Psychology 205: Research Methods

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

Fall, 2013

Outline

R: Statistics for all us

1. What is it?
2. Why use it?
3. Common (mis)perceptions of R
4. Examples for psychologists
    - graphical displays
    - basic statistics
    - advanced statistics
    - Although programming is easy in R, that is beyond the scope of today

R: What is it?

1. R: An international collaboration
2. R: The open source - public domain version of S+
3. R: Written by statistician (and all of us) for statisticians (and the rest of us)
4. R: Not just a statistics system, also an extensible language.
   - This means that as new statistics are developed they tend to appear in R far sooner than elsewhere.
   - For example, the most recent issue of *Psychological Methods* had at least three articles with examples or supplementary work done in R
   - R facilitates asking questions that have not already been asked.

Statistical Programs for Psychologists

- General purpose programs
  - R
  - S+
  - SAS
  - SPSS
  - STATA
  - Systat
- Specialized programs
  - Mx
  - EQS
  - AMOS
  - LISREL
  - MPlus
  - Your favorite program

Statistical Programs for Psychologists

- General purpose programs
  - R
  - $+
  - A
  - $P$$
  - $TATA
  - $y$tat
- Specialized programs
  - Mx (OpenMx is part of R)
  - EQ$
  - AMO$
  - LI$REL
  - MPlu$
  - Your favorite program

R: A way of thinking

- "R is the lingua franca of statistical research. Work in all other languages should be discouraged."
- "This is R. There is no if. Only how."
- "Overall, SAS is about 11 years behind R and S-Plus in statistical capabilities (last year it was about 10 years behind) in my estimation."

Taken from the R.-fortunes (selections from the R.-help list serve)

R is open source, how can you trust it?

- Q: "When you use it [R], since it is written by so many authors, how do you know that the results are trustable?"
- A: "The R engine [...] is pretty well uniformly excellent code but you have to take my word for that. Actually, you don't. The whole engine is open source so, if you wish, you can check every line of it. If people were out to push dodgy software, this is not the way they'd go about it."

What is R?: Technically

- R is an open source implementation of S (S-Plus is a commercial implementation)
- R is available under GNU Copy-left
- The current version of R is 3.0.2
- The development version of R 3.1.0 is available to test and will be released next spring
- R is a group project run by a core group of developers (with new releases semiannually)

(Adapted from Robert Gentleman)

R: A brief history

- 1991-93: Ross Dhaka and Robert Gentleman begin work on R project at U. Auckland
- 1995: R available by ftp under the GPL
- 96-97: mailing list and R core group is formed
- 2000: John Chambers, designer of S joins the Rcore (wins a prize for best software from ACM for S)
- 2001-2011: Core team continues to improve base package with a new release every 6 months.
- Many others contribute "packages" to supplement the functionality for particular problems
    - 2003-04-01: 250 packages
    - 2004-10-01: 500 packages
    - 2007-04-12: 1,000 packages
    - 2009-10-04: 2,000 packages
    - 2011-05-12 3,000 packages
    - 2012-01-10 4,000 packages

Misconception: R is hard to use

1. R doesn't have a GUI (Graphical User Interface)
   - Partly true, many use syntax
   - Partly not true, GUIs exist (e.g., R Commander, R-Studio)
   - Quasi GUIs for Mac and PCs make syntax writing easier
2. R syntax is hard to use
   - Not really, unless you think an iPhone is hard to use
   - Easier to give instructions of 1-4 lines of syntax rather than pictures of what menu to pull down.
   - Keep a copy of your syntax, modify it for the next analysis.
3. R is not user friendly: A personological description of R
   - R is introverted: it will tell you what you want to know if you ask, but not if you don't ask.
   - R is conscientious: it wants commands to be correct.
   - R is not agreeable: its error messages are at best cryptic.
   - R is stable: it does not break down under stress.
   - R is open: new ideas about statistics are easily developed.

