Lab Activity

Today’s class will be spent working through Investigation 1.1. I encourage you to work in pairs (but not in larger groups). Discussing the problems with your partner is a good way to make sure you understand the material.

The directions in the text are quite clear and explicit, but ask if you have any trouble. In particular, make sure you understand relative risk and odds ratios. Pay careful attention to the terminology used. (You might like to highlight terms in your book as you come across them.)

You are not required to turn anything in for this lab, but you will have homework problems that cover the material learned.

For Thursday

If you finish early, you can get going on the following things which are due at the beginning of class on Thursday:

- Practice Problems 1.1.1 and 1.1.2.
- Investigation 1.2.1.a–i. (We want to begin at Investigation 1.2.1.j in class on Thursday.)

Other Matters

Problem Set 1

Problem Set 1 will be due next Monday. It is available on the course web page. Please observe the guidelines listed there for all work turned in this class.

Email

You should have received a test message from me yesterday (and probably two other messages as well). Let me know if you have not received your email.

Installing R

If you plan to use R on your own computer for doing homework, I encourage you to install it soon so it is ready to go next week. You should find all the directions you need on the R-project web page (+r-project.org).

Saving your work

If you work in the lab and want to save something, I recommend that you use one of the following two options: You can use Net Storage (see technology information on the course web page) or use a USB drive.
Miscellaneous Notes

- We will use Excel again today. In particular, you will need to use formulas to get Excel to do some computations for you. There is a formula toolbar that can be made visible by selecting it in the View menu. Also, you can (usually) tell Excel that something is a formula (rather than text or a date or something else) by typing = before you type the formula. If you forget, or if this doesn’t work, you can reformat cells by selecting them and then using the menus Format > Cells > Number. You will see lots of other format options there as well.

- I have put a link to brief solutions to the Investigations on the course webpage. Use these wisely. After you have done an investigation or if you are unsure about a particular item, you might do a quick check to see how things compare. But do not consult the solutions too early or too often. If you do you won’t gain the benefits of thinking through the question for yourself.

In Class

Investigation 1.2.1: Smoking and Lung Cancer, Wynder and Graham (1950), pages 10–15

- You should have come to class having already done items a–i. This part was mostly about relative risk and odds ratios.

- The remainder of this investigation is mostly about how studies can and cannot be used to make inferences about cause and effect relationships. The terminology detour is very important.

- Answer Practice Problem 1.2.1 before continuing. (It will not be collected, however.)

Investigation 1.2.2: Smoking and Lung Cancer, Hammond and Horn (1958), pages 15–17

- This describes another study of smoking and lung cancer. As you go through this investigation, be sure to pay attention to how the design of this study differs from the previous one and why that matters.

Exploration 1.2: More Odds Ratios and Relative Risk, pages 17–24

- If you do not finish this in class, finish it on your own before class tomorrow.

What to turn in

- Tomorrow I will collect from you Practice Problem 1.2.3. Label this Lab 1.
Goal: To explore how and why randomization is used in statistical experiments.

Investigation 1.4.1: Foreign Language and SAT Scores (outside of class)

This investigation discusses the advantages (and disadvantages) of a randomized controlled experiment and talks about some of the issues involved (blinding, placebos, etc.). It is a follow-up to our brief discussion of experiments in class last time.

You can read this on your own either before class or after class. Use class time to work on Investigation 1.4.2.

Investigation 1.4.2: Have a Nice Trip, pages 35–42

- You will need to have some playing cards for this investigation (8 of one color and 4 of another). You can get these from me. Please return them when class is over.

- Work in pairs for this investigation. (I don’t have enough cards for you to work individually.)

- In item 1.4.2.d you need to combine data with data from others in the class. We’ll do this by making a dot plot on the board.

- The Randomizing Subjects applet is available from the course webpage. (Look under From Class > Java Applets and Data.)

- Notice the the investigation resumes again after the Practice Problems on pages 39–40.

What to turn in

Label this Lab 2:

- Practice Problems 1.4.2 and 1.4.4
  Turn in one sheet per pair. (Be sure to put both your names on it.) You can turn it in at the end of class today or at the beginning of class tomorrow.

- The playing cards.
  (Don’t put your names on these ☺.)

I recommend that you read and think about the other practice problems as well, but you needn’t turn in your responses.

Summary: Block what you know, randomize the rest.
Goal: To explore how understanding randomness allows us to determine when our data provide evidence that is “statistically significant.”

Investigation 1.5.1: Friendly Observers, pages 42–51

- **Work in pairs** again today.
- You will again need **playing cards** today (11 of black for success; 13 red for failure). Please return them when you are finished.
- The applet you need is available via the course webpage.
- In item r you will be asked to compute a **weighted average**. Here is a simple example of how this works: Suppose that homework problem is graded on a 0–10 scale and the results for a class are as follows:

<table>
<thead>
<tr>
<th>score</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of students</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>relative frequency</td>
<td>0.10</td>
<td>0.20</td>
<td>0.30</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Now let’s compute the average score in the class. First, we could add all the scores and divide by 10 (the number of students in the class). This gives

\[
\frac{7 + 8 + 8 + 9 + 9 + 10 + 10 + 10 + 10 + 10}{10} = \frac{(1 \cdot 7) + (2 \cdot 8) + (3 \cdot 9) + (4 \cdot 10)}{10}
\]

\[
= (\frac{1}{10} \cdot 7) + (\frac{2}{10} \cdot 8) + (\frac{3}{10} \cdot 9) + (\frac{4}{10} \cdot 10)
\]

\[
= (0.10 \cdot 7) + (0.20 \cdot 8) + (0.30 \cdot 9) + (0.40 \cdot 10)
\]

This shows that the average can be obtained by multiplying relative frequencies by the scores and adding. This process is usually called a **weighted sum**. (Our textbook calls it a weighted average, but weighted sum is a better name.)

In any case, this weighted sum gives us a way to compute the average directly from the relative frequencies even if we do not know how many students there are.

What to turn in

This may be turned in at the end of the hour today or the beginning of class next time. Turn in one sheet per pair with both your names on it. Label it Lab 3.

- From Investigation 1.5.1:
  - Your table from item h.
  - Your answers to item j.
  - Your table from item q, and the answer to item r.
- Practice Problem 1.5.1
Goal: To learn the basics of empirical and theoretical probability

- Random processes have two important properties:
  - *unpredictable* in the short run
  - *predictable* in the long run

- Important terms: **outcome** (possible result of a random process), **sample space** (set of all outcomes), **event** (subset of sample space, i.e., a set of outcomes).

- Probability assigns numbers (between 0 and 1) to events indicating how likely they are.

- Empirical probability: repeat the random process (a lot of times). Then
  \[ P(E) = \frac{\text{number of times } E \text{ occurs}}{\text{number of times process was repeated}} = \text{relative frequency of } E \]

  - Because we may get different results if we do this again, we can think of this as an approximation method.
  - The approximation gets better if we do more repetitions. (We’ll learn more about how much better later in the term.)
  - Computers are very useful tools for doing empirical probability since they can generate and tabulate a large number of results quickly.

- Theoretical Probability is based on four rules. Here are the first three:
  - \( P(A \text{ happens}) + P(A \text{ doesn’t happen}) = 1 \) \( [ P(A) + P(A^c) = 1 ] \)
  - \( P(\text{sample space}) = 1 \)
  - If \( A \) and \( B \) can’t both happen at the same time (in which case we say that they are **mutually exclusive**), then
    \[ P(A \text{ or } B) = P(A) + P(B) = 1 \]

- If a sample space consists of \( n \) *equally likely* outcomes, then \( P(E) = \frac{|E|}{|S|} \).

  - This follows from the three rules above.
Goal: To gain experience working with empirical and theoretical probability.

Investigation 1.6.1: Random Babies, pages 51–58

We’ll use this somewhat silly, but hopefully memorable, investigation to practice computing probabilities and to learn about random variables and their expected value. We’ll go through this investigation together as a class. Actually, we covered a fair amount of this already yesterday, so we won’t go through quite it is presented in the text.


Once we complete Investigation 1.6.1, you can work on this investigation either alone or in pairs.

c Once again, “Show tallies” won’t work. You can do this in R, however. I’ll generate a table for you on the screen, but if you want to know how to do this, here are step by step directions. Type each command listed (although only the last is really needed); comments follow the commands.

- `getCalvin('m243')` (followed by enter).
  This will load any code that I have gotten ready for us.

- `1:14`.
  You should see a list of the numbers 1 through 14. We’ll use these to represent the 14 rats.

- `sample(1:14,7)`.
  This will sample seven of the 14 numbers. Think of this as a random assignment of rat numbers to group A.

- `sample(1:14,7) -> groupA`.
  This stores a sample in the variable named `groupA` (Type `groupA` to see it.)

