SDSC 3006
Lab 1. Introduction to R

Name: Yiren Liu
Email: yirenliu2-c@my.cityu.edu.hk

School of Data Science
City University of Hong Kong

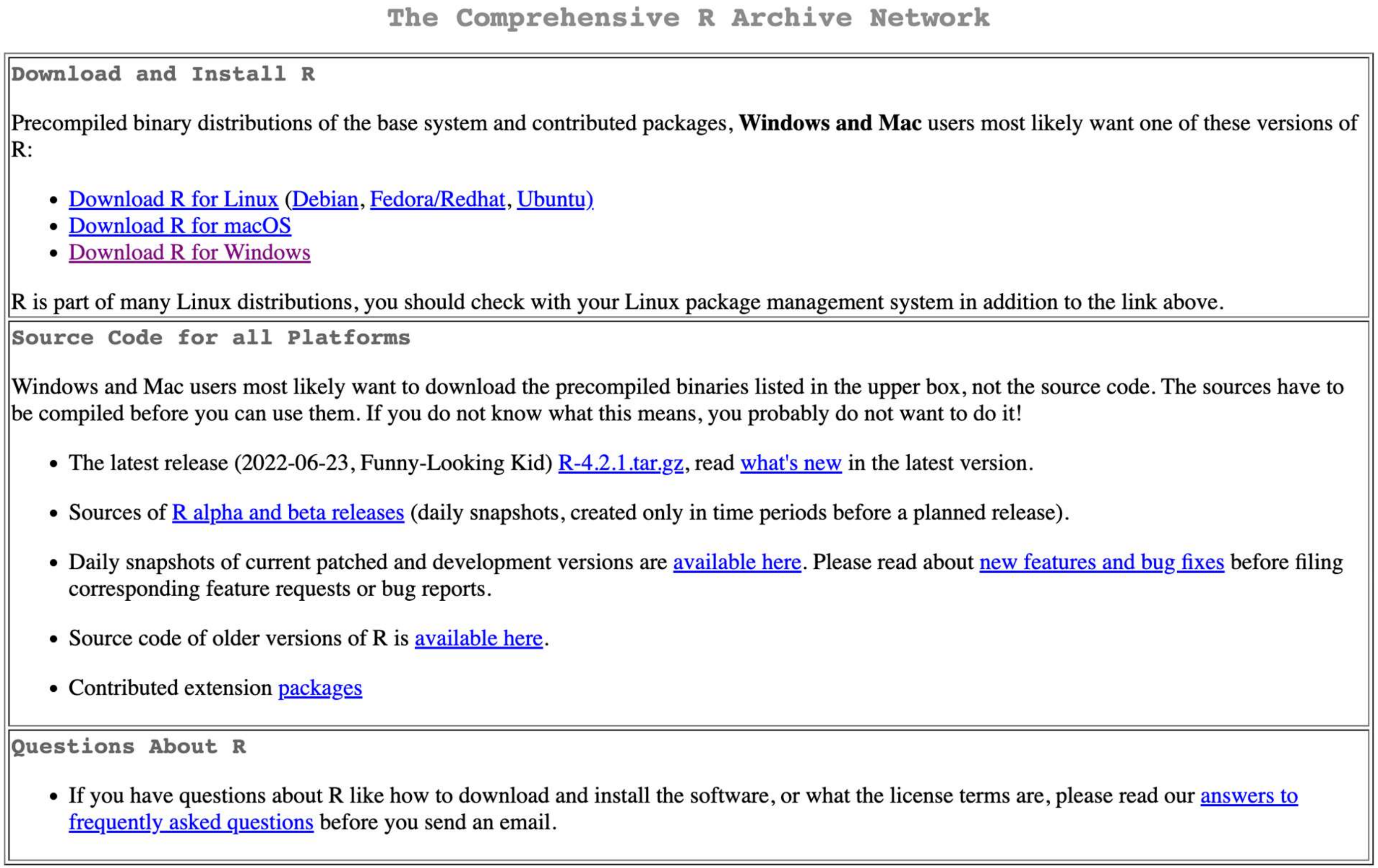
Outline

- Install R, and R Studio as IDE
- Install data packages
- Basic commands
- Example of preliminary analysis of a dataset

Install R, R Studio

Install R

https://cran.r-project.org/



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

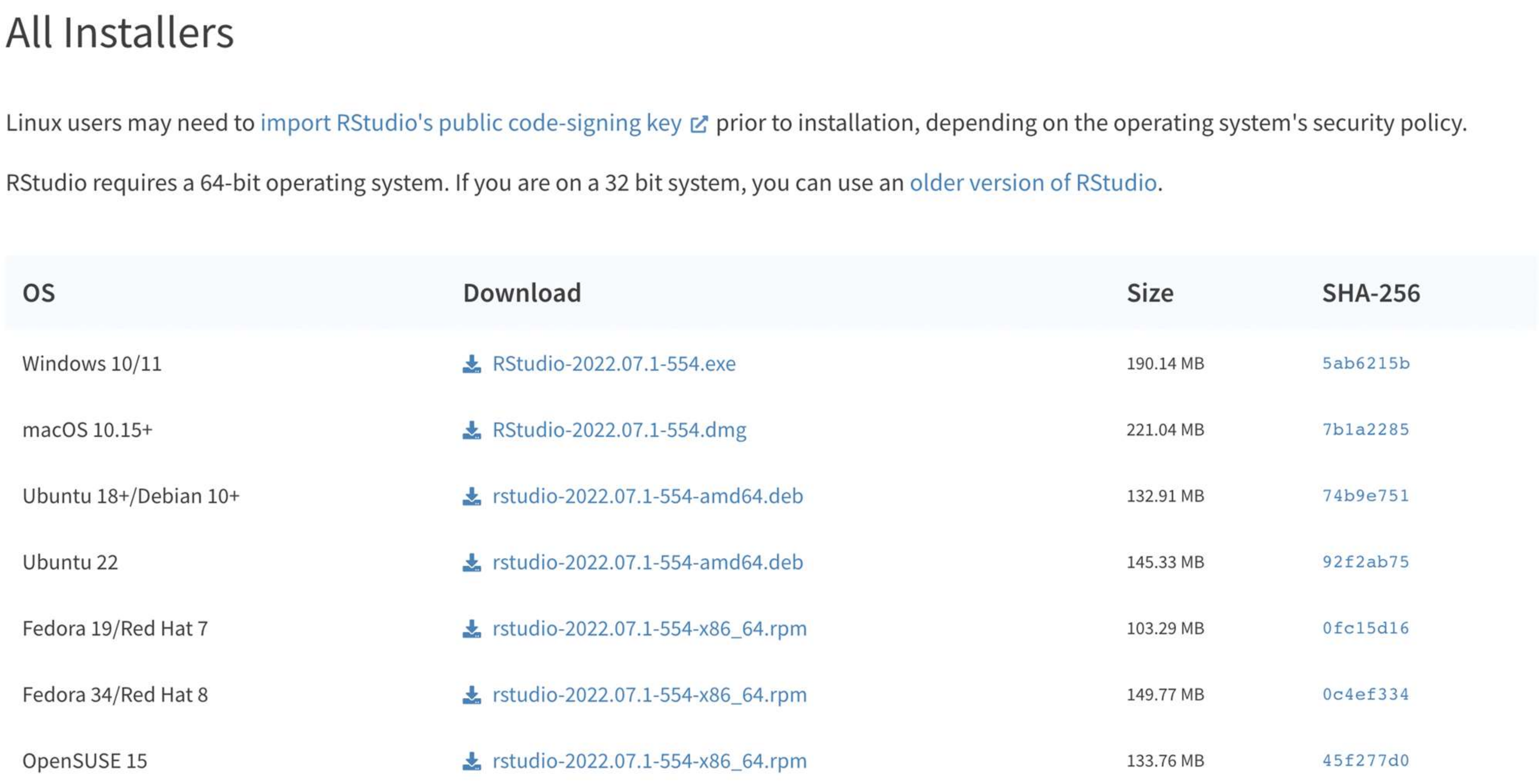
- The latest release (2022-06-23, Funny-Looking Kid) R-4.2.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Install R Studio