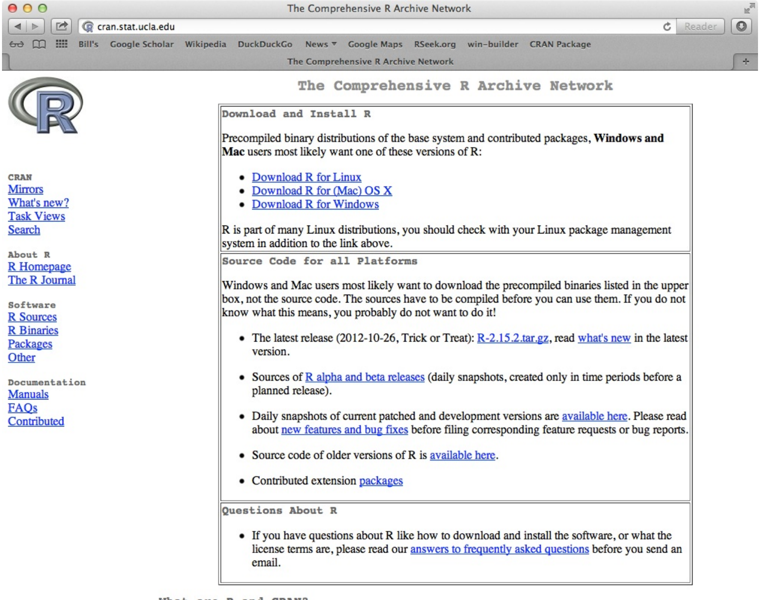
Misconceptions: R is hard to learn

1. With a brief web based tutorial at http://personality-project.org/r, 2nd and 3rd year undergraduates in psychological methods and personality research courses are using R for descriptive and inferential statistics and producing publication quality graphics.
2. More and more psychology departments are using it for graduate and undergraduate instruction.
3. R is easy to learn, hard to master
   - R-help newsgroup is very supportive
   - There are multiple web based and pdf tutorials see (e.g., http://www.r-project.org/)
   - Short courses using R for many applications
4. Books and websites are available for for SPSS and SAS users trying to learn R (e.g.,http://oit.utk.edu/scc/RforSAS&SPSSusers.pdf by Bob Muenchen).

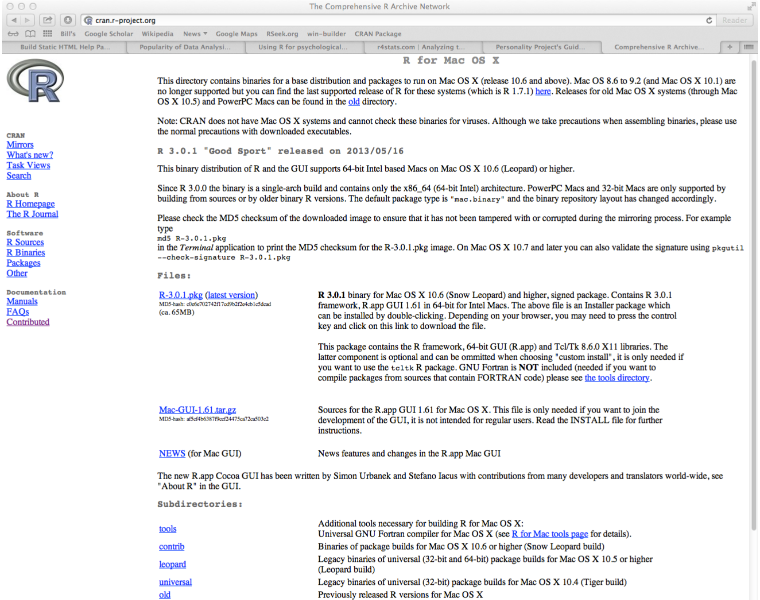
Ok, how do I get it? Getting started with R

- Download from R Cran (http://cran.r-project.org/)
    - Choose appropriate operating system and download compiled R
- Install R (current version is 3.0.2)
- Start R
- Add useful packages (you just need to do this once)
    - install.packages("ctv") #this downloads the task view package
    - library(ctv) #this activates the ctv package
    - install.views("Psychometrics") #among others
    - Take a 5 minute break
- Activate the package(s) you want to use today (e.g., *psych*)
    - library(psych) #necessary for most of today's examples
- Use R

Go to the R.project.org



Go to the Comprehensive R Archive Network (CRAN)



R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 3.0.1 "Good Sport" released on 2013/05/16

This binary distribution of R and the GUI supports 64-bit Intel based Macs on Mac OS X 10.6 (Leopard) or higher.

