STAT 243: Introductory Statistics

S DeRuiter & R Pruim (based on an original by R Pruim)

Spring 2019

Contents

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, often leading to a calculation that produced the number.

Thought experiment – How many coins?

Here's a thought experiment for you. Suppose a middle school class has collected a large number of coins in a sack. Before bringing the money to the bank, they would like to estimate how many coins 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 coins in the sack.

Some important terms

estimand/measureand The number we want to know. The "truth." In our example this is the number of coins 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 coins, and therefore overestimate the number of coins. 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 Rand RStudio are freely available for Mac, PC, and Linux.

We have set up an RStudio server on campus, which allows you to run R in a web browser on any computer without installing the software yourself. Your session is restored each time you log in, so you can work on multiple computers without losing your work when you move from one to the other. The RStudio server is the recommended interface for using R and RStudio for this course. You can access the RStudio server via a web browser. (For best results, avoid Internet Explorer.)

If you prefer to install R and RStudio directly on your own computer, you can get R at `<http://cran.r-project.org>` and RStudio at `<http://rstudio.org/>`.

To access the Calvin RStudio server, go to `http://rstudio.calvin.edu`.

1.1.1 Logging in

When you navigate to the RStudio server, you will be prompted to login. Your login and password are the same ones you would use for the Calvin CS servers, if you have previously taken a CS course. If not, you should have received an email at the beginning of the semester from "RStudio account creation robot" to your Calvin email account, telling you how to set up your password.

If you have forgotten your password and need to reset it, visit `https://cs.calvin.edu/sysadmin/linux-forgotpassword.php`

Once you are logged in, you will see something like Figure 1.1.

1.1.2 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.

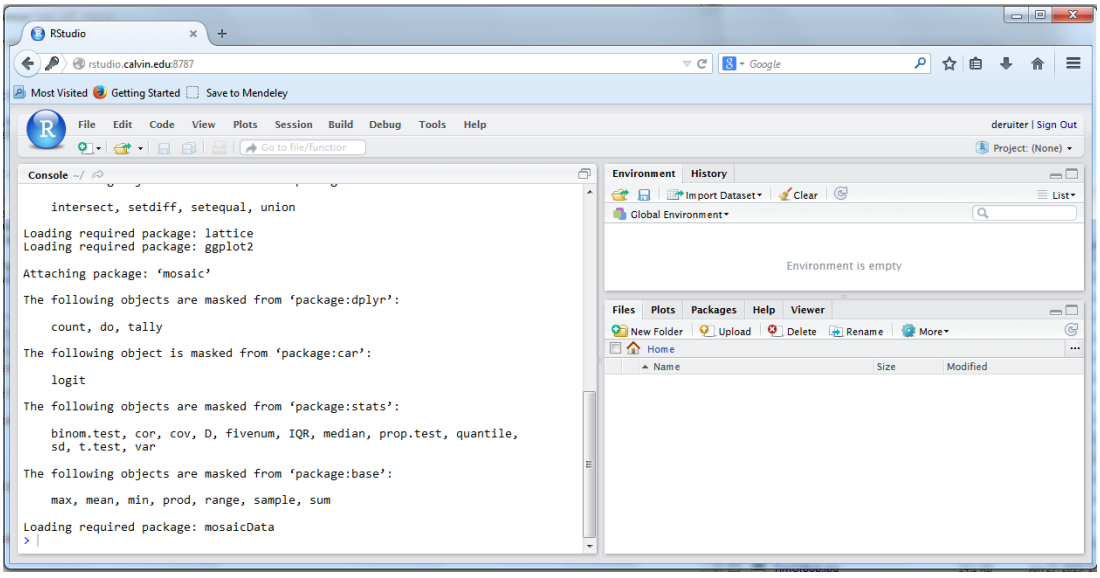


Figure 1.1: Welcome to RStudio.

```
5 + 3
```

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

```
15.3 * 23.4
```

```
## [1] 358.02
```

```
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] 358.02
```

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

```
## [1] 358.02
```

```
.5 * product                 # half of the product
```

```
## [1] 179.01
```

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

## [1] 5.880589

log10(product)              # base 10 log of the product

## [1] 2.553907

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

## [1] 8.483896
```

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

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

## [1] 358.02
```

1.1.3 Loading packages

R is divided up into packages. You can think of the packages as software toolkits designed to do particular jobs. A few of these, known as "base R", 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. There are two steps to follow before you can use a package in R:

1. Install the package. This operation downloads the relevant files to your computer, and lets R know where they are located. It does *not* give the current R session permission to use the tools contained in the package! The packages you will need for work in this course have already been installed on the Calvin RStudio server. For this course, you will probably not need to install any packages yourself, unless you are using a local copy of R and RStudio installed on your own computer. If you need to install packages, an easy way to do it is to use the **Packages** tab in the lower right panel of RStudio. Just click on **Install** (upper left corner of the **Packages** tab) and then type the name of the package.
2. Load the package. This operation gives the current R session permission to access and use the tools contained in the package. Even if you are using the RStudioserver, you will often need to load required packages at the beginning of each R session.

You an also load packages with commands like:

```
require(mosaic)       # loads the mosaic package if it is not already loaded
library(mosaic)       # library() and require() are essentially equivalent
```

1.1.4 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 printed out to the console window – which probably isn't what you want at this point.

3. 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 **cases** (or subjects when they are people). For measurements collected over time, the cases would be the individual time-points at which data points were collected. Each case is represented by one row in the data frame. The different pieces of information recorded for each case 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:

```
data(package="mosaicData")
data(package="DAAG")
```

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 (cases) 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. You can also ask R to count the rows and columns in the dataset.

```
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
## 1   housed 13 26  1     39          1  yes 25.111990 58.41369      0   black      no
## 2 homeless 56 62  2     43         NA <NA> 26.670307 36.03694      1   white      no
## 3   housed  0  0  3     41          0   no  6.762923 74.80633     13   black      no
## 4   housed  5  5  4     28          0   no 43.967880 61.93168     11   white     yes
## 5 homeless 10 13  5     38          1  yes 21.675755 37.34558     10   black      no
## 6   housed  4  4  6     29          0   no 55.508991 46.47521      5   black      no
##   sexrisk substance treat avg_drinks max_drinks
## 1       4   cocaine   yes         13         26
## 2       7   alcohol   yes         56         62
## 3       2    heroin    no          0          0
## 4       4    heroin    no          5          5
## 5       6   cocaine    no         10         13
## 6       5   cocaine   yes          4          4
```

```
nrow(HELPrct)
```

```
## [1] 453
```

```
ncol(HELPrct)
```

```
## [1] 29
```

The commands and R output above tell us that there are 453 observational units in this data set and 29 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, and another way to get a quick look at the variable names and data types in a dataset:

```
glimpse(KidsFeet)
```

```
## Observations: 39
## Variables: 8
## $ name       <fct> David, Lars, Zach, Josh, Lang, Scotty, Edward, Caitlin, Eleanor, D...
## $ birthmonth <int> 5, 10, 12, 1, 2, 3, 2, 6, 5, 9, 9, 3, 8, 3, 11, 4, 12, 3, 6, 3, 6,...
## $ birthyear  <int> 88, 87, 87, 88, 88, 88, 88, 88, 88, 88, 87, 88, 87, 88, 87, 88, 87...
## $ length     <dbl> 24.4, 25.4, 24.5, 25.2, 25.1, 25.7, 26.1, 23.0, 23.6, 22.9, 27.5, ...
## $ width      <dbl> 8.4, 8.8, 9.7, 9.8, 8.9, 9.7, 9.6, 8.8, 9.3, 8.8, 9.8, 8.9, 9.1, 9...
## $ sex        <fct> B, B, B, B, B, B, B, G, G, B, B, B, B, B, G, G, G, G, G, G, B, B, ...
## $ biggerfoot <fct> L, L, R, L, L, R, L, L, R, R, R, L, L, L, L, R, R, R, L, R, R, L, ...
## $ domhand    <fct> R, L, R, R, R, R, R, R, R, L, R, R, R, R, R, R, L, R, L, R, R, R, ...
```

1.2.5 The oldfaith data set

A final example data set comes from the alr3 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

```
require(alr3)
```

Once you have done that, you will have access to the data set containing information about eruptions of Old Faithful, a geyser in Yellowstone National Park.

```
glimpse(oldfaith)
```

```
## Observations: 270
## Variables: 2
## $ Duration <int> 216, 108, 200, 137, 272, 173, 282, 216, 117, 261, 110, 235, 252, 105...
## $ Interval <int> 79, 54, 74, 62, 85, 55, 88, 85, 51, 85, 54, 84, 78, 47, 83, 52, 62, ...
```

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 cases 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

For detailed examples of how to import data from a Google Sheet or how to upload a data set to RStudio, please review the relevant sections of the tutorial at `http://rsconnect.calvin.edu:3939/connect/#/apps/44/access`

1.3 Graphing the Distribution of One Variable

A **distribution** tells which values a variable takes on, and with what frequency. That is, the distribution answers two questions:

- What values?
- How often?

Several standard statistical graphs can help us see distributions visually.

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_boxplot()`, etc.)
- The name(s) of the variable(s) to plot
- The name of the data frame this variable is a part of.

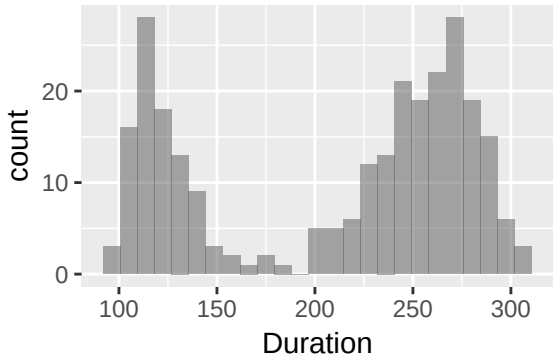
Note: The same syntax works for numerical summaries as well – thanks to the `mosaic` package we can apply the same syntax for `mean()`, `median()`, `sd()`, `var()`, `max()`, `min()`, etc. Later we will use this syntax again to fit linear and nonlinear models to data.

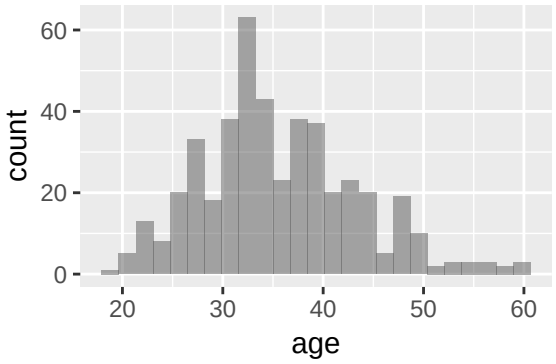
1.3.1 Histograms (and density plots) for quantitative variables

Histograms are a way of displaying the distribution of a quantitative variable.

Here are a couple examples:

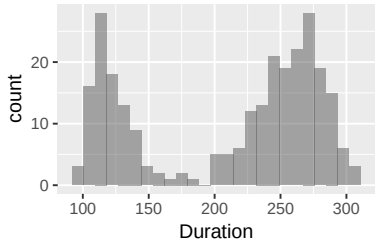
```
gf_histogram( ~ Duration, data=oldfaith )
gf_histogram( ~ age, data=HELPrct )
```

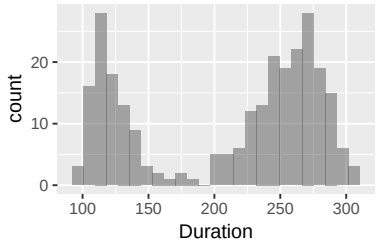


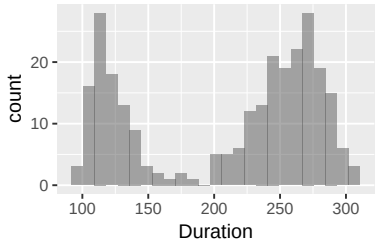


We can control the number of bins using the `bins` argument. The number of bins can make a histogram look quite different, and we aim to avoid using too few or too many.

```
gf_histogram( ~ Duration, data=oldfaith, n=15 )
gf_histogram( ~ Duration, data=oldfaith, n=30 )
gf_histogram( ~ Duration, data=oldfaith, n=50 )
```

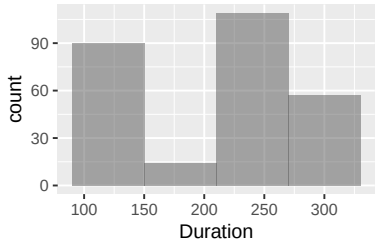


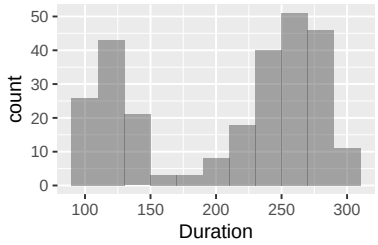


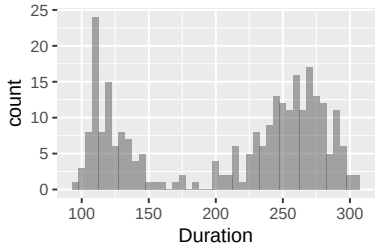


Another way to control the position of the bins is to set their width (instead of the overall number of bins). For example,

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

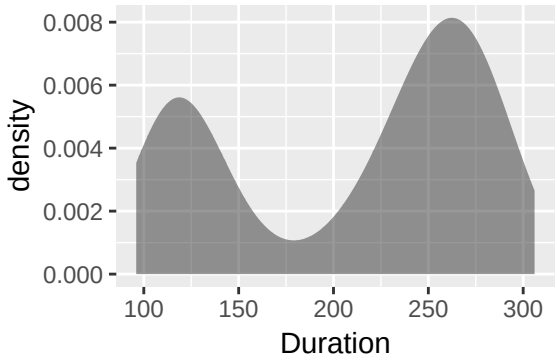






R also provides a "smooth" version called a density plot; just change the function name from `gf_histogram()` to `gf_density()`.

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



1.3.2 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, compare several histograms with different bin settings before you decide which one provides the most informative summary of the data.)

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. So right-skewed (also known as positive-skewed) data have a "fat right tail" – more observations of larger values than of small ones.)

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 "peak" where there is a lot of data.

bimodal There are two peaks.

multimodal There are more than two peaks.

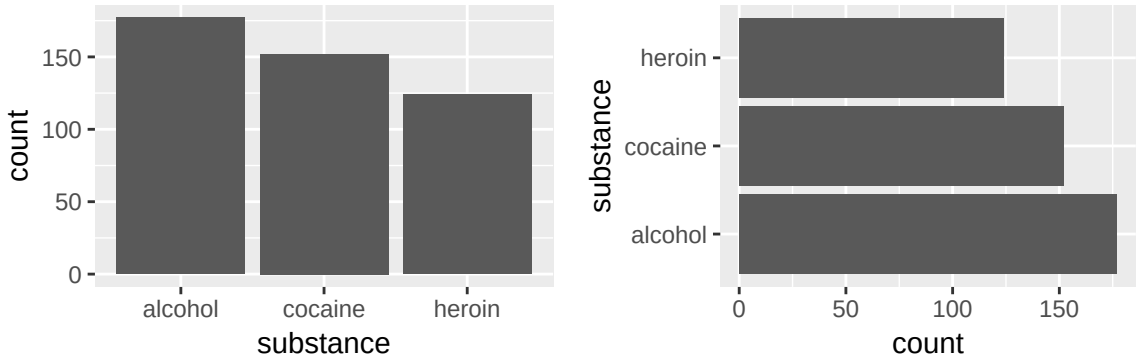
outlier An observation that does not fit the overall pattern of the rest of the data.

We'll learn about another graph used for quantitative variables (a boxplot, `gf_boxplot()` in R) soon.

1.3.3 Bar graphs for categorical variables

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

```
gf_bar( ~ substance, data=HELPrct)
gf_bar( ~ substance, data=HELPrct) %>%
  gf_refine(coord_flip())
```



A side note: we usually prefer bar graphs to pie charts in this course. Many data analysts argue that pie charts are difficult to read and interpret, and often use space ineffectively, especially if they are divided into more than two slices. Unless you are *sure* there is a good reason to use one, don't. See `http://rsconnect.calvin.edu:3939/connect/#/apps/94/access` for and example if you want to give it a try.

1.4 Looking at Multiple Variables at Once

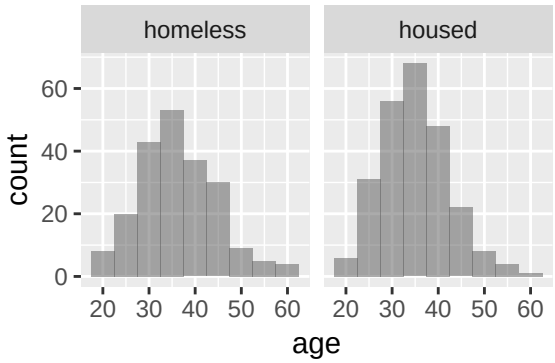
1.4.1 Conditional plots

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

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

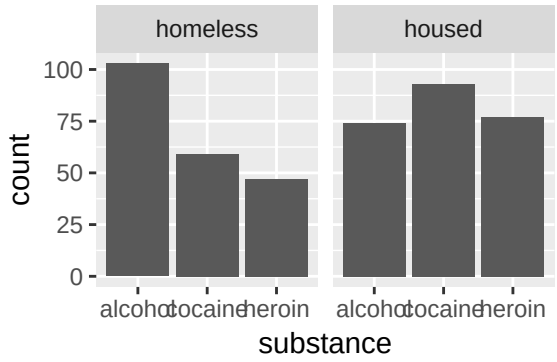
For example, we might like to see how the ages of homeless and housed people compare in the HELP study.

```
gf_histogram( ~ age | homeless, data=HELPrct, binwidth=5)
```



We can do the same thing for bar graphs.

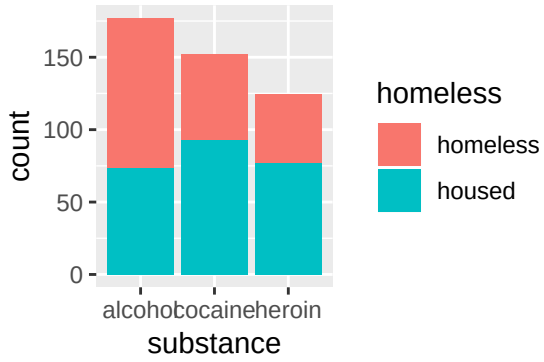
```
gf_bar( ~ substance | homeless, data=HELPrct)
```



1.4.2 Stacked or Side-By-Side Bar Graphs

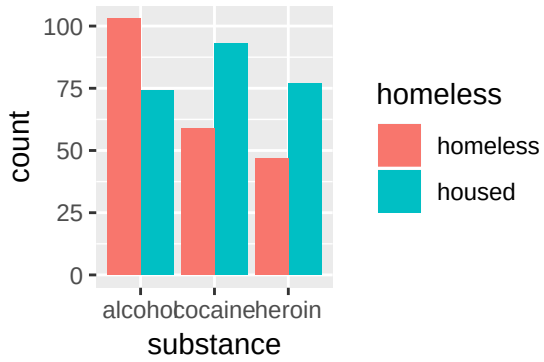
Instead of showing bar graphs for different groups in separate panels or "facets", as shown above, we might want to use a stacked bar graph:

```
gf_bar( ~ substance, fill = ~ homeless, data = HELPrct)
```



Similarly, we can use a grouped bar graph with side-by-side sets of bars:

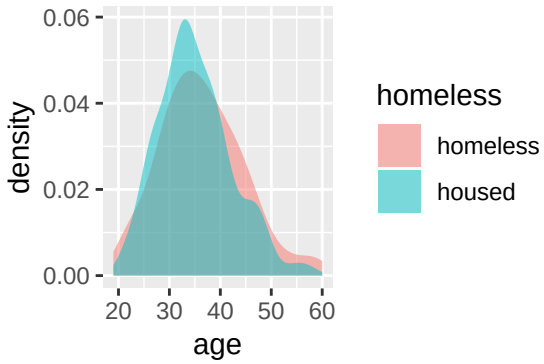
```
gf_bar( ~ substance, fill = ~ homeless, position = 'dodge', data = HELPrct)
```



1.4.3 Side-by-side Density Curves

We can do something similar with density curves to show individual curves for each group:

```
gf_density(~age, fill = ~homeless, data=HELPrct)
```