https://www.rstudio.com/products/rstudio/download/#download



All Installers

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/11	RStudio-2022.07.1-554.exe	190.14 MB	5ab6215b
macOS 10.15+	RStudio-2022.07.1-554.dmg	221.04 MB	7b1a2285
Ubuntu 18+/Debian 10+	rstudio-2022.07.1-554-amd64.deb	132.91 MB	74b9e751
Ubuntu 22	rstudio-2022.07.1-554-amd64.deb	145.33 MB	92f2ab75
Fedora 19/Red Hat 7	rstudio-2022.07.1-554-x86_64.rpm	103.29 MB	0fc15d16
Fedora 34/Red Hat 8	rstudio-2022.07.1-554-x86_64.rpm	149.77 MB	0c4ef334
OpenSUSE 15	rstudio-2022.07.1-554-x86_64.rpm	133.76 MB	45f277d0

Install R Studio

We need to install both R and R Studio separately.

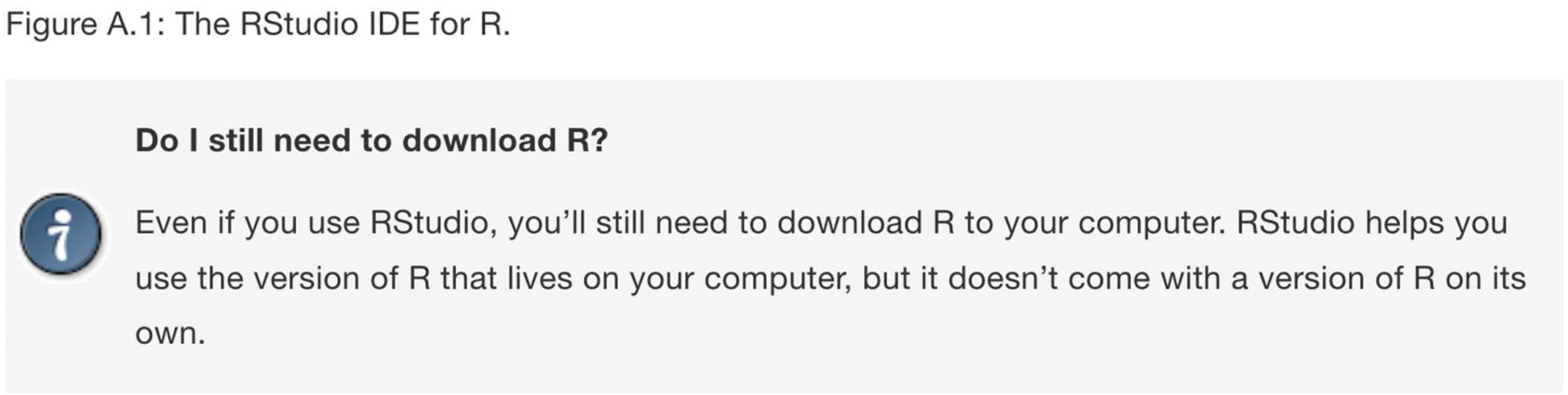


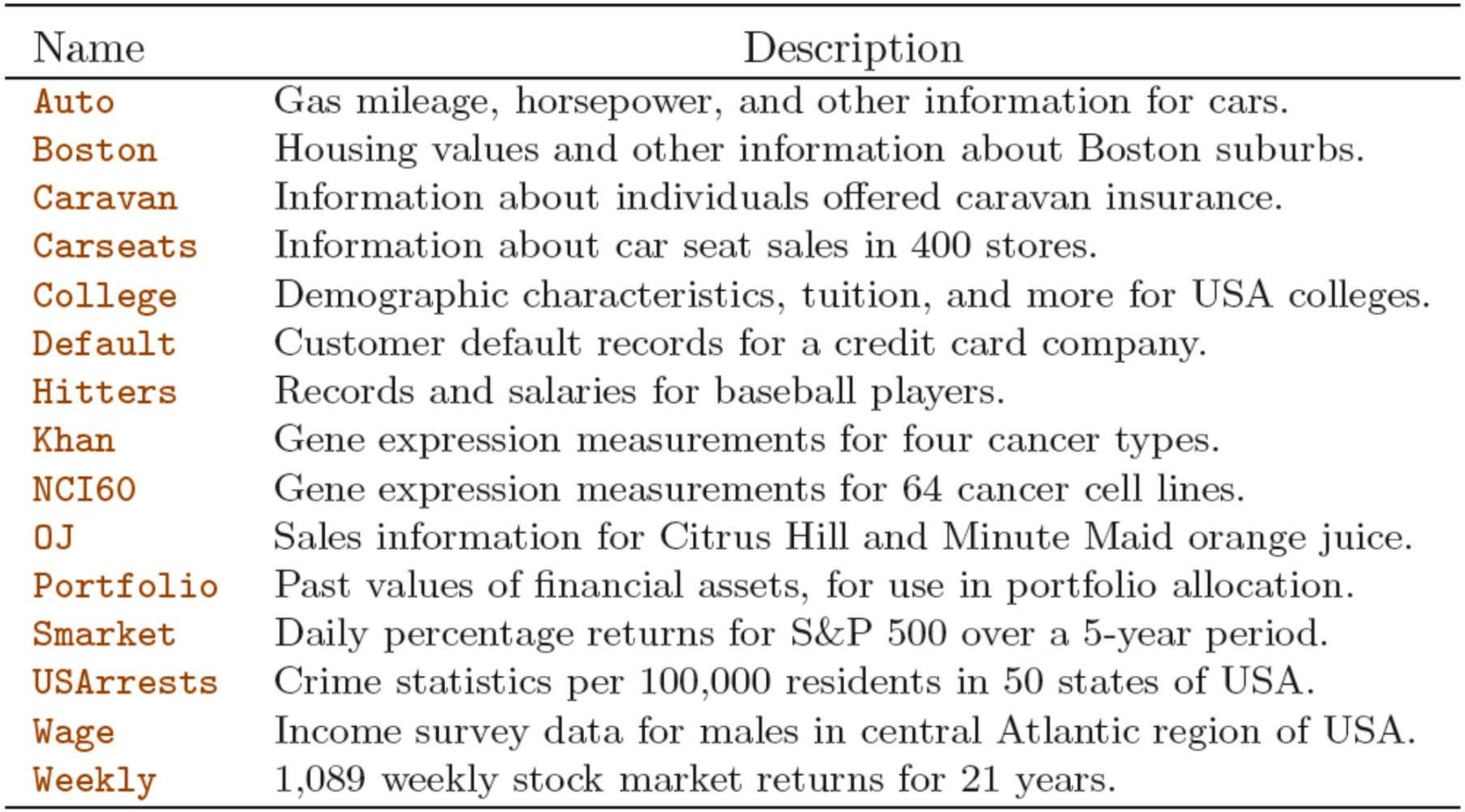
Figure A.1: The RStudio IDE for R.

