

APCC 2016 Statistical Downscaling Training Program Quick R start

August 2016

Dr. Moosup Kim



About Lecturer

- Name: Moosup Kim
- Research Fellow in APEC Climate Center
- Ph. D. in Statistics
- Mission: Statistical Downscaling
- Email: moosupkim@apcc21.org



Contents

- Preparation
- Variables and Operations
- Data Analysis using R
- Reading your Observation Data



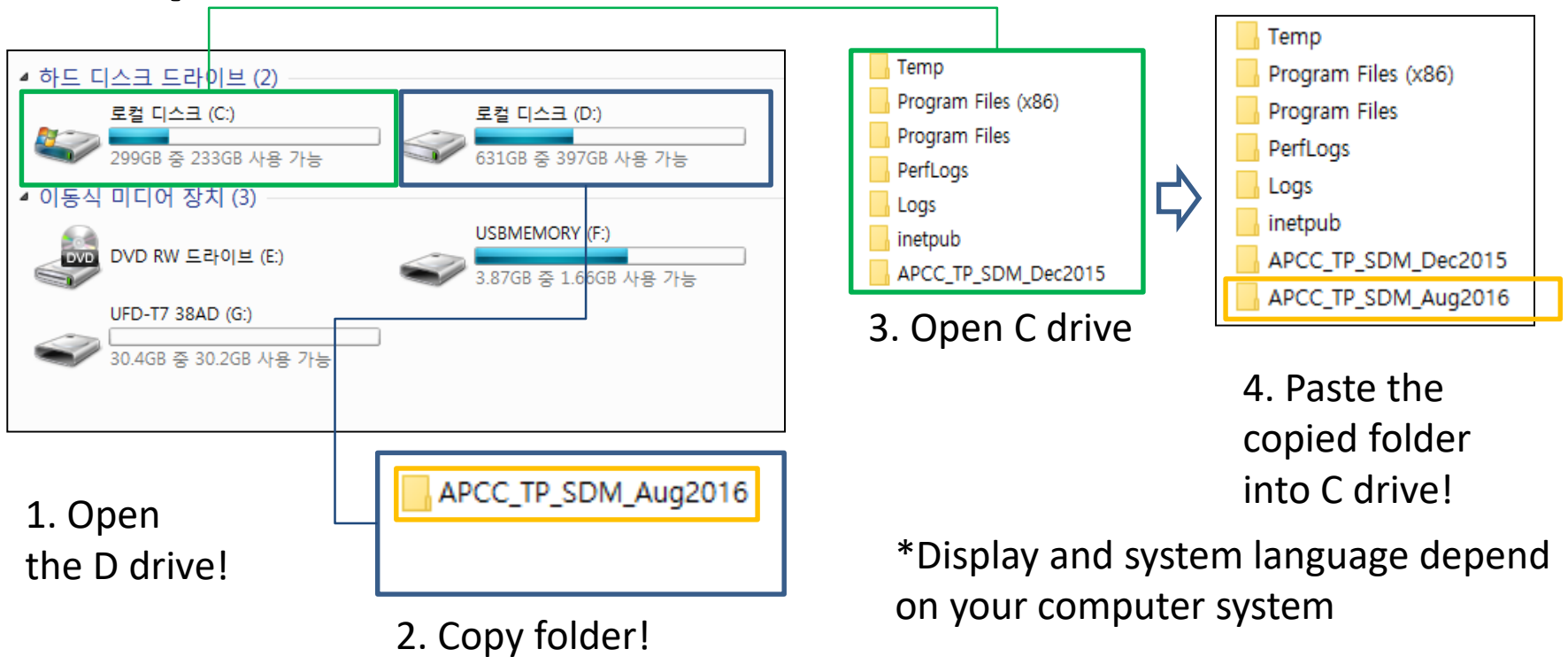
Contents

- Preparation
- Variables and Operations
- Data Analysis using R
- Reading your Observation Data



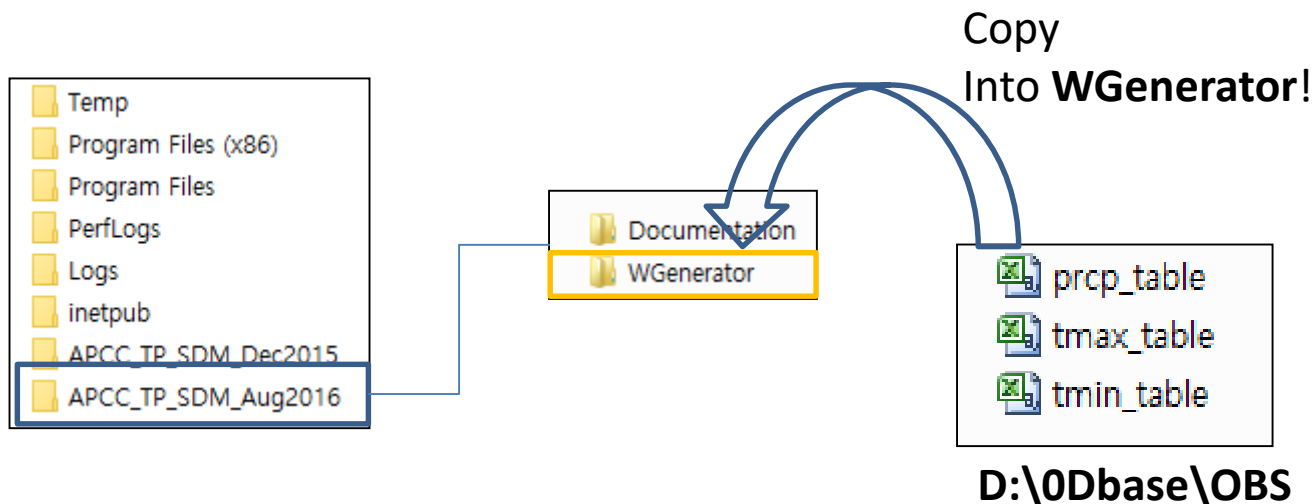
Preparation

- Copy D:\1Kim\APCC_TP_SDM_Aug2016 into C:\.



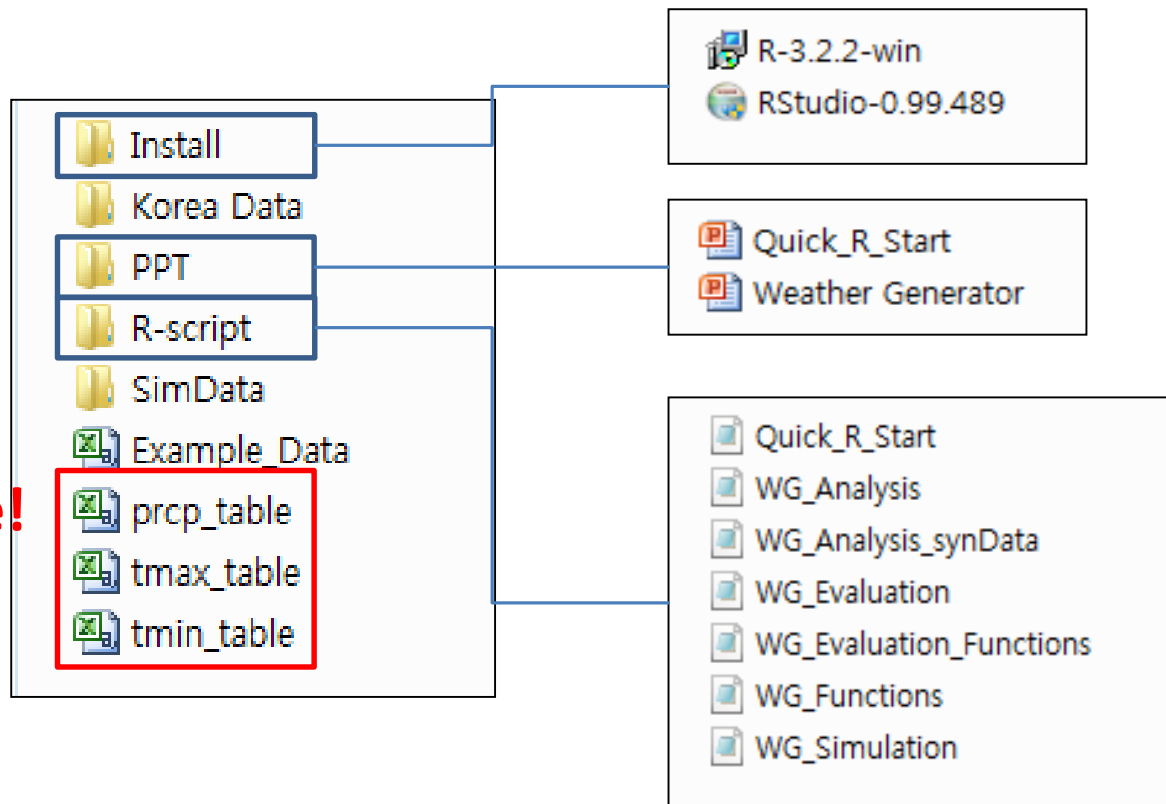
Preparation

- Copy your observation data into **C:\APCC_TP_SDM_Aug2016\WGenerator**
- At **D:\0Dbase\OBS**,
prcp_table, tmax_table, tmin_table.



Preparation

- **APCC_TP_SDM_Aug2016/WGenerator**



**Your data sets.
Check the file name!
(all small letters)
prcp_table
tmax_table
tmin_table**



Preparation

- Check your data files!
 - Data file format is CSV.
 - First line shows names: YEAR, MONTH, DAY, rests are site names.

Check spelling and capital/small letters of the names!

Data have to be sorted in date (YEAR,MONTH,DAY)!!

No missing in YEAR, MONTH,DAY!!

All capital letters!

Site names check!

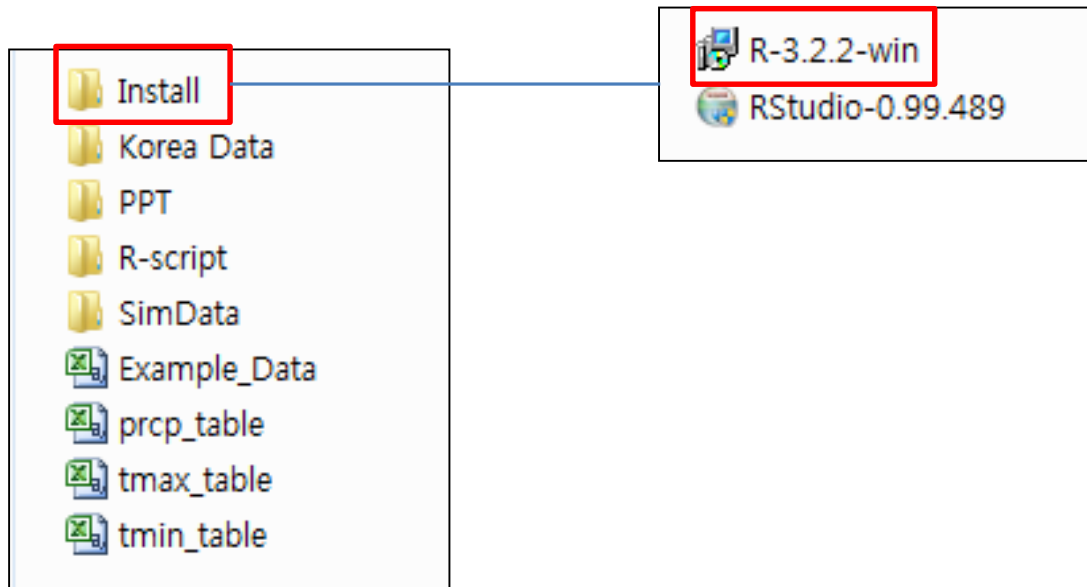
	A	B	C	D	E	F	G	H	I
1	YEAR	MONTH	DAY	Andong	Bongwhoa	Tawbaekk			
2	1988	1	1	0	0	0			
3	1988	1	2	0	0	0			
4	1988	1	3	0	0	0			
5	1988	1	4	1	0.3	0.2			
6	1988	1	5	0	0	0			
7	1988	1	6	0	0	0			
8	1988	1	7	0	0	0			
9	1988	1	8	0.2	0.8	1.6			
10	1988	1	9	0	0	0			
11	1988	1	10	0	0	0			
12	1988	1	11	0	0	0			
13	1988	1	12	0	0	0			
14	1988	1	13	0	0	0			
15	1988	1	14	0.2	0	0			

Display depends on your data.



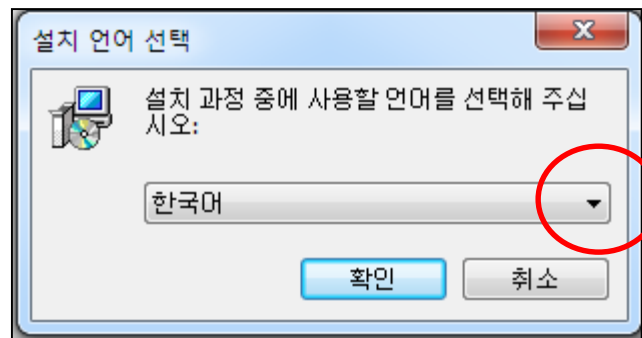
Preparation

- Install R
 - Open **Install** folder
 - Double click **R-3.2.2-win** (Start to install)



Preparation

- Install R
 - Select your language.



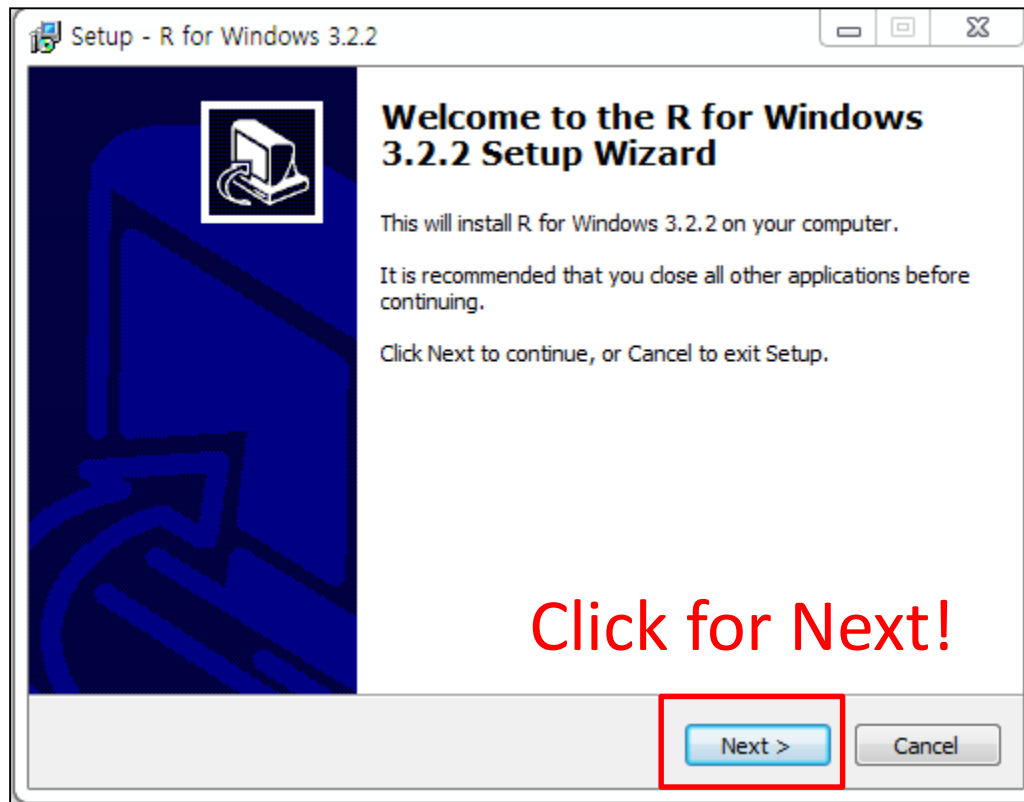
Click here!
Drop down menu
appears.
Choose language!

Now, the window displays Korean.
The display language depends on your
computer system!



