

APCC 2016 Statistical Downscaling Training Program Seasonal Prediction Downscaling

August 2016

Moosup Kim



Contents

- Seasonal Prediction Downscaling using Weather Generator
 - Seasonal Prediction and Downscaling Scheme,
 - Running Weather Generator.
- Analysis of Simulation Data



Seasonal Prediction

- Seasonal Prediction
 - APCC provides Seasonal Prediction every month,
 - 3 months ahead / 6 months ahead,
 - Precipitation, Temperature, and others.
 - <http://www.apcc21.org/ser/outlook.do?lang=en>



Seasonal Prediction

- Seasonal Prediction
 - Probabilistic MME Forecast.
 - Check your Basin Outlook!

• Probabilistic MME Forecast

TOP

SEASON/MONTH	VARIABLE	REGION	
ASO ▾	T2m Temperature ▾	South America ▾	search



Seasonal Prediction

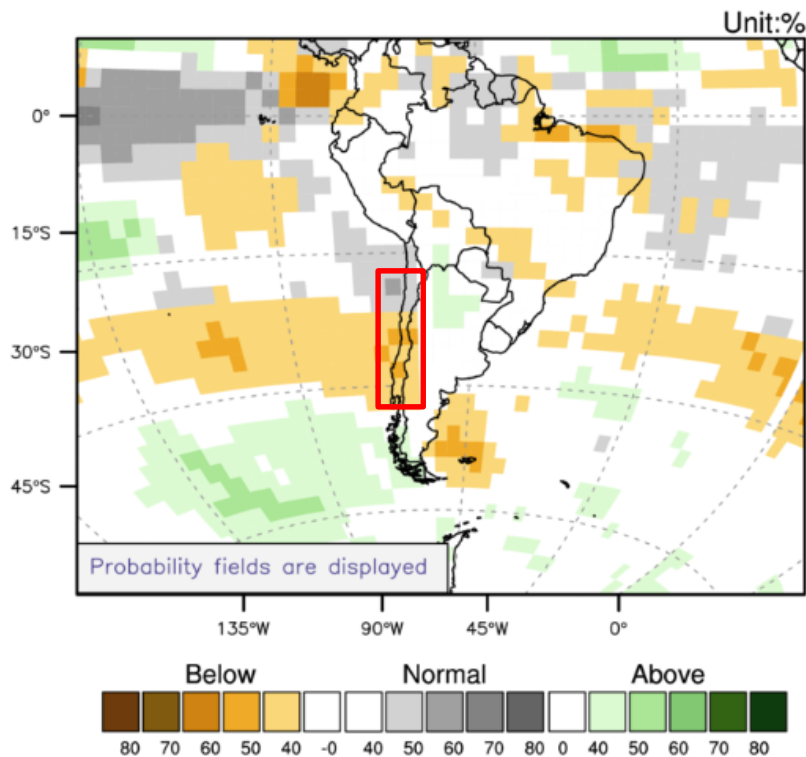
- What is Seasonal Prediction, if
 - Season: ASO (August to October),
 - Variables: Precipitation, (2m) Temperature,
 - Basin: Santiago, Concepcion, Pureto Montt in Chile.



SEASON/MONTH	VARIABLE	REGION
ASO	Precipitation	South America

search

Precipitation for August-October 2016

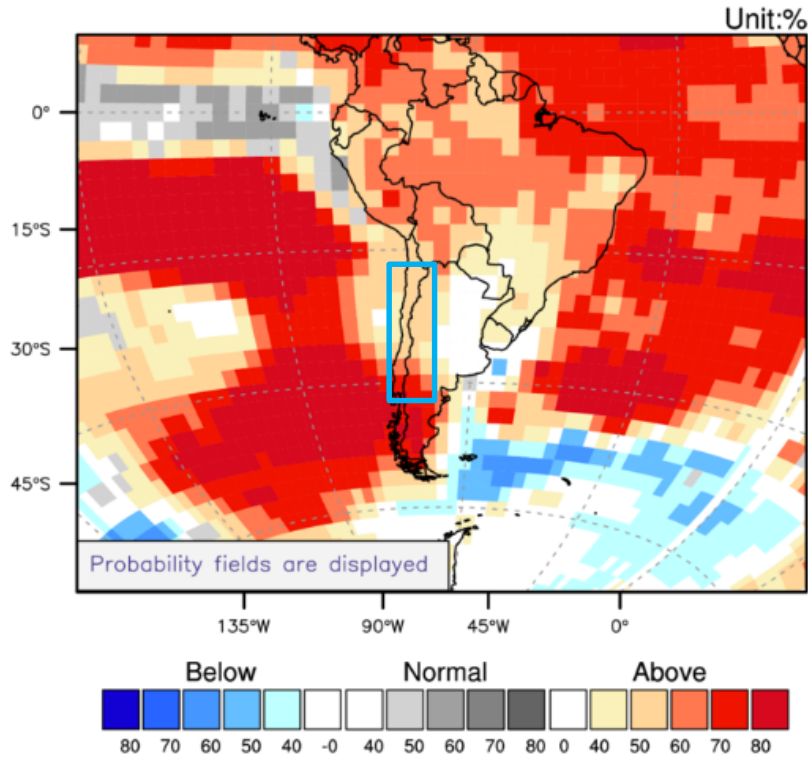


- Station location in the box,
- Precipitation: prevailing Below

SEASON/MONTH	VARIABLE	REGION
ASO	T2m Temperature	South America

search

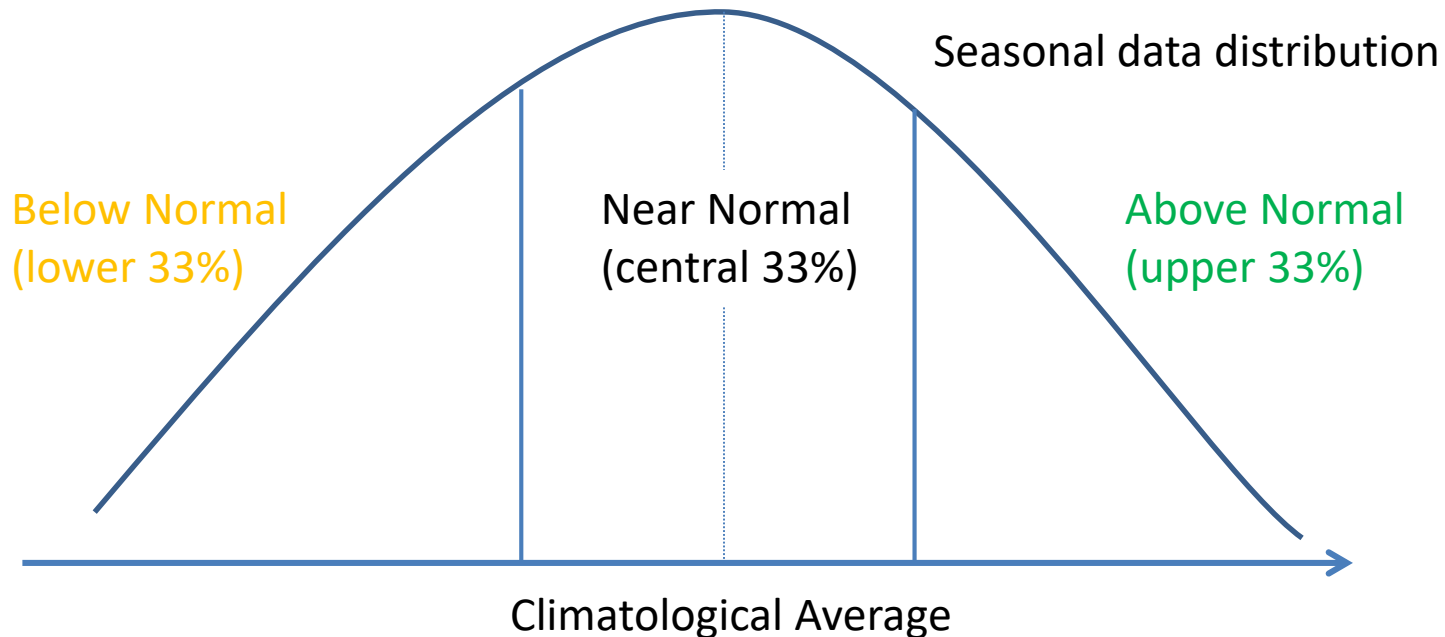
Temperature at 2m for August-October 2016



- Station location in the box,
- 2m Temperature: Above

Seasonal Prediction

- Seasonal Prediction
 - Below, Normal, Above,
 - Normal = Near Normal:
Around Climatological Average!



Seasonal Prediction

- Seasonal Prediction
 - Above = Above Normal:
 - Temperature: Relatively Warm/Hot,
 - Precipitation: Relatively Wet.
 - Below = Below Normal:
 - Temperature: Relatively Cool/Cold,
 - Precipitation: Relatively Dry.



Seasonal Prediction

- Seasonal Prediction (Chile case)
 - Precipitation: Below Normal
 - Temperature: Above Normal
- More Detailed Information! -> Downscaling!
 - Daily time scale and with Focus on Basin.



Seasonal Prediction

- Now, Check Seasonal Outlook for your Basin!



