conduct a hypothesis test using a randomization test, to include all 4 steps.

19 - Sampling Distributions

1. Differentiate between various statistical terminologies such as *point estimate, parameter, sampling error, bias, sampling distribution,* and *standard error*, and construct examples to demonstrate their proper use in context.
2. Construct a sampling distribution for various statistics, including the sample mean, using `R`.
3. Using a sampling distribution, make decisions about the population. In other words, understand the effect of sampling variation on our estimates.

20 - Bootstrap

1) Differentiate between various statistical terminologies such as *sampling distribution, bootstrapping, bootstrap distribution, resample, sampling with replacement*, and *standard error*, and construct examples to demonstrate their proper use in context.
2) Apply the bootstrap technique to estimate the standard error of a sample statistic.
3) Utilize bootstrap methods to construct and interpret confidence intervals for unknown parameters from random samples.
4) Analyze and discuss the advantages, disadvantages, and underlying assumptions of bootstrapping for constructing confidence intervals.

21 - Hypothesis Testing with Simulation

1) Differentiate between various statistical terminologies such as *null hypothesis, alternative hypothesis, test statistic, p-value, randomization test, one-sided test, two-sided test, statistically significant, significance level, type I error, type II error, false positive, false negative, null distribution,* and *sampling distribution*, and construct examples to demonstrate their proper use in context.
2) Apply and evaluate all four steps of a hypothesis test using randomization methods: formulating hypotheses, calculating a test statistic, determining the p-value through randomization, and making a decision based on the test outcome.
3) Analyze and discuss the concepts of decision errors (type I and type II errors), the differences between one-sided and two-sided tests, and the impact of choosing a significance level. Evaluate how these factors influence the conclusions and reliability of hypothesis tests and their practical implications in statistical decision-making.
4) Analyze how confidence intervals and hypothesis testing complement each other in making statistical inferences.

22 - Hypothesis Testing with Known Distributions

1) Differentiate between various statistical terminologies such as *permutation test, exact test, null hypothesis, alternative hypothesis, test statistic, p-value*, and *power*, and construct examples to demonstrate their proper use in context.
2) Apply and evaluate all four steps of a hypothesis test using probability models: formulating hypotheses, calculating a test statistic, determining the p-value through randomization, and making a decision based on the test outcome.

23 - Hypothesis Testing with the Central Limit Theorem

1) Explain the central limit theorem and when it can be used for inference.
2) Apply the CLT to conduct hypothesis tests using `R` and interpret the results with an understanding of the CLT's role in justifying normal approximations.
3) Analyze the relationship between the t-distribution and normal distribution, explain usage contexts, and evaluate how changes in parameters impact their shape and location using visualizations.

24 - Confidence Intervals

1) Apply asymptotic methods based on the normal distribution to construct and interpret confidence intervals for unknown parameters.
2) Analyze the relationships between confidence intervals, confidence level, and sample size.
3) Analyze how confidence intervals and hypothesis testing complement each other in making statistical inferences.

25 - Additional Hypothesis Tests

1) Conduct and interpret a goodness of fit test using both Pearson's chi-squared and randomization to evaluate the independence between two categorical variables. Evaluate the assumptions for Pearson's chi-square test.
2) Analyze the relationship between the chi-squared and normal distributions, explain usage contexts, and evaluate the effects of changing degrees of freedom on the chi-squared distribution using visualizations.
3) Conduct and interpret a hypothesis test for equality of two means and equality of two variances using both permutation and the CLT. Evaluate the assumptions for two-sample t-tests.

26 - Analysis of Variance

1) Conduct and interpret a hypothesis test for equality of two or more means using both permutation and the F distribution. Evaluate the assumptions of ANOVA.

Predictive Statistical Modeling

27 - Linear Regression Case Study

1) Using `R`, generate a linear regression model and use it to produce a prediction model.
2) Using plots, check the assumptions of a linear regression model.

28 - Linear Regression Basics

1) Differentiate between various statistical terminologies such as *response, predictor, linear regression, simple linear regression, coefficients, residual,* and *extrapolation*, and construct examples to demonstrate their proper use in context.
2) Estimate the parameters of a simple linear regression model using a given sample of data.
3) Interpret the coefficients of a simple linear regression model.
4) Create and evaluate scatterplots with regression lines.
5) Identify and assess the assumptions underlying linear regression models.

29 - Linear Regression Inference

1) Apply statistical inference methods for β_0 and β_1, and evaluate the implications for the predictor-response relationship.
2) Write the estimated simple linear regression model and calculate and interpret the predicted response for a given value of the predictor.
3) Construct and interpret confidence and prediction intervals for the response variable.

30 - Linear Regression Diagnostics

1) Calculate and interpret the R-squared and F-statistic for a linear regression model. Evaluate these metrics to assess the model's goodness-of-fit and overall significance.
2) Use `R` to evaluate the assumptions underlying a linear regression model.
3) Identify, analyze, and explain the impact of outliers and leverage points in a linear regression model.

31 - Simulated-Based Linear Regression

1) Apply the bootstrap to generate and interpret confidence intervals and estimates of standard error for parameter estimates in a linear regression model.
2) Apply the bootstrap to generate and interpret confidence intervals for predicted values from a linear regression model.
3) Generate bootstrap samples using two methods: sampling rows of the data and sampling residuals. Justify why you might prefer one method over the other.
4) Generate and interpret regression coefficients in a linear regression model with categorical explanatory variables.

32 - Multiple Linear Regression

1) Generate and interpret the coefficients of a multiple linear regression model. Asses the assumptions underlying multiple linear regression models.
2) Write the estimates multiple linear regression model and calculate and interpret the predicted response for given values of the predictors.
3) Generate and interpret confidence intervals for parameter estimates in a multiple linear regression model.
4) Generate and interpret confidence and prediction intervals for predicted values in a multiple linear regression model.
5) Explain the concepts of adjusted R-squared and multicollinearity in the context of multiple linear regression.
6) Develop and interpret multiple linear regression models that include higher-order terms, such as polynomial terms or interaction effects.

33 - Logistic Regression

1) Apply logistic regression using `R` to analyze binary outcome data. Interpret the regression output, and perform model selection.
2) Write the estimated logistic regression, and calculate and interpret the predicted outputs for given values of the predictors.
3) Calculate and interpret confidence intervals for parameter estimates and predicted probabilities in a logistic regression model.
4) Generate and analyze a confusion matrix to evaluate the performance of a logistic regression model.

Part I

Descriptive Statistical Modeling

1 Data Case Study

1.1 Objectives

1. Use `R` for basic analysis and visualization.
2. Compile a PDF report using `knitr`.

1.2 Introduction to descriptive statistical modeling

In this first block of material, we will focus on data types, collection methods, summaries, and visualizations. We also intend to introduce computing via the `R` package. Programming in `R` requires some focus early in this book and we will supplement with some online courses. There is relatively little mathematics in this first block.

1.3 The data analytic process

Scientists seek to answer questions using rigorous methods and careful observations. These observations – collected from the likes of field notes, surveys, and experiments – form the backbone of a statistical investigation and are called **data**. Statistics is the study of how best to collect, analyze, and draw conclusions from data. It is helpful to put statistics in the context of a general process of investigation:

1. Identify a question or problem.
2. Collect relevant data on the topic.
3. Explore and understand the data.
4. Analyze the data.
5. Form a conclusion.
6. Make decisions based on the conclusion.

This is typical of an explanatory process because it starts with a research question and proceeds. However, sometimes an analysis is exploratory in nature. There is data but not necessarily a research question. The purpose of the analysis is to find interesting features in the data and sometimes generate hypotheses. In this book, we focus on the explanatory aspects of analysis.

Statistics as a subject focuses on making stages 2-5 objective, rigorous, and efficient. That is, statistics has three primary components:

- How best can we collect data?
- How should it be analyzed?
- And what can we infer from the analysis?

The topics scientists investigate are as diverse as the questions they ask. However, many of these investigations can be addressed with a small number of data collection techniques, analytic tools, and fundamental concepts in statistical inference. This chapter provides a glimpse into these and other themes we will encounter throughout the rest of the book.

1.4 Case study

In this chapter, we will consider an experiment that studies effectiveness of stents in treating patients at risk of stroke. [1] [2] Stents are small mesh tubes that are placed inside narrow or weak arteries to assist in patient recovery after cardiac events and reduce the risk of an additional heart attack or death. Many doctors have hoped that there would be similar benefits for patients at risk of stroke. We start by writing the principal question the researchers hope to answer:

1.4.1 Research question

> Does the use of stents reduce the risk of stroke?

[1] Chimowitz MI, Lynn MJ, Derdeyn CP, et al. 2011. Stenting versus Aggressive Medical Therapy for Intracranial Arterial Stenosis. New England Journal of Medicine 365:993-1003.

[2] NY Times article reporting on the study: http://www.nytimes.com/2011/09/08/health/research/08stent.html

1.4.2 Collect the relevant data

The researchers who asked this question collected data on 451 at-risk patients. Each volunteer patient was randomly assigned to one of two groups:

Treatment group. Patients in the treatment group received a stent and medical management. The medical management included medications, management of risk factors, and help in lifestyle modification.

Control group. Patients in the control group received the same medical management as the treatment group but did not receive stents.

Researchers randomly assigned 224 patients to the treatment group and 227 to the control group. In this study, the control group provides a reference point against which we can measure the medical impact of stents in the treatment group.

This is an experiment and not an observational study. We will learn more about these ideas in this block.

Researchers studied the effect of stents at two time points: 30 days after enrollment and 365 days after enrollment.

1.4.3 Import data

We begin our first use of `R`.

If you need to install a package, most likely it will be on CRAN, the Comprehensive R Archive Network. Before a package can be used, it must be installed on the computer (once per computer or account) and loaded into a session (once per `R` session). When you exit `R`, the package stays installed on the computer but will not be reloaded when `R` is started again.

In summary, `R` has packages that can be downloaded and installed from online repositories such as CRAN. When you install a package, which only needs to be done once per computer or account, in `R` all it is doing is placing the source code in a library folder designated during the installation of `R`. Packages are typically collections of functions and variables that are specific to a certain task or subject matter.

For example, to install the **mosaic** package, enter:

```
install.packages("mosaic") # fetch package from CRAN
```

In RStudio, there is a *Packages* tab that makes it easy to add and maintain packages.

To use a package in a session, we must load it. This makes it available to the current session only. When you start `R` again, you will have to load packages again. The command `library()` with the package name supplied as the argument is all that is needed. For this session, we

will load **tidyverse** and **mosaic**. Note: the box below is executing the `R` commands, this is known as reproducible research since you can see the code and then you can run or modify as you need.

```
library(tidyverse)
library(mosaic)
```

Next read in the data into the working environment.

```
# This code reads the `stent_study.csv` file into the `stent_study` object.
stent_study <- read_csv("data/stent_study.csv")
```

Note on commenting code: It is good practice to comment code. Here are some of the best practices for commenting computer code:

Comments should explain why code is written the way it is, rather than explaining what the code does. This means that you should explain the intent of the code, not just the steps that it takes to achieve that intent.
Comments should be brief and to the point. There is no need to write long, rambling comments. Just write enough to explain what the code is doing and why.
Comments should be clear and concise. Use plain language that is easy to understand. Avoid jargon and technical terms that the reader may not be familiar with.
Comments should be consistent with the style of the code. If the code is written in a formal style, then the comments should also be formal. If the code is written in a more informal style, then the comments should be informal.
Comments should be up-to-date. If you make changes to the code, then you should also update the comments to reflect those changes.

In additional, consider the following practices in writing your code:

Using a consistent comment style. This will make it easier for other people to read and understand your code.
Using meaningful names for variables and functions. This will help to reduce the need for comments.
Use indentation and whitespace to make your code easier to read. This will also help to reduce the need for comments.
Document your code. This means writing a separate document that explains the purpose of the code, how to use it, and any known limitations.

By following these best practices, you can write code that is easy to understand and maintain. This will make your code more reusable and will help to prevent errors.

Now back to our code. Let's break this code down. We are reading from a .csv file and assigning the results into an object called `stent_study`. The assignment arrow `<-` means

we assign what is on the right to what is on the left. The R function we use in this case is `read_csv()`. When using R functions, you should ask yourself:

1. What do I want R to do?
2. What information must I provide for R to do this?

We want R to read in a .csv file. We can get help on this function by typing `?read_csv` or `help(read_csv)` at the prompt. The only required input to `read_csv()` is the file location. We have our data stored in a folder called "data" under the working directory. We can determine the working directory by typing `getwd()` at the prompt.

```
getwd()
```

Similarly, if we wish to change the working directory, we can do so by using the `setwd()` function:

```
setwd('C:/Users/Brianna.Hitt/Documents/ProbStat/Another Folder')
```

In R if you use the `view()`, you will see the data in what looks like a standard spreadsheet.

```
View(stent_study)
```

1.4.4 Explore data

Before we attempt to answer the research question, let's look at the data. We want R to print out the first 10 rows of the data. The appropriate function is `head()` and it needs the data object. By default, R will output the first 6 rows. By using the `n =` argument, we can specify how many rows we want to view.

```
head(stent_study, n = 10)
```

```
# A tibble: 10 x 3
   group   outcome30 outcome365
   <chr>   <chr>     <chr>
 1 control no_event  no_event
 2 trmt    no_event  no_event
 3 control no_event  no_event
 4 trmt    no_event  no_event
 5 trmt    no_event  no_event
 6 control no_event  no_event
```

```
 7 trmt    no_event  no_event
 8 control no_event  no_event
 9 control no_event  no_event
10 control no_event  no_event
```

We also want to "inspect" the data. The function is `inspect()` and `R` needs the data object `stent_study`.

```
inspect(stent_study)
```

```
categorical variables:
        name     class levels   n missing
1      group character      2 451       0
2  outcome30 character      2 451       0
3 outcome365 character      2 451       0
                                   distribution
1 control (50.3%), trmt (49.7%)
2 no_event (89.8%), stroke (10.2%)
3 no_event (83.8%), stroke (16.2%)
```

To keep things simple, we will only look at the `outcome30` variable in this case study. We will summarize the data in a table. Later in the book, we will learn to do this using the **tidy** package; for now we use the **mosaic** package. This package makes use of the modeling formula that you will use extensively later in this book. The modeling formula is also used in Math 378.

We want to summarize the data by making a table. From `mosaic`, we use the `tally()` function. Before using this function, we have to understand the basic formula notation that `mosaic` uses. The basic format is:

```
goal(y ~ x, data = MyData, ...) # pseudo-code for the formula template
```

We read `y ~ x` as "y tilde x" and interpret it in the equivalent forms: "y broken down by x"; "y modeled by x"; "y explained by x"; "y depends on x"; or "y accounted for by x." For graphics, it's reasonable to read the formula as "y vs. x", which is exactly the convention used for coordinate axes.

For this exercise, we want to apply `tally()` to the variables `group` and `outcome30`. In this case it does not matter which we call `y` and `x`; however, it is more natural to think of `outcome30` as a dependent variable.

```
tally(outcome30 ~ group, data = stent_study, margins = TRUE)
```

```
          group
outcome30  control trmt
  no_event     214  191
  stroke        13   33
  Total        227  224
```

The `margins` option totals the columns.

Of the 224 patients in the treatment group, 33 had a stroke by the end of the first month. Using these two numbers, we can use `R` to compute the proportion of patients in the treatment group who had a stroke by the end of their first month.

```
33 / (33 + 191)
```

```
[1] 0.1473214
```

> **Exercise**:
> What proportion of the control group had a stroke in the first 30 days of the study? And why is this proportion different from the proportion reported by `inspect()`?

Let's have `R` calculate proportions for us. Use `?` or `help()` to look at the help menu for `tally()`. Note that one of the option arguments of the `tally()` function is `format =`. Setting this equal to `proportion` will output the proportions instead of the counts.

```
tally(outcome30 ~ group, data = stent_study, format = 'proportion', margins = TRUE)
```

```
          group
outcome30     control       trmt
  no_event 0.94273128 0.85267857
  stroke   0.05726872 0.14732143
  Total    1.00000000 1.00000000
```

We can compute summary statistics from the table. A **summary statistic** is a single number summarizing a large amount of data.[3] For instance, the primary results of the study after 1 month could be described by two summary statistics: the proportion of people who had a stroke in the treatment group and the proportion of people who had a stroke in the control group.

[3] Formally, a summary statistic is a value computed from the data. Some summary statistics are more useful than others.

- Proportion who had a stroke in the treatment (stent) group: $33/224 = 0.15 = 15\%$
- Proportion who had a stroke in the control group: $13/227 = 0.06 = 6\%$

1.4.5 Visualize the data

It is often important to visualize the data. The table is a type of visualization, but in this section we will introduce a graphical method called bar charts.

We will use the **ggformula** package to visualize the data. It is a wrapper to the **ggplot2** package which is becoming the industry standard for generating professional graphics. However, the interface for **ggplot2** can be difficult to learn and we will ease into it by using `ggformula`, which makes use of the formula notation introduced above. The **ggformula** package was loaded when we loaded `mosaic`.[4]

To generate a basic graphic, we need to ask ourselves what information we are trying to see, what particular type of graph is best, what corresponding `R` function to use, and what information that `R` function needs in order to build a plot. For categorical data, we want a bar chart and the `R` function `gf_bar()` needs the data object and the variable(s) of interest.

Here is our first attempt. In Figure **??**, we leave the `y` portion of our formula blank. Doing this implies that we simply want to view the number/count of `outcome30` by type. We will see the two levels of `outcome30` on the x-axis and counts on the y-axis.

(ref:ggfbold) Using **ggformula** to create a bar chart.

```
gf_bar(~outcome30, data = stent_study)
```

[4]https://cran.r-project.org/web/packages/ggformula/vignettes/ggformula-blog.html

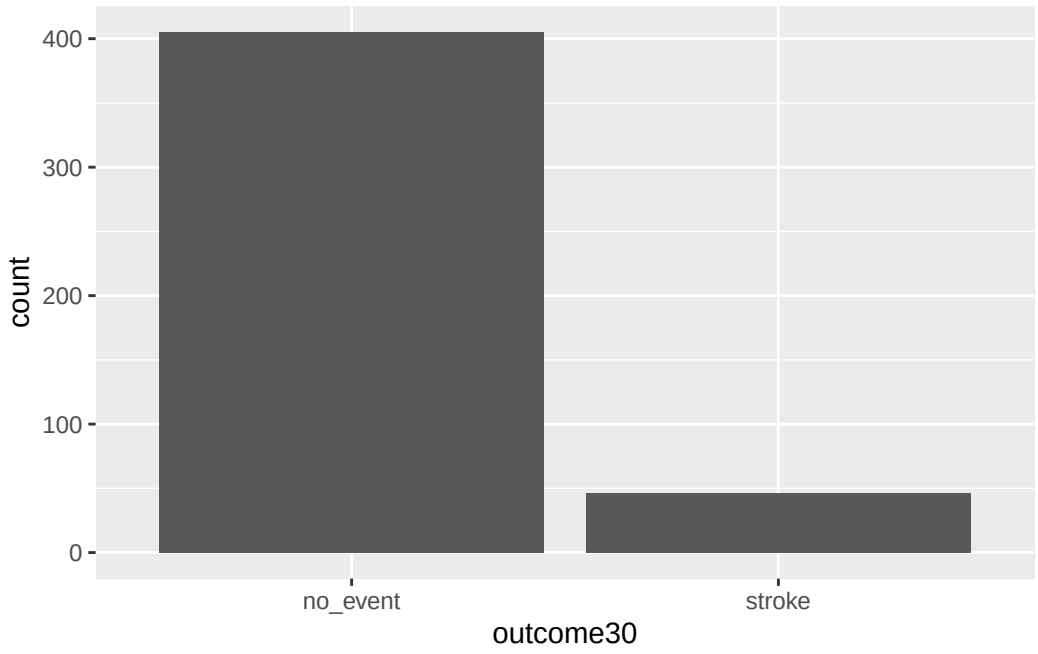


Figure 1.1: Using **ggformula** to create a bar chart.

Exercise:
Explain Figure **??**.

This plot graphically shows us the total number of "stroke" and the total number of "no_event". However, this is not what we want. We want to compare the 30-day outcomes for both treatment groups. So, we need to break the data into different groups based on treatment type. In the formula notation, we now update it to the form:

```
goal(y ~ x|z, data = MyData, ...) # pseudo-code for the formula template
```

We read `y ~ x|z` as "y tilde x by z" and interpret it in the equivalent forms: "y modeled by x for each z"; "y explained by x within each z"; or "y accounted for by x within z." For graphics, it's reasonable to read the formula as "y vs. x for each z". Figure Figure **??** shows the results.

```
gf_bar(~outcome30|group, data = stent_study)
```

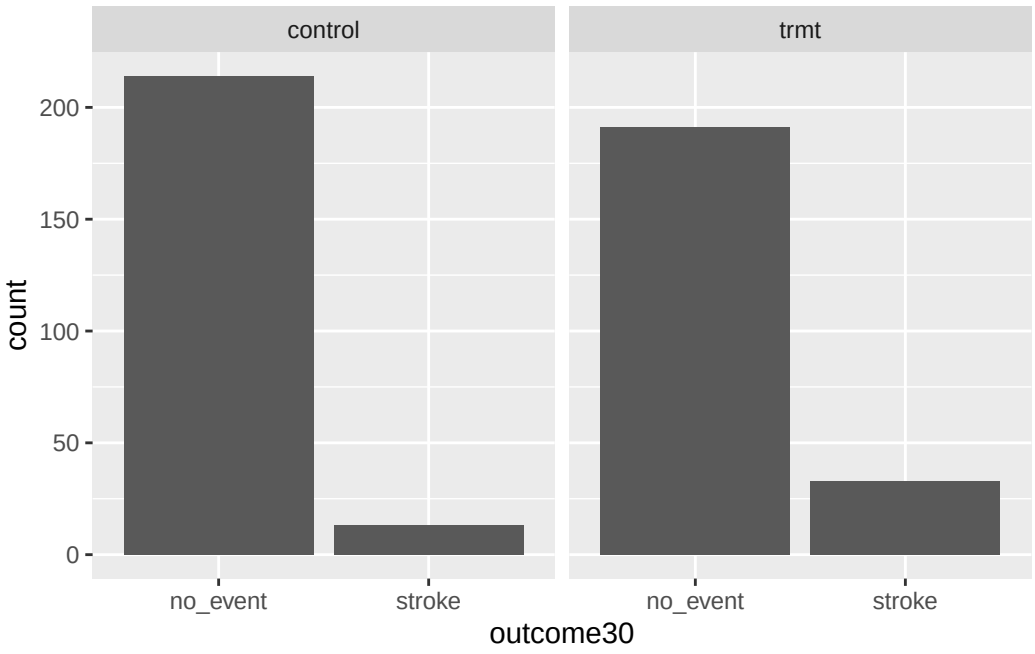


Figure 1.2: Bar charts conditioned on the `group` variable.

1.4.5.1 More advanced graphics

As a prelude for things to come, the above graphic needs work. The labels don't help and there is no title. We could add color. Does it make more sense to use proportions? Here is the code and results for a better graph, see Figure Figure **??**. Don't worry if this seems a bit advanced, but feel free to examine each new component of this code.

```
# This code creates a graph showing the impact of stents on stroke.
# The `gf_props()` function creates a bar graph showing the number of events
# for each experimental group. The `fill` argument specifies the fill color
# for each group. The `position = 'fill'` argument specifies that the bars
# should be filled to the top.

# The `gf_labs()` function adds the title, subtitle, x-axis label, and y-axis
# label to the graph.

# The `gf_theme()` function applies a black-and-white theme to the graph.

stent_study %>%
gf_props(~group, fill = ~outcome30, position = 'fill') %>%
  gf_labs(title = "Impact of Stents of Stroke",
```

```
         subtitle = 'Experiment with 451 Patients',
         x = "Experimental Group",
         y = "Number of Events") %>%
gf_theme(theme_bw())
```

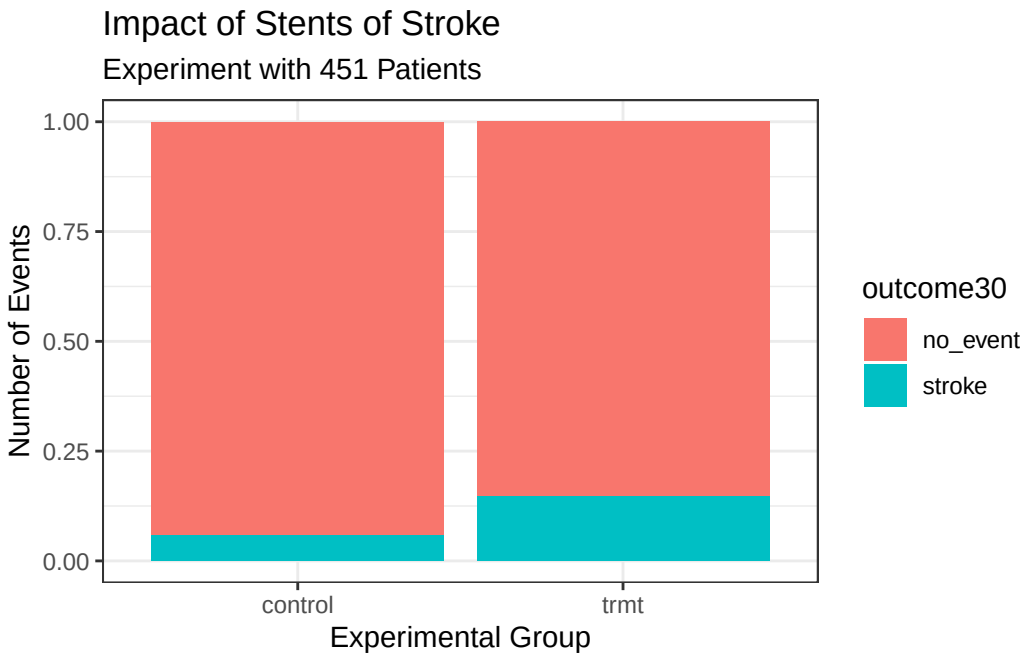


Figure 1.3: Better graph.

Notice that we used the pipe operator, `%>%`. This operator allows us to string functions together in a manner that makes it easier to read the code. In the above code, we are sending the data object `stent_study` into the function `gf_props()` to use as data, so we don't need the `data =` argument. In math, this is a composition of functions. Instead of `f(g(x))` we could use a pipe `f(g(x)) = g(x) %>% f()`.

1.4.6 Conclusion

These two summary statistics (the proportions of people who had a stroke) are useful in looking for differences in the groups, and we are in for a surprise: an additional 9% of patients in the treatment group had a stroke! This is important for two reasons. First, it is contrary to what doctors expected, which was that stents would *reduce* the rate of strokes. Second, it leads to a statistical question: do the data show a real difference due to the treatment?

This second question is subtle. Suppose you flip a coin 100 times. While the chance a coin lands heads in any given coin flip is 50%, we probably won't observe exactly 50 heads. This

type of fluctuation is part of almost any type of data generating process. It is possible that the 9% difference in the stent study is due to this natural variation. However, the larger the difference we observe (for a particular sample size), the less believable it is that the difference is due to chance. So what we are really asking is the following: is the difference so large that we should reject the notion that it was due to chance?

This is a preview of step 4, analyze the data, and step 5, form a conclusion, of the analysis cycle. While we haven't yet covered statistical tools to fully address these steps, we can comprehend the conclusions of the published analysis: there was compelling evidence of harm by stents in this study of stroke patients.

Be careful: Do not generalize the results of this study to all patients and all stents. This study looked at patients with very specific characteristics who volunteered to be a part of this study and who may not be representative of all stroke patients. In addition, there are many types of stents and this study only considered the self-expanding Wingspan stent (Boston Scientific). However, this study does leave us with an important lesson: we should keep our eyes open for surprises.

2 Data Basics

2.1 Objectives

1) Differentiate between various statistical terminologies such as *case, observational unit, variables, data frame, tidy data, numerical variable, discrete numeric variable, continuous numeric variable, categorical variable, levels, scatterplot, associated variables*, and *independent*, and construct examples to demonstrate their proper use in context.

2) Within a given dataset, evaluate different types of variables and justify their classifications (e.g. categorical, discrete numerical, continuous numerical).

3) Given a study description, develop an appropriate research question and justify the organization of data as tidy.

4) Create and interpret scatterplots using `R` to analyze the relationship between two numerical variables by evaluating the strength and direction of the association.

2.2 Data basics

Effective presentation and description of data is a first step in most analyses. This chapter introduces one structure for organizing data, as well as some terminology that will be used throughout this book.

2.2.1 Observations, variables, and data matrices

For reference we will be using a data set concerning 50 emails received in 2012. These observations will be referred to as the `email50` data set, and they are a random sample from a larger data set. This data is in the **openintro** package so let's install and then load this package.

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

Table **??** shows 4 rows of the `email50` data set and we have elected to only list 5 variables for ease of observation.

Each row in the table represents a single email or **case**. A case is also sometimes called a **unit of observation** or an **observational unit**. The columns represent **variables**, which represent characteristics for each of the cases (emails). For example, the first row represents email 1, which is not spam, contains 21,705 characters, 551 line breaks, is written in HTML format, and contains only small numbers.

Table 2.1: First 5 rows of email data frame

spam	num_char	line_breaks	format	number
0	21.705	551	1	small
0	7.011	183	1	big
1	0.631	28	0	none
0	15.829	242	1	small

Let's look at the first 10 rows of data from `email50` using `R`. Remember to ask the two questions:

What do we want `R` to do? and

What must we give `R` for it to do this?

We want the first 10 rows so we use `head()` and `R` needs the data object and the number of rows. The data object is called `email50` and is accessible once the **openintro** package is loaded.

```
head(email50, n = 10)
```

```
# A tibble: 10 x 21
   spam  to_multiple from     cc sent_email time                image attach
   <fct> <fct>       <fct> <int> <fct>      <dttm>              <dbl>  <dbl>
 1 0     0           1         0 1          2012-01-04 13:19:16     0      0
 2 0     0           1         0 0          2012-02-16 20:10:06     0      0
 3 1     0           1         4 0          2012-01-04 15:36:23     0      2
 4 0     0           1         0 0          2012-01-04 17:49:52     0      0
 5 0     0           1         0 0          2012-01-27 09:34:45     0      0
 6 0     0           1         0 0          2012-01-17 17:31:57     0      0
 7 0     0           1         0 0          2012-03-18 04:18:55     0      0
 8 0     0           1         0 1          2012-03-31 13:58:56     0      0
 9 0     0           1         1 1          2012-01-11 01:57:54     0      0
10 0     0           1         0 0          2012-01-07 19:29:16     0      0
```

```
# i 13 more variables: dollar <dbl>, winner <fct>, inherit <dbl>, viagra <dbl>,
#   password <dbl>, num_char <dbl>, line_breaks <int>, format <fct>,
#   re_subj <fct>, exclaim_subj <dbl>, urgent_subj <fct>, exclaim_mess <dbl>,
#   number <fct>
```

In practice, it is especially important to ask clarifying questions to ensure important aspects of the data are understood. For instance, it is always important to be sure we know what each variable means and the units of measurement. Descriptions of all variables in the `email50` data set are given in its documentation which can be accessed in `R` by using the `?` command:

```
?email50
```

(Note that not all data sets will have associated documentation; the authors of **openintro** package included this documentation with the `email50` data set contained in the package.)

The data in `email50` represent a **data matrix**, or in `R` terminology a **data frame** or **tibble** [1], which is a common way to organize data. Each row of a data matrix corresponds to a unique case, and each column corresponds to a variable. This is called **tidy data**.[2] The data frame for the stroke study introduced in the previous chapter had patients as the cases and there were three variables recorded for each patient. If we are thinking of patients as the unit of observation, then this data is tidy.

```
# A tibble: 10 x 3
   group   outcome30 outcome365
   <chr>   <chr>     <chr>
 1 control no_event  no_event
 2 trmt    no_event  no_event
 3 control no_event  no_event
 4 trmt    no_event  no_event
 5 trmt    no_event  no_event
 6 control no_event  no_event
 7 trmt    no_event  no_event
 8 control no_event  no_event
 9 control no_event  no_event
10 control no_event  no_event
```

If we think of an outcome as a unit of observation, then it is not tidy since the two outcome columns are variable values (month or year). The tidy data for this case would be:

[1] A tibble is a data frame with attributes for such things as better display and printing.

[2] Tidy data is data in which each row corresponds to a unique case and each column represents a single variable. For more information on tidy data, see the *Simply Statistics* blog and the *R for Data Science* book by Hadley Wickham and Garrett Grolemund.

```
# A tibble: 10 x 4
   patient_id group   time  result
        <int> <chr>   <chr> <chr>
 1          1 control month no_event
 2          1 control year  no_event
 3          2 trmt    month no_event
 4          2 trmt    year  no_event
 5          3 control month no_event
 6          3 control year  no_event
 7          4 trmt    month no_event
 8          4 trmt    year  no_event
 9          5 trmt    month no_event
10          5 trmt    year  no_event
```

There are three interrelated rules which make a data set tidy:

1. Each variable must have its own column.

2. Each observation must have its own row.

3. Each value must have its own cell.

Why ensure that your data is tidy? There are two main advantages:

1. There's a general advantage to picking one consistent way of storing data. If you have a consistent data structure, it's easier to learn the tools that work with it because they have an underlying uniformity.

2. There's a specific advantage to placing variables in columns because it allows `R`'s vectorized nature to shine. This will be more clear as we progress in our studies. Since most built-in `R` functions work with vectors of values, it makes transforming tidy data feel particularly natural.

