Stat 241: Statistics for the Physical Sciences and Engineering

R Pruim

January 2021

Contents

0 Where do numbers come from? 7

1 Graphical Summaries of Data 9

1.1 Getting Started With RStudio . . . 9
1.2 Data in R . . . 12
1.3 Graphical and Numerical Summaries of Data . . . 16
1.4 Scatterplots . . . 17
1.5 Graphing the Distribution of One Variable . . . 21
1.6 Labeling plots . . . 26
1.7 Exporting Plots . . . 26
1.8 Reproducible Research . . . 27
1.9 Getting Help in RStudio . . . 30
1.10 Graphical Summaries – Important Ideas . . . 31
1.11 Exercises . . . 33

2 Numerical Summaries 35

2.1 Tabulating Data . . . 35
2.2 Working with Pre-Tabulated Data . . . 38
2.3 Summarizing Distributions of Quantitative Variables . . . 39
2.4 Measures of Center . . . 40
2.5 Measures of Spread . . . 40
2.6 Summarizing Categorical Variables . . . 45

2.7 Relationships Between Two Variables . . . 45
2.8 Exercises . . . 46

3 Probability **47**
3.1 Key Definitions and Ideas . . . 47
3.2 Calculating Probabilities Empirically . . . 49
3.3 Calculating Probabilities Theoretically . . . 52
3.4 Conditional Probability . . . 56
3.5 Exercises . . . 61

4 Random Variables **65**
4.1 Discrete Random Variables . . . 65
4.2 Continuous Random Variables . . . 66
4.3 Mean and Variance . . . 72
4.4 Some Important Families of Distributions . . . 76
4.5 Fitting Distributions to Data . . . 88
4.6 Quantile-Quantile Plots . . . 96
4.7 Exercises . . . 102

5 Transformation and Combinations of Random Variables **107**
5.1 Simulations . . . 107
5.2 Propagation of Mean and Variance . . . 109
5.3 Normal distributions are special . . . 111
5.4 Estimating the Mean of a Population by Sampling . . . 114
5.5 Exercises . . . 123

6 Propagation of Uncertainty **131**
6.1 Error and Uncertainty . . . 131
6.2 An Example: Estimating the number of dimes in a sack of dimes . . . 132
6.3 Reporting Measurements and Estimates . . . 136
6.4 Additional Propagation of Uncertainty Examples . . . 138
6.5 Experimental Error and Its Causes . . . 140
6.6 Uncertainty – How Much Error Might There Be? . . . 142
6.7 Exercises . . . 146

7 Linear Models **149**

7.1 The Simple Linear Regression Model . . . 149
7.2 Fitting the Simple Linear Model . . . 150
7.3 Estimating the Response . . . 155
7.4 Parameter Estimates . . . 156
7.5 Checking Assumptions . . . 158
7.6 How Good Are Our Estimates? . . . 162
7.7 Exercises . . . 166

8 Beyond Linear Regression **171**

8.1 How big is your R^2? . . . 171
8.2 Violations of Linear Regression Assumptions . . . 173
8.3 Non-Normal Errors . . . 174
8.4 Non-Independence of Errors . . . 174
8.5 Heteroscedasticity (Non-constant Error Variance) . . . 177
8.6 Non-linear Relationships . . . 178
8.7 Transformations in Linear Regression . . . 178
8.8 Nonlinear Least Squares . . . 191
8.9 Exercises . . . 199

9 Hypothesis Testing **203**

9.1 Experimental Design in Statistics . . . 203
9.2 Coins and Cups . . . 204
9.3 A General Framework . . . 207
9.4 Statistical Significance . . . 210
9.5 T-tests . . . 210
9.6 Connection to Confidence Intervals . . . 222
9.7 Exercises . . . 224

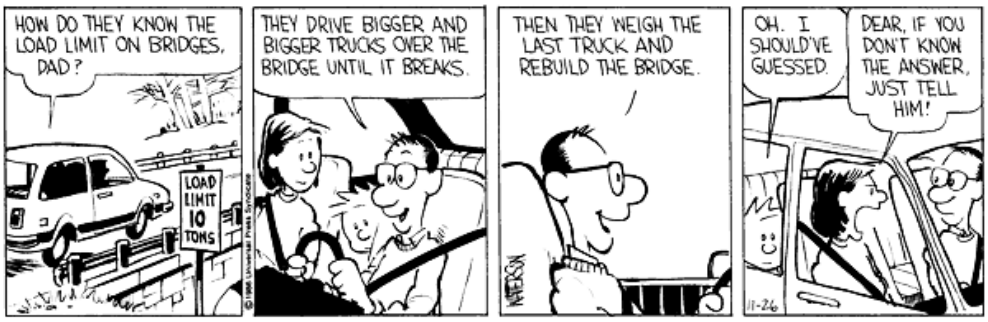
10 More Examples **227**

10.1 Heat Exchanger Example . . . 227
10.2 Standard Errors in `fitdistr()` output . . . 238
10.3 R^2 . . . 238
10.4 Exercises . . . 241

0

Where do numbers come from?

Scientists and engineers work with numbers constantly. Physical constants, values for the specific heat index or measures of strength or flexibility of some material, resistance of some component in an electrical device, etc., etc. Most of these numbers come from some process that generated data and led to a calculation that produced the number. Like this one ...



Thought experiment – How many dimes?

Here's a thought experiment for you. Suppose a middle school class has collected a large number of dimes (10-cent coinds) in a sack. Before bringing the money to the bank, they would like to estimate how many dimes they have (using tools and methods that 6th graders have at their disposal). You've been brought in to consult with them about how they should do this.

1. What method would you suggest? Why?
2. What other methods would be possible? What makes your proposed method better?
3. For your favorite method and others, identify factors that lead the resulting estimate to be different from the exact number of dimes in the sack.

Some important terms

estimand/measureand The number we want to know. The "truth." In our example this is the number of

dimes in the bag. Typically this will be a number that describes some process or population, and typically it will be impossible to know the value exactly.

estimate/measurement The value calculated from our data. This may be as simple as recording a value reported by some device, or it may involve recording multiple values, perhaps of multiple variables, maybe at multiple times, and making some computations with that data.

error The difference between the estimate and the estimand. Because we don't know the estimand exactly, we can't know the error exactly either. But thinking about what the error could be is a big part of understanding the statistical properties of an estimation method. Generally, we want methods where errors tend to be small (so our estimate is "likely to be close to the estimand") and centered around 0 (so we're "right on average").

systematic (component of) error a component of error that makes our estimate biased – in other words, leads the estimate to be either an over- or under-estimate. For example, neglecting the weight of the sack would lead us to overestimate the weight of the dimes, and therefore overestimate the number of dimes. Another way to express this idea is "a tendency to be off in a certain direction."

random (component of) error a component of error that leads to variability in estimates (but not a particular tendency toward over- or under-estimation). If random errors are larger, there will be more variability in estimates, so we will be less confident that the estimand and estimate are close together – although some estimates may still be very close to the estimand, just by chance.

One of the big questions in statistics is this: *What does our estimate tell us about the estimand?* We will eventually learn techniques for quantifying (and attempting to reduce) the effects of error in our measurements.

1 Graphical Summaries of Data

1.1 Getting Started With RStudio

RStudio is an integrated development environment (IDE) for R, a freely available language and environment for statistical computing and graphics. Both are freely available for Mac, PC, and Linux.

In addition to running RStudio on your local machine, you have the option of accessing an RStudio server via a web browser. (For best results, avoid Internet Explorer.)

1.1.1 Using R as a calculator

Notice that RStudio divides its world into four panels. Several of the panels are further subdivided into multiple tabs. The console panel is where we type commands that R will execute.

R can be used as a calculator. Try typing the following commands in the console panel.

```
5 + 3
```

```
## [1] 8
```

```
15.3 * 23.4
```

```
## [1] 358
```

```
sqrt(16)
```

```
## [1] 4
```

You can save values to named variables for later reuse

```
product = 15.3 * 23.4        # save result
product                      # show the result
```

```
## [1] 358
```

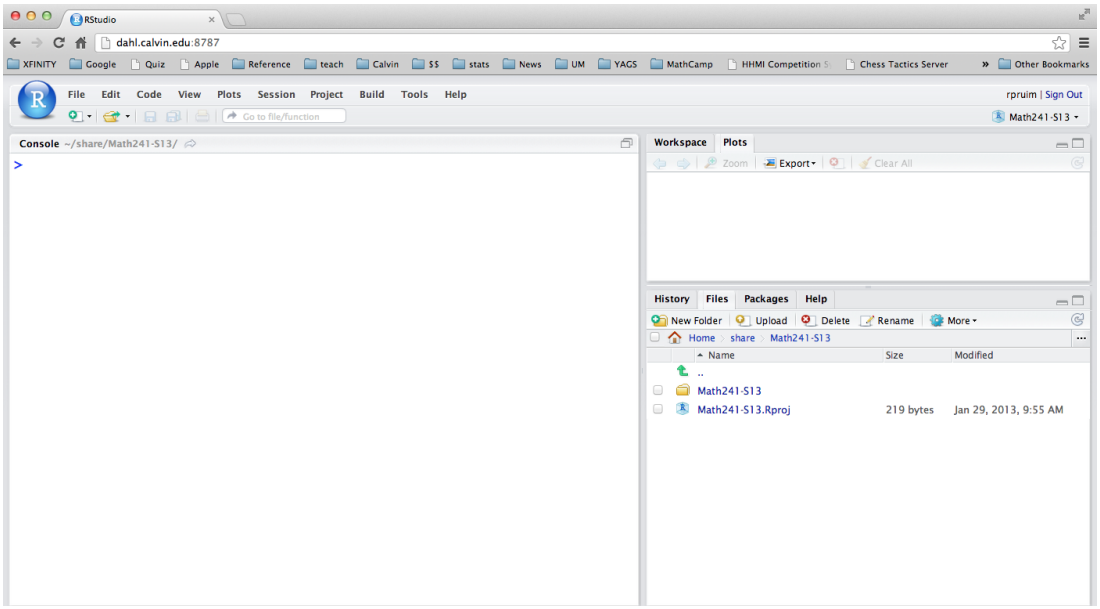


Figure 1.1: Welcome to RStudio.

```
product <- 15.3 * 23.4      # <- is assignment operator, same as =
product

## [1] 358

15.3 * 23.4 -> newproduct   # -> assigns to the right
newproduct

## [1] 358

.5 * product                # half of the product

## [1] 179

log(product)                # (natural) log of the product

## [1] 5.881

log10(product)              # base 10 log of the product

## [1] 2.554

log(product, base = 2)        # base 2 log of the product

## [1] 8.484
```

The semi-colon can be used to place multiple commands on one line. One frequent use of this is to save and print a value all in one go:

```
15.3 * 23.4 -> product; product    # save result and show it

## [1] 358
```

1.1.2 Loading packages

R is divided up into packages. A few of these are loaded every time you run R, but most have to be selected. This way you only have as much of R as you need.

In the Packages tab, check the boxes next to the following packages to load them:

- `mosaic` (a package from Project MOSAIC)
- `DAAG` (a package that goes with the book *Data Analysis and Graphic*)

You an also load packages by typing, for example

```
library(DAAG)       # loads the DAAG package if it is not already loaded
```

1.1.3 Four Things to Know About R

1. R is case-sensitive

   If you mis-capitalize something in R, it won't do what you want.

2. Functions in R use the following syntax:

   ```
   functionname( argument1, argument2, ... )
   ```

   - The arguments are always *surrounded by (round) parentheses* and *separated by commas.*
     Some functions (like `data()`) have no required arguments, but you still need the parentheses.
   - If you type a function name without the parentheses, you will see the *code* for that function – which probably isn't what you want at this point.

3. TAB completion and arrows can improve typing speed and accuracy.

   If you begin a command and hit the TAB key, R will show you a list of possible ways to complete the command. If you hit TAB after the opening parenthesis of a function, it will show you the list of arguments it expects. The up and down arrows can be used to retrieve past commands.

4. Hit ESCAPE to break out of a mess.

   If you get into some sort of mess typing (usually indicated by extra '+' signs along the left edge, indicating that R is waiting for more input – perhaps because you have some sort of error in what has gone before), you can hit the escape key to get back to a clean prompt.

1.2 Data in R

1.2.1 Data Frames

Most often, data sets in R are stored in a structure called a **data frame**. A data frame is designed to hold "rectangular data". The people or things being measured or observed are called **observational units** (or subjects or cases when they are people). Each observational unit is represented by one row. The different pieces of information recorded for each observational unit are stored in separate columns, called **variables**.

1.2.2 Data in Packages

There are a number of data sets built into R and many more that come in various add on packages.

You can see a list of data sets in a particular package like this:

```
library(mosaicData)             # load the package
data(package = "mosaicData")    # see what data sets are in it
```

You can find a longer list of all data sets available in any loaded package using

```
data()
```

1.2.3 The HELPrct data set

The `HELPrct` data frame from the `mosaic` package contains data from the Health Evaluation and Linkage to Primary Care randomized clinical trial. You can find out more about the study and the data in this data frame by typing

```
?HELPrct
```

Among other things, this will tell us something about the subjects (observational units) in this study:

> Eligible subjects were adults, who spoke Spanish or English, reported alcohol, heroin or cocaine as their first or second drug of choice, resided in proximity to the primary care clinic to which they would be referred or were homeless. Patients with established primary care relationships they planned to continue, significant dementia, specific plans to leave the Boston area that would prevent research participation, failure to provide contact information for tracking purposes, or pregnancy were excluded.
>
> Subjects were interviewed at baseline during their detoxification stay and follow-up interviews were undertaken every 6 months for 2 years.

It is often handy to look at the first few rows of a data frame. It will show you the names of the variables and the kind of data in them:

```
head(HELPrct)
```

```
##   age anysubstatus anysub cesd d1 daysanysub dayslink drugrisk e2b female    sex g1b
## 1  37            1    yes   49  3        177      225        0  NA      0   male yes
```

```
## 2  37           1    yes   30 22          2       NA        0  NA      0   male yes
## 3  26           1    yes   39  0          3      365       20  NA      0   male  no
## 4  39           1    yes   15  2        189      343        0   1      1 female  no
## 5  32           1    yes   39 12          2       57        0   1      0   male  no
## 6  47           1    yes    6  1         31      365        0  NA      1 female  no
##   homeless i1 i2 id indtot linkstatus link    mcs   pcs pss_fr racegrp satreat sexrisk
## 1   housed 13 26  1     39          1  yes 25.112 58.41      0   black      no       4
## 2 homeless 56 62  2     43         NA <NA> 26.670 36.04      1   white      no       7
## 3   housed  0  0  3     41          0   no  6.763 74.81     13   black      no       2
## 4   housed  5  5  4     28          0   no 43.968 61.93     11   white     yes       4
## 5 homeless 10 13  5     38          1  yes 21.676 37.35     10   black      no       6
## 6   housed  4  4  6     29          0   no 55.509 46.48      5   black      no       5
##   substance treat avg_drinks max_drinks hospitalizations
## 1   cocaine   yes         13         26               3
## 2   alcohol   yes         56         62              22
## 3    heroin    no          0          0               0
## 4    heroin    no          5          5               2
## 5   cocaine    no         10         13              12
## 6   cocaine   yes          4          4               1
```

When there are a lot of variable, this format is hard to read. The `glimps()` or `inspect()` functions proved some other options.

```
glimpse(HELPrct)
```

```
## Rows: 453
## Columns: 30
## $ age              <int> 37, 37, 26, 39, 32, 47, 49, 28, 50, 39, 34, 58, 58, 60, 36, ...
## $ anysubstatus     <int> 1, 1, 1, 1, 1, 1, NA, 1, 1, 1, NA, 0, 1, 1, 1, 1, 1, 0, 0, 1...
## $ anysub           <fct> yes, yes, yes, yes, yes, yes, NA, yes, yes, yes, NA, no, yes...
## $ cesd             <int> 49, 30, 39, 15, 39, 6, 52, 32, 50, 46, 46, 49, 22, 36, 43, 3...
## $ d1               <int> 3, 22, 0, 2, 12, 1, 14, 1, 14, 4, 0, 3, 5, 10, 2, 6, 1, 2, 0...
## $ daysanysub       <int> 177, 2, 3, 189, 2, 31, NA, 47, 31, 115, NA, 192, 6, 6, 0, 27...
## $ dayslink         <int> 225, NA, 365, 343, 57, 365, 334, 365, 365, 382, 365, 365, 36...
## $ drugrisk         <int> 0, 0, 20, 0, 0, 0, 0, 7, 18, 20, 8, 0, 0, 0, 0, 0, 0, 0, 10,...
## $ e2b              <int> NA, NA, NA, 1, 1, NA, 1, 8, 7, 3, NA, NA, NA, 1, NA, 2, NA, ...
## $ female           <int> 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, ...
## $ sex              <fct> male, male, male, female, male, female, female, male, female...
## $ g1b              <fct> yes, yes, no, no, no, no, yes, yes, no, no, no, no, no, no, ...
## $ homeless         <fct> housed, homeless, housed, housed, homeless, housed, housed, ...
## $ i1               <int> 13, 56, 0, 5, 10, 4, 13, 12, 71, 20, 0, 13, 20, 13, 51, 0, 0...
## $ i2               <int> 26, 62, 0, 5, 13, 4, 20, 24, 129, 27, 0, 13, 31, 20, 51, 0, ...
## $ id               <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 1...
## $ indtot           <int> 39, 43, 41, 28, 38, 29, 38, 44, 44, 44, 34, 11, 40, 41, 38, ...
## $ linkstatus       <int> 1, NA, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,...
## $ link             <fct> yes, NA, no, no, yes, no, no, no, no, no, no, no, no, yes, n...
## $ mcs              <dbl> 25.112, 26.670, 6.763, 43.968, 21.676, 55.509, 21.793, 9.161...
## $ pcs              <dbl> 58.41, 36.04, 74.81, 61.93, 37.35, 46.48, 24.52, 65.14, 38.2...
## $ pss_fr           <int> 0, 1, 13, 11, 10, 5, 1, 4, 5, 0, 0, 13, 13, 1, 1, 7, 9, 1, 1...
## $ racegrp          <fct> black, white, black, white, black, black, black, white, whit...
## $ satreat          <fct> no, no, no, yes, no, no, yes, yes, no, yes, no, yes, yes, no...
## $ sexrisk          <int> 4, 7, 2, 4, 6, 5, 8, 6, 8, 0, 2, 0, 1, 4, 8, 3, 4, 4, 3, 7, ...
## $ substance        <fct> cocaine, alcohol, heroin, heroin, cocaine, cocaine, cocaine,...
## $ treat            <fct> yes, yes, no, no, no, yes, no, yes, no, yes, yes, no, no, ye...
```

```
## $ avg_drinks       <int> 13, 56, 0, 5, 10, 4, 13, 12, 71, 20, 0, 13, 20, 13, 51, 0, 0...
## $ max_drinks       <int> 26, 62, 0, 5, 13, 4, 20, 24, 129, 27, 0, 13, 31, 20, 51, 0, ...
## $ hospitalizations <int> 3, 22, 0, 2, 12, 1, 14, 1, 14, 4, 0, 3, 5, 10, 2, 6, 1, 2, 0...
```

```
inspect(HELPrct)
```

```
##
## categorical variables:
##        name  class levels   n missing                                   distribution
## 1    anysub factor      2 246     207 yes (77.2%), no (22.8%)
## 2       sex factor      2 453       0 male (76.4%), female (23.6%)
## 3       g1b factor      2 453       0 no (72%), yes (28%)
## 4  homeless factor      2 453       0 housed (53.9%), homeless (46.1%)
## 5      link factor      2 431      22 no (62.2%), yes (37.8%)
## 6   racegrp factor      4 453       0 black (46.6%), white (36.6%) ...
## 7   satreat factor      2 453       0 no (71.5%), yes (28.5%)
## 8 substance factor      3 453       0 alcohol (39.1%), cocaine (33.6%) ...
## 9     treat factor      2 453       0 no (50.3%), yes (49.7%)
##
## quantitative variables:
##                   name   class    min     Q1 median     Q3    max     mean       sd   n
## ...1               age integer 19.000  30.00  35.00  40.00  60.00  35.6534   7.7103 453
## ...2      anysubstatus integer  0.000   1.00   1.00   1.00   1.00   0.7724   0.4202 246
## ...3              cesd integer  1.000  25.00  34.00  41.00  60.00  32.8477  12.5145 453
## ...4                d1 integer  0.000   1.00   2.00   3.00 100.00   3.0596   6.1876 453
## ...5        daysanysub integer  0.000   5.00  33.00 164.25 268.00  75.3074  79.2374 244
## ...6          dayslink integer  2.000  74.00 361.00 365.00 456.00 255.6056 151.0227 431
## ...7          drugrisk integer  0.000   0.00   0.00   1.00  21.00   1.8872   4.3365 452
## ...8               e2b integer  1.000   1.00   2.00   3.00  21.00   2.5047   2.5245 214
## ...9            female integer  0.000   0.00   0.00   0.00   1.00   0.2362   0.4252 453
## ...10               i1 integer  0.000   3.00  13.00  26.00 142.00  17.9073  20.0202 453
## ...11               i2 integer  0.000   4.00  18.00  33.00 184.00  24.5475  28.0202 453
## ...12               id integer  1.000 119.00 233.00 348.00 470.00 233.4018 134.7467 453
## ...13           indtot integer  4.000  32.00  38.00  41.00  45.00  35.7285   7.1522 453
## ...14       linkstatus integer  0.000   0.00   0.00   1.00   1.00   0.3782   0.4855 431
## ...15              mcs numeric  6.763  21.68  28.60  40.94  62.18  31.6767  12.8393 453
## ...16              pcs numeric 14.074  40.38  48.88  56.95  74.81  48.0485  10.7846 453
## ...17           pss_fr integer  0.000   3.00   7.00  10.00  14.00   6.7064   3.9950 453
## ...18          sexrisk integer  0.000   3.00   4.00   6.00  14.00   4.6424   2.8002 453
## ...19       avg_drinks integer  0.000   3.00  13.00  26.00 142.00  17.9073  20.0202 453
## ...20       max_drinks integer  0.000   4.00  18.00  33.00 184.00  24.5475  28.0202 453
## ...21 hospitalizations integer  0.000   1.00   2.00   3.00 100.00   3.0596   6.1876 453
##       missing
## ...1        0
## ...2      207
## ...3        0
## ...4        0
## ...5      209
## ...6       22
## ...7        1
## ...8      239
## ...9        0
## ...10       0
## ...11       0
```

```
## ...12       0
## ...13       0
## ...14      22
## ...15       0
## ...16       0
## ...17       0
## ...18       0
## ...19       0
## ...20       0
## ...21       0
```

From this we see that there are 453 observational units in this data set and 30 variables. That's plenty of variables to get us started with exploration of data.

1.2.4 The KidsFeet data set

Here is another data set in the `mosaic` package:

```
head(KidsFeet)
```

```
##     name birthmonth birthyear length width sex biggerfoot domhand
## 1  David          5        88   24.4   8.4   B          L       R
## 2   Lars         10        87   25.4   8.8   B          L       L
## 3   Zach         12        87   24.5   9.7   B          R       R
## 4   Josh          1        88   25.2   9.8   B          L       R
## 5   Lang          2        88   25.1   8.9   B          L       R
## 6 Scotty          3        88   25.7   9.7   B          R       R
```

1.2.5 The oldfaith data set

A final example data set comes from the `alr4` package. This package is probably not loaded (unless you already loaded it). You can load it from the Packages tab or by typing the command

```
library(alr4)      # require(alr4) will also work
```

Once you have done that, you will have access to the data set containing information about Old Faithful eruptions.

```
head(oldfaith)
```

```
##   Duration Interval
## 1      216       79
## 2      108       54
## 3      200       74
## 4      137       62
## 5      272       85
## 6      173       55
```

If you want to know the size of your data set, you can ask it how many rows and columns it has with `nrow()`, `ncol()`, or `dim()`:

```
nrow(oldfaith)

## [1] 270

ncol(oldfaith)

## [1] 2

dim(oldfaith)

## [1] 270   2
```

In this case we have 270 observations of each of two variables. In a data frame, the observational units are always in the rows and the variables are always in the columns. If you create data for use in R (or most other statistical packages), you need to make sure your data are also in this shape.

1.2.6 Using your own data

In the Environment tab you will "Import Dataset". Click on this import data from a CSV file, Excel spreadsheet, or a few other formats. When you do this, the R code will be displayed, so you can see how it is done in R code.

If you are using the RStudio server, you will first need to upload your file to the server (unless you can access the file via URL). To do this, choose "Upload" from the Files tab.

1.3 Graphical and Numerical Summaries of Data

1.3.1 The Most Important Template

Using the `mosiac` and `ggformula` packages, we can compute a wide variety of graphical and numerical summaries using the following general template:

`goal ( y ~ x , data = mydata )`

We will see this same template used again for linear and non-linear modeling as well, so it is is important to master it.[1]

- `goal`: The name of the function generally describes your goal, the thing you want the computer to produce for you. In the case of plotting, it is the name of the plot. When we do numerical summaries it will be the name of the numerical summary (mean, median, etc.).
- `y`: For plots, this is the variable that goes on the y-axis.
- `x`: For plots, this is the variable that goes on the x-axis.

[1] This is textbook speak for "you should really take note of this – probably memorize it."

- `formula`: Together, `y ~ x` is called a **formula**. Very often we can think of `y ~ x` as "y depends on x". We will see that sometimes we can omit `y` ore replace `x` with `.` (there must always be something on the right-hand side). We will even see things like `y ~ x | z`. But the most important formula to learn is `y ~ x`.
- `mydata`: A data frame must be given in which the variables mentioned in the formula can be found. Variables not found there will be looked for in the enclosing environment. Sometimes we will take advantage of this to avoid creating a temporary data frame just to make a quick plot, but generally it is best to have all the information inside a data frame.

1.4 Scatterplots

The most common way to look at two quantitative variables is with a scatter plot. The ggformula function for this is `gf_point()`, and the basic syntax is

```
gf_point( yvar ~ xvar, data = dataName)
```

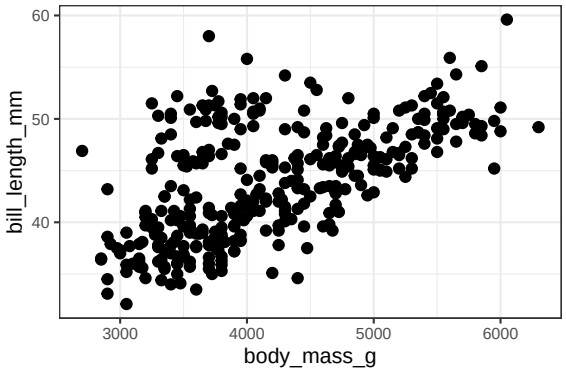
Let's look at an example. Let's see how bill length is realted to body mass in some penguins.

```
library(palmerpenguins)
head(penguins)
```

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex    year
##   <fct>   <fct>              <dbl>         <dbl>             <int>       <int> <fct> <int>
## 1 Adelie  Torgersen           39.1          18.7               181        3750 male   2007
## 2 Adelie  Torgersen           39.5          17.4               186        3800 fema~  2007
## 3 Adelie  Torgersen           40.3          18                 195        3250 fema~  2007
## 4 Adelie  Torgersen           NA            NA                  NA          NA <NA>   2007
## 5 Adelie  Torgersen           36.7          19.3               193        3450 fema~  2007
## 6 Adelie  Torgersen           39.3          20.6               190        3650 male   2007
```

```
gf_point(bill_length_mm ~ body_mass_g, data = penguins)
```

```
## Warning:  Removed 2 rows containing missing values (geom_point).
```



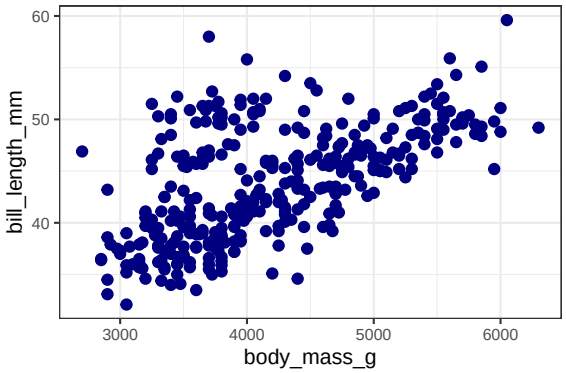
That's all there is to it. We can replace `bill_length_mm`, `body_mass_g`, and `penguins` with any variables and data set we like to get the scatter plot we want.

1.4.1 Adding Color

Let's add some color. Consider the next two examples.

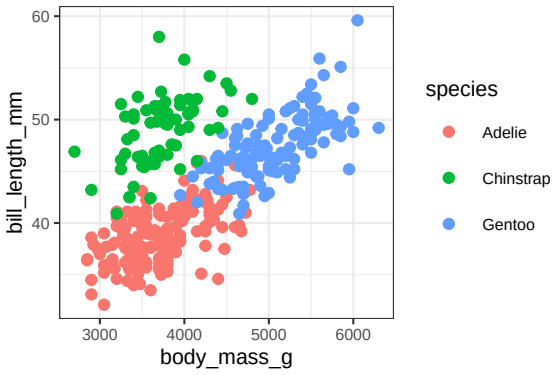
```
gf_point(bill_length_mm ~ body_mass_g, color = "navy", data = penguins)

## Warning:  Removed 2 rows containing missing values (geom_point).
```




```
gf_point(bill_length_mm ~ body_mass_g, color = ~ species, data = penguins)

## Warning:  Removed 2 rows containing missing values (geom_point).
```



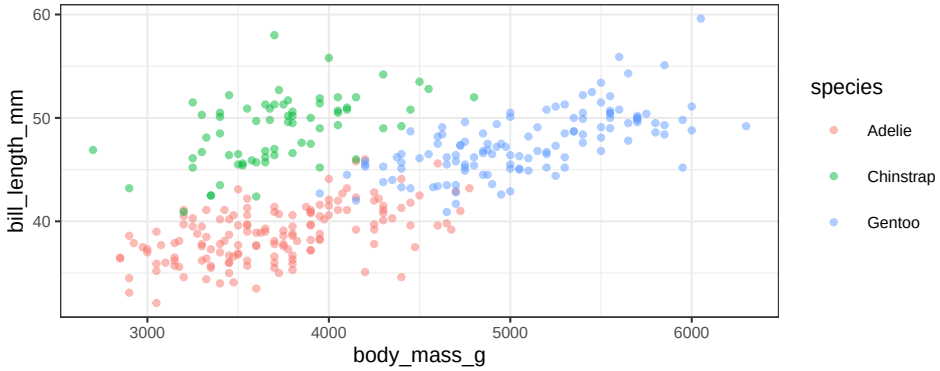
- In the first we are **setting** the color of the dots to be navy.
- In the second, we are **mapping** color based on `species`. Think of `color = species` as "color depends on species".

1.4.2 Transparency and dot size

With so much data in so little space, overplotting (dots on top of each other) can make it hard to see what is going on. We can improve this plot by making the dots smaller and semi-transparent.

```
gf_point(bill_length_mm ~ body_mass_g, color = ~ species, data = penguins,
         size = 0.8, alpha = 0.5)

## Warning:  Removed 2 rows containing missing values (geom_point).
```



There are many other options we can use to refine our plots. We'll learn about some of them as we go along. You can use R's built-in help to find out more. Our you can type

```
gf_point()
```

```
## gf_point() uses
##    * a formula with shape y ~ x.
##    * geom:  point
##    * key attributes:  alpha, color, size, shape, fill, group, stroke
##
## For more information, try ?gf_point
```

1.4.3 Conditional plots (aka Faceting)

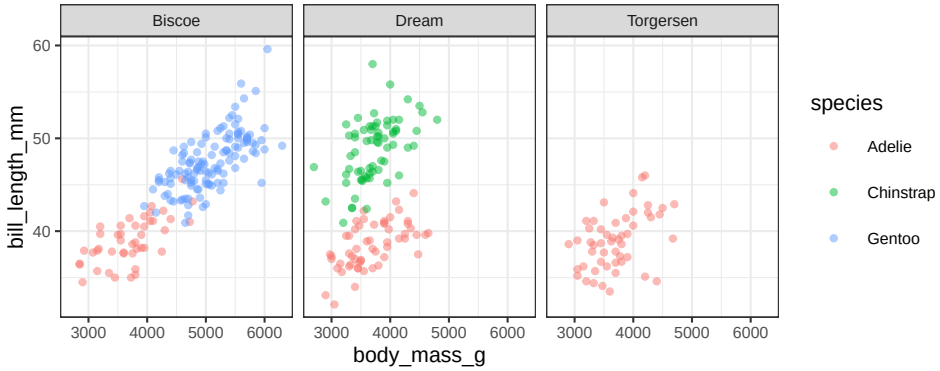
The formula for a ggformula plot can be extended to create multiple panels, called facets, based on a "condition", often given by another variable. The general syntax for this becomes

```
plotname( y ~ x | condition, data = dataName )
```

You can read the formula `y ~ x | condition` as saying that we want to know how y depends on x separately for each condition. In our penguins example, we might divide up the data according to the islands on which the penguins were spotted.

```
gf_point(bill_length_mm ~ body_mass_g | island, color = ~ species, data = penguins,
         size = 0.8, alpha = 0.5)
```

```
## Warning:  Removed 2 rows containing missing values (geom_point).
```



1.4.4 Other types of plots

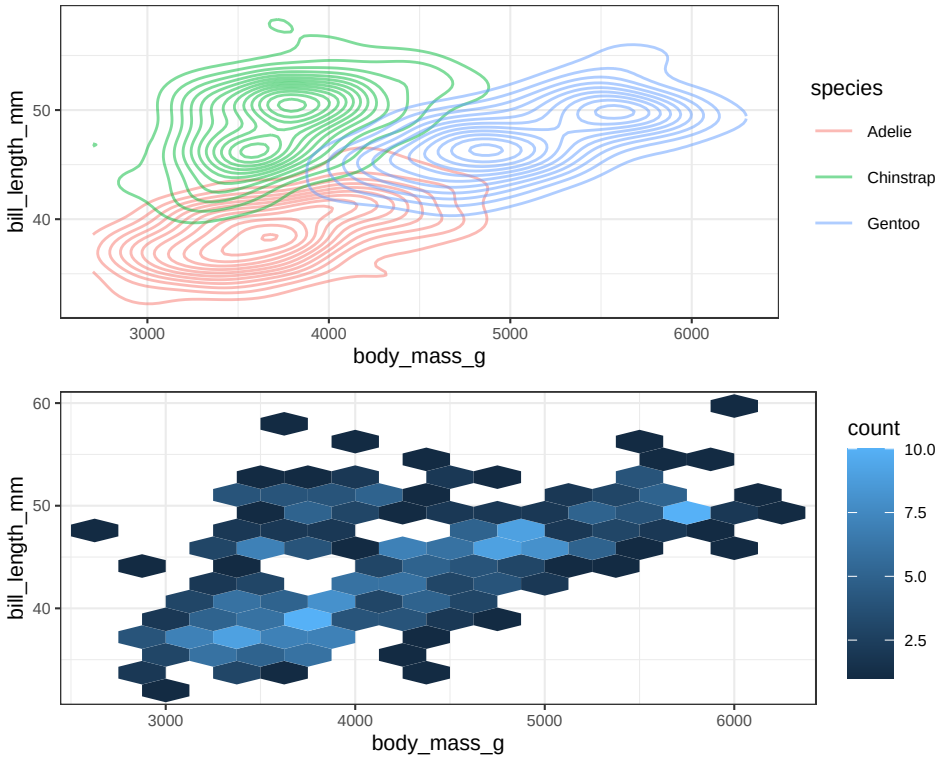
A scatter plot will be our most common plot for two quantitative variables, but we can use the same template for any otehr type of plot. Here are two examples.

```
gf_density2d(bill_length_mm ~ body_mass_g, color = ~ species, data = penguins, alpha = 0.5)

## Warning:  Removed 2 rows containing non-finite values (stat_density2d).

gf_hex(bill_length_mm ~ body_mass_g, data = penguins, binwidth = c(250, 2))

## Warning:  Removed 2 rows containing non-finite values (stat_binhex).
```



1.5 Graphing the Distribution of One Variable

A **distribution** is described by telling what values occur and with what frequency. That is, the distribution answers two questions:

- What values?
- How often?

Statisticians have devised a number of graphs to help us see distributions of a variable visually. In these graphs, R can compute the y-variable for us. In this case, we simply omit the `y` part of the formula, so the general syntax for making a graph or numerical summary of one variable in a data frame is

```
plotname( ~ variable, data = dataName )
```

In other words, there are three pieces of information we must provide to R in order to get the plot we want:

- The kind of plot. (`gf_histogram()`, `gf_bar()`, `gf_density()`, `gf_boxplot()`, etc.)
- The name of the variable
- The name of the data frame this variable is a part of.

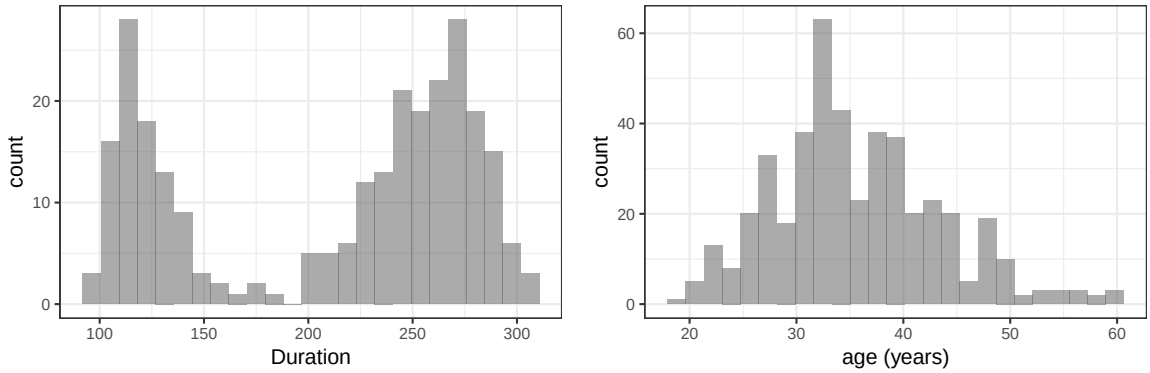
Note: The same syntax works for numerical summaries as well – thanks to the `mosaic` package we can apply the same syntax for `mean()`, `median()`, `sd()`, `var()`, `max()`, `min()`, etc. Later we will use this syntax again to compute linear and nonlinear models.

1.5.1 Histograms (and density plots) for quantitative variables

Histograms (and density plots) are the two most common ways of displaying the distribution of a quantitative variable.

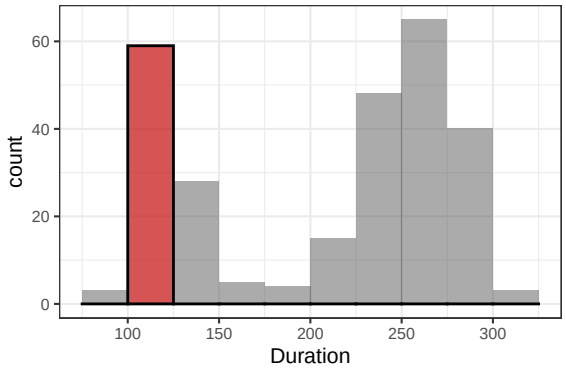
Here are a couple examples:

```
gf_histogram( ~ Duration, data = oldfaith)
gf_histogram( ~ age, data = HELPrct)
```



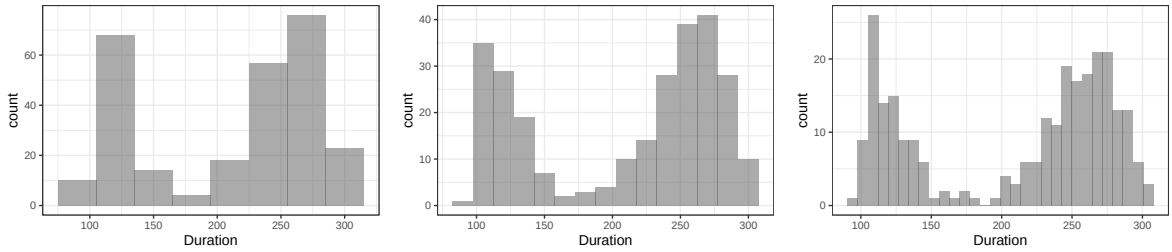
In each of these plots the height of the bar indicates how many observations fall within the range indicated by the bottom of the bar. So in the histogram below, the red bar indicates that there are almost 60 eruptions of duration between 100 and 125 seconds in this data set.

```
gf_histogram( ~ Duration, data = oldfaith, binwidth = 25, boundary = 100) %>%
  gf_histogram( ~ Duration, data = oldfaith %>% filter(Duration >100, Duration <= 125),
                fill = "red", color = "black",
                binwidth = 25, boundary = 100)
```



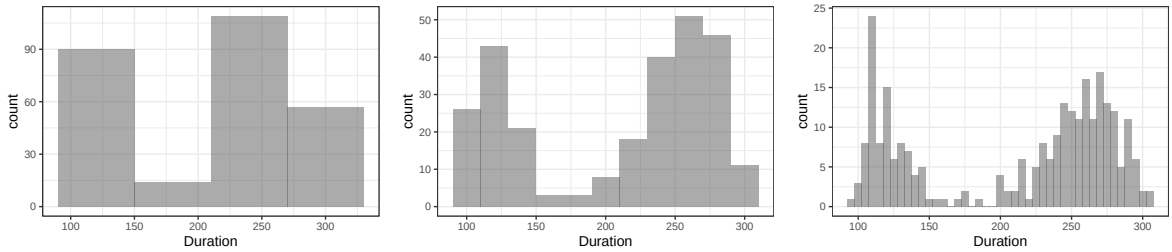
We can control the (approximate) number of bins using the `bins`. The number of bins (and to a lesser extent the positions of the bins) can make a histogram look quite different.

```
gf_histogram( ~ Duration, data = oldfaith, bins = 8 )
gf_histogram( ~ Duration, data = oldfaith, bins = 15 )
gf_histogram( ~ Duration, data = oldfaith, bins = 30 )
```



We can use `binwidth` to set the width of the bins.

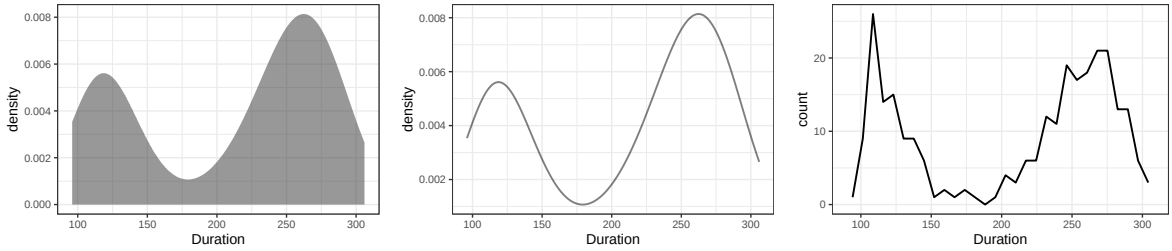
```
gf_histogram( ~ Duration, data = oldfaith, binwidth = 60 )
gf_histogram( ~ Duration, data = oldfaith, binwidth = 20 )
gf_histogram( ~ Duration, data = oldfaith, binwidth = 5 )
```



R also provides a "smooth" version called a density plot and a triangular version called a frequency polygon:;

```
gf_density( ~ Duration, data = oldfaith )
gf_dens( ~ Duration, data = oldfaith )
gf_freqpoly( ~ Duration, data = oldfaith )
```

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



1.5.2 Describing the shape of a distribution

If we make a histogram of our data, we can describe the overall shape of the distribution. Keep in mind that the shape of a particular histogram may depend on the choice of bins. Choosing too many or too few bins can hide the true shape of the distribution. (When in doubt, make more than one histogram.)

Here are some words we use to describe shapes of distributions.

symmetric The left and right sides are mirror images of each other.

skewed The distribution stretches out farther in one direction than in the other. (We say the distribution is skewed toward the long tail.)

uniform The heights of all the bars are (roughly) the same. (So the data are equally likely to be anywhere within some range.)

unimodal There is one major "bump" where there is a lot of data.

bimodal There are two "bumps".

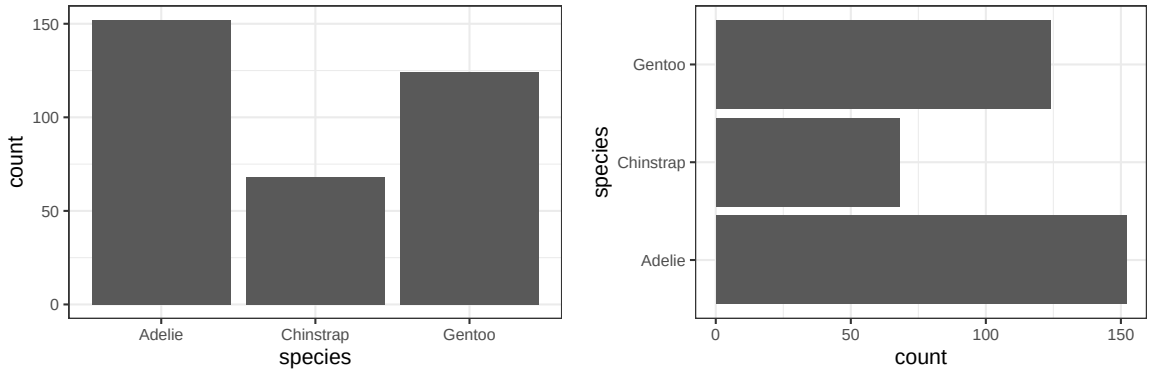
outlier An observation that does not fit the overall pattern of the rest of the data.

We'll learn about another graph used for quantitative variables (boxplots) soon.

1.5.3 Bar graphs for categorical variables

Bar graphs are a way of displaying the distribution of a categorical variable.

```
gf_bar( ~ species, data = penguins)    # vertical bars
gf_bar(species ~ ., data = penguins)   # horizontal bars
```



Statisticians rarely use pie charts because they are harder to read except in a few special cases (like comparing a proportion to 50%).

1.5.4 Overlaying and faceting data

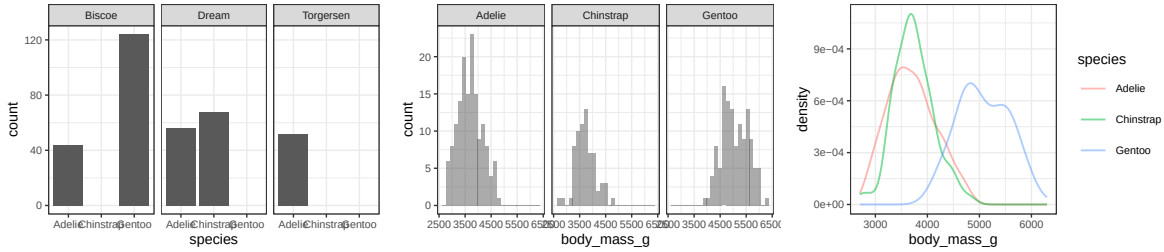
Overlaying and faceting work the same for these one variables plots as they did for the scatterplots above.

```
gf_bar( ~ species | island, data = penguins)
gf_histogram( ~ body_mass_g | species, data = penguins)

## Warning:  Removed 2 rows containing non-finite values (stat_bin).

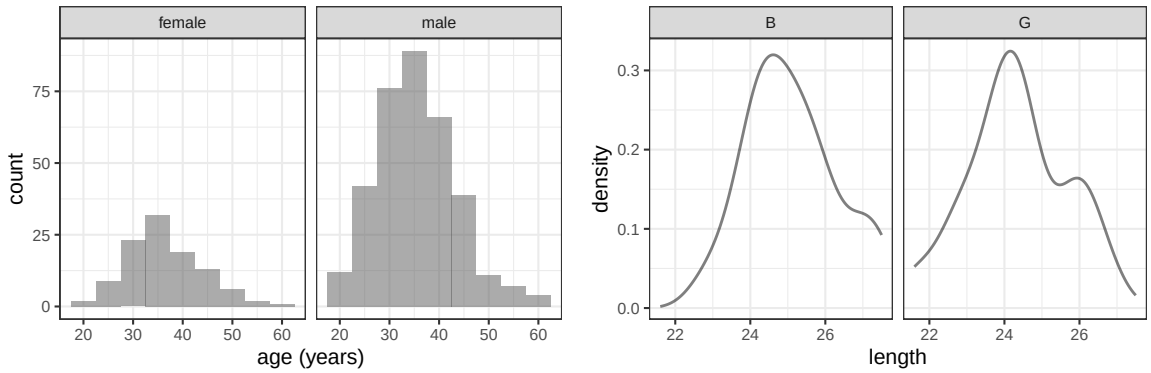
gf_dens( ~ body_mass_g, color = ~ species, data = penguins)

## Warning:  Removed 2 rows containing non-finite values (stat_density).
```



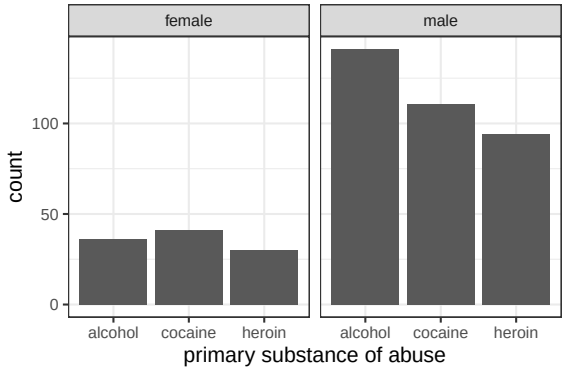
For example, we might like to see how the ages of men and women compare in the HELP study, or whether the distribution of weights of male mosquitoes is different from the distribution for females.

```
gf_histogram( ~ age | sex, data = HELPrct, binwidth = 5)
gf_dens( ~ length | sex, data = KidsFeet )
```



We can do the same thing for bar graphs.

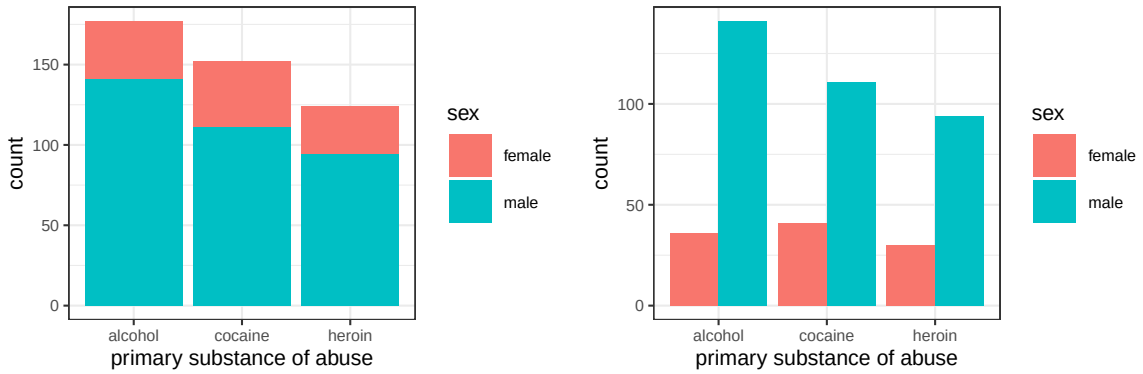
```
gf_bar( ~ substance | sex, data = HELPrct)
```



1.5.5 Grouping and bar charts

When dividing bar charts into multiple colors, we can present the segmented bars "stacked" (the default) or "dodged":

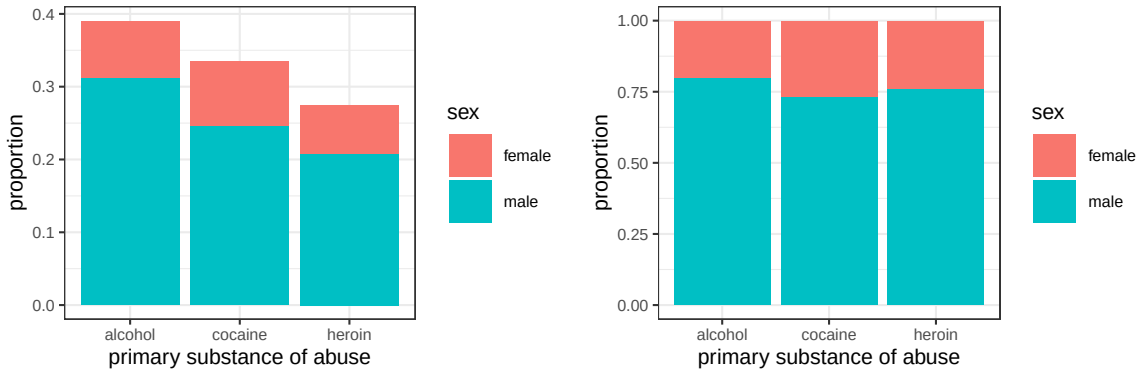
```
gf_bar(~substance, fill = ~sex, data = HELPrct)
gf_bar(~substance, fill = ~sex, data = HELPrct, position = "dodge")
```



1.5.6 Proportions and bar charts

Sometimes it is better to display bars with proportions rather than counts. But then we must decide what to use for the denominator. In the first example below, the total of all teh bars adds to 1. In the second plot, the total adds to one for each x variable, which makes it easier to see how the proportions of male and female .

```
gf_props(~substance, fill = ~sex, data = HELPrct)
gf_props(~substance, fill = ~sex, data = HELPrct, denom = ~ x)
```

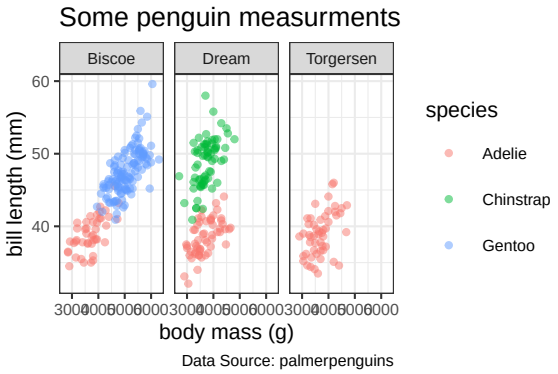


1.6 Labeling plots

Often the defaults labels are not ideal for publication purposes. You can add titles and captions and change the labeling of variables using `gf_labs()`.

```
gf_point(bill_length_mm ~ body_mass_g | island, color = ~ species, data = penguins,
         size = 0.8, alpha = 0.5) %>%
  gf_labs(x = "body mass (g)", y = "bill length (mm)",
          title = "Some penguin measurments", caption = "Data Source: palmerpenguins")

## Warning:  Removed 2 rows containing missing values (geom_point).
```



Notice the %>% in the example above. This important and connects the labeling below to the plot above.

1.7 Exporting Plots

You can save plots to files or copy them to the clipboard using the Export menu in the Plots tab. It is quite simple to copy the plots to the clipboard and then paste them into a Word document, for example. You can even adjust the height and width of the plot first to get it the shape you want. *But there are much better ways to produce documents with* R *graphics in them!* See the next section.

1.8 Reproducible Research

Copy-and-paste is a bad workflow for lots of reasons, including:

- It is tedious, unless there is very little to copy and paste.
- It is error-prone – it's easy to copy to little or too much, or to grab the wrong thing, or to copy when you want to cut or cut when you want to copy.
- If something changes, you have to start all over.
- You have no record of what you did (unless you are an unusual person who takes detailed notes about all the copying and pasting).

So while copy-and-paste seems easy and convenient, it is not *reproducible*. Reproducibility is important when projects are large, when it is important to have record of exactly what was done, or when the same analysis is applied to multiple data sets (or a data set that is growing over time).

RStudio makes it easy to use techniques of reproducible research to create documents that include text, R commands, R output, and R graphics.

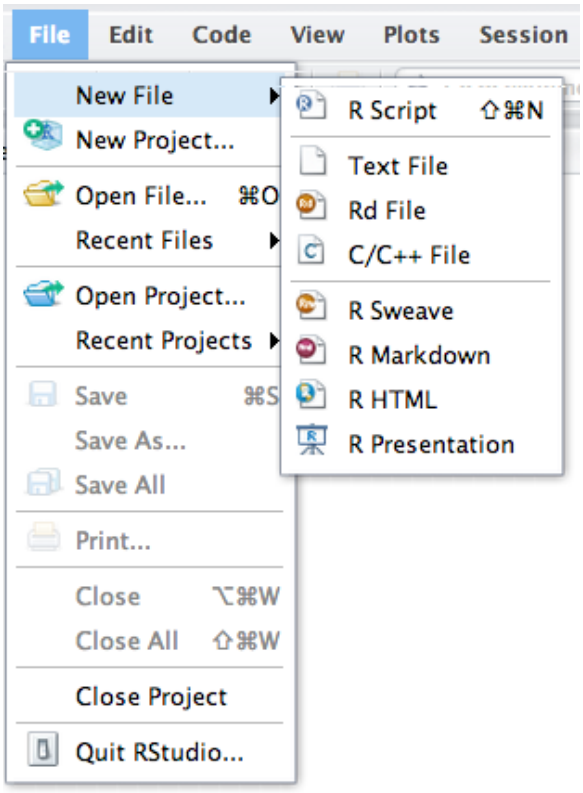
1.8.1 R Markdown

The simplest version of this uses a format called R Markdown. Markdown is a simple mark up language that allows for a few basic improvements on plain text (section headers, bulleted lists, numbered lists, bold, italics, etc.) R Markdown adds the ability to mix in the R stuff. The end product is an HTML file, so it is especially good for producing web documents.[2]

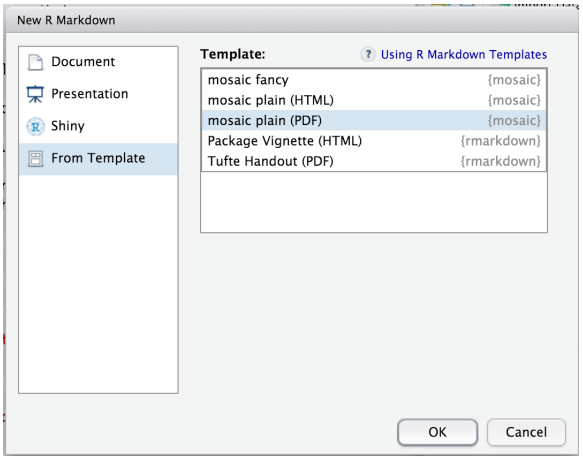
Creating a new document

To create a new R Markdown document, go to "File", "New File", then "R Markdown":

[2] You can actually mix in arbirary HTML and even css, so if you are good at HTML, you can have quite a bit of control over how things look. Here will will focus on the basics.

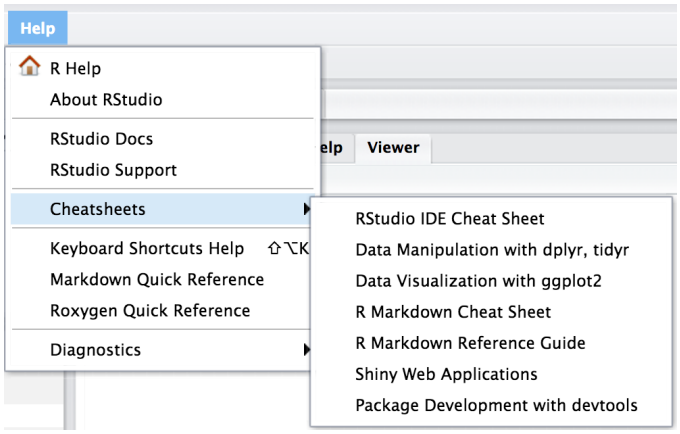


From there, choose either "document" or "from template". The `mosaic` plain template is a good starting point.

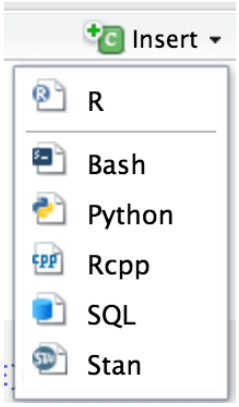


When you do this, a file editing pane will open with a template inserted. If you click on "Knit HTML", RStudio will turn this into an HTML file and display it for you. Give it a try. You will be asked to name your file if you haven't already done so. If you are using the RStudio server in a browser, then your file will live on the server ("in the cloud") rather than on your computer.

If you look at the template file you will see that the file has two kinds of sections. Some of this file is just normal text (with some extra symbols to make things bold, add in headings, etc.) You can get a list of all of these mark up options by selecting the "Markdown Quick Reference" in the help menu.



The second type of section is an R code chunk. These are colored differently to make them easier to see. You can insert a new code chunk by selecting by selecting the appropriate chunk type (R in our case) from the menu below.



You can also type

```
```{r}
```
```

to begin and end the code chunk if you would rather type. You can put any R code in these code chunks and the results (text output or graphics) as well as the R code will be displayed in your HTML file.

R Markdown files must be self-contained

R Markdown files do not have access to things you have done in your console. (This is good, else your document would change based on things not in the file.) This means that you must explicitly load any data packages that you use *in the R Markdown file*. In this class, this means that most of your R Markdown files will have a chunk near the beginning that includes

```
library(mosaic)
```

If you use one of the RMarkdown templates provided, this (and some otehr things) will be included in the template and save you some time.

1.8.2 Chunk options

R Markdown provides a number of chunk options that control how R code is processed. You can use them to do things like:

- run the code without displaying it (good for polished reports – your client doesn't want to see the code)
- show the code without running it – mainly useful for demonstration purposes
- control the size and alignment of graphics

You can set default values for the chunk options and you can also override them in individual chunks. See the R Markdown help for more information about chunk options.

The default plots are often bigger than required. The following chunk options are a place to start. They can be adjusted as necessary.

```
library(knitr)
opts_chunk$set(fig.width = 5, fig.height = 2, fig.align = "center", fig.show = "hold")
```

Some of these document settings can also be set in the YAML header (the top few lines of the RMarkdown file).

1.8.3 knitr/latex

There is another system that produces PDFs by combining LaTeX and R. This is the system used to create this document and it gives much more control over paper-like formatting. The quality is good enough for profession publishing. If you already knwow LaTeX, it is very easy to learn. If you don't know LaTeX, then you need to learn the basics of LaTeX to get going, but it isn't very difficult.

R code and output can be copied and pasted as well. It's best to use a fixed width font (like Courier) for R code so that things align properly.

Note: RStudio provide some nice utilities for creating documents that include text, code, graphics, and statistical analyses all in one document. That's how this document was produced. The simpler of these is called RMarkdown. The resulting file will be an HTML file with embedded plots. You can gain more control over the output by using `knitr` which provides a way to combine LaTeX and Rin a single document. The resulting document in this case is a high quality PDF

1.9 Getting Help in RStudio

1.9.1 The RStudio help system

There are several ways to get RStudio to help you when you forget something. Most objects in packages have help files that you can access by typing something like:

```
?bargraph
?histogram
?HELPrct
```

You can search the help system using

```
help.search('Grand Rapids')    # Does R know anything about Grand Rapids?
```

This can be useful if you don't know the name of the function or data set you are looking for.

1.9.2 Tab completion

As you type the name of a function in RStudio, you can hit the tab key and it will show you a list of all the ways you could complete that name, and after you type the opening parenthesis, if you hit the tab key, you will get a list of all the arguments and (sometimes) some helpful hints about what they are.)

1.9.3 History

If you know you have done something before, but can't remember how, you can search your history. The history tab shows a list of recently executed commands. There is also a search bar to help you find things from longer ago.

1.9.4 Error messages

When things go wrong, R tries to help you out by providing an error message. If you can't make sense of the message, you can try copying and pasting your command and the error message and sending to me in an email. One common error message is illustrated below.

```
fred <- 23
frd


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

The object `frd` is not found because it was mistyped. It should have been `fred`. If you see an "object not found" message, check your typing and check to make sure that the necessary packages have been loaded.

1.10 Graphical Summaries – Important Ideas

1.10.1 The Most Important Template

The plots we have created have all followed a single template

`goal` (`formula` , data = `mydata` , ...)

We will see this same template used again for numerical summaries and linear and non-linear modeling as well, so it is is important to master it.

- `goal`: The name of the function generally describes your goal, the thing you want the computer to produce for you. In the case of plotting, it is the name of the plot. When we do numerical summaries it will be the name of the numerical summary (mean, median, etc.).

- `formula`: For plotting, the formula describes which variables are used on the x-axis, the y-axis and for conditioning. The general scheme is

  ```
  y ~ x | z
  ```

  where `z` is the conditioning variable. Sometimes `y` or `z` are missing (but the right-hand side `x` must always be included in a formula).

