Business Statistics

TA session -
R programming

Shuo-Chieh Huang

CHICAGO BOOTH

Topics

- **R basics**
  - Vectors, matrix, for-loops.
  - Statistics: mean, sd, rnorm, etc.
- **The hint script**
  - Loading data.
  - Sanity checks.
  - Calculate mean and sd from data.
  - Efficient frontier and Monte-Carlo simulation.

What is R?

- Equivalent to a calculator, but has a huge number of users developing tools on it (for free).
- As a calculator, it…
  - is dumb. You need to be *grammatically* correct/precise for it to understand.
  - reads top-down.
- As a toolbox, it…
  - has many off-the-shelf commands, for a variety of applications.
  - has a great deal of resource online.

R basics

- First things first, set working directory. Use the command `setwd( )`. It tells R where is the default place to find your data. To get current directory, use `getwd( )`.

- Vector: a sequence of numbers. Like

$$x = (1, 4, 7, -1, 0.3)$$

- Matrix: an array of numbers. Like

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 6 \\ -1 & 0.5 & -0.1 \end{pmatrix}$$

- In R, you can create a vector by

```
x = c( 1, -1, 3, 0.5, 6, -0.1 )
```

- and a matrix by

```
X = matrix( c(1, -1, 3, 0.5, 6, -0.1), ncol=3, nrow=2 )
```

- When you load data into R, it is most likely in *data.frame* or *matrix*, a useful sanity check is to print a first few rows to see if the variable names, index numbers/dates, are aligned as desired. You can do this by

```
head( )
```

- To subset a vector, or a matrix, use square brackets. For example,

```
> x = c( 1, 1.3, -1.5, 0 )
> x[ c(2,4) ]
[1] 1.3 0.0
```

- To generate a vector of numbers equally spaced from some number *a* to another number *b*, we can use the command `seq( from=a, to=b, length.out=n )` (you need to specify a,b,n). For example,

```
> seq( from=0, to=1, length.out=11 )
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

R basics - statistics

- Some useful functions in R to compute statistics from data.
  - mean: `mean( x )`
  - standard deviation: `sd( x )`
  - covariance and correlation: `cov( x, y )`, `cor( x, y )`
  - generate n random numbers from normal distribution with mean m and sd s: `rnorm( n, mean=m, sd=s )`
- Letters marked in red above are the arguments you need to supply R.

R basics - for-loop

- In Monte-Carlo simulations we need to repeat the same experiment hundreds or even thousands of times. R has a flow control feature to automate the process — the for-loop.

- Example. I want to know the first 10 multiples of 3.14.

```
for( i in 1:10 ){
  print( i * 3.14 )
}
```

```
> for( i in 1:10 ){
+   print( i * 3.14 )
+ }
[1] 3.14
[1] 6.28
[1] 9.42
[1] 12.56
[1] 15.7
[1] 18.84
[1] 21.98
[1] 25.12
[1] 28.26
[1] 31.4
```

Troubleshooting

- Use ? to see R help page. For example, `?rnorm`

- Google your error message. There is a fair chance someone is asking the same question.