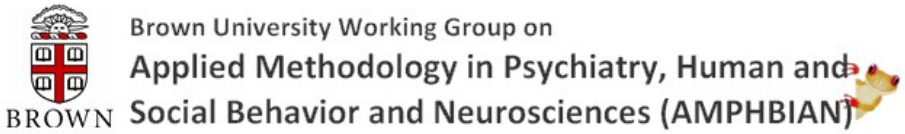
AMPHIBIAN Working Group 2014-2015

Switch2R 02

Rich Jones

Monday, November 10, 2014



2014-2015 presentation series: Switch2R

- October 2014: R basics
- November 2014: Clean data with R
- December 2014 Analysis with R
- January 2015 Graphics with R (1/2)
- February 2015: Graphics with R (2/2)
- March 2015 Reproducible research with R (Dale)
- April 2015: GIS with R
- May 2015: Power and sample size
- June, July Aug. 2015: Summer break

Updates from October 2014

- If you are a Windows user, you should be using Revolution R Open
  - *It is much faster*
  - *Otherwise, no big difference from regular R*
  - *Integrates seamlessly with RStudio*
  - *A few weeks behind new R releases*
- If you are a Mac user, don't use Revolution R Open
  - *Unless someone knows why you should*
- If you are a Mac user, there was an issue with R, RStudio with the Yosemite Mac OS X 10.10 update
  - *read more about it here*

November 2014: Clean Data with R

- Use R to clean data
  - *read data (review from last time)*
  - *make new variables*
  - *produce dataset documentation*
  - *label variables and values (sjPlot)*

Things to remember about R

- There are about 57 different ways to do the same thing
- There is undoubtedly a better way to do the things I am going to show you
- When you find out please share

Things to remember about data in R

- R *natively* does not support variable and value labels
- You can load multiple data sets in R at once
- Data can be
  - *in data frames*
  - *in matrix*
  - *attached (but don't do that)*
  - *in a list*

What are clean data?

- Clean data are *tidy* data
- Tidy data are not *messy* data
- Tidy data have the following characteristics
  - *1 variable : 1 column*
  - *1 observation : 1 row*
  - *Each type of observational unit forms a table, or,*
  - *data are only stored in one place (you may need different tables)*

Example from last month

Atkins, D. C. (2005). Using multilevel models to analyze couple and family treatment data: Basic and advanced issues. *Journal of Family Psychology*, 19, 98-110.

Data from UCLA IDRE

Article: findit at Brown full text (http://goo.gl/D6m2ID)

R Session

- Decide where you will do your work
- Set the working folder
- Download data
- Write (and save) code

What I do

- I work in folder *c:/work/project-name-here*

```
work <- "c:/work/shows/switch2r"
setwd(work)
getwd()
```

```
## [1] "c:/work/shows/switch2r"
```

Load some packages

(note: order matters)

```
# install.packages("lubridate")
# install.packages("psych")
# install.packages("gmodels")
# install.packages("Hmisc")
# install.packages("sjPlot")
require(lubridate) # easy handling of dates and times
require(psych)    # using the scoreItems command
require(Hmisc)    # using the describe command
require(sjPlot)   # tools for reading SPSS formatted data
require(gmodels)  # using the CrossTable command
```

Read in the data set from the web

```
# Read data from UCLA Web site
url <- "http://www.ats.ucla.edu/stat/paperexamples/atkins_mlm/Atkins_JFP_data.txt"
data <- read.csv(url, sep="\t", header=TRUE)
# I like all lowercase variable names
names(data)<- tolower(names(data))
# show the first 15 lines
head(data, 15)
```

```
##    id sex therapy time       das pilot miss m.ind
## 1   1   0    -0.5    0  94.51204     1    0     1
## 2   1   0    -0.5   13  87.53364     1    0     1
## 3   1   0    -0.5   26  81.46659     1    1     1
## 4   1   0    -0.5   35  83.44614     1    1     1
## 5   1   1    -0.5    0  81.27981     1    0     1
## 6   1   1    -0.5   13  68.80343     1    0     1
## 7   1   1    -0.5   26  71.16971     1    1     1
## 8   1   1    -0.5   35  76.88723     1    1     1
## 9   2   0     0.5    0  79.53347     1    0     0
## 10  2   0     0.5   13  98.75209     1    0     0
## 11  2   0     0.5   26 100.15992     1    0     0
## 12  2   0     0.5   35 127.55920     1    0     0
## 13  2   1     0.5    0 106.98098     1    0     0
## 14  2   1     0.5   13 124.13277     1    0     0
## 15  2   1     0.5   26 143.09363     1    0     0
```

Are these data *tidy*?

