Tutorial of some statistical methods using R language

Nobuhiko ENDO

Japan Agency for Marine-Earth Science and Technology

1. Introduction

Dr. Tomoshige Inoue (JAMSTEC) found that a linear relationship between ENSO and rainfall in northeastern Thailand during March-April became strong in recent two decades (Fig. 1). When sea surface temperature in central-eastern tropical Pacific is warm (cool) during preceding winter, rainfall in northeastern Thailand tends to below (above) average in March-April in recent decades. Prof. Mizoguchi (Univ. Tokyo, Leader of GRENE-ei project) was very interesting in his results. Then he want to prepare a research tool for analyzing linear relation between two variables which can be easily used by undergraduate students. I will introduce **R** language for GRENE-ei colleagues.

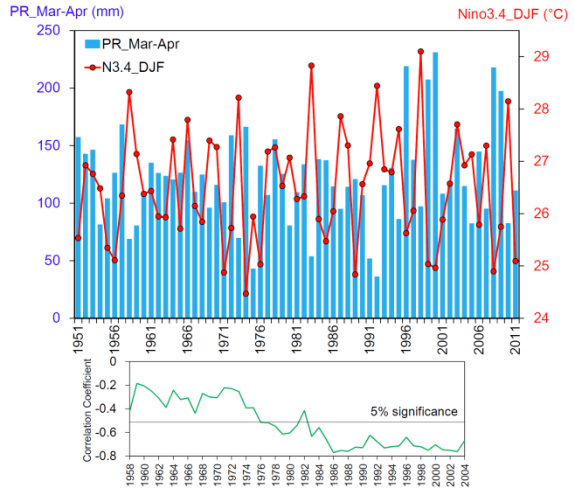

Fig.1. Rainfall total in northeast Thailand during March-April and Nino3.4 index during the previous winter (December – February) in upper panel. Unit of rainfall and that of Nino3.4 index are mm and °C, respectively. Time series of 15-years sliding correlation is shown in lower panel. Courtesy of Dr. Inoue (JAMSTEC).

2. R language

"R" is a free software for statistical study, and has been developed extensively in recent years. R project web page is found at https://www.r-project.org/. Introductory free text "simpleR - Using R for Introductory Statistics John Verzani" also can be obtain from https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf.

R base package can obtain from CRAN mirror site at the Institute of Statistical Mathematics, Tokyo, Japan (http://cran.ism.ac.jp/). Most recent version of R is Version 3.2.2, and there is binary package for Linux, Mac OS X, and Microsoft Windows. Please download R base package, and install your personal computer. After installation, you can find an Icon "R" at your Desktop in PC. Please click and start "R", then you will find "startup window of R" (see, Fig. 2).

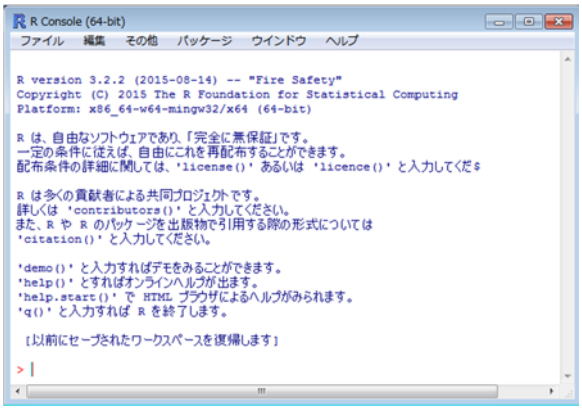

Fig.2. Startup window of "R".

Next, choose [Edit] in menu bar, then [GUI preference]. You will find a window shown in Fig. 3, then select "SDI" in preference menu. Save preferences, then close the window.

