Introduction to R

Date: August 22, 2017
Venue: School of Industrial Technology, Universiti Sains Malaysia

Yusri Yusup, PhD
Environmental Technology
School of Industrial Technology
Universiti Sains Malaysia

Why use R?

- R is free and available on many operating systems (OSX, Windows, Linux).
- R has many statistical tools (10000+ packages).
  - “openair”, “ggplot2”, “ggmap", etc.
- R statistical analysis is reproducible.

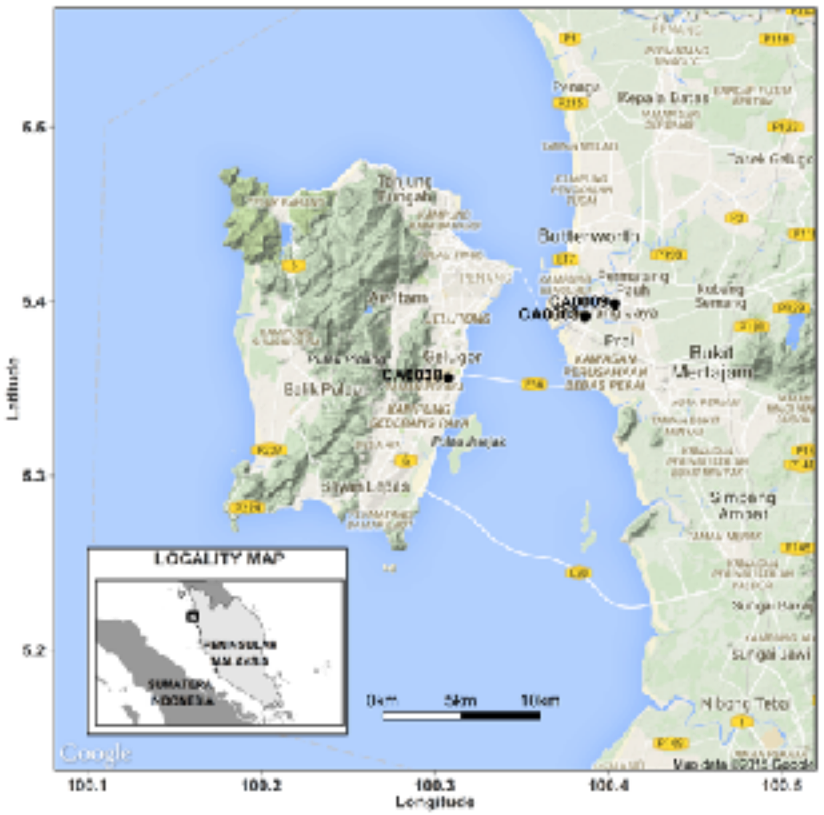
R compared to other commercial softwares

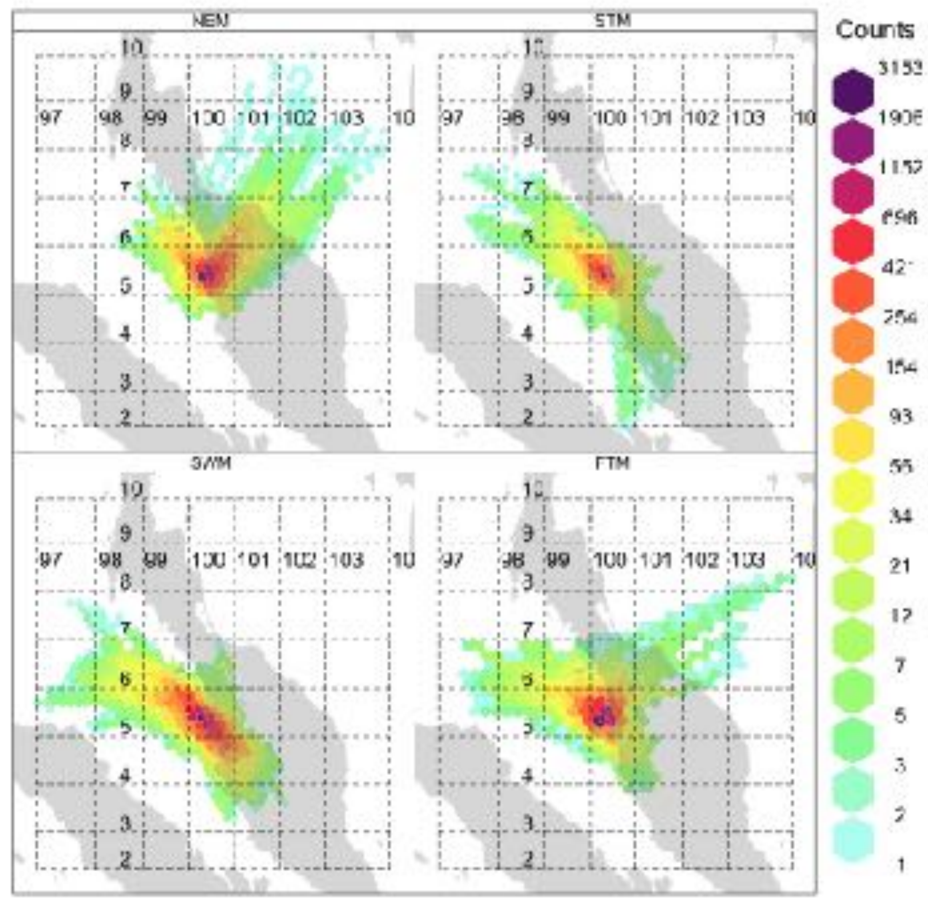
R **Price:** Free **OS:** Available on all OS **Interface:** Command-based **Analyses:** Reproducible **Update:** User-dependent and frequent **Customizable:** High	**Minitab (5 users)** **Price:** RM6000 (RM3500 to update) **OS:** Windows **Interface:** Point-and-click **Analyses:** Not reproducible **Update:** Developer-dependent and infrequent **Customizable:** Low
SPSS (standard) **Price:** RM23000 per year **OS:** Windows, Mac OS, Linux **Interface:** Point-and-click **Analyses:** Not reproducible **Update:** Developer-dependent and infrequent **Customizable:** Low	**SAS** **Price:** RM36400 per year (commercial) **OS:** Windows, Linux, Unix **Interface:** Point-and-click and command-based **Analyses:** Reproducible **Update:** Developer-dependent and in frequent **Customizable:** Low

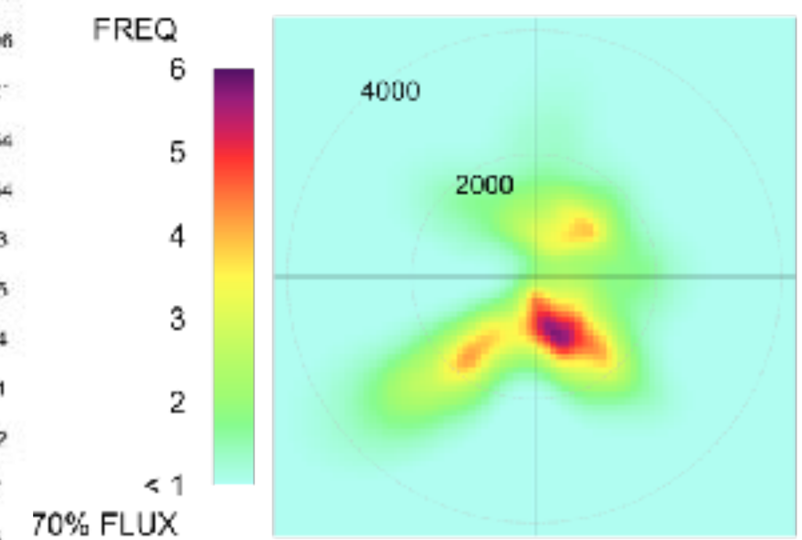
Notable R applications

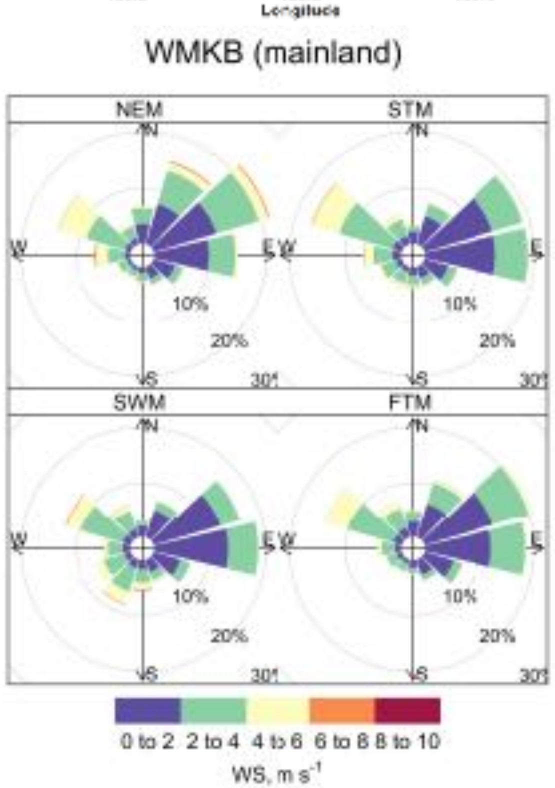
- Genomics - study the structure of genes
  - VERY large sets of data (millions upon millions of data points)
  - exploratory-based research, discovering trends and relationships in large datasets
  - sometimes freely available on the web, waiting for somebody to make discoveries
- Large physical systems - e.g., Earth's atmosphere
  - VERY large data sets available online (MODIS data)

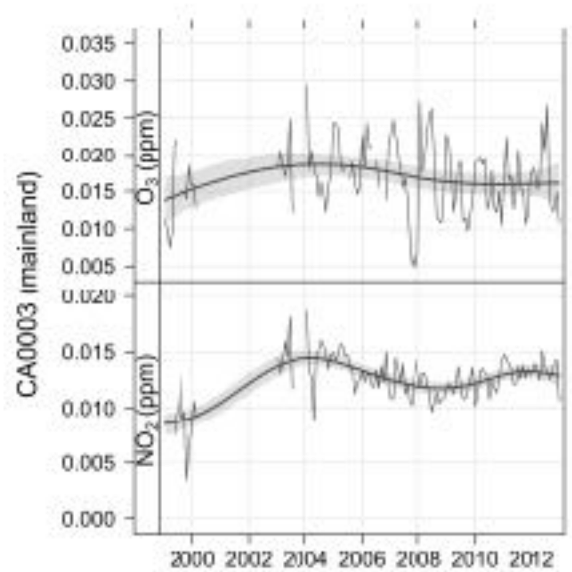
Notable R applications











Course objectives

- To instruct participants on how to use R to manage data.