- `sample(14,7) -> groupA; groupA; groupA <= 10; sum(groupA <=10)`
  The semicolon separates commands. (You could also put these on separate lines.) Since R treats TRUE as 1 and FALSE as 0, the last command counts the number that are less than or equal to 10 in our sample, i.e., the number of successes in group A. Use up arrow to repeat this a couple times.

- `replicate(10,sum(sample(1:14,7) <= 10))`.
  This does the sampling 10 times and displays the number of successes in group A for each.

- `table(replicate(1000,sum(sample(1:14,7) <= 10)))`.
  For a larger number, we don’t really want to see all the numbers, we want to tabulate them. That’s pretty complicated for your first day using R. Congratulations. Fortunately, most of the things we will do with R are much less complicated.

d–f You can calculate \(^7C_4\) in R using the command `choose(14,7)`.

? You can get help with R commands by typing something like `?choose` or `?sample`.

Nothing to turn in today

Enjoy your weekend.

©2006 Randall Pruim (rpruim@calvin.edu)
Goals:  
1. To learn the mathematics of the **hypergeometric distribution**.  
2. To learn how the hypergeometric distribution can be applied as **Fisher’s Exact Test**.  
3. To become acquainted with the software package **R** and use it to automate certain calculations.

**Investigation 1.7.1: More Friendly Observers, pages 63–72**

- We’ll cover the material through item j together as a class. This introduces the mathematics of the **hypergeometric distribution**.

The remainder of Investigation 1.6.1 uses the hypergeometric distribution to develop **Fisher’s Exact Test**. We’ll use R in place of Minitab throughout.

- Begin by loading code for this class: `getCalvin('m243')`

- We can check our “hand” calculations using the R command `dhyper()`. To do this we need to provide four numbers, the value for which we want the probability (for example, $X = 3$ successes in group A), the three parameters of the hypergeometric distribution ($m = 11$ successes, $n = 13$ failures, and $k = 12$ subjects in group A). The complete command is `dhyper(x=3,m=11,n=13,k=12)` or `dhyper(3,11,13,12)`. In R, you are not required to use the labels (x, m, n, and k) if you put them in the expected order. You should see this order displayed for you when you type `dhyper()`.

- You should make the necessary modifications to this command to get the probabilities that $X = 0$, $X = 1$, and $X = 2$ as well.

\[ \text{Calculating cumulative hypergeometric probabilities.} \]

You could add the numbers you obtained above by hand (or using R as a calculator), but there are better ways to do it.

- You can get all the probabilities you need at once: `dhyper(0:3,11,13,12)`. (0:3 tells R to generate a list of the numbers from 0 to 3.)
- `cumsum(dhyper(0:3,11,13,12))` will give the **cumulative sums**.
- But the easiest way to get $P(X \leq 3)$ is `phyper(3,11,13,12)`. (Think of ‘d’ for distribution or densities and ‘p’ for p-value).

\[ \text{Displaying the entire hypergeometric probability distribution.} \]

- To store the values of $X$ in a variable called `value`, type `value <- 0:12`
- To store the probabilities in a variable called `prob`, type `prob <- dhyper(value,11,13,12)`.
- Now make the plot by typing `xyplot(prob~value)`.

In general, `xyplot(y~x)` will plot the values of y against the values of x, placing y along the vertical axis and x along the horizontal axis. This command has many additional options they we will explore later. If you are working with your own copy of R, you may need to first issue the command `library(lattice)` to make `xyplot` available to you.

©2006 Randall Pruim (rpruim@calvin.edu)
To see the numerical values of \( \text{prob} \), simply type \( \text{prob} \) (and return).

Why is one of these values 0?

A value like 4.8074e-06 means \( 4.8074 \cdot 10^{-6} \). If you want more or fewer digits, you can use \( \text{round(prob,6)} \), changing 6 to whatever number of digits you desire.

**q** sum of probabilities

- You can get the sum of the probabilities by \( \text{sum(prob)} \).
- How could you use \( \text{phyper()} \) to get the sum you want?

**r** Calculating the expected value.

- In R, if you multiply two lists of numbers, you will get a new list of numbers formed by multiplying the first numbers, the second numbers, the third numbers, etc. from the two lists. What should you get if you type \( 1:3 \ast 5:7 \)? Now try it and see if you were correct. Try a few more examples until you are sure you understand what is happening here.
- Now the expected value of \( X \) is easy to calculate (using the variables \( \text{prob} \) and \( \text{value} \) from above and the command \( \text{sum()} \). Can you figure out how?

**s** There is a missing blank here. It should be \( P(Y \geq 8) = 1 - P(Y \leq \boxed{\phantom{0}}) \)

**t** You can get the desired probability either using the complement method as described, or you can get it directly using \( \text{phyper}(7,11,13,12,\text{lower.tail}=\text{FALSE}) \). The \( \text{lower.tail}=\text{FALSE} \) part says to take the probabilities of values above rather than below.

**u-z** The effect of sample size.

If you have understood what we have done to this point, you should be able to do item u through z in a similar way.

- As you might guess, R has a method of doing the Fisher exact test directly.
  - To do it, we first need to put the data into matrix form: \( \text{observers<-matrix(c(3,8, 9,4), nrow=2, byrow=TRUE)} \). (Actually there is another way, too, but we’ll only look at one way now.)
    - Type \( \text{observers} \) to see what this does. The \( \text{c()} \) command stands for concatenate and makes a sequence of the four numbers; \( \text{matrix()} \) makes this 2-dimensional.
  - Now we can do the test: \( \text{fisher.test(observers)} \).
    - Look through the output and find the p-value. As is it won’t agree with what you have calculated earlier. Can you figure out why not? [Hint: \( \text{dhyper(0:12,11,13,12)} \) might help.]
  - To get the p-value we have been computing, try \( \text{fisher.test(observers,alternative='less')} \).

**Investigation 1.7.2: Minority Baseball Coaches, pages 72–74**

This investigation covers another example of Fisher’s exact test. The methods of computing the p-values are the same, but notice that the design of the study is different. In particular, this study is an observational study rather than a randomized experiment. For this reason, the interpretation is somewhat different.

**What to turn in**

- Practice Problem 1.7.1 should be turned in by Thursday, February 16.
Goals:
1. To begin thinking about the distributions quantitative variables.
2. To learn about some useful numerical and graphical summaries for quantitative variables, including the following: mean, median, mode, skew, five number summary, histogram, dotplot, density plot, boxplot, interquartile range and outlier.
3. To learn how to get R to calculate numerical and graphical summaries of quantitative data.

Investigation 2.1.1: Class Survey, pages 102–103
We’ll do this together as a class, but it will involve some discussion with your partner(s).

Investigation 2.1.2: Cloud Seeding, pages 103–110
This investigation leads you through most of the terms listed in the goals at the top of the page. Depending on your background some or even many of these may be familiar to you, but make sure you ask if any of these terms or ideas are not clear to you.

Most of this investigation does not require R, but when you get to the “Minitab Detour”, you can do the “R Detour” below, then continue with the last bit on transformations.

R (Scenic) Detour
So how do we get R to generate the pictures and numbers like the ones we’ve seen today? Here’s how.

Loading Data
- getCalvin('m243')
  - This loads code I have made available for the class, including some nice shortcuts for loading data sets that accompany our text.
- getData('iscam/cloudseeding') -> clouds
  - All the datasets from the book will begin iscam (abbreviated title of the book).
  - I have used the same file names as the book uses, but there is no capitalization and no file extension is necessary.
  - This command only does you any good if you store the results somewhere. That’s what -> clouds is doing.
- clouds
  - This will let you see the data. If it’s a big data set, it will just fly by and not be too useful.
  - You can look at just the first few lines with the command clouds[1:10,].
  - Another important command is names(clouds). This will list the names of all the variables in the data frame (a data frame is what R calls the format most commonly used for data sets).
Summarizing the Data

Now we want to look at some numerical and graphical summaries of the data.

- I just installed a new version of R and the package **Hmisc** this morning. Check that things are working as they should be by typing the command below to see if you get the same output I do.

  ```R
  > summary(rainfall~treatment,data=clouds,fun=quantile)
  rainfall  N=52
+-----------------------------------------------+--------+-----------------+-----------------+----------+----------+----------+
|              |          | N | 10% | 25% | 50% | 75% | 100% |
+-----------------------------------------------+--------+-----------------+-----------------+----------+----------+----------+
| treatment | seeded  | 26| 4.1 | 98.125| 221.6 | 406.02| 2745.6|
|           | unseeded| 26| 1.0 | 24.825| 44.2  | 159.20| 1202.6|
+-----------------------------------------------+--------+-----------------+-----------------+----------+----------+----------+
| Overall   |         | 52| 1.0 | 28.900| 116.8 | 307.40| 2745.6|
+-----------------------------------------------+--------+-----------------+-----------------+----------+----------+----------+
```

R uses a different algorithm to find the quartiles than the one described in the book. Actually, it has two different algorithms. Try the following: `rbind(quantile(1:11), fivenum(1:11))`.