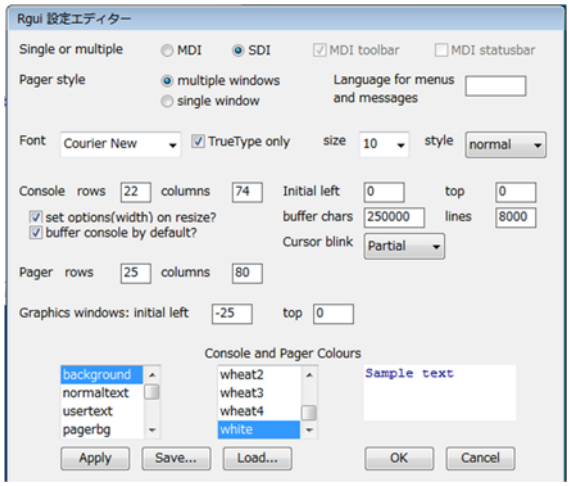

Fig. 3. Select "SDI".

We assume that you put R scripts and data files in "C:¥Work¥R" and "C:¥Work¥R¥Data" in your hard drive. In R console, type following command and return.

```
> setwd( "c:/Work/R" )
> getwd()
```

Working directory (folder) will change to "c:/Work/R". It is note that we do not use "¥" in R, and instead use "/". To quit "R", type in "quit()" and push "return key".

```
> quit()
```

3. Global surface air temperature anomaly

Japan Meteorological Agency (JMA) compiles climatological anomaly time series of surface air temperature and precipitation based on their own observations and observed data distributed in World Meteorological Organization network.

We will use the JMA's climatological anomaly time series. The csv file (JMA_Anomaly.csv)

includes 1) global land surface air temperature anomaly, 2) northern hemisphere and southern hemisphere land average temperature anomalies, 3) surface air temperature anomaly of Japan which was compiled from 15 JMA stations, and 4) precipitation anomaly of Japan which was compiled from 51 JMA stations. For calculating temperature anomaly in Japan, JMA selected 15 stations which are rural station and to exclude urban heat island effects.

Start "R", then key in following commands. Note that line with "#" treated as "comment line" in R, and do not need key in "comment line".

```
# read JMA climatological anomaly data
> x <- read.csv("./Data/JMA_Anomaly.csv", header = TRUE)
# In R, "<-" mean "substitution".
# show first 10 rows in x.
> head(x)
  YEAR GL_TEMP NH_TEMP SH_TEMP JP_TEMP JP_RAIN
1 1898   -0.66   -0.65   -0.68   -0.75    15.1
2 1899   -0.56   -0.58   -0.55   -0.81   199.2
3 1900   -0.49   -0.48   -0.51   -1.06   -43.3
4 1901   -0.58   -0.55   -0.63   -1.03    48.6
5 1902   -0.70   -0.75   -0.66   -1.03   154.7
6 1903   -0.77   -0.78   -0.77   -0.77   266.2
>
```

There is 5 column. Column "YEAR" is AD Year. "GL_TEMP", "NH_TEMP", "SH_TEMP" are globally averaged, northern hemisphere averaged, and southern hemisphere averaged surface air temperature anomaly, respectively. "JP_TEMP", "JP_RAIN" are surface air temperature anomaly and precipitation anomaly averaged over Japan.
Let's plot global temperature anomaly.

```
> plot(x$YEAR, x$GL_TEMP, type="l")
```

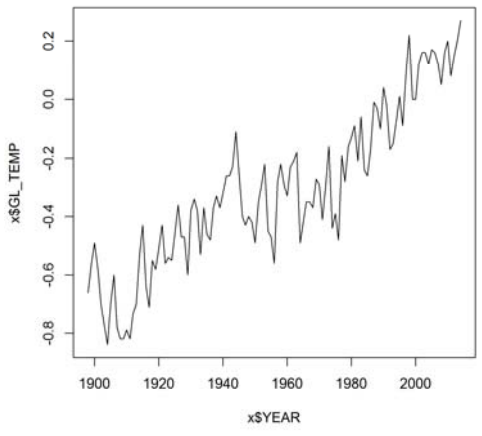

Fig. 4. Time series of global surface temperature anomaly.