- Discuss

Make some variables

The data file has a variable *therapy* that is coded -.5/+.5.

Let's say I'd like to have a version of that variable that was coded 1/2.

I'll call it *tx*

```
data$tx <- 1 # New variable tx in data.frame data = 1 (by default)
data$tx[which(data$therapy==0.5)] <- 2 # if therapy == 0.5 data$tx = 2
str(data[,c("tx","therapy")]) # show the characteristics of two variables
```

```
## 'data.frame':    1072 obs. of  2 variables:
##  $ tx     : num  1 1 1 1 1 1 1 2 2 ...
##  $ therapy: num  -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 0.5 0.5 ...
```

```
# Now check
table(data$tx, data$therapy) # a crosstab of tx and therapy in data.frame data
```

```
##
##     -0.5 0.5
##   1  536   0
##   2    0 536
```

A nicer looking cross-tab

Using *CrossTable* from the *gmodels* package

```
CrossTable(data$tx, data$therapy,
  missing.include=TRUE,
  prop.r=FALSE,
  prop.c=FALSE,
  prop.t=FALSE,
  prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  1072
##
##
##              | data$therapy
##      data$tx |      -0.5 |       0.5 | Row Total |
## -------------|------------|------------|-----------|
##            1 |       536 |         0 |       536 |
## -------------|------------|------------|-----------|
##            2 |         0 |       536 |       536 |
## -------------|------------|------------|-----------|
## Column Total |       536 |       536 |      1072 |
## -------------|------------|------------|-----------|
##
##
```

But that is so cumbersome to code…

Welcome to R

```
tab <- function(r,c) {
  CrossTable(r, c,
    missing.include=TRUE, prop.r=FALSE, prop.c=FALSE, prop.t=FALSE,
    prop.chisq=FALSE)
}
```

Now a nice cross-tab

```
tab(data$tx,data$therapy)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  1072
##
##
##              | c
##            r |      -0.5 |       0.5 | Row Total |
## -------------|-----------|-----------|-----------|
##            1 |       536 |         0 |       536 |
## -------------|-----------|-----------|-----------|
##            2 |         0 |       536 |       536 |
## -------------|-----------|-----------|-----------|
## Column Total |       536 |       536 |      1072 |
## -------------|-----------|-----------|-----------|
##
##
```

And anyway the coding looks right. But notice the variable names are now the not-so-helpful “r” and “c”. So whatever.

Sex

Another example. I'll code a variable *male* from the original *sex*

```
data$male <- 0 # initialize to 0
data$male[which(data$sex!=1)] <- 1 # sex is 0:Husband 1:Wife
tab(data$male,data$sex)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  1072
##
##
##              | c
##            r |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |         0 |       536 |       536 |
## -------------|-----------|-----------|-----------|
##            1 |       536 |         0 |       536 |
## -------------|-----------|-----------|-----------|
## Column Total |       536 |       536 |      1072 |
## -------------|-----------|-----------|-----------|
##
##
```

About indicator variables

male is a binary indicator (0/1). When coding binary indicators with the values (0/1), always code so that the name of the variable always matches the label for value 1. For example,

R sex is really	Categorical Variable Gender codes	Binary indicator Female	Binary indicator Male
a man	1 = male	0	1
a woman	2 = female	1	0

Dates

The example data set does not have any date information.

Public data rarely will.

But you will surely have to deal with dates.

So let's pretend it has dates.

To pretend it has dates, we will generate some date data. So now you get to see how we generate data with R.

```
set.seed(3481)
data$year <- sample(1:3,nrow(data),replace=T)+2008
data$day <-  sample(1:27,nrow(data),replace=T)
data$month <- round(runif(nrow(data),0.51,12.49))
```

The first line says: make a new variable *year* in data frame *data*. *Assign* to it values, *sample* from the range 1 to 3, as many times as there are *rows* in the data frame named data. Sample with replacement. Then add 2008.

The third line says: make a new variable *month* in data frame *data*. *Assign* to it a random numbers (as many as there are rows in the data frame) drawn from a uniform distribution ranging from 0.51 to 12.49. Oh but round it to the nearest whole number too.

