Introduction to Data Science, Topic 1 & 2

- Instructor: Professor Henry Horng-Shing Lu,

  Institute of Statistics, National Chiao Tung University, Taiwan

  Email: hslu@stat.nctu.edu.tw

- WWW: http://www.stat.nctu.edu.tw/misg/hslu/course/DataScience.htm
- Reference:

  M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.
- Evaluation: Homework: 50%, Term Project: 50%
- Office hours: By appointment

Course Outline

- **Introduction of data science**
- **Introduction of R**
- **Data Visualization**
- **Exploratory Data Analysis**
- **Regression**
- **Classification**
- **Text Mining**
- **Clustering**

Topic 1: Introduction of data science

References:

Ch. 1, M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

https://en.wikipedia.org/wiki/Data_science

What Is Data Science?

- Data science, also known as data-driven science, is an interdisciplinary field of scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured, similar to data mining. [wiki]
- Video: The Human Face of Big Data

  https://www.youtube.com/watch?v=4VeITe6EJDU

  (with Chinese caption)

  https://ihavenotv.com/the-human-face-of-big-data-nova-pbs

  (without caption)

Why R?

- Open Source
- R works on most of the mainstream computer systems: GNU/Linux, OS X, Windows
- Statistics community based, provides many high level packages and functions
- Complete documentation and discussion area
- The program is easy to modify according to the needs

Homework 1 (submitted to e3new.nctu.edu.tw before Sept 24, 2019)

- Find at least one data set that you plan to study for your future homework and final project
- Explain the features in your data set
- Discuss possible problems you plan to investigate based on the data sets you select
- Possible source of open data:

  UCI Machine Learning Repository

  (https://archive.ics.uci.edu/ml/datasets.php)

Topic 2 - Introduction of R

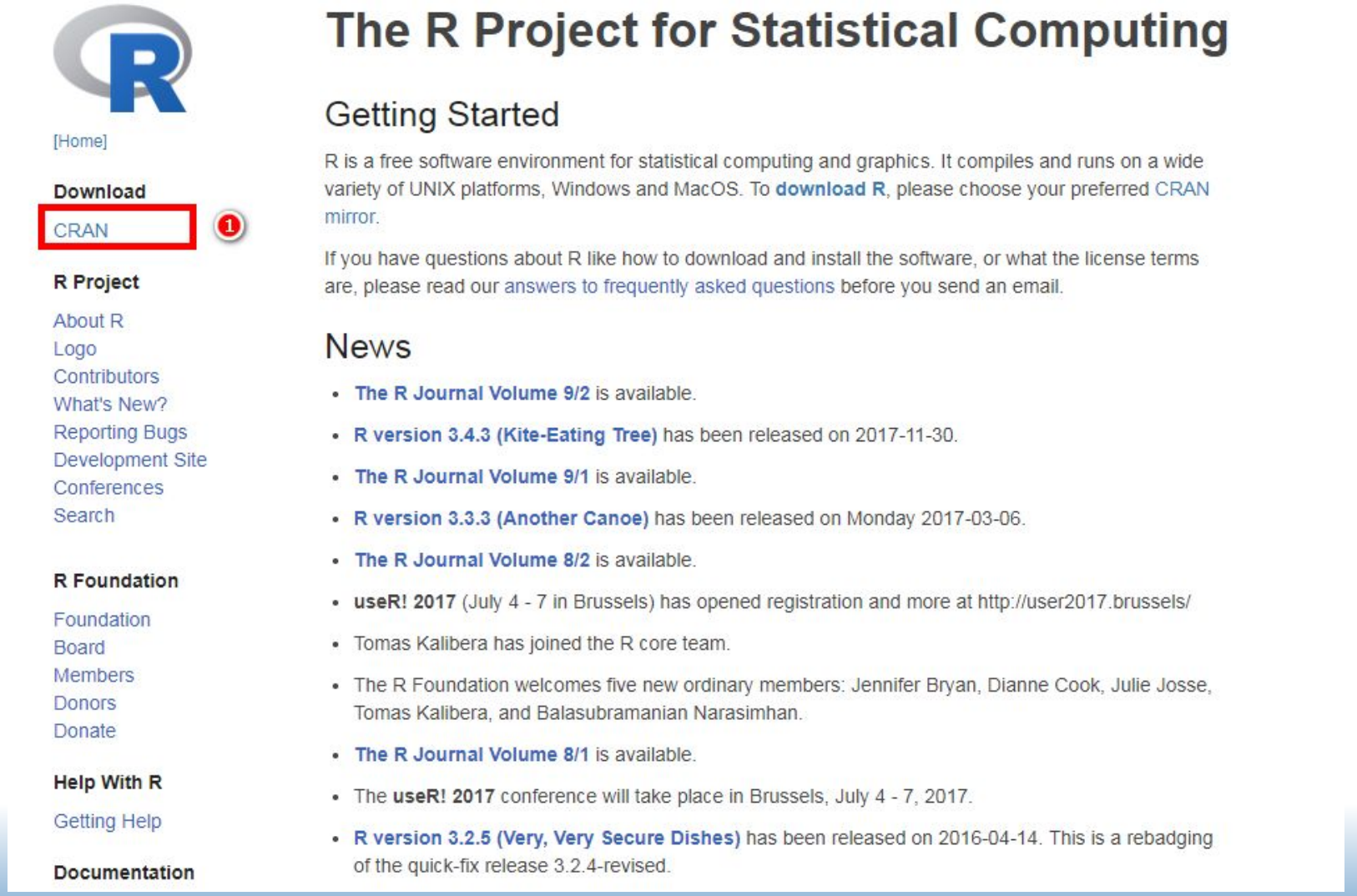
References:

Ch. 2, M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

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

https://www.statmethods.net/input/datatypes.html

Installing R - Download

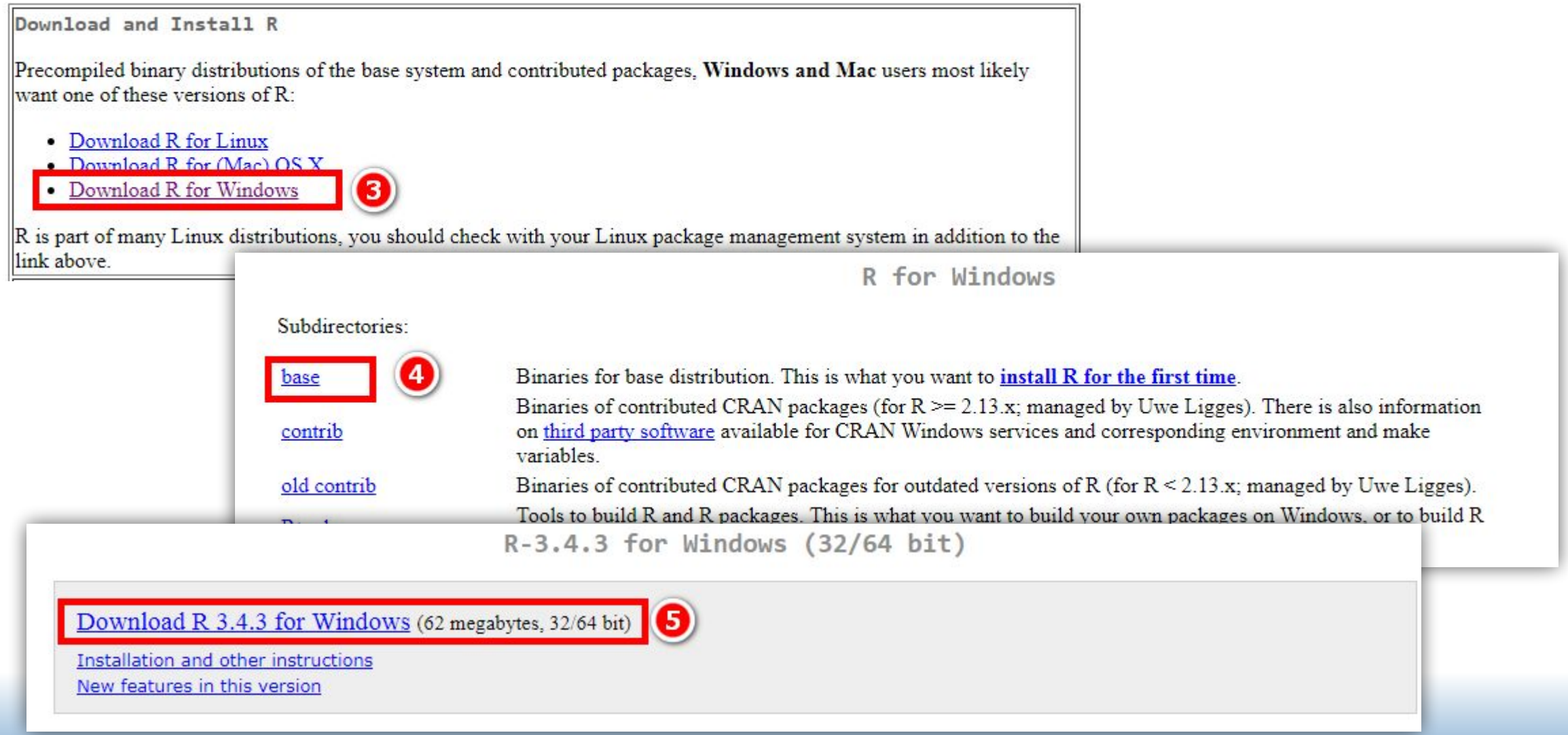


Installing R - Download

Sweden	
https://ftp.acc.umu.se/mirror/CRAN/	Academic Computer Club, Umeå University
http://ftp.acc.umu.se/mirror/CRAN/	Academic Computer Club, Umeå University
Switzerland	
https://stat.ethz.ch/CRAN/	ETH Zürich
http://stat.ethz.ch/CRAN/	ETH Zürich
Taiwan	
https://ftp.yzu.edu.tw/CRAN/	Department of Computer Science and Engineering, Yuan Ze University
http://ftp.yzu.edu.tw/CRAN/	Department of Computer Science and Engineering, Yuan Ze University
http://cran.csie.ntu.edu.tw/	National Taiwan University, Taipei
Thailand	
http://mirrors.psu.ac.th/pub/cran/	Prince of Songkla University, Hatyai
Turkey	
https://cran.pau.edu.tr/	Pamukkale University, Denizli
http://cran.pau.edu.tr/	Pamukkale University, Denizli
https://cran.ncc.metu.edu.tr/	Middle East Technical University Northern Cyprus Campus, Mersin
http://cran.ncc.metu.edu.tr/	Middle East Technical University Northern Cyprus Campus, Mersin

choose one ❷

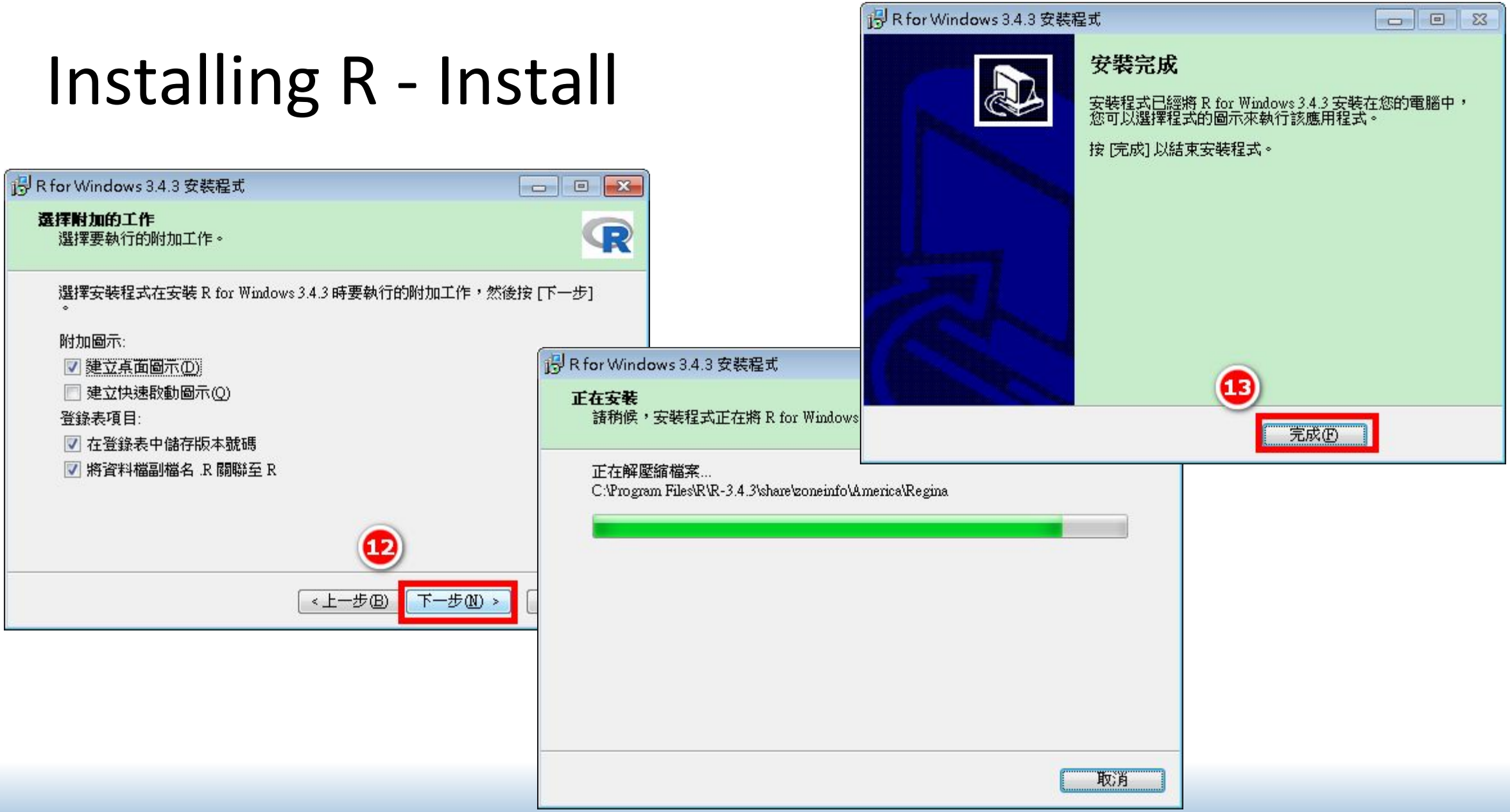
Installing R - Download



Installing R - Install

Installing R - Install

Installing R - Install



Demo

- The R Project for Statistical Computing

Writing and Reading Data

- Read/Write for Text File

  Ex.

```
getwd() #check work directory
write.table(iris, "iris.txt")
data = read.table("iris.txt", sep=" ", header=TRUE)
data
```

- Read/Write for Comma-Separated File(CSV)

  Ex.

```
write.csv(iris, "iris.csv")
data2 = read.csv("iris.csv")
data2
```

Reading Data – Check Type and Structure

```
> class(data)
[1] "data.frame“

> str(data)
'data.frame':    150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1
1 1 1 1 1 1 ...
```

Read Data from other statistics software

Function	Description
read.dbf	Read a DBF File
read.dta	Read Stata binary files
read.epiinfo	Read Epi Info data files
read.mtp	Read a Minitab PortableWorksheet
read.octave	Read Octave Text Data Files
read.spss	Read an SPSS data file
read.ssd	Read a data frame from a SAS Permanent Dataset
read.systat	Read a data frame from a Systat File
read.xport	Read a SAS XPORT Format Library

R - Operators

Arithmetic Operators

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2 is 2

Logical Operators

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x | y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

Data Type

- Vectors
- Matrices
- Arrays
- Data Frames
- Lists
- Factors

Vectors

- Ex.

```
> # Vectors
> a <- c(1,2,5,6,-2,4);a # numeric vector
[1]  1  2  5  6 -2  4
> b <- c("one", "two", "three");b # character vector
[1] "one"   "two"   "three"
> c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE);c #logical vector
[1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE
```