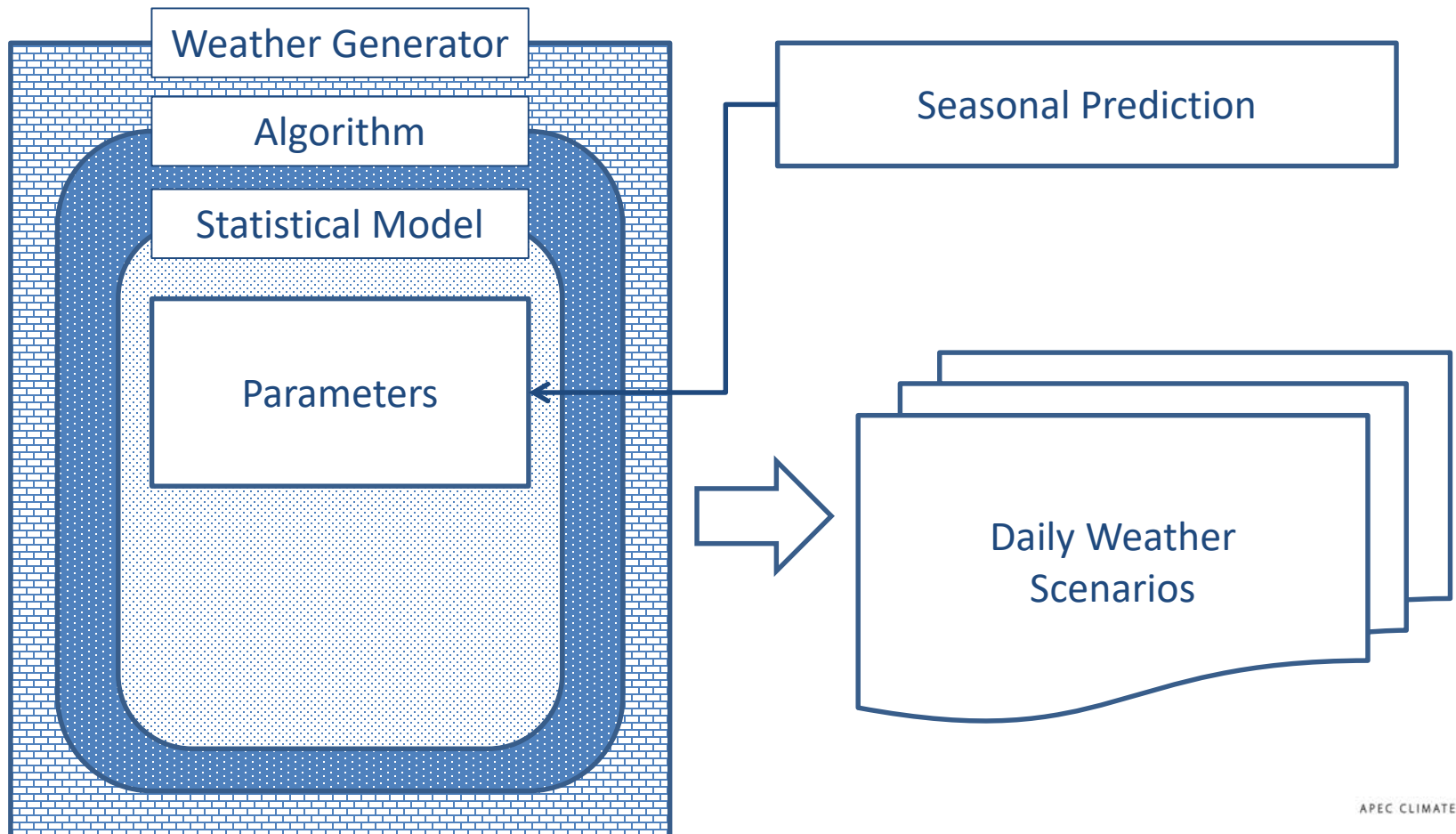
Downscaling Scheme

- Weather Generator calibrated by Climatology!
 - Inter-annual variability is not considered!
- Re-calibration of Weather Generator according to Seasonal Prediction!
 - Adjusting parameter values!



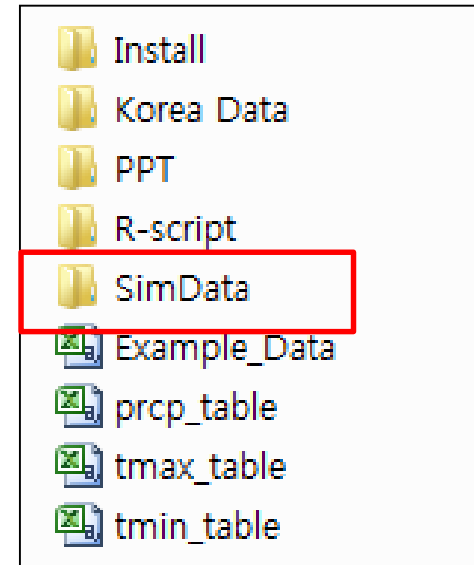
Downscaling Scheme

- General Scheme



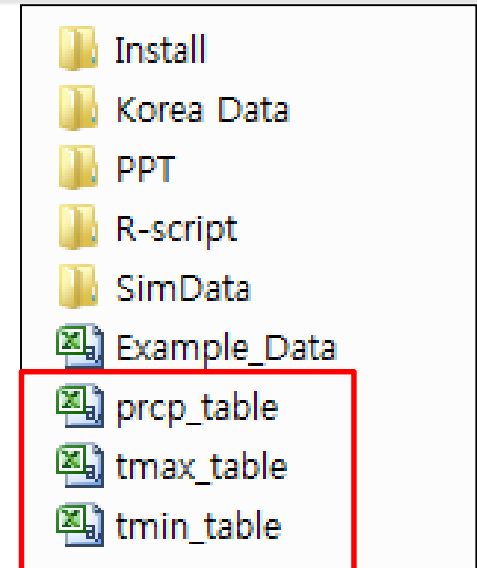
Running WG

- Preparation for Running WG.
 - Working directory in C drive:
**C:\APCC_TP_SDM_Aug2016\
WGenerator**
 - Simulation data repository:
**C:\APCC_TP_SDM_Aug2016\
WGenerator
\SimData**



Running WG

- Preparation (cont.)
 - Check your observation data in **C:\APCC_TP_SDM_Aug2016\WGenerator**
 - Start R Studio.



Running WG

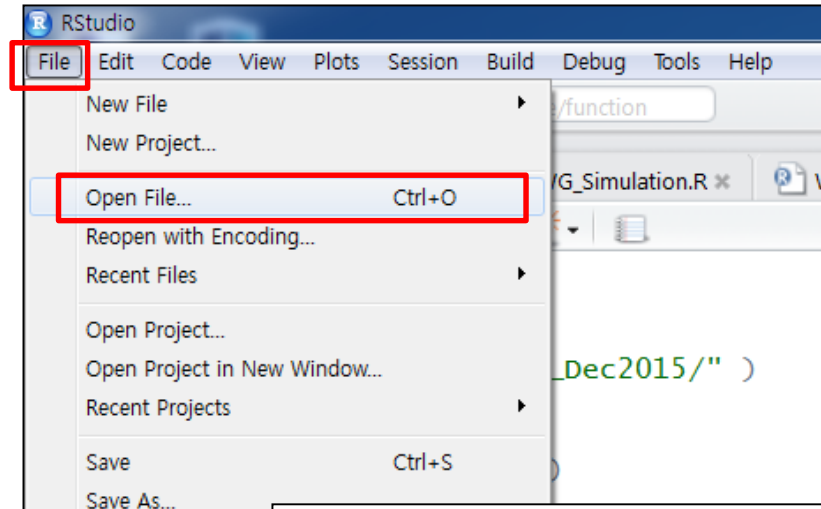
- Preparation (cont.)
 - Open R-code file **WG_Simulation.R**
 - File menu > Open File
 - Selecting the file in
C:\APCC_TP_SDM_Aug2016\WGenerator\R-script



Running WG

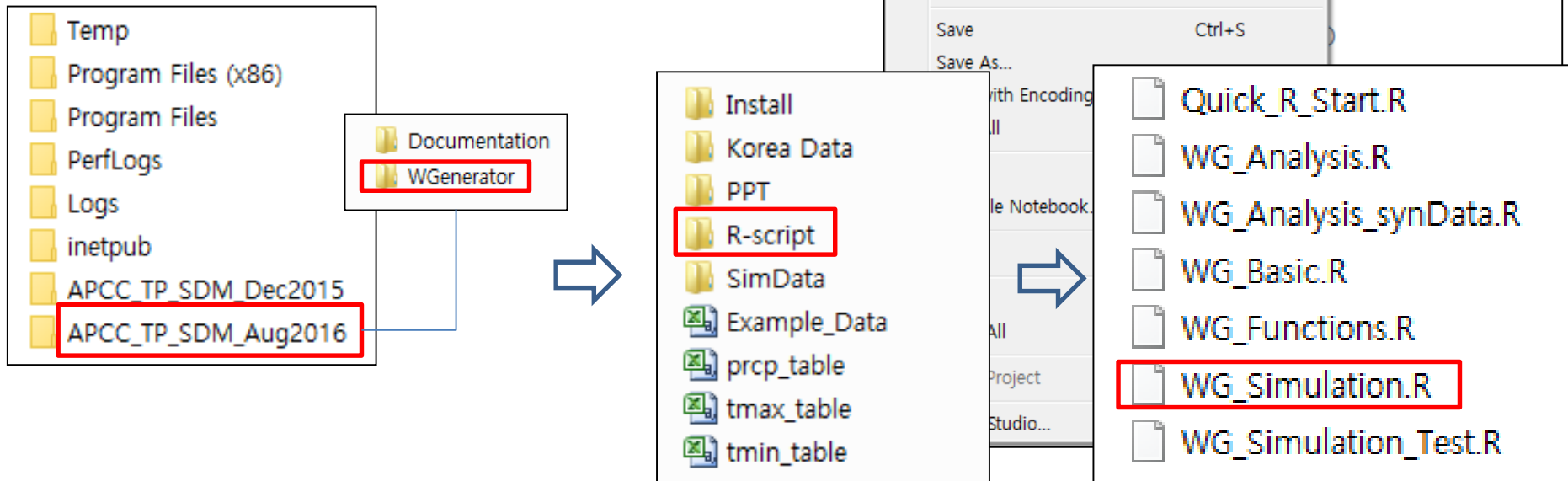
- Open **WG_Simulation.R**.

