Chapter4:
Data structures, Missing data, NA, and CSV file in R Programming

Asst. Prof. Dr. Supakit Nootyaskool
supakit@it.kmitl.ac.th

Learning Objective

- To obtain five data structure in R
- To examine missing data, inf and n/a data.
- To understand access to CSV file and object file.

Topic

- Five data structure in R
- Missing Data, Infinity value, and Not a number
- Reading CSV file
- Writing CSV file
- Get data by condition
- Save or load object

4.1 Five types of data structure

Data structures are used to store data in the organization and doing data manipulation. R has five data structures consist:

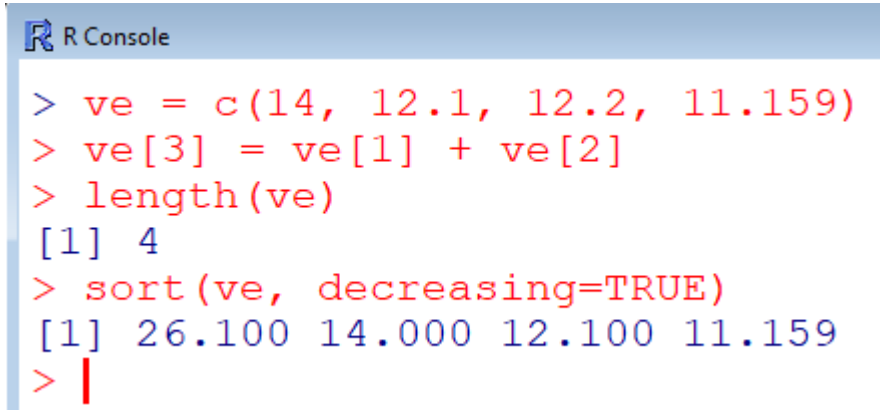
1. Vector
2. List
3. Matrix
4. Data frame
5. Factor

```
ve = c(12.1, 12.2, 14)
li = list(ve, c(1,2,3,4), “Jan”, “Hello”)
ma = matrix(c(1:9), nrow=3, ncol=3, byrow=true)
dfId = c(1:4)
dfName = c(“Justin”, “Donald”, “Mark”, “Adele”)
dfNum = c(225, 3100, 73, 135)
dfType = c(2,1,3,2)
df = data.frame(dfId, dfName, dfNum, dfType)
```

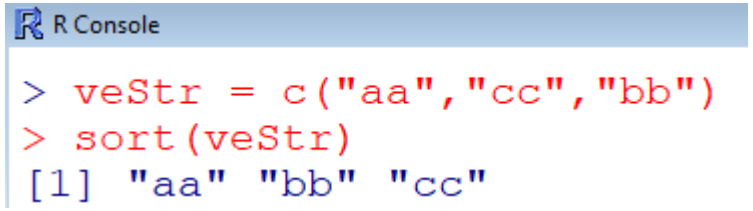
4.1.1 Vector

Vector is homogenous data that contains elements of the same data types by numeric, integer, character, complex or logical

```
ve = c(14, 12.1, 12.2, 11.159)
ve[3] = ve[1] + ve[2]
length(ve)
sort(ve, decreasing=TRUE)
```