Data frames are a convenient way to record and store data. If another individual or case is added to the data set, an additional row can be easily added. Similarly, another column can be added for a new variable.

> **Exercise**:
> We consider a publicly available data set that summarizes information about the 3,142 counties in the United States, and we create a data set called `county_subset` data set. This data set will include information about each county: its name, the state where it resides, its population in 2000 and 2010, per capita federal spending, poverty rate, and four additional characteristics. We create this data object in the code following this description. The parent data set is part of the `usdata` library

and is called `county_complete`. The variables are summarized in the help menu built into the **usdata** package[3]. How might these data be organized in a data matrix? [4]

Using `R` we will create our data object. First we load the library `usdata`.

```
library(usdata)
```

We only want a subset of the columns and we will use the `select` verb in `dplyr` to select and rename columns. We also create a new variable which is federal spending per capita using the `mutate` function.

```
county_subset <- county_complete %>%
  select(name, state, pop2000, pop2010, fed_spend = fed_spending_2009,
         poverty = poverty_2010, homeownership = homeownership_2010,
         multi_unit = housing_multi_unit_2010, income = per_capita_income_2010,
         med_income = median_household_income_2010) %>%
  mutate(fed_spend = fed_spend / pop2010)
```

Using `R`, we will display seven rows of the `county_subset` data frame.

```
head(county_subset, n = 7)
```

```
            name   state pop2000 pop2010 fed_spend poverty homeownership
1 Autauga County Alabama   43671   54571  6.068095    10.6          77.5
2 Baldwin County Alabama  140415  182265  6.139862    12.2          76.7
3 Barbour County Alabama   29038   27457  8.752158    25.0          68.0
4    Bibb County Alabama   20826   22915  7.122016    12.6          82.9
5  Blount County Alabama   51024   57322  5.130910    13.4          82.0
6 Bullock County Alabama   11714   10914  9.973062    25.3          76.9
7  Butler County Alabama   21399   20947  9.311835    25.0          69.0
  multi_unit income med_income
1        7.2  24568      53255
2       22.6  26469      50147
3       11.1  15875      33219
4        6.6  19918      41770
5        3.7  21070      45549
6        9.9  20289      31602
7       13.7  16916      30659
```

[3] These data were collected from the US Census website.

[4] Each county may be viewed as a case, and there are ten pieces of information recorded for each case. A table with 3,142 rows and 10 columns could hold these data, where each row represents a county and each column represents a particular piece of information.

2.2.2 Types of variables

Examine the `fed_spend`, `pop2010`, and `state` variables in the `county` data set. Each of these variables is inherently different from the others, yet many of them share certain characteristics.

First consider `fed_spend`. It is said to be a **numerical variable** (sometimes called a quantitative variable) since it can take a wide range of numerical values, and it is sensible to add, subtract, or take averages with those values. On the other hand, we would not classify a variable reporting telephone area codes as numerical; even though area codes are made up of numerical digits, their average, sum, and difference have no clear meaning.

The `pop2010` variable is also numerical; it is sensible to add, subtract, or take averages with those values, although it seems to be a little different than `fed_spend`. This variable of the population count can only be a whole non-negative number (0, 1, 2, ...). For this reason, the population variable is said to be **discrete** since it can only take specific numerical values. On the other hand, the federal spending variable is said to be **continuous** because it can take on any value in some interval. Now technically, there are no truly continuous numerical variables since all measurements are finite up to some level of accuracy or measurement precision (e.g., we typically measure federal spending in dollars and cents). However, in this book, we will treat both types of numerical variables the same, that is as continuous variables for statistical modeling. The only place this will be different in this book is in probability models, which we will see in the probability modeling block.

The variable `state` can take up to 51 values, after accounting for Washington, DC, and are summarized as: *Alabama*, *Alaska*, ..., and *Wyoming*. Because the responses themselves are categories, `state` is a **categorical variable** (sometimes also called a qualitative variable), and the possible values are called the variable's **levels**.

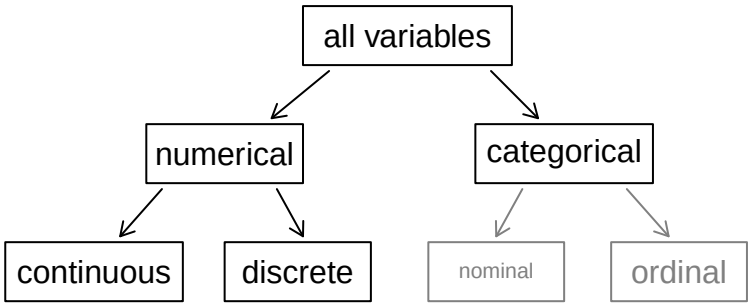


Figure 2.1: Taxonomy of Variables.

Finally, consider a hypothetical variable on education, which describes the highest level of education completed and takes on one of the values *noHS*, *HS*, *College* or *Graduate_school*. This variable seems to be a hybrid: it is a categorical variable but the levels have a natural ordering. A variable with these properties is called an **ordinal** variable. A categorical variable

with levels that do not have a natural ordering is called a **nominal** variable. To simplify analyses, any ordinal variables in this book will be treated as nominal categorical variables. In `R`, categorical variables can be treated in different ways; one of the key differences is that we can leave them as character values (character strings, or text) or as factors. A factor is essentially a categorical variable with defined *levels*. When `R` handles factors, it is only concerned about the *levels* of the factors. We will learn more about this as we progress.

Figure **??** captures this classification of variables we have described.

> **Exercise**:
> Data were collected about students in a statistics course. Three variables were recorded for each student: number of siblings, student height, and whether the student had previously taken a statistics course. Classify each of the variables as continuous numerical, discrete numerical, or categorical.[5]

> **Exercise**:
> Consider the variables `group` and `outcome30` from the stent study in the case study chapter. Are these numerical or categorical variables? [6]

2.2.3 Relationships between variables

Many analyses are motivated by a researcher looking for a relationship between two or more variables. This is the heart of statistical modeling. A social scientist may like to answer some of the following questions:

1. Is federal spending, on average, higher or lower in counties with high rates of poverty?

2. If homeownership is lower than the national average in one county, will the percent of multi-unit structures in that county likely be above or below the national average?

These are what statisticians refer to as **research questions**, specific and measurable questions that guide the data collection and analysis process. To answer these questions, data must be collected, such as the `county_complete` data set. Examining summary statistics could provide insights for each of the two questions about counties. Graphs can be used to visually summarize data and are useful for answering such questions as well.

Scatterplots are one type of graph used to study the relationship between two numerical variables. Figure **??** compares the variables `fed_spend` and `poverty`. Each point on the plot

[5] The number of siblings and student height represent numerical variables. Because the number of siblings is a count, it is discrete. Height varies continuously, so it is a continuous numerical variable. The last variable classifies students into two categories – those who have and those who have not taken a statistics course – which makes this variable categorical.

[6] There are only two possible values for each variable, and in both cases they describe categories. Thus, each is a categorical variable.

represents a single county. For instance, the highlighted dot corresponds to County 1088 in the `county_subset` data set: Owsley County, Kentucky, which had a poverty rate of 41.5% and federal spending of $21.50 per capita. The dense cloud in the scatterplot suggests a relationship between the two variables: counties with a high poverty rate also tend to have slightly more federal spending. We might brainstorm as to why this relationship exists and investigate each idea to determine which is the most reasonable explanation.

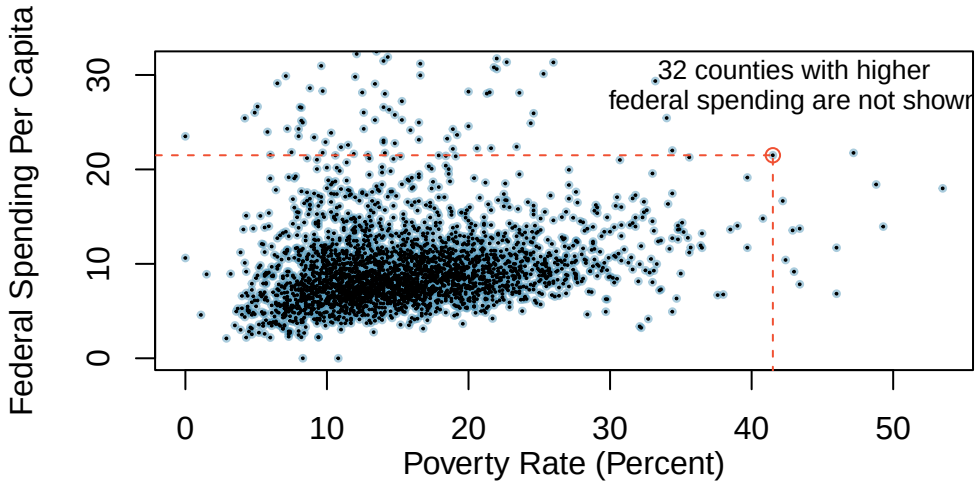


Figure 2.2: A scatterplot showing fed_spend against poverty. Owsley County of Kentucky, with a poverty rate of 41.5% and federal spending of $21.50 per capita, is highlighted.

Exercise:
Examine the variables in the `email50` data set. Create two research questions about the relationships between these variables that are of interest to you.[7]

The `fed_spend` and `poverty` variables are said to be associated because the plot shows a discernible pattern. When two variables show some connection with one another, they are called **associated variables**. Associated variables can also be called **dependent** variables and vice-versa.

Example:
The relationship between the homeownership rate and the percent of units in multi-unit structures (e.g. apartments, condos) is visualized using a scatterplot in Figure **??**. Are these variables associated?

It appears that the larger the fraction of units in multi-unit structures, the lower the homeownership rate. Since there is some relationship between the variables, they are associated.

[7] Two sample questions: (1) Intuition suggests that if there are many line breaks in an email then there would also tend to be many characters: does this hold true? (2) Is there a connection between whether an email format is plain text (versus HTML) and whether it is a spam message?

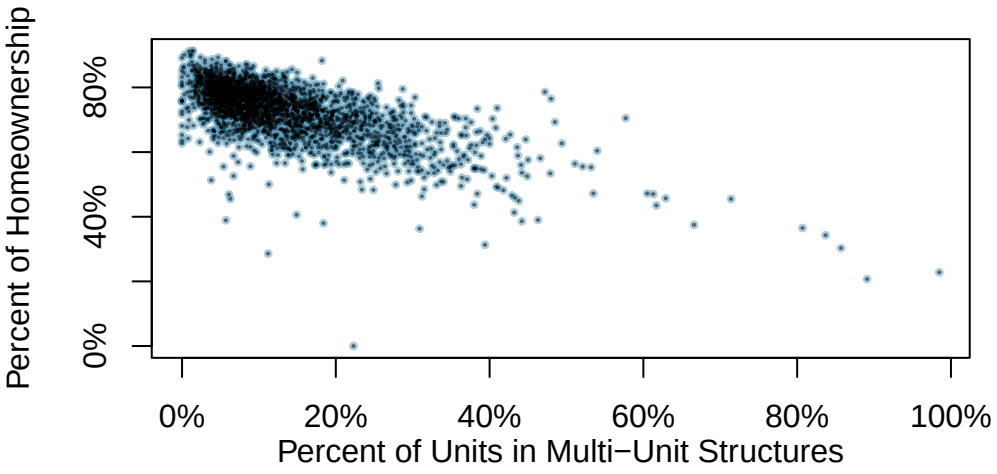


Figure 2.3: A scatterplot of the homeownership rate versus the percent of units that are in multi-unit structures for all 3,143 counties.

Because there is a downward trend in Figure **??** – counties with more units in multi-unit structures are associated with lower homeownership – these variables are said to be **negatively associated**. A **positive association** (upward trend) is shown in the relationship between the `poverty` and `fed_spend` variables represented in Figure **??**, where counties with higher poverty rates tend to receive more federal spending per capita.

If two variables are not associated, then they are said to be **independent**. That is, two variables are independent if there is no evident relationship between the two.

> A pair of variables are either related in some way (associated) or not (independent).
> No pair of variables is both associated and independent.

2.2.4 Creating a scatterplot

In this section, we will create a simple scatterplot and then ask you to create one on your own. First, we will recreate the scatterplot seen in Figure **??**. This figure uses the `county_subset` data set.

Here are two questions:

What do we want `R` to do? and

What must we give `R` for it to do this?

We want `R` to create a scatterplot and to do this it needs, at a minimum, the data object, what we want on the x-axis, and what we want on the y-axis. More information on **ggformula** can be found here.

```
county_subset %>%
  gf_point(fed_spend ~ poverty)
```

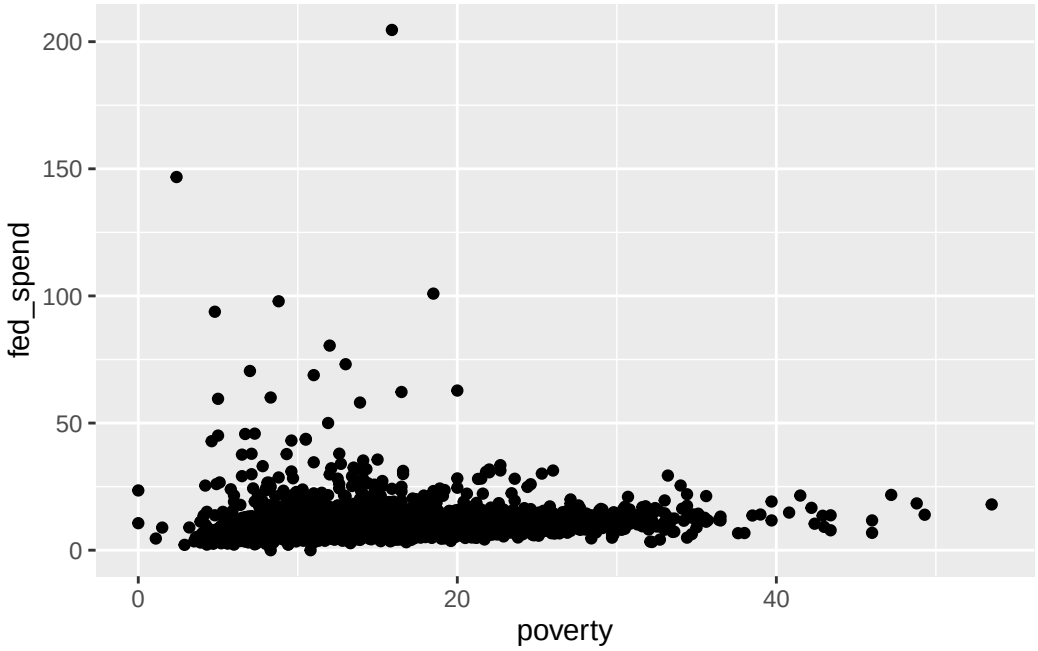


Figure 2.4: Scatterplot with **ggformula**.

Figure **??** is bad. There are poor axis labels, no title, dense clustering of points, and the y-axis is being driven by a couple of extreme points. We will need to clear this up. Again, try to read the code and use `help()` or `?` to determine the purpose of each command in Figure **??**.

```
county_subset %>%
  filter(fed_spend < 32) %>%
  gf_point(fed_spend ~ poverty,
           xlab = "Poverty Rate (Percent)",
           ylab = "Federal Spending Per Capita",
           title = "A scatterplot showing fed_spend against poverty",
           cex = 1, alpha = 0.2) %>%
  gf_theme(theme_classic())
```

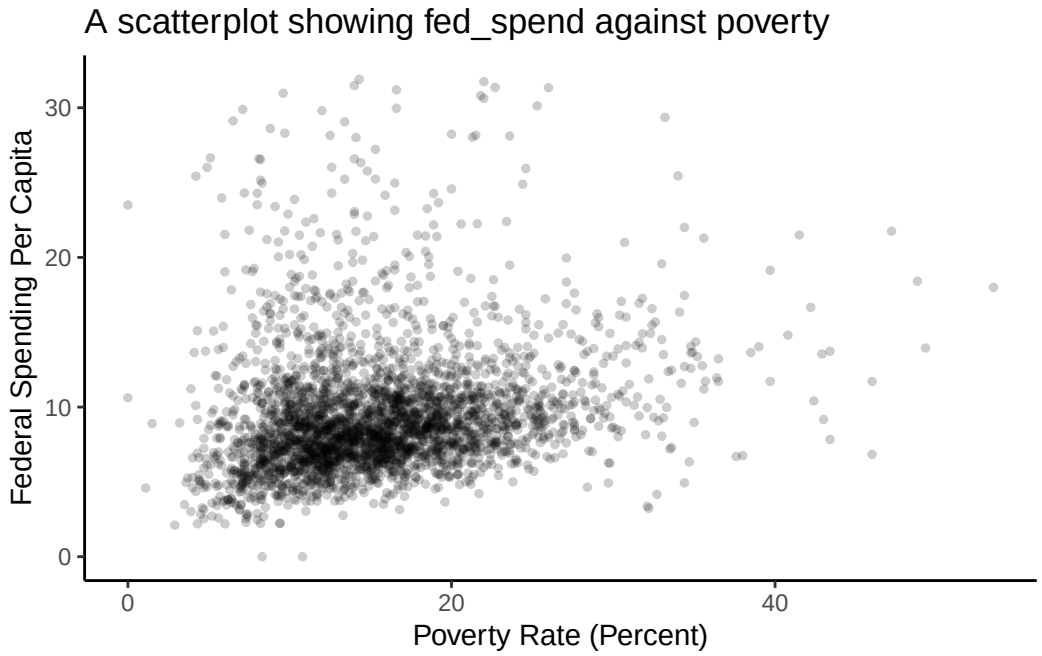


Figure 2.5: Better example of a scatterplot.

Exercise:
Create the scatterplot in Figure **??**.

3 Overview of Data Collection Principles

3.1 Objectives

1) Differentiate between various statistical terminologies such as *population, sample, anecdotal evidence, bias, simple random sample, systematic sample, representative sample, non-response bias, convenience sample, explanatory variable, response variable, observational study, cohort, experiment, randomized experiment,* and *placebo*, and construct examples to demonstrate their proper use in context.

2) Evaluate descriptions of research project to identify the population of interest, assess the generalizability of the study, determine the explanatory and response variables, classify the study as observational or experimental, and determine the type of sample used.

3) Design and justify sampling procedures for various research contexts, comparing the strengths and weaknesses of different sampling methods (simple random, systematic, convenience, etc.) and proposing improvements to minimize bias and enhance representativeness.

3.2 Overview of data collection principles

The first step in conducting research is to identify topics or questions that are to be investigated. A clearly laid out research question is helpful in identifying what subjects or cases should be studied and what variables are important. It is also important to consider *how* data are collected so that they are reliable and help achieve the research goals.

3.2.1 Populations and samples

Consider the following three research questions:

1. What is the average mercury content in swordfish in the Atlantic Ocean?

2. Over the last 5 years, what is the average time to complete a degree for Duke undergraduate students?

3. Does a new drug reduce the number of deaths in patients with severe heart disease?

Each research question refers to a target **population**, the entire collection of individuals about which we want information. In the first question, the target population is all swordfish in the Atlantic Ocean, and each fish represents a case. It is usually too expensive to collect data for every case in a population. Instead, a sample is taken. A **sample** represents a subset of the cases and is often a small fraction of the population. For instance, 60 swordfish (or some other number) in the population might be selected, and this sample data may be used to provide an estimate of the population average and answer the research question.

> **Exercise**:
> For the second and third questions above, identify the target population and what represents an individual case.[1]

3.2.2 Anecdotal evidence

Consider the following possible responses to the three research questions:

1. A man on the news got mercury poisoning from eating swordfish, so the average mercury concentration in swordfish must be dangerously high.
2. I met two students who took more than 7 years to graduate from Duke, so it must take longer to graduate at Duke than at many other colleges.
3. My friend's dad had a heart attack and died after they gave him a new heart disease drug, so the drug must not work.

Each conclusion is based on data. However, there are two problems. First, the data only represent one or two cases. Second, and more importantly, it is unclear whether these cases are actually representative of the population. Data collected in this haphazard fashion are called **anecdotal evidence**.

[1] 2) Notice that the second question is only relevant to students who complete their degree; the average cannot be computed using a student who never finished her degree. Thus, only Duke undergraduate students who have graduated in the last five years represent cases in the population under consideration. Each such student would represent an individual case. 3) A person with severe heart disease represents a case. The population includes all people with severe heart disease.

Figure 3.1: In February 2010, some media pundits cited one large snow storm as evidence against global warming. As comedian Jon Stewart pointed out, *It's one storm, in one region, of one country.*

Anecdotal evidence: Be careful of data collected haphazardly. Such evidence may be true and verifiable, but it may only represent extraordinary cases.

Anecdotal evidence typically is composed of unusual cases that we recall based on their striking characteristics. For instance, we are more likely to remember the two people we met who took 7 years to graduate than the six others who graduated in four years. Instead of looking at the most unusual cases, we should examine a sample of many cases that represent the population.

3.2.3 Sampling from a population

We might try to estimate the time to graduation for Duke undergraduates in the last 5 years by collecting a sample of students. All graduates in the last 5 years represent the *population*, and graduates who are selected for review are collectively called the *sample*. In general, we always seek to *randomly* select a sample from a population. The most basic type of random selection is equivalent to how raffles are conducted. For example, in selecting graduates, we could write each graduate's name on a raffle ticket and draw 100 tickets. The selected names would represent a random sample of 100 graduates. This is illustrated in Figure **??** .

Why pick a sample randomly? Why not just pick a sample by hand? Consider the following scenario.

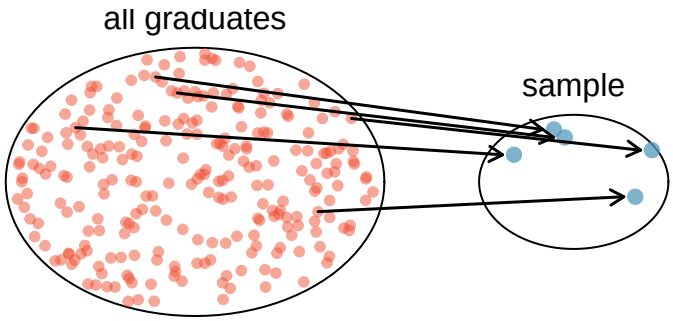


Figure 3.2: In this graphic, five graduates are randomly selected from the population to be included in the sample.

Example:
Suppose we ask a student who happens to be majoring in nutrition to select several graduates for the study. What kind of students do you think she might collect? Do you think her sample would be representative of all graduates? [2]

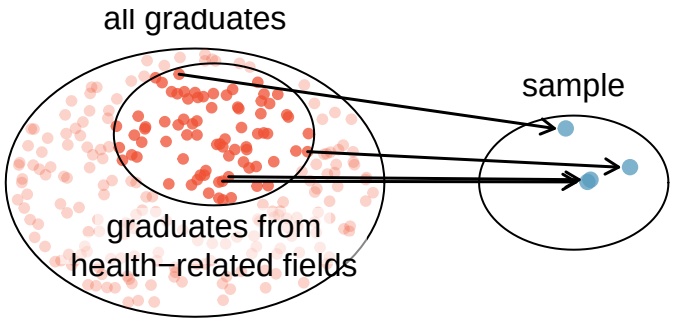


Figure 3.3: Instead of sampling from all graduates equally, a nutrition major might inadvertently pick graduates with health-related majors disproportionately often.

If someone was permitted to pick and choose exactly which graduates were included in the sample, it is entirely possible that the sample could be skewed to that person's interests, which may be entirely unintentional. This introduces **sampling bias** (see Figure **??**), where some individuals in the population are more likely to be sampled than others. Sampling randomly helps resolve this problem. The most basic random sample is called a **simple random sample**, which is equivalent to using a raffle to select cases. This means that each case in the population has an equal chance of being included and there is no implied connection between the cases in the sample.

Sometimes a simple random sample is difficult to implement and an alternative method is

[2] Perhaps she would pick a disproportionate number of graduates from health-related fields. Or perhaps her selection would be well-representative of the population. When selecting samples by hand, we run the risk of picking a *biased* sample, even if that bias is unintentional or difficult to discern.

helpful. One such substitute is a **systematic sample**, where one case is sampled after letting a fixed number of others, say 10 other cases, pass by. Since this approach uses a mechanism that is not easily subject to personal biases, it often yields a reasonably representative sample. This book will focus on simple random samples since the use of systematic samples is uncommon and requires additional considerations of the context.

The act of taking a simple random sample helps minimize bias. However, bias can crop up in other ways. Even when people are picked at random, e.g. for surveys, caution must be exercised if the non-response is high. For instance, if only 30% of the people randomly sampled for a survey actually respond, and it is unclear whether the respondents are **representative**[3] of the entire population, the survey might suffer from **non-response bias**[4].

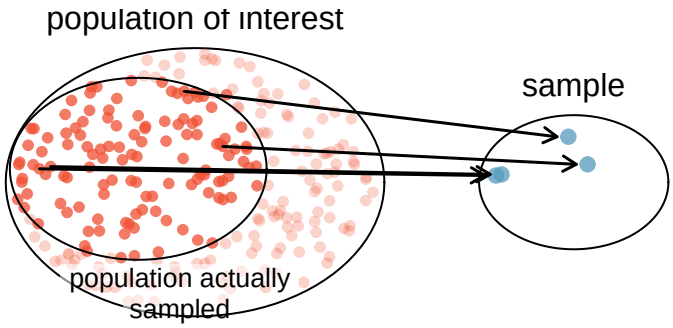


Figure 3.4: Due to the possibility of non-response, surveys studies may only reach a certain group within the population. It is difficult, and often impossible, to completely fix this problem.

Another common pitfall is a **convenience sample**, where individuals who are easily accessible are more likely to be included in the sample, see Figure **??**. For instance, if a political survey is done by stopping people walking in the Bronx, it will not represent all of New York City. It is often difficult to discern what sub-population a convenience sample represents.

> **Exercise**:
> We can easily access ratings for products, sellers, and companies through websites. These ratings are based only on those people who go out of their way to provide a rating. If 50% of online reviews for a product are negative, do you think this means that 50% of buyers are dissatisfied with the product?[5]

[3] A representative sample accurately reflects the characteristics of the population.

[4] Non-response bias is bias that can be introduced when subjects elect not to participate in a study. Often, the individuals that do participate are systematically different from the individuals who do not.

[5] Answers will vary. From our own anecdotal experiences, we believe people tend to rant more about products that fell below expectations than rave about those that perform as expected. For this reason, we suspect there is a negative bias in product ratings on sites like Amazon. However, since our experiences may not be representative, we also keep an open mind.

3.2.4 Explanatory and response variables

Consider the following question for the `county` data set:

Is federal spending, on average, higher or lower in counties with high rates of poverty?

If we suspect poverty might affect spending in a county, then poverty is the **explanatory** variable and federal spending is the **response** variable in the relationship.[6] If there are many variables, it may be possible to consider a number of them as explanatory variables.

> **Explanatory** and **response** variables
> To identify the explanatory variable in a pair of variables, identify which of the two variables is suspected as explaining or causing changes in the other. In data sets with more than two variables, it is possible to have multiple explanatory variables. The response variable is the outcome or result of interest.
>
> **Caution**: Association does not imply causation. Labeling variables as *explanatory* and *response* does not guarantee the relationship between the two is actually causal, even if there is an association identified between the two variables. We use these labels only to keep track of which variable we suspect affects the other. We also use this language to help in our use of `R` and the formula notation.

In some cases, there is no explanatory or response variable. Consider the following question:

If homeownership in a particular county is lower than the national average, will the percent of multi-unit structures in that county likely be above or below the national average?

It is difficult to decide which of these variables should be considered the explanatory and response variable; i.e. the direction is ambiguous, so no explanatory or response labels are suggested here.

3.2.5 Introducing observational studies and experiments

There are two primary types of data collection: observational studies and experiments.

Researchers perform an **observational study** when they collect data in a way that does not directly interfere with how the data arise. For instance, researchers may collect information via surveys, review medical or company records, or follow a **cohort**[7] of many similar individuals to study why certain diseases might develop. In each of these situations, researchers

[6] Sometimes the explanatory variable is called the **independent** variable and the response variable is called the **dependent** variable. However, this becomes confusing since a *pair* of variables might be independent or dependent, so be careful and consider the context when using or reading these words.

[7] A cohort is a group of individuals who are similar in some way.

merely observe what happens. In general, observational studies can provide evidence of a naturally occurring association between variables, but by themselves, they cannot show a causal connection.

When researchers want to investigate the possibility of a causal connection, they conduct an **experiment**, a study in which the explanatory variables are assigned rather than observed. For instance, we may suspect administering a drug will reduce mortality in heart attack patients over the following year. To check if there really is a causal connection between the explanatory variable and the response, researchers will collect a sample of individuals and split them into groups. The individuals in each group are *assigned* a treatment. When individuals are *randomly* assigned to a treatment group, and we are *comparing* at least two treatments, the experiment is called a **randomized comparative experiment**. For example, each heart attack patient in the drug trial could be randomly assigned, perhaps by flipping a coin, into one of two groups: the first group receives a **placebo** (fake treatment) and the second group receives the drug. The case study at the beginning of the book is another example of an experiment, though that study did not employ a placebo. Math 359 is a course on the design and analysis of experimental data, DOE, at USAFA. In the Air Force, these types of experiments are an important part of test and evaluation. Many Air Force analysts are expert practitioners of DOE. In this book though, we will minimize our discussion of DOE.

Association $\neq$ Causation
Again, association does not imply causation. In a data analysis, association does not imply causation, and causation can only be inferred from a randomized experiment. Although, a hot field is the analysis of causal relationships in observational data. This is important because consider cigarette smoking, how do we know it causes lung cancer? We only have observational data and clearly cannot do an experiment. We think analysts will be charged in the near future with using causal reasoning on observational data.

4 Studies

4.1 Objectives

1) Differentiate between various statistical terminologies such as *observational study, confounding variable, prospective study, retrospective study, simple random sampling, stratified sampling, strata, cluster sampling, multistage sampling, experiment, randomized experiment, control, replicate, blocking, treatment group, control group, blinded study, placebo, placebo effect,* and *double-blind*, and construct examples to demonstrate their proper use in context.
2) Evaluate study descriptions using appropriate terminology, and analyze the study design for potential biases or confounding variables.
3) Given a scenario, identify and assess flaws in reasoning, and propose robust study and sampling methodologies to address these flaws.

4.2 Observational studies, sampling strategies, and experiments

4.2.1 Observational studies

Generally, data in **observational studies** are collected only by monitoring what occurs, while experiments require the primary explanatory variable in a study be assigned for each subject by the researchers.

Making causal conclusions based on experiments is often reasonable. However, making the same causal conclusions based on observational data can be treacherous and is not recommended. Thus, observational studies are generally only sufficient to show associations.

> **Exercise**:
> Suppose an observational study tracked sunscreen use and skin cancer, and it was found that the more sunscreen someone used, the more likely the person was to have skin cancer. Does this mean sunscreen *causes* skin cancer?[1]

[1] No. See the paragraph following the exercise for an explanation.

Some previous research[2] tells us that using sunscreen actually reduces skin cancer risk, so maybe there is another variable that can explain this hypothetical association between sunscreen usage and skin cancer. One important piece of information that is absent is sun exposure. If someone is out in the sun all day, she is more likely to use sunscreen *and* more likely to get skin cancer. Exposure to the sun is unaccounted for in the simple investigation.

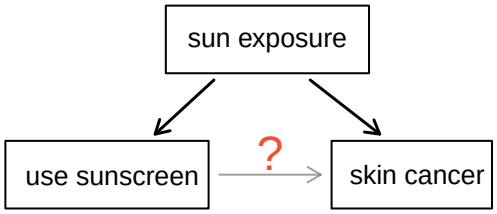


Figure 4.1: Sun exposure is a confounding variable because it is related to both response and explanatory variables.

Sun exposure is what is called a **confounding variable**,[3] which is a variable that is correlated with both the explanatory and response variables, see Figure **??**. While one method to justify making causal conclusions from observational studies is to exhaust the search for confounding variables, there is no guarantee that all confounding variables can be examined or measured.

Let's look at an example of confounding visually. Using the `SAT` data from the **mosaic** package let's look at expenditure per pupil versus SAT scores. Figure **??** is a plot of the data.

Exercise:
What conclusion do you reach from the plot in Figure **??**?[4]

The implication that spending less might give better results is not justified. Expenditures are confounded with the proportion of students who take the exam, and scores are higher in states where fewer students take the exam.

It is interesting to look at the original plot if we place the states into two groups depending on whether more or fewer than 40% of students take the SAT. Figure **??** is a plot of the data broken down into the 2 groups.

Once we account for the fraction of students taking the SAT, the relationship between expenditures and SAT scores changes.

In the same way, the `county` data set is an observational study with confounding variables, and its data cannot easily be used to make causal conclusions.

[2] http://www.sciencedirect.com/science/article/pii/S0140673698121682
http://archderm.ama-assn.org/cgi/content/abstract/122/5/537
Study with a similar scenario to that described here:
http://onlinelibrary.wiley.com/doi/10.1002/ijc.22745/full

[3] Also called a **lurking variable**, **confounding factor**, or a **confounder**.

[4] It appears that average SAT score declines as expenditures per student increases.