Do I still need to download R?

Even if you use RStudio, you'll still need to download R to your computer. RStudio helps you use the version of R that lives on your computer, but it doesn't come with a version of R on its own.

Data Sets Used in Labs and Exercises

- The ISLR package



Name	Description
Auto	Gas mileage, horsepower, and other information for cars.
Boston	Housing values and other information about Boston suburbs.
Caravan	Information about individuals offered caravan insurance.
Carseats	Information about car seat sales in 400 stores.
College	Demographic characteristics, tuition, and more for USA colleges.
Default	Customer default records for a credit card company.
Hitters	Records and salaries for baseball players.
Khan	Gene expression measurements for four cancer types.
NCI60	Gene expression measurements for 64 cancer cell lines.
OJ	Sales information for Citrus Hill and Minute Maid orange juice.
Portfolio	Past values of financial assets, for use in portfolio allocation.
Smarket	Daily percentage returns for S&P 500 over a 5-year period.
USArrests	Crime statistics per 100,000 residents in 50 states of USA.
Wage	Income survey data for males in central Atlantic region of USA.
Weekly	1,089 weekly stock market returns for 21 years.

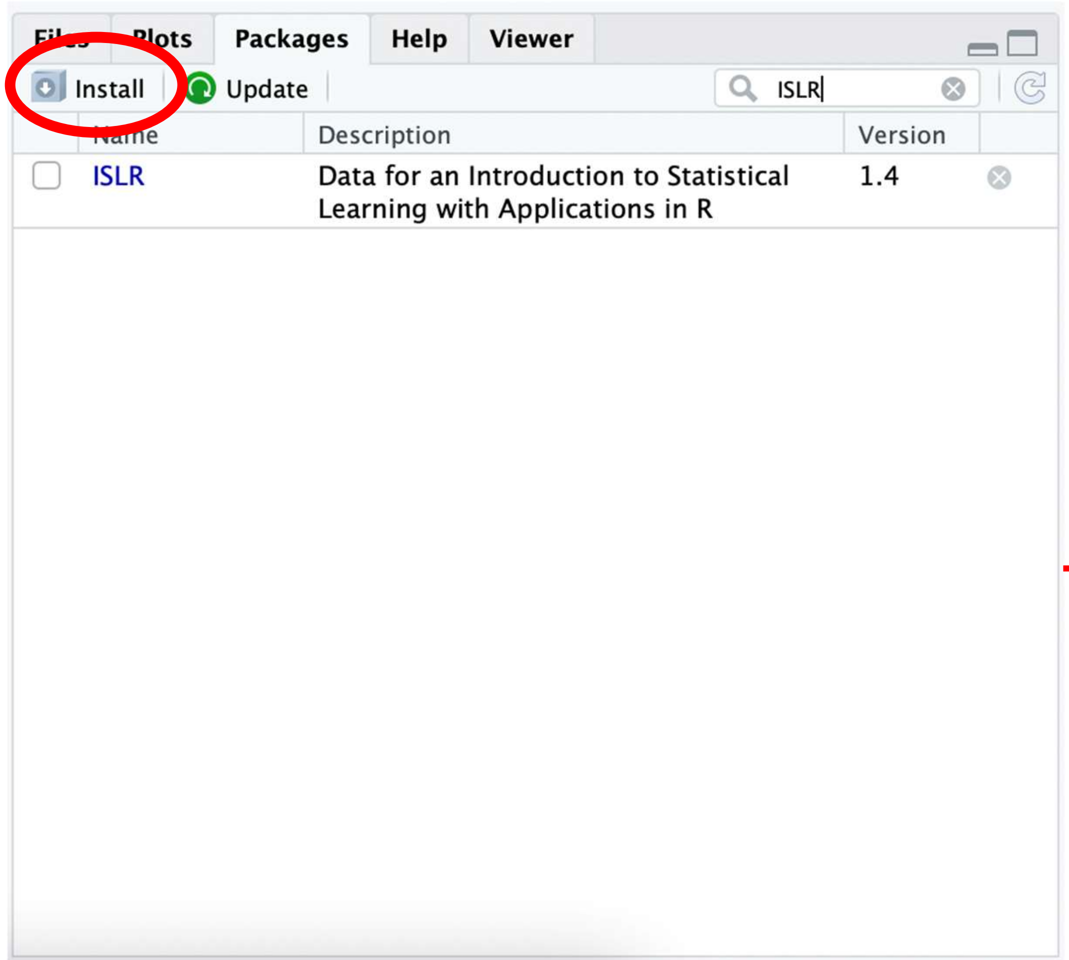
- The MASS library
- Base R

Install ISLR Package

Method1:

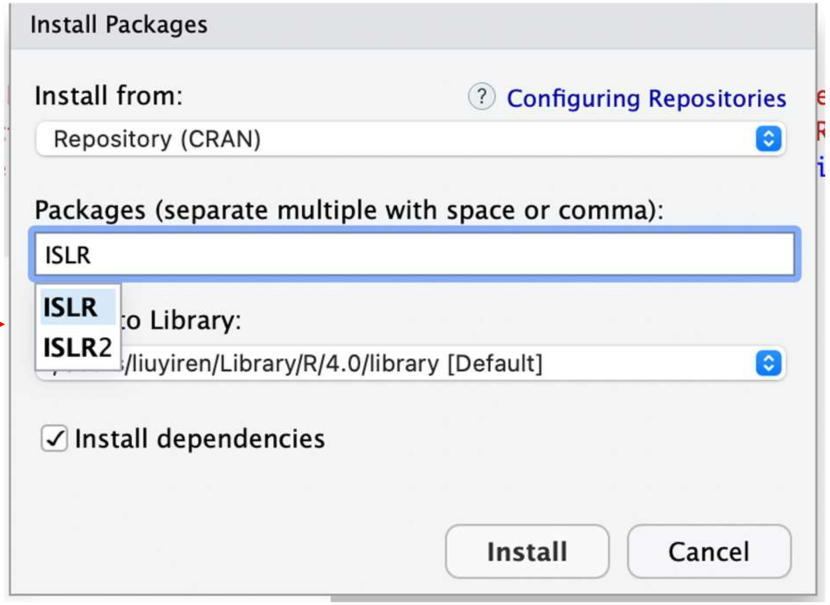
install.packages("ISLR")

