

An R Companion to Introduction to Statistical Investigations

Preliminary Edition

Randall Pruim and Lana Park

September 1, 2014

Contents

0 Preliminaries	5
0.0 Getting Started with R and RStudio	5
0.1 Introduction to the Six-Step Method	7
0.2 Exploring Data	11
0.3 Exploring random Processes	15
0.4 Other Visualizations	17
1 Significance: How strong is the evidence?	21
1.1 Introduction to Chance Models	21
1.2 Measuring the Strength of Evidence	24
1.3 Alternative Measure of Strength of Evidence	27
1.4 What Impacts Strength of Evidence?	32
1.5 Inference on a single proportion: Theory-based approach	39
2 Generalization: How Broadly Do the Results Apply?	51
2.1 Sampling from a Finite Population	51
2.2 Inference for a Single Quantitative Variable	59
2.3 Errors and Significance	71
3 Estimation: How Large is the Effect?	75
3.1 Statistical Inference - Confidence Intervals	75
3.2 2SD and Theory-Based Confidence Intervals for a Single Proportion	83

3.3	2SD and Theory-Based Confidence Intervals for a Single Mean	91
3.4	Factors That Affect the Width of a Confidence Interval	94
3.5	Cautions When Conducting Inference	99
4	Causation: Can We Say What Caused the Effect?	107
4.1	Association and Confounding	107
4.2	Observational studies versus experiments	108
5	Comparing Two Proportions	109
5.1	Comparing Two Groups: Categorical Response	109
5.2	Comparing Two Properties: Simulation-Based Approach	113
5.3	Comparing Two Proportions: Theory-Based Approach	126
6	Comparing Two Means	137
6.1	Comparing Two Groups: Quantitative Response	137
6.2	Comparing Two Means: Simulation-Based Approach	140
6.3	Comparing Two Means: Theory-Based Approach	146
7	Paired Data: One Quantitative Variable	153
7.1	Paired Designs	153
7.2	Simulation-Based Approach for Analyzing Paired Data	153
7.3	Theory-Based Approach to Analyzing Data from Paired Samples	160

0

Preliminaries

0.0 Getting Started with R and RStudio

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

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

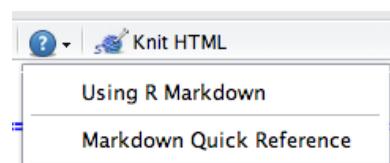
- `mosaic` (a package from Project MOSAIC)
- `Tintle1` (data sets)

RStudio provides several ways to create documents that include text, R code, R output, graphics, even mathematical notation all in one document. The simplest of these is R Markdown.

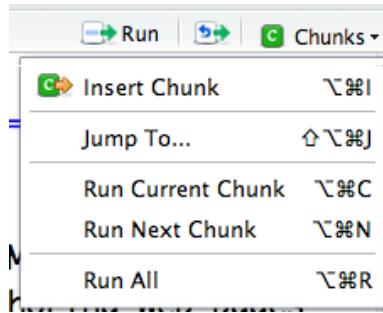
To create a new R Markdown document, go to “File”, “New”, then “R Markdown.”

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.



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.

There are options to do things like (a) run R code without displaying it, (b) run R code without displaying the output, (c) controlling size of plots, etc., etc. But for starting out, this is really all you need to know.

R Markdown files are self-contained, meaning they do not have access to things you have done in your console. (This is good, else your document would change based on things not in the file.) This means that you must explicitly load data, and require packages *in the R Markdown file* in order to use them. For this text, this means that most of your R Markdown files will have a chunk near the beginning that includes

```
require(mosaic) # load the mosaic package
```

Functions in R use the following syntax:

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

function-syntax

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.

Most of what we will do in the subsequent chapters makes use of a single R template:

$$\boxed{\quad}(\boxed{\quad} \sim \boxed{\quad}, \text{data} = \boxed{\quad})$$

It is useful if we name the slots in this template:

$$\boxed{\text{goal}}(\boxed{y} \sim \boxed{x}, \text{data} = \boxed{\text{mydata}})$$

However, there are some variations on this template:

```
## Simpler version
goal(~x, data = mydata)
## Fancier version:
goal(y ~ x | z, data = mydata)
## Unified version:
goal(formula, data = mydata)
```

To use the template, you just need to know what goes in each slot. This can be determined by asking yourself two questions:

1. What do you want R to do?
 - this determines what function to use (goal).
2. What must R know to do that?
 - this determines the inputs to the function
 - for describing data, must identify *which data frame* and *which variable(s)*.

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

Additional R functionality will be introduced as we go along. The `mosaic` package includes several vignettes with additional information about using the package and using R.

0.1 Introduction to the Six-Step Method

Example P.1: Organ Donations

Now that we've explained a few basics for using R, let's take a look at a data set.

Data sets in R are usually stored as **data frames** in a rectangular arrangement with rows corresponding to **observational units** and columns corresponding to **variables**. A number of data sets are built into R and its packages. The package for our text is `Tintle1` which comes with a number of data sets.

```
require(Tintle1) # tell R to use the package for our textbook
data(OrganDonor) # load the OrganDonor dataset
```

If you want a list of all data sets available to you in loaded packages, use `data()` without any arguments. If you want to view the entire data set, just typing the name will show the details in the console.

```
data() # list all datasets available in loaded packages
OrganDonor # show entire dataset in console
```

For large data sets, it may be more practical to look at different types of summaries or subsets of the data.

```
head(OrganDonor)      # first six cases of the dataset

  default choice
1  opt-in   donor
2  opt-in   donor
3  opt-in   donor
4  opt-in   donor
5  opt-in   donor
6  opt-in   donor

summary(OrganDonor)    # summary of each variable

  default      choice
opt-in :55    donor:108
opt-out:50   not  : 53
neutral:56
```

```

str(OrganDonor)      # structure of the dataset

'data.frame': 161 obs. of  2 variables:
 $ default: Factor w/ 3 levels "opt-in","opt-out",...: 1 1 1 1 1 1 1 1 1 ...
 $ choice : Factor w/ 2 levels "donor","not": 1 1 1 1 1 1 1 1 1 ...

dim(OrganDonor)      # number of rows and columns

[1] 161   2

nrow(OrganDonor)     # number of rows

[1] 161

ncol(OrganDonor)     # number of columns

[1] 2

```

Now that we have a general sense of how the data is structured, we can take a more detailed look by using the R template. Let's say we want a count of observational units of each variable. We can tally the number by using the `tally()` function.

```
tally(~choice, data = OrganDonor)
```

```

donor    not
 108      53

```

```
tally(~default, data = OrganDonor)
```

```

opt-in  opt-out  neutral
    55       50       56

```

This didn't really show us any more information than the `summary()` from above so instead, let's tally the variables together.

```
tally(~choice + default, data = OrganDonor)
```

```

            default
choice  opt-in  opt-out  neutral
  donor    23      41      44
  not     32       9      12

```

```
tally(~choice + default, data = OrganDonor, margins = TRUE)
```

```

            default
choice  opt-in  opt-out  neutral Total
  donor    23      41      44    108
  not     32       9      12     53
  Total    55      50      56    161

```

Notice that the default for `tally()` was to exclude the total counts of each row and column. You could have used either tab completion or search `tally()` in the help section to find `margins` and set `margins=TRUE`. There will be many instances where you will need to change the default settings of a function.

Moreover, we can change the organization of the variables for a slightly different output:

```
tally(choice ~ default, data = OrganDonor)

      default
choice  opt-in opt-out neutral
donor  0.4182  0.8200  0.7857
not    0.5818  0.1800  0.2143

tally(choice ~ default, data = OrganDonor, format = "percent")

      default
choice  opt-in opt-out neutral
donor  41.82   82.00   78.57
not    58.18   18.00   21.43
```

This may be a little confusing now (proportions will be covered in chapter 2) but let's focus more on the the changed organization of the variables in the `tally()` function. This version of tallying calculated the proportions (and percentages) of participants who agreed and did not agree to become organ donors (`choice`) in each of the groups opt-in, opt-out, and neutral (`default`).

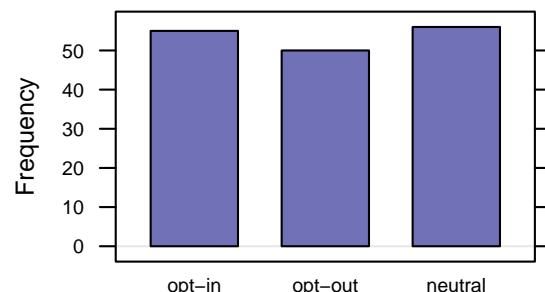
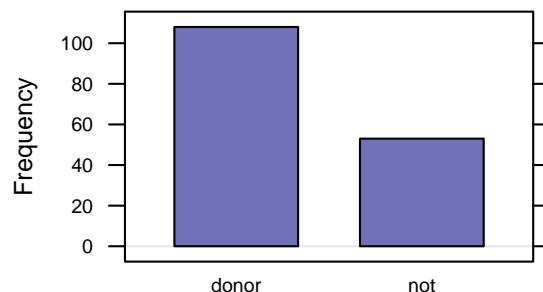
R also has many tools to visualize data. The general syntax for making a graph of one variable in a data frame is

```
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.

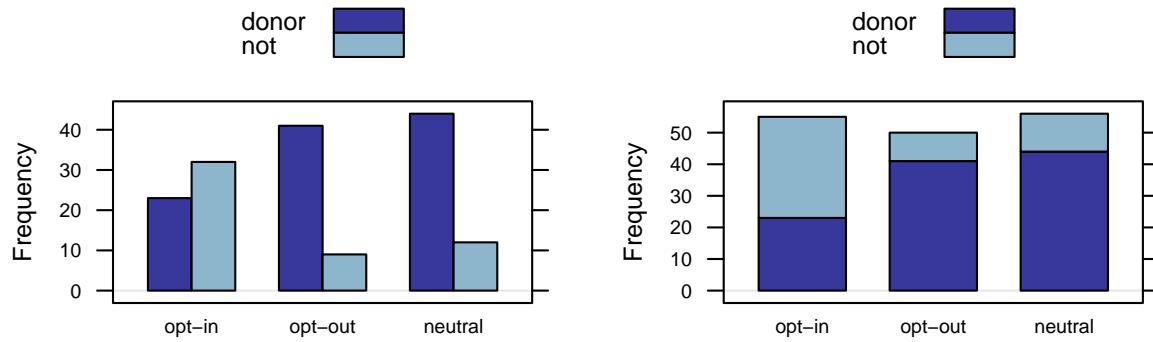
```
bargraph(~choice, data = OrganDonor)
bargraph(~default, data = OrganDonor)
```



Notice that the `bargraph()` uses the frequency, or counts.

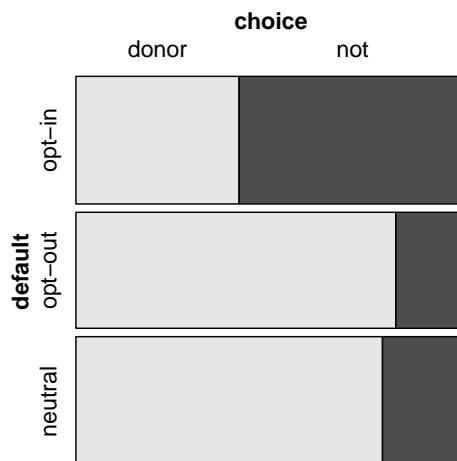
In order to graph the variable `default` and show what `choice` each option made, we can utilize the argument `groups=`.

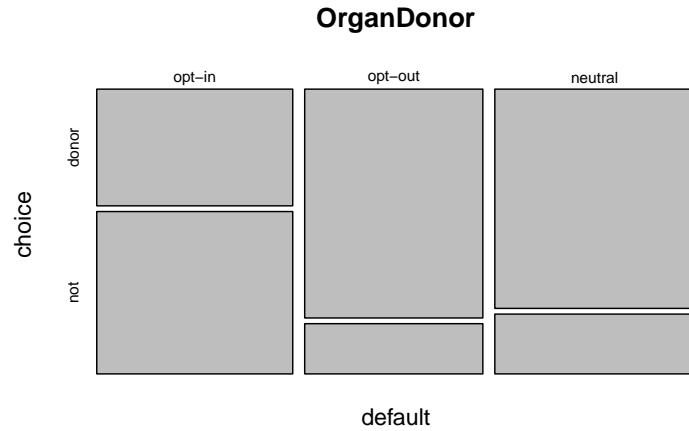
```
bargraph(~default, groups = choice, data = OrganDonor, auto.key = TRUE)
bargraph(~default, groups = choice, data = OrganDonor, auto.key = TRUE, stack = TRUE)
```



Although the bargraph is useful, the y-axis shows counts and not the percentages as in the text. The function `mosiac()` or `mosaicplot()` plots the variables relative to each other, in a way that reveals proportions, or percentages.

```
mosaic(choice ~ default, data = OrganDonor)
mosaicplot(default ~ choice, data = OrganDonor)
```





0.2 Exploring Data

Example P.2: Old Faithful

Everytime you use a new data set, it is beneficial to look at some key summary statistics.

```
head(OldFaithful)
```

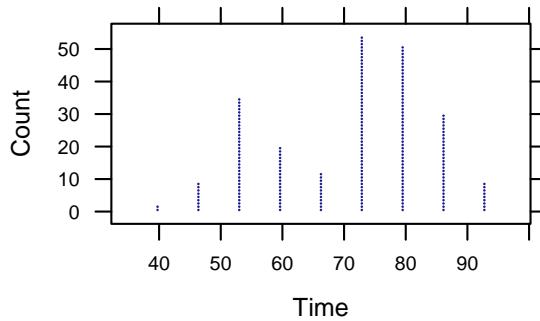
```
Time
1  55
2  58
3  56
4  50
5  51
6  60
```

```
summary(OldFaithful)
```

```
Time
Min.   :42
1st Qu.:60
Median :75
Mean   :71
3rd Qu.:81
Max.   :95
```

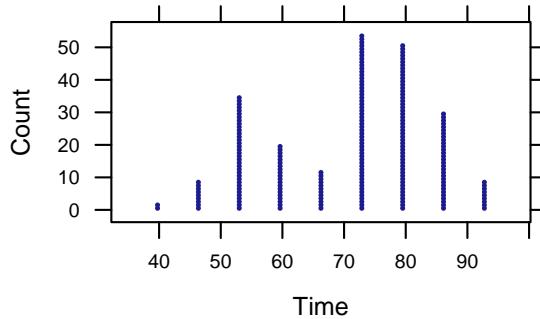
Another useful graph for examining the **shape**, **center**, and **variability** is the **dotplot**:

```
dotPlot(~Time, data = OldFaithful)
```



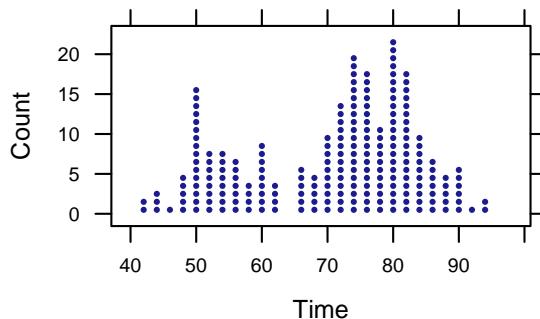
The dots in this plot are a bit small. The defaults for `dotPlot()` may not be the best way to examine a particular data set. We can increase the size of the dots using the `cex` argument. (`cex` stands for “character expansion” and is used to scale up or down the size of plotting characters – in this case the dots.)

```
dotPlot(~Time, data = OldFaithful, cex = 2)
```



Or we can change the distance between columns of dots

```
dotPlot(~Time, data = OldFaithful, width = 2)
```

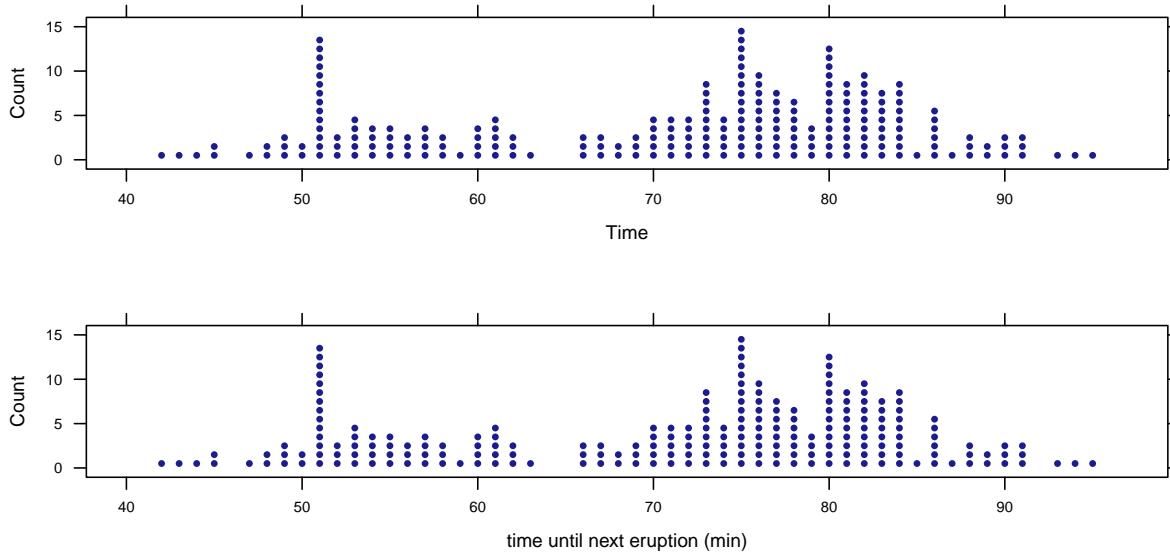


Notice that the dots have been automatically resized when we do this.

The appropriate choice may depend on the intended size and shape of the plot. The plots below are much wider, allowing us to present a finer view of the data. In the second plot, we have also added a more informative label.

```
dotPlot(~ Time, data = OldFaithful, width = 1)
dotPlot(~ Time, data = OldFaithful, width = 1,
        xlab = "time until next eruption (min)")
```

FigureP.3



Similar to the bargraph, we can organize the variables a little differently for the dotplot to graph them in relation to one another.

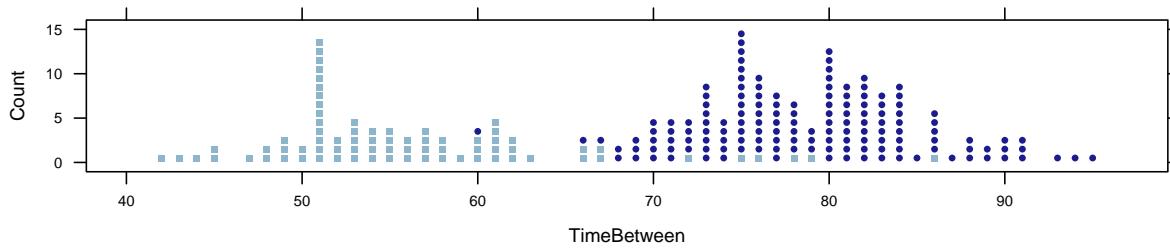
```
head(OldFaithful2)
```

	EruptionType	TimeBetween
1	short	55
2	short	58
3	short	56
4	short	50
5	short	51
6	short	60

```
summary(OldFaithful2)
```

	EruptionType	TimeBetween
long	:146	Min. :42
short	: 76	1st Qu.:60
		Median :75
		Mean :71
		3rd Qu.:81
		Max. :95

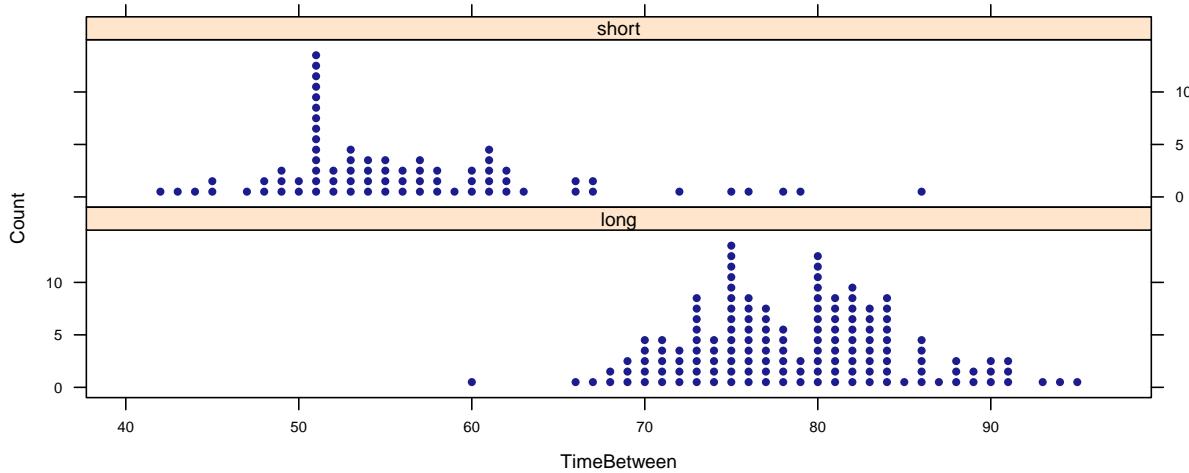
```
dotPlot(~TimeBetween, groups = EruptionType, data = OldFaithful2, width = 1)
```



The formula for a `lattice` plot can be extended to create multiple panels (sometimes called **facets**) based on a “condition”, often given by another variable. This is another way to look at multiple groups simultaneously. The general syntax for this becomes

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

```
dotPlot(~TimeBetween | EruptionType, data = OldFaithful2, width = 1, layout = c(1, 2))
```



For more key numerical summaries of the data set, we can use the `favstats()` for “favorite” statistics.

```
favstats(~TimeBetween, data = OldFaithful2)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	42	60	75	81	95	71.01	12.8	222	0

```
favstats(TimeBetween ~ EruptionType, data = OldFaithful2)
```

.group	min	Q1	median	Q3	max	mean	sd	n	missing
1 long	60	75	78.5	83.00	95	78.69	6.252	146	0
2 short	42	51	54.0	60.25	86	56.25	8.457	76	0

TableP1

Here are ways to find the mean and the standard deviation separately:

```
mean(~TimeBetween, data = OldFaithful2)
```

```
[1] 71.01
```

```
sd(~TimeBetween, data = OldFaithful2)
```

```
[1] 12.8
```

```
mean(TimeBetween ~ EruptionType, data = OldFaithful2)
```

```
long short  
78.69 56.25
```

```
sd(TimeBetween ~ EruptionType, data = OldFaithful2)
```

```
long short  
6.252 8.457
```

```
mean(~TimeBetween | EruptionType, data = OldFaithful2)
```

```
long short  
78.69 56.25
```

```
sd(~TimeBetween | EruptionType, data = OldFaithful2)
```

```
long short  
6.252 8.457
```

0.3 Exploring random Processes

Exploration P.3: Cars or Goats

The `mosaic` package has a function `rflip()` that simulates coin tosses. We define arguments `n` (the number of flips) and `prob` (the probability of heads).

```
rflip(n = 1, prob = 0.5)
```

```
Flipping 1 coin [ Prob(Heads) = 0.5 ] ...
```

```
H
```

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

```
rflip(n = 5, prob = 0.5)
```

```
Flipping 5 coins [ Prob(Heads) = 0.5 ] ...
T H T H H
Number of Heads: 3 [Proportion Heads: 0.6]
```

Although `rflip()` simulates coin tosses, where the probability of heads should be 0.5, we can also simulate any **random process** by changing the **probability**.

```
rflip(n = 15, prob = 1/3)

Flipping 15 coins [ Prob(Heads) = 0.3333333333333333 ] ...
T T T T H T T H H T H T H T T
Number of Heads: 5 [Proportion Heads: 0.33333333333333]
```

This is equivalent to the playing 15 games (flips), each game having a 1/3 chance of picking the car (heads).

Further, we can repeat each simulation many times by multiplying it by `do()`. When using `do()`, you should assign the simulation a name by using an arrow (`->`) so that you are creating a new data set with all of the repetitions. In this case, we are naming the simulation `GameSims`.

```
# 1000 samples, each of size 200 and proportion 1/3
GameSims <- do(1000) * rflip(n = 10, prob = 1/3)

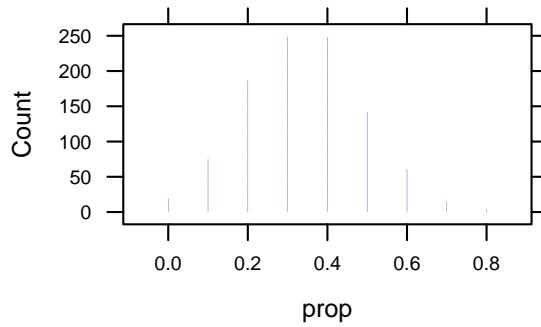
Loading required package: parallel

head(GameSims)

   n heads tails prop
1 10     3     7  0.3
2 10     4     6  0.4
3 10     4     6  0.4
4 10     5     5  0.5
5 10     3     7  0.3
6 10     2     8  0.2
```

Now we can create a dotplot of the proportion of wins but note that because of there are so many observations (1000), we will not be able to see the individual dots.

```
dotPlot(~ prop, data = GameSims, width = 0.1)
```



0.4 Other Visualizations

Several other types of plots can be used in place of dot plots to visualize the distribution of a single quantitative variable. The most familiar of these is the histogram, which replaces the dots of a histogram with rectangles and stacks them up touching each other to form bars. If instead we draw lines connecting the tops of each bar in a histogram (and then erase the bars), the result in a frequency polygon. A density plot is a smoother version of this idea.

Notice that to create these plots (and various numerical summaries), all we have to change is the name of the R function – all of them follow the same general template.

```

dotPlot(~ prop, data = GameSims, width = 0.1)
histogram(~ prop, data = GameSims, width = 0.1)
freqpolygon(~ prop, data = GameSims, width = 0.1, ylim=c(0,4))
densityplot(~ prop, data = GameSims)
densityplot(~ prop, data = GameSims, adjust=2)      # "smoother"
densityplot(~ prop, data = GameSims, adjust=0.5)    # less "smooth"
favstats(~ prop, data = GameSims)

min   Q1 median   Q3 max   mean      sd   n missing
0  0.2    0.3  0.4  0.8  0.3401  0.1485  1000       0

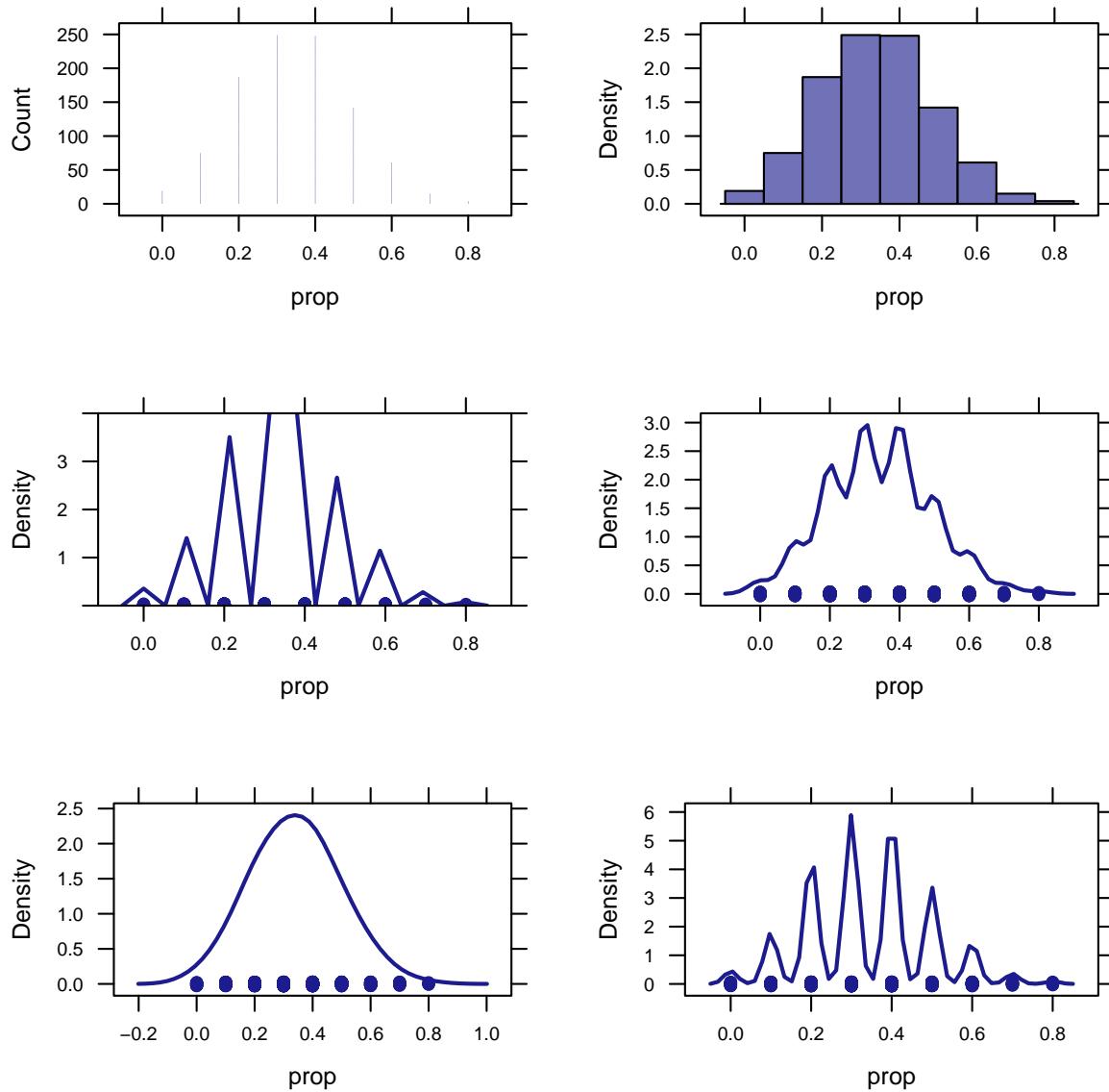
mean(~ prop, data = GameSims)

[1] 0.3401

sd(~ prop, data = GameSims)

[1] 0.1485

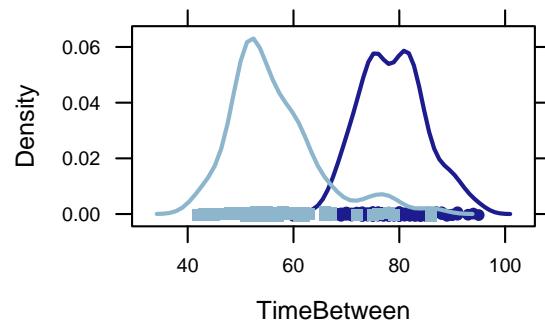
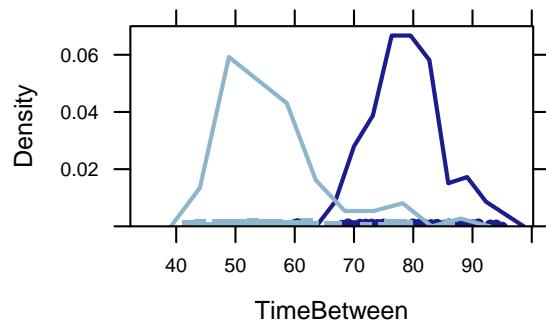
```



For this data set, a histogram is probably best. This is in part due to the discreteness of the data – there are only 11 possible values for `prop`.

Compared to dot plots, histograms, frequency polygons, and density plots handle a wider range of data sizes. The “sweet spot” for dot plots is around 100–1000 observations. Also, frequency polygons and density plot have the advantage that they can be overlaid.

```
freqpolygon(~TimeBetween, groups = EruptionType, data = OldFaithful2, ylim = c(0, 0.07))
densityplot(~TimeBetween, groups = EruptionType, data = OldFaithful2)
```



(The current version of `freqpolygon()` is not too clever about choosing the limits for the y-axis – sometimes you need to give it a hand.)

1

Significance: How strong is the evidence?

1.1 Introduction to Chance Models

Example 1.1: Can Dolphins Communicate?

The Chance Model

```
rflip(n = 16, prob = 0.5) # a sequence of 16 coin flips
```

Figure1.2

Flipping 16 coins [Prob(Heads) = 0.5] ...

T T T T T H H T T T T T T T H T

Number of Heads: 3 [Proportion Heads: 0.1875]

```
rflip(n = 16, prob = 0.5) # another sequence of 16 coin flips
```

Figure1.3

Flipping 16 coins [Prob(Heads) = 0.5] ...

