Writing Functions
 Basic Commands
 Writing a function
 Simulations
 Useful functions
 Even more help
 References
 References

 0
 00000
 00
 0000
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00</td

Psychology 350: Special Topics An introduction to R for psychological research Writing functions (part 2)

William Revelle Northwestern University Evanston, Illinois USA

https://personality-project.org/courses/350



May, 2024



 Writing Functions
 Basic Commands
 Writing a function
 Simulations
 Useful functions
 Even more help
 References
 References

 0
 00000
 00
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000
 00000</td

Outline

Writing Functions

The basic commands Input and moving around

Writing a function

Simulations as way of testing theory

Useful functions

And even more help

References





Functions allow for more compact scripts

Three basic approaches to using R

- 1. Do everything from console run each line as you think about it.
 - Basic step through your operations, use R like a desk calculator
 - Try a command (calling some functions from a package),
- 2. Use the text window and write out a few lines at a time and then run them (ala what we have been doing in class)
 - Keep a list of your most useful set of lines
 - Annotate them to remind you what you have done
 - Use RMarkdown to annotate your scripts and to show your output
- 3. Write little (initially) functions to which you can pass parameters and make into your own library of useful functions
- 4. You can then 'source' these functions when you do your work,.



A few of the most useful data manipulations functions (adapted from Rpad-refcard). Use ? for details

file.choose	() find a file	dim	(x) dimensions of x
file choose	(new_TRUE) create a	str	(x) Structure of an object
me.choose	new file	list	() create a list
read.table	(filename)	colnames	(x) set or find column
read.csv	(filename) reads a		names
	comma separated file	rownames	(x) set or find row names
read.delim	(filename) reads a tab	ncol(x), nrow(z)	number of row, columns
delin	elimited file	rbind	() combine by rows
С	() combine arguments	cbind	() combine by columns
from:to	e.g., 4:8	is.na	(x) also is.null(x), is
seq	(from,to, by)	na.omit	(x) ignore missing data
rep	(x,times,each) repeat \times	table	(x)
gl	(n,k,) generate factor	merge	(x,y)
	levels	apply	(x,rc,FUNCTION)
matrix	(x,nrow=,ncol=) create a matrix	ls	() show workspace
data.frame	() create a data frame	rm	() remove variables from workspace 4

4/18



Input commands

file.choose () search for a file on the computer. Returns the path to the file. fn<- file.choose() file.choose (new=TRUE) create a new file. Uses the normal computer search functions, but the creates a new file. read.table(filename) Read tabular input from the file location fn. read.csv (filename) reads a comma separated file read.delim (filename) reads a tab delimited file load(filename) "loads" objects from an .Rds file read.file(filename)* combines file.choose and appropriate read command (e.g., read.table, read.csv, load, etc.).

* psych or ** psychTools function. These require you to library(psych) and library(psychTools)



Writing Functions O	Basic Commands	Writing a function	Simulations 0000	Useful functions	Even more help O	References	References
Input and moving around							
Data description							
ls() List the objects in your workspace.							

- rm(list=ls()) Remove all the objects in your workspace.
 - dim(x) What are the size of the dimensions of x. Does not work for a vector.
 - length(x) Length of a vector or a list
 - names(x) Names of the various objects inside object x.
 - str(x) What is the structure of x?

summary(x) Give the summary statistics of the object x. More
useful for results of various functions. Not
particularily useful for raw data

table(x) Tabulate the elements of a vector (e.g., table(sat.act["education"))





More data description and manipulation

matrix (x,nrow=,ncol=) create a matrix

- data.frame (...) create a data frame
 - list (...) create a list
 - - <- c("first", "second") set the names of the first two columns.
- rownames (x) set or find row names
- NCOL(x), NROW(z) Number of rows and columns of a vector, matrix, or data.frame
 - rbind (...) combine two matrices or data frames by rows (ncol() and colnames must match)
 - cbind (...) combine two matrices or data frames by columns (Number of rows much match)