Matrices

```
> # Matrices
> # generates 5 x 4 numeric matrix
> x <- matrix(1:20, nrow=5, ncol=4);x
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20

> # Identify rows, columns or elements using subscripts
> x[,4]
[1] 16 17 18 19 20
> x[3,4]
[1] 18
> x[2:4, 1:3]
     [,1] [,2] [,3]
[1,]    2    7   12
[2,]    3    8   13
[3,]    4    9   14
```

Arrays

- Arrays are similar to matrices but can have more than two dimensions.
- Ex.

```
> # Arrays
> x = array(1:24, c(2,3,4));x
, , 1
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2
     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12
```

Data Frames

- A data frame is more general than a matrix, in that different columns can have different modes (numeric, character, factor, etc.)
- Ex.

```
> # Data Frames
> d <- c(1,2,3,4)
> e <- c("red", "white", "red", NA)
> f <- c(TRUE, TRUE, TRUE, FALSE)
> myData <- data.frame(d,e,f)
> names(myData) <- c("ID", "Color", "Passed") # variable names
> myData
  ID Color Passed
1  1   red   TRUE
2  2 white   TRUE
3  3   red   TRUE
4  4  <NA>  FALSE
```

Data Frames - identify the elements

- Ex.

```
> # identify the elements
> myData[2:3] # columns 2,3 of data frame
  Color Passed
1   red   TRUE
2 white   TRUE
3   red   TRUE
4  <NA>  FALSE
> myData[c("ID", "Passed")] # columns ID and Passed from data
frame
  ID Passed
1  1   TRUE
2  2   TRUE
3  3   TRUE
4  4  FALSE
> myData$ID # Variable ID in the data frame
[1] 1 2 3 4
```

Lists

- Ex.

```
> # Lists
> # example of list with 3 components -
> # a string, a numeric vectors, and a scaler
> w <- list(name = "Henry", cost=c(5,10,15), age=20);w
$name
[1] "Henry"

$cost
[1]  5 10 15

$age
[1] 20r
```

Lists - Identify elements using [[]]

- Ex.

```
> # identify elements of a list using the [[]] convention
> w[[2]]
[1]  5 10 15
> w[["name"]]
[1] "Henry"
```

Factors

- Ex.

```
> # Factors
> # variable gender with 20 "male" entries and
> # 30 "female" entries
> gender <- c(rep("male", 20), rep("female", 30))
> gender <- factor(gender)
> # R treats gender as a nominal variable
>
> summary(gender)
female   male
    30     20
```

Check Data Type : Class Function

- class(): We identify the data type of a variable using the class() function.

- Ex.

```
> x = 5
> class(x)
[1] "numeric"
```

Data Type Transformation

- as.[DataType]
  - as.interger()
  - as.numeric()
  - as.charactor()
  - ……
- Ex.

```
> as.integer(2.5)
[1] 2
> as.numeric("2.5")
[1] 2.5
> as.character(2.5)
[1] "2.5"
```

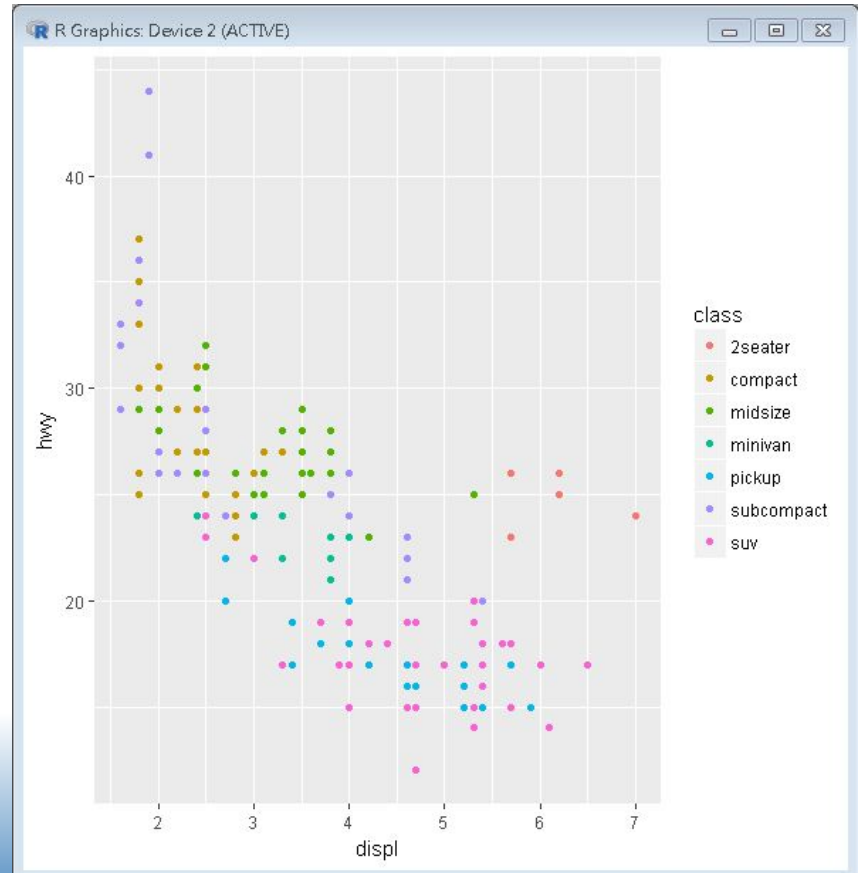
User Define Function

- Ex. average function

```
> avg = function(x) {
+          return(sum(x)/length(x))
+ }
> avg(1:10)
[1] 5.5
```

Install and import Packages

```
> install.packages("ggplot2")
Installing package into ‘C:/Users/famous/Documents/R/win-library/3.4’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/ggplot2_2.2.1.zip'
Content type 'application/zip' length 2783267 bytes (2.7 MB)
downloaded 2.7 MB

package ‘ggplot2’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\famous\AppData\Local\Temp\RtmpOgapa1\downloaded_packages
> library(ggplot2)
> ggplot(mpg, aes(displ, hwy, colours=class))+geom_point()
```



R Help System

- help() or ?: we can look up the documentation.
- example(): we can also see the example

- Ex.

```
> example(cat)

cat> iter <- stats::rpois(1, lambda = 10)
cat> ## print an informative message
cat> cat("iteration = ", iter <- iter + 1, "\n")
iteration =  21
cat> ## 'fill' and label lines:
cat> cat(paste(letters, 100* 1:26), fill = TRUE, labels = paste0("{", 1:10, "}:"))
{1}: a 100 b 200 c 300 d 400 e 500 f 600 g 700 h 800 i 900 j 1000 k 1100
{2}: l 1200 m 1300 n 1400 o 1500 p 1600 q 1700 r 1800 s 1900 t 2000 u 2100
{3}: v 2200 w 2300 x 2400 y 2500 z 2600
```

Homework 2 (submitted to e3new.nctu.edu.tw before Oct 1, 2019)

- Use R or the programming language you prefer to analyze the data set that you select
- Explain the results you obtain
- Discuss possible problems you plan to investigate for future studies
- Possible source of open data:

  UCI Machine Learning Repository

  (https://archive.ics.uci.edu/ml/datasets.php)