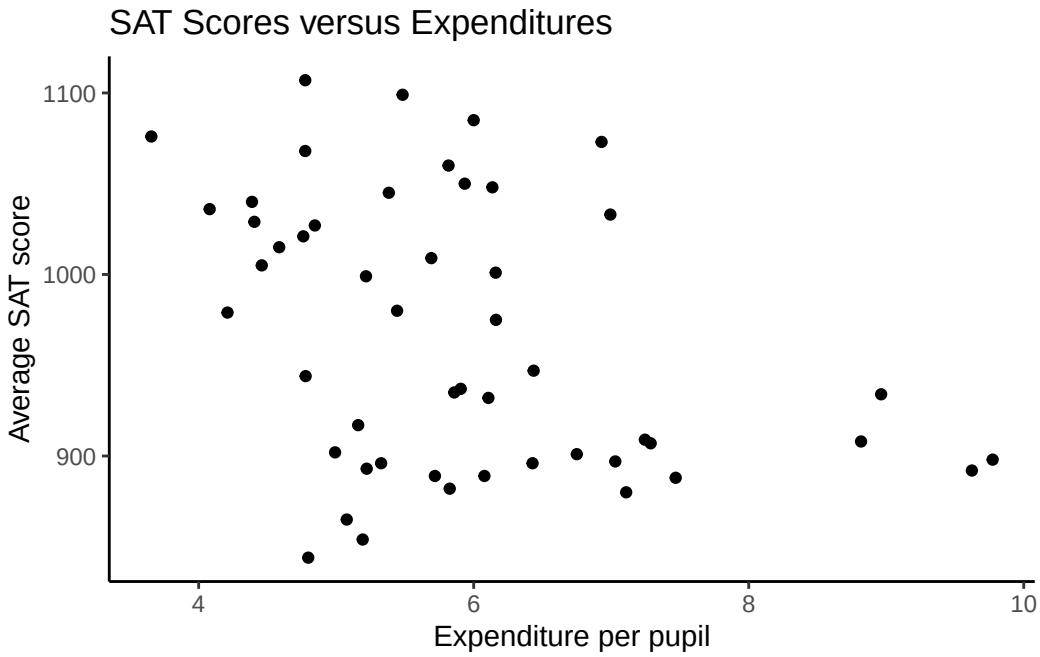


Figure 4.2: Average SAT score versus expenditure per pupil; reminder: each observation represents an individual state.

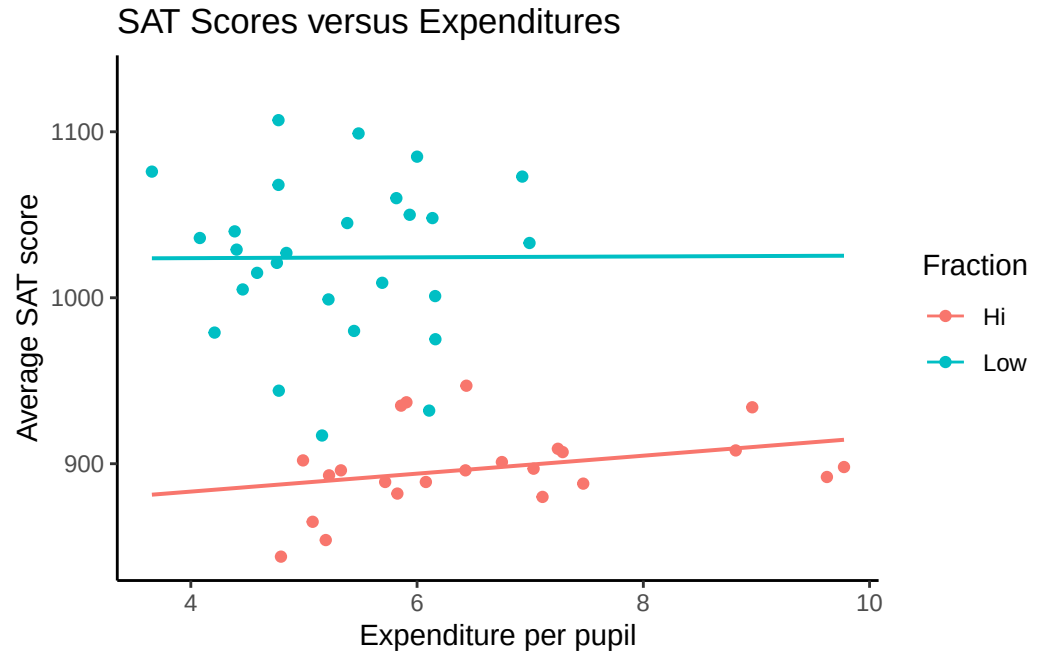


Figure 4.3: Average SAT score versus expenditure per pupil; broken down by level of participation.

Exercise:
Figure **??** shows a negative association between the homeownership rate and the percentage of multi-unit structures in a county. However, it is unreasonable to conclude that there is a causal relationship between the two variables. Suggest one or more other variables that might explain the relationship in the Figure **??**.[5]

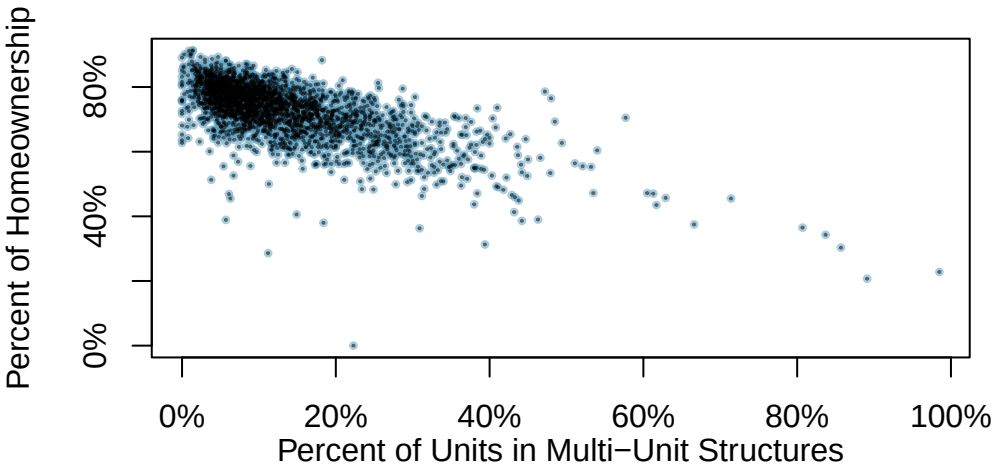


Figure 4.4: A scatterplot of the homeownership rate versus the percent of units that are in multi-unit structures for all 3,143 counties.

Observational studies come in two forms: prospective and retrospective studies. A **prospective study** identifies individuals and collects information as events unfold. For instance, medical researchers may identify and follow a group of similar individuals over many years to assess the possible influences of behavior on cancer risk. One example of such a study is The Nurses Health Study, started in 1976 and expanded in 1989.[6] This prospective study recruits registered nurses and then collects data from them using questionnaires.

Retrospective studies collect data after events have taken place; e.g. researchers may review past events in medical records. Some data sets, such as `county`, may contain both prospectively- and retrospectively-collected variables. Local governments prospectively collect some variables as events unfolded (e.g. retail sales) while the federal government retrospectively collected others during the 2010 census (e.g. county population).

[5] Answers will vary. Population density may be important. If a county is very dense, then a larger fraction of residents may live in multi-unit structures. Additionally, the high density may contribute to increases in property value, making homeownership infeasible for many residents.

[6] http://www.channing.harvard.edu/nhs/

4.2.2 Three sampling methods

Almost all statistical methods are based on the notion of implied randomness. If observational data are not collected in a random framework from a population, results from these statistical methods are not reliable. Here we consider three random sampling techniques: simple, stratified, and cluster sampling. Figure **??**, Figure **??**, and Figure **??** provide a graphical representation of these techniques.

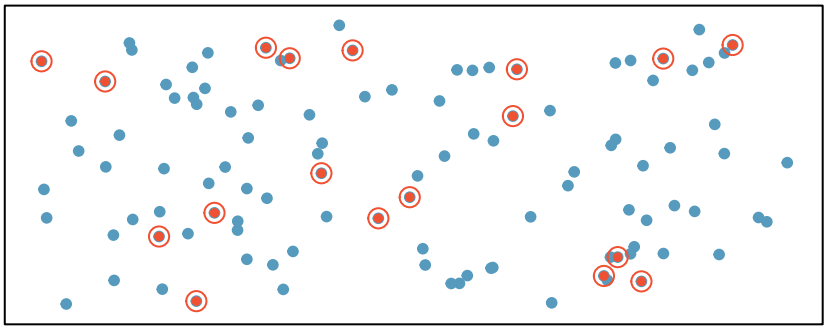


Figure 4.5: Examples of simple random sampling. In this figure, simple random sampling was used to randomly select the 18 cases.

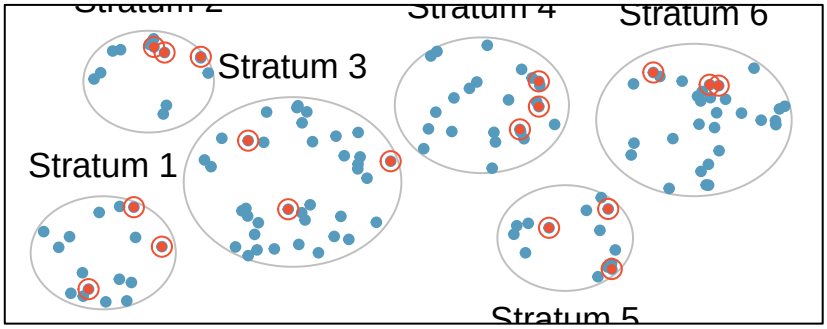


Figure 4.6: In this figure, stratified sampling was used: cases were grouped into strata, and then simple random sampling was employed within each stratum.

Simple random sampling is probably the most intuitive form of random sampling, in which each individual in the population has an equal chance of being chosen. Consider the salaries of Major League Baseball (MLB) players, where each player is a member of one of the league's 30 teams. To take a simple random sample of 120 baseball players and their salaries from the 2010 season, we could write the names of that season's 828 players onto slips of paper, drop the slips into a bucket, shake the bucket around until we are sure the names are all mixed up, then draw out slips until we have the sample of 120 players. In general, a sample is referred to as "simple random" if each case in the population has an equal chance of being included in the final sample *and* knowing that a case is included in a sample does not provide useful information about which other cases are included or not.

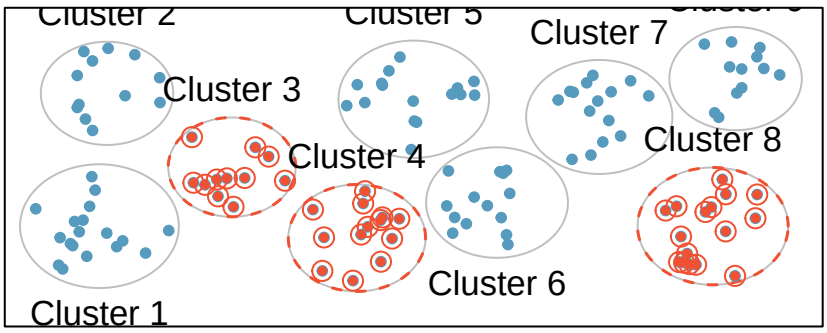


Figure 4.7: In this figure, cluster sampling was used, where data were binned into nine clusters, and three of the clusters were randomly selected.

Stratified sampling is a divide-and-conquer sampling strategy. The population is divided into groups called **strata**. The strata are chosen so that similar cases are grouped together, then a second sampling method, usually simple random sampling, is employed within each stratum. In the baseball salary example, the teams could represent the strata; some teams have a lot more money (we're looking at you, Yankees). Then we might randomly sample 4 players from each team for a total of 120 players.

Stratified sampling is especially useful when the cases in each stratum are very similar with respect to the outcome of interest. The downside is that analyzing data from a stratified sample is a more complex task than analyzing data from a simple random sample. The analysis methods introduced in this book would need to be extended to analyze data collected using stratified sampling.

Example:
Why would it be good for cases within each stratum to be very similar?[7]

In **cluster sampling**, we group observations into clusters, then randomly sample some of the clusters. Sometimes cluster sampling can be a more economical technique than the alternatives. Also, unlike stratified sampling, cluster sampling is most helpful when there is a lot of case-to-case variability within a cluster but the clusters themselves don't look very different from one another. For example, if neighborhoods represented clusters, then this sampling method works best when the neighborhoods are very diverse. A downside of cluster sampling is that more advanced analysis techniques are typically required, though the methods in this book can be extended to handle such data.

Example:
Suppose we are interested in estimating the malaria rate in a densely tropical portion of rural Indonesia. We learn that there are 30 villages in that part of the

[7]We might get a more stable estimate for the subpopulation in a stratum if the cases are very similar. These improved estimates for each subpopulation will help us build a reliable estimate for the full population.

Indonesian jungle, each more or less similar to the next. What sampling method should be employed?[8]

Another technique called **multistage sampling** is similar to cluster sampling, except that we take a simple random sample within each selected cluster. For instance, if we sampled neighborhoods using cluster sampling, we would next sample a subset of homes within each selected neighborhood if we were using multistage sampling.

4.2.3 Experiments

Studies where the researchers assign treatments to cases are called **experiments**. When this assignment includes randomization, e.g. using a coin flip to decide which treatment a patient receives, it is called a **randomized experiment**. Randomized experiments are fundamentally important when trying to show a causal connection between two variables.

4.2.3.1 Principles of experimental design

Randomized experiments are generally built on four principles.

1. **Controlling**. Researchers assign treatments to cases, and they do their best to **control** any other differences in the groups. For example, when patients take a drug in pill form, some patients take the pill with only a sip of water while others may have it with an entire glass of water. To control for the effect of water consumption, a doctor may ask all patients to drink a 12 ounce glass of water with the pill.
2. **Randomization**. Researchers randomize patients into treatment groups to account for variables that cannot be controlled. For example, some patients may be more susceptible to a disease than others due to their dietary habits. Randomizing patients into the treatment or control group helps even out such differences, and it also prevents accidental bias from entering the study.
3. **Replication**. The more cases researchers observe, the more accurately they can estimate the effect of the explanatory variable on the response. In a single study, we **replicate** by collecting a sufficiently large sample. Additionally, a group of scientists may replicate an entire study to verify an earlier finding. You replicate to the level of variability you want to estimate. For example, in flight test, we can run the same flight conditions again to get a replicate; however, if the same plane and pilot are being used, the replicate is not getting the pilot-to-pilot or the plane-to-plane variability.

[8] A simple random sample would likely draw individuals from all 30 villages, which could make data collection extremely expensive. Stratified sampling would be a challenge since it is unclear how we would build strata of similar individuals. However, cluster sampling seems like a very good idea. We might randomly select a small number of villages. This would probably reduce our data collection costs substantially in comparison to a simple random sample and would still give us helpful information.

4. **Blocking**. Researchers sometimes know or suspect that variables, other than the treatment, influence the response. Under these circumstances, they may first group individuals based on this variable and then randomize cases within each block, or group, to the treatments. This strategy is often referred to as **blocking**. For instance, if we are looking at the effect of a drug on heart attacks, we might first split patients into low-risk and high-risk blocks, then randomly assign half the patients from each block to the control group and the other half to the treatment group, as shown in Figure **??**. This strategy ensures each treatment group has an equal number of low-risk and high-risk patients.

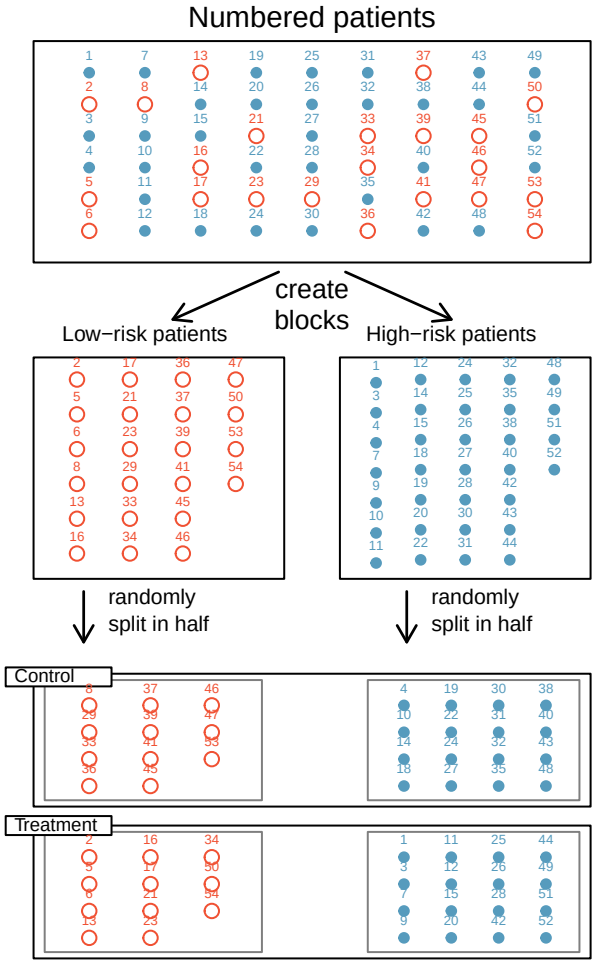


Figure 4.8: Blocking using a variable depicting patient risk. Patients are first divided into low-risk and high-risk blocks, then each block is evenly divided into the treatment groups using randomization. This strategy ensures an equal representation of patients in each treatment group from both the low-risk and high-risk categories.

It is important to incorporate the first three experimental design principles into any study, and this chapter describes methods for analyzing data from such experiments. Blocking is a

slightly more advanced technique, and statistical methods in this chapter may be extended to analyze data collected using blocking. Math 359 is an entire course at USAFA devoted to the design and analysis of experiments.

4.2.3.2 Reducing bias in human experiments

Randomized experiments are the gold standard for data collection, but they do not ensure an unbiased perspective into the cause and effect relationships in all cases. Human studies are perfect examples where bias can unintentionally arise. Here we reconsider a study where a new drug was used to treat heart attack patients.[9] In particular, researchers wanted to know if the drug reduced deaths in patients.

These researchers designed a randomized experiment because they wanted to draw causal conclusions about the drug's effect. Study volunteers[10] were randomly placed into two study groups. One group, the **treatment group**, received the experimental treatment of interest (the new drug to treat heart attack patients). The other group, called the **control group**, did not receive any drug treatment. The comparison between the treatment and control groups allows researchers to determine whether the treatment really has an effect.

Put yourself in the place of a person in the study. If you are in the treatment group, you are given a fancy new drug that you anticipate will help you. On the other hand, a person in the other group doesn't receive the drug and sits idly, hoping her participation doesn't increase her risk of death. These perspectives suggest there are actually two effects: the one of interest is the effectiveness of the drug, and the second is an emotional effect that is difficult to quantify.

Researchers aren't usually interested in the emotional effect, which might bias the study. To circumvent this problem, researchers do not want patients to know which group they are in. When researchers keep the patients uninformed about their treatment, the study is said to be **blind**. But there is one problem: if a patient doesn't receive a treatment, she will know she is in the control group. The solution to this problem is to give fake treatments to patients in the control group. A fake treatment is called a **placebo**, and an effective placebo is the key to making a study truly blind. A classic example of a placebo is a sugar pill that is made to look like the actual treatment pill. Often times, a placebo results in a slight but real improvement in patients. This effect has been dubbed the **placebo effect**.

The patients are not the only ones who should be blinded: doctors and researchers can accidentally bias a study. When a doctor knows a patient has been given the real treatment, she might inadvertently give that patient more attention or care than a patient that she knows is on the placebo. To guard against this bias, which again has been found to have a measurable effect in some instances, most modern studies employ a **double-blind** setup where doctors or

[9] Anturane Reinfarction Trial Research Group. 1980. Sulfinpyrazone in the prevention of sudden death after myocardial infarction. New England Journal of Medicine 302(5):250-256.

[10] Human subjects are often called **patients**, **volunteers**, or **study participants**.

researchers who interact with patients are, just like the patients, unaware of who is or is not receiving the treatment.[11]

Exercise:
Look back to the stent study in the first chapter where researchers were testing whether stents were effective at reducing strokes in at-risk patients. Is this an experiment? Was the study blinded? Was it double-blinded?[12]

[11] There are always some researchers in the study who do know which patients are receiving which treatment. However, they do not interact with the study's patients and do not tell the blinded health care professionals who is receiving which treatment.

[12] The researchers assigned the patients into their treatment groups, so this study was an experiment. However, the patients could distinguish what treatment they received, so this study was not blind. The study could not be double-blind since it was not blind.

5 Numerical Data

5.1 Objectives

1) Differentiate between various statistical terminologies such as *scatterplot, mean, distribution, point estimate, weighted mean, histogram, data density, right skewed, left skewed, symmetric, mode, unimodal, bimodal, multimodal, density plot, variance, standard deviation, box plot, median, interquartile range, first quartile, third quartile, whiskers, outlier, robust estimate*, and *transformation*, and construct examples to demonstrate their proper use in context.

2) Using `R`, generate and interpret summary statistics for numerical variables.

3) Create and evaluate graphical summaries of numerical variables using `R`, choosing the most appropriate types of plots for different data characteristics and research questions.

4) Synthesize numerical and graphical summaries to provide interpretations and explanations of a data set.

5.2 Numerical Data

This chapter introduces techniques for exploring and summarizing numerical variables. The `email50` and `mlb` data sets from the **openintro** package and a subset of `county_complete` from the **usdata** package provide rich opportunities for examples. Recall that outcomes of numerical variables are numbers on which it is reasonable to perform basic arithmetic operations. For example, the `pop2010` variable, which represents the population of counties in 2010, is numerical since we can sensibly discuss the difference or ratio of the populations in two counties. On the other hand, area codes and zip codes are not numerical.

5.2.1 Scatterplots for paired data

A **scatterplot** provides a case-by-case view of data for two numerical variables. In Figure **??**, we again present a scatterplot used to examine how federal spending and poverty are related in the `county` data set.

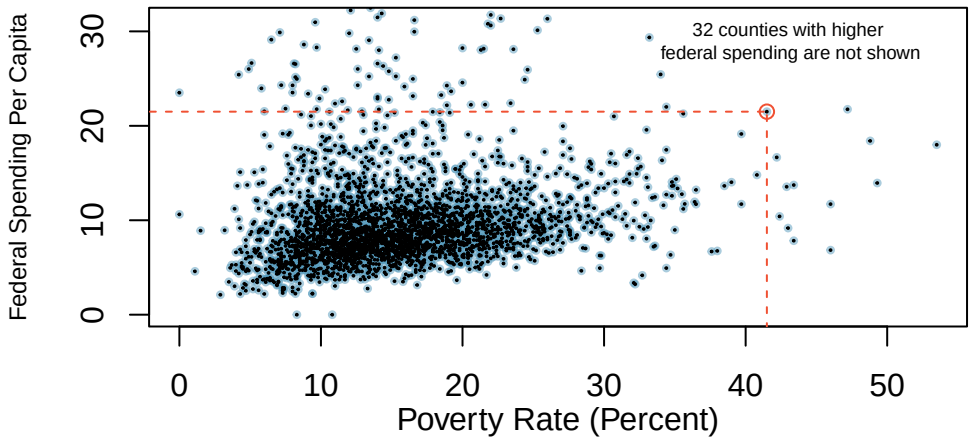


Figure 5.1: A scatterplot showing fed_spend against poverty. Owsley County of Kentucky, with a poverty rate of 41.5% and federal spending of $21.50 per capita, is highlighted.

Another scatterplot is shown in Figure **??**, comparing the number of `line_breaks` and number of characters, `num_char`, in emails for the `email50` data set. In any scatterplot, each point represents a single case. Since there are 50 cases in `email50`, there are 50 points in Figure **??**.

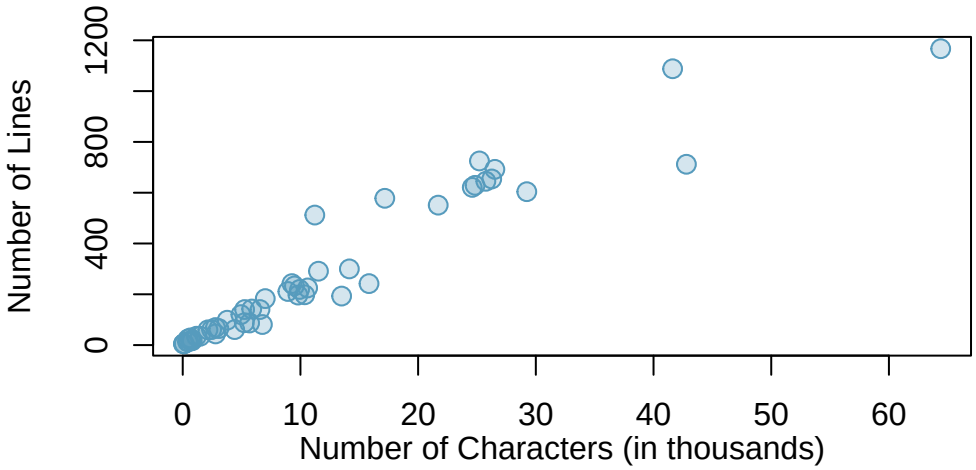


Figure 5.2: A scatterplot of `line_breaks` versus `num_char` for the `email50` data.

To put the number of characters in perspective, this paragraph in the text has 357 characters. Looking at Figure **??**, it seems that some emails are incredibly long! Upon further investigation, we would actually find that most of the long emails use the HTML format, which means most of the characters in those emails are used to format the email rather than provide text.

Exercise:

What do scatterplots reveal about the data, and how might they be useful?[1]

Example:
Consider a new data set of 54 cars with two variables: vehicle price and weight.[2] A scatterplot of vehicle price versus weight is shown in Figure **??**. What can be said about the relationship between these variables?

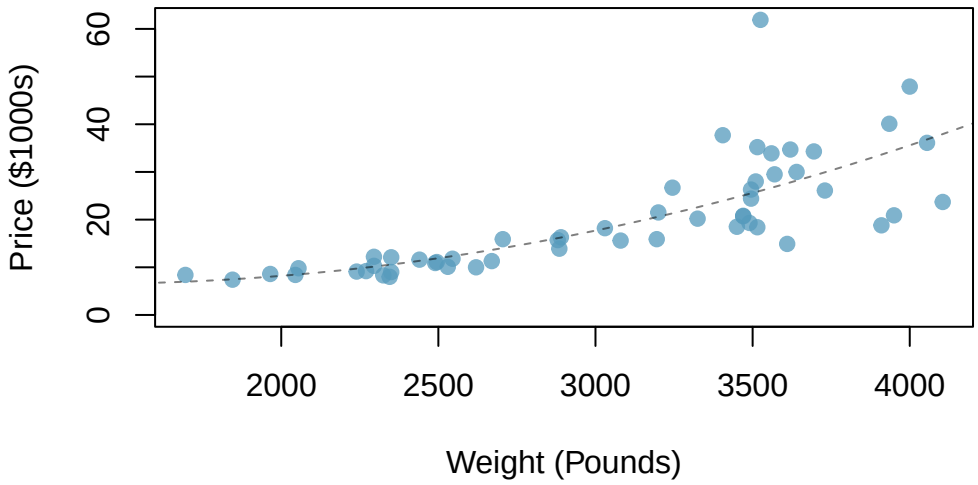


Figure 5.3: A scatterplot of *price* versus *weight* for 54 cars.

The relationship is evidently nonlinear, as highlighted by the dashed line. This is different from previous scatterplots we've seen which show relationships that are very linear.

Exercise:
Describe two variables that would have a horseshoe-shaped association in a scatterplot.[3]

5.2.2 The mean

The **mean**, sometimes called the average, is a common way to measure the center of a **distribution**[4] of data. To find the mean number of characters in the 50 emails, we add up all

[1] Answers may vary. Scatterplots are helpful in quickly spotting associations between variables, whether those associations represent simple or more complex relationships.
[2] Subset of data from http://www.amstat.org/publications/jse/v1n1/datasets.lock.html
[3] Consider the case where your vertical axis represents something "good" and your horizontal axis represents something that is only good in moderation. Health and water consumption fit this description since water becomes toxic when consumed in excessive quantities.
[4] The distribution of a variable is essentially the collection of all values of the variable in the data set. It tells us what values the variable takes on and how often. In the `email50` data set, we used a dotplot to view the distribution of `num_char`.

the character counts and divide by the number of emails. For computational convenience, the number of characters is listed in the thousands and rounded to the first decimal.

$$\bar{x} = \frac{21.7 + 7.0 + \cdots + 15.8}{50} = 11.6$$

The sample mean is often labeled $\bar{x}$. There is a bar over the letter, and the letter x is being used as a generic placeholder for the variable of interest, `num_char`.

Mean
The sample mean of a numerical variable is the sum of all of the observations divided by the number of observations, Equation 1.

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{1}$$

where $x_1, x_2, \ldots, x_n$ represent the n observed values.

Exercise:
Examine the two equations above. What does x_1 correspond to? And x_2? Can you infer a general meaning to what x_i might represent?[5]

Exercise:
What was n in this sample of emails?[6]

The `email50` data set is a sample from a larger population of emails that were received in January and March. We could compute a mean for this population in the same way as the sample mean. However, there is a difference in notation: the population mean has a special label: μ. The symbol μ is the Greek letter *mu* and represents the average of all observations in the population. Sometimes a subscript, such as $_x$, is used to represent which variable the population mean refers to, e.g. μ_x.

Example: The average number of characters across all emails can be estimated using the sample data. Based on the sample of 50 emails, what would be a reasonable estimate of μ_x, the mean number of characters in all emails in the `email` data set? (Recall that `email50` is a sample from `email`.)

[5] x_1 corresponds to the number of characters in the first email in the sample (21.7, in thousands), x_2 to the number of characters in the second email (7.0, in thousands), and x_i corresponds to the number of characters in the i^{th} email in the data set.
[6] The sample size, $n = 50$.

The sample mean, 11.6, may provide a reasonable estimate of μ_x. While this number will not be perfect, it provides a **point estimate**, a single plausible value, of the population mean. Later in the text, we will develop tools to characterize the accuracy of point estimates, and we will find that point estimates based on larger samples tend to be more accurate than those based on smaller samples.

> *Example*:
> We might like to compute the average income per person in the US. To do so, we might first think to take the mean of the per capita incomes from the 3,143 counties in the `county` data set. What would be a better approach?

The `county` data set is special in that each county actually represents many individual people. If we were to simply average across the `income` variable, we would be treating counties with 5,000 and 5,000,000 residents equally in the calculations. Instead, we should compute the total income for each county, add up all the counties' totals, and then divide by the number of people in all the counties. If we completed these steps with the `county` data, we would find that the per capita income for the US is $27,348.43. Had we computed the *simple* mean of per capita income across counties, the result would have been just $22,504.70!

This previous example used what is called a **weighted mean**[7], which will be a key topic in the probability section. As a look ahead, the probability mass function gives the population proportions of each county's mean value, and thus, to find the population mean μ, we will use a weighted mean.

5.2.3 Histograms and shape

Rather than showing the exact value of each observation for a single variable, think of the value as belonging to a *bin*. For example, in the `email50` data set, we create a table of counts for the number of cases with character counts between 0 and 5,000, then the number of cases between 5,000 and 10,000, and so on. Observations that fall on the boundary of a bin (e.g. 5,000) are allocated to the lower bin. This tabulation is shown below.

```
  (0,5]  (5,10] (10,15] (15,20] (20,25] (25,30] (30,35] (35,40] (40,45] (45,50]
     19      12       6       2       3       5       0       0       2       0
(50,55] (55,60] (60,65]
      0       0       1
```

These binned counts are plotted as bars in Figure **??** in what is called a **histogram**[8].

[7] A weighted mean is an average in which some observations contribute more "weight" than others. In the `county` data set, we "weighted" the income for each county by dividing income by the county population.

[8] A histogram displays the distribution of a quantitative variable. It shows binned counts, the number of observations in a bin, or range of values.

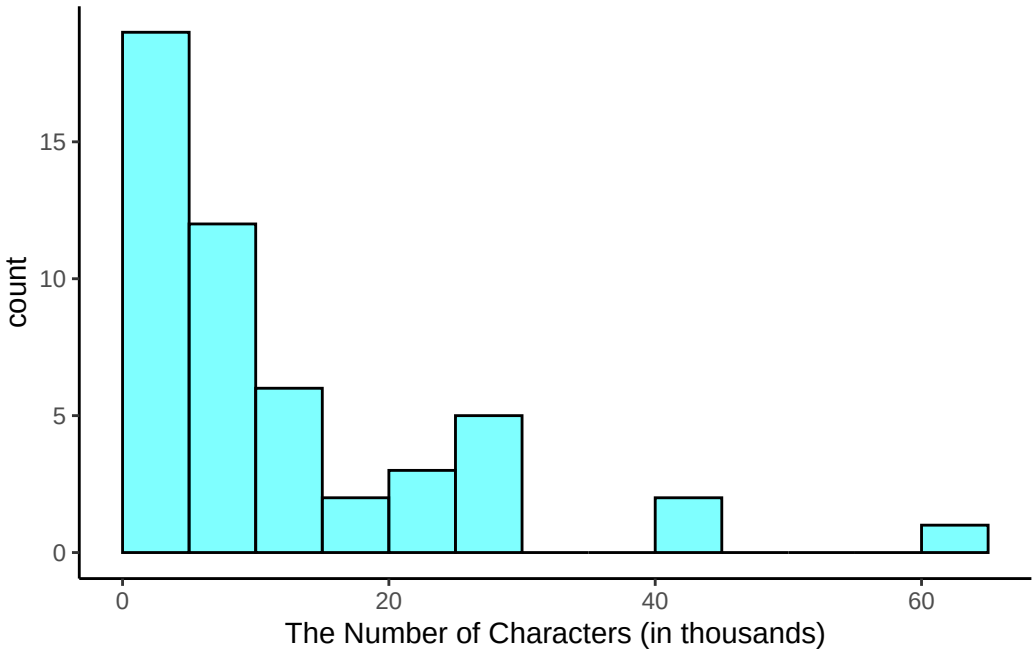


Figure 5.4: A histogram of `num_char`. This distribution is very strongly skewed to the right.