- `mydata:` A data frame must be given in which the variables mentioned in the formula can be found. Variables not found there will be looked for in the enclosing environment. Sometimes we will take advantage of this to avoid creating a temporary data frame just to make a quick plot, but generally it is best to have all the information inside a data frame.

- `...` There are many optional arguments to control sizes, colors, etc. We will introduce these as they are needed. Consult the help files for assistance.

Just fill in the boxes and get your plot.

1.10.2 Patterns and Deviations from Patterns

The goal of a statistical plot is to help us see

- potential patterns in the data, and
- deviations from those patterns.

1.10.3 Different Plots for Different Kinds of Variables

Graphical summaries can help us see the *distribution* of a variable or the *relationships* between two (or more) variables. The type of plot used will depend on the kinds of variables involved. There is a nice summary of these on page 48. You can use `demo()` to see how to get R to make the plots in this section.

Later, when we do statistical analysis, we will see that the analysis we use will also depend on the kinds of variables involved, so this is an important idea.

1.10.4 Side-by-side Plots and Overlays Can Reveal Importance of Additional Factors

The `ggformula` graphics plots make it particularly easy to generate plats that divide the data into groups and either produce a panel for each group (using `|`) or display each group in a different way (different colors or symbols, using the `groups` argument). These plots can reveal the possible influence of additional variables – sometimes called covariates.

1.10.5 Area = (relative) frequency

Many plots are based on the key idea that our eyes are good at comparing areas. Plots that use area (e.g., histograms, mosaic plots, bar charts, pie charts) should always obey this principle

$$\text{Area} = \text{(relative) frequency}$$

Plots that violate this principle can be deceptive and distort the true nature of the data.

1.11 Exercises

The solutions to these exercises should be done using an RMarkdown document and knitTHML in RStudio. Include both the plots and the code you used to make them as well as any required discussion. Once you get the plots figured out, feel free to use some of the bells and whistles to make the plots even better.

1.1 Create a scatterplot using the two variables in the `oldfaith` data frame (available in the `alr4` package). What do we learn about Old Faithful eruptions from this plot?

1.2 Where do the data in the `CPS85` data frame (in the `mosaic` package) come from? What are the observational units? How many are there?

1.3 Choose a quantitative variable that interests you in the `CPS85` data set. Make an appropriate plot and comment on what you see.

1.4 Choose a categorical variable that interests you in the `CPS85` data set. Make an appropriate plot and comment on what you see.

1.5 Create a plot that displays two or more variables from the `CPS85` data. At least one should be quantitative and at least one should be categorical. Comment on what you can learn from your plot.

1.6 Where do the data in the `mpg` data frame (in the `ggplot2` package) come from? What are the observational units? How many are there?

1.7 Choose a quantitative variable that interests you in the `mpg` data set. Make an appropriate plot and comment on what you see.

1.8 Choose a categorical variable that interests you in the `mpg` data set. Make an appropriate plot and comment on what you see.

1.9 Create a plot that displays two or more variables from the `mpg` data. At least one should be quantitative and at least one should be categorical. Comment on what you can learn from your plot.

1.10 The file at `http://www.calvin.edu/~rpruim/data/Fires.csv` is a csv file containing data on wild lands fires in the US over a number of years. You can load this data one of two ways.

- Go to the workspace tab, select Import Data Set, choose From Web URL... and follow the instructions.
- Use the following command in R:

```
# equivalent to "From Text (base)" in RStudio
Fires <- read.csv("https://rpruim.github.io/Engineering-Statistics/data/Fires.csv")

# alternative method -- equivalent to "From Text (readr)" in RStudio
library(readr)
Fires <- read_csv("https://rpruim.github.io/Engineering-Statistics/data/Fires.csv")

##
## - Column specification --------------------------------
## cols(
##   Year = col_double(),
##   Fires = col_double(),
##   Acres = col_double()
## )
```

You can also use either of these methods to read from a file rather than from a web URL, so this is a good way to get your own data into R.

a) The source for these data claim that data before a certain year should not be compared to data from after that year because the older data were computed a different way and are not considered as reliable. What year is the break point? Use graphs of the data over time to estimate when something changed.

b) You can trim the data to just the subset you want using `subset()`. For example, to get just the subset of years since 1966, you would use

```
Fires2 <- subset(Fires, Year > 1966)
```

Be sure to use a new name if you want to keep the original data available.

Do this to create a data set that contains only the data from the new data regime (based on your answer in the previous problem).

c) Using only the data from this smaller set, how would describe what is happening with fires over time?

1.11 Use R's help system to find out what the `i1` and `i2` variables are in the `HELPrct` data frame. Make histograms for each variable and comment on what you find out. How would you describe the shape of these distributions? Do you see any outliers (observations that don't seem to fit the pattern of the rest of the data)?

1.12 Compare the distributions of `i1` and `i2` among men and women.

1.13 Compare the distributions of `i1` and `i2` among the three `substance` groups.

1.14 The `SnowGR` contains historical data on snowfall in Grand Rapids, MI. The snowfall total for January, 2014 was 36.6 inches.

a) Create a histogram of January snowfall totals. How unusual is 36.6 inches of snow in January?

b) If there is a lot of snow in January, should we expect to have unusually much or little snow in February? Make a scatter plot comparing January and February snowfall totals and comment on what you see there.

2 Numerical Summaries

2.1 Tabulating Data

A table is one kind of numerical summary of a data set. In fact, you can think of histograms and bar graphs as graphical representations of summary tables. But sometimes it is nice to have the table itself. R provides several ways of obtaining such tables.

2.1.1 Tabulating a categorical variable

The formula interface

There are several functions for tabulating categorical variables. `tally()` uses a syntax that is very similar to `bargraph()`. We'll call this method the **formula interface**. (R calls anything with a tilde (`~`) a formula.)

```
tally( ~ sex, data = KidsFeet )
```

```
## sex
##  B  G
## 20 19
```

```
tally( ~ sex, data = KidsFeet, format = "prop" )
```

```
## sex
##      B      G
## 0.5128 0.4872
```

```
tally( ~ sex, data = KidsFeet, format = "perc" )
```

```
## sex
##     B     G
## 51.28 48.72
```

The $-interface

`table()` and its cousins use the `$` operator which selects one variable out of a data frame.

```
KidsFeet$sex      # general syntax: dataframe$variable

##  [1] B B B B B B B G G B B B B B G G G G G G B B G G G B G B B B G G G B B G G G G
## Levels: B G

table(KidsFeet$sex)

##
##  B  G
## 20 19

# tally() is ambidextrous:
tally(KidsFeet$sex)

## X
##  B  G
## 20 19
```

We'll call this interface the `$`-interface.

Two interfaces

Some functions in R require the formula interface, some require the `$`-interface, and some allow you to use either one.[1]

My advice is to use formula interfaces whenever they are available and to choose tools that make this possible.

2.1.2 Tabulating a quantitative variable

Although `tally()` and `table()` work with quantitative variables as well as categorical variables, this is only useful when there are not too many different values for the variable.

```
tally( ~age, data = HELPrct )

## age
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
##  1  2  3  8  5  8  7 13 18 15 18 18 20 28 35 18 25 23 20 18 27 10 20 10 13  7 13  5 14  5
## 49 50 51 52 53 54 55 56 57 58 59 60
##  8  2  1  1  3  1  2  1  2  2  2  1
```

[1] One of the things that the `mosaic` package does is provide a formula interface for many functions that only had a `$`-interface before.

Tabulating in bins (optional)

Usually a graph is the best way to display and summarize quantitative data, but if you need to creat a summary table, you may need to group quantitative data into bins. We just have to tell R what the bins are. For example, suppose we wanted to group the 20s, 30s, 40s, etc. together.

```
# let's add a new variable to HELPrct
HELPrct <- HELPrct %>%
  mutate(binnedAge = cut(age, breaks=c(10,20,30,40,50,60,70) ))
head(HELPrct)
```

```
##   age anysubstatus anysub cesd d1 daysanysub dayslink drugrisk e2b female    sex g1b
## 1  37            1    yes   49  3        177      225        0  NA      0   male yes
## 2  37            1    yes   30 22          2       NA        0  NA      0   male yes
## 3  26            1    yes   39  0          3      365       20  NA      0   male  no
## 4  39            1    yes   15  2        189      343        0   1      1 female  no
## 5  32            1    yes   39 12          2       57        0   1      0   male  no
## 6  47            1    yes    6  1         31      365        0  NA      1 female  no
##   homeless i1 i2 id indtot linkstatus link    mcs   pcs pss_fr racegrp satreat sexrisk
## 1   housed 13 26  1     39          1  yes 25.112 58.41      0   black      no       4
## 2 homeless 56 62  2     43         NA <NA> 26.670 36.04      1   white      no       7
## 3   housed  0  0  3     41          0   no  6.763 74.81     13   black      no       2
## 4   housed  5  5  4     28          0   no 43.968 61.93     11   white     yes       4
## 5 homeless 10 13  5     38          1  yes 21.676 37.35     10   black      no       6
## 6   housed  4  4  6     29          0   no 55.509 46.48      5   black      no       5
##   substance treat avg_drinks max_drinks hospitalizations binnedAge
## 1   cocaine   yes         13         26                3   (30,40]
## 2   alcohol   yes         56         62               22   (30,40]
## 3    heroin    no          0          0                0   (20,30]
## 4    heroin    no          5          5                2   (30,40]
## 5   cocaine    no         10         13               12   (30,40]
## 6   cocaine   yes          4          4                1   (40,50]
```

```
tally( ~ binnedAge, data = HELPrct)
```

```
## binnedAge
## (10,20] (20,30] (30,40] (40,50] (50,60] (60,70]
##       3     113     224      97      16       0
```

That's not quite what we wanted: 30 is in with the 20s, for example. Here's how we fix that.

```
HELPrct <- HELPrct %>%
  mutate(binnedAge = cut(age, breaks=c(10,20,30,40,50,60,70), right=FALSE) )
tally( ~ binnedAge, data = HELPrct )
```

```
## binnedAge
## [10,20) [20,30) [30,40) [40,50) [50,60) [60,70)
##       1      97     232     105      17       1
```

We won't use this very often, since typically seeing this information in a histogram is more useful.

2.1.3 Cross-tables: Tabulating two or more variables

`tally()` can also compute cross tables for two (or more) variables.

```
tally(sex ~ substance, data=HELPrct)
```

```
##         substance
## sex      alcohol cocaine heroin
##   female      36      41     30
##   male       141     111     94
```

```
tally(~ sex + substance, data=HELPrct)
```

```
##         substance
## sex      alcohol cocaine heroin
##   female      36      41     30
##   male       141     111     94
```

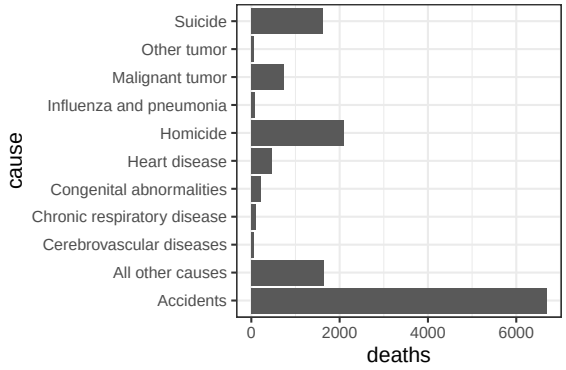
2.2 Working with Pre-Tabulated Data

Sometimes data arrive pre-tabulated. We can use `gf_col()` instead of `gf_bar()` to graph pre-tabulated data.

```
library(abd)            # data sets from Analysis of Biological Data
TeenDeaths
```

```
##                          cause deaths
## 1                    Accidents   6688
## 2                     Homicide   2093
## 3                      Suicide   1615
## 4               Malignant tumor    745
## 5                 Heart disease    463
## 6     Congenital abnormalities    222
## 7  Chronic respiratory disease    107
## 8      Influenza and pneumonia     73
## 9     Cerebrovascular diseases     67
## 10                 Other tumor     52
## 11            All other causes   1653
```

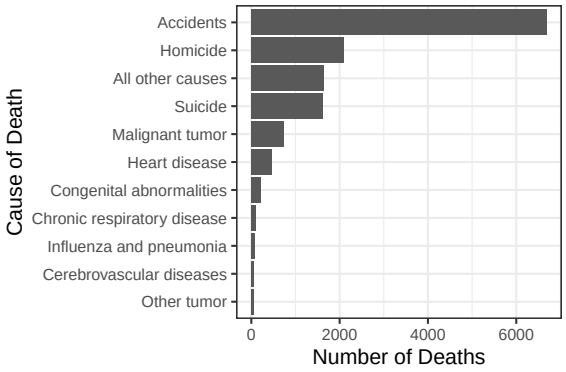
```
gf_col(deaths ~ cause, data=TeenDeaths) %>%
  gf_refine(coord_flip())
```



Notice that by default the causes are displayed in alphabetical order. R assumes that categorical data is nominal (that is, there is no particular natural or logical ordering to the categories) unless you say otherwise.

Here is an easy way to have things appear in a different order. The causes of death are reordered in order of increasing number of `deaths` caused.

```
gf_col( deaths ~ reorder(cause, deaths), data = TeenDeaths) %>%
  gf_refine(coord_flip()) %>%
  gf_labs(x = 'Cause of Death', y = 'Number of Deaths')
```



2.3 Summarizing Distributions of Quantitative Variables

Important Note

Numerical summaries are a convenient way to describe a distribution, but remember that numerical summaries do not necessarily tell you everything there is to know about a distribution. When working with a new dataset, it is *always* important to explore the data as fully as possible (commonly including graphical as well as numerical summaries, and sometimes even examining the data table directly) before accepting any simplified summary as a good representation of the data. You might discover certain patterns in the data, interesting features, or even outliers or mistakes in the data, that make certain summaries misrepresentations of the whole.

Notation

In statistics n (or sometimes N) almost always means the number of observations (i.e., the number of rows in a data frame).

If y is a variable in a data set with n cases, we can denote the n values of y as

- $y_1, y_2, y_3, \ldots, y_n$ (in the original order of the data).
- $y_{(1)}, y_{(2)}, y_{(3)}, \ldots y_{(n)}$ (in sorted order from smallest to largest).

The symbol $\sum$ represents summation (adding up a bunch of values).

2.4 Measures of Center

Measures of center attempt to give us a sense of what is a typical value for the distribution.

$$\text{mean of } y = \overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} = \frac{\text{sum of values}}{\text{number of values}}$$

$$\text{median of } y = \text{the "middle" number (after putting the numbers in increasing order)}$$

- The mean is the "balancing point" of the distribution.
- The median[2] is the 50th percentile: half of the distribution is below the median, half is above.
- If the distribution is symmetric, then the mean and median are the same.
- In a skewed distribution, the mean is pulled farther toward the tail than the median is.
- *A few very large or very small values can change the mean a lot,* so the mean is **sensitive to outliers** and is a better measure of center when the distribution is symmetric than when it is skewed.
- The median is a **resistant measure** (resistant to the presence of outlier) – it is not affected much by a few very large or very small values.

2.5 Measures of Spread

$$\text{variance of } y = s_y^2 = \frac{\sum_{i=1}^{n} (y_i - \overline{y})^2}{n-1}$$

$$\begin{aligned} \text{standard deviaiton of } y = s_y &= \sqrt{s_y^2} \\ &= \text{square root of variance} \end{aligned}$$

$$\begin{aligned} \text{interquartile range} = \text{IQR} &= Q_3 - Q_1 \\ &= \text{difference between first and third quartiles (defined shortly)} \end{aligned}$$

[2] A note about calculating medians: If the number of datapoints is odd, the median is the middle value (after putting the observations in increasing order). In cases where there is an even number of observations, the median is the average of the middle two observations.

- Roughly, the standard deviation is the "average deviation from the mean". (That's not exactly right because of the squaring involved and because we are dividing by $n-1$ instead of by n. More on that denominator later.)
- The mean and standard deviation are especially useful for describing **normal distributions** and other unimodal, symmetric distributions that are roughly "bell-shaped". (We'll learn more about normal distributions later.)
- Like the mean, the variance and standard deviation are sensitive to outliers and less suited for summarizing skewed distributions.
- It is perhaps of some value to compute the variance and standard deviation by hand once or twice to make sure you understand how these measures are defined, but we will typically let R do the calculations for us.

To get a numerical summary of a variable (a statistic), we need to tell R which statistic we want and the variable and data frame involved. There several ways we can do this in R. Here are several ways to get the mean, for example:

```
mean(HELPrct$age)            # this is the old fashioned way
```

```
## [1] 35.65
```

```
mean(~ age, data = HELPrct)  # similar to our plotting methods; only works for some functions
```

```
## [1] 35.65
```

```
df_stats(~ age, data = HELPrct, mean)  # formula-based and very flexible
```

```
##   response  mean
## 1      age 35.65
```

Using the formula style, we can now compute several different statistics.

```
mean( ~ age, data = HELPrct)
```

```
## [1] 35.65
```

```
sd( ~ age, data = HELPrct)
```

```
## [1] 7.71
```

```
var( ~ age, data = HELPrct)
```

```
## [1] 59.45
```

```
median( ~ age, data = HELPrct)
```

```
## [1] 35
```

```
IQR( ~ age, data = HELPrct)

## [1] 10

favstats( ~ age, data = HELPrct)  # this computes several statistics at once

##  min Q1 median Q3 max  mean   sd   n missing
##   19 30     35 40  60 35.65 7.71 453       0
```

It is also possible to compute these statistics separately for each of several groups. The syntax is much like the the syntax we used when plotting. In fact, we have two choices for the formula: `y ~ x` or `~ x | z`.

```
mean(age ~ sex, data = HELPrct)

## female   male
##  36.25  35.47

sd(age ~ sex, data = HELPrct)

## female   male
##  7.585  7.750

favstats( ~ age | sex, data = HELPrct )

##      sex min Q1 median   Q3 max  mean    sd   n missing
## 1 female  21 31     35 40.5  58 36.25 7.585 107       0
## 2   male  19 30     35 40.0  60 35.47 7.750 346       0
```

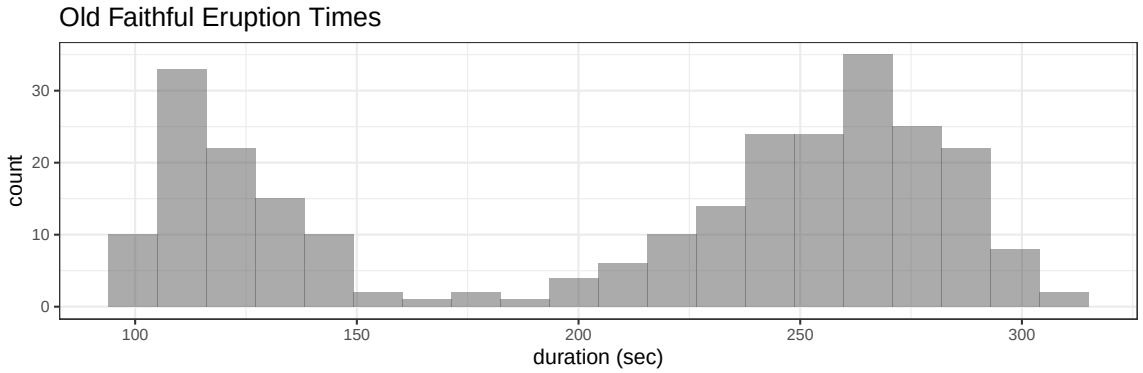
2.5.1 A word of caution

None of these measures (especially the mean and median) is a particularly good summary of a distribution if the distribution is not unimodal. The histogram below shows the lengths of eruptions of the Old Faithful geyser at Yellowstone National Park.

```
df_stats(~ Duration, data = oldfaith)

##   response min  Q1 median    Q3 max  mean    sd   n missing
## 1 Duration  96 130    240 267.8 306 209.9 68.39 270       0

gf_histogram( ~ Duration, data = oldfaith,  bins = 20) %>%
        gf_labs(title = "Old Faithful Eruption Times", x = "duration (sec)")
```



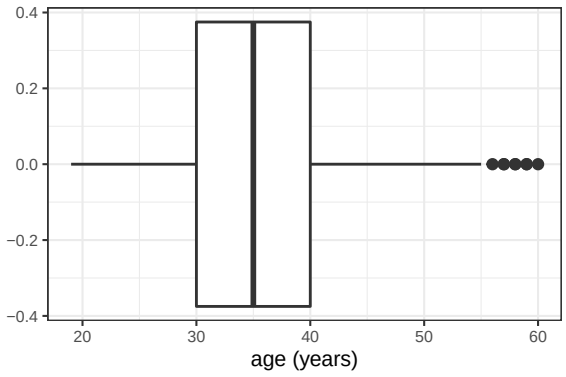
Notice that the mean and median do not represent typical eruption times very well. Nearly all eruptions are either quite a bit shorter or quite a bit longer. (This is especially true of the mean.)

2.5.2 Box plots

Boxplots (also called box-and-whisker plots) are a graphical representation of a **5-number summary** of a quantitative variable. The five numbers are the five **quantiles**:

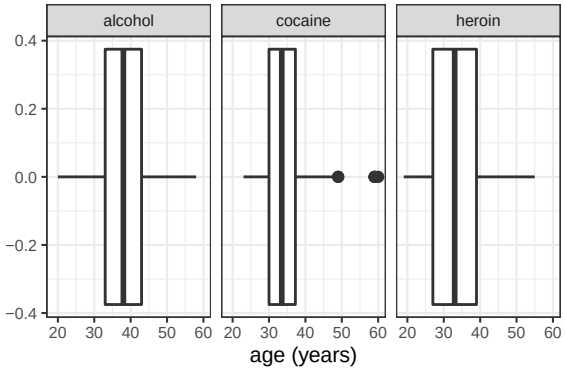
- Q_0, the minimum
- Q_1, the first quartile (25th percintile)
- Q_2, the median (50th percentile)
- Q_3, the third quartile (75th percentile)
- Q_4, the maximum

```
gf_boxplot(~age, data=HELPrct)
```



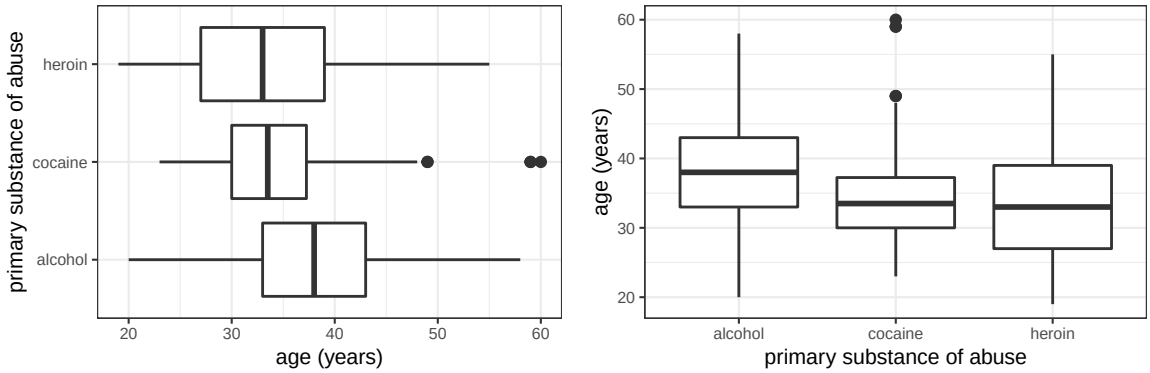
Boxplots provide a way of comparing multiple groups that is especially informative and visually effective. Here is one way to make boxplots of multiple groups (it should look familiar from what we know about histogram):

```
gf_boxplot(~age | substance, data=HELPrct)
```



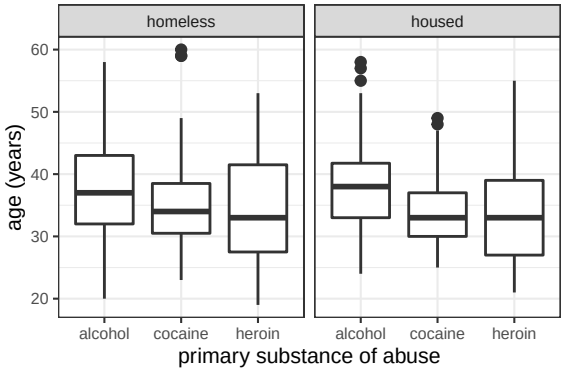
But `gf_boxplot()` has a better way. Put the quantitative variable on one side of the wiggle and the categorical on the other. The placement determines which goes along the vertical axis and which along the horizontal axis – just like it did for `gf_point()`.

```
gf_boxplot(substance ~ age, data=HELPrct)
gf_boxplot(age ~ substance, data=HELPrct)
```



And we can combine this idea with conditioning. Careful: The quantitative variable must be the "y" variable in the formula.

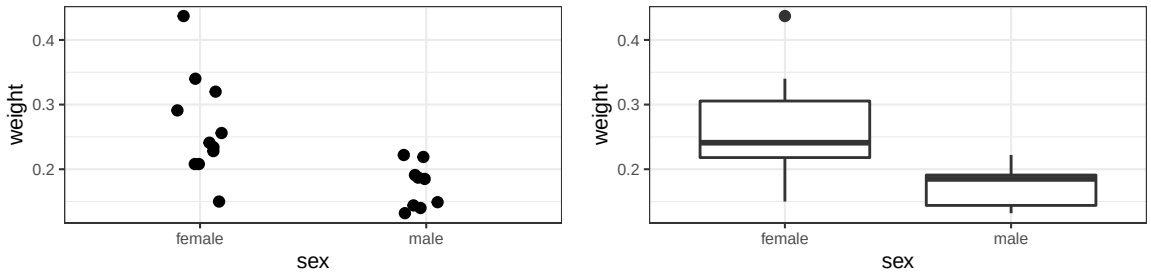
```
gf_boxplot(age ~ substance | homeless, data=HELPrct)
```



2.5.3 Small data sets

When we have relatively small data sets, it may not make sense to use a boxplot. With very few observations, boxplots can be misleading, in that they suggest the presence of more observations than are really contained in the dataset. In these cases, it may be better to display all the data. `gf_jitter()` allows you to put a categorical variable along one axis and a quantitative variable along the other. For some data sets, either option can produce a plot that gives a good picture of the data.

```
gf_jitter( weight ~ sex, data = Mosquitoes, width = 0.1, height = 0)
gf_boxplot( weight ~ sex, data = Mosquitoes)
```



Note the effect of the `width = 0.1, height = 0` – this tells `gf_jitter()` to move each data point slightly left or right, but not at all up or down. to reduces overplotting (data points being plotted exactly on top of one another) without losing any information, making it clearer how many data points were observed for each possible combination of x- and y-values.

2.6 Summarizing Categorical Variables

The most common summary of a categorical variable is the **proportion** of observations in each category. For a single category:

$$\hat{p} = \frac{\text{number in one category}}{n}$$

Proportions can be expressed as fractions, decimals or percents. For example, if there are 10 observations in one category and $n = 50$ observations in all, then

$$\hat{p} = \frac{10}{25} = \frac{2}{5} = 0.40 = 40\%$$

If we code our categorical variable using 1 for observations in a single category of interst – "the one category" – and 0 for observations in any other category, then *a proportion is a sample mean.*

$$\frac{1+1+1+1+1+1+1+1+1+1+0+0+0+0+0+0+0+0+0+0+0+0+0+0+0}{25} = \frac{10}{25}$$

2.7 Relationships Between Two Variables

It is also possible to give numerical summaries of the relationship between two variables. The most common one is the **correlation coefficient**, which we will learn about later.

2.8 Exercises

2.1 Create a data set with $n = 6$ values, each an integer between 0 and 10 (inclusive) that has the smallest possible variance. Compute the mean and variance of this data set "by hand" (that is, without using `mean()` or `sd()` or `var()` in R or similar features on a calculator).

2.2 Create a data set with $n = 6$ values, each an integer between 0 and 10 (inclusive) that has the largest possible variance. Compute the variance of this data set "by hand" (that is, without using `mean()` or `sd()` or `var()` in R or similar features on a calculator).

2.3 Create side-by-side boxplots of the variable `i1` (average number of drinks per day) comparing the different `substance` groups in the `HELPrct` data frame.

For each `substance` group, explain how you can tell from the boxplots whether the mean will be larger than the median or the median larger than the mean.

2.4 Compute the mean and median values of `i1` (average number of drinks per day) for each of the `substance` groups in the `HELPrct` data frame.

3 Probability

3.1 Key Definitions and Ideas

random process A repeatable process that has multiple unpredictable potential outcomes.

Although we sometimes use language that suggests that a *particular result* is random, it is really the *process* that is random, not its results.

outcome A potential result of a random process.

sample space The set of all possible potential outcomes of a random process.

event A subset of the sample space. That is, a set of outcomes (possibly all or none of the outcomes).

Statisticians often use capital letters from the beginning of the alphabet for events.

trial One repetition of a random process.

mutually exclusive events. Events that cannot happen on the same trial.

probability A numerical value between 0 and 1 assigned to an event to indicate how often the event occurs (in the long run).

random variable A random variable is a variable whose value is a numerical outcome of a random process.

Examples of random variables:

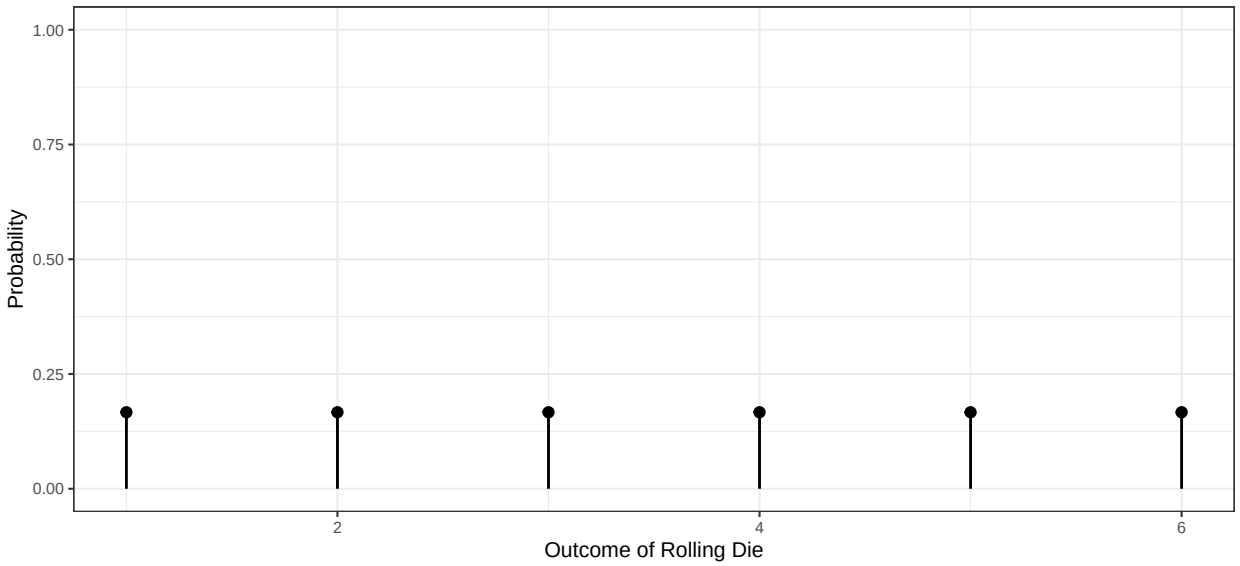
- Roll a die and record the number.
- Roll two dice and record the sum.
- Flip 100 coins and count the number of heads.
- Sample 1000 people and count how many approve of the job the president is doing.

Note: Statisticians usually use capital letters (often from the end of the alphabet) for random variables, like this: Let X be the number of heads in 10 flips of a fair coin. What is $P(X = 5)$?

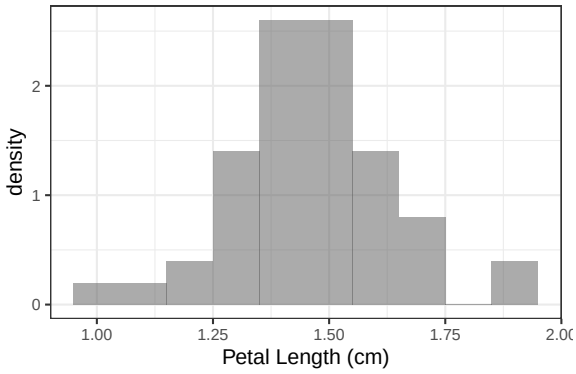
probability distribution The distribution of a random variable. (Remember that a distribution describes *what values?* and *with what freqency?*)

As an example of a probability distribution, we can first consider a *discrete* random variable. Most of the examples of random variables given above are discrete. In other words, the values they can take on come from a set containing a finite number of possible values. For example, if you roll a 6-sided die and record the number that comes up, there are only size possible outcomes, which are equally likely: the integers 1, 2, 3, 4, 5 and 6.

For discrete random variables, the probability distribution shows all the possible values on the x-axis, and the likelihood of observing each of those values on the y-axis. Since there are a finite number of possible values that can be observed, these likelihoods are actually the *probabilities* of observing each outcome, and the sum of all the probabilities must be 1 (see section 3.3 for details). For our example, where we rolled a die and recorded the value:



Things are a bit more complicated for *continuous* random variables (the ones that can take on any numerical value). Here, there sample space (the set of possible distinct values the random variable can take on) is infinite. One consequence of this fact is that the interpretation of the y-axis values of the probability distribution changes. The y-axis will still indicate the relative likelihood of observing any given value of the random variable. However, here the random variable can take on an infinite number of possible values. In this case, we can't interpret the y-axis values as probabilities. They y axis units are called "Likelihood" or "Density", and they indicate the relative frequency of each outcome. For a densityplot, Density is scaled such that the integral over all possible x-values (the area under the curve) is 1. (For a histogram, Density is scaled so that the total area of all the boxes added together is 1.) We can think of the histograms and density plots we have been creating using continuous variables from R datasets as attempts to use data to approximate the distributions of random variables. For example, we might consider the growth of flower petals of the iris *Iris setosa* as a random process, and let `X` be a random variable that is the length of each iris petal. We could plot a histogram to approximate the distribution of `X` using the variable `Petal.Length` from the `iris` data (from the `datasets` package in base R).



3.2 Calculating Probabilities Empirically

We would like to calculate the probability of an event A, denoted $\mathrm{P}(A)$.

In the next section, we will see how to calculate probabilities based on the Axioms of probability, and logic. But first, we will consider ways to make the calculations empirically – based on observing many repetitions of a random process (in real life or in a computer simulation) and observing how often an event of interest occurs.

Random processes are repeatable, so practically, we can calculate empirical probabilities by simply repeating the process over and over and keeping track of how often the event A occurs. For example, we could flip a coin 10,000 times and see what fraction are heads.[1]

$$\text{Empirical Probability} = \frac{\text{number of times } A \text{ occured}}{\text{number of times random process was repeated}}$$

Modern computing provides another way to compute empirical probabilities. If we can simulate our random process on a computer, then we can repeat the process many times very quickly.

Example 3.2.1. Q. What is the probability of getting exactly 5 heads if you flip a fair coin 10 times? Using our random variable notation, let X be the number of heads in 10 flips of a fair coin. We want to know $\mathrm{P}(X = 5)$.

A. The `rflip()` function simulates flipping a coin as many times as we like.

```
rflip(10)
```

```
##
## Flipping 10 coins [ Prob(Heads) = 0.5 ] ...
##
## T T H H H T T T H H
##
## Number of Heads: 5 [Proportion Heads: 0.5]
```

The `do()` function allows us to execute an R command ("do" somthing in R) over and over, as many times as we choose. Here, our `rflip()` command simulates 10 coin-flips. First we'll "do" our command three times and show the results. Then we'll do it 10,000 times and store the results in a variable called `tosses`, so we can create a table and a plot showing the empirical distribution.

```
do(3) * rflip(10)
```

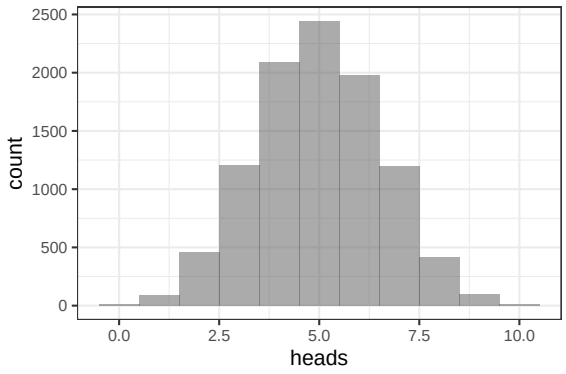
```
##    n heads tails prop
## 1 10     6     4  0.6
## 2 10     5     5  0.5
## 3 10     4     6  0.4
```

```
do(10000) * rflip(10) -> tosses
tally(~ heads, data = tosses, format = "prop")
```

```
## heads
##      0      1      2      3      4      5      6      7      8      9     10
## 0.0009 0.0088 0.0459 0.1205 0.2091 0.2442 0.1978 0.1194 0.0419 0.0102 0.0013
```

[1] This has actually been done a couple of times in history, including once by mathematician John Kerrich while he was a prisoner of war during World War II.

```
gf_histogram( ~ heads, data = tosses, binwidth = 1)
```



Based on this sample, we would estimate that $\mathrm{P}(X = 5) \approx 0.244$.

Example 3.2.2. Q. Use simulations to estimate the probability of rolling doubles using two fair standard dice.

A. We can simulate rolling a die with the following code:

```
1:6                # the numbers 1 through 6

## [1] 1 2 3 4 5 6

resample(x = 1:6, size=10)  # ten rolls of a 6-sided die

##  [1] 3 3 4 5 6 1 6 2 6 3
```

The first 2 input arguments of `resample()` are `x` (the set of values from which you want to resample) and `size` (the number of items to choose from `x`). You can also think of `size` as the number of *times* to sample from `x`, if you are imagining sampling one item from `x` each time.

If we do this 10,000 times for each of two dice...

```
die1 <- resample(1:6, 10000)
die2 <- resample(1:6, 10000)
# let's check that things look reasonable
head(die1)

## [1] 3 2 4 5 6 5

head(die2)

## [1] 1 6 5 3 1 3
```

Then we can tabulate how often the two numbers matched in one of two ways:

```
tally( ~(die1==die2) )    # NOTE the double == here


## (die1 == die2)
##  TRUE FALSE
##  1708  8292


prop( ~(die1==die2) )     # NOTE the double == here


## prop_TRUE
##    0.1708
```

So the probability appears to be approximately 0.171.

Example 3.2.3. Q. Use simulation to estimate the probability of rolling a sum of 8 when rolling two fair six-sided dice.

A. We have already generated 10000 random rolls, so let's just reuse them. (Alternatively, we could generate new rolls.)

```
s <- die1 + die2
# R adds element-wise:
#   first entry of die1 + first of die2,
#   second to second, etc.
prop( ~ (s == 8) )


## prop_TRUE
##    0.1504
```

We can estimate the probability of any sum the same way.

```
tally( ~ s )


## s
##    2    3    4    5    6    7    8    9   10   11   12
##  276  568  836 1061 1353 1637 1504 1108  825  531  301


tally( ~ s, format = "percent" )   # if we are too lazy to divide by 10000 ourselves


## s
##     2     3     4     5     6     7     8     9    10    11    12
##  2.76  5.68  8.36 10.61 13.53 16.37 15.04 11.08  8.25  5.31  3.01
```

Here's a slightly fancier version that puts all the information into a data frame. Note the use of the function `data.frame()` to create the data table:

```
rolls <- data.frame( first = die1, second = die2, sum = die1 + die2 )
head(rolls)

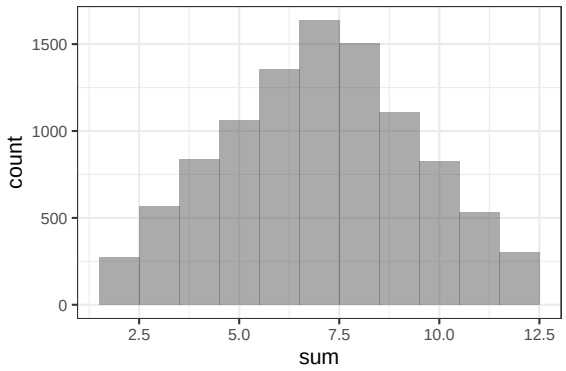

##   first second sum
## 1     3      1   4
```

```
## 2     2      6   8
## 3     4      5   9
## 4     5      3   8
## 5     6      1   7
## 6     5      3   8
```

```
tally( ~sum, data = rolls, format = "proportion")
```

```
## sum
##      2      3      4      5      6      7      8      9     10     11     12
## 0.0276 0.0568 0.0836 0.1061 0.1353 0.1637 0.1504 0.1108 0.0825 0.0531 0.0301
```

```
gf_histogram( ~ sum, data = rolls, binwidth = 1)    # setting width is important for integer data
```



3.3 Calculating Probabilities Theoretically

The theoretical method combines

1. Some basic facts about probability (the Probability Axioms and Rules),
2. Some assumptions about the particular situation at hand, and
3. Mathematical reasoning (arithmetic, algebra, logic, etc.).

3.3.1 The Three Probability Axioms

Let S be the sample space and let A and B be events.

1. Probability is between 0 and 1: $0 \leq \mathrm{P}(A) \leq 1$.
2. The probability of the sample space is 1: $\mathrm{P}(S) = 1$.
3. Additivity: If A and B are mutually exclusive, then $\mathrm{P}(A \text{ or } B) = \mathrm{P}(A) + \mathrm{P}(B)$.

Notation Notes

$\mathrm{P}(A \text{ or } B)$ is the probability that either A or B (or both) occurs. Often this is written $\mathrm{P}(A \cup B)$. $A \cup B$ is usually read "A union B". The union of two sets is the set that contains all elements of both sets.

$\mathrm{P}(A \text{ and } B)$ is the probability that *both* A and B occur. This is also written $\mathrm{P}(A \cap B)$. $A \cap B$ is usually read "A intersect B".

Saying that A and B are mutually exclusive is the same as saying that there are no outcomes in $A \cap B$, i.e., that $A \cap B = \emptyset$.

3.3.2 Other Probability Rules

These rules all follow from the axioms (although we will not necessarily prove them all here).

The Addition Rule

If events A and B are mutually exclusive, then

$$\mathrm{P}(A \text{ or } B) = \mathrm{P}(A) + \mathrm{P}(B) \, .$$

More generally,

$$\mathrm{P}(A \text{ or } B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \text{ and } B) \, .$$

The Complement Rule

$$\mathrm{P}(\text{not } A) = 1 - \mathrm{P}(A)$$

The Equally Likely Rule

If the sample space consists of n equally likely outcomes, then the probability of an event A is given by

$$\mathrm{P}(A) = \frac{\text{number of outcomes in } A}{n} = \frac{|A|}{|S|} \, .$$

Warning: One of the most common mistakes in probability is to apply this rule when the outcomes are not equally likely.

Examples 3.3.1.

1. Coin Toss: $\mathrm{P}(\text{heads}) = \frac{1}{2}$ if heads and tails are equally likely.
2. Rolling a Die: $\mathrm{P}(\text{even}) = \frac{3}{6}$ if the die is fair (each of the six numbers equally likely to occur).

3. Sum of two Dice: the sum is a number between 2 and 12, but these numbers are NOT equally likely.

   There are 36 equally likely combinations of two dice:

1,1	2,1	3,1	4,1	5,1	6,1
1,2	2,2	3,2	4,2	5,2	6,2
1,3	2,3	3,3	4,3	5,3	6,3
1,4	2,4	3,4	4,4	5,4	6,4
1,5	2,5	3,5	4,5	5,5	6,5
1,6	2,6	3,6	4,6	5,6	6,6

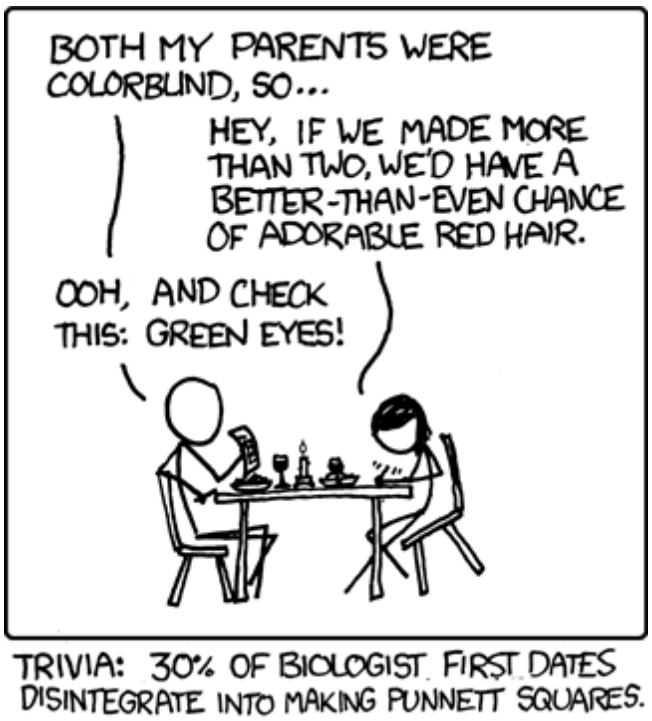
   Let X be the sum of two dice.

   - $\mathrm{P}(X = 3) = \frac{2}{36} = \frac{1}{18}$
   - $\mathrm{P}(X = 7) = \frac{6}{36} = \frac{1}{6}$
   - $\mathrm{P}(\text{doubles}) = \frac{6}{36} = \frac{1}{6}$

4. Punnet Squares

	A	a
A	AA	Aa
a	Aa	aa

   This example comes from animal or human genetics. Here, we consider a gene with two alleles: A is the dominant allele, and a is the recessive one. Each individual has two copies of every gene, so there are three possible combinations of alleles (called "genotypes"): AA, Aa, and aa. AA and Aa individuals have the dominant A physical characteristic (called the "phenotype"); aa individuals have the recessive a phenotype. Imagine that two Aa individuals mate and produce offspring. In this Aa $\times$ Aa cross, if A is the dominant allele, then the probability of the dominant phenotype is $\frac{3}{4}$, and the probability of the recessive phenotype is $\frac{1}{4}$ because each of the four possible crossings is equally likely.



Cartoon credit: `http://xkcd.com/634/`

3.4 Conditional Probability

Example 3.4.1. Q. Suppose a family has two children and one of them is a boy. What is the probability that the other is a girl?

A. We'll make the simplifying assumption that boys and girls are equally likely (which is not exactly true). Under that assumption, there are four equally likely families: BB, BG, GB, and GG. But only three of these have at least one boy, and we already know our family has at least one boy, so our sample space is really $\{BB, BG, GB\}$. Of these, two have a girl as well as a boy. So the probability is 2/3 (see Figure 3.1).

GG **GB** **BG** BB probability = 2/3

Figure 3.1: Illustrating the sample space for Example 3.4.1.

We can also think of this in a different way. In our original sample space of four equally likely families,

$$\begin{aligned} \mathrm{P}(\text{at least one girl}) &= 3/4\ , \\ \mathrm{P}(\text{at least one girl } \textit{and} \text{ at least one boy}) &= 2/4\ , \text{ and} \\ \frac{2/4}{3/4} &= 2/3\ ; \end{aligned}$$

so 2/3 of the time when there is at least one boy, there is also a girl. We will denote this probability as P(at least one girl | at least one boy). We'll read this as "the probability that there is at least one girl *given that* there is at least one boy". See Figure 3.2 and Definition 3.4.

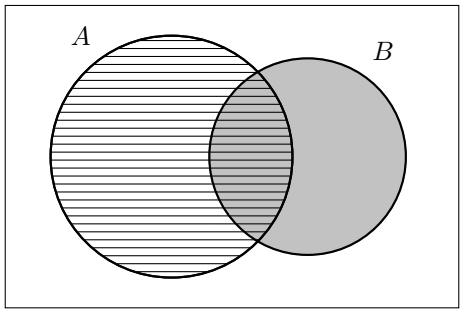


Figure 3.2: A Venn diagram illustrating the definition of conditional probability. $\mathrm{P}(A \mid B)$ is the ratio of the area of the football shaped region that is both shaded and striped ($A \cap B$) to the area of the shaded circle (B).

Let A and B be two events such that $\mathrm{P}(B) \neq 0$. The **conditional probability** of A given B is defined by

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)}\ .$$

If $\mathrm{P}(B) = 0$, then $\mathrm{P}(A \mid B)$ is undefined.

Example 3.4.2. A class of 5th graders was asked what color should be used for the class T-shirt, red or purple. The table below contains a summary of the students' responses:

	Color	
	Red	Purple
Girls	7	9
Boys	10	8

Q. Suppose we randomly select a student from this class. Let R be the event that a child prefers a red T-shirt. Let B be the event that the child is a boy, and let G be the event that the child is a girl. Express each of the following probabilities in words and determine their values:

- $\mathrm{P}(R)$,
- $\mathrm{P}(B \mid R)$,
- $\mathrm{P}(G \mid R)$,
- $\mathrm{P}(R \mid B)$,
- $\mathrm{P}(R \mid G)$,
- $\mathrm{P}(B \mid G)$.

A. The conditional probabilities can be computed in two ways. We can use the formula from the definition of conditional probability directly, or we can consider the condition event to be a new, smaller sample space and read the conditional probability from the table.

- $\mathrm{P}(R) = 17/34 = 1/2$ because 17 of the 34 kids prefer red
  This is the probability that a randomly selected student prefers red
- $\mathrm{P}(R \mid B) = \dfrac{10/34}{18/34} = \dfrac{10}{18}$ because 10 of the 18 boys prefer red
  This is the probability that a randomly selected boy prefers red
- $\mathrm{P}(B \mid R) = \dfrac{10/34}{17/34} = \dfrac{10}{17}$ because 10 of the 17 students who prefer red are boys.
  This is the probability that a randomly selected student who prefers red is a boy.
- $\mathrm{P}(R \mid G) = \dfrac{7/34}{16/34} = \dfrac{7}{16}$ because 7 of the 16 girls prefer red
  This is the probability that a randomly selected girl prefers red
- $\mathrm{P}(G \mid R) = \dfrac{7/34}{17/34} = \dfrac{7}{17}$ because 7 of the 17 kids who prefer red are girls.
  This is the probability that a randomly selected kid who prefers red is a girl.
- $\mathrm{P}(B \mid G) = \dfrac{0}{16/34} = 0$ because none of the girls are boys.
  This is the probability that a randomly selected girl is a boy.

One important use of conditional probability is as a tool to calculate the probability of an intersection.

Let A and B be events with non-zero probability. Then

$$\begin{aligned}\mathrm{P}(A \cap B) &= \mathrm{P}(A) \cdot \mathrm{P}(B \mid A) \\ &= \mathrm{P}(B) \cdot \mathrm{P}(A \mid B)\ .\end{aligned}$$

This follows directly from the definition of conditional probability by a little bit of algebra and can be generalized to more than two events.

Example 3.4.3. Q. If you roll two standard dice, what is the probability of doubles? (Doubles is when the two numbers match.)

A. Let A be the event that we get a number between 1 and 6 on the first die. So $\mathrm{P}(A) = 1$. Let B be the event that the second number matches the first. Then the probability of doubles is $\mathrm{P}(A \cap B) = \mathrm{P}(A) \cdot \mathrm{P}(B \mid A) = 1 \cdot \frac{1}{6} = \frac{1}{6}$ since regardless of what is rolled on the first die, 1 of the 6 possibilities for the second die will match it.

Example 3.4.4. Q. A 5-card hand is dealt from a standard 52-card deck. What is the probability of getting a flush (all cards the same suit)?

A. Imagine dealing the cards in order. Let A_i be the event that the ith card is the same suit as all previous cards. Then

$$\begin{aligned}
\mathrm{P}(\text{flush}) &= \mathrm{P}(A_1 \cap A_2 \cap A_3 \cap A_4 \cap A_5) \\
&= \mathrm{P}(A_1) \cdot \mathrm{P}(A_2 \mid A_1) \cdot \mathrm{P}(A_3 \mid A_1 \cap A_2) \cdot \mathrm{P}(A_4 \mid A_1 \cap A_2 \cap A_3) \\
&\quad \cdot \mathrm{P}(A_5 \mid A_1 \cap A_2 \cap A_3 \cap A_4) \\
&= 1 \cdot \frac{12}{51} \cdot \frac{11}{50} \cdot \frac{10}{49} \cdot \frac{9}{48}
\end{aligned}$$

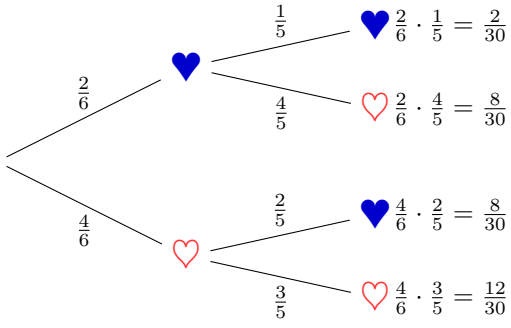
Example 3.4.5. Q. In a bowl are 4 red Valentine hearts and 2 blue Valentine hearts.

If you reach in without looking and select two of the Valentines, let X be the number of blue Valentines. Fill in the following probability table.

value of X	0	1	2
probability			

A. $\mathrm{P}(X = 2) = \mathrm{P}(\text{first is blue and second is blue}) = \mathrm{P}(\text{first is blue}) \cdot \mathrm{P}(\text{second is blue} \mid \text{first is blue}) = \frac{2}{6} \cdot \frac{1}{5} = \frac{2}{30}$. Similarly $\mathrm{P}(X = 0) = \mathrm{P}(\text{first is red and second is red}) = \mathrm{P}(\text{first is red}) \cdot \mathrm{P}(\text{second is red} \mid \text{first is red}) = \frac{4}{6} \cdot \frac{3}{5} = \frac{12}{30}$ Finally, $\mathrm{P}(X = 1) = 1 - \mathrm{P}(X = 0) - \mathrm{P}(X = 2) = 1 - \frac{14}{30} = \frac{16}{30}$

We can represent this using a **tree diagram** as well.



The edges in the tree represent conditional probabilities which we can multiply together to the probability that all events on a particular branch happen. The first level of branching represents what kind of Valentine is selected first, the second level represents the second selection.

Example 3.4.6. Q. Suppose a test correctly identifies diseased people 99% of the time and correctly identifies healthy people 98% of the time. Furthermore assume that in a certain population, one person in 1000 has the disease. If a random person is tested and the test comes back positive, what is the probability that the person has the disease?

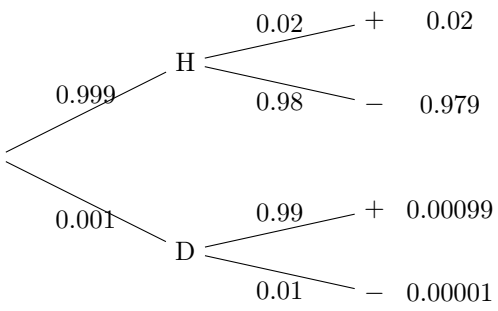
A. We begin by introducing some notation. Let D be the event that a person has the disease. Let H be the event that the person is healthy. Let $+$ be the event that the test comes back positive (meaning it indicates disease – probably a negative from the perspective of the person tested). Let $-$ be the event that the test is negative.

- $\mathrm{P}(D) = 0.001$, so $\mathrm{P}(H) = 0.999$.
- $\mathrm{P}(+ \mid D) = 0.99$, so $\mathrm{P}(- \mid D) = 0.01$.

  $\mathrm{P}(+ \mid D)$ is called the **sensitivity** of the test. (It tells how sensitive the test is to the presence of the disease.)
- $\mathrm{P}(- \mid H) = 0.98$, so $\mathrm{P}(+ \mid H) = 0.02$.

  $\mathrm{P}(- \mid H)$ is called the **specificity** of the test.
- $$\begin{aligned} \mathrm{P}(D \mid +) &= \frac{\mathrm{P}(D \cap +)}{\mathrm{P}(+)} \\ &= \frac{\mathrm{P}(D) \cdot \mathrm{P}(+ \mid D)}{\mathrm{P}(D \cap +) + \mathrm{P}(H \cap +)} \\ &= \frac{0.001 \cdot 0.99}{0.001 \cdot 0.99 + 0.999 \cdot 0.02} = 0.047. \end{aligned}$$

A tree diagram is a useful way to visualize these calculations.



This low probability surprises most people the first time they see it. This means that if the test result of a random person comes back positive, the probability that that person has the disease is less than 5%, even though the test is "highly accurate". This is one reason why we do not routinely screen an entire population for a rare disease – such screening would produce many more false positives than true positives.

Of course, if a doctor orders a test, it is usually because there are some other symptoms. This changes the *a priori* probability that the patient has the disease.

3.4.1 Independence

Let A and B be two events such that $\mathrm{P}(B) = \mathrm{P}(B \mid A)$. Such events are called **independent**.

When events are independent, then $\mathrm{P}(A \text{ and } B) = \mathrm{P}(A) \cdot \mathrm{P}(B \mid A) = \mathrm{P}(A) \cdot \mathrm{P}(B)$. This makes probability calculations much simpler – but it only applies for independent events.

Example 3.4.7. Q. What is the probability of rolling double sixes with standard 6-sided dice?

A. Let A be the event that the first die is a 6 and let B be the event that the second die is a 6. Since A and B are independent, $\mathrm{P}(A \text{ and } B) = \mathrm{P}(A) \cdot \mathrm{P}(B) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$.

Example 3.4.8. Q. What is the probability of flipping a coin five times and getting 5 heads?

A. Since each coin toss is independent of the others, the probability of getting five heads is the product of the probabilities of each coin coming up heads:

$$\mathrm{P}(5 \text{ heads in } 5 \text{ flips}) = (0.5)^5 = 0.031$$

Example 3.4.9. Q. A manufacturer claims that 99% of its parts will still be functioning properly two years after purchase. If you purchase 10 of these parts, what is the probability that all 10 of them are still functioning properly two years later (assuming the manufacturer's claim is correct)?

A. Let G_i be the event that part i is still functioning properly after two years. We want to calculate

$$\mathrm{P}(G_1 \text{ and } G_2 \text{ and } \cdots \text{ and } G_{10}) \, .$$

If we assume the lifetimes of the parts are independent, then

$$\mathrm{P}(G_1 \text{ and } G_2 \text{ and } \cdots \text{ and } G_{10}) = \underbrace{.99 \cdot .99 \cdot .99 \cdots .99 = .99^{10}}_{10 \text{ of these}} = 0.904 \, .$$

The independence assumption may or may not be valid. That depends on the manufacturing process. For example, if the primary way a part goes bad is that the package is dropped during shipping, then if you by a box of 10 and the first part is bad, they will all be bad. And if the box was handled carefully and never dropped, and the first part used is good, they will likely all be good. So in that extreme case, the probability that all 10 are functioning properly after two years is 99%.

3.5 Exercises

3.1 Amy is a 92% free throw shooter. If she shoots 100 free throws after practice, what is the probability that she makes at least 95 of them? Use simulation to estimate this probability.

(You can use `rflip()` to simulate shooting free throws. The `prob` argument lets you set the probability. In this case, you need to set it to 0.92. Then think of a head as a made free throw and a tail as a missed free throw.)

3.2

a) Use simulation to estimate the probability of rolling a difference of 2 when rolling two fair six-sided dice.

b) Make a histogram showing the results for all of the possible differences.

3.3 Use simulation to estimate the probability that when dealing 5 cards from a standard (well-shuffled) deck of 52 cards all five are diamonds.

You can simulate the deck of cards using the numbers 1 through 52 and consider the numbers 1 through 13 to be the diamonds. Instead of using `resample()`, which would allow you to get the same card more than once, we need to use `sample()`, which does not. (You can also use `deal()` which does the same thing.)

```
sample(1:52, 5)

## [1] 10 38 43 25 37

sample(1:52, 5)

## [1] 19 15 36 24 31

deal(1:52, 5)

## [1] 46 43 20 29 40

deal(1:52, 5)

## [1] 17 24 10 11 37
```

There is another way to make the calculation, using the function `sum()`. R can tell you how many cards are below 14 using `sum()` because R turns TRUE into 1 and FALSE into 0 when you do a sum.

```
sum( sample(1:52, 5) < 14 )
```

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

```
sum( sample(1:52, 5) < 14 )
```

```
## [1] 0
```

```
sum( sample(1:52, 5) < 14 )
```

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

You can use `do()` to do this many times. (Three is *not* many We just do a small number here for illustration purposes.)

```
do(3) * sum( sample( 1:52, 5 ) < 14 )
```

```
##   sum
## 1   0
## 2   0
## 3   1
```

3.4 Parts in a manufacturing plant go through two quality control checks before they are shipped. 99% of parts pass inspection A and 98% parts pass inspection B. 0.5% fail both inspections.

What percentage of parts pass both inspections?

3.5 Let X be the sum of the results of rolling two fair six-sided dice.

a) What is P(X is even and $X < 5$)?

b) What is P(X is even or $X < 5$)?

3.6 Let Y be the difference between the larger and smaller number when two fair dice are rolled. (So if you roll a 2 and a 4, then the value of Y is 2.)

a) What is P($Y = 2$)?

b) What are the other possible values of Y?

c) Calculate the probability for each possible value of Y and put those values in a table.

3.7 For the probabilities below, you may assume the that for each birth, the probability of having a boy or a girl is 1/2 and that each birth is independent of other births.

a) Suppose a family has three kids. What is the probability that at least one of the kids is a boy?

b) Suppose a family has three kids, at least one of which is a girl. Now what is the probability that at least one of the kids is a boy?

c) Suppose a family has three kids, at least two of which are girls. Now what is the probability that at least one of the kids is a boy?

3.8 A device is assembled from two primary parts. 2% of the first type of part are defective and 3% of the other type of part are defective. The device only functions properly if both parts are functioning properly.

a) What assumption do you need to make to calculate the probability that a device assembled in this way will function properly? Is it a reasonable assumption in this situation? Explain.

b) What is the probability that that a device assembled in this way will function properly?

3.9 In the situation of Example 3.4.6, how does the answer change if the baseline probability of having the disease is 1/10 instead of 1/1000? (This might be the case if a person is exhibiting symptoms, for example.)

3.10 According to the CDC, "Compared to nonsmokers, men who smoke are about 23 times more likely to develop lung cancer and women who smoke are about 13 times more likely." According to the American Lung Association: "In 2008, 21.1 million (18.3%) women smoked in the United States compared to 24.8 million (23.1%) men."

a) If you learn that a person is a smoker and no nothing else about the person, what is the probability that the person is a woman?

b) If you learn that a woman has been diagnosed with lung cancer, and you know nothing else about her, what is the probability that she is a smoker?

c) If you learn that a man has been diagnosed with lung cancer, and you know nothing else about him, what is the probability that he is a smoker?

3.11 A manufacturing plant has kept records that show that the number of parts produced each day and on the proportion of parts that are defective.

	Monday	Tuesday	Wednesday	Thursday
Proportion of weekly production	20%	25%	28%	27%
Rate of defective parts	2%	1.5%	1%	3%

a) If you order a part from this company, what is the probability that it was produced on a Monday or a Thursday?

b) If you order a part from this company and it is defective, what is the probability that it was produced on a Monday or a Thursday?

c) If you order a part from this company and it functions properly, what is the probability that it was produced on a Monday or Thursday?

Express your answers to 3 significant digits and avoid internal rounding.

3.12 An engineer orders a shipment of 100 indentical parts. Before accepting the shipment, he tests three of them. If the they all test good, he accepts the entire shipment. If any of them tests bad, he rejects the shipment.

Given the good price he has gotten on these parts, the engineer would be satisfied if at least 95% of the parts are good.

a) Suppose that there are 5 bad parts in the shipment. What is the probability that the shipment is rejected (even though the engineer would actually have been satisfied with the shipment)?

b) Suppose that there are 10 bad parts in the shipment. What is the probability that the shipment is accepted (even though the engineer would not be satisfied with this shipment)?

3.13 The blue M&M was introduced in 1995. Before then, the color mix in a bag of plain M&Ms was 30% Brown, 20% Yellow, 20% Red, 10% Green, 10% Orange, 10% Tan. Afterward it was 24% Blue , 20% Green, 16% Orange, 14% Yellow, 13% Red, 13% Brown.

A friend of mine has two bags of M&Ms, and he tells me that one is from 1994 and one from 1996. He won't tell me which is which, but he gives me one M&M from each bag. One is yellow and one is green. What is the probability that the yellow M&M came from the 1994 bag?

Excellent health statistics - smokers are less likely to die of age related illnesses.'

4 Random Variables

random variable a random process that results in a number

We have already seen notation for and a few examples of random variables. In this chapter we will learn a bit more about random variables. We will focus our attention on two types of random variables: discrete random variables and continuous random variables.[1]

4.1 Discrete Random Variables

A discrete random variable takes on values from a discrete set of possibilities, typically either a finite set or a subset of the integers. Here are some examples.