T T T T T H T H T T T H T H H T

Number of Heads: 5 [Proportion Heads: 0.3125]

Using and evaluating the coin flip chance model

```
sim <- do(1000) * rflip(16, 0.5) # 1000 samples, each of size 16 and proportion 0.5
```

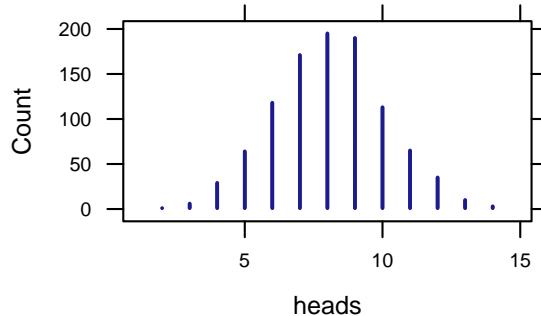
Figure1.4

Loading required package: parallel

```
head(sim, 3)

  n heads tails  prop
1 16      5     11 0.3125
2 16      9      7 0.5625
3 16      6     10 0.3750

dotPlot(~heads, data = sim, width = 1, cex = 3)
```



Another Doris and Buzz study

```
sim2 <- do(1000) * rflip(28, 0.5)

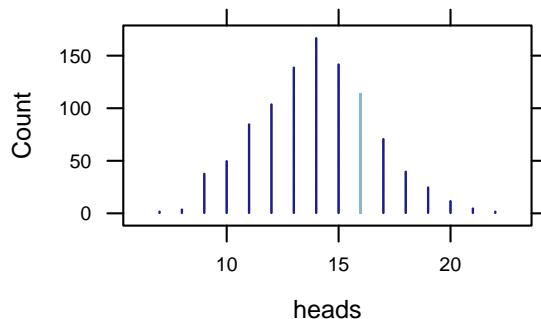
Loading required package: parallel

head(sim2, 3)

  n heads tails  prop
1 28     12     16 0.4286
2 28     17     11 0.6071
3 28     14     14 0.5000

dotPlot(~heads, data = sim2, width = 1, cex = 3, groups = (heads == 16))
```

Figure1.6



Notice the way we defined `groups` as `(groups = (heads == 16))` in order to differentiate the observations where `heads` equals 16. The `==` operator means “are equal to”. (We could also have used `groups = (heads != 16)` and the colors would be reversed.)

Exploration 1.1: Can Dogs Understand Human Cues?

The Chance Model

```
sim.harley <- do(1) * rflip(10, 0.5)
sim.harley
```

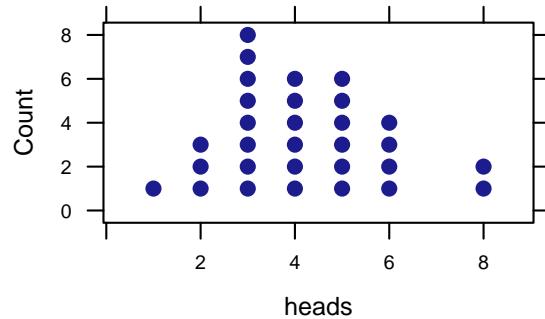
Exploration1.1.13

	n	heads	tails	prop
1	10	9	1	0.9

```
sim.class <- do(30) * rflip(10, 0.5)
head(sim.class, 3)
```

	n	heads	tails	prop
1	10	2	8	0.2
2	10	8	2	0.8
3	10	2	8	0.2

```
dotPlot(~heads, data = sim.class, width = 1, cex = 0.5)
```

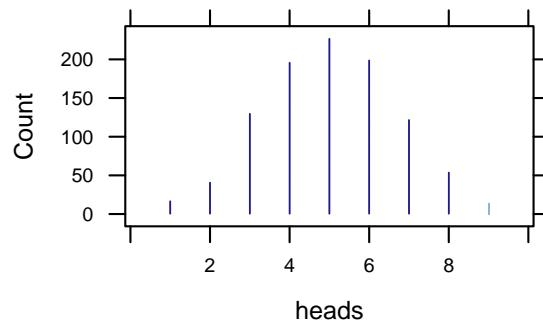


```
sim.harley2 <- do(1000) * rflip(10, 0.5)
head(sim.harley2, 3)
```

Exploration1.1.14

	n	heads	tails	prop
1	10	4	6	0.4
2	10	6	4	0.6
3	10	7	3	0.7

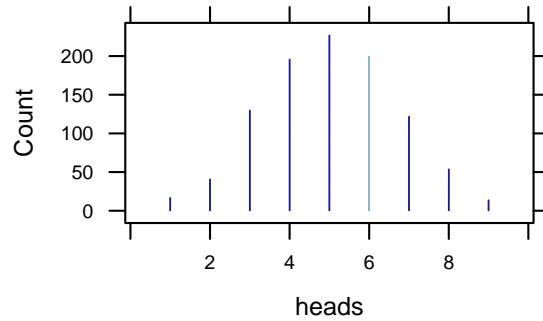
```
dotPlot(~heads, data = sim.harley2, width = 1, cex = 3, groups = (heads == 9))
```



Another Study

```
dotPlot(~heads, data = sim.harley2, width = 1, cex = 3, groups = (heads == 6))
```

Exploration1.1.23



1.2 Measuring the Strength of Evidence

Example 1.2: Rock Paper Scissors

1. $H_0: \pi = 1/3$

$$H_a: \pi < 1/3$$

Test statistic: $\hat{p} = 0.167$ (the sample proportion of 1/6)

2. We simulate a world in which $\pi = 1/3$:

```
sim.sci <- do(1000) * rflip(12, 1/3)

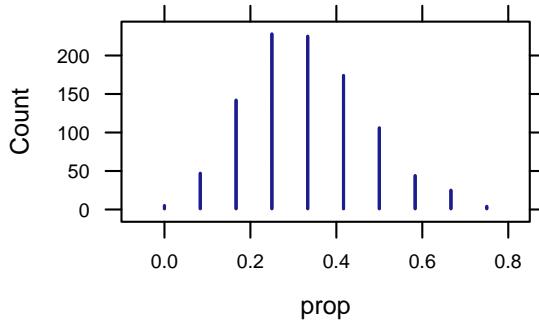
Loading required package: parallel

head(sim.sci, 3)

  n heads tails prop
1 12      3     9 0.2500
2 12      3     9 0.2500
3 12      4     8 0.3333
```

Figure1.7

```
dotPlot(~prop, data = sim.sci, width = 1/12, cex = 3)
```



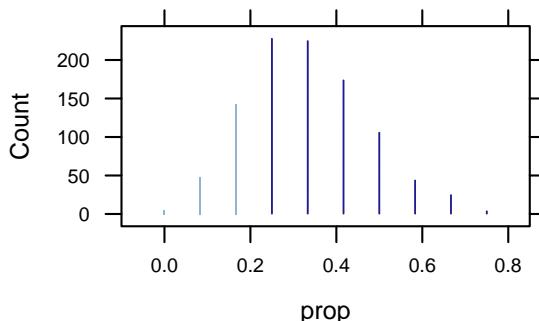
3. Strength of evidence:

For the **p-value**, you can use the `prop()` function and input `(prop <= 1/6)` to find the proportion of samples that is less than or equal to the observed proportion in the data set `sim.sci`.

```
dotPlot(~prop, data = sim.sci, cex = 3, width = 1/12, groups = (prop <= 1/6))
prop(~(prop <= 1/6), data = sim.sci)
```

Figure 1.8

```
TRUE
0.194
```

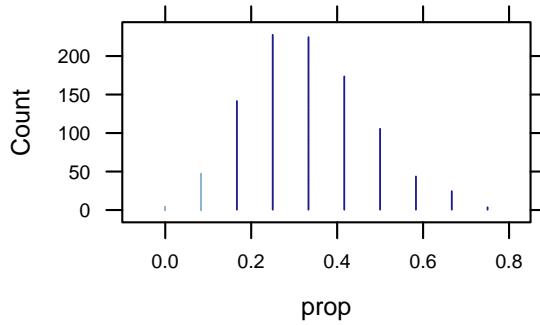


Conclusions

```
dotPlot(~prop, data = sim.sci, cex = 3, width = 1/12, groups = (prop <= 1/12))
prop(~(prop <= 1/12), data = sim.sci)
```

Figure 1.9

```
TRUE
0.052
```



Exploration 1.2: Tasting Water

1. $H_0: \pi = 1/4$
 $H_a: \pi < 1/4$
 Test statistic: $\hat{\pi} = 0.111$ (the sample proportion of 3/27)
2. We simulate a world in which $\pi = 1/4$:

```
sample.tap <- do(1) * rflip(27, 1/4)
```

Exploration1.2.18

Loading required package: parallel

```
sample.tap
```

n	heads	tails	prop
1	27	3	24

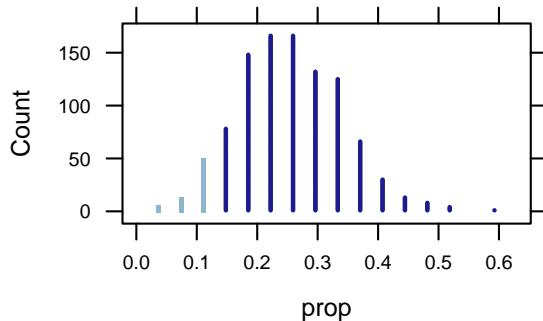
0.1111

```
sim.tap <- do(1000) * rflip(27, 1/4)
head(sim.tap, 3)
```

n	heads	tails	prop
1	27	8	19
2	27	5	22
3	27	3	24

0.2963
0.1852
0.1111

```
dotPlot(~prop, data = sim.tap, width = 1/27, cex = 3, groups = (prop <= 3/27))
```



3. Strength of evidence:

```
prop(~(prop <= 3/27), data = sim.tap)
```

Exploration1.2.20

```
TRUE  
0.063
```

Alternate Analysis

- $H_0: \pi = 3/4$

$$H_a: \pi > 3/4$$

Test statistic: $\hat{p} = 0.889$ (the sample proportion of 24/27)

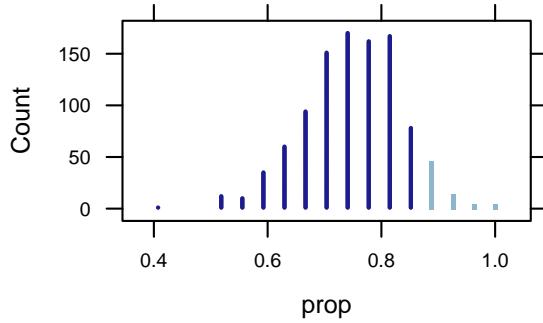
- We simulate a world in which $\pi = 3/4$:

```
sim.bottled <- do(1000) * rflip(27, 3/4)  
head(sim.bottled, 3)
```

Exploration1.2.26

	n	heads	tails	prop
1	27	19	8	0.7037
2	27	18	9	0.6667
3	27	20	7	0.7407

```
dotPlot(~prop, data = sim.bottled, width = 1/27, cex = 3, groups = (prop >= 24/27))
```



- Strength of evidence:

```
prop(~(prop >= 24/27), data = sim.bottled)
```

Exploration1.2.26b

```
TRUE  
0.06
```

1.3 Alternative Measure of Strength of Evidence

Example 1.3: Heart Transplant Operations

- $H_0: \pi = 0.15$

$H_a: \pi > 0.15$

Test statistic: $\hat{p} = 0.80$ (the sample proportion of 8/10)

2. We simulate a world in which $\pi = 0.15$:

```
sim.heart <- do(1000) * rflip(10, 0.15)
head(sim.heart, 3)

  n heads tails prop
1 10      1     9 0.1
2 10      1     9 0.1
3 10      1     9 0.1

mean(~prop, data = sim.heart)

[1] 0.1475

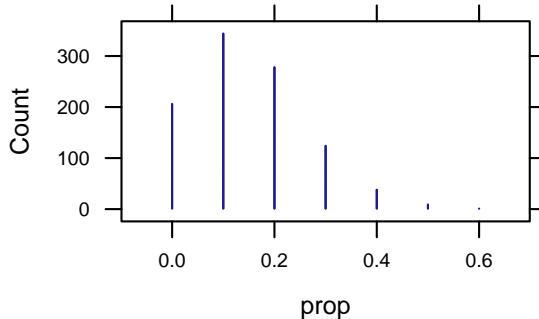
sd(~prop, data = sim.heart)

[1] 0.1125

favstats(~prop, data = sim.heart)

  min   Q1 median   Q3 max   mean      sd    n missing
  0 0.1  0.1 0.2 0.6 0.1475 0.1125 1000       0

dotPlot(~prop, data = sim.heart, width = 0.1, cex = 3, groups = (prop >= 8/10))
```



3. Strength of evidence:

```
prop(~(prop >= 8/10), data = sim.heart)

TRUE
0
```

Digging deeper into the St. George's mortality data

1. $H_0: \pi = 0.15$

$H_a: \pi > 0.15$

Test statistic: $\hat{p} = 0.197$ (the sample proportion of 71/361)

2. We simulate a world in which $\pi = 0.15$:

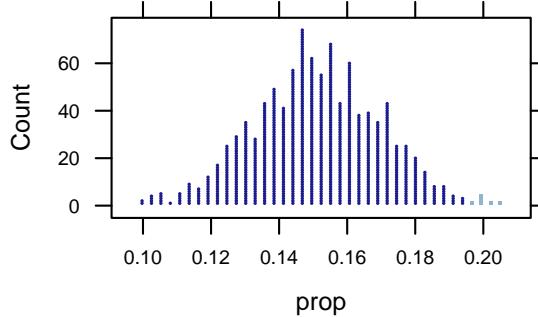
```
sim.1986 <- do(1000) * rflip(361, 0.15)
head(sim.1986, 3)

  n heads tails  prop
1 361     62    299 0.1717
2 361     57    304 0.1579
3 361     53    308 0.1468

favstats(~prop, data = sim.1986)

  min      Q1 median      Q3   max   mean      sd      n missing
0.09972 0.1385 0.1496 0.1634 0.205 0.151 0.01822 1000          0

dotPlot(~prop, data = sim.1986, width = 1/361, groups = (prop >= 71/361))
```



3. Strength of evidence:

```
prop(~(prop >= 71/361), data = sim.1986)

TRUE
0.007
```

An alternative to the p-value: Standardized value of a statistic

R can be used as a calculator so we can compute the **z-score** manually:

```
z <- (71/361 - 0.15) / 0.018; z  # z-score for sample size 361

[1] 2.593

z <- (8/10 - 0.15) / 0.113; z  # z-score for sample size 10

[1] 5.752
```

Figure1.11

Figure1.11b

Example1.3

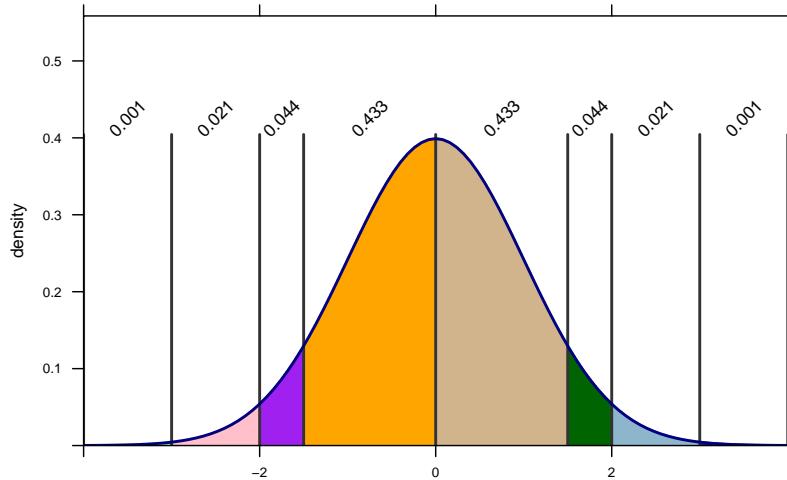
A very simple way to calculate the standardized statistic, find the p-value, and plot the bell-shaped curve is with the `xpnorm()` function. We'll examine `xpnorm()` in more detail later but for now, we just define a vector of quantiles (z-scores), `mean`, and `sd`.

```
xpnorm(c(-3, -2, -1.5, 0, 1.5, 2, 3), mean = 0, sd = 1)
```

Figure1.12

If $X \sim N(0,1)$, then

```
P(X <= -3) = P(Z <= -3) = 0.0013
P(X <= -2) = P(Z <= -2) = 0.0228
P(X <= -1.5) = P(Z <= -1.5) = 0.0668
P(X <= 0) = P(Z <= 0) = 0.5
P(X <= 1.5) = P(Z <= 1.5) = 0.9332
P(X <= 2) = P(Z <= 2) = 0.9772
P(X <= 3) = P(Z <= 3) = 0.9987
P(X > -3) = P(Z > -3) = 0.9987
P(X > -2) = P(Z > -2) = 0.9772
P(X > -1.5) = P(Z > -1.5) = 0.9332
P(X > 0) = P(Z > 0) = 0.5
P(X > 1.5) = P(Z > 1.5) = 0.0668
P(X > 2) = P(Z > 2) = 0.0228
P(X > 3) = P(Z > 3) = 0.0013
[1] 0.00135 0.02275 0.06681 0.50000 0.93319 0.97725 0.99865
```



In the example above, we input standardized values. However, we can input non-standardized statistics (observed statistic) with a new `mean` and `sd` in order to calculate the z-score.

```
xpnorm(71/361, mean = 0.15, sd = 0.018, plot = FALSE)
```

Example1.3b

If $X \sim N(0.15, 0.018)$, then

```
P(X <= 0.196675900277008) = P(Z <= 2.593) = 0.9952
P(X > 0.196675900277008) = P(Z > 2.593) = 0.0048
```

```
[1] 0.9952

xpnorm(8/10, mean = 0.15, sd = 0.113, plot = FALSE)

If X ~ N(0.15, 0.113), then

P(X <= 0.8) = P(Z <= 5.752) = 1
P(X > 0.8) = P(Z > 5.752) = 0
[1] 1
```

We'll ignore the p-values and plots for now and just realize that `xpnorm()` has computed the z-score for us so that we do not need to manually compute z by using R as a calculator.

Exploration 1.3: Do People Use Facial Prototyping?

$$1. H_0: \pi = 0.5$$

$$H_a: \pi > 0.5$$

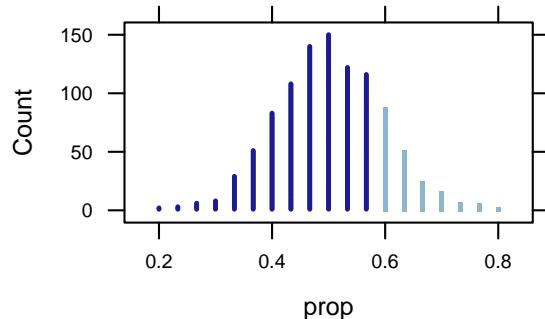
Test statistic: $\hat{p} = 0.6$ (the sample proportion of 18/30 for a fictitious class)

$$2. \text{ We simulate a world in which } \pi = 0.5:$$

```
sim.tim <- do(1000) * rflip(30, 0.5)
head(sim.tim, 3)

  n heads tails  prop
1 30     17    13 0.5667
2 30     15    15 0.5000
3 30     15    15 0.5000

dotPlot(~prop, data = sim.tim, width = 1/30, cex = 3, groups = (prop >= 18/30))
```



$$3. \text{ Strength of evidence:}$$

```
prop(~(prop >= 18/30), data = sim.tim)
```

```
TRUE
0.182
```

Exploration1.3.7

Exploration1.3.7b

```
mean(~prop, data = sim.tim)

[1] 0.4999

sd <- sd(~prop, data = sim.tim)
sd # assign the standard deviation to sd

[1] 0.09195

z <- (0.6 - 0.5)/sd
z # z-score using the assigned sd

[1] 1.088
```

Exploration1.3.8

Again, we can input the observed statistic, mean, and standard deviation to `xpnorm()` for the standardized statistic:

```
xpnorm(0.6, mean = 0.5, sd = sd, plot = FALSE)
```

Figure1.13

If $X \sim N(0.5, 0.0919456336924743)$, then

```
P(X <= 0.6) = P(Z <= 1.088) = 0.8616
P(X > 0.6) = P(Z > 1.088) = 0.1384
[1] 0.8616
```

Figure1.14

1.4 What Impacts Strength of Evidence?

Example 1.4: Predicting Elections from Faces?

1. $H_0: \pi = 0.5$

$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.719$ (the sample proportion of 23/32)

2. We simulate a world in which $\pi = 0.5$:

```
sim.senate <- do(1000) * rflip(32, 0.5)
head(sim.senate, 3)

   n heads tails prop
1 32     15    17 0.4688
2 32     13    19 0.4062
3 32     16    16 0.5000

favstats(~prop, data = sim.senate)
```

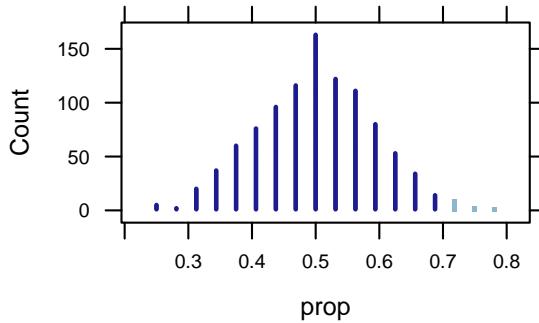
Figure1.14

```

min      Q1 median      Q3      max    mean      sd     n missing
0.25 0.4375 0.5 0.5625 0.7812 0.4999 0.09005 1000        0

dotPlot(~prop, data = sim.senate, groups = (prop >= 23/32), width = 1/32, cex = 3)

```



3. Strength of evidence:

```

prop(~(prop >= 23/32), data = sim.senate)

TRUE
0.011

```

Figure1.14b

Strength of evidence with the standardized statistic:

```

mean(~prop, data = sim.senate)

[1] 0.4999

sd <- sd(~prop, data = sim.senate)
sd

[1] 0.09005

xpnorm(23/32, 0.5, sd, plot = FALSE)

```

Figure1.14c

If $X \sim N(0.5, 0.090053631417792)$, then

```

P(X <= 0.71875) = P(Z <= 2.429) = 0.9924
P(X > 0.71875) = P(Z > 2.429) = 0.0076
[1] 0.9924

```

Figure1.15

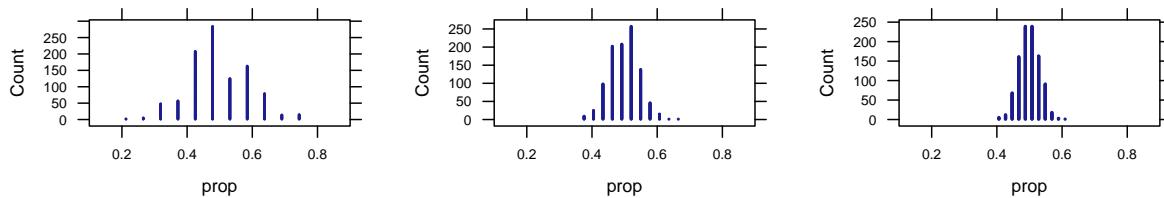
What impacts strength of evidence?

```
senate.32 <- do(1000) * rflip(32, 0.5)
```

Figure1.15

Loading required package: parallel

```
dotPlot(~prop, data = senate.32, xlim = c(0.1, 0.9), cex = 5)
senate.128 <- do(1000) * rflip(128, 0.5)
dotPlot(~prop, data = senate.128, xlim = c(0.1, 0.9), cex = 5)
senate.256 <- do(1000) * rflip(256, 0.5)
dotPlot(~prop, data = senate.256, xlim = c(0.1, 0.9), cex = 5)
```



```
sd(~prop, data = senate.32)
```

```
[1] 0.08823
```

```
sd(~prop, data = senate.128)
```

```
[1] 0.04412
```

```
sd(~prop, data = senate.256)
```

```
[1] 0.03056
```

Figure 1.15b

```
prop(~(prop >= 23/32), data = senate.32)
```

```
TRUE  
0.015
```

Figure 1.15c

```
prop(~(prop >= 23/32), data = senate.128)
```

```
TRUE  
0
```

```
prop(~(prop >= 23/32), data = senate.256)
```

```
TRUE  
0
```

$$1. H_0: \pi = 0.5$$

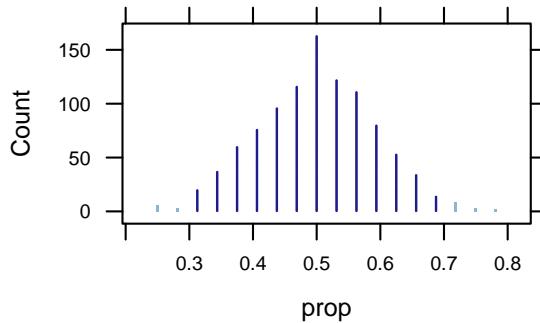
$$H_a: \pi \neq 0.5$$

Test statistic: $\hat{p} = 0.719$ (the sample proportion of 23/32)

2. We use the simulated world in which $\pi = 0.5$:

```
dotPlot(~ prop, data = sim.senate, groups = (prop >= 23/32 | prop <= 9/32),
       width = 1/32, cex = 3)
```

Figure1.16



Notice that because we are doing a two-sided test, we differentiate the samples with proportions greater than or equal to $23/32$ and proportions less than or equal to $9/32$ (the proportion that is as extreme as $23/32$) by using the bar |.

3. Strength of evidence:

```
prop(~(prop <= 9/32 | prop >= 23/32), data = sim.senate)

TRUE
0.018
```

Figure1.16b

Follow-up Study

- $H_0: \pi = 0.5$
 - $H_a: \pi \neq 0.5$
- Test statistic: $\hat{p} = 0.677$ (the sample proportion of 189/279)
- We simulate a world in which $\pi = 0.5$:

```
sim.house <- do(1000) * rflip(279, 0.5)
head(sim.house, 3)

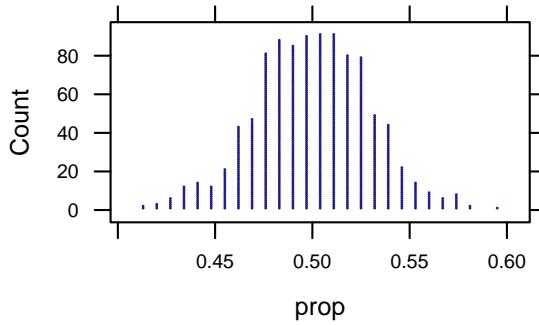
  n heads tails   prop
1 279    147    132 0.5269
2 279    134    145 0.4803
3 279    150    129 0.5376

favstats(~prop, data = sim.house)

  min      Q1 median      Q3    max    mean      sd      n missing
0.4122 0.4803 0.4982 0.5197 0.595 0.4996 0.02948 1000         0

dotPlot(~prop, data = sim.house, groups = (prop >= 189/279 | prop <= 90/279), width = 0.007)
```

Figure1.17



3. Strength of evidence:

```
prop(~(prop >= 189/279 | prop <= 90/279), data = sim.house)
```

TRUE
0

Figure1.17b

Strength of evidence with the standardized statistic:

```
mean(~prop, data = sim.house)
```

[1] 0.4996

```
sd <- sd(~prop, data = sim.house)
sd
```

[1] 0.02948

```
xpnorm(189/279, 0.5, sd, plot = FALSE)
```

Figure1.17c

If $X \sim N(0.5, 0.0294794207694169)$, then

```
P(X <= 0.67741935483871) = P(Z <= 6.018) = 1
P(X > 0.67741935483871) = P(Z > 6.018) = 0
[1] 1
```

Exploration 1.4: Competitive Advantage to Uniform Colors?

1. $H_0: \pi = 0.5$

$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.543$ (the sample proportion of 248/457)

2. We simulate a world in which $\pi = 0.5$:

```
sim.red <- do(1000) * rflip(457, 0.5)
head(sim.red, 3)
```

Exploration1.4.3

```

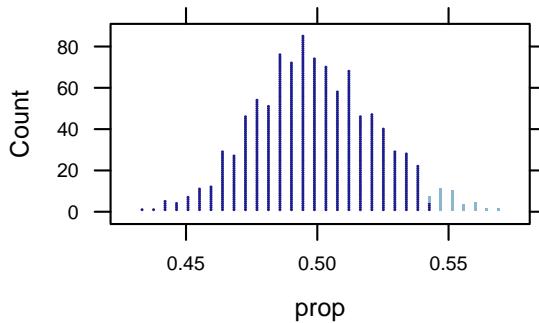
n heads tails prop
1 457 228 229 0.4989
2 457 234 223 0.5120
3 457 247 210 0.5405

favstats(~prop, data = sim.red)

min      Q1 median      Q3    max   mean      sd     n missing
0.4354 0.4858 0.4989 0.5142 0.5689 0.5002 0.0228 1000       0

dotPlot(~prop, data = sim.red, groups = (prop >= 0.543), width = 2/457)

```



3. Strength of evidence:

```
prop(~(prop >= 0.543), data = sim.red)
```

Exploration1.4.3b

```
TRUE
0.033
```

1. $H_0: \pi = 0.5$

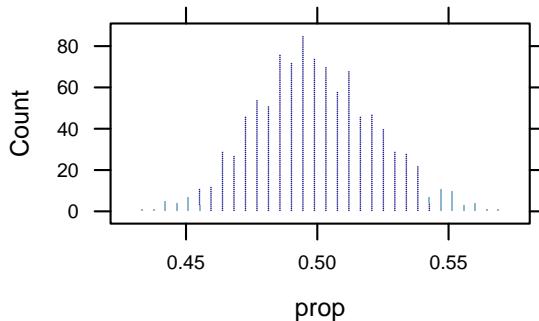
$H_a: \pi \neq 0.5$

Test statistic: $\hat{p} = 0.543$ (the sample proportion of 248/457)

2. We use the simulated world in which $\pi = 0.5$ from the one-sided test:

```
dotPlot(~prop, data = sim.red, groups = (prop <= 0.457 | prop >= 0.543), width = 2/457)
```

Exploration1.4.5



3. Strength of evidence:

```
prop(~(prop <= 0.457 | prop >= 0.543), data = sim.red)
```

TRUE
0.054

Exploration1.4.5b

Difference between statistic and null hypothesis parameter value

- $H_0: \pi = 0.5$

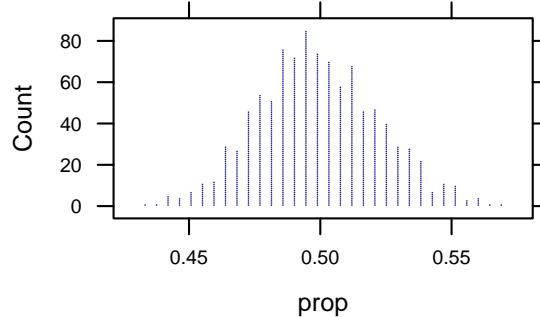
$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.57$ (the sample proportion)

- We use the simulated world in which $\pi = 0.5$:

```
dotPlot(~prop, data = sim.red, groups = (prop >= 0.57), width = 2/457)
```

Exploration1.4.6



3. Strength of evidence:

```
prop(~(prop >= 0.57), data = sim.red)
```

TRUE
0

Exploration1.4.6b

Sample size

- $H_0: \pi = 0.5$

$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.551$ (the sample proportion of 150/272)

- We simulate a world in which $\pi = 0.5$:

```
sim.box <- do(1000) * rflip(272, 0.5)
head(sim.box, 3)
```

Exploration1.4.7

```

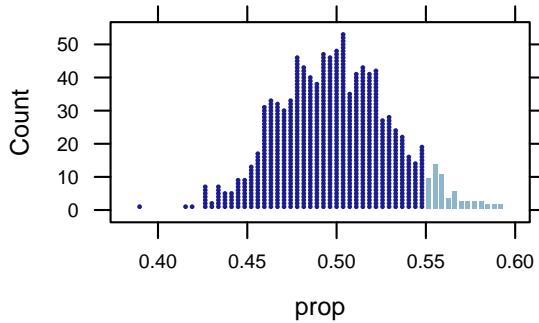
      n heads tails   prop
1 272    134    138 0.4926
2 272    120    152 0.4412
3 272    128    144 0.4706

favstats(~prop, data = sim.box)

min     Q1 median     Q3   max   mean      sd    n missing
0.3897 0.4779     0.5 0.5184 0.5919 0.4988 0.03054 1000        0

dotPlot(~prop, data = sim.box, groups = (prop >= 0.551), width = 1/272)

```



3. Strength of evidence

```

prop(~(prop >= 0.551), data = sim.box)

TRUE
0.051

```

Exploration1.4.7b

1.5 Inference on a single proportion: Theory-based approach

Example 1.5: Halloween Treats

1. $H_0: \pi = 0.5$
 $H_a: \pi \neq 0.5$
Test statistic: $\hat{\pi} = 0.523$ (the sample proportion of 148/283)

2. We simulate a world in which $\pi = 0.5$:

```

sim.candy <- do(1000) * rflip(283, 0.5)
head(sim.candy, 3)

      n heads tails   prop
1 283    157    126 0.5548
2 283    153    130 0.5406
3 283    136    147 0.4806

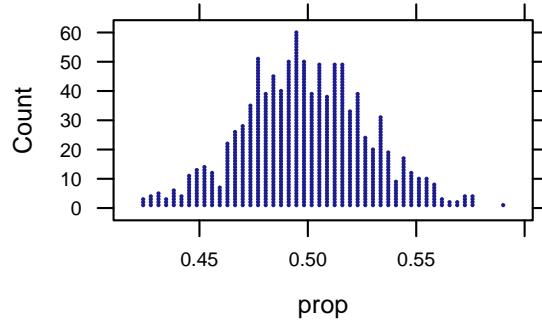
```

Figure1.19

```
favstats(~prop, data = sim.candy)

min      Q1 median      Q3    max   mean      sd     n missing
0.424  0.4806  0.4982  0.5159  0.5901  0.4991  0.02852  1000        0

dotPlot(~prop, data = sim.candy, width = 1/283)
```



Theory-based approach (One proportion z test)

Calculating predicted standard deviation:

```
mean <- 0.5
n <- 283
sd <- sqrt(mean * (1 - mean)/n)
sd

[1] 0.02972
```

Example1.5

Calculating z-score:

```
z <- (0.523 - mean)/sd
z

[1] 0.7738

xpnorm(148/283, 0.5, sd, plot = FALSE)
```

Example1.5b

If $X \sim N(0.5, 0.0297219149138882)$, then

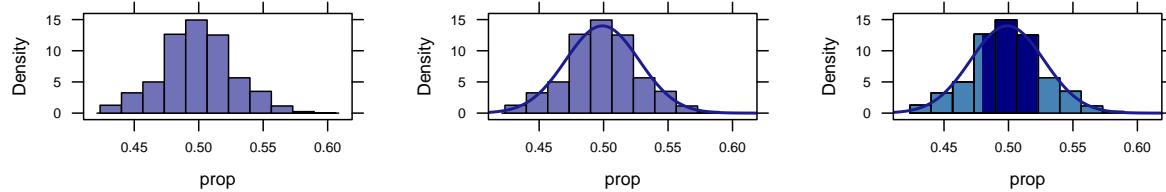
```
P(X <= 0.522968197879859) = P(Z <= 0.773) = 0.7802
P(X > 0.522968197879859) = P(Z > 0.773) = 0.2198
[1] 0.7802
```

To overlay a normal approximation, let's graph a histogram using `histogram()` instead of a dotplot:

```
histogram(~prop, data = sim.candy)
histogram(~prop, data = sim.candy, fit = "normal")
histogram(~prop, data = sim.candy, fit = "normal", group = cut(prop, c(0, 135/283, 148/283,
  1)), fcol = c("steelblue", "navy", "steelblue"))
prop(~(prop <= 135/283 | prop >= 148/283), data = sim.candy)
```

Figure1.20

TRUE
0.459



The two main functions we need for working with normal distributions are `pnorm()` and `qnorm()`. `pnorm()` computes the proportion of a normal distribution below a specified value:

$$\text{pnorm}(x, \text{mean}=\mu, \text{sd}=\sigma) = \Pr(X \leq x)$$

when $X \sim \text{Norm}(\mu, \sigma)$.