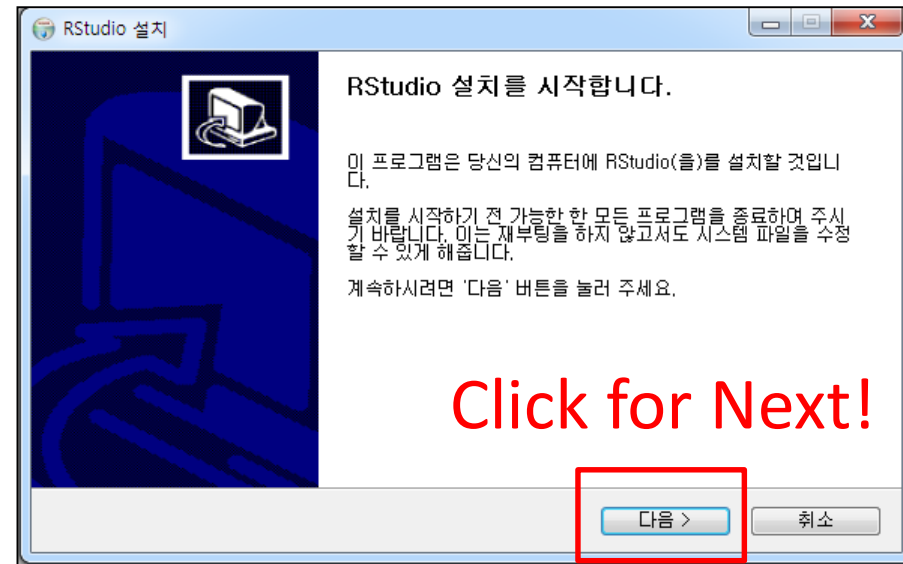
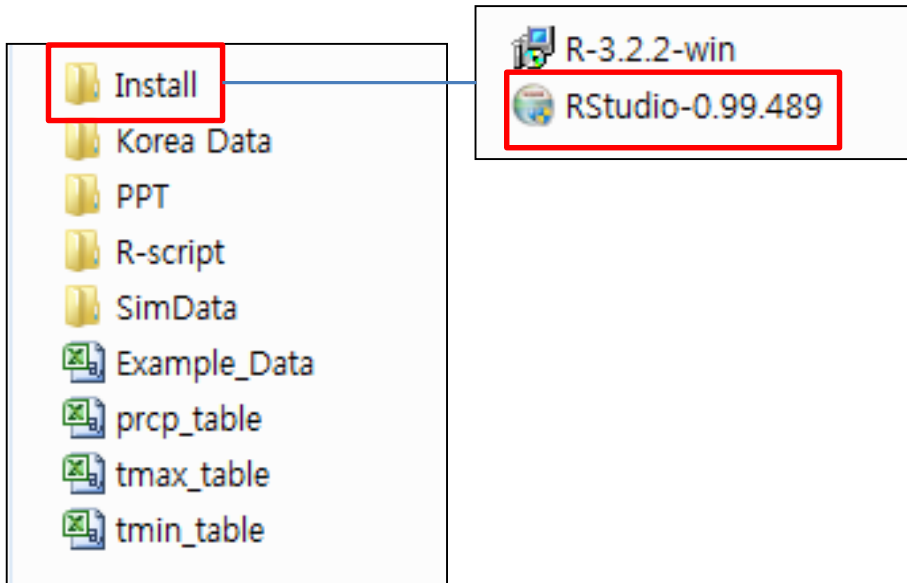
Preparation

- Install R
 - Click **Next** button repeatedly by completing.



Preparation

- Install R Studio
 - Go to **Install** folder
 - Double-click **Rstudio-0.99.489**



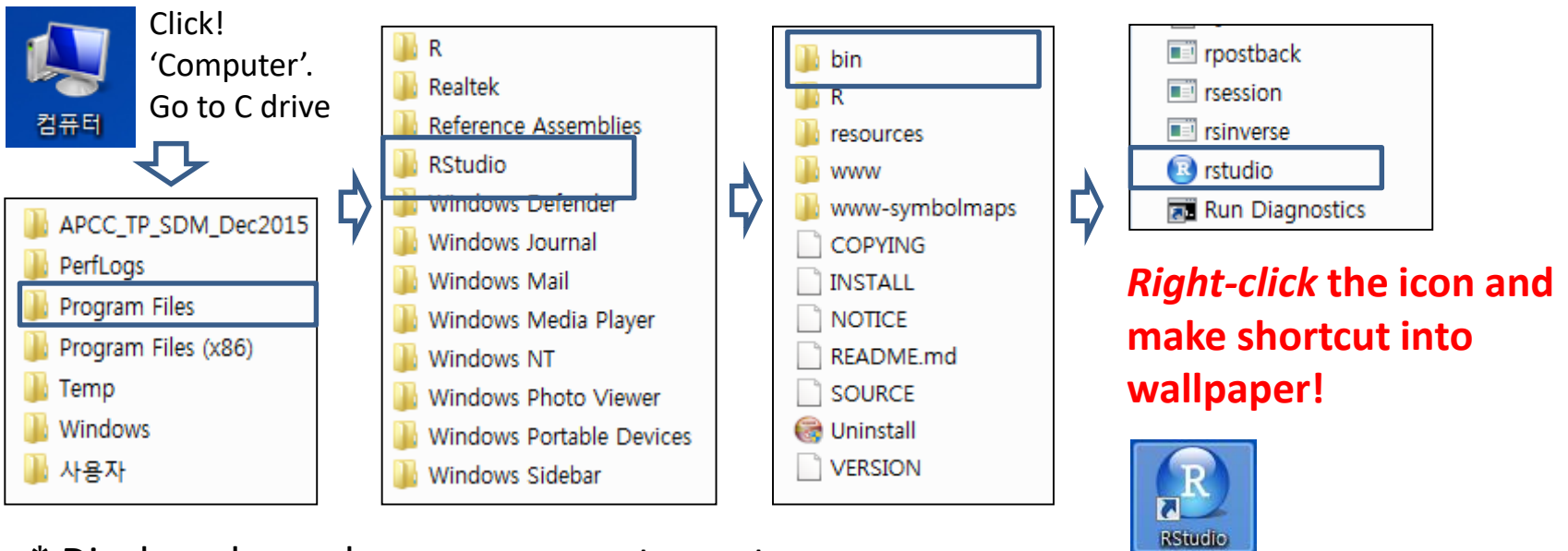
Now, the window displays Korean.

The display language depends on your computer system!



Preparation

- Make icon on wallpaper



* Displays depend on your computer system



Preparation

- Start R Studio by following a way:

- Double-clicking in wallpaper
Then, R Studio starts!



- If the icon not appear, then
Start > Search > Searching **RStudio** > Click!
(See the next page for illustration!)



Preparation

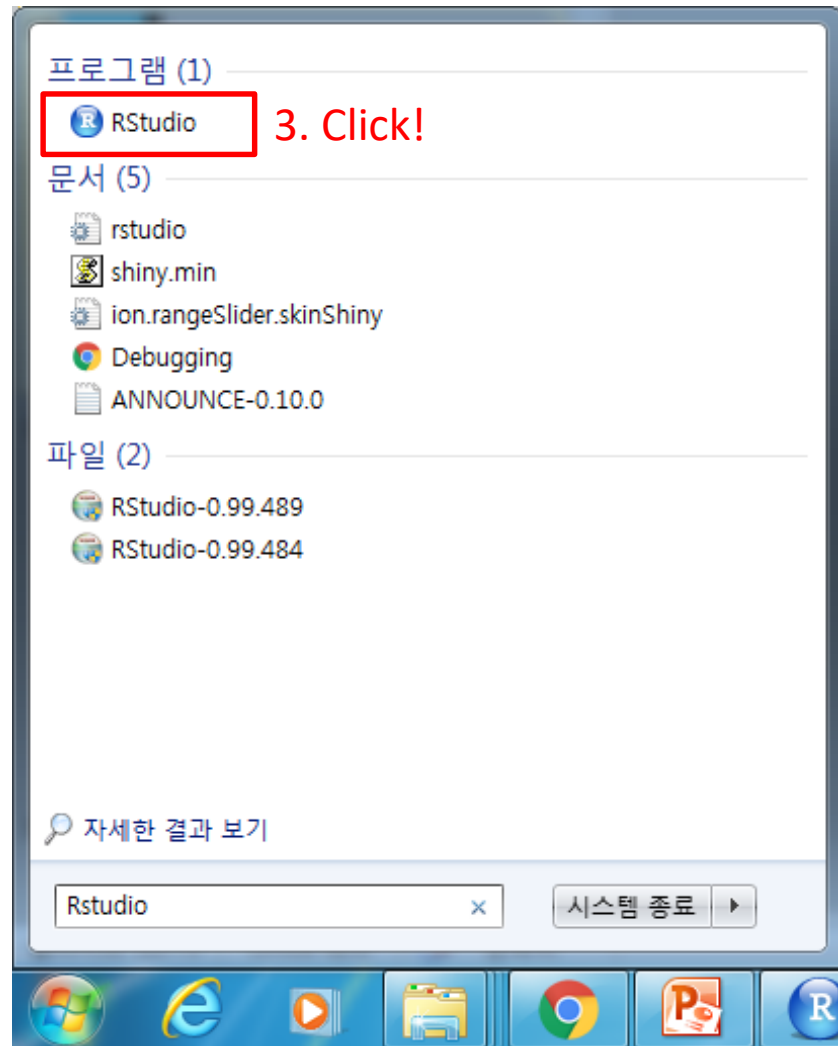
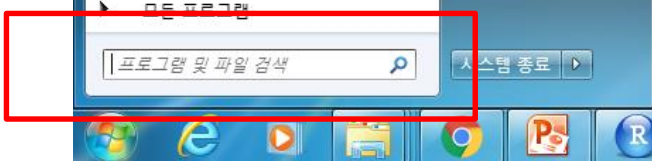
- Start R Studio



1. Click **Start** Button

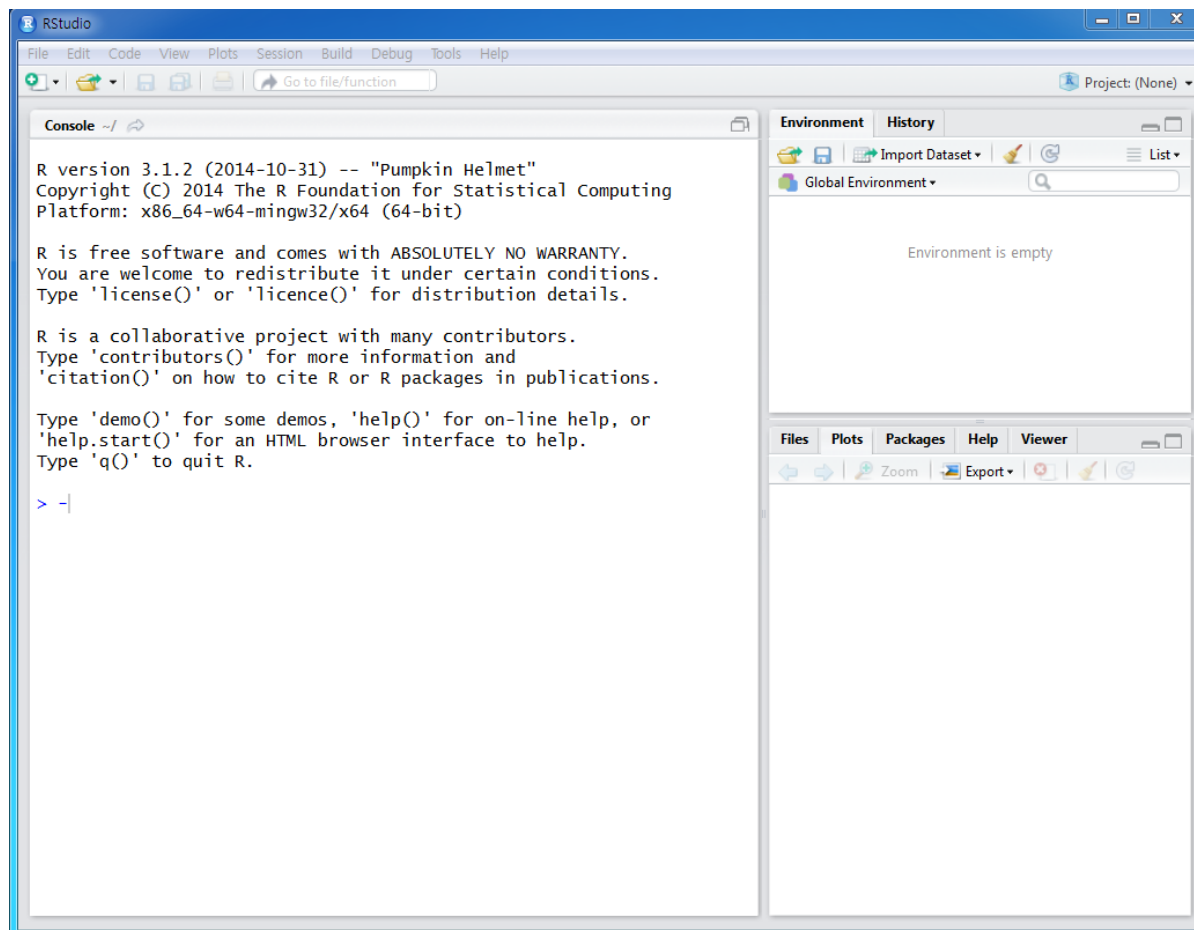


2. Type **RStudio** in Search blank



Preparation

- Start Rstudio! (Display can be slightly different.)



Preparation

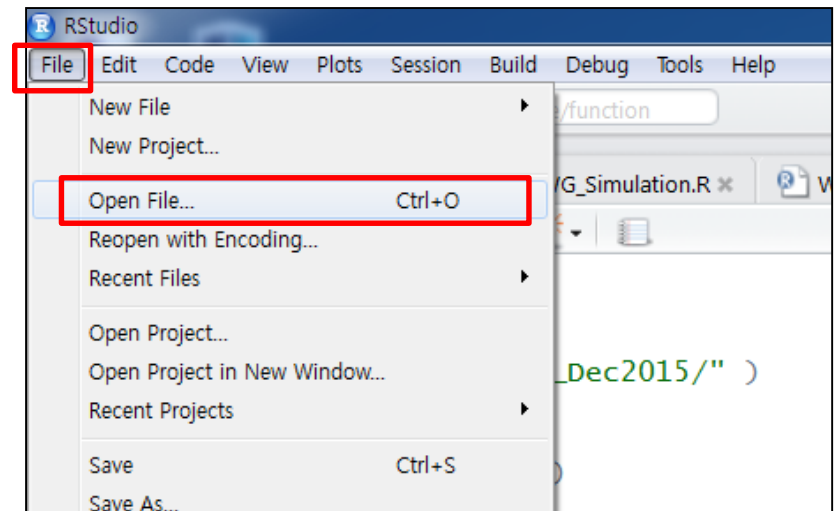
- Open R-code file **Quick_R_Start**
 - File menu > Open File
 - Selecting the file in
**C:\APCC_TP_SDM_Aug2016\WGenerator
\R-script\
Next page, the illustration is presented.**



