Math 241: Engineering Statistics

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

Spring 2015

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, use the links from our course Moodle site, or connect directly at `http://rstudio.calvin.edu:8787`.

1.1.1 Logging in

When you navigate to the RStudio server, you will be prompted to login. Your login is your Calvin Gmail address, and your password is the corresponding password. 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.

```
5 + 3
```

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

```
15.3 * 23.4
```

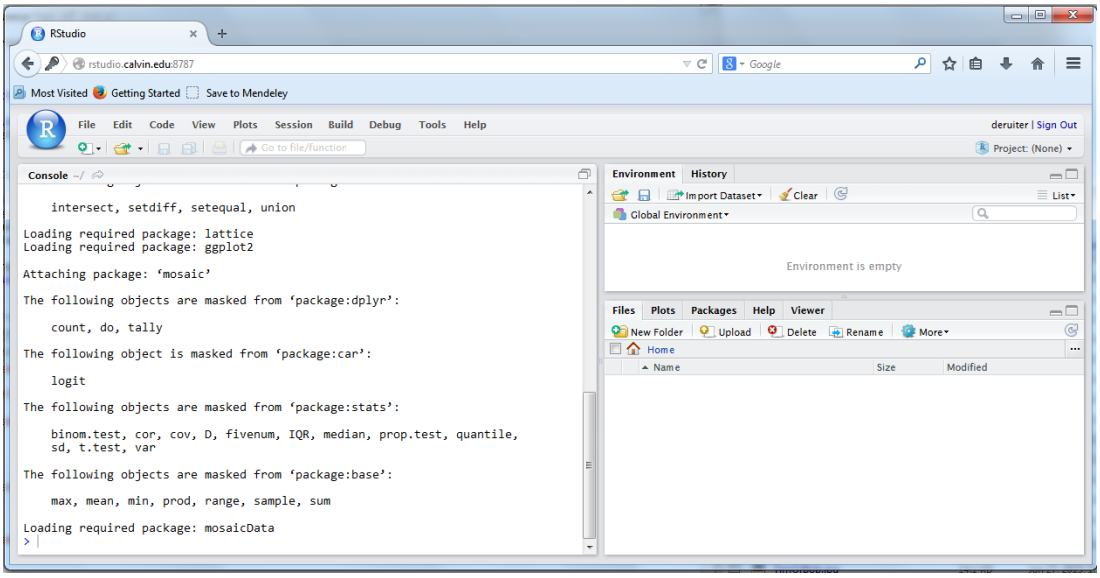


Figure 1.1: Welcome to RStudio.

```
## [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
```

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

```
## [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. There are several ways to load packages, as detailed below.

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

- `mosaic` (a package from Project MOSAIC; this should already be loaded)
- `DAAG` (a package that goes with the book *Data Analysis and Graphics*; probably not loaded, check the box to load it.)

You an also load packages by typing, for example

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

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

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

4. Hit ESCAPE to break out of a mess.

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

1.2 Data in R

1.2.1 Data Frames

Most often, data sets in R are stored in a structure called a **data frame**. A data frame is designed to hold "rectangular data". The people or things being measured or observed are called **observational units** (or subjects or cases when they are people). For measurements collected over time, the observational units would be the individual time-points at which data points were collected. Each observational unit is represented by one row in the data frame. The different pieces of information recorded for each observational unit are stored in separate columns, called **variables**.

1.2.2 Data in Packages

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

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

```
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 (observational units) in this study:

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

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

```
head(HELPrct)
```

```
##   age anysubstatus anysub cesd d1 daysanysub dayslink drugrisk e2b female    sex g1b
## 1  37            1    yes   49  3        177      225        0  NA      0   male yes
## 2  37            1    yes   30 22          2       NA        0  NA      0   male yes
## 3  26            1    yes   39  0          3      365       20  NA      0   male  no
## 4  39            1    yes   15  2        189      343        0   1      1 female  no
## 5  32            1    yes   39 12          2       57        0   1      0   male  no
## 6  47            1    yes    6  1         31      365        0  NA      1 female  no
##   homeless i1 i2 id indtot linkstatus link       mcs      pcs pss_fr racegrp satreat
## 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
## 1       4   cocaine   yes
## 2       7   alcohol   yes
## 3       2    heroin    no
## 4       4    heroin    no
## 5       6   cocaine    no
## 6       5   cocaine   yes
```

```
dim(HELPrct)
```

```
## [1] 453  27
```

The commands and R output above tell us that there are 453 observational units in this data set and 27 variables. That's plenty of variables to get us started with exploration of data.

1.2.4 The KidsFeet data set

Here is another data set in the mosaic package:

```
head(KidsFeet)
```

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

1.2.5 The oldfaith data set

A final example data set comes from the 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.

```
head(oldfaith)
```

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

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

```
nrow(oldfaith)
```

```
## [1] 270
```

```
ncol(oldfaith)
```

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

```
dim(oldfaith)
```

```
## [1] 270   2
```

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

1.2.6 Using your own data

We will postpone for now a discussion about getting your own data into RStudio, but any data you can get into a reasonable format (like csv) can be imported into RStudio pretty easily.

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 (`histogram()`, `bargraph()`, `densityplot()`, `bwplot()`, etc.)
- The name of the variable
- The name of the data frame this variable is a part of.

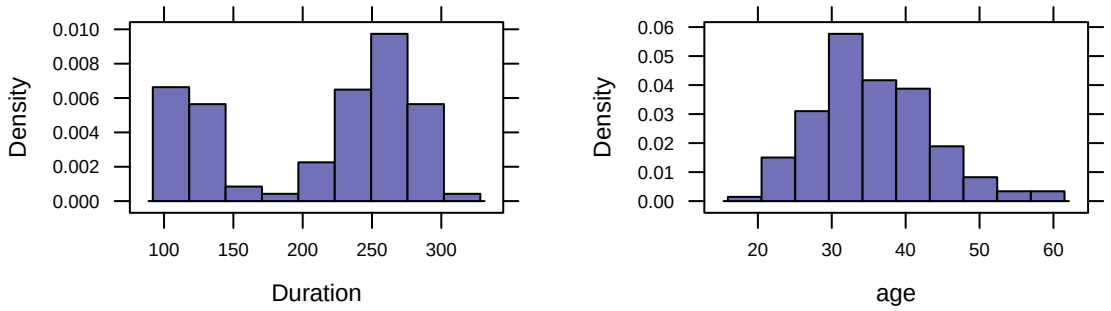
Note: The same syntax works for numerical summaries as well – thanks to the `mosaic` package we can apply the same syntax for `mean()`, `median()`, `sd()`, `var()`, `max()`, `min()`, etc. Later we will use this syntax again to fit linear and nonlinear models to data.

1.3.1 Histograms (and density plots) for quantitative variables

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

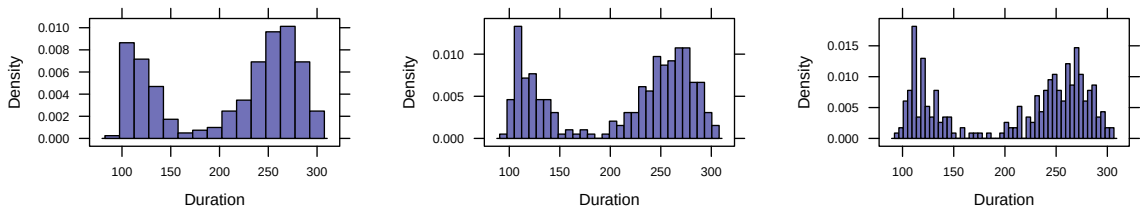
Here are a couple examples:

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



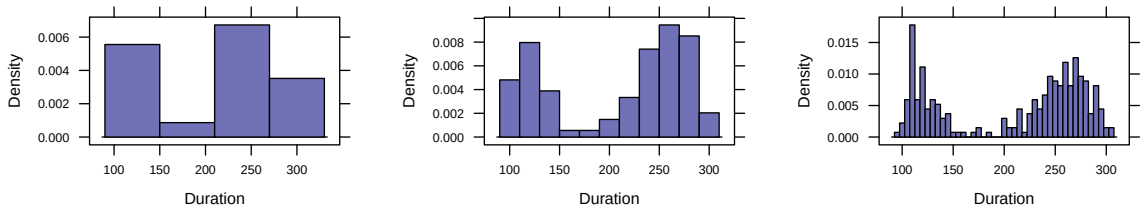
We can control the (approximate) number of bins using the `nint` argument, which may be abbreviated as `n`. The number of bins (and to a lesser extent the positions of the bins) can make a histogram look quite different.

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



The `histogram()`[1] function in the `mosaic` package lets you describe the bins in terms of center and width instead of in terms of the number of bins. This is especially nice for count data.

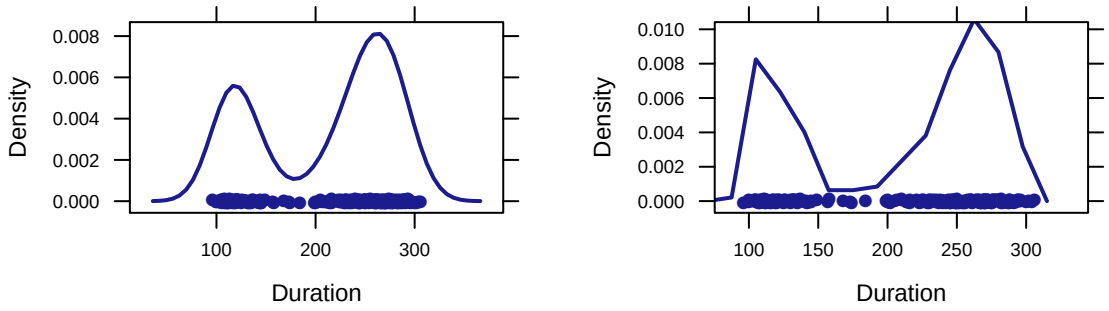
```
histogram(~Duration, data = oldfaith, width = 60)
histogram(~Duration, data = oldfaith, width = 20)
histogram(~Duration, data = oldfaith, width = 5)
```



R also provides a "smooth" version called a density plot and a triangular version called a frequency polygon; just change the function name from `histogram()` to `densityplot()` or `freqpolygon()`.

```
densityplot(~Duration, data = oldfaith)
freqpolygon(~Duration, data = oldfaith)
```

[1] The `mosaic` version of the `histogram()` function has some extra features (like the `width` argument) but is otherwise is very similar to regular `histogram()` function in `lattice`.



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 "bump" where there is a lot of data.

bimodal There are two "bumps".

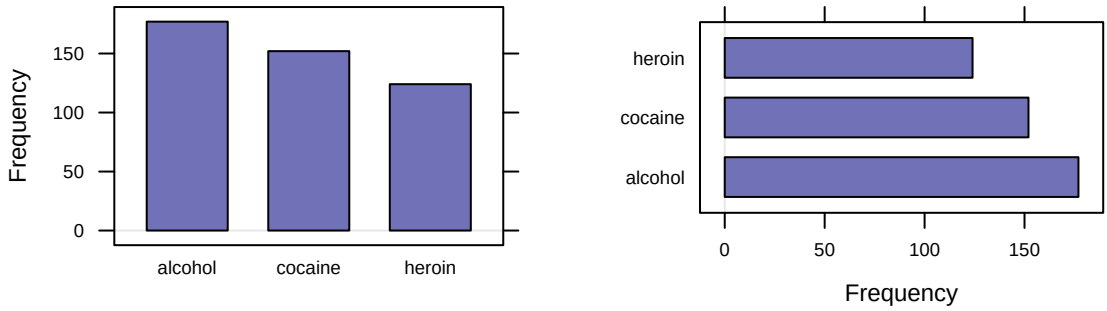
outlier An observation that does not fit the overall pattern of the rest of the data.

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

1.3.3 Bar graphs for categorical variables

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

```
bargraph(~substance, data = HELPrct)
bargraph(~substance, data = HELPrct, horizontal = TRUE)
```



A side note: we will be unlikely to use 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.

1.4 Looking at Multiple Variables at Once

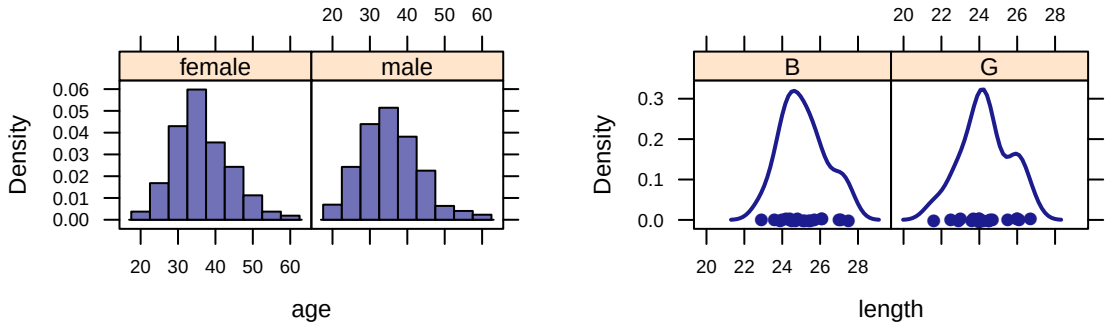
1.4.1 Conditional plots

The formula for a `lattice` 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)
```

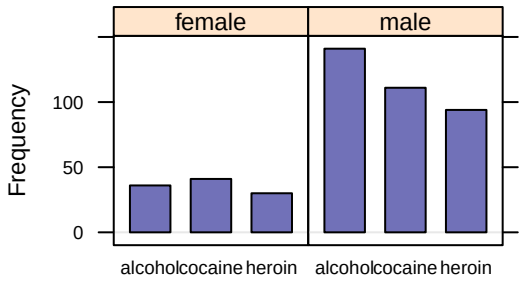
For example, we might like to see how the ages of men and women compare in the HELP study, or whether the distribution of weights of male mosquitoes is different from the distribution for females.

