

Data Science for Biological, Medical and Health Research: Notes for PQHS 431

Thomas E. Love, Ph.D.

Version: 2019-11-20 23:07:31

Contents

Introduction	13
Structure	14
Course Philosophy	14
Working with this Document	15
1 Data Science	17
1.1 Why a unicorn?	17
1.2 Data Science Project Cycle	18
1.3 What Will We Discuss in 431?	19
2 Setting Up R	21
2.1 R Markdown	21
2.2 R Packages	22
2.3 Other Packages	22
Part A. Exploring Data	27
3 Visualizing Data	27
3.1 The NHANES data: Collecting a Sample	27
3.2 Age and Height	28
3.3 Subset of Subjects with Known Age and Height	29
3.4 Age-Height and Sex?	31
3.5 A New Subset: Ages 21-79	34
3.6 Distribution of Heights	35
3.7 Height and Sex	38
3.8 Looking at Pulse Rate	43
3.9 General Health Status	51
3.10 Conclusions	59
4 Data Structures and Types of Variables	61
4.1 Data require structure and context	61
4.2 A New NHANES Adult Sample	62
4.3 Types of Variables	67

4.4	Creating a Data Report Quickly	70
5	Summarizing Quantitative Variables	71
5.1	The <code>summary</code> function for Quantitative data	71
5.2	Measuring the Center of a Distribution	72
5.3	Measuring the Spread of a Distribution	76
5.4	Measuring the Shape of a Distribution	80
5.5	More Detailed Numerical Summaries for Quantitative Variables .	82
6	Summarizing Categorical Variables	87
6.1	The <code>summary</code> function for Categorical data	87
6.2	Tables to describe One Categorical Variable	88
6.3	The Mode of a Categorical Variable	89
6.4	<code>describe</code> in the <code>Hmisc</code> package	90
6.5	Cross-Tabulations	92
6.6	Constructing Tables Well	96
7	NHANES National Youth Fitness Survey (<code>nnyfs</code>)	99
7.1	The Variables included in <code>nnyfs</code>	100
7.2	Looking over the Data Set	102
7.3	Basic Numerical Summaries	113
7.4	Additional Summaries from <code>favstats</code>	114
7.5	The Histogram	114
7.6	The Stem-and-Leaf	119
7.7	The Dot Plot to display a distribution	123
7.8	The Frequency Polygon	123
7.9	Plotting the Probability Density Function	124
7.10	The Boxplot	125
7.11	A Simple Comparison Boxplot	127
7.12	Using <code>describe</code> in the <code>psych</code> library	130
7.13	Assessing Skew	131
7.14	Assessing Kurtosis (Heavy-Tailedness)	132
7.15	The <code>describe</code> function in the <code>Hmisc</code> library	133
7.16	<code>xda</code> from GitHub for numerical summaries for exploratory data analysis	134
7.17	What Summaries to Report	139
8	Assessing Normality	141
8.1	Empirical Rule Interpretation of the Standard Deviation	141
8.2	Describing Outlying Values with Z Scores	142
8.3	Comparing a Histogram to a Normal Distribution	143
8.4	Does a Normal model work well for the <code>waist</code> circumference? .	145
8.5	The Normal Q-Q Plot	147
8.6	Interpreting the Normal Q-Q Plot	148
8.7	Can a Normal Distribution Fit the <code>nnyfs</code> <code>energy</code> data Well? .	156

9 Using Transformations to “Normalize” Distributions	161
9.1 The Ladder of Power Transformations	161
9.2 Using the Ladder	162
9.3 Protein Consumption in the NNYFS data	162
9.4 Can we transform the <code>protein</code> data?	166
9.5 What if we considered all 9 available transformations?	170
9.6 A Simulated Data Set	172
9.7 What if we considered all 9 available transformations?	177
10 Summarizing data within subgroups	181
10.1 Using <code>dplyr</code> and <code>summarise</code> to build a tibble of summary information	181
10.2 Another Example	183
10.3 Boxplots to Relate an Outcome to a Categorical Predictor	185
10.4 Using Multiple Histograms to Make Comparisons	191
10.5 Using Multiple Density Plots to Make Comparisons	192
10.6 Building a Violin Plot	195
10.7 A Ridgeline Plot	197
11 Straight Line Models and Correlation	201
11.1 Assessing A Scatterplot	201
11.2 Correlation Coefficients	208
11.3 The Pearson Correlation Coefficient	208
11.4 A simulated example	209
11.5 Estimating Correlation from Scatterplots	216
11.6 The Spearman Rank Correlation	220
12 Studying Crab Claws (<code>crabs</code>)	227
12.1 Association of Size and Force	229
12.2 The <code>loess</code> smooth	231
12.3 Fitting a Linear Regression Model	235
12.4 Is a Linear Model Appropriate?	237
12.5 Making Predictions with a Model	240
13 The Western Collaborative Group Study	243
13.1 The Western Collaborative Group Study (<code>wcgs</code>) data set	243
13.2 Are the SBPs Normally Distributed?	246
13.3 Describing Outlying Values with Z Scores	249
13.4 Does Weight Category Relate to SBP?	251
13.5 Re-Leveling a Factor	251
13.6 Are Weight and SBP Linked?	254
13.7 SBP and Weight by Arcus Senilis groups?	255
13.8 Linear Model for SBP-Weight Relationship: subjects without Arcus Senilis	257
13.9 Linear Model for SBP-Weight Relationship: subjects with Arcus Senilis	258
13.10 Including Arcus Status in the model	259

13.11 Predictions from these Linear Models	260
13.12 Scatterplots with Facets Across a Categorical Variable	260
13.13 Scatterplot and Correlation Matrices	261
14 Part A: A Few of the Key Points	265
14.1 Key Graphical Descriptive Summaries for Quantitative Data	265
14.2 Key Numerical Descriptive Summaries for Quantitative Data	265
14.3 The Empirical Rule - Interpreting a Standard Deviation	266
14.4 Identifying “Outliers” Using Fences and/or Z Scores	266
14.5 Summarizing Bivariate Associations: Scatterplots and Regression Lines	267
14.6 Summarizing Bivariate Associations With Correlations	267
Part B. Making Comparisons	271
15 Introduction to Part B	271
15.1 Point Estimation and Confidence Intervals	271
15.2 One-Sample Confidence Intervals	271
15.3 Comparing Two Groups	272
15.4 Special Tools for Categorical Data	273
15.5 Our First Three Studies	273
15.6 Data Sets used in Part B	273
15.7 The <code>Love-boost.R</code> script	273
16 The Serum Zinc Study	275
16.1 Serum Zinc Levels in 462 Teenage Males (<code>serzinc</code>)	275
16.2 Our Goal: A Confidence Interval for the Population Mean	276
16.3 Exploratory Data Analysis for Serum Zinc	276
16.4 Inference about a Population Mean from this Sample	279
16.5 Comparison to “Normal” Zinc Levels	280
16.6 Inference about a Population Proportion from this Sample	281
17 Estimating a Population Mean	285
17.1 Defining a Confidence Interval	285
17.2 Estimating the Population Mean from the Serum Zinc data	286
17.3 Confidence vs. Significance Level	287
17.4 The Standard Error of a Sample Mean	287
17.5 The t distribution and Confidence Intervals for μ	288
17.6 Bootstrap Confidence Intervals for μ	295
17.7 Large-Sample Normal Approximation CIs for μ	302
17.8 Wilcoxon Signed Rank Procedure for CIs	304
17.9 General Advice	306
18 Estimating a Population Proportion	307
18.1 A First Example: Serum Zinc in the “Normal” Range?	307

18.2 A Second Example: Ebola Mortality Rates through 9 Months of the Epidemic	313
18.3 Can the Choice of Confidence Interval Method Matter?	315
19 The Ibuprofen in Sepsis Trial	317
19.1 The Ibuprofen in Sepsis Randomized Clinical Trial	317
19.2 Our Key Questions for an Independent Samples Comparison	318
19.3 Exploratory Data Analysis	319
19.4 Estimating the Difference in Population Means	323
19.5 Categorizing the Outcome	324
19.6 Estimating the Difference in Proportions	325
20 Comparing Means/Quantities using Two Independent Samples	327
20.1 t-based CI for population mean difference $\mu_1 - \mu_2$ from Independent Samples	328
20.2 Bootstrap CI for $\mu_1 - \mu_2$ from Independent Samples	331
20.3 Wilcoxon-Mann-Whitney “Rank Sum” CI from Independent Samples	331
20.4 Summary: Specifying A Two-Sample Study Design	332
21 Comparing Rates/Proportions with Two Independent Samples	339
21.1 A First Example: Ibuprofen and Sepsis Trial	339
21.2 Relating a Treatment to an Outcome	340
21.3 Definitions of Probability and Odds	341
21.4 Defining the Relative Risk	341
21.5 Defining the Risk Difference	342
21.6 Defining the Odds Ratio, or the Cross-Product Ratio	342
21.7 Comparing Rates in a 2x2 Table	343
21.8 The <code>twobytwo</code> function in R	343
21.9 Estimating a Rate More Accurately: Use $(x + 1)/(n + 2)$ rather than x/n	345
21.10A Second Example: Ebola Virus Disease Study, again	347
22 A Paired Sample Study: Lead in the Blood of Children	349
22.1 The Lead in the Blood of Children Study	349
22.2 Exploratory Data Analysis for Paired Samples	351
22.3 Looking at Separate Samples: Using <code>pivot_longer</code>	355
22.4 Estimating the Difference in Means with Paired Samples	358
22.5 Comparing Proportions in Paired Samples	360
23 Comparing Means/Quantities using Two Paired Samples	361
23.1 Estimating the Population Mean of the Paired Differences	363
23.2 t-based CI for Population Mean of Paired Differences, μ_d	363
23.3 Bootstrap CI for mean difference using paired samples	367
23.4 Wilcoxon Signed Rank-based CI for paired samples	368
23.5 Choosing a Confidence Interval Approach	369
23.6 Conclusions for the <code>bloodlead</code> study	369

23.7 The Sign test	370
23.8 A More Complete Decision Support Tool: Comparing Means	371
23.9 Paired (Dependent) vs. Independent Samples	372
24 Comparing Proportions/Rates using Two Paired Samples	377
24.1 An Example: Access to Specialty Care and Insurance in Children	377
24.2 What comparison are we making?	380
24.3 Building a Confidence Interval for the McNemar Odds Ratio	381
24.4 Estimating the difference in proportions	385
25 Hypothesis Testing: What is it good for?	389
25.1 Five Steps Required in Completing a Hypothesis Test	390
25.2 Type I and Type II Error: Power and Confidence	390
25.3 The Courtroom Analogy	391
25.4 Significance vs. Importance	391
25.5 Errors in Hypothesis Testing	392
25.6 The Two Types of Hypothesis Testing Errors	392
25.7 The Significance Level, α , is the Probability of a Type I Error	393
25.8 The Probability of avoiding a Type II Error is called Power, symbolized $1-\beta$	393
25.9 Incorporating the Costs of Various Types of Errors	393
25.10 Relation of α and β to Error Types	394
25.11 What does Dr. Love dislike about p values and “statistical significance”?	394
25.12 The ASA Articles in 2016 and 2019 on Statistical Significance and P-Values	395
26 Power and Sample Size Considerations	397
26.1 Sample Size and Power Considerations for a Single-Sample t test	398
26.2 Paired Sample t Tests and Power/Sample Size	403
26.3 A Toy Example	403
26.4 Using the <code>power.t.test</code> function	404
26.5 Changing Assumptions in a Power Calculation	404
26.6 Two Independent Samples: Power for t Tests	408
26.7 A New Example	408
26.8 Power for Independent Sample T tests with Unbalanced Designs	410
26.9 Power and Sample Size for Comparing Two Population Proportions	411
26.10 Tuberculosis Prevalence Among IV Drug Users	412
26.11 Designing a New TB Study	412
26.12 Using <code>power.prop.test</code> for Balanced Designs	413
26.13 How <code>power.prop.test</code> works	413
26.14 Another Scenario	414
26.15 Using the <code>pwr</code> library to assess sample size for Unbalanced Designs	414
26.16 Using <code>pwr.2p2n.test</code> in R	415
27 A Review: Two Examples, Comparing Means	417

CONTENTS	9
27.1 A Study of Battery Life	417
27.2 The Breakfast Study: Does Oat Bran Cereal Lower Serum LDL Cholesterol?	422
27.3 Power, Sample Size and the Breakfast Study	425
28 Comparing 3 or more Population Means: Analysis of Variance	429
28.1 Comparing Gross Motor Quotient Scores by Income Level (3 Categories)	430
28.2 Alternative Procedures for Comparing More Than Two Means	433
28.3 The Analysis of Variance	436
28.4 Interpreting the ANOVA Table	437
28.5 The Residual Standard Error	440
28.6 The Proportion of Variance Explained by the Factor, or η^2	440
28.7 The Regression Approach to Compare Population Means based on Independent Samples	440
28.8 Equivalent approach to get ANOVA Results	442
28.9 The Problem of Multiple Comparisons	443
28.10 What if we consider another outcome, BMI?	446
29 Comparisons with Contingency Tables	451
29.1 Larger Contingency Tables - Testing for Independence	451
29.2 A 2x3 Table: Comparing Response to Active vs. Placebo	451
29.3 Getting the Chi-Square Test Results	453
29.4 Getting a 2x3 Table into R using a .csv file	454
29.5 Turning the Data Frame into a Table That R Recognizes	454
29.6 Collapsing Levels / Categories in a Factor	455
29.7 Accuracy of Death Certificates (A 6x3 Table)	458
29.8 The Pearson Chi-Square Test of Independence	458
29.9 Three-Way Tables: A 2x2xK Table and a Mantel-Haenszel Analysis	460
30 Partial Review to help you prepare for Quiz 2	467
30.1 Review Items 1-7	467
30.2 Review Items 8-9	468
30.3 Review Items 10-13	468
30.4 Review Items 14-15	469
30.5 Answer Sketch for Review Items	469
Part C. Building Models	479
31 Introduction for Part C	479
31.1 Additional Reading	480
31.2 Scatterplots	480
31.3 Correlation Coefficients	481
31.4 Fitting a Linear Model	481
31.5 Building Predictions from a Linear Model	481

31.6 Data Sets for Part C	481
32 Re-Expression, Tukey's Ladder & Box-Cox Plot	483
32.1 "Linearize" The Association between Quantitative Variables	483
32.2 A New Tool: the Box-Cox Plot	483
32.3 A Simulated Example	484
32.4 Checking on a Transformation or Re-Expression	487
33 Dehydration Recovery in Kids: A Small Study	491
33.1 A Scatterplot Matrix	492
33.2 Are the recovery scores well described by a Normal model?	493
34 Simple Regression: Using Dose to predict Recovery	497
34.1 The Scatterplot, with fitted Linear Model	497
34.2 The Fitted Linear Model	498
34.3 The Summary Output	499
34.4 Viewing the complete ANOVA table	504
34.5 Plotting Residuals vs. Fitted Values	505
35 Multiple Regression with the <code>hydrate</code> data	507
35.1 Another Scatterplot Matrix for the <code>hydrate</code> data	507
35.2 A Multiple Regression for <code>recov.score</code>	508
35.3 ANOVA for Sequential Comparison of Models	514
35.4 Standardizing the Coefficients of a Model	516
35.5 Comparing Fits of Several Possible Models for Recovery Score	518
35.6 Comparing Model Fit: The AIC, or Akaike Information Criterion	519
35.7 Comparing Model Fit with the BIC, or Bayesian Information Criterion	519
35.8 Making Predictions for New Data: Prediction vs. Confidence Intervals	520
35.9 Interpreting the Regression Model: Two Key Questions	522
36 Regression Diagnostics	523
36.1 The Four Key Regression Assumptions	524
36.2 The Linearity Assumption	524
36.3 The Independence Assumption	528
36.4 The Constant Variance Assumption	529
36.5 The Normality Assumption	532
36.6 Outlier Diagnostics: Points with Unusual Residuals	532
36.7 Outlier Diagnostics: Identifying Points with Unusually High Leverage	535
36.8 Outlier Diagnostics: Identifying Points with High Influence on the Model	537
36.9 Running a Regression Model While Excluding A Point	540
36.10 Summarizing Regression Diagnostics for 431	541
36.11 Back to <code>hydrate</code> : Residual Diagnostics for Dose + Weight Model	543

36.12	Violated Assumptions: Problematic Residual Plots?	544
36.13	Problems with Linearity	544
36.14	Problems with Non-Normality: An Influential Point	548
36.15	Problems with Non-Normality: Skew	552
37	Model Selection and Out-of-Sample Validation	557
37.1	Using the WCGS Data to predict Cholesterol Level	557
37.2	Separating the Data into a Training and a Test Sample	558
37.3	Stepwise Regression to Select Predictors	559
37.4	AIC, ANOVA and BIC to assess Candidate Models	561
37.5	Comparing Models in the Test Sample (MSPE, MAPE)	562
38	Missing Data Mechanisms and Simple Imputation	565
38.1	A Toy Example	565
38.2	Missing-data mechanisms	568
38.3	Options for Dealing with Missingness	568
38.4	Complete Case (and Available Case) analyses	569
38.5	Single Imputation	569
38.6	Multiple Imputation	570
38.7	Building a Complete Case Analysis	570
38.8	Single Imputation with the Mean or Mode	570
38.9	Doing Single Imputation with <code>simputation</code>	572
39	Dealing with Missing Data	575
39.1	Identifying Missingness	575
39.2	Complete Case Analysis: A model for <code>chol</code>	578
39.3	Using Multiple Imputation to fit our Regression Model	581
39.4	Comparing Two Models After Imputation with <code>pool.compare</code> .	583
40	BMI and Employment: Working with Categorical Predictors	585
40.1	The Data	585
40.2	The “Kitchen Sink” Model	589
40.3	Using Categorical Variables (Factors) as Predictors	590
40.4	Scatterplot Matrix with Categorical Predictors	598
40.5	Residual Plots when we have Categorical Predictors	600
40.6	Stepwise Regression and Categorical Predictors	601
40.7	Pooling Results after Multiple Imputation	603
41	Species Found on the Galapagos Islands	605
41.1	A Little Background	605
41.2	DTDP: A Scatterplot Matrix	608
41.3	Fitting A “Kitchen Sink” Linear Regression model	610
41.4	Finding Confidence Intervals for our Coefficient Estimates	611
41.5	Measuring Collinearity - the Variance Inflation Factor	612
41.6	Global (F) Testing of Overall Significance	612
41.7	Sequential Testing in a Regression Model with ANOVA	614

41.8 An ANOVA table for the Model as a Whole	615
41.9 Assumption Checking for our Galápagos Islands models	616
41.10 My First Plot: Studentized Residuals vs. Fitted Values	616
41.11 Automatic Regression Diagnostics for Model 1	618
41.12 Model 1: Diagnostic Plot 1	618
41.13 Diagnostic Plot 2: Assessing Normality	620
41.14 Diagnostic Plot 3: Assessing Constant Variance	621
41.15 Obtaining Fitted Values and Residuals from a Model	622
41.16 Relationship between Fitted and Observed Values	624
41.17 Standardizing Residuals	625
41.18 Three Types of Residuals	626
42 Influence Measures for Multiple Regression	629
42.1 DFBETAs	630
42.2 Other Available Influence Measures	631
43 Building Predictions from our models	635
43.1 Predictions for a “typical” island	635
43.2 Making a Prediction with New Data	636
44 Standardizing/Rescaling in Regression Models	637
44.1 Scaling Predictors using Z Scores: Semi-Standardized Coefficients	637
44.2 Fully Standardized Regression Coefficients	639
44.3 Robust Standardization of Regression Coefficients	640
44.4 Scaling Inputs by Dividing by 2 Standard Deviations	641
This is Work in Progress	645

Introduction

These Notes provide a series of examples using R to work through issues that are likely to come up in PQHS/CRSP/MPHP 431.

While these Notes share some of the features of a textbook, they are neither comprehensive nor completely original. The main purpose is to give 431 students a set of common materials on which to draw during the course. In class, we will sometimes:

- reiterate points made in this document,
- amplify what is here,
- simplify the presentation of things done here,
- use new examples to show some of the same techniques,
- refer to issues not mentioned in this document,

but what we don't do is follow these notes very precisely. We assume instead that you will read the materials and try to learn from them, just as you will attend classes and try to learn from them. We welcome feedback of all kinds on this document or anything else. Just email us at 431-help at case dot edu, or submit a pull request.

What you will mostly find are brief explanations of a key idea or summary, accompanied (most of the time) by R code and a demonstration of the results of applying that code.

Everything you see here is available to you as HTML or PDF. You will also have access to the R Markdown files, which contain the code which generates everything in the document, including all of the R results. We will demonstrate the use of R Markdown (this document is generated with the additional help of an R package called `bookdown`) and RStudio (the “program” which we use to interface with the R language) in class.

All data and R code related to these notes are available through the course website.

Structure

The Notes, like the 431 course, fall in three main parts.

Part A is about **visualizing data and exploratory data analyses**. These Notes focus on using R to work through issues that arise in the process of exploring data, managing (cleaning and manipulating) data into a tidy format to facilitate useful work downstream, and describing those data effectively with visualizations, numerical summaries, and some simple models.

Part B is about **making comparisons** with data. The Notes discuss the use of R to address comparisons of means and of rates/proportions, primarily. The main ideas include confidence intervals, using the bootstrap and making decisions about power and sample size. We'll also discuss the value (or lack thereof) of p values for assessing hypotheses. Key ideas from Part A that have an impact here include visualizations to check the assumptions behind our inferences, and cleaning/manipulating data to facilitate our comparisons.

Part C is about **building models** with data. The Notes are primarily concerned (in 431) with linear regression models for continuous quantitative outcomes, using one or more predictors. We'll see how to use models to accomplish many of the comparisons discussed in Part B, and make heavy use of visualization and data management tools developed in Part A to assess our models.

Course Philosophy

In developing this course, we adopt a modern approach that places data at the center of our work. Our goal is to teach you how to do truly reproducible research with modern tools. We want you to be able to answer real questions using data and equip you with the tools you need in order to answer those questions well (Çetinkaya-Rundel (2017) has more on a related teaching philosophy.)

The curriculum includes more on several topics than you might expect from a standard graduate introduction to statistics.

- data gathering
- data wrangling
- exploratory data analysis and visualization
- multivariate modeling
- communication

It also nearly completely avoids formalism and is extremely applied - this is most definitely **not** a course in theoretical or mathematical statistics.

There's very little of this:

$$f(x) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

Instead, there's a lot of this:

```
nyfs1 %>%
  group_by(bmi.cat) %>%
  summarise(mean = mean(waist.circ),
            median = median(waist.circ),
            sd = sd(waist.circ))
```

The 431 course is about **getting things done**. It's not a statistics course alone, nor is it a course in computer programming alone. I think of it as a course in **data science**.

Working with this Document

1. This document is broken down into multiple chapters. Use the table of contents at left to navigate.
2. At the top of the document, you'll see icons which you can click to
 - search the document,
 - change the size, font or color scheme of the page, and
 - download a PDF or EPUB (Kindle-readable) version of the entire document.
3. The document is updated occasionally through the semester. Check the Version information above to verify the last update time.

Chapter 1

Data Science

The definition of **data science** can be a little slippery. One current view of data science, is exemplified by Steven Geringer's 2014 Venn diagram.

- The field encompasses ideas from mathematics and statistics and from computer science, but with a heavy reliance on subject-matter knowledge. In our case, this includes clinical, health-related, medical or biological knowledge.
- As Gelman and Nolan (2017) suggest, the experience and intuition necessary for good statistical practice are hard to obtain, and teaching data science provides an excellent opportunity to reinforce statistical thinking skills across the full cycle of a data analysis project.
- The principal form in which computer science (coding/programming) play a role in this course is to provide a form of communication. You'll need to learn how to express your ideas not just orally and in writing, but also through your code.

1.1 Why a unicorn?

Data Science is a **team** activity. Everyone working in data science brings some part of the necessary skillset, but no one person can cover all three areas alone for excellent projects.

[The individual who is truly expert in all three key areas (mathematics/statistics, computer science and subject-matter knowledge) is] a mythical beast with magical powers who's rumored to exist but is never actually seen in the wild.

<http://www.kdnuggets.com/2016/10/battle-data-science-venn-diagrams.html>

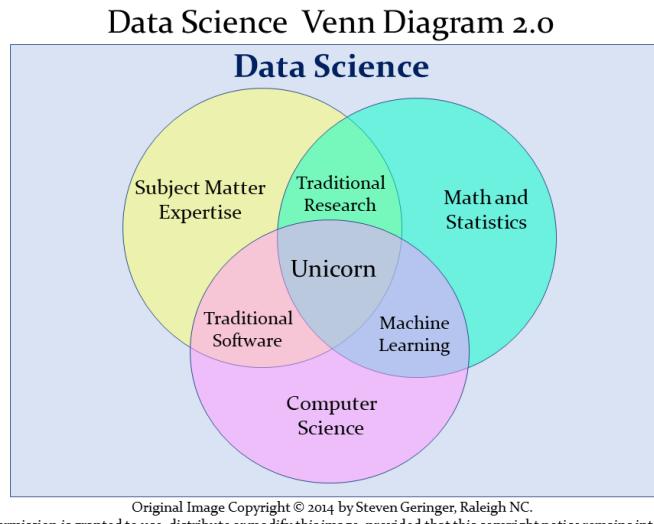


Figure 1.1: Data Science Venn Diagram from Steven Geringer

1.2 Data Science Project Cycle

A typical data science project can be modeled as follows, which comes from the introduction to the amazing book **R for Data Science**, by Garrett Grolemund and Hadley Wickham, which is a key text for this course (Grolemund and Wickham 2019).

This diagram is sometimes referred to as the Krebs Cycle of Data Science. For more on the steps of a data science project, we encourage you to read the Introduction of Grolemund and Wickham (2019).

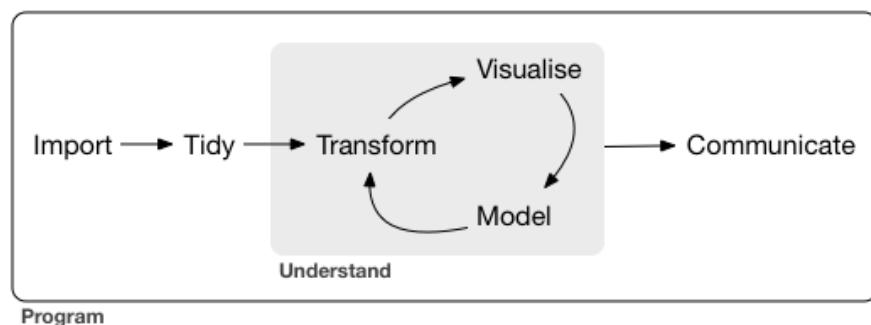


Figure 1.2: Source: R for Data Science: Introduction

1.3 What Will We Discuss in 431?

We'll discuss each of these elements in the 431 course, focusing at the start on understanding our data through transformation, modeling and (especially in the early stages) visualization. In 431, we learn how to get things done.

- We get people working with R and R Studio and R Markdown, even if they are completely new to coding. A gentle introduction is provided at Ismay and Kim (2019)
- We learn how to use the `tidyverse` (<http://www.tidyverse.org/>), an array of tools in R (mostly developed by Hadley Wickham and his colleagues at R Studio) which share an underlying philosophy to make data science faster, easier, more reproducible and more fun. A critical text for understanding the tidyverse is Grolemund and Wickham (2019). Tidyverse tools facilitate:
 - **importing** data into R, which can be the source of intense pain for some things, but is really quite easy 95% of the time with the right tool.
 - **tidying** data, that is, storing it in a format that includes one row per observation and one column per variable. This is harder, and more important, than you might think.
 - **transforming** data, perhaps by identifying specific subgroups of interest, creating new variables based on existing ones, or calculating summaries.
 - **visualizing** data to generate actual knowledge and identify questions about the data - this is an area where R really shines, and we'll start with it in class.
 - **modeling** data, taking the approach that modeling is complementary to visualization, and allows us to answer questions that visualization helps us identify.
 - and last, but definitely not least, **communicating** results, models and visualizations to others, in a way that is reproducible and effective.
- Some programming/coding is an inevitable requirement to accomplish all of these aims. If you are leery of coding, you'll need to get past that, with the help of this course and our stellar teaching assistants. Getting started is always the most challenging part, but our experience is that most of the pain of developing these new skills evaporates by early October.
- Having completed some fundamental work in Part A of the course, we then learn how to use a variety of R packages and statistical methods to accomplish specific inferential tasks (in Part B, mostly) and modeling tasks (in Part C, mostly.)

Chapter 2

Setting Up R

These Notes make extensive use of

- the statistical software language R, and
- the development environment R Studio,

both of which are free, and you'll need to install them on your machine. Instructions for doing so are in found in the course syllabus.

If you need an even gentler introduction, or if you're just new to R and RStudio and need to learn about them, we encourage you to take a look at <http://moderndive.com/>, which provides an introduction to statistical and data sciences via R at Ismay and Kim (2019).

2.1 R Markdown

These notes were written using R Markdown. R Markdown, like R and R Studio, is free and open source.

R Markdown is described as an *authoring framework* for data science, which lets you

- save and execute R code
- generate high-quality reports that can be shared with an audience

This description comes from <http://rmarkdown.rstudio.com/lesson-1.html> which you can visit to get an overview and quick tour of what's possible with R Markdown.

Another excellent resource to learn more about R Markdown tools is the Communicate section (especially the R Markdown chapter) of Grolemund and Wickham (2019).

2.2 R Packages

To start, I'll present a series of commands I run at the beginning of these Notes. These particular commands set up the output so it will look nice as either an HTML or PDF file, and also set up R to use several packages (libraries) of functions that expand its capabilities. A chunk of code like this will occur near the top of any R Markdown work.

```
knitr::opts_chunk$set(comment = NA)

# library(pander); library(pwr)

library(grid); library(devtools);
library(magrittr); library(patchwork);
library(knitr); library(NHANES); library(boot);
library(broom); library(janitor); library(tidyverse)

# source("data/Love-boost.R")
```

I have deliberately set up this list of loaded packages/libraries to be relatively small, and will add some other packages later, as needed. You only need to install a package once, but you need to reload it every time you start a new session.

2.3 Other Packages

I may also make use of functions in the following packages/libraries, but when I do so, I will explicitly specify the package name, using a command like `Hmisc::describe(x)`, rather than just `describe(x)`, so as to specify that I want the Hmisc package's version of `describe` applied to whatever `x` is. Those packages are:

- `aplypack` which provides `stem.leaf` and `stem.leaf.backback` for building fancier stem-and-leaf displays
- `arm` which provides a set of functions for model building and checking that are used in Gelman and Hill (2007)
- `broom` which turns the results lots of different analyses in R into more useful tidy data frames (tibbles.)
- `car` which provides some tools for building scatterplot matrices, but also many other functions described in Fox and Weisberg (2011)
- `cowplot` which is used in Part C to put multiple graphical objects in the same plot, like `gridExtra`: <https://cran.r-project.org/web/packages/cowplot/vignettes/introduction.html>
- `DataExplorer` for generating highly detailed profiles of a data frame

- **Epi** for 2x2 table analyses and materials for classical epidemiology: <http://BendixCarstensen.com/Epi/>
- **exact2x2** for calculating McNemar odds ratios and confidence intervals in paired comparisons of proportions
- **GGally** for scatterplot and correlation matrix visualizations: <http://ggobi.github.io/ggally/>
- **ggridges** which is used to make ridgeline plots
- **gridExtra** which includes a variety of functions for manipulating graphs: <https://github.com/baptiste/gridextra>
- **Hmisc** from Frank Harrell at Vanderbilt U., for its version of **describe** and for many regression modeling functions we'll use in 432. Details on Hmisc are at <http://biostat.mc.vanderbilt.edu/wiki/Main/Hmisc>. Frank has written several books - the most useful of which for 431 students is probably Harrell and Slaughter (2019)
- **mice**, which we'll use (a little) in 431 for multiple imputation to deal with missing data: <http://www.stefvanbuuren.nl/mi/>
- **mosaic**, mostly for its **favstats** summary, but Project MOSAIC is a community of educators you might be interested in: <http://mosaic-web.org/>
- **naniar**, for wrangling and visualizing missingness, and for checking imputations. See <http://naniar.njtierney.com/>.
- **PropCIs** for computing confidence intervals for differences in proportions in paired samples.
- **psych** for its own version of **describe**, but other features are described at <http://personality-project.org/r/psych/>
- **imputation** for some imputation work
- **skimr** for its ability to provide a “skimmed” descriptive analysis of a data set

We'll also use some packages that get loaded via **devtools** and Github by the code in these notes, including:

- **xda** for two functions called **numSummary** and **charSummary**
- **visdat** for two functions called **vis_missandvis_dat**
- **patchwork**, which is a framework for composing **ggplot2** objects (actually this is now loaded above).

Several other packages are included below, even though they are not used in these Notes, because they will be used in class sessions or in 432.

When compiling the Notes from the original code files, these packages will need to be installed (but not loaded) in R, or an error will be thrown when compiling this document. To install all of the packages used within these Notes, type in (or copy and paste) the following commands and run them in the R Console. Again, you only need to install a package once, but you need to reload it every time you start a new session.

```
pkgs <- c("aplypack", "arm", "babynames", "boot", "broom", "car", "cowplot",
        "DataExplorer", "devtools", "Epi", "exact2x2", "faraway", "fivethirtyeight",
```

```
"foreign", "gapminder", "GGally", "ggridges", "gridExtra", "Hmisc",
"janitor", "kableExtra", "knitr", "lme4", "magrittr", "markdown",
"MASS", "mice", "mosaic", "multcomp", "naniar", "NHANES", "pander",
"PropCIs", "psych", "pwr", "qcc", "rmarkdown", "rmdformats", "rms",
"sandwich", "simputation", "skimr", "survival", "tableone",
"tidyverse", "vcd")
```

```
install.packages(pkgs)
```

Part A. Exploring Data

Chapter 3

Visualizing Data

Part A of these Notes is designed to ease your transition into working effectively with data, so that you can better understand it. We'll start by visualizing some data from the US National Health and Nutrition Examination Survey, or NHANES. We'll display R code as we go, but we'll return to all of the key coding ideas involved later in the Notes.

3.1 The NHANES data: Collecting a Sample

To begin, we'll gather a random sample of 1,000 subjects participating in NHANES, and then identify several variables of interest about those subjects¹. Some of the motivation for this example came from a Figure in Baumer, Kaplan, and Horton (2017).

```
# library(NHANES) # already loaded NHANES package/library of functions, data
set.seed(431001)
# use set.seed to ensure that we all get the same random sample
# of 1,000 NHANES subjects in our nh_data collection

nh_dat1 <- sample_n(NHANES, size = 1000) %>%
  select(ID, Gender, Age, Height)

nh_dat1
# A tibble: 1,000 x 4
  ID   Gender   Age   Height
```

¹For more on the NHANES data available in the NHANES package, type ?NHANES in the Console in R Studio.

```

<int> <fct> <int> <dbl>
1 69638 female     5   106.
2 70782 male      64   176.
3 52408 female     54   162.
4 59031 female     15   155.
5 64530 male      53   185.
6 71040 male      63   169.
7 55186 female     30   168.
8 60211 male      5   103.
9 55730 male      66   161.
10 68229 female    36   170.
# ... with 990 more rows

```

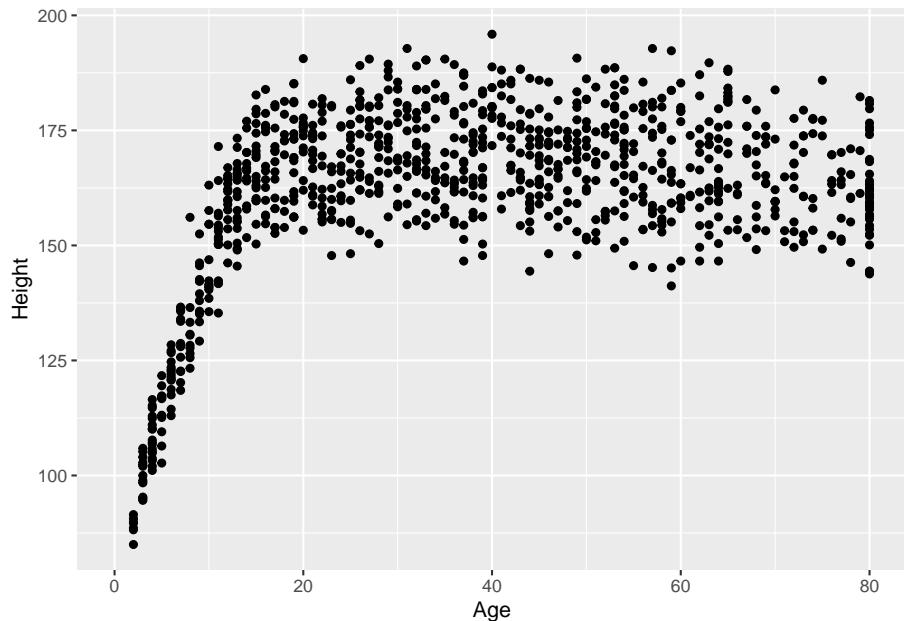
We have 1000 rows (observations) and 4 columns (variables) that describe the subjects listed in the rows.

3.2 Age and Height

Suppose we want to visualize the relationship of Height and Age in our 1,000 NHANES observations. The best choice is likely to be a scatterplot.

```
ggplot(data = nh_dat1, aes(x = Age, y = Height)) +
  geom_point()
```

Warning: Removed 37 rows containing missing values (geom_point).



We note several interesting results here.

1. As a warning, R tells us that it has “Removed 37 rows containing missing values (geom_point).” Only 963 subjects plotted here, because the remaining 37 people have missing (NA) values for either Height, Age or both.
2. Unsurprisingly, the measured Heights of subjects grow from Age 0 to Age 20 or so, and we see that a typical Height increases rapidly across these Ages. The middle of the distribution at later Ages is pretty consistent at around a Height somewhere between 150 and 175. The units aren’t specified, but we expect they must be centimeters. The Ages are clearly reported in Years.
3. No Age is reported over 80, and it appears that there is a large cluster of Ages at 80. This may be due to a requirement that Ages 80 and above be reported at 80 so as to help mask the identity of those individuals.²

As in this case, we’re going to build most of our visualizations using tools from the `ggplot2` package, which is part of the `tidyverse` series of packages. You’ll see similar coding structures throughout this Chapter, most of which are covered as well in Chapter 3 of Grolemund and Wickham (2019).

3.3 Subset of Subjects with Known Age and Height

Before we move on, let’s manipulate the data set a bit, to focus on only those subjects who have complete data on both Age and Height. This will help us avoid that warning message.

```
nh_dat2 <- nh_dat1 %>%
  filter(complete.cases(Age, Height))

summary(nh_dat2)
```

ID	Gender	Age	Height
Min. :51624	female:484	Min. : 2.0	Min. : 85
1st Qu.:57034	male :479	1st Qu.:19.0	1st Qu.:156
Median :62056		Median :37.0	Median :165
Mean :61967		Mean :38.3	Mean :162
3rd Qu.:67269		3rd Qu.:56.0	3rd Qu.:174
Max. :71875		Max. :80.0	Max. :196

Note that the units and explanations for these variables are contained in the NHANES help file, available via typing `?NHANES` in the Console of R Studio, or by typing `NHANES` into the Search bar in R Studio’s Help window.

²If you visit the NHANES help file with `?NHANES`, you will see that subjects 80 years or older were indeed recorded as 80.

3.3.1 The Distinction between Gender and Sex

The `Gender` variable here is a mistake. These data refer to the biological status of these subjects, which is their `Sex`, and not the social construct of `Gender` which can be quite different. In our effort to avoid further confusion, we'll rename the variable `Gender` to instead more accurately describe what is actually measured here.

To do this, we can use this approach...

```
nh_dat2 <- nh_dat1 %>%
  rename(Sex = Gender) %>%
  filter(complete.cases(Age, Height))

summary(nh_dat2)
```

ID	Sex	Age	Height
Min. :51624	female:484	Min. : 2.0	Min. : 85
1st Qu.:57034	male :479	1st Qu.:19.0	1st Qu.:156
Median :62056		Median :37.0	Median :165
Mean :61967		Mean :38.3	Mean :162
3rd Qu.:67269		3rd Qu.:56.0	3rd Qu.:174
Max. :71875		Max. :80.0	Max. :196

That's better. How many observations do we have now? We could use `dim` to find out the number of rows and columns in this new data set.

```
dim(nh_dat2)
```

```
[1] 963 4
```

Or, we could simply list the data set and read off the result.

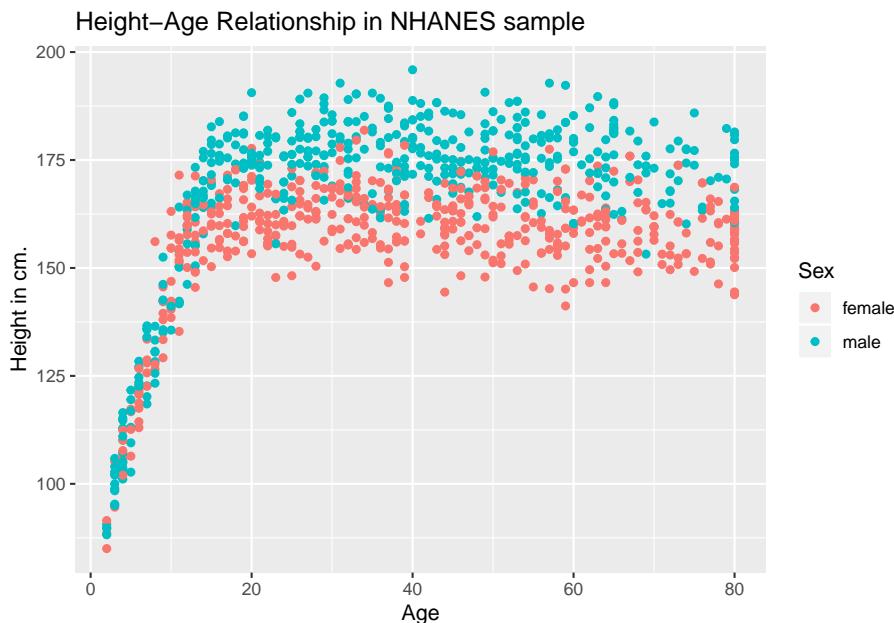
```
nh_dat2
```

```
# A tibble: 963 x 4
  ID   Sex     Age Height
  <int> <fct> <int>  <dbl>
1 69638 female    5   106.
2 70782 male      64   176.
3 52408 female    54   162.
4 59031 female    15   155.
5 64530 male      53   185.
6 71040 male      63   169.
7 55186 female    30   168.
8 60211 male      5   103.
9 55730 male      66   161.
10 68229 female   36   170.
# ... with 953 more rows
```

3.4 Age-Height and Sex?

Let's add Sex to the plot using color, and also adjust the y axis label to incorporate the units of measurement.

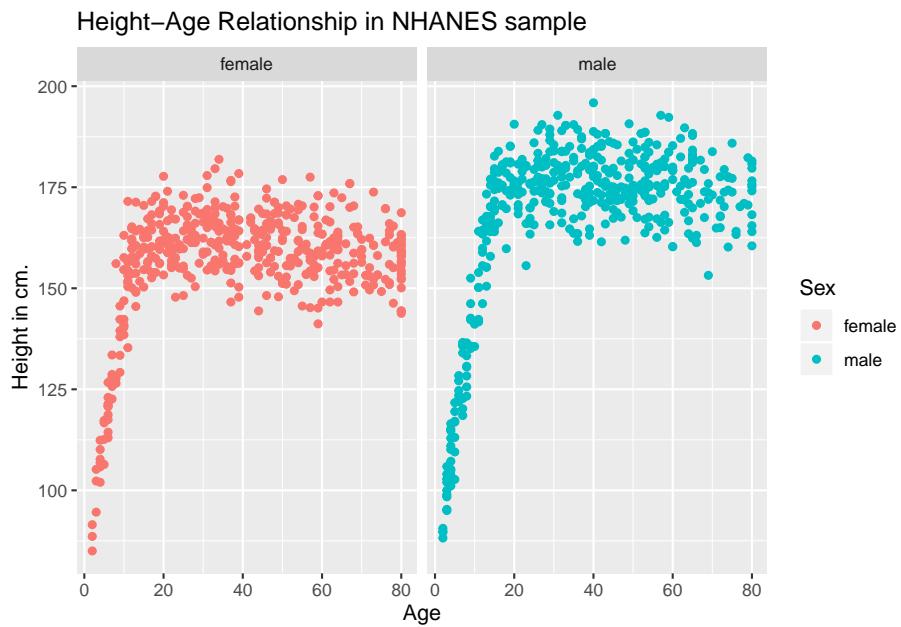
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
  geom_point() +
  labs(title = "Height-Age Relationship in NHANES sample",
       y = "Height in cm.")
```



3.4.1 Can we show the Female and Male relationships in separate panels?

Sure.

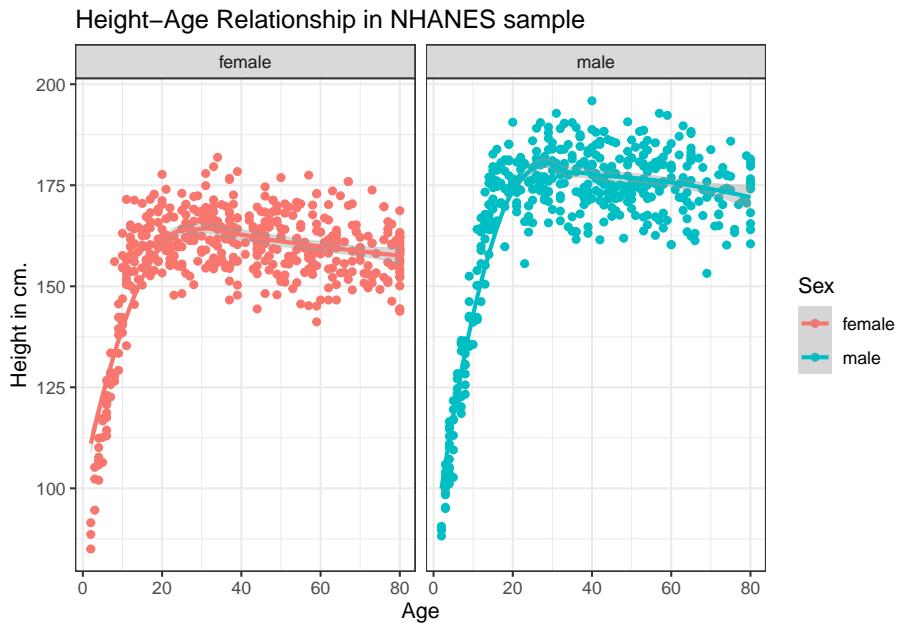
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
  geom_point() +
  labs(title = "Height-Age Relationship in NHANES sample",
       y = "Height in cm.") +
  facet_wrap(~ Sex)
```



3.4.2 Can we add a smooth curve to show the relationship in each plot?

Yep, and let's change the theme of the graph to remove the gray background, too.

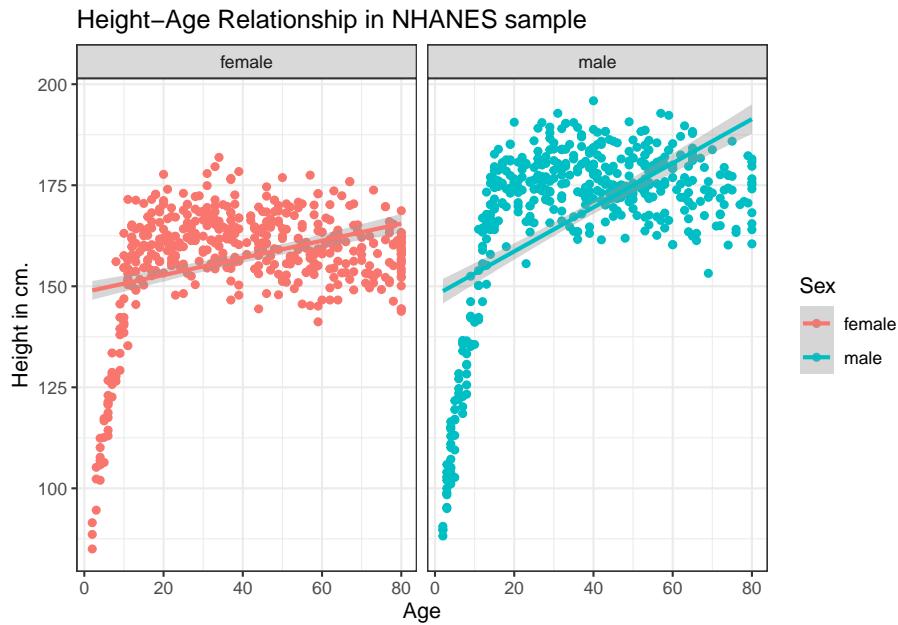
```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Height-Age Relationship in NHANES sample",
       y = "Height in cm.") +
  theme_bw() +
  facet_wrap(~ Sex)
```



3.4.3 What if we want to assume straight line relationships?

We could look at a linear model in the plot. Does this make sense here?

```
ggplot(data = nh_dat2, aes(x = Age, y = Height, color = Sex)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Height-Age Relationship in NHANES sample",
       y = "Height in cm.") +
  theme_bw() +
  facet_wrap(~ Sex)
```



3.5 A New Subset: Ages 21-79

Suppose we wanted to look only at those observations (subjects) whose Age is at least 21 and at most 79. Suppose also that we want to look at some of the additional variables available in NHANES. To start, we'll do the following:

1. Set a seed for the random sampling from the original NHANES data, so we get the same original sample of 1000 people we started with earlier.
2. Select 1000 people from the NHANES data.
3. Filter the sample to only those people whose age is more than 20 and less than 80 years.
4. Select the variables we will be using in the remainder of this chapter, which will be:
 - **Age** as we've seen before, in years.
 - **Height** as we've seen before, in centimeters.
 - **Gender** which we'll rename as **Sex** again.
 - **Pulse** = 60 second pulse rate (in beats per minute).
 - **BPSysAve** = Systolic Blood Pressure, in mm Hg (and we'll rename this SBP).
 - **SleepTrouble** = Yes means the subject has told a health professional that they had trouble sleeping.
 - **PhysActive** = Yes means the subject does moderate or vigorous-intensity sports, fitness or recreational activity.

- **MaritalStatus** = one of Married, Widowed, Divorced, Separated, NeverMarried or LivePartner (living with partner.)
 - **HealthGen** = self-reported rating of general health, one of Excellent, Vgood (Very Good), Good, Fair or Poor.
5. Rename **Gender** as **Sex**, to more accurately describe what is being measured.
 6. Omit subjects with any missingness on *any* of the variables we've selected.

Can you see how the code below accomplishes these tasks?

```
set.seed(431001) # again, this will ensure the same sample

nh_data_2179 <- sample_n(NHANES, size = 1000) %>%
  filter(Age > 20 & Age < 80) %>%
  select(ID, Gender, Age, Height,
         Pulse, BPSysAve, SleepTrouble, PhysActive,
         MaritalStatus, HealthGen) %>%
  rename(Sex = Gender, SBP = BPSysAve) %>%
  na.omit

nh_data_2179

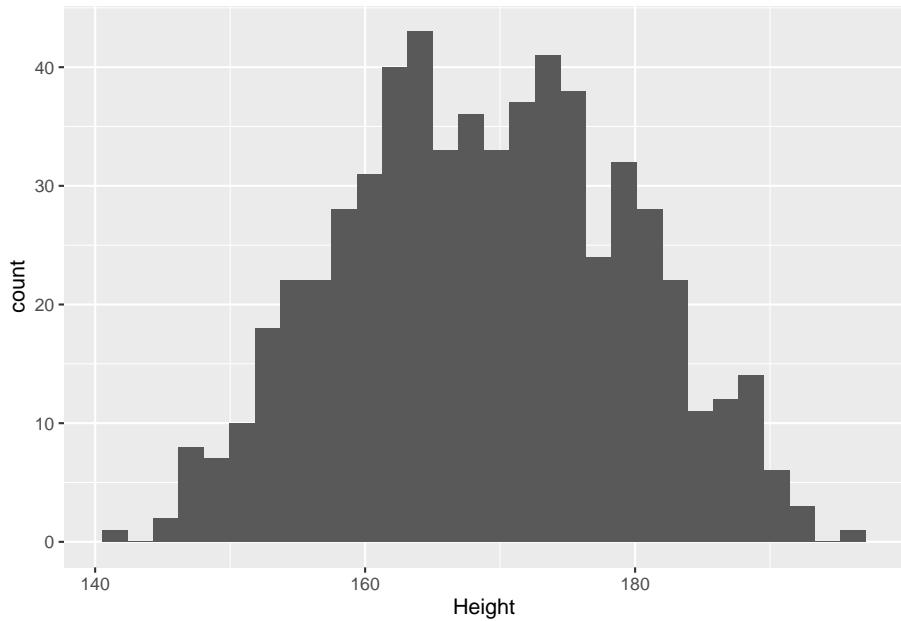
# A tibble: 603 x 10
  ID Sex     Age Height Pulse   SBP SleepTrouble PhysActive
  <int> <fct> <int>  <dbl> <int> <int> <fct>       <fct>
1 70782 male    64    176.    78    127 No        No
2 52408 fema~   54    162.    80    135 No        No
3 64530 male    53    185.    100   131 No        No
4 71040 male    63    169.    70    124 Yes       Yes
5 55186 fema~   30    168.    76    107 No        No
6 55730 male    66    161.    78    133 No        No
7 68229 fema~   36    170.    90    105 No        Yes
8 63762 male    23    180.    66    118 No        No
9 66290 fema~   63    162.    88    116 No        No
10 66984 male   75    174.    84    141 No        No
# ... with 593 more rows, and 2 more variables: MaritalStatus <fct>,
#   HealthGen <fct>
```

3.6 Distribution of Heights

What is the distribution of height in this new sample?

```
ggplot(data = nh_data_2179, aes(x = Height)) +
  geom_histogram()

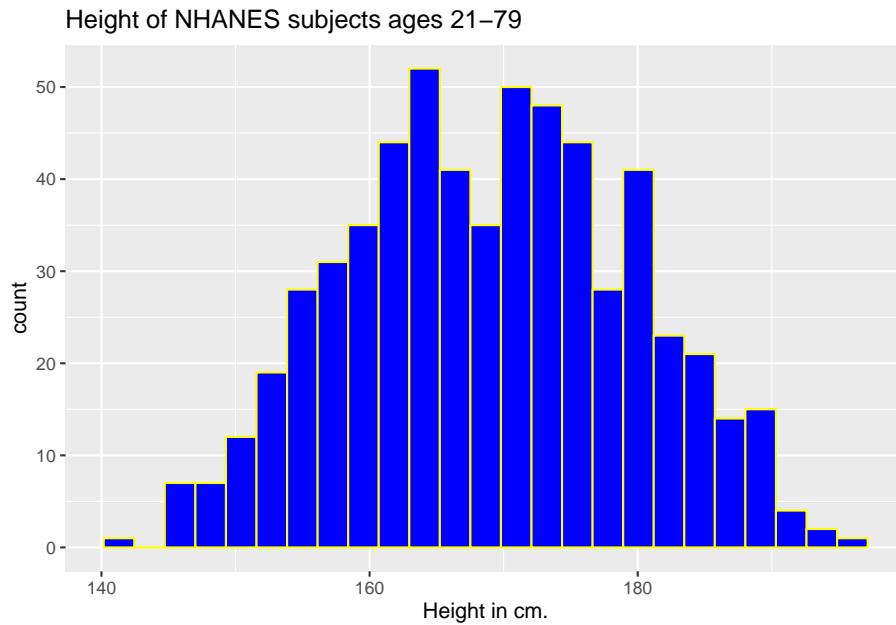
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can do several things to clean this up.

1. We'll change the color of the lines for each bar of the histogram.
2. We'll change the fill inside each bar to make them stand out a bit more.
3. We'll add a title and relabel the horizontal (x) axis to include the units of measurement.
4. We'll avoid the warning by selecting a number of bins (we'll use 25 here) into which we'll group the heights before drawing the histogram.

```
ggpplot(data = nh_data_2179, aes(x = Height)) +
  geom_histogram(bins = 25, col = "yellow", fill = "blue") +
  labs(title = "Height of NHANES subjects ages 21-79",
       x = "Height in cm.")
```

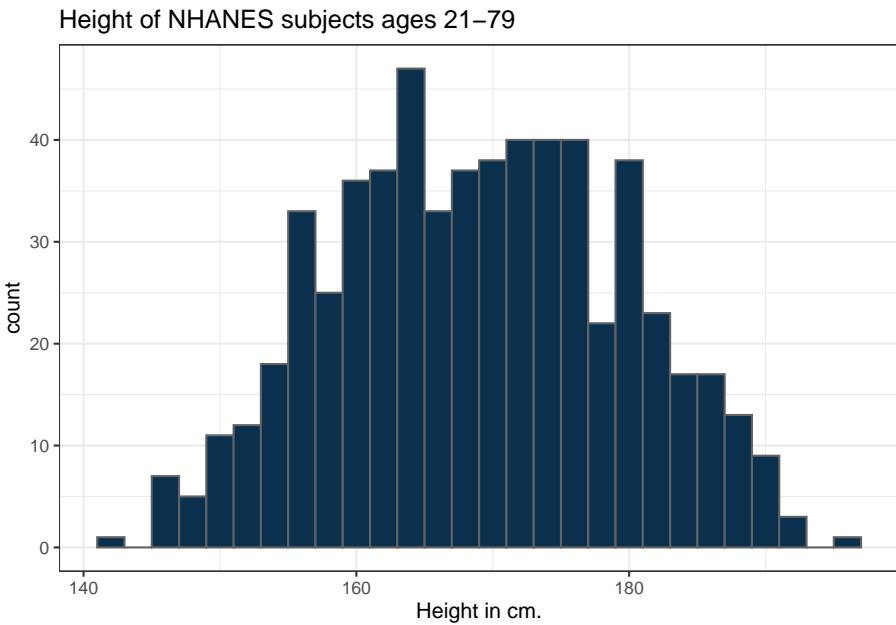


3.6.1 Changing a Histogram's Fill and Color

The CWRU color guide (<https://case.edu/umc/our-brand/visual-guidelines/>) lists the HTML color schemes for CWRU blue and CWRU gray. Let's match that color scheme.

```
cwru.blue <- '#0a304e'
cwru.gray <- '#626262'

ggplot(data = nh_data_2179, aes(x = Height)) +
  geom_histogram(binwidth = 2, col = cwru.gray, fill = cwru.blue) +
  labs(title = "Height of NHANES subjects ages 21-79",
       x = "Height in cm.") +
  theme_bw()
```

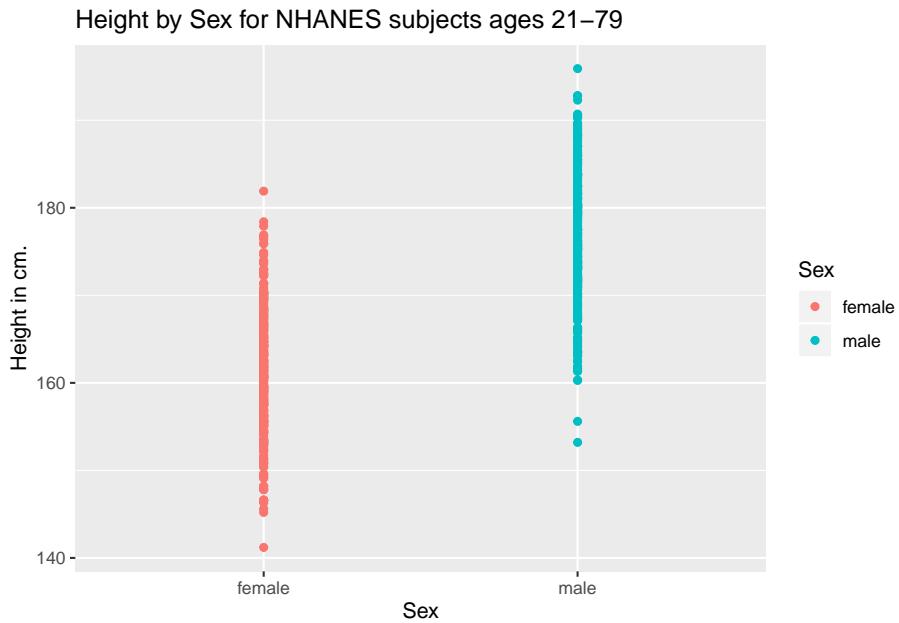


Note the other changes to the graph above.

1. We changed the theme to replace the gray background.
2. We changed the bins for the histogram, to gather observations into groups of 2 cm. each.

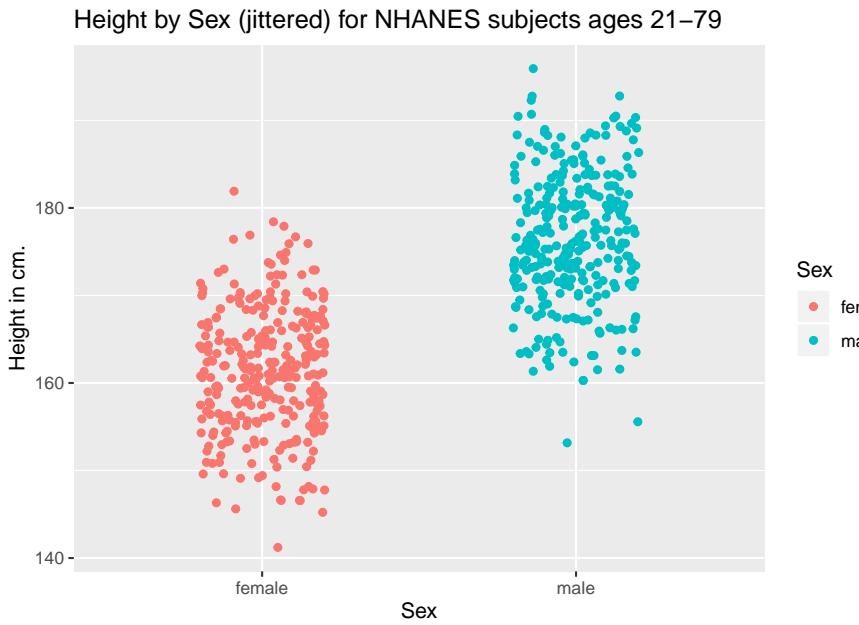
3.7 Height and Sex

```
ggplot(data = nh_data_2179, aes(x = Sex, y = Height, color = Sex)) +
  geom_point() +
  labs(title = "Height by Sex for NHANES subjects ages 21-79",
       y = "Height in cm.")
```



This plot isn't so useful. We can improve things a little by jittering the points horizontally, so that the overlap is reduced.

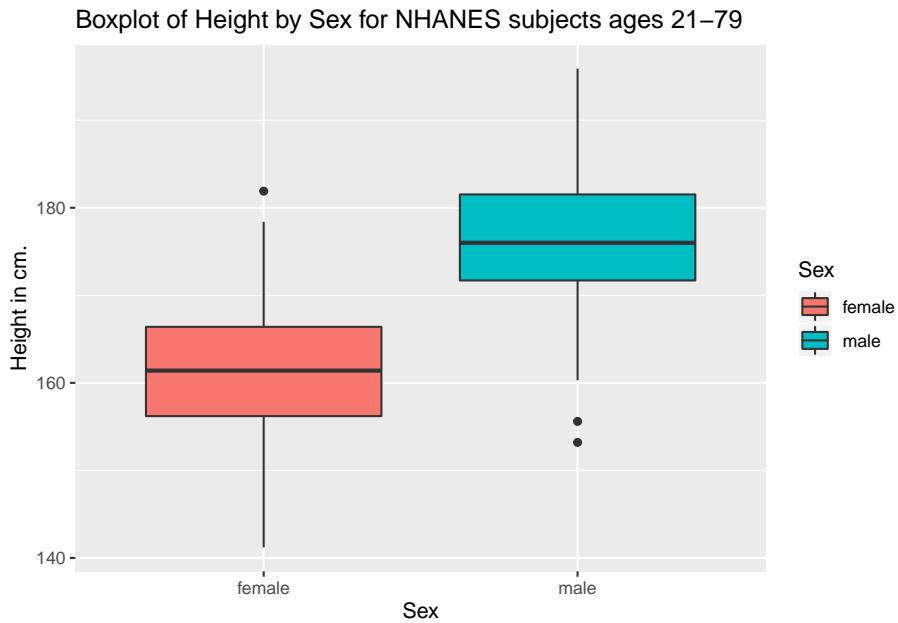
```
ggplot(data = nh_data_2179, aes(x = Sex, y = Height, color = Sex)) +
  geom_jitter(width = 0.2) +
  labs(title = "Height by Sex (jittered) for NHANES subjects ages 21-79",
       y = "Height in cm.")
```



Perhaps it might be better to summarise the distribution in a different way. We might consider a boxplot of the data.

3.7.1 A Boxplot of Height by Sex

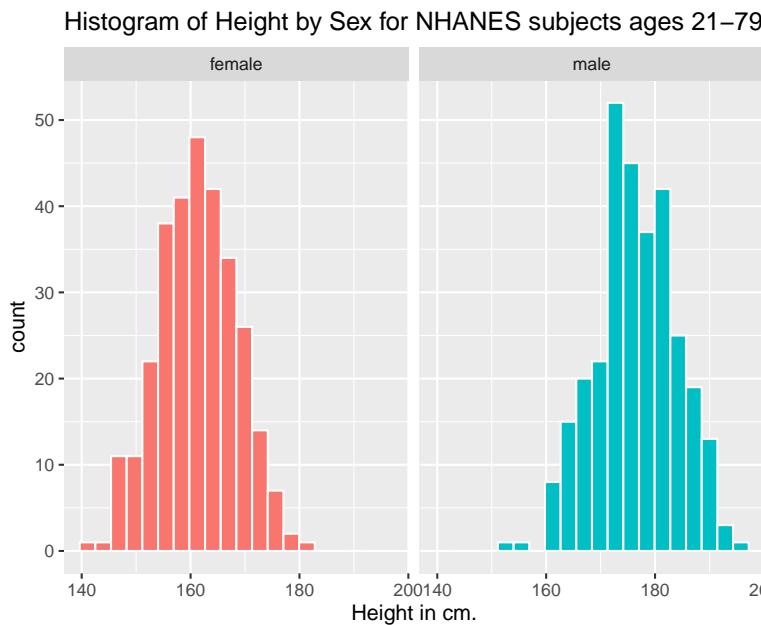
```
ggplot(data = nh_data_2179, aes(x = Sex, y = Height, fill = Sex)) +
  geom_boxplot() +
  labs(title = "Boxplot of Height by Sex for NHANES subjects ages 21-79",
       y = "Height in cm.")
```



Or perhaps we'd like to see a pair of histograms?

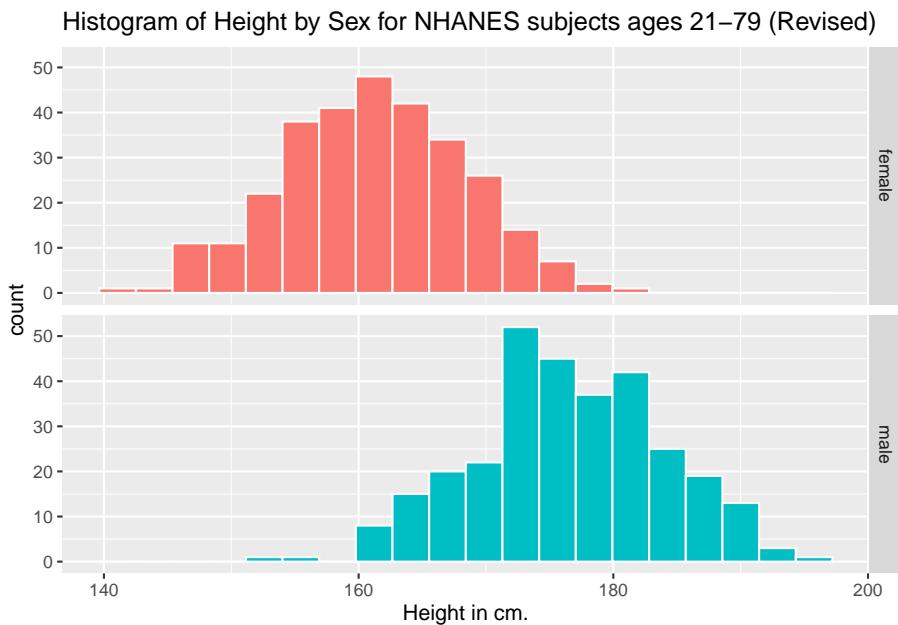
3.7.2 Histograms of Height by Sex

```
ggplot(data = nh_data_2179, aes(x = Height, fill = Sex)) +  
  geom_histogram(color = "white", bins = 20) +  
  labs(title = "Histogram of Height by Sex for NHANES subjects ages 21-79",  
       x = "Height in cm.") +  
  facet_wrap(~ Sex)
```



Can we redraw these histograms so that they are a little more comparable, and to get rid of the unnecessary legend?

```
ggplot(data = nh_data_2179, aes(x = Height, fill = Sex)) +
  geom_histogram(color = "white", bins = 20) +
  labs(title = "Histogram of Height by Sex for NHANES subjects ages 21–79 (Revised)",
       x = "Height in cm.") +
  guides(fill = FALSE) +
  facet_grid(Sex ~ .)
```



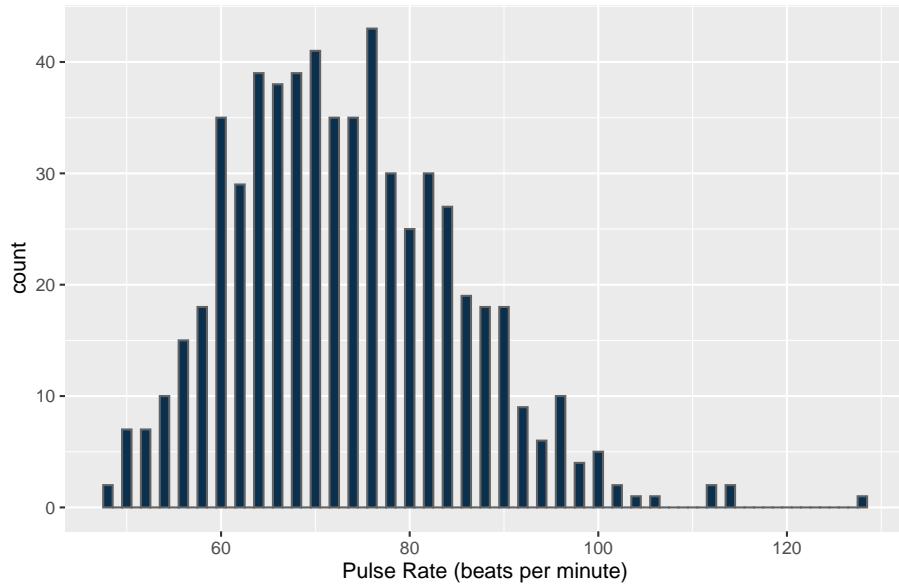
3.8 Looking at Pulse Rate

Let's look at a different outcome, the *pulse rate* for our subjects.

Here's a histogram, again with CWRU colors, for the pulse rates in our sample.

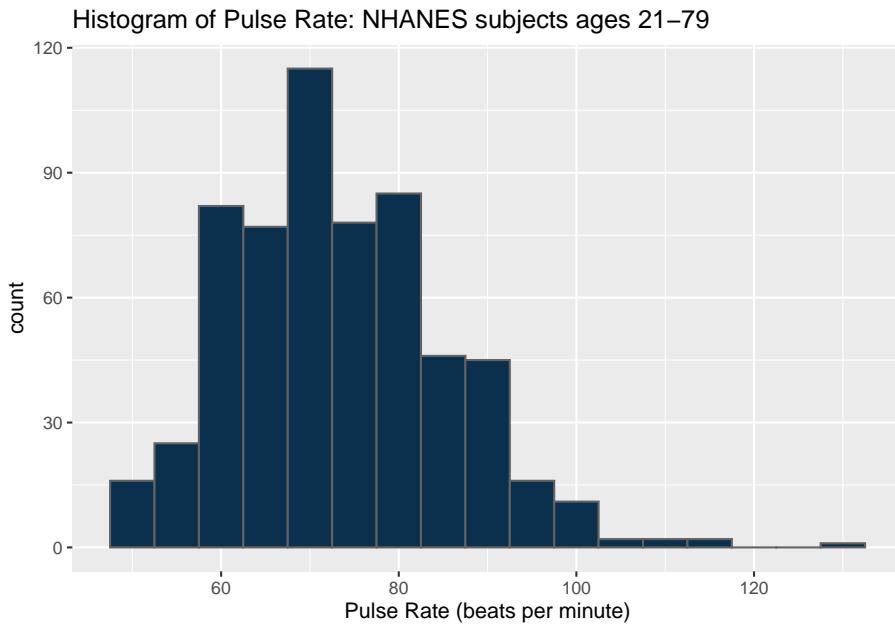
```
ggplot(data = nh_data_2179, aes(x = Pulse)) +
  geom_histogram(binwidth = 1, fill = cwrublue, col = cwrugray) +
  labs(title = "Histogram of Pulse Rate: NHANES subjects ages 21-79",
       x = "Pulse Rate (beats per minute)")
```

Histogram of Pulse Rate: NHANES subjects ages 21–79



Suppose we instead bin up groups of 5 beats per minute together as we plot the Pulse rates.

```
ggplot(data = nh_data_2179, aes(x = Pulse)) +
  geom_histogram(binwidth = 5, fill = cwrugrey, col = cwrugray) +
  labs(title = "Histogram of Pulse Rate: NHANES subjects ages 21–79",
       x = "Pulse Rate (beats per minute)")
```

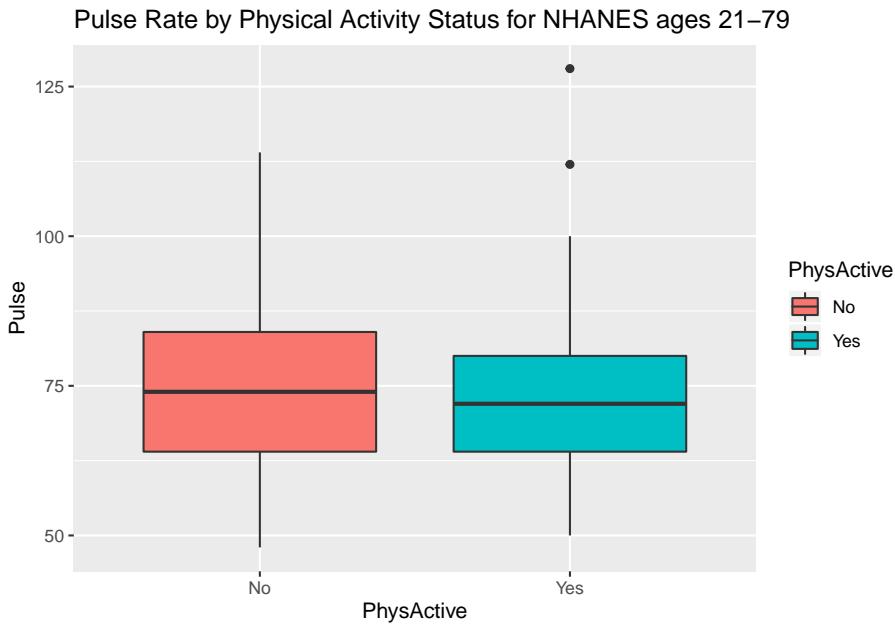


Which is the more useful representation will depend a lot on what questions you're trying to answer.

3.8.1 Pulse Rate and Physical Activity

We can also split up our data into groups based on whether the subjects are physically active. Let's try a boxplot.

```
ggplot(data = nh_data_2179, aes(y = Pulse, x = PhysActive, fill = PhysActive)) +  
  geom_boxplot() +  
  labs(title = "Pulse Rate by Physical Activity Status for NHANES ages 21-79")
```



As an accompanying numerical summary, we might ask how many people fall into each of these `PhysActive` categories, and what is their “average” Pulse rate.

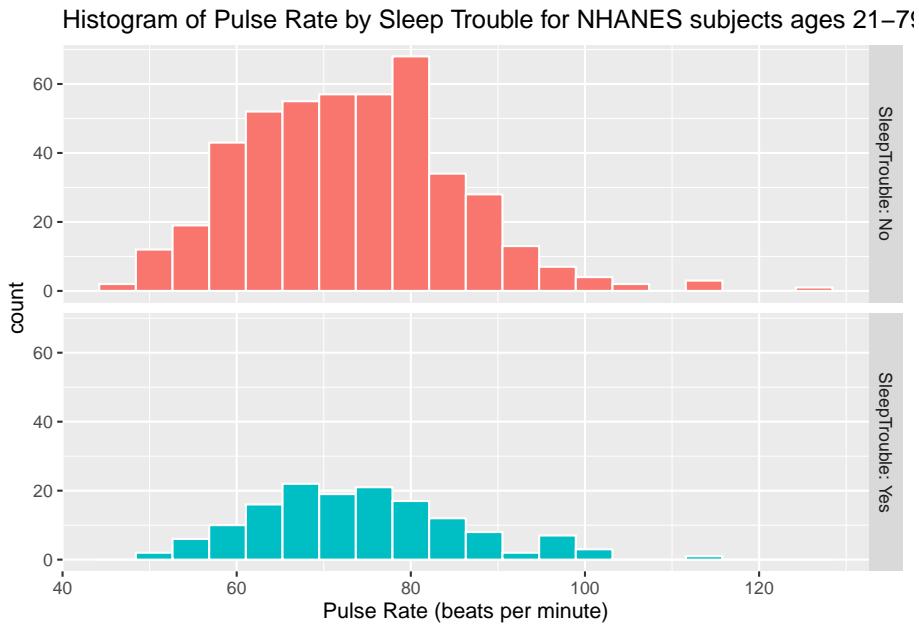
```
nh_data_2179 %>%
  group_by(PhysActive) %>%
  summarise(count = n(), mean(Pulse), median(Pulse)) %>%
  knitr::kable(digits = 2)
```

PhysActive	count	mean(Pulse)	median(Pulse)
No	293	74.2	74
Yes	310	72.4	72

The `knitr::kable(digits = 2)` piece of this command tells R Markdown to generate a table with some attractive formatting, and rounding any decimals to two figures.

3.8.2 Pulse by Sleeping Trouble

```
ggplot(data = nh_data_2179, aes(x = Pulse, fill = SleepTrouble)) +
  geom_histogram(color = "white", bins = 20) +
  labs(title = "Histogram of Pulse Rate by Sleep Trouble for NHANES subjects ages 21+",
       x = "Pulse Rate (beats per minute)") +
  guides(fill = FALSE) +
  facet_grid(SleepTrouble ~ ., labeller = "label_both")
```



How many people fall into each of these `SleepTrouble` categories, and what is their “average” Pulse rate?

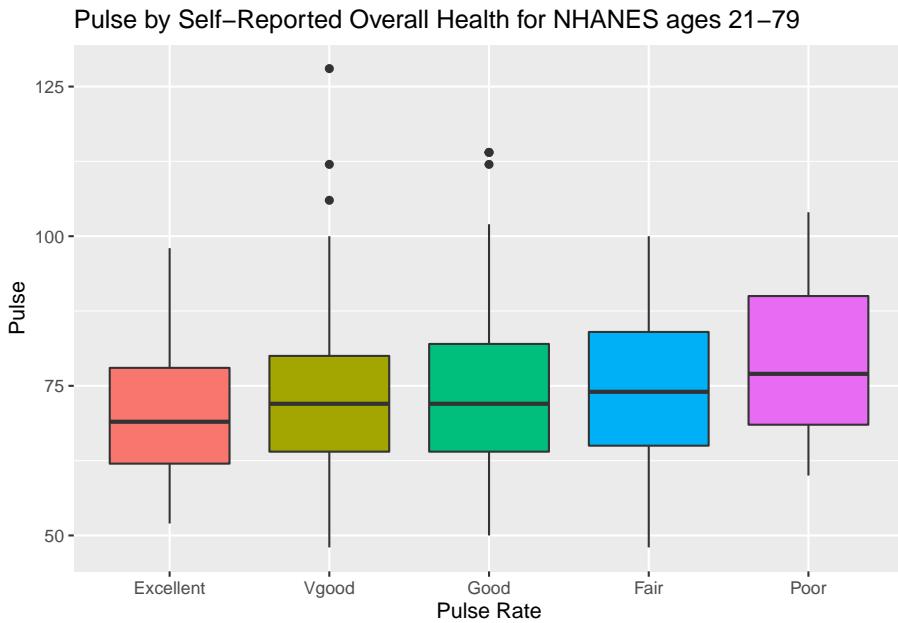
```
nh_data_2179 %>%
  group_by(SleepTrouble) %>%
  summarise(count = n(), mean(Pulse), median(Pulse)) %>%
  knitr::kable(digits = 2)
```

SleepTrouble	count	mean(Pulse)	median(Pulse)
No	457	73	72
Yes	146	74	72

3.8.3 Pulse and HealthGen

We can compare the distribution of Pulse rate across groups by the subject’s self-reported overall health (`HealthGen`), as well.

```
ggplot(data = nh_data_2179, aes(x = HealthGen, y = Pulse, fill = HealthGen)) +
  geom_boxplot() +
  labs(title = "Pulse by Self-Reported Overall Health for NHANES ages 21–79",
       x = "Pulse Rate") +
  guides(fill = FALSE)
```



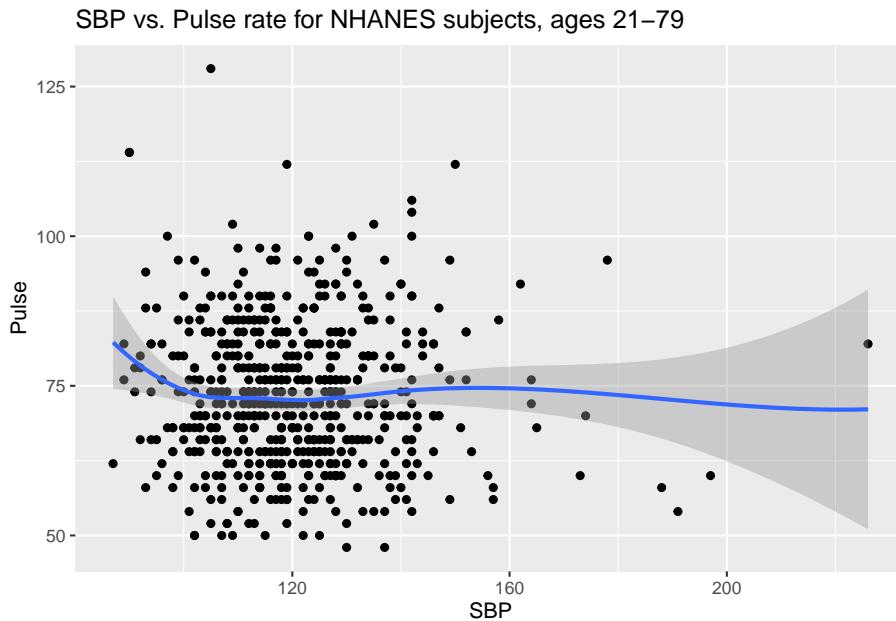
How many people fall into each of these HealthGen categories, and what is their “average” Pulse rate?

```
nh_data_2179 %>%
  group_by(HealthGen) %>%
  summarise(count = n(), mean(Pulse), median(Pulse)) %>%
  knitr::kable(digits = 2)
```

HealthGen	count	mean(Pulse)	median(Pulse)
Excellent	64	70.0	69
Vgood	196	72.8	72
Good	238	73.7	72
Fair	83	74.2	74
Poor	22	79.1	77

3.8.4 Pulse Rate and Systolic Blood Pressure

```
ggplot(data = nh_data_2179, aes(x = SBP, y = Pulse)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "SBP vs. Pulse rate for NHANES subjects, ages 21-79")
```

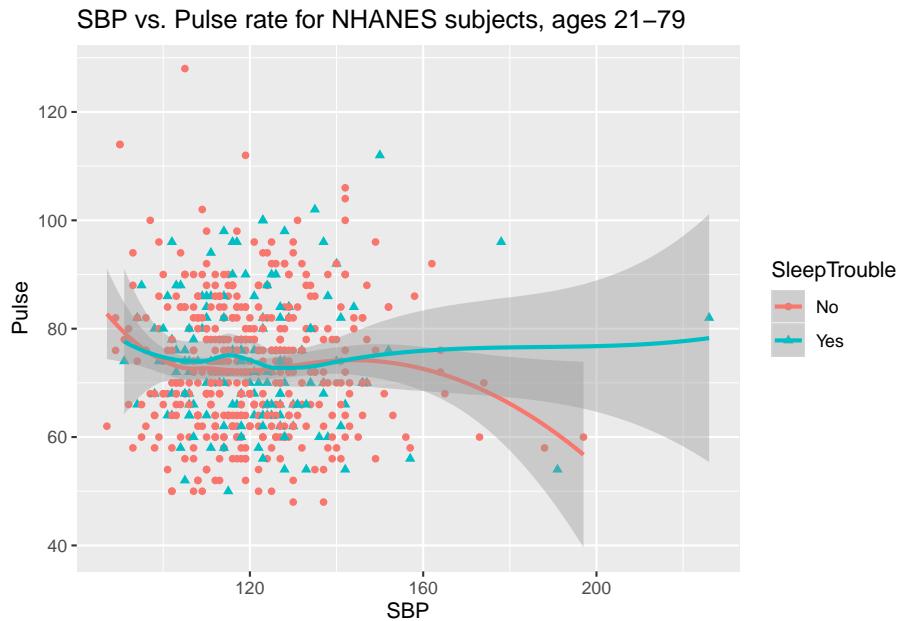


3.8.5 Sleep Troubls vs. No Sleep Trouble?

Could we see whether subjects who have described `SleepTrouble` show different SBP-pulse rate patterns than the subjects who haven't?

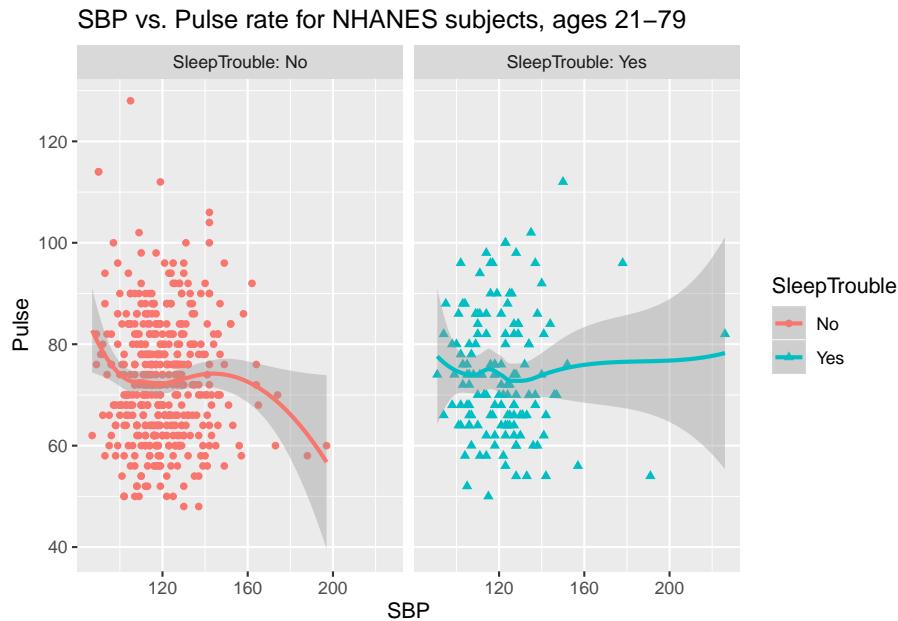
- Let's try doing this by changing the shape *and* the color of the points based on `SleepTrouble`.

```
ggplot(data = nh_data_2179,
       aes(x = SBP, y = Pulse,
           color = SleepTrouble, shape = SleepTrouble)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "SBP vs. Pulse rate for NHANES subjects, ages 21-79")
```



This plot might be easier to interpret if we facet by `SleepTrouble`, as well.

```
ggplot(data = nh_data_2179,
       aes(x = SBP, y = Pulse,
           color = SleepTrouble, shape = SleepTrouble)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "SBP vs. Pulse rate for NHANES subjects, ages 21–79") +
  facet_wrap(~ SleepTrouble, labeller = "label_both")
```



3.9 General Health Status

Here's a Table of the General Health Status results. Again, this is a self-reported rating of each subject's health on a five point scale (Excellent, Very Good, Good, Fair, Poor.)

```
nh_data_2179 %>%
  select(HealthGen) %>%
  table()
```

Excellent	Vgood	Good	Fair	Poor
64	196	238	83	22

The HealthGen data are categorical, which means that summarizing them with averages isn't as appealing as looking at percentages, proportions and rates.

Another, somewhat simpler way to get a table of this sort of information uses the `tabyl` function from the `janitor` package in R.

```
# tabyl is part of the janitor package
# already loaded: library(janitor)

nh_data_2179 %>%
  tabyl(HealthGen)
```

```
HealthGen    n  percent
Excellent   64  0.1061
Vgood      196  0.3250
Good       238  0.3947
Fair        83  0.1376
Poor        22  0.0365
```

I don't actually like the title of `percent` here, as it's really a proportion, but that can be adjusted, and we can add a total.

```
nh_data_2179 %>%
  tabyl(HealthGen) %>%
  adorn_totals() %>%
  adorn_pct_formatting()
```

```
HealthGen    n  percent
Excellent   64  10.6%
Vgood      196  32.5%
Good       238  39.5%
Fair        83  13.8%
Poor        22  3.6%
Total      603  100.0%
```

When working with an unordered categorical variable, like `MaritalStatus`, the same approach can work.

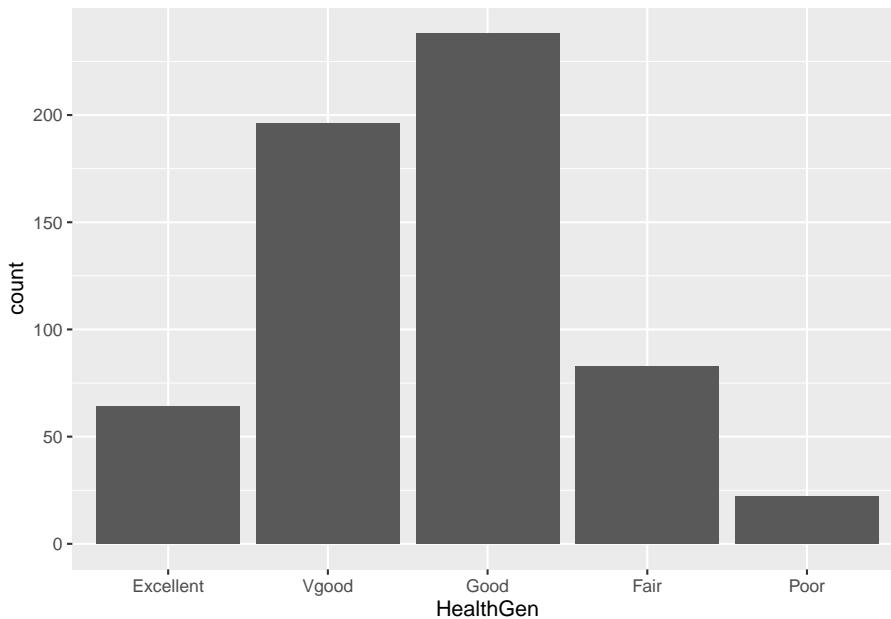
```
nh_data_2179 %>%
  tabyl(MaritalStatus) %>%
  adorn_totals() %>%
  adorn_pct_formatting()
```

```
MaritalStatus    n  percent
Divorced       61  10.1%
LivePartner    43  7.1%
Married        349  57.9%
NeverMarried   104  17.2%
Separated       8  1.3%
Widowed        38  6.3%
Total          603  100.0%
```

3.9.1 Bar Chart for Categorical Data

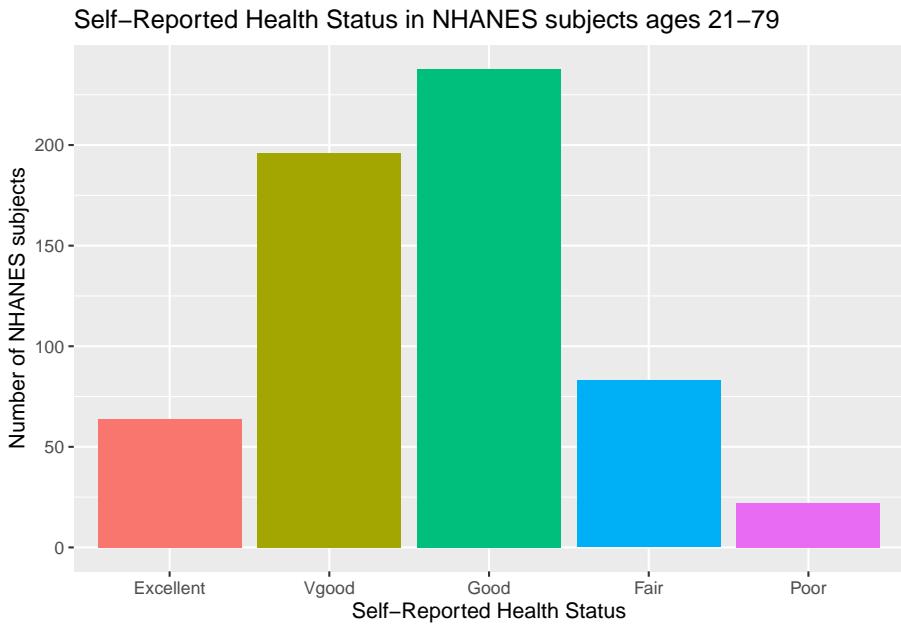
Usually, a **bar chart** is the best choice for a graphing a variable made up of categories.

```
ggplot(data = nh_data_2179, aes(x = HealthGen)) +
  geom_bar()
```



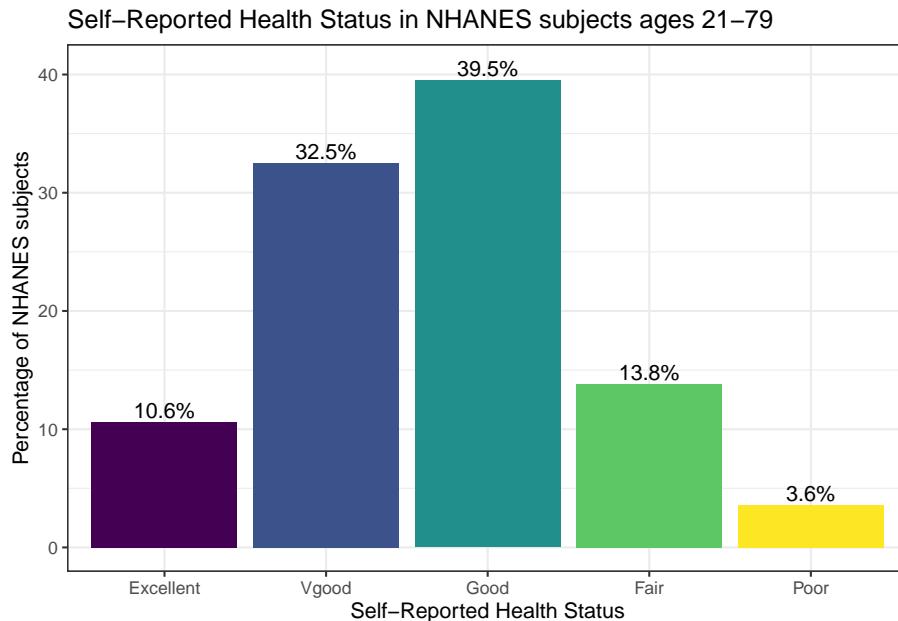
There are lots of things we can do to make this plot fancier.

```
ggplot(data = nh_data_2179, aes(x = HealthGen, fill = HealthGen)) +  
  geom_bar() +  
  guides(fill = FALSE) +  
  labs(x = "Self-Reported Health Status",  
       y = "Number of NHANES subjects",  
       title = "Self-Reported Health Status in NHANES subjects ages 21-79")
```



Or, we can really go crazy...

```
nh_data_2179 %>%
  count(HealthGen) %>%
  ungroup() %>%
  mutate(pct = round(prop.table(n) * 100, 1)) %>%
  ggplot(aes(x = HealthGen, y = pct, fill = HealthGen)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  geom_text(aes(y = pct + 1,      # nudge above top of bar
                label = paste0(pct, '%')), # prettyfy
            position = position_dodge(width = .9),
            size = 4) +
  labs(x = "Self-Reported Health Status",
       y = "Percentage of NHANES subjects",
       title = "Self-Reported Health Status in NHANES subjects ages 21-79") +
  theme_bw()
```



3.9.2 Working with Tables

We can add both row and column marginal totals, and compare subjects by Sex, as follows...

```
nh_data_2179 %>%
  tabyl(Sex, HealthGen) %>%
  adorn_totals(c("row", "col"))
```

Sex	Excellent	Vgood	Good	Fair	Poor	Total
female	27	96	121	41	14	299
male	37	100	117	42	8	304
Total	64	196	238	83	22	603

If we like, we can make this look a little more polished with the `knitr::kable` function...

```
nh_data_2179 %>%
  tabyl(Sex, HealthGen) %>%
  adorn_totals(c("row", "col")) %>%
  knitr::kable()
```

Sex	Excellent	Vgood	Good	Fair	Poor	Total
female	27	96	121	41	14	299
male	37	100	117	42	8	304
Total	64	196	238	83	22	603

Or, we can get a complete cross-tabulation, including (in this case) the percentages of people within each Sex that fall in each HealthGen category (percentages within each row) like this.

```
nh_data_2179 %>%
  tabyl(Sex, HealthGen) %>%
  adorn_totals("row") %>%
  adorn_percentages("row") %>%
  adorn_pct_formatting() %>%
  adorn_ns() %>%
  knitr::kable()
```

Sex	Excellent	Vgood	Good	Fair	Poor
female	9.0% (27)	32.1% (96)	40.5% (121)	13.7% (41)	4.7% (14)
male	12.2% (37)	32.9% (100)	38.5% (117)	13.8% (42)	2.6% (8)
Total	10.6% (64)	32.5% (196)	39.5% (238)	13.8% (83)	3.6% (22)

And, if we wanted the column percentages, to determine which sex had the higher rate of each HealthGen status level, we can get that by changing the adorn_percentages to describe results at the column level:

```
nh_data_2179 %>%
  tabyl(Sex, HealthGen) %>%
  adorn_totals("col") %>%
  adorn_percentages("col") %>%
  adorn_pct_formatting() %>%
  adorn_ns() %>%
  knitr::kable()
```

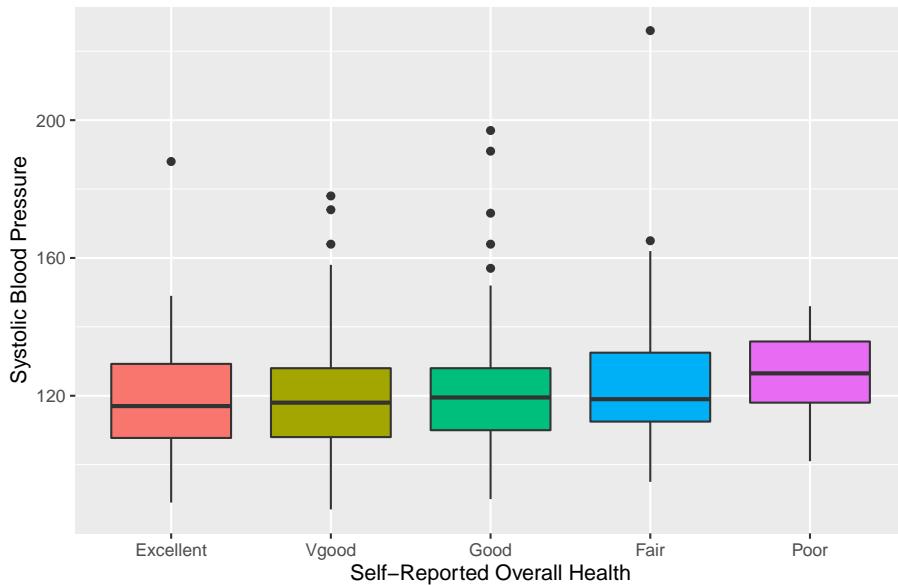
Sex	Excellent	Vgood	Good	Fair	Poor	Total
female	42.2% (27)	49.0% (96)	50.8% (121)	49.4% (41)	63.6% (14)	49.6% (299)
male	57.8% (37)	51.0% (100)	49.2% (117)	50.6% (42)	36.4% (8)	50.4% (304)

3.9.3 SBP by General Health Status

Let's consider now the relationship between self-reported overall health and systolic blood pressure.

```
ggplot(data = nh_data_2179, aes(x = HealthGen, y = SBP, fill = HealthGen)) +
  geom_boxplot() +
  labs(title = "SBP by Health Status, Overall Health for NHANES ages 21-79",
       y = "Systolic Blood Pressure", x = "Self-Reported Overall Health") +
  guides(fill = FALSE)
```

SBP by Health Status, Overall Health for NHANES ages 21–79



We can see that not too many people self-identify with the “Poor” health category.

```
nh_data_2179 %>%
  group_by(HealthGen) %>%
  summarise(count = n(), mean(SBP), median(SBP)) %>%
  knitr::kable()
```

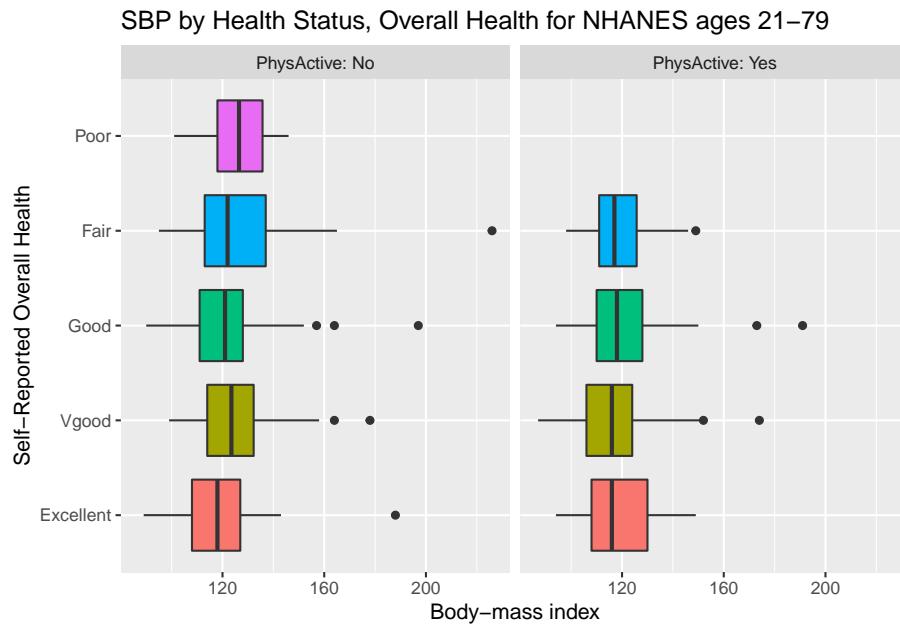
HealthGen	count	mean(SBP)	median(SBP)
Excellent	64	119	117
Vgood	196	119	118
Good	238	120	120
Fair	83	124	119
Poor	22	126	126

3.9.4 SBP by Physical Activity and General Health Status

We'll build a panel of boxplots to try to understand the relationships between Systolic Blood Pressure, General Health Status and Physical Activity. Note the use of `coord_flip` to rotate the graph 90 degrees, and the use of `labeler` within `facet_wrap` to include both the name of the (Physical Activity) variable and its value.

```
ggplot(data = nh_data_2179, aes(x = HealthGen, y = SBP, fill = HealthGen)) +
  geom_boxplot() +
  labs(title = "SBP by Health Status, Overall Health for NHANES ages 21-79",
```

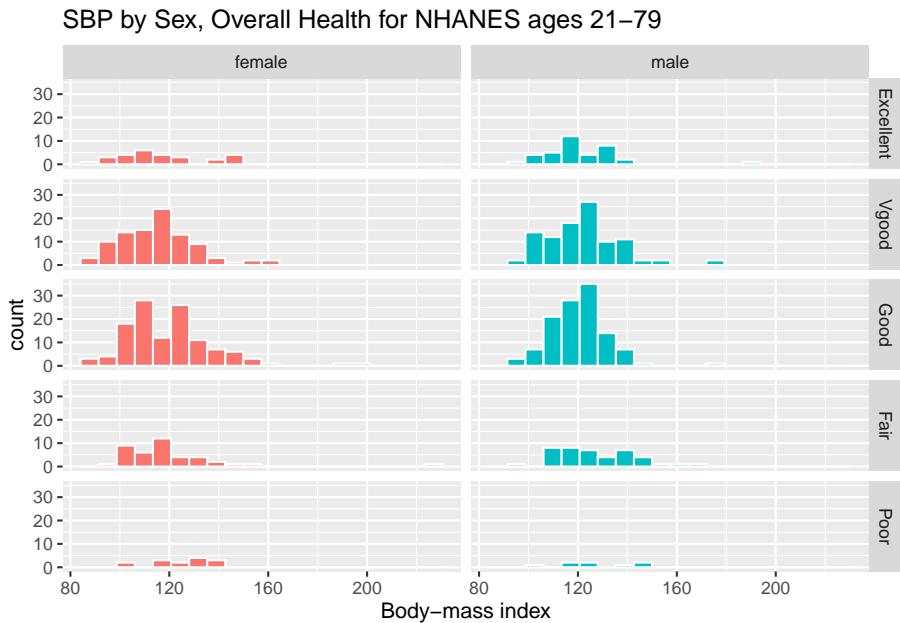
```
y = "Body-mass index", x = "Self-Reported Overall Health") +
guides(fill = FALSE) +
facet_wrap(~ PhysActive, labeller = "label_both") +
coord_flip()
```



3.9.5 SBP by Sleep Trouble and General Health Status

Here's a plot of faceted histograms, which might be used to address similar questions related to the relationship between Overall Health, Systolic Blood Pressure and Sex.

```
ggplot(data = nh_data_2179, aes(x = SBP, fill = Sex)) +
  geom_histogram(color = "white", bins = 20) +
  labs(title = "SBP by Sex, Overall Health for NHANES ages 21–79",
       x = "Body-mass index") +
  guides(fill = FALSE) +
  facet_grid(HealthGen ~ Sex)
```



3.10 Conclusions

This is just a small piece of the toolbox for visualizations that we'll create in this class. Many additional tools are on the way, but the main idea won't change. Using the `ggplot2` package, we can accomplish several critical tasks in creating a visualization, including:

- Identifying (and labeling) the axes and titles
- Identifying a type of `geom` to use, like a point, bar or histogram
- Changing fill, color, shape, size to facilitate comparisons
- Building “small multiples” of plots with faceting

Good data visualizations make it easy to see the data, and `ggplot2`'s tools make it relatively difficult to make a really bad graph.

Chapter 4

Data Structures and Types of Variables

4.1 Data require structure and context

Descriptive statistics are concerned with the presentation, organization and summary of data, as suggested in Norman and Streiner (2014). This includes various methods of organizing and graphing data to get an idea of what those data can tell us.

As Vittinghoff et al. (2012) suggest, the nature of the measurement determines how best to describe it statistically, and the main distinction is between **numerical** and **categorical** variables. Even this is a little tricky - plenty of data can have values that look like numerical values, but are just numerals serving as labels.

As Bock, Velleman, and De Veaux (2004) point out, the truly critical notion, of course, is that data values, no matter what kind, are useless without their contexts. The Five W's (Who, What [and in what units], When, Where, Why, and often How) are just as useful for establishing the context of data as they are in journalism. If you can't answer Who and What, in particular, you don't have any useful information.

In general, each row of a data frame corresponds to an individual (respondent, experimental unit, record, or observation) about whom some characteristics are gathered in columns (and these characteristics may be called variables, factors or data elements.) Every column / variable should have a name that indicates *what* it is measuring, and every row / observation should have a name that indicates *who* is being measured.

4.2 A New NHANES Adult Sample

In Chapter 3, we spent some time with a sample from the National Health and Nutrition Examination. Now, by changing the value of the `set.seed` function which determines the starting place for the random sampling, and changing some other specifications, we'll generate a new sample describing 500 adult subjects who completed the 2011-12 version of the survey when they were between the ages of 21 and 64.

Note also that what is listed in the NHANES data frame as `Gender` should be more correctly referred to as `sex`. `Sex` is a biological feature of an individual, while `Gender` is a social construct. This is an important distinction, so I'll change the name of the variable. I'm also changing the names of three other variables, to create `Race`, `SBP` and `DBP`.

```
# library(NHANES) # NHANES package/library of functions, data

nh_temp <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  filter(Age >= 21 & Age < 65) %>%
  mutate(Sex = Gender, Race = Race3, SBP = BPSysAve, DBP = BPDiaAve) %>%
  select(ID, Sex, Age, Race, Education, BMI, SBP, DBP,
         Pulse, PhysActive, Smoke100, SleepTrouble,
         MaritalStatus, HealthGen)

set.seed(431002)
# use set.seed to ensure that we all get the same random sample

nh_adults <- sample_n(nh_temp, size = 500)

nh_adults

# A tibble: 500 x 14
   ID Sex     Age Race Education   BMI   SBP   DBP Pulse PhysActive
   <int> <fct> <int> <fct> <fct>   <dbl> <int> <int> <int> <fct>
 1 71531 male    35 White Some Col~  22.4   143    90    84 Yes
 2 68613 fema~   61 White Some Col~  27.7   119    86   112 No
 3 67064 male    31 White College ~  26.6   110    76    86 Yes
 4 63924 fema~   29 Black High Sch~  41.9    98    56    74 No
 5 62840 male    60 White 8th Grade  35.8   127     0   110 No
 6 68058 male    50 White Some Col~  30.6    NA    NA    NA No
 7 68936 fema~   36 Black High Sch~  30.5   119    69    60 No
 8 71189 male    51 White College ~  25.6   112    70    54 Yes
 9 69936 fema~   54 Asian College ~  21.8   126    80    78 Yes
10 70687 male    59 White College ~  25.5   149    89    62 Yes
# ... with 490 more rows, and 4 more variables: Smoke100 <fct>,
```

```
# SleepTrouble <fct>, MaritalStatus <fct>, HealthGen <fct>
```

The data consist of 500 rows (observations) on 13 variables (columns). Essentially, we have 13 pieces of information on each of 500 adult NHANES subjects who were included in the 2011-12 panel.

4.2.1 Summarizing the Data's Structure

We can identify the number of rows and columns in a data frame or tibble with the `dim` function.

```
dim(nh_adults)
```

```
[1] 500 14
```

The `str` function provides a lot of information about the structure of a data frame or tibble.

```
str(nh_adults)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame': 500 obs. of 14 variables:
 $ ID          : int 71531 68613 67064 63924 62840 ...
 $ Sex         : Factor w/ 2 levels "female","male": 2 1 2 1 2 2 1 2 1 2 ...
 $ Age          : int 35 61 31 29 60 50 36 51 54 59 ...
 $ Race         : Factor w/ 6 levels "Asian","Black",...
 $ Education    : Factor w/ 5 levels "8th Grade","9 - 11th Grade",...
 $ BMI          : num 22.4 27.7 26.6 41.9 35.8 ...
 $ SBP          : int 143 119 110 98 127 NA 119 112 126 149 ...
 $ DBP          : int 90 86 76 56 0 NA 69 70 80 89 ...
 $ Pulse         : int 84 112 86 74 110 NA 60 54 78 62 ...
 $ PhysActive   : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 2 2 2 ...
 $ Smoke100     : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 2 1 2 1 1 ...
 $ SleepTrouble : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 2 1 1 1 1 ...
 $ MaritalStatus: Factor w/ 6 levels "Divorced","LivePartner",...
 $ HealthGen     : Factor w/ 5 levels "Excellent","Vgood",...
```

To see the first few observations, use `head`, and to see the last few, try `tail`...

```
tail(nh_adults, 5) # shows the last five observations in the data set
```

```
# A tibble: 5 x 14
  ID Sex     Age Race Education BMI   SBP   DBP Pulse PhysActive
  <int> <fct> <int> <fct> <fct> <dbl> <int> <int> <int> <fct>
1 66770 fema~  22 White Some Col~ 44.6   100    90    92 Yes
2 68754 male   57 White Some Col~ 23.2   124    85    82 No
3 70911 male   59 White College ~ 24.5   118    57    76 No
4 71393 male   27 White High Sch~ 25.7   116    61    88 Yes
5 70458 fema~  35 Black 9 - 11th~ 21.9   115    64    84 No
```

```
# ... with 4 more variables: Smoke100 <fct>, SleepTrouble <fct>,
#   MaritalStatus <fct>, HealthGen <fct>
```

4.2.2 What are the variables?

We can use the `glimpse` function to get a short preview of the data.

```
glimpse(nh_adults)
```

```
Observations: 500
Variables: 14
$ ID           <int> 71531, 68613, 67064, 63924, 62840, 68058, 68936, ...
$ Sex          <fct> male, female, male, female, male, male, female, ...
$ Age           <int> 35, 61, 31, 29, 60, 50, 36, 51, 54, 59, 59, 27, ...
$ Race          <fct> White, White, White, Black, White, White, Black, ...
$ Education     <fct> Some College, Some College, College Grad, High S...
$ BMI           <dbl> 22.4, 27.7, 26.6, 41.9, 35.8, 30.6, 30.5, 25.6, ...
$ SBP           <int> 143, 119, 110, 98, 127, NA, 119, 112, 126, 149, ...
$ DBP           <int> 90, 86, 76, 56, 0, NA, 69, 70, 80, 89, 75, 78, 6...
$ Pulse          <int> 84, 112, 86, 74, 110, NA, 60, 54, 78, 62, 82, 68...
$ PhysActive    <fct> Yes, No, Yes, No, No, No, Yes, Yes, Yes, Yes, No, ...
$ Smoke100      <fct> No, No, Yes, Yes, Yes, Yes, No, Yes, No, No, ...
$ SleepTrouble   <fct> Yes, No, No, Yes, Yes, Yes, No, No, No, No, ...
$ MaritalStatus  <fct> NeverMarried, Widowed, Married, Separated, Marri...
$ HealthGen     <fct> Good, Vgood, Good, Fair, Poor, Good, Good, NA, G...
```

The variables we have collected are described in the brief table below¹.

Variable	Description	Sample Values
ID	a numerical code identifying the subject	64427, 63788
Sex	sex of subject (2 levels)	male, female
Age	age (years) at screening of subject	37, 40
Race	reported race of subject (6 levels)	White, Asian
Education	educational level of subject (5 levels)	College Grad, High School
BMI	body-mass index, in kg/m ²	36.5, 18.2
SBP	systolic blood pressure in mm Hg	111, 115
DBP	diastolic blood pressure in mm Hg	72, 74
Pulse	60 second pulse rate in beats per minute	56, 102
PhysActive	Moderate or vigorous-intensity sports?	Yes, No
Smoke100	Smoked at least 100 cigarettes lifetime?	Yes, No
SleepTrouble	Told a doctor they have trouble sleeping?	Yes, No

¹Descriptions are adapted from the ?NHANES help file. Remember that what NHANES lists as Gender is captured here as Sex, and similarly Race3, BPSysAve and BPDiaAve from NHANES are here listed as Race, SBP and DBP.

Variable	Description	Sample Values
MaritalStatus	Marital Status	Married, Divorced
HealthGen	Self-report general health rating (5 lev.)	Vgood, Good

The levels for the multi-categorical variables are:

- **Race:** Mexican, Hispanic, White, Black, Asian, or Other.
- **Education:** 8th Grade, 9 - 11th Grade, High School, Some College, or College Grad.
- **MaritalStatus:** Married, Widowed, Divorced, Separated, NeverMarried or LivePartner (living with partner).
- **HealthGen:** Excellent, Vgood, Good, Fair or Poor.

A more detailed summary can be obtained using either the `summary` function applied to our data frame, or perhaps the `skim` function from the `skimr` package.

```
summary(nh_adults)
```

ID	Sex	Age	Race	
Min. :62199	female:221	Min. :21.0	Asian : 42	
1st Qu.:64522	male :279	1st Qu.:31.0	Black : 63	
Median :67192		Median :42.0	Hispanic: 26	
Mean :67122		Mean :41.9	Mexican : 38	
3rd Qu.:69654		3rd Qu.:53.0	White :313	
Max. :71911		Max. :64.0	Other : 18	
Education	BMI	SBP	DBP	
8th Grade : 24	Min. :17.3	Min. : 84	Min. : 0.0	
9 - 11th Grade: 60	1st Qu.:23.8	1st Qu.:110	1st Qu.: 66.0	
High School : 81	Median :27.5	Median :118	Median : 72.0	
Some College :153	Mean :28.5	Mean :119	Mean : 72.1	
College Grad :182	3rd Qu.:31.6	3rd Qu.:127	3rd Qu.: 78.0	
	Max. :63.3	Max. :209	Max. :103.0	
	NA's :5	NA's :15	NA's :15	
Pulse	PhysActive	Smoke100	SleepTrouble	MaritalStatus
Min. : 40.0	No :215	No :297	No :380	Divorced : 51
1st Qu.: 64.0	Yes:285	Yes:203	Yes:120	LivePartner : 51
Median : 72.0				Married :259
Mean : 73.4				NeverMarried:112
3rd Qu.: 82.0				Separated : 16
Max. :112.0				Widowed : 11
NA's :15				
HealthGen				
Excellent: 50				
Vgood :154				

```

Good      : 184
Fair      : 49
Poor      : 14
NA's      : 49

```

Note the appearance of NA's (indicating missing values) in some columns, and that some variables are summarized by a list of their (categorical) values and some (quantitative/numeric) variables are summarized with a minimum, quartiles and mean.

```
skimr::skim(nh_adults)
```

```

Skim summary statistics
n obs: 500
n variables: 14

-- Variable type:factor -----
variable missing complete   n n_unique
Education      0      500 500      5
HealthGen     49      451 500      5
MaritalStatus  0      500 500      6
PhysActive    0      500 500      2
Race          0      500 500      6
Sex            0      500 500      2
SleepTrouble   0      500 500      2
Smoke100      0      500 500      2
top_counts ordered
Col: 182, Som: 153, Hig: 81, 9 -: 60 FALSE
Goo: 184, Vgo: 154, Exc: 50, Fai: 49 FALSE
Mar: 259, Nev: 112, Div: 51, Liv: 51 FALSE
Yes: 285, No: 215, NA: 0 FALSE
Whi: 313, Bla: 63, Asi: 42, Mex: 38 FALSE
mal: 279, fem: 221, NA: 0 FALSE
No: 380, Yes: 120, NA: 0 FALSE
No: 297, Yes: 203, NA: 0 FALSE

-- Variable type:integer -----
variable missing complete   n      mean       sd      p0      p25      p50
Age            0      500 500    41.91    12.35    21      31      42
DBP           15      485 500    72.13    11.1     0       66      72
ID             0      500 500  67121.85  2871.51  62199  64521.75  67192
Pulse          15      485 500    73.41    12.01    40      64      72
SBP           15      485 500   119.25   15.26    84     110     118
p75  p100      hist
53              64 <U+2587><U+2587><U+2587><U+2587><U+2587><U+2587><U+2586>
78              103 <U+2581><U+2581><U+2581><U+2581><U+2582><U+2587><U+2583><U+2581>
```

```

69653.75 71911 <U+2586><U+2585><U+2585><U+2586><U+2585><U+2586><U+2587><U+2585>
82       112 <U+2581><U+2582><U+2585><U+2587><U+2585><U+2582><U+2581><U+2581>
127      209 <U+2582><U+2587><U+2587><U+2583><U+2581><U+2581><U+2581><U+2581>

-- Variable type:numeric -----
variable missing complete n mean sd p0 p25 p50 p75 p100 hist
    BMI      5     495 500 28.48 6.3 17.3 23.8 27.5 31.6 63.3 <U+2583><U+2587><U+2586><U+2581>

```

4.3 Types of Variables

4.3.1 Quantitative Variables

Variables recorded in numbers that we use as numbers are called **quantitative**. Familiar examples include incomes, heights, weights, ages, distances, times, and counts. All quantitative variables have measurement units, which tell you how the quantitative variable was measured. Without units (like miles per hour, angstroms, yen or degrees Celsius) the values of a quantitative variable have no meaning.

- It does little good to be promised a salary of 80,000 a year if you don't know whether it will be paid in Euros, dollars, yen or Estonian kroon.
- You might be surprised to see someone whose age is 72 listed in a database on childhood diseases until you find out that age is measured in months.
- Often just seeking the units can reveal a variable whose definition is challenging - just how do we measure "friendliness", or "success," for example.
- Quantitative variables may also be classified by whether they are **continuous** or can only take on a **discrete** set of values. Continuous data may take on any value, within a defined range. Suppose we are measuring height. While height is really continuous, our measuring stick usually only lets us measure with a certain degree of precision. If our measurements are only trustworthy to the nearest centimeter with the ruler we have, we might describe them as discrete measures. But we could always get a more precise ruler. The measurement divisions we make in moving from a continuous concept to a discrete measurement are usually fairly arbitrary. Another way to think of this, if you enjoy music, is that, as suggested in Norman and Streiner (2014), a piano is a *discrete* instrument, but a violin is a *continuous* one, enabling finer distinctions between notes than the piano is capable of making. Sometimes the distinction between continuous and discrete is important, but usually, it's not.
 - The `nh_adults` data includes several quantitative variables, specifically `Age`, `BMI`, `SBP`, `DBP` and `Pulse`.

- We know these are quantitative because they have units: `Age` in years, `BMI` in kg/m^2 , the `BP` measurements in mm Hg, and `Pulse` in beats per minute.
- Depending on the context, we would likely treat most of these as *discrete* given that the measurements are fairly crude (this is certainly true for `Age`, measured in years) although `BMI` is probably *continuous* in most settings, even though it is a function of two other measures (`Height` and `Weight`) which are rounded off to integer numbers of centimeters and kilograms, respectively.
- It is also possible to separate out quantitative variables into **ratio** variables or **interval** variables. An interval variable has equal distances between values, but the zero point is arbitrary. A ratio variable has equal intervals between values, and a meaningful zero point. For example, weight is an example of a ratio variable, while IQ is an example of an interval variable. We all know what zero weight is. An intelligence score like IQ is a different matter. We say that the average IQ is 100, but that's only by convention. We could just as easily have decided to add 400 to every IQ value and make the average 500 instead. Because IQ's intervals are equal, the difference between an IQ of 70 and an IQ of 80 is the same as the difference between 120 and 130. However, an IQ of 100 is not twice as high as an IQ of 50. The point is that if the zero point is artificial and moveable, then the differences between numbers are meaningful but the ratios between them are not. On the other hand, most lab test values are ratio variables, as are physical characteristics like height and weight. A person who weighs 100 kg is twice as heavy as one who weighs 50 kg; even when we convert kg to pounds, this is still true. For the most part, we can treat and analyze interval or ratio variables the same way.
 - Each of the quantitative variables in our `nh_adults` data can be thought of as ratio variables.
- Quantitative variables lend themselves to many of the summaries we will discuss, like means, quantiles, and our various measures of spread, like the standard deviation or inter-quartile range. They also have at least a chance to follow the Normal distribution.

4.3.1.1 A look at BMI (Body-Mass Index)

The definition of BMI (*body-mass index*) for adult subjects (which is expressed in units of kg/m^2) is:

$$\text{Body Mass Index} = \frac{\text{weight in kg}}{(\text{height in meters})^2} = 703 \times \frac{\text{weight in pounds}}{(\text{height in inches})^2}$$

[BMI is essentially] ... a measure of a person's *thinness* or *thickness*... BMI was designed for use as a simple means of classifying

average sedentary (physically inactive) populations, with an average body composition. For these individuals, the current value recommendations are as follow: a BMI from 18.5 up to 25 may indicate optimal weight, a BMI lower than 18.5 suggests the person is underweight, a number from 25 up to 30 may indicate the person is overweight, and a number from 30 upwards suggests the person is obese.

Wikipedia, https://en.wikipedia.org/wiki/Body_mass_index

4.3.2 Qualitative (Categorical) Variables

Qualitative or categorical variables consist of names of categories. These names may be numerical, but the numbers (or names) are simply codes to identify the groups or categories into which the individuals are divided. Categorical variables with two categories, like yes or no, up or down, or, more generally, 1 and 0, are called **binary** variables. Those with more than two-categories are sometimes called **multi-categorical** variables.

- When the categories included in a variable are merely names, and come in no particular order, we sometimes call them **nominal** variables. The most important summary of such a variable is usually a table of frequencies, and the mode becomes an important single summary, while the mean and median are essentially useless.
 - In the `nh_adults` data, `Race` is a nominal variable with multiple unordered categories. So is `MaritalStatus`.
- The alternative categorical variable (where order matters) is called **ordinal**, and includes variables that are sometimes thought of as falling right in between quantitative and qualitative variables.
 - Examples of ordinal multi-categorical variables in the `nh_adults` data include the `Education` and `HealthGen` variables.
 - Answers to questions like “How is your overall physical health?” with available responses Excellent, Very Good, Good, Fair or Poor, which are often coded as 1-5, certainly provide a perceived *order*, but a group of people with average health status 4 (Very Good) is not necessarily twice as healthy as a group with average health status of 2 (Fair).
- Sometimes we treat the values from ordinal variables as sufficiently scaled to permit us to use quantitative approaches like means, quantiles, and standard deviations to summarize and model the results, and at other times, we'll treat ordinal variables as if they were nominal, with tables and percentages our primary tools.
- Note that all binary variables may be treated as ordinal, or nominal.

- Binary variables in the `nh_adults` data include `Sex`, `PhysActive`, `Smoke100`, `SleepTrouble`. Each can be thought of as either ordinal or nominal.

Lots of variables may be treated as either quantitative or qualitative, depending on how we use them. For instance, we usually think of age as a quantitative variable, but if we simply use age to make the distinction between “child” and “adult” then we are using it to describe categorical information. Just because your variable’s values are numbers, don’t assume that the information provided is quantitative.

4.4 Creating a Data Report Quickly

The `DataExplorer` package has a function called `create_report` that you might like. It is easy to run, but the results can be a bit overwhelming, so I’ll just show the code, but not run the report here.

```
DataExplorer::create_report(nh_adults)
```

Chapter 5

Summarizing Quantitative Variables

Most numerical summaries that might be new to you are applied most appropriately to quantitative variables. The measures that will interest us relate to:

- the **center** of our distribution,
- the **spread** of our distribution, and
- the **shape** of our distribution.

5.1 The **summary** function for Quantitative data

R provides a small sampling of numerical summaries with the **summary** function, for instance.

```
nh_adults %>%
  select(Age, BMI, SBP, DBP, Pulse) %>%
  summary()
```

Age	BMI	SBP	DBP
Min. :21.0	Min. :17.3	Min. : 84	Min. : 0.0
1st Qu.:31.0	1st Qu.:23.8	1st Qu.:110	1st Qu.: 66.0
Median :42.0	Median :27.5	Median :118	Median : 72.0
Mean :41.9	Mean :28.5	Mean :119	Mean : 72.1
3rd Qu.:53.0	3rd Qu.:31.6	3rd Qu.:127	3rd Qu.: 78.0
Max. :64.0	Max. :63.3	Max. :209	Max. :103.0
	NA's :5	NA's :15	NA's :15
	Pulse		

```
Min.    : 40.0
1st Qu.: 64.0
Median  : 72.0
Mean    : 73.4
3rd Qu.: 82.0
Max.    :112.0
NA's    :15
```

This basic summary includes a set of five **quantiles**¹, plus the sample's **mean**.

- **Min.** = the **minimum** value for each variable, so, for example, the youngest subject's Age was 21.
- **1st Qu.** = the **first quartile** (25th percentile) for each variable - for example, 25% of the subjects were Age 31 or younger.
- **Median** = the **median** (50th percentile) - half of the subjects were Age 42 or younger.
- **Mean** = the **mean**, usually what one means by an *average* - the sum of the Ages divided by 500 is 41.9,
- **3rd Qu.** = the **third quartile** (75th percentile) - 25% of the subjects were Age 53 or older.
- **Max.** = the **maximum** value for each variable, so the oldest subject was Age 64.

The summary also specifies the number of missing values for each variable. Here, we are missing 5 of the BMI values, for example.

5.2 Measuring the Center of a Distribution

5.2.1 The Mean and The Median

The **mean** and **median** are the most commonly used measures of the center of a distribution for a quantitative variable. The median is the more generally useful value, as it is relevant even if the data have a shape that is not symmetric. We might also collect the **sum** of the observations, and the **count** of the number of observations, usually symbolized with *n*.

For variables without missing values, like `Age`, this is pretty straightforward.

```
nh_adults %>%
  summarise(n = n(), Mean = mean(Age), Median = median(Age), Sum = sum(Age))

# A tibble: 1 x 4
  n   Mean Median   Sum
  <int> <dbl> <dbl> <int>
1   500  41.9    42 20953
```

¹The quantiles (sometimes referred to as percentiles) can also be summarised with a boxplot.

And again, the Mean is just the Sum (20953), divided by the number of non-missing values of Age (500), or 41.906.

The Median is the middle value when the data are sorted in order. When we have an odd number of values, this is sufficient. When we have an even number, as in this case, we take the mean of the two middle values. We could sort and list all 500 Ages, if we wanted to do so.

```
nh_adults %>% select(Age) %>%
  arrange(Age)
```

```
# A tibble: 500 x 1
  Age
  <int>
1 21
2 21
3 21
4 21
5 21
6 21
7 21
8 21
9 22
10 22
# ... with 490 more rows
```

But this data set figures we don't want to output more than 10 observations to a table like this.

If we really want to see all of the data, we can use `View(nh_adults)` to get a spreadsheet-style presentation, or use the `sort` command...

```
sort(nh_adults$Age)
```

```
[1] 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22 22 22 22 22 22 23 23 23
[24] 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
[47] 25 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 27
[70] 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 28 28 28 28 28 28 28 28 28 28 28
[93] 28 28 28 28 29 29 29 29 29 29 29 29 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
[116] 30 30 30 31 31 31 31 31 31 31 31 31 31 31 31 32 32 32 32 32 32 32 32 32 32 32
[139] 32 32 32 32 32 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 34 34 34 34 34 34
[162] 34 34 34 35 35 35 35 35 35 35 35 35 36 36 36 36 36 36 36 36 36 37 37 37 37 37
[185] 37 37 37 37 37 37 37 37 37 37 37 37 37 37 38 38 38 38 38 38 38 38 38 38 38 38
[208] 38 38 38 39 39 39 39 39 39 39 39 39 39 39 39 39 39 40 40 40 40 40 40 40 40 40
[231] 40 40 40 40 41 41 41 41 41 41 41 41 41 41 41 41 42 42 42 42 42 42 42 42 42
[254] 42 42 42 42 42 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 44 44 44 44 44
[277] 44 44 44 44 44 44 44 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45
[300] 46 46 46 46 46 46 47 47 47 47 47 47 47 47 47 47 47 47 48 48 48 48 48 48 48
```

```
[323] 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 49 50 50 50 50 50  

[346] 50 50 50 50 50 50 50 50 50 51 51 51 51 51 51 51 51 51 52 52 52 52 52 52  

[369] 52 52 52 53 53 53 53 53 53 53 53 53 53 53 53 53 53 53 54 54 54 54 54 54 54  

[392] 54 54 54 54 54 54 55 55 55 55 55 55 55 55 55 55 55 55 56 56 56 56 56 56 56  

[415] 56 56 56 56 56 56 56 56 56 57 57 57 57 57 57 57 57 57 57 57 57 57 57 58 58  

[438] 58 58 58 58 58 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 60 60  

[461] 60 60 60 60 60 60 60 60 61 61 61 61 61 61 61 61 61 61 62 62 62 62  

[484] 62 62 63 63 63 63 63 63 63 64 64 64 64 64 64 64 64 64 64 64
```

Again, to find the median, we would take the mean of the middle two observations in this sorted data set. That would be the 250th and 251st largest Ages.

```
sort(nh_adults$Age) [250:251]
```

```
[1] 42 42
```

5.2.2 Dealing with Missingness

When calculating a mean, you may be tempted to try something like this...

```
nh_adults %>%
  summarise(mean(Pulse), median(Pulse))

# A tibble: 1 x 2
`mean(Pulse)` `median(Pulse)`
<dbl>           <int>
1             NA            NA
```

This fails because we have some missing values in the Pulse data. We can address this by either omitting the data with missing values before we run the summarise function, or tell the mean and median summary functions to remove missing values².

```
nh_adults %>%
  filter(complete.cases(Pulse)) %>%
  summarise(count = n(), mean(Pulse), median(Pulse))

# A tibble: 1 x 3
count `mean(Pulse)` `median(Pulse)`
<int>       <dbl>        <int>
1     485         73.4         72
```

Or, we could tell the summary functions themselves to remove NA values.

```
nh_adults %>%
  summarise(mean(Pulse, na.rm=TRUE), median(Pulse, na.rm=TRUE))

# A tibble: 1 x 2
```

²We could also use `!is.na` in place of `complete.cases` to accomplish the same thing.

```
`mean(Pulse, na.rm = TRUE)` `median(Pulse, na.rm = TRUE)`
<dbl> <int>
1       73.4      72
```

While we eventually discuss the importance of **imputation** when dealing with missing data, this doesn't apply to providing descriptive summaries of actual, observed values.

5.2.3 The Mode of a Quantitative Variable

One other less common measure of the center of a quantitative variable's distribution is its most frequently observed value, referred to as the **mode**. This measure is only appropriate for discrete variables, be they quantitative or categorical. To find the mode, we usually tabulate the data, and then sort by the counts of the numbers of observations.

```
nh_adults %>%
  group_by(Age) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
# A tibble: 44 x 2
  Age   count
  <int> <int>
1 37     18
2 49     17
3 24     15
4 27     15
5 30     15
6 43     15
7 45     15
8 50     15
9 56     15
10 59    15
# ... with 34 more rows
```

Note the use of three different “verbs” in our function there - for more explanation of this strategy, visit Grolemund and Wickham (2019).

As an alternative, the **modeest** package's **mfv** function calculates the sample mode (or most frequent value).³

³See the documentation for the **modeest** package's **mlv** function to look at other definitions of the mode.

5.3 Measuring the Spread of a Distribution

Statistics is all about variation, so spread or dispersion is an important fundamental concept in statistics. Measures of spread like the inter-quartile range and range (maximum - minimum) can help us understand and compare data sets. If the values in the data are close to the center, the spread will be small. If many of the values in the data are scattered far away from the center, the spread will be large.

5.3.1 The Range and the Interquartile Range (IQR)

The **range** of a quantitative variable is sometimes interpreted as the difference between the maximum and the minimum, even though R presents the actual minimum and maximum values when you ask for a range...

```
nh_adults %>%
  select(Age) %>%
  range()
```

```
[1] 21 64
```

And, for a variable with missing values, we can use...

```
nh_adults %>%
  select(BMI) %>%
  range(., na.rm=TRUE)
```

```
[1] 17.3 63.3
```

A more interesting and useful statistic is the **inter-quartile range**, or IQR, which is the range of the middle half of the distribution, calculated by subtracting the 25th percentile value from the 75th percentile value.

```
nh_adults %>%
  summarise(IQR(Age), quantile(Age, 0.25), quantile(Age, 0.75))
```

```
# A tibble: 1 x 3
`IQR(Age)` `quantile(Age, 0.25)` `quantile(Age, 0.75)`
<dbl>           <dbl>           <dbl>
1          22             31             53
```

We can calculate the range and IQR nicely from the summary information on quantiles, of course:

```
nh_adults %>%
  select(Age, BMI, SBP, DBP, Pulse) %>%
  summary()
```

Age	BMI	SBP	DBP
21	17.3	63.3	53

```

Min.   :21.0   Min.   :17.3   Min.   : 84   Min.   : 0.0
1st Qu.:31.0   1st Qu.:23.8   1st Qu.:110   1st Qu.: 66.0
Median :42.0   Median :27.5   Median :118   Median : 72.0
Mean   :41.9   Mean   :28.5   Mean   :119   Mean   : 72.1
3rd Qu.:53.0   3rd Qu.:31.6   3rd Qu.:127   3rd Qu.: 78.0
Max.   :64.0   Max.   :63.3   Max.   :209   Max.   :103.0
NA's    :5       NA's    :15      NA's    :15      NA's    :15

Pulse
Min.   : 40.0
1st Qu.: 64.0
Median : 72.0
Mean   : 73.4
3rd Qu.: 82.0
Max.   :112.0
NA's    :15

```

5.3.2 The Variance and the Standard Deviation

The IQR is always a reasonable summary of spread, just as the median is always a reasonable summary of the center of a distribution. Yet, most people are inclined to summarise a batch of data using two numbers: the **mean** and the **standard deviation**. This is really only a sensible thing to do if you are willing to assume the data follow a Normal distribution: a bell-shaped, symmetric distribution without substantial outliers.

But **most data do not (even approximately) follow a Normal distribution**. Summarizing by the median and quartiles (25th and 75th percentiles) is much more robust, explaining R's emphasis on them.

5.3.3 Obtaining the Variance and Standard Deviation in R

Here are the variances of the quantitative variables in the `nh_adults` data. Note the need to include `na.rm = TRUE` to deal with the missing values in some variables.

```

nh_adults %>%
  select(Age, BMI, SBP, DBP, Pulse) %>%
  summarise_all(var, na.rm = TRUE)

# A tibble: 1 x 5
  Age    BMI    SBP    DBP  Pulse
  <dbl> <dbl> <dbl> <dbl> <dbl>
1 152.  39.7  233.  123.  144.

```

And here are the standard deviations of those same variables.