- To instruct participants on how to use R to analyze data.

- To instruct participants on how to plot figures using R.

Course topics

1. Overview of R
2. Installing and navigating R
3. Data management (with some plotting)
4. Descriptive statistics
5. Correlation and regression
6. Plot maps
7. openair

Topic 1: Overview of R

Learning outcome

At the end of this topic, the participant will be able to:

- explain what R is about.

Topic 1: Overview of R

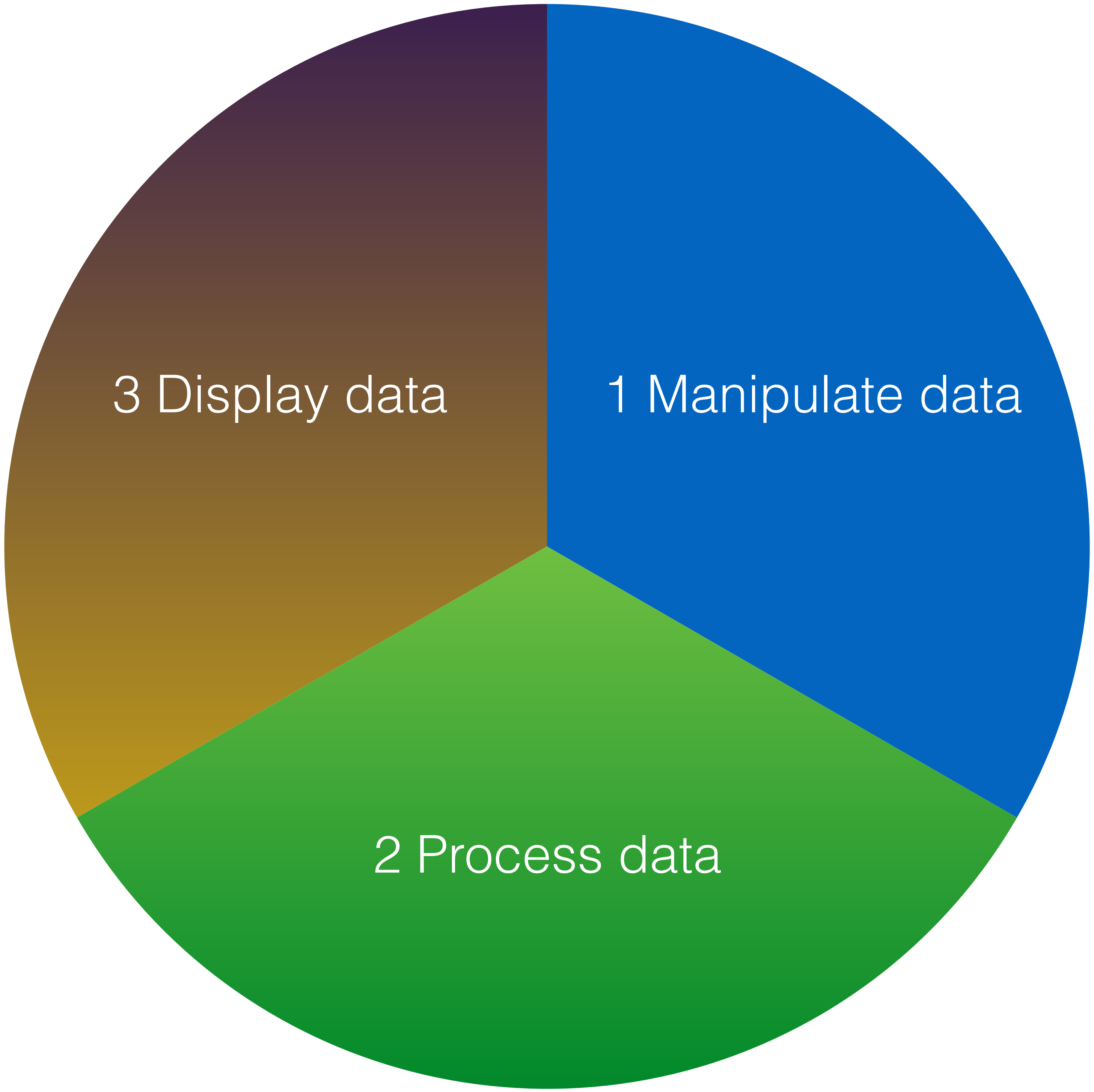
- R is a <u>robust</u> data analysis tool
- R is open source (FREE!!!)
- R is popular

  "R is the [most popular language for data scientists]() — and it's been around for almost 20 years — and so by sheer force of numbers and time, R has more extensions than any other data science software. R is the primary tool used for statistical research: when new methods are developed, they're not just published as a paper — they're also published as an R package. That means R is always at the cutting edge of new methodologies." - Revolutions, Microsoft (2017)

- Command-based interface makes it easy to document and reproduce the data analysis method
- Large online user community (stackoverflow.com and you can use Google to search)

Topic 1: Overview of R

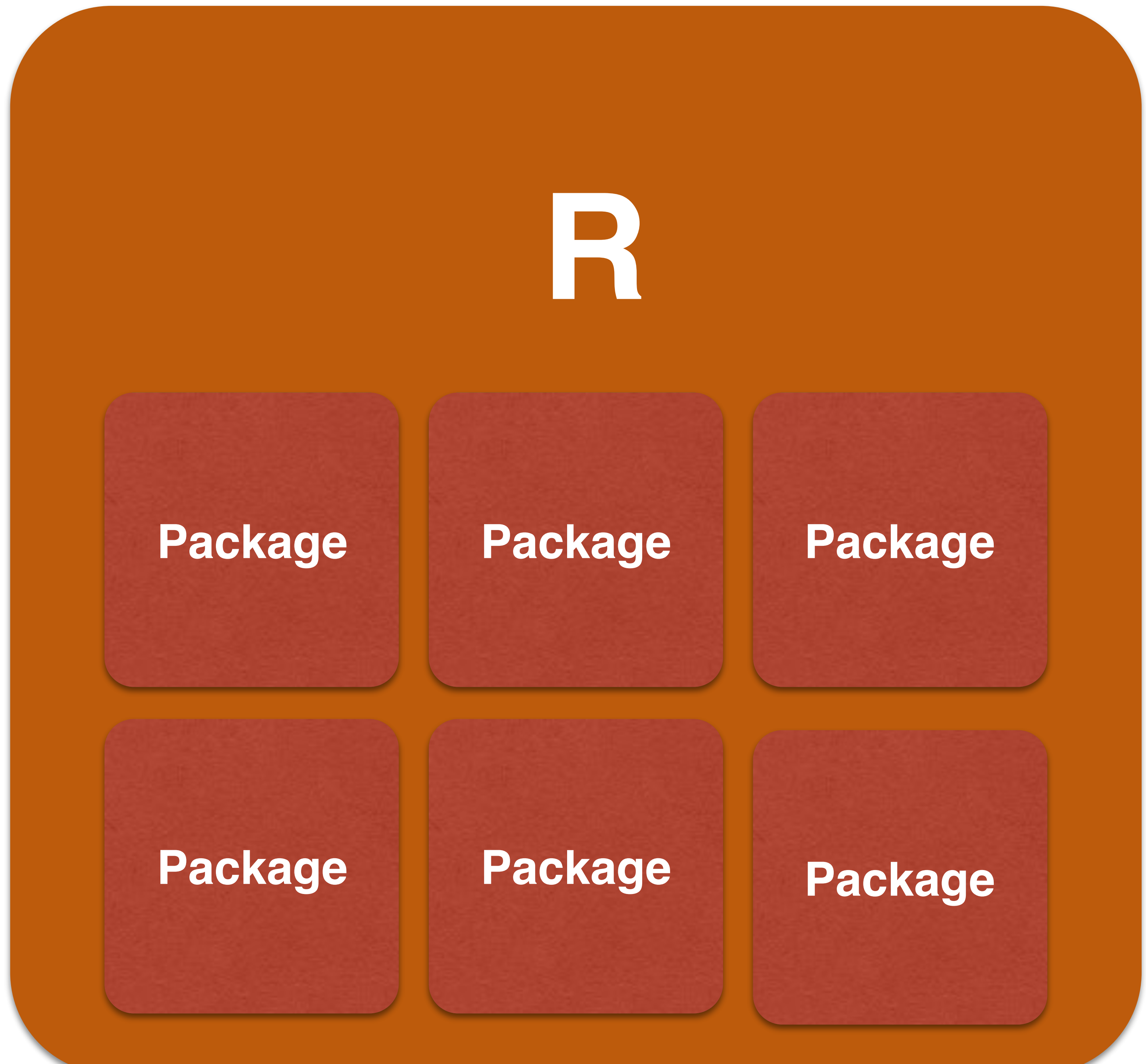
What can you do in R?



Topic 1: Overview of R

- R consists of about 25 standard/base packages

- Other packages (>10000) available to download within R or RStudio



Resources

- You can download R's notes at: http://cran.r-project.org/doc/manuals/R-intro.pdf

- You can download a brief intro of R at: http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

Citing R in Research Papers

- The programmers of R ask that any analysis done using R be cited as:

  R Development Core Team (2017). **R: A Language and Environment for Statistical Computing**. Vienna, Austria: R Foundation for Statistical Computing.

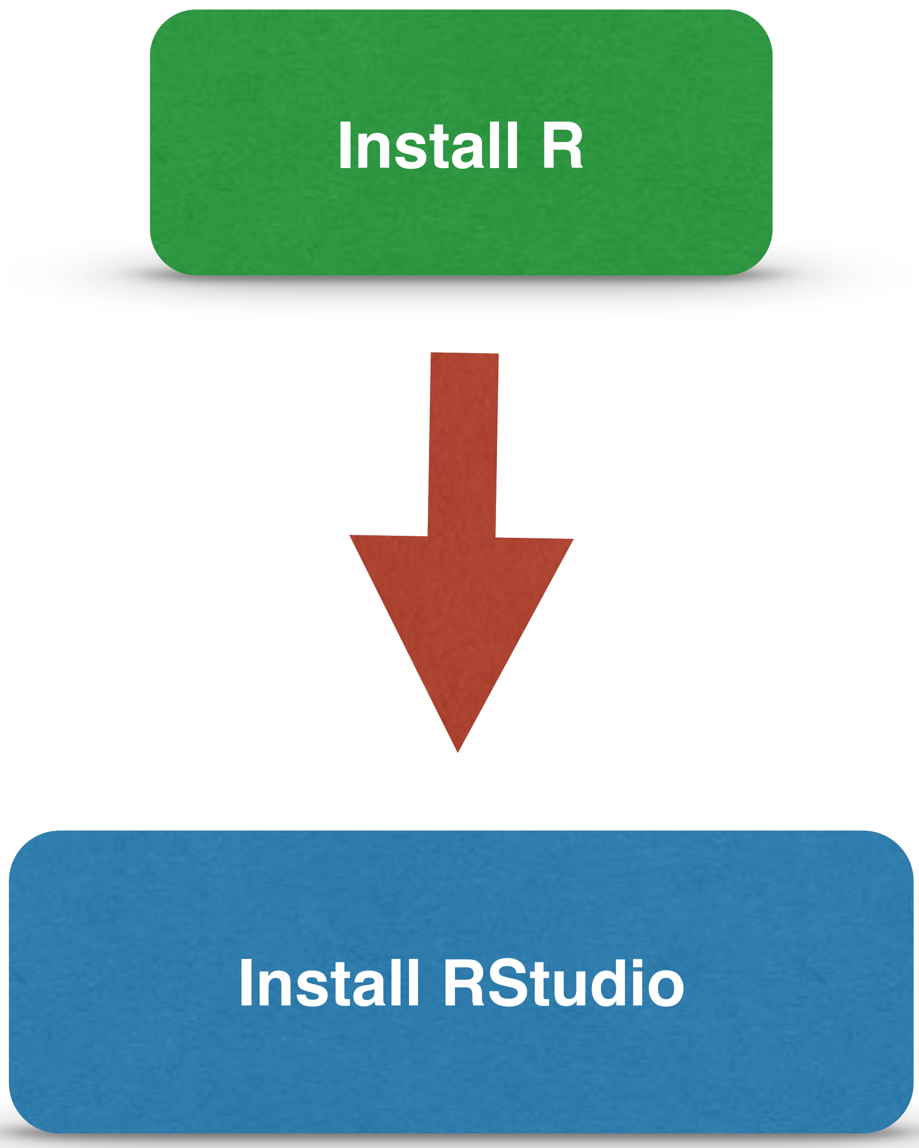
Topic 2: Installing and Navigating R

Learning outcome

At the end of this topic, the participant will be able to:

- install and navigate RStudio.

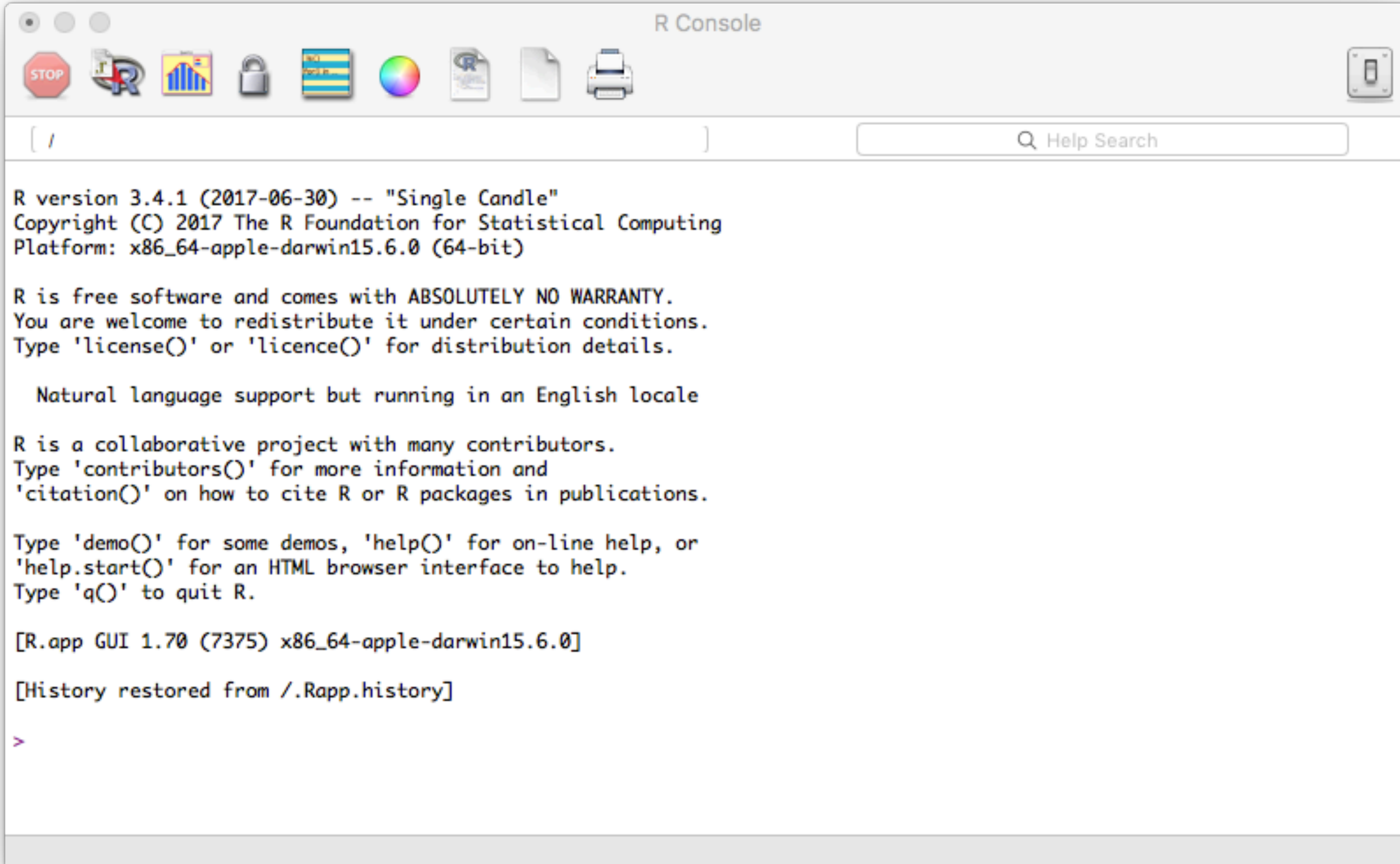
Topic 2: Installing and Navigating RStudio



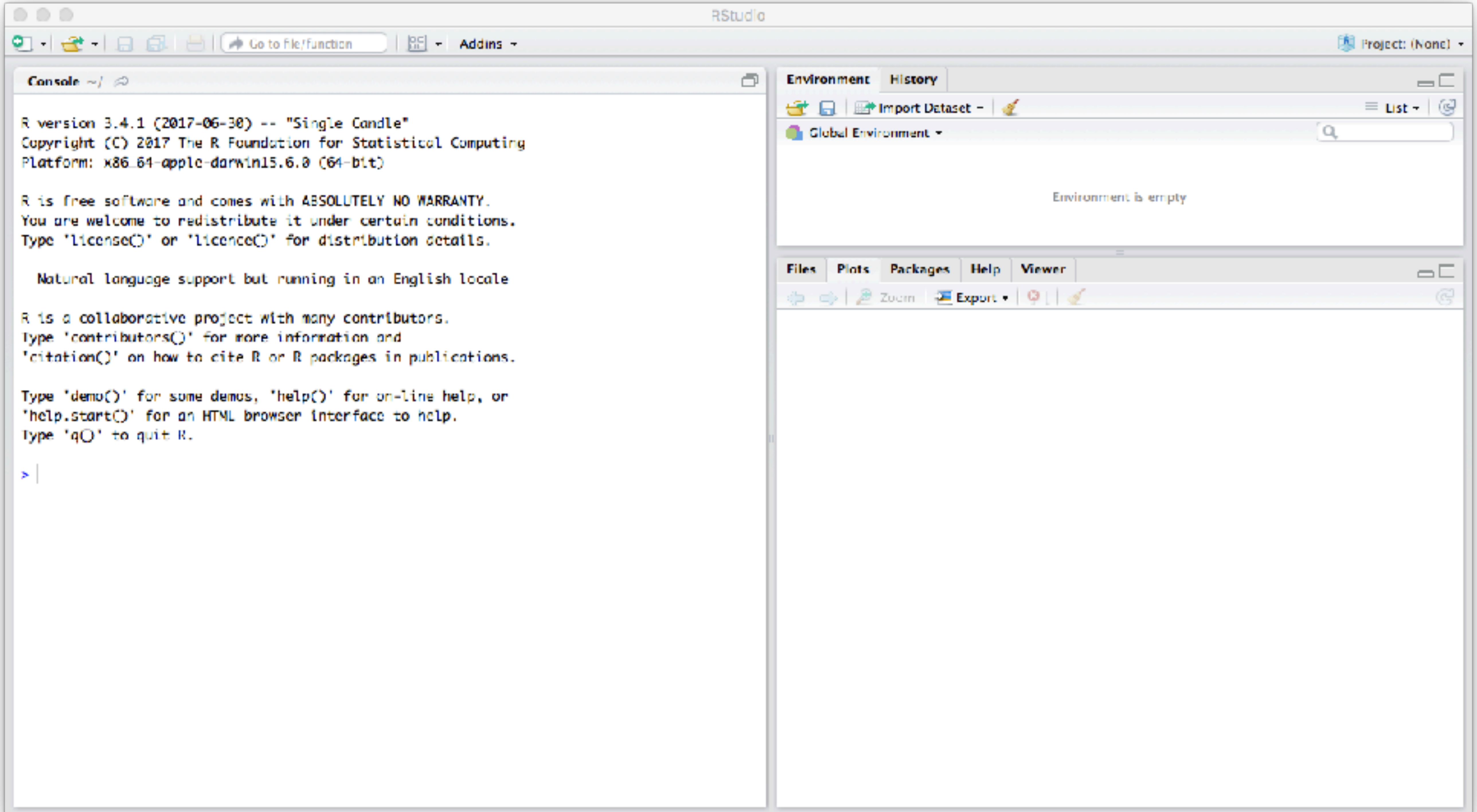
Download R ver. 3.4.1 from
http://cran.r-project.org/bin/windows/base/

Download RStudio ver. 1.0.153 from
https://www.rstudio.com/products/rstudio/download2/

Run R


R Console
Help Search
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[R.app GUI 1.70 (7375) x86_64-apple-darwin15.6.0]
[History restored from /.Rapp.history]
>

Run RStudio


RStudio
Go to File/function
Addins
Project: (None)
Console ~/
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Zoom
Export

Topic 2: Installing and Navigating RStudio

- You can determine the R version by looking at the console.