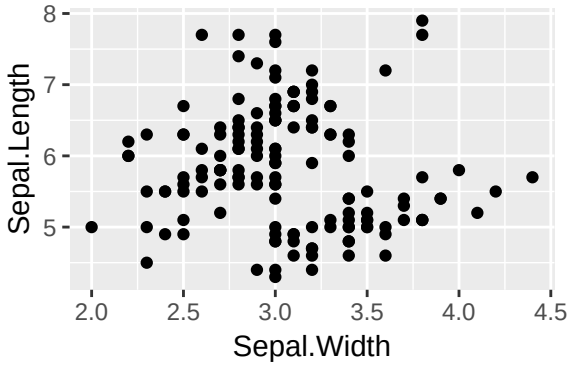
1.4.4 Scatterplots

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

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

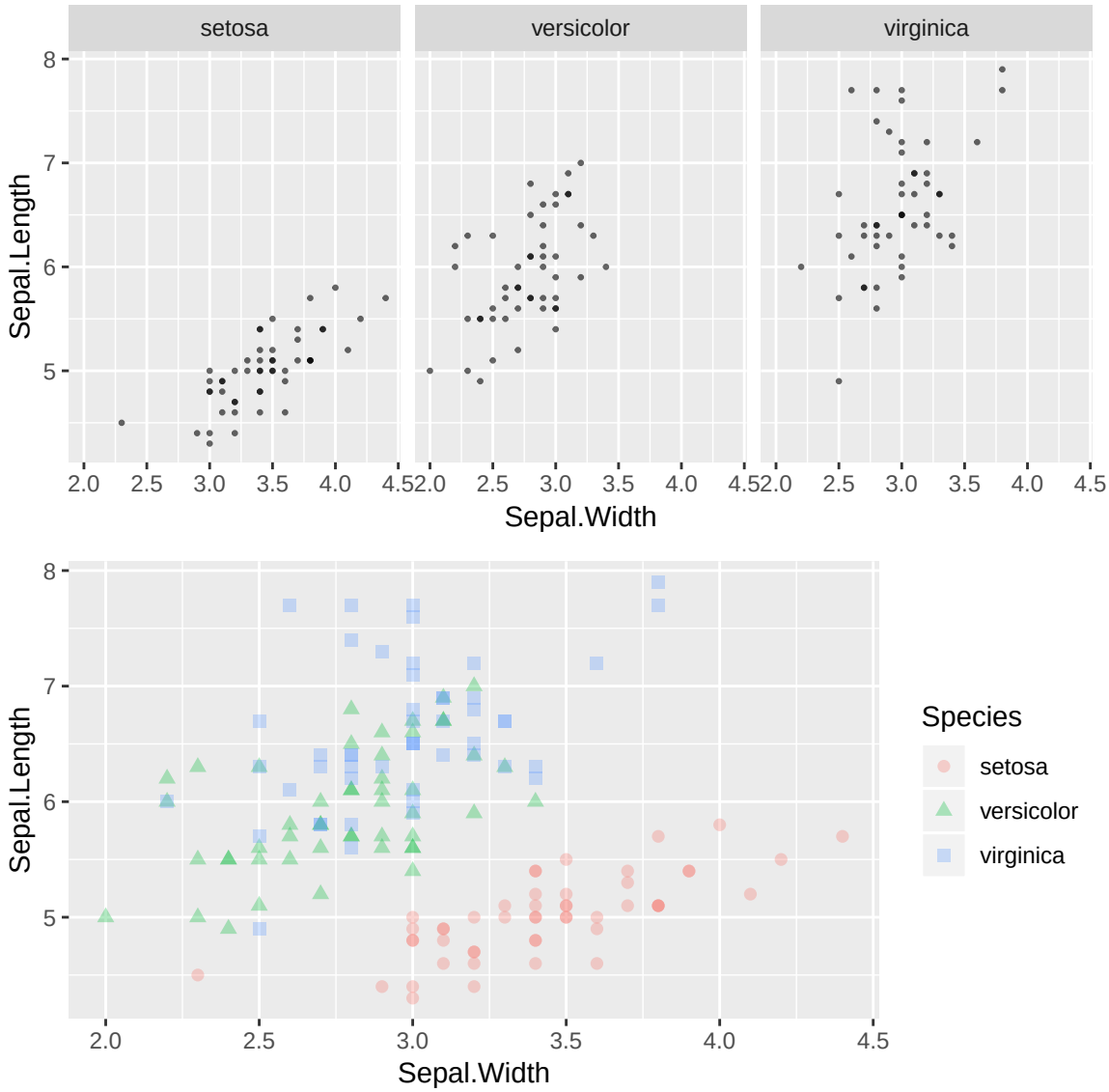
Notice that now we have something on both sides of the ~ since we need to tell R about two variables.

```
head(iris) # data on iris plants


##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa


gf_point( Sepal.Length ~ Sepal.Width, data=iris )
```



Grouping and conditioning work just as before and can be used to see the relationship between sepal length and sepal width broken down by species of iris plant. With large data set, it can be helpful to make the dots semi-transparent so it is easier to see where there are overlaps. This is done with `alpha`. We can also make the dots smaller (or larger) using `size` (multaplicative; for example, 2 means double the usual size).

```
gf_point( Sepal.Length ~ Sepal.Width | Species, data=iris, alpha=.6, size=.5 )
gf_point( Sepal.Length ~ Sepal.Width, color = ~ Species, shape = ~ Species, data=iris,
          alpha=.3, size = 2)
```



1.5 Reproducible Research

When starting to learn to use R for data analysis, it may be tempting to work by typing commands into the R console directly, or maybe by copying and pasting commands from some other source (for example, these notes, a website, etc.).

There are many reasons to avoid working this way, including:

- It is tedious, unless there is very little to type, or to copy and paste.

- It is error-prone – it's easy to copy too 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 everything you copied and pasted, or typed into the R console).

So while copy and paste seems easy and convenient at first, it is not *reproducible*. Reproducible, here, means something that can easily be repeated in exactly the same way (or with some desired modification), because the exact procedure that was followed has been clearly documented in a format that is simple to access. Reproducibility is important when projects are large, when it is important to have record of exactly what was done, or when the same analysis will be 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.

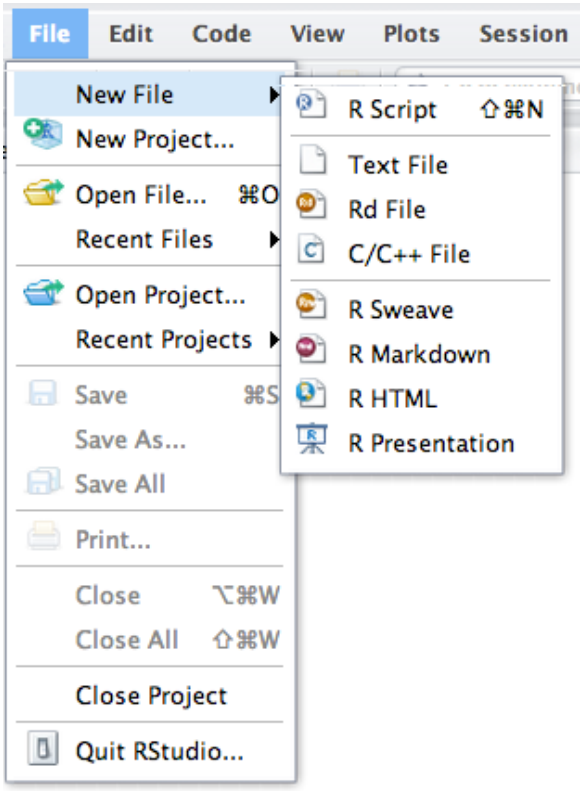
1.5.1 R Markdown

One simple way to do reproducible work is to use 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 (R commands and output, including figures). The end product is a PDF or an HTML file, so it is especially good for producing web documents.[1]

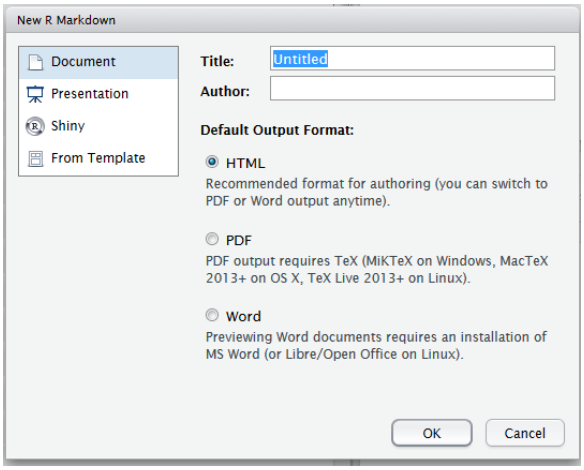
Creating a new document

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

[1] 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 we will focus on the basics.

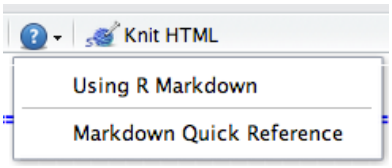


A small pop-up window will appear; a great choice for this course is to select"From Template" and then "Template for STAT 241 Homework".



When you do this, a file editing pane will open with a template inserted. If you click on "Knit", RStudio will turn this into a PDF 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 (at the top of the Markdown document in the editing pane).



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 "Insert Chunk" from the "Code" menu, or clicking the little green square with a "C" in it (and choosing "R").

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 PDF or 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.) Within each R Markdown file, you must explicitly load data, and require packages *in the R Markdown file* in order to use them. In this class, this means that most of your R Markdown files will have a chunk near the beginning that loads required packages and datasets.

R Markdown files do not have access to the console environment

One thing you need to remember about R Markdown documents is that the file must be self-contained. This ensures that the document is portable. It also means that the docuemnt does not have access to the things in your console environment. All data must be loaded in the file. Similarly, all packages you use must also be loaded in the file. If you start getting messages about objects not being found, one possible cause is that you have forgotten to get some data or some package loaded inside your file. (Typos are another cause for these messages – check your spelling and capitalization.)

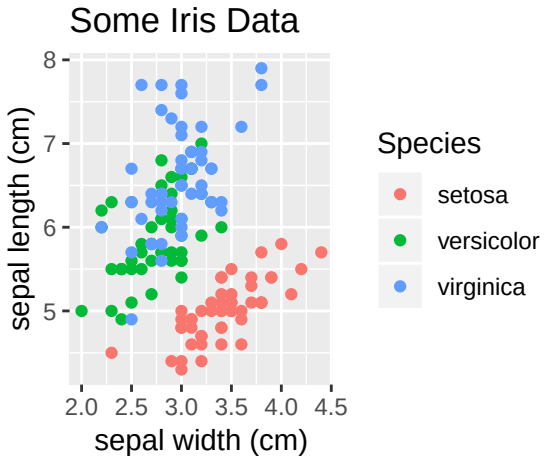
1.6 Customizing Graphics: A Few Bells and Whistles

There are lots of arguments that control the appearance of plots created in R. Here are just a few examples, some of which we have already seen.

Labels

You can add a title, or change the default labels of the axes.

```
gf_point(Sepal.Length ~ Sepal.Width, color=~Species, data=iris) %>%
        gf_labs(title="Some Iris Data",
                x="sepal width (cm)",
                y="sepal length (cm)")
```



1.6.1 More

Nearly every feature of a plot can be controlled: fonts, colors, symbols, line thicknesses, colors, etc. These settings can also be collected into a theme (for example, there is a theme that makes your plots look just like they came from *The Economist*).

For details and examples, review the tutorials at `http://rsconnect.calvin.edu:3939/connect/#/apps/45/access` and `http://rsconnect.calvin.edu:3939/connect/#/apps/46/access`.

1.7 Getting Help in RStudio

1.7.1 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:

```
?gf_bar
?gf_point
?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.

1.7.2 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.

1.7.3 Error messages

When things go wrong, R tries to help you out by providing an error message. Typos are probably the most common cause of errors: for example, you might misspell a function or argument name, forget to close a set of parentheses or brackets, or misplace a comma. 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`. Another common mistake is forgetting to load required packages. If you see an "object not found" message, check your typing and check to make sure that the necessary packages have been loaded. If you get an error and 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.

1.8 Graphical Summaries – Important Ideas

1.8.1 The Most Important Template

The plots we have created have all following 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).
- `data`: 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.

1.8.2 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.

1.8.3 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. Later, when we do more quantitative 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.

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

Some plots divide data into groups and either produce a panel for each group (using |) or display each group in a different way (different colors or symbols). These plots can reveal the possible influence of additional variables – sometimes called covariates.

1.8.5 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, 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.

Exercises

In your answers to these questions, include both the plots and the code you used to make them as well as any required discussion. Once you have obtained a basic plot that satisfies the requirements of the question, feel free to use some of the "bells and whistles" to make the plots even better.

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

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

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

1.4 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.

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

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

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

1.8 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.

1.9 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:

```
Fires <- read.csv("http://www.calvin.edu/~rpruim/data/Fires.csv")
```

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 `filter()`. For example, to get just the subset of years since 1966, you would use

```
Fires2 <- Fires %>%
  filter( Year > 1966)
```

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

Use `filter()` 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 you describe what is happening with fires over time?

1.10 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)?

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

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

1.13 The `SnowGR` contains historical data on snowfall in Grand Rapids, MI. The snowfall totals for November and December 2014 were 31 inches and 1 inch, respectively.

a) Create histograms of November and December snowfall totals. How unusual were the snowfall totals we had in 2014?

b) If there is very little snow in December, should we expect to have unusually much or little snow in February? Make a scatter plot comparing December and February historic 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( ~ substance, data=HELPrct )

## substance
## alcohol cocaine  heroin
##     177     152     124

tally( ~ substance, data=HELPrct, format="prop" )

## substance
##   alcohol   cocaine    heroin
## 0.3907285 0.3355408 0.2737307

tally( ~ substance, data=HELPrct, format="perc" )

## substance
##  alcohol  cocaine   heroin
## 39.07285 33.55408 27.37307
```

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
```

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

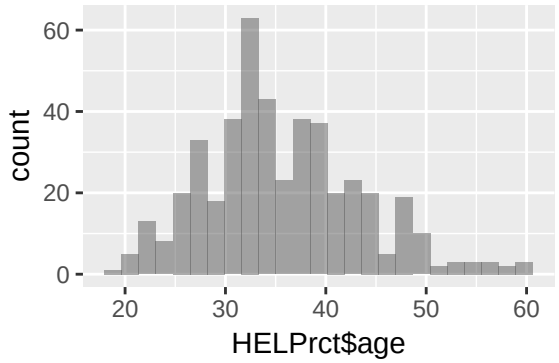
```
table( HELPrct$substance )

##
## alcohol cocaine  heroin
##     177     152     124

perctable( HELPrct$substance )     # display percents instead of counts

## Error: 'perctable' is defunct.
## Use 'tally' instead.
## See help("Defunct")

proptable( HELPrct$substance )     # display proportions instead of counts

## Warning: 'proptable' is deprecated.
## Use 'tally' instead.
## See help("Deprecated")

##
##   alcohol   cocaine    heroin
## 0.3907285 0.3355408 0.2737307
```

Two interfaces

Some functions in R require the formula interface, some require the `$`-interface, and some allow you to use either one.[1] For example, histogram will also work like this.

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

```
gf_histogram( ~HELPrct$age )
```



But notice that the output is not quite as nice, since the default label for the horizontal axis now shows both the data frame name and the variable name with a `$` between. *My advice is to use formula interfaces whenever they are available.*

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
```

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
## 1   housed 13 26  1     39          1  yes 25.111990 58.41369      0   black      no
## 2 homeless 56 62  2     43         NA <NA> 26.670307 36.03694      1   white      no
## 3   housed  0  0  3     41          0   no  6.762923 74.80633     13   black      no
## 4   housed  5  5  4     28          0   no 43.967880 61.93168     11   white     yes
## 5 homeless 10 13  5     38          1  yes 21.675755 37.34558     10   black      no
## 6   housed  4  4  6     29          0   no 55.508991 46.47521      5   black      no
##   sexrisk substance treat avg_drinks max_drinks binnedAge
## 1       4   cocaine   yes         13         26   (30,40]
## 2       7   alcohol   yes         56         62   (30,40]
## 3       2    heroin    no          0          0   (20,30]
## 4       4    heroin    no          5          5   (30,40]
## 5       6   cocaine    no         10         13   (30,40]
## 6       5   cocaine   yes          4          4   (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
```

```
table( 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
```

```
table( 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
```

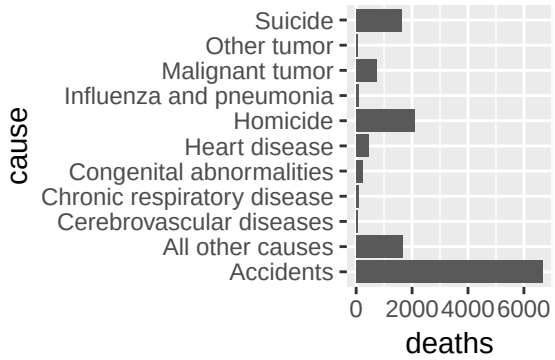
2.2 Working with Pre-Tabulated Data

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

```
require(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
```

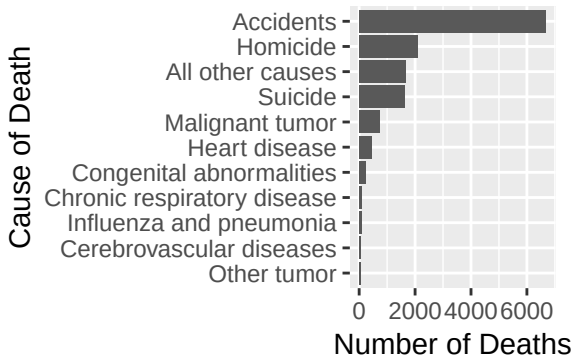
```
gf_col(deaths ~ cause, data=TeenDeaths) %>%
  gf_refine(coord_flip())
```



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 ~ fct_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

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

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

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

- 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.

[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.

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] 35.65342

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

## [1] 35.65342
```

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

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

## [1] 35.65342

sd(~ age, data=HELPrct)

## [1] 7.710266

var(~ age, data=HELPrct)

## [1] 59.4482
```

```
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 35.65342 7.710266 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`.

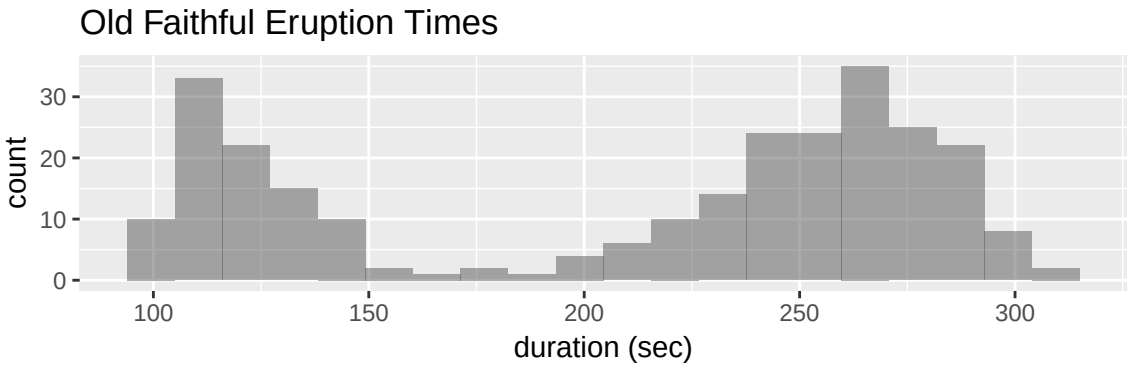
```
mean(age ~ sex, data=HELPrct)


##   female     male
## 36.25234 35.46821
```

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


##   female     male
## 7.584858 7.750110


favstats( ~ age | sex, data=HELPrct )


##      sex min Q1 median   Q3 max     mean       sd   n missing
## 1 female  21 31     35 40.5  58 36.25234 7.584858 107       0
## 2   male  19 30     35 40.0  60 35.46821 7.750110 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 set of data if the distribution of the data is not unimodal. The histogram below shows the lengths of eruptions of the Old Faithful geyser at Yellowstone National Park.

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


##  min  Q1 median     Q3 max     mean       sd   n missing
##   96 130    240 267.75 306 209.8778 68.39213 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.)

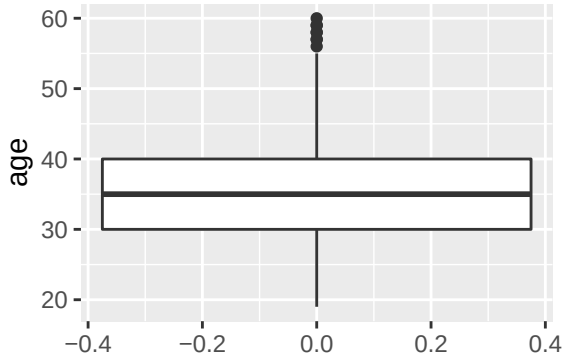
2.5.2 Box plots

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)

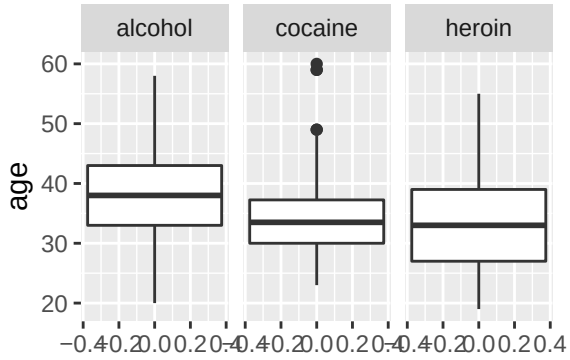
- Q_2, the median (50th percentile)
- Q_3, the third quartile (75th percentile)
- Q_4, the maximum

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



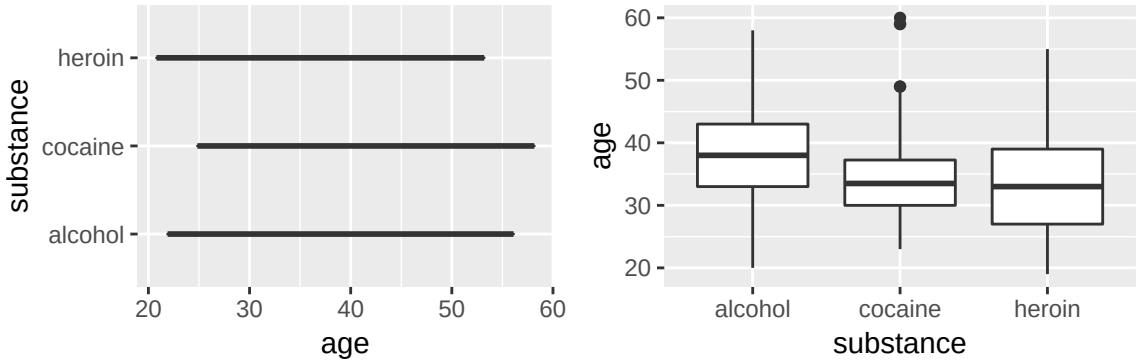
Boxplots provide a way of comparing multiple groups that is especially informative and visually effective. Here is one way to make boxplots of multiple groups (it should look familiar from what we know about histogram):

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



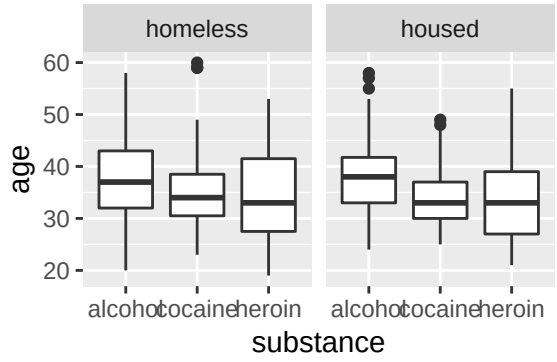
But `gf_boxplot()` has a better way. Put the quantitative variable on one side of the wiggle and the categorical on the other. The placement determines which goes along the vertical axis and which along the horizontal axis – just like it did for `gf_point()`.