1. Click File



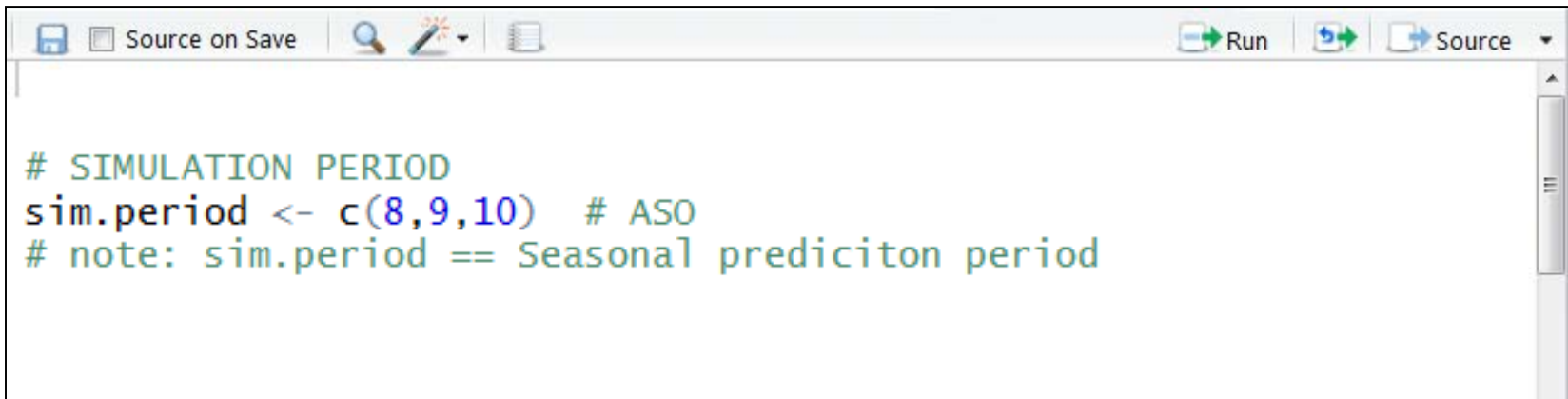
2. Click Open File...

- 3. Find **WG_Simulation.R** by double click. (Searching start C:\)



Running WG

- Set simulation period.

A screenshot of an R script editor window. The window has a toolbar at the top with icons for 'Source on Save', search, and editing, and buttons for 'Run' and 'Source'. The main text area contains the following R code:

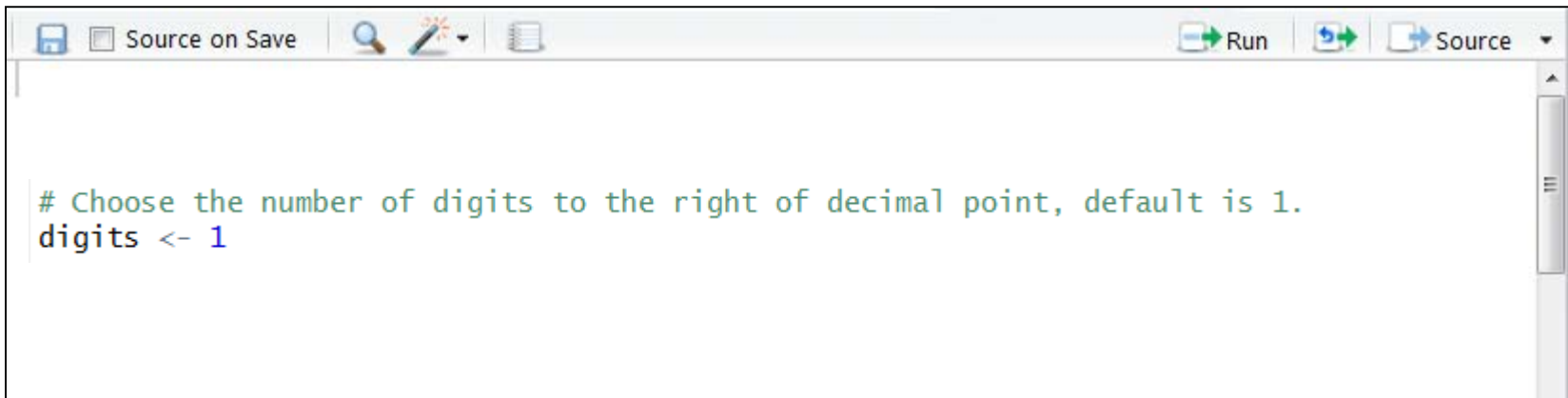
```
# SIMULATION PERIOD  
sim.period <- c(8,9,10) # ASO  
# note: sim.period == Seasonal prediction period
```

– Simulation Period = Seasonal Prediction Period!



Running WG

- Setting for decimal
 - Setting the number of digits to the right of decimal point in simulation data.

A screenshot of a code editor window. The window has a toolbar at the top with icons for save, search, and run. The main area contains two lines of code: a comment and an assignment statement.

```
# Choose the number of digits to the right of decimal point, default is 1.  
digits <- 1
```



Running WG

- Seasonal Prediction

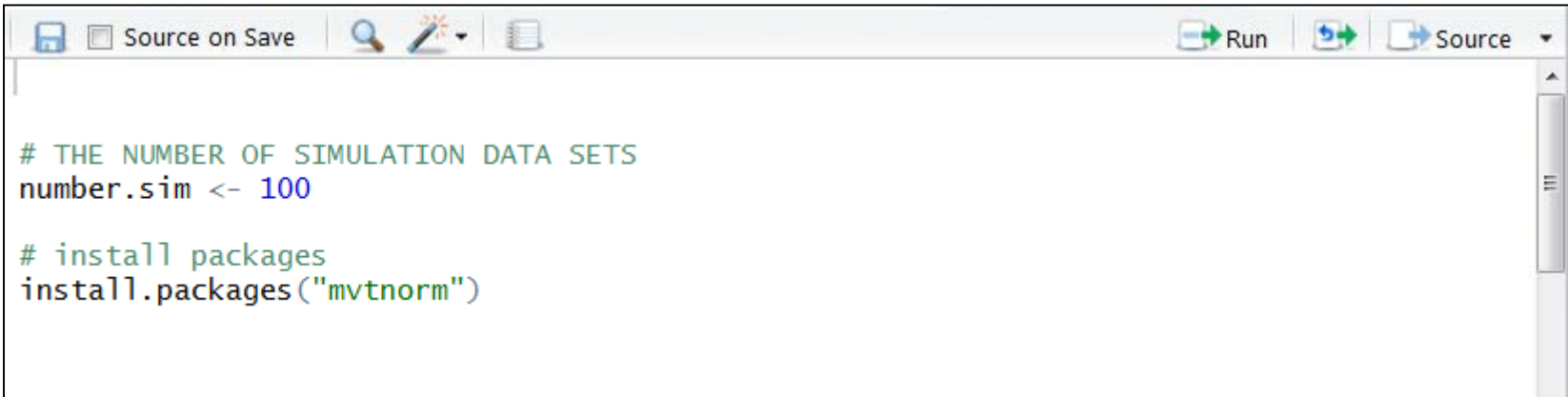
```
Source on Save | Run | Source
```

```
# SEASONAL PREDICTION ( Temperature , Precipitation )  
# The Prediction result must be one of "BelowNormal", "NearNormal", "AboveNormal".  
# Choose one line and run it:  
seasonal.prediction <- c( "NearNormal" , "NearNormal" )  
seasonal.prediction <- c( "BelowNormal" , "NearNormal" )  
seasonal.prediction <- c( "AboveNormal" , "NearNormal" )  
seasonal.prediction <- c( "NearNormal" , "BelowNormal" )  
seasonal.prediction <- c( "NearNormal" , "AboveNormal" )  
seasonal.prediction <- c( "BelowNormal" , "BelowNormal" )  
seasonal.prediction <- c( "BelowNormal" , "AboveNormal" )  
seasonal.prediction <- c( "AboveNormal" , "AboveNormal" )  
seasonal.prediction <- c( "AboveNormal" , "BelowNormal" )
```



Running WG

- Next,



```
# THE NUMBER OF SIMULATION DATA SETS
number.sim <- 100

# install packages
install.packages("mvtnorm")
```



Running WG

- Loading data

```
Source on Save Run Source
# WORK DIRECTORY SETTING!
setwd( "C:/APCC_TP_SDM_Aug2016/WGenerator/" )

# Including another R-script
source("R-script/WG_Functions.R")

# Observation loading
prcp.table0 <- as.matrix( read.csv( "prcp_table.csv" ) )
tmax.table0 <- as.matrix( read.csv( "tmax_table.csv" ) )
tmin.table0 <- as.matrix( read.csv( "tmin_table.csv" ) )

colnames(prcp.table0)[1:3] <- c( "YEAR" , "MONTH" , "DAY" )
colnames(tmax.table0)[1:3] <- c( "YEAR" , "MONTH" , "DAY" )
colnames(tmin.table0)[1:3] <- c( "YEAR" , "MONTH" , "DAY" )

# Station names: making the station name be same with those of precipitation data.
site.names <- colnames( prcp.table0[,-(1:3)] )
colnames( tmax.table0 )[-(1:3)] <- site.names
colnames( tmin.table0 )[-(1:3)] <- site.names

d <- length(site.names)
```



Running WG

- Typo process,

```
Source on Save Run Source
# Caution: characters and symbols in the data sets are converted to missing values!
if( is.character( prcp.table0[1,1] ) )
{
  prcp.table1 <- matrix( nrow = nrow(prcp.table0) , ncol = ncol(prcp.table0) ,
                        dimnames = list( NULL , colnames(prcp.table0) ) )
  for( j in 1:ncol(prcp.table0) )
  {
    prcp.table1[,j] <- as.numeric( prcp.table0[,j] )
  }
  prcp.table0 <- prcp.table1
  rm( prcp.table1 )
}
if( is.character( tmax.table0[1,1] ) )
{
  tmax.table1 <- matrix( nrow = nrow(tmax.table0) , ncol = ncol(tmax.table0) ,
                        dimnames = list( NULL , colnames(tmax.table0) ) )
  for( j in 1:ncol(tmax.table0) )
  {
    tmax.table1[,j] <- as.numeric( tmax.table0[,j] )
  }
  tmax.table0 <- tmax.table1
  rm( tmax.table1 )
}
if( is.character( tmin.table0[1,1] ) )
{
  tmin.table1 <- matrix( nrow = nrow(tmin.table0) , ncol = ncol(tmin.table0) ,
                        dimnames = list( NULL , colnames(tmin.table0) ) )
  for( j in 1:ncol(tmin.table0) )
  {
    tmin.table1[,j] <- as.numeric( tmin.table0[,j] )
  }
  tmin.table0 <- tmin.table1
  rm( tmin.table1 )
}
```



Running WG

- Missing values treatment.

```
Source on Save | Run | Source
# Modificatyng the observation data accroding to the selected seasonal prediction:
result <- modificating.data( sim.period , prcp.table0 , tmax.table0 , tmin.table0 )
prcp.table <- result$prcp.table
tmax.table <- result$tmax.table
tmin.table <- result$tmin.table
rm(result)

# MISSING VALUES TREATMENT

prcp.table[,site.names][ prcp.table[,site.names] < 0 ] <- NA
tmax.table[,site.names][ tmax.table[,site.names] < -50 | tmax.table[,site.names] > 212 ] <- NA
tmin.table[,site.names][ tmin.table[,site.names] < -50 | tmin.table[,site.names] > 212 ] <- NA

result <- missing.treatment( site.names , prcp.table , digits )
prcp.table <- result$prcp.table
avail <- result$avail
# Note: it can take a couple of minutes.
```



Running WG

- Final step.

```
Source on Save | 🔍 | 🚀 | 📄 | Run | ↻ | Source ▾

# The Period of observation data
s <- head(prcp.table[,"YEAR"],1)
t <- tail(prcp.table[,"YEAR"],1)

period <- s:t

result.detrrending <- detrrending( period , site.names , tmax.table , tmin.table , prcp.table )
fit.anomaly <- fitVAR( result.detrrending$anomaly , order = 2 )
# Note: detrrending and fit.anomlay can take a couple of minutes.

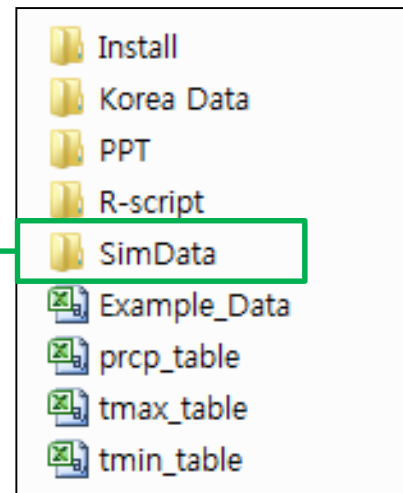
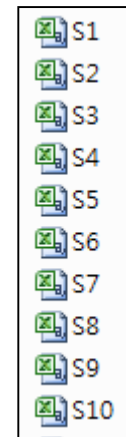
simulation.output <- simulation( seasonal.prediction , number.sim , sim.period ,
                                site.names , period , result.detrrending , fit.anomaly ,
                                prcp.table , tmax.table , tmin.table , digits = digits )
```



Running WG

- 100 sets of simulation scenario data will be generated in
**C:\APCC_TP_SDM_Aug2016\
WGenerator\SimData**
- Check that simulation data files well generated.

S1, S2, ... , S100



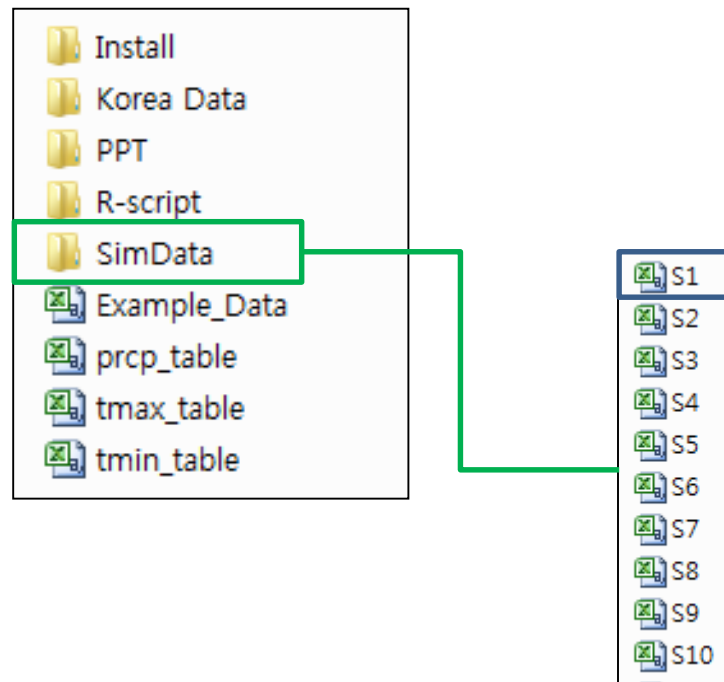
Contents

- Seasonal Prediction Downscaling using Weather Generator
- Analysis of Simulation Data



Simulation Data Analysis

- Look at the scenarios



Choose any one.
Open it by double click!



Simulation Data Analysis

- Look at the scenarios
 - PRCP (precipitation) , TMAX (max of daily temperature) , TMIN (min of daily temperature) of the sites (Andong, Bongwhoa, Tawbaekk)

	A	B	C	D	E	F	G	H	I	J	K	L
1	MONTH	DAY	PRCP_Andong	TMAX_Andong	TMIN_Andong	PRCP_Bongwhoa	TMAX_Bongwhoa	TMIN_Bongwhoa	PRCP_Tawbaekk	TMAX_Tawbaekk	TMIN_Tawbaekk	
2	1	1	1	0	0.8	-13.3	0	-0.8	-19.4	0	-2.7	-14.1
3	2	1	2	0.3	1.2	-5.2	0.3	0.8	-11.3	0.5	2.2	-6.4
4	3	1	3	8.5	1.7	-6.2	0.7	1	-11	0	1.1	-12.4
5	4	1	4	0	-0.1	-11.2	0	-2.5	-15.5	0	-1.6	-13
6	5	1	5	1.5	-1.1	-7.2	1.3	-2.9	-9.6	0	-6	-11
7	6	1	6	0	-2.9	-5.3	0.3	-4.3	-11.9	0.5	-4.7	-7.8
8	7	1	7	0	0.9	-7.1	0	-0.9	-10.6	0	-1.5	-11.5
9	8	1	8	0	4.2	-9.9	0	3.3	-13.7	0	1.1	-7.2
10	9	1	9	0	6.8	-11.6	0	7.1	-14.5	0	4.6	-8.8
11	10	1	10	0	6.9	-5.2	0	5.9	-6.8	0	2.7	-6
12	11	1	11	0	5.8	-2.6	0	3.3	-7.9	0	7.6	-4.8
13	12	1	12	0	7.9	-1.9	0	8.1	-3.5	0	5.5	0.5

The result depends on your data



Simulation Data Analysis

- Question
 - Seasonal precipitation accumulation: mean, standard deviation, quantiles?
 - If the season is wet, choose a high threshold. How many days precipitation is above the threshold?
 - If the season is dry, how many days are dry?



Simulation Data Analysis

- Question
 - Seasonal TMAX/TMIN average: mean, standard deviation, quantiles?
 - If the season is warm/hot, choose a high threshold. How many days TMAX is above the threshold?
 - If the season is cool/cold, choose a low threshold. How many days TMIN is below the threshold?