Since R 3.0.0 the binary is a single-arch build and contains only the x86_64 (64-bit Intel) architecture. PowerPC Macs and 32-bit Macs are only supported by building from sources or by older binary R versions. The default package type is `"mac.binary"` and the binary repository layout has changed accordingly.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type
`md5 R-3.0.1.pkg`
in the *Terminal* application to print the MD5 checksum for the R-3.0.1.pkg image. On Mac OS X 10.7 and later you can also validate the signature using `pkgutil --check-signature R-3.0.1.pkg`

Files:

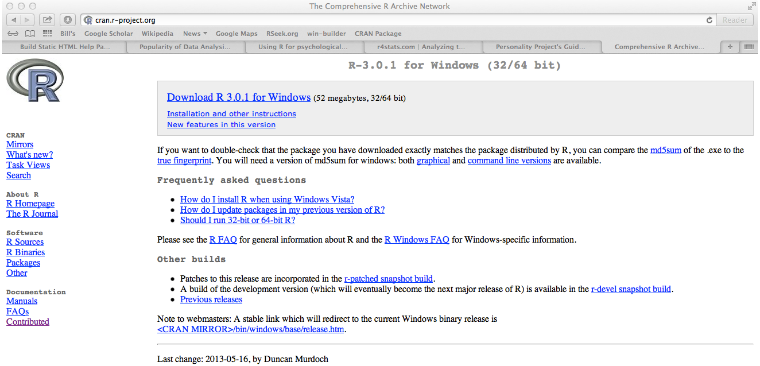
R-3.0.1.pkg (latest version) MD5-hash: c0e6e702742f17cd9b2f2e4cb1c5dcad (ca. 65MB)	**R 3.0.1** binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.0.1 framework, R.app GUI 1.61 in 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file. This package contains the R framework, 64-bit GUI (R.app) and Tcl/Tk 8.6.0 X11 libraries. The latter component is optional and can be ommitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package. GNU Fortran is **NOT** included (needed if you want to compile packages from sources that contain FORTRAN code) please see the tools directory.
Mac-GUI-1.61.tar.gz MD5-hash: af5cf4b6387f9ccf24475ca72ca503c2	Sources for the R.app GUI 1.61 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.
NEWS (for Mac GUI)	News features and changes in the R.app Mac GUI

The new R.app Cocoa GUI has been written by Simon Urbanek and Stefano Iacus with contributions from many developers and translators world-wide, see "About R" in the GUI.

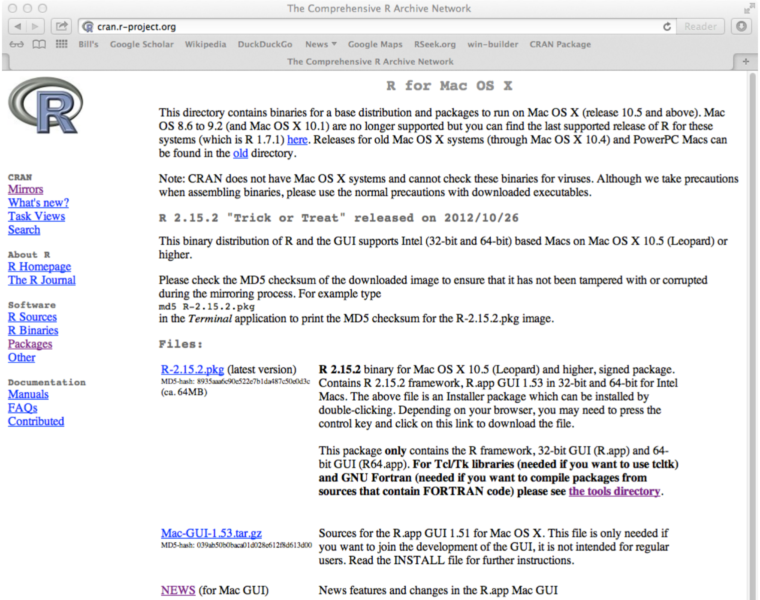
Subdirectories:

