LearnR (A Guide for Clinician Scientists):Initial Data Exploration

Alvin D. Jeffery, PhD, RN

Last Updated 2017-10-08

Contents

Demo 1: Exploring Your Data

Using Built-In R Functions

Let's start by using built-in R functions before using packages that others have written.

```r
View(support) # looks most like SPSS/Stata viewers
class(support) # figuring out what we're working with
colnames(support)
head(support)
nrow(support)

mean(support$ph) # some descriptive statistics
?mean
mean(support$ph, na.rm=TRUE)
median(support$ph, na.rm=TRUE)

hist(support$ph)
boxplot(support$ph)

summary(support$ph) # more information
sd(support$ph, na.rm=TRUE)

# Troubleshooting Tidbit: know your objects' structure
str(support$ph)

table(support$race) # non-numeric exploration

plot(support$race, support$ph) # the variables we've seen so far
plot(support$age, support$ph) # using 2 continuous variables

cor(support$age, support$ph)
?cor
cor(support$age, support$ph, use='complete.obs')
hist(support$ph) # remember, it's skewed, so what kind of correlation is best?
?cor
cor(support$age, support$ph, use='complete.obs', method='pearson') # default
cor(support$age, support$ph, use='complete.obs', method='spearman') # more appropriate

# consider exploring the stats package in the Help tab
```

Demo 2: Getting Your Data into R & (cont'd) Exploration

Introduction to Object-Oriented Programming

```r
# function(object) - this will yield a value
# object <- function(object) - this will store value in a new object

rnorm(n=10, mean=0, sd=2) # produce some random data
random_data <- rnorm(n=10, mean=0, sd=2) # store data in object
random_data
blah <- rnorm(n=10, mean=0, sd=2)  # you can name it anything you want

summary(blah)
blah_descr <- summary(blah)
blah_descr[4] # mean
```

Read Data with Built-In R Functions

```r
mydata <- read.csv('/Users/AlvinMBA/Desktop/learnr/support.csv')

?read.table # not as many good defaults here for a csv file
mydata <- read.table(file = '/Users/AlvinMBA/Desktop/learnr/support.csv', # file name
                 header = TRUE, # the first row is a header
                 sep = ',') # CSV file, but also have... '' | '\t' | ','
```

Read Data Using Packages

Packages are user-developed functions (typically from built-in R functions) that can be downloaded for free.

```r
library(foreign)
?read.dta # stata
?read.spss # spss (use to.data.frame=TRUE)

library(Hmisc)
?csv.get
mydata <- csv.get('/Users/AlvinMBA/Desktop/learnr/support.csv', # file name
              lowernames = TRUE, # convert variable names to lower case
              charfactor = TRUE) # factor conversion if n/2 unique values

describe(mydata$ph) # even more information (from Frank's package)
describe(mydata$race)

describe(mydata)
```

Demo 3: More Data Exploration

Using Matrix & Computer Science Syntax (without packages)

```r
support <- mydata
support$age[1]
support$age[2]; support$age[3] # multiple commands/line

support[2, 1] # specific location of matrix (note that headings don't count)
support[2:3, 1]

support[1:2, 3] # gender
support$age[1:2]

head(support) # like we did above
support[1:6, ] # should provide same results

head(support[,1:3])
support[1:6, 1:3]

# What if you want to set criteria as opposed to specific rows/columns?
support[which(support$age<22), 1:3]

# But this gets a little complicated, especially if you wanted to describe gender, e.g...
table(support[which(support$age<22), 3])

# instead, let's break this up into steps
young_patients <- support # simply create a new object (for teaching purposes)
young_patients <- subset(support, age < 22)
summary(young_patients$age) # confirm it worked
summary(young_patients$ph) # just as an example
```