- You can experiment with different sets of numbers to see when the two agree and when they differ. If time permits, I’ll tell you how the algorithms work and why they differ.

- `bwplot(rainfall~treatment,data=clouds)` and `bwplot(treatment~rainfall,data=clouds)`

  These make box and whisker plots. Notice that this uses the y-x format that we have seen before.

- `histogram(~rainfall|treatment,data=clouds, layout=c(1,2))`

  This makes a histogram of rainfall for each treatment group. Notice that there is nothing to the right of ~. That’s because for a histogram, the value on the y-axis is computed; it’s not part of the data you provide. Instead, we have a new element ~ to separate the variable to be plotted from the “conditioning” variable. We’ll learn more about this syntax as the course goals along.

  The layout option says we want one column and two rows. Other options you might like to try:

  - `n=30` will make a histogram with (approximately) 30 bins. You can adjust this number to get what you think is the most descriptive picture.
  - `main='My favorite graph title'`
  - `breaks=seq(-25,3000,by=50)`

    This sets the breakpoints between the bins. (Type `seq(-25,3000,by=50)` to see what this does.) This particular choice will match the bins used in the dotplots on page 105.
  - `col='blue'`

- `dotPlot(~rainfall|treatment,data=clouds, layout=c(1,2))`

  **Notice the capital P.** The R command **dotplot** doesn’t make what our book calls a dotplot, so I wrote a command to do the job. It was loaded when you did `getCalvin('m243')`. The options for `dotPlot()` are similar to those for `histogram()`. Let me know if you notice any funny behavior, since `dotPlot()` hasn’t been thoroughly tested.

---

1 `rbind()` binds the results as rows, one above the other. There is also a command called `cbind()` to bind things as columns.
densityplot(~rainfall|treatment, data=clouds, layout=c(1,2))

This is a “smooth” version of a histogram. There are options to control just how the smoothing is done, but we won’t worry about that for the moment.

Investigation 2.1.2: Cloud Seeding (continued), page 112

You can add new variables to data frames in R quite easily. The easiest way is to form a new variable by transforming an old one:

- clouds$logRainfall <- log(clouds$rainfall)

The $ selects a single variable from a data frame. Try clouds[1:10,] to see the first 10 rows of the data frame now.

- On the other hand, it usually isn’t necessary to store these transformed variables, since we can do the transformations “on the fly”. The following commands should yield the same plots:

  histogram(~logRainfall|treatment, data=clouds, layout=c(1,2))
  histogram(~log(rainfall)|treatment, data=clouds, layout=c(1,2))

The same is true for the other plots.

Nothing to turn in, but ...

I recommend that you do Practice Problem 2.1.1 on your own and ask if you have any difficulties.

How do I do this at home?

If you want to use getCalvin() from home (with an internet connection), you need to tell your copy of R what the command getCalvin() is. The easiest way to do this is to copy the Rprofile file to your computer. The fancy way to use it is to make that file your Rprofile file (exact details about what this file is called and where it is located depend on the platform of your machine). If you put it into your Rprofile file, it will load each time you start R. Another options is to store the file (with any name you like) and “source it” when you want to load it: source('Rprofile.txt').

My Rprofile file is now available on the Tech Info page. Once this file has been loaded, you can use getCalvin('m243') which loads the other code which I prepare for this class and you should be good to go.

In addition to the basic functionality that loads each time you start R, you can install and load additional packages from a public repository of R code (visit www.r-project.org). There are a number of packages loaded on the machines in the lab. You can use the package manager to see the list. I’m hoping to add Hmisc to the list sometime soon.

Packages need to be installed once. Many version of R have menu options to do this (provided you have enough permissions, which you do not in the lab). Or you can do it from the command line with install.packages(). Use ?install.packages for details. To use a package, you also need to load it each time you start R. This is done with the library() command. For example, library(Hmisc) would load the Hmisc package if it were installed.
Goals: 1. To learn about several measures of spread of a distribution, especially interquartile range and standard deviation.

2. To learn about standardization (z-scores).

We will work through Investigations 2.1.3 and 2.1.5 as a class lecture/discussion rather than in groups, but I am including R notes in case you want to go back and do things on your own and want some help with the R commands.

Warning

There are a few places in these sections where the text is a bit sloppy and says some things that are not quite true. In particular:

- The Discussion on page 122 is just wrong. It is not enough for the distribution to merely be symmetric. This is only true for a certain class (of very important) distributions, but not in general.

- The Empirical Rule on page 126 is only true for distributions that are (approximately) normal. There are mound-shaped, symmetric distributions that are not normal, and these estimates really don’t apply to those. We’ll have become much more familiar with the normal distributions over the course of this semester.

Some R Notes before we begin

- For old R commands, I will be giving less and less detail, so as new commands are introduced, make sure you understand how they work so you can use them again in the future.

- It would be a good idea to keep your old handouts and bring them with you for reference.

- Don’t forget about the reference cards I handed out. They are handy for quick reference. Note that we have been using lattice graphics and not the base graphics. (Lattice graphics are more flexible.)
Investigation 2.1.3

- Load the course stuff: `getCalvin('m243')`
- Load the data:
  - `getData('oldfaithful') -> faith1`
  - This file has an unfortunate format. Take a look with `faith[1:10,]`. Usually we like each row to correspond to one unit/subject. In this case we have three basically unrelated columns. Let’s make a new version that fixes this problem as follows: `stack(faith1) -> faith2`
  - Take a quick look at the top of the two formats to see what’s going on: `faith1[1:10,]; faith2[1:10,]`
  - The names in our stacked version are not so nice. Let’s change them: `names(faith2) <- c('time','year').² Now repeat the previous command (up arrow) to see the difference.

b Use `bwplot()` to make boxplots. You probably want to use `faith2`.

d You can use `summary()` to help with the calculations.

e You can sort data in R using `sort()`. For example, `sort(faith1$X2003)`

g Estimate from the boxplots, then confirm using `summary()` (with the `fun` parameter set to `quantile`).

j Try `summary(time~year,data=faith1,fun=sd)` or `summary(time~year,data=faith1,fun=var)`.
  Or get really fancy (optional) and try `summary(time~year,data=faith1,fun=function(x) { c(sd(x),var(x)) })`

l R is not particularly good for editing data. (In real statistical applications you usually don’t get to “edit” the data once you begin analysis, I suppose.) You can hand edit a data frame with `edit(faith2)`, but you can also just work with a subset of the data, and that is something R does do nicely. Try this `summary(time~year,data=faith1,subset=(time>60),fun=sd)`
  Note: There is a minor typo in the text; the second shortest time is 58, not 57. Since there are no times between 58 and 71, using a cutoff of 60 works fine.

m Can you figure out good ways to get R to help you with this? (Hints: There are many options including using `sort()`, `summary()` with the `subset` parameter, `dotPlot()`, …)

n–p You already know how to make histograms. If you don’t want all of the data, you can use `subset`, or do something like `histogram(~faith1$full2003)`.

Investigation 2.1.4

This is a pencil and paper investigation.

Investigation 2.1.5

We have seen almost all of the R stuff you need to know for this investigation.

h To convert the temperatures from Fahrenheit to centigrade and store the results in a new variable called `centigrade` use the command `temps$centigrade <- (5/9)*(temps$body.temp - 32)`.

²You can also change the names one at a time: `names(faith2)[1] <- 'time'; names(faith2) <- 'year'.

©2006 Randall Pruim (rpruim@calvin.edu)
Investigation 2.1.4: Hypothetical Quiz Scores

This investigation is primarily a check that you can correctly compute the IQR of a data set and that you are correctly interpreting histograms and standard variation. These are important concepts, so be sure to ask if you have any questions.

Investigation 2.2.1: Sleep Deprivation and Visual Learning, pages 141–148

In Chapter 1 we looked at statistical significance in the context where both the explanatory and response variable were categorical (e.g., Friendly Observers). Section 2.2 investigates statistical context in a new setting.

- Follow the usual start-up routine: `getCalvin('m243')`

a–i These items make use of physical simulation with some “cards”. You can get some small index cards from me. Write on the lined side of the cards, and return your set when you are with the simulation. When you get to item h, record your results on the dotplots on the board.

j Open the data set, give it a name, and take a quick look at it:

```
- getData('iscam/sleepdeprivation') -> sleep; sleep.
- summary(improvement~sleep.condition, favstats)
```
This will give some useful summary statistics. `favstats()` is a function that I defined. It gives a five number summary (using `quantile()` and the mean and standard deviation of the data.

k Writing a function in R; using external files.