1. If we roll a die and record the number, there are six possible values, so the random variable is discrete.
2. If we keep flipping a coin until we get a head and record the number of coin tosses, then the possible values of the random variable are $1, 2, 3, \ldots$ This is also a discrete random variable.

For each of the possible values of a discrete random variable, there is some probability of that value occuring. So to specify a discrete random variable, we need to specify those probabilities. When there are only a small number of possible value, we can do this with a table.

value of X	0	1	2
probability	0.2	0.5	0.3

This is really just one way of describing a function, called the **probability mass function** (or pmf). The pmf satisfies

$$f(x) = \mathrm{P}(X = x)$$

Sometimes instead of providing a table, we will be able to specify the pmf using a formula. For example, we could define a pmf g by

$$g(y) = (2 - |1 - y|)/4 \text{ for } y \in \{0, 1, 2\}$$

which is the same as specifying g with the following table.

[1] There are important examples of random variables that are neither discrete nor continuous.

value of Y	0	1	2
probability	0.25	0.5	0.25

Here's one more example. Let W be a random varaible that can take on any integer value and has pmf given by

$$h(w) = \left(\frac{1}{2}\right)^{w+1} \text{ for } w = 0, 1, 2, 3, \ldots$$

So, for example, $\mathrm{P}(W = 2) = h(2) = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$.

Probabilities can be obtained from the pmf by adding:

- $\mathrm{P}(X > 0) = 0.5 + 0.3 - 0.8$
- $\mathrm{P}(Y > 0) = 0.5 + 0.25$
- $\mathrm{P}(W > 0) = \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots = \frac{1}{2}$
- $\mathrm{P}(W > 0) = 1 - \mathrm{P}(W = 0) = 1 - \frac{1}{2} = \frac{1}{2}$

The only restrictions on a pmf are that

1. the values must all be non-negative, and
2. the sum (over all possible values of the random variable) must be 1.

That way the probabilities will behave the way probabilities should.

Example 4.1.1. Q. Amy is a 92% free throw shooter. We watch her take shots until she misses and let X be the number of shots. What is the pmf for X?

A. This time our table would be infinite (since there is no limit to how many consecutive free throws Amy might make), so we won't be able to write the whole table down. But we can work out the first few probabilities:

- $f(0) = \mathrm{P}(X = 0) = 0.08$. (She has to miss the first shot.)
- $f(1) = \mathrm{P}(X = 1) = (0.92)(0.08)$. (She has to make the first and miss the second.)
- $f(2) = \mathrm{P}(X = 2) = (0.92)^2(0.08)$. (She has to make the first two, then miss the third.)

At this point, we see there is a general pattern that allows us to write down an algebraic form for the pmf:

$$f(x) = \mathrm{P}(X = x) = (0.92)^{x-1}(0.08)$$

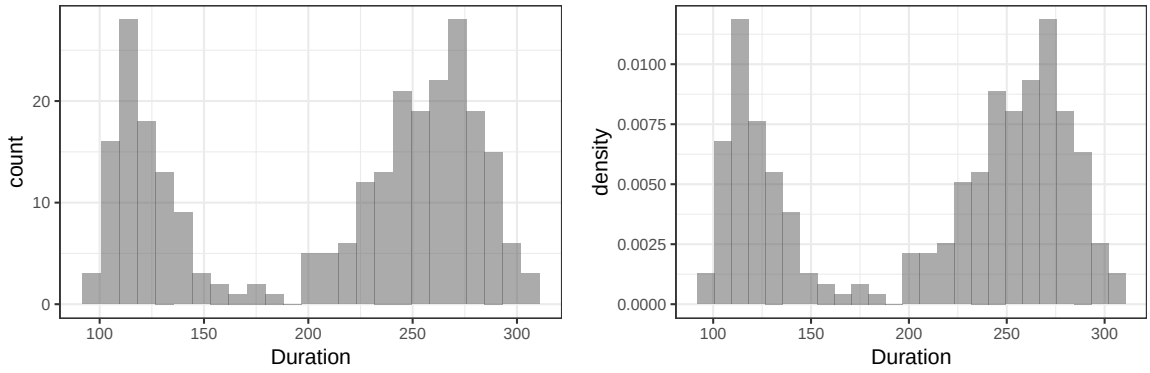
where x is a positive integer. (If x is not an integer, or $x < 1$, then $f(x) = 0$, since those values are not possible.)

4.2 Continuous Random Variables

4.2.1 Density histograms, density plots, density functions

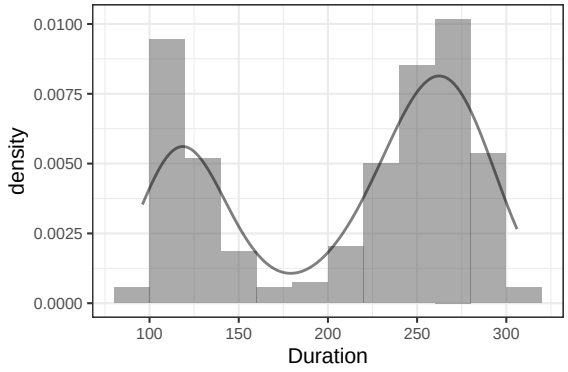
A histogram is a simple picture describing the "density" of data. Histogram bars are tall in regions where there is more data – i.e., where the data are more "dense".

```
library(alr4)
gf_histogram(  ~ Duration, data = oldfaith)
gf_dhistogram( ~ Duration, data = oldfaith)
```



The density scale is the same scale that is used by `gf_dens()` and `gf_density()`, and it is the default scale for histograms created using `gf_dhistogram()`.

```
library(alr4)
gf_dhistogram( ~ Duration, data = oldfaith, binwidth = 20, center = 110) %>%
gf_dens( ~ Duration, data = oldfaith)
```



The density scale is chosen so that the area of each rectangular bar (width times height) is equal to the proportion of the data set represented by the rectangle.

Example 4.2.1. Q. Use the histogram of Old Faithful eruption times to estimate the proportion of eruptions that last between 100 and 120 seconds.

A. In our histogram of Old Faithful eruption durations, the bar corresponding to the bin from 100–120 appears to have a height of about 0.09. That gives an area of 0.18 and indicates that approximately 18% of the eruptions last between 100 and 120 seconds.

```
tally( ~ ( 100 < Duration & Duration <= 120), data = oldfaith, format = "prop" )


## (100 < Duration & Duration <= 120)
##   TRUE  FALSE
## 0.1889 0.8111
```

The key idea behind the density scale can be expressed as

$$\text{Probability} = \text{area}$$

This association of area with probability means that the total area of all the bars will always be equal to 1 if we use the density scale.

It also provides us with a way to describe a distribution with a mathematical function.

Let f be a function such that
1. $f(x) \geq 0$ for all x,
2. $\int_{-\infty}^{\infty} f(x)\, dx = 1$.

Then f is called a **density function** (or probability density function, abbreviated pdf) and describes a continuous random variable X such that

$$\mathrm{P}(a \leq X \leq b) = \int_a^b f(x)\, dx\ .$$

Example 4.2.2. Let f be defined by

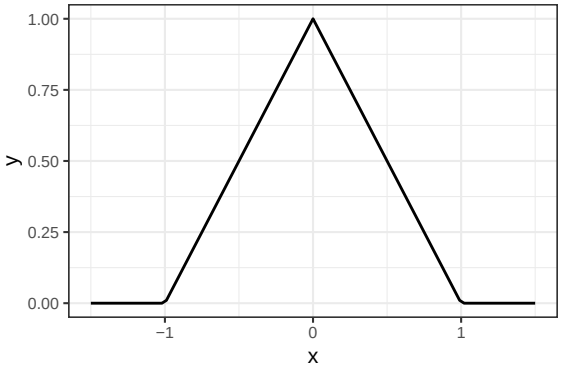
$$f(x) = \begin{cases} 1 - |x| & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

Show that f is a density function. Let X be the associated random variable, and compute the following probabilities:

1. $\mathrm{P}(X \leq 0)$
2. $\mathrm{P}(X \leq 1)$
3. $\mathrm{P}(X \leq \frac{1}{2})$
4. $\mathrm{P}(-\frac{1}{2}X \leq \frac{1}{2})$

A. While we could set up integrals for these, it is easier to solve them using geometry.[2]

```
f <- makeFun( (1 - abs(x)) * (abs(x) <= 1) ~ x )
gf_fun( f(x) ~ x, xlim = c(-1.5, 1.5) )
```



[2] R cleverly turns TRUE and FALSE into 1 and 0 when you use them in arithmetic expressions. The definition of `f()` makes use of this conversion to simplify specifying the cases.

The entire area under the curve can be found as the area of a triangle with base 2 and height 1.

$$\int_{-\infty}^{\infty} f(x)\,dx = \int_{-1}^{1} f(x)\,dx = \frac{1}{2} \cdot 2 \cdot 1 = 1$$

This implies that f is a density function.

1. $\mathrm{P}(X \le 1) = \int_{-\infty}^{1} f(x)\,dx = \int_{-1}^{1} f(x)\,dx = 1$
2. $\mathrm{P}(X \le \frac{1}{2}) = \int_{-\infty}^{1/2} f(x)\,dx = \int_{-1}^{1/2} f(x)\,dx = 1 - \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{7}{8}$
3. $\mathrm{P}(-\frac{1}{2} \le X \le \frac{1}{2}) = \int_{-1/2}^{1/2} f(x)\,dx = 1 - \frac{2}{8} = \frac{3}{4}$

We can also let R do (numerical) integration for us. There are two ways to do this. The first method uses the `integrate()` function.

```
integrate( f, -Inf, 1 )
```

```
## 1 with absolute error < 9.2e-05
```

```
# this will be more accurate since we aren't asking R to approximate
# something that we already know is exactly 0
integrate( f, -1, 1)
```

```
## 1 with absolute error < 1.1e-14
```

```
integrate( f, -.5, .5 )
```

```
## 0.75 with absolute error < 8.3e-15
```

```
# if you just want the value without the text saying how accurate the approximation is
# here are two equivalent ways to get it
value(integrate( f, -.5, .5 ))        # extract the value using val()
```

```
## [1] 0.75
```

```
integrate( f, -.5, .5 ) %>% value()   # %>% is the "then" operator
```

```
## [1] 0.75
```

An alternative approach uses `antiD()` from the `mosaic` package.

```
F <- antiD( f(x) ~ x)
F(1) - F(-1)        # total probability -- better be 1
```

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

```
F(.5) - F(-1)       # P( -1 <= X <= 0.5 )
```

```
## [1] 0.875
```

```
F(.5) - F(-.5)       # P( -.5 <= X <= .5 )
```

```
## [1] 0.75
```

4.2.2 Kernels

The **kernel** of a continuous random variable is a function that is a constant multiple of the pdf. The reason that these are interesting is that any kernel can be converted into a pdf by dividing by this constant. In particular, if

$$\int_{-\infty}^{\infty} k(x)\, dx = A\,,$$

then k is the kernel of a random variable with pdf

$$f(x) = \frac{k(x)}{A}\,.$$

Example 4.2.3. Q. The kernel of a random variable is given by

$$k(x) = x^2\ [\![x \in [0,2]]\!]\,.$$

Determine the pdf.

A. First we determine the value of the integral

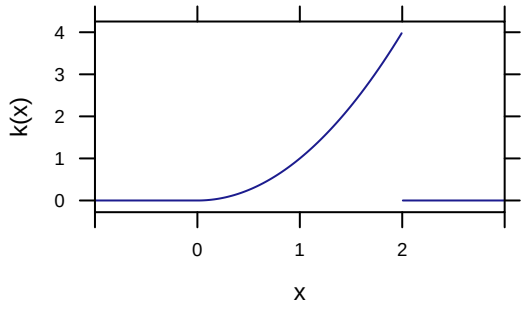
$$\int_{-\infty}^{\infty} k(x)\, dx\,.$$

```
k <- makeFun( x^2 * ( 0 <= x & x <= 2) ~ x )
plotFun(k(x) ~ x, xlim = c(-1,3))
integrate( k, 0, 2)
```

```
## 2.667 with absolute error < 3e-14
```

```
K <- antiD(k(x) ~ x, lower.bound = 0)
K(2)
```

```
## [1] 2.667
```



Since the total area is 8/3, if $\frac{k(x)}{8/3}$ is the pdf.

4.2.3 Cumulative distribution functions

If X is a random variable, then the **cumulative distribution function** (cdf) for X, often denoted F_X, is the function defined by

$$F_X(x) = \mathrm{P}(X \leq x)$$

That is, the output of the cdf reports the probability of being below a particular value.

For a continuous random variable, the cdf is a particular anti-derivative of the pdf. The derivative of the cdf is the pdf.

Example 4.2.4. Continuing with our previous example, if we choose -1 as our lower endpoint, then the anti-derivative will be the cdf.

```
f <- makeFun((1 - abs(x)) * (abs(x) <= 1) ~ x)
F <- antiD( f(x) ~ x, lower.bound = -1)    # We can use -1 instead of -Inf here.
F(-1)                # this should be 0 since we chose -1 as the lower bound.

## [1] 0

F(1)                 # P(X <= 1); should be 1

## [1] 1

F(.5)                # P(X <= 0.5)

## [1] 0.875

F(.5) - F(-.5)       # P( -0.5 <= X <= 0.5 )

## [1] 0.75
```

We have already seen that we can use a pdf f to calculate probabilities via integration, and that there is a special anti-derivative of f called the cdf such that the cdf F satisfies

$$F(x) = \mathrm{P}(X \leq x)$$

This function can also be used to compute probabilities, since

$$\mathrm{P}(a \leq X \leq b) = \int_a^b f(x)\, dx = F(b) - F(a)$$

Indeed, once we learn how to get the cdf function in R this will be our primary way to calculate probabilities in applications.

4.3 Mean and Variance

4.3.1 The mean of a random variable

The definition for the mean of a random variable will be motivated by the calculation of a mean of some data.

Example 4.3.1. Q. Suppose a student has taken 10 courses and received 5 A's, 4 B's, and 1 C. Using the traditional numerical scale where an A is worth 4, a B is worth 3, and a C is worth 2, what is this student's GPA (grade point average)?

A. The first thing to notice is that $\frac{4+3+2}{3} = 3$ is *not* correct. We cannot simply add up the values and divide by the number of values. Clearly this student should have a GPA that is higher than 3.0, since there were more A's than C's.

Consider now a correct way to do this calculation:

$$\begin{aligned}
\text{GPA} &= \frac{4+4+4+4+4+3+3+3+3+2}{10} \\
&= \frac{5\cdot 4 + 4\cdot 3 + 1\cdot 2}{10} \\
&= \frac{5}{10}\cdot 4 + \frac{4}{10}\cdot 3 + \frac{1}{10}\cdot 2 \\
&= 4\cdot\frac{5}{10} + 3\cdot\frac{4}{10} + 2\cdot\frac{1}{10} \\
&= 3.4\ .
\end{aligned}$$

The key idea here is that the mean is a **sum of values times probabilities**.

$$\text{mean} = \sum \text{value}\cdot\text{probability}$$

For a discrete random variable this translates to

$$\mathrm{E}(X) = \sum x f(x)$$

where the sum is taken over all possible values of X.

The mean of a random variable also goes by another name: **expected value**. We can denote the mean of X by either μ_X or $\mathrm{E}(X)$.

Example 4.3.2. Let X be discrete random variable with probablilities given in the table below.

value of X	0	1	2
probability	0.2	0.5	0.3

Q. What is the mean (expected value) of X?

A. $E(X) = 0\cdot 0.2 + 1\cdot 0.5 + 2\cdot 0.3 = 0.5 + 0.6 = 1.1$ This value reflects the fact that the random variable is larger than 1 a bit more often than it is less than 1.

Example 4.3.3. A local charity is holding a raffle. They are selling 1000 raffle tickets for \$5 each. The owners of five of the raffle tickets will win a prize. The five prizes are valued at \$25, \$50, \$100, \$1000, and \$2000. Let X be the value of the prize associated with a random raffle ticket (0 for non-winning tickets). Then:

- P(the ticket wins a prize) = $\mathrm{P}(X > 0) = 5/1000$.

- P(the ticket wins the grand prize) = $P(X = 2000) = 1/1000$.
- P(the ticket wins a prize worth more than \$75) = $P(X > 75) = 3/1000$.

The expected value of a ticket is

$$0 \cdot \frac{995}{1000} + 25 \cdot \frac{1}{1000} + 50 \cdot \frac{1}{1000} + 100 \cdot \frac{1}{1000} + 1000 \cdot \frac{1}{1000} + 2000 \cdot \frac{1}{1000}$$

```
25 * .001 + 50 * 0.001 + 100 * 0.001 + 1000 * 0.001 + 2000 * 0.001
```

```
## [1] 3.175
```

```
# R can help us set up this sum:
sum(c(25, 50, 100, 1000, 2000) * 0.001)
```

```
## [1] 3.175
```

When working with a continuous random variable, we replace the sum with an integral and replace the probabilities with our density function to get the following definition:

$$\mathrm{E}(X) = \mu_X = \int_{-\infty}^{\infty} x f(x)\, dx$$

If you recall doing center of mass problems you may recognize this integral as the first moment. (For pdfs, we don't need to divide by the "mass" because the total "mass" is the area under the curve, which will always be 1 for a random variable).

Note: It is possible that the integral used to define the mean will fail to converge. In that case, we say that the random variable has no mean or that the mean fails to exist.[3]

Example 4.3.4. Q. Compute the mean of our triangle distribution from Example 4.2.2.

A. We simply compute the integral from the definition.

$$\begin{aligned}
\mathrm{E}(X) &= \int_{-1}^{1} x f(x)\, dx \\
&= \int_{-1}^{0} x(x-1)\, dx + \int_{0}^{1} x(1-x)\, dx \\
&= \int_{-1}^{0} x^2 - x)\, dx + \int_{0}^{1} x - x^2)\, dx \\
&= \frac{x^3}{3} - \frac{x^2}{2}\Big|_{-1}^{0} + \frac{x^2}{2} - \frac{x^3}{3}\Big|_{0}^{1} \\
&= \frac{1}{3} - \frac{1}{2} + \frac{1}{2} - \frac{1}{3} = 0
\end{aligned}$$

This isn't surprising, by symmetry we would expect this result.

[3] Actually, we will require that $\int_{\infty}^{\infty} |x| f(x)\, dx$ converges. If this integral fails to converge, we will also say that the distribution has no mean.

We could also calculate this numerically in R:

```
f <- makeFun( (1 - abs(x)) * (abs(x) <= 1) ~ x)
xf <- makeFun( x * f(x) ~ x )
integrate(xf, -1, 1)
```

```
## 0 with absolute error < 3.7e-15
```

```
F <- antiD( x * f(x) ~ x, lower.bound = -1)
F(-1)  # should be 0
```

```
## [1] 0
```

```
F(1)
```

```
## [1] 0
```

4.3.2 Variance

Arguing similarly, we can compute the variance of a discrete or continuous random variable using

- discrete: $\text{Var}(X) = \sigma_X^2 = \sum_x (x - \mu_X)^2 f(x)\, dx$
- continuous: $\text{Var}(X) = \sigma_X^2 = \int_{-\infty}^{\infty} (x - \mu_X)^2 f(x)\, dx$

These can be combined into a single definition by writing

$$\text{Var}(X) = \text{E}((X - \mu_X)^2) \, .$$

Note: It is possible that the sum or integral used to define the mean (or the variance) will fail to converge. In that case, we say that the random variable has no mean (or variance) or that the mean (or variance) fails to exist.[4]

Example 4.3.5. Q. Compute the variance of the triangle random variable from the Example 4.2.2.

A.

```
f <- makeFun( (1 - abs(x)) * (abs(x) <= 1) ~ x)
xxf <- makeFun( (x-0)^2 * f(x) ~ x )
integrate(xxf, -1, 1)
```

```
## 0.1667 with absolute error < 1.9e-15
```

```
G <- antiD( (x-0)^2 * f(x) ~ x)
G(1) - G(-1)
```

```
## [1] 0.1667
```

[4] Actually, we will require that $\int_{\infty}^{\infty} |x| f(x)\, dx$ converges and $\int_{\infty}^{\infty} |x|^2 f(x)\, dx$ converges. If these integrals (or the corresponding sums for discrete random variables) fail to converge, we will say that the distribution has no mean (or variance).

Some simple algebraic manipulations of the sum or integral above shows that

$$\text{Var}(X) = \text{E}(X^2) - \text{E}(X)^2 \tag{4.1}$$

Example 4.3.6. Q. Compute the mean and variance of the random variable with pdf given by

$$g(x) = \frac{3x^2}{8}[\![x \in [0,2]]\!] .$$

This is the pdf computed in Example 4.2.3.

A.

```
g <- makeFun( (3 * x^2/8 ) * (0 <= x & x <= 2) ~ x )
m <- antiD( x * g(x) ~ x, lower.bound = 0)(2)  # all in one step instead of defining F or G
m
```

```
## [1] 1.5
```

```
v <- antiD( (x - m)^2 * g(x) ~ x, m = m, lower.bound = 0)(2)
v
```

```
## [1] 0.15
```

```
# here's the alternate computation
antiD( x^2 * g(x) ~ x, lower.bound = 0)(2) - m^2
```

```
## [1] 0.15
```

As with data, the standard deviation is the square root of the variance.

4.3.3 Quantiles

Quantiles solve equations of the form

$$\int_{-\infty}^{x} f(t)\, dt = F(x) = \text{P}(X \le x) = q$$

where q is known and x is unknown. So the 50th percentile (which is the 0.5-quantile or the median) is the number such that

$$\text{P}(X \le x) = 0.5 .$$

Example 4.3.7. Q. What is the 25th percentile of the triangle distribution in Example 4.2.2?

A. We need to solve for x in the following equation:

$$0.25 = \text{P}(X \le x) .$$

We can do this by working out the integral involved:

$$\begin{aligned} 0.25 &= \int_{-1}^{x} 1 - |t| \, dt \\ &= \int_{-1}^{x} 1 + t \, dt \\ &= t + t^2/2 \Big|_{-1}^{x} \\ &= x + x^2/2 + 1 - 1^2/2 \\ &= x + x^2/2 + 1/2 \\ 0 &= x^2/2 + x + 1/4 \\ 0 &= 2x^2 + 4x + 1 \end{aligned}$$

So by the quadratic formula, $x = \frac{1}{2}\sqrt{2} - 1 = -0.293$.

We can check this by evaluating the cdf.

```
x <- 1/2*sqrt(2) - 1
F(x)
```

```
## [1] 0
```

This could also be done geometrically by solving $\frac{1}{2}y^2 = \frac{1}{4}$ and letting $x = -1 + y$.

4.4 Some Important Families of Distributions

For now, we will consider only distributions of continuous random variables (probability density functions). We will leave set aside discrete random variables (probability mass function) until quite a bit later in the course.

A family of distributions is a collection of distributions that share some common features. Typically, these are described by giving a pdf that has one or more **parameters**. A parameter is simply a number that describes (a feature of) a distribution that distinguishes between members of the family. In this section we describe briefly some of the important distributions and how to work with them in R.

4.4.1 Triangle Distributions

The example distribution in the previous section is usually referred to as a triangle distribution (or triangular distribution) because of the shape of its pdf. There are, of course, many triangle distributions. A triangle distribution is specified with three numbers: a, the minimum; b, the maximum, and c, the location of the peak. A triangle distribution is symmetric if the peak is halfway between the minimum and maximum ($c = \frac{a+b}{2}$).

When X is a random variable with a triangle distribution, we will write $X \sim \mathsf{Triangle}(a, b, c)$. For many of the most common distributions, R has several functions that facilitate computation with those distributions. The triangle distributions are not in the base R distribution, but they can be added by requiring the `triangle` package.

For each distribution, there are four functions in R that always start with a single letter followed by a name for the distribution. In the case of the triangle distributions, these functions are

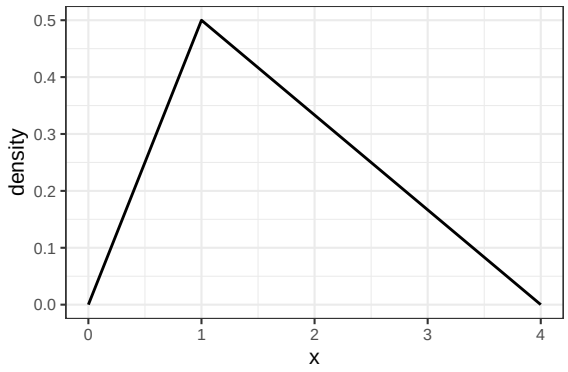
Function	What it does
`dtriangle(x,a,b,c)`	Computes value of the pdf at `x`
`ptriangle(q,a,b,c)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qtriangle(p,a,b,c)`	Computes quantiles, that is a value q so that $P(X \leq q) = p$
`rtriangle(n,a,b,c)`	Randomly samples `n` values from the Triangle(a, b, c) distribution

Example 4.4.1. Q. Let $X \sim \text{Triangle}(0, 4, 1)$. Use R to answer the following questions.

1. Plot the pdf for X.
2. What is $P(X \leq 1)$?
3. What is $P(X \leq 2)$?
4. What is the median of X?
5. What is the mean of X?

A. The `gf_dist()` function in the `ggformula` package allows us to graph the pdf for any function R knows how to work with in the standard way. For example, here is a plot of the pdf of a Triangle$(0, 4, 1)$-distribution.

```
library(triangle)      # a package that knows about triangle distributions
gf_dist("triangle", a = 0, b = 4, c = 1)
```



Here is the R code to answer the remaining questions.

```
ptriangle(1, 0, 4, 1)   # P(X <= 4); notice that his is NOT 1/2

## [1] 0.25

ptriangle(2, 0, 4, 1)   # P(X <= 4); also NOT 1/2

## [1] 0.6667

qtriangle(0.5, 0, 4, 1) # median is the 0.5-quantile

## [1] 1.551
```

```
T <- antiD( x * dtriangle(x, 0,4,1) ~ x, lower.bound = 0)
T(4)                    # mean of X

## [1] 1.667

integrate( makeFun( x * dtriangle(x, 0,4,1) ~ x) , 0, 4)

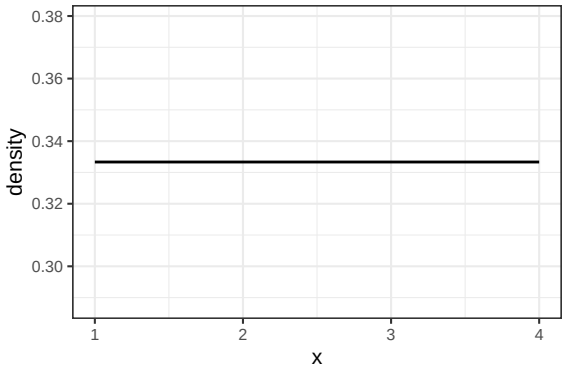
## 1.667 with absolute error < 1.9e-14
```

4.4.2 Uniform Distributions

A uniform distribution is a described by a constant function over some interval. Its shape is a rectangle. This makes it particularly easy to calculate probabilities for a uniform distribution. Despite its simplicity, the family of uniform distributions has many applications.

We will let $X \sim \text{Unif}(a, b)$ denote that X is a uniform random variable on the interval from a to b. In R the parameters a and b are given more meaningful names: `min` and `max`. We can use the following code to graph the $\text{Unif}(1, 4)$ distribution.

```
gf_dist("unif", min = 1, max = 4, xlim = c(-1, 6))
```



Notice that the width of the non-zero portion of the pdf is 3, so the height must be 1/3.

Probabilities involving uniform distributions are easily calculated using simple geometry, but R also provides several functions for working with uniform probability distributions.

Function	What it does
`dunif(x,min,max)`	Computes value of the pdf at `x`
`punif(x,min,max)`	Computes value of the cdf at `x`, i.e., $\text{P}(X \leq \texttt{x})$
`qunif(p,min,max)`	Computes quantiles, that is a value of x so that $\text{P}(X \leq x) = \texttt{q}$
`runif(n,min,max)`	Randomly samples `n` values from the $\text{Unif}(\texttt{min}, \texttt{max})$ distribution

Notice the pattern to these names. They start with the same letters as the functions for the triangle distributions, but replace `triangle` with `unif`. *There are similar functions for all of the distributions in this chapter.*

Example 4.4.2. Q. Let $X \sim \text{Unif}(1, 4)$. Use R to calculate the following values and check the values using geometry:

1. $P(X \leq 2)$
2. the 80th percentile of the distribution

A.

```
punif(2,1,4)   # P(X <= 2 )
```

```
## [1] 0.3333
```

```
(2-1) * 1/3    # P(X <= 2 ) using area
```

```
## [1] 0.3333
```

```
qunif(.8, 1,4) # 80th percentile
```

```
## [1] 3.4
```

We could also get the 80th percentile by solving the equation $\frac{1}{3}(x-1) = 0.8$ From this we get $\frac{x}{3} = 0.8 + 1/3$, so $x = 3(0.8 + 1/3) = 2.4 + 1 = 3.4$.

4.4.3 Exponential Distributions

The exponential distributions are useful for modeling the time until some "event" occurs. The model is based on the assumptions that

1. The probability of an event occurring in any small interval of time is proportional to the length of the time interval. The constant of proportionality is the rate parameter, usually denoted by λ.
2. The probabilities of events occurring in two small non-overlapping intervals are independent.

Examples 4.4.3. Here are some situations that might be well modeled by an exponential distribution:

1. The time until the next radioactive decay event is detected on a Geiger counter
2. The time until a space satellite is struck by a meteor (or some other space junk) and disabled.

   The model would be good if (over some time span of interest) the chances of getting struck are always the same. It would not be such a good model if the satellite moves through time periods of relatively higher and then relatively lower chances of being struck (perhaps because we pass through regions of more or less space debris at different times of the year.)
3. The lifetime of some manufactured device.

   This is a pretty simple model (we'll learn better ones later) and most often is *too* simple to describe the interesting features of the lifetime of a device. In this model, failure is due to some external thing "happening to" the device; the device itself does not wear (or improve) over time.

We will let $X \sim \mathsf{Exp}(\lambda)$ denote that X has an exponential distribution with rate parameter λ. The kernel of such a distribution is

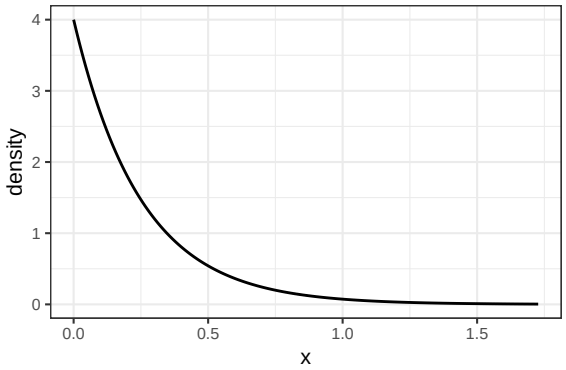
$$k(x;\lambda) = e^{-\lambda x}\, [\![x \geq 0]\!]$$

Notice that the function describing this distribution is defined only for x-values that are real numbers greater than or equal to zero (in mathematical notation, the interval $[0, \infty)$.) This interval is sometimes called the "support" of the distribution. When using probability distributions to model data, it's important to think about whether the support of the distribution matches well with the range of possible values observed in the data.

The exponential distribution function is a pretty easy function to integrate, but R provides the now familiar functions to make things even easier.

Function	What it does
`dexp(x,rate)`	Computes value of the pdf at `x`
`pexp(q,rate)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qexp(p,rate)`	Computes quantiles, that is a value q so that $P(X \leq q) = p$
`rexp(n,rate)`	Randomly samples `n` values from the $\mathsf{Exp}(\lambda)$ distribution

```
gf_dist("exp", rate = 4)
```



4.4.4 Gamma and Weibull Distributions

The Gamma and Weibull families of distributions are generalizations of the exponential distribution. Each family has two parameters, a rate parameter as in the exponential distribution, and an additional parameter called the shape parameter (denoted by α below). The reciprocal of the rate parameter is called the scale parameter. For the Gamma distribution, R lets us use either rate or scale (and the default is rate). For the Weibull, we must use the scale.

distribution	kernel
$\mathsf{Gamma}(\alpha, \lambda)$	$k(x) = x^{\alpha-1} e^{-\lambda x} \, [\![x \geq 0]\!]$
$\mathsf{Weibull}(\alpha, \lambda)$	$k(x) = x^{\alpha-1} e^{-\lambda x^{\alpha}} \, [\![x \geq 0]\!]$

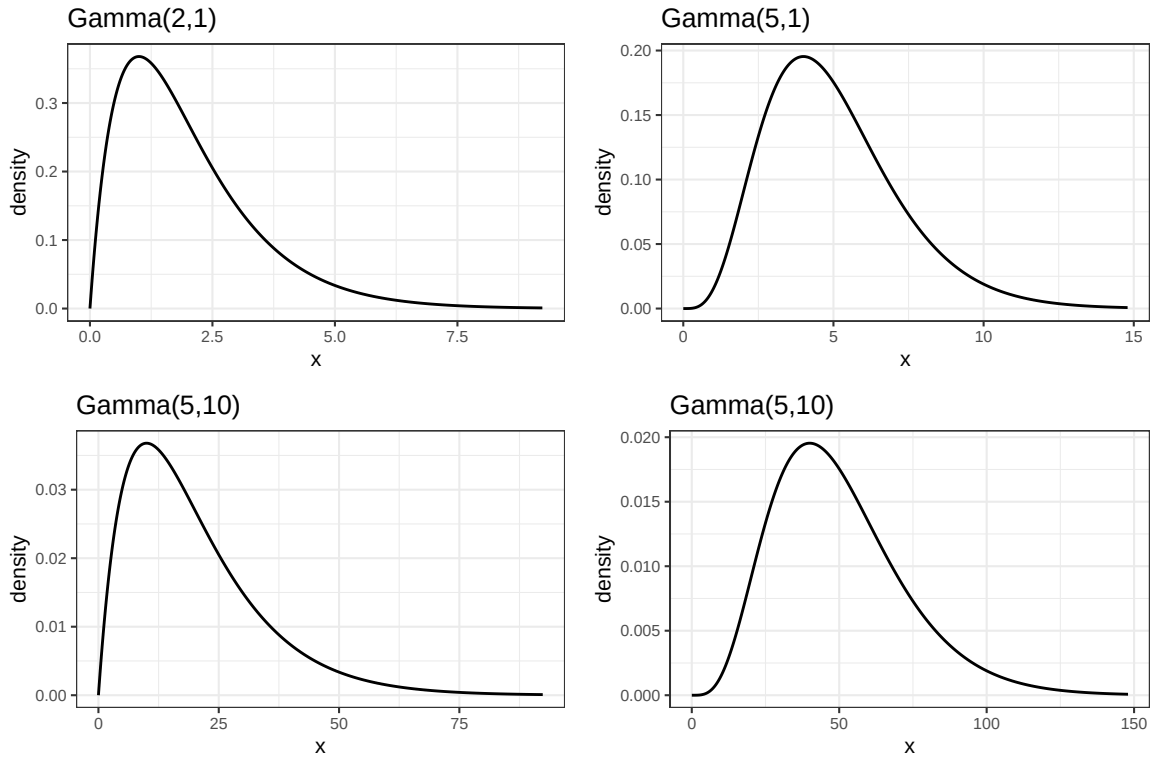
Both families of distributions are supported on the interval $[0, \infty)$.) For the most part, we won't use these formulas in calculations, preferring to let R do the work for us. However, notice that each of these distributions has a pdf that allows for relatively simple integration. For the Gamma distributions, we need to use integration by parts ($\alpha - 1$ times). For the Weibull distributions we can use a substitution: $u = x^{\alpha}$. In each case, when $\alpha = 1$ we get an exponential distribution.

The now familiar functions are available for each of these distributions.

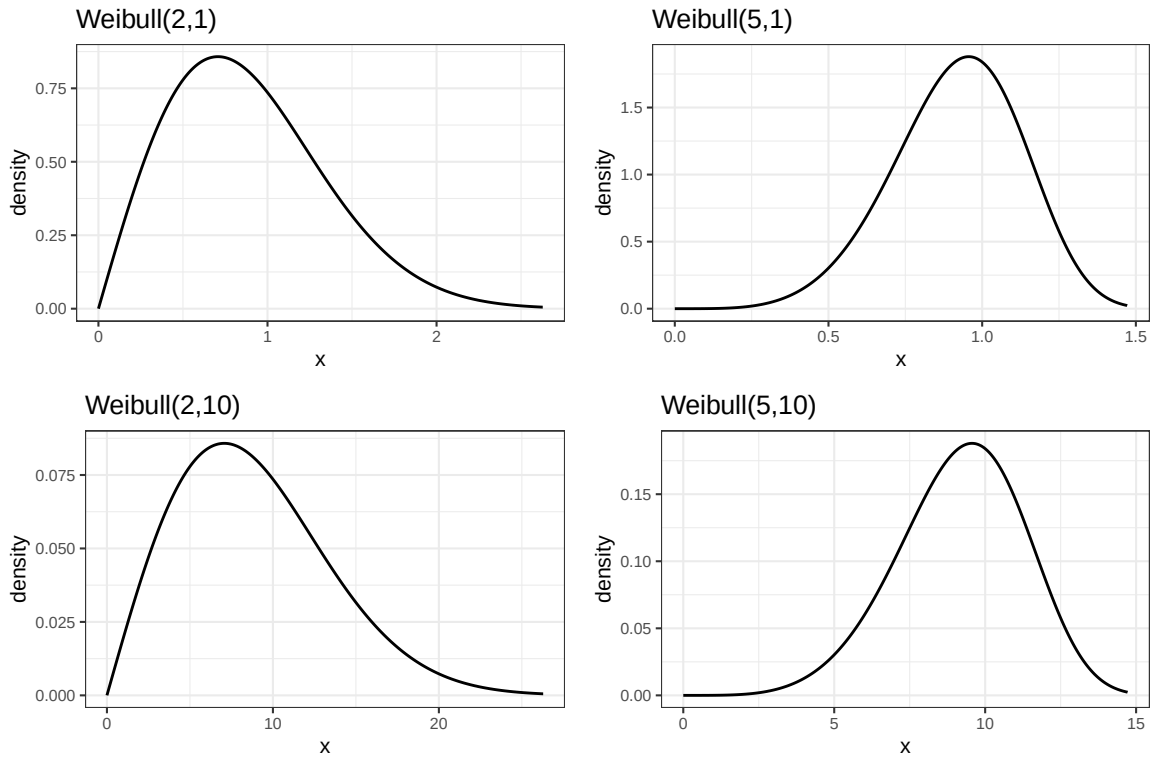
Function	What it does
`dgamma(x, shape, rate, scale = 1/rate)`	Computes value of the pdf at `x`
`pgamma(q, shape, rate, scale = 1/rate)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qgamma(p, shape, rate, scale = 1/rate)`	Computes quantiles, that is a value q so that $P(X \leq q) = p$
`rgamma(n, shape, rate, scale = 1/rate)`	Randomly samples `n` values from a Gamma distribution.
`dweibull(x, shape, scale = 1/rate)`	Computes value of the pdf at `x`
`pweibull(q, shape, scale)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qweibull(p, shape, scale)`	Computes quantiles, that is a value q so that $P(X \leq q) = p$
`rweibull(n,shape,scale)`	Randomly samples `n` values from a Weibull distribution.

Like the exponential distributions, these distributions are skewed and only take on positive values. These distributions arise in many applications, including as more general models for lifetime. As the pictures below indicate, the shape and scale parameters are aptly named.

```
gf_dist("gamma", params = list(shape = 2, rate = 1), title = "Gamma(2,1)")
gf_dist("gamma", params = list(shape = 5, rate = 1), title = "Gamma(5,1)")
gf_dist("gamma", params = list(shape = 2, scale = 10), title = "Gamma(5,10)")
gf_dist("gamma", params = list(shape = 5, scale = 10), title = "Gamma(5,10)")
```



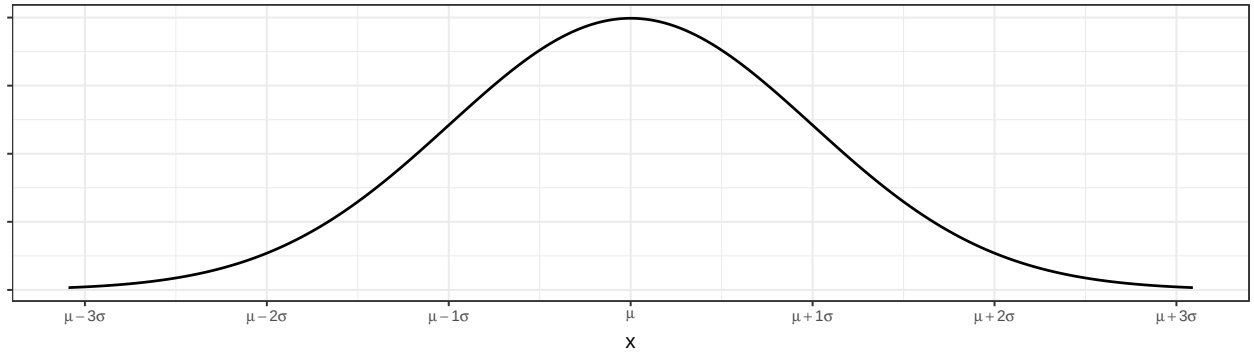

```
gf_dist("weibull", params = list(shape = 2, scale = 1),title = "Weibull(2,1)")
gf_dist("weibull", params = list(shape = 5, scale = 1),title = "Weibull(5,1)")
gf_dist("weibull", params = list(shape = 2, scale = 10),title = "Weibull(2,10)")
gf_dist("weibull", params = list(shape = 5, scale = 10),title = "Weibull(5,10)")
```



4.4.5 Normal Distributions

We come now to the most famous family of distributions – the normal distributions (also called Gaussian distributions). These symmetric distributions have the famous "bell shape" and are described by two parameters, the mean μ and the standard deviation σ. The pdf for a $\mathsf{Norm}(\mu, \sigma)$ distribution is

distribution	pdf
$\mathsf{Norm}(\mu, \sigma)$	$f(x) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$



The inflection points of the normal distributions are always at $\mu - \sigma$ and $\mu + \sigma$.

Among the normal distributions is one special distribution – the **standard normal distribution** – which has mean 0 and standard deviation 1. All other normal distributions are simply linear transformations of the

standard normal distribution. That is, If $Z \sim \text{Norm}(0,1)$ and $Y = a + bX$, then $Y \sim \text{Norm}(a,b)$. Conversely, if $Y \sim \text{Norm}(\mu,\sigma)$, then $Z = \frac{Y-\mu}{\sigma} \sim \text{Norm}(0,1)$.

As with the other distributions we have encountered, we have four functions that allow us to work with normal distributions in R:

Function	What it does
`dnorm(x,mean,sd)`	Computes value of the pdf at `x`
`pnorm(q,meand,sd)`	Computes value of the cdf at `x`, i.e., $\text{P}(X \leq \text{q})$
`qnorm(p,mean,sd)`	Computes quantiles, that is a value q so that $\text{P}(X \leq \text{q}) = \text{p}$
`rnorm(n,mean,sd)`	Randomly samples `n` values from a normal distribution.

The 68-95-99.7 Rule

Also known as the "Empirical Rule", the 68-95-99.7 Rule provides a set of probability benchmarks for the normal distributions because for any normal distribution:

- $\approx 68\%$ of the normal distribution is between $\mu - \sigma$ and $\mu + \sigma$.
- $\approx 95\%$ of the normal distribution is between $\mu - 2\sigma$ and $\mu + 2\sigma$.
- $\approx 99.7\%$ of the normal distribution is between $\mu - 3\sigma$ and $\mu + 3\sigma$.

Example 4.4.4. Q. Before they were rescaled, SAT scores used to be approximately normally distributed with a mean of 500 and a standard deviation of 100.

1. Approximately what percent of test takers scored between 400 and 600?
2. Approximately what percent of test takers scored above 600?
3. Approximately what percent of test takers scored below 300?
4. Approximately what percent of test takers scored between 400 and 700?

A.

1. 68%
2. Since 68% are bewteen 400 and 600, the other 32% must be outside that range, half above and half below. So 16% are above 600.
3. Since 95% are between 300 and 700, the other 5% must be outside that range, half above and half below. So 2.5% are below 300.
4. 16% are below 400 and 2.5% are above 700, so the remaining 81.5% must be between 400 and 700.

Of course, we can get more accurate results using R:

```
pnorm( 600, 500, 100) - pnorm(400, 500, 100)
```

```
## [1] 0.6827
```

```
pnorm( 700, 500, 100) - pnorm(300, 500, 100)
```

```
## [1] 0.9545
```

```
pnorm( 300, 500, 100)
```

```
## [1] 0.02275
```

```
pnorm( 700, 500, 100) - pnorm(400, 500, 100)
```

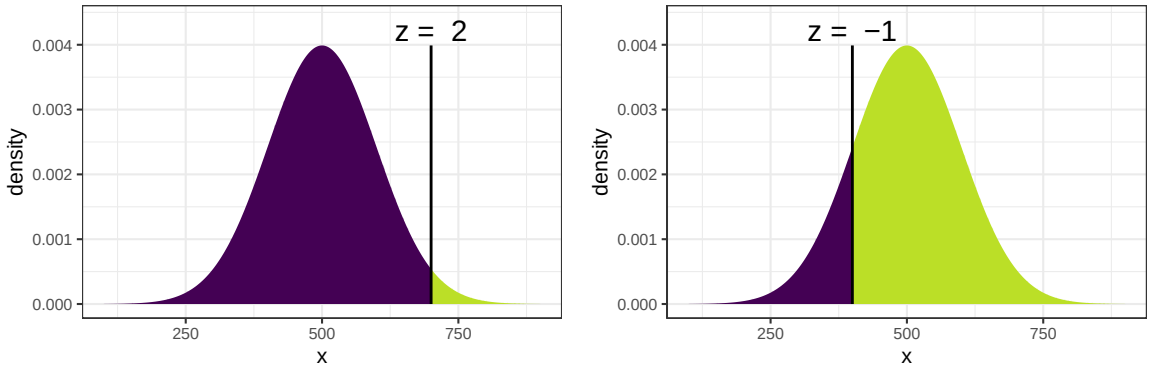
```
## [1] 0.8186
```

The `xpnorm()` function will additionally draw pictures of the normal distribution with a portion of the distribution shaded in.

```
xpnorm(700,500,100) - xpnorm(400, 500, 100)
```

```
##
## If X ~ N(500, 100), then
## P(X <= 700) = P(Z <= 2) = 0.9772
## P(X >  700) = P(Z >  2) = 0.02275
##
## If X ~ N(500, 100), then
## P(X <= 400) = P(Z <= -1) = 0.1587
## P(X >  400) = P(Z >  -1) = 0.8413
##
```

```
## [1] 0.8186
```



Example 4.4.5. We can use `qnorm()` to compute percentiles. For example, let's calculate the 75th percentile for SAT distributions.

```
qnorm(.75, 500, 100)

## [1] 567.4
```

4.4.6 Beta Distributions

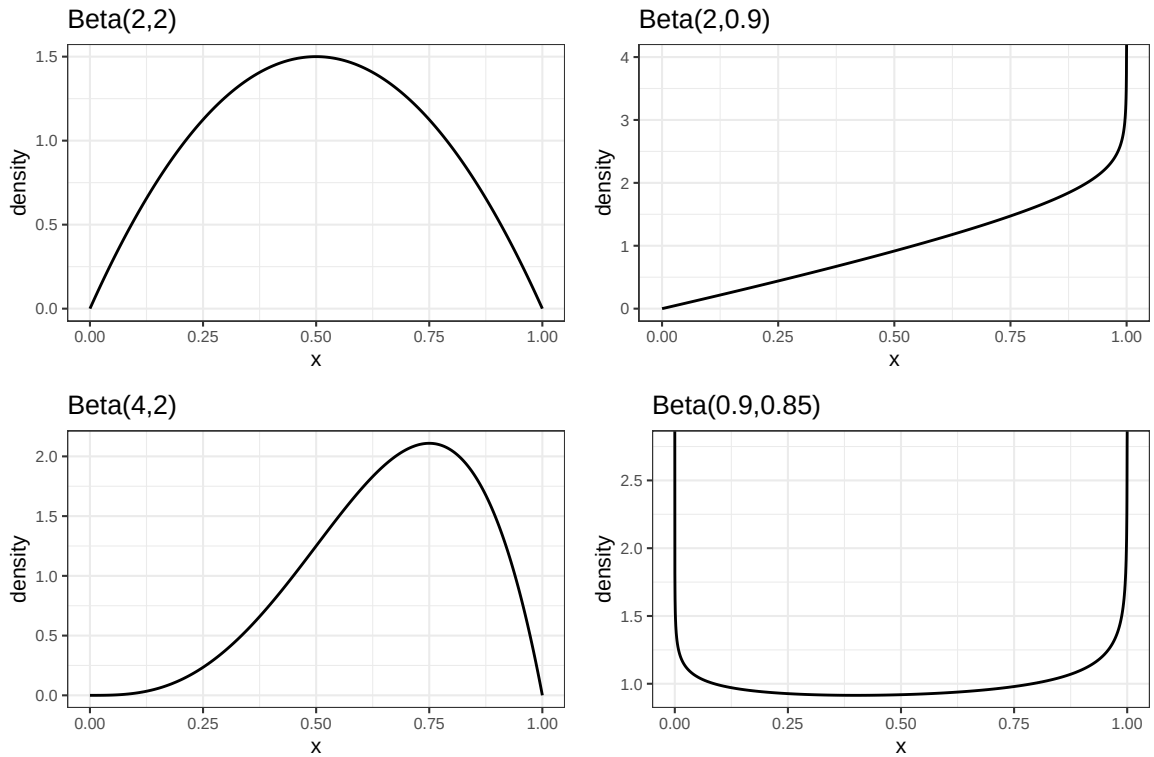
The Beta distributions have support on the interval $(0, 1)$, so they can provide a model for proportions or other quantities that are bounded between 0 and 1.[5] The Beta distributions have two parameters, imaginatively called `shape1` and `shape2`. The kernel of the Beta distributions is a product of a power of x and a power of $(1 - x)$:

$$k(x; \alpha, \beta) = x^{\alpha-1}(1-x)^{\beta-1} \; [\![x \in [0,1]]\!]$$

When $\alpha = \beta$, the distribution is symmetric, and when $\alpha = \beta = 1$, we have the $\mathsf{Unif}(0, 1)$-distribution.

The two shape parameters provide a wide variety of shapes.

```
gf_dist("beta", params = list(shape1 = 2, shape2 = 2), title = "Beta(2,2)")
gf_dist("beta", params = list(shape1 = 2, shape2 = 0.9), title = "Beta(2,0.9)")
gf_dist("beta", params = list(shape1 = 4, shape2 = 2), title = "Beta(4,2)")
gf_dist("beta", params = list(shape1 = 0.9, shape2 = 0.85), title = "Beta(0.9,0.85)")
```



[5] A more general version of the Beta distributions can do the same thing for quantities bounded by any two numbers. This more general family of distributions has four parameters.

Function	What it does
`dbeta(x,shape1,shape2)`	Computes value of the pdf at `x`
`pbeta(q,shape1d,shape2)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qbeta(p,shape1,shape2)`	Computes quantiles, that is a value q so that $P(X \leq q) = p$
`rbeta(n,shape1,shape2)`	Randomly samples `n` values from a Beta distribution.

4.4.7 Binomial Distributions

A binomial distribution is a discrete distribution with two parameters (n and p, or as R calls them `size` and `prob`) describing a situation in which

1. Our random process consists of n identical sub-processes (called trials).
2. Each trial has one of two outcomes (traditionally called success and failure).
3. The probability of success is p for each trial.
4. The outcome of each trial is independent of the others.

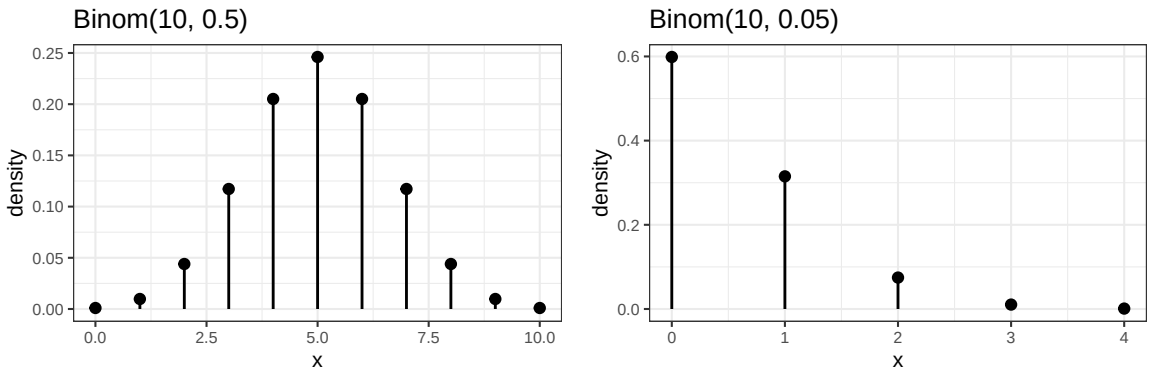
The binomial random varialbe counts the number of successes.

Example 4.4.6.

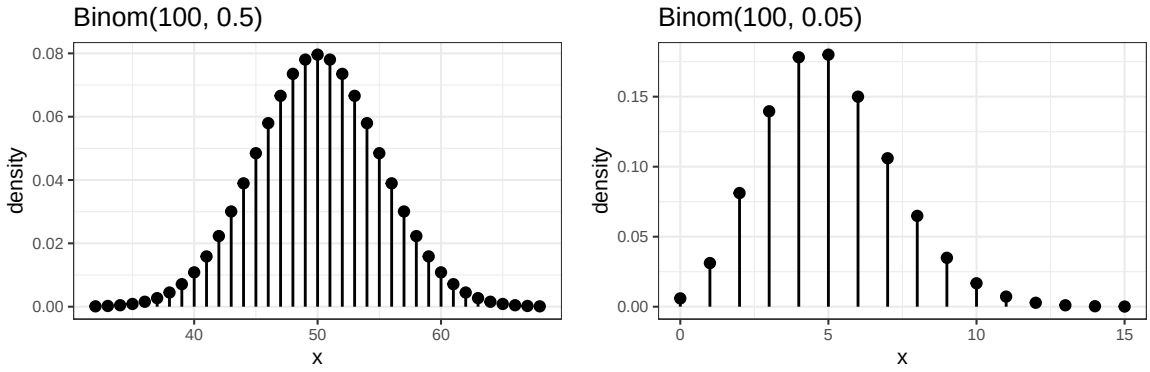
1. If we flip a coin 100 times and let X be the number of heads, then $X \sim \text{Binom}(100, 0.5)$.
2. Amy is a 92% free throw shooter. If she attempts 50 free throws and we let Y be the number that she makes, then $Y \sim \text{Binom}(50, 0.92)$ (assuming that each shot is independent of the others.[6])

Here are some example binomial distributions. The distributions are symmetric when $p = 0.5$. For a fixed size n, the distributions become more and more skewed as p gets closer to 0 or 1. For a fixed probability p, the distributions become more and more symmetric as n gets larger.

```
gf_dist("binom", size = 10, prob = 0.5, title = "Binom(10, 0.5)")
gf_dist("binom", size = 10, prob = 0.05, title = "Binom(10, 0.05)")
gf_dist("binom", size = 100, prob = 0.5, title = "Binom(100, 0.5)")
gf_dist("binom", size = 100, prob = 0.05, title = "Binom(100, 0.05)")
```



[6] Whether an individual shooter's shots are independent or exhibit longer runs of "hot" and "cold" streaks that we would expect under indpendence has been investigated by many people. The general conclusion seems to be that the independence assumption matches reality pretty closely.



The pmf for a binomial distribution is given by

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

where $\binom{n}{x} = \frac{n!}{x!(n-x)!}$ is the binomial coefficient. As with the continuous distributions, we have our usual functions available.

Function	What it does
`dbinom(x, size, prob)`	Computes value of the pmf at `x`
`pbinom(q, size, prob)`	Computes value of the cdf at `x`, i.e., $\mathrm{P}(X \leq \mathtt{q})$
`qbinom(p, size, prob)`	Computes quantiles, that is the smallest value of x so that $\mathrm{P}(X \leq \mathtt{x}) \geq \mathtt{p}$
`rbinom(n, size, prob)`	Randomly samples `n` values from a Binomial distribution.

4.4.8 Poisson Distributions

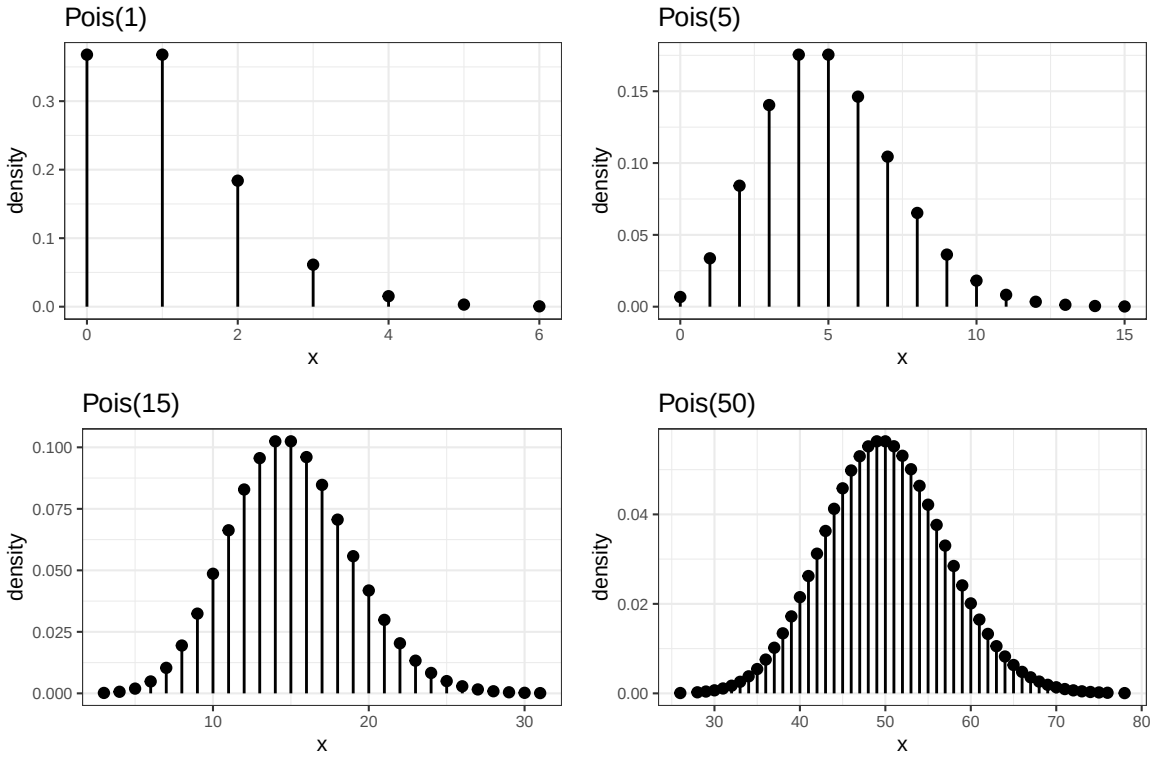
The Poisson distributions are generally used as models for counting "events" that happen in a specified amount of time or space. If the probability of an event happening at any moment is the same and independent of events happening at other moments, then the count of events in a fixed amount of time will be a Poisson random variable. The Poisson family has one parameter – often denoted λ and called the *rate parameter* – which is the average number of events that happen over the fixed amount of time or space we are observing.

Example 4.4.7.

1. Let X be the number of clicks of a Gieger counter in a 1 second interval. Since each click corresponds to a radioactive decay event which we generally assume occur "at random" but according to some average rate, a Poisson random variable would be a good model for this. The rate parameter would be the average number of decay events per second.
2. If you stand on along a busy highway and count the number of red cars that go by in 30 minutes, a Poisson random variable might be a good model. Sometimes you will get bunches of red cars or periods of time with few or no red cars, that is just as a Poisson model predicts.

The Poisson distributions are skewed right, but become less and less skewed as the rate parameter increases. Here are a few example plots.

```
gf_dist("pois", lambda = 1, title = "Pois(1)")
gf_dist("pois", lambda = 5, title = "Pois(5)")
gf_dist("pois", lambda = 15, title = "Pois(15)")
gf_dist("pois", lambda = 50, title = "Pois(50)")
```



The pmf for a Pois(λ) random variable is

$$f(x) = \frac{e^{-\lambda}\lambda^x}{x!}$$

and the functions in R for working with Poisson distributions are the following:

Function	What it does
`dpois(x, lambda)`	Computes value of the pmf at `x`
`ppois(q, lambda)`	Computes value of the cdf at `x`, i.e., $P(X \leq q)$
`qpois(p, lambda)`	Computes quantiles, that is the smallest value of x so that $P(X \leq x) \geq p$
`rpois(n, lambda)`	Randomly samples `n` values from a Poisson distribution.

4.5 Fitting Distributions to Data

Suppose we think a family of distributions would make a good model for some situation. How do we decide which member of the family to use? The simple answer is that we should choose the one that fits "best." The trick is deciding what it means to fit well. In fact there is more than one way to measure how well a distribution fits a data set.

Example 4.5.1. We can use the following code to load a data set that contains three year's worth of mean hourly wind speeds (mph) in Twin Falls, ID. This kind of data is often used to estimate how much power could be generated from a windmill placed in a given location.

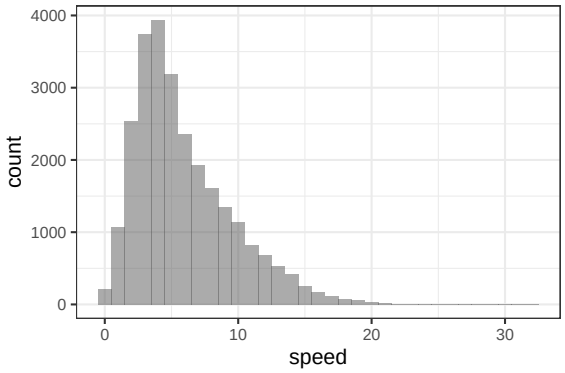
```
Wind <-
  read.csv("https://rpruim.github.io/Engineering-Statistics/data/stob/TwinfallsWind.csv")
head(Wind, 2)
```

```
##       date time speed
## 1 1/1/2010 0:00  2.24
## 2 1/1/2010 1:00  2.42
```

```
tail(Wind, 2)
```

```
##             date  time speed
## 26272 12/31/2012 22:00  3.88
## 26273 12/31/2012 23:00  5.04
```

```
gf_histogram( ~ speed, data = Wind, binwidth = 1 )
```



As we can see, the distribution is skewed, but it doesn't look like an exponential distribution would be a good fit. Of the distributions we have seen, it seems like a Weibull or Gamma distribution would be a potentially good choice. A Weibull model has often been used as a model for mean hourly wind speed, and the shape of our histogram indicates that this is a reasonable family of distributions.

Q. Which Weibull distribution is the best model for our data?

A. The `fitdistr()` in the `MASS` package uses the method of **maximum likelihood** to fit univariate (one variable) distributions.

```
fitdistr( Wind$speed, "weibull" )
```

```
## Error in fitdistr(Wind$speed, "weibull"):  Weibull values must be > 0
```

For `fitdistr()` to fit a Weibull distribution, all of the data must be positive, but our data includes some 0's.

```
tally( ~ (speed == 0), data = Wind)
```

```
## (speed == 0)
##  TRUE FALSE
##    48 26225
```

Let's see how small the smallest non-zero measurements are.

```
min( ~ speed, data = Wind %>% filter(speed > 0))


## [1] 0.01
```

This may well be a simple rounding issue, since the wind speeds are recorded to the nearest 0.01 and 0.01 is the smallest positive value. Let's create a new variable that moves each value of 0 to 0.0025 and try again. Why 0.0025? If we think that 0.01 represents anything in the range 0.005 to 0.015, which would round to 0.01, then 0 represents anything in the range 0 to 0.005. 0.0025 is the middle of that range.