Histograms provide a view of the **data density**. Higher bars represent where the data are relatively more dense. For instance, there are many more emails between 0 and 10,000 characters than emails between 10,000 and 20,000 characters in the data set. The bars make it easy to see how the density of the data changes relative to the number of characters.

Histograms are especially convenient for describing the shape of the data distribution. Figure **??** shows that most emails have a relatively small number of characters, while fewer emails have a very large number of characters. When data trail off to the right in this way and have a longer right **tail**, the shape is said to be **right skewed**.[9]

Data sets with the reverse characteristic – a long, thin tail to the left – are said to be **left skewed**. We also say that such a distribution has a long left tail. Data sets that show roughly equal trailing off in both directions are called **symmetric**.

Long tails to identify skew
When data trail off in one direction, the distribution has a **long tail**. If a distribution has a long left tail, it is left skewed. If a distribution has a long right tail, it is right skewed.

[9]Other ways to describe data that are skewed to the right: **skewed to the right**, **skewed to the high end**, or **skewed to the positive end**.

5.2.3.1 Making our own histogram

Let's take some time to make a simple histogram. We will use the **ggformula** package, which is a wrapper for the **ggplot2** package.

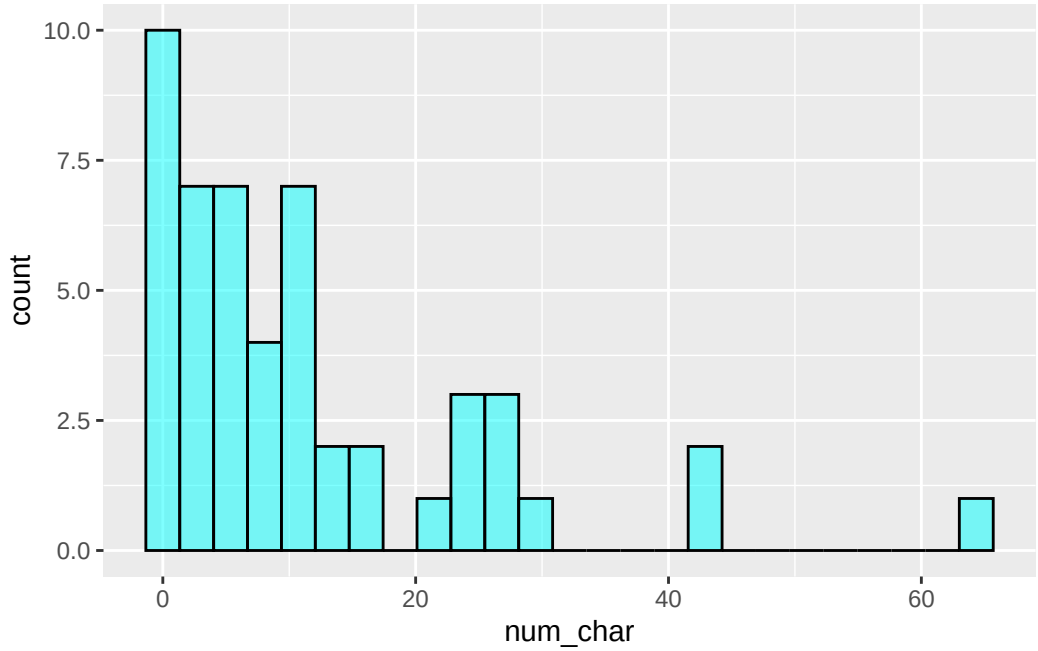
Here are two questions:
What do we want `R` to do? and
What must we give `R` for it to do this?

We want `R` to make a histogram. In `ggformula`, the plots have the form `gf_plottype` so we will use the `gf_histogram()`. To find options and more information about the function, type:

```
?gf_histogram
```

To start, we just have to give the formulas and data to `R`.

```
gf_histogram(~num_char, data = email50, color = "black", fill = "cyan")
```



Exercise:
Look at the help menu for `gf_histogram` and change the x-axis label, change the bin width to 5, and have the left bin start at 0.

Here is the code for the exercise:

```
email50 %>%
   gf_histogram(~num_char, binwidth = 5,boundary = 0,
   xlab = "The Number of Characters (in thousands)",
   color = "black", fill = "cyan") %>%
   gf_theme(theme_classic())
```

In addition to looking at whether a distribution is skewed or symmetric, histograms can be used to identify modes. A **mode** is represented by a prominent peak in the distribution.[10] There is only one prominent peak in the histogram of `num_char`.

Figure **??** shows histograms that have one, two, or three prominent peaks. Such distributions are called **unimodal**, **bimodal**, and **multimodal**, respectively. Any distribution with more than 2 prominent peaks is called multimodal. Notice that there was one prominent peak in the unimodal distribution with a second less prominent peak that was not counted since the separation between the two peaks is relatively small, and it only differs from its neighboring bins by a few observations.

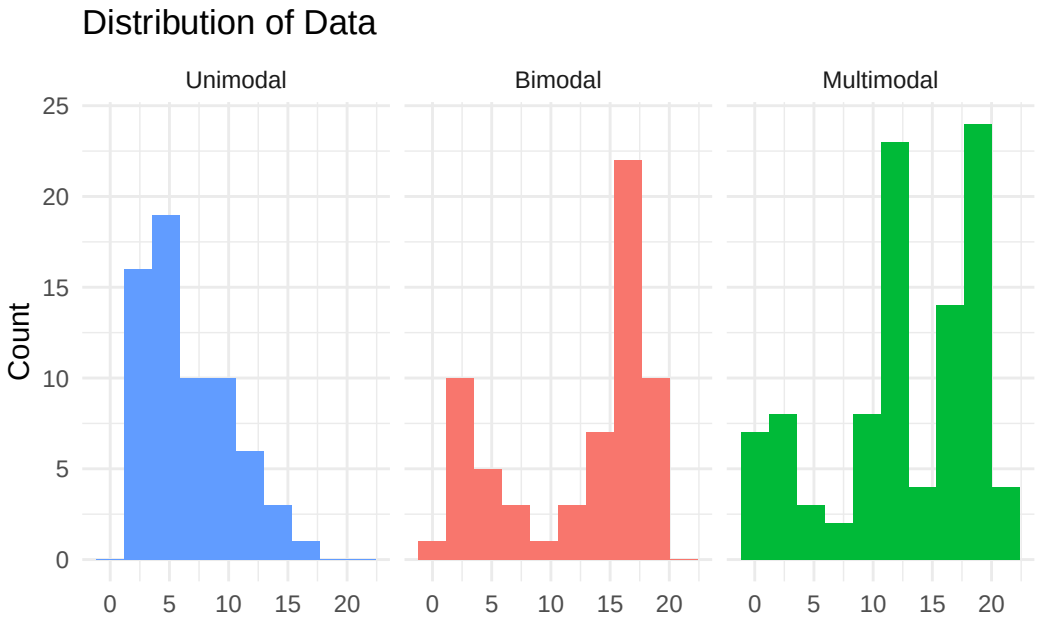


Figure 5.5: Histograms that demonstrate unimodal, bimodal, and multimodal data.

Exercise:
Height measurements of young students and adult teachers at a K-3 elementary

[10] Another definition of mode, which is not typically used in statistics, is the value with the most occurrences. It is common to have *no* observations with the same value in a data set, which makes this other definition useless for many real data sets.

school were taken. How many modes would you anticipate in this height data set?[11]

Looking for modes
Looking for modes isn't about finding a clear and correct answer about the number of modes in a distribution, which is why **prominent** is not rigorously defined in these notes. The important part of this examination is to better understand your data and how it might be structured.

5.2.4 Density plots

Another useful plotting method uses **density plots** to visualize numerical data. A histogram bins data but is highly dependent on the number and boundary of the bins. A density plot also estimates the distribution of a numerical variable but does this by estimating the density of data points in a small window around each data point. The overall curve is the sum of this small density estimate. A density plot can be thought of as a smooth version of the histogram. Several options go into a density estimate, such as the width of the window and type of smoothing function. These ideas are beyond the scope here and we will just use the default options. Figure **??** shows the same distribution of the number of characters but in a smoother way than a histogram.

Exercise:
Compare and contrast the histogram in Figure **??** and the density plot in Figure **??**. What can you see in the histogram you can't see in the density plot?

5.2.5 Variance and standard deviation

The mean is used to describe the center of a data set, but the *variability* in the data is also important. Here, we introduce two measures of variability: the **variance** and the **standard deviation**. Both of these are very useful in data analysis, even though the formulas are a bit tedious to calculate by hand. The standard deviation is the easier of the two to conceptually understand; it roughly describes how far away the typical observation is from the mean. Equation 2 is the equation for sample variance. We will demonstrate it with data so that the notation is easier to understand.

[11] There might be two height groups visible in the data set: one for the students and one for the adults. That is, the data are probably bimodal. But it could be multimodal because within each group we may be able to see a difference in males and females.

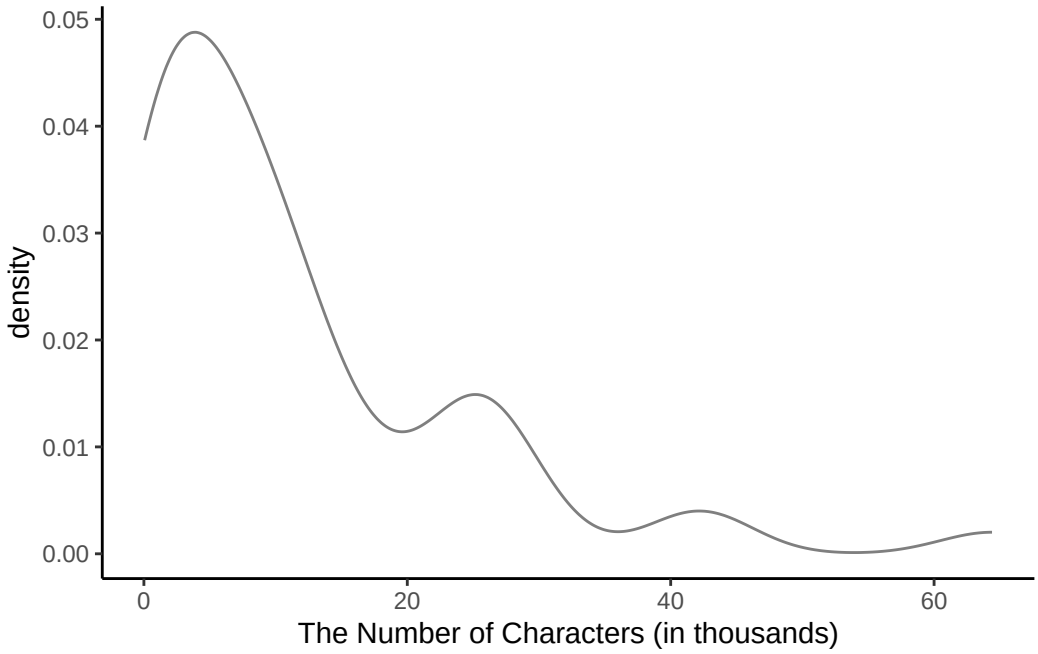


Figure 5.6: A histogram of `num_char`. This distribution is very strongly skewed to the right.

$$s^2 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \tag{5.1}$$

$$= \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n-1} \tag{2}$$

where $x_1, x_2, \ldots, x_n$ represent the n observed values.

We call the distance of an observation from its mean the **deviation**. Below are the deviations for the 1^{st}, 2^{nd}, 3^{rd}, and 50^{th} observations of the `num_char` variable. For computational convenience, the number of characters is listed in the thousands and rounded to the first decimal.

$$\begin{aligned} x_1 - \bar{x} &= 21.7 - 11.6 = 10.1 \\ x_2 - \bar{x} &= 7.0 - 11.6 = -4.6 \\ x_3 - \bar{x} &= 0.6 - 11.6 = -11.0 \\ &\vdots \\ x_{50} - \bar{x} &= 15.8 - 11.6 = 4.2 \end{aligned}$$

If we square these deviations and then take an average, the result is equal to the **sample variance**, denoted by s^2:

$$
\begin{aligned}
s^2 &= \frac{10.1^2 + (-4.6)^2 + (-11.0)^2 + \cdots + 4.2^2}{50 - 1} \\
&= \frac{102.01 + 21.16 + 121.00 + \cdots + 17.64}{49} \\
&= 172.44
\end{aligned}
$$

We divide by $n - 1$, rather than dividing by n, when computing the variance; you need not worry about this mathematical nuance yet. Notice that squaring the deviations does two things. First, it makes large values much larger, seen by comparing 10.1^2, $(-4.6)^2$, $(-11.0)^2$, and 4.2^2. Second, it gets rid of any negative signs.

The sample **standard deviation**, s, is the square root of the variance:

$$s = \sqrt{172.44} = 13.13$$

The sample standard deviation of the number of characters in an email is 13.13 thousand. A subscript of $_x$ may be added to the variance and standard deviation, i.e. s_x^2 and s_x, as a reminder that these are the variance and standard deviation of the observations represented by x_1, x_2, ..., x_n. The $_x$ subscript is usually omitted when it is clear which data the variance or standard deviation is referencing.

Variance and standard deviation
The variance is roughly the average squared distance from the mean. The standard deviation is the square root of the variance and describes how close the data are to the mean.

Formulas and methods used to compute the variance and standard deviation for a population are similar to those used for a sample.[12] However, like the mean, the population values have special symbols: σ^2 for the variance and σ for the standard deviation. The symbol σ is the Greek letter *sigma*.

Tip: standard deviation describes variability
Focus on the conceptual meaning of the standard deviation as a descriptor of variability rather than the formulas. Usually 70% of the data will be within one standard deviation of the mean and about 95% will be within two standard deviations. However, as we have seen, these percentages are not strict rules.

Exercise:
Earlier, the concept of shape of a distribution was introduced. A good description of the shape of a distribution should include modality and whether the distribution

[12] The only difference is that the population variance has a division by n instead of $n - 1$.

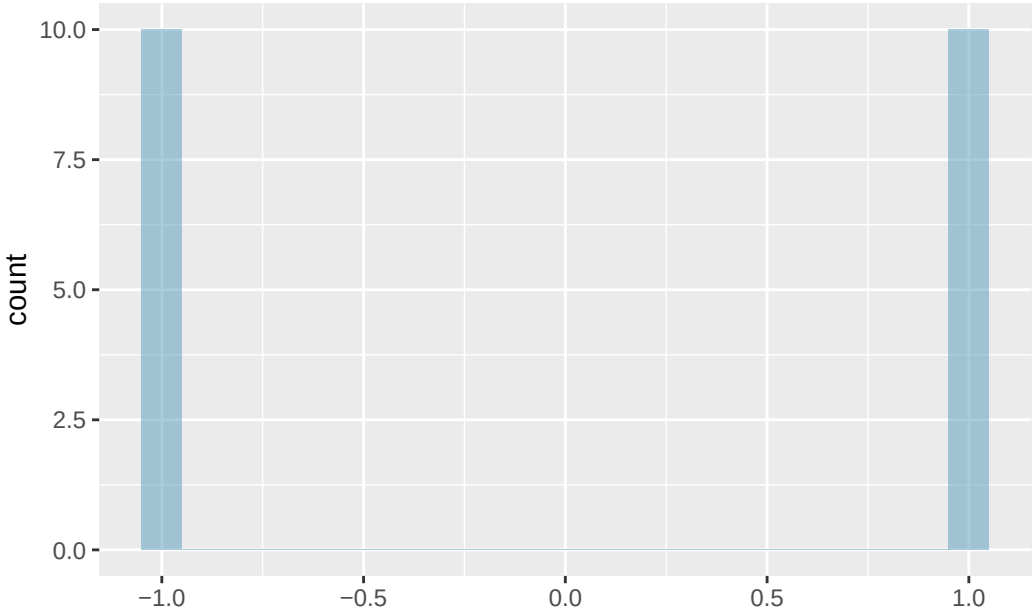


Figure 5.7: The first of three very different population distributions with the same mean, 0, and standard deviation, 1.

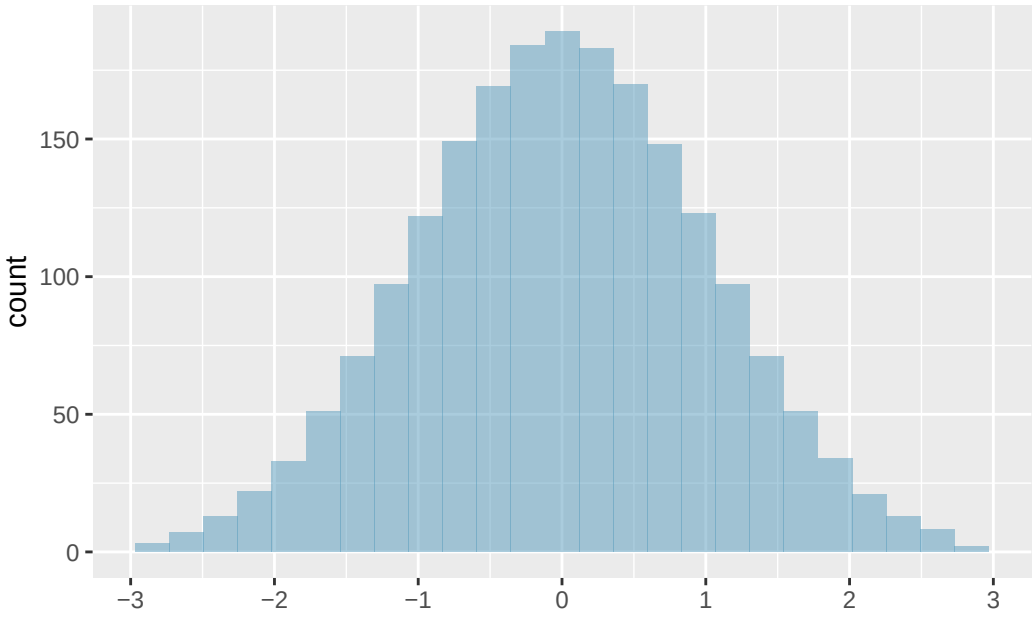


Figure 5.8: The second plot with mean 0 and standard deviation 1.

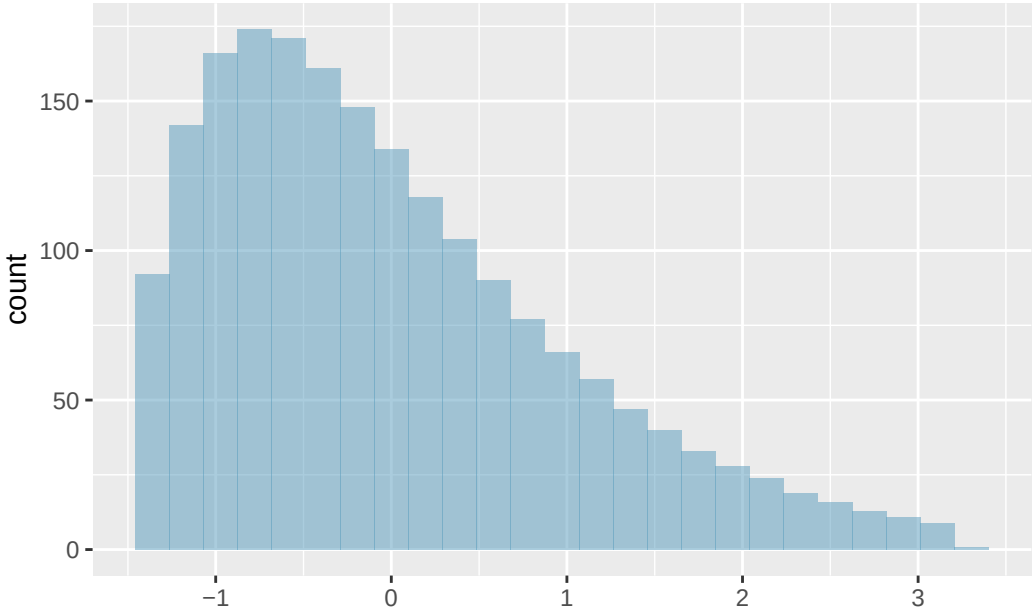


Figure 5.9: The final plot with mean 0 and standard deviation 1.

is symmetric or skewed to one side. Using the three figures, Figures **??**, **??**, **??** as examples, explain why such a description is important.[13]

Example:
Describe the distribution of the `num_char` variable using the histogram in Figure **??**. The description should incorporate the center, variability, and shape of the distribution, and it should also be placed in context: the number of characters in emails. Also note any especially unusual cases/observations.[14]

In practice, the variance and standard deviation are sometimes used as a means to an end, where the *end* is being able to accurately estimate the uncertainty associated with a sample statistic. For example, later in the book we will use the variance and standard deviation to assess how close the sample mean is to the population mean.

[13] Starting with Figure @ref(fig:hist53-fig), the three figures show three distributions that look quite different, but all have the same mean, variance, and standard deviation. Using modality, we can distinguish between the first plot (bimodal) and the last two (unimodal). Using skewness, we can distinguish between the last plot (right skewed) and the first two. While a picture, like a histogram, tells a more complete story, we can use modality and shape (symmetry/skew) to characterize basic information about a distribution.

[14] The distribution of email character counts is unimodal and very strongly skewed to the high end (right skewed). Many of the counts fall near the mean at 11,600, and most fall within one standard deviation (13,130) of the mean. There is one exceptionally long email with about 65,000 characters.

5.2.6 Box plots, quartiles, and the median

A **box plot** summarizes a data set using five statistics, while also plotting unusual observations. Figure **??** provides an annotated vertical dot plot alongside a box plot of the `num_char` variable from the `email50` data set.

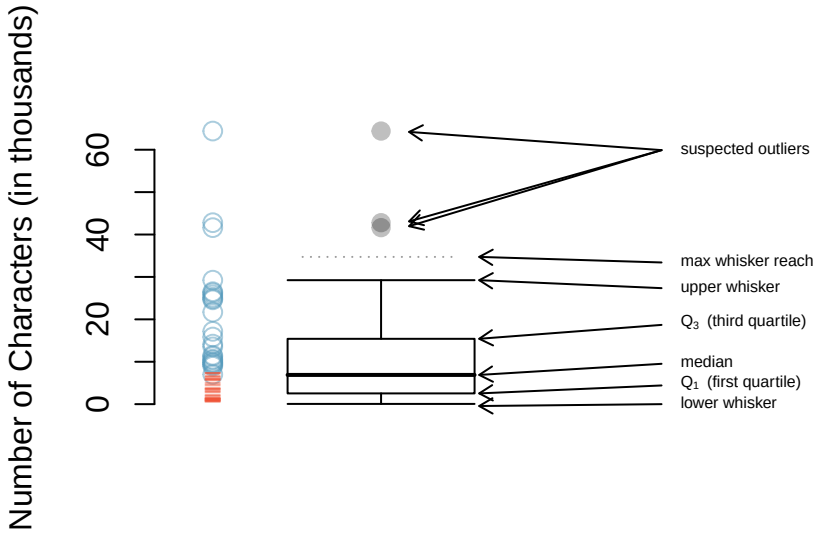


Figure 5.10: A vertical dot plot next to a labeled box plot for the number of characters in 50 emails. The median (6,890), splits the data into the bottom 50% and the top 50%, marked in the dot plot by horizontal dashes and open circles, respectively.

The first step in building a box plot is drawing a dark line denoting the **median**, which splits the data in half. Figure **??** shows 50% of the data falling below the median (red dashes) and the other 50% falling above the median (blue open circles). There are 50 character counts in the data set (an even number) so the data are perfectly split into two groups of 25. We take the median in this case to be the average of the two observations closest to the 50^{th} percentile: $(6{,}768 + 7{,}012)/2 = 6{,}890$. When there are an odd number of observations, there will be exactly one observation that splits the data into two halves, and in this case that observation is the median (no average needed).

Median: the number in the middle
If the data are ordered from smallest to largest, the **median** is the observation in the middle. If there are an even number of observations, there will be two values in the middle, and the median is taken as their average.

The second step in building a box plot is drawing a rectangle to represent the middle 50% of the data. The total length of the box, shown vertically in Figure **??**, is called the **interquartile range** (IQR, for short). It, like the standard deviation, is a measure of variability in the data. The more variable the data, the larger the standard deviation and IQR. The two boundaries

of the box are called the **first quartile** (the 25^{th} percentile, i.e. 25% of the data fall below this value) and the **third quartile** (the 75^{th} percentile), and these are often labeled Q_1 and Q_3, respectively.

Interquartile range (IQR)
The IQR is the length of the box in a box plot. It is computed as

$$IQR = Q_3 - Q_1$$

where Q_1 and Q_3 are the 25^{th} and 75^{th} percentiles, respectively.

Exercise:
What percent of the data fall between Q_1 and the median? What percent is between the median and Q_3?[15]

Extending out from the box, the **whiskers** attempt to capture the data outside of the box, however, their reach is never allowed to be more than $1.5 \times IQR$.[16] They capture everything within this reach. In Figure **??**, the upper whisker does not extend to the last three points, which are beyond $Q_3 + 1.5 \times IQR$, and so it extends only to the last point below this limit. The lower whisker stops at the lowest value, 33, since there is no additional data to reach; the lower whisker's limit is not shown in the figure because the plot does not extend down to $Q_1 - 1.5 \times IQR$. In a sense, the box is like the body of the box plot and the whiskers are like its arms trying to reach the rest of the data.

Any observation that lies beyond the whiskers is labeled with a dot. The purpose of labeling these points – instead of just extending the whiskers to the minimum and maximum observed values – is to help identify any observations that appear to be unusually distant from the rest of the data. Unusually distant observations are called **outliers**. In this case, it would be reasonable to classify the emails with character counts of 41,623, 42,793, and 64,401 as outliers since they are numerically distant from most of the data.

Outliers are extreme
An **outlier** is an observation that is extreme, relative to the rest of the data.

Why it is important to look for outliers
Examination of data for possible outliers serves many useful purposes, including:
1. Identifying **strong skew** in the distribution.
2. Identifying data collection or entry errors. For instance, we re-examined the email purported to have 64,401 characters to ensure this value was accurate.
3. Providing insight into interesting properties of the data.

[15] Since Q_1 and Q_3 capture the middle 50% of the data and the median splits the data in the middle, 25% of the data fall between Q_1 and the median, and another 25% fall between the median and Q_3.
[16] While the choice of exactly 1.5 is arbitrary, it is the most commonly used value for box plots.

Exercise:
The observation with value 64,401, an outlier, was found to be an accurate observation. What would such an observation suggest about the nature of character counts in emails?[17]

Exercise:
Using Figure **??**, estimate the following values for `num_char` in the `email50` data set:
(a) Q_1,
(b) Q_3, and
(c) IQR.[18]

Of course, `R` can calculate these summary statistics for us. First, we will do these calculations individually and then in one function call. Remember to ask yourself what you want `R` to do and what it needs to do this.

```
mean(~num_char, data = email50)
```

```
[1] 11.59822
```

```
sd(~num_char, data = email50)
```

```
[1] 13.12526
```

```
quantile(~num_char, data = email50)
```

```
      0%      25%      50%      75%     100%
 0.05700  2.53550  6.88950 15.41075 64.40100
```

```
iqr(~num_char, data = email50)
```

```
[1] 12.87525
```

```
favstats(~num_char, data = email50)
```

```
   min     Q1 median       Q3    max     mean       sd  n missing
 0.057 2.5355 6.8895 15.41075 64.401 11.59822 13.12526 50       0
```

[17] That occasionally there may be very long emails.
[18] These visual estimates will vary a little from one person to the next: $Q_1 \sim$ 3,000, $Q_3 \sim$ 15,000, IQR $= Q_3 - Q_1 \sim$ 12,000. (The true values: \$Q_1 = \$ 2,536, \$Q_3 = \$ 15,411, IQR = 12,875.)

5.2.7 Robust statistics

How are the *sample statistics* of the `num_char` data set affected by the observation with value 64,401? What would we see if this email wasn't present in the data set? What would happen to these *summary statistics* if the observation at 64,401 had been even larger, say 150,000? These scenarios are plotted alongside the original data in Figure **??**, and sample statistics are computed in `R`.

First, we create a new data frame containing the three scenarios: 1) the original data, 2) the data with the extreme observation dropped, and 3) the data with the extreme observation increased.

```
# code to create the `robust` data frame
p1 <- email50$num_char
p2 <- p1[-which.max(p1)]
p3 <- p1
p3[which.max(p1)] <- 150

robust <- data.frame(value = c(p1, p2, p3),
                     group = c(rep("Original", 50),
                               rep("Dropped", 49), rep("Increased", 50)))
head(robust)
```

```
   value    group
1 21.705 Original
2  7.011 Original
3  0.631 Original
4  2.454 Original
5 41.623 Original
6  0.057 Original
```

Now, we create a side-by-side boxplots for each scenario.

```
gf_boxplot(value ~ group, data = robust, xlab = "Data Group",
           ylab = "Number of Characters (in thousands)") %>%
   gf_theme(theme_classic())
```

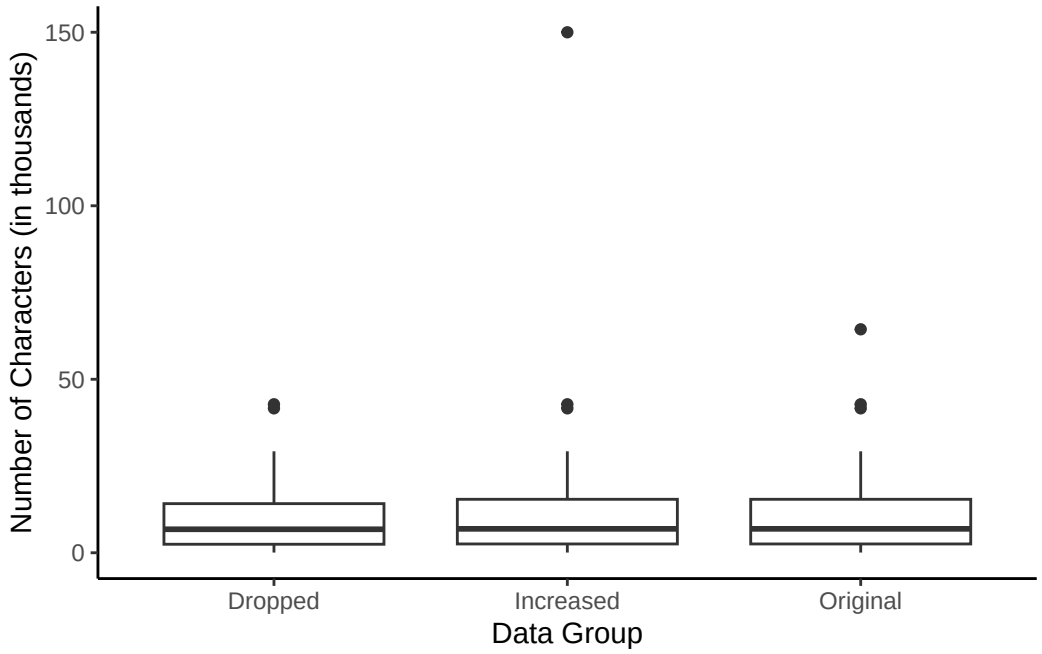


Figure 5.11: Box plots of the original character count data and two modified data sets, one where the outlier at 64,401 is dropped and one where its value is increased.

We can also use `favstats()` to calculate summary statistics of `value` by `group`, using the `robust` data frame created above.

```
favstats(value ~ group, data = robust)
```

```
      group   min     Q1 median       Q3     max     mean       sd  n missing
1   Dropped 0.057 2.4540 6.7680 14.15600  42.793 10.52061 10.79768 49       0
2 Increased 0.057 2.5355 6.8895 15.41075 150.000 13.31020 22.43436 50       0
3  Original 0.057 2.5355 6.8895 15.41075  64.401 11.59822 13.12526 50       0
```

Notice by using the formula notation, we were able to calculate the summary statistics within each group.

Exercise:
(a) Which is affected more by extreme observations, the mean or median? The data summary may be helpful.[19]
(b) Which is affected more by extreme observations, the standard deviation or IQR?[20]

[19]The mean is affected more.
[20]The standard deviation is affected more.

The median and IQR are called **robust statistics** because extreme observations have little effect on their values. The mean and standard deviation are affected much more by changes in extreme observations.

Example:
The median and IQR do not change much under the three scenarios above. Why might this be the case?[21]

Exercise:
The distribution of vehicle prices tends to be right skewed, with a few luxury and sports cars lingering out into the right tail. If you were searching for a new car and cared about price, should you be more interested in the mean or median price of vehicles sold, assuming you are in the market for a regular car?[22]

5.2.8 Transforming data

When data are very strongly skewed, we sometimes transform them so they are easier to model. Consider the histogram of Major League Baseball players' salaries from 2010, which is shown in Figure **??**.