We want to simulate randomization using R (so we don’t have to keep shuffling the cards). We will automate this process by defining a function. We’ll store this function in an external file, something that will be useful when we do more complicated things, or when we just want to save our work.

Our function will perform one randomization and compute the difference in the means of the two groups, just like we did by hand with the cards.

- Open a connection to your network space. (See the course web page for instructions.)
- Tell R to work in a directory in your network space using Misc > Change Working Directory.
- Open up the editor in R using the R menus: File > New Document.
- In this editor window we will write code for our function, named `simulate()`. Enter the following in that window (the end of the lines beginning with # are comments and can be omitted.) This defines a function that takes two parameters, `data` and `size`. The second of these has a default value (11) which it will assume if no value is given.

```r
simulate <- function(data, size=11) {
  k <- length(data);            # k will be 21 in our example
  group1 <- sample(1:k,size);   # this selects a subset of the indices
  group2 <- (1:21)[-group1];   # this is the unselected indices
  mean1 <- mean(data[group1]);
  mean2 <- mean(data[group2]);
  return(mean2-mean1);         # this is the result of the function
}
```

©2006 Randall Pruim (rpruim@calvin.edu)
– Use File > Save as to save this file in the directory you selected above. I suggest you give it a name that ends .R, something like invsetigation2-2-1.R or simulate.R.

– Now have R read the commands in your file: source('simulate.R'). (Change the file name as necessary.)

– Now you should be able to use your new function just like the built in functions we have been using. Give it a try: simulate(data=sleep$improvement,size=11). Do it a couple more times. (Use up arrow.) You will probably get different values each time.

• Viewing code for functions.

For many of the functions in R, you can view the code. This can be handy for figuring out how to do things or if you want to write a new function that is similar to one that already exists. To see the code, simply type the name of the function (without the parentheses). For example, type favstats to see how I wrote that function, or type simulate to see your simulate() function. Be warned, however, that sometimes the functions are complicated because they are written to handle very flexible types of inputs, or are merely wrappers around other functions that do all the work, and so the code is not always very enlightening.

l Repeating the simulation many times.

replicate(1000,simulate(data=sleep$improvement,size=11)) -> meandiffs This will repeat the simulation 1000 times and store the results. (Type meandiffs if you want to see them.)

Make a histogram (histogram(~meandiffs)) to take a look at the distribution, and describe the distribution, paying careful attention to the difference in the means in the actual data.

m Use sum(meandiffs > 15.92) in place of the Minitab.

p You can use choose() to do the calculations.

Investigation 2.2.2: More Sleep Deprivation, pages 148–150

This is a comparison of the data in the pervious investigation with some hypothetical data. Load the data and reuse your simulate() function to do 1000 simulations based on the hypothetical data.

Investigation 2.2.3: Lifetimes of Notables, pages 150–152

a Compute the quartiles by hand. Once you open the data set (in item d) you can check your answers using summary().

e You don’t need to modify your function, simply use with the correct parameters values for data (whatever you call the data this time) and size (20, because you want to randomly pick 20 to put into one of the groups).

What to turn in

Practice Problem 2.2.1 is due Friday, February 24. Call it Lab 5. If you work in pairs, you only need to submit one paper for the pair.
Goals: Today we will will
1. consider sampling from a process instead of a population.
2. learn how to calculate the variance and standard deviation of a random variable.
3. learn about Bernoulli random variables.
4. learn about Binomial random variables.
5. encounter the fourth probability rule, the Multiplication Rule.

Pop Quiz

We will begin today with a pop quiz. Each multiple choice item has 4 options, one of which is correct (A, B, C, or D). Record your answers on the blanks below:

1. ________ 2. ________ 3. ________ 4. ________ 5. ________

Investigation 3.2.1

We will proceed through this investigation together as a class.

Investigation 3.2.2

We will start this together, then you may work at your own pace. I encourage you to work with a partner.

• The Simulating Binomial Probability applet is in the usual place.

  Notice that we are extending our definition of parameter.

  Use R to get the probabilities. The function you need is called dbinom(). To get the probabilities for our `pop quiz` example, type `dbinom(0:5,size=5,p=.25)`. For a nicer looking output, try `cbind(0:5,dbinom(0:5,size=5,p=.25))`.

  It is easy to compute the expected value in R:

  – Store the probabilities by typing `dbinom(0:5,size=5,p=.25) -> prob`
  – Store the values by typing `0:5 -> value`
  – Check that things look correct by typing `value` and `prob`
  – What does this do? `value * prob`
  – How do you get the expected value from this information?
  – Questions? Try typing `args(pbinom)` or `?pbinom`

  You can use `pbinom()` in the same way you used `phyper()` in Chapters 1 and 2.

  Inverting is done using `qbinom()`. `q` stands for quantile, a generalization of quartile to any percentage you like. Can you figure out how to use `qbinom()`?

©2006 Randall Pruim (rpruim@calvin.edu)
Goals:

1. To learn how to use the binomial distributions as part of statistical inference.

2. To learn the formal notation and terminology used for hypothesis tests and confidence intervals.

Summary of Binomial Distributions (including R commands)

Suppose $X \sim \text{Binom}(n, \pi)$ (read: “$X$ is a binomial random variable with parameters $n$ and $\pi$”). We will make repeated use of the following facts about the binomial distributions:

- $P(X = x) = \binom{n}{x} \pi^x (1 - \pi)^{n-x}$. (For example, if $X \sim \text{Binom}(10, .3)$, then $P(X = 2) = \binom{10}{2} (0.3)^2 (0.7)^8$.)

- $E(X) = n\pi$.
  Here’s how we get this formula: For any random variables $Y$ and $Z$, $E(Y + Z) = E(Y) + E(Z)$. Repeating this argument for a sum of $n$ random variables tells us that the expected value of a binomial random variable should be $n$ times the expected value of a Bernoulli random variable (which we showed was $\pi$).

- $\text{Var}(X) = n\pi(1 - \pi)$; $\text{SD}(X) = \sqrt{n\pi(1 - \pi)}$.
  If $Y$ and $Z$ are independent random variables, then $\text{Var}(Y + Z) = \text{Var}(Y) + \text{Var}(Z)$. Repeating this argument for a sum of $n$ random variables tells us that the variance of the binomial random variable should be $n$ times the variance of a Bernoulli random variable (which we showed was $\pi(1 - \pi)$).

- $\text{dbinom}(x,n,\pi) = P(X = x) = \binom{n}{x} \pi^x (1 - \pi)^{n-x}$.

- $\text{pbinom}(x,n,\pi) = P(X \leq x)$.

- $\text{qbinom}(p,n,\pi)$ gives the $p$-quantile of the binomial distribution, that is, the largest number $a$ such that $P(X \leq a) \leq p$.

- $\text{rbinom}(100,n,\pi)$ will generate 100 random values using the binomial distribution.

- Example plots: $\text{xyplot}(\text{dbinom}(0:10,10,0.25) \sim 0:10)$ or $\text{plot}(\text{dbinom}(0:10,10,0.25))$.

Investigation 3.3.1: Water Oxygen Levels; pages 215–219

We will work through this example together in class. This investigation formally introduces using binomial distributions for inference. The terminology in this section is very important.

Investigation 3.3.2: Heart Transplant; pages 219–223

This example is similar to the one in Investigation 3.3.1 (but do take note of the differences) and will give you a chance to try your own hand at binomial inference.

What to turn in

Do practice problem 3.3.1 for Friday.
Goals:
1. To get more practice doing hypothesis tests.
2. To learn how to do a two-sided test and how this differs from a one-sided test.

Investigation 3.3.4: Holloween Treat Choices; pages 226–231

What’s new: Two-sided p-values vs One-sided p-values.

Some of you had questions about this issue already in Chapters 1 and 2. We didn’t spend the time to develop formal notation and terminology to handle this topic carefully then, and I promised we would return to it. Now we are.

You may use either R or the applet to get your binomial probabilities. (The numbers are large enough in this example that you probably don’t want to use your calculator unless it has functions that do essentially the same thing as the one’s we have encountered in R).

Investigation 3.3.5: Kissing the Right Way; pages 229–231

What’s new: Two-sided p-values when \( p_0 \neq 0.5 \).

By Monday at the start of class you should have worked through parts a–d. We’ll continue from there on Monday, but I encourage you to read through page 231 ahead of time.

Investigation 3.3.3: Do Pets Look Like Their Owners? pages 224–226

What’s new: Using two binomial distributions in the same problem (to do different things).

This section is optional, but I encourage you to work through it, especially if you feel insecure about either the binomial distribution or using the binomial distribution to get a p-value.