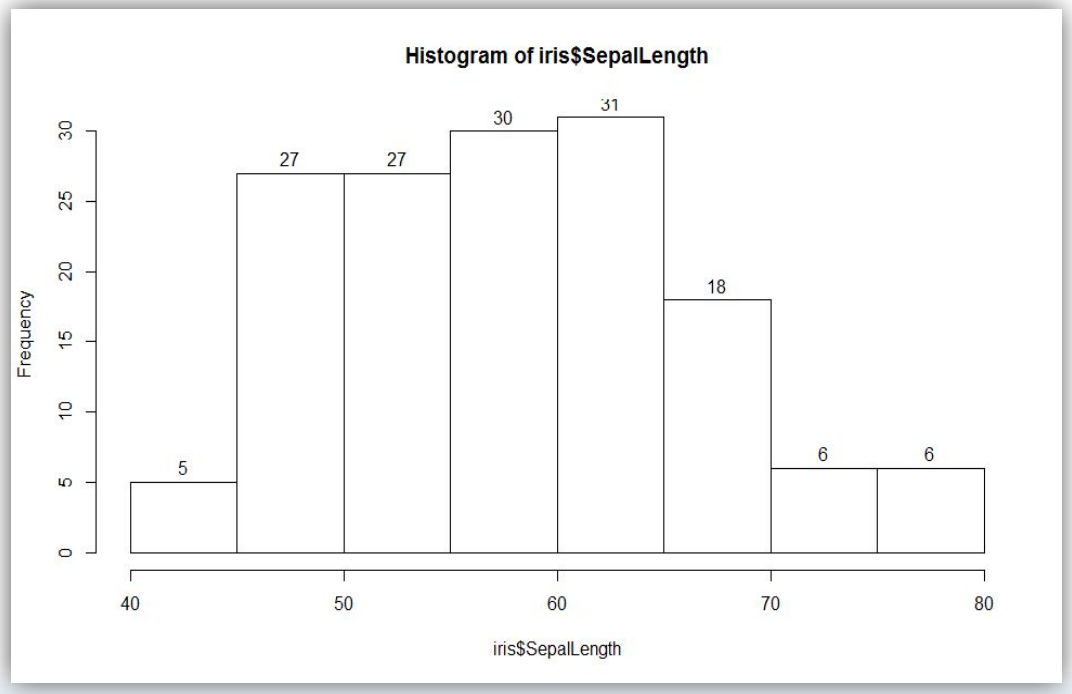
Topic 2.0.1: Black and White Histogram with R

References:

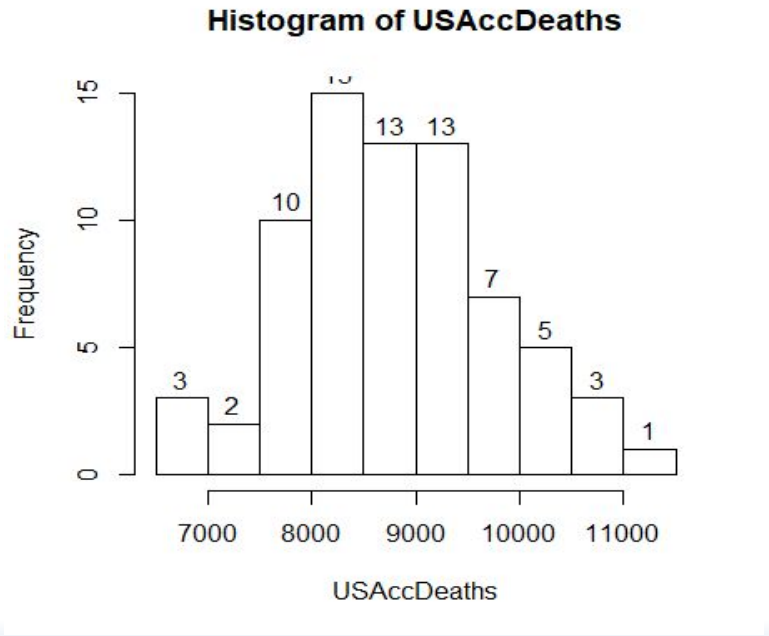
http://stat.ethz.ch/R-manual/R-devel/library/graphics/html/hist.html

Black and White Histogram With R

```
iris = read.csv("d:/workspace/iris.csv")
hist(iris$SepalLength, labels = TRUE)
```




```
library('datasets')
# the monthly totals of accidental deaths in the USA(1973-1978)
hist(USAccDeaths,labels = TRUE)
```



Topic 2.0.2: Color Histogram

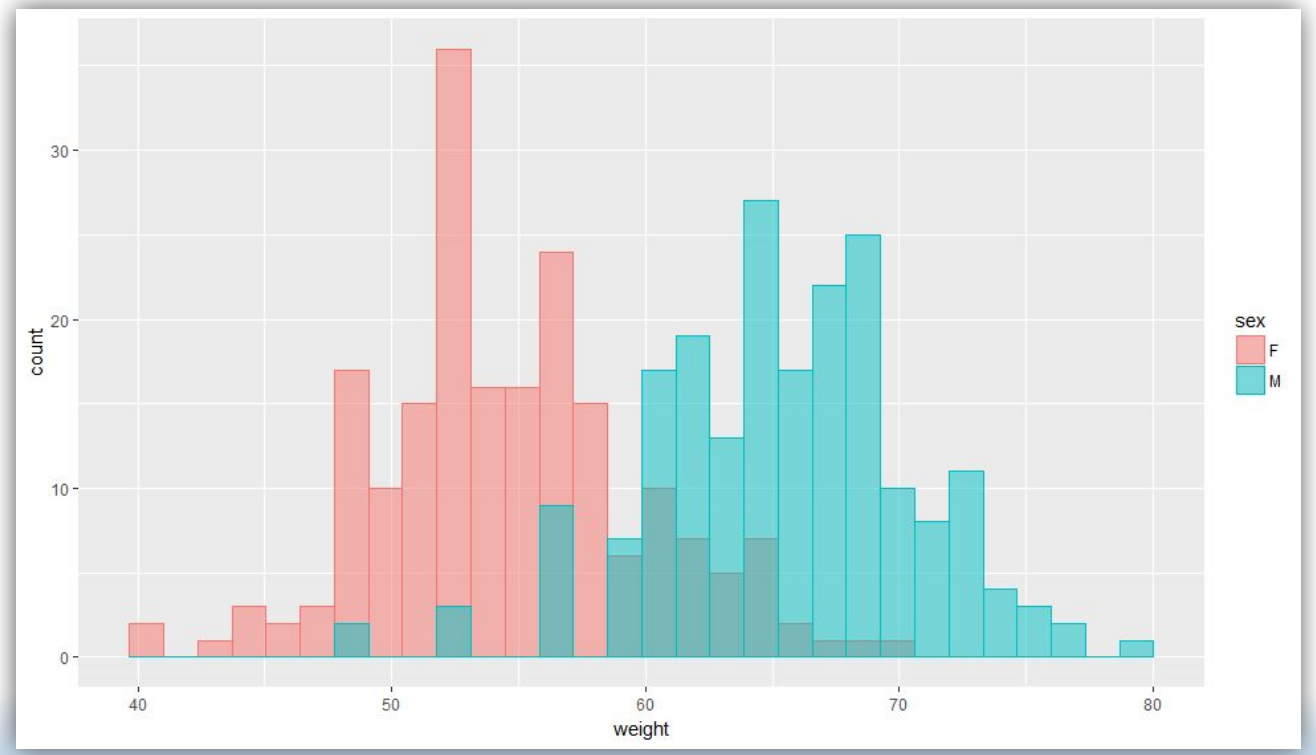
References:

http://www.r-tutor.com/elementary-statistics/quantitative-data/histogram

http://www.sthda.com/english/wiki/ggplot2-histogram-plot-quick-start-guide-r-software-and-data-visualization

Color Histogram With R

```
set.seed(1234)
df <- data.frame(
  sex=factor(rep(c("F", "M"), each=200)),
  weight=round(c(rnorm(200, mean=55, sd=5), rnorm(200,
mean=65, sd=5)))
)

library(ggplot2)
# Use semi-transparent fill
p<-ggplot(df, aes(x=weight, fill=sex, color=sex)) +
  geom_histogram(position="identity", alpha=0.5)
p
```