```
histogram(~age | sex, HELPrct, width = 5)
densityplot(~length | sex, KidsFeet)
```



We can do the same thing for bar graphs.

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

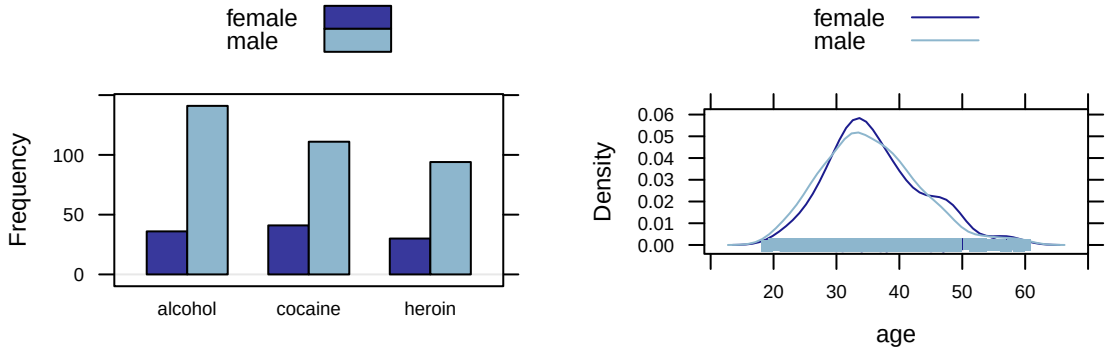


1.4.2 Grouping

Another way to look at multiple groups simultaneously is by using the `groups` argument. What `groups` does depends a bit on the type of graph, but it will put the information in one panel rather than multiple panels. Using `groups` with `histogram()` doesn't work so well because it is difficult to overlay histograms.[2] Density plots work better when you wish to look at the shapes of several distributions in a single plot panel.

Here are some examples. We use `auto.key=TRUE` to build a simple legend so we can tell which groups are which.

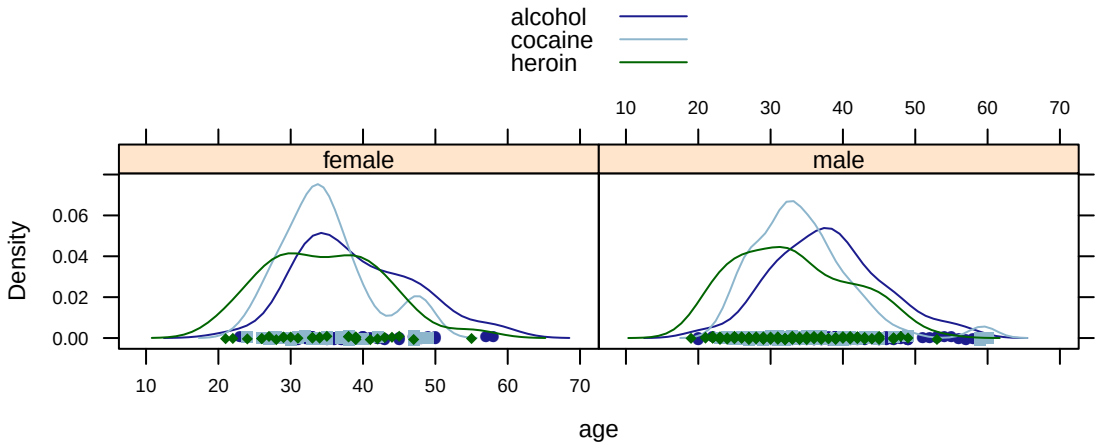
```
bargraph(~substance, groups = sex, data = HELPrct, auto.key = TRUE)
densityplot(~age, groups = sex, data = HELPrct, auto.key = TRUE)
```



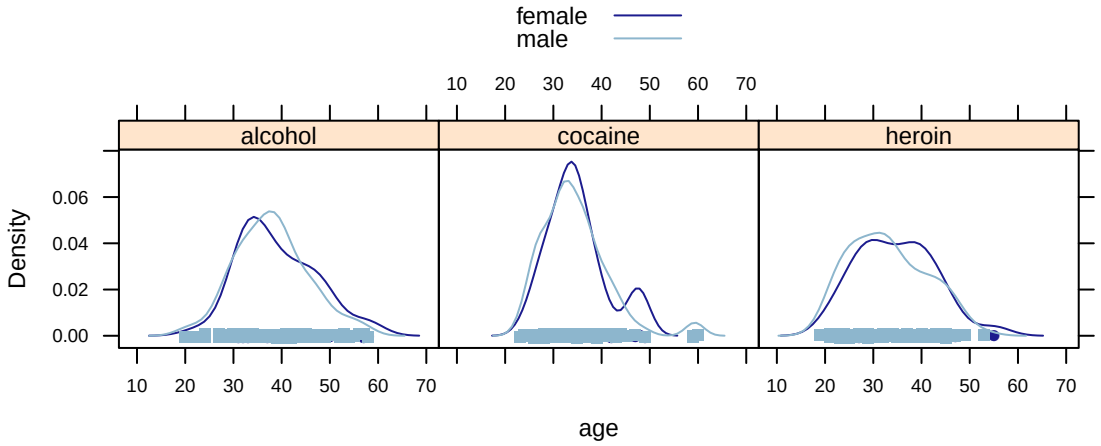
We can even combine grouping and conditioning in the same plot.

```
densityplot(~age | sex, groups = substance, data = HELPrct, auto.key = TRUE)
```

[2] The `mosaic` function `histogram()` does do something meaningful with `groups` in some situations.




```
densityplot(~age | substance, groups = sex, data = HELPrct, auto.key = TRUE, layout = c(3,
    1))
```



This plot shows that for each substance, the age distributions of men and women are quite similar, but that the distributions differ from substance to substance.

1.4.3 Scatterplots

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

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

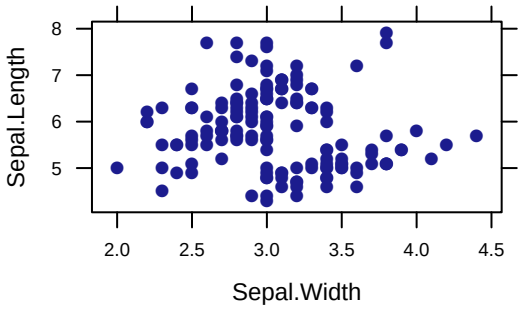
Notice that now we have something on both sides of the ~ since we need to tell R about two variables.

```
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
```

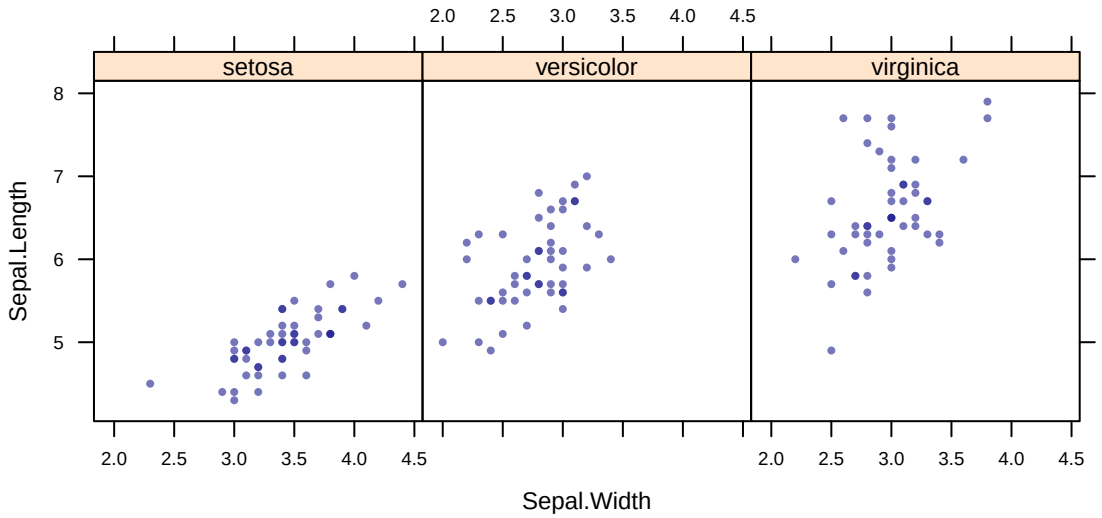
```
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

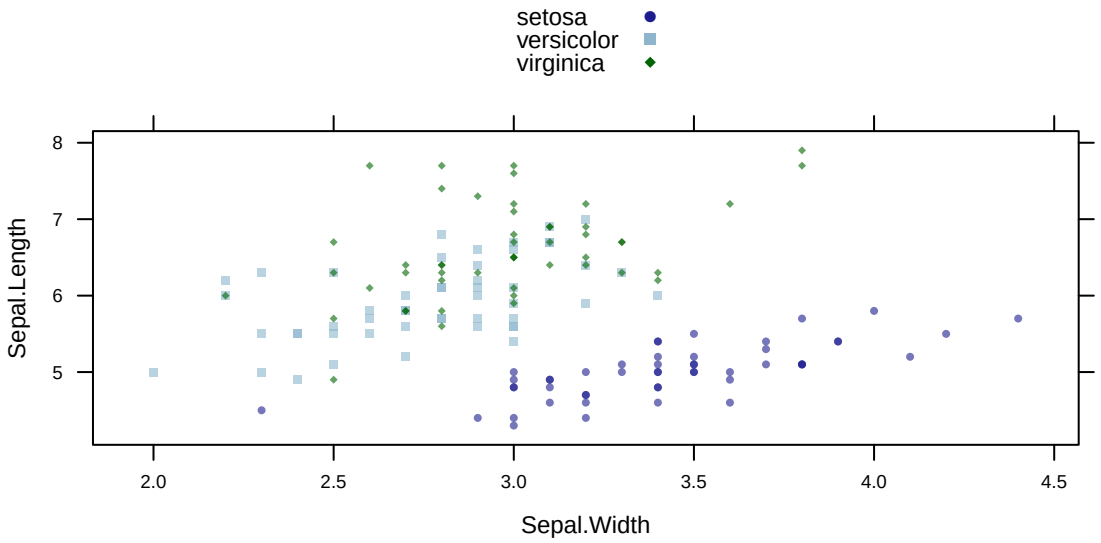
```
xyplot( Sepal.Length ~ Sepal.Width, data=iris )
```



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 `cex`.

```
xyplot( Sepal.Length ~ Sepal.Width | Species, data=iris, alpha=.6, cex=.5 )
xyplot( Sepal.Length ~ Sepal.Width, groups = Species, data=iris, alpha=.6, cex=.5,
          auto.key=TRUE )
```





1.5 Exporting Plots

You can save plots to files or copy them to the clipboard using the Export menu in the Plots tab. It is quite simple to copy the plots to the clipboard and then paste them into a Word document, for example. You can even adjust the height and width of the plot first to get it the shape you want.

1.6 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).

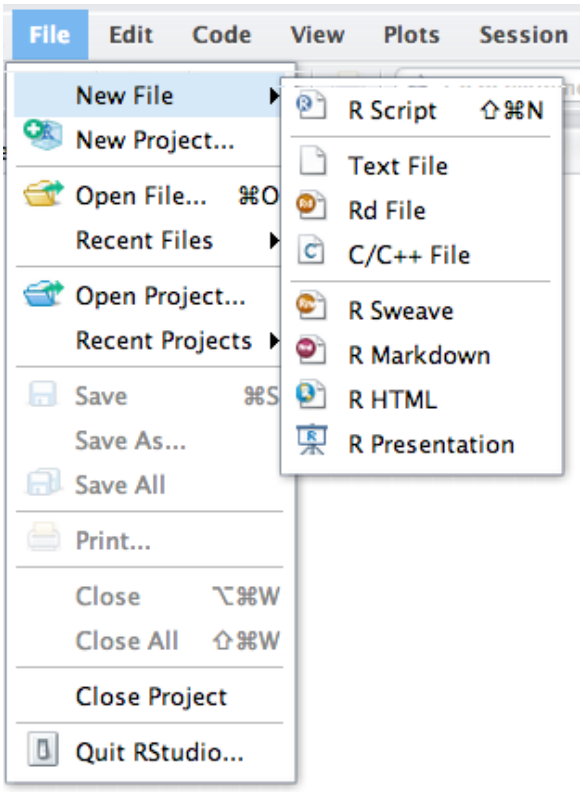
RStudio makes it easy to use techniques of reproducible research to create documents that include text, R commands, R output, and R graphics.

1.6.1 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 an HTML file, so it is especially good for producing web documents.[3]

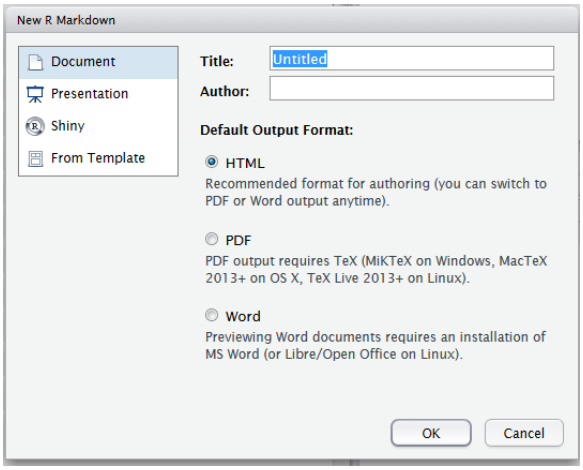
Creating a new document

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



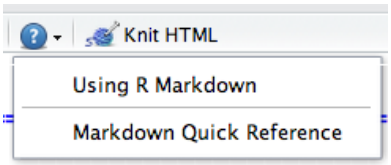
A small pop-up window will appear; for current purposes, you can select all the default options (a Document, with HTML output format, with the Title and Author blanks filled in or left blank, as you wish).

[3] 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.

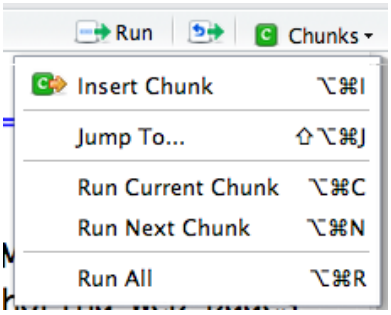


When you do this, a file editing pane will open with a template inserted. If you click on "Knit HTML", RStudio will turn this into an HTML file and display it for you. Give it a try. You will be asked to name your file if you haven't already done so. If you are using the RStudio server in a browser,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

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



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

In addition to knitting the document to HTML, you can do a number of other things that will make your work more efficient. In the "Chunk" menu, you can choose to run a single chunk or all the chunks. This will execute your commands in the console so you can make sure your R code is working one chunk at a time. There is also a "run" button that allows you to run just one line from within a chunk.

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.

Chunk options

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

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

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

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

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

1.6.2 knitr/latex

There is another system that produces PDFs by combining LaTeX and R, using the Rpackage `knitr`. This is the system used to create this document; it gives much more control over document formatting. The quality is good enough for professional publishing. If you already knwow LaTeX, it is very easy to learn. If you don't know LaTeX, then you need to learn the basics of LaTeX to get going, which is not covered in detail here. If you are interested in learning more, consult the documentation for the `knitr` package, or the knitr website.

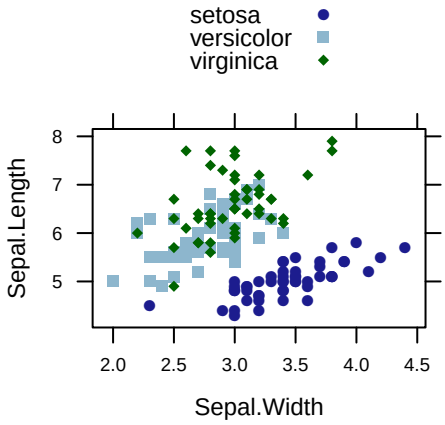
1.7 Customizing Graphics: A Few Bells and Whistles

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.

1.7.1 auto.key

`auto.key=TRUE` turns on a simple legend for the grouping variable. (There are ways to have more control, if you need it.)

```
xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data = iris, auto.key = TRUE)
```

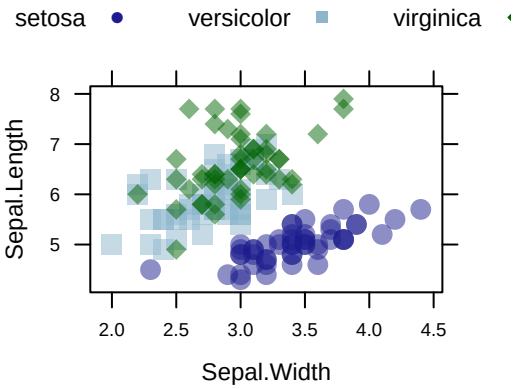


1.7.2 alpha, cex

Sometimes it is nice to have elements of a plot be partly transparent. When such elements overlap, they get darker, showing us where data are "piling up." Setting the `alpha` argument to a value between 0 and 1 controls the degree of transparency: 1 is completely opaque, 0 is invisible. The `cex` argument controls "character expansion" and can be used to make the plotting "characters" larger or smaller by specifying the scaling ratio.

Here is another example using data on 150 iris plants of three species.