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

## Warning: position_dodge requires non-overlapping x intervals

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.

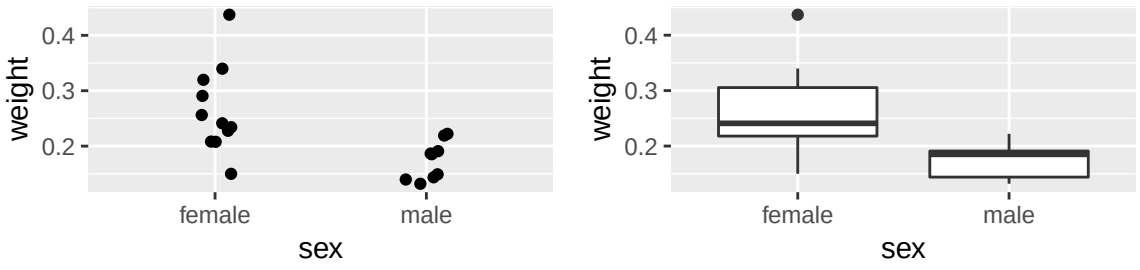
```
gf_boxplot(age ~ substance | homeless, data=HELPrct)
```



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 is 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)
gf_boxplot( weight ~ sex, data=Mosquitoes)
```



Note the effect of the `width` argument – `gf_jitter()` moves each data point slightly up or down, to reduce

overplotting (data points being plotted exactly on top of one another) and make it clearer how many data-points were observed for each possible combination of x- and y-values. The `width` input controls the amount of "jittering."

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.*

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

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.

Practice 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.

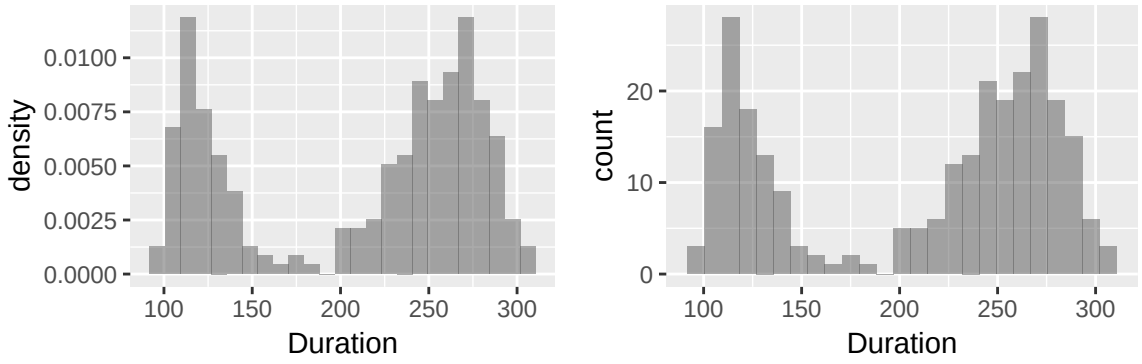
2.4 Compute the mean and median values of `i1` (average number of drinks per day) for each of the `substance` groups in the `HELPrct` data frame.

3 Densities

3.1 Density histograms, density plots, density functions

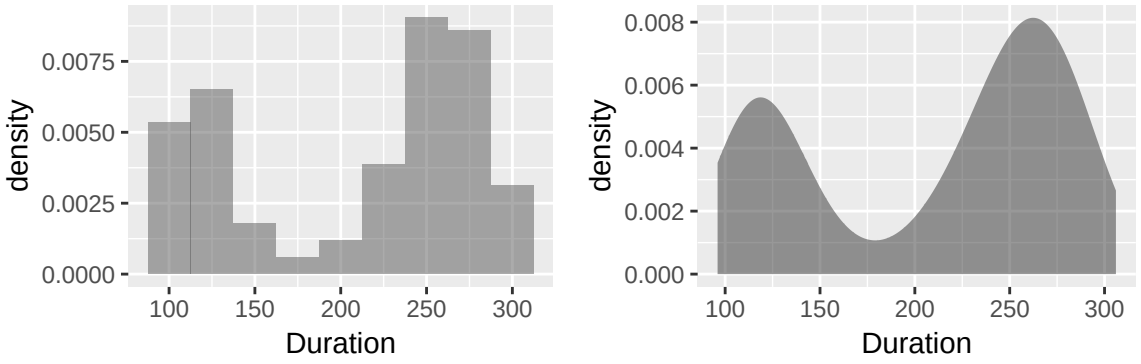
A histogram is a simple picture describing the "density" of data. Histogram bars are tall in regions where there is more data – i.e., where the data are more "dense".

```
require(alr3)
gf_dhistogram( ~ Duration, data=oldfaith )
gf_histogram( ~ Duration, data=oldfaith, type="count" )
```



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

```
require(alr3)
gf_dhistogram( ~ Duration, data=oldfaith, binwidth=25 )
gf_density( ~ 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.

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. $\displaystyle\int_{-\infty}^{\infty} f(x)\, dx = 1$.

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 3.1.1. Let f be defined by

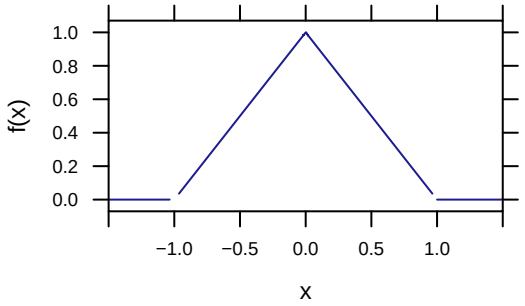
$$f(x) = \begin{cases} 1 - |x| & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

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.[1]

```
f <- makeFun( (1 - abs(x)) * (abs(x) <= 1) ~ x )
plotFun( f(x) ~ x, x.lim=c(-1.5, 1.5) )
```



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
integrate( f, -.5, .5 )$value

## [1] 0.75
```

[1] 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.

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
```

If we help R choose the anti-derivative, we get a useful function called the **cumulative distribution function**, abbreviated cdf.

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. The derivative of the cdf is the pdf.

Example 3.1.2. Continuing with our previous example, if we choose -1 as our lower endpoint, then the anti-derivative will be the cdf.

```
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
```

3.2 Working with Probability Density Functions

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.

3.2.1 Kernels

The **kernel** of a 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 the appropriate 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 3.2.1. 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

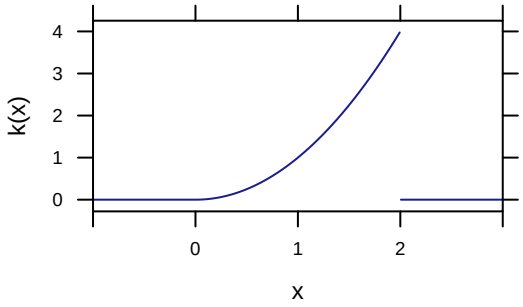
$$\int_{-\infty}^{\infty} k(x)\, dx\,.$$

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

```
## 2.666667 with absolute error < 3e-14
```

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

```
## [1] 2.666667
```



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

3.2.2 The mean of a continuous random variable

The definition for the mean of a continuous random variable will be motivated by the calculation of a mean of some data. **Example 3.2.2.** 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**.

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

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.[2]

Example 3.2.3. Q. Compute the mean of our triangle distribution from Example 3.1.1.

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.

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
```

3.2.3 The variance of a continuous random variable

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

$$\mathrm{Var}(X) = \sigma_X^2 = \int_{-\infty}^{\infty} (x - \mu_X)^2 f(x)\, dx$$

Note: It is possible that the integral used to define the variance will fail to converge. In that case, we say that

[2] 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.

the random variable has no variance or that the variance fails to exist.[3]

Example 3.2.4. Q. Compute the variance of the triangle random variable from the Example 3.1.1.

A.

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


## 0.1666667 with absolute error < 1.9e-15


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


## [1] 0.1666667
```

Some simple algebraic manipulations of the integral above shows that

$$\mathrm{Var}(X) = \mathrm{E}(X^2) - \mathrm{E}(X)^2 \tag{3.1}$$

Example 3.2.5. 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 3.2.1.

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.

[3] Actually, we will require that $\int_{\infty}^{\infty} |x|^2 f(x)\, dx$ converges. If this integral fails to converge, we will say that the distribution has no variance.

3.2.4 Quantiles

Quantiles solve equations of the form

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

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

$$\mathrm{P}(X \leq x) = 0.5\ .$$

Example 3.2.6. Q. What is the 25th percentile of the triangle distribution in Example 3.1.1?

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

$$0.25 = \mathrm{P}(X \leq x)\ .$$

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, $x = \frac{1}{2}\sqrt{2} - 1 = -0.2928932$.

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 $x = -1 + y$.

3.3 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.

3.3.1 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

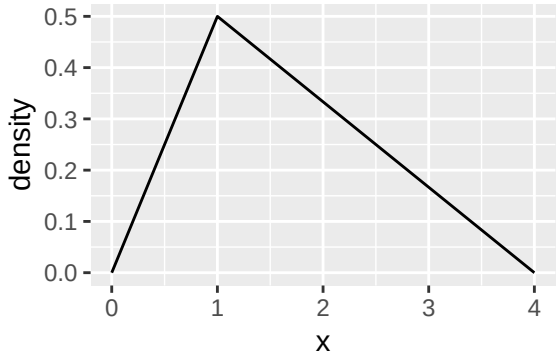
Function	What it does
`dtriangle(x,a,b,c)`	Computes value of the pdf at `x`
`ptriangle(q,a,b,c)`	Computes value of the cdf at `x`, i.e., $\mathrm{P}(X \le \mathtt{q})$
`qtriangle(p,a,b,c)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \le \mathtt{q}) = \mathtt{p}$
`rtriangle(n,a,b,c)`	Randomly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

Example 3.3.1. Q. Let $X \sim \mathsf{Triangle}(0, 4, 1)$. Use R to answer the following questions.

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

A. The `gf_dist()` function in the `mosaic` 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 $\mathsf{Triangle}(0, 4, 1)$-distribution.

```
gf_dist("triangle", params=list(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] 0.6666667
```

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

```
## [1] 1.55051
```

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

```
## [1] 1.666667
```

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

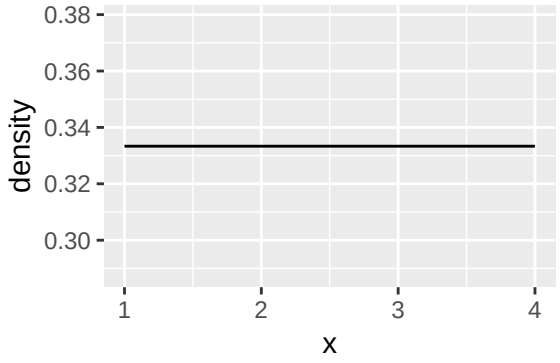
```
## 1.666667 with absolute error < 1.9e-14
```

3.3.2 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 $X \sim \mathsf{Unif}(a, b)$ 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 $\mathsf{Unif}(1, 4)$ distribution.

```
gf_dist(dist="unif", params=list(min=1, max=4), xlim=c(0,5))  # using parameter names
```



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., $P(X \leq x)$
`qunif(p,min,max)`	Computes quantiles, that is a value of x so that $P(X \leq x) = p$
`runif(n,min,max)`	Randomly samples `n` values from the Unif(`min`, `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 3.3.2. 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] 0.3333333
```

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

```
## [1] 0.3333333
```

```
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 $x = 3(0.8 + 1/3) = 2.4 + 1 = 3.4$.

3.3.3 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 3.3.3. 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

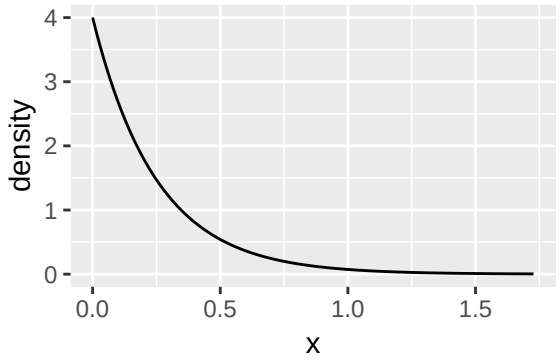
$$k(x;\lambda) = e^{-\lambda x}\,[\![x \geq 0]\!]$$

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., $\mathrm{P}(X \leq \mathtt{q})$
`qexp(p,rate)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \leq \mathtt{q}) = \mathtt{p}$
`rexp(n,rate)`	Randomly samples `n` values from the $\mathsf{Exp}(\lambda)$ distribution

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



3.3.4 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
Gamma(α, λ)	$k(x) = x^{\alpha-1}e^{-\lambda x}\ [\![x \geq 0]\!]$
Weibull(α, λ)	$k(x) = x^{\alpha-1}e^{-\lambda x^\alpha}\ [\![x \geq 0]\!]$

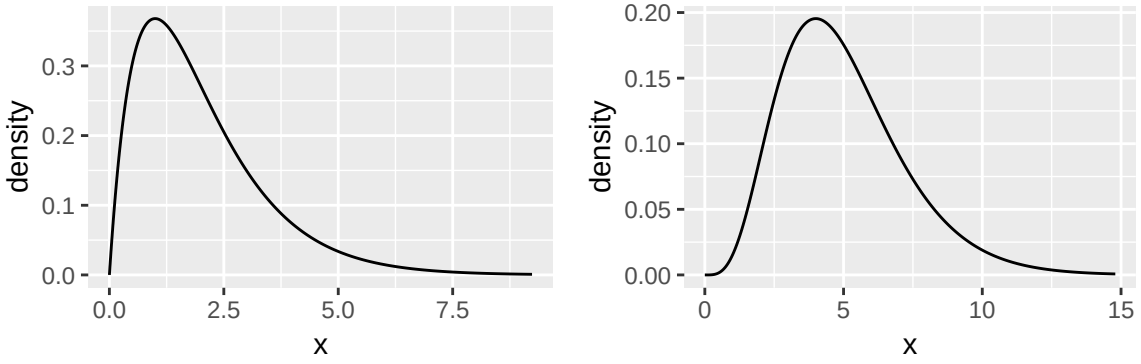
Both families of distributions are supported on the interval $[0, \infty)$.) For the most part, we won't use these formulas in calculations, preferring to let R do the work for us. However, notice that each of these distributions has a pdf that allows for relatively simple integration. For the Gamma 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.

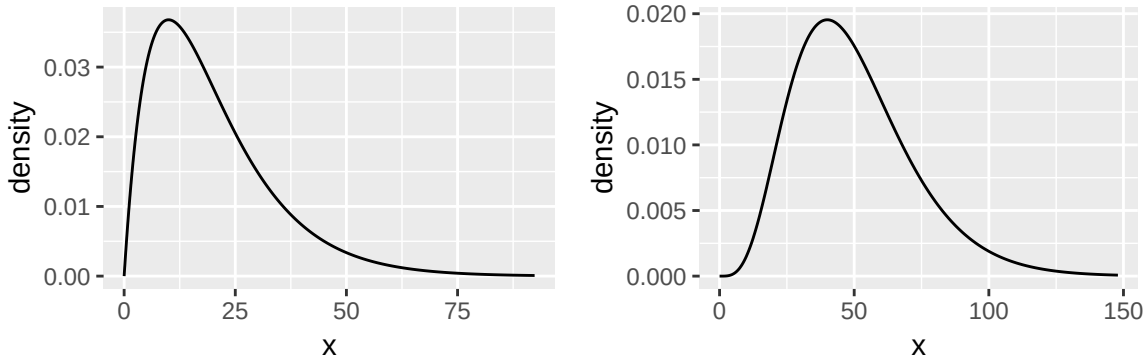
The now familiar functions are available for each of these distributions.

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.

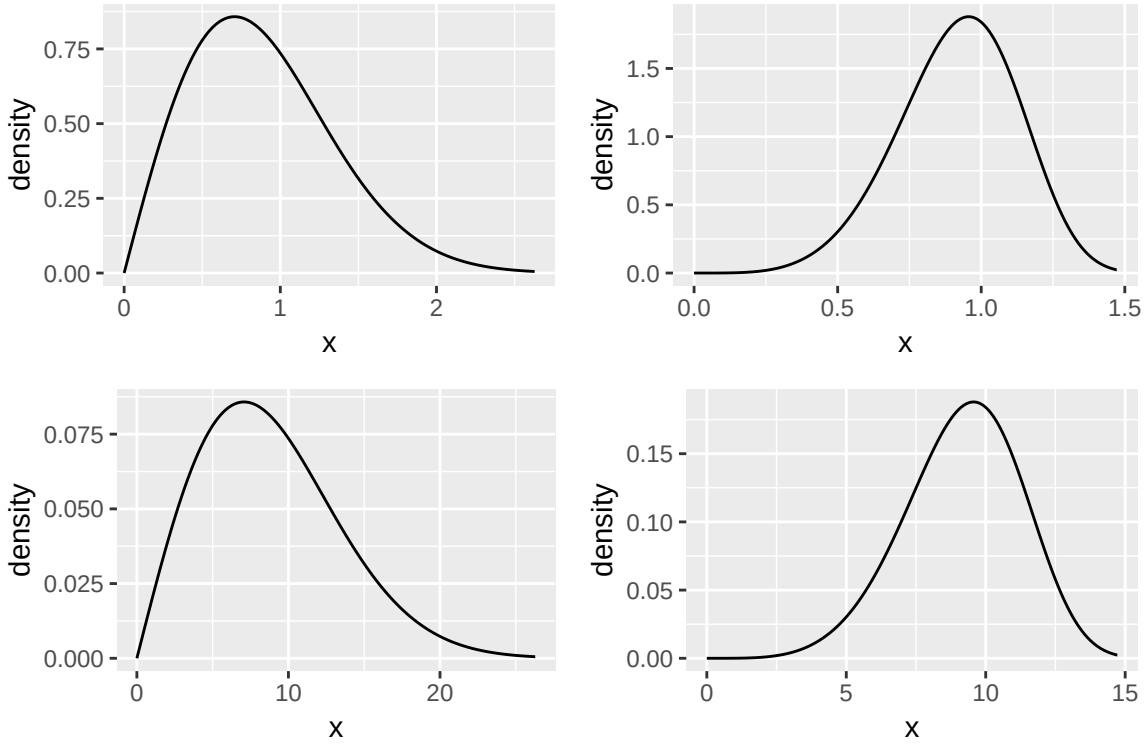
Like the exponential distributions, these distributions are skewed and only take on positive values. These distributions arise in many applications, including as more general models for lifetime. As the pictures below indicate, the shape and scale parameters are aptly named.

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





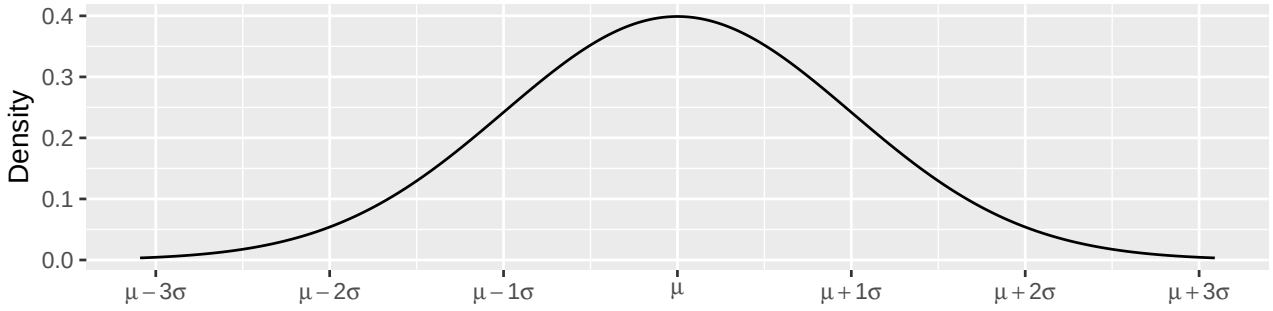

```
gf_dist("weibull", params=list(shape=2, scale=1),main="Weibull(2,1)")
gf_dist("weibull", params=list(shape=5, scale=1),main="Weibull(5,1)")
gf_dist("weibull", params=list(shape=2, scale=10),main="Weibull(2,10)")
gf_dist("weibull", params=list(shape=5, scale=10),main="Weibull(5,10)")
```



3.3.5 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
$\mathsf{Norm}(\mu, \sigma)$	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$



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 $Z \sim \mathsf{Norm}(0, 1)$ and $Y = a + bX$, then $Y \sim \mathsf{Norm}(a, b)$. Conversely, if $Y \sim \mathsf{Norm}(\mu, \sigma)$, then $Z = \frac{Y-\mu}{\sigma} \sim \mathsf{Norm}(0, 1)$.