Topic 2.0.2: Different Color Models

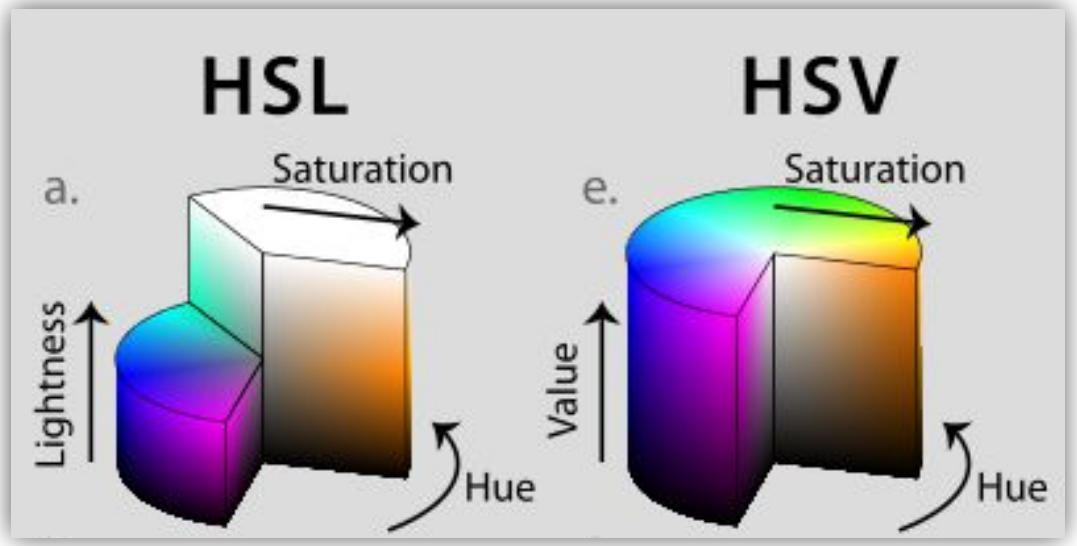
References:

https://en.wikipedia.org/wiki/Color_model

https://en.wikipedia.org/wiki/HSL_and_HSV

Color model

- HSL (Hue, Saturation, Lightness)
- HSV (Hue, Saturation, Value)



Different Color Models

```
> library(ColorPalette)
> library(grDevices)
> library(stringi)
> library(imager)
> library(ImaginR)
```

```
> (rc <- col2rgb("red"))
      [,1]
red    255
green    0
blue     0

> (hc <- rgb2hsv(rc))
  [,1]
h    0
s    1
v    1

>  hex2rgb(hsv2rgb(hc[1], hc[2], hc[3]))
      R G B
[1,] 255 0 0
```

Topic 2.1: Introduction of Matlab

References:

http://www.cyclismo.org/tutorial/matlab/vector.html

https://www.youtube.com/watch?v=qGiKv3-02vw

Read/Write Data Set

- Write data To CSV

```
writetable(data,"iris_test.csv")
```

- Read CSV To Data Set

```
filename = 'iris.csv‘
data = readtable(filename);
```

Introduction of Matlab

data

```
>> data

data =

  150×5 table

    Sepal_Length    Sepal_Width    Petal_Length    Petal_Width     Species
    ____________    ___________    ____________    ___________    ___________

    5.1             3.5            1.4             0.2            'setosa'
```

5.1	3.5	1.4	0.2	'setosa'
4.9	3	1.4	0.2	'setosa'
4.7	3.2	1.3	0.2	'setosa'
4.6	3.1	1.5	0.2	'setosa'
5	3.6	1.4	0.2	'setosa'
5.4	3.9	1.7	0.4	'setosa'
4.6	3.4	1.4	0.3	'setosa'
5	3.4	1.5	0.2	'setosa'
4.4	2.9	1.4	0.2	'setosa'
4.9	3.1	1.5	0.1	'setosa'

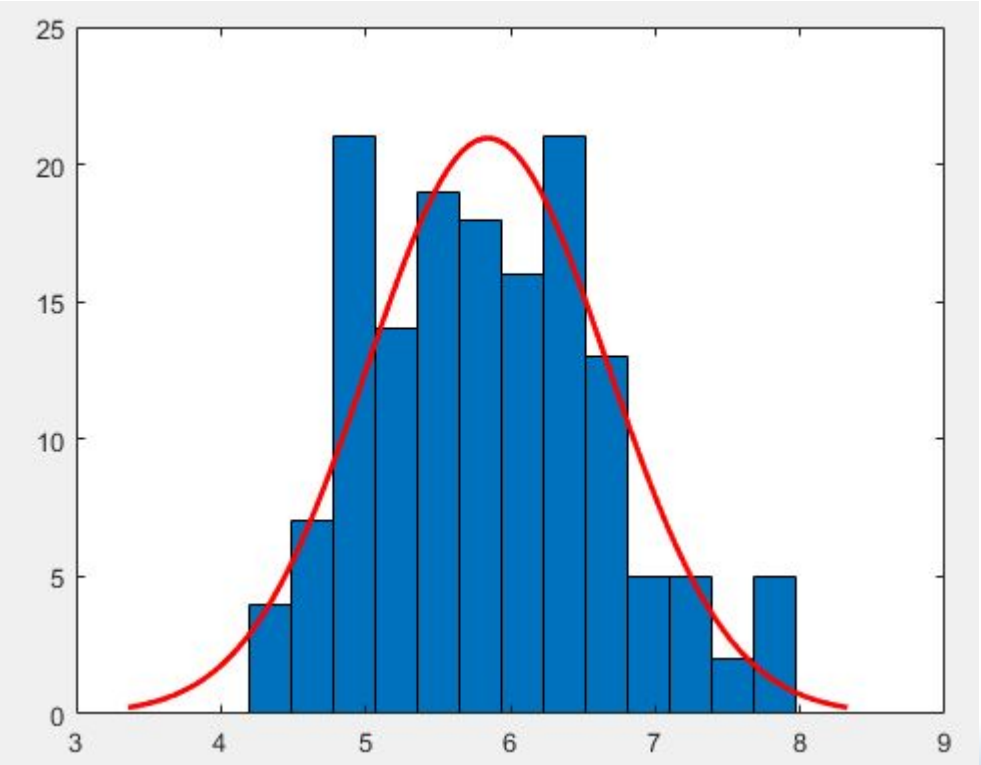
Introduction of Matlab

summary(data)

```
Variables:

    Sepal_Length: 150×1 double

        Description:  Original column heading: 'Sepal.Length'
        Values:

            Min       4.3
            Median    5.8
            Max       7.9

    Sepal_Width: 150×1 double

        Description:  Original column heading: 'Sepal.Width'
        Values:

            Min         2
            Median      3
            Max       4.4

    Petal_Length: 150×1 double

        Description:  Original column heading: 'Petal.Length'
        Values:

            Min         1
            Median   4.35
            Max       6.9
```