Preparation

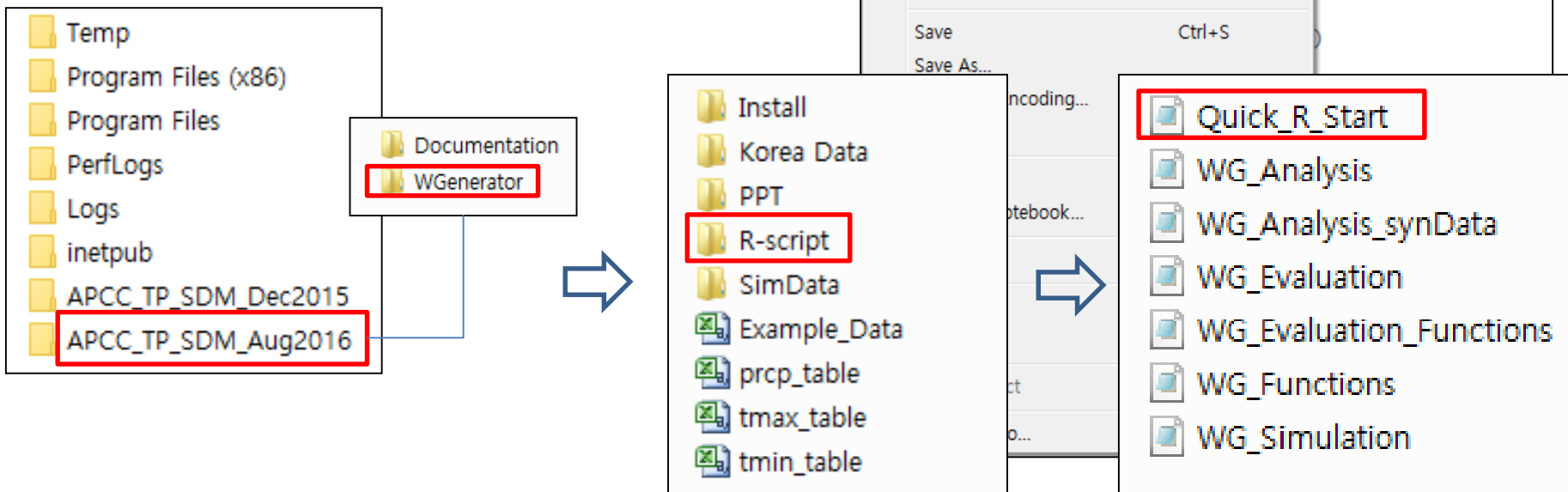
- Open Quick_R_Start

1. Click File



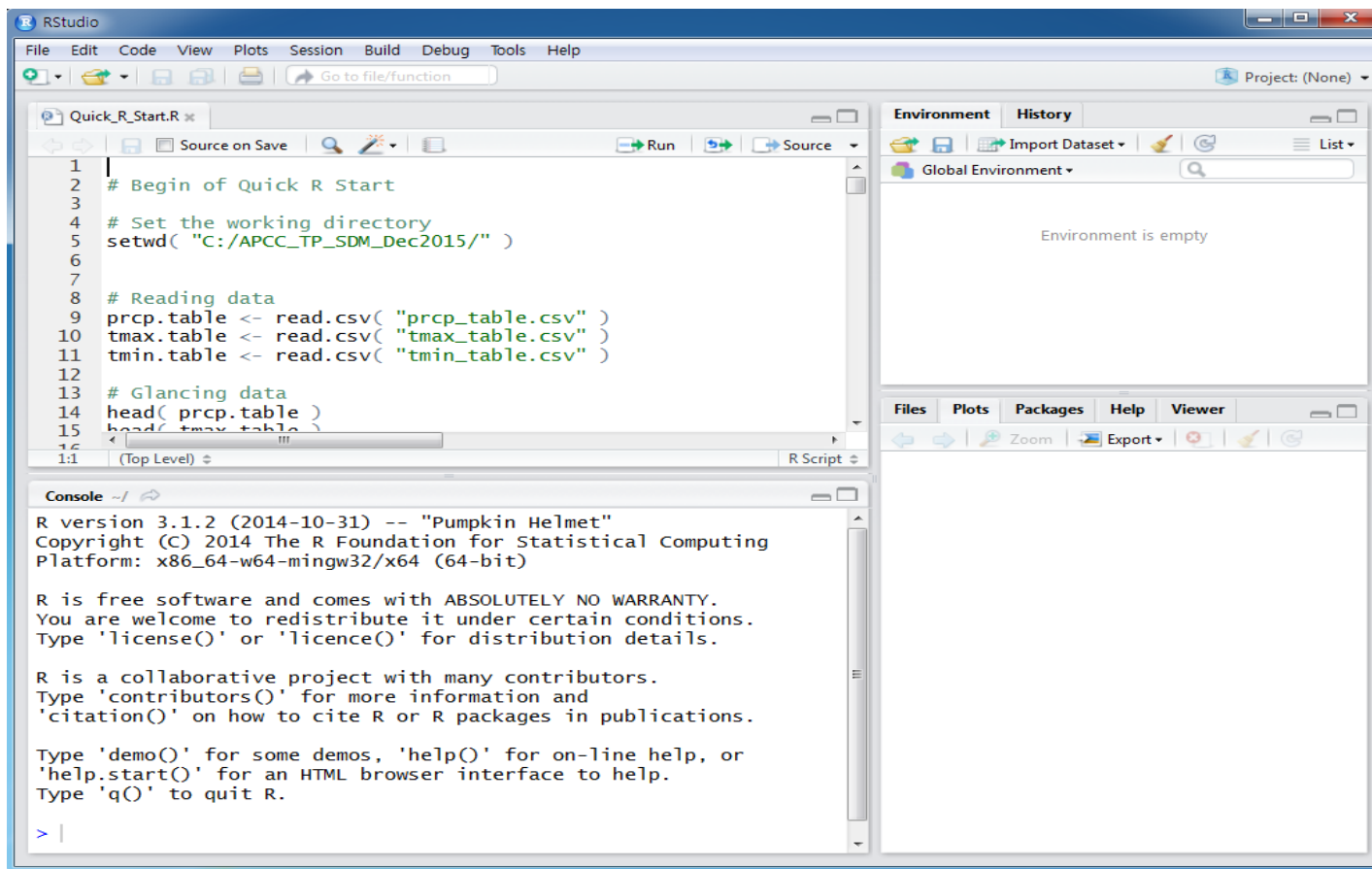
2. Click Open File...

- 3. Find Quick_R_Start by double click.
(Searching start C:\)



Preparation

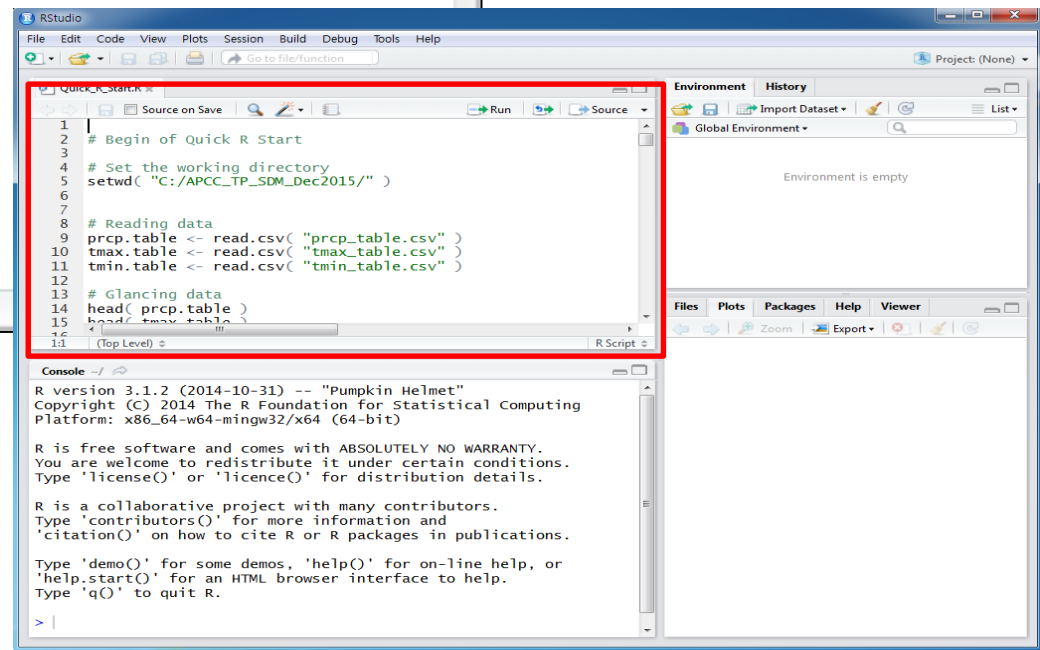
- R studio has several windows



Preparation

- R-script window

```
1  
2 # Begin  
3  
4 setwd( "C:/APCC_TP_SDM_Dec2015/" )  
5  
6 prcp.table <- read.csv( "prcp_table.csv" )  
7 tmax.table <- read.csv( "tmax_table.csv" )  
8 tmin.table <- read.csv( "tmin_table.csv" )  
9  
10 head( prcp.table )  
11 head( tmax.table )  
12 head( tmin.table )  
13  
6:1 (Top Level) ▾
```



Preparation

- Console

```
Console - / ↻

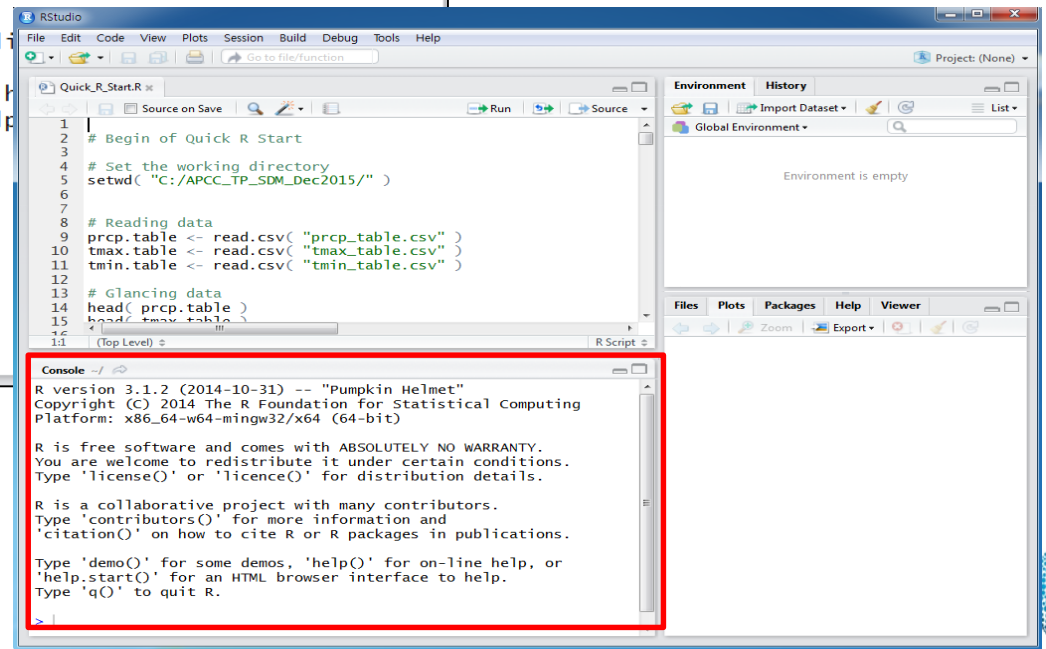
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

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.

> |
```



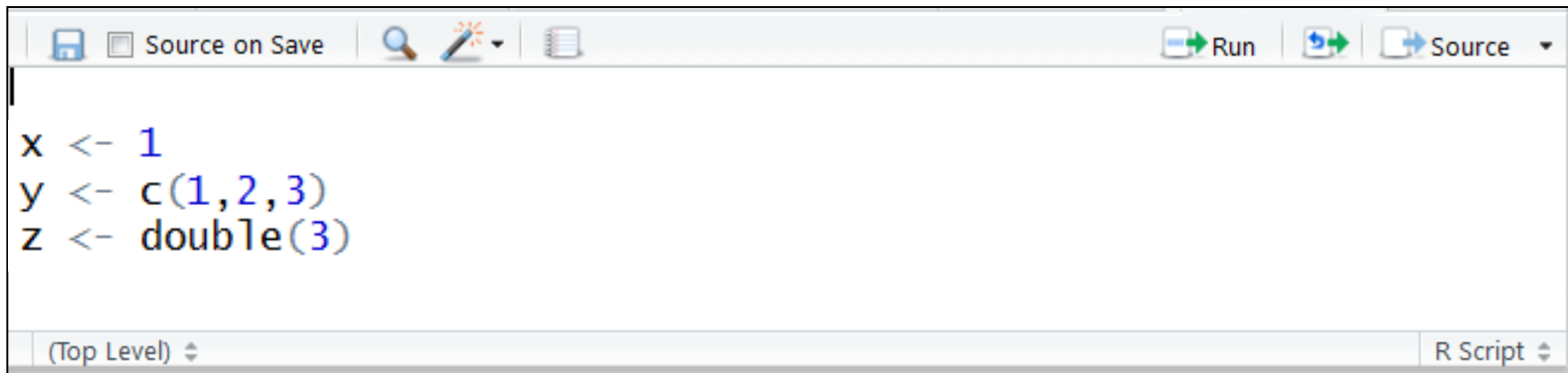
Contents

- Preparation and installation
- **Variables and Operations**
- Data Analysis using R
- Reading your Observation Data



Variables and Operation

- Creating variable and vector



```
x <- 1
y <- c(1,2,3)
z <- double(3)
```

The screenshot shows an R script editor window. The title bar includes 'Source on Save', a search icon, a refresh icon, and buttons for 'Run' and 'Source'. The main text area contains three lines of R code: 'x <- 1', 'y <- c(1,2,3)', and 'z <- double(3)'. The status bar at the bottom shows '(Top Level)' and 'R Script'.

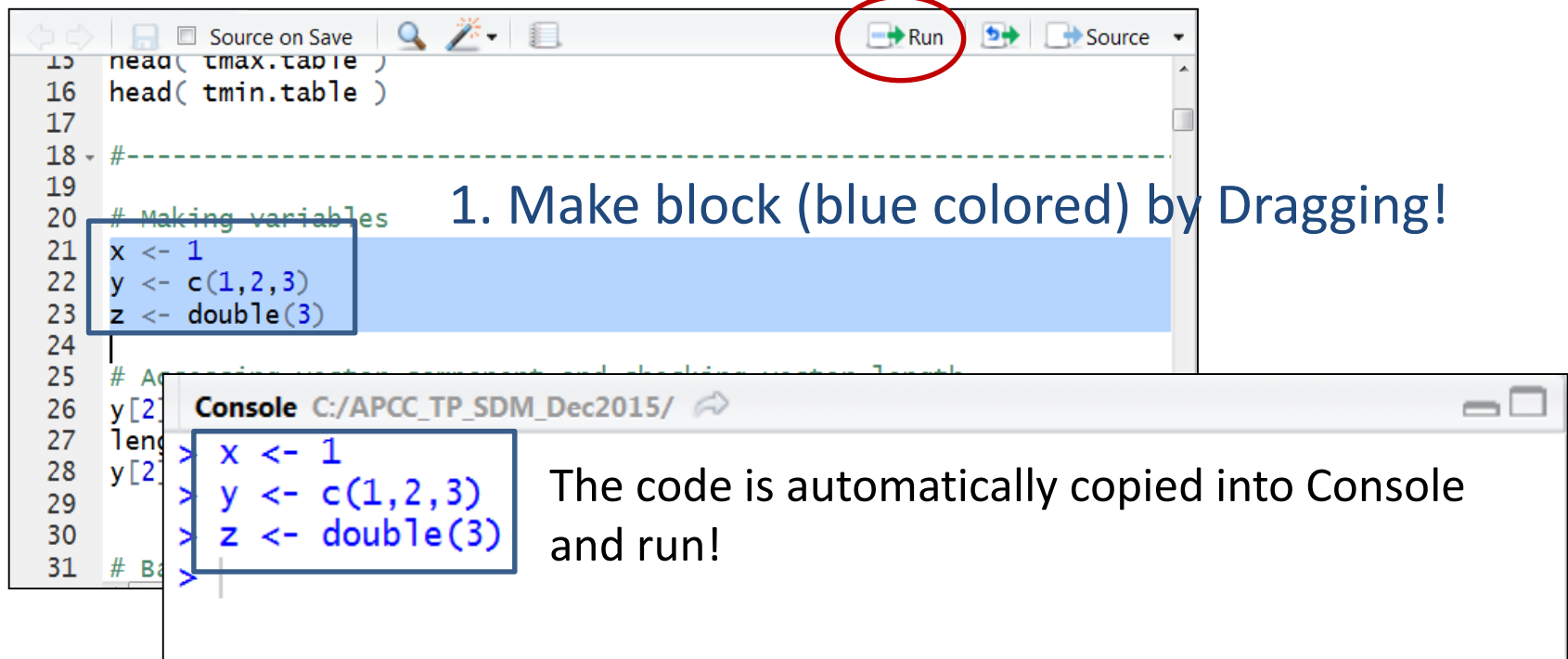
- Variable 'x' is created with value 1.
- Variable 'y' is a vector with initial values.
- Variable 'z' is a vector with length 3 and default value 0.
- Other characters and digits can be used for variable name. e.g., 'x1', 'z2', 'ab1', etc.