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., $\mathrm{P}(X \leq \mathtt{q})$
`qnorm(p,mean,sd)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \leq \mathtt{q}) = \mathtt{p}$
`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 3.3.4. 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.

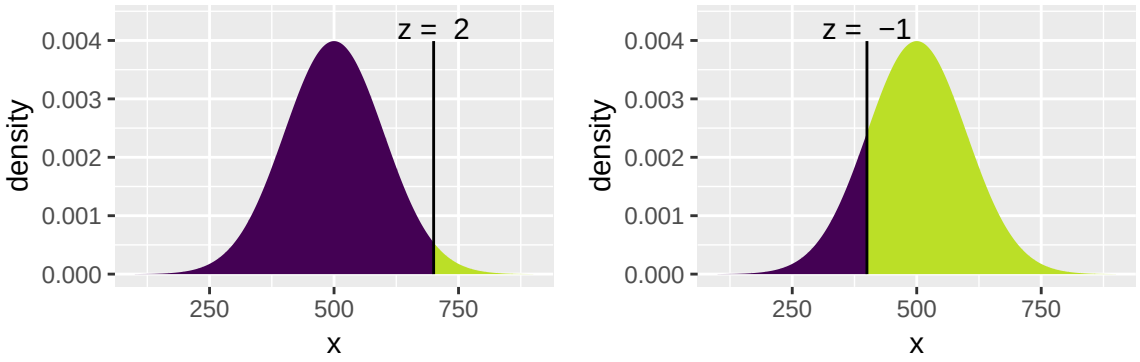
4. 16% are below 400 and 2.5% are above 700, so the remaining 81.5% must be between 400 and 700.

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

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

## [1] 0.6826895

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

## [1] 0.9544997

pnorm( 300, 500, 100)

## [1] 0.02275013

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

## [1] 0.8185946
```

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] 0.8185946
```



Example 3.3.5. We can use `qnorm()` to compute percentiles. For example, let's calculate the 75th percentile for SAT distributions.

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

```
## [1] 567.449
```

3.3.6 Beta Distributions

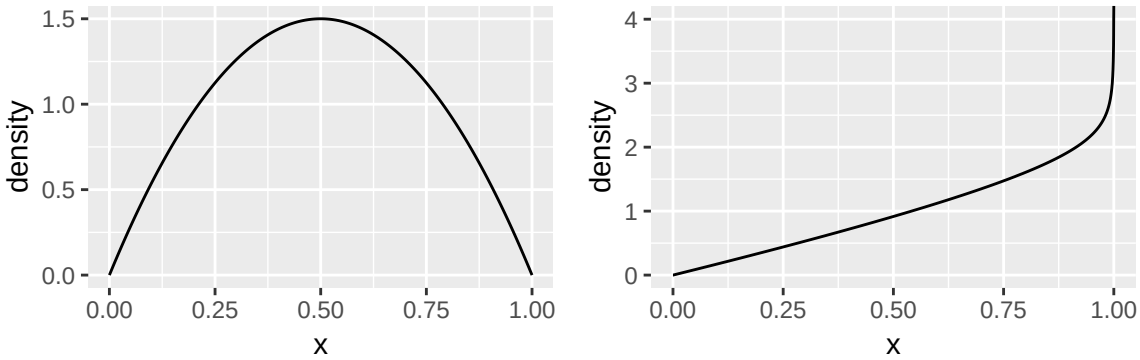
The Beta distributions have support on the interval $(0, 1)$, so they can provide a model for proportions or other quantities that are bounded between 0 and 1.[4] The Beta distributions have two parameters, imaginatively called `shape1` and `shape2`. The kernel of the Beta distributions is a product of a power of x and a power of $(1 - x)$:

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

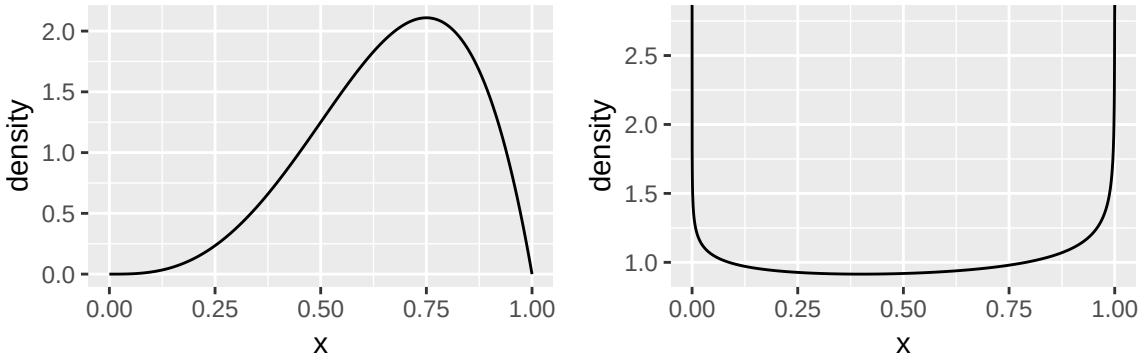
When $\alpha = \beta$, the distribution is symmetric, and when $\alpha = \beta = 1$, we have the $\mathsf{Unif}(0, 1)$-distribution.

The two shape parameters provide a wide variety of shapes.

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



[4] 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.

3.4 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 3.4.1. 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.

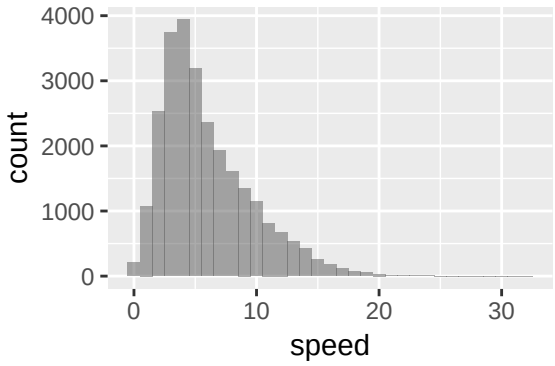
```
read.csv('http://www.calvin.edu/~rpruim/data/stob/TwinfallsWind.csv') -> Wind
head(Wind, 2)
```

```
##       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=subset(Wind, 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.

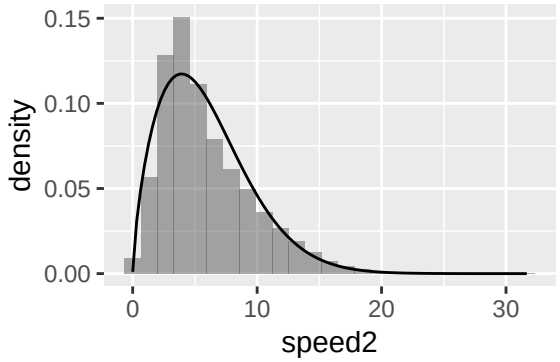
```
Wind <- Wind %>%
  mutate(speed2 = ifelse( speed > 0, speed, 0.0025) )
fitdistr( Wind$speed2, "weibull" )
```

```
##       shape         scale
##   1.694422851   6.650586935
##  (0.007957624) (0.025551827)
```

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

We can combine the `gf_dhistogram()` function with `gf_fitdistr()` so we can see how good the fit is graphically.

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

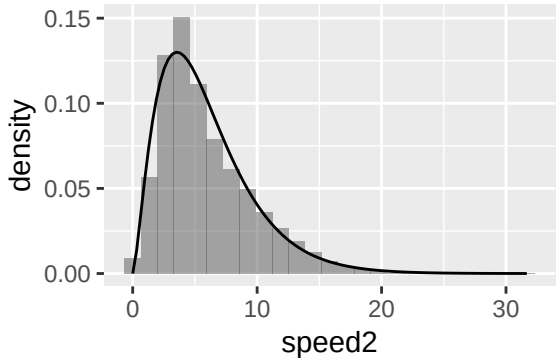


Example 3.4.2. As an alternative, we could fit a Gamma distribution to the wind speed data.

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

```
##       shape         rate
##   2.495582854   0.421178362
##  (0.020485581) (0.003828652)
```

```
gf_dhistogram( ~ speed2, data=Wind, fit="gamma" ) %>%
  gf_fitdistr(dist='gamma')
```



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." [**?**]

Example 3.4.3. 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.

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

```
##       mean          sd
##   37.0853659    9.8639541
##  ( 1.0892915) ( 0.7702454)
```

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

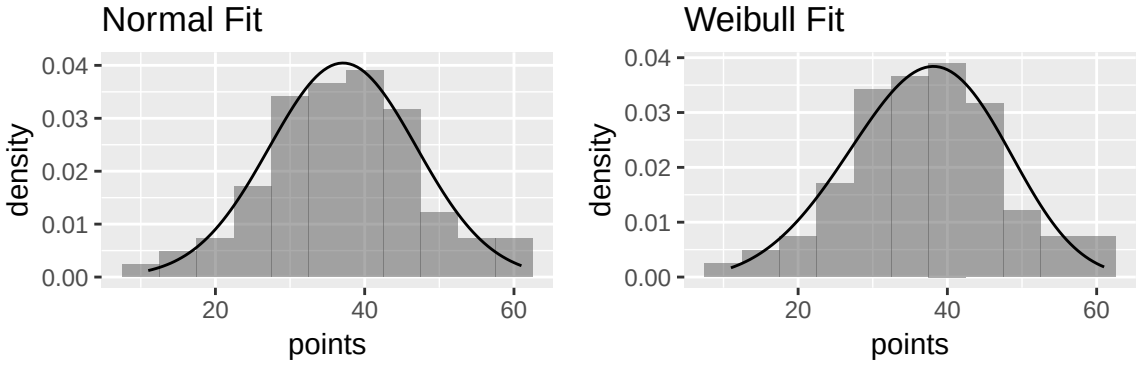
```
##      shape        scale
##    4.1227692   40.7746012
##  ( 0.3454908) ( 1.1516943)
```

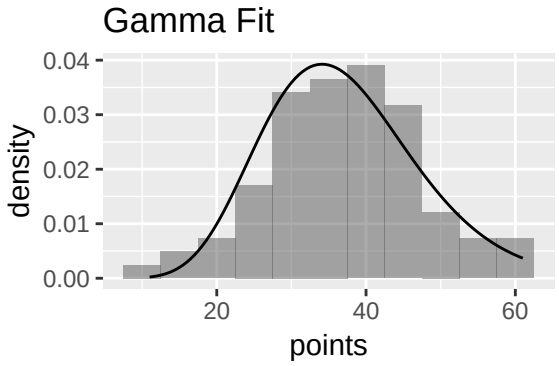
```
fitdistr(Jordan8687$points, "gamma")
```

```
##      shape         rate
##   12.4284300    0.3351303
##  ( 1.9153529) ( 0.0527028)
```

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, title="Normal Fit") %>%
  gf_fitdistr(dist='norm')
gf_dhistogram(~ points, data=Jordan8687, binwidth=5, title="Weibull Fit") %>%
  gf_fitdistr(dist='weibull')
gf_dhistogram(~ points, data=Jordan8687, binwidth=5, title="Gamma Fit") %>%
  gf_fitdistr(dist='gamma')
```



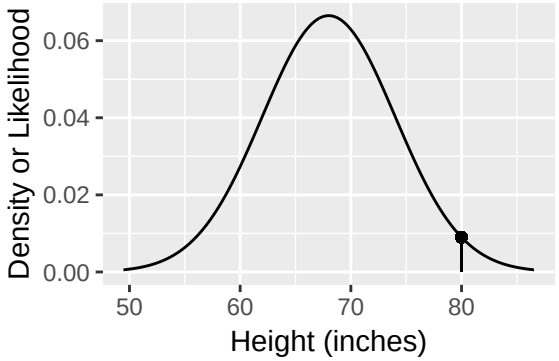


The three fits are similar, but not identical.

3.4.1 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 3.4.4. 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

$$\left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^7 = 0.0599323 \, .$$

  (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

$$\left(\frac{1}{3}\right)^3 \left(\frac{2}{3}\right)^7 = 0.0021677 \, .$$

- If P(roll 1 or 2) $= \frac{1}{2}$, then the probability of obtaining exactly Michael's data is

$$\left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^7 = 0.0016777 \, .$$

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.

3.4.2 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.[5]

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

[5] If our distribution has more than 2 parameters, we will need higher moments, which we will not cover here.

formulas for the means and variances in terms of the parameters involved. These are listed in Table 3.1

Example 3.4.5. Let's return to the wind speeds in Example 3.4.1. 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 3.1, 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] 0.8862269
```

```
sqrt(pi)/2
```

```
## [1] 0.8862269
```

From Table 3.1 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

$$\hat{\beta} \frac{\sqrt{\pi}}{2} = \overline{x}; .$$

That is,

$$\hat{\beta} = \frac{2\overline{x}}{\sqrt{\pi}}$$

```
x.bar <- mean(~speed, data=Wind)
x.bar
```

```
## [1] 5.925238
```

```
beta.hat <- x.bar * 2 / sqrt(pi)
beta.hat
```

```
## [1] 6.685915
```

So our method of moments fit for the data is a $\mathsf{Rayleigh}(6.69) = \mathsf{Weibull}(2, 6.69)$

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 3.4.6. 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] 5.925238
```

```
v <- var(~speed, data=Wind); v
```

```
## [1] 13.34635
```

```
sqrt(v)
```

```
## [1] 3.653265
```

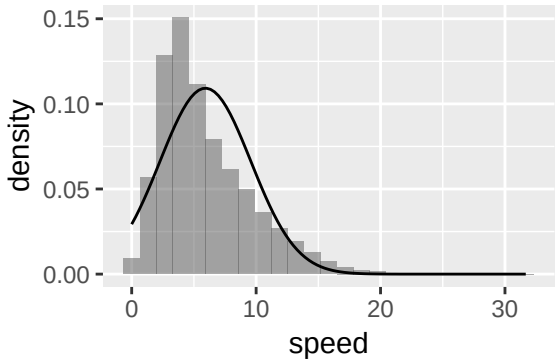
So the method of moments suggests a Norm$(5.93, 3.65)$ distribution. In this case, the method of moments and maximum likelihood methods give the same results.

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

```
##       mean          sd
##   5.92523770   3.65319577
##  (0.02253814) (0.01593687)
```

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.

```
gf_dhistogram(~speed, data=Wind) %>%
  gf_fitdistr(dist="norm")
```



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.

3.5 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()`.

3.5.1 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 3.5.1. Let's begin by forming a randomly selected sample of 10 basketball games.

```
set.seed(123)              # so you can get the same sample if you like.
SmallJordan <- sample(Jordan8687, 10)
SmallJordan
```

```
##    game points orig.id
## 24   24     27      24
## 64   64     44      64
## 33   33     31      33
## 70   70     40      70
## 74   74     26      74
## 4     4     33       4
## 41   41     27      41
## 67   67     40      67
```

distribution pdf	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: $\mathsf{Beta}(\alpha,\beta)$ $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

Table 3.1: 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.

```
## 76   76     31      76
## 34   34     43      34
```

```
probs <- seq(0.05, 0.95, by=0.10)
probs
```

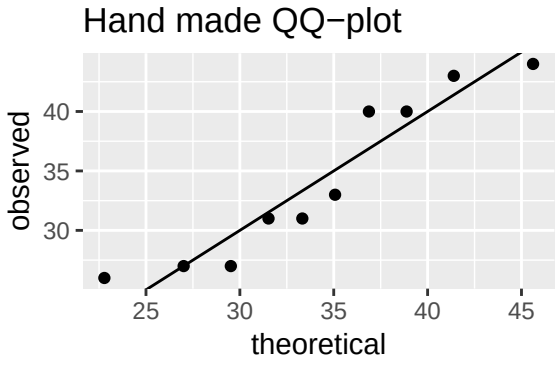
```
##  [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
```

```
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        26    22.78304
## 2        27    27.00609
## 3        27    29.51835
## 4        31    31.52548
## 5        31    33.32778
## 6        33    35.07222
## 7        40    36.87452
## 8        40    38.88165
## 9        43    41.39391
## 10       44    45.61696
```

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.

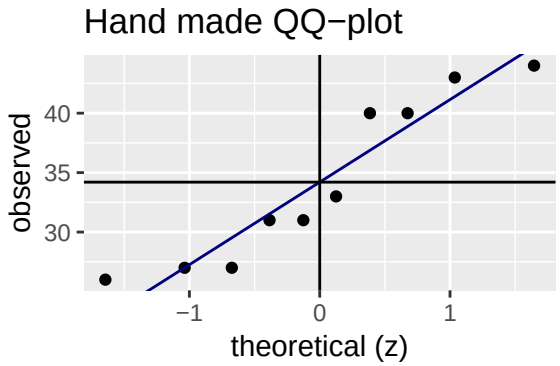
```
gf_point( observed ~ theoretical, data=QQData, title="Hand made QQ-plot" ) %>%
  gf_abline(intercept = 0, slope=1)
```



Even better, we don't need to know the mean and standard deviation in advance, because all normal distributions are linear transformations of the $\mathsf{Norm}(0,1)$-distribution. So our standard practice will be to compare our data to the $\mathsf{Norm}(0,1)$-distribution. If $X \sim \mathsf{Norm}(\mu,\sigma)$, then $X = \mu + \sigma Z$ where $Z \sim \mathsf{Norm}(0,1)$, so a plot of X vs. Z will have slope σ and intercept μ.

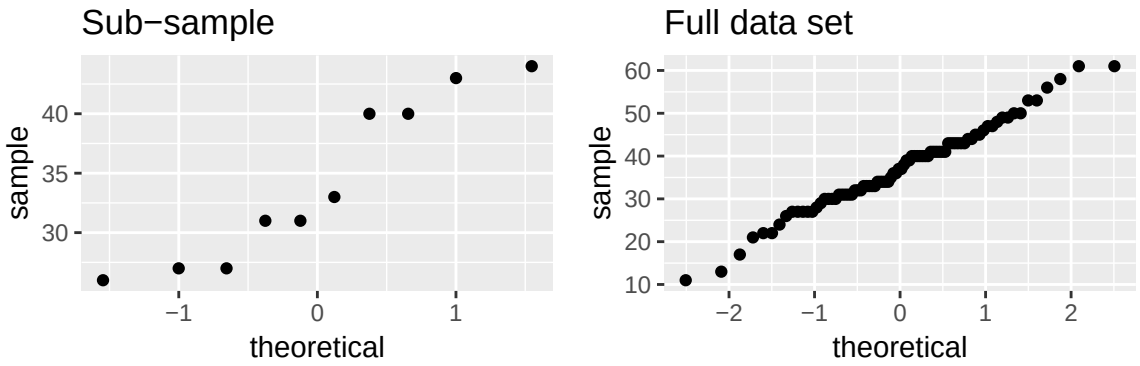
```
theoretical2 <- qnorm( probs, mean=0, sd=1 ) # theoretical quantiles from Norm(0,1)
QQData2 <- data.frame(observed=observed, theoretical=theoretical2)
```

```
gf_point( observed ~ theoretical, data=QQData2, title="Hand made QQ-plot", xlab="theoretical (z)" ) %>%
  gf_abline(slope = sd(SmallJordan$points), intercept = mean(SmallJordan$points), color='navy') %>%
  gf_hline(yintercept = mean(SmallJordan$points)) %>%
  gf_vline(xintercept = 0)
```



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" )
```



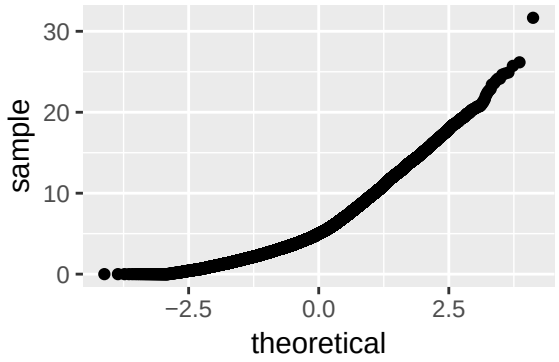
3.5.2 Other distributions

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).[6]

Example 3.5.2. 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.

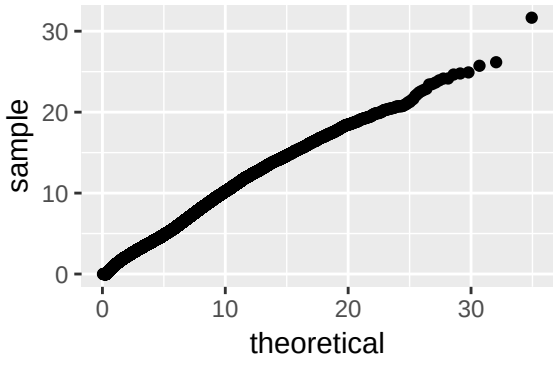
```
gf_qq( ~ speed2, data=Wind )  # normal-quantile plot; normal is not a good model
```

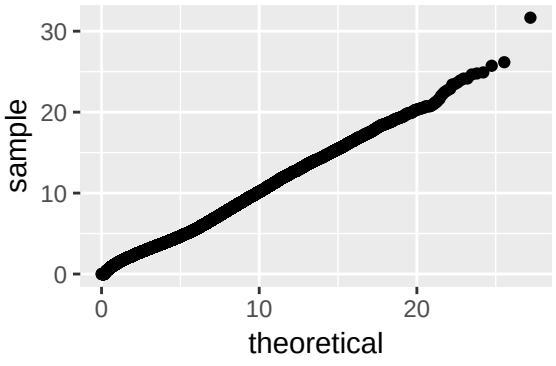
[6]There are a few other families of distributions that have a prototypical member such that all other members are a linear transformation of the prototype. The exponential family is one such family.



The normal model does not fit well, but both Gamma and Weibull are reasonable models:

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


##       shape         rate
##   2.495582854   0.421178362
##  (0.020485581) (0.003828652)


fitdistr(Wind$speed2, "Weibull")


##       shape         scale
##   1.694422851   6.650586935
##  (0.007957624) (0.025551827)


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 )
```





Exercises

3.1 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 $\mathrm{P}(X = \frac{1}{2})$.