Method2:



ISLR

	Name	Description	Version
□	ISLR	Data for an Introduction to Statistical Learning with Applications in R	1.4


Install Packages

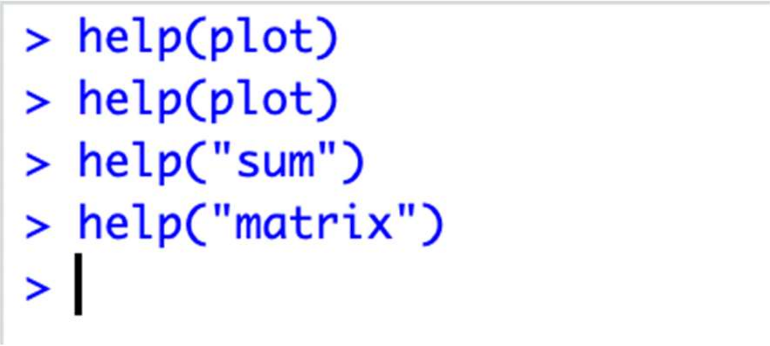
Install from: Repository (CRAN)

Packages (separate multiple with space or comma): ISLR

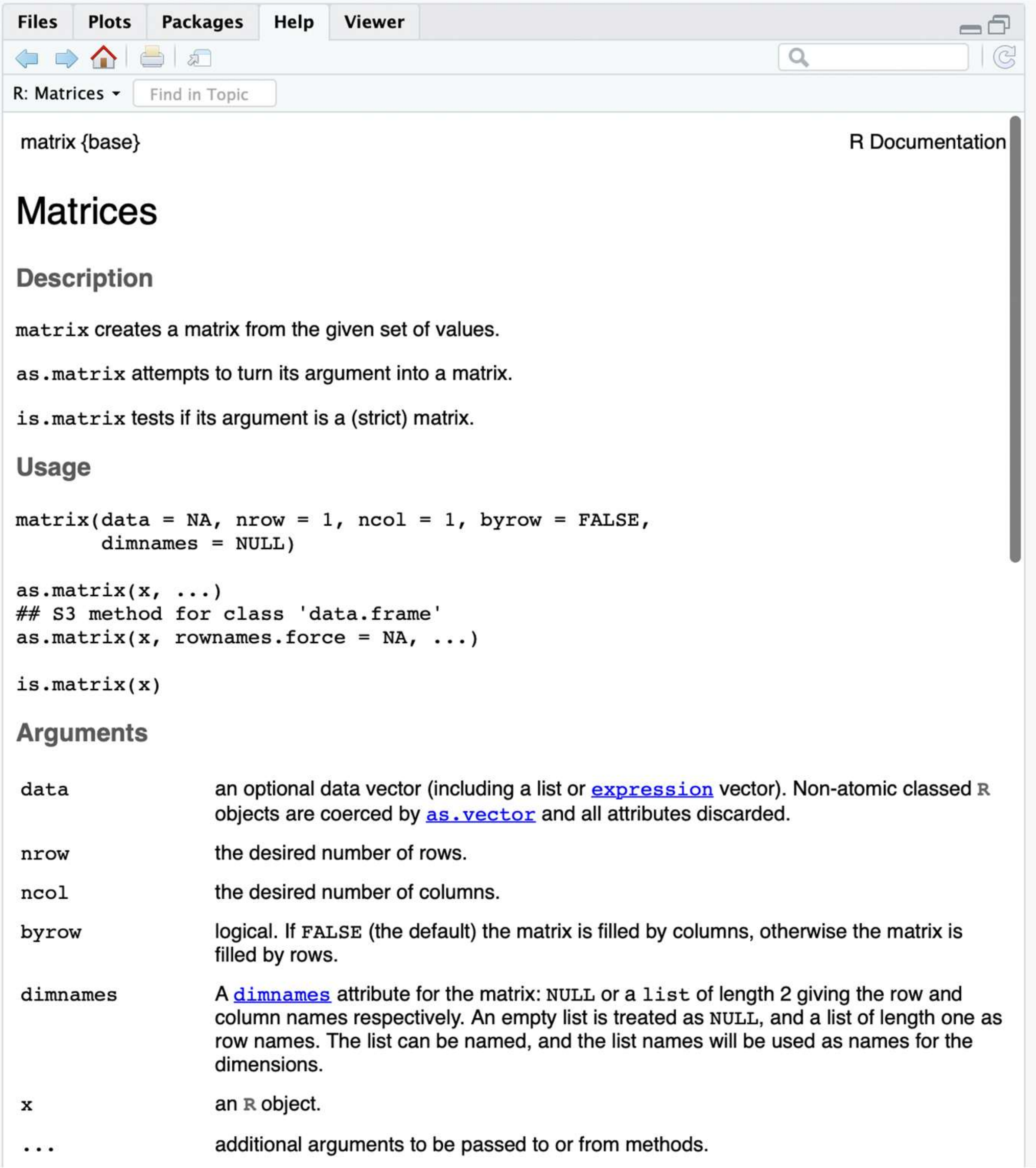
to Library: /liuyiren/Library/R/4.0/library [Default]

☑ Install dependencies

Get Help



```
> help(plot)
> help(plot)
> help("sum")
> help("matrix")
> |
```



matrix {base} R Documentation

Matrices

Description

`matrix` creates a matrix from the given set of values.

`as.matrix` attempts to turn its argument into a matrix.

`is.matrix` tests if its argument is a (strict) matrix.