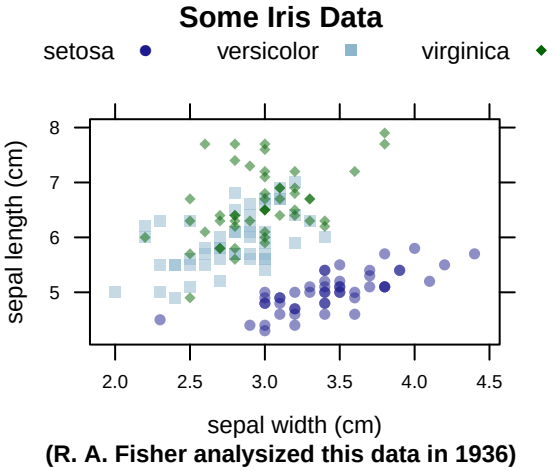
```
xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data = iris, auto.key = list(columns = 3),
    alpha = 0.5, cex = 1.3)
```



main, sub, xlab, ylab

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

```
xyplot(Sepal.Length ~ Sepal.Width, groups = Species, data = iris, main = "Some Iris Data",
    sub = "(R. A. Fisher analysized this data in 1936)", xlab = "sepal width (cm)", ylab = "sepal length
    alpha = 0.5, auto.key = list(columns = 3))
```



(R. A. Fisher analysized this data in 1936)

layout

`layout` can be used to control the arrangement of panels in a multi-panel plot. The format is

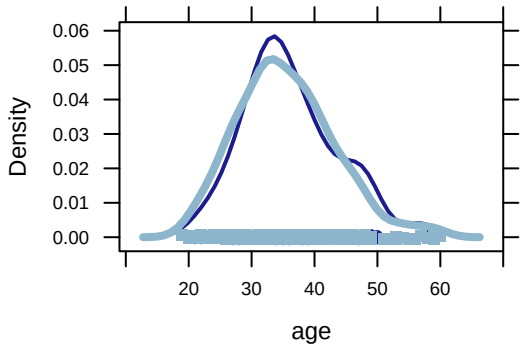
```
layout = c(cols, rows)
```

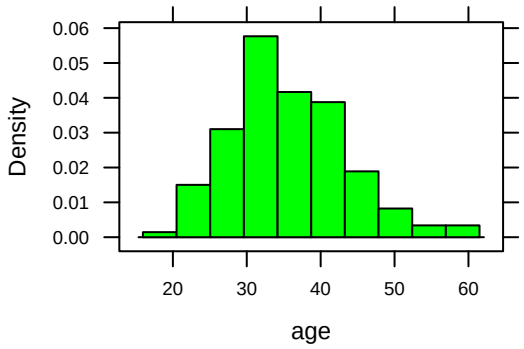
where `cols` is the number of columns and `rows` is the number of rows. (Columns first because that is the x-coordinate of the plot.)

lty, lwd, pch, col

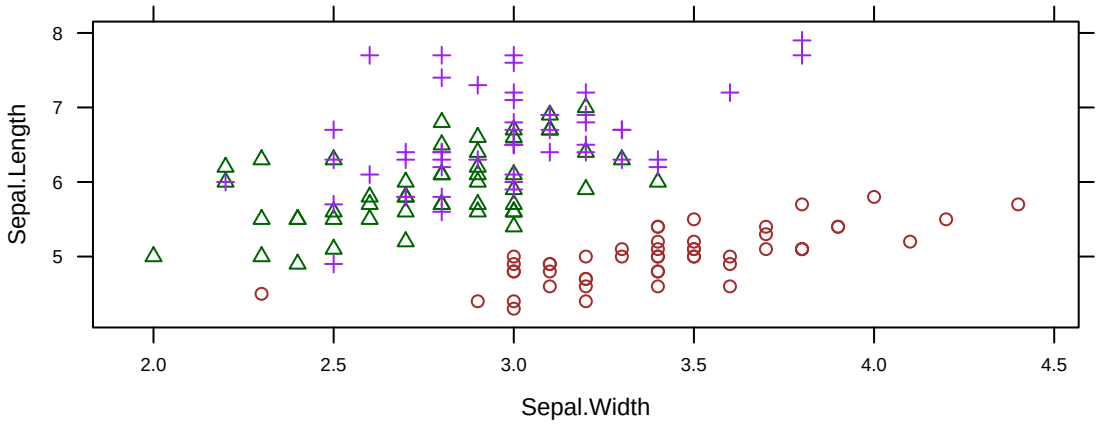
These can be used to change the line type, line width, plot symbol, and color, respectively. To specify multiples (one for each group), use the `c()` function (to remember this function, remember that "c" is for "concatenate"). An example is below.

```
densityplot(~age, data = HELPrct, groups = sex, lty = 1, lwd = c(2, 4))
histogram(~age, data = HELPrct, col = "green")
```






```
# There are 25 numbered plot symbols
xyplot( Sepal.Length ~ Sepal.Width, data=iris, groups=Species,
            pch=c(1,2,3), col=c('brown', 'darkgreen', 'purple'), cex=.75 )
```



Note: If you change the colors and symbols this way, they will *not* match what is generated in the legend using `auto.key=TRUE`. So it can be better to set these things in a different way if you are using `groups`. See below.

You can see a list of the hundreds of available color names using

```
colors()
```

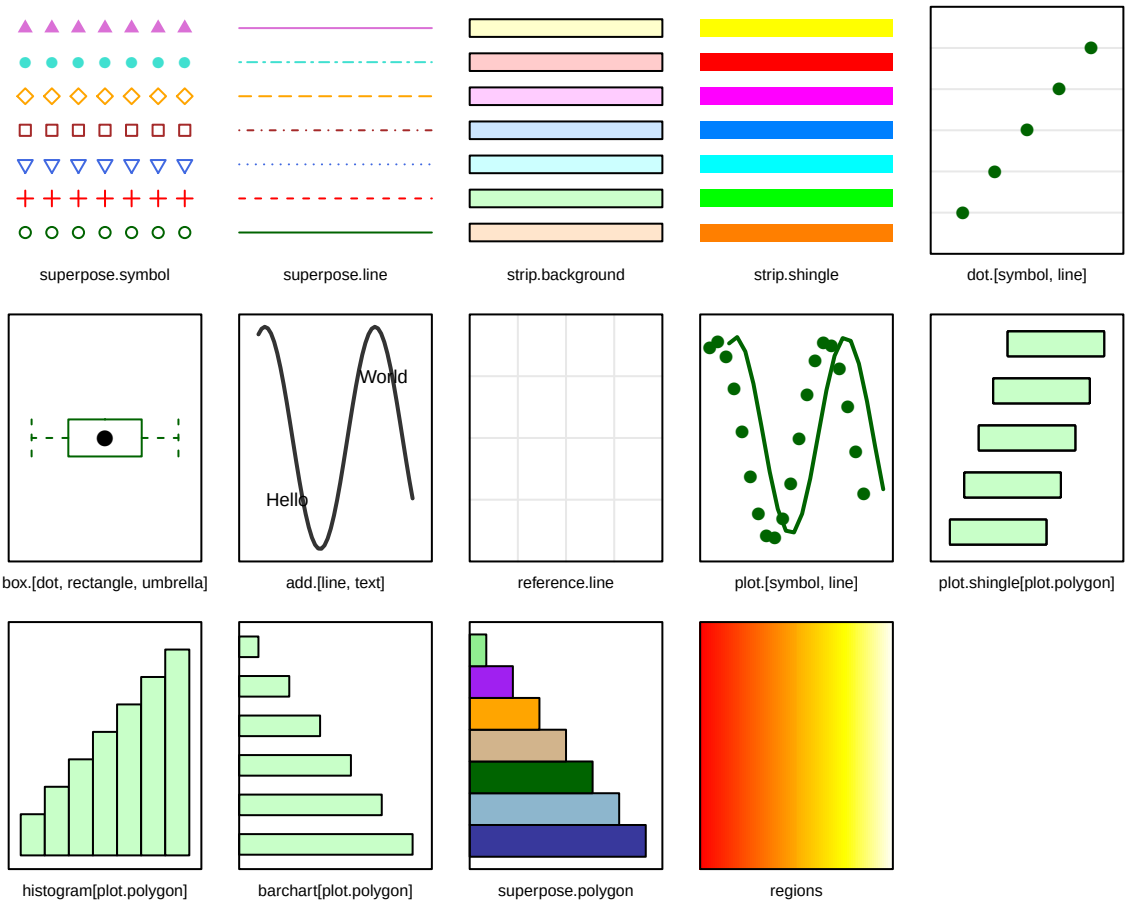
1.7.3 trellis.par.set()

Default settings for lattice graphics are set using `trellis.par.set()`. Don't like the default font sizes? You can change them! For example, change to a 7 point (base) font using

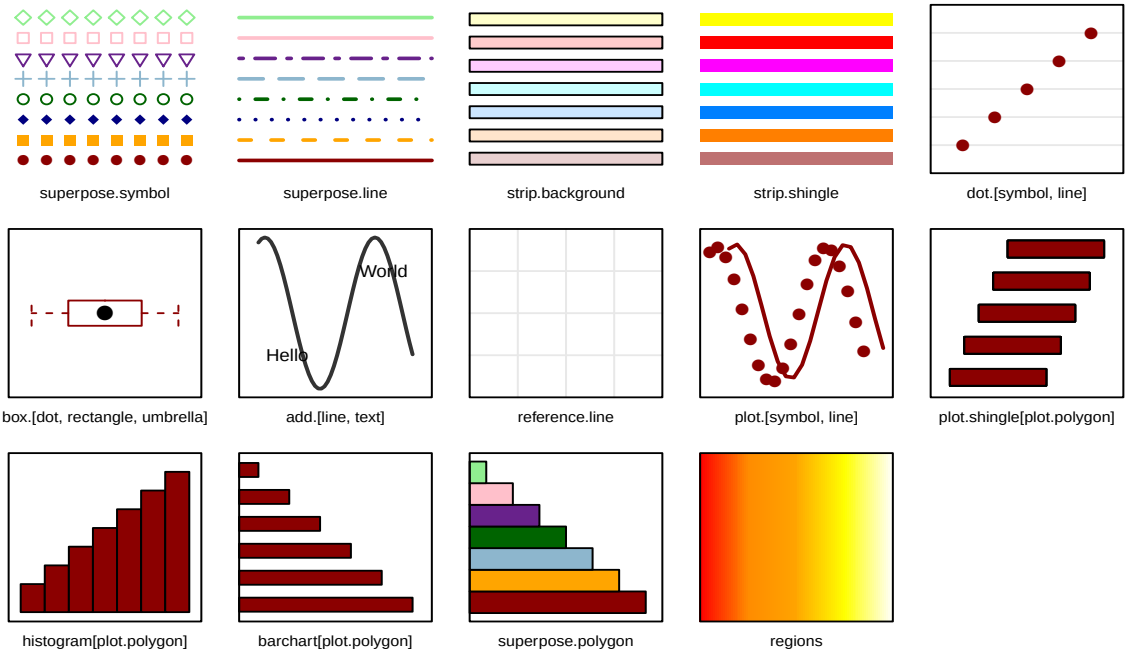
```
trellis.par.set(fontsize = list(text = 7))  # base size for text is 7 point
```

Nearly every feature of a lattice plot can be controlled: fonts, colors, symbols, line thicknesses, colors, etc. Rather than describe them all here, we'll mention only that groups of these settings can be collected into a theme. `show.settings()` will show you what the current theme looks like. Below are a few examples of changing the theme settings, then viewing the results.

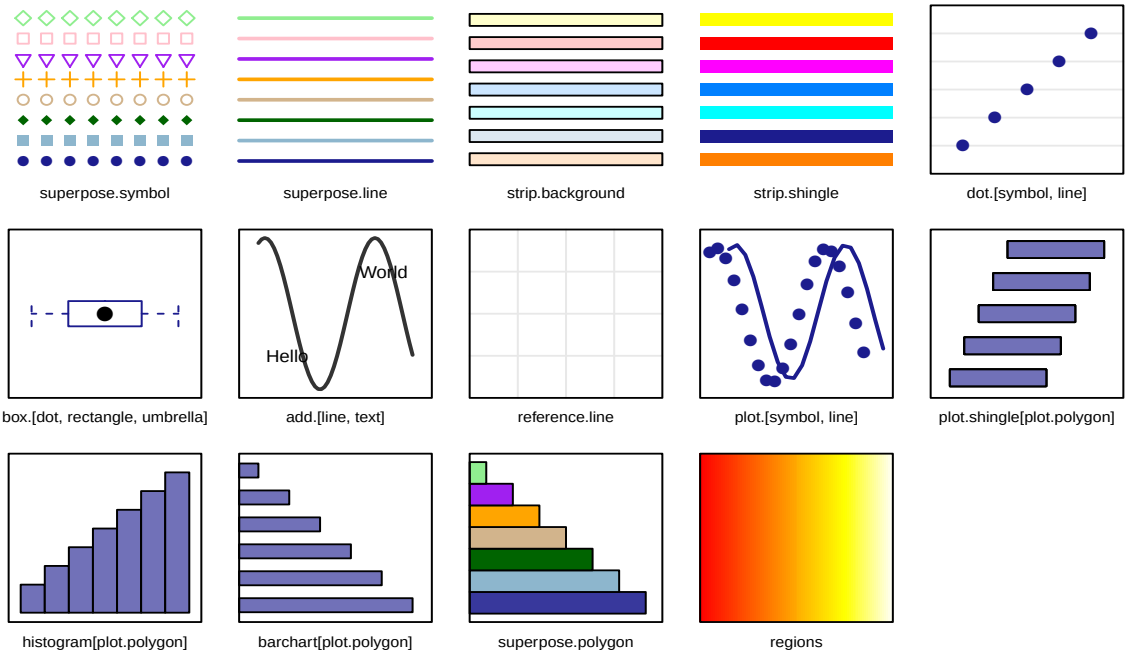
```
trellis.par.set(theme = col.whitebg())  # a theme in the lattice package
show.settings()
```


superpose.symbol
superpose.line
strip.background
strip.shingle
dot.[symbol, line]
box.[dot, rectangle, umbrella]
World
Hello
add.[line, text]
reference.line
plot.[symbol, line]
plot.shingle[plot.polygon]
histogram[plot.polygon]
barchart[plot.polygon]
superpose.polygon
regions

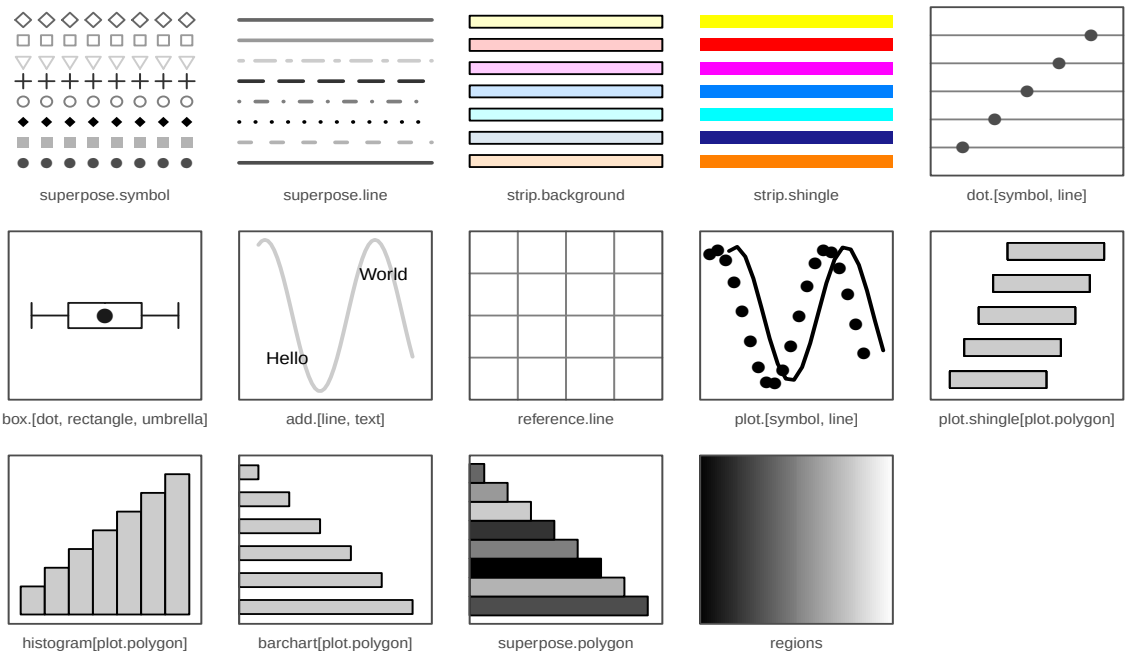
```
require(abd)
trellis.par.set(theme = col.abd())  # a theme in the abd package
show.settings()
```




```
trellis.par.set(theme = col.mosaic)  # a theme in the mosaic package
show.settings()
```



```
trellis.par.set(theme = col.mosaic(bw = TRUE))  # a b/w theme in the mosaic package
show.settings()
```




```
trellis.par.set(theme = col.mosaic())  # back to the mosaic theme
trellis.par.set(fontsize = list(text = 9))  # and back to a 10 point font
```

Want to save your settings?

```
# save current settings
mySettings <- trellis.par.get()
# switch to abd defaults
trellis.par.set(theme = col.abd())
# switch back to my saved settings
trellis.par.set(mySettings)
```

1.8 Getting Help in RStudio

1.8.1 The RStudio help system

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

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

You can search the help system using

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

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

1.8.2 Tab completion

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

1.8.3 History

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

1.8.4 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.