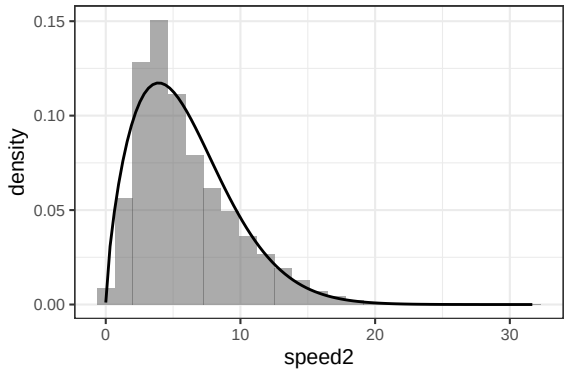
```
Wind <- Wind %>% mutate(speed2 = ifelse( speed > 0, speed, 0.0025))
fitdistr( Wind$speed2, "weibull" )


##     shape      scale
##   1.694423   6.650587
##  (0.007958) (0.025552)
```

This says that the best fitting (in the sense of maximum likelihood) Weibull distribution is the Weibull$(1.69, 6.65)$-distribution.

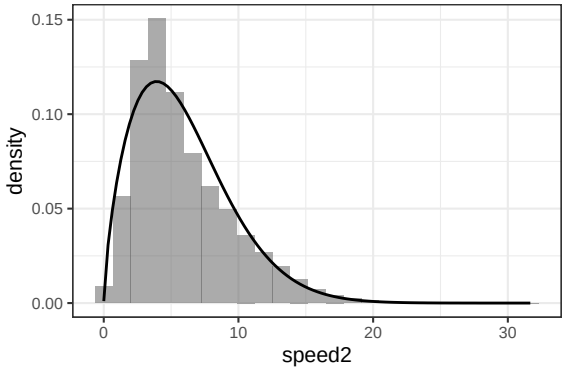
The `gf_histogram()` function has an option to overlay the distribution fit by `fitdistr()` so we can see how good the fit is graphically.

```
gf_dhistogram( ~ speed2, data = Wind) %>%
  gf_fitdistr( ~ speed2, data = Wind, dist = "weibull")
```



This can be abbreviated a bit:

```
gf_dhistogram( ~ speed2, data = Wind) %>%
  gf_fitdistr(dist = "weibull")
```



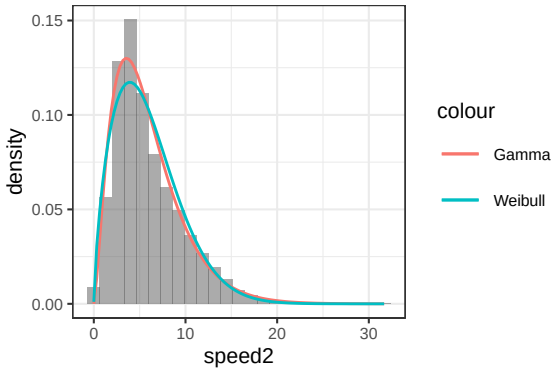
`gf_fitdistr()` is inheriting the formula and data from `gf_dhistogram()`.

Example 4.5.2. As an alternative, we could fit a Gamma distribution to the wind speed data.

```
fitdistr(Wind$speed2, "gamma")


##      shape       rate
##   2.495583   0.421178
##  (0.020486) (0.003829)


gf_dhistogram( ~ speed2, data = Wind) %>%
  gf_fitdistr(dist = "gamma" , color = ~ "Gamma") %>%
  gf_fitdistr(dist = "weibull" , color = ~ "Weibull")
```



By eye, it appears that the Gamma distribution fits this data set slightly better, but there may other reasons to prefer the Weibull distribution. In fact, there has been a good deal of research done regarding which distributions to use for wind speed data fitting. The answer to the question of which distributions should be used seems to be that it depends on the purpose for your modeling: "The fact that different distributions excel under different applications motivates further research on model selection based upon the engineering parameter of interest." [MLVB11]

Example 4.5.3. 1986–87 was a good season for Michael Jordan, a famous former NBA basketball player. Possible models for the points scored each game that season are normal, Weibull, and Gamma distributions. The normal distributions might be a good choice if we think that the distributions is roughly symetric (very good games are about the same amount above average as the very poor games are below average). Weibull and Gamma distributions have the built in feature that scores cannot be negative and would allow for a skewed distribution. The `fitdistr()` function in the `MASS` package can fit each of these.

```
library(fastR2)      # the Jordan8687 data set is in this package
fitdistr(Jordan8687$points, "normal")
```

```
##     mean       sd
##   37.0854    9.8640
##  ( 1.0893) ( 0.7702)
```

```
fitdistr(Jordan8687$points, "weibull")
```

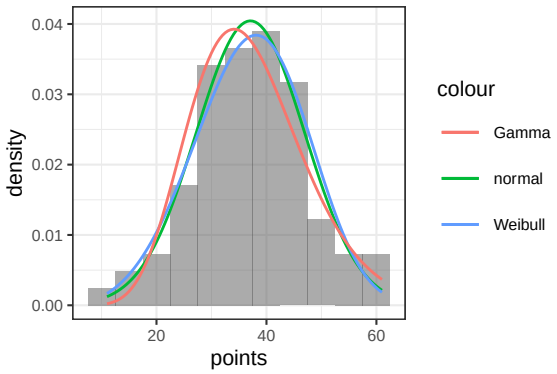
```
##     shape     scale
##    4.1228   40.7746
##  ( 0.3455) ( 1.1517)
```

```
fitdistr(Jordan8687$points, "gamma")
```

```
##     shape     rate
##   12.4284    0.3351
##  ( 1.9154) ( 0.0527)
```

We can use a histogram with overlaid density curve to see how well these fits compare to the data.

```
gf_dhistogram(~ points, data = Jordan8687, binwidth = 5) %>%
  gf_fitdistr(dist = "dnorm", color = ~"normal") %>%
  gf_fitdistr(dist = "dweibull", color = ~ "Weibull") %>%
  gf_fitdistr(dist = "dgamma", color = ~ "Gamma")
```



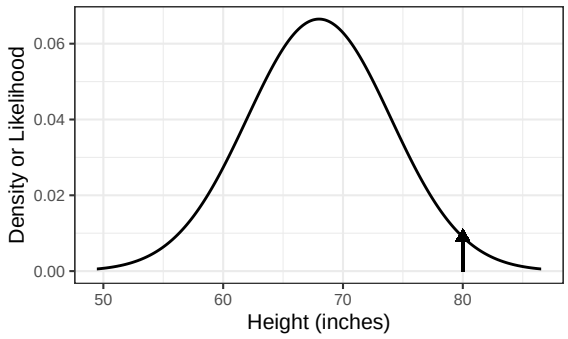
The three fits are similar, but not identical.

4.5.1 Maximum Likelihood

The `fitdistr()` function uses the maximum likelihood method to estimate distribution parameters. The maximum likelihood method is one of the most commonly used estimation methods in all of statistics because (1) it can be used in a wide range of applications, and (2) the resulting estimators have some some desirable properties. Maximum likelihood estimation tries to choose the parameter values that *maximize* the *likelihood* of the observed data.

First, let's think about the "likelihood" of an individual observed data-point. The likelihood of the data-point is just the probability density function (or probability mass function) for the distribution of interest, evaluated at

the value observed in the data. The likelihood gives some indication of how frequently we'd expect to observe this value, but it is *not* a probability (for one thing, likelihoods can exceed 1). The figure below illustrates that the likelihood of observing a person 80 inches (6 feet, 8 inches) tall, if the person comes from a population whose heights are Normally distributed with a mean of 68 inches and a standard deviation of 6 inches is about 0.009:



Given a set of specific parameter values, the likelihood of an entire observed data-set can be calculated by obtaining the value of the likelihood of each observed data-point, and summing these over all the observed data points. Then, we can find the maximum likelihood parameter estimates by trying many candidate parameter values until satisfied that we have found the ones that maximize the likelihood. (The numerical methods used are usually a bit more sophisticated than "guessing lots of random candidate values", but we won't get into the details here. In some cases, it is also possible to write down a mathematical expression for the likelihood of the data given the parameters, and maximize it analytically.)

We'll illustrate the main ideas of maximum likelihood with a simple example.

Example 4.5.4. Michael has three dice in his pocket. One is a standard die with six sides, another has four sides, and the third has ten sides. He challenges you to a game. Without showing you which die he is using, Michael is going to roll a die 10 times and report to you how many times the resulting number is a 1 or a 2. Your challenge is to guess which die he is using.

Q. Michael reports that 3 of the 10 rolls resulted in a 1 or a 2. Which die do you think he was using?

A. The probability of obtaining a 1 or a 2 is one of $\frac{1}{2}$, $\frac{1}{3}$, or $\frac{1}{5}$, depending on which die is being used. Our data are possible with any of the three dice, but let's see how likely they are in each case.

- If P(roll 1 or 2) $= \frac{1}{5}$, then the probability of obtaining exactly Michael's data is

$$\left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^7 = 0.06 \, .$$

  (Whatever the order, there will be 3 events with probability 1/5 and 7 with probability 4/5. Since the events are independent, we can multiply all of these probabilities.)

- If P(roll 1 or 2) $= \frac{1}{3}$, then the probability of obtaining exactly Michael's data is

$$\left(\frac{1}{3}\right)^3 \left(\frac{2}{3}\right)^7 = 0.002 \, .$$

- If P(roll 1 or 2) $= \frac{1}{2}$, then the probability of obtaining exactly Michael's data is

$$\left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^7 = 0.002 \, .$$

Of these, the largest likelihood is for the case that P(roll 1 or 2) $= \frac{1}{3}$, i.e., for the standard, six-sided die. Our data would be more likely to occur with that die than with either of the other two – it is the maximum likelihood die.

In general, maximum likelihood calculations are harder because instead of having only 3 choices, there will be infinitely many choices, and instead of having only one parameter, there may be multiple parameters. So techniques from (multi-variable) calculus or numerical approximation methods are often used to maximize the likelihood function. The `fitdistr()` function uses pre-derived formulas for some distributions and numerical approximation methods for others. In some cases, you will get warning messages about attempts to apply a function to values that don't make sense (trying to take logs or square roots of negative numbers, zero in the denominator, etc.) as the numerical approximation algorithm explores options in an attempt to find the best fit. The help documenation for `fitdistr()` explains which distributions it can handle and what method is used for each.

4.5.2 The method of moments

An easy (but sometimes fairly crude) way to estimate the parameters of a distribution is the method of moments. You will often see this method used in engineering textbooks, espeically if they do not rely on software that implements others methods (like the maximum likelihood method).

The basic idea is to set up a system of equations where we set the mean of the data equal to the mean of the distribution, the variance of the data equal to the variance of the distribution, etc.[7]

To employ this method, we need to know the means and variances of our favorite families of distributions (in terms of the parameters of the distributions). For all of the distributions we have seen, one can work out formulas for the means and variances in terms of the parameters involved. These are listed in Table 4.1

Example 4.5.5. Let's return to the wind speeds in Example 4.5.1. The formulas for the mean and variance of a Weibull distribution involve the gamma function $\Gamma()$, which might be unfamiliar to you. So let's simplify things.

Theoretical properties and observations of wind speeds at other locations suggest that using a shape parameter of $\alpha = 2$ is often a good choice (but shape does differ from location to location depending on how consistent or variable the wind speeds are). The Weibull distributions with $\alpha = 2$ have a special name, they are called the **Rayleigh** distributions. So $\mathsf{Rayleigh}(\beta) = \mathsf{Weibull}(\alpha = 2, \beta)$. In this case, from Table 4.1, we see that to calculate the mean we need the value of $\Gamma(1 + \frac{1}{2}) = \Gamma(1.5) = \sqrt{\pi}/2$.

```
gamma(1.5)
```

```
## [1] 0.8862
```

```
sqrt(pi)/2
```

```
## [1] 0.8862
```

From Table 4.1 we see that the mean of a $\mathsf{Rayleigh}(\beta)$-distribution is

$$\mathrm{E}(X) = \beta \frac{\sqrt{\pi}}{2}$$

Now we can choose our estimate $\hat{\beta}$ for β so that

$$\hat{\beta} \frac{\sqrt{\pi}}{2} = \overline{x};.$$

[7] If our distribution has more than 2 parameters, we will need higher moments, which we will not cover here.

That is,

$$\hat{\beta} = \frac{2\overline{x}}{\sqrt{\pi}}$$

```
x.bar <- mean(~speed, data = Wind)
x.bar
```

```
## [1] 5.925
```

```
beta.hat <- x.bar * 2 / sqrt(pi)
beta.hat
```

```
## [1] 6.686
```

So our method of moments fit for the data is a Rayleigh(6.69) = Weibull$(2, 6.69)$

Although the Rayleigh distributions are not as flexible as the Weibull or Gamma distributions, and although maximum likelihood is generally preferred over the method of moments, the method of moments fit of a Rayliegh distribution does have one advantage: it can be computed even if all you know is the mean of some sample data. Sometimes, that is all you can easily get your hands on (because the people who collected the raw data only report numerical summaries). You can find average wind speeds of for many locations online, for example here: `http://www.wrcc.dri.edu/htmlfiles/westwind.final.html`

Example 4.5.6. For distributions with two parameters, we solve a system of two equations with two unknowns. For the normal distributions this is particularly easy since the parameters are the mean and standard deviation, so we get

$$\hat{\mu} = \overline{x}$$
$$\hat{\sigma}^2 = s_x^2$$

```
x.bar <- mean(~speed, data = Wind); x.bar
```

```
## [1] 5.925
```

```
v <- var(~speed, data = Wind); v
```

```
## [1] 13.35
```

```
sqrt(v)
```

```
## [1] 3.653
```

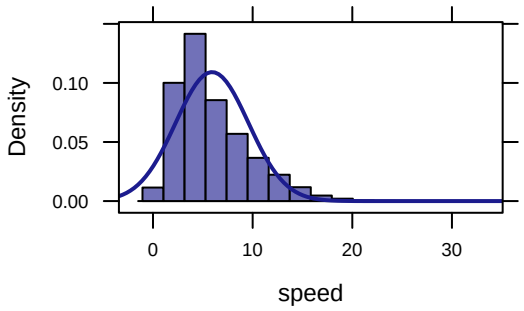
So the method of moments suggests a Norm$(5.93, 3.65)$ distribution. In this case, the method of moments and maximum likelihood methods give the same results.

```
fitdistr(Wind$speed, "normal")


##     mean       sd
##   5.92524   3.65320
##  (0.02254) (0.01594)
```

But this doesn't mean that the fit is particularly good. Indeed, a normal distribution is not a good choice for this data. We know that wind speeds can't be negative and we have other distributions (exponential, Weibull, and Gamma, for example) that are also never negative. So choosing one of those seems like a better idea. The following plot shows, as we expected, that the normal distribution is not a particularly good fit.

```
histogram(~speed, data = Wind, fit = "normal")
```



It is important to remember that the best fit using a poor choice for the family of distriubtions might not be a useful fit. The choice of distributions is made based on a combination of theoretical considerations, experience from previous data sets, and the quality of the fit for the data set at hand.

4.6 Quantile-Quantile Plots

To this point we have looked at how well a distribution fits the data by overlaying a density curve on a histogram. While this is instructive, it is not the easiest way to make a graphical comparison between a data set and a theoretical distribution. Our eyes are much better and judging whether something is linear than they are at judging whether shapes have a particular kind of curve. Furthermore, certain optical misperceptions tend to cause people to exaggerate some kinds of differences and underestimate others.

Quantile-quantile plots offer an alternative approach. As the name suggests, the idea is to compare the quantiles of our data to the quantiles of a theoretical distribution. These are then plotted as a scatter plot. Let's go through those steps with a small data set so we can see all the moving parts, then we'll learn how to automate the whole process using `gf_qq()`.

4.6.1 Normal-Quantile Plots

The normal distributions are especially important for statistics, so normal-quantile plots will be our most important example of quantile-quantiles plots. Also, special properties of the normal distributions make normal-quantile plots especially easy and useful. We will illustrate the construction of these plots using a data set containing Michael Jordan's game by game scoring output from the 1986–87 basketball season.

Example 4.6.1. Let's begin by forming a randomly selected sample of 10 basketball games.

distribution pdf or pmf	mean	variance
Triangle: $\mathsf{Triangle}(a,b,c)$ $\begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{if } x \in [a,c] \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{if } x \in [c,b] \\ 0 & \text{otherwise} \end{cases}$	$\frac{a+b+c}{3}$	$\frac{a^2+b^2+c^2-ab-ac-bc}{18}$
Uniform: $\mathsf{Unif}(a,b)$ $\begin{cases} \frac{1}{b-a} & \text{if } x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$	$\frac{b+a}{2}$	$\frac{(b-a)^2}{12}$
Standard normal: $\mathsf{Norm}(0,1)$ $\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$	0	1
Normal: $\mathsf{Norm}(\mu,\sigma)$ $\frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$	μ	σ^2
Exponential: $\mathsf{Exp}(\lambda)$ $\lambda e^{-\lambda x}$	$1/\lambda$	$1/\lambda^2$
Gamma: $\mathsf{Gamma}(\alpha, \lambda = \frac{1}{\beta})$ $\frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$	$\alpha/\lambda = \alpha\beta$	$\alpha/\lambda^2 = \alpha\beta^2$
Weibull: $\mathsf{Weibull}(\alpha, \beta = \frac{1}{\lambda})$ $\frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$	$\beta\Gamma(1+\frac{1}{\alpha})$	$\beta^2 \left[\Gamma(1+\frac{2}{\alpha}) - \left[\Gamma(1+\frac{1}{\alpha})\right]^2\right]$
Beta: $\mathsf{Beta}(\alpha,\beta)$ $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Binomial: $\mathsf{Binom}(n,p)$ $\binom{n}{x} p^x (1-p)^{n-x}$	np	$np(1-p)$
Poisson: $\mathsf{Pois}(\lambda)$ $\frac{e^{-\lambda}\lambda^x}{x!}$	λ	λ

Table 4.1: Some common continuous distributions. Standard names for parameters that appear in several distributions include `rate` (λ), `shape` (α), and `scale` (β). In the normal distributions, μ and σ are called `mean` and `sd` in R, and in the uniform distirbutions, a and b are called `min` and `max`. The function $\Gamma(x)$ that appears in the formulas for the Weibull and Beta distributions is a kind of continuous extrapolation from the factorial function. The `gamma()` function will calculate these values.

```r
set.seed(123)              # so you can get the same sample if you like.
SmallJordan <- sample(Jordan8687, 10)
SmallJordan
```

```
##    game points orig.id
## 31   31     27      31
## 79   79     53      79
## 51   51     43      51
## 14   14     40      14
## 67   67     40      67
## 42   42     49      42
## 50   50     33      50
## 43   43     38      43
## 81   81     61      81
## 25   25     43      25
```

```r
probs <- seq(0.05, 0.95, by = 0.10)
probs
```

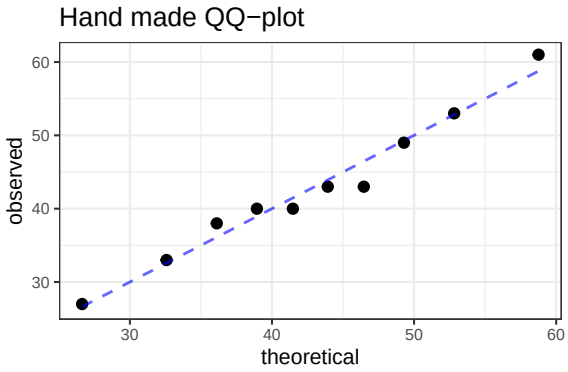
```
##  [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
```

```r
observed <- sort(SmallJordan$points)                                    # sorted observations
theoretical <- qnorm( probs, mean = mean(observed), sd = sd(observed) ) # theoretical quantiles

QQData <- data.frame(observed = observed, theoretical = theoretical)
QQData
```

```
##    observed theoretical
## 1        27       26.64
## 2        33       32.58
## 3        38       36.11
## 4        40       38.94
## 5        40       41.47
## 6        43       43.93
## 7        43       46.46
## 8        49       49.29
## 9        53       52.82
## 10       61       58.76
```

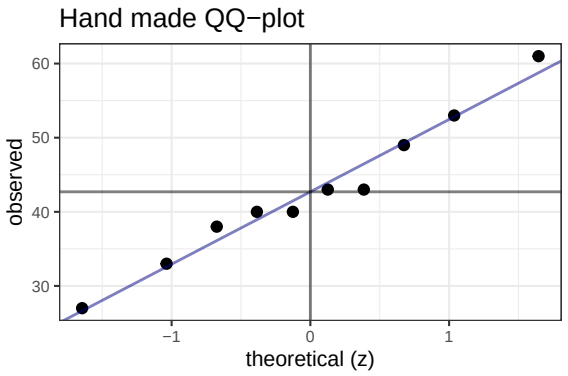
If the observed data matched the theoretical quantiles perfectly, a scatter plot would place all the points on the line with slope 1 passing through the origin.

```r
gf_point( observed ~ theoretical, data = QQData, title = "Hand made QQ-plot" ) %>%
  gf_fun( x ~ x, alpha = 0.6, color = "blue", linetype = "dashed")
```



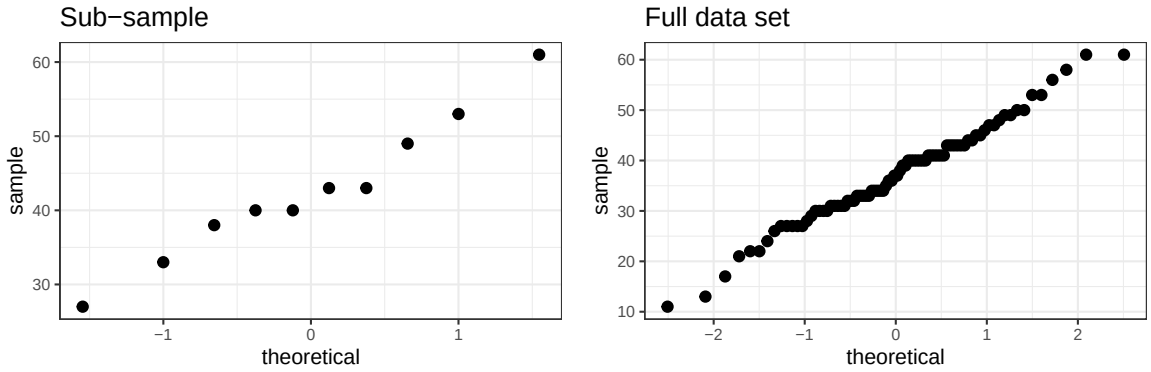
Even better, we don't need to know the mean and standard deviation in advance, because all normal distributions are linear transformations of the Norm$(0,1)$-distribution. So our standard practice will be to compare our data to the Norm$(0,1)$-distribution. If $X \sim \text{Norm}(\mu, \sigma)$, then $X = \mu + \sigma Z$ where $Z \sim \text{Norm}(0,1)$, so a plot of X vs. Z will have slope σ and intercept μ.

```
theoretical2 <- qnorm( probs, mean = 0, sd = 1 ) # theoretical quantiles from Norm(0,1)
QQData2 <- data.frame(observed = observed, theoretical = theoretical2)
gf_point(observed ~ theoretical, data = QQData2, title = "Hand made QQ-plot", xlab = "theoretical (z)" )
  gf_abline(intercept = ~ mean(SmallJordan$points), slope = ~ sd(SmallJordan$points),
            alpha = 0.5, color = "navy", data = NA) %>%
  gf_hline(yintercept = ~ mean(SmallJordan$points), alpha = 0.5) %>%
  gf_vline(xintercept = ~ 0, alpha = 0.5)
```



This whole process is automated by the `gf_qq()` function.

```
gf_qq( ~ points, data = SmallJordan, title = "Sub-sample" )
gf_qq( ~ points, data = Jordan8687, title = "Full data set" )
```

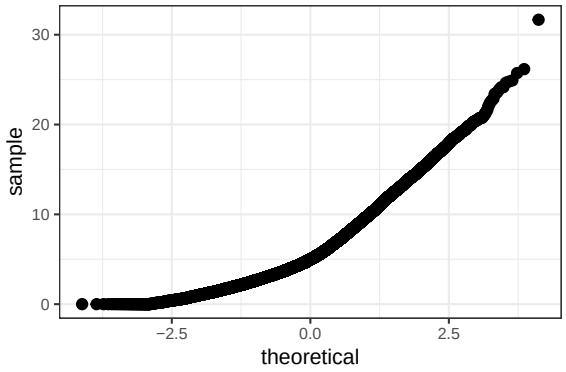


4.6.2 Other distributions

Working with other distributions is similar, but most families of distributions don't have a single "master example" to which we can make all comparisons, so we need to pick a particular member of the family (either by fitting or for some theoretical reason).[8]

Example 4.6.2. Let's build a quantile-quantile plot for our wind speed data comparing to normal, gamma and Weibull distributions. We can automate this, but we need to tell `gf_qq()` how to calculate the quantiles.

```
gf_qq( ~ speed2, data = Wind)  # normal-quantile plot; normal is not a good model
```

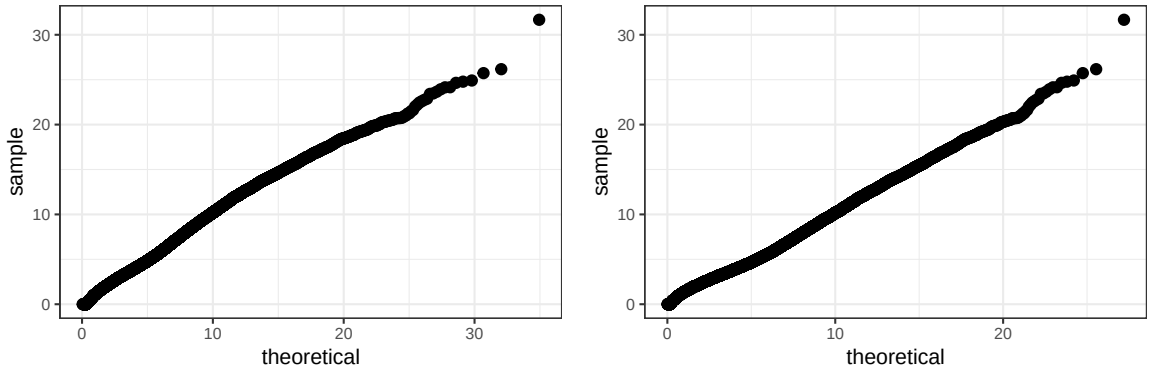


The normal model does not fit well, but both Gamma and Weibull are reasonable models:

```
fitdistr(Wind$speed2, "gamma")

##     shape       rate
##   2.495583   0.421178
##  (0.020486) (0.003829)

fitdistr(Wind$speed2, "Weibull")

##     shape      scale
##   1.694423   6.650587
##  (0.007958) (0.025552)
```

[8] There are a few other families of distributions that have a prototypical member such that all other members are a linear transformation of the prototype. The exponential family is one such family.

```
fittedqgamma <- makeFun( qgamma(p, shape = 2.496, rate = 0.421 ) ~ p )
fittedqweibull <- makeFun( qweibull(p, shape = 1.694, scale = 6.651) ~ p )
gf_qq( ~speed2, data = Wind, distribution = fittedqgamma )
gf_qq( ~speed2, data = Wind, distribution = fittedqweibull )
```



4.7 Exercises

4.1 Let $f(x) = 5/4 - x^3$ on $[0, 1]$.

a) Show that f is a pdf.

b) Calculate $\mathrm{P}(X \leq \frac{1}{2})$.

c) Calculate $\mathrm{P}(X \geq \frac{1}{2})$.

d) Calculate $\mathrm{P}(X = \frac{1}{2})$.

4.2 Repeat parts (2) – (4) of Example 4.4.1 using geometry rather than R.

4.3 Let $k(x) = (1 - x^2) \cdot [\![x \in [-1, 1]]\!] = \begin{cases} 1 - x^2 & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$ be the kernel of a continuous distribution.

a) Determine the pdf for this distribution.

b) Compute the mean and variance for this distribution

4.4 Let $Y \sim \mathsf{Triangle}(0, 10, 4)$. Compute $\mathrm{E}(Y)$ and the median of Y.

4.5 Let $W \sim \mathsf{Unif}(0, 10)$. Compute $\mathrm{E}(W)$ and $\mathrm{Var}(W)$.

4.6

a) Let $X \sim \mathsf{Exp}(4)$. Use R to compute $\mathrm{E}(X)$.

b) Let $X \sim \mathsf{Exp}(10)$. Use R to compute $\mathrm{E}(X)$.

c) Let $X \sim \mathsf{Exp}(1/5)$. Use R to compute $\mathrm{E}(X)$.

d) What pattern do you notice. Explain in terms of the definition of the exponential distribution why this makes sense.

4.7 Use R to plot the pdf and compute the mean and variance of each of the following distributions.

a) $\mathsf{Beta}(2, 3)$

b) $\mathsf{Beta}(20, 30)$

c) $\mathsf{Gamma}(\texttt{shape} = 2, \texttt{scale} = 3)$

d) Weibull(`shape` $= 2$, `scale` $= 3$)

4.8 For each of the following distributions, determine the proportion of the distribution that lies between 0.5 and 1.

a) Exp(`rate` $= 2$)

b) Beta(`shape1` $= 3$, `shape2` $= 2$)

c) Norm(`mean` $= 1$, `sd = 2`)

d) Weibull(`shape` $= 2$, `scale = 1/2`)

e) Gamma(`shape` $= 2$, `scale = 1/2`)

4.9

a) Using Table 4.1 and the method of moments, fit an exponential distribution to the Twin Falls wind speed data.

```
Wind <-
  read.csv("https://rpruim.github.io/Engineering-Statistics/data/stob/TwinfallsWind.csv")
```

What is the estimated value of the rate parameter?

b) Now use `fitdistr()` to fit an exponential distribution using maximum likelihood.

c) How do the two estimates for the rate parameter compare?

d) How well does an exponential distribution fit this data?

4.10 A Gamma distribution can also be fit using the method of moments. Because there are two parameters (shape and rate or shape and scale), you will need to solve a system of two equations with two unknowns.

a) Using Table 4.1 and the method of moments, fit a Gamma distribution to the Twin Falls wind speed data. What are the estimated values of the shape and rate parameters?

b) How do the method of moments estimates for the parameters compare to the maximum likelihood estimates from `fitdistr()`?

4.11 Sam has found some information about wind speed at a location he is interested in online. Unfortunately, the web site only provides the mean and standard deviation of wind speed.

mean:	10.2 mph
standard deviation:	5.1 mph

a) Use this information and the method of moments to estimate the shape and rate parameters of a Gamma distribution.

b) In principal, we could do the same for a Weibull distribution, but the formulas aren't as easy to work with. Fit a Rayliegh distribution instead (i.e., a Weibull distribution with shape parameter equal to 2).

4.12 In 1964, a study was undertaken to see if IQ at 3 years of age is associated with amount of crying at newborn age. In the study, 38 newborns were made to cry after being tapped on the foot, and the number of distinct cry vocalizations within 20 seconds was counted. The subjects were followed up at 3 years of age and their IQs were measured. You can load this data using

```
Baby <- read.csv("https://rpruim.github.io/Engineering-Statistics/data/BabyCryIQ.csv")
head(Baby)
```

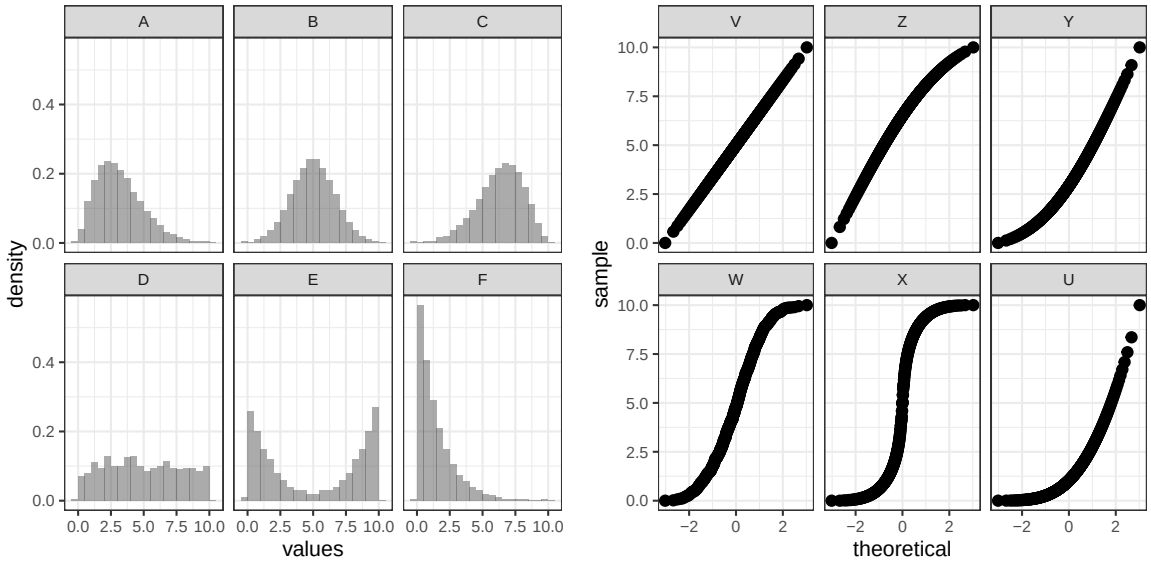
```
##   cry.count  IQ
## 1        10  87
## 2        20  90
## 3        17  94
## 4        12  94
## 5        12  97
## 6        15 100
```

The `cry.count` variable records the number of distinct cry vocalizations within 20 seconds. Choose a family of distributions to fit to this data and do the fit using `fitdistr()`. Also include a plot showing a histogram and your fitted density curve.

4.13 Create normal quantile plots for the ages of patients in the `HELPrct` data set separated by `substance`. (Getting separate or overlaid plots using `gf_qq()` works just like it does for other `ggformula` plots).

Comment on the plots.

4.14 Match the normal-quantile plots to the histograms.



4.15 Show that $\text{Var}(X) = \text{E}(X^2) - \text{E}(X)^2$ by showing that

$$\int_{-\infty}^{\infty} (x - \mu_X)^2 f(x)\, dx = \int_{-\infty}^{\infty} x^2 f(x)\, dx - \mu_X^2$$

whenever f is a pdf and all the integrals involved converge.

4.16 The heights of 18–22 year olds in the US follow approximately normal distributions within each sex. Estimated means and standard deviations appear in the table below.

	mean	standard deviation
women	64.3 in	2.6 in
men	70 in	2.8 in

Answer the following questions without using a computer or calculator (except for basic arithmetic).

a) If a woman is 68 inches tall, what is her z-score?

b) If a man is 74 inches tall, what is his z-score?

c) What is more unusual, a woman who is at least 68 inches tall or a man who is at least 74 inches tall?

d) Big Joe has decided to open a club for tall people. To join his club, you must be in the tallest 2.5% of people of your sex. How tall must a woman be to join Big Joe's club?

e) How tall must a man be to join Big Joe's club?

4.17 Use the information from the previous problem to answer the following questions.

a) What proportion of women are 5'10" or taller?

b) What proportion of men are 6'4" or taller?

c) If a man is in the 75th percentile for height, how tall is he?

d) If a woman is in the 30th percentile for height, how tall is she?

5 Transformation and Combinations of Random Variables

We will often be interested in random variables that are formed by transformations or combinations other random variables.

- If we roll to dice and let X and Y be the results on each die, then the sum is $X + Y$.
- If R is the radius of a circle, then $A = \pi R^2$ is its area.
- If X and Y are the length and width of a rectangle, then the area is given by $A = XY$.
- If F is a temperature measured in degrees Fahrenheit, then $C = \frac{5}{9}(F - 32) = \frac{5}{9}F - \frac{160}{9}$ is the temperature in degrees Celsius. (Most other unit conversions are even simpler linear functions.)
- If W is weight in kg and H is height in meters, then $B = W/H^2$ is bmi (body mass index)
- If $X_1, X_2, \ldots, X_n$ is a random sample, then the mean of the sample is

$$\overline{X} = \frac{X_1 + X_2 + \cdots X_n}{n} = \frac{1}{n}X_1 + \frac{1}{n}X_2 + \cdots + \frac{1}{n}X_n$$

More generally, we are interested in determining the distribution of the random variable Y if

$$Y = f(X_1, X_2, \ldots, X_n)$$

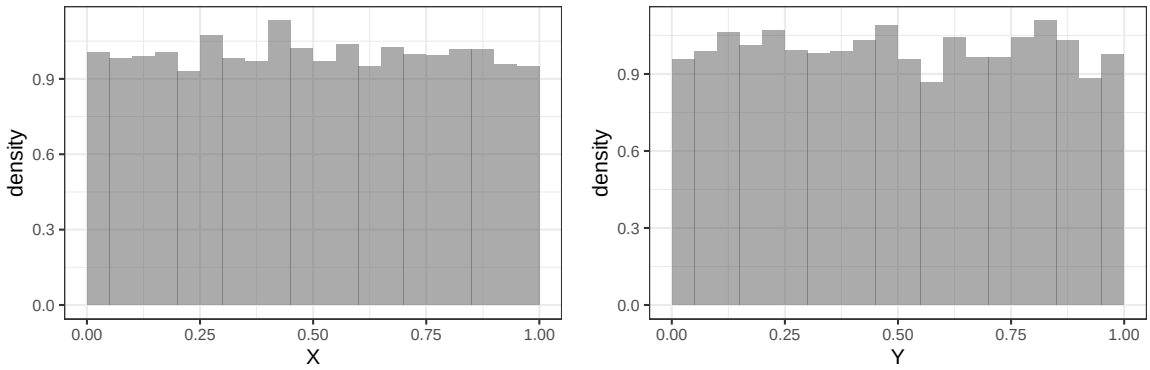
and we already know the distributions of $X_1, X_2, \ldots, X_n$. There are methods for working out the distribution of Y in many such situations, Often it is possible to figure out the distribution of variables formed this way, but the methods require more techniques of probability than we will develop in this course. We will generally be satisfied with one of the following approaches:

1. Use simulations to approximate the new distribution.
2. Calculate (or estimate) the mean and variance of the new distribution, which is often much easier than determining the pdf.
3. Rely on theorems that tell us the new distribution in certain special cases.

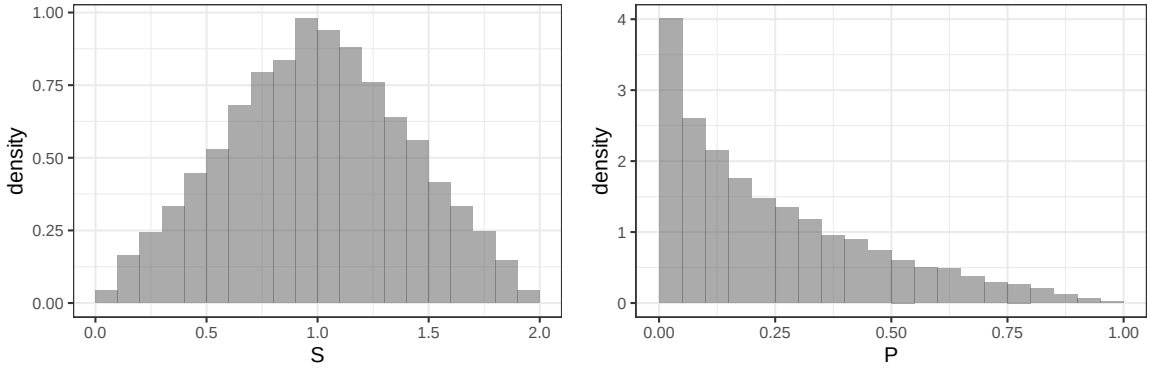
5.1 Simulations

In this section we will make use of the fact that each of our familiar distributions has a function in R that lets us simulate randomly sampling data from that distribution.

```
X <- runif(10000, 0,1)
Y <- runif(10000, 0,1)
S <- X + Y
P <- X * Y
gf_dhistogram( ~ X , main = "Sample from Unif(0,1)", binwidth = .05, center = 0.025)
gf_dhistogram( ~ Y , main = "Another sample from Unif(0,1)", binwidth = .05, center = 0.025)
```



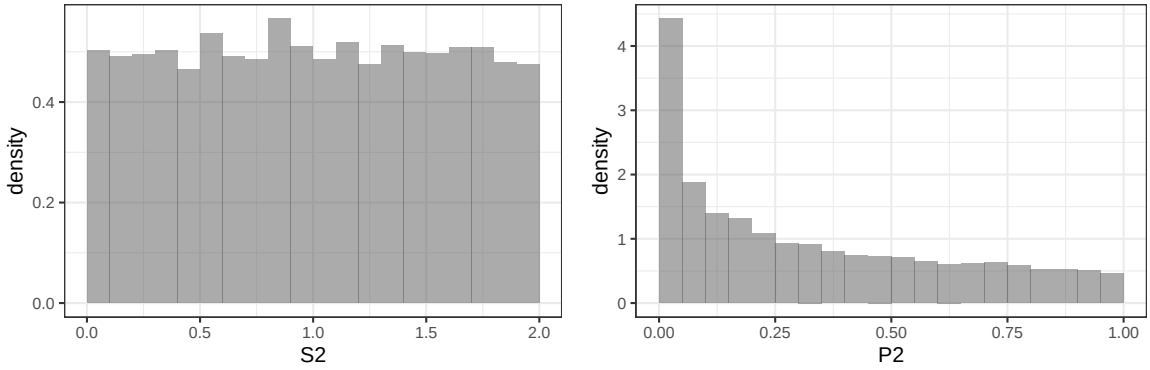

```
gf_dhistogram( ~ S , main = "Sum of two iid Unif(0,1) rvs", binwidth = 0.1, center = 0.05)
gf_dhistogram( ~ P , main = "Product of two iid Unif(0,1) rvs", binwidth = 0.05, center = 0.025)
```



Independence Matters

It is important that we have created `x` and `y` independently. Independent random variables that have the same distribution are called **independent identically distributed** (iid) random variables. As an illustration of an extreme situation where the variables in our sum are not independent, let's use the values of `x` in both roles:

```
S2 <- X + X
P2 <- X * X
gf_dhistogram( ~ S2 , main = "Sum of two non-iid Unif(0,1) rvs", binwidth = 0.1, center = 0.05)
gf_dhistogram( ~ P2 , main = "Product of two non-iid Unif(0,1) rvs", binwidth = 0.05, center = 0.025)
```



Notice how different the resulting distributions are, especially for the sum.

Similar procedures can be used to give an approximate distribution for any combination of random variables that we can simulate.

5.2 Propagation of Mean and Variance

Sometimes it is not necessary to know everything about a distribution. Sometimes knowing the mean or variance suffices, and there are several common situations where the mean and variance are easy to calculate

5.2.1 Linear Transformations

The easiest of these is a linear transformation of a random variable.

If X is a random variable with known mean and variance, then

$$\mathrm{E}(aX+b) = a\,\mathrm{E}(X) + b \text{ , and}$$
$$\mathrm{Var}(aX+b) = a^2\,\mathrm{Var}(X) \text{ .}$$

These are actually pretty easy to prove from the definitions of mean and variance. (Just write down the integrals and do some algebra.) But these results also match our intuition.

1. If we add or subtract a constant b, that increases or decreases every value by the same amount. This will increase the mean by that amount, but does nothing to the variance (since everything is no more or less spread out than it was before). This explains the $+b$ in the first equation and why b does not appear at all in the formula for the variance.

$$\mathrm{E}(X+b) = \int_{-\infty}^{\infty} (x+b) f(x)\,dx = \int_{-\infty}^{\infty} x f(x)\,dx + \int_{-\infty}^{\infty} b f(x)\,dx = \mathrm{E}(X) + b$$
$$\mathrm{Var}(X+b) = \int_{-\infty}^{\infty} (x+b-(\mu+b))^2 f(x)\,dx = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx = \mathrm{Var}(X)$$

2. Now consider multiplying each value by the same amount a. This scales all the values by a, and hence scales the mean by a as well. This also makes the values more or less spread out (more when $|a| > 1$, less

when $|a| < 1$). But it is the standard deviation – not the variance – that increases or decreases by the factor $|a|$. The variance scales with a^2.

$$\mathrm{E}(aX) = \int_{-\infty}^{\infty} axf(x)\,dx = a\int_{-\infty}^{\infty} xf(x)\,dx = a\,\mathrm{E}(X)$$
$$\mathrm{Var}(aX) = \int_{-\infty}^{\infty} (ax-a\mu)^2 f(x)\,dx = a^2\int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx = a^2\,\mathrm{Var}(X)$$

Example 5.2.1. Q. Suppose X has a mean of 5 and a standard deviation of 2. What are the mean and standard deviation of $3X+4$?

A. $\mathrm{E}(3X+4) = 3\,\mathrm{E}(X)+4 = 3\cdot 5+4 = 19$

$\mathrm{Var}(3X+4) = 3^2\,\mathrm{Var}(X) = 3^2\cdot 2^2 = 36$. So he standard deviation of $3X+4$ is $\sqrt{36} = 6$. Notice that $6 = 2\cdot 3$.

5.2.2 Sums

The most important combination of two random variables is a sum.

Let X and Y be two random variables with known means and variances, then

$$\mathrm{E}(X+Y) = a\,\mathrm{E}(X)+\mathrm{E}(Y)\text{ , and}$$
$$\mathrm{Var}(X+Y) = \mathrm{Var}(X)+\mathrm{Var}(Y)\text{ , provided } X \text{ and } Y \text{ are independent.}$$

That is,

- The expected value of a sum is the sum of the expected values.
- The variance of a sum is the sum of the variances – *provided the variables are independent.*

The independence condition for the variance rule is critical.[1]

Example 5.2.2. Q. Suppose X and Y are independent random variables with means 3 and 4 and standard deviations 1 and 2. What are the mean and standard deviation of $X+Y$?

A. $\mathrm{E}(X+Y) = 3+4$. $\mathrm{Var}(X+Y) = 1^2+2^2 = 5$, so $\mathrm{SD}(X+Y) = \sqrt{5} \approx 2.236$.

Example 5.2.3. Q. Let $X \sim \mathsf{Unif}(0,1)$ and $Y \sim \mathsf{Unif}(0,1)$ be independent random variables and let $S = X+Y$. What are the mean and variance of $S = X+Y$?

A. $\mathrm{E}(S) = \mathrm{E}(X)+\mathrm{E}(Y) = \frac{1}{2}+\frac{1}{2} = 1$. $\mathrm{Var}(S) = \mathrm{Var}(X)+\mathrm{Var}(Y) = \frac{1}{12}+\frac{1}{12} = \frac{1}{6}$.

Note that this matches the mean and variance of a $\mathsf{Triangle}(0,2,1)$-distribution, since

$$\frac{0+2+1}{3} = 1\text{ ,}$$

and

$$\frac{0^2+2^2+1^2-0\cdot 2-0\cdot 1-1\cdot 2}{18} = \frac{3}{18} = \frac{1}{6}\text{ .}$$

In fact, it can be shown that $S \sim \mathsf{Triangle}(0,2,1)$.

[1] These results can be proved by setting up the appropriate integrals and rearranging them algebraically. In this case, one needs to know a bit about joint, marginal, and conditional distributions, so for the sake of time we will omit the proofs.

If we express the rule for variances in terms of standard deviations we get

The Pythagorean Theorem for standard deviations. If X and Y are independent random variables, then

$$\mathrm{SD}(X+Y) = \sqrt{\mathrm{SD}(X)^2 + \mathrm{SD}(Y)^2}\,.$$

The independence condition plays the role of the right triangle condition in the usual Pythagorean Theorem.

5.2.3 Linear Combinations

The results in the preceding sections can be combined and iterated to get results for arbitrary linear combinations of random variables.

Let $Y = a_1X_1 + a_2X_2 + \cdots + a_kX_k$, then

$$\mathrm{E}(Y) = a_1\,\mathrm{E}(X_1) + a_2\,\mathrm{E}(X_1) + \cdots a_k\,\mathrm{E}(X_k)$$

$$\mathrm{Var}(Y) = a_1^2\,\mathrm{Var}(X_1) + a_2^2\,\mathrm{Var}(X_1) + \cdots a_k^2\,\mathrm{Var}(X_k)\text{ , provided } X_1, X_2, \ldots, X_k \text{ are independent.}$$

Example 5.2.4. Q. Suppose the means and standard deviations of three independent random variables are as in the table below.

variable	mean	standard deviation
X	100	15
Y	120	20
Z	110	25

Determine the mean and standard deviation of $X + 2Y - 3Z$.

A. The mean is $100 + 2(120) - 3(110) = 10$.

The variance is $1^2 \cdot 15^2 + 2^2 \cdot 20^2 + (-3)^2 \cdot 25^2 = 7450$, so the standard deviation is $\sqrt{7450} = 86.3$.

5.3 Normal distributions are special

Normal distributions are special because

Special Properties of Normal Distributions

1. Linear combinations of independent normal random variables are again normal.

2. Sums of iid random variables from *any distribution* are approximately normal provided the number of terms in the sum is large enough.

   This result follows from what is known as the **Central Limit Theorem**. The Central Limit Theorem explains why the normal distributions are so important and why so many things have approximately normal distributions.

This means that just knowing the mean and standard deviation tells us everything we need to know about the distribution of the linear combination of normal random variables.

Example 5.3.1. Let $X \sim \text{Norm}(10, 2)$ and $Y \sim \text{Norm}(12, 4)$. If X and Y are independent, then

$$X + Y \sim \text{Norm}(10 + 12, \sqrt{2^2 + 4^2}) = \text{Norm}(22, 4.47) \ .$$

Example 5.3.2. Let $X \sim \text{Norm}(10, 2)$ and $Y \sim \text{Norm}(12, 4)$. If X and Y are independent, then

$$X - Y \sim \text{Norm}(10 - 12, \sqrt{2^2 + 4^2}) = \text{Norm}(-2, 4.47) \ .$$

Example 5.3.3. Q. Use simulation to illustrate the previous two results.

A.

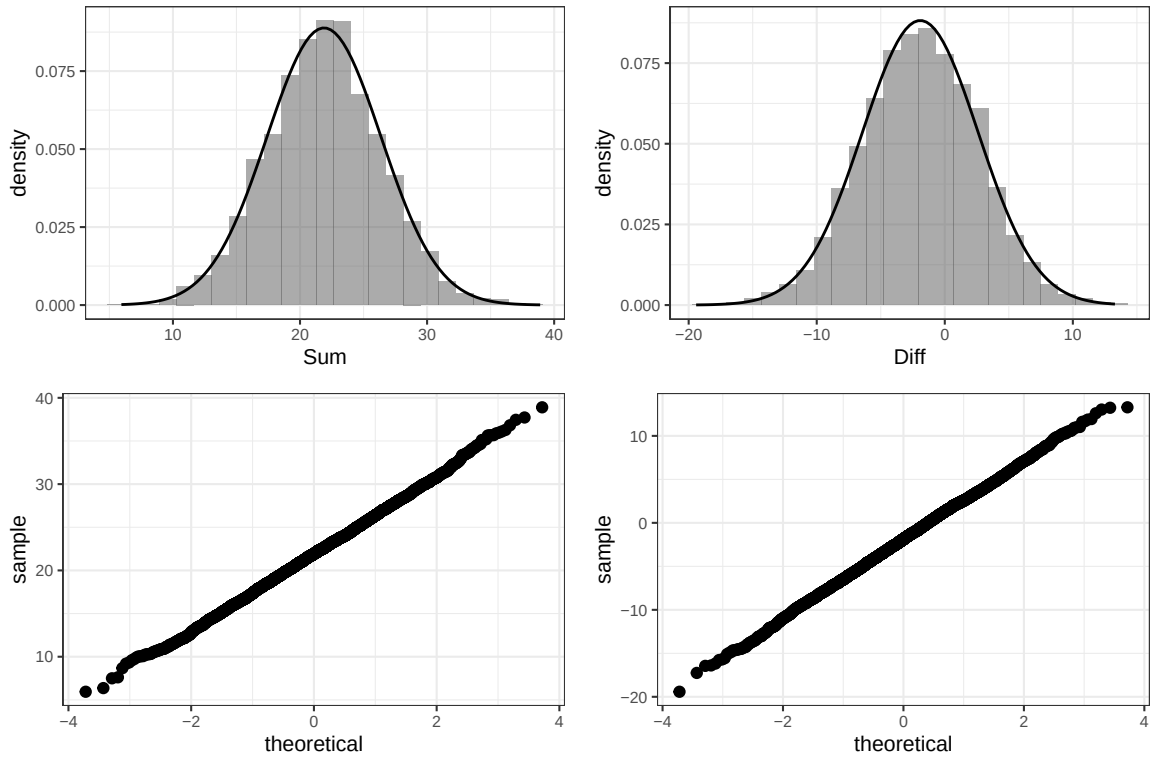
```
X <- rnorm(5000, 10, 2)
Y <- rnorm(5000, 12, 4)
Sum <- X + Y
Diff <- X - Y
fitdistr(Sum, "normal")
```

```
##     mean       sd
##   21.8897    4.4900
##  ( 0.0635) ( 0.0449)
```

```
fitdistr(Diff, "normal")
```

```
##      mean        sd
##   -1.91518    4.52236
##  ( 0.06396) ( 0.04522)
```

```
gf_dhistogram(~ Sum) %>% gf_fitdistr(dist = "norm")
gf_dhistogram(~ Diff) %>% gf_fitdistr(dist = "norm")
gf_qq(~ Sum)
gf_qq(~ Diff)
```



Example 5.3.4. Let $X_i \overset{\text{iid}}{\sim} \text{Unif}(0,1)$. Consider $S = \sum_{i=1}^{12} X_i$. Since $\text{E}(X_i) = \frac{1}{2}$ and $\text{Var}(X_i) = \frac{1^2}{12} = \frac{1}{12}$

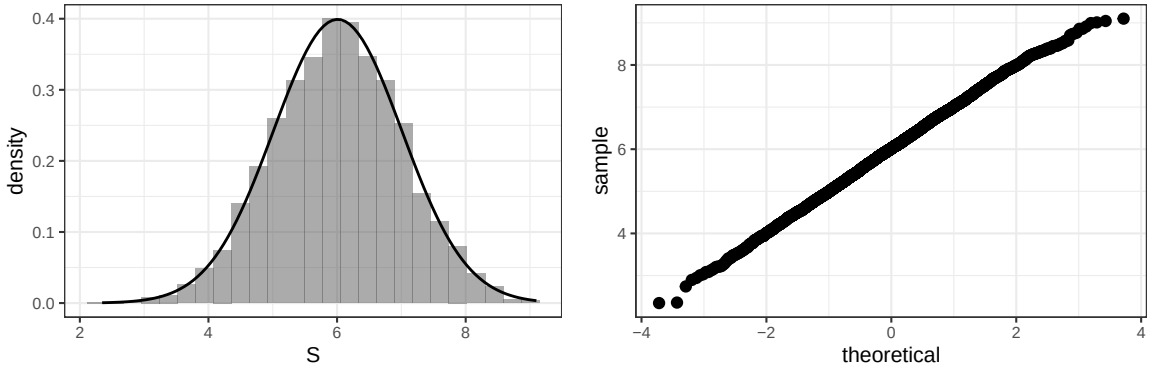
$$\text{E}(S) = \frac{1}{2} + \frac{1}{2} + \cdots \frac{1}{2} = 12 \cdot \frac{1}{2} = 6$$
$$\text{Var}(S) = \frac{1}{12} + \frac{1}{12} + \cdots \frac{1}{12} = 12 \cdot \frac{1}{12} = 1$$

Furthermore, the normal approximation for S is quite good:

```
X1 <- runif(5000,0,1); X2 <- runif(5000,0,1); X3 <- runif(5000,0,1)
X4 <- runif(5000,0,1); X5 <- runif(5000,0,1); X6 <- runif(5000,0,1)
X7 <- runif(5000,0,1); X8 <- runif(5000,0,1); X9 <- runif(5000,0,1)
X10 <- runif(5000,0,1); X11 <- runif(5000,0,1); X12 <- runif(5000,0,1)
S <- X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12
fitdistr(S, "normal")
```

```
##      mean        sd
##   6.009021   0.999654
##  (0.014137) (0.009997)
```

```
gf_dhistogram(~S)  %>% gf_fitdistr(dist = "norm")
gf_qq(~S)
```



This means that $S - 6 \approx \mathsf{Norm}(0, 1)$. This has been used in computer software as a relatively easy way to simulate normal data given a good psuedorandom number generator for $\mathsf{Unif}(0, 1)$.

5.4 Estimating the Mean of a Population by Sampling

As important as data are in statistics, typically we are not interested in our data set, but rather in what we can learn from our data about some larger situation. For example,

1. Quality assurance engineers test a few parts to make a decision about whether the production process is working correctly for all the parts.
2. Automobile manufacturers crash a small number of vehicles to learn how their (other) cars might perform in an accident.
3. Public opinion pollsters survey a (relatively) small number of people in order to learn about the opinions of millions of people.
4. In order to estimate the number of dimes in a large sack of times, you decide to weigh the sack of dimes and divide by the mean weight of a dime. To do this you need to know the mean weight of a dime. You decide to carefully weigh 30 dimes and use those weights to estimate the mean weight of a dime.

We have now developed enough background to begin learning how this process works. We begin by introducing some key terms:

population The collection of individuals, objects, or processes we want to know something about.

parameter A number that describes (a feature of) a population.

In typical applications, parameters are unknown and data are collected for the purpose of estimating parameters.

sample The collection of individuals, objects, or processes we have data about. Ideally, the sample is a well chosen subset of the population.

statistic A number that describes (a feature of) a sample.

sampling distribution The distribution of a statistic under random sampling.

The process of random sampling leads to a random sample, from which a statistic could be computed. Since that number depends on a random process (sampling), it is a random variable. The sampling distribution should not be confused with the distribution of an individual sample (nor with the distribution of the population).

Examples 5.4.1.

1. Quality assurance engineers test a few parts to make a decision about whether the production process is working correctly for all the parts.
   - population: all parts produced at the plant
   - sample: the parts tested in the quality control protocol
   - parameter: mean strength of all parts produced at the plant
   - statistic: mean strength of the tested parts
2. Automobile manufacturers crash a small number of vehicles to learn how their (other) cars might perform in an accident.

   The tested cars are the sample. All of the cars produced are the population.
   - population: all cars (of a certain model) produced
   - sample: the small number of cars that were used in the crash test
3. Public opinion pollsters survey a (relatively) small number of people in order to learn about the opinions of millions of people.
   - population: all voters
   - sample: people actually contacted
   - parameter: proportion of all voters who will vote for candidate A
   - statistic: proportion of sample who claim they will vote for candidate A
4. The mean weight of a dime can be estimated from the weights of 30 dimes.
   - population: all dimes in the sack
   - sample: 30 dimes actually weighed
   - parameter: the mean weight of all the dimes in the sack
   - statistic: the mean weight of the 30 dimes actually weighed.

The Central Limit Theorem. If $X_1, X_1, \ldots, X_n$ is an iid random sample (of some quantitative variable) from a population with mean μ and standard deviation σ, then the sampling distribution of the sample mean or sample sum of a large enough random sample is approximately normally distributed. In fact,

- $\overline{X} \approx \mathsf{Norm}(\mu, \frac{\sigma}{\sqrt{n}})$
- $\sum_{i=1}^{n} X_i \approx \mathsf{Norm}(\mu, \sigma\sqrt{n})$

The approximations are better

- when the population distribution is similar to a normal distribution, and
- when the sample size is larger,

and are exact when the population distribution is normal.

The Central Limit Theorem is illustrated nicely in an applet available from the Rice Virtual Laboratory in Statistics (`http://onlinestatbook.com/stat_sim/sampling_dist/index.html`).

Important things to note about the Central Limit Theorem

1. There are three distributions involved: the population, individual sample(s), and the sampling distribution.
2. Large enough is usually not all that large (30–40 is large enough for most quantitative population distributions you are likely to encounter).
3. We could already calculate the expected value and variance of means and sums, the new information in the Central Limit Theorem is about the shape of the resulting sampling distribution.
4. The Central Limit Theorem requires a random sample. In situations where random sampling is not possible, the Central Limit Theorem may still be approximately correct or other more complicated methods may be required.

5.4.1 Estimands, estimates, estimators

When the goal of sampling is to estimate a parameter, it is handy to have the following terminology:

estimand A parameter we are trying to estimate. (Sometimes also called the **measureand**.)

estimate A statistic calculated from a particular data set and used to estimate the estimand. (Sometimes called a **measurement**.)

estimator A random variable obtained by calculating an estimate from a random sample.

unbiased estimator An estimator for which the expected value is equal to the estimand. So an unbiased estimator is "correct on average".

In this section, our estimand is the mean of the population (mu) and our estimator is $\overline{X}$, the mean of a random sample. We will use $\overline{x}$ to denote the estimate computed from a particular sample; this is an estimate.

5.4.2 If we knew σ

Typically we will not know σ or μ. (To know them, one would typically need to know the entire population, but then we would not need to use statistics to estimate μ because we would know the exact answer.) But for the moment, let's pretend we live in a fantasy world where we know σ.

Example 5.4.2. Suppose the standard deviation of the weight of all dimes in our sack of dimes is 0.03. If we collect a random sample of 25 dimes, then

$$\overline{X} \approx \mathsf{Norm}(\mu, \frac{\sigma}{\sqrt{n}}) = \mathsf{Norm}(\mu, 0.006)$$

so

$$\overline{X} - \mu \approx \mathsf{Norm}(0, \frac{\sigma}{\sqrt{n}}) = \mathsf{Norm}(0, 0.006) \ .$$

This means that

$$\mathrm{P}(|\overline{X} - \mu| \leq 0.006) \approx 0.68$$

$$\mathrm{P}(|\overline{X} - \mu| \leq 0.012) \approx 0.95$$

So we can be quite confident that our sample mean will be within 0.012 g of the μ, mean weight of all dimes in the sack.

Expressed in words, the claim is that 95% of random samples lead to a sample mean that is within 0.012 g of the true mean. Of course, that means 5% of samples lead to a sample mean that is farther away than that. For any given sample, there is no way to know if it is one of the 95% or one of the 5%.

5.4.3 Confidence Intervals (σ known)

The typical way of expressing this is with a confidence interval. The key idea is this:

$$\text{If } \overline{X} \text{ is close to } \mu, \text{ then } \mu \text{ is close to } \overline{X}.$$

So an approximate 95% confidence interval is

$$\overline{x} \pm 2SE = \overline{x} \pm 2\frac{\sigma}{\sqrt{n}}$$

or more precisely

$$\overline{x} \pm 1.96\frac{\sigma}{\sqrt{n}}$$

because

```
qnorm(0.975)
```

```
## [1] 1.96
```

Notice the switch from $\overline{X}$ to $\overline{x}$. We used $\overline{X}$ when we were considering the random variable formed by taking a random sample and computing the sample mean. $\overline{X}$ is a random variable with a distribution. When we are considering a specific data set, we write $\overline{x}$ instead.

There is a subtly here that often gets people confused about the interpretation of a confidence interval. Although 95% of samples result in 95% confidence intervals that contain the true mean, it is not correct to say that a particular confidence interval has a 95% chance of containing the true mean. Neither the particular confidence interval nor the true mean are random, so no reasonable probability statement can be made *about a particular confidence interval computed from a particular data set.*

5.4.4 Confidence Intervals (σ unknown)

The more typical situation is that σ is not known and needs to be estimated from the data. It has been known for a long time that when randomly sampling from a normal population,

$$\mathrm{E}\left(\sum_{i=1}^{n}(X_i - \overline{X})^2\right) = (n-1)\sigma^2$$

This means that

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}$$

is an unbiased estimator of σ^2.[2] This explains the reason for the $n-1$ in the denominator of the sample variance.[3]

[2] In fact more is known. The distribution of S^2 is a member of the Gamma family of distributions.