tools	Additional tools necessary for building R for Mac OS X: Universal GNU Fortran compiler for Mac OS X (see R for Mac tools page for details).
contrib	Binaries of package builds for Mac OS X 10.6 or higher (Snow Leopard build)
leopard	Legacy binaries of universal (32-bit and 64-bit) package builds for Mac OS X 10.5 or higher (Leopard build)
universal	Legacy binaries of universal (32-bit) package builds for Mac OS X 10.4 (Tiger build)
old	Previously released R versions for Mac OS X

Download and install the appropriate version – PC



Download and install the appropriate version – Mac


The Comprehensive R Archive Network
cran.r-project.org
Bill's Google Scholar Wikipedia DuckDuckGo News Google Maps RSeek.org win-builder CRAN Package

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.5 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.4) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 2.15.2 "Trick or Treat" released on 2012/10/26

This binary distribution of R and the GUI supports Intel (32-bit and 64-bit) based Macs on Mac OS X 10.5 (Leopard) or higher.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type
`md5 R-2.15.2.pkg`
in the *Terminal* application to print the MD5 checksum for the R-2.15.2.pkg image.

Files:

R-2.15.2.pkg (latest version) MD5-hash: 8935aaa6c90e522e7b1da487c50e0d3c (ca. 64MB)	**R 2.15.2** binary for Mac OS X 10.5 (Leopard) and higher, signed package. Contains R 2.15.2 framework, R.app GUI 1.53 in 32-bit and 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file. This package **only** contains the R framework, 32-bit GUI (R.app) and 64-bit GUI (R64.app). **For Tcl/Tk libraries (needed if you want to use tcltk) and GNU Fortran (needed if you want to compile packages from sources that contain FORTRAN code) please see the tools directory.**
Mac-GUI-1.53.tar.gz MD5-hash: 039ab50b0baca01d028e612f8d613d00	Sources for the R.app GUI 1.51 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.
NEWS (for Mac GUI)	News features and changes in the R.app Mac GUI

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Starting R on a PC

RGui

File Edit View Misc Packages Windows Help

R Console

```
R version 2.13.0 (2011-04-13)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Start up R and get ready to play (Mac version)

```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.62 (6558) x86_64-apple-darwin10.8.0]
>    # >  is the prompt for all commands    #is for comments
```

Annotated installation guide: don't type the >

```
> install.packages("ctv")
```

- Install the task view installer package. You might have to choose a "mirror" site.

```
> library(ctv)
```

- Make it active

```
> install.views("Psychometrics")
```

- Install all the packages in the "Psychometrics" task view. This will take a few minutes.

```
#or just install a few packages
> install.packages("psych")
```

- Or, just install one package (e.g., psych)

```
> install.packages("GPArotation")
> install.packages("MASS")
> install.packages("mvtnorm")
> install.packages("lavaan")
```

- as well as a few suggested packages that add functionality for factor rotation, multivariate normal distributions, etc.

Installing just the psych package

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
R version 2.13.0 (2011-04-13)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("psych")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.stat.ucla.edu/bin/windows/contrib/2.13/psych_1.0-97.zip'
Content type 'application/zip' length 1952216 bytes (1.9 Mb)
opened URL
downloaded 1.9 Mb
```

Or, install and use ctv package to load a task view on a PC

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("ctv")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.stat.ucla.edu/bin/windows/contrib/2.13/ctv_0.7-2.zip'
Content type 'application/zip' length 298753 bytes (291 Kb)
opened URL
downloaded 291 Kb

package 'ctv' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\users\revelle\Temp\RtmpwNzUtt\downloaded_packages
> library(ctv)
>
```

Use the package menu to select a mirror

Check the version number for R (should be $\geq$ 3.02) and for psych ($\geq$1.3.2)

```
> library(psych)
> sessionInfo()