3.2 Repeat parts (2) – (4) of Example 3.3.1 using geometry rather than R.

3.3 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

3.4 Let $Y \sim \mathsf{Triangle}(0, 10, 4)$. Compute $\mathrm{E}(Y)$ and the median of Y.

3.5 Let $W \sim \mathsf{Unif}(0, 10)$. Compute $\mathrm{E}(W)$ and $\mathrm{Var}(W)$.

3.6

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.

3.7 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` $= 3$)

3.8 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`)

3.9

a) Using Table 3.1 and the method of moments, fit an exponential distribution to the Twin Falls wind speed data. 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?

3.10 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 3.1 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()`?

3.11 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).

3.12 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

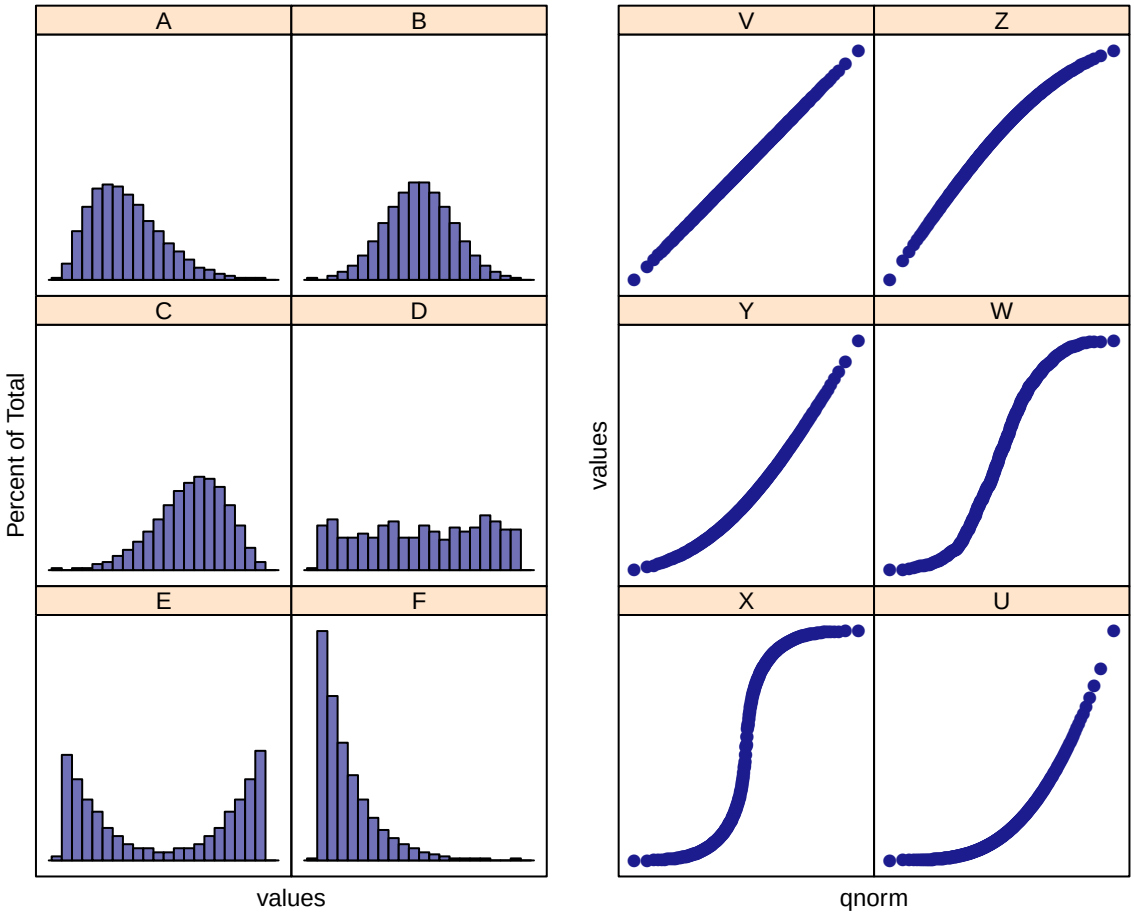
```
Baby <- read.file("http://www.calvin.edu/~rpruim/data/BabyCryIQ.csv")

## Reading data with read.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.

3.13 Create normal quantile plots for the ages of patients in the `HELPrct` data set separated by `substance`. (Getting separate or overlaid plots using `qqmath()` works just like it does for other lattice plots).

Comment on the plots.

3.14 Match the normal-quantile plots to the histograms.



3.15 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.

3.16 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?

3.17 Use the infromation 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?

4 Random Variables and Probability

4.1 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.

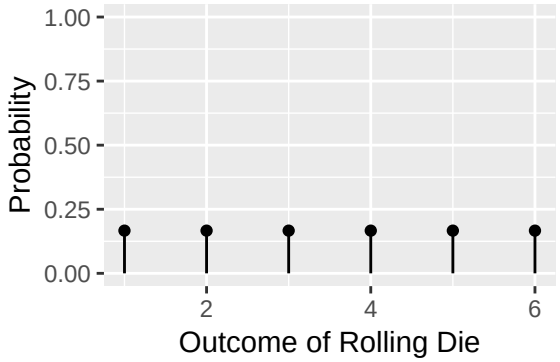
Examples of random variables:

- 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.
- Sample 100 people and measure how long their right toenail is in mm. (Is this a good example or not? It's a topic for discussion...)

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 $\mathrm{P}(X = 5)$?

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

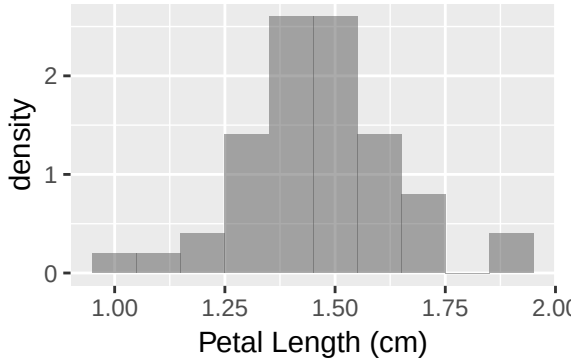
As an example of a probability distribution, we can first consider a *discrete* random variable. Most of the examples of random variables given above are discrete. In other words, the values they can take on come from a set containing a finite number of possible values. For example, if you roll a 6-sided 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. 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, which shows a smoothed version of the silhouette of a histogram (take STAT 343 and/or read about Kernel Density Estimation if you want a better explanation), Density is scaled such that the integral over all possible x-values (the area under the curve) is 1. For a histogram, Density is relative frequency, 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).



4.2 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 4.2.1. 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 T H T T T T T T
##
## Number of Heads: 1 [Proportion Heads: 0.1]
```

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     1     9  0.1
## 2 10     5     5  0.5
## 3 10     5     5  0.5
```

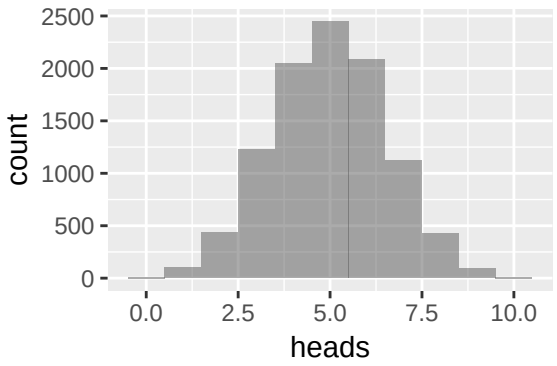
```
do(10000) * rflip(10) -> tosses
tally(~ heads, data=tosses, format="prop")
```

```
## heads
```

[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.

```
##      0      1      2      3      4      5      6      7      8      9     10
## 0.0010 0.0101 0.0438 0.1225 0.2049 0.2444 0.2086 0.1124 0.0423 0.0090 0.0010
```

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



Based on this sample, we would estimate that $\mathrm{P}(X = 5) \approx 0.2444$.

Example 4.2.2. 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] 5 4 3 4 2 3 4 4 2 5
```

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] 1 6 5 6 1 1
```

```
head(die2)
```

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

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
##  1628  8372


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


## prop_TRUE
##    0.1628
```

So the probability appears to be approximately 0.1628.

Example 4.2.3. 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
##     0.141
```

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
##  280  584  840 1069 1296 1735 1410 1121  822  570  273


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


## s
##     2     3     4     5     6     7     8     9    10    11    12
##  2.80  5.84  8.40 10.69 12.96 17.35 14.10 11.21  8.22  5.70  2.73
```

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:

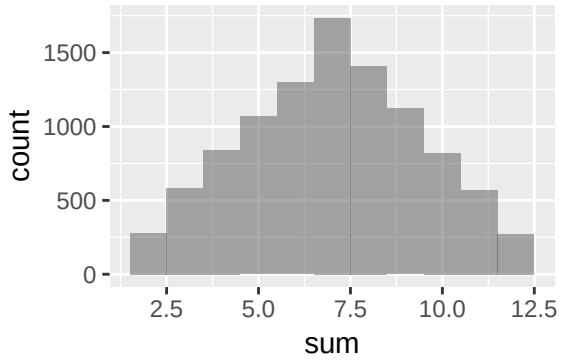
```
rolls <- data.frame( first = die1, second = die2, sum = die1 + die2 )
head(rolls)


##   first second sum
```

```
## 1     1      3   4
## 2     6      3   9
## 3     5      5  10
## 4     6      1   7
## 5     1      1   2
## 6     1      6   7
```

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

```
## sum
##      2      3      4      5      6      7      8      9     10     11     12
## 0.0280 0.0584 0.0840 0.1069 0.1296 0.1735 0.1410 0.1121 0.0822 0.0570 0.0273
```

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



4.3 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.).

4.3.1 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

P(A or B) is the probability that either A or B (or both) occurs. Often this is written 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.

P(A and B) is the probability that *both* A and B occur. This is also written 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 $A \cap B = \emptyset$.

4.3.2 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 4.3.1.

1. Coin Toss: P(heads) $= \frac{1}{2}$ if heads and tails are equally likely.
2. Rolling a Die: P(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

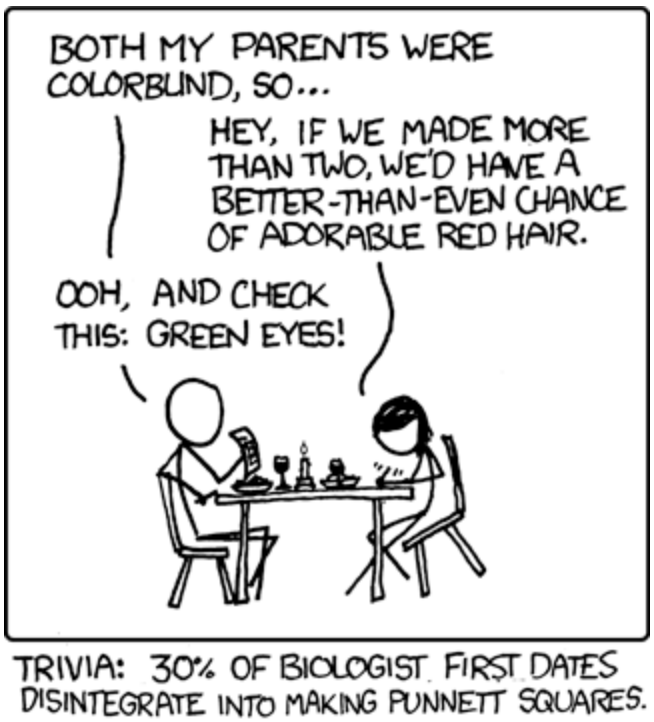
Let X be the sum of two dice.

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

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/`

4.4 Conditional Probability

Example 4.4.1. 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 4.1).

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

Figure 4.1: Illustrating the sample space for Example 4.4.1.

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 4.2 and Definition 4.4.

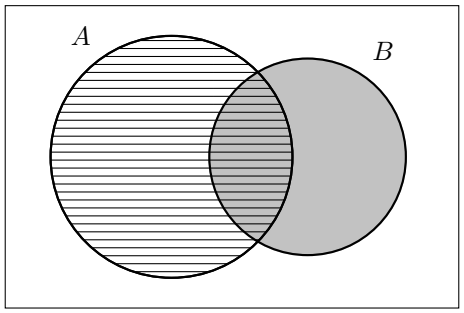


Figure 4.2: A Venn diagram illustrating the definition of conditional probability. $\mathrm{P}(A \mid B)$ is the ratio of the area of the football shaped region that is both shaded and striped ($A \cap B$) to the area of the shaded circle (B).

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 4.4.2. 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:

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

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 4.4.3. 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 4.4.4. 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

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

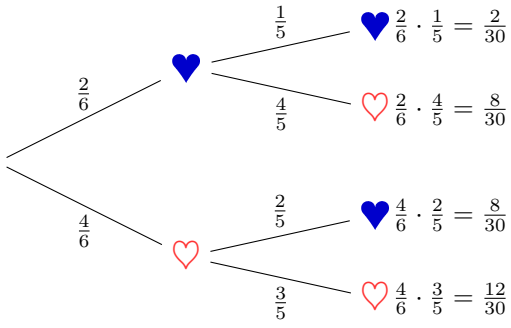
Example 4.4.5. Q. In a bowl are 4 red Valentine hearts and 2 blue Valentine hearts.

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 4.4.6. 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?

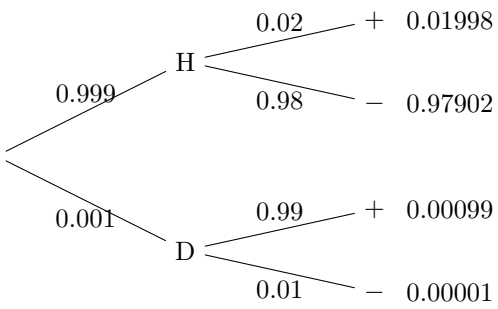
A. We begin by introducing some notation. Let D be the event that a person has the disease. Let H be the event that the person is healthy. Let $+$ be the event that the test comes back positive (meaning it indicates disease – probably a negative from the perspective of the person tested). Let $-$ be 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.
- $$\begin{aligned} \mathrm{P}(D \mid +) &= \frac{\mathrm{P}(D \cap +)}{\mathrm{P}(+)} \\ &= \frac{\mathrm{P}(D) \cdot \mathrm{P}(+ \mid D)}{\mathrm{P}(D \cap +) + \mathrm{P}(H \cap +)} \\ &= \frac{0.001 \cdot 0.99}{0.001 \cdot 0.99 + 0.999 \cdot 0.02} = 0.0472. \end{aligned}$$

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.

4.4.1 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 4.4.7. 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 4.4.8. 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:

$$\mathrm{P}(5 \text{ heads in } 5 \text{ flips}) = (0.5)^5 = 0.03125$$

Example 4.4.9. 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

$$\mathrm{P}(G_1 \text{ and } G_2 \text{ and } \cdots \text{ and } G_{10}) = \underbrace{.99 \cdot .99 \cdot .99 \cdots .99 = .99^{10}}_{10 \text{ of these}} = 0.9043821 \, .$$

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%.

Exercises

4.1 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.)

4.2

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.

4.3 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] 19 26 36 30 21
```

```
sample(1:52, 5)
```

```
## [1] 52 10 21 28 12
```

```
deal(1:52, 5)
```

```
## [1] 19  8 52 51 18
```

```
deal(1:52, 5)
```

```
## [1] 30 23 20 51 15
```

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] 1
```

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

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

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

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

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   1
## 3   1
```

4.4 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?

4.5 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 $X < 5$)?

4.6 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.

4.7 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?

4.8 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?

4.9 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.

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

5 Linear Models

In this chapter we will explore how to use data to investigate the relationship among two (or more) variables when this relationship is not known in advance. We will start with the simplest case: determining whether two quantitative variables are linearly associated with each other. 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.

5.1 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 (and the *uncertainty* of our estimates!).
- Estimate the value Y associated with a particular value of x (*with uncertainty!*).
- Say something about how well a line fits the data.

5.2 Fitting the Simple Linear Model

5.2.1 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

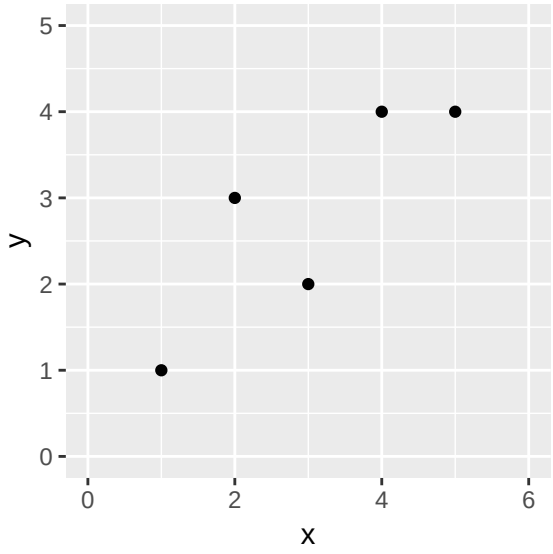
$$\text{residual} = \text{observed} - \text{predicted}$$

Example 5.2.1. Consider the following small data set.

```
someData <- data.frame(
  x=1:5,
  y=c(1,3,2,4,4)
)
someData
```

```
##   x y
## 1 1 1
## 2 2 3
## 3 3 2
## 4 4 4
## 5 5 4
```

```
gf_point( y ~ x, data=someData) %>%
  gf_lims(y=c(0,5), x=c(0,6))
```

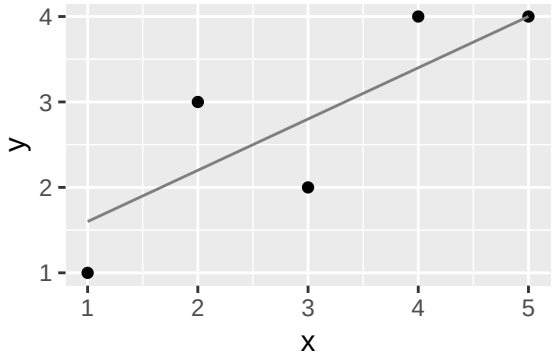


1. Add a line to the plot that "fits the data well". Don't do any calculations, just add the line.
2. Now estimate the residuals for each point relative to your line
3. Compute the sum of the squared residuals, SSE.
4. Estimate the slope and intercept of your line.

[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 $(0, 1)$ and $(5, 4)$. the equation for this line is $y = 1 + .6x$, and it looks like a pretty good fit:

```
f <- makeFun( 1 + .6 * x ~ x)
gf_point( y ~ x, data=someData, xlim=c(0,6), ylim=c(0,5) ) %>%
gf_fun( f(x) ~ x, color="gray50")
```



The residuals for this function are

```
someData <- someData %>%
  mutate(resids = y - f(x))
someData %>% select(resids)


##   resids
## 1   -0.6
## 2    0.8
## 3   -0.8
## 4    0.6
## 5    0.0
```

and SSE is

```
someData %>%
  summarize(SEE = sum(resids^2))


##   SEE
## 1   2
```

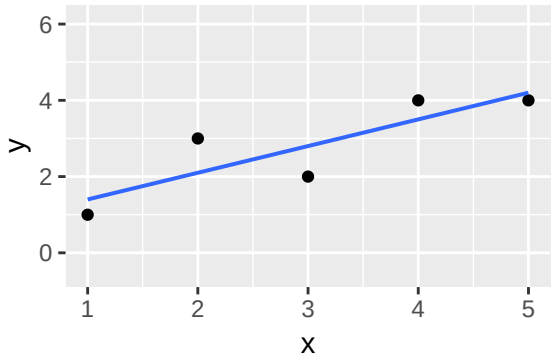
If your line is a good fit, then SSE will be small. The best fitting line will have the smallest possible SSE. The `lm()` function will find this best fitting line for us.

```
model1 <- lm( y ~ x, data=someData ); model1


##
## Call:
## lm(formula = y ~ x, data = someData)
##
## Coefficients:
## (Intercept)            x
##         0.7          0.7
```

This says that the equation of the best fit line is

$$\hat{y} = 0.7 + 0.7x$$

```
gf_point( y ~ x, data=someData ) %>%
  gf_lm()
```



We can compute SSE using the `resid()` function.

```
SSE <- sum ( resid(model1)^2 ); SSE


## [1] 1.9
```

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.

5.2.2 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 (or linear algebra...neither of which we will detail) 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.

5.2.3 An Example: Estimating OSA

Example 5.2.2. 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.