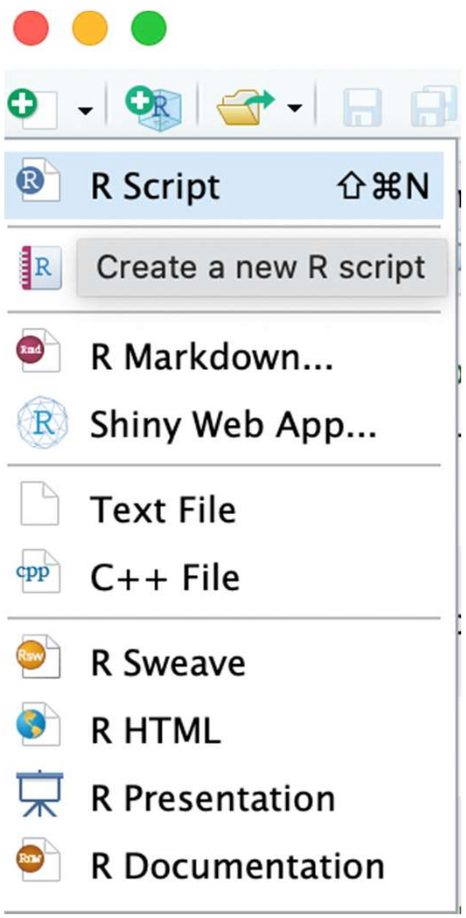
Usage

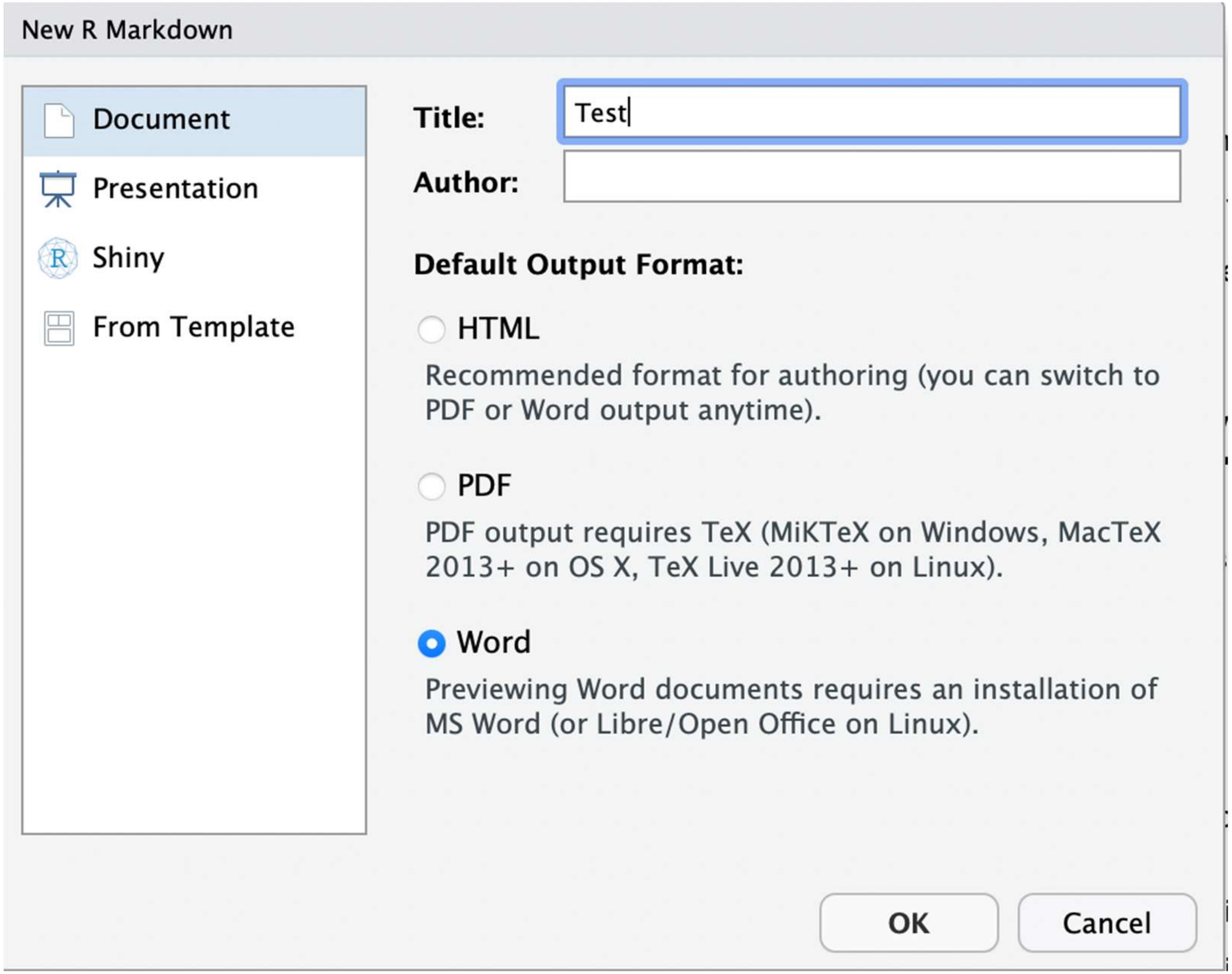
```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
       dimnames = NULL)

as.matrix(x, ...)
## S3 method for class 'data.frame'
as.matrix(x, rownames.force = NA, ...)

is.matrix(x)
```

Arguments

`data`	an optional data vector (including a list or `expression` vector). Non-atomic classed R objects are coerced by `as.vector` and all attributes discarded.
`nrow`	the desired number of rows.
`ncol`	the desired number of columns.
`byrow`	logical. If `FALSE` (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.
`dimnames`	A `dimnames` attribute for the matrix: `NULL` or a `list` of length 2 giving the row and column names respectively. An empty list is treated as `NULL`, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.
`x`	an R object.
`...`	additional arguments to be passed to or from methods.

Create Markdown/Script





Title: Test

Author:

Default Output Format:

○ HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

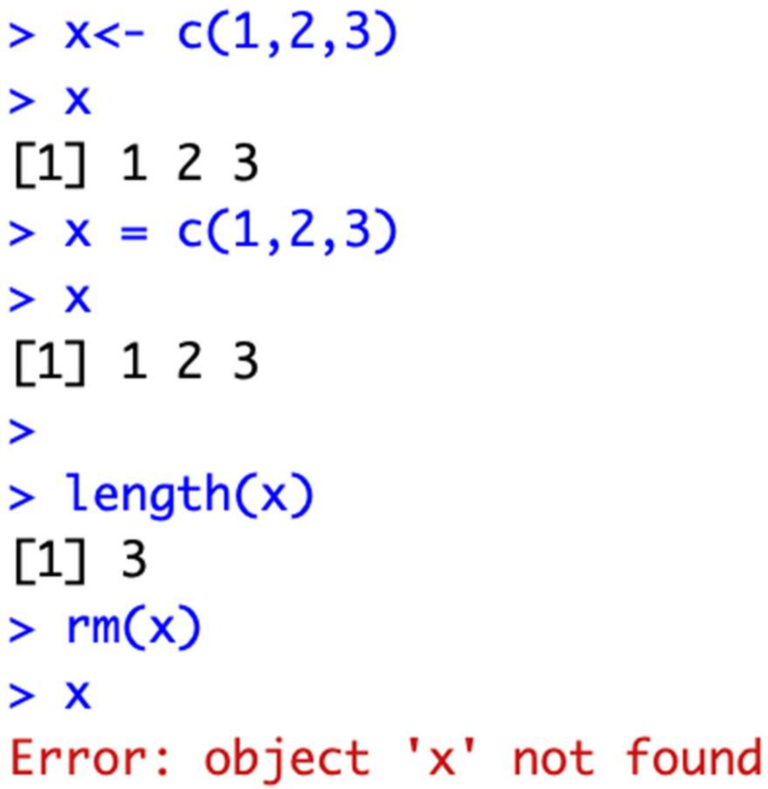
● Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Basic commands