R version 3.0.2 (2013-09-25)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] psych_1.3.10
```

Statistical Review

Problem set 1 asked for a variety of analyses.

Here I show the direct answers, but also do the analyses in a variety of ways.

I use the statistical program R.

For help on R, go to the short tutorial on using R for research methods http://personality-project.org/r/r.205.tutorial.html.

In the following, I assume that you have downloaded R and installed the *psych* package.

These notes are also available as a pdf from the syllabus.

Comparing two groups

An investigator believes that caffeine facilitates performance on a simple spelling test. Two groups of subjects are given either 200 mg of caffeine or a placebo. Although there are several ways of testing if these two groups differ, the most conventional would be a t-test. Apply a t-test to the data in Table 1:

Table : The effect of caffeine on spelling performance

placebo	caffeine
24	24
25	29
27	26
26	23
26	25
22	28
21	27
22	24
23	27
25	28
25	27
25	26

The t-test

Many statistical tests may be thought of as comparing a statistic to the error of the statistic. One of the most used tests, the t-test (developed by Gossett but published under the name of Student), compares the difference between two means to the expected error of the difference between to means. As we know, the standard error (se) of a single group with mean, $\bar{X}$ with standard deviation, s, and variance, s^2

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2/}{n-1} \tag{1}$$

is just

$$s.e. = \sqrt{\frac{s^2}{n}} = \frac{s}{\sqrt{n}} \tag{2}$$

and the standard error of the difference of two, uncorrelated groups is

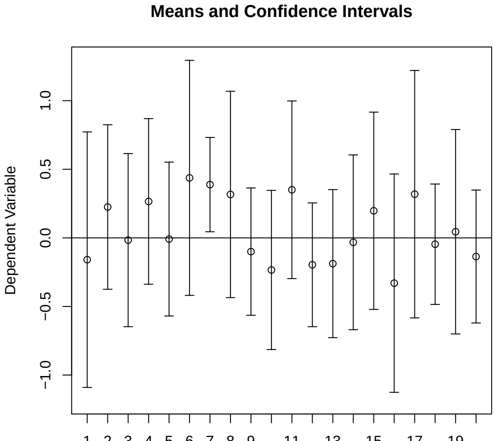
$$se_{x_1 - x_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \tag{3}$$

Gosset (aka student) and the t-test

How best can we understand the notion of a standard error? One way is to draw repeated samples from a known population and examine their variability. Although this was the procedure used by Gossett, it is also possible to simulate this using random samples drawn by computer from a known or unknown distribution. Using R it is easy to simulate distributions, either the normal or resampled from our data. Consider 20 samples from a normal distribution of size 12 (Figure 28). For each sample we show the mean and the confidence interval of the mean. Note how some of the means are very far apart. That is, even though the mean for the population is known to be zero, the means of samples vary around that. The vertical lines in the graph represent 1.96 * the standard error of the mean. Note how the confidence region around almost all sample means includes the population mean. But note how some do not.

Confidence intervals of 20 samples

```
> x <- matrix(rnorm(240),ncol=20)
> error.bars(x, xlab="sample", main="Means and Confidence I
> abline(h=0)
```

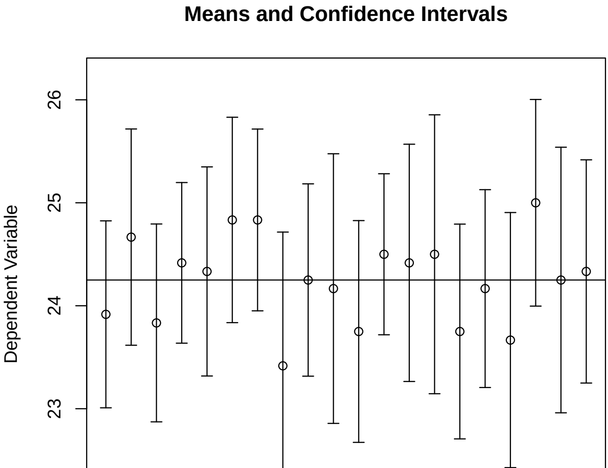


Resampling

An alternative to sampling from the normal population is to resample from the actual data that we collect. Figure 1 shows the mean and confidence regions for 20 samples of size 12, where each sample was drawn with replacement from the original data. Once again, note how much variability there is from sample to sample, even though they come from the same population.

Resampling

```
> x <- matrix(sample(spelling[,1],240,replace=TRUE),ncol=20)
> error.bars(x, xlab="sample", main="Means and Confidence Intervals")
> abline(h=24.25)
```



Differences between two samples

Just as we can find the standard deviation of the data and standard error of the mean of a sample, so we can find the standard deviation and associated standard error of the mean for differences between two samples. The standard error of the difference of two, uncorrelated groups is two, uncorrelated groups is

$$se_{x_1-x_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \tag{4}$$

Given that samples from the same population differ a great deal, how much do the spelling scores of the placebo and caffeine groups differ? Do they differ more than would be expected by chance if in the population there was no effect of caffeine?

The t-test

The t-test compares the differences between the means to the standard error of the differences between sample means.
That is,

$$t = \frac{\bar{X}_1 - \bar{X}_2}{se_{x_1 - x_2}} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{5}$$

This looks somewhat complicated, but because it is such a common operation, the t-test is a basic function in R (as well as all major statistics programs).

A minor inconvenience–stacking the data

From the point of view of most statistical programs, the data need to be rearranged to show the Independent Variable (IV) and the Dependent Variable (DV). Then we try to find how much the DV varies as a function of the IV.
In R, this is done by first loading in the *psych* package, then reading the clipboard using the `read.clipboard` and then using the `stack` function to convert from the way the data look in Table 1 to the way the data look in Table 2.

```
>library(psych)       #this loads the psych package into
                       your active workspace
>spelling <- read.clipboard()    #copy into your clipboard
                        and then read the clipboard into R
```

The t-test:

```
spelling  #show the data
with(spelling, t.test(Placebo,Drug))
```

```
> spelling  #show the data
   Placebo Drug
1       24   24
2       25   29
3       27   26
...
10      25   28
11      25   27
12      25   26
> with(spelling, t.test(Placebo,Drug))
Welch Two Sample t-test
data:  Placebo and Drug
t = -2.5273, df = 21.999, p-value = 0.01918
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.4894368 -0.3438965
sample estimates:
mean of x mean of y
 24.25000  26.16667
```

Describe the data

describe(spelling)

```
        var  n  mean   sd median trimmed  mad min max range  skew kurtosis   se
Placebo   1 12 24.25 1.86   25.0    24.3 1.48  21  27     6 -0.33    -1.33 0.54
Drug      2 12 26.17 1.85   26.5    26.2 2.22  23  29     6 -0.22    -1.33 0.53
```

Another way of doing it: Stacking the data

Table : The data have been read and stacked

```
> prob1 <- stack(spelling)   #convert the data into an array with DV and IV
> prob1

   values     ind
1      24 Placebo
2      25 Placebo
3      27 Placebo
4      26 Placebo
5      26 Placebo
6      22 Placebo
7      21 Placebo
8      22 Placebo
9      23 Placebo
10     25 Placebo
11     25 Placebo
12     25 Placebo
13     24    Drug
14     29    Drug
15     26    Drug
16     23    Drug
17     25    Drug
18     28    Drug
19     27    Drug
20     24    Drug
21     27    Drug
22     28    Drug
23     27    Drug
24     26    Drug
```

Describe the data using describe.by

It is always useful to describe the data, both numerically and graphically. Numerically we can do this using the `describe.by` function.

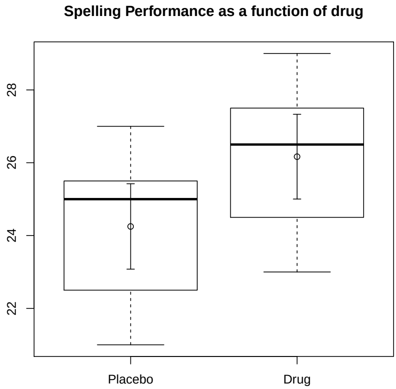
```
> describe.by(prob1$values,prob1$ind)

group: Drug
  var  n  mean   sd median trimmed  mad min max range  skew kurtosis   se
1   1 12 26.17 1.85   26.5    26.2 2.22  23  29     6 -0.22    -1.33 0.53
------------------------------------------------------------------------------
group: Placebo
  var  n  mean   sd median trimmed  mad min max range  skew kurtosis   se
1   1 12 24.25 1.86     25    24.3 1.48  21  27     6 -0.33    -1.33 0.54
```

Graphically: show a box plot

Graphically, we can do a boxplot and then add the standard errors to the data (Figure 2).