1.9 Graphical Summaries – Important Ideas

1.9.1 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.

1.9.2 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.

1.9.3 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.

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

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

1.9.5 Area = (relative) frequency

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

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

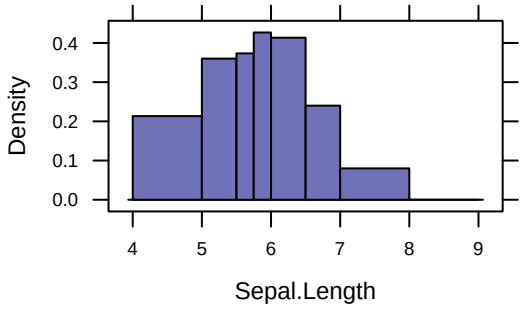
Plots that violate this principle can be deceptive and distort the true nature of the data.

An Example: Histogram with unequal bin widths

It is possible to make histograms with bins that have different widths. But in this case it is important that the height of the bars is chosen so that area (*NOT height*) is proportional to frequency. Using height instead of area would distort the picture.

When unequal bin sizes are specified, `histogram()` by default chooses the density scale:

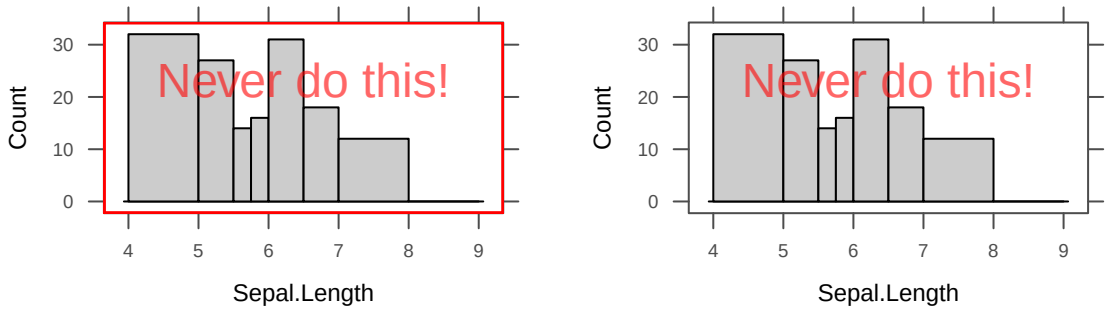
```
histogram(~Sepal.Length, data = iris, breaks = c(4, 5, 5.5, 5.75, 6, 6.5, 7, 8, 9))
```



The density scale is important. It tells R to use a scale such that the area (height $\times$ width) of the rectangles is equal to the relative frequency. For example, the bar from 5.0 to 5.5 has width $\frac{1}{2}$ and height about 0.36, so the area is 0.18, which means approximately 18% of the sepal lengths are between 5.0 and 5.5.

It would be incorrect to choose `type="count"` or `type="proportion"` since this distorts the picture of the data. Fortunately, R will warn you if you try:

```
## Warning in histogram.formula(~Sepal.Length, data = iris, breaks = c(4, 5, :  type='count' can
be misleading in this context
```



Notice how different this looks. Now the heights are equal to the relative frequency, but this makes the wider bars have too much area.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Use `subset()` 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?

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

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

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

1.14 The `SnowGR` contains historical data on snowfall in Grand Rapids, MI. The snowfall 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 wiggle (`~`) a formula.)

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

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

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

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

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

```
##
##  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
```

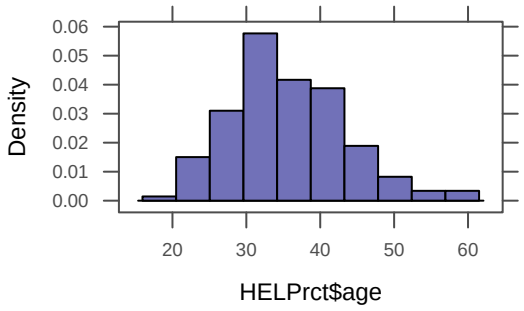
We'll call this interface the `$`-interface.

```
table(HELPrct$substance)

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

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

##
##  alcohol  cocaine   heroin
## 39.07285 33.55408 27.37307

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

##
##   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.

```
histogram(HELPrct$age)
```



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

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)
```

```
##
## 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)

It is often more convenient 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 <- transform(HELPrct, 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 binnedAge
## 1       4   cocaine   yes   (30,40]
## 2       7   alcohol   yes   (30,40]
## 3       2    heroin    no   (20,30]
## 4       4    heroin    no   (30,40]
## 5       6   cocaine    no   (30,40]
## 6       5   cocaine   yes   (40,50]
```

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

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

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

```
table( HELPrct$binnedAge)
```

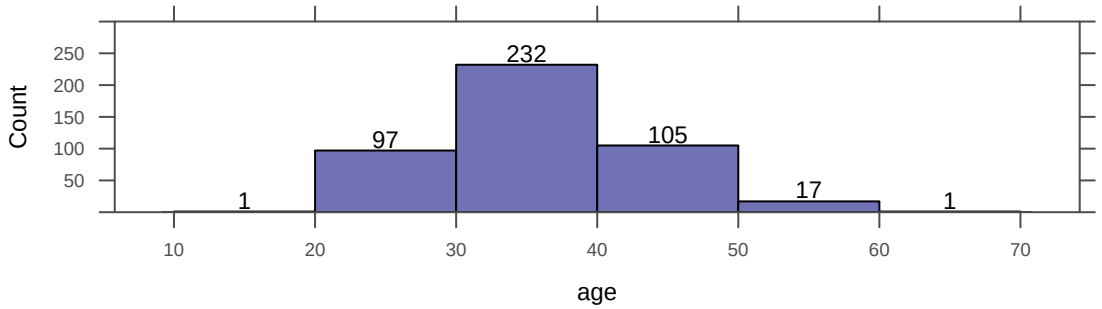
```
##
## [10,20) [20,30) [30,40) [40,50) [50,60) [60,70)
##       1      97     232     105      17       1
```

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

Labeling a histogram

The `histogram()` function offers you the option of adding the counts to the graph, via the logical (TRUE/FALSE) `label` argument.

```
histogram(~age, data = HELPrct, label = TRUE, type = "count", width = 10, center = 5, ylim = c(0,
    300), right = FALSE)
```



2.1.3 Cross-tables: Tabulating two or more variables

`xtabs()` 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
```

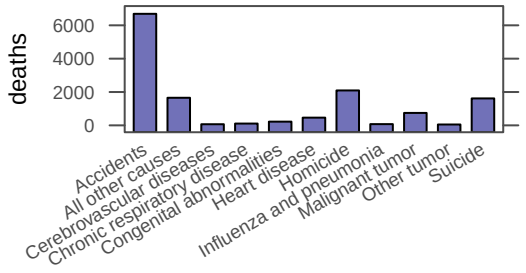
2.2 Working with Pre-Tabulated Data

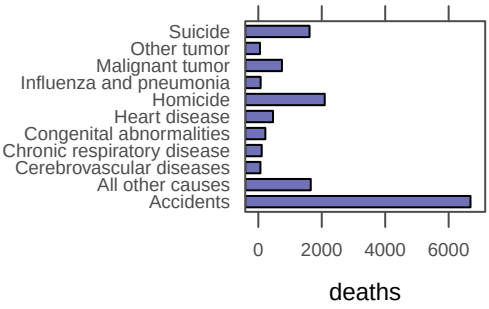
Sometimes data arrive pre-tabulated. We can use `barchart()` instead of `bargraph()` to graph pre-tabulated data.[2]

```
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
```

```
barchart(deaths ~ cause, data = TeenDeaths, scales = list(x = list(rot = 30)))
# ('scales' argument above is to rotate the x-labels so they don't overlap) you may find it
# useful, but learning it is optional.
barchart(cause ~ deaths, data = TeenDeaths)
```

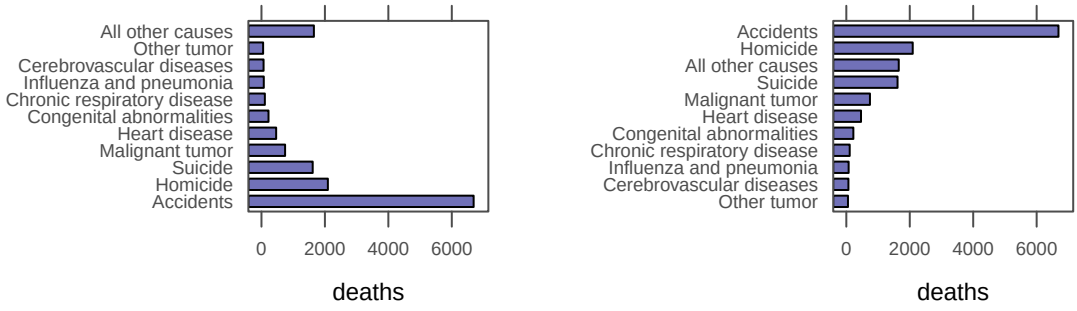




[2] `bargraph()` converts raw data into a summary table and then calls `barchart()` to do the plotting.

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 are two ways to have things appear in a different order. The first uses the order in which categories (to R "levels") appear in the `TeenDeaths` data frame. The second reorders based on the value of the `deaths` variable.

```
barchart(ordered(cause, levels = cause) ~ deaths, TeenDeaths)
barchart(reorder(cause, deaths) ~ deaths, TeenDeaths)
```



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 observational units, 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[3] 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.

[3] 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 most traditional way; should always work

## [1] 35.65342

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

## [1] 35.65342

with(HELPrct, mean(age))   # one more way this can be done

## [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
```

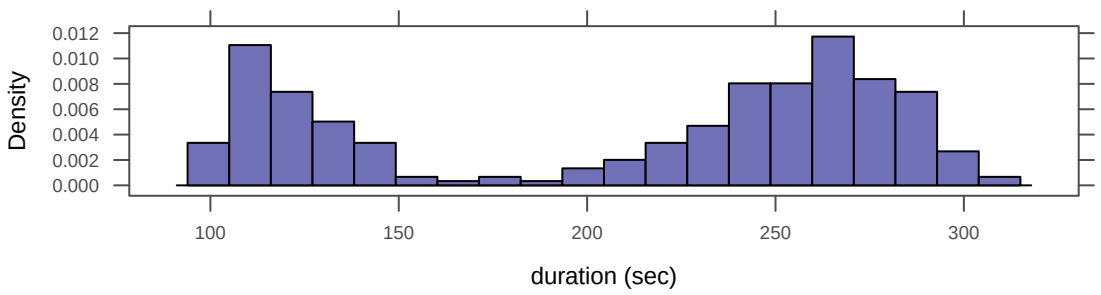
It is also possible to compute these statistics separately for each of several groups. The syntax is much like the the syntax we used when plotting. In fact, we have two choices for the formula: `y ~ x` or `~ x | z`.

```
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)

##   .group 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

histogram( ~ Duration, data=oldfaith,  n=20,
        main="Old Faithful Eruption Times", xlab="duration (sec)")
```



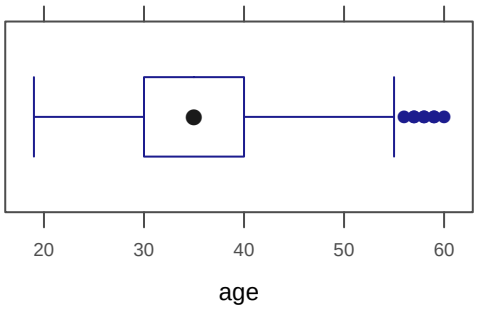
Notice that the mean and median do not represent typical eruption times very well. Nearly all eruptions are either quite a bit shorter or quite a bit longer. (This is especially true of the mean.)

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**:

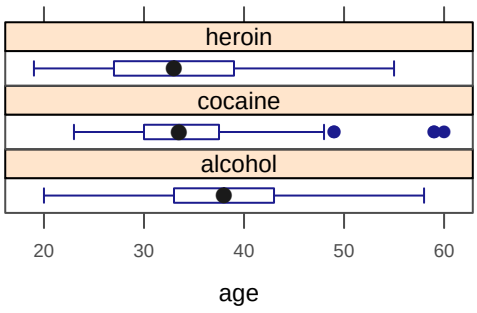
- Q_0, the minimum
- Q_1, the first quartile (25th percintile)
- Q_2, the median (50th percentile)
- Q_3, the third quartile (75th percentile)
- Q_4, the maximum

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



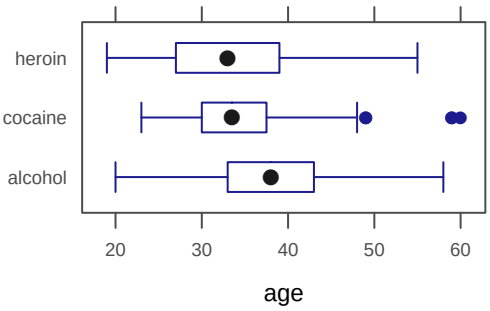
Boxplots provide a way of comparing multiple groups that is especially informative and visually effective. Here is one way to make boxplots of multiple groups (it should look familiar from what we know about histogram):

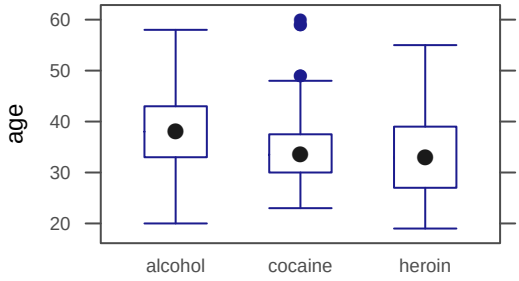
```
bwplot(~age | substance, data = HELPrct, layout = c(1, 3))
```



But `bwplot()` 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 `xyplot()`.