Data manipulation

- c (...) combine (concatenate) arguments
- cs() ** concatenate and convert to character strings
- from:to e.g., 4:8
 - seq (from,to, by) seq(from=5, to =15, by = 2)
 - rep (x,times,each) repeat x rep(c(1,2,3),4) compare
 with rep(c(1,2,3),each=4)
 - == equals
 - != does not equal
- is.na(x) or is.null(x) is a missing value or is a NULL value
 - na.omit (x) ignore missing data
 - merge (x,y) Combine two data.frames or matrices based upon an id field. A very powerful function that has a long and useful help menu.
 - apply (x,rc,FUNCTION). On either the rows (rc=1) or columns (rc=2) do the function (see also lapply).





Logical tests and repetitive operations

- - if(x) If x is TRUE, then do something; e.g., if (x < 10) {do something }
- if(condition) { do something ... } else {do something else } if the condition is TRUE then do something, otherwise, do something else.

while(x) As long as x is TRUE, the do something

switch(index, first = {...}, second = {...}) Multiple alternatives depending upon index.





The structure of a function

- 1. Function definition includes a list of the parameters to be passed to the function.
- 2. "myfunction" <- function(parameter list) { ... body of function }
- 3. The parameters can be given default values if desired.
- 4. First part of the function should check if the parameters make sense
- 5. Body of function does something useful
- 6. Last line of function returns a value (can be any type of object, including a list).





Writing a function

- 1. Decide what it should do and give it a reasonable (memorable) name.
- 2. List the parameters that the function will take. Give default values if this is useful.
- 3. Write the body of the function
- 4. Try running it.
- Debug by debug(functionname) (Turn off debugging by undebug(functionname)
- 6. Use the function.

7. To see how functions work, just type the name of the function. (See http://personality-project.org/courses/350/350.wk8.html for worked examples.)



Theory testing through simulation

- 1. Some problems do not have an obvious answer
- 2. We can simulate these to explore the parameter space
- 3. Bootstrapping (random sampling with replacement) and simple random sampling can be used for demonstrations
- 4. Consider the question of how to weight items when forming scales
 - Should we optimally weight (for the sample) using factor analysis weights
 - Or should we use unit weights (all items treated equally)
- 5. A recent paper suggests factor weights, others of us think unit weights (Widaman & Revelle, 2022)
- 6. Lets simulate this.





Simulation: detail

- the spi data set has 4,000 observations with 10 criteria and 135 personality items
- 2. The Neuroticism scale of 14 items can be used as an example
- 3. For each of 100 random samples of size 200 from the 4000,
 - Form unit weighted scores
 - Form factor scores (based upon a one dimensional factor analysis)
 - Find the validities of these scores for the 10 criteria
 - Cross validate on the other 3800 subjects
- 4. Pool these 100 samples
- 5. Graph the results
- 6. This small function calls several other functions (the power of R is this ability to use other functions)

See http://personality-project.org/courses/350/350.wk7c.html







Simulation results

reighted and factor scores derivation and cross validation for Neuroticism

unit.crossp1edu							
unitp1edu						·····	
unit.crossneaith							
unimeann footor aroan ED					0	-0-	
factor crossemoke					ő		
factorsmoke					· · · · · · · · · · · · · · · · · · ·		
factorER					· · · · · · ·		
factorwellness					- 		
factor.crosswellness					0		
factorexer					0-		
factor.crossexer					0		
factorp1edu				····			
factorhealth				-			
factor.crossp1edu				0			
factor.crosshealth				0			
UNIT.CLOSSER				0			
unit.crosssmoke				8			
unit.crossweimess				5			
			-	2			
unitemoko			· · · · · · · · · · · · · · · · · · ·	<u></u>			
unitwollnose			è				
unitexer			ē				
factorsex		e					
unitage							
factorage							
unitsex							
factor.crosssex		· · · · · · · · · · • • • • • • • • • •					
unit.crossage		• • • • • • •					
factoreducation		·····					
tactorp2edu							
lactor.crossage		ē					
factor crossn2odu		ő					
factor crossoducation		ő					
uniteducation							
unito2edu	- ŏ-						
unit crosseducation	o						
unit.crossp2edu	0						
	L						
	-0.3	-0.2	-0.1	0.0	0.1	0.2	