You will get figure 4. It is obvious that the global surface temperature anomaly has increasing trend (long-term linearly upward tendency) over the period.

We will estimate the trend used by the least square fit in R.

```
# Y = a0 + a1 * X
#  x$YEAR as X. x$GL_TEMP as Y.
#  lm is a function for least square fit
> res <- lm(x$GL_TEMP ~ x$YEAR)
# show summary of results.
> summary(res)

Call:
lm(formula = x$GL_TEMP ~ x$YEAR)

Residuals:
     Min       1Q   Median       3Q      Max
-0.31745 -0.08373  0.00179  0.08435  0.28574

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.456e+01  6.292e-01  -23.14   <2e-16 ***
x$YEAR       7.287e-03  3.216e-04   22.66   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1175 on 115 degrees of freedom
Multiple R-squared:  0.817,     Adjusted R-squared:  0.8154
F-statistic: 513.3 on 1 and 115 DF,  p-value: < 2.2e-16

> slope <- res$coefficients[2]
> intercept <- res$coefficients[1]
> plot(x$YEAR, x$GL_TEMP, type="l")
> abline(res, col="red")
```

Figure 5 shows the global surface air temperature anomaly and the trend in same graph. The Trend in the global surface temperature anomaly was 7.287e-03 °C/year (0.73 °C/100year).

The residual temperature anomaly is defined as

Residual Anomaly = Raw Temperature Anomaly – Long-term Trend

The residual anomaly (also called as detrended time series) includes several time scale of variability. We will remove small temporal scale variability (less than 10 years) in the residual anomaly.

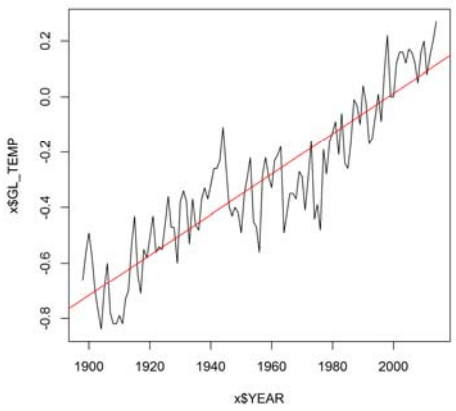

Fig. 5. The global surface air temperature anomaly (black line) and trend (red line).

```
# pick up “YEAR” column from data frame “x”
> year1 <- x$YEAR
# include R script for “running average”.
> source("./MyRunAve.R")
# set window size of running average. 11-year window.
> window.size <- 11
# get running averaged temperature anomaly.
> t.runave <- MyRunAve( res$residual, window.size )
# t.runave is vector at this time.
# we will convert from vector to data frame
> t.runave <- as.data.frame( cbind(year1, t.runave ) )
# show dataframe. First 10 rows. NA means “missing value”.
> head(t.runave, n = 10)
# plot the residual anomaly. Do not draw axes, labels.
# positive (negative) anomaly is drawn in red (blue)
# set limit range for y-axis from -0.35 to 0.35
# type = “h” means vertical bar plot
> plot( x$YEAR, res$residual, type = "h", ylim = c(-0.35,0.35), axes = FALSE, xlab = "",
ylab = "", col = ifelse(res$residual > 0.0, "red", "blue"))
# to overlay second plot, use “par()” command.
> par( new = TRUE )
# plot running average time series. “lwd” is thickness of line.
> plot( t.runave$year1, t.runave$t.runave, type = "l", ylim = c(-0.35,0.35), lwd = 4,
col = "orange", xlab = "Year", ylab = "Temperature Anomaly")
```