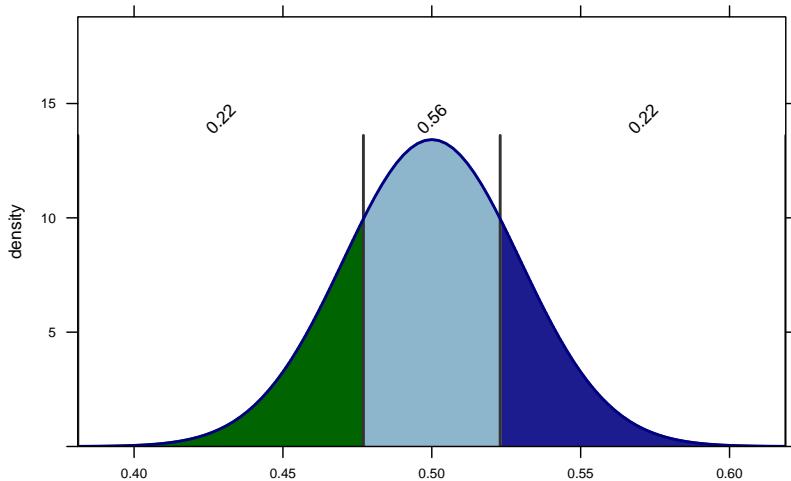
We can obtain arbitrary probabilities using `pnorm()`. We can now examine the rest of the output from `xpnorm()`, which is an augmented version of `pnorm()`. Because it's a two-sided test, we can input both the observed statistic (148/283) and the statistic that is as extreme as the observed (135/283).

```
xpnorm(c(135/283, 148/283), 0.5, sd)
```

Figure1.20b

If $X \sim N(0.5, 0.0297219149138882)$, then

```
P(X <= 0.477031802120141) = P(Z <= -0.773) = 0.2198
P(X <= 0.522968197879859) = P(Z <= 0.773) = 0.7802
P(X > 0.477031802120141) = P(Z > -0.773) = 0.7802
P(X > 0.522968197879859) = P(Z > 0.773) = 0.2198
[1] 0.2198 0.7802
```



The output gives the z-scores for both statistics and the p-value. We know now that this p-value is found using the predicted standard deviation and normal approximation. The p-value for the two-sided test is the sum of $P(Z \leq -0.773)$ and $P(Z \geq 0.773)$.

We can also use the just observed statistic as we have done before but only we will need to change the `lower.tail` to `FALSE`.

```
xpnorm(148/283, 0.5, sd, lower.tail = FALSE, plot = FALSE)
```

Figure1.20c

If $X \sim N(0.5, 0.0297219149138882)$, then

```
P(X <= 0.522968197879859) = P(Z <= 0.773) = 0.7802
P(X > 0.522968197879859) = P(Z > 0.773) = 0.2198
[1] 0.2198
```

```
2 * xpnorm(148/283, 0.5, sd, lower.tail = FALSE, plot = FALSE)
```

If $X \sim N(0.5, 0.0297219149138882)$, then

```
P(X <= 0.522968197879859) = P(Z <= 0.773) = 0.7802
P(X > 0.522968197879859) = P(Z > 0.773) = 0.2198
[1] 0.4397
```

Figure1.20d

This results in the p-value of the alternative hypothesis that π is greater than the observed statistic (the default is the alternative hypothesis that π is less than the observed statistic). For the two-sided test, we have multiplied the resulting p-value by two.

The function `pnorm()` can be used just to find the p-value:

```
2 * pnorm(148/283, 0.5, sd, lower.tail = FALSE)
```

```
[1] 0.4397
```

Further, we can input the standardized statistic (z-score) to find the p-value:

```
2 * pnorm(z, 0, 1, lower.tail = FALSE)
[1] 0.439
```

Figure1.20e

The most convenient way to find the p-value for a proportion using normal approximation is to use `prop.test()` by inputting the number of successes and the number of samples:

```
prop.test(148, n = 283)

1-sample proportions test with continuity correction

data: x and n
X-squared = 0.5088, df = 1, p-value = 0.4756
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
0.4631 0.5822
sample estimates:
p
0.523
```

Example1.5c

Note that the default for the prop test is with a $\pi = 0.5$, two-sided test, and a continuity correction. The continuity correction results in a more accurate p-value but if you want the p-value found with `pnorm()` we can change the default.

```
prop.test(148, 283, correct = FALSE)

1-sample proportions test without continuity correction

data: x and n
X-squared = 0.5972, df = 1, p-value = 0.4397
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
0.4649 0.5805
sample estimates:
p
0.523
```

Figure1.5d

A situation where a theory-based approach doesn't work

```
mean <- 1/3
n <- 12
sd <- sqrt(mean * (1 - mean)/n)
sd
```

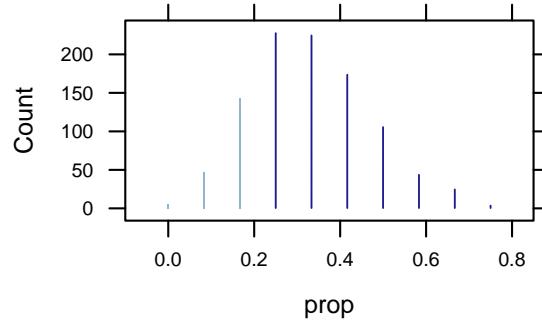
Example1.5e

```
[1] 0.1361
```

```
dotPlot(~prop, data = sim.sci, group = (prop <= 1/6), width = 1/12, cex = 3)
prop(~(prop <= 1/6), data = sim.sci)
```

TRUE
0.194

Figure1.21

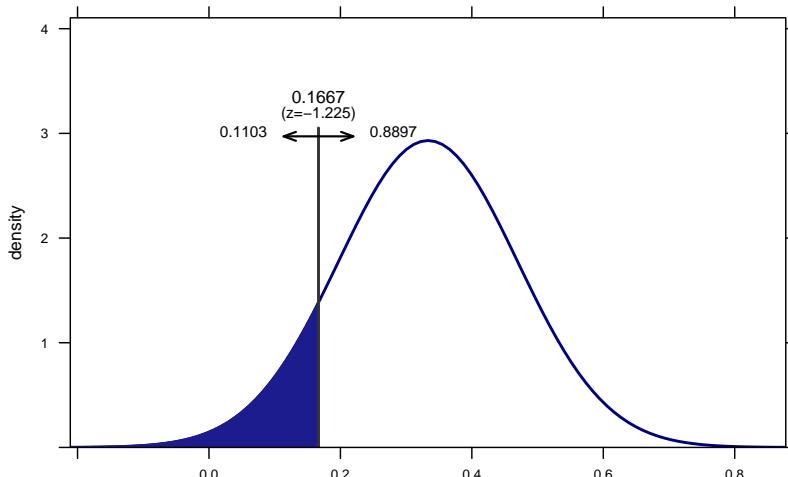


```
xpnorm(1/6, 1/3, sd)
```

If $X \sim N(0.33333333333333, 0.136082763487954)$, then

```
P(X <= 0.166666666666667) = P(Z <= -1.225) = 0.1103
P(X > 0.166666666666667) = P(Z > -1.225) = 0.8897
[1] 0.1103
```

Figure1.21b



Exploration 1.5: Calling Heads or Tails

$$1. H_0: \pi = 0.5$$

$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.651$ (the sample proportion of 54/83)

2. We simulate a world in which $\pi = 0.5$:

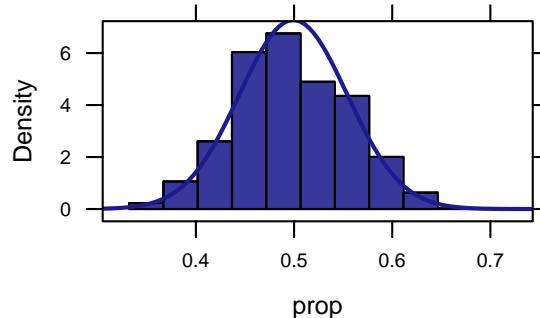
```
sim.heads <- do(1000) * rflip(83, 0.5)
head(sim.heads, 3)

  n heads tails prop
1 83     42     41 0.5060
2 83     39     44 0.4699
3 83     38     45 0.4578

favstats(~prop, data = sim.heads)

  min      Q1 median      Q3    max   mean      sd      n missing
0.3373 0.4578 0.494 0.5301 0.6867 0.4992 0.05489 1000         0

histogram(~prop, data = sim.heads, groups = (prop >= 54/83), fit = "normal")
```



3. Strength of evidence

```
prop(~(prop >= 54/83), data = sim.heads)

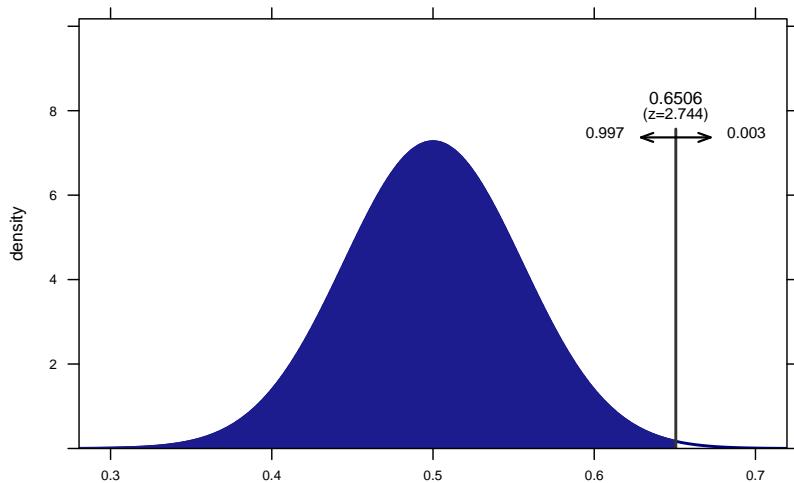
TRUE
0.002
```

Normal approximation using simulated sd:

```
sd <- sd(~prop, data = sim.heads)
xpnorm(54/83, 0.5, sd, lower.tail = FALSE)
```

If $X \sim N(0.5, 0.0548884586448383)$, then

```
P(X <= 0.650602409638554) = P(Z <= 2.744) = 0.997
P(X > 0.650602409638554) = P(Z > 2.744) = 0.003
[1] 0.003037
```



Formulas

```
sd <- sqrt(0.5 * (1 - 0.5)/83)
sd
```

Exploration1.5.8

```
[1] 0.05488
```

```
xpnorm(54/83, 0.5, sd, plot = FALSE, lower.tail = FALSE)
```

Exploration1.5.9

If $X \sim N(0.5, 0.05488^2)$, then

```
P(X <= 0.6506) = P(Z <= 2.744) = 0.997
P(X > 0.6506) = P(Z > 2.744) = 0.003
[1] 0.003034
```

```
prop.test(54, 83, alt = "greater", correct = FALSE)
```

```
1-sample proportions test without continuity correction
```

```
data: x and n
X-squared = 7.53, df = 1, p-value = 0.003034
alternative hypothesis: true p is greater than 0.5
95 percent confidence interval:
 0.561 1.000
sample estimates:
      p
0.6506
```

Follow-up Analysis #1

$$1. H_0: \pi = 0.5$$

$$H_a: \pi \neq 0.5$$

Test statistic: $\hat{p} = 0.70$ (the sample proportion)

$$2. \text{ Normal approximation using predicted sd:}$$

```
sd <- sqrt(0.5 * (1 - 0.5)/83)
sd

[1] 0.05488

2 * xpnorm(0.7, 0.5, sd, plot = FALSE, lower.tail = FALSE)
```

Exploration1.5.12

If $X \sim N(0.5, 0.05488^2)$, then

$P(X \leq 0.7) = P(Z \leq 3.644) = 0.9999$
 $P(X > 0.7) = P(Z > 3.644) = 1e-04$
[1] 0.0002683

Approximate test for proportions without continuity correction:

```
prop.test(58.1, 83, correct = FALSE) # 58.1 = 0.70 * 83
```

Exploration1.5.12b

1-sample proportions test without continuity correction
data: x and n
X-squared = 13.28, df = 1, p-value = 0.0002683
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
0.5944 0.7879
sample estimates:
p
0.7

Follow-up Analysis # 2

$$1. H_0: \pi = 0.5$$

$$H_a: \pi \neq 0.5$$

Test statistic: $\hat{p} = 0.875$ (the sample proportion of 7/8)

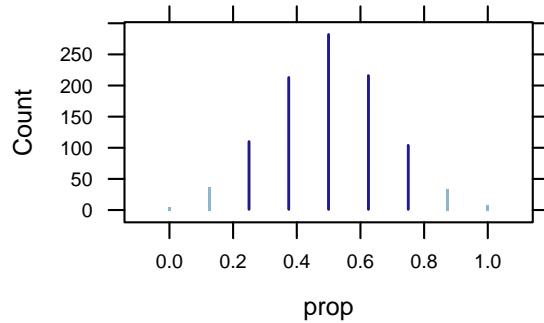
$$2. \text{ We simulate a world in which } \pi = 0.5:$$

```
sim.small <- do(1000) * rflip(8, 0.5)
head(sim.small, 3)

  n heads tails prop
1 8      4      4 0.500
2 8      3      5 0.375
3 8      6      2 0.750

dotPlot(~prop, data = sim.small, groups = (prop <= 0.125 | prop >= 0.875), width = 1/8, cex = 3)
```

Exploration1.5.13



3. Strength of evidence:

```
prop(~(prop <= 0.125 | prop >= 0.875), data = sim.small)
```

Exploration1.5.13b

```
TRUE  
0.075
```

Approximate test for proportions without continuity correction:

```
prop.test(7, 8, correct = FALSE)
```

Exploration1.5.13c

```
1-sample proportions test without continuity correction

data: x and n
X-squared = 4.5, df = 1, p-value = 0.03389
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5291 0.9776
sample estimates:
      p
0.875
```

There is also another test that will compute the p-value for a proportion and that the binomial test. `binom.test()` utilizes a binomial probability distribution while `prop.test()` utilizes a normal probability distribution. The tests are similar but the binomial test will result in the most accurate p-value.

```
binom.test(7, 8)
```

Exact binomial test

```
data: x and n
number of successes = 7, number of trials = 8, p-value = 0.07031
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4735 0.9968
sample estimates:
probability of success
          0.875
```

```
binom.test(58, 83)
```

Exact binomial test

```
data: x and n  
number of successes = 58, number of trials = 83, p-value = 0.0003783  
alternative hypothesis: true probability of success is not equal to 0.5  
95 percent confidence interval:  
 0.5882 0.7947  
sample estimates:  
probability of success  
 0.6988
```


2

Generalization: How Broadly Do the Results Apply?

2.1 Sampling from a Finite Population

Example 2.1A: Sampling Students

```
head(CollegeMidwest, 8)
```

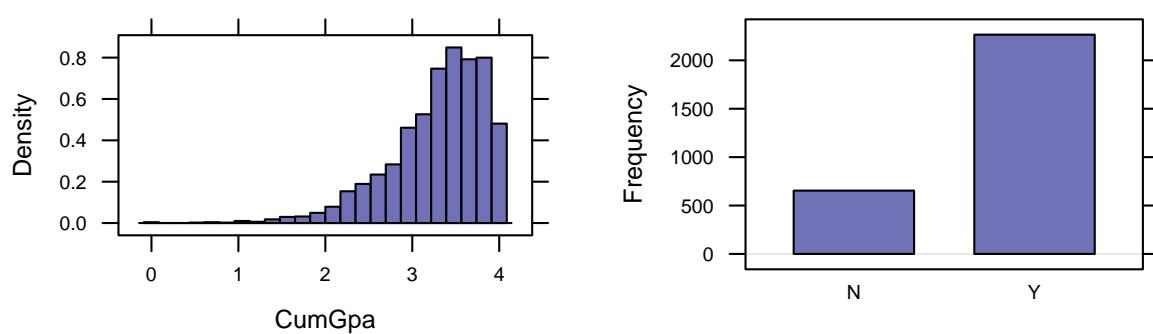
Table2.1

	OnCampus	CumGpa
1	N	2.92
2	N	3.59
3	N	3.36
4	N	2.47
5	N	3.46
6	Y	2.98
7	Y	3.07
8	Y	3.79

In chapter one, we used **histograms** a few times instead of dotplots and changed their widths. You can also control the number of bins by defining **nint**, or **n** for short.

```
histogram(~CumGpa, data = CollegeMidwest, n = 24)
bargraph(~OnCampus, data = CollegeMidwest)
```

Figure2.1



Simple Random Samples

For a **simple random sample** of a data set, we use `sample()` and define the size of the same we want.

```
sample1 <- sample(CollegeMidwest, 30)
sample1
```

Table2.2

	OnCampus	CumGpa	orig.ids
178	N	3.02	178
176	Y	3.42	176
679	N	3.38	679
1988	Y	2.72	1988
2046	Y	3.43	2046
133	Y	2.22	133
768	Y	3.76	768
1328	N	3.10	1328
2312	Y	3.93	2312
2048	N	3.57	2048
408	N	3.77	408
1151	Y	3.43	1151
2076	N	2.13	2076
1269	N	3.36	1269
1437	N	3.69	1437
296	Y	3.99	296
1867	Y	3.07	1867
1195	Y	3.00	1195
930	Y	3.36	930
1858	Y	4.00	1858
1569	N	1.87	1569
2108	Y	3.30	2108
172	Y	3.80	172
2342	Y	3.66	2342
2202	N	2.81	2202
1348	Y	3.76	1348
1732	Y	3.17	1732
1185	N	2.36	1185
220	Y	3.77	220
2399	Y	2.95	2399

```
sample2 <- sample(CollegeMidwest, 30)
sample3 <- sample(CollegeMidwest, 30)
```

```
sample4 <- sample(CollegeMidwest, 30)
sample5 <- sample(CollegeMidwest, 30)
```

```
mean(~CumGpa, data = sample1)
[1] 3.26

mean(~CumGpa, data = sample2)
[1] 3.244

mean(~CumGpa, data = sample3)
[1] 3.403

mean(~CumGpa, data = sample4)
[1] 3.35

mean(~CumGpa, data = sample5)
[1] 3.277

prop(~OnCampus, level = "Y", data = sample1)
Y
0.6333

prop(~OnCampus, level = "Y", data = sample2)
Y
0.7667

prop(~OnCampus, level = "Y", data = sample3)
Y
0.7333

prop(~OnCampus, level = "Y", data = sample4)
Y
0.8333

prop(~OnCampus, level = "Y", data = sample5)
Y
0.6333
```

Table2.3

Notice the `level` in order to find the proportion of students who said “yes” instead of the default “no”.

Similar to the simulation of random processes in chapter one, we can repeat taking different simple random samples. Conveniently, R will let you set `data=` to a simple random sample so we can repeat finding the mean or the proportion of a different simple random sample many times.

```
sample.gpa <- do(1000) * mean(~CumGpa, data = sample(CollegeMidwest, 30))
```

Figure2.2

Loading required package: parallel

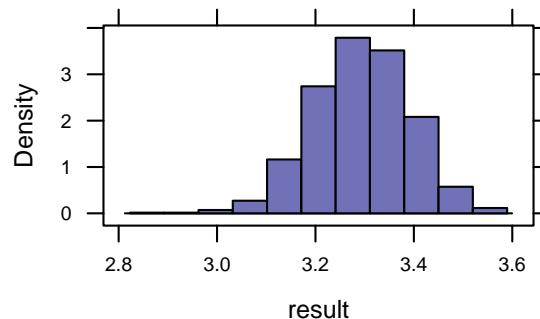
```
head(sample.gpa)
```

```
result
1 3.401
2 3.261
3 3.283
4 3.334
5 3.353
6 2.949
```

```
favstats(~result, data = sample.gpa)
```

min	Q1	median	Q3	max	mean	sd	n	missing
2.876	3.228	3.295	3.364	3.573	3.294	0.09847	1000	0

```
histogram(~result, data = sample.gpa)
```



```
sample.campus <- do(1000) * prop(~OnCampus, level = "Y", data = sample(CollegeMidwest, 30))
head(sample.campus)
```

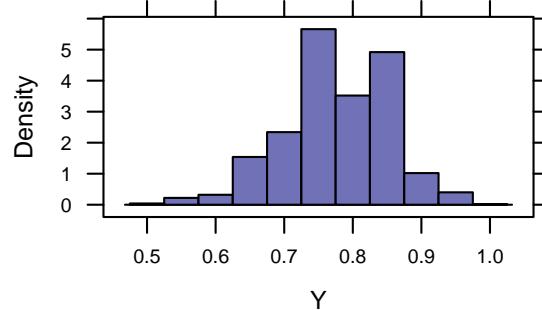
Figure2.2b

```
Y
1 0.6667
2 0.8333
3 0.7333
4 0.8000
5 0.7667
6 0.7667
```

```
favstats(~Y, data = sample.campus)

min      Q1 median      Q3 max   mean      sd    n missing
0.5 0.7333 0.7667 0.8333    1 0.7768 0.07767 1000        0

histogram(~Y, data = sample.campus)
```



Exploration 2.1A: Sampling Words

```
head(GettysburgAddress)

[1] "Four"   "score"  "and"    "seven"  "years"  "ago"

words <- sample(GettysburgAddress, 10)
nchar(words[1:10])

[1] 2 4 4 9 4 9 2 4 5 3
```

Example 2.1B: Should Supersize Drinks be Banned?

1. $H_0: \pi = 0.5$
 $H_a: \pi < 0.5$
Test statistic: $\hat{p} = 0.46$ (the sample proportion of 503/1093)
2. We simulate a world in which $\pi = 0.5$:

```
sim.ban <- do(1000) * rflip(1093, 0.5)
head(sim.ban, 3)

  n heads tails prop
1 1093    550    543 0.5032
2 1093    537    556 0.4913
3 1093    556    537 0.5087

favstats(~prop, data = sim.ban)
```

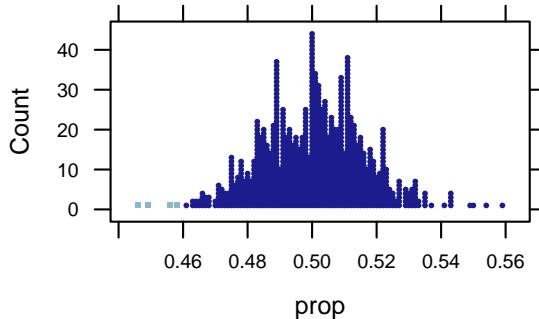
Figure2.3

```

      min      Q1 median      Q3      max      mean      sd      n missing
0.4465 0.4895 0.5014 0.5105 0.559 0.5004 0.01552 1000         0

dotPlot(~prop, data = sim.ban, groups = (prop <= 0.46), width = 0.001)

```



3. Strength of evidence:

```

prop(~(prop <= 0.46), data = sim.ban)

TRUE
0.004

```

Figure2.3b

Normal approximation using predicted standard deviation:

```

sd <- sqrt(0.5 * (1 - 0.5) / 1093)
sd

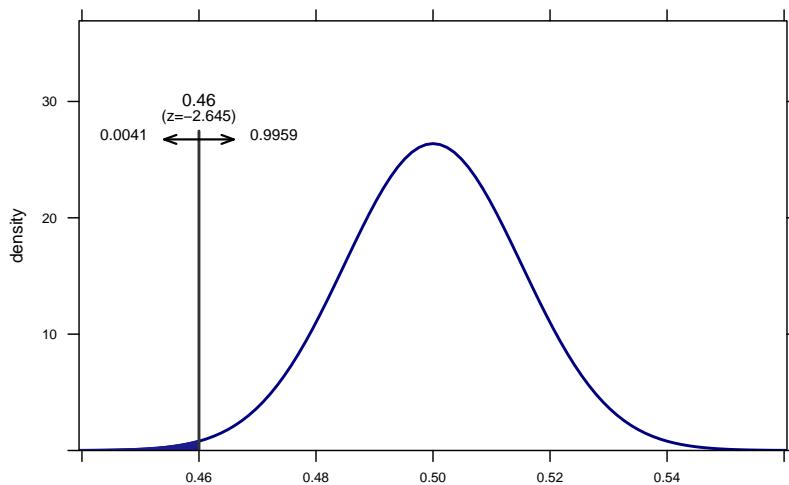
[1] 0.01512

xpnorm(0.46, 0.5, sd)

If X ~ N(0.5, 0.0151237651004726), then
P(X <= 0.46) = P(Z <= -2.645) = 0.0041
P(X > 0.46) = P(Z > -2.645) = 0.9959
[1] 0.004086

```

Figure2.4



Approximate test for proportions with continuity correction:

```
prop.test(503, 1093, alt = "less")
```

Figure 2.4b

```
1-sample proportions test with continuity correction

data: x and n
X-squared = 6.767, df = 1, p-value = 0.004644
alternative hypothesis: true p is less than 0.5
95 percent confidence interval:
0.0000 0.4855
sample estimates:
p
0.4602
```

Exact test for proportions:

```
binom.test(503, 1093, alt = "less")
```

Figure 2.4c

```
Exact binomial test

data: x and n
number of successes = 503, number of trials = 1093, p-value = 0.004628
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
0.0000 0.4855
sample estimates:
probability of success
0.4602
```

Exploration 2.1B: Banning Smoking in Cars?