```
> boxplot(spelling,main="Spelling Performance as a functio
> error.bars(spelling,add=TRUE)
```



The t-test using R

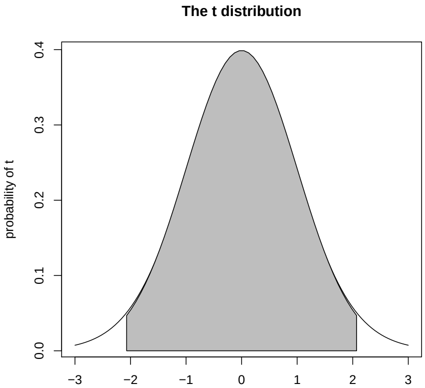
Now, we can do the t-test using the `t.test` function. The distribution of t depends upon the degrees of freedom. Figure 3 shows the .05 rejection region (.025 on the left tail, .025 on the right tail.))

```
> t.test(values~ind,data=prob1)

Welch Two Sample t-test
data:  values by ind
t = 2.5273, df = 21.999, p-value = 0.01918
alternative hypothesis: true difference in means is not equ
95 percent confidence interval:
 0.3438965 3.4894368
sample estimates:
   mean in group Drug mean in group Placebo
             26.16667              24.25000
```

The t-test as a distribution

```
> curve(dt(x,24),-3,3,xlab="t",ylab="probability of t",mai
> xvals <- seq(-2.07,2.07,length=50)
> dvals <- dnorm(xvals)
> polygon(c(xvals,rev(xvals)),c(rep(0,50),rev(dvals)),col="
```



Analysis of Variance (ANOVA)

The t-test compares the difference between two means with respect to the standard error of the differences. Another test, developed by Ronald Fisher, is the Analysis of Variance (ANOVA). Here we are comparing an estimate of the population variance derived from the variance of the means to an estimate of the population variance derived from the variability within each group. For two groups, the variance estimate has 1 degree of freedom. We use the `aov` function and then ask for the `summary` of the results. Compare the results of this analysis with the previous one. The F statistic for a 1 degree of freedom comparison (one between two groups) is the same as t^2. The probability of observing an F of this size or bigger is the same as observing the t of that size or larger (in absolute value).

ANOVA

```
> summary(aov(values~ind,data=prob1))
            Df Sum Sq Mean Sq F value  Pr(>F)
ind          1 22.042 22.0417  6.3875 0.01918 *
Residuals   22 75.917  3.4508
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.0
```

Regression as a generalized ANOVA

Yet another way of thinking about this problem is to use linear regression. That is, if we estimate β in the linear regression equation:

$$\hat{y} = \beta x + e \tag{6}$$

and we use the `lm` (for linear model) function

frame

```
> summary(lm(values~ind,data=prob1))
Call:
lm(formula = values ~ ind, data = prob1)

Residuals:
   Min     1Q Median     3Q    Max
-3.250 -1.479  0.750  1.062  2.833

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.1667     0.5362  48.796   <2e-16 ***
indPlacebo   -1.9167     0.7584  -2.527   0.0192 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.0

Residual standard error: 1.858 on 22 degrees of freedom
```

t-test, ANOVA and Linear Regression

We find that the difference between the two IV conditions is 1.917 (this is the same as the difference between the means found in the t-test) and that the probability of this difference happening by chance if there were no difference is .0192. This is, of course, the same probability as that found by the t-test or the ANOVA.

Correlation and regression

Another investigator believes that introversion/extraversion has a linear relationship to spelling ability and reports the following data (Table 3). This can be solved by finding the linear regression of Spelling on Introversion or by finding the correlation between spelling and introversion. Do either one (or both).

Table : Does introversion predict spelling ability?

Introversion	Spelling
21	31
14	33
13	39
13	24
20	35
21	37
11	36
15	20
23	46
12	31
17	44
26	44

A correlation problem

For this problem, we need to read in the data from the clipboard using the `read.clipboard` function and then can use the `cor` function to the find the correlation, or the `lm` function to find the linear regression, or use the `pairs.panels` function to find the correlation as well as to graph the data.