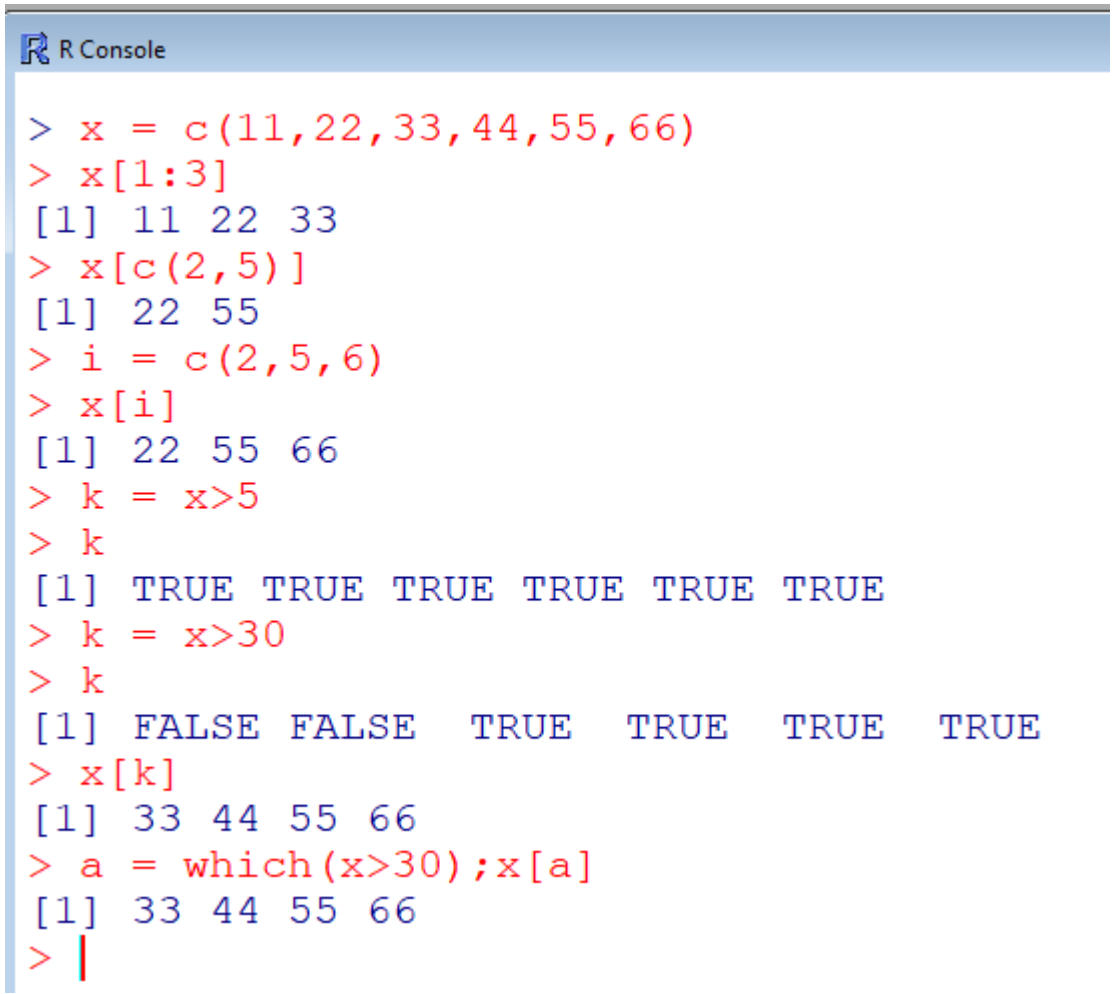



```
veStr = c(“aa”, “cc” “bb”)
sort(veStr)
```



