Psychology 405: Psychometric Theory
Getting Started with R

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

April, 2019
Outline

What is R?
   Where did it come from, why use it?

Install R
   Installing R on your computer and adding packages

R is a calculator
   Basic R capabilities: Calculation, Statistical tables, Graphics

R for graphics

R for statistics
   4 steps: read, explore, test, graph
   Basic descriptive and inferential statistics

R for psychometrics
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.
   • R facilitates asking questions that have not already been asked.
5. R: encourages publications of "Reproducible Research"
   • integrate data, code, text into one document
   • Sweave and knitr
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.” (Jan de Leeuw, 2003)
• “This is R. There is no if. Only how.” (Simon ‘Yoda’ Blomberg, 2005)
• “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.” (Frank Harrell, 2003)
• ”I quit using SAS in 1991 because my productivity jumped at least 20% within one month of using S-Plus.” (Frank Harrell, 2003)

Taken from the R.-fortunes (selections from the R.-help list serve)
“You must realize that R is written by experts in statistics and statistical computing who, despite popular opinion, do not believe that everything in SAS and SPSS is worth copying. Some things done in such packages, which trace their roots back to the days of punched cards and magnetic tape when fitting a single linear model may take several days because your first 5 attempts failed due to syntax errors in the JCL or the SAS code, still reflect the approach of “give me every possible statistic that could be calculated from this model, whether or not it makes sense”. The approach taken in R is different. The underlying assumption is that the useR is thinking about the analysis while doing it. ” (Douglas Bates, 2007)
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.” (Bill Venables, 2004)

• “It’s interesting that SAS Institute feels that non-peer-reviewed software with hidden implementations of analytic methods that cannot be reproduced by others should be trusted when building aircraft engines.” – Frank Harrell (in response to the statement of the SAS director of technology product marketing: ”We have customers who build engines for aircraft. I am happy they are not using freeware when I get on a jet.”) R-help (January 2009)
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 3.2.4 (3.3.0 coming soon)
- 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 for Macs at U. Auckland (S for Macs).
- 1995: R available by ftp under the General Public License.
- 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)
- 2001-2016: Core team continues to improve base package with a new release every 6 months (now more like yearly).
- 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
  - 2012-08-27: 4,000 packages
  - 2014-05-16: 5,547 packages (on CRAN) + 824 bioinformatic packages on BioConductor
  - 2015-05-20 6,678 packages (on CRAN) + 1,024 bioinformatic packages + ?,000s on GitHub/R-Forge
  - 2016-03-21 8,120 packages (on CRAN) + 1,104 bioinformatic packages + ?,000s on GitHub/R-Forge
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
   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
   • 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 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 3.5.2 ) with 3.6.0 coming this soon
- Start R
- Add useful packages (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
Install R

> install.packages("ctv")

> library(ctv)

> install.views("Psychometrics")

#or just install a few packages

> install.packages("psych")

> install.packages("GPArotation")
> install.packages("MASS")
> install.packages("mnormt")