```
nh_adults %>%
  select(Age, BMI, SBP, DBP, Pulse) %>%
  summarise_all(sd, na.rm = TRUE)

# A tibble: 1 x 5
  Age    BMI   SBP   DBP Pulse
  <dbl> <dbl> <dbl> <dbl> <dbl>
1 12.3  6.30 15.3  11.1 12.0
```

5.3.4 Defining the Variance and Standard Deviation

Bock, Velleman, and De Veaux (2004) have lots of useful thoughts here, which are lightly edited here.

In thinking about spread, we might consider how far each data value is from the mean. Such a difference is called a *deviation*. We could just average the deviations, but the positive and negative differences always cancel out, leaving an average deviation of zero, so that's not helpful. Instead, we *square* each deviation to obtain non-negative values, and to emphasize larger differences. When we add up these squared deviations and find their mean (almost), this yields the **variance**.

$$\text{Variance} = s^2 = \frac{\sum(y - \bar{y})^2}{n - 1}$$

Why almost? It would be the mean of the squared deviations only if we divided the sum by n , but instead we divide by $n - 1$ because doing so produces an estimate of the true (population) variance that is *unbiased*⁴. If you're looking for a more intuitive explanation, this Stack Exchange link awaits your attention.

- To return to the original units of measurement, we take the square root of s^2 , and instead work with s , the **standard deviation**.

$$\text{Standard Deviation} = s = \sqrt{\frac{\sum(y - \bar{y})^2}{n - 1}}$$

5.3.5 Empirical Rule Interpretation of the Standard Deviation

For a set of measurements that follow a Normal distribution, the interval:

⁴When we divide by $n-1$ as we calculate the sample variance, the average of the sample variances for all possible samples is equal to the population variance. If we instead divided by n , the average sample variance across all possible samples would be a little smaller than the population variance.

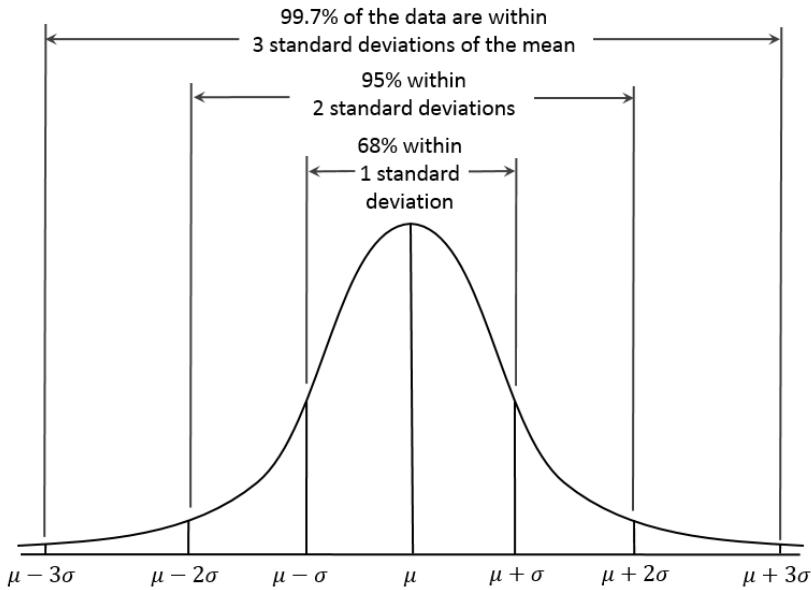


Figure 5.1: The Normal Distribution and the Empirical Rule

- Mean \pm Standard Deviation contains approximately 68% of the measurements;
- Mean \pm 2(Standard Deviation) contains approximately 95% of the measurements;
- Mean \pm 3(Standard Deviation) contains approximately all (99.7%) of the measurements.

We often refer to the population or process mean of a distribution with μ and the standard deviation with σ , leading to the Figure below.

But if the data are not from an approximately Normal distribution, then this Empirical Rule is less helpful.

5.3.6 Chebyshev's Inequality: One Interpretation of the Standard Deviation

Chebyshev's Inequality tells us that for any distribution, regardless of its relationship to a Normal distribution, no more than $1/k^2$ of the distribution's values can lie more than k standard deviations from the mean. This implies, for instance, that for **any** distribution, at least 75% of the values must lie within two standard deviations of the mean, and at least 89% must lie within three standard deviations of the mean.

Again, most data sets do not follow a Normal distribution. We'll return to this

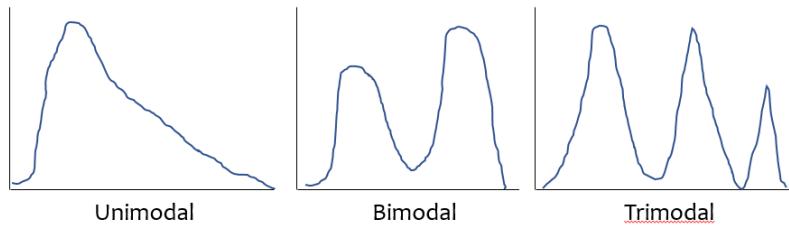


Figure 5.2: Unimodal and Multimodal Sketches

notion soon. But first, let's try to draw some pictures that let us get a better understanding of the distribution of our data.

5.4 Measuring the Shape of a Distribution

When considering the shape of a distribution, one is often interested in three key points.

- The number of modes in the distribution, which I always assess through plotting the data.
- The **skewness**, or symmetry that is present, which I typically assess by looking at a plot of the distribution of the data, but if required to, will summarise with a non-parametric measure of **skewness**.
- The **kurtosis**, or heavy-tailedness (outlier-proneness) that is present, usually in comparison to a Normal distribution. Again, this is something I nearly inevitably assess graphically, but there are measures.

A Normal distribution has a single mode, is symmetric and, naturally, is neither heavy-tailed nor light-tailed as compared to a Normal distribution (we call this mesokurtic).

5.4.1 Multimodal vs. Unimodal distributions

A unimodal distribution, on some level, is straightforward. It is a distribution with a single mode, or “peak” in the distribution. Such a distribution may be skewed or symmetric, light-tailed or heavy-tailed. We usually describe as multimodal distributions like the two on the right below, which have multiple local maxima, even though they have just a single global maximum peak.

Truly multimodal distributions are usually described that way in terms of shape. For unimodal distributions, skewness and kurtosis become useful ideas.

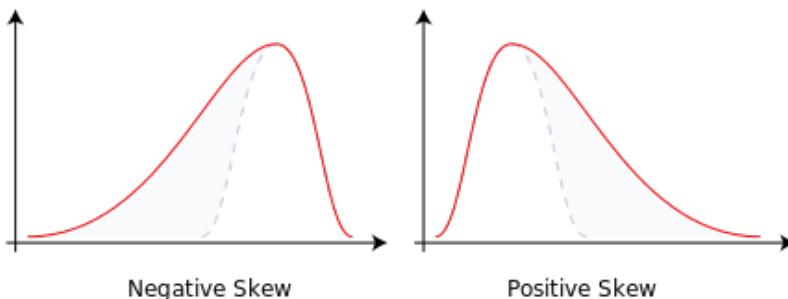


Figure 5.3: Negative (Left) Skew and Positive (Right) Skew

5.4.2 Skew

Whether or not a distribution is approximately symmetric is an important consideration in describing its shape. Graphical assessments are always most useful in this setting, particularly for unimodal data. My favorite measure of skew, or skewness if the data have a single mode, is:

$$skew_1 = \frac{\text{mean} - \text{median}}{\text{standard deviation}}$$

- Symmetric distributions generally show values of $skew_1$ near zero. If the distribution is actually symmetric, the mean should be equal to the median.
- Distributions with $skew_1$ values above 0.2 in absolute value generally indicate meaningful skew.
- Positive skew (mean > median if the data are unimodal) is also referred to as *right skew*.
- Negative skew (mean < median if the data are unimodal) is referred to as *left skew*.

5.4.3 Kurtosis

When we have a unimodal distribution that is symmetric, we will often be interested in the behavior of the tails of the distribution, as compared to a Normal distribution with the same mean and standard deviation. High values of kurtosis measures (and there are several) indicate data which has extreme outliers, or is heavy-tailed.

- A mesokurtic distribution has similar tail behavior to what we would expect from a Normal distribution.
- A leptokurtic distribution is a thinner distribution, with lighter tails (fewer observations far from the center) than we'd expect from a Normal distribution.

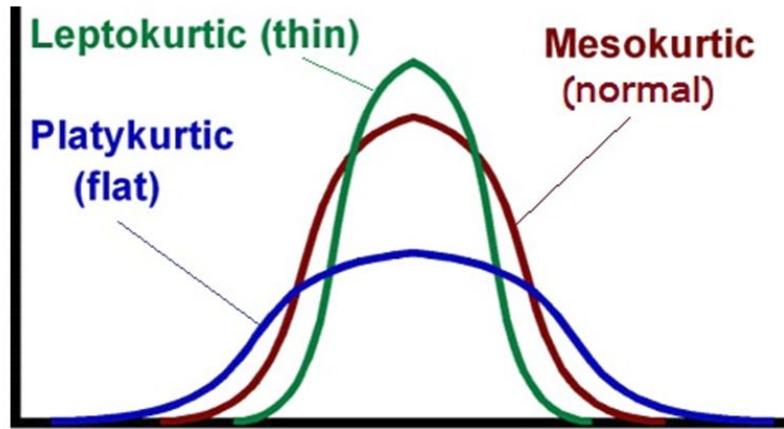


Figure 5.4: The Impact of Kurtosis

- A platykurtic distribution is a flatter distribution, with heavier tails (more observations far from the center) than we'd expect from a Normal distribution.

Graphical tools are in most cases the best way to identify issues related to kurtosis.

5.5 More Detailed Numerical Summaries for Quantitative Variables

5.5.1 favstats in the mosaic package

The `favstats` function adds the standard deviation, and counts of overall and missing observations to our usual `summary` for a continuous variable. Let's look at systolic blood pressure, because we haven't yet.

```
mosaic::favstats(~ SBP, data = nh_adults)
```

```
min   Q1 median   Q3 max mean    sd   n missing
 84  110    118  127 209  119 15.3 485      15
```

We could, of course, duplicate these results with a rather lengthy set of `summarise` pieces...

```
nh_adults %>%
  filter(complete.cases(SBP)) %>%
  summarise(min = min(SBP), Q1 = quantile(SBP, 0.25), median = median(SBP),
```

5.5. MORE DETAILED NUMERICAL SUMMARIES FOR QUANTITATIVE VARIABLES 83

```
Q3 = quantile(SBP, 0.75), max = max(SBP),
mean = mean(SBP), sd = sd(SBP), n = n(), missing = sum(is.na(SBP)))  
  
# A tibble: 1 x 9
  min    Q1 median    Q3   max   mean    sd     n missing
  <int> <dbl> <int> <dbl> <int> <dbl> <dbl> <int>    <int>
1    84    110    118    127    209   119.   15.3    485      0
```

The somewhat unusual structure of `favstats` (complete with an easy to forget ~) is actually helpful. It allows you to look at some interesting grouping approaches, like this:

```
mosaic::favstats(SBP ~ Education, data = nh_adults)  
  
Education min    Q1 median    Q3   max   mean    sd     n missing
1 8th Grade  95    114    122    132    167   124.   18.9    22      2
2 9 - 11th Grade  92    108    114    125    170   117.   13.7    60      0
3 High School  91    112    119    129    209   123.   19.7    77      4
4 Some College 85    110    119    128    165   119.   13.5   149      4
5 College Grad 84    109    118    126    171   118.   14.3   177      5
```

Of course, we could accomplish the same comparison with `dplyr` commands, too, but the `favstats` approach has much to offer.

```
nh_adults %>%
  filter(complete.cases(SBP, Education)) %>%
  group_by(Education) %>%
  summarise(min = min(SBP), Q1 = quantile(SBP, 0.25), median = median(SBP),
            Q3 = quantile(SBP, 0.75), max = max(SBP),
            mean = mean(SBP), sd = sd(SBP), n = n(), missing = sum(is.na(SBP)))  
  
# A tibble: 5 x 10
  Education      min    Q1 median    Q3   max   mean    sd     n missing
  <fct>        <int> <dbl> <dbl> <dbl> <int> <dbl> <dbl> <int>    <int>
1 8th Grade     95    114    122    132.   167   124.   18.9    22      0
2 9 - 11th Grade  92    108    114    125.   170   117.   13.7    60      0
3 High School   91    112    119    129    209   123.   19.7    77      0
4 Some College   85    110    119    128    165   119.   13.5   149      0
5 College Grad   84    109    118    126    171   118.   14.3   177      0
```

5.5.2 `skim` in the `skimr` package

```
nh_adults %>%
  select(Pulse) %>%
  skimr::skim()
```

Skim summary statistics

```

n obs: 500
n variables: 1

-- Variable type:integer -----
variable missing complete n mean sd p0 p25 p50 p75 p100 hist
Pulse      15     485 500 73.41 12.01 40  64  72  82 112 <U+2581><U+2582><U+2585>

```

5.5.3 `describe` in the `psych` package

The `psych` package has a more detailed list of numerical summaries for quantitative variables that lets us look at a group of observations at once.

```
psych::describe(nh_adults %>% select(Age, BMI, SBP, DBP, Pulse))
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
Age	1	500	41.9	12.3	42.0	41.9	16.31	21.0	64.0	43	0.03
BMI	2	495	28.5	6.3	27.5	27.8	5.63	17.3	63.3	46	1.35
SBP	3	485	119.2	15.3	118.0	118.2	13.34	84.0	209.0	125	1.27
DBP	4	485	72.1	11.1	72.0	72.3	8.90	0.0	103.0	103	-0.58
Pulse	5	485	73.4	12.0	72.0	73.0	11.86	40.0	112.0	72	0.30
			kurtosis	se							
Age			-1.20	0.55							
BMI			3.32	0.28							
SBP			4.63	0.69							
DBP			3.58	0.50							
Pulse			0.15	0.55							

The additional statistics presented here are:

- `trimmed` = a trimmed mean (by default in this function, this removes the top and bottom 10% from the data, then computes the mean of the remaining values - the middle 80% of the full data set.)
- `mad` = the median absolute deviation (from the median), which can be used in a manner similar to the standard deviation or IQR to measure spread.
 - If the data are Y_1, Y_2, \dots, Y_n , then the `mad` is defined as $\text{median}(|Y_i - \text{median}(Y_i)|)$.
 - To find the `mad` for a set of numbers, find the median, subtract the median from each value and find the absolute value of that difference, and then find the median of those absolute differences.
 - For non-normal data with a skewed shape but tails well approximated by the Normal, the `mad` is likely to be a better (more robust) estimate of the spread than is the standard deviation.
- a measure of `skew`, which refers to how much asymmetry is present in the shape of the distribution. The measure is not the same as the *nonparametric skew* measure that we will usually prefer. The [Wikipedia page on

5.5. MORE DETAILED NUMERICAL SUMMARIES FOR QUANTITATIVE VARIABLES85

skewness][<https://en.wikipedia.org/wiki/Skewness>] is very detailed.

- a measure of **kurtosis**, which refers to how outlier-prone, or heavy-tailed the shape of the distribution is, mainly as compared to a Normal distribution.
- **se** = the standard error of the sample mean, equal to the sample sd divided by the square root of the sample size.

5.5.4 **describe** in the **Hmisc** package

```
Hmisc::describe(nh_adults %>% select(Age, BMI, SBP, DBP, Pulse))
```

```
nh_adults %>% select(Age, BMI, SBP, DBP, Pulse)
```

```
5 Variables      500 Observations
```

Age

	n	missing	distinct	Info	Mean	Gmd	.05	.10
	500	0	44	0.999	41.91	14.27	23	25
	.25	.50	.75	.90	.95			
	31	42	53	59	61			

```
lowest : 21 22 23 24 25, highest: 60 61 62 63 64
```

BMI

	n	missing	distinct	Info	Mean	Gmd	.05	.10
	495	5	198	1	28.48	6.704	20.70	21.90
	.25	.50	.75	.90	.95			
	23.80	27.50	31.60	35.68	41.00			

```
lowest : 17.3 17.8 18.2 18.3 18.4, highest: 47.7 54.1 54.4 56.8 63.3
```

SBP

	n	missing	distinct	Info	Mean	Gmd	.05	.10
	485	15	73	0.999	119.2	16.18	98	102
	.25	.50	.75	.90	.95			
	110	118	127	137	143			

```
lowest : 84 85 91 92 93, highest: 170 171 182 202 209
```

DBP

	n	missing	distinct	Info	Mean	Gmd	.05	.10
	485	15	61	0.999	72.13	12.02	54.0	58.0
	.25	.50	.75	.90	.95			
	66.0	72.0	78.0	85.6	89.0			

```
lowest : 0 41 42 44 45, highest: 98 99 100 102 103
-----
Pulse
      n   missing  distinct     Info      Mean      Gmd      .05      .10
    485        15       35    0.997    73.41    13.47    54.4    60.0
    .25        .50       .75      .90      .95
    64.0      72.0      82.0    88.0    94.0
-----
lowest : 40 44 48 50 52, highest: 104 106 108 110 112
-----
```

The `Hmisc` package's version of `describe` for a distribution of data presents three new ideas, in addition to a more comprehensive list of quartiles (the 5th, 10th, 25th, 50th, 75th, 90th and 95th are shown) and the lowest and highest few observations. These are:

- `distinct` - the number of different values observed in the data.
- `Info` - a measure of how “continuous” the variable is, related to how many “ties” there are in the data, with `Info` taking a higher value (closer to its maximum of one) if the data are more continuous.
- `Gmd` - the Gini mean difference - a robust measure of spread that is calculated as the mean absolute difference between any pairs of observations. Larger values of `Gmd` indicate more spread-out distributions.

5.5.5 Other options

The package `summarytools` has a function called `dfSummary` which I like and Dominic Comtois has also published Recommendations for Using `summarytools` with R Markdown. Note that this isn't really for Word documents.

The `naniar` package is helpful for wrangling and visualizing missing values, and checking imputations.

As mentioned previously, `DataExplorer` can be used for more automated exploratory data analyses (and don't forget about `skimr`) and `visdat`, as well.

Chapter 6

Summarizing Categorical Variables

Summarizing categorical variables numerically is mostly about building tables, and calculating percentages or proportions. We'll save our discussion of modeling categorical data for later. Recall that in the `nh_adults` data set we built in Section 4.2 we had the following categorical variables. The number of levels indicates the number of possible categories for each categorical variable.

Variable	Description	Levels	Type
Sex	sex of subject	2	binary
Race	subject's race	6	nominal
Education	subject's educational level	5	ordinal
PhysActive	Participates in sports?	2	binary
Smoke100	Smoked 100+ cigarettes?	2	binary
SleepTrouble	Trouble sleeping?	2	binary
HealthGen	Self-report health	5	ordinal

6.1 The `summary` function for Categorical data

When R recognizes a variable as categorical, it stores it as a *factor*. Such variables get special treatment from the `summary` function, in particular a table of available values (so long as there aren't too many.)

```
nh_adults %>%
  select(Sex, Race, Education, PhysActive, Smoke100,
         SleepTrouble, HealthGen, MaritalStatus) %>%
  summary()
```

	Sex	Race	Education	PhysActive	Smoke100
female:221	Asian : 42	8th Grade : 24	No :215	No :297	
male :279	Black : 63	9 - 11th Grade: 60	Yes:285	Yes:203	
	Hispanic: 26	High School : 81			
	Mexican : 38	Some College :153			
	White :313	College Grad :182			
	Other : 18				
	SleepTrouble	HealthGen	MaritalStatus		
No :380	Excellent: 50	Divorced : 51			
Yes:120	Vgood :154	LivePartner : 51			
	Good :184	Married :259			
	Fair : 49	NeverMarried:112			
	Poor : 14	Separated : 16			
	NA's : 49	Widowed : 11			

6.2 Tables to describe One Categorical Variable

Suppose we build a table (using the `tabyl` function from the `janitor` package) to describe the `HealthGen` distribution.

```
nh_adults %>%
  tabyl(HealthGen) %>%
  adorn_pct_formatting()
```

HealthGen	n	percent	valid_percent
Excellent	50	10.0%	11.1%
Vgood	154	30.8%	34.1%
Good	184	36.8%	40.8%
Fair	49	9.8%	10.9%
Poor	14	2.8%	3.1%
<NA>	49	9.8%	-

Note how the missing (<NA>) values are not included in the `valid_percent` calculation, but are in the `percent` calculation. Note also the use of percentage formatting.

What if we want to add a total count, sometimes called the *marginal* total?

```
nh_adults %>%
  tabyl(HealthGen) %>%
  adorn_totals() %>%
  adorn_pct_formatting()
```

HealthGen	n	percent	valid_percent
Excellent	50	10.0%	11.1%

Vgood	154	30.8%	34.1%
Good	184	36.8%	40.8%
Fair	49	9.8%	10.9%
Poor	14	2.8%	3.1%
<NA>	49	9.8%	-
Total	500	100.0%	100.0%

What about marital status, which has no missing data in our sample?

```
nh_adults %>%
  tabyl(MaritalStatus) %>%
  adorn_totals() %>%
  adorn_pct_formatting()
```

	MaritalStatus	n	percent
	Divorced	51	10.2%
	LivePartner	51	10.2%
	Married	259	51.8%
	NeverMarried	112	22.4%
	Separated	16	3.2%
	Widowed	11	2.2%
	Total	500	100.0%

6.3 The Mode of a Categorical Variable

A common measure applied to a categorical variable is to identify the mode, the most frequently observed value. To find the mode for variables with lots of categories (so that the `summary` may not be sufficient), we usually tabulate the data, and then sort by the counts of the numbers of observations, as we did with discrete quantitative variables.

```
nh_adults %>%
  group_by(HealthGen) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
Warning: Factor `HealthGen` contains implicit NA, consider using
`forcats::fct_explicit_na`
```

```
# A tibble: 6 x 2
  HealthGen count
  <fct>     <int>
1 Good       184
2 Vgood      154
3 Excellent   50
4 Fair        49
```

5 <NA>	49
6 Poor	14

6.4 `describe` in the `Hmisc` package

```
Hmisc::describe(nh_adults %>%
  select(Sex, Race, Education, PhysActive,
  Smoke100, SleepTrouble,
  HealthGen, MaritalStatus))

nh_adults %>% select(Sex, Race, Education, PhysActive, Smoke100, SleepTrouble, HealthG

  8 Variables      500 Observations
-----
Sex
  n   missing distinct
  500       0       2

  Value     female    male
  Frequency     221     279
  Proportion   0.442   0.558
-----
Race
  n   missing distinct
  500       0       6

  Value      Asian    Black Hispanic Mexican    White    Other
  Frequency      42      63      26      38      313      18
  Proportion   0.084   0.126   0.052   0.076   0.626   0.036
-----
Education
  n   missing distinct
  500       0       5

  Value      8th Grade 9 - 11th Grade    High School    Some College
  Frequency      24                  60                  81                  153
  Proportion   0.048                0.120                0.162                0.306
-----
PhysActive
  n   missing distinct
```

500	0	2
-----	---	---

Value	No	Yes
Frequency	215	285
Proportion	0.43	0.57

Smoke100		
n	missing	distinct
500	0	2

Value	No	Yes
Frequency	297	203
Proportion	0.594	0.406

SleepTrouble		
n	missing	distinct
500	0	2

Value	No	Yes
Frequency	380	120
Proportion	0.76	0.24

HealthGen		
n	missing	distinct
451	49	5

Value	Excellent	Vgood	Good	Fair	Poor
Frequency	50	154	184	49	14
Proportion	0.111	0.341	0.408	0.109	0.031

MaritalStatus		
n	missing	distinct
500	0	6

Value	Divorced	LivePartner	Married	NeverMarried
Frequency	51	51	259	112
Proportion	0.102	0.102	0.518	0.224

Value	Separated	Widowed
Frequency	16	11
Proportion	0.032	0.022

6.5 Cross-Tabulations

It is very common for us to want to describe the association of one categorical variable with another. For instance, is there a relationship between Education and SleepTrouble in these data?

```
nh_adults %>%
  tabyl(Education, SleepTrouble) %>%
  adorn_totals(where = c("row", "col"))
```

	Education	No	Yes	Total
8th Grade	18	6	24	
9 - 11th Grade	45	15	60	
High School	62	19	81	
Some College	118	35	153	
College Grad	137	45	182	
Total	380	120	500	

Note the use of `adorn_totals` to get the marginal counts, and how we specify that we want both the row and column totals. We can add a title for the columns with...

```
nh_adults %>%
  tabyl(Education, SleepTrouble) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_title(placement = "combined")
```

Education/SleepTrouble	No	Yes	Total
8th Grade	18	6	24
9 - 11th Grade	45	15	60
High School	62	19	81
Some College	118	35	153
College Grad	137	45	182
Total	380	120	500

Often, we'll want to show percentages in a cross-tabulation like this. To get row percentages so that we can directly see the probability of `SleepTrouble = Yes` for each level of `Education`, we can use:

```
nh_adults %>%
  tabyl(Education, SleepTrouble) %>%
  adorn_totals(where = "row") %>%
  adorn_percentages(denominator = "row") %>%
  adorn_pct_formatting() %>%
  adorn_title(placement = "combined")
```

Education/SleepTrouble	No	Yes
8th Grade	75.0%	25.0%

9 - 11th Grade	75.0%	25.0%
High School	76.5%	23.5%
Some College	77.1%	22.9%
College Grad	75.3%	24.7%
Total	76.0%	24.0%

If we want to compare the distribution of Education between the two levels of SleepTrouble with column percentages, we can use the following...

```
nh_adults %>%
  tabyl(Education, SleepTrouble) %>%
  adorn_totals(where = "col") %>%
  adorn_percentages(denominator = "col") %>%
  adorn_pct_formatting() %>%
  adorn_title(placement = "combined")
```

Education/SleepTrouble	No	Yes	Total
8th Grade	4.7%	5.0%	4.8%
9 - 11th Grade	11.8%	12.5%	12.0%
High School	16.3%	15.8%	16.2%
Some College	31.1%	29.2%	30.6%
College Grad	36.1%	37.5%	36.4%

If we want overall percentages in the cells of the table, so that the total across all combinations of Education and SleepTrouble is 100%, we can use:

```
nh_adults %>%
  tabyl(Education, SleepTrouble) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_percentages(denominator = "all") %>%
  adorn_pct_formatting() %>%
  adorn_title(placement = "combined")
```

Education/SleepTrouble	No	Yes	Total
8th Grade	3.6%	1.2%	4.8%
9 - 11th Grade	9.0%	3.0%	12.0%
High School	12.4%	3.8%	16.2%
Some College	23.6%	7.0%	30.6%
College Grad	27.4%	9.0%	36.4%
Total	76.0%	24.0%	100.0%

Another common approach is to include both counts and percentages in a cross-tabulation. Let's look at the breakdown of HealthGen by MaritalStatus.

```
nh_adults %>%
  tabyl(MaritalStatus, HealthGen) %>%
  adorn_totals(where = c("row")) %>%
  adorn_percentages(denominator = "row") %>%
  adorn_pct_formatting() %>%
```

```
adorn_ns(position = "front") %>%
  adorn_title(placement = "combined") %>%
  knitr::kable()
```

MaritalStatus/HealthGen	Excellent	Vgood	Good	Fair	Poor	NA
Divorced	7 (13.7%)	14 (27.5%)	20 (39.2%)	5 (9.8%)	2 (3.9%)	3 (5.9%)
LivePartner	1 (2.0%)	18 (35.3%)	16 (31.4%)	11 (21.6%)	1 (2.0%)	4 (7.8%)
Married	23 (8.9%)	84 (32.4%)	102 (39.4%)	15 (5.8%)	4 (1.5%)	31 (11.1%)
NeverMarried	14 (12.5%)	31 (27.7%)	43 (38.4%)	13 (11.6%)	3 (2.7%)	8 (7.1%)
Separated	4 (25.0%)	4 (25.0%)	1 (6.2%)	4 (25.0%)	1 (6.2%)	2 (12.5%)
Widowed	1 (9.1%)	3 (27.3%)	2 (18.2%)	1 (9.1%)	3 (27.3%)	1 (9.1%)
Total	50 (10.0%)	154 (30.8%)	184 (36.8%)	49 (9.8%)	14 (2.8%)	49 (9.8%)

What if we wanted to ignore the missing HealthGen values? Most often, I filter down to the complete observations.

```
nh_adults %>%
  filter(complete.cases(MaritalStatus, HealthGen)) %>%
  tabyl(MaritalStatus, HealthGen) %>%
  adorn_totals(where = c("row")) %>%
  adorn_percentages(denominator = "row") %>%
  adorn_pct_formatting() %>%
  adorn_ns(position = "front") %>%
  adorn_title(placement = "combined")
```

MaritalStatus/HealthGen	Excellent	Vgood	Good	Fair
Divorced	7 (14.6%)	14 (29.2%)	20 (41.7%)	5 (10.4%)
LivePartner	1 (2.1%)	18 (38.3%)	16 (34.0%)	11 (23.4%)
Married	23 (10.1%)	84 (36.8%)	102 (44.7%)	15 (6.6%)
NeverMarried	14 (13.5%)	31 (29.8%)	43 (41.3%)	13 (12.5%)
Separated	4 (28.6%)	4 (28.6%)	1 (7.1%)	4 (28.6%)
Widowed	1 (10.0%)	3 (30.0%)	2 (20.0%)	1 (10.0%)
Total	50 (11.1%)	154 (34.1%)	184 (40.8%)	49 (10.9%)
Poor				
2 (4.2%)				
1 (2.1%)				
4 (1.8%)				
3 (2.9%)				
1 (7.1%)				
3 (30.0%)				
14 (3.1%)				

For more on working with `tabyls`, see the vignette in the `janitor` package. There you'll find a complete list of all of the `adorn` functions, for example.

Here's another approach, to look at the cross-classification of Race and HealthGen:

```
xtabs(~ Race + HealthGen, data = nh_adults)
```

Race	HealthGen				
	Excellent	Vgood	Good	Fair	Poor
Asian	3	11	17	3	0
Black	8	11	19	11	6
Hispanic	3	3	11	4	1
Mexican	2	8	17	6	3
White	33	113	114	22	4
Other	1	8	6	3	0

6.5.1 Cross-Classifying Three Categorical Variables

Suppose we are interested in `Smoke100` and its relationship to `PhysActive` and `SleepTrouble`.

```
nh_adults %>%
  tabyl(Smoke100, PhysActive, SleepTrouble) %>%
  adorn_title(placement = "top")
```

\$No

PhysActive		
Smoke100	No	Yes
No	99	142
Yes	62	77

\$Yes

PhysActive		
Smoke100	No	Yes
No	21	35
Yes	33	31

The result here is a tabyl of `Smoke100` (rows) by `PhysActive` (columns), split into a list by `SleepTrouble`. Another approach to get the same table is:

```
xtabs(~ Smoke100 + PhysActive + SleepTrouble, data = nh_adults)

, , SleepTrouble = No

  PhysActive
Smoke100 No Yes
  No    99 142
  Yes   62  77

, , SleepTrouble = Yes
```

PhysActive		
Smoke100	No	Yes
No	21	35
Yes	33	31

We can also build a **flat** version of this table, as follows:

```
ftable(Smoke100 ~ PhysActive + SleepTrouble, data = nh_adults)
```

			Smoke100	No	Yes
PhysActive	SleepTrouble				
No	No		99	62	
	Yes		21	33	
Yes	No		142	77	
	Yes		35	31	

And we can do this with **dplyr** functions, as well, for example...

```
nh_adults %>%
  select(Smoke100, PhysActive, SleepTrouble) %>%
  table()
```

```
, , SleepTrouble = No
```

PhysActive		
Smoke100	No	Yes
No	99	142
Yes	62	77

```
, , SleepTrouble = Yes
```

PhysActive		
Smoke100	No	Yes
No	21	35
Yes	33	31

6.6 Constructing Tables Well

The prolific Howard Wainer is responsible for many interesting books on visualization and related issues, including Wainer (2005) and Wainer (2013). These rules come from Chapter 10 of Wainer (1997).

1. Order the rows and columns in a way that makes sense.
2. Round, a lot!
3. ALL is different and important

6.6.1 Alabama First!

Which of these Tables is more useful to you?

2013 Percent of Students in grades 9-12 who are obese

State	% Obese	95% CI	Sample Size
Alabama	17.1	(14.6 - 19.9)	1,499
Alaska	12.4	(10.5-14.6)	1,167
Arizona	10.7	(8.3-13.6)	1,520
Arkansas	17.8	(15.7-20.1)	1,470
Connecticut	12.3	(10.2-14.7)	2,270
Delaware	14.2	(12.9-15.6)	2,475
Florida	11.6	(10.5-12.8)	5,491
...			
Wisconsin	11.6	(9.7-13.9)	2,771
Wyoming	10.7	(9.4-12.2)	2,910

or ...

State	% Obese	95% CI	Sample Size
Kentucky	18.0	(15.7 - 20.6)	1,537
Arkansas	17.8	(15.7 - 20.1)	1,470
Alabama	17.1	(14.6 - 19.9)	1,499
Tennessee	16.9	(15.1 - 18.8)	1,831
Texas	15.7	(13.9 - 17.6)	3,039
...			
Massachusetts	10.2	(8.5 - 12.1)	2,547
Idaho	9.6	(8.2 - 11.1)	1,841
Montana	9.4	(8.4 - 10.5)	4,679
New Jersey	8.7	(6.8 - 11.2)	1,644
Utah	6.4	(4.8 - 8.5)	2,136

It is a rare event when Alabama first is the best choice.

6.6.2 Order rows and columns sensibly

- Alabama First!
 - Size places - put the largest first. We often look most carefully at the top.
- Order time from the past to the future to help the viewer.

- If there is a clear predictor-outcome relationship, put the predictors in the rows and the outcomes in the columns.

6.6.3 Round - a lot!

- Humans cannot understand more than two digits very easily.
- We almost never care about accuracy of more than two digits.
- We can almost never justify more than two digits of accuracy statistically.
- It's also helpful to remember that we are almost invariably publishing progress to date, rather than a truly final answer.

Suppose, for instance, we report a correlation coefficient of 0.25. How many observations do you think you would need to justify such a choice?

- To report 0.25 meaningfully, we want to be sure that the second digit isn't 4 or 6.
- That requires a standard error less than 0.005
- The *standard error* of any statistic is proportional to 1 over the square root of the sample size, n .

So $\frac{1}{\sqrt{n}} \sim 0.005$, but that means $\sqrt{n} = \frac{1}{0.005} = 200$. If $\sqrt{n} = 200$, then $n = (200)^2 = 40,000$.

Do we usually have 40,000 observations?

6.6.4 ALL is different and important

Summaries of rows and columns provide a measure of what is typical or usual. Sometimes a sum is helpful, at other times, consider presenting a median or other summary. The ALL category, as Wainer (1997) suggests, should be both visually different from the individual entries and set spatially apart.

On the whole, it's *far* easier to fall into a good graph in R (at least if you have some ggplot2 skills) than to produce a good table.

Chapter 7

NHANES National Youth Fitness Survey (`nnyfs`)

The `nnyfs.csv` and the `nnyfs.Rds` data files were built by Professor Love using data from the 2012 National Youth Fitness Survey.

The NHANES National Youth Fitness Survey (NNYFS) was conducted in 2012 to collect data on physical activity and fitness levels in order to provide an evaluation of the health and fitness of children in the U.S. ages 3 to 15. The NNYFS collected data on physical activity and fitness levels of our youth through interviews and fitness tests.

In the `nnyfs` data file (either `.csv` or `.Rds`), I'm only providing a modest fraction of the available information. More on the NNYFS (including information I'm not using) is available at <https://www.cdc.gov/nchs/nhanes/search/nnyfs12.aspx>.

The data elements I'm using fall into four main groups, or components:

- Demographics
- Dietary
- Examination and
- Questionnaire

What I did was merge a few elements from each of the available components of the NHANES National Youth Fitness Survey, reformulated (and in some cases simplified) some variables, and restricted the sample to kids who had completed elements of each of the four components.

7.1 The Variables included in `nnyfs`

This section tells you where the data come from, and briefly describe what is collected.

7.1.1 From the NNYFS Demographic Component

All of these come from the `Y_DEMO` file.

In <code>nnyfs</code>	In <code>Y_DEMO</code>	Description
<code>SEQN</code>	<code>SEQN</code>	Subject ID, connects all of the files
<code>sex</code>	<code>RIAGENDR</code>	Really, this is sex, not gender
<code>age_child</code>	<code>RIDAGEYR</code>	Age in years at screening
<code>race_eth</code>	<code>RIDRETH1</code>	Race/Hispanic origin (collapsed to 4 levels)
<code>educ_child</code>	<code>DMDEDUC3</code>	Education Level (for children ages 6-15). 0 = Kindergarten, 9 = Ninth grade or higher
<code>language</code>	<code>SIALANG</code>	Language in which the interview was conducted
<code>sampling_wt</code>	<code>WTMEC</code>	Full-sample MEC exam weight (for inference)
<code>income_pov</code>	<code>INDFMPIR</code>	Ratio of family income to poverty (ceiling is 5.0)
<code>age_adult</code>	<code>DMDHRAGE</code>	Age of adult who brought child to interview
<code>educ_adult</code>	<code>DMDHREDU</code>	Education level of adult who brought child

7.1.2 From the NNYFS Dietary Component

From the `Y_DR1TOT` file, we have a number of variables related to the child's diet, with the following summaries mostly describing consumption "yesterday" in a dietary recall questionnaire.

In <code>nnyfs</code>	In <code>Y_DR1TOT</code>	Description
<code>respondent</code>	<code>DR1MNRSP</code>	who responded to interview (child, Mom, someone else)
<code>salt_used</code>	<code>DBQ095Z</code>	uses salt, lite salt or salt substitute at the table
<code>energy</code>	<code>DR1TKCAL</code>	energy consumed (kcal)
<code>protein</code>	<code>DR1TPROT</code>	protein consumed (g)
<code>sugar</code>	<code>DR1TSUGR</code>	total sugar consumed (g)
<code>fat</code>	<code>DR1TTFAT</code>	total fat consumed (g)
<code>diet_yesterday</code>	<code>DR1_300</code>	compare food consumed yesterday to usual amount
<code>water</code>	<code>DR1_320Z</code>	total plain water drank (g)

7.1.3 From the NNYFS Examination Component

From the Y_BMX file of Body Measures:

In nnyfs	In Y_BMX	Description
height	BMXHT	standing height (cm)
weight	BMXWT	weight (kg)
bmi	BMXBMI	body mass index (kg/m^2)
bmi_cat	BMDBMIC	BMI category (4 levels)
arm_length	BMXARML	Upper arm length (cm)
waist	BMXWAIST	Waist circumference (cm)
arm_circ	BMXARMC	Arm circumference (cm)
calf_circ	BMXCALF	Maximal calf circumference (cm)
calf_s Skinfold	BMXCALFF	Calf skinfold (mm)
triceps_s Skinfold	BMXTRI	Triceps skinfold (mm)
subscapular_s Skinfold	BMXSUB	Subscapular skinfold (mm)

From the Y_PLX file of Plank test results:

In nnyfs	In Y_PLX	Description
plank_time	MPXPLANK	# of seconds plank position is held

7.1.4 From the NNYFS Questionnaire Component

From the Y_PAQ file of Physical Activity questions:

In nnyfs	In Y_PAQ	Description
active_days	PAQ706	Days physically active (≥ 60 min.) in past week
tv_hours	PAQ710	Average hours watching TV/videos past 30d
computer_hours	PAQ715	Average hours on computer past 30d
physical_last_week	PAQ722	Any physical activity outside of school past week
enjoy_recess	PAQ750	Enjoy participating in PE/recess

From the Y_DBQ file of Diet Behavior and Nutrition questions:

In nnyfs	In Y_DBQ	Description
meals_out	DBD895	# meals not home-prepared in past 7 days

102CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

From the `Y_HIQ` file of Health Insurance questions:

In <code>nnyfs</code>	In <code>Y_HIQ</code>	Description
<code>insured</code>	HIQ011	Covered by Health Insurance?
<code>insurance</code>	HIQ031	Type of Health Insurance coverage

From the `Y_HUQ` file of Access to Care questions:

In <code>nnyfs</code>	In <code>Y_HUQ</code>	Description
<code>phys_health</code>	HUQ010	General health condition (Excellent - Poor)
<code>access_to_care</code>	HUQ030	Routine place to get care?
<code>care_source</code>	HUQ040	Type of place most often goes to for care

From the `Y_MCQ` file of Medical Conditions questions:

In <code>nnyfs</code>	In <code>Y_MCQ</code>	Description
<code>asthma_ever</code>	MCQ010	Ever told you have asthma?
<code>asthma_now</code>	MCQ035	Still have asthma?

From the `Y_RXQ_RX` file of Prescription Medication questions:

In <code>nnyfs</code>	In <code>Y_RXQ_RX</code>	Description
<code>med_use</code>	RXDUSE	Taken prescription medication in last month?
<code>med_count</code>	RXDCOUNT	# of prescription meds taken in past month

7.2 Looking over the Data Set

Now, I'll take a look at the `nnyfs` data, which I've made available in a comma-separated version (`nnyfs.csv`), if you prefer, as well as in an R data set (`nnyfs.Rds`) which loads a bit faster. After loading the file, let's get a handle on its size and contents.

```
nnyfs <- readRDS("data/nnyfs.Rds") %>% as_tibble()

## size of the tibble
dim(nnyfs)
```

[1] 1518 45

There are 1518 rows (subjects) and 45 columns (variables), by which I mean that there are 1518 kids in the `nnyfs` data frame, and we have 45 pieces of information on each subject. So, what do we have, exactly?

```
nnyfs # this is a tibble, has some nice features in a print-out like this
```

```
# A tibble: 1,518 x 45
  SEQN sex    age_child race_eth educ_child language sampling_wt
  <dbl> <chr>   <dbl> <chr>    <dbl> <chr>      <dbl>
1 71917 Fema~     15 3_Black~      9 English    28299.
2 71918 Fema~     8 3_Black~       2 English    15127.
3 71919 Fema~     14 2_White~      8 English    29977.
4 71920 Fema~     15 2_White~      8 English    80652.
5 71921 Male     3 2_White~      NA English   55592.
6 71922 Male     12 1_Hispa~      6 English    27365.
7 71923 Male     12 2_White~      5 English    86673.
8 71924 Fema~     8 4_Other~       2 English   39549.
9 71925 Male     7 1_Hispa~       0 English   42333.
10 71926 Male    8 3_Black~      2 English   15307.
# ... with 1,508 more rows, and 38 more variables: income_pov <dbl>,
#   age_adult <dbl>, educ_adult <chr>, respondent <chr>, salt_used <chr>,
#   energy <dbl>, protein <dbl>, sugar <dbl>, fat <dbl>,
#   diet_yesterday <chr>, water <dbl>, plank_time <dbl>, height <dbl>,
#   weight <dbl>, bmi <dbl>, bmi_cat <chr>, arm_length <dbl>, waist <dbl>,
#   arm_circ <dbl>, calf_circ <dbl>, calf_s Skinfold <dbl>,
#   triceps_s Skinfold <dbl>, subscapular_s Skinfold <dbl>, active_days <dbl>,
#   tv_hours <dbl>, computer_hours <dbl>, physical_last_week <chr>,
#   enjoy_recess <chr>, meals_out <dbl>, insured <chr>, phys_health <chr>,
#   access_to_care <chr>, care_source <chr>, asthma_ever <chr>,
#   asthma_now <chr>, med_use <chr>, med_count <dbl>, insurance <chr>
```

Tibbles are a modern reimagining of the main way in which people have stored data in R, called a data frame. Tibbles were developed to keep what time has proven to be effective, and throwing out what is not. We can learn something about the structure of the tibble from such functions as `str` or `glimpse`.

```
str(nnyfs)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame': 1518 obs. of 45 variables:
$ SEQN           : num  71917 71918 71919 71920 71921 ...
$ sex            : chr "Female" "Female" "Female" "Female" ...
$ age_child      : num  15 8 14 15 3 12 12 8 7 8 ...
$ race_eth       : chr "3_Black Non-Hispanic" "3_Black Non-Hispanic" "2_White Non-Hispanic"
$ educ_child     : num  9 2 8 8 NA 6 5 2 0 2 ...
$ language        : chr "English" "English" "English" "English" ...
$ sampling_wt    : num  28299 15127 29977 80652 55592 ...
$ income_pov     : num  0.21 5 5 0.87 4.34 5 5 2.74 0.46 1.57 ...
$ age_adult      : num  46 46 42 53 31 42 39 31 45 56 ...
```

104 CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

```

$ educ_adult      : chr  "2_9-11th Grade" "3_High School Graduate" "5_College Grad...
$ respondent     : chr  "Child" "Mom" "Child" "Child" ...
$ salt_used       : chr  "Yes" "Yes" "Yes" "Yes" ...
$ energy          : num   2844 1725 2304 1114 1655 ...
$ protein         : num   169.1 55.2 199.3 14 50.6 ...
$ sugar           : num   128.2 118.7 81.4 119.2 90.3 ...
$ fat              : num   127.9 63.7 86.1 36 53.3 ...
$ diet_yesterday  : chr  "2_Usual" "2_Usual" "2_Usual" "2_Usual" ...
$ water            : num   607 178 503 859 148 ...
$ plank_time      : num   NA 45 121 45 11 107 127 44 184 58 ...
$ height          : num   NA 131.6 172 167.1 90.2 ...
$ weight          : num   NA 38.6 58.7 92.5 12.4 66.4 56.7 22.2 20.9 28.3 ...
$ bmi              : num   NA 22.3 19.8 33.1 15.2 25.9 22.5 14.4 15.9 17 ...
$ bmi_cat         : chr  NA "4_Obese" "2_Normal" "4_Obese" ...
$ arm_length      : num   NA 27.7 38.4 35.9 18.3 34.2 33 26.5 24.2 26 ...
$ waist            : num   NA 71.9 79.4 96.4 46.8 90 72.3 56.1 54.5 59.7 ...
$ arm_circ        : num   NA 25.4 26 37.9 15.1 29.5 27.9 17.6 17.7 19.9 ...
$ calf_circ       : num   NA 32.3 35.3 46.8 19.4 36.9 36.8 24 24.3 27.3 ...
$ calf_skinfold   : num   NA 22 18.4 NA 8.4 22 18.3 7 7.2 8.2 ...
$ triceps_skinfold: num   NA 19.9 15 20.6 8.6 22.8 20.5 12.9 6.9 8.8 ...
$ subscapular_skinfold: num   NA 17.4 9.8 22.8 5.7 24.4 12.6 6.8 4.8 6.1 ...
$ active_days     : num   3 5 3 3 7 2 5 3 7 7 ...
$ tv_hours         : num   2 2 1 3 2 3 0 4 2 2 ...
$ computer_hours   : num   1 2 3 3 0 1 0 3 1 1 ...
$ physical_last_week: chr  "No" "No" "Yes" "Yes" ...
$ enjoy_recess    : chr  "1_Strongly Agree" "1_Strongly Agree" "3_Neither Agree n...
$ meals_out        : num   0 2 3 2 1 1 2 1 0 2 ...
$ insured          : chr  "Has Insurance" "Has Insurance" "Has Insurance" "Has Ins...
$ phys_health      : chr  "1_Excellent" "3_Good" "1_Excellent" "3_Good" ...
$ access_to_care   : chr  "Has Usual Care Source" "Has Usual Care Source" "Has Us...
$ care_source      : chr  "Clinic or Health Center" "Doctor's Office" "Doctor's Off...
$ asthma_ever      : chr  "Never Had Asthma" "History of Asthma" "Never Had Asthma...
$ asthma_now       : chr  "No Asthma Now" "Asthma Now" "No Asthma Now" "Asthma Now...
$ med_use          : chr  "No Medications" "Had Medication" "No Medications" "Had M...
$ med_count        : num   0 1 0 2 0 0 0 0 0 0 ...
$ insurance        : chr  "State Sponsored" "State Sponsored" "Private" "State Spon...
```

There are a lot of variables here. Let's run through the first few rather slowly, and then we'll speed up.

7.2.1 SEQN

The first variable, **SEQN** is just a (numerical) identifying code attributable to a given subject of the survey. This is *nominal* data, which will be of little interest down the line. On some occasions, as in this case, the ID numbers are sequential,

in the sense that subject 71919 was included in the data base after subject 71918, but this fact isn't particularly interesting here, because the protocol remained unchanged throughout the study.

7.2.2 sex

The second variable, `sex`, is listed as a character variable (R uses `factor` and `character` to refer to categorical, especially non-numeric information). Here, as we can see below, we have two levels, *Female* and *Male*.

```
nyfs %>%
  tabyl(sex) %>%
  adorn_totals() %>%
  adorn_pct_formatting()
```

sex	n	percent
Female	760	50.1%
Male	758	49.9%
Total	1518	100.0%

Obviously, we don't actually need more than a decimal place here for any real purpose.

7.2.3 age_child

The third variable, `age_child`, is the age of the child at the time of their screening to be in the study, measured in years. Note that age is a continuous concept, but the measure used here (number of full years alive) is a common discrete approach to measurement. Age, of course, has a meaningful zero point, so this can be thought of as a ratio variable; a child who is 6 is half as old as one who is 12. We can tabulate the observed values, since there are only a dozen or so.

```
nyfs %>% tabyl(age_child) %>%
  adorn_pct_formatting()
```

age_child	n	percent
3	110	7.2%
4	112	7.4%
5	114	7.5%
6	129	8.5%
7	123	8.1%
8	112	7.4%
9	99	6.5%
10	124	8.2%
11	111	7.3%

12	137	9.0%
13	119	7.8%
14	130	8.6%
15	98	6.5%

At the time of initial screening, these children should have been between 3 and 15 years of age, so things look reasonable. Since this is a meaningful quantitative variable, we may be interested in a more descriptive summary.

```
nnyfs %>% select(age_child) %>%
  summary()
```

```
age_child
Min.    : 3.00
1st Qu.: 6.00
Median  : 9.00
Mean    : 9.03
3rd Qu.:12.00
Max.    :15.00
```

These six numbers provide a nice, if incomplete, look at the ages.

- Min. = the minimum, or youngest age at the examination was 3 years old.
- 1st Qu. = the first quartile (25th percentile) of the ages was 6. This means that 25 percent of the subjects were age 6 or less.
- Median = the second quartile (50th percentile) of the ages was 9. This is often used to describe the center of the data. Half of the subjects were age 9 or less.
- 3rd Qu. = the third quartile (75th percentile) of the ages was 12
- Max. = the maximum, or oldest age at the examination was 15 years.

We could get the standard deviation and a count of missing and non-missing observations with `favstats` from the `mosaic` package.

```
mosaic::favstats(~ age_child, data = nnyfs)
```

```
min Q1 median Q3 max mean   sd     n missing
 3   6       9 12   15 9.03 3.71 1518      0
```

7.2.4 race_eth

The fourth variable in the data set is `race_eth`, which is a multi-categorical variable describing the child's race and ethnicity.

```
nnyfs %>% tabyl(race_eth) %>%
  adorn_pct_formatting() %>%
  knitr::kable()
```

race_eth	n	percent
1_Hispanic	450	29.6%
2_White Non-Hispanic	610	40.2%
3_Black Non-Hispanic	338	22.3%
4_Other Race/Ethnicity	120	7.9%

And now, we get the idea of looking at whether our numerical summaries of the children's ages varies by their race/ethnicity...

```
mosaic::favstats(age_child ~ race_eth, data = nnyfs)
```

	race_eth	min	Q1	median	Q3	max	mean	sd	n	missing
1	1_Hispanic	3	5.25	9.0	12	15	8.79	3.73	450	0
2	2_White Non-Hispanic	3	6.00	9.0	12	15	9.14	3.80	610	0
3	3_Black Non-Hispanic	3	6.00	9.0	12	15	9.04	3.58	338	0
4	4_Other Race/Ethnicity	3	7.00	9.5	12	15	9.38	3.43	120	0

7.2.5 income_pov

Skipping down a bit, let's look at the family income as a multiple of the poverty level. Here's the summary.

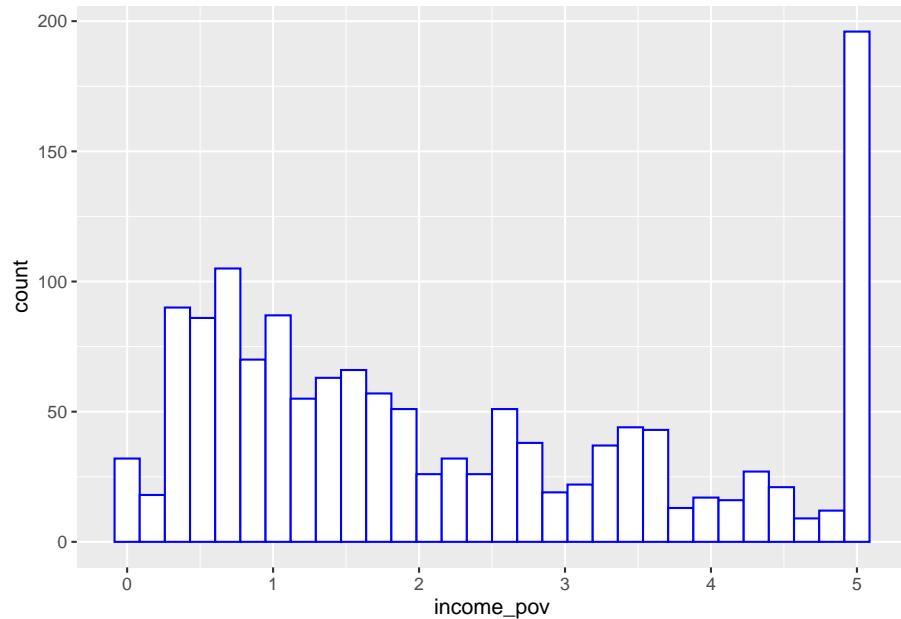
```
nnyfs %>% select(income_pov) %>% summary()
```

```
income_pov
Min.    :0.0
1st Qu.:0.9
Median  :1.7
Mean    :2.2
3rd Qu.:3.5
Max.    :5.0
NA's    :89
```

We see there is some missing data here. Let's ignore that for the moment and concentrate on interpreting the results for the children with actual data. We should start with a picture.

```
ggplot(nnyfs, aes(x = income_pov)) +
  geom_histogram(bins = 30, fill = "white", col = "blue")
```

Warning: Removed 89 rows containing non-finite values (stat_bin).



The histogram shows us that the values are truncated at 5, so that children whose actual family income is above 5 times the poverty line are listed as 5. We also see a message reminding us that some of the data are missing for this variable.

Is there a relationship between `income_pov` and `race_eth` in these data?

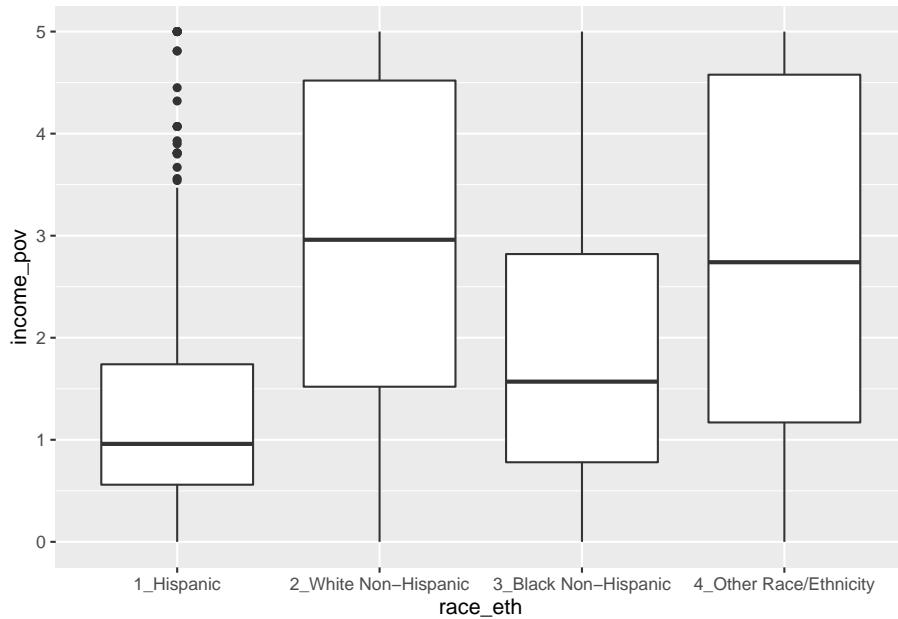
```
mosaic::favstats(income_pov ~ race_eth, data = nnyfs)
```

	race_eth	min	Q1	median	Q3	max	mean	sd	n	missing
1	1_Hispanic	0	0.56	0.96	1.74	5	1.34	1.10	409	41
2	2_White Non-Hispanic	0	1.52	2.96	4.52	5	2.92	1.62	588	22
3	3_Black Non-Hispanic	0	0.78	1.57	2.82	5	1.97	1.49	328	10
4	4_Other Race/Ethnicity	0	1.17	2.74	4.58	5	2.85	1.67	104	16

This deserves a picture. Let's try a boxplot.

```
ggplot(nnyfs, aes(x = race_eth, y = income_pov)) +
  geom_boxplot()
```

Warning: Removed 89 rows containing non-finite values (stat_boxplot).



7.2.6 bmi

Moving into the body measurement data, `bmi` is the body-mass index of the child. The BMI is a person's weight in kilograms divided by his or her height in meters squared. Symbolically, $BMI = \text{weight in kg} / (\text{height in m})^2$. This is a continuous concept, measured to as many decimal places as you like, and it has a meaningful zero point, so it's a ratio variable.

```
nnyfs %>% select(bmi) %>% summary()
```

```
bmi
Min.   :11.9
1st Qu.:15.9
Median :18.1
Mean   :19.6
3rd Qu.:21.9
Max.   :48.3
NA's    :4
```

Why would a table of these BMI values not be a great idea, for these data? A hint is that R represents this variable as `num` or numeric in its depiction of the data structure, and this implies that R has some decimal values stored. Here, I'll use the `head()` function and the `tail()` function to show the first few and the last few values of what would prove to be a very long table of `bmi` values.

```
nnyfs %>% tabyl(bmi) %>%
  adorn_pct_formatting() %>%
  head()
```

bmi	n	percent	valid_percent
11.9	1	0.1%	0.1%
12.6	1	0.1%	0.1%
12.7	1	0.1%	0.1%
12.9	1	0.1%	0.1%
13.0	2	0.1%	0.1%
13.1	1	0.1%	0.1%

```
nnyfs %>% tabyl(bmi) %>%
  adorn_pct_formatting() %>%
  tail()
```

bmi	n	percent	valid_percent
42.8	1	0.1%	0.1%
43.0	1	0.1%	0.1%
46.9	1	0.1%	0.1%
48.2	1	0.1%	0.1%
48.3	1	0.1%	0.1%
NA	4	0.3%	-

7.2.7 bmi_cat

Next I'll look at the `bmi_cat` information. This is a four-category ordinal variable, which divides the sample according to BMI into four groups. The BMI categories use sex-specific 2000 BMI-for-age (in months) growth charts prepared by the Centers for Disease Control for the US. We can get the breakdown from a table of the variable's values.

```
nnyfs %>% tabyl(bmi_cat) %>% adorn_pct_formatting()
```

bmi_cat	n	percent	valid_percent
1_Underweight	41	2.7%	2.7%
2_Normal	920	60.6%	60.8%
3_Overweight	258	17.0%	17.0%
4_Obese	295	19.4%	19.5%
<NA>	4	0.3%	-

In terms of percentiles by age and sex from the growth charts, the meanings of the categories are:

- Underweight (BMI < 5th percentile)
- Normal weight (BMI 5th to < 85th percentile)
- Overweight (BMI 85th to < 95th percentile)

- Obese ($\text{BMI} \geq 95\text{th percentile}$)

Note how I've used labels in the `bmi_cat` variable that include a number at the start so that the table results are sorted in a rational way. R sorts tables alphabetically, in general. We'll use the `forcats` package to work with categorical variables that we store as *factors* eventually, but for now, we'll keep things relatively simple.

Note that the `bmi_cat` data don't completely separate out the raw `bmi` data, because the calculation of percentiles requires different tables for each combination of `age` and `sex`.

```
mosaic::favstats(bmi ~ bmi_cat, data = nnyfs)
```

	bmi_cat	min	Q1	median	Q3	max	mean	sd	n	missing
1	1_Underweight	11.9	13.4	13.7	15.0	16.5	14.1	1.10	41	0
2	2_Normal	13.5	15.4	16.5	18.7	24.0	17.2	2.30	920	0
3	3_Overweight	16.9	18.3	21.4	23.4	27.9	21.2	2.92	258	0
4	4_Obese	17.9	22.3	26.2	30.2	48.3	26.7	5.72	295	0

7.2.8 waist

Let's also look briefly at `waist`, which is the circumference of the child's waist, in centimeters. Again, this is a numeric variable, so perhaps we'll stick to the simple summary, rather than obtaining a table of observed values.

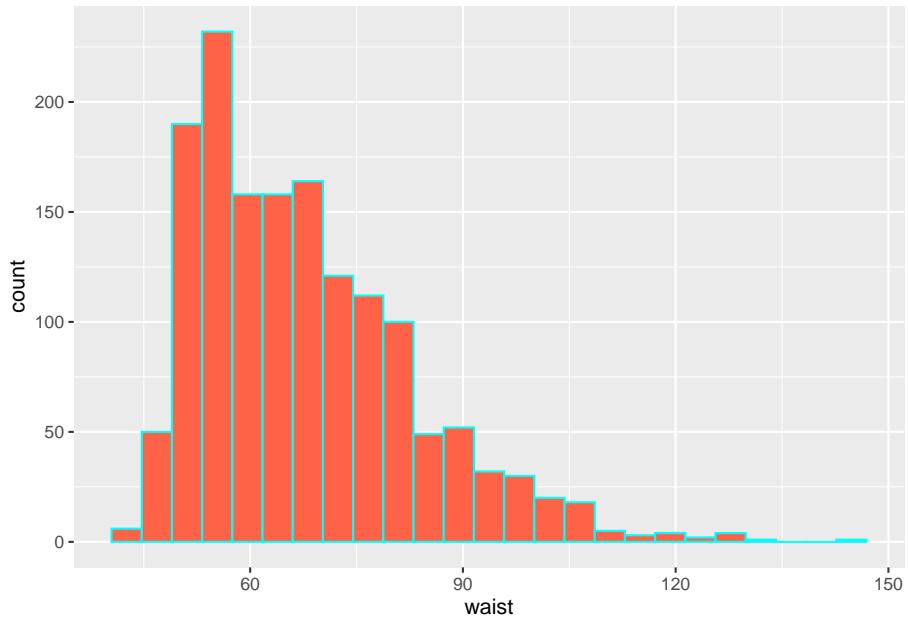
```
mosaic::favstats(~ waist, data = nnyfs)
```

```
min   Q1 median   Q3 max mean   sd    n missing
42.5 55.6   64.8 76.6 145 67.7 15.2 1512      6
```

Here's a histogram of the waist circumference data.

```
ggplot(nnyfs, aes(x = waist)) +
  geom_histogram(bins = 25, fill = "tomato", color = "cyan")
```

Warning: Removed 6 rows containing non-finite values (stat_bin).



7.2.9 triceps_skinfold

The last variable I'll look at for now is `triceps_skinfold`, which is measured in millimeters. This is one of several common locations used for the assessment of body fat using skinfold calipers, and is a frequent part of growth assessments in children. Again, this is a numeric variable according to R.

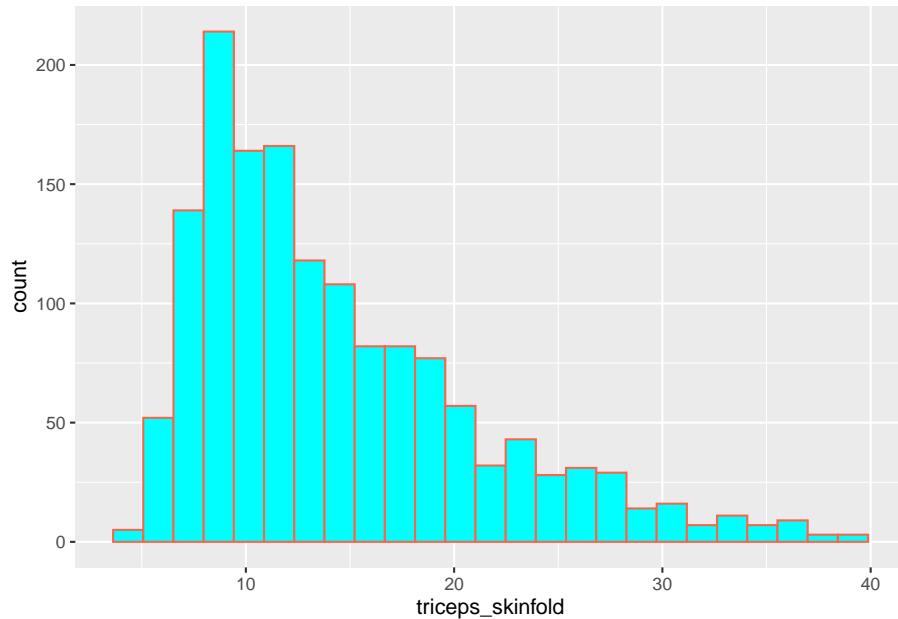
```
mosaic::favstats(~ triceps_skinfold, data = nnyfs)
```

```
min   Q1 median Q3  max mean    sd      n missing
4 9.1   12.4 18 38.8 14.4 6.76 1497      21
```

And here's a histogram of the triceps skinfold data, with the fill and color flipped from what we saw in the plot of the waist circumference data a moment ago.

```
ggplot(nnyfs, aes(x = triceps_skinfold)) +
  geom_histogram(bins = 25, fill = "cyan", color = "tomato")
```

Warning: Removed 21 rows containing non-finite values (stat_bin).



OK. We've seen a few variables, and we'll move on now to look more seriously at the data.

7.3 Basic Numerical Summaries

7.3.1 The Five Number Summary, Quantiles and IQR

The **five number summary** is most famous when used to form a box plot - it's the minimum, 25th percentile, median, 75th percentile and maximum. For numerical and integer variables, the `summary` function produces the five number summary, plus the mean, and a count of any missing values (NA's).

```
nyfs %>%
  select(waist, energy, sugar) %>%
  summary()
```

	waist	energy	sugar
Min.	: 42.5	Min. : 257	Min. : 1
1st Qu.	: 55.6	1st Qu.:1368	1st Qu.: 83
Median	: 64.8	Median :1794	Median :117
Mean	: 67.7	Mean :1877	Mean :124
3rd Qu.	: 76.6	3rd Qu.:2306	3rd Qu.:157
Max.	:144.7	Max. :5265	Max. :405
NA's	:6		

114 CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

As an alternative, we can use the \$ notation to indicate the variable we wish to study inside a data set, and we can use the `fivenum` function to get the five numbers used in developing a box plot. We'll focus for a little while on the number of kilocalories consumed by each child, according to the dietary recall questionnaire. That's the `energy` variable.

```
fivenum(nnyfs$energy)
```

```
[1] 257 1367 1794 2306 5265
```

- As mentioned in 5.3.1, the **inter-quartile range**, or IQR, is sometimes used as a competitor for the standard deviation. It's the difference between the 75th percentile and the 25th percentile. The 25th percentile, median, and 75th percentile are referred to as the quartiles of the data set, because, together, they split the data into quarters.

```
IQR(nnyfs$energy)
```

```
[1] 938
```

We can obtain **quantiles** (percentiles) as we like - here, I'm asking for the 1st and 99th:

```
quantile(nnyfs$energy, probs=c(0.01, 0.99))
```

```
1%   99%
567 4052
```

7.4 Additional Summaries from `favstats`

If we're focusing on a single variable, the `favstats` function in the `mosaic` package can be very helpful. Rather than calling up the entire `mosaic` library here, I'll just specify the function within the library.

```
mosaic::favstats(~ energy, data = nnyfs)
```

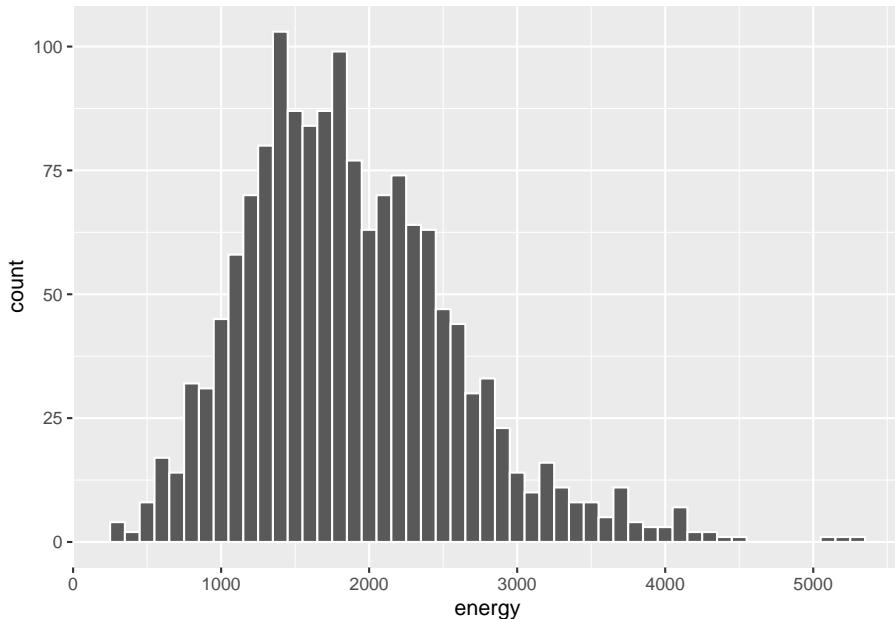
```
min   Q1 median   Q3 max mean   sd      n missing
257   1368    1794  2306 5265 1877  722 1518       0
```

This adds three useful results to the base summary - the standard deviation, the sample size and the number of missing observations.

7.5 The Histogram

As we saw in 3, obtaining a basic **histogram** of, for example, the `energy` (kilocalories consumed) in the `nnyfs` data is pretty straightforward.

```
ggplot(data = nnyfs, aes(x = energy)) +
  geom_histogram(binwidth = 100, col = "white")
```



7.5.1 Freedman-Diaconis Rule to select bin width

If we like, we can suggest a particular number of cells for the histogram, instead of accepting the defaults. In this case, we have $n = 1518$ observations. The **Freedman-Diaconis rule** can be helpful here. That rule suggests that we set the bin-width to

$$h = \frac{2 * IQR}{n^{1/3}}$$

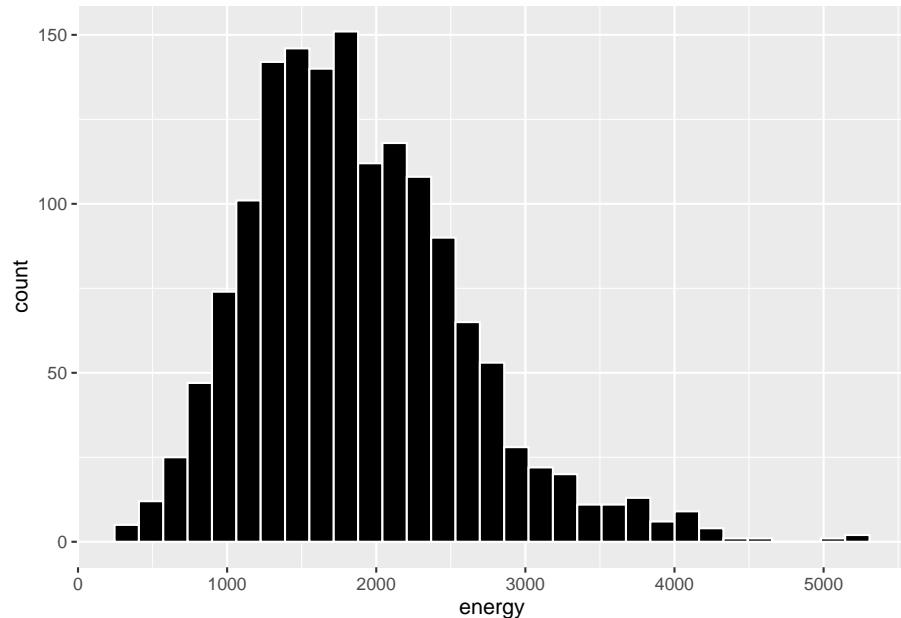
so that the number of bins is equal to the range of the data set (maximum - minimum) divided by h .

For the `energy` data in the `nnyfs` tibble, we have

- IQR of 938.5, $n = 1518$ and range = 5008
- Thus, by the Freedman-Diaconis rule, the optimal binwidth h is 163.32, or, realistically, 163.
- And so the number of bins would be 30.664, or, realistically 31.

Here, we'll draw the graph again, using the Freedman-Diaconis rule to identify the number of bins, and also play around a bit with the fill and color of the bars.

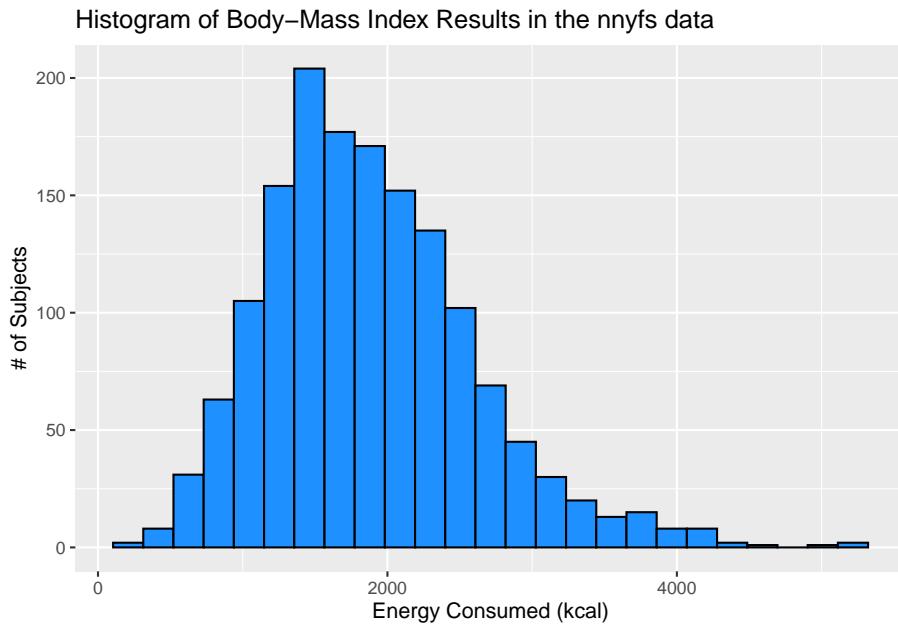
```
bw <- 2 * IQR(nnyfs$energy) / length(nnyfs$energy)^(1/3)
ggplot(data = nnyfs, aes(x = energy)) +
  geom_histogram(binwidth=bw, color = "white", fill = "black")
```



This is a nice start, but it is by no means a finished graph.

Let's improve the axis labels, add a title, and fill in the bars with a distinctive blue and use a black outline around each bar. I'll just use 25 bars, because I like how that looks in this case, and optimizing the number of bins is rarely important.

```
ggplot(data = nnyfs, aes(x = energy)) +
  geom_histogram(bins=25, color = "black", fill = "dodgerblue") +
  labs(title = "Histogram of Body-Mass Index Results in the nnyfs data",
       x = "Energy Consumed (kcal)", y = "# of Subjects")
```



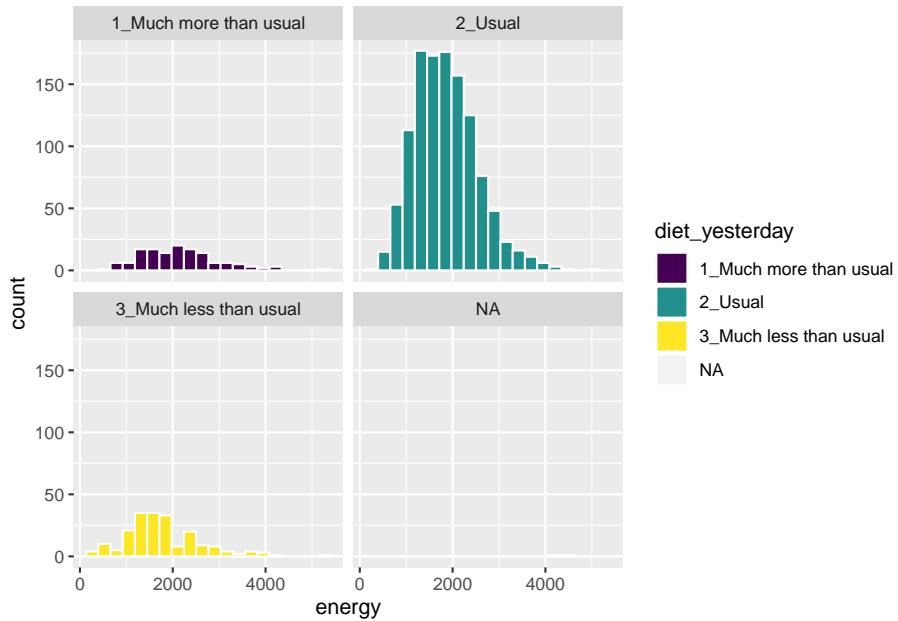
7.5.2 A Note on Colors

The simplest way to specify a color is with its name, enclosed in parentheses. My favorite list of R colors is <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>. In a pinch, you can usually find it by googling **Colors in R**. You can also type `colors()` in the R console to obtain a list of the names of the same 657 colors.

When using colors to make comparisons, you may be interested in using a scale that has some nice properties. The `viridis` package vignette describes four color scales (`viridis`, `magma`, `plasma` and `inferno`) that are designed to be colorful, robust to colorblindness and gray scale printing, and perceptually uniform, which means (as the package authors describe it) that values close to each other have similar-appearing colors and values far away from each other have more different-appearing colors, consistently across the range of values. We can apply these colors with special functions within `ggplot`.

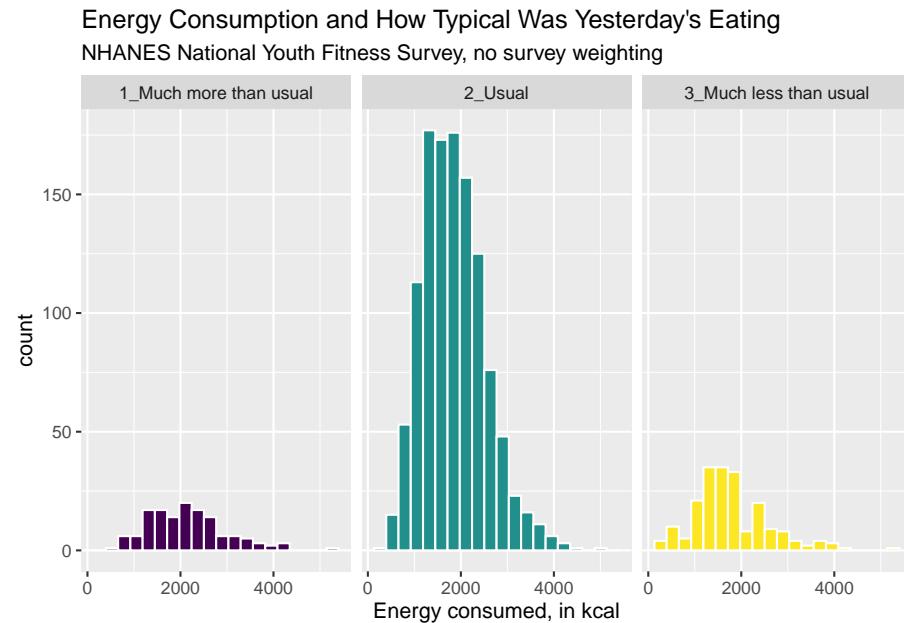
Here's a comparison of several histograms, looking at `energy` consumed as a function of whether yesterday was typical in terms of food consumption.

```
ggplot(data = nnyfs, aes(x = energy, fill = diet_yesterday)) +
  geom_histogram(bins = 20, col = "white") +
  scale_fill_viridis_d() +
  facet_wrap(~ diet_yesterday)
```



We don't really need the legend here, and perhaps we should restrict the plot to participants who responded to the `diet_yesterday` question, and put in a title and better axis labels?

```
nnyfs %>% filter(complete.cases(energy, diet_yesterday)) %>%
  ggplot(data = ., aes(x = energy, fill = diet_yesterday)) +
  geom_histogram(bins = 20, col = "white") +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  facet_wrap(~ diet_yesterday) +
  labs(x = "Energy consumed, in kcal",
       title = "Energy Consumption and How Typical Was Yesterday's Eating",
       subtitle = "NHANES National Youth Fitness Survey, no survey weighting")
```



7.6 The Stem-and-Leaf

We might consider a **stem-and-leaf display** (a John Tukey invention) to show the actual data values while retaining the shape of a histogram. The `scale` parameter can help expand the size of the diagram, so you can see more of the values. Stem and leaf displays are usually used for relatively small samples, perhaps with 10-200 observations, so we'll first take a sample of 150 of the BMI values from the complete set gathered in the `nnyfs` tibble.

```
set.seed(431) # set a seed for the random sampling so we can replicate the results

sampleA <- sample_n(nnyfs, 150, replace = FALSE) # draw a sample of 150 unique rows from nnyfs

stem(sampleA$bmi) # build a stem-and-leaf for those 150 sampled BMI values
```

The decimal point is at the |

```
12 | 936678999
14 | 012457880011233555566777789999
16 | 001223333335556688890001222355888889
18 | 03455778991112556889
20 | 02344678912378
22 | 00013455779125
```

120CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

```
24 | 346978
26 | 03799446
28 | 25
30 | 2
32 | 065
34 | 14
36 | 31
38 |
40 |
42 |
44 |
46 |
48 | 3
```

We can see that the minimum BMI value in this small sample is 12.9 and the maximum BMI value is 48.3.

```
summary(sampleA$bmi)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
12.9	15.9	17.8	19.7	22.0	48.3	1

If we really wanted to, we could obtain a stem-and-leaf of all of the BMI values in the entire `nnyfs` data. The `scale` parameter lets us see some more of the values.

```
stem(nnyfs$bmi, scale = 2)
```

The decimal point is at the |

```
11 | 9
12 | 679
13 | 0012344455556666677788888999999999999
14 | 00000001111111111222222233333333344444444555555555666666667+48
15 | 000000000000000111111111122222222222222333333333333333333333+134
16 | 0000000000000000000000000111111111112222222222223333333333333333+114
17 | 0000000000000000000000000111111111112222222222222333333333333333+75
18 | 00000000000000011111111111222222222233333333444444455555555555+39
19 | 00000111111111111222222222223333333333334444444555555555555+36
20 | 000000000000111111111222222233333333334444445556666666667777778+3
21 | 00000001111112222223333333444444445555555566666667777778888+7
22 | 000000000000111111122222233333334444444555556666666777778899999
23 | 0000000111122222223333344444445556677888889999
24 | 0000001122222333344445555566666677888889999
25 | 0001112222233344444555666666777788889999
26 | 00011222223345555667789999
27 | 002233444455566679999
```

```

28 | 11122344456667778999
29 | 0112222333567788
30 | 1122222344556788999
31 | 0013445567788
32 | 0023344669
33 | 1255
34 | 01224456679
35 | 0179
36 | 369
37 | 01458
38 | 138
39 |
40 | 0
41 | 6
42 | 8
43 | 0
44 |
45 |
46 | 9
47 |
48 | 23

```

Note that some of the rows extend far beyond what is displayed in the data (as indicated by the + sign, followed by a count of the number of unshown data values.)

7.6.1 A Fancier Stem-and-Leaf Display

We can use the `stem.leaf` function in the `aplypack` package to obtain a fancier version of the stem-and-leaf plot, that identifies outlying values. Below, we display this new version for the random sample of 150 BMI observations we developed earlier.

```

aplypack::stem.leaf(sampleA$bmi)

1 | 2: represents 1.2
leaf unit: 0.1
n: 149
  1   12 | 9
  9   13 | 36678999
 17   14 | 01245788
 39   15 | 0011233555566777789999
 60   16 | 00122333333555668889
(16)  17 | 0001222355888889
 73   18 | 0345577899
 63   19 | 1112556889

```

122 CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

```

53    20 | 023446789
44    21 | 12378
39    22 | 00013455779
28    23 | 125
25    24 | 3469
21    25 | 78
19    26 | 03799
14    27 | 446
     28 |
11    29 | 25
     30 | 2
HI: 32 32.6 33.5 34.1 34.4 36.3 37.1 48.3
NA's: 1

```

We can also produce back-to-back stem and leaf plots to compare, for instance, body-mass index by sex.

```

samp.F <- filter(sampleA, sex=="Female")
samp.M <- filter(sampleA, sex=="Male")

aplpack::stem.leaf.backback(samp.F$bmi, samp.M$bmi)

```

```

-----  

1 | 2: represents 1.2, leaf unit: 0.1  

      samp.F$bmi      samp.M$bmi  

-----  

1          9| 12 |  

5          9863| 13 |6799           4  

7          10| 14 |245788         10  

19         999875532100| 15 |1355667779       20  

25          863330| 16 |012233335556889   35  

29          8210| 17 |002235588889 (12)  

(4)          9750| 18 |345789        38  

31          985211| 19 |1568        32  

25          943| 20 |024678        28  

22          8731| 21 |2          22  

18          000| 22 |13455779       21  

15          51| 23 |2          13  

13          63| 24 |49         12  

11          87| 25 |  

9           0| 26 |3799        10  

8           644| 27 |  

           | 28 |  

5           2| 29 |5          6  

4           2| 30 |  

           | 31 |
-----
```

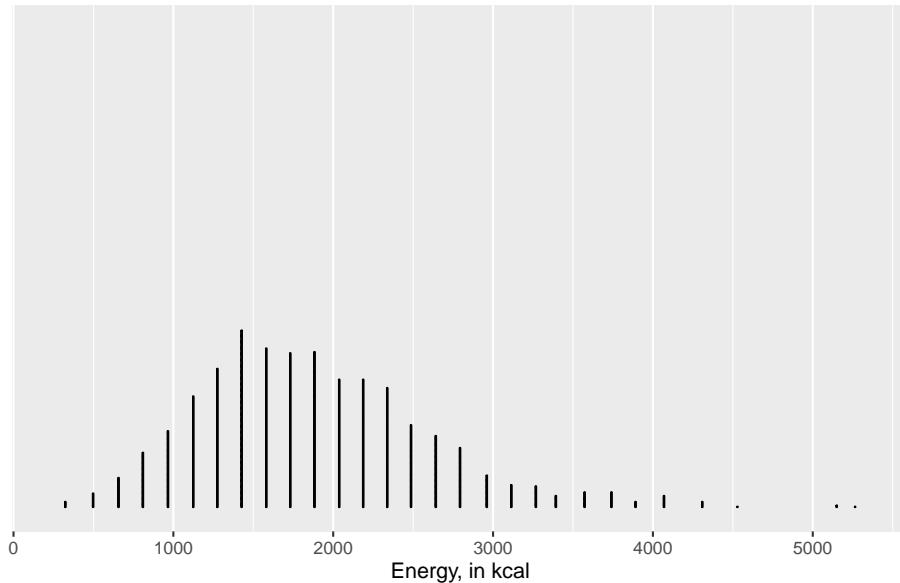
```
HI: 33.5 37.1 48.3      HI: 32 32.6 34.1 34.4
36.3
n:          65      85
NAs:        1       0
-----
```

7.7 The Dot Plot to display a distribution

We can plot the distribution of a single continuous variable using the `dotplot` geom:

```
ggplot(data = nnyfs, aes(x = energy)) +
  geom_dotplot(dotsize = 0.05, binwidth=150) +
  scale_y_continuous(NULL, breaks = NULL) # hides y-axis since it is meaningless
  labs(title = "Dotplot of nnyfs Kilocalories consumed",
       x = "Energy, in kcal")
```

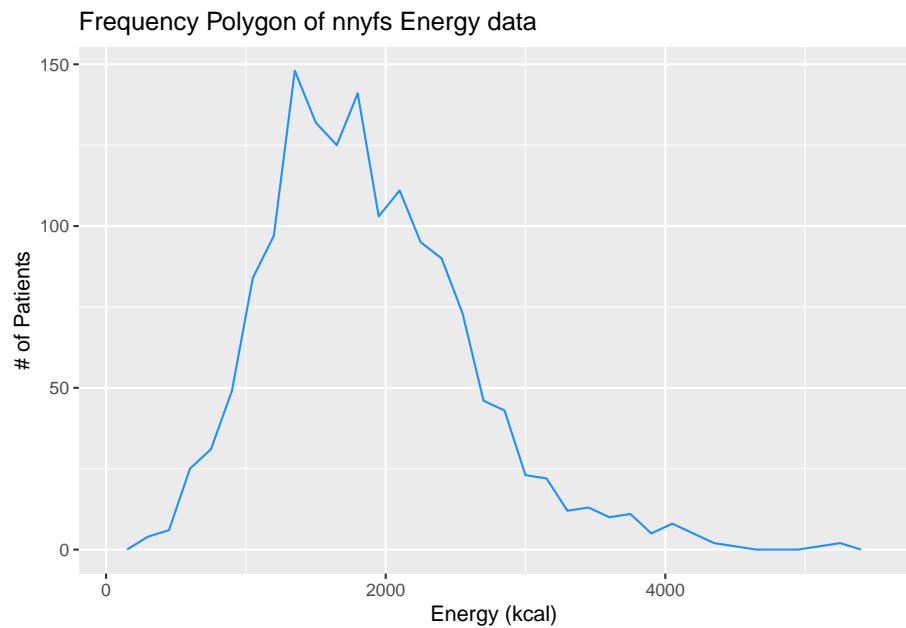
Dotplot of nnyfs Kilocalories consumed



7.8 The Frequency Polygon

We can plot the distribution of a single continuous variable using the `freqpoly` geom:

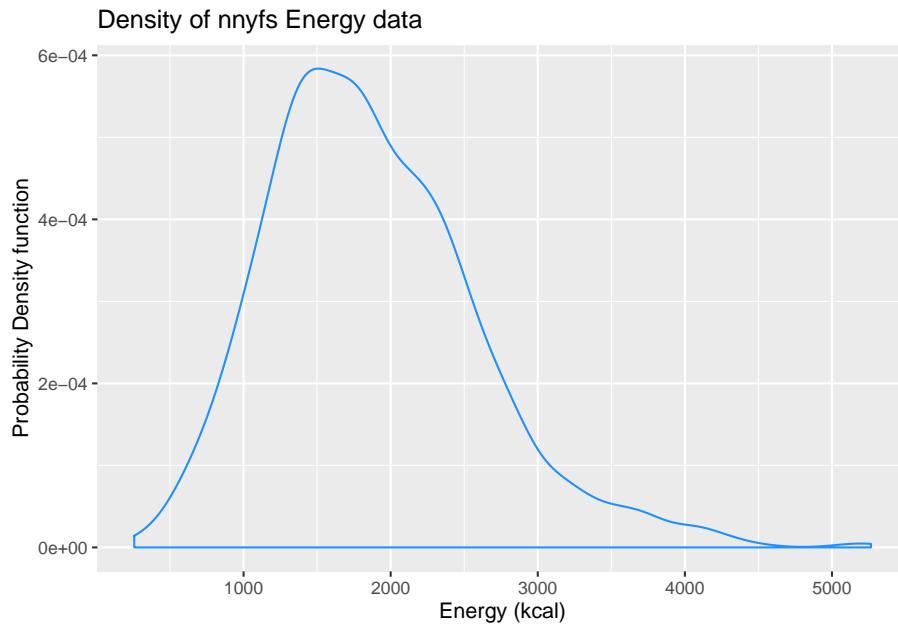
```
ggplot(data = nnyfs, aes(x = energy)) +
  geom_freqpoly(binwidth = 150, color = "dodgerblue") +
  labs(title = "Frequency Polygon of nnyfs Energy data",
       x = "Energy (kcal)", y = "# of Patients")
```



7.9 Plotting the Probability Density Function

We can also produce a density function, which has the effect of smoothing out the bumps in a histogram or frequency polygon, while also changing what is plotted on the y-axis.

```
ggplot(data = nnyfs, aes(x = energy)) +
  geom_density(kernel = "gaussian", color = "dodgerblue") +
  labs(title = "Density of nnyfs Energy data",
       x = "Energy (kcal)", y = "Probability Density function")
```



So, what's a density function?

- A probability density function is a function of a continuous variable, x , that represents the probability of x falling within a given range. Specifically, the integral over the interval (a,b) of the density function gives the probability that the value of x is within (a,b) .
- If you're interested in exploring more on the notion of density functions for continuous (and discrete) random variables, some nice elementary material is available at Khan Academy.

7.10 The Boxplot

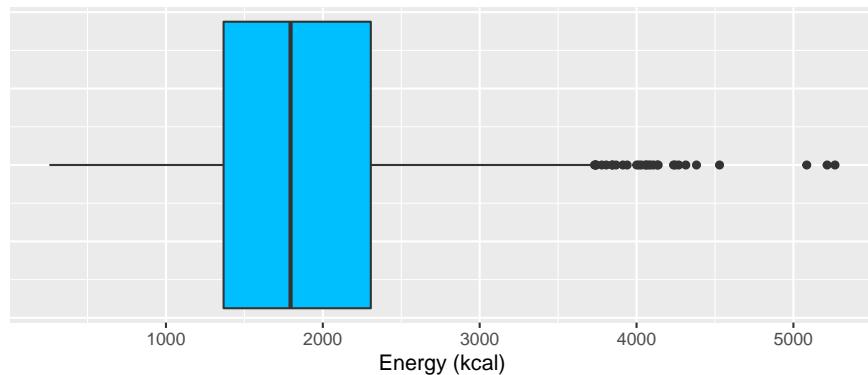
Sometimes, it's helpful to picture the five-number summary of the data in such a way as to get a general sense of the distribution. One approach is a **boxplot**, sometimes called a box-and-whisker plot.

7.10.1 Drawing a Boxplot for One Variable in `ggplot2`

The `ggplot2` library easily handles comparison boxplots for multiple distributions, as we'll see in a moment. However, building a boxplot for a single distribution requires a little trickiness.

```
ggplot(nnyfs, aes(x = 1, y = energy)) +
  geom_boxplot(fill = "deepskyblue") +
  coord_flip() +
  labs(title = "Boxplot of Energy for kids in the NNYFS",
       y = "Energy (kcal)",
       x = "") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
```

Boxplot of Energy for kids in the NNYFS



7.10.2 About the Boxplot

The boxplot is another John Tukey invention.

- R draws the box (here in yellow) so that its edges of the box fall at the 25th and 75th percentiles of the data, and the thick line inside the box falls at the median (50th percentile).
- The whiskers then extend out to the largest and smallest values that are not classified by the plot as candidate *outliers*.
- An outlier is an unusual point, far from the center of a distribution.
- Note that I've used the `horizontal` option to show this boxplot in this direction. Most comparison boxplots, as we'll see below, are oriented vertically.

The boxplot's **whiskers** that are drawn from the first and third quartiles (i.e. the 25th and 75th percentiles) out to the most extreme points in the data that do not meet the standard of “candidate outliers.” An outlier is simply a point that is far away from the center of the data - which may be due to any number of reasons, and generally indicates a need for further investigation.

Most software, including R, uses a standard proposed by Tukey which describes a “candidate outlier” as any point above the **upper fence** or below the **lower fence**. The definitions of the fences are based on the inter-quartile range (IQR).

If $IQR = 75\text{th percentile} - 25\text{th percentile}$, then the upper fence is $75\text{th percentile} + 1.5 \cdot IQR$, and the lower fence is $25\text{th percentile} - 1.5 \cdot IQR$.

So for these `energy` data,

- the upper fence is located at $2306 + 1.5(938.5) = 3713.75$
- the lower fence is located at $1367 - 1.5(938.5) = -40.75$

In this case, we see no points identified as outliers in the low part of the distribution, but quite a few identified that way on the high side. This tends to identify about 5% of the data as a candidate outlier, *if* the data follow a Normal distribution.

- This plot is indicating clearly that there is some asymmetry (skew) in the data, specifically right skew.
- The standard R uses is to indicate as outliers any points that are more than 1.5 inter-quartile ranges away from the edges of the box.

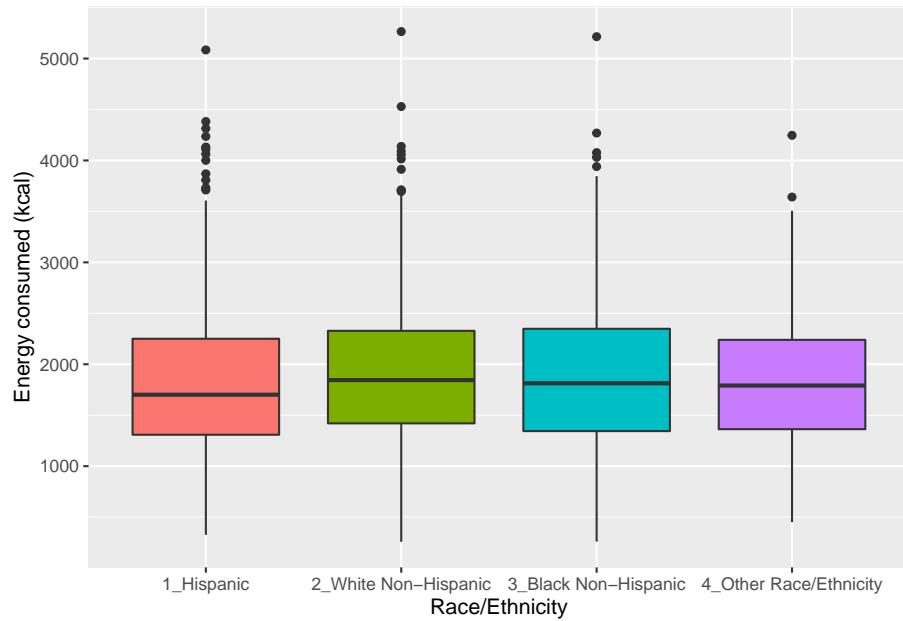
The horizontal orientation I've chosen here clarifies the relationship of direction of skew to the plot. A plot like this, with multiple outliers on the right side is indicative of a long right tail in the distribution, and hence, positive or right skew - with the mean being larger than the median. Other indications of skew include having one side of the box being substantially wider than the other, or one side of the whiskers being substantially longer than the other. More on skew later.

7.11 A Simple Comparison Boxplot

Boxplots are most often used for comparison. We can build boxplots using `ggplot2`, as well, and we'll discuss that in detail later. For now, here's a boxplot built to compare the `energy` results by the subject's race/ethnicity.

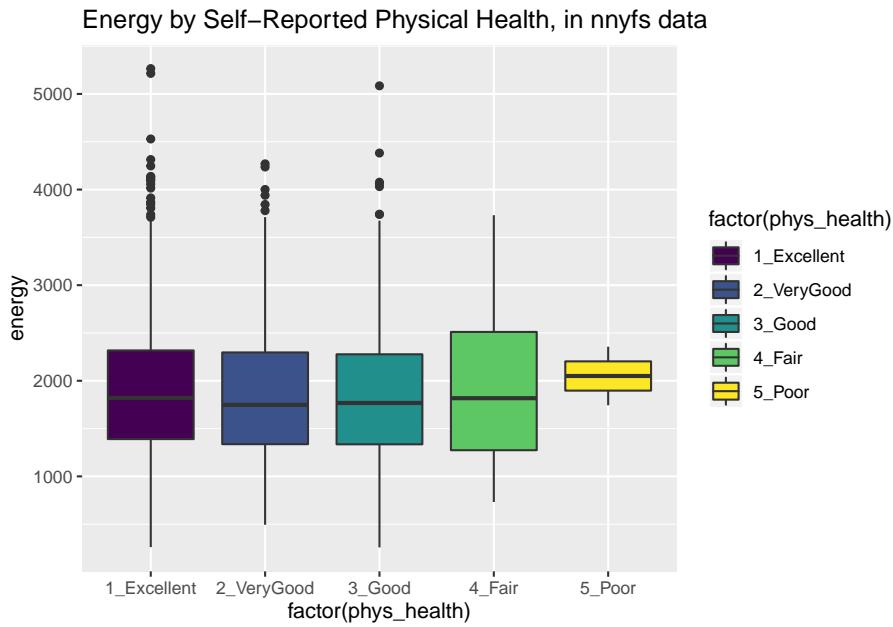
```
ggplot(nnyfs, aes(x = factor(race_eth), y = energy, fill=factor(race_eth))) +
  geom_boxplot() +
  guides(fill = FALSE) +
  labs(y = "Energy consumed (kcal)", x = "Race/Ethnicity")
```

128CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)



Let's look at the comparison of observed energy levels across the five categories in our `phys_health` variable, now making use of the `viridis` color scheme.

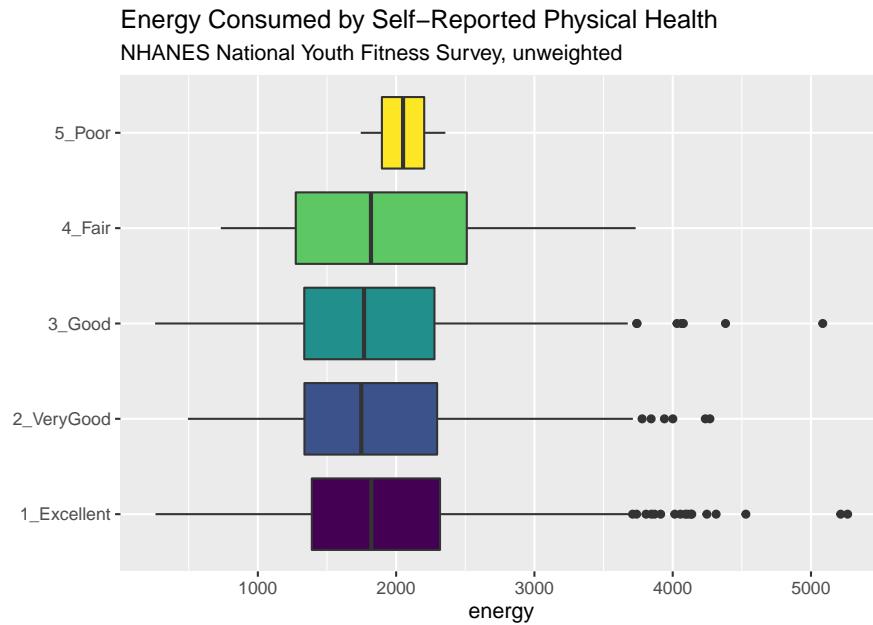
```
ggplot(nnyfs, aes(x = factor(phys_health), y = energy, fill = factor(phys_health))) +  
  geom_boxplot() +  
  scale_fill_viridis_d() +  
  labs(title = "Energy by Self-Reported Physical Health, in nnyfs data")
```



As a graph, that's not bad, but what if we want to improve it further?

Let's turn the boxes in the horizontal direction, and get rid of the perhaps unnecessary `phys_health` labels.

```
ggplot(nnyfs, aes(x = factor(phys_health), y = energy, fill = factor(phys_health))) +
  geom_boxplot() +
  scale_fill_viridis_d() +
  coord_flip() +
  guides(fill=FALSE) +
  labs(title = "Energy Consumed by Self-Reported Physical Health",
       subtitle = "NHANES National Youth Fitness Survey, unweighted",
       x = "")
```



7.12 Using `describe` in the `psych` library

For additional numerical summaries, one option would be to consider using the `describe` function from the `psych` library.

```
psych::describe(nnyfs$energy)
```

```
vars     n  mean   sd median trimmed mad min  max range skew kurtosis    se
X1      1 1518 1877  722    1794    1827  678 257 5265  5008  0.8     1.13 18.5
```

This package provides, in order, the following...

- `n` = the sample size
- `mean` = the sample mean
- `sd` = the sample standard deviation
- `median` = the median, or 50th percentile
- `trimmed` = mean of the middle 80% of the data
- `mad` = median absolute deviation
- `min` = minimum value in the sample
- `max` = maximum value in the sample
- `range` = max - min
- `skew` = skewness measure, described below (indicates degree of asymmetry)
- `kurtosis` = kurtosis measure, described below (indicates heaviness of tails, degree of outlier-proneness)

- **se** = standard error of the sample mean = $sd / \sqrt{\text{sample size}}$, useful in inference

7.12.1 The Trimmed Mean

The **trimmed mean** trim value in R indicates proportion of observations to be trimmed from each end of the outcome distribution before the mean is calculated. The trim value provided by the `psych::describe` package describes what this particular package calls a 20% trimmed mean (bottom and top 10% of `energy` values are removed before taking the mean - it's the mean of the middle 80% of the data.) I might call that a 10% trimmed mean in some settings, but that's just me.

```
mean(nnyfs$energy, trim=.1)
```

```
[1] 1827
```

7.12.2 The Median Absolute Deviation

An alternative to the IQR that is fancier, and a bit more robust, is the **median absolute deviation**, which, in large sample sizes, for data that follow a Normal distribution, will be (in expectation) equal to the standard deviation. The MAD is the median of the absolute deviations from the median, multiplied by a constant (1.4826) to yield asymptotically normal consistency.

```
mad(nnyfs$energy)
```

```
[1] 678
```

7.13 Assessing Skew

A relatively common idea is to assess **skewness**, several measures of which (including the one below, sometimes called type 3 skewness, or Pearson's moment coefficient of skewness) are available. Many models assume a Normal distribution, where, among other things, the data are symmetric around the mean.

Skewness measures asymmetry in the distribution - left skew ($\text{mean} < \text{median}$) is indicated by negative skewness values, while right skew ($\text{mean} > \text{median}$) is indicated by positive values. The skew value will be near zero for data that follow a Normal distribution.

7.13.1 Non-parametric Skew via skew_1

A simpler measure of skew, sometimes called the **nonparametric skew** and closely related to Pearson's notion of median skewness, falls between -1 and +1 for any distribution. It is just the difference between the mean and the median, divided by the standard deviation.

- Values greater than +0.2 are sometimes taken to indicate fairly substantial right skew, while values below -0.2 indicate fairly substantial left skew.

```
(mean(nnyfs$energy) - median(nnyfs$energy))/sd(nnyfs$energy)
```

```
[1] 0.114
```

The Wikipedia page on skewness, from which some of this material is derived, provides definitions for several other skewness measures.

7.14 Assessing Kurtosis (Heavy-Tailedness)

Another measure of a distribution's shape that can be found in the `psych` library is the **kurtosis**. Kurtosis is an indicator of whether the distribution is heavy-tailed or light-tailed as compared to a Normal distribution. Positive kurtosis means more of the variance is due to outliers - unusual points far away from the mean relative to what we might expect from a Normally distributed data set with the same standard deviation.

- A Normal distribution will have a kurtosis value near 0, a distribution with similar tail behavior to what we would expect from a Normal is said to be *mesokurtic*
- Higher kurtosis values (meaningfully higher than 0) indicate that, as compared to a Normal distribution, the observed variance is more the result of extreme outliers (i.e. heavy tails) as opposed to being the result of more modest sized deviations from the mean. These heavy-tailed, or outlier prone, distributions are sometimes called *leptokurtic*.
- Kurtosis values meaningfully lower than 0 indicate light-tailed data, with fewer outliers than we'd expect in a Normal distribution. Such distributions are sometimes referred to as *platykurtic*, and include distributions without outliers, like the Uniform distribution.

Here's a table:

Fewer outliers than a Normal	Approximately Normal	More outliers than a Normal
Light-tailed <i>platykurtic</i> (kurtosis < 0)	“Normalish” <i>mesokurtic</i> (kurtosis = 0)	Heavy-tailed <i>leptokurtic</i> (kurtosis > 0)

```
psych::kurtosi(nnyfs$energy)
```

```
[1] 1.13
```

7.14.1 The Standard Error of the Sample Mean

The **standard error** of the sample mean, which is the standard deviation divided by the square root of the sample size:

```
sd(nnyfs$energy)/sqrt(length(nnyfs$energy))
```

```
[1] 18.5
```

7.15 The describe function in the Hmisc library

The **Hmisc** library has lots of useful functions. It's named for its main developer, Frank Harrell. The **describe** function in **Hmisc** knows enough to separate numerical from categorical variables, and give you separate (and detailed) summaries for each.

- For a categorical variable, it provides counts of total observations (n), the number of missing values, and the number of unique categories, along with counts and percentages falling in each category.
- For a numerical variable, it provides:
 - counts of total observations (n), the number of missing values, and the number of unique values
 - an Info value for the data, which indicates how continuous the variable is (a score of 1 is generally indicative of a completely continuous variable with no ties, while scores near 0 indicate lots of ties, and very few unique values)
 - the sample Mean
 - many sample percentiles (quantiles) of the data, specifically (5, 10, 25, 50, 75, 90, 95, 99)
 - either a complete table of all observed values, with counts and percentages (if there are a modest number of unique values), or
 - a table of the five smallest and five largest values in the data set, which is useful for range checking

```
nnyfs %>%
  select(waist, energy, bmi) %>%
  Hmisc::describe()
```

```
-----
waist
    n   missing  distinct      Info     Mean      Gmd      .05      .10
    1512       6      510        1    67.71    16.6    49.40    51.40
    .25       .50      .75        .90      .95
    55.60    64.80    76.60      88.70    96.84

lowest :  42.5  43.4  44.1  44.4  44.5, highest: 125.8 126.0 127.0 132.3 144.7
-----
energy
    n   missing  distinct      Info     Mean      Gmd      .05      .10
    1518       0     1137        1    1877    796.1    849    1047
    .25       .50      .75        .90      .95
    1368    1794    2306      2795    3195

lowest :  257  260  326  349  392, highest: 4382 4529 5085 5215 5265
-----
bmi
    n   missing  distinct      Info     Mean      Gmd      .05      .10
    1514       4      225        1    19.63    5.269    14.30    14.90
    .25       .50      .75        .90      .95
    15.90   18.10   21.90      26.27    30.20

lowest : 11.9 12.6 12.7 12.9 13.0, highest: 42.8 43.0 46.9 48.2 48.3
-----
```

More on the `Info` value in `Hmisc::describe` is available here

7.16 xda from GitHub for numerical summaries for exploratory data analysis

```
## next two commands needed if xda is not already installed
```

```
library(devtools)
install_github("ujjwalkarn/xdar")
```

```
Skipping install of 'xda' from a github remote, the SHA1 (86cf14db) has not changed since previous
  Use `force = TRUE` to force installation
```

```
xda::numSummary(nnyfs)
```

	n	mean	sd	max	min	range
SEQN	1518	7.27e+04	4.57e+02	73492.0	71917.00	1575.0
age_child	1518	9.03e+00	3.71e+00	15.0	3.00	12.0
educ_child	1181	4.26e+00	2.82e+00	9.0	0.00	9.0

	1518	3.41e+04	1.60e+04	104673.9	9412.87	95261.0
seqn	1518	2.24e+00	1.61e+00	5.0	0.00	5.0
age_adult	1518	4.00e+01	9.60e+00	80.0	18.00	62.0
energy	1518	1.88e+03	7.22e+02	5265.0	257.00	5008.0
protein	1518	6.69e+01	3.10e+01	241.8	4.18	237.7
sugar	1518	1.24e+02	5.90e+01	405.5	1.00	404.5
fat	1518	6.74e+01	3.37e+01	235.2	1.70	233.5
water	1518	5.43e+02	6.50e+02	8591.2	0.00	8591.2
plank_time	1384	6.11e+01	4.58e+01	450.0	1.00	449.0
height	1514	1.37e+02	2.30e+01	188.9	89.70	99.2
weight	1514	3.98e+01	2.08e+01	136.9	12.30	124.6
bmi	1514	1.96e+01	5.08e+00	48.3	11.90	36.4
arm_length	1511	2.90e+01	5.63e+00	42.5	17.30	25.2
waist	1512	6.77e+01	1.52e+01	144.7	42.50	102.2
arm_circ	1513	2.29e+01	5.50e+00	46.8	13.70	33.1
calf_circ	1509	2.98e+01	6.19e+00	55.9	18.30	37.6
calf_skinfold	1390	1.37e+01	6.81e+00	39.9	3.70	36.2
triceps_skinfold	1497	1.44e+01	6.76e+00	38.8	4.00	34.8
subscapular_skinfold	1451	1.07e+01	6.31e+00	38.0	3.70	34.3
active_days	1513	5.39e+00	2.12e+00	7.0	0.00	7.0
tv_hours	1514	1.90e+00	1.37e+00	5.0	0.00	5.0
computer_hours	1515	1.04e+00	1.35e+00	5.0	0.00	5.0
meals_out	1510	1.79e+00	2.13e+00	20.0	0.00	20.0
med_count	1518	2.96e-01	7.56e-01	6.0	0.00	6.0
	nunique	nzeros	iqr	lowerbound	upperbound	
SEQN	1518	0	792.00	71121.25	74288.75	
age_child	13	0	6.00	-3.00	21.00	
educ_child	11	146	5.00	-5.50	14.50	
sampling_wt	845	0	19987.74	-6664.16	73223.34	
income_pov	238	15	2.65	-3.10	7.49	
age_adult	55	0	12.00	15.00	63.00	
energy	1137	0	939.00	-41.00	3714.50	
protein	1482	0	37.25	-10.55	138.44	
sugar	1491	0	74.35	-28.86	268.57	
fat	1474	0	42.48	-20.57	149.34	
water	342	326	636.94	-851.73	1696.05	
plank_time	190	0	55.00	-55.50	164.75	
height	694	0	38.45	60.67	214.45	
weight	615	0	30.55	-23.30	98.90	
bmi	226	0	6.00	6.90	30.90	
arm_length	222	0	9.60	9.80	48.20	
waist	511	0	21.00	24.10	108.10	
arm_circ	240	0	7.80	6.70	37.90	
calf_circ	259	0	9.57	10.34	48.66	
calf_skinfold	253	0	8.40	-4.00	29.60	
triceps_skinfold	263	0	8.90	-4.25	31.35	

136 CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

subscapular_skinfold	230	0	7.00	-4.30	23.70		
active_days	9	68	3.00	-0.50	11.50		
tv_hours	7	254	2.00	-2.00	6.00		
computer_hours	7	759	2.00	-3.00	5.00		
meals_out	17	425	2.00	-3.00	5.00		
med_count	7	1241	0.00	0.00	0.00		
	noutlier	kurtosis	skewness	mode	miss	miss%	
SEQN	0	-1.2050	0.000814	71917.0	0	0.000	
age_child	0	-1.2317	-0.025225	12.0	0	0.000	
educ_child	0	-1.2195	0.010719	NA	337	22.200	
sampling_wt	37	1.9218	1.123744	53977.6	0	0.000	
income_pov	0	-1.1165	0.501565	5.0	89	5.863	
age_adult	20	0.8409	0.631691	40.0	0	0.000	
energy	30	1.1305	0.797210	1915.0	0	0.000	
protein	45	2.0114	1.121048	82.5	0	0.000	
sugar	30	1.4265	0.921237	94.7	0	0.000	
fat	43	1.6609	1.024582	76.7	0	0.000	
water	79	24.9830	3.486903	0.0	0	0.000	
plank_time	40	5.3557	1.538689	NA	134	8.827	
height	0	-1.0648	-0.057423	154.0	4	0.264	
weight	18	1.0902	1.031371	22.0	4	0.264	
bmi	63	3.3768	1.600300	15.5	4	0.264	
arm_length	0	-1.0848	-0.023759	34.0	7	0.461	
waist	25	1.1911	1.041341	55.4	6	0.395	
arm_circ	19	0.5714	0.862535	17.0	5	0.329	
calf_circ	6	-0.1366	0.535805	33.0	9	0.593	
calf_skinfold	49	0.7705	1.114848	NA	128	8.432	
triceps_skinfold	39	0.8936	1.135255	8.0	21	1.383	
subscapular_skinfold	77	2.2220	1.573722	NA	67	4.414	
active_days	0	0.0351	-1.095239	7.0	5	0.329	
tv_hours	0	-0.2414	0.538424	2.0	4	0.264	
computer_hours	0	0.8827	1.293648	0.0	3	0.198	
meals_out	70	12.4583	2.823921	0.0	8	0.527	
med_count	277	14.3174	3.409516	0.0	0	0.000	
	1%	5%	25%	50%	75%	95%	
SEQN	7.19e+04	71993.85	72309.25	72704.50	73100.75	73415.2	
age_child	3.00e+00	3.00	6.00	9.00	12.00	15.0	
educ_child	0.00e+00	0.00	2.00	4.00	7.00	9.0	
sampling_wt	1.07e+04	13352.52	23317.44	31605.91	43241.74	62215.7	
income_pov	2.80e-03	0.31	0.87	1.74	3.52	5.0	
age_adult	2.12e+01	26.00	33.00	40.00	45.00	56.0	
energy	5.67e+02	849.00	1367.50	1794.50	2306.00	3195.4	
protein	1.67e+01	26.53	45.33	61.25	82.57	125.7	
sugar	2.41e+01	44.55	82.66	116.91	157.05	234.0	
fat	1.16e+01	22.74	43.16	61.98	85.62	129.9	
water	0.00e+00	0.00	103.69	375.00	740.64	1718.2	

plank_time	2.00e+00	5.00	27.00	54.00	82.25	144.0
height	9.45e+01	100.73	118.35	137.60	156.78	172.2
weight	1.36e+01	15.70	22.52	34.95	53.08	77.7
bmi	1.35e+01	14.30	15.90	18.10	21.90	30.2
arm_length	1.88e+01	20.00	24.20	29.00	33.80	37.5
waist	4.61e+01	49.40	55.60	64.80	76.60	96.8
arm_circ	1.50e+01	16.10	18.40	22.00	26.20	33.3
calf_circ	1.98e+01	21.40	24.70	29.20	34.30	40.2
calf_skinfold	4.89e+00	5.95	8.60	12.00	17.00	27.7
triceps_skinfold	5.60e+00	6.80	9.10	12.40	18.00	28.0
subscapular_skinfold	4.20e+00	4.80	6.20	8.20	13.20	24.1
active_days	0.00e+00	1.00	4.00	7.00	7.00	7.0
tv_hours	0.00e+00	0.00	1.00	2.00	3.00	5.0
computer_hours	0.00e+00	0.00	0.00	0.00	2.00	4.0
meals_out	0.00e+00	0.00	0.00	1.00	2.00	5.0
med_count	0.00e+00	0.00	0.00	0.00	0.00	2.0
		99%				
SEQN	73476.8					
age_child	15.0					
educ_child	9.0					
sampling_wt	86783.8					
income_pov	5.0					
age_adult	66.0					
energy	4051.7					
protein	165.8					
sugar	309.1					
fat	172.8					
water	2841.3					
plank_time	200.9					
height	180.5					
weight	103.2					
bmi	36.9					
arm_length	39.5					
waist	112.2					
arm_circ	38.3					
calf_circ	45.3					
calf_skinfold	33.8					
triceps_skinfold	35.5					
subscapular_skinfold	31.4					
active_days	7.0					
tv_hours	5.0					
computer_hours	5.0					
meals_out	10.0					
med_count	4.0					

138CHAPTER 7. NHANES NATIONAL YOUTH FITNESS SURVEY (NNYFS)

Most of the elements of this `numSummary` should be familiar. Some new pieces include:

- `nunique` = number of unique values
- `nzeroes` = number of zeroes
- `noutlier` = number of outliers (using a standard that isn't entirely transparent to me)
- `miss` = number of rows with missing value
- `miss%` = percentage of total rows with missing values $((\text{miss}/n) * 100)$
- `5%` = 5th percentile value of that variable (value below which 5 percent of the observations may be found)

```
xda::charSummary(nnyfs)
```

	n	miss	miss%	unique
sex	1518	0	0.000	2
race_eth	1518	0	0.000	4
language	1518	0	0.000	2
educ_adult	1496	22	1.449	6
respondent	1506	12	0.791	4
salt_used	1505	13	0.856	3
diet_yesterday	1516	2	0.132	4
bmi_cat	1514	4	0.264	5
physical_last_week	1514	4	0.264	3
enjoy_recess	1240	278	18.314	6
insured	1518	0	0.000	2
phys_health	1518	0	0.000	5
access_to_care	1518	0	0.000	2
care_source	1518	0	0.000	6
asthma_ever	1518	0	0.000	2
asthma_now	1518	0	0.000	2
med_use	1518	0	0.000	2
insurance	1518	0	0.000	10
sex				
race_eth				2_White Non-Hispanic:610, 1_Hispanic:450, 3_Latin
language				
educ_adult				4_Some College:464, 5_College Graduate:386, 3_High School Graduate:368
respondent				
salt_used				
diet_yesterday				2_Usual:1175, 3_Much:342, 1_Never:101
bmi_cat				2_Normal:920, 1_Overweight:598, 3_Obesity:100
physical_last_week				
enjoy_recess				1_Strongly Agree:903, 2_Agree:257, 3_Neither Agree nor Disagree:352
insured				
phys_health				1_Excellent:743, 2_Good:214, 3_Fair:101, 4_Poor:101
access_to_care				

```
care_source          Doctor's Office:1164, Clinic or Health Center:297, No Usual Care Source:27  
asthma_ever          Never  
asthma_now           Never  
med_use              Never  
insurance             Private:736, Medicaid:470, Stat
```

The `top5levels:count` provides the top 5 unique values for each variable, sorted by their counts.

7.17 What Summaries to Report

It is usually helpful to focus on the shape, center and spread of a distribution. Bock, Velleman and DeVeaux provide some useful advice:

- If the data are skewed, report the median and IQR (or the three middle quantiles). You may want to include the mean and standard deviation, but you should point out why the mean and median differ. The fact that the mean and median do not agree is a sign that the distribution may be skewed. A histogram will help you make that point.
- If the data are symmetric, report the mean and standard deviation, and possibly the median and IQR as well.
- If there are clear outliers and you are reporting the mean and standard deviation, report them with the outliers present and with the outliers removed. The differences may be revealing. The median and IQR are not likely to be seriously affected by outliers.

Chapter 8

Assessing Normality

Data are well approximated by a Normal distribution if the shape of the data's distribution is a good match for a Normal distribution with mean and standard deviation equal to the sample statistics.

- the data are symmetrically distributed about a single peak, located at the sample mean
- the spread of the distribution is well characterized by a Normal distribution with standard deviation equal to the sample standard deviation
- the data show outlying values (both in number of candidate outliers, and size of the distance between the outliers and the center of the distribution) that are similar to what would be predicted by a Normal model.

We have several tools for assessing Normality of a single batch of data, including:

- a histogram with superimposed Normal distribution
- histogram variants (like the boxplot) which provide information on the center, spread and shape of a distribution
- the Empirical Rule for interpretation of a standard deviation
- a specialized *normal Q-Q plot* (also called a normal probability plot or normal quantile-quantile plot) designed to reveal differences between a sample distribution and what we might expect from a normal distribution of a similar number of values with the same mean and standard deviation

8.1 Empirical Rule Interpretation of the Standard Deviation

For a set of measurements that follows a Normal distribution, the interval:

- Mean \pm Standard Deviation contains approximately 68% of the measurements;
- Mean \pm 2(Standard Deviation) contains approximately 95% of the measurements;
- Mean \pm 3(Standard Deviation) contains approximately all (99.7%) of the measurements.

Again, most data sets do not follow a Normal distribution. We will occasionally think about transforming or re-expressing our data to obtain results which are better approximated by a Normal distribution, in part so that a standard deviation can be more meaningful.

For the energy data we have been studying, here again are some summary statistics...

```
mosaic::favstats(nnyfs$energy)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	257	1368	1794	2306	5265	1877	722	1518	0

The mean is 1877 and the standard deviation is 722, so if the data really were Normally distributed, we'd expect to see:

- About 68% of the data in the range (1155, 2600). In fact, 1085 of the 1518 energy values are in this range, or 71.5%.
- About 95% of the data in the range (432, 3322). In fact, 1450 of the 1518 energy values are in this range, or 95.5%.
- About 99.7% of the data in the range (-290, 4044). In fact, 1502 of the 1518 energy values are in this range, or 98.9%.

So, based on this Empirical Rule approximation, do the energy data seem to be well approximated by a Normal distribution?

8.2 Describing Outlying Values with Z Scores

The maximum energy consumption value here is 5265. One way to gauge how extreme this is (or how much of an outlier it is) uses that observation's **Z score**, the number of standard deviations away from the mean that the observation falls.

Here, the maximum value, 5265 is 4.69 standard deviations above the mean, and thus has a Z score of 4.7.

A negative Z score would indicate a point below the mean, while a positive Z score indicates, as we've seen, a point above the mean. The minimum body-mass index, 257 is 2.24 standard deviations *below* the mean, so it has a Z score of -2.2.

Recall that the Empirical Rule suggests that if a variable follows a Normal distribution, it would have approximately 95% of its observations falling inside a

Z score of (-2, 2), and 99.74% falling inside a Z score range of (-3, 3).

8.2.1 Fences and Z Scores

Note the relationship between the fences (Tukey's approach to identifying points which fall within the whiskers of a boxplot, as compared to candidate outliers) and the Z scores.

The upper inner fence in this case falls at 3713.75, which indicates a Z score of 2.5, while the lower inner fence falls at -40.25, which indicates a Z score of -2.7. It is neither unusual nor inevitable for the inner fences to fall at Z scores near -2.0 and +2.0.

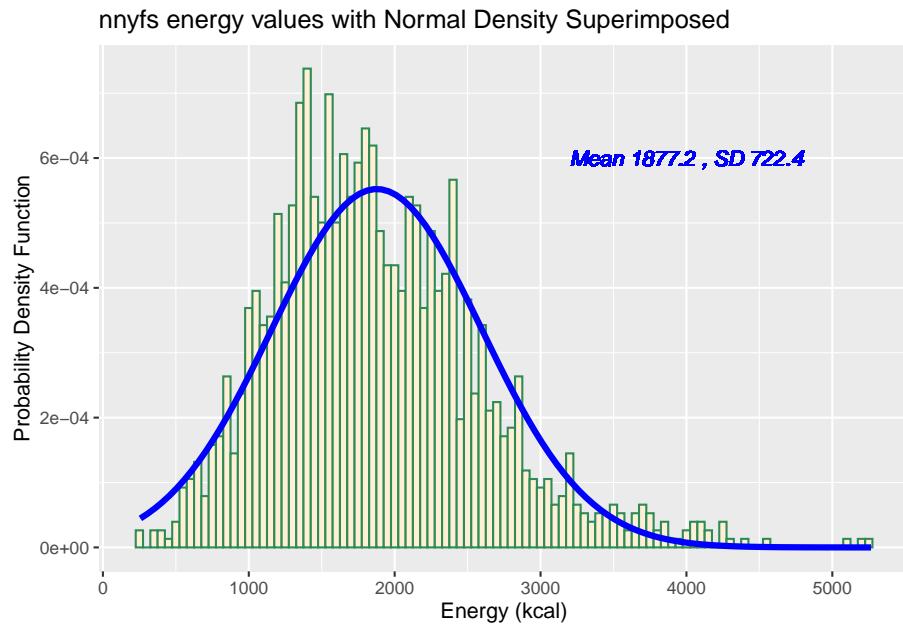
8.3 Comparing a Histogram to a Normal Distribution

Most of the time, when we want to understand whether our data are well approximated by a Normal distribution, we will use a graph to aid in the decision.

One option is to build a histogram with a Normal density function (with the same mean and standard deviation as our data) superimposed. This is one way to help visualize deviations between our data and what might be expected from a Normal distribution.

```
res <- mosaic::favstats(~ energy, data = nnyfs)
bin_w <- 50 # specify binwidth

ggplot(nnyfs, aes(x=energy)) +
  geom_histogram(aes(y = ..density..), binwidth = bin_w,
                 fill = "papayawhip", color = "seagreen") +
  stat_function(fun = dnorm,
                args = list(mean = res$mean, sd = res$sd),
                lwd = 1.5, col = "blue") +
  geom_text(aes(label = paste("Mean", round(res$mean,1),
                           ", SD", round(res$sd,1))),
            x = 4000, y = 0.0006,
            color="blue", fontface = "italic") +
  labs(title = "nnys energy values with Normal Density Superimposed",
       x = "Energy (kcal)", y = "Probability Density Function")
```



Does it seem as though the Normal model (as shown in the blue density curve) is an effective approximation to the observed distribution shown in the bars of the histogram?

We'll return shortly to the questions:

- Does a Normal distribution model fit our data well? *and*
- If the data aren't Normal, but we want to use a Normal model anyway, what should we do?

8.3.1 Histogram of energy with Normal model (with Counts)

But first, we'll demonstrate an approach to building a histogram of counts (rather than a probability density) and then superimposing a Normal model.

```
res <- mosaic::favstats(~ energy, data = nnyfs)
bin_w <- 50 # specify binwidth

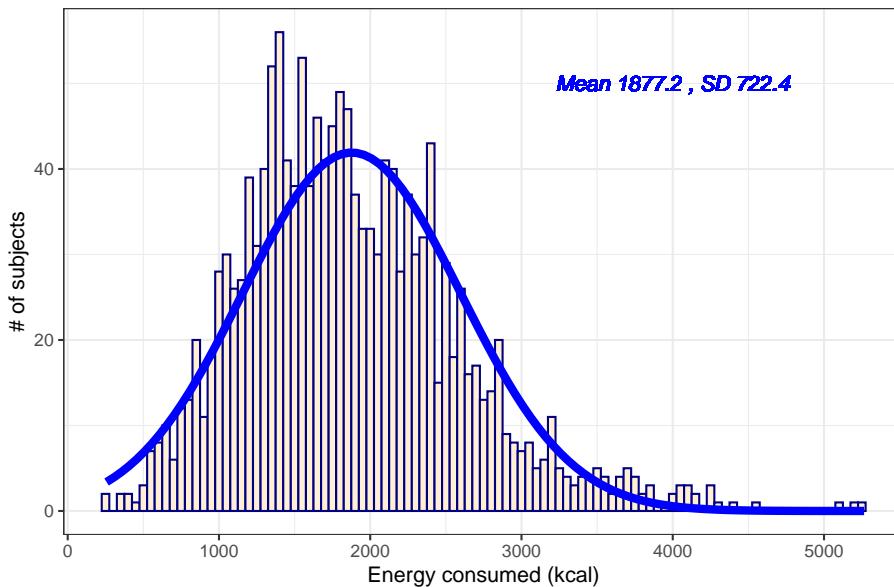
ggplot(nnyfs, aes(x = energy)) +
  geom_histogram(binwidth = bin_w,
                 fill = "papayawhip",
                 col = "navy") +
  theme_bw() +
  stat_function(
```

```

fun = function(x) dnorm(x, mean = res$mean,
                        sd = res$sd) * res$n * bin_w,
col = "blue", size = 2) +
geom_text(aes(label = paste("Mean", round(res$mean,1),
                           ", SD", round(res$sd,1))),
          x = 4000, y = 50,
          color="blue", fontface = "italic") +
labs(title = "Histogram of energy, with Normal Model",
     x = "Energy consumed (kcal)", y = "# of subjects")

```

Histogram of energy, with Normal Model



8.4 Does a Normal model work well for the waist circumference?

Now, suppose we instead look at the `waist` data, remembering to filter the data to the complete cases before plotting. Do these data appear to follow a Normal distribution?

```

res <- mosaic::favstats(~ waist, data = nnyfs)
bin_w <- 5 # specify binwidth

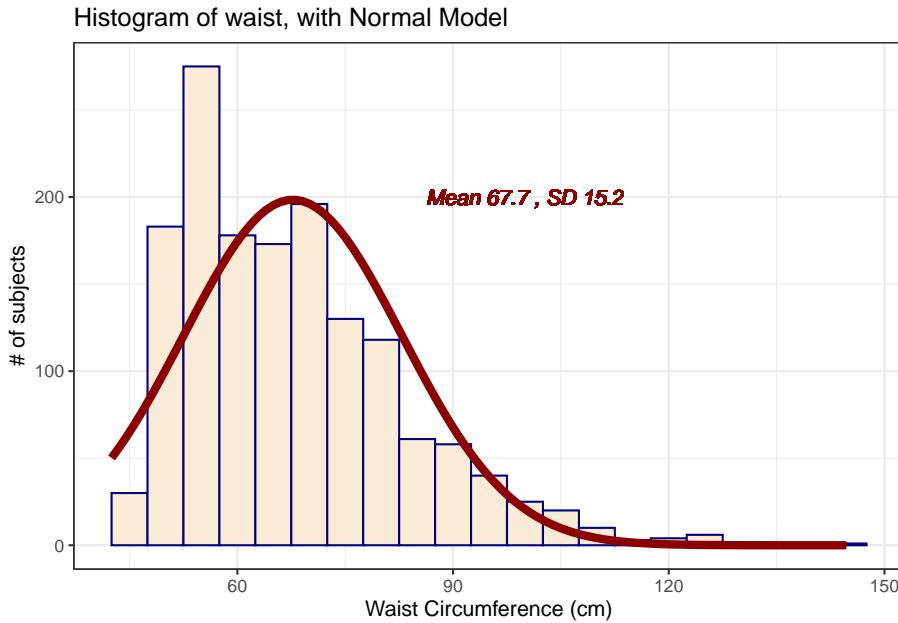
nnnyfs %>% filter(complete.cases(waist)) %>%
  ggplot(., aes(x = waist)) +
  geom_histogram(binwidth = bin_w,

```

```

            fill = "antiquewhite",
            col = "navy") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) *
      res$n * bin_w,
    col = "darkred", size = 2) +
  geom_text(aes(label = paste("Mean", round(res$mean,1),
                           ", SD", round(res$sd,1))),
            x = 100, y = 200,
            color="darkred", fontface = "italic") +
  labs(title = "Histogram of waist, with Normal Model",
       x = "Waist Circumference (cm)", y = "# of subjects")

```



```
mosaic::favstats(~ waist, data = nnyfs)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	42.5	55.6	64.8	76.6	145	67.7	15.2	1512	6

The mean is 67.71 and the standard deviation is 15.2 so if the `waist` data really were Normally distributed, we'd expect to see:

- About 68% of the data in the range (52.51, 82.9). In fact, 1076 of the 1512 Age values are in this range, or 71.2%.
- About 95% of the data in the range (37.31, 98.1). In fact, 1443 of the 1512

Age values are in this range, or 95.4%.

- About 99.7% of the data in the range (22.11, 113.3). In fact, 1500 of the 1512 Age values are in this range, or 99.2%.

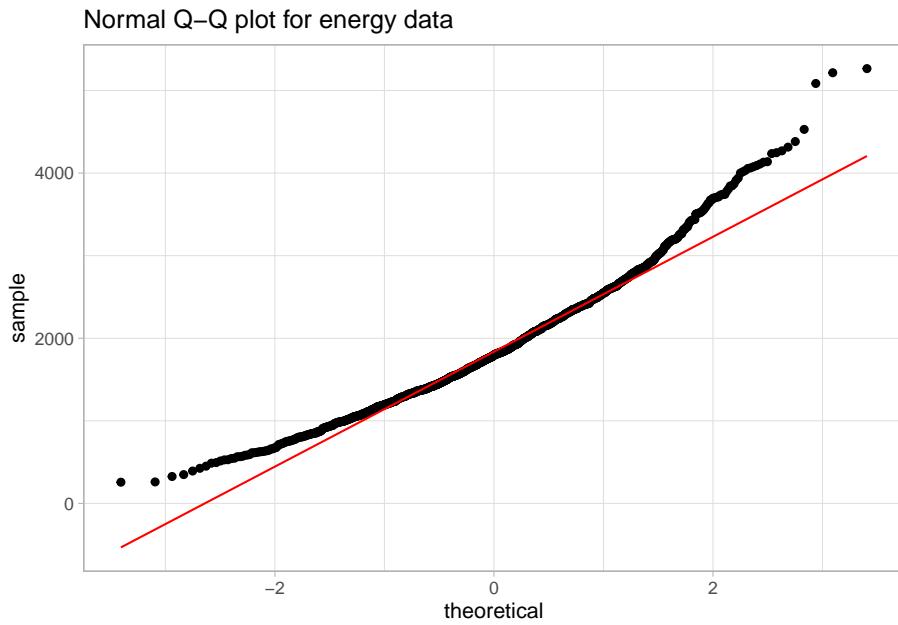
How does the Normal approximation work for waist circumference, according to the Empirical Rule?

8.5 The Normal Q-Q Plot

A normal probability plot (or normal quantile-quantile plot) of the energy results from the `nnyfs` data, developed using `ggplot2` is shown below. In this case, this is a picture of 1518 energy consumption assessments. The idea of a normal Q-Q plot is that it plots the observed sample values (on the vertical axis) and then, on the horizontal, the expected or theoretical quantiles that would be observed in a standard normal distribution (a Normal distribution with mean 0 and standard deviation 1) with the same number of observations.

A Normal Q-Q plot will follow a straight line when the data are (approximately) Normally distributed. When the data have a different shape, the plot will reflect that.

```
ggplot(nnyfs, aes(sample = energy)) +  
  geom_qq() + geom_qq_line(col = "red") +  
  theme_light() +  
  labs(title = "Normal Q-Q plot for energy data")
```



8.6 Interpreting the Normal Q-Q Plot

The purpose of a Normal Q-Q plot is to help point out distinctions from a Normal distribution. A Normal distribution is symmetric and has certain expectations regarding its tails. The Normal Q-Q plot can help us identify data as well approximated by a Normal distribution, or not, because of:

- skew (including distinguishing between right skew and left skew)
- behavior in the tails (which could be heavy-tailed [more outliers than expected] or light-tailed)

8.6.1 Data from a Normal distribution shows up as a straight line in a Normal Q-Q plot

We'll demonstrate the looks that we can obtain from a Normal Q-Q plot in some simulations. First, here is an example of a Normal Q-Q plot, and its associated histogram, for a sample of 200 observations simulated from a Normal distribution.