Variables and Operation

- How to run the code

2. Click!



The screenshot shows an R IDE window with a code editor and a console. The code editor contains the following R code:

```
15 head( tmax.table )
16 head( tmin.table )
17
18 #-----
19
20 # Making variables
21 x <- 1
22 y <- c(1,2,3)
23 z <- double(3)
24
25 # A
26 y[2]
27 leng
28 y[2]
29
30
31 # B
```

The code block from line 20 to 23 is highlighted in blue. A red circle highlights the 'Run' button in the toolbar. A blue box highlights the code in the console window, which is automatically copied from the editor.

1. Make block (blue colored) by Dragging!

The code is automatically copied into Console and run!



Variables and Operation

- Checking variables

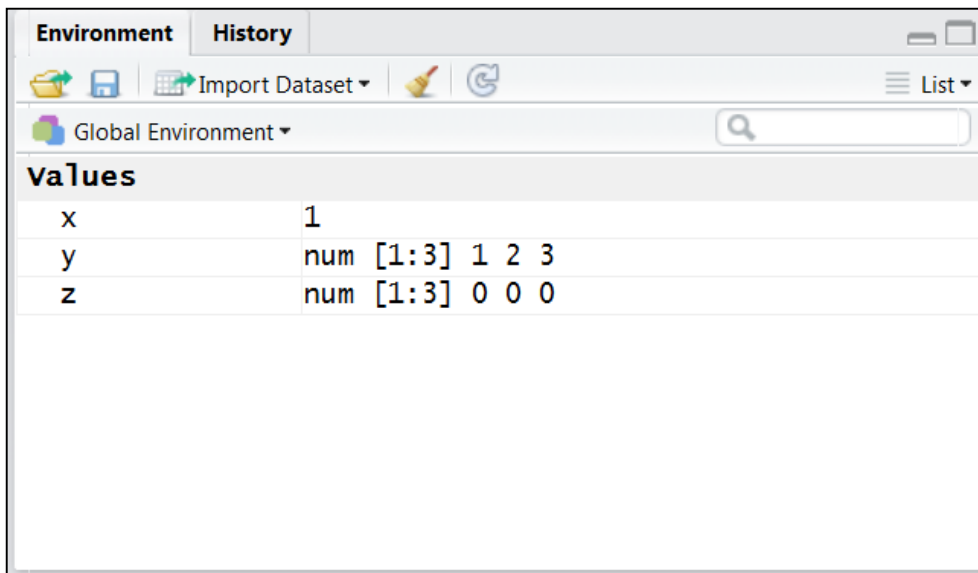
```
Console C:/APCC_TP_SDM_Dec2015/
> x <- 1
> y <- c(1,2,3)
> z <- double(3)
> x
[1] 1
> y
[1] 1 2 3
> z
[1] 0 0 0
> |
```

- Type “x” on Console *directly* and push ‘Enter’!
You can check the value of variable ‘x’.
- The others can be checked in the same way!



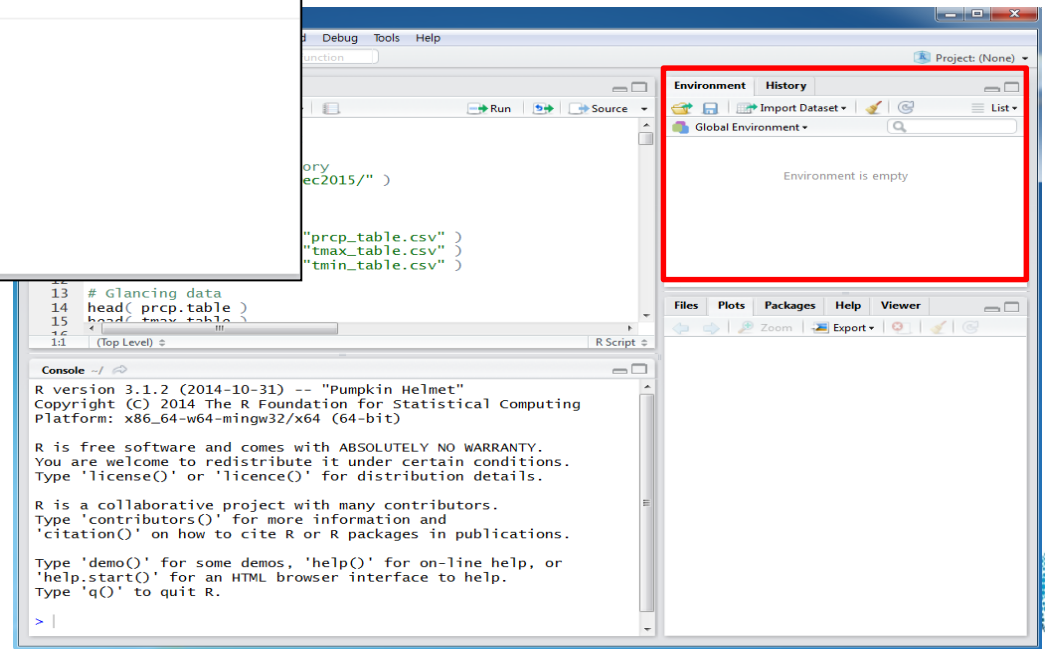
Variables and Operation

- You can check by seeing Environment window.



The Environment window in R Studio displays the following variables and their values:

Variable	Value
x	1
y	num [1:3] 1 2 3
z	num [1:3] 0 0 0



The screenshot shows the R Studio interface with the Environment window highlighted by a red box. The Environment window displays "Environment is empty". The script editor shows the following code:

```
13 # Glancing data
14 head( prcp.table )
15 head( tmax.table )
16
17 # Glancing data
18 head( prcp.table )
19 head( tmax.table )
20
21 # Glancing data
22 head( prcp.table )
23 head( tmax.table )
24
25 # Glancing data
26 head( prcp.table )
27 head( tmax.table )
28
29 # Glancing data
30 head( prcp.table )
31 head( tmax.table )
32
33 # Glancing data
34 head( prcp.table )
35 head( tmax.table )
36
37 # Glancing data
38 head( prcp.table )
39 head( tmax.table )
40
41 # Glancing data
42 head( prcp.table )
43 head( tmax.table )
44
45 # Glancing data
46 head( prcp.table )
47 head( tmax.table )
48
49 # Glancing data
50 head( prcp.table )
51 head( tmax.table )
52
53 # Glancing data
54 head( prcp.table )
55 head( tmax.table )
56
57 # Glancing data
58 head( prcp.table )
59 head( tmax.table )
60
61 # Glancing data
62 head( prcp.table )
63 head( tmax.table )
64
65 # Glancing data
66 head( prcp.table )
67 head( tmax.table )
68
69 # Glancing data
70 head( prcp.table )
71 head( tmax.table )
72
73 # Glancing data
74 head( prcp.table )
75 head( tmax.table )
76
77 # Glancing data
78 head( prcp.table )
79 head( tmax.table )
80
81 # Glancing data
82 head( prcp.table )
83 head( tmax.table )
84
85 # Glancing data
86 head( prcp.table )
87 head( tmax.table )
88
89 # Glancing data
90 head( prcp.table )
91 head( tmax.table )
92
93 # Glancing data
94 head( prcp.table )
95 head( tmax.table )
96
97 # Glancing data
98 head( prcp.table )
99 head( tmax.table )
100
```

The console shows the R version and platform information:

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

> |
```

Variables and Operation

- # Comment

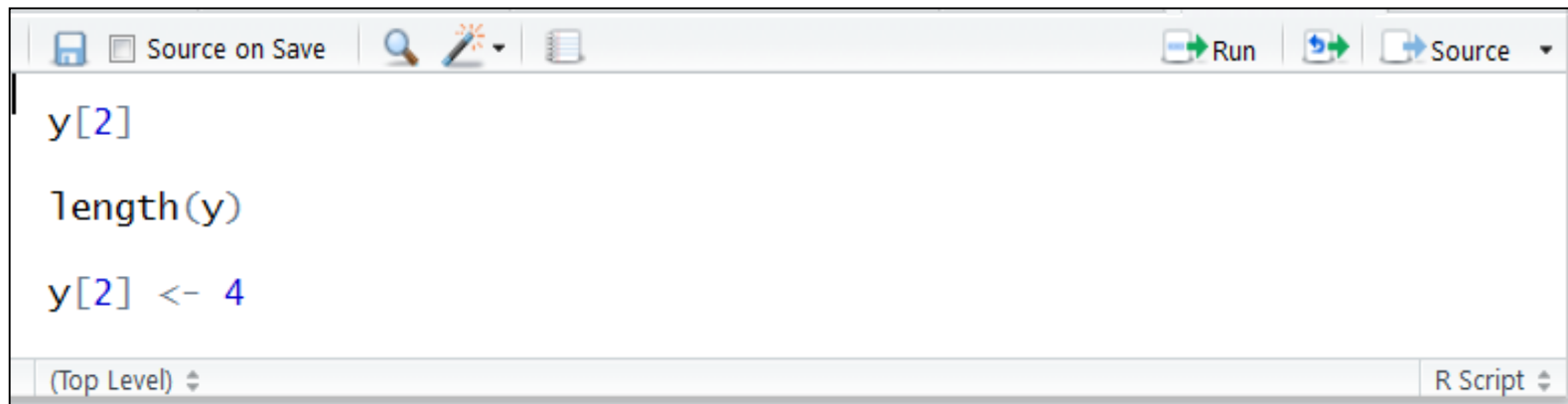
```
15 head( tmax.table )
16 head( tmin.table )
17
18 #-----
19
20 # Making variables
21 x <- 1
22 y <- c(1,2,3)
23 z <- double(3)
24
25 # Accessing vector component and checking vector length
26 y[2]
27 length(y)
28 y[2] <- 4
29
30
31 # Basic Operations
```

- The sentences with # head and green color are **comments** *which describes the code*.
- **They are not run when copied into Console.**



Variables and Operation

- Operation



```
Source on Save | Run | Source
y[2]
length(y)
y[2] <- 4
(Top Level) | R Script
```

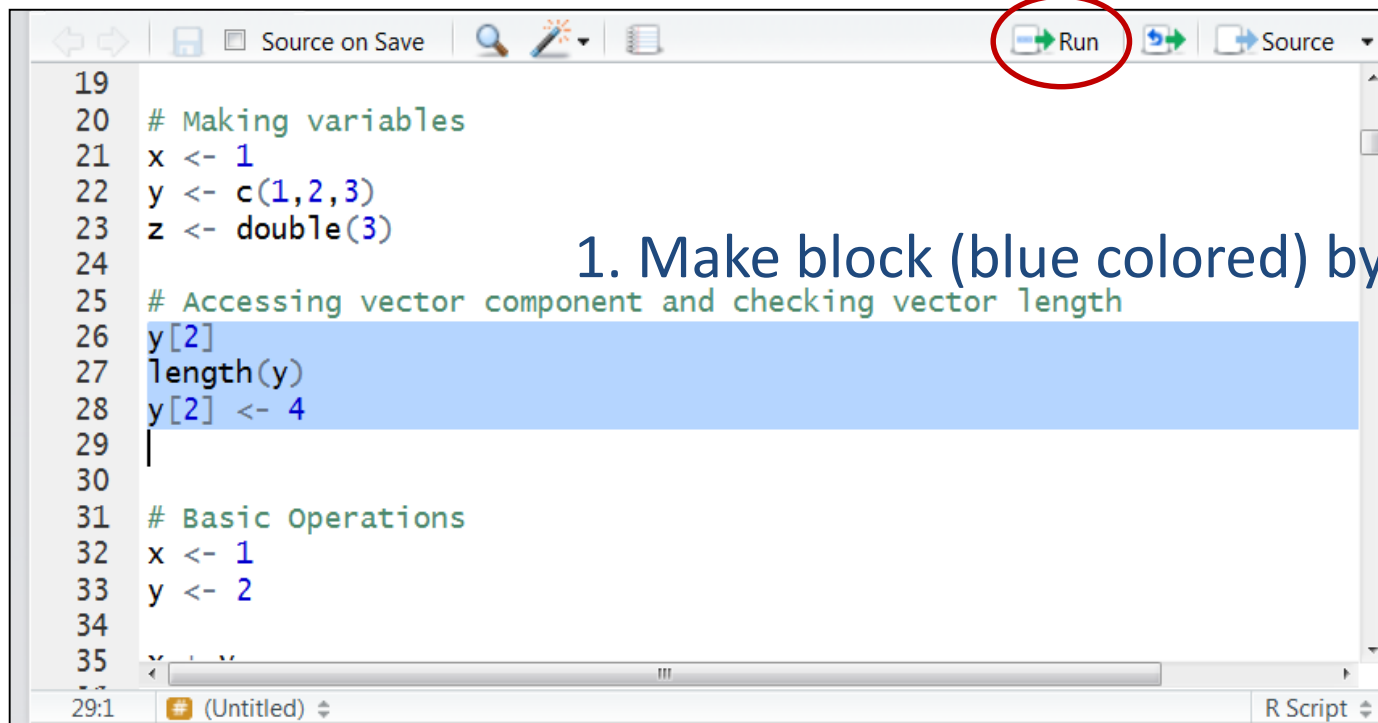
- Accessing the second entry of 'y'.
- Length of vector 'y'.
- Changing the value of y[2] to 4.



Variables and Operation

- Operation

2. Click!



The screenshot shows an R script editor window. The code is as follows:

```
19
20 # Making variables
21 x <- 1
22 y <- c(1,2,3)
23 z <- double(3)
24
25 # Accessing vector component and checking vector length
26 y[2]
27 length(y)
28 y[2] <- 4
29
30
31 # Basic Operations
32 x <- 1
33 y <- 2
34
35
```

The lines 26, 27, and 28 are highlighted in blue. The 'Run' button in the toolbar is circled in red. The status bar at the bottom shows '29:1 (Untitled)' and 'R Script'.

1. Make block (blue colored) by Dragging!

Remind: The sentences with # head and green color are **comments** which describes the code. They are not run even though copied into Console.

Variables and Operation

- Operation

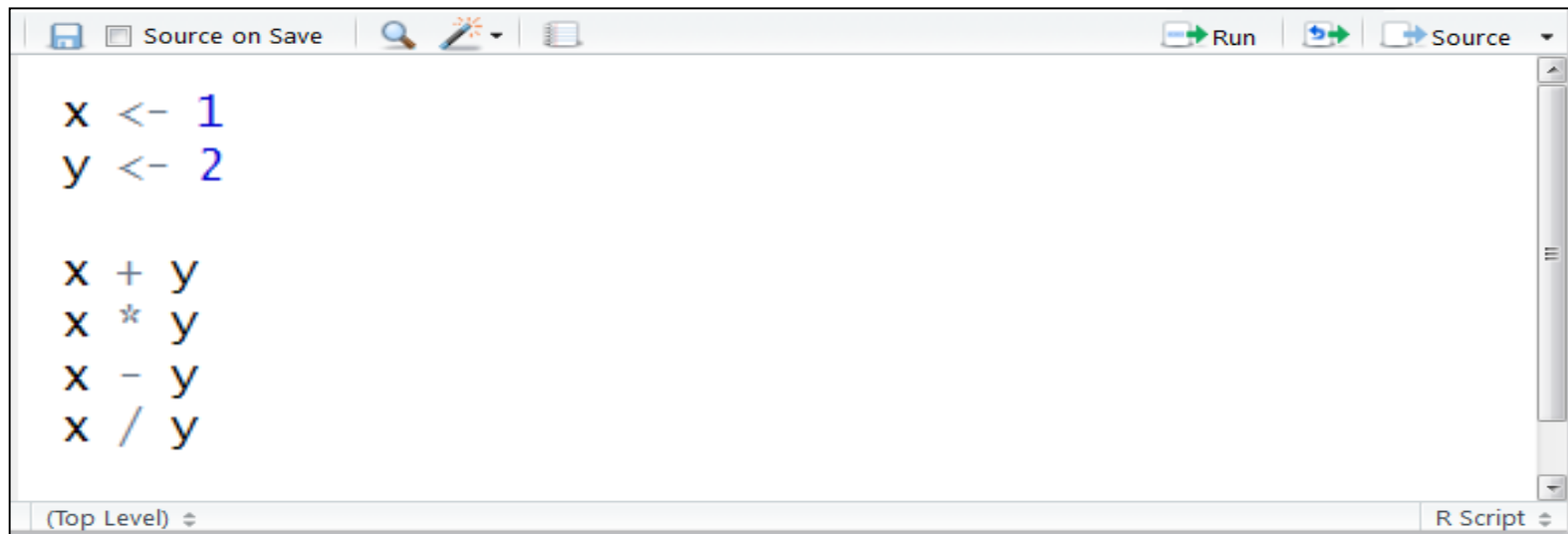
```
Console C:/APCC_TP_SDM_Dec2015/
> x <- 1
> y <- c(1,2,3)
> z <- double(3)
> x
[1] 1
> y
[1] 1 2 3
> z
[1] 0 0 0
> y[2]
[1] 2
> length(y)
[1] 3
> y[2] <- 4
> y
[1] 1 4 3
> |
```

Value changing!



Variables and Operation

- Operation