Display the *str*ucture of the day month year variables

```
str(data[,c("day","month","year")])
```

```
## 'data.frame':    1072 obs. of  3 variables:
##  $ day  : int  5 1 27 16 3 18 19 24 3 6 ...
##  $ month: num  2 4 3 10 5 5 5 11 3 5 ...
##  $ year : num  2011 2010 2010 2010 2011 ...
```

Tabulate the day month year variables

```
table(data$day)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## 41 37 45 43 37 49 34 36 44 31 32 33 40 37 42 33 44 35 47 40 49 35 37 47 42
## 26 27
## 42 40
```

```
table(data$month)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12
##  75  83  87  72 102  93  84  83  72 106 105 110
```

```
table(data$year)
```

```
##
## 2009 2010 2011
##  363  363  346
```

Dates using lubridate

(luridate vignette)[http://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html]

Some interesting things about dates

```
the.time.is.now <- now()
the.time.is.now.numeric <- as.numeric(now())
the.time.is.now
```

```
## [1] "2014-11-08 14:20:38 EST"
```

```
the.time.is.now.numeric
```

```
## [1] 1415474439
```

```
as.numeric(mdy("1/1/1970"))
```

```
## [1] 0
```

```
as.numeric(mdy("1/2/1970"))
```

```
## [1] 86400
```

```
60*60*24
```

```
## [1] 86400
```

```
str(the.time.is.now)
```

```
##  POSIXct[1:1], format: "2014-11-08 14:20:38"
```

Making a datetime variable

```
data$year <- as.character(data$year)
data$month <- as.character(data$month)
data$day <- as.character(data$day)
data$date <- mdy(paste0(data$month,"/",data$day,"/",data$year))
head(data[,c("year","month","day","date")])
```

```
##   year month day       date
## 1 2011     2   5 2011-02-05
## 2 2010     4   1 2010-04-01
## 3 2010     3  27 2010-03-27
## 4 2010    10  16 2010-10-16
## 5 2011     5   3 2011-05-03
## 6 2009     5  18 2009-05-18
```

New Example: National Comorbidity Study Replication (NCS-R 2002-2003)

Data from ICPSR (public use)

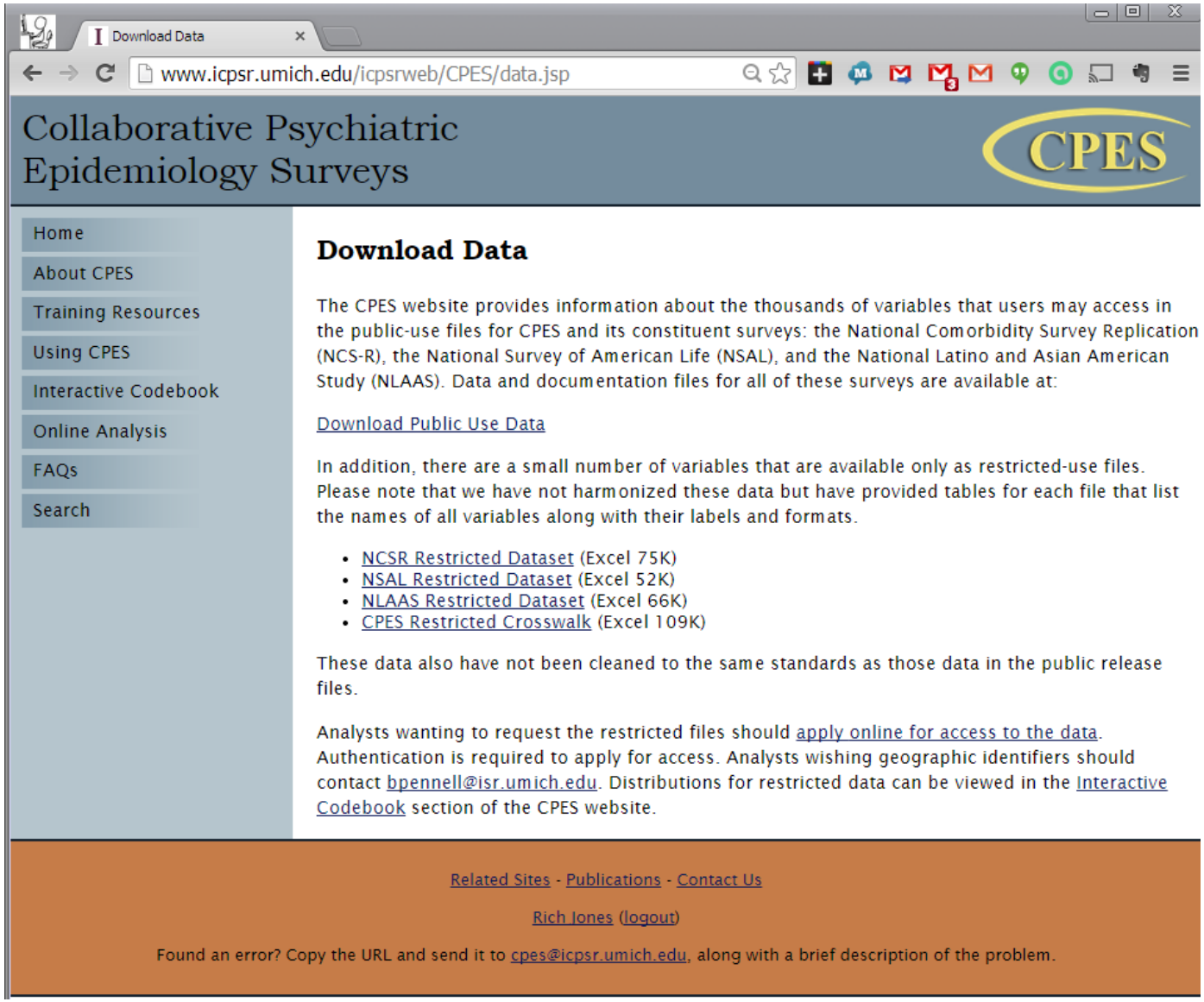
I can redistribute to Brown faculty, students, and staff (authorized users)

If you are not Brown faculty, students, and staff, see about getting access to the NCS-R public use files at ICPSR

NCS-R data

I am pulling the NCS-R data as archived along with the Collaborative Psychiatric Epidemiology Surveys.

CPES Website


Download Data
www.icpsr.umich.edu/icpsrweb/CPES/data.jsp
Collaborative Psychiatric Epidemiology Surveys
CPES
Home
About CPES
Training Resources
Using CPES
Interactive Codebook
Online Analysis
FAQs
Search
Download Data
The CPES website provides information about the thousands of variables that users may access in the public-use files for CPES and its constituent surveys: the National Comorbidity Survey Replication (NCS-R), the National Survey of American Life (NSAL), and the National Latino and Asian American Study (NLAAS). Data and documentation files for all of these surveys are available at:
Download Public Use Data
In addition, there are a small number of variables that are available only as restricted-use files. Please note that we have not harmonized these data but have provided tables for each file that list the names of all variables along with their labels and formats.
NCSR Restricted Dataset (Excel 75K)
NSAL Restricted Dataset (Excel 52K)
NLAAS Restricted Dataset (Excel 66K)
CPES Restricted Crosswalk (Excel 109K)
These data also have not been cleaned to the same standards as those data in the public release files.
Analysts wanting to request the restricted files should apply online for access to the data. Authentication is required to apply for access. Analysts wishing geographic identifiers should contact bpennell@isr.umich.edu. Distributions for restricted data can be viewed in the Interactive Codebook section of the CPES website.
Related Sites - Publications - Contact Us
Rich Jones (logout)
Found an error? Copy the URL and send it to cpes@icpsr.umich.edu, along with a brief description of the problem.

Download the SPSS data file

DS2: National Comorbidity Survey Replication (NCS-R), 2001-2003 -
Download All Files (401.5 MB) **large dataset**

Documentation: Codebook.pdf Documentation.pdf

Data: SAS SPSS Stata ASCII Delimited
ASCII + SAS Setup SPSS Setup Stata Setup

Analyze Online: SDA

Let's get set

```
work <- "c:/work/shows/switch2r" # my working folder you edit here
setwd(work) # change to it
getwd() # check it
```

```
## [1] "c:/work/shows/switch2r"
```

I've downloaded the SPSS file to a subfolder here (NCSR)

```
dir("./NCSR")
```

```
## [1] "20240-0002-Codebook.pdf"       "20240-0002-Data.sav"
## [3] "20240-0002-Data.txt"           "20240-0002-Documentation.pdf"
## [5] "20240-0002-Setup.sas"
```