```
set.seed(123431) # so the results can be replicated

# simulate 200 observations from a Normal(20, 5) distribution and place them
# in the d variable within the temp.1 data frame
```

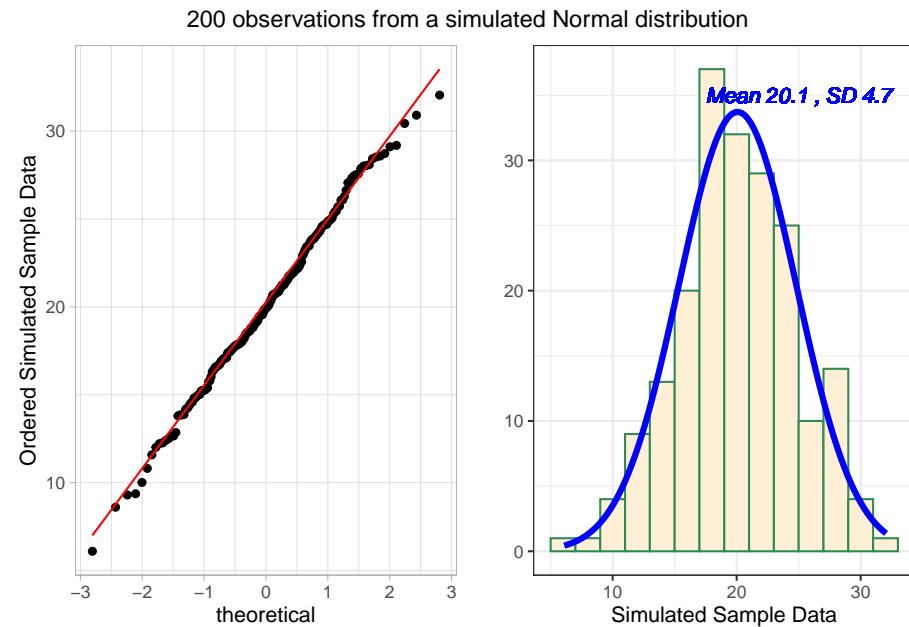
```
temp.1 <- data.frame(d = rnorm(200, mean = 20, sd = 5))

# left plot - basic Normal Q-Q plot of simulated data
p1 <- ggplot(temp.1, aes(sample = d)) +
  geom_qq() + geom_qq_line(col = "red") +
  theme_light() +
  labs(y = "Ordered Simulated Sample Data")

# right plot - histogram with superimposed normal distribution
res <- mosaic::favstats(~ d, data = temp.1)
bin_w <- 2 # specify binwidth

p2 <- ggplot(temp.1, aes(x = d)) +
  geom_histogram(binwidth = bin_w,
                 fill = "papayawhip",
                 col = "seagreen") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
      res$n * bin_w,
    col = "blue", size = 1.5) +
  geom_text(aes(label = paste("Mean", round(res$mean,1),
                            ", SD", round(res$sd,1))),
            x = 25, y = 35,
            color="blue", fontface = "italic") +
  labs(x = "Simulated Sample Data", y = "")

gridExtra::grid.arrange(p1, p2, ncol=2,
                       top ="200 observations from a simulated Normal distribution")
```



These simulated data appear to be well-modeled by the Normal distribution, because the points on the Normal Q-Q plot follow the diagonal reference line. In particular,

- there is no substantial curve (such as we'd see with data that were skewed)
- there is no particularly surprising behavior (curves away from the line) at either tail, so there's no obvious problem with outliers

8.6.2 Skew is indicated by monotonic curves in the Normal Q-Q plot

Data that come from a skewed distribution appear to curve away from a straight line in the Q-Q plot.

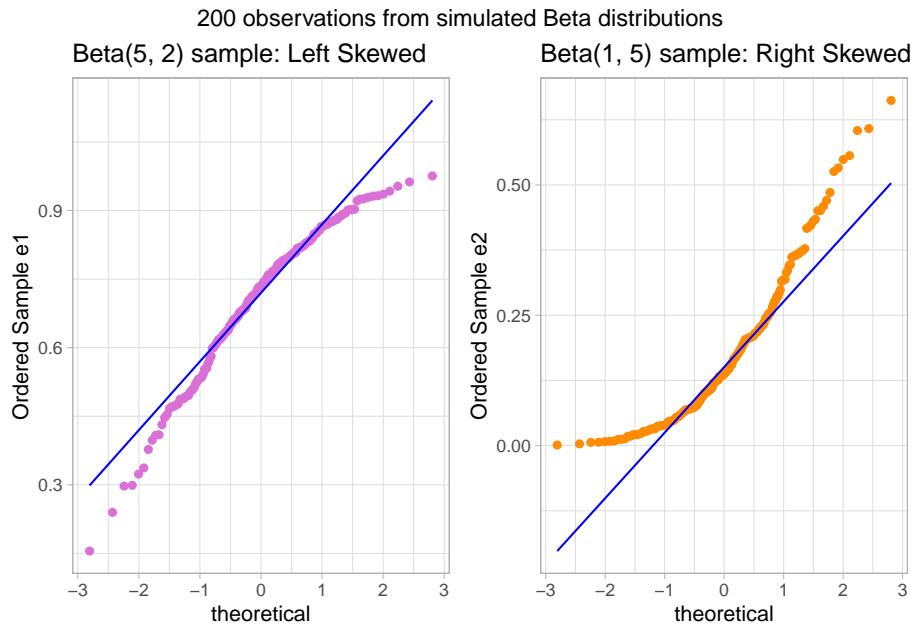
```
set.seed(123431) # so the results can be replicated

# simulate 200 observations from a beta(5, 2) distribution into the e1 variable
# simulate 200 observations from a beta(1, 5) distribution into the e2 variable
temp.2 <- data.frame(e1 = rbeta(200, 5, 2), e2 = rbeta(200, 1, 5))

p1 <- ggplot(temp.2, aes(sample = e1)) +
  geom_qq(col = "orchid") + geom_qq_line(col = "blue") +
  theme_light() +
  labs(y = "Ordered Sample e1",
       title = "Beta(5, 2) sample: Left Skewed")
```

```
p2 <- ggplot(temp.2, aes(sample = e2)) +
  geom_qq(col = "darkorange") + geom_qq_line(col = "blue") +
  theme_light() +
  labs(y = "Ordered Sample e2",
       title = "Beta(1, 5) sample: Right Skewed")

gridExtra::grid.arrange(p1, p2, ncol=2, top ="200 observations from simulated Beta distributions")
```



Note the bends away from a straight line in each sample. The non-Normality may be easier to see in a histogram.

```
res1 <- mosaic::favstats(~ e1, data = temp.2)
bin_w1 <- 0.025 # specify binwidth

p1 <- ggplot(temp.2, aes(x = e1)) +
  geom_histogram(binwidth = bin_w1,
                 fill = "orchid",
                 col = "black") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res1$mean,
                           sd = res1$sd) *
      res1$n * bin_w1,
    col = "blue", size = 1.5) +
  labs(x = "Sample e1", y = "")
```

```

title = "Beta(5,2) sample: Left Skew")

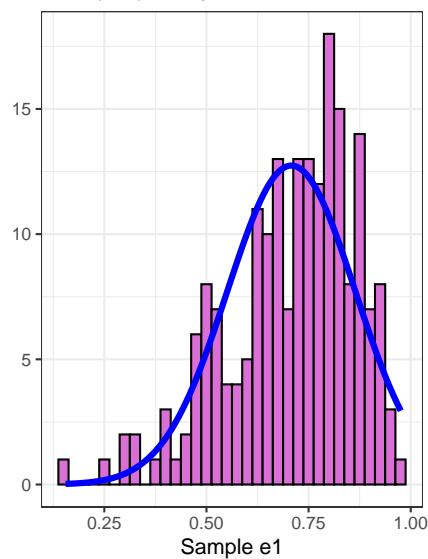
res2 <- mosaic::favstats(~ e2, data = temp.2)
bin_w2 <- 0.025 # specify binwidth

p2 <- ggplot(temp.2, aes(x = e2)) +
  geom_histogram(binwidth = bin_w2,
                 fill = "darkorange",
                 col = "black") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res2$mean,
                            sd = res2$sd) *
      res2$n * bin_w2,
    col = "blue", size = 1.5) +
  labs(x = "Sample e1", y = "",
       title = "Beta(1,5) sample: Right Skew")

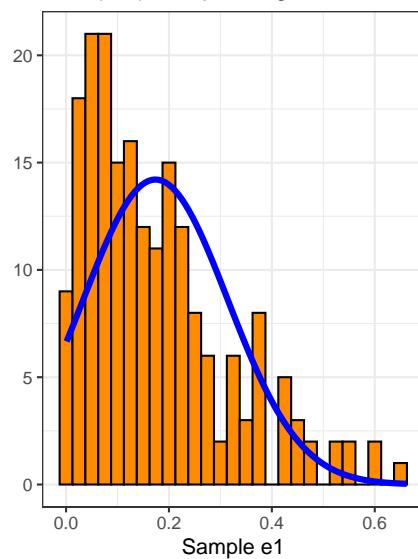
gridExtra::grid.arrange(p1, p2, ncol=2,
bottom ="Histograms with Normal curve superimposed")

```

Beta(5,2) sample: Left Skew



Beta(1,5) sample: Right Skew



Histograms with Normal curve superimposed

8.6.3 Direction of Skew

In each of these pairs of plots, we see the same basic result.

- The left plot (for data e1) shows left skew, with a longer tail on the left hand side and more clustered data at the right end of the distribution.
- The right plot (for data e2) shows right skew, with a longer tail on the right hand side, the mean larger than the median, and more clustered data at the left end of the distribution.

8.6.4 Outlier-proneness is indicated by “s-shaped” curves in a Normal Q-Q plot

- Heavy-tailed but symmetric distributions are indicated by reverse “S”-shapes, as shown on the left below.
- Light-tailed but symmetric distributions are indicated by “S” shapes in the plot, as shown on the right below.

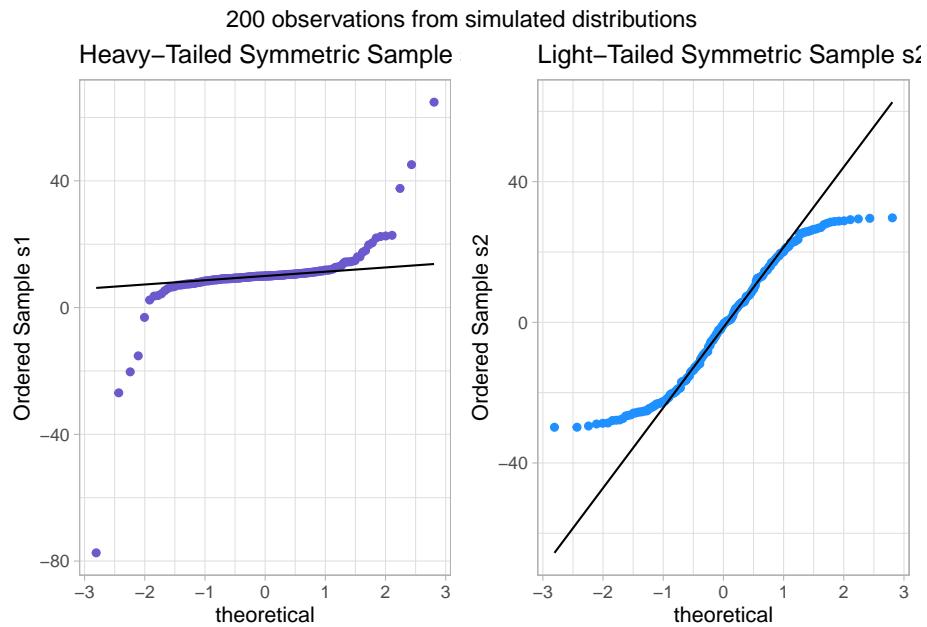
```
set.seed(4311) # so the results can be replicated

# sample 200 observations from each of two probability distributions
temp.3 <- data.frame(s1 = rcauchy(200, location=10, scale = 1),
                      s2 = runif(200, -30, 30))

p1 <- ggplot(temp.3, aes(sample = s1)) +
  geom_qq(col = "slateblue") + geom_qq_line(col = "black") +
  theme_light() +
  labs(y = "Ordered Sample s1",
       title = "Heavy-Tailed Symmetric Sample s1")

p2 <- ggplot(temp.3, aes(sample = s2)) +
  geom_qq(col = "dodgerblue") + geom_qq_line(col = "black") +
  theme_light() +
  labs(y = "Ordered Sample s2",
       title = "Light-Tailed Symmetric Sample s2")

gridExtra::grid.arrange(p1, p2, ncol=2, top ="200 observations from simulated distributions")
```



And, we can also visualize these simulations with histograms, although they're less helpful for understanding tail behavior than they are for skew.

```
res1 <- mosaic::favstats(~ s1, data = temp.3)
bin_w1 <- 20 # specify binwidth

p1 <- ggplot(temp.3, aes(x = s1)) +
  geom_histogram(binwidth = bin_w1,
                 fill = "slateblue",
                 col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res1$mean,
                           sd = res1$sd) *
      res1$n * bin_w1,
    col = "blue") +
  labs(x = "Sample s1", y = "",
       title = "Cauchy sample: Heavy Tails")

res2 <- mosaic::favstats(~ s2, data = temp.3)
bin_w2 <- 2 # specify binwidth

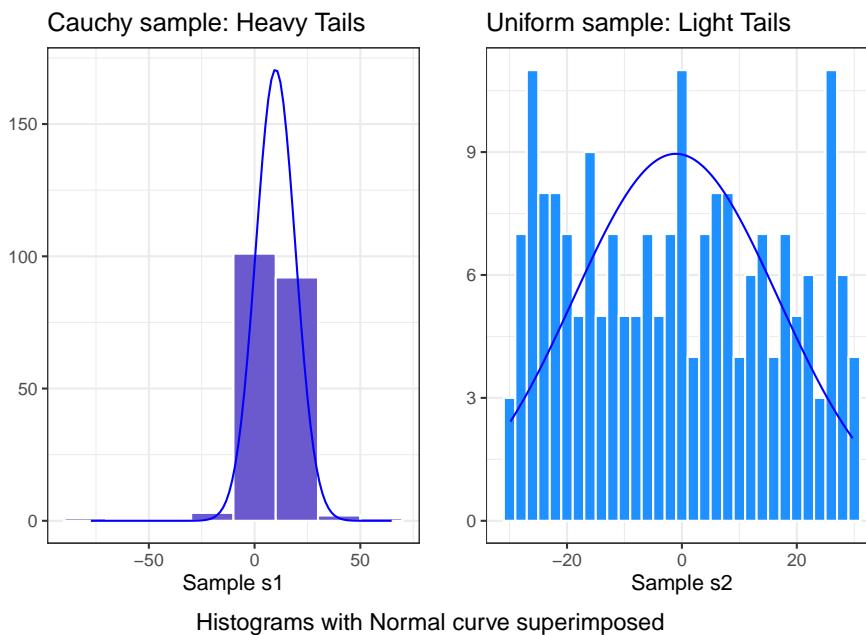
p2 <- ggplot(temp.3, aes(x = s2)) +
  geom_histogram(binwidth = bin_w2,
                 fill = "dodgerblue",
                 col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res2$mean,
                           sd = res2$sd) *
      res2$n * bin_w2,
    col = "blue") +
  labs(x = "Sample s2", y = "",
       title = "Normal sample: Light Tails")
```

```

            col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res2$mean,
                            sd = res2$sd) *
    res2$n * bin_w2,
    col = "blue") +
  labs(x = "Sample s2", y = "",
       title = "Uniform sample: Light Tails")

gridExtra::grid.arrange(p1, p2, ncol=2,
bottom ="Histograms with Normal curve superimposed")

```



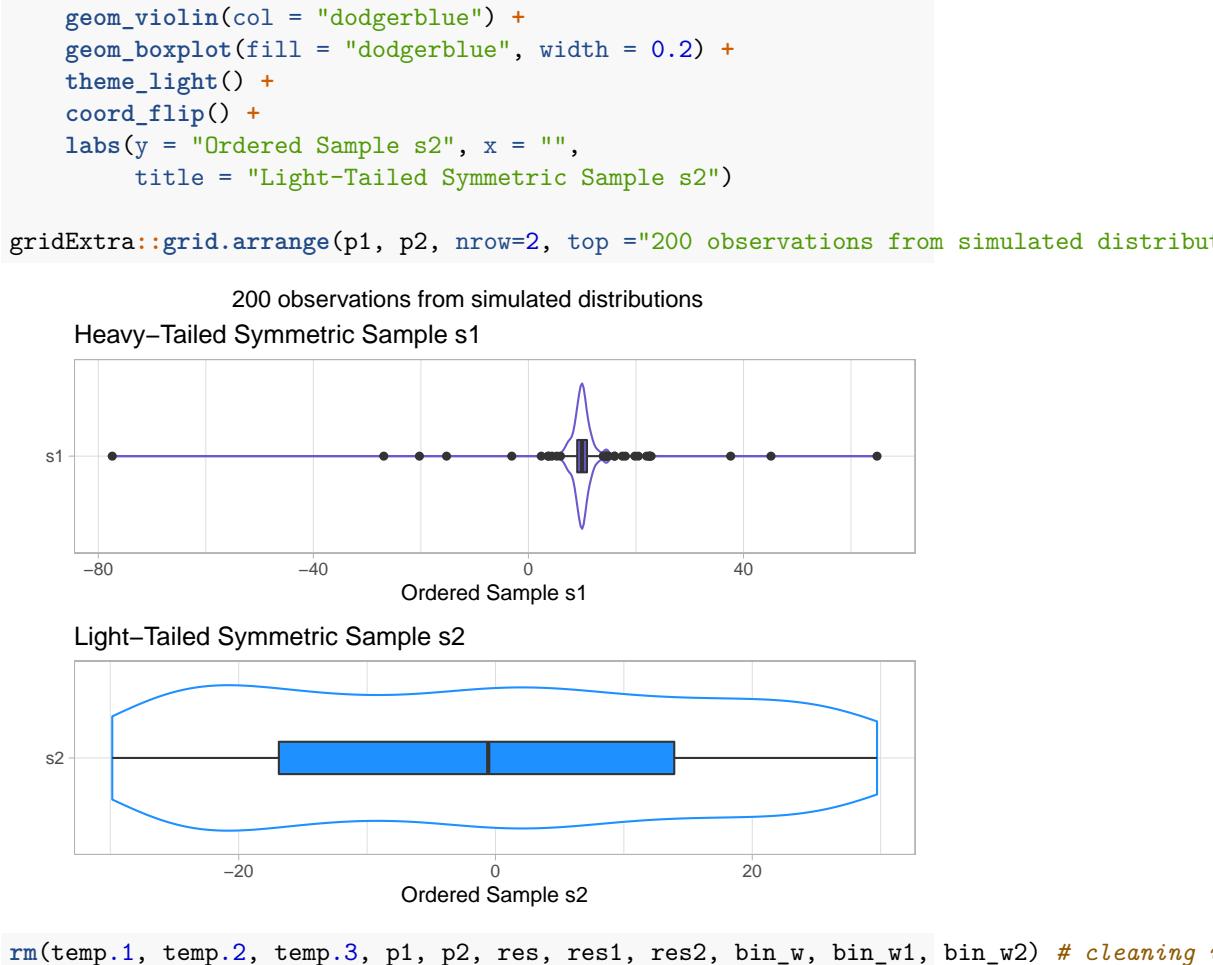
Instead, boxplots (here augmented with violin plots) can be more helpful when thinking about light-tailed vs. heavy-tailed distributions.

```

p1 <- ggplot(temp.3, aes(x = "s1", y = s1)) +
  geom_violin(col = "slateblue") +
  geom_boxplot(fill = "slateblue", width = 0.2) +
  theme_light() +
  coord_flip() +
  labs(y = "Ordered Sample s1", x = "",
       title = "Heavy-Tailed Symmetric Sample s1")

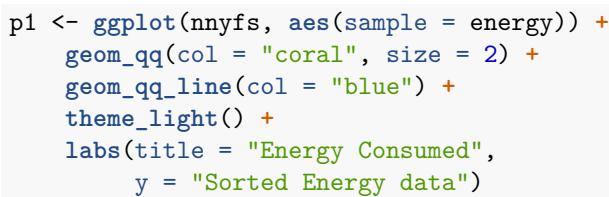
p2 <- ggplot(temp.3, aes(x = "s2", y = s2)) +

```



8.7 Can a Normal Distribution Fit the `nnyfs` energy data Well?

The `energy` data we've been studying shows meaningful signs of right skew.



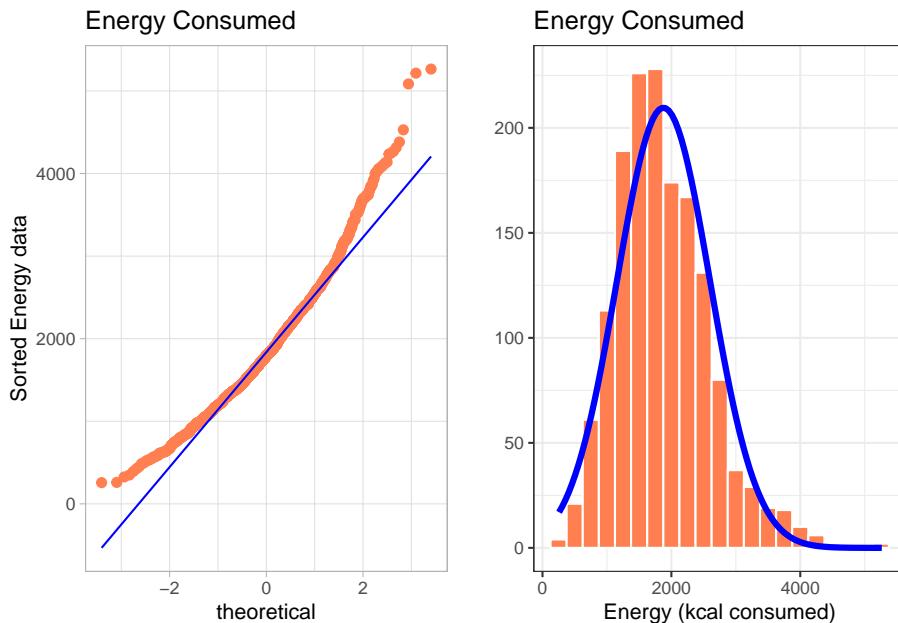
```

res <- mosaic::favstats(~ energy, data = nnyfs)
bin_w <- 250 # specify binwidth

p2 <- ggplot(nnyfs, aes(x = energy)) +
  geom_histogram(binwidth = bin_w,
                 fill = "coral",
                 col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) *
    res$n * bin_w,
    col = "blue", size = 1.5) +
  labs(x = "Energy (kcal consumed)", y = "",
       title = "Energy Consumed")

gridExtra::grid.arrange(p1, p2, ncol = 2)

```



- Skewness is indicated by the curve in the Normal Q-Q plot. Curving up and away from the line in both tails suggests right skew, as does the histogram.

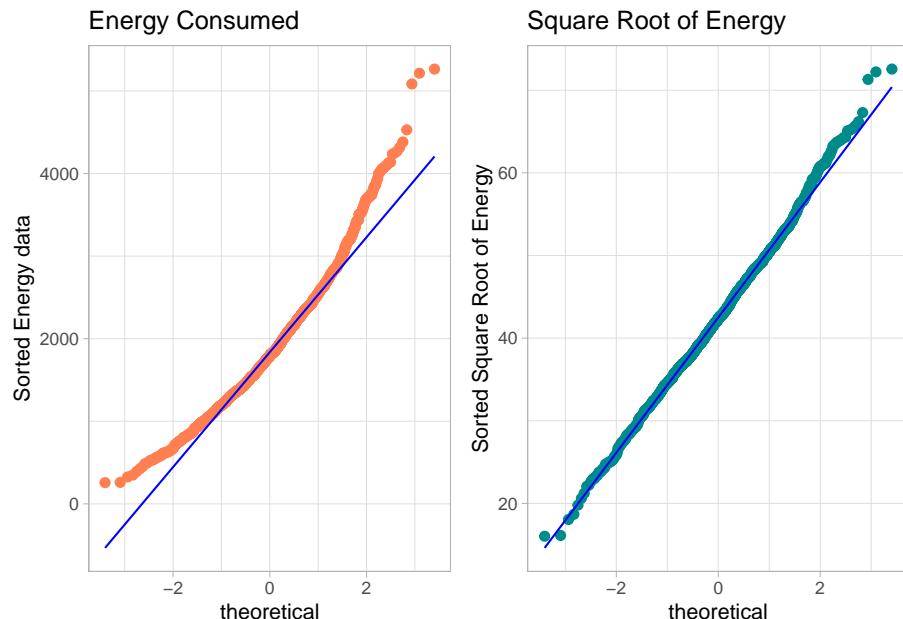
What if we plotted not the original `energy` values (all of which are positive) but instead plotted the square roots of the `energy` values?

- Compare these two plots - the left describes the distribution of the original energy data from the NNYFS data frame, and the right plot shows the distribution of the square root of those values.

```
p1 <- ggplot(nnyfs, aes(sample = energy)) +
  geom_qq(col = "coral", size = 2) +
  geom_qq_line(col = "blue") +
  theme_light() +
  labs(title = "Energy Consumed",
       y = "Sorted Energy data")

p2 <- ggplot(nnyfs, aes(sample = sqrt(energy))) +
  geom_qq(col = "darkcyan", size = 2) +
  geom_qq_line(col = "blue") +
  theme_light() +
  labs(title = "Square Root of Energy",
       y = "Sorted Square Root of Energy")

gridExtra::grid.arrange(p1, p2, ncol = 2)
```



- The left plot shows substantial **right** or *positive* skew
- The right plot shows there's much less skew after the square root has been taken.

Our conclusion is that a Normal model is a far better fit to the square root of the energy values than it is to the raw energy values.

The effect of taking the square root may be clearer from the histograms below, with Normal models superimposed.

```

res <- mosaic::favstats(~ energy, data = nnyfs)
bin_w <- 250 # specify binwidth

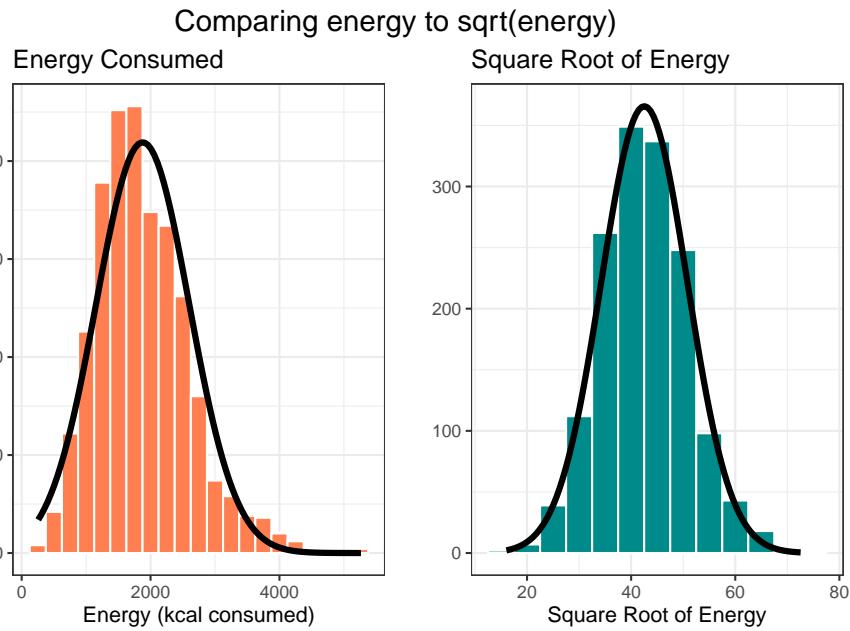
p1 <- ggplot(nnyfs, aes(x = energy)) +
  geom_histogram(binwidth = bin_w,
                 fill = "coral",
                 col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) *
      res$n * bin_w,
    col = "black", size = 1.5) +
  labs(x = "Energy (kcal consumed)", y = "",
       title = "Energy Consumed")

res2 <- mosaic::favstats(~ sqrt(energy), data = nnyfs)
bin_w2 <- 5 # specify binwidth

p2 <- ggplot(nnyfs, aes(x = sqrt(energy))) +
  geom_histogram(binwidth = bin_w2,
                 fill = "darkcyan",
                 col = "white") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res2$mean,
                            sd = res2$sd) *
      res2$n * bin_w2,
    col = "black", size = 1.5) +
  labs(x = "Square Root of Energy", y = "",
       title = "Square Root of Energy")

gridExtra::grid.arrange(p1, p2, ncol=2,
                       top =
textGrob("Comparing energy to sqrt(energy)",
        gp=gpar(fontsize=15)))

```



```
rm(p1, p2, bin_w, bin_w2, res, res2) # cleanup
```

When we are confronted with a variable that is not Normally distributed but that we wish was Normally distributed, it is sometimes useful to consider whether working with a **transformation** of the data will yield a more helpful result, as the square root does in this instance.

The next Chapter provides some initial guidance about choosing between a class of power transformations that can reduce the impact of non-Normality in unimodal data.

Chapter 9

Using Transformations to “Normalize” Distributions

- When we are confronted with a variable that is not Normally distributed but that we wish was Normally distributed, it is sometimes useful to consider whether working with a transformation of the data will yield a more helpful result.
- Many statistical methods, including t tests and analyses of variance, assume Normal distributions.
- We'll discuss using R to assess a range of what are called Box-Cox power transformations, via plots, mainly.

9.1 The Ladder of Power Transformations

The key notion in re-expression of a single variable to obtain a distribution better approximated by the Normal or re-expression of an outcome in a simple regression model is that of a **ladder of power transformations**, which applies to any unimodal data.

Power	Transformation
3	x^3
2	x^2
1	x (unchanged)
0.5	$x^{0.5} = \sqrt{x}$
0	$\ln x$
-0.5	$x^{-0.5} = 1/\sqrt{x}$
-1	$x^{-1} = 1/x$
-2	$x^{-2} = 1/x^2$

Power	Transformation
-------	----------------

9.2 Using the Ladder

As we move further away from the *identity* function (power = 1) we change the shape more and more in the same general direction.

- For instance, if we try a logarithm, and this seems like too much of a change, we might try a square root instead.
- Note that this ladder (which like many other things is due to John Tukey) uses the logarithm for the “power zero” transformation rather than the constant, which is what x^0 actually is.
- If the variable x can take on negative values, we might take a different approach. If x is a count of something that could be zero, we often simply add 1 to x before transformation.

The ladder of power transformations is particularly helpful when we are confronted with data that shows skew.

- To handle right skew (where the mean exceeds the median) we usually apply powers below 1.
- To handle left skew (where the median exceeds the mean) we usually apply powers greater than 1.

The most common transformations are the square (power 2), the square root (power 1/2), the logarithm (power 0) and the inverse (power -1), and I usually restrict myself to those options in practical work.

9.3 Protein Consumption in the NNYFS data

Here are the protein consumption (in grams) results from the NNYFS data.

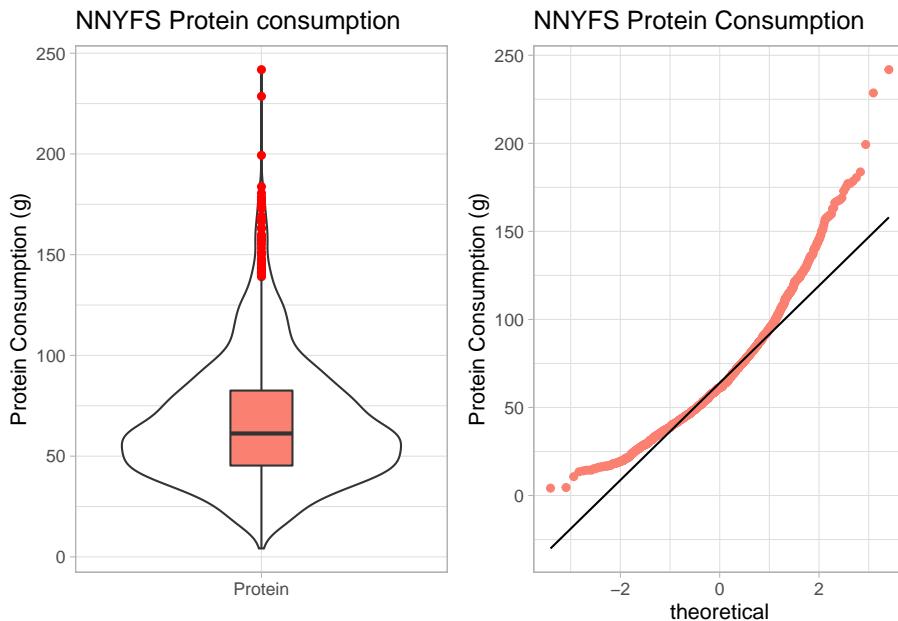
```
mosaic::favstats(~ protein, data = nnyfs)

    min   Q1 median   Q3 max mean sd      n missing
4.18 45.3   61.3 82.6 242 66.9 31 1518       0

p1 <- ggplot(nnyfs, aes(x = "Protein", y = protein)) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "salmon",
               outlier.color = "red") +
  theme_light() +
  labs(title = "NNYFS Protein consumption",
       x = "", y = "Protein Consumption (g)")
```

```
p2 <- ggplot(nnyfs, aes(sample = protein)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "NNYFS Protein Consumption",
       y = "Protein Consumption (g)")

gridExtra::grid.arrange(p1, p2, nrow = 1)
```



The key point here is that we see several signs of meaningful right skew, and we'll want to consider a transformation that might make a Normal model more plausible.

9.3.1 Using `cowplot::plot_grid()` to compose plots

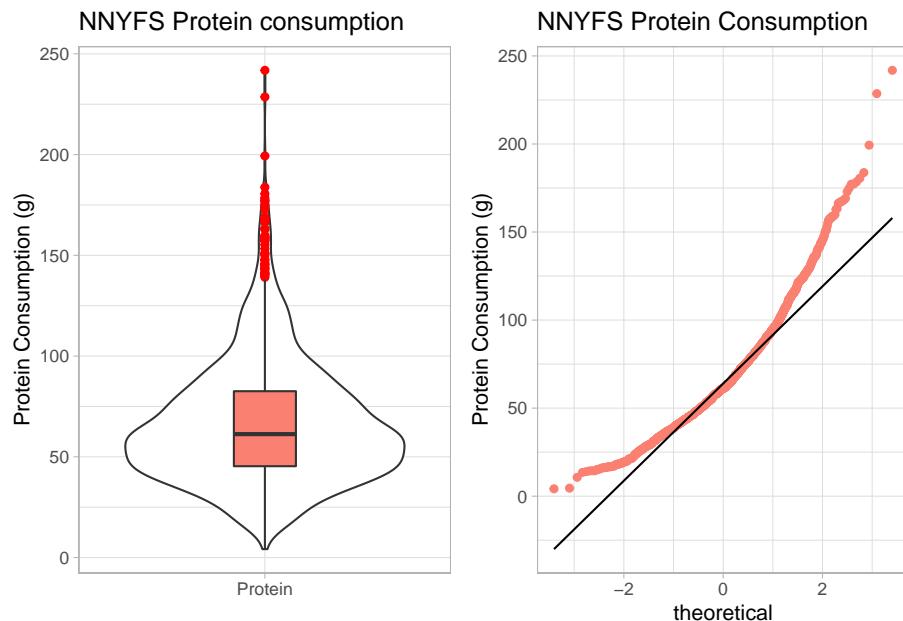
In addition to the `grid.arrange` function in `gridExtra`, we could also use `plot_grid()` in the `cowplot` package to put our plots together.

```
p1 <- ggplot(nnyfs, aes(x = "Protein", y = protein)) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "salmon",
               outlier.color = "red") +
  theme_light() +
  labs(title = "NNYFS Protein consumption",
```

```
x = "", y = "Protein Consumption (g)"

p2 <- ggplot(nnyfs, aes(sample = protein)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "NNYFS Protein Consumption",
       y = "Protein Consumption (g)")

cowplot::plot_grid(p1, p2, nrow = 1)
```



`plot_grid()` has some functionality that's easier to work with than `grid.arrange` for some people, so you might want to learn more about it. Check out its web site at <https://wilkelab.org/cowplot/>.

9.3.2 Using patchwork to compose plots

For me, the slickest approach to composing how a series of plots are placed together is available in the `patchwork` package, which must be installed via Github using `devtools` at this writing.

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
```

```
library(patchwork)

res <- mosaic::favstats(~ protein, data = nnyfs)
bin_w <- 5 # specify binwidth

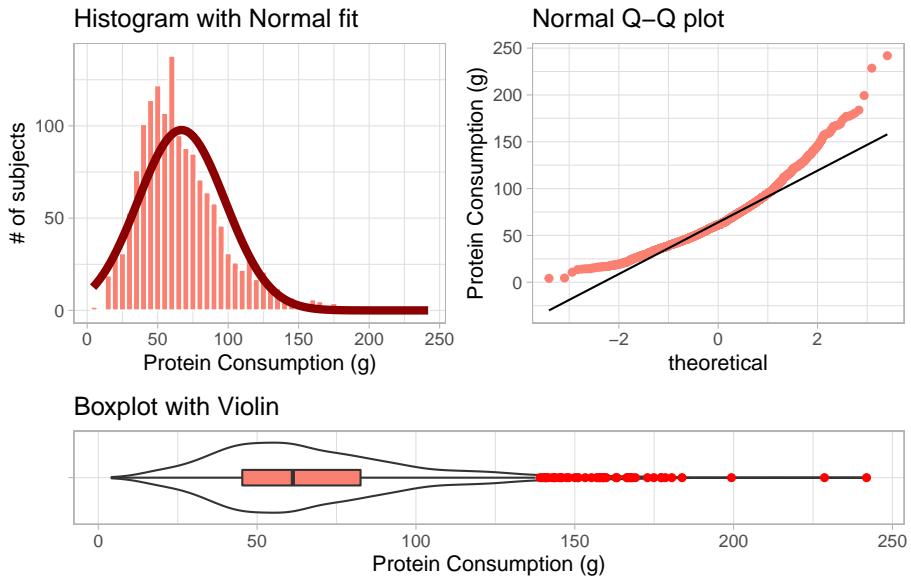
p1 <- ggplot(nnyfs, aes(x = protein)) +
  geom_histogram(binwidth = bin_w,
                 fill = "salmon",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) *
      res$n * bin_w,
    col = "darkred", size = 2) +
  labs(title = "Histogram with Normal fit",
       x = "Protein Consumption (g)", y = "# of subjects")

p2 <- ggplot(nnyfs, aes(sample = protein)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
       y = "Protein Consumption (g)")

p3 <- ggplot(nnyfs, aes(x = "", y = protein)) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "salmon",
               outlier.color = "red") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
       x = "", y = "Protein Consumption (g)")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "NNYFS Protein Consumption")
```

NNYFS Protein Consumption



For more on the `patchwork` package, check out its repository at <https://github.com/thomasp85/patchwork>.

9.4 Can we transform the protein data?

As we've seen, the `protein` data are right skewed, and all of the values are strictly positive. If we want to use the tools of the Normal distribution to describe these data, we might try taking a step “down” our ladder from power 1 (raw data) to lower powers.

9.4.1 The Square Root

Would a square root applied to the protein data help alleviate that right skew?

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

res <- mosaic::favstats(~ sqrt(protein), data = nnyfs)
bin_w <- 1 # specify binwidth

p1 <- ggplot(nnyfs, aes(x = sqrt(protein))) +
  geom_histogram(binwidth = bin_w,
```

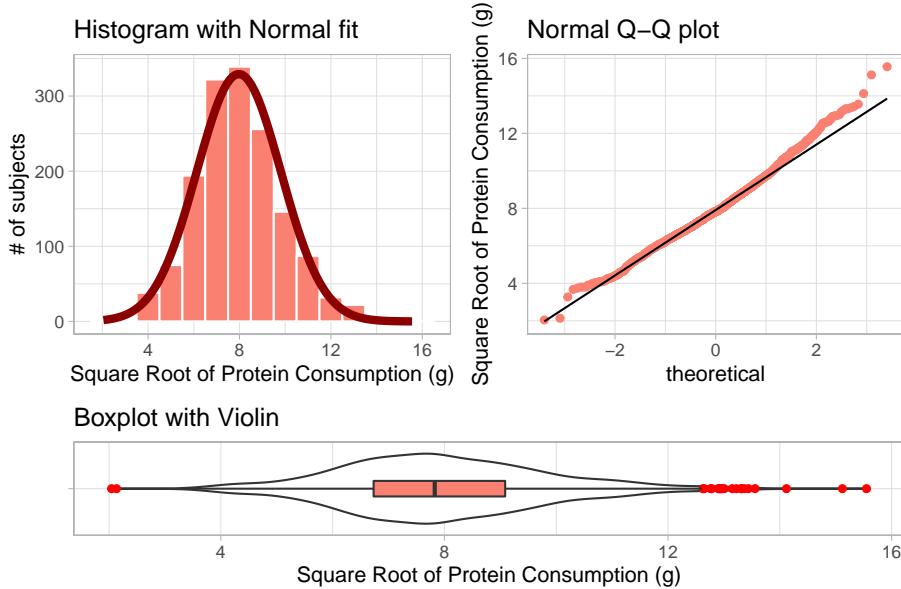
```
    fill = "salmon",
    col = "white") +
theme_light() +
stat_function(
  fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
  res$n * bin_w,
  col = "darkred", size = 2) +
labs(title = "Histogram with Normal fit",
  x = "Square Root of Protein Consumption (g)", y = "# of subjects")

p2 <- ggplot(nnyfs, aes(sample = sqrt(protein))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
    y = "Square Root of Protein Consumption (g)")

p3 <- ggplot(nnyfs, aes(x = "", y = sqrt(protein))) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "salmon",
               outlier.color = "red") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
    x = "", y = "Square Root of Protein Consumption (g)")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Square Root of NNYFS Protein Consumption")
```

Square Root of NNYFS Protein Consumption



That looks like a more symmetric distribution, certainly, although we still have some outliers on the right side of the distribution. Should we take another step down the ladder?

9.4.2 The Logarithm

We might also try a logarithm of the energy circumference data. We can use either the natural logarithm (`log`, in R) or the base-10 logarithm (`log10`, in R) - either will have the same impact on skew.

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

res <- mosaic::favstats(~ log(protein), data = nnyfs)
bin_w <- 0.5 # specify binwidth

p1 <- ggplot(nnyfs, aes(x = log(protein))) +
  geom_histogram(binwidth = bin_w,
                 fill = "salmon",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
```

```

    res$n * bin_w,
    col = "darkred", size = 2) +
  labs(title = "Histogram with Normal fit",
       x = "Log of Protein Consumption (g)", y = "# of subjects")

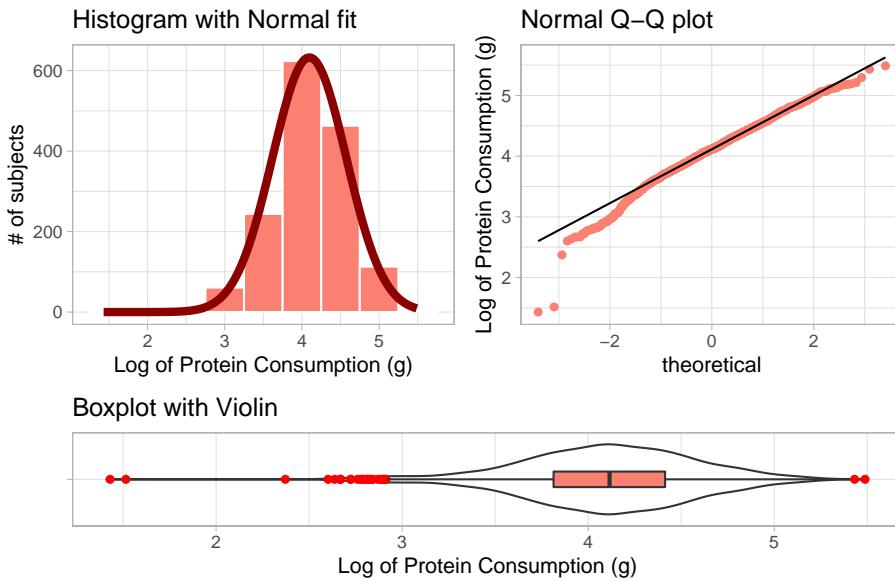
p2 <- ggplot(nnyfs, aes(sample = log(protein))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
       y = "Log of Protein Consumption (g)")

p3 <- ggplot(nnyfs, aes(x = "", y = log(protein))) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "salmon",
               outlier.color = "red") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
       x = "", y = "Log of Protein Consumption (g)")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Logarithm of NNYFS Protein Consumption")

```

Logarithm of NNYFS Protein Consumption



Now, it looks like we may have gone too far in the other direction. It looks like the square root is a sensible choice to try to improve the fit of a Normal model to the protein consumption data.

9.4.3 This course uses Natural Logarithms, unless otherwise specified

In this course, we will assume the use of natural logarithms unless we specify otherwise. Following R’s convention, we will use `log` for natural logarithms.

9.5 What if we considered all 9 available transformations?

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

p1 <- ggplot(nnyfs, aes(sample = protein^3)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Cube (power 3)",
       y = "Protein, Cubed")

p2 <- ggplot(nnyfs, aes(sample = protein^2)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Square (power 2)",
       y = "Protein, Squared")

p3 <- ggplot(nnyfs, aes(sample = protein)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Original Data",
       y = "Protein (g)")

p4 <- ggplot(nnyfs, aes(sample = sqrt(protein))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
```

```

theme_light() +
  labs(title = "sqrt (power 0.5)",
       y = "Square Root of Protein")

p5 <- ggplot(nnyfs, aes(sample = log(protein))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "log (power 0)",
       y = "Natural Log of Protein")

p6 <- ggplot(nnyfs, aes(sample = protein-0.5)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/sqrt (power -0.5)",
       y = "1/Square Root(Protein)")

p7 <- ggplot(nnyfs, aes(sample = 1/protein)) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Inverse (power -1)",
       y = "1/Protein")

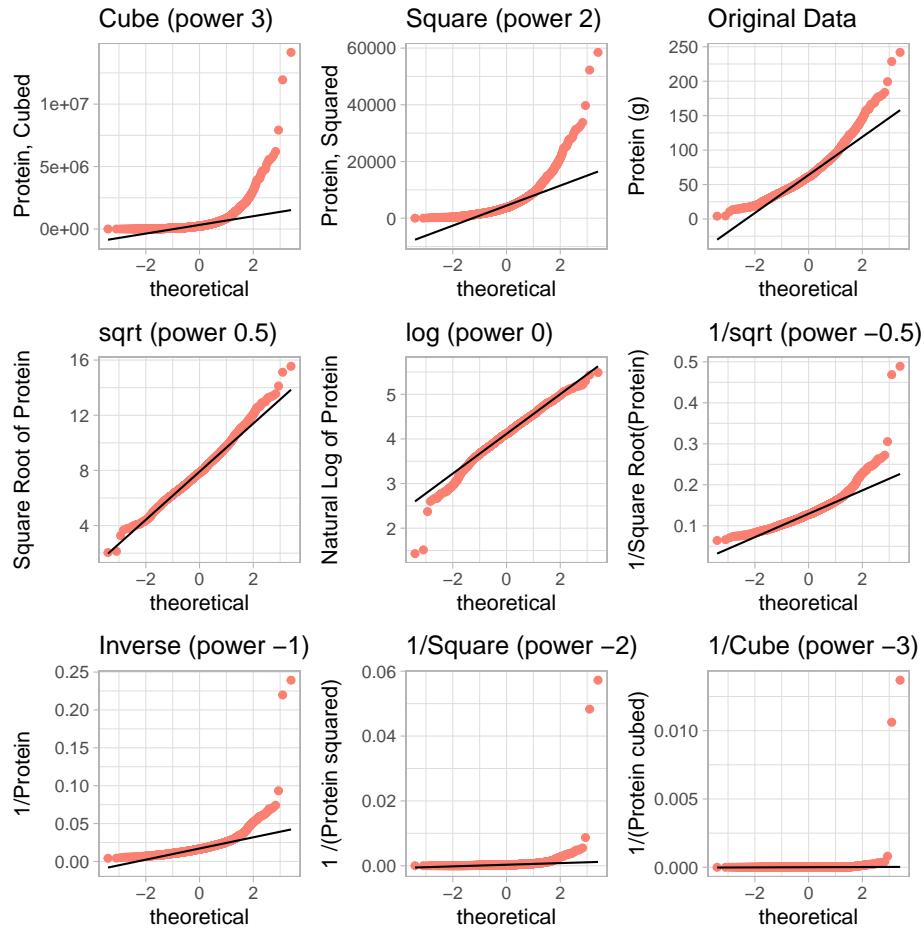
p8 <- ggplot(nnyfs, aes(sample = 1/(protein2))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/Square (power -2)",
       y = "1 /(Protein squared)")

p9 <- ggplot(nnyfs, aes(sample = 1/(protein3))) +
  geom_qq(col = "salmon") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/Cube (power -3)",
       y = "1/(Protein cubed)")

p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 +
  plot_layout(nrow = 3) +
  plot_annotation(title = "Transformations of NNYFS Protein Consumption")

```

Transformations of NNYFS Protein Consumption



The square root still appears to be the best choice of transformation here, even after we consider all 8 transformation of the raw data.

9.6 A Simulated Data Set

```
set.seed(431);
data2 <- data.frame(sample2 = 100*rbeta(n = 125, shape1 = 5, shape2 = 2))
```

If we'd like to transform these data so as to better approximate a Normal distribution, where should we start? What transformation do you suggest?

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

res <- mosaic::favstats(~ sample2, data = data2)
bin_w <- 4 # specify binwidth

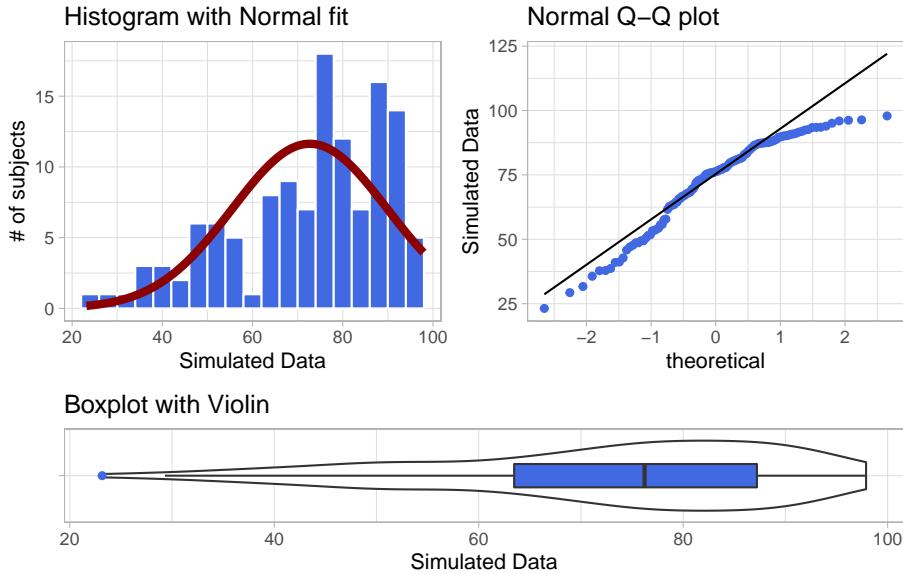
p1 <- ggplot(data2, aes(x = sample2)) +
  geom_histogram(binwidth = bin_w,
                 fill = "royalblue",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
      res$n * bin_w,
    col = "darkred", size = 2) +
  labs(title = "Histogram with Normal fit",
       x = "Simulated Data", y = "# of subjects")

p2 <- ggplot(data2, aes(sample = sample2)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
       y = "Simulated Data")

p3 <- ggplot(data2, aes(x = "", y = sample2)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "royalblue",
               outlier.color = "royalblue") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
       x = "", y = "Simulated Data")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Simulated Data")
```

Simulated Data



Given the left skew in the data, it looks like a step up in the ladder is warranted, perhaps by looking at the square of the data?

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

res <- mosaic::favstats(~ sample2^2, data = data2)
bin_w <- 600 # specify binwidth

p1 <- ggplot(data2, aes(x = sample2^2)) +
  geom_histogram(binwidth = bin_w,
                 fill = "royalblue",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
      res$n * bin_w,
    col = "darkred", size = 2) +
  labs(title = "Histogram with Normal fit",
       x = "Squared Simulated Data", y = "# of subjects")

p2 <- ggplot(data2, aes(sample = sample2^2)) +
```

```

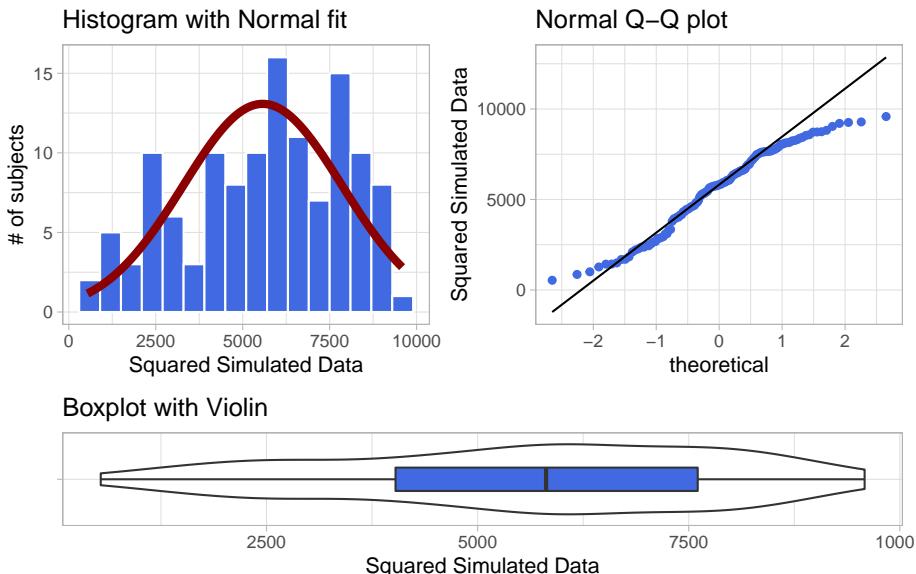
geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
       y = "Squared Simulated Data")

p3 <- ggplot(data2, aes(x = "", y = sample2^2)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "royalblue",
               outlier.color = "royalblue") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
       x = "", y = "Squared Simulated Data")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Squared Simulated Data")

```

Squared Simulated Data



Looks like at best a modest improvement. How about cubing the data, instead?

```

## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

res <- mosaic::favstats(~ sample2^3, data = data2)

```

```

bin_w <- 100000 # specify binwidth

p1 <- ggplot(data2, aes(x = sample2^3)) +
  geom_histogram(binwidth = bin_w,
                 fill = "royalblue",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) *
    res$n * bin_w,
    col = "darkred", size = 2) +
  labs(title = "Histogram with Normal fit",
       x = "Cubed Simulated Data", y = "# of subjects")

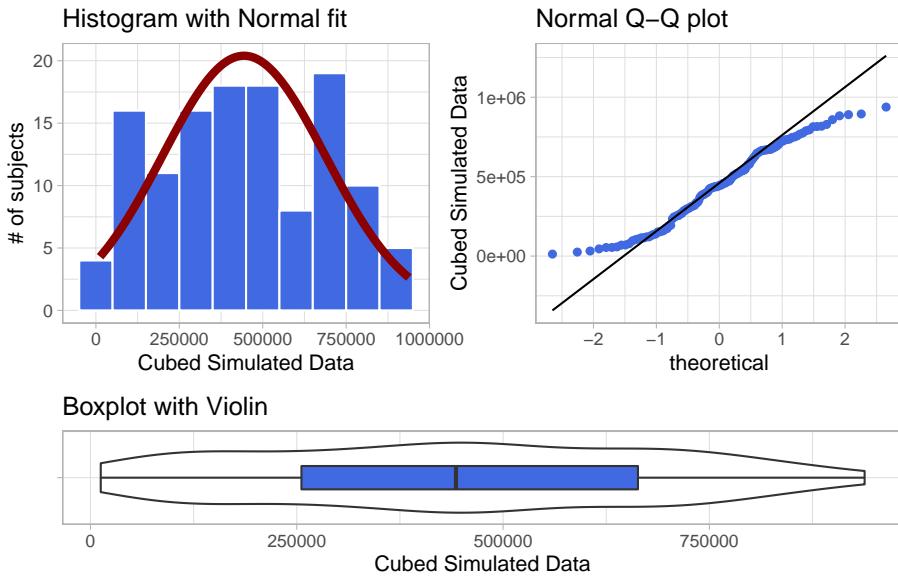
p2 <- ggplot(data2, aes(sample = sample2^3)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Normal Q-Q plot",
       y = "Cubed Simulated Data")

p3 <- ggplot(data2, aes(x = "", y = sample2^3)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "royalblue",
               outlier.color = "royalblue") +
  theme_light() +
  coord_flip() +
  labs(title = "Boxplot with Violin",
       x = "", y = "Cubed Simulated Data")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Cubed Simulated Data")

```

Cubed Simulated Data



The newly transformed (cube of the) data appears more symmetric, although somewhat light-tailed. Perhaps a Normal model would be more appropriate now, although the standard deviation is likely to overstate the variation we see in the data due to the light tails. Again, I wouldn't be thrilled using a cube in practical work, as it is so hard to interpret, but it does look like a reasonable choice here.

9.7 What if we considered all 9 available transformations?

```
## must install patchwork from Github using devtools
## with devtools::install_github("thomasp85/patchwork")
## library(patchwork)

p1 <- ggplot(data2, aes(sample = sample2^3)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Cube (power 3)")

p2 <- ggplot(data2, aes(sample = sample2^2)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
```

```

theme_light() +
  labs(title = "Square (power 2)")

p3 <- ggplot(data2, aes(sample = sample2)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Original Data")

p4 <- ggplot(data2, aes(sample = sqrt(sample2))) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "sqrt (power 0.5)")

p5 <- ggplot(data2, aes(sample = log(sample2))) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "log (power 0)")

p6 <- ggplot(data2, aes(sample = sample2^(0.5))) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/sqrt (power -0.5)")

p7 <- ggplot(data2, aes(sample = 1/sample2)) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "Inverse (power -1)")

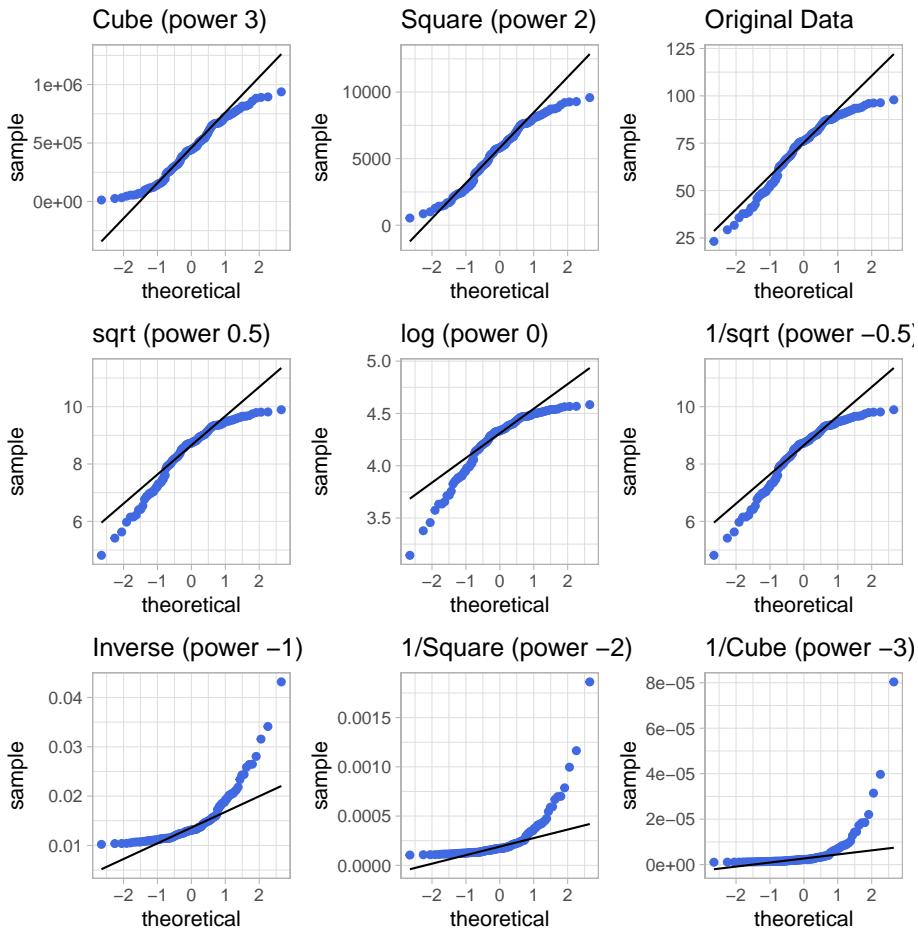
p8 <- ggplot(data2, aes(sample = 1/(sample2^2))) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/Square (power -2)")

p9 <- ggplot(data2, aes(sample = 1/(sample2^3))) +
  geom_qq(col = "royalblue") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(title = "1/Cube (power -3)")

```

```
p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 +
plot_layout(nrow = 3) +
plot_annotation(title = "Transformations of Simulated Sample")
```

Transformations of Simulated Sample



Again, either the cube or the square looks like best choice here, in terms of creating a more symmetric (albeit light-tailed) distribution.

```
rm(p1, p2, p3, p4, p5, p6, p7, p8, p9, res, bin_w, data2)
```


Chapter 10

Summarizing data within subgroups

10.1 Using dplyr and summarise to build a tibble of summary information

Suppose we want to understand how the subjects whose diet involved consuming much more than usual yesterday compare to those who consumer their usual amount, or to those who consumed much less than usual, in terms of the energy they consumed, as well as the protein. We might start by looking at the medians and means.

```
nyfs %>%
  group_by(diet_yesterday) %>%
  select(diet_yesterday, energy, protein) %>%
  summarise_all(funs(median, mean))

# A tibble: 4 x 5
  diet_yesterday   energy_median protein_median energy_mean protein_mean
  <chr>           <dbl>        <dbl>       <dbl>        <dbl>
1 1_Much more than u~    2098        69.4      2150.       75.1
2 2_Usual            1794        61.3      1858.       67.0
3 3_Much less than u~   1643        53.9      1779.       60.1
4 <NA>                4348        155.      4348        155.
```

Perhaps we should restrict ourselves to the people who were not missing the diet_yesterday category, and look now at their sugar and water consumption.

```
nyfs %>%
  filter(complete.cases(diet_yesterday)) %>%
```

```
group_by(diet_yesterday) %>%
  select(diet_yesterday, energy, protein, sugar, water) %>%
  summarise_all(funs(median))

# A tibble: 3 x 5
  diet_yesterday      energy  protein  sugar  water
  <chr>            <dbl>    <dbl>    <dbl>   <dbl>
1 1_Much more than usual    2098     69.4   137.   500
2 2_Usual                1794     61.3   114.   385.
3 3_Much less than usual   1643     53.9   115.   311.
```

It looks like the children in the “Much more than usual” category consumed more energy, protein, sugar and water than the children in the other two categories. Let’s draw a picture of this.

```
library(patchwork)

temp_dat <- nnyfs %>%
  filter(complete.cases(diet_yesterday)) %>%
  mutate(diet_yesterday = fct_recode(diet_yesterday,
    "Much more" = "1_Much more than usual",
    "Usual diet" = "2_Usual",
    "Much less" = "3_Much less than usual"))

p1 <- ggplot(temp_dat, aes(x = diet_yesterday, y = energy)) +
  geom_violin() +
  geom_boxplot(aes(fill = diet_yesterday), width = 0.2) +
  theme_light() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Energy Comparison")

p2 <- ggplot(temp_dat, aes(x = diet_yesterday, y = protein)) +
  geom_violin() +
  geom_boxplot(aes(fill = diet_yesterday), width = 0.2) +
  theme_light() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Protein Comparison")

p3 <- ggplot(temp_dat, aes(x = diet_yesterday, y = sugar)) +
  geom_violin() +
  geom_boxplot(aes(fill = diet_yesterday), width = 0.2) +
  theme_light() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
```

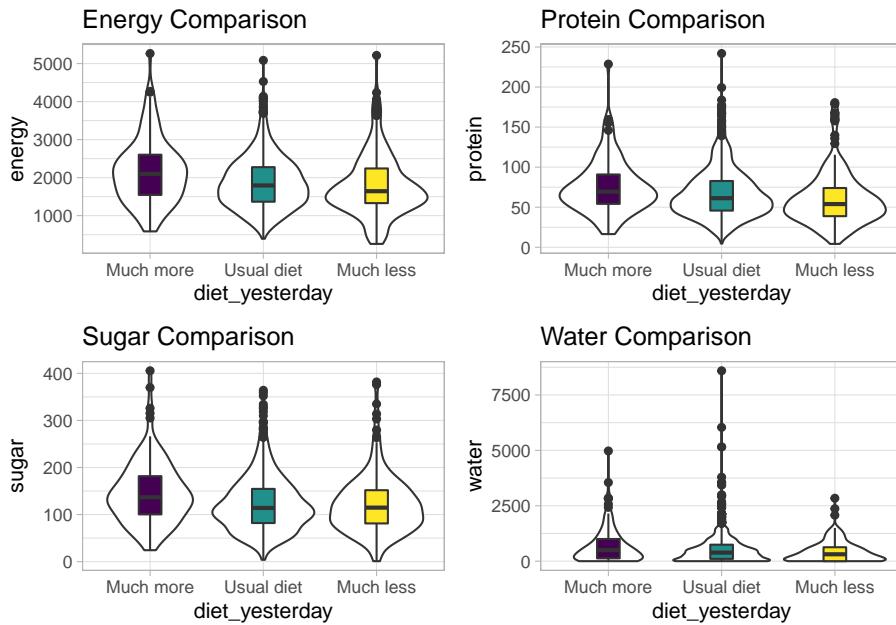
```

  labs(title = "Sugar Comparison")

p4 <- ggplot(temp_dat, aes(x = diet_yesterday, y = water)) +
  geom_violin() +
  geom_boxplot(aes(fill = diet_yesterday), width = 0.2) +
  theme_light() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Water Comparison")

p1 + p2 + p3 + p4

```



We can see that there is considerable overlap in these distributions, regardless of what we're measuring.

10.2 Another Example

Suppose now that we ask a different question. Do kids in larger categories of BMI have larger waist circumferences?

```

nnyfs %>%
  group_by(bmi_cat) %>%
  summarise(mean = mean(waist), sd = sd(waist),
            median = median(waist),

```

```
skew_1 = round((mean(waist) - median(waist)) /
sd(waist), 2))
```

```
# A tibble: 5 x 5
  bmi_cat      mean     sd median skew_1
  <chr>       <dbl>   <dbl>  <dbl>   <dbl>
1 1_Underweight 55.2    7.58   54.5    0.09
2 2_Normal      NA     NaN     NA     NA
3 3_Overweight  72.3   11.9    74    -0.14
4 4_Obese        NA     NaN     NA     NA
5 <NA>          NA     NaN     NA     NA
```

Oops. Looks like we need to filter for cases with complete data on both BMI category and waist circumference in order to get meaningful results. We should add a count, too.

```
nnyfs %>%
  filter(complete.cases(bmi_cat, waist)) %>%
  group_by(bmi_cat) %>%
  summarise(count = n(), mean = mean(waist),
            sd = sd(waist), median = median(waist),
            skew_1 =
              round((mean(waist) - median(waist)) / sd(waist), 2))
```

```
# A tibble: 4 x 6
  bmi_cat      count    mean     sd median skew_1
  <chr>       <int>   <dbl>   <dbl>  <dbl>   <dbl>
1 1_Underweight    41    55.2    7.58   54.5    0.09
2 2_Normal        917   61.2    9.35   59.5    0.19
3 3_Overweight    258   72.3   11.9    74    -0.14
4 4_Obese          294   85.6   17.1    86.8   -0.07
```

Or, we could use something like `favstats` from the `mosaic` package, which automatically accounts for missing data, and omits it when calculating summary statistics within each group.

```
mosaic::favstats(waist ~ bmi_cat, data = nnyfs)
```

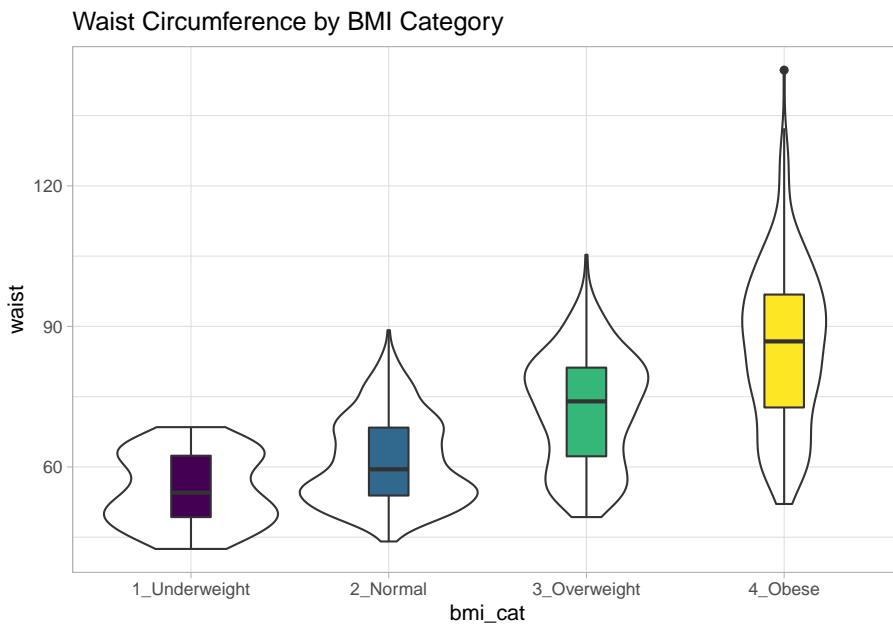
	bmi_cat	min	Q1	median	Q3	max	mean	sd	n	missing
1	1_Underweight	42.5	49.3	54.5	62.4	68.5	55.2	7.58	41	0
2	2_Normal	44.1	53.9	59.5	68.4	89.2	61.2	9.35	917	3
3	3_Overweight	49.3	62.3	74.0	81.2	105.3	72.3	11.94	258	0
4	4_Obese	52.1	72.7	86.8	96.8	144.7	85.6	17.11	294	1

While patients in the heavier groups generally had higher waist circumferences, the standard deviations suggest there may be some meaningful overlap. Let's draw the picture, in this case a comparison boxplot accompanying a violin plot.

```

nnyfs %>%
  filter(complete.cases(bmi_cat, waist)) %>%
  ggplot(., aes(x = bmi_cat, y = waist)) +
  geom_violin() +
  geom_boxplot(aes(fill = bmi_cat), width = 0.2) +
  theme_light() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Waist Circumference by BMI Category")

```



The data transformation with dplyr cheat sheet found under the Help menu in RStudio is a great resource. And, of course, for more details, visit Grolemund and Wickham (2019).

10.3 Boxplots to Relate an Outcome to a Categorical Predictor

Boxplots are much more useful when comparing samples of data. For instance, consider this comparison boxplot describing the triceps skinfold results across the four levels of BMI category.

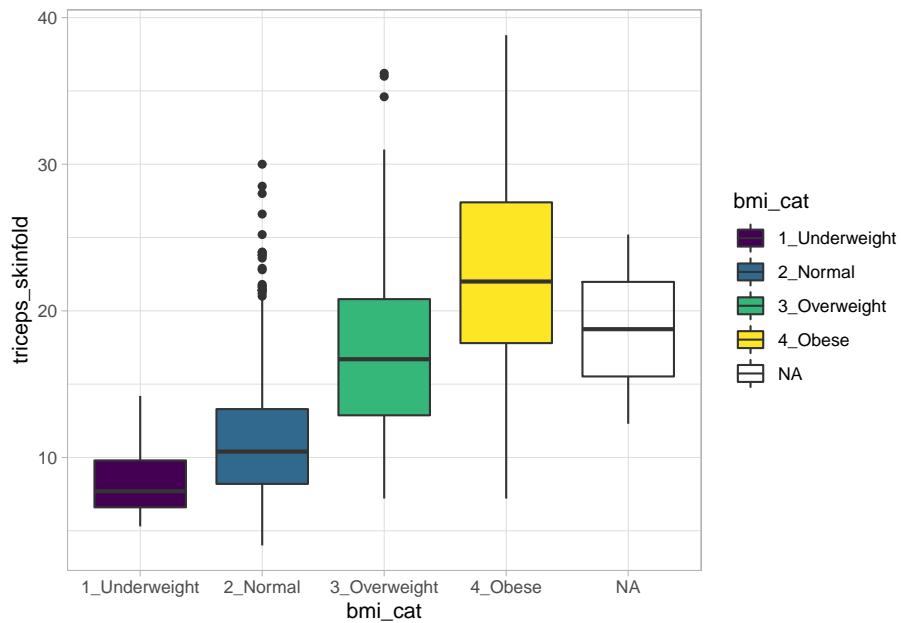
```

ggplot(nnyfs, aes(x = bmi_cat, y = triceps_skinfold,
                  fill = bmi_cat)) +

```

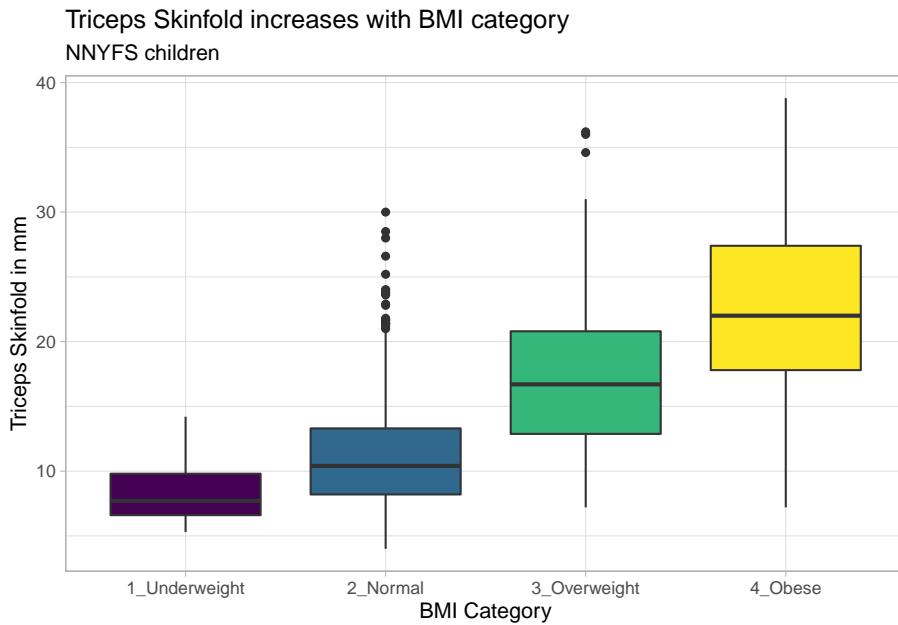
```
geom_boxplot() +
scale_fill_viridis_d() +
theme_light()
```

Warning: Removed 21 rows containing non-finite values (stat_boxplot).



Again, we probably want to omit those missing values (both in `bmi_cat` and `triceps_skinfold`) and also eliminate the repetitive legend (guides) on the right.

```
nnyfs %>%
  filter(complete.cases(bmi_cat, triceps_skinfold)) %>%
  ggplot(., aes(x = bmi_cat, y = triceps_skinfold,
                 fill = bmi_cat)) +
  geom_boxplot() +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  theme_light() +
  labs(x = "BMI Category", y = "Triceps Skinfold in mm",
       title = "Triceps Skinfold increases with BMI category",
       subtitle = "NNYFS children")
```

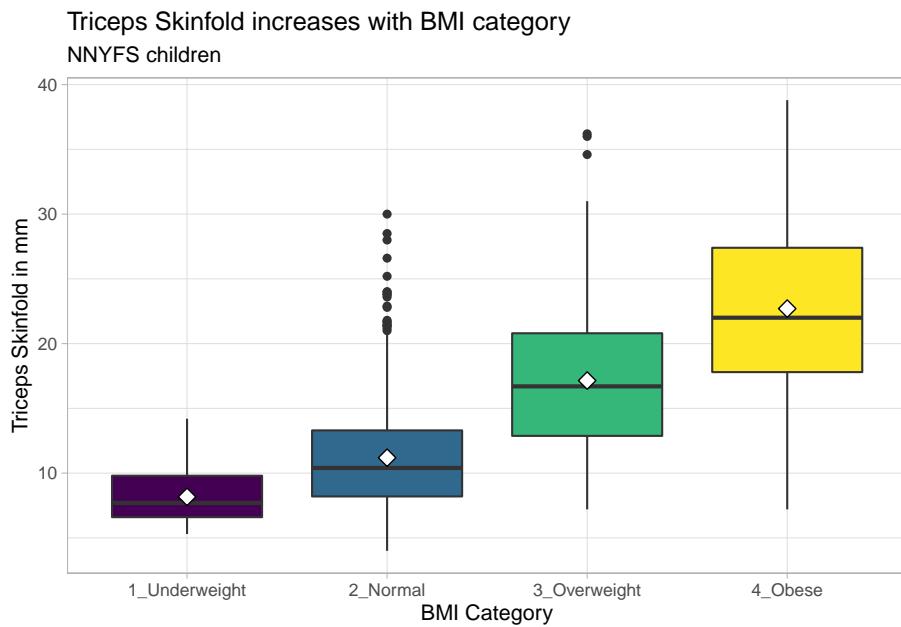


As always, the boxplot shows the five-number summary (minimum, 25th percentile, median, 75th percentile and maximum) in addition to highlighting candidate outliers.

10.3.1 Augmenting the Boxplot with the Sample Mean

Often, we want to augment such a plot, perhaps by adding a little diamond to show the **sample mean** within each category, so as to highlight skew (in terms of whether the mean is meaningfully different from the median.)

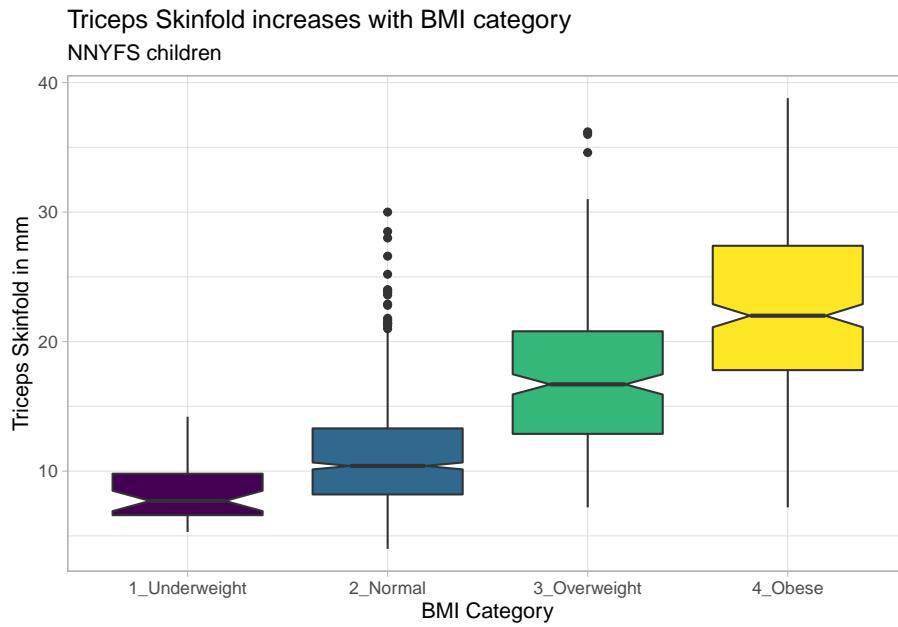
```
nnyfs %>%
  filter(complete.cases(bmi_cat, triceps_skinfold)) %>%
  ggplot(., aes(x = bmi_cat, y = triceps_skinfold,
                 fill = bmi_cat)) +
  geom_boxplot() +
  stat_summary(fun.y="mean", geom="point",
              shape=23, size=3, fill="white") +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  theme_light() +
  labs(x = "BMI Category", y = "Triceps Skinfold in mm",
       title = "Triceps Skinfold increases with BMI category",
       subtitle = "NNYFS children")
```



10.3.2 Adding Notches to a Boxplot

Notches are used in boxplots to help visually assess whether the medians of the distributions across the various groups actually differ to a statistically detectable extent. Think of them as confidence regions around the medians. If the notches do not overlap, as in this situation, this provides some evidence that the medians in the populations represented by these samples may be different.

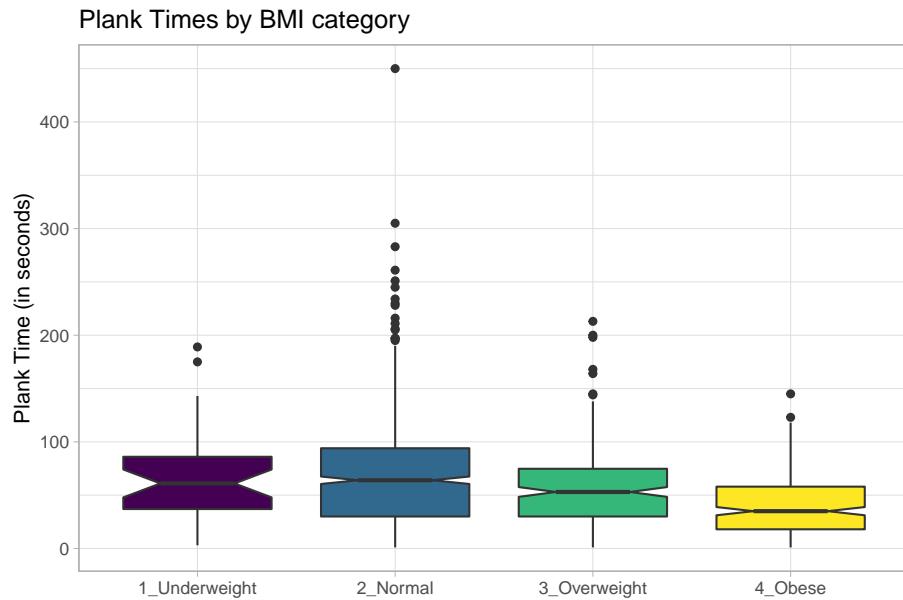
```
nnyfs %>%
  filter(complete.cases(bmi_cat, triceps_skinfold)) %>%
  ggplot(., aes(x = bmi_cat, y = triceps_skinfold,
                fill = bmi_cat)) +
  geom_boxplot(notch = TRUE) +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  theme_light() +
  labs(x = "BMI Category", y = "Triceps Skinfold in mm",
       title = "Triceps Skinfold increases with BMI category",
       subtitle = "NNYFS children")
```



There is no overlap between the notches for each of the four categories, so we might reasonably conclude that the true median triceps skinfold values across the four categories are statistically significantly different.

For an example where the notches do overlap, consider the comparison of plank times by BMI category.

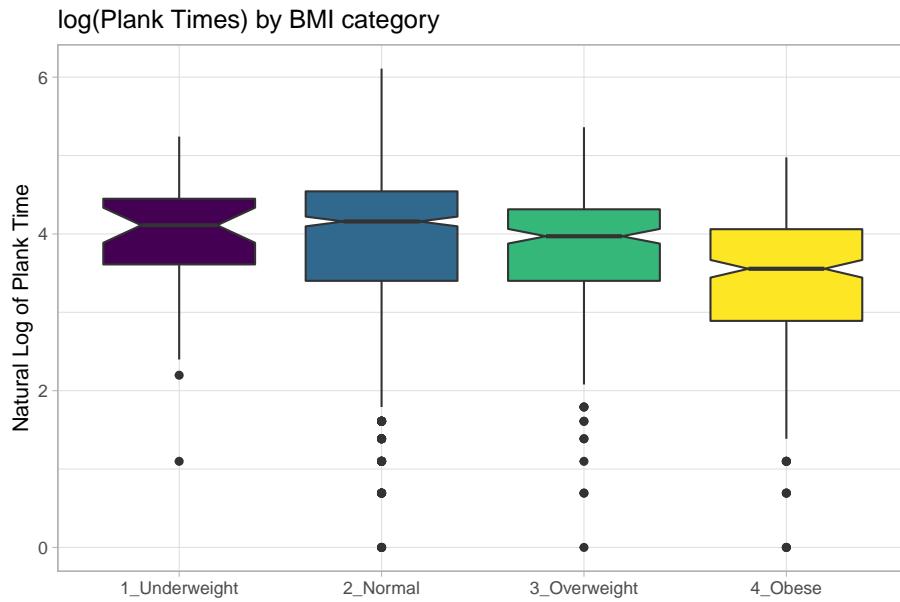
```
nnyfs %>%
  filter(complete.cases(bmi_cat, plank_time)) %>%
  ggplot(., aes(x=bmi_cat, y=plank_time, fill=bmi_cat)) +
  geom_boxplot(notch=TRUE) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  theme_light() +
  labs(title = "Plank Times by BMI category",
       x = "", y = "Plank Time (in seconds)")
```



The overlap in the notches (for instance between Underweight and Normal) suggests that the median plank times in the population of interest don't necessarily differ in a meaningful way by BMI category, other than perhaps the Obese group which may have a shorter time.

These data are somewhat right skewed. Would a logarithmic transformation in the plot help us see the patterns more clearly?

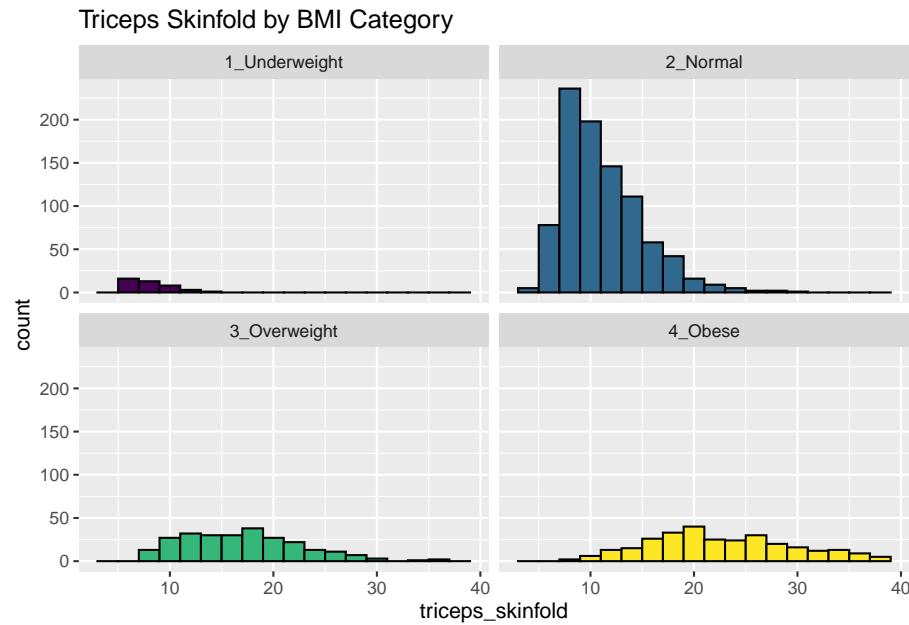
```
nnyfs %>%
  filter(complete.cases(bmi_cat, plank_time)) %>%
  ggplot(., aes(x=bmi_cat, y = log(plank_time), fill=bmi_cat)) +
  geom_boxplot(notch=TRUE) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  theme_light() +
  labs(title = "log(Plank Times) by BMI category",
       x = "", y = "Natural Log of Plank Time")
```



10.4 Using Multiple Histograms to Make Comparisons

We can make an array of histograms to describe multiple groups of data, using `ggplot2` and the notion of **faceting** our plot.

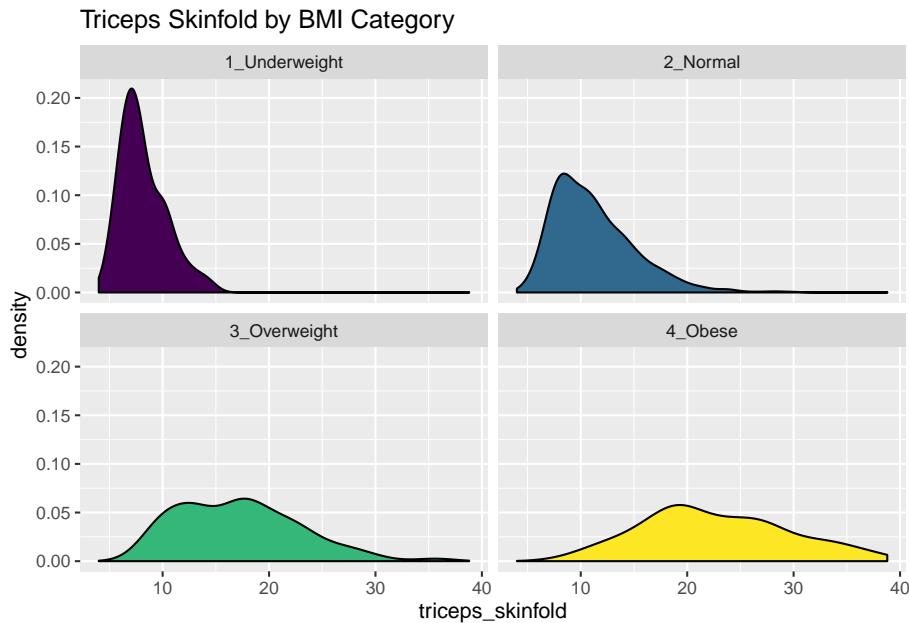
```
nnyfs %>%
  filter(complete.cases(triceps_skinfold, bmi_cat)) %>%
  ggplot(., aes(x=triceps_skinfold, fill = bmi_cat)) +
  geom_histogram(binwidth = 2, color = "black") +
  facet_wrap(~ bmi_cat) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  labs(title = "Triceps Skinfold by BMI Category")
```



10.5 Using Multiple Density Plots to Make Comparisons

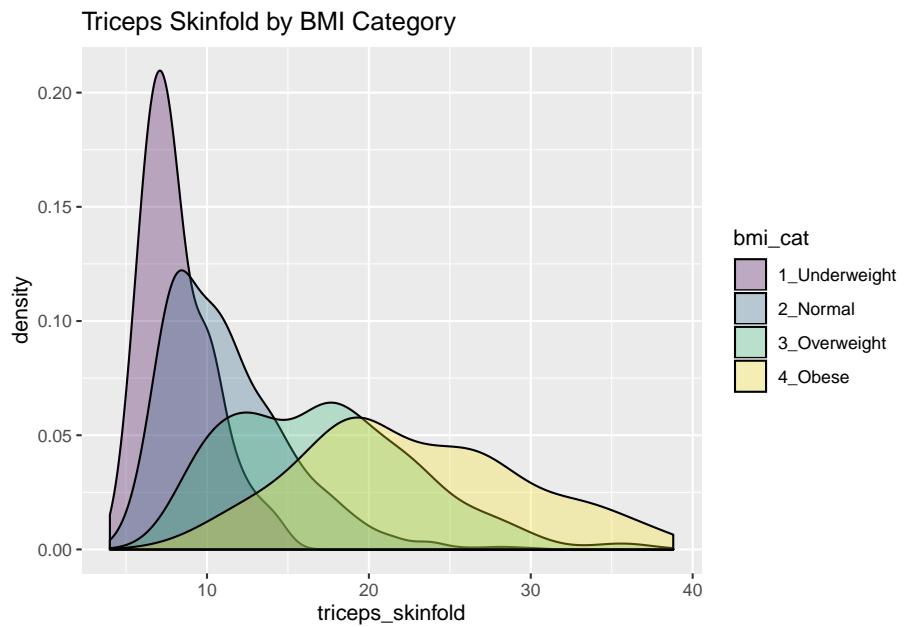
Or, we can make a series of density plots to describe multiple groups of data.

```
nyfs %>%
  filter(complete.cases(triceps_skinfold, bmi_cat)) %>%
  ggplot(., aes(x=triceps_skinfold, fill = bmi_cat)) +
  geom_density(color = "black") +
  facet_wrap(~ bmi_cat) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  labs(title = "Triceps Skinfold by BMI Category")
```



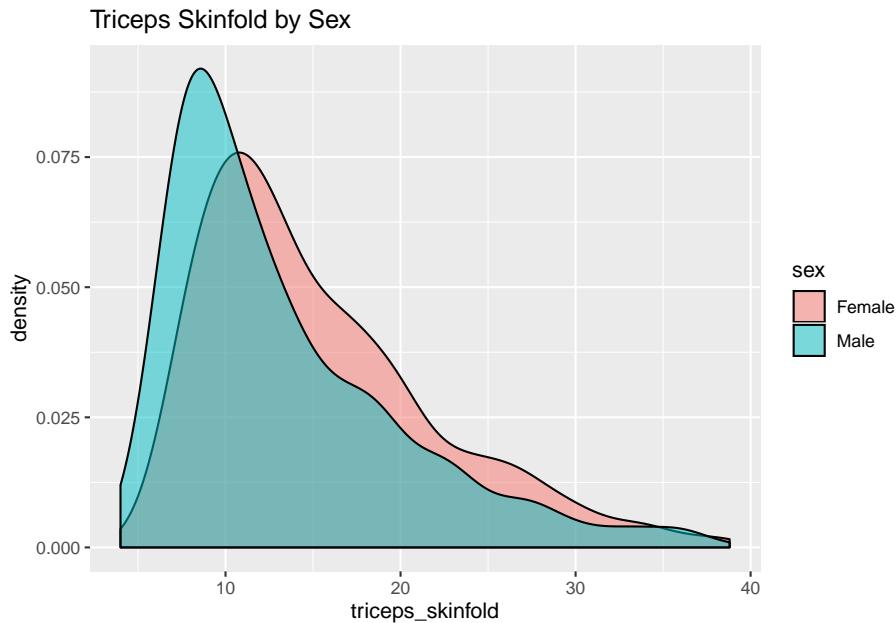
Or, we can plot all of the densities on top of each other with semi-transparent fills.

```
nyyfs %>%
  filter(complete.cases(triceps_skinfold, bmi_cat)) %>%
  ggplot(., aes(x=triceps_skinfold, fill = bmi_cat)) +
  geom_density(alpha=0.3) +
  scale_fill_viridis_d() +
  labs(title = "Triceps Skinfold by BMI Category")
```



This really works better when we are comparing only two groups, like females to males.

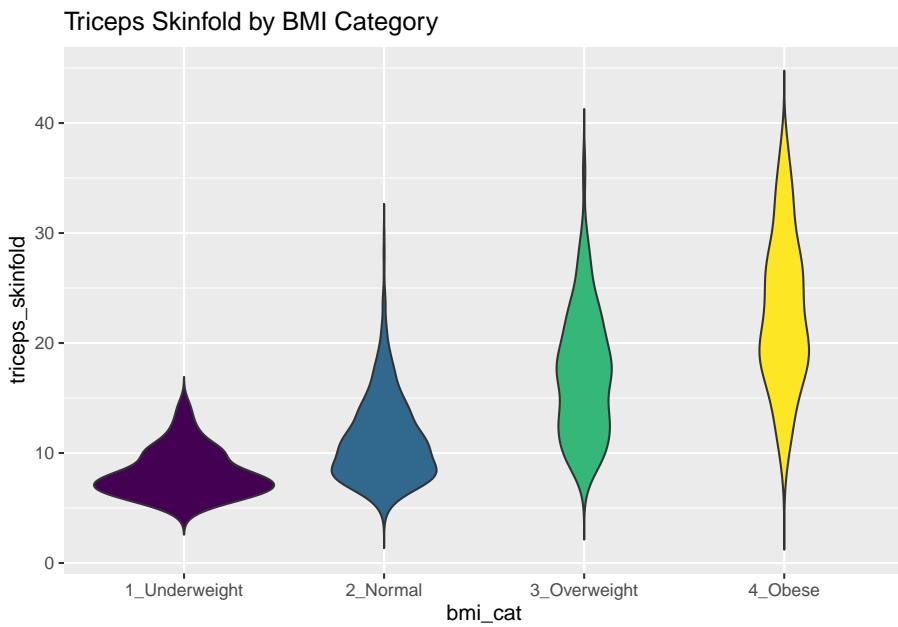
```
nyfs %>%
  filter(complete.cases(triceps_skinfold, sex)) %>%
  ggplot(., aes(x=triceps_skinfold, fill = sex)) +
  geom_density(alpha=0.5) +
  labs(title = "Triceps Skinfold by Sex")
```



10.6 Building a Violin Plot

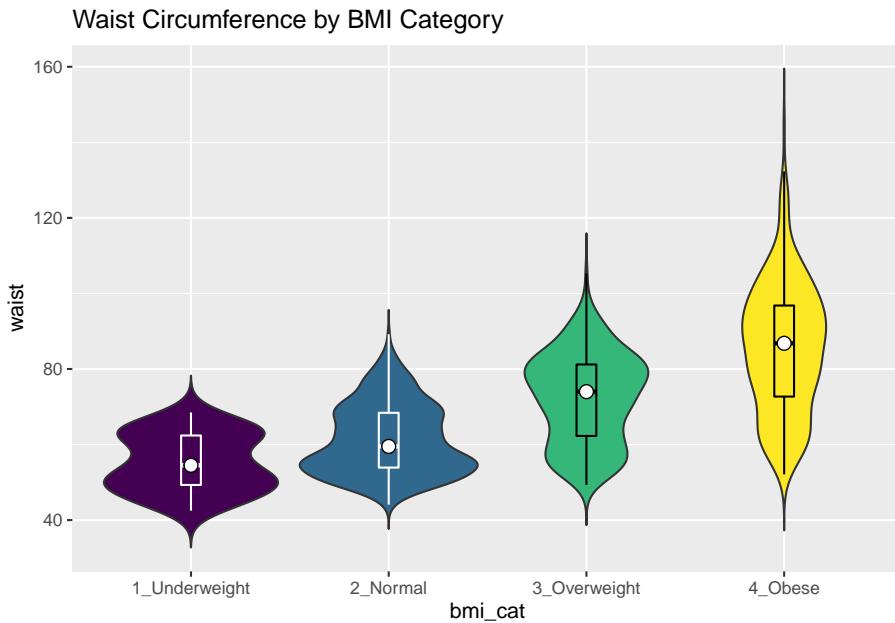
There are a number of other plots which compare distributions of data sets. An interesting one is called a **violin plot**. A violin plot is a kernel density estimate, mirrored to form a symmetrical shape.

```
nnyfs %>%
  filter(complete.cases(triceps_skinfold, bmi_cat)) %>%
  ggplot(., aes(x=bmi_cat, y=triceps_skinfold,
                 fill = bmi_cat)) +
  geom_violin(trim=FALSE) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  labs(title = "Triceps Skinfold by BMI Category")
```



Traditionally, these plots are shown with overlaid boxplots and a white dot at the median, like this example, now looking at waist circumference again.

```
nnyfs %>%
  filter(complete.cases(waist, bmi_cat)) %>%
  ggplot(., aes(x = bmi_cat, y = waist,
                fill = bmi_cat)) +
  geom_violin(trim=FALSE) +
  geom_boxplot(width=.1, outlier.colour=NA,
               color = c(rep("white",2), rep("black",2))) +
  stat_summary(fun.y=median, geom="point",
               fill="white", shape=21, size=3) +
  scale_fill_viridis_d() +
  guides(fill = "none") +
  labs(title = "Waist Circumference by BMI Category")
```



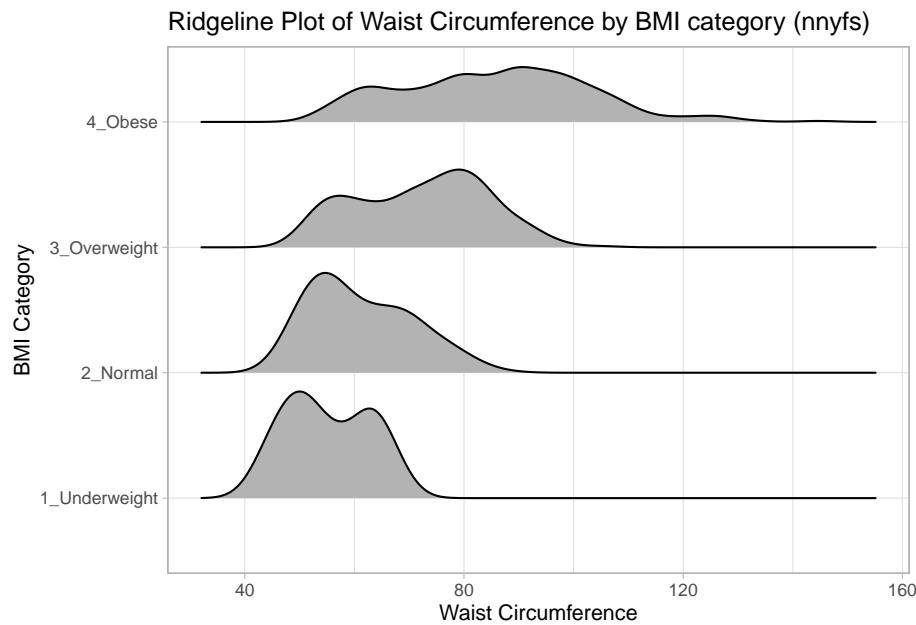
10.7 A Ridgeline Plot

Some people don't like violin plots - for example, see <https://simplystatistics.org/2017/07/13/the-joy-of-no-more-violin-plots/>. A relatively new alternative plot is available. This shows the distribution of several groups simultaneously, especially when you have lots of subgroup categories, and is called a **ridgeline plot**¹.

```
nnyfs %>%
  filter(complete.cases(waist, bmi_cat)) %>%
  ggplot(., aes(x = waist, y = bmi_cat, height = ..density..)) +
  ggridges::geom_density_ridges(scale = 0.85) +
  theme_light() +
  labs(title = "Ridgeline Plot of Waist Circumference by BMI category (nnyfs)",
       x = "Waist Circumference", y = "BMI Category")
```

Picking joint bandwidth of 3.47

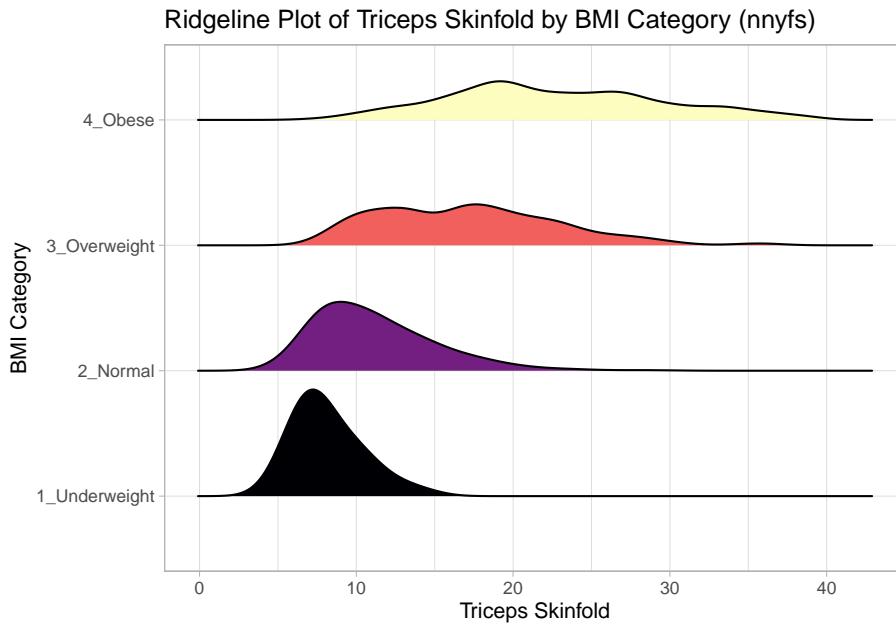
¹These were originally called joy plots, and the tools were contained in the `ggjoy` package but that name and package has been deprecated in favor of `ggridges`.



And here's a ridgeline plot for the triceps skinfolds. We'll start by sorting the subgroups by the median value of our outcome (triceps skinfold) in this case, though it turns out not to matter. We'll also add some color.

```
nnyfs %>%
  filter(complete.cases(bmi_cat, triceps_skinfold)) %>%
  mutate(bmi_cat = fct_reorder(bmi_cat,
                               triceps_skinfold,
                               .fun = median)) %>%
  ggplot(., aes(x = triceps_skinfold, y = bmi_cat,
                fill = bmi_cat, height = ..density..)) +
  ggridges::geom_density_ridges(scale = 0.85) +
  scale_fill_viridis_d(option = "magma") +
  guides(fill = FALSE) +
  labs(title = "Ridgeline Plot of Triceps Skinfold by BMI Category (nnyfs)",
       x = "Triceps Skinfold", y = "BMI Category") +
  theme_light()
```

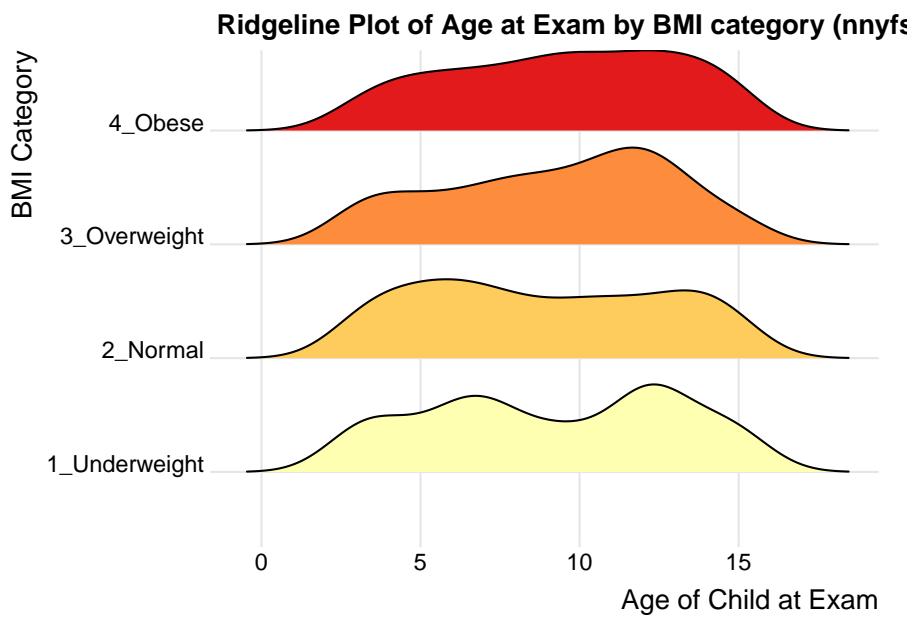
Picking joint bandwidth of 1.37



For one last example, we'll look at age by BMI category, so that sorting the BMI subgroups by the median matters, and we'll try an alternate color scheme, and a theme specially designed for the ridgeline plot.

```
nnyfs %>%
  filter(complete.cases(bmi_cat, age_child)) %>%
  mutate(bmi_cat = reorder(bmi_cat, age_child, median)) %>%
  ggplot(aes(x = age_child, y = bmi_cat, fill = bmi_cat, height = ..density..)) +
  ggridges::geom_density_ridges(scale = 0.85) +
  scale_fill_brewer(palette = "YlOrRd") +
  guides(fill = FALSE) +
  labs(title = "Ridgeline Plot of Age at Exam by BMI category (nnyfs)",
       x = "Age of Child at Exam", y = "BMI Category") +
  ggridges::theme_ridges()
```

Picking joint bandwidth of 1.15



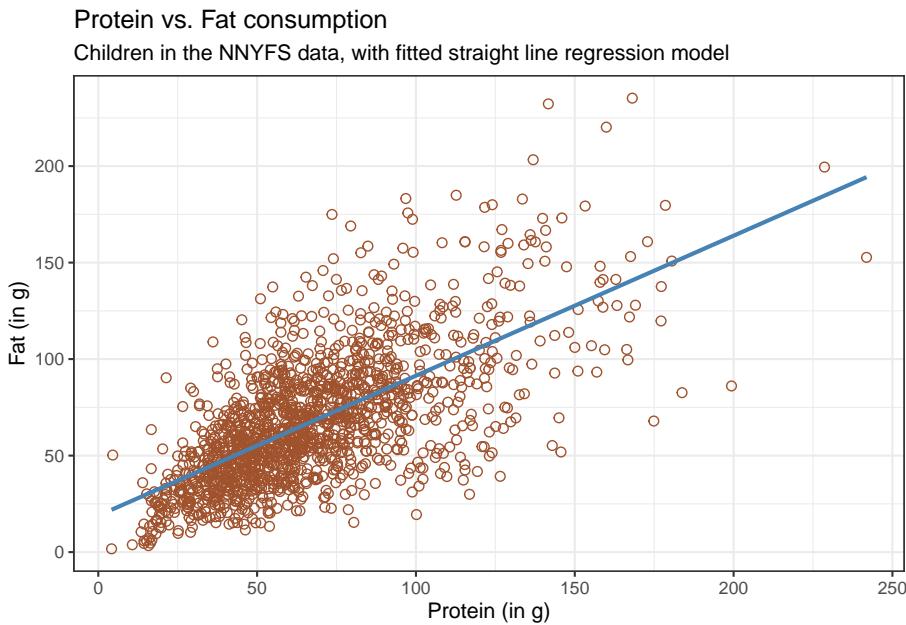
Chapter 11

Straight Line Models and Correlation

11.1 Assessing A Scatterplot

Let's consider the relationship of `protein` and `fat` consumption for children in the `nnyfs` data. We'll begin our investigation, as we always should, by drawing a relevant picture. For the association of two quantitative variables, a `scatterplot` is usually the right start. Each subject in the `nnyfs` data is represented by one of the points below. To the plot, I've also used `geom_smooth` to add a straight line regression model, which we'll discuss later.

```
ggplot(data = nnyfs, aes(x = protein, y = fat)) +
  geom_point(shape = 1, size = 2, col = "sienna") +
  geom_smooth(method = "lm", se = FALSE, col = "steelblue") +
  theme_bw() +
  labs(title = "Protein vs. Fat consumption",
       subtitle = "Children in the NNYFS data, with fitted straight line regression model",
       x = "Protein (in g)", y = "Fat (in g)")
```



Here, I've arbitrarily decided to place `fat` on the vertical axis, and `protein` on the horizontal. Fitting a prediction model to this scatterplot will then require that we predict `fat` on the basis of `protein`.

In this case, the pattern appears to be:

1. **direct**, or positive, in that the values of the x variable (`protein`) increase, so do the values of the y variable (`fat`). Essentially, it appears that subjects who consumed more protein also consumed more fat, but we don't know cause and effect here.
2. fairly **linear** in that most of the points cluster around what appears to be a pattern which is well-fitted by a straight line.
3. moderately **strong** in that the range of values for `fat` associated with any particular value of `protein` is fairly tight. If we know someone's protein consumption, that should meaningfully improve our ability to predict their fat consumption, among the subjects in these data.
4. that we see some unusual or **outlier** values, further away from the general pattern of most subjects shown in the data.

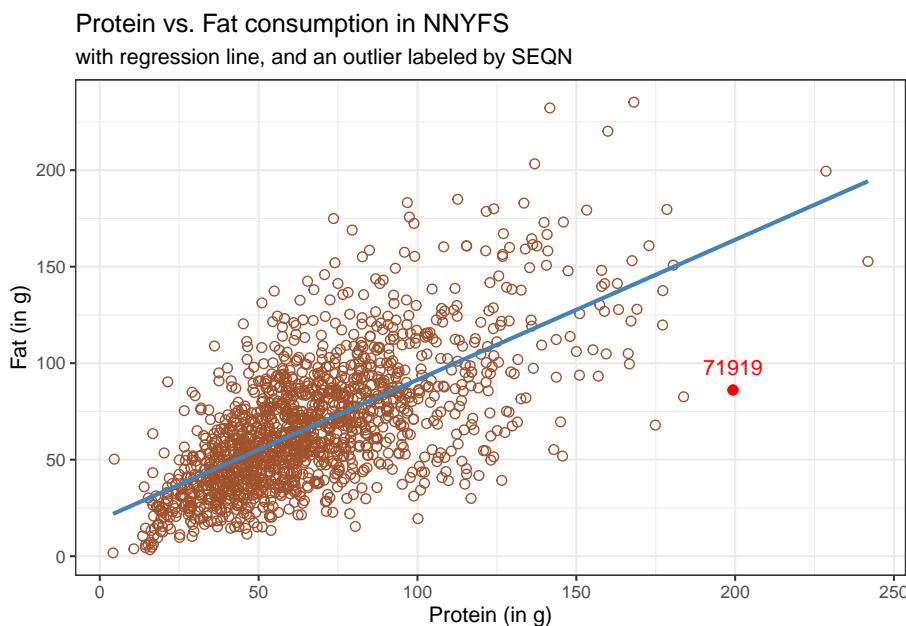
11.1.1 Highlighting an unusual point

Consider the subject with protein consumption close to 200 g, whose fat consumption is below 100 g. That's well below the prediction of the linear model for example. We can identify the subject because it is the only person with `protein` > 190 and `fat` < 100 with `BMI` > 35 and `waist.circ` < 70. So I'll create a subset

of the `nnyfs` data containing the point that meets that standard, and then add a red point and a label to the plot.

```
# identify outlier and place it in data frame s1
s1 <- filter(nnyfs, protein > 190 & fat < 100)

ggplot(data = nnyfs, aes(x = protein, y = fat)) +
  geom_point(shape = 1, size = 2, col = "sienna") +
  geom_smooth(method = "lm", se = FALSE, col = "steelblue") +
  theme_bw() +
  geom_point(data = s1, size = 2, col = "red") +
  geom_text(data = s1, label = s1$SEQN,
            vjust = -1, col = "red") +
  labs(title = "Protein vs. Fat consumption in NNYFS",
       subtitle = "with regression line, and an outlier labeled by SEQN",
       x = "Protein (in g)", y = "Fat (in g)")
```



While this subject is hardly the only unusual point in the data set, it is one of the more unusual ones, in terms of its vertical distance from the regression line. We can identify the subject by printing (part of) the tibble we created.

```
s1 %>% select(SEQN, sex, race_eth, age_child, protein, fat) %>% knitr::kable()
```

SEQN	sex	race_eth	age_child	protein	fat
71919	Female	2_White Non-Hispanic	14	199	86.1

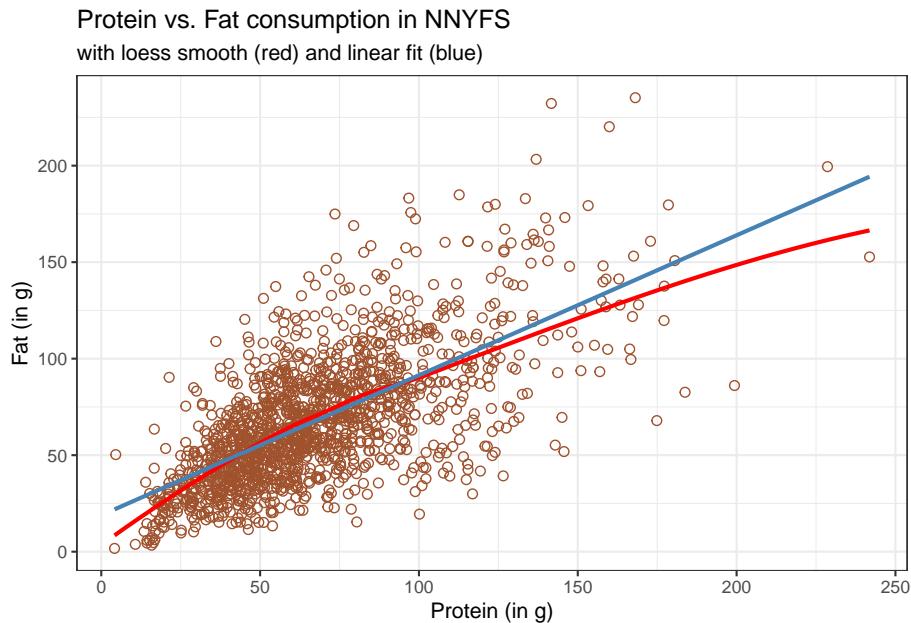
Now, does it seem to you like a straight line model will describe this protein-fat

relationship well?

11.1.2 Adding a Scatterplot Smooth using loess

Next, we'll use the **loess** procedure to fit a smooth curve to the data, which attempts to capture the general pattern.

```
ggplot(data = nnyfs, aes(x = protein, y = fat)) +
  geom_point(shape = 1, size = 2, col = "sienna") +
  geom_smooth(method = "loess", se = FALSE, col = "red") +
  geom_smooth(method = "lm", se = FALSE, col = "steelblue") +
  theme_bw() +
  labs(title = "Protein vs. Fat consumption in NNYFS",
       subtitle = "with loess smooth (red) and linear fit (blue)",
       x = "Protein (in g)", y = "Fat (in g)")
```



This “loess” smooth curve is fairly close to the straight line fit, indicating that perhaps a linear regression model might fit the data well.

A **loess smooth** is a method of fitting a local polynomial regression model that R uses as its default smooth for scatterplots with fewer than 1000 observations. Think of the loess as a way of fitting a curve to data by tracking (at point x) the points within a neighborhood of point x , with more emphasis given to points near x . It can be adjusted by tweaking two specific parameters, in particular:

- a **span** parameter (defaults to 0.75) which is also called α in the litera-

ture, that controls the degree of smoothing (essentially, how larger the neighborhood should be), and

- a `degree` parameter (defaults to 2) which specifies the degree of polynomial to be used. Normally, this is either 1 or 2 - more complex functions are rarely needed for simple scatterplot smoothing.

In addition to the curve, smoothing procedures can also provide confidence intervals around their main fitted line. Consider the following plot, which adjusts the span and also adds in the confidence intervals.

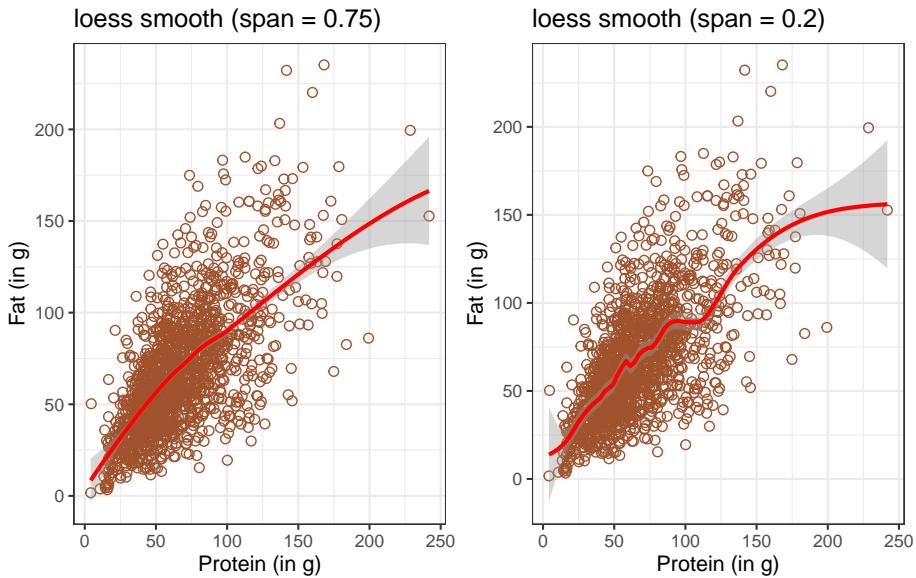
```
library(patchwork)

p1 <- ggplot(data = nnyfs, aes(x = protein, y = fat)) +
  geom_point(shape = 1, size = 2, col = "sienna") +
  geom_smooth(method = "loess", span = 0.75, se = TRUE,
              col = "red") +
  theme_bw() +
  labs(title = "loess smooth (span = 0.75)",
       x = "Protein (in g)", y = "Fat (in g)")

p2 <- ggplot(data = nnyfs, aes(x = protein, y = fat)) +
  geom_point(shape = 1, size = 2, col = "sienna") +
  geom_smooth(method = "loess", span = 0.2, se = TRUE,
              col = "red") +
  theme_bw() +
  labs(title = "loess smooth (span = 0.2)",
       x = "Protein (in g)", y = "Fat (in g)")

p1 + p2 +
  plot_annotation(title = "Impact of adjusting loess smooth span, in NNYFS data")
```

Impact of adjusting loess smooth span, in NNYFS data



By reducing the size of the span, the plot on the right shows a somewhat less “smooth” function than the plot on the left.

11.1.3 What Line Does R Fit?

Returning to the linear regression model, how can we, mathematically, characterize that line? As with any straight line, our model equation requires us to specify two parameters: a slope and an intercept (sometimes called the y-intercept.)

To identify the equation R used to fit this line (using the method of least squares), we use the `lm` command

```
lm(fat ~ protein, data = nnyfs)
```

```
Call:  
lm(formula = fat ~ protein, data = nnyfs)
```

```
Coefficients:  
(Intercept) protein  
18.895      0.725
```

So the fitted line is specified as

$$\text{fat} = 18.8945 + 0.7251 \text{ protein}$$

A detailed summary of the fitted linear regression model is also available.

```
summary(lm(fat ~ protein, data = nnyfs))
```

```
Call:
lm(formula = fat ~ protein, data = nnyfs)

Residuals:
    Min      1Q  Median      3Q     Max 
-77.80 -14.84  -2.45  13.60 110.60 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 18.8945   1.5330   12.3   <2e-16 ***  
protein      0.7251   0.0208   34.9   <2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.1 on 1516 degrees of freedom
Multiple R-squared:  0.445, Adjusted R-squared:  0.445 
F-statistic: 1.22e+03 on 1 and 1516 DF,  p-value: <2e-16
```

Another way we'll frequently summarize a model's coefficient is to use the `broom` package's `tidy` function to put the coefficient estimates into a tibble.

```
broom::tidy(lm(fat ~ protein, data = nnyfs),
            conf.int = TRUE, conf.level = 0.95) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	18.895	1.533	12.3	0	15.887	21.902
protein	0.725	0.021	34.9	0	0.684	0.766

We'll spend a lot of time working with these regression summaries, especially in Part C of the course.

For now, it will suffice to understand the following:

- The outcome variable in this model is `fat`, and the predictor variable is `protein`.
- The straight line model for these data fitted by least squares is $\text{fat} = 18.9 + 0.725 \text{ protein}$
- The slope of `protein` is positive, which indicates that as `protein` increases, we expect that `fat` will also increase. Specifically, we expect that for every additional gram of protein consumed, the fat consumption will be 0.725 gram larger.
- The multiple R-squared (squared correlation coefficient) is 0.445, which implies that 44.5% of the variation in `fat` is explained using this linear

model with `protein`.

- This also implies that the Pearson correlation between `fat` and `protein` is the square root of 0.445, or 0.667. More on the Pearson correlation soon.

So, if we plan to use a simple (least squares) linear regression model to describe fat consumption as a function of protein consumption in the NNYFS data, does it look like a least squares (or linear regression) model will be an effective choice?

11.2 Correlation Coefficients

Two different correlation measures are worth our immediate attention.

- The one most often used is called the *Pearson* correlation coefficient, and is symbolized with the letter `r` or sometimes the Greek letter rho (ρ).
- Another tool is the Spearman rank correlation coefficient, also occasionally symbolized by ρ .

For the `nnyfs` data, the Pearson correlation of `fat` and `protein` can be found using the `cor()` function.

```
nnyfs %$% cor(fat, protein)
```

```
[1] 0.667
```

Note that the correlation of any variable with itself is 1, and that the correlation of `fat` with `protein` is the same regardless of whether you enter `fat` first or `protein` first.

11.3 The Pearson Correlation Coefficient

Suppose we have n observations on two variables, called X and Y. The Pearson correlation coefficient assesses how well the relationship between X and Y can be described using a linear function.

- The Pearson correlation is **dimension-free**.
- It falls between -1 and +1, with the extremes corresponding to situations where all the points in a scatterplot fall exactly on a straight line with negative and positive slopes, respectively.
- A Pearson correlation of zero corresponds to the situation where there is no linear association.
- Unlike the estimated slope in a regression line, the sample correlation coefficient is symmetric in X and Y, so it does not depend on labeling one of them (Y) the response variable, and one of them (X) the predictor.

Suppose we have n observations on two variables, called X and Y , where \bar{X} is the sample mean of X and s_x is the standard deviation of X . The **Pearson** correlation coefficient r_{XY} is:

$$r_{XY} = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

11.4 A simulated example

The `correx1` data file contains six different sets of (x,y) points, identified by the `set` variable.

```
correx1 <- read.csv("data/correx1.csv") %>%tbl_df
```

```
summary(correx1)
```

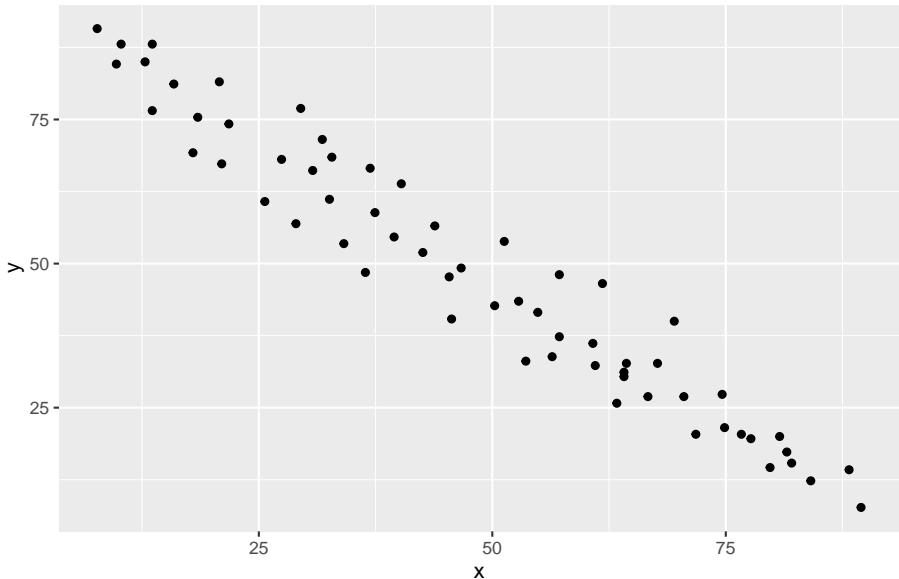
	set	x	y
Alex	:62	Min. : 5.9	Min. : 7.3
Bonnie	:37	1st Qu.:29.5	1st Qu.:30.4
Colin	:36	Median :46.2	Median :46.9
Danielle	:70	Mean :46.5	Mean :49.1
Earl	:15	3rd Qu.:63.3	3rd Qu.:68.1
Fiona	:57	Max. :98.2	Max. :95.4

11.4.1 Data Set Alex

Let's start by working with the **Alex** data set.

```
ggplot(filter(corrrex1, set == "Alex"), aes(x = x, y = y)) +
  geom_point() +
  labs(title = "correx1: Data Set Alex")
```

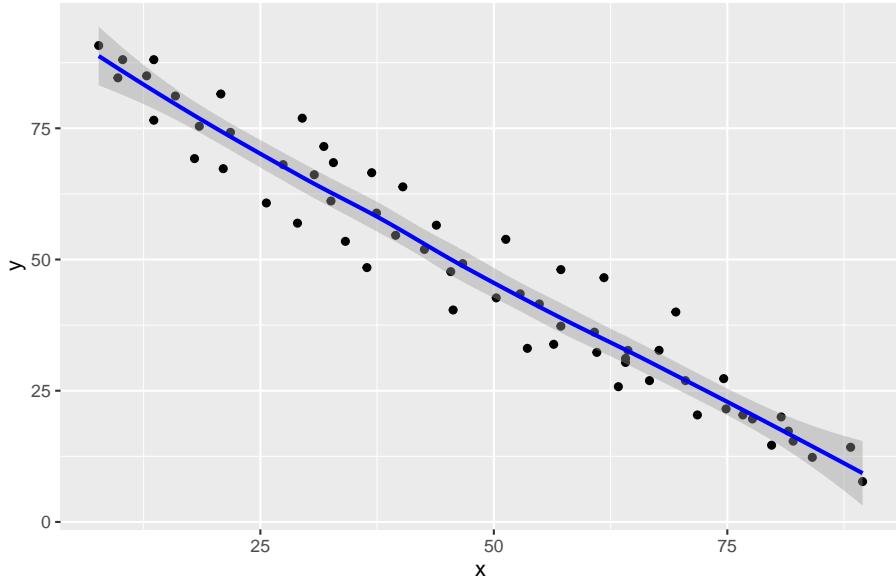
correx1: Data Set Alex



```
ggplot(filter(correx1, set == "Alex"), aes(x = x, y = y)) +  
  geom_point() +  
  geom_smooth(col = "blue") +  
  labs(title = "correx1: Alex, with loess smooth")
```

`geom_smooth()` using method = 'loess' and formula 'y ~ x'

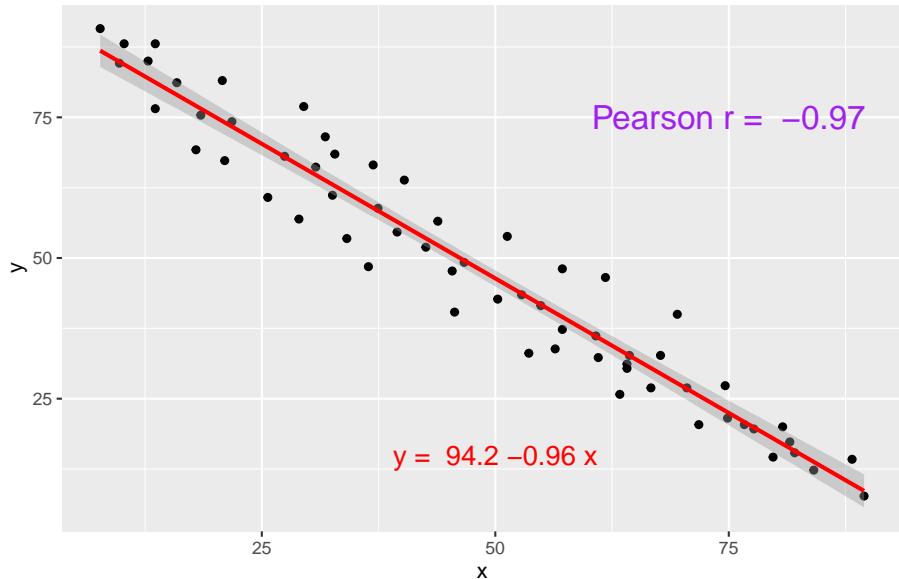
correx1: Alex, with loess smooth



```
setA <- filter(correx1, set == "Alex")

ggplot(setA, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "correx1: Alex, with Fitted Linear Model") +
  annotate("text", x = 75, y = 75, col = "purple", size = 6,
          label = paste("Pearson r = ", signif(cor(setA$x, setA$y),3))) +
  annotate("text", x = 50, y = 15, col = "red", size = 5,
          label = paste("y = ", signif(coef(lm(setA$y ~ setA$x))[1],3),
                        signif(coef(lm(setA$y ~ setA$x))[2],2), "x"))
```

correx1: Alex, with Fitted Linear Model

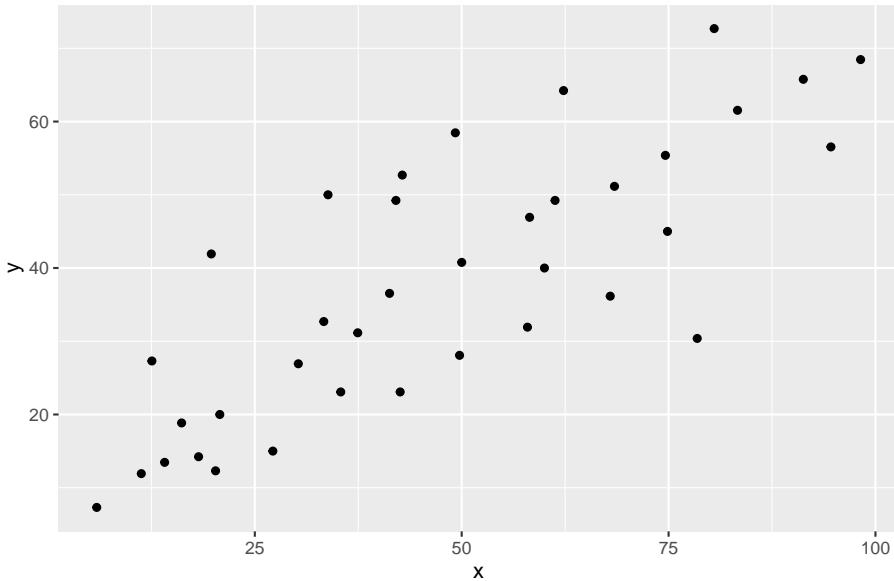


11.4.2 Data Set Bonnie

```
setB <- dplyr::filter(correx1, set == "Bonnie")

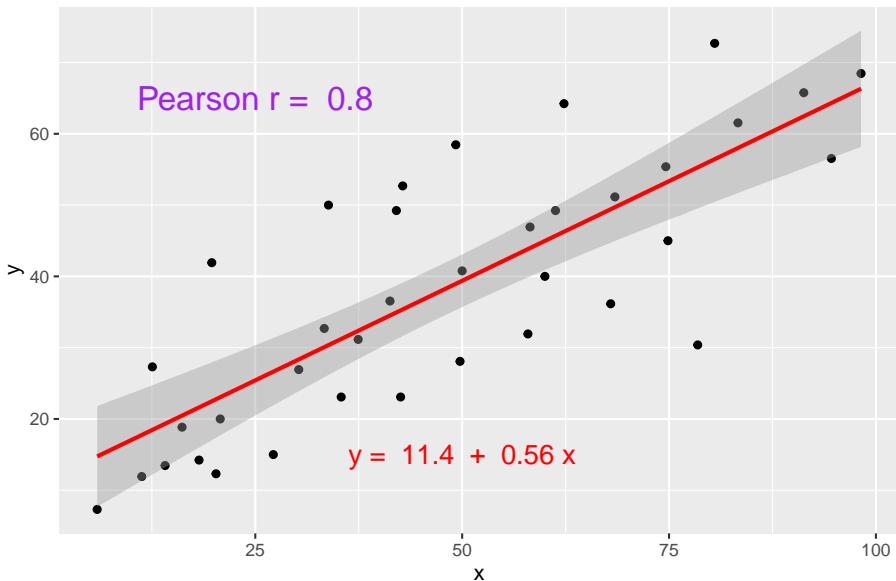
ggplot(setB, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "correx1: Data Set Bonnie")
```

correx1: Data Set Bonnie



```
ggplot(setB, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "correx1: Bonnie, with Fitted Linear Model") +
  annotate("text", x = 25, y = 65, col = "purple", size = 6,
          label = paste("Pearson r = ", signif(cor(setB$x, setB$y), 2))) +
  annotate("text", x = 50, y = 15, col = "red", size = 5,
          label = paste("y = ", signif(coef(lm(setB$y ~ setB$x))[1], 3),
                        " + ",
                        signif(coef(lm(setB$y ~ setB$x))[2], 2), "x"))
```

correx1: Bonnie, with Fitted Linear Model



11.4.3 Correlations for All Six Data Sets in the Correx1 Example

Let's look at the Pearson correlations associated with each of the six data sets contained in the `correx1` example.

```
tab1 <- correx1 %>%
  group_by(set) %>%
  summarise("Pearson r" = round(cor(x, y, use="complete"), 2))

knitr::kable(tab1)
```

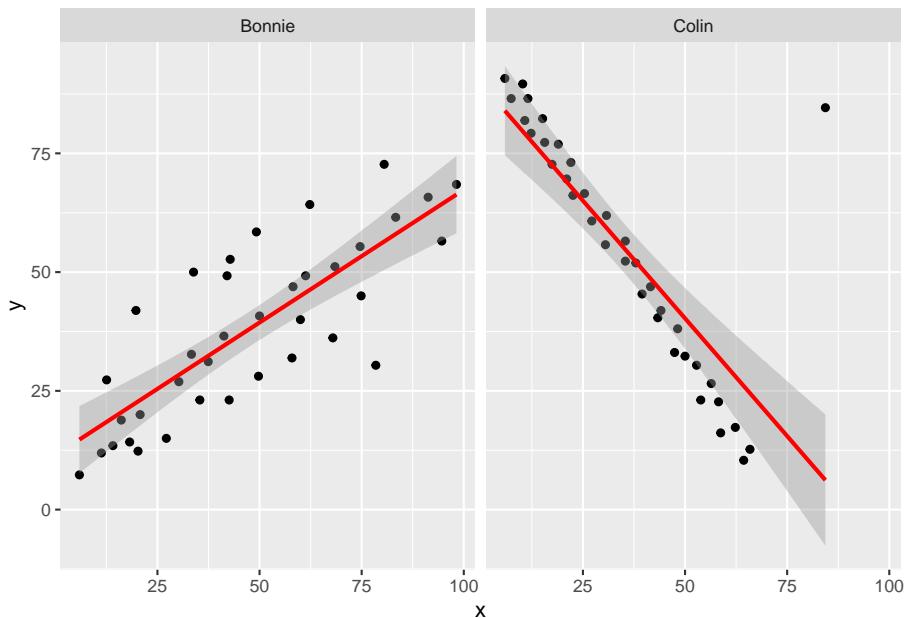
set	Pearson r
Alex	-0.97
Bonnie	0.80
Colin	-0.80
Danielle	0.00
Earl	-0.01
Fiona	0.00

11.4.4 Data Set Colin

It looks like the picture for Colin should be very similar (in terms of scatter) to the picture for Bonnie, except that Colin will have a negative slope, rather than

the positive one Bonnie has. Is that how this plays out?

```
setBC <- filter(correx1, set == "Bonnie" | set == "Colin")  
  
ggplot(setBC, aes(x = x, y = y)) +  
  geom_point() +  
  geom_smooth(method = "lm", col = "red") +  
  facet_wrap(~ set)
```

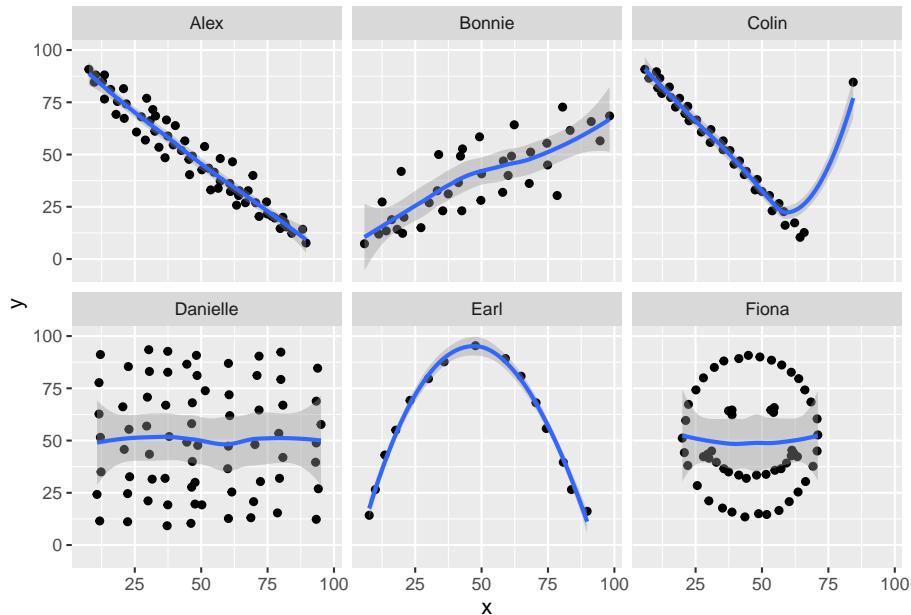


Uh, oh. It looks like the point in Colin at the top right is twisting what would otherwise be a very straight regression model with an extremely strong negative correlation. There's no better way to look for outliers than to examine the scatterplot.

11.4.5 Draw the Picture!

We've seen that Danielle, Earl and Fiona all show Pearson correlations of essentially zero. However, the three data sets look very different in a scatterplot.

```
ggplot(correx1, aes(x = x, y = y)) +  
  geom_point() +  
  geom_smooth(method = "loess") +  
  facet_wrap(~ set)
```



When we learn that the correlation is zero, we tend to assume we have a picture like the Danielle data set. If Danielle were our real data, we might well think that x would be of little use in predicting y .

- But what if our data looked like Earl? In the Earl data set, x is incredibly helpful in predicting y , but we can't use a straight line model - instead, we need a non-linear modeling approach.
- You'll recall that the Fiona data set also had a Pearson correlation of zero. But here, the picture is rather more interesting.