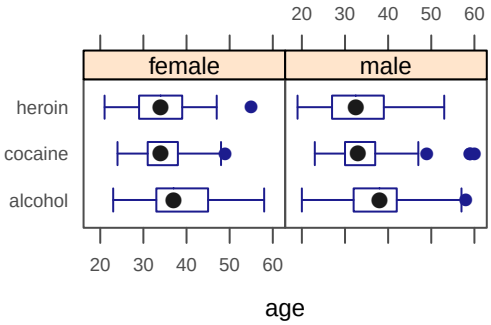
```
bwplot(substance ~ age, data = HELPrct)
bwplot(age ~ substance, data = HELPrct)
```





And we can combine this idea with conditioning:

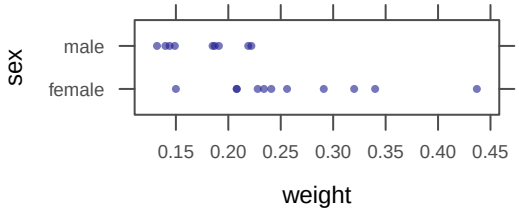
```
bwplot(substance ~ age | sex, data = HELPrct)
```

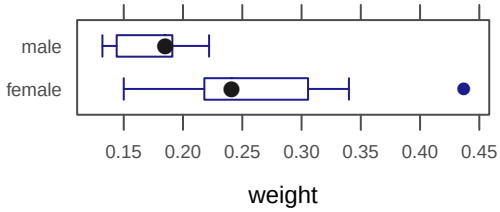


2.5.3 Small data sets

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. `xyplot()` 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.

```
xyplot(sex ~ weight, data = Mosquitoes, cex = 0.5, alpha = 0.6)
bwplot(sex ~ weight, data = Mosquitoes)
```

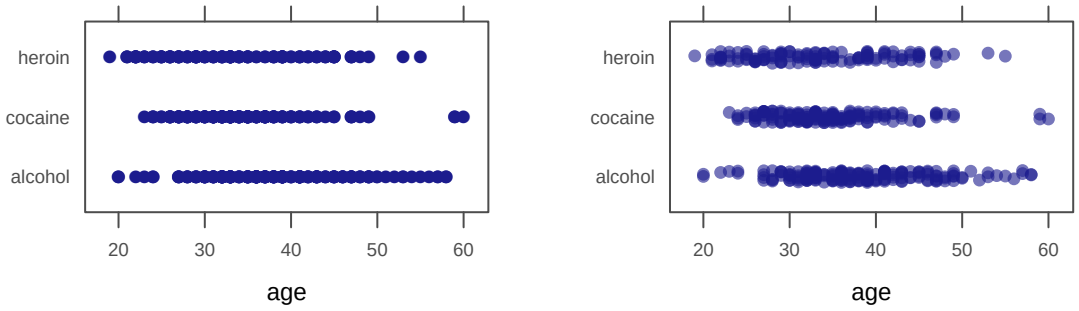




R actually provides a separate function, `stripplot()`, for this type of scatter plot (where one variable is categorical). Note the effect of the `jitter` argument – it moves each data point slightly up or down, to reduce

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.

```
stripplot(substance ~ age, data = HELPrct)
stripplot(substance ~ age, data = HELPrct, jitter = TRUE, alpha = 0.6)
```



2.6 Summarizing Categorical Variables

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

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

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

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

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

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

2.7 Relationships Between Two Variables

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

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 Probability

3.1 Key Definitions and Ideas

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

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

outcome A potential result of a random process.

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

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

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

trial One repetition of a random process.

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

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

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

Examples of random variables:

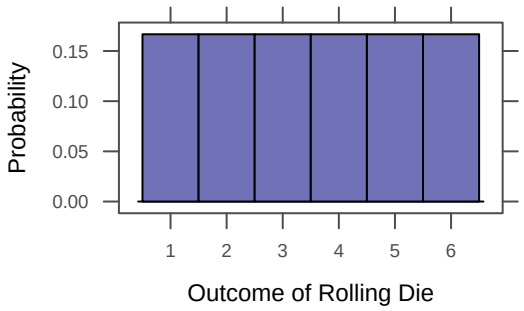
- Roll a die and record the number.
- Roll two dice and record the sum.
- Flip 100 coins and count the number of heads.
- Sample 1000 people and count how many approve of the job the president is doing.

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

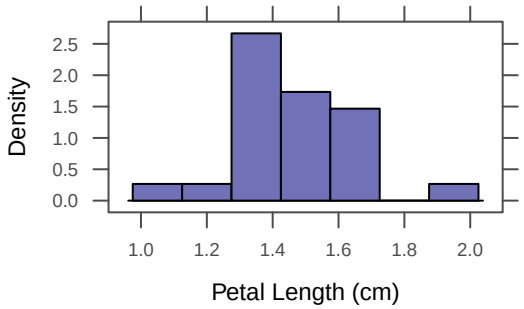
probability distribution The distribution of a random variable. (Remember that a distribution describes *what values?* and *with what freqency?*)

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 (see section 3.3 for details). For our example, where we rolled a die and recorded the value:



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



3.2 Calculating Probabilities Empirically

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

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

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

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

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

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

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

```
rflip(10)
```

```
##
## Flipping 10 coins [ Prob(Heads) = 0.5 ] ...
##
## T T H H H H H H H T
##
## Number of Heads: 7 [Proportion Heads: 0.7]
```

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     7     3  0.7
## 2 10     7     3  0.7
## 3 10     5     5  0.5
```

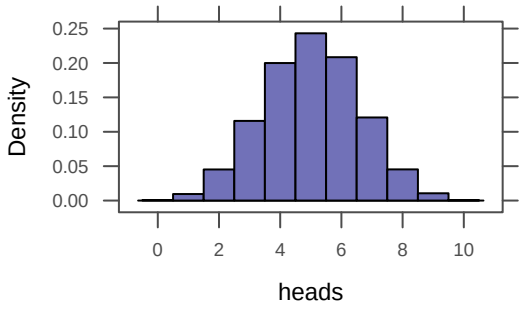
```
tosses <- do(10000) * rflip(10)
tally(~heads, data = tosses, format = "prop")
```

```
##
```

[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
## 0.0008 0.0095 0.0452 0.1158 0.1999 0.2431 0.2084 0.1208 0.0453 0.0106 0.0006
```

```
histogram(~heads, data = tosses, width = 1)
```



Based on this sample, we would estimate that $\mathrm{P}(X = 5) \approx 0.2431$.

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

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

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

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

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

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

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] 5 3 3 2 2 3
```

```
head(die2)
```

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

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

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


##
##  TRUE FALSE
##  1651  8349


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


##    target level:  TRUE;  other levels:  FALSE


##   TRUE
## 0.1651
```

So the probability appears to be approximately 0.1651.

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

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

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


##    target level:  TRUE;  other levels:  FALSE


##   TRUE
## 0.1428
```

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

```
tally(~s)


##
##    2    3    4    5    6    7    8    9   10   11   12
##  241  561  803 1089 1427 1648 1428 1103  854  593  253


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


##
##     2     3     4     5     6     7     8     9    10    11    12
##  2.41  5.61  8.03 10.89 14.27 16.48 14.28 11.03  8.54  5.93  2.53
```

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:

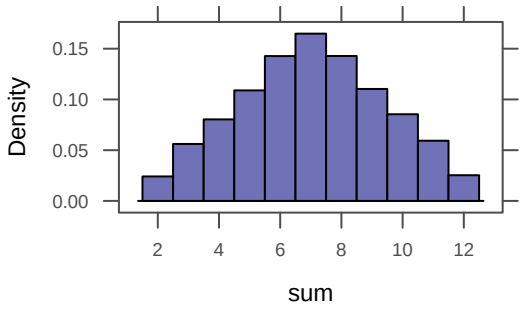
```
rolls <- data.frame(first = die1, second = die2, sum = die1 + die2)
head(rolls)
```

```
##   first second sum
## 1     5      5  10
## 2     3      4   7
## 3     3      4   7
## 4     2      4   6
## 5     2      5   7
## 6     3      6   9
```

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

```
##
##      2      3      4      5      6      7      8      9     10     11     12
## 0.0241 0.0561 0.0803 0.1089 0.1427 0.1648 0.1428 0.1103 0.0854 0.0593 0.0253
```

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



3.3 Calculating Probabilities Theoretically

The theoretical method combines

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

3.3.1 The Three Probability Axioms

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

1. Probability is between 0 and 1: $0 \leq \mathrm{P}(A) \leq 1$.
2. The probability of the sample space is 1: $\mathrm{P}(S) = 1$.

3. Additivity: If A and B are mutually exclusive, then $\mathrm{P}(A \text{ or } B) = \mathrm{P}(A) + \mathrm{P}(B)$.

Notation Notes

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

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

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

3.3.2 Other Probability Rules

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

The Addition Rule

If events A and B are mutually exclusive, then

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

More generally,

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

The Complement Rule

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

The Equally Likely Rule

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

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

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

Examples 3.3.1.

1. Coin Toss: $\mathrm{P}(\text{heads}) = \frac{1}{2}$ if heads and tails are equally likely.
2. Rolling a Die: $\mathrm{P}(\text{even}) = \frac{3}{6}$ if the die is fair (each of the six numbers equally likely to occur).
3. Sum of two Dice: the sum is a number between 2 and 12, but these numbers are NOT equally likely.

   There are 36 equally likely combinations of two dice:

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

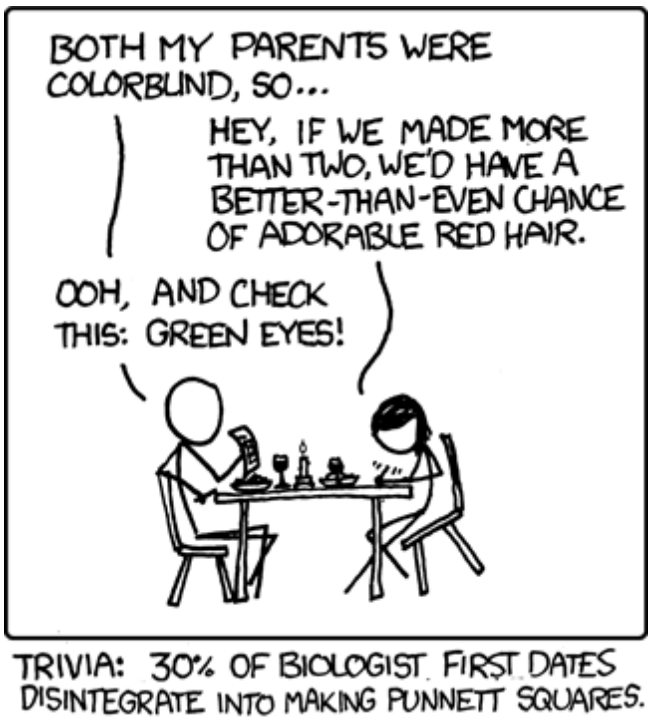
Let X be the sum of two dice.

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

4. Punnet Squares

	A	a
A	AA	Aa
a	Aa	aa

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



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

3.4 Conditional Probability

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

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

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

Figure 3.1: Illustrating the sample space for Example 3.4.1.

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

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

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

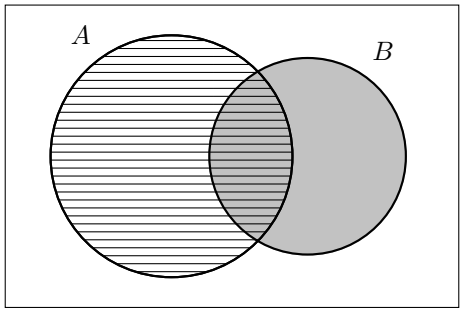


Figure 3.2: A Venn diagram illustrating the definition of conditional probability. $\mathrm{P}(A \mid B)$ is the ratio of the area of the football shaped region that is both shaded and striped ($A \cap B$) to the area of the shaded circle (B).

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

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

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

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

	Color	
	Red	Purple
Girls	7	9
Boys	10	8

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

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

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

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

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

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

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

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

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

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

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

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

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

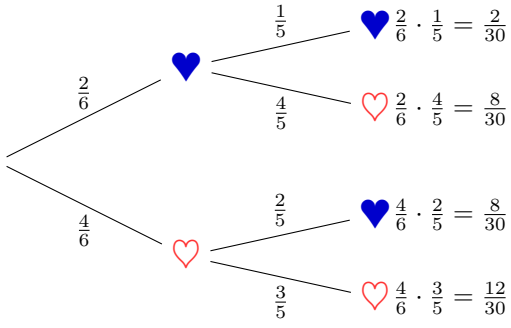
Example 3.4.5. Q. In a bowl are 4 red Valentine hearts and 2 blue Valentine hearts.

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

value of X	0	1	2
probability			

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

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



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

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

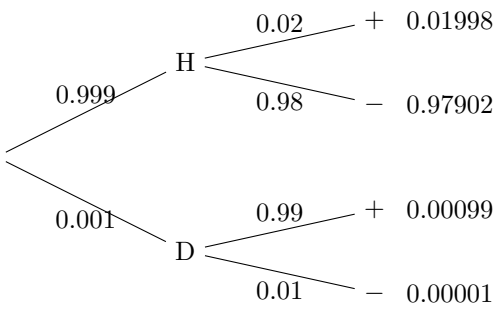
A. We begin by introducing some notation. Let D be the event that a person has the disease. Let H be the event that the person is healthy. Let $+$ be the event that the test comes back positive (meaning it indicates disease – probably a negative from the perspective of the person tested). Let $-$ be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

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

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

  $\mathrm{P}(- \mid H)$ is called the **specificity** of the test.
- $$\begin{aligned} \mathrm{P}(D \mid +) &= \frac{\mathrm{P}(D \cap +)}{\mathrm{P}(+)} \\ &= \frac{\mathrm{P}(D) \cdot \mathrm{P}(+ \mid D)}{\mathrm{P}(D \cap +) + \mathrm{P}(H \cap +)} \\ &= \frac{0.001 \cdot 0.99}{0.001 \cdot 0.99 + 0.999 \cdot 0.02} = 0.0472. \end{aligned}$$

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.

3.4.1 Independence

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.2

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

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

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

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

```
sample(1:52, 5)

## [1] 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)
```

```
##   result
## 1      0
## 2      1
## 3      1
```

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

What percentage of parts pass both inspections?

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

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

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

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

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

b) What are the other possible values of Y?

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

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

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

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

3.8 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?

3.9 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.

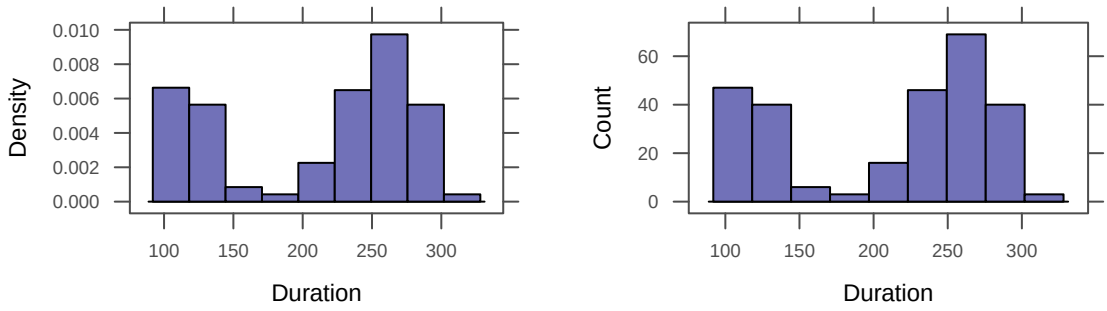
Excellent health statistics - smokers are less likely to die of age related illnesses.'

4 Densities

4.1 Density histograms, density plots, density functions

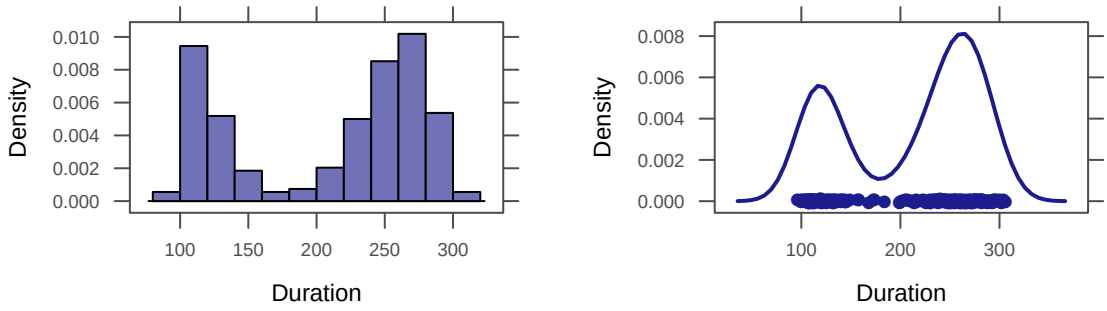
A histogram is a simple picture describing the "density" of data. Histogram bars are tall in regions where there is more data – i.e., where the data are more "dense".

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



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