1. $H_0: \pi = 0.5$
- $H_a: \pi > 0.5$

Test statistic: $\hat{p} = 0.55$ (the sample proportion)

2. We simulate a world in which $\pi = 0.5$:

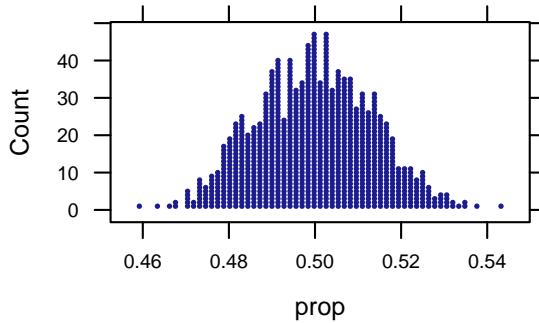
```
Exploration2.1B.10
sim.smoke <- do(1000) * rflip(1421, 0.5)
head(sim.smoke, 3)

  n heads tails prop
1 1421    712   709 0.5011
2 1421    731   690 0.5144
3 1421    753   668 0.5299

favstats(~prop, data = sim.smoke)

  min     Q1 median     Q3   max mean      sd      n missing
0.4588 0.4905 0.5004 0.5095 0.5433 0.5 0.01337 1000         0

dotPlot(~prop, data = sim.smoke, groups = (prop >= 0.55), width = 0.0014)
```



3. Strength of evidence:

```
Exploration2.1B.10b
prop(~(prop >= 0.55), data = sim.smoke)
```

```
TRUE
0
```

Normal approximation using predicted standard deviation:

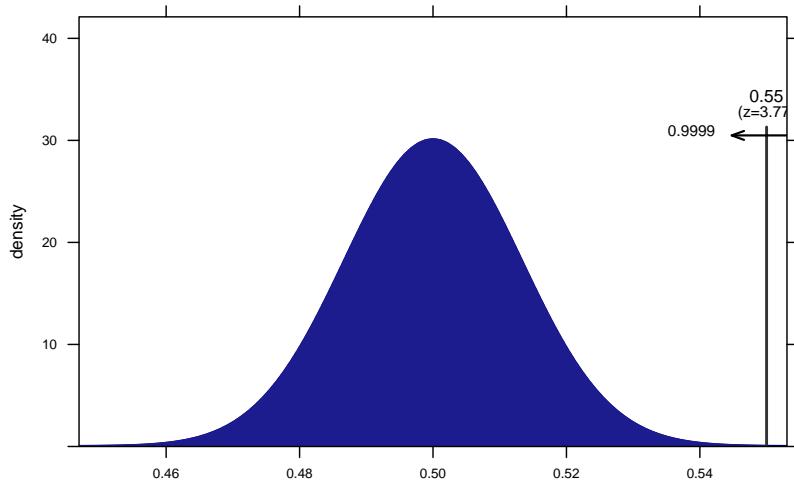
```
Exploration2.1B.14
sd <- sqrt(0.5 * (1 - 0.5) / 1421)
sd

[1] 0.0132639527269323

xpnorm(0.55, 0.5, sd, lower.tail = FALSE)
```

If $X \sim N(0.5, 0.0132639527269323)$, then

```
P(X <= 0.55) = P(Z <= 3.77) = 0.9999
P(X > 0.55) = P(Z > 3.77) = 1e-04
[1] 8.175e-05
```



Approximate test for proportions with continuity correction:

```
prop.test(782, 1421, alt = "greater") # 782 = 1421 * 0.55
```

Exploration2.1B.14b

```
1-sample proportions test with continuity correction

data: x and n
X-squared = 14.19, df = 1, p-value = 8.262e-05
alternative hypothesis: true p is greater than 0.5
95 percent confidence interval:
 0.5282 1.0000
sample estimates:
      p
0.5503
```

Exact test for proportions:

```
binom.test(782, 1421, alt = "greater")
```

Exploration2.1B.14c

```
Exact binomial test

data: x and n
number of successes = 782, number of trials = 1421, p-value = 8.166e-05
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.5282 1.0000
sample estimates:
probability of success
 0.5503
```

2.2 Inference for a Single Quantitative Variable

Example 2.2: Estimating Elapsed Time

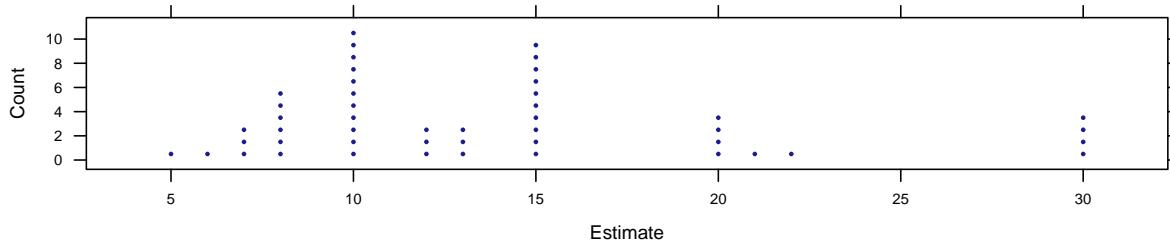
```
head(TimeEstimate)

  Estimate
1      10
2      12
3       6
4      13
5      15
6      10

favstats(~Estimate, data = TimeEstimate)

  min  Q1 median  Q3 max  mean   sd  n missing
  5 10     12 15   30 13.71 6.5 48      0

dotPlot(~Estimate, data = TimeEstimate, width = 1, cex = 0.5)
```



```
TimeEstimate %>% mutate(Rank = rank(Estimate, ties.method = "random")) %>% arrange(Rank)
```

Table2.5

	Estimate	Rank
1	5	1
2	6	2
3	7	3
4	7	4
5	7	5
6	8	6
7	8	7
8	8	8
9	8	9
10	8	10
11	8	11
12	10	12
13	10	13
14	10	14
15	10	15
16	10	16
17	10	17
18	10	18
19	10	19
20	10	20
21	10	21
22	10	22

```
23      12  23
24      12  24
25      12  25
26      13  26
27      13  27
28      13  28
29      15  29
30      15  30
31      15  31
32      15  32
33      15  33
34      15  34
35      15  35
36      15  36
37      15  37
38      15  38
39      20  39
40      20  40
41      20  41
42      20  42
43      21  43
44      22  44
45      30  45
46      30  46
47      30  47
48      30  48
```

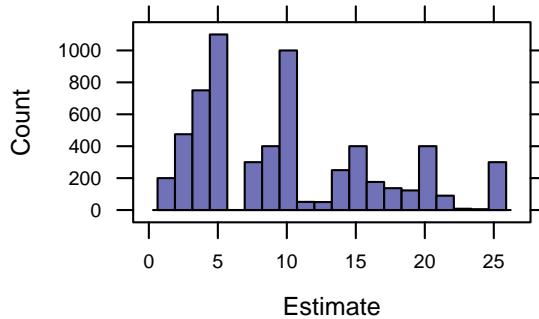
```
head(TimePopulation, 3) Figure2.6

Estimate
1      5
2      8
3      2

favstats(~Estimate, data = TimePopulation)

min  Q1 median  Q3 max  mean    sd    n missing
1   5     9  15  25   10  6.49  6215      0

histogram(~Estimate, data = TimePopulation, type = "count", nint = 20)
```



```
sample1 <- sample(TimePopulation, 48)
head(sample1, 3)
```

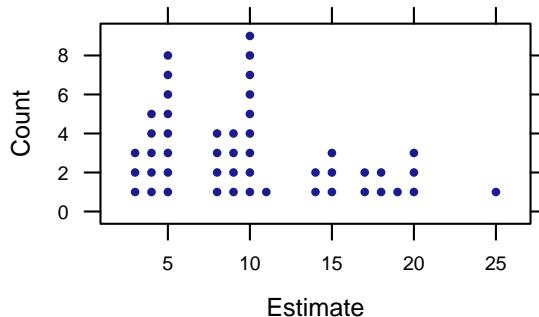
Figure 2.7

```
Estimate orig.ids
2198      5    2198
4116     18    4116
2718      4    2718
```

```
favstats(~Estimate, data = sample1)
```

```
min Q1 median      Q3 max mean      sd n missing
 3   5     9.5 14.25 25 10.1 5.624 48       0
```

```
dotPlot(~Estimate, data = sample1, width = 1, cex = 0.3)
```



1. $H_0: \mu = 10$

$H_a: \mu \neq 10$

Test statistic: $\bar{x} = 13.71$ (the sample mean)

2. We simulate random samples from a finite population:

```
sim.time <- do(1000) * mean(~Estimate, data = sample(TimePopulation, 48))
head(sim.time, 3)
```

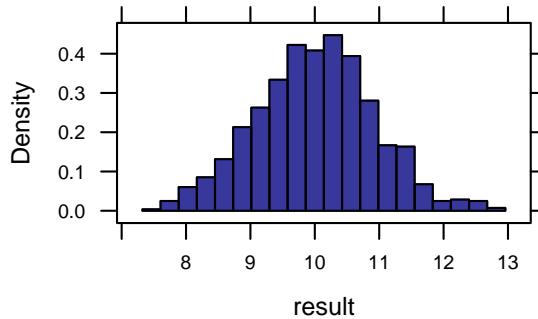
Figure 2.8

```

  result
1 9.750
2 9.667
3 9.375

histogram(~result, data = sim.time, groups = (result <= 6.29 | result >= 13.71), nint = 20,
          center = 10)

```



3. Strength of evidence:

```

prop(~(result <= 6.29 | result >= 13.71), data = sim.time)

TRUE
0

```

Figure2.8b

Strength of evidence with the standardized statistic:

```

mean(~result, data = sim.time)

[1] 10.01

sd <- sd(~result, data = sim.time)
sd

[1] 0.9219

pnorm(13.71, 10, sd, lower.tail = FALSE, plot = FALSE)

```

Figure2.8c

If $X \sim N(10, 0.9219)$, then

```

P(X <= 13.71) = P(Z <= 4.024) = 1
P(X > 13.71) = P(Z > 4.024) = 0
[1] 2.857e-05

```

Theory-based approach: One-sample t-test

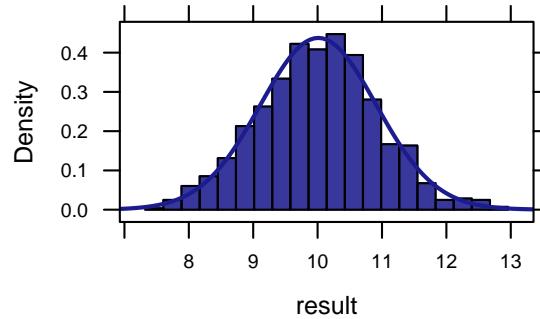
```
xbar <- 13.71
mu <- 10
s <- 6.5
n <- 48
t <- (xbar - mu)/(s/sqrt(n))
t
```

Example2.2

```
[1] 3.954
```

```
histogram(~result, data = sim.time, groups = (result <= 6.29 | result >= 13.71), nint = 20,
          center = 10, fit = "t")
```

Figure2.9



```
2 * pt(t, df = 47, lower.tail = FALSE)
```

Figure2.10

```
[1] 0.0002571
```

Alternative Analysis: What about the median?

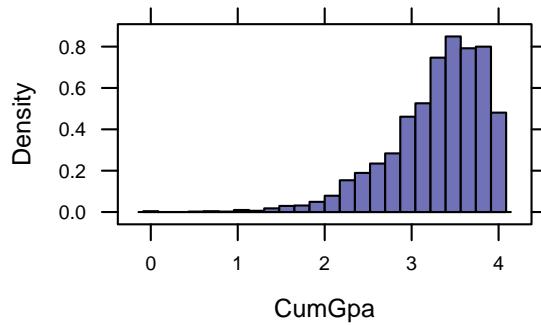
```
sim.median <- do(1000) * median(~Estimate, data = sample(TimePopulation, 48))
head(sim.median, 3)
```

Figure2.11

```
result
1      10
2      9
3      9
```

```
histogram(~result, data = sim.median, groups = (result < 8 | result > 12), width = 0.5, type = "count")
prop(~(result < 8 | result > 12), data = sim.median)
```

```
TRUE
0.117
```



Exploration 2.2: Sleepless Nights?

```
head(SleepTimes, 3)
```

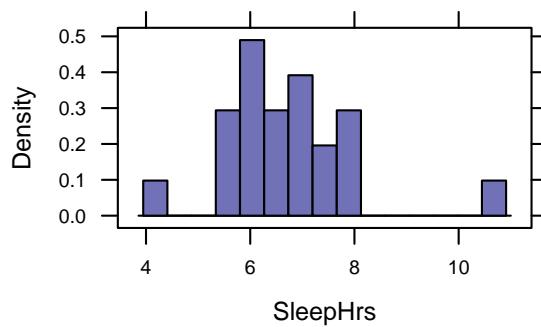
Exploration2.2.1

	SleepHrs
1	7.0
2	5.5
3	8.0

Shape

```
histogram(~SleepHrs, data = SleepTimes, nint = 15)
```

Exploration2.2.10



Center

```
mean(~SleepHrs, data = SleepTimes)
```

Exploration2.2.11

```
[1] 6.705
```

```
median(~SleepHrs, data = SleepTimes)
```

[1] 6.5

Exploration2.2.16

Variability

```
sd(~SleepHrs, data = SleepTimes)
```

[1] 1.297

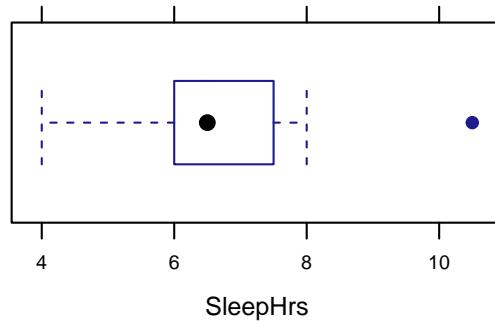
Exploration2.2.18

Unusual observations

We could examine the entire data set to find any outliers but there is a quicker way to see if there potential outliers. The `bwplot()` function plots a box-and-whisker plot which identifies *possible* outliers with a dot beyond the whiskers.

```
bwplot(~SleepHrs, data = SleepTimes)
```

Exploration2.2.20



Instead of using the hypothetical population provided in the applet, we can create our own hypothetical population by assigning a variable (`SleepHrs`) a random normal distribution (`rnorm()`) of count (18000), mean (8 hrs), and standard deviation (1.5 hrs). Additionally, let's round each value to the nearest hundredth (2) using `round()`

```
Pop1 <- data.frame(SleepHrs = round(rnorm(18000, 8, 1.5), 2))  
head(Pop1)
```

Exploration2.2.24

	SleepHrs
1	9.22
2	6.94
3	11.05

```

4      9.15
5      7.73
6     10.19

favstats(SleepHrs, data = Pop1)

min   Q1 median   Q3   max   mean      sd      n missing
1.75 6.98    7.99 9.01 13.96 7.994 1.494 18000       0

```

```

mean(~SleepHrs, data = SleepTimes) # test statistic

[1] 6.705

sim.pop1 <- do(1000) * mean(~SleepHrs, data = sample(Pop1, 48))

```

Exploration2.2.25

Loading required package: parallel

```
head(sim.pop1, 3)
```

```

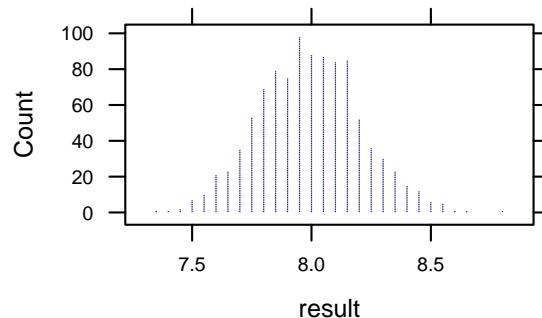
result
1 7.785
2 8.297
3 7.777

dotPlot(~result, data = sim.pop1, width = 0.05)
favstats(~result, data = sim.pop1)

min   Q1 median   Q3   max   mean      sd      n missing
7.335 7.846    7.99 8.136 8.795 7.992 0.2128 1000       0

```

Exploration2.2.26



```
prop(~(result <= 6.705), data = sim.pop1)
```

Exploration2.2.26

```
TRUE
0
```

```
sd <- sd(~result, data = sim.pop1)
xpnorm(6.705, 8, sd, plot = FALSE)
```

Exploration2.2.27

If $X \sim N(8, 0.212790941098959)$, then

```
P(X <= 6.705) = P(Z <= -6.086) = 0
P(X > 6.705) = P(Z > -6.086) = 1
[1] 5.796e-10
```

```
t <- (6.705 - 8)/(1.5/sqrt(48))
t
```

Exploration2.2.30

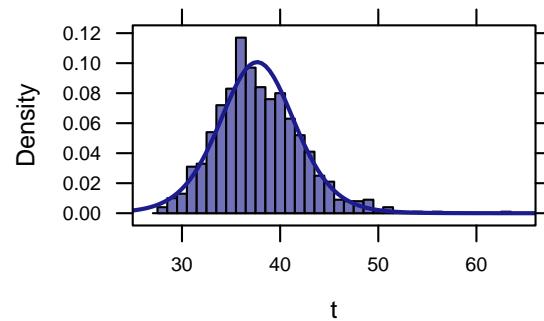
```
[1] -5.981
```

```
sim.t <- do(1000) * stat(t.test(~SleepHrs, data = sample(Pop1, 48)))
head(sim.t, 3)
```

Exploration2.2.33

```
t
1 30.43
2 41.43
3 42.29
```

```
histogram(~t, data = sim.t, width = 1, fit = "t")
```



```
prop(~(t <= 5.981), data = sim.t)
```

Exploration2.2.34

```
TRUE
0
```

```
t.test(~SleepHrs, data = Pop1)
```

Exploration2.2.35

One Sample t-test

```
data: data$SleepHrs
t = 717.7, df = 17999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 7.972 8.016
sample estimates:
mean of x
 7.994
```

Follow-up # 1

```
head(Pop)
```

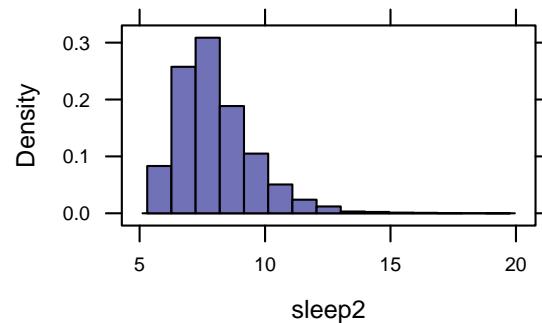
Exploration2.2.40

```
sleep1 sleep2 sleep3
1   6.50   8.50   5.00
2   6.00  10.00   4.75
3   6.00   6.75   2.75
4   6.75  10.00   4.50
5   9.00   7.75  14.00
6   7.75   7.00  10.25
```

```
favstats(~sleep2, data = Pop)
```

```
min Q1 median   Q3   max   mean      sd      n missing
 6    7    7.75  8.75 19.5  7.999  1.501  18000       0
```

```
histogram(~sleep2, data = Pop)
```



```
mean(~SleepHrs, data = SleepTimes) # test statistic
```

Exploration2.2.40b

```
[1] 6.705

sim.pop2 <- do(1000) * mean(~sleep2, data = sample(Pop, 48))
head(sim.pop2, 3)

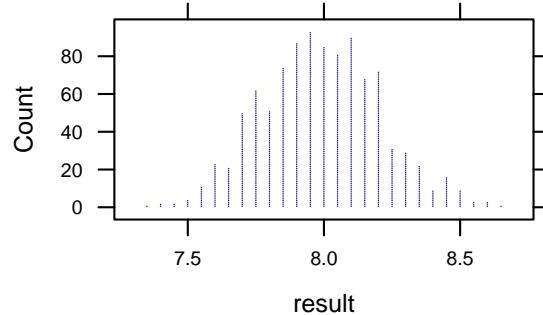
result
1 7.938
2 8.005
3 7.901

dotPlot(~result, data = sim.pop2, width = 0.05)
favstats(~result, data = sim.pop2)

min      Q1 median      Q3    max   mean      sd      n missing
7.344 7.844 7.987 8.141 8.641 7.991 0.2156 1000         0

prop(~(result <= 6.705), data = sim.pop2)

TRUE
0
```



```
t.test(~sleep2, data = Pop)
```

Exploration2.2.41

One Sample t-test

```
data: data$sleep2
t = 715, df = 17999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 7.978 8.021
sample estimates:
mean of x
 7.999
```

Follow-up # 2

```
median(~SleepHrs, data = SleepTimes) # test statistic
[1] 6.5

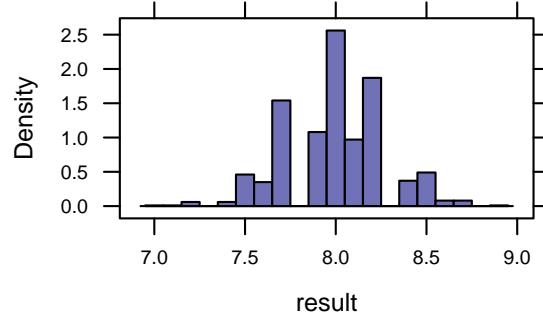
sim.pop1med <- do(1000) * median(~sleep1, data = sample(Pop, 48))
head(sim.pop1med, 3)

result
1 7.875
2 8.125
3 7.750

histogram(~result, data = sim.pop1med, width = 0.1)
prop(~(result <= 6.5), data = sim.pop1med)

TRUE
0
```

Exploration2.2.46



2.3 Errors and Significance

Exploration 2.3: Parapsychology Studies

1. $H_0: \pi = 0.25$

$$H_a: \pi > 0.25$$

Test statistic: $\hat{p} = 0.333$ (the sample proportion of 709/2124)

2. We simulate a world in which $\pi = 0.25$:

```
sim.esp <- do(1000) * rflip(2124, 0.25)
head(sim.esp, 3)

   n heads tails prop
1 2124    507   1617 0.2387
2 2124    539   1585 0.2538
3 2124    528   1596 0.2486
```

Exploration2.3.4

3. Strength of evidence:

```
prop(~(prop >= 0.333), data = sim.esp)

TRUE
0
```

Exploration2.3.4b

Approximate test for proportions:

```
prop.test(709, 2124, p = 0.25, alt = "greater")

1-sample proportions test with continuity correction

data: x and n
X-squared = 79.11, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is greater than 0.25
95 percent confidence interval:
 0.317 1.000
sample estimates:
      p
0.3338
```

Exploration2.3.5

Approximate test for $\hat{p} = 15/50$ if $\pi = 0.25$:

```
prop.test(15, 50, p = 0.25, alt = "greater")
```

Exploration2.3.12

```
1-sample proportions test with continuity correction

data: x and n
X-squared = 0.4267, df = 1, p-value = 0.2568
alternative hypothesis: true p is greater than 0.25
95 percent confidence interval:
 0.1974 1.0000
sample estimates:
      p
0.3
```

Approximate test for $\hat{p} = 15/50$ if $\pi = 0.33$:

```
prop.test(15, 50, p = 0.33, alt = "greater")
```

Exploration2.3.16

```
1-sample proportions test with continuity correction

data: x and n
X-squared = 0.0905, df = 1, p-value = 0.6182
alternative hypothesis: true p is greater than 0.33
95 percent confidence interval:
```

```
0.1974 1.0000
sample estimates:
  p
0.3
```


3

Estimation: How Large is the Effect?

3.1 Statistical Inference - Confidence Intervals

Example 3.1: Can Dogs Sniff Out Cancer?

1. $H_0: \pi = 0.20$

$H_a: \pi > 0.20$

Test statistic: $\hat{p} = 0.909$ (the sample proportion of 30/33)

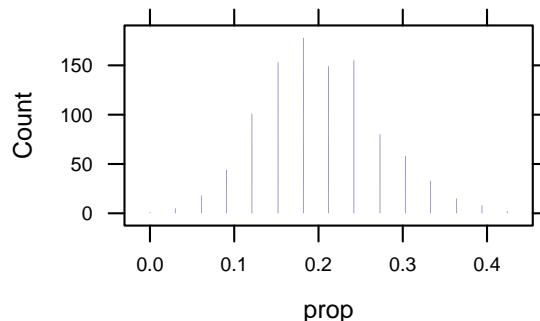
2. We simulate a world in which $\pi = 0.20$:

```
sim.cancer <- do(1000) * rflip(33, 0.2)
head(sim.cancer, 3)

  n heads tails   prop
1 33     8    25 0.2424
2 33     6    27 0.1818
3 33     4    29 0.1212

dotPlot(~prop, data = sim.cancer, groups = (prop >= 0.909), width = 0.001)
```

Figure3.1



3. Strength of evidence:

```
favstats(~prop, data = sim.cancer)

min      Q1 median      Q3    max   mean      sd     n missing
0 0.1515 0.197 0.2424 0.4242 0.2021 0.06952 1000       0

prop(~(prop >= 0.909), data = sim.cancer)

TRUE
0
```

Figure3.1b

$$1. H_0: \pi = 0.70$$

$$H_a: \pi \neq 0.70$$

Test statistic: $\hat{p} = 0.909$ (the sample proportion of 30/33)

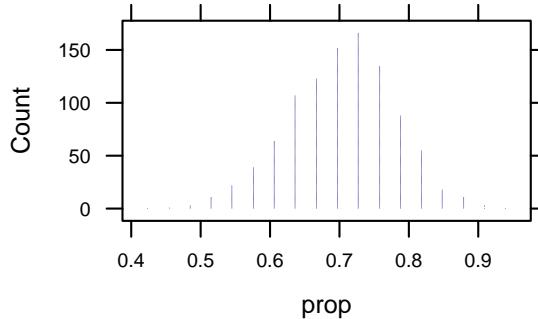
$$2. \text{ We simulate a world in which } \pi = 0.70:$$

```
sim.cancer2 <- do(1000) * rflip(33, 0.7)
head(sim.cancer2, 3)

  n heads tails prop
1 33    22    11 0.66667
2 33    23    10 0.69700
3 33    20    13 0.60611

dotPlot(~prop, data = sim.cancer2, groups = (prop <= 0.4545 | prop >= 0.909), width = 0.001)
```

Figure3.2



$$3. \text{ Strength of evidence:}$$

```
favstats(~prop, data = sim.cancer2)

min      Q1 median      Q3    max   mean      sd     n missing
0.4242 0.66667 0.697 0.7576 0.9394 0.7032 0.07659 1000       0

prop(~(prop <= 0.4545 | prop >= 0.909), data = sim.cancer2)

TRUE
0.005
```

Figure3.2b

1. $H_0: \pi = 0.80$

$H_a: \pi \neq 0.80$

Test statistic: $\hat{p} = 0.909$ (the sample proportion of 30/33)

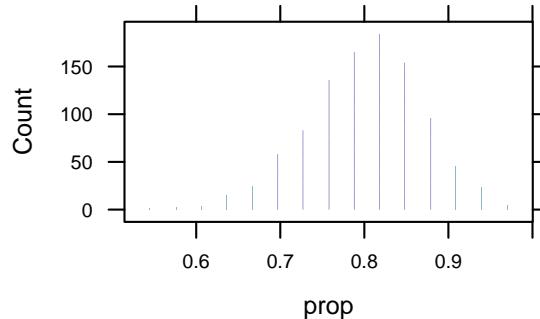
2. We simulate a world in which $\pi = 0.80$:

```
sim.cancer3 <- do(1000) * rflip(33, 0.8)
head(sim.cancer3, 3)

  n heads tails  prop
1 33     27      6 0.8182
2 33     28      5 0.8485
3 33     24      9 0.7273

dotPlot(~prop, data = sim.cancer3, groups = (prop <= 0.691 | prop >= 0.909), width = 0.001)
```

Figure3.3



3. Strength of evidence:

```
favstats(~prop, data = sim.cancer3)

  min      Q1 median      Q3    max   mean      sd    n missing
0.5455  0.7576  0.8182  0.8485  0.9697  0.8001  0.06915 1000        0

prop(~(prop <= 0.6667 | prop >= 0.909), data = sim.cancer3)

  TRUE
0.124
```

Figure3.3b

Results of testing different values of probabilities under the null hypothesis:

```
pval(binom.test(30, 33, p = 0.93))

p.value
0.5007

pval(binom.test(30, 33, p = 0.94))
```

Table3.1

```

p.value
0.4474

pval(binom.test(30, 33, p = 0.95))

p.value
0.2272

pval(binom.test(30, 33, p = 0.96))

p.value
0.1442

pval(binom.test(30, 33, p = 0.97))

p.value
0.07564

pval(binom.test(30, 33, p = 0.98))

p.value
0.02793

pval(binom.test(30, 33, p = 0.99))

p.value
0.00436

```

Exploration 3.1: Kissing Right?

1. $H_0: \pi = 0.5$

$$H_a: \pi > 0.5$$

Test statistic: $\hat{p} = 0.645$ (the sample proportion of 80/124)

2. We simulate a world in which $\pi = 0.5$:

```

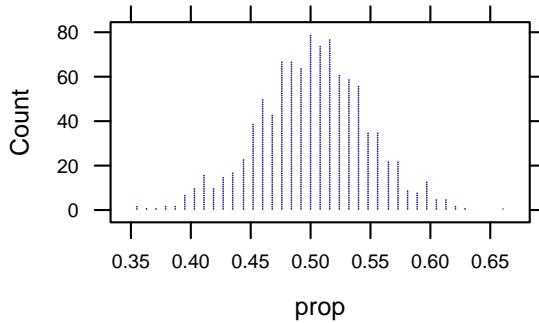
sim.kiss <- do(1000) * rflip(124, 0.5)
head(sim.kiss, 3)

   n heads tails prop
1 124     65    59 0.5242
2 124     55    69 0.4435
3 124     74    50 0.5968

dotPlot(~prop, data = sim.kiss, groups = (prop >= 0.645), width = 0.001)

```

Exploration3.1.7



3. Strength of evidence:

```
favstats(~prop, data = sim.kiss)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	0.3548	0.4758	0.5	0.5323	0.6613	0.5015	0.0452	1000	0

```
prop(~(prop >= 0.645), data = sim.kiss)
```

	TRUE	0.001
--	------	-------

Exploration3.1.7b

Approximate test for proportions:

```
prop.test(80, 124, alt = "greater")
```

1-sample proportions test with continuity correction

data: x and n
X-squared = 9.879, df = 1, p-value = 0.0008359
alternative hypothesis: true p is greater than 0.5
95 percent confidence interval:
0.568 1.000
sample estimates:
p
0.6452

Exploration3.1.7c

Exact test for proportions:

```
binom.test(80, 124, alt = "greater")
```

Exact binomial test

data: x and n
number of successes = 80, number of trials = 124, p-value = 0.0007824
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
0.5684 1.0000
sample estimates:
probability of success
0.6452

Exploration3.1.7d

1. $H_0: \pi = 0.6$

$H_a: \pi \neq 0.6$

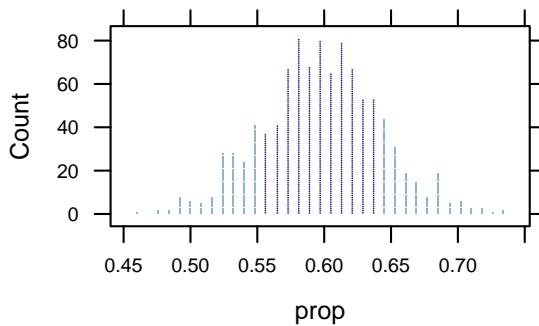
Test statistic: $\hat{p} = 0.645$ (the sample proportion of 80/124)

2. We simulate a world in which $\pi = 0.6$:

```
Exploration3.1.8a
sim.kiss2 <- do(1000) * rflip(124, 0.6)
head(sim.kiss2, 3)

  n heads tails  prop
1 124     75     49 0.6048
2 124     75     49 0.6048
3 124     67     57 0.5403

dotPlot(~prop, data = sim.kiss2, groups = (prop <= 0.555 | prop >= 0.645), width = 0.001)
```



3. Strength of evidence:

```
Exploration3.1.8b
favstats(~prop, data = sim.kiss2)

  min      Q1 median      Q3      max      mean       sd      n missing
0.4597  0.5726  0.5968  0.629  0.7339  0.5982  0.04392  1000        0

prop(~(prop <= 0.555 | prop >= 0.645), data = sim.kiss2)

  TRUE
0.309
```

Approximate test for proportions:

```
Exploration3.1.8c
prop.test(80, 124, p = 0.6)

 1-sample proportions test with continuity correction

data: x and n
X-squared = 0.874, df = 1, p-value = 0.3499
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
 0.5536 0.7276
```

```
sample estimates:  
    p  
0.6452
```

Exact test for proportions:

```
binom.test(80, 124, p = 0.6)
```

Exploration3.1.8d

Exact binomial test

```
data: x and n  
number of successes = 80, number of trials = 124, p-value = 0.3151  
alternative hypothesis: true probability of success is not equal to 0.6  
95 percent confidence interval:  
 0.5542 0.7290  
sample estimates:  
probability of success  
 0.6452
```

```
pval(binom.test(80, 124, p = 0.54))
```

Exploration3.1.11

```
p.value  
0.01915
```

```
pval(binom.test(80, 124, p = 0.55))
```

```
p.value  
0.03757
```

```
pval(binom.test(80, 124, p = 0.56))
```

```
p.value  
0.05778
```

```
pval(binom.test(80, 124, p = 0.57))
```

```
p.value  
0.1024
```

```
pval(binom.test(80, 124, p = 0.58))
```

```
p.value  
0.1465
```

```
pval(binom.test(80, 124, p = 0.59))
```

```
p.value  
0.2355
```

```
pval(binom.test(80, 124, p = 0.6))
```

```
p.value  
0.3151
```

```
pval(binom.test(80, 124, p = 0.7))
```

Exploration3.1.11b

```
p.value  
0.2024
```

```
pval(binom.test(80, 124, p = 0.71))
```

```
p.value  
0.114
```

```
pval(binom.test(80, 124, p = 0.72))
```

```
p.value  
0.07146
```

```
pval(binom.test(80, 124, p = 0.73))
```

```
p.value  
0.04242
```

```
pval(binom.test(80, 124, p = 0.74))
```

```
p.value  
0.0185
```

```
pval(binom.test(80, 124, p = 0.75))
```

```
p.value  
0.009269
```

```
pval(binom.test(80, 124, p = 0.76))
```

```
p.value  
0.004281
```

```
confint(binom.test(80, 124, p = 0.6))
```

Exploration3.1.13

probability of success	lower	upper
0.6452	0.5542	0.7290
level		
0.9500		

```
confint(binom.test(80, 124, p = 0.6, conf.level = 0.99))
```

Exploration3.1.15

probability of success	lower	upper
0.6452	0.5265	0.7524
level		
0.9900		

3.2 2SD and Theory-Based Confidence Intervals for a Single Proportion

Example 3.2: The Affordable Care Act

An easy way to find a confidence interval in R is to use `prop.test()` or `binom.test()` which by default calculates a 95% confidence interval in its results.