So, remember, draw the appropriate scatterplot whenever you make use of a correlation coefficient.

```
rm(setA, setB, setBC, tab1)
```

11.5 Estimating Correlation from Scatterplots

The correx2 data set is designed to help you calibrate yourself a bit in terms of estimating a correlation from a scatterplot. There are 11 data sets buried within the correx2 example, and they are labeled by their Pearson correlation coefficients, ranging from $r = 0.01$ to $r = 0.999$

```
correx2 <- read.csv("data/correx2.csv") %>%tbl_df  
correx2 %>%
```

```

group_by(set) %>%
summarise(cor = round(cor(x, y, use="complete"),3))

# A tibble: 11 x 2
  set      cor
  <fct>  <dbl>
1 Set 01  0.01
2 Set 10  0.102
3 Set 20  0.202
4 Set 30  0.301
5 Set 40  0.403
6 Set 50  0.499
7 Set 60  0.603
8 Set 70  0.702
9 Set 80  0.799
10 Set 90  0.902
11 Set 999 0.999

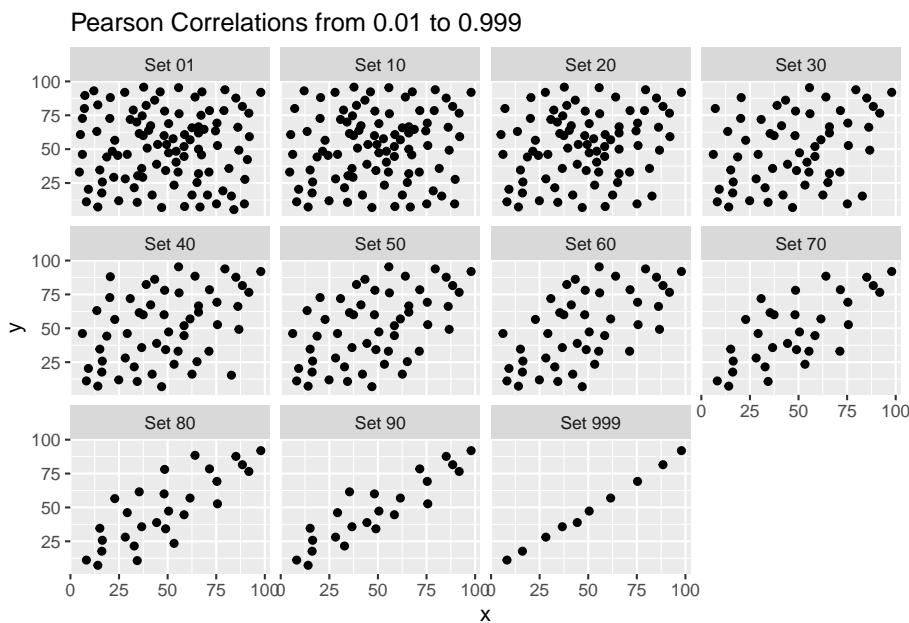
```

Here is a plot of the 11 data sets, showing the increase in correlation from 0.01 (in Set 01) to 0.999 (in Set 999).

```

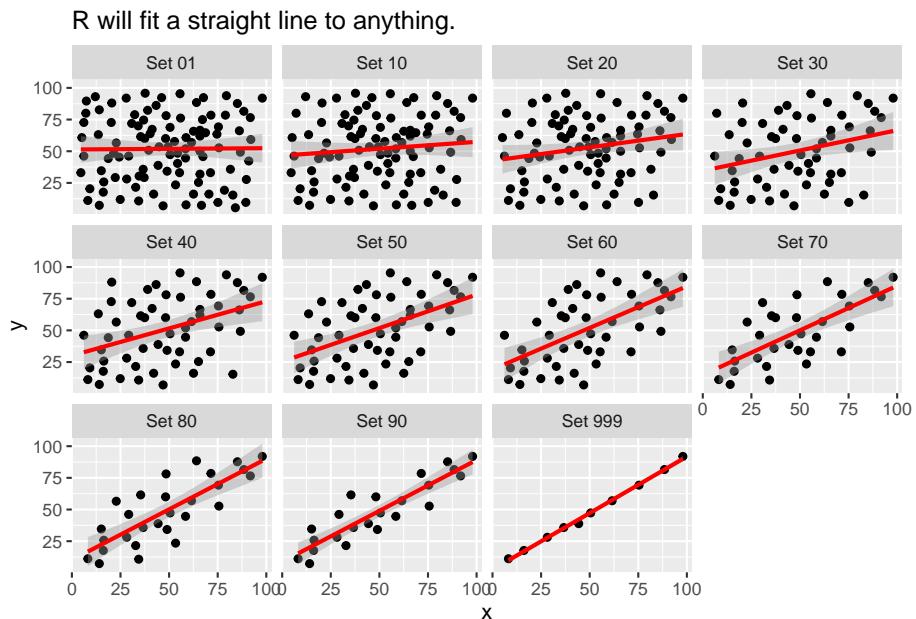
ggplot(correx2, aes(x = x, y = y)) +
  geom_point() +
  facet_wrap(~ set) +
  labs(title = "Pearson Correlations from 0.01 to 0.999")

```



Note that R will allow you to fit a straight line model to any of these relationships, no matter how appropriate it might be to do so.

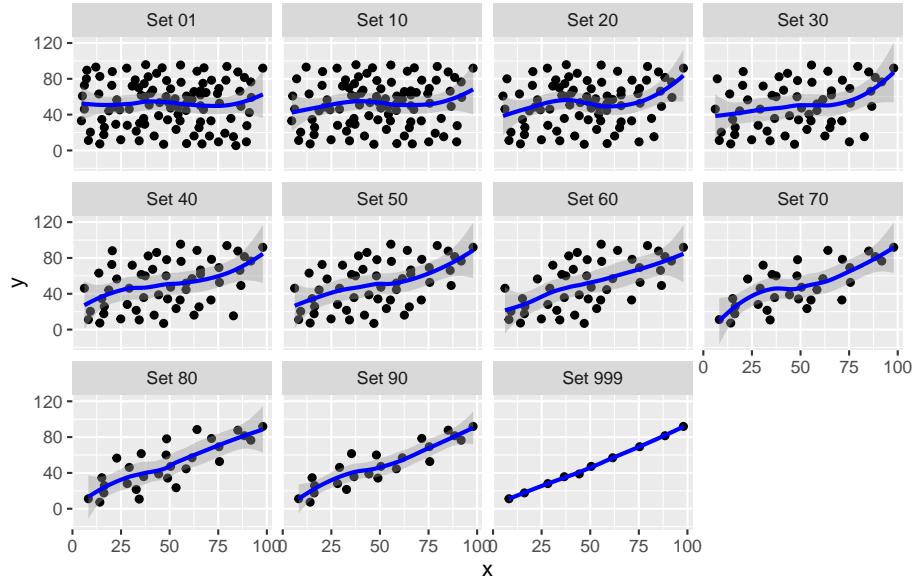
```
ggplot(correx2, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  facet_wrap(~ set) +
  labs(title = "R will fit a straight line to anything.")
```



```
ggplot(correx2, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(col = "blue") +
  facet_wrap(~ set) +
  labs(title = "Even if a loess smooth suggests non-linearity.")
```

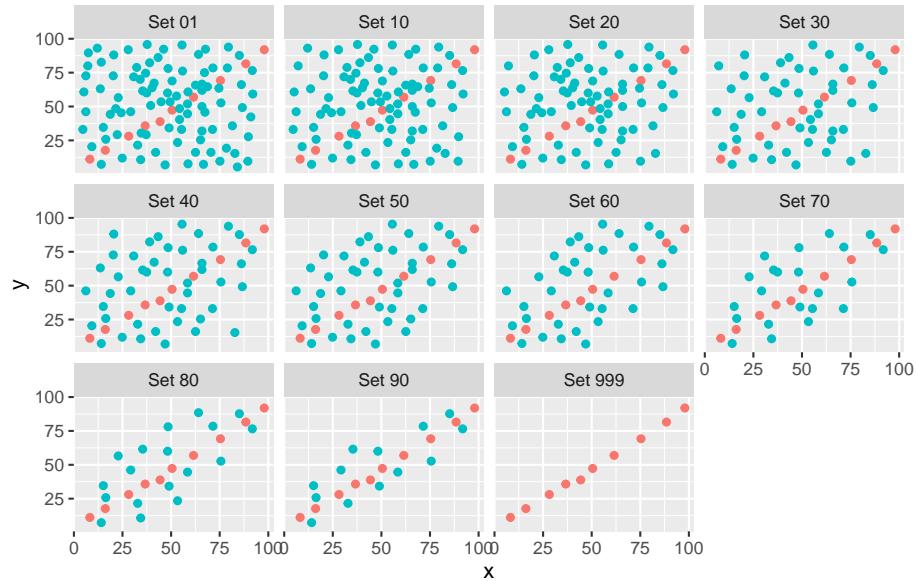
`geom_smooth()` using method = 'loess' and formula 'y ~ x'

Even if a loess smooth suggests non-linearity.



```
ggplot(correx2, aes(x = x, y = y, color = factor(group))) +  
  geom_point() +  
  guides(color = "none") +  
  facet_wrap(~ set) +  
  labs(title = "Note: The same 10 points (in red) are in each plot.")
```

Note: The same 10 points (in red) are in each plot.



Note that the same 10 points are used in each of the data sets. It's always possible that a lurking subgroup of the data within a scatterplot follows a very strong linear relationship. This is why it's so important (and difficult) not to go searching for such a thing without a strong foundation of logic, theory and prior empirical evidence.

11.6 The Spearman Rank Correlation

The Spearman rank correlation coefficient is a rank-based measure of statistical dependence that assesses how well the relationship between X and Y can be described using a **monotone function** even if that relationship is not linear.

- A monotone function preserves order, that is, Y must either be strictly increasing as X increases, or strictly decreasing as X increases.
- A Spearman correlation of 1.0 indicates simply that as X increases, Y always increases.
- Like the Pearson correlation, the Spearman correlation is dimension-free, and falls between -1 and +1.
- A positive Spearman correlation corresponds to an increasing (but not necessarily linear) association between X and Y, while a negative Spearman correlation corresponds to a decreasing (but again not necessarily linear) association.

11.6.1 Spearman Formula

To calculate the Spearman rank correlation, we take the ranks of the X and Y data, and then apply the usual Pearson correlation. To find the ranks, sort X and Y into ascending order, and then number them from 1 (smallest) to n (largest). In the event of a tie, assign the average rank to the tied subjects.

11.6.2 Comparing Pearson and Spearman Correlations

Let's look at the `nnyfs` data again.

```
nnyfs %$% cor(fat, protein)

[1] 0.667

nnyfs %$% cor(fat, protein, method = "spearman")

[1] 0.658
```

The Spearman and Pearson correlations are not especially different in this case.

11.6.3 Spearman vs. Pearson Example 1

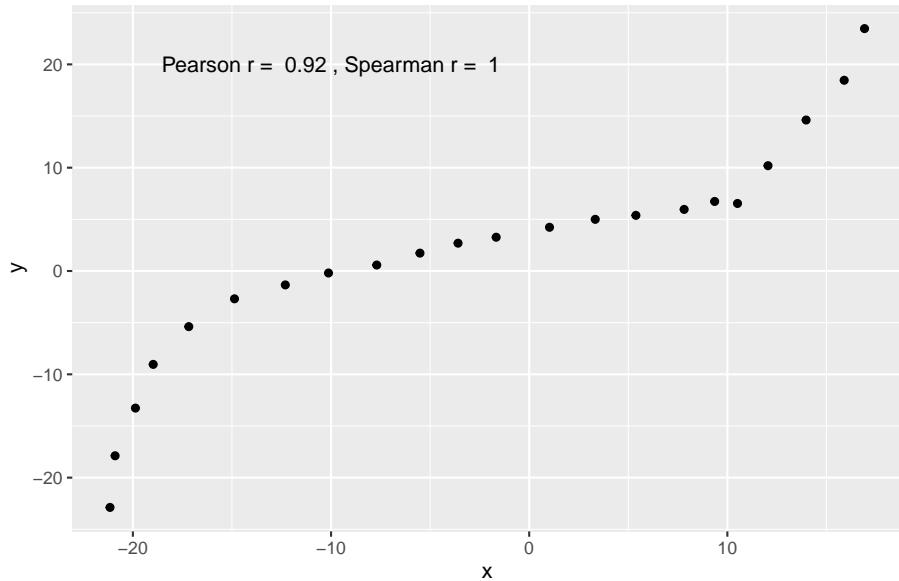
The next few plots describe relationships where we anticipate the Pearson and Spearman correlations might differ in their conclusions.

```
spear1 <- read.csv("data/spear1.csv")
spear2 <- read.csv("data/spear2.csv")
spear3 <- read.csv("data/spear3.csv")
spear4 <- read.csv("data/spear4.csv")
# used read.csv above because these are just toy examples with
# two columns per data set and no row numbering
```

Example 1 shows a function where the Pearson correlation is 0.925 (a strong but not perfect linear relation), but the Spearman correlation is 1 because the relationship is monotone, even though it is not perfectly linear.

```
ggplot(spear1, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "Spearman vs. Pearson, Example 1") +
  annotate("text", x = -10, y = 20,
           label = paste("Pearson r = ",
                         signif(cor(spear1$x, spear1$y),2),
                         ", Spearman r = ",
                         signif(cor(spear1$x, spear1$y, method = "spearman"),2)))
```

Spearman vs. Pearson, Example 1



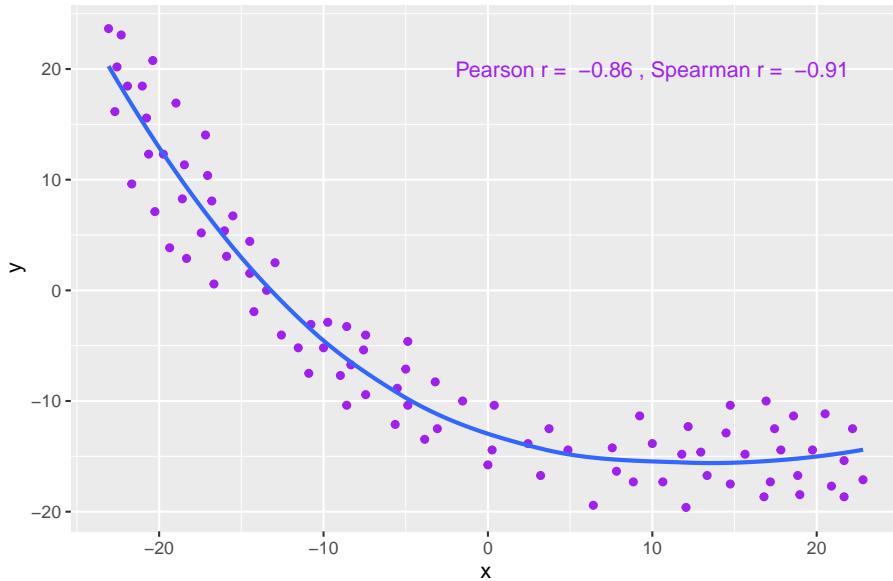
So, a positive Spearman correlation corresponds to an increasing (but not necessarily linear) association between x and y .

11.6.4 Spearman vs. Pearson Example 2

Example 2 shows that a negative Spearman correlation corresponds to a decreasing (but, again, not necessarily linear) association between x and y .

```
ggplot(spear2, aes(x = x, y = y)) +
  geom_point(col = "purple") +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Spearman vs. Pearson, Example 2") +
  annotate("text", x = 10, y = 20, col = "purple",
          label = paste("Pearson r = ",
                        signif(cor(spear2$x, spear2$y),2),
                        ", Spearman r = ",
                        signif(cor(spear2$x, spear2$y, method = "spearman"),2)))
```

Spearman vs. Pearson, Example 2



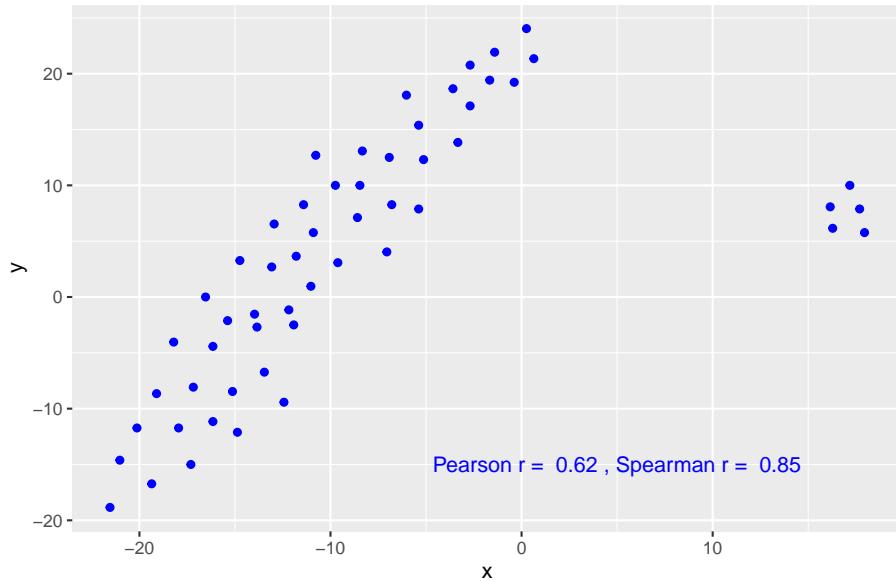
11.6.5 Spearman vs. Pearson Example 3

The Spearman correlation is less sensitive than the Pearson correlation is to strong outliers that are unusual on either the X or Y axis, or both. That is because the Spearman rank coefficient limits the outlier to the value of its rank.

In Example 3, for instance, the Spearman correlation reacts much less to the outliers around $X = 12$ than does the Pearson correlation.

```
ggplot(spear3, aes(x = x, y = y)) +
  geom_point(col = "blue") +
  labs(title = "Spearman vs. Pearson, Example 3") +
  annotate("text", x = 5, y = -15, col = "blue",
          label = paste("Pearson r = ",
                        signif(cor(spear3$x, spear3$y),2),
                        ", Spearman r = ",
                        signif(cor(spear3$x, spear3$y, method = "spearman"),2)))
```

Spearman vs. Pearson, Example 3

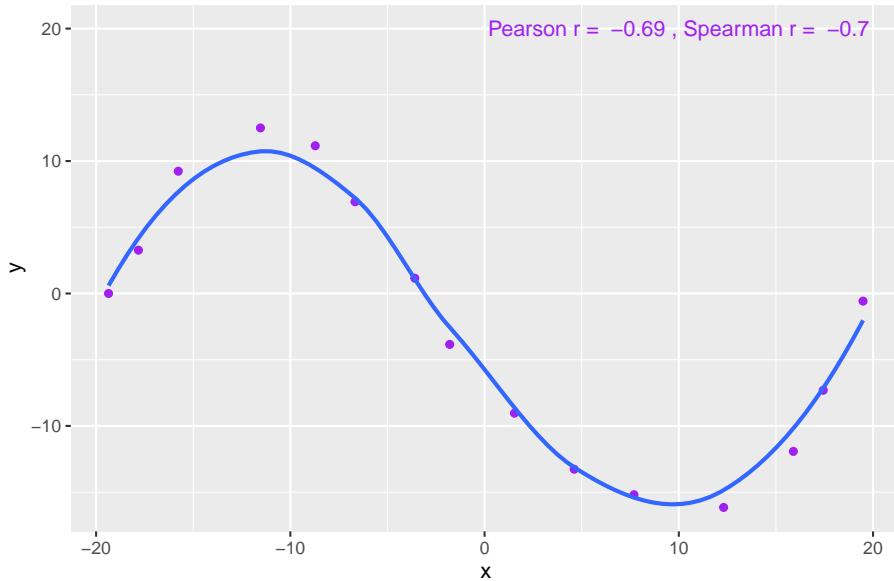


11.6.6 Spearman vs. Pearson Example 4

The use of a Spearman correlation is no substitute for looking at the data. For non-monotone data like what we see in Example 4, neither the Spearman nor the Pearson correlation alone provides much guidance, and just because they are (essentially) telling you the same thing, that doesn't mean what they're telling you is all that helpful.

```
ggplot(spear4, aes(x = x, y = y)) +
  geom_point(col = "purple") +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Spearman vs. Pearson, Example 4") +
  annotate("text", x = 10, y = 20, col = "purple",
           label = paste("Pearson r = ",
                         signif(cor(spear4$x, spear4$y),2),
                         ", Spearman r = ",
                         signif(cor(spear4$x, spear4$y, method = "spearman"),2)))
```

Spearman vs. Pearson, Example 4



Chapter 12

Studying Crab Claws (`crabs`)

For our next example, we'll consider a study from zoology, specifically carcinology - the study of crustaceans. My source for these data is Chapter 7 in Ramsey and Schafer (2002) which drew the data from a figure in Yamada and Boulding (1998).

The available data are the mean closing forces (in Newtons) and the propodus heights (mm) of the claws on 38 crabs that came from three different species. The *propodus* is the segment of the crab's clawed leg with an immovable finger and palm.

This was part of a study of the effects that predatory intertidal crab species have on populations of snails. The three crab species under study are:

- 14 *Hemigrapsus nudus*, also called the purple shore crab (14 crabs)
- 12 *Lophopanopeus bellus*, also called the black-clawed pebble crab, and
- 12 *Cancer productus*, one of several species of red rock crabs (12)

```
crabs <- read_csv("data/crabs.csv")  
  
Parsed with column specification:  
cols(  
  crab = col_double(),  
  species = col_character(),  
  force = col_double(),  
  height = col_double()  
)  
crabs  
  
# A tibble: 38 x 4
```

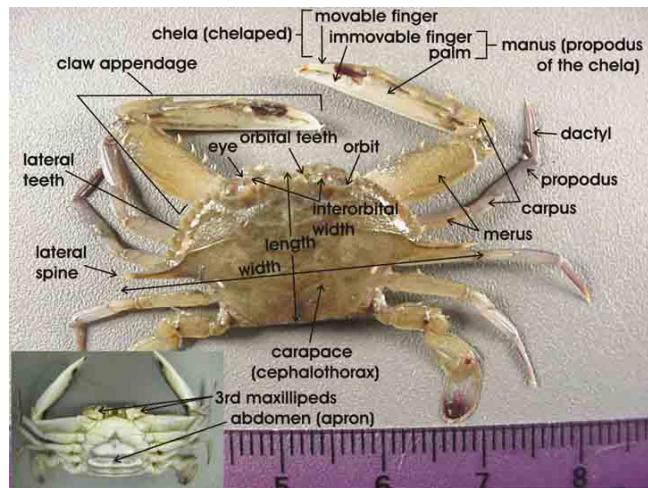


Figure 12.1: Source: <http://txmarspecies.tamug.edu/crustglossary.cfm>

	crab species	force	height
	<dbl> <chr>	<dbl>	<dbl>
1	1 Hemigrapsus nudus	4	8
2	2 Lophopanopeus bellus	15.1	7.9
3	3 Cancer productus	5	6.7
4	4 Lophopanopeus bellus	2.9	6.6
5	5 Hemigrapsus nudus	3.2	5
6	6 Hemigrapsus nudus	9.5	7.9
7	7 Cancer productus	22.5	9.4
8	8 Hemigrapsus nudus	7.4	8.3
9	9 Cancer productus	14.6	11.2
10	10 Lophopanopeus bellus	8.7	8.6
# ... with 28 more rows			

The `species` information is stored here as a character variable. How many different crabs are we talking about in each `species`?

```
crabs %>% tabyl(species)
```

species	n	percent
Cancer productus	12	0.316
Hemigrapsus nudus	14	0.368
Lophopanopeus bellus	12	0.316

As it turns out, we're going to want to treat the `species` information as a **factor** with three levels, rather than as a character variable.

```
crabs <- crabs %>%
  mutate(species = factor(species))
```

Here's a quick summary of the data. Take care to note the useless results for the first two variables. At least the function flags with a * those variables it thinks are non-numeric.

```
psych::describe(crabs)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
crab	1	38	19.50	11.11	19.50	19.50	14.08	1	38.0	37.0	0.00
species*	2	38	2.00	0.81	2.00	2.00	1.48	1	3.0	2.0	0.00
force	3	38	12.13	8.98	8.70	11.53	9.04	2	29.4	27.4	0.47
height	4	38	8.81	2.23	8.25	8.78	2.52	5	13.1	8.1	0.19
			kurtosis	se							
crab			-1.30	1.80							
species*			-1.50	0.13							
force			-1.25	1.46							
height			-1.14	0.36							

Actually, we're more interested in these results after grouping by species.

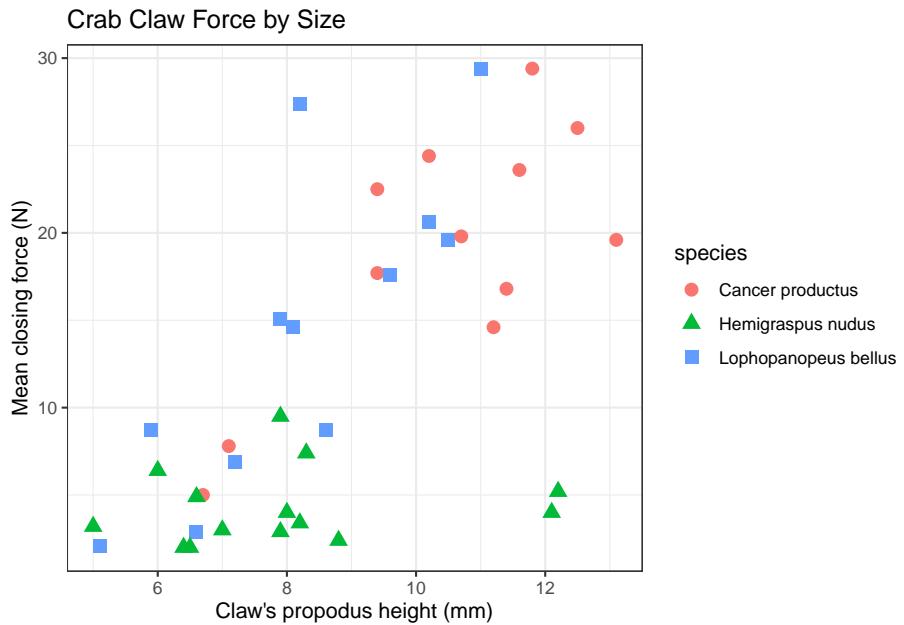
```
crabs %>%
  group_by(species) %>%
  summarise(n = n(), median(force), median(height))
```

	species	n	median(force)	median(height)
1	Cancer productus	12	19.7	11.0
2	Hemigrapsus nudus	14	3.7	7.9
3	Lophopanopeus bellus	12	14.8	8.15

12.1 Association of Size and Force

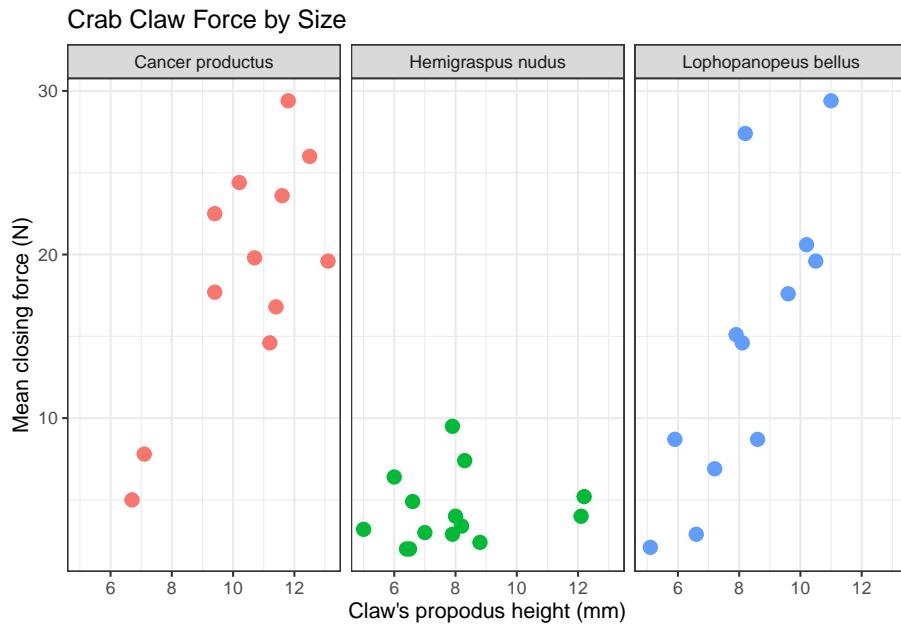
Suppose we want to describe force on the basis of height, across all 38 crabs. We'll add titles and identify the three species of crab, using shape and color.

```
ggplot(crabs, aes(x = height, y = force, color = species, shape = species)) +
  geom_point(size = 3) +
  labs(title = "Crab Claw Force by Size",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)") +
  theme_bw()
```



A faceted plot for each species really highlights the difference in force between the *Hemigrapsus nudus* and the other two species of crab.

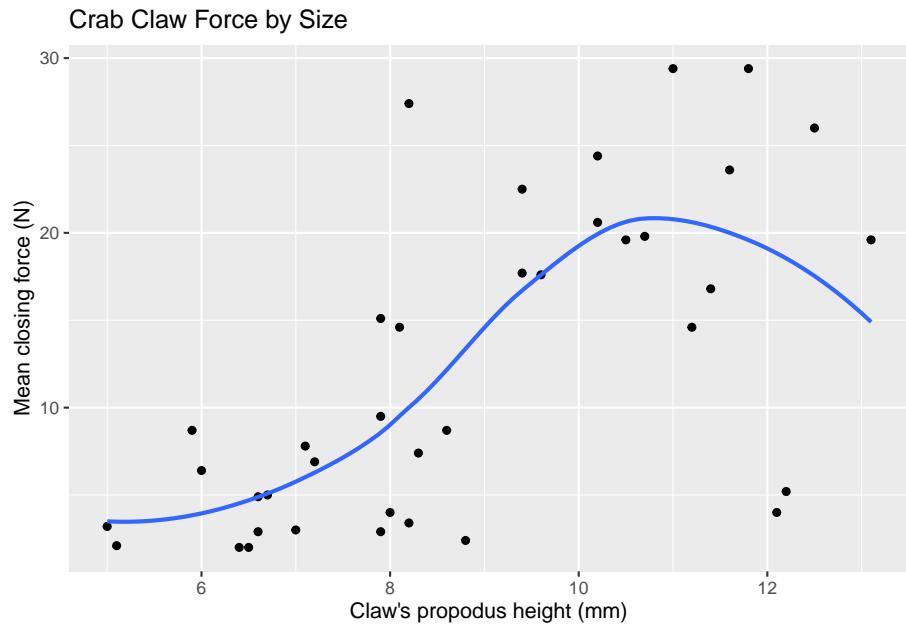
```
gplot(crabs, aes(x = height, y = force, color = species)) +
  geom_point(size = 3) +
  facet_wrap(~ species) +
  guides(color = FALSE) +
  labs(title = "Crab Claw Force by Size",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)") +
  theme_bw()
```



12.2 The loess smooth

We can obtain a smoothed curve (using several different approaches) to summarize the pattern presented by the data in any scatterplot. For instance, we might build such a plot for the complete set of 38 crabs, adding in a non-linear smooth function (called a loess smooth.)

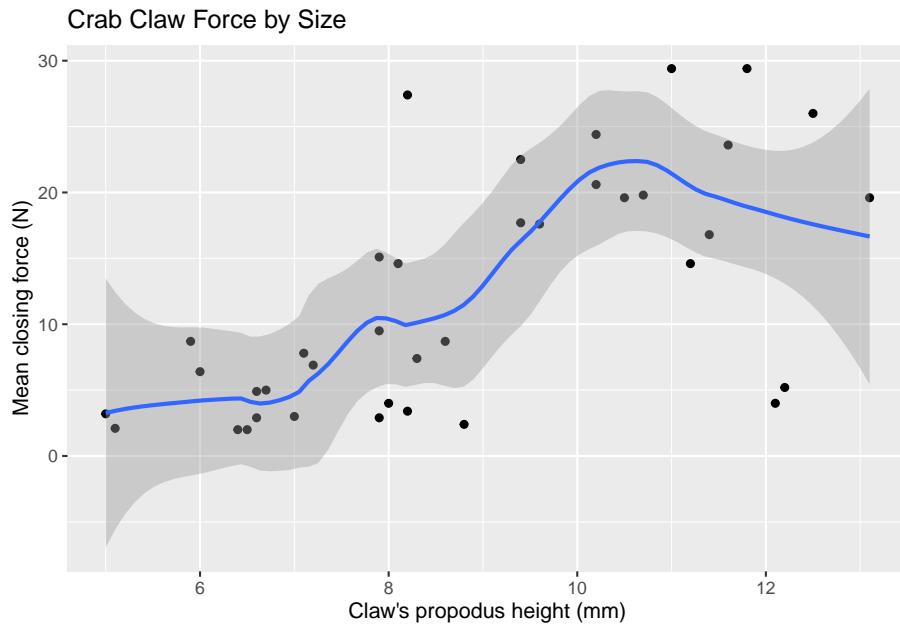
```
ggplot(crabs, aes(x = height, y = force)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Crab Claw Force by Size",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)")
```



As we have discussed previously, a **loess smooth** fits a curve to data by tracking (at point x) the points within a neighborhood of point x , with more emphasis given to points near x . It can be adjusted by tweaking the `span` and `degree` parameters.

In addition to the curve, smoothing procedures can also provide confidence intervals around their main fitted line. Consider the following plot of the `crabs` information, which adjusts the span (from its default of 0.75) and also adds in the confidence intervals.

```
ggplot(crabs, aes(x = height, y = force)) +
  geom_point() +
  geom_smooth(method = "loess", span = 0.5, se = TRUE) +
  labs(title = "Crab Claw Force by Size",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)")
```

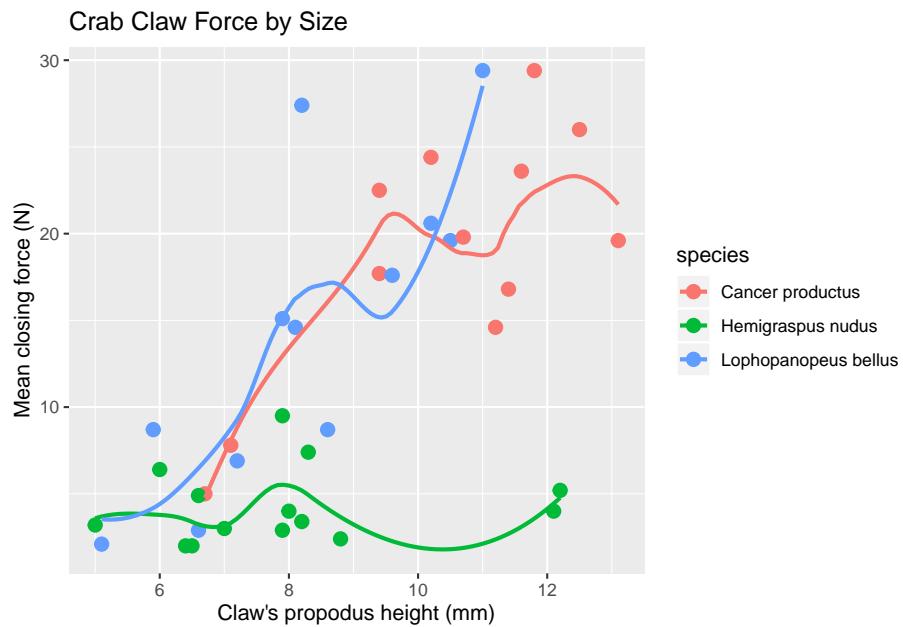


By reducing the size of the span, our resulting picture shows a much less smooth function that we generated previously.

12.2.1 Smoothing within Species

We can, of course, produce the plot above with separate smooths for each of the three species of crab.

```
ggplot(crabs, aes(x = height, y = force, group = species, color = species)) +
  geom_point(size = 3) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Crab Claw Force by Size",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)")
```



If we want to add in the confidence intervals (here I'll show them at 90% rather than the default of 95%) then this plot should be faceted. Note that by default, what is displayed when `se = TRUE` are 95% prediction intervals - the `level` function in `stat_smooth` [which can be used in place of `geom_smooth`] is used here to change the coverage percentage from 95% to 90%.

```
ggplot(crabs, aes(x = height, y = force, group = species, color = species)) +
  geom_point() +
  stat_smooth(method = "loess", level = 0.90, se = TRUE) +
  guides(color = FALSE) +
  labs(title = "Crab Claw Force by Size",
       caption = "with loess smooths and 90% confidence intervals",
       x = "Claw's propodus height (mm)", y = "Mean closing force (N)") +
  facet_wrap(~ species)
```

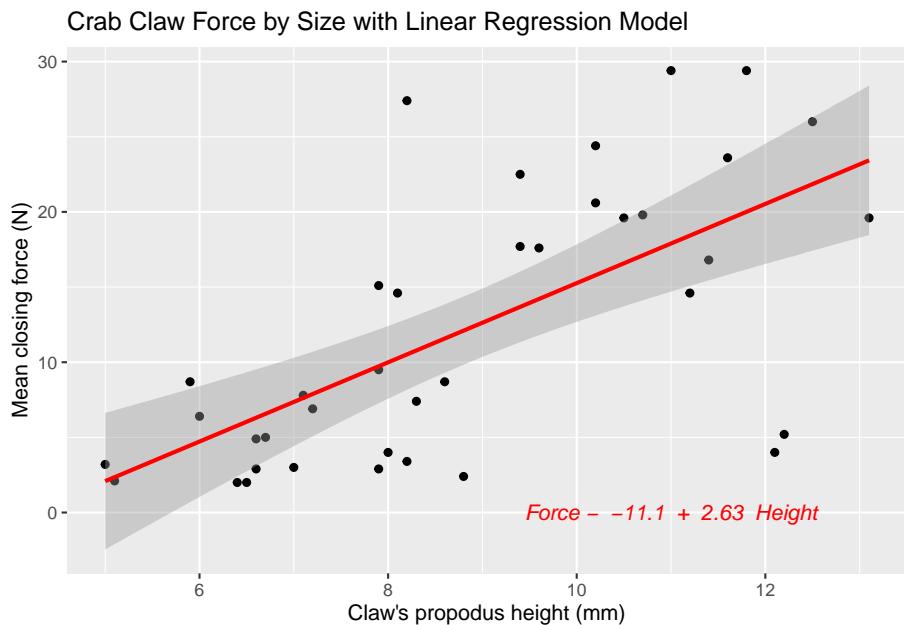


More on these and other confidence intervals later, especially in part B.

12.3 Fitting a Linear Regression Model

Suppose we plan to use a simple (least squares) linear regression model to describe force as a function of height. Is a least squares model likely to be an effective choice here?

The plot below shows the regression line predicting closing force as a function of propodus height. Here we annotate the plot to show the actual fitted regression line, which required fitting it with the `lm` statement prior to developing the graph.



```
rm(mod)
```

The **lm** function, again, specifies the linear model we fit to predict force using height. Here's the summary.

```
summary(lm(force ~ height, data = crabs))
```

Call:

```
lm(formula = force ~ height, data = crabs)
```

Residuals:

Min	1Q	Median	3Q	Max
-16.794	-3.811	-0.239	4.144	16.881

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-11.087	4.622	-2.40	0.022 *
height	2.635	0.509	5.18	8.7e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.89 on 36 degrees of freedom
 Multiple R-squared: 0.427, Adjusted R-squared: 0.411
 F-statistic: 26.8 on 1 and 36 DF, p-value: 8.73e-06

Again, the key things to realize are:

- The outcome variable in this model is **force**, and the predictor variable is **height**.
- The straight line model for these data fitted by least squares is force = $-11.1 + 2.63 \text{ height}$.
- The slope of height is positive, which indicates that as height increases, we expect that force will also increase. Specifically, we expect that for every additional mm of height, the force will increase by 2.63 Newtons.
- The multiple R-squared (squared correlation coefficient) is 0.427, which implies that 42.7% of the variation in force is explained using this linear model with height. It also implies that the Pearson correlation between force and height is the square root of 0.427, or 0.653.

12.4 Is a Linear Model Appropriate?

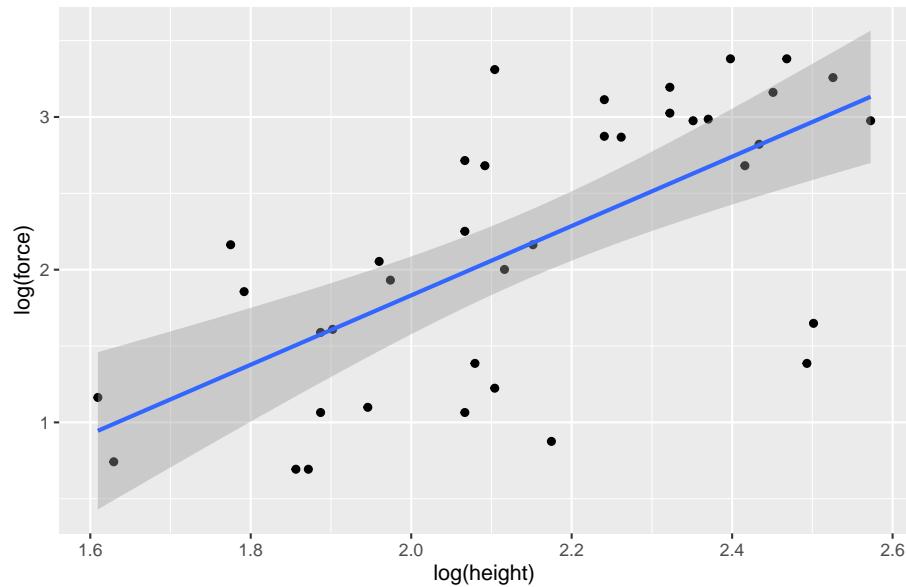
The zoology (at least as described in Ramsey and Schafer (2002)) suggests that the actual nature of the relationship would be represented by a log-log relationship, where the log of force is predicted by the log of height.

This log-log model is an appropriate model when we think that percentage increases in X (height, here) lead to constant percentage increases in Y (here, force).

To see the log-log model in action, we plot the log of force against the log of height. We could use either base 10 (`log10` in R) or natural (`log` in R) logarithms.

```
ggplot(crabs, aes(x = log(height), y = log(force))) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Log-Log Model for Crabs data")
```

Log–Log Model for Crabs data



The correlations between the raw force and height and between their logarithms turn out to be quite similar, and because the log transformation is monotone in these data, there's actually no change at all in the Spearman correlations.

Correlation of	Pearson r	Spearman r
force and height	0.653	0.657
log(force) and log(height)	0.662	0.657

12.4.1 The log–log model

```
crab_loglog <- lm(log(force) ~ log(height), data = crabs)
summary(crab_loglog)
```

Call:
`lm(formula = log(force) ~ log(height), data = crabs)`

Residuals:

Min	1Q	Median	3Q	Max
-1.566	-0.445	0.188	0.480	1.242

Coefficients:

```

Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.710     0.925   -2.93   0.0059 **
log(height)  2.271     0.428    5.30    6e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 0.675 on 36 degrees of freedom
 Multiple R-squared: 0.438, Adjusted R-squared: 0.423
 F-statistic: 28.1 on 1 and 36 DF, p-value: 5.96e-06

Our regression equation is $\log(\text{force}) = -2.71 + 2.27 \log(\text{height})$.

So, for example, if we found a crab with propodus height = 10 mm, our prediction for that crab's claw force (in Newtons) based on this log-log model would be...

- $\log(\text{force}) = -2.71 + 2.27 \log(10)$
- $\log(\text{force}) = -2.71 + 2.27 \times 2.303$
- $\log(\text{force}) = 2.519$
- and so predicted force = $\exp(2.519) = 12.417$ Newtons, which, naturally, we would round to 12.4 Newtons to match the data set's level of precision.

12.4.2 How does this compare to our original linear model?

```

crab_linear <- lm(force ~ height, data = crabs)

summary(crab_linear)

```

Call:
`lm(formula = force ~ height, data = crabs)`

Residuals:

Min	1Q	Median	3Q	Max
-16.794	-3.811	-0.239	4.144	16.881

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-11.087	4.622	-2.40	0.022 *
height	2.635	0.509	5.18	8.7e-06 ***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.' 0.1 ' ' 1

Residual standard error: 6.89 on 36 degrees of freedom
 Multiple R-squared: 0.427, Adjusted R-squared: 0.411
 F-statistic: 26.8 on 1 and 36 DF, p-value: 8.73e-06

The linear regression equation is $\text{force} = -11.1 + 2.63 \text{ height}$.

So, for example, if we found a crab with propodus height = 10 mm, our prediction for that crab's claw force (in Newtons) based on this linear model would be...

- $\text{force} = -11.087 + 2.635 \times 10$
- $\text{force} = -11.087 + 26.348$
- so predicted force = 15.261, which we would round to 15.3 Newtons.

So, it looks like the two models give meaningfully different predictions.

12.5 Making Predictions with a Model

A simpler way to get predictions for a new value like height = 10 mm from our models is available.

```
predict(crab_linear, data.frame(height = 10), interval = "prediction")
fit lwr upr
1 15.3 1.05 29.5
```

We'd interpret this result as saying that the linear model's predicted force associated with a single new crab claw with propodus height 10 mm is 15.3 Newtons, and that a 95% prediction interval for the true value of such a force for such a claw is between 1.0 and 29.5 Newtons. More on prediction intervals later.

12.5.1 Predictions After a Transformation

We can also get predictions from the log-log model.

```
predict(crab_loglog, data.frame(height = 10), interval = "prediction")
fit lwr upr
1 2.52 1.13 3.91
```

Of course, this prediction is of the `log(force)` for such a crab claw. To get the prediction in terms of simple force, we'd need to back out of the logarithm, by exponentiating our point estimate and the prediction interval endpoints.

```
exp(predict(crab_loglog, data.frame(height = 10), interval = "prediction"))
fit lwr upr
1 12.4 3.08 50
```

We'd interpret this result as saying that the log-log model's predicted force associated with a single new crab claw with propodus height 10 mm is 12.4 Newtons, and that a 95% prediction interval for the true value of such a force for such a claw is between 3.1 and 50.0 Newtons.

12.5.2 Comparing Model Predictions

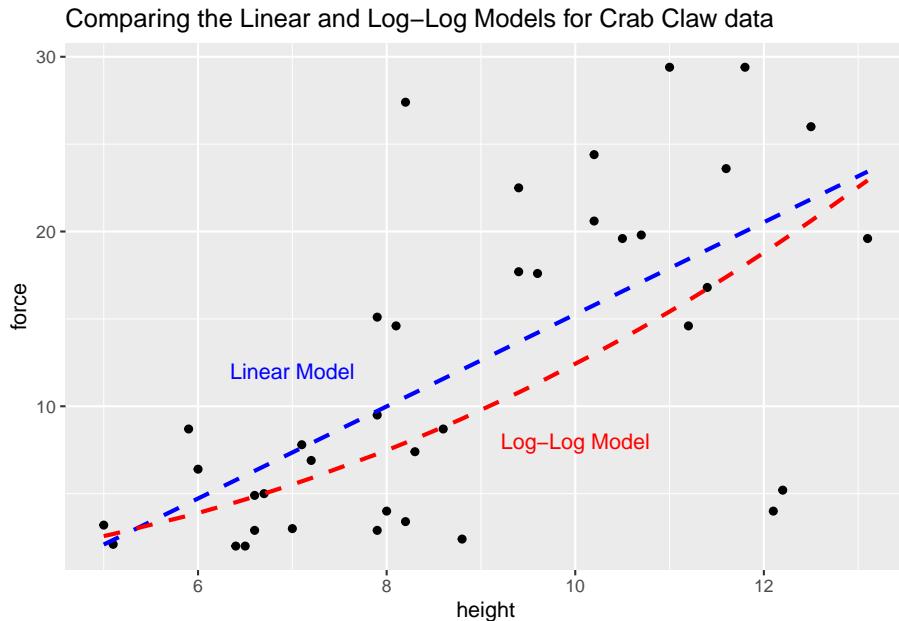
Suppose we wish to build a plot of force vs height with a straight line for the linear model's predictions, and a new curve for the log-log model's predictions, so that we can compare and contrast the implications of the two models on a common scale. The `predict` function, when not given a new data frame, will use the existing predictor values that are in our `crabs` data. Such predictions are often called fitted values.

To put the two sets of predictions on the same scale despite the differing outcomes in the two models, we'll exponentiate the results of the log-log model, and build a little data frame containing the heights and the predicted forces from that model.

```
loglogdat <- tibble(height = crabs$height, force = exp(predict(crab_loglog)))
```

Now, we're ready to use the `geom_smooth` approach to plot the linear fit, and `geom_line` (which also fits curves) to display the log-log fit.

```
ggplot(crabs, aes(x = height, y = force)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col="blue", linetype = 2) +
  geom_line(data = loglogdat, col = "red", linetype = 2, size = 1) +
  annotate("text", 7, 12, label = "Linear Model", col = "blue") +
  annotate("text", 10, 8, label = "Log-Log Model", col = "red") +
  labs(title = "Comparing the Linear and Log-Log Models for Crab Claw data")
```



Based on these 38 crabs, we see some modest differences between the predictions of the two models, with the log-log model predicting generally lower closing force for a given propodus height than would be predicted by a linear model.

```
rm(loglogdat, crab_linear, crab_loglog)
```

Chapter 13

The Western Collaborative Group Study

13.1 The Western Collaborative Group Study (wcgs) data set

Vittinghoff et al. (2012) explore data from the Western Collaborative Group Study (WCGS) in great detail¹. We'll touch lightly on some key issues in this Chapter.

```
wcgs <- read.csv("data/wcgs.csv") %>%tbl_df  
  
wcgs  
  
# A tibble: 3,154 x 22  
  id    age  agec   height weight lnwght wghtcat    bmi    sbp lnsbp    dbp  
  <int> <int> <fct>  <int>  <int>  <dbl> <fct>  <dbl> <int>  <dbl> <int>  
1 2343    50 46-50     67    200    5.30 170-200  31.3   132   4.88    90  
2 3656    51 51-55     73    192    5.26 170-200  25.3   120   4.79    74  
3 3526    59 56-60     70    200    5.30 170-200  28.7   158   5.06    94  
4 22057   51 51-55     69    150    5.01 140-170  22.1   126   4.84    80  
5 12927   44 41-45     71    160    5.08 140-170  22.3   126   4.84    80  
6 16029   47 46-50     64    158    5.06 140-170  27.1   116   4.75    76  
7 3894    40 35-40     70    162    5.09 140-170  23.2   122   4.80    78  
8 11389   41 41-45     70    160    5.08 140-170  23.0   130   4.87    84  
9 12681   50 46-50     71    195    5.27 170-200  27.2   112   4.72    70  
10 10005   43 41-45    68    187    5.23 170-200  28.4   120   4.79    80
```

¹For more on the WCGS, you might look at <http://www.epi.umn.edu/cvdepi/study-synopsis/western-collaborative-group-study/>

```
# ... with 3,144 more rows, and 11 more variables: chol <int>,
#   behpat <fct>, dibpat <fct>, smoke <fct>, ncigs <int>, arcus <int>,
#   chd69 <fct>, typchd69 <int>, time169 <int>, t1 <dbl>, uni <dbl>
```

Here, we have 3154 rows (subjects) and 22 columns (variables). Since I used `read.csv` to import the data, and then converted to a tibble, all variables containing character data will appear as factors.

13.1.1 Structure of `wcgs`

We can specify the (sometimes terrible) variable names, through the `names` function, or we can add other elements of the structure, so that we can identify elements of particular interest.

```
str(wcgs)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame': 3154 obs. of 22 variables:
 $ id      : int  2343 3656 3526 22057 12927 16029 3894 11389 12681 10005 ...
 $ age     : int  50 51 59 51 44 47 40 41 50 43 ...
 $ agec    : Factor w/ 5 levels "35-40","41-45",...: 3 4 5 4 2 3 1 2 3 2 ...
 $ height  : int  67 73 70 69 71 64 70 70 71 68 ...
 $ weight  : int  200 192 200 150 160 158 162 160 195 187 ...
 $ lnwght  : num  5.3 5.26 5.3 5.01 5.08 ...
 $ wghtcat: Factor w/ 4 levels "< 140","> 200",...: 4 4 4 3 3 3 3 3 4 4 ...
 $ bmi     : num  31.3 25.3 28.7 22.1 22.3 ...
 $ sbp     : int  132 120 158 126 126 116 122 130 112 120 ...
 $ lnsbp   : num  4.88 4.79 5.06 4.84 4.84 ...
 $ dbp     : int  90 74 94 80 80 76 78 84 70 80 ...
 $ chol    : int  249 194 258 173 214 206 190 212 130 233 ...
 $ behpat  : Factor w/ 4 levels "A1","A2","B3",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ dibpat  : Factor w/ 2 levels "Type A","Type B": 1 1 1 1 1 1 1 1 1 1 ...
 $ smoke   : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 2 1 2 1 2 ...
 $ ncigs   : int  25 25 0 0 0 80 0 25 0 25 ...
 $ arcus   : int  1 0 1 1 0 0 0 0 1 0 ...
 $ chd69   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ typchd69: int  0 0 0 0 0 0 0 0 0 0 ...
 $ time169 : int  1367 2991 2960 3069 3081 2114 2929 3010 3104 2861 ...
 $ t1      : num  -1.63 -4.06 0.64 1.12 2.43 ...
 $ uni     : num  0.486 0.186 0.728 0.624 0.379 ...
```

13.1.2 Codebook for `wcgs`

This table was lovingly hand-crafted, and involved a lot of typing. We'll look for better ways in 432.

13.1. THE WESTERN COLLABORATIVE GROUP STUDY (*wcgs*) DATA SET245

Name	Stored As	Type	Details (units, levels, etc.)
<code>id</code>	integer	(nominal)	ID #, nominal and uninteresting
<code>age</code>	integer	quantitative	age, in years - no decimal places
<code>agec</code>	factor (5)	(ordinal)	age: 35-40, 41-45, 46-50, 51-55, 56-60
<code>height</code>	integer	quantitative	height, in inches
<code>weight</code>	integer	quantitative	weight, in pounds
<code>lnwght</code>	number	quantitative	natural logarithm of weight
<code>wghtcat</code>	factor (4)	(ordinal)	wt: < 140, 140-170, 170-200, > 200
<code>bmi</code>	number	quantitative	body-mass index: $703 * \text{weight in lb} / (\text{height in in})^2$
<code>sbp</code>	integer	quantitative	systolic blood pressure, in mm Hg
<code>lnsbp</code>	number	quantitative	natural logarithm of sbp
<code>dbp</code>	integer	quantitative	diastolic blood pressure, mm Hg
<code>chol</code>	integer	quantitative	total cholesterol, mg/dL
<code>behpatt</code>	factor (4)	(nominal)	behavioral pattern: A1, A2, B3 or B4
<code>dibpat</code>	factor (2)	(binary)	behavioral pattern: A or B
<code>smoke</code>	factor (2)	(binary)	cigarette smoker: Yes or No
<code>ncigs</code>	integer	quantitative	number of cigarettes smoked per day
<code>arcus</code>	integer	(nominal)	arcus senilis present (1) or absent (0)
<code>chd69</code>	factor (2)	(binary)	CHD event: Yes or No
<code>typchd69</code>	integer	(4 levels)	event: 0 = no CHD, 1 = MI or SD, 2 = silent MI, 3 = angina
<code>time169</code>	integer	quantitative	follow-up time in days
<code>t1</code>	number	quantitative	heavy-tailed (random draws)
<code>uni</code>	number	quantitative	light-tailed (random draws)

13.1.3 Quick Summary

```
summary(wcgs)
```

<code>id</code>		<code>age</code>	<code>agec</code>	<code>height</code>	<code>weight</code>
Min.	: 2001	Min. :39.0	35-40: 543	Min. :60.0	Min. : 78
1st Qu.	: 3741	1st Qu.:42.0	41-45:1091	1st Qu.:68.0	1st Qu.:155
Median	:11406	Median :45.0	46-50: 750	Median :70.0	Median :170
Mean	:10478	Mean :46.3	51-55: 528	Mean :69.8	Mean :170
3rd Qu.	:13115	3rd Qu.:50.0	56-60: 242	3rd Qu.:72.0	3rd Qu.:182
Max.	:22101	Max. :59.0		Max. :78.0	Max. :320
<code>lnwght</code>		<code>wghtcat</code>	<code>bmi</code>	<code>sbp</code>	<code>lnsbp</code>
Min.	:4.36	< 140 : 232	Min. :11.2	Min. : 98	Min. :4.58
1st Qu.	:5.04	> 200 : 213	1st Qu.:23.0	1st Qu.:120	1st Qu.:4.79
Median	:5.14	140-170:1538	Median :24.4	Median :126	Median :4.84
Mean	:5.13	170-200:1171	Mean :24.5	Mean :129	Mean :4.85

```

3rd Qu.:5.20          3rd Qu.:25.8   3rd Qu.:136   3rd Qu.:4.91
Max.    :5.77          Max.    :38.9    Max.    :230    Max.    :5.44

      dbp       chol     behpat     dibpat     smoke
Min.  : 58   Min.  :103   A1: 264   Type A:1589   No :1652
1st Qu.: 76   1st Qu.:197   A2:1325   Type B:1565   Yes:1502
Median : 80   Median :223   B3:1216
Mean   : 82   Mean   :226   B4: 349
3rd Qu.: 86   3rd Qu.:253
Max.    :150   Max.    :645
NA's    :12

      ncigs      arcus     chd69     typchd69     time169
Min.  : 0.0   Min.  :0.000   No :2897   Min.  :0.000   Min.  : 18
1st Qu.: 0.0   1st Qu.:0.000   Yes: 257   1st Qu.:0.000   1st Qu.:2842
Median : 0.0   Median :0.000
Mean   :11.6   Mean   :0.299
3rd Qu.:20.0   3rd Qu.:1.000
Max.    :99.0   Max.    :1.000
NA's    :2

      t1        uni
Min.  :-47.4   Min.  :0.001
1st Qu.: -1.0   1st Qu.:0.257
Median :  0.0   Median :0.516
Mean   :  0.0   Mean   :0.505
3rd Qu.:  1.0   3rd Qu.:0.756
Max.    :47.0    Max.  :0.999
NA's    :39

```

For a more detailed description, we might consider `Hmisc::describe`, `psych::describe`, `mosaic::favstats`, etc.

13.2 Are the SBPs Normally Distributed?

Consider the question of whether the distribution of the systolic blood pressure results is well-approximated by the Normal.

```

res <- mosaic::favstats(~ sbp, data = wcgs)
bin_w <- 5 # specify binwidth

ggplot(wcgs, aes(x = sbp)) +
  geom_histogram(binwidth = bin_w,
                 fill = "orchid",
                 col = "blue") +
  theme_bw() +
  stat_function(

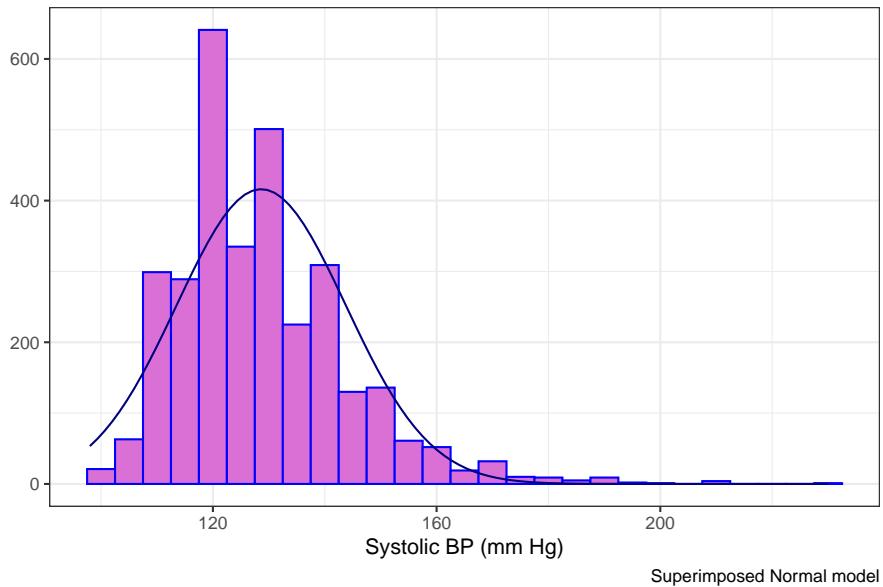
```

```

fun = function(x) dnorm(x, mean = res$mean,
                        sd = res$sd) *
  res$n * bin_w,
  col = "navy") +
  labs(title = "Systolic BP for `wcgs` subjects",
       x = "Systolic BP (mm Hg)", y = "",
       caption = "Superimposed Normal model")

```

Systolic BP for `wcgs` subjects



Since the data contain both `sbp` and `lnsbp` (its natural logarithm), let's compare them. Note that in preparing the graph, we'll need to change the location for the text annotation.

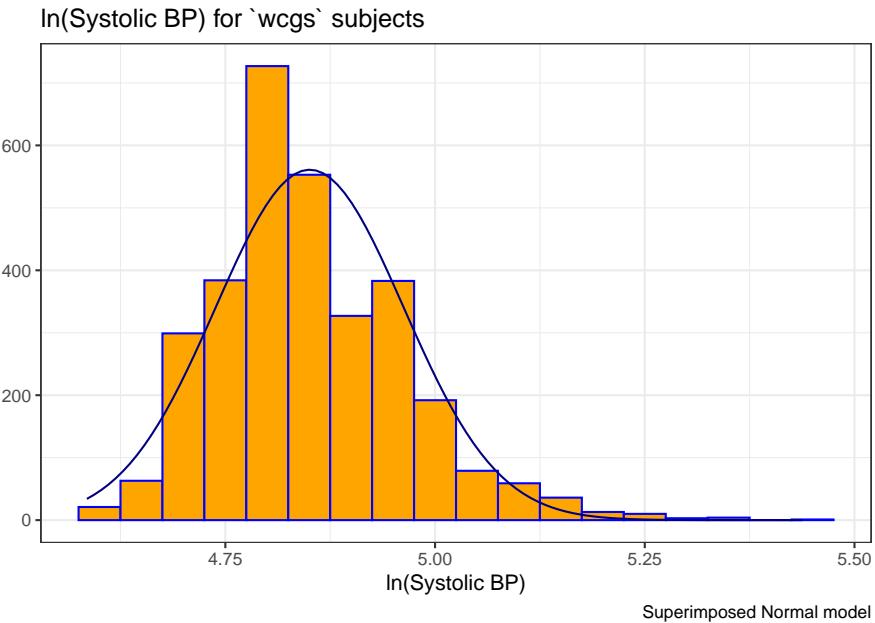
```

res <- mosaic::favstats(~ lnsbp, data = wcgs)
bin_w <- 0.05 # specify binwidth

ggplot(wcgs, aes(x = lnsbp)) +
  geom_histogram(binwidth = bin_w,
                 fill = "orange",
                 col = "blue") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                           sd = res$sd) *
      res$n * bin_w,
    col = "navy") +

```

```
labs(title = "ln(Systolic BP) for `wcgs` subjects",
  x = "ln(Systolic BP)", y = "",
  caption = "Superimposed Normal model")
```



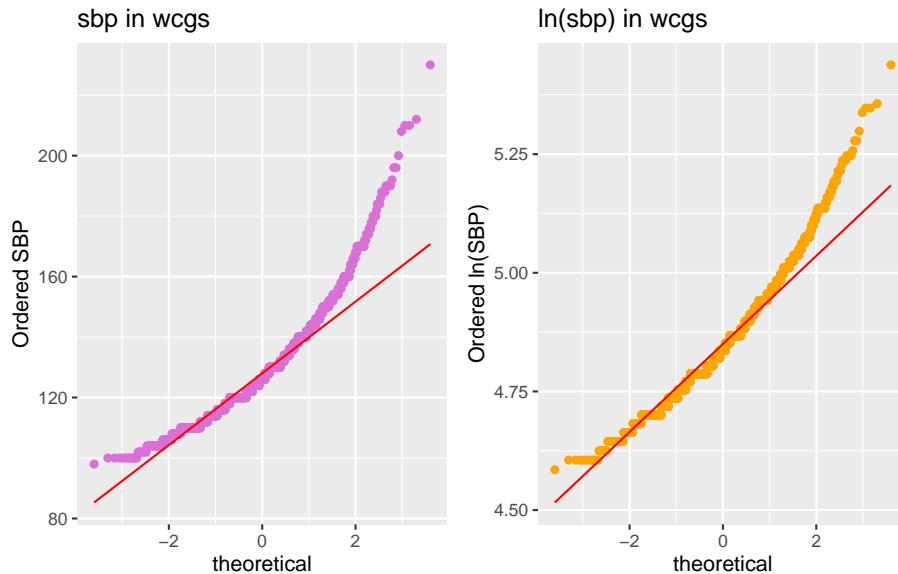
We can also look at Normal Q-Q plots, for instance...

```
p1 <- ggplot(wcgs, aes(sample = sbp)) +
  geom_qq(color = "orchid") +
  geom_qq_line(color = "red") +
  labs(y = "Ordered SBP", title = "sbp in wcgs")

p2 <- ggplot(wcgs, aes(sample = lnsbp)) +
  geom_qq(color = "orange") +
  geom_qq_line(color = "red") +
  labs(y = "Ordered ln(SBP)", title = "ln(sbp) in wcgs")

## next step requires library(patchwork)

p1 + p2 +
  plot_annotation(title = "Normal Q-Q plots of SBP and ln(SBP) in wcgs")
```

Normal Q–Q plots of SBP and $\ln(\text{SBP})$ in wcgs

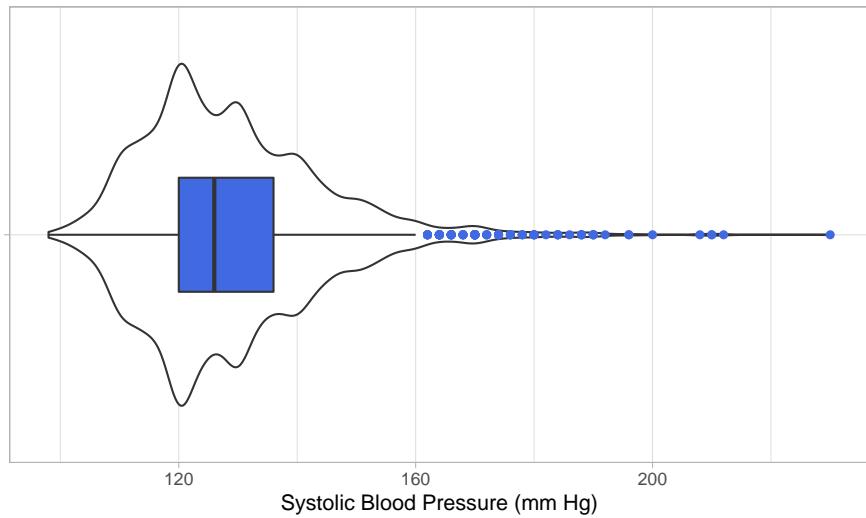
There's at best a small improvement from `sbp` to `lnsbp` in terms of approximation by a Normal distribution.

13.3 Describing Outlying Values with Z Scores

It looks like there's an outlier (or a series of them) in the SBP data.

```
ggplot(wcgs, aes(x = "", y = sbp)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "royalblue",
               outlier.color = "royalblue") +
  labs(title = "Boxplot with Violin of SBP in `wcgs` data",
       y = "Systolic Blood Pressure (mm Hg)",
       x = "") +
  theme_light() +
  coord_flip()
```

Boxplot with Violin of SBP in `wcgs` data



```
wcgs %$% Hmisc::describe(sbp)
```

sbp	n	missing	distinct	Info	Mean	Gmd	.05	.10
3154	0	62	0.996	128.6	16.25	110	112	
.25	.50	.75	.90	.95				
120	126	136	148	156				

lowest : 98 100 102 104 106, highest: 200 208 210 212 230

The maximum value here is 230, and is clearly the most extreme value in the data set. One way to gauge this is to describe that observation's **Z score**, the number of standard deviations away from the mean that the observation falls. Here, the maximum value, 230 is 6.71 standard deviations above the mean, and thus has a Z score of 6.7.

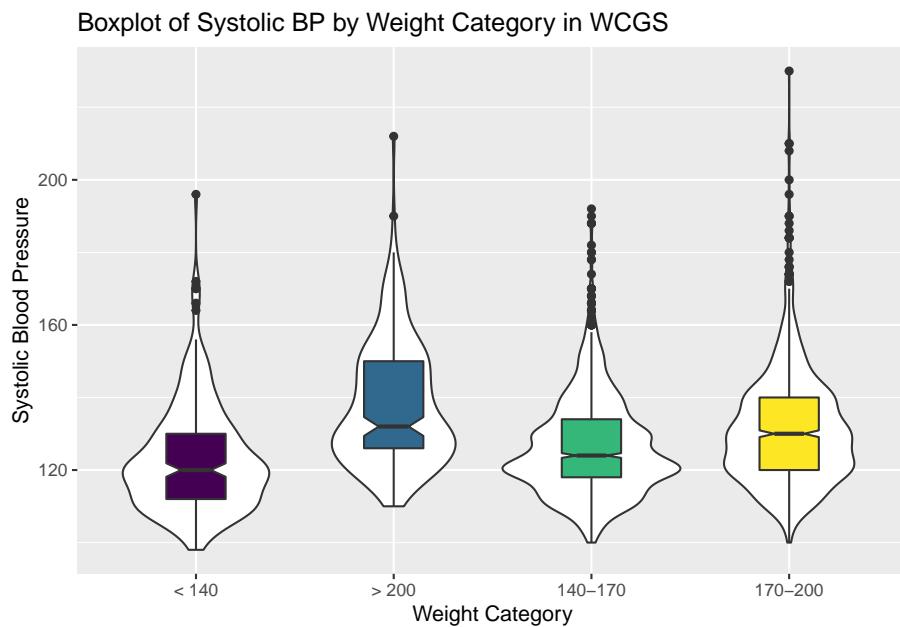
A negative Z score would indicate a point below the mean, while a positive Z score indicates, as we've seen, a point above the mean. The minimum systolic blood pressure, 98 is 2.03 standard deviations *below* the mean, so it has a Z score of -2.

Recall that the Empirical Rule suggests that if a variable follows a Normal distribution, it would have approximately 95% of its observations falling inside a Z score of (-2, 2), and 99.74% falling inside a Z score range of (-3, 3). Do the systolic blood pressures appear Normally distributed?

13.4 Does Weight Category Relate to SBP?

The data are collected into four groups based on the subject's weight (in pounds).

```
ggplot(wcgs, aes(x = wghtcat, y = sbp)) +
  geom_violin() +
  geom_boxplot(aes(fill = wghtcat), width = 0.3, notch = TRUE) +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Boxplot of Systolic BP by Weight Category in WCGS",
       x = "Weight Category", y = "Systolic Blood Pressure")
```



13.5 Re-Leveling a Factor

Well, that's not so good. We really want those weight categories (the *levels*) to be ordered more sensibly.

```
wcgs %>% tabyl(wghtcat)
```

wghtcat	n	percent
< 140	232	0.0736
> 200	213	0.0675
140-170	1538	0.4876
170-200	1171	0.3713

Like all *factor* variables in R, the categories are specified as levels.

```
levels(wcgs$wghtcat)
```

```
[1] "< 140"    "> 200"    "140-170" "170-200"
```

We want to change the order of the levels in a new version of this factor variable so they make sense. There are multiple ways to do this, but I prefer the `fct_relevel` function from the `forcats` package. Which order is more appropriate?

I'll add a new variable to the `wcgs` data called `weight_f` that relevels the `wghtcat` data.

```
wcgs <- wcgs %>%
  mutate(weight_f = fct_relevel(wghtcat, "< 140", "140-170", "170-200", "> 200"))

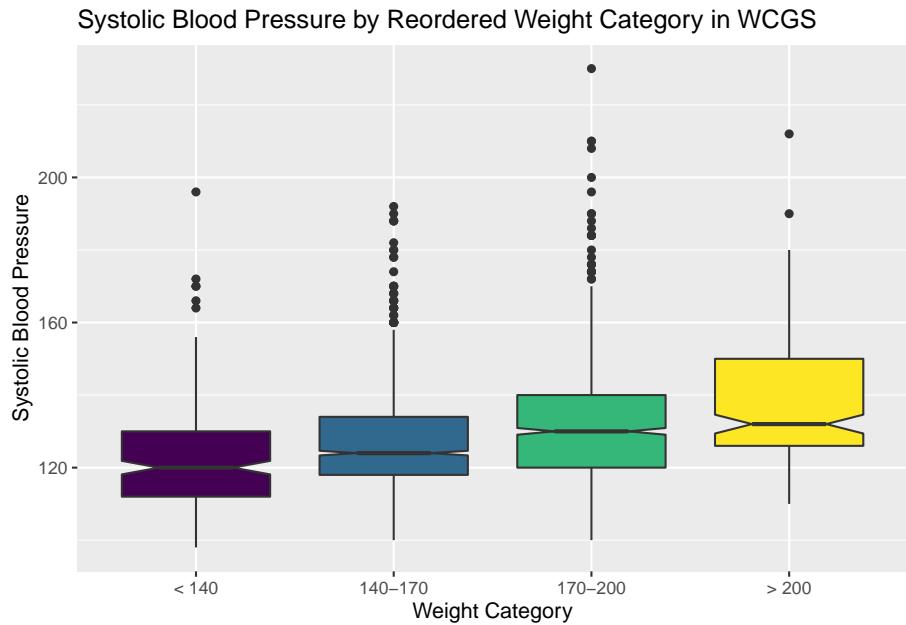
wcgs %>% tabyl(weight_f)
```

weight_f	n	percent
< 140	232	0.0736
140-170	1538	0.4876
170-200	1171	0.3713
> 200	213	0.0675

For more on the `forcats` package, check out Golemund and Wickham (2019), especially the Section on Factors.

13.5.1 SBP by Weight Category

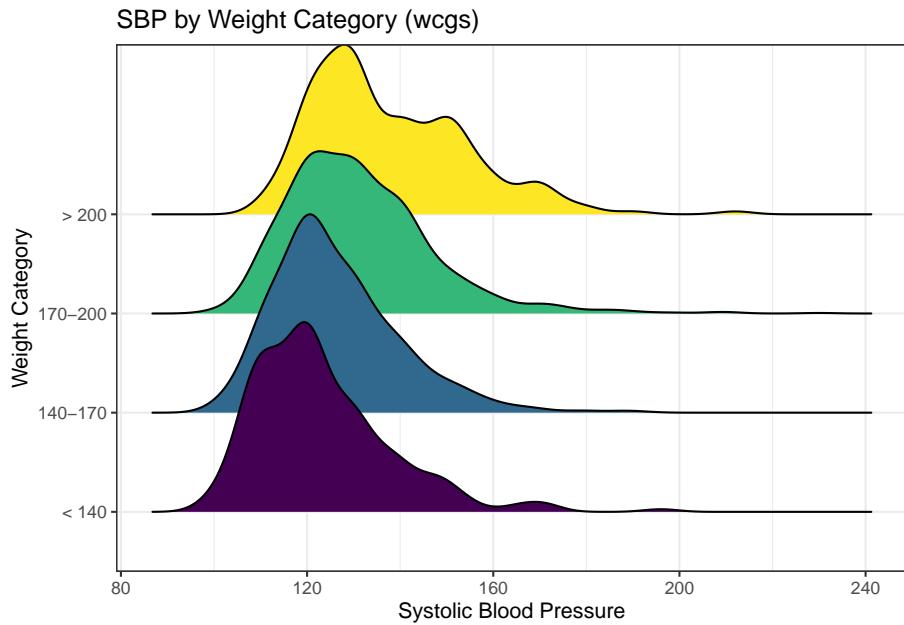
```
ggplot(wcgs, aes(x = weight_f, y = sbp, fill = weight_f)) +
  geom_boxplot(notch = TRUE) +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Systolic Blood Pressure by Reordered Weight Category in WCGS",
       x = "Weight Category", y = "Systolic Blood Pressure")
```



We might see some details well with a **ridgeline plot**, too.

```
wcgs %>%
  ggplot(aes(x = sbp, y = weight_f, fill = weight_f, height = ..density..)) +
  ggridges::geom_density_ridges(scale = 2) +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "SBP by Weight Category (wcgs)",
       x = "Systolic Blood Pressure",
       y = "Weight Category") +
  theme_bw()
```

Picking joint bandwidth of 3.74



As the plots suggest, patients in the heavier groups generally had higher systolic blood pressures.

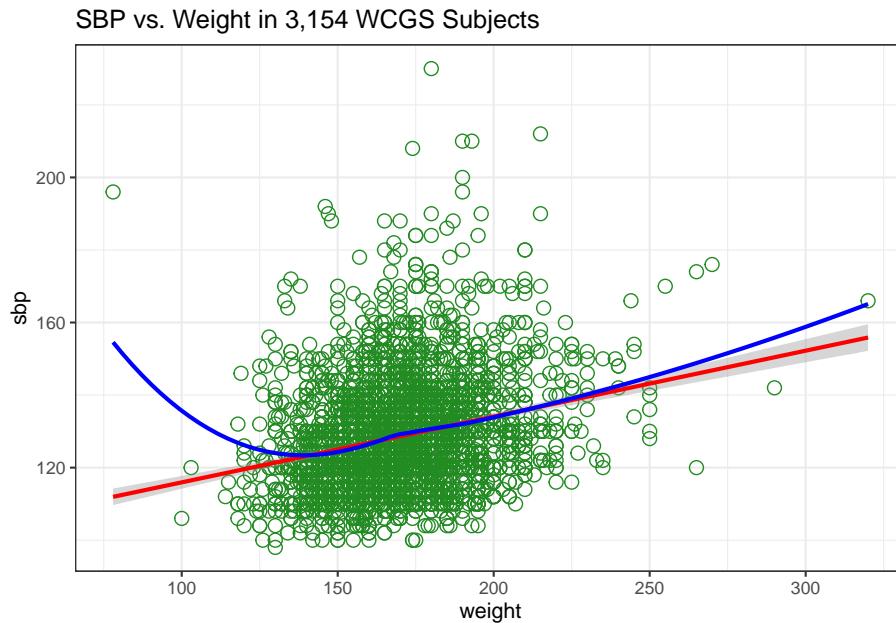
```
mosaic::favstats(sbp ~ weight_f, data = wcgs)
```

	weight_f	min	Q1	median	Q3	max	mean	sd	n	missing
1	< 140	98	112	120	130	196	123	14.7	232	0
2	140-170	100	118	124	134	192	126	13.7	1538	0
3	170-200	100	120	130	140	230	131	15.6	1171	0
4	> 200	110	126	132	150	212	138	16.8	213	0

13.6 Are Weight and SBP Linked?

Let's build a scatter plot of SBP (Outcome) by Weight (Predictor), rather than breaking down into categories.

```
ggplot(wcgs, aes(x = weight, y = sbp)) +
  geom_point(size=3, shape=1, color="forestgreen") + ## default size = 2
  stat_smooth(method=lm, color="red") + ## add se=FALSE to hide conf. interval
  stat_smooth(method=loess, se=FALSE, color="blue") +
  ggtitle("SBP vs. Weight in 3,154 WCGS Subjects") +
  theme_bw()
```



- The mass of the data is hidden from us - showing 3154 points in one plot can produce little more than a blur where there are lots of points on top of each other.
- Here the least squares regression line (in red), and loess smoother, (in blue) can help.

The relationship between systolic blood pressure and weight appears to be very close to linear, but of course there is considerable scatter around that generally linear relationship. It turns out that the Pearson correlation of these two variables is 0.253.

13.7 SBP and Weight by Arcus Senilis groups?

An issue of interest to us will be to assess whether the SBP-Weight relationship we see above is similar among subjects who have arcus senilis and those who do not.

Arcus senilis is an old age syndrome where there is a white, grey, or blue opaque ring in the corneal margin (peripheral corneal opacity), or white ring in front of the periphery of the iris. It is present at birth but then fades; however, it is quite commonly present in the elderly. It can also appear earlier in life as a result of hypercholesterolemia.

Wikipedia article on Arcus Senilis, retrieved 2017-08-15

Let's start with a quick look at the `arcus` data.

```
wcgs %>% tabyl(arcus)
```

	n	percent	valid_percent
0	2211	0.701015	0.701
1	941	0.298351	0.299
NA	2	0.000634	NA

We have 2 missing values, so we probably want to do something about that before plotting the data, and we may also want to create a factor variable with more meaningful labels than 1 (which means yes, arcus senilis is present) and 0 (which means no, it isn't.)

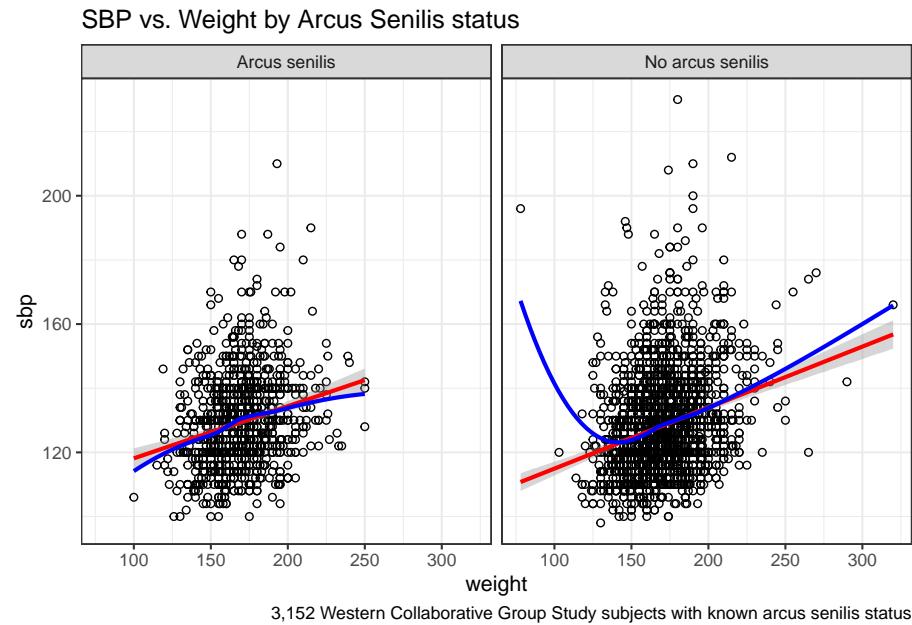
```
wcgs <- wcgs %>%
  mutate(arcus_f = fct_recode(factor(arcus),
    "Arcus senilis" = "1",
    "No arcus senilis" = "0"),
  arcus_f = fct_relevel(arcus_f, "Arcus senilis"))

wcgs %>% tabyl(arcus_f, arcus)
```

	0	1	NA_
Arcus senilis	0	941	0
No arcus senilis	2211	0	0
<NA>	0	0	2

Let's build a version of the `wcgs` data that eliminates all missing data in the variables of immediate interest, and then plot the SBP-weight relationship in groups of patients with and without arcus senilis.

```
wcgs %>%
  filter(complete.cases(arcus_f, sbp, weight)) %>%
  ggplot(aes(x = weight, y = sbp, group = arcus_f)) +
  geom_point(shape = 1) +
  stat_smooth(method=lm, color="red") +
  stat_smooth(method=loess, se=FALSE, color="blue") +
  labs(title = "SBP vs. Weight by Arcus Senilis status",
    caption = "3,152 Western Collaborative Group Study subjects with known arcus s") +
  facet_wrap(~ arcus_f) +
  theme_bw()
```



13.8 Linear Model for SBP-Weight Relationship: subjects without Arcus Senilis

```
model.noarcus <-
  lm(sbp ~ weight, data = filter(wcgs, arcus == 0))

summary(model.noarcus)
```

Call:
`lm(formula = sbp ~ weight, data = filter(wcgs, arcus == 0))`

Residuals:

Min	1Q	Median	3Q	Max
-29.01	-10.25	-2.45	7.55	99.85

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	95.9219	2.5552	37.5	<2e-16 ***
weight	0.1902	0.0149	12.8	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
Residual standard error: 14.8 on 2209 degrees of freedom
Multiple R-squared:  0.0687,    Adjusted R-squared:  0.0683
F-statistic: 163 on 1 and 2209 DF,  p-value: <2e-16
```

The linear model for the 2211 patients without Arcus Senilis has $R^2 = 6.87\%$.

- The regression equation is $95.92 - 0.19 \text{ weight}$, for those patients without Arcus Senilis.

13.9 Linear Model for SBP-Weight Relationship: subjects with Arcus Senilis

```
model.witharcus <-
  lm(sbp ~ weight, data = filter(wcgs, arcus == 1))

summary(model.witharcus)
```

```
Call:
lm(formula = sbp ~ weight, data = filter(wcgs, arcus == 1))
```

Residuals:

Min	1Q	Median	3Q	Max
-30.34	-9.64	-1.96	7.97	76.74

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	101.879	3.756	27.13	< 2e-16 ***							
weight	0.163	0.022	7.39	3.3e-13 ***							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'. '	0.1	' '	1

```
Residual standard error: 14.2 on 939 degrees of freedom
Multiple R-squared:  0.0549,    Adjusted R-squared:  0.0539
F-statistic: 54.6 on 1 and 939 DF,  p-value: 3.29e-13
```

The linear model for the 941 patients with Arcus Senilis has $R^2 = 5.49\%$.

- The regression equation is $101.88 - 0.163 \text{ weight}$, for those patients with Arcus Senilis.

13.10 Including Arcus Status in the model

```
model3 <- lm(sbp ~ weight * arcus, data = filter(wcgs, !is.na(arcus)))

summary(model3)
```

```
Call:
lm(formula = sbp ~ weight * arcus, data = filter(wcgs, !is.na(arcus)))

Residuals:
    Min      1Q  Median      3Q     Max 
-30.34 -10.15  -2.35   7.67  99.85 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  95.9219    2.5244   38.00  <2e-16 ***
weight       0.1902    0.0147   12.92  <2e-16 ***
arcus        5.9566    4.6197    1.29    0.20    
weight:arcus -0.0276    0.0270   -1.02    0.31    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.6 on 3148 degrees of freedom
Multiple R-squared:  0.066, Adjusted R-squared:  0.0651 
F-statistic: 74.1 on 3 and 3148 DF,  p-value: <2e-16
```

The actual regression equation in this setting includes both weight, and an indicator variable (1 = yes, 0 = no) for arcus senilis status, and the product of weight and that 1/0 indicator.

- Note the use of the product term `weight*arcus` in the setup of the model to allow both the slope of weight and the intercept term in the model to change depending on arcus senilis status.
 - For a patient who has arcus, the regression equation is $SBP = 95.92 + 0.19 \text{ weight} + 5.96 (1) - 0.028 \text{ weight} (1) = 101.88 + 0.162 \text{ weight}$.
 - For a patient without arcus senilis, the regression equation is $SBP = 95.92 + 0.19 \text{ weight} + 5.96 (0) - 0.028 \text{ weight} (0) = 95.92 + 0.19 \text{ weight}$.

The linear model including the interaction of weight and arcus to predict sbp for the 3152 patients with known Arcus Senilis status has $R^2 = 6.6\%$.

13.11 Predictions from these Linear Models

What is our predicted SBP for a subject weighing 175 pounds?

How does that change if our subject weighs 200 pounds?

Recall that

- *Without* Arcus Senilis, linear model for SBP = $95.9 + 0.19 \times \text{weight}$
- *With* Arcus Senilis, linear model for SBP = $101.9 + 0.16 \times \text{weight}$

So the predictions for a 175 pound subject are:

- $95.9 + 0.19 \times 175 = 129$ mm Hg without Arcus Senilis, and
- $101.9 + 0.16 \times 175 = 130$ mm Hg with Arcus Senilis.

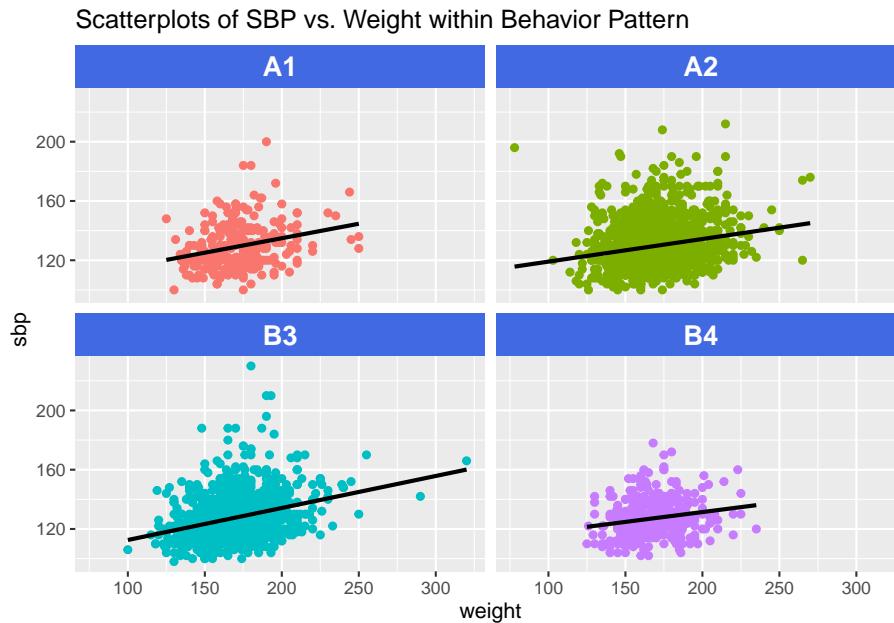
And thus, the predictions for a 200 pound subject are:

- $95.9 + 0.19 \times 200 = 134$ mm Hg without Arcus Senilis, and
- $101.9 + 0.16 \times 200 = 134.4$ mm Hg with Arcus Senilis.

13.12 Scatterplots with Facets Across a Categorical Variable

We can use facets in `ggplot2` to show scatterplots across the levels of a categorical variable, like `behpat`.

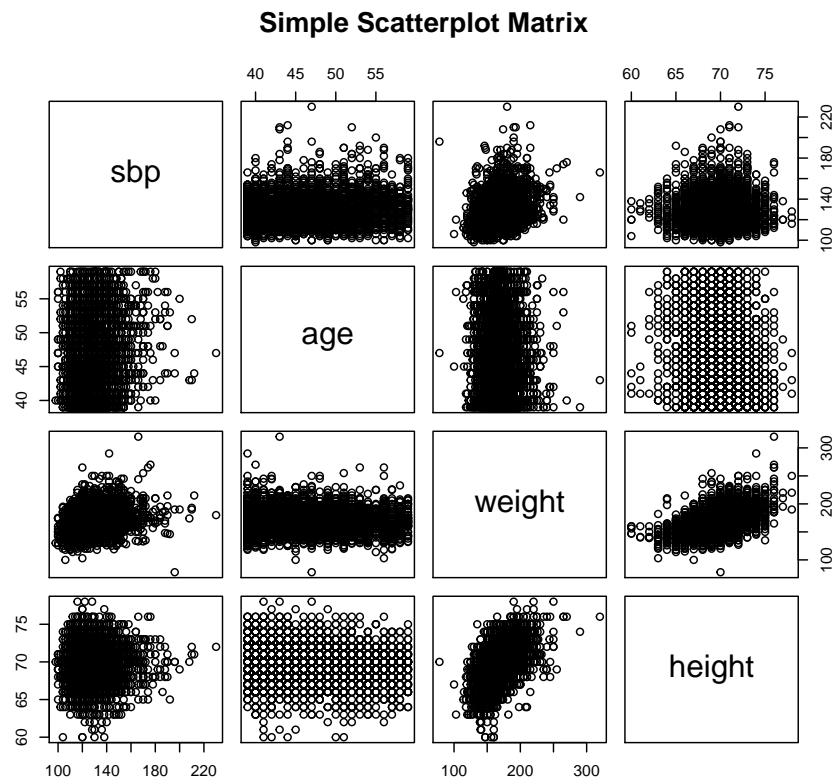
```
ggplot(wcgs, aes(x = weight, y = sbp, col = behpat)) +
  geom_point() +
  facet_wrap(~ behpat) +
  geom_smooth(method = "lm", se = FALSE, col = "black") +
  guides(color = FALSE) +
  theme(strip.text = element_text(face="bold", size=rel(1.25), color="white"),
        strip.background = element_rect(fill="royalblue")) +
  labs(title = "Scatterplots of SBP vs. Weight within Behavior Pattern")
```



13.13 Scatterplot and Correlation Matrices

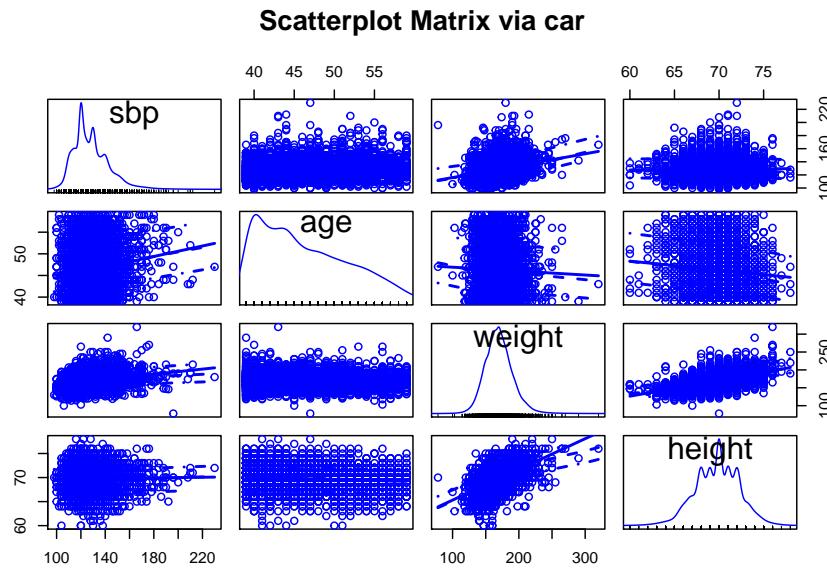
A **scatterplot matrix** can be very helpful in understanding relationships between multiple variables simultaneously. There are several ways to build such a thing, including the `pairs` function...

```
pairs (~ sbp + age + weight + height, data=wcgs, main="Simple Scatterplot Matrix")
```



13.13.1 Using the car package

Or, we can use the `scatterplotMatrix` function from the `car` package, which adds some detail and fitting to the plots, and places density estimates (with rug plots) on the diagonals.



13.13.2 Displaying a Correlation Matrix

```
wcgs %>%
  dplyr::select(sbp, age, weight, height) %>%
  cor() %>% # obtain correlation coefficients for this subgroup
  signif(., 3) # round them off to three significant figures before printing
```

	sbp	age	weight	height
sbp	1.0000	0.1660	0.2530	0.0184
age	0.1660	1.0000	-0.0344	-0.0954
weight	0.2530	-0.0344	1.0000	0.5330
height	0.0184	-0.0954	0.5330	1.0000

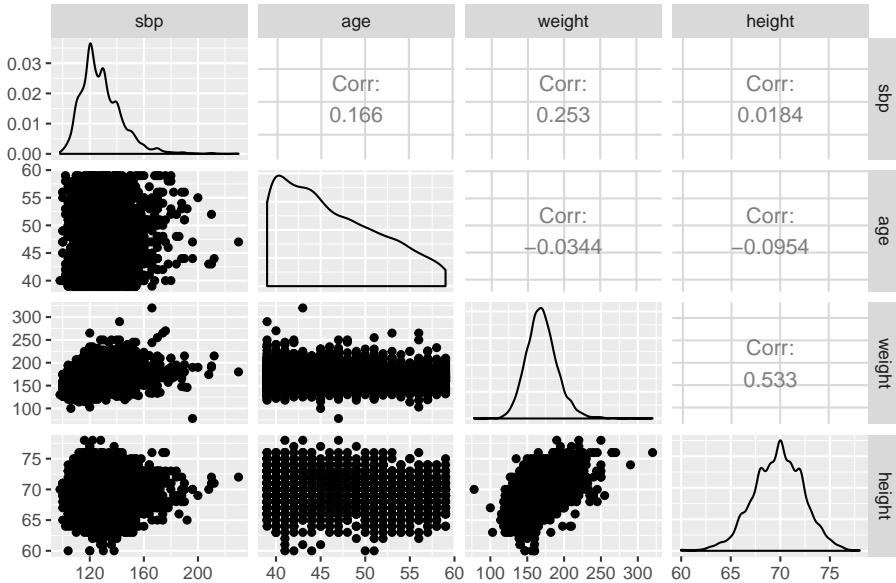
13.13.3 Using the GGally package

The `ggplot2` system doesn't have a built-in scatterplot system. There are some nice add-ins in the world, though. One option I sort of like is in the `GGally` package, which can produce both correlation matrices and scatterplot matrices.

The `ggpairs` function provides a density plot on each diagonal, Pearson correlations on the upper right and scatterplots on the lower left of the matrix.

```
GGally::ggpairs(wcgs %>% select(sbp, age, weight, height),  
                 title = "Scatterplot Matrix via ggpairs")
```

Scatterplot Matrix via ggpairs



Chapter 14

Part A: A Few of the Key Points

14.1 Key Graphical Descriptive Summaries for Quantitative Data

- **Histograms** and their variants, including smooth density curves, and normal density functions based on the sample mean and sample standard deviation
- **Boxplots** and the like, including ridgeline plots and violin plots, that show more of the distribution in a compact format that is especially useful for comparisons
- **Normal QQ Plots** which are plots of the ordered data (technically, the order statistics) against certain quantiles of the Normal distribution - show curves to indicate skew, and “S” shaped arcs to indicate seriously heavy- or light-tailed distributions compared to the Normal.

14.2 Key Numerical Descriptive Summaries for Quantitative Data

- Measures of *Location* (including Central Tendency), such as the **mean**, **median**, **quantiles** and even the **mode**.
- Measures of *Spread*, including the **range**, **IQR** (which measures the variability of points near the center of the distribution), **standard deviation** (which is less appropriate as a summary measure if the data show substantial skew or heavy-tailedness), **variance**, **standard error**, **median**

absolute deviation (which is less affected by outlying values in the tails of the distribution than a standard deviation).

- I'll mention the **coefficient of variation** (ratio of the standard deviation to the mean, expressed as a percentage, note that this is only appropriate for variables that take only positive values.)
- One Key Measure of *Shape* is nonparametric skew (skew_1), which can be used to help confirm plot-based decisions about data shape.

14.3 The Empirical Rule - Interpreting a Standard Deviation

If the data are approximately Normally distributed, then the mean and median will be very similar, and there will be minimal skew and no large outlier problem.

Should this be the case, the mean and standard deviation describe the distribution well, and the **Empirical Rule** will hold reasonably well.

If the data are (approximately) Normally distributed, then

- About 68% of the data will fall within one standard deviation of the mean
- Approximately 95% of the data will fall within two standard deviations of the mean
- Approximately 99.7% of the data will fall within three standard deviations of the mean.

14.4 Identifying “Outliers” Using Fences and/or Z Scores

- Distributions can be symmetric, but still not Normally distributed, if they are either outlier-prone (heavy-tailed) or light-tailed.
- Outliers can have an important impact on other descriptive measures.
- John Tukey described **fences** which separated non-outlier from outlier values in a distribution. Generally, the fences are set 1.5 IQR away from the 25th and 75th percentiles in a boxplot.
- Or, we can use **Z scores** to highlight the relationship between values and what we might expect if the data were normally distributed.
- The Z score for an individual value is that value minus the data's mean, all divided by the data's standard deviation.
- If the data are normally distributed, we'd expect all but 5% of its observations to have Z scores between -2 and +2, for example.

14.5 Summarizing Bivariate Associations: Scatterplots and Regression Lines

- The most important tools are various **scatterplots**, often accompanied by **regression lines** estimated by the method of least squares, and by (loess) **smooths** which permit local polynomial functions to display curved relationships.
- In a multivariate setting, we will occasionally consider plots in the form of a **scatterplot matrix** to enable simultaneous comparisons of multiple two-way associations.
- We fit linear models to our data using the `lm` function, and we evaluate the models in terms of their ability to predict an outcome given a predictor, and through R^2 , which is interpreted as the proportion of variation in the outcome accounted for by the model.

14.6 Summarizing Bivariate Associations With Correlations

- **Correlation coefficients**, of which by far the most commonly used is the **Pearson correlation**, which is a unitless (scale-free) measure of bivariate linear association for the variables X and Y, symbolized by r , and ranging from -1 to +1. The Pearson correlation is a function of the slope of the least squares regression line, divided by the product of the standard deviations of X and Y.
- Also relevant to us is the **Spearman rank correlation coefficient**, which is obtained by using the usual formula for a Pearson correlation, but on the ranks (1 = minimum, n = maximum, with average ranks applied to the ties) of the X and Y values. This approach (running a correlation of the orderings of the data) substantially reduces the effect of outliers. The result still ranges from -1 to +1, with 0 indicating no monotone association.

Part B. Making Comparisons

Chapter 15

Introduction to Part B

15.1 Point Estimation and Confidence Intervals

The basic theory of estimation can be used to indicate the probable accuracy and potential for bias in estimating based on limited samples. A point estimate provides a single best guess as to the value of a population or process parameter.

A confidence interval is a particularly useful way to convey to people just how much error one must allow for in a given estimate. In particular, a confidence interval allows us to quantify just how close we expect, for instance, the sample mean to be to the population or process mean. The computer will do the calculations; we need to interpret the results.

The key tradeoffs are cost vs. precision, and precision vs. confidence in the correctness of the statement. Often, if we are dissatisfied with the width of the confidence interval and want to make it smaller, we have little choice but to reconsider the sample – larger samples produce shorter intervals.

15.2 One-Sample Confidence Intervals

Very often, sample data indicate that something has happened – a change in the proportion, a shift in the mean, etc. Before we get excited, it's worth checking whether the apparent result might possibly be the result of random sampling error. The next few classes will be devoted to ideas of testing—seeing whether an apparent result might possibly be attributable to sheer randomness. Confidence intervals provide a way to assess this chance.

Statistics provides a number of tools for reaching an informed choice (informed by sample information, of course.) Which tool, or statistical method, to use

depends on various aspects of the problem at hand.

15.3 Comparing Two Groups

In making a choice between two alternatives, questions such as the following become paramount.

- Is there a status quo?
- Is there a standard approach?
- What are the costs of incorrect decisions?
- Are such costs balanced?

The process of comparing the means/medians/proportions/rates of the populations represented by two independently obtained samples can be challenging, and such an approach is not always the best choice. Often, specially designed experiments can be more informative at lower cost (i.e. smaller sample size). As one might expect, using these more sophisticated procedures introduces trade-offs, but the costs are typically small relative to the gain in information.

When faced with such a comparison of two alternatives, a test based on **paired** data is often much better than a test based on two distinct, independent samples. Why? If we have done our experiment properly, the pairing lets us eliminate background variation that otherwise hides meaningful differences.

15.3.1 Model-Based Comparisons and ANOVA/Regression

Comparisons based on independent samples of quantitative variables are also frequently accomplished through other equivalent methods, including the analysis of variance approach and dummy variable regression, both of which produce identical confidence intervals to the pooled variance t test for the same comparison.

We will also discuss some of the main ideas in developing, designing and analyzing statistical experiments, specifically in terms of making comparisons. The ideas we will present in this section allow for the comparison of more than two populations in terms of their population means. The statistical techniques employed analyze the sample variance in order to test and estimate the population means and for this reason the method is called the analysis of variance (ANOVA), and we will discuss this approach alone, and within the context of a linear regression model using dummy or indicator variables.

15.4 Special Tools for Categorical Data

We will also turn briefly to some methods for dealing with qualitative, categorical variables. In particular, we begin with a test of how well the frequencies of various categories fit a theoretical set of probabilities. We also consider a test for the relation between two qualitative variables. We'll examine some of the key measures used in describing such relationships, like odds ratios and relative risks.

15.5 Our First Three Studies

We'll focus, for a while, on three studies, and the next few Chapters in these Notes summarize each of them, graphically and numerically, and then use them to demonstrate various approaches to making inferences.

- The Serum Zinc study, which uses a single sample of quantitative data.
- A randomized controlled trial comparing ibuprofen vs. placebo in patients with sepsis, which uses an *independent samples* design to compare two samples of quantitative data.
- The Lead in the Blood of Children study, which uses a *paired samples* design to compare two samples of quantitative data.

15.6 Data Sets used in Part B

```
serzinc <- read_csv("data/serzinc.csv")
bloodlead <- read_csv("data/bloodlead.csv")
sepsis <- read_csv("data/sepsis.csv")
specialty <- read_csv("data/specialty.csv")
battery <- read.csv("data/battery.csv") %>% tbl_df
breakfast <- read.csv("data/breakfast.csv") %>% tbl_df
nyfs2 <- read.csv("data/nyfs2.csv") %>% tbl_df
survey1 <- read.csv("data/surveyday1.csv") %>% tbl_df
active2x3 <- read.csv("data/active2x3.csv") # deliberately NOT a tibble
darwin <- read.csv("data/darwin.csv") %>% tbl_df
```

15.7 The Love-boost.R script

In this part of the course, we'll also make use of a few scripts I've gathered for you.

```
source("data/Love-boost.R")
```

Chapter 16

The Serum Zinc Study

16.1 Serum Zinc Levels in 462 Teenage Males (`serzinc`)

The `serzinc` data include serum zinc levels in micrograms per deciliter that have been gathered for a sample of 462 males aged 15-17. My source for these data is Appendix B1 of Pagano and Gauvreau (2000). Serum zinc deficiency has been associated with anemia, loss of strength and endurance, and it is thought that 25% of the world's population is at risk of zinc deficiency. Such a deficiency can indicate poor nutrition, and can affect growth and vision, for instance. "Typical" values¹ are said to be 0.66-1.10 mcg/ml, which is 66 - 110 micrograms per deciliter.

```
serzinc
```

```
# A tibble: 462 x 2
  ID      zinc
  <chr> <dbl>
1 M-001   142
2 M-002    88
3 M-003    83
4 M-004   100
5 M-005   123
6 M-006    63
7 M-007   102
8 M-008    80
9 M-009   117
10 M-010   86
```

¹Reference values for those over the age of 10 years at <http://www.mayomedicallaboratories.com/test-catalog/Clinical+and+Interpretive/8620> , visited 2019-09-17.

```
# ... with 452 more rows
```

16.2 Our Goal: A Confidence Interval for the Population Mean

After we assess the data a bit, and are satisfied that we understand it, our first inferential goal will be to produce a **confidence interval for the true (population) mean** of males age 15-17 based on this sample, assuming that these 462 males are a random sample from the population of interest, that each serum zinc level is drawn independently from an identical distribution describing that population.

To do this, we will have several different procedures available, including:

1. A confidence interval for the population mean based on a t distribution, when we assume that the data are drawn from an approximately Normal distribution, using the sample standard deviation. (Interval corresponding to a t test, and it will be a good choice when the data really are approximately Normally distributed.)
2. A resampling approach to generate a bootstrap confidence interval for the population mean, which does not require that we assume either that the population standard deviation is known, nor that the data are drawn from an approximately Normal distribution, but which has some other weaknesses.
3. A rank-based procedure called the Wilcoxon signed rank test can also be used to yield a confidence interval statement about the population pseudo-median, a measure of the population distribution's center (but not the population's mean).

16.3 Exploratory Data Analysis for Serum Zinc

16.3.1 Graphical Summaries

The code presented below builds:

- a histogram (with Normal model superimposed),
- a boxplot (with median notch) and
- a Normal Q-Q plot (with guiding straight line through the quartiles)

for the `zinc` results from the `serzinc` tibble. It does this while making use of several functions contained in the script `Love-boost.R`.

These functions include:

- `fd_bins` to estimate the Freedman-Diaconis bins setting for the histogram

- `qq_int` and `qq_slope` to facilitate the drawing of a line on the Normal Q-Q plot

```

rezinc <- mosaic:::favstats(~ zinc, data = serzinc)
bin_w1 <- 10 # specify binwidth

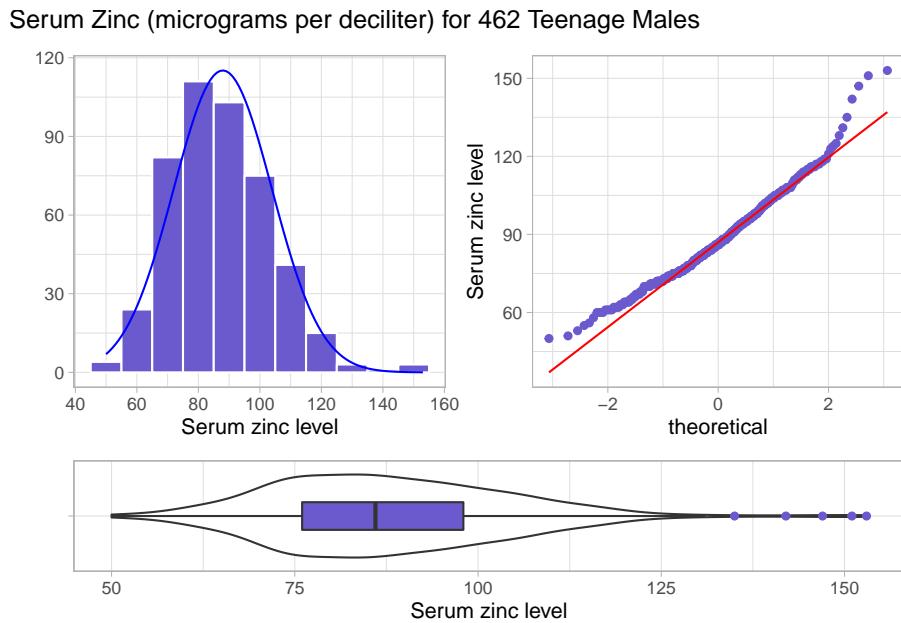
p1 <- ggplot(serzinc, aes(x = zinc)) +
  geom_histogram(binwidth = bin_w1,
                 fill = "slateblue",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = rezinc$mean,
                            sd = rezinc$sd) *
      rezinc$n * bin_w1,
    col = "blue") +
  labs(x = "Serum zinc level", y = "")

p2 <- ggplot(serzinc, aes(sample = zinc)) +
  geom_qq(col = "slateblue") +
  geom_qq_line(col = "red") +
  theme_light() +
  labs(y = "Serum zinc level")

p3 <- ggplot(serzinc, aes(x = "", y = zinc)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "slateblue",
               outlier.color = "slateblue") +
  theme_light() +
  coord_flip() +
  labs(x = "", y = "Serum zinc level")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Serum Zinc (micrograms per deciliter) for 462 Teenage Males")

```



These results include some of the more useful plots and numerical summaries when assessing shape, center and spread. The `zinc` data in the `serzinc` data frame appear to be slightly right skewed, with five outlier values on the high end of the scale, in particular.

You could potentially add `coord_flip() +` to the histogram, and this would have the advantage of getting all three plots oriented in the same direction, but then we (or at least I) lose the ability to tell the direction of skew at a glance from the direction of the histogram.

16.3.2 Numerical Summaries

This section describes some numerical summaries of interest to augment the plots in summarizing the center, spread and shape of the distribution of serum zinc among these 462 teenage males.

The tables below are built using two functions from the `Love-boost.R` script.

- `skew1` provides the `skew1` value for the `zinc` data and
- `Emp_Rule` provides the results of applying the 68-95-99.7 Empirical Rule to the `zinc` data.

```
mosaic::favstats(~ zinc, data = serzinc)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	50	76	86	98	153	87.9	16	462	0

```
serzinc %$% skew1(zinc)
```

```
[1] 0.121
```

The `skew1` value backs up our graphical assessment, that the data are slightly right skewed.

We can also assess how well the 68-95-99.7 Empirical Rule for a Normal distribution holds up for these data. Not too badly, as it turns out.

```
serzinc %$% Emp_Rule(zinc)
```

	count	proportion
Mean +/- 1 SD	323	0.699
Mean +/- 2 SD	447	0.968
Mean +/- 3 SD	458	0.991
Entire Data Set	462	1.000

```
serzinc %$% psych::describe(zinc)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	462	87.9	16	86	87.2	16.3	50	153	103	0.62	0.87	0.74

Rounded to two decimal places, the standard deviation of the serum zinc data turns out to be 16, and so the standard error of the mean, shown as `se` in the `psych::describe` output, is 16 divided by the square root of the sample size, $n = 462$. This standard error is about to become quite important to us in building statistical inferences about the mean of the entire population of teenage males based on this sample.

16.4 Inference about a Population Mean from this Sample

The notion of a confidence interval will be important to us, and is explained in further detail in Chapter 17. For now, we'll focus on building a confidence interval for the mean of a population or process.

A confidence interval for a population or process mean uses data from a sample (and perhaps some additional information) to identify a range of potential values for the population mean, which, if certain assumptions hold, can be assumed to provide a reasonable estimate for the true population mean. A confidence interval consists of:

1. An interval estimate describing the population parameter of interest (here the population mean), and
2. A probability statement, expressed in terms of a confidence level.

Now, let's identify a 90% confidence interval for the mean of the population zinc levels for teenagers like those sampled in this scenario. Under certain assumptions we'll discuss later, a reasonable approach uses an intercept-only linear model to estimate the population mean zinc level, both in terms of a *point estimate* and a *confidence interval* for a given confidence level.

```
model_zinc <- lm(zinc ~ 1, data = serzinc)

coef(model_zinc) # for the point estimate of pop. mean

(Intercept)
87.9

confint(model_zinc, conf = 0.90) # establish 90% CI for pop. mean

2.5 % 97.5 %
(Intercept) 86.5 89.4
```

Equivalently, we could use the `tidy()` from the `broom` package to obtain these results within a tibble for our intercept-only model.

```
# requires library(broom)
tidy(model_zinc, conf.int = TRUE, conf = 0.90)

# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept) 87.9      0.745    118.       0     86.7     89.2
```

This method is identical to estimating a confidence interval for the population mean using the `t.test` function in R. Here are those results.

```
tidy(t.test(serzinc$zinc, conf.level = 0.90))

# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
  <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl> <chr>
1 87.9      118.      0     461     86.7     89.2 One S~
# ... with 1 more variable: alternative <chr>
```

In Chapter 17, we will discuss the development of this confidence interval, and compare this approach to several alternatives.

16.5 Comparison to “Normal” Zinc Levels

Recall that the “Normal” zinc level would be between 66 and 110. What percentage of the sampled 462 teenagers meet that standard?

There are at least two ways to get what we need.

```
serzinc %>%
  count(zinc > 65 & zinc < 111) %>%
  mutate(proportion = n / sum(n), percentage = 100 * n / sum(n))

# A tibble: 2 x 4
`zinc > 65 & zinc < 111`    n proportion percentage
<lgl>                      <int>      <dbl>        <dbl>
1 FALSE                      67       0.145       14.5
2 TRUE                       395       0.855       85.5
```

Or, we could combine the “explosion pipe” with `tabyl` to obtain:

```
serzinc %$% tabyl(zinc > 65 & zinc < 111) %>%
  adorn_pct_formatting()
```

zinc > 65 & zinc < 111	n	percent
FALSE	67	14.5%
TRUE	395	85.5%

16.6 Inference about a Population Proportion from this Sample

Previously, we estimated a confidence interval for the *mean* of the population zinc levels. Now, we want to estimate a confidence interval for the *proportion* of the population whose serum zinc levels are in the range of 66 to 110. We’ll discuss this further in Chapter 18, but for now, we’ll just provide a little insight.

As before, we want to build both a point estimate for the population proportion, and a confidence interval for the population proportion.

Now, let’s identify a 90% confidence interval for the proportion of the population whose zinc levels are within the “normal” range. We have seen that 395 / 462 subjects (or a proportion of 0.855) fall in the “normal range” in our sample. For now, that will also be our *point estimate* of the proportion in the “normal range” across the entire population of teenagers like those in our sample.

```
serzinc <- serzinc %>%
  mutate(in_range = ifelse(zinc > 65 & zinc < 111, 1, 0))

serzinc %>% tabyl(in_range)

in_range    n percent
  0   67   0.145
  1  395   0.855
```

Once we’ve created this 0-1 variable, there are several available approaches for wrapping a confidence interval around this proportion.

16.6.1 Using an Intercept-Only Regression Again?

We might consider taking the same approach as we did with the population mean earlier:

```
model_zincprop <- lm(in_range ~ 1, data = serzinc)

tidy(model_zincprop, conf.int = TRUE, conf = 0.90) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	0.855	0.016	52.1	0	0.828	0.882

While there are more powerful approaches to estimate a confidence interval around this proportion, this simple approach turns out not to be too bad, so long as the sample proportion isn't very close to either 0 or 1.

16.6.2 Better Approaches: `binom.test` and `prop.test`

Better approaches to estimating a confidence interval for the population proportion use specialized functions such as `binom.test` or `prop.test` in R. Here are those results. These approaches want us first to summarize our information into two counts:

- x = number of observations where the event of interest occurs
- n = number of observations, overall

Here, as we've seen, we have 395 "events" in 462 "trials".

```
serzinc %>% tabyl(in_range) %>% adorn_totals()
```

in_range	n	percent
0	67	0.145
1	395	0.855
Total	462	1.000

So, our data can be summarized as: $x = 395$, $n = 462$.

```
binom.test(x = 395, n = 462, conf.level = 0.90)
```

```
Exact binomial test

data: 395 and 462
number of successes = 395, number of trials = 462, p-value <2e-16
alternative hypothesis: true probability of success is not equal to 0.5
90 percent confidence interval:
0.825 0.881
sample estimates:
```

16.6. INFERENCE ABOUT A POPULATION PROPORTION FROM THIS SAMPLE 283

```
probability of success
0.855
prop.test(x = 395, n = 462, conf.level = 0.90)

1-sample proportions test with continuity correction

data: 395 out of 462, null probability 0.5
X-squared = 231, df = 1, p-value <2e-16
alternative hypothesis: true p is not equal to 0.5
90 percent confidence interval:
0.825 0.881
sample estimates:
p
0.855

Or, if we want to see tidied output, we have...
binom.test(x = 395, n = 462, conf.level = 0.90) %>% tidy()

# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
  <dbl>     <dbl>    <dbl>      <dbl>    <dbl>    <chr>
1 0.855      395 1.23e-57      462     0.825     0.881 Exact
# ... with 1 more variable: alternative <chr>

prop.test(x = 395, n = 462, conf.level = 0.90) %>% tidy()

# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
  <dbl>     <dbl>    <dbl>      <int>    <dbl>    <chr>
1 0.855      231. 2.88e-52      1     0.825     0.881 1-sam~
# ... with 1 more variable: alternative <chr>
```

In Chapter 18, we will discuss the development of confidence intervals for population proportions, and develop additional alternatives for building inference about proportions, percentages and rates.

Chapter 17

Estimating a Population Mean

Suppose that we are interested in learning something about a population or process, from which we can obtain a sample that consists of a subset of potential results from that population or process. The main goal for many of the parametric models that are a large part of statistics is to estimate population parameters, like a population mean, or regression coefficient, on the basis of a sample. When we do this, we want to describe not only our best guess at the parameter – referred to as a *point estimate*, but also say something useful about the uncertainty in our estimate, to let us more completely assess what the data have to tell us. A key tool for doing this is a **confidence interval**, described here in some detail.

Essentially every textbook on introductory statistics describes the development of a confidence interval, at least for a mean. Good supplemental resources include Diez, Barr, and Çetinyaka-Rundel (n.d.), Bock, Velleman, and De Veaux (2004) and Pagano and Gauvreau (2000), for instance.

We'll develop confidence intervals to compare parameters about two populations (either through matched pairs or independent samples) with confidence intervals soon. Here, we'll consider the problem of estimating a confidence interval to describe the mean (or median) of the population represented by a single sample of quantitative data. Our main example uses data from the Serum Zinc study, as described in Chapter 16.

17.1 Defining a Confidence Interval

A confidence interval for a population or process mean uses data from a sample (and perhaps some additional information) to identify a range of potential values

for the population mean, which, if certain assumptions hold, can be assumed to provide a reasonable estimate for the true population mean. A confidence interval consists of:

1. An interval estimate describing the population parameter of interest (here the population mean), and
2. A probability statement, expressed in terms of a confidence level.

17.2 Estimating the Population Mean from the Serum Zinc data

As an example, suppose that we are willing to assume that the mean serum zinc level across the entire population of teenage males, μ , follows a Normal distribution (and so, summarizing it with a mean is a rational thing to do.) Suppose that we are also willing to assume that the 462 teenage males contained in the `serzinc` tibble are a random sample from that complete population. While we know the mean of the sample of 462 boys, we don't know μ , the mean across all teenage males. So we need to estimate it.

Earlier we estimated that a 90% confidence interval for the mean serum zinc level (μ) across the entire population of teenage males was (86.71, 89.16) micrograms per deciliter. How should we interpret this result?

- Some people think this means that there is a 90% chance that the true mean of the population, μ , falls between 86.71 and 89.16 micrograms per deciliter. That's not correct.
- The population mean is a constant **parameter** of the population of interest. That constant is not a random variable, and does not change. So the actual probability of the population mean falling inside that range is either 0 or 1.
- Our confidence is in our process.
 - It's in the sampling method (random sampling) used to generate the data, and in the assumption that the population follows a Normal distribution.
 - It's captured in our accounting for one particular type of error (called *sampling error*) in developing our interval estimate, while assuming all other potential sources of error are negligible.

So, what's closer to the truth is:

- If we used this same method to sample data from the true population of teenage males, and built 100 such 90% confidence intervals, then about 90 of them would contain the true population mean.

17.3 Confidence vs. Significance Level

We've estimated a 90% confidence interval for the population mean serum zinc level among teenage boys using the `serzinc` data.

- We call $100(1-\alpha)\%$, here, 90%, or 0.90, the *confidence* level, and
- $\alpha = 10\%$, or 0.10 is called the *significance* level.

If we had instead built a series of 100 different 95% confidence intervals, then about 95 of them would contain the true value of μ .

Let's look more closely at the issue of estimating a population **mean** based on a sample of observations. We will need three critical pieces - the sample, the confidence level, and the margin of error, which is based on the standard error of a sample mean, when we are estimating a population mean.

17.4 The Standard Error of a Sample Mean

The standard error, generally, is the name we give to the standard deviation associated with any particular parameter estimate.

- If we are using a sample mean based on a sample of size n to estimate a population mean, the **standard error of that sample mean** is σ/\sqrt{n} , where σ is the standard deviation of the measurements in the population.
- We often estimate this particular standard error with its sample analogue, s/\sqrt{n} , where s is the sample standard deviation.
- Other statistics have different standard errors.
 - $\sqrt{p(1-p)/n}$ is the standard error of the sample proportion p estimated using a sample of size n .
 - $\sqrt{\frac{1-r^2}{n-2}}$ is the standard error of the sample Pearson correlation r estimated using n pairs of observations.

In developing a confidence interval for a population mean, we may be willing to assume that the data in our sample are drawn from a Normally distributed population. If so, the most common and useful means of building a confidence interval makes use of the t distribution (sometimes called Student's t) and the notion of a *standard error*.

17.5 The t distribution and Confidence Intervals for μ

In practical settings, we will use the t distribution to estimate a confidence interval from a population mean whenever we:

- are willing to assume that the sample is drawn at random from a population or process with a Normal distribution,
- are using our sample to estimate both the mean and standard deviation, and
- have a small sample size.

17.5.1 The Formula

We can build a $100(1-\alpha)\%$ confidence interval using the t distribution, using the sample mean \bar{x} , the sample size n , and the sample standard deviation s .

The two-sided $100(1-\alpha)\%$ confidence interval (based on a t test) is:

$$\bar{x} \pm t_{\alpha/2, n-1}(s/\sqrt{n})$$

where $t_{\alpha/2, n-1}$ is the value that cuts off the top $\alpha/2$ percent of the t distribution, with $n - 1$ degrees of freedom.

We obtain the relevant cutoff value in R by substituting in values for `alphaover2` and `n-1` into the following line of R code:

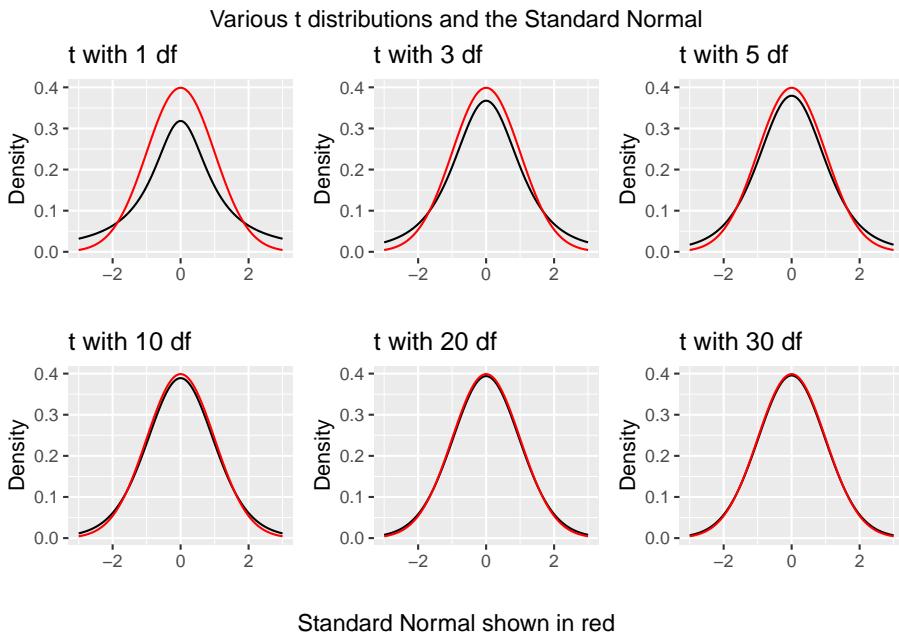
```
qt(alphaover2, df = n-1, lower.tail=FALSE)
```

17.5.2 Student's t distribution

Student's t distribution looks a lot like a Normal distribution, when the sample size is large. Unlike the normal distribution, which is specified by two parameters, the mean and the standard deviation, the t distribution is specified by one parameter, the degrees of freedom.

- t distributions with large numbers of degrees of freedom are more or less indistinguishable from the standard Normal distribution.
- t distributions with smaller degrees of freedom (say, with $df < 30$, in particular) are still symmetric, but are more outlier-prone than a Normal distribution

17.5. THE T DISTRIBUTION AND CONFIDENCE INTERVALS FOR μ_{289}



17.5.3 Building the CI “by hand” for the Serum Zinc data

In the serum zinc data, we observe the following results in our sample.

```
mosaic::favstats(serzinc$zinc)
```

```
min Q1 median Q3 max mean sd n missing
50 76 86 98 153 87.9 16 462 0
```

Suppose we wish to build a 90% confidence interval for the true mean serum zinc level across the entire population of teenage males. The confidence level will be 90%, or 0.90, and so the α value, which is $1 - \text{confidence} = 0.10$.

So what we know going in is that:

- We want $\alpha = 0.10$, because we’re creating a 90% confidence interval.
- The sample size $n = 462$ serum zinc measurements.
- The sample mean of those measurements, $\bar{x} = 87.937$ micrograms per deciliter.
- The sample standard deviation of those measurements, $s = 16.005$ micrograms per deciliter.
- As a result, our standard error of the sample mean is estimated well with $s/\sqrt{n} = 16.005/\sqrt{462} = 0.745$.

So now, we are ready to calculate our 90% confidence interval.

The two-sided $100(1-\alpha)\%$ confidence interval (based on a t test) is:
 $\bar{x} \pm t_{\alpha/2,n-1}(s/\sqrt{n})$, or

- The 90% CI for μ is thus $87.937 \pm t_{0.10/2,462-1} (0.745)$
 – To calculate the t cutoff value for $\alpha = 0.10$ and $n = 462$, we use

```
qt(0.10/2, df = 462-1, lower.tail=FALSE) = 1.648
```

- So the 90% CI for μ is $87.937 \pm 1.648 \times 0.745$, or
- 87.937 ± 1.228 , or
- $(86.71, 89.16)$

So, our 90% confidence interval for the true population mean serum zinc level, based on our sample of 462 patients, is $(86.71, 89.16)$ micrograms per deciliter.

17.5.4 Getting R to build a CI for the Serum Zinc data

Happily, R does all of this work, and with less inappropriate rounding, in the context of fitting an intercept-only linear regression model. An intercept-only model is fitted by putting the number 1 on the right hand side of our linear model. The resulting model simply fits the overall mean of the data as a prediction for all subjects.

```
model_zinc <- lm(zinc ~ 1, data = serzinc)

summary(model_zinc)
```

```
Call:
lm(formula = zinc ~ 1, data = serzinc)

Residuals:
    Min      1Q  Median      3Q     Max 
-37.94 -11.94  -1.94   10.06  65.06 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  87.937     0.745    118    <2e-16 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16 on 461 degrees of freedom
```

```
confint(model_zinc, level = 0.90)

      5 % 95 %
(Intercept) 86.7 89.2
```

17.5. THE T DISTRIBUTION AND CONFIDENCE INTERVALS FOR μ 291

Generally, though, I'll use the `tidy()` function in `broom` to obtain the key information from a model like this:

```
# requires library(broom)
tidy(model_zinc, conf.int = TRUE, conf = 0.90)

# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept) 87.9      0.745    118.       0     86.7     89.2
```

As an alternative, we could also use the `t.test` function, which can build (in this case) a two-sided confidence interval for the zinc levels like this:

```
tt <- t.test(serzinc$zinc, conf.level = 0.90, alternative = "two.sided")

tt
```

```
One Sample t-test

data: serzinc$zinc
t = 118, df = 461, p-value <2e-16
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 86.7 89.2
sample estimates:
mean of x
 87.9
```

and the `tidy()` function from the `broom` package works here, too.

```
# requires library(broom)
tidy(tt, conf.int = TRUE, conf = 0.90) %>%
  knitr::kable(digits = 2)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
87.9	118	0	461	86.7	89.2	One Sample t-test	two.sided

And again, our 90% confidence interval for the true population mean serum zinc level, based on our sample of 462 patients, is (86.71, 89.16) micrograms per deciliter¹.

17.5.5 Interpreting the Result

An appropriate interpretation of the 90% two-sided confidence interval above follows:

¹Since the measured zinc levels appear as integers, we should probably be rounding even further in our confidence interval, down to perhaps one decimal place.

- (86.71, 89.16) micrograms per deciliter is a 90% two-sided confidence interval for the population mean serum zinc level among teenage males.
- Our point estimate for the true population mean serum zinc level is 87.94. The values in the interval (86.71, 89.16) represent a reasonable range of estimates for the true population mean serum zinc level, and we are 90% confident that this method of creating a confidence interval will produce a result containing the true population mean serum zinc level.
- Were we to draw 100 samples of size 462 from the population described by this sample, and use each such sample to produce a confidence interval in this manner, approximately 90 of those confidence intervals would cover the true population mean serum zinc level.

17.5.6 What if we want a 95% or 99% confidence interval instead?

We can obtain them using `tidy` and the same modeling approach.

```
tidy(model_zinc, conf.int = TRUE, conf.level = 0.95)

# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>      <dbl>     <dbl>      <dbl>    <dbl>     <dbl>     <dbl>
1 (Intercept)   87.9     0.745     118.      0     86.5     89.4

tidy(model_zinc, conf.int = TRUE, conf.level = 0.99)

# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>      <dbl>     <dbl>      <dbl>    <dbl>     <dbl>     <dbl>
1 (Intercept)   87.9     0.745     118.      0     86.0     89.9
```

17.5.7 Using the `broom` package with the `t` test

The `broom` package takes the messy output of built-in functions in R, such as `lm`, `t.test` or `wilcox.test`, and turns them into tidy data frames. A detailed description of the package and three of its key functions is found at <https://github.com/tidyverse/broom>.

For example, we can use the `tidy` function within `broom` to create a single-row tibble of the key results from a `t` test.

```
tt <- t.test(serzinc$zinc, conf.level = 0.95, alternative = "two.sided")
tidy(tt)

# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
  <dbl>     <dbl>    <dbl>     <chr>      <dbl>     <dbl>     <chr>
```

```

<dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <chr>
1     87.9      118.       0      461      86.5      89.4 One S~
# ... with 1 more variable: alternative <chr>

```

We can thus pull the endpoints of a 99% confidence interval directly from this output. `broom` also has a `glance` function, which returns the same information as `tidy` in the case of a t-test.

```

tt2 <- t.test(serzinc$zinc, conf.level = 0.90, alternative = "less")
glance(tt2)

# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
  <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <chr>
1     87.9      118.       1      461      -Inf      88.9 One S~
# ... with 1 more variable: alternative <chr>

```

The `t.test` function in R also has an argument to specify the desired confidence level, for example:

```
t.test(serzinc$zinc, conf.level = 0.95, alternative = "two.sided")
```

```

One Sample t-test

data: serzinc$zinc
t = 118, df = 461, p-value <2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 86.5 89.4
sample estimates:
mean of x
 87.9

```

17.5.8 Effect of Changing the Confidence Level

Below, we see two-sided confidence intervals for various levels of α .

Confidence Level	α	Two-Sided Interval	
		Estimate for Zinc Level Population Mean, μ	Point Estimate of μ
80% or 0.80	0.20	(87, 88.9)	87.9
90% or 0.90	0.10	(86.7, 89.2)	87.9
95% or 0.95	0.05	(86.5, 89.4)	87.9
99% or 0.99	0.01	(86, 89.9)	87.9

What happens to the width of the confidence interval in this table as the confidence level changes?

17.5.9 One-sided vs. Two-sided Confidence Intervals

Occasionally, we want to estimate either an upper limit for the population mean μ , or a lower limit for μ , but not both.

If we, as before, have a sample of size n , with sample mean \bar{x} and sample standard deviation s , then:

- The upper bound for a one-sided $100(1-\alpha)\%$ confidence interval for the population mean is $\mu \leq \bar{x} + t_{\alpha,n-1}(\frac{s}{\sqrt{n}})$, with lower “bound” $-\infty$.
- The corresponding lower bound for a one-sided $100(1 - \alpha)$ CI for μ would be $\mu \geq \bar{x} - t_{\alpha,n-1}(\frac{s}{\sqrt{n}})$, with upper “bound” ∞ .

17.5.10 Calculating a one-sided confidence interval for the population mean

```
t.test(serzinc$zinc, conf.level = 0.90, alternative = "greater")
```

```
One Sample t-test

data: serzinc$zinc
t = 118, df = 461, p-value <2e-16
alternative hypothesis: true mean is greater than 0
90 percent confidence interval:
 87 Inf
sample estimates:
mean of x
 87.9

t.test(serzinc$zinc, conf.level = 0.90, alternative = "less")
```

```
One Sample t-test

data: serzinc$zinc
t = 118, df = 461, p-value = 1
alternative hypothesis: true mean is less than 0
90 percent confidence interval:
-Inf 88.9
sample estimates:
mean of x
```

87.9

17.5.11 Relationship between One-Sided and Two-Sided CIs

Note the relationship between the *two-sided* 80% confidence interval, and the *one-sided* 90% confidence intervals.

Confidence	α	Type of Interval	Interval Estimate for Zinc Level Population Mean, μ
80% (.80)	0.20	Two-Sided	(86.98, 88.89)
90% (.90)	0.10	One-Sided (Less Than)	$\mu < 88.89$.
90% (.90)	0.10	One-Sided (Greater Than)	$\mu > 86.98$.

Why does this happen? The 80% two-sided interval is placed so as to cut off the top 10% of the distribution with its upper bound, and the bottom 10% of the distribution with its lower bound. The 90% “less than” one-sided interval is placed so as to have its lower bound cut off the top 10% of the distribution.

The same issue appears when we consider two-sided 90% and one-sided 95% confidence intervals.

Confidence	α	Type of Interval	Interval Estimate for Zinc Level Population Mean, μ
90% (.90)	0.10	Two-Sided	(86.71, 89.16)
95% (.95)	0.05	One-Sided (Less Than)	$\mu < 89.16$.
95% (.95)	0.05	One-Sided (Greater Than)	$\mu > 86.71$.

Again, the 90% two-sided interval cuts off the top 5% and bottom 5% of the distribution with its bounds. The 95% “less than” one-sided interval also has its lower bound cut off the top 5% of the distribution.

17.6 Bootstrap Confidence Intervals for μ

17.6.1 What is a Bootstrap and Why Should I Care?

The bootstrap (and in particular, what’s known as bootstrap resampling) is a really good idea that you should know a little bit about. Good (2005) and Good and Hardin (2006) are excellent resources, for instance.

If we want to know how accurately a sample mean estimates the population mean, we would ideally like to take a very, very large sample, because if we did so, we could conclude with something that would eventually approach mathematical certainty that the sample mean would be very close to the population mean.

But we can rarely draw enormous samples. So what can we do?

17.6.2 Resampling is A Big Idea

One way to find out how precise our estimates are is to run them on multiple samples of the same size. This *resampling* approach was codified originally by Brad Efron in, for example, Efron (1979).

Oversimplifying a lot, the idea is that if we sample (with replacement) from our current sample, we can draw a new sample of the same size as our original.

- And if we repeat this many times, we can generate as many samples of, say, 462 zinc levels, as we like.
- Then we take these thousands of samples and calculate (for instance) the sample mean for each, and plot a histogram of those means.
- If we then cut off the top and bottom 5% of these sample means, we obtain a reasonable 90% confidence interval for the population mean.

17.6.3 When is a Bootstrap Confidence Interval for μ Reasonable?

The interval will be reasonable as long as we're willing to believe that:

- the original sample was a random sample (or at least a completely representative sample) from a population,
- and that the samples are independent of each other,

even if the population of interest doesn't follow a Normal, or even a symmetric distribution.

A downside of the bootstrap is that you and I will get (somewhat) different answers if we resample from the same data without setting the same random seed.

17.6.4 Bootstrap: Steps to estimate a confidence interval for μ

To avoid the Normality assumption, and take advantage of modern computing power, we use R to obtain a bootstrap confidence interval for the population mean based on a sample.

What the computer does:

1. Resample the data with replacement, until it obtains a new sample that is equal in size to the original data set.
2. Calculates the statistic of interest (here, a sample mean.)
3. Repeat the steps above many times (the default is 1,000 using our approach) to obtain a set of 1,000 sample means.
4. Sort those 1,000 sample means in order, and estimate the 95% confidence interval for the population mean based on the middle 95% of the 1,000 bootstrap samples.
5. Send us a result, containing the sample mean, and a 95% confidence interval for the population mean

17.6.5 Using R to estimate a 90% CI for μ with the bootstrap

The command that we use to obtain a CI for μ using the basic nonparametric bootstrap and without assuming a Normally distributed population, is `smean.cl.boot`, a part of the `Hmisc` package in R.

```
set.seed(431)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)

Mean Lower Upper
87.9 86.8 89.2

• Remember that the t-based 90% CI for  $\mu$  was (86.71, 89.16), according to the following output...

tidy(lm(zinc ~ 1, data = serzinc), conf.int = TRUE, conf.level = 0.90)

# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>       <dbl>     <dbl>     <dbl>    <dbl>     <dbl>     <dbl>
1 (Intercept)   87.9     0.745    118.       0     86.7     89.2
```

17.6.6 Comparing Bootstrap and T-Based Confidence Intervals

- The `smean.cl.boot` function (unlike most R functions) deletes missing data automatically, as does the `smean.cl.normal` function, which can also be used to produce the t-based confidence interval.

```
set.seed(431)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)

Mean Lower Upper
```

```
87.9 86.8 89.2
serzinc %$% Hmisc::smean.cl.normal(zinc, conf.int = 0.90)
```

Mean	Lower	Upper
87.9	86.7	89.2

Bootstrap resampling confidence intervals do not follow the general confidence interval strategy using a point estimate \pm a margin for error.

- A bootstrap interval is often asymmetric, and while it will generally have the point estimate (the sample mean) near its center, for highly skewed data, this will not necessarily be the case.
- We will usually use either 1,000 (the default) or 10,000 bootstrap replications for building confidence intervals – practically, it makes little difference.

17.6.7 90% CI for μ via bootstrap, changing minor details

Suppose we change the random seed that we set, or change the number (B) of desired bootstrap replications.