- You can determine the RStudio version by clicking on About RStudio

- Create new **R scripts** by clicking on **File > New File > R Script**. You can create custom analyses using scripts.

- The other popular way to communicate your R analysis is by using **R Markdown**.

    - For a working example, visit atmosfera.usm.my

Topic 2: Installing and Navigating RStudio

- If your console gets messy, you can clear it by pressing **‘ctrl + L’**.

- You can import data by point-and-click by clicking on the **‘Import Dataset’** button in the **‘Environment’** panel.

- You can export your plots in the lower left panel and clicking on the button **‘Export’**.

- You can view help on functions by typing **‘?’** in front of the function in the **‘console’**. Example: ?mean

Topic 3: Data management (and some plotting)

<u>Learning outcome</u>

At the end of this topic, the participant will be able to:

- input or generate data into RStudio.
- import data.
- differentiate the types of data classes used in R.
- use ‘functions’.
- plot histogram, barplot, time series, pie chart.
- modify plots.

Project Management

- My recommendations on how to start a **data analysis project**:
  - Create a **main folder**
  - Create **subfolders**:
    - **data** - to house all your data
    - **R** - to store all your scripts
    - **figs** - to store all your generated figures
    - **docs** - to keep any relevant documents
- Next level: **version control** - e.g. GitHub (after you are familiar with R), example: http://yusriy.github.io/R_stat_analysis/

Topic 3: Data management (and some plotting)

- R uses command-based user interface. Command prompt is “>”
- Insert values into variables by using the “arrow” operator (<-) or equal operator“=“
- Variables are case-sensitive
- Try,

```
> x <- 3

> y = 4

> data <- c(1,2,3,4)

> list_of_data <- 1:20 #Create a sequence from -1 to 20 with interval of 1

> data2 <- seq(from=0,to=5,by=0.5) #Create a sequence with interval of 0.5
```

Topic 3: Data management (and some plotting)

- There are many different “functions” in R, some of them only available in installed “packages”

- Create a 2 by 2 with element 1, 2, 3, 4

```
> matrix_A <- matrix(c(1,2,3,4),2,2)
```

Topic 3: Data management (and some plotting)

- Find out more about the function by using the symbol ‘?’ like ?matrix

- Functions can be used by inserting “arguments” into “()”. There could more than 1 argument and sometimes return a value.

- In the case of the *matrix* function, the value return is the matrix itself.

Function: `ls()`

- Type `ls()`.
- This function would list all the variables in the workspace
- There are different data types
  - logical: TRUE, FALSE
  - character/string: a, b, c, computer, statistics, research
  - **numeric: 0.2, -1.0, 101325.2 (default setting)**
  - integer: -1, 0, 3, 4, -1201
  - atomic (same as vector, matrix): [2, 3], [1,4], [1, 2; 3, 4]
- `class()` can be used to determine the class of the the variable

Function: `rm()`

- `rm()` can be used to remove variables from the workspace
- Example,

```
> rm(a, matrix_a)

> rm(list=ls()) #delete all variables in the workspace
```

Popular plots in R

- Many plotting options
  - `graphics` - default graphics device
  - `ggplot` - used in “`ggmap`”
  - `lattice` - used in “`openair`”
- Histogram (`graphics`)
- Bar (`graphics`)
- Time series (`graphics`)
- Pie chart (`graphics`)

Hands-on 1: Distribution Analysis (graphics)

Defective products distribution analysis
For thirty days, a factory concerned about the quality of its products collected data on the number of defective products returned by customers. Using R, construct a frequency distribution chart and analyze its distribution.

Data: ho1_data.txt

17	24	28	27	23	22	26	21	29	19
33	27	19	26	25	18	25	26	24	25
21	15	30	31	25	16	25	19	23	29

Hands-on 1 Solution: Distribution Analysis (graphics)

Solution:

<u>1. Import data from text file</u>

Ensure your working directory is the same as the data file.

```
> getwd()
```

```
> ho1_data <- read.table(“data/ho1_data.txt”,header = TRUE)
```

Note: You can import data from directories other than the working directory by using the argument in read.table “file”. “Header” is an argument to tell read.table that the data file contains headers (default is FALSE). Try leaving “header” as FALSE, what would happen?

```
> ho1_data <- read.table(file=“/Desktop/ho1_data.txt”,header =
TRUE)
```

Hands-on 1 Solution: Distribution Analysis (graphics)

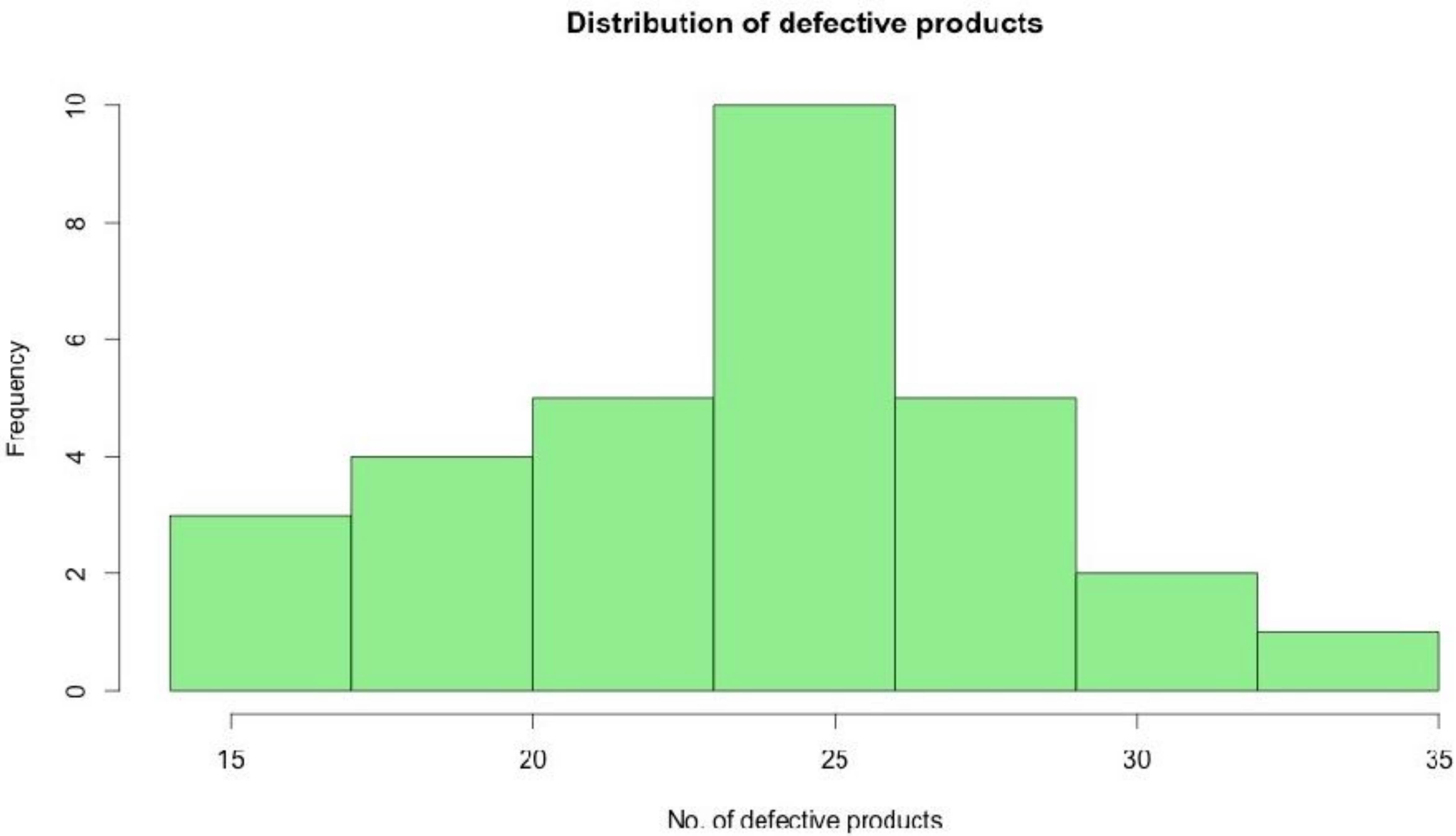
2. Plot the distribution

```
> hist_info<-
hist(ho1_data$no_defect,breaks=seq(14,35,by=3),xla
b="No. of defective products",main="Distribution
of defective products",col='lightgreen')

> hist_info
```

You can see classes, counts, etc. by typing `hist_info` because you assigned the info of the histogram into it by using the operator `<-`.

In this case you have specified the classes by looking at the data first and setting `breaks` and using the function `seq()`.

Hands-on 1 Solution: Distribution Analysis (graphics)



Hands-on 2:
Barplot (graphics)

Heavy metals in water

A study was conducted to determine the average concentration of heavy metals (in mg/L) in water. Construct a bar plot based on the following data.