Vector

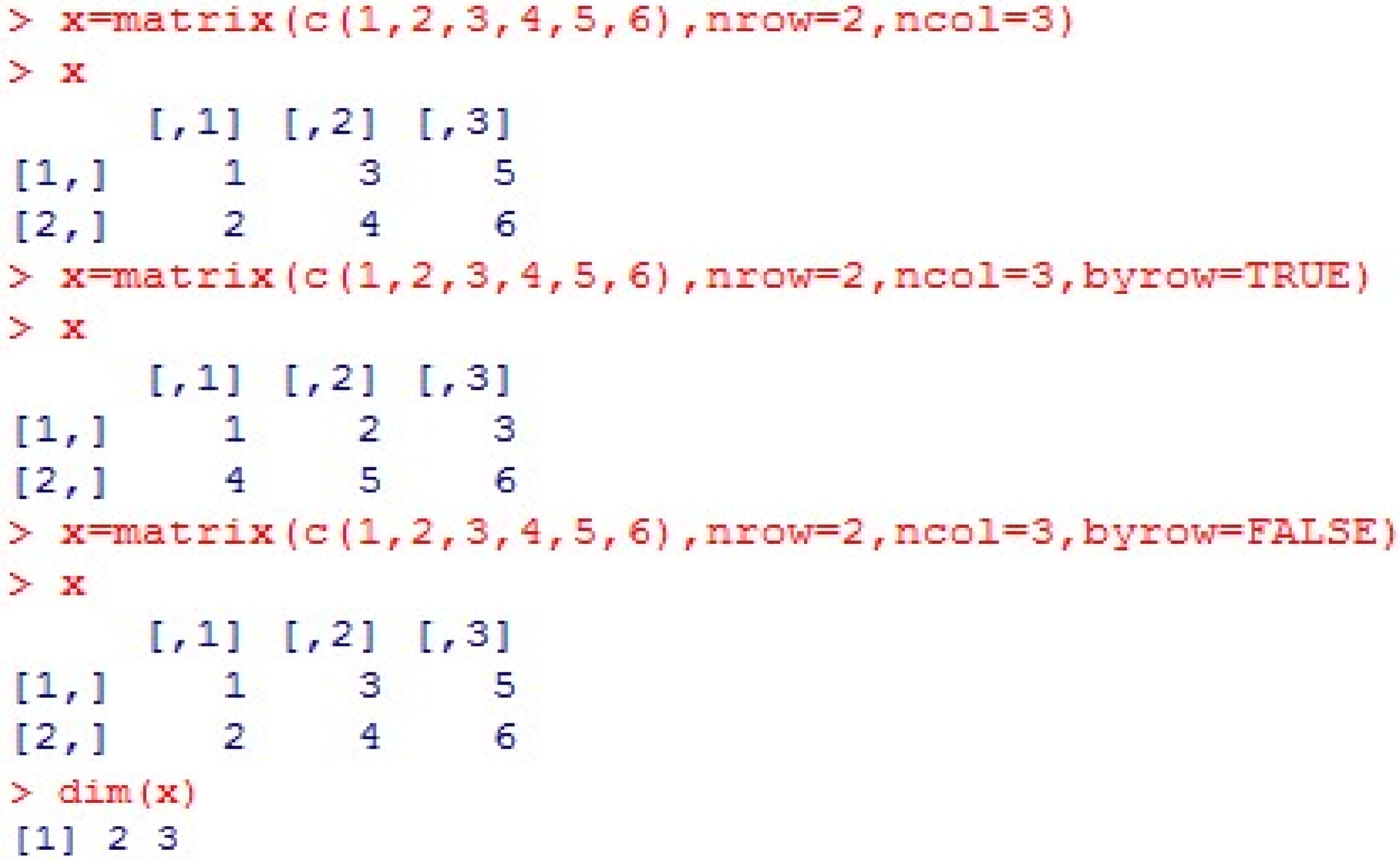
- Save things, use “<-” or “=”
- Insert vector using function “c()”
- Check length of vector using “length()”
- Delete vector: rm(x), Delete all vectors: rm(list = ls())



```
> x<- c(1,2,3)
> x
[1] 1 2 3
> x = c(1,2,3)
> x
[1] 1 2 3
>
> length(x)
[1] 3
> rm(x)
> x
Error: object 'x' not found
```

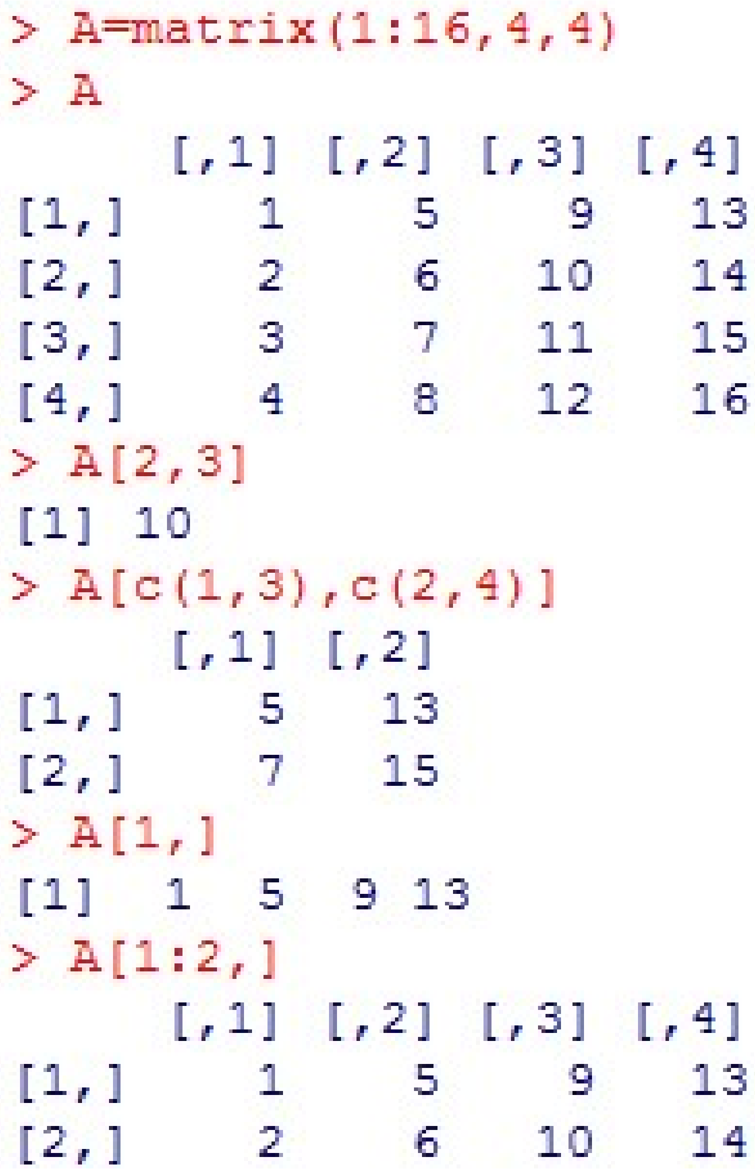
Matrix

- Declare a matrix using function “matrix()”
- Use “byrow=TRUE/FALSE” to specify order
- Use “dim()” to find dimension of a matrix