4.1.1 Vector: Indexing, selection

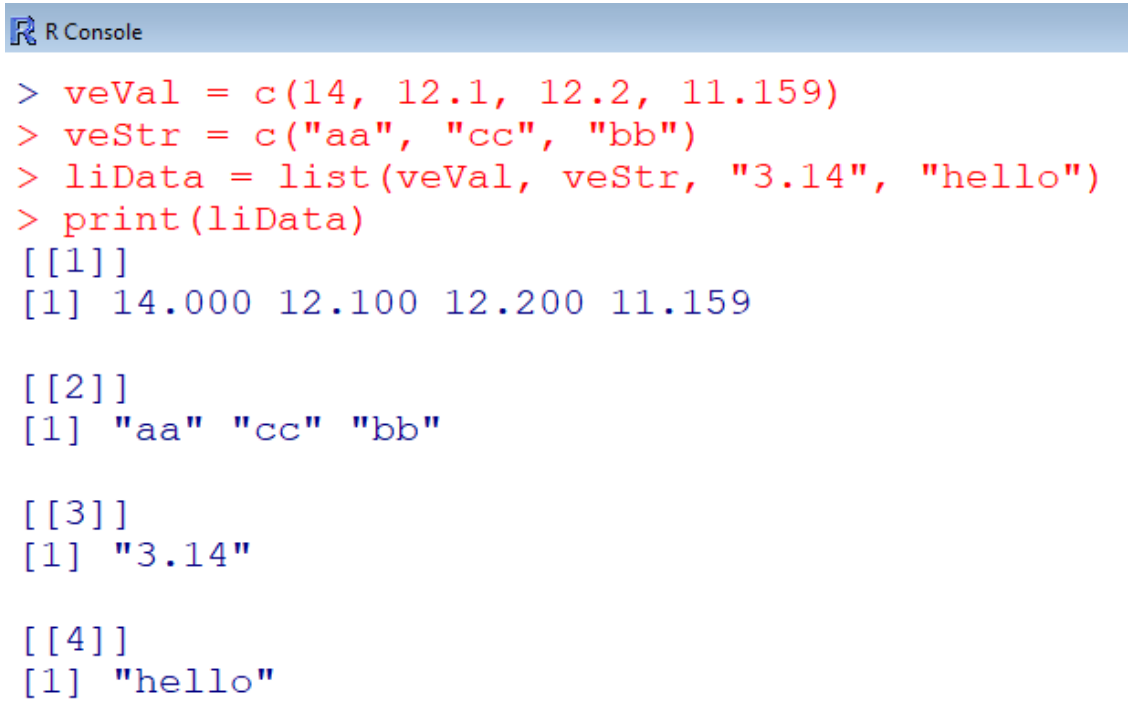
```
x[1]                    #first element of x

x[1:5]                  #Get element 1 to 5
x[c(2,5,6)]             #Get elements 3,5,6
z=c(3,5,6) ; x[z]

x[x>=5]                 #Find elements of x ≥5
k=x>=5 ; x[k]
a=which(x>=5);x[a]
```



4.1.2 List

List is non-homogenous data that can contain different data types.

```
veVal = c(14, 12.1, 12.2, 11.159)
veStr = c(“aa”, “cc”, “bb”)
liData = list(veVal, veStr, “3.14”, “hello”)
print(liData)
```