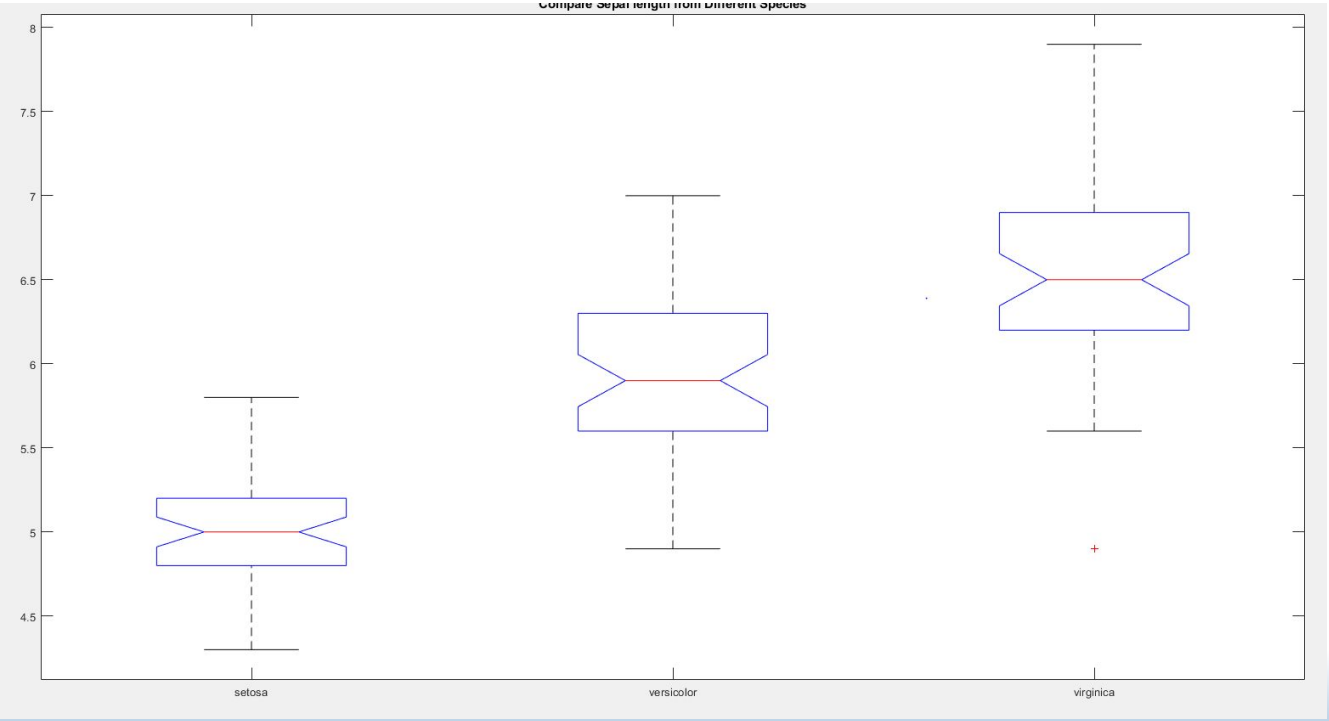
Introduction of Matlab – Data Visualization

```
figure()
data{:,{'Sepal_Length'}}
histfit(x)
```



Introduction of Matlab – Data Visualization

```
x = data{:,{'Sepal_Length'}}
x1 = x(1:50)
x2 = x(51:100)
x3 =x(101:150)
figure
boxplot([x1,x2,x3],'Notch','on','Labels',
{'setosa','versicolor','virginica'})
title('Compare Sepal length from Different Species')
```



Topic 2.2: Introduction of SAS

References:

https://en.wikipedia.org/wiki/SAS_(software)

http://support.sas.com/documentation/cdl/en/acpcref/63184/HTML/default/viewer.htm#a003102702.htm

http://support.sas.com/documentation/cdl/en/procstat/66703/HTML/default/viewer.htm#procstat_corr_syntax01.htm

Introduction of SAS

- The SAS language is a computer programming language for statistical analysis. It can read data from general spreadsheets and databases, and output statistical analysis results in the form of tables, graphs, and RTF, HTML, and PDF documents. The SAS language runs under a compiler that can be used for Microsoft Windows, Linux, and various other UNIX and mainframe computers. The SAS system and the World Programming System (WPS) are SAS language compilers.
- The statistical analysis system company continuously develops **business data analysis** and **forecasting technologies** with all walks of life. The important application areas include the **government's economic decision-making** and the **company's decision support applications**. It has become the world's fifth-largest software company.

Introduction of SAS

- Iris Data Set: (in sashelp library)
  - Sepal Length (mm)
  - Sepal Width (mm)
  - Petal Length (mm)
  - Petal Width (mm)
  - Species

Read/Write Data Set

- Write SAS Data Set To CSV

```
proc export data=sashelp.iris
  outfile='D:\workspace\iris_sas.csv'
  dbms=csv
  replace;
run;
```

- Read CSV To SAS Data Set

```
data work.iris_sas;
        length Species $ 20;
        infile 'D:\workspace\iris_sas.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;
        input Species $ SepalLength SepalWidth PetalLength PetalWidth;
run;
```

Introduction of SAS

proc print data=sashelp.iris;run;

Iris Data Set

Obs	Species	SepalLength	SepalWidth	PetalLength	PetalWidth
1	Setosa	50	33	14	2
2	Setosa	46	34	14	3
3	Setosa	46	36	10	2
4	Setosa	51	33	17	5
5	Setosa	55	35	13	2

101	Virginica	64	28	56	22
102	Virginica	67	31	56	24
103	Virginica	63	28	51	15
104	Virginica	69	31	51	23
105	Virginica	65	30	52	20
106	Virginica	65	30	55	18
107	Virginica	58	27	51	19
108	Virginica	68	32	59	23
109	Virginica	62	34	54	23
110	Virginica	77	38	67	22

51	Versicolor	65	28	46	15
52	Versicolor	62	22	45	15
53	Versicolor	59	32	48	18
54	Versicolor	61	30	46	14
55	Versicolor	60	27	51	16
56	Versicolor	56	25	39	11
57	Versicolor	57	28	45	13
58	Versicolor	63	33	47	16
59	Versicolor	70	32	47	14
60	Versicolor	64	32	45	15

Introduction of SAS – Descriptive statistics

```
proc corr data=sashelp.iris;run;
```

Iris Data Set

The CORR Procedure

4 Variables:	SepalLength SepalWidth PetalLength PetalWidth

Simple Statistics

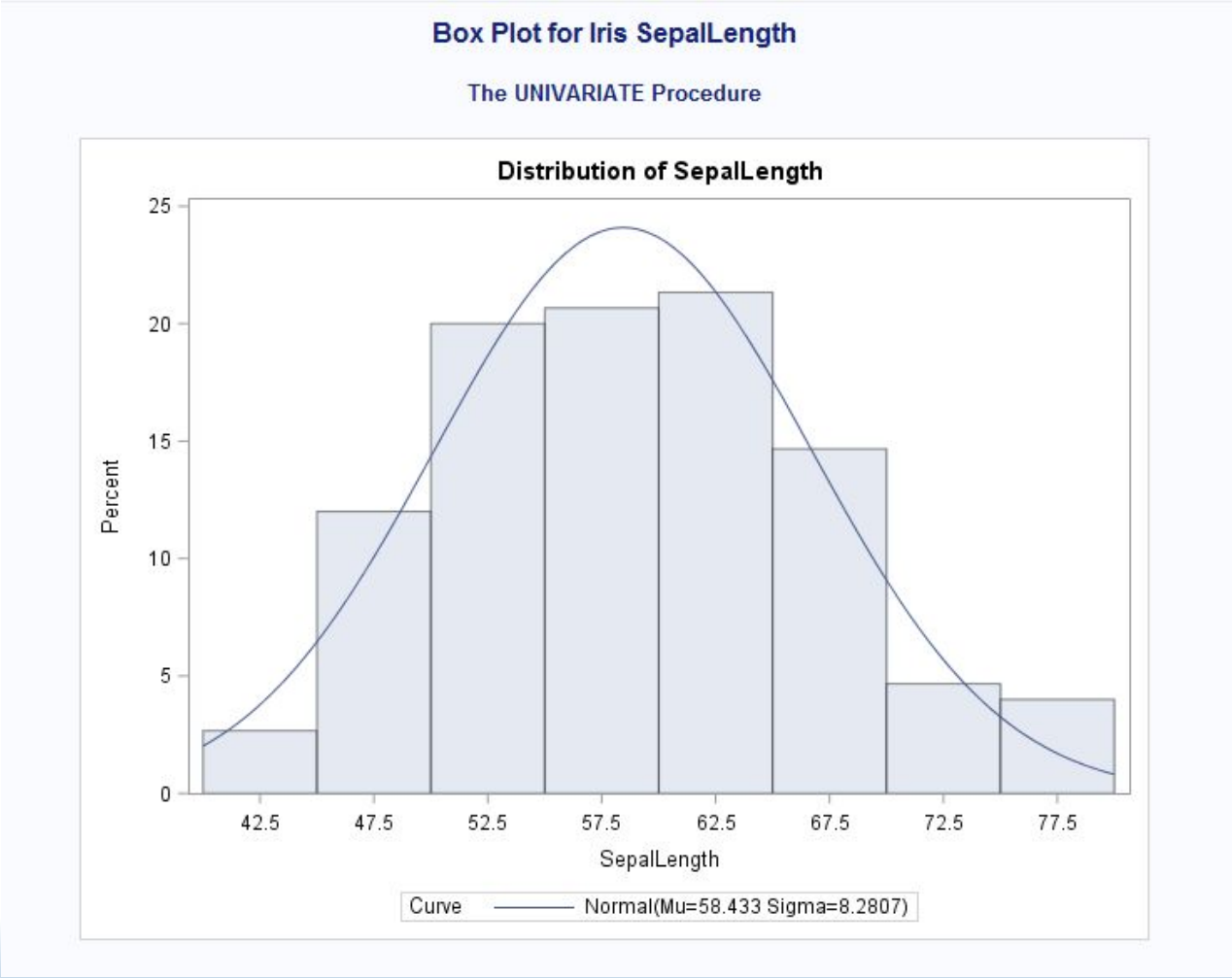
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
SepalLength	150	58.43333	8.28066	8765	43.00000	79.00000	Sepal Length (mm)
SepalWidth	150	30.57333	4.35866	4586	20.00000	44.00000	Sepal Width (mm)
PetalLength	150	37.58000	17.65298	5637	10.00000	69.00000	Petal Length (mm)
PetalWidth	150	11.99333	7.62238	1799	1.00000	25.00000	Petal Width (mm)