Example:
The histogram of MLB player salaries is somewhat useful because we can see that the data are extremely skewed and centered (as gauged by the median) at about \$1 million. What about this plot is not useful?[23]

There are some standard transformations that are often applied when much of the data cluster near zero (relative to the larger values in the data set) and all observations are positive. A **transformation** is a rescaling of the data using a function. For instance, a plot of the natural logarithm[24] of player salaries results in a new histogram in Figure **??**. Transformed data are sometimes easier to work with when applying statistical models because the transformed data are much less skewed and outliers are usually less extreme.

Transformations can also be applied to one or both variables in a scatterplot. A scatterplot of the original `line_breaks` and `num_char` variables is shown in Figure **??** above. We can see a positive association between the variables and that many observations are clustered near zero. Later in this text, we might want to use a straight line to model the data. However, we'll find

[21] The median and IQR are only sensitive to numbers near Q_1, the median, and Q_3. Since values in these regions are relatively stable – there aren't large jumps between observations – the median and IQR estimates are also quite stable.

[22] Buyers of a *regular car* should be more concerned about the median price. High-end car sales can drastically inflate the mean price while the median will be more robust to the influence of those sales.

[23] Most of the data are collected into one bin in the histogram and the data are so strongly skewed that many details in the data are obscured.

[24] Statisticians often write the natural logarithm as log. You might be more familiar with it being written as ln.

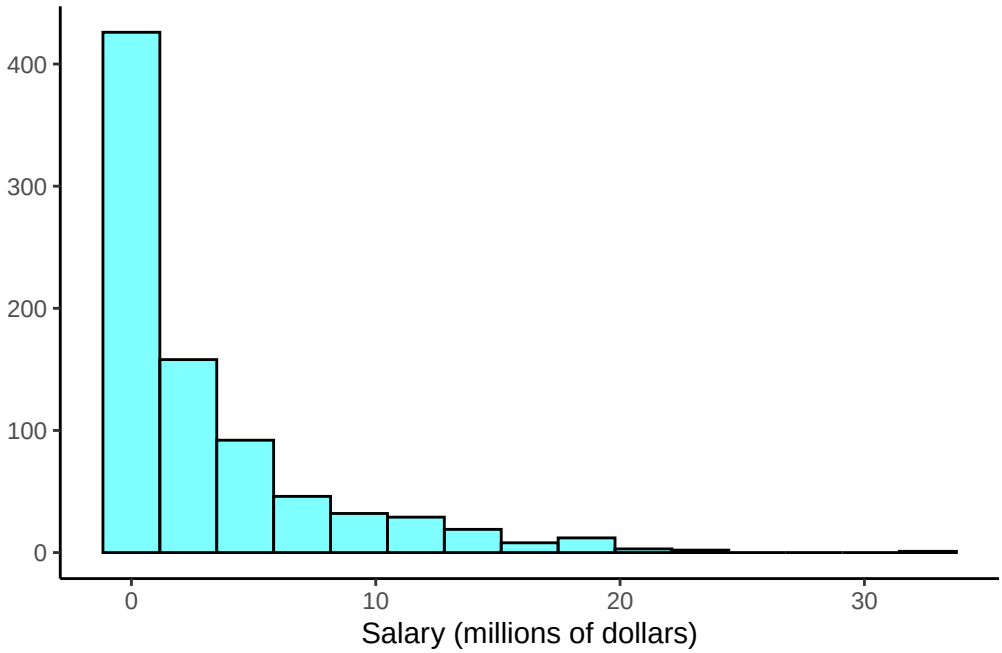


Figure 5.12: Histogram of MLB player salaries for 2010, in millions of dollars.

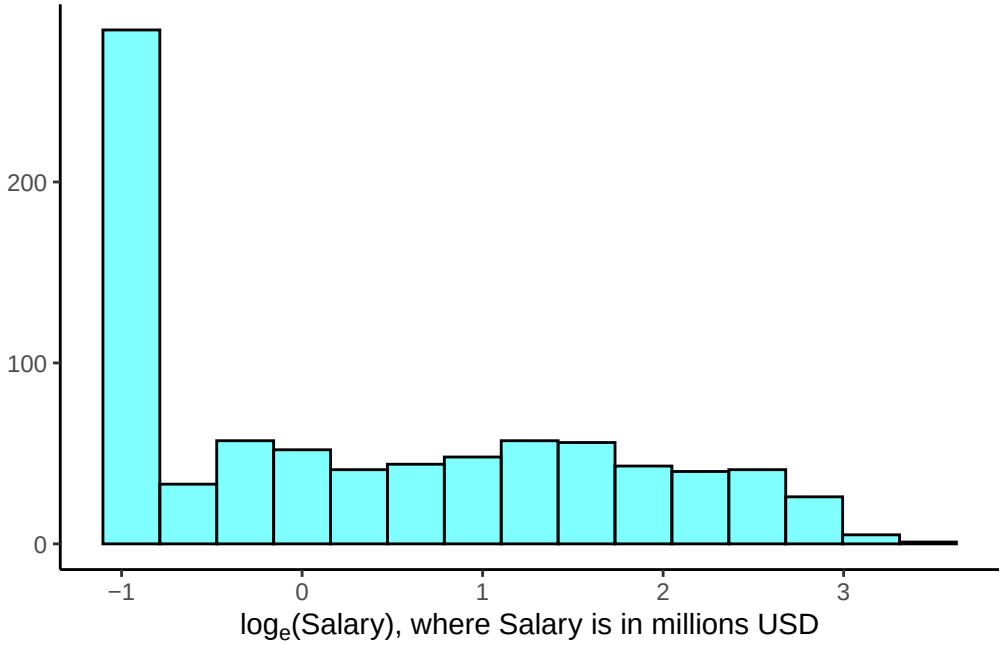


Figure 5.13: Histogram of the log-transformed MLB player salaries for 2010.

that the data in their current state cannot be modeled very well. Figure **??** shows a scatterplot where both `line_breaks` and `num_char` have been transformed using a natural log (log base e) transformation. While there is a positive association in each plot, the transformed data show a steadier trend, which is easier to model than the original (un-transformed) data.

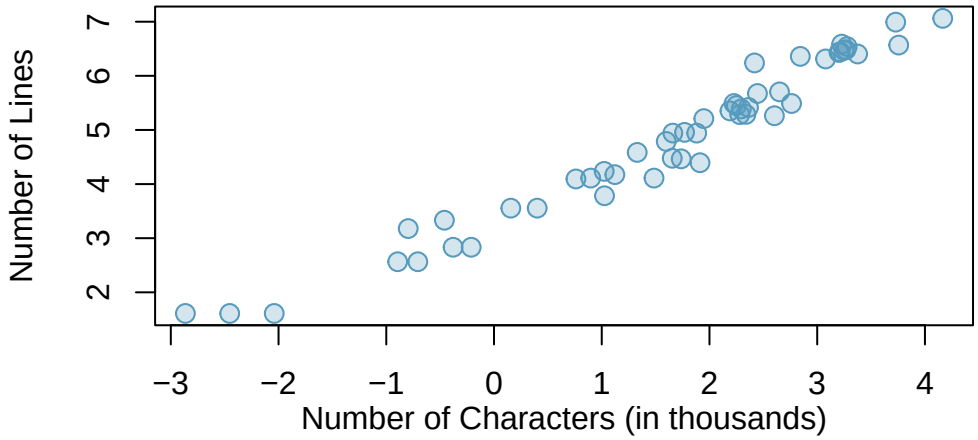


Figure 5.14: A scatterplot of `line_breaks` versus `num_char` for the `email50` data, where both variables have been log-transformed.

Transformations other than the logarithm can be useful, too. For instance, the square root ($\sqrt{\text{original observation}}$) and inverse $\left(\frac{1}{\text{original observation}}\right)$ are used commonly by statisticians. Common goals in transforming data are to see the data structure differently, reduce skew, assist in modeling, or straighten a nonlinear relationship in a scatterplot.

6 Categorical Data

6.1 Objectives

1) Differentiate between various statistical terminologies such as *factor, contingency table, marginal counts, joint counts, frequency table, relative frequency table, bar plot, conditioning, segmented bar plot, mosaic plot, pie chart, side-by-side box plot,* and *density plot*, and construct examples to demonstrate their proper use in context.

2) Using `R`, generate and interpret tables for categorical variables.

3) Using `R`, generate and interpret summary statistics for numerical variables by groups.

4) Create and evaluate graphical summaries of both categorical and numerical variables using `R`, selecting the most appropriate visualization techniques for different types of data and research questions.

5) Synthesize numerical and graphical summaries to provide interpretations and explanations of a data set.

6.2 Categorical data

Like numerical data, categorical data can also be organized and analyzed. This section introduces tables and other basic tools for use with categorical data. Remember at the beginning of this block of material, our case study had categorical data so we have already seen some of the ideas in this chapter.

The `email50` data set represents a sample from a larger email data set called `email`. This larger data set contains information on 3,921 emails. In this section, we will use the `email` data set to examine whether the presence of numbers, small or large, in an email provides any useful information in classifying email as spam or not spam.

6.2.1 Contingency tables and bar plots

In the `email` data set, we have two variables, `spam` and `number`, that we want to summarize. Let's use `inspect()` to get information and insight about the two variables. We can also type `?email` or `help(email)` to learn more about the data. First, load the `openintro` library.

```
library(openintro)
```

```
email %>%
  select(spam, number) %>%
  inspect()
```

```
categorical variables:
    name  class levels    n missing
1 number factor      3 3921       0
                                   distribution
1 small (72.1%), none (14%) ...

quantitative variables:
  name   class min Q1 median Q3 max       mean        sd    n missing
1 spam numeric   0  0      0  0   1 0.09359857 0.2913066 3921       0
```

Notice the use of the `pipe` operator and how it adds to the ease of reading the code. The `select()` function allows us to narrow down the columns/variables to the two of interest. Then `inspect()` gives us information about those variables. We read from top line; we start with the data set `email`, input it into `select()` and select variables from it, and then use `inspect()` to summarize the variables.

As indicated above, `number` is a categorical variable (a *factor*) that describes whether an email contains no numbers, only small numbers (values under 1 million), or at least one big number (a value of 1 million or more). The variable `spam` is a numeric variable, where `1` indicates the email is spam and `0` indicates the email is not spam. To treat `spam` as categorical, we will want to change it to a *factor*, but first we will build a table that summarizes data for the two variables (Table **??**). This table is called a **contingency table**[1]. Each value in the table represents the number of times a particular combination of variable outcomes occurred.

Table 6.1: A contingency table for the `email` data.

[1] A contingency table is a two-way table that shows the distribution of one variable in rows and a second variable in columns.

Spam	Number			
	none	small	big	Total
0	400	2659	495	3554
1	149	168	50	367
Total	549	2827	545	3921

Below is the R code to generate the contingency table.

```
tally(~spam + number, data = email, margins = TRUE)
```

```
       number
spam    none small  big Total
  0      400  2659  495  3554
  1      149   168   50   367
  Total  549  2827  545  3921
```

The value 149 corresponds to the number of emails in the data set that are spam *and* had no numbers listed in the email. Row and column totals are also included. The **row totals** provide the total counts across each row (e.g. $149 + 168 + 50 = 367$), and **column totals** are total counts down each column. The row and column totals are known as **marginal**[2] counts (hence, `margins = TRUE`) and the values in the table are known as **joint**[3] counts.

Let's turn `spam` into a factor and update the `email` data object. We will use `mutate()` to do this.

```
email <- email %>%
  mutate(spam = factor(email$spam, levels = c(1, 0),
                       labels = c("spam", "not spam")))
```

Now, let's check the data again.

```
email %>%
  select(spam, number) %>%
  inspect()
```

[2]Marginal counts are counts based on only one of the variables in a contingency table. For example, there are 367 spam emails in the table.

[3]Joint counts are counts based on both variables in a contingency table. For example, there are 149 emails that are spam *and* contain no numbers.

```
categorical variables:
    name  class levels    n missing
1   spam factor      2 3921       0
2 number factor      3 3921       0
                                  distribution
1 not spam (90.6%), spam (9.4%)
2 small (72.1%), none (14%) ...
```

Let's generate the contingency table again.

```
tally(~spam + number, data = email, margins = TRUE)
```

```
          number
spam       none small  big Total
  spam      149   168   50   367
  not spam  400  2659  495  3554
  Total     549  2827  545  3921
```

A table for a single variable is called a **frequency table**. The table below is a frequency table for the `number` variable.

```
tally(~number, data = email)
```

```
number
 none small   big
  549  2827   545
```

If we replaced the counts with percentages or proportions, the table would be called a **relative frequency table**.

```
tally(~number, data = email, format = 'proportion')
```

```
number
     none     small       big
0.1400153 0.7209895 0.1389952
```

```
round(tally(~number, data = email, format = 'percent'), 2)
```

```
number
 none small   big
 14.0  72.1  13.9
```

A bar plot is a common way to display a single categorical variable. Figure **??** shows a **bar plot** for the `number` variable.

```
email %>%
  gf_bar(~number) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Size of Number", y = "Count")
```

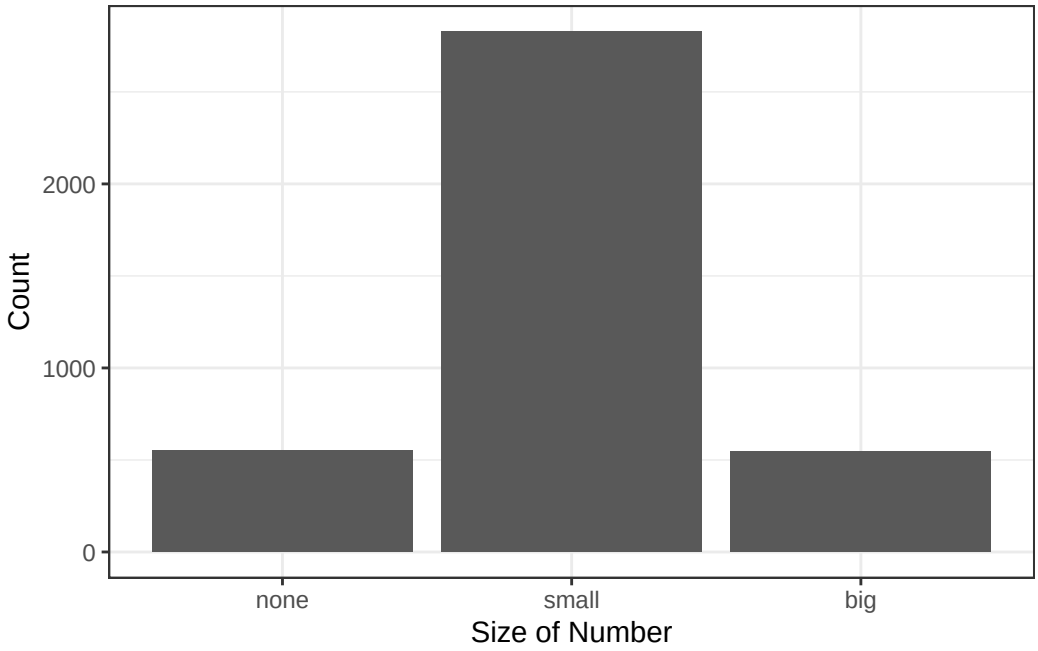


Figure 6.1: Bar chart of the `number` variable.

Next, the counts are converted into proportions (e.g., 549/3921 = 0.140 for `none`) in Figure **??**.

```
email %>%
  gf_props(~number) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Size of Number", y = "Proportion")
```

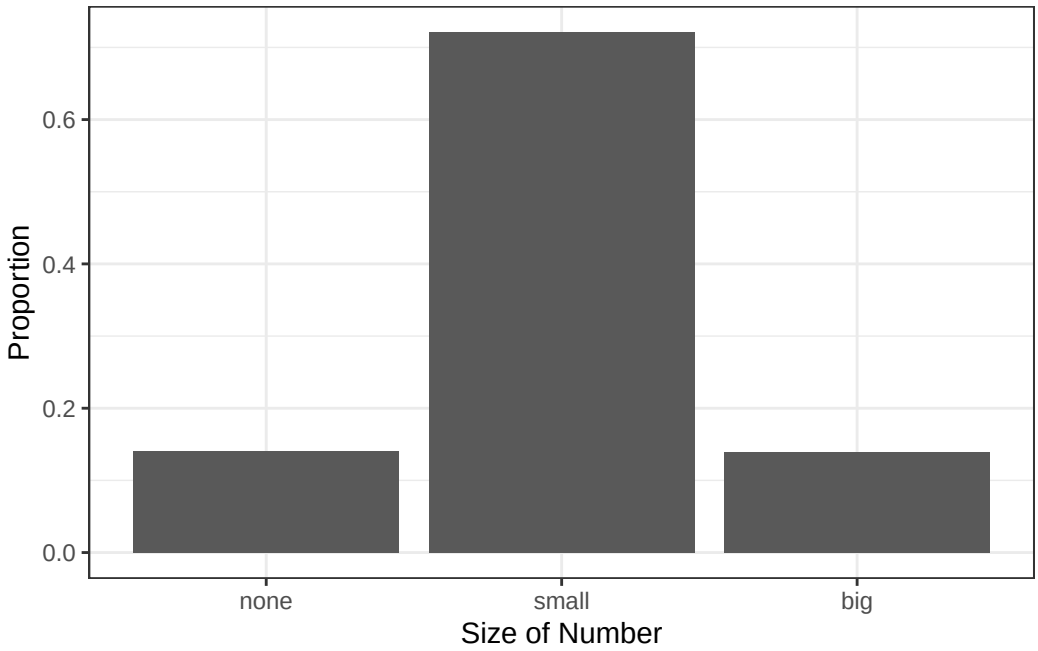
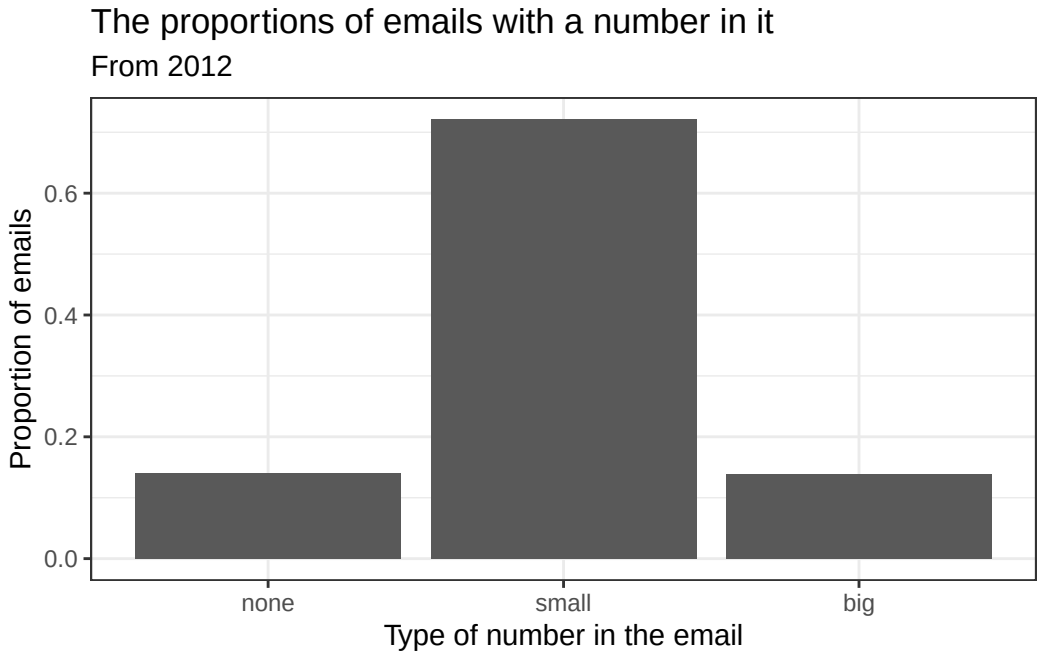


Figure 6.2: Bar chart of the `number` variable as a proportion.

Again, let's clean up the plot into a style that we could use in a report.

```
email %>%
  gf_props(~number,
           title = "The proportions of emails with a number in it",
           subtitle = "From 2012", xlab = "Type of number in the email",
           ylab = "Proportion of emails") %>%
  gf_theme(theme_bw())
```



6.2.2 Column proportions

The table below shows the column proportions. The **column proportions** are computed as the counts divided by their column totals. The value 149 at the intersection of *spam* and *none* is replaced by $149/549 = 0.271$, i.e., 149 divided by its column total, 549. So what does 0.271 represent? It corresponds to the proportion of emails in the sample with no numbers that are spam. That is, the proportion of emails that are spam, out of all the emails with no numbers. We are **conditioning**, restricting, on emails with no number. This rate of spam is much higher than emails with only small numbers (5.9%) or big numbers (9.2%). Because these spam rates vary between the three levels of `number` (*none*, *small*, *big*), this provides evidence that the `spam` and `number` variables are associated.

```
tally(spam ~ number, data = email, margins = TRUE, format = 'proportion')
```

```
          number
spam              none      small        big
  spam     0.27140255 0.05942695 0.09174312
  not spam 0.72859745 0.94057305 0.90825688
  Total    1.00000000 1.00000000 1.00000000
```

The `tally()` function will always condition on the variable on the right-hand side of the tilde, ~, when calculating proportions. Thus, `tally()` only generates column or overall proportions.

It cannot generate row proportions. The more general `table()` function of `R` will allow either column or row proportions.

Exercise:
Create a table of column proportions where the variable `spam` is the column variable.

```
tally(number ~ spam, data = email, margins = TRUE, format = 'proportion')
```

```
       spam
number       spam  not spam
  none  0.4059946 0.1125492
  small 0.4577657 0.7481711
  big   0.1362398 0.1392797
  Total 1.0000000 1.0000000
```

Exercise:
In the table you just created, what does 0.748 represent?[4]

Exercise: Create a table of proportions, where `spam` is the column variable and the values shown represent the proportion of the entire sample in each category.

```
tally(~ number + spam, data = email, margins = TRUE, format = "proportion")
```

```
       spam
number        spam   not spam      Total
  none  0.03800051 0.10201479 0.14001530
  small 0.04284621 0.67814333 0.72098954
  big   0.01275185 0.12624331 0.13899515
  Total 0.09359857 0.90640143 1.00000000
```

Example:
Data scientists use statistics to filter spam from incoming email messages. By noting specific characteristics of an email, a data scientist may be able to classify some emails as spam or not spam with high accuracy. One of those characteristics is whether the email contains no numbers, small numbers, or big numbers. Another characteristic is whether or not an email has any HTML content (given by the `format` variable). A contingency table for the `spam` and `format` variables is needed.
1. Make `format` into a categorical factor variable. The levels should be "text" and "HTML".[5]
2. Create a contingency table from the `email` data set with `format` in the columns and `spam` in the rows.

[4]This is the proportion of `not spam` emails that had a small number in it.
[5]From the help menu on the data, HTML is coded as a 1.

```
email <- email %>%
  mutate(format = factor(email$format, levels = c(1, 0),
                         labels = c("HTML", "text")))
```

In deciding which variable to use as a column, the data scientist would be interested in how the proportion of spam changes within each email format. This corresponds to column proportions based on `format`: the proportion of spam in plain text emails and the proportion of spam in HTML emails.

```
tally(spam ~ format, data = email, margins = TRUE, format = "proportion")
```

```
          format
spam             HTML       text
  spam     0.05796038 0.17489540
  not spam 0.94203962 0.82510460
  Total    1.00000000 1.00000000
```

In generating the column proportions, we can see that a higher fraction of plain text emails are spam ($209/1195 = 17.5\%$) compared to HTML emails ($158/2726 = 5.8\%$). This information on its own is insufficient to classify an email as spam or not spam, as over 80% of plain text emails are not spam. Yet, when we carefully combine this information with many other characteristics, such as `number` and other variables, we stand a reasonable chance of being able to classify an email as spam or not spam.

In constructing a table, we need to think about which variable we want in the column and which in the row. The formula notation in some ways makes us think about the response and predictor variables, with the response variable (left-hand side) displayed in the rows and the predictor variable (right-hand side) displayed in the columns. However, in some cases, it is not clear which variable should be in the column and row and the analyst must decide what is being communicated with the table. Before settling on one form for a table, it is important to consider the audience and the message they are to receive from the table.

> **Exercise**:
> Create two tables with `number` and `spam`: one where `number` is in the columns, and one where `spam` is in the columns. Which table would be more useful to someone hoping to identify spam emails based on the type of numbers in the email?[6]

[6] The table with `number` in the columns will probably be most useful. This table makes it easier to see that emails with small numbers are spam about 5.9% of the time (relatively rare). In contrast, we see that about 27.1% of emails with no numbers are spam, and 9.2% of emails with big numbers are spam.

```
tally(spam ~ number, data = email, format = 'proportion', margin = TRUE)
```

```
          number
spam              none      small        big
  spam     0.27140255 0.05942695 0.09174312
  not spam 0.72859745 0.94057305 0.90825688
  Total    1.00000000 1.00000000 1.00000000
```

```
tally(number ~ spam, data = email, format = 'proportion', margin = TRUE)
```

```
       spam
number       spam  not spam
  none  0.4059946 0.1125492
  small 0.4577657 0.7481711
  big   0.1362398 0.1392797
  Total 1.0000000 1.0000000
```

6.2.3 Segmented bar and mosaic plots

Contingency tables using column proportions are especially useful for examining how two categorical variables are related. Segmented bar and mosaic plots provide a way to visualize the information in these tables.

A **segmented bar plot** is a graphical display of contingency table information. For example, a segmented bar plot representing the table with `number` in the columns is shown in Figure **??**, where we have first created a bar plot using the `number` variable and then separated each group by the levels of `spam` using the `fill` argument.

```
email %>%
  gf_bar(~number, fill = ~spam) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Size of Number", y = "Count")
```

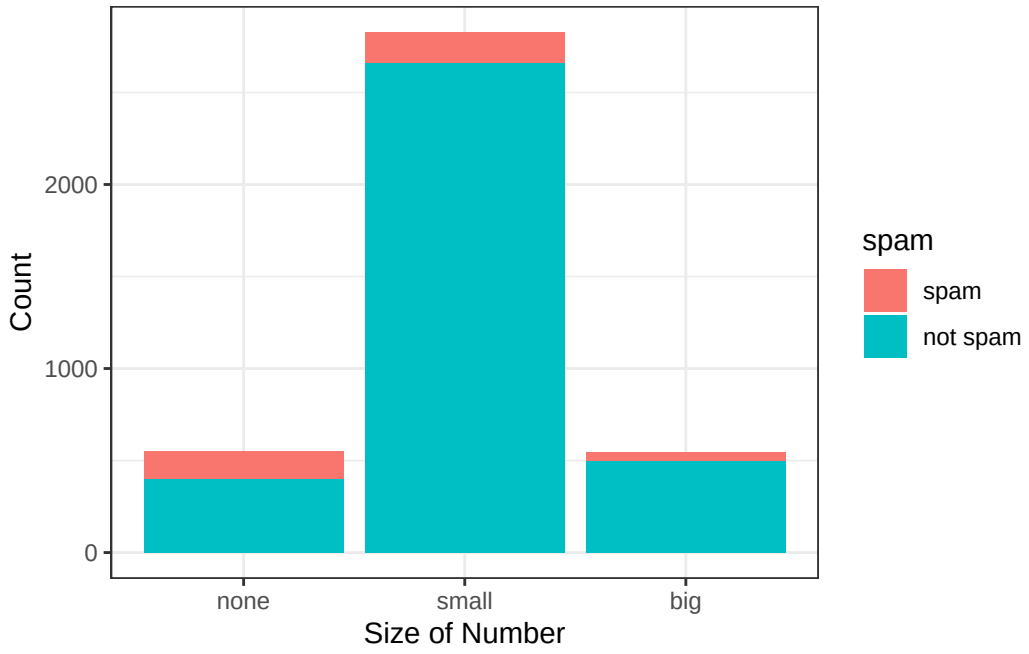


Figure 6.3: Segmented bar plot for numbers found in `emails`, where the counts have been further broken down by `spam`.

The column proportions of the table have been translated into a standardized segmented bar plot in Figure **??**, which is a helpful visualization of the fraction of spam emails within each level of `number`.

```
email %>%
  gf_props(~number, fill = ~spam, position = 'fill') %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Size of Number", y = "Proportion")
```

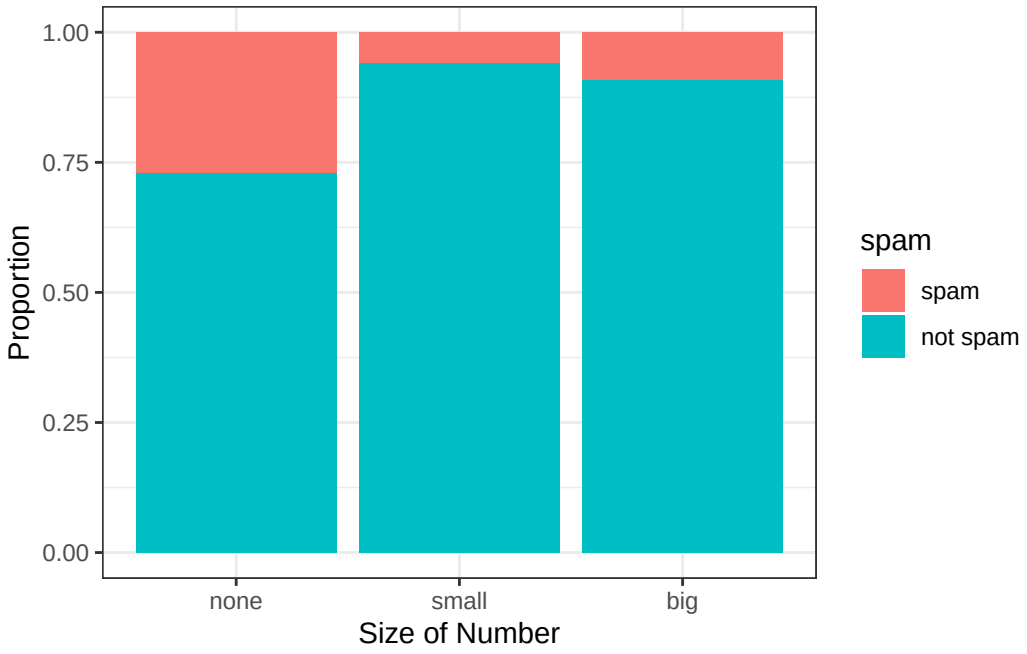


Figure 6.4: Standardized version of Figure **??**.

Example:
Examine both of the segmented bar plots. Which is more useful?[7]

Since the proportion of spam changes across the groups in Figure **??**, we can conclude the variables are dependent, which is something we were also able to discern using table proportions. Because both the `none` and `big` groups have relatively few observations compared to the `small` group, the association is more difficult to see in Figure **??**.

In other cases, a segmented bar plot that is not standardized will be more useful in communicating important information. Before settling on a particular segmented bar plot, create standardized and non-standardized forms and decide which is more effective at communicating features of the data.

A **mosaic plot** is a graphical display of contingency table information that is similar to a bar plot for one variable or a segmented bar plot when using two variables. It seems strange, but mosaic plots are not part of the **mosaic** package. We must load another set of packages called **vcd** and **vcdExtra**. Mosaic plots help to visualize the pattern of associations among variables in two-way and larger tables. Mosaic plots are controversial because they rely on the perception of area; human vision is not good at distinguishing areas.

[7]Figure **??** contains more information, but Figure **??** presents the information more clearly. This second plot makes it clear that emails with no number have a relatively high rate of spam email – about 27%! On the other hand, less than 10% of emails with small or big numbers are spam.

We introduce mosaic plots as another way to visualize contingency tables. Figure **??** shows a one-variable mosaic plot for the `number` variable. Each row represents a level of `number`, and the row heights correspond to the proportion of emails of each number type. For instance, there are fewer emails with no numbers than emails with only small numbers, so the `none` outcome row is shorter in height. In general, mosaic plots use box *areas* to represent the number of observations. Since there is only one variable, the widths are all constant. Thus area is simply related to row height making this visual easy to read.

```
library(vcd)
```

```
mosaic(~number, data = email)
```

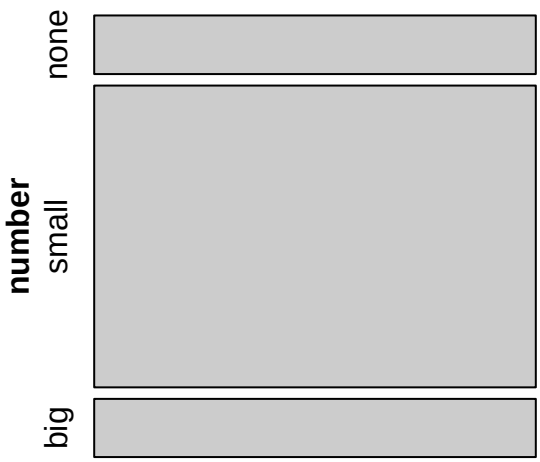


Figure 6.5: Mosaic plot where emails are grouped by the `number` variable.

