Introduction to R

Peter Schmidt

Advanced Econometrics: Microeconometrics from a Semiparametric Perspective

Why should we use R?

Reasons for using R

- It's freeware.
- There are numerous packages.

Why should we use R?

Reasons for using R

- It's freeware.
- There are numerous packages.

Reasons for not using R

- Matrix operations are not so intuitively as in other statistical packages.
- Maybe you **want** to spend money.

Assignments, Vectors & calculation

For assignment purposes we use the operator <-:

```
> x <- 4
> x
[1] 4
```

Vectors can be constructed using the c(...) command:

```
> vec <- c(1,3,4,5,10,100)
> vec
[1]   1   3   4   5  10 100
```

Using operators like *,/,^ means that the operations are carried out elementwise. For instance:

```
> vec2 <- c(1,3,2,2.5,2,20)
> vec/vec2
[1] 1 1 2 2 5 5
```

Other useful Vector commands

The *Sequence* Command:

```
> seq(1,10)
[1]  1  2  3  4  5  6  7  8  9 10
> 1:10
[1]  1  2  3  4  5  6  7  8  9 10
```

Or with a different step size:

```
> seq(1,19,2)
[1]  1  3  5  7  9 11 13 15 17 19
```

Also useful is the *Replicate* Command:

```
> vec <- c(1,2,3)
> rep(vec,3)
[1] 1 2 3 1 2 3 1 2 3
> rep(vec,1:3)
[1] 1 2 2 3 3 3
```

The usual statistical calculations

If we want to calculate the Mean, the Variance or something else, R provides very intuitive commands.

Suppose we want to calculate the mean of a Standard Normal distributed Variable. By drawing 1000 realizations of a SND variable using `rnorm(1000)`, we get:

```
> mean(rnorm(1000))
[1] -0.05590491
```

The same applies for the variance, standard deviation and other statistical calculations.

```
> var(rnorm(1000))
[1] 1.029674
> sd(rnorm(1000))
[1] 0.9776675
```

Some useful things

The commands are *case sensitive*. For example `t` is the transpose of a matrix and `T` stand for the logical expression `"TRUE"`.

The variables in the workspace can be shown with `ls()`.
The Workspace can be cleared using the command `rm(list=ls(all=TRUE))`.

Graphical Elements

Define:

```
> x = round(runif(100,0,200))
> y <- 100 + 1.5*x + sqrt(1000)*rnorm(100)
```

Then we can visualize this relationship with:

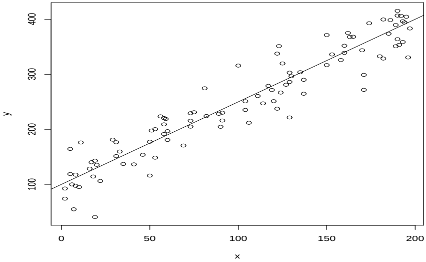
```
> plot(x,y)
```

Additionaly we can add lines by using:

```
> abline(100,1.5)
```

Graphical Elements

Figure: Scatter-Plot



Graphical Elements

Figure: