Using `uwIntroStats`

Authors: Brian D. Williamson and Scott S. Emerson, M.D., Ph.D.
University of Washington Department of Biostatistics

Contents

1 Introduction

Each statistics and biostatistics department around the world, in conjunction with their collaborators (for us at the University of Washington this includes departments like Epidemiology and Global Health), must choose a statistical software to teach in introductory courses. At the University of Washington, STATA has been taught for many years. While STATA is a powerful software, with relatively user-friendly syntax, it is not very flexible. Only one data set can be read in at a time, and there is only one data type. Simulations are especially difficult to run. While STATA is an important tool to know, we argue that learning the fundamentals of R opens up many possibilities and a lot of power.

We have written another document, "An Introduction to R", hosted at http://www.emersonstatistics.com/GeneralMaterials/R/IntroToR.pdf, which serves as an introduction to the R philosophy of programming and lays out some of the basic data manipulation strategies. In this document, we assume that the reader has read "An Introduction to R" or is at least familiar with the basic R data types, data manipulation, basic functions for descriptive statistics, installing and loading packages. We now present a detailed walkthrough of functions in R that perform the same task as functions in STATA, with syntax provided.

Many of these functions are available in the base R package, which is automatically installed when R is installed and automatically loaded each time you boot up R. However, some functions are only available in other packages which you have to install and load manually. In particular, one of the goals of the `uwIntroStats` package is to facilitate easy adoption of R for STATA users. We leave introduction of this package to a separate document entitled "Using the `uwIntroStats` Package".

2 Descriptive Statistics

Quick view of functions:

STATA	R (base)
`summarize`	`summary(), mean(), sd(), var(), min(), max()`

In STATA, the `summarize` command calculates the number of observations, mean, standard deviation, minimum, and maximum value for each variable in the data set. For example, if we use the `mri` data set from http://www.emersonstatistics.com/datasets/mri.txt, then we can run (showing only the first five variables):

```
summarize
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        ptid |       735         368    212.3205         1        735
     mridate |       735    76422.93    31896.42     10192     123191
         age |       735    74.56599    5.451364        65         99
        male |       735    .4979592    .5003363         0          1
        race |       735    1.318367    .6659304         1          4
-------------+--------------------------------------------------------
```

This conveniently displays all of these summary statistics with one command. In R, this is a bit harder. To find the mean or the median, we simply type `mean(data)` or `median(data)`. Many of the other simple functions (`sum` - compute a sum, `dim` - return the dimensions of an object) operate in a similar way.

So on the `mri` data, from the `uwIntroStats` package, we would see

```
> library(uwIntroStats)
> data(mri)
> mean(mri$age)
[1] 74.56599
> median(mri$age)
[1] 74
> min(mri$age)
[1] 65
> max(mri$age)
[1] 99
> sd(mri$age)
[1] 5.451364
> length(mri$age)
[1] 735
```

These give us flexible options in case we ever need only a subset of these summary statistics. If we wanted to apply one of these functions to the entire data, we could use the `apply()` function:

```
> apply(mri, 2, mean, na.rm = TRUE)
        ptid      mridate          age         male         race       weight       height      packyrs      yrsquit        alcoh      ph
3.680000e+02 7.642293e+04 7.456599e+01 4.979592e-01 1.318367e+00 1.599499e+02 1.657774e+02 1.960048e+01 9.661224e+00 2.109365e+00 1.92233
```

The last argument, `na.rm = TRUE`, is passed to the `mean()` function and makes sure that missing values are removed from the variable when attempting to calculate the mean. Otherwise, the function will return `NA`.