```
require(Devore6)
head(xmp12.01, 3)
```

```
##   palprebal  OSA
## 1      0.40 1.02
## 2      0.42 1.21
## 3      0.48 0.88
```

```
# note misspelling in this data set; let's fix it
names(xmp12.01)[1] <- "palpebral"
names(xmp12.01)
```

```
## [1] "palpebral" "OSA"
```

```
x.bar <- mean( ~palpebral, data=xmp12.01)
y.bar <- mean( ~OSA, data=xmp12.01)
s_x <- sd( ~palpebral, data=xmp12.01)
s_y <- sd( ~OSA, data=xmp12.01)
r <- cor( xmp12.01$palpebral, xmp12.01$OSA)
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
## 1.0513333 2.8403333 0.3798160 1.2083374 0.9681245
```

```
slope <- r * s_y/s_x
intercept <- y.bar - slope * x.bar
c(intercept=intercept, slope=slope)
```

```
##  intercept      slope
## -0.3977389  3.0799672
```

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 5.2.3. 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=xmp12.01)
osa.model
```

```
##
## Call:
## lm(formula = OSA ~ palpebral, data = xmp12.01)
```

```
##
## Coefficients:
## (Intercept)    palpebral
##     -0.3977       3.0800
```

`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
##  -0.3977389   3.0799672
```

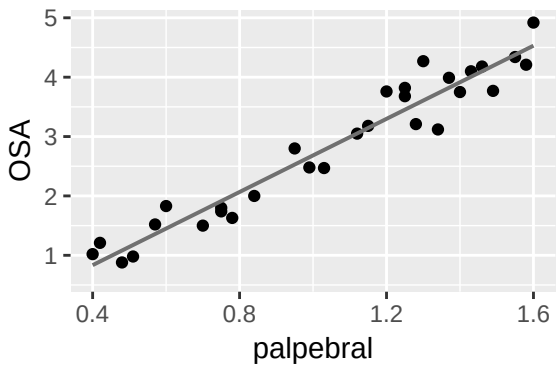
This means that the equation of the least squares regression line is

$$\hat{y} = -0.398 + 3.08x$$

As we have before with the "hat" symbol, 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=xmp12.01 ) %>%
  gf_lm(color='grey44')
```



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.

5.2.4 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).

When posing a regression model, think about which variable to choose as the response and which to choose as the predictor. Some things to consider:

- If you expect that two variables should be causally associated, it generally makes sense to make the *cause* the *predictor* and the *effect* the *response*.

- If you are especially interested in predicting or knowing the value of one variable, but it is associated with another easier-to-measure variable, then you can predict the variable of interest (make it the response) on the basis of the easy-to-measure or known quantity (the predictor).

5.3 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 5.3.1. 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
## 3.298222
```

We can also use the built-in R function `predict()` or `fitted()` if we want to make predictions for the original dataset (`predict()` can also make predictions for a new dataset that contains new values of the predictor variable – see example below).

```
predict(osa.model)
```

```
##         1         2         3         4         5         6         7         8         9
## 0.8342480 0.8958474 1.0806454 1.1730444 1.3578424 1.4502415 1.7582382 1.9122365 1.9122365
##        10        11        12        13        14        15        16        17        18
## 2.0046356 2.1894336 2.5282300 2.6514287 2.7746274 3.0518244 3.1442234 3.2982218 3.4522202
##        19        20        21        22        23        24        25        26        27
## 3.4522202 3.5446192 3.6062185 3.7294172 3.8218162 3.9142152 4.0066143 4.0990133 4.1914123
##        28        29        30
## 4.3762103 4.4686093 4.5302087
```

```
new.data <- data.frame(palpebral = c(0.5, 0.77))
predict(osa.model, newdata = new.data)
```

```
##        1        2
## 1.142245 1.973836
```

It might be neater to store these predictions in the dataset with the predictor variable, to keep them together and make plotting easier later on:

```
new.data <- new.data %>%
  mutate(preds = predict(osa.model, newdata = new.data))
new.data
```

```
##   palpebral    preds
## 1      0.50 1.142245
## 2      0.77 1.973836
```

As it turns out, the 17th measurement in our data set had a `palpebral` measurement of 1.2 cm.

```
xmp12.01[17,]


##    palpebral  OSA
## 17       1.2 3.76
```

The corresponding OSA of 3.76 means that the residual for this observation is

$$\text{observed} - \text{predicted} = 3.76 - 3.2982218 = 0.4617782$$

5.3.1 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.

5.4 Parameter Estimates

5.4.1 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.

5.4.2 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 = xmp12.01)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60942 -0.19875 -0.01902  0.21727  0.66378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3977     0.1680  -2.367   0.0251
## palpebral     3.0800     0.1506  20.453   <2e-16
##
## Residual standard error: 0.308 on 28 degrees of freedom
## Multiple R-squared:  0.9373,Adjusted R-squared:  0.935
## F-statistic: 418.3 on 1 and 28 DF,  p-value: < 2.2e-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 a Chi-Squared distribution with $n - 2$ degrees of freedom.

5.5 Checking Assumptions

5.5.1 What have we assumed?

In fitting a linear regression model, we have assumed:

- A linear relationship between the explanatory and response variables
- 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
- The errors (ϵ) are Normally distributed

5.5.2 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. (Meaning: There is a clear linear relationship, or there is no obvious trend; in either case it's fine to try a linear model. If there is a *clear nonlinear* relationship present in the data, a linear model is a *bad* choice.) If there is a nonlinear relationship apparent, 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.

5.5.3 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-1th data point). Lag 2 is between the ith and i-2th 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:

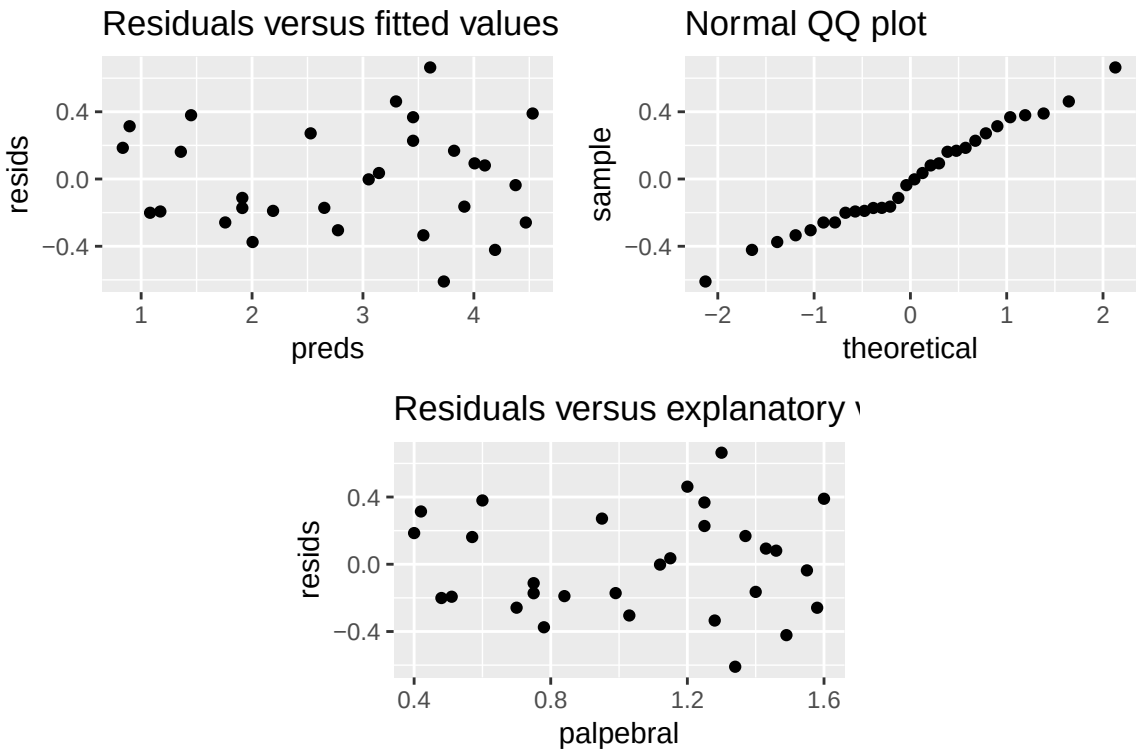
- Residuals as a function of "fitted values", or model predictions for the x-values obseved in the actual data set.
- (For our simple linear model with one predictor) Residuals as a function of the observed values of the explanatory variable (from the actual data set).
- Normal quantile-quantile plot of the residuals (note: in studying PDFs earlier, 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 or space (if you know the order in which they were collected, or have other reason to suspect some kind of non-independence).

- Residual autocorrelation function plot (if you know the time-space order in which data were collected, or have other reason to suspect some kind of non-independence).

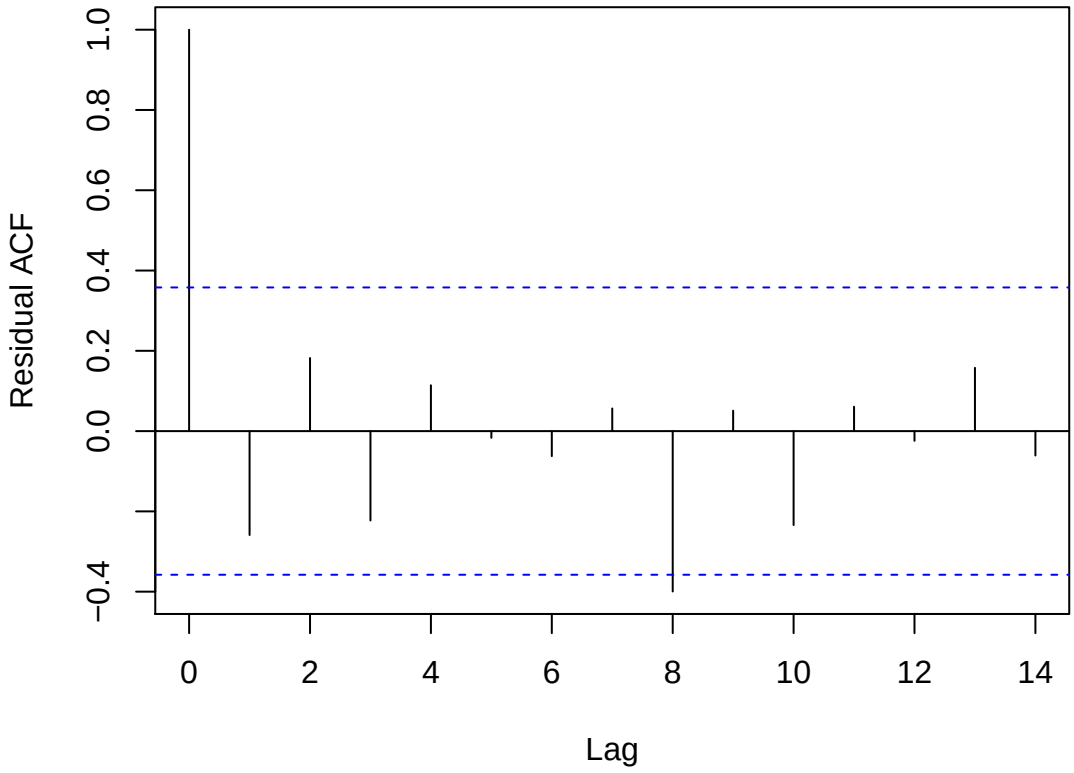
For all the scatter plots, we want to make sure the residuals "look random" – the extent of the spread of the residuals should not vary with time or x or y ("trumpet"-shaped plot), except if the variation is just because there are fewer observations present in a specific zone of the plot. If there is 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 5.5.1. Returning to our OSA data, we can obtain the residuals using the `resid()` function and plot them.

```
xmp12.01 <- xmp12.01 %>%
  mutate(preds = predict(osa.model),
         resids = resid(osa.model))
gf_point( resids ~ preds, data=xmp12.01,
        title="Residuals versus fitted values")
gf_qq( ~ resids,  data=xmp12.01 , title='Normal QQ plot')
gf_point( resids ~ palpebral, data=xmp12.01,
        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.

5.5.4 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).

5.6 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.

5.6.1 Estimating the βs

Example 5.6.1. 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 = xmp12.01)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60942 -0.19875 -0.01902  0.21727  0.66378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3977     0.1680  -2.367   0.0251
## palpebral     3.0800     0.1506  20.453   <2e-16
##
## Residual standard error: 0.308 on 28 degrees of freedom
## Multiple R-squared:  0.9373,Adjusted R-squared:  0.935
## F-statistic: 418.3 on 1 and 28 DF,  p-value: < 2.2e-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) -0.3977389  0.1680090 -2.367367 2.506081e-02
## palpebral    3.0799672  0.1505882 20.452918 2.252825e-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] 2.048407
```

```
t.star * 0.151
```

```
## [1] 0.3093095
```

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) -0.7418897 -0.05358811
## palpebral    2.7715014  3.38843310
```

A 68% confidence interval should have a margin of error of approximately 1 standard uncertainty:

```
confint(osa.model, level=0.68)
```

```
##                    16 %       84 %
## (Intercept) -0.5678369 -0.2276409
## palpebral    2.9275067  3.2324278
```

```
apply(confint(osa.model, level=0.68) , 1, diff) / 2  # half width of CIs
```

```
## (Intercept)   palpebral
##   0.1700980   0.1524606
```

Compare this with the standard error from the model summary - does it check out?

```
summary(osa.model)
```

```
##
## Call:
## lm(formula = OSA ~ palpebral, data = xmp12.01)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60942 -0.19875 -0.01902  0.21727  0.66378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3977     0.1680  -2.367   0.0251
## palpebral     3.0800     0.1506  20.453   <2e-16
##
## Residual standard error: 0.308 on 28 degrees of freedom
## Multiple R-squared:  0.9373,Adjusted R-squared:  0.935
## F-statistic: 418.3 on 1 and 28 DF,  p-value: < 2.2e-16
```

5.6.2 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 3.298222 3.174238 3.422206
```

```
estimated.osa( 0.8, interval="confidence")
```

```
##        fit      lwr      upr
## 1 2.066235 1.927384 2.205086
```

Or alternatively:

```
predict(osa.model, interval = 'confidence')
```

```
##          fit       lwr      upr
## 1  0.8342480 0.6026547 1.065841
## 2  0.8958474 0.6695853 1.122109
## 3  1.0806454 0.8701023 1.291188
## 4  1.1730444 0.9701842 1.375905
## 5  1.3578424 1.1699228 1.545762
## 6  1.4502415 1.2695443 1.630939
## 7  1.7582382 1.6000801 1.916396
## 8  1.9122365 1.7642201 2.060253
## 9  1.9122365 1.7642201 2.060253
```

```
## 10 2.0046356 1.8622481 2.147023
## 11 2.1894336 2.0570757 2.321791
## 12 2.5282300 2.4088732 2.647587
## 13 2.6514287 2.5346943 2.768163
## 14 2.7746274 2.6592485 2.890006
## 15 3.0518244 2.9347021 3.168947
## 16 3.1442234 3.0250795 3.263367
## 17 3.2982218 3.1742380 3.422206
## 18 3.4522202 3.3217424 3.582698
## 19 3.4522202 3.3217424 3.582698
## 20 3.5446192 3.4095478 3.679691
## 21 3.6062185 3.4678254 3.744612
## 22 3.7294172 3.5838226 3.875012
## 23 3.8218162 3.6703849 3.973248
## 24 3.9142152 3.7566197 4.071811
## 25 4.0066143 3.8425638 4.170665
## 26 4.0990133 3.9282502 4.269776
## 27 4.1914123 4.0137081 4.369116
## 28 4.3762103 4.1840383 4.568382
## 29 4.4686093 4.2689530 4.668266
## 30 4.5302087 4.3254825 4.734935
```

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 3.298222 2.655228 3.941216


estimated.osa( 0.8, interval="prediction")


##        fit      lwr     upr
## 1 2.066235 1.420209 2.71226
```

or, for the full (original) dataset:

```
newdat <- data.frame(palpebral = c(0.8, 1.2))
predict(osa.model, newdata = newdat, interval = 'prediction')


##        fit      lwr      upr
## 1 2.066235 1.420209 2.712260
## 2 3.298222 2.655228 3.941216
```

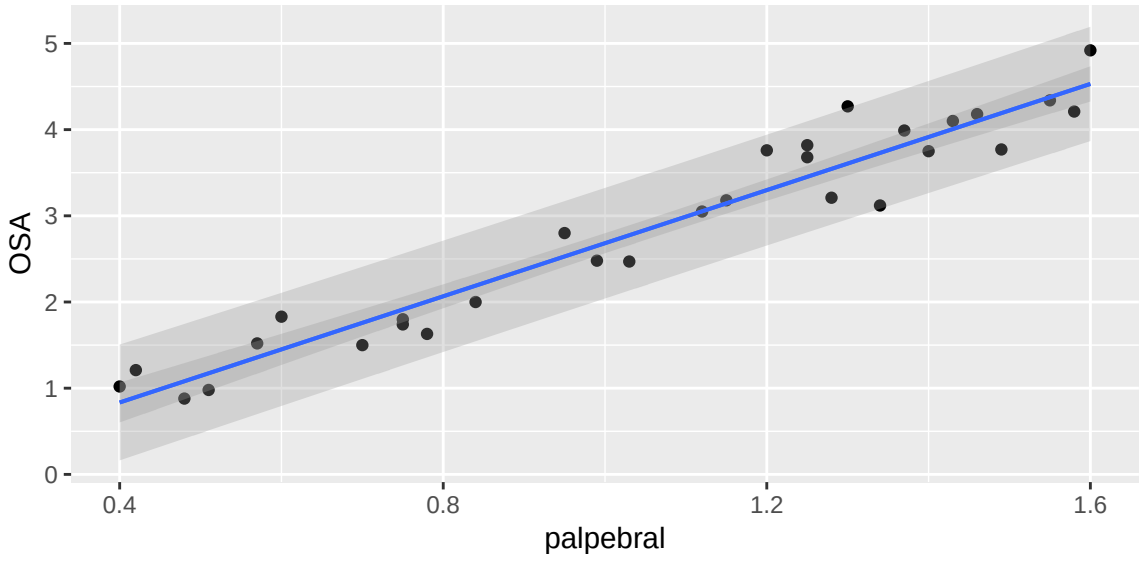
Prediction intervals are typically much wider than confidence intervals. We have to "cast a wider net" to create an interval that contains a new observation (which might be quite a bit above or below the mean) than it is to contain the mean of the distribution.

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.

```
gf_point( OSA ~ palpebral, data=xmp12.01) %>%
  gf_lm(interval = 'confidence') %>%
  gf_lm(interval = 'prediction')
```



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.

Exercises

5.1 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
## -8.279 -4.424  1.205  3.145  8.261
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.12830    2.36778  -0.477    0.642
## rainfall     0.82697    0.03652  22.642  7.9e-12
##
## Residual standard error: 5.24 on 13 degrees of freedom
## Multiple R-squared:  0.9753,Adjusted R-squared:  0.9734
## F-statistic: 512.7 on 1 and 13 DF,  p-value: 7.896e-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 $\hat{\sigma}$?

5.2 Data from a 1993 study to see how well lichens serve as an indicator for air pollution are in the `ex12.20` data set in the `Devore6` package. In that paper, a simple linear model was fit to see how the wet deposition of NO_3^- (gN/m^2) related to the percentage dry weight of lichen.

a) What are the least squares estimates for the intercept and slope of a line that can be used to estimate deposition from the amount of lichen?

b) What is the estimated value of σ?

c) Predict the amount of deposition if the dry weight of the lichen is measured to be 0.7.

d) Give a 95% confidence interval for the mean amount deposition among among samples with a dry weight of lichen measured to be 0.7%.

5.3 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.)

5.4 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 `Devore6` package.

a) Write out an equation for the least squares regression line for predicting separation distance from available space.

b) Given 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.