 $15 \, / \, 18$

Writing FunctionsBasic CommandsWriting a functionSimulationsUseful functionsEven more helpReferencesReferences000000000

A few of the most useful data manipulations functions (adapted from Rpad-refcard). Use ? for details

file.choose	() find a file	dim	(x) dimensions of x
file choose	(new-TRUE) create a	str	(x) Structure of an object
me.choose	new file	list	() create a list
read.table	(filename)	colnames	(x) set or find column
read.csv	(filename) reads a		names
	comma separated file	rownames	(x) set or find row names
read.delim	(filename) reads a tab	ncol(x), nrow(z)	number of row, columns
	delimited file	rbind	() combine by rows
С	() combine arguments	cbind	() combine by columns
from:to	e.g., 4:8	is.na	(x) also is.null(x), is
seq	(from,to, by)	na.omit	(x) ignore missing data
rep	(x,times,each) repeat \times	table	(x)
gl	(n,k,) generate factor	merge	(x,y)
	levels	apply	(x,rc,FUNCTION)
matrix	(x,nrow=,ncol=) create a matrix	ls	() show workspace
data.frame	() create a data frame	rm	() remove variables from workspace 16

16 / 18

Writing Functions
0Basic Commands
0Writing a function
0Simulations
0000Useful functions
0Even more help
0ReferencesReferences

More useful statistical functions, Use ? for details

- mean (x,na.rm=TRUE) *
- is.na (x) also is.null(x), is...
- na.omit (x) ignore missing data
 - sum (x)
- rowSums (x) see also colSums(x)
- colSums (x) see also rowSums(x) min (x,na.rm=TRUE)*

 - max (x) *ignores NA values
 - range (x)
 - table (x)
- summary (x) depends upon x
 - sd (x) standard deviation
 - cor (x,use="pairwise") correlation
 - cov (x) covariance
 - solve (x) inverse of x
 - lm (y~x) linear model
 - aov (y~x) ANOVA

- Selected functions from psych package
 - describe (x) descriptive stats
 - describeBy (x,y) descriptives by group
 - pairs.panels (x) SPLOM
 - error.bars (x) means + error bars
 - error.bars.by (x) Error bars by groups
 - fa (x,n) Factor analysis
 - principal (x,n) Principal components
 - iclust (x) Item cluster analysis
 - scoreltems (x) score multiple scales
- score.multiple.choice (x) score multiple choice scales
 - alpha (x) Cronbach's alpha
 - omega (x) MacDonald's omega
 - irt.fa (x) Item response theory through factor analysis
 - mediate (y,x,m,data) Mediation/moderation





More help

- An introduction to R as HTML, PDF or EPUB from http://cran.r-project.org/manuals.html (many different links on this page
- 2. FAQ General and then Mac and PC specific
- R reference card http://cran.r-project.org/doc/ contrib/Baggott-refcard-v2.pdf
- 4. Various "cheat sheets" from RStudio http://www.rstudio.com/resources/cheatsheets/
- 5. Using R for psychology http://personality-project.org/r/
- 6. Package vignettes (e.g., http://personality-project. org/r/psych/vignettes/overview.pdf)
- 7. R listserve, StackOverflow, your students and colleagues



 Writing Functions
 Basic Commands
 Writing a function
 Simulations
 Useful functions
 Even more help
 References
 References

Widaman, K. F. & Revelle, W. (2022). Thinking thrice about sum scores, and then some more about measurement and analysis. *Behavior Research Methods*.