4.1.3 Matrix

Matrix is 2-dimensional data in form homogenous data.

```
maA = matrix(c(1:9), nrow=3, ncol=3, byrow=TRUE)
print(maA)

maB = matrix(c(1:9), nrow=3, ncol=3, byrow=FALSE)
print(maB)
```

```
> maA = matrix(c(1:9), nrow=3, ncol=3, byrow=TRUE)
> print(maA)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
>
> maB = matrix(c(1:9), nrow=3, ncol=3, byrow=FALSE)
> print(maB)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

4.1.3 Matrix: R/C

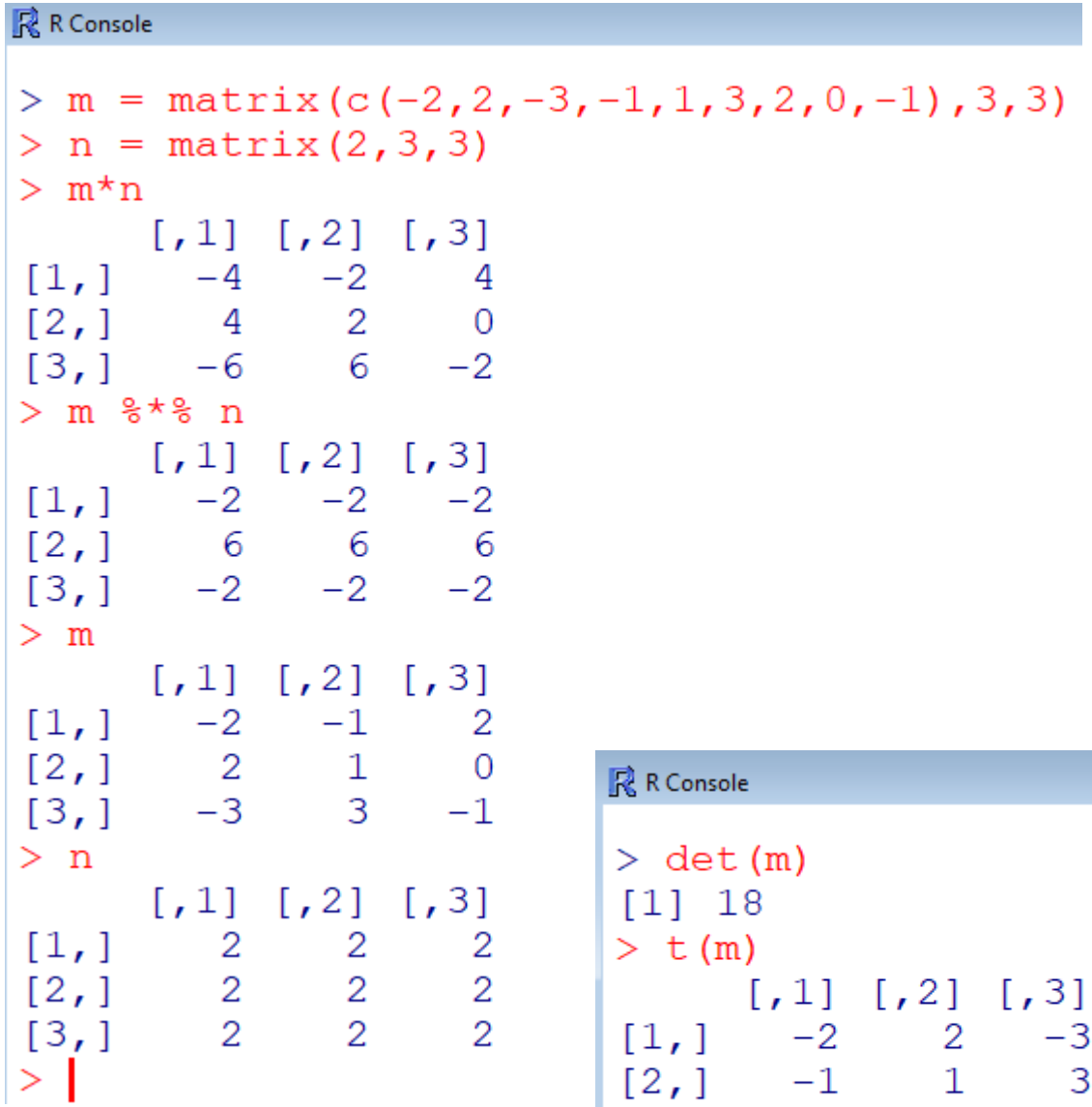
m[r,] #Get by row

m[,c] #Get by column

m1%*%m2 # Production matrix

det(m) #Determinent matrix

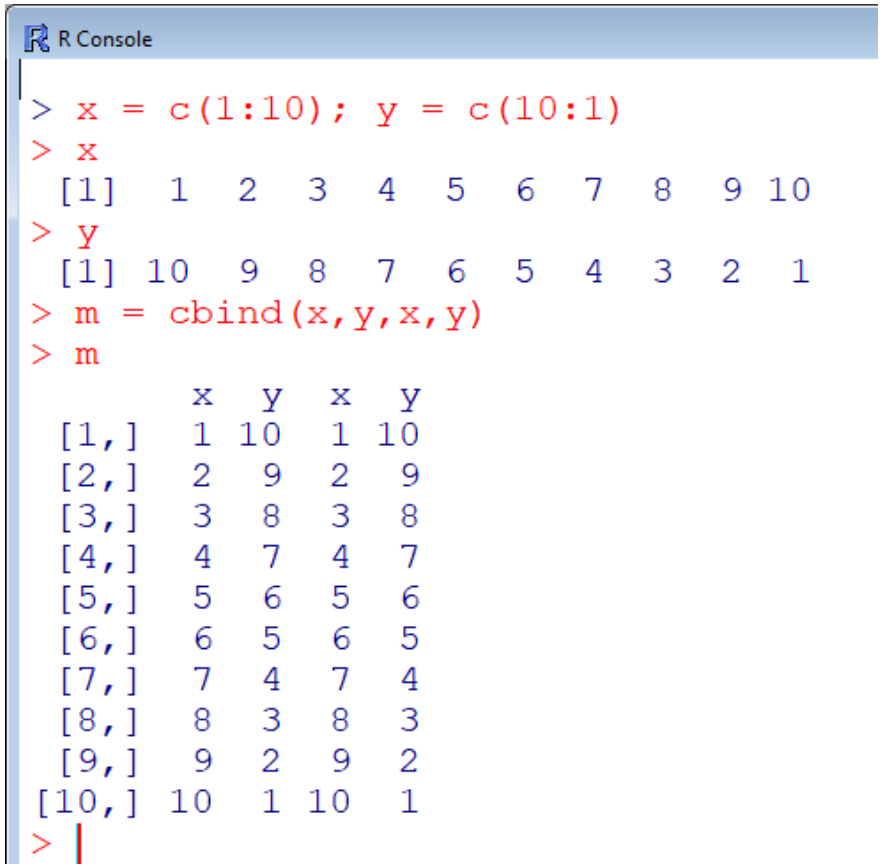
t(m) #Transpose matrix

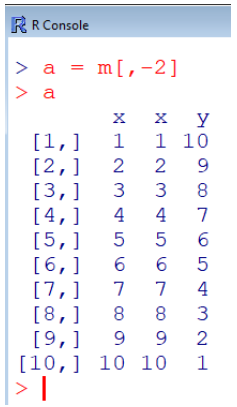


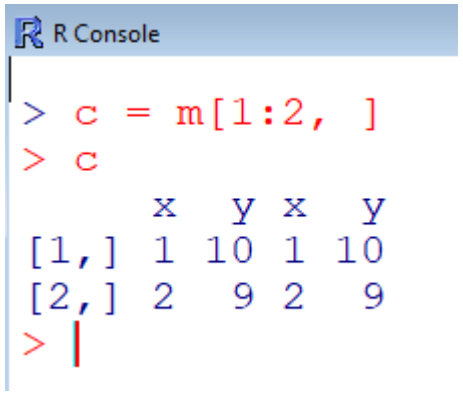
4.1.3 Matrix: Combine, remove,row,column of matrices