This investigation does not introduce anything new in terms of hypothesis testing itself, but it offers another example and it uses the binomial distribution twice.

Remember that you can check your answers from a link on the course web page.
Investigation 3.3.5: Kissing the Right Way (2-sided p-values); pages 229–231

Page 231 discusses methods for defining two-sided p-values.

Investigation 3.3.6: Kissing the Right Way (confidence intervals); pages 232–235

There is an important typo on page 235. The description of a confidence interval should read:

\[
C\% \text{ confidence Interval for } \pi.
\]

The set of values such that the two-sided p-value based on the observed count is larger than the \((1 - C)\) level of significance.

The text has \((1 - C)/2\) where it should have \((1 - C)\). Please fix this typo in the your book.

Some Useful R Code

As you might expect, R has ways of automating these calculations of p-values and confidence intervals. The function to use is called \texttt{binom.test()}. Here are some examples of how to use this function with the kissing data:

- the default version tests the null hypothesis that \(\pi = 0.5\) using a two-sided alternative and reports a 95% confidence interval: \texttt{binom.test(80,124)}
  - R does not use either the “twice the 1-sided” or “small p-values” method to determine 2-sided p-values, so the results will not always agree with the p-values you compute using one of these two methods.

- To get a 1-sided p-value:
  - \texttt{binom.test(80,124,alternative='greater')}
  - \texttt{binom.test(80,124,alternative='less')}

- To change the null hypothesis to 2/3: \texttt{binom.test(80,124,p=2/3)}

- To change the confidence level to 99%: \texttt{binom.test(80,124,conf.level=0.99)}

- These options can be combined as needed.

Investigation 3.3.7: Improved Batting Averages; 236–244

This section introduces the important concepts of type I error, type II error, and power.
Goals: 
- To learn why we can often use a binomial distribution in place of a hypergeometric distribution, and why this is a useful thing.
- To be introduced to *conditional probability*.
- To see some more examples of hypothesis tests and confidence intervals.

Before We Start

In October, 2003, a nation-wide poll was conducted. One of the questions asked if people “feel good about the quality of your life overall”. Take a guess for proportion of American adults that would answer “yes” to this question. Record your guess.

Guess for proportion who feel good about their life overall: ________

**Investigation 3.4.1: Sampling Words; pages 246–250**

We’ll cover this section as a class discussion.

**Investigation 3.4.2: Feeling Good; pages 250–253**

This investigation will give you a chance to practice what we discussed in Investigation 3.4.1.

c Use your recorded guess from above.

f You can use `binom.test()`, if appropriate. (See yesterday’s notes for examples.) Else use the functions for the hypergeometric distribution.

h Record your null hypothesis, the p-value, and whether you reject the null hypothesis in the table on the board.

i Use `binom.test()`.

**Investigation 3.4.3: Long-Term Effects of Agent Orange; pages 254–256**

You should have all the tools you need to complete this investigation.
Goals:
1. To encounter the normal probability density and see how well it models various data sets.
2. To learn how normal probability plots and normal quantile plots can be used to show how well or poorly a distribution agrees with a normal model. (Similar plots can be used to compare with other distributions, too.)

Investigation 4.1.1: Potpourri; pages 280–285

- As always (I’m going to stop reminding you) load the course stuff using `getCalvin('m243')`
- Load the data using `getData()`. Don’t forget to save the data. Remember our naming conventions. (I’m also going to stop reminding you how to load data.)
  (Example: `getData('iscam/misc') -> misc; I’ll assume the name `misc` from now on, but you may use whatever you like.)
- Look at the top of the data with `misc[1:10,]` or use `names(misc)` to see the names of the variables. For more details use `describe(misc)`. (`describe()` is a function in the `Hmisc` package that loads when you execute `getData('m243')`).

  a To get a histogram of a particular variable, do something like `histogram(~percent,data=misc)`.

  b Opening new graphics windows. There are four ways:
    - Click on the icon at the top of the console
    - Type `quartz()`. (Quartz is the Mac window manager, you would need a different command in different operating systems.)
    - Type `shift-apple-N` (all three keys held down at once).
    - Choose the appropriate option in the `Window` menu.

  c You can reactivate an old window by selecting the window and then
    - typing `shift-apple-A`, or
    - selecting the appropriate item in the `Window` menu.

  It is also possible to do this using the command `dev.set()`.

  d I’ve written a function `xhistogram()` that adds some extra features to histograms. In particular, it can overlay a normal (or some other) density over a histogram. The syntax is basically just like for `histogram()`. To get the normal density added, simply add the parameter setting `density=T` to the argument list: `xhistogram(~percent,data=misc,density=T)`.

  e You can get the heights of the bars labeled using `xhistogram()` with `labels=T`.

  You can also control the break points of the histogram bins (so you will know just how wide they are) by adding something like `breaks=seq(0,100,by=5)` to the `xhistogram()` command. Be sure that the first number is less than the smallest data value and the second number larger than the largest.

  e,f Repeat the process for each of the 8 variables
You should return to read pages 282–284, but for now head to Investigation 4.1.2. Pages 282–284 talk about places models arise in statistics and give several examples of important models. There is a minor typo on page 284: The last sentence should end “$\alpha = 1$.”

**Investigation 4.1.2: Body Measurements; pages 285–287**

a. You should know how to do this now.

b. R makes a plot called a *quantile-quantile* plot that is similar to a probability plot.
   - To get an (unadorned) quantile-quantile plot, use `qqmath(~wrist.diameter, data=body)` or `qqmath(body$wrist.diameter)`
   - I’ve written `xqqplot()` to do the same thing but add a useful reference line. (Sorry no bands – at least not yet.)
   - I have also added a probability plot to our class functions: `probplot(body$wrist.diameter)`. This function is not as flexible, but it does show you what the book describes. What differences do you notice between the qqplots and the probability plot? (Hint: look at the axis labels.)
   - The main story for both kinds of plots is the same: a linear plot indicates data that fit the model well. Deviations from linearity indicate deviations from what the model would predict.

c. A useful way to think about qqplots and probability plots is to ask “How would I need to move the data points to make them all lie on (or near) the line?” Movement is only allowed in the direction (either horizontally or vertically – it depends on which axis is used for what purpose) that corresponds to changing the value of the data. To move in the other direction would require adding or deleting data, which is harder to think about.

d. In the data set the variable name is `biliac.diameter` (one less ‘i’).

g. R’s lattice graphics has some wonderful features for *splitting up plots into subplots*. The syntax uses the pipe symbol (|) to specify the variable(s) used to split the plot into separate “panels”. Try these (filling in the . . . appropriately):
   - `histogram(~forearm.girth|gender,...)`
   - `xqqplot(~forearm.girth|gender,...)`

• An asside: If you want a quick picture of all the variables in a data frame, try `hist.data.frame()`, another Hmisc function.

h. Generate random data in R using `rnorm(100, mean=10, sd=5)` -> `random`. (The same sort of thing works for other distributions as well: `rhyper()`, `rbinom()`, etc.)

i. You will probably find it handy to give your four sets of random values different names. (Or see the fancier way described below.)

• I have written a function called `multirandom()` that will generate multiple samples from a distribution and store them in a data frame. For example, try the following:
   - `multirandom(n=100, samples=9, mean=10, sd=5)` -> `randata`
   - `xqqplot(~value|sample, data=randata)`
   - `xhistogram(~value|sample, data=randata, density=T)`
Goals:

1. To review the use of discrete and continuous pdf’s as models.
2. To learn some basic facts about the normal probability model.
3. To see some examples where the normal probability model is a good approximation.

Investigation 4.2.1: Fuel Capacity

We will be skipping this example, but covering the information about normal distributions contained in it using different examples. Feel free to work through this example on your own for more practice.

Investigation 4.2.2: Body Measurements

Many physical measurements of animals and plants are modeled well by a normal distribution. We will use the example of heights of young American adults to learn about the normal distribution. Useful R commands:

- `xhistogram()`: My extension of R’s `histogram()` function makes it easy to add a graph of the density function to a histogram.
  
  Example: `xhistogram(~height|gender, data=heights, fit='normal'). fit='exponential'` and `fit='gamma'` work too.

- `xqqplot()`: My extension of R’s `qqmath()` function, both of which can be used to see how well normal distribution fits a data set.
  
  Example: `xqqplot(~height|gender, data=heights).

- `pnorm()`, `qnorm()`, `dnorm()`, `rnorm()`: These work just like the functions we have already seen for other distributions (like `pbinom()`), for example.

- `xpnorm()`, `xqnorm()`: I have written extensions to two of the functions above that will draw graphs as well as return the numerical results.

Exploration 4.2: Birth Weight

This exploration is designed to be used with an applet, but we will work through it using R instead. You are welcome to explore the applet on your own.