```
x <- 1
y <- 2

x + y
x * y
x - y
x / y
```

The screenshot shows an R script editor window with a toolbar at the top containing icons for Save, Source on Save, Search, Run, and Source. The main text area contains the following R code: `x <- 1`, `y <- 2`, `x + y`, `x * y`, `x - y`, and `x / y`. The status bar at the bottom indicates '(Top Level)' and 'R Script'.

- Two variables 'x' and 'y' with initial values
- Addition (+), Multiplication (*), Subtraction (-), Division (/)



Variables and Operation

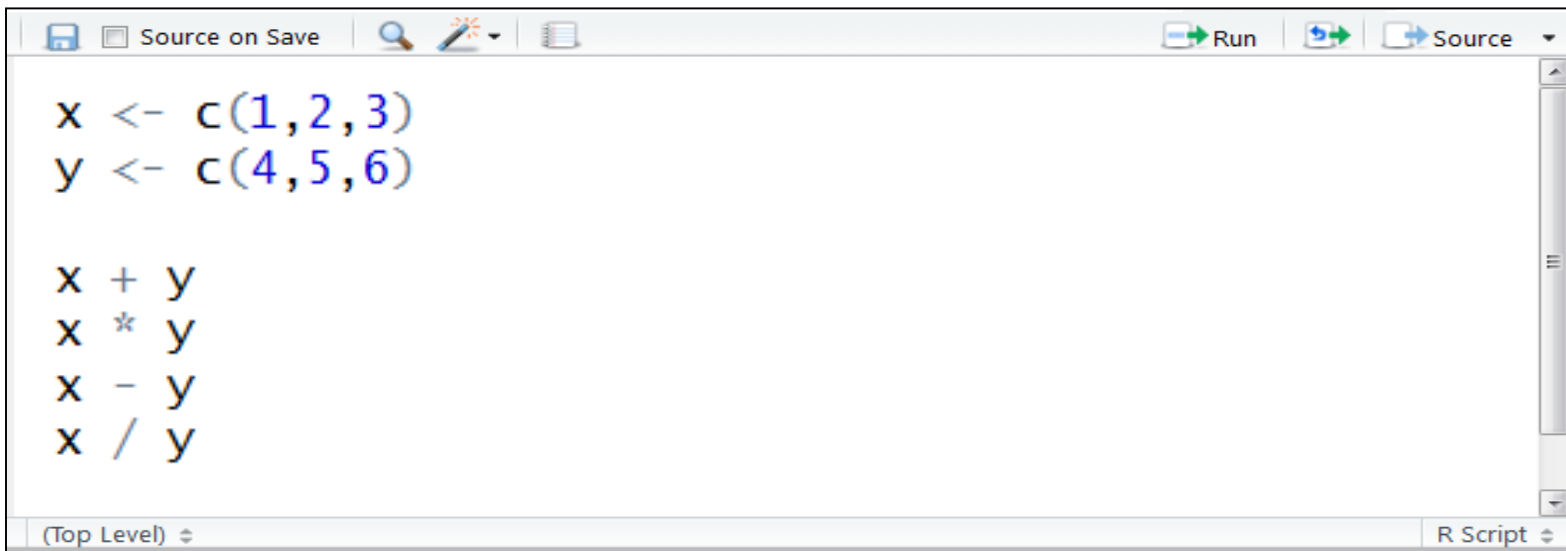
- Operation

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- 1
> y <- 2
>
> x + y
[1] 3
> x * y
[1] 2
> x - y
[1] -1
> x / y
[1] 0.5
> |
```



Variables and Operation

- Operation



```
x <- c(1, 2, 3)
y <- c(4, 5, 6)

x + y
x * y
x - y
x / y
```

The screenshot shows an R script editor window with a toolbar at the top containing icons for Save, Source on Save, Search, Run, and Source. The main text area contains the R code shown above. The status bar at the bottom indicates '(Top Level)' and 'R Script'.

- Two vectors 'x' and 'y'.
- Operations are same, but every operation is made *component-wisely*!



Variables and Operation

- Operation

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- c(1,2,3)
> y <- c(4,5,6)
>
> x + y
[1] 5 7 9
> x * y
[1] 4 10 18
> x - y
[1] -3 -3 -3
> x / y
[1] 0.25 0.40 0.50
> |
```

Note: operations are applied to every component, separately.



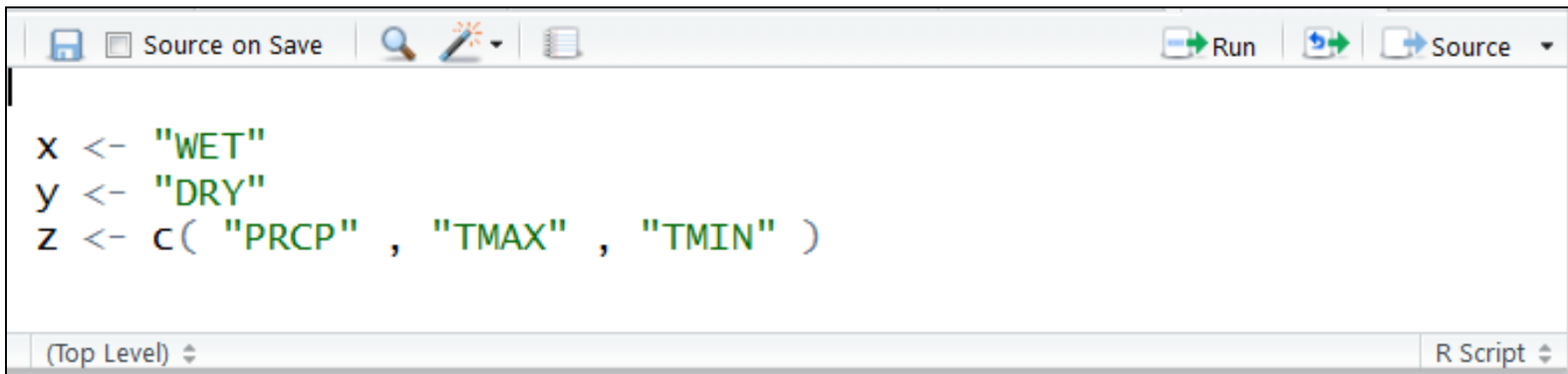
Variables and Operation

- Practice:
 - Create variables 'a', 'b' and vectors 'z', 'w' with length 5 and initial values you want.
 - Access components of vectors 'z' and 'w'.
 - Operate the variables. For instance, $z + w$. Check the operation results.



Variables and Operation

- Character variable and vector



```
x <- "WET"  
y <- "DRY"  
z <- c( "PRCP" , "TMAX" , "TMIN" )
```

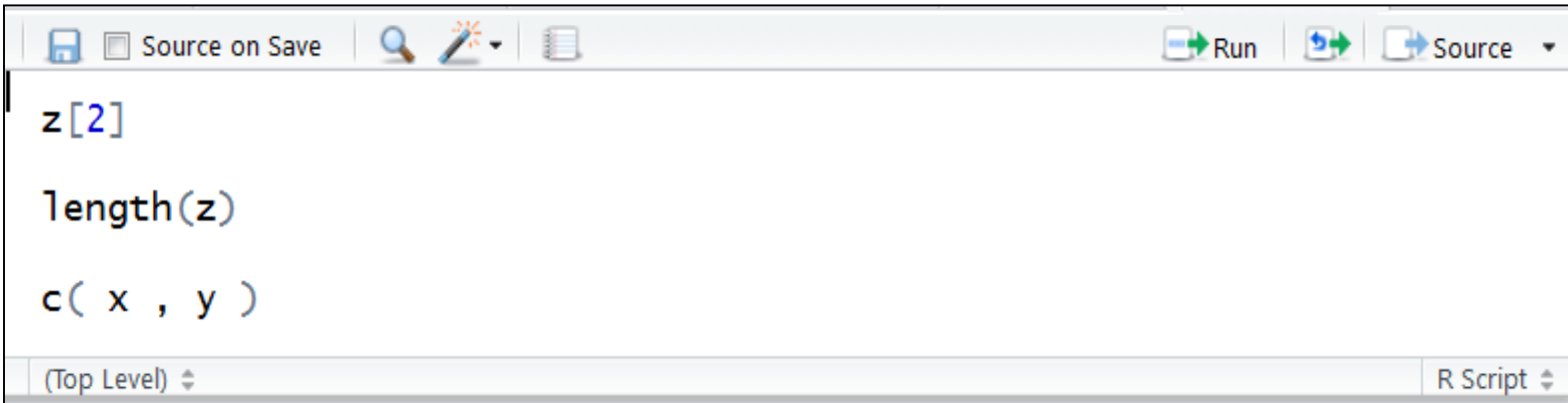
The screenshot shows an R script editor window. The title bar includes icons for save, source on save, search, and run. The main area contains three lines of R code: `x <- "WET"`, `y <- "DRY"`, and `z <- c("PRCP" , "TMAX" , "TMIN")`. The status bar at the bottom indicates the current level is '(Top Level)' and the file type is 'R Script'.

- Variable 'x' is created with "WET" as initial.
- Variable 'y' is created with "DRY" as initial.
- Variable 'z' is a vector of length 3 with the initial characters.



Variables and Operation

- Character variable and vector



```
z[2]

length(z)

c(x, y)
```

- Accessing the second entry of vector 'z'.
- Length of vector 'z'.
- Connecting two character variables 'x' and 'y'.



Variables and Operation

- Character variable and vector

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- "WET"
> y <- "DRY"
> z <- c("PRCP", "TMAX", "TMIN")
>
> z[2]
[1] "TMAX"
>
> length(z)
[1] 3
>
> c(x, y)
[1] "WET" "DRY"
>
```

Note: "c(x,y)" connecting two variables 'x' and 'y' into a vector of length 2.



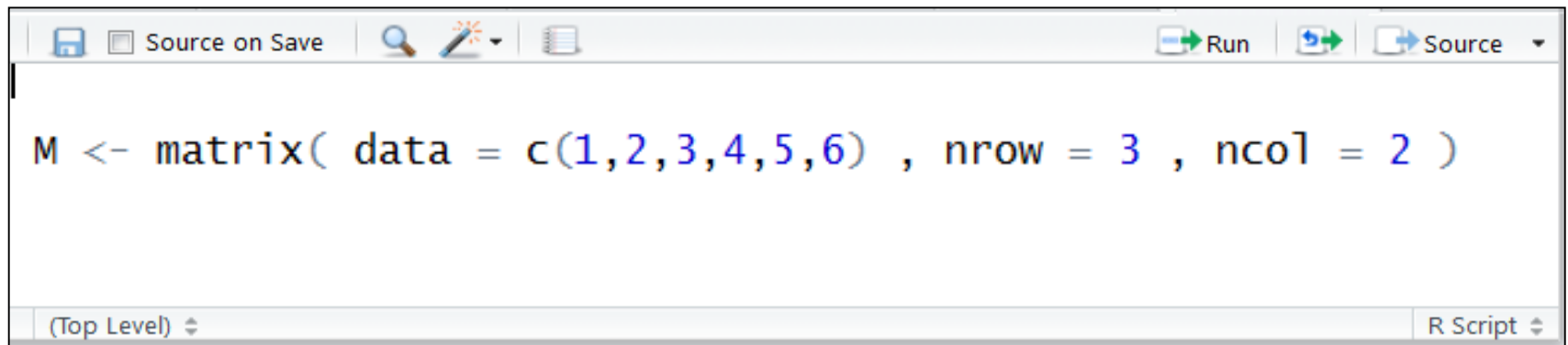
Variables and Operation

- Practice:
 - Create character variables 'x', 'y' and vectors 'z', 'w' with length 3 and initial characters you want.
 - Access components of character vectors,
 - Connecting
 - 'x' and 'y',
 - 'z' and 'w',
 - 'x' and 'z'.



Variables and Operation

- Creating matrix and access



```
M <- matrix( data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2 )
```

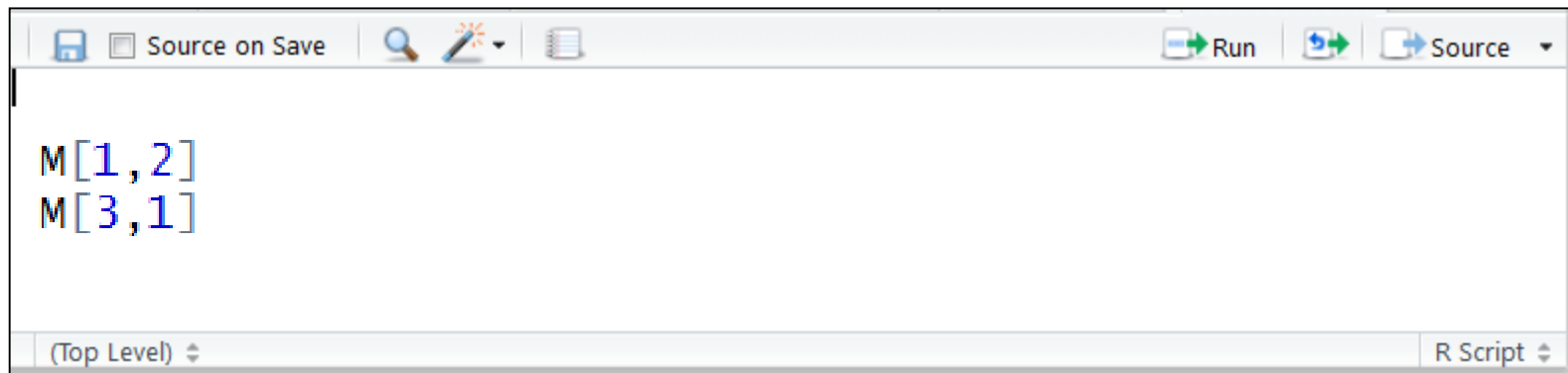
The screenshot shows an R script editor window. The title bar includes a save icon, a checkbox for 'Source on Save', a search icon, a refresh icon, and a 'Run' button. The main text area contains the R code: `M <- matrix(data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2)`. The status bar at the bottom shows '(Top Level)' on the left and 'R Script' on the right.

- ‘M’ is a matrix of 3×2 with initial values.



Variables and Operation

- Creating matrix and access (cont.)



The image shows a screenshot of an R script editor window. The window has a toolbar at the top with icons for saving, running, and sourcing. The main text area contains two lines of R code: `M[1,2]` and `M[3,1]`. The bottom status bar shows "(Top Level)" and "R Script".

```
M[1,2]
M[3,1]
```

- Accessing the entry of matrix ‘M’
 - The entry of the first row and second column.
 - The entry of the third row and first column.