```
x = c(1:10); y = c(10:1)      #Generating data
m = cbind(x,y,x,y)            #Combine data
```

```
a = m[    , -2]   #Remove the second column
b = m[ -2 ,   ]   #Remove the second row
a = m[1:2, ]      #Select column 1 to 2
```







List vs Matrix

- Lists are not restricted to a single mode and can encompass any mixture of data types.

- Matrixes are not a separate type of object but simply an atomic vector with dimensions; the number of rows and columns.

```
R Console
> m = matrix(c(11,22,33,44),2,2)
> m
     [,1] [,2]
[1,]   11   33
[2,]   22   44
> l = list("pi",3.14,"eV",1.602e-12)
> l
[[1]]
[1] "pi"

[[2]]
[1] 3.14

[[3]]
[1] "eV"

[[4]]
[1] 1.602e-12
```

```
> is.matrix(l)
[1] FALSE
> is.list(m)
[1] FALSE
```

4.1.4 Data frame

Data frame is 2-dimensional data like the matrix, but it is non-homogenous.

```
dfId = c(1:4)
dfName = c(“Justin”, “Donald”, “Mark”, “Adele”)
dfNum = c(225, 3100, 73, 135)
dfType = c(2,1,3,2)
df = data.frame(dfId, dfName, dfNum, dfType)
summary(df)
```

```
> df = data.frame(dfId, dfName, dfNum, dfType)
>
> df
  dfId dfName dfNum dfType
1    1 Justin   225      2
2    2 Donald  3100      1
3    3   Mark    73      3
4    4  Adele   135      2
> summary(df)
      dfId         dfName              dfNum           dfType
 Min.   :1.00   Length:4           Min.   :  73.0   Min.   :1.00
 1st Qu.:1.75   Class :character   1st Qu.: 119.5   1st Qu.:1.75
 Median :2.50   Mode  :character   Median : 180.0   Median :2.00
 Mean   :2.50                      Mean   : 883.2   Mean   :2.00
 3rd Qu.:3.25                      3rd Qu.: 943.8   3rd Qu.:2.25
 Max.   :4.00                      Max.   :3100.0   Max.   :3.00
```

4.1.5 Factor

Factor is data type to analyze and categorize unique values in columns or vector.