This one-variable mosaic plot can be further divided into pieces as in Figure **??** using the `spam` variable. The first variable in the formula is used to determine row height. That is, each row is split proportionally according to the fraction of emails in each number category. These heights are similar to those in Figure **??**. Next, each row is split horizontally according to the proportion of emails that were spam in that number group. For example, the second row, representing emails with only small numbers, was divided into emails that were spam (left) and not spam (right). The area of the rectangles represents the overall proportions in the table, where each cell count is divided by the total count. First, we will generate the table and then represent it as a mosaic plot.

```
tally(~number + spam, data = email, format = 'proportion')
```

```
       spam
```

```
number        spam   not spam
  none  0.03800051 0.10201479
  small 0.04284621 0.67814333
  big   0.01275185 0.12624331
```

```
mosaic(~number + spam, data = email)
```

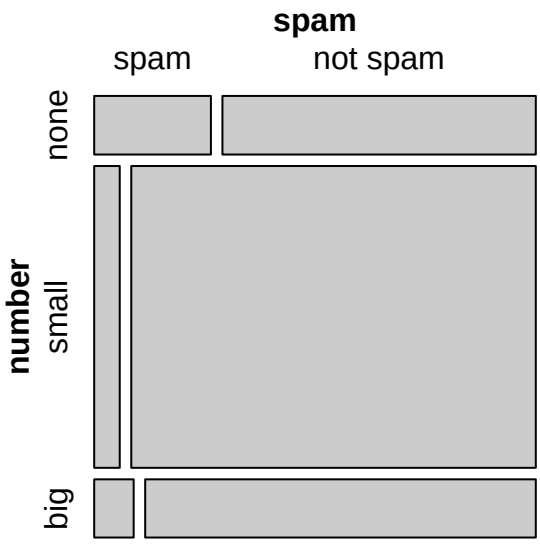


Figure 6.6: Mosaic plot with `number` as the first (row) variable.

These plots are hard to use in a visual comparison of area. For example, is the area for *small* number *spam* emails different from *none* number *spam* emails? The rectangles have different shapes but from the table we can tell the areas are very similar.

An important use of the mosaic plot is to determine if an association between variables may be present. The bottom row of the first column represents spam emails that had big numbers, and the bottom row of the second column represents regular emails that had big numbers. We can again use this plot to see that the `spam` and `number` variables are associated since some rows are divided in different vertical locations than others, which was the same technique used for checking an association in the standardized version of the segmented bar plot.

In a similar way, a mosaic plot representing column proportions where *spam* is in the column could be constructed.

```
mosaic(~spam + number, data = email)
```

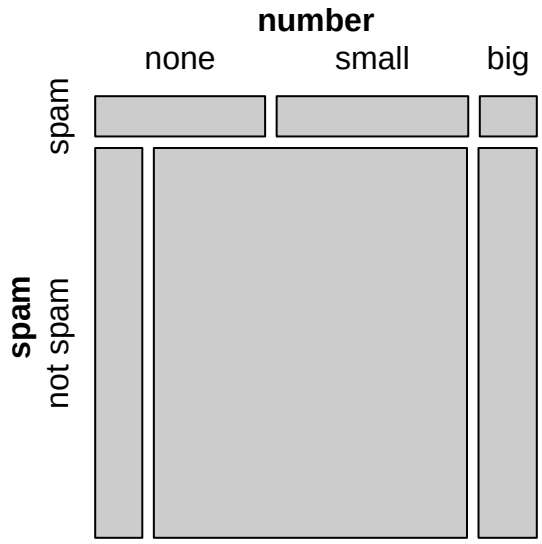


Figure 6.7: Mosaic plot with `spam` as the first (row) variable.

To completely understand the mosaic plot as shown in Figure **??**, let's first find the proportions of `spam`.

```
tally(~spam, data = email, format = "proportion")
```

```
spam
      spam   not spam
0.09359857 0.90640143
```

So, the row heights will be split 90-10. Next, let's find the proportions of `number` within each value of `spam`. In the spam row, *none* will be 41%, *small* will be 46%, and *big* will be 13%. In the not spam row, *none* will be 11%, *small* will be 75%, and *big* will be 14%.

```
tally(number ~ spam, data = email, margins = TRUE, format = "proportion")
```

```
       spam
number       spam  not spam
  none   0.4059946 0.1125492
  small  0.4577657 0.7481711
  big    0.1362398 0.1392797
  Total  1.0000000 1.0000000
```

However, because it is more insightful for this application to consider the fraction of spam in each category of the `number` variable, we prefer Figure **??**.

6.2.4 The only pie chart you will see in this book, hopefully

While pie charts are well known, they are typically not as useful as other charts in a data analysis. A **pie chart** is shown in Figure **??**. It is generally more difficult to compare group sizes in a pie chart than in a bar plot, especially when categories have nearly identical counts or proportions. Just as human vision is bad at distinguishing areas, human vision is also bad at distinguishing angles. In the case of the *none* and *big* categories, the difference is so slight you may be unable to distinguish any difference in group sizes.

```
pie(table(email$number), col = COL[c(3, 1, 2)], radius = 0.75)
```

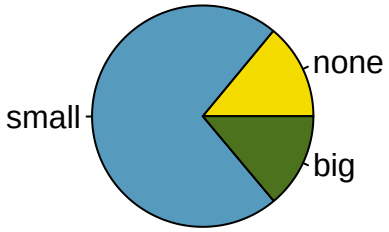


Figure 6.8: A pie chart for `number` in the email data set.

Pie charts are popular in the Air Force due to the ease of generating them in Excel and PowerPoint. However, the values for each slice are often printed on top of the chart making the chart irrelevant. We recommend a minimal use of pie charts in your work.

6.2.5 Comparing numerical data across groups

Some of the more interesting investigations can be done by examining numerical data across groups. This is the case where one variable is categorical and the other is numerical. The methods required here aren't really new. All that is required is to make a numerical plot for each group. Here, two convenient methods are introduced: side-by-side box plots and density plots.

We will again take a look at the subset of the `county_complete` data set. Let's compare the median household income for counties that gained population from 2000 to 2010 versus counties that had no gain. While we might like to make a causal connection here, remember that these are observational data, so such an interpretation would be unjustified.

This section will give us a chance to perform some data wrangling. We will be using the `tidyverse` verbs in the process. Data wrangling is an important part of analysis work and typically makes up a significant portion of the analysis work.

Here is the code to generate the data we need.

```
library(usdata)
```

```
county_tidy <- county_complete %>%
  select(name, state, pop2000, pop2010, fed_spend = fed_spending_2009,
         poverty = poverty_2010, homeownership = homeownership_2010,
         multi_unit = housing_multi_unit_2010, income = per_capita_income_2010,
         med_income = median_household_income_2010) %>%
  mutate(fed_spend = fed_spend / pop2010)
```

First, as a reminder, let's look at the data.

What do we want `R` to do?

We want to select the variables `pop2000`, `pop2010`, and `med_income`.

What does `R` need in order to do this?

It needs the data object, and the desired variable names.

We will use the `select()` and `inspect()` functions.

```
county_tidy %>%
  select(pop2000, pop2010, med_income) %>%
  inspect()
```

```
quantitative variables:
        name   class   min       Q1 median    Q3     max     mean        sd
1    pop2000 numeric    67 11223.50  24621 61775 9519338 89649.99 292547.67
2    pop2010 numeric    82 11114.50  25872 66780 9818605 98262.04 312946.70
3 med_income numeric 19351 36956.25  42450 49144  115574 44274.12  11547.49
     n missing
1 3139       3
2 3142       0
3 3142       0
```

Notice that three counties are missing population values for the year 2000, reported as `NA`. Let's remove them and find which counties increased in population by creating a new variable.

```
cc_reduced <- county_tidy %>%
  drop_na(pop2000) %>%
  select(pop2000, pop2010, med_income) %>%
  mutate(pop_gain = sign(pop2010-pop2000))
```

```
tally(~pop_gain, data = cc_reduced)
```

```
pop_gain
  -1    0    1
1097    1 2041
```

There were 2,041 counties where the population increased from 2000 to 2010, and there were 1,098 counties with no gain. Only 1 county had a net of zero, and 1,0987 had a loss. Let's just look at the counties with a gain or loss in a side-by-side boxplot. Again, we will use `filter()` to select the two groups and then make the variable `pop_gain` into a categorical variable. It's time for more data wrangling.

```
cc_reduced <- cc_reduced %>%
  filter(pop_gain != 0) %>%
  mutate(pop_gain = factor(pop_gain, levels = c(-1, 1),
                           labels = c("Loss", "Gain")))
```

```
inspect(cc_reduced)
```

```
categorical variables:
      name  class levels    n missing
1 pop_gain factor      2 3138       0
                                  distribution
1 Gain (65%), Loss (35%)

quantitative variables:
        name   class   min       Q1  median      Q3     max     mean        sd
1    pop2000 numeric    67 11217.25 24608.0 61783.5 9519338 89669.37 292592.28
2    pop2010 numeric    82 11127.00 25872.0 66972.0 9818605 98359.23 313133.28
3 med_income numeric 19351 36950.00 42443.5 49120.0  115574 44253.24  11528.95
     n missing
1 3138       0
2 3138       0
3 3138       0
```

The **side-by-side box plot** is a traditional tool for comparing across groups. An example is shown in Figure **??** where there are two box plots, one for each group, drawn on the same scale.

```
cc_reduced %>%
  gf_boxplot(med_income ~ pop_gain,
             subtitle = "The income data were collected between 2006 and 2010.",
             xlab = "Population change from 2000 to 2010",
             ylab = "Median Household Income") %>%
  gf_theme(theme_bw())
```

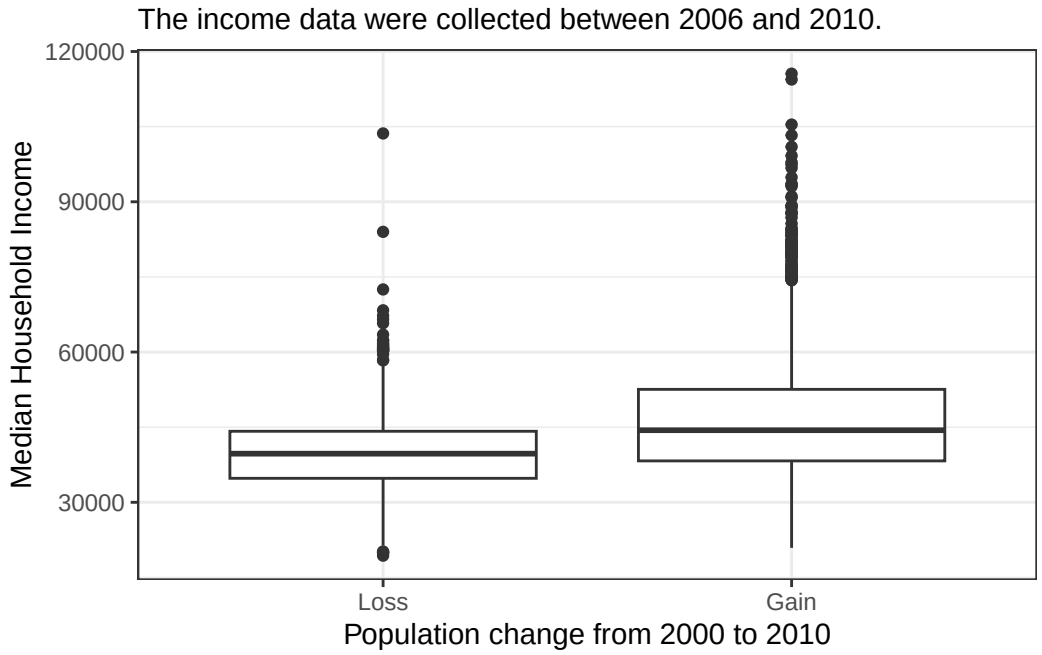


Figure 6.9: Side-by-side box plot for median household income, where the counties are split by whether there was a population gain or loss from 2000 to 2010.

Figure **??** is a plot of the two density curves as another way of comparing median income by whether there was a population gain or loss.

```
cc_reduced %>%
  gf_dens(~med_income, color = ~pop_gain, lwd = 1) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Median household income", y = "Density", col = "Population \nChange")
```

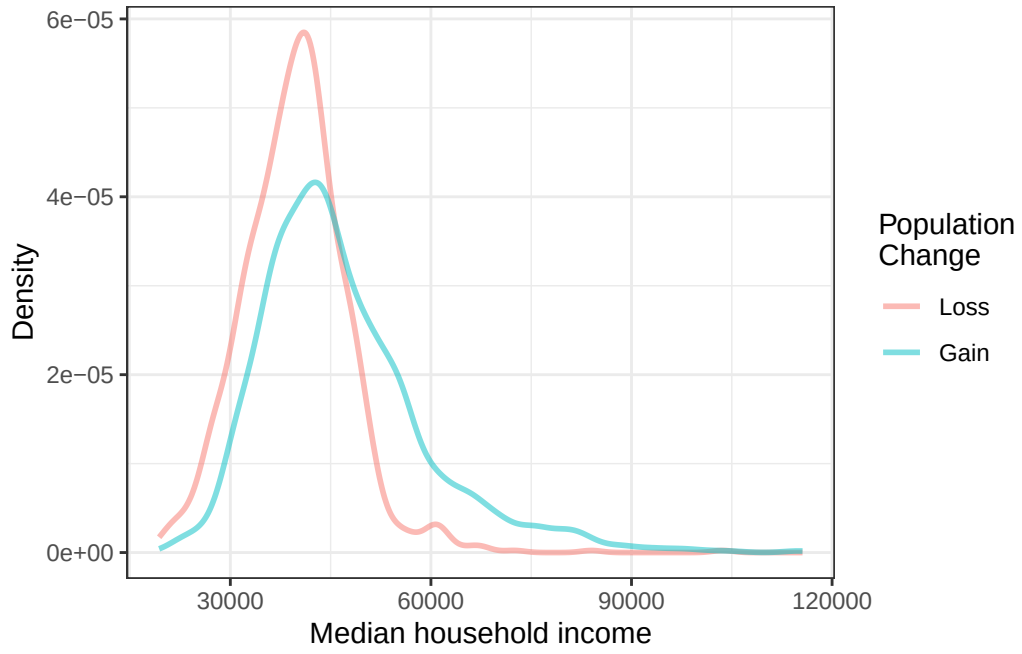


Figure 6.10: Density plots of median household income for counties with population gain versus population loss.

Exercise:
Use the box plots and density plots to compare the incomes for counties across the two groups. What do you notice about the approximate center of each group? What do you notice about the variability between groups? Is the shape relatively consistent between groups? How many *prominent* modes are there for each group?[8]

Exercise:
What components of Figures **??**, **??** do you find most useful?[9]

[8] Answers may vary a little. The counties with population gains tend to have higher income (median of about $45,000) versus counties without a gain (median of about $40,000). The variability is also slightly larger for the population gain group. This is evident in the IQR, which is about 50% bigger in the *gain* group. Both distributions show slight to moderate right skew and are unimodal. There is a secondary small bump at about $60,000 for the *no gain* group, visible in the density plot, that seems out of place. (Looking into the data set, we would find that 8 of these 15 counties are in Alaska and Texas.) The box plots indicate there are many observations far above the median in each group, though we should anticipate that many observations will fall beyond the whiskers when using such a large data set.

[9] The side-by-side box plots are especially useful for comparing centers and spreads, while the density plots are more useful for seeing distribution shape, skew, and groups of anomalies.

Part II

Probability Modeling

7 Probability Case Study

7.1 Objectives

1) Use `R` to simulate a probabilistic model.
2) Gain an introduction to probabilistic thinking through computational, mathematical, and data science approaches.

7.2 Introduction to probability models

In this second block of material, we will focus on probability models. We will provide both a mathematical approach and a computational approach. We will focus on the latter and leave much of the mathematical details to the interested learner. In some cases we can use both methods on a problem and in others only the computational approach is feasible. The mathematical approach to probability modeling allows us insight into the problem and the ability to understand the process. The computational approach has a much greater ability to generalize but can be time intensive to run and often requires the writing of custom functions.

This case study is extensive and may seem overwhelming, but do not worry. We will discuss these ideas again in the many chapters we have coming up this block.

7.3 Probability models

Probability models are an important tool for data analysts. They are used to explain variation in outcomes that cannot be explained by other variables. We will use these ideas in the Statistical Modeling Block to help us make decisions about our statistical models.

Often probability models are used to answer a question of the form "What is the chance that …..?" This means that we typically have an experiment or trial where multiple outcomes are possible and we only have an idea of the frequency of those outcomes. We use this frequency as a measure of the probability of a particular outcome.

For this block we will focus just on probability models. To apply a probability model we will need to

1. Describe the experiment and its possible outcomes.
2. Determine probability values for the outcomes, which may include **parameters** that determine the probabilities.
3. Understand the assumptions behind the model.

7.4 Case study

There is a famous example of a probability question that we will examine in this case study. The question we want to answer is "In a room of n people, what is the chance at least two people have the same birthday?"

> **Exercise**:
> The typical classroom at USAFA has 18 students in it. What do you think is the chance that at least two students have the same birthday?[1]

7.4.1 Break down the question

The first action we should take is to understand what is being asked.

1. What is the experiment or trial?
2. What does it mean to have the same birthday?
3. How should we handle leap years?
4. Should we consider the frequency of births? Are some days less likely than others?

> **Exercise**:
> Discuss these questions and others that you think are relevant.[2]

The best first step is to develop a simple model. Often these are the only ones that will have a mathematical solution. For our problem this means we answer the above questions.

1. We have a room of 18 people and we look at their birthdays. We either have two or more birthdays matching or not; thus there are two outcomes.
2. At least two people having the same birthday means we could have multiple matches on the same day or we could have several different days where multiple people have matching birthdays.
3. We don't care about the year, only the day and month. Thus, two people born on May 16th are a match.
4. We will ignore leap years, for now.
5. We will assume that a person has equal probability of being born on any of the 365 days of the year.

[1] The answer is around 34.7%. How close were you? Did you think it was lower or higher?
[2] Another question may be "What does it mean at least two people have matching birthdays?"

7.4.2 The computational approach (simulation)

Now that we have an idea about the structure of the problem, we next need to think about how we would simulate a single classroom. We have 18 students in the classroom and they all could have any of the 365 days of the year as a birthday. What we need to do is sample birthdays for each of the 18 students. But how do we code the days of the year?

An easy solution is to just label the days from 1 to 365. The function `seq()` does this for us.

```
days <- seq(1,365)
```

Next we need to pick one of the days using the `sample()` function. Note that we set a seed to make our results reproducible. This is not required, but is strongly encouraged.

```
set.seed(2022)
sample(days,1)
```

```
[1] 228
```

The first student in the classroom was born on the 228th day of the year, which corresponds to the 16th of August.

Since `R` works on vectors of information, we don't have to write a loop to select 18 days. We have the `sample()` function do it for us.

Remember to ask yourself:

- *What do we want `R` to do?*

We want `R` to sample 18 birthdays from the numbers 1 to 365. We want to sample *with replacement* because we're allowing students to have the same birthday.

- *What does `R` need to do that?*

A quick look at the documentation for the `sample()` function (using `?sample` or `help(sample)`) tells us we need a vector of data, `size` specifying the number of items to choose from the vector, and a logical value for `replace`, specifying whether to sample with replacement or not.

```
set.seed(2022)
class <- sample(days, size = 18, replace = TRUE)
class
```

```
 [1] 228 206 311 331 196 262 191 206 123 233 270 248   7 349 112   1 307 288
```

Notice in our sample we have at least one match, although it is difficult to look at this list and see the match. Let's sort them to make it easier for us to see.

```
sort(class)
```

```
 [1]   1   7 112 123 191 196 206 206 228 233 248 262 270 288 307 311 331 349
```

There are two birthdays on day 206, corresponding to the 25th of July.

The next step is to find a way in `R` for the code to detect that there is a match.

> **Exercise**:
> What idea(s) do you have to determine if a match exists?

We could sort the data and look at differences in sequential values and then check if the set of differences contains a zero. This seems to be computationally expensive. Instead we will use the function `unique()` which gives a vector of unique values in an object. The function `length()` gives the number of elements in the vector.

```
length(unique(class))
```

```
[1] 17
```

Since we only have 17 unique values in a vector of size 18, we have a match. Now let's put this all together to generate another classroom of size 18.

```
length(unique(sample(days, size = 18, replace = TRUE)))
```

```
[1] 16
```

The next problem that needs to be solved is how to repeat the classrooms and keep track of those that have a match. There are several functions we could use to include `replicate()`, but we will use `do()` from the **mosaic** package because it returns a data frame so we can use `tidyverse` verbs to wrangle the data.

The `do()` function allows us to repeat an operation many times. The following template can be used,

```
do(n) * {stuff to do}              # pseudo-code
```

where {stuff to do} is typically a single R command, but may be something more complicated.

Let's try it out. First, we'll load the libraries.

```
library(mosaic)
library(tidyverse)
```

```
do(5)*length(unique(sample(days,size=18,replace = TRUE)))
```

```
  length
1     18
2     17
3     17
4     17
5     18
```

Let's repeat this process for a larger number of simulated classrooms. Remember, you should be asking yourself two questions:

- *What do I want R to do?*
- *What does R need to do this?*

```
(do(1000)*length(unique(sample(days, size = 18, replace = TRUE)))) %>%  # simulate 1000 class
  mutate(match = if_else(length == 18, 0, 1)) %>%                       # check for matches,
  summarise(prob = mean(match))                                         # probability is the
```

```
   prob
1 0.365
```

This is within two decimal places of the mathematical solution we will develop shortly.

How many classrooms do we need to simulate to get an accurate estimate of the probability of a match? That is a statistical modeling question and it depends on how much variability we can accept. We will discuss these ideas later in the book. For now, we can run the code multiple times and see how the estimate varies. When computational power is cheap, we can increase the number of simulations.

```
(do(10000)*length(unique(sample(days, size = 18, replace = TRUE)))) %>%
  mutate(match = if_else(length == 18, 0, 1)) %>%
  summarise(prob = mean(match))
```

```
    prob
1 0.3402
```

7.4.3 Plotting

The method we have used to create the data allows us to summarize the number of unique birthdays using a table or bar chart. Let's do that now. Note that since the first argument in `tally()` is not data, the **pipe** operator will not work without some extra effort. We must tell `R` that the data is the previous argument in the pipeline and thus use the symbol `.` to denote this.

```
(do(1000)*length(unique(sample(days, size = 18, replace = TRUE)))) %>%
  tally(~length, data = .)
```

```
length
 15  16  17  18
  5  46 269 680
```

Figure **??** is a plot of the number of unique birthdays (out of 18) in our sample.

```
(do(1000)*length(unique(sample(days, size = 18, replace = TRUE)))) %>%
  gf_bar(~length) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Number of unique birthdays", y = "Count")
```

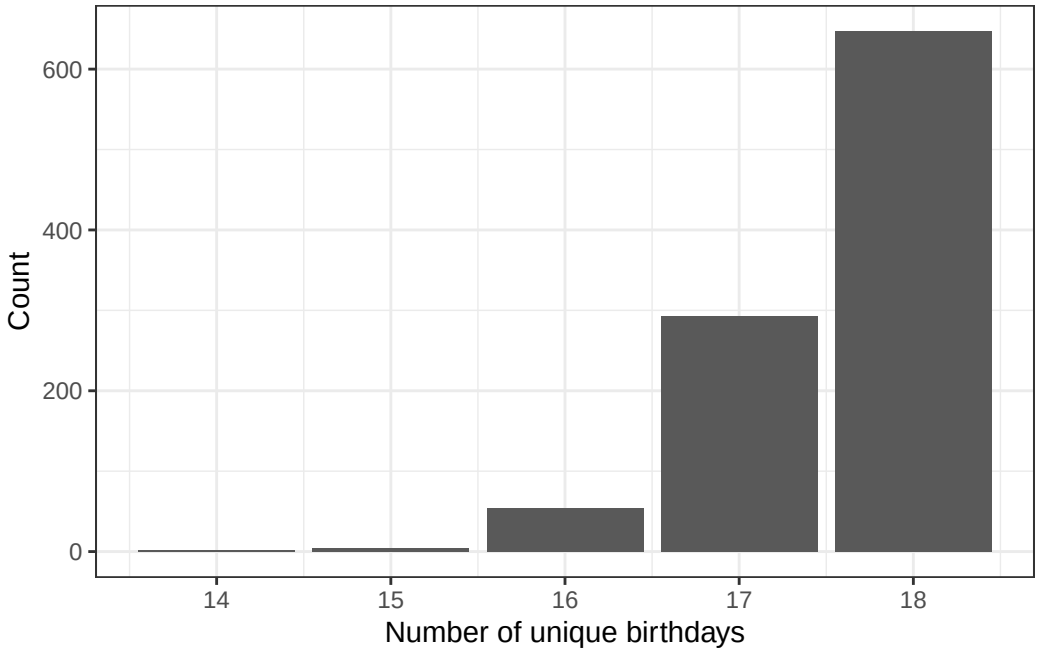


Figure 7.1: Bar chart of the number of unique birthdays in the sample.

Exercise:
What does it mean if the length of unique birthdays is 16, in terms of matches?[3]

7.4.4 The mathematical approach

To solve this problem mathematically, we will work through the logic one step at a time. One of the key ideas that we will see many times is the idea of the **multiplication** rule. This idea is the foundation for **permutations** and **combinations**, which are counting methods frequently used in probability calculations.

The first step that we take is to understand the idea of two or more people having the same birthday. With 18 people, there are a great deal of possibilities for two or more people to have the same birthday. We could have exactly two people with the same birthday. We could have 18 people with the same birthday. We could have three people with the same birthday and another two people with the same birthday but different from the other three. Accounting for all these possibilities is too large a counting process. Instead, we will take the approach of finding the probability of no one having a matching birthday. Then the probability of at least two people having a matching birthday is one minus the probability that no one has a matching birthday. This is known as a **complementary** probability. A simpler example is to think about rolling a single die. The probability of rolling a six is equivalent to one minus the probability of not rolling a six (rolling any number other than six).

We first need to think about all the different ways we could get 18 birthdays. This is going to be our denominator in the probability calculation. The first person could have 365 different days for their birthday. The second person could also have 365 different birthdays. The same is true for all 18 people. This is an example of the *multiplication rule*. For 18 people, there are 365^{18} possible sets of birthdays.[4] Again, this will be our denominator in calculating the probability.

Because we're using the complement, the numerator is the number of ways that 18 people can have birthdays with no matches.

Exercise:
What is the number of ways for 18 people to have no birthday matches?

The first person can have a birthday on any day of the year, so there are 365 possibilities. Because we don't want a match, the second person only has 364 possibilities for a birthday. The third person can't match either of the first two, so there are only 363 possibilities for that birthday. Thus, there are $365 \times 364 \times 363... \times 349 \times 348$ ways for 18 people to have no birthday matches.

[3] It is possible that 3 people all have the same birthday or two sets of 2 people have the same birthday but different from the other pair.
[4] $365^{18} = 1.322 \times 10^{46}$

This looks like a truncated factorial. Remember a factorial, written as $n!$ with an explanation point, is the product of successive positive integers. For example, 3! is $3 \times 2 \times 1$ or 6. In our birthday example, we could write the multiplication for the numerator as

$$365 \times 364 \times ... \times 348 = \frac{365!}{(365 - 18)!}$$

As we will learn, the multiplication rule for the numerator is known as a **permutation**.[5]

We are ready to put it all together. For 18 people, the probability of two or more people with the same birthday is one minus the probability that no one has the same birthday, which is

$$1 - \frac{\frac{365!}{(365-18)!}}{365^{18}}$$

or

$$1 - \frac{\frac{365!}{347!}}{365^{18}}$$

In `R`, there is a `factorial()` function but factorials get large fast and will **overflow** the memory. Try `factorial(365)` in `R` to see what happens.

```
factorial(365)
```

```
[1] Inf
```

It is returning *infinity* because the number is too large for the buffer. As is often the case when using a computational method, we must be clever about our approach. Instead of using factorials, we can make use of `R` and its ability to work on vectors. If we provide `R` with a vector of values, the `prod()` function will calculate the product of all the elements.

```
365*364
```

```
[1] 132860
```

```
prod(365:364)
```

```
[1] 132860
```

[5] This could be more generally written as $\frac{365!}{(365-n)!}$ for a group of n people.

Now, we calculate the probability of at least two people in a room of 18 having the same birthday.

```
1- prod(365:348) / (365^18)
```

```
[1] 0.3469114
```

This is close to the probability we found with simulation.

7.4.5 General solution

We now have the mathematics to understand the problem. We can easily generalize this to any number of people. To do this, we have to write a function in `R`. As with everything in `R`, we save a function as an object. The general format for creating a function is

```
my_function <- function(parameters){
  code for function
}
```

For this problem we will call the function `birthday_prob()`. The only parameter we need is the number of people in the room, `n`. Let's write this function.

```
birthday_prob <- function(n=20){
  1 - prod(365:(365 - (n - 1))) / (365^n)
}
```

Notice we assigned the function to the name `birthday_prob`, we told `R` to expect one argument to the function, which we are calling `n`, and then we provide `R` with the code to find the probability. We set a default value for `n` in case one is not provided to prevent an error when the function is run. We will learn more about writing functions throughout this book and in the follow-on USAFA course, Math 378: Applied Statistical Modeling.

Let's test the code with a known answer.

```
birthday_prob(18)
```

```
[1] 0.3469114
```

Now we can determine the probability for any size room. You may have heard that it only takes about 23 people in a room to have a 50% probability of at least two people matching birthdays.

```
birthday_prob(23)
```

```
[1] 0.5072972
```

Let's create a plot of the probability versus the number of people in the room. To do this, we need to apply the function to a vector of values. The function `sapply()` will work or we can also use `Vectorize()` to alter our existing function. We choose the latter option.

First notice what happens if we input a vector into our function.

```
birthday_prob(1:20)
```

```
Warning in 365:(365 - (n - 1)): numerical expression has 20 elements: only the
first used
```

```
 [1] 0.0000000 0.9972603 0.9999925 1.0000000 1.0000000 1.0000000 1.0000000
 [8] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[15] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
```

It only uses the first value. There are several ways to solve this problem. We can use the `map()` function in the **purrr** package. This idea of mapping a function to a vector is important in data science. It is used in scenarios where there is a lot of data. In this case, the idea of map-reduce is used to make the analysis amenable to parallel computing.

```
map_dbl(1:20, birthday_prob)
```

```
 [1] 0.000000000 0.002739726 0.008204166 0.016355912 0.027135574 0.040462484
 [7] 0.056235703 0.074335292 0.094623834 0.116948178 0.141141378 0.167024789
[13] 0.194410275 0.223102512 0.252901320 0.283604005 0.315007665 0.346911418
[19] 0.379118526 0.411438384
```

We could also just vectorize the function.

```
birthday_prob <- Vectorize(birthday_prob)
```

Now notice what happens.

```
birthday_prob(1:20)
```

```
 [1] 0.000000000 0.002739726 0.008204166 0.016355912 0.027135574 0.040462484
 [7] 0.056235703 0.074335292 0.094623834 0.116948178 0.141141378 0.167024789
[13] 0.194410275 0.223102512 0.252901320 0.283604005 0.315007665 0.346911418
[19] 0.379118526 0.411438384
```

We now have what we want, so let's create our line plot, Figure **??**.

```
gf_line(birthday_prob(1:100) ~ seq(1, 100),
        xlab = "Number of People",
        ylab = "Probability of Match",
        title = "Probability of at least two people with matching birthdays by room size") %>
  gf_theme(theme_bw())
```

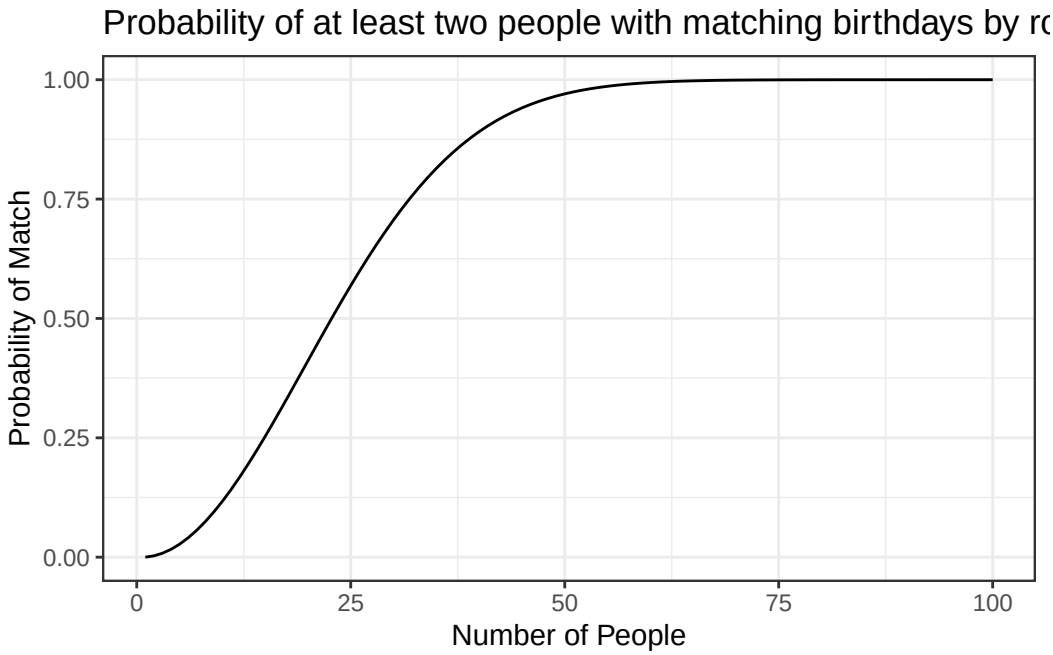


Figure 7.2: The probability of at least 2 people having matching birthdays.

Is this what you expected the curve to look like? We, the authors, did not expect this. It has a sigmoidal shape with a large increase in the middle range and flattening in the tails.