<u>Data:</u> ho2_data.xlsx

Cu	Pb	Zn	Cd	Cr
0.048	0.306	0.091	0.171	0.115

Data courtesy of School of Industrial Technology, USM

Hands-on 2 Solution: Barplot (graphics)

Solution:

1. Import data

```
> ho2_data <- read.csv(‘data/ho2_data.csv)
```

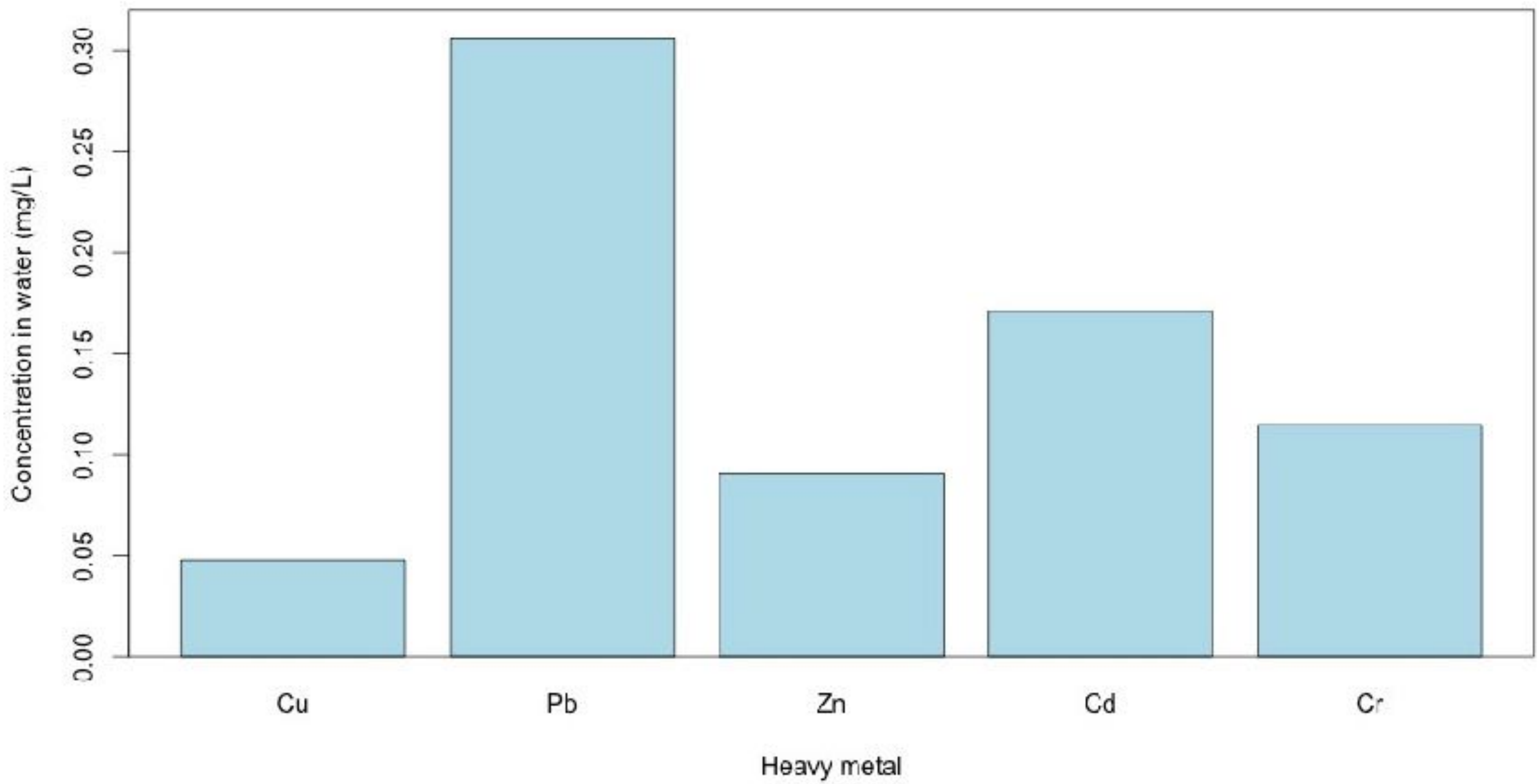
You can rename the headers of the data frame by using the following commands:

```
> names(ho2_data)[1] <- “metal”

> names(ho2_data)[2] <- “concentration”
```

2. Draw the bar plot

```
> barplot(ho2_data$concentration,xlab="Heavy
metal",ylab="Concentration in water (mg/
L)",names.arg=ho2_data$metal,ylim=c(0,0.32),col='lightblue')
```

Hands-on 2 Solution: Barplot (graphics)



Hands-on 3: Time Series (graphics)

Starch hydrolysis of noodles

An experiment was run to study the changes in hydrolysis of starch (%) in noodles over a period of 3 hours. Two types of noodles were used: one as a control (without banana flour) and the other with Cavendish peel flour (i.e., banana flour) or GCPe. Construct a time series plot.

Data: ho3_data.xlsx

Time (min)	0	30	60	90	120	150	180
Control	0	8.495	11.038	15.239	20.887	20.839	20.909
GCPe	0	5.720	8.803	11.640	12.902	12.826	12.896

Data courtesy of School of Industrial Technology, USM

Hands-on 3 Solution: Time Series (graphics)

Solution:

1. Import the data

```
> ho3_data<-read.csv('data/ho3_data.csv')
```

2. Plot the time series

```
> plot(ho3_data$Time..min.,ho3_data$Control,type='o',xlab='Time (min)',ylab='Hydrolysis (%)',axes=FALSE)

> lines(ho3_data$Time..min.,ho3_data$GCPe,type='o',pch=19,col='blue')
```

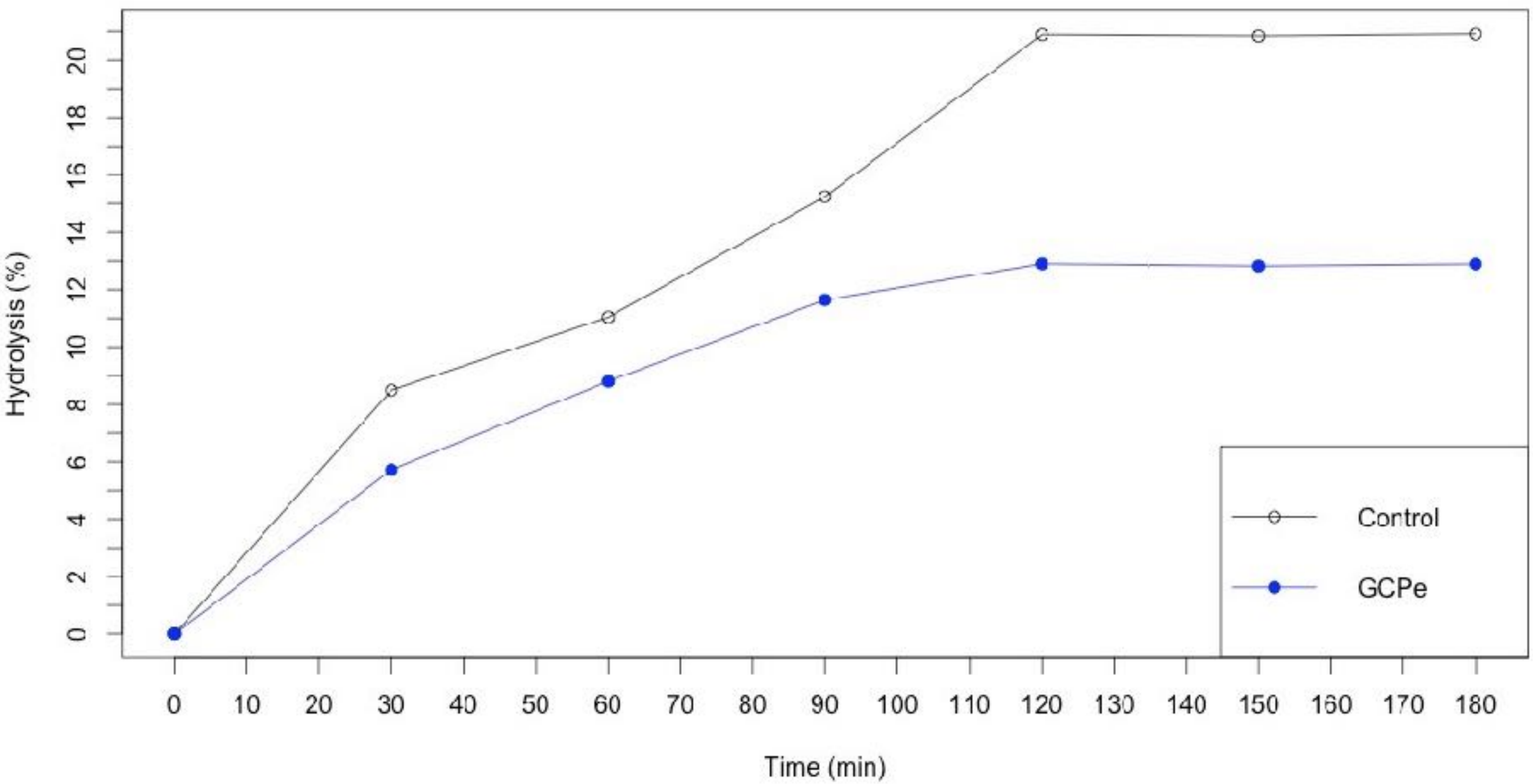
Note #1: `type='o'` *is to set the type of lines plotted onto the figure,* `'o'` *is 'overplotted', type* `?plot` *in the R console to see other types of lines.*

Note #2: `axes=FALSE` *is used so that you can control how R plots the axes.*

Note #3: You can overlap plots by plotting the initial data and then plotting over this original plot with the next set of data using `lines()` *or* `points()`*. Caution that the default axes follows the initial data plotted.*

```
> legend("bottomright",c("Control","GCPe"),lty=c(1,1),pch=c(1,19),col=c('black','blue'))

> axis(1,at=seq(0,200,by=10))

> axis(2,at=seq(0,30,by=1))

> box()
```