```
require(alr3)
histogram(~Duration, data = oldfaith, width = 20, center = 110)
densityplot(~Duration, data = oldfaith)
```



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

Example 4.1.1. Q. Use the histogram of Old Faithful eruption times to estimate the proportion of eruptions that last between 100 and 120 seconds.

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

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


##
##      TRUE     FALSE
## 0.1888889 0.8111111
```

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 4.1.2. Let f be defined by

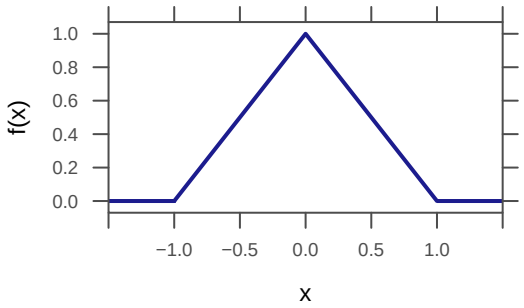
$$f(x) = \begin{cases} 1 - |x| & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

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

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

A. While we could set up integrals for these, it is easier to solve them using geometry.[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. $\mathrm{P}(X \leq 1) = \int_{-\infty}^{1} f(x)\,dx = \int_{-1}^{1} f(x)\,dx = 1$
2. $\mathrm{P}(X \leq \frac{1}{2}) = \int_{-\infty}^{1/2} f(x)\,dx = \int_{-1}^{1/2} f(x)\,dx = 1 - \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{7}{8}$
3. $\mathrm{P}(-\frac{1}{2} \leq X \leq \frac{1}{2}) = \int_{-1/2}^{1/2} f(x)\,dx = 1 - \frac{2}{8} = \frac{3}{4}$

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] 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.

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

integrate(f, -0.5, 0.5)

## 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, -0.5, 0.5)$value

## [1] 0.75
```

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

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

## [1] 1

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

## [1] 0.875

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

## [1] 0.75
```

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 4.1.3. 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
```

4.2 Working with Probability Density Funcitons

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

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

This function can also be used to compute probabilities, since

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

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

4.2.1 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 4.2.1. Q. The kernel of a random variable is given by

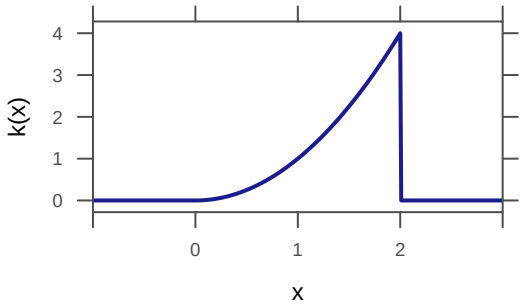
$$k(x) = x^2\, [\![x \in [0,2]]\!]\,.$$

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.

4.2.2 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 4.2.2.** 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 4.2.3. Q. Compute the mean of our triangle distribution from Example 4.1.2.

A. We simply compute the integral from the definition.

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

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

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
```

[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.

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

Example 4.2.4. Q. Compute the variance of the triangle random variable from the Example 4.1.2.

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

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

Example 4.2.5. 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 4.2.1.

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
```

[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.

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


## [1] 0.15
```

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

4.2.4 Quantiles

Quantiles solve equations of the form

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

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

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

Example 4.2.6. Q. What is the 25th percentile of the triangle distribution in Example 4.1.2?

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

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

We can do this by working out the integral involved:

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

So by the quadratic formula, $x = \frac{1}{2}\sqrt{2} - 1 = -0.2928932$.

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 $x = -1 + y$.

4.3 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.

4.3.1 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

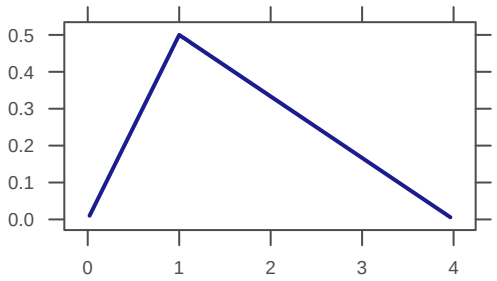
Function	What it does
`dtriangle(x,a,b,c)`	Computes value of the pdf at `x`
`ptriangle(q,a,b,c)`	Computes value of the cdf at `x`, i.e., $\mathrm{P}(X \le \mathtt{q})$
`qtriangle(p,a,b,c)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \le \mathtt{q}) = \mathtt{p}$
`rtriangle(n,a,b,c)`	Randomly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

Example 4.3.1. 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 `plotDist()` 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.

```
require(triangle)  # a package that knows about triangle distributions

## Loading required package:  triangle

plotDist("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)
```

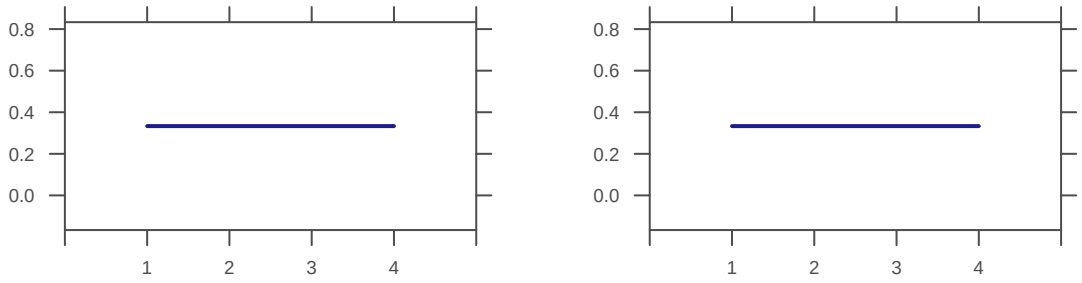
```
## 1.666667 with absolute error < 1.9e-14
```

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

```
plotDist("unif", params = list(min = 1, max = 4), xlim = c(0, 5))  # using parameter names
plotDist("unif", params = list(1, 4), xlim = c(0, 5))  # without 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 4.3.2. 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(0.8, 1, 4)  # 80th percentile
```

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

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

4.3.3 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 4.3.3. 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

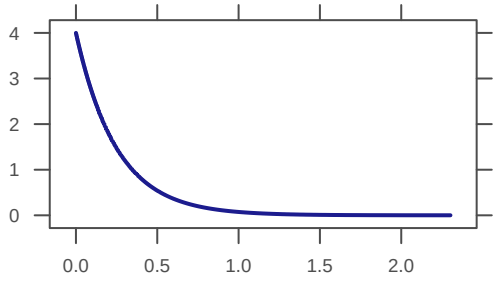
$$k(x;\lambda) = e^{-\lambda x}\,[\![x \geq 0]\!]$$

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

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



4.3.4 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)	$k(x) = x^{\alpha-1}e^{-\lambda x^{\alpha}}\ [\![x \geq 0]\!]$

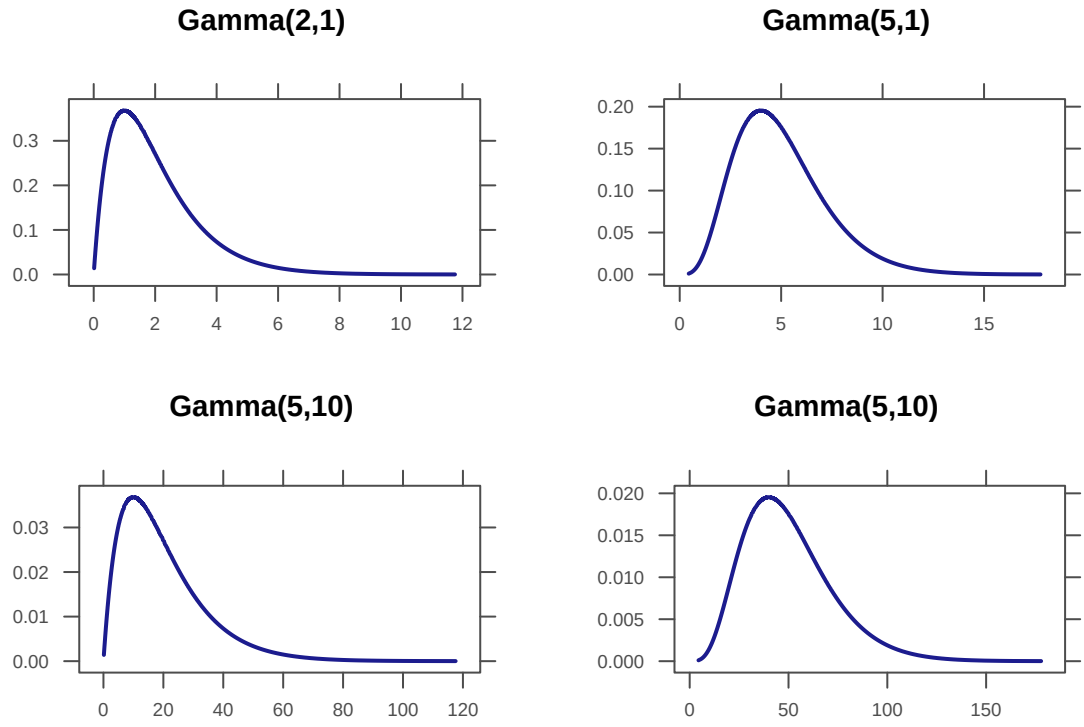
Both families of distributions are supported on the interval $[0, \infty)$.) For the most part, we won't use these formulas in calculations, preferring to let R do the work for us. However, notice that each of these distributions has a pdf that allows for relatively simple integration. For the Gamma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

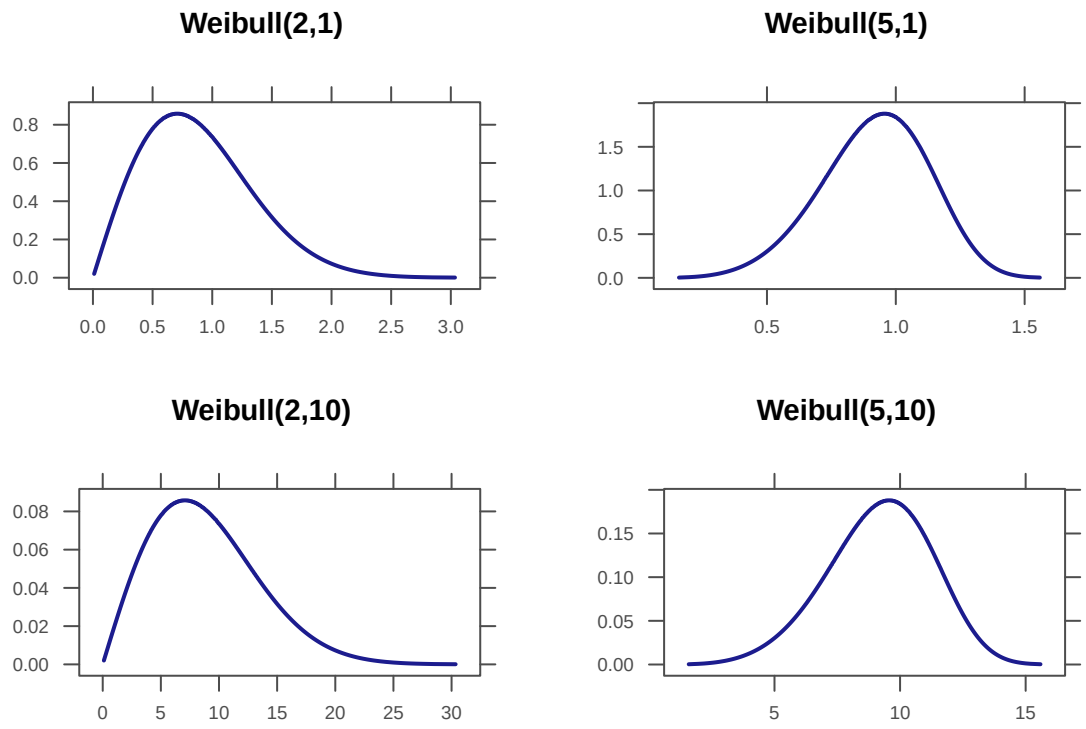
Function	What it does
`dgamma(x, shape, rate, scale=1/rate)`	Computes value of the pdf at `x`
`pgamma(q, shape, rate, scale=1/rate)`	Computes value of the cdf at `x`, i.e., $\mathrm{P}(X \leq \mathtt{q})$
`qgamma(p, shape, rate, scale=1/rate)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \leq \mathtt{q}) = \mathtt{p}$
`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., $\mathrm{P}(X \leq \mathtt{q})$
`qweibull(p, shape, scale)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \leq \mathtt{q}) = \mathtt{p}$
`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.

