Why R

- Programming and Statistical Language
  - Easy statistical software
  - A programming language for other purposes

- Data Analysis and Visualization
  - Apart from doing data analysis
  - R is a very capable data visualization tool

Install R

- Go to https://cran.r-project.org/
- Chose your OS
- Download
- Install

Install R Studio

- Why?
  - An IDE (integrated development environment) makes is easer to use and program
  - Debug

- **Installers for Supported Platforms**

Installers	Size	Date	MD5
RStudio 1.0.153 - Windows Vista/7/8/10	81.9 MB	2017-07-20	b3b4bbc82865ab105c21cb70b17271b3
RStudio 1.0.153 - Mac OS X 10.6+ (64-bit)	71.2 MB	2017-07-20	8773610566b74ec3e1a88b2fdb10c8b5
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	85.5 MB	2017-07-20	981be44f91fc07e5f69f52330da32659
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	91.7 MB	2017-07-20	2d0769bea2bf6041511d6901a1cf69c3
RStudio 1.0.153 - Ubuntu 16.04+/Debian 9+ (64-bit)	61.9 MB	2017-07-20	d584cbab01041777a15d62cbef69a976
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2017-07-20	8dfee96059b05a063c49b705eca0ceb4
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2017-07-20	16c2c8334f961c65d9bfa8fb813ad7e7

- **Zip/Tarballs**

Zip/tar archives	Size	Date	MD5
RStudio 1.0.153 - Windows Vista/7/8/10	117.6 MB	2017-07-20	024b5714fa6ef337fe0c6f5e2894cbcb
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	86.2 MB	2017-07-20	f8e0ffa7ec62665524f9e2477facd346
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	92.7 MB	2017-07-20	2077c181311d1aad6fb8d435f8f1f45f
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	85.4 MB	2017-07-20	92e1a22d14952273ec389e5a55be614f
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	86.6 MB	2017-07-20	0b71c5a7fc53c84b3fe67242240b3531

Install R

- Schedule
  - Value assignment
  - Data operation
  - Flow Control

Value Assignment

- Value assignment is the most basic operation
  - a = 25
  - b <- “How are you”
  - Use quotations when this is a text string
  - Try b <- how are you instead
  - FALSE -> c
  - Or F->d
  - False is unacceptable. Beware

To run a line of code: Ctrl + Enter
To run multiple line of code: Highlight the lines and Ctrl + Enter
To change line: Enter
There is no need to define a data type beforehand

Operators

- Arithmetic operators
  - d = 5
  - a + d
  - a
  - 5 %% 2 #remainder of division
  - 2 ^ 3 # power function

- Relational operators
  - 2 > 3
  - 2 < 3
  - 2 == 2
  - 2 != 2

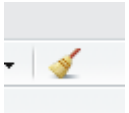
Operators

- Logic operator
  - (3 ==2) & (3 == 3) # and operator
  - X = TRUE
  - Y = FALSE
  - x & x
  - x & y
  - y & y
  - (3 ==2) | (3 == 3) # or operator
  - x | x
  - x | y
  - y | y

Operators

- Generate a series of numbers
  - X = 2:8 # x= [2,3,4,5,6,7,8]
  - Y=6
  - Y %in% x # check if 6 is in the list

Clear Things

- Ctrl + l to clear the console
-  is to clear the variables

Vectors

- A series of numbers or other types of data
  - Vtr = c(1,3,5,11,45,23,67,55,41)
  - Sortedvtr=sort(vtr)
  - Vtr[2]
  - Vtr[2:6]
  - Vtr[-1] # delete the first
  - vtr[1] <- 12 # substitute the 1st number to 12
  - vtr[13] <- 12 # NA will be filled
  - Extract the last three element? (hint: the length of the vector is length())

List

- List1 <- list(‘Hey’, FALSE, 54)
- List can have different types of data inside
- List2 <- list(‘Hello’, ‘R’, 45, 28)
- List3 <- merge(list1, list2)
- List3[1]

Matrices

- Most of the data set we will be dealing with is panel data
  - Essentially a matrix
  - matrix(data = xxx, nrow = xxx, ncol = xxx, byrow = TRUE/FALSE, dimnames = xxx)
  - data is a vector input
  - Nrow: specify the number of rows
  - Ncol: specify the number of columns
  - By row or by column
  - Dimname: specify the name of each row/col

Matrices

Try type it
Not just copy
and paste!

Get familiar with
the syntax

- Try the following
  - dat <- matrix(c(1,2,3, 11,12,13), nrow = 2, ncol = 3, byrow = TRUE,
    dimnames = list(c("row1", "row2"),
    c("C.1", "C.2", "C.3")))
  - dat2 <- matrix(c(1,2,3, 11,12,13), nrow = 2, ncol = 3, byrow = FALSE)
  - dat3 <- matrix(c(1,2,3, 11,12,13), 2, 3, FALSE)
  - By default, byrow is FALSR
  - vtr1=c(1,2,3,4,5)
  - vtr2=c(6,7,8,9,10)
  - dat4 = matrix(c(vtr1,vtr2),5,2)
  - dat5 = matrix(c(vtr1,vtr2),5,5), repeat

Flow Controls – If Statement

- If statement
  - X <- 5
  - if (x>3){print ("x is greater than 3")}
  - But usually we write something like this
  - if (x>3){
    print ("x is greater than 3")
    }
  - if (x>3){
    cat (x," is greater than 3")
    }

  Cat is the concatenate function

Flow Controls – If Statement

- If statement
  - X <- 5
  - if (x>3){
    cat (x, “ is greater than 3")
    } else {
    cat (x, " is smaller than 3")
    }

Flow Controls – If Statement

- If statement

```
if (x>6){
 cat (x, " is greater than 3")
} else if (x==5){
 cat (x, “ is equal to 5")
} else {
 cat (x, “ is smaller than 3")
}
```

Flow Controls - Loops

- Using loops to repeat actions

- The repeat loop

```
repeat{
  commands
  if(condition){
    break
  }
}
```

Flow Controls - Loops

- Example

```
x=2
repeat{
  x=x^2
  if(x>100){
    break
  }
}
```

Flow Controls - Loops

- The for loop
  - To calculate the sum in the vector

```
vtr=c(1,2,42,4,5,66)
sum=0
for(i in vtr){
  sum <- sum+i
}
sum
```

Flow Controls - Loops

- The for loop
  - To calculate 1+2+3…..+100
  - Try it yourself

The 10 students got their score (98,86,65,59,91,87,77,77,77,100)
Generate a list or a vector of the letter grades of the 10 students

Try it out

Hint:
First generate an empty list e.g. gradelist=list()
To append an element to a list: gradelist=c(gradelist, “A”)

Similarly, if you chose to use a vector to record the letter grade
First generate an empty vector e.g. gradevector=c()
To append an element to a vector: gradevector[i]<-”A”