Simulation Data Analysis

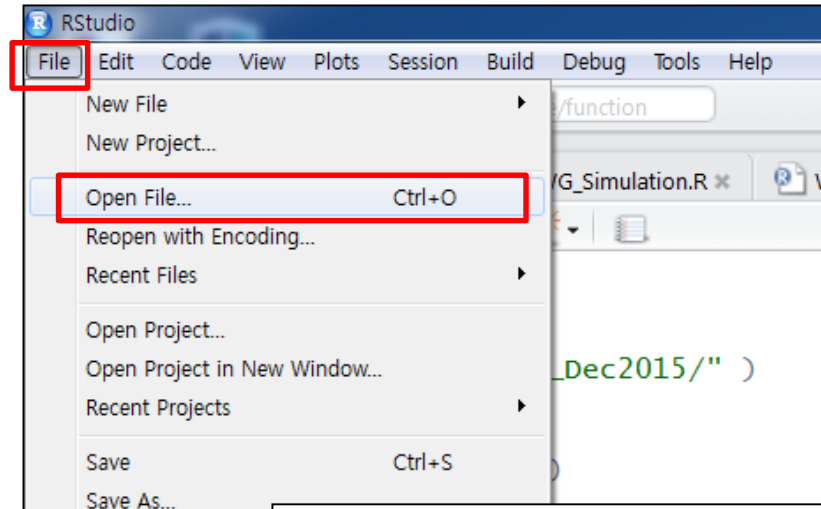
- Open **WG_Analysis.R**
 - File menu > Open File
 - Selecting the file in
**C:\APCC_TP_SDM_Aug2016\WGenerator\
R-script**
 - Next page presents illustration.



Simulation Data Analysis

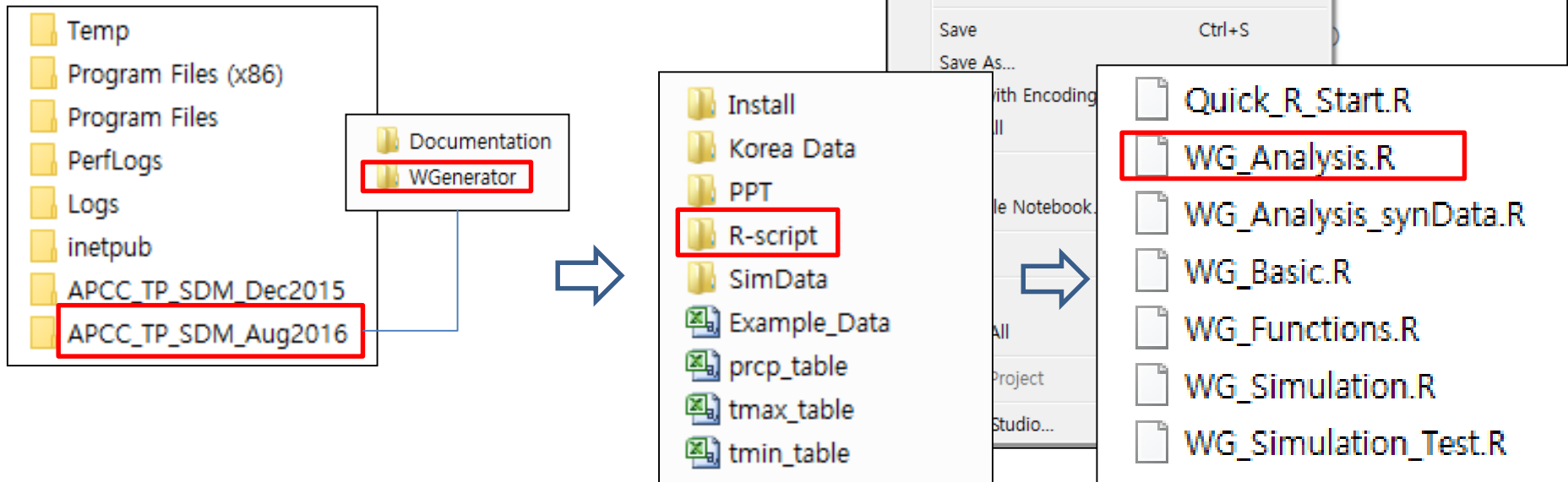
- Open **WG_Analysis.R**.

1. Click File



2. Click Open File...

**3. Find WG_Analysis.R by double click.
(Searching start C:\)**



Simulation Data Analysis

- Simulation data reading

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

```
setwd( "C:/APCC_TP_SDM_Aug2016/WGenerator" )  
num.data.set <- 100  
source("R-script/WG_Analysis_synData.R" )
```



Simulation Data Analysis

- Simulation data reading

```
Console C:/APCC_TP_SDM_Dec2015/WGenerator/
> setwd( "C:/APCC_TP_SDM_Dec2015/WGenerator" )
>
> num.data.set <- 100
>
> source("R-script/WG_Analysis_synData.R" )
>
```

- It create

The screenshot shows the RStudio Environment pane with the following variables:

Variable	Value
d	3L
i	100L
num.data.set	100
site.names	chr [1:3] "Andong" "Bongwhoa" "Tawbaekk"
syn.prcp.table	Large list (100 elements, 1.5 Mb)
syn.tmax.table	Large list (100 elements, 1.5 Mb)
syn.tmin.table	Large list (100 elements, 1.5 Mb)

Simulation Data Analysis

- Loop

```
Source on Save | Run | Source
1:10
y <- double(10)
for( i in 1:10 )
{
  y[i] <- i * i
}
y
```

- 1:10 is a vector whose elements run from 1 to 10 by 1, i.e., $1:10 == c(1,2,3,4,5,6,7,8,9,10)$
- Variable 'i' runs on 1:10 during the loop.



Simulation Data Analysis

- Loop

```
Console C:/APCC_TP_SDM_Dec2015/
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
>
> y <- double(10)      Vector 'y' is created with length 10.
>
> for( i in 1:10 )     Loop: "y[i] <- i * i" is repeated while 'i' starts 1 to 10
+ {                   For each step, 'i' is replaced
+   y[i] <- i * i     by the corresponding value of 1:10.
+ }
>
> y
[1] 1 4 9 16 25 36 49 64 81 100
>
```

Simulation Data Analysis

- List of matrices

```
Source on Save | Run | Source
# List of several matrices
M1 <- matrix( data = c(1,2,3,4,5,6) , nrow = 3 , ncol = 2 ,
              dimnames = list( NULL , c("Seoul" , "Busan") ) )
M2 <- matrix( data = c(2,3,4,5,6,7) , nrow = 3 , ncol = 2 ,
              dimnames = list( NULL , c("Seoul" , "Busan") ) )
M3 <- matrix( data = c(3,4,5,6,7,8) , nrow = 3 , ncol = 2 ,
              dimnames = list( NULL , c("Seoul" , "Busan") ) )

Mat.list <- vector( "list" , 3 )

Mat.list[[1]] <- M1
Mat.list[[2]] <- M2
Mat.list[[3]] <- M3
```

Note : double brackets [[]] are used in 'list'!!

Simulation Data Analysis

- List of matrices

```
Console C:/APCC_TP_SDM_Dec2015/WGenerator/ ↵
> Mat.list[[1]]
      Seou| Busan
[1,]      1      4
[2,]      2      5
[3,]      3      6
> Mat.list[[2]]
      Seou| Busan
[1,]      2      5
[2,]      3      6
[3,]      4      7
> Mat.list[[3]]
      Seou| Busan
[1,]      3      6
[2,]      4      7
[3,]      5      8
> |
```



Simulation Data Analysis

- List of matrices
 - Conceptually, matrix array!

Mat.list[[1]]

	Seoul	Busan
[1,]	1	4
[2,]	2	5
[3,]	3	6

Mat.list[[2]]

	Seoul	Busan
[1,]	2	5
[2,]	3	6
[3,]	4	7

Mat.list[[3]]

	Seoul	Busan
[1,]	3	6
[2,]	4	7
[3,]	5	8



Simulation Data Analysis

- List of matrices

```
Source on Save Run Source  
# Accessing the second matrix  
Mat.list[[2]]  
  
# Before seeing Console, guess the result!  
Mat.list[[2]][,"Busan"]
```



Simulation Data Analysis

- List of matrices

```
Console C:/APCC_TP_SDM_Aug2016/WGenerator/ ↵
> # Accessing the second matrix
> Mat.list[[2]]
      Seoul Busan
[1,]    2     5
[2,]    3     6
[3,]    4     7
>
>
> # Before seeing Console, guess the result!
> Mat.list[[2]][,"Busan"]
[1] 5 6 7      "Busan" column chosen!
> |
```



Simulation Data Analysis

- List of matrices

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

```
x <- double(3)
for( i in 1:3 )
{
  x[i] <- sum( Mat.list[[i]][,"Busan"] )
}
x
```

Busan column is summated!