Pearson Correlation Coefficients, N = 150
Prob > |r| under H0: Rho=0

	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength Sepal Length (mm)	1.00000	-0.11757 0.1519	0.87175 <.0001	0.81794 <.0001
SepalWidth Sepal Width (mm)	-0.11757 0.1519	1.00000	-0.42844 <.0001	-0.36613 <.0001
PetalLength Petal Length (mm)	0.87175 <.0001	-0.42844 <.0001	1.00000	0.96287 <.0001
PetalWidth Petal Width (mm)	0.81794 <.0001	-0.36613 <.0001	0.96287 <.0001	1.00000

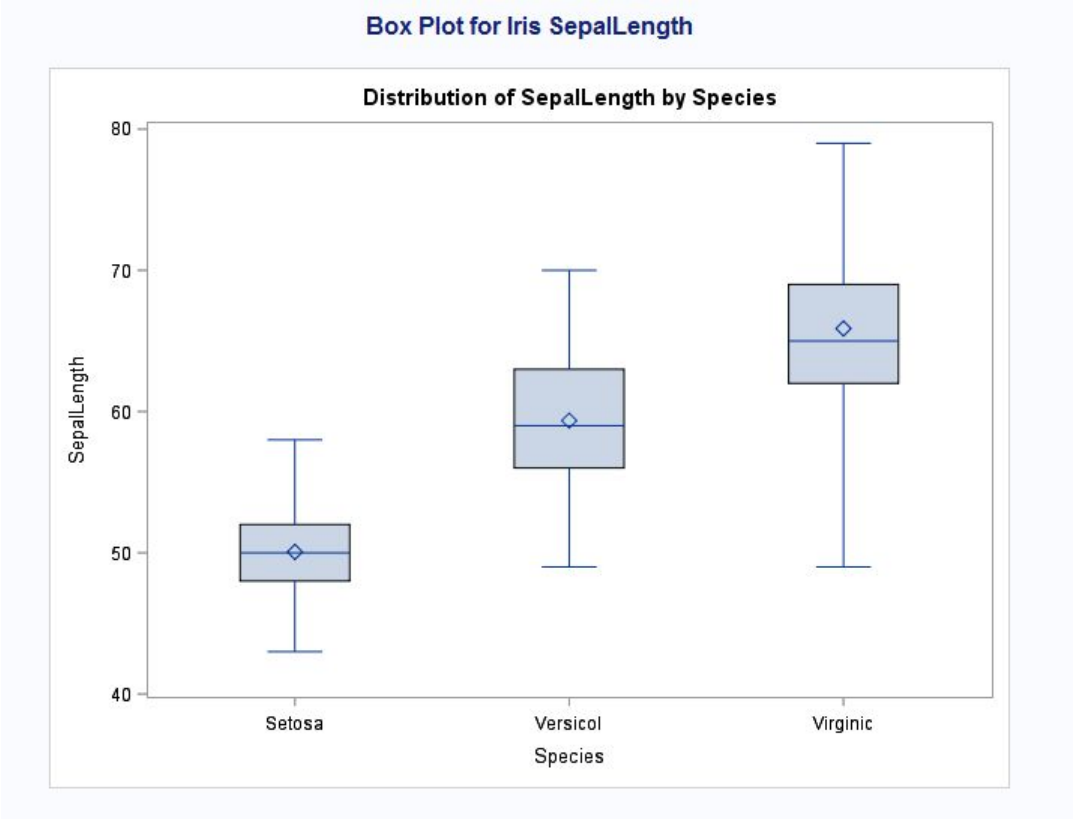
Introduction of SAS – Data Visualization

```
proc univariate data=iris_sas;
        histogram &varname/ normal;
run;
```



Introduction of SAS – Data Visualization

```
proc boxplot data=iris_sas;
        title "Box Plot for Iris SepalLength";
        plot SepalLength*species;
run;
```

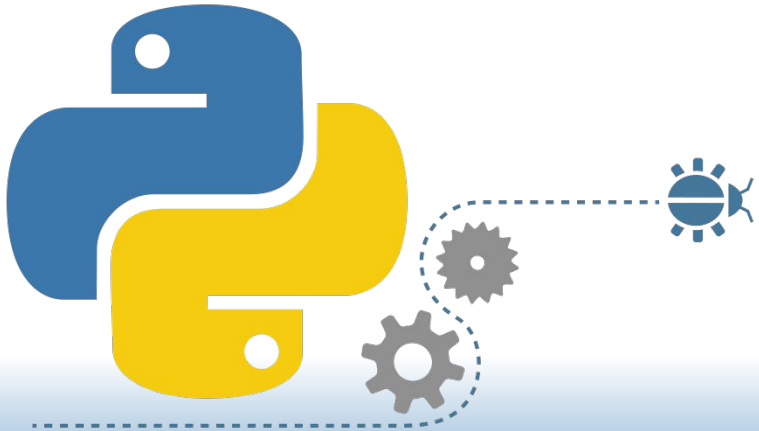


Topic 2.2: Introduction of Python

References:

https://pandas.pydata.org/pandas-docs/stable/index.html

https://seaborn.pydata.org/examples/many_pairwise_correlations.html



Introduction of Python

- Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Read/Write Data Set with Python

- Read CSV to Data Frame:

```
import pandas as pd
iris_df = pd.read_csv("D:/workspace/iris_sas.csv")
```

- Write Data Frame to CSV:

```
iris_df.to_csv("d:/workspace/iris_python.csv")
```

Introduction of Python – View Data

```
In [100]: iris_df.head()

Out[100]:
   Species  SepalLength  SepalWidth  PetalLength  PetalWidth
0  Setosa            50          33           14           2
1  Setosa            46          34           14           3
2  Setosa            46          36           10           2
3  Setosa            51          33           17           5
4  Setosa            55          35           13           2
```

Introduction of Python – Information

```
In [101]: iris_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
Species       150 non-null object
SepalLength    150 non-null int64
SepalWidth     150 non-null int64
PetalLength    150 non-null int64
PetalWidth     150 non-null int64
dtypes: int64(4), object(1)
memory usage: 5.9+ KB
```

```
In [102]: iris_df.info(memory_usage='deep')

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
Species       150 non-null object
SepalLength    150 non-null int64
SepalWidth     150 non-null int64
PetalLength    150 non-null int64
PetalWidth     150 non-null int64
dtypes: int64(4), object(1)
memory usage: 14.3 KB
```

Introduction of Python – Descriptive Statistics

```
In[110]: iris_df.corr()

Out[110]:
             SepalLength  SepalWidth  PetalLength  PetalWidth
SepalLength     1.000000   -0.117570     0.871754    0.817941
SepalWidth     -0.117570    1.000000    -0.428440   -0.366126
PetalLength     0.871754   -0.428440     1.000000    0.962865
PetalWidth      0.817941   -0.366126     0.962865    1.000000
```