- Install the task view installer package. You might have to choose a “mirror” site.
- Make it active
- Install all the packages in the “Psychometrics” task view. This will take a few minutes.
- Or, just install one package (e.g., psych) as well as a few suggested packages that add functionality for factor rotation, multivariate normal distributions, etc.
Check the version number for R (should be $\geq 3.2.4$) and for psych ($\geq 1.5.8$)

```r
> library(psych)
> sessionInfo()
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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 1.70 (7612) x86_64-apple-darwin15.6.0]
[Workspace restored from /Users/wr/.RData]
R is extensible: The use of “packages”

- More than 8,120 packages are available for R (and growing daily)
- Can search all packages that do a particular operation by using the sos package
  - `install.packages("sos")` #if you haven’t already
  - `library(sos)` # make it active once you have it
    - `findFn("X")` #will search a web data base for all packages/functions that have "X"
    - `findFn("principal components analysis ")` #will return 2,554 matches and reports the top 400 and download 385 links to 174 packages
    - `findFn("Item Response Theory")` # will return 503 matches with 326 links in 76 packages
    - `findFn("INDSCAL ")` # will return 20 matches in 5 packages.
  - `install.packages("X")` will install a particular package (add it to your R library – you need to do this just once)
  - `library(X)` #will make the package X available to use if it has been installed (and thus in your library)
A small subset of very useful packages

**General use**
- core R
- MASS
- lattice
- lme4 (core)
- psych
- Zelig

**Special use**
- ltm
- sem
- lavaan
- OpenMx
- GPArotation
- mvtnorm
- > 8,120 known
- + ?

**General applications**
- most descriptive and inferential stats
- Modern Applied Statistics with S
- Lattice or Trellis graphics
- Linear mixed-effects models
- Personality and psychometrics
- General purpose toolkit

**More specialized packages**
- Latent Trait Model (IRT)
- SEM and CFA (one group)
- SEM and CFA (multiple groups)
- SEM and CFA (multiple groups +)
- Jennrich rotations
- Multivariate distributions
- Thousands of more packages on CRAN
- Code on webpages/journal articles
Installing packages

1. Just need to install a package once.
2. Typically do this from “Packages and Data ” menu using the install packages option.
   - This defaults to CRAN binaries
   - Can be adjusted to CRAN sources (if working on bleeding edge develop versions of R)
   - Can be specified as “another repository”
3. Can also do this by command
   ```r
   install.packages("psych",repos="http://personality-project.org/r/",type="source")
   ```
   trying URL 'http://personality-project.org/r/src/contrib/psych_1.9.4.tar.gz'
   Content type 'application/x-gzip' length 2216674 bytes (2.1 Mb)
   opened URL
   ==============================================================
downloaded 2.1 Mb
* installing *source* package "psych"..
** R
** data
*** moving datasets to lazyload DB
** inst
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded
* DONE (psych)
The downloaded source packages are in
/private/var/folders/sp/lzq1bshd0014mf1yrrmt9h_m0000gn/T/Rtmp36Wbhy/downloaded_packages
R is just a fancy calculator. Add, subtract, sum, products, group

> 2 + 2
[1] 4

> 3^4
[1] 81

> sum(1:10)
[1] 55

> prod(c(1, 2, 3, 5, 7))
[1] 210

It is also a statistics table (the normal distribution, the t distribution)

> pnorm(q = 1)
[1] 0.8413447

> pt(q = 2, df = 20)
[1] 0.9703672
What is R?

Install R

R is a calculator

R for graphics

R for statistics

R for psychometrics

R is a set of distributions. Don’t buy a stats book with tables!

Table: To obtain the density, prefix with \textit{d}, probability with \textit{p}, quantiles with \textit{q} and to generate random values with \textit{r}. (e.g., the normal distribution may be chosen by using \texttt{dnorm}, \texttt{pnorm}, \texttt{qnorm}, or \texttt{rnorm}.)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>base name</th>
<th>P 1</th>
<th>P 2</th>
<th>P 3</th>
<th>example application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>norm</td>
<td>mean</td>
<td>sigma</td>
<td></td>
<td>Most data</td>
</tr>
<tr>
<td>Multivariate normal</td>
<td>mvnorm</td>
<td>mean</td>
<td>r</td>
<td>sigma</td>
<td>Most data</td>
</tr>
<tr>
<td>Log Normal</td>
<td>lnorm</td>
<td>log mean</td>
<td>log sigma</td>
<td></td>
<td>income or reaction time</td>
</tr>
<tr>
<td>Uniform</td>
<td>unif</td>
<td>min</td>
<td>max</td>
<td></td>
<td>rectangular distributions</td>
</tr>
<tr>
<td>Binomial</td>
<td>binom</td>
<td>size</td>
<td>prob</td>
<td></td>
<td>Bernuilli trials (e.g. coin flips)</td>
</tr>
<tr>
<td>Student’s t</td>
<td>t</td>
<td>df</td>
<td></td>
<td>nc</td>
<td>Finding significance of a t-test</td>
</tr>
<tr>
<td>Multivariate t</td>
<td>mvt</td>
<td>df</td>
<td>corr</td>
<td>nc</td>
<td>Multivariate applications</td>
</tr>
<tr>
<td>Fisher’s F</td>
<td>f</td>
<td>df1</td>
<td>df2</td>
<td>nc</td>
<td>Testing for significance of F test</td>
</tr>
<tr>
<td>(\chi^2)</td>
<td>chisq</td>
<td>df</td>
<td></td>
<td>nc</td>
<td>Testing for significance of (\chi^2)</td>
</tr>
<tr>
<td>Exponential</td>
<td>exp</td>
<td>rate</td>
<td></td>
<td></td>
<td>Exponential decay</td>
</tr>
<tr>
<td>Gamma</td>
<td>gamma</td>
<td>shape</td>
<td>rate</td>
<td>scale</td>
<td>distribution theory</td>
</tr>
<tr>
<td>Hypergeometric</td>
<td>hyper</td>
<td>m</td>
<td>n</td>
<td>k</td>
<td>Item Response Theory</td>
</tr>
<tr>
<td>Logistic</td>
<td>logis</td>
<td>location</td>
<td>scale</td>
<td></td>
<td>Count data</td>
</tr>
<tr>
<td>Poisson</td>
<td>pois</td>
<td>lambda</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weibull</td>
<td>weibull</td>
<td>shape</td>
<td>scale</td>
<td></td>
<td>Reaction time distributions</td>
</tr>
</tbody>
</table>
R can draw distributions