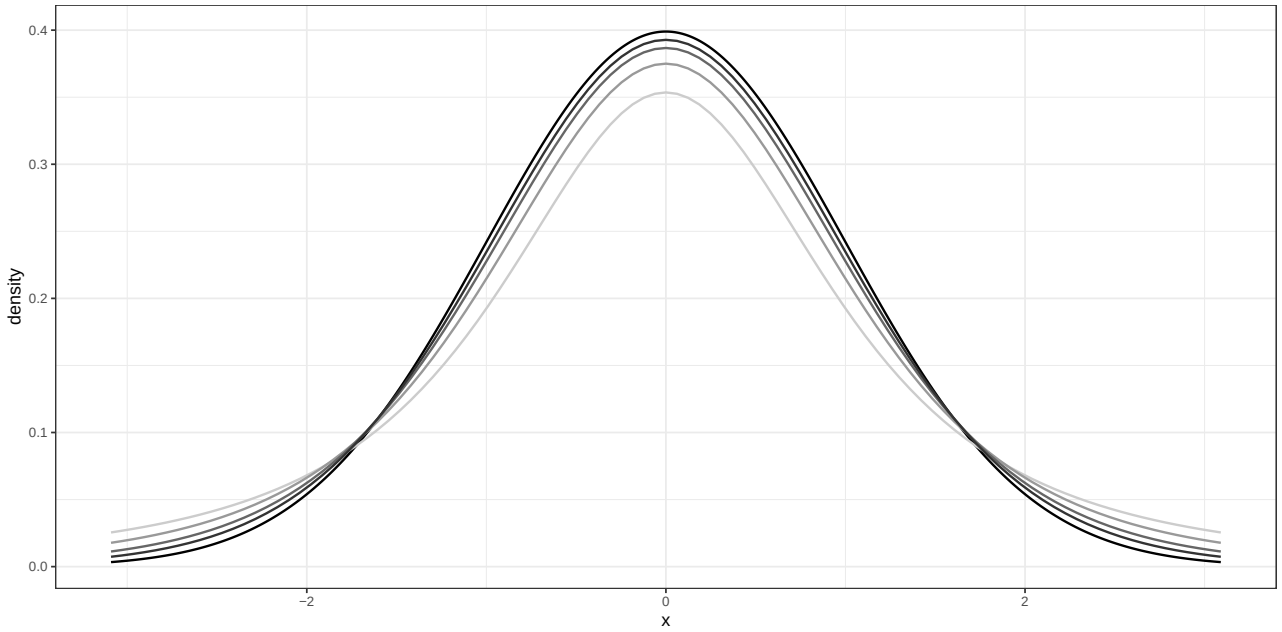
[3] Side note: the sample standard deviation is a *biased* estimator for the population standard deviation. (On average it will be too small.)

An obvious, but not quite correct solution to our unknown σ dilemma is to use s in pace of σ. In fact this was routinely done until 1908 [Stu08], when William Gosset, publishing under the pseudonym Student, pointed out that when sampling from a $\mathsf{Norm}(\mu, \sigma)$ popoulation,

$$\frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \sim \mathsf{Norm}(0, 1) \text{ but } \frac{\overline{X} - \mu}{s/\sqrt{n}} \sim \mathsf{T}(n-1)$$

The new family of distributions (called Student's t-distributions) are very similar to the normal distributions – but "shorter and fatter". This means that one must go farther into the tails of a t-distribution to capture the central 95%.

```
gf_dist("norm", main = "Normal and T-distributions") %>%
  gf_fun(dt(x,df = 2) ~ x, col = 'gray80') %>%
  gf_fun(dt(x,df = 4) ~ x, col = 'gray60') %>%
  gf_fun(dt(x,df = 8) ~ x, col = 'gray40') %>%
  gf_fun(dt(x,df = 16) ~ x, col = 'gray20')
```



The resulting confidence interval has the form

$$\overline{x} \pm t_* \frac{s}{\sqrt{n}}$$

5.4.5 Standard Error

The Central Limit Theorem tells us that (under certain condidtions), the standard deviation of the sampling distribution for the sample mean is $\frac{\sigma}{\sqrt{n}}$. Typically we don't know σ so we estimate this quantity with $\frac{s}{\sqrt{n}}$. To avoid having to say "the estimated standard deviation of the sampling distribution", we introduce a new term

standard error the estimated standard deviation of a sampling distribution

We will typically abbreviate standard error as SE. (Some authors use se.) Statistical software often includes standard errors in output.

Confidence intervals for the mean can now be expressed as

$$\overline{x} \pm t_* SE$$

We will see other intervals that make use of the t-distributions. All of them share a common structure:

$$\text{estimate} \pm t_* SE$$

The value of t_* needed for a 95% confidence interval is calculated similar to the way we calculated z_*, but we need to know the degrees of freedom parameter for the t-distribution ($n-1$ for this situation).

Example 5.4.3. Suppose a sample of 30 dimes has a mean weight of 2.258 g and a standard deviation of 0.022 g. We can calculate a 95% confidence interval as follows:

```
x_bar <- 2.258
t_star <- qt(0.975, df = 29); t_star

## [1] 2.045

SE <- 0.022/ sqrt(30); SE      # standard error

## [1] 0.004017

ME <- t_star * SE; ME          # margin of error

## [1] 0.008215

x_bar + c(-1,1) * ME

## [1] 2.250 2.266
```

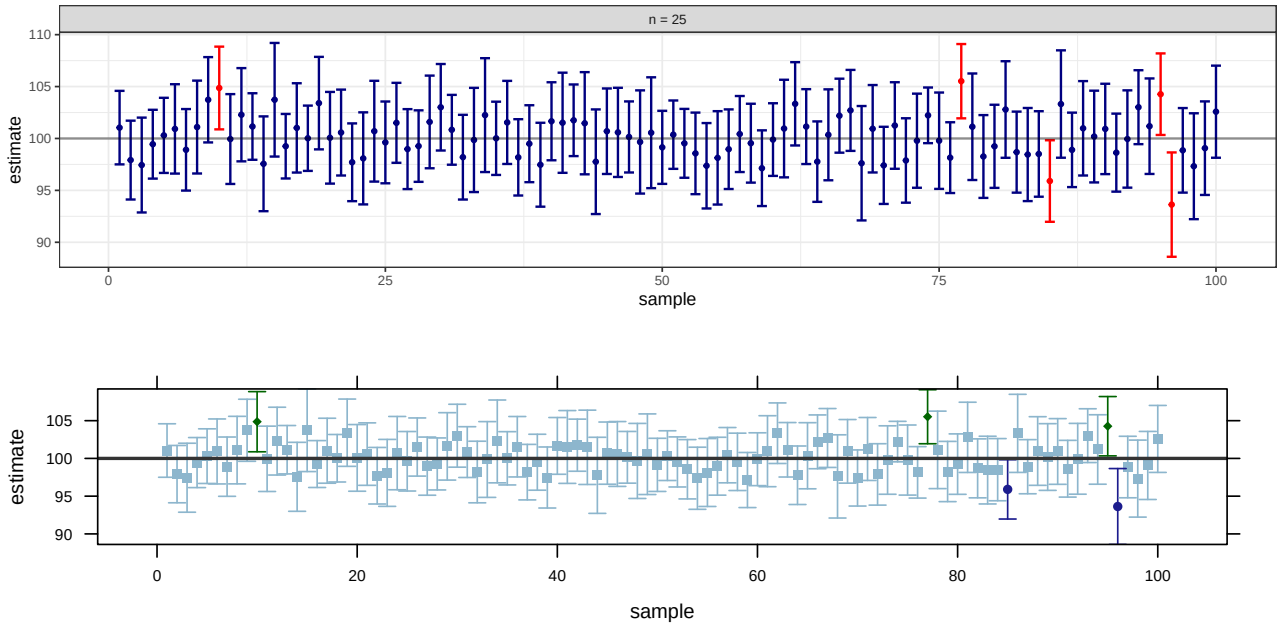
If you have the data (and not just the the summary statistics $\overline{x}$ and s), R can automate this entire computation for us with the `t.test()` function.

```
t.test( ~ mass, data = Dimes)

##
##  One Sample t-test
##
## data:  mass
## t = 560, df = 29, p-value <2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.250 2.266
## sample estimates:
## mean of x
##     2.258
```

```
confint(t.test( ~ mass, data = Dimes))   # just the CI without the other stuff


##   mean of x lower upper level
## 1     2.258  2.25 2.266  0.95
```

5.4.6 Interpreting Confidence Intervals

Here is an illustration of 100 confidence intervals computed by sampling from a normal population with mean $\mu = 100$.



Notice that some of the samples have larger means (the dots) and some smaller means. Also some have wider intervals and some narrower (because s varies from sample to sample). But most of the intervals contain the estimand (100). A few do not.

In the long-run, 95% of the intervals should "cover" the estimand and 5% should fail to cover. 95% is referred to as the **confidence level** or **coverage rate**.

As we can see, the estimand is not always contained in the confidence interval. But, in a way that we will be able to make more formal later, a confidence interval is a range of plausible values for the estimand – values that are consistent with the data in a probabilistic sense. The level of confidence is related to how strong the evidence must be for us to declare that a value is not consistent with the data.

5.4.7 Other confidence levels

We can use other **confidence levels** by using a different **critical value** t_*. So the general form for our confidence interval is

$$\overline{x} \pm t_* SE$$

Example 5.4.4. A 98% confidence interval, for example, requires a larger value of t_*. If the sample size is

$n = 30$, then we use

```
qt(0.99, df = 29)

## [1] 2.462
```

Notice the use of 0.99 in this command. We want to find the limits of the central 98% of the standard normal distribution. If the central portion contains 98% of the distribution, then each tail contains 1%.

We could also have used the following to calculate t_*.

```
qt(0.01, df = 29)

## [1] -2.462
```

Example 5.4.5. Q. Compute a 98% confidence interval for the mean weight of a dime based on our `dimes` data set.

A. We can do this by hand:

```
x_bar <- 2.258
t_star <- qt(0.99, df = 29); t_star

## [1] 2.462

SE <- 0.022/ sqrt(30); SE      # standard error

## [1] 0.004017

ME <- t_star * SE; ME          # margin of error

## [1] 0.009889

x_bar + c(-1,1) * ME

## [1] 2.248 2.268
```

or let R do the work for us:

```
t.test( ~ mass, data = Dimes, conf.level = 0.98)

##
##  One Sample t-test
##
## data:  mass
## t = 560, df = 29, p-value <2e-16
## alternative hypothesis: true mean is not equal to 0
## 98 percent confidence interval:
##  2.248 2.268
```

```
## sample estimates:
## mean of x
##     2.258
```

```
confint(t.test( ~ mass, data = Dimes, conf.level = 0.98))
```

```
##   mean of x lower upper level
## 1     2.258 2.248 2.268  0.98
```

5.4.8 Robustness

The confidence intervals based on the t-distributions assume that the population is normal. The degree to which a statistical procedure works even when some or all of the assumptions used to derive its mathematical properties are not satisfied is referred to as the **robustness** of the procedure. The t-based confidence intervals are quite robust.

Quantifying robustness precisely is difficult because how well a procedure works may depend on many factors. The general principles are

- The bigger the better.

  The larger the sample size, the less it mattes what the population distribution is.

- The more normal the better.

  The closer the population is to a normal distribution, the smaller the sample sizes may be.

For assistance in particular applications, we offer the following rules of thumb.

1. If the population is normal, the confidence intervals achieve the stated coverage rate for all sample sizes.

   But since small data sets provide very little indication of the shape of the population distribution, the normality assumption must be justified by something other than the data. (Perhaps other larger data sets collected in a similar fashion have shown that normality is a good assumption or perhaps there is some theoretical reason to accept the normality assumption.)

2. For modestly sized samples ($15 \leq n \leq 40$), the t-based confidence is acceptable as long as the distribution appears to be unimodal and is not strongly skewed.

3. For large sample size ($n \geq 40$), the t-procedure will work acceptably well for most unimodal distributions.

   But keep in mind, if the distribution is strongly skewed, the mean might not be the best parameter to estimate.

4. Because both the sample mean and the sample variance are sensitive to outliers, one should proceed with caution when outliers are present.

   Outliers that are due to mistakes and can be corrected, should be. Outliers that can be verified to be incorrect but cannot be corrected should be removed. It is not acceptable to remove an outlier just because you don't want it in your data. But sometimes statisticians do "leave one out analysis" where they run the analysis with and without the outlier. If the conclusions are the same, then the conclusions can be safely drawn. But if the conclusions are different, likely additional data will be needed to resolve the differences.

   Don't forget: sometimes the outliers are the interesting part of the story. Determining what makes them different from the rest of the data may be the most important thing.

5.5 Exercises

5.1 Let X and Y be independent $\mathsf{Gamma}(\texttt{shape} = 2, \texttt{scale} = 3)$ random variables. Let $S = X + Y$ and let $D = X - Y$. Use simulations (with 5000 replications) and quantile-quantile pltos to answer the following:

a) Fit a normal distribution to S using `fitdistr()`. Is the normal distribution a good fit?

b) Fit a Gamma distribution to S using `fitdistr()`. Is the Gamma distribution a good fit?

c) Fit a normal distribution to D using `fitdistr()`. Is the normal distribution a good fit?

d) Why is it not a good idea to fit a Gamma distribution to D?

5.2 If $X \sim \mathsf{Norm}(110, 15)$ and $Y \sim \mathsf{Norm}(100, 20)$ are independent random varaibles:

a) What is $\mathrm{P}(X \geq 140)$?

b) What is $\mathrm{P}(Y \geq 140)$?

c) What is $\mathrm{P}(X \geq 150)$?

d) What is $\mathrm{P}(Y \geq 150)$?

e) What is $\mathrm{P}(X + Y \geq 250)$?

f) What is $\mathrm{P}(X \geq Y)$? (Hint: $X \geq Y \Leftrightarrow X - Y \geq 0$.)

5.3 Suppose X and Y are independent random variables with means and standard deviations as listed below.

	mean	standard deviation
X	54	12
Y	48	9

What are the mean and standard deviation of each of the following:

a) $X + Y$

b) $2X$

c) $2X + 3Y$

d) $2X - 3Y$

5.4 You are interested to know the mean height of male Calvin students. Assuming the standard deviation is similar to that of the population at large, we will assume $\sigma = 2.8$ inches.

a) What is the distribution of $\overline{X} - \mu$? (Hint: start by determining the distribution of $\overline{X}$.)

b) If you measure the heights of a sample of 20 students, what is the probability that your mean will be within 1 inch of the actual mean?

c) How large would your sample need to be to make this probability be 95%?

5.5 Give an approximate 95% confidence interval for a population mean μ if the sample of size $n = 25$ has mean $\overline{x} = 8.5$ and the population standard deviation is $\sigma = 1.7$.

5.6 Determine the critical value t_* for each of the following confidence levels and sample sizes.

a) 95% confidence level; $n = 4$

b) 95% confidence level; $n = 24$

c) 98% confidence level; $n = 15$

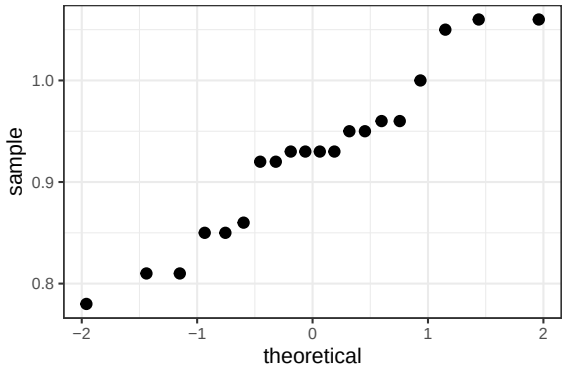
d) 90% confidence level; $n = 20$

e) 99% confidence level; $n = 12$

f) 95% confidence level; $n = 123$

5.7 Below is a normal-quantile plot and some summary information from a sample of the stride rates (strides per second) of healthy men.

```
##     response  min     Q1 median   Q3  max   mean      sd  n missing
## 1 strideRate 0.78 0.8575   0.93 0.96 1.06 0.9255 0.08095 20       0
```



a) What is the standard error of the mean for this sample?

b) Construct a 98% confidence interval for the mean stride rate of healthy men.

c) Does the normal-quantile plot suggest any reasons to worry about the normality assumption?

5.8 Lake Mary problem.

5.9 A random sample of size $n = 8$ E-glass fiber test specimens of a certain type yielded a sample mean interfacial shear yield stress of 30.2 and a sample standard deviation of 3.1. Assuming that the population of interfacial shear yield stress measurements is approximately normal, compute a 95% confidence interval for the true average stress.

5.10 The code below will create a data set containing a sample of observations of polymerization degree for some paper specimens. The data have been sorted to assist in typing. (If the data actually occurred in this order, we would probably be doing a different sort of analysis.)

```
Paper <-
        tibble(
          polymer =
            c(418, 421, 421, 422, 425, 427, 431, 434,
        437, 439, 446, 447, 448, 453, 454, 463, 465))
```

a) Create a normal-quantile plot to see if there are any reasons to worry about the assumption that the population is approximately normal.

b) Calculate a 95% confidence interval for the mean degree of polymerization for the population of such paper runs. The authors of the paper did this too.

c) Based on your confidence interval, is 440 a plausible value for the mean degree of polymerization? Explain.

5.11 Using the same data, Alice constructs a 95% confidence interval and Bob creates a 98% confidence interval. Which interval will be wider? Why?

5.12 Charlie and Denise are working on the same physics lab. Charlie leaves lab early and only has a sample size of $n = 15$. Denise stays longer and has a sample size of $n = 45$. Each of them construct a 95% confidence interval from their samples.

a) Whose confidence interval would you expect to be wider?

b) Under what conditions could it be the other way around?

5.13 Find an article from the engineering or science literature that computes a confidence interval for a mean (be careful, you may see confidence intervals for many other parameters) and also reports the sample mean and standard deviation. Check their computation to see if you both get the same confidence interval. Give a full citation for the article you used.

Google scholar might be a useful tool for this. Or you might ask an engineering or physics professor for an appropriate engineering journal to page through in the library. Since the chances are small that two students will find the same article if working independently, I expect to see lots of different articles used for this problem.

If your article looks particularly interesting or contains statistical things that you don't understand but would like to understand, let me know, and perhaps we can do something later in the semester with your article. It's easiest to do this if you can give me a URL for locating the paper online.

Review Exercises

5.14 Even when things are running smoothly, 5% of the parts produced by a certain manufacturing process are defective.

a) If you select 10 parts at random, what is the probability that none of them are defective?

Suppose you have a quality control procedure for testing parts to see if they are defective, but that the test procedure sometimes makes mistakes:

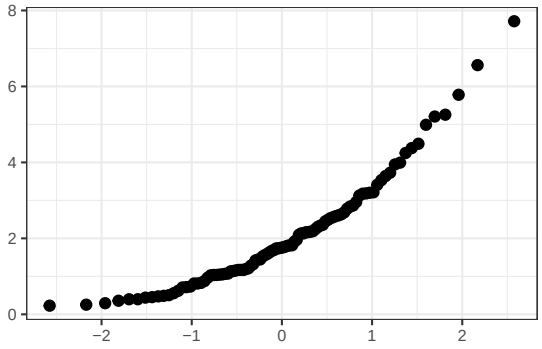
- If a part is good, it will fail the quality control test 10% of the time.
- 20% of the defective parts go undetected by the test.

b) What percentage of the parts will fail the quality control test?

c) If a part passes the quality control test, what is the probability that the part is defective?

d) The parts that fail inspection are sold as "seconds". If you purchase a "second", what is the probability that it is defective?

5.15 Here is a normal quantile plot from a data set.



Sketch what a density plot of this same data would look like.

5.16 You should know how to compute confidence intervals for a single quantitative variable both "by hand" and using `t.test()` if you have data. Here is a template problem you can use to practice both of these.

a) Select a data set and a quantitative variable in that data set. For example, `Length` in the `KidsFeet` data set.

b) Use `favstats()` to compute some summary statistics.

```
favstats ( ~ length, data = KidsFeet)

##   min Q1 median   Q3  max  mean    sd  n missing
##  21.6 24   24.5 25.6 27.5 24.72 1.318 39       0
```

c) Pick a confidence level. Example: 90% confidence.

d) From this information, compute a confidence interval

e) Now check that you got it right using `t.test()`

f) It is possible that you chose a variable for which this is not an appropriate procedure. Be sure to check for that, too.

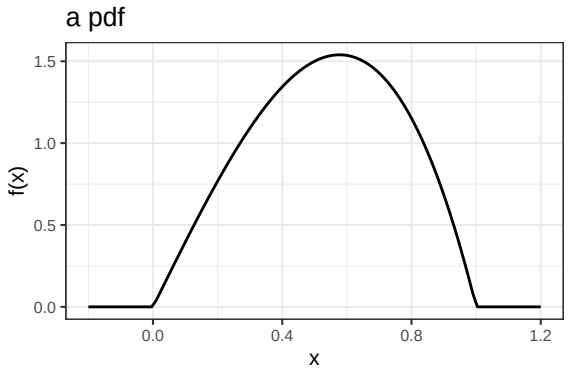
You can find other data sets using

```
data()
```

You won't run out of examples before you run out of energy for doing these.

5.17 The **pdf** for a continuous random variable X is

$$f(x) = \begin{cases} 4(x - x^3) & \text{when } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



a) Determine $\mathrm{P}(X \leq \frac{1}{2})$.

b) Determine the mean and variance of X.

5.18 The kernel of a continuous distribution is given by $k(x) = 4 - x^2$ on the interval $[-2, 2]$.

a) Determine the pdf of the distribution.

b) Compute the mean and variance of the distribution.

5.19 Let $X \sim \mathsf{Gamma}(\texttt{shape} = 3, \texttt{rate} = 2)$.

a) Plot the pdf of the distribution of X.

b) Determine $\mathrm{P}(X \leq 1)$.

c) What proportion of the distribution is between 1 and 3?

d) Use the table to determine the mean, variance, and standard deviation of X.

e) Use integration to determine the mean, variance, and standard deviation of X. (You should get the same answers as above.)

f) Make up other problems like this for any of the distributions in the table if you want additional practice.

5.20 Suppose X an Y are independent random variables with means and standard deviations as given in the following table.

variable	mean	standard deviation
X	40	3
Y	50	4

Determine the mean and standard deviation of the following:

variable	mean	standard deviation
a) $X+Y$		
b) $X-Y$		
c) $\frac{1}{2}X+7$		

Reminder: Expected value is another term for mean.

5.21

a) What is the difference between a statistic and a parameter?

b) What is a sampling distribution?

c) Create other similar problems using important terms from this class.

5.22 Critical flicker frequency (called `Flicker` in the data set below) is the lowest flicker rate at which the human eye can detect that a light source (from a flourenscent bulb or a computer screen, for example) is flickering. Knowing the cff is important for product manufacturing (since detectable flickering is annoying for the consumer). The command below loads data from from a 1973 study that attempted to determine whether the cff, which varies from person to person, is partially determined by eye color.

```
Flicker <- read.file("http://www.statsci.org/data/general/flicker.txt")


## Reading data with read.table()
```

Create a plot that can be used to visualise the data. What does your plot suggest the answer might be? (Does eye color matter?)

6 Propagation of Uncertainty

You have probably been told (in physics labs, for example) to report all measurements along with an **uncertainty**. The reporting often uses the notation:

$$\text{measurement} \pm \text{uncertainty} .$$

But what is uncertainty and how is it calculated? These are topics for this chapter.

6.1 Error and Uncertainty

Although many people mistakenly conflate the terms **error** and **uncertainty**, these are two different, but related concepts. The word "error" in the context of scientific measurement has a rather different meaning from its use in everyday English. It does not mean blunder or goof (although a blunder or goof could increase the amount of experimental error). Instead, error refers to the unavoidable fact that the measurements scientists record are not exactly correct.

Error is easily defined:

$$\text{error} = \text{estimate} - \text{estimand}$$

or, equivalently,

$$\text{error} = \text{measurement} - \text{measurand}$$

That is, error is the difference between the number we have measured or calculated and the number that calculation is attempting to estimate. In most applications, we do not know the error exactly, because we do not know the estimand. This is where uncertainty comes in.

Uncertainty is a numerical measure summarizing how large the error *might be*. There are several different types, or definitions, or uncertainty. The different definitions of uncertainty have in common that they are all trying to describe a statistical distribution of errors. Knowing something about this distribution of the errors can tell us how close our estimates tend to be to the estimand.

We will use **standard uncertainty** unless we say otherwise.

Standard uncertainty is the (estimated) standard deviation of the distribution of errors.

You may wonder how we can know the distribution of errors when we cannot know the error. That is a good question, which we will begin to address via an example.

6.2 An Example: Estimating the number of dimes in a sack of dimes

Suppose you want to estimate the number of dimes in a large sack of dimes. Here is one method you could use:

1. Measure the weight of all the dimes in the bag by placing them (without the bag) on an appropriately sized scale. (Call this $\hat{B}$, our estimate for B, the actual weight of the dimes in the bag.)
2. Measure the weight of 30 individual dimes and use those measurements to estimate the mean weight of dimes. (Call this $\hat{D}$.)
3. Combine these two estimates to compute an estimated number of dimes in the bag. ($\hat{N} = \hat{B}/\hat{D}$.)

Suppose that the dimes in our our bag together weigh 10.2 kg and the mean weight of our 30 measured dimes is 2.258. Then we would estimate the number of dimes to be

$$10200/2.258 = 4516.805 \, .$$

But how good is this estimate? Do we expect to be within a small handful of dimes? Might we be off by 100 or 500? Standard uncertainty provides a way to quantify this. But first, we need to calculate uncertainty for the two ingredients in this recipe: the total weight of all the dimes in the bag ($\hat{B}$), and the mean weight of the 30 measured dimes ($\hat{D}$).

6.2.1 Calculating a Standard Uncertainty without Using Data

We only measure the bag of dimes once (and might expect that the value observed on the digital read out would be the same if we measured it repeatedly anyway), so the distribution involved in our uncertainty calculation will be based on some assumptions about the workings of our scale. For example, we could model a reading of 10.2 kg with a $\mathsf{Unif}(10.15, 10.25)$-distribution.[1] This model reflects the assumption that if the actual weight is anywhere between 10.15 and 10.25, the reading will be 10.2 and that the actual weight is equally likely to be anywhere within that range.

If we interpret the 10.2 reading in this way, then the uncertainty can be calculated as the standard deviation of a $\mathsf{Unif}(10.15, 10.25)$-distribution:

$$u_{\hat{B}} = \frac{b-a}{\sqrt{12}} = \frac{10.25 - 10.15}{\sqrt{12}} = \frac{0.1}{\sqrt{12}} = 0.029kg = 28.868g$$

6.2.2 Calculating a Standard Uncertainty Using Data

The situation for our estimated mean weight of a dime is a little different. We weighed 30 dimes, and calculated the mean mass of one dime from those data. But if we repeated the measurements many times – taking

[1] Other models are possible, and the choice of model matters for the uncertainty calculation that will result.

another 30 dimes, calculating the average...taking *another* 30 dimes, calculating *another* average...and so on many times...how much variability would there be in the *calculated averages*? We will learn how to estimate this quantity ourselves soon; for now, we will take it as a given that it is $\frac{s}{\sqrt{n}}$, where s is the standard deviation of one sample (the masses of 30 dimes), and n is the sample size (here, 30). So the uncertainty in our mean dime weight is about:

$$u_{\hat{D}} = \frac{s}{\sqrt{n}} = \frac{0.022}{\sqrt{30}} = 0.004 \, .$$

So we would report our estimate for the mean weight of a dime as

$$2.258 \pm 0.004 \, .$$

This notation looks like a confidence interval, and indeed it is a confidence interval. Since we are using $t_* = 1$, this is approximately a 68% confidence interval:

```
pt(1,df = 29) - pt(-1,df = 29)
```

```
## [1] 0.6744
```

6.2.3 Propagating Error by the Delta Method

Now that we have computed the uncertainties for B and D, we need to find a way to combine them to determine the uncertainty for B/D. For this we use a linear approximation of the function $f(B, D) = B/D$.

Recall from calculus that

$$f(x, y) \approx f(a, b) + \frac{\partial f}{\partial x}(x - a) + \frac{\partial f}{\partial y}(y - b) \, ,$$

where the partial derivatives are evaluated at $(x, y) = (a, b)$.

If we apply this to our estimators $\hat{B}$ and $\hat{D}$, we get

$$f(\hat{B}, \hat{D}) \approx f(B, D) + \frac{\partial f}{\partial \hat{B}}(\hat{B} - B) + \frac{\partial f}{\partial \hat{D}}(\hat{D} - D) \, .$$

From this it follows that

$$\begin{aligned} \mathrm{E}(f(\hat{B}, \hat{D})) &\approx \mathrm{E}\left(f(B, D)\right) + \mathrm{E}\left(\frac{\partial f}{\partial \hat{B}}(\hat{B} - B)\right) + \mathrm{E}\left(\frac{\partial f}{\partial \hat{D}}(\hat{D} - D)\right) \\ &\approx f(B, D) + 0 + 0 \\ &= f(B, D) \, . \end{aligned}$$

(The 0's come because our estimators are approximately unbiased: $\mathrm{E}(\hat{B}) \approx B$ and $\mathrm{E}(\hat{D}) \approx D$.) This says that $\hat{B}/\hat{D}$ is a reasonable estimate for B/D – it is approximately unbiased.[2]

But we really want an expression for the uncertainty – the variance (which we will turn into a standard deviation). We use similar logic to the expectation calculation above, and we will need to use the finding (not

[2]There is a small sleight of hand here. Technically, we should evaluate the partial derivatives at the unknown values B and D. We will instead plug in our particular estimates (from our data) for $\hat{B}$ and $\hat{D}$. To denote all of this completely rigorously, we would need to have separate notation for $\hat{B}$ considered as a random variable (that has an expected value and variance) and as a number (the value computed from our particular data). We're avoiding this extra layer of notation.

proven here) that for constants a and b and random variable X, $Var(aX+b) = a^2 Var(X)$. Assuming $\hat{B}$ and $\hat{D}$ are independent,[3] a reasonable assumption in this situation, we get

$$\begin{aligned}\text{Var}(f(\hat{B},\hat{D})) &\approx \text{Var}\left(f(B,D)\right) + \text{Var}\left(\frac{\partial f}{\partial \hat{B}}(\hat{B}-B)\right) + \text{Var}\left(\frac{\partial f}{\partial \hat{D}}(\hat{D}-D)\right) \\ &= 0 + \left(\frac{\partial f}{\partial \hat{B}}\right)^2 \text{Var}(\hat{B}) + \left(\frac{\partial f}{\partial \hat{D}}\right)^2 \text{Var}(\hat{D})\end{aligned}$$

where again we evaluate the partial derivatives with $\hat{B}$ and $\hat{D}$.

Applying this to $f(\hat{B},\hat{D}) = \hat{B}/\hat{D}$, we get

$$\begin{aligned}\frac{\partial f}{\partial \hat{B}} &= \frac{1}{\hat{D}} \\ \frac{\partial f}{\partial \hat{D}} &= \frac{-\hat{B}}{\hat{D}^2}\end{aligned}$$

So our uncertainty for the estimated number of dimes (remember, we want the standard deviation, which will be the square root of the variance estimate we just derived) is

$$\sqrt{\frac{1}{2.258^2}\cdot 28.868^2 + \left(\frac{1.02\times 10^4}{2.258^2}\right)^2 \cdot 0.004^2} = 15.112\ ,$$

and we would report our estimated number of dimes as

$$4517 \pm 15 \text{ dimes}\ .$$

The method described in this example is generally referred to as the **Delta Method**. Often scientists and engineers who are using this method don't use the hat notation to distinguish between estimates/estimators and estimands. In the box below, we've dropped the hats.

The Delta Method for independent estimates

Let X and Y be independent estimates with uncertainties u_X and u_Y, and let $W = f(X,Y)$. Then the uncertainty in the estimate for W can be estimated as

$$u_W \approx \sqrt{\left(\frac{\partial f}{\partial X}\right)^2 u_X^2 + \left(\frac{\partial f}{\partial Y}\right)^2 u_Y^2}$$

where the partial derivatives are evaluated using estimated values of X and Y.

The Delta Method can be extended to functions of more (or fewer) than two variables by adding (or removing) terms. Slightly more complicated formulas exist to handle situations where the estimators are not independent (but we will not cover those in this course).

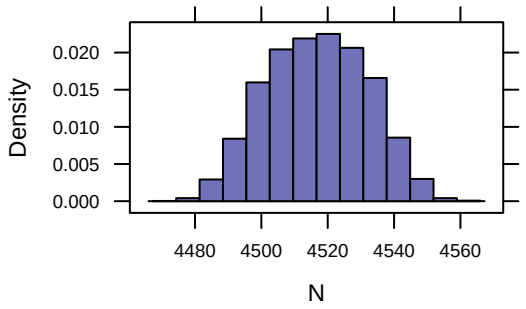
Because this method is based on using a linear approximation to f, it works better when the linear approximation is better. In particular, when $\frac{\partial^2 f}{\partial X^2}$ or $\frac{\partial^2 f}{\partial Y^2}$ are large near the estimated values of X and Y, the approximations might not be very good.

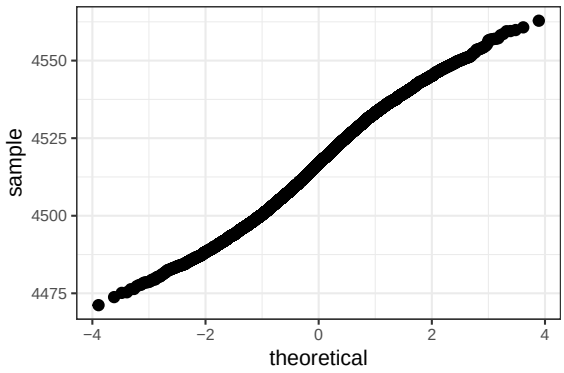
[3] A more general formula can approximate the propagation of uncertainty in cases where $\hat{B}$ and $\hat{D}$ cannot be assumed to be independent.

6.2.4 Estimating Uncertainty via Simulations

We can also estimate the uncertainty in the estimated number of dimes using simulations.

```
B <- runif(10000, 10150, 10250)
SampleMeans <- do(10000) * mean( ~ mass, data = resample(Dimes) )
head(SampleMeans ,3)


##    mean
## 1 2.259
## 2 2.257
## 3 2.253


D <- SampleMeans$mean
N <- B / D
histogram( ~ N)
gf_qq( ~ N)
sd(N)


## [1] 15.15
```





A few explanatory notes regarding the computation above.

1. `resample()` samples from a data frame *with replacement*. That is, some of the rows may appear more than once, others not at all. Resampling is a common way to estimate a sampling distribution from a single sample. `resample()` can also be used on a "bare" vector (that is, a vector of data points not contained within a data frame, as exemplified below).

```
resample(1:10)     # some items may be chosen more than once

##  [1]  9  3  9  7  9  7  2  6 10  4

sample(1:10)       # no item may be chosen more than once, so this just shuffles

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

shuffle(1:10)      # this also shuffles

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

2. When `do()` doesn't know what to call the result of what it is "doing", it calls it `result`. (Sometimes `do()` can figure out a better name.)

3. The simulated distribution of $N = B/D$ is unimodal, roughly symmetric, and reasonably well approximated by a normal distribution (but clearly not exactly normal). The Delta Method does not guarantee that the distribution will be approximately normal – it only estimates the variance. Sometimes the distribution will be skewed or have heavier or lighter tails than a normal distribution.

4. The results of the Delta Method and simulation are very close. Each is an acceptable method for approximating the same thing, so this is not surprising.

6.3 Reporting Measurements and Estimates

6.3.1 What to record, What to report

When you record the results of a measurement for which there is an *a priori* estimate of uncertainty, the uncertainty should be recorded along with the measurement itself. Similarly, reports of quantities estimated from data should also include estimated uncertainties.

As a general guideline, a properly reported scientific estimated quantity includes the following five elements:

1. A number (the estimate)
2. Units (e.g., m or kg or seconds)
3. A statement about how it was measured or calculated
4. A statement about most likely sources of (the largest components of) error
5. An estimate of the uncertainty

Example 6.3.1. If you measured the length of a pendulum using a meter stick, you might report the measurement this way:

- Length $= 0.834 \pm 0.002$ m
- Measured with a meter stick from pivot point to the center of the steel weight.
- Uncertainty reflects the limited accuracy of measurement with a meter stick.

In plots, the number is given by the scales of the plot, the units are typically included in the axes labels, uncertainties may be represented by "error bars", and a statement describing the method of measurement or calculation should appear in the plot legend.

6.3.2 How many decimal places?

Numerical values and their uncertainties should be recorded to the proper number of decimal places. Most software either reports too many significant digits or rounds numbers too much. For correct professional presentation of your data, follow these guidelines:

1. The experimental uncertainty should be rounded to one significant figure unless the leading digit is a 1, in which case, it is generally better to use two digits.

2. A measurement should be displayed to the same number of decimal places as the uncertainty on that measurement.

Note carefully the difference between significant figure and decimal place. The following examples will help:

Example 6.3.2. The timer reports a value of 0.3451 seconds. The uncertainty on the measurement is 0.0038 seconds. By Rule 1, the uncertainty should be reported to one significant figure, so we round it to 0.004 seconds. By Rule 2, the measurement must also be rounded to the third decimal place. Thus, the measurement should be reported as 0.345 ± 0.004 seconds.

Example 6.3.3. The measured value is $7.92538 \cdot 10^4$, and its uncertainty is $2.3872 \cdot 10^2$. By Rule 1, the uncertainty should be rounded to one significant figure, so $2 \cdot 10^2$. By Rule 2, we report the measurement to the same decimal place as the uncertainty, so $7.93 \cdot 10^4$. Putting it together, the measurement should be reported as $(7.93 \pm 0.02)10^4$.

Example 6.3.4. The estimated value is 89.231, and its uncertainty is 0.1472. By Rule 1, the uncertainty should be rounded to two significant figures, so 0.15. By Rule 2, we report the estimate to the same decimal place as the uncertainty, so 89.23 ± 0.15.

6.3.3 Reporting numbers in a table

Multiple similar measurements should be reported in a table. The column headings should clearly and concisely indicate the quantity in each column; the column heading must include the units. Uncertainties should be listed in a separate column, located just to the right of the measurement column. (Sometimes, uncertainties are listed in parentheses after the estimate instead; just make sure the header and legend of the table makes it clear what values are being reported, and where.)

Example 6.3.5. A lab group calculated these numbers for kinetic energy and its uncertainty:

Kinetic Energy	uncertainty
0.8682	0.059
1.0661	0.071
1.0536	0.070
1.3881	0.058
0.8782	0.108

This should be reported with appropriate rounding as

Kinetic Energy	uncertainty
0.87	0.06
1.07	0.07
1.05	0.07
1.39	0.06
0.88	0.11

6.4 Additional Propagation of Uncertainty Examples

Example 6.4.1. Q. The side of a square is measured and reported as 12.3 ± 0.2 mm. How should the area be reported?

A. Our estimate for the area is $12.3^2 = 151.29$. Our transformation is $f(x) = x^2$, so $\frac{\partial f}{\partial x} = f'(x) = 2x$. Applying the Delta Method, our uncertainty is

$$\sqrt{(2(12.3))^2(0.2)^2} = 2(12.3)(0.2) = 4.92$$

and we report the area as 151 ± 5.

It is worth looking at the relative uncertainty of the linear and area measurements.

$$\frac{0.2}{12.3} = 0.016$$
$$\frac{5}{151} = 0.032$$

So the relative uncertainty of the area measurement is twice the relative uncertainty of the linear measurement.

The preceding example demonstrates a simplified version of the Delta Method formula when we are dealing with only one estimator.

The Delta Method for one estimator

Let X be an estimator with uncertainty u_X and let $\hat{W} = f(\hat{X})$. Then the uncertainty in the estimate W can be estimated as

$$u_W \approx \left| \frac{df}{dX} \right| u_X$$

where the derivative is evaluated using the estimated value of X, $\hat{X}$.

Example 6.4.2. Q. The sides of a rectangle are measured and reported as 12.3 ± 0.2 mm and 6.3 ± 0.1 mm. How should the area be reported?

A. Our estimate for the area is $12.3 \cdot 6.3 = 77.49$.

Now we need to estimate the uncertainty. Our transformation is $f(x, y) = xy$, so $\frac{\partial f}{\partial x} = y$ and $\frac{\partial f}{\partial y} = x$.

$$\sqrt{6.3^2 \cdot 0.2^2 + 12.3^2 \cdot 0.1^2} = 1.761$$

and we should report the area as

$$77.5 \pm 1.8$$

Sometimes it is more convenient to think about **relative uncertainty**:

Absolute and relative uncertainty

$$\text{relative uncertainty} = \frac{\text{uncertainty of measurement}}{\text{magnitude of measurement}}$$

For example, if we measure a mass to be 10.2 g with an uncertainty of 0.3 g, the relative uncertainty is

$$\frac{0.3}{10.2} = 0.029 = 2.9\%$$

Often it is the case that uncertainty grows with the magnitude of the estimate, and relative uncertainty is a way of comparing the uncertainty in large measured values with the uncertainty of small measured values on a more equal basis. Relative uncertainty is also independent of the units used.

In the example above, we get a nice formula if we compute relative uncertainty instead of absolute uncertainty. Let $P = XY$ where X and Y have uncertainties u_X and u_Y. Then $\frac{\partial P}{\partial X} = Y$ and $\frac{\partial P}{\partial Y} = X$, so

$$\begin{aligned}
\frac{u_P}{P} &= \sqrt{\frac{Y^2u_X^2 + X^2u_Y^2}{P^2}} \\
&= \sqrt{\frac{Y^2u_X^2 + X^2u_Y^2}{X^2Y^2}} \\
&= \sqrt{\frac{u_X^2}{X^2} + \frac{u_Y^2}{Y^2}} \\
&= \sqrt{\left(\frac{u_X}{X}\right)^2 + \left(\frac{u_Y}{Y}\right)^2}
\end{aligned}$$

which gives a Pythagorean identity for the relative uncertainties.

The computations are the same for any product. So this Pythagorean identity for relative uncertainties can be applied to estimate uncertainties for quantities such as area (length $\times$ width), work (force $\times$ distance), distance (velocity $\times$ time), etc.

Example 6.4.3. Q. Use relative uncertainty to estimate the area of the rectangle in Example 6.4.2.

A. The relative uncertainties in the length and width are

$$\frac{0.2}{12.3} = 0.016 \text{ and } \frac{0.1}{6.3} = 0.016 \, .$$

So the relative uncertainty in the area estimation is

$$\sqrt{(0.016)^2 + (0.016)^2} = 0.023 \, .$$

Now we solve

$$\frac{u_A}{77.49} = 0.023$$

to get

$$u_A = (77.49)(0.023) = 1.761 \, .$$

Notice that this matches the result from Example 6.4.2.

Example 6.4.4. Q. When two resistors with resistances R_1 and R_2 are connected in parallel, the combined resistance satisfies

$$R = \frac{R_1R_2}{R_1 + R_2}$$

Suppose the resistances of the two resistors are reported as 20 ± 0.7 ohms and 50 ± 1.2 ohms. How should you report the combined resistance?

A. Our estimate is $\hat{R} = \frac{20 \cdot 50}{20+50} = 14.286$. To estimate the uncertainty, we need the partial derivatives $\frac{\partial R}{\partial R_1}$ and $\partial R R_2$.

$$\begin{aligned}\frac{\partial R}{\partial R_1} &= \frac{(R_1 + R_2)R_2 - (R_1 R_2)}{(R_1 + R_2)^2} \\ &= \left(\frac{R_2}{R_1 + R_2}\right)^2 \\ &= \left(\frac{50}{20 + 50}\right)^2 = 0.51\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\partial R}{\partial R_2} &= \left(\frac{R_1}{R_1 + R_2}\right)^2 \\ &= \left(\frac{20}{20 + 50}\right)^2 = 0.082\end{aligned}$$

So our estimated uncertainty is given by

$$u_R = \sqrt{(0.51)^2(0.7)^2 + (0.082)^2(1.2^2)} = 0.37 \, .$$

So we report the combined resistance as

$$14.3 \pm 0.4$$

6.5 Experimental Error and Its Causes

There are many reasons for experimental error, and it is important to identify potential causes for experimental error, to reduce their effects when possible, and to handle them appropriately in any case.

6.5.1 Random error: Same procedure, different results

Even if measurements are taken by carefully trained scientists using highly precise instruments, repeated measurements of the "same thing" may not always give the same value.

All data collection is done in the context of variability, and statistics allows us to interpret our data in this context.

The moving target

One reason that a measurement may change is that what we are measuring may be changing. Some quantities depend on environmental factors (like temperature and atmospheric pressure, for example) that may change between measurements. This sort of variability can often be reduced by attempting to control factors that might lead to such variability. For example, a delicate experiment might be conducted in a climate controlled chamber. Another solution is to use a model that includes additional variables for these quantities. Often a combination of these two approaches is used.

If measurements are made using similar (but not identical) objects, then differences among those objects may lead to variability in measurements. If measurements are made on a sample of living things, the variability from one individual to the next could be quite large. But even in the physical sciences, each run of an experiment may require the use of different "consumables" that have slightly different properties that affect our measurements.

Measurement error

Another source of error is the measuring process itself. Every measuring device has its limits, as do the humans who are using them. If you repeatedly timed how long it takes for a steel ball to fall from a fixed height, it is quite likely that you would not get exactly the same result each time. The amount of variability would likely increase if several different students were each asked to measure the time as different students might employ slightly different methods, or be more or less skillful in their measuring.

The variability from one measurement to another that would exist even if there were no moving target effect is called **measurement error**. In practice, it can be difficult or impossible to isolate the moving target effect from measurement error, so they may be combined into one source of variability which we will call **random error**. In physics, the moving target effect is often small – at least in carefully designed experiments – so that random error is dominated by the difficulties of measuring the quantity under study. In other disciplines, the relative magnitudes may be reversed.

The best way to estimate the effects of random error is by making repeated measurements and comparing them. The **discrepancies** (differences between measurements) provide an indication of the amount of random error.

Sometimes additional information can also be used to help us estimate random error. This can be especially important when our ability to take repeated measurements is limited or when limitations of our measurement apparatus make it impossible to directly observe the effects of random error.

Invisible measurement error

Although there is no theoretical limit to the precision of a numerical quantity like mass, time, velocity, etc., every measurement device has limited precision. Because of this lack of precision in our measurement device, it may well be that repeated measuremets will all look identical. *But this does not mean that they are exactly correct.*

For example, if we measure temperature using a digital thermometer that has a display showing tenths of a degree C, then any temperature between 57.15 and 57.25 will be displayed as 57.2. Any variability within that range will be invisible to our thermometer. Similarly, if we meaure with a ruler with a 1/8 inch scale, we can use interpolation to get not only to the nearest 1/8 inch, but likely to the nearest 1/16 inch or (with some practice) perhaps to the nearest 1/32 inch. But beyond that, we really cannot tell. If more precise measurements are required, a different measuring tool will need to be used.

If other sources of variability are small, this kind of measurement error may completely mask them.

6.5.2 Systematic error: A tendency to over- or under-estimate

Even if your target were not moving, and even if there were no variability in measurements from time to time, and even if our measurement apparatus were perfectly precise, there is still a chance that a measurement might not give the value we are searching for because the procedure used might tend to give results that over- or under-estimate the quantity being measured. This kind of error may be referred to as either **bias** or **systematic error**.

Bias or **Systematic error** refers to the tendency to either over- or under-estimate a quantity. It is a tendency to "be off in a particular direction".

Calibration errors

Perhaps the easiest type of systematic error to understand is **calibration error**. If a measuring device is not properly calibrated, the resulting measurements may be too large or too small. For example, if a timing device uses an internal clock that runs a bit slow, it will tend to underestimate times. It might be possible to correct for this bias by performing calibration exercises comparing this timing device to another (more accurate) device. If several similar timing devices systematically disagree, but there is no reference to calibrate with, then we have evidence that at least some of the devices are introducing systematic error, but we may not know which ones or how much.

Calibrating equipment and procedures by using them to measure or compute standard quantities is an important part of quality scientific experimentation because it helps reduce the effects of systematic bias.

Design flaws

Poorly designed experiments can also introduce systematic error. For example, imagine an experiment where a steel ball is dropped from a platform at different heights and the time is recorded until the ball hits the ground. To save time, the researchers set the platform at a specified height and drop the ball multiple times before moving the platform to a new height and repeating. Using this method, any error in measuring the height of the platform will affect all the drops from that height *in the same way*. This introduces an unknown amount of systematic error into the measurements taken at each height. An alternative design in which the heights are done in random order and in which the platform height is reset before each drop would likely have somewhat more random error but would avoid this source of systematic error.

Dealing with systematic error

Systematic error is generally more difficult to handle than random error in part because there is often no good way to measure how large systematic error might be or even to detect that it is occurring.

6.5.3 Relative Magnitudes of Errors from Different Sources

We have identified several potential sources of error. It is good to get in the habit of qualitatively determining which sources you expect to contribute relatively larger and smaller amounts of error. Often we can ignore the sources of relatively smaller potential error and focus our attention on the sources of relatively larger potential error.

6.6 Uncertainty – How Much Error Might There Be?

In Section 6.4 we did several examples of propagation of uncertainty. But uncertainty cannot be propagated to derived estimates unless we already have estimated uncertainties for the components of the derivation. Where do these uncertainty estimates come from?

As we have mentioned before, it is not possible to measure error directly. If we knew the amount of an error exactly, we would correct for it and obtain the exact, correct estimate of the value we were trying to measure.

Since we do not know the error exactly, we have to try to use our data (and our prior knowledge about the situation) to try to estimate it. In this section we focus our attention on this part of the uncertainty calculation.

6.6.1 Looking at variability in your data

If you have repeated measurements, a histogram, boxplot, or density plot of these measurements can provide a visual representations that shows both what is "typical" or "average" and how much variability there is in the data.

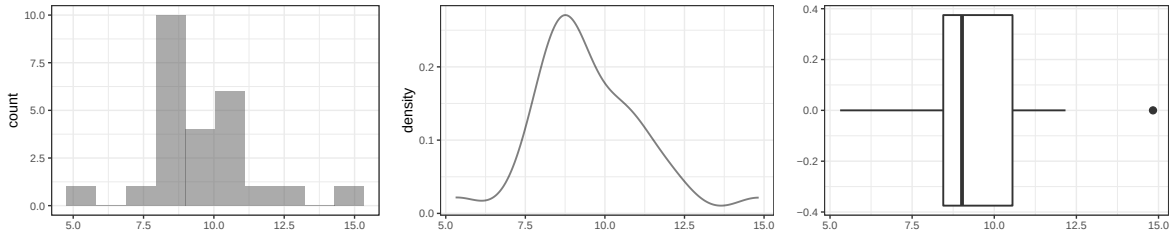


Figure 6.1: Three graphical representations of the same 25 measurents.

In addition, graphical displays of your data may help isolate outliers – values that don't seem to fit the pattern of the rest of the data. Outliers should not be removed from your data without furhter investigation. You need to know why these values are different from the rest. Was there a mistake in the measurement? Was the value recorded incorrectly (decimal point in the wrong place, wrong units, transposed digits, typo)? Or was there something different going on – so that the potential outlier is *really* an important, informative data point that is trying to tell you something about the process you are measuring? An outlier might be the key observation in your data set – don't throw it away without investigating.

6.6.2 Estimating uncertainty from your data

Whenever possible, you should make multiple measurements of the same phenomenon. The variability among these measurements allows us to estimate the uncertainty. If we are using the mean of multiple measurements as our estimate, then the uncertainty is the "standard error of the mean" (which we will derive in the following chapter):

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

This is how we estimated the uncertainty in our estimate for the mean weight of a dime.

In other more complicated study designs, some other standard error formula may be used.

6.6.3 Estimating uncertainty in other ways

Estimating uncertainty from data is only possible if

- You have multiple measurements from which to estimate the variability, and
- The measurements actually vary (aren't masked by imprecise measuring devices, for example).

Frequently, these conditions are not met, but we still want to quantify the uncertainty.

The uniform model for measurement (im)precision

There are many measuring devises that provide limited precision so that the best we may be able to say is that we know the measured value to be in some interval $[a, b]$. This would be the case for measuring devices with digital displays, for example. Imagine a device that displays 8.03 on its digital display. Presumably, this means that the actual measurement can lie anywhere between 8.025 and 8.035. This value has been rounded for digital display, and we have no way of knowing where within that interval the value might be.

We don't want to report the uncertainty as $b - a$ or even $(b - a)/2$. These would be overestimates of the "average" amount of error. Here, we aren't looking for an upper bound on the potential error; we want to estimate something like the average amount of error.

In this case we can estiamte a standard uncertainty based on the **uniform distribution**. Recall, a uniform distribution is one in which every value in some interval is "equally likely". A uniform distribution has standard deviation $s = \frac{b-a}{\sqrt{12}}$, so we will use this as our estimate for standard uncertainty as well. Notice that $\frac{2}{\sqrt{12}} = \frac{1}{\sqrt{3}} = 0.577$. This means that adding and subtracting one standard uncertainty from the center of the interval

$$\frac{a+b}{2} \pm \frac{b-a}{\sqrt{12}}$$

will cover about 58% of the interval. This is quite a bit less than the 68% covered by the central portion of a normal distribution (within 1 standard deviation of the mean). For a "back of the envelope" computation, to use an uncertainty with a similar amount of "coverage" to the "coverage" that the standard deviation has for the normal distribution, we might choose to use the approximation $\frac{b-a}{\sqrt{12}} \approx \frac{b-a}{3}$, since this will slightly over-estimate the uncertainty and lead to a central covering probability of $2/3 \approx 68\%$.

This same idea can be used when working with analog scales, but in this case, we typically can see that the value is closer to one end than the other or closer to the center than to the edges. This reduces our uncertainty by a factor of 2. For example, given a ruler marked in mm, if we can tell than a reading is closer to 12.3 mm than it is to 12.4 mm, then we are saying we know the value is in the interval $[12.3, 12.35]$, which is only half as wide as the scale of the ruler. We can then use a uniform distribution as above, except that the limits a and b of the distribution are a bit narrower now. (On some analog scales, it may be possible to do even better than this.)

Example 6.6.1. Measuring length on a ruler with a 1 mm scale, we could use $\frac{1/2}{\sqrt{12}}$ as the estimated uncertainty of the measurement if the primary source of error is reading the scale. Of course, it may be that lining up the scale with the object being measured introduces more uncertainty than reading the scale does. That may lead us to choose a larger value for our estimated uncertainty.

Other models for measurement (im)precision

Other distributions, most notably the triangle and normal distributions, are sometimes used instead of the uniform distribution to model errors in measurements made with various devices. Each of these models produces a somewhat smaller estimate for the uncertainty than you would get using a uniform distribution. In the case of the normal distribution, typically one considers half the width of the interval to be spanned by 3 standard deviations of the normal distribution (since that would capture 99.7% of the distribution. If a and b are the lower and upper limits of plausible values corresponding to a measurement, the three uncertainty calculations are as follows:

distribution	uncertainty
uniform	$\frac{b-a}{2\sqrt{3}}$
triangle	$\frac{b-a}{2\sqrt{6}}$
normal	$\frac{b-a}{2 \cdot 3}$

This makes the uniform distribution the most conservative and the normal distribution the least conservative.

Taking advantage of other data

Sometimes previous experience with a device or protocol may provide us with a good estimate for the uncertainty even before we collect our data. In such cases, we can use these *a priori* uncertainties both in planning and in analysis, but it is also good to check that the data collected are consistent with the estimated uncertainties.

6.7 Exercises

6.1 A clinical trial with 30 patients has been performed in which the volume of distribution (the theoretical volume that would be necessary to contain the total amount of an administered drug at the same concentration that it is observed in the blood plasma) of a new anti-diabetes drug was measured for each patient. The sample mean was 10.2 L with a standard deviation of 1.9 L.

a) Calculate a 95% confidence interval for the the mean volume of distribution.

b) Calculate a 99% confidence interval for the the mean volume of distribution.

c) Calculate the standard uncertainty for the mean volume of distribution.

6.2 A handbook gives the value of the coefficient of linear thermal expansion of pure copper at 20 degrees C, $\alpha_{20}(\text{Cu})$, as $16.52\times10^{-6}\,^{\circ}C^{-1}$ and simply states that "the error in this value should not exceed $0.40\times10^{-6}\,^{\circ}C^{-1}$."

a) Based on this limited information, and assuming a rectangular distribution, compute the standard uncertainty.

b) Based on this limited information, and assuming a triangular distribution, compute the standard uncertainty.

c) Why might one prefer one of these over the other?

6.3 The following data are given in in the certificate of a standard solution: C(HCl) = (0.10000 ± 0.00010) mol/l. No additional information is given on the type of the uncertainty. (The $\pm$ part here is not the uncertainty but is supposed to indicate upper and lower bounds on the error.)

a) Convert the uncertainty to standard uncertainty assuming a rectangular distribution.

b) Convert the uncertainty to standard uncertainty assuming a triangular distribution.

6.4 The area of a circle is to be calculated from a measured radius. The measurement and its standard uncertainty are reported as

$$12.5 \pm 0.3\text{m} .$$

What should the researchers report as the area?

6.5 Below are some computer-computed estimates and uncertainties. They are far too precise (there are way too many digits reported). Use standard practice to report each estimate with the correct number of digits.

estimate	uncertainty
5.43210	0.024135
1535.68	12.7342
576.3415	3.453567
0.00148932	0.0000278

6.6 A student is calculating the volume of a rectangular tank by measuring the length, width, and height. These measurements are recorded as $L = 2.65 \pm 0.02$cm, $W = 3.10 \pm 0.02$cm, and $H = 4.61 \pm 0.05$ cm.

How should the volume be reported?

6.7 Estimate (with uncertainty) the amount of gasoline burned by personal cars in a particular year in the US from the following estimates and uncertainties for that year:

quantity	estimate	uncertainty
cars per person	0.80	0.12
population (millions of people)	311.6	.2
fleet fuel efficiency (mpg)	23.7	1.7
average distance driven per vehicle (miles)	12,000	2,000

Note: Various estimates for these quantities are available online, but most do not report uncertainty. The uncertainties here reflect the number of significant figures used to report these numbers and the variability between estimates found at different web sites. Also, the methodology for determining these values is not always clear. So treat this as an exercise in propagation of error, but understand that better estimates of fuel consumption (and uncertainty) would be possible with better data.

6.8 A physics student is calculating the speed of a falling object by measuring the time it takes for the object to move between two timing sensors. If she records the time as 0.43 ± 0.02 seconds and the distance as 1.637 ± 0.006 m, how should she report the speed in m/s?

6.9

a) Work out a formula for the relative uncertainty of $Q = X/Y$ given relative uncertainties for X and Y.

b) Redo problem 6.8 using your new formula.

7 Linear Models

In Chapter 6 we learned how to estimate one quantity based on its (known) relationship to other quantities. For example, we estimated the number of dimes in a sack of dimes from our estimates of the weight of the dimes and the average weight of a dime.

In this chapter we will explore how to use data to determine the relationship among two or more variables when this relationship is not known in advance. The general framework we will use is

$$Y = f(x_1, x_2, \ldots, x_k) + \varepsilon$$

- Y is the **response** variable that we are trying to estimate from k **explanatory** or **predictor** variables $x_1, x_2, \ldots, x_k$.
- The relationship between the explanatory variables and the response variables is described by a function f.
- The relationship described by f need not be a perfect fit. The **error** term in the model, ε, describes how individual responses differ from the value give by f.

  We will model ε with a distribution – typically a distribution with a mean of 0 – so another way to think about this model is the for a given values of the predictors, the values of Y have a distribution. The mean of this distribution is specified by f and the shape by ε.

7.1 The Simple Linear Regression Model

$$Y = \beta_0 + \beta_1 x + \varepsilon \qquad \text{where } \varepsilon \sim \mathsf{Norm}(0, \sigma).$$

In other words:

- The mean response for a given predictor value x is given by a linear formula

  $$\text{mean response} = \beta_0 + \beta_1 x$$

  This can also be written as

  $$\mathrm{E}(Y \mid X = x) = \beta_0 + \beta_1 x$$

- The distribution of all responses for a given predictor value x is normal.

- The standard deviation of the responses is the same for each predictor value,

Furthermore, in this model the values of ε are independent.

There are many different things we might want to do with a linear model, for example:

- Estimate the coefficients β_0 and β_1.
- Estimate the value Y associated with a particular value of x.
- Say something about how well a line fits the data.

7.2 Fitting the Simple Linear Model

7.2.1 The Least Squares Method

We want to determine the best fitting line to the data. The usual method is the method of least squares[1], which chooses the line that has the *smallest possible sum of squares of residuals*, where residuals are defined by

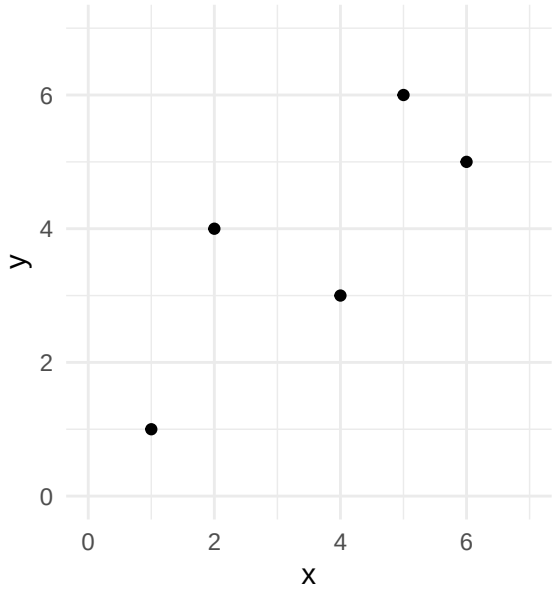
$$\text{residual} = \text{observed} - \text{predicted}$$

Example 7.2.1. Consider the following small data set.

```
someData <- data.frame(
  x = c(1,2,4,5,6),
  y = c(1,4,3,6,5)
)
someData
```

```
##   x y
## 1 1 1
## 2 2 4
## 3 4 3
## 4 5 6
## 5 6 5
```

```
gf_point( y ~ x, data = someData) %>%
  gf_lims(y = c(0, 7), x = c(0, 7))
```

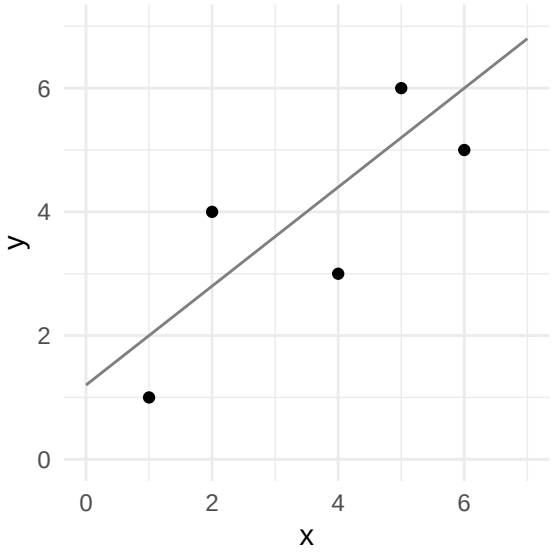


1. Add a line to the plot that "fits the data well". Don't do any calculations, just add the line.
2. Now estimate the residuals for each point relative to your line
3. Compute the sum of the squared residuals, SSE.
4. Estimate the slope and intercept of your line.

[1] In this case, it turns out that the least squares and maximum likelihood methods produce exactly the same results.

For example, suppose we we select a line that passes through $(1, 2)$ and $(6, 6)$. the equation for this line is $y = 1.2 + 0.8x$, and it looks like a pretty good fit:

```
f <- makeFun( 1.2 + 0.8 * x ~ x)
gf_point(y ~ x, data = someData) %>%
  gf_lims(x = c(0, 7), y = c(0, 7)) %>%
  gf_fun( f(x) ~ x, col = "gray50" )
```



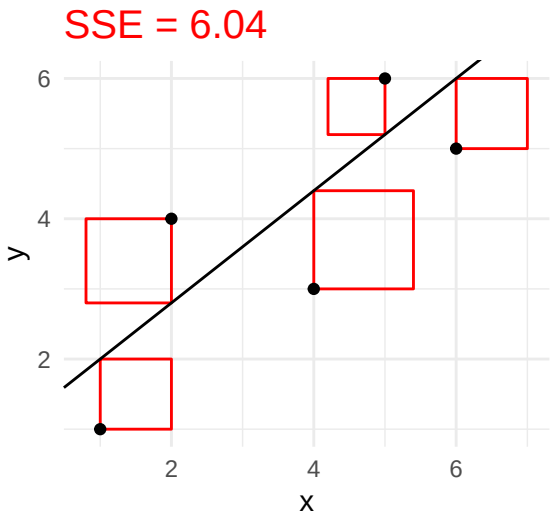
The residuals for this function are

```
resids <- with(someData, y - f(x)) ; resids

## [1] -1.0  1.2 -1.4  0.8 -1.0
```

and SSE is

```
sum(resids^2)

## [1] 6.04
```

The following plot provides a way to visualize the sum of the squared residuals (SSE).



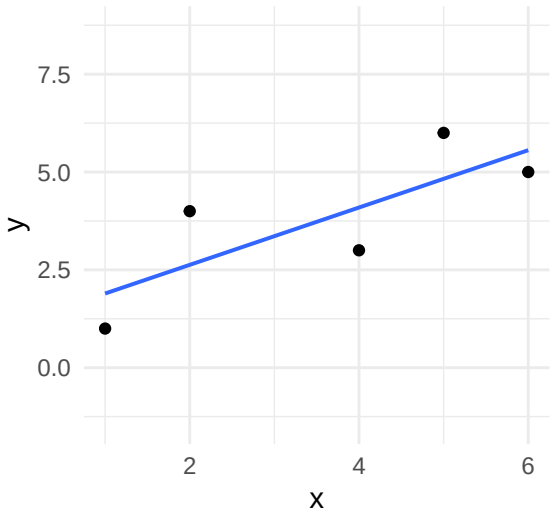
If your line is a good fit, then SSE will be small. The best fitting line will have the smallest possible SSE. The `lm()` function will find this best fitting line for us.