Hands-on 3 Solution: Time Series (graphics)



Hands-on 4:
Pie Charts (graphics)

Pie Charts in R

Construct a pie chart from the following data:

<u>Data:</u>

Class	Frequency
Not good	5
Good	11
Very good	8
Total	**24**

Hands-on 4 Solution: Pie Charts (graphics)

Solution:

1. Input data into R

```
> ho4_data<-data.frame(c('Not Good’, 'Good','Very Good'),c(5,11,8))

> names(ho4_data)<-c(‘Class’,'Frequency')
```

Note: You can name the headers of the data frame by using the function `names()`*.*

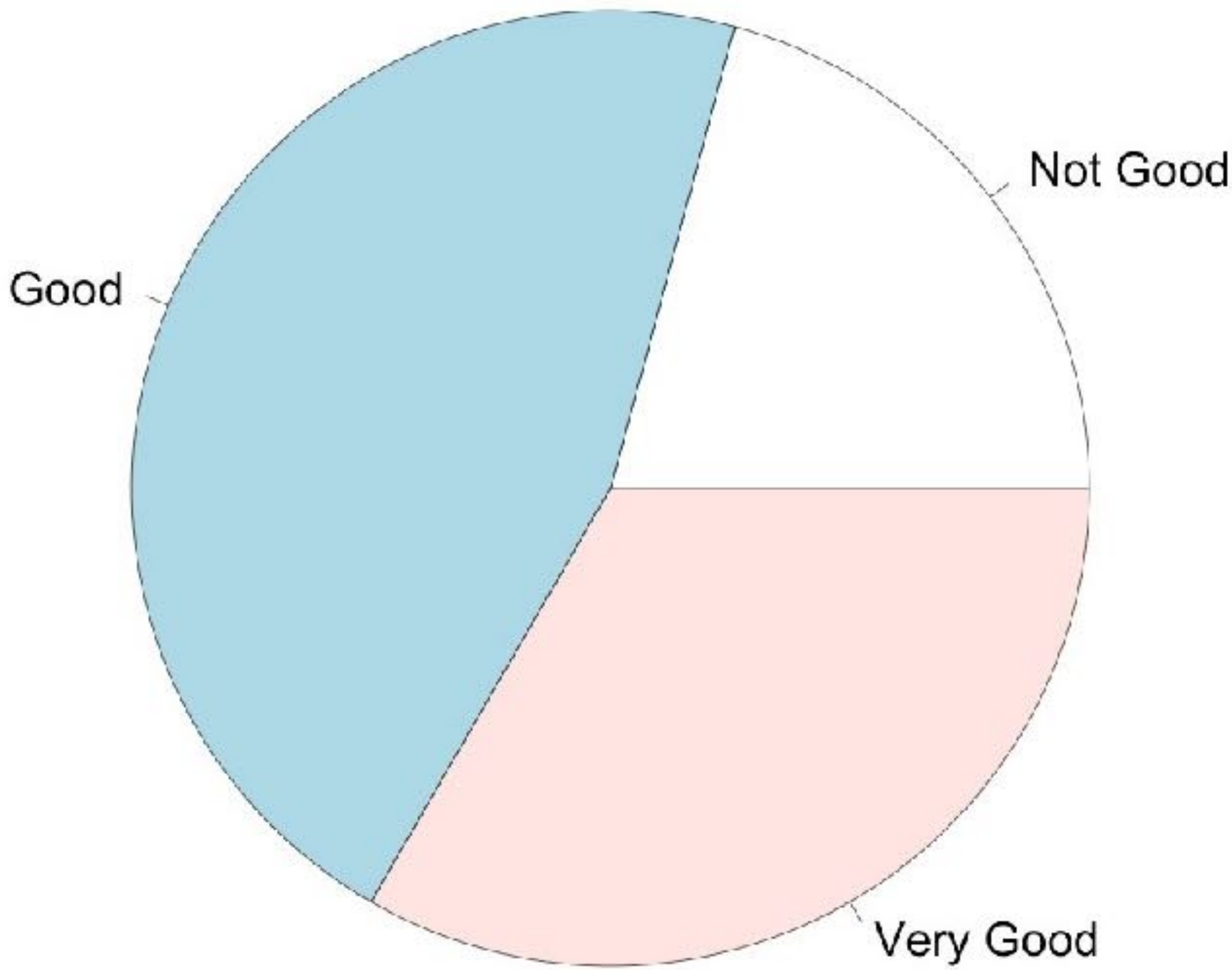
2. Plot the pie chart

```
> par(mar=c(0.01,0.01,0.01,0.01)) # To adjust the margins

> pie(ho4_data$Frequency,ho4_data$Class,cex=2)
```

Note: You can change the margins of the plots using the function `par()` *and the argument mar. The first numeric item is the bottom margin, left, top, and right margin sequentially.*

Note: `cex = 2` *because to increase the labels so that it is clearer if you want to increase the size of the pie chart.*

Hands-on 4 Solution: Pie Charts (graphics)



Exporting Plots

- Point-and-click method (less pretty)
  - In the “plots” tab, click “Export” and then save.
- Code method (recommended, looks professional)
  - See Hands-on 3 for an example.

Topic 4: Descriptive statistics

<u>Learning outcome</u>

At the end of this topic, the participant will be able to:

- describe data using descriptive statistics in R.

Topic 4: Mean, median, mode, variance, and standard deviation

- You can use R to conduct any statistical analyses you require.
- We will start with the basics: mean, median, mode, variance, and standard deviation.
- Functions you will use are:
  - `mean()`
  - `median()`
  - `table()`, `names()`, and `max()`
  - `var()`
  - `sd()`

Hands-on 6: Mean pH

Mean pH of green banana pulp
The pH of green banana pulp is an important physico-chemical parameter. The pH data obtained from 12 different samples are listed below. Find the mean pH of the 12 samples.

Data:

4.49	4.37	4.75	5.64	4.73	4.59
4.54	5.37	5.58	5.65	5.53	5.47

Data courtesy of School of Industrial Technology, USM

Hands-on 6 Solution: Mean pH

Solution:

1. Input data

```
> ho6_data <-
c(4.49,4.37,4.75,5.64,4.73,4.59,4.54,5.37,5.58
,5.65,5.53,5.47)
```

2. Calculate the mean

```
> x <- mean(ho6_data)
```

Note: If you want to use the mean in further calculations, then you can assign the value to another variable such as `x` *above.*

Hands-on 7: Median

Determining median from a dataset

The following dataset has 9 data points (odd number of data points). Determine the median of this dataset using R.

<u>Data:</u>

0.20	0.50	0.51	0.53	0.67	0.70	0.78	0.78	0.81

Hands-on 7 Solution: Median

Solution:

1. Input the data

```
> ho7_data<-
c(0.20,0.50,0.51,0.53,0.67,0.70,0.78,0.78,0.81
)
```

2. Calculate the median

```
> MD <- median(ho7_data)
```

Note: If the number of data points is even, then R would average the two middle numbers to obtain the median.

Hands-on 8: Mode and `table()`

Mode of particulate matter (PM1) at a palm oil mill
The following data represent the amount of PM1 (μg/m^3) in air at a Penang palm oil mill. Find the mode.

Data:

58.19	67.31	120.28	67.36	80.25	108.76	74.08	40.61	30.96	108.64

Data courtesy of School of Industrial Technology, USM

Hands-on 8 Solution: Mode and `table()`

Solution:

1. Import the data

```
> ho8_data <- read.csv('ho8_data.csv')
```

Note: csv stands for 'comma-separated values' a quite common format.

2. Find the mode

```
> table(ho8_data$pm1)
```

Note #1: `table()` *would group the data according to number of instances. You can determine the mode from the group with the highest number of instances.*

Note #2: In this example, since there is no group with the highest number of instances, then there is no mode for this dataset. Trying another case where there is a mode, let's add another value where we will create a dataset with a mode.

```
> test_data <- c(ho8_data$pm1,108.64)

> test_data <- table(test_data)

> mode <- names(test_data)[test_data==max(test_data)]
```

You can see that the class "108.64" has "2" has its number of instances and thus this class is the mode. We use `names()` *here because the mode class is the header of the highest number of instances. We use* `max()` *to find the maximum number of instances.*

Hands-on 9: Range

Range of Mg concentration in water

The following data are the Mg concentration in water. Find the range of Mg concentration in water.

Data: ho9_data.csv

10.53	37.4	16.8	37.785	20.37	30.95	15.135	32.28	42.46	8.255
17.145	13.895	4.35	16.125	9.35	25.26	15.45	4.08	7.86	9.745

Data courtesy of School of Industrial Technology, USM

Hands-on 9 Solution: Range

Solution:

1. Import the data

```
> ho9_data <- read.csv('ho9_data.csv')
```

2. Calculate the range

```
> temp_data <- range(ho9_data$Mg)

> mg_range <- temp_data[2] - temp_data[1]
```

Note: `range()` *will create a variable with two values: the lowest and the highest values of the dataset. To obtain the difference between these two values, i.e., the range, you have to minus them such as shown above.*

Hands-on 10:
Variance and Standard Deviation

Variance and standard deviation of Mg in water
Using the Mg data before, calculate the variance and standard deviation of this dataset.

Data: ho9_data.csv

Hands-on 10 Solution: Variance and standard deviation

Solution:

1. Calculate variance

```
> Mg_var <- var(ho9_data$Mg)
```

2. Calculate standard deviation

```
> Mg_std <- sd(ho9_data$Mg)
```

Note: There are a number of options when calculating variance and standard deviations, the same goes for other functions as well. You can take a look at these options by typing `?var` *or* `?sd`*.*

Topic 5: Correlation and regression

Learning outcome

At the end of this topic, the participant will be able to:

- calculate correlation coefficient.
- conduct regression analyses.
- conduct multi-variable linear regression.

Hands-on 19: Correlation coefficient

Relationship between copper and cadmium in sediment

The concentration of copper and cadmium in sediment is shown below. Construct a scatter diagram between the two metal concentrations and calculate the correlation coefficient.