```
set.seed(431)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)
```

Mean	Lower	Upper
87.9	86.8	89.2

```
set.seed(431212)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)
```

Mean	Lower	Upper
87.9	86.7	89.1

```
set.seed(431212)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)
```

Mean	Lower	Upper
87.9	86.7	89.1

17.6.8 Bootstrap: Changing the Confidence Level

```
set.seed(431654)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.95)
```

Mean	Lower	Upper
87.9	86.5	89.4

```
set.seed(431321)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.99)

Mean Lower Upper
87.9 86.2 89.7
```

17.6.9 Bootstrap: Obtaining a One-sided Confidence Interval

If you want to estimate a one tailed confidence interval for the population mean using the bootstrap, then the procedure is as follows:

1. Determine α , the significance level you want to use in your one-sided confidence interval. Remember that α is 1 minus the confidence level. Let's assume we want a 90% one-sided interval, so $\alpha = 0.10$.
2. Double α to determine the significance level we will use in the next step to fit a two-sided confidence interval.
3. Fit a two-sided confidence interval with confidence level $100(1 - 2 * \alpha)$. Let the bounds of this interval be (a, b) .
4. The one-sided (greater than) confidence interval will have a as its lower bound.
5. The one-sided (less than) confidence interval will have b as its upper bound.

Suppose that we want to find a 95% one-sided upper bound for the population mean serum zinc level among teenage males, μ , using the bootstrap.

Since we want a 95% confidence interval, we have $\alpha = 0.05$. We double that to get $\alpha = 0.10$, which implies we need to instead fit a two-sided 90% confidence interval.

```
set.seed(43101)
serzinc %$% Hmisc::smean.cl.boot(zinc, B = 1000, conf.int = 0.90)

Mean Lower Upper
87.9 86.7 89.1
```

The upper bound of this two-sided 90% CI will also be the upper bound for a 95% one-sided CI.

17.6.10 Bootstrap CI for the Population Median

If we are willing to do a small amount of programming work in R, we can obtain bootstrap confidence intervals for other population parameters besides the mean. One statistic of common interest is the median. How do we find a confidence interval for the population median using a bootstrap approach? The easiest way I know of makes use of the `boot` package, as follows.

In step 1, we specify a new function to capture the medians from our sample.

```
f.median <- function(y, id)
{   median ( y[id]) }
```

In step 2, we summon the `boot` package and call the `boot.ci` function:

```
set.seed(431787)
boot.ci(boot (serzinc$zinc, f.median, 1000), conf=0.90, type="basic")
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates
```

```
CALL :
boot.ci(boot.out = boot(serzinc$zinc, f.median, 1000), conf = 0.9,
        type = "basic")

Intervals :
Level      Basic
90%    (84, 87 )
Calculations and Intervals on Original Scale
```

This yields a 90% confidence interval² for the population median serum zinc level.

Recall that the sample median for the serum zinc levels in our sample of 462 teenage males was 86 micrograms per deciliter.

```
mosaic::favstats(~ zinc, data = serzinc)
```

```
min Q1 median Q3 max mean sd n missing
50 76     86 98 153 87.9 16 462      0
```

17.6.11 Bootstrap CI for the IQR

If for some reason, we want to find a 95% confidence interval for the population value of the inter-quartile range via the bootstrap, we can do it.

```
IQR(serzinc$zinc)
```

```
[1] 22
f.IQR <- function(y, id)
{   IQR (y[id]) }

set.seed(431207)
```

²Actually, the `boot.ci` function can provide up to five different types of confidence interval (see the help file) if we change to `type="all"`, and some of those other versions have attractive properties. However, we'll stick with the basic approach in 431.

```
boot.ci(boot (serzinc$zinc, f.IQR, 1000),
        conf=0.95, type="basic")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
 boot.ci(boot.out = boot(serzinc\$zinc, f.IQR, 1000), conf = 0.95,
 type = "basic")

Intervals :
 Level Basic
 95% (20.0, 24.2)
 Calculations and Intervals on Original Scale

17.6.12 Bootstrap Resampling: Advantages and Caveats

The bootstrap may seem like the solution to all estimation problems. In theory, we could use the same approach to find a confidence interval for any other parameter – it's not perfect, but it is very useful. Bootstrap procedures exist for virtually any statistical comparison - the t-test analog is just one of many possibilities, and bootstrap methods are rapidly gaining on more traditional approaches in the literature thanks mostly to faster computers.

The great advantage of the bootstrap is its relative simplicity, but don't forget that many of the original assumptions of the t-based confidence interval still hold.

- Using a bootstrap does eliminate the need to worry about the Normality assumption in small sample size settings, but it still requires independent and identically distributed samples from the population of interest.

The bootstrap produces clean and robust inferences (such as confidence intervals) in many tricky situations. It is still possible that the results can be both:

- **inaccurate** (i.e. they can include the true value of the unknown population mean less often than the stated confidence probability) and
- **imprecise** (i.e., they can include more extraneous values of the unknown population mean than is desirable).

17.7 Large-Sample Normal Approximation CIs for μ

If we were in the position of knowing the standard deviation of the population of interest precisely³, we could use that information to build a $100(1-\alpha)\%$ confidence interval using the Normal distribution, based on the sample mean \bar{x} , the sample size n , and the (known) population standard deviation σ .

When we have a large sample size (often as little as 60 observations), we can use this approach to get a very close approximation to the result we would get using the t distribution, and there are many settings where obtaining the Z test result is more appropriate in estimating more complicated parameters than the population mean. When estimating a population mean in practice, though, I do not use this approach.

17.7.1 The Large Sample Formula for the CI around μ

The two-sided $100(1-\alpha)\%$ confidence interval for a population mean μ (based on the Normal distribution) is:

- The Lower Bound is $\bar{x} - Z_{\alpha/2}(\sigma/\sqrt{n})$ and the Upper Bound is $\bar{x} + Z_{\alpha/2}(\sigma/\sqrt{n})$

where $Z_{\alpha/2}$ is the value that cuts off the top $\alpha/2$ percent of the standard Normal distribution (the Normal distribution with mean 0 and standard deviation 1).

17.7.2 Obtaining the $Z_{\alpha/2}$ value using `qnorm`

We can obtain this cutoff value from R by substituting in the desired proportion for `alphaover2` into the `qnorm` function as follows:

```
qnorm(alphaover2, lower.tail=FALSE)
```

For example, if we are building a 95% confidence interval, we have $100(1-\alpha) = 95$, so that α is 0.05, or 5%. This means that the cutoff value we need to find is $Z_{0.05/2} = Z_{.025}$, and this turns out to be 1.96.

```
qnorm(0.025, lower.tail=FALSE)
```

```
[1] 1.96
```

³Practical applications usually demand a subtler approach, but this normal distribution-based approach can help us fix some key ideas

17.7.3 Commonly Used Cutoffs based on the Normal Distribution

- If we're building a two-sided 95% confidence interval, we'll use $Z_{.025} = 1.96$
- For a two-sided 90% confidence interval, we use $Z_{.05} = 1.645$
- For a two-sided 99% confidence interval, we use $Z_{.005} = 2.576$
- For a two-sided 50% confidence interval, we use $Z_{.25} = 0.67$
- For a two-sided 68% confidence interval, we use $Z_{.16} = 0.99$

17.7.4 Lots of CIs use the Normal distribution

- The usual 95% confidence interval for large samples is an estimate ± 2 standard errors⁴.
- Also, from the Normal distribution, an estimate ± 1 standard error is a 68% confidence interval, and an estimate $\pm 2/3$ of a standard error is a 50% confidence interval.
- A 50% interval is particularly easy to interpret because the true value should be inside the interval about as often as it is not.
- A 95% interval is thus about three times as wide as a 50% interval.
- In general, the larger the confidence required, the wider the interval will need to be.

17.7.5 Large-Sample Confidence Interval for Zinc Levels

Since we have a fairly large sample ($n = 462$) in the `serzinc` data, we could consider using a large-sample approach (assuming the sample standard deviation is equal to the population standard deviation, and then using the Normal distribution) to estimate a confidence interval for the mean zinc levels in the population of all 15-17 year old males like those in our sample.

In the zinc levels within the `serzinc` data, we have

- a sample of $n = 462$ observations
- with sample mean $\bar{x} = 87.94$ and standard deviation $s = 16$
- and suppose we want to, at first, find a 95% confidence interval, so $\alpha = 0.05$

The 95% confidence interval is calculated as $\bar{x} \pm Z_{\alpha/2}(\sigma/\sqrt{n})$, and here we will assume that $s = \sigma$ which may be reasonable with a fairly large sample size:

$$87.94 \pm (1.96)(16 / \sqrt{462}) = 87.94 \pm 1.46, \text{ or } (86.48, 89.4)$$

⁴The use of 2 standard errors for a confidence interval for a population mean is certainly reasonable whenever n is 60 or more. This is because the t distribution with 59 degrees of freedom has a 0.025 cutoff of 2.0, anyway.

Our 95% confidence interval for the population mean is $(86.48, 89.4) \mu\text{g/dl}$. Were we to generate 100 such intervals, approximately 95 of those intervals would be expected to include the true mean of the entire population of 15-17 year old males like those in our sample.

17.7.6 Comparing Z and t-based Intervals for Serum Zinc

For the serum zinc data, we had $n = 462$ observations in our sample.

Do the z-based and t-based confidence intervals differ much?

α	Confidence Level	Confidence Interval	Method
0.05	95%	$(86.48, 89.40)$	Z (known σ ; large n)
0.05	95%	$(86.47, 89.40)$	t (σ unknown)
0.10	90%	$(86.72, 89.16)$	Z (known σ ; large n)
0.10	90%	$(86.71, 89.16)$	t (σ unknown)

17.7.7 One-Sided Confidence Intervals in Large Samples

The upper bound for a one-sided $100(1-\alpha)\%$ confidence interval for the population mean is:

$$\mu \leq \bar{x} + Z_\alpha \left(\frac{\sigma}{\sqrt{n}} \right), \text{ with lower "bound" } -\infty.$$

The corresponding lower bound for a one-sided $100(1 - \alpha)$ CI for μ would be:

$$\mu \geq \bar{x} - Z_\alpha \left(\frac{\sigma}{\sqrt{n}} \right), \text{ with upper "bound" } \infty.$$

17.8 Wilcoxon Signed Rank Procedure for CIs

17.8.1 Confidence Intervals for the Median of a Population

It turns out to be difficult, without the bootstrap, to estimate an appropriate confidence interval for the median of a population, which might be an appealing thing to do, particularly if the sample data are clearly not Normally distributed, so that a median seems like a better summary of the center of the data. Bootstrap procedures are available to perform the task.

The Wilcoxon signed rank approach can be used as an alternative to t-based procedures to build interval estimates for the population *pseudo-median* when the population cannot be assumed to follow a Normal distribution.

As it turns out, if you're willing to assume the population is **symmetric** (but not necessarily Normally distributed) then the pseudo-median is actually equal to the population median.

17.8.2 What is a Pseudo-Median?

The pseudo-median of a particular distribution G is the median of the distribution of $(u + v)/2$, where both u and v have the same distribution (G).

- If the distribution G is symmetric, then the pseudomedian is equal to the median.
- If the distribution is skewed, then the pseudomedian is not the same as the median.
- For any sample, the pseudomedian is defined as the median of all of the midpoints of pairs of observations in the sample.

17.8.3 Getting the Wilcoxon Signed Rank-based CI in R

```
serzinc %$% wilcox.test(zinc, conf.int = TRUE, conf.level = 0.95)

Wilcoxon signed rank test with continuity correction

data: zinc
V = 1e+05, p-value <2e-16
alternative hypothesis: true location is not equal to 0
95 percent confidence interval:
86.0 88.5
sample estimates:
(pseudo)median
87.5
```

17.8.4 Interpreting the Wilcoxon CI for the Population Median

If we're willing to believe the `zinc` levels come from a population with a symmetric distribution, the 95% Confidence Interval for the population median would be (86, 88.5)

For a non-symmetric population, this only applies to the *pseudo-median*.

Note that the pseudo-median (87.5) is actually closer here to the sample mean (87.9) than it is to the sample median (86).

17.8.5 Using the `broom` package with the Wilcoxon test

We can also use the `tidy` function within `broom` to create a single-row tibble of the key results from a Wilcoxon test, so long as we run `wilcox.test` specifying that we want a confidence interval.

```
wt <- serzinc %$% wilcox.test(zinc, conf.int = TRUE, conf.level = 0.95)
tidy(wt)

# A tibble: 1 x 7
  estimate statistic p.value conf.low conf.high method      alternative
    <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>      <chr>
1     87.5    106953  2.00e-77     86.0     88.5 Wilcoxon sign~ two.sided
```

17.9 General Advice

We have described four different approaches to estimating a confidence interval for the center of a distribution of quantitative data.

1. The most commonly used approach uses the t distribution to estimate a confidence interval for a population/process mean. This requires some extra assumptions, most particularly that the underlying distribution of the population values is at least approximately Normally distributed. This is identical to the result we get from an intercept-only linear regression model.
2. A more modern and very general approach uses the idea of the bootstrap to estimate a confidence for a population/process parameter, which could be a mean, median or other summary statistic. The bootstrap, and the underlying notion of *resampling* is an important idea that lets us avoid some of the assumptions (in particular Normality) that are required by other methods. Bootstrap confidence intervals involve random sampling, so that the actual values obtained will differ a bit across replications.
3. A third approach makes more substantial assumptions - it uses the Normal distribution rather than a t , and assumes (among other things) very large samples. For estimating a single mean, we'll never do this, but for estimating more complex parameters, particularly in Part C when discussing modeling, we will occasionally use this approach.
4. Finally, the Wilcoxon signed-rank method is one of a number of inferential tools which transform the data to their *ranks* before estimating a confidence interval. This avoids some assumptions, but yields inferences about a less-familiar parameter - the pseudo-median.

Most of the time, the `bootstrap` provides a reasonably adequate confidence interval estimate of the population value of a parameter (mean or median, most commonly) from a distribution when our data consists of a single sample of quantitative information.

Chapter 18

Estimating a Population Proportion

We've focused on creating statistical inferences about a population mean, or difference between means, where we care about a quantitative outcome. Now, we'll tackle **categorical** outcomes, by estimating a confidence interval around a population proportion.

18.1 A First Example: Serum Zinc in the “Normal” Range?

Recall that in the serum zinc study, we have 462 teenage male subjects, of whom 395 (or 85.5%) fell in the “normal range” of 66 to 110 micrograms per deciliter.

```
serzinc <- serzinc %>%
  mutate(in_range = ifelse(zinc >= 66 & zinc <= 110, 1, 0))

serzinc %>% tabyl(in_range) %>%
  adorn_totals() %>% adorn_pct_formatting()
```

in_range	n	percent
0	67	14.5%
1	395	85.5%
Total	462	100.0%

18.1.1 A $100(1-\alpha)\%$ Confidence Interval for a Population Proportion

Suppose we want to estimate a confidence interval for an unknown population proportion, π , on the basis of a random sample of n observations from that population which yields a sample proportion of p . Note that this p is the sample proportion – it's not a p value.

- In our serum zinc example, we have $n = 462$ observations, with a sample proportion (“in range”) of $p = 0.855$.

A $100(1-\alpha)\%$ confidence interval for the population proportion π can be created by using the standard normal distribution, the sample proportion, p , and the standard error of a sample proportion, which is defined as the square root of p multiplied by $(1-p)$ divided by the sample size, n .

- So the standard error is estimated in our serum zinc example as:

$$\sqrt{\frac{p(1-p)}{n}} = \sqrt{\frac{0.855(1-0.855)}{462}} = \sqrt{0.000268} = 0.016$$

And thus, our confidence interval is $p \pm Z_{\alpha/2} \sqrt{\frac{p(1-p)}{n}}$

where $Z_{\alpha/2}$ = the value from a standard Normal distribution cutting off the top $\alpha/2$ of the distribution, obtained in R by substituting the desired $\alpha/2$ value into the following command: `qnorm(alpha/2, lower.tail=FALSE)`.

Note: This interval is reasonably accurate so long as np and $n(1-p)$ are each at least 5.

- For the serum zinc data, we have $np = (462)(0.855) = 395$ and $n(1-p) = 462(1 - 0.855) = 67$, so this should be ok.
- For $\alpha = 0.05$, we have $Z_{\alpha/2} = 1.96$, approximately.

```
qnorm(0.025, lower.tail = FALSE)
```

[1] 1.96

- Thus, for the serum zinc estimate, this confidence interval would be:

$$p \pm Z_{\alpha/2} \sqrt{\frac{p(1-p)}{n}} = \frac{395}{462} \pm 1.96 \sqrt{\frac{0.855(1-0.855)}{462}} = 0.855 \pm 0.032$$

or $(0.823, 0.887)$.

I am aware of at least three different procedures for estimating a confidence interval for a population proportion using R. All have minor weaknesses: none is importantly different from the others in many practical situations.

18.1.2 The `prop.test` approach (Wald test)

The `prop.test` function can be used to establish a very similar confidence interval to the one we calculated above, based on something called the Wald test.

Here, we specify the `x` and `n` values. `n` is the total number of observations, and `x` is the number where the event of interest (in this case, serum zinc levels in the normal range) occurs. So `x = 395` and `n = 462`.

```
prop.test(x = 395, n = 462)
```

```
1-sample proportions test with continuity correction

data: 395 out of 462, null probability 0.5
X-squared = 231, df = 1, p-value <2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.819 0.885
sample estimates:
      p
0.855
```

The 95% confidence interval by this method is (0.819, 0.885), which is close, but not quite the same, to our original estimate of (0.823, 0.887).

The difference from our calculated interval is attributable to differences in rounding, plus the addition of something called a *continuity correction*, since we are using a Normal approximation to the exact binomial distribution to establish our margin for error. R, by default, includes this continuity correction for the Wald test in `prop.test` but we didn’t include it in our original calculation.

For most purposes, when using `prop.test`, I’ll use `tidy` from the `broom` package to present the results.

```
prop.test(x = 395, n = 462, conf.level = 0.95) %>% tidy()
```

```
# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
    <dbl>     <dbl>     <dbl>     <int>     <dbl>     <dbl>   <chr>
1     0.855     231. 2.88e-52       1     0.819     0.885 1-sam-
# ... with 1 more variable: alternative <chr>
```

18.1.3 The `binom.test` approach (Clopper and Pearson “exact” test)

The `binom.test` command can be used to establish an “exact” confidence interval. This uses the method of Clopper and Pearson from 1934, and is exact in the sense that it guarantees, for instance, that the confidence level associated with the interval is at least as large as the nominal level of 95%, but not that the interval isn’t wider than perhaps it needs to be.

```
binom.test(x = 395, n = 462)
```

```
Exact binomial test

data: 395 and 462
number of successes = 395, number of trials = 462, p-value <2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.820 0.886
sample estimates:
probability of success
 0.855
```

The 95% confidence interval by this method is (0.820, 0.886), which is in the same general range as our previous estimates.

For most purposes, when using `binom.test`, I’ll again use `tidy` from the `broom` package to present the results.

```
binom.test(x = 395, n = 462, conf.level = 0.95) %>% tidy()
```

```
# A tibble: 1 x 8
  estimate statistic p.value parameter conf.low conf.high method
    <dbl>      <dbl>     <dbl>      <dbl>      <dbl>     <chr>
1     0.855      395 1.23e-57      462      0.820      0.886 Exact~
# ... with 1 more variable: alternative <chr>
```

18.1.4 SAIFS: single augmentation with an imaginary failure or success

SAIFS stands for “single augmentation with an imaginary failure or success” and the method I’ll describe is one of several similar approaches. The next subsection describes the R code for calculating the relevant confidence interval.

An approach I like for the estimation of a confidence interval for a single population proportion/rate¹ is to estimate the lower bound of a confidence

¹See Borkowf CB (2006) Constructing binomial confidence intervals with near nom-

18.1. A FIRST EXAMPLE: SERUM ZINC IN THE “NORMAL” RANGE?311

interval with an imaginary failure added to the observed data, and estimate the upper bound of a confidence interval with an imaginary success added to the data.

Suppose we have X successes in n trials, and we want to establish a confidence interval for the population proportion of successes.

Let $p_1 = (X + 0)/(n + 1)$, $p_2 = (X + 1)/(n + 1)$, $q_1 = 1 - p_1$, $q_2 = 1 - p_2$

- The lower bound of a $100(1-\alpha)\%$ confidence interval for the population proportion of successes using the SAIFS procedure is then $LB_{SAIFS}(x, n, \alpha) = p_1 - t_{\alpha/2, n-1} \sqrt{\frac{p_1 q_1}{n}}$
- The upper bound of that same $100(1-\alpha)\%$ confidence interval for the population proportion of successes using the SAIFS procedure is $UB_{SAIFS}(x, n, \alpha) = p_2 + t_{\alpha/2, n-1} \sqrt{\frac{p_2 q_2}{n}}$

Returning to the serum zinc example, we’ve got 395 “successes” (value in the normal range) out of 462 “trials” (values measured), so that $X = 395$ and $n = 462$

So we have $p_1 = \frac{X+0}{n+1} = \frac{395}{463} = 0.8531$, $p_2 = \frac{X+1}{n+1} = \frac{396}{463} = 0.8553$, and $q_1 = 1 - p_1 = 0.1469$ and $q_2 = 1 - p_2 = 0.1447$

We have $n = 462$ so if we want a 95% confidence interval ($\alpha = 0.05$), then we have $t_{\alpha/2, n-1} = t_{0.025, 461} = 1.9651$, which I determined using R’s `qt` function:

```
qt(0.025, df = 461, lower.tail=FALSE)
```

```
[1] 1.97
```

- Thus, our lower bound for a 95% confidence interval is $p_1 - t_{\alpha/2, n-1} \sqrt{\frac{p_1 q_1}{n}}$, or $0.8531 - 1.9651 \sqrt{\frac{0.8531(0.1469)}{462}}$, which is $0.8531 - 0.0324$ or 0.8207 .
- Our upper bound is $p_2 + t_{\alpha/2, n-1} \sqrt{\frac{p_2 q_2}{n}}$, or $0.8553 + 1.9651 \sqrt{\frac{0.8553(0.1447)}{462}}$, which is $0.8553 + 0.0323$, or 0.8876 .

So the 95% SAIFS confidence interval estimate for the population proportion, π , of teenage males whose serum zinc levels fall within the “normal range” is $(0.821, 0.888)$.

18.1.5 A Function in R to Calculate the SAIFS Confidence Interval

I built an R function, called `saifs.ci` and contained in the Markdown for this document as well as the `Love-boost.R` script on the web site, which takes as its

final coverage by adding a single imaginary failure or success. Statistics in Medicine. 25(21): 3679-3695. doi: 10.1002/sim.2469, or get the whole PDF of the paper at <http://onlinelibrary.wiley.com/doi/10.1002/sim.2469/pdf>

arguments a value for X = the number of successes, n = the number of trials, and conf.level = the confidence level, and produces the sample proportion, the SAIFS lower bound and upper bound for the specified two-sided confidence interval for the population proportion, using the equations above.

Here, for instance, are 95%, 90% and 99% confidence intervals for the population proportion π that we have been studying in the serum zinc data.

```
saifs.ci(x = 395, n = 462)
```

Sample Proportion	0.025	0.975
0.855	0.821	0.887

```
saifs.ci(x = 395, n = 462, conf=0.9)
```

Sample Proportion	0.05	0.95
0.855	0.826	0.882

```
saifs.ci(x = 395, n = 462, conf=0.99, dig=5)
```

Sample Proportion	0.005	0.995
0.855	0.811	0.898

Note that in the final interval, I asked the machine to round to five digits rather than the default of three. On my desktop (and probably yours), doing so results in this output:

Sample Proportion	0.005	0.995
0.85498	0.81054	0.89763

I've got some setting wrong in my `bookdown` work so that this doesn't show up above when the function is called. Sorry!

18.1.6 The `saifs.ci` function in R

```
`saifs.ci` <-  
  function(x, n, conf.level=0.95, dig=3)  
  {  
    p.sample <- round(x/n, digits=dig)  
  
    p1 <- x / (n+1)  
    p2 <- (x+1) / (n+1)  
  
    var1 <- (p1*(1-p1))/n  
    se1 <- sqrt(var1)  
    var2 <- (p2*(1-p2))/n  
    se2 <- sqrt(var2)
```

```

lowq = (1 - conf.level)/2
tcut <- qt(lowq, df=n-1, lower.tail=FALSE)

lower.bound <- round(p1 - tcut*se1, digits=dig)
upper.bound <- round(p2 + tcut*se2, digits=dig)
res <- c(p.sample, lower.bound, upper.bound)
names(res) <- c('Sample Proportion', lowq, 1-lowq)
res
}

```

18.2 A Second Example: Ebola Mortality Rates through 9 Months of the Epidemic

The World Health Organization's Ebola Response Team published an article² in the October 16, 2014 issue of the New England Journal of Medicine, which contained some data I will use in this example, focusing on materials from their Table 2.

As of September 14, 2014, a total of 4,507 confirmed and probable cases of Ebola virus disease (EVD) had been reported from West Africa. In our example, we will look at a set of 1,737 cases, with definitive outcomes, reported in Guinea, Liberia, Nigeria and Sierra Leone.

Across these 1,737 cases, a total of 1,229 cases led to death. Based on these sample data, what can be said about the case fatality rate in the population of EVD cases with definitive outcomes for this epidemic?

18.2.1 Working through the Ebola Virus Disease Example

We have $n = 1,737$ subjects, of whom we observed death in 1,229, for a sample proportion of $p = \frac{1229}{1737} = 0.708$. The standard error of that sample proportion will be

$$SE(p) = \sqrt{\frac{p(1-p)}{n}} = \sqrt{\frac{0.708(1-0.708)}{1737}} = 0.011$$

And our 95% confidence interval (so that we'll use $\alpha = 0.05$) for the true population proportion, π , of EVD cases with definitive outcomes, who will die is $p \pm Z_{0.025} \sqrt{\frac{p(1-p)}{n}}$, or $0.708 \pm 1.96(0.011) = 0.708 \pm 0.022$, or $(0.686, 0.730)$

Note that I simply recalled from our prior work that $Z_{0.025} = 1.96$, but we can verify this:

²WHO Ebola Response Team (2014) Ebola virus disease in West Africa: The first 9 months of the epidemic and forward projections. *New Engl J Med* 371: 1481-1495 doi: 10.1056/NEJMoa1411100

```
qnorm(0.025, lower.tail=FALSE)
```

```
[1] 1.96
```

Since both $np = (1737)(0.708) = 1230$ and $n(1-p) = (1737)(1-0.708) = 507$ are substantially greater than 5, this should be a reasonably accurate confidence interval.

We are 95% confident that the true population proportion is between 0.686 and 0.730. Equivalently, we could say that we're 95% confident that the true case fatality rate expressed as a percentage rather than a proportion, is between 68.6% and 73.0%.

18.2.2 Using R to estimate the CI for our Ebola example

```
pt <- prop.test(x = 1229, n = 1737)
tidy(pt, conf.int = 0.95) %>%
  knitr::kable(digits = 3)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method
0.708	298	0	1	0.685	0.729	1-sample proportions test with com

The 95% confidence interval by this method is (0.685, 0.729), which is close, but not quite the same, to our original estimate of (0.686, 0.730.)

```
bt <- binom.test(x = 1229, n = 1737)
tidy(bt, conf.int = 0.95) %>%
  knitr::kable(digits = 3)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
0.708	1229	0	1737	0.686	0.729	Exact binomial test	two.sided

```
saifs.ci(x = 1229, n = 1737, conf.level=0.95)
```

Sample Proportion	0.025	0.975
0.708	0.686	0.729

18.2.3 Comparing the Confidence Intervals for the Ebola Virus Disease Example

These three approaches give the following results:

Approach	95% CI for Population Proportion
<code>prop.test</code> (Wald)	(0.685, 0.729)
<code>binom.test</code> (Clopper & Pearson)	(0.686, 0.729)
<code>saifs.ci</code> (Borkowf)	(0.686, 0.729)

So in this case, it really doesn't matter which one you choose. With a smaller sample, we may not come to the same conclusion about the relative merits of these different approaches.

18.3 Can the Choice of Confidence Interval Method Matter?

The SAIFS approach will give a substantially different confidence interval than either the Wald or Clopper-Pearson approaches with a small sample size, and a probability of "success" that is close to either 0 or 1. For instance, suppose we run 10 trials, and obtain a single success, then use these data to estimate the true proportion of success, π .

The 95% confidence intervals under this circumstance are very different.

Method	R Command	95% CI for π
Wald	<code>prop.test(x = 1, n = 10)</code>	0.005, 0.459
Clopper-Pearson	<code>binom.test(x = 1, n = 10)</code>	0.003, 0.445
SAIFS	<code>saifs.ci(x = 1, n = 10)</code>	-0.115, 0.458

Note that the Wald and Clopper-Pearson procedures at least force the confidence interval to appear in the (0, 1) range. The SAIFS approach gives us some impossible values, and is thus a bit hard to take seriously – in reporting the result, we'd probably have to report the SAIFS interval as (0, 0.458).

If instead, we consider a situation where our null hypothesis is that the true proportion π is 0.10 (or 10%), and we run each of these three methods to obtain a 95% confidence interval, then we will come to somewhat different conclusions depending on the choice of method, if we observe 4 successes in 100 trials.

Method	R Command	95% CI for π
Wald	<code>prop.test(x = 4, n = 100)</code>	0.013, 0.105
Clopper-Pearson	<code>binom.test(x = 4, n = 100)</code>	0.011, 0.099
SAIFS	<code>saifs.ci(x = 4, n = 100)</code>	0.001, 0.093

Now, the Wald test suggests we retain the null hypothesis, the Clopper-Pearson test suggests we reject it (barely) and the SAIFS interval is more convinced that we should reject $H_0 : \pi = 0.10$ in favor of $H_A : \pi \neq 0.10$.

None of these three approaches is always better than any of the others. When we have a sample size below 100, or the sample proportion of success is either below 0.10 or above 0.90, caution is warranted, although in many cases,

the various methods give similar responses.

Data	Wald 95% CI	Clopper-Pearson 95% CI	SAIFS 95% CI
10 successes in 30 trials	0.179, 0.529	0.173, 0.528	0.148, 0.534
10 successes in 50 trials	0.105, 0.341	0.1, 0.337	0.083, 0.333
90 successes in 100 trials	0.82, 0.948	0.824, 0.951	0.829, 0.96
95 successes in 100 trials	0.882, 0.981	0.887, 0.984	0.894, 0.994

Chapter 19

The Ibuprofen in Sepsis Trial

19.1 The Ibuprofen in Sepsis Randomized Clinical Trial

We will be working with a sample from the Ibuprofen in Sepsis study, which is also studied in Dupont (2002). Quoting the abstract from Bernard et al. (1997):

Ibuprofen has been shown to have effects on sepsis in humans, but because of their small samples (fewer than 30 patients), previous studies have been inadequate to assess effects on mortality. We sought to determine whether ibuprofen can alter rates of organ failure and mortality in patients with the sepsis syndrome, how the drug affects the increased metabolic demand in sepsis (e.g., fever, tachypnea, tachycardia, hypoxemia, and lactic acidosis), and what potential adverse effects the drug has in the sepsis syndrome.

In this study, patients meeting specific criteria (including elevated temperature) for a diagnosis of sepsis were recruited if they fulfilled an additional set of study criteria in the intensive care unit at one of seven participating centers.

The full trial involved 455 patients, of which our sample includes 300. 150 of our patients were randomly assigned to the Ibuprofen group and 150 to the Placebo group¹. I picked the `sepsis` sample we will work with excluding patients with missing values for our outcome of interest, and then selected a random sample of 150 Ibuprofen and 150 Placebo patients from the rest of the group, and converted the temperatures and changes from Fahrenheit to Celsius.

¹This was a *double-blind* study, where neither the patients nor their care providers know, during the execution of the trial, what intervention group was assigned to each patient.

For the moment, we focus on two variables:

- **treat**, which specifies the treatment group (intravenous Ibuprofen or intravenous Placebo), which was assigned via randomization to each patient, and
- **temp_drop**, the outcome of interest, measured as the change from baseline to 2 hours later in degrees Celsius. Positive values indicate improvement, that is, a *drop* in temperature over the 2 hours following the baseline measurement.

The **sepsis.csv** file also contains each subject's

- *id*, which is just a code
- *race* (three levels: White, AfricanA or Other)
- *apache* = baseline APACHE II score, a severity of disease score ranging from 0 to 71 with higher scores indicating more severe disease and a higher mortality risk
- *temp_0* = baseline temperature, degrees Celsius.

but for the moment, we won't worry about those.

```
sepsis <- sepsis %>%
  mutate(treat = factor(treat),
        race = factor(race))

mosaic::favstats(temp_drop ~ treat, data = sepsis)

  treat  min      Q1 median   Q3 max   mean      sd    n missing
1 Ibuprofen -1.5  0.000    0.5 0.9 3.1 0.464 0.688 150      0
2 Placebo   -2.7 -0.175    0.1 0.4 1.9 0.153 0.571 150      0
```

19.2 Our Key Questions for an Independent Samples Comparison

19.2.1 What is the population under study?

- All patients in the intensive care unit with sepsis who meet the inclusion and exclusion criteria of the study, at the entire population of health centers like the ones included in the trial.

19.2.2 What is the sample? Is it representative of the population?

- The sample consists of 300 patients. It is a convenient sample from the population under study.

- This is a randomized clinical trial. 150 of the patients were assigned to Ibuprofen, and the rest to Placebo. It is this treatment assignment that is randomized, not the selection of the sample as a whole.
- In expectation, randomization of individuals to treatments, as in this study, should be expected to eliminate treatment selection bias.

19.2.3 Who are the subjects / individuals within the sample?

- 150 patients who received Ibuprofen and a completely different set of 150 patients who received Placebo.
- There is no match or link between the patients. They are best thought of as independent samples.

19.2.4 What data are available on each individual?

- The key variables are the treatment indicator (Ibuprofen or Placebo) and the outcome (drop in temperature in the 2 hours following administration of the randomly assigned treatment.)

19.2.5 RCT Caveats

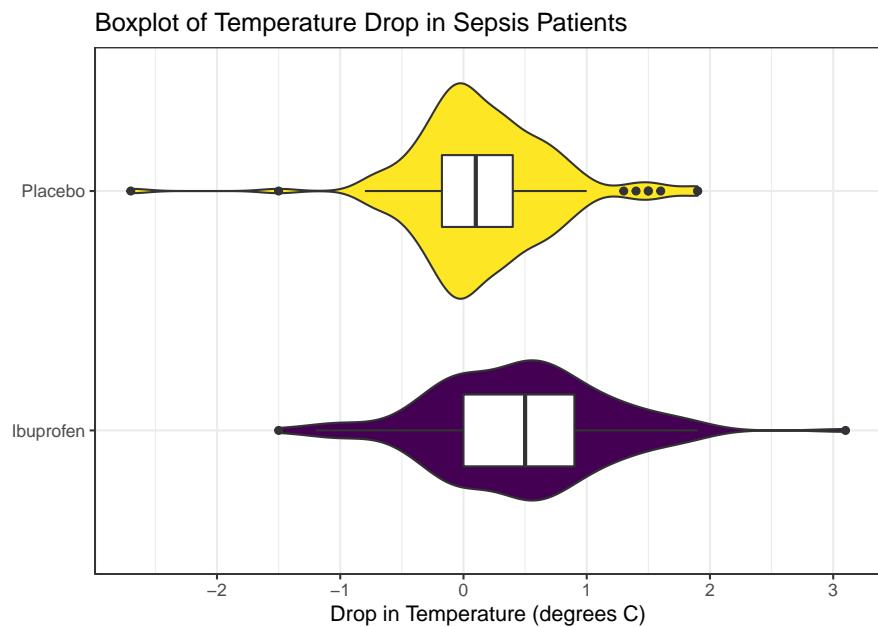
The placebo-controlled, double-blind randomized clinical trial, especially if pre-registered, is often considered the best feasible study for assessing the effectiveness of a treatment. While that's not always true, it is a very solid design. The primary caveat is that the patients who are included in such trials are rarely excellent representations of the population of potentially affected patients as a whole.

19.3 Exploratory Data Analysis

Consider the following boxplot with violin of the `temp_drop` data within each `treat` group.

```
ggplot(sepsis, aes(x = treat, y = temp_drop, fill = treat)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "white") +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Boxplot of Temperature Drop in Sepsis Patients",
       x = "", y = "Drop in Temperature (degrees C)") +
```

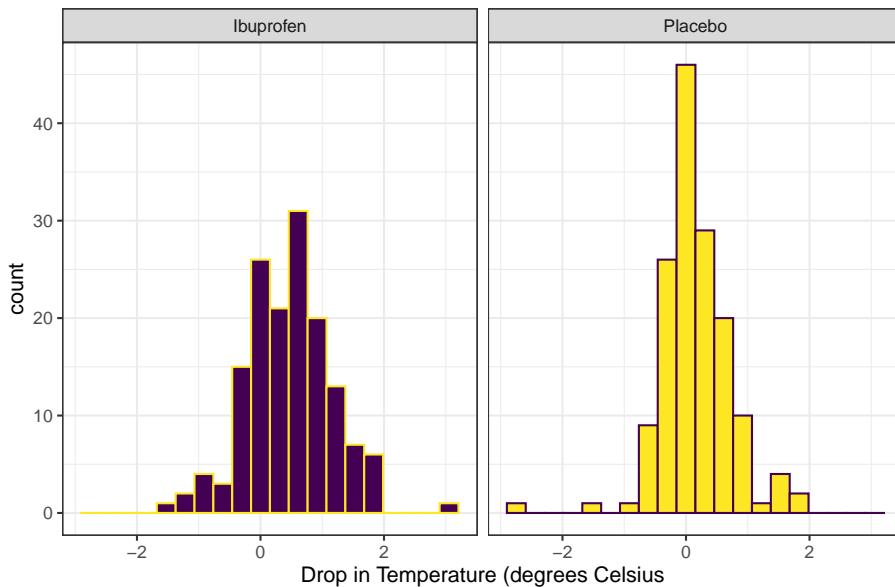
```
coord_flip() +
theme_bw()
```



Next, we'll consider faceted histograms of the data.

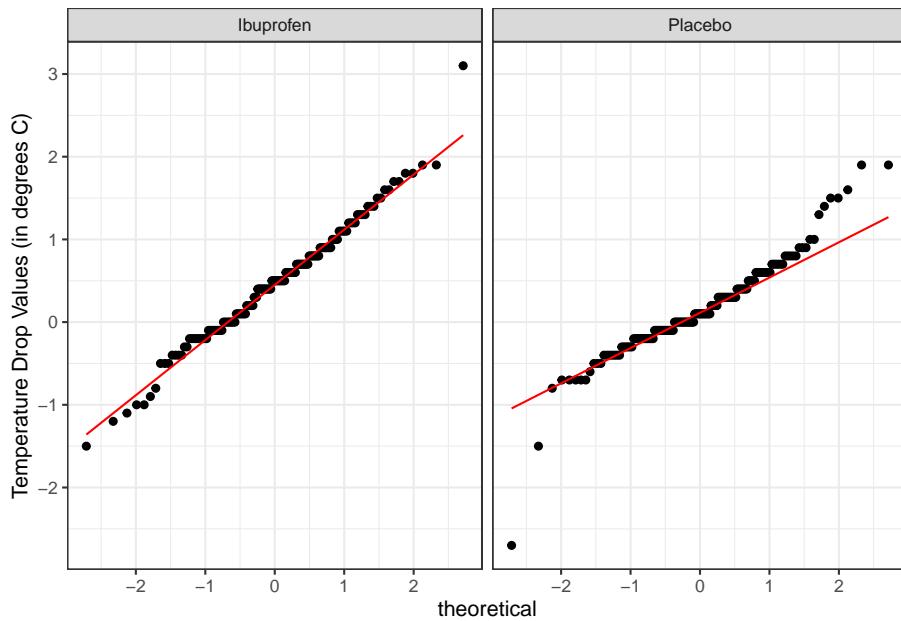
```
ggplot(sepsis, aes(x = temp_drop, fill = treat, color = treat)) +
  geom_histogram(bins = 20) +
  scale_fill_viridis_d() +
  scale_color_viridis_d(direction = -1) +
  guides(fill = FALSE, color = FALSE) +
  labs(title = "Histograms of Temperature Drop in Sepsis Patients",
       x = "Drop in Temperature (degrees Celsius)") +
  theme_bw() +
  facet_wrap(~ treat)
```

Histograms of Temperature Drop in Sepsis Patients



Here's a pair of Normal Q-Q plots. It's not hard to use a Normal model to approximate the Ibuprofen data, but such a model is probably not a good choice for the Placebo results.

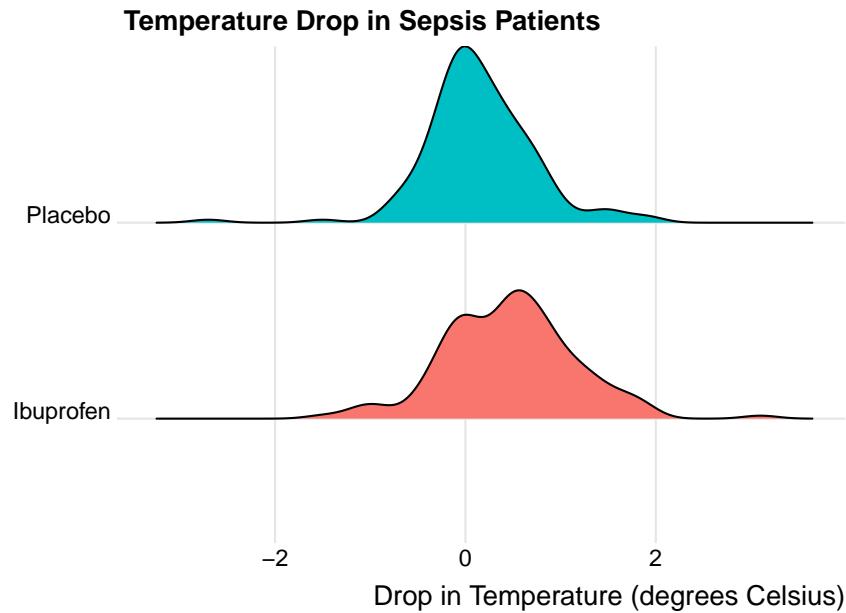
```
ggplot(sepsis, aes(sample = temp_drop)) +
  geom_qq() + geom_qq_line(col = "red") +
  theme_bw() +
  facet_wrap(~ treat) +
  labs(y = "Temperature Drop Values (in degrees C)")
```



We'll could perhaps also look at a ridgeline plot.

```
ggplot(sepsis, aes(x = temp_drop, y = treat, fill = treat)) +
  ggridges::geom_density_ridges(scale = 0.9) +
  guides(fill = FALSE) +
  labs(title = "Temperature Drop in Sepsis Patients",
       x = "Drop in Temperature (degrees Celsius)", y = "") +
  ggridges::theme_ridges()
```

Picking joint bandwidth of 0.182



The center of the ibuprofen distribution is shifted a bit towards the more positive (greater improvement) direction, it seems, than is the distribution for the placebo patients. This conclusion matches what we see in some key numerical summaries, within the treatment groups.

```
mosaic::favstats(temp_drop ~ treat, data = sepsis)
```

treat	min	Q1	median	Q3	max	mean	sd	n	missing
Ibuprofen	-1.5	0.000	0.5	0.9	3.1	0.464	0.688	150	0
Placebo	-2.7	-0.175	0.1	0.4	1.9	0.153	0.571	150	0

19.4 Estimating the Difference in Population Means

Next, we will build a point estimate and 90% confidence interval for the difference between the mean `temp_drop` if treated with Ibuprofen and the mean `temp_drop` if treated with Placebo. We'll use a regression model with a single predictor (the `treat` group) to do this.

```
model_sep <- lm(temp_drop ~ treat == "Ibuprofen", data = sepsis)

tidy(model_sep, conf.int = TRUE, conf.level = 0.90) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	0.153	0.052	2.96	0.003	0.068	0.238
treat == "Ibuprofen"	0.311	0.073	4.27	0.000	0.191	0.432

The point estimate for the “Ibuprofen - Placebo” difference in population means is 0.311 degrees C, and the 90% confidence interval is (0.191, 0.432) degrees C.

We could also have run the model like this:

```
model_sep2 <- lm(temp_drop ~ treat, data = sepsis)

tidy(model_sep2, conf.int = TRUE, conf.level = 0.90) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	0.464	0.052	8.99	0	0.379	0.549
treatPlacebo	-0.311	0.073	-4.27	0	-0.432	-0.191

and would therefore conclude that the *Placebo - Ibuprofen* difference was estimated as -0.311, with 90% confidence interval (-0.432, -0.191), which is of course equivalent to our previous estimate.

This is just one possible way for us to estimate the difference between population means. We'll see more in Chapter 20.

19.5 Categorizing the Outcome

Suppose we were interested in comparing the percentage of patients in each arm of the trial (Ibuprofen vs. Placebo) that showed an improvement in their temperature (`temp_drop > 0`). To build the cross-tabulation of interest, we could create a new variable, called `dropped` which indicates whether the subject's temperature dropped, and then use `tabyl`.

```
sepsis <- sepsis %>%
  mutate(dropped = ifelse(temp_drop > 0, "Drop", "No Drop"))

sepsis %>% tabyl(treat, dropped)

  treat Drop No Drop
Ibuprofen 107    43
  Placebo   80    70
```

Our primary interest is in comparing the percentage of Ibuprofen patients whose temperature dropped to the percentage of Placebo patients whose temperature dropped.

```
sepsis %>% tabyl(treat, dropped) %>%
  adorn_totals() %>%
  adorn_percentages(denom = "row") %>%
```

```

adorn_pct_formatting(digits = 1) %>%
  adorn_ns(position = "front")

  treat      Drop    No Drop
Ibuprofen 107 (71.3%) 43 (28.7%)
  Placebo   80 (53.3%) 70 (46.7%)
  Total    187 (62.3%) 113 (37.7%)

```

19.6 Estimating the Difference in Proportions

In our sample, 71.3% of the Ibuprofen subjects, and 53.3% of the Placebo subjects, experienced a drop in temperature. So our *point estimate* of the difference in percentages would be 18.0 percentage points, but we will usually set this instead in terms of proportions, so that the difference is 0.180.

Now, we'll find a confidence interval for that difference, which we can do in several ways, including the `twoby2` function in the `Epi` package.

```
sepsis %$% table(treat, dropped) %>% Epi::twoby2(alpha = 0.10)
```

2 by 2 table analysis:

```

-----  

Outcome : Drop  

Comparing : Ibuprofen vs. Placebo  

      Drop  No Drop    P(Drop) 90% conf. interval  

Ibuprofen 107      43      0.713    0.649    0.770  

Placebo     80      70      0.533    0.466    0.599  

                           90% conf. interval  

      Relative Risk: 1.34      1.1492    1.557  

      Sample Odds Ratio: 2.18      1.4583    3.251  

      Conditional MLE Odds Ratio: 2.17      1.4177    3.344  

      Probability difference: 0.18      0.0881    0.268  

      Exact P-value: 0.0019  

      Asymptotic P-value: 0.0014
-----
```

While there is a lot of additional output here, we'll look for now just at the Probability difference row, where we see the point estimate (0.180) and the 90% confidence interval estimate for the difference in proportions (0.088, 0.268) comparing Ibuprofen vs. Placebo for the outcome of Dropping in Temperature.

More on estimation of the difference in population proportions will be found in Chapter 21.

Chapter 20

Comparing Means/Quantities using Two Independent Samples

Here, we'll consider the problem of estimating a confidence interval to describe the difference in population means (or medians) based on a comparison of two samples of quantitative data, gathered using an independent samples design. Specifically, we'll look at the randomized controlled trial of Ibuprofen in Sepsis patients described in Chapter 19.

In that trial, 300 patients meeting specific criteria (including elevated temperature) for a diagnosis of sepsis were randomly assigned to either the Ibuprofen group (150 patients) and 150 to the Placebo group. Group information (our exposure) is contained in the `treat` variable. The key outcome of interest to us was `temp_drop`, the change in body temperature (in $^{\circ}\text{C}$) from baseline to 2 hours later, so that positive numbers indicate drops in temperature (a good outcome.) Here's the comparison of `temp_drop` summary statistics in the two `treat` groups.

```
mosaic::favstats(temp_drop ~ treat, data = sepsis)
```

	treat	min	Q1	median	Q3	max	mean	sd	n	missing
1	Ibuprofen	-1.5	0.000	0.5	0.9	3.1	0.464	0.688	150	0
2	Placebo	-2.7	-0.175	0.1	0.4	1.9	0.153	0.571	150	0

20.1 t-based CI for population mean difference $\mu_1 - \mu_2$ from Independent Samples

20.1.1 The Pooled t procedure

The most commonly used t-procedure for building a confidence interval assumes not only that each of the two populations being compared follows a Normal distribution, but also that they have the same population variance. This is the pooled t-test, and it is what people usually mean when they describe a two-sample t test.

```
sepsis %$% t.test(temp_drop ~ treat,
                    conf.level = 0.90,
                    alt = "two.sided",
                    var.equal = TRUE)
```

```
Two Sample t-test

data: temp_drop by treat
t = 4, df = 298, p-value = 3e-05
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 0.191 0.432
sample estimates:
mean in group Ibuprofen   mean in group Placebo
          0.464                  0.153
```

Or, we can use `tidy` on this object:

```
tt1 <- sepsis %$% t.test(temp_drop ~ treat,
                         conf.level = 0.90,
                         alt = "two.sided",
                         var.equal = TRUE)
tidy(tt1)

# A tibble: 1 x 9
  estimate1 estimate2 statistic p.value parameter conf.low conf.high method
    <dbl>     <dbl>     <dbl>      <dbl>     <dbl>     <dbl>     <chr>
1     0.464     0.153     4.27 2.68e-5      298     0.191     0.432 "Two-
# ... with 1 more variable: alternative <chr>
```

20.1.2 Using linear regression to obtain a pooled t confidence interval

A linear regression model, using the same outcome and predictor (group) as the pooled t procedure, produces the same confidence interval, again, under the assumption that the two populations we are comparing follow a Normal distribution with the same (population) variance.

```
model1 <- lm(temp_drop ~ treat, data = sepsis)

tidy(model1, conf.int = TRUE, conf.level = 0.90)

# A tibble: 2 x 7
  term      estimate std.error statistic p.value conf.low conf.high
  <chr>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>     <dbl>
1 (Intercept) 0.464     0.0516     8.99 2.91e-17   0.379     0.549
2 treatPlacebo -0.311     0.0730    -4.27 2.68e- 5  -0.432    -0.191
```

We see that our point estimate from the linear regression model is that the difference in `temp_drop` is -0.311, where Ibuprofen subjects have higher `temp_drop` values than do Placebo subjects, and that the 90% confidence interval for this difference ranges from -0.432 to -0.191.

We can obtain a t-based confidence interval for each of the parameter estimates in a linear model directly using `confint`. Linear models usually summarize only the estimate and standard error. Remember that a reasonable approximation in large samples to a 95% confidence interval for a regression estimate (slope or intercept) can be obtained from $\text{estimate} \pm 2 * \text{standard error}$.

```
summary(model1)
```

```
Call:
lm(formula = temp_drop ~ treat, data = sepsis)

Residuals:
    Min      1Q  Median      3Q      Max 
-2.8527 -0.3640 -0.0527  0.3473  2.6360 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.46400   0.05160   8.99 < 2e-16 ***
treatPlacebo -0.31130   0.07300  -4.27 2.7e-05 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.632 on 298 degrees of freedom
Multiple R-squared:  0.0575,    Adjusted R-squared:  0.0544
```

```
F-statistic: 18.2 on 1 and 298 DF, p-value: 2.68e-05
```

So, in the case of the `treatPlacebo` estimate, we can obtain an approximate 95% confidence interval with $-0.311 \pm 2 \times 0.073$ or $(-0.457, -0.165)$. Compare this to the 95% confidence interval available from the model directly, shown in the tidied output above, or with the `confint` command below, and you'll see only a small difference.

```
confint(model1, level = 0.95)
```

	2.5 %	97.5 %
(Intercept)	0.362	0.566
treatPlacebo	-0.455	-0.168

20.1.3 The Welch t procedure

The default confidence interval based on the t test for independent samples in R uses something called the Welch test, in which the two populations being compared are not assumed to have the same variance. Each population is assumed to follow a Normal distribution.

```
sepsis %$% t.test(temp_drop ~ treat,
                    conf.level = 0.90,
                    alt = "two.sided")
```

Welch Two Sample t-test

```
data: temp_drop by treat
t = 4, df = 288, p-value = 3e-05
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
0.191 0.432
sample estimates:
mean in group Ibuprofen mean in group Placebo
0.464 0.153
```

Tidying works in this situation, too.

```
tt0 <- sepsis %$% t.test(temp_drop ~ treat,
                         conf.level = 0.90,
                         alt = "two.sided")

tidy(tt0)

# A tibble: 1 x 10
estimate estimate1 estimate2 statistic p.value parameter conf.low
<dbl>     <dbl>     <dbl>     <dbl>    <dbl>      <dbl>     <dbl>
```

20.2. BOOTSTRAP CI FOR $MU_1 - \mu_2$ FROM INDEPENDENT SAMPLES 331

```
1    0.311    0.464    0.153    4.27 2.71e-5    288.    0.191
# ... with 3 more variables: conf.high <dbl>, method <chr>,
#   alternative <chr>
```

When there is a *balanced design*, that is, when the same number of observations appear in each of the two samples, then the Welch t test and the Pooled t test produce the same confidence interval. Differences appear if the sample sizes in the two groups being compared are different.

20.2 Bootstrap CI for $\mu_1 - \mu_2$ from Independent Samples

The `bootdif` function contained in the `Love-boost.R` script, that we will use in this setting is a slightly edited version of the function at <http://biostat.mc.vanderbilt.edu/wiki/Main/BootstrapMeansSoftware>. Note that this approach uses a comma to separate the outcome variable (here, `temp_drop`) from the variable identifying the exposure groups (here, `treat`).

```
set.seed(431212)

sepsis %$% bootdif(temp_drop, treat, conf.level = 0.90)
```

Mean Difference	0.05	0.95
-0.311	-0.431	-0.183

20.3 Wilcoxon-Mann-Whitney “Rank Sum” CI from Independent Samples

As in the one-sample case, a rank-based alternative attributed to Wilcoxon (and sometimes to Mann and Whitney) provides a two-sample comparison of the pseudomedians in the two `treat` groups in terms of `temp_drop`. This is called a **rank sum** test, rather than the Wilcoxon **signed rank** test that is used for inference about a single sample. Here's the resulting 90% confidence interval for the difference in pseudomedians.

```
wt <- sepsis %$% wilcox.test(temp_drop ~ treat,
                           conf.int = TRUE, conf.level = 0.90,
                           alt = "two.sided")

wt
```

Wilcoxon rank sum test with continuity correction

```

data: temp_drop by treat
W = 14614, p-value = 7e-06
alternative hypothesis: true location shift is not equal to 0
90 percent confidence interval:
 0.2 0.4
sample estimates:
difference in location
 0.3

tidy(wt)

# A tibble: 1 x 7
  estimate statistic p.value conf.low conf.high method      alternative
    <dbl>     <dbl>    <dbl>     <dbl>     <dbl>   <chr>      <chr>
1     0.300     14614.  7.28e-6     0.200     0.400 Wilcoxon rank~ two.sided

```

20.4 Summary: Specifying A Two-Sample Study Design

These questions will help specify the details of the study design involved in any comparison of two populations on a quantitative outcome, perhaps with means.

1. What is the outcome under study?
2. What are the (in this case, two) treatment/exposure groups?
3. Were the data collected using matched / paired samples or independent samples?
4. Are the data a random sample from the population(s) of interest? Or is there at least a reasonable argument for generalizing from the sample to the population(s)?
5. What is the significance level (or, the confidence level) we require here?
6. Are we doing one-sided or two-sided testing/confidence interval generation?
7. If we have paired samples, did pairing help reduce nuisance variation?
8. If we have paired samples, what does the distribution of sample paired differences tell us about which inferential procedure to use?
9. If we have independent samples, what does the distribution of each individual sample tell us about which inferential procedure to use?

20.4.1 For the `sepsis` study

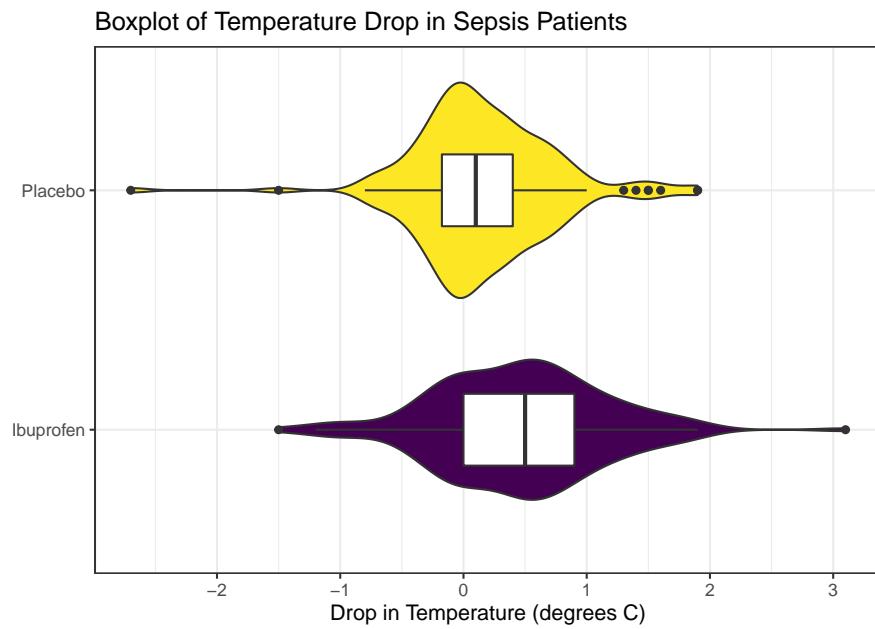
1. The outcome is `temp_drop`, the change in body temperature (in $^{\circ}\text{C}$) from baseline to 2 hours later, so that positive numbers indicate drops in temperature (a good outcome.)
2. The groups are **Ibuprofen** and **Placebo** as contained in the `treat` variable in the `sepsis` tibble.

3. The data were collected using independent samples. The Ibuprofen subjects are not matched or linked to individual Placebo subjects - they are separate groups.
4. The subjects of the study aren't drawn from a random sample of the population of interest, but they are randomly assigned to their respective treatments (Ibuprofen and Placebo) which will provide the reasoned basis for our inferences.
5. We'll use a 10% significance level (or 90% confidence level) in this setting, as we did in our previous work on these data.
6. We'll use a two-sided testing and confidence interval approach.

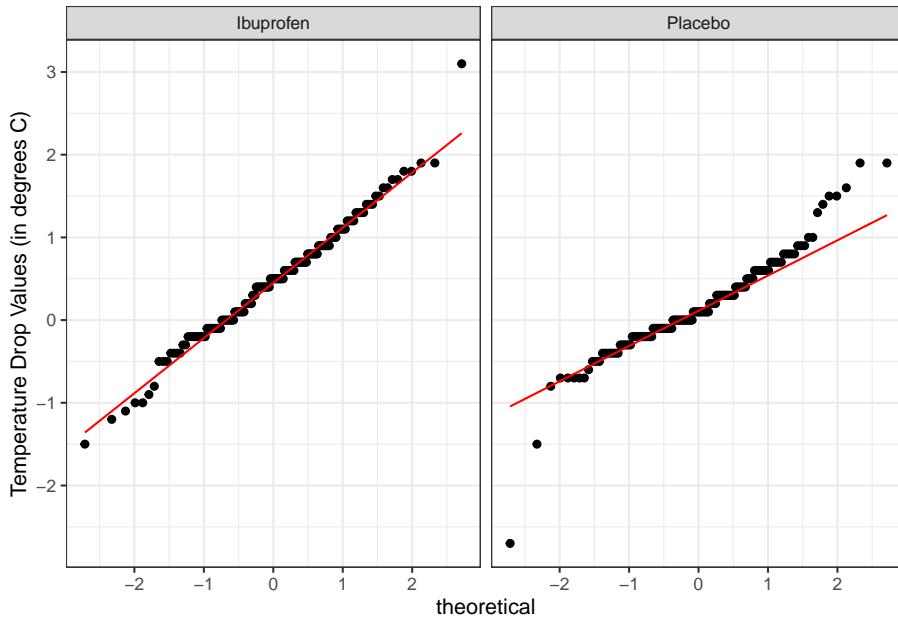
Questions 7 and 8 don't apply, because these are independent samples of data, rather than paired samples.

To address question 9, we'll need to look at the data in each sample. We'll repeat the boxplot and Normal Q-Q plots from Chapter 19, that allow us to assess the Normality of the distributions of (separately) the `temp_drop` results in the Ibuprofen and Placebo groups.

```
ggplot(sepsis, aes(x = treat, y = temp_drop, fill = treat)) +
  geom_violin() +
  geom_boxplot(width = 0.3, fill = "white") +
  scale_fill_viridis_d() +
  guides(fill = FALSE) +
  labs(title = "Boxplot of Temperature Drop in Sepsis Patients",
       x = "", y = "Drop in Temperature (degrees C)") +
  coord_flip() +
  theme_bw()
```



```
ggplot(sepsis, aes(sample = temp_drop)) +  
  geom_qq() + geom_qq_line(col = "red") +  
  theme_bw() +  
  facet_wrap(~ treat) +  
  labs(y = "Temperature Drop Values (in degrees C)")
```



From these plots we conclude that the data in the Ibuprofen sample follow a reasonably Normal distribution, but this isn't quite as true for the Placebo sample. It's hard to know whether the apparent Placebo group outliers will affect whether the Normal distribution assumption is reasonable, so we can see if the confidence intervals change much when we *don't* assume Normality (for instance, comparing the bootstrap to the t-based approaches), as a way of understanding whether a Normal model has a large impact on our conclusions.

20.4.2 Sepsis Estimation Results

Here's a set of confidence interval estimates (we'll use 90% confidence here) using the methods discussed in this Chapter.

```
mosaic::favstats(temp_drop ~ treat, data = sepsis)

  treat min      Q1 median      Q3 max      mean      sd     n missing
1 Ibuprofen -1.5    0.000    0.5 0.9 3.1 0.464 0.688 150       0
2 Placebo   -2.7   -0.175    0.1 0.4 1.9 0.153 0.571 150       0

s_pooled_t_test <- sepsis %$% t.test(temp_drop ~ treat,
                                      conf.level = 0.90,
                                      alt = "two.sided",
                                      var.equal = TRUE)

tidy(s_pooled_t_test) %>%
```

```

  select(conf.low, conf.high)

# A tibble: 1 x 2
  conf.low conf.high
  <dbl>     <dbl>
1 0.191     0.432

s_welch_t_test <- sepsis %$% t.test(temp_drop ~ treat,
                                      conf.level = 0.90,
                                      alt = "two.sided",
                                      var.equal = FALSE)

tidy(s_welch_t_test) %>%
  select(estimate, conf.low, conf.high)

# A tibble: 1 x 3
  estimate conf.low conf.high
  <dbl>     <dbl>     <dbl>
1 0.311     0.191     0.432

s_wilcoxon_test <- sepsis %$% wilcox.test(temp_drop ~ treat,
                                             conf.int = TRUE, conf.level = 0.90,
                                             alt = "two.sided")

tidy(s_wilcoxon_test) %>%
  select(estimate, conf.low, conf.high)

# A tibble: 1 x 3
  estimate conf.low conf.high
  <dbl>     <dbl>     <dbl>
1 0.300     0.200     0.400

set.seed(431212)
s_bootstrap <- sepsis %$% bootdif(temp_drop, treat,
                                     conf.level = 0.90)

s_bootstrap

Mean Difference          0.05          0.95
                         -0.311        -0.431       -0.183

```

Procedure	Compares...	Point Estimate	90% CI
Pooled t	Means	0.311	(0.191, 0.432)
Welch t	Means	0.311	(0.191, 0.432)
Bootstrap	Means	0.311	(0.183, 0.431)
Wilcoxon rank sum	Pseudo-Medians	0.3	(0.2, 0.4)

What conclusions can we draw in this setting?

Chapter 21

Comparing Rates/Proportions with Two Independent Samples

Often, when analyzing data from an independent samples design, we are interested in comparing the proportions of subjects that achieve some outcome, across two levels of an exposure, or treatment. In this circumstance, we will first summarize the available data in terms of a 2x2 cross-tabulation, and then apply a series of inferential methods for 2x2 tables to obtain point and interval estimates of interest.

21.1 A First Example: Ibuprofen and Sepsis Trial

As we saw in Chapter 19, we are interested in comparing the percentage of patients in each arm of the trial (Ibuprofen vs. Placebo) that showed an improvement in their temperature (`temp_drop > 0`). Our primary interest is in comparing the percentage of Ibuprofen patients whose temperature dropped to the percentage of Placebo patients whose temperature dropped.

- We can summarize the data behind the two proportions we are comparing in a contingency table with two rows which identify the exposure or treatment of interest, and two columns to represent the outcomes of interest.
- In this case, we are comparing two groups of subjects based on treatment (`treat`): those who received Ibuprofen and those who received a placebo. The outcome of interest is whether the subject's temperature dropped (`temp_drop > 0`), or not.

- In the table, we place the frequency for each combination of a row and a column.
- The rows need to be mutually exclusive and collectively exhaustive: each patient must either receive Ibuprofen or Placebo. Similarly, the columns must meet the same standard: every patient's temperature either drops or does not drop.

Here's the contingency table.

```
sepsis <- sepsis %>%
  mutate(dropped = ifelse(temp_drop > 0, "Drop", "No Drop"))

sepsis %>% tabyl(treat, dropped) %>%
  adorn_totals() %>%
  adorn_percentages(denom = "row") %>%
  adorn_pct_formatting(digits = 1) %>%
  adorn_ns(position = "front")
```

	Drop	No Drop
Ibuprofen	107 (71.3%)	43 (28.7%)
Placebo	80 (53.3%)	70 (46.7%)
Total	187 (62.3%)	113 (37.7%)

In our sample, we observe 71.3% of the 150 Ibuprofen subjects with a positive `temp_drop` as compared to 53.3% of the 150 Placebo subjects. We want to compare these two probabilities (represented using proportions as 0.713 vs. 0.533) to estimate the size of the difference between the proportions with a point estimate and 90% confidence interval.

But there are other comparisons we could make, too. The tricky part is that we have multiple ways to describe the relationship between treatment and outcome. We might compare outcome “risks” directly using the difference in probabilities, or the ratio of the two probabilities, or we might convert the risks to odds, and compare the ratio of those odds. In any case, we’ll get different point estimates and confidence intervals, all of which will help us make conclusions about the evidence available in this trial speaking to differences between Ibuprofen and Placebo.

21.2 Relating a Treatment to an Outcome

The question of interest is whether the percentage of subjects whose temperature dropped is different (and probably larger) in the subjects who received Ibuprofen than in those who received the Placebo.

Treatment Arm	Dropped	Did Not Drop	Total	Proportion who dropped
Ibuprofen	107	43	150	0.713

Treatment Arm	Dropped	Did Not Drop	Total	Proportion who dropped
Placebo	80	70	150	0.533

In other words, what is the relationship between the treatment and the outcome?

21.3 Definitions of Probability and Odds

- Proportion = Probability = Risk of the trait = number with trait / total
- Odds of having the trait = (number with the trait / number without the trait) to 1

If p is the proportion of subjects with a trait, then the **odds** of having the trait are $\frac{p}{1-p}$ to 1.

So, the probability of a good result (temperature drop) in this case is $\frac{107}{150} = 0.713$ in the Ibuprofen group. The **odds** of a good result are thus $\frac{0.713}{1-0.713} = 2.484$ to 1 in the Ibuprofen group.

Treatment Arm	Dropped	Did Not Drop	Total	Pr(dropped)	Odds(dropped)
Ibuprofen	107	43	150	0.713	2.484
Placebo	80	70	150	0.533	1.141

21.4 Defining the Relative Risk

Among the Ibuprofen subjects, the risk of a good outcome (drop in temperature) is 71.3% or, stated as a proportion, 0.713. Among the Placebo subjects, the risk of a good outcome is 53.3% or, stated as a proportion, 0.533.

Our “crude” estimate of the **relative risk** of a good outcome for Ibuprofen subjects as compared to Placebo subjects, is the ratio of these two risks, or $0.713/0.533 = 1.338$

- The fact that this relative risk is greater than 1 indicates that the probability of a good outcome is higher for Ibuprofen subjects than for Placebo subjects.
- A relative risk of 1 would indicate that the probability of a good outcome is the same for Ibuprofen subjects and for Placebo subjects.
- A relative risk less than 1 would indicate that the probability of a good outcome is lower for Ibuprofen subjects than for Placebo subjects.

21.5 Defining the Risk Difference

Our “crude” estimate of the **risk difference** of a good outcome for Ibuprofen subjects as compared to Placebo subjects, is $0.713 - 0.533 = 0.180$ or 18.0 percentage points.

- The fact that this risk difference is greater than 0 indicates that the probability of a good outcome is higher for Ibuprofen subjects than for Placebo subjects.
- A risk difference of 0 would indicate that the probability of a good outcome is the same for Ibuprofen subjects and for Placebo subjects.
- A risk difference less than 0 would indicate that the probability of a good outcome is lower for Ibuprofen subjects than for Placebo subjects.

21.6 Defining the Odds Ratio, or the Cross-Product Ratio

Among the Ibuprofen subjects, the odds of a good outcome (temperature drop) are 2.484. Among the placebo subjects, the odds of a good outcome (temperature drop) are 1.141.

So our “crude” estimate of the **odds ratio** of a good outcome for Ibuprofen subjects as compared to placebo subjects, is $2.484 / 1.141 = 2.18$

Another way to calculate this odds ratio is to calculate the **cross-product ratio**, which is equal to $(a \times d) / (b \times c)$, for the 2 by 2 table with counts specified as shown:

A Generic Table	Good Outcome	Bad Outcome
Treatment Group 1	a	b
Treatment Group 2	c	d

So, for our table, we have $a = 107$, $b = 43$, $c = 80$, and $d = 70$, so the cross-product ratio is $\frac{107 \times 70}{43 \times 80} = \frac{7490}{3440} = 2.18$. As expected, this is the same as the “crude” odds ratio estimate.

- The fact that this odds ratio risk is greater than 1 indicates that the odds of a good outcome are higher for Ibuprofen subjects than for Placebo subjects.
- An odds ratio of 1 would indicate that the odds of a good outcome are the same for Ibuprofen subjects and for Placebo subjects.
- An odds ratio less than 1 would indicate that the odds of a good outcome are lower for Ibuprofen subjects than for Placebo subjects.

So, we have several different ways to compare the outcomes across the treatments. Are these differences and ratios large enough to rule out chance?

21.7 Comparing Rates in a 2x2 Table

What is the relationship between the treatment (Ibuprofen vs. Placebo) and the outcome (drop in temperature) in the following two-by-two table?

21.8 The twobytwo function in R

I built the `twobytwo` function in R (based on existing functions in the `Epi` library, which you need to have in your installed packages list in order for this to work) to do the work for this problem. All that is required is a single command, and a two-by-two table in standard epidemiological format (with the outcomes in the columns, and the treatments in the rows.)

Treatment Arm	Dropped	Did Not Drop
Ibuprofen	107	43
Placebo	80	70

The command just requires you to read off the cells of the table, followed by the labels for the two treatments, then the two outcomes, then a specification of the names of the rows (exposures) and columns (outcomes) from the table, and a specification of the confidence level you desire. We'll use 90% here.

The resulting output follows.

```
twobytwo(107, 43, 80, 70,
         "Ibuprofen", "Placebo", "Dropped", "No Drop",
         conf.level = 0.90)
```

2 by 2 table analysis:

Outcome : Dropped
Comparing : Ibuprofen vs. Placebo

	Dropped	No Drop	P(Dropped)	90% conf. interval
Ibuprofen	107	43	0.713	0.649 0.770
Placebo	80	70	0.533	0.466 0.599

	90% conf. interval		
Relative Risk:	1.34	1.1492	1.557

```

Sample Odds Ratio: 2.18    1.4583    3.251
Conditional MLE Odds Ratio: 2.17    1.4177    3.344
Probability difference: 0.18    0.0881    0.268

Exact P-value: 0.0019
Asymptotic P-value: 0.0014
-----
```

21.8.1 Standard Epidemiological Format

This table is in **standard epidemiological format**, which means that:

- The rows of the table describe the “treatment” (which we’ll take here to be `treat`). The more interesting (sometimes also the more common) “treatment” is placed in the top row. That’s Ibuprofen here.
- The columns of the table describe the “outcome” (which we’ll take here to be whether the subject’s temperature dropped.) Typically, the more common “outcome” is placed to the left.

21.8.2 Outcome Probabilities and Confidence Intervals Within the Treatment Groups

The `twobytwo` output starts with estimates of the probability (risk) of a “Dropped” outcome among subjects who fall into the two treatment groups (Ibuprofen or Placebo), along with 90% confidence intervals for each of these probabilities.

`2 by 2 table analysis:`

```

Outcome : Dropped
Comparing : Ibuprofen vs. Placebo
```

	Dropped	No Drop	P(Dropped)	90% conf.	interval
Ibuprofen	107	43	0.7133	0.6490	0.7701
Placebo	80	70	0.5333	0.4661	0.5993

The conditional probability of a temperature drop given that the subject is in the Ibuprofen group, is symbolized as $\Pr(\text{Dropped} \mid \text{Ibuprofen}) = 0.7133$, and the 90% confidence interval around that proportion is $(0.6490, 0.7701)$.

- Note that these two confidence intervals fail to overlap, and so we expect to see a fairly large difference in the estimated probability of a temperature drop when we compare Ibuprofen to Placebo.

21.8.3 Relative Risk, Odds Ratio and Risk Difference, with Confidence Intervals

These elements are followed by estimates of the relative risk, odds ratio, and risk difference, each with associated 90% confidence intervals.

	90% conf. interval		
Relative Risk:	1.3375	1.1492	1.5567
Sample Odds Ratio:	2.1773	1.4583	3.2509
Conditional MLE Odds Ratio:	2.1716	1.4177	3.3437
Probability difference:	0.1800	0.0881	0.2677

- The **relative risk**, or the ratio of $P(\text{Temperature Drop} \mid \text{Ibuprofen})$ to $P(\text{Temperature Drop} \mid \text{Placebo})$, is shown first. Note that the 90% confidence interval is entirely greater than 1.
- The **odds ratio** is presented using two different definitions (the sample odds ratio is the cross-product ratio we mentioned earlier). Note that the 90% confidence interval using either approach is entirely greater than 1.
- The **probability (or risk) difference** [$P(\text{Temperature Drop} \mid \text{Ibuprofen}) - P(\text{Temperature Drop} \mid \text{Placebo})$] is presented last. Note that the 90% confidence interval is entirely greater than 0.
- Note carefully that if there had been no difference between Ibuprofen and Placebo, the relative risk and odds ratios would be 1, but the probability difference would be zero.

21.9 Estimating a Rate More Accurately: Use $(x + 1)/(n + 2)$ rather than x/n

Suppose you have some data involving n independent tries, with x successes. A natural estimate of the “success rate” in the data is x / n .

But, strangely enough, it turns out this isn’t an entirely satisfying estimator. Alan Agresti provides substantial motivation for the $(x + 1)/(n + 2)$ estimate as an alternative¹. This is sometimes called a *Bayesian augmentation*.

- The big problem with x / n is that it estimates $p = 0$ or $p = 1$ when $x = 0$ or $x = n$.
- It’s also tricky to compute confidence intervals at these extremes, since the usual standard error for a proportion, $\sqrt{np(1 - p)}$, gives zero, which isn’t quite right.
- $(x + 1)/(n + 2)$ is much cleaner, especially when you build a confidence interval for the rate.

¹This note comes largely from a May 15 2007 entry in Andrew Gelman’s blog at <http://andrewgelman.com/2007/05/15>

- The only place where $(x + 1)/(n + 2)$ will go wrong (as in the SAIFS approach) is if n is small and the true probability is very close to 0 or 1.

For example, if $n = 10$, and p is 1 in a million, then x will almost certainly be zero, and an estimate of $1/12$ is much worse than the simple $0/10$. However, how big a deal is this? If p might be 1 in a million, you're not going to estimate it with a $n = 10$ experiment².

Applying this method to our Ibuprofen and Sepsis Trial data, we would simply add one to each frequency in the main four cells in our 2×2 table.

So instead of using

```
twobytwo(107, 43, 80, 70,
         "Ibuprofen", "Placebo", "Dropped", "No Drop",
         conf.level = 0.90)
```

the Bayesian augmentation would encourage us to look at

```
twobytwo(108, 44, 81, 71,
         "Ibuprofen", "Placebo", "Dropped", "No Drop",
         conf.level = 0.90)
```

2 by 2 table analysis:

Outcome : Dropped
Comparing : Ibuprofen vs. Placebo

	Dropped	No Drop	P(Dropped)	90% conf. interval
Ibuprofen	108	44	0.711	0.646 0.767
Placebo	81	71	0.533	0.466 0.599

	90% conf. interval	
Relative Risk:	1.333	1.1463 1.551
Sample Odds Ratio:	2.151	1.4457 3.202
Conditional MLE Odds Ratio:	2.146	1.4061 3.292
Probability difference:	0.178	0.0863 0.265

	Exact P-value:	Asymptotic P-value:
	0.0020	0.0015

As you can see, the odds ratio and relative risk estimates are (a little) closer to 1, and the probability difference is also a little closer to 0. The Bayesian augmentation provides a slightly more conservative set of estimates of the impact of Ibuprofen as compared to Placebo.

²Andrew Gelman's example is "I'm not going to try ten 100-foot golf putts, miss all of them, and then estimate my probability of success as $1/12$."

21.10. A SECOND EXAMPLE: EBOLA VIRUS DISEASE STUDY, AGAIN³⁴⁷

It is likely that the augmented version is a more accurate estimate here, but the two estimates will be comparable, generally, so long as either (a) the sample size in each exposure group is more than, say, 30 subjects, and/or (b) the sample probability of the outcome is between 10% and 90% in each exposure group.

21.10 A Second Example: Ebola Virus Disease Study, again

For instance, recall the Ebola Virus Disease study from the *New England Journal of Medicine* that we described in Chapter 18. Suppose we want to compare the proportion of deaths among cases that had a definitive outcome who were hospitalized to the proportion of deaths among cases that had a definitive outcome who were not hospitalized.

The article suggests that of the 1,737 cases with a definitive outcome, there were 1,153 hospitalized cases. Across those 1,153 hospitalized cases, 741 people (64.3%) died, which means that across the remaining 584 non-hospitalized cases, 488 people (83.6%) died.

Here is the initial contingency table, using only the numbers from the previous paragraph.

Initial Ebola Table	Deceased	Alive	Total
Hospitalized	741	—	1153
Not Hospitalized	488	—	584
Total			1737

Now, we can use arithmetic to complete the table, since the rows and the columns are each mutually exclusive and collectively exhaustive.

Ebola 2x2 Table	Deceased	Alive	Total
Hospitalized	741	412	1153
Not Hospitalized	488	96	584
Total	1229	508	1737

We want to compare the fatality risk (probability of being in the deceased column) for the population of people in the hospitalized row to the population of people in the not hospitalized row.

We can run these data through R, using the Bayesian augmentation (adding a death and a survival to the hospitalized and also to the not hospitalized groups.) We'll use a 95% confidence level this time.

```
twobytwo(741+1, 412+1, 488+1, 96+1,
          "Hospitalized", "Not Hospitalized", "Deceased", "Alive",
          conf.level = 0.95)
```

2 by 2 table analysis:

Outcome : Deceased
Comparing : Hospitalized vs. Not Hospitalized

	Deceased	Alive	P(Deceased)	95% conf. interval
Hospitalized	742	413	0.642	0.614 0.670
Not Hospitalized	489	97	0.835	0.802 0.862

	95% conf. interval	
Relative Risk:	0.770	0.728 0.814
Sample Odds Ratio:	0.356	0.278 0.457
Conditional MLE Odds Ratio:	0.357	0.275 0.460
Probability difference:	-0.192	-0.232 -0.150

Exact P-value: 0.0000
Asymptotic P-value: 0.0000

I'll leave it as an exercise for you to interpret these results and draw some conclusions.

Chapter 22

A Paired Sample Study: Lead in the Blood of Children

One of the best ways to eliminate a source of variation and the errors of interpretation associated with it is through the use of matched pairs. Each subject in one group is matched as closely as possible by a subject in the other group. If a 45-year-old African-American male with hypertension is given a [treatment designed to lower their blood pressure], then we give a second, similarly built 45-year old African-American male with hypertension a placebo.

- Good (2005), section 5.2.4

22.1 The Lead in the Blood of Children Study

Morton et al. (1982) studied the absorption of lead into the blood of children. This was a matched-sample study, where the exposed group of interest contained 33 children of parents who worked in a battery manufacturing factory (where lead was used) in the state of Oklahoma. Specifically, each child with a lead-exposed parent was matched to another child of the same age, exposure to traffic, and living in the same neighborhood whose parents did not work in lead-related industries. So the complete study had 66 children, arranged in 33 matched pairs. The outcome of interest, gathered from a sample of whole blood from each of the children, was lead content, measured in mg/dl.

One motivation for doing this study is captured in the Abstract from Morton et al. (1982).

350CHAPTER 22. A PAIRED SAMPLE STUDY: LEAD IN THE BLOOD OF CHILDREN

It has been repeatedly reported that children of employees in a lead-related industry are at increased risk of lead absorption because of the high levels of lead found in the household dust of these workers.

The data are available in several places, including Table 5 of Pruzek and Helmreich (2009), in the BloodLead data set within the `PairedData` package in R, but we also make them available in the `bloodlead.csv` file. A table of the first few pairs of observations (blood lead levels for one child exposed to lead and the matched control) is shown below.

`bloodlead`

```
# A tibble: 33 x 3
  pair   exposed control
  <chr>    <dbl>    <dbl>
1 P01      38       16
2 P02      23       18
3 P03      41       18
4 P04      18       24
5 P05      37       19
6 P06      36       11
7 P07      23       10
8 P08      62       15
9 P09      31       16
10 P10     34       18
# ... with 23 more rows

  • In each pair, one child was exposed (to having a parent working in the factory) and the other was not.
  • Otherwise, though, each child was very similar to its matched partner.
  • The data under exposed and control are the blood lead content, in mg/dl.
```

Our primary goal will be to estimate the difference in lead content between the exposed and control children, and then use that sample estimate to make inferences about the difference in lead content between the population of all children like those in the exposed group and the population of all children like those in the control group.

22.1.1 Our Key Questions for a Paired Samples Comparison

1. What is the **population** under study?
 - All pairs of children living in Oklahoma near the factory in question, in which one had a parent working in a factory that exposed them to lead, and the other did not.
2. What is the **sample**? Is it representative of the population?

- The sample consists of 33 pairs of one exposed and one control child.
 - This is a case-control study, where the children were carefully enrolled to meet the design criteria. Absent any other information, we're likely to assume that there is no serious bias associated with these pairs, and that assuming they represent the population effectively (and perhaps the broader population of kids whose parents work in lead-based industries more generally) may well be at least as reasonable as assuming they don't.
3. Who are the subjects / **individuals** within the sample?
 - Each of our 33 pairs of children includes one exposed child and one unexposed (control) child.
 4. What **data** are available on each individual?
 - The blood lead content, as measured in mg/dl of whole blood.

22.1.2 Lead Study Caveats

Note that the children were not randomly selected from general populations of kids whose parents did and did not work in lead-based industries.

- To make inferences to those populations, we must make **strong assumptions** to believe, for instance, that the sample of exposed children is as representative as a random sample of children with similar exposures across the world would be.
- The researchers did have a detailed theory about how the exposed children might be at increased risk of lead absorption, and in fact as part of the study gathered additional information about whether a possible explanation might be related to the quality of hygiene of the parents (all of them were fathers, actually) who worked in the factory.
- This is an observational study, so that the estimation of a causal effect between parental work in a lead-based industry and children's blood lead content can be made, without substantial (and perhaps heroic) assumptions.

22.2 Exploratory Data Analysis for Paired Samples

We'll begin by adjusting the data in two ways.

- We'd like that first variable (**pair**) to be a **factor** rather than a **character** type in R, because we want to be able to summarize it more effectively. So we'll make that change.
- Also, we'd like to calculate the difference in lead content between the exposed and the control children in each pair, and we'll save that within-pair difference in a variable called **lead_diff**. We'll take **lead_diff** =

```

exposed - control so that positive values indicate increased lead in the
exposed child.

bloodlead_original <- bloodlead

bloodlead <- bloodlead_original %>%
  mutate(pair = factor(pair),
        lead_diff = exposed - control)

bloodlead

# A tibble: 33 x 4
  pair   exposed control lead_diff
  <fct>    <dbl>    <dbl>     <dbl>
1 P01      38       16      22
2 P02      23       18       5
3 P03      41       18      23
4 P04      18       24      -6
5 P05      37       19      18
6 P06      36       11      25
7 P07      23       10      13
8 P08      62       15      47
9 P09      31       16      15
10 P10     34       18      16
# ... with 23 more rows

```

22.2.1 The Paired Differences

To begin, we focus on `lead_diff` for our exploratory work, which is the `exposed - control` difference in lead content within each of the 33 pairs. So, we'll have 33 observations, as compared to the 462 in the serum zinc data, but most of the same tools are still helpful.

```

res_lead <- mosaic::favstats(~ lead_diff, data = bloodlead)
bin_w1 <- 5 # specify binwidth

p1 <- ggplot(bloodlead, aes(x = lead_diff)) +
  geom_histogram(binwidth = bin_w1,
                 fill = "firebrick",
                 col = "white") +
  theme_light() +
  stat_function(
    fun = function(x) dnorm(x, mean = res_lead$mean,
                           sd = res_lead$sd) *
    res_lead$n * bin_w1,
    col = "blue") +

```

```

labs(x = "Diff. in Lead Content (mg/dl)", y = "")

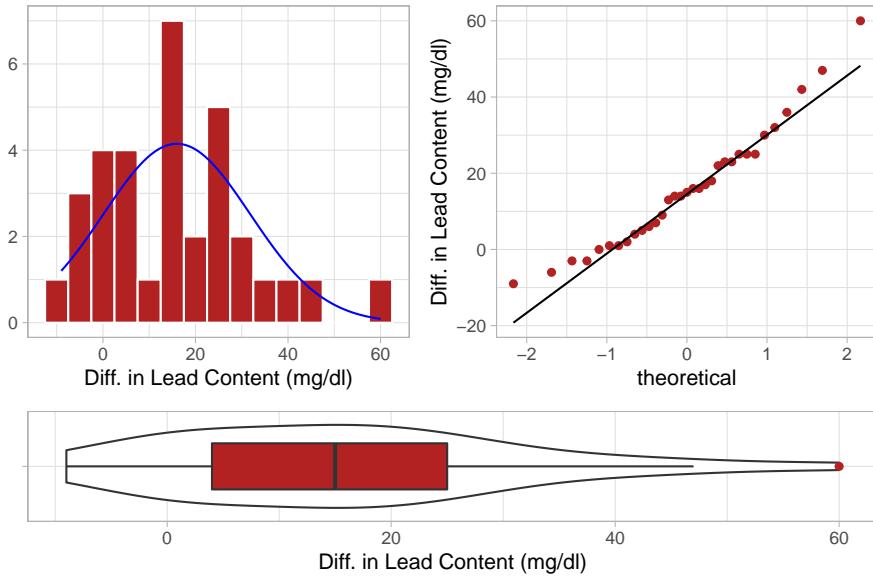
p2 <- ggplot(bloodlead, aes(sample = lead_diff)) +
  geom_qq(col = "firebrick") +
  geom_qq_line(col = "black") +
  theme_light() +
  labs(y = "Diff. in Lead Content (mg/dl)")

p3 <- ggplot(bloodlead, aes(x = "", y = lead_diff)) +
  geom_violin() +
  geom_boxplot(width = 0.5, fill = "firebrick",
               outlier.color = "firebrick") +
  theme_light() +
  coord_flip() +
  labs(x = "", y = "Diff. in Lead Content (mg/dl)")

p1 + p2 - p3 + plot_layout(ncol = 1, height = c(3, 1)) +
  plot_annotation(title = "Difference in Blood Lead Content (mg/dl) for 33 Pairs of Children")

```

Difference in Blood Lead Content (mg/dl) for 33 Pairs of Children



Note that in all of this work, I plotted the paired differences. One obvious way to tell if you have paired samples is that you can pair every single subject from one exposure group to a unique subject in the other exposure group. Everyone has to be paired, so the sample sizes will always be the same in the two groups.

22.2.2 Numerical Summaries

```
mosaic::favstats(~ lead_diff, data = bloodlead)
```

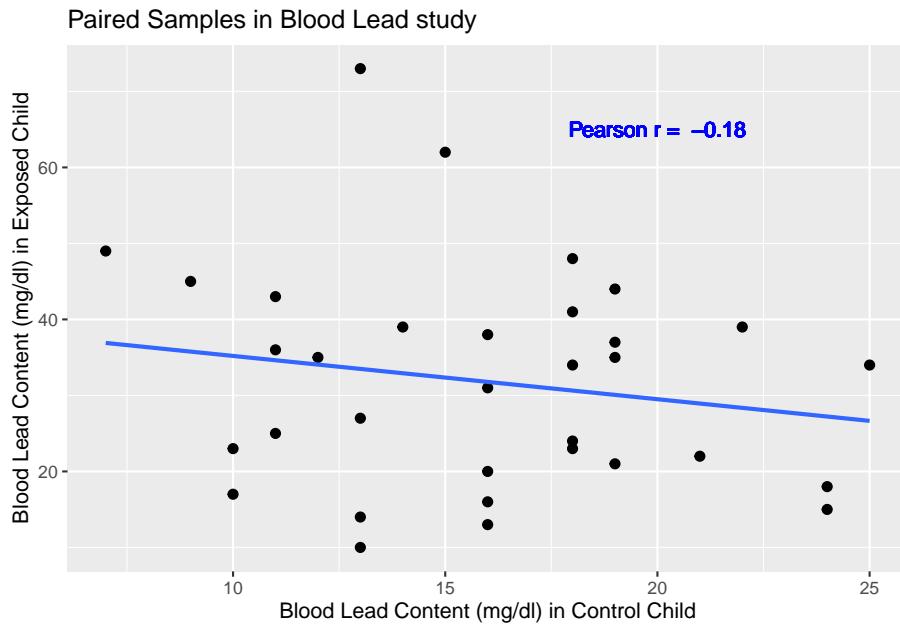
```
min Q1 median Q3 max mean   sd  n missing
-9    4     15  25   60   16 15.9 33      0
bloodlead %$% skew1(lead_diff)
```

```
[1] 0.0611
```

22.2.3 Impact of Matching - Scatterplot and Correlation

Here, the data are paired by the study through matching on neighborhood, age and exposure to traffic. Each individual child's outcome value is part of a pair with the outcome value for his/her matching partner. We can see this pairing in several ways, perhaps by drawing a scatterplot of the pairs.

```
ggplot(bloodlead, aes(x = control, y = exposed)) +
  geom_point(size = 2) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_text(x = 20, y = 65, col = "blue",
            label =
              paste("Pearson r = ",
                    round(bloodlead %$%
                           cor(control, exposed), 2))) +
  labs(title = "Paired Samples in Blood Lead study",
       x = "Blood Lead Content (mg/dl) in Control Child",
       y = "Blood Lead Content (mg/dl) in Exposed Child")
```



Each point here represents a **pair** of observations, one from a control child, and one from the matched exposed child. If there is a strong linear relationship (usually with a positive slope, thus positive correlation) between the paired outcomes, then the pairing will be more helpful in terms of improving statistical power of the estimates we build than if there is a weak relationship.

- The stronger the Pearson correlation coefficient, the more helpful pairing will be.
- Here, a straight line model using the control child's blood lead content accounts for about 3.2% of the variation in blood lead content in the exposed child.
- As it turns out, pairing will have only a modest impact here on the inferences we draw in the study. We still will treat the data as paired, despite this.

22.3 Looking at Separate Samples: Using `pivot_longer`

For the purpose of estimating the difference between the exposed and control children, the summaries of the paired differences are what we'll need.

In some settings, however, we might also look at a boxplot, or violin plot, or ridgeline plot that showed the distributions of exposed and control children separately. But we will run into trouble because one variable (blood lead content)

is spread across multiple columns (control and exposed.) The solution is to “pivot” the tibble from its current format to build a new, tidy tibble. Because the data aren’t *tidied* here, so that we have one row for each subject and one column for each variable, we have to do some work to get them in that form for our usual plotting strategy to work well.

- `pivot_longer()` “lengthens” the data, increasing the number of rows and decreasing the number of columns.
- `pivot_wider()` performs the inverse of that transformation, “widening” the data.

In our original `bloodlead` data, if we drop the `lead_diff` addition we made, we have *wide* data, with each row representing two different subjects.

```
head(bloodlead_original, 3)
```

```
# A tibble: 3 x 3
  pair   exposed control
  <chr>  <dbl>    <dbl>
1 P01      38      16
2 P02      23      18
3 P03      41      18
```

And what we want to accomplish is to have one row for each subject, instead of one row for each pair of subjects. So we want to make the data **longer**.

```
bloodlead_longer <- bloodlead_original %>%
  pivot_longer(
    cols = -c(pair),
    names_to = "status",
    values_to = "lead_level")
```

```
bloodlead_longer
```

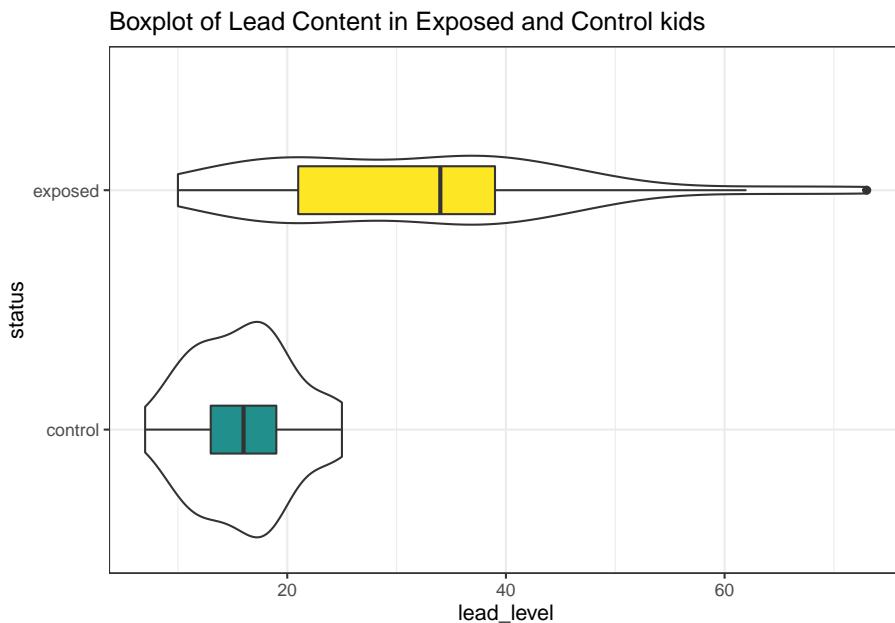
```
# A tibble: 66 x 3
  pair   status lead_level
  <chr>  <chr>     <dbl>
1 P01   exposed      38
2 P01   control      16
3 P02   exposed      23
4 P02   control      18
5 P03   exposed      41
6 P03   control      18
7 P04   exposed      18
8 P04   control      24
9 P05   exposed      37
10 P05  control      19
# ... with 56 more rows
```

For more on this approach (in this case, we’re making the data “longer” and its opposite would be be making the data “wider”), visit the Tidy data chapter in Grolemund and Wickham (2019) and the `tidyr` repository on Github at <https://github.com/tidyverse/tidyr>.

And now, we can plot as usual to compare the two samples.

First, we’ll look at a boxplot, showing all of the data.

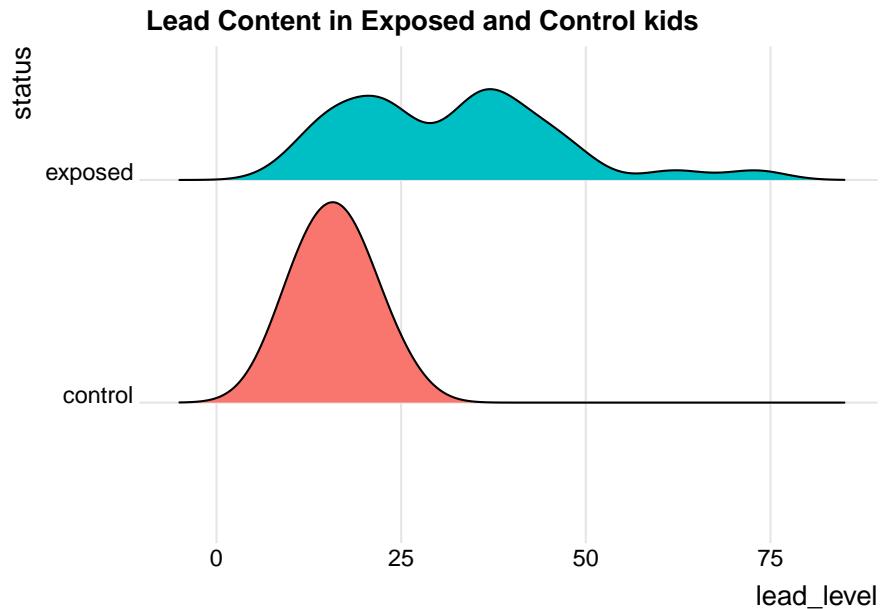
```
ggplot(bloodlead_longer, aes(x = status, y = lead_level)) +
  geom_violin() +
  geom_boxplot(aes(fill = status), width = 0.2) +
  scale_fill_viridis_d(begin = 0.5) +
  guides(fill = FALSE) +
  coord_flip() +
  labs(title = "Boxplot of Lead Content in Exposed and Control kids") +
  theme_bw()
```



We’ll also look at a ridgeline plot, because Dr. Love likes them, even though they’re really more useful when we’re comparing more than two samples.

```
ggplot(bloodlead_longer, aes(x = lead_level, y = status, fill = status)) +
  ggridges::geom_density_ridges(scale = 0.9) +
  guides(fill = FALSE) +
  labs(title = "Lead Content in Exposed and Control kids") +
  ggridges::theme_ridges()
```

Picking joint bandwidth of 4.01



Both the center and the spread of the distribution are substantially larger in the exposed group than in the controls. Of course, numerical summaries show these patterns, too.

```
mosaic::favstats(lead_level ~ status, data = bloodlead_longer)
```

	status	min	Q1	median	Q3	max	mean	sd	n	missing
1	control	7	13	16	19	25	15.9	4.54	33	0
2	exposed	10	21	34	39	73	31.8	14.41	33	0

22.4 Estimating the Difference in Means with Paired Samples

Suppose we want to estimate the difference in the mean blood level across the population of children represented by the sample taken in this study. To do so, we must take advantage of the matched samples design, and complete our estimation on the paired differences, treating them as if they were a single sample of data.

One way to accomplish this is simply to run the usual intercept-only linear regression model on the paired differences.

```
model_lead <- lm(lead_diff ~ 1, data = bloodlead)
tidy(model_lead, conf.int = TRUE, conf.level = 0.90)
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic   p.value conf.low conf.high
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept) 16.0      2.76      5.78 0.00000204    11.3     20.6
```

Our point estimate for the difference (exposed - control) in lead levels is 15.97 mg/dl, and our 90% confidence interval is (11.29, 20.65) mg/dl.

22.4.1 Paired Data in Longer Format?

If we had the data in “longer” format, as in `bloodlead_longer`, with the pairs identified by the `pair` variable, then we could obtain the same confidence interval using:

```
model2_lead <- lm(lead_level ~ status + factor(pair), data = bloodlead_longer)

tidy(model2_lead, conf.int = TRUE, conf.level = 0.90)

# A tibble: 34 x 7
  term      estimate std.error statistic   p.value conf.low conf.high
  <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept) 19.0      8.05      2.36  2.44e-2    5.38     32.7
2 statusexposed 16.0      2.76      5.78  2.04e-6    11.3     20.6
3 factor(pair)P~ -6.50     11.2     -0.579 5.66e-1   -25.5     12.5
4 factor(pair)P~  2.5       11.2      0.223 8.25e-1   -16.5     21.5
5 factor(pair)P~ -6.00     11.2     -0.535 5.96e-1   -25.0     13.0
6 factor(pair)P~  1.0       11.2      0.0891 9.30e-1   -18.0     20.0
7 factor(pair)P~ -3.50     11.2     -0.312 7.57e-1   -22.5     15.5
8 factor(pair)P~ -10.5     11.2     -0.936 3.56e-1   -29.5      8.50
9 factor(pair)P~  11.5     11.2      1.03  3.13e-1   -7.50     30.5
10 factor(pair)P~ -3.50     11.2     -0.312 7.57e-1   -22.5     15.5
# ... with 24 more rows
```

and the key elements are found in the `statusexposed` row, which we can focus on nicely (since the output of the `tidy()` function is always a tibble) with:

```
tidy(model2_lead, conf.int = TRUE, conf.level = 0.90) %>%
  filter(term == "statusexposed") %>%
  knitr::kable(digits = 2)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
statusexposed	16	2.76	5.78	0	11.3	20.6

and again, we have our 90% confidence interval estimate of the population mean difference between exposed and control children.

More approaches to making inferences about paired sample means as well as other summaries of quantitative outcomes will be found in Chapter 23.

22.5 Comparing Proportions in Paired Samples

We discuss some ideas relevant to the comparison of proportions in paired samples in Chapter 24, in the context of a different example.

Chapter 23

Comparing Means/Quantities using Two Paired Samples

Here, we'll consider the problem of estimating a confidence interval to describe the difference in population means (or medians) based on a comparison of two samples of quantitative data, gathered using a matched pairs design. Specifically, we'll use as our example the Lead in the Blood of Children study, described in Section 22.

Recall that in that study, we measured blood lead content, in mg/dl, for 33 matched pairs of children, one of which was exposed (had a parent working in a battery factory) and the other of which was control (no parent in the battery factory, but matched to the exposed child by age, exposure to traffic and neighborhood). We then created a variable called `lead_diff` which contained the (exposed - control) differences within each pair.

```
bloodlead
```

```
# A tibble: 33 x 4
  pair   exposed control lead_diff
  <fct>    <dbl>    <dbl>     <dbl>
1 P01      38       16      22
2 P02      23       18       5
3 P03      41       18      23
4 P04      18       24      -6
5 P05      37       19      18
6 P06      36       11      25
7 P07      23       10      13
8 P08      62       15      47
```

362CHAPTER 23. COMPARING MEANS/QUANTITIES USING TWO PAIRED SAMPLES

```
9 P09      31      16      15
10 P10     34      18      16
# ... with 23 more rows
```

23.0.1 Matched Pairs vs. Two Independent Samples

These data were NOT obtained from two independent samples, but rather from matched pairs.

- We only have matched pairs if each individual observation in the “treatment” group is matched to one and only one observation in the “control” group by the way in which the data were gathered. Paired (or matched) data can arise in several ways.
 - The most common is a “pre-post” study where subjects are measured both before and after an exposure happens.
 - In observational studies, we often match up subjects who did and did not receive an exposure so as to account for differences on things like age, sex, race and other covariates. This, of course, is what happens in the Lead in the Blood of Children study from Chapter 22.
- If the data are from paired samples, we should (and in fact) must form paired differences, with no subject left unpaired.
 - If we cannot line up the data comparing two samples of quantitative data so that the links between the individual “treated” and “control” observations to form matched pairs are evident, then the data are not paired.
 - If the sample sizes were different, we’d know we have independent samples, because matched pairs requires that each subject in the “treated” group be matched to a single, unique member of the “control” group, and thus that we have exactly as many “treated” as “control” subjects.
 - But having as many subjects in one treatment group as the other (which is called a *balanced design*) is only necessary, and not sufficient, for us to conclude that matched pairs are used.

As Bock, Velleman, and De Veaux (2004) suggest,

... if you know the data are paired, you can take advantage of that fact - in fact, you *must* take advantage of it. ... You must decide whether the data are paired from understanding how they were collected and what they mean. ... There is no test to determine whether the data are paired.

23.1 Estimating the Population Mean of the Paired Differences

There are two main approaches used frequently to estimate the population mean of paired differences.

- Estimation using the t distribution (and assuming at least an approximately Normal distribution for the paired differences)
- Estimation using the bootstrap (which doesn't require the Normal assumption)

In addition, we might consider estimating an alternate statistic when the data don't follow a symmetric distribution, like the median, with the bootstrap. In other settings, a rank-based alternative called the Wilcoxon signed rank test is available to estimate a psuedo-median. All of these approaches mirror what we did with a single sample, back in Chapter 17.

23.2 t-based CI for Population Mean of Paired Differences, μ_d .

In R, there are at least five different methods for obtaining the t-based confidence interval for the population difference in means between paired samples. They are all mathematically identical. The key idea is to calculate the paired differences (exposed - control, for example) in each pair, and then treat the result as if it were a single sample and apply the methods discussed in Chapter 17.

23.2.1 Method 1

We can use the single-sample approach, applied to the variable containing the paired differences. Let's build a 90% two-sided confidence interval for the population mean of the difference in blood lead content across all possible pairs of an exposed (parent works in a lead-based industry) and a control (parent does not) child, μ_d .

```
tt1 <- bloodlead %$% t.test(lead_diff, conf.level = 0.90,
                             alt = "two.sided")
```

```
tt1
```

```
One Sample t-test
```

```
data: lead_diff
t = 6, df = 32, p-value = 2e-06
```

```

alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 11.3 20.6
sample estimates:
mean of x
 16

```

```
tidy(tt1) %>% knitr::kable(digits = 2)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
16	5.78	0	32	11.3	20.6	One Sample t-test	two.sided

The 90% confidence interval is (11.29, 20.65) according to this t-based procedure. An appropriate interpretation of the 90% two-sided confidence interval would be:

- (11.29, 20.65) milligrams per deciliter is a 90% two-sided confidence interval for the population mean difference in blood lead content between exposed and control children.
- Our point estimate for the true population difference in mean blood lead content is 15.97 mg.dl. The values in the interval (11.29, 20.65) mg/dl represent a reasonable range of estimates for the true population difference in mean blood lead content, and we are 90% confident that this method of creating a confidence interval will produce a result containing the true population mean difference.
- Were we to draw 100 samples of 33 matched pairs from the population described by this sample, and use each such sample to produce a confidence interval in this manner, approximately 90 of those confidence intervals would cover the true population mean difference in blood lead content levels.

23.2.2 Method 2

Or, we can apply the single-sample approach to a calculated difference in blood lead content between the exposed and control groups. Here, we'll get a **95%** two-sided confidence interval for μ_d , instead of the 90% interval we obtained above.

```

tt2 <- bloodlead %$% t.test(exposed - control,
                           conf.level = 0.95, alt = "two.sided")

```

```
tt2
```

```
One Sample t-test
```

```

data: exposed - control
t = 6, df = 32, p-value = 2e-06

```

23.2. T-BASED CI FOR POPULATION MEAN OF PAIRED DIFFERENCES, μ_D .365

```
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 10.3 21.6
sample estimates:
mean of x
 16
tidy(tt2) %>% knitr::kable(digits = 2)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
16	5.78	0	32	10.3	21.6	One Sample t-test	two.sided

23.2.3 Method 3

Or, we can provide R with two separate samples (unaffected and affected) and specify that the samples are paired. Here, we'll get a **99% one-sided** confidence interval (lower bound) for μ_d , the population mean difference in blood lead content.

```
tt3 <- bloodlead %$% t.test(exposed, control, conf.level = 0.99,
                           paired = TRUE, alt = "greater")
tt3
```

```
Paired t-test

data: exposed and control
t = 6, df = 32, p-value = 1e-06
alternative hypothesis: true difference in means is greater than 0
99 percent confidence interval:
 9.21 Inf
sample estimates:
mean of the differences
 16
tidy(tt3) %>% knitr::kable(digits = 2)
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
16	5.78	0	32	9.21	Inf	Paired t-test	greater

Again, the three different methods using `t.test` for paired samples will all produce identical results if we feed them the same confidence level and type of interval (two-sided, greater than or less than).

23.2.4 Method 4

As we saw in Chapter 22, we can also use an intercept-only linear regression model to estimate the population mean of the paired differences with a two-tailed confidence interval, by creating a variable containing those paired differences.

```
model_lead <- lm(lead_diff ~ 1, data = bloodlead)

tidy(model_lead, conf.int = TRUE, conf.level = 0.95)

# A tibble: 1 x 7
  term      estimate std.error statistic   p.value conf.low conf.high
  <chr>        <dbl>     <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
1 (Intercept)    16.0      2.76      5.78 0.00000204    10.3     21.6
```

23.2.5 Method 5

As we also saw in Chapter 22, if we have the data in a longer format, with a variable identifying the matched pairs, we can use a different specification for a linear model to obtain the same estimate.

```
model2_lead <- lm(lead_level ~ status + factor(pair), data = bloodlead_longer)

tidy(model2_lead, conf.int = TRUE, conf.level = 0.95) %>%
  filter(term == "statusexposed")

# A tibble: 1 x 7
  term      estimate std.error statistic   p.value conf.low conf.high
  <chr>        <dbl>     <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
1 statusexposed    16.0      2.76      5.78 0.00000204    10.3     21.6
```

23.2.6 Assumptions

If we are building a confidence interval based on a sample of observations drawn from a population, then we must pay close attention to the assumptions of those procedures. The confidence interval procedure for the population mean paired difference μ_d using the t distribution assumes that:

1. We want to estimate the population mean paired difference μ_d .
2. We have drawn a sample of paired differences at random from the population of interest.
3. The sampled paired differences are drawn from the population set of paired differences independently and have identical distributions.
4. The population follows a Normal distribution. At the very least, the sample itself is approximately Normal.

23.3 Bootstrap CI for mean difference using paired samples

The same bootstrap approach is used for paired differences as for a single sample. We again use the `smean.cl.boot()` function in the `Hmisc` package to obtain bootstrap confidence intervals for the population mean, μ_d , of the paired differences in blood lead content.

```
set.seed(431555)
bloodlead %$% Hmisc::smean.cl.boot(lead_diff, B = 1000,
                                     conf.int = 0.95)
```

Mean	Lower	Upper
16.0	10.8	21.5

Note that in this case, the confidence interval for the difference in means is a bit less wide than the 95% confidence interval generated by the t test, which was (10.34, 21.59). It's common for the bootstrap to produce a narrower range (i.e. an apparently more precise estimate) for the population mean, but it's not automatic that the endpoints from the bootstrap will be inside those provided by the t test, either.

For example, this bootstrap CI doesn't contain the t-test based interval, since its upper bound exceeds that of the t-based interval:

```
set.seed(431002)
bloodlead %$% Hmisc::smean.cl.boot(lead_diff, B = 1000,
                                     conf.int = 0.95)
```

Mean	Lower	Upper
16.0	10.8	21.7

This demonstration aside, the appropriate thing to do when applying the bootstrap to specify a confidence interval is select a seed and the number ($B = 1,000$ or $10,000$, usually) of desired bootstrap replications, then run the bootstrap just once and move on, rather than repeating the process multiple times looking for a particular result.

23.3.1 Assumptions

The bootstrap confidence interval procedure for the population mean (or median) of a set of paired differences assumes that:

1. We want to estimate the population mean μ_d of the paired differences (or the population median).
2. We have drawn a sample of observations at random from the population of interest.

3. The sampled observations are drawn from the population of paired differences independently and have identical distributions.
4. We are willing to put up with the fact that different people (not using the same random seed) will get somewhat different confidence interval estimates using the same data.

As we've seen, a major part of the bootstrap's appeal is the ability to relax some assumptions.

23.4 Wilcoxon Signed Rank-based CI for paired samples

We could also use the Wilcoxon signed rank procedure to generate a CI for the pseudo-median of the paired differences.

```
wt <- bloodlead %$% wilcox.test(lead_diff, conf.int = TRUE,
                                    conf.level = 0.90,
                                    exact = FALSE)
wt

Wilcoxon signed rank test with continuity correction

data: lead_diff
V = 499, p-value = 1e-05
alternative hypothesis: true location is not equal to 0
90 percent confidence interval:
 11.0 20.5
sample estimates:
(pseudo)median
 15.5

tidy(wt)

# A tibble: 1 x 7
  estimate statistic p.value conf.low conf.high method      alternative
    <dbl>     <dbl>    <dbl>     <dbl>     <dbl> <chr>      <chr>
1     15.5      499  1.15e-5      11.0      20.5 Wilcoxon sign~ two.sided
```

As in the one sample case, we can revise this code slightly to specify a different confidence level, or gather a one-sided rather than a two-sided confidence interval.

23.4.1 Assumptions

The Wilcoxon signed rank confidence interval procedure in working with paired differences assumes that:

1. We want to estimate the population **pseudo-median** of the paired differences.
2. We have drawn a sample of observations at random from the population of paired differences of interest.
3. The sampled observations are drawn from the population of paired differences independently and have identical distributions.
4. The population follows a symmetric distribution. At the very least, the sample itself shows no substantial skew, so that the sample pseudo-median is a reasonable estimate for the population median.

23.5 Choosing a Confidence Interval Approach

Suppose we want to find a confidence interval for the mean of a population, μ , or, the population mean difference μ_d between two populations based on matched pairs.

1. If we are willing to assume that the population distribution is **Normal**
 - we usually use a t-based CI.
2. If we are **unwilling** to assume that the population is Normal,
 - use a **bootstrap** procedure to get a CI for the population mean, or even the median
 - but are willing to assume the population is symmetric, consider a **Wilcoxon signed rank** procedure to get a CI for the median, rather than the mean.

The two methods you'll use most often are the bootstrap (especially if the data don't appear to be at least pretty well fit by a Normal model) and the t-based confidence intervals (if the data do appear to fit a Normal model reasonably well.)

23.6 Conclusions for the **bloodlead** study

Using any of these procedures, we would conclude that the null hypothesis (that the true mean of the paired differences is 0 mg/dl) is not tenable, and that it should be rejected at the 10% significance level. The smaller the p value, the stronger is the evidence that the null hypothesis is incorrect, and in this case, we have some fairly tiny p values.

Procedure	Comparing ...	90% CI for $\mu_{Exposed-Control}$
Paired t	Means	11.3, 20.6
Wilcoxon signed rank	Pseudo-medians	11, 20.5
Bootstrap CI	Means	11.6, 20.6

Note that **one-sided** or **one-tailed** hypothesis testing procedures work the same way for paired samples as they did for a single sample in Chapter 17.

23.7 The Sign test

The **sign test** is something we've skipped in our discussion so far. It is a test for consistent differences between pairs of observations, just as the paired t, Wilcoxon signed rank and bootstrap for paired samples can provide. It has the advantage that it is relatively easy to calculate by hand, and that it doesn't require the paired differences to follow a Normal distribution. In fact, it will even work if the data are substantially skewed.

- Calculate the paired difference for each pair, and drop those with difference = 0.
- Let N be the number of pairs that remain, so there are $2N$ data points.
- Let W , the test statistic, be the number of pairs (out of N) in which the difference is positive.
- Assuming that H_0 is true, then W follows a binomial distribution with probability 0.5 on N trials.

For example, consider our data on blood lead content:

```
bloodlead$lead_diff
```

```
[1] 22 5 23 -6 18 25 13 47 15 16 6 1 2 7 0 4 -9 -3 36 25 1 16 42
[24] 30 25 23 32 17 9 -3 60 14 14
```

Difference	# of Pairs
Greater than zero	28
Equal to zero	1
Less than zero	4

So we have $N = 32$ pairs, with $W = 28$ that are positive. We then use the **binom.test** approach in R:

```
binom.test(x = 28, n = 32, p = 0.5,
            alternative = "two.sided")
```

23.8. A MORE COMPLETE DECISION SUPPORT TOOL: COMPARING MEANS371

Exact binomial test

```
data: 28 and 32
number of successes = 28, number of trials = 32, p-value = 2e-05
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.710 0.965
sample estimates:
probability of success
 0.875
```

- A one-tailed test can be obtained by substituting in “less” or “greater” as the alternative of interest.
- The confidence interval provided doesn’t relate back to our original population means, of course. It’s just showing the confidence interval around the probability of the exposed mean being greater than the control mean for a pair of children.

23.8 A More Complete Decision Support Tool: Comparing Means

1. Are these paired or independent samples?
2. If paired samples, then are the paired differences approximately Normally distributed?
 - a. If yes, then a paired t test or confidence interval is likely the best choice.
 - b. If no, is the main concern outliers (with generally symmetric data), or skew?
 1. If the paired differences appear to be generally symmetric but with substantial outliers, a Wilcoxon signed rank test is an appropriate choice, as is a bootstrap confidence interval for the population mean of the paired differences.
 2. If the paired differences appear to be seriously skewed, then we’ll usually build a bootstrap confidence interval, although a sign test is another reasonable possibility, although it doesn’t provide a confidence interval for the population mean of the paired differences.
3. If independent, is each sample Normally distributed?
 - a. No → use Wilcoxon-Mann-Whitney rank sum test or bootstrap via `bootdif`.
 - b. Yes → are sample sizes equal?
 1. Balanced Design (equal sample sizes) - use pooled t test

2. Unbalanced Design - use Welch test

23.9 Paired (Dependent) vs. Independent Samples

One area that consistently trips students up in this course is the thought process involved in distinguishing studies comparing means that should be analyzed using *dependent* (i.e. paired or matched) samples and those which should be analyzed using *independent* samples. A dependent samples analysis uses additional information about the sample to pair/match subjects receiving the various exposures. That additional information is not part of an independent samples analysis (unpaired testing situation.) The reasons to do this are to (a) increase statistical power, and/or (b) reduce the effect of confounding. Here are a few thoughts on the subject.

In the design of experiments, **blocking** is the term often used for the process of arranging subjects into groups (blocks) that are similar to one another. Typically, a blocking factor is a source of variability that is not of primary interest to the researcher. An example of a blocking factor might be the sex of a patient; by blocking on sex, this source of variability is controlled for, thus leading to greater accuracy.

1. If the sample sizes are not balanced (not equal), the samples must be treated as independent, since there would be no way to precisely link all subjects. So, if we have 10 subjects receiving exposure A and 12 subjects receiving exposure B, a dependent samples analysis (such as a paired *t* test) is not correct.
2. The key element is a meaningful link between each observation in one exposure group and a specific observation in the other exposure group. Given a balanced design, the most common strategy indicating dependent samples involves two or more *repeated measures* on the same subjects. For example, if we are comparing outcomes *before* and *after* the application of an exposure, and we have, say, 20 subjects who provide us data both *before* and *after* the exposure, then the comparison of results *before* and *after* exposure should use a dependent samples analysis. The link between the subjects is the subject itself - each exposed subject serves as its own control.
3. The second most common strategy indicating dependent samples involves deliberate matching of subjects receiving the two exposures. A matched set of observations (often a pair, but it could be a trio or quartet, etc.) is determined using baseline information and then (if a pair is involved) one subject receives exposure A while the other member of the pair receives exposure B, so that by calculating the paired difference, we learn about

the effect of the exposure, while controlling for the variables made similar across the two subjects by the matching process.

4. In order for a dependent samples analysis to be used, we need (a) a link between each observation across the exposure groups based on the way the data were collected, *and* (b) a consistent measure (with the same units of measurement) so that paired differences can be calculated and interpreted sensibly.
5. If the samples are collected to facilitate a dependent samples analysis, the correlation of the outcome measurements across the groups will often be moderately strong and positive. If that's the case, then the use of a dependent samples analysis will reduce the effect of baseline differences between the exposure groups, and thus provide a more precise estimate. But even if the correlation is quite small, a dependent samples analysis should provide a more powerful estimate of the impact of the exposure on the outcome than would an independent samples analysis with the same number of observations.

23.9.1 Three “Tricky” Examples

1. Suppose we take a convenient sample of 200 patients from the population of patients who complete a blood test in April 2017 including a check of triglycerides, and who have a triglyceride level in the high category (200 to 499 mg/dl). Next, we select a patient at random from this group of 200 patients, and then identify another patient from the group of 200 who is the same age (to within 2 years) and also the same sex. We then randomly assign our intervention to one of these two patients and usual care without our intervention to the other patient. We then set these two patients aside and return to our original sample, repeating the process until we cannot find any more patients in the same age range and of the same gender. This generates a total of 77 patients who receive the intervention and 77 who do not. If we are trying to assess the effect of our intervention on triglyceride level in October 2017 using this sample of 154 people, should we use dependent (paired) or independent samples?
2. Suppose we take a convenient sample of 77 patients from the population of patients who complete a blood test in April 2017 including a check of triglycerides, and who have a triglyceride level in the high category (200 to 499 mg/dl). Next, we take a convenient sample of 77 patients from the population of patients who complete a blood test in May 2017 including a check of triglycerides, and who have a triglyceride level in the high category (200 to 499 mg/dl). We flip a coin to determine whether the intervention will be given to each of the 77 patients from April 2017 (if the coin comes up “HEADS”) or instead to each of the 77 patients from May 2017 (if the coin comes up “TAILS”). Then, we assign our intervention to

the patients seen in the month specified by the coin and assign usual care without our intervention to the patients seen in the other month. If we are trying to assess the effect of our intervention on triglyceride level in October 2017 using this sample of 154 people, should we use dependent (paired) or independent samples?

3. Suppose we take a convenient sample of 200 patients from the population of patients who complete a blood test in April 2017 including a check of triglycerides, and who have a triglyceride level in the high category (200 to 499 mg/dl). For each patient, we re-measure them again in October 2017, again checking their triglyceride level. But in between, we take the first 77 of the patients in a randomly sorted list and assign them to our intervention (which takes place from June through September 2017) and take an additional group of 77 patients from the remaining part of the list and assign them to usual care without our intervention over the same time period. If we are trying to assess the effect of our intervention on each individual's change in triglyceride level (from April/May to October) using this sample of 154 people, should we use dependent (paired) or independent samples?

23.9.2 Answers for the Three “Tricky” Examples

Answer for 1. Our first task is to identify the outcome and the exposure groups. Here, we are comparing the distribution of our outcome (triglyceride level in October) across two exposures: (a) receiving the intervention and (b) not receiving the intervention. We have a sample of 77 patients receiving the intervention, and a different sample of 77 patients receiving usual care. Each of the 77 subjects receiving the intervention is matched (on age and sex) to a specific subject not receiving the intervention. So, we can calculate paired differences by taking the triglyceride level for the exposed member of each pair and subtracting the triglyceride level for the usual care member of that same pair. Thus our comparison of the exposure groups should be accomplished using a *dependent* samples analysis, such as a paired t test.

Answer for 2. Again, we begin by identifying the outcome (triglyceride level in October) and the exposure groups. Here, we compare two exposures: (a) receiving the intervention and (b) receiving usual care. We have a sample of 77 patients receiving the intervention, and a different sample of 77 patients receiving usual care. But there is no pairing or matching involved. There is no connection implied by the way that the data were collected that implies that, for example, patient 1 in the intervention group is linked to any particular subject in the usual care group. So we need to analyze the data using independent samples.

Answer for 3. Once again, we identify the outcome (now it is the within-subject change in triglyceride level from April to October) and the exposure groups. Here again, we compare two exposures: (a) receiving the intervention and (b)

receiving usual care. We have a sample of 77 patients receiving the intervention, and a different sample of 77 patients receiving usual care. But again, there is no pairing or matching between the patients receiving the intervention and the patients receiving usual care. While each outcome value is a difference (or change) in triglyceride levels, there's no connection implied by the way that the data were collected that implies that, for example, patient 1 in the intervention group is linked to any particular subject in the usual care group. So, again, we need to analyze the data using independent samples.

For more background and fundamental material, you might consider the Wikipedia pages on Paired Difference Test and on Blocking (statistics).

Chapter 24

Comparing Proportions/Rates using Two Paired Samples

This Chapter compares two population proportions (rates/percentages) when working with a paired samples design. Some of the motivation for this material (and a couple of key functions) come from Bilder and Loughlin (2015).

24.1 An Example: Access to Specialty Care and Insurance in Children

Bisgaier and Rhodes (2011) measured children's access to outpatient specialty care to identify potential disparities in providers' acceptance of Medicaid and the Children's Health Insurance Program (CHIP) - a public program versus private insurance. In their study, the research team, posing as mothers of pediatric patients, called 273 specialty clinics in Cook County, Illinois twice each, separated by one month. Following a standardized script, the two calls differed only by the insurance status of the child. A key point of interest is to identify whether there were meaningful disparities in provider acceptance of Medicaid-CHIP (public) versus private insurance overall, and within several specialties.

To start, we'll focus on the data describing the 82 calls made to 41 psychiatry clinics, where each clinic was called once representing public (Medicaid-CHIP) insurance, and once representing private insurance. Out of the 41 clinics,

- 18 denied an appointment to both the Medicaid-CHIP (Public Insurance) and Private insurance child

- 5 accepted both insurance types and granted appointments
- 16 denied public insurance but granted an appointment to the privately insured child
- 2 accepted public insurance but denied private insurance

We can summarize this in a table, where each count describes a *pair* of telephone calls to a single clinic.

	– Private Accepted	Private Denied	Total
Public Accepted	5	2	7
Public Denied	16	18	34
Total	21	20	41

24.1.1 Working with the `specialty` data

I've gathered the complete data (for all 273 clinics reported in Bisgaier and Rhodes (2011)) into a table called `specialty`.

```
specialty
```

```
# A tibble: 546 x 4
  clinic_id clinic_type insurance appointment
  <chr>      <chr>       <chr>      <chr>
1 C-001      Psychiatry   Private    Accepted
2 C-001      Psychiatry   Public     Accepted
3 C-002      Asthma       Private    Accepted
4 C-002      Asthma       Public     Denied
5 C-003      Orthopedics  Private    Accepted
6 C-003      Orthopedics  Public     Accepted
7 C-004      Orthopedics  Private    Accepted
8 C-004      Orthopedics  Public     Accepted
9 C-005      Orthopedics  Private    Accepted
10 C-005     Orthopedics  Public     Denied
# ... with 536 more rows
```

We're interested here in the data among the Psychiatry clinics, so we can filter and tabulate:

```
specialty %>% filter(clinic_type == "Psychiatry") %>%
  tabyl(insurance, appointment) %>%
  adorn_totals(where = c("row", "col"))
```

insurance	Accepted	Denied	Total
Private	21	20	41
Public	7	34	41
Total	28	54	82

but this represents the data as a single count for each of the 82 calls that was made to a Psychiatry clinic, without accounting for the pairing by clinic_id (since each clinic provides a result for a Private and a Public call.) A proper table that accounts for the pairing should wind up with 41 clinics being represented, not the 82 individual calls.

24.1.2 Using `pivot_wider` to get a paired samples table

So let's go back to the full `specialty` data, with 273 clinics and 546 calls. We want to "widen" the tibble, creating more columns but half as many rows, so that our new tibble has 273 rows (one per clinic) as opposed to 546 rows (one per call). To accomplish this, we'll use `pivot_wider`, creating more columns but fewer rows.

```
specialty_wider <- specialty %>%
  pivot_wider(names_from = insurance,
             values_from = appointment)

specialty_wider

# # A tibble: 273 x 4
#   clinic_id clinic_type Private  Public
#   <chr>      <chr>     <chr>    <chr>
# 1 C-001      Psychiatry Accepted Accepted
# 2 C-002      Asthma     Accepted Denied
# 3 C-003      Orthopedics Accepted Accepted
# 4 C-004      Orthopedics Accepted Accepted
# 5 C-005      Orthopedics Accepted Denied
# 6 C-006      Orthopedics Accepted Denied
# 7 C-007      Orthopedics Accepted Denied
# 8 C-008      Neurology   Accepted Accepted
# 9 C-009      Psychiatry  Denied   Denied
#10 C-010     Orthopedics Accepted Denied
# ... with 263 more rows
```

Now we have one row per clinic, and so now we can try again on the table for our 41 Psychiatry clinics.

```
specialty_wider %>% filter(clinic_type == "Psychiatry") %>%
  tabyl(Public, Private) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_title()

  Private
  Public Accepted Denied Total
Accepted      5      2      7
Denied       16     18     34
```

Total	21	20	41
-------	----	----	----

There we go.

24.2 What comparison are we making?

To describe the size of the effect in this situation, we have at least two options:

24.2.1 Concordant and Discordant Pairs in a Paired Samples 2x2 Table

If we have a paired-samples 2x2 table with counts a , b , c and d arranged as follows:

a	b
c	d

then a and d describe *concordant pairs*, where the same result is observed in both the rows and the columns, and b and c together describe the *discordant pairs*, where the opposite result is observed in the rows and columns.

In our insurance example, we have $a = 5$, $b = 2$, $c = 16$ and $d = 18$.

- So we have 23 ($a + d = 5 + 18$) *concordant clinics* where the same response (either Accept or Deny) was provided to both Public and Private.
- The remaining 18 ($b + c = 2 + 16$) are *discordant clinics* where the response to the Public call was different than the response to the Private call.

It turns out that the *discordant pairs*, generally, will be of maximum interest to us, as they give us an indication of the relative appeal of Public vs. Private insurance to these clinics, while the *concordant* results don't allow us to make any meaningful progress in building our comparison.

24.2.2 Measuring the McNemar Odds Ratio in a Paired Samples 2x2 Table

In a paired samples table we calculate something called the *McNemar odds ratio* using only the discordant pairs. The McNemar odds ratio is the larger of (b/c) or (c/b) .

Remember our insurance data, with $a = 5$, $b = 2$, $c = 16$ and $d = 18$:

	– Private Accepted	Private Denied	Total
Public Accepted	5	2	7

	– Private Accepted	Private Denied	Total
Public Denied	16	18	34
Total	21	20	41

Our odds ratio is $16/2$ or 8, since that is larger than its inverse. Of the Psychiatry clinics who accepted one of the two types of insurance, the odds ratio suggests a much greater likelihood of accepting an appointment for a child with private insurance than one with public (Medicaid-CHIP) insurance.

24.2.3 Measuring Cohen's g in a Paired Samples 2x2 Table

Cohen's g statistic (see Cohen (1988)) is also measured using only the counts of discordant pairs. First, we identify the larger of $\frac{b}{b+c}$ or $\frac{c}{b+c}$. Then Cohen's g is that value minus 0.5.

For our insurance data, we have:

$$\frac{b}{b+c} = \frac{2}{18} = 0.111$$

and

$$\frac{c}{b+c} = \frac{16}{18} = 0.889$$

which is the larger of the two. We subtract 0.5 from this result to obtain Cohen's $g = 0.389$. It turns out that Cohen's g is just a straightforward function of the McNemar odds ratio, and we'll concentrate our work on the odds ratio.

24.3 Building a Confidence Interval for the McNemar Odds Ratio

As a reminder, our table is:

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %>%
  tabyl(Public, Private)
```

	Public	Accepted	Denied
Accepted	5	2	
Denied	16	18	

```
library(exact2x2)

specialty_wider %>%
  filter(clinic_type == "Psychiatry") %$%
  exact2x2(Public, Private, paired = TRUE,
            conf.int = TRUE, conf.level = 0.90)
```

Exact McNemar test (with central confidence intervals)

```
data: Public and Private
b = 2, c = 16, p-value = 0.001
alternative hypothesis: true odds ratio is not equal to 1
90 percent confidence interval:
0.0205 0.4498
sample estimates:
odds ratio
0.125
```

This gives us the point estimate and 90% confidence interval for a McNemar odds ratio. The odds of being accepted for an appointment are lower for children with Public insurance than Private. We might prefer to develop an odds ratio point estimate that exceeds 1. To do so, we can simply invert the rows and columns.

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %$%
  exact2x2(Private, Public, paired = TRUE,
            conf.int = TRUE, conf.level = 0.90)
```

Exact McNemar test (with central confidence intervals)

```
data: Private and Public
b = 16, c = 2, p-value = 0.001
alternative hypothesis: true odds ratio is not equal to 1
90 percent confidence interval:
2.22 48.72
sample estimates:
odds ratio
8
```

This is equivalent to simply taking the reciprocal of the point and interval estimates we saw previously.

Bisgaier and Rhodes (2011) published (in their Table 2) a 95% confidence interval for this odds ratio, as (1.9 - 71.7). Does our result match theirs?

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %$%
  exact2x2(Private, Public, paired = TRUE,
            conf.int = TRUE, conf.level = 0.95)
```

```
Exact McNemar test (with central confidence intervals)

data: Private and Public
b = 16, c = 2, p-value = 0.001
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.88 71.72
sample estimates:
odds ratio
 8
```

24.3.1 A One-Sided Confidence Interval for the McNemar Odds Ratio

Suppose we simply wanted an upper bound, perhaps now with a 95% confidence level, for our McNemar odds ratio. We just need to specify an appropriate alternative in the exact2x2 function.

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %$%
  exact2x2(Private, Public, paired = TRUE,
            conf.int = TRUE, conf.level = 0.95,
            alternative = "less")
```

```
Exact McNemar-type test

data: Private and Public
b = 16, c = 2, p-value = 1
alternative hypothesis: true odds ratio is less than 1
95 percent confidence interval:
 0.0 48.7
sample estimates:
odds ratio
 8
```

And if we wanted a lower bound, that is also available.

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %$%
```

```
exact2x2(Private, Public, paired = TRUE,
          conf.int = TRUE, conf.level = 0.95,
          alternative = "greater")
```

Exact McNemar-type test

```
data: Private and Public
b = 16, c = 2, p-value = 7e-04
alternative hypothesis: true odds ratio is greater than 1
95 percent confidence interval:
 2.22 Inf
sample estimates:
odds ratio
      8
```

As you'll note, the lower and upper bounds of one-sided 95% confidence intervals mirror the two-sided 90% confidence interval, as usual.

24.3.2 What if we looked at all of the clinic types?

Here's the table for our complete set of 273 clinics.

```
specialty_wider %>%
  tabyl(Public, Private)
```

	Public	Accepted	Denied
Accepted	89	5	
Denied	155	24	

The McNemar odds ratio should be $155/5$ or 31. Cohen's g is $(155/(155 + 5)) - 0.5 = 0.96875 - 0.5 = 0.46875$. These are enormous effect sizes. For example, Cohen (1988) tentatively identified large effects as those with odds ratios exceeding 3 or Cohen's g exceeding 0.25. (Small effects were those with Cohen's g below 0.15, or Odds Ratios below 1.86).

What's a 90% confidence interval for the McNemar odds ratio across all 273 clinics?

```
specialty_wider %$%
  exact2x2(Private, Public, paired = TRUE,
            conf.int = TRUE, conf.level = 0.9)
```

Exact McNemar test (with central confidence intervals)

```
data: Private and Public
```

```
b = 155, c = 5, p-value <2e-16
alternative hypothesis: true odds ratio is not equal to 1
90 percent confidence interval:
 14.5 79.7
sample estimates:
odds ratio
 31
```

Bisgaier and Rhodes (2011) published a 95% confidence interval for this odds ratio in Table 2, as (13.0, 96.8).

```
specialty_wider %$%
  exact2x2(Private, Public, paired = TRUE,
            conf.int = TRUE, conf.level = 0.95)
```

Exact McNemar test (with central confidence intervals)

```
data: Private and Public
b = 155, c = 5, p-value <2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 13.0 96.8
sample estimates:
odds ratio
 31
```

Again, our result matches theirs.

24.4 Estimating the difference in proportions

Within the Psychiatry clinics, we saw the following:

```
specialty_wider %>%
  filter(clinic_type == "Psychiatry") %>%
  tabyl(Public, Private) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_title()
```

	Private		Total
	Accepted	Denied	
Accepted	5	2	7
Denied	16	18	34
Total	21	20	41

- In the Private calls, 21 of 41 (51.2%) were Accepted.
- In the Public calls, 7 of 41 (17.1%) were Accepted.

Can we compare these percentages (or perhaps their equivalent proportions) directly, and estimate that difference with a confidence interval?

Yes, thanks to the `PropCIs` package, and its `diffpropci.Wald.mp()` and `diffpropci.mp()` functions. To use these tools, we load the `PropCIs` package, and specify the values of `b`, `c` and `n`, the total sample size.

We will specify the values of `b = 2` and `c = 16` here so that the difference in conditional probability that we are comparing is the probability, assuming the clinic accepted the patient only once, of accepting private insurance minus the probability of accepting public insurance. We are estimating:

$$Pr(\text{Accepted for Private} | \text{only Accepted once}) - Pr(\text{Accepted for Public} | \text{only Accepted once})$$

based on a sample of `n = 41` Psychiatry clinics.

There are two confidence interval estimation procedures available. The first is the *Wald* confidence interval:

```
library(PropCIs)

diffpropci.Wald.mp(b = 2, c = 16, n = 41, conf.level = 0.90)
```

```
data:
90 percent confidence interval:
 0.196 0.487
sample estimates:
[1] 0.341
```

Also available in the `PropCIs` package is the *Agresti-Min* confidence interval.

```
diffpropci.mp(b = 2, c = 16, n = 41, conf.level = 0.90)
```

```
data:
90 percent confidence interval:
 0.180 0.471
sample estimates:
[1] 0.326
```

As you can see, each interval is quite wide here, and they also produce somewhat different point estimates, because they are making different sorts of approximations. These approaches, again, are only appropriate with **paired** comparisons of proportions.

Our estimate describes, assuming a clinic only gives an appointment to one of the two insurance types, the probability of Private insurance getting that appointment minus the probability of Public insurance getting the appointment.

24.4.1 What if we looked at all of the clinic types?

Here, again is the table for our complete set of 273 clinics.

```
specialty_wider %>%
  tabyl(Public, Private) %>%
  adorn_totals(where = c("row", "col")) %>%
  adorn_title()
```

		Private		
		Accepted	Denied	Total
Accepted	89	5	94	
Denied	155	24	179	
Total	244	29	273	

And here are the 90% Wald and Agresti-Min confidence intervals for matched pairs describing the proportion accepted for Private but not Public, minus the proportion accepted for Public but not Private.