```
brand = c(
"honda",
"honda",
"toyota",
"mini",
"mini",
"saab",
"saab")

factor(brand)
```

```
> brand = c(
+ "honda",
+ "honda",
+ "toyota",
+ "mini",
+ "mini",
+ "saab",
+ "saab")
> factor(brand)
[1] honda  honda  toyota mini   mini   saab   saab
Levels: honda mini saab toyota
> f = factor(brand)
> f
[1] honda  honda  toyota mini   mini   saab   saab
Levels: honda mini saab toyota
```

4.2 Missing Data, Infinity value, and Not a number

R supports missing data in the vector

```
x = c(0.5, NA, 0.7)      #create a vector with NA
is.na(x)                 #show which one NA in the vector
anyNA(x)                 #Is the vector has NA?

y = 1/0                  #y has Inf
is.na(y); is.nan(y)

z = 0/0                  #z has NaN
is.na(z); is.nan(z)
```

4.3.1 Reading CSV file

A function is used to import the CSV file to R variable. Download an example CSV from
http://161.246.38.75/download/babd/chap04/corona.csv

c = read.csv("corona.csv") #get data

summary(c) #calculation summary

head(c) #show head of the data

```
> c = read.csv("corona.csv")
> head(c)
        date    county      state  fips cases deaths
1 2020-01-21 Snohomish Washington 53061     1      0
2 2020-01-22 Snohomish Washington 53061     1      0
3 2020-01-23 Snohomish Washington 53061     1      0
4 2020-01-24      Cook   Illinois 17031     1      0
5 2020-01-24 Snohomish Washington 53061     1      0
6 2020-01-25    Orange California  6059     1      0
> summary(c)
     date              county             state                fips
 Length:975686      Length:975686      Length:975686      Min.   : 1001
 Class :character   Class :character   Class :character   1st Qu.:19003
 Mode  :character   Mode  :character   Mode  :character   Median :29219
                                                          Mean   :31280
                                                          3rd Qu.:46101
                                                          Max.   :78030
                                                          NA's   :9072
     cases              deaths
 Min.   :      0   Min.   :    0.00
 1st Qu.:     45   1st Qu.:    0.00
 Median :    294   Median :    5.00
 Mean   :   2438   Mean   :   60.03
 3rd Qu.:   1259   3rd Qu.:   26.00
 Max.   :1098363   Max.   :26856.00
```

4.3.1 Access in each row/column

```
is.list(c)        #List?
c[1,1]            #Get first row and first column
c[ , 1]           #Get first row
c[1, ]            #Get first column

head(c)
c$state           #Get all data in “state”
c$state[1:1]      #Get 1 to 10 in “state”
```

```
> is.list(c)
[1] TRUE
> c[1,1]
[1] "2020-01-21"
> c[1, ]
        date    county      state  fips cases deaths
1 2020-01-21 Snohomish Washington 53061     1      0
> head(c)
        date    county      state  fips cases deaths
1 2020-01-21 Snohomish Washington 53061     1      0
2 2020-01-22 Snohomish Washington 53061     1      0
3 2020-01-23 Snohomish Washington 53061     1      0
4 2020-01-24      Cook   Illinois 17031     1      0
5 2020-01-24 Snohomish Washington 53061     1      0
6 2020-01-25    Orange California  6059     1      0
> c$state[5:10]
[1] "Washington" "California" "Illinois"   "Washington" "Arizona"
[6] "California"
```

4.3.2 Write CSV file

