

Psychology 405: Psychometric Theory

Getting Started with R

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA



Spring, 2024

Use R



Outline

What is R?

Where did it come from, why use it?

Install R

Installing R on your computer and adding packages

Implementations

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
6. Many journals and chapters include R code appendices to allow for open science.

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\$stat
- Specialized programs
 - Mx (OpenMx is part of R)
 - EQ\$
 - AMO\$
 - LI\$REL
 - MPlu\$
 - Your favorite program

R: A way of thinking (from the fortunes package)

- “R is the lingua franca of statistical research. Work in all other languages should be discouraged.” (Jan de Leeuw , 2003)([Mizumoto & Plonsky, 2015](#))
- “Evelyn Hall: I would like to know how (if) I can extract some of the information from the summary of my nlme. Simon Blomberg: This is R. There is no if. Only how. – Evelyn Hall and Simon 'Yoda' Blomberg R-help (April 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)
- Actually, I see it as part of my job to inflict R on people who are perfectly happy to have never heard of it. Happiness doesn't equal proficient and efficient. In some cases the proficiency of a person serves a greater good than their momentary happiness. – Patrick Burns R-help (April 2005)

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

More fortunes

“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)

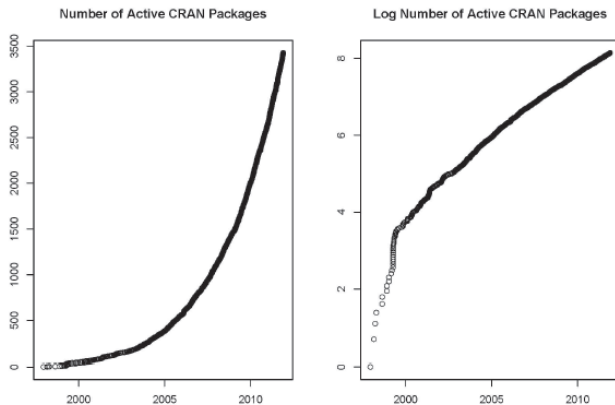
R is open source, how can you trust it?

- 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)

R: A brief history

- 1991-93: Ross Ihaka 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)
- 2001-2019: 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
 - 2016-03-21 8,120 packages (on CRAN) + 1,104 bioinformatic packages + ?,000s on GitHub/R-Forge
 - 2020-04-04 15,514 packages (CRAN) + 1,823 on BioConductor + ?,000s on GitHub
 - 2022-03-25 19,017 packages (CRAN) + 1,974 on BioConductor > 70,000 on GitHub
 - 2024-03-25 21,356 packages (CRAN) + 2,366 on BioConductor > 70,000 on GitHub

Has R grown too much? Exponential growth rate continues



See also <http://r4stats.com/articles/popularity/>

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 4.3.3) with 4.4.0 coming 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

Annotated installation guide: don't type the >

```
> 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 $\geq 4.3.3$) and for psych ($\geq 2.4.3$)

R Under development (unstable) (2023-03-17 r83997) -- "Unsuffered Consequences"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (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 1.79 (8196) aarch64-apple-darwin20]

[Workspace restored from /Users/WR/.RData]

[History restored from /Users/WR/.Rapp.history]

Run sessionInfo to check version numbers

R code

```
library(psych,psychTools)
sessionInfo()
```

R Under development (unstable) (2023-03-17 r83997)

Platform: aarch64-apple-darwin20 (64-bit)

Running under: macOS Ventura 13.2.1

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib

Random number generation:

RNG: Mersenne-Twister

Normal: Inversion

Sample: Rounding

locale:

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

time zone: America/Chicago

tzcode source: internal

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] psychTools_2.3.3 psych_2.3.3

loaded via a namespace (and not attached):

[1] compiler_4.3.0 tools_4.3.0 parallel_4.3.0 foreign_0.8-84 nlme_3.1-162 mnormt_2.1.1 grid_4.2-0

[8] lattice_0.20-45

R is extensible: The use of “packages”

- More than 21,356 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 400 matches and reports the top 400 and download 400 links to 215 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
- Special use
 - ltm
 - sem
 - lavaan
 - OpenMx
 - GPArotation
 - mvtnorm
 - > 21,356 known
 - + ?
- General applications
 - most descriptive and inferential stats
 - Modern Applied Statistics with S
 - Lattice or Trellis graphics
 - Linear mixed-effects models
 - Personality and psychometrics
- 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 development versions of R)
 - Can be specified as “another repository” (e.g.,
<https://personality-project.org/r>)
3. Can also do this by command

```
install.packages("psych", repos="https://personality-project.org/r/",
                 type="source")
```