```
diffpropci.Wald.mp(b = 5, c = 155, n = 273, conf.level = 0.90)
```

```
data:
```

```
90 percent confidence interval:
```

```
0.496 0.603
```

```
sample estimates:
```

```
[1] 0.549
```

```
diffpropci.mp(b = 5, c = 155, n = 273, conf.level = 0.90)
```

```
data:
```

```
90 percent confidence interval:
```

388 CHAPTER 24. COMPARING PROPORTIONS/RATES USING TWO PAIRED SAMPLES

```
0.492 0.599  
sample estimates:  
[1] 0.545
```

Chapter 25

Hypothesis Testing: What is it good for?

Hypothesis testing or significance testing uses sample data to attempt to reject the hypothesis that nothing interesting is happening – that is, to reject the notion that chance alone can explain the sample results¹. We can, in many settings, use confidence intervals to summarize the results, as well, and confidence intervals and hypothesis tests are closely connected.

In particular, it's worth stressing that:

- **A significant effect is not necessarily the same thing as an interesting effect.** For example, results calculated from large samples are nearly always “significant” even when the effects are quite small in magnitude. Before doing a test, always ask if the effect is large enough to be of any practical interest. If not, why do the test?
- **A non-significant effect is not necessarily the same thing as no difference.** A large effect of real practical interest may still produce a non-significant result simply because the sample is too small.
- **There are assumptions behind all statistical inferences.** Checking assumptions is crucial to validating the inference made by any test or confidence interval.

¹Some of this is adapted from @GoodHardin, and @Utts1999

25.1 Five Steps Required in Completing a Hypothesis Test

1. Specify the null hypothesis, H_0 (which usually indicates that there is no difference or no association between the results in various groups of subjects)
2. Specify the research or alternative hypothesis, H_A , sometimes called H_1 (which usually indicates that there is some difference or some association between the results in those same groups of subjects).
3. Specify the test procedure or test statistic to be used to make inferences to the population based on sample data. Here is where we usually specify α , the probability of incorrectly rejecting H_0 that we are willing to accept. In the absence of other information, we often use $\alpha = 0.05$
4. Obtain the data, and summarize it to obtain the relevant test statistic, which gets summarized as a p value.
5. Use the p value to either
 - **reject** H_0 in favor of the alternative H_A (concluding that there is a statistically significant difference/association at the α significance level) or
 - **retain** H_0 (and conclude that there is no statistically significant difference/association at the α significance level)

25.2 Type I and Type II Error: Power and Confidence

Once we know how unlikely the results would have been if the null hypothesis were true, we must make one of two choices:

1. The p value is not small enough to convincingly rule out chance. Therefore, we cannot reject the null hypothesis as an explanation for the results.
2. The p value was small enough to convincingly rule out chance. We reject the null hypothesis and accept the alternative hypothesis.

Making choice 2 is equivalent to declaring that the result is statistically significant. We can rephrase the two choices as:

1. There is no statistically significant difference or relationship in the data.
2. There is a statistically significant difference or relationship in the data.

How small must the p value be in order to rule out the null hypothesis? The standard choice is 5%. This standardization has advantages and disadvantages², and it is not compulsory. It is simply a convention that has become accepted

²Ingelfinger JA, Mosteller F, Thibodeau LA and Ware JH (1987) Biostatistics in Clinical Medicine, 2nd Edition, New York: MacMillan. pp. 156-157.

over the years, and there are many situations for which a 5% cutoff may be unwise. While it does give a specific, objectively chosen level to keep in mind, it suggests a rather mindless cutpoint having nothing to do with the importance of the decision nor the costs or losses associated with outcomes.

25.3 The Courtroom Analogy

Consider the analogy of the jury in a courtroom.

1. The evidence is not strong enough to convincingly rule out that the defendant is innocent. Therefore, we cannot reject the null hypothesis, or innocence of the defendant.
2. The evidence was strong enough that we are willing to rule out the possibility that an innocent person (as stated in the null hypothesis) produced the observed data. We reject the null hypothesis, that the defendant is innocent, and assert the alternative hypothesis.

Consistent with our thinking in hypothesis testing, in many cases we would not accept the hypothesis that the defendant is innocent. We would simply conclude that the evidence was not strong enough to rule out the possibility of innocence.

The p value is the probability of getting a result as extreme or more extreme than the one observed if the proposed null hypothesis is true. Notice that it is not valid to actually accept that the null hypothesis is true. To do so would be to say that we are essentially convinced that chance alone produced the observed results – a common mistake.

25.4 Significance vs. Importance

Remember that a statistically significant relationship or difference does not necessarily mean an important one. A result that is significant in the statistical meaning of the word may not be significant clinically. Statistical significance is a technical term. Findings can be both statistically significant and practically significant or either or neither.

When we have very large samples, we may find small differences statistically significant even though they have no clinical importance. At the other extreme, with small samples, even large differences will often not be significant at the levels usually required to recognize the difference as real. We must distinguish between statistical and practical/clinical significance.

25.5 Errors in Hypothesis Testing

In testing hypotheses, there are two potential decisions and each one brings with it the possibility that a mistake has been made.

Let's use the courtroom analogy. Here are the potential choices and associated potential errors. Although the seriousness of errors depends on the seriousness of the crime and punishment, the potential error for choice 2 is usually more serious.

1. We cannot rule out that the defendant is innocent, so (s)he is set free without penalty.
 - Potential Error: A criminal has been erroneously freed.
2. We believe that there is enough evidence to conclude that the defendant is guilty.
 - Potential Error: An innocent person is convicted / penalized and a guilty person remains free.

As another example, consider being tested for disease. Most tests for diseases are not 100% accurate. The lab technician or physician must make a choice:

1. In the opinion of the medical practitioner, you are healthy. The test result was weak enough to be called "negative" for the disease.
 - Potential Error: You are actually diseased but have been told you are not. This is called a **false negative**.
2. In the opinion of the medical practitioner, you are diseased. The test results were strong enough to be called "positive" for the disease.
 - Potential Error: You are actually healthy but have been told you are diseased. This is called a **false positive**.

25.6 The Two Types of Hypothesis Testing Errors

	H_A is true	H_0 is true
Test Rejects H_0	Correct Decision	Type I Error (False Positive)
Test Retains H_0	Type II Error (False Negative)	Correct Decision

- A Type I error can only be made if the null hypothesis is actually true.
- A Type II error can only be made if the alternative hypothesis is actually true.

25.7 The Significance Level, α , is the Probability of a Type I Error

If the null hypothesis is true, the p value is the probability of making an error by choosing the alternative hypothesis instead. Alpha (α) is defined as the probability of concluding significance [rejection of H_0] when there isn't (and H_0 is true, making a Type I error), also called the significance level, so that $100(1-\alpha)$ is the confidence level – the probability of correctly concluding that there is no difference (retaining H_0) when H_0 is true.

25.8 The Probability of avoiding a Type II Error is called Power, symbolized $1-\beta$

A Type II error is made if the alternative hypothesis is true, but you fail to choose it. The probability depends on exactly which part of the alternative hypothesis is true, so that computing the probability of making a Type II error is not feasible. The power of a test is the probability of making the correct decision when the alternative hypothesis is true. Beta (β) is defined as the probability of concluding that there was no difference, when in fact there was one (a Type II error). Power is then just $1 - \beta$, the probability of concluding that there was a difference, when, in fact, there was one.

Traditionally, people like the power of a test to be at least 80%, meaning that β is at most 0.20. Often, I'll be arguing for 90% as a minimum power requirement, or we'll be presenting a range of power calculations for a variety of sample size choices.

25.9 Incorporating the Costs of Various Types of Errors

Which error is more serious in medical testing, where we think of our H_0 : patient is healthy vs. H_A : disease is present?

It depends on the disease and on the consequences of a negative or positive test result. A false negative in a screening test for cancer could lead to a fatal delay in treatment, whereas a false positive would probably lead to a retest. A more troublesome example occurs in testing for an infectious disease. Inevitably, there is a trade-off between the two types of errors. It all depends on the consequences.

It would be nice if we could specify the probability that we were making an error with each potential decision. We could then weigh the consequence of the error against its probability. Unfortunately, in most cases, we can only specify the

conditional probability of making a Type I error, given that the null hypothesis is true.

In deciding whether to reject a null hypothesis, we will need to consider the consequences of the two potential types of errors. If a Type I error is very serious, then you should reject the null hypothesis only if the p value is very small. Conversely, if a Type II error is more serious, you should be willing to reject the null hypothesis with a larger p value, perhaps 0.10 or 0.20, instead of 0.05.

25.10 Relation of α and β to Error Types

- α is the probability of rejecting H_0 when H_0 is true.
 - So $1 - \alpha$, the confidence level, is the probability of retaining H_0 when that's the right thing to do.
- β is the probability of retaining H_0 when H_A is true.
 - So $1 - \beta$, the power, is the probability of rejecting H_0 when that's the right thing to do.

	H_A is True	H_0 is True
Test Rejects H_0	Correct Decision ($1 - \beta$)	Type I Error (α)
Test Retains H_0	Type II Error (β)	Correct Decision ($1 - \alpha$)

25.11 What does Dr. Love dislike about p values and “statistical significance”?

A lot of things. A major issue is that I believe that p values are impossible to explain in a way that is both [a] technically correct and [b] straightforward at the same time. As evidence of this, you might want to look at this article and associated video by Christie Aschwanden at 538.com

The notion of a p value was an incredibly impressive achievement back when Wald and others were doing the work they were doing in the 1940s, and might still have been useful as recently as 10 years ago. But the notion of a p value relies on a lot of flawed assumptions, and null hypothesis significance testing is fraught with difficulties. Nonetheless, researchers use p values every day.

25.12 The ASA Articles in 2016 and 2019 on Statistical Significance and P-Values

However, my primary motivation for taking the approach I'm taking in 431 and 432 going forward comes from the pieces in two key reference collections we'll read and discuss more thoroughly at the start of 432

1. The American Statistical Association's 2016 Statement on p-Values: Context, Process and Purpose.

The ASA Statement on p-Values and Statistical Significance (Wasserstein and Lazar 2016) was developed primarily because after decades, warnings about the don'ts had gone mostly unheeded. The statement was about what not to do, because there is widespread agreement about the don'ts.

2. Statistical Inference in the 21st Century: A World Beyond $p < 0.05$ from 2019 in *The American Statistician*

This is a world where researchers are free to treat “ $p = 0.051$ ” and “ $p = 0.049$ ” as not being categorically different, where authors no longer find themselves constrained to selectively publish their results based on a single magic number. In this world, where studies with “ $p < 0.05$ ” and studies with “ $p > 0.05$ ” are not automatically in conflict, researchers will see their results more easily replicated-and, even when not, they will better understand why. As we venture down this path, we will begin to see fewer false alarms, fewer overlooked discoveries, and the development of more customized statistical strategies. Researchers will be free to communicate all their findings in all their glorious uncertainty, knowing their work is to be judged by the quality and effective communication of their science, and not by their p-values. As “statistical significance” is used less, statistical thinking will be used more. The ASA Statement on P-Values and Statistical Significance started moving us toward this world.... Now we must go further.

The ASA Statement on P-Values and Statistical Significance stopped just short of recommending that declarations of “statistical significance” be abandoned. We take that step here. We conclude, based on our review of the articles in this special issue and the broader literature, that it is time to stop using the term “statistically significant” entirely. Nor should variants such as “significantly different,” “ $p < 0.05$,” and “nonsignificant” survive, whether expressed in words, by asterisks in a table, or in some other way.... Regardless of whether it was ever useful, a declaration of “statistical significance” has today become meaningless.

For the moment, I will say this. I emphasize confidence intervals over p values, which is at best a partial solution. But ...

1. Very rarely does a situation emerge in which a p value can be available in which looking at the associated confidence interval isn't far more helpful for making a comparison of interest.
2. The use of a p value requires making at least as many assumptions about the population, sample, individuals and data as does a confidence interval.
3. Most null hypotheses are clearly not exactly true prior to data collection, and so the test summarized by a p value is of questionable value most of the time.
4. No one has a truly adequate definition of a p value, in terms of both precision and parsimony. Brief, understandable definitions always fail to be technically accurate.
5. Bayesian approaches avoid some of these pitfalls, but come with their own issues.
6. Many smart people agree with me, and have sworn off of p values whenever they can.

Again, in 432, we'll look at these issues in far greater depth.

Chapter 26

Power and Sample Size Considerations

For most statistical tests, it is theoretically possible to estimate the power of the test in the design stage, (before any data are collected) for various sample sizes, so we can hone in on a sample size choice which will enable us to collect data only on as many subjects as are truly necessary.

A power calculation is likely the most common element of a scientific grant proposal on which a statistician is consulted. This is a fine idea in theory, but in practice...

- The tests that have power calculations worked out in intensive detail using R are mostly those with more substantial assumptions. Examples include t tests that assume population normality, common population variance and balanced designs in the independent samples setting, or paired t tests that assume population normality in the paired samples setting.
- These power calculations are also usually based on tests rather than confidence intervals, which would be much more useful in most settings. Simulation is your friend here.
- Even more unfortunately, this process of doing power and related calculations is **far more of an art than a science**.
- As a result, the value of many power calculations is negligible, since the assumptions being made are so arbitrary and poorly connected to real data.
- On several occasions, I have stood in front of a large audience of medical statisticians actively engaged in clinical trials and other studies that require power calculations for funding. When I ask for a show of hands of people who have had power calculations prior to such a study whose assumptions matched the eventual data perfectly, I get lots of laughs. It doesn't happen.

- Even the underlying framework that assumes a power of 80% with a significance level of 5% is sufficient for most studies is pretty silly.

All that said, I feel obliged to show you some examples of power calculations done using R, and provide some insight on how to make some of the key assumptions in a way that won't alert reviewers too much to the silliness of the enterprise. All of the situations described in this Chapter are toy problems, but they may be instructive about some fundamental ideas.

26.1 Sample Size and Power Considerations for a Single-Sample t test

For a t test, R can estimate any one of the following elements, given the other four, using the `power.t.test` command, for either a one-tailed or two-tailed single-sample t test...

- n = the sample size
- δ = delta = the true difference in population means between the null hypothesis value and a particular alternative
- s = sd = the true standard deviation of the population
- α = `sig.level` = the significance level for the test (maximum acceptable risk of Type I error)
- $1 - \beta$ = `power` = the power of the t test to detect the effect of size δ

26.1.1 A Toy Example

Suppose that in a recent health survey, the average beef consumption in the U.S. per person was 90 pounds per year. Suppose you are planning a new study to see if beef consumption levels have changed. You plan to take a random sample of 25 people to build your new estimate, and test whether the current pounds of beef consumed per year is 90. Suppose you want to do a two-sided (two-tailed) test at 95% confidence (so $\alpha = 0.05$), and that you expect that the true difference will need to be at least $\delta = 5$ pounds (i.e. 85 or less or 95 or more) in order for the result to be of any real, practical interest. Suppose also that you are willing to assume that the true standard deviation of the measurements in the population is 10 pounds.

That is, of course, a lot to suppose.

Now, we want to know what power the proposed experiment will have to detect a change of 5 pounds (or more) away from the original 90 pounds, with these specifications, and how tweaking these specifications will affect the power of the study.

So, we have - $n = 25$ data points to be collected - $\delta = 5$ pounds is the minimum clinically meaningful effect size - $s = 10$ is the assumed population standard deviation, in pounds per year - α is 0.05, and we'll do a two-sided test

26.1.2 Using the `power.t.test` function

```
power.t.test(n = 25, delta = 5, sd = 10, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

One-sample t test power calculation

```
n = 25
delta = 5
sd = 10
sig.level = 0.05
power = 0.67
alternative = two.sided
```

So, under this study design, we would expect to detect an effect of size $\delta = 5$ pounds with just under 67% power, i.e. with a probability of incorrect retention of H_0 of just about 1/3. Most of the time, we'd like to improve this power, and to do so, we'd need to adjust our assumptions.

26.1.3 Changing Assumptions in a Power Calculation

We made assumptions about the sample size n , the minimum clinically meaningful effect size (change in the population mean) δ , the population standard deviation s , and the significance level α , not to mention decisions about the test, like that we'd do a one-sample t test, rather than another sort of test for a single sample, and that we'd do a two-tailed, or two-sided test. Often, these assumptions are tweaked a bit to make the power look more like what a reviewer/funder is hoping to see.

26.1.4 Increasing the Sample Size, absent other changes, will Increase the Power

Suppose, we committed to using more resources and gathering data from 40 subjects instead of the 25 we assumed initially – what effect would this have on our power?

```
power.t.test(n = 40, delta = 5, sd = 10, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

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

```
n = 40
delta = 5
sd = 10
sig.level = 0.05
power = 0.869
alternative = two.sided
```

With more samples, we should have a more powerful test, able to detect the difference with greater probability. In fact, a sample of 40 paired differences yields 87% power. As it turns out, we would need at least 44 observations with this scenario to get to 90% power, as shown in the calculation below, which puts the power in, but leaves out the sample size.

```
power.t.test(power=0.9, delta = 5, sd = 10, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

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

```
n = 44
delta = 5
sd = 10
sig.level = 0.05
power = 0.9
alternative = two.sided
```

We see that we would need at least 44 observations to achieve 90% power. Note: we always round the sample size up in doing a power calculation – if this calculation had actually suggested $n = 43.1$ paired differences were needed, we would still have rounded up to 44.

26.1.5 Increasing the Effect Size, absent other changes, will increase the Power

A larger effect should be easier to detect. If we go back to our original calculation, which had 67% power to detect an effect of size $\delta = 5$, and now change the desired effect size to $\delta = 6$ pounds (i.e. a value of 84 or less or 96 or more), we should obtain a more powerful design.

```
power.t.test(n = 25, delta = 6, sd = 10, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

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

26.1. SAMPLE SIZE AND POWER CONSIDERATIONS FOR A SINGLE-SAMPLE T TEST401

```
n = 25
delta = 6
sd = 10
sig.level = 0.05
power = 0.821
alternative = two.sided
```

We see that this change in effect size from 5 to 6, leaving everything else the same, increases our power from 67% to 82%. To reach 90% power, we'd need to increase the effect size we were trying to detect to at least 6.76 pounds.

```
power.t.test(n = 25, power = 0.9, sd = 10, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

One-sample t test power calculation

```
n = 25
delta = 6.76
sd = 10
sig.level = 0.05
power = 0.9
alternative = two.sided
```

- Again, note that I am rounding up here.
- Using $\delta = 6.75$ would not quite make it to 90.00% power.
- Using $\delta = 6.76$ guarantees that the power will be 90% or more, and not just round up to 90%..

26.1.6 Decreasing the Standard Deviation, absent other changes, will increase the Power

The choice of standard deviation is usually motivated by a pilot study, or else pulled out of thin air - it's relatively easy to convince yourself that the true standard deviation might be a little smaller than you'd guessed initially. Let's see what happens to the power if we reduce the sample standard deviation from 10 pounds to 9. This should make the effect of 5 pounds easier to detect, because it will have smaller variation associated with it.

```
power.t.test(n = 25, delta = 5, sd = 9, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

One-sample t test power calculation

```
n = 25
```

```

delta = 5
sd = 9
sig.level = 0.05
power = 0.76
alternative = two.sided

```

This change in standard deviation from 10 to 9, leaving everything else the same, increases our power from 67% to nearly 76%. To reach 90% power, we'd need to decrease the standard deviation of the population paired differences to no more than 7.39 pounds.

```
power.t.test(n = 25, delta = 5, sd = NULL, power = 0.9, sig.level = 0.05,
             type="one.sample", alternative="two.sided")
```

One-sample t test power calculation

```

n = 25
delta = 5
sd = 7.4
sig.level = 0.05
power = 0.9
alternative = two.sided

```

Note I am rounding down here.

- Using s = 7.4 pounds would not quite make it to 90.00% power.

Note also that in order to get R to treat the sd as unknown, I must specify it as NULL in the formula...

26.1.7 Tolerating a Larger α (Significance Level), without other changes, increases Power

We can trade off some of our Type II error (lack of power) for Type I error. If we are willing to trade off some Type I error (as described by the α), we can improve the power. For instance, suppose we decided to run the original test with 90% confidence.

```
power.t.test(n = 25, delta = 5, sd = 10, sig.level = 0.1,
             type="one.sample", alternative="two.sided")
```

One-sample t test power calculation

```

n = 25
delta = 5
sd = 10

```

```

sig.level = 0.1
power = 0.783
alternative = two.sided

```

The calculation suggests that our power would thus increase from 67% to just over 78%.

26.2 Paired Sample t Tests and Power/Sample Size

For a paired-samples t test, R can estimate any one of the following elements, given the other four, using the `power.t.test` command, for either a one-tailed or two-tailed paired t test...

- n = the sample size (# of pairs) being compared
- δ = delta = the true difference in means between the two groups
- s = sd = the true standard deviation of the paired differences
- α = `sig.level` = the significance level for the comparison (maximum acceptable risk of Type I error)
- $1 - \beta$ = `power` = the power of the paired t test to detect the effect of size δ

26.3 A Toy Example

As a toy example, suppose you are planning a paired samples experiment involving $n = 30$ subjects who will each provide a “Before” and an “After” result, which is measured in days.

Suppose you want to do a two-sided (two-tailed) test at 95% confidence (so $\alpha = 0.05$), and that you expect that the true difference between the “Before” and “After” groups will have to be at least $\delta = 5$ days to be of any real interest. Suppose also that you are willing to assume that the true standard deviation of those paired differences will be 10 days.

That is, of course, a lot to suppose.

Now, we want to know what power the proposed experiment will have to detect this difference with these specifications, and how tweaking these specifications will affect the power of the study.

So, we have - $n = 30$ paired differences will be collected - $\delta = 5$ days is the minimum clinically meaningful difference - $s = 10$ days is the assumed population standard deviation of the paired differences - α is 0.05, and we'll do a two-sided test

26.4 Using the `power.t.test` function

```
power.t.test(n = 30, delta = 5, sd = 10, sig.level = 0.05,
             type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 30
delta = 5
sd = 10
sig.level = 0.05
power = 0.754
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

So, under this study design, we would expect to detect an effect of size $\delta = 5$ days with 75% power, i.e. with a probability of incorrect retention of H_0 of 0.25. Most of the time, we'd like to improve this power, and to do so, we'd need to adjust our assumptions.

26.5 Changing Assumptions in a Power Calculation

We made assumptions about the sample size n, the minimum clinically meaningful difference in means δ , the population standard deviation s, and the significance level α , not to mention decisions about the test, like that we'd do a paired t test, rather than another sort of test for paired samples, or use an independent samples approach, and that we'd do a two-tailed, or two-sided test. Often, these assumptions are tweaked a bit to make the power look more like what a reviewer/funder is hoping to see.

26.5.1 Increasing the Sample Size, absent other changes, will Increase the Power

Suppose, we committed to using more resources and gathering “Before” and “After” data from 40 subjects instead of the 30 we assumed initially – what effect would this have on our power?

```
power.t.test(n = 40, delta = 5, sd = 10, sig.level = 0.05,
             type="paired", alternative="two.sided")
```

```
Paired t test power calculation
```

```
n = 40
delta = 5
sd = 10
sig.level = 0.05
power = 0.869
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

With more samples, we should have a more powerful test, able to detect the difference with greater probability. In fact, a sample of 40 paired differences yields 87% power. As it turns out, we would need at least 44 paired differences with this scenario to get to 90% power, as shown in the calculation below, which puts the power in, but leaves out the sample size.

```
power.t.test(power=0.9, delta = 5, sd = 10, sig.level = 0.05,
             type="paired", alternative="two.sided")
```

```
Paired t test power calculation
```

```
n = 44
delta = 5
sd = 10
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

We see that we would need at least 44 paired differences to achieve 90% power.

Note: we always round the sample size up in doing a power calculation – if this calculation had actually suggested $n = 43.1$ paired differences were needed, we would still have rounded up to 44.

26.5.2 Increasing the Effect Size, absent other changes, will increase the Power

A larger effect should be easier to detect. If we go back to our original calculation, which had 75% power to detect an effect (i.e. a true population mean difference) of size $\delta = 5$, and now change the desired effect size to $\delta = 6$, we should obtain a more powerful design.

```
power.t.test(n = 30, delta = 6, sd = 10, sig.level = 0.05,
             type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 30
delta = 6
sd = 10
sig.level = 0.05
power = 0.888
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

We see that this change in effect size from 5 to 6, leaving everything else the same, increases our power from 75% to nearly 89%. To reach 90% power, we'd need to increase the effect size we were trying to detect to at least 6.13 days.

- Again, note that I am rounding up here.
- Using $\delta = 6.12$ would not quite make it to 90.00% power.
- Using $\delta=6.13$ guarantees that the power will be 90% or more, and not just round up to 90%..

26.5.3 Decreasing the Standard Deviation, absent other changes, will increase the Power

The choice of standard deviation is usually motivated by a pilot study, or else pulled out of thin air. It's relatively easy to convince yourself that the true standard deviation might be a little smaller than you'd guessed initially. Let's see what happens to the power if we reduce the sample standard deviation from 10 days to 9 days. This should make the effect of 5 days easier to detect as being different from the null hypothesized value of 0, because it will have smaller variation associated with it.

```
power.t.test(n = 30, delta = 5, sd = 9, sig.level = 0.05,
             type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 30
delta = 5
sd = 9
sig.level = 0.05
power = 0.837
```

```
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

This change in standard deviation from 10 to 9, leaving everything else the same, increases our power from 75% to nearly 84%. To reach 90% power, we'd need to decrease the standard deviation of the population paired differences to no more than 8.16 days.

Note I am rounding down here, because using $s = 8.17$ days would not quite make it to 90.00% power. Note also that in order to get R to treat the sd as unknown, I must specify it as NULL in the formula...

```
power.t.test(n = 30, delta = 5, sd = NULL, power = 0.9,
             sig.level = 0.05, type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 30
delta = 5
sd = 8.16
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

26.5.4 Tolerating a Larger α (Significance Level), without other changes, increases Power

We can trade off some of our Type II error (lack of power) for Type I error. If we are willing to trade off some Type I error (as described by the α), we can improve the power. For instance, suppose we decided to run the original test with 90% confidence.

```
power.t.test(n = 30, delta = 5, sd = 10, sig.level = 0.1,
             type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 30
delta = 5
sd = 10
sig.level = 0.1
power = 0.848
```

```
alternative = two.sided

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

The calculation suggests that our power would thus increase from 75% to nearly
85%.
```

26.6 Two Independent Samples: Power for t Tests

For an independent-samples t test, with a balanced design (so that $n_1 = n_2$), R can estimate any one of the following elements, given the other four, using the `power.t.test` command, for either a one-tailed or two-tailed t test...

- n = the sample size in each of the two groups being compared
- δ = delta = the true difference in means between the two groups
- s = sd = the true standard deviation of the individual values in each group (assumed to be constant – since we assume equal population variances)
- α = `sig.level` = the significance level for the comparison (maximum acceptable risk of Type I error)
- $1 - \beta$ = `power` = the power of the t test to detect the effect of size δ

This method only produces power calculations for balanced designs – where the sample size is equal in the two groups. If you want a two-sample power calculation for an unbalanced design, you will need to use a different library and function in R, as we'll see.

26.7 A New Example

Suppose we plan a study of the time to relapse for patients in a drug trial, where subjects will be assigned randomly to a (new) treatment or to a placebo. Suppose we anticipate that the placebo group will have a mean of about 9 months, and want to detect an improvement (increase) in time to relapse of 50%, so that the treatment group would have a mean of at least 13.5 months. We'll use $\alpha = .10$ and $\beta = .10$, as well. Assume we'd do a two-sided test, with an equal number of observations in each group, and we'll assume the observed standard deviation of 9 months in a pilot study will hold here, as well.

We want the sample size required by the test under a two sample setting where:

- $\alpha = .10$,
- with 90% power (so that $\beta = .10$),

- and where we will have equal numbers of samples in the placebo group (group 1) and the treatment group (group 2).
- We'll plug in the observed standard deviation of 9 months.
- We'll look at detecting a change from 9 [the average in the placebo group] to 13.5 (a difference of 50%, giving delta = 4.5)
- using a two-sided pooled t-test.

The appropriate R command is:

```
power.t.test(delta = 4.5, sd = 9,
             sig.level = 0.10, power = 0.9,
             type="two.sample",
             alternative="two.sided")
```

Two-sample t test power calculation

```
n = 69.2
delta = 4.5
sd = 9
sig.level = 0.1
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

This suggests that we will need a sample of at least 70 subjects in the treated group and an additional 70 subjects in the placebo group, for a total of 140 subjects.

26.7.1 Another Scenario

What if resources are sparse, and we'll be forced to do the study with no more than 120 subjects, overall? If we require 90% confidence in a two-sided test, what power will we have?

```
power.t.test(n = 60, delta = 4.5, sd = 9,
             sig.level = 0.10,
             type="two.sample",
             alternative="two.sided")
```

Two-sample t test power calculation

```
n = 60
delta = 4.5
```

```

sd = 9
sig.level = 0.1
power = 0.859
alternative = two.sided

NOTE: n is number in *each* group

```

It looks like the power under those circumstances would be just under 86%. Note that the $n = 60$ refers to half of the total sample size, since we'll need 60 drug and 60 placebo subjects in this balanced design.

26.8 Power for Independent Sample T tests with Unbalanced Designs

Using the `pwr` package, R can do sample size calculations that describe the power of a two-sample t test that does not require a balanced design using the `pwr.t2n.test` command.

Suppose we wanted to do the same study as we described above, using 100 “treated” patients but as few “placebo” patients as possible. What sample size would be required to maintain 90% power? There is one change here – the effect size d in the `pwr.t2n.test` command is specified using the difference in means δ that we used previously, divided by the standard deviation s that we used previously. So, in our old setup, we assumed $\delta = 4.5$, $s = 9$, so now we'll assume $d = 4.5/9$ instead.

```

pwr::pwr.t2n.test(n1 = 100, d = 4.5/9,
                   sig.level = 0.1, power = 0.9,
                   alternative="two.sided")

```

```

t test power calculation

n1 = 100
n2 = 52.8
d = 0.5
sig.level = 0.1
power = 0.9
alternative = two.sided

```

We would need at least 53 subjects in the “placebo” group.

26.8.1 The most efficient design for an independent samples comparison will be balanced.

- Note that if we use $n_1 = 100$ subjects in the treated group, we need at least $n_2 = 53$ in the placebo group to achieve 90% power, and a total of 153 subjects.
- Compare this to the balanced design, where we needed 70 subjects in each group to achieve the same power, thus, a total of 140 subjects.

We saw earlier that a test with 60 subjects in each group would yield just under 86% power. Suppose we instead built a test with 80 subjects in the treated group, and 40 in the placebo group, then what would our power be?

```
pwr::pwr.t2n.test(n1 = 80, n2 = 40, d = 4.5/9,
                   sig.level = 0.10,
                   alternative="two.sided")
```

```
t test power calculation

n1 = 80
n2 = 40
d = 0.5
sig.level = 0.1
power = 0.822
alternative = two.sided
```

As we'd expect, the power is stronger for a balanced design than for an unbalanced design with the same overall sample size.

Note that I used a two-sided test to establish my power calculation – in general, this is the most conservative and defensible approach for any such calculation, **unless there is a strong and specific reason to use a one-sided approach in building a power calculation, don't.**

26.9 Power and Sample Size for Comparing Two Population Proportions

We'll focus in 431 on comparisons of proportions using independent samples.

26.10 Tuberculosis Prevalence Among IV Drug Users

Pagano and Gauvreau (2000) describe a study to investigate factors affecting tuberculosis prevalence among intravenous drug users¹. Among 97 individuals who admit to sharing needles, 24 (24.7%) had a positive tuberculin skin test result; among 161 drug users who deny sharing needles, 28 (17.4%) had a positive test result. To start, we'll test the null hypothesis that the proportions of intravenous drug users who have a positive tuberculin skin test result are identical for those who share needles and those who do not.

```
twobytwo(24,73, 28, 133,
          "Sharing Needles", "Not Sharing",
          "TB test+", "TB test-")
```

2 by 2 table analysis:

	TB test+	TB test-	P(TB test+)	95% conf. interval
Sharing Needles	24	73	0.247	0.172 0.343
Not Sharing	28	133	0.174	0.123 0.240
<hr/>				
95% conf. interval				
Relative Risk: 1.4227 0.8772 2.307				
Sample Odds Ratio: 1.5616 0.8439 2.890				
Conditional MLE Odds Ratio: 1.5588 0.8014 3.019				
Probability difference: 0.0735 -0.0265 0.181				
<hr/>				
Exact P-value: 0.1996				
Asymptotic P-value: 0.1557				
<hr/>				

What conclusions should we draw?

26.11 Designing a New TB Study

Now, suppose we wanted to design a new study with as many non-sharers as needle-sharers participating, and suppose that we wanted to detect any difference in the proportion of positive skin test results between the two groups that was

¹Original Data Source: Graham NMH et al. (1992) Prevalence of Tuberculin Positivity and Skin Test Anergy in HIV-1-Seropositive and Seronegative Intravenous Drug Users. JAMA, 267, 369-373.

identical to the data presented above or larger with at least 90% power, using a two-sided test and $\alpha = .05$. What sample size would be required to accomplish these aims?

26.12 Using `power.prop.test` for Balanced Designs

Our constraints are that we want to find the sample size for a two-sample comparison of proportions using a balanced design, we will use $\alpha = .05$, and power = .90, and that we estimate that the non-sharers will have a .174 proportion of positive tests, and we will try to detect a difference between this group and the needle sharers, who we estimate will have a proportion of .247, using a two-sided hypothesis test.

```
power.prop.test(p1 = .174, p2 = .247, sig.level = 0.05, power = 0.90)
```

Two-sample comparison of proportions power calculation

```
n = 653
p1 = 0.174
p2 = 0.247
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

So, we'd need at least 654 non-sharing subjects, and 654 more who share needles to accomplish the aims of the study.

26.13 How `power.prop.test` works

`power.prop.test` works much like the `power.t.test` we saw for means.

Again, we specify 4 of the following 5 elements of the comparison, and R calculates the fifth.

- The sample size (interpreted as the # in each group, so half the total sample size)
- The true probability in group 1
- The true probability in group 2
- The significance level (α)
- The power ($1 - \beta$)

The big weakness with the `power.prop.test` tool is that it doesn't allow you to work with unbalanced designs.

26.14 Another Scenario

Suppose we can get exactly 800 subjects in total (400 sharing and 400 non-sharing). How much power would we have to detect a difference in the proportion of positive skin test results between the two groups that was identical to the data presented above or larger, using a one-sided test, with $\alpha = .10$?

```
power.prop.test(n=400, p1=.174, p2=.247, sig.level = 0.10,
                 alternative="one.sided")
```

Two-sample comparison of proportions power calculation

```
n = 400
p1 = 0.174
p2 = 0.247
sig.level = 0.1
power = 0.895
alternative = one.sided
```

NOTE: n is number in *each* group

We would have just under 90% power to detect such an effect.

26.15 Using the `pwr` library to assess sample size for Unbalanced Designs

The `pwr.2p2n.test` function in the `pwr` library can help assess the power of a test to determine a particular effect size using an unbalanced design, where n_1 is not equal to n_2 .

As before, we specify four of the following five elements of the comparison, and R calculates the fifth.

- `n1` = The sample size in group 1
- `n2` = The sample size in group 2
- `sig.level` = The significance level (α)
- `power` = The power ($1 - \beta$)
- `h` = the effect size h, which can be calculated separately in R based on the two proportions being compared: `p1` and `p2`.

26.15.1 Calculating the Effect Size h

To calculate the effect size for a given set of proportions, just use `ES.h(p1, p2)` which is available in the `pwr` library.

For instance, in our comparison, we have the following effect size.

```
pwr::ES.h(p1 = .174, p2 = .247)
```

```
[1] -0.18
```

26.16 Using `pwr.2p2n.test` in R

Suppose we can have 700 samples in group 1 (the not sharing group) but only half that many in group 2 (the group of users who share needles). How much power would we have to detect this same difference ($p_1 = .174$, $p_2 = .247$) with a 5% significance level in a two-sided test?

```
pwr::pwr.2p2n.test(h = pwr::ES.h(p1 = .174, p2 = .247),
                    n1 = 700, n2 = 350,
                    sig.level = 0.05)
```

```
difference of proportion power calculation for binomial distribution (arcsine transformation)

h = 0.18
n1 = 700
n2 = 350
sig.level = 0.05
power = 0.784
alternative = two.sided
```

NOTE: different sample sizes

Note that the headline for this output actually reads:

```
difference of proportion power calculation for binomial distribution
(arcsine transformation)
```

It appears we will have about 78% power under these circumstances.

26.16.1 Comparison to Balanced Design

How does this compare to the results with a balanced design using only 1000 drug users in total, so that we have 500 patients in each group?

```
pwr::pwr.2p2n.test(h = pwr::ES.h(p1 = .174, p2 = .247), n1 = 500, n2 = 500, sig.level =
```

difference of proportion power calculation for binomial distribution (arcsine transform)

```
h = 0.18
n1 = 500
n2 = 500
sig.level = 0.05
power = 0.811
alternative = two.sided
```

NOTE: different sample sizes

or we could instead have used...

```
power.prop.test(p1 = .174, p2 = .247, sig.level = 0.05, n = 500)
```

Two-sample comparison of proportions power calculation

```
n = 500
p1 = 0.174
p2 = 0.247
sig.level = 0.05
power = 0.809
alternative = two.sided
```

NOTE: n is number in *each* group

Note that these two sample size estimation approaches are approximations, and use slightly different approaches, so it's not surprising that the answers are similar, but not completely identical.

Chapter 27

A Review: Two Examples, Comparing Means

27.1 A Study of Battery Life

Should you buy generic rather than brand-name batteries? Bock, Velleman, and De Veaux (2004) describe a designed experiment to test battery life. A (male) student obtained six pairs of AA alkaline batteries from two major battery manufacturers; a well-known brand name and a generic brand, so that battery brand was the factor of interest.

To estimate the difference in mean lifetimes across the two manufacturers, the student kept a battery-powered CD player with the same CD running continuously, with the volume control fixed at 5, and measured the time until no more music was heard through the headphones. (He ran an initial trial to find out approximately how long that would take, so he didn't have to spend the first 3 hours of each run listening to the same CD.) The outcome was the time in minutes until the sound stopped. To account for changes in the CD player's performance over time, he randomized the run order by choosing pairs of batteries (the CD-player required two batteries to run) at random.

Here are the results for the 6 brand name and 6 generic tests, in minutes, found in the `battery.csv` data file, where `run` indicates the order in which the tests were run...

```
battery  
  
# A tibble: 12 x 4  
# ... with 4 variables:  
#   run    <int>   
#   test   <int>   
#   type   <fct>   
#   time   <dbl>  
# ... and 1 grouped variable:  
#   .by.   <dbl>   
#   #> #> #> #> #> #> #> #> #> #> #> #> #>
```

2	2	2	brand name	206.
3	6	3	brand name	199.
4	8	4	brand name	172.
5	9	5	brand name	184
6	12	6	brand name	170.
7	3	1	generic	194
8	4	2	generic	204.
9	5	3	generic	204.
10	7	4	generic	206.
11	10	5	generic	222.
12	11	6	generic	209.

27.1.1 Question 1. What is the outcome under study?

We are studying battery lifetimes (time until the sound stopped) in minutes.

27.1.2 Question 2. What are the treatment/exposure groups?

We are comparing the two brands of batteries: the well-known vs. the generic.

27.1.3 Question 3. Are the data collected using paired or independent samples?

Of course, if we had different numbers of samples in the two groups, then we'd know without further thought that independent samples were required. Since we have 6 observations in the brand name group, and also have 6 observations in the generic group, i.e. a balanced design, we need to pause now to decide whether paired or independent samples testing is appropriate in this setting.

Two samples are paired if each data point in one sample is naturally linked to a specific data point in the other sample. So, do we have paired or independent samples?

- Despite the way I've set up the data table, there is no particular reason to pair, say, run #1 (a brand name run) with any particular experimental run in the generic group. So the samples are independent. This is not a matched-pairs design.
- In each trial, the student either used two of the well-known batteries, or two of the generic batteries.
- Any of the tests/confidence intervals for the independent samples methods suggests a statistically significant (at the 5% level) difference between the generic and brand name batteries.

27.1.4 Question 4. Are the data a random sample from the population of interest?

Probably not. The data are likely to come from a convenient sample of batteries. I don't know how this might bias the study, though. It seems unlikely that there would be a particular bias unless, for example, the well-known batteries were substantially older or younger than the generic.

27.1.5 Question 5. What significance level will we use?

We have no reason not to use a 95% confidence level, so $\alpha = 0.05$

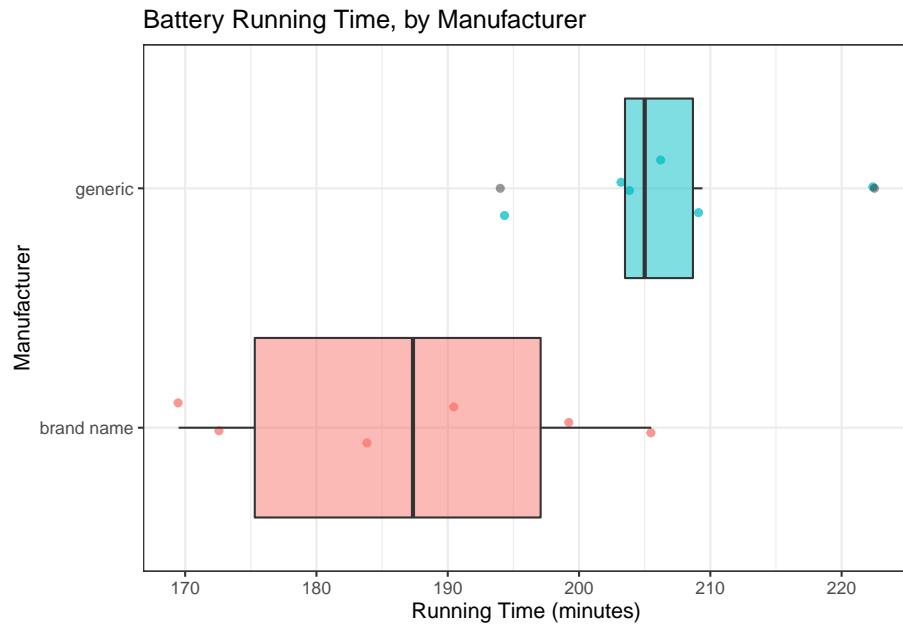
27.1.6 Question 6. Are we using a one-sided or two-sided comparison?

We could argue for a one-sided comparison, but I'll be safe and use the two-sided version.

27.1.7 Question 9. What does the distribution of outcomes in each group tell us?

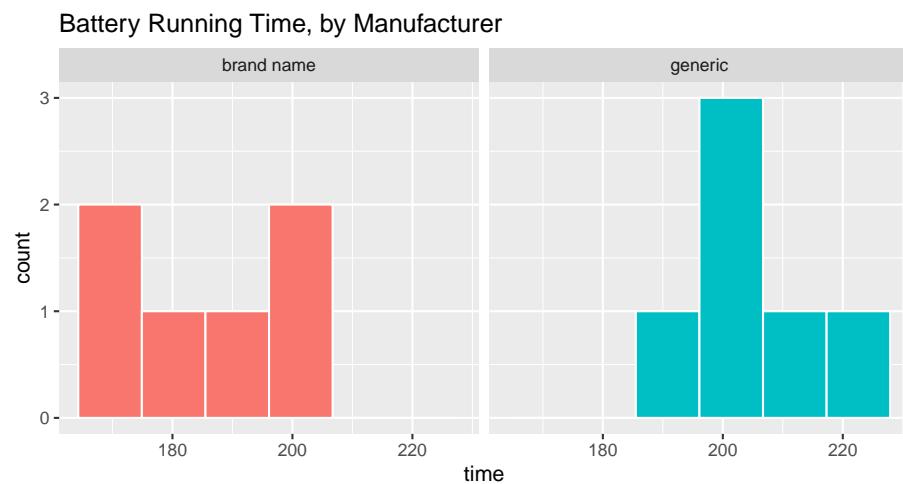
```
ggplot(battery, aes(x = type, y = time, fill = type)) +
  geom_jitter(aes(color = type), alpha = 0.75, width = 0.125) +
  geom_boxplot(alpha = 0.5) +
  theme_bw() +
  coord_flip() +
  guides(fill = FALSE, col = FALSE) +
  labs(title = "Battery Running Time, by Manufacturer",
       y = "Running Time (minutes)", x = "Manufacturer")
```

420 CHAPTER 27. A REVIEW: TWO EXAMPLES, COMPARING MEANS



We can generate histograms, too, but that's an issue, because we have so few observations.

```
ggplot(battery, aes(x = time, fill = type)) +
  geom_histogram(bins = 6, col = "white") +
  facet_wrap(~ type) +
  guides(fill = FALSE) +
  labs(title = "Battery Running Time, by Manufacturer")
```



```
by(battery$time, battery$type, mosaic::favstats)

battery$type: brand name
  min   Q1 median   Q3 max mean   sd n missing
  170 175    187 197 206 187 144.6      0
-----
battery$type: generic
  min   Q1 median   Q3 max mean   sd n missing
  194 204    205 209 222 207 9.376      0
```

It sure looks like the generic batteries lasted longer. And they also look like they were more consistent. The sample means are 206.6 for the generic group, 186.9 minutes for brand name, so the point estimate of the difference is 19.7 minutes.

The question is: can we be confident that the difference we observe here is more than just random fluctuation, at a 5% significance level?

27.1.8 Inferential Results for the Battery Study

In the table below, I have summarized the two-sided testing results for most of the ways in which we have looked at a two sample comparison so far, with 95% confidence intervals. If the samples really are paired, then we must choose from the paired samples comparisons described in the table. If the samples really are independent, then we must choose from the independent samples comparisons.

27.1.9 Paired Samples Approaches

	Method	<i>p</i> Value	95% CI for Generic - Brand Name
	Paired t	0.058	-1.0, 40.4
	Wilcoxon signed rank	0.063	-2.0, 39.9
	Bootstrap via smean.cl.boot	—	6.7, 33.0

27.1.10 Independent Samples Approaches

	Method	<i>p</i> Value	95% CI for Generic - Brand Name
	Pooled t	0.018	4.1, 35.3
	Welch's t	0.021	3.7, 35.6
	Wilcoxon Mann Whitney rank sum	0.030	3.3, 37.0
	Bootstrap via bootdif	—	7.7, 32.2

27.2 The Breakfast Study: Does Oat Bran Cereal Lower Serum LDL Cholesterol?

Norman and Streiner (2014) describe a crossover study that was conducted to investigate whether oat bran cereal helps to lower serum cholesterol levels in hypercholesterolemic males. Fourteen such individuals were randomly placed on a diet that included either oat bran or corn flakes; after two weeks, their low-density lipoprotein (LDL) cholesterol levels, in mmol/l were recorded. Each subject was then switched to the alternative diet. After a second two-week period, the LDL cholesterol level of each subject was again recorded.

`breakfast`

# A tibble: 14 x 3			
	subject	cornflakes	oatbran
	<int>	<dbl>	<dbl>
1	1	4.61	3.84
2	2	6.42	5.57
3	3	5.4	5.85
4	4	4.54	4.8
5	5	3.98	3.68
6	6	3.82	2.96
7	7	5.01	4.41
8	8	4.34	3.72
9	9	3.8	3.49
10	10	4.56	3.84
11	11	5.35	5.26
12	12	3.89	3.73
13	13	2.25	1.84
14	14	4.24	4.14

27.2.1 Question 1. What is the outcome under study?

We are studying levels of LDL cholesterol, in mmol/l. Note that if we wanted to convert to a more familiar scale, specifically mg/dl, we would multiply the mmol/l by 18, as it turns out.

27.2.2 Question 2. What are the treatment/exposure groups?

We are comparing subjects after two weeks of eating corn flakes to the same subjects after two weeks of eating oat bran.

27.2.3 Question 3. Are the data collected using paired or independent samples?

These are matched pairs, paired by subject. Each subject produced an oat bran result and a corn flakes result.

27.2.4 Question 4. Are the data a random sample from the population of interest?

Probably not. The data are likely to come from a convenient sample of 14 individuals but they were randomly assigned to cornflakes first or to oat bran first, then crossed over.

27.2.5 Question 5. What significance level will we use?

We have no reason not to use our usual 95% confidence level, so $\alpha = 0.05$

27.2.6 Question 6. Are we using a one-sided or two-sided comparison?

We could argue for a one-sided comparison, but I'll be safe and use the two-sided version.

27.2.7 Question 7. Did pairing help reduce nuisance variation?

After we drop the `breakfast.csv` file into the `breakfast` data frame, we look at the correlation of cornflakes and oatbran results across our 14 subjects.

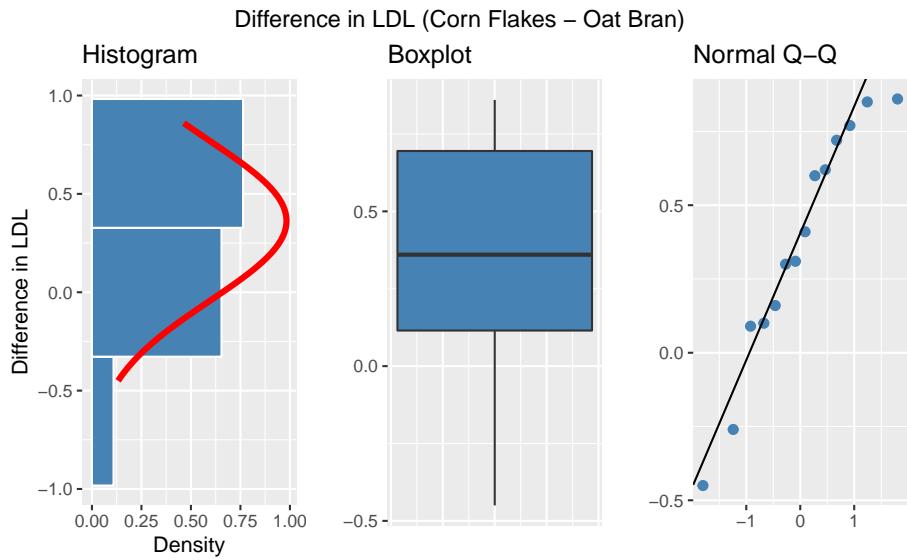
```
cor(breakfast$cornflakes, breakfast$oatbran)
```

```
[1] 0.923
```

The sample Pearson correlation coefficient is very strong and positive at 0.92, so the paired samples approach will use these data far more effectively than the (incorrect) independent samples approach.

27.2.8 Question 8. What does the distribution of paired differences tell us?

We summarize the distribution of the paired differences (cornflakes - oatbran) below.



The Normal distribution doesn't look too ridiculous in this case for the paired (cornflakes-oatbran) differences. Suppose we assume Normality and run the paired t test.

```
t.test(breakfast$cornflakes - breakfast$oatbran)
```

One Sample t-test

```
data: breakfast$cornflakes - breakfast$oatbran
t = 3, df = 13, p-value = 0.005
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.128 0.597
sample estimates:
mean of x
 0.363
```

Based on this sample of 14 subjects in the crossover study, we reject H_0 here and conclude that there is a statistically significant (detectable) difference between the LDL cholesterol levels after eating corn flakes and eating oat bran, at a 5% significance level.

27.3 Power, Sample Size and the Breakfast Study

These results look very promising. Suppose that in a new study, you wish to be able to detect a difference in LDL cholesterol between two exposures: subjects who eat cornflakes (as in the original study) and subjects who continue to eat cornflakes but also take a supplemental dosage of what you believe to be the crucial ingredient in oatbran.

Suppose you believe that the effect of taking the new supplement will be about half the size of the effect you observed in the original breakfast study on hypercholesterolemic males, but that males generally may be more likely to take your supplement regularly than switch from cornflakes to a less appetizing breakfast choice, making your supplement attractive.

What sample size will be required to yield 90% power to detect an effect half the size of the effect we observed in the breakfast study, in a new paired samples study using a two-tailed 5% significance level? What if we only required 80% power?

27.3.1 The Setup

We want to know n , the minimum required sample size for the new study, and we have:

- A specified effect size of half of what we saw in the breakfast study, where the sample mean difference between cornflakes and oatbran was 0.36 mmol/l, so our effect size is assumed to be $\delta = 0.18$ mmol/l.
- An assumed standard deviation equal to the standard deviation of the differences in the pilot breakfast study, which turns out to have been $s = 0.41$ mmol/l.
- We also have a pre-specified $\alpha = 0.05$ using a two-tailed test.
- We also want the power to be at least 90% for our new study.

27.3.2 The R Calculations

Question 1. What sample size will be required to yield 90% power to detect an effect half the size of the effect we observed in the breakfast study, in a new paired samples study using a two-tailed 5% significance level?

```
power.t.test(delta = 0.18, sd = 0.41, sig.level = 0.05,
             power = 0.9, type="paired", alternative="two.sided")
```

Paired t test power calculation

426 CHAPTER 27. A REVIEW: TWO EXAMPLES, COMPARING MEANS

```
n = 56.5
delta = 0.18
sd = 0.41
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

And so our new study will require at least **57 subjects** (each measured in two circumstances, so 114 total measurements) in order to achieve at least 90% power to detect the difference of 0.18 mmol/l while meeting these specifications.

Question 2. What if we were willing to accept only 80% power?

```
power.t.test(delta = 0.18, sd = 0.41, sig.level = 0.05,
              power = 0.8, type="paired", alternative="two.sided")
```

Paired t test power calculation

```
n = 42.7
delta = 0.18
sd = 0.41
sig.level = 0.05
power = 0.8
alternative = two.sided
```

NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs

It turns out that this would require at least **43 subjects**.

27.3.3 Independent samples, instead of paired samples?

What would happen if, instead of doing a paired samples study, we did one using independent samples? Assuming we used a balanced design, and assigned the same number of different people at random to either the oatbran supplement or regular cornflakes alone, we could do such a study, but it would require many more people to obtain similar power to the paired samples study.

```
power.t.test(delta = 0.18, sd = 0.41, sig.level = 0.05,
              power = 0.9, type="two.sample", alternative="two.sided")
```

Two-sample t test power calculation

```
n = 110
```

```
delta = 0.18
sd = 0.41
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

In all, **220 people** would be required in the independent samples study (110 in each exposure group), as compared to only **57 people** (each measured twice) in the paired study.

Chapter 28

Comparing 3 or more Population Means: Analysis of Variance

Recall the National Youth Fitness Survey, which we explored a small piece of in some detail back in Section 7. We'll look at a different part of the same survey here - specifically the 280 children whose data are captured in the `nyfs2` file.

```
nyfs2
```

```
# A tibble: 280 x 21
  subject.id sex    age.exam race.eth english income.cat3 income.detail
  <int> <fct>   <int> <fct>     <int> <fct>      <fct>
1 73228 Male     4 5 Other~      1 Low (below~ 0 to 4999
2 72393 Male     4 2 Non-H~      1 Low (below~ 0 to 4999
3 73303 Male     3 2 Non-H~      1 Low (below~ 0 to 4999
4 72786 Male     5 1 Non-H~      1 Low (below~ 0 to 4999
5 73048 Male     3 2 Non-H~      1 Low (below~ 0 to 4999
6 72556 Fema~    4 2 Non-H~      1 Low (below~ 0 to 4999
7 72580 Fema~    5 2 Non-H~      1 Low (below~ 0 to 4999
8 72532 Fema~    4 4 Other~      0 Low (below~ 0 to 4999
9 73012 Male     4 1 Non-H~      1 Low (below~ 0 to 4999
10 72099 Male    6 1 Non-H~      1 Low (below~ 0 to 4999
# ... with 270 more rows, and 14 more variables: inc.to.pov <dbl>,
# weight.kg <dbl>, height.cm <dbl>, bmi <dbl>, bmi.group <int>,
# bmi.cat <fct>, arm.length <dbl>, arm.circ <dbl>, waist.circ <dbl>,
# calf.circ <dbl>, calf.skinfold <dbl>, triceps.skinfold <dbl>,
# subscap.skinfold <dbl>, GMQ <int>
```

28.1 Comparing Gross Motor Quotient Scores by Income Level (3 Categories)

```
nyfs2a <- nyfs2 %>%
  select(subject.id, income.cat3, GMQ) %>%
  arrange(subject.id)
```

In this first analysis, we'll compare the population mean on the Gross Motor Quotient evaluation of these kids across three groups defined by income level. Higher values of this GMQ measure indicate improved levels of gross motor development, both in terms of locomotor and object control. See https://www.cdc.gov/Nchs/Nnyfs/Y_GMX.htm for more details.

```
nyfs2a %>%
  group_by(income.cat3) %>%
  summarise(n = n(), mean(GMQ), median(GMQ))
```

	# A tibble: 3 x 4	income.cat3	n	mean(GMQ)	median(GMQ)
		<fct>	<int>	<dbl>	<dbl>
1	High (65K or more)	92	95.7	97	
2	Low (below 25K)	98	97.0	97	
3	Middle (25 - 64K)	90	95.4	94	

Uh, oh. We should rearrange those income categories to match a natural order from low to high.

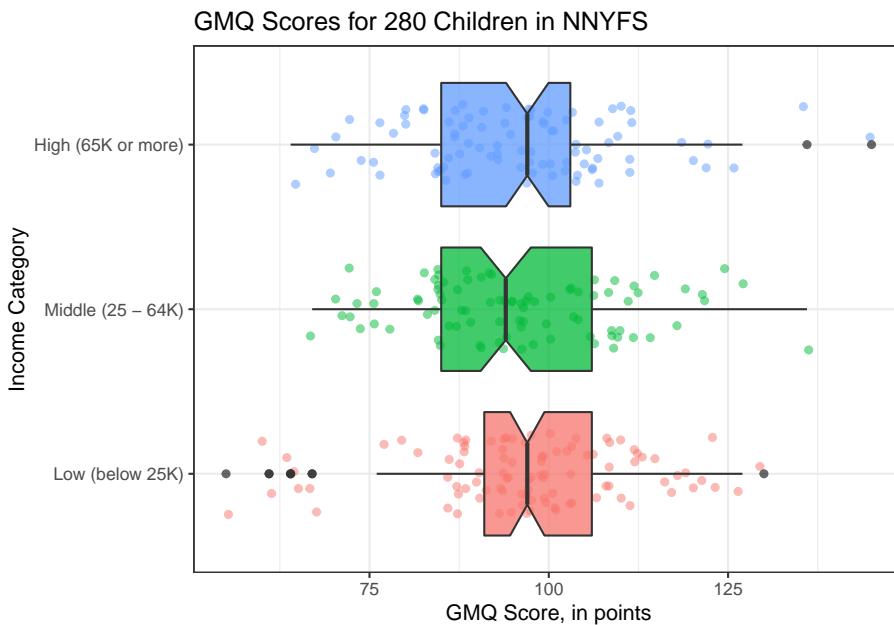
```
nyfs2a$income.cat3 <-
  forcats::fct_relevel(nyfs2a$income.cat3,
                        "Low (below 25K)",
                        "Middle (25 - 64K)",
                        "High (65K or more)")
```

28.1.1 Graphical Summaries

When working with three independent samples, I use graphs analogous to those we built for two independent samples.

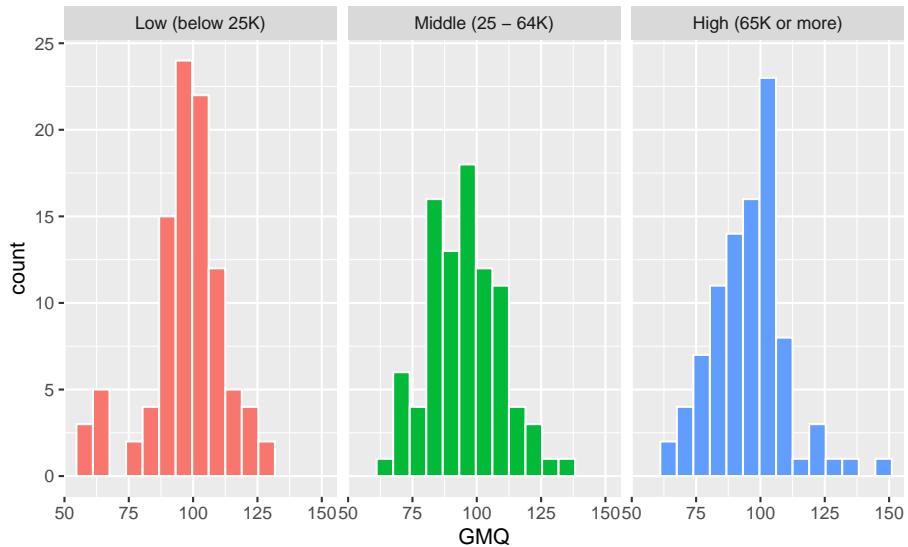
```
ggplot(nyfs2a, aes(x = income.cat3, y = GMQ, fill = income.cat3)) +
  geom_jitter(aes(color = income.cat3), alpha = 0.5, width = 0.25) +
  geom_boxplot(notch = TRUE, alpha = 0.75) +
  theme_bw() +
  coord_flip() +
  guides(fill = FALSE, col = FALSE) +
  labs(title = "GMQ Scores for 280 Children in NNYFS",
       y = "GMQ Score, in points", x = "Income Category")
```

28.1. COMPARING GROSS MOTOR QUOTIENT SCORES BY INCOME LEVEL (3 CATEGORIES) 431



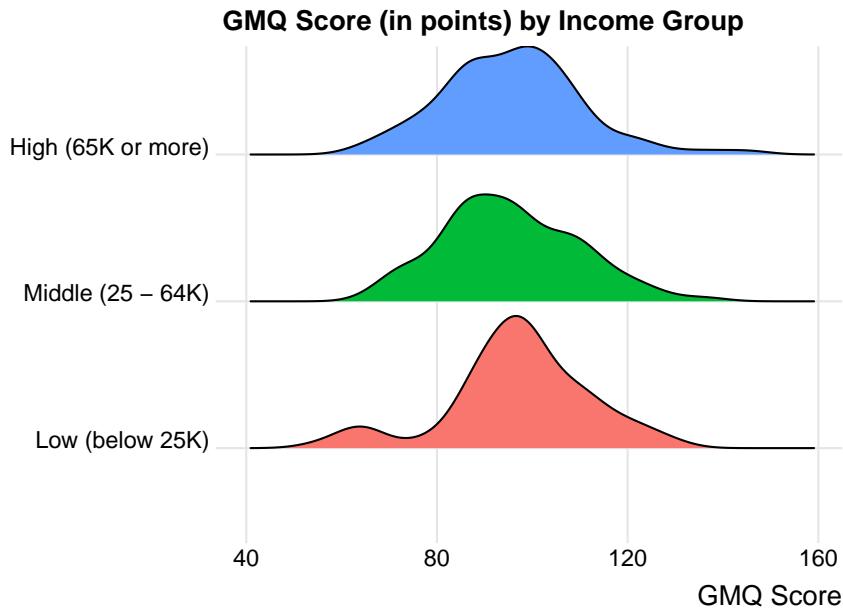
In addition to this comparison boxplot, we might consider faceted plots, like these histograms.

```
ggplot(nyfs2a, aes(x = GMQ, fill = income.cat3)) +  
  geom_histogram(bins = 15, col = "white") +  
  guides(fill = FALSE) +  
  facet_wrap(~ income.cat3)
```



Or, if we want to ignore the (modest) sample size differences, we might consider density functions, perhaps through a ridgeline plot.

```
ggplot(nyfs2a, aes(x = GMQ, y = income.cat3, fill = income.cat3)) +
  ggridges::geom_density_ridges(scale = 0.9) +
  guides(fill = FALSE) +
  labs(title = "GMQ Score (in points) by Income Group",
       x = "GMQ Score", y = "") +
  ggridges::theme_ridges()
```



28.1.2 Numerical Summaries

```
by(nyfs2a$GMQ, nyfs2a$income.cat3, mosaic::favstats)
```

```
nyfs2a$income.cat3: Low (below 25K)
  min   Q1   median   Q3   max   mean   sd   n missing
    55    91     97  106  130    97  14.8  98      0
-----
nyfs2a$income.cat3: Middle (25 - 64K)
  min   Q1   median   Q3   max   mean   sd   n missing
    67    85     94  106  136  95.4  14.2  90      0
-----
nyfs2a$income.cat3: High (65K or more)
  min   Q1   median   Q3   max   mean   sd   n missing
    64    85     97  103  145  95.7  14.5  92      0
```

28.2 Alternative Procedures for Comparing More Than Two Means

Now, if we only had two independent samples, we'd be choosing between a pooled t test, a Welch t test, and a non-parametric procedure like the Wilcoxon-

Mann-Whitney rank sum test, or even perhaps a bootstrap alternative.

In the case of more than two independent samples, we have methods analogous to the Welch test, and the rank sum test, and even the bootstrap, but we're going to be far more likely to select the **analysis of variance** (ANOVA) or an equivalent regression-based approach. These are the extensions of the pooled t test. Unless the sample outcome data are very clearly not Normally distributed, and no transformation is available which makes them appear approximately Normal in all of the groups we are comparing, we will stick with ANOVA.

28.2.1 Extending the Welch Test to > 2 Independent Samples

It is possible to extend the Welch two-sample t test (not assuming equal population variances) into an analogous one-factor analysis for comparing population means based on independent samples from more than two groups.

If we want to compare the population mean GMQ levels across those three income groups without assuming equal population variances, `oneway.test` is up to the task. The hypotheses being tested here are:

- $H_0: \mu_{Low} = \mu_{Middle} = \mu_{High}$ vs.
- H_A : At least one of the population means is different than the others.

```
oneway.test(GMQ ~ income.cat3, data = nyfs2a)
```

One-way analysis of means (not assuming equal variances)

```
data: GMQ and income.cat3
F = 0.3, num df = 2, denom df = 184, p-value = 0.7
```

At our usual 5% significance level, since the p value for the one-way test is 0.71, we'd conclude that we cannot reject H_0 and can only conclude that there is no significant difference between the true mean GMQ levels across the three income categories. You'll note that this isn't much help, though, because we don't have any measure of effect size, nor do we have any confidence intervals.

Like the analogous Welch t test, this approach allows us to forego the assumption of equal population variances in each of the three income groups, but it still requires us to assume that the populations are Normally distributed.

That said, most of the time when we have more than two levels of the factor of interest, we won't bother worrying about the equal population variance assumption, and will just use the one-factor ANOVA approach (with pooled variances) described below, to make the comparisons of interest.

28.2.2 Extending the Rank Sum Test to > 2 Independent Samples

It is also possible to extend the Wilcoxon-Mann-Whitney two-sample test into an analogous one-factor analysis called the **Kruskal-Wallis test** for comparing population measures of location based on independent samples from more than two groups.

If we want to compare the centers of the distributions of population GMQ score across our three income groups without assuming Normality, we can use `kruskal.test`.

The hypotheses being tested here are still as before, but the μ referred to here is a measure of location other than the population mean:

- $H_0: \mu_{Low} = \mu_{Middle} = \mu_{High}$ vs.
- H_A : At least one of the population means is different than the others.

```
kruskal.test(GMQ ~ income.cat3, data = nyfs2a)
```

```
Kruskal-Wallis rank sum test

data: GMQ by income.cat3
Kruskal-Wallis chi-squared = 2, df = 2, p-value = 0.3
```

So, in this case, at our usual 5% significance level, since the p value for the Kruskal-Wallis test is 0.31, we'd conclude that we cannot reject H_0 and can only conclude that there is no significant difference between the true mean GMQ scores across the three income categories. Again, note that this isn't much help, though, because we don't have any measure of effect size, nor do we have any confidence intervals.

That said, most of the time when we have more than two levels of the factor of interest, we won't bother worrying about potential violations of the Normality assumption unless they are glaring, and will just use the usual one-factor ANOVA approach (with pooled variances) described below, to make the comparisons of interest.

28.2.3 Can we use the bootstrap to compare more than two means?

Sure. There are both ANOVA and ANCOVA analogues using the bootstrap, and in fact, there are power calculations based on the bootstrap, too. If you want to see some example code, look at <https://sammancuso.com/2017/11/01/model-based-bootstrapped-anova-and-ancova/>

28.3 The Analysis of Variance

Extending the two-sample t test (assuming equal population variances) into a comparison of more than two samples uses the **analysis of variance** or ANOVA.

This is an analysis of a continuous outcome variable on the basis of a single categorical factor, in fact, it's often called one-factor ANOVA or one-way ANOVA to indicate that the outcome is being split up into the groups defined by a single factor.

The null hypothesis is that the population means are all the same, and the alternative is that this is not the case. When there are just two groups, then this boils down to an F test that is equivalent to the Pooled t test.

28.3.1 The `oneway.test` approach

R will produce some elements of a one-factor ANOVA using the `oneway.test` command:

```
oneway.test(GMQ ~ income.cat3, data = nyfs2a, var.equal=TRUE)
```

One-way analysis of means

```
data: GMQ and income.cat3
F = 0.3, num df = 2, denom df = 277, p-value = 0.7
```

This isn't the full analysis, though, which would require a more complete ANOVA table. There are two equivalent approaches to obtaining the full ANOVA table when comparing a series of 2 or more population means based on independent samples.

28.3.2 Using the `aov` approach and the `summary` function

Here's one possible ANOVA table, which doesn't require directly fitting a linear model.

```
summary(aov(GMQ ~ income.cat3, data = nyfs2a))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

28.3.3 Using the `anova` function after fitting a linear model

An equivalent way to get identical results in a slightly different format runs the linear model behind the ANOVA approach directly.

```
anova(lm(GMQ ~ income.cat3, data = nyfs2a))
```

Analysis of Variance Table

```
Response: GMQ
          Df Sum Sq Mean Sq F value Pr(>F)
income.cat3    2    146    72.8    0.35   0.71
Residuals   277  58174   210.0
```

28.4 Interpreting the ANOVA Table

28.4.1 What are we Testing?

The null hypothesis for the ANOVA table is that the population means of the outcome across the various levels of the factor of interest are all the same, against a two-sided alternative hypothesis that the level-specific population means are not all the same.

Specifically, if we have a grouping factor with k levels, then we are testing:

- $H_0: \mu_1 = \mu_2 = \dots = \mu_k$ vs.
- H_A : At least one of the population means $\mu_1, \mu_2, \dots, \mu_k$ is different from the others.

28.4.2 Elements of the ANOVA Table

The ANOVA table breaks down the variation in the outcome explained by the k levels of the factor of interest, and the variation in the outcome which remains (the Residual, or Error).

Specifically, the elements of the ANOVA table are:

1. the degrees of freedom (labeled Df) for the factor of interest and for the Residuals
2. the sums of squares (labeled Sum Sq) for the factor of interest and for the Residuals
3. the mean square (labeled Mean Sq) for the factor of interest and for the Residuals
4. the ANOVA F test statistic (labeled F value), which is used to generate
5. the p value for the comparison assessed by the ANOVA model, labeled $\text{Pr}(>F)$

28.4.3 The Degrees of Freedom

```
anova(lm(GMQ ~ income.cat3, data = nyfs2a))
```

Analysis of Variance Table

Response: GMQ

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

- The **degrees of freedom** attributable to the factor of interest (here, Income category) is the number of levels of the factor minus 1. Here, we have three Income categories (levels), so $df(income.cat3) = 2$.
- The total degrees of freedom are the number of observations (across all levels of the factor) minus 1. We have 280 GMQ scores in the `nyfs2a` data, so the $df(\text{Total})$ must be 279, although the Total row isn't shown by R in its output.
- The Residual degrees of freedom are the Total df - Factor df. So, here, that's $279 - 2 = 277$.

28.4.4 The Sums of Squares

```
anova(lm(GMQ ~ income.cat3, data = nyfs2a))
```

Analysis of Variance Table

Response: GMQ

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

- The sum of squares (often abbreviated SS or Sum Sq) represents variation explained.
- The factor SS is the sum across all levels of the factor of the sample size for the level multiplied by the squared difference between the level mean and the overall mean across all levels. Here, $SS(income.cat3) = 146$
- The total SS is the sum across all observations of the square of the difference between the individual values and the overall mean. Here, that is $146 + 58174 = 58320$
- Residual SS = Total SS - Factor SS.
- Also of interest is a calculation called η^2 , ("eta-squared"), which is equivalent to R^2 in a linear model.
 - $\eta^2 = SS(\text{Factor})/SS(\text{Total})$ = the proportion of variation in our outcome (here, GMQ) explained by the variation between groups

(here, income groups)

- In our case, $\eta^2 = 146 / (146 + 58174) = 146 / 58320 = 0.0025$
- So, Income Category alone accounts for about 0.25% of the variation in GMQ levels observed in these data.

28.4.5 The Mean Square

```
anova(lm(GMQ ~ income.cat3, data = nyfs2a))
```

Analysis of Variance Table

Response: GMQ

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

- The Mean Square is the Sum of Squares divided by the degrees of freedom, so $MS(\text{Factor}) = SS(\text{Factor})/\text{df}(\text{Factor})$.
- In our case, $MS(\text{income.cat3}) = SS(\text{income.cat3})/\text{df}(\text{income.cat3}) = 146 / 2 = 72.848$ (notice that R maintains more decimal places than it shows for these calculations) and
- $MS(\text{Residuals}) = SS(\text{Residuals}) / \text{df}(\text{Residuals}) = 58174 / 277 = 210.014$.
 - $MS(\text{Residuals})$ or $MS(\text{Error})$ is an estimate of the residual variance which corresponds to σ^2 in the underlying linear model for the outcome of interest, here GMQ.

28.4.6 The F Test Statistic and p Value

```
anova(lm(GMQ ~ income.cat3, data = nyfs2a))
```

Analysis of Variance Table

Response: GMQ

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

- The ANOVA F test is obtained by calculating $MS(\text{Factor}) / MS(\text{Residuals})$. So in our case, $F = 72.848 / 210.014 = 0.3469$
- The F test statistic is then compared to a specific F distribution to obtain a p value, which is shown here to be 0.7072
- Specifically, the observed F test statistic is compared to an F distribution with numerator df = Factor df, and denominator df = Residual df to obtain the p value.

- Here, we have $SS(\text{Factor}) = 146$ (approximately), and $df(\text{Factor}) = 2$, leaving $MS(\text{Factor}) = 72.848$
- We have $SS(\text{Residual}) = 58174$, and $df(\text{Residual}) = 277$, leaving $MS(\text{Residual}) = 210.014$
- $MS(\text{Factor}) / MS(\text{Residual}) = F \text{ value} = 0.3469$, which, when compared to an F distribution with 2 and 277 degrees of freedom, yields a p value of 0.7072

28.5 The Residual Standard Error

The residual standard error is simply the square root of the variance estimate $MS(\text{Residual})$. Here, $MS(\text{Residual}) = 210.014$, so the Residual standard error = 14.49 points.

28.6 The Proportion of Variance Explained by the Factor, or η^2

We will often summarize the proportion of the variation explained by the factor. The summary statistic is called eta-squared (η^2), and is equivalent to the R^2 value we have seen previously in linear regression models.

Again, $\eta^2 = SS(\text{Factor}) / SS(\text{Total})$

Here, we have - $SS(\text{income.cat3}) = 146$ and $SS(\text{Residuals}) = 58174$, so $SS(\text{Total}) = 58320$ - Thus, $\eta^2 = SS(\text{Factor})/SS(\text{Total}) = 146/58320 = 0.0025$

The income category accounts for 0.25% of the variation in GMQ levels: only a tiny fraction.

28.7 The Regression Approach to Compare Population Means based on Independent Samples

This approach is equivalent to the ANOVA approach, and thus also (when there are just two samples to compare) to the pooled-variance t test. We run a linear regression model to predict the outcome (here, GMQ) on the basis of the categorical factor with three levels (here, `income.cat3`)

```
summary(lm(GMQ ~ income.cat3, data=nyfs2a))
```

Call:

28.7. THE REGRESSION APPROACH TO COMPARE POPULATION MEANS BASED ON INDEPENDENT SAMPLES

```
lm(formula = GMQ ~ income.cat3, data = nyfs2a)

Residuals:
    Min      1Q  Median      3Q     Max 
-42.03 -9.03 -0.03  8.97 49.27 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    97.03     1.46   66.28 <2e-16 ***
income.cat3Middle (25 - 64K) -1.66     2.12   -0.79    0.43  
income.cat3High (65K or more) -1.30     2.10   -0.62    0.54  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.5 on 277 degrees of freedom
Multiple R-squared:  0.0025,    Adjusted R-squared:  -0.0047 
F-statistic: 0.347 on 2 and 277 DF,  p-value: 0.707
```

28.7.1 Interpreting the Regression Output

This output tells us many things, but for now, we'll focus just on the coefficients output, which tells us that:

- the point estimate for the population mean GMQ score across “Low” income subjects is 97.03
- the point estimate (sample mean difference) for the difference in population mean GMQ level between the “Middle” and “Low” income subjects is -1.66 (in words, the Middle income kids have lower GMQ scores than the Low income kids by 1.66 points on average.)
- the point estimate (sample mean difference) for the difference in population mean GMQ level between the “High” and “Low” income subjects is -1.30 (in words, the High income kids have lower GMQ scores than the Low income kids by 1.30 points on average.)

Of course, we knew all of this already from a summary of the sample means.

```
nyfs2a %>%
  group_by(income.cat3) %>%
  summarise(n = n(), mean(GMQ))

# A tibble: 3 x 3
  income.cat3      n `mean(GMQ)` 
  <fct>        <int>     <dbl>    
1 Low (below 25K)    98      97.0    
2 Middle (25 - 64K)  90      95.4    
3 High (65K or more) 92      95.7
```

The model for predicting GMQ is based on two binary (1/0) indicator variables, specifically, we have:

- Estimated GMQ = 97.03 - 1.66 x [1 if Middle income or 0 if not] - 1.30 x [1 if High income or 0 if not]

The coefficients section also provides a standard error and t statistic and two-sided *p* value for each coefficient.

28.7.2 The Full ANOVA Table

To see the full ANOVA table corresponding to any linear regression model, we run...

```
anova(lm(GMQ ~ income.cat3, data=nyfs2a))
```

Analysis of Variance Table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	146	72.8	0.35	0.71
Residuals	277	58174	210.0		

28.7.3 ANOVA Assumptions

The assumptions behind analysis of variance are the same as those behind a linear model. Of specific interest are:

- The samples obtained from each group are independent.
- Ideally, the samples from each group are a random sample from the population described by that group.
- In the population, the variance of the outcome in each group is equal. (This is less of an issue if our study involves a balanced design.)
- In the population, we have Normal distributions of the outcome in each group.

Happily, the F test is fairly robust to violations of the Normality assumption.

28.8 Equivalent approach to get ANOVA Results

$H_0: \mu_{Low} = \mu_{Middle} = \mu_{High}$ vs. $H_A: H_0$ not true.

```
summary(aov(GMQ ~ income.cat3, data = nyfs2a))
```

```
Df Sum Sq Mean Sq F value Pr(>F)
income.cat3  2    146    72.8    0.35   0.71
Residuals  277  58174   210.0
```

So which of the pairs of means are significantly different?

28.9 The Problem of Multiple Comparisons

1. Suppose we compare High to Low, using a test with $\alpha = 0.05$
2. Then we compare Middle to Low on the same outcome, also using $\alpha = 0.05$
3. Then we compare High to Middle, also with $\alpha = 0.05$

What is our overall α level across these three comparisons?

- It could be as bad as $0.05 + 0.05 + 0.05$, or 0.15.
- Rather than our nominal 95% confidence, we have something as low as 85% confidence across this set of simultaneous comparisons.

28.9.1 The Bonferroni solution

1. Suppose we compare High to Low, using a test with $\alpha = 0.05/3$
2. Then we compare Middle to Low on the same outcome, also using $\alpha = 0.05/3$
3. Then we compare High to Middle, also with $\alpha = 0.05/3$

Then across these three comparisons, our overall α can be (at worst)

- $0.05/3 + 0.05/3 + 0.05/3 = 0.05$
- So by changing our nominal confidence level from 95% to 98.333% in each comparison, we wind up with at least 95% confidence across this set of simultaneous comparisons.
- This is a conservative (worst case) approach.

Goal: Simultaneous p values comparing White vs AA, AA vs Other and White vs Other

```
pairwise.t.test(nyfs2a$GMQ, nyfs2a$income.cat3, p.adjust="bonferroni")
```

```
Pairwise comparisons using t tests with pooled SD

data: nyfs2a$GMQ and nyfs2a$income.cat3

Low (below 25K)  Middle (25 - 64K)
Middle (25 - 64K) 1          -
High (65K or more) 1          1
```

```
P value adjustment method: bonferroni
```

In this case, none of these p values are even remotely close to being small enough to indicate statistical significance. They're all 1, in effect.

28.9.2 Pairwise Comparisons of Group Means using Tukey's Honestly Significant Differences

Goal: Simultaneous (less conservative) confidence intervals and p values for our three pairwise comparisons (High vs. Low, High vs. Middle, Middle vs. Low)

```
TukeyHSD(aov(GMQ ~ income.cat3, data = nyfs2a))
```

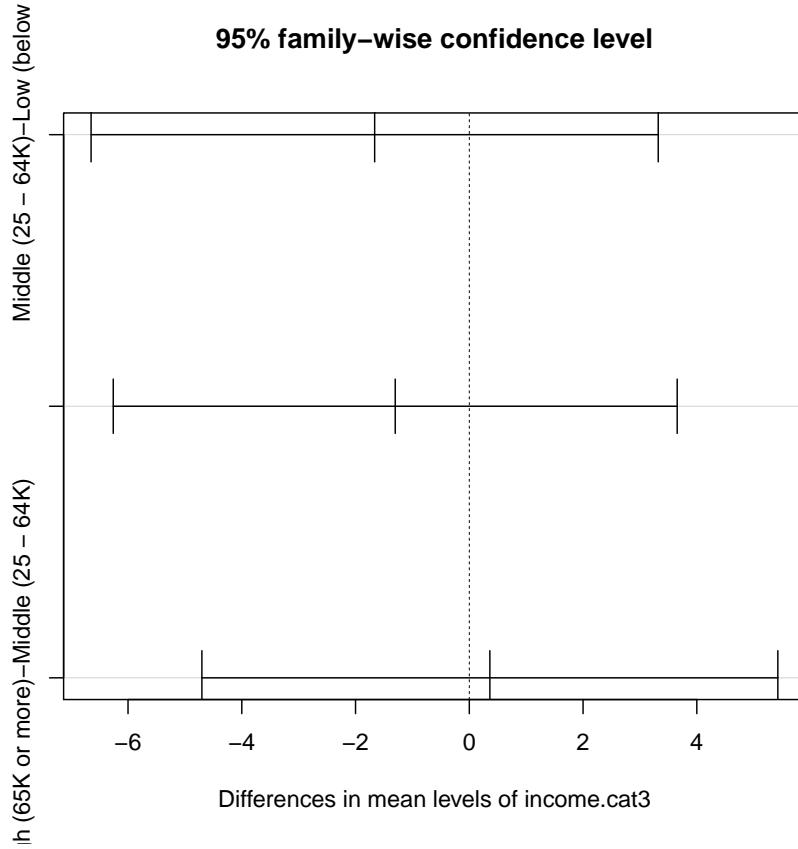
```
Tukey multiple comparisons of means
95% family-wise confidence level
```

```
Fit: aov(formula = GMQ ~ income.cat3, data = nyfs2a)
```

```
$income.cat3
            diff    lwr   upr p adj
Middle (25 - 64K)-Low (below 25K) -1.664 -6.65 3.32 0.712
High (65K or more)-Low (below 25K) -1.302 -6.26 3.65 0.810
High (65K or more)-Middle (25 - 64K)  0.362 -4.70 5.42 0.985
```

28.9.3 Plotting the Tukey HSD results

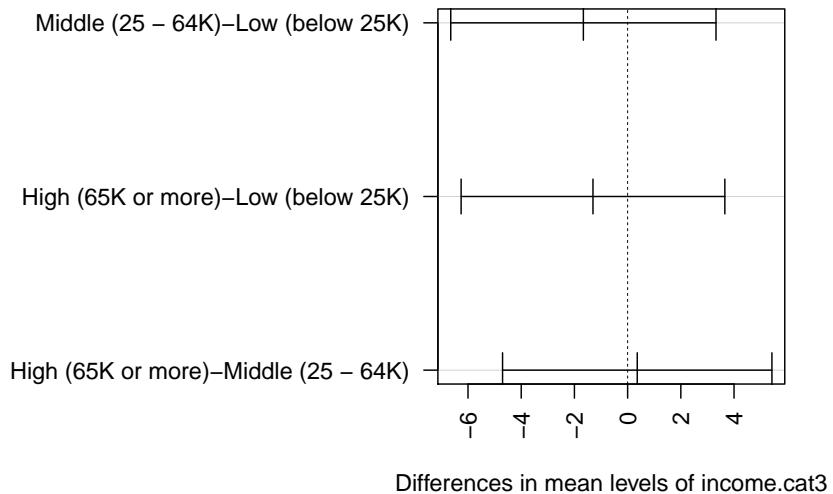
```
plot(TukeyHSD(aov(GMQ ~ income.cat3, data = nyfs2a)))
```



Note that the default positioning of the y axis in the plot of Tukey HSD results can be problematic. If we have longer names, in particular, for the levels of our factor, R will leave out some of the labels. We can alleviate that problem either by using the `fct_recode` function in the `forcats` package to rename the factor levels, or we can use the following code to reconfigure the margins of the plot.

```
mar.default <- c(5,6,4,2) + 0.1 # save default plotting margins

par(mar = mar.default + c(0, 12, 0, 0))
plot(TukeyHSD(aov(GMQ ~ income.cat3, data = nyfs2a)), las = 2)
```

95% family-wise confidence level

Differences in mean levels of income.cat3

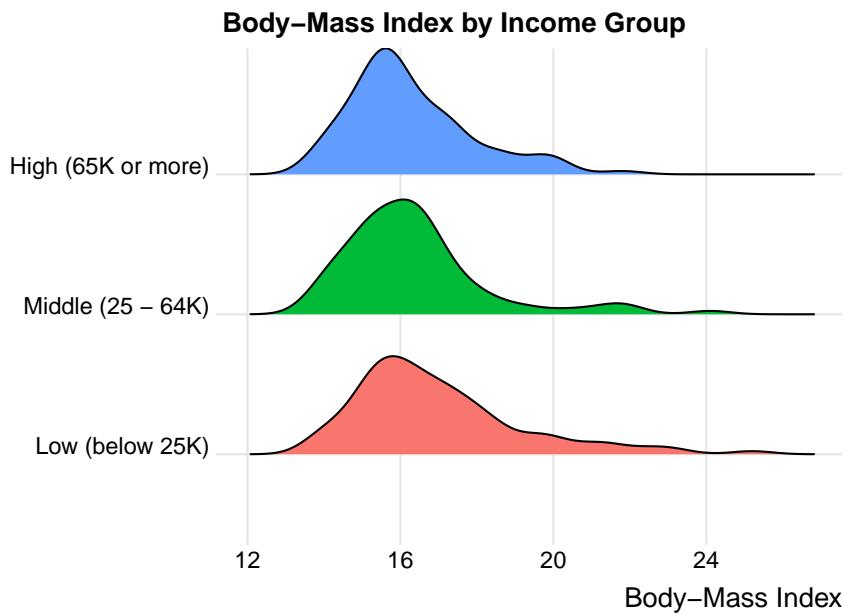
```
par(mar = mar.default) # return to normal plotting margins
```

28.10 What if we consider another outcome, BMI?

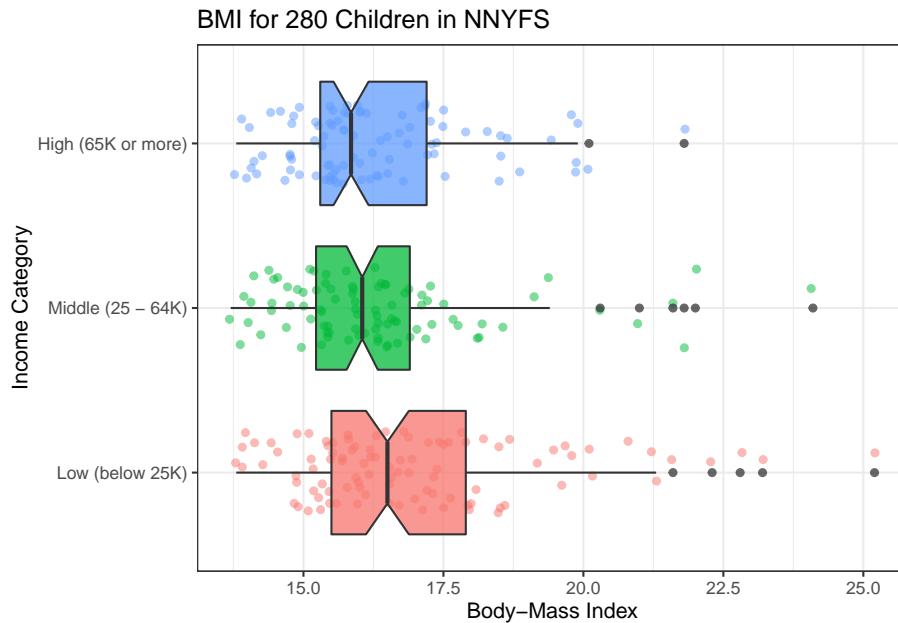
We'll look at the full data set in `nyfs2` now, so we can look at BMI as a function of income.

```
nyfs2$income.cat3 <-
  forcats::fct_relevel(nyfs2$income.cat3,
    "Low (below 25K)",
    "Middle (25 – 64K)",
    "High (65K or more)")

ggplot(nyfs2, aes(x = bmi, y = income.cat3, fill = income.cat3)) +
  ggridges::geom_density_ridges(scale = 0.9) +
  guides(fill = FALSE) +
  labs(title = "Body-Mass Index by Income Group",
       x = "Body-Mass Index", y = "") +
  ggridges::theme_ridges()
```



```
ggplot(nyfs2, aes(x = income.cat3, y = bmi, fill = income.cat3)) +  
  geom_jitter(aes(color = income.cat3), alpha = 0.5, width = 0.25) +  
  geom_boxplot(notch = TRUE, alpha = 0.75) +  
  theme_bw() +  
  coord_flip() +  
  guides(fill = FALSE, col = FALSE) +  
  labs(title = "BMI for 280 Children in NNYFS",  
       y = "Body–Mass Index", x = "Income Category")
```



Here are the descriptive numerical summaries:

```
by(nyfs2$bmi, nyfs2$income.cat3, mosaic::favstats)
```

```
nyfs2$income.cat3: Low (below 25K)
  min   Q1 median   Q3 max mean   sd n missing
  13.8 15.5  16.5 17.9 25.2 17 2.19 98      0
```

```
-----
```

```
nyfs2$income.cat3: Middle (25 - 64K)
  min   Q1 median   Q3 max mean   sd n missing
  13.7 15.2  16.1 16.9 24.1 24.1 1.64 90      0
```

```
-----
```

```
nyfs2$income.cat3: High (65K or more)
  min   Q1 median   Q3 max mean   sd n missing
  13.8 15.3  15.9 17.2 21.8 16.3 1.61 92      0
```

Here is the ANOVA table.

```
anova(lm(bmi ~ income.cat3, data = nyfs2))
```

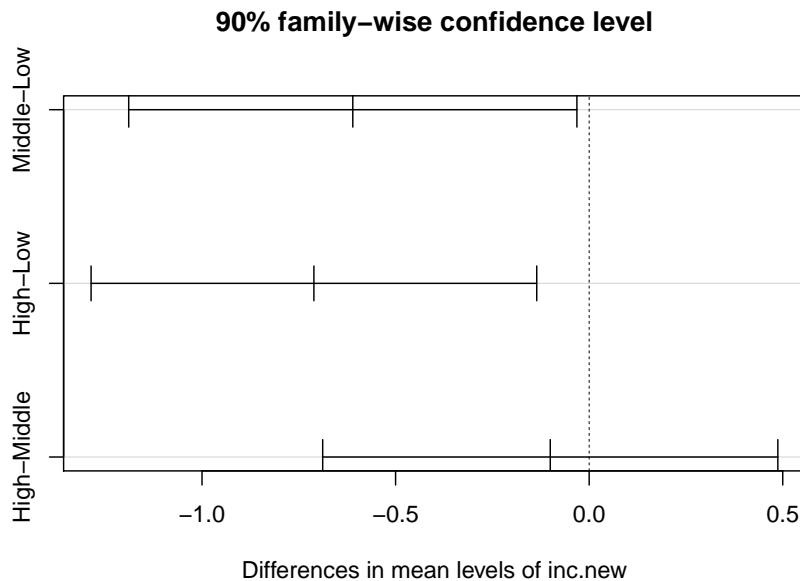
Analysis of Variance Table

Response: bmi					
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
income.cat3	2	28	14.2	3.83	0.023 *
Residuals	277	1025	3.7		

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And we see a statistically significant result at usual α levels. Let's consider the Tukey HSD results. First, we'll create a factor with shorter labels.

```
nyfs2$inc.new <-  
  forcats::fct_recode(nyfs2$income.cat3,  
    "Low" = "Low (below 25K)",  
    "Middle" = "Middle (25 - 64K)",  
    "High" = "High (65K or more)")  
  
plot(TukeyHSD(aov(bmi ~ inc.new, data = nyfs2),  
  conf.level = 0.90))
```



It appears that there is a statistically significant difference between the `bmi` means of the “Low” group and both the “High” and “Middle” group at the 10% significance level, but no significant difference between “Middle” and “High.” Details of those p values and confidence intervals for those pairwise comparisons follow.

```
TukeyHSD(aov(bmi ~ inc.new, data = nyfs2),  
  conf.level = 0.90)
```

```
Tukey multiple comparisons of means  
90% family-wise confidence level
```

```
Fit: aov(formula = bmi ~ inc.new, data = nyfs2)
```

450CHAPTER 28. COMPARING 3 OR MORE POPULATION MEANS: ANALYSIS OF VARIANCE

```
$inc.new
      diff     lwr      upr p adj
Middle-Low -0.611 -1.189 -0.0317 0.078
High-Low   -0.711 -1.287 -0.1354 0.031
High-Middle -0.100 -0.688  0.4874 0.934
```

Chapter 29

Comparisons with Contingency Tables

29.1 Larger Contingency Tables - Testing for Independence

What will we do with tables describing data from more than two categories at a time, returning to the notion of independent (rather than paired or matched) samples? The chi-square tests we have already seen in our `twobytwo` table output will extend nicely to this scenario, especially the Pearson χ^2 (asymptotic) test.

29.2 A 2x3 Table: Comparing Response to Active vs. Placebo

The table below, containing 2 rows and 3 columns of data (ignoring the marginal totals) specifies the number of patients who show *complete*, *partial*, or *no response* after treatment with either **active** medication or a **placebo**.

Group	None	Partial	Complete
Active	16	26	29
Placebo	24	26	18

Is there a statistically significant association here? That is to say, is there a statistically significant difference between the treatment groups in the distribution of responses?

29.2.1 Getting the Table into R

To answer this, we'll have to get the data from this contingency table into a matrix in R. Here's one approach...

```
T1 <- matrix(c(16,26,29,24,26,18), ncol=3, nrow=2, byrow=TRUE)
rownames(T1) <- c("Active", "Placebo")
colnames(T1) <- c("None", "Partial", "Complete")
```

T1

	None	Partial	Complete
Active	16	26	29
Placebo	24	26	18

29.2.2 Manipulating the Table's presentation

We can add margins to the matrix to get a table including row and column totals.

```
addmargins(T1)
```

	None	Partial	Complete	Sum
Active	16	26	29	71
Placebo	24	26	18	68
Sum	40	52	47	139

Instead of the counts, we can tabulate the proportion of all patients within each cell.

```
prop.table(T1)
```

	None	Partial	Complete
Active	0.115	0.187	0.209
Placebo	0.173	0.187	0.129

Or, we can tabulate the probabilities, rather than the proportions, after some rounding.

```
round(100*prop.table(T1),1)
```

	None	Partial	Complete
Active	11.5	18.7	20.9
Placebo	17.3	18.7	12.9

R can also plot the percentages conditional on the rows...

```
round(100*prop.table(T1,1),1)
```

	None	Partial	Complete
--	------	---------	----------

```
Active  22.5    36.6    40.8
Placebo 35.3    38.2    26.5
```

The 40.8 in Active/Complete means that of the Active medication patients, 40.8% had a complete response.

R can also plot the percentages conditional on the columns...

```
round(100*prop.table(T1,2),1)
```

	None	Partial	Complete
Active	40	50	61.7
Placebo	60	50	38.3

If we add the row of column totals for these percentages as shown below, it clarifies that the 61.7 in Active/Complete here means that of the patients with a Complete response, 61.7% were on the active medication.

```
addmargins(round(100*prop.table(T1,2),1),1)
```

	None	Partial	Complete
Active	40	50	61.7
Placebo	60	50	38.3
Sum	100	100	100.0

29.3 Getting the Chi-Square Test Results

Now, to actually obtain a p value and perform the significance test with H_0 : rows and columns are independent vs. H_A : rows and columns are associated, we simply run a Pearson chi-square test on T1 ...

```
chisq.test(T1)
```

```
Pearson's Chi-squared test

data: T1
X-squared = 4, df = 2, p-value = 0.1
```

Thanks to a p -value of about 0.13 (using the Pearson χ^2 test) our conclusion would be to retain the null hypothesis of independence in this setting.

We could have run a Fisher's exact test, too, if we needed it.

```
fisher.test(T1)
```

```
Fisher's Exact Test for Count Data

data: T1
```

```
p-value = 0.1
alternative hypothesis: two.sided
rm(T1)
```

The Fisher exact test p value is also 0.13. Either way, there is insufficient evidence to conclude that there is a (true) difference in the distributions of responses.

29.4 Getting a 2x3 Table into R using a .csv file

Suppose we have a table like this available, and want to compute the Pearson and Fisher χ^2 test results in R, without having to set up the whole matrix piece discussed above?

Group	None	Partial	Complete
Active	16	26	29
Placebo	24	26	18

We could do so by building a .csv file (a spreadsheet) containing the table above. In particular, I built a file called `active2x3.csv`, available on the course website. It is simply the table below.

Group	None	Partial	Complete
Active	16	26	29
Placebo	24	26	18

When we pull this .csv file into R, it emerges as a data frame and deliberately NOT as a tibble.

```
active2x3
```

	Group	None	Partial	Complete
1	Active	16	26	29
2	Placebo	24	26	18

29.5 Turning the Data Frame into a Table That R Recognizes

We need to turn this data frame into something that R can recognize as a contingency table. The steps to do this are:

1. Establish row names from the Group variable.
2. Delete the Group variable, retaining only the variables containing counts.
3. Convert the data frame into a matrix.

Specifically in this case, the steps are:

```
rownames(active2x3) <- active2x3$Group
active2x3$Group <- NULL
tab2x3 <- as.matrix(active2x3)
tab2x3
```

	None	Partial	Complete
Active	16	26	29
Placebo	24	26	18

And this resulting `tab2x3` object can be treated as the matrix was previously, using `addmargins` or `chisq.test` or `prop.table` or whatever. For instance,

```
fisher.test(tab2x3)
```

```
Fisher's Exact Test for Count Data

data: tab2x3
p-value = 0.1
alternative hypothesis: two.sided
```

29.6 Collapsing Levels / Categories in a Factor

Returning to the `survey1` data, let's build a table of the relationship between response to the `sex` and favorite `color` questions.

```
table(survey1$sex, survey1$color)
```

	aqua	aquamarine	black	blue	brown	chartreuse	dark blue	depends	gray
f	2	0	1	21	0	0	2	0	0
m	0	1	2	41	3	1	0	1	1

	green	grey	light blue	navy	none	orange	pink	purple	red	royal blue
f	11	1	1	0	1	0	4	10	5	1
m	12	2	0	2	0	3	1	1	8	0

	sapphire	silver	teal	violet	white	yellow
f	1	1	1	1	3	3
m	0	0	0	0	2	1


```
chisq.test(table(survey1$sex, survey1$color))
```

Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be incorrect

Pearson's Chi-squared test

```
data: table(survey1$sex, survey1$color)
X-squared = 41, df = 24, p-value = 0.02
```

Note the warning message. With all those small cell counts, and particular, so many zeros, this might not be so useful.

We might reasonably consider collapsing the `colors` data into four new categories:

- Blue – in which I will include the old aqua, aquamarine, blue, dark blue and sapphire;
- Green, which will include the old chartreuse and green,
- Red - which will include the old orange, pink and red, and
- Other - which will include all other colors, including purple and violet

One way to do this sort of work uses the `plyr` library in R, and the `revalue` function in particular. I learned about this in Chapter 15 of Chung's R Graphics Cookbook, which is invaluable.

```
survey1$color.new <- factor(survey1$color)
levels(survey1$color.new)

[1] "aqua"      "aquamarine" "black"       "blue"        "brown"
[6] "chartreuse" "dark blue"   "depends"     "gray"        "green"
[11] "grey"       "light blue"  "navy"        "none"        "orange"
[16] "pink"       "purple"     "red"         "royal blue"  "sapphire"
[21] "silver"     "teal"       "violet"      "white"       "yellow"

survey1$color.new <- dplyr::recode(survey1$color.new,
  aqua = "Blue", aquamarine="Blue", black = "Other",
  blue = "Blue", brown = "Other", chartreuse = "Green",
  'dark blue' = "Blue", depends = "Other", gray = "Other",
  green = "Green", grey = "Other", 'light blue' = "Blue",
  navy = "Blue", none = "Other", orange = "Red",
  pink = "Red", purple = "Other", red = "Red",
  'royal blue' = "Blue", sapphire = "Blue", silver = "Other",
  teal = "Blue", violet = "Other", white = "Other", yellow = "Other")
```

So, what I've done here is create a new color variable that assigns the original colors into the four categories: Blue, Green, Red and Other that I defined above. Let's run a sanity check on our recoding, then look at the relationship between sex and this new four-category variable, in a 2x4 table...

```
table(survey1$color, survey1$color.new) ## sanity check
```

	Blue	Other	Green	Red
aqua	2	0	0	0

aquamarine	1	0	0	0
black	0	3	0	0
blue	62	0	0	0
brown	0	3	0	0
chartreuse	0	0	1	0
dark blue	2	0	0	0
depends	0	1	0	0
gray	0	1	0	0
green	0	0	23	0
grey	0	3	0	0
light blue	1	0	0	0
navy	2	0	0	0
none	0	1	0	0
orange	0	0	0	3
pink	0	0	0	5
purple	0	11	0	0
red	0	0	0	13
royal blue	1	0	0	0
sapphire	1	0	0	0
silver	0	1	0	0
teal	1	0	0	0
violet	0	1	0	0
white	0	5	0	0
yellow	0	4	0	0

```
table(survey1$sex, survey1$color.new)
```

	Blue	Other	Green	Red
f	29	21	11	9
m	44	13	13	12

To neaten this output up, we might want to have Other show up last in the color.new group.

```
survey1$color.new2 <- factor(survey1$color.new,
  levels=c("Blue", "Green", "Red", "Other"))
table(survey1$sex, survey1$color.new2)
```

	Blue	Green	Red	Other
f	29	11	9	21
m	44	13	12	13

Now, we run the new Pearson χ^2 test, and conclude that there is no evidence of statistically significant association (at the 5% significance level) between sex and these collapsed favorite color categories.

```
chisq.test(table(survey1$sex, survey1$color.new2))
```

Pearson's Chi-squared test

```
data: table(survey1$sex, survey1$color.new2)
X-squared = 5, df = 3, p-value = 0.2
```

29.7 Accuracy of Death Certificates (A 6x3 Table)

The table below compiles data from six studies designed to investigate the accuracy of death certificates¹. 5373 autopsies were compared to the causes of death listed on the certificates. Of those, 3726 were confirmed to be accurate, 783 either lacked information or contained inaccuracies but did not require recoding of the underlying cause of death, and 864 were incorrect and required recoding. Do the results across studies appear consistent?

Date of Study	[Confirmed] Accurate	[Inaccurate] No Change	[Incorrect] Recoding	Total
1955-1965	2040	367	327	2734
1970	149	60	48	257
1970-1971	288	25	70	383
1975-1977	703	197	252	1152
1977-1978	425	62	88	575
1980	121	72	79	272
Total	3726	783	864	5373

29.8 The Pearson Chi-Square Test of Independence

We can assess the homogeneity of the confirmation results (columns) we observe in the table using a Pearson chi-squared test of independence.

- The null hypothesis is that the rows and columns are independent.
- The alternative hypothesis is that there is an association between the rows and the columns.

¹Source: Pagano M, Gauvreau K (2000) Principles of Biostatistics, 2nd Edition, Pacific Grove, CA: Duxbury, pp. 367-8. The original citation is Kircher T, Nelson J, Burdo H (1985) The autopsy as a measure of accuracy of the death certificate. NEJM, 313, 1263-1269.

```

z <- matrix(c(2040,367,327,149,60,48,288,25,70,703,
             197,252,425,62,88,121,72,79), byrow=TRUE, nrow=6)
rownames(z) <- c("1955-65", "1970", "1970-71", "1975-77", "1977-78",
                  "1980")
colnames(z) <- c("Confirmed", "Inaccurate", "Incorrect")

addmargins(z)

```

	Confirmed	Inaccurate	Incorrect	Sum
1955-65	2040	367	327	2734
1970	149	60	48	257
1970-71	288	25	70	383
1975-77	703	197	252	1152
1977-78	425	62	88	575
1980	121	72	79	272
Sum	3726	783	864	5373

To see the potential heterogeneity across rows in these data, we should perhaps also look at the proportions of autopsies in each of the three accuracy categories for each study.

```
addmargins(round(100*prop.table(z),1),2)
```

	Confirmed	Inaccurate	Incorrect	Sum
1955-65	74.6	13.4	12.0	100
1970	58.0	23.3	18.7	100
1970-71	75.2	6.5	18.3	100
1975-77	61.0	17.1	21.9	100
1977-78	73.9	10.8	15.3	100
1980	44.5	26.5	29.0	100

In three of the studies, approximately 3/4 of the results were confirmed. In the other three, 45%, 58% and 61% were confirmed. It looks like there's a fair amount of variation in results across studies. To see if this is true, formally, we run Pearson's chi-square test of independence, where the null hypothesis is that the rows and columns are independent, and the alternative hypothesis is that there is an association between the rows and the columns.

```
chisq.test(z)
```

Pearson's Chi-squared test

```

data: z
X-squared = 209, df = 10, p-value <2e-16

```

The chi-square test statistic is 200 on 10 degrees of freedom, yielding $p < 0.0001$.

Autopsies are not performed at random; in fact, many are done because the cause of death listed on the certificate is uncertain. What problems may arise if you attempt to use the results of these studies to make inference about the population as a whole?

29.9 Three-Way Tables: A 2x2xK Table and a Mantel-Haenszel Analysis

The material I discuss in this section is attributable to Jeff Simonoff and his book *Analyzing Categorical Data*. The example is taken from Section 8.1 of that book.

A three-dimensional or three-way table of counts often reflects a situation where the rows and columns refer to variables whose association is of primary interest to us, and the third factor (a layer, or strata) describes a control variable, whose effect on our primary association is something we are *controlling* for in the analysis.

29.9.1 Smoking and Mortality in the UK

In the early 1970s and then again 20 years later, in Whickham, United Kingdom, surveys yielded the following relationship between whether a person was a smoker at the time of the original survey and whether they were still alive 20 years later².

```
whickham1 <- matrix(c(443, 139, 502, 230), byrow=TRUE, nrow=2)
rownames(whickham1) <- c("Smoker", "Non-Smoker")
colnames(whickham1) <- c("Alive", "Dead")
addmargins(whickham1)
```

	Alive	Dead	Sum
Smoker	443	139	582
Non-Smoker	502	230	732
Sum	945	369	1314

Here's the two-by-two table analysis.

```
Epi::twoby2(whickham1)
```

```
2 by 2 table analysis:
```

```
Outcome   : Alive
Comparing : Smoker vs. Non-Smoker
```

²See Appleton et al. 1996. Ignoring a Covariate: An Example of Simpson's Paradox. The American Statistician, 50, 340-341.

	Alive	Dead	P(Alive)	95% conf. interval
Smoker	443	139	0.761	0.725 0.794
Non-Smoker	502	230	0.686	0.651 0.718
				95% conf. interval
Relative Risk:	1.1099		1.0381	1.187
Sample Odds Ratio:	1.4602		1.1414	1.868
Conditional MLE Odds Ratio:	1.4598		1.1335	1.884
Probability difference:	0.0754		0.0266	0.123
				Exact P-value: 0.0030
				Asymptotic P-value: 0.0026

```
chisq.test(whickham1)
```

Pearson's Chi-squared test with Yates' continuity correction

```
data: whickham1
X-squared = 9, df = 1, p-value = 0.003
```

There is a significant association between smoking and mortality ($\chi^2 = 8.75$ on 1 df, $p = 0.003$), but it isn't the one you might expect.

- The odds ratio is 1.46, implying that the odds of having lived were 46% higher for smokers than for non-smokers.
- Does that mean that smoking is *good* for you?

Not likely. There is a key “lurking” variable here - a variable that is related to both smoking and mortality that is obscuring the actual relationship - namely, age.

29.9.2 The `whickham` data including age, as well as smoking and mortality

The table below gives the mortality experience separated into subtables by initial age group.

```
age <- c(rep("18-24", 4), rep("25-34", 4),
       rep("35-44", 4), rep("45-54", 4),
       rep("55-64", 4), rep("65-74", 4),
       rep("75+", 4))
smoking <- c(rep(c("Smoker", "Smoker", "Non-Smoker", "Non-Smoker"), 7))
status <- c(rep(c("Alive", "Dead"), 14))
counts <- c(53, 2, 61, 1, 121, 3, 152, 5,
```

```

95, 14, 114, 7, 103, 27, 66, 12,
64, 51, 81, 40, 7, 29, 28, 101,
0, 13, 0, 64)
whickham2 <- data.frame(smoking, status, age, counts) %>%tbl_df()
whickham2$smoking <- factor(whickham2$smoking, levels = c("Smoker", "Non-Smoker"))
whickham2.tab1 <- xtabs(counts ~ smoking + status + age, data = whickham2)
whickham2.tab1

, , age = 18-24

      status
smoking   Alive  Dead
  Smoker      53    2
Non-Smoker    61    1

, , age = 25-34

      status
smoking   Alive  Dead
  Smoker     121    3
Non-Smoker    152    5

, , age = 35-44

      status
smoking   Alive  Dead
  Smoker      95   14
Non-Smoker    114    7

, , age = 45-54

      status
smoking   Alive  Dead
  Smoker     103   27
Non-Smoker     66   12

, , age = 55-64

      status
smoking   Alive  Dead
  Smoker      64   51
Non-Smoker    81   40

, , age = 65-74

      status

```

29.9. THREE-WAY TABLES: A 2X2XK TABLE AND A MANTEL-HAENSZEL ANALYSIS 463

smoking	Alive	Dead
Smoker	7	29
Non-Smoker	28	101

, , age = 75+

smoking	Alive	Dead
Smoker	0	13
Non-Smoker	0	64

The odds ratios for each of these subtables, except the last one, where it is undefined are as follows:

Age Group	Odds Ratio
18-24	0.43
25-34	1.33
35-44	0.42
45-54	0.69
55-64	0.62
65-74	0.87
75+	undefined

Thus, for all age groups except 25-34 year olds, smoking is associated with higher mortality.

Why? Not surprisingly, there is a strong association between age and mortality, with mortality rates being very low for young people (2.5% for 18-24 year olds) and increasing to 100% for 75+ year olds.

There is also an association between age and smoking, with smoking rates peaking in the 45-54 year old range and then falling off rapidly. In particular, respondents who were 65 and older at the time of the first survey had very low smoking rates (25.4%) but very high mortality rates (85.5%). Smoking was hardly the cause, however, since even among the 65-74 year olds mortality was higher among smokers (80.6%) than it was among non-smokers (78.3%). A flat version of the table (`ftable` in R) can help us with these calculations.

```
ftable(whickham2.tab1)
```

smoking	status	age	18-24	25-34	35-44	45-54	55-64	65-74	75+
Smoker	Alive		53	121	95	103	64	7	0
Non-Smoker	Alive		61	152	114	66	81	28	0
Smoker	Dead		2	3	14	27	51	29	13
Non-Smoker	Dead		1	5	7	12	40	101	64

29.9.3 The Cochran-Mantel-Haenszel Test

So, the marginal table looking at smoking and mortality combining all age groups isn't the most meaningful summary of the relationship between smoking and mortality. Instead, we need to look at the *conditional* association of smoking and mortality, **given age**, to address our interests.

The null hypothesis would be that, in the population, smoking and mortality are independent within strata formed by age group. In other words, H_0 requires that smoking be of no value in predicting mortality once age has been accounted for.

The alternative hypothesis would be that, in the population, smoking and mortality are associated within the strata formed by age group. In other words, H_A requires that smoking be of at least some value in predicting mortality even after age has been accounted for.

We can consider the evidence that helps us choose between these two hypotheses with a Cochran-Mantel-Haenszel test, which is obtained in R through the `mantelhaen.test` function. This test requires us to assume that, in the population and within each age group, the smoking-mortality odds ratio is the same. Essentially, this means that the association of smoking with mortality is the same for older and younger people.

```
mantelhaen.test(whickham2.tab1, conf.level = 0.90)
```

```
Mantel-Haenszel chi-squared test with continuity correction

data: whickham2.tab1
Mantel-Haenszel X-squared = 5, df = 1, p-value = 0.02
alternative hypothesis: true common odds ratio is not equal to 1
90 percent confidence interval:
0.490 0.875
sample estimates:
common odds ratio
0.655



- The Cochran-Mantel-Haenszel test statistic is 5 (after a continuity correction) leading to a  $p$  value of 0.02, indicating strong rejection of the null hypothesis of conditional independence of smoking and survival given age.
- The estimated common conditional odds ratio is 0.65. This implies that (given age) being a smoker is associated with a 35% lower odds of being alive 20 years later than being a non-smoker.
- A 90% confidence interval for that common odds ratio is (0.49, 0.87), reinforcing rejection of the conditional independence (where the odds ratio would be 1).

```

29.9.4 Checking Assumptions: The Woolf test

We can also obtain a test (using the `woolf_test` function, in the `vcd` library) to see if the common odds ratio estimated in the Mantel-Haenszel procedure is reasonable for all age groups. In other words, the Woolf test is a test of the assumption of homogeneous odds ratios across the six age groups.

If the Woolf test is significant, it suggests that the Cochran-Mantel-Haenszel test is not appropriate, since the odds ratios for smoking and mortality vary too much in the sub-tables by age group. Here, we have the following log odds ratios (estimated using conditional maximum likelihood, rather than cross-product ratios) and the associated Woolf test.

```
## Next two results use the vcd library

vcd::oddsratio(whickham2.tab1, log = TRUE)

log odds ratios for smoking and status by age

 18-24   25-34   35-44   45-54   55-64   65-74     75+
-0.6502  0.2247 -0.8407 -0.3461 -0.4742 -0.0993  1.5640

vcd::woolf_test(whickham2.tab1)

Woolf-test on Homogeneity of Odds Ratios (no 3-Way assoc.)

data: whickham2.tab1
X-squared = 3, df = 6, p-value = 0.8
```

As you can see, the Woolf test is not close to statistically significant, implying the common odds ratio is at least potentially reasonable for all age groups (or at least the ones under ages 75, where some data are available.)

29.9.5 Without the Continuity Correction

By default, R presents the Mantel-Haenszel test with a continuity correction, when used for a 2x2xK table. In virtually all cases, go ahead and do this, but as you can see below, the difference it makes in this case is modest.

```
mantelhaen.test(whickham2.tab1, correct=FALSE, conf.level = 0.90)

Mantel-Haenszel chi-squared test without continuity correction

data: whickham2.tab1
Mantel-Haenszel X-squared = 6, df = 1, p-value = 0.02
alternative hypothesis: true common odds ratio is not equal to 1
```

90 percent confidence interval:

0.490 0.875

sample estimates:

common odds ratio

0.655

Chapter 30

Partial Review to help you prepare for Quiz 2

There are 15 items here, and then a set of answer sketches follow the questions. This **isn't a complete review** - there are no questions here about some topics that could show up on Quiz 2.

30.1 Review Items 1-7

Researchers comparing the effectiveness of two pain medications randomly selected a group of patients who had been complaining of a certain kind of joint pain. They randomly divided those people into two groups, then administered the medications. Of the 85 people in the group who received medication A, 65 said that it provided relief. Of the 70 people in the group receiving medication B, 45 reported that it provided relief.

1. Use the single augmentation with an imaginary failure or success (SAIFS) approach to specify a 95% confidence interval for the proportion of people who find relief from this kind of joint pain by using medication A.
2. Now use the same approach to specify a 95% confidence interval for the proportion of people who find relief using medication B.
3. Do the confidence intervals in items 1 and 2 overlap? What conclusions can you draw in light of that overlap (or lack thereof) about whether medication A or medication B is significantly more effective?
4. Specify and display the correct 2x2 table (incorporating a Bayesian augmentation) analysis to enable you to study the A - B difference in the true proportions of people who find these medications effective.
5. Use the 2x2 table results to specify an appropriate odds ratio and its 95% confidence interval in this situation, and explain what the values mean in

context.

6. Specify the hypotheses (H_0 and H_A) tested by the Fisher exact test you obtain in your 2x2 table. What does the provided p value tell you about what conclusion you should draw in this case regarding those hypotheses?
7. If you have made an error in your conclusion for item 6, was it a Type I error or a Type II error? How do you know?

30.2 Review Items 8-9

For each of the following statements, indicate whether or not the statement is true or false, and specify how you know.

8. If there is sufficient evidence to reject a null hypothesis at the 10% level, then there is sufficient evidence to reject it at the 5% level.
9. A sample histogram will follow a normal distribution if the sample size is large enough.

30.3 Review Items 10-13

Charles Darwin carried out an experiment to study whether seedlings from cross-fertilized plants tend to be superior to those from self-fertilized plants. He covered a number of plants with fine netting so that insects would be unable to fertilize them. He fertilized a number of flowers on each plant with their own pollen and he fertilized an equal number of flowers on the same plant with pollen from a distant plant. (He did not specify how he decided which flowers received which treatments.) The seeds from the flowers were allowed to ripen and were set in wet sand to germinate. He placed two seedlings of the same age in a pot, one from a seed from a self-fertilized flower and one from a cross-fertilized flower.

He repeated this process with a total of 15 such pots. Each pot was then set aside for a time, so that the two plants in the plot would receive similar exposure to atmospheric conditions (sun, rainfall, etc.). Later, he gathered the heights of the plants (in inches) that came from those 15 cross-fertilized and 15 self-fertilized seeds at certain points in time. Those data are contained in the `darwin.csv` data set on our course website.

10. Does this study call for a paired samples or independent samples comparison? How do you know?
11. Display and interpret an appropriate graph to determine whether a t-test or a Wilcoxon test would be more appropriate for these data.
12. Use the method (t or Wilcoxon) you specified in item 11 to find an appropriate 95% confidence interval for the average height difference between cross-fertilized and self-fertilized seedlings. Verify that your confidence

- interval describes the “cross” - “self” difference, rather than the opposite direction.
13. Use an appropriate bootstrap procedure (setting your random seed to be 4310) to provide an alternative answer for the question posed in item 12. Is this bootstrap confidence interval wider or narrower than the interval you produced in item 12?

30.4 Review Items 14-15

You have been asked how large a sample size will be required for a clinical trial comparing two different approaches to blood pressure control. In approach A, we believe that the average systolic blood pressure will drop by 7 mm Hg, on the basis of our prior work in this area, while in the new approach B, we hope to see a clinically meaningful additional decline - specifically, we are looking for at least a 50% larger decline, so that the average systolic blood pressure will drop by 10.5 mm Hg or more over the same amount of time. Thus, the minimum clinically meaningful difference we are looking for is 3.5 mm Hg. Suppose we believe that the relevant standard deviation is 9 mm Hg, and we want to complete the trial using a 5% significance level and a two-sided t test.

14. What will be the power of the test if we have a balanced design with 120 subjects in approach A and 120 different subjects in approach B? Show your calculation, and state your final result in a sentence.
15. What is the smallest total sample size that we can use in a balanced design to maintain at least 90% power to detect the difference of interest, while still using independent samples? Show your calculation, and state your final result in a sentence.

30.5 Answer Sketch for Review Items

30.5.1 Answer 1

`saifs.ci(65, 85)`

Sample Proportion	0.025	0.975
	0.765	0.859

The 95% confidence interval for the proportion of people using medication A who obtain relief is (0.663, 0.859). We are 95% confident that the true percentage of people who find relief using medication A is between 66.3% and 85.9%.

470CHAPTER 30. PARTIAL REVIEW TO HELP YOU PREPARE FOR QUIZ 2

30.5.2 Answer 2

```
saifs.ci(45, 70)
```

Sample Proportion	0.025	0.975
0.643	0.519	0.762

The 95% confidence interval for the proportion of people using medication B who obtain relief is (0.519, 0.762).

30.5.3 Answer 3

The confidence intervals do overlap, so we cannot conclude from the separate intervals that there is (or isn't) a statistically significant difference in the effectiveness rates for medications A and B. If the confidence intervals didn't overlap, then we would know that there was a statistically significant difference in effectiveness between the two medications.

30.5.4 Answer 4

```
twobytwo(65+1, 20+1, 45+1, 25+1, "Med. A", "Med. B", "Relief", "No Relief")
```

2 by 2 table analysis:

```
-----  
Outcome : Relief  
Comparing : Med. A vs. Med. B
```

	Relief	No Relief	P(Relief)	95% conf. interval
Med. A	66	21	0.759	0.658 0.837
Med. B	46	26	0.639	0.522 0.741

	95% conf. interval		
Relative Risk:	1.19	0.9623	1.465
Sample Odds Ratio:	1.78	0.8934	3.532
Conditional MLE Odds Ratio:	1.77	0.8446	3.749
Probability difference:	0.12	-0.0223	0.259

```
-----  
Exact P-value: 0.1174  
Asymptotic P-value: 0.1013
```

30.5.5 Answer 5

The odds ratio is 1.78, with 95% confidence interval (0.89, 3.53). The point estimate states that the odds of finding relief with medication A are 78% higher than the odds of finding relief with medication B. But the confidence interval indicates that, with 95% confidence, we can conclude only that the odds of relief with medication A are between 0.89 and 3.53 times as high as the odds of relief with medication B. Since 1 is in that confidence interval, we must conclude that there is no statistically significant effect of the medication choice on the odds of this outcome.

30.5.6 Answer 6

- H_0 : Medication Choice (A or B) is unrelated to the probability of Relief
- H_A : Medication Choice and Relief are associated

The p value is 0.12, from the Fisher exact test. This means that we must retain the null hypothesis, and conclude that there is no significant association between Medication choice and the probability of Relief.

30.5.7 Answer 7

You would have made a Type II error. A Type II error can be made if you incorrectly retain H_0 . Since we retain H_0 , if we've made an error, it must have been a Type II error, since a Type I error occurs when you incorrectly reject H_0 .

30.5.8 Answer 8

This is FALSE. Sufficient evidence to reject H_0 at the 10% level means that we have a p value < 0.10 . In order to have sufficient evidence to reject H_0 at the 5% level, we'd need to have a p value < 0.05 . If our $p < 0.10$, this doesn't guarantee that it is also true that $p < 0.05$.

30.5.9 Answer 9

Also FALSE. The mean of a sample will approach a Normal distribution, but if the data are skewed, the data will still be skewed no matter how many observations we see.

30.5.10 Answer 10

These samples are paired by the pot. Each pot provides a cross-fertilized seedling height and a self-fertilized seedling height. We should be comparing paired differences.

30.5.11 Answer 11

We need a plot of the 15 paired differences, for example a boxplot, or a normal Q-Q plot.

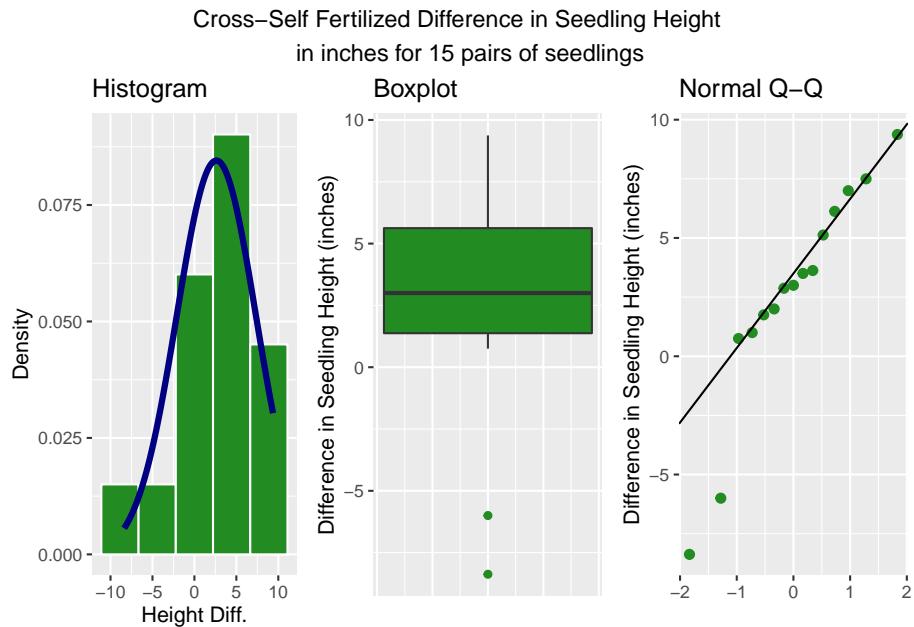
```
darwin$diffs <- darwin$cross.fertilized - darwin$self.fertilized

p1 <- ggplot(darwin, aes(x = diffs)) +
  geom_histogram(aes(y = ..density..), bins = fd_bins(darwin$diffs),
                 fill = "forestgreen", col = "white") +
  stat_function(fun = dnorm,
                args = list(mean = mean(darwin$diffs),
                            sd = sd(darwin$diffs)),
                lwd = 1.5, col = "navy") +
  labs(title = "Histogram",
       x = "Height Diff.", y = "Density")

p2 <- ggplot(darwin, aes(x = 1, y = diffs)) +
  geom_boxplot(fill = "forestgreen", outlier.color = "forestgreen") +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  labs(title = "Boxplot",
       y = "Difference in Seedling Height (inches)", x = "")

p3 <- ggplot(darwin, aes(sample = diffs)) +
  geom_qq(col = "forestgreen", size = 2) +
  geom_abline(intercept = qq_int(darwin$diffs),
              slope = qq_slope(darwin$diffs)) +
  labs(title = "Normal Q-Q",
       y = "Difference in Seedling Height (inches)", x = "")

gridExtra::grid.arrange(p1, p2, p3, nrow=1,
                       top = "Cross-Self Fertilized Difference in Seedling Height
in inches for 15 pairs of seedlings")
```



It appears that we have two low outliers out of the 15 paired differences. Assuming normality seems inappropriate here. I would probably use a Wilcoxon approach instead.

30.5.12 Answer 12

```
wilcox.test(darwin$diffs, conf.int=TRUE, conf.level=0.95)
```

```
Wilcoxon signed rank test

data: darwin$diffs
V = 96, p-value = 0.04
alternative hypothesis: true location is not equal to 0
95 percent confidence interval:
 0.50 5.19
sample estimates:
(pseudo)median
            3.12
```

The cross-self differences appear to have a population pseudomedian which we are 95% confident is between 0.5 and 5.2 inches. The cross-fertilized plants appear to be statistically significantly taller on average than the self-fertilized plants in the same pot.

30.5.13 Answer 13

```
set.seed(4310); Hmisc::smean.cl.boot(darwin$diffs)
```

Mean	Lower	Upper
2.617	0.316	4.759

This confidence interval is a bit narrower than the interval in item 12, and also shifted a bit closer to zero. We are 95% confident that the population mean cross-self difference is between 0.2 and 4.7 inches.

30.5.14 Answer 14

```
power.t.test(n = 120, delta = 3.5, sd = 9, sig.level = 0.05)
```

Two-sample t test power calculation

```
n = 120
delta = 3.5
sd = 9
sig.level = 0.05
power = 0.851
alternative = two.sided
```

NOTE: n is number in *each* group

Such a test will have just over 85% power to detect the specified minimum clinically meaningful difference of 3.5 mm Hg, using a 5% two-sided significance level.

30.5.15 Answer 15

```
power.t.test(power=0.9, delta = 3.5, sd = 9, sig.level = 0.05)
```

Two-sample t test power calculation

```
n = 140
delta = 3.5
sd = 9
sig.level = 0.05
power = 0.9
alternative = two.sided
```

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

The minimum sample size we'll need is 140 subjects in each approach (A and B), so that's a total sample size of 280, to achieve 90% or higher power for the specified test while still using independent samples.

Part C. Building Models

Chapter 31

Introduction for Part C

In 431, my primary goal is to immerse you in several cases, which will demonstrate good statistical practice in the analysis of data using multiple regression models. Often, we will leave gaps for 432, but the principal goal is to get you to the point where you can do a solid (if not quite complete) analysis of data for the modeling part of your project.

The ten main topics to be discussed or reviewed in these notes are:

1. Describing the multivariate relationship
 - a. Scatterplots and smoothing
 - b. Correlation coefficients, Correlation matrices
2. Transformations and Re-expression
 - a. The need for transformation
 - b. Using a Box-Cox method to help identify effective transformation choices
3. Testing the significance of a multiple regression model
 - a. T tests for individual predictors as last predictor in
 - b. Global F tests based on ANOVA to assess overall predictive significance
 - c. Incremental and Sequential testing of groups of predictors
4. Interpreting the predictive value of a model
 - a. R^2 and Adjusted R^2 , along with AIC and BIC
 - b. Residual standard deviation and RMSE
 - c. Estimating the effect size in terms of raw units, standard deviations or IQRs
 - d. Fitted values; Distinguishing prediction from confidence intervals
5. Checking model assumptions
 - a. Residual Analysis including studentized residuals, and the major plots
 - b. Identifying points with high Leverage
 - c. Assessing Influence numerically and graphically

- d. Measuring and addressing collinearity
- 6. Model Selection
 - a. The importance of parsimony
 - b. Stepwise regression and other automated techniques
- 7. Assessing Predictive Accuracy through Cross-Validation
 - a. Summaries of predictive error
- 8. Dealing with Missing Values sensibly
 - a. Imputation vs. Complete Case analyses
 - b. Including a missing data category vs. simple imputation vs. removal
- 9. Dealing with Categorical Predictors
 - a. Indicator variables
 - b. Impact of Categorical Variables on the rest of our Modeling
- 10. Summarizing the Key Findings of the Model, briefly and accurately
 - a. Making the distinction between causal findings and associations
 - b. The importance of logic, theory and empirical evidence. (LTE)

31.1 Additional Reading

Vittinghoff et al. (2012) is strong in this area. The relevant sections of the text for 431 Part C are

- Section 3.3 on the Simple Linear Regression Model
- Chapter 4 on Linear Regression, where most of the material is relevant to 431, although we'll postpone the discussion of cubic splines, mostly, to 432.
- Chapters 10 (Model Selection) in particular the alternatives to R^2 in 10.1.3.2 and some of the material on cross-validation, though we'll do much more in 432.
- A little of Chapter 11 (Missing Data), specifically, section 11.1.1 and a little of section 11.3, although we'll do more on this in 432 as well.

31.2 Scatterplots

We have often accompanied our scatterplots with regression lines estimated by the method of least squares, and by loess smooths which permit local polynomial functions to display curved relationships, and occasionally presented in the form of a scatterplot matrix to enable simultaneous comparisons of multiple two-way associations.

31.3 Correlation Coefficients

By far the most commonly used is the Pearson correlation, which is a unitless (scale-free) measure of bivariate linear association for the variables X and Y, symbolized by r , and ranging from -1 to +1. The Pearson correlation is a function of the slope of the least squares regression line, divided by the product of the standard deviations of X and Y.

We have also mentioned the Spearman rank correlation coefficient, which is obtained by using the usual formula for a Pearson correlation, but on the ranks (1 = minimum, n = maximum, with average ranks are applied to the ties) of the X and Y values. This approach (running a correlation of the orderings of the data) substantially reduces the effect of outliers. The result still ranges from -1 to +1, with 0 indicating no linear association.

31.4 Fitting a Linear Model

We have fit several styles of linear model to date, including both *simple* regressions, where our outcome Y is modeled as a linear function of a single predictor X, and *multiple* regression models, where more than one predictor is used. Important elements of a regression fit, obtained through the `summary` function for a `lm` object, include

- the estimated coefficients (intercept and slope(s)) of the fitted model, and
- the R^2 or coefficient of determination, which specifies the proportion of variation in our outcome accounted for by the linear model.

31.5 Building Predictions from a Linear Model

We've also used the `predict` function applied to a `lm` object to obtain point and interval estimates for our outcome based on new values of the predictor(s). We've established both *confidence intervals* from such models, which describe the mean result across a population of subjects with the new predictor values, and *prediction intervals* which describe an individual result for a new subject with those same new values. Prediction intervals are much wider than confidence intervals.

31.6 Data Sets for Part C

```
hydrate <- read.csv("data/hydrate.csv") %>% tbl_df  
hers1race <- read.csv("data/hers1race.csv") %>% tbl_df
```

```
wcgs <- read.csv("data/wcgs.csv") %>% tbl_df  
emp_bmi <- read.csv("data/emp_bmi.csv") %>% tbl_df  
gala <- read.csv("data/gala.csv") %>% tbl_df
```

Chapter 32

Re-Expression, Tukey’s Ladder & Box-Cox Plot

32.1 “Linearize” The Association between Quantitative Variables

Confronted with a scatterplot describing a monotone association between two quantitative variables, we may decide the data are not well approximated by a straight line, and thus, that a least squares regression may not be sufficiently useful. In these circumstances, we have at least two options, which are not mutually exclusive:

- a. Let the data be as they may, and summarize the scatterplot using tools like loess curves, polynomial functions, or cubic splines to model the relationship.
- b. Consider re-expressing the data (often we start with re-expressions of the outcome data [the Y variable]) using a transformation so that the transformed data may be modeled effectively using a straight line.

32.2 A New Tool: the Box-Cox Plot

As before, Tukey’s ladder of power transformations can guide our exploration.

Power (λ)	-2	-1	-1/2	0	1/2	1	2
Transformation	$1/y^2$	$1/y$	$1/\sqrt{y}$	$\log y$	\sqrt{y}	y	y^2

The **Box-Cox plot**, from the `boxCox` function in the `car` package, sifts through the ladder of options to suggest a transformation (for Y) to best linearize the outcome-predictor(s) relationship.

32.2.1 A Few Caveats

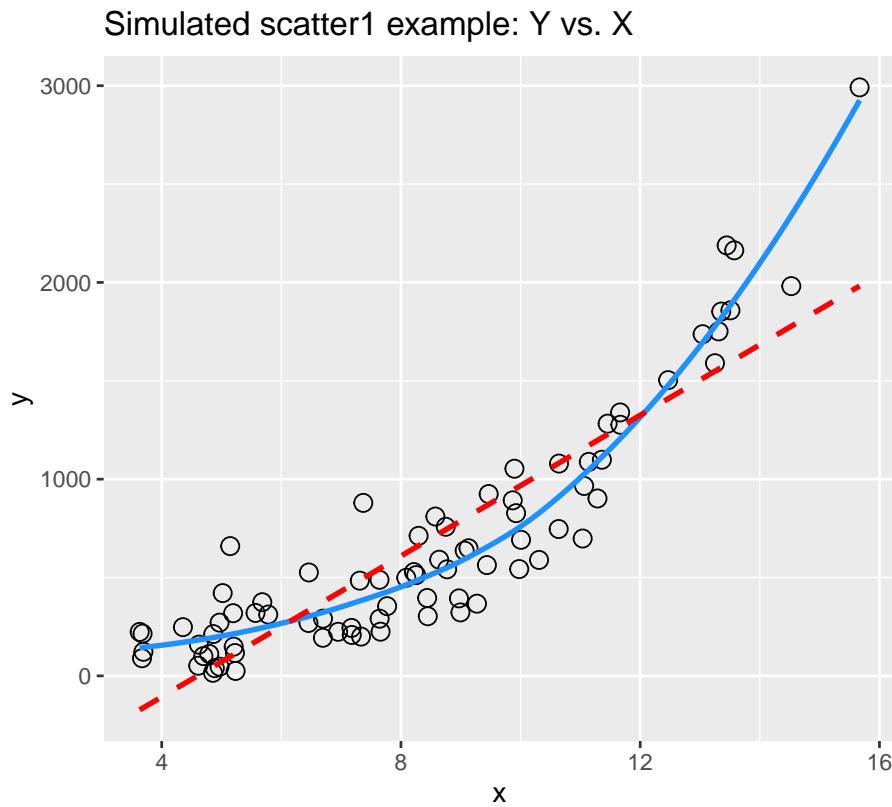
1. These methods work well with *monotone* data, where a smooth function of Y is either strictly increasing, or strictly decreasing, as X increases.
2. Some of these transformations require the data to be positive. We can rescale the Y data by adding a constant to every observation in a data set without changing shape.
3. We can use a natural logarithm (`log` in R), a base 10 logarithm (`log10`) or even sometimes a base 2 logarithm (`log2`) to good effect in Tukey's ladder. All affect the association's shape in the same way, so we'll stick with `log` (base e).
4. Some re-expressions don't lead to easily interpretable results. Not many things that make sense in their original units also make sense in inverse square roots. There are times when we won't care, but often, we will.
5. If our primary interest is in making predictions, we'll generally be more interested in getting good predictions back on the original scale, and we can back-transform the point and interval estimates to accomplish this.

32.3 A Simulated Example

```
set.seed(999); x.rand <- rbeta(80, 2, 5) * 20 + 3
set.seed(1000); y.rand <- abs(50 + 0.75*x.rand^(3) - 0.65*x.rand + rnorm(80, 0, 200))
scatter1 <- data.frame(x = x.rand, y = y.rand) %>% tbl_df
rm(x.rand, y.rand)

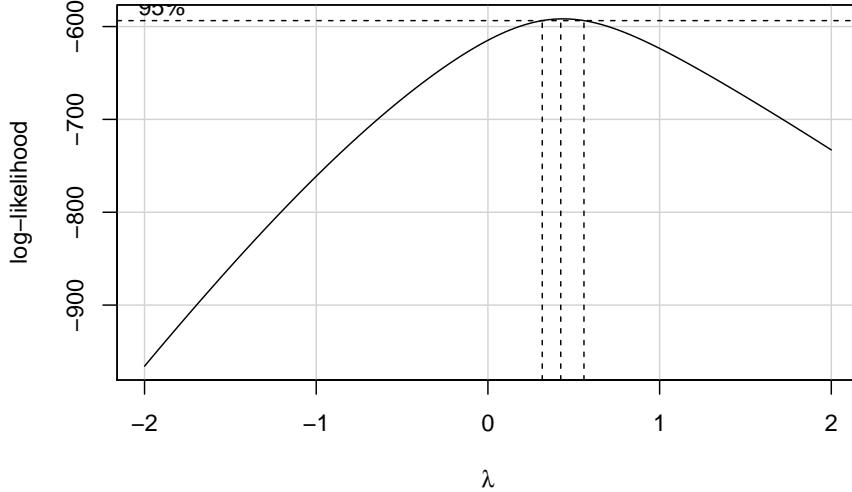
ggplot(scatter1, aes(x = x, y = y)) +
  geom_point(shape = 1, size = 3) +
  ## add loess smooth
  geom_smooth(se = FALSE, col = "dodgerblue") +
  ## then add linear fit
  geom_smooth(method = "lm", se = FALSE, col = "red", linetype = "dashed") +
  labs(title = "Simulated scatter1 example: Y vs. X")

`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Having simulated data that produces a curved scatterplot, I will now use the Box-Cox plot to lead my choice of an appropriate power transformation for Y in order to “linearize” the association of Y and X.