<u>Data:</u> ho19_data.csv

Cu (mg/L)	0.63	0.73	0.35	0.76	0.6	0.36	0.63	0.52	0.55	0.47
Cd (mg/L)	1.95	1.99	1.94	1.98	1.94	1.95	1.98	1.93	1.97	1.92

Data courtesy of School of Industrial Technology, USM

Hands-on 19 Solution: Correlation coefficient

Solution:

1. Import data

```
> ho19_data <- read.csv('data/ho19_data.csv', sep = ',', header = TRUE)

> names(ho19_data)<-c('Cu','Cd')
```

2. Plot cadmium versus copper concentration

```
> plot(ho19_data$Cu,ho19_data$Cd,xlab='Cu (mg/L)',ylab='Cd (mg/
L)',pch=19,col='red')
```

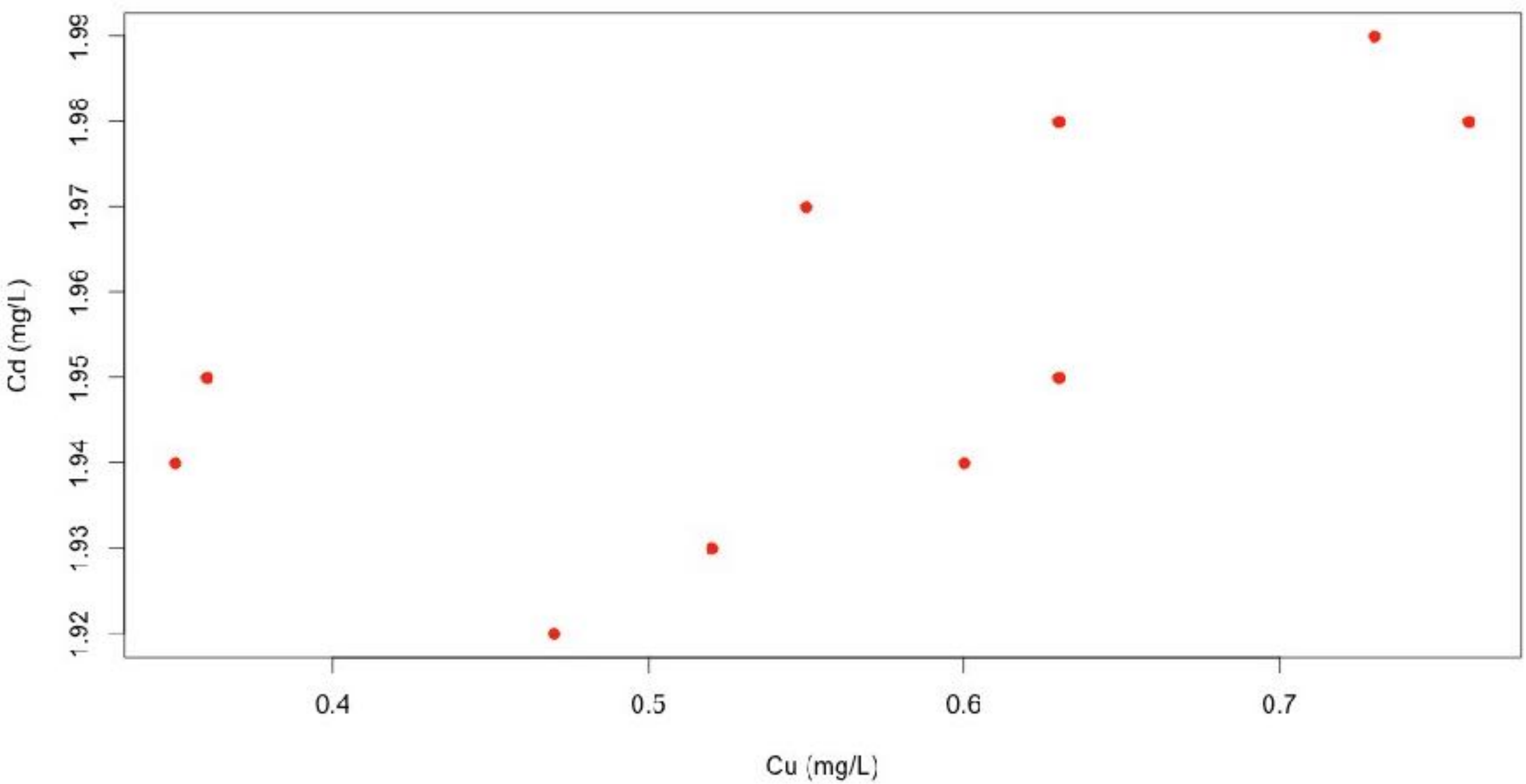
The plot shows a linear relationship between copper and cadmium concentrations.

3. Calculate the correlation coefficient, r

```
> r_cu_cd <- cor(ho19_data$Cu,ho19_data$Cd)

> r_cu_cd

[1] 0.6709394
```

$$r = \frac{n \sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{\sqrt{[n \sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2]\,[n \sum_{i=1}^{n} Y_i^2 - (\sum_{i=1}^{n} Y_i)^2]}}$$

Hands-on 19 Solution: Correlation coefficient



Hands-on 20:
Effect of one variable to another

Effect of FRAP and total phenolic content of date palm fruits

An experiment was carried out to analyze the edible parts of date palm fruits for their antioxidant activities using a ferric reducing/antioxidant method (FRAP). The objective of the study is to determine the effect of FRAP on total phenolic content (TPC). The results are given below.

Data: ho20_data.csv

FRAP (X)	20.00	26.93	16.00	13.32	29.34	11.66	19.12
TPC (Y)	2.71	4.8	2.23	1.6	4.4	2.19	3.23

Data courtesy of School of Industrial Technology, USM

Hands-on 20 Solution: Effect of one variable to another

Solution:

1. Import the data

```
> ho20_data <- read.csv('data/ho20_data.xlsx')
```

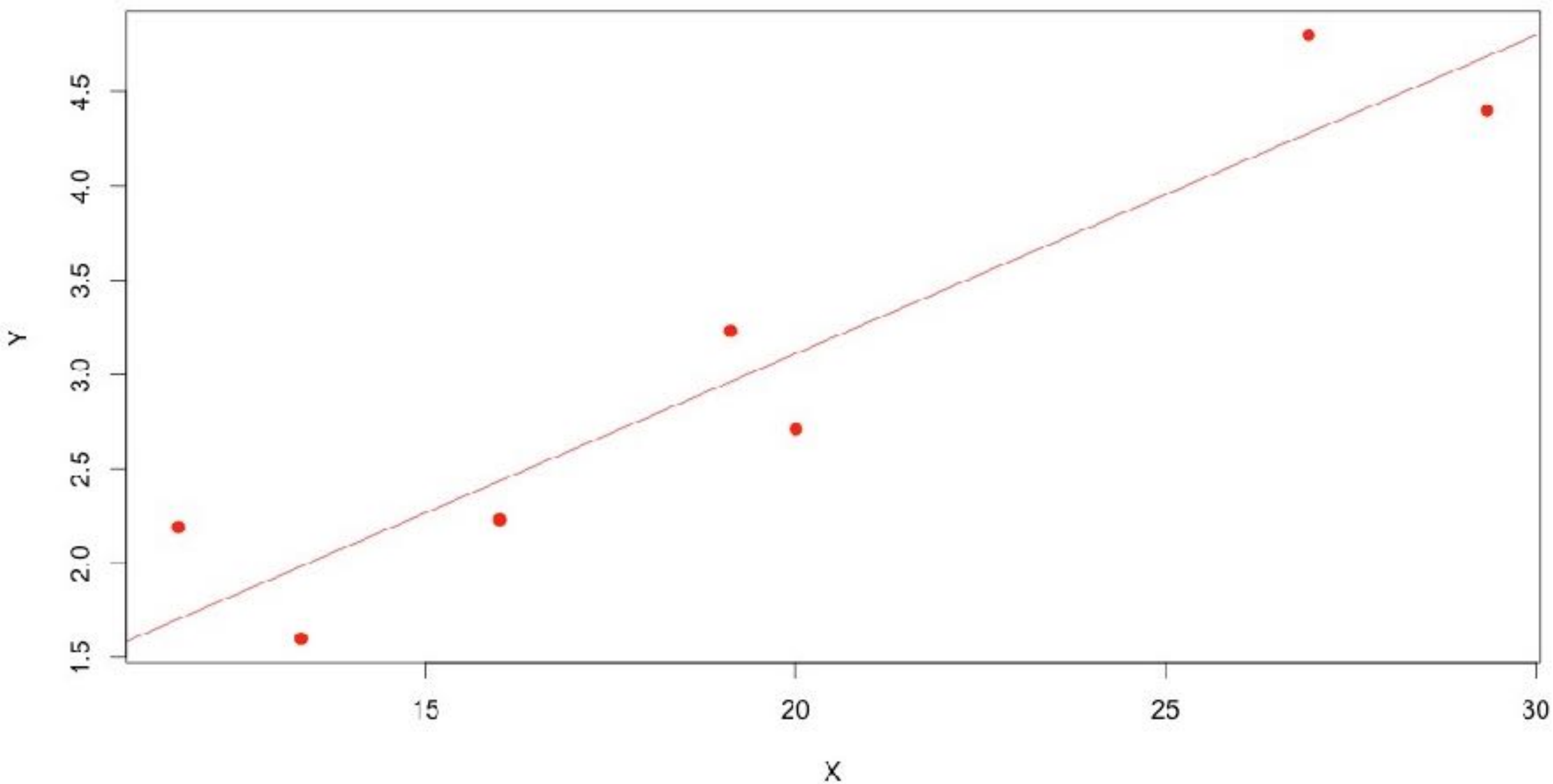
2. Calculate the linear regression coefficient

```
> lm_Y_X <- lm(Y ~ X, data = ho20_data)

> lm_Y_X

Call:

lm(formula = Y ~ X, data = ho20_data)


Coefficients:

(Intercept)            X

    -0.2655       0.1688
```

Using normal notations, b_0 = -0.2655 and b_1 = 0.1688 from the general equation Y = b_0 + b_1 X. The linear regression equation is Y = -0.2655 + 0.1688 X.