5.5 Select only the non-diabetic men from the `pheno` data set using

```
men <- subset(pheno, 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
## 3  1012 control 30.47048   M 53.86161 former 5.02   104   94.6  176.2 0.9327354 143  89
## 20 1110 control 26.75386   M 68.07944  never 5.63    99   81.0  174.0 0.9252336 162  91
## 29 1146 control       NA   M 62.14521   <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?

5.6 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 [**?**].

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` to control the size of the plots in RMarkdown.)

c) Comment on the results. Why do you think Anscombe invented these data?

5.7 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.

6 Beyond Linear Regression

6.1 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.

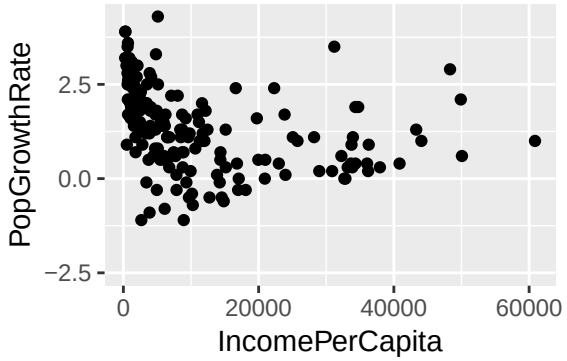
Often, the predictive power of a model, and the R^2 value, can be improved by adding additional explanatory variables – that is, fitting a model with more than one explanatory variable. It could have two predictors, three, or as many as you can (sensibly) come up with. This kind of model is called multiple 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
## -2.68051 -0.70998 -0.02006  0.70727  2.78944
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.657e+00  1.048e-01   15.81  < 2e-16
## IncomePerCapita -2.867e-05  6.218e-06   -4.61 7.72e-06
##
## Residual standard error: 1.041 on 176 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.1077,Adjusted R-squared:  0.1027
## F-statistic: 21.25 on 1 and 176 DF,  p-value: 7.723e-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:

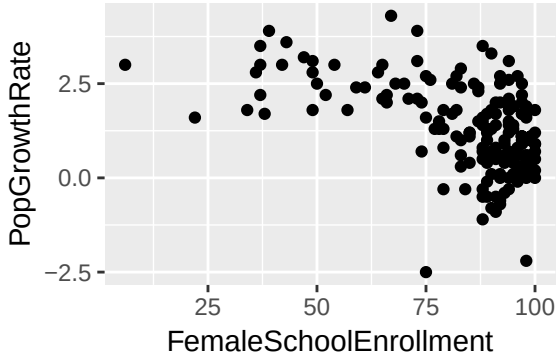
```
gf_point(PopGrowthRate~FemaleSchoolEnrollment, data=whodat)
```

```
## 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
## -2.43374 -0.52979  0.06017  0.56619  2.48052
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.095e+00  3.626e-01  11.293  < 2e-16
## IncomePerCapita         -1.102e-05  6.028e-06  -1.827   0.0694
## FemaleSchoolEnrollment -3.105e-02  4.504e-03  -6.894 1.06e-10
##
## Residual standard error: 0.9086 on 167 degrees of freedom
##   (32 observations deleted due to missingness)
## Multiple R-squared:  0.3101,Adjusted R-squared:  0.3018
## F-statistic: 37.52 on 2 and 167 DF,  p-value: 3.475e-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.

6.2 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())`

6.3 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.

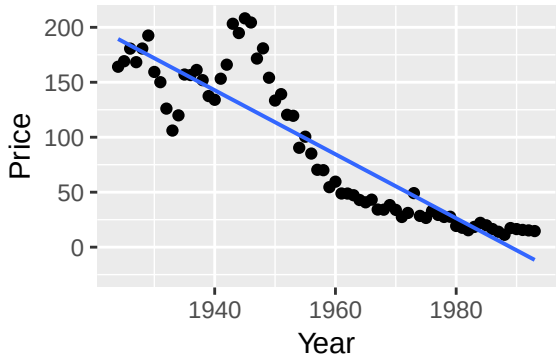
If non-normal residuals are the only apparent problem with a linear regression model, adding additional explanatory variables *might* help in some cases. Most of the time, you would have to turn to a more sophisticated regression model called a generalized linear model (GLM). Fitting GLMs is beyond the scope of this class, and you will not be asked to do it.

6.4 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.

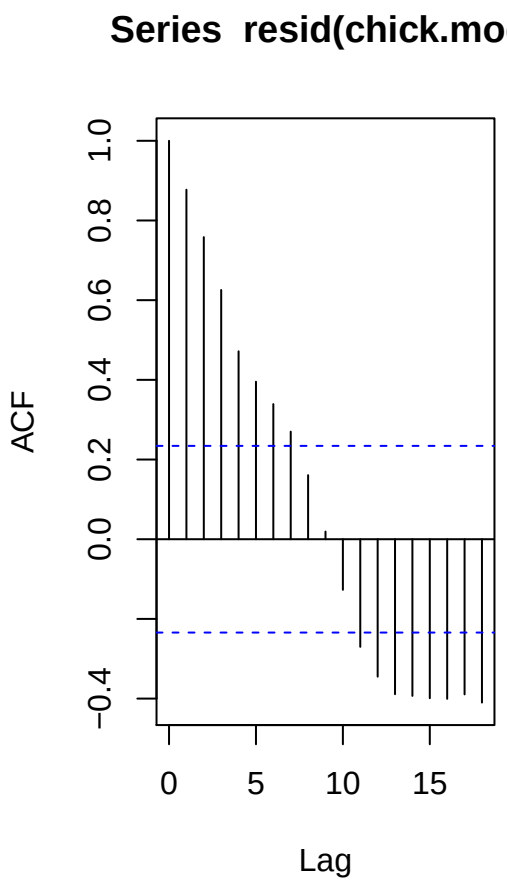
We will consider a simple example using the price of chicken over time (in constant dollars, adjusted for inflation over time). It seems to make sense to try to predict the price of chicken as a function of time (it's been getting progressively cheaper for the last century or so):

```
chickn <- read.csv("http://www.calvin.edu/~sld33/data/chickn.csv", header=TRUE)
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) 5792.6918   329.9948   17.55   <2e-16
## Year          -2.9123     0.1685  -17.29   <2e-16
##
## Residual standard error: 28.48 on 68 degrees of freedom
## Multiple R-squared:  0.8146,Adjusted R-squared:  0.8119
## F-statistic: 298.8 on 1 and 68 DF,  p-value: < 2.2e-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
##   557.8043 566.6823 -274.9022
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.9483234
##
## Coefficients:
##                Value Std.Error   t-value p-value
## (Intercept) 4764.172 1600.4333  2.976802  0.0040
## Year          -2.387    0.8171 -2.921499  0.0047
##
##  Correlation:
##      (Intr)
## Year -1
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.0593757 -0.5673406 -0.1571088  0.3360972  2.0991647
##
## Residual standard error: 41.39414
## 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.

6.5 Heteroscedasticity (Non-constant Error Variance)

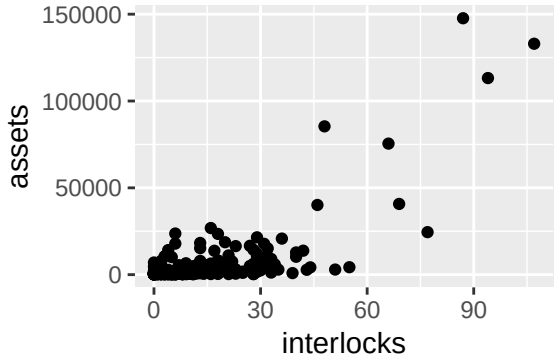
Sometimes, model diagnostics for a linear regression indicate that variance of errors is not constant over the full range of response values. Often, it is the case that the error variance grows larger as the predicted response value grows larger, resulting in a "trumpet-like" shape in the plot of residuals versus fitted values.

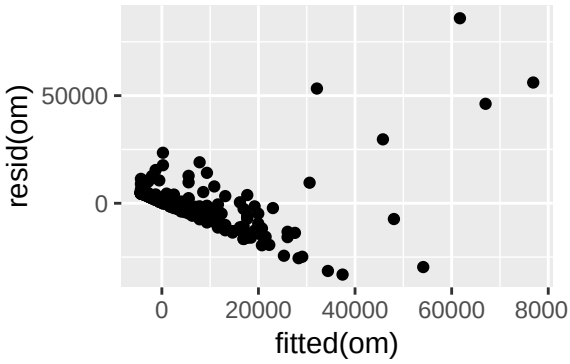
If you spot this problem, first, make sure that the linearity assumption is not violated. Next, if you have the option of including additional predictors in your model, it may be helpful. Next, transforming the response variable may help. Specifically, a log or square-root transformation of the response 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.

6.6 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.

6.7 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. As a simple example, if we have an apparatus that can accelerate objects by applying a constant (but unknown) force F, then since $F = ma$, we would expect the relationship between the mass of the objects tested and the acceleration measured to satisfy

$$a = \frac{F}{m} .$$

This might lead us to fit a model with no intercept (e.g., in R, lm(y 0 + x)) after applying an inverse transformation to the predictor m:

$$a = \beta_1 \cdot \frac{1}{m} .$$

The parameter β_1 corresponds to the (unknown) force being applied and could be estimated by fitting this model.

Alternatively, we could use a logarithmic transformation

$$\log(a) = \beta_0 + \beta_1 \log(m) ,$$
$$a = e^{\beta_0} m^{\beta_1} .$$

In this model, e_0^{β} corresponds to the unknown force and we can test whether $\beta_1 = -1$ is consistent with our data.

Many non-linear relationships can be transformed to linearity. Exercise 6.3 presents several examples – similar to the force example above – 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 6.7.3 provides some guidance for selecting transformations in this situation.

- To obtain better residual behavior.

  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.

6.7.1 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) ~ 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.

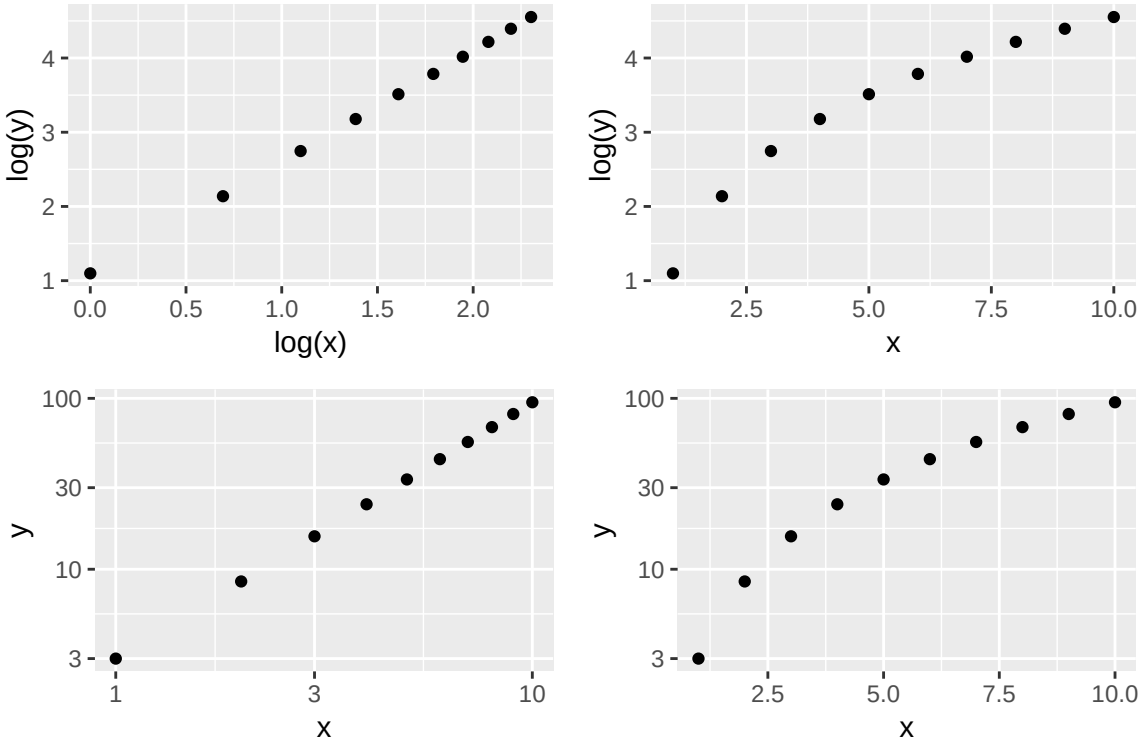
Some common situations that are modeled with exponential laws include population growth and radioactive decay. Note that exponential growth models are typically only good approximations over a limited range since exponential functions eventually grow quickly, and often some external constraints will 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.

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())
```



6.7.2 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 6.7.1. 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

$$\log(C^{-\beta}) = -\beta \log(C) = \log(\beta k t) = \log(\beta k) + \log t$$
$$\log(C) = -\frac{\log(\beta k)}{\beta} - \frac{1}{\beta} \log t$$

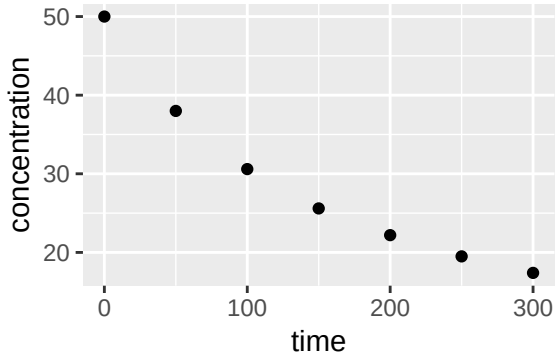
Now if we fit a model of the form

```
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 6.7.2. 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
## -1.175e-04 -4.144e-05  8.281e-05  2.259e-04 -3.128e-05 -3.398e-05 -8.447e-05
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.012e-02  8.762e-05   229.6 2.97e-11
## time        1.248e-04  4.860e-07   256.8 1.70e-11
##
## Residual standard error: 0.0001286 on 5 degrees of freedom
## Multiple R-squared:  0.9999,Adjusted R-squared:  0.9999
## F-statistic: 6.593e+04 on 1 and 5 DF,  p-value: 1.7e-11
```

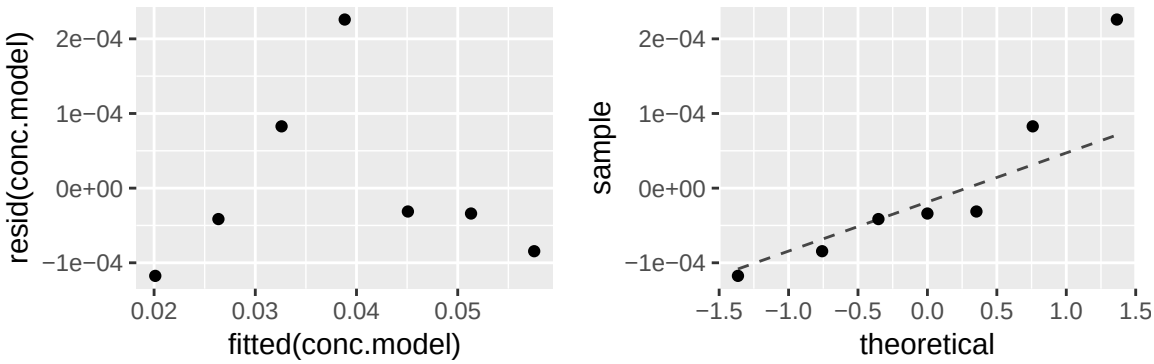
```
confint(conc.model)
```

```
##                    2.5 %       97.5 %
## (Intercept) 0.0198922974 0.0203427516
## time        0.0001235447 0.0001260434
```

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.

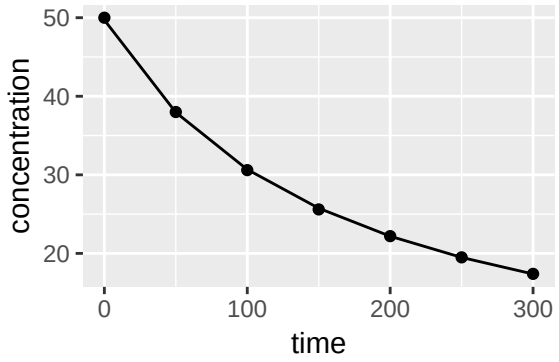
```
gf_point(resid(conc.model) ~ fitted(conc.model))
gf_qq(~resid(conc.model)) %>% gf_qqline()
```



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)
```



6.7.3 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.

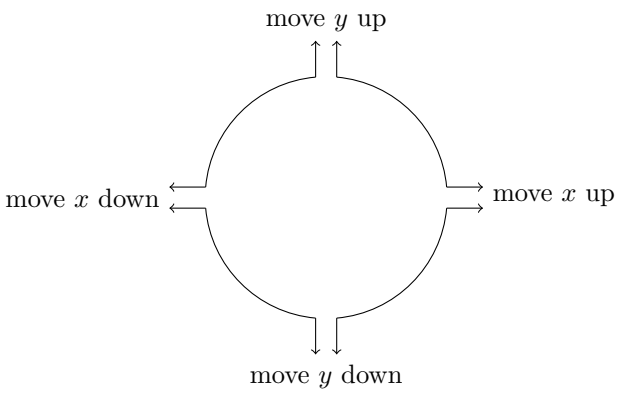
In the 1970s, Mosteller and Tukey introduced what they called the **ladder of re-expression** and **bulge rules** [**?**, **?**] that can be used to suggest an appropriate transformation to improve the fit when the relationship between two variables (x and y in our examples) is monotonic and has a single 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 6.1, 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 6.1, 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 6.5). When several different transformations lead to reasonable linear fits, these other factors may lead us to prefer one over another.

Example 6.7.3. Q. The scatterplot in Figure 6.2 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 6.1, 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 6.3, 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 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 6.7.4. Some physics students conducted an experiment in which they dropped steel balls from



ladder of re-expression

power	transformation
$\vdots$	$\vdots$
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	$x \mapsto \frac{1}{x^2}$
$\vdots$	$\vdots$

Figure 6.1: Bulge rules and ladder of re-expression.

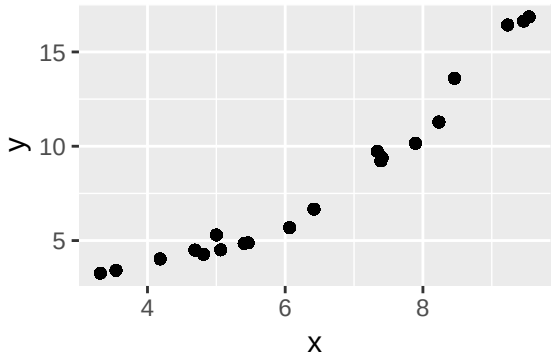


Figure 6.2: A scatterplot illustrating a non-linear relationship between x and y.

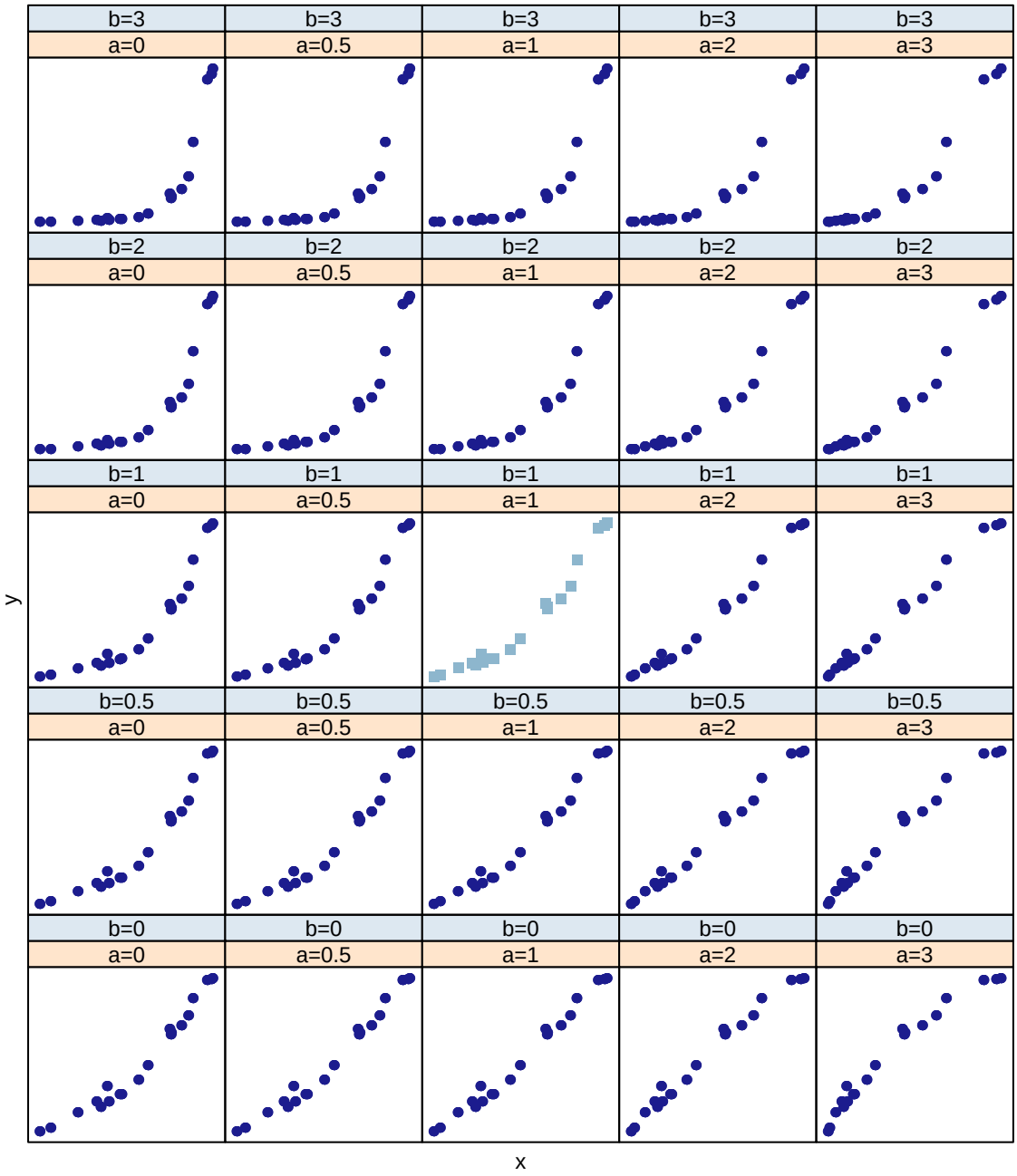
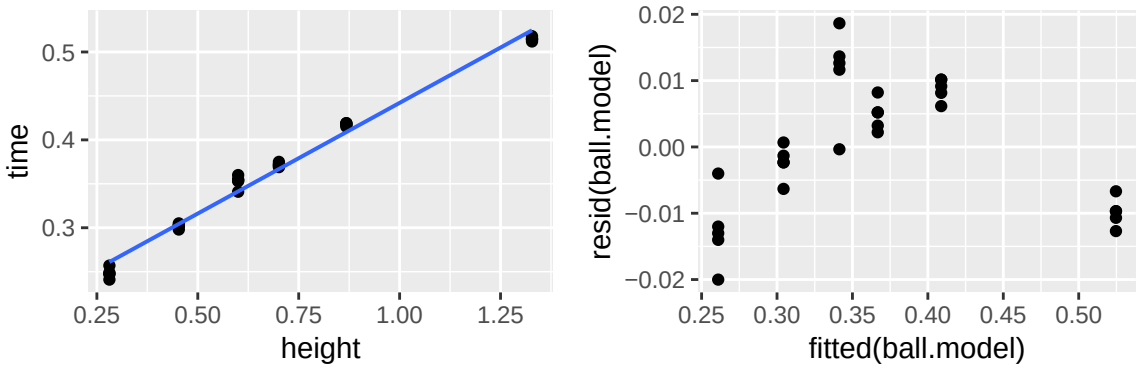


Figure 6.3: Using the ladder of re-expression to find a better fit.

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,balldrop)
summary(ball.model)


##
## Call:
## lm(formula = time ~ height, data = balldrop)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0200108 -0.0089383  0.0001623  0.0082016  0.0186519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.190243   0.004303   44.21   <2e-16
## height      0.251841   0.005516   45.66   <2e-16
##
## Residual standard error: 0.01009 on 28 degrees of freedom
## Multiple R-squared:  0.9867,Adjusted R-squared:  0.9863
## F-statistic:  2085 on 1 and 28 DF,  p-value: < 2.2e-16


gf_point(time~height,data=balldrop) %>% gf_lm()
gf_point(resid(ball.model) ~ fitted(ball.model))
```