7.4.6 Data science approach

The final approach we will take is one based on data, a data science approach. The **mosaicData** package includes a data set called `Births` that contains the number of births in the US from 1969 to 1988. This data will allow us to estimate the number of births on any day of the year. This allows us to eliminate the reliance on the assumption that each day is equally likely. Let's first `inspect()` the data object.

```
inspect(Births)
```

```
categorical variables:
  name   class levels    n missing
1 wday ordered      7 7305       0
                                   distribution
1 Wed (14.3%), Thu (14.3%), Fri (14.3%) ...

Date variables:
  name class      first       last min_diff max_diff    n missing
1 date  Date 1969-01-01 1988-12-31   1 days   1 days 7305       0

quantitative variables:
          name   class  min   Q1 median    Q3   max        mean          sd
1       births integer 6675 8792   9622 10510 12851 9648.940178 1127.315229
2         year integer 1969 1974   1979  1984  1988 1978.501027    5.766735
3        month integer    1    4      7    10    12    6.522930    3.448939
4  day_of_year integer    1   93    184   275   366  183.753593  105.621885
5 day_of_month integer    1    8     16    23    31   15.729637    8.800694
6  day_of_week integer    1    2      4     6     7    4.000274    1.999795
     n missing
1 7305       0
2 7305       0
3 7305       0
4 7305       0
5 7305       0
6 7305       0
```

Notice there are leap years present in this data. It could be argued that we could randomly pick one year and use it. Let's see what happens if we just used 1969, a non-leap year. Figure **??** is a scatter plot of the number of births in 1969 for each day of the year.

```
Births %>%
  filter(year == 1969) %>%
  gf_point(births ~ day_of_year) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Day of the Year", y = "Number of Births")
```

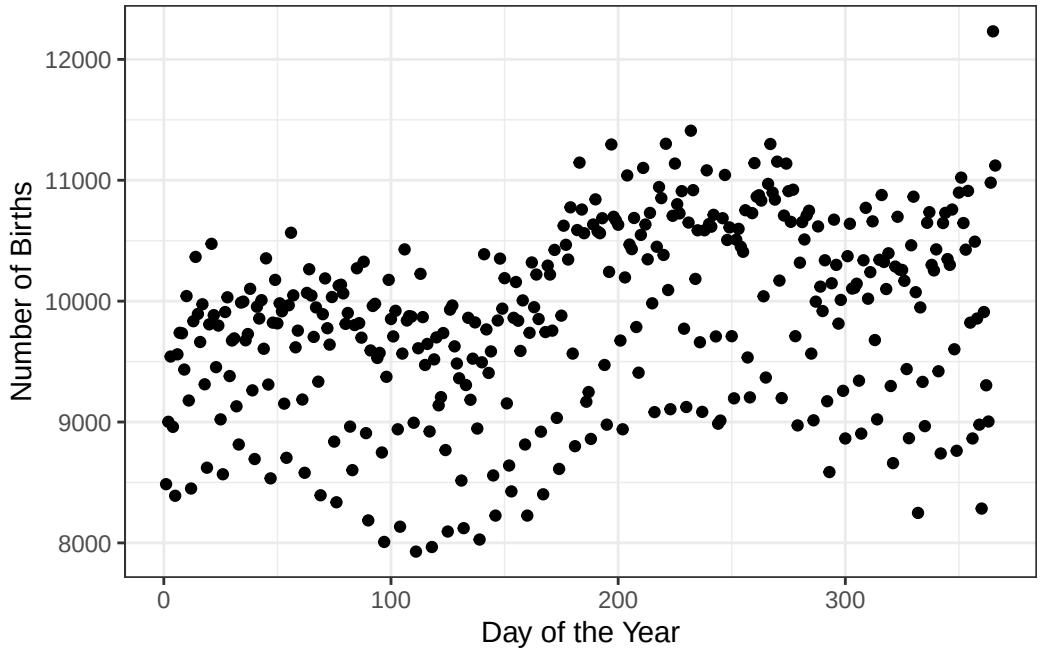


Figure 7.3: The number of births for each day of the year in 1969.

Exercise:
What patterns do you see in Figure **??**? What might explain them?[6]

There are definitely bands appearing in the data which could be the day of the week; there are fewer birthdays on the weekend. There is also seasonality with more birthdays in the summer and fall. There is also probably an impact from holidays.

Quickly, let's look at the impact of day of the week by using color for day of the week. Figure **??** makes it clear that the weekends have fewer births as compared to the work week.

```
Births %>%
  filter(year == 1969) %>%
  gf_point(births ~ day_of_year, color = ~factor(day_of_week)) %>%
```

[6]This could be due to doctors tending not to schedule inductions or C-sections on weekends. Fun fact: while more than 30% of all births were C-sections in 2023, only around 5% of births were C-sections in 1969.

```
  gf_labs(x = "Day of the Year", col = "Day of Week") %>%
  gf_theme(theme_bw())
```

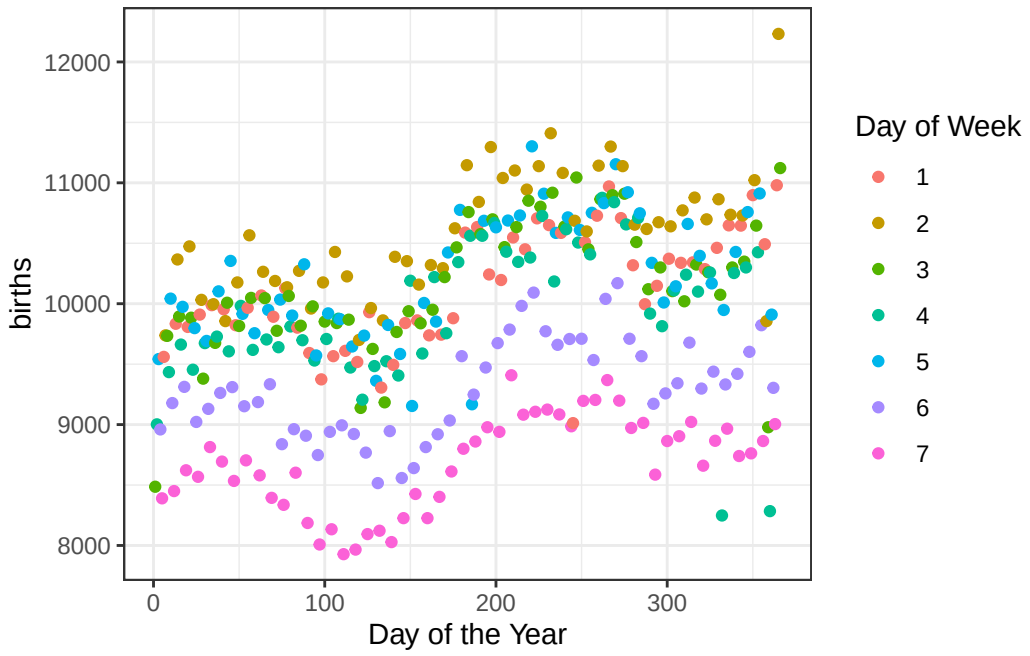


Figure 7.4: The number of births for each day of the year in 1969 broken down by day of the week.

By only using one year, this data might give poor results because holidays will fall on certain days of the week and the weekends will also be impacted. Note that we also still have the problem of leap years.

```
Births %>%
  group_by(year) %>%
  summarise(n = n())
```

```
# A tibble: 20 x 2
    year     n
   <int> <int>
 1  1969   365
 2  1970   365
 3  1971   365
 4  1972   366
 5  1973   365
```

```
 6  1974   365
 7  1975   365
 8  1976   366
 9  1977   365
10  1978   365
11  1979   365
12  1980   366
13  1981   365
14  1982   365
15  1983   365
16  1984   366
17  1985   365
18  1986   365
19  1987   365
20  1988   366
```

The years 1972, 1976, 1980, 1984, and 1988 are all leap years. At this point, to make the analysis easier, we will drop those years.

```
Births %>%
  filter(!(year %in% c(1972, 1976, 1980, 1984, 1988))) %>%
  group_by(year) %>%
  summarise(n = n())
```

```
# A tibble: 15 x 2
    year     n
   <int> <int>
 1  1969   365
 2  1970   365
 3  1971   365
 4  1973   365
 5  1974   365
 6  1975   365
 7  1977   365
 8  1978   365
 9  1979   365
10  1981   365
11  1982   365
12  1983   365
13  1985   365
14  1986   365
15  1987   365
```

Notice that we used the `%in%` operator inside the `filter()` function. This is a **logical** argument checking whether `year` is one of the specified values. The `!` at the front negates this, in a sense requiring `year` to not be one of those values.

We are almost ready to simulate. We need to get the count of `births` on each day of the year for the non-leap years.

```
birth_data <- Births %>%
  filter(!(year %in% c(1972, 1976, 1980, 1984, 1988))) %>%
  group_by(day_of_year) %>%
  summarise(n = sum(births))
```

```
head(birth_data)
```

```
# A tibble: 6 x 2
  day_of_year      n
        <int>  <int>
1           1 120635
2           2 129042
3           3 135901
4           4 136298
5           5 137319
6           6 140044
```

Let's look at a plot of the number of births versus day of the year for all the non-leap years in Figure **??**.

```
birth_data %>%
  gf_point(n ~ day_of_year,
           xlab = "Day of the year",
           ylab = "Number of births") %>%
  gf_theme(theme_bw())
```

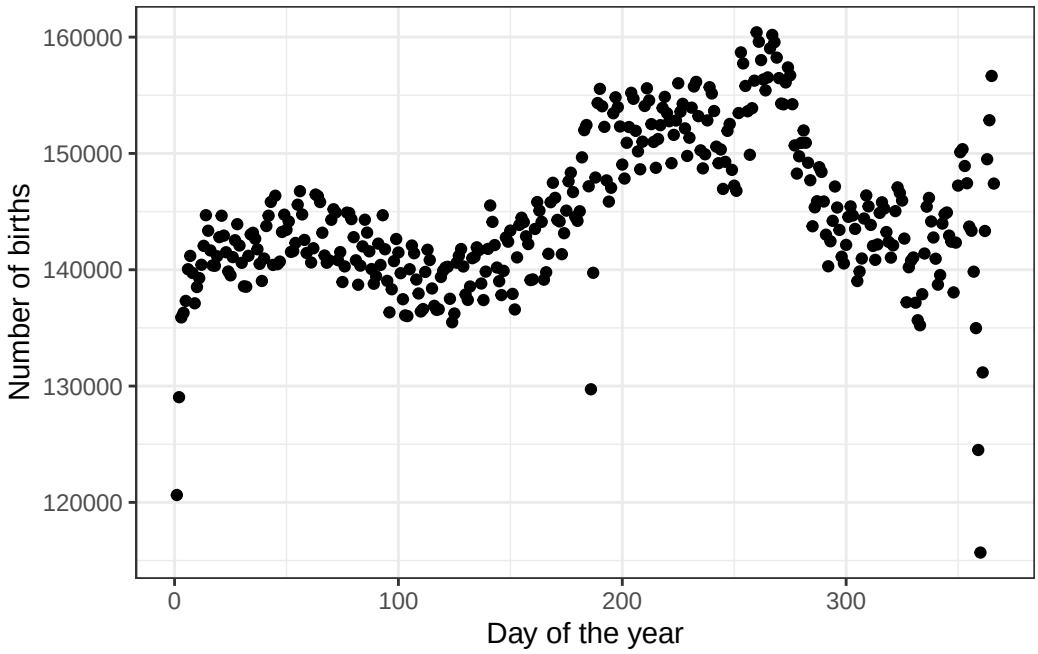


Figure 7.5: Number of births by day of the year for all years.

This curve has the seasonal cycling we would expect. The smaller scale cycling is unexpected. Maybe because we are dropping the leap years, we are getting some days appearing in our time interval more frequently on weekends. We leave it to you to investigate this phenomenon.

We use these counts as weights in a sampling process. Days with more births will have a higher probability of being selected. Days such as Christmas and Christmas Eve have a lower probability of being selected. Let's save the weights in an object to use in the `sample()` function.

```
birth_data_weights <- birth_data %>%
  select(n) %>%
  pull()
```

The `pull()` function pulls the vectors of values out of the data frame format into a vector format which the `sample()` function needs.

Now let's simulate the problem. We will use our code from before, but add probability weights in the `prob` argument. The probability of a match should change slightly, but not much because most of the days have about the same probability or number of occurrences.

```r
set.seed(20)
(do(1000)*length(unique(sample(days, size = 18, replace = TRUE, prob = birth_data_weights)))) %>%
  mutate(match = if_else(length == 18, 0, 1)) %>%
  summarise(prob = mean(match))
```

```
   prob
1 0.352
```

It would not be possible to solve this problem of varying frequency of birthdays using mathematics, at least as far as we know.

This is fascinating stuff! Let's get to learning more about probability models in the following chapters.

8 Probability Rules

8.1 Objectives

1) Differentiate between various statistical terminologies such as *sample space, outcome, event, subset, intersection, union, complement, probability, mutually exclusive, exhaustive, independent, multiplication rule, permutation,* and *combination*, and construct examples to demonstrate their proper use in context.
2) Apply basic probability properties and counting rules to calculate the probabilities of events in different scenarios. Interpret the calculated probabilities in context.
3) Explain and illustrate the basic axioms of probability.
4) Use `R` to perform calculations and simulations for determining the probabilities of events.

8.2 Probability vs Statistics

Remember this book is divided into four general blocks: data collection/summary, probability models, inference and statistical modeling/prediction. This second block on probability models is the study of stochastic (random) processes and their properties. Specifically, we will explore random experiments. As the name suggests, a random experiment is an experiment whose outcome is not predictable with exact certainty. In the statistical models we develop in the last two blocks of this book, we will use other variables to explain the variance of the outcome of interest. Any remaining variance is modeled with probability models.

Even though an outcome is determined by chance, this does not mean that we know nothing about the random experiment. Our favorite simple example is that of a coin flip. If we flip a coin, the possible outcomes are heads and tails. We don't know for sure what outcome will occur, but we still know something about the experiment. If we assume the coin is fair, we know that each outcome is equally likely. Also, we know that if we flip the coin 100 times (independently), we are likely to see around 50 heads, and very unlikely to see 10 heads or fewer.

It is important to distinguish probability from inference and modeling. In probability, we consider a known random experiment, including knowing the parameters, and answer questions

about what we expect to see from this random experiment. In statistics (inference and modeling), we consider data (the results of a mysterious random experiment) and infer about the underlying process. For example, suppose we have a coin and we are unsure whether this coin is fair or unfair (i.e., the parameter is unknown). We might flip it 20 times and observe it land on heads 14 times. Inferential statistics will help us answer questions about the underlying process (e.g., could this coin be unfair?).

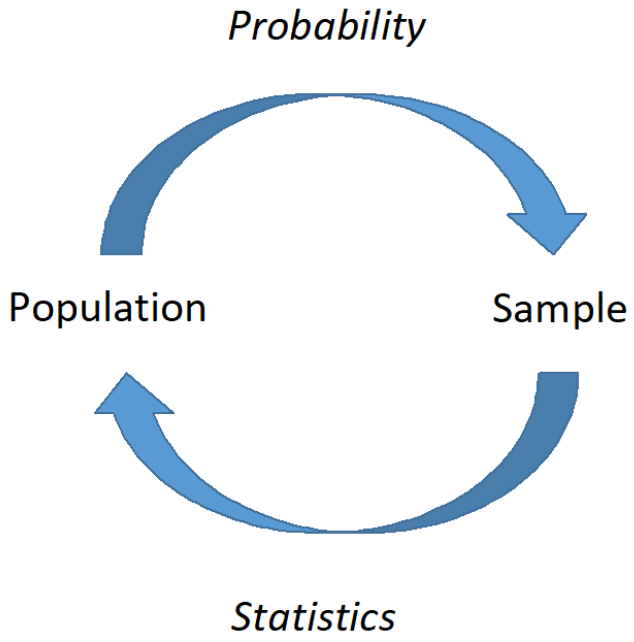


Figure 8.1: A graphical representation of probability and statistics. In probability, we describe what we expect to happen if we know about the underlying process. In statistics, we don't know the underlying process, and must infer based on representative samples.

This block (10 chapters or so) is devoted to the study of random experiments. First, we will explore simple experiments, counting rule problems, and conditional probability. Next, we will introduce the concept of a random variable and the properties of random variables. Following this, we will cover common distributions of discrete and continuous random variables. We will end the block on multivariate probability (joint distributions and covariance).

8.3 Basic probability terms

We will start our work with some definitions and examples.

8.3.1 Sample space

Suppose we have a random experiment. The *sample space* of this experiment, S, is the set of all possible results of that experiment. For example, in the case of a coin flip, we could write $S = \{H, T\}$ because the sample space has two outcomes, heads and tails. Each element of the sample space is considered an *outcome*. An *event* is a set of outcomes, a subset of the sample space.

> *Example*:
> Let `R` flip a coin for us and record the number of heads and tails. We will have `R` flip the coin twice. What is the sample space, what is an example of an outcome, and what is an example of an event?

We will load the **mosaic** package as it has a function `rflip()` that will simulate flipping a coin.

```
library(mosaic)
```

```
set.seed(18)
rflip(2)
```

```
Flipping 2 coins [ Prob(Heads) = 0.5 ] ...

H H

Number of Heads: 2 [Proportion Heads: 1]
```

The sample space is $S = \{HH, TH, HT, TT\}$, an example of an outcome is HH which we see in the output from `R`, and an example of an event is two tails, TT. Another example of an event is "at least one heads", $\{HH, TH, HT\}$. Also, notice that TH is different from HT as an outcome; this is because those are different outcomes from flipping a coin twice.

> *Example of Event*:
> Suppose you arrive at a rental car counter and they show you a list of available vehicles, and one is picked for you at random. The sample space in this experiment is
>
> $S = \{\text{red sedan}, \text{blue sedan}, \text{red truck}, \text{grey truck}, \text{grey SUV}, \text{black SUV}, \text{blue SUV}\}$.
>
> Each vehicle represents a possible outcome of the experiment.

8.3.2 Union and intersection

Suppose we have two events, A and B.

1) A is considered a *subset* of B if all of the outcomes of A are also contained in B. This is denoted as $A \subset B$.

2) The *intersection* of A and B is all of the outcomes contained in both A and B. This is denoted as $A \cap B$.

3) The *union* of A and B is all of the outcomes contained in either A or B, or both. This is denoted as $A \cup B$.

4) The *complement* of A is all of the outcomes not contained in A. This is denoted as A^C or A'.

Note: Here we are treating events as sets and the above definitions are basic set operations.

It is sometimes helpful when reading probability notation to think of Union as an *or* and Intersection as an *and*.

> *Example*:
> Consider our rental car example above. Let A be the event that a blue vehicle is selected, let B be the event that a black vehicle is selected, and let C be the event that an SUV is selected.

First, let's list all of the outcomes of each event. $A = \{\text{blue sedan}, \text{blue SUV}\}$, $B = \{\text{black SUV}\}$, and $C = \{\text{grey SUV}, \text{black SUV}, \text{blue SUV}\}$.

Because all outcomes in B are contained in C, we know that B is a subset of C. This can be written as $B \subset C$. Also, because A and B have no outcomes in common, $A \cap B = \emptyset$. Note that $\emptyset = \{\}$ is the empty set and contains no elements.[1] Further, $A \cup C = \{\text{blue sedan}, \text{grey SUV}, \text{black SUV}, \text{blue SUV}\}$. The complement of C is $C' = \{\text{red sedan, red truck, grey truck}\}$.

8.4 Probability

Probability is a number assigned to an event or outcome that describes how likely it is to occur. A probability model assigns a probability to each element of the sample space. What makes a probability model is not just the values assigned to each element but the idea that this model contains all the information about the outcomes and there are no other explanatory variables involved.

[1] We call events A and B *mutually exclusive*. We'll define this term a little later in the chapter.

A probability model can be thought of as a function that maps outcomes, or events, to a real number in the interval $[0, 1]$.

8.4.1 Probability axioms

There are some basic axioms of probability you should know, although this list is not complete. Let S be the sample space of a random experiment and let A be an event where $A \subset S$.

1) $\mathrm{P}(A) \geq 0$.
2) $\mathrm{P}(S) = 1$.

These two axioms essentially say that probability must be positive, and the probability of all outcomes must sum to one.

8.4.2 Probability properties

Let A and B be events in a random experiment. Most of the probabilities below can be proven fairly easily.

1) $\mathrm{P}(\emptyset) = 0$.
2) $\mathrm{P}(A') = 1 - \mathrm{P}(A)$. We used this in the case study.
3) If $A \subset B$, then $\mathrm{P}(A) \leq \mathrm{P}(B)$.
4) $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cap B)$. This property can be generalized to more than two events. The intersection is subtracted because outcomes in both events A and B get counted twice in the first sum.
5) Law of Total Probability: Let $B_1, B_2, ..., B_n$ be **mutually exclusive**, this means disjoint or no outcomes in common, and **exhaustive**, this means the union of all the events labeled with a B is the sample space. Then

$$\mathrm{P}(A) = \mathrm{P}(A \cap B_1) + \mathrm{P}(A \cap B_2) + ... + \mathrm{P}(A \cap B_n)$$

A specific application of this law appears in Bayes' Rule (more to follow in Chapter **??**). It says that $\mathrm{P}(A) = \mathrm{P}(A \cap B) + \mathrm{P}(A \cap B')$. Essentially, it points out that A can be partitioned into two parts: 1) everything in A and B and 2) everything in A and not in B.

Example:
Consider rolling a six sided die. Let event A be that a number less than five is showing on the die. Let event B be that the number is even. Then,

$$\mathrm{P}(A) = \mathrm{P}(A \cap B) + \mathrm{P}(A \cap B')$$

$$\mathrm{P}(< 5) = \mathrm{P}(< 5 \cap \mathrm{Even}) + \mathrm{P}(< 5 \cap \mathrm{Odd})$$

6) DeMorgan's Laws:

$$\mathrm{P}((A \cup B)') = \mathrm{P}(A' \cap B')$$
$$\mathrm{P}((A \cap B)') = \mathrm{P}(A' \cup B')$$

Exercise:
Let A, B, and C be events such that $P(A) = 0.5$, $P(B) = 0.3$, and $P(C) = 0.4$. Also, we know that $P(A \cap B) = 0.2$, $P(B \cap C) = 0.12$, $P(A \cap C) = 0.1$, and $P(A \cap B \cap C) = 0.05$. Find the following:

- $P(A \cup B)$
- $P(A \cup B \cup C)$
- $P(B' \cap C')$
- $P(A \cup (B \cap C))$
- $P((A \cup B \cup C) \cap (A \cap B \cap C)')$

It can be a good idea to draw a picture to work through exercises like these. Figure **??** is an illustration of the above probability problem. Letters A, B, and C denote the corresponding events, represented by the entire circle in which they reside. The letter S represents the entire sample space. The numbers in the diagram represent the probability of each exclusive event. For example, the value 0.25 represents the probability of only event A (and no other events). Can you figure out where the other probability values come from?[2]

[2]The easiest way to create this illustration is to start from the middle intersection piece. $P(A \cap B \cap C) = 0.05$, represented by the 0.05 at the center of the diagram. $P(A \cap B) = 0.2$, which leaves 0.15 for the rest of the $A \cap B$ piece of the diagram. Because $P(A \cap C) = 0.1$, the remaining piece of $A \cap C$ not yet accounted for is 0.05. Then the remaining piece of A is 0.25. Thus, $P(A)$ can be found by summing all the pieces in the A circle, which adds up to 0.5. Make sure you can track how the rest of the diagram is appropriately labeled.

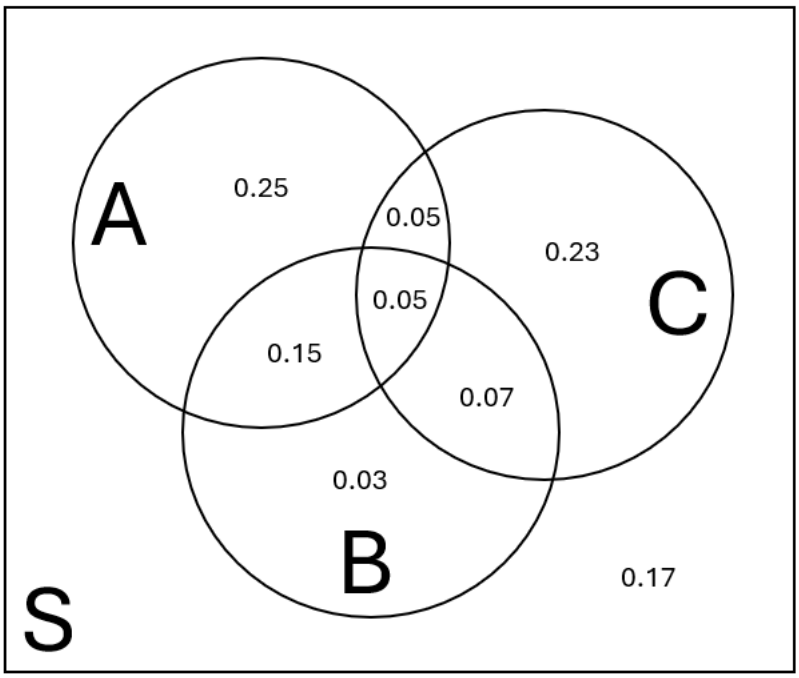


Figure 8.2: Probability illustration.

8.4.3 Equally likely scenarios

In some random experiments, outcomes can be defined such that each individual outcome is equally likely. In this case, probability becomes a counting problem. Let A be an event in an experiment where each outcome is equally likely.

$$\mathrm{P}(A) = \frac{\text{Number of outcomes in A}}{\text{Number of outcomes in S}}$$

> *Example*:
> Suppose a family has three children, with each child being either male (M) or female (F). Assume that the likelihood of males and females are equal and **independent**. This is the idea that the probability of the sex of the second child does not change based on the sex of the first child. The sample space can be written as:
>
> $$S = \{\mathrm{MMM}, \mathrm{MMF}, \mathrm{MFM}, \mathrm{FMM}, \mathrm{MFF}, \mathrm{FMF}, \mathrm{FFM}, \mathrm{FFF}\}$$
>
> What is the probability that the family has exactly 2 female children?

This only happens in three ways: MFF, FMF, and FFM. Thus, the probability of exactly 2 females is 3/8 or 0.375.

8.4.4 Simulation with `R` (Equally likely scenarios)

The previous example above is an example of an "Equally Likely" scenario, where the sample space of a random experiment contains a list of outcomes that are equally likely. In these cases, we can sometimes use `R` to list out the possible outcomes and count them to determine probability. We can also use `R` to simulate the scenario.

> *Example*:
> Use `R` to simulate the family of three children where each child has the same probability of being male or female.

Instead of writing our own function, we can use `rflip()` in the **mosaic** package. We will let H stand for female.

First simulate one family.

```
set.seed(73)
rflip(3)
```

```
Flipping 3 coins [ Prob(Heads) = 0.5 ] ...
```

```
T T H

Number of Heads: 1 [Proportion Heads: 0.333333333333333]
```

In this case, we got 1 female. Next, we will use the `do()` function to repeat this simulation.

```
results <- do(10000)*rflip(3)
head(results)
```

```
  n heads tails      prop
1 3     1     2 0.3333333
2 3     3     0 1.0000000
3 3     3     0 1.0000000
4 3     3     0 1.0000000
5 3     1     2 0.3333333
6 3     1     2 0.3333333
```

Next, we can visualize the distribution of the number of females, heads, in Figure Figure **??**.

```
results %>%
  gf_bar(~heads) %>%
  gf_theme(theme_bw()) %>%
  gf_labs(x = "Number of females", y = "Count")
```

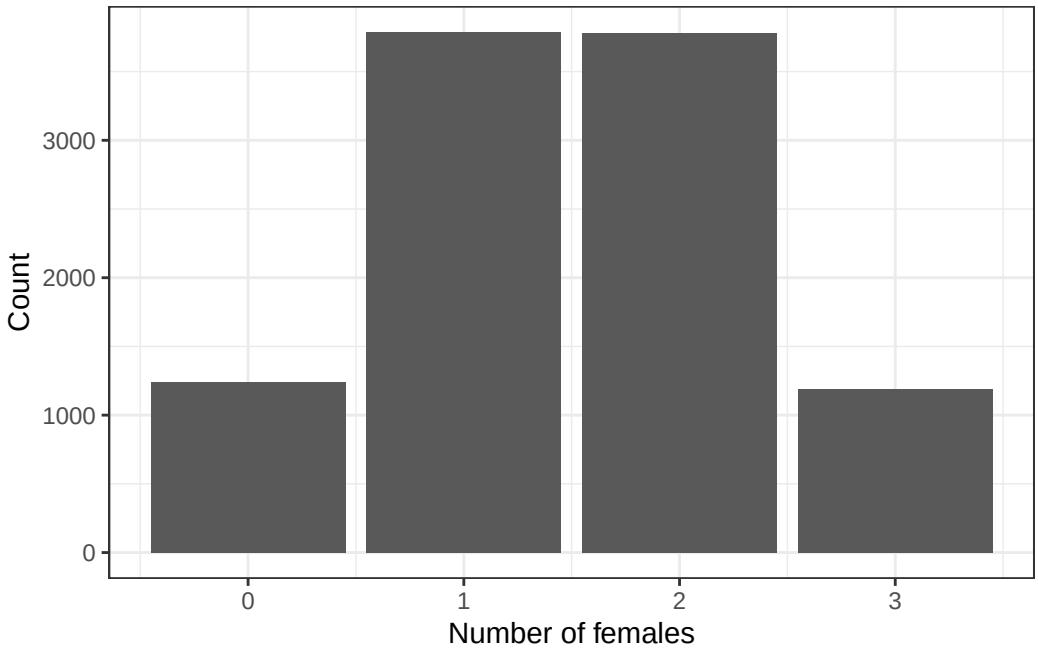


Figure 8.3: Number of females in a family of size 3.

Finally, we can estimate the probability of exactly 2 females. We need the **tidyverse** library.

```
library(tidyverse)
```

```
results %>%
  filter(heads == 2) %>%
  summarize(prob = n()/10000)
```

```
    prob
1 0.3782
```

Or we can use slightly different code.

```
results %>%
  count(heads) %>%
  mutate(prop = n/sum(n))
```

```
  heads    n   prop
1     0 1241 0.1241
```

```
2     1 3786 0.3786
3     2 3782 0.3782
4     3 1191 0.1191
```

This is not a bad estimate of the exact probability, 0.375.

Let's now use an example of cards to simulate some probabilities and learn more about counting. The file `Cards.csv` contains the data for cards from a standard 52-card deck. There are four suits and two colors: spades and clubs are black, hearts and diamonds are red. Within each suit, there are 13 card values or ranks. These include ace, king, queen, jack, and numbers ranging from ten down to two. Let's read in the data and summarize.

```
Cards <- read_csv("data/Cards.csv")
inspect(Cards)
```

```
categorical variables:
  name     class levels  n missing
1 rank character     13 52       0
2 suit character      4 52       0
                                  distribution
1 10 (7.7%), 2 (7.7%), 3 (7.7%) ...
2 Club (25%), Diamond (25%) ...

quantitative variables:
   name   class        min         Q1     median         Q3        max
1 probs numeric 0.01923077 0.01923077 0.01923077 0.01923077 0.01923077
        mean sd  n missing
1 0.01923077  0 52       0
```

```
head(Cards)
```

```
# A tibble: 6 x 3
  rank  suit   probs
  <chr> <chr>  <dbl>
1 2     Club  0.0192
2 3     Club  0.0192
3 4     Club  0.0192
4 5     Club  0.0192
5 6     Club  0.0192
6 7     Club  0.0192
```

We can see 4 suits, and 13 ranks, the value on the face of the card.

> *Example*:
> Suppose we draw one card out of a standard deck. Let A be the event that we draw a Club. Let B be the event that we draw a 10 or a face card (Jack, Queen, King or Ace). We can use R to define these events and find probabilities.

Let's find all the Clubs.

```
Cards %>%
  filter(suit == "Club") %>%
  select(rank, suit)
```

```
# A tibble: 13 x 2
   rank  suit
   <chr> <chr>
 1 2     Club
 2 3     Club
 3 4     Club
 4 5     Club
 5 6     Club
 6 7     Club
 7 8     Club
 8 9     Club
 9 10    Club
10 J     Club
11 Q     Club
12 K     Club
13 A     Club
```

So just by counting, we find the probability of drawing a Club is $\frac{13}{52}$ or 0.25. We can also do this with simulation. This may be overkill for this example, but it gets the idea of simulation across.

Remember, ask yourself what we want R to do and what R needs to do this. Below we sample one card from "the deck" (i.e., the data set) 10,000 times.

```
set.seed(573)
results <- do(10000)*sample(Cards, 1)
head(results)
```

```
# A tibble: 6 x 6
  rank  suit     probs orig.id  .row .index
  <chr> <chr>    <dbl> <chr>   <int>  <dbl>
1 2     Diamond 0.0192 14          1      1
2 7     Spade   0.0192 45          1      2
3 6     Spade   0.0192 44          1      3
4 3     Heart   0.0192 28          1      4
5 5     Diamond 0.0192 17          1      5
6 10    Spade   0.0192 48          1      6
```

```
results %>%
  filter(suit == "Club") %>%
  summarize(prob = n()/10000)
```

```
# A tibble: 1 x 1
   prob
  <dbl>
1 0.240
```

```
results %>%
  count(suit) %>%
  mutate(prob = n/sum(n))
```

```
# A tibble: 4 x 3
  suit        n  prob
  <chr>   <int> <dbl>
1 Club     2405 0.240
2 Diamond  2594 0.259
3 Heart    2529 0.253
4 Spade    2472 0.247
```

In 10,000 samples, each suit occurs around 25% of the time, just as we'd expect.