Simulation Data Analysis

- List of matrices

```
Console C:/APCC_TP_SDM_Dec2015/ ↵
> x <- double(3)
> for( i in 1:3 )
+ {
+   x[i] <- sum( Mat.list[[i]][,"Busan"] )
+ }
>
> x
[1] 15 18 21
> |
```



Simulation Data Analysis

- Tip

```
Source on Save | Run | Source
# table() and sum of logical variables
x <- c(1,1,3,3,3,3,5,5)
table( x )
x <- c( TRUE , FALSE , TRUE , TRUE , FALSE )
sum(x)
# sum(x) == "The number of TRUEs"
```



Simulation Data Analysis

- Tip

```
Console C:/APCC_TP_SDM_Aug2016/WGenerator/ ↵
> x <- c(1,1,3,3,3,3,5,5)
>
> table( x )
x
1 3 5
2 4 2
>
> x <- c( TRUE , FALSE , TRUE , TRUE , FALSE )
> sum(x)
[1] 3
> # sum(x) == "The number of TRUES"
> |
```



Simulation Data Analysis

- **syn.prcp.table, syn.tmax.table, syn.tmin.table** are *list of matrices*.

```
Console C:/APCC_TP_SDM_Dec2015/WGenerator/
> head( syn.prcp.table[[1]] )
  MONTH DAY Andong Bongwhoa Tawbaekk
[1,]    1  1    0.0    0.0    0.0
[2,]    1  2    0.0    0.0    0.0
[3,]    1  3    0.0    0.0    0.0
[4,]    1  4   13.0    8.0   13.6
[5,]    1  5   14.8   17.8   17.3
[6,]    1  6   13.0   11.5   12.8
> head( syn.tmax.table[[1]] )
  MONTH DAY Andong Bongwhoa Tawbaekk
[1,]    1  1    2.3    2.1    0.8
[2,]    1  2    0.9    2.2    3.0
[3,]    1  3    3.4    2.5   -2.3
[4,]    1  4    2.3    1.0   -2.0
[5,]    1  5    0.3   -1.7   -2.1
[6,]    1  6   -1.9   -2.7   -4.2
> |
```

Note : the result depend on your data set.

Simulation Data Analysis

- Calculating PRCP accumulation

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

```
# Choose target site
target.site <- site.names[1]

prcp.accumulation <- double(num.data.set)

for( id in 1:num.data.set )
{
  prcp.accumulation[id] <-
  sum( syn.prcp.table[[id]][,target.site] )
}
```

For scenario syn.prcp.table[[id]], rainfalls of target site are summated!



Simulation Data Analysis

- Mean, standard deviation, quantile, and histogram

```
Source on Save Run Source
mean( prcp.accumulation )
sd( prcp.accumulation )
quantile( prcp.accumulation )

hist( prcp.accumulation ,
      xlab = "Rainfall Accumulation" ,
      main = "PRCP Accumulation in the Season" )
```



Simulation Data Analysis

- Mean, standard deviation, quantile, and histogram

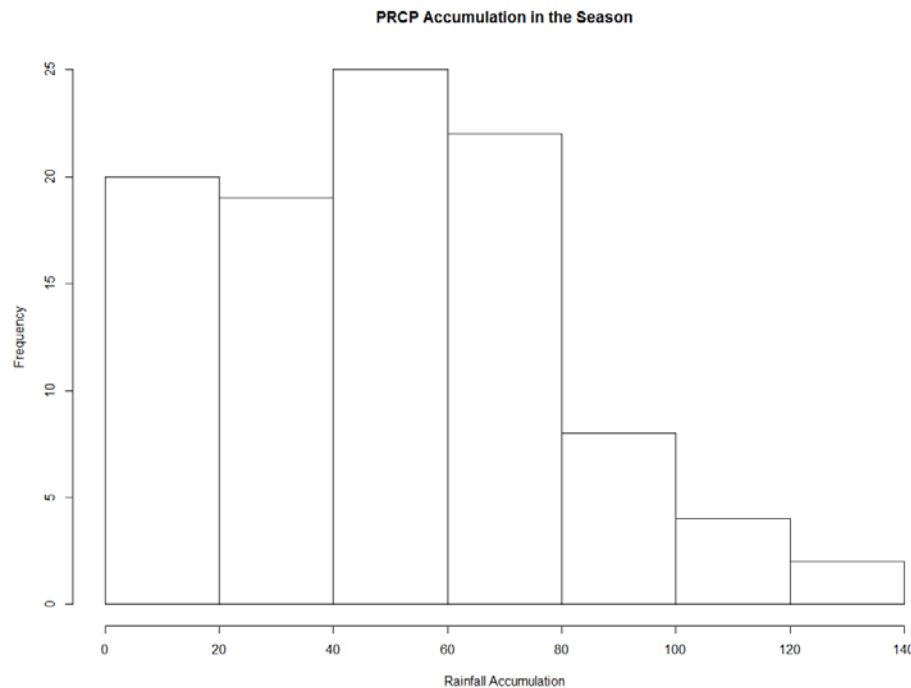
```
Console C:/APCC_TP_SDM_Aug2016/WGenerator/ ↵
> mean( prcp.accumulation )
[1] 50.242
> sd( prcp.accumulation )
[1] 29.80149
> quantile( prcp.accumulation )
      0%      25%      50%      75%     100%
 3.200 25.325 51.950 70.425 136.900
>
> hist( prcp.accumulation ,
+       xlab = "Rainfall Accumulation" ,
+       main = "PRCP Accumulation in the Season" )
> |
```

Note : the results depend on your data set.



Simulation Data Analysis

- Mean, standard deviation, quantile, and histogram



Note : the result depend on your data set.



Simulation Data Analysis

- How many days precipitation $>$ threshold?

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

```
# If the season is wet, how many days precipitation is over threshold?  
# Choose target site and threshold  
target.site <- site.names[1]  
threshold <- 10 # You can change the value!  
HowManyOverThreshold <- integer(num.data.set) threshold  
for( id in 1:num.data.set )  
{  
  HowManyOverThreshold[id] <-  
  sum( syn.prcp.table[[id]][,target.site] > threshold )  
}
```

Change value to an interesting

Count the day when rainfall amount is greater than threshold



Simulation Data Analysis

- Frequency table and its display

```
Source on Save | Run | Source
table( HowManyOverThreshold ) # frequency table!

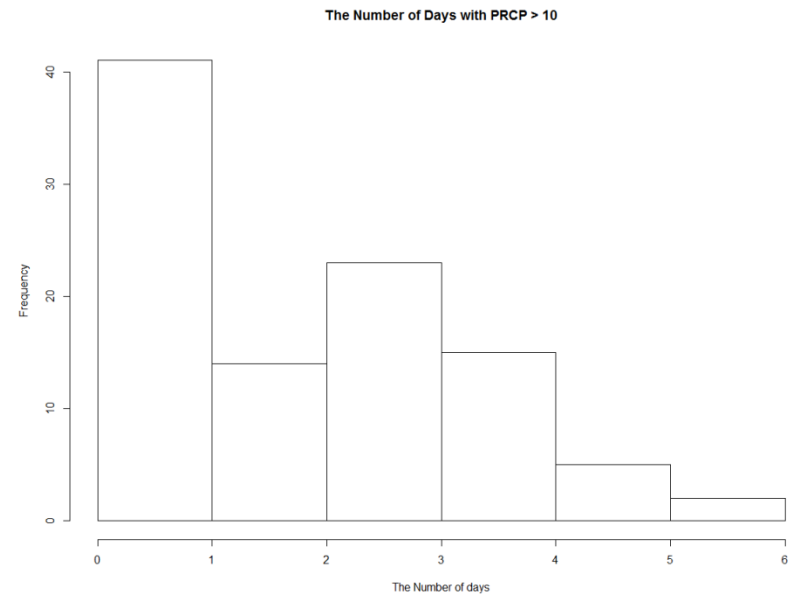
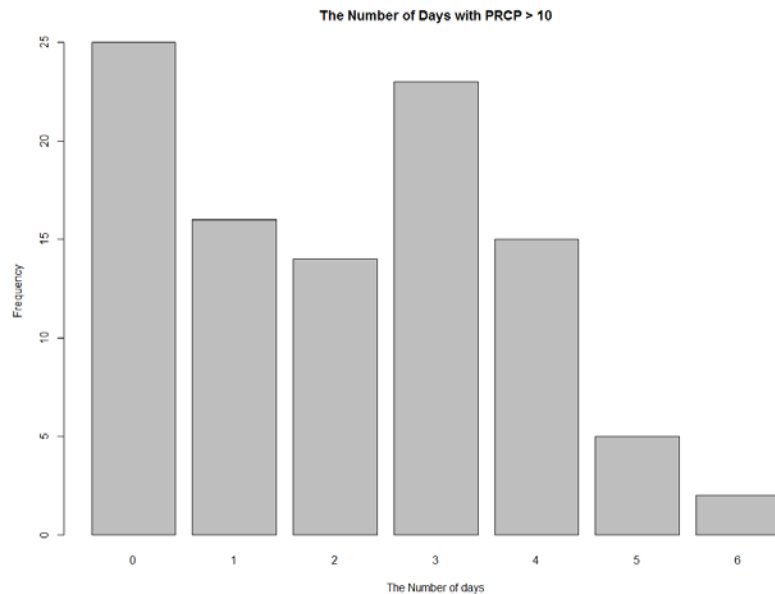
# frequency table displayed
barplot( table( HowManyOverThreshold ) ,
         main = paste( "The Number of Days with PRCP >" , threshold ) ,
         xlab = "The Number of days" , ylab = "Frequency" )

# histogram
hist( HowManyOverThreshold ,
      main = paste( "The Number of Days with PRCP >" , threshold ) ,
      xlab = "The Number of days" , ylab = "Frequency" )
```



Simulation Data Analysis

- Frequency table and its display



the result depend on your data set.



Simulation Data Analysis

- How many days are dry?