Variables and Operation

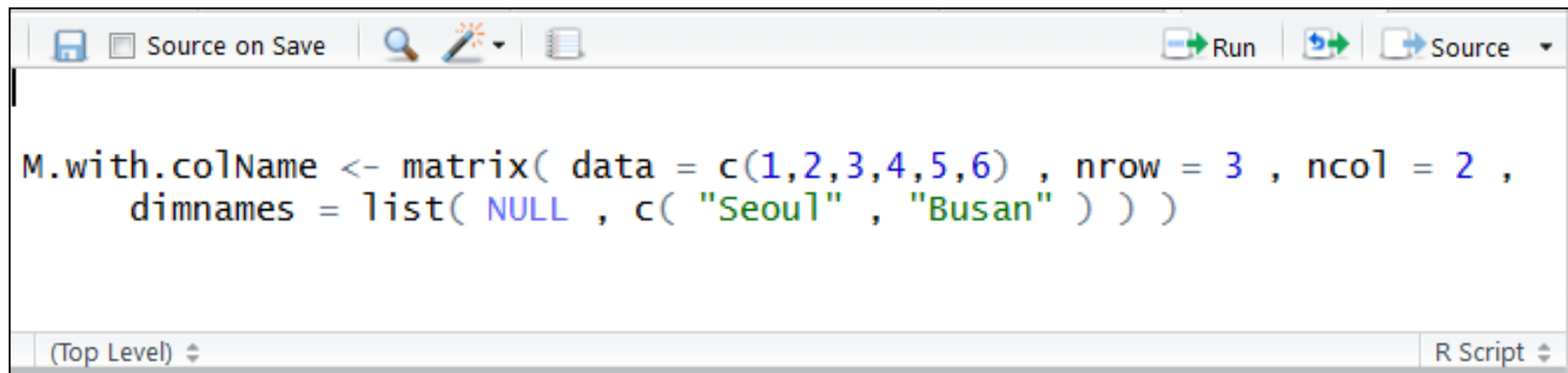
- Creating matrix and access (cont.)

```
Console C:/APCC_TP_SDM_Dec2015/
> M <- matrix( data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2 )
>
> M
      [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
>
> M[1,2]
[1] 4
> M[3,1]
[1] 3
> |
```



Variables and Operation

- Creating matrix with column names



```
M.with.colName <- matrix( data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2 ,  
  dimnames = list( NULL , c( "Seoul" , "Busan" ) ) )
```

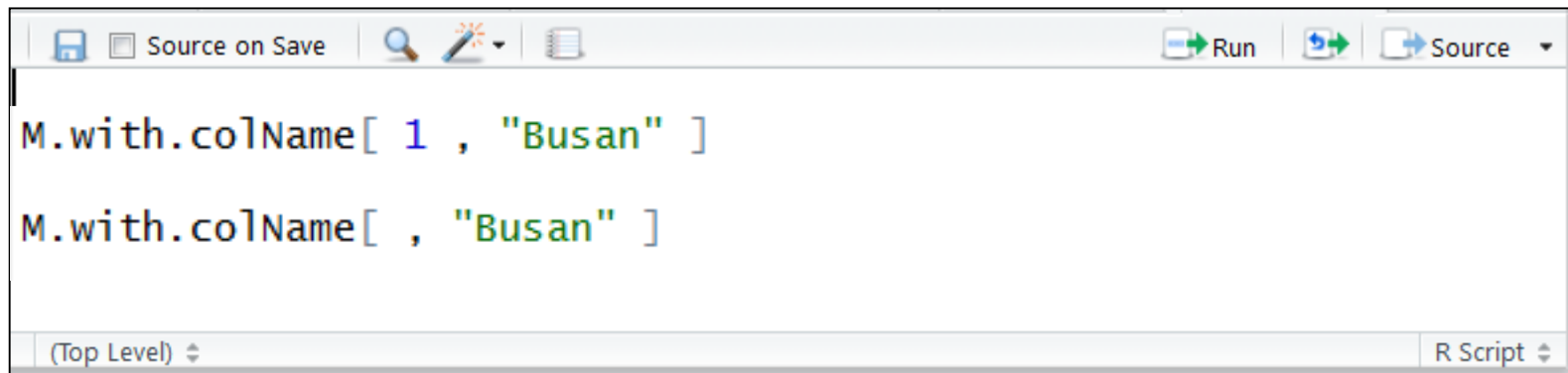
The screenshot shows an R script editor window. The title bar includes 'Source on Save', 'Run', and 'Source' buttons. The main area contains the R code for creating a matrix with column names. The status bar at the bottom indicates '(Top Level)' and 'R Script'.

- ‘M.with.colName’ is created with column names “Seoul” and “Busan”.



Variables and Operation

- Accessing matrix with column names




```
M.with.colName[ 1 , "Busan" ]  
M.with.colName[ , "Busan" ]
```

- Accessing the entry of the first row and “Busan” column.
- Accessing the entries of “Busan” column.



Variables and Operation

- Accessing matrix with column names

```
Console ~/ 
> M.with.colName <- matrix( data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2 ,
+   dimnames = list( NULL , c( "Seoul" , "Busan" ) ) )
>
> M.with.colName
      Seoul Busan
[1,]    1    4
[2,]    2    5
[3,]    3    6
>
> M.with.colName[ 1 , "Busan" ]
Busan
  4
>
> M.with.colName[ , "Busan" ]
[1] 4 5 6
> |
```



Variables and Operation

- Practice
 - Create a new matrix 'Mat' of another size.
Note: *'data vector length' = nrow × ncol!*
 - Access entries of the created matrix 'Mat'.
 - Access entries of "Seoul" column of 'M.with.colName'.



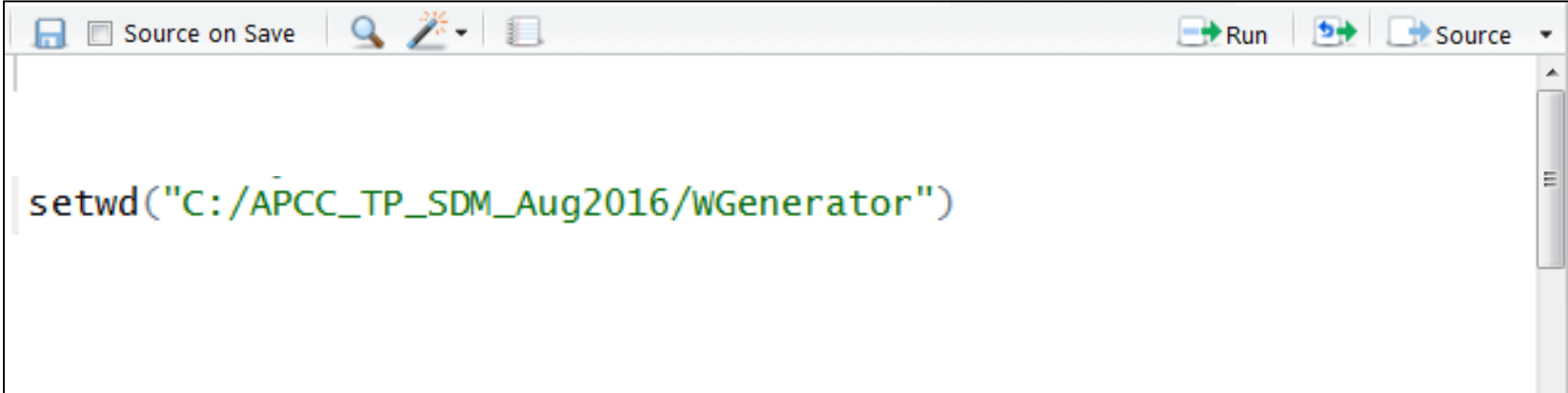
Contents

- Preparation and installation
- Variables and Operations
- **Data Analysis using R**
- Reading your Observation Data



Data Analysis

- Set working directory.



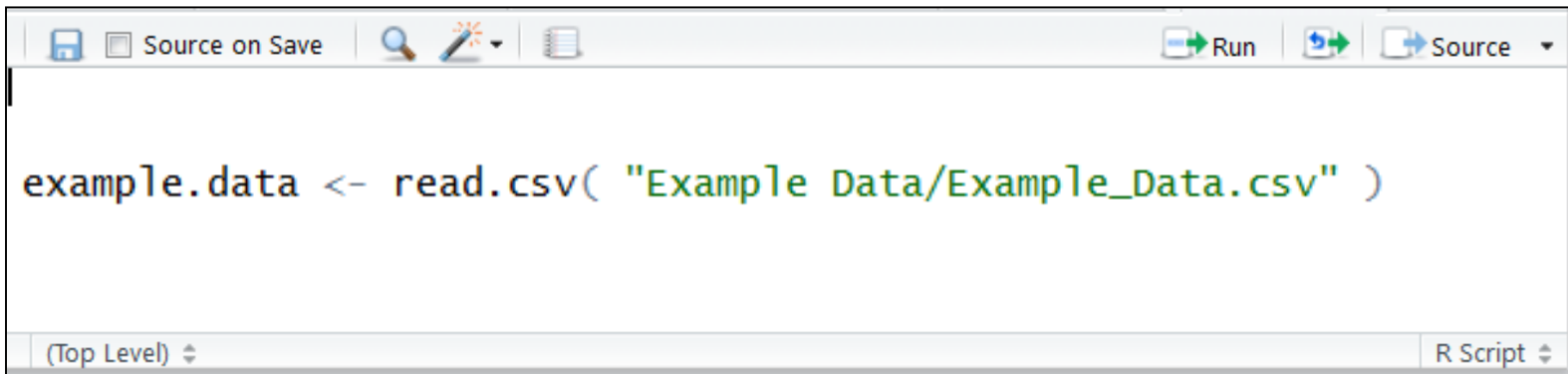
A screenshot of a code editor window. The window has a light gray title bar with several icons: a save icon, a checkbox labeled "Source on Save", a search icon, a pencil icon, and a notepad icon. On the right side of the title bar, there are three buttons: "Run" with a green arrow, a refresh icon, and "Source" with a blue arrow and a dropdown arrow. The main area of the window is white and contains a single line of code: `setwd("C:/APCC_TP_SDM_Aug2016/WGenerator")`. The text is in a monospaced font, with the string content in green. A vertical scrollbar is visible on the right side of the code area.

```
setwd("C:/APCC_TP_SDM_Aug2016/WGenerator")
```



Data Analysis

- Read data file **Example_Data.csv**



```
example.data <- read.csv( "Example Data/Example_Data.csv" )
```

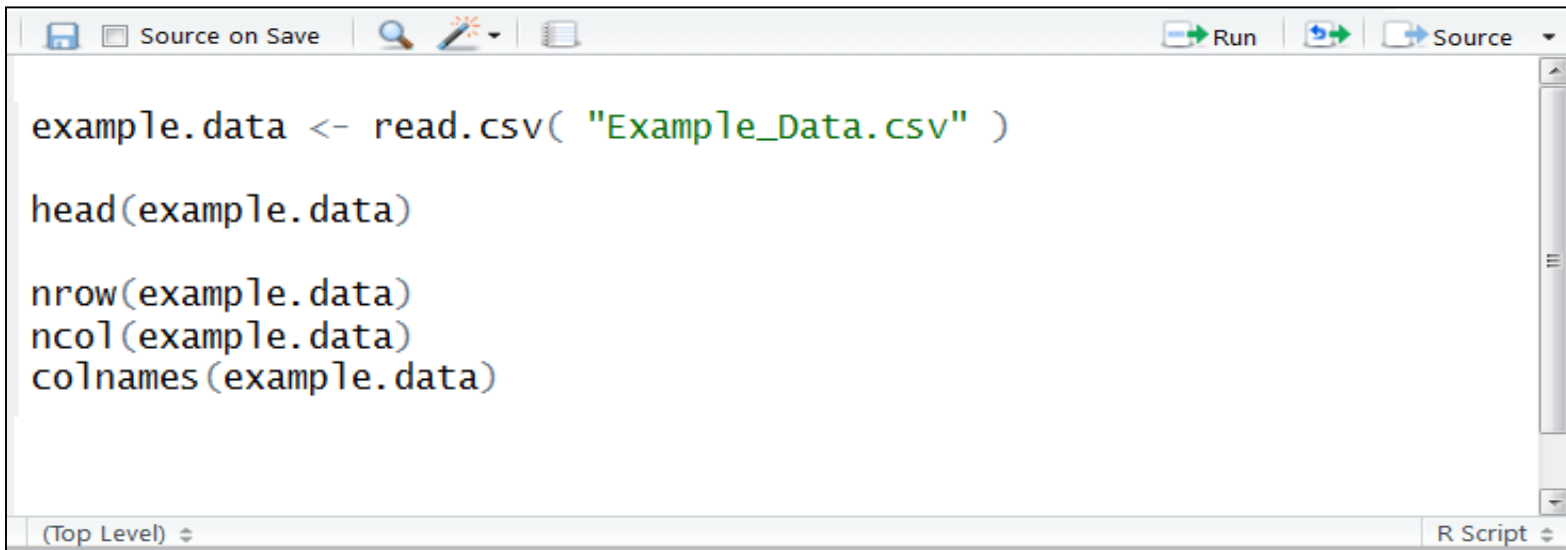
The screenshot shows an R script editor window. The title bar includes icons for save, source on save, search, and a palette, along with 'Run' and 'Source' buttons. The main text area contains the R code: `example.data <- read.csv("Example Data/Example_Data.csv")`. The status bar at the bottom shows '(Top Level)' and 'R Script'.

- Variable **example.data** is a matrix.
- It has 5 columns whose names are MONTH, DAY, PRCP, TMAX, TMIN.



Data Analysis

- Variable **example.data**



```
example.data <- read.csv( "Example_Data.csv" )  
  
head(example.data)  
  
nrow(example.data)  
ncol(example.data)  
colnames(example.data)
```

The screenshot shows an R script editor window with a toolbar at the top containing icons for Save, Source on Save, Search, Run, and Source. The script content is as follows:

- `nrow()`: the number of rows
- `ncol()`: the number of columns
- `colnames()`: the names of columns



Data Analysis

- Variable **example.data**

```
Console C:/APCC_TP_SDM_Dec2015/ ↵
> example.data <- read.csv( "Example_Data.csv" )
> head(example.data)
  MONTH DAY PRCP TMAX TMIN
1     6   1    0 26.4 15.7
2     6   2    0 27.8 14.8
3     6   3    0 29.0 13.6
4     6   4    0 30.8 15.8
5     6   5    0 31.6 17.4
6     6   6    0 29.2 18.4
> nrow(example.data)
[1] 61
> ncol(example.data)
[1] 5
> colnames(example.data)
[1] "MONTH" "DAY"   "PRCP"  "TMAX"  "TMIN"
> |
```



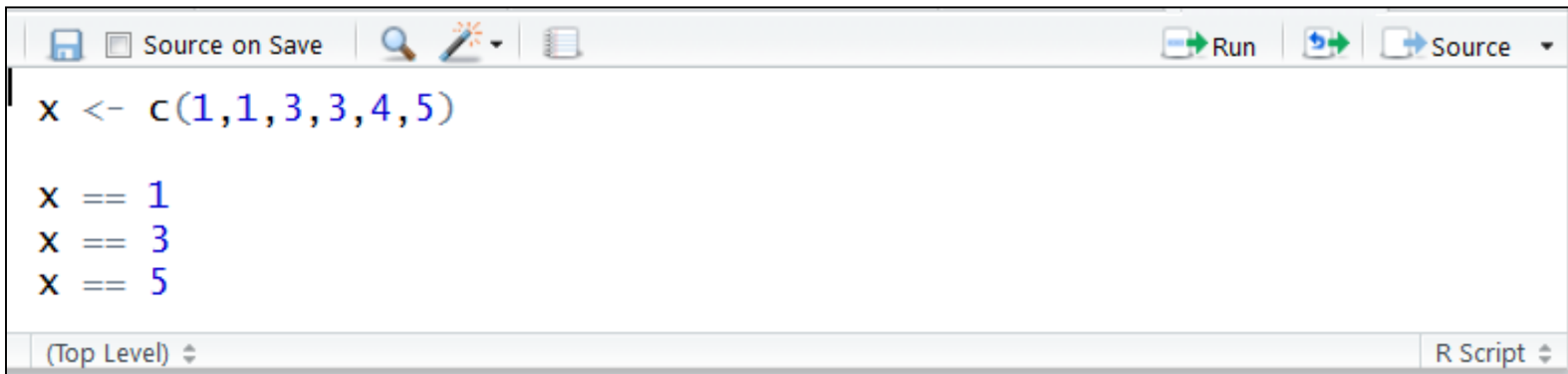
Data Analysis

- Let's calculate
 - Averages of TMAX and TMIN for each month.
 - Accumulations of PRCP for each month.
 - Averages of TMAX and TMIN for wet/dry days.
 - wet day: $PRCP > 0$.
 - dry day : $PRCP == 0$



Data Analysis

- Choosing a part of vector by condition.



```
x <- c(1,1,3,3,4,5)

x == 1
x == 3
x == 5
```

The screenshot shows an R script editor window. The title bar includes icons for 'Source on Save', 'Run', and 'Source'. The script content is as follows:

- Variable 'x' is a vector of length 6.
- 'x == 1' : every component is examined to be equal to 1.
If true, the result is represented as TRUE, otherwise, FALSE.