```
binom.test(713, 1034) # 713 = 1034 * 0.69
```

Example3.2

```
Exact binomial test

data: x and n
number of successes = 713, number of trials = 1034, p-value < 2.2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.6604 0.7177
sample estimates:
probability of success
 0.6896
```

Theory-Based Approach

```
xpnorm(c(-1.645, 1.645), 0, 1)
```

Figure3.6

If $X \sim N(0,1)$, then

```
P(X <= -1.645) = P(Z <= -1.645) = 0.05
P(X <= 1.645) = P(Z <= 1.645) = 0.95
P(X > -1.645) = P(Z > -1.645) = 0.95
P(X > 1.645) = P(Z > 1.645) = 0.05
[1] 0.04998 0.95002
```

```
xpnorm(c(-1.96, 1.96), 0, 1)
```

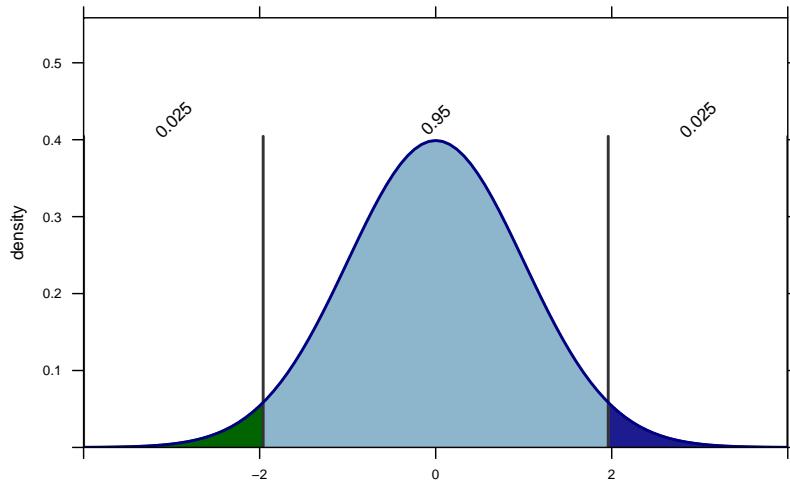
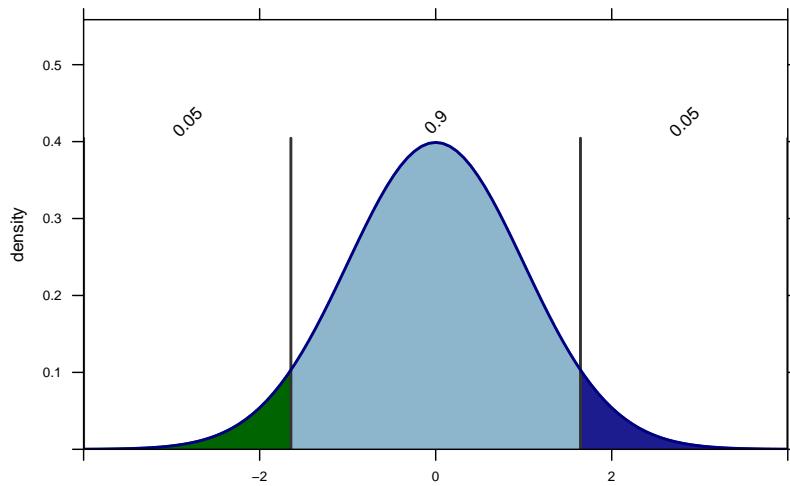
If $X \sim N(0,1)$, then

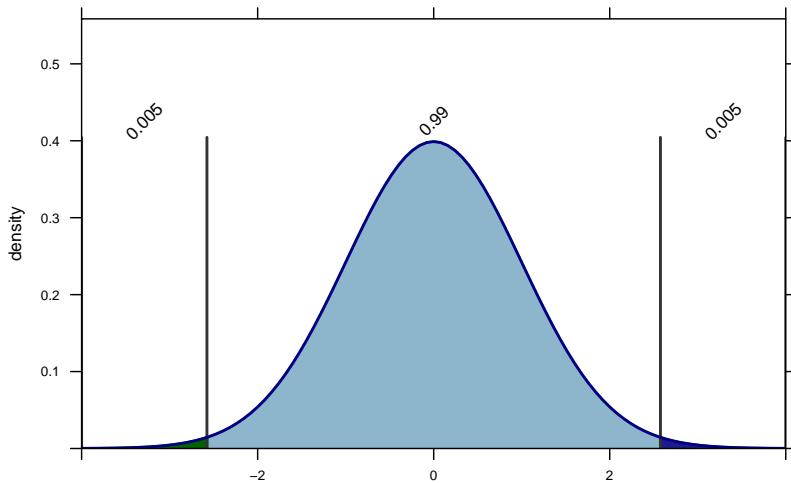
```
P(X <= -1.96) = P(Z <= -1.96) = 0.025
P(X <= 1.96) = P(Z <= 1.96) = 0.975
P(X > -1.96) = P(Z > -1.96) = 0.975
P(X > 1.96) = P(Z > 1.96) = 0.025
[1] 0.025 0.975
```

```
xpnorm(c(-2.576, 2.576), 0, 1)
```

If $X \sim N(0,1)$, then

```
P(X <= -2.576) = P(Z <= -2.576) = 0.005
P(X <= 2.576) = P(Z <= 2.576) = 0.995
P(X > -2.576) = P(Z > -2.576) = 0.995
P(X > 2.576) = P(Z > 2.576) = 0.005
[1] 0.004998 0.995002
```





Using 2SD method and standard error of the observed sample proportion (Theory-Based Inference applet):

```
n <- 1034
p.hat <- 0.69; p.hat
# 0.69 = 713 / 1034

[1] 0.69

SE <- sqrt( p.hat * (1 - p.hat) / n ) # standard error
MoE <- 1.96 * SE; MoE
# margin of error

[1] 0.02819

p.hat - MoE
# lower limit of 95% CI

[1] 0.6618

p.hat + MoE
# upper limit of 95% CI

[1] 0.7182
```

Figure3.7

Exploration 3.2: American Exceptionalism

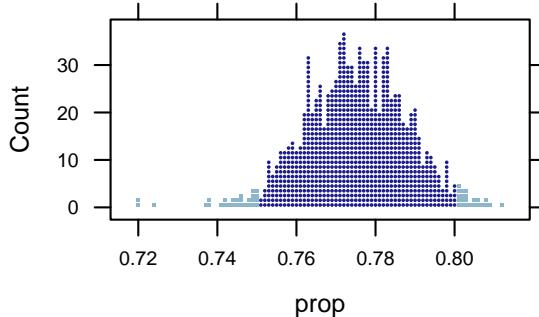
1. $H_0: \pi = 0.775$
 $H_a: \pi \neq 0.775$
Test statistic: $\hat{p} = 0.80$ (the sample proportion of 85/1019)
2. We simulate a world in which $\pi = 0.775$:

```
sim.amer <- do(1000) * rflip(1019, 0.775)
head(sim.amer, 3)
```

Exploration3.2.6

```
n heads tails prop
1 1019 823 196 0.8077
2 1019 794 225 0.7792
3 1019 793 226 0.7782
```

```
dotPlot(~prop, data = sim.amer, groups = (prop <= 0.75 | prop >= 0.8), width = 0.001)
```



3. Strength of evidence:

```
favstats(~prop, data = sim.amer)
Exploration3.2.6b
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	0.7203	0.7664	0.7753	0.7841	0.8116	0.7751	0.01292	1000	0

```
prop(~(prop <= 0.75 | prop >= 0.8), data = sim.amer)
Exploration3.2.6c
```

	TRUE
	0.051

Approximate test for proportions:

```
prop.test(815, 1019, p = 0.775)
Exploration3.2.6d
```

1-sample proportions test with continuity correction

data: x and n
X-squared = 3.454, df = 1, p-value = 0.06308
alternative hypothesis: true p is not equal to 0.775
95 percent confidence interval:
0.7736 0.8237
sample estimates:
p
0.7998

Exact test for proportions:

```
binom.test(815, 1019, p = 0.775)
```

```
Exact binomial test

data: x and n
number of successes = 815, number of trials = 1019, p-value = 0.06064
alternative hypothesis: true probability of success is not equal to 0.775
95 percent confidence interval:
 0.7739 0.8240
sample estimates:
probability of success
              0.7998
```

$$1. H_0: \pi = 0.5$$

$$H_a: \pi \neq 0.5$$

Test statistic: $\hat{p} = 0.80$ (the sample proportion of 815/1019)

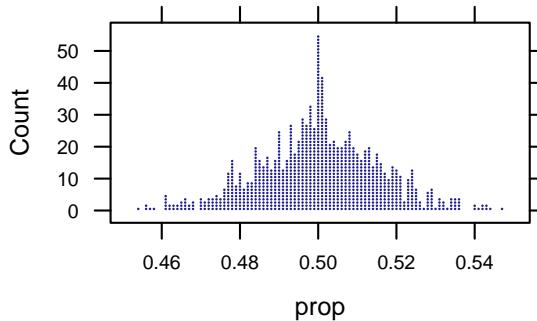
$$2. \text{ We simulate a world in which } \pi = 0.5:$$

```
Exploration3.2.8

sim.amer2 <- do(1000) * rflip(1019, 0.5)
head(sim.amer2, 3)

   n heads tails prop
1 1019    531    488 0.5211
2 1019    522    497 0.5123
3 1019    517    502 0.5074

dotPlot(~prop, data = sim.amer2, groups = (prop <= 0.2 | prop >= 0.8), width = 0.001)
```



$$3. \text{ Strength of evidence:}$$

```
Exploration3.2.8b

favstats(~prop, data = sim.amer2)

   min      Q1 median      Q3     max   mean      sd     n missing
0.4544 0.4897 0.5005 0.5103 0.5466 0.5002 0.01553 1000        0

prop(~(prop <= 0.2 | prop >= 0.8), data = sim.amer2)

TRUE
 0
```

Approximate test for proportions:

```
prop.test(815, 1019)
```

Exploration3.2.8c

1-sample proportions test with continuity correction

data: x and n
X-squared = 365.2, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
0.7736 0.8237
sample estimates:
p
0.7998

Exact test for proportions:

```
binom.test(815, 1019)
```

Exploration3.2.8d

Exact binomial test

data: x and n
number of successes = 815, number of trials = 1019, p-value < 2.2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
0.7739 0.8240
sample estimates:
probability of success
0.7998

Finding the standard deviation using simulated deviation:

```
sd <- sd(~prop, data = sim.amer)
sd
```

Exploration3.2.9

[1] 0.01292

```
z <- (0.8 - 0.775)/sd
z
```

[1] 1.936

```
xpnorm(0.8, 0.775, sd, lower.tail = FALSE, plot = FALSE)
```

If $X \sim N(0.775, 0.0129157552720886)$, then

$P(X \leq 0.8) = P(Z \leq 1.936) = 0.9735$
 $P(X > 0.8) = P(Z > 1.936) = 0.0265$
[1] 0.02646

Determining a 95% confidence interval using the 2SD Method and standard deviation of the null distribution:

```

p.hat <- 0.80          # given sample proportion
sd           # previously found simulated standard deviation

[1] 0.01292

MoE <- 2 * sd; MoE    # margin of error for 95% CI

[1] 0.02583

p.hat - MoE          # lower limit of 95% CI

[1] 0.7742

p.hat + MoE          # upper limit of 95% CI

[1] 0.8258

```

Exploration3.2.11

Determining a 95% confidence interval using the 2SD Method and standard error of the observed sample proportion:

```

n <- 1019
p.hat <- 0.80          # given sample proportion
SE <- sqrt(p.hat * (1 - p.hat) / n); SE

[1] 0.01253

MoE <- 2 * SE; MoE    # margin of error for 95% CI

[1] 0.02506

p.hat - MoE          # lower limit of 95% CI

[1] 0.7749

p.hat + MoE          # upper limit of 95% CI

[1] 0.8251

```

Exploration3.2.12

Determining a 95% confidence interval using more accurate multipliers and standard error of the observed sample proportion (Theory-Based Inference applet):

```

n <- 1019
p.hat <- 0.80           # given sample proportion
SE <- sqrt(p.hat * (1 - p.hat) / n); SE

[1] 0.01253

MoE <- 1.96 * SE; MoE # margin of error for 95% CI with more accurate multiplier

[1] 0.02456

p.hat - MoE             # lower limit of 95% CI

[1] 0.7754

p.hat + MoE             # upper limit of 95% CI

[1] 0.8246

```

Exploration3.2.13

Another way to create a 95% confidence interval is to use the middle 95% of the simulated null distribution. This is not exactly the same as the interval found by the 2SD Method, but it is very close.

```

cdata(0.95, prop, data = sim.amer)

      low      hi central.p
0.7498  0.7998  0.9500

```

Exploration3.2.13b

The `binom.test()` calculates the exact confidence interval for any confidence level:

```

binom.test(815, 1019, p = 0.775, conf.level = 0.95)

Exact binomial test

data: x and n
number of successes = 815, number of trials = 1019, p-value = 0.06064
alternative hypothesis: true probability of success is not equal to 0.775
95 percent confidence interval:
0.7739 0.8240
sample estimates:
probability of success
0.7998

```

Exploration3.2.13c

```
binom.test(815, 1019, p = 0.775, conf.level = 0.99)
```

```
Exact binomial test
```

```

data: x and n
number of successes = 815, number of trials = 1019, p-value = 0.06064
alternative hypothesis: true probability of success is not equal to 0.775
99 percent confidence interval:
0.7656 0.8311
sample estimates:
probability of success
0.7998

binom.test(815, 1019, p = 0.775, conf.level = 0.9)

Exact binomial test

data: x and n
number of successes = 815, number of trials = 1019, p-value = 0.06064
alternative hypothesis: true probability of success is not equal to 0.775
90 percent confidence interval:
0.7781 0.8203
sample estimates:
probability of success
0.7998

```

Note that the specified π , the $p = 0.775$, only matters in calculating the p-value and does not affect the confidence interval.

3.3 2SD and Theory-Based Confidence Intervals for a Single Mean

Example 3.3: Used Cars

```

head(UsedCars)                                         Figure3.9

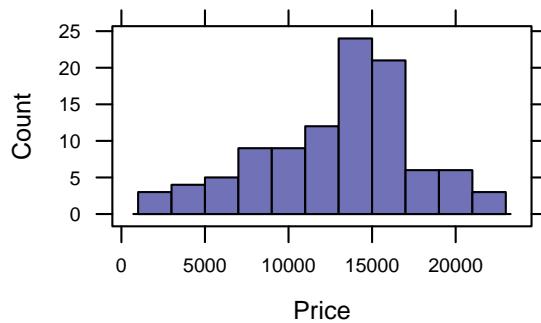
Price
1 21990
2 21990
3 21987
4 20955
5 20955
6 19995

favstats(~Price, data = UsedCars)

min     Q1 median     Q3   max   mean    sd    n missing
1200 10067 13992 15999 21990 13292 4535 102        0

histogram(~Price, data = UsedCars, type = "count", width = 2000)

```



Determining a 95% confidence interval using the 2SD Method and standard error of the sample population:

```
n <- nrow(UsedCars); n
[1] 102

mean <- mean(~ Price, data = UsedCars); mean
[1] 13292

sd <- sd(~ Price, data = UsedCars); sd
[1] 4535

SE <- sd / sqrt(n)
MoE <- 2 * SE; MoE      # margin of error for 95% CI
[1] 898

mean - MoE                  # lower limit of 95% CI
[1] 12394

mean + MoE                  # upper limit of 95% CI
[1] 14190
```

Example3.3

Theory-based approach

```
confint(t.test(~Price, data = UsedCars))
mean of x      lower      upper      level
13292.33    12401.66   14183.01      0.95
```

Figure3.10

```
confint(t.test(~Price, data = UsedCars, conf.level = 0.9))

mean of x      lower      upper      level
13292.3     12547.0    14037.7        0.9

confint(t.test(~Price, data = UsedCars, conf.level = 0.99))

mean of x      lower      upper      level
13292.33   12113.56   14471.10        0.99
```

Figure3.11

Exploration 3.3: Sleepless Nights? (continued)

```
head(SleepTimes)

SleepHrs
1      7.0
2      5.5
3      8.0
4      7.0
5      7.5
6      6.0

favstats(~SleepHrs, data = SleepTimes)

min Q1 median      Q3  max  mean      sd  n missing
 4   6     6.5  7.375 10.5  6.705  1.297  22       0
```

Exploration3.3.1

Determining a 95% confidence interval using the 2SD Method and standard error of the sample population:

```
n <- nrow(SleepTimes); n

[1] 22

mean <- mean(~ SleepHrs, data = SleepTimes); mean

[1] 6.705

sd <- sd(~ SleepHrs, data = SleepTimes); sd

[1] 1.297

SE <- sd / sqrt(n)
MoE <- 2 * SE; MoE      # margin of error for 95% CI
```

Exploration3.3.6

```
[1] 0.5531
mean - MoE           # lower limit of 95% CI
[1] 6.151
mean + MoE           # upper limit of 95% CI
[1] 7.258
```

Theory-based approach

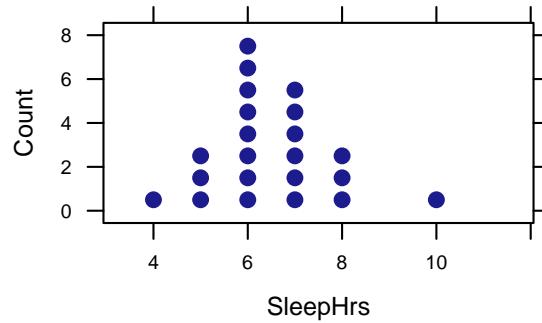
```
confint(t.test(~SleepHrs, data = SleepTimes))
```

Exploration3.3.8

mean of x	lower	upper	level
6.705	6.129	7.280	0.950

```
dotPlot(~SleepHrs, data = SleepTimes, width = 1) # to check the distribution
```

Exploration3.3.9



3.4 Factors That Affect the Width of a Confidence Interval

Example 3.4: The Affordable Care Act (continued)

```
confint(binom.test(713, 1034, conf.level = 0.9)) # 1034 * 0.69 = 713
```

Table3.5

probability of success	lower	upper
0.6896	0.6650	0.7133
level		
0.9000		

```
confint(binom.test(713, 1034, conf.level = 0.95))
```

probability of success	lower	upper
0.6896	0.6604	0.7177
level		
0.9500		

```
confint(binom.test(713, 1034, conf.level = 0.99))
```

probability of success	lower	upper
0.6896	0.6512	0.7262
level		
0.9900		

Sample size

```
confint(binom.test(70, 100))
```

Figure3.12

probability of success	lower	upper
0.7000	0.6002	0.7876
level		
0.9500		

```
confint(binom.test(140, 200))
```

probability of success	lower	upper
0.7000	0.6314	0.7626
level		
0.9500		

```
confint(binom.test(280, 400))
```

probability of success	lower	upper
0.7000	0.6525	0.7445
level		
0.9500		

Optional: Effect of sample proportion

Sample proportions will affect confidence intervals calculated by using accurate multipliers and the standard error of the observed sample proportion (Theory-Based Inference applet). However, the sample proportions will not affect confidence intervals found by using the exact test for proportions, `binom.test()`.

```
confint(binom.test(838, 1034))
```

Figure3.13

probability of success	lower	upper
------------------------	-------	-------

```

0.8104          0.7852          0.8339
  level
0.9500

MoE838 <- 0.8339078 - 0.7852004
MoE838

[1] 0.04871

confint(binom.test(196, 1034))

probability of success           lower           upper
0.1896                  0.1661          0.2148
  level
0.9500

```

MoE196 <- 0.2147996 - 0.1660922
MoE196

[1] 0.04871

Exploration 3.4: Holiday Spending Habits

Determining a 95% confidence interval using the 2SD Method and standard error of the sample population:

```

n <- 1039
mean <- 704
sd <- 150
SE <- sd / sqrt(n)
MoE <- 2 * SE; MoE      # margin of error for 95% CI

[1] 9.307

mean - MoE      # lower limit of 95% CI

[1] 694.7

mean + MoE      # upper limit of 95% CI

[1] 713.3

```

```

n <- 1039
mean <- 704
sd <- 300
SE <- sd / sqrt(n)
MoE <- 2 * SE; MoE      # margin of error for 95% CI

```

```
[1] 18.61

mean - MoE          # lower limit of 95% CI

[1] 685.4

mean + MoE          # upper limit of 95% CI

[1] 722.6
```

The impact of sample size

```
n <- 477
mean <- 704
sd <- 300
SE <- sd / sqrt(n)
MoE <- 2 * SE; MoE      # margin of error for 95% CI

[1] 27.47

mean - MoE          # lower limit of 95% CI

[1] 676.5

mean + MoE          # upper limit of 95% CI

[1] 731.5
```

Exploration3.4.B.4

Exploration 3.4B: Reese's Pieces

Simulate 1 sample proportion and calculate the 95% confidence interval:

```
sample.CI <- CI.sim(100, samples = 1, rdist = rbinom, args = list(size = 1, prob = 0.5), method = binom.test,
method.args = list(success = 1), verbose = FALSE, estimand = 0.5)
sample.CI

lower upper estimate cover sample
1 0.3886 0.592     0.49    Yes      1
```

Exploration3.4B.5

Simulate 100 sample proportions and calculate the 95% confidence intervals:

```
sim.CI <- CI.sim(100, samples = 100, rdist = rbinom, args = list(size = 1, prob = 0.5), method = binom.test,
method.args = list(success = 1), verbose = FALSE, estimand = 0.5)
```

Exploration3.4.B.5

Proportion of intervals produced that do not contain $\pi = 0.5$:

```
prop(~cover, data = sim.CI)
```

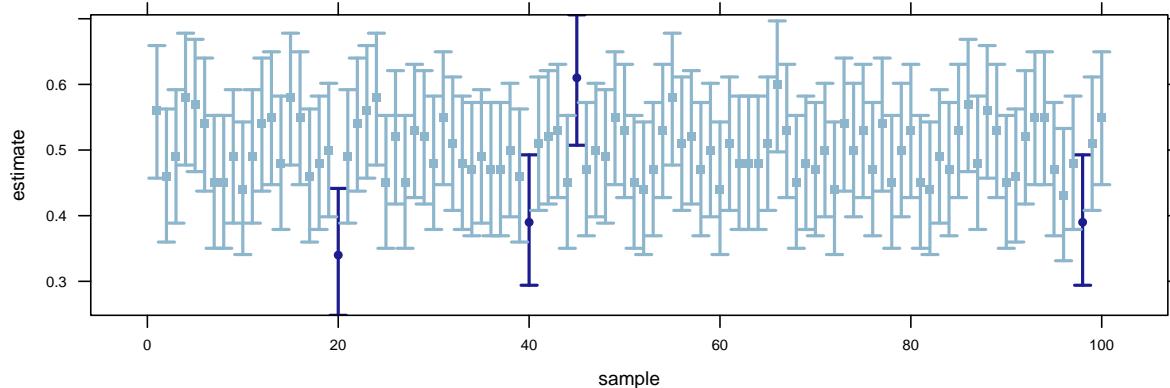
Exploration3.4B.5b

```
No  
0.04
```

Plot the 95% confidence intervals of the simulation of 100 sample proportions:

```
require(Hmisc)  
xYplot(Cbind(estimate, lower, upper) ~ sample, data = sim.CI, par.settings = col.mosaic(),  
       groups = cover)
```

Exploration3.4B.5c



Simulate 1000 sample proportions and calculate the 95% confidence intervals:

```
sim.CI2 <- CISim(100, samples = 1000, rdist = rbinom,  
                  args = list(size = 1, prob = 0.5), method = binom.test,  
                  method.args = list(success = 1), verbose = FALSE,  
                  estimand = 0.5)
```

Exploration3.4B.5d

Proportion of intervals produced that do not contain $\pi = 0.5$:

```
prop(~cover, data = sim.CI2)
```

Exploration3.4B.5e

```
No  
0.037
```

Simulate 1000 sample proportions and calculate the 90% confidence intervals:

```
sim.CI3 <- CISim(100, samples = 1000, rdist = rbinom,
```

Exploration3.4B.6

```
args = list(size = 1, prob = 0.5), conf.level = 0.90,
method = binom.test, method.args = list(success = 1),
verbose = FALSE, estimand = 0.5)
```

Proportion of intervals produced that do not contain $\pi = 0.5$:

```
prop(~cover, data = sim.CI3)
```

Exploration3.4B.6b

No
0.095

Simulate 1000 sample proportions and calculate the 90% confidence intervals (sample size = 400):

```
sim.CI4 <- CI(sim(400, samples = 100, rdist = rbinom,
                  args = list(size = 1, prob = 0.5), conf.level = 0.90,
                  method = binom.test, method.args = list(success = 1),
                  verbose = FALSE, estimand = 0.5))
```

Exploration3.4B.6c

Proportion of intervals produced that do not contain $\pi = 0.5$:

```
prop(~cover, data = sim.CI4)
```

Exploration3.4B.6d

No
0.14

3.5 Cautions When Conducting Inference

1. $H_0: \pi = 0.3645$

$H_a: \pi > 0.3645$

Test statistic: $\hat{p} = 0.41$ (the sample proportion)

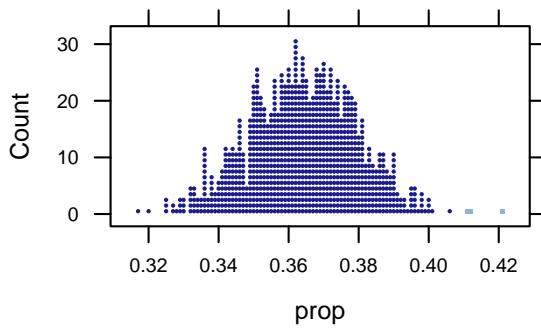
2. We simulate a world in which $\pi = 0.3645$:

```
sim.obama <- do(1000) * rflip(1000, 0.3645)
head(sim.obama, 3)
```

Figure3.14

	n	heads	tails	prop
1	1000	371	629	0.371
2	1000	358	642	0.358
3	1000	347	653	0.347

```
dotPlot(~prop, data = sim.obama, groups = (prop >= 0.41), width = 0.001)
```



3. Strength of evidence:

```
favstats(~prop, data = sim.obama)

min      Q1 median      Q3    max    mean      sd     n missing
0.317  0.353  0.364  0.375  0.421  0.3641  0.01531  1000        0

prop(~(prop >= 0.41), data = sim.obama)

TRUE
0.003
```

Figure3.14b

Exploration 3.5A: Voting for President

Finding the 99% confidence interval using the exact test for proportions:

```
confint(binom.test(1783, 2613, conf.level = 0.99))

probability of success              lower              upper
                  0.6824            0.6584            0.7057
level
                  0.9900
```

Exploration3.5A.3

Another famous case of problems in Presidential election polling

Finding the 99% confidence interval using the exact test for proportions:

```
confint(binom.test(1368000, 2400000, conf.level = 0.999)) # 1368000 = 2400000 * 0.57

probability of success              lower              upper
                  0.5700            0.5689            0.5711
level
                  0.9990
```

Exploration3.5A.9

Example 3.5B: Parapsychology Studies (continued)

```
confint(binom.test(709, 2124, conf.level = 0.95))
```

Example3.5B

probability of success	lower	upper
0.3338	0.3138	0.3543
level		
0.9500		

```
confint(binom.test(709, 2124, conf.level = 0.99))
```

probability of success	lower	upper
0.3338	0.3076	0.3607
level		
0.9900		

- $H_0: \pi = 0.25$

$$H_a: \pi > 0.25$$

Test statistic: $\hat{p} = 0.38$ (the sample proportion of 19/50)

- We simulate a world in which $\pi = 0.25$:

```
sim.esp2 <- do(10000) * rflip(50, 0.25)
```

Figure3.15

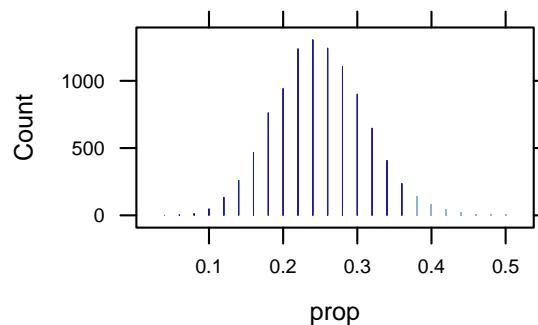
Loading required package: parallel

```
head(sim.esp2, 3)
```

	n	heads	tails	prop
1	50	12	38	0.24
2	50	14	36	0.28
3	50	18	32	0.36

```
dotPlot(~prop, data = sim.esp2, groups = (prop >= 0.38), width = 0.01, cex = 10)
prop(~(prop >= 0.38), data = sim.esp2)
```

```
TRUE
0.0281
```



1. $H_0: \pi = 1/3$

$H_a: \pi > 1/3$

Test statistic: $\hat{p} = 0.38$ (the sample proportion of 19/50)

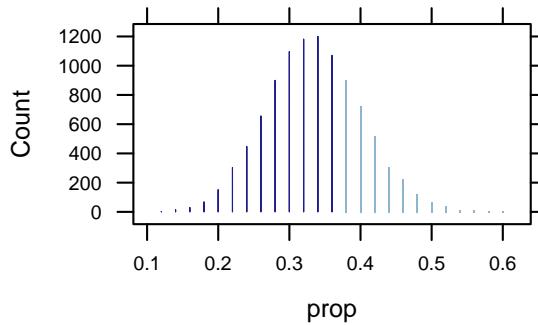
2. We simulate a world in which $\pi = 1/3$:

```
sim.esp3 <- do(10000) * rflip(50, 1/3)
head(sim.esp3, 3)

  n heads tails prop
1 50     13     37 0.26
2 50     13     37 0.26
3 50     14     36 0.28

dotPlot(~prop, data = sim.esp3, groups = (prop >= 0.38), width = 0.01, cex = 10)
prop(~(prop >= 0.38), data = sim.esp3)

TRUE
0.2873
```



1. $H_0: \pi = 1/2$

$H_a: \pi > 1/2$

Test statistic: $\hat{p} = 0.38$ (the sample proportion of 19/50)

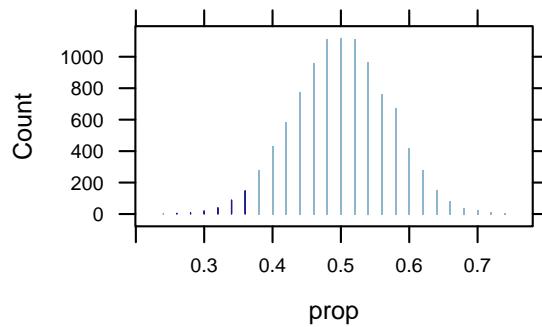
2. We simulate a world in which $\pi = 1/2$:

```
sim.esp4 <- do(10000) * rflip(50, 1/2)
head(sim.esp4, 3)

  n heads tails prop
1 50     22     28 0.44
2 50     21     29 0.42
3 50     28     22 0.56

dotPlot(~prop, data = sim.esp4, groups = (prop >= 0.38), width = 0.01, cex = 10)
prop(~(prop >= 0.38), data = sim.esp4)

TRUE
0.9688
```



3.5.1 Exploration 3.5B: Cat Households

1. $H_0: \pi = 1/3$

$$H_a: \pi < 1/3$$

Test statistic: $\hat{p} = 0.324$ (the sample proportion of 15228/47000)

2. Exact test for proportions:

```
binom.test(15228, 47000, p = 1/3, conf.level = 0.999, alt = "less")
```

Exploration3.5B.3

Exact binomial test

```
data: x and n
number of successes = 15228, number of trials = 47000, p-value = 8.654e-06
alternative hypothesis: true probability of success is less than 0.3333
99.9 percent confidence interval:
 0.0000 0.3307
sample estimates:
probability of success
 0.324
```

```
binom.test(15228, 47000, p = 1/3, alt = "less")
```

Exact binomial test

```
data: x and n
number of successes = 15228, number of trials = 47000, p-value = 8.654e-06
alternative hypothesis: true probability of success is less than 0.3333
95 percent confidence interval:
 0.0000 0.3276
sample estimates:
probability of success
 0.324
```

Exploration3.5B.9

3. We simulate a world in which $\pi = 1/3$:

```
sim.pets <- do(1000) * rflip(100, 1/3)
head(sim.pets, 3)
```

	n	heads	tails	prop
1	100	35	65	0.35
2	100	37	63	0.37
3	100	34	66	0.34

We could use trial-and-error to determine values of the sample proportion that would produce a p-value of less than 0.05. R can quickly calculate try possible values that would result in the significance level of 0.05 but we can also have R calculate them for us.

```
cdata(0.95, prop, data = sim.pets)
```

Exploration3.5B.9b

low	hi	central.p
0.24	0.43	0.95

- $H_0: \pi = 0.30$

$$H_a: \pi < 0.30$$

Test statistic: $\hat{p} = 0.243$ (the sample proportion)

- We simulate a world in which $\pi = 0.30$:

```
sim.pets2 <- do(1000) * rflip(100, 0.3)
head(sim.pets2, 3)
```

Exploration3.5B.11

	n	heads	tails	prop
1	100	30	70	0.30
2	100	30	70	0.30
3	100	24	76	0.24

```
prop(~(prop <= 0.243), data = sim.pets2)
```

```
TRUE  
0.101
```

```
cdata(0.9, prop, data = sim.pets2)
```

Exploration3.5B.11b

low	hi	central.p
0.23	0.38	0.90

```
confint(binom.test(33, 100, p = 1/3))
```

probability of success	lower	upper
0.3300	0.2392	0.4312
level		
0.9500		

```
binom.test(24, 100, p = 0.3, alt = "less")
```

```
Exact binomial test
```

```
data: x and n
number of successes = 24, number of trials = 100, p-value = 0.1136
alternative hypothesis: true probability of success is less than 0.3
95 percent confidence interval:
 0.0000 0.3206
sample estimates:
probability of success
 0.24
```

```
confint(binom.test(33, 100, p = 1/3, conf.level = 0.9))
```

probability of success	lower	upper
0.3300	0.2523	0.4155
level		
0.9000		

```
binom.test(25, 100, p = 0.3, alt = "less", conf.level = 0.9)
```

```
Exact binomial test
```

```
data: x and n
number of successes = 25, number of trials = 100, p-value = 0.1631
alternative hypothesis: true probability of success is less than 0.3
90 percent confidence interval:
 0.000 0.314
sample estimates:
probability of success
 0.25
```

```
confint(binom.test(167, 500, p = 1/3))
```

probability of success	lower	upper
0.3340	0.2927	0.3772
level		
0.9500		

```
binom.test(146, 500, p = 0.3, alt = "less")
```

```
Exact binomial test
```

```
data: x and n
number of successes = 146, number of trials = 500, p-value = 0.3685
alternative hypothesis: true probability of success is less than 0.3
95 percent confidence interval:
 0.0000 0.3273
sample estimates:
probability of success
 0.292
```

```
confint(binom.test(33, 100, p = 1/3))
```

```
probability of success           lower           upper
      0.3300                  0.2392          0.4312
      level
      0.9500
```

```
binom.test(24, 100, p = 0.2, alt = "less")
```

```
Exact binomial test
```

```
data: x and n
number of successes = 24, number of trials = 100, p-value = 0.8686
alternative hypothesis: true probability of success is less than 0.2
95 percent confidence interval:
 0.0000 0.3206
sample estimates:
probability of success
      0.24
```

4

Causation: Can We Say What Caused the Effect?