Now let's count the number of outcomes in B.

```
Cards %>%
  filter(rank %in% c(10, "J", "Q", "K", "A")) %>%
  select(rank,suit)
```

```
# A tibble: 20 x 2
   rank  suit
   <chr> <chr>
 1 10    Club
 2 J     Club
 3 Q     Club
 4 K     Club
 5 A     Club
 6 10    Diamond
 7 J     Diamond
 8 Q     Diamond
 9 K     Diamond
10 A     Diamond
11 10    Heart
12 J     Heart
13 Q     Heart
14 K     Heart
15 A     Heart
16 10    Spade
17 J     Spade
18 Q     Spade
19 K     Spade
20 A     Spade
```

So just by counting, we find the probability of drawing a 10 or greater is $\frac{20}{52}$ or 0.3846154.

> **Exercise**:
> Use simulation to estimate the probability of a 10 or higher (10, jack, queen, king).

We can simply examine the previously generated 10,000 samples to answer this question via simulation.

```
results %>%
  filter(rank %in% c(10, "J", "Q", "K", "A")) %>%
  summarize(prob = n()/10000)
```

```
# A tibble: 1 x 1
   prob
  <dbl>
1 0.382
```

This is close to the value of 0.385 we found with just counting the possibilities.

Notice that this code is not robust to change in the number of simulations. If we use a different number of simulations, then we have to adjust the denominator in the `summarize()` function. We can do this by using `mutate()` instead of `filter()`.

```
results %>%
  mutate(face = rank %in% c(10, "J", "Q", "K", "A"))%>%
  summarize(prob = mean(face))
```

```
# A tibble: 1 x 1
   prob
  <dbl>
1 0.382
```

Notice in the `mutate()` function we are creating a new logical variable called `face`. This variable takes on the values of TRUE and FALSE. In the next line, we use a `summarize()` command with the function `mean()`. In `R`, a function that requires numeric input takes a logical variable and converts TRUE into 1 and FALSE into 0. Thus, the `mean()` will find the proportion of TRUE values and that is why we report it as a probability.

Next, let's find a card that is 10 or greater **and** a club.

```
Cards %>%
  filter(rank %in% c(10, "J", "Q", "K", "A"), suit == "Club") %>%
  select(rank, suit)
```

```
# A tibble: 5 x 2
  rank  suit
  <chr> <chr>
1 10    Club
2 J     Club
3 Q     Club
4 K     Club
5 A     Club
```

By counting, we find the probability of drawing a 10 or greater club is $\frac{5}{52}$ or 0.0961538.

> **Exercise**:
> Simulate drawing one card and estimate the probability of a club that is 10 or greater.

We can again utilize the previously generated 10,000 samples.

```r
results %>%
  mutate(face = (rank %in% c(10, "J", "Q", "K", "A")) & (suit == "Club"))%>%
  summarize(prob = mean(face))
```

```
# A tibble: 1 x 1
    prob
   <dbl>
1 0.0903
```

Again, our simulation results are very close to the exact value found by counting.

8.4.5 Note

We have been using R to count the number of outcomes in an event. This helped us to determine probabilities, but we limited the problems to simple ones. In our cards example, it would be more interesting for us to explore more complex events such as drawing five cards from a standard 52-card deck. Each draw of five cards is equally likely, so in order to find the probability of a flush (five cards of the same suit), we could simply list all the possible flushes and compare that to the entire sample space. Because of the large number of possible outcomes, this becomes difficult. Thus, we need to explore counting rules in more detail to help us solve more complex problems. In this course, we will limit our discussion to three basic cases. You should know that there are entire courses on discrete math and counting rules, so we will still be limited in our methods and the type of problems we can solve in this course.

8.5 Counting rules

There are three types of counting problems we will consider. In each case, we are utilizing (and possibly augmenting) the multiplication rule. All that changes is whether an element is allowed to be reused (replacement), and whether the order of selection matters. This latter question is difficult. Each case will be demonstrated with an example.

8.5.1 Rule 1: Order matters, sample with replacement

The *multiplication rule* is at the center of each of the three methods In this first case, we are using the idea that order matters and items can be reused. This is the multiplication rule without any modifications. Let's use an example to help our understanding.

Example:
A license plate consists of three numeric digits (0-9) followed by three single letters (A-Z). How many possible license plates exist?

We can divide this problem into two sections. In the numeric section, we are selecting 3 objects from 10, *with replacement*. This means that a value can be used more than once. Clearly, *order matters* because a license plate starting with "432" is distinct from a license plate starting with "234". There are $10^3 = 1000$ ways to select the first three digits; 10 for the first, 10 for the second, and 10 for the third.

Question: Why do we multiply and not add these probabilities?[3]

In the alphabet section, we are selecting 3 objects from 26, where again order matters. Thus, there are $26^3 = 17576$ ways to select the last three letters of the plate. Combined, there are $10^3 \times 26^3 = 17576000$ ways to select license plates. Visually,

$$\underbrace{10}_{\text{number}} \times \underbrace{10}_{\text{number}} \times \underbrace{10}_{\text{number}} \times \underbrace{26}_{\text{letter}} \times \underbrace{26}_{\text{letter}} \times \underbrace{26}_{\text{letter}} = 17,576,000$$

Next, we are going to use this new counting method to find a probability.

Exercise:
What is the probability a license plate starts with the number "8" or "0" and ends with the letter "B"?

In order to find this probability, we simply need to determine the number of ways to select a license plate starting with "8" or "0" and ending with the letter "B". We can visually represent this event as:

$$\underbrace{2}_{\text{8 or 0}} \times \underbrace{10}_{\text{number}} \times \underbrace{10}_{\text{number}} \times \underbrace{26}_{\text{letter}} \times \underbrace{26}_{\text{letter}} \times \underbrace{1}_{\text{B}} = 135,200$$

Dividing this number by the total number of possible license plates yields the probability of this event.

```
denom <- 10*10*10*26*26*26
num <- 2*10*10*26*26*1
num / denom
```

```
[1] 0.007692308
```

[3] Multiplication is repeated addition, so in a sense we are adding. For this problem, every value for the first number has 10 possibilities for the second number, and for every value of the second number there are 10 possibilities for the third number. This is an "and" problem and requires multiplication.

The probability of obtaining a license plate starting with "8" or "0" and ending with "B" is 0.0077. Simulating this would be difficult because we would need special functions to check the first number and last letter. This gets into **text mining**, an important subject in data science, but unfortunately we don't have much time in this book for the topic.

8.5.2 Rule 2 (Permutation): Order Matters, Sampling Without Replacement

Consider a random experiment where we sample from a group of size n, without replacement, and the outcome of the experiment depends on the order of the outcomes so order matters. The number of ways to select k objects is given by $n(n-1)(n-2)...(n-k+1)$. This is known as a **permutation** and is sometimes written as

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

Recall that $n!$ is read as "n factorial" and represents the number of ways to arrange n objects.

> *Example*:
> Twenty-five friends participate in a Halloween costume party. Three prizes are given during the party: most creative costume, scariest costume, and funniest costume. No one can win more than one prize. How many possible ways can the prizes by distributed?

There are $k = 3$ prizes to be assigned to $n = 25$ people. Once someone is selected for a prize, they are removed from the pool of eligible prize winners. In other words, we are sampling *without replacement*. Also, *order matters* because there are specific labels on the awards. For example, if Tom, Mike, and Jane win most creative, scariest and funniest costume, respectively, we have a different outcome than if Mike won creative, Jane won scariest and Tom won funniest costume. Thus, the number of ways the prizes can be distributed is given by $_{25}P_3 = \frac{25!}{22!} = 13,800$. A way to visualize this expression is shown as:

$$\underbrace{25}_{\text{most creative}} \times \underbrace{24}_{\text{scariest}} \times \underbrace{23}_{\text{funniest}} = 13,800$$

It is sometimes difficult to determine whether order matters in a problem, but in this example the named prizes were a hint that order matters. This is usually the case when there is some type of label to be attributed to individuals.

Let's use the idea of a permutation to calculate a probability.

> *Exercise*:
> Assume that all 25 participants are equally likely to win any one of the three prizes. What is the probability that Tom doesn't win any of them?

Just like in the previous probability calculation, we simply need to count the number of ways Tom doesn't win any prize. In other words, we need to count the number of ways that prizes are distributed without Tom. So, remove Tom from the group of 25 eligible participants. The number of ways Tom doesn't get a prize is ${}_{24}P_3 = \frac{24!}{21!} = 12,144$. Again visually:

$$\underbrace{24}_{\text{most creative}} \times \underbrace{23}_{\text{scariest}} \times \underbrace{22}_{\text{funniest}} = 12,144$$

The probability Tom doesn't get a prize is simply the second number divided by the first:

```
denom <- factorial(25) / factorial(25 - 3)
# Or, denom <- 25*24*23
num <- 24*23*22
num / denom
```

```
[1] 0.88
```

8.5.3 Rule 3 (Combination): Order Does Not Matter, Sampling Without Replacement

Consider a random experiment where we sample from a group of size n, without replacement, and the outcome of the experiment does not depend on the order of the outcomes (i.e., order does not matter). The number of ways to select k objects is given by $\frac{n!}{(n-k)!k!}$. This is known as a combination and is written as:

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

This is read as "n choose k". Take a moment to compare combinations to permutations, discussed in Rule 2. The difference between these two rules is that in a combination, order no longer matters. A combination is equivalent to a permutation divided by $k!$, the number of ways to arrange the k objects selected.

> *Example*:
> Suppose we draw five cards out of a standard 52-card deck (no jokers). How many possible five card hands are there?

In this example, *order does not matter*. I don't care if I receive 3 jacks then 2 queens or 2 queens then 3 jacks. Either way, it's the same collection of five cards in my hand. Also, we are drawing *without replacement*. Once a card is selected, it cannot be selected again. Thus, the number of ways to select five cards is given by:

$$\binom{52}{5} = \frac{52!}{(52-5)!5!} = 2,598,960$$

Example:
When drawing 5 cards, what is the probability of drawing a "flush" (five cards of the same suit)?

Let's determine how many ways to draw a flush. Recall there are four suits (clubs, hearts, diamonds and spades) and each suit has 13 ranks or values. We would like to pick five of those 13 cards and 0 of the remaining 39. Let's consider just one of those suits (clubs):

$$\text{P}(5 \text{ clubs}) = \frac{\binom{13}{5}\binom{39}{0}}{\binom{52}{5}}$$

The second part of the numerator ($\binom{39}{0}$) isn't necessary, since it simply represents the number of ways to select 0 objects from a group (1 way), but it helps clearly lay out the events. This brings up the point of what 0! equals. By definition it is 1. This allows us to use 0! in our work.

Now, we expand this to all four suits by multiplying by 4, or $\binom{4}{1}$ since we are selecting 1 suit out of the 4:

$$\text{P}(\text{flush}) = \frac{\binom{4}{1}\binom{13}{5}\binom{39}{0}}{\binom{52}{5}}$$

```
num<-4*choose(13,5)*1
denom<-choose(52,5)
num/denom
```

```
[1] 0.001980792
```

There is a probability of 0.0020 of drawing a flush in a draw of five cards from a standard 52-card deck.

Exercise:
When drawing five cards, what is the probability of drawing a "full house" (three cards of the one rank and the two of another)?

This problem uses several ideas from this chapter. We need to pick the rank of the three of a kind. Then pick 3 cards from the 4 possible. Next we pick the rank of the pair from the remaining 12 ranks. Finally pick 2 cards of that rank from the 4 possible.

$$\text{P}(\text{full house}) = \frac{\binom{13}{1}\binom{4}{3}\binom{12}{1}\binom{4}{2}}{\binom{52}{5}}$$

```r
num<-choose(13,1)*choose(4,3)*choose(12,1)*choose(4,2)
denom<-choose(52,5)
num/denom
```

```
[1] 0.001440576
```

Question:
Why can't we use $\binom{13}{2}$ instead of $\binom{13}{1}\binom{12}{1}$?[4]

We have just determined that a full house has a lower probability of occurring than a flush. This is why in gambling, a flush is valued less than a full house.

8.5.4 Summary of counting rules

It can be difficult to remember which counting rule to use in each situation. Remember to consider whether we are sampling with or without replacement, and whether order matters. Figure **??** summarizes the three counting rules we discussed in this chapter.

Order matters	Sampling with replacement: Yes	Sampling with replacement: No
Yes	Multiplication rule (e.g., license plates)	Permutations (e.g., party prizes)
No		Combinations (e.g., drawing cards)

Figure 8.4: Summary of three counting rules

[4]Because this implies the order selection of the ranks does not matter. In other words, this assumes that, for example, three Kings and two fours is the same full house as 3 fours and 2 Kings. This is not true so we break the rank selection about essentially making it a permutation.

9 Conditional Probability

9.1 Objectives

1) Define and differentiate between conditional probability and joint probability, and provide real-world examples to illustrate these concepts and their differences.
2) Calculate conditional probabilities from given data or scenarios using their formal definition, and interpret these probabilities in the context of practical examples.
3) Using conditional probability, determine whether two events are independent and justify your conclusion with appropriate calculations and reasoning.
4) Apply Bayes' Rule to solve problems both mathematically and through simulation usinng `R`.

9.2 Conditional Probability

So far, we've covered the basic axioms of probability, the properties of events (set theory) and counting rules. Another important concept, perhaps one of the most important, is conditional probability. Often, we know a certain event or sequence of events has occurred and we are interested in the probability of another event.

> *Example*:
> Suppose you arrive at a rental car counter and they show you a list of available vehicles, and one is picked for you at random. The sample space in this experiment is
>
> $S = \{\text{red sedan}, \text{blue sedan}, \text{red truck}, \text{grey truck}, \text{grey SUV}, \text{black SUV}, \text{blue SUV}\}$.
>
> What is the probability that a blue vehicle is selected, given a sedan was selected?

Since we know that a sedan was selected, our sample space has been reduced to just "red sedan" and "blue sedan". The probability of selecting a blue vehicle out of this sample space is simply 1/2.

In set notation, let A be the event that a blue vehicle is selected. Let B be the event that a sedan is selected. We are looking for P(A given B), which is also written as P($A|B$). By definition,

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

It is important to distinguish between the event $A|B$ and $A \cap B$. This is a common misunderstanding about probability. $A \cap B$ is the event that an outcome was selected at random from the total sample space, and that outcome was contained in both A and B. On the other hand, $A|B$ assumes the B has occurred, and an outcome was drawn from the remaining sample space, and that outcome was contained in A.

Another common misunderstanding involves the direction of conditional probability. Specifically, $A|B$ is NOT the same event as $B|A$. For example, consider a medical test for a disease. The probability that someone tests positive given they had the disease is different than the probability that someone has the disease given they tested positive. We will explore this example further in our Bayes' Rule section.

9.3 Independence

Two events, A and B, are said to be independent if the probability of one occurring does not change the probability of the other occurring. We looked at this idea in the last chapter, but now we have another way of thinking about it using conditional probabilities. For example, let's say the probability that a randomly selected student has seen the latest superhero movie is 0.55. What if we randomly select a student and we see that he/she is wearing a black backpack? Does that probability change? Likely not, since movie attendance is probably not related to choice of backpack color. These two events are independent.

Mathematically, A and B are considered independent if and only if

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

Result: A and B are independent if and only if

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B)$$

This follows from the definition of conditional probability and from above:

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

Thus, $\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B)$.

Example:
Consider the rental car example from before. Recall events A and B: let A be the event that a blue vehicle is selected and let B be the event that a sedan is selected. Are A and B independent?

No, events A and B are not independent. First, recall that $\mathrm{P}(A|B) = 0.5$. The probability of selecting a blue vehicle ($\mathrm{P}(A)$) is $2/7$ (the number of blue vehicles in our sample space divided by 7, the total number vehicles in S). This value is different from 0.5; thus, A and B are not independent.

We could also use the result above to determine whether A and B are independent. Note that $\mathrm{P}(A) = 2/7$. Also, we know that $\mathrm{P}(B) = 2/7$. So, $\mathrm{P}(A)\mathrm{P}(B) = 4/49$. But, $\mathrm{P}(A \cap B) = 1/7$, since there is just one blue sedan in the sample space. $4/49$ is not equal to $1/7$; thus, A and B are not independent.

9.4 Bayes' Rule

As mentioned in the introduction to this section, $\mathrm{P}(A|B)$ is not the same quantity as $\mathrm{P}(B|A)$. However, if we are given information about $A|B$ and B, we can use Bayes' Rule to find $\mathrm{P}(B|A)$. Let $B_1, B_2, ..., B_n$ be mutually exclusive and exhaustive events and let $\mathrm{P}(A) > 0$. Then,

$$\mathrm{P}(B_k|A) = \frac{\mathrm{P}(A|B_k)\mathrm{P}(B_k)}{\sum_{i=1}^{n} \mathrm{P}(A|B_i)\mathrm{P}(B_i)}$$

Let's use an example to dig into where this comes from.

Example:
Suppose a doctor has developed a blood test for a certain rare disease (only one out of every 10,000 people have this disease). After careful and extensive evaluation of this blood test, the doctor determined the test's sensitivity and specificity.

Sensitivity is the probability of detecting the disease for those who actually have it. We can also write this as the probability of someone testing positive, given they actually have the disease. Note that this is a conditional probability.

Specificity is the probability of correctly identifying "no disease" for those who do not have it. Again, this is a conditional probability.

See Figure **??** for a visual representation of these terms and others related to what is termed a **confusion matrix**.

In fact, this test had a sensitivity of 100% and a specificity of 99.9%. Now suppose a patient walks in, the doctor administers the blood test, and it returns positive. What is the probability the patient actually has the disease?

		Disease		Predictive Value	
		⊕	⊖		
Test	⊕	**A** True Positive (TP)	**B** False Positive (FP)	Positive Predictive Value (PPV) $\frac{TP}{TP+FP} = \frac{A}{A+B}$	Total Positive Results (A + B)
	⊖	**C** False Negative (FN)	**D** True Negative (TN)	Negative Predictive Value (NPV) $\frac{TN}{FN+TN} = \frac{D}{C+D}$	Total Negative Results (C + D)
Sensitivity & Specificity		Sensitivity $\frac{TP}{TP+FN} = \frac{A}{A+C}$	Specificity $\frac{TN}{FP+TN} = \frac{D}{B+D}$		
		All diseased patients (A + C)	All non-diseased patients (B + D)		

Figure 9.1: A table of true results and test results for a hypothetical disease. The terminology is included in the table. These ideas are important when evaluating machine learning classification models.

This is a classic example of how probability could be misunderstood. Upon reading this question, you might guess that the answer to our question is quite high. After all, this is a nearly perfect test. After exploring the problem more in depth, we find a different result.

9.4.1 Approach using whole numbers

Without going directly to the formulaic expression above, let's consider a collection of 100,000 randomly selected people. What do we know?

1) Based on the prevalence of this disease (one out of every 10,000 people have this disease), we know that 10 of them should have the disease.
2) This test is perfectly sensitive. Thus, of the 10 people that have the disease, all of them test positive.
3) This test has a specificity of 99.9%. Of the 99,990 that don't have the disease, $0.999 * 99990 \approx 99890$ will test negative. The remaining 100 will test positive.

Thus, of our 100,000 randomly selected people, 110 will test positive. Of these 110, only 10 actually have the disease. Thus, the probability that someone has the disease given they've tested positive is actually around $10/110 = 0.0909$.

9.4.2 Simulation

To do the simulation, we can think of it as flipping a coin. First let's assume we are pulling 1,000,000 people from the population. The probability that any one person has the disease is 0.0001. We will use `rflip()` to get 1,000,000 people and designate them as "no disease" or "disease".

```
set.seed(43)
results <- rflip(1000000, 0.0001, summarize = TRUE)
results
```

```
      n heads  tails  prob
1 1e+06   100 999900 1e-04
```

In this case, 100 people had the disease (heads). Now, let's find the positive test results. Of the 100 with the disease, all will test positive. Of those without disease, there is a 0.001 probability of testing positive.

```
rflip(as.numeric(results['tails']), prob = 0.001, summarize = TRUE)
```

```
       n heads  tails  prob
1 999900   959 998941 0.001
```

Now, 959 of those without the disease tested positive. Thus, the probability of having the disease given a positive test result is approximately:

```
100 / (100 + 959)   # number with disease divided by total number who tested positive
```

```
[1] 0.09442871
```

9.4.3 Mathematical approach

Now let's put this in context of Bayes' Rule as stated above. First, let's define some events. Let D be the event that someone has the disease. Thus, D' would be the event that someone does not have the disease. Similarly, let T be the event that someone has tested positive. What do we already know?

$$\begin{aligned} \mathrm{P}(D) &= 0.0001 & \mathrm{P}(D') &= 0.9999 \\ \mathrm{P}(T|D) &= 1 & \mathrm{P}(T'|D) &= 0 \\ \mathrm{P}(T'|D') &= 0.999 & \mathrm{P}(T|D') &= 0.001 \end{aligned}$$

We are looking for $\mathrm{P}(D|T)$, the probability that someone has the disease, given he/she has tested positive. By the definition of conditional probability,

$$\mathrm{P}(D|T) = \frac{\mathrm{P}(D \cap T)}{\mathrm{P}(T)}$$

The numerator can be rewritten, again utilizing the definition of conditional probability: $\mathrm{P}(D \cap T) = \mathrm{P}(T|D)\mathrm{P}(D)$.

The denominator can be rewritten using the Law of Total Probability (discussed in Section **??**) and then the definition of conditional probability: $\mathrm{P}(T) = \mathrm{P}(T \cap D) + \mathrm{P}(T \cap D') = \mathrm{P}(T|D)\mathrm{P}(D) + \mathrm{P}(T|D')\mathrm{P}(D')$. So, putting it all together,

$$\mathrm{P}(D|T) = \frac{\mathrm{P}(T|D)\mathrm{P}(D)}{\mathrm{P}(T|D)\mathrm{P}(D) + \mathrm{P}(T|D')\mathrm{P}(D')}$$

Now we have stated our problem in the context of quantities we know:

$$\mathrm{P}(D|T) = \frac{1 \cdot 0.0001}{1 \cdot 0.0001 + 0.001 \cdot 0.9999} = 0.0909$$

Note that in the original statement of Bayes' Rule, we considered n partitions, $B_1, B_2, ..., B_n$. In this example, we only have two: D and D'.

10 Discrete Random Variables

10.1 Objectives

1) Differentiate between various statistical terminologies such as *random variable, discrete random variable, continuous random variable, sample space/support, probability mass function, cumulative distribution function, moment, expectation, mean,* and *variance*, and construct examples to demonstrate their proper use in context.
2) For a given discrete random variable, derive and interpret the probability mass function (pmf) and apply this function to calculate the probabilities of various events.
3) Simulate random variables for a discrete distribution using `R`.
4) Calculate and interpret the moments, such as expected value/mean and variance, of a discrete random variable.
5) Calculate and interpret the expected value/mean and variance of a linear transformation of a random variable.

10.2 Random variables

We have already discussed random experiments. We have also discussed S, the sample space for an experiment. A random variable essentially maps the events in the sample space to the real number line. For a formal definition: A random variable X is a function $X : S \to \mathbb{R}$ that assigns exactly one number to each outcome in an experiment.

Example:
Suppose you flip a coin three times. The sample space, S, of this experiment is

$$S = \{\text{HHH}, \text{HHT}, \text{HTH}, \text{HTT}, \text{THH}, \text{THT}, \text{TTH}, \text{TTT}\}$$

Let the random variable X be the number of heads in three coin flips.

Whenever introduced to a new random variable, we should take a moment to think about what possible values X can take. When flipping a coin three times, we can get zero, one, two, or three heads. The random variable X assigns each outcome in our experiment to one of these values. We can visualize the possible outcomes and possible values of X:

$$S = \{\underbrace{\text{HHH}}_{X=3}, \underbrace{\text{HHT}}_{X=2}, \underbrace{\text{HTH}}_{X=2}, \underbrace{\text{HTT}}_{X=1}, \underbrace{\text{THH}}_{X=2}, \underbrace{\text{THT}}_{X=1}, \underbrace{\text{TTH}}_{X=1}, \underbrace{\text{TTT}}_{X=0}\}$$

The sample space of X is the list of numerical values that X can take:

$$S_X = \{0, 1, 2, 3\}$$

Because the sample space of X is a countable list of numbers, we consider X to be a *discrete* random variable (more on that later).

> *Question:*
> What are some other examples of random variables for this coin flip experiment? What are some counterexamples?[1]

10.2.1 How does this help?

Sticking with our example, we can now frame a problem of interest in the context of our random variable X. For example, suppose we wanted to know the probability of at least two heads. Without our random variable, we have to write this as:

$$\text{P(at least two heads)} = \text{P}(\{\text{HHH}, \text{HHT}, \text{HTH}, \text{THH}\})$$

In the context of our random variable, this simply becomes $\text{P}(X \geq 2)$. It may not seem important in a case like this, but imagine if we were flipping a coin 50 times and wanted to know the probability of obtaining at least 30 heads. It would not be feasible to write out all possible ways to obtain at least 30 heads. It is much easier to write $\text{P}(X \geq 30)$ and then explore the distribution of X.

Essentially, a random variable often helps us reduce a complex random experiment to a simple variable that is easy to characterize.

[1] Remember, a random variable must assign each outcome in the experiment to exactly one number. Other examples of random variables when flipping a coin three times are 1) the number of tails in three coin flips, 2) the number of times the result changes (from heads to tails, or from tails to heads), 3) the position of the first head (i.e., does the first head occur on the first, second, or third coin flip?), or 4) the longest run of consecutive heads. Can you think of other examples?

Some counterexamples (or non-examples) of a random variable for this experiment are 1) whether the coin flip results in heads or tails, and 2) the position of each head in the three flips. Because the first example does not map the possible outcomes to a single number, this is not a random variable. The second is also not a random variable because it maps the possible outcomes to multiple numbers (e.g., if a head occurs on the first and third flip ,the result would be 1 and 3).

10.2.2 Discrete versus continuous random variables

A *discrete* random variable has a sample space that consists of a countable set of values. X in our example above is a discrete random variable. Note that "countable" does not necessarily mean "finite". For example, a random variable with a Poisson distribution (a topic for a later chapter) has a sample space of $\{0, 1, 2, ...\}$. This sample space is unbounded, but it is considered *countably* infinite, and thus the random variable would be considered discrete.

A *continuous* random variable has a sample space that is a continuous interval. For example, let Y be the random variable corresponding to the height of a randomly selected individual. Y is a continuous random variable because a person could measure 68.1 inches, 68.2 inches, or perhaps any value in between. Note that when we measure height, our precision is limited by our measuring device, so we are technically "discretizing" height. However, even in these cases, we typically consider height to be a continuous random variable.

A *mixed* random variable is exactly what it sounds like. It has a sample space that is both discrete and continuous. How could such a thing occur? Consider an experiment where a person rolls a standard six-sided die. If it lands on anything other than one, the result of the die roll is recorded. If it lands on one, the person spins a wheel, and the angle in degrees of the resulting spin, divided by 360, is recorded. If our random variable Z is the number that is recorded in this experiment, the sample space of Z is $[0, 1] \cup \{2, 3, 4, 5, 6\}$. We will not be spending much time on mixed random variables in this course. However, they do occur in practice. Consider the job of analyzing bomb error data. If the bomb hits within a certain radius, the error is 0. Otherwise, the error is measured in a radial direction. This is a mixed random variable.

10.2.3 Discrete distribution functions

Now that we have defined a random variable, we need a way to describe its behavior. We will use probabilities for this purpose.

Distribution functions describe the behavior of random variables. We can use these functions to determine the probability that a random variable takes a value or range of values. For discrete random variables, there are two distribution functions of interest: the *probability mass function* (pmf) and the *cumulative distribution function* (cdf).

10.2.4 Probability mass function

Let X be a discrete random variable. The probability mass function (pmf) of X, given by $f_X(x)$, is a function that assigns probability to each possible outcome of X.

$$f_X(x) = \mathrm{P}(X = x)$$

Note that the pmf is a *function*. Functions have input and output. The input of a pmf is any real number. The output of a pmf is the probability that the random variable takes the inputted value. The pmf must follow the axioms of probability described in the Chapter **??**. Primarily,

1) For all $x \in \mathbb{R}$, $0 \leq f_X(x) \leq 1$. That is, for all values of x, the outcome of $f_X(x)$ is between 0 and 1.

2) $\sum_x f_X(x) = 1$, where the x in the index of the sum simply denotes that we are summing across the entire sample space of X. In words, all the probabilities must sum to one.

Example:
Recall our coin flip example again. We flip a coin three times. Let X be the number of heads. We know that X can only take values 0, 1, 2 or 3. But at what probability does it take these three values? We previously listed out the possible outcomes of the experiment and denoted the value of X corresponding to each outcome.

$$S = \{\underbrace{\text{HHH}}_{X=3}, \underbrace{\text{HHT}}_{X=2}, \underbrace{\text{HTH}}_{X=2}, \underbrace{\text{HTT}}_{X=1}, \underbrace{\text{THH}}_{X=2}, \underbrace{\text{THT}}_{X=1}, \underbrace{\text{TTH}}_{X=1}, \underbrace{\text{TTT}}_{X=0}\}$$

Each of these eight outcomes is equally likely (each with a probability of $\frac{1}{8}$). Thus, building the pmf of X becomes a matter of counting the number of outcomes associated with each possible value of X:

$$f_X(x) = \begin{cases} \frac{1}{8}, & x = 0 \\ \frac{3}{8}, & x = 1 \\ \frac{3}{8}, & x = 2 \\ \frac{1}{8}, & x = 3 \\ 0, & \text{otherwise} \end{cases}$$

Note that this function specifies the probability that X takes any of the four values in the sample space (0, 1, 2, and 3). It also specifies that the probability X takes any other value is 0.

Graphically, the pmf is not terribly interesting. The pmf is 0 at all values of X except for 0, 1, 2 and 3, Figure **??** .

Example:
We can use a pmf to answer questions about an experiment. For example, consider the same coin flip context. What is the probability that we flip at least one head? We can write this in the context of X:

$$\text{P(at least one head)} = \text{P}(X \geq 1) = \text{P}(X = 1) + \text{P}(X = 2) + \text{P}(X = 3)$$

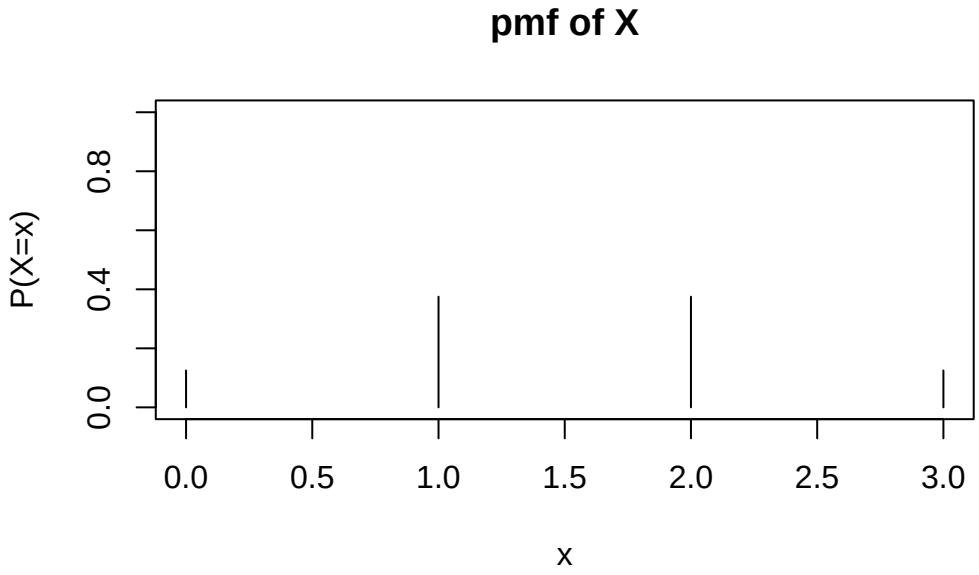


Figure 10.1: Probability Mass Function of X from Coin Flip Example

$$= \frac{3}{8} + \frac{3}{8} + \frac{1}{8} = \frac{7}{8}$$

Alternatively, we can recognize that $\mathrm{P}(X \geq 1) = 1 - \mathrm{P}(X = 0) = 1 - \frac{1}{8} = \frac{7}{8}$.

10.2.5 Cumulative distribution function

Let X be a discrete random variable. The cumulative distribution function (cdf) of X, given by $F_X(x)$, is a function[2] that assigns to each value of X the probability that X takes that value or lower:

$$F_X(x) = \mathrm{P}(X \leq x)$$

If we know the pmf, we can obtain the cdf:

$$F_X(x) = \mathrm{P}(X \leq x) = \sum_{y \leq x} f_X(y)$$

The main idea behind the cdf is a summation of probabilities of interest. In this course, we de-emphasize the derivation of and symbolic notation for the cdf. We instead focus on using the idea of a cdf to calculate probabilities.

[2]Again, note that the cdf is a *function* with an input and output. The input of a cdf is any real number. The output of a cdf is the probability that the random variable takes the inputted value *or less*.

Like the pmf, the value of the cdf must be between 0 and 1. Also, since the pmf is always non-negative, the cdf must be non-decreasing.