```
trying URL 'http://personality-project.org/r/src/contrib/psych_2.1.3.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
```

Implementations

1. Base R in the Unix/Linux/Mac X11 framework
2. Base R on the Mac/PC
 - Mac has prompts at bottom of window
 - Convenient interface, particularly for development
3. Graphical User Interfaces
 - R Commander
 - **R studio** as a convenient shell
4. R studio integrates with *Rmarkdown* to allow easy note taking /annotation while writing scripts
5. Documenting scripts allows for more open science (and helps you remember what you have done)

RStudio (particularly nice for PCs)

The screenshot displays the RStudio integrated development environment (IDE) interface. The top toolbar includes icons for file operations, running code, and viewing help. The main window is divided into four panes:

- Source Editor (Top Left):** Contains a script file named '01_hello.R'. The script includes comments and R code for loading the 'psych' package, creating a data frame 'data' with columns for 'gender', 'education', and 'age', and calculating correlations between these variables.
- Console (Bottom Left):** Shows the output of the R script, including the message 'R under development (unstable) (2024-01-25 r8486)' and the successful execution of the 'psych' package.
- Environment/History (Top Right):** Displays the objects in the current environment, including 'data' and 'psych'.
- Plots (Bottom Right):** Shows a 'Correlation plot' as a heatmap. The x and y axes are labeled with the variables 'gender', 'education', and 'age'. The color scale ranges from -0.8 (blue) to 1.0 (red), indicating the strength and direction of the correlations.

Using R

1. Install the relevant packages (just once!)
 - Either one at a time, or by using a “task view”
2. Make the packages you want to use “active” by `library(package name)` e.g., `library(psych)`, `library(psychTools)`.
 - You need to do this for each session or it
 - Can be automatized at startup with

```
.First() <- function() {library(psych); library(psychTools)}
```
3. Use the functions in a package
 - To see all functions in a package go to the index of the package or use the objects function: e.g., `objects(package:psych)`
 - Apply a function to data
 - All functions require an object to act upon. Most require this in parentheses. All functions return an object. This may be saved for later.
 - `function(object)` #apply the function to the object, show it
 - `sqrt(2)`
 - `result <- function(object)` #apply the function to object, save it
 - `answer <- alpha(ability)` #lots and lots of output is saved

What makes R useful?

1. The basic philosophy of open source allows one to see the code and modify it.
2. Object oriented meaning that all functions return objects that can be acted upon by other functions.
 - e.g., output of mean and sd is included in describe or describeBy
 - Output of cor is used in fa, alpha, etc.
 - Can add other functions (e.g. tetrachoric to do correlations instead of cor)
3. This is perhaps the most powerful feature of R!
4. R is vectorized, so that if it works for X , it will work for \vec{X}

Basic R commands – remember don't enter the >

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
```

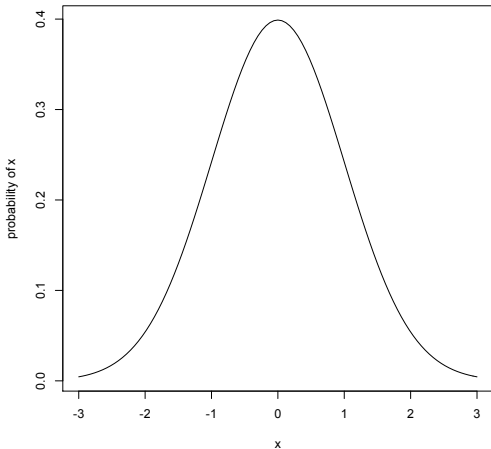
R is a set of distributions. Don't buy a stats book with tables!

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

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

R can draw distributions

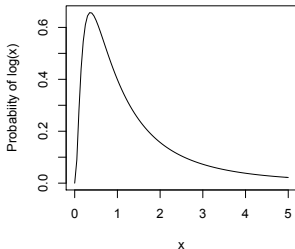
A normal curve



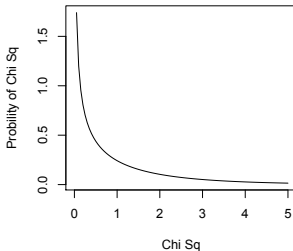
```
curve(dnormal(x),-3,3,  
      ylab="probability of  
x",main="A normal  
curve")
```