```
model1 <- lm( y ~ x, data = someData ); model1


##
## Call:
## lm(formula = y ~ x, data = someData)
##
## Coefficients:
## (Intercept)            x
##       1.163        0.733
```

This says that the equation of the best fit line is

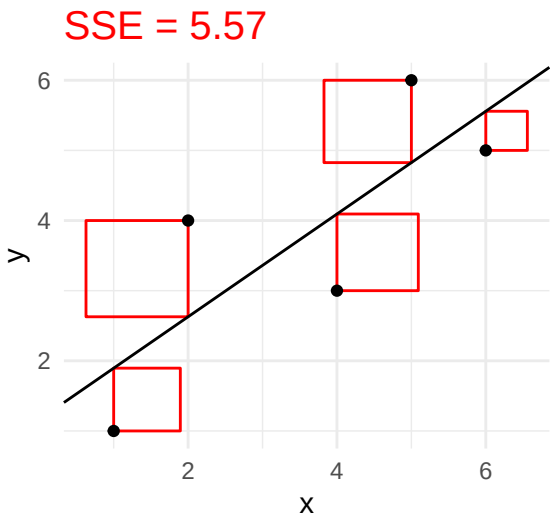
$$\hat{y} = 1.163 + 0.733x$$

```
gf_point(y ~ x, data = someData) %>%
  gf_lm()
```



We can compute SSE using the `resid()` function.

```
SSE <- sum (resid(model1)^2); SSE

## [1] 5.57
```



As we see, this is a better fit than our first attempt – at least according to the least squares criterion. It will better than *any* other attempt – it is the least squares regression line.

7.2.2 Properties of the Least Squares Regression Line

For a line with equation $y = \hat{\beta}_0 + \hat{\beta}_1 x$, the residuals are

$$e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x)$$

and the sum of the squares of the residuals is

$$SSE = \sum e_i^2 = \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x))^2$$

Simple calculus (which we won't do here) allows us to compute the best $\hat{\beta}_0$ and $\hat{\beta}_1$ possible. These best values define the least squares regression line. We always compute these values using software, but it is good to note that the least squares line satisfies two very nice properties.

1. The point $(\overline{x}, \overline{y})$ is on the line.

   This means that $\overline{y} = \hat{\beta}_0 + \hat{\beta}_1 \overline{x}$ (and $\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$)

2. The slope of the line is $b = r\dfrac{s_y}{s_x}$ where r is the **correlation coefficient**:

$$r = \frac{1}{n-1} \sum \frac{x_i - \overline{x}}{s_x} \cdot \frac{y_i - \overline{y}}{s_y}$$

Since we have a point and the slope, it is easy to compute the equation for the line if we know $\overline{x}$, s_x, $\overline{y}$, s_y, and r.

7.2.3 An Example: Estimating OSA

Example 7.2.2. In a study of eye strain caused by visual display terminals, researchers wanted to be able to estimate ocular surface area (OSA) from palpebral fissure (the horizontal width of the eye opening in cm) because palpebral fissue is easier to measure than OSA.

```
Eyes <-
  read.table("https://rpruim.github.io/Engineering-Statistics/data/PalpebralFissure.txt",  header = TRUE)
head(Eyes, 3)
```

```
##   palpebral  OSA
## 1      0.40 1.02
## 2      0.42 1.21
## 3      0.48 0.88
```

```
x.bar <- mean( ~ palpebral, data = Eyes)
y.bar <- mean( ~ OSA, data = Eyes)
s_x <- sd( ~ palpebral, data = Eyes)
s_y <- sd( ~ OSA, data = Eyes)
r <- cor( palpebral ~ OSA, data = Eyes)
c( x.bar = x.bar, y.bar = y.bar, s_x = s_x, s_y = s_y, r = r )
```

```
##  x.bar  y.bar    s_x    s_y      r
## 1.0513 2.8403 0.3798 1.2083 0.9681
```

```
slope <- r * s_y / s_x
intercept <- y.bar - slope * x.bar
c(intercept = intercept, slope = slope)
```

```
## intercept     slope
##   -0.3977    3.0800
```

Fortunately, statistical software packages do all this work for us, so the caclulations of the preceding example don't need to be done in practice.

Example 7.2.3. In a study of eye strain caused by visual display terminals, researchers wanted to be able to estimate ocular surface area (OSA) from palpebral fissure (the horizontal width of the eye opening in cm) because palpebral fissue is easier to measure than OSA.

```
osa.model <- lm(OSA ~ palpebral, data = Eyes)
osa.model
```

```
##
## Call:
## lm(formula = OSA ~ palpebral, data = Eyes)
##
## Coefficients:
## (Intercept)    palpebral
##      -0.398        3.080
```

`lm()` stands for linear model. The default output includes the estimates of the coefficients ($\hat{\beta}_0$ and $\hat{\beta}_1$) based on the data. If that is the only information we want, then we can use

```
coef(osa.model)


## (Intercept)   palpebral
##     -0.3977      3.0800
```

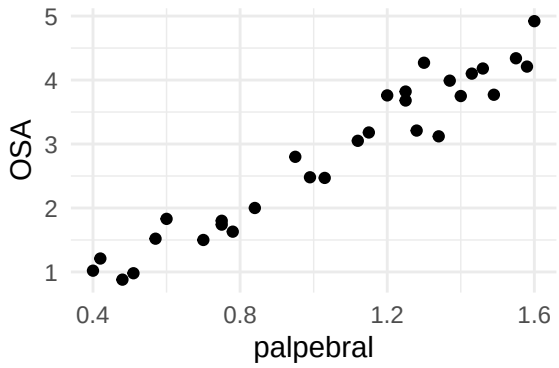
This means that the equation of the least squares regression line is

$$\hat{y} = -0.398 + 3.08x$$

We use $\hat{y}$ to indicate that this is not an observed value of the response variable but an estimated value (based on the linear equation given).

R can add a regression line to our scatter plot if we ask it to.

```
gf_point( OSA ~ palpebral, data = Eyes, type = c('p','r') )
```



We see that the line does run roughly "through the middle" of the data but that there is some variability above and below the line.

7.2.4 Explanatory and Response Variables Matter

It is important that the explanatory variable be the "x" variable and the response variable be the "y" variable when doing regression. If we reverse the roles of `OSA` and `palpebral` we do not get the same model. This is because the residuals are measured vertically (in the y direction).

7.3 Estimating the Response

We can use our least squares regression line to estimate the value of the response variable from the value of the explanatory variable.

Example 7.3.1. If the palpebral width is 1.2 cm, then we would estimate OSA to be

$$\widehat{\texttt{osa}} = -0.398 + 3.08 \cdot 1.2 = 3.298$$

R can automate this for us too. The `makeFun()` function will create a function from our model. If we input a palpebral measurement into this function, the function will return the estimated OSA.

```
estimated.osa <- makeFun(osa.model)
estimated.osa(1.2)


##     1
## 3.298
```

As it turns out, the 17th measurement in our data set had a `palpebral` measurement of 1.2 cm.

```
Eyes[17,]


##    palpebral  OSA
## 17       1.2 3.76
```

The corresponding OSA of 3.76 means that the residual for this observation is

$$\text{observed} - \text{predicted} = 3.76 - 3.298 = 0.462$$

7.3.1 Cautionary Note: Don't Extrapolate

While it often makes sense to generate model predictions corresponding to x-values *within* the range of values measured in the dataset, it is dangerous to *extrapolate* and make predictions for values *outside* the range included in the dataset. To assume that the linear relationship observed in the dataset holds for explanatory variable values outside the observed range, we would need a convincing, valid justification, which is usually not available. If we extrapolate anyway, we risk generating erroneous or even nonsense predictions. The problem generally gets worse as we stray further from the observed range of explanatory-variable values.

7.4 Parameter Estimates

7.4.1 Interpreting the Coefficients

The coefficients of the linear model tell us how to construct the linear function that we use to estimate response values, but they can be interesting in their own right as well.

The intercept β_0 is the mean response value when the explanatory variable is 0. This may or may not be interesting. Often β_0 is not interesting because we are not interested in the value of the response variable when the predictor is 0. (That might not even be a possible value for the predictor.) Furthermore, if we do not collect data with values of the explanatory variable near 0, then we will be extrapolating from our data when we talk about the intercept. (Extrapolating is dangerous because we can't really be sure that the relationships we've uncovered with our model really hold for variable values outside the range we measured.)

The estimate for β_1, on the other hand, is nearly always of interest. The slope coefficient β_1 tells us how quickly the response variable changes per unit change in the predictor. This is an interesting value in many more situations. Furthermore, when $\beta_1 = 0$, then our model does not depend on the predictor at all. So if we construct a confidence interval for β_1, and it contains 0, then we do *not* have sufficient evidence to be convinced that our predictor is of any use in predicting the response.

7.4.2 Estimating σ

There is one more parameter in our model that we have been mostly ignoring so far: σ (or equivalently σ^2). This is the parameter that describes how tightly things should cluster around the regression line. We can estimate σ^2 from our residuals:

$$\hat{\sigma}^2 = MSE = \frac{\sum_i e_i^2}{n-2}$$
$$\hat{\sigma} = RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_i e_i^2}{n-2}}$$

The acronyms MSE and $RMSE$ stand for **Mean Squared Error** and **Root Mean Squared Error**. The numerator in these expressions is the sum of the squares of the residuals

$$SSE = \sum_i e_i^2 \, .$$

This is precisely the quantity that we were minimizing to get our least squares fit.

$$MSE = \frac{SSE}{DFE}$$

where $DFE = n - 2$ is the **degrees of freedom** associated with the estimation of σ^2 in a simple linear model. We lose two degrees of freedom when we estimate β_0 and β_1, just like we lost 1 degree of freedom when we had to estimate μ in order to compute a sample variance.

$RMSE = \sqrt{MSE}$ is listed in the summary output for the linear model as the **residual standard error** because it is the estimated standard deviation of the error terms in the model.

```
summary(osa.model)
```

```
##
## Call:
## lm(formula = OSA ~ palpebral, data = Eyes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.609 -0.199 -0.019  0.217  0.664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.398      0.168   -2.37    0.025
## palpebral      3.080      0.151   20.45   <2e-16
##
## Residual standard error: 0.308 on 28 degrees of freedom
## Multiple R-squared:  0.937,Adjusted R-squared:  0.935
## F-statistic:  418 on 1 and 28 DF,  p-value: <2e-16
```

We will learn about other parts of this summary output shortly. Much is known about the estimator σ^2, including

- $\hat{\sigma}^2$ is unbiased (on average it is σ^2), and

- the sampling distribution is related to a Chi-Squared distribution with $n-2$ degrees of freedom. (Chi-Squared distributions are a special case of Gamma distributions.) More specifically,

$$\frac{SSE}{\sigma^2} = \frac{(n-2)\hat{\sigma}^2}{\sigma^2} \sim \mathsf{Chisq}(n-2) \; .$$

7.5 Checking Assumptions

7.5.1 What have we assumed?

In fitting a linear regression model, we have assumed:

- A linear relationship between the explanatory and response variables
- The errors (ϵ) are Normally distributed
- Independence of the errors (in particular, no correlation over time between successive errors for data points collected over time)
- Homoscedasticity of the errors – this means that the variance (spread) of the errors is constant over time, and over the full range of explanatory and predictor variables

7.5.2 Don't Fit a Line If a Line Doesn't Fit

The least squares method can be used to fit a line to any data – even if a line is not a useful representation of the relationship between the variables. When doing regression we should always look at the data to see if a line is a good fit. If it is not, then the simple linear model is not a good choice and we should look for some other model that does a better job of describing the relationship between our two variables.

7.5.3 Checking the Residuals

We look at the residuals (not just the data scatter plot) because some of our assumptions refer specifically to them. Also, often, it is easier to assess the linear fit by looking at a plot of the residuals than by looking at the natural scatter plot, because on the scale of the residuals, violations of our assumptions are easier to see.

So, to verify that our linear regression assumptions are sensible, we can examine the model residuals. Residuals should be checked to see that their distribution looks approximately normal and that thier standard deviation (the spread of the residuals) remains consistent across the range of our data (and across time).

In addition, especially if the data were collected over time (measurements made in order during an experiment; data points collected at a series of time points), it is important to verify that the residuals are *independent* of one another over time. To look for this problem, we can look at a scatter plot of the residuals as a function of time, and suspect a problem if we see series of very large, or very small, residuals all in a row. Another plot that can help us look for non-independence in the residuals is a plot of the autocorrelation function (ACF), obtained using the `acf()` function in R. This function computes and plots the correlation coefficient R for the residuals at various "lags". For example, the correlation coefficient for lag 1 is the correlation coefficient between each residual (corresponding to the ith datapoint) and the preceding one (the *i-1*th data point). Lag 2 is between the ith and *i-2*th data point, and so on. If the residuals are not independent, then these coefficients will have large absolute values. (Note: the "lag 0" coefficient measures the correlation of the ith residual with itself, so it is always 1. This does NOT indicate any problem with the linear regression model.)

In general, we might want to check the following plots of the residuals:

- Residuals as a function of "fitted values", or model predictions for the x-values obseved in the actual data set.

- Residuals as a function of the observed values of the explanatory variable (from the actual data set).

- Normal quantile-quantile plot of the residuals (note: in Chapter 4, we made these by hand for any distribution; you may also use the shortcut function qqmath() as illustrated below, to make them for the Normal distribution.)

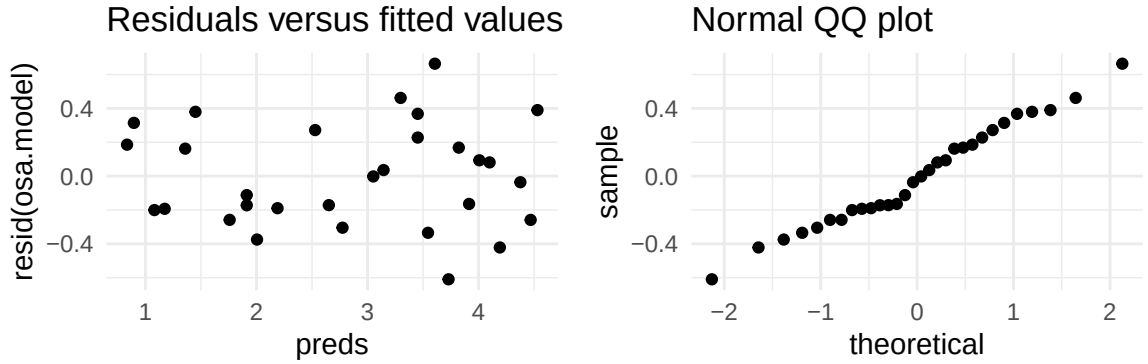
- Residuals as a function of time (if you know the order in which they were collected, or if the explanatory variable is a time-related one).

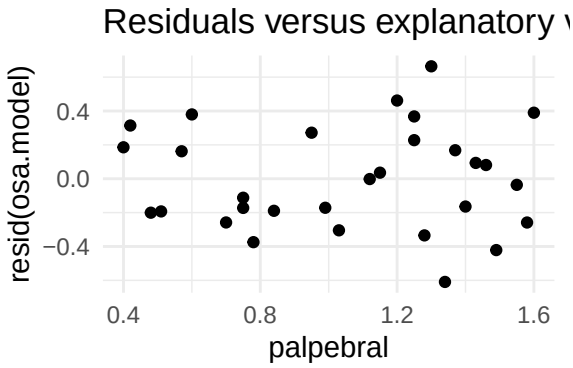
- Residual autocorrelation function plot (if the data points were collected over time, or if the explanatory variable is a time-related one).

For all the scatter plots, we want to make sure the residuals "look random" – the extent of the spread of the residuals should not vary with time or x or y ("trumpet"-shaped plot). If there is a pattern, it suggests a problem with the homoscedasticity assumption. There should not be long runs of similar residuals, especially over time; if there are, it suggests non-independence of the residuals. There should also be no apparent trends in the plot, linear or non-linear; if there are, it suggests that the relationship between the predictor and response variables was not linear. In an autocorrelation plot, the correlation coefficients (except for lag 0) should not be too large, far exceeding the dotted guide-lines on the plot; if they are, there is probably a problem with the independence assumption.

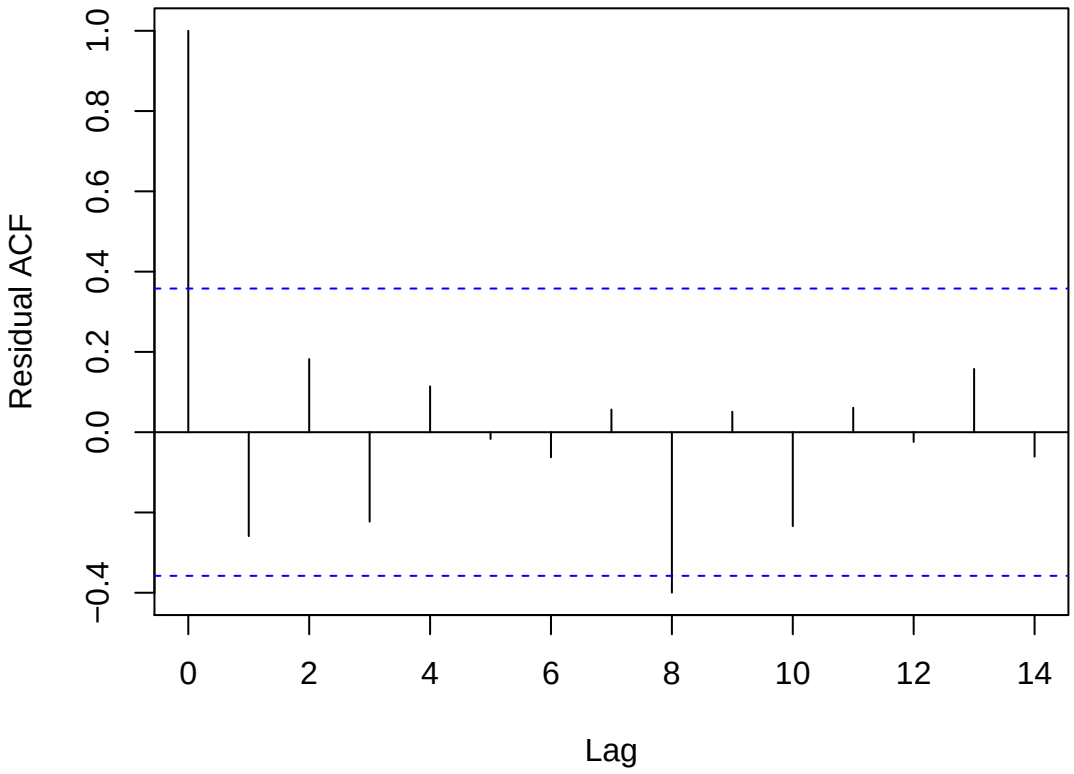
Example 7.5.1. Returning to our OSA data, we can obtain the residuals using the `resid()` function and plot them.

```
osa.hat <- makeFun(osa.model)
preds <- osa.hat(Eyes$palpebral)
gf_point( resid(osa.model) ~ preds, data = Eyes,
        title ="Residuals versus fitted values")
gf_qq( ~ resid(osa.model),  data = Eyes, title = "Normal QQ plot")
gf_point(resid(osa.model) ~ palpebral, data = Eyes,
        title = "Residuals versus explanatory variable")
```






```
acf(resid(osa.model), ylab = "Residual ACF", main = "")
```

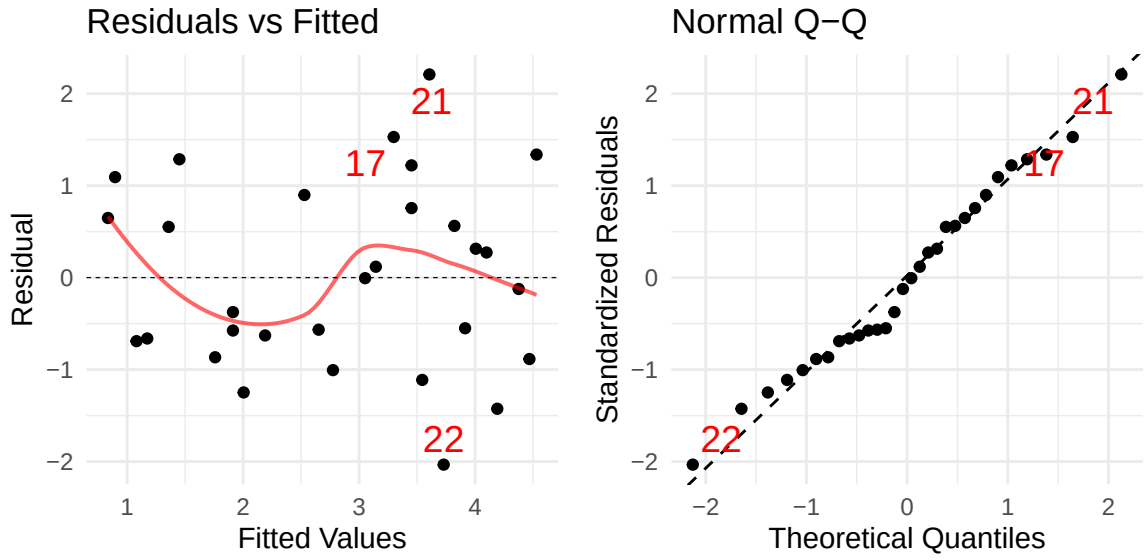


If the assumptions of the model are correct, there should be no distinct patterns to these scatter plots of the residuals, and the normal-quantile plot should be roughly linear (since the model says that differences between observed responses and the true linear fit should be random noise following a normal distribution with constant standard deviation).

In this case things look pretty good.

We can save ourselves a little typing if we create these plots using `plot()` or `mplot()`.

```
mplot(osa.model, w = 1:2)

## [[1]]

## `geom_smooth()` using formula 'y ~ x'

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



The "standardized" residuals are the residuals that have been adjusted to have an expected variance (and hence also standard deviation) of 1. Roughly, they are the residuals divided by $\hat{\sigma}$, but there is an additional adjustment that is made as well. This gives us a kind of unitless version of the residual and the normal-quantile plot using standardized residuals does a better job of indicating whether the normality assumption is compatibly with the data. (Remember, the normality assumption is about the *errors* not the residuals. The standardized residuals should behave roughly like a normal distribution if the errors are normally distributed.) Typically the shape of the normal-quantile plot made with raw residuals and the one made with standarized residuals will be very similar.

7.5.4 Outliers in Regression

Outliers can be very influential in regression, especially in small data sets, and especially if they occur for extreme values of the explanatory variable. Outliers cannot be removed just because we don't like them, but they should be explored to see what is going on (data entry error? special case? etc.)

Some researchers will do "leave-one-out" analysis, or "leave some out" analysis where they refit the regression with each data point left out once. If the regression summary changes very little when we do this, this means that the regression line is summarizing information that is shared among all the points relatively equally. But if removing one or a small number of values makes a dramatic change, then we know that that point is exerting a lot of influence over the resulting analysis (a cause for caution).

This kind of analysis can be very helpful, especially if you have one or several large potential outliers in your data set, but in this class, we will not generally do it as a matter of course (it's not a required part of model assessment for coursework).

7.6 How Good Are Our Estimates?

Assuming our diagnostics indicate that fitting a linear model is reasonable for our data, our next question is *How good are our estimates?* Notice that there are several things we have estimated:

- The intercept coefficient β_0 [estimate: $\hat{\beta}_0$]
- The slope coefficient β_1 [estimate: $\hat{\beta}_1$]
- Values of y for given values of x. [estimate: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$]

We would like to be able to compute uncertainties and confidence intervals for these. Fortunately, R makes this straightforward.

7.6.1 Estimating the βs

Example 7.6.1. Q. Returning to the OSA data, compute standard uncertainties and 95% confidence intervals for β_0 and β_1.

A. The `summary()` function provides additional information about the model:

```
summary(osa.model)
```

```
##
## Call:
## lm(formula = OSA ~ palpebral, data = Eyes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.609 -0.199 -0.019  0.217  0.664
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.398      0.168   -2.37    0.025
## palpebral      3.080      0.151   20.45   <2e-16
##
## Residual standard error: 0.308 on 28 degrees of freedom
## Multiple R-squared:  0.937,Adjusted R-squared:  0.935
## F-statistic:  418 on 1 and 28 DF,  p-value: <2e-16
```

We don't know what to do with all of the information displayed here, but we can see some familiar things in the coefficient table. If we only want the coefficients part of the summary output we can get that using

```
coef(summary(osa.model))
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -0.3977     0.1680  -2.367 2.506e-02
## palpebral     3.0800     0.1506  20.453 2.253e-18
```

From this we see the estimates ($\hat{\beta}$'s) displayed again. Next to each of those is a standard error. That is the standard uncertainty for these estimates. So we could report our estimated coefficients as

$$\beta_0 : -0.40 \pm 0.17 \qquad \beta_1 : 3.08 \pm 0.15$$

A confidence interval can be computed using

$$\hat{\beta}_i \pm t_* SE_{\beta_i}$$

because

- the sampling distribution for $\hat{\beta}_i$ is normal,
- the sampling distribution for $\hat{\beta}_i$ is unbiased (the mean is β_i), and
- the standard deviation of the sampling distribution depends on σ (and some other things), but
- we don't know σ, so we have to estimate it using $RMSE = \sqrt{MSE}$.

```
t.star <- qt(.975, df = 28); t.star    # n-2 degrees of freedom for simple linear regression
```

```
## [1] 2.048
```

```
t.star * 0.151
```

```
## [1] 0.3093
```

So a 95% confidence interval for β_1 is

$$3.08 \pm 0.31$$

The degrees of freedom used are $DFE = n - 2$, the same as used in the estimate of σ^2. (We are using a t-distribution instead of a normal distribution because we don't know σ. The degrees of freedom are those associated with using $RMSE = \sqrt{MSE}$ as our estimate for σ.)

R can compute confidence intervals for both parameters using the function `confint()`:

```
confint(osa.model)
```

```
##                2.5 %   97.5 %
## (Intercept) -0.7419 -0.05359
## palpebral    2.7715  3.38843
```

A 68% confidence interval should have a margin of error of approximately 1 standard uncertainty:

```
confint(osa.model, level = 0.68, "palpebral")
```

```
##            16 %  84 %
## palpebral 2.928 3.232
```

```
(3.2325 - 2.9275) / 2    # margin of error
```

```
## [1] 0.1525
```

```
coef(summary(osa.model))
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -0.3977     0.1680  -2.367 2.506e-02
## palpebral     3.0800     0.1506  20.453 2.253e-18
```

7.6.2 Confidence and Prediction Intervals for the Response Value

We can also create interval estimates for the response. R will compute this if we simply ask:

```
estimated.osa <- makeFun(osa.model)
estimated.osa(1.2, interval = "confidence")
```

```
##     fit   lwr   upr
## 1 3.298 3.174 3.422
```

```
estimated.osa(0.8, interval = "confidence")
```

```
##     fit   lwr   upr
## 1 2.066 1.927 2.205
```

These intervals are confidence intervals for the *mean* response. Sometimes it is desirable to create an interval that will have a 95% chance of containing a new *observation* – that is, including the anticipated error as well as the mean response. These intervals are called **prediction intervals** to distinguish them from the usual confidence interval.

```
estimated.osa <- makeFun(osa.model)
estimated.osa(1.2, interval = "prediction")
```

```
##     fit   lwr   upr
## 1 3.298 2.655 3.941
```

```
estimated.osa(0.8, interval = "prediction")
```

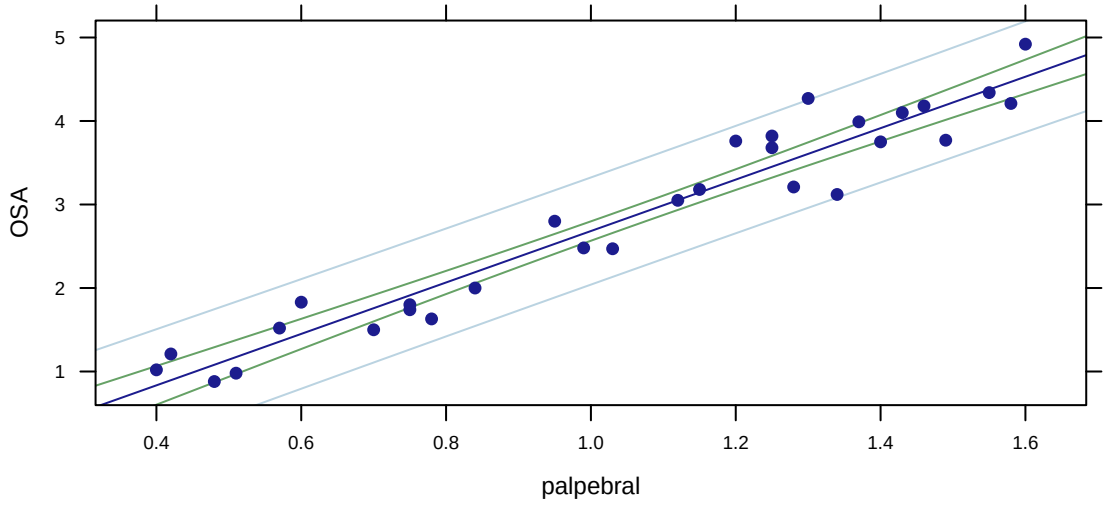
```
##     fit  lwr   upr
## 1 2.066 1.42 2.712
```

Prediction intervals are typically much wider than confidence intervals. We have to "cast a wider net" to create an interval that is highly likely to contain a new observation (which might be quite a bit above or below the mean).

The widths of both types of intervals depend on the value(s) of the explanatory variable(s) from which we are making the estimate. Estimates are more precise near the mean of the predictor variable and become less precise as we move away from there. Extrapolation beyond the observed data range is both less precise, and risky, because we don't have data to know whether the linear pattern seen in the data extends into that region.

The plot below illustrates both confidence (dotted) and prediction (dashed) intervals. Notice how most of the dots are within the prediction bands, but not within the confidence bands.

```
xyplot(OSA ~ palpebral, data = Eyes, panel = panel.lmbands)
```



A Caution Regarding Prediction Intervals

Prediction intervals are much more sensitive to the normality assumption than confidence intervals are because the Central Limit Theorem does not help when we are thinking about individual observations (essentially samples of size 1). So if the true distribution of errors is not really normal, then the prediction intervals we compute using the normality assumption will not be accurate.

7.7 Exercises

7.1 Use the output below to answer some questions about rainfall volume and runoff volume (both in m^3) for a particular stretch of a Texas highway.

```
##
## Call:
## lm(formula = runoff ~ rainfall, data = TexasHighway)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.28  -4.42   1.21   3.15   8.26
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.1283     2.3678   -0.48     0.64
## rainfall      0.8270     0.0365   22.64  7.9e-12
##
## Residual standard error: 5.24 on 13 degrees of freedom
## Multiple R-squared:  0.975,Adjusted R-squared:  0.973
## F-statistic:  513 on 1 and 13 DF,  p-value: 7.9e-12
```

a) How many times were rainfall and runoff recorded?

b) What is the equation for the least squares regression line?

c) Report the slope together with its standard uncertainty.

d) Give a 95% confidence interval for the slope of this line.

e) What does this slope tell you about runoff on this stretch of highway?

f) What is $\hat{\sigma}$?

7.2 The `KidsFeet` data set contains variables giving the widths and lengths of feet of some grade school kids.

a) Perform our usual diagnostics to see whether there are any reasons to be concerned about using a simple linear model in this situation.

b) Based on this data, what estimate would you give for the width of a Billy's foot if Billy's foot is 24 cm long? (Use a 95% confidence level.)

c) Based on this data, what estimate would you give for the average width of a kids' feet that 24 cm long? (Use a 95% confidence level.)

7.3 Some traffic engineers were interested to study interactions between bicycle and automobile traffic. One part of the study involved comparing the amount of "available space" for a bicyclist (distance in feet from bicycle to centerline of the roadway) and "separation distance" (the average distance between cyclists and passing car, also measured in feet, determined by averaging based on photography over an extend period of time). Data were collected at 10 different sites with bicycle lanes. The data are available in the `ex12.21` data set in the `Devore7` package.

a) Write out an equation for the least squares regression line for predicting separation distance from available space.

b) Give an estimate (with uncertainty) for the slope and interpret it.

c) A new bicycle lane is planned for a street that has 15 feet of available space. Give an interval estimate for the separation distance on this new street. Should you use a confidence interval or a prediction interval? Why?

d) Give a scenario in which you would use the other kind of interval.

7.4 Select only the non-diabetic men from the `pheno` data set using

```
library(fastR2)
Men <- Pheno %>% filter(sex=="M" & t2d=="control")  # note the double == and quotes here
head(Men, 3)
```

```
##     id     t2d   bmi sex   age smoker chol waist weight height    whr sbp dbp
## 1 1012 control 30.47   M 53.86 former 5.02   104   94.6  176.2 0.9327 143  89
## 2 1110 control 26.75   M 68.08  never 5.63    99   81.0  174.0 0.9252 162  91
## 3 1146 control    NA   M 62.15   <NA>   NA    NA     NA     NA     NA  NA  NA
```

This data set contains some phenotype information for subjects in a large genetics study. You can find out more about the data set with

```
?pheno
```

a) Using this data, fit a linear model that can be used to predict weight from height. What is the equation of the least squares regression line?

b) Give a 95% confidence interval for the slope of this regression and interpret it in context. (Hint: what are the units?)

c) Give a 95% confidence interval for the mean weight of all non-diabetic men who are 6 feet tall.

Note the heights are in cm and the weights are in kg, so you will need to convert units to use inches and pounds. (2.54 cm per inch, 2.2 pounds per kg)

d) Perform regression diagnostics. Is there any reason to be concerned about this analysis?

7.5 The `anscombe` data set contains four pairs of explanatory (`x1`, `x2`, `x3`, and `x4`) and response (`y1`, `y2`, `y3`, and `y4`) variables. These data were constructed by Anscombe [Ans73].

a) For each of the four pairs, us R to fit a linear model and compare the results. Use, for example,

```
model1 <- lm(y1 ~ x1, data = anscombe); summary(model1)
```

Briefly describe what you notice looking at this output. (You do not have to submit the output itself – let's save some paper.)

b) For each model, create a scatterplot that includes the regression line. (Make the plots fairly small and submit them. Use `fig.width` and `fig.height` or "output options" (the little gear icon) to control the size of the plots in RMarkdown.)

c) Comment on the results. Why do you think Anscombe invented these data?

7.6 Find an article from the engineering or science literature that uses a simple linear model and report the following information:

a) Print the first page of the article (with title and abstract) and write a full citation for the article on it. Staple this at the end of your assigment.

b) If the article is available online, provide a URL where it can be found. (You can write that on the printout of the first page of the article, too.)

c) How large was the data set used to fit the linear model? How do you know? (How did the authors communicate this information?)

d) What are the explanatory and response variables?

e) Did the paper give an equation for the least squares regression line (or the coefficients, from which you can determine the regression equation)? If so, report the equation

f) Did the paper show a scatter plot of the data? Was the regression line shown on the plot?

g) Did the paper provide confidence intervals or uncertainties for the coefficients in the model?

h) Did the paper show any diagnostic plots (normal-quantile, residuals plots, etc.)? If not, did the authors say anything in the text about checking that a linear model is appropriate in their situation?

i) What was the main conclusion of the analysis of the linear model?

j) If there is an indication that the data are available online, let me know where in case we want to use these data for an example.

Google scholar might be a useful tool for this. JSTOR (available through Heckman Library) also has a large number of scientific articles. Or you might ask an engineering or physics professor for an appropriate engineering journal to page through in the library. Since the chances are small that two students will find the same article if working independently, I expect to see lots of different articles used for this problem.

If your article looks particularly interesting or contains statistical things that you don't understand but would like to understand, let me know, and perhaps we can do something later in the semester with your article. It's easiest to do this if you can give me a URL for locating the paper online.

7.7 High population density in Japan leads to many resource usage problems, including human waste removal. In a study of a new (in the 1990's) copression machine for processing sewage sludge, researchers measures the moisture content of the compressed pellets (%) and the machines filtration rate (kg-DS/m/hr). You can load the data using

```
data(xmp12.06, package = "Devore7")
```

a) What are the least squares estimates for the intercept and slope of a line that can be used to estimate the moisture content from the filtration rate? (Give them in our usual manner with both estimate and uncertainty.)

b) The first row of the data is

```
head(xmp12.06, 1)

##   moistcon filtrate
## 1     77.9    125.3
```

Compute the residual for this observation.

c) What is $\hat{\sigma}$, the estimated value of σ?

d) Give a 95% confidence interval for the slope.

e) Give a 95% confidence interval for the mean moisture content when the filtration rate is 170 kg-DS/m/hr.

8 Beyond Linear Regression

8.1 How big is your R^2?

One part of regression model diagnostics is to check the fitted model's R^2 value, which gives an indication of the proportion of the variance in the response that has been "explained" by the model. A low value (closer to 0) means that data points are spread far around the best fit line; a high one (close to 1) means that data points are clustered very tightly around the line. A model with a low R^2 value is not necessarily "bad" – it may still provide helpful information about a real relationship between your response and predictor. However, that relationship is very "noisy," which means that your model will have poor predictive power – it will be unable to make predictions with the accuracy and precision you might hope for.

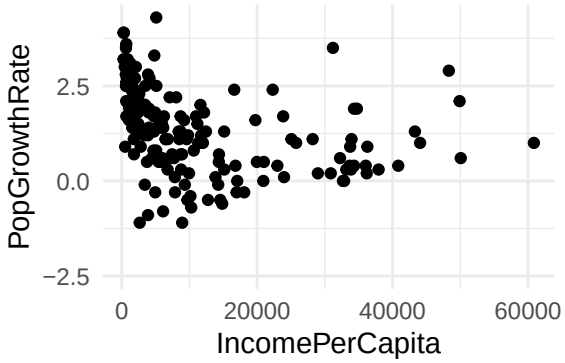
Often, the predictive power of a model, and the R^2 value, can be improved by adding additional explanatory variables – that is, fitting a model with more than one explanatory variable. It could have two predictors, three, or as many as you can (sensibly) come up with. This kind of model is called multiple regression. Mathematically, it means fitting a model of the form:

$$y\ \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 ...$$

Multiple regression often makes sense when you are studying a complex process where there is most likely "more than one thing going on." For example, you might consider modelling population growth rates worldwide using a data set including a set of social, economic, health, and political indicators compiled using data from the World Health Organization and partner organizations. The dataset description is available online: `http://www.exploredata.net/Downloads/WHO-Data-Set`. One idea might be to look for a linear relationship between per-capita income and population growth rate:

```
whodat <- read.csv("http://www.exploredata.net/ftp/WHO.csv",
                   header=TRUE, strip.white=TRUE, sep=",")
#simplify some variable names
names(whodat)[10] <- "PopGrowthRate"
names(whodat)[6] <- "IncomePerCapita"
names(whodat)[7] <- "FemaleSchoolEnrollment"
gf_point(PopGrowthRate~IncomePerCapita, data=whodat)

## Warning:  Removed 24 rows containing missing values (geom_point).

who.m1 <- lm(PopGrowthRate~IncomePerCapita , data=whodat)
summary(who.m1)
```

```
##
## Call:
## lm(formula = PopGrowthRate ~ IncomePerCapita, data = whodat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6805 -0.7100 -0.0201  0.7073  2.7894
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.66e+00   1.05e-01   15.81  < 2e-16
## IncomePerCapita -2.87e-05   6.22e-06   -4.61  7.7e-06
##
## Residual standard error: 1.04 on 176 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.108,Adjusted R-squared:  0.103
## F-statistic: 21.2 on 1 and 176 DF,  p-value: 7.72e-06
```



The R^2 value of this model is very low. But that could be because, unsurprisingly, there are *many* factors contributing to population growth rate, not *just* income. For example, what about education? Perhaps more-educated women have fewer children, lowering the population growth rate. So we might want to model population growth rate as a function of *both* income and education.

In R, a multiple regression model can be fitted with a call to `lm()`. We just add additional predictors to the right hand side of the model formula, separated by + signs. For the WHO example discussed above, for example, we could try:

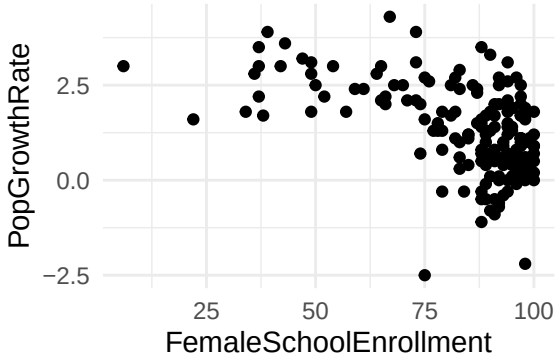
```
gf_point(PopGrowthRate~FemaleSchoolEnrollment, data=whodat)
```

```
## Warning:  Removed 23 rows containing missing values (geom_point).
```

```
who.m2 <- lm(PopGrowthRate~IncomePerCapita + FemaleSchoolEnrollment, data=whodat)
summary(who.m2)
```

```
##
## Call:
## lm(formula = PopGrowthRate ~ IncomePerCapita + FemaleSchoolEnrollment,
##     data = whodat)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.4337 -0.5298  0.0602  0.5662  2.4805
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.10e+00   3.63e-01   11.29  < 2e-16
## IncomePerCapita         -1.10e-05   6.03e-06   -1.83    0.069
## FemaleSchoolEnrollment -3.10e-02   4.50e-03   -6.89  1.1e-10
##
## Residual standard error: 0.909 on 167 degrees of freedom
##   (32 observations deleted due to missingness)
## Multiple R-squared:  0.31,Adjusted R-squared:  0.302
## F-statistic: 37.5 on 2 and 167 DF,  p-value: 3.48e-14
```



We would need to follow up with our diagnostics to fully assess these two models, but comparison of the R^2 values immediately shows that R^2 is much higher for the second model. In other words, the multiple regression has succeeded in explaining more of the variance in population growth rates than the simple linear regression with only one predictor.

8.2 Violations of Linear Regression Assumptions

In the previous chapter, we learned how to carry out regression diagnostics – to check whether or not the assumptions of linear regression analysis were valid for a particular analysis. If the assumptions are violated, then the conclusions (parameter estimates, but especially standard errors) will be incorrect, and the model results can not be trusted.

For each type of violation, there are some fixes or modifications we can try in order to fit a valid, trustworthy model to our data and still draw reliable conclusions. In this course, we will focus mainly on type of "fix": applying transformations to linearize non-linear relationships, and allow us to apply linear regression to the transformed data. This approach is covered in detail in the rest of this chapter.

Before beginning our detailed discussion of transformation, we will briefly discuss several other types of "fixes". The mathematical foundations of these more complex models are essentially beyond the scope of this class, but you should understand when they might be useful (for example, if you see a certain type of pattern in residual diagnostic plots, which technique might help solve the problem?) and be able to implement them in R.

The table below provides an overview of various problems you might uncover as you do regression diagnostics, along with possible solutions. Each entry in the table is covered in a bit more detail in the subsequent sections of this chapter.

Assumption	Description of Problem	Options
Linearity	Scatterplot (or residual plots) indicate nonlinear relationship	Transform explanatory and or response variables. Alternative: fit a non-linear model using the R function `nls()`
Normality of errors	Residual QQ plots indicates departure from normality	First check if other assumptions may also be violated, and try options listed there. If that fails, you may need to add additional predictor variables to your model; or to fit a generalized linear model, a more sophisticated type of regression that we will not cover in this course.
Independence of errors	ACF plot indicates strong dependence of errors over time (or space)	Fit a "autoregressive" model, where this relationship between subsequent or nearby measurements is expected and accounted for. To do this in R, replace `lm(y∼x)` with something like `gls(y∼x, correlation = corAR1(form=∼1))`.
Homoscedasticity	Variance of errors is not constant over the full range of response values, or over time	First, make sure that the linearity assumption is not violated. Next, if you have the option of including additional predictors in your model, it may be helpful. Next, transforming the response variable may help. Finally, if none of those options provide a solution, you can fit a model with non-constant error variance. For example, if variance increases with fitted response values, you can replace `lm(y∼x)` with something like `nls(y∼x, weights=varPower())`

8.3 Non-Normal Errors

Sometimes, during diagnostics for a linear regression model, you will find that residual quantile-quantile plots indicate that linear regression residuals are far from normally distributed. In this case, before trying to modify your model in any way, it is useful to check whether any *other* assumptions of the linear regression have *also* been violated. If they have, it is worthwhile to try to deal with those problems first, and see if solving them makes the residuals more normal.

If non-normal residuals are the only apparent problem with a linear regression model, adding additional explanatory variables *might* help in some cases. Most of the time, you would have to turn to a more sophisticated regression model called a generalized linear model (GLM). Fitting GLMs is beyond the scope of this class, and you will not be asked to do it.

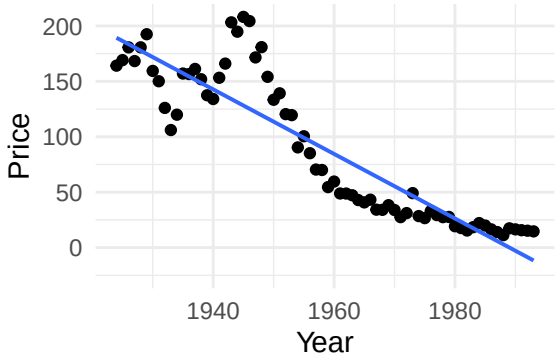
8.4 Non-Independence of Errors

Sometimes, regression diagnostics (particularly a plot of residuals as a function of time, or an ACF plot) will show that the residuals are not independent. This happens most often when the predictor variable is a temporal or spatial one; data points collected at similar times, or similar locations, are often similar to each other rather than independent.

We will consider a simple example using the price of chicken over time (in constant dollars, adjusted for inflation over time). It seems to make sense to try to predict the price of chicken as a function of time (it's been getting progressively cheaper for the last century or so):

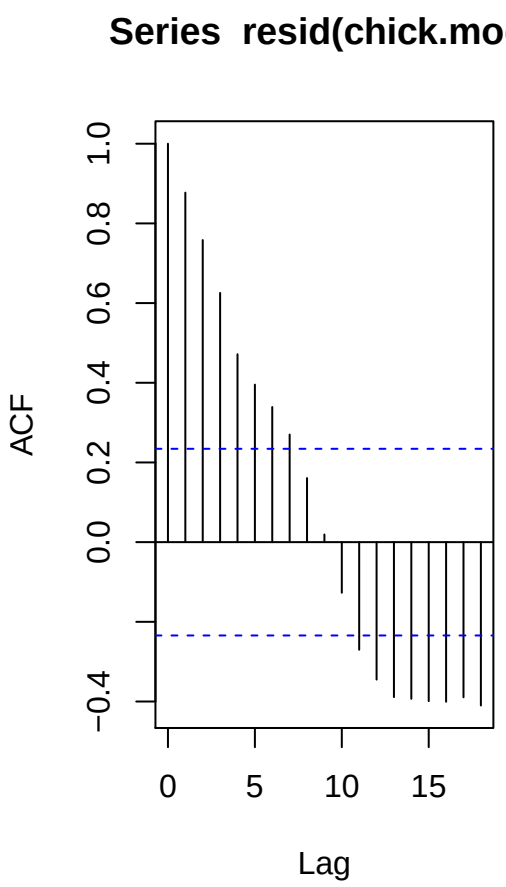
```
chickn <- read.csv("https://sldr.netlify.app/data/chickn.csv")
gf_point(Price~Year, data=chickn ) %>% gf_lm()
chick.mod <- lm(Price~Year, data=chickn)
summary(chick.mod)
```

```
##
## Call:
## lm(formula = Price ~ Year, data = chickn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57.07 -20.23  -4.75  13.20  79.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5792.692    329.995    17.6   <2e-16
## Year          -2.912      0.168   -17.3   <2e-16
##
## Residual standard error: 28.5 on 68 degrees of freedom
## Multiple R-squared:  0.815,Adjusted R-squared:  0.812
## F-statistic:  299 on 1 and 68 DF,  p-value: <2e-16
```



However, there seems to be a problem with non-independence of the residuals. Price is not independent from year to year; if you know the price was a bit high one year, it's likely to remain so for the next several years:

```
acf(resid(chick.mod))
```



This is a big problem, because it tends to result in standard error estimates that are artificially small. In other words: we think we have estimated our slope and intercept parameters *much* more precisely than we really have, and would report falsely narrow confidence intervals. To fix the problem, we can consider replacing our simple linear regression:

$$y\ \beta_0 + \beta_1 x + \epsilon$$

(where $\epsilon \sim N(0, \sigma)$) with a model that expects that subsequent residuals to depend on previous ones, so that the residual for the data point collected at time t is:

$$e_t = \rho e_{t-1} + \epsilon$$

(where, still, $\epsilon \sim N(0, \sigma)$; and ρ is a new parameter indicating how strong the dependence over time is.) This is called an AR(1) process, or an auto-regressive process of order 1. It can be fit easily in R using the function `gls()` instead of lm(). `gls()` does "generalized least-squares" fitting, and is found in the package `nlme`. The function call syntax illustrated in this example will work any time the explanatory variable is the time (or space) one that is causing the non-independence.

```
require(nlme)
chick.mod2 <-
  gls(Price~Year, data=chickn, correlation = corAR1(form = ~1))
summary(chick.mod2)


## Generalized least squares fit by REML
##   Model: Price ~ Year
##   Data: chickn
```

```
##     AIC   BIC logLik
##   557.8 566.7 -274.9
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##    Phi
## 0.9483
##
## Coefficients:
##             Value Std.Error t-value p-value
## (Intercept)  4764    1600.4   2.977  0.0040
## Year           -2       0.8  -2.921  0.0047
##
##  Correlation:
##      (Intr)
## Year -1
##
## Standardized residuals:
##     Min      Q1     Med      Q3     Max
## -1.0594 -0.5673 -0.1571  0.3361  2.0992
##
## Residual standard error: 41.39
## Degrees of freedom: 70 total; 68 residual
```

If you plot the residuals of this new model, and plot the ACF, you will see that the correlation coefficients *still have high vaules*. However, in the new `gls()` fit, this correlation has now been taken into account in the standard errors (which are larger – compare the coefficient tables to verify it), so it is OK now to trust the model parameter estimates and predictions.

8.5 Heteroscedasticity (Non-constant Error Variance)

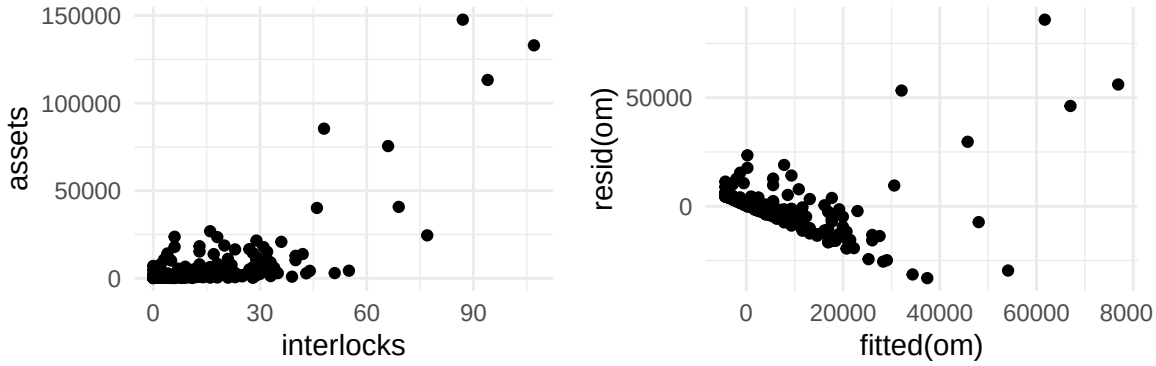
Sometimes, model diagnostics for a linear regression indicate that variance of errors is not constant over the full range of response values. Often, it is the case that the error variance grows larger as the predicted response value grows larger, resulting in a "trumpet-like" shape in the plot of residuals versus fitted values.

If you spot this problem, first, make sure that the linearity assumption is not violated. Next, if you have the option of including additional predictors in your model, it may be helpful. Next, transforming the response variable may help. Specifically, a log or square-root transformation of the response variable may be useful. (See more details and examples later in this chapter, when transformations are discussed in detail.)

Finally, if none of the previous options provide a solution, you can fit a model that actually *expects* and accounts for non-constant error variance. We will not cover this topic in any detail, but this brief example is included for your future reference (outside this class). Example: if variance increases with fitted response values, you can fit an appropriate model with the `gls` function from the `nlme` package. To do so, replace `lm(y x)` with something like `nls(y x, weights=varPower())`.

Here is a brief example, using the `Ornstein` dataset from the `car` package. It gives data on 248 Canadian companies, collected in the mid-1970s. The variable `assets` gives each company's assets in millions of dollars, and `interconnects` gives the number of director and executive positions that are shared with other firms. A scatter plot shows that the richest companies have many of these "interlocks", so we might model assets as a function of interlocks...however, the residuals have non-constant variance:

```
require(car)
gf_point(assets ~interlocks, data=Ornstein)
om <- lm(assets ~interlocks, data=Ornstein)
gf_point(resid(om) ~ fitted(om))
```



We can try to correct for this problem by fitting a model that "expects" this non-constant variance, by using the function `gls` with the input `weights=varPower()`. (There are many other ways to model non-constant error variance; this small example gives you just a taste, and for this course, you would not be expected to deal with any cases other than ones like this, where error variance increases with fitted values.)

```
om2 <- gls(assets ~interlocks, data=Ornstein, weights=varPower())
```

As with the non-independence case, if you plot the residuals for this model, you will see that they DO still have non-constant variance...but again, now it is OK because our model has taken it into account, and computed parameter estimates and standard errors appropriately.

8.6 Non-linear Relationships

The rest of this chapter will provide detailed information on how to deal with some non-linear relationships in regression.

Linear regression assumes a linear relationship between predictor and response variables, but not all relationships between pairs of quantitative variables are linear. There are two common ways to deal with nonlinear relationships:

1. Transform the data before beginning linear regression analysis, so that there *is* a linear relationship between the (transformed) variables.
2. Fit a model that explicitly expects, and accounts for, the nonlinear relationship between the two variables.

8.7 Transformations in Linear Regression

The applicability of linear models can be extended through the use of various transformations of the data. There are several reasons why one might consider a transformation of the predictor or response (or both).

- To correspond to a theoretical model.

Sometimes we have *a priori* information that tells us what kind of non-linear relationship we should anticipate. For example, an experiment to estimate Planck's constant ($\hbar$) using LED lights and a voltage meter is based on the relationship

$$V_a = \frac{\hbar c}{e\lambda} + k$$

where V_a is the activation voltage (the voltage at which the LED just begins to emit light), c is the speed of light, e is the energy of an electron, λ is the frequency of the light emitted, and k is a constant that relates to the energy losses inside the semiconductor's p-n junction. If we take c and e as known for now (in a fancier version we would work their uncertainties into this, too), we can design an experiment that measures V_a and λ for a number of different colors.

A little algebra gives us

$$V_a = \frac{\hbar c}{e} \cdot \frac{1}{\lambda} + k$$

So if we fit a model with V_a as the response and $\frac{1}{\lambda}$ as the predictor, then the slope and intercept of the resulting least squares regression line will give us and estimate for $\frac{\hbar c}{e}$, from which we can solve for $\hbar$. (Note: if we know uncertainties for c, for e, and for the slope, we can compute an estimated uncertainty for $\hbar$ using our propagation of uncertainty methods.)

Theory says that a scatter plot of V_a and $1/\lambda$ should form a straight line, so the the model we would fit would look something like

```
lm(voltage ~ I(1/wavelength), data = mydata)
```

We need to wrap `1/lambda` in `I()` because the arithmetic symbols (`+`, `-`, `*`, `/`, and `^` have special meanings inside the formula for a model. `I()` stands for *inhibit* special interpretation.

Notice that the intercpet is not directly involved in estimating $\hbar$, but that we can't fit the line and obtain our slope without it.

Many non-linear relationships can be transformed to linearity. Exercise 8.3 presents several more examples and asks you to determine a suitable transformation.

- To obtain a better fit.

  If a scatterplot or residual plot shows a clearly non-linear pattern to the data, then it would be inappropriate to fit a linear regression (and conclusions drawn from that model would be incorrect and misleading). In the absence of theoretical reasons to expect a particular mathematical relationship between the variables being studied, we may select transformations based on the shape of the relationship as revealed in a scatterplot. Section 8.7.3 provides some guidance for selecting transformations in this situation.

- To obtain better residual behavior.

  Sometimes transformations are used to improve the agreement between the data and the assumptions about the error terms in the model. For example, if the variance in the response appears to increases as the predictor increases, a logarithmic or square root transformation of the response may decrease the disparity in variance. Some transformations are used to improve the agreement between the data and the assumptions about the error terms in the model. For example, if data are heteroscedastic – for example, if the variance in the response appears to increase as the predictor increases – a logarithmic or square root transformation of the response may help.

In practice, all three of these issues are intertwined. A transformation that improves the fit, for example, may or may not have a good theoretical interpretation. Similarly, a transformation performed to achieve **homoskedasticity** (equal variance; the opposite is called **heteroskedasticity**) may result in a fit that does not match the overall shape of the data very well. Despite these potential problems, there are many situations where a relatively simple transformation is all that is needed to greatly improve the model. Here, when we say "improve" the model, we mean that the assumptions of the model are satisfied, and the model fits the data acceptably well.

8.7.1 Three Important "Laws"

In the sciences, relationships between variables based on some scientific theory are often referred to as laws. Many of these fall into one of three categories that are easily handled by transforming the data and fitting a linear regression model.

Linear Laws

We've already talked about linear relationships, but it is worth mentioning them again because there are so many situations in which a linear relationship arises.

Power Laws

Relationships of the form

$$y = Ax^p$$

are often called power laws. The two parameters are the exponent p and a constant of proportionality A. Power laws can be linearized by taking logarithms:

$$\log(y) = \log(Ax^p) = \log(A) + p\log(x)$$

So if we fit a model of the form

```
lm(log(y) ~ log(x))
```

Then $\beta_0 = \log(A)$ and $\beta_1 = p$. If a power law is a good fit for the data then

```
gf_point( log(y) ~ log(x) )
```

will produce a roughly linear plot.

Fitting a power law results in estimates for the parameters $\beta_0 = \log(A)$ and $\beta_1 = p$. Note that we can use logarithms with any base for this transformation. Typically natural logarithms are used (that's what `log()` does in R). In some specific applications we might use base 10 logarithms (`log10()` in R) or base 2 logarithms (`log2()` in R); this yields the commonly used scale for $\beta_0 = \log(A)$, the constant of proportionality.

Some common situations that are modeled with power laws include drag force vs speed, velocity vs. force, and frequency vs. force.

Exponential Laws

Relationships of the form

$$y = AB^x = Ae^{Cx}$$

are often called exponential laws. The two parameters are the base $B = e^C$ and a constant of proportionality A. Exponential laws can also be linearized by taking logarithms:

$$\log(y) = \log(AB^x) = \log(A) + x\log(B)$$

So if we fit a model of the form

```
lm(log(y) ~ x)
```

Then $\beta_0 = \log(A)$ and $\beta_1 = \log(B) = C$. If an exponential law is a good fit for the data then

```
gf_point( log(y) ~ x )
```

will produce a roughly linear plot.

Fitting an exponential law results in estimates for the parameters $\beta_0 = \log(A)$ and $\beta_1 = \log(B) = C$. Again, we will generally use natural logarithms. In this course, if you see a `log()` without an indication of the base of the logartihm, you can assume it is base "e", a natural logarithm. Similarly, remember that for R, the function `log()` takes the natural logarithm.

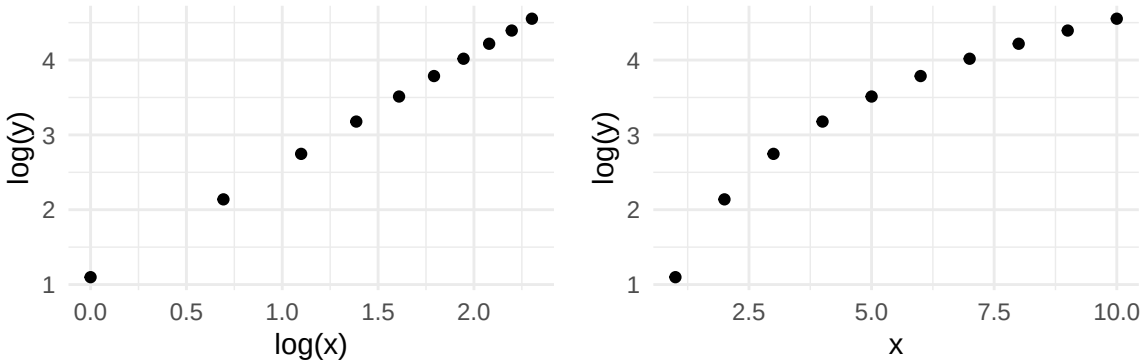
Some common situations that are modeled with exponential laws include population growth and radioactive decay. Note that exponential growth models are typically only good approximations over a limited range since exponential functions eventually grow quickly, and often some external constraints will limit this growth. For example, a culture of bacteria may grow roughly exponentially for a while, but eventually, limits on space and nourishment will make it impossible for exponential growth to continue.

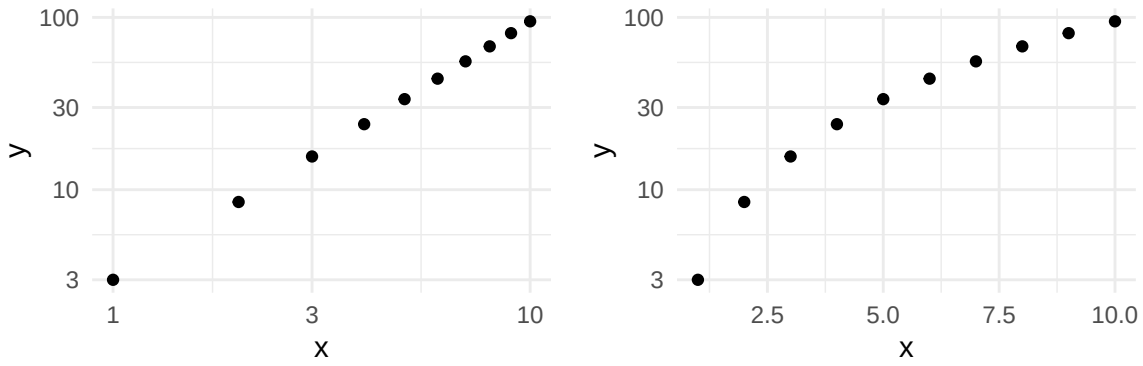
Log-log and semi-log plots

Graphs of $\log(y)$ vs. $\log(x)$ (log-log) or $\log(y)$ vs x (semi-log) can be used to assess whether the power law or exponential law appears to apply in a given situation. If the law were a perfect description of the situation, all the points on the log-log or semi-log plot would fall along a straight line. In practice, the fit won't be perfect, but the plot is a useful diagnostic. For example, you can compare a plot of y as a function of x with a log-log or semi-log plot, and see which one shows the most linear relationship between the two variables.

In the old days, before computers could readily transform the data, special graph paper was produced with semi-log or log-log scales to facilitate this sort of plot. R can easily create plots with transformed scales. Use `gf_refine()` with input `scales_*_log10()` to your call to `gf_point()`, as detailed in the example below:

```
x <- 1:10
y <- 3 * x^1.5
gf_point(log(y) ~ log(x))
gf_point(log(y) ~ x)
gf_point(y ~ x) %>%
  gf_refine(scale_x_log10(), scale_y_log10())

gf_point(y ~ x) %>%
  gf_refine(scale_y_log10())
```





As expected, the log-log transformation makes things linear. Of course, with real data, the fit won't be perfect like this.

8.7.2 Other Models That Can Be Transformed to Linear

The three laws above are not the only kinds of relationships that can be transformed to linear.

Example 8.7.1. A chemical engineering text book suggest a law of the form

$$\log(-\frac{dC}{dt}) = \log(k) + \alpha \log(C)$$

where C is concentration and t is time.