Data Analysis

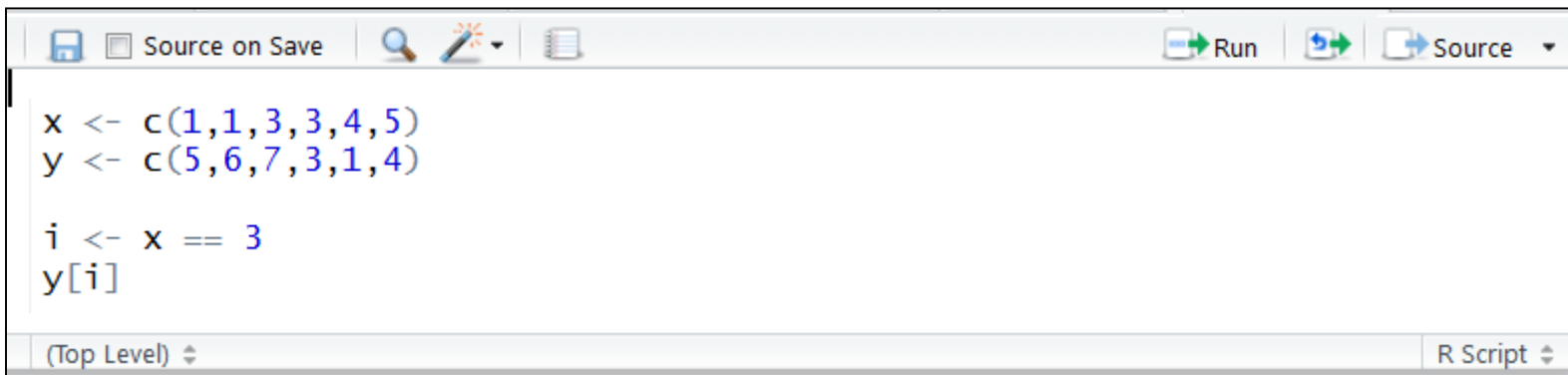
- Choosing a part of vector by condition.

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- c(1,1,3,3,4,5)
>
> x == 1
[1] TRUE TRUE FALSE FALSE FALSE FALSE
> x == 3
[1] FALSE FALSE TRUE TRUE FALSE FALSE
> x == 5
[1] FALSE FALSE FALSE FALSE FALSE TRUE
> |
```



Data Analysis

- Choosing a part of vector by condition.



```
x <- c(1,1,3,3,4,5)
y <- c(5,6,7,3,1,4)

i <- x == 3
y[i]
```

The screenshot shows an R script editor window. The title bar includes icons for save, source on save, search, and run. The main text area contains the R code shown above. The status bar at the bottom indicates '(Top Level)' and 'R Script'.

- Variable ‘i’ is a vector whose values are TRUE or FALSE.
- ‘y[i]’ is a sub-vector of ‘y’ consisting of components with TRUE.



Data Analysis

- Choosing a part of vector by condition.

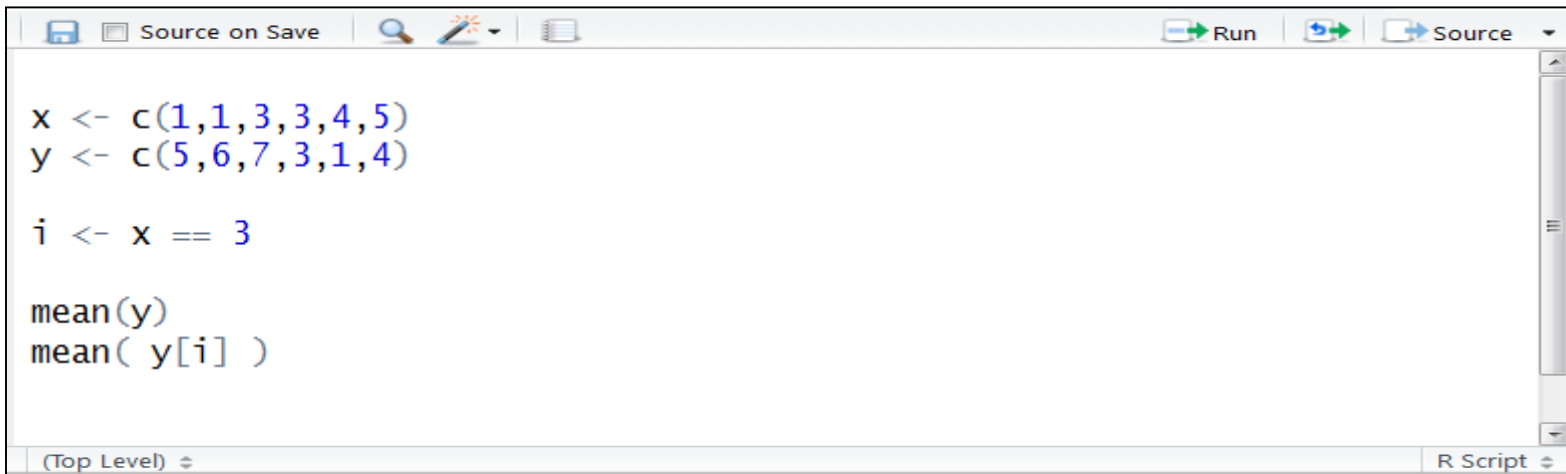
```
Console C:/APCC_TP_SDM_Dec2015/
> x <- c(1,1,3,3,4,5)
> y <- c(5,6,7,3,1,4)
>
> i <- x == 3
>
> i
[1] FALSE FALSE TRUE TRUE FALSE FALSE
>
> y[i]
[1] 7 3
> |
```

The 3rd and 4th of 'i' components are TRUE.

By 'i', the 3rd and 4th components of 'y' are chosen!

Data Analysis

- Averaging



```
Source on Save | Run | Source
x <- c(1,1,3,3,4,5)
y <- c(5,6,7,3,1,4)

i <- x == 3

mean(y)
mean( y[i] )
(Top Level) | R Script
```

- Function 'mean' is used.
- mean(y): averaging all components
- mean(y[i]): averaging the chosen components

Data Analysis

- Averaging

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- c(1,1,3,3,4,5)
> y <- c(5,6,7,3,1,4)
>
> i <- x == 3
>
> y
[1] 5 6 7 3 1 4
> y[i]
[1] 7 3
>
> mean(y)
[1] 4.333333
> mean( y[i] )
[1] 5
> |
```

Averaging!

Data Analysis

- Exercise
 - Choose part of 'y' corresponding to $x == 1$
 - Average the part.



Data Analysis

- Choosing and averaging column of matrix

```
Source on Save | Run | Source
M <- matrix( data = c(1,1,2,2,2,1,2,3,4,5) ,
              nrow = 5 , ncol = 2 ,
              dimnames = list( NULL , c( "group" , "value" ) ) )

i <- M["group"] == 1

M[i,"value"]

mean( M[i,"value"] )

(Top Level) | R Script
```



Data Analysis

- Choosing and averaging column of matrix

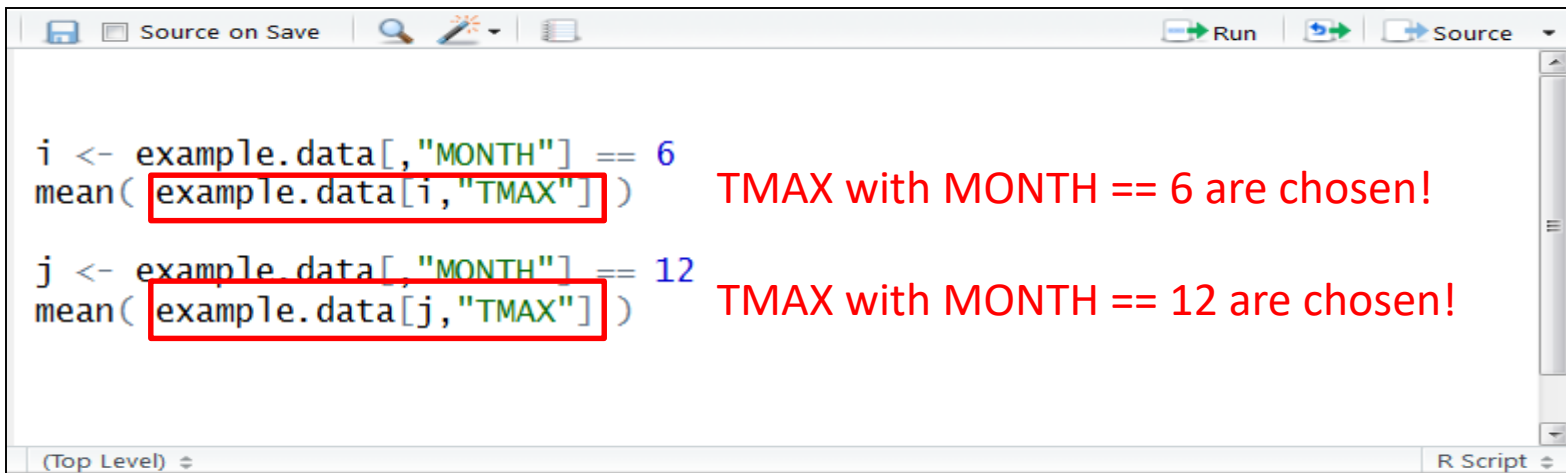
```
Console C:/APCC_TP_SDM_Dec2015/
> M <- matrix( data = c(1,1,2,2,2,1,2,3,4,5) ,
+             nrow = 5 , ncol = 2 ,
+             dimnames = list( NULL , c( "group" , "value" ) ) )
>
> M
      group value
[1,]      1      1
[2,]      1      2
[3,]      2      3
[4,]      2      4
[5,]      2      5
>
> i <- M[,"group"] == 1
>
> i
[1] TRUE  TRUE FALSE FALSE FALSE
> M[i,"value"]
[1] 1 2
> mean( M[i,"value"] )
[1] 1.5
> |
```

'value' with 'group' == 1 are chosen!



Data Analysis

- Return to **example.data**
- Calculating averages of TMAX for Jun. and Dec.



```
i <- example.data[, "MONTH" == 6  
mean( example.data[i, "TMAX"] )  
  
j <- example.data[, "MONTH" == 12  
mean( example.data[j, "TMAX"] )
```

TMAX with MONTH == 6 are chosen!

TMAX with MONTH == 12 are chosen!



Data Analysis

- Calculating averages of TMAX for Jun. and Dec.

```
Console C:/APCC_TP_SDM_Dec2015/ ↵
>
> i <- example.data[, "MONTH"] == 6
> mean( example.data[i, "TMAX"] )
[1] 28.92667
>
> j <- example.data[, "MONTH"] == 12
> mean( example.data[j, "TMAX"] )
[1] 5.909677
> |
```



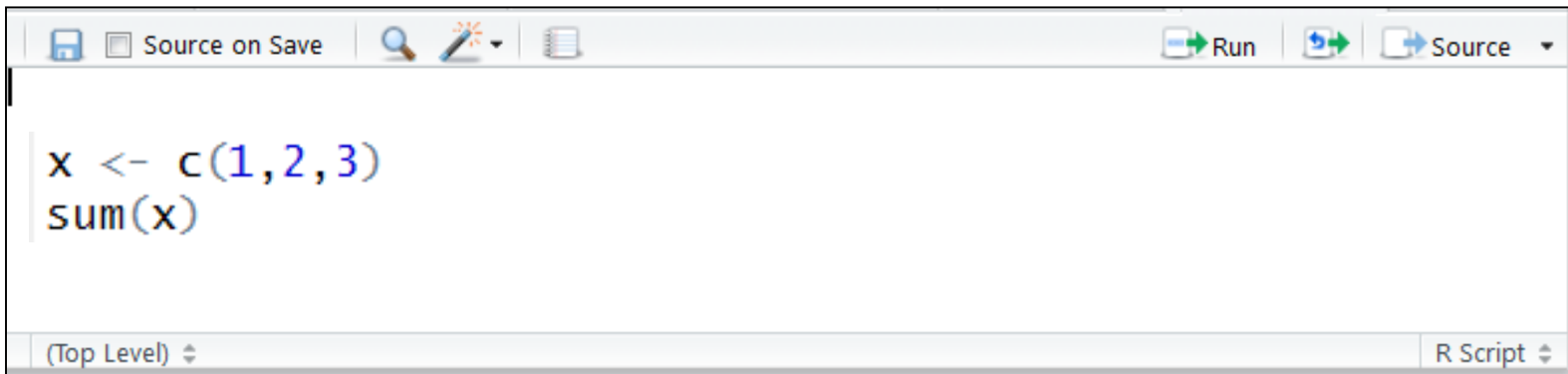
Data Analysis

- Exercise:
 - Calculate averages of TMIN for Jun. and Dec.



Data Analysis


- **sum()**



```
Source on Save | Search | Run | Source
```

```
x <- c(1,2,3)  
sum(x)
```

(Top Level) | R Script



```
Console C:/APCC_TP_SDM_Dec2015/
```

```
> x <- c(1,2,3)  
> sum(x)  
[1] 6  
> |
```



Data Analysis

- Calculating the accumulation of PRCP

```
Source on Save | Run | Source
i <- example.data[, "MONTH"] == 6
sum( example.data[i, "PRCP"] )    PRCP with MONTH == 6 are chosen!

j <- example.data[, "MONTH"] == 12
sum( example.data[j, "PRCP"] )    PRCP with MONTH == 12 are chosen!

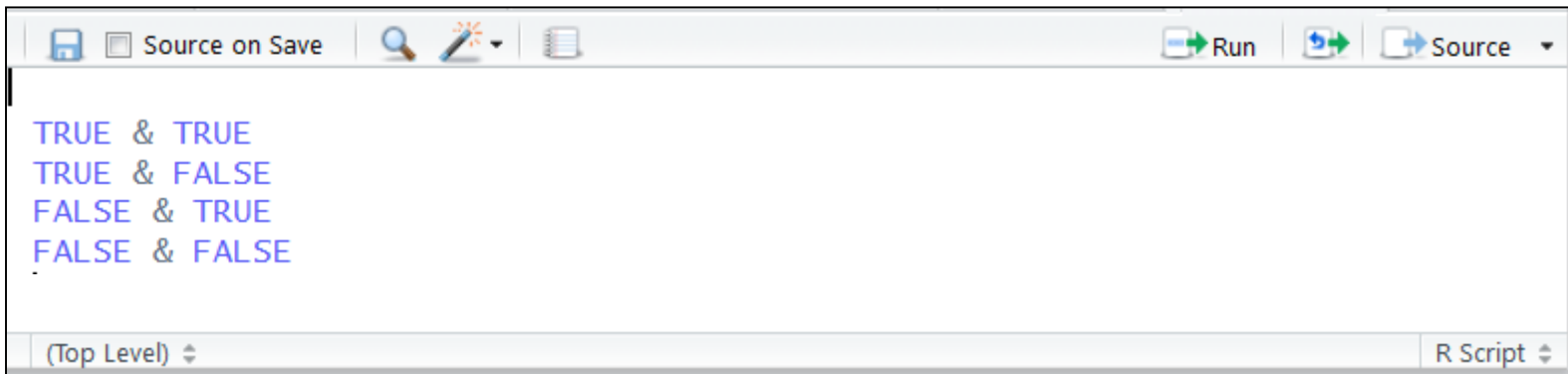
(Top Level) | R Script
```

```
Console C:/APCC_TP_SDM_Dec2015/
> i <- example.data[, "MONTH"] == 6
> sum( example.data[i, "PRCP"] )
[1] 187.2
>
> j <- example.data[, "MONTH"] == 12
> sum( example.data[j, "PRCP"] )
[1] 12
> |
```