4.1 Association and Confounding

Example 4.1: Night Lights and Near-Sightedness

Often, when a dataset has only categorical variables, it may come in the form of a table and not a frame.

Here is a way to create a data frame in R.

```
NightLight1
```

	Darkness	NightLight	RoomLight
Near	18	78	41
Not	154	154	34

```
NightLight <- rbind(
  do(18) * data.frame(light = "Darkness", nearsight = "Near"),
  do(154) * data.frame(light = "Darkness", nearsight = "Not"),
  do(78) * data.frame(light = "NightLight", nearsight = "Near"),
  do(154) * data.frame(light = "NightLight", nearsight = "Not"),
  do(41) * data.frame(light = "RoomLight", nearsight = "Near"),
  do(34) * data.frame(light = "RoomLight", nearsight = "Not")
)
```

```
head(NightLight)
```

	light	nearsight	.row	.index
1	Darkness	Near	1	1
2	Darkness	Near	1	2
3	Darkness	Near	1	3
4	Darkness	Near	1	4
5	Darkness	Near	1	5
6	Darkness	Near	1	6

```
tally(nearsight ~ light, data = NightLight)
```

Table4.1

```

light
nearsight Darkness NightLight RoomLight
Near    0.1047    0.3362    0.5467
Not     0.8953    0.6638    0.4533

tally(~nearsight | light, data = NightLight)

light
nearsight Darkness NightLight RoomLight
Near    0.1047    0.3362    0.5467
Not     0.8953    0.6638    0.4533

tally(~nearsight + light, data = NightLight, margins = TRUE)

light
nearsight Darkness NightLight RoomLight Total
Near        18       78       41    137
Not       154      154       34    342
Total      172      232       75    479

```

4.2 Observational studies versus experiments

Exploration 4.2: Have a Nice Trip

```

sim <- do(2) * rflip(12, 16/24)
sim

  n heads tails  prop
1 12      8      4 0.6667
2 12      7      5 0.5833

```

5

Comparing Two Proportions

5.1 Comparing Two Groups: Categorical Response

Example 5.1: Good and Bad Perceptions

```
head(GoodandBad, 30)
```

Table5.1

```
Wording Perception
1 goodyear positive
2 goodyear negative
3 badyear positive
4 goodyear positive
5 goodyear negative
6 badyear positive
7 goodyear positive
8 goodyear positive
9 goodyear positive
10 badyear negative
11 goodyear negative
12 badyear negative
13 goodyear positive
14 badyear negative
15 goodyear positive
16 goodyear positive
17 badyear positive
18 goodyear positive
19 goodyear positive
20 goodyear positive
21 badyear negative
22 goodyear positive
23 badyear negative
24 goodyear positive
25 badyear negative
26 goodyear positive
27 badyear negative
28 goodyear positive
29 badyear positive
30 badyear negative
```

Table5.2

```
tally(~Perception + Wording, data = GoodandBad, margins = TRUE)
```

		Wording		Total
Perception	badyear	goodyear		
		negative	8	3
positive	4	15	19	
Total	12	18	30	

```
tally(Perception ~ Wording, data = GoodandBad)
```

		Wording		Perception
badyear	goodyear	negative	positive	
		0.6667	0.1667	
0.3333	0.8333			

```
prop(Perception ~ Wording, data = GoodandBad)
```

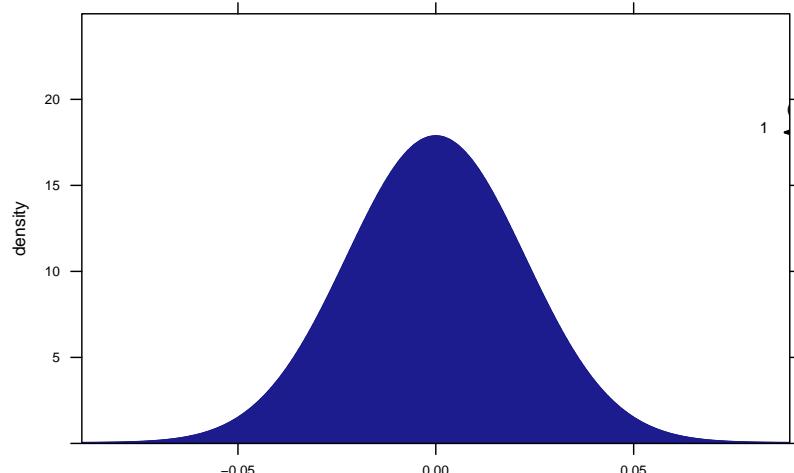
negative.badyear	negative.goodyear
0.6667	0.1667

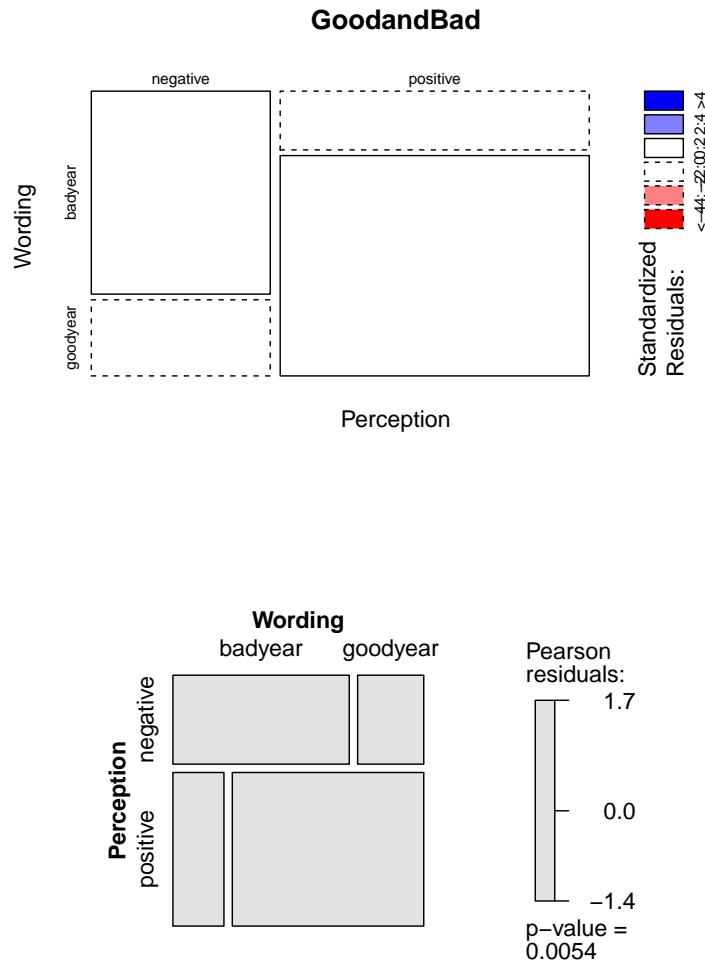
```
prop(Perception ~ Wording, level = "positive", data = GoodandBad)
```

positive.badyear	positive.goodyear
0.3333	0.8333

Figure5.1

```
bargraph(~Perception, groups = Wording, data = GoodandBad, stack = TRUE, auto.key = TRUE)
mosaicplot(~Perception + Wording, data = GoodandBad, shade = TRUE)
mosaic(~Perception + Wording, data = GoodandBad, shade = TRUE)
```





Summarizing the data

Exploration 5.1: Murderous Nurse?

```
Nurse <- rbind(
  do(40) * data.frame(patient = "Death", shift = "Gilbert"),
  do(34) * data.frame(patient = "Death", shift = "NoGilbert"),
  do(217) * data.frame(patient = "NoDeath", shift = "Gilbert"),
  do(1350) * data.frame(patient = "NoDeath", shift = "NoGilbert")
)
```

Exploration5.1.7

```
tally(~patient + shift, data = Nurse, margins = TRUE)
```

Exploration5.1.7b

shift

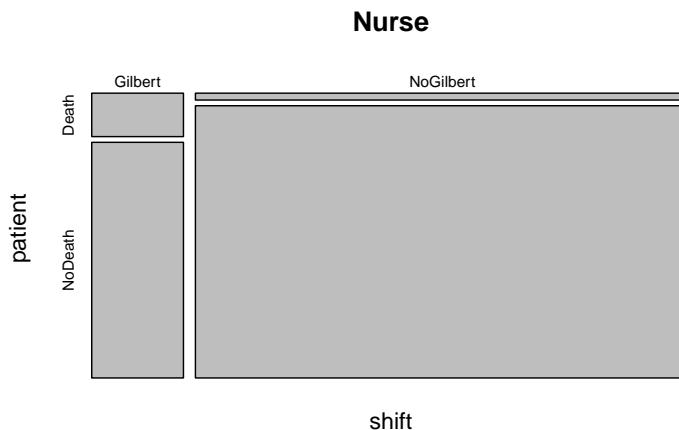
patient	Gilbert	NoGilbert	Total
Death	40	34	74
NoDeath	217	1350	1567
Total	257	1384	1641

```
tally(patient ~ shift, data = Nurse) # conditional prop
```

shift	patient	Gilbert	NoGilbert
Death	0.15564	0.02457	
NoDeath	0.84436	0.97543	

```
mosaicplot(shift ~ patient, data = Nurse)
```

Exploration5.1.10



```
prop(patient ~ shift, data = Nurse)
```

Exploration5.1.14

```
Death.Gilbert Death.NoGilbert
0.15564      0.02457
```

```
diff(prop(patient ~ shift, data = Nurse))
```

```
Death.NoGilbert
-0.1311
```

Further Analysis

```
Nurse2 <- rbind(
```

Exploration5.1.18

```
do(100) * data.frame(patient = "Death", shift = "Gilbert"),
do(357) * data.frame(patient = "Death", shift = "NoGilbert"),
do(157) * data.frame(patient = "NoDeath", shift = "Gilbert"),
do(1027) * data.frame(patient = "NoDeath", shift = "NoGilbert")
)
```

```
tally(~patient + shift, data = Nurse2, margin = TRUE)
```

Exploration 5.1.18b

patient	shift		Total
	Gilbert	NoGilbert	
Death	100	357	457
NoDeath	157	1027	1184
Total	257	1384	1641

```
tally(patient ~ shift, data = Nurse2)
```

patient	shift	
	Gilbert	NoGilbert
Death	0.3891	0.2579
NoDeath	0.6109	0.7421

```
diff(prop(patient ~ shift, data = Nurse2)) # diff in conditional prop
```

```
Death.NoGilbert
-0.1312
```

5.2 Comparing Two Properties: Simulation-Based Approach

Example 5.2: Swimming with Dolphins

```
head(Dolphin)
```

Table5.3

	Swimming	Response
1	Dolphin	Improve
2	Dolphin	Improve
3	Dolphin	Improve
4	Dolphin	Improve
5	Dolphin	Improve
6	Dolphin	Improve

```
tally(~Response + Swimming, data = Dolphin, margin = TRUE)
```

Response	Swimming		Total
	Control	Dolphin	
Improve	3	10	13
Not Improve	12	5	17
Total	15	15	30

```
tally(Response ~ Swimming, data = Dolphin)
```

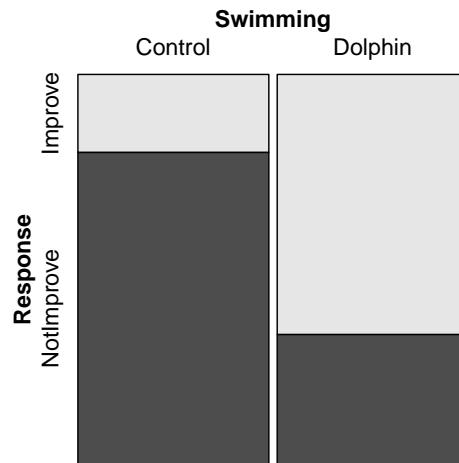
Response	Swimming	
	Control	Dolphin
Improve	0.2000	0.6667
NotImprove	0.8000	0.3333

```
diff(prop(Response ~ Swimming, data = Dolphin))
```

```
Improve.Dolphin  
0.4667
```

```
mosaic(Response ~ Swimming, data = Dolphin, dir = "v")
```

Figure5.2



```
mosaic(shuffle(Response) ~ Swimming, data = Dolphin, dir = "v")
```

Figure5.4

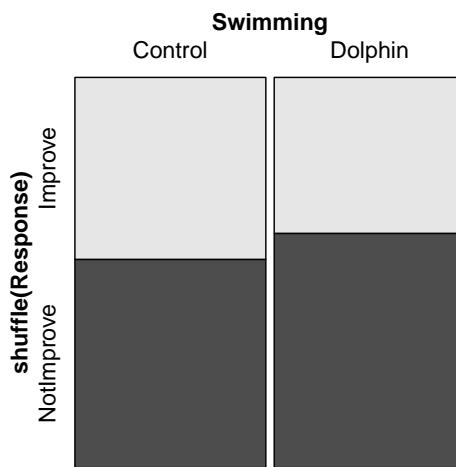


Figure 5.5

```
tally(~shuffle(Response) + Swimming, data = Dolphin, margins = TRUE)
```

		Swimming	
		Control	Dolphin
shuffle(Response)	Improve	7	6
	Not Improve	8	9
Total	15	15	30

```
tally(~shuffle(Response) + Swimming, data = Dolphin, margins = TRUE)
```

		Swimming	
		Control	Dolphin
shuffle(Response)	Improve	5	8
	Not Improve	10	7
Total	15	15	30

```
tally(~shuffle(Response) + Swimming, data = Dolphin, margins = TRUE)
```

		Swimming	
		Control	Dolphin
shuffle(Response)	Improve	8	5
	Not Improve	7	10
Total	15	15	30

```
diff(prop(Response ~ Swimming, data = Dolphin))
```

```
Improve.Dolphin
0.4667
```

```
diff(prop(shuffle(Response) ~ Swimming, data = Dolphin))
```

```
Improve.Dolphin
0.2
```

$$1. H_0: \pi_{dolphins} - \pi_{control} = 0$$

$$H_a: \pi_{dolphins} - \pi_{control} > 0$$

Test statistic: $\hat{p}_{dolphins} - \hat{p}_{control} = 0.4667$ (the difference in the conditional sample proportions)

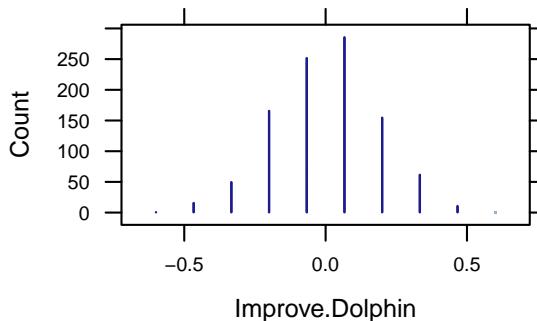
2. We simulate a world in which $\pi_{dolphins} - \pi_{control} = 0$:

```
sim.dol <- do(1000) * diff(prop(shuffle(Response) ~ Swimming, data = Dolphin))
head(sim.dol, 3)

Improve.Dolphin
1      0.2000
2     -0.2000
3      0.3333

dotPlot(~Improve.Dolphin, data = sim.dol, groups = (Improve.Dolphin >= 0.4667), width = 1/15,
       cex = 5)
```

Figure 5.6



3. Strength of evidence:

```
favstats(~Improve.Dolphin, data = sim.dol)

min      Q1   median      Q3   max      mean      sd      n missing
-0.6 -0.06667  0.06667  0.06667  0.6  0.001733  0.1852  1000        0

prop(~(Improve.Dolphin >= 0.4667), data = sim.dol)

TRUE
0.001
```

Figure 5.6b

Approximate test for difference in proportions:

```
prop.test(Response ~ Swimming, data = Dolphin)

2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 4.887, df = 1, p-value = 0.02706
alternative hypothesis: two.sided
95 percent confidence interval:
```

Figure 5.6c

```
-0.84620 -0.08713
sample estimates:
prop 1 prop 2
0.2000 0.6667
```

Estimation

Determining a 95% confidence interval using the 2SD Method and simulated standard deviation of the null distribution:

```
# given difference in sample proportions
diff <- diff(prop(Response ~ Swimming, data = Dolphin))
# simulated standard deviation
sd <- sd(~Improve.Dolphin, data = sim.dol)
# margin of error for 95% CI
MoE <- 2 * sd
MoE
```

```
[1] 0.3704
```

```
# lower limit of 95% CI
diff - MoE
```

```
Improve.Dolphin
0.09622
```

```
# upper limit of 95% CI
diff + MoE
```

```
Improve.Dolphin
0.8371
```

Example5.2

Determining a 95% confidence interval using the approximate test for proportions:

```
confint(prop.test(Response ~ Swimming, data = Dolphin))
```

prop 1	prop 2	lower	upper	level
0.20000	0.66667	-0.84620	-0.08713	0.95000

Example5.2b

Follow-up Analysis

```
Dolphin2 <- rbind(
  do(8) * data.frame(Response = "Improve", Swimming = "Control"),
  do(5) * data.frame(Response = "Improve", Swimming = "Dolphin"),
  do(7) * data.frame(Response = "NotImprove", Swimming = "Control"),
```

Figure5.7

```
do(10) * data.frame(Response = "NotImprove", Swimming = "Dolphin")
)
```

```
tally(~Response + Swimming, data = Dolphin2, margin = TRUE)
```

Figure5.7b

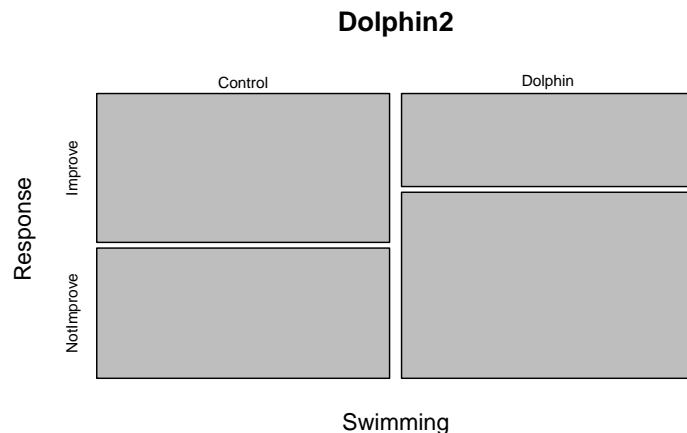
Response	Swimming		Total
	Control	Dolphin	
Improve	8	5	13
Not Improve	7	10	17
Total	15	15	30

```
diff(prop(Response ~ Swimming, data = Dolphin2))
```

```
Improve.Dolphin
-0.2
```

```
mosaicplot(Swimming ~ Response, data = Dolphin2)
```

Figure5.7c



$$1. H_0: \pi_{dolphins} - \pi_{control} = 0$$

$$H_a: \pi_{dolphins} - \pi_{control} > 0$$

Test statistic: $\hat{p}_{dolphins} - \hat{p}_{control} = 0.20$ (the difference in the conditional sample proportions)

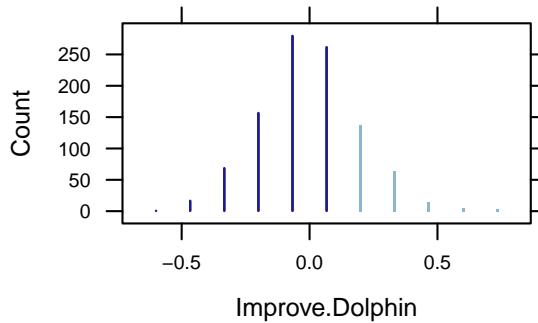
2. We simulate a world in which $\pi_{dolphins} - \pi_{control} = 0$:

```
sim.dol2 <- do(1000) * diff(prop(shuffle(Response) ~ Swimming, data = Dolphin2))
head(sim.dol2, 3)
```

Figure5.7d

```
Improve.Dolphin
1      0.06667
2     -0.20000
3      0.46667
```

```
dotPlot(~Improve.Dolphin, data = sim.dol2, groups = (Improve.Dolphin >=0.20),
       width = 1/15, cex = 5)
```



3. Strength of evidence:

```
favstats(~Improve.Dolphin, data = sim.dol2)

min      Q1   median      Q3      max      mean      sd     n missing
-0.6 -0.06667 -0.06667 0.06667 0.7333 -0.008133 0.1922 1000        0

prop(~(Improve.Dolphin >= 0.2), data = sim.dol2)

TRUE
0.214
```

Figure5.7e

Approximate test for difference in proportions:

```
prop.test(Response ~ Swimming, data = Dolphin2, alt = "greater")

2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 0.543, df = 1, p-value = 0.2306
alternative hypothesis: greater
95 percent confidence interval:
-0.1582 1.0000
sample estimates:
prop 1 prop 2
0.5333 0.3333
```

Figure5.7f

Relative Risk

Exploration 5.2: Is Yawning Contagious?

```
head(Yawning, 3)
```

Exploration5.2.9

```

YawnSeed Response
1 Seeded      Yawn
2 Seeded      Yawn
3 Seeded      Yawn

tally(~Response + YawnSeed, data = Yawning, margin = TRUE)

```

		YawnSeed	
		Control	Seeded
Response	Total		
NoYawn	13	23	36
Yawn	3	11	14
Total	16	34	50

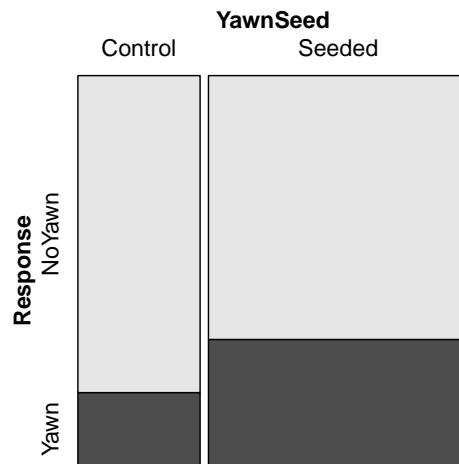
tally(Response ~ YawnSeed, data = Yawning)

		YawnSeed	
		Control	Seeded
Response	Total		
NoYawn	0.8125	0.6765	
Yawn	0.1875	0.3235	

diff(prop(Response ~ YawnSeed, level = "Yawn", data = Yawning))

Yawn.Seeded
0.136

mosaic(Response ~ YawnSeed, data = Yawning, dir = "v")



tally(~shuffle(Response) + YawnSeed, data = Yawning, margins = TRUE)

		YawnSeed		
shuffle(Response)		Control	Seeded	Total
NoYawn		12	24	36
Yawn		4	10	14
Total		16	34	50

$$1. H_0: \pi_{seeded} - \pi_{control} = 0$$

$$H_a: \pi_{seeded} - \pi_{control} > 0$$

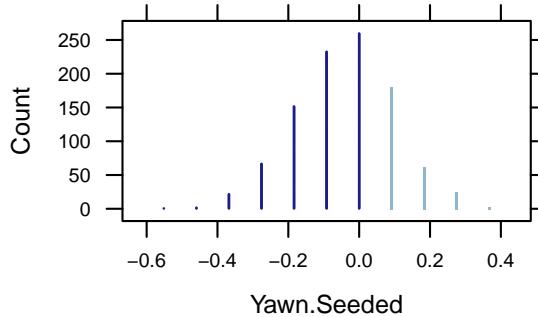
Test statistic: $\hat{p}_{seeded} - \hat{p}_{control} = 0.136$ (the difference in the conditional sample proportions)

2. We simulate a world in which $\pi_{seeded} - \pi_{control} = 0$:

```
Exploration5.2.16
sim.yawn <- 
  do(1000) * diff(prop(shuffle(Response) ~ YawnSeed, level = "Yawn", data = Yawning))
head(sim.yawn, 3)

Yawn.Seeded
1   -0.04779
2    0.13603
3    0.13603

dotPlot(~Yawn.Seeded, data = sim.yawn, groups = (Yawn.Seeded >= 0.136), cex = 5)
```



3. Strength of evidence:

```
Exploration5.2.16b
favstats(~Yawn.Seeded, data = sim.yawn)

  min      01 median      Q3      max      mean      sd     n missing
-0.5074 -0.04779 0.04412 0.136 0.4118 0.0009191 0.1396 1000       0

prop(~(Yawn.Seeded >= 0.136), data = sim.yawn)

  TRUE
0.263
```

Approximate test for difference in proportions:

```
prop.test(Response ~ YawnSeed, data = Yawning, alt = "greater")
```

Exploration5.2.16c

Warning: Chi-squared approximation may be incorrect

2-sample test for equality of proportions with continuity correction

```
data: t(table_from_formula)
X-squared = 0.4379, df = 1, p-value = 0.2541
alternative hypothesis: greater
95 percent confidence interval:
 -0.1177 1.0000
sample estimates:
prop 1 prop 2
0.8125 0.6765
```

```
Yawning2 <- rbind(
  do(12) * data.frame(Response = "NoYawn", YawnSeed = "Control"),
  do(24) * data.frame(Response = "NoYawn", YawnSeed = "Seeded"),
  do(4) * data.frame(Response = "Yawn", YawnSeed = "Control"),
  do(10) * data.frame(Response = "Yawn", YawnSeed = "Seeded")
)
```

Exploration5.2.21

```
head(Yawning2, 3)
```

Exploration5.2.21b

	Response	YawnSeed	.row	.index
1	NoYawn	Control	1	1
2	NoYawn	Control	1	2
3	NoYawn	Control	1	3

```
tally(~Response + YawnSeed, data = Yawning2, margin = TRUE)
```

Response	YawnSeed		Total
	Control	Seeded	
NoYawn	12	24	36
Yawn	4	10	14
Total	16	34	50

```
tally(Response ~ YawnSeed, data = Yawning2)
```

Exploration5.2.21c

Response	Control	Seeded
NoYawn	0.7500	0.7059
Yawn	0.2500	0.2941

```
diff(prop(Response ~ YawnSeed, level = "Yawn", data = Yawning2))
```

```
Yawn.Seeded
0.04412
```

$$1. H_0: \pi_{seeded} - \pi_{control} = 0$$

$$H_a: \pi_{seeded} - \pi_{control} > 0$$

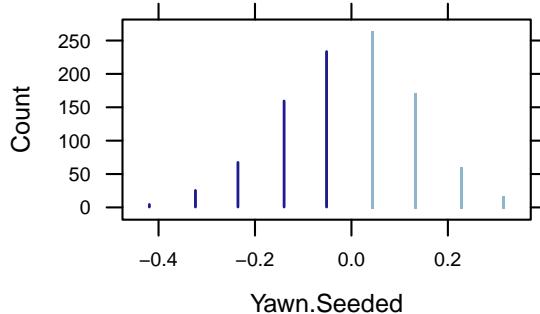
Test statistic: $\hat{p}_{seeded} - \hat{p}_{control} = 0.0441$ (the difference in the conditional sample proportions)

$$2. \text{ We simulate a world in which } \pi_{seeded} - \pi_{control} = 0:$$

```
Exploration5.2.23
sim.yawn2 <-
  do(1000) * diff(prop(shuffle(Response) ~ YawnSeed, level = "Yawn", data = Yawning2))
head(sim.yawn2, 3)
```

```
Yawn.Seeded
1      -0.2316
2      -0.3235
3      -0.1397
```

```
dotPlot(~Yawn.Seeded, data = sim.yawn2, groups = (Yawn.Seeded >= 0.0441),
       cex = 5, width = 1/136)
```



```
Exploration5.2.23b
favstats(~Yawn.Seeded, data = sim.yawn2)

  min      Q1   median      Q3      max      mean      sd     n missing
-0.4154 -0.1397  0.04412  0.04412  0.3199 -0.006801  0.1383  1000        0

prop(~(Yawn.Seeded >= 0.0441), data = sim.yawn2)

  TRUE
0.507
```

Approximate test for difference in proportions:

```
Exploration5.2.23c
prop.test(Response ~ YawnSeed, data = Yawning2, alt = "greater")
```

```
2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 0, df = 1, p-value = 0.5
alternative hypothesis: greater
95 percent confidence interval:
 -0.2196 1.0000
sample estimates:
prop 1 prop 2
0.7500 0.7059
```

Estimation

```
sd <- sd(~Yawn.Seeded, data = sim.yawn2)
sd
[1] 0.1383
```

Exploration5.2.24

Determining a 95% confidence interval using the 2SD Method and simulated standard deviation of the null distribution:

```
# given difference in sample proportions
diff <- diff(prop(Response ~ YawnSeed, level = "Yawn", data = Yawning2))
# previously found simulated standard deviation
sd
[1] 0.1383

# margin of error for 95% CI
MoE <- 2 * sd
MoE
[1] 0.2766

# lower limit of 95% CI
diff - MoE

Yawn.Seeded
-0.2324

# upper limit of 95% CI
diff + MoE

Yawn.Seeded
0.3207
```

Exploration5.2.24b

Determining a 95% confidence interval using the approximate test for proportions:

```
confint(prop.test(Response ~ YawnSeed, data = Yawning2))

prop 1  prop 2    lower   upper   level
0.7500 0.7059 -0.2617  0.3499  0.9500
```

Exploration5.2.24c

Effect of Sample Size

```
Yawning3 <- rbind(
  do(240) * data.frame(Response = "NoYawn", YawnSeed = "Control"),
  do(120) * data.frame(Response = "NoYawn", YawnSeed = "Seeded"),
  do(100) * data.frame(Response = "Yawn", YawnSeed = "Control"),
  do(40)  * data.frame(Response = "Yawn", YawnSeed = "Seeded")
)
```

Exploration5.2.31

```
head(Yawning3, 3)

  Response YawnSeed .row .index
1  NoYawn  Control     1      1
2  NoYawn  Control     1      2
3  NoYawn  Control     1      3

tally(~Response + YawnSeed, data = Yawning3, margin = TRUE)

  YawnSeed
  Response Control Seeded Total
  NoYawn      240     120   360
  Yawn        100      40   140
  Total       340     160   500
```

Exploration5.2.31b

$$1. H_0: \pi_{seeded} - \pi_{control} = 0$$

$$H_a: \pi_{seeded} - \pi_{control} > 0$$

Test statistic: $\hat{p}_{seeded} - \hat{p}_{control} = 0.0441$ (the difference in the conditional sample proportions)

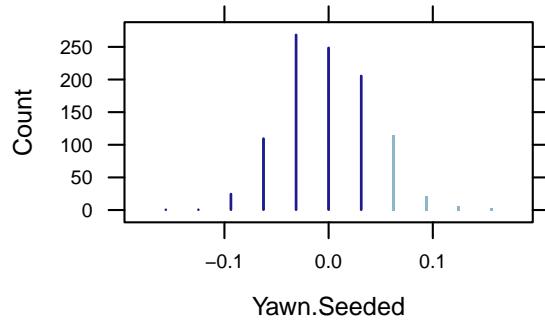
$$2. \text{ We simulate a world in which } \pi_{seeded} - \pi_{control} = 0:$$

```
sim.yawn3 <-
  do(1000) * diff(prop(shuffle(Response) ~ YawnSeed, level = "Yawn", data = Yawning3))
head(sim.yawn3, 3)

  Yawn.Seeded
1      0.02022
2      0.05699
3      0.02022

dotPlot(~Yawn.Seeded, data = sim.yawn3, groups = (Yawn.Seeded >= 0.0441), cex = 5)
```

Exploration5.2.32



3. Strength of evidence:

```
favstats(~Yawn.Seeded, data = sim.yawn3)
```

Exploration5.2.32b

	min	Q1	median	Q3	max	mean	sd	n	missing
	-0.1636	-0.02574	0.001838	0.02941	0.1489	-0.001415	0.04163	1000	0


```
prop(~(Yawn.Seeded >= 0.0441), data = sim.yawn3)
```