We'll use the sjPlot package to read in the SPSS data

and generate a nice data dictionary

```
spssdata <- sji.SPSS("./NCSR/20240-0002-Data.sav")
# sji.viewSPSS(spssdata) # opens in the Rstudio Viewer Panel
# Or in a browser window if you're running the R Console
```

Let's save that codebook

```
#sji.viewSPSS(spssdata, file="codebook.html", showFreq = TRUE, useViewer = FALSE)
codebookpath <- file.path(work,"codebook.html")
```

The codebook

But that link only works on your (i.e., my) local computer

Viewers of this presentation can see the codebook I generated on my dropbox

Let's make a summated rating scale score

Psychotic experiences

- PS1A Ever see vision that others couldn't see
- PS1B Ever hear voices others couldn't hear
- PS1C Ever have mind control experience
- PS1D Ever feel mind taken over by strange forces
- PS1E Ever exp communication attempts from strange forces
- PS1F Unjust plot to harm you/have people follow-nobody believe

Each coded 1=Yes, 5=No

Look at the data

```
psitems <- c("PS1A", "PS1B", "PS1C", "PS1D", "PS1E", "PS1F")
describe(spssdata[psitems])
```

```
## spssdata[psitems]
##
##  6  Variables      9282  Observations
## --------------------------------------------------------------------------
## PS1A
##       n missing  unique    Info    Mean
##    2349    6933       2    0.25    4.63
##
## 1 (217, 9%), 5 (2132, 91%)
## --------------------------------------------------------------------------
## PS1B
##       n missing  unique    Info    Mean
##    2349    6933       2    0.17   4.753
##
## 1 (145, 6%), 5 (2204, 94%)
## --------------------------------------------------------------------------
## PS1C
##       n missing  unique    Info    Mean
##    2352    6930       2    0.02   4.968
##
## 1 (19, 1%), 5 (2333, 99%)
## --------------------------------------------------------------------------
## PS1D
##       n missing  unique    Info    Mean
##    2353    6929       2    0.01   4.981
##
## 1 (11, 0%), 5 (2342, 100%)
## --------------------------------------------------------------------------
## PS1E
##       n missing  unique    Info    Mean
##    2352    6930       2    0.04   4.951
##
## 1 (29, 1%), 5 (2323, 99%)
## --------------------------------------------------------------------------
## PS1F
##       n missing  unique    Info    Mean
##    2352    6930       2    0.05   4.937
##
## 1 (37, 2%), 5 (2315, 98%)
## --------------------------------------------------------------------------
```

Looks like a skip pattern

Kessler says…

- A total of 9282 respondents participated (2001-2003)
- All respondents completed a Part I diagnostic interview (WHO-CIDI)
- A probability sample of 5692 also received Part II (additional disorders)
- A random sub-sample of Part II respondents (n = 2322) was administered the NAP screen
- NAP (Non-affective psychosis)
- Our N=2349-2353 is pretty close to that

Kessler RC, Birnbaum H, Demler O, Falloon IR, Gagnon E, Guyer M, Howes MJ, Kendler KS, Shi L, Walters E. The prevalence and correlates of nonaffective psychosis in the National Comorbidity Survey Replication (NCS-R). Biol Psychiatry. 2005;58(8):668-76.

Sum score

```
key <- c(1,1,1,1,1,1) # the "right" answers
results <- scoreItems(
  items = spssdata[psitems],
  keys = key,
  totals = TRUE,
  missing = TRUE, # missing data are imputed
  impute = "none"  # person's non-missing mean response used for missing
  )
```

Display scoreItem results

```
results
```

```
## Call: scoreItems(keys = key, items = spssdata[psitems], totals = TRUE,
##     missing = TRUE, impute = "none")
##
## (Standardized) Alpha:
##         A1
## alpha 0.53
##
## Standard errors of unstandardized Alpha:
##         [,1]
## ASE   0.0094
##
## Standardized Alpha of observed scales:
##               [,1]
## alpha.observed 0.53
##
## Average item correlation:
##           [,1]
## average.r 0.16
##
##  Guttman 6* reliability:
##          [,1]
## Lambda.6 0.52
##
## Signal/Noise based upon av.r :
##             [,1]
## Signal/Noise  1.1
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, alpha on the diagonal
##  corrected correlations above the diagonal:
##
## Note that these are the correlations of the complete scales based on the correlation matrix,
##  not the observed scales based on the raw items.
##      [,1]
## [1,] 0.53
##
##  In order to see the item by scale loadings and frequency counts of the data
##  print with the short option = FALSE
```