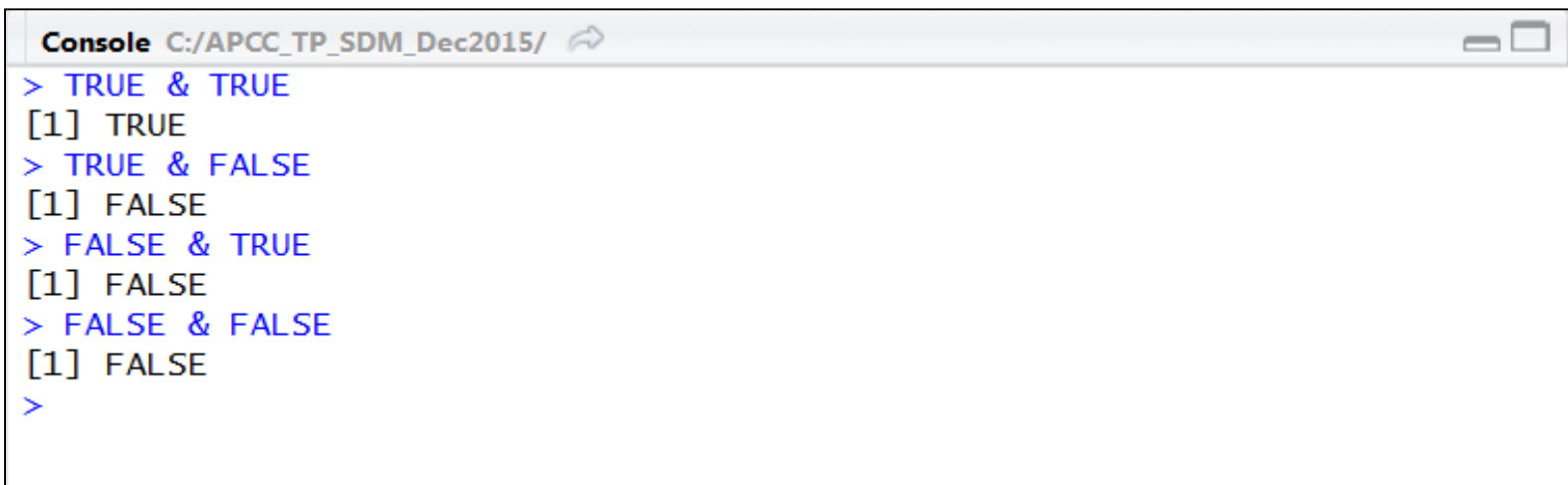
Data Analysis

- Operator '&'



A screenshot of an R script editor window. The window has a toolbar at the top with icons for saving, searching, and running. Below the toolbar, the script contains four lines of code: TRUE & TRUE, TRUE & FALSE, FALSE & TRUE, and FALSE & FALSE. The status bar at the bottom indicates '(Top Level)' and 'R Script'.

```
TRUE & TRUE
TRUE & FALSE
FALSE & TRUE
FALSE & FALSE
```



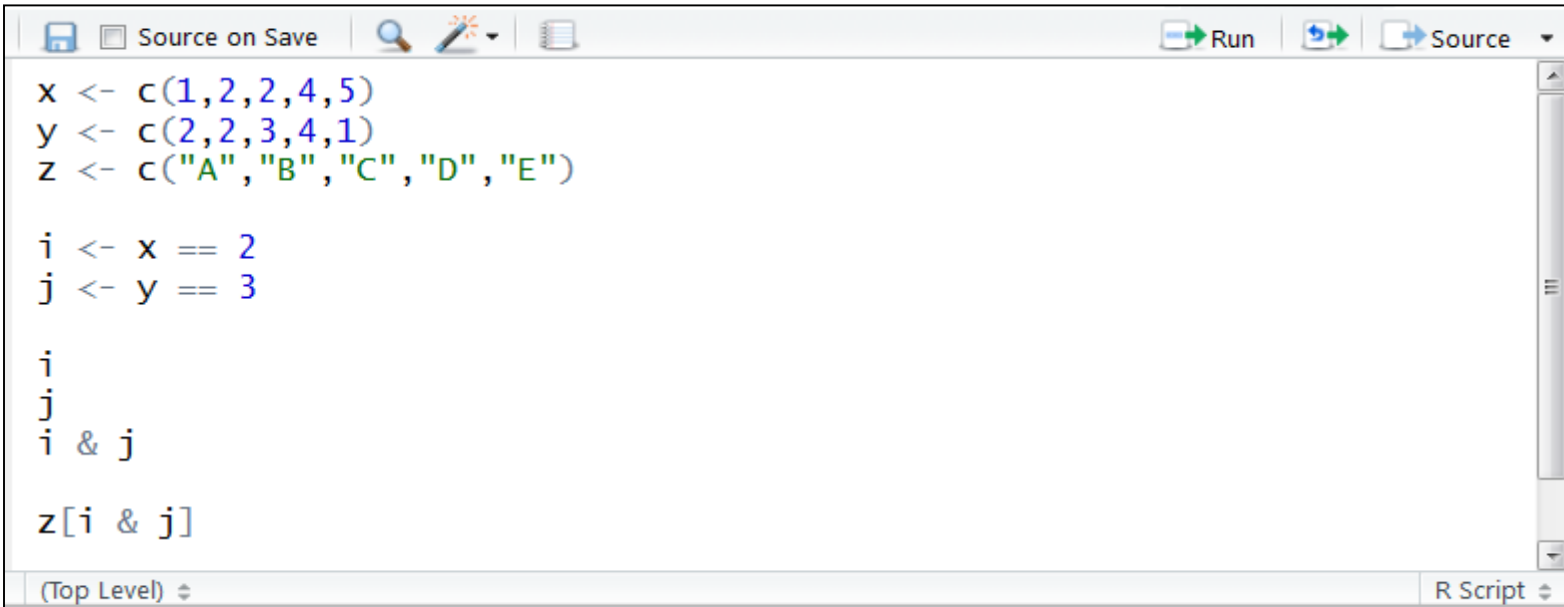
A screenshot of an R console window. The window title is 'Console C:/APCC_TP_SDM_Dec2015/'. The console shows the output of the four logical expressions from the script above: TRUE & TRUE returns [1] TRUE, TRUE & FALSE returns [1] FALSE, FALSE & TRUE returns [1] FALSE, and FALSE & FALSE returns [1] FALSE. The prompt '>' is visible at the end of each line.

```
> TRUE & TRUE
[1] TRUE
> TRUE & FALSE
[1] FALSE
> FALSE & TRUE
[1] FALSE
> FALSE & FALSE
[1] FALSE
>
```



Data Analysis

- Operator '&'



```
x <- c(1,2,2,4,5)
y <- c(2,2,3,4,1)
z <- c("A","B","C","D","E")

i <- x == 2
j <- y == 3

i
j
i & j

z[i & j]
```

(Top Level) ↕ R Script ↕



Data Analysis

- Operator '&'

```
Console C:/APCC_TP_SDM_Dec2015/
> x <- c(1,2,2,4,5)
> y <- c(2,2,3,4,1)
> z <- c("A","B","C","D","E")
>
> i <- x == 2
> j <- y == 3
>
> i
[1] FALSE TRUE TRUE FALSE FALSE
> j
[1] FALSE FALSE TRUE FALSE FALSE
> i & j
[1] FALSE FALSE TRUE FALSE FALSE
> z[i & j]
[1] "C"
```

Operation is done component-wisely!

The components of 'z' with x == 2 and y == 3 are chosen

Data Analysis

- Average of TMAX for wet day in June.

```
Source on Save | Run | Source
```

```
i <- example.data[, "MONTH"] == 6  
j <- example.data[, "PRCP"] > 0  
k <- i & j  
mean( example.data[k, "TMAX"] ) TMAX with 'MONTH == 6 and PRCP > 0' are chosen
```

(Top Level) | R Script

```
Console C:/APCC_TP_SDM_Dec2015/
```

```
> i <- example.data[, "MONTH"] == 6  
> j <- example.data[, "PRCP"] > 0  
> k <- i & j  
>  
> mean( example.data[k, "TMAX"] )  
[1] 27.33  
> |
```



Data Analysis

- Exercise: Calculate the others:
 - Average of TMAX for dry day in June,
 - Average of TMAX for wet day in Dec.,
 - Average of TMAX for dry day in Dec.

 - The same averages of TMIN.



Data Analysis

- `mean()`: average
 - `sum()`: total sum,
 - `sd()`: standard deviation,
 - `max()`: maximum,
 - `min()`: minimum.
-
- Their usages are the same to `mean()`.



Data Analysis

- Exercise: Calculate
 - Standard deviation of TMAX in June,
 - Standard deviation of TMAX in Dec.,
 - Minimum of TMIN in Dec.,
 - Maximum of PRCP in June.



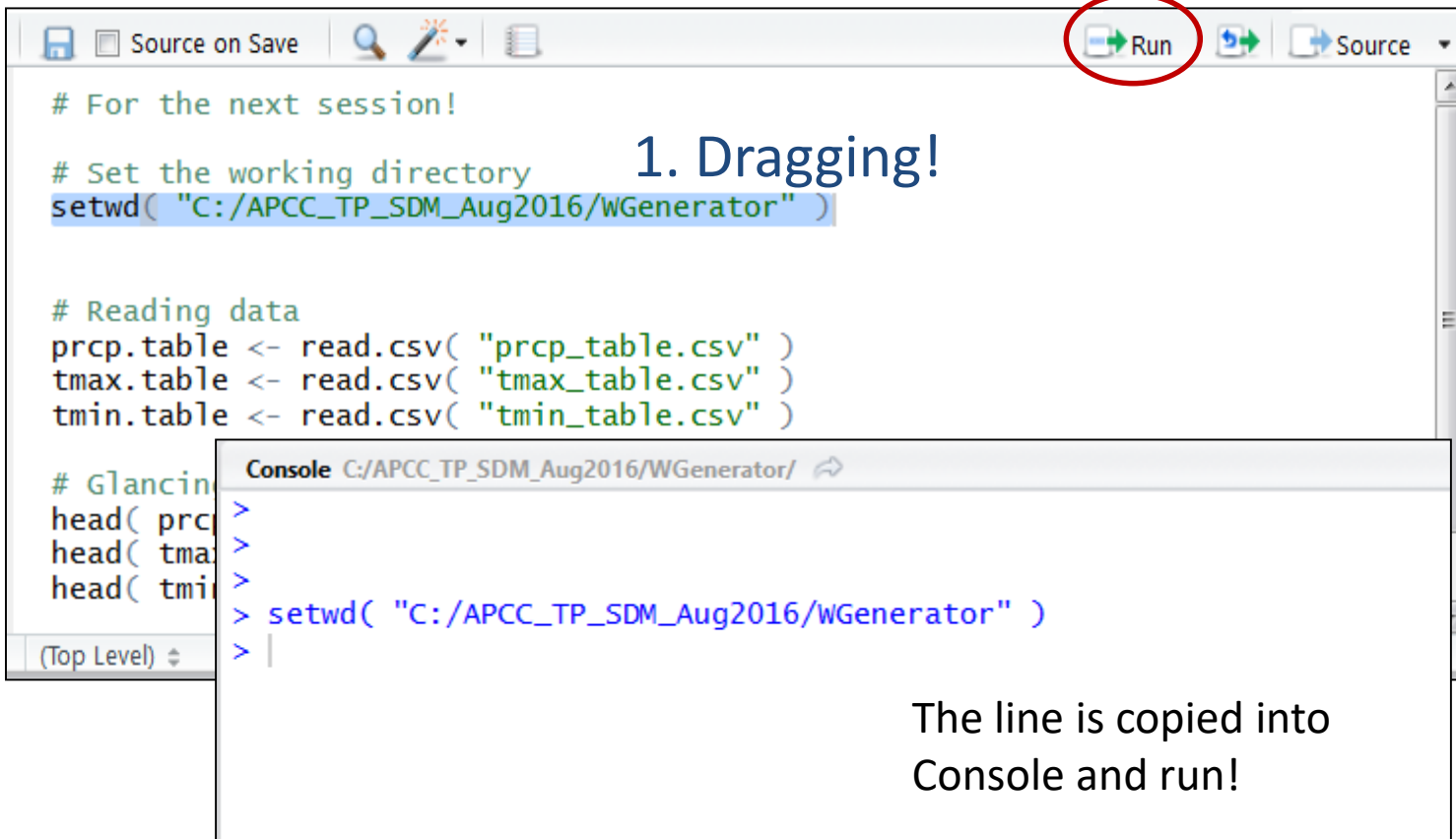
Contents

- Preparation and installation
- Variables and Operations
- Data Analysis using R
- **Reading your Observation Data**



Data Reading

- First, set working directory. **2. Click!**



The screenshot shows an RStudio editor window with the following R code:

```
# For the next session!  
# Set the working directory  
setwd( "C:/APCC_TP_SDM_Aug2016/WGenerator" )  
  
# Reading data  
prcp.table <- read.csv( "prcp_table.csv" )  
tmax.table <- read.csv( "tmax_table.csv" )  
tmin.table <- read.csv( "tmin_table.csv" )  
  
# Glancing  
head( prcp )  
head( tmax )  
head( tmin )
```

The line `setwd("C:/APCC_TP_SDM_Aug2016/WGenerator")` is highlighted in blue. A red circle highlights the **Run** button in the top right corner of the editor window. The text **1. Dragging!** is placed above the highlighted line, and **2. Click!** is placed above the **Run** button.

The console window shows the following output:

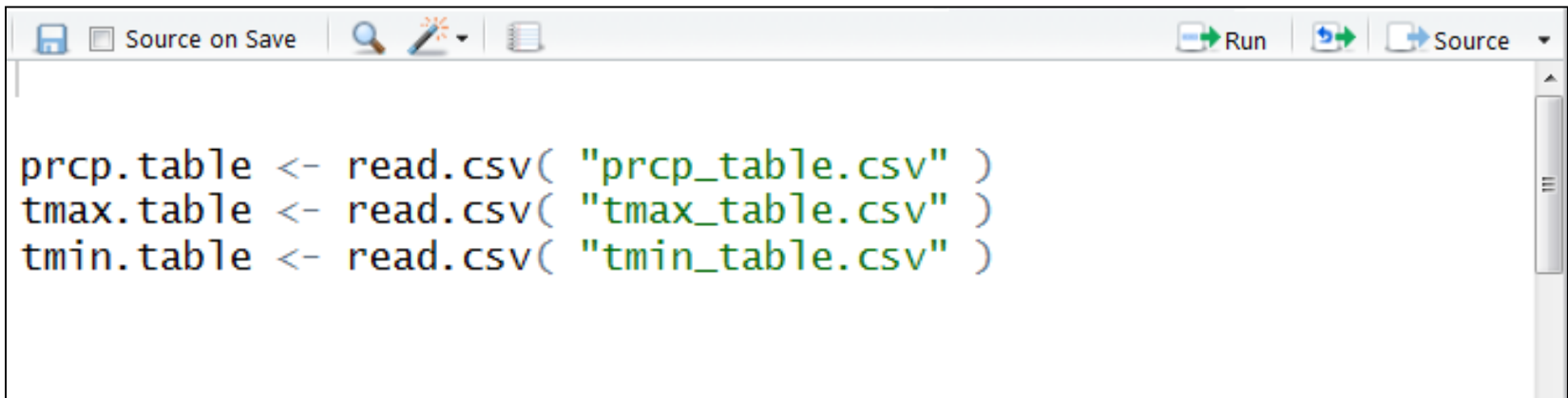
```
Console C:/APCC_TP_SDM_Aug2016/WGenerator/  
>  
>  
>  
> setwd( "C:/APCC_TP_SDM_Aug2016/WGenerator" )  
> |
```

The text **The line is copied into Console and run!** is placed below the console window.



Data Reading

- Read data files.



```
prcp.table <- read.csv( "prcp_table.csv" )
tmax.table <- read.csv( "tmax_table.csv" )
tmin.table <- read.csv( "tmin_table.csv" )
```

- The files have to be in the working directory!



Data Reading

- Glancing at the data.

```
Console C:/APCC_TP_SDM_Dec2015/
> head( prcp.table )
  YEAR MONTH DAY Andong Bongwhoa Tawbaekk Yeongju Mungyeong Uisung Gumi Daegu
1 1988     1   1     0     0.0     0.0     0.0         0     0.0  0.0  0.0
2 1988     1   2     0     0.0     0.0     0.0         0     0.0  0.0  0.0
3 1988     1   3     0     0.0     0.0     0.0         0     0.0  0.0  0.0
4 1988     1   4     1     0.3     0.2     0.2         0     0.8  0.7  0.9
5 1988     1   5     0     0.0     0.0     0.0         0     0.0  0.0  0.0
6 1988     1   6     0     0.0     0.0     0.0         0     0.0  0.0  0.0
  Yeongcheon Geochang Sancheong Hapcheon Milyang Jinju
1           0         0.0         0.0         0.0         0.0         0
2           0         0.0         0.0         0.0         0.0         0
3           0         0.0         0.0         0.0         0.0         0
4           1         0.8         2.5         0.8         2.4         1
5           0         0.0         0.0         0.0         0.0         0
6           0         0.0         0.0         0.0         0.0         0
> |
```

* NOTE: The display depends on your data set.

The End of Session 1

Welcome any Questions.