Exploration5.2.32c

	TRUE	0.139
--	------	-------

Approximate test for difference in proportions:

```
prop.test(Response ~ YawnSeed, data = Yawning3, alt = "greater")
```

Exploration5.2.32c

2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 0.843, df = 1, p-value = 0.8207
alternative hypothesis: greater
95 percent confidence interval:
-0.1182 1.0000
sample estimates:
prop 1 prop 2
0.7059 0.7500

Relative risk

5.3 Comparing Two Proportions: Theory-Based Approach

Example 5.3: Smoking and Birth Gender

```
head(Smoking, 3)
```

Figure5.9

```
Parents Child
1 smokers girl
2 smokers girl
3 smokers girl
```

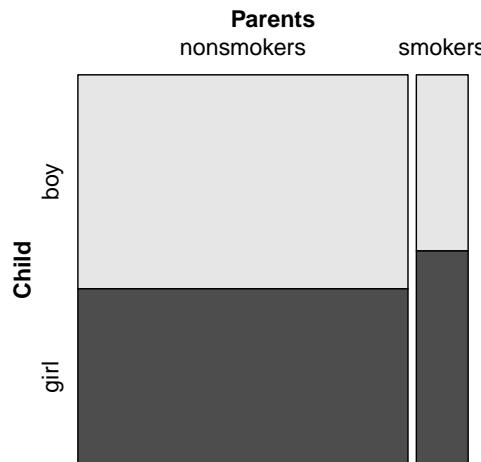
```
summary(Smoking)
```

Parents	Child
nonsmokers:3602	boy :2230
smokers : 565	girl:1937

```
tally(~Parents + Child, data = Smoking, margin = TRUE)
```

Child			
Parents	boy	girl	Total
nonsmokers	1975	1627	3602
smokers	255	310	565
Total	2230	1937	4167

```
mosaic(Child ~ Parents, data = Smoking, dir = "v")
```



```
tally(Child ~ Parents, data = Smoking)
```

Figure5.10

Parents			
Child	nonsmokers	smokers	
boy	0.5483	0.4513	
girl	0.4517	0.5487	

```
diff(prop(Child ~ Parents, data = Smoking))
```

```
boy.smokers
-0.09698
```

$$1. H_0: \pi_{smoker} - \pi_{nonsmoker} = 0$$

$$H_a: \pi_{smoker} - \pi_{nonsmoker} \neq 0$$

Test statistic: $\hat{p}_{smoker} - \hat{p}_{nonsmoker} = -0.097$ (the difference in the conditional sample proportions)

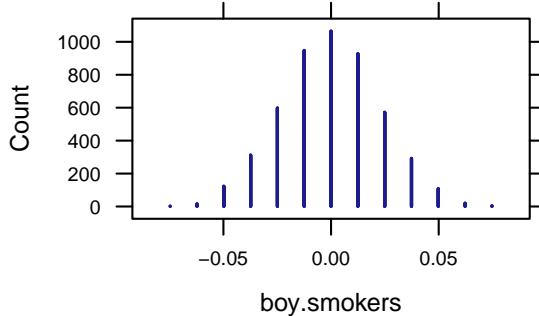
2. We simulate a world in which $\pi_{smoker} - \pi_{nonsmoker} = 0$:

```
sim.smoke <- do(5000) * diff(prop(shuffle(Child) ~ Parents, data = Smoking))
head(sim.smoke, 3)
```

Figure5.10b

```
boy.smokers
1   -0.013030
2    0.015635
3   -0.006888
```

```
dotPlot(~boy.smokers, data = sim.smoke, cex = 25)
```



3. Strength of evidence:

```
favstats(~boy.smokers, data = sim.smoke)
```

Figure5.10c

min	Q1	median	Q3	max	mean	sd	n	missing
-0.07446	-0.01508	-0.0007449	0.01564	0.07501	-0.0001204	0.02236	5000	0

```
prop(~(boy.smokers <= -0.097 | boy.smokers >= 0.097), data = sim.smoke)
```

```
TRUE
0
```

Normal approximation (using simulated standard deviation):

```
sd <- sd(~boy.smokers, data = sim.smoke)
2 * xpnorm(0.097, 0, sd, lower.tail = FALSE) # 2 times because two-sided
```

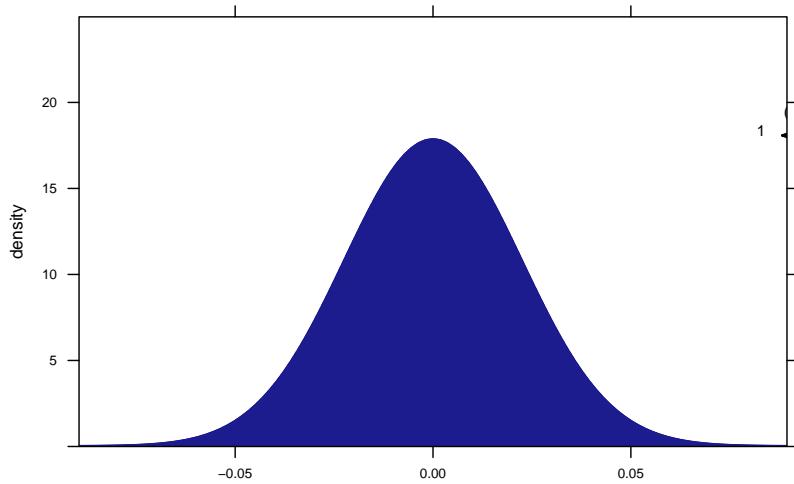
Figure5.11

If $X \sim N(0, 0.0223635667624233)$, then

$$P(X \leq 0.097) = P(Z \leq 4.337) = 1$$

$$P(X > 0.097) = P(Z > 4.337) = 0$$

```
[1] 1.442e-05
```



Approximate test for difference in proportions:

```
prop.test(Child ~ Parents, data = Smoking)
```

Figure5.12

```
2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 18.08, df = 1, p-value = 2.122e-05
alternative hypothesis: two.sided
95 percent confidence interval:
 0.05182 0.14214
sample estimates:
prop 1 prop 2
0.5483 0.4513
```

Estimation

```
confint(prop.test(Child ~ Parents, data = Smoking))
```

Figure5.13

prop 1	prop 2	lower	upper	level
0.54831	0.45133	0.05182	0.14214	0.95000

```
confint(prop.test(Child ~ Parents, data = Smoking, conf.level = 0.99))
```

Figure5.14

prop 1	prop 2	lower	upper	level
0.54831	0.45133	0.03795	0.15600	0.99000

Formulas

```
prop(Child ~ Parents, data = Smoking)
```

Example5.3

```
boy.nonsmokers    boy.smokers
0.5483          0.4513
```

```
p.1 <- 0.548
p.2 <- 0.451
p.hat <- prop(~Child, data = Smoking)
p.hat # pooled prop of success
```

```
boy
0.5352
```

```
n.1 <- 565
n.2 <- 3602
```

```
z <- (p.1 - p.2) / sqrt((p.hat * (1 - p.hat)) * (1/n.1 + 1/n.2))
z
```

Example5.3b

```
boy
4.298
```

```
SE <- sqrt(p.1 * (1 - p.1) / n.1 + p.2 * (1 - p.2) / n.2)
SE
```

Example5.3c

```
[1] 0.02252
```

```
MoE <- 2 * SE
MoE
```

Example5.3d

```
[1] 0.04504
```

Exploration 5.3: Donating Blood

```
sample(Blood, 5)
```

Exploration5.3.2

	Year	Response	orig.ids
2650	2004	did.not	2650
718	2002	did.not	718
820	2002	did.not	820
2445	2004	did.not	2445
2133	2004	did.not	2133

```
tally(Response ~ Year, data = Blood, format = "count", margin = TRUE)
```

	Year	
Response	2002	2004
did.not	1152	1106
donated	210	230
Total	1362	1336

```
tally(Response ~ Year, data = Blood)
```

Exploration5.3.3

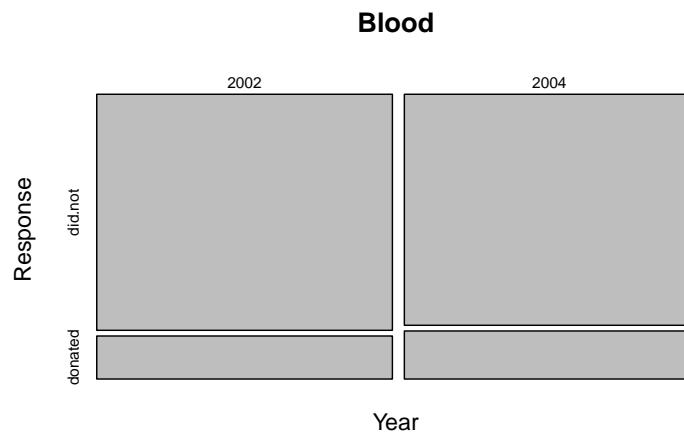
	Year	
Response	2002	2004
did.not	0.8458	0.8278
donated	0.1542	0.1722

```
diff(prop(Response ~ Year, level = "donated", data = Blood))
```

```
donated.2004  
0.01797
```

```
mosaicplot(Year ~ Response, data = Blood)
```

Exploration5.3.4



$$1. H_0: \pi_{2004} - \pi_{2002} = 0$$

$$H_a: \pi_{2004} - \pi_{2002} \neq 0$$

Test statistic: $\hat{p}_{2004} - \hat{p}_{2002} = 0.0180$ (the difference in the conditional sample proportions)

2. We simulate a world in which $\pi_{2004} - \pi_{2002} = 0$:

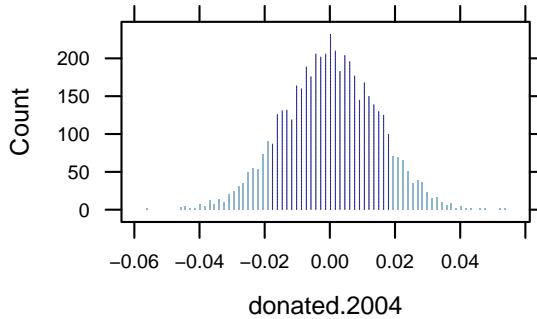
```

sim.blood <-
  do(5000) * diff(prop(shuffle(Response) ~ Year, level = "donated", data = Blood))
head(sim.blood, 3)

donated.2004
1   -0.035407
2    0.009074
3   -0.007236

dotPlot(~ donated.2004, data = sim.blood,
        groups = (donated.2004 <= -0.018 | donated.2004 >= 0.018), width = 0.0001, cex = 2)

```



3. Strength of evidence:

```

favstats(~donated.2004, data = sim.blood)

min      Q1      median      Q3      max      mean      sd      n missing
-0.05617 -0.008718 0.0001781 0.01056 0.05356 8.791e-05 0.01419 5000       0

prop(~(donated.2004 <= -0.018 | donated.2004 >= 0.018), data = sim.blood)

TRUE
0.1886

```

Normal approximation (using simulated standard deviation):

```

sd <- sd(~donated.2004, data = sim.blood)
2 * xpnorm(0.018, 0, sd, lower.tail = FALSE) # 2 times because two-sided

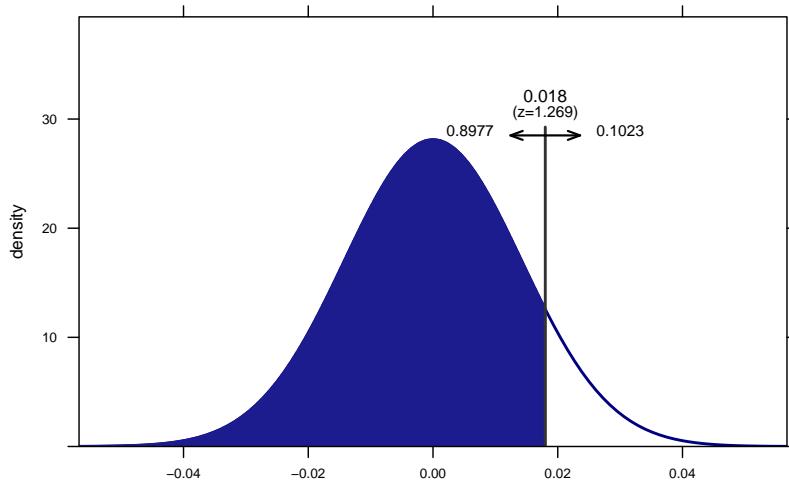
```

If $X \sim N(0, 0.0141881889473206)$, then

```

P(X <= 0.018) = P(Z <= 1.269) = 0.8977
P(X > 0.018) = P(Z > 1.269) = 0.1023
[1] 0.2046

```



Approximate test for difference in proportions:

```
prop.test(Response ~ Year, data = Blood)
```

Exploration5.3.11

```
2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 1.467, df = 1, p-value = 0.2258
alternative hypothesis: two.sided
95 percent confidence interval:
-0.01066 0.04660
sample estimates:
prop 1 prop 2
0.8458 0.8278
```

```
confint(prop.test(Response ~ Year, data = Blood))
```

Exploration5.3.10

prop 1	prop 2	lower	upper	level
0.84581	0.82784	-0.01066	0.04660	0.95000

```
Blood2 <- rbind(
  do(239) * data.frame(Response = "donated", Sex = "Male"),
  do(201) * data.frame(Response = "donated", Sex = "Female"),
  do(1032) * data.frame(Response = "did.not", Sex = "Male"),
  do(1226) * data.frame(Response = "did.not", Sex = "Female")
)
```

Exploration5.3.15

```
tally(~Response + Sex, data = Blood2, margin = TRUE)
```

Exploration5.3.15b

	Sex		Total
Response	Male	Female	
donated	239	201	440
did.not	1032	1226	2258
Total	1271	1427	2698

```
tally(Response ~ Sex, data = Blood2)
```

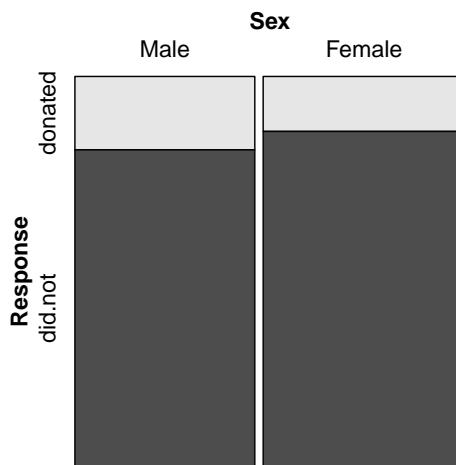
	Sex	
Response	Male	Female
donated	0.1880	0.1409
did.not	0.8120	0.8591

```
diff(prop(Response ~ Sex, data = Blood2))
```

```
donated.Female  
-0.04719
```

```
mosaic(Response ~ Sex, data = Blood2, dir = "v")
```

Exploration5.3.15c



$$1. H_0: \pi_{female} - \pi_{male} = 0$$

$$H_a: \pi_{female} - \pi_{male} \neq 0$$

Test statistic: $\hat{p}_{female} - \hat{p}_{male} = -0.0472$ (the difference in the conditional sample proportions)

2. We simulate a world in which $\pi_{female} - \pi_{male} = 0$:

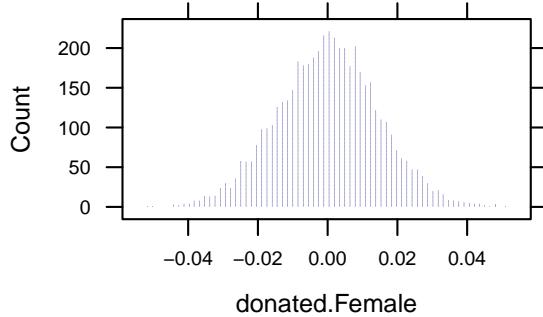
```
sim.blood2 <- do(5000) * diff(prop(shuffle(Response) ~ Sex, data = Blood2))
```

Exploration5.3.15d

```
donated.Female  
1 0.019754
```

```
2      -0.009997
3      0.007853
```

```
dotPlot(~ donated.Female, data = sim.blood2,
       groups = (donated.Female <= -0.0472 | donated.Female >= 0.0472), width = 0.0001)
```



3. Strength of evidence:

```
favstats(~donated.Female, data = sim.blood2)
Exploration5.3.15e
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	-0.05165	-0.009997	0.0004157	0.009341	0.05099	8.519e-05	0.01449	5000	0

```
prop(~(donated.Female <= -0.0472 | donated.Female >= 0.0472), data = sim.blood2)
Exploration5.3.15f
```

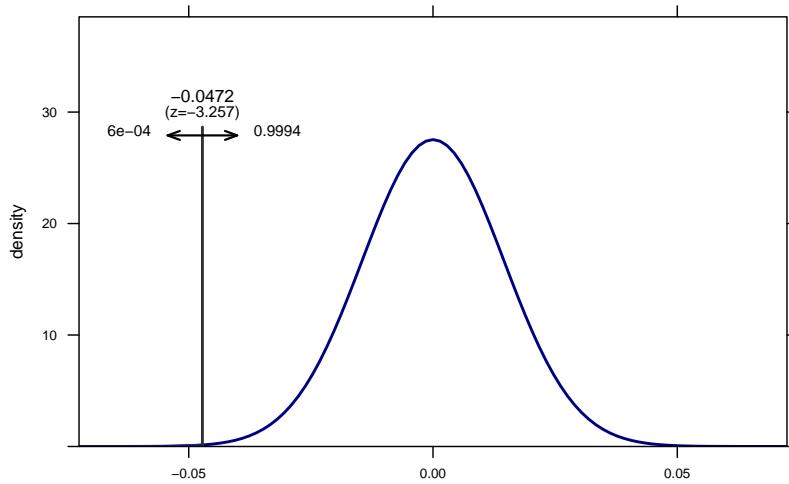
	TRUE
	0.0014

Normal approximation (using simulated standard deviation):

```
sd <- sd(~donated.Female, data = sim.blood2)
2 * xpnorm(-0.0472, 0, sd, xlim = 0 + c(-5, 5) * sd) # 2 times because two-sided
Exploration5.3.15f
```

If $X \sim N(0, 0.0144907969105176)$, then

```
P(X <= -0.0472) = P(Z <= -3.257) = 6e-04
P(X > -0.0472) = P(Z > -3.257) = 0.9994
[1] 0.001125
```



Approximate test for difference in proportions:

```
prop.test(Response ~ Sex, data = Blood2)
```

Exploration5.3.15g

```
2-sample test for equality of proportions with continuity correction

data: t(table_from_formula)
X-squared = 10.62, df = 1, p-value = 0.001117
alternative hypothesis: two.sided
95 percent confidence interval:
 0.01838 0.07599
sample estimates:
prop 1 prop 2
0.1880 0.1409
```

6

Comparing Two Means

6.1 Comparing Two Groups: Quantitative Response

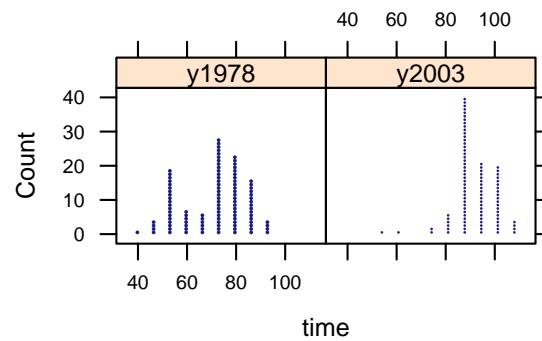
Example 6.1: Geyser Eruptions

```
head(OldFaithful, 3)
```

Figure6.1

```
  year time
1 y1978 78
2 y1978 74
3 y1978 68
```

```
dotPlot(~time | year, data = OldFaithful)
```



```
fivenum(~time, data = OldFaithful)
```

Example6.1

```
[1] 42 73 84 91 110
```

```
fivenum(time ~ year, data = OldFaithful)
```

```
y19781 y19782 y19783 y19784 y19785 y20031 y20032 y20033 y20034 y20035
 42.0    59.0   75.0   80.5   95.0   56.0   87.0   91.0   97.0  110.0
```

```
IQR(~time, data = OldFaithful)
```

```
[1] 18
```

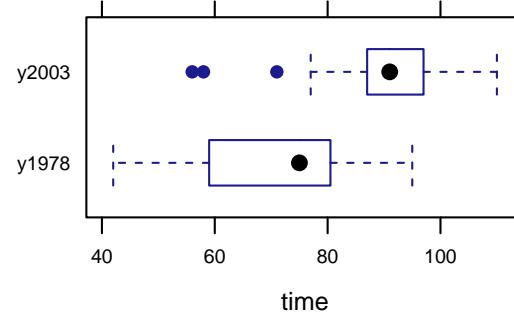
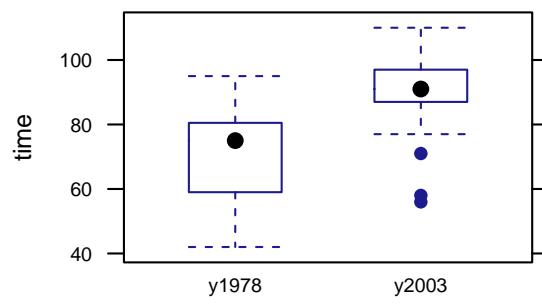
```
IQR(~time | year, data = OldFaithful)
```

```
y1978 y2003
20.75 10.00
```

Example6.1b

```
bwplot(time ~ year, data = OldFaithful)
bwplot(year ~ time, data = OldFaithful, horizontal = TRUE)
```

Figure6.2



Exploration 6.1A: Haircut Prices

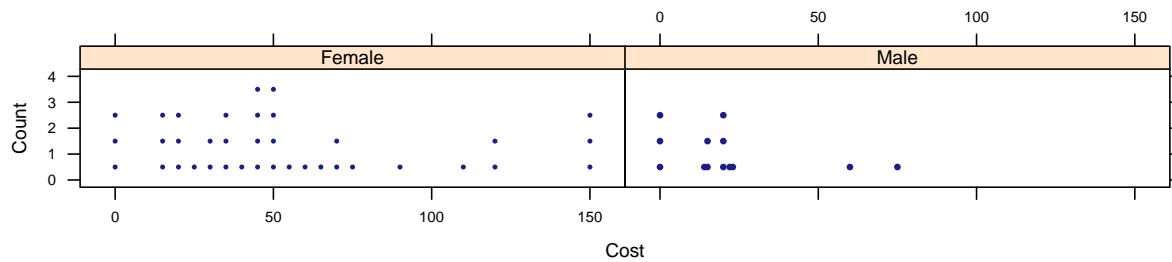
```
head(Haircuts)
```

Exploration6.1A.1

```
Sex Cost
1 Female 50
2 Male 20
3 Female 60
4 Male 75
5 Female 150
6 Male 23
```

```
dotPlot(~Cost | Sex, data = Haircuts, width = 1, cex = 0.25)
```

Exploration6.1A.4



```
favstats(~Cost ~ Sex, data = Haircuts)
```

Exploration6.1A.8

	group	min	Q1	median	Q3	max	mean	sd	n	missing
1	Female	0	25	45	70	150	54.05	41.61	37	0
2	Male	0	14	20	22	75	21.85	22.14	13	0

```
diff(mean(Cost ~ Sex, data = Haircuts))
```

Exploration6.1A.10

Male
-32.21

Further Analyses

```
median(Cost ~ Sex, data = Haircuts)
```

Exploration6.1A.14

Female Male
 45 20

```
fivenum(~Cost, data = Haircuts)
```

Exploration6.1A.16

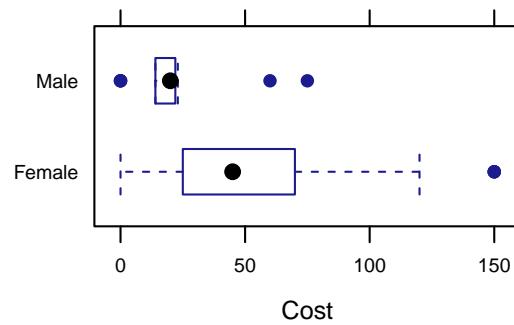
[1] 0 20 35 60 150

```
fivenum(~Cost ~ Sex, data = Haircuts)
```

Female1	Female2	Female3	Female4	Female5	Male1	Male2	Male3	Male4	Male5
0	25	45	70	150	0	14	20	22	75

```
bwplot(Sex ~ Cost, data = Haircuts, horizontal = TRUE)
```

Exploration6.1A.17



```
IQR(Cost ~ Sex, data = Haircuts)
```

Exploration6.1A.18

Female	Male
45	8

6.2 Comparing Two Means: Simulation-Based Approach

Example 6.2: Bicycling to Work

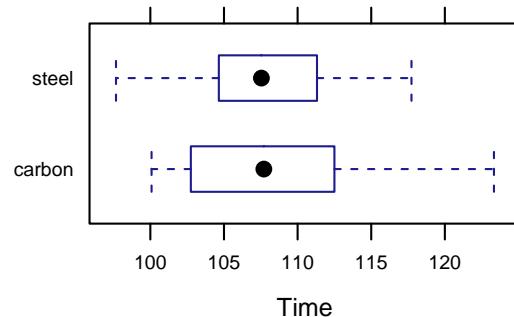
```
head(BikeTimes)
```

Table6.2

Frame	Time
1 steel	115.8
2 steel	115.7
3 steel	108.7
4 steel	117.7
5 steel	112.6
6 steel	109.6

```
bwplot(Frame ~ Time, data = BikeTimes, horizontal = TRUE)
```

Figure6.3



```
favstats(Time ~ Frame, data = BikeTimes)
```

Table6.3

.group	min	Q1	median	Q3	max	mean	sd	n	missing
1 carbon	100.08	102.8	107.7	112.5	123.3	108.3	6.248	26	0
2 steel	97.67	104.7	107.5	111.2	117.7	107.8	4.892	30	0

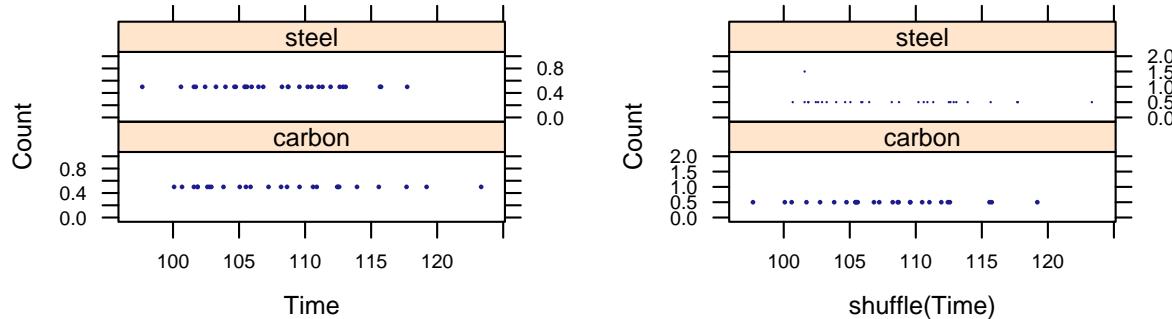
```
dotPlot(~Time | Frame, data = BikeTimes, width = 0.01, cex = 0.1, layout = c(1, 2))
diff(mean(Time ~ Frame, data = BikeTimes))
```

Figure6.4

```
steel
-0.5347
```

```
dotPlot(~shuffle(Time) | Frame, data = BikeTimes, width = 0.01, cex = 0.1, layout = c(1, 2))
diff(mean(shuffle(Time) ~ Frame, data = BikeTimes))
```

```
steel
-0.1817
```



1. $H_0: \mu_{carbon} - \mu_{steel} = 0$
 $H_a: \mu_{carbon} - \mu_{steel} \neq 0$
Test statistic: $\bar{x}_{carbon} - \bar{x}_{steel} = 0.53$ (the difference in the sample means)
2. We simulate a world in which $\mu_{carbon} - \mu_{steel} = 0$:

```
sim.bike <- do(1000) * diff(mean(shuffle(Time) ~ Frame, data = BikeTimes))
```

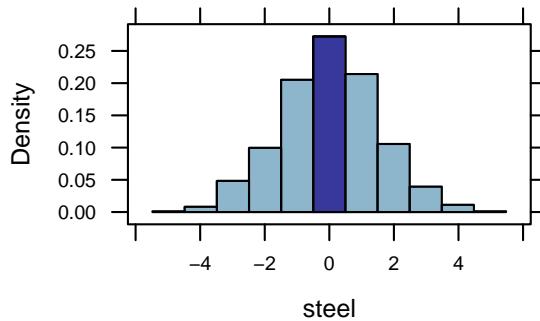
Figure6.7

Loading required package: parallel

```
head(sim.bike, 3)
```

```
steel
1 -2.3104
2 -0.0106
3 2.9976
```

```
histogram(~ steel, data = sim.bike,
groups = (steel <= -0.53 | steel >= 0.53))
```



3. Strength of evidence:

```
favstats(~steel, data = sim.bike)

min      Q1   median      Q3    max      mean      sd      n missing
-4.699 -0.99  0.03667  0.9754  5.25  0.008164  1.515  1000        0

prop(~(steel <= -0.53 | steel >= 0.53), data = sim.bike)

TRUE
0.717
```

Figure6.8

Estimating a confidence interval

Determining a 95% confidence interval using the 2SD Method and standard deviation of the null distribution:

```
diff <- -diff(mean(Time ~ Frame, data = BikeTimes)) # note the negative sign
sd <- sd(~steel, data = sim.bike)
diff - 2 * sd # lower limit of 95% CI

steel
-2.495

diff + 2 * sd # upper limit of 95% CI

steel
3.564
```

Example6.2

Exploration 6.2: Lingering Effects of Sleep Deprivation

```
head(Sleep)
```

Exploration6.2.2

```
sleep time
1 unrestricted -7.0
2 unrestricted 11.6
3 unrestricted 12.1
4 unrestricted 12.6
5 unrestricted 14.5
6 unrestricted 18.6
```

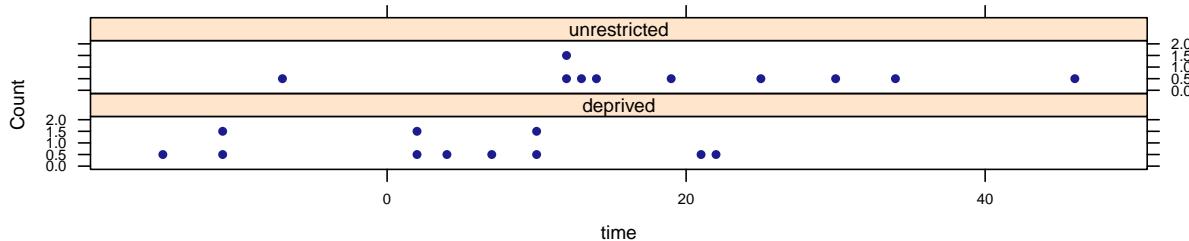
Exploration6.2.5

```
dotPlot(~time | sleep, data = Sleep, cex = 0.5, width = 1, layout = c(1, 2))
favstats(time ~ sleep, data = Sleep)

    .group   min     Q1 median     Q3 max   mean     sd n missing
1   deprived -14.7 -4.25   4.50  9.80 21.8  3.90 12.17 11      0
2 unrestricted -7.0 12.22  16.55 29.18 45.6 19.82 14.73 10      0

diff(mean(time ~ sleep, data = Sleep))

unrestricted
15.92
```



Exploration6.2.9

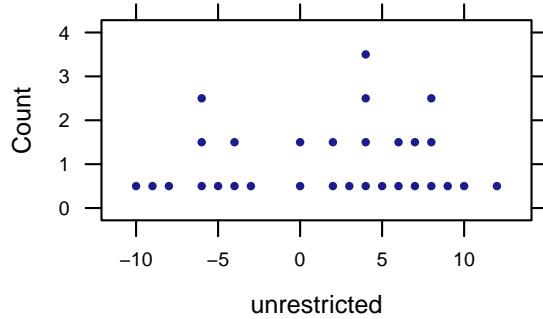
```
diff(mean(shuffle(time) ~ sleep, data = Sleep))

unrestricted
6.355

sample <- do(30) * diff(mean(shuffle(time) ~ sleep, data = Sleep))
head(sample, 3)

unrestricted
1      -4.450
2      -3.095
3      2.155

dotPlot(~unrestricted, data = sample, width = 1, cex = 0.25)
```



$$1. H_0: \mu_{unrestricted} - \mu_{deprived} = 0$$

$$H_a: \mu_{unrestricted} - \mu_{deprived} > 0$$

Test statistic: $\bar{x}_{unrestricted} - \bar{x}_{deprived} = 15.92$ (the difference in the sample means)

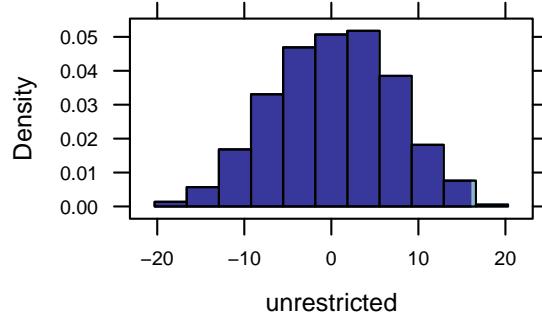
2. We simulate a world in which $\mu_{unrestricted} - \mu_{deprived} = 0$:

```
sim.sleep <- do(1000) * diff(mean(shuffle(time) ~ sleep, data = Sleep))
head(sim.sleep, 3)
```

Exploration6.2.10

```
unrestricted
1      -4.4691
2      -0.4409
3     -11.6664
```

```
histogram(~ unrestricted, data = sim.sleep,
          groups = (unrestricted >= 15.92))
```



3. Strength of evidence:

```
favstats(~unrestricted, data = sim.sleep)
```

Exploration6.2.10b

min	Q1	median	Q3	max	mean	sd	n	missing
-18.92	-4.488	0.485	5.329	17.98	0.2984	6.833	1000	0

```
prop(~(unrestricted >= 15.92), data = sim.sleep)
```

```
TRUE
0.005
```

Determining a 95% confidence interval using the 2SD Method and standard deviation of the null distribution:

```
diff <- diff(mean(time ~ sleep, data = Sleep))
sd <- sd(~unrestricted, data = sim.sleep)
diff - 2 * sd # lower limit of 95% CI
```

Exploration6.2.13

```
unrestricted
2.255
```

```
diff + 2 * sd # upper limit of 95% CI
```

```
unrestricted
29.59
```

Another statistic

```
median(time ~ sleep, data = Sleep)
```

Exploration6.2.16

```
deprived unrestricted
4.50 16.55
```

```
diff(median(time ~ sleep, data = Sleep))
```

```
unrestricted
12.05
```

$$1. H_0: \text{median}_{\text{unrestricted}} - \text{median}_{\text{deprived}} = 0$$

$$H_a: \text{median}_{\text{unrestricted}} - \text{median}_{\text{deprived}} > 0$$

Test statistic: $\text{median}_{\text{unrestricted}} - \text{median}_{\text{deprived}} = 12.05$ (the difference in the sample medians)

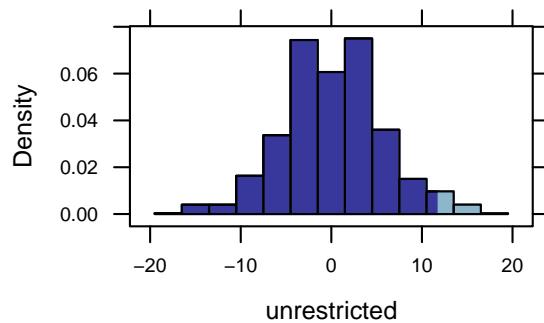
$$2. \text{ We simulate a world in which } \text{median}_{\text{unrestricted}} - \text{median}_{\text{deprived}} = 0:$$

```
sim.med <- do(1000) * diff(median(shuffle(time) ~ sleep, data = Sleep))
head(sim.med, 3)
```

Exploration6.2.16b

```
unrestricted
1 2.25
2 8.40
3 2.50
```

```
histogram(~ unrestricted, data = sim.med,
          groups = (unrestricted >= 12.05),
          width = 3)
```



3. Strength of evidence:

```
favstats(~unrestricted, data = sim.med)
```

Exploration6.2.16c

	min	Q1	median	Q3	max	mean	sd	n	missing
	-16.5	-3.3	-0.55	3.3	17.55	0.1122	5.502	1000	0

```
prop(~(unrestricted >= 12.05), data = sim.med)
```

	TRUE
	0.029

6.3 Comparing Two Means: Theory-Based Approach

Example 6.3: Breastfeeding and Intelligence

```
head(BreastFeedIntell)
```

Table6.4

	Feeding	GCI
1	Breastfed	126.70
2	Breastfed	124.69
3	Breastfed	99.79
4	Breastfed	104.97
5	Breastfed	97.25
6	Breastfed	131.28

```
favstats(GCI ~ Feeding, data = BreastFeedIntell)
```

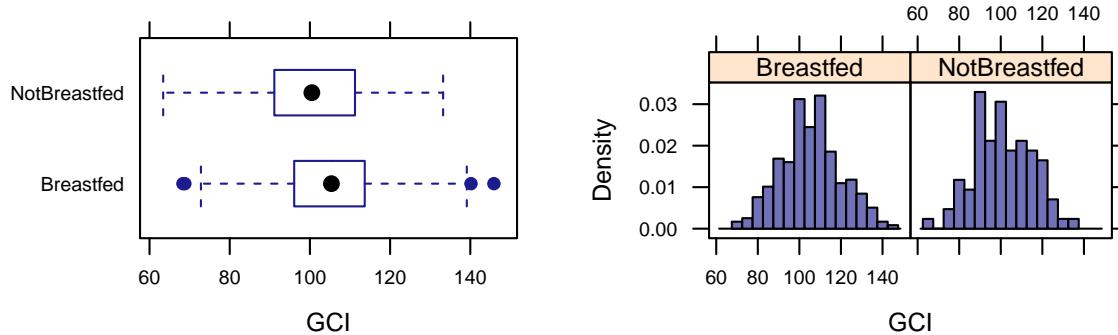
	.group	min	Q1	median	Q3	max	mean	sd	n	missing
1	Breastfed	68.33	96.08	105.4	113.7	145.9	105.3	14.5	237	0
2	NotBreastfed	63.41	91.13	100.5	111.2	133.2	100.9	14.0	85	0

```
diff(mean(GCI ~ Feeding, data = BreastFeedIntell))
```

	NotBreastfed
	-4.4

```
bwplot(Feeding ~ GCI, horizontal = TRUE, data = BreastFeedIntell)
histogram(~GCI | Feeding, data = BreastFeedIntell, width = 5)
```

Figure6.10



$$1. H_0: \mu_{breastfed} - \mu_{not} = 0$$

$$H_a: \mu_{breastfed} - \mu_{not} \neq 0$$

Test statistic: $\bar{x}_{breastfed} - \bar{x}_{not} = 4.40$ (the difference in the sample means)

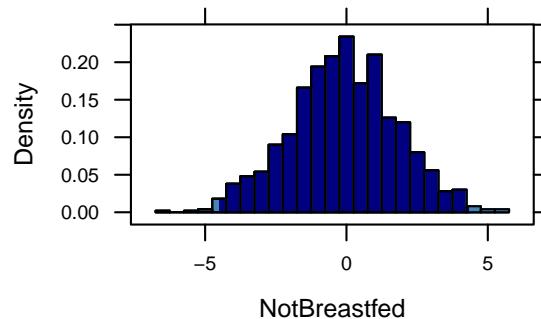
2. We simulate a world in which $\mu_{breastfed} - \mu_{not} = 0$:

```
sim.GCI <- do(1000) * diff(mean(shuffle(GCI) ~ Feeding, data = BreastFeedIntell))
head(sim.GCI, 3)
```

Figure6.11

```
NotBreastfed
1      -0.8512
2       1.6837
3       2.0941
```

```
histogram(~ NotBreastfed, data = sim.GCI, width = 0.5,
          group = cut(NotBreastfed, c(-7, -4.40, 4.40, 7)),
          fcol = c("steelblue", "navy", "steelblue"))
```



3. Strength of evidence:

```
favstats(~NotBreastfed, data = sim.GCI)

   min     Q1    median      Q3     max     mean      sd     n missing
-6.551 -1.323 -0.06835  1.162  5.624 -0.0942  1.885  1000       0

prop(~(NotBreastfed <= -4.4 | NotBreastfed >= 4.4), data = sim.GCI)

  TRUE
0.016
```

Figure6.12

Determining a 95% confidence interval using the 2SD Method and standard deviation of the null distribution:

```
diff <- -diff(mean(GCI ~ Feeding, data = BreastFeedIntell)) # note the negative sign
sd <- sd(~NotBreastfed, data = sim.GCI)
sd

[1] 1.885

diff - 2 * sd # lower limit of 95% CI

NotBreastfed
0.6305

diff + 2 * sd # upper limit of 95% CI

NotBreastfed
8.17
```

Example6.3

```
t.test(GCI ~ Feeding, data = BreastFeedIntell)

Welch Two Sample t-test

data: GCI by Feeding
t = 2.462, df = 153, p-value = 0.01491
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.8699 7.9302
sample estimates:
mean in group Breastfed mean in group NotBreastfed
105.3                      100.9

stat(t.test(GCI ~ Feeding, data = BreastFeedIntell))

t
2.462
```

Figure6.13

Exploration 6.3: Close Friends

```
head(CloseFriends)
```

Exploration6.3.1

	Sex	Friends
1	Men	0
2	Men	0
3	Men	0
4	Men	0
5	Men	0
6	Men	0


```
tally(~Friends + Sex, data = CloseFriends, margin = TRUE)
```

		Sex		
Friends	Men	Women	Total	
0	196	201	397	
1	135	146	281	
2	108	155	263	
3	100	132	232	
4	42	86	128	
5	40	56	96	
6	33	37	70	
Total	654	813	1467	

```
favstats(Friends ~ Sex, data = CloseFriends)
```

Exploration6.3.7

	.group	min	Q1	median	Q3	max	mean	sd	n	missing
1	Men	0	0	1	3	6	1.861	1.777	654	0
2	Women	0	1	2	3	6	2.089	1.760	813	0

```
diff(mean(Friends ~ Sex, data = CloseFriends))
```

Women
0.2277

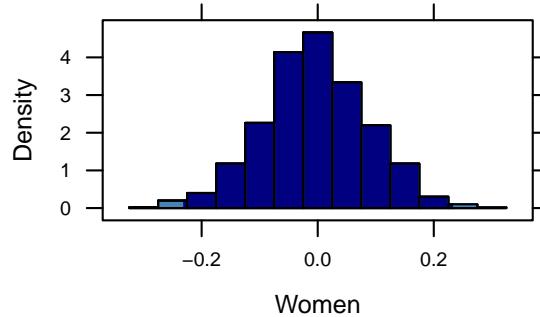
1. $H_0: \mu_{men} - \mu_{women} = 0$
 $H_a: \mu_{men} - \mu_{women} \neq 0$
Test statistic: $\bar{x}_{men} - \bar{x}_{women} = -0.228$ (the difference in the sample means)
2. We simulate a world in which $\mu_{men} - \mu_{women} = 0$:

```
sim.fri <- do(1000) * diff(mean(shuffle(Friends) ~ Sex, data = CloseFriends))
head(sim.fri, 3)
```

Exploration6.3.8

	Women
1	-0.006816
2	-0.015093
3	0.062161

```
histogram(~ Women, data = sim.fri, width = 0.05,
          group = cut(Women, c(-0.4, -0.228, 0.228, 0.4)),
          fcol = c("steelblue", "navy", "steelblue"))
```



3. Strength of evidence:

```
favstats(~Women, data = sim.fri)
```

Exploration6.3.10

	min	01	median	Q3	max	mean	sd	n	missing
	-0.2938	-0.05924	-0.001298	0.05664	0.2856	-0.00327	0.0903	1000	0

```
prop(~(Women <= -0.228 | Women >= 0.228), data = sim.fri)
```

```
TRUE  
0.015
```

```
t.test(Friends ~ Sex, data = CloseFriends)
```

Exploration6.3.13

```
Welch Two Sample t-test

data: Friends by Sex
t = -2.45, df = 1393, p-value = 0.01442
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.41004 -0.04537
sample estimates:
mean in group Men mean in group Women
1.861                 2.089

stat(t.test(Friends ~ Sex, data = CloseFriends))

t
-2.45
```

```
pval(t.test(Friends ~ Sex, data = CloseFriends))
```

Exploration6.3.17

```
p.value  
0.01442
```

Validity Conditions

```
confint(t.test(Friends ~ Sex, data = CloseFriends))
```

Exploration6.3.20

mean in group Men	mean in group Women	lower	upper
1.86086	2.08856	-0.41004	-0.04537
level			
0.95000			

7

Paired Data: One Quantitative Variable

7.1 Paired Designs

7.2 Simulation-Based Approach for Analyzing Paired Data

Example 7.2: Rounding First Base (continued)

Let's begin by creating a data frame that organizes this data differently. We'll call the new data frame `FirstBase2`.

```
require(tidyr)  
  
Loading required package: tidyR  
  
FirstBase2 <- FirstBase %>% gather(angle, time)  
sample(FirstBase2, 5)  
  
  angle time orig.ids  
22 narrow 6.30      22  
10 narrow 5.80      10  
 6 narrow 5.55       6  
19 narrow 5.45      19  
 2 narrow 5.70       2
```

Example7.2

```
head(FirstBase, 10)  
  
  narrow wide  
1    5.50 5.55  
2    5.70 5.75  
3    5.60 5.50  
4    5.50 5.40  
5    5.85 5.70
```

Table7.1

```

6      5.55 5.60
7      5.40 5.35
8      5.50 5.35
9      5.15 5.00
10     5.80 5.70

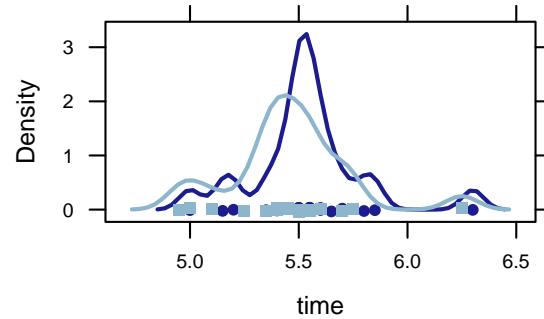
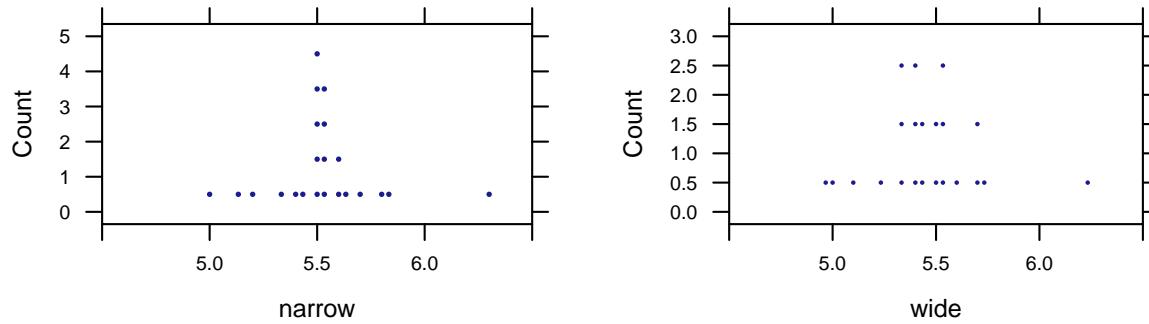
```

```

dotPlot(~narrow, data = FirstBase, nint = 40, cex = 0.2, xlim = c(4.5, 6.5))
dotPlot(~wide, data = FirstBase, nint = 40, cex = 0.1, xlim = c(4.5, 6.5))
densityplot(~time, groups = angle, data = FirstBase2)

```

Figure7.3



```
favstats(~(narrow - wide), data = FirstBase)
```

Table7.2

	min	Q1	median	Q3	max	mean	sd	n	missing
	-0.1	0.05	0.1	0.1375	0.2	0.075	0.0883	22	0

```
dotPlot(~(narrow - wide), data = FirstBase)
```

Figure7.4

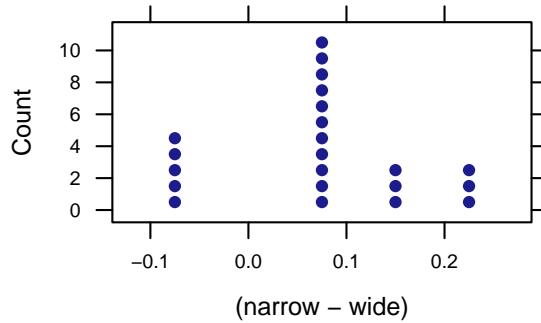


Table7.3

```
Swap.Base <- swap(FirstBase, c("narrow", "wide"))
Swap.Base
```

	narrow	wide
1	5.50	5.55
2	5.75	5.70
3	5.50	5.60
4	5.50	5.40
5	5.85	5.70
6	5.55	5.60
7	5.40	5.35
8	5.50	5.35
9	5.00	5.15
10	5.70	5.80
11	5.20	5.10
12	5.55	5.45
13	5.35	5.45
14	5.00	4.95
15	5.40	5.50
16	5.50	5.55
17	5.55	5.35
18	5.50	5.55
19	5.45	5.25
20	5.40	5.60
21	5.65	5.55
22	6.25	6.30

```
mean(~(narrow - wide), data = Swap.Base)
```

```
[1] 0.01136
```

We simulate a world in which $\mu_d = 0$:

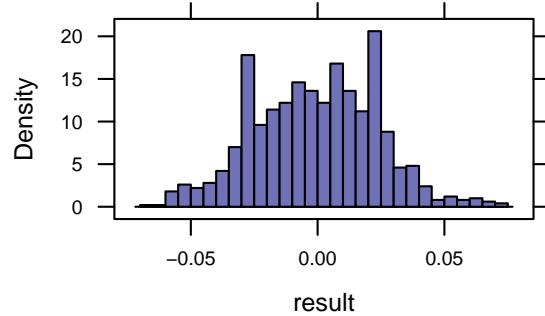
Figure7.6

```
Sim.Base <- do(1000) * mean(~ (narrow - wide), data = swap(FirstBase, c("narrow", "wide")))
head(Sim.Base, 3)
```

```
result
1 0.002273
```

```
2 0.025000
3 0.034091
```

```
histogram(~ result, data = Sim.Base, width = 0.005, center = 0.0025)
```



```
histogram(~result, data = Sim.Base, width = 0.005, center = 0.0025, groups = (result >= 0.075))
sd <- sd(~result, data = Sim.Base)
sd
```

```
[1] 0.02432
```

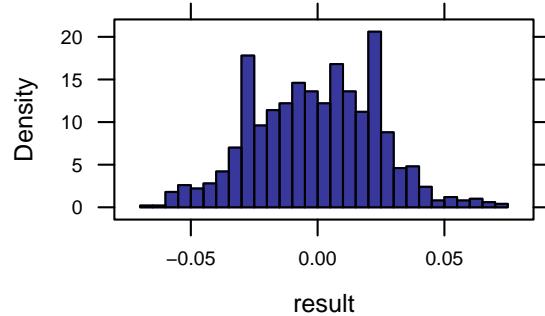
```
0.075 - 2 * sd
```

```
[1] 0.02636
```

```
0.075 + 2 * sd
```

```
[1] 0.1236
```

Figure7.7



```
sim.firstbase <- do(1000) * diffmean(time ~ shuffle(angle), data = FirstBase2)
head(sim.firstbase, 3)
```

Figure7.8

```

diffmean
1 -0.16591
2  0.07955
3 -0.05227

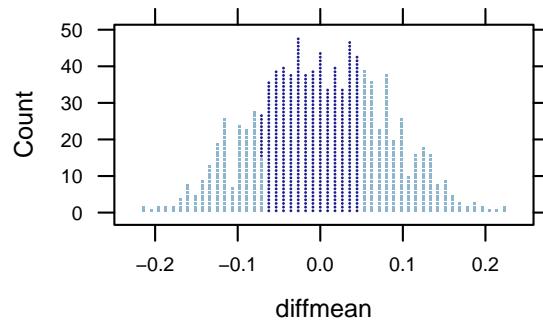
favstats(~diffmean, data = sim.firstbase)

   min      Q1 median      Q3  max      mean      sd    n missing
-0.2114 -0.05682 0.002273 0.05682 0.225 0.0002773 0.07889 1000        0

dotPlot(~diffmean, data = sim.firstbase, nint = 50, groups = (diffmean <= -0.075 | diffmean >= 0.05))
prop(~(diffmean <= -0.075 | diffmean >= 0.075), data = sim.firstbase)

TRUE
0.382

```



Exploration 7.2: Exercise and Heart Rate

```
head(JJvsBicycle)
```

Exploration7.2.5

```

JJ bicycle
1 118     118
2 146     124
3 134     92
4  94     80
5 146     111
6 114     112

```

```
favstats(~JJ, data = JJvsBicycle)
```

Exploration7.2.7

```

min      Q1 median      Q3  max      mean      sd    n missing
70 102.2    115 129.2 146 114.6 19.57 22        0

```

```
favstats(~bicycle, data = JJvsBicycle)
```

```
min   Q1 median   Q3 max   mean    sd  n missing
70  87.25  97.5 121.8 143 102.7 20.66 22       0
```

```
mean(JJvsBicycle$JJ - JJvsBicycle$bicycle)
```

```
[1] 11.95
```

```
swap.bike <- swap(JJvsBicycle, c("JJ", "bicycle"))
mean(~(JJ - bicycle), data = swap.bike)
```

Exploration7.2.8

```
[1] 1.682
```

```
sd(~(JJ - bicycle), data = swap.bike)
```

```
[1] 23.47
```

$$1. H_0: \mu_d = 0$$

$$H_a: \mu_d \neq 0$$

Test statistic: $\bar{x}_d = -6.773$ (the mean difference in sample)

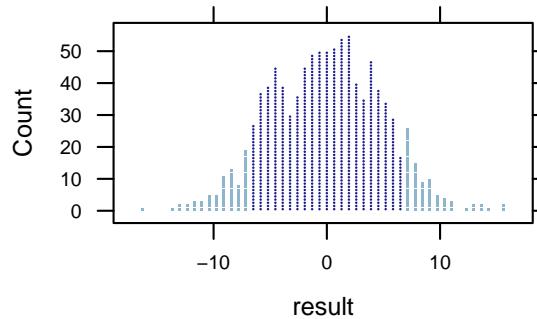
2. We simulate a world in which $\mu_d = 0$:

```
sim.bike <- do(1000) * mean(~(JJ - bicycle),
                           data = swap(JJvsBicycle, c("JJ", "bicycle")))
head(sim.bike, 3)

result
1 -6.136
2  1.409
3 -0.500

dotPlot(~ result, data = sim.bike, nint = 50, groups = (result <=-6.773 | result >=6.773))
```

Exploration7.2.10



3. Strength of evidence:

```
favstats(~result, data = sim.bike)

  min      Q1   median   Q3    max     mean     sd     n missing
-16.23 -3.773  0.04545  3.5  15.59 -0.03773 4.861 1000       0

prop(~(result <= -6.773 | result >= 6.773), data = sim.bike)

TRUE
0.153
```

Exploration7.2.12

Standardized statistic:

```
sd <- sd(~result, data = sim.bike)
xpnorm(-6.773, 0, sd, plot = FALSE)
```

Exploration7.2.14

If $X \sim N(0, 4.86055331118666)$, then

```
P(X <= -6.773) = P(Z <= -1.393) = 0.0817
P(X > -6.773) = P(Z > -1.393) = 0.9183
[1] 0.08174
```

Exploration7.2.15

95% confidence interval using 2SD Method:

```
sd <- sd(~result, data = sim.bike)
-6.773 - 2 * sd

[1] -16.49

-6.773 + 2 * sd

[1] 2.948
```

Exploration7.2.15

Let's again create the stacked data.

```
require(tidyverse)
JJvsBicycle2 <- JJvsBicycle %>% gather(exercise, heartrate)
sample(JJvsBicycle2, 5)

  exercise heartrate orig.ids
27  bicycle      111      27
42  bicycle      96       42
32  bicycle      82       32
39  bicycle      87       39
1     JJ        118       1
```

Exploration7.2.17

```

sim.bike2 <- do(1000) * diffmean(heartrate ~ shuffle(exercise), data = JJvsBicycle2)
head(sim.bike2, 3)

diffmean
1 -0.7727
2 12.9545
3 14.2273

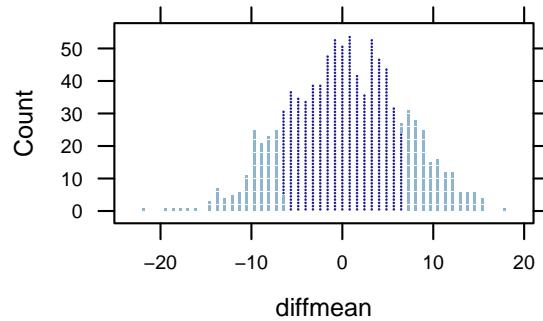
favstats(~diffmean, data = sim.bike2)

min      Q1 median      Q3    max   mean    sd     n missing
-21.68 -4.227  0.2727  4.614  17.95  0.3147  6.36  1000        0

dotPlot(~diffmean, data = sim.bike2, nint = 50, groups = (diffmean <= -6.773 | diffmean >=
6.773))
prop(~(diffmean <= -6.773 | diffmean >= 6.773), data = sim.bike2)

TRUE
0.306

```



7.3 Theory-Based Approach to Analyzing Data from Paired Samples

Example 7.3: How Many M&Ms Would You Like?

```
head(BowlsMMs)
```

Table7.4

	Small	Large
1	33	41
2	24	92
3	35	61
4	24	19
5	40	21
6	33	35

```
favstats(~Small, data = BowlsMMs)

min Q1 median Q3 max mean sd n missing
24 26      34 40    88 38.59 16.9 17      0

favstats(~Large, data = BowlsMMs)

min Q1 median Q3 max mean sd n missing
11 33      42 62   104 49.47 27.21 17      0

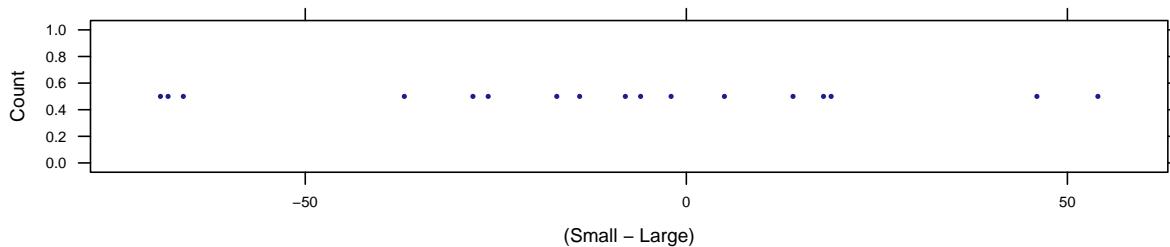
favstats(BowlsMMs$Small - BowlsMMs$Large)

min Q1 median Q3 max mean sd n missing
-69 -28     -8 14    54 -10.88 36.3 17      0
```

Table7.5

```
dotPlot(~(Small - Large), data = BowlsMMs, width = 1, cex = 0.05)
```

Figure7.9



$$1. H_0: \mu_d = 0$$

$$H_a: \mu_d < 0$$

Test statistic: $\bar{x}_d = -10.88$ (the mean difference in paired samples)

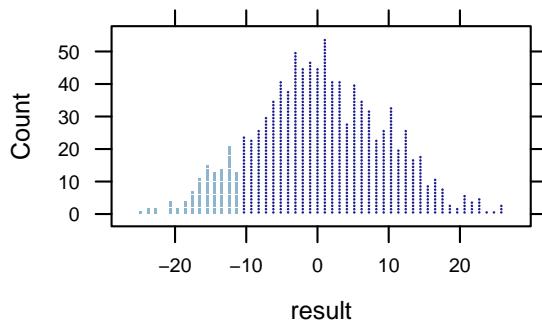
$$2. \text{ We simulate a world in which } \mu_d = 0:$$

```
sim.mm <- do(1000) * mean(~(Small - Large), data = swap(BowlsMMs, c("Small", "Large")))
head(sim.mm, 3)

result
1 11.000
2 6.647
3 -5.353

dotPlot(~ result, data = sim.mm, nint = 50, groups = (result <= -10.88))
```

Figure7.10



3. Strength of evidence:

```
favstats(~result, data = sim.mm)

min      Q1 median      Q3   max   mean      sd     n missing
-24.41  -5.588  0.1765  6.647  26.18  0.4391  9.094  1000        0

prop(~(result <= -10.88), data = sim.mm)

TRUE
0.109
```

Figure7.11

Theory-based approach

```
t.test(BowlsMMs$Small, BowlsMMs$Large, paired = TRUE, alt = "less")

Paired t-test

data: x and BowlsMMs$Large
t = -1.236, df = 16, p-value = 0.1171
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf 4.489
sample estimates:
mean of the differences
-10.88
```

Figure7.12

Exploration 7.3: comparing Auction Formats

```
head(Auction)

dutch    FP
1      25 26.25
2      24 25.25
```

Exploration7.3.1

```
3    26 27.00
4    20 20.75
5    20 20.75
6    15 15.25
```

```
summary(Auction)

      dutch          FP
Min.   : 0.15   Min.   : 0.10
1st Qu.: 2.00   1st Qu.: 1.19
Median : 3.00   Median : 2.27
Mean   : 5.16   Mean   : 4.78
3rd Qu.: 7.00   3rd Qu.: 6.05
Max.   :26.00   Max.   :27.00

favstats(Auction$dutch - Auction$FP)

  min   Q1 median   Q3 max   mean      sd   n missing
-1.25  0   0.25  0.5 2.4 0.3835 0.6752 88       0
```

Exploration7.3.5

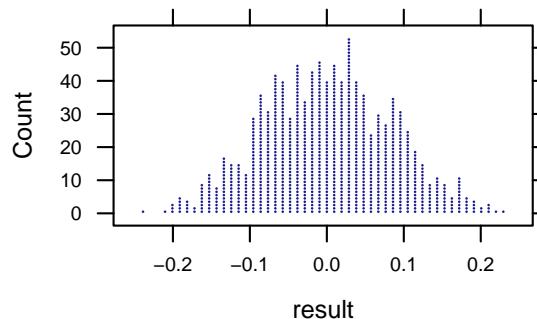
1. $H_0: \mu_d = 0$
 $H_a: \mu_d \neq 0$
Test statistic: $\bar{x}_d = 0.384$ (the mean difference in paired samples)
2. We simulate a world in which $\mu_d = 0$:

```
sim.auction <- do(1000) * mean(~ (dutch - FP), data = swap(Auction, c("dutch", "FP")))
head(sim.auction, 3)

  result
1  0.1142
2 -0.1313
3 -0.1858

dotPlot(~ result, data = sim.auction, groups = (result <= -0.384 | result >= 0.384), nint = 50)
```

Exploration7.3.5b



3. Strength of evidence:

```
favstats(~result, data = sim.auction)

      min      01      median      Q3      max      mean      sd      n missing
-0.2381 -0.05881 -0.0005682 0.0608 0.2301 0.000758 0.083 1000         0

prop(~(result <= -0.384 | result >= 0.384), data = sim.auction)

TRUE
0
```

Exploration7.3.5c

4. t-test for paired samples (theory-based approach):

```
t.test(Auction$dutch, Auction$FP, paired = TRUE)

Paired t-test

data: x and Auction$FP
t = 5.328, df = 87, p-value = 7.692e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
0.2405 0.5266
sample estimates:
mean of the differences
0.3835
```

Exploration7.3.7

```
t.test(~(dutch - FP), data = Auction)
```

One Sample t-test

```
data: data$(dutch - FP)
t = 5.328, df = 87, p-value = 7.692e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
0.2405 0.5266
sample estimates:
mean of x
0.3835
```

Exploration7.3.8

95% confidence interval using the t-test:

```
confint(t.test(Auction$dutch, Auction$FP, paired = TRUE))

mean of the differences      lower      upper
0.3835          0.2405        0.5266
level           0.9500
```

Index

FirstBase2, 153
GameSims., 16
Tintle1, 5, 7
bargraph(), 9
bargraph, 10
binom.test(), 48, 83
binom.test, 90, 95
bwplot(), 9, 66
data(), 6, 7
densityplot(), 9
do(), 16
dotPlot(), 12
do, 16
favestats, 14
freqpolygon(), 19
histogram(), 9, 40
lattice, 14
mosaicplot, 10
mosaic, 5, 7, 15
mosiac, 10
pnorm(), 42, 43
pnorm, 41
prop(), 25
prop.test(), 43, 48, 83
qnorm, 41
rflip(), 15
rflip, 16
rnorm, 66
round(), 66
sample(), 52
sim.sci, 25
summary(), 8
tally(), 8
tally, 9
xpnorm(), 30, 32
xpnorm, 30, 31, 41