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

```
# If the season is dry, how many days are dry?  
  
# Choose target site  
target.site <- site.names[1]  
  
HowManyDry <- integer(num.data.set)  
for( id in 1:num.data.set )  
{  
  HowManyDry[id] <-  
    sum( syn.prcp.table[[id]][,target.site] == 0 )  
}
```

Count dry day



Simulation Data Analysis

- Frequency table and its display

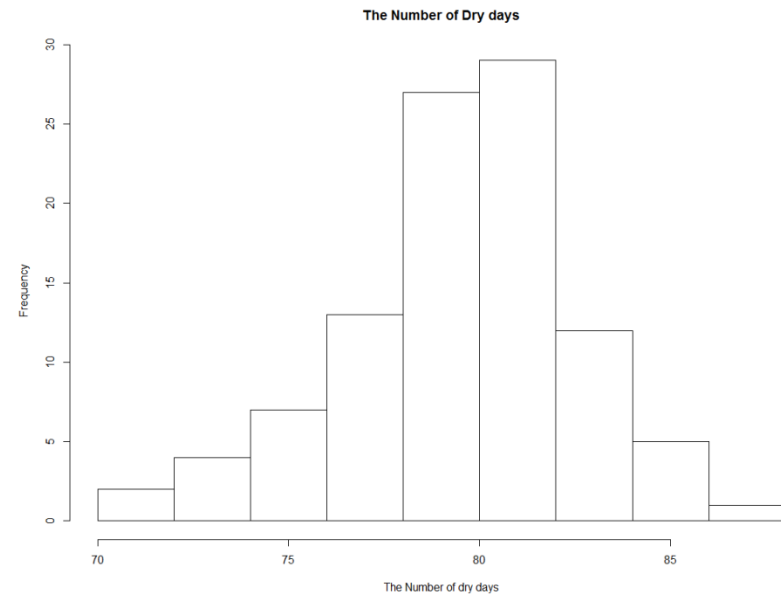
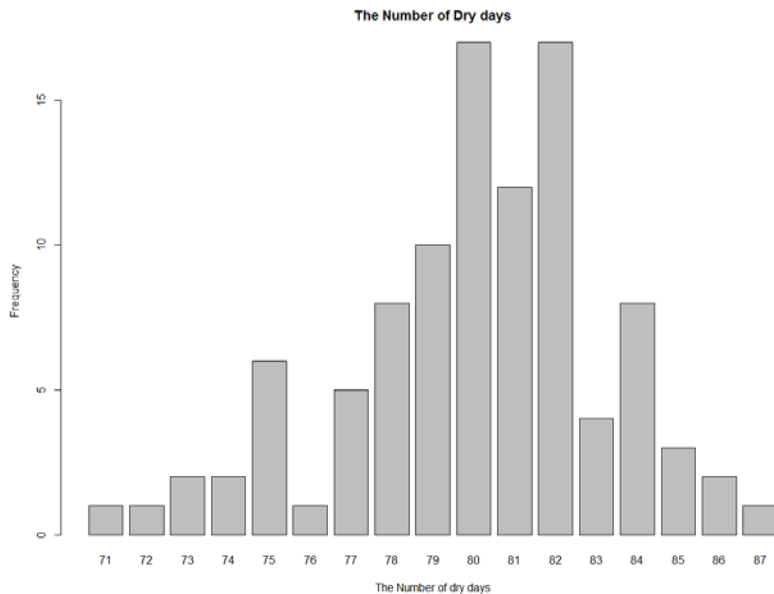
```
table( HowManyDry )
barplot( table( HowManyDry ) ,
         main = "The Number of Dry days" ,
         xlab = "The Number of dry days" , ylab = "Frequency" )

hist( HowManyDry ,
      main = "The Number of Dry days" ,
      xlab = "The Number of dry days" , ylab = "Frequency" )
```



Simulation Data Analysis

- Frequency table and its display



the result depend on your data set.



Simulation Data Analysis

- TMAX, TMIN seasonal averages

```
Source on Save | Run | Source
# TMAX, TMIN seasonal average?
# Choose target site
target.site <- site.names[1]
tmax.average <- double(num.data.set)
tmin.average <- double(num.data.set)
for( id in 1:num.data.set )
{
  tmax.average[id] <-
    mean( syn.tmax.table[[id]][,target.site] )
  tmin.average[id] <-
    mean( syn.tmin.table[[id]][,target.site] )
}
```



Simulation Data Analysis

- TMAX, TMIN seasonal averages

```
Source on Save Run Source
mean( tmax.average )
sd( tmax.average )
quantile( tmax.average )

hist( tmax.average ,
      xlab = "TMAX Average" ,
      main = "TMAX Seasonal Average in the Season" )

mean( tmin.average )
sd( tmin.average )
quantile( tmin.average )

hist( tmin.average ,
      xlab = "TMIN Average" ,
      main = "TMIN Seasonal Average in the Season" )
```



Simulation Data Analysis

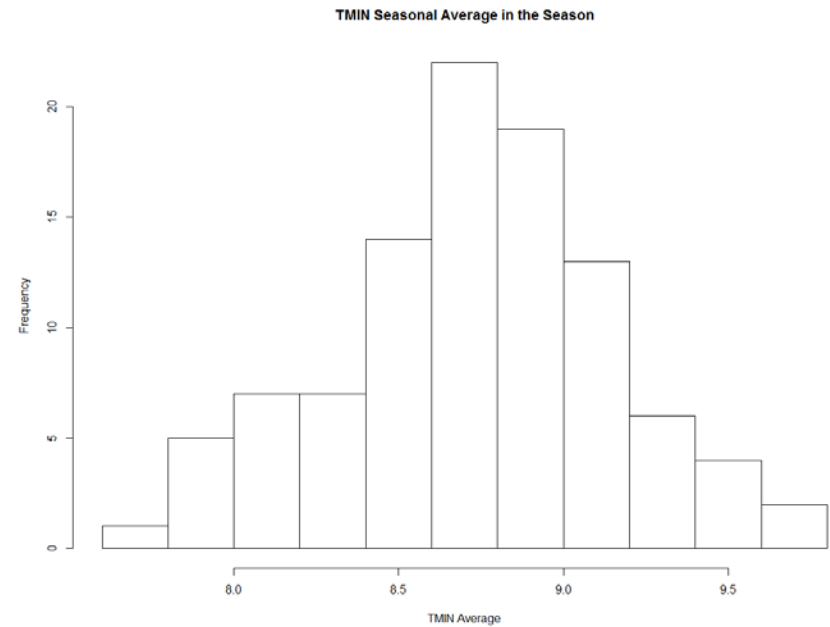
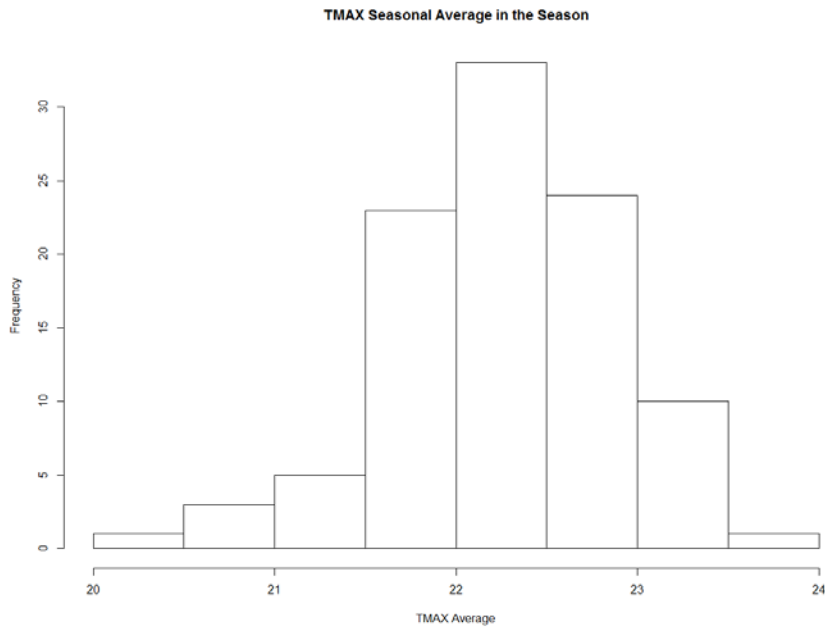
- TMAX, TMIN seasonal averages

```
Console C:/APCC_TP_SDM_Aug2016/WGenerator/ ↵
> mean( tmax.average )
[1] 22.2691
> sd( tmax.average )
[1] 0.6290205
> quantile( tmax.average )
      0%      25%      50%      75%     100%
20.19670 21.86236 22.30604 22.74286 23.52747
>
> hist( tmax.average ,
+       xlab = "TMAX Average" ,
+       main = "TMAX Seasonal Average in the Season" )
>
> mean( tmin.average )
[1] 8.731956
> sd( tmin.average )
[1] 0.425267
> quantile( tmin.average )
      0%      25%      50%      75%     100%
7.748352 8.468132 8.747802 9.004121 9.730769
>
> hist( tmin.average ,
+       xlab = "TMIN Average" ,
+       main = "TMIN Seasonal Average in the Season" )
> |
```



Simulation Data Analysis

- TMAX, TMIN seasonal averages



Simulation Data Analysis

- How many days $TMAX > \text{threshold}$

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

```
# If season is warm/hot, how many days TMAX > threshold  
# Choose target site and threshold  
target.site <- site.names[1]  
threshold <- 30 # change the value of threshold you want!  
  
HowManyOverThreshold <- integer(num.data.set) Change value to an  
interesting threshold  
for( id in 1:num.data.set )  
{  
  HowManyOverThreshold[id] <-  
    sum( syn.tmax.table[[id]][,target.site] > threshold )  
}
```