Display the structure of the *results*

```
str(results)
```

```
## List of 17
##  $ scores        : num [1:9282, 1] NaN NaN NaN NaN NaN NaN NaN 5 NaN NaN ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "A1"
##  $ missing       : num [1:9282, 1] 6 6 6 6 6 6 6 0 6 6 ...
##  $ alpha         : num [1, 1] 0.527
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "alpha"
##   .. ..$ : chr "A1"
##  $ av.r          : num [1, 1] 0.157
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "average.r"
##   .. ..$ : NULL
##  $ sn            : num [1, 1] 1.11
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "Signal/Noise"
##   .. ..$ : NULL
##  $ n.items       : num 6
##  $ item.cor      : num [1:6, 1] 0.755 0.726 0.445 0.359 0.517 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "PS1A" "PS1B" "PS1C" "PS1D" ...
##   .. ..$ : chr "A1"
##  $ cor           : num [1, 1] 1
##  $ corrected     : num [1, 1] 0.527
##  $ G6            : num [1, 1] 0.523
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "Lambda.6"
##   .. ..$ : NULL
##  $ item.corrected: num [1:6, 1] 0.44 0.521 0.427 0.353 0.498 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "PS1A" "PS1B" "PS1C" "PS1D" ...
##   .. ..$ : chr "A1"
##  $ response.freq : num [1:6, 1:3] 0.09238 0.06173 0.00808 0.00467 0.01233 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "PS1A" "PS1B" "PS1C" "PS1D" ...
##   .. ..$ : chr [1:3] "1" "5" "miss"
##  $ raw           : logi FALSE
##  $ alpha.ob      : num [1, 1] 0.527
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "alpha.observed"
```

```
##   .. ..$ : NULL
##  $ num.ob.item   : num 6
##  $ ase           : num [1, 1] 0.00941
##  $ Call          : language scoreItems(keys = key, items = spssdata[psitems], totals = TRUE, missing = TRUE,      impute = "none")
##  - attr(*, "class")= chr [1:2] "psych" "score.items"
```

Tabulate missing values

```
table(results$missing)
```

```
##
##    0    1    2    6
## 2344    7    2 6929
```

Display the internal consistency reliability coefficient

```
results$alpha
```

```
##              A1
## alpha 0.5271367
```

Save alpha to two decimal places

```
alpha <- round(results$alpha,2)
```

My markdown code

```
The internal consistency reliability coefficient for the sum of the
psychotic experiences scale is 'r alpha'.
```

The internal consistency reliability coefficient for the sum of the psychotic experiences scale is 0.53.

Display the structure and report the item response frequencies

```
str(results$response.freq)
```

```
##  num [1:6, 1:3] 0.09238 0.06173 0.00808 0.00467 0.01233 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:6] "PS1A" "PS1B" "PS1C" "PS1D" ...
##   ..$ : chr [1:3] "1" "5" "miss"
```

```
results$response.freq
```

```
##                1         5      miss
## PS1A 0.092379736 0.9076203 0.7469295
## PS1B 0.061728395 0.9382716 0.7469295
## PS1C 0.008078231 0.9919218 0.7466063
## PS1D 0.004674883 0.9953251 0.7464986
## PS1E 0.012329932 0.9876701 0.7466063
## PS1F 0.015731293 0.9842687 0.7466063
```

Add to working data file

```
spssdata$pscount <- results$score
```

Review of things we accomplished today

- Big ideas
- Packages
- Commands and functions
- Skills and tasks

Big ideas

- using public data
- tidy data
- documentation
- seeds

Packages

- lubridate
- gmodels (CrossTab)
- sjPlot (sji.SPSS and others)
- psych

Commands and functions

- setwd, getwd
- CrossTable (gModels)
- sample
- runif
- names
- tolower
- as.numeric, as.character
- head
- str
- file.path
- paste0
- scoreItems (psych)

Skills and tasks

- install packages
- load packages
- set the working directory
- relative paths
- read data from the internet
- read data from SPSS/SAV file
- refer to specific variables
- refer to specific observations
- made a cross tab
- made new variables
- generate random data
- define a new function
- working with dates

Next meeting

Analysis with R

Use R to do multivariable analysis

- Linear regression
- Logistic regression
- Survival analysis
- Repeated measures mixed models

Monday, December 8, 2014, 2-3pm.