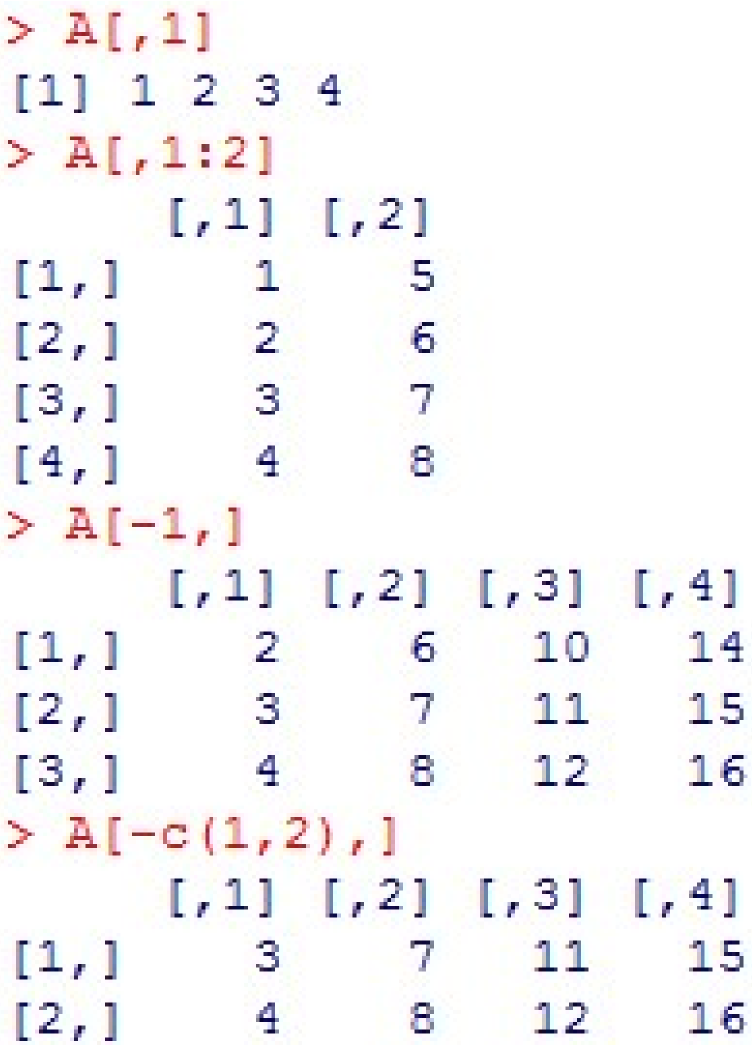


```
> x=matrix(c(1,2,3,4,5,6),nrow=2,ncol=3)
> x
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> x=matrix(c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=TRUE)
> x
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> x=matrix(c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=FALSE)
> x
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> dim(x)
[1] 2 3
```

Select Elements in A Matrix



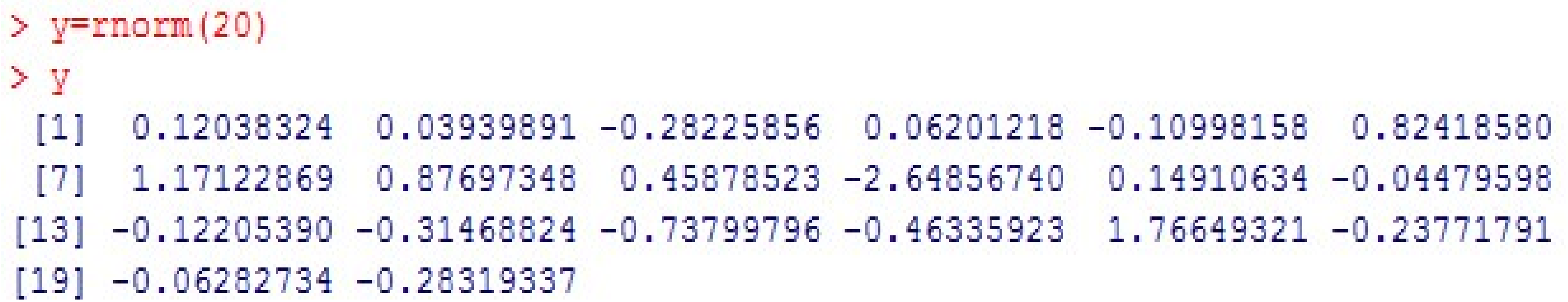
```
> A=matrix(1:16,4,4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> A[2,3]
[1] 10
> A[c(1,3),c(2,4)]
     [,1] [,2]
[1,]    5   13
[2,]    7   15
> A[1,]
[1]  1  5  9 13
> A[1:2,]
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
```



```
> A[,1]
[1] 1 2 3 4
> A[,1:2]
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
> A[-1,]
     [,1] [,2] [,3] [,4]
[1,]    2    6   10   14
[2,]    3    7   11   15
[3,]    4    8   12   16
> A[-c(1,2),]
     [,1] [,2] [,3] [,4]
[1,]    3    7   11   15
[2,]    4    8   12   16
```

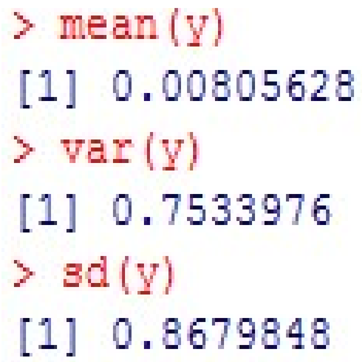
Generate Random Numbers

- Generate random numbers from a standard normal distribution using “rnorm(n)”



```
> y=rnorm(20)
> y
 [1]  0.12038324  0.03939891 -0.28225856  0.06201218 -0.10998158  0.82418580
 [7]  1.17122869  0.87697348  0.45878523 -2.64856740  0.14910634 -0.04479598
[13] -0.12205390 -0.31468824 -0.73799796 -0.46335923  1.76649321 -0.23771791
[19] -0.06282734 -0.28319337
```