Introduction of Python – Descriptive Statistics

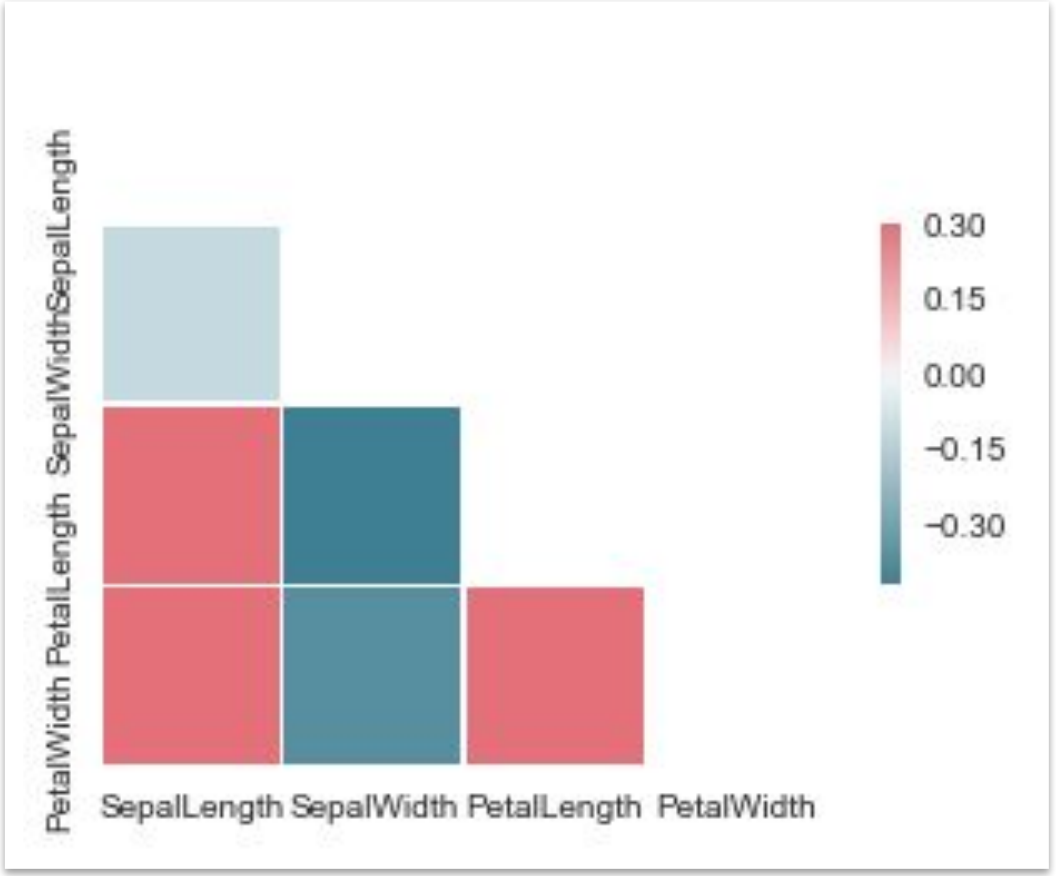
In[112]: iris_df.describe()

Out[112]:

	SepalLength	SepalWidth	PetalLength	PetalWidth
count	150.000000	150.000000	150.000000	150.000000
mean	58.433333	30.573333	37.580000	11.993333
std	8.280661	4.358663	17.652982	7.622377
min	43.000000	20.000000	10.000000	1.000000
25%	51.000000	28.000000	16.000000	3.000000
50%	58.000000	30.000000	43.500000	13.000000
75%	64.000000	33.000000	51.000000	18.000000
max	79.000000	44.000000	69.000000	25.000000

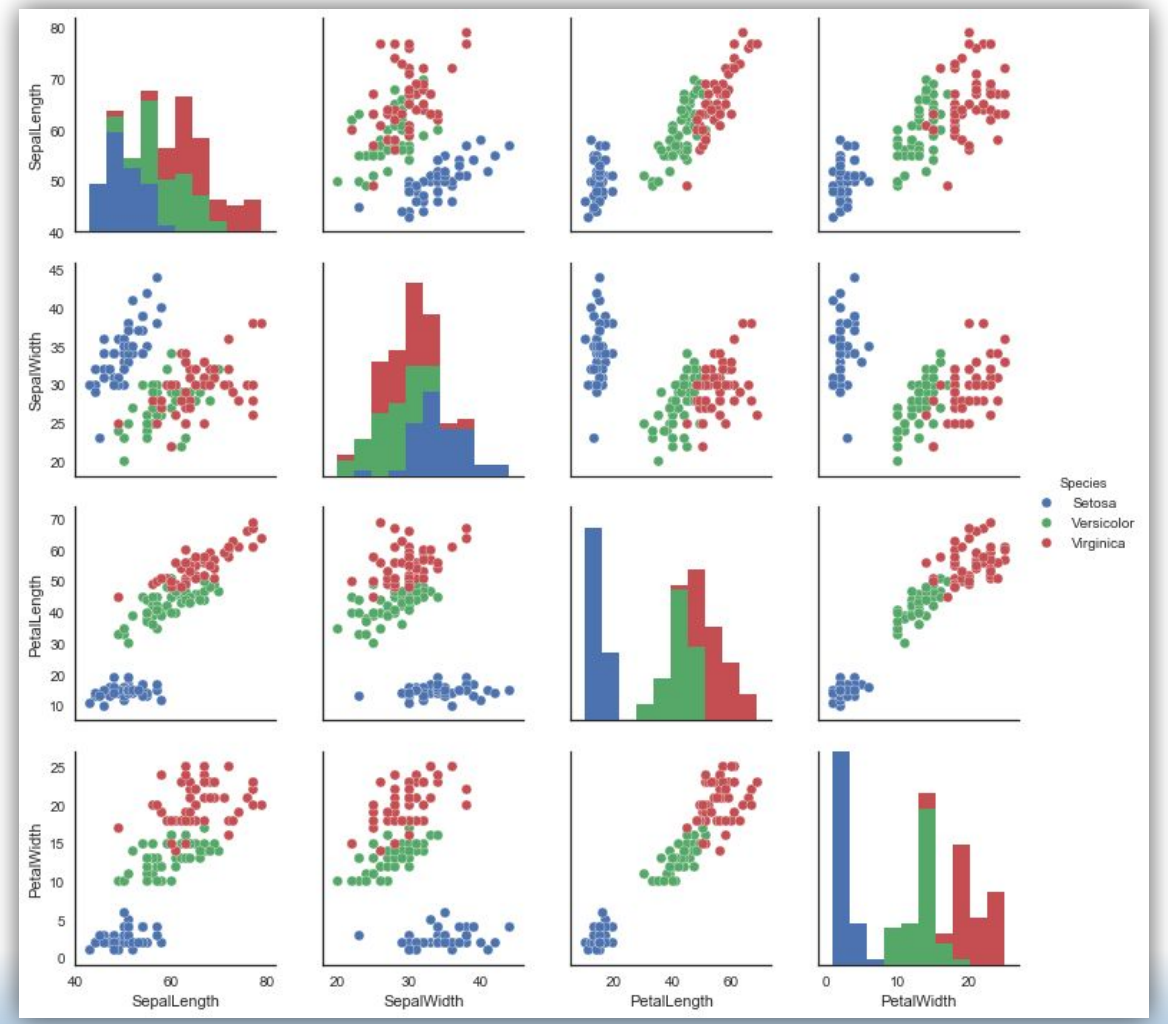
Introduction of Python – Data Visualization

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

corr = iris_df.corr()
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(11, 9))
cmap = sns.diverging_palette(220, 10, as_cmap=True)

sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3,
center=0, square=True, linewidths=.5,
cbar_kws={"shrink": .5})
```



Introduction of Python – Data Visualization

```
import seaborn as sns
g = sns.pairplot(iris_df, hue="Species")
```



Introduction of Python – Data Vistualization

```
import seaborn as sns
ax = sns.boxplot(x="Species", y="SepalLength",
        data=iris_df)
```