Lab 7: What to Turn In

To make sure you are understanding how to work with the normal distribution, do Practice Problem 4.2.4 on page 301. Label it Lab 7 and turn it in tomorrow, March 28.

- Be sure to include sketches.
- Use R to get the numerical results. Write down the R commands you use and the values you get. (You don’t need to copy and paste from R to another file – hand-copying is fine.)
Goals:

1. To explore the meaning of “confidence level” in terms of coverage rate.
2. To explore the effects of confidence level, sample size, and population proportion on confidence intervals.
3. To compare Wald confidence intervals and adjusted Wald confidence intervals (based on the Wilson estimator).

Before We Start

Answer the following question: If someone tells you that they have some good news and some bad news to tell you, which would you prefer to hear first, the good news or the bad news? Check the appropriate box below.

☐ Good News First  ☐ Bad News First

Exploration 4.3: Reeses’ Pieces; pages 327–331

This exploration uses an applet to simulate drawing many random samples and computing a (Wald) confidence interval for each sample. Pay careful attention to goals 1 and 2 as you work through this exploration.

Investigation 4.6: Good News First or Bad News First; pages 331–335.

This investigation compares the Wald and adjusted Wald confidence intervals.

Getting R to do the Work

The good news: R can easily compute confidence intervals for proportions. The bad news: the Wald method we have been using is not one of its built-in methods.

- `binom.test()` will compute confidence intervals based on the binomial distribution.
- `prop.test()` uses another method that we have not studied.
- We can do our Wald method manually:

```r
> p <- 80/124; p + c(-1,1) * qnorm(.975) * sqrt(p * (1-p) / 124)
```

```r
[1] 0.56095 0.72938
```

What to Turn In

- Practice Problem 4.3.10
- Practice Problem 4.3.11
Rules for Combinations of Random Variables

Let $X$ and $Y$ be random variables and let $a$ and $b$ be numbers. Then

1. $E(aX + b) = aE(X) + b$

2. $E(X + Y) = E(X) + E(Y)$

3. $\sigma_{aX+b} = |a|\sigma_X$

4. If $X$ and $Y$ are INDEPENDENT, $\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2$
   (Note: this rule uses variances.)

5. If $X$ is normally distributed, then so is $aX + b$.

6. If $X$ and $Y$ are normally distributed and independent, $X + Y$ is normally distributed.

Example. Do tall men and women tend to marry each other? Or are the heights of spouses independent? We have already seen that the heights of men ($M$) and of women ($W$) are approximately normally distributed. If the heights were independent, the $M - W$ is approximately normal, too, so we could use the rules above to determine the probability that the husband is taller than the wife, $P(M - W > 0)$. If we had data on heights of couples, we could compare these theoretical values with the observed values to see if spouse selection seems to be independent of height.

Some Problems

1. If $A \sim N(20, 4)$, and $B \sim N(15, 3)$, and $A$ and $B$ are independent, determine the following probabilities:
   
   a) $P(A + B < 35)$
   b) $P(A - B < 5)$
   c) $P(A + B < 30)$
   d) $P(A - B < 10)$
   e) $P(2A + 3B > 100)$
   f) $P(3A - 2B > 0)$

2. Let $M$ be the height of a random American male. Let $W$ be the height of a random American female. Use the values in Practice Problem 4.2.4 to determine $P(W - M > 0)$ (woman taller than man) under the assumption of independence.
Recall from last time:

<table>
<thead>
<tr>
<th>Central Limit Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $X_1, X_2, \ldots, X_n$ be a simple random sample of size $n$ from a population with mean $\mu$ and standard deviation $\sigma$. Then we know several important things about the sampling distribution of the sample mean $(\overline{X})$:</td>
</tr>
<tr>
<td>1. $E(\overline{X}) = \mu$ (so the sample mean is unbiased)</td>
</tr>
<tr>
<td>2. $SD(\overline{X}) = \sigma/\sqrt{n}$ (so the variance can be reduced by taking larger samples)</td>
</tr>
<tr>
<td>3. The distribution of $\overline{X}$ is approximately normal for large enough sample sizes, no matter what the population distribution is.</td>
</tr>
<tr>
<td>• The approximation gets better and better as $n$ increases, and is usually good enough for practical applications if $n \geq 30$.</td>
</tr>
<tr>
<td>• The distribution of $\overline{X}$ is exactly normal for any sample size if the population distribution is normal.</td>
</tr>
<tr>
<td>• The distribution of $\overline{X}$ is approximately normal even for relatively small sample sizes if the population distribution is close to normal (unimodal, symmetric).</td>
</tr>
</tbody>
</table>

Items 1 and 2 above follow directly from our rules for combining random variables. The important addition of the Central Limit Theorem is that the distribution is approximately normal.

Inference, if we knew $\sigma$

If we knew $\sigma$, the Central Limit Theorem would provide all the information we need for most of inference about a population mean. (We would still not know what to do with small samples from populations that we don’t believe are much like a normal distribution.)

- Hypothesis testing
  - $H_0: \mu = \mu_0$, $H_0: \mu \neq \mu_0$
  - test statistic: $z_0 = \frac{\overline{x} - \mu_0}{\sigma/\sqrt{n}}$
  - p-value: obtained from standard normal distribution using \texttt{pnorm}().

- Confidence intervals
  - $\overline{x} \pm \frac{\sigma}{\sqrt{n}}$

A “clever idea”

A clever idea that doesn’t quite work is to put the sample standard deviation $s = \sqrt{\frac{\sum(x_i - \overline{x})^2}{n-1}}$ into the expression above in place of $\sigma$. Computer simulation of confidence intervals shows that the coverage rate is not what it should be. The problem is that sometimes our estimate of $\sigma$ is too small, making the confidence intervals too narrow. (Sometimes our estimate is too large as well, but this doesn’t happen as often and has a smaller effect on the coverage rate.)

©2006 Randall Pruim (rpruim@calvin.edu)
The Fix

The fix (figured out by Gosset, a.k.a. Student) is to determine the right distribution for

\[
\frac{\bar{x} - \mu}{s/\sqrt{n}}
\]

That distribution is called the \textit{t} distribution with \(n - 1\) degrees of freedom. The \textit{t} distributions are quite similar to the normal distributions (unimodal, symmetric, bell-shaped) but are “shorter and fatter”. As \(n - 1\) increases, the corresponding \textit{t} distribution becomes more and more like the standard normal distribution.

\begin{center}
Summary: The way to handle inference for one sample mean when we don’t know \(\sigma\) (i.e., almost always) is to
\begin{itemize}
  \item substitute \(s\) for \(\sigma\), and
  \item use a \textit{t} distribution instead of the standard normal distribution.
\end{itemize}
This is exactly correct if the population is exactly normal. Fortunately, the \textit{t}-test and \textit{t}-confidence interval are quite \textit{robust}, and work well in the same situations where the Central Limit Theorem applies.
\end{center}

See page 359 for a summary of the one-sample \textit{t}-procedures.
Goals: 1. To examine how population distribution and sample size affect the $t$ confidence intervals.

2. To practice $t$-based confidence intervals and hypothesis tests.

3. To learn how to use R for these procedures.

Exploration 4.4: Exploring Confidence Interval Coverage Rates; pages 350–352

You get to try out the confidence interval applet yourself this time. Begin at part g; we did parts 1–f in class already.

g Did you read the instructions above?

Investigation 4.4.3: Basketball Scoring; pages 352–359

a Load the data using `getData()`. Use whatever name you like for the data. Use `histogram()`, `describe()`, etc. to summarize the data.

f Use `xqqplot()` to get the normal-quantile plot.

h Estimate based on your graph. (You’ll get R to give you the p-value in a moment.)

i Use `pt()` to get the p-value.

k Get the critical value using `qt()`.

l A nice method for tabulating here: Use `describe(nba$points > 200)` to determine how many games in the data set had a combined score over 200. You’ll need to modify this to fit the name of your data set and whatever number you need in place of 200.

The $ in R is used to separate the name of a data set from the name of a variable inside the data set.

Another method: `sum(nba$points>200)`

n–r SKIP THESE for now.

s–t Use `t.test()`. It works much like `binom.test()`, but it takes data rather than summaries as input.

Example: `t.test(nba$points)`. You will need to set some other parameters to get what you need.

For more information, type `?t.test`.

u–v Optional.

What to Turn In

There is no other lab report for this activity. If you finish early, feel free to work on Thursday’s problem set.

©2006 Randall Pruim (rpruim@calvin.edu)
Goals: To learn how to do inference based on (approximate) normal distributions when we want to learn about the difference in the proportions or means of two populations or two treatments. These are the so-called two-sample procedures.

Some Useful R Commands

Comparing Two Means

We’ll demonstrate with the body temperature data from our text. (getData('iscam/temps') -> temps).