```
library(car)
boxCox(scatter1$y ~ scatter1$x)
```



```
powerTransform(scatter1$y ~ scatter1$x)
```

```
Estimated transformation parameter
Y1
0.437
```

The Box-Cox plot peaks at the value $\lambda = 0.44$, which is pretty close to $\lambda = 0.5$. Now, 0.44 isn't on Tukey's ladder, but 0.5 is.

Power (λ)	-2	-1	-1/2	0	1/2	1	2
Transformation	$1/y^2$	$1/y$	$1/\sqrt{y}$	$\log y$	\sqrt{y}	y	y^2

If we use $\lambda = 0.5$, on Tukey's ladder of power transformations, it suggests we look at the relationship between the square root of Y and X, as shown next.

```
p1 <- ggplot(scatter1, aes(x = x, y = y)) +
  geom_point(size = 2) +
  geom_smooth(se = FALSE, col = "dodgerblue") +
  geom_smooth(method = "lm", se = FALSE, col = "red", linetype = "dashed") +
  labs(title = "scatter1: Y vs. X")

p2 <- ggplot(scatter1, aes(x = x, y = sqrt(y))) +
  geom_point(size = 2) +
  geom_smooth(se = FALSE, col = "dodgerblue") +
  geom_smooth(method = "lm", se = FALSE, col = "red", linetype = "dashed") +
```

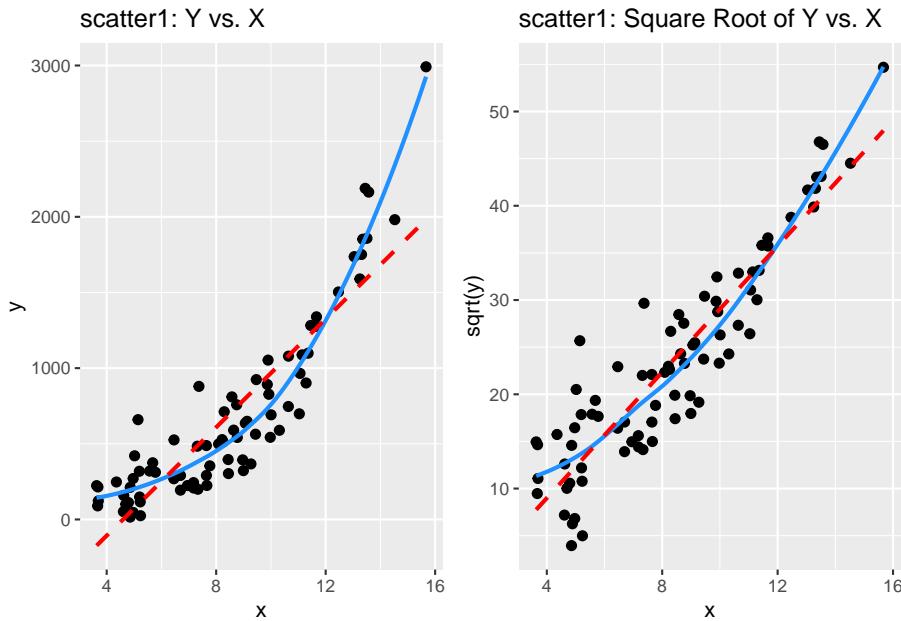
```

  labs(title = "scatter1: Square Root of Y vs. X")

gridExtra::grid.arrange(p1, p2, nrow = 1)

`geom_smooth()` using method = 'loess' and formula 'y ~ x'
`geom_smooth()` using method = 'loess' and formula 'y ~ x'

```



By eye, I think the square root plot better matches the linear fit.

32.4 Checking on a Transformation or Re-Expression

We can do three more things to check on our transformation.

1. We can calculate the correlation of our original and re-expressed associations.
2. We can use the `testTransform` function in the `car` library in R to perform a statistical test comparing the optimal choice of power ($\lambda = 0.44$) to various other transformations.
3. We can go ahead and fit the regression models using each approach and compare the plots of studentized residuals against fitted values from the data to see if the re-expression reduces the curve in that residual plot, as well.

Option 3 is by far the most important in practice, and it's the one we'll focus on going forward, but we'll demonstrate all three here.

32.4.1 Checking the Correlation Coefficients

Here, we calculate the correlation of original and re-expressed associations.

```
cor(scatter1$y, scatter1$x)

[1] 0.891

cor(sqrt(scatter1$y), scatter1$x)

[1] 0.914
```

The Pearson correlation is a little stronger after the transformation. as we'd expect.

32.4.2 Using the `testTransform` function

Here, we use the `testTransform` function (also from the `car` package) to compare the optimal choice determined by the `powerTransform` function (here $\lambda = 0.44$) to $\lambda = 0$ (logarithm), 0.5 (square root) and 1 (no transformation).

```
testTransform(powerTransform(scatter1$y ~ scatter1$x), 0)

      LRT df  pval
LR test, lambda = (0) 46.2  1 1e-11

testTransform(powerTransform(scatter1$y ~ scatter1$x), 0.5)

      LRT df  pval
LR test, lambda = (0.5) 1.02  1  0.3

testTransform(powerTransform(scatter1$y ~ scatter1$x), 1)

      LRT df  pval
LR test, lambda = (1) 63.8  1 1e-15
```

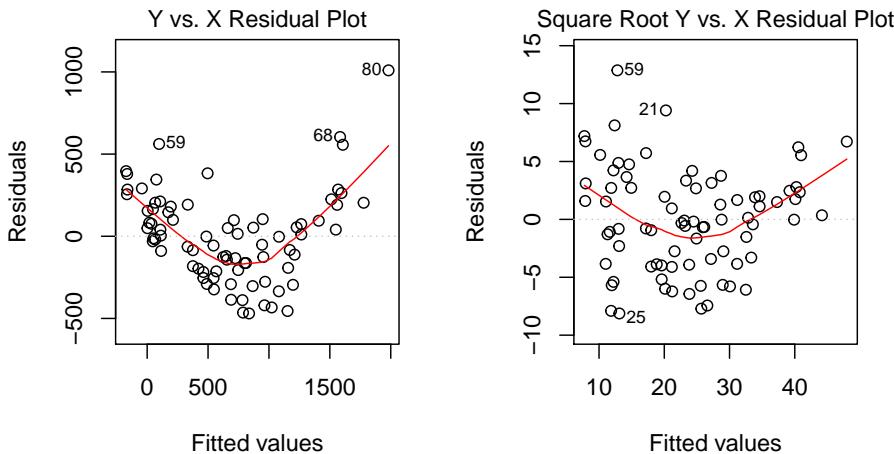
- It looks like only the square root ($\lambda = 0.5$) of these three options is not significantly worse by the log-likelihood criterion applied here than the optimal choice.
- That's because it's the only one with a p value larger than our usual standard for statistical significance, of 0.05.

32.4.3 Comparing the Residual Plots

We can fit the regression models, obtain plots of residuals against fitted values, and compare them to see which one has less indication of a curve in the residuals.

```
model.orig <- lm(scatter1$y ~ scatter1$x)
model.sqrt <- lm(sqrt(scatter1$y) ~ scatter1$x)

par(mfrow=c(1,2))
plot(model.orig, which = 1, caption = "Y vs. X Residual Plot")
plot(model.sqrt, which = 1, caption = "Square Root Y vs. X Residual Plot")
```



```
par(mfrow=c(1,1))
```

What we're looking for in such a plot is the absence of a curve, among other things, we want to see “fuzzy football” shapes. The square root version, on the right, is a modest improvement in this regard over the original plot. It does look a bit less curved, and a bit more like a random cluster of points, so that's nice.

Chapter 33

Dehydration Recovery in Kids: A Small Study

The `hydrate` data describe the degree of recovery that takes place 90 minutes following treatment of moderate to severe dehydration, for 36 children diagnosed at a hospital's main pediatric clinic.

Upon diagnosis and study entry, patients were treated with an electrolytic solution at one of seven `dose` levels (0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 mEq/l) in a frozen, flavored, ice popsicle. The degree of rehydration was determined using a subjective scale based on physical examination and parental input, converted to a 0 to 100 point scale, representing the percent of recovery (`recov.score`). Each child's `age` (in years) and `weight` (in pounds) are also available.

First, we'll check ranges (and for missing data) in the `hydrate` file.

```
hydrate
```

```
# A tibble: 36 x 5
  id recov.score dose   age weight
  <int>      <int> <dbl> <int>  <int>
1     1          77    0     4    28
2     2          65   1.5    5    35
3     3          75   2.5    8    55
4     4          63    1     9    76
5     5          75   0.5    5    31
6     6          82    2     5    27
7     7          70    1     6    35
8     8          90   2.5    6    47
9     9          49    0     9    59
10    10         72    3     8    50
# ... with 26 more rows
```

```
summary(hydrate)
```

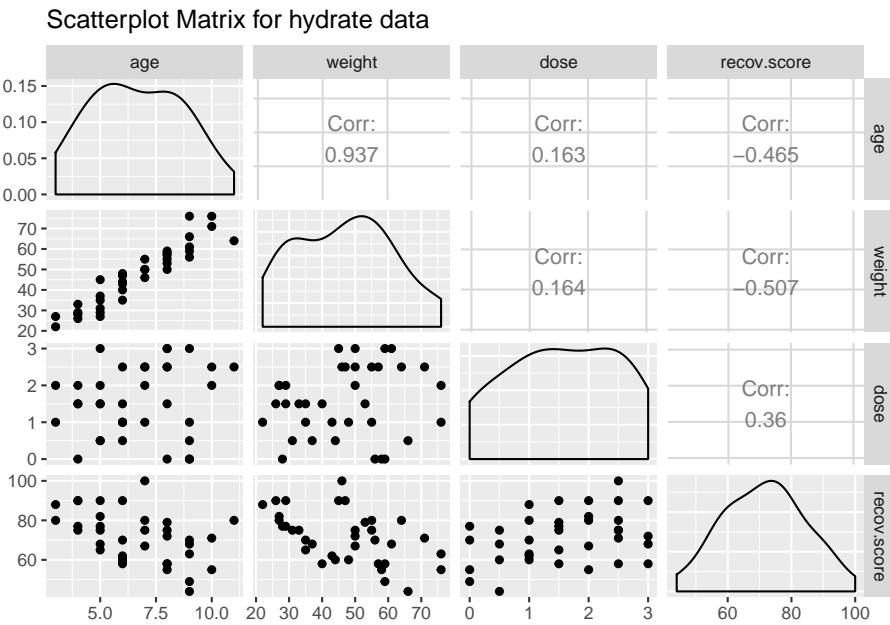
	<code>id</code>	<code>recov.score</code>	<code>dose</code>	<code>age</code>
Min.	: 1.0	Min. : 44.0	Min. :0.00	Min. : 3.00
1st Qu.	: 9.8	1st Qu.: 61.5	1st Qu.:1.00	1st Qu.: 5.00
Median	:18.5	Median : 71.5	Median :1.50	Median : 6.50
Mean	:18.5	Mean : 71.6	Mean :1.57	Mean : 6.67
3rd Qu.	:27.2	3rd Qu.: 80.0	3rd Qu.:2.50	3rd Qu.: 8.00
Max.	:36.0	Max. :100.0	Max. :3.00	Max. :11.00
		weight		
		Min. :22.0		
		1st Qu.:34.5		
		Median :47.5		
		Mean :46.9		
		3rd Qu.:57.2		
		Max. :76.0		

There are no missing values, and all of the ranges make sense. There are no especially egregious problems to report.

33.1 A Scatterplot Matrix

Next, we'll use a scatterplot matrix to summarize relationships between the outcome `recov.score` and the key predictor `dose` as well as the ancillary predictors `age` and `weight`, which are of less interest, but are expected to be related to our outcome. The one below uses the `ggpairs` function in the `GGally` package, as introduced in Part A of the Notes. We place the outcome in the bottom row, and the key predictor immediately above it, with `age` and `weight` in the top rows, using the `select` function within the 'ggpairs' call.

```
GGally::ggpairs(dplyr::select(hydrate, age, weight, dose, recov.score),
                 title = "Scatterplot Matrix for hydrate data")
```



What can we conclude here?

- It looks like `recov.score` has a moderately strong negative relationship with both `age` and `weight` (with correlations in each case around -0.5), but a positive relationship with `dose` (correlation = 0.36).
- The distribution of `recov.score` looks to be pretty close to Normal. No potential predictors (`age`, `weight` and `dose`) show substantial non-Normality.
- `age` and `weight`, as we'd expect, show a very strong and positive linear relationship, with $r = 0.94$
- Neither `age` nor `weight` shows a meaningful relationship with `dose`. ($r = 0.16$)

33.2 Are the recovery scores well described by a Normal model?

Next, we'll do a more thorough graphical summary of our outcome, recovery score, arranging the plots with the help of the `cowplot` package.

```
p1 <- ggplot(hydrate, aes(x = recov.score)) +
  geom_histogram(aes(y = ..density..),
    bins = fd_bins(hydrate$recov.score),
    fill = '#440154', col = '#FDE725') +
  stat_function(fun = dnorm,
```

```

    args = list(mean = mean(hydrate$recov.score),
                sd = sd(hydrate$recov.score)),
    lwd = 1.5, col = '#1FA187') +
  labs(title = "Histogram", x = "Recovery Score", y = "") +
  theme_bw()

p2 <- ggplot(hydrate, aes(x = 1, y = recov.score)) +
  geom_boxplot(fill = '#FDE725', notch = TRUE,
               col = '#440154', outlier.color = '#440154') +
  labs(title = "Boxplot", x = "", y = "") +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

p3 <- ggplot(hydrate, aes(sample = recov.score)) +
  geom_qq(geom = "point", col = '#440154', size = 2) +
  geom_abline(slope = qq_slope(hydrate$recov.score),
              intercept = qq_int(hydrate$recov.score),
              col = '#1FA187', size = 1.25) +
  labs(title = "Normal QQ", x = "", y = "") +
  theme_bw()

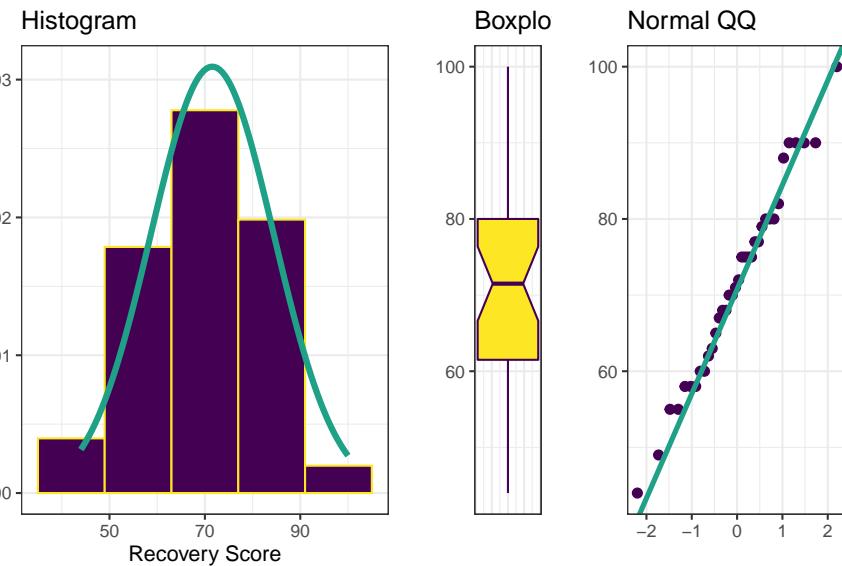
p <- cowplot::plot_grid(p1, p2, p3, align = "h", nrow = 1,
                        rel_widths = c(3, 1, 2))

title <- cowplot::ggdraw() +
  cowplot::draw_label("Recovery Scores from 36 children in the hydrate study",
                      fontface = "bold")

cowplot::plot_grid(title, p, ncol = 1, rel_heights=c(0.1, 1))

```

Recovery Scores from 36 children in the hydrate study



I see no serious problems with assuming Normality for these recovery scores. Our outcome variable doesn't in any way *need* to follow a Normal distribution, but it's nice when it does, because summaries involving means and standard deviations make sense.

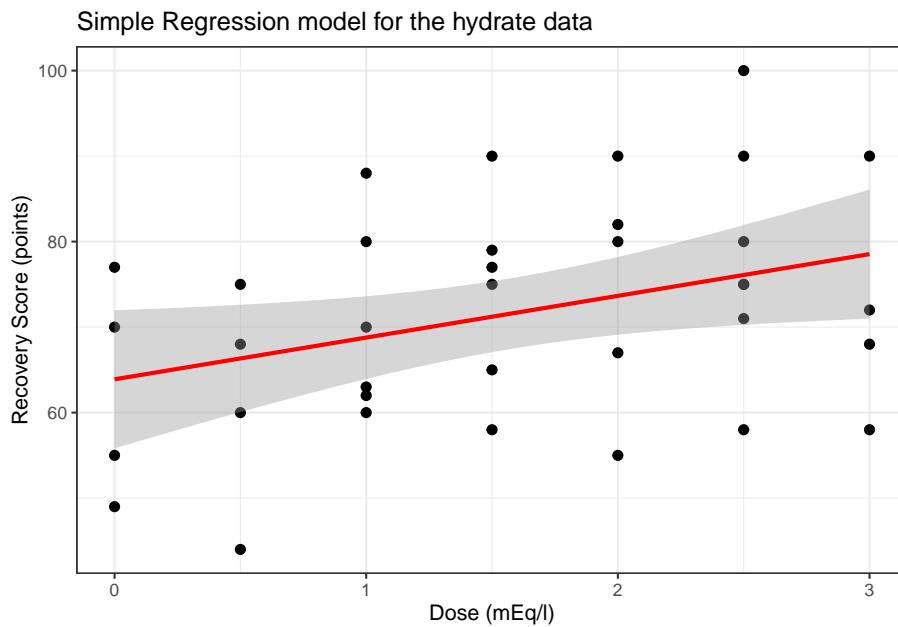
Chapter 34

Simple Regression: Using Dose to predict Recovery

To start, consider a simple (one predictor) regression model using `dose` alone to predict the % Recovery (`recov.score`). Ignoring the `age` and `weight` covariates, what can we conclude about this relationship?

34.1 The Scatterplot, with fitted Linear Model

```
ggplot(hydrate, aes(x = dose, y = recov.score)) +  
  geom_point(size = 2) +  
  geom_smooth(method = "lm", col = "red") +  
  theme_bw() +  
  labs(title = "Simple Regression model for the hydrate data",  
       x = "Dose (mEq/l)", y = "Recovery Score (points)")
```



34.2 The Fitted Linear Model

To obtain the fitted linear regression model, we use the `lm` function:

```
lm(recov.score ~ dose, data = hydrate)
```

Call:
`lm(formula = recov.score ~ dose, data = hydrate)`

Coefficients:
`(Intercept) dose`
`63.90 4.88`

So, our fitted regression model (prediction model) is `recov.score = 63.9 + 4.88 dose`.

34.2.1 Confidence Intervals

We can obtain confidence intervals around the coefficients of our fitted model using the `confint` function.

```
confint(lm(recov.score ~ dose, data = hydrate))
```

	2.5 %	97.5 %
(Intercept)	55.827	72.0
dose	0.466	9.3

34.3 The Summary Output

To get a more complete understanding of the fitted model, we'll summarize it.

```
summary(lm(recov.score ~ dose, data = hydrate))
```

```
Call:
lm(formula = recov.score ~ dose, data = hydrate)

Residuals:
    Min      1Q  Median      3Q     Max 
-22.336 -7.276  0.063  8.423 23.903 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  63.90      3.97   16.09 <2e-16 ***
dose        4.88      2.17    2.25   0.031 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.2 on 34 degrees of freedom
Multiple R-squared:  0.129, Adjusted R-squared:  0.104 
F-statistic: 5.05 on 1 and 34 DF,  p-value: 0.0313
```

34.3.1 Model Specification

1. The first part of the output specifies the model that has been fit.
 - Here, we have a simple regression model that predicts `recov.score` on the basis of `dose`.
 - Notice that we're treating `dose` here as a quantitative variable. If we wanted `dose` to be treated as a factor, we'd have specified that in the model.

34.3.2 Residual Summary

2. The second part of the output summarizes the regression `residuals` across the subjects involved in fitting the model.

- The **residual** is defined as the Actual value of our outcome minus the predicted value of that outcome fitted by the model.
- In our case, the residual for a given child is their actual `recov.score` minus the predicted `recov.score` according to our model, for that child.
- The residual summary gives us a sense of how “incorrect” our predictions are for the `hydrate` observations.
 - A positive residual means that the observed value was higher than the predicted value from the linear regression model, so the prediction was too low.
 - A negative residual means that the observed value was lower than the predicted value from the linear regression model, so the prediction was too high.
 - The residuals will center near 0 (the ordinary least squares model fitting process is designed so the mean of the residuals will always be zero)
 - We hope to see the median of the residuals also be near zero, generally. In this case, the median prediction is 0.06 point too low.
 - The minimum and maximum show us the largest prediction errors, made in the subjects used to fit this model.
 - Here, we predicted a recovery score that was 22.3 points too high for one patient, and another of our predicted recovery scores was 23.9 points too low.
 - The middle half of our predictions were between 8.4 points too low and 7.3 points too high.

34.3.3 Coefficients Output

3. The **Coefficients** output begins with a table of the estimated coefficients from the regression equation.
 - Generally, we write a simple regression model as $y = \beta_0 + \beta_1 x$.
 - In the `hydrate` model, we have `recov.score` = $\beta_0 + \beta_1 \text{dose}$.
 - The first column of the table gives the estimated β coefficients for our model
 - Here the estimated intercept $\hat{\beta}_0 = 63.9$
 - The estimated slope of dose $\hat{\beta}_1 = 4.88$
 - Thus, our model is `recov.score` = $63.9 + 4.88 \text{dose}$

We interpret these coefficients as follows:

- The intercept (63.9) is the predicted `recov.score` for a patient receiving a `dose` of 0 mEq/l of the electrolytic solution.
- The slope (4.88) of the `dose` is the predicted *change* in `recov.score` associated with a 1 mEq/l increase in the dose of electrolytic solution.

- Essentially, if we have two children like the ones studied here, and we give Roger a popsicle with dose X and Sarah a popsicle with dose X + 1, then this model predicts that Sarah will have a recovery score that is 4.88 points higher than will Roger.
- From the confidence interval output we saw previously with the function `confint(lm(recov.score ~ dose))`, we are 95% confident that the true slope for `dose` is between (0.47, 9.30) mEq/l. We are also 95% confident that the true intercept is between (55.8, 72.0).

34.3.4 Correlation and Slope

If we like, we can use the `cor` function to specify the Pearson correlation of `recov.score` and `dose`, which turns out to be 0.36. - Note that the `slope` in a simple regression model will follow the sign of the Pearson correlation coefficient, in this case, both will be positive.

```
cor(hydrate$recov.score, hydrate$dose)
```

```
[1] 0.36
```

34.3.5 Coefficient Testing

```
summary(lm(recov.score ~ dose, data = hydrate))
```

Call:

```
lm(formula = recov.score ~ dose, data = hydrate)
```

Residuals:

Min	1Q	Median	3Q	Max
-22.336	-7.276	0.063	8.423	23.903

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	63.90	3.97	16.09	<2e-16 ***
dose	4.88	2.17	2.25	0.031 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.2 on 34 degrees of freedom

Multiple R-squared: 0.129, Adjusted R-squared: 0.104

F-statistic: 5.05 on 1 and 34 DF, p-value: 0.0313

Next to each coefficient in the summary regression table is its estimated standard error, followed by the coefficient's t value (the coefficient value divided by the

standard error), and the associated two-tailed p value for the test of:

- H_0 : This coefficient's β value = 0 vs.
- H_A : This coefficient's β value $\neq 0$.

For the slope coefficient, we can interpret this choice as:

- H_0 : This predictor adds no predictive value to the model vs.
- H_A : This predictor adds statistically significant predictive value to the model.

The t test of the intercept is rarely of interest, because a. we rarely care about the situation that the intercept predicts, where all of the predictor variables are equal to zero and b. we usually are going to keep the intercept in the model regardless of its statistical significance.

In the `hydrate` simple regression model,

- the intercept is statistically significantly different from zero at all reasonable α levels since $\Pr(>|t|)$, the p value is (for all intents and purposes) zero.
- A significant p value for this intercept implies that the predicted recovery score for a patient fed a popsicle with 0 mEq/l of the electrolytic solution will be different than 0%.
- By running the `confint` function we have previously seen, we can establish a confidence interval for the intercept term (and the slope of dose, as well).

```
confint(lm(recov.score ~ dose, data = hydrate), level = .95)
```

	2.5 %	97.5 %
(Intercept)	55.827	72.0
dose	0.466	9.3

The t test for the slope of `dose`, on the other hand, is important. This tests the hypothesis that the true slope of `dose` is zero vs. a two-tailed alternative.

If the slope of dose was in fact zero, then this would mean that knowing the dose information would be of no additional value in predicting the outcome over just guessing the mean of `recov.score` for every subject.

So, since the slope of dose is significantly different than zero (as it is at the 5% significance level, since $p = 0.031$),

- `dose` has statistically significant predictive value for `recov.score`,
- more generally, this model has statistically significant predictive value as compared to a model that ignores the `dose` information and simply predicts the mean of `recov.score` for each subject.

34.3.6 Summarizing the Quality of Fit

4. The next part of the regression summary output is a summary of fit quality.

The **residual standard error** estimates the standard deviation of the prediction errors made by the model.

- If assumptions hold, the model will produce residuals that follow a Normal distribution with mean 0 and standard deviation equal to this residual standard error.
 - So we'd expect roughly 95% of our residuals to fall between $-2(12.21)$ and $+2(12.21)$, or roughly -24.4 to +24.4 and that we'd see virtually no residuals outside the range of $-3(12.21)$ to $+3(12.21)$, or roughly -36.6 to +36.6.
 - The output at the top of the summary tells us about the observed regression residuals, and that they actually range from -22 to +24.
 - In context, it's hard to know whether or not we should be happy about this. On a scale from 0 to 100, rarely missing by more than 24 seems OK to me, but not terrific.
- The **degrees of freedom** here are the same as the denominator degrees of freedom in the ANOVA to follow. The calculation is $n - k$, where n = the number of observations and k is the number of coefficients estimated by the regression (including the intercept and any slopes).
 - Here, there are 36 observations in the model, and we fit $k = 2$ coefficients; the slope and the intercept, as in any simple regression model, so $df = 36 - 2 = 34$.

The multiple R² value is usually just referred to as R² or R-squared.

- This is interpreted as the proportion of variation in the outcome variable that has been accounted for by our regression model.
 - Here, we've accounted for just under 13% of the variation in % Recovery using Dose.
- The R in multiple R-squared is the Pearson correlation of `recov.score` and `dose`, which in this case is 0.3595.
 - Squaring this value gives the R² for this simple regression.
 - $(0.3595)^2 = 0.129$

R² is greedy.

- R² will always suggest that we make our models as big as possible, often including variables of dubious predictive value.
- As a result, there are various methods for adjusting or penalizing R² so that we wind up with smaller models.
- The **adjusted R²** is often a useful way to compare multiple models for the same response.
 - $R_{adj}^2 = 1 - \frac{(1-R^2)(n-1)}{n-k}$, where n = the number of observations and k is the number of coefficients estimated by the regression (including the intercept and any slopes).
 - So, in this case, $R_{adj}^2 = 1 - \frac{(1-0.1293)(35)}{34} = 0.1037$
 - The adjusted R² value is not, technically, a proportion of anything, but it is comparable across models for the same outcome.

- The adjusted R^2 will always be less than the (unadjusted) R^2 .

34.3.7 ANOVA F test

5. The last part of the standard summary of a regression model is the overall ANOVA F test.

The hypotheses for this test are:

- H_0 : The model has **no** statistically significant predictive value, at all vs.
- H_A : The model has statistically significant predictive value.

This is equivalent to:

- H_0 : Each of the coefficients in the model (other than the intercept) has $\beta = 0$ vs.
- H_A : At least one regression slope has $\beta \neq 0$

Since we are doing a simple regression with just one predictor, the ANOVA F test hypotheses are exactly the same as the t test for dose:

- H_0 : The slope for **dose** has $\beta = 0$ vs.
- H_A : The slope for **dose** has $\beta \neq 0$

In this case, we have an F statistic of 5.05 on 1 and 34 degrees of freedom, yielding $p = 0.03$

- At $\alpha = 0.05$, we conclude that there is statistically significant predictive value somewhere in this model, since $p < 0.05$.
 - This is conclusive evidence that “something” in our model (here, **dose** is the only predictor) predicts the outcome to a degree beyond that easily attributed to chance alone.
- Another appropriate conclusion is that the R^2 value (13%) is a statistically significant amount of variation in **recov.score** that is accounted for by a linear regression on **dose**.
- In *simple regression* (regression with only one predictor), the t test for the slope (**dose**) always provides the same p value as the ANOVA F test.
 - The F test statistic in a *simple regression* is always by definition just the square of the slope’s t test statistic.
 - Here, $F = 5.047$, and this is the square of $t = 2.247$ from the Coefficients output

34.4 Viewing the complete ANOVA table

We can obtain the complete ANOVA table associated with this particular model, and the details behind this F test using the **anova** function:

```
anova(lm(recov.score ~ dose, data = hydrate))
```

Analysis of Variance Table

Response: recov.score

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
dose	1	752	752	5.05	0.031 *
Residuals	34	5067	149		

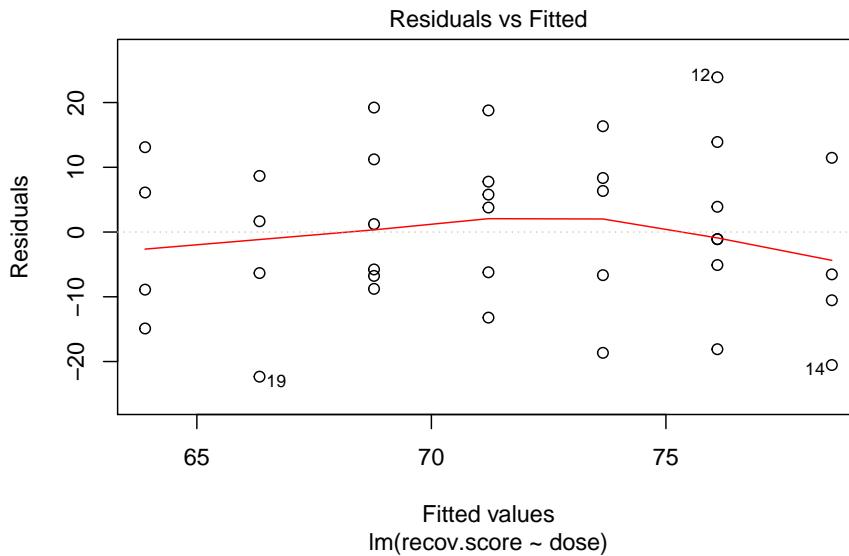
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- The R^2 for our regression model is equal to the η^2 for this ANOVA model.
 - If we divide $SS(dose) = 752.2$ by the total sum of squares ($752.2 + 5066.7$), we'll get the multiple R^2 [0.1293]
- Note that this is *not* the same ANOVA model we would get if we treated `dose` as a factor with seven levels, rather than as a quantitative variable.

34.5 Plotting Residuals vs. Fitted Values

We can obtain the plot of residuals vs. fitted values from this model using:

```
plot(lm(recov.score ~ dose, data = hydrate), which = 1)
```



We hope in this plot to see a generally random scatter of points, perhaps looking

like a “fuzzy football”. Since we only have seven possible `dose` values, we obtain only seven distinct predicted values, which explains the seven vertical lines in the plot. Here, the smooth red line indicates a gentle curve, but no evidence of a strong curve, or any other regular pattern in this residual plot.

To save the residuals and predicted (fitted) values from this simple regression model, we can use the `resid` and `fitted` commands, respectively, or we can use the `augment` function in the `broom` package to obtain a tidy data set containing these objects and others.

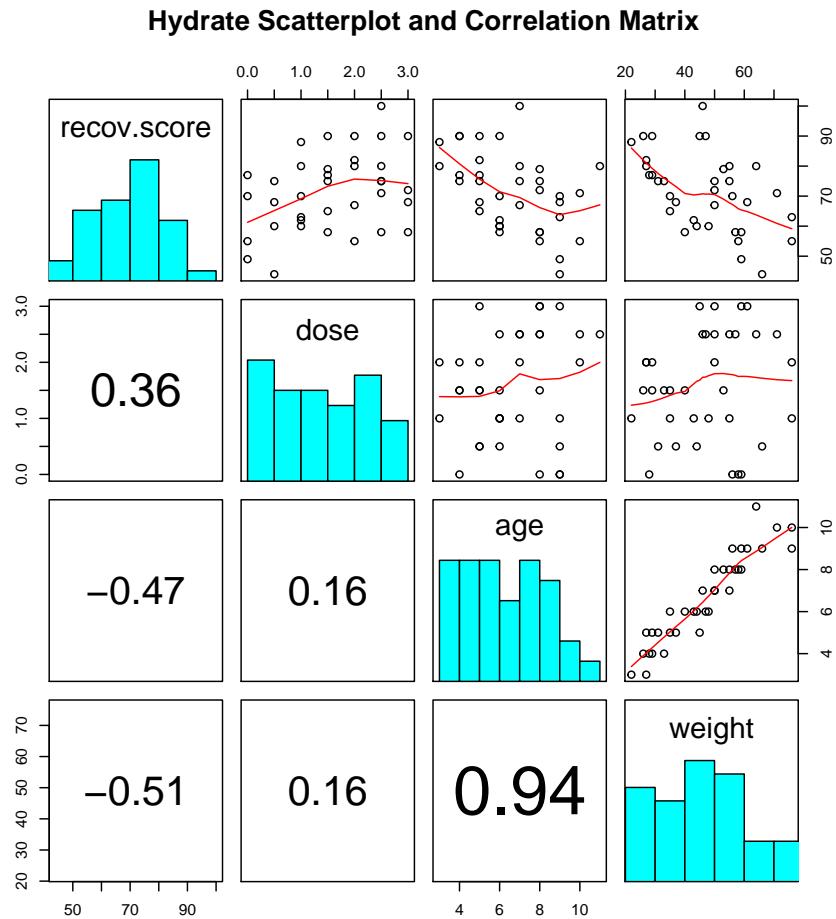
Chapter 35

Multiple Regression with the `hydrate` data

35.1 Another Scatterplot Matrix for the `hydrate` data

Along the diagonals of the scatterplot matrix, we have histograms of each of the variables. In the first row of the matrix, we have scatterplots of Recovery (on the y-axis) against each of the predictor variables in turn. We see a positive relationship with Dose, and a negative relationship with both Age, and then Weight. All possible scatterplots are shown, including plots that look at the association between the predictor variables.

```
pairs (~ recov.score + dose + age + weight, data=hydrate,  
      main="Hydrate Scatterplot and Correlation Matrix",  
      upper.panel = panel.smooth,  
      diag.panel = panel.hist,  
      lower.panel = panel.cor)
```



We see the positive correlation between Recovery and Dose (+.36) and the negative correlations between Recovery and Age (-.47) and Recovery and Weight (-.51). We can also see a very strong positive correlation between Weight and Age (+.94), which implies that it may be very difficult to separate out the effect of Weight from the effect of Age on our response.

35.2 A Multiple Regression for `recov.score`

Our first multiple linear regression model will predict `recov.score` using three predictors: `dose`, `age` and `weight`.

```
summary(lm(recov.score ~ dose + age + weight, data = hydrate))
```

```

Call:
lm(formula = recov.score ~ dose + age + weight, data = hydrate)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.68  -6.49  -2.20   7.67  22.13 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  85.476     5.965   14.33  1.8e-15 ***
dose         6.170     1.791    3.45   0.0016 **  
age          0.277     2.285    0.12   0.9043    
weight       -0.543     0.324   -1.68   0.1032    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.92 on 32 degrees of freedom
Multiple R-squared:  0.458, Adjusted R-squared:  0.408 
F-statistic: 9.03 on 3 and 32 DF,  p-value: 0.000177

```

35.2.1 Model Specification

```

Call:
lm(formula = recov.score ~ dose + age + weight)

```

The output begins by presenting the R function call. Here, we have a linear model, with `recov.score` being predicted using `dose`, `age` and `weight`, all from the `hydrate` data.

35.2.2 Model Residuals

```

Residuals:
    Min      1Q  Median      3Q     Max 
-16.682  -6.492  -2.204   7.667  22.128

```

Next, we summarize the residuals, where Residual = Actual Value - Predicted Value.

- This gives us a sense of how incorrect our predictions are for the `hydrate` observations.
 - The residuals will center near 0 (least squares is designed so the mean will always be zero, here the median prediction is 2.2 points too high)
 - Here, we predicted a `recov.score` that was 16.68 points too high for one patient, and another of our predicted `recov.score` values was

22.13 points too low.

- The middle half of our predictions were between 6.5 points too high (so the residual = observed - predicted is -6.5) and 7.7 points too low.

35.2.3 Model Coefficients

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.4764	5.9653	14.329	1.79e-15 ***
dose	6.1697	1.7908	3.445	0.00161 **
age	0.2770	2.2847	0.121	0.90428
weight	-0.5428	0.3236	-1.677	0.10325

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

The first column of this table gives the estimated coefficients for our model (including the intercept, and slopes for each of our three predictors). Our least squares equation is `recov.score` = 85.48 + 6.17 `dose` + 0.28 `age` - 0.54 `weight`.

35.2.4 Interpreting the t tests (last predictor in)

Next to each coefficient is its estimated standard error, followed by the coefficient's t value (the coefficient value divided by the standard error), and the associated two-tailed p value. The p value addresses H_0 : This coefficient's β value = 0, as **last predictor into the model**.

- The t test of the intercept is rarely of interest, because
 - a. we rarely care about the situation that the intercept predicts; where all of the predictor variables are equal to zero, and
 - b. we usually are going to keep the intercept in the model regardless of its statistical significance.

The t test for dose tests the hypothesis that the slope of dose should be zero, as **last predictor in**.

- If the slope of `dose` was in fact zero, that would mean that knowing the `dose` information would be of no additional value in predicting the outcome after you have accounted for `age` and `weight` in your model.
- **Last predictor in**
 - This **last predictor in** business means that the test is comparing a model with `dose`, `age` and `weight` to a model with `age` and `weight` alone, to see if the incremental benefit of adding `dose` provides statistically significant additional predictive value for `recov.score`.
 - The t test tells us if `dose` is a useful addition to a model that already contains the other two variables.

- The p value = .0016, so, at any reasonable α level, the `dose` received is a significant part of the predictive model, even after accounting for `age` and `weight`.

35.2.4.1 The t test of `age`

The t test of `age` tests H_0 : `age` has no predictive value for `recov.score` as last predictor in.

- Here $p = .9043$, which indicates that `age` doesn't add statistically significant predictive value to the model once we have adjusted for `dose` and `weight`.
- This **does not** mean that `age`, by itself, has no linear relationship with `recov.score`, but it does mean that in this model, it doesn't help to add `age` once we already have `dose` and `weight`.
- This shouldn't be surprising, given the strong correlation between `age` and `weight` shown in the scatterplot matrix.

35.2.4.2 The t test of `weight`

The t test of `weight` tests the null hypothesis that `weight` has no predictive value for `recov.score` as last predictor in.

- Here $p = .1032$, which is better than we saw in the `age` variable, but still indicates that `weight` adds no significant predictive value to the model for `recov.score` if we already have `dose` and `age`.

35.2.5 The Effect of Collinearity

The situation we see here with `age` and `weight`, where two predictors are highly correlated with one another, making it hard to see significant predictive value for either one of them after the other is already in the model is referred to as **collinearity**.

- Our usual approach to dealing with this collinearity will be to consider dropping one predictor from our model, and refit.
- Dropping either predictor will likely have a fairly small impact on the fit quality of our model, but if we drop both, we may do a lot of damage.

35.2.6 Confidence Intervals for the Slopes in a Multiple Regression Model

Here are the confidence intervals for the coefficients of our multiple regression model.

```
confint(lm(recov.score ~ dose + age + weight, data = hydrate))

2.5 % 97.5 %
(Intercept) 73.33 97.627
dose         2.52  9.817
age          -4.38  4.931
weight       -1.20  0.116
```

We conclude, for instance, with 95% confidence, that that true slope of dose is between 2.52 and 9.82 points on the recovery score scale.

This is pretty different from the confidence interval we found for the slope of dose in a simple regression on dose alone that we saw previously, and repeat below.

```
confint(lm(recov.score ~ dose, data = hydrate), level = .95)

2.5 % 97.5 %
(Intercept) 55.827  72.0
dose        0.466   9.3
```

In both cases, the reasonable range of values for the slope of `dose` appears to be positive, but the range of values is much tighter in the multiple regression.

35.2.7 Model Summaries

```
Residual standard error: 9.923 on 32 degrees of freedom
Multiple R-squared:  0.4584,    Adjusted R-squared:  0.4077
F-statistic:  9.03 on 3 and 32 DF,  p-value: 0.0001769
```

The **residual standard error** estimates the standard deviation of our model's errors to be 9.923.

- We'd expect roughly 95% of our residuals to fall between $-2(9.923)$ and $+2(9.923)$, or roughly -20 to +20.
- We'd expect to see virtually no residuals outside the range of $\pm 3(9.923)$ or roughly -30 to +30.

The **coefficient of determination**, R^2 , is estimated to be 0.4584

- We accounted for just under 46% of the variation in `recov.score` using `dose`, `age` and `weight`.

R^2 will always suggest that we make our models as big as possible, often including variables of dubious predictive value.

- An important thing to realize is that **if you add a variable to a model, R^2 cannot decrease**.
- Similarly, removing a variable from a model, no matter how unrelated to the outcome, cannot increase the R^2 value.

- Thus, using R^2 as the be-all and end-all of model fit for a regression model goes against the general notion that we would like to have parsimonious models; that is to say, we favor simple models when possible.

The overall ANOVA F test is presented last.

- The test gives an F-statistic of 9.03 on 3 and 32 degrees of freedom, yielding a p value of .00018.
- We can conclude (at any reasonable α level) that there is statistically significant predictive value somewhere in this model.
- The F test for a multiple regression is a very low standard; specifying only that we have conclusive evidence that some part of the model predicts the outcome to a degree beyond that easily attributed to chance alone.
- We return to the individual t tests to assess significance after adjusting for the other predictors.
- Concluding that the F test is significant means that the multiple R^2 accounted for by the model is also statistically significant.

35.2.7.1 Verifying the R^2 Calculations

The formula for R^2 is simply the sum of squares accounted for by the regression model divided by the total sum of squares.

- This is the same as 1 minus [residual sum of squares divided by total sum of squares].
- We can verify the calculation with the complete ANOVA table for this model

```
anova(lm(recov.score ~ dose + weight + age, data = hydrate))
```

Analysis of Variance Table

```
Response: recov.score
          Df Sum Sq Mean Sq F value Pr(>F)
dose      1    752     752   7.64 0.00940 **
weight    1  1914    1914  19.44 0.00011 ***
age       1      1       1   0.01 0.90428
Residuals 32  3151     98
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\text{So } R^2 = 1 - \frac{SS(\text{Residual})}{SS(\text{Total})} = 1 - \frac{3151.22}{752.15+1914.07+1.45+3151.22} = 1 - \frac{3151.22}{5818.89} = 1 - 0.5416 = .4584$$

If we are fitting a model to n data points, and we have k coefficients (slopes + intercept) in our model, then the model's adjusted R^2 is $R_{adj}^2 = 1 - \frac{(1-R^2)(n-1)}{n-k}$.

In the `hydrate` data, we have $n = 36$ data points, and $k = 4$ coefficients (3 slopes, plus the intercept), so $R_{adj}^2 = 1 - \frac{(1-.4584)(36-1)}{36-4} = .4077$

- Again, the adjusted R^2 is *not* interpreted as a percentage of anything, and can in fact be negative.
- R_{adj}^2 will always be less than the original R^2 so long as there is at least one predictor besides the intercept term, so that $k > 1$ in the equation above.

35.3 ANOVA for Sequential Comparison of Models

Outside of the main summary, we can also run a sequence of comparisons of the impact of various predictors in our model with the `anova` function.

```
anova(lm(recov.score ~ dose + age + weight, data = hydrate))
```

Analysis of Variance Table

```
Response: recov.score
          Df Sum Sq Mean Sq F value    Pr(>F)
dose      1    752     752   7.64 0.00940 ***
age       1   1639     1639  16.64 0.00028 ***
weight    1    277     277   2.81 0.10325
Residuals 32   3151      98
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This ANOVA table is very different from the one we saw in our simple regression model. The various p values shown here indicate the significance of predictors taken in turn, specifically...

1. The p value for H_0 : `dose` has significant predictive value by itself is 0.009
2. The p value for H_0 : `age` adds significant predictive value once you already have `dose` in the model is 0.0003
3. The p value for H_0 : `weight` adds significant predictive value once you already have `dose` and `age` in the model (i.e. as last predictor in) is 0.1032

Note that this last p value is the same as the t test for `weight` we have already seen in the main summary of the model.

If we change the order of the predictors entering the model, the main summary of our linear model will not change, but these ANOVA results will change.

```
summary(lm(recov.score ~ dose + weight + age, data = hydrate))
```

Call:

```

lm(formula = recov.score ~ dose + weight + age, data = hydrate)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.68  -6.49  -2.20   7.67  22.13 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  85.476    5.965   14.33  1.8e-15 ***
dose         6.170    1.791    3.45   0.0016 **  
weight       -0.543    0.324   -1.68   0.1032    
age          0.277    2.285    0.12   0.9043    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.92 on 32 degrees of freedom
Multiple R-squared:  0.458, Adjusted R-squared:  0.408 
F-statistic: 9.03 on 3 and 32 DF,  p-value: 0.000177

anova(lm(recov.score ~ dose + weight + age, data = hydrate))

```

Analysis of Variance Table

```

Response: recov.score
           Df Sum Sq Mean Sq F value Pr(>F)    
dose        1    752    752    7.64 0.00940 **  
weight       1   1914   1914   19.44 0.00011 ***  
age          1      1      1    0.01 0.90428    
Residuals  32   3151     98                
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

- Does `age` add significant predictive value to the model including `dose` and `weight`?
 - **No**, because the p value for `age` is 0.904 once we have previously accounted for `dose` and `weight`.
- Does `weight` add significant predictive value to the model that includes `dose` only?
 - **Yes**, because the p value for `weight` is 0.00011 once we have accounted for `dose`.

35.3.1 Building The ANOVA Table for The Model

Sometimes, we want an ANOVA table for the Regression as a whole, to compare directly to the Residuals. We can build this by adding up the sums of squares and degrees of freedom from the individual predictors, then calculating the Mean

Square and F ratio for the Regression.

- The degrees of freedom are easy. We have three predictors (slopes), each accounting for one DF, and we have 32 DF applied to the residuals, so total DF = n - 1 = 35

ANOVA Table	DF	SS	MS	F	p value
Regression	3	?	?	?	?
Residuals	32	3151.22	98.48		
Total	35	?			

- To obtain the sum of squares due to the regression model, we just add the sums of squares for the individual predictors, so SS(Regression) = 752.15 + 1914.07 + 1.45 = 2667.67
- The total sum of squares is then the sum of SS(Regression) and SS(Residuals): here, 2667.67 + 3151.22 = 5818.89
- Now, we recall that the Mean Square for any row in this table is just the Sum of Squares divided by the degrees of freedom, so that MS(Regression) = 2667.67 / 3 = 889.22

ANOVA Table	DF	SS	MS	F	p value
Regression	3	2667.67	889.22	?	?
Residuals	32	3151.22	98.48		
Total	35	5818.89			

- The F ratio and p value can be obtained from the original summary of the model.

ANOVA Table	DF	SS	MS	F	p value
Regression	3	2667.67	889.22	9.03	0.00018
Residuals	32	3151.22	98.48		
Total	35	5818.89			

We can use this to verify that the R² for the model is also equal to SS(Regression) / SS(Total)

$$R^2 = \frac{SS(\text{Regression})}{SS(\text{Total})} = \frac{2667.67}{5818.89} = 0.4584$$

35.4 Standardizing the Coefficients of a Model

Which of the three predictors: `dose`, `age` and `weight`, in our model for `recov.score`, has the largest effect?

Sometimes, we want to express the coefficients of the regression in a standardized way, to compare the impact of each predictor within the model on a fairer scale. A common trick is to “standardize” each input variable (predictor) by subtracting its mean and dividing by its standard deviation. Each coefficient in this semi-standardized model is the expected difference in the outcome, comparing subjects that differ by one standard deviation in one variable with all other variables fixed at their average. R can do this rescaling quite efficiently, with the use of the `scale` function.

35.4.1 A Semi-Standardized Model for the `hydrate` data

```
summary(lm(recov.score ~ scale(dose) + scale(weight) + scale(age), data = hydrate))

Call:
lm(formula = recov.score ~ scale(dose) + scale(weight) + scale(age),
    data = hydrate)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.68  -6.49  -2.20   7.67  22.13 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  71.556     1.654   43.26 <2e-16 ***
scale(dose)   5.860     1.701    3.45  0.0016 **  
scale(weight) -8.040     4.794   -1.68  0.1032    
scale(age)     0.581     4.793    0.12  0.9043    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.92 on 32 degrees of freedom
Multiple R-squared:  0.458, Adjusted R-squared:  0.408 
F-statistic: 9.03 on 3 and 32 DF,  p-value: 0.000177
```

The only things that change here are the estimates and standard errors of the coefficients: every other bit of the output is unchanged from our original summary.

- Each of the scaled covariates has mean zero and standard deviation one.
 - `scale(dose)`, for instance, is obtained by subtracting the mean from the original `dose` variable (so the result is centered at zero) and dividing that by the standard deviation (so that `scale(dose)` has mean 0 and standard deviation 1.)
 - We interpret the coefficient of `scale(dose) = 5.86` as the change

in our outcome (`recov.score`) that we anticipate when the dose increases by one standard deviation from its mean, while all of the other inputs (`weight` and `age`, specifically) remain at their means.

This allows us to compare the effects on `recov.score` due to `dose`, to `weight` and to `age` in terms of a change of one standard deviation in each, while holding the others constant.

- Which of the inputs appears to have the biggest impact on recovery score in this sense?

35.4.1.1 Interpreting the Intercept in a Model with Semi-Standardized Coefficients

The semi-standardized model has an interesting intercept term. The intercept is equal to the mean of our outcome (`recov.score`) across the full set of subjects.

- When `scale(dose)`, `scale(weight)` and `scale(age)` are all zero, this means that `dose`, `weight` and `age` are at their means.
- So the intercept tells you the value of `recov.score` that we would predict when all of the scaled predictors are zero, e.g., when all of the original inputs to the model (`dose`, `weight` and `age`) are at their average values.

35.5 Comparing Fits of Several Possible Models for Recovery Score

Below, I summarize the results for five possible models for `recov.score`. What can we conclude? By this set of results, which of the models looks best?

Model	Fitted Equation	R^2	R^2_{adj}	RSE	F test	p
[Int]	71.6	-	-	12.89	-	-
D	$63.9 + 4.88 \text{ dose}$.1293	.1037	12.21	0.0313	
DA	$84.1 + 6.07 \text{ dose} - 3.31 \text{ age}$.4108	.3751	10.19	0.0002	
DW	$85.6 + 6.18 \text{ dose} - 0.51 \text{ weight}$.4582	.4254	9.77	4.1e-05	
DAW	$85.5 + 6.17 \text{ dose} + 0.28 \text{ age} - 0.54 \text{ weight}$.4584	.4077	9.92	0.0002	

It appears as though the DW model is almost as good as the DAW model using the multiple R^2 as a criterion, and is the best of these five models using any of the other criteria.

The five summaries I used to obtain this table were obtained with the following code (not evaluated here):

```
summary(lm(recov.score ~ 1))
```

```
summary(lm(recov.score ~ dose))
summary(lm(recov.score ~ dose + age))
summary(lm(recov.score ~ dose + weight))
summary(lm(recov.score ~ dose + age + weight))
```

35.6 Comparing Model Fit: The AIC, or Akaike Information Criterion

Another summary we'll use to evaluate a series of potential regression models for the same outcome is the Akaike Information Criterion or AIC. Smaller values indicate better models, by this criterion, which is just a measure of relative quality.

Model	Intercept only	D	DA	DW	DAW
AIC	289.24	286.25	274.19	271.17	273.16

By the AIC, DW looks best out of these five models.

Note: R uses the AIC function applied to a model to derive these results. For example,

```
AIC(lm(recov.score ~ 1, data = hydrate))

[1] 289

AIC(lm(recov.score ~ dose + age + weight, data = hydrate))

[1] 273
```

35.7 Comparing Model Fit with the BIC, or Bayesian Information Criterion

Another summary we'll use to evaluate potential regression models for the same outcome is the Bayesian Information Criterion or BIC. Like the AIC, smaller values indicate better models, by this criterion, which is also a measure of relative quality.

Model	Intercept only	D	DA	DW	DAW
BIC	292.4	291.0	280.5	277.5	281.1

By the BIC, as well, DW looks best out of these five models.

To obtain the results, just apply BIC instead of AIC to the model.

```
BIC(lm(recov.score ~ dose + weight, data = hydrate))
```

```
[1] 278
```

35.8 Making Predictions for New Data: Prediction vs. Confidence Intervals

The `predict` function, like the `fitted` function, when applied to a linear model, can produce the fitted values predicted by the model. Yet there is more we can do with `predict`.

Suppose we want to use the model to predict our outcome (recovery score, or `recov.score`) on the basis of the three predictors (`dose`, `age` and `weight`). Building an interval forecast around a fitted value requires us to decide whether we are:

- predicting the `recov.score` for **one particular child** with the specified characteristics (in which case we use something called a **prediction interval**) or
- predicting the mean `recov.score` across **all children** that have the specified `dose`, `age` and `weight` characteristics (in which case we use a **confidence interval**).

The *prediction* interval will always be wider than the related *confidence* interval.

35.8.1 Making a Prediction using the `hydrate` data

Now, suppose that we wish to predict a `recov.score` associated with `dose = 2`, `age = 7` and `weight = 50`. The approach I would use follows...

```
modela <- lm(recov.score ~ dose + age + weight, data = hydrate)
newdata <- data.frame(dose = 2, age = 7, weight = 50)
predict(modela, newdata, interval="prediction", level=0.95)

      fit  lwr  upr
1 72.6 52.1 93.2

predict(modela, newdata, interval="confidence", level=0.95)

      fit  lwr  upr
1 72.6 68.9 76.4
```

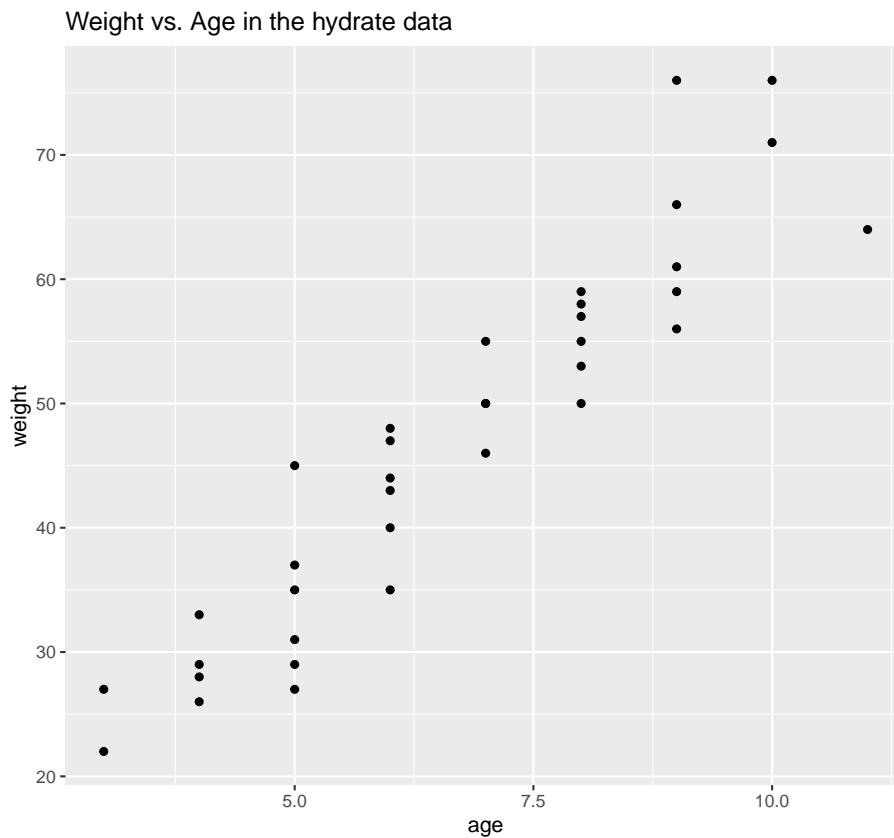
So, our conclusions are:

- If we have one particular child with `dose = 2`, `age = 7` and `weight = 50`, we have 95% confidence that the `recov.score` for this child will be between 52.1 and 93.2.
- If we have a large group of children with `dose = 2`, `age = 7` and `weight = 50`, we have 95% confidence that the average `recov.score` for these children will be between 68.9 and 76.4.

35.8.2 A New Prediction?

Suppose that now you wanted to make a prediction for a `recov.score` with `dose = 1`, `age = 4` and `weight = 60`. Why would this be a meaningfully worse idea than the prediction we just made, and how does the plot below tell us this?

```
ggplot(hydrate, aes(x = age, y = weight)) +
  geom_point() +
  labs(title = "Weight vs. Age in the hydrate data")
```



35.9 Interpreting the Regression Model: Two Key Questions

Suppose we land, finally, on the DW model...

```
summary(lm(recov.score ~ dose + weight, data = hydrate))
```

Call:

```
lm(formula = recov.score ~ dose + weight, data = hydrate)
```

Residuals:

Min	1Q	Median	3Q	Max
-16.61	-6.49	-2.12	7.60	22.25

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.594	5.797	14.77	4.3e-16 ***
dose	6.175	1.763	3.50	0.0013 **
weight	-0.506	0.113	-4.48	8.6e-05 ***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1	'	'	1

Residual standard error: 9.77 on 33 degrees of freedom

Multiple R-squared: 0.458, Adjusted R-squared: 0.425

F-statistic: 14 on 2 and 33 DF, p-value: 4.06e-05

Suppose we decide that this model is a reasonable choice, based on adherence to regression assumptions (as we'll discuss shortly), and quality of fit as measured both by R^2 measures, the various hypothesis tests, and the information criteria (AIC and BIC).

Question 1. Can we summarize the model and how well it fits the data in a reasonable English sentence or two?

- Together, dose and weight account for just under 46% of the variation in recovery scores, and this is highly statistically significant ($p < 0.001$). Higher recovery scores are associated with higher doses and with lower weights among the 36 children assessed in this study.

Question 2. How might we interpret the coefficient of dose for someone who was smart but not a student of statistics?

- If we have two kids who are the same weight, then if kid A receives a dose that is 1 mEq/L larger than kid B, we'd expect kid A's recovery score to be a little over 6 points better than the score for kid B.

Chapter 36

Regression Diagnostics

Some of this discussion comes from Bock, Velleman, and De Veaux (2004).

Multiple linear regression (also called ordinary least squares, or OLS) has four main assumptions that are derived from its model.

For a simple regression, the model underlying the regression line is $E(Y) = \beta_0 + \beta_1 x$

- where $E(Y)$ = the expectation (mean) of the outcome Y ,
- β_0 is the intercept, and
- β_1 is the slope

Now, if we have a multiple regression with three predictors x_1, x_2 and x_3 , as we do in the `hydrate` case (`dose`, `age` and `height`), then the model becomes $E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$.

- As the simple regression model did, the multiple regression model predicts the mean of our outcome, for a subject (or group of subjects) with specific values of the predictors.
- In the `hydrate` example, our model is $E(\text{'recov.score'}) = \beta_0 + \beta_{\text{dose}} \text{ dose} + \beta_{\text{age}} \text{ age} + \beta_{\text{weight}} \text{ weight}$
 - In a larger version of this study, we can imagine finding many kids with the same `age` and `weight` receiving the same `dose`, but they won't all have exactly the same `recov.score`.
 - Instead, we'd have many different recovery score values.
 - It is the mean of that distribution of recovery scores that the regression model predicts.

Alternatively, we can write the model to relate individual y 's to the x 's by adding an individual error term: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$

Of course, the multiple regression model is not limited to three predictors. We will often use k to represent the number of coefficients (slopes plus intercept) in

the model, and will also use p to represent the number of predictors (usually $p = k - 1$).

36.1 The Four Key Regression Assumptions

The assumptions and conditions for a multiple regression model are nearly the same as those for simple regression. The key assumptions that must be checked in a multiple regression model are:

- Linearity Assumption
- Independence Assumption
- Equal Variance (Constant Variance / Homoscedasticity) Assumption
- Normality Assumption

Happily, R is well suited to provide us with multiple diagnostic tools to generate plots and other summaries that will let us look more closely at each of these assumptions.

36.2 The Linearity Assumption

We are fitting a linear model.

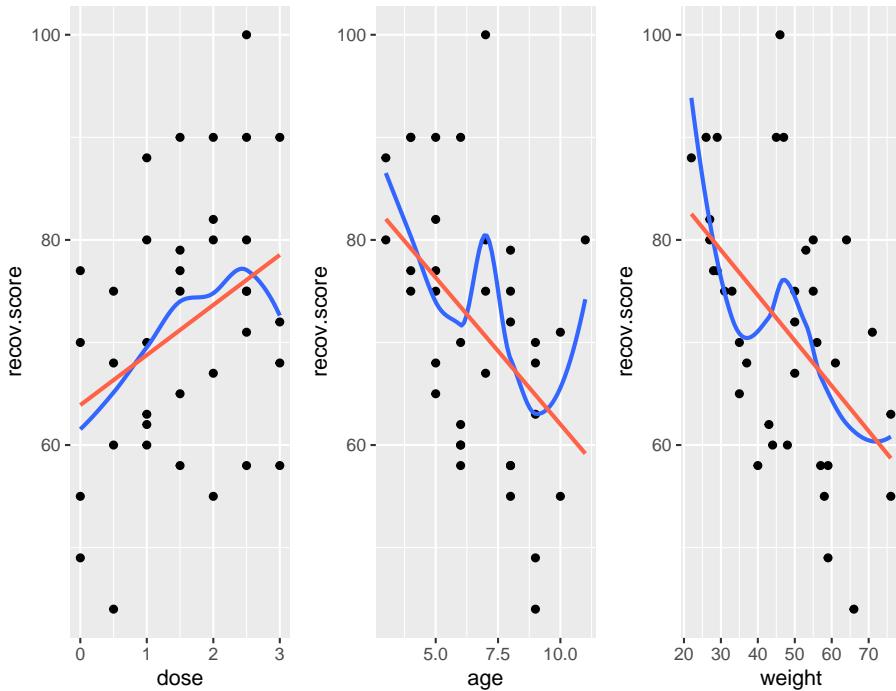
- By *linear*, we mean that each predictor value, x , appears simply multiplied by its coefficient and added to the model.
- No x appears in an exponent or some other more complicated function.
- If the regression model is true, then the outcome y is linearly related to each of the x 's.

Unfortunately, assuming the model is true is not sufficient to prove that the linear model fits, so we check what Bock, Velleman, and De Veaux (2004) call the “Straight Enough Condition”

36.2.1 Initial Scatterplots for the “Straight Enough” Condition

- Scatterplots of y against each of the predictors are reasonably straight.
- The scatterplots need not show a strong (or any!) slope; we just check that there isn't a bend or other nonlinearity.
- Any substantial curve is indicative of a potential problem.
- Modest bends are not usually worthy of serious attention.

For example, in the `hydrate` data, here are the relevant scatterplots (in practice, I would simply look at the scatterplot matrix produced earlier.)



Here, I've simply placed `recov.score` on the vertical (Y) axis, plotted against each of the predictors, in turn. I've added a straight line (OLS) fit [in tomato red] and a loess smooth [in blue] to each plot to guide your assessment of the "straight enough" condition.

- Each of these is "straight enough" for our purposes, in initially fitting the data.
- If one of these was not, we might consider a transformation of the relevant predictor, or, if all were problematic, we might transform the outcome Y.

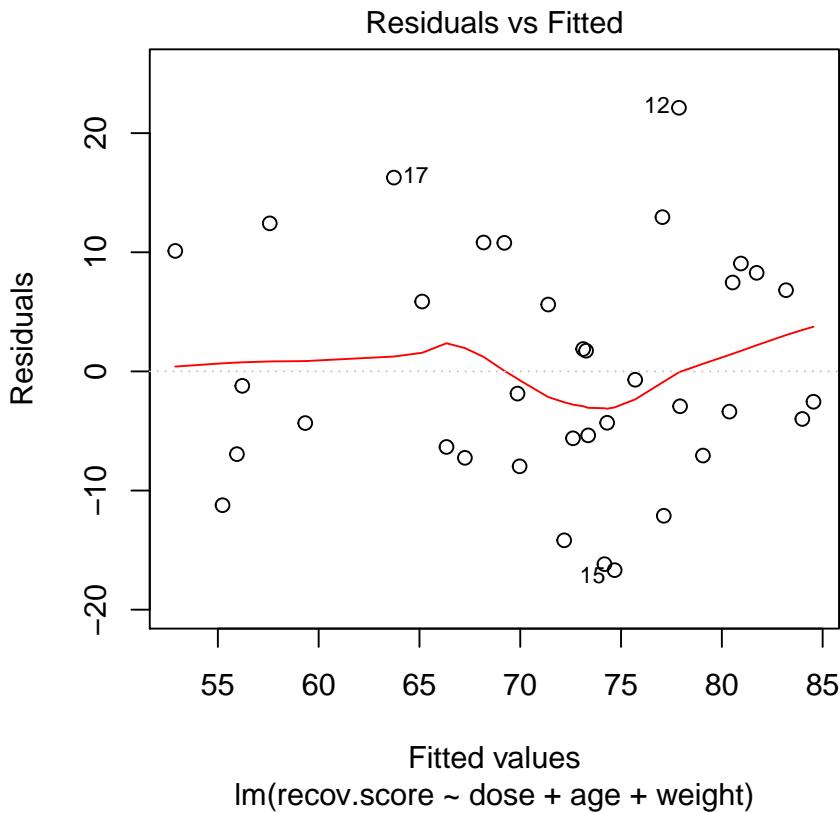
36.2.2 Residuals vs. Predicted Values to Check for Non-Linearity

The residuals should appear to have no pattern (no curve, for instance) with respect to the predicted (fitted) values. It is a very good idea to plot the residuals against the fitted values to check for patterns, especially bends or other indications of non-linearity. For a multiple regression, the fitted values are a combination of the x's given by the regression equation, so they combine the effects of the x's in a way that makes sense for our particular regression model. That makes them a good choice to plot against. We'll check for other things in this plot, as well.

When you ask R to plot the result of a linear model, it will produce up to five

separate plots: the first of which is a plot of **residuals vs. fitted values**. To obtain this plot for the model including `dose`, `age` and `weight` that predicts `recov.score`, we indicate plot 1 using the `which` command within the `plot` function:

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=1)
```



The loess smooth is again added to help you identify serious non-linearity. In this case, I would conclude that there were no serious problems with linearity in these data.

The plot also, by default, identifies the three values¹ with the largest (in absolute value) residuals.

- Here, these are rows 12, 15 and 17, where 12 and 17 have positive residuals

¹If you wanted to identify more or fewer points, you could, i.e. 'plot(modelname, which=1, id.n = 2)'

(i.e. they represent under-predictions by the model) and 15 has a negative residual (i.e. it represents a situation where the prediction was larger than the observed recovery score.)

36.2.3 Residuals vs. Predictors To Further Check for Non-Linearity

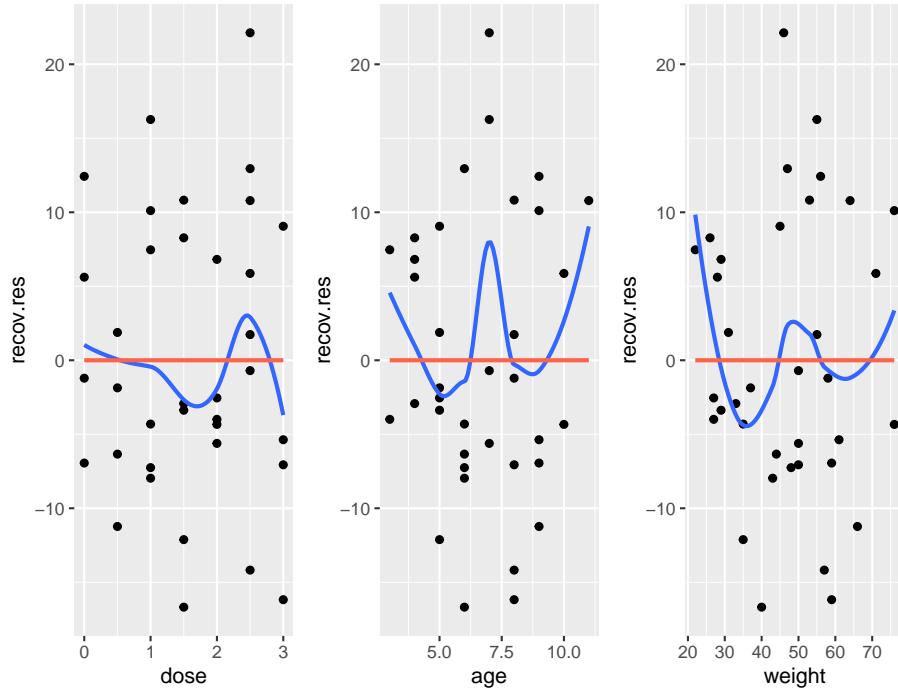
If we do see evidence of non-linearity in the plot of residuals against fitted values, I usually then proceed to look at residual plots against each of the individual predictors, in turn, to try to identify the specific predictor (or predictors) where we may need to use a transformation.

The appeal of such plots (as compared to the initial scatterplots we looked at of the outcome against each predictor) is that they eliminate the distraction of the linear fit, and let us look more closely at the non-linear part of the relationship between the outcome and the predictor.

Although I don't think you need them here, here are these plots for the residuals from this model.

```
hydrate$recov.res <- residuals(lm(recov.score ~ dose + age + weight, data = hydrate))

p1 <- ggplot(hydrate, aes(x = dose, y = recov.res)) +
  geom_point() + geom_smooth(method = "loess", se = FALSE) +
  geom_smooth(method = "lm", se = FALSE, col = "tomato")
p2 <- ggplot(hydrate, aes(x = age, y = recov.res)) +
  geom_point() + geom_smooth(method = "loess", se = FALSE) +
  geom_smooth(method = "lm", se = FALSE, col = "tomato")
p3 <- ggplot(hydrate, aes(x = weight, y = recov.res)) +
  geom_point() + geom_smooth(method = "loess", se = FALSE) +
  geom_smooth(method = "lm", se = FALSE, col = "tomato")
gridExtra::grid.arrange(p1, p2, p3, nrow=1)
```



Again, I see no particularly problematic issues in the scatterplot here. The curve in the `age` plot is a bit worrisome, but it still seems pretty modest, with the few values of 3 and 11 for `age` driving most of the “curved” pattern we see: there’s no major issue. If we’re willing to assume that the multiple regression model is correct in terms of its specification of a linear model, we can move on to assessing other assumptions and conditions.

36.3 The Independence Assumption

The errors in the true underlying regression model must be mutually independent, but there is no way to be sure that the independence assumption is true. Fortunately, although there can be many predictor variables, there is only one outcome variable and only one set of errors. The independence assumption concerns the errors, so we check the residuals.

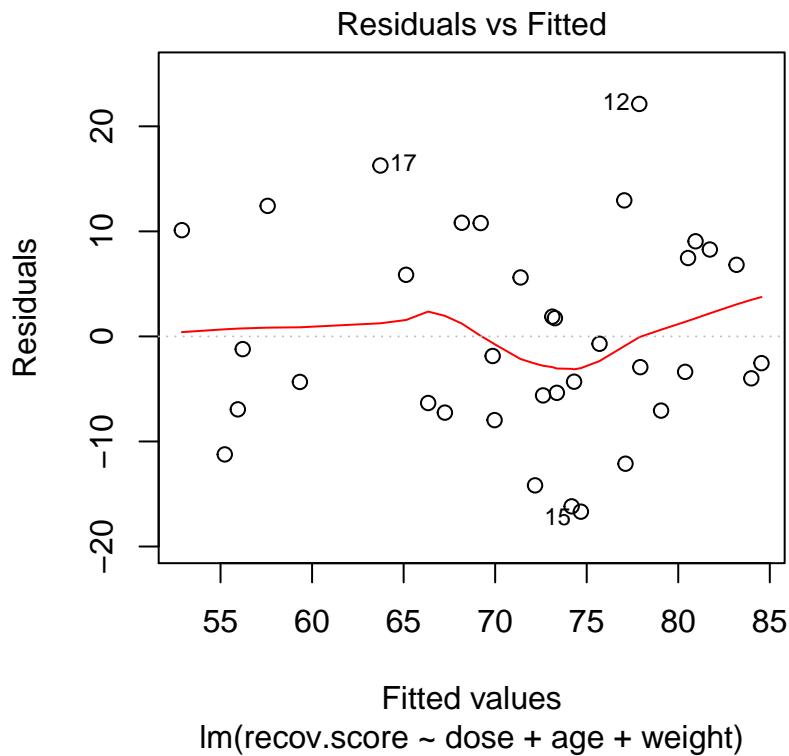
Randomization condition. The data should arise from a random sample, or from a randomized experiment.

- The residuals should appear to be randomly scattered and show no patterns, trends or clumps when plotted against the predicted values.
- In the special case when an x-variable is related to time, make sure that the residuals do not have a pattern when plotted against time.

36.3.1 Residuals vs. Fitted Values to Check for Dependence

The `hydrate` children were not related in any way and were randomly assigned to dosages, so we can be pretty sure that their measurements are independent. The residuals vs. fitted values plot shows no clear trend, cycle or pattern which concerns us.

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=1)
```



36.4 The Constant Variance Assumption

The variability of our outcome, y , should be about the same for all values of every predictor x . Of course, we can't check every combination of x values, so we look at scatterplots and check to see if the plot shows a “fan” shape - in essence, the **Does the plot thicken? Condition**.

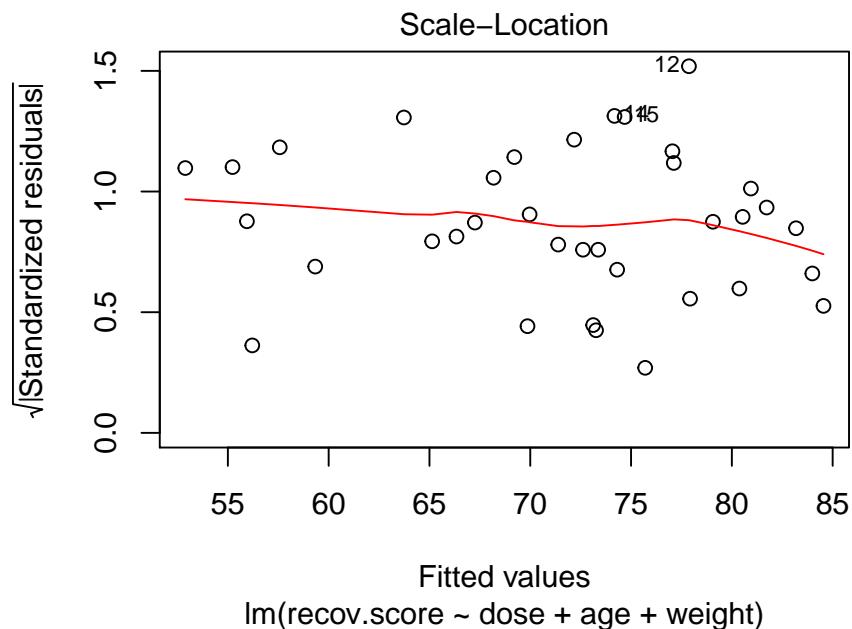
- Scatterplots of residuals against each x or against the predicted values, offer a visual check.
- Be alert for a “fan” or a “funnel” shape showing growing/shrinking variability in one part of the plot.

Reviewing the same plots we have previously seen, I can find no evidence of substantial “plot thickening” in either the plot of residuals vs. fitted values or in any of the plots of the residuals against each predictor separately.

36.4.1 The Scale-Location Plot to Check for Non-Constant Variance

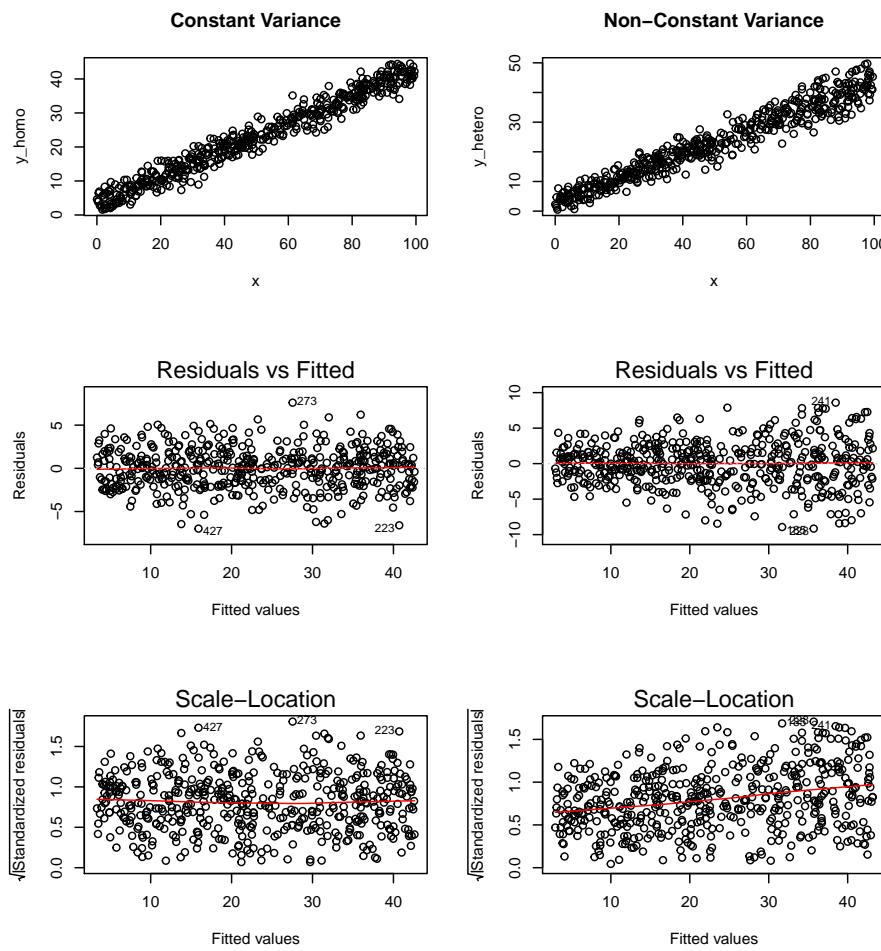
R does provide an additional plot to help assess this issue as linear model diagnostic plot 3. This one looks at the square root of the standardized residual plotted against the fitted values. You want the loess smooth in this plot to be flat, rather than sloped.

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=3)
```



36.4.2 Assessing the Equal Variances Assumption

It's helpful to see how this works in practice. Here are three sets of plots for two settings, where the left plots in each pair show simulated data that are homoscedastic (variance is constant across the levels of the predictor) and the right plots show data that are heteroscedastic (variance is not constant across predictor levels.)



Note the funnel shape for the upper two heteroscedastic plots, and the upward sloping loess line in the last one. *Source:* <http://goo.gl/weMI0U> [from stats.stackexchange.com]

36.5 The Normality Assumption

If the plot is straight enough, the data are independent, and the plots don't thicken, you can now move on to the final assumption, that of Normality.

We assume that the errors around the idealized regression model at any specified values of the x-variables follow a Normal model. We need this assumption so that we can use a Student's t-model for inference. As with other times when we've used Student's t, we'll settle for the residuals satisfying the **Nearly Normal condition**. To assess this, we simply look at a histogram or Normal probability plot of the residuals. Note that the Normality Assumption also becomes less important as the sample size grows.

36.6 Outlier Diagnostics: Points with Unusual Residuals

A multiple regression model will always have a point which displays the most unusual residual (that is, the residual furthest from zero in either a positive or negative direction.) As part of our assessment of the normality assumption, we will often try to decide whether the residuals follow a Normal distribution by creating a Q-Q plot of standardized residuals.

36.6.1 Standardized Residuals

Standardized residuals are scaled to have mean zero and a constant standard deviation of 1, as a result of dividing the original (raw) residuals by an estimate of the standard deviation that uses all of the data in the data set.

The `rstandard` function, when applied to a linear regression model, will generate the standardized residuals, should we want to build a histogram or other plot of the standardized residuals.

If multiple regression assumptions hold, then the standardized residuals (in addition to following a Normal distribution in general) will also have mean zero and standard deviation 1, with approximately 95% of values between -2 and +2, and approximately 99.74% of values between -3 and +3.

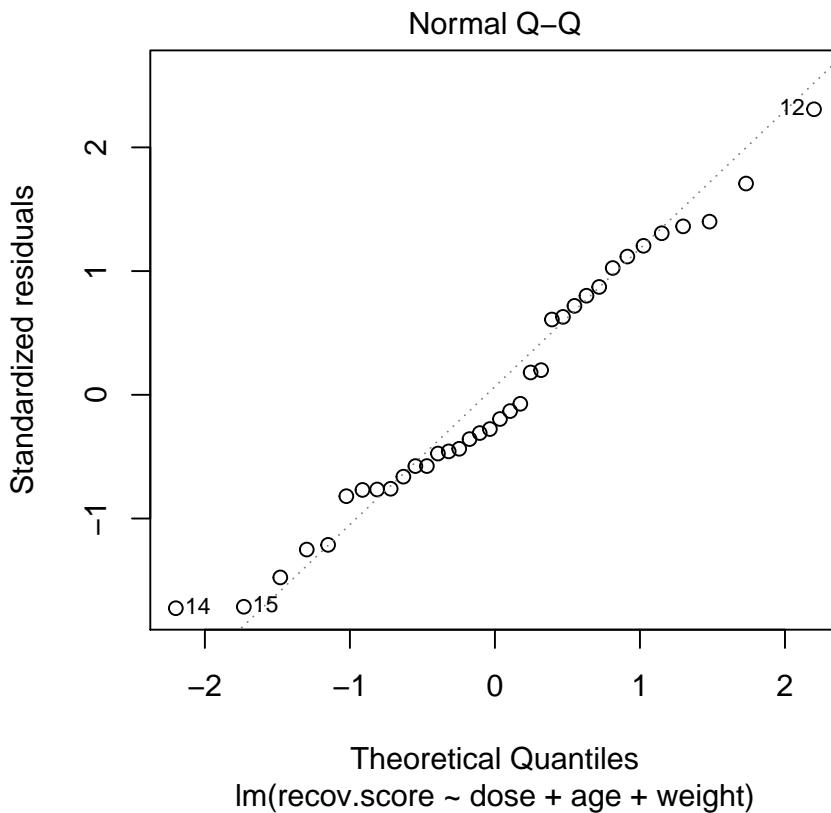
A natural check, therefore, will be to identify the most extreme/unusual standardized residual in our data set, and see whether its value is within the general bounds implied by the Empirical Rule associated with the Normal distribution. If, for instance, we see a standardized residual below -3 or above +3, this is likely to be a very poorly fit point (we would expect to see a point like this about 2.6 times for every 1,000 observations.)

A very poorly fitting point, especially if it also has high influence on the model (a concept we'll discuss shortly), may be sufficiently different from the rest of the points in the data set to merit its removal before modeling. If we did remove such a point, we'd have to have a good reason for this exclusion, and include that explanation in our report.

R's general plot set for a linear model includes (as plot 2) a Normal Q-Q plot of the standardized residuals from the model.

36.6.2 Checking the Normality Assumption with a Plot

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=2)
```



In the `hydrate` data, consider our full model with all three predictors. The Q-Q plot of standardized residuals shows the most extreme three cases are in rows 12,

14 and 15. We see from the vertical axis here that none of these points are as large as 3 in absolute value, and only row 12 appears to be above +2 in absolute value.

36.6.3 Assessing the Size of the Standardized Residuals

I'll begin by once again placing the details of my linear model into a variable called `modela`. Next, I'll extract, round (to two decimal places) and then sort (from lowest to highest) the standardized residuals from `modela`.

```
modela <- lm(recov.score ~ dose + age + weight, data = hydrate)
sort(round(rstandard(modela), 2))
```

14	15	30	2	19	20	9	10	21	33	11	25
-1.73	-1.71	-1.48	-1.25	-1.21	-0.82	-0.77	-0.77	-0.76	-0.66	-0.58	-0.58
18	7	28	23	13	6	32	16	22	3	5	1
-0.47	-0.46	-0.44	-0.36	-0.31	-0.28	-0.20	-0.13	-0.07	0.18	0.20	0.61
26	36	31	27	34	35	4	24	8	29	17	12
0.63	0.72	0.80	0.87	1.02	1.12	1.20	1.31	1.36	1.40	1.71	2.31

We can see that the smallest residual is -1.73 (in row 14) and the largest is 2.31 (in row 12). Another option would be to look at these rows in terms of the absolute value of their standardized residuals...

```
sort(abs(round(rstandard(modela), 2)))
```

22	16	3	5	32	6	13	23	28	7	18	11	25	1	26
0.07	0.13	0.18	0.20	0.20	0.28	0.31	0.36	0.44	0.46	0.47	0.58	0.58	0.61	0.63
33	36	21	9	10	31	20	27	34	35	4	19	2	24	8
0.66	0.72	0.76	0.77	0.77	0.80	0.82	0.87	1.02	1.12	1.20	1.21	1.25	1.31	1.36
29	30	15	17	14	12									
1.40	1.48	1.71	1.71	1.73	2.31									

36.6.4 Assessing Standardized Residuals with an Outlier Test

Is a standardized residual of 2.31 particularly unusual in a sample of 36 observations from a Normal distribution, supposedly with mean 0 and standard deviation 1?

No. The `car` library has a test called `outlierTest` which can be applied to a linear model to see if the most unusual studentized residual in absolute value is a surprise given the sample size (the studentized residual is very similar to the standardized residual - it simply uses a different estimate for the standard deviation for every point, in each case excluding the point it is currently studying)

```
outlierTest(lm(recov.score ~ dose + age + weight, data = hydrate))
```

```
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferroni p
12      2.49          0.0184      0.663
```

Our conclusion is that there's no serious problem with normality in this case, and no outliers with studentized (or for that matter, standardized) residuals outside the range we might reasonably expect given a Normal distribution of errors and 36 observations.

36.7 Outlier Diagnostics: Identifying Points with Unusually High Leverage

An observation can be an outlier not just in terms of having an unusual residual, but also in terms of having an unusual combination of predictor values. Such an observation is described as having high leverage in a data set.

- R will calculate leverage for the points included in a regression model using the `hatvalues` function as applied to the model.
- The average leverage value is equal to k/n , where n is the number of observations included in the regression model, and k is the number of coefficients (slopes + intercept).
- Any point with a leverage value greater than 2.5 times the average leverage of a point in the data set is one that should be investigated closely.

For instance, in the `hydrate` data, we have 36 observations, so the average leverage will be $4/36 = 0.111$ and a high leverage point would have leverage $\geq 10/36$ or .278. We can check this out directly, with a sorted list of leverage values...

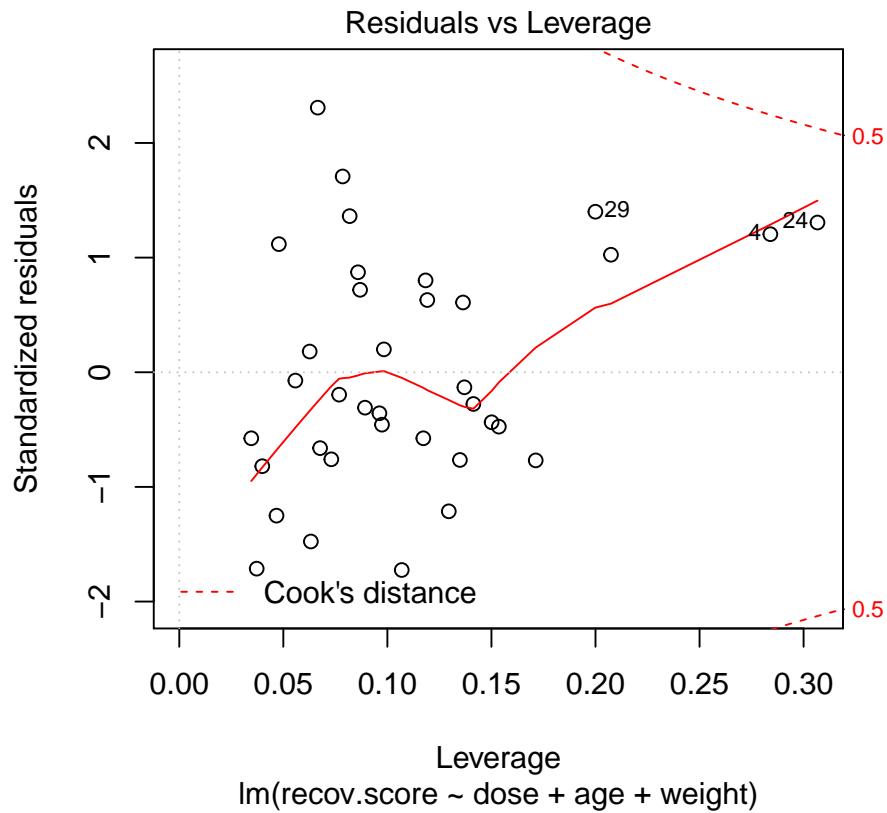
```
sort(round(hatvalues(lm(recov.score ~ dose + age + weight, data=hydrate)),3))
```

11	15	20	2	35	22	3	30	12	33	21	32
0.035	0.037	0.040	0.047	0.048	0.056	0.063	0.063	0.067	0.068	0.073	0.077
17	8	27	36	13	23	7	5	14	25	31	26
0.078	0.082	0.086	0.087	0.089	0.096	0.097	0.098	0.107	0.117	0.118	0.119
19	10	1	16	6	28	18	9	29	34	4	24
0.130	0.135	0.136	0.137	0.141	0.150	0.154	0.171	0.200	0.207	0.284	0.307

Two points - 4 and 24 - exceed our cutoff based on having more than 2.5 times the average leverage.

Or we can look at a plot of residuals vs. leverage, which is diagnostic plot 5 for a linear model.

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=5)
```



We see that the most highly leveraged points (those furthest to the right in this plot) are in rows 4 and 24. They are also the two points that meet our 2.5 times the average standard.

To see why points 4 and 24 are the most highly leveraged points, we remember that this just means that these points are unusual in terms of their predictor values.

```
hydrate %>%
  filter(id %in% c(4, 24)) %>%
  dplyr::select(id, dose, age, weight)

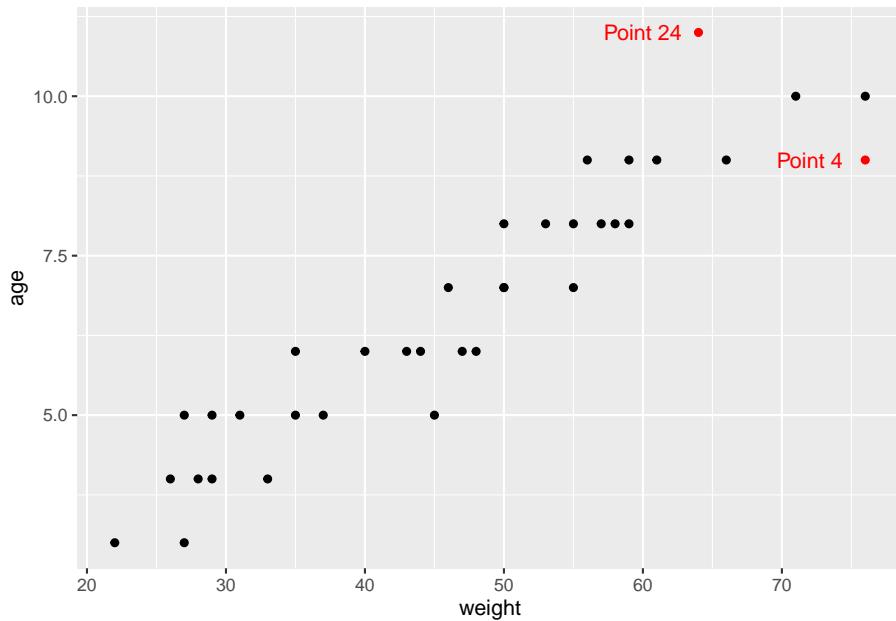
# A tibble: 2 x 4
  id   dose    age  weight
  <dbl> <dbl> <dbl> <dbl>
1     4 0.045  0.04  0.05
2    24 0.245  0.24  0.25
```

```

<int> <dbl> <int> <int>
1     4    1     9    76
2    24   2.5    11    64
# create indicator for high leverage points
hydrate$hilev <- ifelse(hydrate$id %in% c(4,24), "Yes", "No")

ggplot(hydrate, aes(x = weight, y = age, color = hilev)) +
  geom_point() +
  scale_color_manual(values = c("black", "red")) +
  guides(color = FALSE) +
  annotate("text", x = 60, y = 11, label = "Point 24", col = "red") +
  annotate("text", x = 72, y = 9, label = "Point 4", col = "red")

```



36.8 Outlier Diagnostics: Identifying Points with High Influence on the Model

A point in a regression model has high **influence** if the inclusion (or exclusion) of that point will have substantial impact on the model, in terms of the estimated coefficients, and the summaries of fit. The measure I routinely use to describe the level of influence a point has on the model is called **Cook's distance**.

As a general rule, a Cook's distance greater than 1 indicates a point likely to have substantial influence on the model, while a point in the 0.5 to 1.0 range is

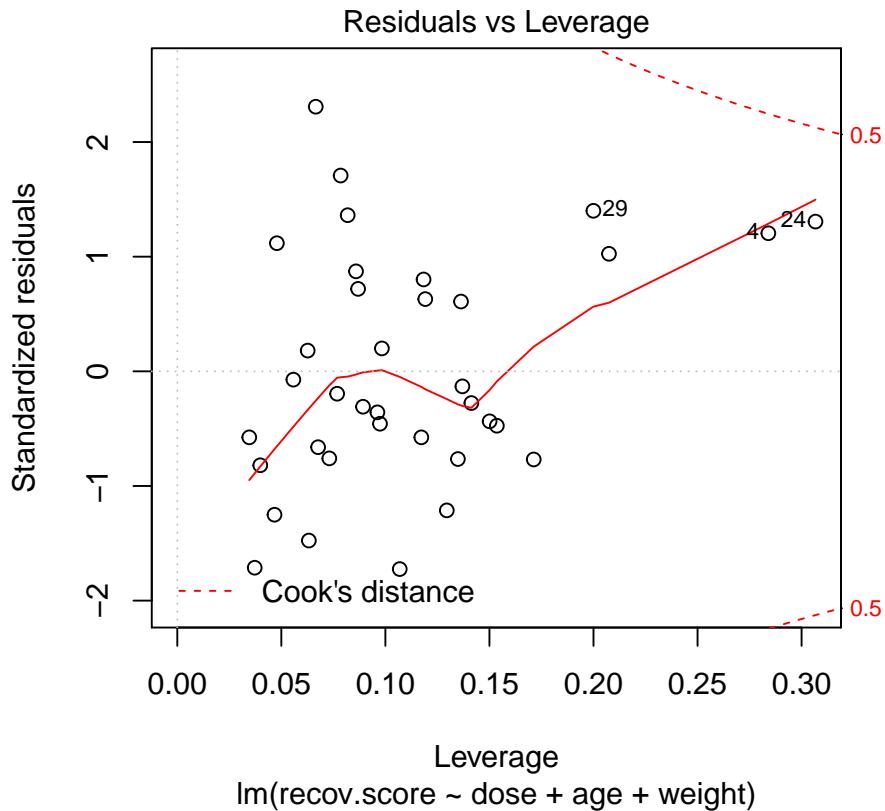
only occasionally worthy of investigation. Observations with Cook's distance below 0.5 are unlikely to influence the model in any substantial way.

36.8.1 Assessing the Value of Cook's Distance

You can obtain information on Cook's distance in several ways in R.

- My favorite is model diagnostic plot 5 (the residuals vs. leverage plot) which uses contours to indicate the value of Cook's distance.
 - This is possible because influence, as measured by Cook's distance, is a function of both the observation's standardized residual and its leverage.

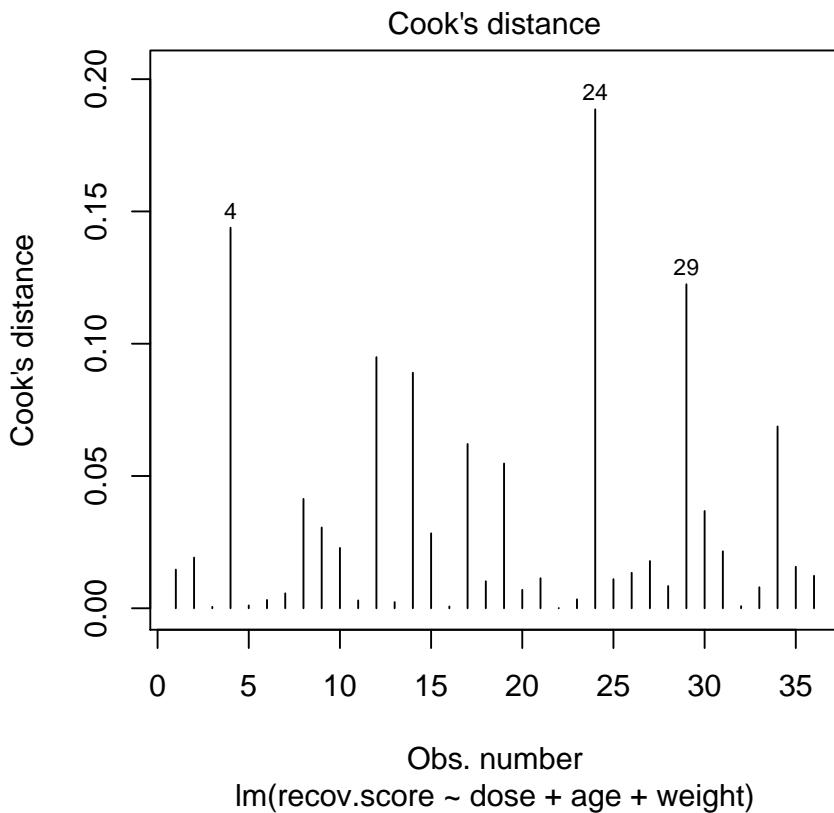
```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=5)
```



36.8.2 Index Plot of Cook's Distance

Model Diagnostic Plot 4 for a linear model is an index plot of Cook's distance.

```
plot(lm(recov.score ~ dose + age + weight, data = hydrate), which=4)
```



It is clear from this plot that the largest Cook's distance (somewhere between 0.15 and 0.2) is for the observation in row 24 of the data set.

To see all of the Cook's distances, we can simply ask for them using the `cooks.distance` function.

```
sort(round(cooks.distance(lm(recov.score ~ dose + age + weight, data=hydrate)),3))
```

22	3	5	16	32	13	6	11	23	7	20	28
0.000	0.001	0.001	0.001	0.001	0.002	0.003	0.003	0.003	0.006	0.007	0.008
33	18	21	25	36	26	1	35	27	2	31	10

```
0.008 0.010 0.011 0.011 0.012 0.013 0.015 0.016 0.018 0.019 0.022 0.023
    15      9     30      8     19     17     34     14     12     29      4     24
0.028 0.031 0.037 0.041 0.055 0.062 0.069 0.089 0.095 0.122 0.144 0.189
```

36.9 Running a Regression Model While Excluding A Point

Suppose that we wanted to remove row 24, the point with the most influence over the model. We could fit a model to the data without row 24 to see the impact...

Note that I have to specify the new data set as `hydrate[-24,]`: **the comma is often forgotten and crucial**.

```
summary(lm(recov.score ~ dose + age + weight, data=hydrate[-24,]))
```

```
Call:
lm(formula = recov.score ~ dose + age + weight, data = hydrate[-24,
])

Residuals:
    Min      1Q  Median      3Q      Max 
-16.07  -5.61  -2.33   6.94  23.62 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  87.798     6.153   14.27 3.6e-15 ***
dose         5.824     1.790    3.25  0.0027 **  
age        -1.413     2.596   -0.54  0.5901    
weight       -0.350     0.352   -0.99  0.3279    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.81 on 31 degrees of freedom
Multiple R-squared:  0.481, Adjusted R-squared:  0.431 
F-statistic: 9.57 on 3 and 31 DF,  p-value: 0.000126
```

Compare these results to those obtained with the full data set, shown below.

- How is the model affected by removing point 24?
- What is the impact on the slopes?
- On the summary measures? Residuals?

```
summary(lm(recov.score ~ dose + age + weight, data=hydrate))
```

```

Call:
lm(formula = recov.score ~ dose + age + weight, data = hydrate)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.68   -6.49   -2.20    7.67   22.13 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  85.476     5.965   14.33  1.8e-15 ***
dose         6.170     1.791    3.45   0.0016 **  
age          0.277     2.285    0.12   0.9043    
weight       -0.543     0.324   -1.68   0.1032    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.92 on 32 degrees of freedom
Multiple R-squared:  0.458, Adjusted R-squared:  0.408 
F-statistic: 9.03 on 3 and 32 DF,  p-value: 0.000177

```

While it is true that we can sometimes improve the performance of the model in some ways by removing this point, there's no good reason to do so. We can't just remove a point from the data set without a good reason (and, to be clear, "I ran my model and it doesn't fit this point well" is NOT a good reason). Good reasons would include:

- This observation was included in the sample in error, for instance, because the subject was not eligible.
- An error was made in transcribing the value of this observation to the final data set.
- And, sometimes, even "This observation is part of a meaningful subgroup of patients that I had always intended to model separately..." assuming that's true.

36.10 Summarizing Regression Diagnostics for 431

1. Check the "straight enough" condition with scatterplots of the y variable (outcome) against each x-variable (predictor), usually via the top row of a scatterplot matrix.
2. If the data are straight enough (that is, if it looks like the regression model is plausible), fit a regression model, to obtain residuals and influence measures.

- If not, consider using the Box-Cox approach to identify a possible transformation for the outcome variable, and then recheck the straight enough condition.

The **plot** function for a fitted linear model builds five diagnostic plots.

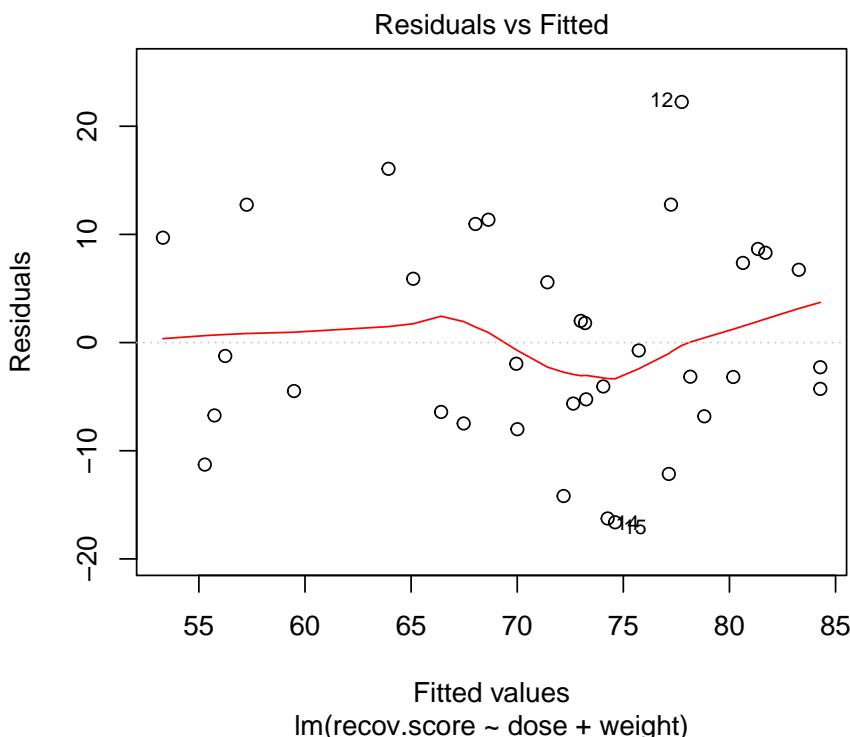
3. [Plot 1] A scatterplot of the residuals against the fitted values.
 - This plot should look patternless. Check in particular for any bend (which would suggest that the data weren't all that straight after all) and for any thickening, which would indicate non-constant variance.
 - If the data are measured over time, check especially for evidence of patterns that might suggest they are not independent. For example, plot the residuals against time to look for patterns.
4. [Plot 3] A scale-location plot of the square root of the standardized residuals against the fitted values to look for a non-flat loess smooth, which indicates non-constant variance. Standardized residuals are obtained via the `rstandard(model)` function.
5. [Plot 2] If the plots above look OK, then consider a Normal Q-Q plot of the standardized residuals to check the nearly Normal condition.
6. [Plot 5] The final plot we often look at is the plot of residuals vs. leverage, with influence contours. Sometimes, we'll also look at [Plot 4] the index plot of the Cook's distance for each observation in the data set.
 - To look for points with substantial **leverage** on the model by virtue of having unusual values of the predictors - look for points whose leverage is at least 2.5 times as large as the average leverage value.
 - The average leverage is always k/n , where k is the number of coefficients fit by the model (including the slopes and intercept), and n is the number of observations in the model.
 - To obtain the leverage values, use the `hatvalues(model)` function.
 - To look for points with substantial **influence** on the model, that is, removing them from the model would change it substantially, consider the Cook's distance, plotted in contours in Plot 5, or in an index plot in Plot 4.
 - Any Cook's $d > 1$ will likely have a substantial impact on the model.
 - Even points with Cook's $d > 0.5$ may merit further investigation.
 - Find the Cook's distances using the `cooks.distance(model)` function.

36.11 Back to hydrate: Residual Diagnostics for Dose + Weight Model

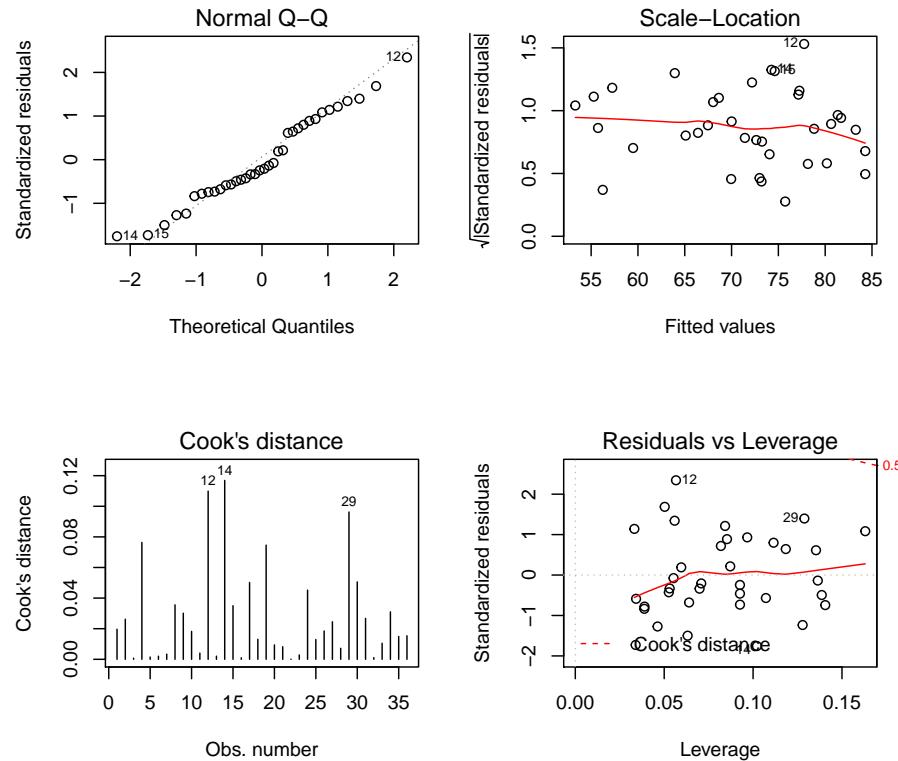
In class, we'll walk through all five plots produced for the DW model (the model for hydrate recovery score using dose and weight alone).

1. What do these 5 plots tell us about the assumption of linearity?
2. What do these 5 plots tell us about the assumption of independence?
3. What do these 5 plots tell us about the assumption of constant variance?
4. What do these 5 plots tell us about the assumption of Normality?
5. What do these 5 plots tell us about leverage of individual observations?
6. What do these 5 plots tell us about influence of individual observations?

```
plot(lm(recov.score ~ dose + weight, data = hydrate), which=1)
```



```
par(mfrow=c(2,2))
plot(lm(recov.score ~ dose + weight, data = hydrate), which=2:5)
```



36.12 Violated Assumptions: Problematic Residual Plots?

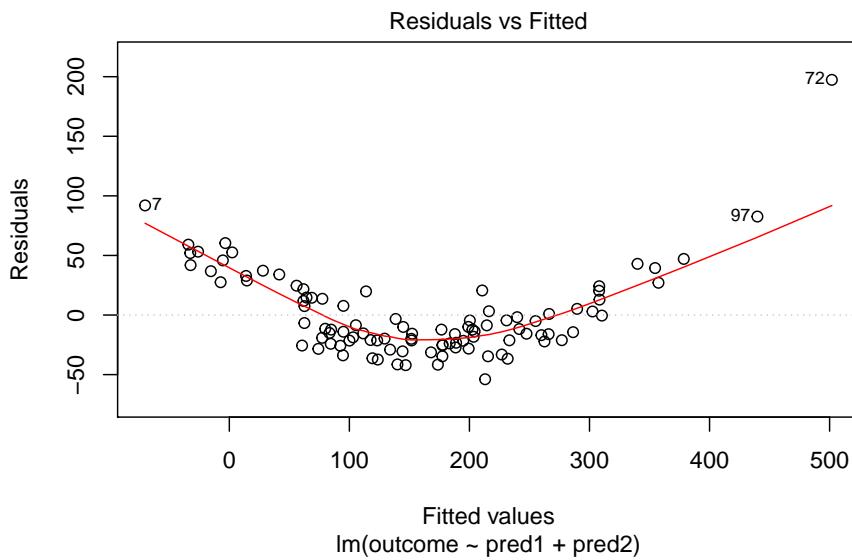
So what do serious assumption violations look like, and what can we do about them?

36.13 Problems with Linearity

Here is a simulated example that shows a clear problem with non-linearity.

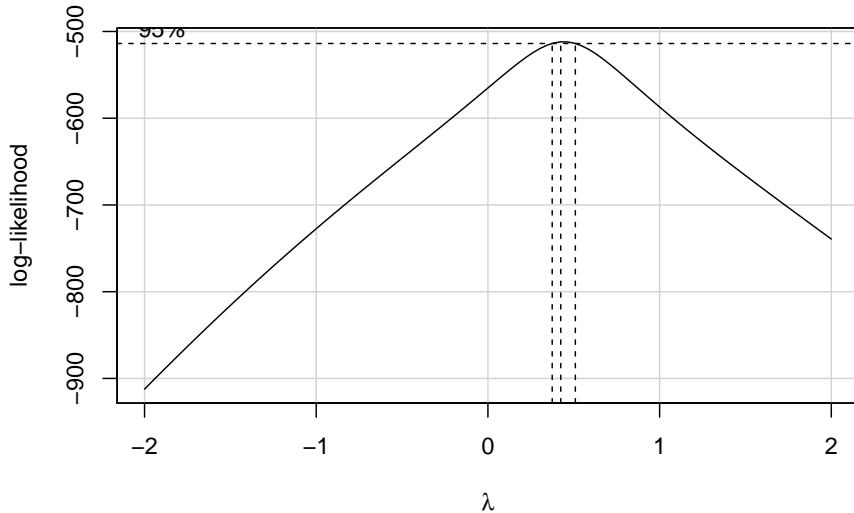
```
set.seed(4311); x1 <- rnorm(n = 100, mean = 15, sd = 5)
set.seed(4312); x2 <- rnorm(n = 100, mean = 10, sd = 5)
set.seed(4313); e1 <- rnorm(n = 100, mean = 0, sd = 15)
y <- 15 + x1 + x2^2 + e1
viol1 <- data.frame(outcome = y, pred1 = x1, pred2 = x2) %>% tbl_df
```

```
model.1 <- lm(outcome ~ pred1 + pred2, data = viol1)
plot(model.1, which = 1)
```



In light of this, I would be looking for a potential transformation of outcome.
Does the Box-Cox plot make any useful suggestions?

```
boxCox(model.1); powerTransform(model.1)
```



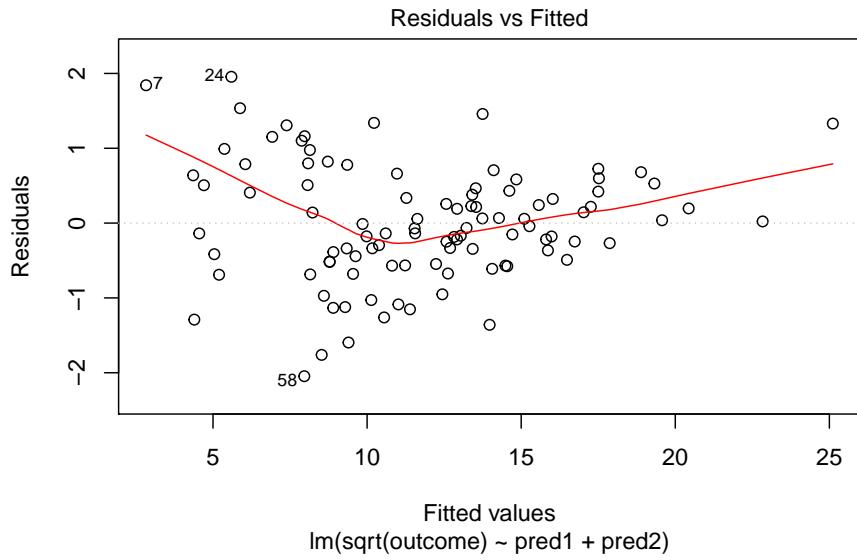
```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
argument ignored
```

Estimated transformation parameter

```
Y1  
0.441
```

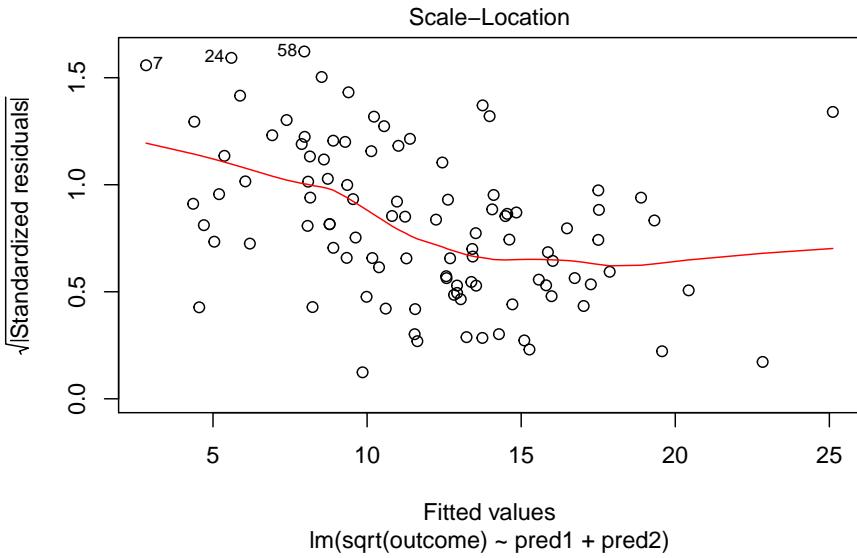
Note that if the outcome was negative, we would have to add some constant value to every outcome in order to get every outcome value to be positive, and Box-Cox to run. This suggests fitting a new model, using the square root of the outcome.

```
model.2 <- lm(sqrt(outcome) ~ pred1 + pred2, data = viol1)
plot(model.2, which = 1)
```



This is meaningfully better in terms of curve, but now looks a bit fan-shaped, indicating a potential problem with heteroscedasticity. Let's look at the scale-location plot for this model.

```
plot(model.2, which = 3)
```

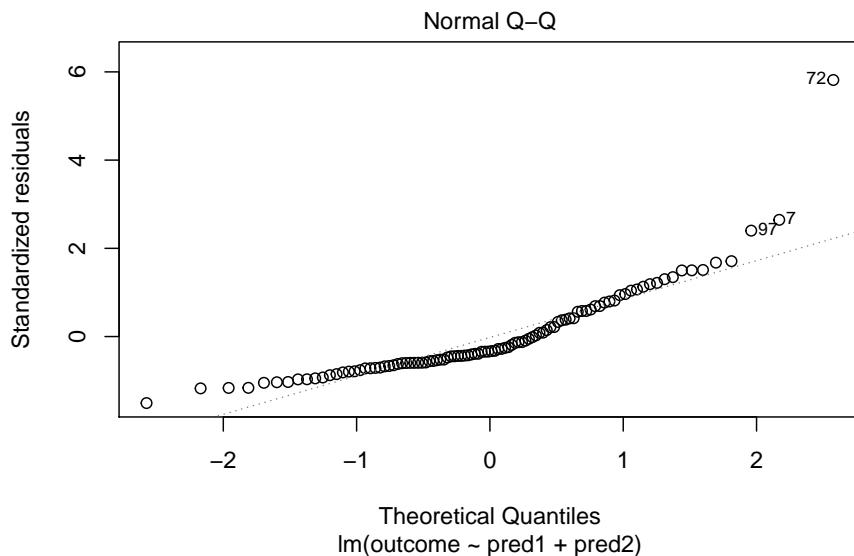


This definitely looks like there's a trend down in this plot. So the square root transformation, by itself, probably hasn't resolved assumptions sufficiently well. We'll have to be very careful about our interpretation of the model.

36.14 Problems with Non-Normality: An Influential Point

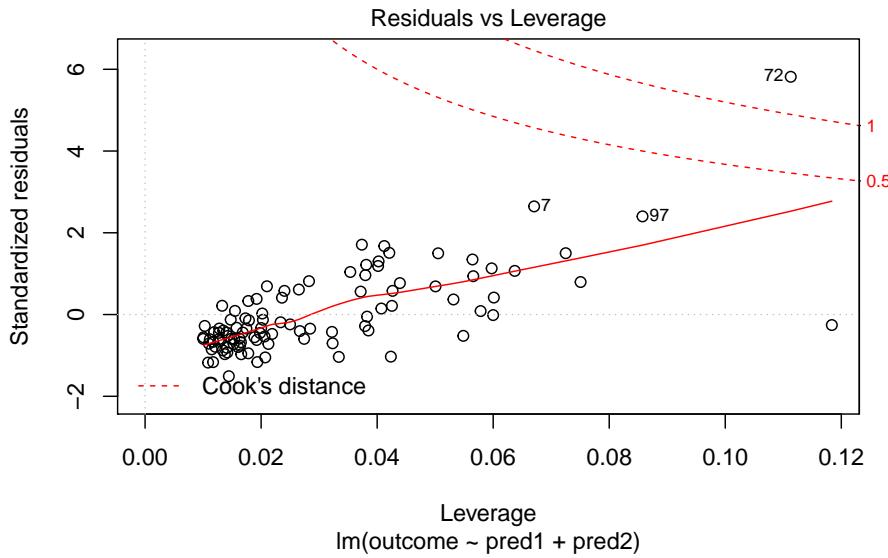
With 100 observations, a single value with a standardized residual above 3 is very surprising. In our initial `model.1` here, we have a standardized residual value as large as 6, so we clearly have a problem with that outlier.

```
plot(model.1, which = 2)
```



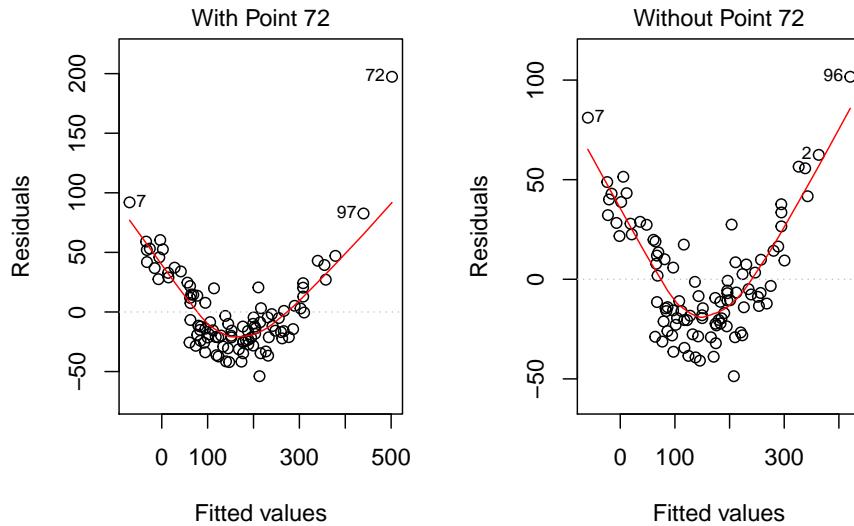
Should we, perhaps, remove point 72, and try again? Only if we have a reason beyond “it was poorly fit” to drop that point. Is point 72 highly leveraged or influential?

```
plot(model.1, which = 5)
```



What if we drop this point (72) and fit our linear model again. Does this resolve our difficulty with the assumption of linearity?

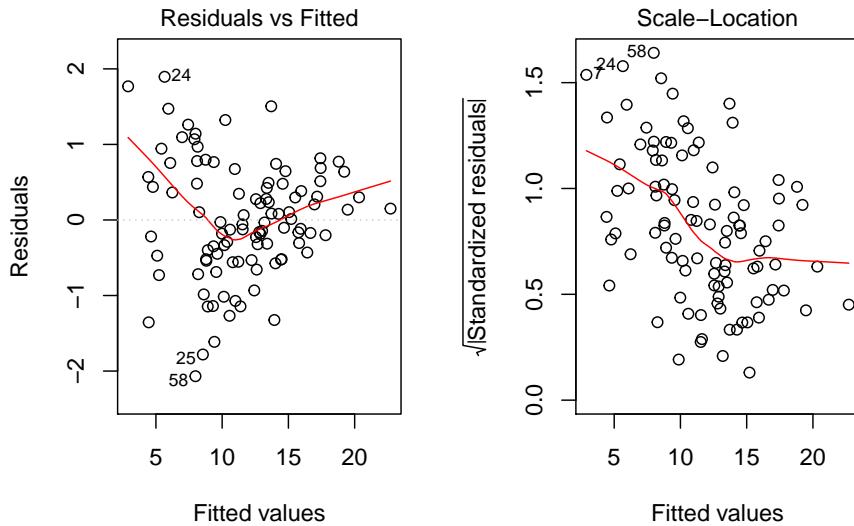
```
model.1.no72 <- lm(outcome ~ pred1 + pred2, data = viol1[-72,])
par(mfrow=c(1,2))
plot(model.1, which = 1, caption = "With Point 72")
plot(model.1.no72, which = 1, caption = "Without Point 72")
```



```
par(mfrow=c(1,1))
```

No, it doesn't. But what if we combine our outcome transformation with dropping point 72?

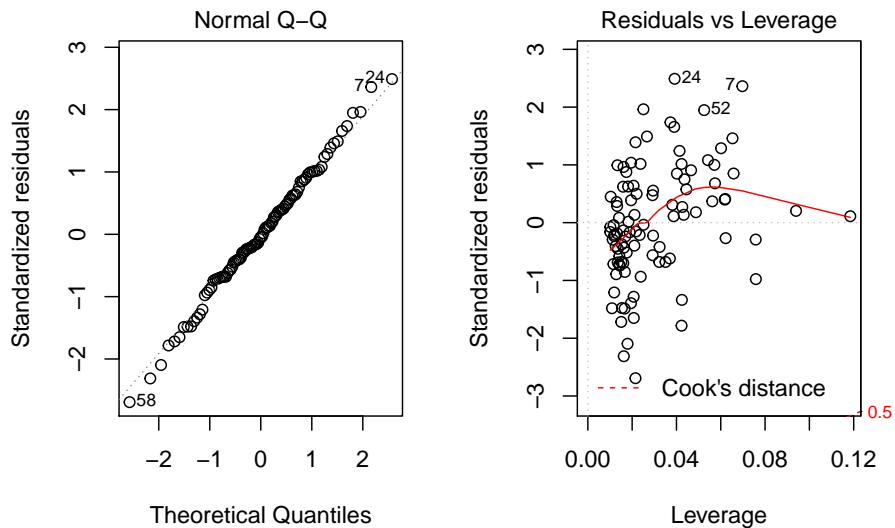
```
model.2.no72 <- lm(sqrt(outcome) ~ pred1 + pred2, data = viol1[-72,])
par(mfrow=c(1,2))
plot(model.2.no72, which = c(1,3))
```



```
par(mfrow=c(1,1))
```

Nope. That still doesn't alleviate the problem of heteroscedasticity very well. At least, we no longer have any especially influential points, nor do we have substantial non-Normality.

```
par(mfrow=c(1,2))
plot(model.2.no72, which = c(2,5))
```

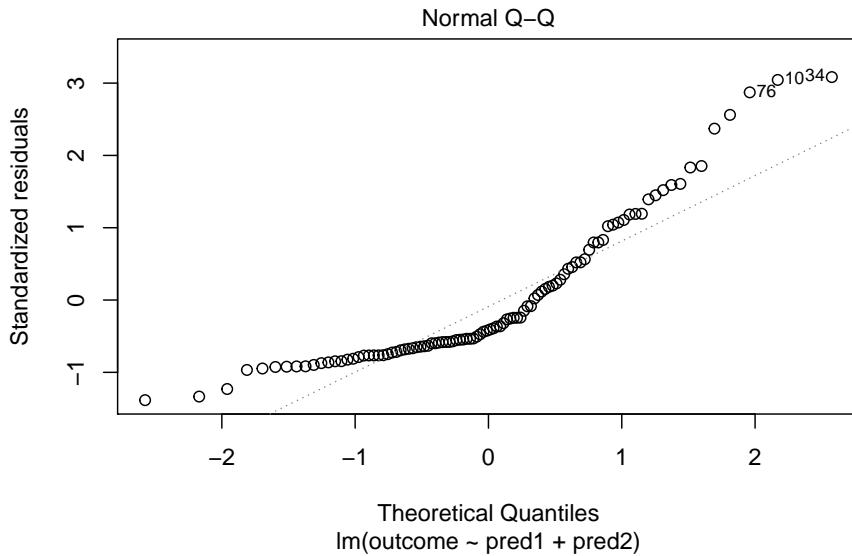


```
par(mfrow=c(1,1))
```

At this point, I would be considering potential transformations of the predictors, quite possibly fitting some sort of polynomial term or cubic spline term in the predictors, but I'll leave that for discussion in 432.

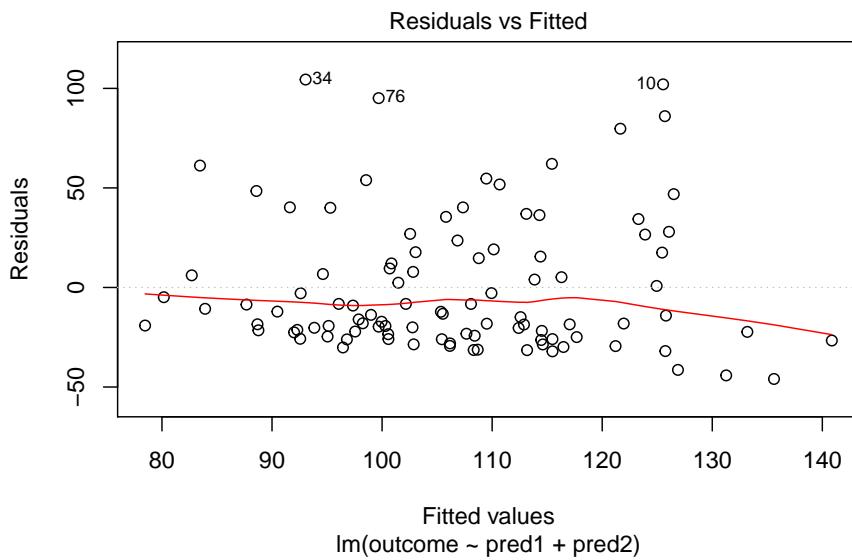
36.15 Problems with Non-Normality: Skew

```
set.seed(4314); x1 <- rnorm(n = 100, mean = 15, sd = 5)
set.seed(4315); x2 <- rnorm(n = 100, mean = 10, sd = 5)
set.seed(4316); e2 <- rnorm(n = 100, mean = 3, sd = 5)
y2 <- 50 + x1 + x2 + e2^2
viol2 <- data.frame(outcome = y2, pred1 = x1, pred2 = x2) %>% tbl_df
model.3 <- lm(outcome ~ pred1 + pred2, data = viol2)
plot(model.3, which = 2)
```



Skewed residuals often show up in strange patterns in the plot of residuals vs. fitted values, too, as in this case.

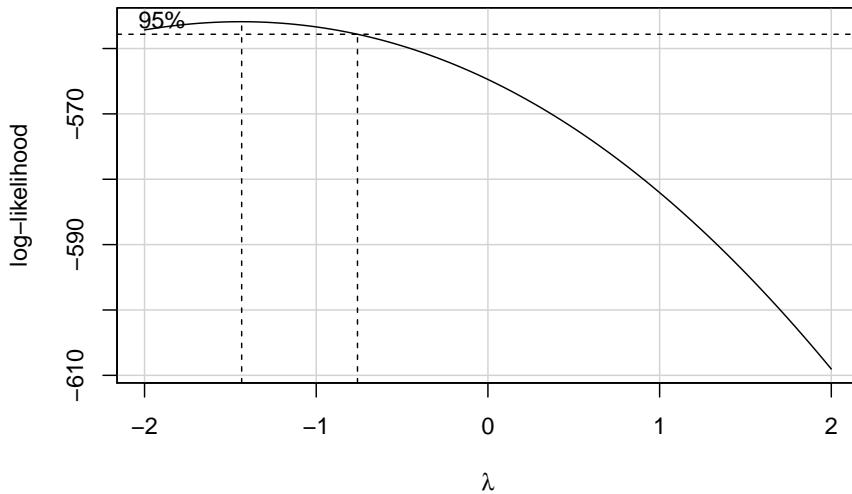
```
plot(model.3, which = 1)
```



Clearly, we have some larger residuals on the positive side, but not on the

negative side. Would an outcome transformation be suggested by Box-Cox?

```
boxCox(model.3); powerTransform(model.3)
```

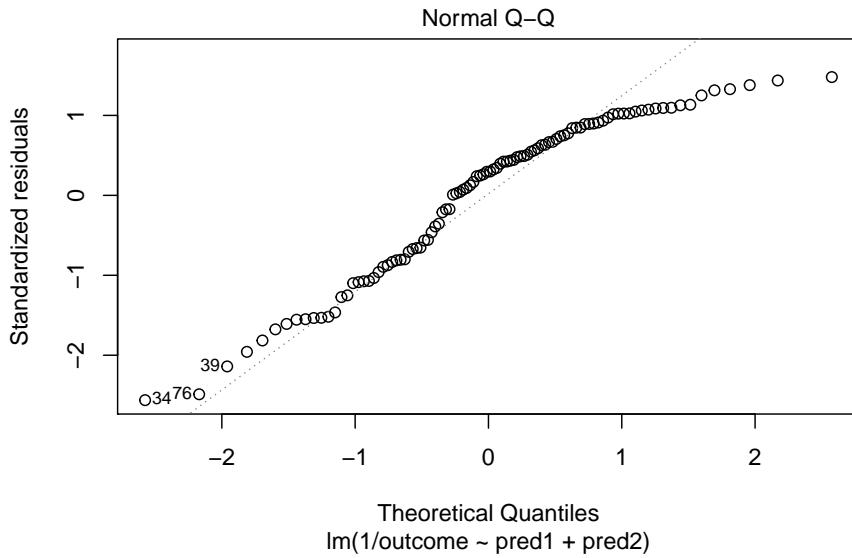


```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
argument ignored
```

```
Estimated transformation parameter
Y1
-1.44
```

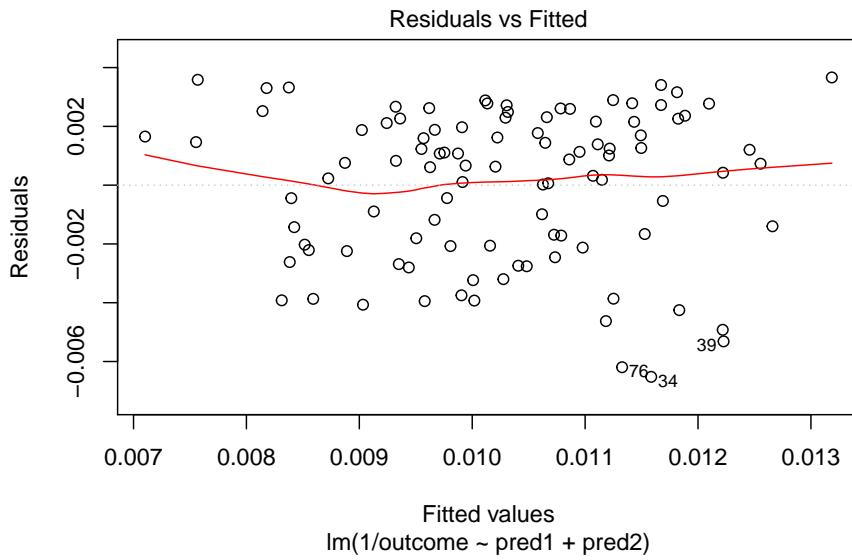
The suggested transformation is either the inverse or the inverse square of our outcome. Let's try the inverse.

```
model.4 <- lm(1/outcome ~ pred1 + pred2, data = viol2)
plot(model.4, which = 2)
```



OK. That's something of an improvement. How about the other residual plots with this transformation?

```
plot(model.4, which = 1)
```



The impact of the skew is reduced, at least. I might well be satisfied enough with this, in practice.

Chapter 37

Model Selection and Out-of-Sample Validation

Sometimes, we use a regression model for description of a multivariate relationship which requires only that we provide an adequate fit to the data at hand. Should we want to use the model for prediction, then a more appropriate standard which requires us to fit a model that does each of the following three things:

1. [fits well] Provides an adequate fit to the data at hand
2. [parsimonious] Includes only those predictors which actually have detectable predictive value
3. [predicts well out of sample] Does a better job predicting our outcome in new data than other alternative models

We'll spend considerable time in 432 studying how best to validate a regression model, but for now, we'll focus on a pair of issues:

- a. Given a set of predictors, how should we let the computer help us understand which subsets of those predictors are most worthy of our additional attention?
- b. Given a pair of models to compare, what tools should we use to determine which one better predicts new data?

37.1 Using the WCGS Data to predict Cholesterol Level

To address these issues, I'll look at one of our old examples: the `wcgs` data (Western Collaborative Group Study), described in Section 13. We'll try to

predict the variable `chol` on the basis of some subset of the following five predictors: `age`, `bmi`, `sbp`, `dbp` and `smoke`.

The steps are:

0. Check the `wcgs` data for missing or out-of-range values in the variables under study¹.
1. Separate the `wcgs` data into a test sample of 500 observations selected at random, and a model development (training) sample consisting of the remaining 2654 observations.
2. Using only the model development sample, run a stepwise regression algorithm working off of the kitchen sink model using backwards selection to identify a reasonable candidate for a model. Call this model A.
3. Develop a second potential model (called model B) for the model development data by eliminating the least clearly helpful predictor in model A.
4. Use the AIC to compare model A to model B in the development sample.
5. Finally, moving forward to the holdout sample, compare the quality of predictions made by model A to those made by model B in terms of two of the many possible criteria:
 - [i] mean squared prediction error and
 - [ii] mean absolute prediction error
 - to see if either model (A or B) is clearly superior in terms of out-of-sample prediction.

37.2 Separating the Data into a Training and a Test Sample

There are several ways to partition the data into training (model development) and test (model checking) samples. For example, we could develop separate data frames for a holdout sample of 500 randomly selected subjects (`wcgs.test`), and then use the remainder as the model development sample (`wcgs.dev`). Remember to set a seed so that you can replicate the selection.

```
set.seed(4311); wcgs.test <- wcgs %>% sample_n(500)
## hold out exactly 500 randomly selected observations

wcgs.dev <- anti_join(wcgs, wcgs.test, "id")
## model development sample - 2654 observations

wcgs.test
```

```
# A tibble: 500 x 22
```

¹ Actually, I will skip this range and missingness check here, and will wind up regretting that later.

```

      id   age agec height weight lnwght wghtcat   bmi    sbp lnsbp    dbp
      <int> <int> <fct> <int> <int> <dbl> <fct> <dbl> <int> <dbl> <int>
1 11424    40 35-40     72    202   5.31 > 200   27.4   138   4.93    84
2 12255    40 35-40     68    165   5.11 140-170   25.1   120   4.79    80
3 11100    44 41-45     72    196   5.28 170-200   26.6   118   4.77    80
4 10162    45 41-45     68    136   4.91 < 140   20.7   122   4.80    78
5 10236    51 51-55     65    159   5.07 140-170   26.5   106   4.66    80
6 19077    52 51-55     70    152   5.02 140-170   21.8   124   4.82    86
7 22064    49 46-50     71    152   5.02 140-170   21.2   124   4.82    76
8 21026    48 46-50     70    170   5.14 140-170   24.4   130   4.87    86
9 12356    42 41-45     74    190   5.25 170-200   24.4   144   4.97    78
10 11232   39 35-40    73    182   5.20 170-200   24.0   116   4.75    76
# ... with 490 more rows, and 11 more variables: chol <int>, behpat <fct>,
#   dibpat <fct>, smoke <fct>, ncigs <int>, arcus <int>, chd69 <fct>,
#   typchd69 <int>, time169 <int>, t1 <dbl>, uni <dbl>
dim(wcgs.dev) # verify size of development sample

```

```
[1] 2654 22
```

37.2.1 Using a specified fraction of the data in the test sample

If we'd wanted to select 20% of the data for our test sample, we could have instead used the `sample_frac` and `anti_join` commands. For the `wcgs` data which has a unique `id` variable that identifies each subject, we'd have...