```r
curve(dnormal(x), -3, 3,
ylab="probability of x",
main="A normal curve")
```
R can draw more interesting distributions

Log normal

Chi Square distribution

Normal and t with 4 df

The normal curve
A simple scatter plot using `plot`

```
plot(iris[1:2], xlab="Sepal.Length", ylab="Sepal.Width", main="Fisher Iris data")
```
A scatter plot matrix plot with loess regressions using `pairs.panels`

1. Correlations above the diagonal
2. Diagonal shows histograms and densities
3. Scatter plots below the diagonal with correlation ellipse
4. Locally smoothed (loess) regressions for each pair
5. Optional color coding of grouping variables.

```r
pairs.panels(iris[1:4], bg=c("red", "yellow", "blue"), [iris$Species], pch=21, main="Fisher Iris data by Species")
```
The simple factor structure

Factor Analysis

- Sentences
- Vocabulary
- Sent.Completion
- First.Letters
- 4.Letter.Words
- Suffixes
- Letter.Series
- Letter.Group
- Pedigrees

Factor Analysis
Sentences
Vocabulary
Sent.Completion
First.Letters
4.Letter.Words
Suffixes
Letter.Series
Letter.Group
Pedigrees
MR1
0.9
0.9
0.8
MR2
0.9
0.7
0.6
MR3
0.8
0.6
0.5
0.6
0.5
0.5

Two ways of viewing the higher order structure

Omega

Hierarchical (multilevel) Structure

![Diagram showing two ways of viewing the higher order structure with nodes and edges labeled with values.](image-url)
A hierarchical cluster structure found by iclust

iclust(Thurstone)
Using R for psychological statistics: Basic statistics

1. Writing syntax
   - For a single line, just type it
   - Mistakes can be redone by using the up arrow key
   - For longer code, use a text editor (built into some GUIs)

2. Data entry
   - Using built in data sets for examples
   - Copying from another program
   - Reading a text or csv file
   - Importing from SPSS or SAS
   - Simulate it (using various simulation routines)

3. Descriptives
   - Graphical displays
   - Descriptive statistics
   - Correlation

4. Inferential
   - the t test
   - the F test
   - the linear model
Data entry overview

1. Using built in data sets for examples
   - `data()` will list > 100 data sets in the datasets package as well as all sets in loaded packages.
   - Most packages have associated data sets used as examples
   - `psych` has > 40 example data sets

2. Copying from another program
   - use copy and paste into R using `read.clipboard` and its variations

3. Reading a text or csv file
   - read a local or remote file

4. Importing from SPSS or SAS

5. Simulate it (using various simulation routines)
Examples of built in data sets from the psych package

```r
> data(package="psych")
```

- Bechtoldt: Seven data sets showing a bifactor solution.
- Dwyer: 8 cognitive variables used by Dwyer for an example.
- Reise: Seven data sets showing a bifactor solution.
- bfi: 25 Personality items representing 5 factors.
- blot: Bond's Logical Operations Test - BLOT.
- burt: 11 emotional variables from Burt (1915).
- epi.bfi: 13 personality scales from the Eysenck Personality Inventory and Big 5 inventory.
- flat (affect): Two data sets of affect and arousal scores as a function of personality and movie conditions.
- galton: Galton's Mid parent child height data.
- iqitems: 14 multiple choice IQ items.
- msq: 75 mood items from the Motivational State Questionnaire for 3896 participants.
- neo: NEO correlation matrix from the NEO_PI_R manual.
- sat.act: 3 Measures of ability: SATV, SATQ, ACT.
- Thurstone: Seven data sets showing a bifactor solution.
- veg (vegetables): Paired comparison of preferences for 9 vegetables.
Reading data from another program – using the clipboard

1. Read the data in your favorite spreadsheet or text editor
2. Copy to the clipboard
3. Execute the appropriate `read.clipboard()` function with or without various options specified