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



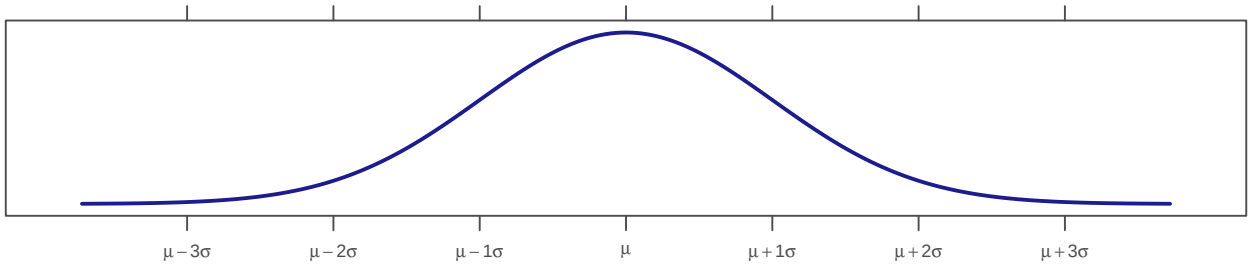

```
plotDist("weibull", params = list(shape = 2, scale = 1), main = "Weibull(2,1)")
plotDist("weibull", params = list(shape = 5, scale = 1), main = "Weibull(5,1)")
plotDist("weibull", params = list(shape = 2, scale = 10), main = "Weibull(2,10)")
plotDist("weibull", params = list(shape = 5, scale = 10),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



4.3.5 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
$\mathsf{Norm}(\mu, \sigma)$	$f(x) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$



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

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

standard normal distribution. That is, If $Z \sim \mathsf{Norm}(0,1)$ 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 4.3.4. 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:

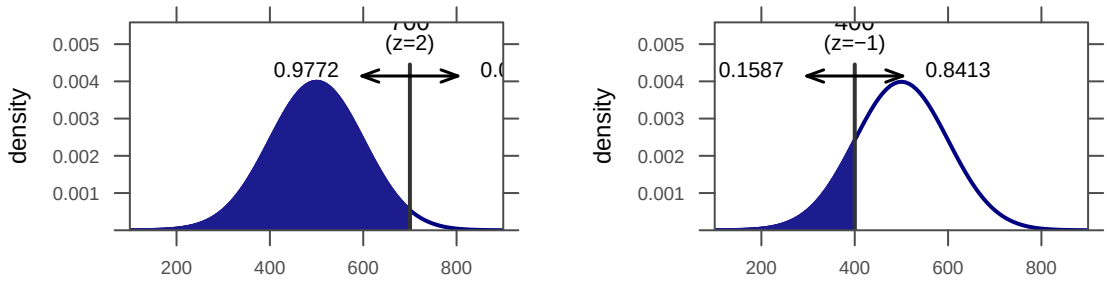
```
pnorm(600, 500, 100) - pnorm(400, 500, 100)
```

```
## [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) = 0.0228
##
## 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 4.3.5. We can use `qnorm()` to compute percentiles. For example, let's calculate the 75th percentile for SAT distributions.

```
qnorm(0.75, 500, 100)


## [1] 567.449
```

4.3.6 Beta Distributions

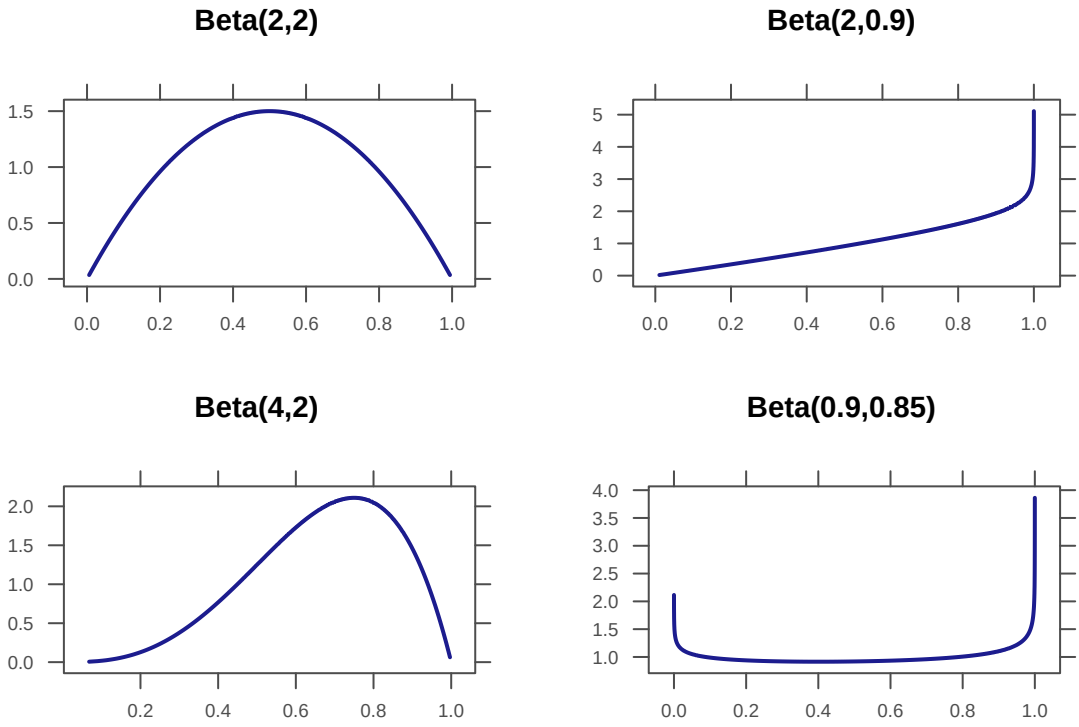
The Beta distributions have support on the interval $(0, 1)$, so they can provide a model for proportions or other quantities that are bounded between 0 and 1.[4] The Beta distributions have two parameters, imaginatively called `shape1` and `shape2`. The kernel of the Beta distributions is a product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

The two shape parameters provide a wide variety of shapes.

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



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., $\mathrm{P}(X \leq \mathtt{q})$
`qbeta(p,shape1,shape2)`	Computes quantiles, that is a value q so that $\mathrm{P}(X \leq \mathtt{q}) = \mathtt{p}$
`rbeta(n,shape1,shape2)`	Randomly samples `n` values from a Beta distribution.

[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.

4.4 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 4.4.1. 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.

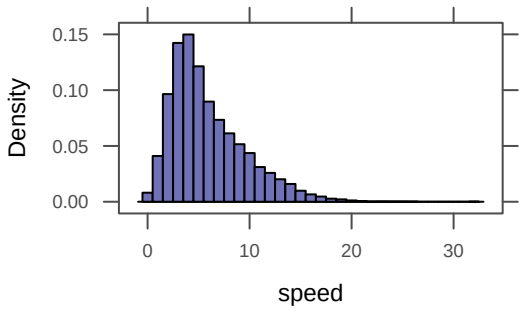
```
Wind <- read.csv("http://www.calvin.edu/~rpruim/data/stob/TwinfallsWind.csv")
head(Wind, 2)
```

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

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

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

```
histogram(~speed, data = Wind, width = 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)
```

```
##
##  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.

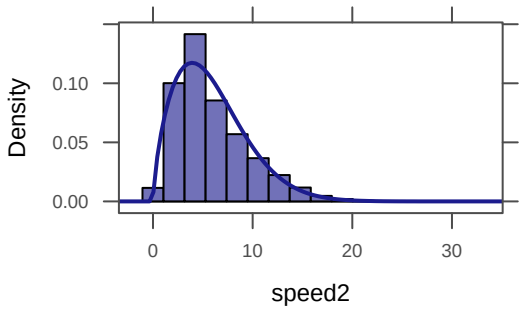
```
Wind <- transform(Wind, speed2 = ifelse(speed > 0, speed, 0.0025))
fitdistr(Wind$speed2, "weibull")
```

```
##       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.

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

```
histogram(~speed2, data = Wind, fit = "weibull")
```

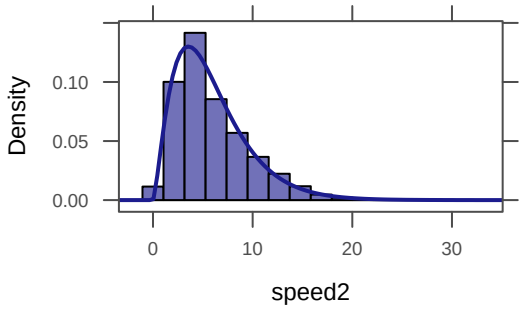


Example 4.4.2. As an alternative, we could fit a Gamma distribution to the wind speed data.

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

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

```
histogram(~speed2, data = Wind, fit = "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 4.4.3. 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")
```

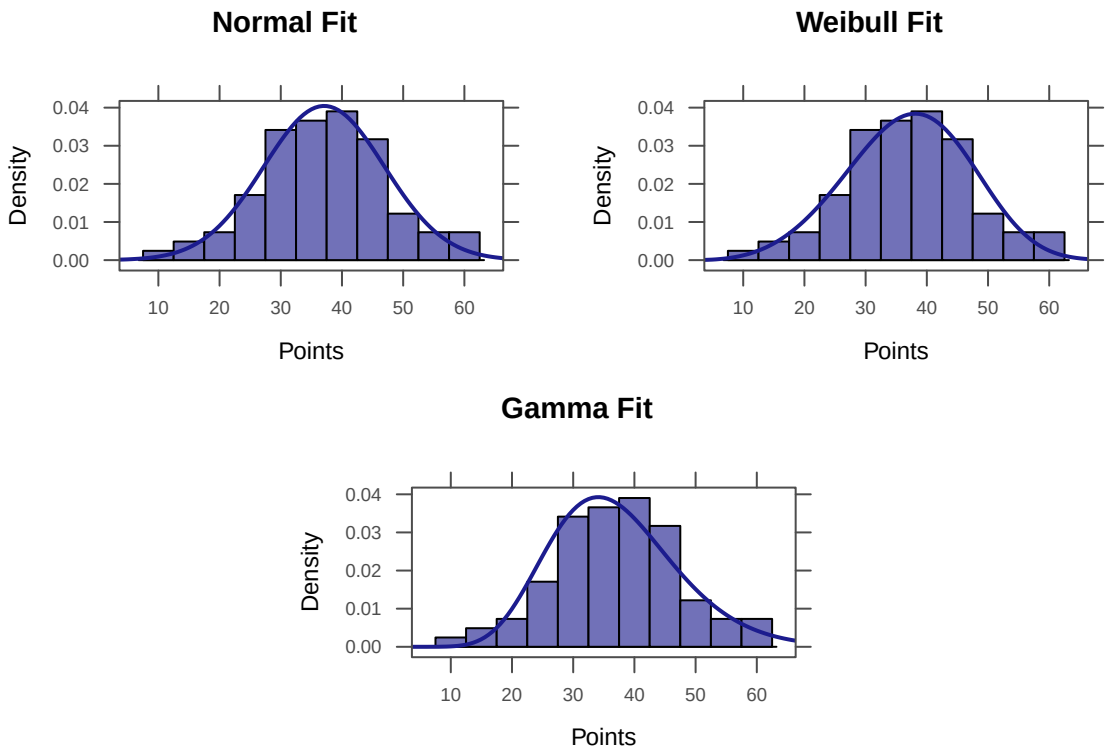
```
##      shape        scale
##    4.1227692   40.7746012
##  ( 0.3454908) ( 1.1516943)
```

```
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.

```
histogram(~Points, Jordan8687, fit = "normal", width = 5, main = "Normal Fit")
histogram(~Points, Jordan8687, fit = "weibull", width = 5, main = "Weibull Fit")
histogram(~Points, Jordan8687, fit = "gamma", width = 5, main = "Gamma Fit")
```

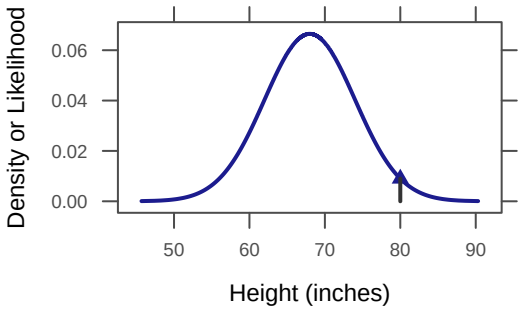


The three fits are similar, but not identical.

4.4.1 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 4.4.4. 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

$$\left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^7 = 9.765625 \times 10^{-4} \, .$$

Of these, the largest likelihood is for the case that P(roll 1 or 2) $= \frac{1}{3}$, i.e., for the standard, six-sided die. Our data would be more likely to occur with that die than with either of the other two – it is the maximum likelihood die.

In general, maximum likelihood calculations are harder because instead of having only 3 choices, there will be infinitely many choices, and instead of having only one parameter, there may be multiple parameters. So

techniques from (multi-variable) calculus or numerical approximation methods are often used to maximize the likelihood function. The `fitdistr()` function uses pre-derived formulas for some distributions and numerical approximation methods for others. In some cases, you will get warning messages about attempts to apply a function to values that don't make sense (trying to take logs or square roots of negative numbers, zero in the denominator, etc.) as the numerical approximation algorithm explores options in an attempt to find the best fit. The help documenation for `fitdistr()` explains which distributions it can handle and what method is used for each.

4.4.2 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 formalas for the means and variances in terms of the parameters involved. These are listed in Table 4.1

Example 4.4.5. Let's return to the wind speeds in Example 4.4.1. 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 4.1, we see that to calculate the mean we need the value of $\Gamma(1 + \frac{1}{2}) = \Gamma(1.5) = \sqrt{\pi}/2$.

```
gamma(1.5)
```

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

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

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

From Table 4.1 we see that the mean of a $\mathsf{Rayleigh}(\beta)$-distribution is

$$\mathrm{E}(X) = \beta \frac{\sqrt{\pi}}{2}$$

Now we can choose our estimate $\hat{\beta}$ for β so that

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

That is,

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

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

```
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 Rayleigh(6.69) = Weibull$(2, 6.69)$

Although the Rayleigh distributions are not as flexible as the Weibull or Gamma distributions, and although maximum likelihood is generally preferred over the method of moments, the method of moments fit of a Rayliegh distribution does have one advantage: it can be computed even if all you know is the mean of some sample data. Sometimes, that is all you can easily get your hands on (because the people who collected the raw data only report numerical summaries). You can find average wind speeds of for many locations online, for example here: `http://www.wrcc.dri.edu/htmlfiles/westwind.final.html`

Example 4.4.6. 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
```

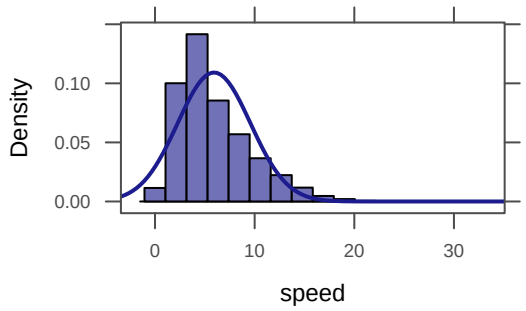
So the method of moments suggests a Norm$(5.93, 3.65)$ distribution. In this case, the method of moments and maximum likelihood methods give the same results.

```
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.

```
histogram(~speed, data = Wind, fit = "normal")
```



It is important to remember that the best fit using a poor choice for the family of distriubtions might not be a useful fit. The choice of distributions is made based on a combination of theoretical considerations, experience from previous data sets, and the quality of the fit for the data set at hand.

4.5 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 `qqmath()`.