```

set.seed(43199); wcgs.train80 <- wcgs %>% sample_frac(0.80)
wcgs.test20 <- anti_join(wcgs, wcgs.train80, by="id")
dim(wcgs.train80)

```

```
[1] 2523 22
```

```
dim(wcgs.test20)
```

```
[1] 631 22
```

Given a large sample size (at least 500 observations in the full data set) I would usually think about holding out somewhere between 15% and 25% of the data in this manner.

37.3 Stepwise Regression to Select Predictors

We next select the `wcgs.dev` (development sample) and run a stepwise procedure, beginning with the kitchen sink model, that includes all potential predictors.

37.3.1 The Kitchen Sink Model

```
summary(lm(chol ~ age + bmi + sbp + dbp + smoke, data = wcgs.dev))
```

Call:
`lm(formula = chol ~ age + bmi + sbp + dbp + smoke, data = wcgs.dev)`

Residuals:

Min	1Q	Median	3Q	Max
-130.8	-28.4	-3.0	25.1	418.5

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	135.3671	11.4050	11.87	< 2e-16 ***
age	0.5555	0.1520	3.65	0.00026 ***
bmi	0.7772	0.3493	2.23	0.02615 *
sbp	0.1307	0.0885	1.48	0.13977
dbp	0.2963	0.1388	2.14	0.03285 *
smokeYes	10.3565	1.6967	6.10	1.2e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42.8 on 2640 degrees of freedom
(8 observations deleted due to missingness)
Multiple R-squared: 0.0352, Adjusted R-squared: 0.0333
F-statistic: 19.2 on 5 and 2640 DF, p-value: <2e-16

37.3.2 Stepwise (Backward Elimination) Procedure

```
step(lm(chol ~ age + bmi + sbp + dbp + smoke, data = wcgs.dev))
```

Start: AIC=19892
`chol ~ age + bmi + sbp + dbp + smoke`

	Df	Sum of Sq	RSS	AIC
<none>		4847288	19892	
- sbp	1	4006	4851294	19892
- dbp	1	8370	4855658	19894
- bmi	1	9092	4856380	19895
- age	1	24516	4871804	19903
- smoke	1	68405	4915693	19927

Call:

```
lm(formula = chol ~ age + bmi + sbp + dbp + smoke, data = wcgs.dev)

Coefficients:
(Intercept)      age        bmi        sbp        dbp
  135.367      0.555      0.777      0.131      0.296
smokeYes
  10.357
```

The stepwise process first eliminates `sbp` from the model, then sees no substantial improvement in AIC after this has been done, so it lands on a four-predictor model with `age`, `bmi`, `dbp` and `smoke`.

37.3.3 Three Candidate Models

For purposes of this exercise, we'll call this four-predictor model `model.A` and compare it to a three-predictor model with `age`, `dbp` and `smoke`, which we'll call `model.B`

```
model.kitchensink <- lm(chol ~ age + bmi + sbp + dbp + smoke, data = wcgs.dev)
model.A <- lm(chol ~ age + bmi + dbp + smoke, data = wcgs.dev)
model.B <- lm(chol ~ age + dbp + smoke, data = wcgs.dev)
```

37.4 AIC, ANOVA and BIC to assess Candidate Models

The stepwise regression output specifies the AIC value for each model, but we can also look at other characteristics, like the ANOVA table comparing the various models, or the Bayesian Information Criterion, abbreviated BIC.

```
AIC(model.kitchensink, model.A, model.B)
```

	df	AIC
<code>model.kitchensink</code>	7	27403
<code>model.A</code>	6	27403
<code>model.B</code>	5	27406

AIC suggests model A (since it has the smallest AIC of these choices)

```
anova(model.kitchensink, model.A, model.B)
```

Analysis of Variance Table

```
Model 1: chol ~ age + bmi + sbp + dbp + smoke
Model 2: chol ~ age + bmi + dbp + smoke
Model 3: chol ~ age + dbp + smoke
```

```

Res.Df   RSS Df Sum of Sq    F Pr(>F)
1  2640 4847288
2  2641 4851294 -1     -4006 2.18   0.14
3  2642 4861218 -1     -9923 5.40   0.02 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The ANOVA model also suggests model A, for the following reasons:

- The p value of 0.35 indicates that moving from what we've called the kitchen sink model (model 1 in the ANOVA output) to what we've called model A (model 2 in the ANOVA output) does not have a statistically significant impact on predictive value.
- On the other hand, the p value of 0.013 indicates that moving from what we've called model A (model 2 in the ANOVA output) to what we've called model B (model 3 in the ANOVA output) does have a statistically significant impact on predictive value.
- Because these models are **nested** (model B is a proper subset of model A which is a proper subset of the kitchen sink) we can make these ANOVA comparisons directly.

```
BIC(model.kitchensink, model.A, model.B)
```

	df	BIC
model.kitchensink	7	27444
model.A	6	27438
model.B	5	27436

BIC disagrees, and prefers model B, since its BIC is smaller. The penalty for fitting additional predictors in BIC varies with the number of observations, and so (especially with larger samples) we can get meaningfully different AIC and BIC selections.

37.5 Comparing Models in the Test Sample (MSPE, MAPE)

Finally, we'll use our two candidate models (Model A and Model B) to predict the results in our holdout sample of 500 observations to see which model performs better in these new data (remember that our holdout sample was **not** used to identify or fit Models A or B.)

To do this, we first carefully specify the two models being compared

```

model.A <- lm(chol ~ age + bmi + dbp + smoke, data=wcgs.dev)
model.B <- lm(chol ~ age + dbp + smoke, data=wcgs.dev)

```

Next, use `predict` to make predictions for the test data:

```
modA.pre <- predict(model.A, newdata=wcgs.test)
modB.pre <- predict(model.B, newdata=wcgs.test)
```

Just to fix ideas, here are the first few predictions for Model A...

```
head(modA.pre)
```

	1	2	3	4	5	6
modA.pre	231	217	231	215	225	224

We can compare these to the first few observed values of `chol` in the test sample.

```
head(wcgs.test$chol)
```

chol	[1]	215	198	175	208	348	315
------	-----	-----	-----	-----	-----	-----	-----

Next, calculate errors (observed value minus predicted value) for each model:

```
modA.err <- wcgs.test$chol - modA.pre
modB.err <- wcgs.test$chol - modB.pre
```

Again, just to be sure we understand, we look at the first few errors for Model A.

```
head(modA.err)
```

	1	2	3	4	5	6
modA.err	-16.36	-19.04	-56.22	-7.47	123.48	90.97

Next, we calculate the absolute errors (as $|observed - predicted|$) from each model in turn:

```
modA.abserr <- abs(modA.err)
modB.abserr <- abs(modB.err)
```

Let's look at the first few absolute errors for Model A.

```
head(modA.abserr)
```

	1	2	3	4	5	6
modA.abserr	16.36	19.04	56.22	7.47	123.48	90.97

Next, we calculate the squared prediction errors from each model in turn:

```
modA.sqerr <- modA.err^2
modB.sqerr <- modB.err^2
```

And again, we look at the first few squared errors for Model A.

```
head(modA.sqerr)
```

	1	2	3	4	5	6
modA.sqerr	267.7	362.7	3160.8	55.8	15246.8	8275.4

564 CHAPTER 37. MODEL SELECTION AND OUT-OF-SAMPLE VALIDATION

To obtain our two key summaries: mean absolute prediction error and mean squared prediction error, I just use the `summary` function.

```
summary(modA.abserr)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.1	10.9	27.2	32.3	48.4	148.2	4

```
summary(modB.abserr)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.0	11.1	27.4	32.5	48.1	147.0	4

```
summary(modA.sqerr)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	119	740	1740	2340	21963	4

```
summary(modB.sqerr)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	123	749	1743	2318	21613	4

	Model	MAPE	MSPE	Maximum Abs. Error
A (age + bmi + dbp + smoke)		32.39	1669	125
B (age + dbp + smoke)		32.45	1668	123

Note that smaller values on these metrics are better, so that MAPE (barely) selects Model A over Model B, and MSPE (barely) selects Model B over Model A. We also sometimes look at the maximum absolute error, and here we see that Model B is slightly favored again. But really, there's little to choose from. The NAs you see above refer to patients in the `wcgs` data with missing values on one or more of the variables included in our kitchen sink model. I absolutely should have identified that problem at the beginning, and either omitted those or done some imputation back at the start. I'll show that in the next section.

So, based on the test sample results, we might slightly favor Model B, but really, there's no meaningful difference.

Chapter 38

Missing Data Mechanisms and Simple Imputation

Almost all serious statistical analyses have to deal with missing data. Data values that are missing are indicated in R, and to R, by the symbol `NA`.

38.1 A Toy Example

In the following tiny data set called `sbp_example`, we have four variables for a set of 15 subjects. In addition to a subject id, we have:

- the treatment this subject received (A, B or C are the treatments),
- an indicator (1 = yes, 0 = no) of whether the subject has diabetes,
- the subject's systolic blood pressure at baseline
- the subject's systolic blood pressure after the application of the treatment

```
# create some temporary variables

library(simputation)

subject <- 101:115
x1 <- c("A", "B", "C", "A", "C", "A", "A", NA, "B", "C", "A", "B", "C", "A", "B")
x2 <- c(1, 0, 0, 1, NA, 1, 0, 1, NA, 1, 0, 0, 1, 1, NA)
x3 <- c(120, 145, 150, NA, 155, NA, 135, NA, 115, 170, 150, 145, 140, 160, 135)
x4 <- c(105, 135, 150, 120, 135, 115, 160, 150, 130, 155, 140, 140, 150, 135, 120)

sbp_example <-
  data.frame(subject, treat = x1, diabetes = x2,
             sbp.before = x3, sbp.after = x4) %>%
```

```

tbl_df

rm(subject, x1, x2, x3, x4) # just cleaning up

sbp_example

# A tibble: 15 x 5
  subject treat diabetes sbp.before sbp.after
  <int> <fct>    <dbl>        <dbl>      <dbl>
1     101 A          1         120       105
2     102 B          0         145       135
3     103 C          0         150       150
4     104 A          1         NA        120
5     105 C         NA        155       135
6     106 A          1         NA        115
7     107 A          0         135       160
8     108 <NA>        1         NA        150
9     109 B         NA        115       130
10    110 C          1         170       155
11    111 A          0         150       140
12    112 B          0         145       140
13    113 C          1         140       150
14    114 A          1         160       135
15    115 B         NA        135       120

```

38.1.1 How many missing values do we have in each column?

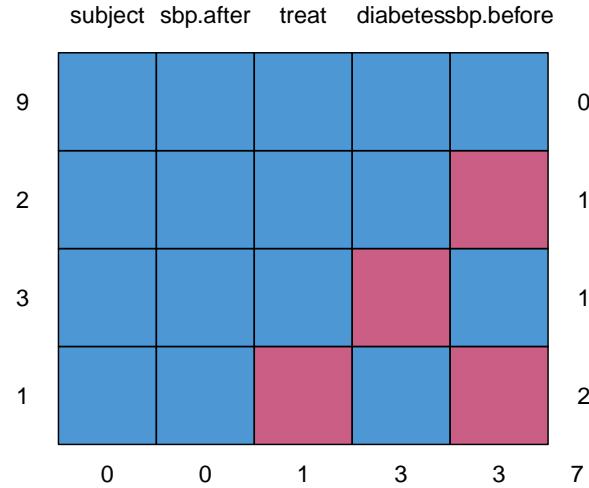
```
colSums(is.na(sbp_example))
```

subject	treat	diabetes	sbp.before	sbp.after
0	1	3	3	0

We are missing one treat, 3 diabetes and 3 sbp.before values.

38.1.2 What is the pattern of missing data?

```
mice::md.pattern(sbp_example)
```



	subject	sbp.after	treat	diabetes	sbp.before
9	1	1	1	1	1 0
2	1	1	1	1	0 1
3	1	1	1	0	1 1
1	1	1	0	1	0 2
	0	0	1	3	3 7

We have nine subjects with complete data, three subjects with missing `diabetes` (only), two subjects with missing `sbp.before` (only), and 1 subject with missing `treat` and `sbp.before`.

38.1.3 How can we identify the subjects with missing data?

```
sbp_example %>% filter(!complete.cases(.))

# A tibble: 6 x 5
  subject treat diabetes sbp.before sbp.after
    <int> <fct>    <dbl>      <dbl>      <dbl>
1     104 A          1        NA       120
2     105 C         NA        155      135
3     106 A          1        NA       115
4     108 <NA>        1        NA       150
5     109 B         NA       115      130
```

6	115 B	NA	135	120
---	-------	----	-----	-----

38.2 Missing-data mechanisms

My source for this description of mechanisms is Chapter 25 of Gelman and Hill (2007), and that chapter is available at this link.

1. **MCAR = Missingness completely at random.** A variable is missing completely at random if the probability of missingness is the same for all units, for example, if for each subject, we decide whether to collect the `diabetes` status by rolling a die and refusing to answer if a “6” shows up. If data are missing completely at random, then throwing out cases with missing data does not bias your inferences.
2. **Missingness that depends only on observed predictors.** A more general assumption, called **missing at random** or **MAR**, is that the probability a variable is missing depends only on available information. Here, we would have to be willing to assume that the probability of nonresponse to `diabetes` depends only on the other, fully recorded variables in the data. It is often reasonable to model this process as a logistic regression, where the outcome variable equals 1 for observed cases and 0 for missing. When an outcome variable is missing at random, it is acceptable to exclude the missing cases (that is, to treat them as NA), as long as the regression controls for all the variables that affect the probability of missingness.
3. **Missingness that depends on unobserved predictors.** Missingness is no longer “at random” if it depends on information that has not been recorded and this information also predicts the missing values. If a particular treatment causes discomfort, a patient is more likely to drop out of the study. This missingness is not at random (unless “discomfort” is measured and observed for all patients). If missingness is not at random, it must be explicitly modeled, or else you must accept some bias in your inferences.
4. **Missingness that depends on the missing value itself.** Finally, a particularly difficult situation arises when the probability of missingness depends on the (potentially missing) variable itself. For example, suppose that people with higher earnings are less likely to reveal them.

Essentially, situations 3 and 4 are referred to collectively as **non-random missingness**, and cause more trouble for us than 1 and 2.

38.3 Options for Dealing with Missingness

There are several available methods for dealing with missing data that are MCAR or MAR, but they basically boil down to:

- Complete Case (or Available Case) analyses

- Single Imputation
- Multiple Imputation

38.4 Complete Case (and Available Case) analyses

In **Complete Case** analyses, rows containing NA values are omitted from the data before analyses commence. This is the default approach for many statistical software packages, and may introduce unpredictable bias and fail to include some useful, often hard-won information.

- A complete case analysis can be appropriate when the number of missing observations is not large, and the missing pattern is either MCAR (missing completely at random) or MAR (missing at random.)
- Two problems arise with complete-case analysis:
 1. If the units with missing values differ systematically from the completely observed cases, this could bias the complete-case analysis.
 2. If many variables are included in a model, there may be very few complete cases, so that most of the data would be discarded for the sake of a straightforward analysis.
- A related approach is *available-case* analysis where different aspects of a problem are studied with different subsets of the data, perhaps identified on the basis of what is missing in them.

38.5 Single Imputation

In **single imputation** analyses, NA values are estimated/replaced *one time* with *one particular data value* for the purpose of obtaining more complete samples, at the expense of creating some potential bias in the eventual conclusions or obtaining slightly *less* accurate estimates than would be available if there were no missing values in the data.

- A single imputation can be just a replacement with the mean or median (for a quantity) or the mode (for a categorical variable.) However, such an approach, though easy to understand, underestimates variance and ignores the relationship of missing values to other variables.
- Single imputation can also be done using a variety of models to try to capture information about the NA values that are available in other variables within the data set.
- The **simputation** package can help us execute single imputations using a wide variety of techniques, within the pipe approach used by the **tidyverse**. Another approach I have used in the past is the **mice** package, which can also perform single imputations.

38.6 Multiple Imputation

Multiple imputation, where NA values are repeatedly estimated/replaced with multiple data values, for the purpose of obtaining more complete samples *and* capturing details of the variation inherent in the fact that the data have missingness, so as to obtain *more* accurate estimates than are possible with single imputation.

- We'll postpone the discussion of multiple imputation for a while.

38.7 Building a Complete Case Analysis

We can drop all of the missing values from a data set with `drop_na` or with `na.omit` or by filtering for `complete.cases`. Any of these approaches produces the same result - a new data set with 9 rows (after dropping the six subjects with any NA values) and 5 columns.

```
cc.1 <- na.omit(sbp_example)
cc.2 <- sbp_example %>% drop_na
cc.3 <- sbp_example %>% filter(complete.cases(..))
```

38.8 Single Imputation with the Mean or Mode

The most straightforward approach to single imputation is to impute a single summary of the variable, such as the mean, median or mode.

```
skimr::skim(sbp_example)
```

```
Skim summary statistics
n obs: 15
n variables: 5

-- Variable type:factor -----
variable missing complete n n_unique          top_counts ordered
  treat      1        14 15            3 A: 6, B: 4, C: 4, NA: 1 FALSE

-- Variable type:integer -----
variable missing complete n mean   sd  p0   p25  p50   p75 p100    hist
  subject     0        15 15  108 4.47 101 104.5 108 111.5 115 <U+2587><U+2587><U+2587>

-- Variable type:numeric -----
variable missing complete n  mean    sd  p0  p25  p50   p75 p100
  diabetes     3        12 15  0.58  0.51  0   0   1   1   1
  sbp.after    0        15 15 136   15.83 105 125 135 150    160
```

```

sbp.before      3      12 15 143.33 15.72 115 135 145 151.25 170
hist
<U+2586><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
<U+2582><U+2582><U+2585><U+2582><U+2587><U+2585><U+2587><U+2585>
<U+2585><U+2581><U+2585><U+2582><U+2585><U+2587><U+2582><U+2582>

```

Here, suppose we decide to impute

- `sbp.before` with the mean (143.33) among non-missing values,
- `diabetes` with its median (1) among non-missing values, and
- `treat` with its most common value, or mode (A)

```

si.1 <- sbp_example %>%
  replace_na(list(sbp.before = 143.33,
                  diabetes = 1,
                  treat = "A"))
si.1

```

	subject	treat	diabetes	sbp.before	sbp.after
	<int>	<fct>	<dbl>	<dbl>	<dbl>
1	101	A	1	120	105
2	102	B	0	145	135
3	103	C	0	150	150
4	104	A	1	143.	120
5	105	C	1	155	135
6	106	A	1	143.	115
7	107	A	0	135	160
8	108	A	1	143.	150
9	109	B	1	115	130
10	110	C	1	170	155
11	111	A	0	150	140
12	112	B	0	145	140
13	113	C	1	140	150
14	114	A	1	160	135
15	115	B	1	135	120

We could accomplish the same thing with, for example:

```

si.2 <- sbp_example %>%
  replace_na(list(sbp.before = mean(sbp_example$sbp.before, na.rm = TRUE),
                  diabetes = median(sbp_example$diabetes, na.rm = TRUE),
                  treat = "A"))

```

38.9 Doing Single Imputation with `simputation`

Single imputation is a potentially appropriate method when missingness can be assumed to be either completely at random (MCAR) or dependent only on observed predictors (MAR). We'll use the `simputation` package to accomplish it.

- The `simputation` vignette is available at <https://cran.r-project.org/web/packages/simputation/vignettes/intro.html>
- The `simputation` reference manual is available at <https://cran.r-project.org/web/packages/simputation/simputation.pdf>

38.9.1 Mirroring Our Prior Approach (imputing means/medians/modes)

Suppose we want to mirror what we did above, simply impute the mean for `sbp.before` and the median for `diabetes` again.

```
si.3 <- sbp_example %>%
  impute_lm(sbp.before ~ 1) %>%
  impute_median(diabetes ~ 1) %>%
  replace_na(list(treat = "A"))

si.3
```

```
# A tibble: 15 x 5
  subject treat diabetes sbp.before sbp.after
*   <int> <fct>    <dbl>      <dbl>     <dbl>
 1     101 A         1        120       105
 2     102 B         0        145       135
 3     103 C         0        150       150
 4     104 A         1        143.      120
 5     105 C         1        155       135
 6     106 A         1        143.      115
 7     107 A         0        135       160
 8     108 A         1        143.      150
 9     109 B         1        115       130
10    110 C         1        170       155
11    111 A         0        150       140
12    112 B         0        145       140
13    113 C         1        140       150
14    114 A         1        160       135
15    115 B         1        135       120
```

38.9.2 Using a model to impute `sbp.before` and `diabetes`

Suppose we wanted to use:

- a robust linear model to predict `sbp.before` missing values, on the basis of `sbp.after` and `diabetes` status, and
- a predictive mean matching approach to predict `diabetes` status, on the basis of `sbp.after`, and
- a decision tree approach to predict `treat` status, using all other variables in the data

```
set.seed(50001)

imp.4 <- sbp_example %>%
  impute_rlm(sbp.before ~ sbp.after + diabetes) %>%
  impute_pmm(diabetes ~ sbp.after) %>%
  impute_cart(treat ~ .)

imp.4

# A tibble: 15 x 5
  subject treat diabetes sbp.before sbp.after
  * <int> <fct>    <dbl>      <dbl>     <dbl>
1     101 A          1       120       105
2     102 B          0       145       135
3     103 C          0       150       150
4     104 A          1       139.      120
5     105 C          1       155       135
6     106 A          1       136.      115
7     107 A          0       135       160
8     108 A          1       155.      150
9     109 B          1       115       130
10    110 C          1       170       155
11    111 A          0       150       140
12    112 B          0       145       140
13    113 C          1       140       150
14    114 A          1       160       135
15    115 B          1       135       120
```

Details on the many available methods in `simputation` are provided in its manual. These include:

- `impute_cart` uses a Classification and Regression Tree approach for numerical or categorical data. There is also an `impute_rf` command which uses Random Forests for imputation.
- `impute_pmm` is one of several “hot deck” options for imputation, this one is predictive mean matching, which can be used with numeric data (only). Missing values are first imputed using a predictive model. Next, these

predictions are replaced with the observed values which are nearest to the prediction. Other imputation options in this group include random hot deck, sequential hot deck and k-nearest neighbor imputation.

- `impute_rlm` is one of several regression imputation methods, including linear models, robust linear models (which use what is called M-estimation to impute numerical variables) and lasso/elastic net/ridge regression models.

The `simputation` package can also do EM-based multivariate imputation, and multivariate random forest imputation, and several other approaches.

Chapter 39

Dealing with Missing Data

So what we should have done at the start of our `wcgs` analysis was to identify any issues with missing values or whether any variables show unreasonably large or small values.

```
wcgs.s <- wcgs %>%
  dplyr::select(id, chol, age, bmi, sbp, dbp, smoke)
```

39.1 Identifying Missingness

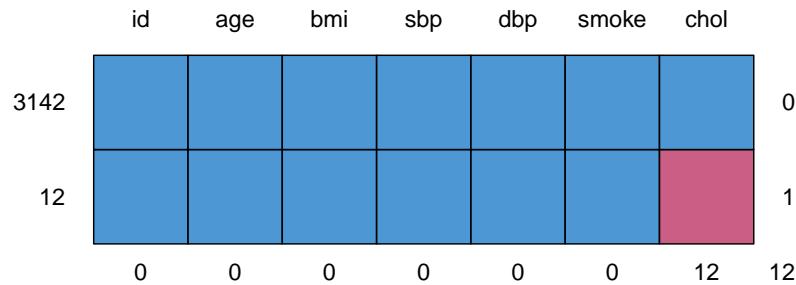
If you just want a count of missing values, you can use `colSums` and `is.na`

```
colSums(is.na(wcgs.s))
```

id	chol	age	bmi	sbp	dbp	smoke
0	12	0	0	0	0	0

The `mice` package provides the `md.pattern` function to identify missingness patterns.

```
library(mice)
md.pattern(wcgs.s)
```



```
      id age bmi sbp dbp smoke chol
3142  1   1   1   1   1     1   1   0
12    1   1   1   1   1     1   0   1
      0   0   0   0   0     0   12  12
```

Given the relatively small set of variables we're studying here, I would run the `describe` function from the `Hmisc` package on each variable (maybe skipping `id`) in order to identify missing and (potentially) unrealistic values through range checks.

```
Hmisc::describe(wcgs.s[-1]) # won't bother summarizing id
```

```
wcgs.s[-1]
```

```
6 Variables 3154 Observations
```

```
chol
```

n	missing	distinct	Info	Mean	Gmd	.05	.10
3142	12	237	1	226.4	47.99	161.1	175.0
.25	.50	.75	.90	.95			
197.2	223.0	253.0	280.0	302.0			

```
lowest : 103 110 111 112 113, highest: 386 390 400 414 645
```

```
age
```

n	missing	distinct	Info	Mean	Gmd	.05	.10
---	---------	----------	------	------	-----	-----	-----

3154	0	21	0.996	46.28	6.256	39	40
.25	.50	.75	.90	.95			
42	45	50	55	57			

lowest : 39 40 41 42 43, highest: 55 56 57 58 59

bmi

	n	missing	distinct	Info	Mean	Gmd	.05	.10
3154	0	679	1	24.52	2.803	20.59	21.52	
.25	.50	.75	.90	.95				
22.96	24.39	25.84	27.45	28.73				

lowest : 11.2 15.7 16.9 17.2 17.2, highest: 36.0 37.2 37.2 37.7 38.9

sbp

	n	missing	distinct	Info	Mean	Gmd	.05	.10
3154	0	62	0.996	128.6	16.25	110	112	
.25	.50	.75	.90	.95				
120	126	136	148	156				

lowest : 98 100 102 104 106, highest: 200 208 210 212 230

dbp

	n	missing	distinct	Info	Mean	Gmd	.05	.10
3154	0	42	0.992	82.02	10.51	68	70	
.25	.50	.75	.90	.95				
76	80	86	94	100				

lowest : 58 60 62 64 66, highest: 125 129 130 136 150

smoke

	n	missing	distinct
3154	0	2	

Value	No	Yes
Frequency	1652	1502
Proportion	0.524	0.476

No values are outside the range of plausibility, and in any case, we see that we have 12 missing chol values in the full data set. Options?

- We might choose to omit those rows before creating our test and training samples.
 - If we’re building a model where chol is the outcome, I would omit those cases, as (for 431, at least) I won’t impute outcomes. This is R’s default.

- We might choose to impute missing values after we partition.

39.2 Complete Case Analysis: A model for chol

Suppose we want to build a model for `chol` using age, body-mass index and systolic blood pressure, using the entire `wcgs` data frame, except those subjects with a missing `chol` value. Note that this is the default approach R takes, regardless of whether `chol` is used as an outcome or predictor.

```
model1 <- lm(chol ~ age + bmi + sbp, data = wcgs.s)
summary(model1)
```

```
Call:
lm(formula = chol ~ age + bmi + sbp, data = wcgs.s)

Residuals:
    Min      1Q  Median      3Q     Max 
-125.4   -28.6   -3.0    25.9   409.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 146.871    10.320   14.23 < 2e-16 ***
age          0.561     0.141    3.98  7.2e-05 ***
bmi          0.681     0.312    2.18   0.029 *  
sbp          0.287     0.054    5.31  1.2e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43 on 3138 degrees of freedom
(12 observations deleted due to missingness)
Multiple R-squared:  0.0214,    Adjusted R-squared:  0.0205 
F-statistic: 22.9 on 3 and 3138 DF,  p-value: 1.16e-14
```

The notification under the residual standard error indicates that 12 observations (the 12 with missing `chol`, since no other variables in this group have missing data) were dropped before the model was fit. You can also tell this from the degrees of freedom.

- We have four coefficients to fit (intercept and slopes of `age`, `bmi` and `sbp`) so we have $(4 - 1) = 3$ numerator degrees of freedom.
- The ANOVA F-statistic report also specifies 3138 denominator degrees of freedom.
- Total df is always one less than the total number of observations used in the model.
- That suggests we are using $(3 + 3138) + 1 = 3142$ subjects in this model.

- But `wcgs.s` actually contains 3,154 people so 12 were omitted.

```
dim(wcgs.s)
```

```
[1] 3154    7
```

39.2.1 Removing all subjects with NA, via `na.omit`

So, if we want to fit a complete case analysis, we don't, technically, have to do anything except specify the model. We can also make this explicit by first pruning all subjects with any missing values on any variables from our tibble.

```
wcgs.subnoNA <- dplyr::select(wcgs, id, chol, age, bmi, sbp) %>%
  na.omit ## the na.omit function drops all cases with NA
dim(wcgs.subnoNA)
```

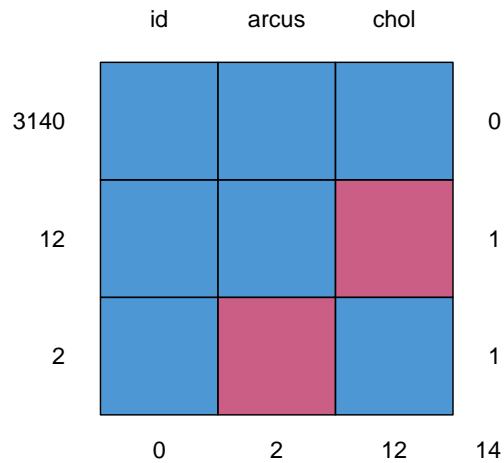
```
[1] 3142    5
```

39.2.2 Removing subjects with NA on one particular variable

If we wanted to specify that only subjects with missing `chol` should be removed from the data set (in case there were missing values in other variables), we could use the following approach.

Here, I'll work with a data set (`wcgs.s3`) that contains 12 missing `chol` values, and 2 missing `arcus` values, and build a new set (`wcgs.s3noNChol`) that retains the subjects who have a `chol` value, regardless of whether they are also missing `arcus`, but not include the 12 subjects with missing `chol` data

```
wcgs.s3 <- dplyr::select(wcgs, id, chol, arcus)
md.pattern(wcgs.s3)
```

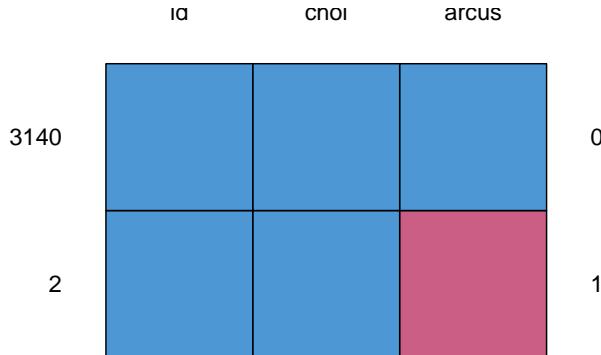


```

      id arcus chol
3140  1     1   1  0
12     1     1   0  1
2      1     0   1  1
      0     2   12 14

wcgs.s3noNACHOL <- wcgs.s3 %>%
  filter(!is.na(chol))
md.pattern(wcgs.s3noNACHOL)

```



```
  id chol arcus
3140  1   1     1 0
2      1   1     0 1
      0   0     2 2
```

39.3 Using Multiple Imputation to fit our Regression Model

Following the approach we outlined in the R-20 document¹, we'll try fitting the model while imputing the 12 cholesterol values, 100 times each, creating 100 new "complete" data sets.

```
wcgs.sub <- wcgs %>% dplyr::select(id, chol, age, bmi, sbp)
chol.mi <- mice(wcgs.sub, m = 100, maxit = 5,
                 meth = "pmm", printFlag = FALSE, seed = 4314)
```

39.3.1 Examining a Single Imputed Data Set

We could look at any one of our 100 imputations in some detail, perhaps the 12th such imputation, as follows:

¹Also see <https://stat.ethz.ch/education/seminsters/ss2012/ams/paper/mice.pdf>

```
imp12 <- mice::complete(chol.mi, 12)
Hmisc::describe(imp12$chol)

imp12$chol
  n   missing distinct      Info      Mean      Gmd      .05      .10
  3154        0     237        1    226.4    47.98    161.7    175.0
  .25       .50     .75        .90      .95
  197.2    223.0    253.0      280.0    302.0

lowest : 103 110 111 112 113, highest: 386 390 400 414 645
```

Working with a particular imputation might allow us to produce plots of the imputed data, or of diagnostics after a regression model, but we'll focus our interpretation of regression results (in 431) on estimates of coefficients and of things like R^2 . For that, we want to use all 100 imputations, in a pooled approach.

39.3.2 Fitting a Pooled Regression Model across the Imputations

Next, we'll fit a set of pooled regression model estimates across all 100 imputations, producing the following results. We'll even estimate both forms of R^2 .

```
model.1mi <- with(chol.mi, lm(chol ~ age + bmi + sbp))
pool(model.1mi)

Class: mipo    m = 100
      estimate      ubar       b       t dfcom   df      riv lambda
(Intercept) 146.993 1.06e+02 8.02e-01 1.07e+02  3150 3118 0.00766 0.00760
age          0.563 1.97e-02 1.47e-04 1.99e-02  3150 3119 0.00752 0.00747
bmi          0.685 9.69e-02 7.06e-04 9.76e-02  3150 3120 0.00735 0.00730
sbp          0.284 2.87e-03 4.84e-05 2.92e-03  3150 3068 0.01702 0.01674
      fmi
(Intercept) 0.00823
age          0.00810
bmi          0.00794
sbp          0.01738

round(summary(pool(model.1mi)),2)

      estimate std.error statistic   df p.value
(Intercept) 146.99      10.33     14.23 3118    0.00
age          0.56       0.14      3.99 3119    0.00
bmi          0.69       0.31      2.19 3120    0.03
sbp          0.28       0.05      5.25 3068    0.00
```

39.4. COMPARING TWO MODELS AFTER IMPUTATION WITH `POOL.COMPARE` 583

```
pool.r.squared(model.1mi, adjusted = FALSE)
```

```
est lo 95 hi 95 fmi  
R^2 0.0214 0.0125 0.0325 NaN
```

```
pool.r.squared(model.1mi, adjusted = TRUE)
```

```
est lo 95 hi 95 fmi  
adj R^2 0.0204 0.0118 0.0314 NaN
```

- `fmi` in the output above contains the fraction of missing information.
- `lambda` in that output is the proportion of the total variance that is attributable to the missing data.

So, how do these estimates after imputation compare to our complete case analysis originally developed as `model1`, and re-summarized below?

```
summary(model1)
```

Call:

```
lm(formula = chol ~ age + bmi + sbp, data = wcgs.s)
```

Residuals:

Min	1Q	Median	3Q	Max
-125.4	-28.6	-3.0	25.9	409.4

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	146.871	10.320	14.23	< 2e-16 ***
age	0.561	0.141	3.98	7.2e-05 ***
bmi	0.681	0.312	2.18	0.029 *
sbp	0.287	0.054	5.31	1.2e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43 on 3138 degrees of freedom

(12 observations deleted due to missingness)

Multiple R-squared: 0.0214, Adjusted R-squared: 0.0205

F-statistic: 22.9 on 3 and 3138 DF, p-value: 1.16e-14

39.4 Comparing Two Models After Imputation with `pool.compare`

Suppose we want to assess whether a model for `chol` with `age`, `bmi` and `sbp` is statistically significantly more effective than a model with `age` alone. As long as

the models we want to compare are *nested* (so that one model is a proper subset of the other), then this can be done as follows, to produce a Wald test.

```
model.1mi <- with(chol.mi, lm(chol ~ age + bmi + sbp))
model.2mi <- with(chol.mi, lm(chol ~ age))
comp1 <- pool.compare(model.1mi, model.2mi, method = "wald")

comp1$qbar1 # pooled estimate of first (larger) model

(Intercept)      age      bmi      sbp
146.993       0.563     0.685     0.284

comp1$qbar0 # pooled estimate of second (smaller) model

(Intercept)      age
194.0          0.7

comp1$Dm # Wald test statistic comparing the models

[,1]
[1,] 16.3

comp1$df1; comp1$df2 # degrees of freedom for Wald test

[1] 2

[1] 1311410

comp1$pvalue # p value for Wald test

[,1]
[1,] 8.71e-08
```

Practically, a significant value of the Wald test here suggests that the difference in predictive value between `model.1mi` and `model.2mi` is statistically significant, and thus, that you need the larger model. A non-significant Wald test would be consistent with a situation where you could use the model with `age` alone.

Here, we clearly want to retain `bmi` and `sbp` as compared to dropping them both from the model. This isn't a surprise, as we saw previously that *both* `bmi` and `sbp` had apparent predictive value (by the t test) even after all other variables were already in the model.

To create the overall F test, we could compare our model to an intercept only model, `model.intmi <- with(chol.mi, lm(chol ~ 1))` but we'll skip that here.

Chapter 40

BMI and Employment: Working with Categorical Predictors

40.1 The Data

A study recently published in *BMJ Open* looked at the differential relationship between employment status and body-mass index among middle-aged and elderly adults living in South Korea¹. Data from this study were available online thanks to the Dryad data package². The original data came from a nationally representative sample of 7228 participants in the Korean Longitudinal Study of Aging. I sampled these data, and did some data “rectangling” (wrangling) to build the `emp_bmi.csv` file on our web site.

The available data in `emp_bmi` describe 999 subjects, and included are 8 variables:

Variable	Description	NA?
<code>pid</code>	subject identification number (categorical)	0
<code>bmi</code>	our outcome, quantitative, body-mass index	0
<code>age</code>	subject's age (between 51 and 95)	1
<code>gender</code>	subject's gender (male or female)	0

¹See Noh J, Kim J, Park J, Oh I, Kwon YD (2016) Age and gender differential relationship between employment status and body mass index among middle-aged and elderly adults: a cross-sectional study. *BMJ Open* 6(11): e012117. <http://dx.doi.org/10.1136/bmjopen-2016-012117>

²Noh J, Kim J, Park J, Oh I, Kwon YD (2016) Data from: Age and gender differential relationship between employment status and body mass index among middle aged and elderly adults: a cross-sectional study. Dryad Digital Repository. <http://dx.doi.org/10.5061/dryad.ng8mn>

Variable	Description	NA?
employed	employment status indicator (1/0)	1
married	marital status indicator (1/0)	1
alcohol	3-level factor	2
education	4-level factor	5

```
Hmisc::describe(emp_bmi)
```

emp_bmi

8 Variables 999 Observations

pid

n	missing	distinct	Info	Mean	Gmd	.05	.10
999	0	999	1	31774	20178	3506	6961
.25	.50	.75	.90	.95			
16671	31761	46902	54635	58433			

lowest : 22 41 52 82 112, highest: 61471 61481 61532 61641 61691

bmi

n	missing	distinct	Info	Mean	Gmd	.05	.10
999	0	373	1	23.24	2.808	19.21	20.13
.25	.50	.75	.90	.95			
21.54	23.24	24.78	26.56	27.34			

lowest : 15.6 15.6 16.2 16.4 16.5, highest: 31.1 31.2 32.0 33.5 34.7

age

n	missing	distinct	Info	Mean	Gmd	.05	.10
998	1	43	0.999	66.29	11.56	52	53
.25	.50	.75	.90	.95			
57	66	74	81	83			

lowest : 51 52 53 54 55, highest: 89 90 92 93 95

gender

n	missing	distinct
999	0	2

Value	female	male
Frequency	570	429
Proportion	0.571	0.429

`employed`

n	missing	distinct	Info	Sum	Mean	Gmd
998	1	2	0.723	404	0.4048	0.4824

`married`

n	missing	distinct	Info	Sum	Mean	Gmd
998	1	2	0.548	758	0.7595	0.3657

`alcohol`

n	missing	distinct
997	2	3

Value	alcohol dependent	heavy drinker
Frequency	45	306
Proportion	0.045	0.307

Value	normal drinker or non-drinker
Frequency	646
Proportion	0.648

`education`

n	missing	distinct
994	5	4

Value	1 elem school grad or lower	2 middle school grad
Frequency	421	180
Proportion	0.424	0.181

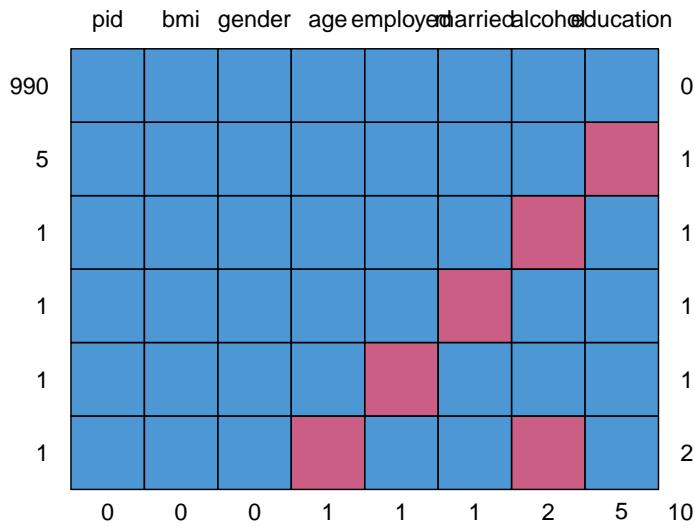
Value	3 high school grad	4 college grad or higher
Frequency	292	101
Proportion	0.294	0.102

40.1.1 Specifying Outcome and Predictors for our Model

In the original study, a key goal was to understand the relationship between employment and body-mass index. Our goal in this example will be to create a model to predict `bmi` focusing on employment status (so our key predictor is `employed`) while accounting for the additional predictors `age`, `gender`, `married`, `alcohol` and `education`. A natural thing to do would be to consider interactions of these predictor variables (for example, does the relationship between `bmi` and `employed` change when comparing men to women?) but we'll postpone that discussion until 432.

40.1.2 Dealing with Missing Predictor Values

```
md.pattern(emp_bmi)
```



	pid	bmi	gender	age	employed	married	alcohol	education
990	1	1	1	1	1	1	1	1 0
5	1	1	1	1	1	1	1	0 1
1	1	1	1	1	1	1	0	1 1
1	1	1	1	1	1	0	1	1 1
1	1	1	1	1	0	1	1	1 1
1	1	1	1	0	1	1	0	1 2
0	0	0	0	1	1	1	2	5 10

We will eventually build a model to predict `bmi` using all of the other variables besides `pid`. So we'll eventually have to account for the 9 people with missing values (one of whom has two missing values, as we see above.) What I'm going to do in this example is to first build a complete-case analysis on the 990 subjects without missing values and then, later, do multiple imputation to account for the 9 subjects with missing values (and their 10 actual missing values) sensibly.

I'll put the "complete cases" data set of 990 subjects in `emp_bmi_noNA`.

```
emp_bmi_noNA <- emp_bmi %>% na.omit  
emp_bmi_noNA
```

```
# A tibble: 990 x 8
```

```

      pid   bmi   age gender employed married alcohol      education
      <int> <dbl> <int> <fct>     <int>   <int> <fct>      <fct>
1    22  20.8    58 male        1       1 heavy drinker 4 college gra-
2    41  21.4    76 female      0       1 normal drinker~ 1 elem school~-
3    52  20.9    66 female      1       1 heavy drinker  1 elem school~-
4    82  23.7    67 male        1       1 alcohol depend~ 2 middle scho~-
5   112  25.5    63 female      0       1 heavy drinker  1 elem school~-
6   181  20.5    51 female      1       1 heavy drinker  3 high school~-
7   182  25.3    51 male        1       1 alcohol depend~ 3 high school~-
8   411  20.8    66 female      1       1 normal drinker~ 2 middle scho~-
9   491  20.8    64 female      0       1 normal drinker~ 3 high school~-
10  531  24.3    84 female      0       0 normal drinker~ 1 elem school~-
# ... with 980 more rows
colSums(is.na(emp_bmi_noNA))

      pid      bmi      age   gender employed married alcohol
      0        0        0        0        0        0        0        0
education
      0

```

40.2 The “Kitchen Sink” Model

A “kitchen sink” model includes all available predictors.

```

ebmodel.1 <- lm(bmi ~ age + gender + employed + married +
                  alcohol + education, data = emp_bmi_noNA)
summary(ebmodel.1)

```

Call:
`lm(formula = bmi ~ age + gender + employed + married + alcohol +
 education, data = emp_bmi_noNA)`

Residuals:

Min	1Q	Median	3Q	Max
-7.703	-1.606	-0.049	1.499	11.730

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	26.1539	0.8836	29.60	< 2e-16
age	-0.0426	0.0107	-4.00	6.9e-05
gendermale	0.2981	0.2027	1.47	0.142
employed	-0.4576	0.1915	-2.39	0.017
married	0.0944	0.2128	0.44	0.657

```

alcoholheavy drinker          0.2532   0.4073   0.62    0.534
alcoholnormal drinker or non-drinker 0.1412   0.4077   0.35    0.729
education2 middle school grad   -0.2886   0.2402   -1.20   0.230
education3 high school grad     -0.5012   0.2219   -2.26   0.024
education4 college grad or higher -0.7986   0.3107   -2.57   0.010

(Intercept)                   ***
age                         ***
gendermale
employed                     *
married
alcoholheavy drinker
alcoholnormal drinker or non-drinker
education2 middle school grad
education3 high school grad    *
education4 college grad or higher   *

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.51 on 980 degrees of freedom
Multiple R-squared:  0.0225,    Adjusted R-squared:  0.0135
F-statistic: 2.51 on 9 and 980 DF,  p-value: 0.00772

```

40.3 Using Categorical Variables (Factors) as Predictors

We have six predictors here, and five of them are categorical. Note that R recognizes each kind of variable in this case and models them appropriately. Let's look at the coefficients of our model.

40.3.1 gender: A binary variable represented by letters

The `gender` variable contains the two categories: male and female, and R recognizes this as a factor. When building a regression model with such a variable, R assigns the first of the two levels of the factor to the baseline, and includes in the model an indicator variable for the second level. By default, R assigns each factor a level order alphabetically.

So, in this case, we have:

```
is.factor(emp_bmi_noNA$gender)
```

```
[1] TRUE
```

```
levels(emp_bmi_noNA$gender)
```

```
[1] "female" "male"
```

As you see in the model, the `gender` information is captured by the indicator variable `gendermale`, which is 1 when `gender = male` and 0 otherwise.

So, when our model includes:

Coefficients:	Estimate	Std. Error	t value	Pr(> t)
<code>gendermale</code>	0.29811	0.20271	1.471	0.1417

this means that a male subject is predicted to have an outcome that is 0.29811 points higher than a female subject, if they have the same values of all of the other predictors.

Note that if we wanted to switch the levels so that “male” came first (and so that R would use “male” as the baseline category and “female” as the 1 value in an indicator), we could do so with the `forcats` package and the `fct_relevel` command. Building a model with this version of `gender` will simply reverse the sign of our indicator variable, but not change any of the other output.

```
emp_bmi_noNA$gender.2 <- fct_relevel(emp_bmi_noNA$gender, "male", "female")
revised.model <- lm(bmi ~ age + gender.2 + employed + married +
                     alcohol + education, data = emp_bmi_noNA)
summary(revised.model)
```

Call:

```
lm(formula = bmi ~ age + gender.2 + employed + married + alcohol +
   education, data = emp_bmi_noNA)
```

Residuals:

Min	1Q	Median	3Q	Max
-7.703	-1.606	-0.049	1.499	11.730

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	26.4520	0.9465	27.95	< 2e-16
<code>age</code>	-0.0426	0.0107	-4.00	6.9e-05
<code>gender.2female</code>	-0.2981	0.2027	-1.47	0.142
<code>employed</code>	-0.4576	0.1915	-2.39	0.017
<code>married</code>	0.0944	0.2128	0.44	0.657
<code>alcoholheavy drinker</code>	0.2532	0.4073	0.62	0.534
<code>alcoholnormal drinker or non-drinker</code>	0.1412	0.4077	0.35	0.729
<code>education2 middle school grad</code>	-0.2886	0.2402	-1.20	0.230
<code>education3 high school grad</code>	-0.5012	0.2219	-2.26	0.024
<code>education4 college grad or higher</code>	-0.7986	0.3107	-2.57	0.010

```
(Intercept) ***  
age ***  
gender.2female  
employed *  
married  
alcoholheavy drinker  
alcoholnormal drinker or non-drinker  
education2 middle school grad  
education3 high school grad *  
education4 college grad or higher *  
---  
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 2.51 on 980 degrees of freedom  
Multiple R-squared: 0.0225, Adjusted R-squared: 0.0135  
F-statistic: 2.51 on 9 and 980 DF, p-value: 0.00772
```

Note that the two categories here need to be both *mutually exclusive* (a subject cannot be in more than one category) and *collectively exhaustive* (all subjects must fit into this set of categories) in order to work properly as a regression predictor.

40.3.2 `employed`: A binary variable represented a 1/0 indicator

The `employed` and `married` variables are each described using an indicator variable, which is 1 if the condition of interest holds and 0 if it does not. R doesn't recognize this as a factor, but rather as a quantitative variable. However, this is no problem for modeling, where we just need to remember that if `employed` = 1, the subject is employed, and if `employed` = 0, the subject is not employed, to interpret the results. The same approach is used for `married`.

Coefficients:	Estimate	Std. Error	t value	Pr(> t)
<code>employed</code>	-0.45761	0.19153	-2.389	0.0171 *
<code>married</code>	0.09438	0.21280	0.444	0.6575

So, in our model, if subject A is employed, they are expected to have an outcome that is 0.46 points lower (-0.46 points higher) than subject B who is not employed but otherwise identical to subject A.

Similarly, if subject X is married, and subject Y is unmarried, but they otherwise have the same values of all predictors, then our model will predict a `bmi` for X that is 0.094 points higher than for Y.

40.3.3 alcohol: A three-category variable coded by names

Our `alcohol` information divides subjects into three categories, which are:

- normal drinker or non-drinker
- heavy drinker
- alcohol dependent

R builds a model using $k - 1$ predictors to describe a variable with k levels. As mentioned previously, R selects a baseline category when confronted with a factor variable, and it always selects the first level as the baseline. The levels are sorted alphabetically, unless we tell R to sort them some other way. So, we have

Coefficients:	Estimate	Std. Error	t value	Pr(> t)
alcoholheavy drinker	0.25317	0.40727	0.622	0.5343
alcoholnormal drinker or non-drinker	0.14121	0.40766	0.346	0.7291

How do we interpret this?

- Suppose subject A is alcohol dependent, B is a heavy drinker and C is a normal drinker or non-drinker, but subjects A-C have the same values of all other predictors.
- Our model predicts that B would have a BMI that is 0.25 points higher than A.
- Our model predicts that C would have a BMI that is 0.14 points higher than A.

A good way to think about this...

Subject	Status	alcoholheavy drinker	alcoholnormal drinker or non-drinker
A	alcohol dependent	0	0
B	heavy drinker	1	0
C	normal drinker or non-drinker	0	1

and so, with two variables, we cover each of these three possible `alcohol` levels.

When we have an ordered variable like this one, we usually want the baseline category to be at either end of the scale (either the highest or the lowest, but not something in the middle.) Another good idea in many settings is to use as the baseline category the most common category. Here, the baseline R chose was “alcohol dependent” which is the least common category, so I might want to use the `fct_relevel` function again to force R to choose, say, normal drinker/non-drinker as the baseline category.

```
emp_bmi_noNA$alcohol.2 <- fct_relevel(emp_bmi_noNA$alcohol,
                                         "normal drinker or non-drinker", "heavy drinker")
revised.model.2 <- lm(bmi ~ age + gender + employed + married +
                      alcohol.2 + education, data = emp_bmi_noNA)
```

```
summary(revised.model.2)
```

Call:

```
lm(formula = bmi ~ age + gender + employed + married + alcohol.2 +
    education, data = emp_bmi_noNA)
```

Residuals:

Min	1Q	Median	3Q	Max
-7.703	-1.606	-0.049	1.499	11.730

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	26.2951	0.8286	31.73	< 2e-16 ***
age	-0.0426	0.0107	-4.00	6.9e-05 ***
gendermale	0.2981	0.2027	1.47	0.142
employed	-0.4576	0.1915	-2.39	0.017 *
married	0.0944	0.2128	0.44	0.657
alcohol.2heavy drinker	0.1120	0.1965	0.57	0.569
alcohol.2alcohol dependent	-0.1412	0.4077	-0.35	0.729
education2 middle school grad	-0.2886	0.2402	-1.20	0.230
education3 high school grad	-0.5012	0.2219	-2.26	0.024 *
education4 college grad or higher	-0.7986	0.3107	-2.57	0.010 *

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 '	' 1		

Residual standard error: 2.51 on 980 degrees of freedom

Multiple R-squared: 0.0225, Adjusted R-squared: 0.0135

F-statistic: 2.51 on 9 and 980 DF, p-value: 0.00772

How do we interpret this revised model?

- Again, subject A is alcohol dependent, B is a heavy drinker and C is a normal drinker or non-drinker, but subjects A-C have the same values of all other predictors.
- Our model predicts that B would have a BMI that is 0.11 points higher than C.
- Our model predicts that A would have a BMI that is 0.14 points lower than C.

So, those are the same conclusions, just rephrased.

40.3.4 t tests and multi-categorical variables

The usual “last predictor in” t test works perfectly for binary factors, but suppose we have a factor like `alcohol` which is represented by two different indicator

40.3. USING CATEGORICAL VARIABLES (FACTORS) AS PREDICTORS 595

variables. If we want to know whether the alcohol information, as a group, adds statistically significant value to the model that includes all of the other predictors, then our best strategy is to compare two models - one with the alcohol information, and one without.

```
model.with.a <- lm(bmi ~ age + gender + alcohol + employed + married + education,
                     data = emp_bmi_noNA)
model.no.a <- lm(bmi ~ age + gender + employed + married + education,
                  data = emp_bmi_noNA)
anova(model.with.a, model.no.a)
```

Analysis of Variance Table

```
Model 1: bmi ~ age + gender + alcohol + employed + married + education
Model 2: bmi ~ age + gender + employed + married + education
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1     980 6190
2     982 6194 -2      -3.63 0.29   0.75
```

The p value for both of the indicator variables associated with alcohol combined is 0.75, according to an ANOVA F test with 2 degrees of freedom.

Note that we can get the same information from an ANOVA table of the larger model if we add the alcohol predictor to the model last.

```
anova(lm(bmi ~ age + gender + employed + married + education + alcohol,
          data = emp_bmi_noNA))
```

Analysis of Variance Table

```
Response: bmi
           Df Sum Sq Mean Sq F value Pr(>F)
age         1    56    55.5   8.79 0.0031 **
gender      1     0     0.4   0.06 0.8090
employed    1    31    30.7   4.87 0.0276 *
married     1     0     0.4   0.06 0.8077
education   3    52    17.3   2.74 0.0422 *
alcohol      2     4     1.8   0.29 0.7504
Residuals  980  6190    6.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again, we see p for the two alcohol indicators is 0.75.

40.3.5 `education`: A four-category variable coded by names

The `education` variable's codes are a little better designed. By preceding the text with a number for each code, we force R to attend to the level order we want to see.

Coefficients:		Estimate	Std. Error	t value	Pr(> t)
<code>education2</code>	middle school grad	-0.28862	0.24020	-1.202	0.2298
<code>education3</code>	high school grad	-0.50123	0.22192	-2.259	0.0241 *
<code>education4</code>	college grad or higher	-0.79862	0.31068	-2.571	0.0103 *

Since we have four education levels, we need those three indicator variables.

- `education2 middle school grad` is 1 if the subject is a middle school graduate, and 0 if they have some other status
- `education3 high school grad` is 1 if the subject is a high school graduate, and 0 if they have some other status
- `education4 college grad or higher` is 1 if the subject is a college graduate or has more education, and 0 if they have some other status.
- So the subjects with only elementary school or lower education are represented by zeros in all three indicators.

Suppose we have four subjects now, with the same values of all other predictors, but different levels of education.

Subject	Education	Estimated BMI
A	elementary school or less	A
B	middle school grad	A - 0.289
C	high school grad	A - 0.501
D	college grad	A - 0.799

Note that the four categories are *mutually exclusive* (a subject cannot be in more than one category) and *collectively exhaustive* (all subjects must fit into this set of categories.) As we have seen, this is a requirement of categorical variables in a regression analysis.

Let's run the ANOVA test for the `education` information captured in those three indicator variables...

```
anova(lm(bmi ~ age + gender + employed + married + alcohol + education,
          data = emp_bmi_noNA))
```

Analysis of Variance Table

```
Response: bmi
          Df Sum Sq Mean Sq F value Pr(>F)
```

```

age          1      56     55.5    8.79 0.0031 ** 
gender       1      0      0.4     0.06 0.8090  
employed     1      31     30.7    4.87 0.0276 *  
married      1      0      0.4     0.06 0.8077  
alcohol       2      4      1.8     0.28 0.7581  
education    3      52     17.4    2.75 0.0418 *  
Residuals   980   6190     6.3    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

So, as a group, the three indicator variables add statistically significant predictive value at the 5% significance level, since the F test for those three variables has $p = 0.042$

40.3.6 Interpreting the Kitchen Sink Model

So, again, here's our model, now pandered into a prettier format using the `pander` function from the `pander` package.

```

ebmodel.1 <- lm(bmi ~ age + gender + employed + married +
                  alcohol + education, data = emp_bmi_noNA)
pander::pander(ebmodel.1)

```

Table 40.4: Fitting linear model: $\text{bmi} \sim \text{age} + \text{gender} + \text{employed}$
+ $\text{married} + \text{alcohol} + \text{education}$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	26.15	0.8836	29.6	4.401e-138
age	-0.0426	0.01065	-3.998	6.858e-05
gendermale	0.2981	0.2027	1.471	0.1417
employed	-0.4576	0.1915	-2.389	0.01707
married	0.09438	0.2128	0.4435	0.6575
alcoholheavy drinker	0.2532	0.4073	0.6216	0.5343
alcoholnormal drinker	0.1412	0.4077	0.3464	0.7291
or non-drinker				
education2 middle school grad	-0.2886	0.2402	-1.202	0.2298
education3 high school grad	-0.5012	0.2219	-2.259	0.02413
education4 college grad or higher	-0.7986	0.3107	-2.571	0.0103

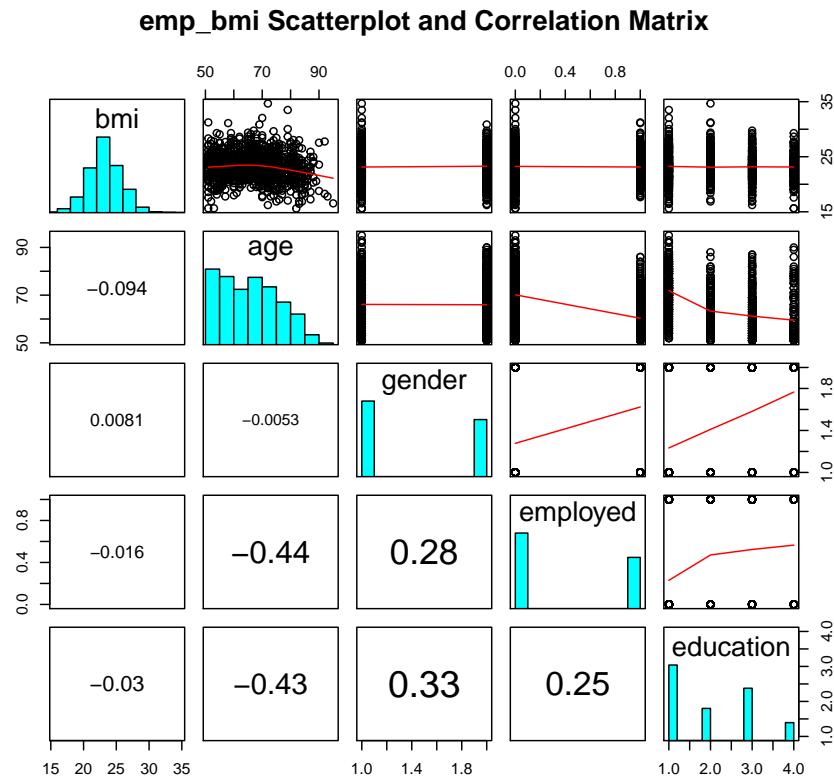
If we wanted to predict a BMI level for a new subject like the ones used in the development of this model, that prediction would be:

- 26.15
- minus 0.426 times the subject's `age`
- plus 0.298 if the subject's `gender` was `male`
- minus 0.458 if the subject's employment status was `employed`
- plus 0.253 if the subject's alcohol classification was `heavy drinker`
- plus 0.141 if the subject's alcohol classification was `normal drinker or non-drinker`
- minus 0.289 if the subject's `education` classification was `2 middle school grad`
- minus 0.501 if the subject's `education` classification was `3 high school grad`
- minus 0.799 if the subject's `education` classification was `4 college grad or higher`

40.4 Scatterplot Matrix with Categorical Predictors

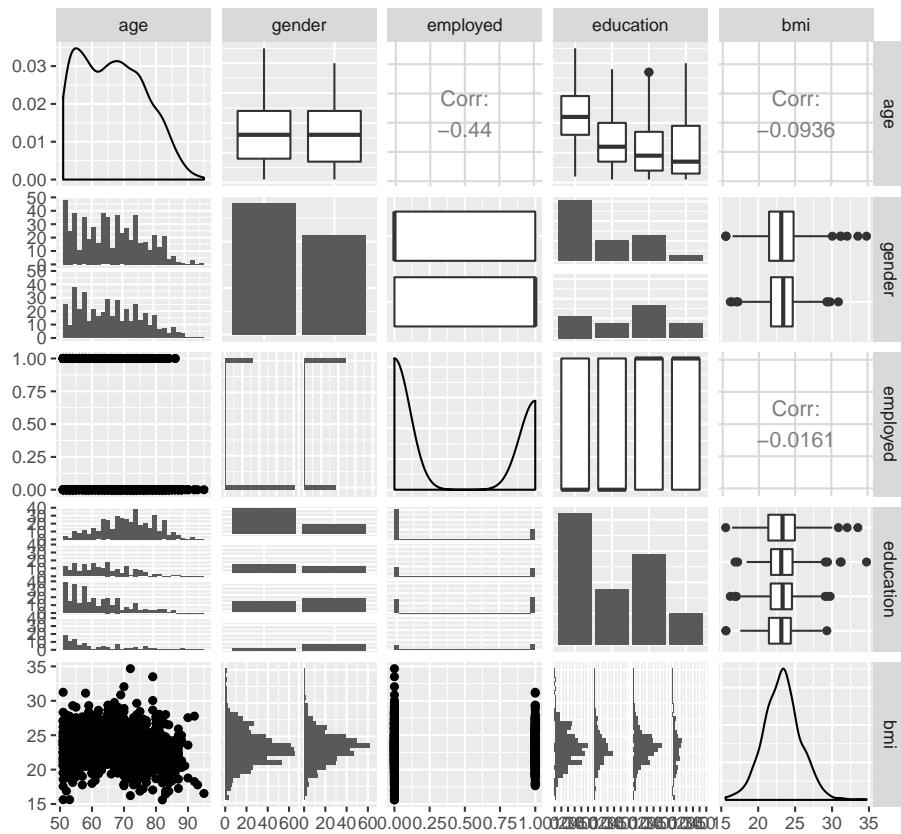
Let's look at a scatterplot matrix of a few key predictors, with my favorite approach (at least for quantitative predictors)...

```
pairs(~ bmi + age + gender + employed + education, data = emp_bmi_noNA,
      main = "emp_bmi Scatterplot and Correlation Matrix",
      upper.panel = panel.smooth,
      diag.panel = panel.hist,
      lower.panel = panel.cor)
```



Notice how the `ggpairs` approach reacts to the inclusion of factors as predictors...

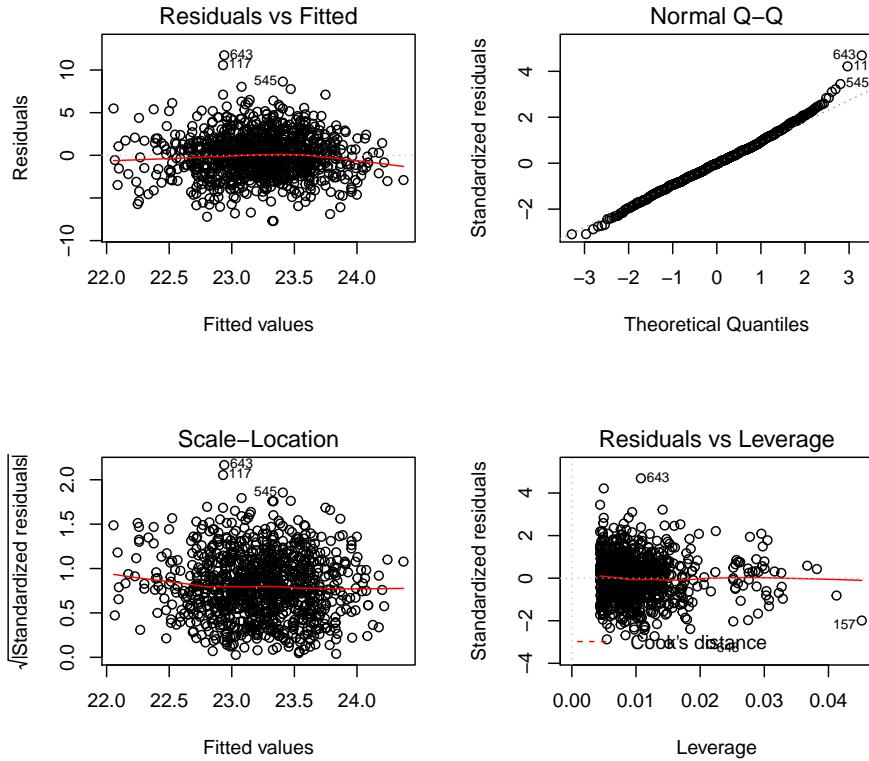
```
GGally::ggpairs(dplyr::select(emp_bmi_noNA, age, gender, employed, education, bmi))
```



40.5 Residual Plots when we have Categorical Predictors

Here are the main residual plots from the kitchen sink model `ebmodel.1` defined previously.

```
par(mfrow=c(2,2))
plot(ebmodel.1)
```



```
par(mfrow=c(1,1))
```

Sometimes, in small samples, the categorical variables will make the regression residuals line up in somewhat strange patterns. But in this case, there's no real problem. The use of categorical variables also has some impact on leverage, as it's hard for a subject to be a serious outlier in terms of a predictor if that predictor only has a few possible levels.

40.6 Stepwise Regression and Categorical Predictors

When R does backwards elimination for stepwise model selection, it makes decisions about each categorical variable as in/out across all of the indicator variables simultaneously, as you'd hope.

```
step(ebmodel.1)
```

602CHAPTER 40. BMI AND EMPLOYMENT: WORKING WITH CATEGORICAL PREDICTORS

Start: AIC=1835

bmi ~ age + gender + employed + married + alcohol + education

	Df	Sum of Sq	RSS	AIC
- alcohol	2	3.6	6194	1831
- married	1	1.2	6191	1833
<none>			6190	1835
- gender	1	13.7	6204	1835
- education	3	52.1	6242	1837
- employed	1	36.1	6226	1838
- age	1	101.0	6291	1849

Step: AIC=1831

bmi ~ age + gender + employed + married + education

	Df	Sum of Sq	RSS	AIC
- married	1	1.0	6195	1829
<none>			6194	1831
- gender	1	19.1	6213	1832
- education	3	51.9	6245	1833
- employed	1	34.7	6228	1835
- age	1	108.6	6302	1846

Step: AIC=1829

bmi ~ age + gender + employed + education

	Df	Sum of Sq	RSS	AIC
<none>			6195	1829
- gender	1	22.2	6217	1831
- education	3	51.3	6246	1832
- employed	1	34.4	6229	1833
- age	1	125.4	6320	1847

Call:

lm(formula = bmi ~ age + gender + employed + education, data = emp_bmi_noNA)

Coefficients:

(Intercept)		age
26.5021		-0.0446
gendermale		employed
0.3406		-0.4457
education2 middle school grad		education3 high school grad
-0.2879		-0.4977
education4 college grad or higher		
-0.7904		

Note that the stepwise approach first drops two degrees of freedom (two indicator variables) for `alcohol` and then drops the one degree of freedom for `married` before it settles on a model with `age`, `gender`, `education` and `employed`.

40.7 Pooling Results after Multiple Imputation

As mentioned earlier, having built a model using complete cases, we should probably investigate the impact of multiple imputation on the missing observations. We'll fit 100 imputations using the `emp_bmi` data and then fit a pooled regression model across those imputations.

```
emp_bmi_mi <- mice(emp_bmi, m = 100, maxit = 5,
                     printFlag = FALSE, seed = 4476)
```

Now, we'll fit the pooled kitchen sink regression model to these imputed data sets and pool them.

```
model.empbmi.mi <- with(emp_bmi_mi, lm(bmi ~ age + gender + employed + married +
                                         alcohol + education))
round(summary(pool(model.empbmi.mi)),3)
```

	estimate	std.error	statistic	df
(Intercept)	26.130	0.877	29.780	985
age	-0.042	0.011	-3.983	985
gendermale	0.299	0.202	1.481	986
employed	-0.464	0.190	-2.441	987
married	0.096	0.212	0.455	987
alcoholheavy drinker	0.259	0.402	0.645	986
alcoholnormal drinker or non-drinker	0.142	0.403	0.353	986
education2 middle school grad	-0.274	0.239	-1.146	984
education3 high school grad	-0.500	0.221	-2.264	982
education4 college grad or higher	-0.797	0.309	-2.581	982
	p.value			
(Intercept)	0.000			
age	0.000			
gendermale	0.139			
employed	0.015			
married	0.649			
alcoholheavy drinker	0.519			
alcoholnormal drinker or non-drinker	0.724			
education2 middle school grad	0.252			
education3 high school grad	0.024			
education4 college grad or higher	0.010			

Note that in summarizing these pooled results, R does some strange things:

- it relabels the indicator variables as `gender2` rather than `gendermale` so you need to know the level order for each variable.
- it leaves as NA the number of missing values imputed for every categorical variable it recognizes as a factor.

Chapter 41

Species Found on the Galapagos Islands

41.1 A Little Background

The `gala` data describe features of the 30 Galapagos Islands.

The Galapagos Islands are found about 900 km west of South America: specifically the continental part of Ecuador. The Islands form a province of Ecuador and serve as a national park and marine reserve. They are noted for their vast numbers of unique (or endemic) species and were studied by Charles Darwin during the voyage of the Beagle. I didn't know most of this, but it's on Wikipedia, so I'll assume it's all true until someone sets me straight.

41.1.1 Sources

The data were initially presented by Johnson M and Raven P (1973) Species number and endemism: the Galapagos Archipelago revisited. *Science* 179: 893-895 and also appear in several regression texts, including my source: Faraway (2015). Note that Faraway filled in some missing data to simplify things a bit. A similar version of the data is available as part of the `faraway` library in R, but I encourage you to use the version I supply on our web site.

41.1.2 Variables in the `gala` data frame

- `id` = island identification code
- `island` = island name
- `species` = our outcome, the number of species found on the island

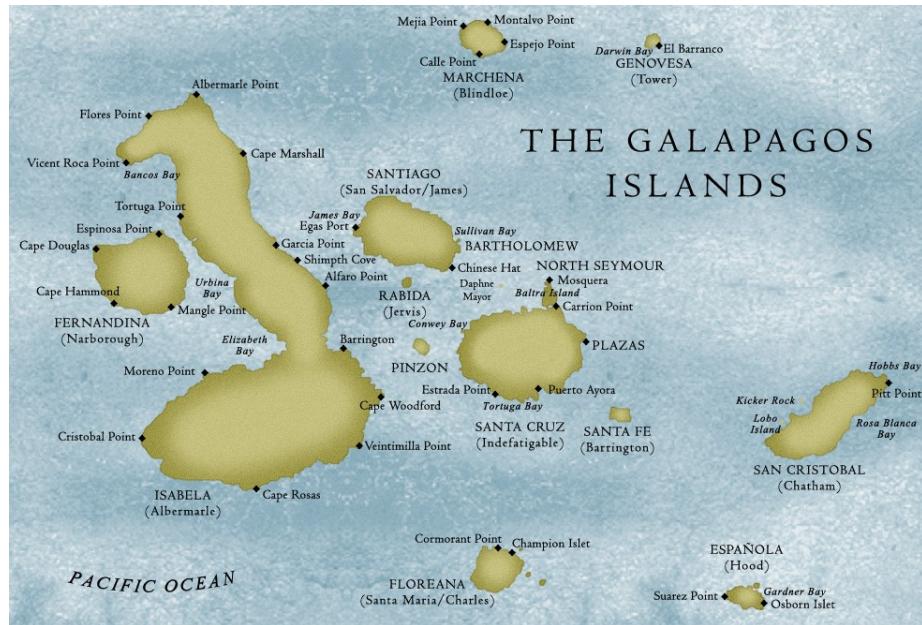


Figure 41.1: galapic.jpg

- **area** = the area of the island, in square kilometers
- **elevation** = the highest elevation of the island, in meters
- **nearest** = the distance from the nearest island, in kilometers
- **scruz** = the distance from Santa Cruz Island, in kilometers. Santa Cruz is the home to the largest human population in the Islands, and to the town of Puerto Ayora.
- **adjacent** = the area of the adjacent island, in square kilometers

gala

#	A tibble: 30 x 8	id	island	species	area	elevation	nearest	scruz	adjacent
		<int>	<fct>	<int>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
1	1	Baltra		58	25.1	346	0.6	0.6	1.84
2	2	Bartolome		31	1.24	109	0.6	26.3	572.
3	3	Caldwell		3	0.21	114	2.8	58.7	0.78
4	4	Champion		25	0.1	46	1.9	47.4	0.18
5	5	Coamano		2	0.05	77	1.9	1.9	904.
6	6	Daphne.Major		18	0.34	119	8	8	1.84
7	7	Daphne.Minor		24	0.08	93	6	12	0.34
8	8	Darwin		10	2.33	168	34.1	290.	2.85
9	9	Eden		8	0.03	71	0.4	0.4	18.0
10	10	Enderby		2	0.18	112	2.6	50.2	0.1

```
# ... with 20 more rows
Hmisc::describe(gala) # check for missing and inexplicable values
```

gala

8 Variables 30 Observations

id

n	missing	distinct	Info	Mean	Gmd	.05	.10
30	0	30	1	15.5	10.33	2.45	3.90
.25	.50	.75	.90	.95			
8.25	15.50	22.75	27.10	28.55			

lowest : 1 2 3 4 5, highest: 26 27 28 29 30

island

n	missing	distinct
30	0	30

lowest : Baltra Bartolome Caldwell Champion Coamano
highest: SantaFe SantaMaria Seymour Tortuga Wolf

species

n	missing	distinct	Info	Mean	Gmd	.05	.10
30	0	27	0.999	85.23	109.5	2.0	2.9
.25	.50	.75	.90	.95			
13.0	42.0	96.0	280.5	319.1			

lowest : 2 3 5 8 10, highest: 237 280 285 347 444

area

n	missing	distinct	Info	Mean	Gmd	.05	.10
30	0	29	1	261.7	478.6	0.0390	0.0770
.25	.50	.75	.90	.95			
0.2575	2.5900	59.2375 578.5460	782.6215				

Value	0	20	30	60	130	170	550	570	630	900
Frequency	17	3	1	2	1	1	1	1	1	1
Proportion	0.567	0.100	0.033	0.067	0.033	0.033	0.033	0.033	0.033	0.033

Value	4670
Frequency	1
Proportion	0.033

elevation

	n	missing	distinct	Info	Mean	Gmd	.05	.10
	30	0	30	1	368	411.1	47.35	68.80
	.25	.50	.75	.90	.95			
	97.75	192.00	435.25	868.20	1229.40			
lowest :	25	46	49	71	76,	highest:	777	864
								906 1494 1707
<hr/>								
nearest								
	n	missing	distinct	Info	Mean	Gmd	.05	.10
	30	0	22	0.997	10.06	13.73	0.445	0.590
	.25	.50	.75	.90	.95			
	0.800	3.050	10.025	34.100	40.205			
lowest :	0.2	0.4	0.5	0.6	0.7,	highest:	16.5	29.1
								34.1 45.2 47.4
<hr/>								
scruz								
	n	missing	distinct	Info	Mean	Gmd	.05	.10
	30	0	29	1	56.98	65.17	0.49	0.60
	.25	.50	.75	.90	.95			
	11.02	46.65	81.08	97.73	193.90			
lowest :	0.0	0.4	0.6	1.9	8.0,	highest:	93.1	95.3
								119.6 254.7 290.2
<hr/>								
adjacent								
	n	missing	distinct	Info	Mean	Gmd	.05	.10
	30	0	21	0.998	261.1	477.8	0.10	0.10
	.25	.50	.75	.90	.95			
	0.52	2.59	59.24	578.55	782.62			
Value	0	20	30	60	130	570	630	900
Frequency	17	2	2	2	2	2	1	1
Proportion	0.567	0.067	0.067	0.067	0.067	0.067	0.033	0.033
								4670
<hr/>								

41.2 DTDP: A Scatterplot Matrix

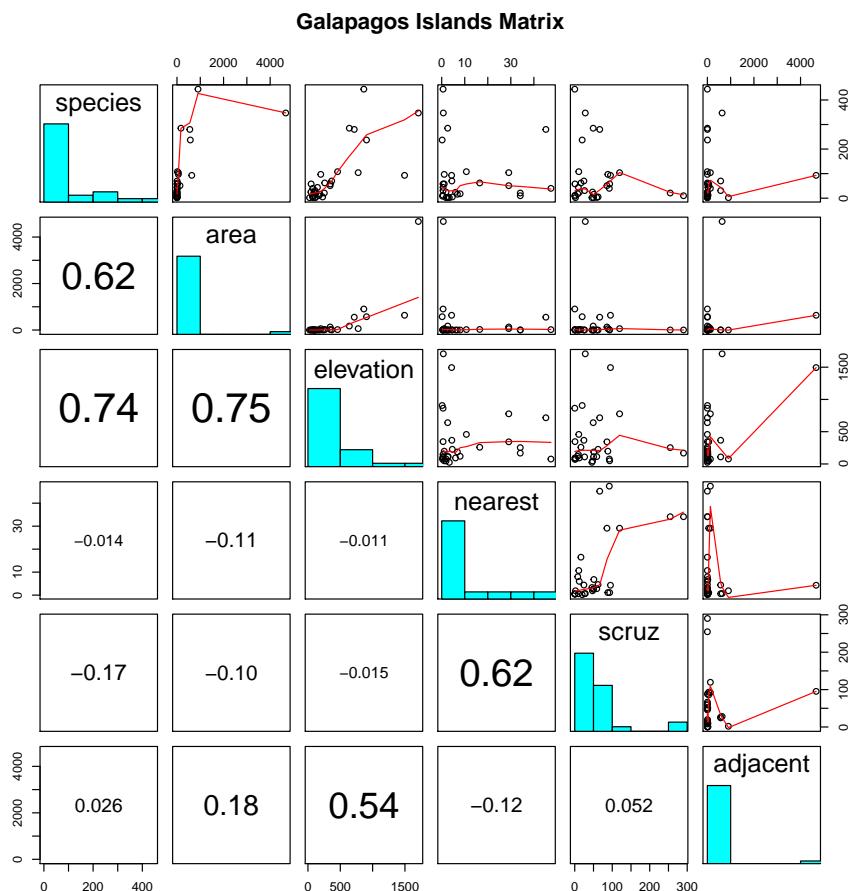
After missingness and range checks, the first step in any data analysis problem is to draw the picture. The most useful picture for me in thinking about a regression problem with a reasonably small number of predictors is a scatterplot matrix.

Our outcome, that we are predicting here is the number of *species*.

We'll use five predictors:

- area
- elevation
- nearest
- scruz and
- adjacent.

```
pairs (~ species + area + elevation + nearest + scruz + adjacent,
  data=gala, main="Galapagos Islands Matrix",
  upper.panel = panel.smooth,
  diag.panel = panel.hist,
  lower.panel = panel.cor)
```



41.2.1 Questions about the Scatterplot Matrix

1. What are we looking for in the scatterplots in the top row?

2. What can we learn from the Pearson correlations in the left column?
3. How do the histograms help increase our understanding of the data?
4. What about the scatterplots that are not in the top row?
5. What can we learn from the Pearson correlations that compare predictors?

41.3 Fitting A “Kitchen Sink” Linear Regression model

Next, we’ll fit a multiple linear regression model to predict the number of species based on the five predictors included in the `gala` data frame (and scatterplot matrix above.) We use the `lm` command to fit the linear model, and use what is called Wilkinson-Rogers notation to specify the model.

```
model1 <- lm(species ~ area + elevation + nearest + scruz +
               adjacent, data=gala)
summary(model1)
```

```
Call:
lm(formula = species ~ area + elevation + nearest + scruz + adjacent,
   data = gala)

Residuals:
    Min      1Q  Median      3Q     Max 
-111.68 -34.90  -7.86  33.46 182.58 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.06822   19.15420    0.37   0.7154    
area        -0.02394   0.02242   -1.07   0.2963    
elevation    0.31946   0.05366    5.95 3.8e-06 ***  
nearest      0.00914   1.05414    0.01   0.9932    
scruz        -0.24052   0.21540   -1.12   0.2752    
adjacent     -0.07480   0.01770   -4.23  0.0003 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61 on 24 degrees of freedom
Multiple R-squared:  0.766, Adjusted R-squared:  0.717 
F-statistic: 15.7 on 5 and 24 DF,  p-value: 6.84e-07
```

41.3.1 Questions about the Kitchen Sink Model Summaries

What conclusions can we draw from the `summary` output for this model? Specifically ...

6. What is being predicted? What is the prediction equation?
7. How do we interpret the `elevation` estimate of 0.32?
8. How do we interpret the `area` estimate of -0.02?
9. How do we interpret the intercept estimate of 7.07?
10. Overall, does the model add statistically significant predictive value over the simplest possible model, using the intercept term alone?
11. What proportion of the variation in `species` counts does this model account for?
12. What does the residual standard error mean in this context?
13. What can we learn from the standard errors in the coefficient output?
14. What can we learn from the `t` values and `Pr(>|t|)` values in the coefficient output?
15. How should we interpret the meaning of the `Adjusted R-squared` value?

41.4 Finding Confidence Intervals for our Coefficient Estimates

```
confint(model1, level = 0.95)
```

	2.5 %	97.5 %
(Intercept)	-32.4641	46.6005
area	-0.0702	0.0223
elevation	0.2087	0.4302
nearest	-2.1665	2.1848
scruz	-0.6851	0.2040
adjacent	-0.1113	-0.0383

41.4.1 Questions about the Confidence Intervals

16. What can we learn from the provided confidence interval for `elevation`?
17. How do the confidence interval results here compare to the t tests in the `summary` output?

41.5 Measuring Collinearity - the Variance Inflation Factor

The **variance inflation factor** (abbreviated VIF) can be used to quantify the impact of multicollinearity in a linear regression model.

The VIF is sometimes interpreted by taking its square root, and then interpreting the result as telling you how much larger the standard error for that coefficient is, as compared to what it would be if that variable were uncorrelated with the other predictors.

In R, the `vif` function from the `car` library, when applied to a linear regression model, specifies the variance inflation factors for each of the model's coefficients, as follows.

```
vif(model1)
```

area	elevation	nearest	scruz	adjacent
2.93	3.99	1.77	1.68	1.83

So, for instance, the VIF of 3.99 for `elevation` implies that the standard error of the elevation coefficient is approximately 2 times larger than it would be if elevation was uncorrelated with the other predictors.

I will look closely at any VIF value that is greater than 5, although some people use a cutoff of 10.

- Another collinearity measure called tolerance is simply $1/VIF$.
- For example, the tolerance for `elevation` would be 0.25, and the cutoff for a potentially problematic tolerance is either 0.2 or lower, or 0.1 or lower.

To calculate the VIF for a predictor x_1 , use all of the other predictors to predict x_1 and find the multiple R^2 value.

- VIF for $x_1 = 1 / (1 - R_{x_1|others}^2)$, and `tolerance = (1 - R_{x_1|others}^2)`.

41.6 Global (F) Testing of Overall Significance

Our Galapagos Islands species count regression model (called `model1`) predicts the count of an island's species using `area`, `elevation`, `nearest`, `scruz` and `adjacent`.

```
nullmodel <- lm(species ~ 1, data=gala)
summary(nullmodel)
```

```
Call:
lm(formula = species ~ 1, data = gala)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-83.2	-72.2	-43.2	10.8	358.8

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.2	20.9	4.07	0.00033 ***

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 115 on 29 degrees of freedom
```

```
anova(model1, nullmodel)
```

Analysis of Variance Table

```
Model 1: species ~ area + elevation + nearest + scruz + adjacent
```

```
Model 2: species ~ 1
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	24	89231			
2	29	381081	-5	-291850	15.7 6.8e-07 ***

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

41.6.1 Questions about the Global Test via ANOVA

18. How do we interpret the null model fit above?
19. What are the hypotheses being tested by this ANOVA output?
20. What conclusions can we draw from the ANOVA output presented here?
21. Where do we find information regarding the result for the previous question in the summary output for the linear model?
22. How would we set up an ANOVA model to test whether the “kitchen sink” model’s predictive value would be significantly impacted by removing the `adjacent` predictor from the model?
23. Where do we find information regarding these result for the previous question in the `summary` output for the linear model?
24. How would we set an ANOVA model to test whether a model with `area` only would be a significant improvement over the null model?

41.7 Sequential Testing in a Regression Model with ANOVA

```
anova(model1)
```

```
Analysis of Variance Table

Response: species
          Df Sum Sq Mean Sq F value Pr(>F)
area       1 145470 145470 39.13 1.8e-06 ***
elevation  1  65664  65664 17.66 0.00032 ***
nearest    1     29      29  0.01 0.93007
scruz      1 14280  14280  3.84 0.06173 .
adjacent   1  66406  66406 17.86 0.00030 ***
Residuals 24  89231    3718
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

41.7.1 Questions about Sequential Testing and ANOVA

25. What conclusions can we draw from the `area` row in the output above?
26. What conclusions can we draw from the `elevation` row?
27. Does `nearest` add statistically significant predictive value to the model including `area` and `elevation`, but none of the other predictors?
28. Does `adjacent` add significant predictive value as last predictor into the model?
29. Where else in the regression output can we find the answer to the previous question?
30. How does the mean square of the residuals (3718) relate to the residual standard error?
31. What percentage of the variation in the species counts is accounted for by `area` alone?
32. What percentage of the variation explained by the kitchen sink model would also be accounted for in a two-predictor regression model including `area` and `elevation` alone?
33. How could we use the original linear model output to whether a model using the four predictors that appear most promising here would be statistically significantly worse than the full model with all five predictors?
34. What does the following output do differently than the output above, and why is that potentially useful here? Why is the p value for `scruz` so different?

```
anova(lm(species ~ area + elevation + adjacent + scruz + nearest, data=gala))
```

```
Analysis of Variance Table
```

```
Response: species
  Df Sum Sq Mean Sq F value Pr(>F)
area      1 145470 145470 39.13 1.8e-06 ***
elevation 1 65664 65664 17.66 0.00032 ***
adjacent   1 73171 73171 19.68 0.00017 ***
scruz     1 7544   7544   2.03 0.16719
nearest    1 0       0       0.00 0.99315
Residuals 24 89231 3718
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

35. Consider the ANOVA below, run on a new model with `elevation` after `adjacent`. What happens? Why?

```
anova(lm(species ~ area + adjacent + elevation + scruz + nearest, data=gala))
```

```
Analysis of Variance Table
```

```
Response: species
  Df Sum Sq Mean Sq F value Pr(>F)
area      1 145470 145470 39.13 1.8e-06 ***
adjacent   1 2850   2850   0.77   0.39
elevation  1 135985 135985 36.58 3.0e-06 ***
scruz     1 7544   7544   2.03   0.17
nearest    1 0       0       0.00   0.99
Residuals 24 89231 3718
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

41.8 An ANOVA table for the Model as a Whole

It's probably also worthwhile to compute a completed ANOVA table for the model as a whole. All elements are in the ANOVA tables above, or the model summary.

Group	DF	SS	MS	F	P
Regression	5	291849	58369.8	15.7	6.838e-07
Residuals	24	89231	3718.0		
Total	29	381080			

36. How did I determine the Mean Square for the Regression model?
 37. What conclusions can we draw from this ANOVA table?

41.9 Assumption Checking for our Galápagos Islands models

Remember that the key assumptions of multiple linear regression are:

- [Linearity] We have also assumed that the structural part of the model is correctly specified (we've included all the predictors that need to be in the model, and expressed them in the most appropriate manner, and we've left out any predictors that don't need to be in the model.)
- [Normality] The regression makes errors that come from a Normal distribution
- [Homoscedasticity = Constant Variance] The regression makes errors that come from a distribution with constant variance at all predictor levels.
- [Independence] The regression errors are independent of each other.

In addition, we need to realize that sometimes a few observations may be particularly problematic. For instance:

1. An observation may simply not fit the model well (i.e. it creates a large residual)
2. An observation may have high leverage over the fit of the model (this happens with observations that have an unusual combination of predictor values, in particular)
3. An observation may actually have high influence on the model (in the sense that whether they are included or excluded has a large impact on the model's fit, and the value of its parameter estimates.)
4. Or any combination of high residual, leverage and influence may occur.

So it is important to check the assumptions that we can with the data we have. Our most important tools are plots and other summaries of residuals, and what are called influence statistics.

41.10 My First Plot: Studentized Residuals vs. Fitted Values

The first diagnostic plot I usually draw for a multiple regression is a scatterplot of the model's **studentized** residuals¹ (on the vertical axis) vs. the model's fitted values (on the horizontal.) This plot can be used to assess potential non-linearity, non-constant variance, and non-Normality in the residuals.

```
gala$stures <- rstudent(model1); gala$fits <- fitted(model1)
ggplot(gala, aes(x = fits, y = stures)) +
  theme_bw() + geom_point(size = 3, shape = 1) +
  geom_smooth(col = "blue", se = FALSE, weight = 0.5) +
```

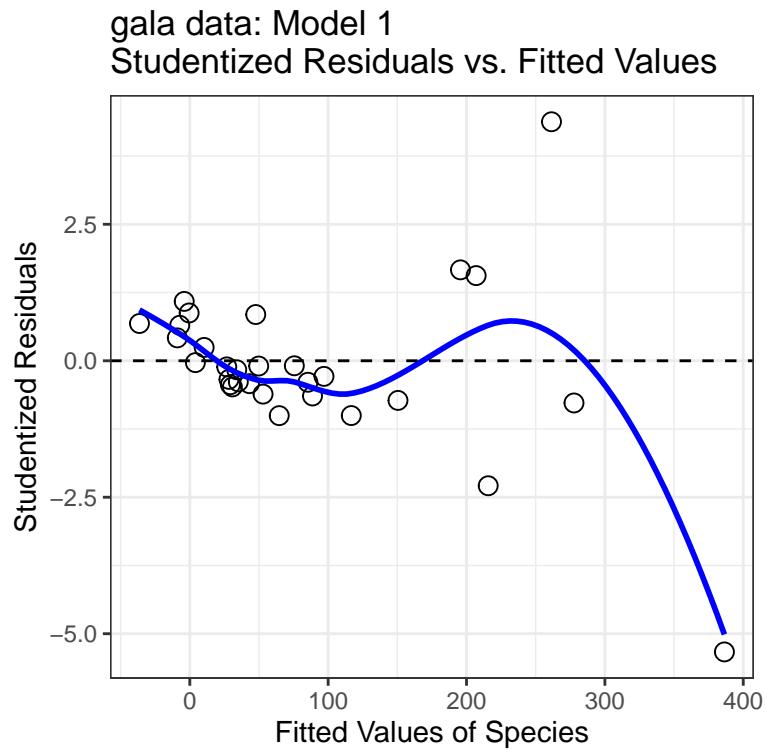
¹More on studentized and standardized residuals later. For now, think of them like z scores.

```

geom_hline(aes(yintercept = 0), linetype = "dashed") +
  labs(x = "Fitted Values of Species",
       y = "Studentized Residuals",
       title = "gala data: Model 1\nStudentized Residuals vs. Fitted Values")

`geom_smooth()` using method = 'loess' and formula 'y ~ x'

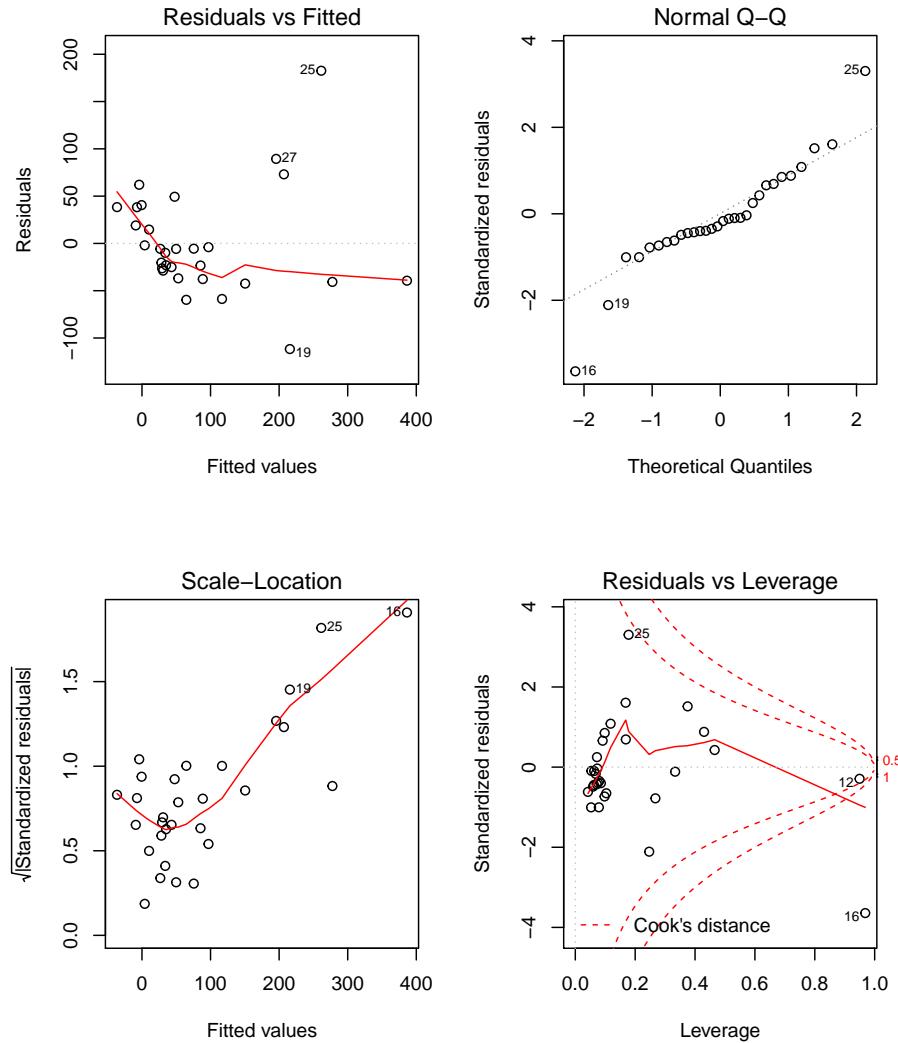
```



41.10.1 Questions about Studentized Residuals vs. Fitted Values

38. Consider the point at bottom right. What can you infer about this observation?
39. Why did I include the dotted horizontal line at Studentized Residual = 0?
40. What is the purpose of the thin blue line?
41. What does this plot suggest about the potential for outliers in the residuals?

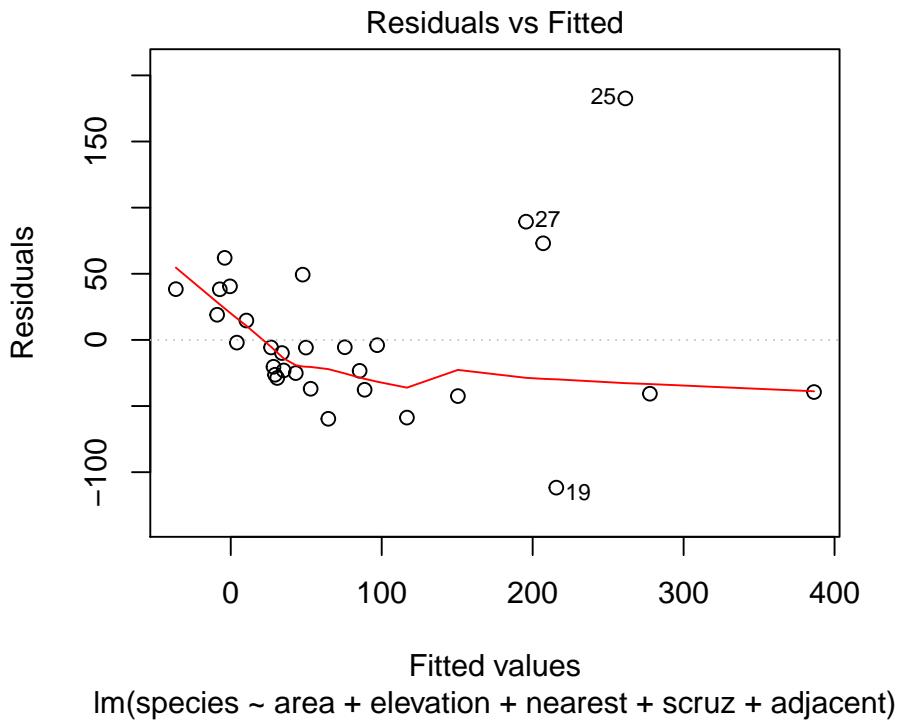
41.11 Automatic Regression Diagnostics for Model 1



41.12 Model 1: Diagnostic Plot 1

As we've seen, the first of R's automated diagnostic plots for a linear model is a plot of the residuals vs. the fitted values.

```
plot(model1, which=1)
```



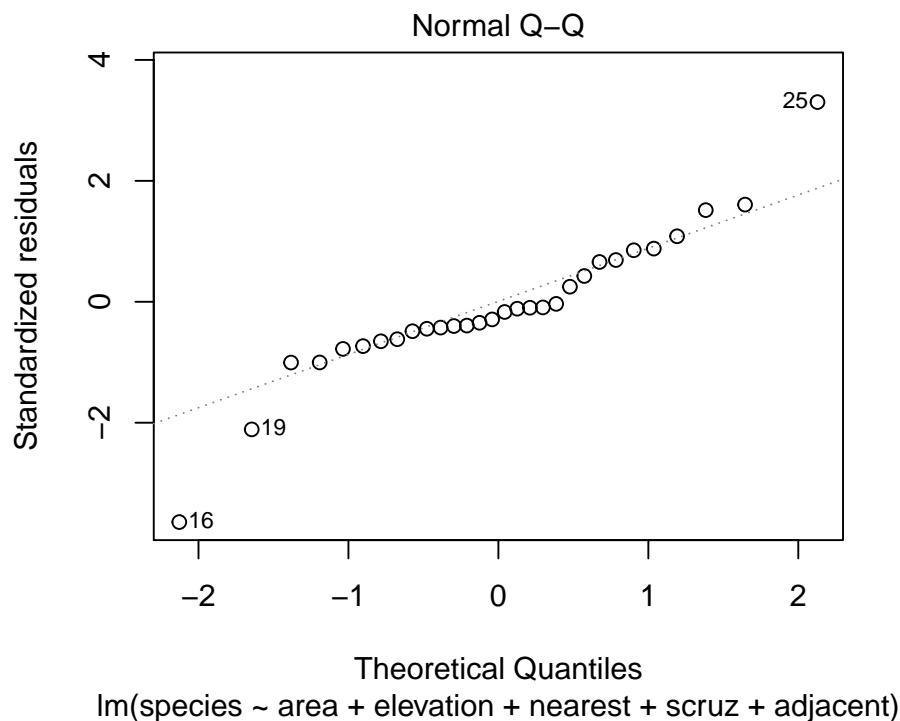
41.12.1 Questions about Diagnostic Plot 1: Residuals vs. Fitted Values

42. What type of regression residuals is R plotting here?
43. Which points are identified by numbers here?
44. Why did R include the gray dotted line at Residual = 0?
45. What is the purpose of the thin red line?
46. What can you tell about the potential for outliers in the model 1 residuals from the plot?
47. What are we looking for in this plot that would let us conclude there were no important assumption violations implied by it? Which assumptions can we assess with it?
48. What would we do if we saw a violation of assumptions in this plot?
49. What are the key differences between this plot and the one I showed earlier?

41.13 Diagnostic Plot 2: Assessing Normality

The second diagnostic plot prepared by R for any linear model using the `plot` command is a Normal Q-Q plot of the standardized residuals from the model.

```
plot(model1, which=2)
```



41.13.1 Questions about Diagnostic Plot 2: Normal Plot of Standardized Residuals

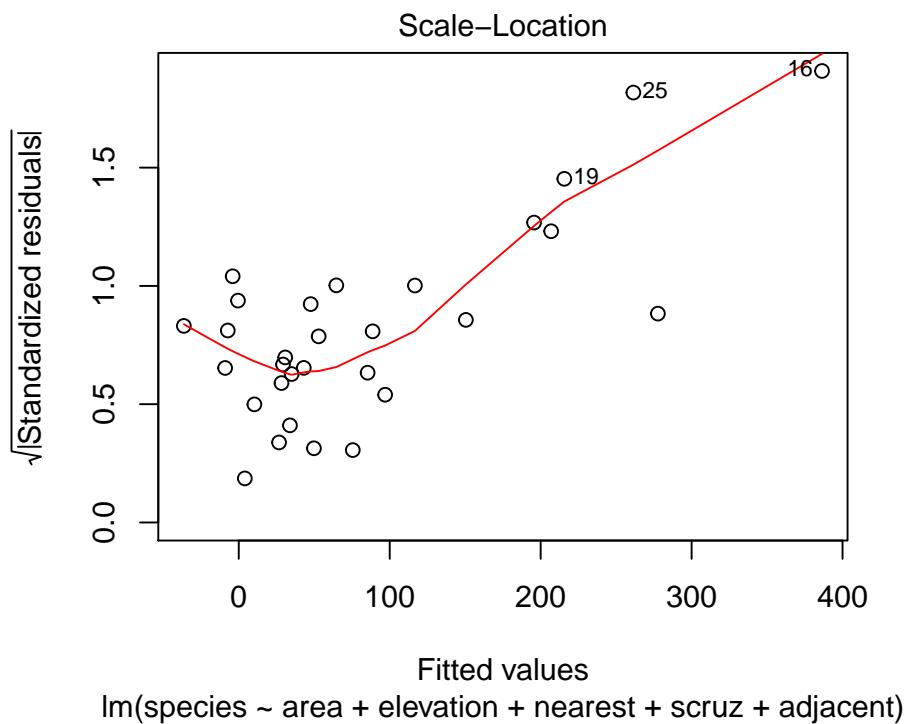
50. Which points are being identified here by number?
51. Which assumption(s) of multiple regression does this plot help us check?
52. What are we looking for in this plot that would let us conclude there were no important assumption violations implied by it?
53. What would we do if we saw a violation of assumptions in this plot?

We could also look at studentized residuals, or we could apply a more complete set of plots and other assessments of normality. Usually, I don't.

41.14 Diagnostic Plot 3: Assessing Constant Variance

The third diagnostic plot prepared by R for any linear model using the `plot` command shows the square root of the model's standardized residuals vs. its fitted values. R calls this a **scale-location plot**.

```
plot(model1, which=3)
```



41.14.1 Questions about Diagnostic Plot 3: Scale-Location Plot

54. Which points are being identified here by number?
55. Which assumption(s) of multiple regression does this plot help us check?
56. What is the role of the thin red line in this plot?
57. What are we looking for in this plot that would let us conclude there were no important assumption violations implied by it?
58. What would we do if we saw a violation of assumptions in this plot?

41.15 Obtaining Fitted Values and Residuals from a Model

Remember that we can use the `fitted` function applied to a model to find the predictions made by the regression model for each of the observations used to create the model.

```
round(fitted(model1), 2)

 1      2      3      4      5      6      7      8      9      10
116.73 -7.27  29.33 10.36 -36.38 43.09 33.92 -9.02 28.31 30.79
     11     12     13     14     15     16     17     18     19     20
 47.66  96.99 -4.03 64.63 -0.50 386.40 88.69  4.04 215.68 150.48
     21     22     23     24     25     26     27     28     29     30
 35.08  75.55 206.95 277.68 261.42  85.38 195.62 49.81  52.94  26.70

gala[1,]

# A tibble: 1 x 10
  id island species area elevation nearest scruz adjacent stures fits
  <int> <fct>    <int> <dbl>       <int>   <dbl> <dbl>    <dbl> <dbl> <dbl>
1     1 Baltra      58  25.1        346     0.6   0.6     1.84 -1.00 117.
```

41.15.1 Questions about Fitted Values

59. Verify that the first fitted value [116.73] is in fact what you get for Baltra (observation 1) when you apply the regression equation:

```
species = 7.07 - 0.02 area + 0.32 elevation
          + 0.009 nearest - 0.24 scruz - 0.07 adjacent
```

We can compare these predictions to the actual observed counts of the number of species on each island. Subtracting the fitted values from the observed values gives us the residuals, as does the `resid` function.

41.15. OBTAINING FITTED VALUES AND RESIDUALS FROM A MODEL623

```
round(resid(model1),2)

 1      2      3      4      5      6      7      8      9
-58.73  38.27 -26.33  14.64  38.38 -25.09 -9.92  19.02 -20.31
   10     11     12     13     14     15     16     17     18
-28.79  49.34 -3.99  62.03 -59.63  40.50 -39.40 -37.69 -2.04
   19     20     21     22     23     24     25     26     27
-111.68 -42.48 -23.08 -5.55  73.05 -40.68 182.58 -23.38  89.38
   28     29     30
  -5.81 -36.94 -5.70
```

41.15.2 Questions about Residuals

60. What does a positive residual indicate?
61. What does a negative residual indicate?
62. The standard deviation of the full set of 30 residuals turns out to be 55.47. How does this compare to the residual standard error?
63. The command below identifies Santa Cruz. What does it indicate about Santa Cruz, specifically?

```
gala$island[which.max(resid(model1))]

[1] SantaCruz
30 Levels: Baltra Bartolome Caldwell Champion Coamano ... Wolf

64. From the results below, what is the model1 residual for Santa Cruz? What does this imply about the species prediction made by Model 1 for Santa Cruz?
```

```
which.max(resid(model1))

25
25

round(resid(model1),2)

 1      2      3      4      5      6      7      8      9
-58.73  38.27 -26.33  14.64  38.38 -25.09 -9.92  19.02 -20.31
   10     11     12     13     14     15     16     17     18
-28.79  49.34 -3.99  62.03 -59.63  40.50 -39.40 -37.69 -2.04
   19     20     21     22     23     24     25     26     27
-111.68 -42.48 -23.08 -5.55  73.05 -40.68 182.58 -23.38  89.38
   28     29     30
  -5.81 -36.94 -5.70

gala[which.max(resid(model1)),]
```

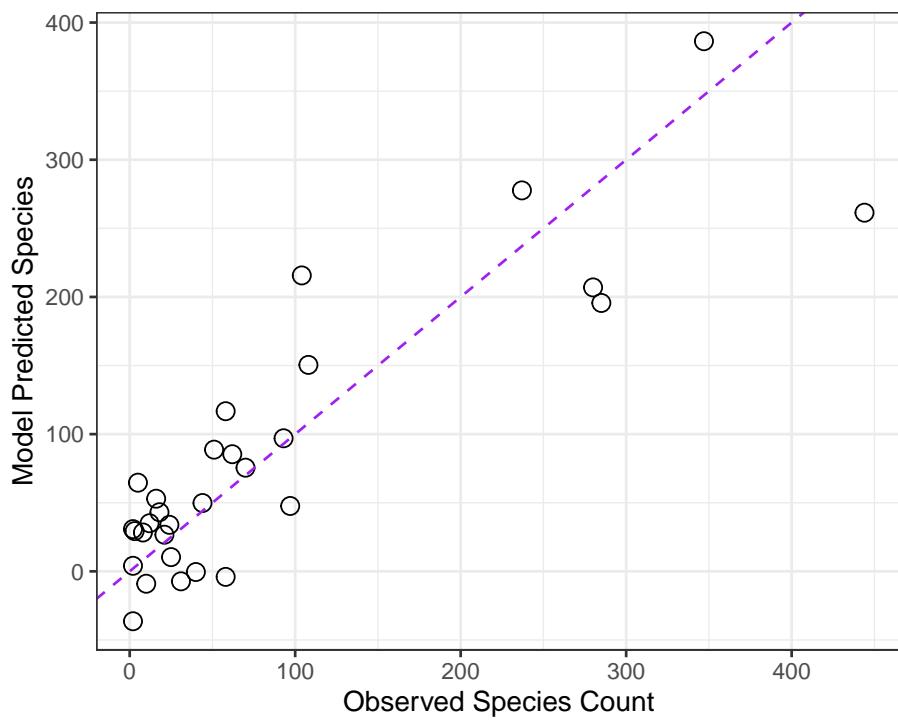
```
# A tibble: 1 x 10
```

	<code>id</code>	<code>island</code>	<code>species</code>	<code>area</code>	<code>elevation</code>	<code>nearest</code>	<code>scruz</code>	<code>adjacent</code>	<code>stures</code>	<code>fits</code>
	<int>	<fct>	<int>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	25	SantaC~	444	904.	864	0.6	0	0.52	4.38	261.

41.16 Relationship between Fitted and Observed Values

We've already seen that the `fitted` command can produce predicted values for each observations used to develop a regression model, and that the `resid` command can produce the residuals (observed - predicted) for those same observations. Returning to our original model1, let's compare the fitted values (stored earlier in `fits`) to the observed values.

```
ggplot(gala, aes(x = species, y = fits)) +
  geom_point(size = 3, shape = 1) + theme_bw() +
  geom_abline(intercept = 0, slope = 1, col = "purple", linetype = "dashed") +
  labs(x = "Observed Species Count", y = "Model Predicted Species")
```



41.16.1 Questions about Fitted and Observed Values

65. Why did I draw the dotted purple line with y-intercept 0 and slope 1? Why is that particular line of interest?
66. If a point on this plot is in the top left here, above the dotted line, what does that mean?
67. If a point is below the dotted line here, what does that mean?
68. How does this plot display the size of an observation's residual?

41.17 Standardizing Residuals

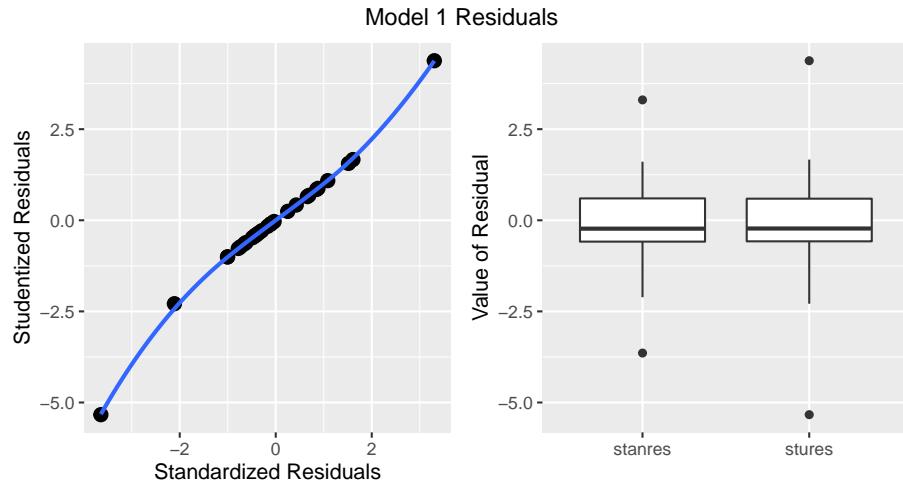
We've already seen that the raw residuals from a regression model can be obtained using the `resid` function. Residuals are defined to have mean 0. This is one of the requirements of the least squares procedure for estimating a linear model, and their true standard deviation is effectively estimated using the residual standard error.

There are two additional types of residuals for us to be aware of: standardized residuals, and studentized (sometimes called externally standardized, or jackknife) residuals. Each approach standardizes the residuals by dividing them by a standard deviation estimate, so the resulting residuals should have mean 0 and standard deviation 1 if assumptions hold.

- **Standardized** residuals are the original (raw) residuals, scaled by a standard deviation estimate developed using the entire data set.
- **Studentized** residuals are the original (raw) residuals, scaled by a standard deviation estimate developed using the entire data set EXCEPT for this particular observation.

The `rstandard` function, when applied to a linear regression model, will generate the standardized residuals, while `rstudent` generates the model's studentized residuals.

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



41.17.1 Questions about Standardized and Studentized Residuals

69. From the plots above, what conclusions can you draw about the two methods of standardizing residuals as they apply in the case of our model1?

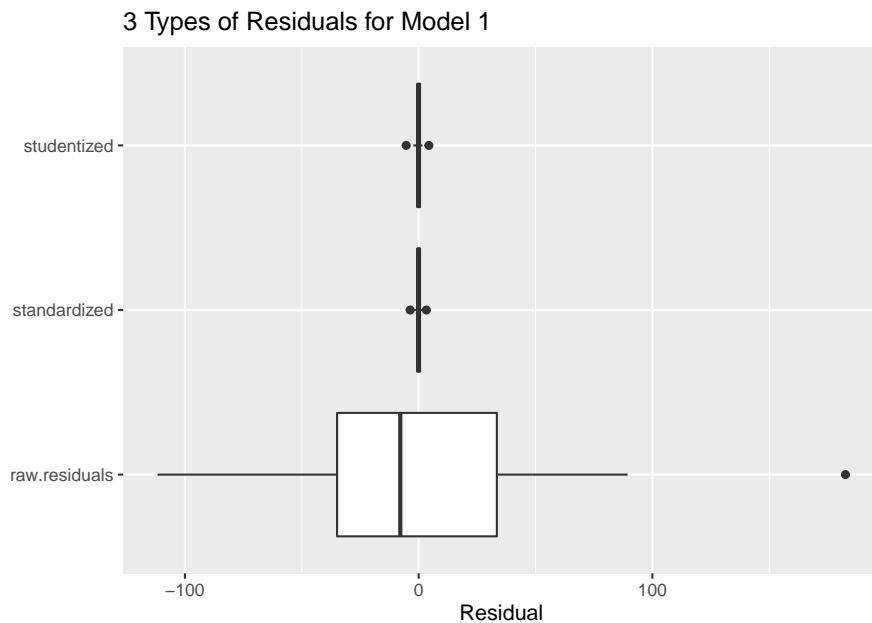
41.18 Three Types of Residuals

```

gala.res <- data.frame(raw.residuals = resid(model1),
                        standardized = rstandard(model1),
                        studentized = rstudent(model1)) %>%tbl_df

gala.res_long <- gather(gala.res, key = "type", value = "res")
ggplot(gala.res_long, aes(x = type, y = res)) +
  geom_boxplot() +
  coord_flip() +
  labs(x = "", y = "Residual", title = "3 Types of Residuals for Model 1")

```



41.18.1 Questions about Three Types of Residuals

70. Consider the three types of residuals, shown above. Can you specify a reason why looking at the raw residuals might be helpful in this case?
71. Why might (either of the two approaches to) standardizing be useful?
72. Does there seem to be a substantial problem with Normality in the residuals?
73. How about the Normality of the studentized residuals? Which seems clearer?

Chapter 42

Influence Measures for Multiple Regression

R can output a series of **influence measures** for a regression model. Let me show you all of the available measures for model 1, but just for three of the data points - #1 (which is not particularly influential) and #12 and #16 (which are).

First, we'll look at the raw data:

```
gala[c(1,12,16),]  
  
# A tibble: 3 x 11  
  id island species   area elevation nearest scrucz adjacent stures   fits  
  <int> <fct>    <int>   <dbl>     <int>   <dbl> <dbl>   <dbl> <dbl>   <dbl>  
1     1 Baltra      58    25.1      346     0.6    0.6    1.84 -1.00   117.  
2    12 Fernan~     93   634.     1494     4.3   95.3  4669.  -0.286   97.0  
3    16 Isabe~     347  4669.     1707     0.7   28.1   634.  -5.33   386.  
# ... with 1 more variable: stanres <dbl>
```

And then, we'll gather the output available in the `influence.measures` function.

```
influence.measures(model1)
```

Here's an edited version of this output...

```
Influence measures of  
lm(formula = species ~ area + elevation + nearest + scrucz + adjacent,  
data = gala) :  
  
  dfb.1_  dfb.area  dfb.elvt dfb.nrst  dfb.scrz  dfb.adjc  
1 -0.15064  0.13572 -0.122412  0.07684  0.084786  1.14e-01  
12  0.16112  0.16395 -0.122578  0.03093 -0.059059 -8.27e-01  
16 -1.18618 -20.87453  4.885852  0.36713 -1.022431 -8.09e-01
```

	dffit	cov.r	cook.d	hat	inf
1	-0.29335	1.0835	1.43e-02	0.0787	
12	-1.24249	25.1101	2.68e-01	0.9497	*
16	-29.59041	0.3275	6.81e+01	0.9685	*

This output presents dfbetas for each coefficient, followed by dffit statistics, covariance ratios, Cook's distance and leverage values (**hat**) along with an indicator of influence.

We'll consider each of these elements in turn.

42.1 DFBETAs

The first part of the influence measures output concerns what are generally called **dfbetas** ...

id	island	dfb.1_	dfb.area	dfb.elvt	dfb.nrst	dfb.scrz	dfb.adjc
1	Baltra	-0.151	0.136	-0.122	0.077	0.085	0.114
12	Fernandina	0.161	0.164	-0.123	0.031	-0.059	-0.827
16	Isabela	-1.186	-20.875	4.886	0.367	-1.022	-0.809

The **dfbetas** look at a standardized difference in the estimate of a coefficient (slope) that will occur if the specified point (here, **island**) is removed from the data set.

- Positive values indicate that deleting the point will yield a smaller coefficient.
- Negative values indicate that deleting the point will yield a larger coefficient.
- If the absolute value of the dfbeta is greater than $2/\sqrt{n}$, where n is the sample size, then the **dfbeta** is considered to be large.

In this case, our cutoff would be $2/\sqrt{30}$ or 0.365, so that the Isabela dfbeta values are all indicative of large influence. Essentially, if we remove Isabela from the data, and refit the model, our regression slopes will change a lot (see below). Fernandina has some influence as well, especially on the **adjacent** coefficient.

Predictor	Coefficient (p) all 30 islands	Coefficient (p) without Isabela
Intercept	7.07 ($p = 0.72$)	22.59 ($p = 0.11$)
area	-0.02 ($p = 0.30$)	0.30 ($p < 0.01$)
elevation	0.32 ($p < 0.01$)	0.14 ($p < 0.01$)
nearest	0.01 ($p = 0.99$)	-0.26 ($p = 0.73$)
scruz	-0.24 ($p = 0.28$)	-0.09 ($p = 0.55$)
adjacent	-0.08 ($p < 0.01$)	-0.07 ($p < 0.01$)

42.2 Other Available Influence Measures

After the dfbetas, the `influence.measures` output presents `dffit`, covariance ratios, Cook's distance and leverage values (`hat`) for each observation, along with an indicator of influence.

id	island	dffit	cov.r	cook.d	hat	inf
1	Baltra	-0.29335	1.0835	1.43e-02	0.0787	
12	Fernandina	-1.24249	25.1101	2.68e-01	0.9497	*
16	Isabela	-29.59041	0.3275	6.81e+01	0.9685	*

42.2.1 Cook's d or Cook's Distance

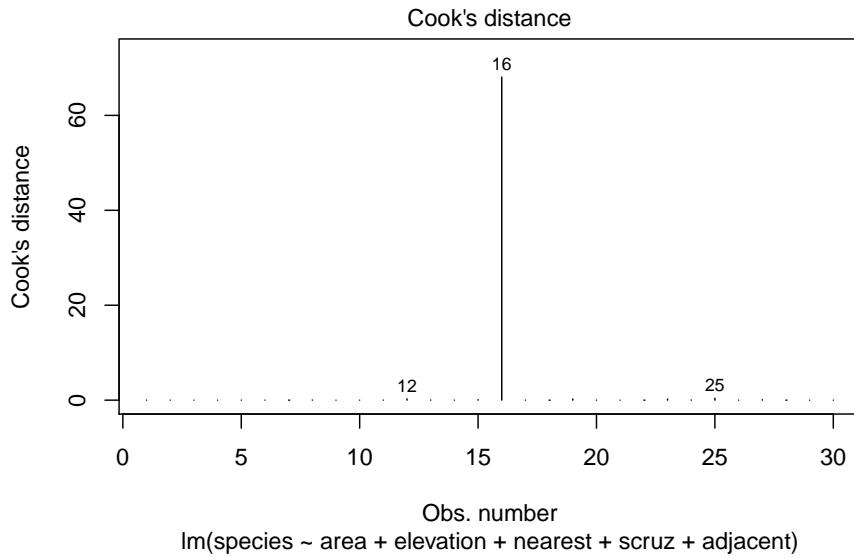
The main measure of influence is Cook's Distance, also called Cook's d. Cook's d provides a summary of the influence of a particular point on all of the regression coefficients. It is a function of the standardized residual and the leverage.

- Cook's distance values greater than 1 are generally indicators of high influence.
- Obviously, Isabela (with a value of Cook's d = 68.1) is a highly influential observation by this measure.

42.2.2 Plotting Cook's Distance

As one of its automated regression diagnostic plots, R will produce an index plot of the Cook's distance values. Note the relatively enormous influence for island 16 (Isabela).

```
plot(model1, which = 4)
```



42.2.3 DFFITS

A similar measure to Cook's distance is called DFFITS. The DFFITS value describes the influence of the point on the fitted value. It's the number of standard deviations that the fitted value changes if the observation is removed. This is defined as a function of the studentized residual and the leverage.

- If the absolute value of DFFITS is greater than $2\sqrt{p/n - p}$, where p is the number of predictors (not including the intercept), we deem the observation influential.
- For the `gala` data, we'd consider any point with DFFITS greater than $2 \times \sqrt{5/(30-5)} = 0.894$ to be influential by this standard, since $n = 30$ and we are estimating $p = 5$ slopes in our model. This is true of both Fernandina and Isabela.

42.2.4 Covariance Ratio

The covariance ratio `cov.r` indicates the role of the observation on the precision of estimation. If `cov.r` is greater than 1, then this observation improves the precision, overall, and if it's less than 1, the observation drops the precision of estimation, and these are the points about which we'll be most concerned.

- As with most of our other influence measures, Isabela appears to be a concern.

42.2.5 Leverage

The `hat` value is a measure of leverage. Specifically, this addresses whether or not the point in question is unusual in terms of its combination of predictor values.

- The usual cutoff for a large leverage value is 2.5 times the average leverage across all observations, where the average leverage is equal to k/n , where n is the number of observations included in the regression model, and k is the number of model coefficients (slopes plus intercept).
- In the `gala` example, we'd regard any observation with a `hat` value larger than $2.5 \times 6/30 = 0.5$ to have large leverage. This includes Fernandina and Isabela.

42.2.6 Indicator of Influence

The little asterisk indicates an observation which is influential according to R's standards for any of these measures. You can take the absence of an asterisk as a clear indication that a point is NOT influential. Points with asterisks may or may not be influential in an important way. In practice, I usually focus on the Cook's distance to make decisions about likely influence, when the results aren't completely clear.

Chapter 43

Building Predictions from our models

The `predict` function, when applied to a linear regression model, produces the fitted values, just as the `fitted` function did, and, as we've seen, it can be used to generate *prediction* intervals for a single new observation, or *confidence* intervals for a group of new observations with the same predictor values.

43.1 Predictions for a “typical” island

Let us, just for a moment, consider a “typical” island, exemplified by the median value of all the predictors¹. There's a trick to creating this and dumping it in a vector I will call `x.medians`.

```
x <- model.matrix(model1)
x.medians <- apply(x, 2, function(x) median(x))
x.medians
```

(Intercept)	area	elevation	nearest	scruz	adjacent
1.00	2.59	192.00	3.05	46.65	2.59

We want to use the model to predict our outcome (`species`) on the basis of the inputs above: a new island with values of all predictors equal to the median of the existing islands. As before, building an interval forecast around a fitted value requires us to decide whether we are:

- predicting the number of species for one particular island with the specified characteristics (in which case we use something called a prediction interval) or

¹This approach is motivated by @Faraway2015, pp. 52-53.

- predicting the mean number of species across all islands that have the specified characteristics (in which case we use the confidence interval).

```
newdata <- data.frame(t(x.medians))
predict(model1, newdata, interval="prediction", level = 0.95)

  fit    lwr   upr
1 57 -72.1 186

predict(model1, newdata, interval="confidence", level = 0.95)

  fit    lwr   upr
1 57 28.5 85.4
```

43.1.1 Questions about the Prediction and Confidence Interval Methods

74. What is the 95% prediction interval for this new observation? Does that make sense?
75. Which interval (prediction or confidence) is wider? Does that make sense?
76. Is there an island that has characteristics that match our new medians variable?
77. What happens if we don't specify new data in making a prediction?

43.2 Making a Prediction with New Data

78. How does the output below help us to make a prediction with a new data point, or series of them? Interpret the resulting intervals.

```
newdata2 <- data.frame(area = 2, elevation = 100, nearest = 3,
                      scruz = 5, adjacent = 1)
predict(model1, newdata2, interval="prediction", level = 0.95)

  fit    lwr   upr
1 37.7 -92.5 168

predict(model1, newdata2, interval="confidence", level = 0.95)

  fit    lwr   upr
1 37.7  4.39  71
```

Chapter 44

Standardizing/Rescaling in Regression Models

44.1 Scaling Predictors using Z Scores: Semi-Standardized Coefficients

We know that the interpretation of the coefficients in a regression model is sensitive to the scale of the predictors. We have already seen how to “standardize” each predictor by subtracting its mean and dividing by its standard deviation.

- Each coefficient in this semi-standardized model has the following interpretation: the expected difference in the outcome, comparing units (subjects) that differ by one standard deviation in the variable of interest, but for which all other variables are fixed at their average.
- Remember also that the intercept in such a model shows the mean outcome across all subjects.

Consider a two-variable model, using `area` and `elevation` to predict the number of `species`...

```
model2 <- lm(species ~ area + elevation, data=gala)
summary(model2)
```

Call:

```
lm(formula = species ~ area + elevation, data = gala)
```

Residuals:

Min	1Q	Median	3Q	Max
-192.62	-33.53	-19.20	7.54	261.51

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	17.1052	20.9421	0.82	0.4212
area	0.0188	0.0259	0.72	0.4748
elevation	0.1717	0.0532	3.23	0.0032 **

Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .
	.	.	.	1

Residual standard error: 79.3 on 27 degrees of freedom

Multiple R-squared: 0.554, Adjusted R-squared: 0.521

F-statistic: 16.8 on 2 and 27 DF, p-value: 1.84e-05

Now compare these results to the ones we get after scaling the area and elevation variables. Remember that the `scale` function centers a variable on zero by subtracting the mean from each observation, and then scales the result by dividing by the standard deviation. This ensures that each regression input has mean 0 and standard deviation 1, and is thus a *z score*.

```
model2.z <- lm(species ~ scale(area) + scale(elevation), data=gala)
summary(model2.z)
```

Call:

```
lm(formula = species ~ scale(area) + scale(elevation), data = gala)
```

Residuals:

Min	1Q	Median	3Q	Max
-192.62	-33.53	-19.20	7.54	261.51

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.2	14.5	5.88	2.9e-06 ***
scale(area)	16.2	22.4	0.72	0.4748
scale(elevation)	72.4	22.4	3.23	0.0032 **

Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .
	.	.	.	1

Residual standard error: 79.3 on 27 degrees of freedom

Multiple R-squared: 0.554, Adjusted R-squared: 0.521

F-statistic: 16.8 on 2 and 27 DF, p-value: 1.84e-05

44.1.1 Questions about the Semi-Standardized Model

79. What changes after centering and rescaling the predictors, and what does not?

80. Why might rescaling like this be a helpful thing to do if you want to compare predictors in terms of importance?

44.2 Fully Standardized Regression Coefficients

Suppose we standardize the coefficients by also taking centering and scaling (using the z score) the outcome variable: `species`, creating a **fully standardized** model.

```
model2.zout <- lm(scale(species) ~
                     scale(area) + scale(elevation), data=gala)
summary(model2.zout)
```

Call:

```
lm(formula = scale(species) ~ scale(area) + scale(elevation),
   data = gala)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.6803	-0.2925	-0.1675	0.0658	2.2813

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.59e-17	1.26e-01	0.00	1.0000
scale(area)	1.42e-01	1.96e-01	0.72	0.4748
scale(elevation)	6.32e-01	1.96e-01	3.23	0.0032 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.692 on 27 degrees of freedom

Multiple R-squared: 0.554, Adjusted R-squared: 0.521

F-statistic: 16.8 on 2 and 27 DF, p-value: 1.84e-05

44.2.1 Questions about the Standardized Model

81. How do you interpret the value 0.142 of the `scale(area)` coefficient here?

You may want to start by reviewing the summary of the original `gala` data shown here.

```
summary(gala[c("species", "area", "elevation")])
```

species	area	elevation
Min. : 2	Min. : 0	Min. : 25
1st Qu.: 13	1st Qu.: 0	1st Qu.: 98

```

Median : 42   Median :  3   Median : 192
Mean   : 85   Mean   : 262   Mean   : 368
3rd Qu.: 96   3rd Qu.: 59   3rd Qu.: 435
Max.   :444   Max.   :4669   Max.   :1707

```

82. How do you interpret the value 0.632 of the `scale(elevation)` coefficient in the standardized model?
83. What is the intercept in this setting? Will this be the case whenever you scale like this?
84. What are some of the advantages of looking at scaled regression coefficients?
85. Why are these called *fully* standardized coefficients while the previous page described semi-standardized coefficients?
86. What would motivate you to use one of these two methods of standardization (fully standardized or semi-standardized) vs. the other?

44.3 Robust Standardization of Regression Coefficients

Another common option for scaling is to specify lower and upper comparison points, perhaps by comparing the impact of a move from the 25th to the 75th percentile for each variable, while holding all of the other variables constant.

Occasionally, you will see robust semi-standardized regression coefficients, which measure the increase in the outcome, Y, associated with an increase in that particular predictor of one IQR (inter-quartile range).

```

gala$area.scaleiqr <- (gala$area - mean(gala$area)) / IQR(gala$area)
gala$elevation.scaleiqr <- (gala$elevation - mean(gala$elevation)) /
                                IQR(gala$elevation)

model2.iqr <- lm(species ~ area.scaleiqr + elevation.scaleiqr,
                  data=gala)
summary(model2.iqr)

```

```

Call:
lm(formula = species ~ area.scaleiqr + elevation.scaleiqr, data = gala)

Residuals:
    Min      1Q  Median      3Q     Max 
-192.62 -33.53 -19.20    7.54  261.51 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 85.23      14.48    5.88  2.9e-06 *** 

```

```

area.scaleiqr      1.11      1.53      0.72   0.4748
elevation.scaleiqr 57.96     17.95     3.23   0.0032 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 79.3 on 27 degrees of freedom
Multiple R-squared:  0.554, Adjusted R-squared:  0.521
F-statistic: 16.8 on 2 and 27 DF,  p-value: 1.84e-05

```

44.3.1 Questions about Robust Standardization

87. How should we interpret the 57.96 value for the scaled `elevation` variable?
 You may want to start by considering the summary of the original elevation data below.

```
summary(gala$elevation)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
25	98	192	368	435	1707

A robust standardized coefficient analysis measures the increase in Y (in IQR of Y) associated with an increase in the predictor of interest of one IQR.

```

gala$species.scaleiqr <- (gala$species - mean(gala$species)) / IQR(gala$species)
model2.iqrout <- lm(species.scaleiqr ~ area.scaleiqr + elevation.scaleiqr, data=gala)
model2.iqrout

```

Call:

```
lm(formula = species.scaleiqr ~ area.scaleiqr + elevation.scaleiqr,
  data = gala)
```

Coefficients:

(Intercept)	area.scaleiqr	elevation.scaleiqr
-1.01e-16	1.34e-02	6.98e-01

88. What can we learn from the R output above?

44.4 Scaling Inputs by Dividing by 2 Standard Deviations

It turns out that standardizing the inputs to a regression model by dividing by a standard deviation creates some difficulties when you want to include a binary predictor in the model.

Instead, Andrew Gelman recommends that you consider centering all of the predictors (binary or continuous) by subtracting off the mean, and then, for the non-binary predictors, also dividing not by one, but rather by two standard deviations.

- Such a standardization can go a long way to helping us understand a model whose predictors are on different scales, and provides an interpretable starting point.
- Another appealing part of this approach is that in the `arm` library, Gelman and his colleagues have created an R function called `standardize`, which can be used to automate the process of checking coefficients that have been standardized in this manner, after the regression model has been fit.

```
model2
```

```
Call:
lm(formula = species ~ area + elevation, data = gala)

Coefficients:
(Intercept)      area    elevation
17.1052        0.0188     0.1717

arm::standardize(model2)
```

```
Call:
lm(formula = species ~ z.area + z.elevation, data = gala)

Coefficients:
(Intercept)      z.area   z.elevation
85.2          32.5       144.8
```

44.4.1 Questions about Standardizing by Dividing by Two SD

89. How does this result compare to the semi-standardized regression coefficients we have seen on the previous few pages?
90. How should we interpret the `z.area` coefficient of 32.5 here? Again, you may want to start by obtaining a statistical summary of the original `area` data, as shown below.

```
summary(gala$area)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	3	262	59	4669

To standardize the outcome in this way, as well, we use

44.4. SCALING INPUTS BY DIVIDING BY 2 STANDARD DEVIATIONS 643

```
arm::standardize(model2, standardize.y=TRUE)
```

Call:

```
lm(formula = z.species ~ z.area + z.elevation, data = gala)
```

Coefficients:

(Intercept)	z.area	z.elevation
1.65e-19	1.42e-01	6.32e-01

91. How should we interpret the `z.area` coefficient of 0.142 here?
92. How does these relate to the standardized regression coefficients we've seen before?

This is Work in Progress

Dr. Love is revising these notes and adding new materials regularly.

Materials will continue to be posted leading up to the start of class, as well as over the course of the Fall semester.

Baumer, Benjamin S., Daniel T. Kaplan, and Nicholas J. Horton. 2017. *Modern Data Science with R*. Boca Raton, FL: CRC Press. <https://mdsr-book.github.io/>.

Bernard, Gordon R., Arthur P. Wheeler, James A. Russell, Roland Schein, Warren R. Summer, Kenneth P. Steinberg, William J. Fulkerson, et al. 1997. “The Effects of Ibuprofen on the Physiology and Survival of Patients with Sepsis.” *New England Journal of Medicine* 336: 912–18. <http://www.nejm.org/doi/full/10.1056/NEJM199703273361303#t=article>.

Bilder, Christopher R., and Thomas M. Loughlin. 2015. *Analysis of Categorical Data with R*. Boca Raton, FL: CRC Press. <http://www.chrisbilder.com/categorical/>.

Bisgaier, Joanna, and Karin V. Rhodes. 2011. “Auditing Access to Specialty Care for Children with Public Insurance.” *New England Journal of Medicine* 364: 2324–33. <https://www.nejm.org/doi/full/10.1056/neJMs1013285>.

Bock, David E., Paul F. Velleman, and Richard D. De Veaux. 2004. *Stats: Modelling the World*. Boston MA: Pearson Addison-Wesley.

Cohen, Jacob. 1988. *Statistical Power Analysis for the Behavioral Sciences*. Second. New York: Lawrence Erlbaum Associates (now Routledge).

Çetinkaya-Rundel, Mine. 2017. “Teaching Data Science to New useRs.” bit.ly/user2017.

Diez, David M., Christopher D. Barr, and Mine Çetinkaya-Rundel. n.d. *Open-Intro Statistics*. Third. https://www.openintro.org/stat/textbook.php?stat_book=os.

Dupont, William D. 2002. *Statistical Modeling for Biomedical Researchers*. New York: Cambridge University Press.

646CHAPTER 44. STANDARDIZING/RESCALING IN REGRESSION MODELS

- Efron, Bradley. 1979. "Bootstrap Methods: Another Look at the Jackknife." *Annals of Statistics* 7(1): 1–26. <https://projecteuclid.org/euclid-aos/1176344552>.
- Faraway, Julian J. 2015. *Linear Models with R*. Second. Boca Raton, FL: CRC Press.
- Fox, John, and Sanford Weisberg. 2011. *An R Companion to Applied Regression*. Second. Thousand Oaks CA: Sage. <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>.
- Gelman, Andrew, and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel-Hierarchical Models*. New York: Cambridge University Press. <http://www.stat.columbia.edu/~gelman/arm/>.
- Gelman, Andrew, and Deborah Nolan. 2017. *Teaching Statistics: A Bag of Tricks*. Second. Oxford, UK: Oxford University Press.
- Good, Phillip I. 2005. *Introduction to Statistics Through Resampling Methods and R/S-Plus*. Hoboken, NJ: Wiley.
- Good, Phillip I., and James W. Hardin. 2006. *Common Errors in Statistics (and How to Avoid Them)*. Second. Hoboken, NJ: Wiley.
- Grolemund, Garrett, and Hadley Wickham. 2019. *R for Data Science*. O'Reilly. <http://r4ds.had.co.nz/>.
- Harrell, Frank E., and James C. Slaughter. 2019. *Biostatistics for Biomedical Research*. Vanderbilt University School of Medicine. biostat.mc.vanderbilt.edu/ClinStat.
- Ismay, Chester, and Albert Y. Kim. 2019. *ModernDive: Statistical Inference via Data Science*. <http://moderndive.com/>.
- Morton, D., A. Saah, S. Silberg, W. Owens, M. Roberts, and M. Saah. 1982. "Lead Absorption in Children of Employees in a Lead Related Industry." *American Journal of Epidemiology* 115: 549–55.
- Norman, Geoffrey R., and David L. Streiner. 2014. *Biostatistics: The Bare Essentials*. Fourth. People's Medical Publishing House.
- Pagano, Marcello, and Kimberlee Gauvreau. 2000. *Principles of Biostatistics*. Second. Duxbury Press.
- Pruzek, Robert M., and James E. Helmreich. 2009. "Enhancing Dependent Sample Analyses with Graphics." *Journal of Statistics Education* 17(1). <http://ww2.amstat.org/publications/jse/v17n1/helmreich.html>.
- Ramsey, Fred L., and Daniel W. Schafer. 2002. *The Statistical Sleuth: A Course in Methods of Data Analysis*. Second. Pacific Grove, CA: Duxbury.
- Vittinghoff, Eric, David V. Glidden, Stephen C. Shiboski, and Charles E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Multivariate Methods*. Second. New York: Springer.

44.4. SCALING INPUTS BY DIVIDING BY 2 STANDARD DEVIATIONS 647

vival, and Repeated Measures Models. Second. Springer-Verlag, Inc. <http://www.biostat.ucsf.edu/vgsm/>.

Wainer, Howard. 1997. *Visual Revelations: Graphical Tales of Fate and Deception from Napoleon Bonaparte to Ross Perot.* New York: Springer-Verlag.

———. 2005. *Graphic Discovery: A Trout in the Milk and Other Visual Adventures.* Princeton, NJ: Princeton University Press.

———. 2013. *Medical Illuminations: Using Evidence, Visualization and Statistical Thinking to Improve Healthcare.* New York: Oxford University Press.

Yamada, SB, and EG Boulding. 1998. "Claw Morphology, Prey Size Selection and Foraging Efficiency in Generalist and Specialist Shell-Breaking Crabs." *Journal of Experimental Marine Biology and Ecology* 220: 191–211. http://www.science.oregonstate.edu/~yamadas/SylviaCV/BehrensYamada_Boulding1998.pdf.