```
d = data.frame(
        name = c("Jon", "Bill", "Maria"),
        age = c(23,41,32))

print (d)

write.csv(df,
        "data.csv",
        row.names = FALSE)
```

R Console

```
> d = data.frame(
+ name = c("Jon", "Bill", "Maria"),
+             age = c(23,41,32))
>
> print (d)
   name age
1   Jon  23
2  Bill  41
3 Maria  32
>
> write.csv(df,
+ "data.csv",
+ row.names = FALSE)
>
```

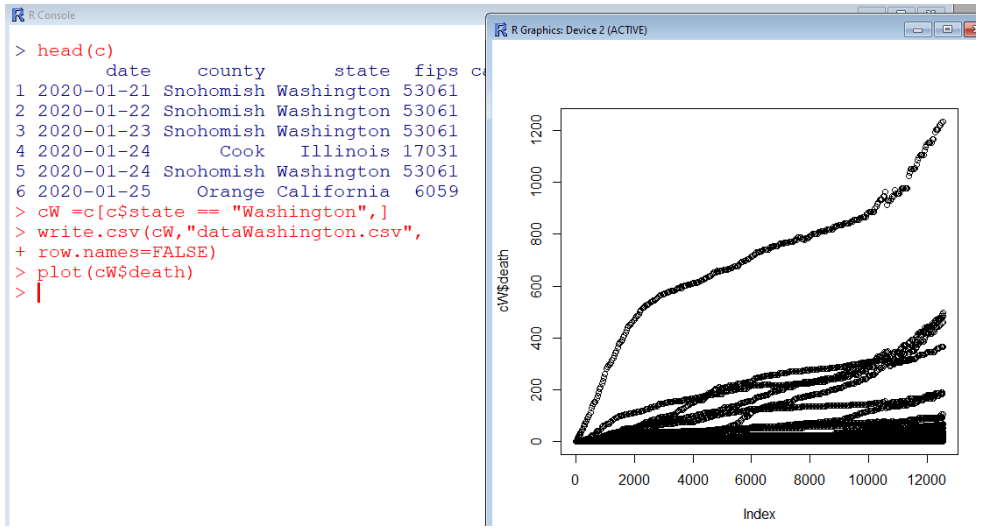
data.csv - Notepad

File Edit Format View Help

```
"dfId","dfName","dfNum","dfType"
1,"Justin",225,2
2,"Donald",3100,1
3,"Mark",73,3
4,"Adele",135,2
```

4.4 Get data by condition

```
head(c)          #Show head
cW = c[c$state == "Washington",]          #Get data by condition
write.csv(cW,
          "dataWashington.csv",
          row.names = FALSE)          #Write to CSV
plot(cW$death)          #Plot
```



4.5 Save / Load Object

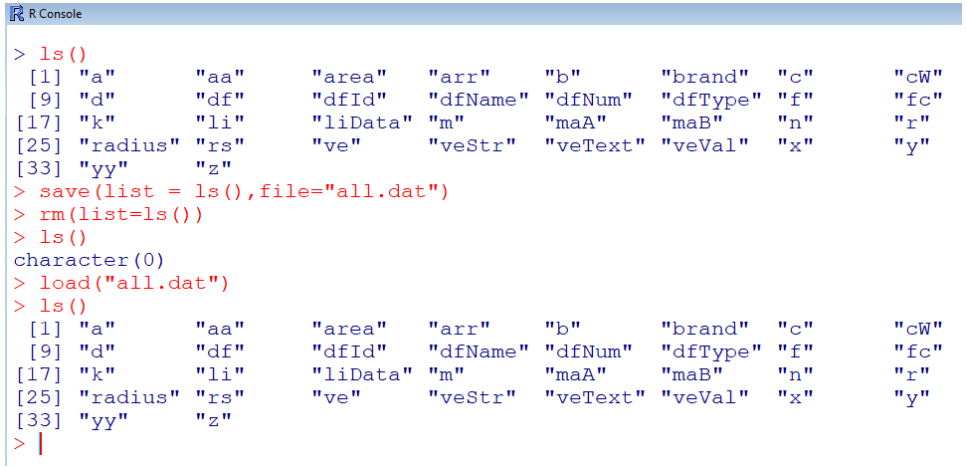
save(cW, file= “cw.dat”) #save an object

save(list = ls(),file="all.dat") #save all object

rm(list=ls()) #clear all memory

dir() #show list file in the directory

load(“all.dat”) #load objects in “all.data”



Quiz