This is equivalent to

$$\begin{aligned} -\frac{dC}{dt} &= k \cdot C^{\alpha} \\ -\int C^{-\alpha}\, dC &= \int k\, dt \\ -\frac{1}{1-\alpha} C^{1-\alpha} &= kt + d \\ \frac{1}{\beta} C^{-\beta} &= kt + d \\ C^{-\beta} &= \beta k t + \beta d \end{aligned}$$

If we know $\beta = \alpha - 1$ (i.e., if we know α), then we can fit a linear model using

```
lm(C^(-1/beta) ~ t)
```

The intercept of such a model will be βd and the slope will be βk, from which we can easily recover d and k.

Alternatively, if we know $d = 0$ (i.e., if we know that $C = 0$ when $t = 0$), then we can use

$$\begin{aligned} \log(C^{-\beta}) = -\beta \log(C) &= \log(\beta k t) = \log(\beta k) + \log t \\ \log(C) &= -\frac{\log(\beta k)}{\beta} - \frac{1}{\beta} \log t \end{aligned}$$

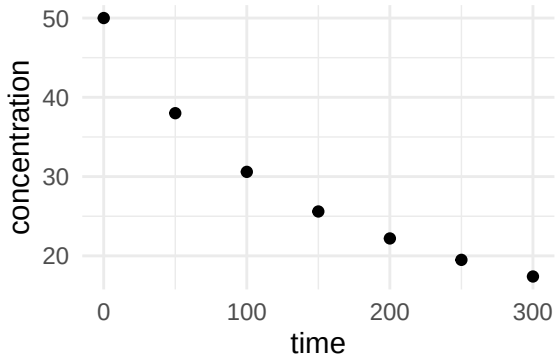
Now if we fit a model of the form

```
lm(C ~ log(t))
```

the intercept will be $\frac{-\log(\beta k)}{\beta}$ and the slope will be $\frac{-1}{\beta}$. From this we can solve for k and β.

Example 8.7.2. Continuing the previous example, we will fit the following data

```
Concentration <- data.frame(
  time = c(0, 50, 100, 150, 200, 250, 300),               # minutes
  concentration = c(50, 38, 30.6, 25.6, 22.2, 19.5, 17.4) # mol/dm^3 * 10^3
)
gf_point(concentration ~ time, data=Concentration)
```



under the assumption that $\alpha = 2$, so $\beta = 1$. In this case, our relationship becomes

$$\frac{1}{C} = -kt - d\,.$$

We can now fit a model and see how well it does.

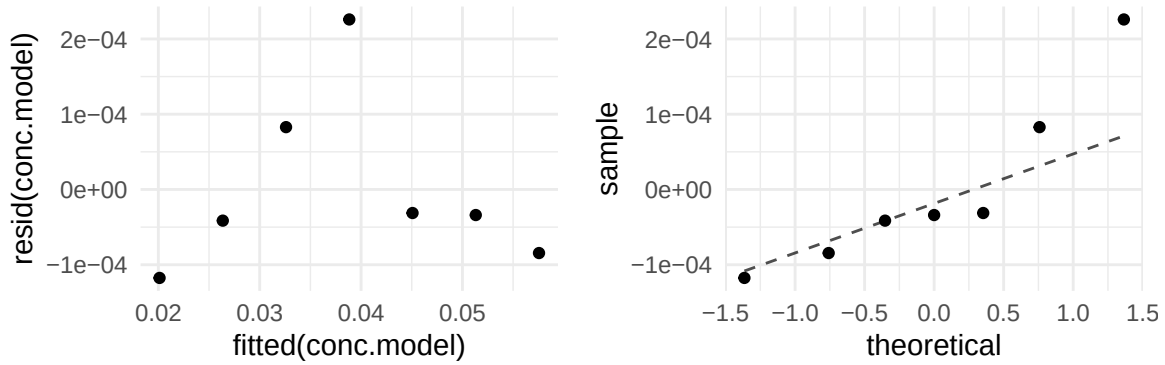
```
conc.model <- lm(1/concentration ~ time, data = Concentration)
summary(conc.model)


##
## Call:
## lm(formula = 1/concentration ~ time, data = Concentration)
##
## Residuals:
##         1         2         3         4         5         6         7
## -1.18e-04 -4.14e-05  8.28e-05  2.26e-04 -3.13e-05 -3.40e-05 -8.45e-05
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.01e-02   8.76e-05     230  3.0e-11
## time        1.25e-04   4.86e-07     257  1.7e-11
##
## Residual standard error: 0.000129 on 5 degrees of freedom
## Multiple R-squared:     1,Adjusted R-squared:     1
## F-statistic: 6.59e+04 on 1 and 5 DF,  p-value: 1.7e-11


confint(conc.model)
```

```
##                  2.5 %   97.5 %
## (Intercept) 0.0198923 0.020343
## time        0.0001235 0.000126
```

This provides estimates for the intercept $-\beta d$ and the slope $-\beta k$ of our model. We can divide by $-\beta$ to obtain estimates for d and k.

Of course, we should always look to see whether the fit is a good fit.

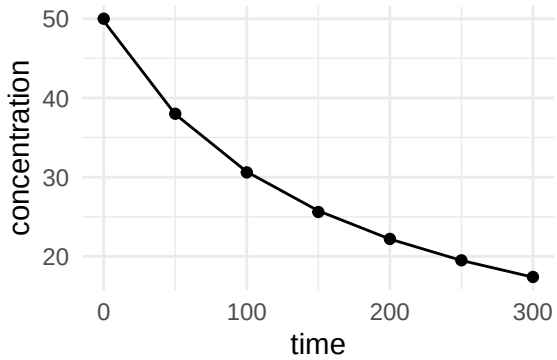
```
gf_point(resid(conc.model) ~ fitted(conc.model))
gf_qq(~resid(conc.model)) %>% gf_qqline()
```

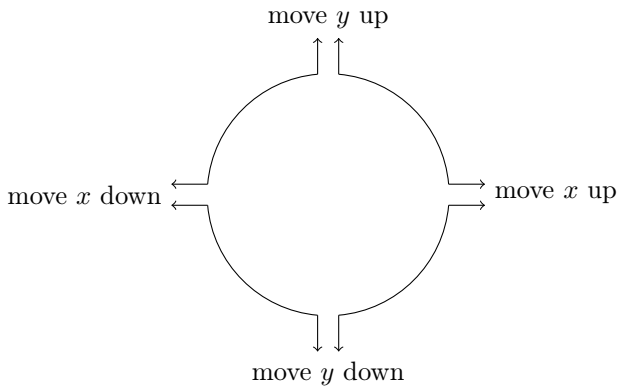


Notice that these residuals are very small relative to the values for concentration. (We can see this from the vertical scale of the plot and also from the small value for residual standard error in the summary output.) The shape of the residual plot would be more disturbing if the magnitudes were larger and if there were more data. As is, even if there is some systematic problem, it is roughly five orders of magnitude smaller than our concentration measurements, which likely can't be measured to that degree of accuracy.

If we want to show the fit on top of the original data, we must remember to untransform the response, since the model we fitted is a model for $1/C$, but we want to show a model for C:

```
gf_point( concentration ~ time, data = Concentration ) %>%
gf_line( 1/fitted(conc.model) ~ time, data = Concentration)
```





ladder of re-expression

power	transformation
⋮	⋮
3	$x \mapsto x^3$
2	$x \mapsto x^2$
1	$x \mapsto x$
$\frac{1}{2}$	$x \mapsto \sqrt{x}$
0	$x \mapsto \log(x)$
-1	$x \mapsto \frac{1}{x}$
-2	$x \mapsto \frac{1}{x^2}$
⋮	⋮

Figure 8.1: Bulge rules and ladder of re-expression.

8.7.3 The Ladder of Re-expression

Sometimes we have data for which there is no theory (yet) to suggest the form of a model. In such a case, we may let the data help suggest a model. If we find a model that fits well, we can return to the question of whether there is an explanation for that type of model.

In the 1970s, Mosteller and Tukey introduced what they called the **ladder of re-expression** and **bulge rules** [**?**, MT77] that can be used to suggest an appropriate transformation to improve the fit when the relationship between two variables (x and y in our examples) is monotonic and has a single bend. Their idea was to apply a power transformation to x or y or both – that is, to work with x^a and y^b for an appropriate choice of a and b. Tukey called this ordered list of transformations the *ladder of re-expression*. The identity transformation has power 1. The logarithmic transformation is a special case and is included in the list associated with a power of 0. The direction of the required transformation can be obtained from Figure 8.1, which shows four bulge types, represented by the curves in each of the four quadrants. A bulge can potentially be straightened by applying a transformation to one or both variables, moving up or down the ladder as indicated by the arrows. More severe bulges require moving farther up or down the ladder. A curve bulging in the same direction as the one in the first quadrant of Figure 8.1, for example, might be straightened by moving up the ladder of transformations for x or y (or both), while a curve like the one in the second quadrant, might be straightened by moving up the ladder for y or down the ladder for x.

This method focuses primarily on transformations designed to improve the overall fit. The resulting models may or may not have a natural, or obvious, interpretation. These transformations also affect the shape of the distributions of the explanatory and response variables and, more importantly, of the residuals from the linear model (see Exercise 8.5). When several different transformations lead to reasonable linear fits, these other factors may lead us to prefer one over another.

Example 8.7.3. Q. The scatterplot in Figure 8.2 shows a curved relationship between x and y. What transformations of x and y improve the linear fit?

A. This type of bulge appears in quadrant IV of Figure 8.1, so we can hope to improve the fit by moving up the ladder for x or down the ladder for y. As we see in Figure 8.3, the fit generally improves as we move down and to the right – but not too far, lest we over-correct. A log-transformation of the response ($a = 1$, $b = 0$) seems

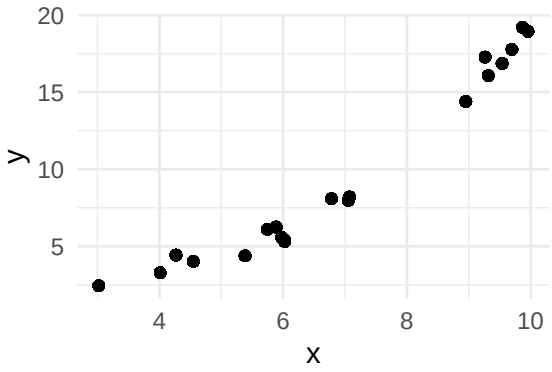


Figure 8.2: A scatterplot illustrating a non-linear relationship between x and y.

to be especially good in this case. Not only is the resulting relationship quite linear, but the residuals appear to have a better distribution as well.

Example 8.7.4. Some physics students conducted an experiment in which they dropped steel balls from various heights and recorded the time until the ball hit the floor. We begin by fitting a linear model to this data.

```
ball.model <- lm(time ~ height, data = BallDrop)
summary(ball.model)
```

```
##
## Call:
## lm(formula = time ~ height, data = BallDrop)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020011 -0.008938  0.000162  0.008202  0.018652
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.19024    0.00430    44.2   <2e-16
## height       0.25184    0.00552    45.7   <2e-16
##
## Residual standard error: 0.0101 on 28 degrees of freedom
## Multiple R-squared:  0.987,Adjusted R-squared:  0.986
## F-statistic: 2.08e+03 on 1 and 28 DF,  p-value: <2e-16
```

```
gf_point(time~height,data=balldrop) %>% gf_lm()
```

```
## Error in gf_ingredients(formula = gformula, data = data, gg_object = object, :  object 'balldrop'
not found
```

```
gf_point(resid(ball.model) ~ fitted(ball.model))
```

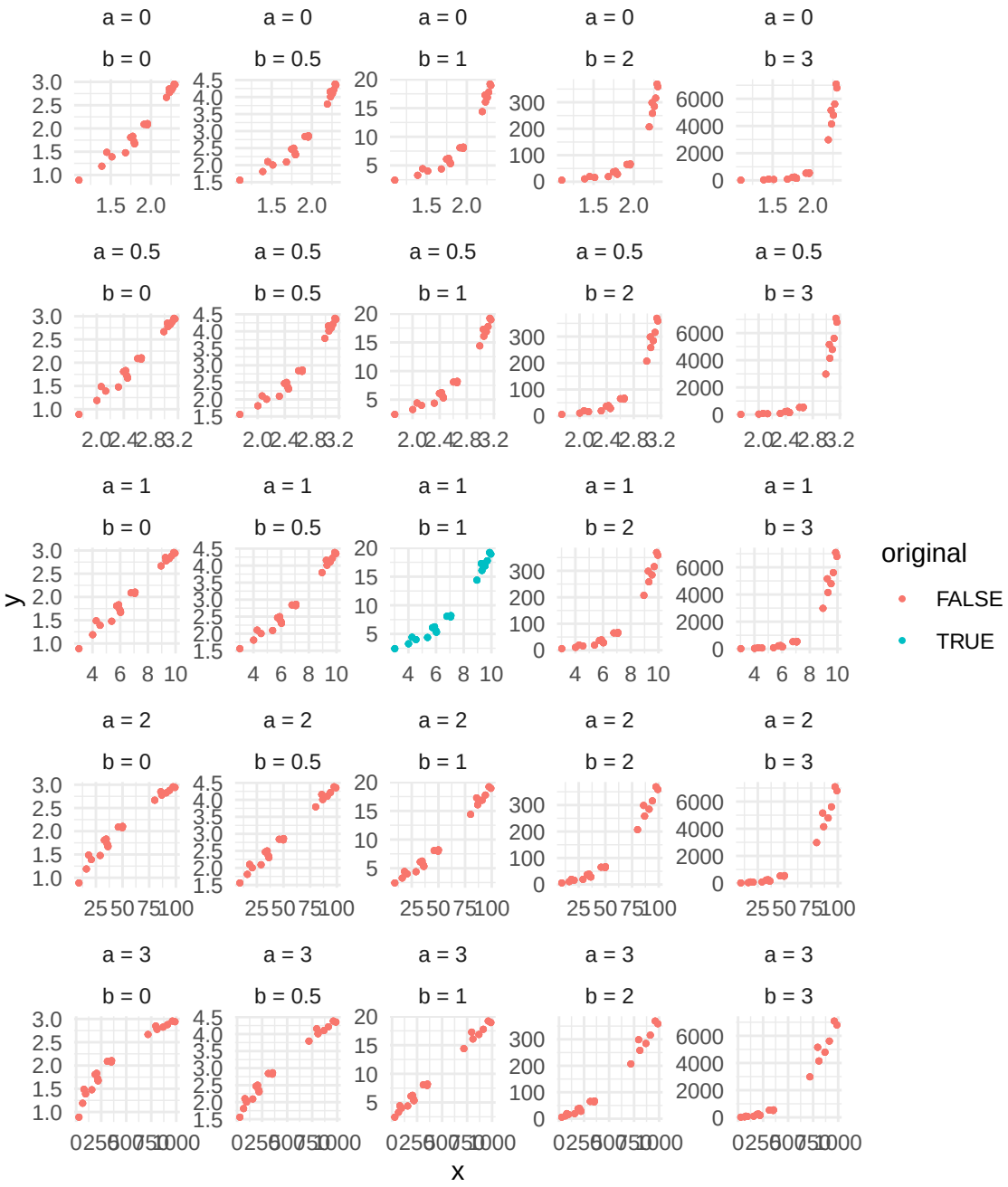
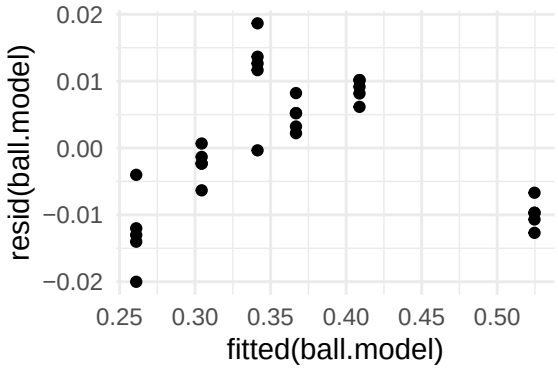


Figure 8.3: Using the ladder of re-expression to find a better fit.



At first glance, the large value of r^2 and the reasonably good fit in the scatterplot might leave us satisfied that we have found a good model. But a look at the residual plot reveals a clear curvilinear pattern in this data. A knowledgeable physics student knows that (ignoring air resistance) the time should be proportional to the *square root* of the height. This transformation agrees with Tukey's ladder of re-expression, which suggests moving down the ladder for `height` or up the ladder for `time`.

```
<<<<<<< HEAD:Stat241-NonLinear.Rnw
ball.modelT <- lm(time ~ sqrt(height), data = BallDrop)
summary(ball.modelT)
plotModel(ball.modelT)
mplot(ball.modelT, w = 1)
=======
ball.modelT <- lm(time ~ sqrt(height), data=balldrop)
summary(ball.modelT)
gf_point(time ~ sqrt(height), data=balldrop) %>%
  gf_lm(time~sqrt(height), data=balldrop)
gf_point(resid(ball.modelT) ~ fitted(ball.modelT))
>>>>>>> c25168dd6c03b5958e6d874fffc45a873a27e0d5:Math241-NonLinear.Rnw


## Error:  <text>:1:1:  unexpected input
## 1:  «
##     ^
```

This model does indeed fit better, but the residual plot indicates that there may be some inaccuracy in the measurement of the height. In this experiment, the apparatus was set up once for each height and then several observations were made. So any error in this set-up affected all time measurements for that height in the same way. This could explain why the residuals for each height are clustered the way they are since it violates the assumption that the errors are *independent*. (See Example 8.7.5 for a simple attempt to deal with this problem.)

Example 8.7.5. One simple way to deal with the lack of independence in the previous example is to average all the readings made at each height. (This works reasonably well in our example because we have nearly equal numbers of observations at each height.) We pay for this data reduction in a loss of degrees of freedom, but it may be easier to justify that the errors in average times at each height are independent (if we believe that the errors in the height set-up are independent and not systematic).

```
balldropavg <- balldrop %>%
  group_by(height) %>%
  dplyr::summarize(time = mean(time))

## Error in group_by(., height):  object 'balldrop' not found

ball.modelA <- lm(time ~ sqrt(height),balldropavg)
```

```
## Error in is.data.frame(data):  object 'balldropavg' not found

summary(ball.modelA)

## Error in h(simpleError(msg, call)):  error in evaluating the argument 'object' in selecting
a method for function 'summary':  object 'ball.modelA' not found

gf_point(time~height, data=balldropavg) %>% gf_lm()

## Error in gf_ingredients(formula = gformula, data = data, gg_object = object, :  object 'balldropavg'
not found

gf_point(resid(ball.modelA) ~ fitted(ball.modelA))

## Error in fitted(ball.modelA): object 'ball.modelA' not found
```

Using a square root transformation on averaged `height` measurements in the `BallDrop` data gives a similar fit but a very different residual plot. The interpretation of this model is also different.

Notice that the parameter estimates are essentially the same as in the preceding example. The estimate for σ has decreased some. This makes sense since we are now estimating the variability in *averaged* measurements rather than in individual measurements.

Of course, we've lost a lot of degrees of freedom, and as a result, the standard error for our parameter estimate is about twice as large as before. This might have been different; had the mean values fit especially well, our standard error might have been smaller despite the reduced degrees of freedom.

One disadvantage of the data reduction is that it is hard to interpret the residuals (because there are fewer of them). At first glance there appears to be a downward trend in the residuals, but this is largely driven by the fact that the largest residual happened to be for the smallest fit.

Example 8.7.6. Q. Rex Boggs of Glenmore State High School in Rockhampton, Queensland, had an interesting hypothesis about the rate at which bar soap is used in the shower. He writes:

> I had a hypothesis that the daily weight of my bar of soap [in grams] in my shower wasn't a linear function, the reason being that the tiny little bar of soap at the end of its life seemed to hang around for just about ever. I wanted to throw it out, but I felt I shouldn't do so until it became unusable. And that seemed to take weeks.
>
> Also I had recently bought some digital kitchen scales and felt I needed to use them to justify the cost. I hypothesized that the daily weight of a bar of soap might be dependent upon surface area,

> and hence would be a quadratic function ....
>
> The data ends at day 22. On day 23 the soap broke into two pieces and one piece went down the plughole.

The data indicate that although Rex showered daily, he failed to record the weight for some of the days.

What do the data say in regard to Rex's hypothesis?

A. Rex's assumption that weight should be a (quadratic) function of time does not actually fit his intuition. His intuition corresponds roughly to the differential equation

$$\frac{\partial t}{\partial W} = kW^{2/3},$$

for some negative constant k since the rate of change should be proportional to the surface area remaining. (We are assuming that the bar shrinks in such a way that its shape remains proportionally unaltered.) Solving this equation (by separation of variables) gives

$$W^{1/3} = kt + C.$$

We can fit untransformed and transformed models (`weight^(1/3) ~ day`) to this data and compare.

```
soap.model1 <- lm(weight ~ day, data = Soap)
summary(soap.model1)
```

```
##
## Call:
## lm(formula = weight ~ day, data = Soap)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.244 -1.295  0.308  1.394  5.504
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  123.141      1.382    89.1   <2e-16
## day           -5.575      0.107   -52.2   <2e-16
##
## Residual standard error: 2.95 on 13 degrees of freedom
## Multiple R-squared:  0.995,Adjusted R-squared:  0.995
## F-statistic: 2.72e+03 on 1 and 13 DF,  p-value: <2e-16
```

The scatterplot in Figure 8.4 (darker line) indicate that the untransformed model is already a good fit.

```
soap.model2 <- lm(I(weight^(1/3)) ~ day, data = Soap)
summary(soap.model2)
```

```
##
## Call:
## lm(formula = I(weight^(1/3)) ~ day, data = Soap)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.311 -0.137  0.016  0.150  0.201
##
## Coefficients:
```

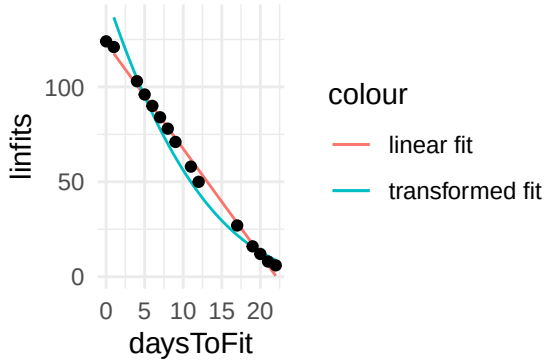


Figure 8.4: Comparing untransformed (darker) and transformed (lighter) fits to soap use data.

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.29771    0.08381    63.2  < 2e-16
## day         -0.14698    0.00648   -22.7  7.7e-12
##
## Residual standard error: 0.179 on 13 degrees of freedom
## Multiple R-squared:  0.975,Adjusted R-squared:  0.973
## F-statistic:  515 on 1 and 13 DF,  p-value: 7.67e-12
```

The transformed model in this case actually fits worse. The higher value of r^2 for the untransformed model is an indication that the untransformed model explains a larger proportion of the variance in soap weights. It is left as an exercise for you to examine diagnostic plots of the model residuals in both cases; you should see that neither one looks markedly better than the other. (There is perhaps an issue with a small amount of non-independence, or correlation over time, of the residuals; we might expect that with data collected over time. However, the dataset is so small that it is hard to tell for sure if the problem is real and worth worrying about.) Figure 8.4 shows a scatterplot with both fits. The data do not support Rex's assumption that a transformation is necessary. The scatterplot and especially the residual plots both show that the residuals are mostly positive near the ends of the data and negative near the center. Part of this is driven by a flattening of the pattern of data points near the end of the measurement period. Perhaps as the soap became very small, Rex used slightly less soap than when the soap was larger. Exercise 8.2 asks you to remove the last few observations and see how that affects the models.

Finally, since a linear model appears to fit at least reasonably well (but see Exercise 8.2), we can give a confidence interval for β_1, the mean amount of soap Rex uses each shower.

```
confint(soap.model1)
```

```
##                2.5 %  97.5 %
## (Intercept) 120.155 126.127
## day          -5.806  -5.344
```

8.8 Nonlinear Least Squares

Another approach to non-linear relationships is called **nonlinear least squares** or **nonlinear regression**. In this approach, instead of attempting to transform the relationship until it becomes linear, we fit a nonlinear function by minimizing the the sum of the squared residuals relative to that (paramterized) nonlinear function

(form). That is, our model now becomes

$$y = f(x) + \varepsilon$$

where f may be any parameterized function.

The R function for fitting these models is `nls()`. This function works much like `lm()`, but there are some important differences:

1. Because the model does not have to be linear, we have to use a more verbose description of the model.
2. Numerical optimization is used to fit the model, and the algorithm used needs to be given a reasonable starting point for its search. Specifying this starting point simultaneously lets R know what the parameters of the model are. (Each quantity with a starting value is considered a parameter, and the algorithm will adjust all the parameters looking for the best fit – i.e., the smallest MSE (and hence also the smallest SSE and RMSE).

Let's illustrate with an example.

Example 8.8.1. Returning to the ball dropping experiment, let's fit

$$\texttt{time} = \alpha_0 + \alpha_1 \sqrt{\texttt{height}} \tag{8.1}$$

using nonlinear least squares.

```
nls.model <- nls(time ~ alpha0 + alpha1 * sqrt(height),
                 data = BallDrop,
                 start = list(alpha0 = 0, alpha1 = 1))
```

Notice how the model formula compares with the formula in (8.1). The starting point for the algorithm is specified with `start = list(alpha0 = 0, alpha1 = 1)`, which also declares the parameters to be fit.

We can obtain the coefficients of the fitted model with

```
nls.model
```

```
## Nonlinear regression model
##   model: time ~ alpha0 + alpha1 * sqrt(height)
##    data: BallDrop
## alpha0 alpha1
## 0.0161 0.4308
##  residual sum-of-squares: 0.000765
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.11e-07
```

or

```
coef(nls.model)
```

```
##  alpha0  alpha1
## 0.01608 0.43080
```

A more complete summary can be obtained by

```
summary(nls.model)
```

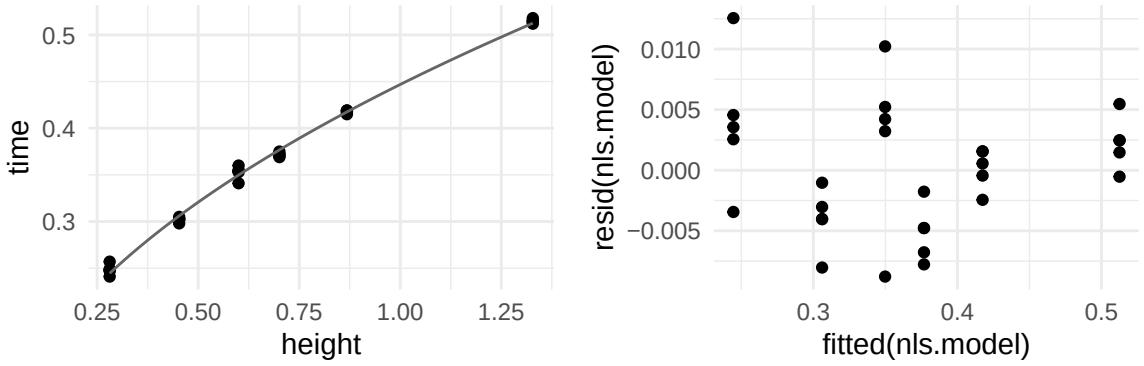
```
##
## Formula: time ~ alpha0 + alpha1 * sqrt(height)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## alpha0  0.01608    0.00408    3.94    5e-04
## alpha1  0.43080    0.00486   88.58   <2e-16
##
## Residual standard error: 0.00523 on 28 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.11e-07
```

We can restrict our attention to the coefficients table with

```
coef(summary(nls.model))
```

```
##        Estimate Std. Error t value  Pr(>|t|)
## alpha0  0.01608   0.004084   3.937 4.976e-04
## alpha1  0.43080   0.004863  88.580 7.732e-36
```

```
f <- makeFun(nls.model)
gf_point(time ~ height, data = BallDrop) %>%
gf_fun(f(height) ~ height, color = 'gray40')
gf_point(resid(nls.model) ~ fitted(nls.model))
```



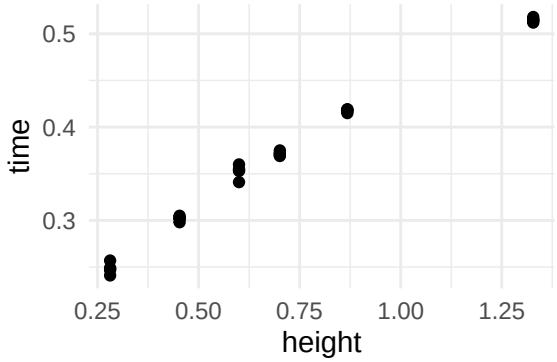
We can compare this to the ordinary least squares model by plotting both together on the same plot.

```
lm.model <- lm(time ~ sqrt(height), data = BallDrop)
g <- makeFun(lm.model)
gf_point(time ~ height, data = BallDrop)
gf_fun(f(height) ~ height, color = 'gray40', size = 3)
```

```
## Error in data.frame(x = xlim):  argument "xlim" is missing, with no default
```

```
gf_fun(g(height) ~ height, color = 'red', size = 1, linetype = 2)
```

```
## Error in data.frame(x = xlim):  argument "xlim" is missing, with no default
```



In this particular case, there is very little difference between the two models, but this is not always the case.

```
coef(nls.model)


##  alpha0  alpha1
## 0.01608 0.43080


coef(lm.model)


##  (Intercept) sqrt(height)
##      0.01608      0.43080
```

Example 8.8.2. Here is example where we fit a different model to the `balldrop` data, namely

$$\texttt{time} = \alpha * \texttt{height}^p$$

```
power.model <- nls(time ~ alpha * height^power, data = BallDrop,
                   start = c(alpha = 1, power = .5))
coef(summary(power.model))


##       Estimate Std. Error t value  Pr(>|t|)
## alpha   0.4472   0.001343  333.09 6.333e-52
## power   0.4797   0.005805   82.63 5.388e-35
```

A power law can also be fit using `lm()` by using a log-log transformation.

```
power.model2 <- lm(log(time) ~ log(height), data = BallDrop)
coef(summary(power.model2))


##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -0.8076   0.004330 -186.49 7.101e-45
## log(height)   0.4719   0.006425   73.45 1.431e-33
```

Again, the parameter estimates (and uncertainties) are very similar. Recall that to compare our intercept in the second model to the α value in the first model, we must untransform:

```
exp(coef(power.model2)[1])


## (Intercept)
##      0.4459
```

We can use the delta method to estimate the uncertainty. Since $\frac{d}{dx}e^x = e^x$ the uncertainty is approximately

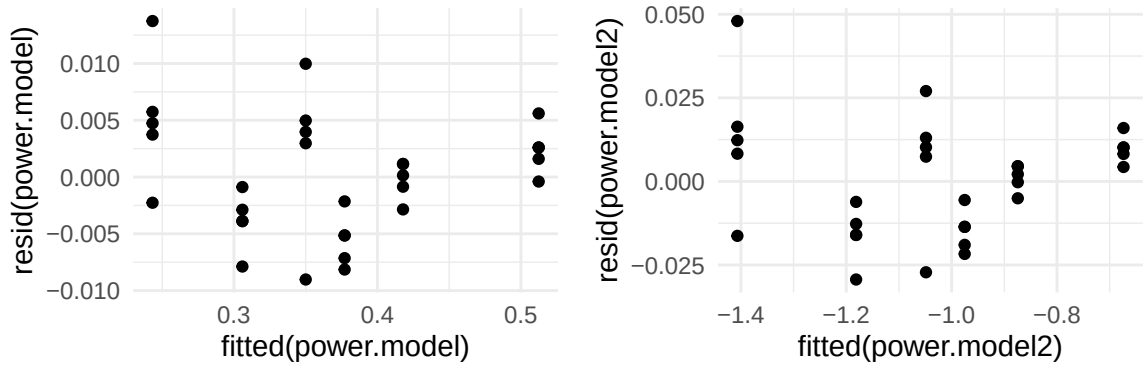
$$0.446 \cdot 0.004 = 0.002$$

Example 8.8.3. In addition to comparing estimated parameters and their uncertainties, we should always look at the residuals of our model. For both the linear regression and the nonlinear least squares models, the assumption is that the error terms are independent, normally distributed, and have a common standard deviation. From the plots below we see

1. The nonlinear least squares model is a better match for these assumptions than the linear regression model.

2. Both models reveal a lack of independence – at a given height, the residuals move up or down as a cluster as was discussed in the previous section. Neither model is designed to handle this flaw in the design of the experiment.

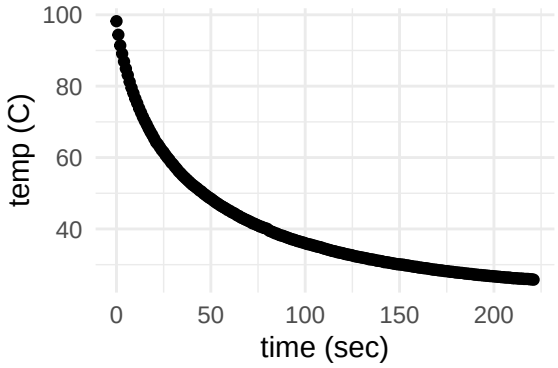
```
gf_qq(resid(power.model), main = "model 1")


## Error:  Invalid formula type for gf_qq.


gf_qq(resid(power.model2))


## Error:  Invalid formula type for gf_qq.


gf_point(resid(power.model) ~ fitted(power.model))
gf_point(resid(power.model2) ~ fitted(power.model2))
```



Now let's take a look at an example where we need the extra flexibilty of the nonlinear least squares approach.

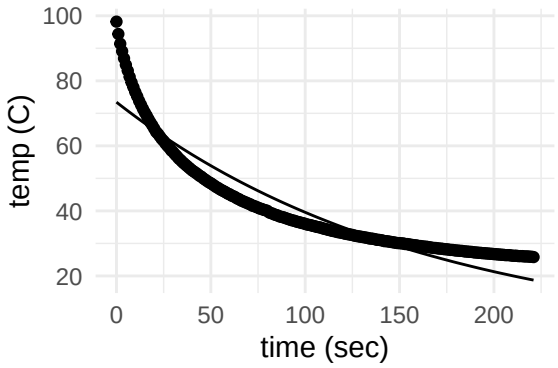
Example 8.8.4. A professor at Macalester College put hot water in a mug and recorded the temperature as it cooled. Let's see if we can fit a reasonable model to this data

```
gf_point(temp ~ time, data = CoolingWater, ylab = "temp (C)", xlab = "time (sec)")
```



Our first guess might be some sort of exponential decay

```
cooling.model1 <-
  nls(temp ~ A * exp( -k * time), data = CoolingWater,
      start = list(A = 100, k = 0.1))
f1 <- makeFun(cooling.model1)
gf_point(temp ~ time, data = CoolingWater, xlim = c(-50,300), ylim = c(0,110),
        ylab = "temp (C)", xlab = "time (sec)") %>%
gf_fun(f1(time) ~ time)
```

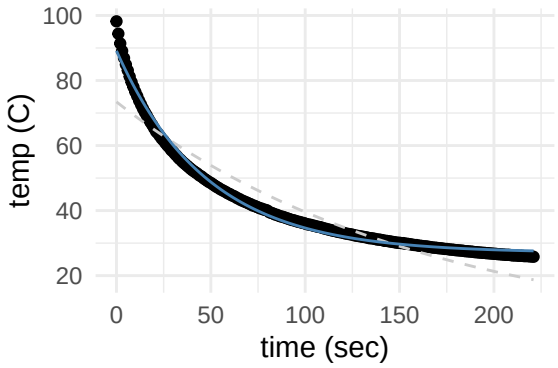


That doesn't fit very well, and there is a good reason. The model says that eventually the water will freeze because

$$\lim_{t \to \infty} Ae^{-kt} = 0$$

when $k > 0$. But clearly our water isn't going to freeze sitting on a lab table. We can fix this by adding in an offset to account for the ambient temperature:

```
cooling.model2 <- nls(temp ~ ambient + A * exp(k * (1+time)), data = CoolingWater,
                      start = list(ambient = 20, A = 80, k = -.1) )
f2 <- makeFun(cooling.model2)
gf_point(temp ~ time, data = CoolingWater, xlim = c(-50,300), ylim = c(0,110),
        ylab = "temp (C)", xlab = "time (sec)") %>%
gf_fun(f1(time) ~ time, linetype = 2, color = "gray80") %>%
gf_fun(f2(time) ~ time, color = "steelblue")
```



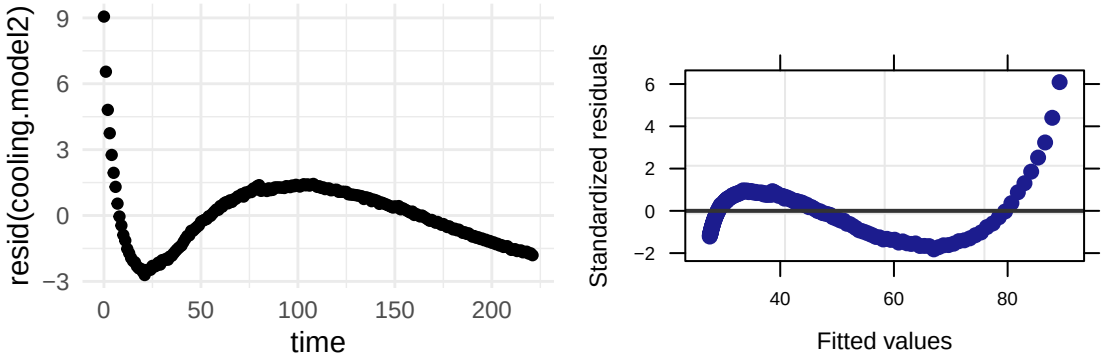
This fits much better. Furthermore, this model can be derived from a differential equation

$$\frac{dT}{dt} = -k(T_0 - T_{\text{ambient}}) \ ,$$

known as Newton's Law of Cooling.

Let's take a look at the residual plot

```
gf_point(resid(cooling.model2) ~ time, data = CoolingWater)
plot(cooling.model2, which = 1)
```



Hmm. These plots show a clear pattern and very little noise. The fit doesn't look as good when viewed this way. It suggests that Newton's Law of Cooling does not take into account all that is going on here. In particular, there is a considerable amount of evaporation (at least at the beginning when the water is warmer). More complicated models that take this into account can fit even better. For a discussion of a model that includes evaporation, see `http://stanwagon.com/public/EvaporationPortmannWagonMiER.pdf`.[1]

8.8.1 Choosing Between Linear and Non-linear Models

So how do we choose between linear and non-linear models? Let's enumerate some of the differences between them:

1. Some models cannot be expressed as linear models, even after transformations.

   In this case we only have one option, the non-linear model.

[1]The model with evaporation adds another complication in that the resulting differential equation cannot be solved algebraically, so there is no algebraic formula to fit with `nls()`. But the method of least squares can still be used by creating a parameterized numerical function that computes the sum of squares and using a numerical minimizer to find the optimal parameter values. Since the use of numerical differential equation solvers is a bit beyond the scope of this course, we'll leave that discussion for another day.

2. Linear models can be fit quickly and accurately without numerical optimization algorithms because they satisfy nice linear algebra properties.

   The use of numerical optimizers in non-linear least squares models makes them subject to potential problems with the optimizers. They may not converge, may converge to the wrong thing, or convergence may depend on choosing an appropriate starting point for the search.

3. The two types of models make different assumptions about the error terms.

   In particular, when we apply transformations to achieve a linear model, those transformations often affect the distribution of the error terms as well. For example, if we apply a log-log transformation to fit a power law, then the model is

$$\begin{aligned} \log(y) &= \beta_0 + \beta_1 \log(x) + \varepsilon \\ y &= e^{\beta_0} x^{\beta_1} e^{\varepsilon} \\ y &= \alpha x^{\beta_1} e^{\varepsilon} \end{aligned}$$

   So the errors are multiplicative rather than additive and they have a normal distribution *after* applying the logarithmic transformation. This implies that the relative errors should be about the same magnitude rather than the absolute errors.

   This is potentially very different from the nonlinear model where the errors are additive:

$$y = \alpha x^{\beta} + \varepsilon$$

   Plots of residuals vs. fits and qq-plots for residuals can help us diagnose whether the assumptions of a model are reasonable for a particular data set.

4. Linear models provide an easy way to produce confidence intervals for a mean response or an individual response.

   The models fit using `nls()` do not have this capability.

8.9 Exercises

8.1 In Example 8.7.4, we applied a square root transformation to the height. Is there another transformation that yields an even better fit?

8.2 Remove the last few days from the `Soap` data set and refit the models in Example 8.7.6. How much do things change? Do the residuals look better, or is there still some cause for concern?

8.3 For each of the following relationships between a response y and an explanatory variable x, if possible find a pair of transformations f and g so that $g(y)$ is a linear function of $f(x)$:

$$g(y) = \beta_0 + \beta_1 f(x) \ .$$

For example, if $y = ae^{bx}$, then $\log(y) = \log(a) + bx$, so $g(y) = \log(y)$, $f(x) = x$, $\beta_0 = \log(a)$, and $\beta_1 = b$.

a) $y = ab^x$.

b) $y = ax^b$.

c) $y = \frac{1}{a+bx}$.

d) $y = \frac{x}{a+bx}$.

e) $y = ax^2 + bx + c$.

f) $y = \dfrac{1}{1+e^{a+bx}}$.

g) $y = \dfrac{100}{1+e^{a+bx}}$.

8.4 What happens to the role of the error terms (ε) when we transform the data? For each transformation from Exercise 8.3, start with the form

$$g(y) = \beta_0 + \beta_1 f(x) + \varepsilon$$

and transform back into a form involving the untransformed y and x to see how the error terms are involved in these transformed linear regression models.

It is important to remember that when we fit a linear model to transformed data, the usual assumptions of the model are that the errors in the (transformed) linear form are additive and normally distributed. The errors may appear differently in the untransformed relationship.

8.5 The transformations in the ladder of re-expression also affects the shape of a distribution.

a) If a distribution is symmetric, how does the shape change as we move up the ladder?

b) If a distribution is symmetric, how does the shape change as we move down the ladder?

c) If a distribution is left skewed, in what direction should we move to make the distribution more symmetric?

d) If a distribution is right skewed, in what direction should we move to make the distribution more symmetric?

8.6 By attaching a heavy object to the end of a string, it is easy to construct pendulums of different lengths. Some physics students did this to see how the period (time in seconds until a pendulum returns to the same location) depends on the length (in meters) of the pendulum. The students constructed pendulums of lengths varying from 10 cm to 16 m and recorded the period length (averaged over several swings of the pendulum). The resulting data are in the `Pendulum` data set in the `fastR2` package.

a) Fit a power law to this data using a transformation and a linear model. How well does the power law fit? What is the estimated power in the power law based on this model?

b) Fit a power law to this data using a nonlinear model. How well does the power law fit? What is the estimated power in the power law based on this model?

c) Compare residual plots and normal-quantile plots for the residuals for the two models. How do the models compare in this regard?

8.7 The `Pressure` data set contains data on the relation between temperature in degrees Celsius and vapor pressure in millimeters (of mercury). With temperature as the predictor and pressure as the response, use transformations or nonlinear models as needed to obtain a good fit. Make a list of all the models you considered and explain how you chose your best model. What does your model say about the relationship between pressure and temperature?

8.8 The `cornnit` data set in the package `faraway` contains data from a study investigating the relationship between corn yield (bushels per acre) and nitrogen (pounds per acre) fertilizer application in Wisconsin. Using nitrogen as the predictor and corn yield as the response, use transformations (if necessary) to obtain a good fit. Make a list of all the models you considered and explain how you chose your best model.

8.9 The data set `ACTgpa` (in the `fastR2` package) contains the ACT composite scores and GPAs of some randomly selected seniors at a Midwest liberal arts college.

a) Give a 95% confidence interval for the mean ACT score of seniors at this school.

b) Give a 95% confidence interval for the mean GPA of seniors at this school.

c) Use the data to estimate with 95% confidence the average GPA for all students who score 25 on the ACT.

d) Suppose you know a high school student who scored 30 on the ACT. Estimate with 95% confidence his GPA as a senior in college.

e) Are there any reasons to be concerned about the analyses you have just done? Explain.

8.10 In the absence of air resistance, a dropped object will continue to accelerate as it falls. But if there is air resistance, the situation is different. The drag force due to air resistance depends on the velocity of an object and operates in the opposite direction of motion. Thus as the object's velocity increases, so does the drag force until it eventually equals the force due to gravity. At this point the net force is 0 and the object ceases to accelerate, remaining at a constant velocity called the terminal velocity.

Now consider the following experiment to determine how terminal velocity depends on the mass (and therefore on the downward force of gravity) of the falling object. A helium balloon is rigged with a small basket and just the right ballast to make it neutrally buoyant. Mass is then added and the terminal velocity is calculated by measuring the time it takes to fall between two sensors once terminal velocity has been reached.

The `Drag` data set (in the `fastR2` package) contains the results of such an experiment conducted by some undergraduate physics students. Mass is measured in grams and velocity in meters per second. (The distance between the two sensors used for determining terminal velocity is given in the `height` variable.)

By fitting models to this data, determine which of the following "drag laws" matches the data best:

- Drag is proportional to velocity.
- Drag is proportional to the square of velocity.
- Drag is proportional to the square root of velocity.
- Drag is proportional to the logarithm of velocity.

8.11 Construct a plot that reveals a likely systematic problem with the `Drag` (see Exercise 8.10) data set. Speculate about a potential cause for this.

8.12 Exercise 8.11 suggests that some of the data should be removed before analyzing the `Drag` data set. Redo Exercise 8.10 after removing this data.

8.13 The `Spheres` data set (in the `fastR2` package) contains measurements of the diameter (in meters) and mass (in kilograms) of a set of steel ball bearings. We would expect the mass to be proportional to the cube of the diameter. Fit a model and see if the data reflect this.

8.14 The `Spheres` data set (in the `fastR2` package) contains measurements of the diameter (in meters) and mass (in kilograms) of a set of steel ball bearings. We would expect the mass to be proportional to the cube of the diameter. Using appropriate transformations fit two models: one that predicts mass from diameter and one that predicts diameter from mass. How do the two models compare?

8.15 The `Utilities` data set has information from utilities bills at a Minnesota residence. Fit a linear model that predicts `thermsPerDay` from `temp`.

a) What observations should you remove from the data before doing the analysis? Why?

b) Are any transformations needed?

c) How happy are you with the fit of your model? Are there any reasons for concern?

d) Interpret your final model (even if it is with some reservations listed in part c)). What does it say about the relationship between average monthly temperature and the amount of gas used at this residence? What do the parameters represent?

9 Hypothesis Testing

This chapter is concerned with statistical hypothesis testing, and how we can use it to make inferences and draw conclusions from data about questions of scientific interest.

9.1 Experimental Design in Statistics

Before we begin to talk about hypothesis testing, let's review the general process of designing and carrying out a statistical experiment.

1. Determine the question of interest.

   Just what is it we want to know? It may take some effort to make a vague idea precise. The precise questions may not exactly correspond to our vague questions, and the very exercise of stating the question precisely may modify our question. Sometimes we cannot come up with any way to answer the question we really want to answer, so we have to live with some other question that is not exactly what we wanted but is something we can study and will (we hope) give us some information about our original question.

2. Determine the **population**.

   Just who or what do we want to know about? For example, are we only interested in one specific person, or women in general, or all women, or all people? Or, are we interested in the energy efficiency of one particular device, or all the machines in a certain factory, or all machines of a certain type, or all machines of a certain class, or all factories in a certain industry?

3. Select **measurements**.

   We are going to need some data. We get our data by making some measurements. These might be physical measurements with some device (like a ruler or a scale). But there are other sorts of measurements too, like the answer to a question on a form. Sometimes it is tricky to figure out just what to measure. (How do we measure happiness or intelligence, for example?) Just how we do our measuring will have important consequences for the subsequent statistical analysis. The recorded values of these measurements are called **variables** (because the values vary from one individual to another).

4. Determine the **sample**.

   Usually we cannot measure every individual in our population; we have to select some to measure. But how many and which ones? These are important questions that must be answered. Generally speaking, bigger is better, but it is also more expensive. Moreover, no size is large enough if the sample is selected inappropriately.

For example, if we wanted to draw conclusions about energy use across a whole industry, we would have to be careful not to sample from just a single factory, or a single type of manufacturing device. If we wanted to draw conclusions about all people, we would have to be careful not to study only male college students. The sample should be a random selection from the whole population (or as close as we can get to that standard).

5. Make and record the measurements.

   Once we have the design figured out, we have to do the legwork of data collection. This can be a time-consuming and tedious process. A study of public opinion may require many thousands of phone calls or personal interviews. In a laboratory setting, each measurement might be the result of a carefully performed laboratory experiment.

6. Organize the data.

   Once the data have been collected, it is often necessary or useful to organize them. Data are typically stored in spreadsheets or in other formats that are convenient for processing with statistical packages. Very large data sets are often stored in databases.

   Part of the organization of the data may involve producing graphical and numerical summaries of the data. These summaries may give us initial insights into our questions or help us detect errors that may have occurred to this point.

7. Draw conclusions from data.

   Once the data have been collected, organized, and analyzed, we need to reach a conclusion. What is the answer to our scientific question? Is our idea or hypothesis about the way things work incorrect, or do the data support it? How sure are we about these conclusions?

8. Produce a report.

   Typically the results of a statistical study are reported in some manner. This may be as a refereed article in an academic journal, as an internal report to a company, or as a solution to a problem on a homework assignment. These reports may themselves be further distilled into press releases, newspaper articles, advertisements, and the like. The mark of a good report is that it provides the essential information about each of the steps of the study.

At this point, you may be wondering who the innovative scientist was and what the results of the experiment were. The scientist was R. A. Fisher, who first described this situation as a pedagogical example in his 1925 book on statistical methodology [Fis25]. Fisher developed statistical methods that are among the most important and widely used methods to this day, and most of his applications were biological.

9.2 Coins and Cups

You might also be curious about how the experiment came out. How many cups of tea were prepared? How many did the woman correctly identify? What was the conclusion?

Fisher never says. In his book he is interested in the method, not the particular results. But let's suppose we decide to test the lady with ten cups of tea. We'll flip a coin to decide which way to prepare the cups. If we flip a head, we will pour the milk in first; if tails, we put the tea in first. Then we present the ten cups to the lady and have her state which ones she thinks were prepared each way.

It is easy to give her a score (9 out of 10, or 7 out of 10, or whatever it happens to be). It is trickier to figure out what to do with her score. Even if she is just guessing and has no idea, she could get lucky and get quite a few correct – maybe even all 10. But how likely is that?

Let's try an experiment. I'll flip 10 coins. You guess which are heads and which are tails, and we'll see how you do.

⋮

Comparing with your classmates, we will undoubtedly see that some of you did better and others worse.

Now let's suppose the lady gets 9 out of 10 correct. That's not perfect, but it is better than we would expect for someone who was just guessing. On the other hand, it is not impossible to get 9 out of 10 just by guessing. So here is Fisher's great idea: Let's figure out how hard it is to get 9 out of 10 by guessing. If it's not so hard to do, then perhaps that's just what happened, so we won't be too impressed with the lady's tea tasting ability. On the other hand, if it is really unusual to get 9 out of 10 correct by guessing, then we will have some evidence that she must be able to tell something.

But how do we figure out how unusual it is to get 9 out of 10 just by guessing? We'll learn another method later, but for now, let's just flip a bunch of coins and keep track. If the lady is just guessing, she might as well be flipping a coin.

So here's the plan. We'll flip 10 coins. We'll call the heads correct guesses and the tails incorrect guesses. Then we'll flip 10 more coins, and 10 more, and 10 more, and .... That would get pretty tedious. Fortunately, computers are good at tedious things, so we'll let the computer do the flipping for us using a tool in the `mosaic` package. The `rflip()` function can flip one coin

```
library(mosaic)
rflip()


##
## Flipping 1 coin [ Prob(Heads) = 0.5 ] ...
##
## T
##
## Number of Heads: 0 [Proportion Heads: 0]
```

or a number of coins

```
rflip(10)


##
## Flipping 10 coins [ Prob(Heads) = 0.5 ] ...
##
## T T H H T T H H H H
##
## Number of Heads: 6 [Proportion Heads: 0.6]
```

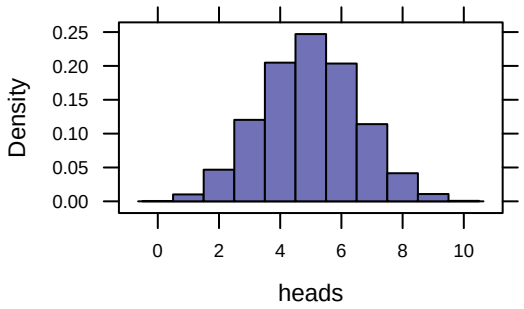
and show us the results.

Typing `rflip(10)` a bunch of times is almost as tedious as flipping all those coins. But it is not too hard to tell R to `do()` this a bunch of times.

```
do(2) * rflip(10)


##    n heads tails prop
## 1 10     6     4  0.6
## 2 10     4     6  0.4
```

Let's get R to `do()` it for us 10,000 times and make a table and a histogram of the results.

```
RandomLadies <- do(10000) * rflip(10)
histogram( ~ heads, data = RandomLadies, width = 1)
```




```
tally( ~ heads, data = RandomLadies)

## heads
##    0    1    2    3    4    5    6    7    8    9   10
##    5  102  467 1203 2048 2470 2035 1140  415  108    7

tally( ~ heads, data = RandomLadies, format = 'percent')

## heads
##     0     1     2     3     4     5     6     7     8     9    10
##  0.05  1.02  4.67 12.03 20.48 24.70 20.35 11.40  4.15  1.08  0.07

tally( ~ heads, data = RandomLadies, format = 'proportion')

## heads
##      0      1      2      3      4      5      6      7      8      9     10
## 0.0005 0.0102 0.0467 0.1203 0.2048 0.2470 0.2035 0.1140 0.0415 0.0108 0.0007
```

You might be surprised to see that the number of correct guesses is exactly 5 (half of the 10 tries) only 25% of the time. But most of the results are quite close to 5 correct. 67% of the results are 4, 5, or 6, for example. And 90% of the results are between 3 and 7 (inclusive). But getting 8 correct is a bit unusual, and getting 9 or 10 correct is even more unusual.

So what do we conclude? It is possible that the lady could get 9 or 10 correct just by guessing, but it is not very likely (it only happened in about 1.2% of our simulations). So *one of two things must be true*:

- The lady got unusually "lucky", or
- The lady is not just guessing.

Although Fisher did not say how the experiment came out, others have reported that the lady correctly identified all 10 cups! [?]

This same reasoning can be applied to answer a wide range of questions that have a similar form. For example, the question of whether dogs can smell cancer could be answered essentially the same way (although it would be a bit more involved than preparing tea and presenting cups to the Lady).

9.3 A General Framework

In statistical hypothesis testing, we can follow the following general procedure. We usually begin with some idea about how the process we are studying should work. For example, we might have a hunch that highway bridges with higher traffic flows are in poorer condition, and therefore merit more frequent repairs. For statistical hypothesis testing, we must translate that "hunch" into a *testable* **null hypothesis**, often called H_0: one that can be demonstrated to be very unlikely in light of our data. In the case of the bridges, a testable null hypothesis might be that bridge condition *does not* depend on traffic flow. Then, if our data shows a strong apparent relationship between condition and traffic, we have evidence to *reject* the null hypothesis, and the data support the idea that there is some relationship between condition and traffic.

Null hypothesis are usually "boring," no-result hypotheses: there is no pattern; there is no relationship between the variables of interest; there is no difference between the two samples of interest. If we can reject the null hypothesis in a certain case, we have some evidence – but NOT proof – that there *is* an interesting pattern in our data, and thus in the population we are trying to draw conclusions about.

The alternative to the null hypothesis is called the alternative hypothesis, often called H_1. The alternative hypothesis, stated most generally, is usually some form of "the null hypothesis is not true" – so there IS a pattern in the data, or a difference between the samples, etc.

hypothesis A statement that can be true or false.

statistical hypothesis A hypothesis about a parameter or parameters.

In our bridge example, a statistical null hypothesis H_0 might be: the true slope of the regression of condition as a function of traffic is 0.

Examples 9.3.1. The following are examples of null hypotheses.

1. $H_0 : \mu = 0$. (The population mean is 0.)
2. $H_0 : \beta_1 = 0$. (The "true" slope is 0 – assuming a model like $\mathrm{E}(Y) = \beta_0 + \beta_1 x$.)
3. $H_0 : \beta_1 = \beta_2$ (Two parameters in the model are equal.)
4. $H_0 : \beta_2 = \beta_3 = 0$ (Two parameters in the model are both equal to 0.)

9.3.1 The Four Step Process

Hypothesis testing generally follows a four-step process.

1. State the null (H_0) and alternative (H_1) hypotheses.

   The null hypothesis is on trial and innocent until proven guilty. We will render one of two verdicts: Reject the null hypothesis (guilty) or do not reject the null hypothesis (not guilty). *Important! We can never* ***accept*** *or* ***prove*** *either hypothesis – only reject the null as unlikely, or fail to reject the null since it seems likely.*

2. Compute a test statistic.

   For a statistical hypothesis test, all the evidence against the null hypothesis must be summarized in a single number called the test statistic. It is a statistic because it is a number computed from the data. It is called a test statistic because we are using it to do hypothesis testing.

3. Determine the p-value.

   The p-value is a probability: *Assuming* the null hypothesis is true, how likely are we to get at least as much evidence against it as we have in our data (i.e., *a test statistic at least as unusual as the one observed*) just by random chance?

4. Interpret the results.

   If the p-value is small, then one of two things is true:

   (a) The null hypothesis is true and something very unlikely occurred in our sample, or

   (b) The null hypothesis is false, so it is unsurprising that our observed data yield statistics that seem unlikely based on that (incorrect, untrue) hypothesis.

   For this reason we consider small p-values to provide evidence against the null hypothesis.

9.3.2 The Lady Tasting Tea, Revisted

Example 9.3.2. For the lady tasting tea, this process looks like

1. H_0: The probability of being correct is 0.5 (she's just guessing).

   H_a The probability of being correct is larger than 0.5 (she can do better than someone who just guesses).

2. Test statistic: $x = 9$ (of $n = 10$) were correct.

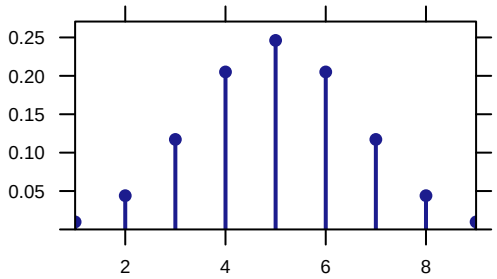
3. P-value

   (a) Based on our empirical method, we estimate a p-value of

```
prop( ~(heads >= 9), data = RandomLadies)

## prop_TRUE
##     0.0115
```

   (b) The probabilities for the number of correct guesses can be worked out theoretically as well. The resulting distribution is called a **binomial** distribution. As with other distributions we have seen, R includes the functions `dbinom()`, `pbinom()`, `qbinom()`, and `rbinom()`.