The constant b_1 is positive, which indicates that Y (FRAP) affects X (TPC) positively, if FRAP increases by 1 unit, TPC would also increase by 0.1688 units

Hands-on 20 Solution: Effect of one variable to another



Hands-on 21: Multi-variable linear regression

Indoor air quality and physical properties
The relationship between multi-factor scores and toluene metabolite concentrations (TDI) was studied to understand the behavior of indoor air components at different polyurethane factories. The data for 2 independent variables: relative humidity (RH, %) and dry bulb temperature (T_d, °C) and 1 dependent variable, TDI ($\mu g/m^3$), were collected.

Data: ho21_data.csv

TDI (Y)	81	79	78	76	75	59	58	57	55	53
RH (X1)	50	50	51	51	53	40	40	40	41	43
T_d (X2)	35	35	33	33	33	30	28	28	27	27

Data courtesy of School of Industrial Technology, USM

Hands-on 21: Multi-variable linear regression

Solution:

1. Import the data

```
> ho21_data <- read.csv('data/ho21_data.csv')
```

2. Perform multi-variable linear regression

```
> lmTDI <- lm(TDI ~ RH + TD,data=ho21_data)

> lmTDI


Call:

lm(formula = TDI ~ RH + TD, data = ho21_data)


Coefficients:

(Intercept)           RH           TD

   -40.2601       0.5842       2.6066
```

The multi-variable linear regression equation is TDI = -40.2601 + 0.5842 RH + 2.6066 TD (or Y = -402601 + 0.5842 X1 + 2.6066 X2)

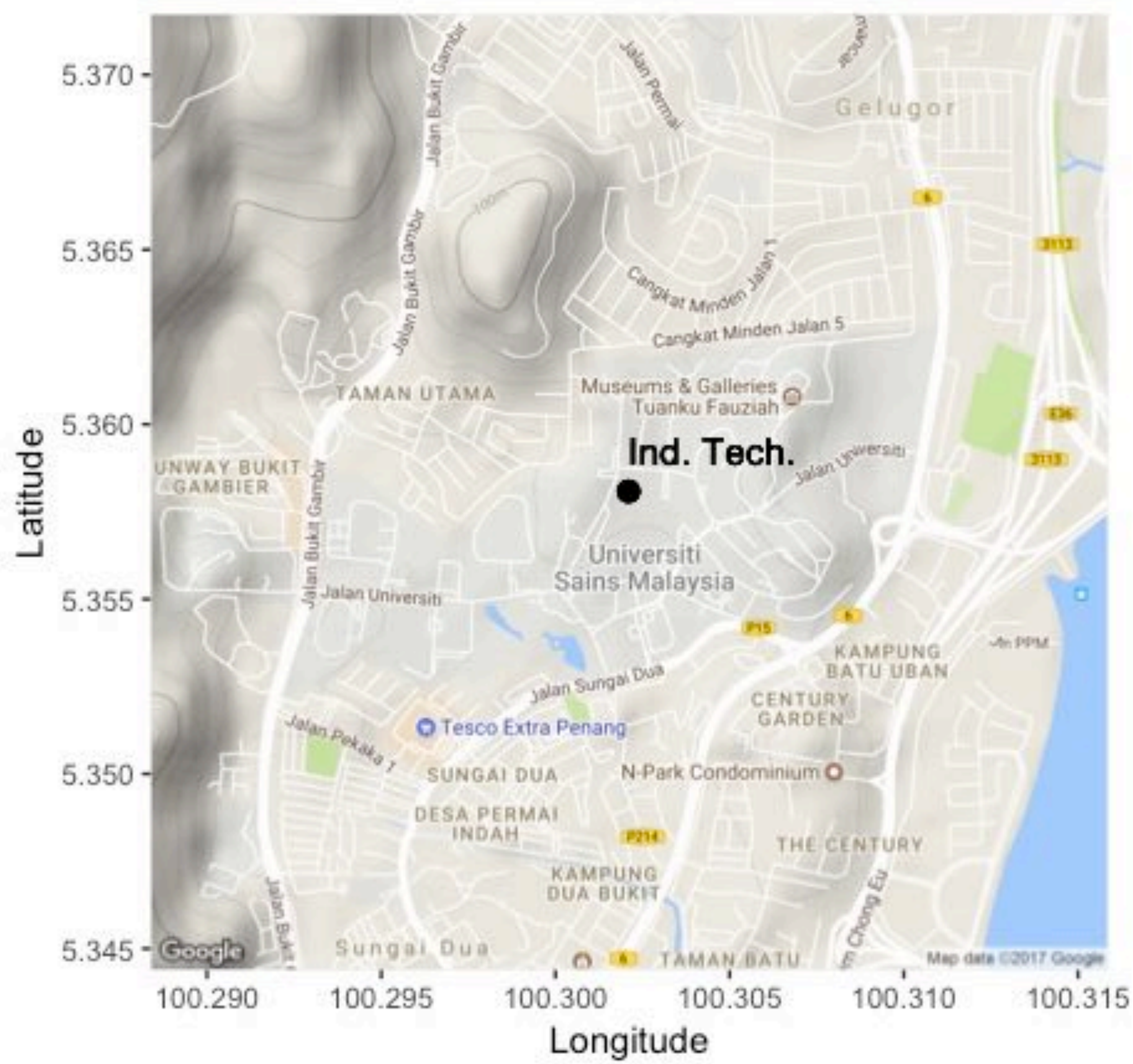
Plotting Maps (ggplot)

```
# Plotting Maps

#install packages (if necessary)

install.packages("ggmap")

#loading libraries

library(ggmap)

#download map from google

sitemap <- get_googlemap(center = c(lon =
100.302100,lat = 5.358086), sensor=TRUE, size =
c(640,640), scale = 2, zoom = 15, maptype = "terrain")
```

Plotting Maps (ggplot)

```
#create map
map_plot <- ggmap(sitemap) +
  geom_point(aes_string(x = "100.302100",y = "5.358086"),size = 3,
             shape = 16,colour = "black") +

geom_text(aes_string(x=“100.302100",y="5.358086"),label="Ind.Tech."
,
            colour="black",size=4,
            fontface="plain",hjust=0,vjust=-1.00) +
  xlab("Longitude") + ylab("Latitude")
#plot map
map_plot
```

Plotting Maps (ggplot)

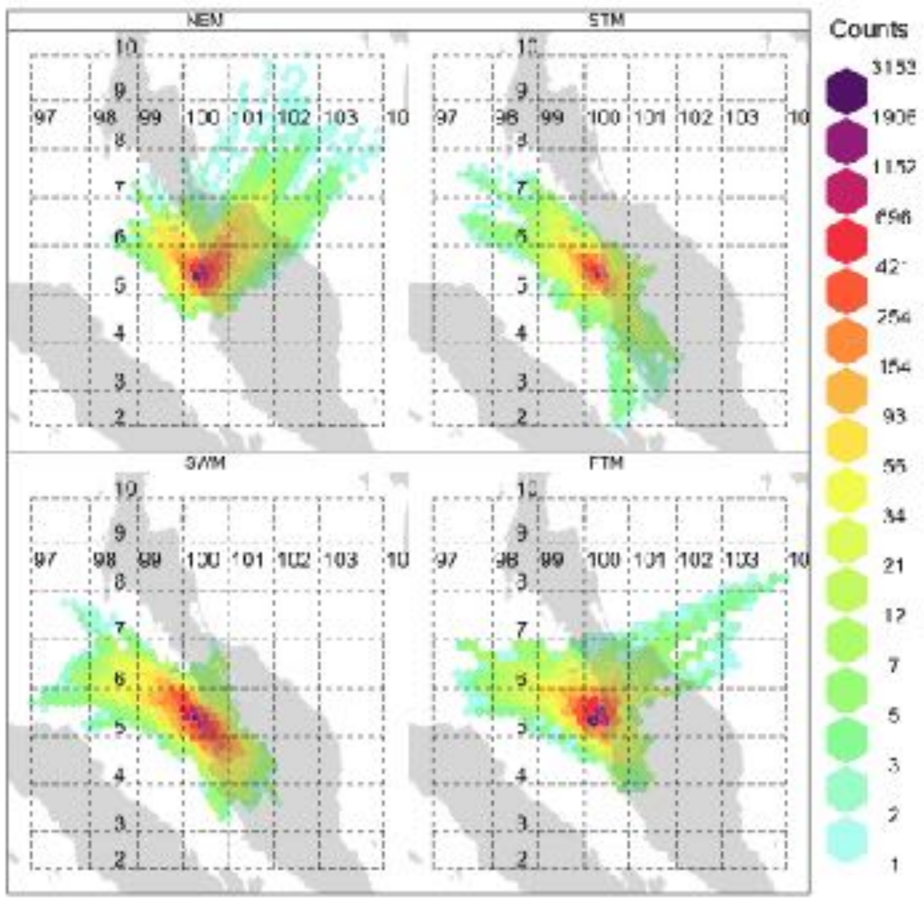


Intro to openair (lattice)

- The package openair is a very useful package for air pollution research and data analysis in general.

- The `timeAverage` function can average/summarize the data based on time scales, e.g., hourly, daily, monthly, etc.

- Can run trajectory analysis (HYSPLIT) through R (complicated, not within this scope)



Intro to openair (lattice)

```
# Intro to openair
# Install package
#install.packages('openair')
# Load package
library(openair)
# Load sample data
data(mydata)
attach(mydata)
# Averaging
daily_avg <- timeAverage(mydata, avg.time = 'day')
halfhour_avg <- timeAverage(mydata, avg.time = '30 min')
```

Intro to openair (lattice)

```
# Plot windrose

windRose(mydata)

# Time series plot

timePlot(mydata, pollutant = 'pm10')

# Time variation plot

timeVariation(mydata, pollutant = 'pm10')

# Polar plot

polarPlot(mydata, pollutant = 'pm10')
```

ICERT

International Conference on Environmental Research and Technology

END

Enjoy ICERT 2017!