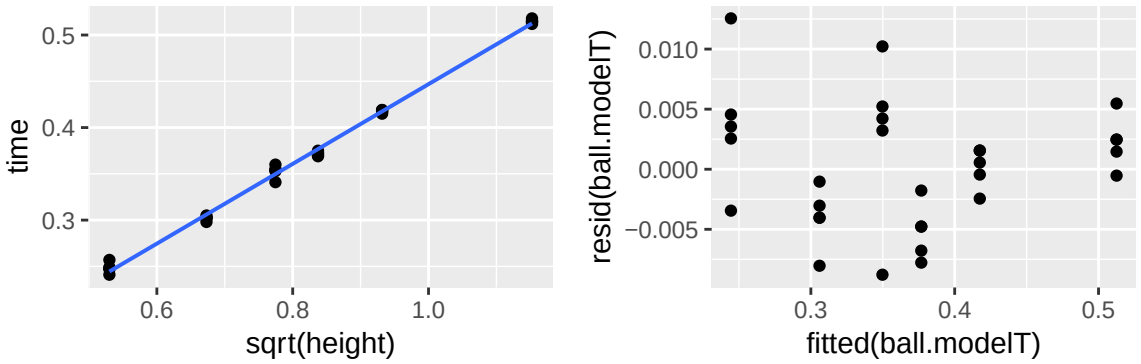


At first glance, the large value of r^2 and the reasonably good fit in the scatterplot might leave us satisfied that we have found a good model. But a look at the residual plot reveals a clear curvilinear pattern in this data. A knowledgeable physics student knows that (ignoring air resistance) the time 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`.

```
ball.modelT <- lm(time ~ sqrt(height), data=balldrop)
summary(ball.modelT)


##
## Call:
## lm(formula = time ~ sqrt(height), data = balldrop)
##
## Residuals:
```

```
##        Min         1Q     Median         3Q        Max
## -0.0087773 -0.0038851  0.0000571  0.0030558  0.0125552
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.016078   0.004084   3.937 0.000498
## sqrt(height) 0.430803   0.004863  88.580  < 2e-16
##
## Residual standard error: 0.005225 on 28 degrees of freedom
## Multiple R-squared:  0.9964,Adjusted R-squared:  0.9963
## F-statistic:  7846 on 1 and 28 DF,  p-value: < 2.2e-16
```

```
gf_point(time ~ sqrt(height), data=balldrop) %>%
  gf_lm(time~sqrt(height), data=balldrop)
gf_point(resid(ball.modelT) ~ fitted(ball.modelT))
```



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 6.7.5 for a simple attempt to deal with this problem.)

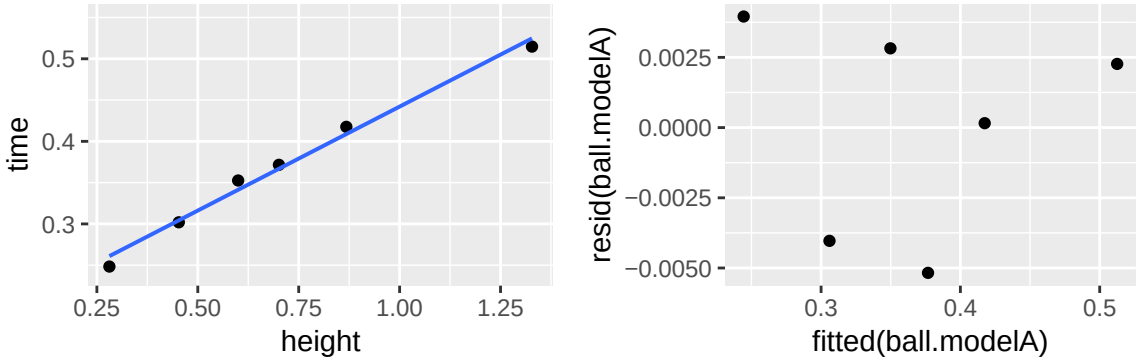
Example 6.7.5. One simple way to deal with the lack of independence in the previous example is to average all the readings made at each height. (This works reasonably well in our example because we have nearly equal numbers of observations at each height.) We pay for this data reduction in a loss of 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))

ball.modelA <- lm(time ~ sqrt(height),balldropavg)
summary(ball.modelA)
```

```
##
## Call:
## lm(formula = time ~ sqrt(height), data = balldropavg)
##
## Residuals:
##          1          2          3          4          5          6
```

```
##  0.0039552 -0.0040318  0.0028227 -0.0051717  0.0001571  0.0022686
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.016078   0.007404   2.172   0.0956
## sqrt(height) 0.430803   0.008816  48.863 1.05e-06
##
## Residual standard error: 0.004236 on 4 degrees of freedom
## Multiple R-squared:  0.9983,Adjusted R-squared:  0.9979
## F-statistic:  2388 on 1 and 4 DF,  p-value: 1.05e-06
```

```
gf_point(time~height, data=balldropavg) %>% gf_lm()
gf_point(resid(ball.modelA) ~ fitted(ball.modelA))
```



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 6.7.6. 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,soap)
summary(soap.model1)


##
## Call:
## lm(formula = Weight ~ Day, data = soap)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2436 -1.2950  0.3078  1.3942  5.5040
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 123.1408     1.3822   89.09   <2e-16
## Day          -5.5748     0.1068  -52.19   <2e-16
##
## Residual standard error: 2.949 on 13 degrees of freedom
## Multiple R-squared:  0.9953,Adjusted R-squared:  0.9949
## F-statistic:  2724 on 1 and 13 DF,  p-value: < 2.2e-16
```

The scatterplot in Figure 6.4 (darker line) indicate that the untransformed model is already a good fit.

```
soap.model2 <- lm(I(Weight^(1/3))~Day,soap)
summary(soap.model2)


##
## Call:
## lm(formula = I(Weight^(1/3)) ~ Day, data = soap)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31107 -0.13666  0.01605  0.15044  0.20095
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.297706   0.083813   63.21  < 2e-16
## Day         -0.146980   0.006477  -22.69 7.67e-12
##
```

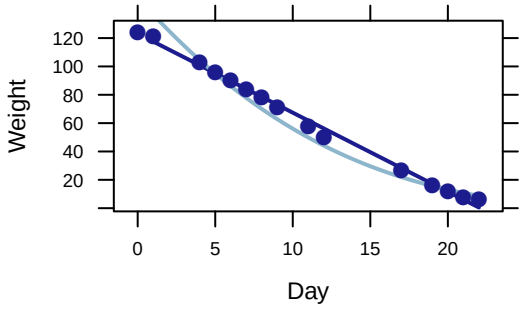


Figure 6.4: Comparing untransformed (darker) and transformed (lighter) fits to soap use data.

```
## Residual standard error: 0.1788 on 13 degrees of freedom
## Multiple R-squared:  0.9754,Adjusted R-squared:  0.9735
## F-statistic:   515 on 1 and 13 DF,  p-value: 7.669e-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 6.4 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 6.2 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 6.2), 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) 120.154672 126.126895
## Day          -5.805514  -5.344014
```

6.8 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 6.8.1. Returning to the ball dropping experiment, let's fit

$$\texttt{time} = \alpha_0 + \alpha_1 \sqrt{\texttt{height}} \tag{6.1}$$

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 (6.1). The starting point for the algorithm is specified with `start=list(alpha0=0, alpha1=1)`, which also declares that 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
## 0.01608 0.43080
##  residual sum-of-squares: 0.0007645
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.112e-07
```

or

```
coef(nls.model)
```

```
##     alpha0     alpha1
## 0.01607833 0.43080348
```

A more complete summary can be obtained by

```
summary(nls.model)
```

```
##
```

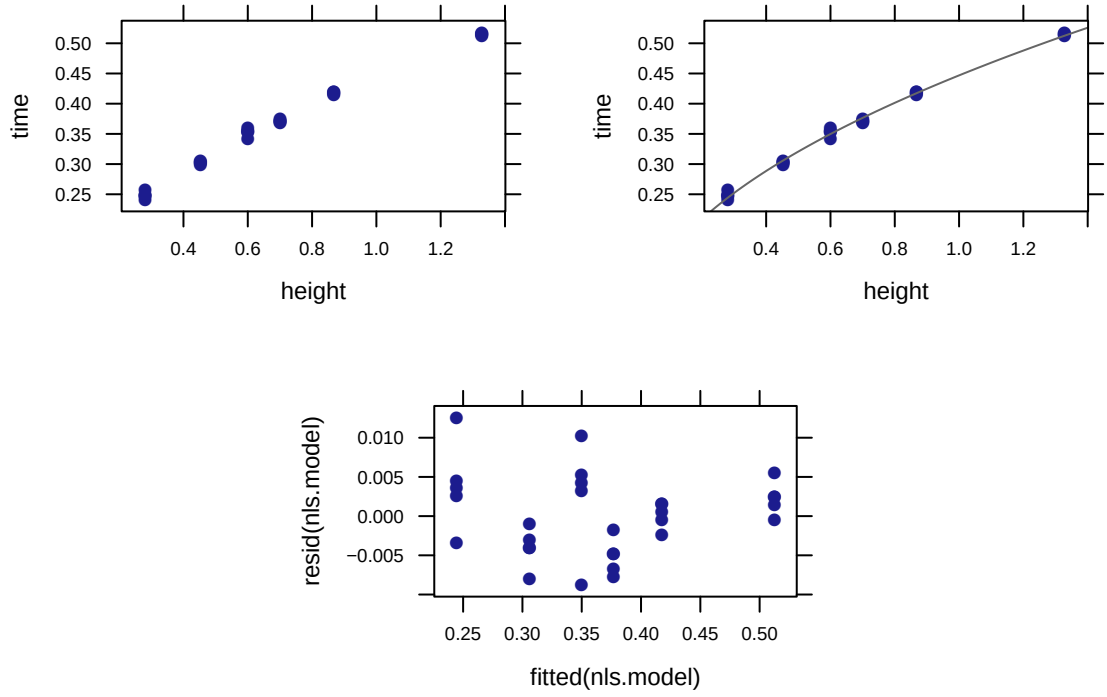
```
## Formula: time ~ alpha0 + alpha1 * sqrt(height)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## alpha0 0.016078   0.004084   3.937 0.000498
## alpha1 0.430803   0.004863  88.580  < 2e-16
##
## Residual standard error: 0.005225 on 28 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.112e-07
```

We can restrict our attention to the coefficients table with

```
coef(summary(nls.model))
```

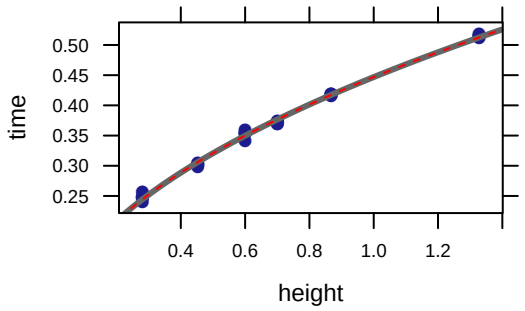
```
##          Estimate  Std. Error   t value     Pr(>|t|)
## alpha0 0.01607833 0.004084015  3.936894 4.975519e-04
## alpha1 0.43080348 0.004863416 88.580433 7.732182e-36
```

```
f <- makeFun(nls.model)
xyplot( time ~ height, data=balldrop )
plotFun( f(height) ~ height, add=TRUE, col='gray40' )
xyplot( 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)
xyplot( time ~ height, data=balldrop )
plotFun( f(height) ~ height, add=TRUE, col='gray40', lwd=3 )
plotFun(g(height) ~ height, add=TRUE, col='red', lwd=1, lty=2)
```



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
## 0.01607833 0.43080348


coef(lm.model)


##  (Intercept) sqrt(height)
##   0.01607833   0.43080348
```

Example 6.8.2. 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 0.4472102 0.001342627 333.08590 6.333427e-52
## power 0.4796679 0.005805313  82.62567 5.387914e-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) -0.807610 0.004330482 -186.49426 7.101233e-45
## log(height)  0.471911 0.006424548   73.45435 1.431476e-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)
##   0.4459225
```

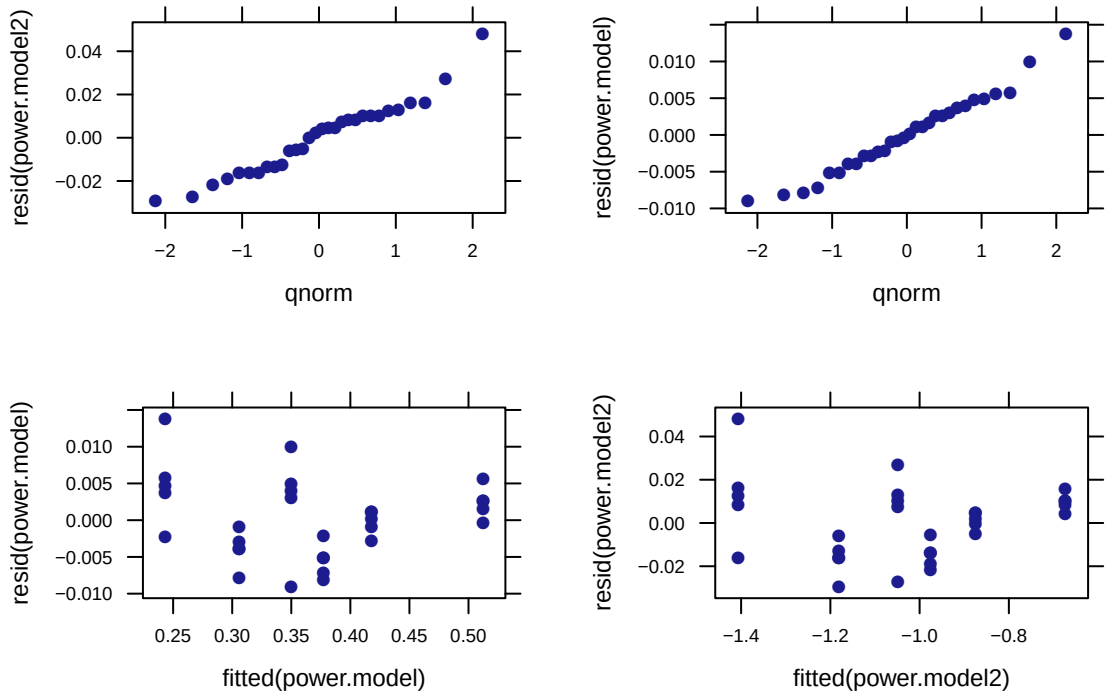
We can use the delta method to estimate the uncertainty. Since $\frac{d}{dx}e^x = e^x$ the uncertainty is approximately

$$0.4459225 \cdot 0.0043305 = 0.0019311$$

Example 6.8.3. 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.

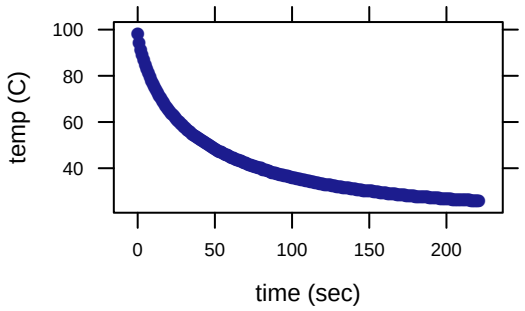
```
qqmath(resid(power.model2))
qqmath(resid(power.model))
xyplot( resid(power.model) ~ fitted(power.model) )
xyplot( resid(power.model2) ~ fitted(power.model2) )
```



Now let's take a look at an example where we need the extra flexibilty of the nonlinear least squares approach.

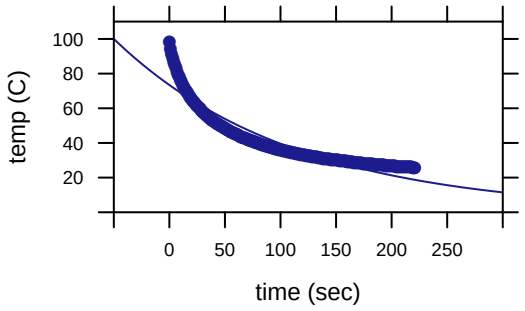
Example 6.8.4. A professor at Macalester College put hot water in a mug and recorded the temperature as it cooled. Let's see if we can fit a reasonable model to this data

```
xyplot( 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)
xyplot( temp ~ time, data=CoolingWater, xlim=c(-50,300), ylim=c(0,110),
        ylab="temp (C)", xlab="time (sec)")
plotFun( f1(time) ~ time, add=TRUE)
```

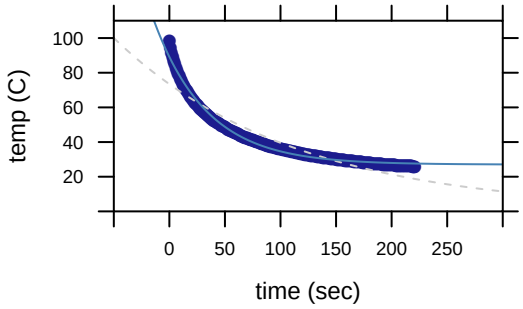


That doesn't fit very well, and there is a good reason. The model says that eventually the water will freeze because

$$\lim_{t\to\infty} Ae^{-kt} = 0$$

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)
xyplot( temp ~ time, data=CoolingWater, xlim=c(-50,300), ylim=c(0,110),
        ylab="temp (C)", xlab="time (sec)")
plotFun( f1(time) ~ time, add=TRUE, lty=2, col="gray80")
plotFun( f2(time) ~ time, add=TRUE, col = "steelblue")
```



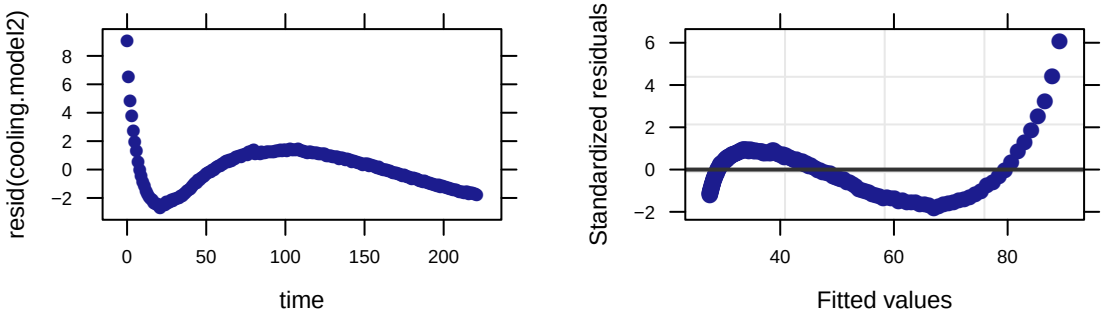
This fits much better. Furthermore, this model can be derived from a differential equation

$$\frac{dT}{dt} = -k(T_0 - T_{\text{ambient}}) \ ,$$

known as Newton's Law of Cooling.

Let's take a look at the residual plot

```
xyplot( 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]

6.8.1 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.

Exercises

6.1 In Example 6.7.4, we applied a square root transformation to the height. Is there another transformation that yields an even better fit?

6.2 Remove the last few days from the `soap` data set and refit the models in Example 6.7.6. How much do things change? Do the residuals look better, or is there still some cause for concern?

6.3 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) $y = \dfrac{100}{1+e^{a+bx}}$.

6.4 What happens to the role of the error terms (ε) when we transform the data? For each transformation from Exercise 6.3, 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.

6.5 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?

6.6 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.

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?

6.7 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?

6.8 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.

6.9 The data set `actgpa` 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.

6.10 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 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.

6.11 Construct a plot that reveals a likely systematic problem with the `drag` (see Exercise 6.10) data set. Speculate about a potential cause for this.

6.12 Exercise 6.11 suggests that some of the data should be removed before analyzing the `drag` data set. Redo Exercise 6.10 after removing this data.

6.13 The `spheres` data set 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.

6.14 The `spheres` data set 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?

6.15 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?