# Introduction to R

Updated by Jinyoung Kim

Center for Social Science Computation & Research
145 Savery Hall
University of Washington
Seattle  WA 98195 U.S.A.
(206)543-8110

August 2006
http://julius.csscr.washington.edu/pdf/r.pdf

# WHAT IS R?

*R* is a system for statistical computation and graphics. It is free software distributed under a GNU-style copyleft, and an official part of the GNU project (“GNU S”). R was initially written by *R*oss Ihaka and *R*obert Gentleman at the Department of Statistics of the University of Auckland in Auckland, New Zealand. In addition, a large group of individuals has contributed to R by sending codes and bug reports. Thanks to these contributors, additional modules (“add-on packages”) are available for a variety of specific purposes; R supports analyses with linear and generalized linear models, nonlinear regression models, time series models, classical parametric and nonparametric tests, clustering and smoothing and simulations, etc.

## Getting and installing

Sources, binaries and documentation for R can be obtained via CRAN, the “Comprehensive R Archive Network”. The CRAN master site at TU Wien, Austria, can be found at the URL
http://cran.R-project.org/

## R tutorial

Even though R does not provide any built-in tutorials like other statistical packages such as SPSS and STATA,  you can go through some demonstrations to start with.

```
> demo(package = .packages(all.available = TRUE))
```

You can see what packages the current version of R has.

```
> demo(lm.glm, package="stats")
```

You can see how to do ordinary least square regression using R.

```
> demo(tkttest,  package=“tcltk”)
```

#        you can see how to do a t-test using R.

## Some Basics

```
>
```

This is the command prompt. This is where we type command to be processed by R. This happens when we hit enter.

```
>q()
```

If you want to close R, type `“q()”`

```
>?plot
>help.search(“logit”)
```

Whenever, you come across what you are not familiar with, try `‘? plot’` or `‘help.search(logit)’`

## Basic Arithmetic Examples

```
>2+3
>2^10
```

This is $2^{10}$

```
>sqrt(2)
```

This is $\sqrt{2}$

```
>exp(2)
```

This is $e^{10}$

```
>log(10)
```

This is $\log_e 10$

```
>log(10, base=10)
```

This is $\log_{10} 10$

```
>log(2)>log(3)
```

True or False will be given as an answer

```
>pi
```

R is smart enough to recognize $\pi$

## Some Vector Operations

```
>x<-c(1,3,2,4)
```

’<-` is an assignment operator. This is assigning a vector consisting of (1,2,3,4) x as its name.

```
>y<-c(106,102,104,100)
>x*y
```

Multiplying two vectors

```
>x+y
```

Adding two vectors

```
>c(x,y)
```

Combining two vectors

```
>z<-c(x,y)
```

Assigning z as its name

```
> sum(x);sum(y);sum(z)
```

Sum of each vector components

```
> sort(x) ;sort(z)
```

Sorting by the components’size

```
>min(z);max(z)
```

minimum and maximum of the vector z

```
>diff(z)
```

First difference of components of z

```
>rev(z)
```

Reversing the order

```
>rep(3,5)
```

The result is 3,3,3,3,3

```
>rep(c(1,2,3),5)
```

Repeating a vector of (1,2,3) by five times

## Some Data Operations

```
>names<-c("homer","margie","lisa","bart","maggie")
>weight<-c(250,140,100,140,70)
>height<-c(180,170,140,160,85)
>sex<-c("m","f","f","m","f")
```

There are 5 observations with three variables(weight, height and sex).

```
>simpsons<-data.frame(names, sex, weight, height)
```

Putting all the variables in a data frame.

```
>simpsons$sex
>simpsons$weight
```

Calling each variable

## Regression and Plotting the Result

```
>attach(simpsons)
```

By attaching a dataframe, you no longer have to tell R where the variables come from. You can drop `‘simpsons$’` in front of the variable name.

```
>fit1<-lm(weight~height)
```

Simple linear regression

```
>hist(weight)
>hist(height)
```

Histogram of each variable

```
>plot(weight,height)
```

Plotting bivariate relationship

```
>abline(fit1)
```

#adding the regression line to the plot

```
>identify(weight,height,names)
```

Identify each data point in the plot by its name

```
>plot(weight,height)
>abline(fit1)
>identify(weight,height,sex)
```

Identify each data point in the plot by its sex

## Logistic Regression

```
>library(car)
```

To do a logistic regression, you need car package

```
>sex1<-recode(sex, "'f'=0; else=1")
```

Recode sex variable

```
>fit2<-glm(sex~weight,family=binomial)
```

Use a generalized linear model, to do logistic regression.

## More Resources

For more documentation, go to http://stat.ethz.ch/R-manual/.