```
plotDist("binom", size = 10, prob = 0.5)     # size = # of flips; prob = probability of heads
dbinom(9, 10, .5) + dbinom(10, 10, .5)

## [1] 0.01074

1 - pbinom(8, 10, .5)         # Note: P(X >=9) = 1 - P(X <=8)

## [1] 0.01074
```



4. Interpret the p-value.

   How small is "small", and how small does a p-value have to be before we reject the null hypothesis? Often, a **significance level** (usually called α) of 0.05 is used. Sometimes $\alpha = 0.01$ is used instead. Basically, this corresponds to a 5% chance of seeing results as extreme as those found in our data, were the null hypothesis really true. If we want to be more conservative about our judgement (not rejecting the null hypothesis unless the evidence in the data is stronger against it), we could use a smaller α value.

   It is also common for researchers to simply report the p-values they obtain from their analysis, allowing readers to draw conclusions on their own.

Example 9.3.3. This situation is so common that there is a function to do the calculations for us. We just need to provide the values of x and n:

```
binom.test(x = 9, n = 10)           # default test is 2-sided
```

```
##
##
##
## data:  9 out of 10
## number of successes = 9, number of trials = 10, p-value = 0.02
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5550 0.9975
## sample estimates:
## probability of success
##                    0.9
```

The output above doesn't match what we obtained in Example 9.3.2 because the alternative hypothesis is different. It accepts both low number of correct identifications (0 or 1) and high numbers (9 or 10) as evidence against the null hypothesis. If we want a one-sided p-value, we just need to ask:

```
binom.test(x = 9, n = 10, alternative = "greater")
```

```
##
##
##
## data:  9 out of 10
## number of successes = 9, number of trials = 10, p-value = 0.01
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.6058 1.0000
## sample estimates:
## probability of success
##                    0.9
```

Approximate p-value	Translation
> 0.10	No convincing evidence against the null hypothesis
$0.05 - 0.10$	Weak evidence against the null hypothesis
$0.01 - 0.05$	Some evidence against the null hypothesis
< 0.01	Strong evidence against the null hypothesis
< 0.001	Very strong evidence against the null hypothesis

9.4 Statistical Significance

The word "significant" has a special meaning in statistics. If we say that a difference or relationship between variables is significant, that means that we have applied a hypothesis test, and have *failed* to reject the null hypothesis – in other words, we have data that provides some evidence against the null hypothesis, and supporting the alternative.

So, a pattern strong enough cause us to reject a null hypothesis of "no difference" or "no pattern" or "no relationship" is called a **statistically significant** difference, pattern, or relationship. Differences or relationships may fail to be statistically significant if they are small or weak, if they are masked by underlying variability, or if there is too little data. Good studies will collect enough data and work to reduce variability (if that is possible) in order to have a reasonable expectation of detecting differences if they are large enough to be scientifically interesting.

9.5 T-tests

Many hypothesis tests are conducted based on a t-distribution, and so they are called "t-tests". This is because, according to the Central Limit Theorem, the sampling distributions of most of our statistics (parameter estimates calculated from data) follow Normal distributions. But just as we did when computing confidence intervals, we'll always have to use the t-distribution rather than the normal distribution, since we don't know σ (the true population standard deviation), and since our sample size is finite. These t-tests all use a similar sort of test statistic:

$$t = \frac{\text{estimate} - \text{hypothesized value}}{\text{standard error}}$$

The numerator tells us that the more the estimate and the hypothesized value differ, the stronger the evidence. The denominator tells us that differences mean more when the standard deviation is small than when the standard deviation is large.

The test statistic is converted to a p-value by comparing it to the t-distribution with appropriate degrees of freedom. For linear models, this is the degrees of freedom associated with the residual standard error. If considering some statistic (say, the mean value) for a single variable, the degrees of freedom will be n-1, where n is the sample size.

9.5.1 The 1-sample t-test

The 1-sample t-test tests the null hypothesis

- $H_0 : \mu = \mu_0$, vs.
- $H_a : \mu \neq 0$.

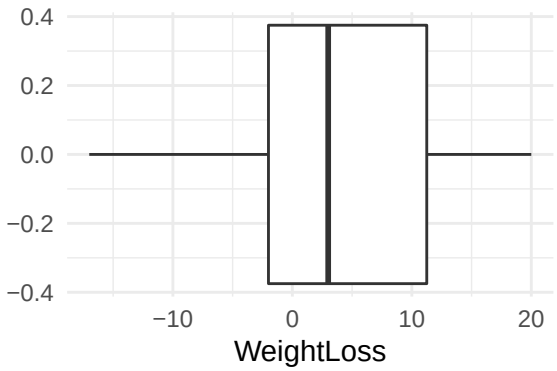
That is, it tests whether there is evidence that the mean of some population (μ) is different from some hypothesized value (μ_0) – often $\mu_0 = 0$.

Example 9.5.1. Let's look at some data on weight loss programs. In this data set, there were two groups. One group received a monetary incentive if they lost weight while following a weight loss program. The controls did not receive a monetary incentive, but followed the same program otherwise. Our null hypothesis is that the control participants would not lose weight – that the true weight loss without incentives would average 0 pounds. Let's see whether on average the controls lost weight:

```
library(Stat2Data)
data(WeightLossIncentive)
Controls <- WeightLossIncentive %>% filter(Group == "Control")
df_stats( ~ WeightLoss, data = Controls )
```

```
##     response min Q1 median    Q3 max  mean    sd  n missing
## 1 WeightLoss -17 -2      3 11.25  20 3.921 9.108 19       0
```

```
gf_boxplot( ~ WeightLoss, data = Controls)
```



The standard error when doing inference for a mean is

$$SE = \frac{s}{\sqrt{n}} = \frac{9.108}{\sqrt{19}} = 2.089$$

```
SE <- 9.108 / sqrt(19); SE
```

```
## [1] 2.09
```

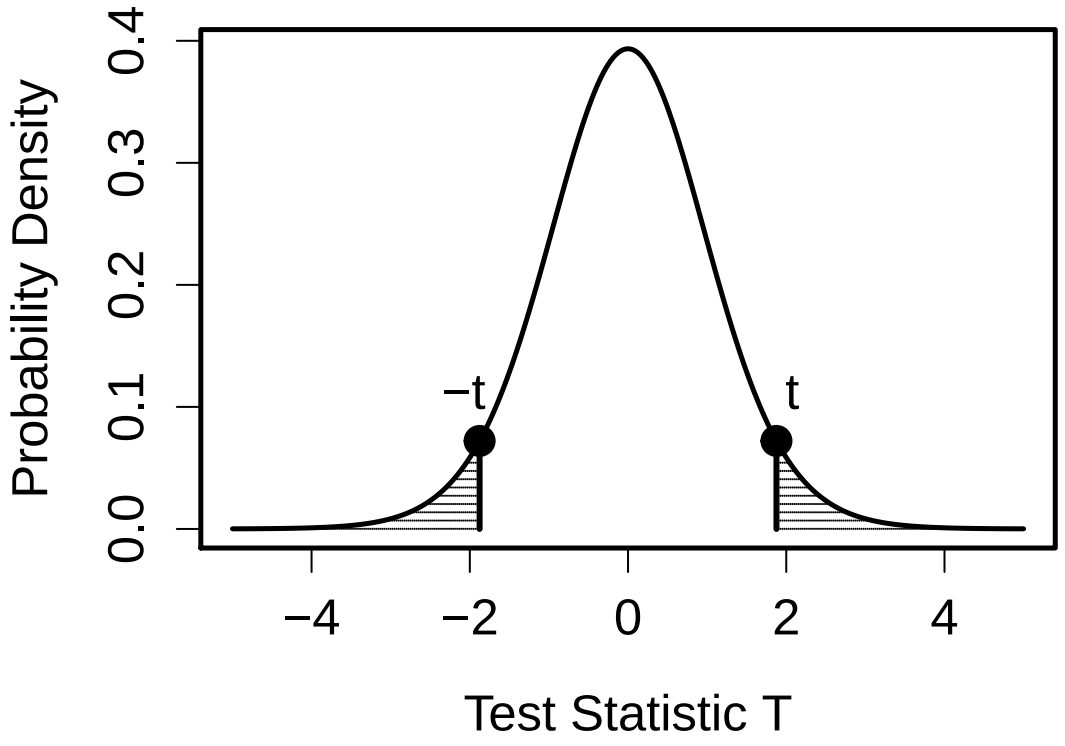
If we want to test our null hypothesis, then we compute a t-statistic:

```
t <- (3.92 - 0) / SE; t
```

```
## [1] 1.876
```

and from this a p-value, which is the tails probability for a t-distribution with 18 degrees of freedom. In other words, the p-value gives the probability of getting a test statistic at least as big as the one we really got, assuming that the test statistic follows a t-distribution with $n - 1$ degrees of freedom.

First, we find the probability of observing a test statistic of $t = 1.876$ or larger. Then, we have to double this value – it would be at least as unlikely to see a test statistic of -1.876 or smaller. Our p-value should give the area under the t-distribution curve for x-values smaller than -1.876 *and* larger than 1.876; it's the shaded area in the figure below:




```
1 - pt( t, df = 19-1 )
```

```
## [1] 0.03848
```

```
2 * ( 1 - pt( t, df = 19-1 ) )
```

```
## [1] 0.07696
```

Our p-value is 0.077 (1 or 2 significant digits are sufficient for reporting p-values). This is not compelling evidence that the weight loss program (without incentives) actually leads to a change in weight. A change this large could occur just by chance in nearly 8% of samples.

Example 9.5.2. In R, there is also a function to automate this t-test:

```
t.test( ~ WeightLoss, data = Controls)
```

```
##
##  One Sample t-test
##
## data:  WeightLoss
## t = 1.9, df = 18, p-value = 0.08
## alternative hypothesis: true mean is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.4688  8.3109
## sample estimates:
## mean of x
##     3.921
```

If we don't want so much out put, we can ask R to report only the p-value:

```
pval(t.test( ~ WeightLoss, data = Controls))


## p.value
## 0.07688
```

By default, `t.test()` uses a significance level $\alpha = 0.05$. If we want to specify a different α, we can use the input `conf.level` as follows:

```
t.test( ~WeightLoss, data=Controls, conf.level=0.01)


##
##  One Sample t-test
##
## data:  WeightLoss
## t = 1.9, df = 18, p-value = 0.08
## alternative hypothesis: true mean is not equal to 0
## 1 percent confidence interval:
##  3.894 3.948
## sample estimates:
## mean of x
##     3.921
```

9.5.2 The paired 2-sample t-test

Sometimes, a dataset contains *paired* observations. These might be, for example, two measurements on the same experimental subject before and after some experimental treatment; measurements on the same subject at two times, or in two different situations; (or others). In this case, we can not consider the measurements within a pair to be independent of each other. But what we are really interested in is the magnitude of the difference between the 2 observations in each pair. So this "paired t-test" problem reduces to a one-sample t-test, where the test statistic is constructed using the *differences* D between the 2 measurements in each pair:

$$t = \frac{\text{observed average difference} - \text{hypothesized average difference}}{\text{standard error}}$$

Let's consider an example. **Example 9.5.3.** The following table provides the corneal thickness in microns of both eyes of patients who have glaucoma in one eye:

Healthy	484	478	492	444	436	398	464	476
Glaucoma	488	478	480	426	440	410	458	460
Difference	4	0	-12	-18	4	12	-6	-16

The corneal thickness is likely to be similar in the two eyes of any single patient, so that the two observations on the same patient cannot be assumed to be independent. But maybe (after accounting for differences between people), there is some difference in corneal thickness that has to do with the presence or absence of glaucoma. First, we can have a look at the data:

```
## Error in plot(x = c(1:length(glaucoma)), y = glaucoma, pch = 19, ylab = "Corneal Thickness
(um)", :  object 'glaucoma' not found

## Error in points(c(1:length(healthy)), healthy, pch = 17, col = "grey", :  object 'healthy'
not found

## Error in strwidth(legend, units = "user", cex = cex, font = text.font):  plot.new has not been
called yet
```

It looks like healthy corneas are often thicker. To try to quantify this, we consider the difference between each pair of observations, denoted by d_i. We wish to test,

$$H_0 : \mu = 0 \qquad \text{vs} \qquad H_1 : \mu \neq 0.$$

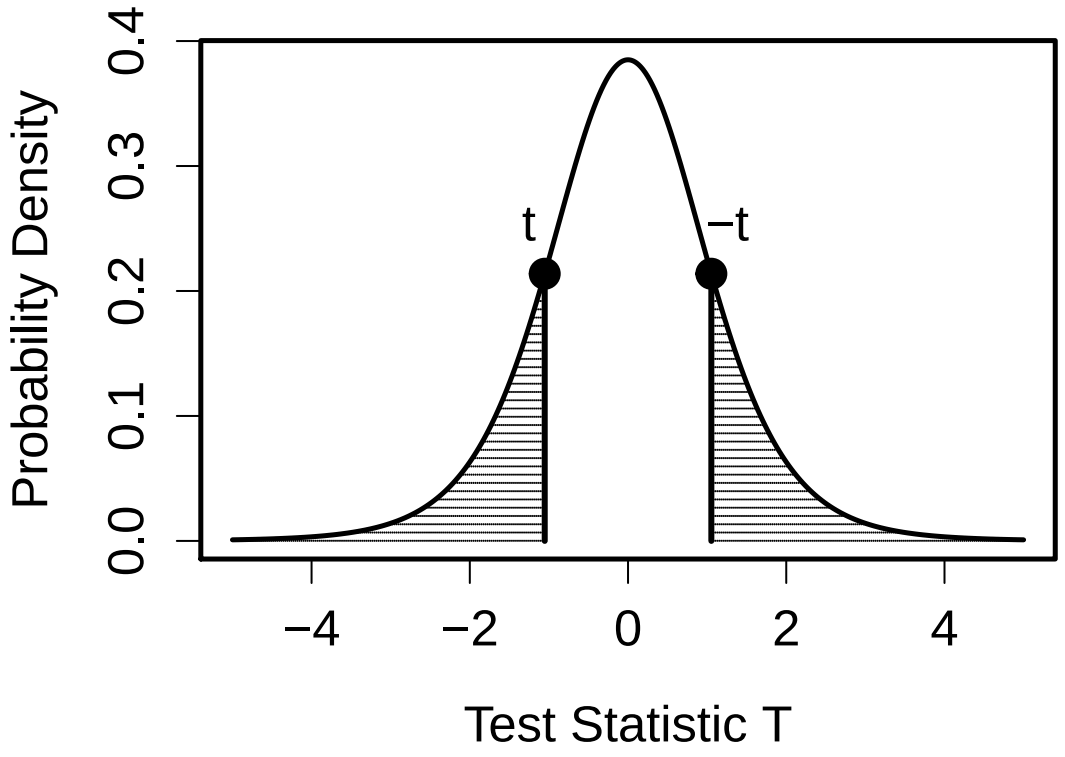
Under H_0,

$$T = \frac{\bar{D}}{S/\sqrt{n}} \sim t_{n-1}.$$

In R, we can compute D (the average of d_i) and its standard error to obtain the test statistic t and the corresponding p-value:

```
model2 <- lm(WeightLoss ~ Group, data = WeightLossIncentive)
n <- 8
healthy <- c(484 , 478 , 492 , 444 , 436 , 398 , 464 , 476)
glaucoma <- c(488 , 478 , 480 , 426 , 440 , 410 , 458 , 460)
diffs <- glaucoma-healthy
D <- mean(diffs)
SE <- sd(diffs)/sqrt(n)
t <- (D - 0)/SE; t


## [1] -1.053


pval <- 2 * pt(t, df=n-1); pval


## [1] 0.3273
```

Note that we used 2 * pt(...) because our test statistic t was negative. If it had been positive, we would have used 2 * (1-pt(...)) instead. The illustration below may help make this clearer – the p-value we are computing corresponds to the shaded areas in the plot.



We can also let R make all the computations for us. Notice that we use the *differences* between pairs as the input data, rather than the raw data itself.

```
favstats(WeightLoss ~ Group, data = WeightLossIncentive)
```

```
##       Group   min   Q1 median    Q3 max   mean    sd  n missing
## 1   Control -17.0 -2.0      3 11.25  20  3.921 9.108 19       0
## 2 Incentive  -0.5  7.5     18 24.00  30 15.676 9.414 17       2
```

```
coef(summary(model2))
```

```
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)       3.921      2.123   1.847 0.0734506
## GroupIncentive   11.755      3.089   3.805 0.0005635
```

```
t.test(~ diffs, df=7)
```

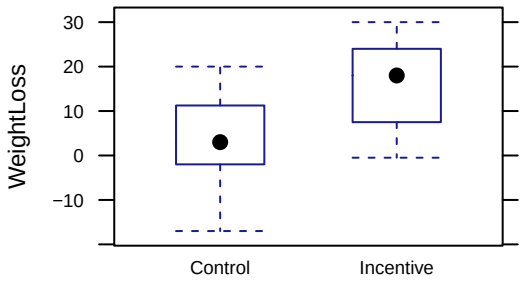
```
##
##  One Sample t-test
##
## data:  diffs
## t = -1.1, df = 7, p-value = 0.3
```

```
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -12.982   4.982
## sample estimates:
## mean of x
##        -4
```

9.5.3 The (unpaired) 2-sample t-test

In some cases, our research sample may contain data from 2 different categories of observational units. For example, in the weight loss data, there were the control participants we considered above; there were also a number of incentivized participants, who received money if they lost weight. We might be interested in considering whether the incentive made a difference. Before we begin, we can plot the data:

```
bwplot(WeightLoss~Group, data=WeightLossIncentive)
```



It looks like the incentive group lost more weight. But how can we judge whether the difference between groups is really an effect of the incentive, and not just random variation?

In this case, the null hypothesis H_0 would be that the average weight loss by control participants (μ_c) was the same as the average weight loss by incentivized participants (μ_i) – $H_0 : \mu_c = \mu_i$. The alternative hypothesis would be $H_1 : \mu_c \neq \mu_i$ – there was a difference between the two groups.

But in this case – with data from 2 different categories – how can we compute the appropriate standard error and test statistic for a t-test? We have to consider the fact that there may be different numbers of data points in the 2 categories. Let n be the sample size within the first category X (control participants in the example), and m be the sample size in the other category Y (incentive participants in the example). How can we compute a standard error for a difference in means between the two groups?

In this case, if we assume that the sample variaces are equal between the two categories, then we can define the pooled sample variance s_p:

$$s_p^2 = \frac{(n-1)s_X^2 + (m-1)s_Y^2}{(m+n-2)}$$

Here, s_Y and s_X are the sample standard deviations within each category. We will not provide proof here, but it is known that in this case, the test statistic is:

$$T = \frac{\bar{X} - \bar{Y}}{S_p\sqrt{\frac{1}{m} + \frac{1}{n}}} \sim t_{m+n-2}.$$

As written in the equation above, this test statistic follows a t-distribution with $m + n - 2$ degrees of freedom. We could carry out a two-sample t-test by hand using the test statistic defined above, or we can use the function `t.test()` and R will do the computations for us. For the weight loss example:

```
head(WeightLossIncentive,4)
```

```
##   WeightLoss   Group Month7Loss
## 1       12.5 Control       -2.0
## 2       12.0 Control        7.0
## 3        1.0 Control       19.5
## 4       -5.0 Control       -0.5
```

```
t.test(~WeightLoss, groups=Group, data=WeightLossIncentive,
       var.equal=TRUE)
```

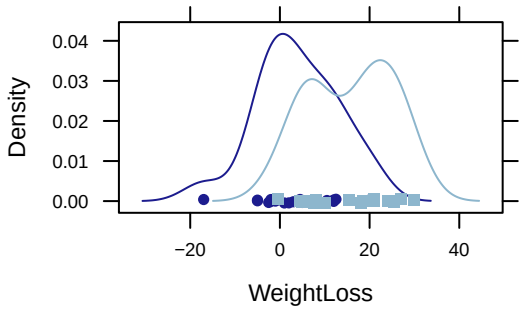
```
##
##  Two Sample t-test
##
## data:  WeightLoss by Group
## t = -3.8, df = 34, p-value = 6e-04
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -18.033  -5.478
## sample estimates:
##   mean in group Control mean in group Incentive
##                   3.921                  15.676
```

Note the "var.equal=TRUE" input argument. With this input, R will carry out the t-test assuming equal variance between the two categories. If we want to avoid making this assumption, there is a modified version of the two-sample t-test called the Welch two-sample t-test, which does not assume equal variances. If we omit the "var.equal" input, or set it to "var.equal=FALSE", then R will do a Welch two-sample t-test for us. (We will not cover in this course how to do a Welch test by hand).

We might be able to judge informally whether the equal-variance assumption is valid by looking at the distributions of our variable of interest grouped by category, and computing the sample standard deviations by groups:

```
densityplot(~WeightLoss, groups=Group, data=WeightLossIncentive)
sd(~WeightLoss|Group, data=WeightLossIncentive, na.rm=TRUE)
```

```
##   Control Incentive
##     9.108     9.414
```



We don't have an obvious indication of unequal variances. (This assumption can also be tested statistically, although we won't learn how in this class, and it's generally not recommended to do such a test prior to running a t-test). If we wanted to do the t-test without the equal variance assumption, in R, we would use:

```
t.test(~WeightLoss, groups=Group, data=WeightLossIncentive)


##
##  Welch Two Sample t-test
##
## data:  WeightLoss by Group
## t = -3.8, df = 33, p-value = 6e-04
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -18.050  -5.461
## sample estimates:
##   mean in group Control mean in group Incentive
##                   3.921                  15.676
```

In practice, there is no real disadvantage to simply using Welch's test all the time. So in general, if you are doing a 2-sample unpaired test, you should always use `var.equal=FALSE` (or omit the input `var.equal`, and it will default to FALSE).

9.5.4 Testing Model Coefficients

We can also use t-tests to test the null hypothesis that there is no relationship between the predictor and explanatory variables in a regression model. If we can't reject that hypothesis, then we have evidence that there *is* really some relationship, and that the response can be predicted based upon the explanatory variable.

Example 9.5.4. Suppose you suspect that drag force should be proportional to the square of velocity. Let's see if that is consistent with the data collected by some physics students. In this experiment, the students rigged up neutrally buoyant balloon and then loaded it with different amounts of weight and dropped it until and recorded its terminal velocity. At that point the force due to gravity (determined by the mass loaded to the balloon) is equal to the drag force (because there is no acceleration).

We'll fit a power law model

$$\texttt{force.drag} = A \cdot \texttt{velocity}^a$$

and test the hypothesis that $a = 2$. We can fit this model using a log-log transformation:

$$\log(\texttt{force.drag}) = \log(A) + a\log(\texttt{velocity})$$

So $a = \beta_1$ in our usual linear model notation.

```
drag.model <- lm(log(force.drag) ~ log(velocity), data = drag)

## Error in is.data.frame(data):  object 'drag' not found

gf_point(log(force.drag) ~ log(velocity), data = drag)

## Error in gf_ingredients(formula = gformula, data = data, gg_object = object, :  object 'drag'
not found
```

The fit is not perfect, and in fact suggests a systematic problem with the way these data were collected:[1]

```
gf_point(resid(drag.model) ~ fitted(drag.model))

## Error in fitted(drag.model):  object 'drag.model' not found

gf_qq( ~ resid(drag.model))

## Error in resid(drag.model):  object 'drag.model' not found
```

```
summary(drag.model)

## Error in h(simpleError(msg, call)):  error in evaluating the argument 'object' in selecting
a method for function 'summary':  object 'drag.model' not found
```

None of the p-values produced in this output is what we want. They are testing the hypotheses that $\beta_0 =$ and that $\beta_1 = 0$. $\beta_1 = 0$. But we can easily calculate the p-value we want since we have the standard error and degrees of freedom.

«««< HEAD:Stat241-HypothesisTesting.Rnw

```
beta1.hat <- 2.051
SE <- 0.05366
t <-  ( beta1.hat - 2 ) / SE; t

## [1] 0.9504
```

[1] There is some evidence in this data that some of the observations did not reach critical velocity. It would be good to refit this data with those observations removed from the data. See Exercise 9.3.

```
2 * pt( - abs(t), df= 40 )
```

```
## [1] 0.3476
```

With this large a p-value, we cannot reject the null hypothesis that $p = 2$. **A large p-value does not prove that $p = 2$, but it does say that our data are consistent with that value.** Of course, our data may be consistent with many other values of p as well.

Example 9.5.5. We could also do the previous example using a nonlinear model

```
drag.model2 <- nls(force.drag ~ A * velocity^p, data = drag, start = list(A = 1,p = 2))
```

```
## Error in nls(force.drag ~ A * velocity^p, data = drag, start = list(A = 1, :  object 'drag'
not found
```

```
summary(drag.model2)
```

```
## Error in h(simpleError(msg, call)):  error in evaluating the argument 'object' in selecting
a method for function 'summary':  object 'drag.model2' not found
```

Again, the p-values listed are not of interest (they are testing the hypotheses that each coefficient is 0). But we can compute the p-value of interest as follows:

```
t <- (2 - 1.92989) / 0.0944; t
```

```
## [1] 0.7427
```

```
2 * pt( - abs(t), df = 40)
```

```
## [1] 0.462
```

Although the two p-values are different, the conclusion is the same using either model: Our data are consistent with the hypothesis that $p = 2$.

Example 9.5.6.

```
data(PorschePrice, package = "Stat2Data")
head(PorschePrice)
```

```
##   Price Age Mileage
## 1  69.4   3    21.5
## 2  56.9   3    43.0
## 3  49.9   2    19.9
## 4  47.4   4    36.0
## 5  42.9   4    44.0
## 6  36.9   6    49.8
```

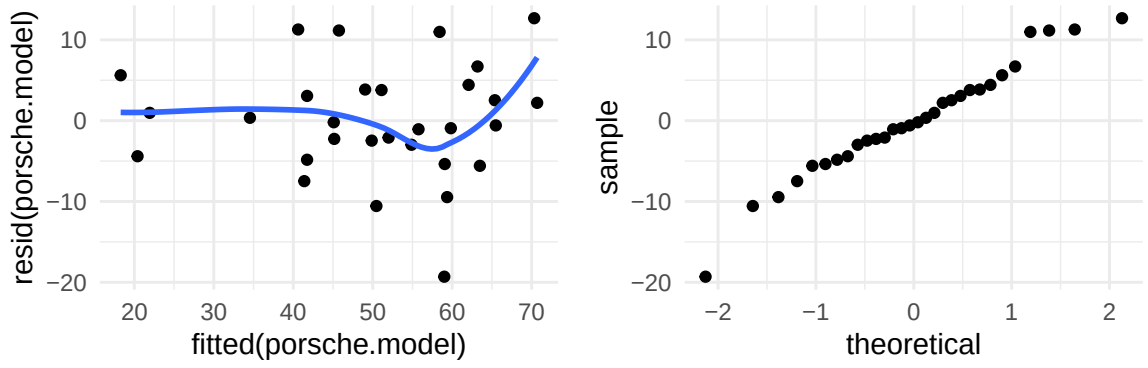
```
porsche.model <- lm(Price ~ Mileage, data = PorschePrice)
summary(porsche.model)
```

```
##
## Call:
## lm(formula = Price ~ Mileage, data = PorschePrice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.308  -4.047  -0.395   3.837  12.676
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.0905     2.3699    30.0   <2e-16
## Mileage      -0.5894     0.0566   -10.4    4e-11
##
## Residual standard error: 7.17 on 28 degrees of freedom
## Multiple R-squared:  0.795,Adjusted R-squared:  0.787
## F-statistic:  108 on 1 and 28 DF,  p-value: 3.98e-11
```

```
gf_point( resid(porsche.model) ~ fitted(porsche.model)) %>% gf_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```

```
gf_qq( ~ resid(porsche.model) )
```



The model looks reasonable. What are the two hypotheses being tested?

1. $H_0 : \beta_0 = 0$.

   Often this is not an interesting test because often we are not so interested in the intercept β_0, and especially not in whether it is 0. In this case, the intercept might be interesting because it tells us the price of a Porsche with no miles. On the other hand, we might not expect a used car, even one with very few miles to fit the same pattern as a new car. There is probably a loss in value that occurs as soon as a car is purchased.

   In any case, it is clear that the intercept will not be 0; we don't need a hypothesis test to tell us that. Indeed, the evidence is incredibly strong.

   A confidence interval for the intercept is more interesting since it gives a sort of "starting price" for used Porches.

   ```
   confint(porsche.model)

   ##                2.5 %   97.5 %
   ```

```
## (Intercept) 66.2360 75.9449
## Mileage     -0.7054 -0.4734
```

2. $H_0 : \beta_1 = 0$.

   There is strong evidence against this hypothesis as well. This is also not surprising. If $\beta_1 = 0$, that would mean that the price of the cars does not depend on the mileage.

   A test of $\beta_1 = 0$ in a simple linear model is often called the **model utility test** because it is testing whether the predictor (without any others) is of any use to us or not.

3. $H_0 : \beta_1 = \beta_{10}$.

   Although the output above doesn't do all of the work for us, we can test other hypotheses as well. (The notation above is a bit tricky, β_{10} should be read "β_1 null" – it is a hypothesized value for β_1.)

   For example, let's test $\beta_1 = -1$. That the hypothesis that the value drops one dollar per mile driven.

   While this example is interesting as an exercise, it is quite rare to have a sensible hypothesized value for a regression slope parameter that we want to test. It is much more common to ask, as we did above, "is there a pattern here indicating that the predictor is a useful predictor of the response"?

```
t <- (-0.5894 - (-1) ) / 0.0566; t

## [1] 7.254

2 * pt( - abs(t), df = 28 )

## [1] 6.749e-08
```

   This p-value is small enough to cause us to reject this value for β_1.

9.6 Connection to Confidence Intervals

There is a natural duality between t-based hypothesis tests and confidence intervals. Since the p-value is computed using tail probabilities of the t-distribution and confidence level describes the central probability, the p-value will be below 0.05 exactly when the hypothesized value is not contained in the 95% confidence interval. (Similar statements can be made for other confidence levels.)

Example 9.6.1. In the preceding example we rejected the null hypothesis that $\beta_1 = -1$. In fact, we will reject (at the $\alpha = 0.05$ level) any hypothesized value not contained in the 95% confidence interval.

```
t <- (-0.5894 - (- .71)) / 0.0566; t

## [1] 2.131

2 * pt( - abs(t), df = 28)

## [1] 0.04203
```

But we won't reject values inside the confidence interval.

```
t <- (-0.5894 - (- .70)) / 0.0566; t
```

```
## [1] 1.954
```

```
2 * pt( - abs(t), df = 28 )
```

```
## [1] 0.06075
```

Example 9.6.2. The output below illustrates this duality.

```
drag.model <- lm(log(force.drag) ~ log(velocity), data = drag)
```

```
## Error in is.data.frame(data):  object 'drag' not found
```

```
coef(summary(drag.model))
```

```
## Error in h(simpleError(msg, call)):  error in evaluating the argument 'object' in selecting
a method for function 'summary':  object 'drag.model' not found
```

```
beta1.hat <- 2.051
SE <- 0.05366
t <-  ( beta1.hat - 2 ) / SE; t
```

```
## [1] 0.9504
```

```
2 * pt( - abs(t), df= 40 )
```

```
## [1] 0.3476
```

```
confint(drag.model)
```

```
## Error in confint(drag.model):  object 'drag.model' not found
```

Since the confidence interval for p (i.e., for β_1) includes 2, 2 is a plausible value for the power (i.e., consistent with our data). A 2-sided p-value larger than 0.05 says the same thing at the same level of confidence.

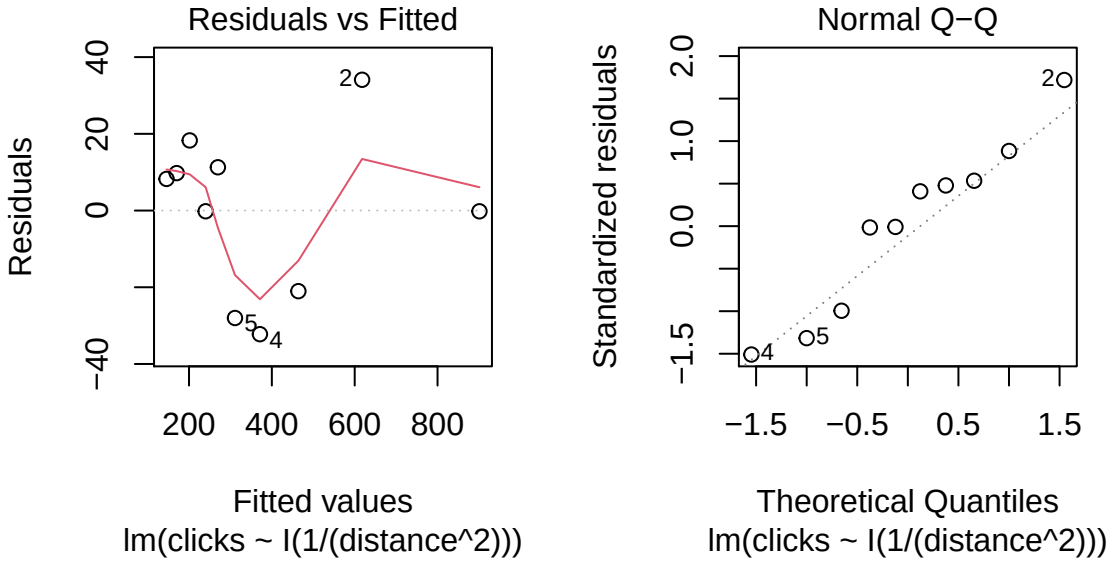
9.7 Exercises

9.1 An experiment was conducted to see if the number of clicks on a Geiger counter in a 7.5 minute interval is related to the distance (in m) between a radioactive source and the detection device according to an inverse square law:

$$\text{clicks} = A + \frac{k}{\text{distance}^2}$$

Answer the questions below using the following output

```
model <- lm( clicks ~ I(1/(distance^2)), data = Geiger)
summary(model)


##
## Call:
## lm(formula = clicks ~ I(1/(distance^2)), data = Geiger)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32.23 -15.82   4.03  10.90  34.09
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          114.3       10.8    10.6  5.5e-06
## I(1/(distance^2))     31.5        1.0    31.5  1.1e-09
##
## Residual standard error: 22.5 on 8 degrees of freedom
## Multiple R-squared:  0.992,Adjusted R-squared:  0.991
## F-statistic:  990 on 1 and 8 DF,  p-value: 1.13e-09


plot(model, w = 1:2)
```



a) Are there are any reasons (from what you can tell in the output above) to be concerned about using this model?

b) What does A tell us about this situation? What would it mean if A were 0? What if $A \neq 0$?

c) What is the estimate for A? Express this as an estimate $\pm$ uncertainty using our rules for numbers of digits.

d) What is the p-value for the test of the null hypothesis that $A = 0$? What conclusion do we draw from this?

e) What does k tell us about this situation? In what situations would k be larger or smaller? What would it mean if k were 0?

f) Express the estimate for k as an estimate $\pm$ uncertainty.

g) What is the p-value for the test of the null hypothesis that $k = 0$? What conclusion do we draw from this?

h) A standard radioactive substance has a value of $k = 29.812$. Might that be the substance we are using here? Conduct an appropriate hypothesis test to answer this question. Carefully show all four steps.

9.2 A gentleman claims he can distinguish between four vintages of a particular wine. His friends, assuming he has probably just had too much of each, decide to test him. They prepare one glass of each vintage and present the gentleman with four unlabeled glasses of wine. What is the probability that the gentleman correctly identifies all four simply by guessing?

9.3 Redo the drag force analysis after removing observations that appear not to have reached terminal velocity.

If you can describe the rows you want to remove logically, the `subset()` command works well for this. You can also remove rows by row number. For example, the following removes rows 1, 3, 5 and 7:

```
drag[ - c(1, 3, 5, 7), ]
```

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

9.4 Sixteen samples of a certain brand of hydrogenated vegetable oil were tested to determine their melting point. The mean melting point for the 16 samples was 94.32 degrees and the standard deviation was 1.2 degrees.

a) Conduct a test of the hypothesis $\mu = 95$. Follow the four step procedure.

b) Will 95 be inside or outside of a 95% confidence interval for the melting point? How do you know?

9.5 The Charpy V-notch impact test (`https://en.wikipedia.org/wiki/Charpy_impact_test`) is a common way to test the toughness of a material. This test was applied to 42 specimens of a particular alloy at 110 degrees F. The mean amount of transverse lateral expansion was computed to be 73.1 mils with a sample standard deviation of 5.9 mils.

To be suitable for a particular application, the true amount of expansion must be less than 75 mils. The alloy will not be used unless their is strong evidence (a p-value below 0.01) that this specification is met.

a) Use a p-value to decide whether this alloy may be used.

b) Use a confidence interval to decide whether this alloy may be used.

c) Are there advantages to one approach over the other? If you had to present an argument to your boss, which approach would you use?

10 More Examples

10.1 Heat Exchanger Example

In this section will discuss several parts of the statistical analysis of a laboratory experiment involving a heat exchanger.

10.1.1 Apparatus and Measurements

Figure 10.1 shows a diagram illustrating the heat exchanger. Fluids of different temperatures flow in the annulus ($\dot{m}_3$) and in the inner tube ($\dot{m}_1$). The entire apparatus is insulated, so we expect little or no heat to be exchanged with the surroundings.

Mass flow rates ($\dot{m}$) are controlled via valves. Two mass flow rates ($\dot{m}_1$ and $\dot{m}_3$) are measured by rotameters.[1] Temperatures (T_1–T_4) are measured by thermocouples.[2]

Each observation consists of four (4) temperature measurements and two (2) mass flow rate measurements. Here is an example data set with one set of measurements at each of 6 experimental conditions:

[1] See `http://www.omega.com/prodinfo/rotameters.html` for an introduction.
[2] See `http://www.omega.com/techref/themointro.html` for an introduction.

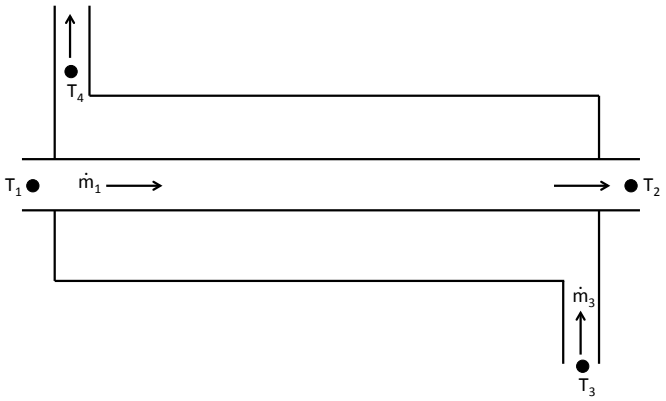
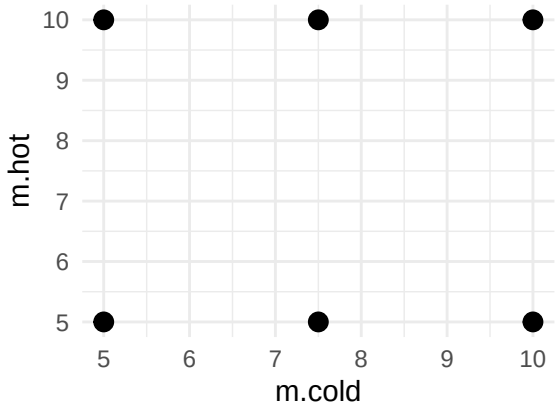


Figure 10.1: Heat exchanger with statepoints.

```
HeatX
```

```
##   trial T.cold.in T.cold.out T.hot.in T.hot.out m.cold m.hot
## 1     1      14.3       18.6     38.2      33.9   10.0    10
## 2     2      14.1       18.7     35.9      32.8    7.5    10
## 3     3      14.1       19.4     35.9      33.4    5.0    10
## 4     4      14.2       18.4     35.8      31.8    5.0     5
## 5     5      14.1       17.4     35.7      30.9    7.5     5
## 6     6      14.1       16.9     35.8      30.4   10.0     5
```

```
gf_point(m.hot ~ m.cold, data = HeatX, main = "Experimental Configurations", size = 3)
```



Note: `read.csv()` can read data from a file as well. If you are using the server version of RStudio you first need to upload your CSV file to the server. Look in the Files pane for the upload button to upload your own data.

10.1.2 Standardizing Units

The recorded flow rates are in L/min. We will convert them to L/sec and use seconds as our standard unit of time troughout the analyses.

```
HeatX <- HeatX %>%
  mutate(
    m.hot = m.hot / 60,
    m.cold = m.cold / 60
  )
```

We could also convert temperatures to degrees Kelvin, but since temperatures only appear as differences between two temperatures in the expression used here, we can leave them in degrees Celsius.

Table 10.1 contains notation and definitions of important quantities involved in the heat exchanger experiment.

10.1.3 Calculating the heat transfer, $\dot{Q}$

The amount of heat exchanged between the hot and cold water can be determined from the temperature change and the mass flow rate as follows:

Table 10.1: Notation for quantities involved in the heat exchanger experiment. Note, some of the physical "constants" are actually temperature dependant. In these cases, a value has been chosen that reflects the temperature range (approx. 15–35 degrees C) seen in the data.

Symbol	Definition	Units	Estimate
$\dot{Q}$	heat transfer rate	W	
$\dot{m}$	mass flow rate	kg/s	
C_p	specific heat	kJ/(kg·K)	4.18 ± 0.1
T	temperature	K	
D	diameter of inner tube	m	0.0143 ± 0.0004 m
			$(9/16 \pm 1/64$ inches)
L	length of the heat exchanger	m	1.626 ± 0.006
			$(64 \pm 1/4$ inches)
A	surface area of the inner tube (πDL) [m^2]		
U	heat transfer coefficient W/(m^2·K)		
h	convective heat transfer coefficient ($\approx 2U$)	W/(m^2·K)	
ΔT_{lm}	logarithmic mean temperature difference	K	
Nu_D	Nusselt number based on D	–	
Re_D	Reynolds number based on D	–	
Pr	Prandtl number	–	
μ	dynamic viscosity	kg/(m·s)	0.00102 ± 0.00001
k	water thermal conductivity	W/(m·K)	0.598 ± 0.004

$$\dot{Q}_1 = \dot{m}_1 C_p (T_2 - T_1) \tag{10.1}$$

$$\dot{Q}_3 = \dot{m}_3 C_p (T_4 - T_3) \tag{10.2}$$

We can estimate the values of $\dot{Q}$ from our data by direct caclulation:

```
C_p <- 4.18
HeatX2 <- HeatX %>%
  mutate(
    Q.cold = m.cold * C_p * (T.cold.out - T.cold.in),
    Q.hot  = m.hot  * C_p * (T.hot.out  - T.hot.in),
    Q.env  = Q.cold + Q.hot
  )
HeatX2
```

```
##   trial T.cold.in T.cold.out T.hot.in T.hot.out  m.cold   m.hot Q.cold  Q.hot      Q.env
## 1     1      14.3       18.6     38.2      33.9 0.16667 0.16667  2.996 -2.996 -2.220e-15
## 2     2      14.1       18.7     35.9      32.8 0.12500 0.16667  2.403 -2.160  2.438e-01
## 3     3      14.1       19.4     35.9      33.4 0.08333 0.16667  1.846 -1.742  1.045e-01
## 4     4      14.2       18.4     35.8      31.8 0.08333 0.08333  1.463 -1.393  6.967e-02
## 5     5      14.1       17.4     35.7      30.9 0.12500 0.08333  1.724 -1.672  5.225e-02
## 6     6      14.1       16.9     35.8      30.4 0.16667 0.08333  1.951 -1.881  6.967e-02
```

10.1.4 Estimating heat exchanged with environment

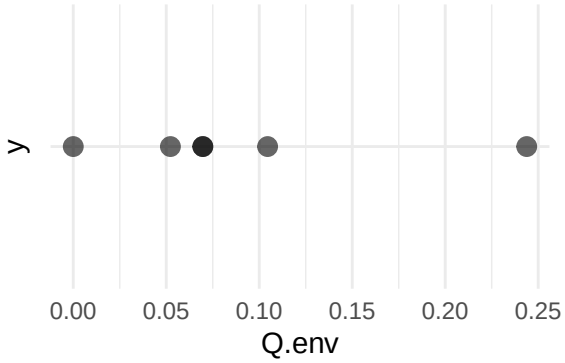
If no heat were exchanged with the environment and all measurements were without error, then our two estimates of $\dot{Q}$ would sum to 0. (Heat lost to one fluid is gained by the other.)

Assuming any loss to (or gain from) the environment is essentially constant for the apparatus over the experimental conditions analysed, we can use our 6 observations to estimate the amount of heat exchanged with the environment:

```
df_stats( ~ Q.env, data = HeatX2)
```

```
##   response       min     Q1  median      Q3    max    mean      sd n missing
## 1    Q.env -2.22e-15 0.0566 0.06967 0.09579 0.2438 0.08999 0.08274 6       0
```

```
gf_point( "" ~ Q.env, data = HeatX2, alpha = .6, cex = 3, jitter.data = TRUE)
```



From this we can compute either a p-value for the hypothesis test that the mean difference in heat change is 0 or create a confidence interval for the mean difference in heat change. The information above is enough to do this "by hand" using the standard error formula $SE = \frac{s}{\sqrt{n}}$ and a t-distribution with $n - 1 = 5$ degrees of freedom.

```
SE = 0.08274 /sqrt(6); SE
```

```
## [1] 0.03378
```

```
0.09003 + c(-1,1) * qt(0.975, df = 5) * SE        # 95% CI
```

```
## [1] 0.0032 0.1769
```

```
t <- (0.08999 - 0) / SE; t
```

```
## [1] 2.664
```

```
2 * pt( -abs(t), df = 5 )                         # p-value
```

```
## [1] 0.04466
```

Or we can let R do all the computations for us:

```
t.test( ~ Q.env, data = HeatX2)
```

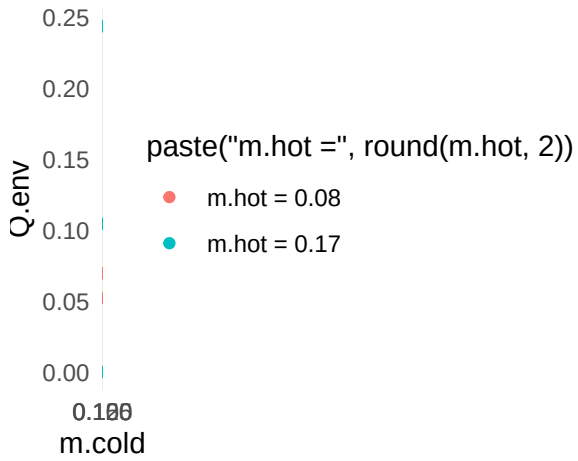
```
##
##  One Sample t-test
##
## data:  Q.env
## t = 2.7, df = 5, p-value = 0.04
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.003159 0.176813
## sample estimates:
## mean of x
##   0.08999
```

In our example data, there is modest evidence for exchange with the environment, but the estimated amount of heat gained from the environment is not very precisely estimated. Even at the highest end of the confidence interval, the heat exchanged with the environment is an order of magnitude smaller than the heat exchanged within in the apparatus.

Notes:

1. The fact that heat is gained to the system suggests that the cold water was in the outer pipe and hot water in the inner pipe.
2. This analysis assumes that the amount of heat gained/lost is constant over the different set-ups. From this small data set, there is not clear evidence of some other relationship between heat exchanged with the environment and the experimental set up:

```
gf_point(Q.env ~ m.cold, data = HeatX2, color = ~ paste("m.hot =", round(m.hot,2)))
```



3. The t-test and interval are based on the assumption that the distribution of deviations between the measured environment heat exchange and the actual is normally distributed. The data set is too small to provide much evidence upon which to judge whether this is a reasonable assumption. The largest value is quite a bit larger than the rest, but even if we remove that observation, our conclusions don't change dramatically:

```
t.test( ~ Q.env, data = subset(HeatX2, Q.env < max(Q.env)) )

##
##  One Sample t-test
##
## data:  Q.env
## t = 3.5, df = 4, p-value = 0.03
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.01184 0.10660
## sample estimates:
## mean of x
##   0.05922
```

Overall, we conclude that there is likely some heat exchanged with the environment, but the amount of heat exchange with the environment appears to be at most a small factor in this situation.

10.1.5 Estimating heat transfer between hot and cold water, $\dot{Q}$

Estimating $\dot{Q}$ with uncertainty is not possible from this data alone since since $\dot{Q}$ is not measured directly, and we have only one measurement for each experimental condition (so no way to look at how variable such measurements are without additional information. Furthermore, the design doesn't provide a method for estimating the uncertainty in $\dot{m}$ and T.

If, however, there are external estimates of the uncertainties for temperature and flow rate, and if we can assume these uncertainties are approximately independent, then we can use propagation of uncertainty to estimate the uncertainty in our estimates for heat exchanged. Such uncertainties might come from specifications of the equipment used or be based on past experience of the researcher.

For example, suppose that the uncertainties in temperature and flow rate measurements are approximately constant (over the range of temperatures involved): u_T, and $u_{\dot{m}}$. Then the uncertainty in the difference between two independent temperatures is $\sqrt{u_T^2 + u_T^2} = \sqrt{2}u_T$, and we can estimate the uncertainty in $\dot{Q}$ using the delta method.

- Let $\dot{Q}(\dot{m}, \Delta T) = C_p \dot{m} \Delta T$
- $\dfrac{\partial \dot{Q}}{\partial \dot{m}} = C_p \Delta T$ and $\dfrac{\partial \dot{Q}}{\partial \Delta T} = C_p \dot{m}$, so
- $u_{\dot{Q}} \approx C_p \sqrt{(\Delta T)^2 u_{\dot{m}}^2 + \dot{m}^2 u_{\Delta T}^2}$

If we estimate the uncertainty in measured temperatures to be 1 degree C and the uncertainty in flow rate to be 0.5 liters per minute (i.e., 0.008 L/sec), then we can compute the uncertainties in the $\dot{Q}$ values as follows

```
HeatX2<- HeatX2 %>%
  mutate(
    u.Q.cold = C_p *sqrt( (T.cold.out - T.cold.in)^2 * (0.5/60)^2  + m.cold^2 * 2),
    u.Q.hot =  C_p *sqrt( (T.hot.out  - T.hot.in)^2  * (0.5/60)^2  +  m.hot^2 * 2)
  )
HeatX2
```

```
##   trial T.cold.in T.cold.out T.hot.in T.hot.out  m.cold   m.hot Q.cold  Q.hot      Q.env
## 1     1      14.3       18.6     38.2      33.9 0.16667 0.16667  2.996 -2.996 -2.220e-15
## 2     2      14.1       18.7     35.9      32.8 0.12500 0.16667  2.403 -2.160  2.438e-01
## 3     3      14.1       19.4     35.9      33.4 0.08333 0.16667  1.846 -1.742  1.045e-01
## 4     4      14.2       18.4     35.8      31.8 0.08333 0.08333  1.463 -1.393  6.967e-02
## 5     5      14.1       17.4     35.7      30.9 0.12500 0.08333  1.724 -1.672  5.225e-02
## 6     6      14.1       16.9     35.8      30.4 0.16667 0.08333  1.951 -1.881  6.967e-02
##   u.Q.cold u.Q.hot
## 1   0.9966  0.9966
## 2   0.7561  0.9911
## 3   0.5261  0.9891
## 4   0.5139  0.5119
## 5   0.7478  0.5202
## 6   0.9901  0.5273
```

Note: We are using the fact that $u_{\Delta T} = \sqrt{2}u_T$.

We can also use `deltaMethod()` in the `car` package to do this arithmetic for us. In fact, `deltaMathod()` can even handle cases where the estimates are not independent. To do it's job, `deltaMethod()` requires

- A named vector of estimates,

```
estimates <- HeatX[1, 2:7] %>% unlist()  # unlist() turns the data frame into a vector
estimates

##  T.cold.in T.cold.out   T.hot.in  T.hot.out     m.cold      m.hot
##    14.3000    18.6000    38.2000    33.9000     0.1667     0.1667
```

- An expression for the derived quantity as a quoted string (it will do the derivatives for us!),

```
exprforQ <- "(T.cold.out - T.cold.in) * C_p * m.cold"
```

- A variance-covariance matrix. In the simple case where things are independent, this is just a matrix with the squared uncertainties on the diagonal and 0 everywhere else.

```
vc <- diag( c(1,1,1,1,.5/60,.5/60)^2)
vc

##      [,1] [,2] [,3] [,4]      [,5]      [,6]
## [1,]    1    0    0    0 0.000e+00 0.000e+00
## [2,]    0    1    0    0 0.000e+00 0.000e+00
## [3,]    0    0    1    0 0.000e+00 0.000e+00
## [4,]    0    0    0    1 0.000e+00 0.000e+00
## [5,]    0    0    0    0 6.944e-05 0.000e+00
## [6,]    0    0    0    0 0.000e+00 6.944e-05
```

Let's see how it compares to the (first row of the) results above:

```
deltaMethod(estimates, exprforQ, vc)

##                                       Estimate    SE 2.5 % 97.5 %
## (T.cold.out - T.cold.in) * C_p * m.cold    2.996 0.997 1.042   4.95
```

The `deltaMethod` package adds additional interfaces for `deltaMethod()` to make this even easier.

```
library(deltaMethod)

## Error in library(deltaMethod):  there is no package called 'deltaMethod'

deltaMethod(HeatX, exprforQ,
  uncertainties = c(T.cold.in = 1.0, T.cold.out = 1.0, m.cold = 0.5 / 60))

## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

Dealing with uncertainties that are not constant

Sometimes the uncertainties of a given variable are different at different values. In that case, we can put the uncertainties into the data frame. For the current example, this is just extra work, but we include it here to show how it works.

```
HeatX3 <- HeatX %>%
  mutate(
    u.cold.in = 1, u.cold.out = 1, u.hot.in = 1, u.hot.out = 1,
    u.m.cold = 0.5/60, u.m.hot = 0.5/60
  )
HeatX3
```

```
##   trial T.cold.in T.cold.out T.hot.in T.hot.out  m.cold   m.hot u.cold.in u.cold.out
## 1     1      14.3       18.6     38.2      33.9 0.16667 0.16667         1          1
## 2     2      14.1       18.7     35.9      32.8 0.12500 0.16667         1          1
## 3     3      14.1       19.4     35.9      33.4 0.08333 0.16667         1          1
## 4     4      14.2       18.4     35.8      31.8 0.08333 0.08333         1          1
## 5     5      14.1       17.4     35.7      30.9 0.12500 0.08333         1          1
## 6     6      14.1       16.9     35.8      30.4 0.16667 0.08333         1          1
##   u.hot.in u.hot.out u.m.cold  u.m.hot
## 1        1         1 0.008333 0.008333
## 2        1         1 0.008333 0.008333
## 3        1         1 0.008333 0.008333
## 4        1         1 0.008333 0.008333
## 5        1         1 0.008333 0.008333
## 6        1         1 0.008333 0.008333
```

```
deltaMethod(HeatX3,
  exprforQ,
  estimates     = c("T.cold.in", "T.cold.out", "m.cold"),     # columns with estimates
  uncertainties = c("u.cold.in", "u.cold.out", "u.m.cold"))   # and uncertainties
```

```
## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

Although our uncertainties are the same in each row, this method allows for uncertainties to be specified separately for each row of the data.

10.1.6 Does the uncertainty in C_p matter?

In the analysis above, we treated C_p as a constant (with arbitrary precision). But this number is also known experimentally and has an uncertainty associated with it. Do our results change if we include this uncertainty in our calculations?

```
deltaMethod( HeatX %>% mutate(C_p = 4.18), exprforQ,      # add in a column for C_p
  uncertainties = c(T.cold.in = 1.0, T.cold.out = 1.0, m.cold = 0.5/60, C_p = 0.1))
```

```
## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

Comparing these results to the results above we see that the uncertainty increases, but only by amounts that are barely visible (a few parts per 1,000 – not enough to affect how we report the uncertainty given our rules for reporting digits). This has no meaningful impact on our analysis.

We can often simplify propagation of uncertainty calculations by identifying which components of the uncertainty are driving the size of the overall uncertainty. In this case, the imprecision in the estimated value of C_p is unimportant. If we want to improve our uncertainty, we must improve our measurements of temperature and/or flow rate.

10.1.7 Using relative uncertainty

Recall that for products, there is a Pythagorean relationship for relative uncertainties.

$$\frac{u_{\dot{Q}}}{\dot{Q}} \approx \sqrt{(\frac{u_{\dot{m}}}{\dot{m}})^2 + (\frac{u_{\Delta T}}{\Delta T})^2 + (\frac{u_{C_p}}{C_p})^2}$$

This is useful in determining what uncertainties contribute most to the overall uncertainty. For example, using the values in the first row of the data set.

- $\frac{u_{\dot{m}}}{\dot{m}} = \frac{0.5/60}{10/60} = 0.05$
- $\frac{u_{\Delta T}}{\Delta T} = \frac{\sqrt{2}}{4.5} = 0.314$
- $\frac{u_{C_p}}{C_p} = \frac{0.1}{4.18} = 0.024$

From this we can see that it is the imprecise temperature measurements that are our biggest problem. Even if we eliminated the other uncertainties, we would still have a relative uncertainty of over 30%. The details vary a bit from row to row, but the uncertainty in in temperature is our biggest obstacle. In addition to increasing the precision of our temperature sensors, we could also potentially improve things by designing a heat exchanger with more dramatic changes in temperature.

Suppose, for example, we could estimate temperature with an uncertainty of 0.1 degrees (ten times better than we have been assuming). Then our uncertainties for $\dot{Q}$ would change pretty dramatically.

```
deltaMethod( HeatX %>% mutate(C_p = 4.18), exprforQ,       # add in a column for C_p
  uncertainties = c(T.cold.in = 0.1, T.cold.out = 0.1, m.cold = 0.5/60, C_p = 0.1))


## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

But if we improve the uncertainty in the mass flow rate by a factor of 10, it has only a modest impact on our uncertainty for $\dot{Q}$.

```
deltaMethod( HeatX %>% mutate(C_p = 4.18), exprforQ,       # add in a column for C_p
  uncertainties = c(T.cold.in = 1.0, T.cold.out = 1.0, m.cold = 0.5/600, C_p = 0.1))


## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

10.1.8 Estimating the heat transfer coefficient, U

Next, we estimate a heat transfer coefficient (U) for both streams,

$$U = \frac{\dot{Q}}{A\Delta T_{lm}}, \tag{10.3}$$

where

$$\Delta T_{lm} \equiv \frac{(T_1 - T_4) - (T_2 - T_3)}{\log\left(\frac{T_1 - T_4}{T_2 - T_3}\right)}. \tag{10.4}$$

We can apply the same ideas to estimate the uncertainty in U. This time, working it out by hand would be considerably more tedious because of the number of variables involved and the form of the expression to be differentiated. Fortunately, R is happy to take care of those details if we just specify the information needed.

```
exprforU <- paste("C_p * m.cold * (T.cold.out - T.cold.in) /",
                  "( pi * D *  L  * ((T.cold.out - T.hot.in) - (T.cold.in - T.hot.out)) /",
                  " log ( (T.cold.out - T.hot.in) / (T.cold.in - T.hot.out) ) )")
HeatX4 <- HeatX %>%
  mutate(D = 0.0143, L = 1.626, C_p = 4.18)
HeatX4
```

```
##   trial T.cold.in T.cold.out T.hot.in T.hot.out  m.cold   m.hot      D     L  C_p
## 1     1      14.3       18.6     38.2      33.9 0.16667 0.16667 0.0143 1.626 4.18
## 2     2      14.1       18.7     35.9      32.8 0.12500 0.16667 0.0143 1.626 4.18
## 3     3      14.1       19.4     35.9      33.4 0.08333 0.16667 0.0143 1.626 4.18
## 4     4      14.2       18.4     35.8      31.8 0.08333 0.08333 0.0143 1.626 4.18
## 5     5      14.1       17.4     35.7      30.9 0.12500 0.08333 0.0143 1.626 4.18
## 6     6      14.1       16.9     35.8      30.4 0.16667 0.08333 0.0143 1.626 4.18
```

```
deltaMethod(HeatX4, exprforU,
  uncertainties = c(
        T.cold.in = 1.0, T.cold.out = 1.0, m.cold = 0.5/60,
        T.hot.in = 1.0, T.hot.out = 1.0, m.hot = 0.5/60,
        C_p = 0.1, D = 0.0004, L = 0.0006),
  constants = list(pi = pi)
)
```

```
## Error in as.vector(sqrt(t(gd) %*% vcov.  %*% gd)):  argument "vcov." is missing, with no default
```

Note the use of the `constants` argument here to specify the value of `pi`. We could also have specified C_p this way if we decided to ignore the uncertainty in that value. But since it is no harder to include that uncertainty, we included it.

10.1.9 Estimating the Nusselt number correlation, a

Finally, we can estimate the parameter a in a Nusselt number correlation for turbulent flow ($Re_D > 2300$):

$$a = \frac{Nu_D}{Re_D^{0.8} Pr^{1/3}}, \tag{10.5}$$

where

$$Nu_D = \frac{hD}{k} \approx \frac{2UD}{k}, \tag{10.6}$$

$$Re_D = \frac{4\dot{m}}{\pi D \mu}, \tag{10.7}$$

and

$$Pr = \frac{\mu C_p}{k}. \tag{10.8}$$

The uncertainty in the estimates for a can be estimated in a similar manner. For that purpose, we assume that the exponents on Re_D and Pr (0.8 and 1/3, respectively) are constant. The values of C_p, k, and μ are given in Table 10.1.

To complete our uncertainty calculation we must

1. Create an expression for the quantity we are interested in (as a quoted string).
2. Create a data frame that has all of the components of this expression that have uncertainties.
3. Create a named vector of uncertainties. The names should correspond to the variables names in the data frame, the values are the uncertainties. (Alternatively, the uncertainties can be put inside the data frame and the arguments `estimtates` and `uncertainties` can be used to specify which columns are the estimates and which are the corresponding uncertainties.)
4. Create a list that contains the value of any constants (or values for which we are ignoring that there is uncertainty because the uncertainty is so small that it doesn't affect the analysis).

This is left as an exercise for you.

10.2 Standard Errors in `fitdistr()` output

We had not yet learned about uncertainty and standard errors when we learned about `fitdistr()`, but the output from the function includes an estimated standard error.

```
library(fastR2)
fitdistr(Jordan8687$points, "normal")


##     mean       sd
##   37.0854    9.8640
##  ( 1.0893) ( 0.7702)
```

The parenthesized numbers are the estimated standard errors associated with each parameter estimate. In this case, it is easy to compute the standard error for the mean ourselves using $SE = s/\sqrt{n}$.

```
sd( ~ points, data = Jordan8687) / sqrt(82)


## [1] 1.096
```

10.3 R^2

Note: the methods in this section assume a linear model with an intercept term.

Recall that linear models are fit by minimizing the sum of the squares of the residuals:[3]

$$RSS = SSE = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

This expression reminds us of[4]

$$SST = \sum_{i=1}^{n} (y_i - \overline{y})^2 \ ,$$

the numerator of the variance of y. SST is a measure of the total variation in the response variable.

A little algebra shows that[5]

$$SSM = SST - SSE = \sum_{i=1}^{n} (\hat{y} - \overline{y})^2 \ ,$$

We can now define R^2 by

$$R^2 = \frac{SSM}{SST} = \frac{SSM}{SSM + SSE} = \text{proportion of variability in the response explained by the model.}$$

R^2 is always between 0 and 1 and is reported in the summary output for linear models. It is the square of the correlation coefficient that we saw earlier.

```
summary(lm(sat ~ expend, data = SAT))  # how does the average SAT score depend on money spent?
```

```
##
## Call:
## lm(formula = sat ~ expend, data = SAT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -145.07  -46.82    4.09   40.03  128.49
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1089.29      44.39   24.54   <2e-16
## expend        -20.89       7.33   -2.85   0.0064
##
## Residual standard error: 69.9 on 48 degrees of freedom
## Multiple R-squared:  0.145,Adjusted R-squared:  0.127
## F-statistic: 8.13 on 1 and 48 DF,  p-value: 0.00641
```

Notice that $R^2 < 15\%$ and that the coefficient on `expend` is negative – indicating that spending more is associated with *worse* scores on the SAT.

One reason for this is that in some states most college bound students take the SAT, but in other states, the ACT is more common so the pool of students taking the SAT is stronger. If we add `frac` – the fraction of students in a given state who took the SAT.

```
summary(lm(sat ~ expend + frac, data = SAT))
```

```
##
```

[3] RSS stands for Residual Sum of Squares; SSE stands for Error Sum of Squares.
[4] SST stands for Total Sum of Squares.
[5] SSM stands for Model Sum of Squares.

```
## Call:
## lm(formula = sat ~ expend + frac, data = SAT)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -88.40 -22.88   1.97  19.14  68.75
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  993.832     21.833   45.52   <2e-16
## expend        12.287      4.224    2.91   0.0055
## frac          -2.851      0.215  -13.25   <2e-16
##
## Residual standard error: 32.5 on 47 degrees of freedom
## Multiple R-squared:  0.819,Adjusted R-squared:  0.812
## F-statistic:  107 on 2 and 47 DF,  p-value: <2e-16
```

Notice how much larger R^2 is now, and that the sign of the coefficient on `expend` is not positive – as we would have expected.

It is important to note that adding in the additional variable `frac` gives a very different impression of the effect of `expend` on `sat`. Most of our examples have dealt with one response to one predictor, but in many situations, this is too simplistic and the inclusion of multiple predictors in the model is required to understand their impact on the response.

10.4 Exercises

10.1 Give an estimate with uncertainty for the Nusselt number correlation using the data in `HeatX` used in this chapter.

Bibliography

[Ans73] F. J. Anscombe, *Graphs in statistical analysis*, The American Statistician **27** (1973), no. 1, 17-21.

[Fis25] R. A. Fisher, *Statistical methods for research workers*, Oliver & Boyd, 1925.

[Fis70] ———, *Statistical methods for research workers*, 14th ed., Oliver & Boyd, 1970.

[MLVB11] Eugene C. Morgan, Matthew Lackner, Richard M. Vogel, and Laurie G. Baise, *Probability distributions for offshore wind speeds*, Energy Conversion and Management **52** (2011), no. 1, 15 – 26.

[MT77] F. Mosteller and J. Tukey, *Data analysis and regression*, Addison-Wesley, 1977.

[Stu08] Student, *The Probable Error of a Mean*, Biometrika **6** (1908), no. 1, 1–25.

Index

ACTgpa, 200
BallDrop, 186, 199
CPS85, 33
DAAG, 11
Devore7, 166
Drag, 200, 201
HELPrct, 12, 34, 46, 104
HeatX, 241
IQR(), 41
KidsFeet, 127, 166
MASS, 89, 91
Pendulum, 199
Pressure, 200
SnowGR, 34
Soap, 189, 199
Spheres, 201
Utilities, 201
alr4, 15, 33
anscombe, 167
antiD(), 69
balldrop, 194
bargraph(), 35
car, 233
confint(), 163, 191
cornit, 200
cornnit, 200
cut(), 37
data(), 11
data.frame(), 51
datasets, 48
dbinom(), 208
deal(), 61
deltaMathod(), 233
deltaMethod(), 233, 234
deltaMethod, 234
demo(), 32
dimes, 121
dim, 15
do(), 49, 62, 136, 205
drag, 200, 201
ex12.21, 166
faraway, 200
fastR2, 199–201
favstats(), 41
favstats, 127
fitdistr(), 89–92, 94, 103, 104, 123, 238
gf_bar(), 21, 38
gf_boxplot(), 21, 44
gf_col(), 38
gf_dens(), 67
gf_density(), 21, 67
gf_dhistogram(), 67, 91
gf_dist, 77
gf_fitdistr(), 91
gf_histogram(), 21, 90
gf_jitter(), 45
gf_labs(), 26
gf_point(), 17, 44
gf_qq(), 96, 99, 100, 104
gf_refine(), 181
ggformula, 16, 17, 19, 32, 77, 104
ggplot2, 33
glimps(), 13
inspect(), 13
integrate(), 69
iris, 48
knitr, 30
lm(), 152, 154, 192, 194
log(), 9, 180
log10(), 9, 180
log2(), 180
makeFun(), 155
max, 21
mean(), 41, 46
mean, 21
median(), 41

median, 21
min, 21
mosaic, 11, 12, 15, 21, 28, 33, 36, 69, 205
mosiac, 16
mpg, 33
mplot(), 160
mtable(), 37
ncol, 15
nls(), 192, 197, 198
nrow, 15
oldfaith, 33
pbinom(), 208
pendulum, 199
pheno, 167
plot(), 160
qbinom(), 208
qnorm(), 84
rbinom(), 208
read.csv(), 228
resample(), 50, 61, 135
resid(), 153, 159
rflip(), 49, 61, 205
sample(), 61
sd(), 41, 46
sd, 21
sqrt(), 9
subset(), 34, 225
summary(), 162
sum, 61
t.test(), 119, 127, 128
table(), 35–37
tally(), 35, 36, 38
triangle, 76
utilities, 201
var(), 41, 46
var, 21
xpnorm(), 84
xtabs(), 37

ACT, 200

Beta distribution, **85**
bulge rule, 185

cards, *58*
cdf, *see* cumulative distribution function
Central Limit Theorem, 111
coefficient of determinism, *188*
conditional probability, **56**
confidence level, 120
correlation coefficient, 153
coverage rate, 120
critical value, 120
cumulative distribution function, **71**

data frame, 12

Delta Method, **134**
density, **66**
density function, **68**
dice, *58*, *93*

exponential distribution, **79**

Fisher, R. A., 204

Gamma distribution, **80**
grade point average, 200

heteroskedasticity, 179
homoskedasticity, 179

independent events, **60**

kernel, **70**

ladder of re-expression, 185

medical testing, *59*

Newton's Law of Cooling, 197
normal distribution, **82**

observation unit, 12

parameter, 76
pdf, *see* probability density function
physics, *186*, *188*, 199–201
population, 203
probability density function, **68**

quantile-quantile plot, 96

relative uncertainty, 138
robustness, 122

sample, 203
soap, *189*

terminal velocity, 200
transformation
    of data, 178, 199
triangle distribution, **76**
triangular distribution, *see* triangle distribution
Tukey bulge, *185*, 199
Tukey, J., 185

uncertainty, **131**
uniform distribution, **78**

variable, 12

Weibull distribution, **80**
wine
    gentleman tasting, 225