R can draw more interesting distributions

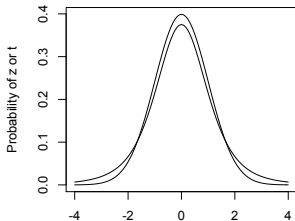
Log normal



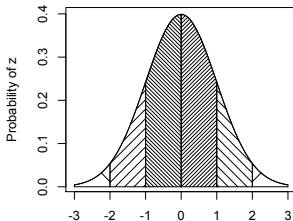
Chi Square distribution



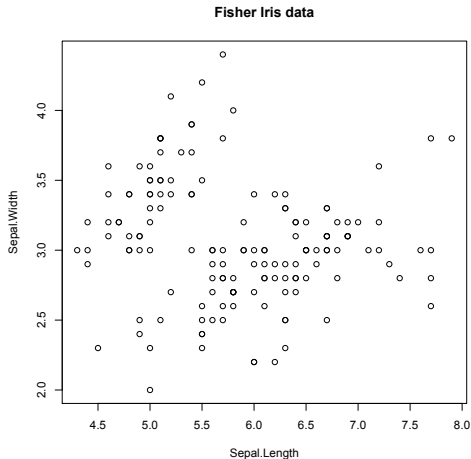
Normal and t with 4 df



The normal curve

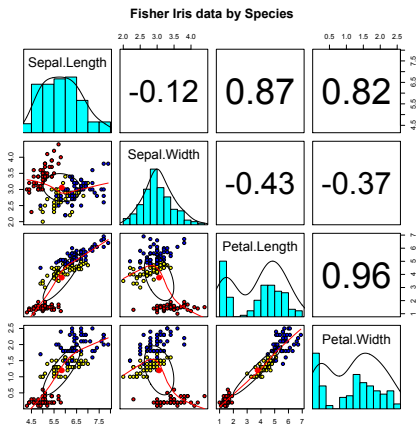


A simple scatter plot using plot



```
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` /images

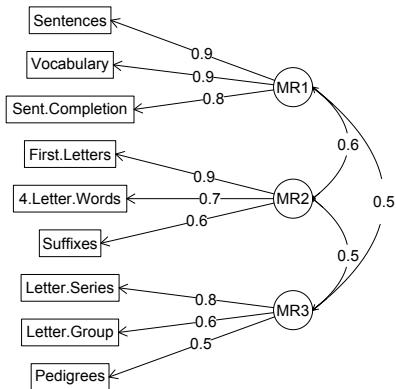


1. Correlations above the diagonal
2. Diagonal shows histograms and densities
3. scatter plots below the diagonal with correlation ellipse
4. locally smoothed (loess) regressions for each pair
5. optional color coding of grouping variables.

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

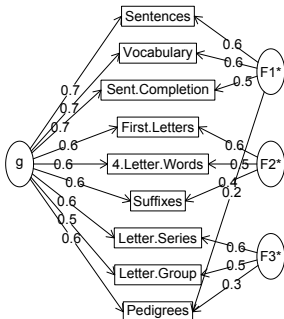
The simple factor structure

Factor Analysis

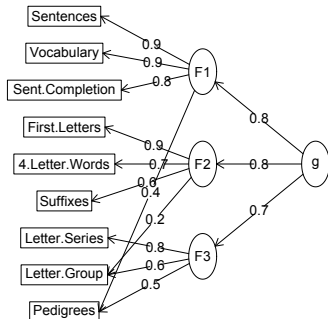


Two ways of viewing the higher order structure

Omega



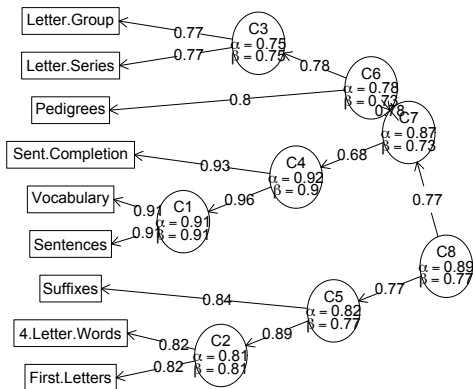
Hierarchical (multilevel) Structure



A hierarchical cluster structure found by iclust

iclust(Thurstone)

iclust



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
 - *psychTools* has another > 60 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) (always may sure you refer to simulated data!)

Examples of built in data sets from the psych package

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

Bechtoldt

Seven data sets showing a bifactor solution.

Dwyer

8 cognitive variables used by Dwyer for an exam

Reise

Seven data sets showing a bifactor solution.

all.income (income)

US family income from US census 2008

bfi

25 Personality items representing 5 factors

blot

Bond's Logical Operations Test - BLOT

burt

11 emotional variables from Burt (1915)

cities

Distances between 11 US cities

epi.bfi

13 personality scales from the Eysenck Personality
and Big 5 inventory

flat (affect)

Two data sets of affect and arousal scores as a
personality and movie conditions

galton

Galton's Mid parent child height data

income

US family income from US census 2008

iqitems

14 multiple choice IQ items

msq

75 mood items from the Motivational State Questi
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 vegetable

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

```
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

- 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

- Two examples of reading data

```
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	kurtosi
Dosage*	1	18	1.89	0.76	2	1.88	1.48	1	3	2	0.16	-1.1
Alertness	2	18	27.67	6.82	27	27.50	8.15	17	41	24	0.25	-0.6

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.

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

	epiE	epiS	epiImp	epilie	epiNeur	bfragee	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	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.

Mizumoto, A. & Plonsky, L. (2015). R as a Lingua Franca: Advantages of Using R for Quantitative Research in Applied Linguistics. *Applied Linguistics*, 37(2), 284–291.