```r
my.data <- read.clipboard()  # assumes headers and tab or space delimited
my.data <- read.clipboard.csv()  # assumes headers and comma delimited
my.data <- read.clipboard.tab()  # assumes headers and tab delimited
  (e.g., from Excel)
my.data <- read.clipboard.lower()  # read in a matrix given the lower
my.data <- read.clipboard.upper()  # or upper off diagonal
my.data <- read.clipboard.fwf()  # read in data using a fixed format width
  (see read.fwf for instructions)
```

4. `read.clipboard()` has default values for the most common cases and these do not need to be specified. Consult `?read.clipboard` for details.
Reading from a local or remote file

1. Perhaps the standard way of reading in data is using the read command.
   - First must specify the location of the file
   - Can either type this in directly or use the file.choose function
   - The file name/location can be a remote URL

2. Two examples of reading data

```r
file.name <- file.choose()  #this opens a window to allow you find the file
my.data <- read.table(file.name)
datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
data.ex1=read.table(datafilename,header=TRUE)  #read the data into a table
> dim(data.ex1)  #what are the dimensions of what we read?
[1] 18 2
> describe(data.ex1) #do the data look right?
   var   n  mean   sd median trimmed  mad min max range skew  kurtosis    se
Dosage* 1 18  1.89 0.76     2  1.88  1.48  1.48  1  3   2 0.16   -1.12  0.18
Alertness 2 18 27.67 6.82    27 27.50  8.15  7.15 17 41  24 0.25   -0.68  1.61
```
read a “foreign” file e.g., an SPSS sav file

read.spss reads a file stored by the SPSS save or export commands.

read.spss(file, use.value.labels = TRUE, to.data.frame = FALSE,
           max.value.labels = Inf, trim.factor.names = FALSE,
           trim_values = TRUE, reencode = NA, use.missings = to.data.frame)

- **file**: Character string: the name of the file or URL to read.
- **use.value.labels**: Convert variables with value labels into R factors with those levels?
- **to.data.frame**: return a data frame? Defaults to FALSE, probably should be TRUE in most cases.
- **max.value.labels**: Only variables with value labels and at most this many unique values will be converted to factors if use.value.labels = TRUE.
- **trim.factor.names**: Logical: trim trailing spaces from factor levels?
- **trim_values**: logical: should values and value labels have trailing spaces ignored when matching for use.value.labels = TRUE?
- **use.missings**: logical: should information on user-defined missing values be used to set the corresponding values to NA?
Simulate data

For many demonstration purposes, it is convenient to generate simulated data with a certain defined structure. The `psych` package has a number of built-in simulation functions. Here are a few of them.

1. Simulate various item structures
   - `sim.congeneric` A one factor congeneric measure model
   - `sim.items` A two factor structure with either simple structure or a circumplex structure.
   - `sim.rasch` Generate items for a one parameter IRT model.
   - `sim.irt` Generate items for a one-four parameter IRT Model

2. Simulate various factor structures
   - `sim.simplex` Default is a four factor structure with a three time point simplex structure.
   - `sim.hierarchical` Default is 9 variables with three correlated factors.
Get the data and look at it

Read in some data, look at the first and last few cases, and then get basic descriptive statistics. For this example, we will use a built in data set.

```r
> my.data <- epi.bfi
> headtail(my.data)

epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen bdi traitanx stateanx
1 18 10 7 3 9 138 96 141 51 138 1 24 22
2 16 8 5 1 12 101 99 107 116 132 7 41 40
3 6 1 3 2 5 143 118 38 68 90 4 37 44
4 12 6 4 3 15 104 106 64 114 101 8 54 40
... ... ... ... ... ... ... ... ... ... ... ... ...
228 12 7 4 3 15 155 129 127 88 110 9 35 34
229 19 10 7 2 11 162 152 163 104 164 1 29 47
230 4 1 1 2 10 95 111 75 123 138 5 39 58
231 8 6 3 2 15 85 62 90 131 96 24 58
```

epi.bfi has 231 cases from two personality measures
Using R in class

• Most examples from class will be done in R and will show the code
  • Usually this will just be one or two lines.
• The (sporadic) homework will be done in R.
  • Can do with any other program, just the answers will show R code.
• For more help, look at the various tutorials and short courses available at http://personality-project.org/r/book
• Read the chapters, do the examples.