• If your data are in a table where one column contains values of the quantitative variable and the other the categorical variable with two levels:

```r
> summary(body.temp~gender,data=temps,fun=favstats)
body.temp N=130
+-------+------+---+----+----+----+----+-----+------+-------+
| | |N |0% |25% |50% |75% |100% |mean |sd |
+-------+------+---+----+----+----+----+-----+------+-------+
|gender |female| 65|96.4|98.0|98.4|98.8|100.8|98.394|0.74349|
| |male | 65|96.3|97.6|98.1|98.6| 99.5|98.105|0.69876|
+-------+------+---+----+----+----+----+-----+------+-------+
|Overall| |130|96.3|97.8|98.3|98.7|100.8|98.249|0.73318|
+-------+------+---+----+----+----+----+-----+------+-------+

> bwplot(body.temp~gender,data=temps) # makes side-by-side box plots

> histogram(~body.temp|gender,data=temps) # makes histogram for each gender

> t.test(body.temp~gender,data=temps)
Welch Two Sample t-test
data: temps$women and temps$men
t = 2.2854, df = 127.51, p-value = 0.02394
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.038813 0.539649
sample estimates:
mean of x  mean of y
 98.394     98.105
```

• If each group’s values are in its own column or vector (a generally less useful format in R), we could get the same t test output using

```r
> t.test(temps$women,temps$men)
```

• The usual optional parameters are available to specify the confidence level, null hypothesis difference, etc. See ?t.test for details.
Comparing Two Proportions

- R does not have a built-in function that does exactly the test described in the text.
- `prop.test` does have a version for working with two samples and should give results that are similar:

```r
> summary( (body.temp>98.6) ~ gender, data=temps, fun=sum )
(body.temp > 98.6)   N=130
+-------+------+---+------------------+
| | |N |(body.temp > 98.6) |
+-------+------+---+------------------+
| gender |female| 65|25 |
| |male | 65|14 |
+-------+------+---+------------------+
|Overall| |130|39 |
+-------+------+---+------------------+
> prop.test(c(25,14),c(65,65)) # c() is used to make a vector

2-sample test for equality of proportions with continuity correction

data: c(25, 14) out of c(65, 65)
X-squared = 3.663, df = 1, p-value = 0.05563
alternative hypothesis: two.sided
95 percent confidence interval:
-0.00099395 0.33945549
sample estimates:
prop 1 prop 2
0.38462 0.21538
```

- We could also do Fisher’s exact test (if the sample sizes are small enough) using `phyper()` or `fisher.test()`. In the latter case, we first need to build a matrix to hold our data values. (cf. Feb 13 & 14)

---

**Summary:** There are nice summaries of these procedures on pages 430–431 and pages 452. Other things to remember:

1. These procedures work for either observational or experimental situations.
2. The mathematics involved is mostly just a recombination of tools we have already developed, especially our rules for combining random variables (cf. April 7).
3. Depending on what we really want to know, these procedures may provide answers to the wrong questions.
Goals:  1. To learn some situations in which a chi-square statistic is used.
   2. To learn how to calculate the chi-square test statistic (including calculation of expected counts).
   3. To learn about the chi-square distribution (including degrees of freedom).
   4. To learn how to get R to do chi-square calculations.

Investigation 6.1.1: Dr. Spock’s Trial; pages 484–491

h We explored various possibilities in class on Friday. Do you remember them? Would they work in this situation? Can you think of any other possibilities?

k Use the function spockSim() which should load with getCalvin('m243').
   This function will produce a data frame containing the counts in the 7 cells of the first row of the simulated table (why aren’t the counts in the second row needed?) as well as the chi-square statistic. Generate and save 500 random simulates using spockSim(500) -> rdata. You can choose some other name for the results in you like.
   Now rdata$chisq contains the 500 chi-square statistics.

m xqplot(rdata$chisq)

n xhistogram(rdata$chisq, fit='gamma') or xhistogram(~chisq, rdata, fit='gamma')
   Note: R will report two parameters: shape is what the book calls \( \alpha \); rate is \( 1/\beta \). You should see that shape is roughly 3 and rate is roughly 1/2.

o Fit a chi-square distribution with 6 degrees of freedom using
   xhistogram(~chisq, rdata, fit='chisq', args=6), or
   xhistogram(~chisq, rdata, fit='gamma', args=list(shape=3,rate=1/2)).

p Calculate the p-value using pchisq(). It works just like pnorm(), etc.

s After completing this, take a moment to get R to do the work for you:
   – spockData contains the data in matrix form.
   – chisq.test(spockData) will give a brief report of the chi-square test.
   – Save this using chisq.test(spockData) -> test.
   – The following commands will give you other interesting information: t$observed, t$expected, t$residuals. Can you figure out what the last of these three is doing?
Investigation 6.1.2: Near-sightedness and Night Lights; pages 491–494

f Use `pchisq()`. Check your work to this point using `chisq.test()`. To do this, you will first need to enter the data into a matrix. Here is one way to do this:

```r
visionData <- rbind(c(40, 39, 12), c(114, 115, 22), c(18, 78, 41))
```

`rbind` stands for row-bind and binds the three vectors together, one above the other, into a 3x3 matrix.

g Use `chisq.test(visionData)$residuals`. What are these numbers?

- These data are also available via

  ```r
  getData('iscam/refraction') -> vision.
  ```

  Each row of this data frame corresponds to one subject. Take a look at the first few rows using `vision[1:5,]`. You will see that there are columns for the variables `refraction` and `light`. The chi-square test can now be performed using

  ```r
  chisq.test(vision$refraction, vision$light).
  ```

  You can get a table of the observed counts using

  ```r
  chisq.test(vision$refraction, vision$light)$observed.
  ```

  Expected counts can be obtained in a similar way.

What to turn in

You don’t need to turn in anything, but I encourage you to work through Practice Problems 6.1.1 and 6.1.2. Ask if you have any difficulties with these problems.
The ANOVA F test

The (1-way) ANOVA test is used to compare the means of several populations or treatment groups. That is, the hypotheses of interest are

- Null Hypothesis: All $k$ groups have the same mean ($\mu_1 = \mu_2 = \cdots = \mu_k$).
- Alternative Hypothesis: Not all the group means are the same.

ANOVA stands for Analysis of Variance, and the basic idea is simple. We will compare the amount of variation within the groups (how different are data values from other values in the same group) to the amount of variation between the groups (how different are the groups from each other).

Ten cardboard boxes of three different types were tested for compression strength (measured in hundreds of pounds). The data are below and can be pulled into R using `getData('iscam/compression') -> comp`. We’ll use this example to demonstrate how the $F$ statistic is calculated.

<table>
<thead>
<tr>
<th>type</th>
<th>strength</th>
<th>grand group mean</th>
<th>mean between groups</th>
<th>(error/residual) within groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$x$</td>
<td>$\overline{x}$</td>
<td>$(\overline{x}_i - \overline{x})$</td>
<td>$(x - \overline{x}_i)^2$</td>
</tr>
<tr>
<td>A</td>
<td>6.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>8.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>7.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>7.80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>7.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>8.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>7.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>6.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>7.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>6.80</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$SST = \quad SSE =$$

$$F = \frac{MST}{MSE} = \frac{SST/(k-1)}{SSE/(n-k)} =$$

p-value =

©2006 Randall Pruim (rpruim@calvin.edu)
You can check your work filling in the table using the command

- `anova.details(strength~type, data=comp)`.

Traditionally, the ANOVA test is presented in tabular form. R will generate this table for you with the command

- `anova(lm(strength~type, data=comp))`.

`lm` stands for linear model. Try it now. You should be able to find $SST$, $SSE$, $MST$, $MSE$, and $F$ in this table to match the values that you calculated.

There is a third sum of squares that is interesting, namely

$$SSTot = \sum (x - \bar{x})^2; \quad MSTot = \frac{SSTot}{n - 1}$$

Notice that $MSTot$ is simply the sample variance. An important fact is that\(^3\)

$$SSTot = SST + SSE$$

so in this sense the ANOVA is really splitting up the variance into two components – one portion “explained” by the groups and the other portion “unexplained” by the groups. The $F$ statistic computes a ratio of these after first dividing by the appropriate degrees of freedom in numerator and denominator.

The resulting statistic $F$ has an F distribution (named after R. A. Fisher of Fisher’s exact test fame) provided (see also page 507):

- Each group has the same population mean (our null hypothesis);
- Each group has the same population variance;
- Each group population is normally distributed; and
- The observations are independent.

In practice, the test is used provided

- The sample standard deviations are similar (largest standard deviation at most double the smallest standard deviation).
- There is no compelling evidence to doubt normality. This is usually checked by looking at the residuals $(x - \bar{x}_i)$.
- The samples are independent random samples or the result of random assignment to treatment groups.