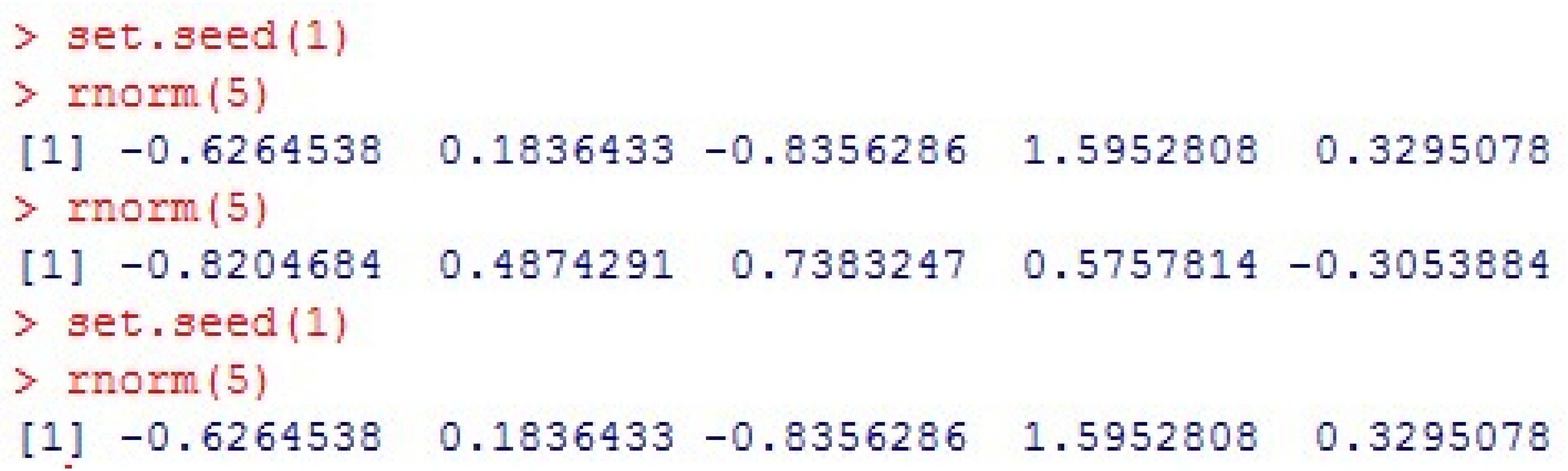
- Calculate mean(), var(), sd() of random numbers



```
> mean(y)
[1] 0.00805628
> var(y)
[1] 0.7533976
> sd(y)
[1] 0.8679848
```

Set the Seed of Random Number Generator

- Set the seed of random number generator using “set.seed()”
- To reproduce the exact same set of random numbers, use the same seed



```
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
> rnorm(5)
[1] -0.8204684  0.4874291  0.7383247  0.5757814 -0.3053884
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

Example of preliminary analysis of a dataset

Load Dataset

- To load a data set in the ISLR package or other packages/libraries, you only need to load the package

```
> library(ISLR)
```

- To load an external data set, first specify the directory "File" → "Change dir…"

- If the data are saved as a text file

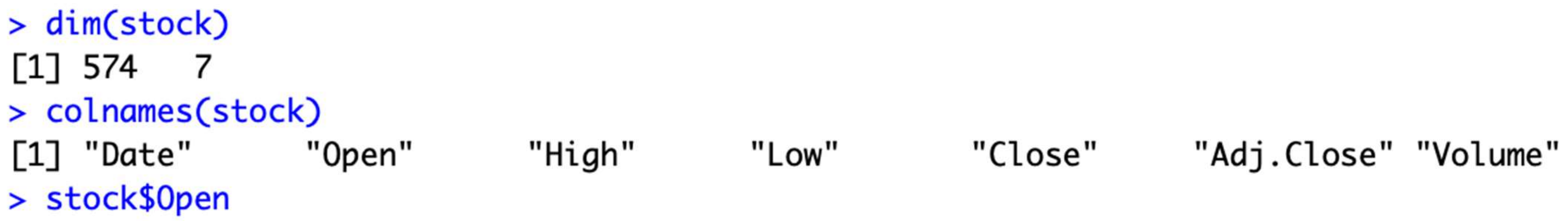
  Auto=read.table('Auto.data',header=T,na.strings='?')

- If the data are saved as a csv file (Excel)

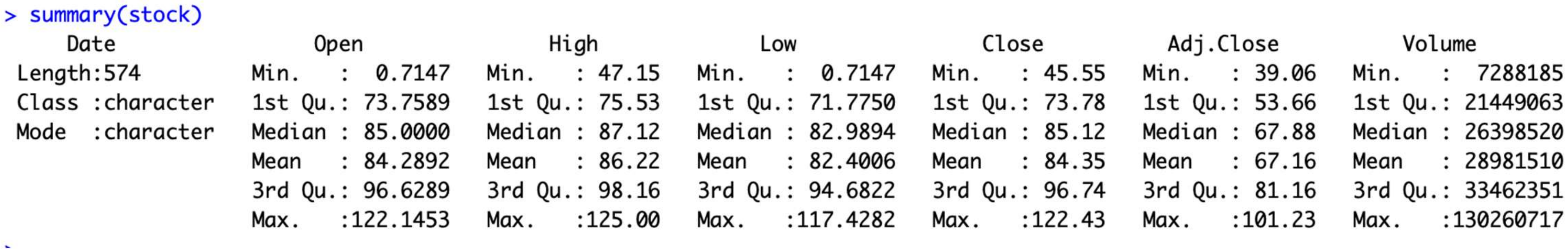
  stock=read.csv('0001.HK.csv',na.strings='?')

Basic operations of Dataset

- Use dim() to check number of row and column
- Use colnames() to check column names
- Use stock$Open or stock[,’Open’] to view the value of a column
- Use stock[1,] to view the first row
- Use Summary to get numerical summaries



```
> dim(stock)
[1] 574   7
> colnames(stock)
[1] "Date"      "Open"      "High"      "Low"       "Close"     "Adj.Close" "Volume"
> stock$Open
```



```
> summary(stock)
     Date                Open               High             Low                Close          Adj.Close          Volume
 Length:574         Min.   :  0.7147   Min.   : 47.15   Min.   :  0.7147   Min.   : 45.55   Min.   : 39.06   Min.   :  7288185
 Class :character   1st Qu.: 73.7589   1st Qu.: 75.53   1st Qu.: 71.7750   1st Qu.: 73.78   1st Qu.: 53.66   1st Qu.: 21449063
 Mode  :character   Median : 85.0000   Median : 87.12   Median : 82.9894   Median : 85.12   Median : 67.88   Median : 26398520
                    Mean   : 84.2892   Mean   : 86.22   Mean   : 82.4006   Mean   : 84.35   Mean   : 67.16   Mean   : 28981510
                    3rd Qu.: 96.6289   3rd Qu.: 98.16   3rd Qu.: 94.6822   3rd Qu.: 96.74   3rd Qu.: 81.16   3rd Qu.: 33462351
                    Max.   :122.1453   Max.   :125.00   Max.   :117.4282   Max.   :122.43   Max.   :101.23   Max.   :130260717
```

Plot something

```
# Plot open price
plot(stock[,'Open'],ylab = c("Open Price"), type = 'l')
grid()
title('HK.0001')

# show open and high price in the same figure
plot(stock[,'Open'],ylab = c("Price"),type = 'l')
lines(stock[,'High'], col = "red")
grid()
legend("topleft",c("Open","High"), lty = 1,col=c("black","red"))
title('HK.0001')

# compare open price and high price
plot(stock[,'Open'],stock[,'High'],xlab = c('Open Price'), ylab = c("High Price"))
grid()
title('HK.0001')
```