4.5.1 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 4.5.1. 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.ids
```

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 4.1: Some common continuous distributions. Standard names for parameters that appear in several distributions include `rate` (λ), `shape` (α), and `scale` (β). In the normal distributions, μ and σ are called `mean` and `sd` in R, and in the uniform distirbutions, a and b are called `min` and `max`. The function $\Gamma(x)$ that appears in the formulas for the Weibull and Beta distributions is a kind of continuous extrapolation from the factorial function. The `gamma()` function will calculate these values.

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

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

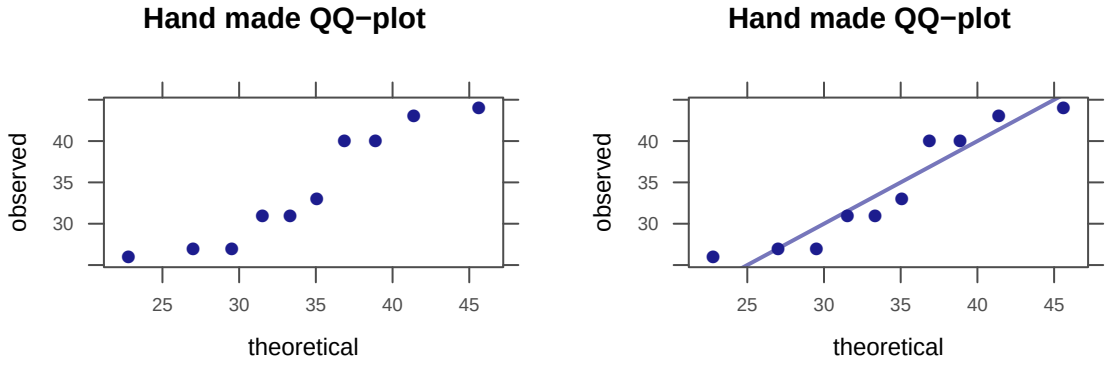
```
##  [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
```

```
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
```

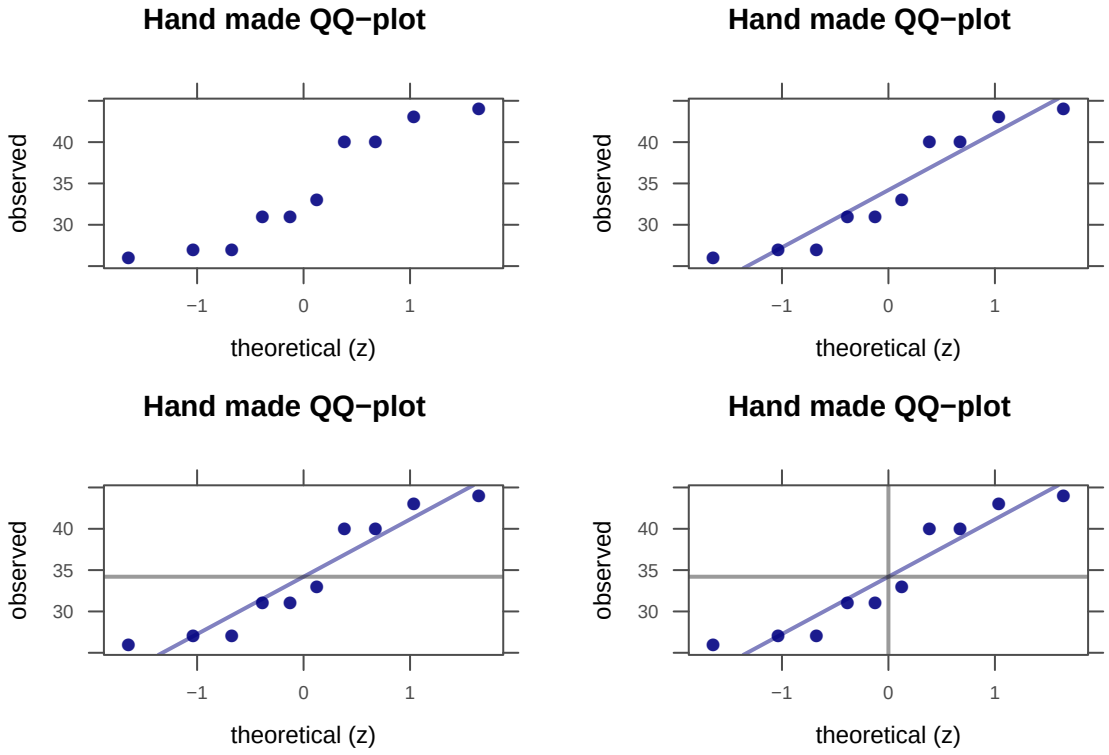
If the observed data matched the theoretical quantiles perfectly, a scatter plot would place all the points on the line with slope 1 passing through the origin.

```
xyplot(observed ~ theoretical, data = QQData, main = "Hand made QQ-plot")
plotFun(x ~ x, add = TRUE, alpha = 0.6)
```



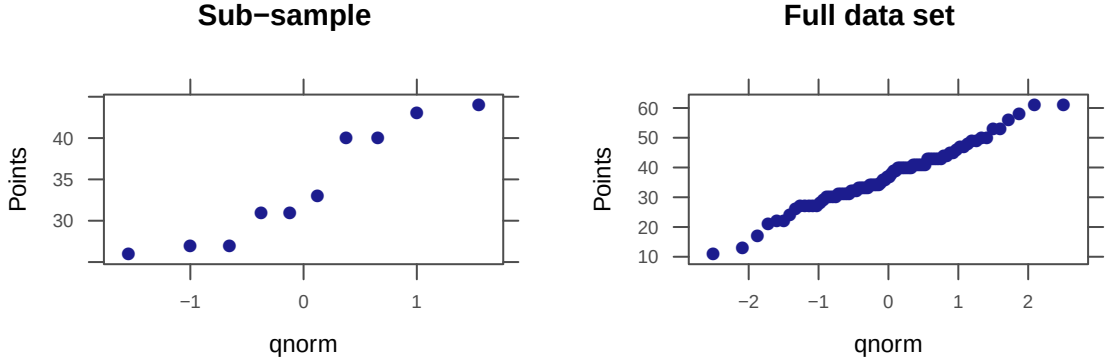
Even better, we don't need to know the mean and standard deviation in advance, because all normal distributions are linear transformations of the Norm$(0,1)$-distribution. So our standard practice will be to compare our data to the Norm$(0,1)$-distribution. If $X \sim \text{Norm}(\mu, \sigma)$, then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

```
theoretical2 <- qnorm(probs, mean = 0, sd = 1)  # theoretical quantiles from Norm(0,1)
QQData2 <- data.frame(observed = observed, theoretical = theoretical2)
xyplot(observed ~ theoretical, data = QQData2, main = "Hand made QQ-plot", xlab = "theoretical (z)")
ladd(panel.abline(mean(SmallJordan$Points), sd(SmallJordan$Points), alpha = 0.5, col = "navy"))
ladd(panel.abline(h = mean(SmallJordan$Points), alpha = 0.5))
ladd(panel.abline(v = 0, alpha = 0.5))
```



This whole process is automated by the `qqmath()` function.

```
qqmath(~Points, SmallJordan, main = "Sub-sample")
qqmath(~Points, Jordan8687, main = "Full data set")
```

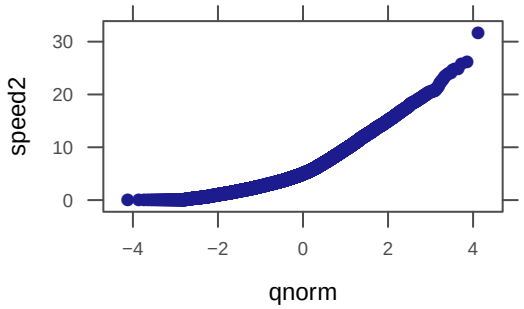


4.5.2 Other distributions

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 4.5.2. 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 `qqmath()` how to calculate the quantiles.

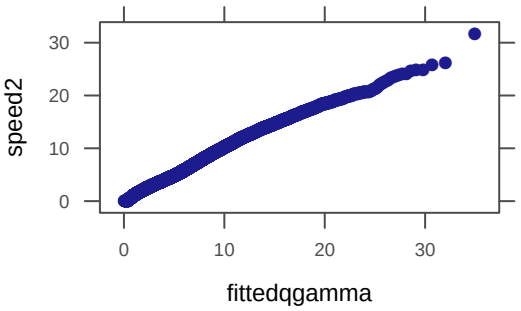
```
qqmath(~speed2, data = Wind)  # normal-quantile plot; normal is not a good model
```

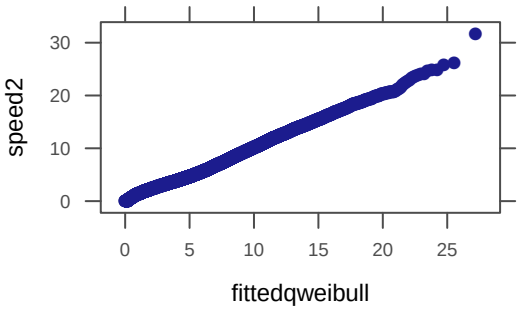


The normal model does not fit well, but both Gamma and Weibull are reasonable models:

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

##       shape         rate
##   2.495582854   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 )
qqmath( ~speed2, data=Wind, distribution=fittedqgamma )
qqmath( ~speed2, data=Wind, distribution=fittedqweibull )
```

[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.


speed2
30
20
10
0
0
10
20
30
fittedqgamma


speed2
30
20
10
0
0
5
10
15
20
25
fittedqweibull

Exercises

4.1 Let $f(x) = 5/4 - x^3$ on $[0, 1]$.

a) Show that f is a pdf.

b) Calculate $\mathrm{P}(X \leq \frac{1}{2})$.

c) Calculate $\mathrm{P}(X \geq \frac{1}{2})$.

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

4.2 Repeat parts (2) – (4) of Example 4.3.1 using geometry rather than R.

4.3 Let $k(x) = (1 - x^2) \cdot [\![x \in [-1, 1]]\!] = \begin{cases} 1 - x^2 & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$ be the kernel of a continuous distribution.

a) Determine the pdf for this distribution.

b) Compute the mean and variance for this distribution

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

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

4.6

a) Let $X \sim \mathsf{Exp}(4)$. Use R to compute $\mathrm{E}(X)$.

b) Let $X \sim \mathsf{Exp}(10)$. Use R to compute $\mathrm{E}(X)$.

c) Let $X \sim \mathsf{Exp}(1/5)$. Use R to compute $\mathrm{E}(X)$.

d) What pattern do you notice. Explain in terms of the definition of the exponential distribution why this makes sense.

4.7 Use R to plot the pdf and compute the mean and variance of each of the following distributions.

a) $\mathsf{Beta}(2, 3)$

b) $\mathsf{Beta}(20, 30)$

c) $\mathsf{Gamma}(\texttt{shape} = 2, \texttt{scale} = 3)$

d) Weibull(`shape` $= 2$, `scale` $= 3$)

4.8 For each of the following distributions, determine the proportion of the distribution that lies between 0.5 and 1.

a) Exp(`rate` $= 2$)

b) Beta(`shape1` $= 3$, `shape2` $= 2$)

c) Norm(`mean` $= 1$, `sd=2`)

d) Weibull(`shape` $= 2$, `scale=1/2`)

e) Gamma(`shape` $= 2$, `scale=1/2`)

4.9

a) Using Table 4.1 and the method of moments, fit an exponential distribution to the Twin Falls wind speed data. What is the estimated value of the rate parameter?

b) Now use `fitdistr()` to fit an exponential distribution using maximum likelihood.

c) How do the two estimates for the rate parameter compare?

d) How well does an exponential distribution fit this data?

4.10 A Gamma distribution can also be fit using the method of moments. Because there are two parameters (shape and rate or shape and scale), you will need to solve a system of two equations with two unknowns.

a) Using Table 4.1 and the method of moments, fit a Gamma distribution to the Twin Falls wind speed data. What are the estimated values of the shape and rate parameters?

b) How do the method of moments estimates for the parameters compare to the maximum likelihood estimates from `fitdistr()`?

4.11 Sam has found some information about wind speed at a location he is interested in online. Unfortunately, the web site only provides the mean and standard deviation of wind speed.

mean:	10.2 mph
standard deviation:	5.1 mph

a) Use this information and the method of moments to estimate the shape and rate parameters of a Gamma distribution.

b) In principal, we could do the same for a Weibull distribution, but the formulas aren't as easy to work with. Fit a Rayliegh distribution instead (i.e., a Weibull distribution with shape parameter equal to 2).

4.12 In 1964, a study was undertaken to see if IQ at 3 years of age is associated with amount of crying at newborn age. In the study, 38 newborns were made to cry after being tapped on the foot, and the number of distinct cry vocalizations within 20 seconds was counted. The subjects were followed up at 3 years of age and their IQs were measured. You can load this data using

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

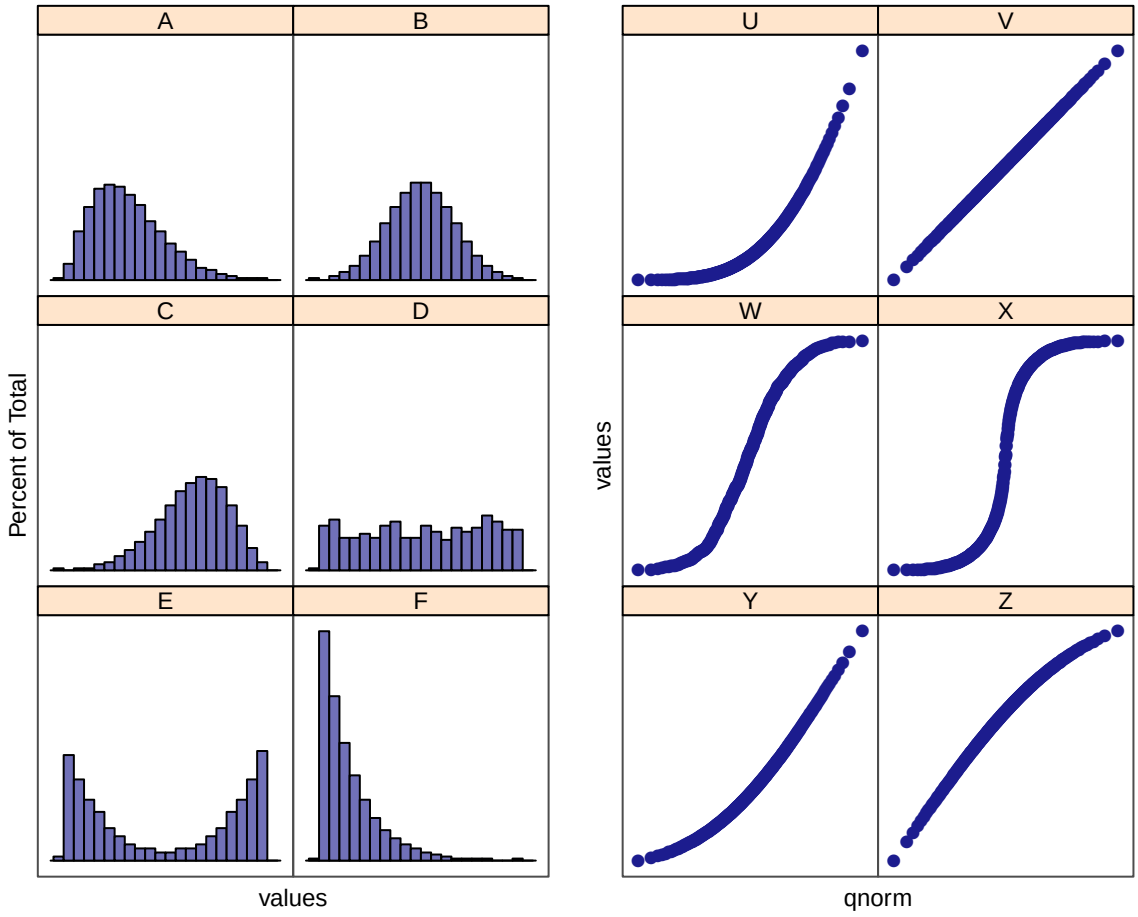
```
##   cry.count  IQ
## 1        10  87
## 2        20  90
## 3        17  94
## 4        12  94
## 5        12  97
## 6        15 100
```

The `cry.count` variable records the number of distinct cry vocalizations within 20 seconds. Choose a family of distributions to fit to this data and do the fit using `fitdistr()`. Also include a plot showing a histogram and your fitted density curve.

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

Comment on the plots.

4.14 Match the normal-quantile plots to the histograms.



4.15 Show that $\text{Var}(X) = \text{E}(X^2) - \text{E}(X)^2$ by showing that

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

whenever f is a pdf and all the integrals involved converge.

4.16 The heights of 18–22 year olds in the US follow approximately normal distributions within each sex. Estimated means and standard deviations appear in the table below.

	mean	standard deviation
women	64.3 in	2.6 in
men	70 in	2.8 in

Answer the following questions without using a computer or calculator (except for basic arithmetic).

a) If a woman is 68 inches tall, what is her z-score?

b) If a man is 74 inches tall, what is his z-score?

c) What is more unusual, a woman who is at least 68 inches tall or a man who is at least 74 inches tall?

d) Big Joe has decided to open a club for tall people. To join his club, you must be in the tallest 2.5% of people of your sex. How tall must a woman be to join Big Joe's club?

e) How tall must a man be to join Big Joe's club?

4.17 Use the 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?

Bibliography

Index