```
>int_spelling <- read.clipboard()
```

```
> round(cor(int_spelling),2)
              Introversion Spelling
Introversion          1.00     0.51
Spelling              0.51     1.00
> cor.test(int_spelling$Introversion,int_spelling$Spelling)
Pearson's product-moment correlation

data:  int_spelling$Introversion and int_spelling$Spelling
t = 1.8761, df = 10, p-value = 0.0901
```

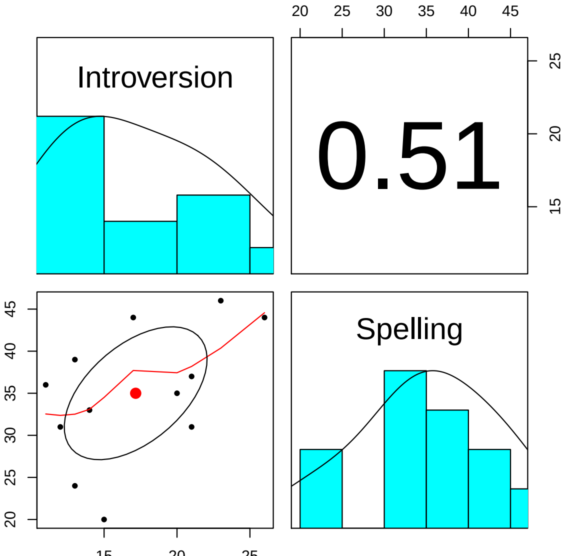
Another way to do it

Or, use the `corr.test` function in *psych*

```
corr.test(ie)
```

```
Call:corr.test(x = ie)
Correlation matrix
             Introversion Spelling
Introversion         1.00     0.51
Spelling             0.51     1.00
Sample Size
             Introversion Spelling
Introversion           12       12
Spelling               12       12
Probability values (Entries above the diagonal are adjusted for multiple tests.
             Introversion Spelling
Introversion         0.00     0.09
Spelling             0.09     0.00
```

Better yet, graph it



A two way ANOVA

Still another investigator believes that spelling performance is a function of the interaction of caffeine and time of day. She administors 0 or 200 mg of caffeine to subjects at 9 am and 9 pm. These data are typically examined using an Analysis of Variance (ANOVA), although a multiple regression using the general linear model would work as well. If the results are as below (Table 4), do the ANOVA.

Table : Time of day, caffeine, and spelling performance

9am 0 mg	9 am 200 mg	9pm 0 mg	9pm 200 mg
26	27	28	24
27	30	27	23
25	28	25	25
22	32	25	21
27	25	31	23
23	29	32	21
21	31	25	25
28	28	32	21
21	28	26	26
23	26	25	22
20	29	27	23
23	31	26	26

2 way ANOVA – reading the data

We first read in the data (but without the labels for the columns) and then add colnames to the data

```
>tod.data<- read.clipboard(header=FALSE)
```

Unfortunately, this analysis is a bit more complicated, because we need to string the data out and then add the conditions as additional variables. This will be discussed in more detail in subsequent handouts.

```
> colnames(tod.data) <- c("AP","AC","PP","PC")
> tod.stacked <- stack(tod.data)
> tod.df <- data.frame(spelling = tod.stacked$values,
       drug = rep(c(rep("P",12),rep("C",12)),2),
        time=c(rep("AM",24),rep("PM",24)))
> anova(lm(spelling~drug*time,data=tod.df))
```

2 way ANOVA – the results

```
Analysis of Variance Table

Response: spelling
          Df  Sum Sq Mean Sq F value    Pr(>F)
drug       1   1.688   1.688  0.2971    0.5885
time       1   9.187   9.187  1.6175    0.2101
drug:time  1 238.521 238.521 41.9937 6.633e-08 ***
Residuals 44 249.917   5.680
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.0
```