\(^3\)This can be demonstrated by a bit of algebra, but it really comes from an $n$-dimensional version of the Pythagorean Theorem and the fact that a certain pair of vectors is orthogonal.
Computing F from Summary Statistics

A little clever thinking should allow us to calculate $F$ from the information below (from our data on employment scores for applicants with different apparent handicaps).

```r
> hemp <- getData('iscam/handicapemployment')
> summary(score~handicap,data=hemp,fun=favstats)
```

A little clever thinking should allow us to calculate $F$ from the information below (from our data on employment scores for applicants with different apparent handicaps).

```
+--------+----------+--+---+-----+----+-------+-----+-----+------+
|        |          |N | 0% |25%  |50% |75%    |100%| mean |sd    |
+--------+----------+--+---+-----+----+-------+-----+-----+------+
|handicap|amputee   |14|1.9|3.300|4.30|5.725  |7.2 |4.4286|1.5857|
|        |crutches  |14|3.7|4.500|6.10|7.150  |8.5 |5.9214|1.4818|
|        |hearing   |14|1.4|3.025|4.05|5.300  |6.5 |4.0500|1.5326|
|        |none      |14|1.9|3.725|5.00|6.050  |7.8 |4.9000|1.7936|
|        |wheelchair|14|1.7|4.725|5.70|6.350  |7.6 |5.3429|1.7483|
+--------+----------+--+---+-----+----+-------+-----+-----+------+
|Overall  |          |70|1.4|3.700|5.05|6.100  |8.5 |4.9286|1.7188|
+---------------------------------------------------------------+
```

Here is one way to proceed (there are others):

- First calculate $SST_{tot}$ from the overall standard deviation.
- Next calculate $SSE$ from the groupwise standard deviations.
- $SST = SST_{tot} - SSE$.
- From here it is easy, since we know the degrees of freedom.
- Use `pf()` to get the p-value.

Have R generate an ANOVA table to check your work.

Exploring the ANOVA F; pages 509–511

Starting at “Effects of Sample Size” on the bottom of page 511, use the applet to explore the effects of sample sizes, population means, and population variances on the ANOVA $F$ test.
Goals:  
1. To learn how a scatterplot displays the relationship between two quantitative variables.  
2. To learn how the correlation coefficient ($r$) measures the strength of a linear relationship between two quantitative variables.

Investigation 6.3.2: Drive for Show, Putt for Dough; pages 514–520

- `xyplot()` is the R function for making scatterplots. The general form of the command is `xyplot(y~x,data)`, where $y$ and $x$ are the variables to be placed on the vertical and horizontal axes, and `data` is the name of the data frame containing the variables.

- Hint: Determine the units of various parts of the definition and then combine.

- $\sum (x - \bar{x})^2$?

- The R function to compute the correlation coefficient is `cor()`. There are several ways to use it.
  - To find the correlation coefficient between two variables, use `cor(x,y)`. (You will need to use the `$` notation to select variables from inside data frames.)
  - `cor(dataframe,use='pairwise')` will give the correlation coefficients of every pair of variables in the data frame for which this makes sense. (It will display `NA` where things don’t make sense.)

Exploration 6.3.1: Random Correlation Coefficients; pages 520–522

This Exploration uses an applet to help train your eye for correlation coefficient values. If you do not finish it in class, then do it on your own before the next class period.
Least Squares Regression Line (Review)

1. Simple linear regression situation
   (a) Data: two quantitative variables for each unit
      (often use $x$ for explanatory and $y$ for response variable)
   (b) Goal: two find a line that fits the data “best”
      \[
      \hat{y} = a + b(x - \bar{x}) \quad \text{or} \quad \hat{y} = b_0 + b_1 x
      \]

2. What line is best? We’ll pick the line that minimizes the sum of the squares of the residuals.
   - residuals: $e_i = y_i - \hat{y}_i$
   - $SSE = \sum_i e_i^2$
   - minimize $SSE$ by solving $\frac{\partial}{\partial a} SSE = 0$ and $\frac{\partial}{\partial b} SSE = 0$.
     - fortunately the solution is pretty easy
     - $a = \overline{y}$, so the line goes through the point $(\bar{x}, \bar{y})$
     - $b = \frac{S_{xy}}{S_{xx}} = r \frac{s_y}{s_x}$

3. Notation:
   \[
   S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad s_x = S_{xx}/(n-1)
   \]
   \[
   S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \quad s_y = S_{yy}/(n-1)
   \]
   \[
   S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \quad r = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(x_i - \bar{x})}{s_x} \frac{(y_i - \bar{y})}{s_y}
   \]
   \[
   SSTot = S_{yy} \quad SSM = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2
   \]

4. $SSTot = SSM + SSE$  \hspace{1cm} (Not immediately obvious, but can be proven algebraically)
   - $SSTot$ = total variation
   - $SSM$ = variation explained by regression line
   - $SSE$ = variation not explained by regression line
   - $r^2 = \frac{SSM}{SSTot} = 1 - \frac{SSE}{SSTot} =$ proportion of variation explained by linear relationship

5. $F$ test
   - $H_0$: $\beta_1 = 0$ \hspace{1cm} (linear relationship not useful for prediction)
   - test stat: $F = \frac{MSM}{MSE} = \frac{SSM/1}{SSE/(n-2)}$
   - sampling distribution: when $H_0$ is true, then $F$ has the same F distribution as in ANOVA, with $1$ and $n-2$ degrees of freedom.

©2006 Randall Pruim (rpruim@calvin.edu)
Regression Inference

1. Based on regression model (two forms):
\[ y_i = \beta_0 + \beta_1 x_i + \epsilon_i \]
\[ y_i = \alpha + \beta_1 (x_i - \bar{x}) + \epsilon_i \]
where the \( \epsilon_i \)'s are
- independent
- normal
- have mean 0
- have standard deviation \( \sigma \)

Note: the regression line is telling us about the mean response \( y \) for a given explanatory value \( x \).

2. As always, inference is based upon knowing the sampling distributions. The sampling distribution for each of the estimators below is normal with the standard deviation listed. Since we won’t usually know \( \sigma \), we will use the standard error and switch to a \( t \) distribution with \( n - 2 \) degrees of freedom, and confidence intervals will have the form
\[ \text{estimate} \pm t_{\alpha/2} \text{SE} \]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>SD</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma )</td>
<td>( s_e = \sqrt{MSE} = \sqrt{\frac{SSE}{n-2}} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>( b_1 = \frac{S_{xy}}{S_{xx}} )</td>
<td>( \frac{\sigma}{\sqrt{S_{xx}}} )</td>
<td>( \frac{s_e}{\sqrt{S_{xx}}} )</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>( a = \bar{y} )</td>
<td>( \frac{\sigma}{\sqrt{n}} )</td>
<td>( \frac{s_e}{\sqrt{n}} )</td>
</tr>
<tr>
<td>( y )-intercept = ( \beta_0 = \alpha - \beta_1 \bar{x} )</td>
<td>( \bar{y} - b_1(\bar{x}) )</td>
<td>( \sigma \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}} )</td>
<td>( s_e \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}} )</td>
</tr>
<tr>
<td>( \alpha + \beta_1(x - \bar{x}) )</td>
<td>( \hat{y} = a + b_1(x - \bar{x}) )</td>
<td>( \sigma \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}} )</td>
<td>( s_e \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}} )</td>
</tr>
<tr>
<td>( y^* = \alpha + \beta_1(x^* - \bar{x}) + \epsilon^* )</td>
<td>( \hat{y} = a + b_1(x - \bar{x}) )</td>
<td>( \sigma \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}} )</td>
<td>( s_e \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}} )</td>
</tr>
</tbody>
</table>

3. Good news: R will compute these things for us (and it is just like using R for ANOVA)
- `xyplot(y~x,data)` makes a scatterplot
- `xyplot(y~x,data,panel=panel.lm)` adds regression line
- `model <- lm(y~x,data)` builds a model
- `summary(model)` and `anova(model)` display useful tables of information
- `model$residuals`, `model$fitted`, etc.
- `predict(model,newdata=data.frame(x=15))` makes new prediction for \( x = 15 \)
- `predict(model,newdata=data.frame(x=15),interval='conf')` gives a confidence interval
- `predict(model,newdata=data.frame(x=15),interval='pred')` gives a prediction interval
- `xyplot(y~x,data,panel=panel.lmbands)` adds confidence and prediction bands to plot

4. Checking model assumptions/technical conditions:
- **LINE:** Linear relationship, Independent residuals, Normal residuals, Equal variance
- Except for prediction intervals, these procedures are quite robust (similar to \( t \) test)

©2006 Randall Pruim (rpruim@calvin.edu)