Simulation Data Analysis

- How many days $TMAX > \text{threshold}$

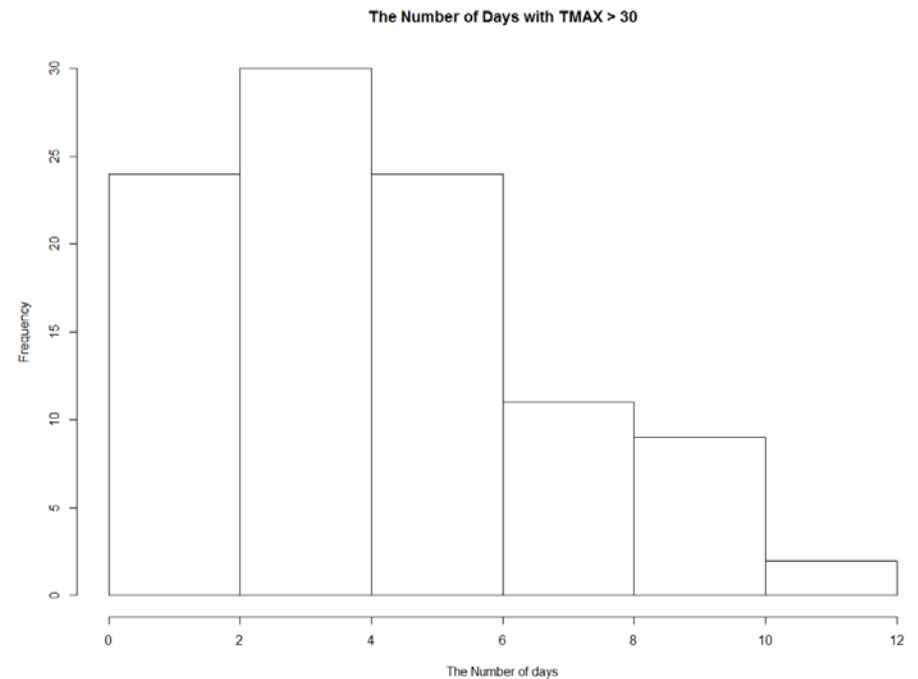
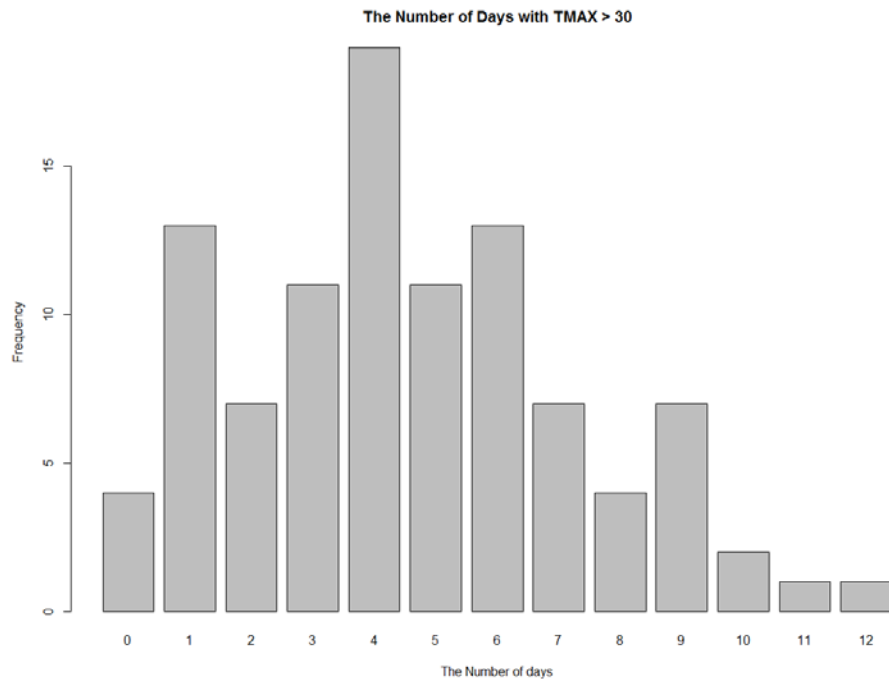
```
table( HowManyOverThreshold )
barplot( table( HowManyOverThreshold ) ,
         main = paste( "The Number of Days with TMAX >" , threshold ) ,
         xlab = "The Number of days" , ylab = "Frequency" )

hist( HowManyOverThreshold ,
      main = paste( "The Number of Days with TMAX >" , threshold ) ,
      xlab = "The Number of days" , ylab = "Frequency" )
```



Simulation Data Analysis

- How many days $TMAX > \text{threshold}$



Simulation Data Analysis

- How many days $TMIN < threshold$

```
Source on Save | Run | Source
# If season is cool/cold, how many days TMIN < threshold
# Choose target site and threshold
target.site <- site.names[1]
threshold <- 5 # change the value of threshold you want!
HowManyBelowThreshold <- integer(num.data.set) Change value to an interesting threshold
for( id in 1:num.data.set )
{
  HowManyBelowThreshold[id] <-
    sum( syn.tmin.table[[id]][,target.site] < threshold )
}
```



Simulation Data Analysis

- How many days $TMIN < \text{threshold}$

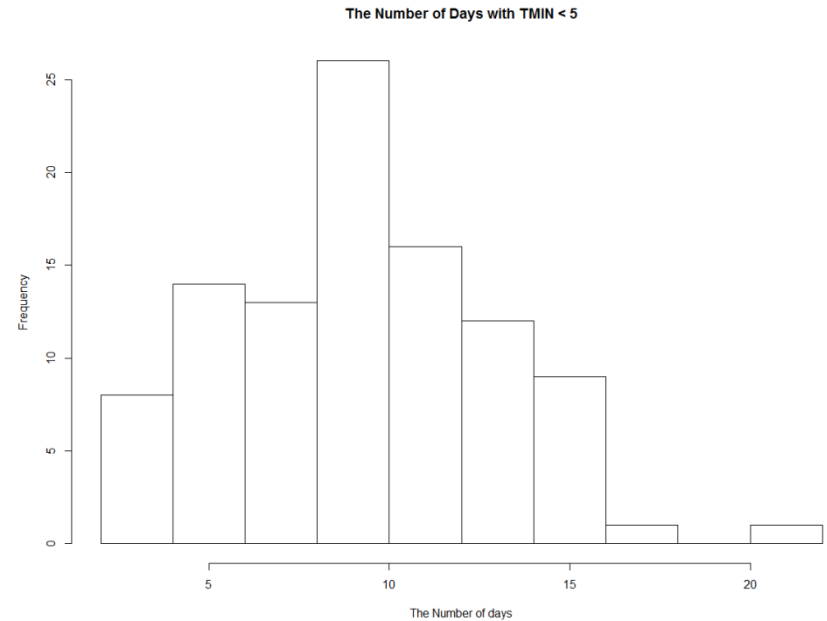
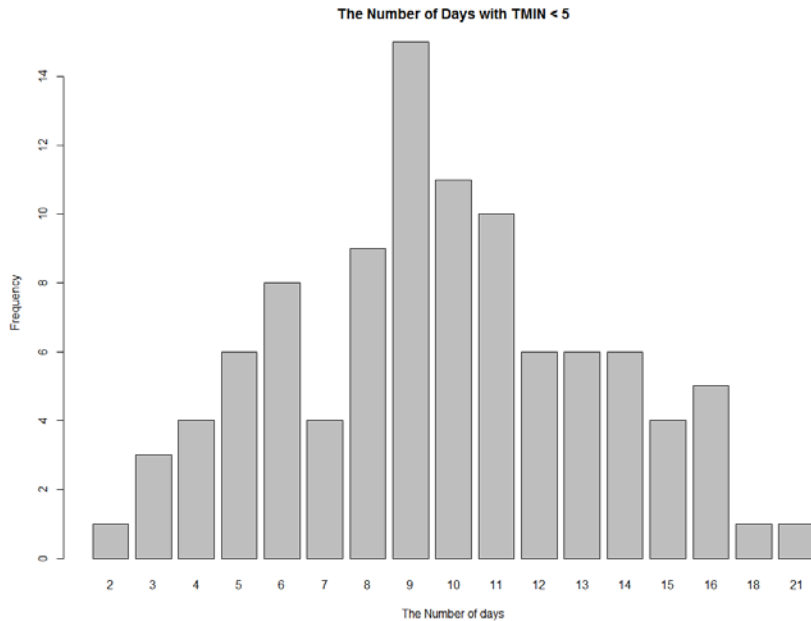
```
Source on Save | Run | Source
```

```
table( HowManyBelowThreshold )  
barplot( table( HowManyBelowThreshold ) ,  
         main = paste( "The Number of Days with TMIN <" , threshold ) ,  
         xlab = "The Number of days" , ylab = "Frequency" )  
  
hist( HowManyBelowThreshold ,  
      main = paste( "The Number of Days with TMIN <" , threshold ) ,  
      xlab = "The Number of days" , ylab = "Frequency" )
```



Simulation Data Analysis

- How many days $TMIN < \text{threshold}$



The End of Session 3

Welcome any Questions.