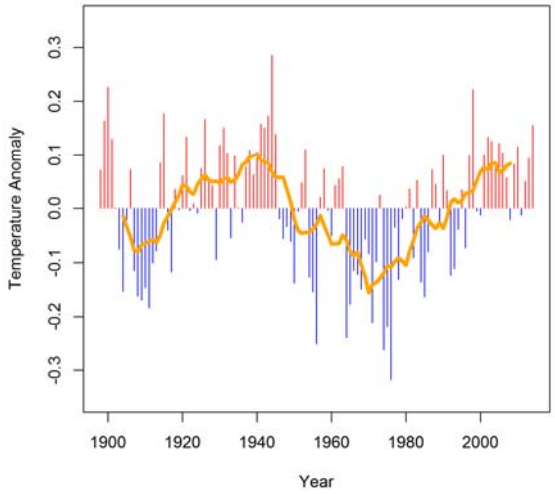

Fig. 6. The residual temperature anomaly time series (bar plot) and The 11-year running averaged temperature (orange line). Red (blue) mean positive (negative) anomaly.

We get the residual time series and the 11-year running averaged temperature anomaly (Fig. 6). It is obvious that there are relatively cold period in the early 20th century and from mid-1940's to mid-1990's and relatively warm period from 1920's to mid-1940's and recent decades. To quit "R", type "quit()".

4. Some topics in R

In the previous section, we read the land surface temperature anomaly from "CSV" file.

```
> x <- read.csv("./Data/JMA_Anomaly.csv", header = TRUE)
```

In R, we can read "space (tab) separated" and "comma separated" ASCII file. "read.table()" and "read.csv()" functions for use reading "space (tab) separated" and "comma separated" text file, respectively.

```
> x <- read.table("JMA_Tokyo_2.txt", header = TRUE)
# show first 6 rows
> head(x)
  YEAR MON TAVE TMAX TMIN  RAIN
1 1951   1  3.3  8.6 -1.0  36.4
2 1951   2  4.5  9.7  0.4 120.4
3 1951   3  8.8 14.2  4.1 149.1
4 1951   4 13.3 18.3  9.0 177.1
5 1951   5 18.0 23.3 13.9  86.4
6 1951   6 21.2 25.9 17.3 144.3
```

A data frame “x” includes six columns, which are “YEAR”, “MON”, “TAVE”, “TMAX”, “TMIN” and “RAIN”. Those are monthly observed value at JMA Tokyo HQ. In R, we will use “data frame” structure. If you read the “space separated file” / “CSV file”, data are automatically transformed to “data frame”.
To create a vector, we will use “c”.

```
> x <- c(1,2,3,4,5)
# to show first part of vector x
> head(x)
[1] 1 2 3 4 5
# to get length of vector x
> length(x)
[1] 5
```

“length()” is function for obtaining length of vector “x”. To create regular pattern vector, “seq()”, “rep()” functions.

```
> c(1:5)
[1] 1 2 3 4 5
> c(3:-3)
[1]  3  2  1  0 -1 -2 -3
> rep(1:3,length=5)
[1] 1 2 3 1 2
> seq(1,10,length=5)
[1]  1.00  3.25  5.50  7.75 10.00
```

A data frame can include “numeric vector”, “character vector”, and “factor vector”.

```
    Type   Length  Weight
1     A      112       30
2     B      153       55
3     A      123       42
```

“Type” is “factor”, and “Length” / ”Weight” is “numeric”. Next example show how to create a data frame.

```
# numeric vector
```

```
> nvec <- c(1951,1961,1971,1981,1991)
# vector contain factor
> fvec <- c("A","A","B","C","B")
# create a data frame. "cbind()" works for binding columns.
> new.dt <- cbind(nvec,fvec)
# attach column names
> colnames(new.dt) <- c("YEAR", "TYPE")
> head(new.dt)
     YEAR   TYPE
[1,] "1951" "A"
[2,] "1961" "A"
[3,] "1971" "B"
[4,] "1981" "C"
[5,] "1991" "B"
# pick up an element.
> new.dt[1,2]
TYPE
 "A"
> new.dt[2,1]
  YEAR
"1961"
# select a column
> new.dt[,1]
[1] "1951" "1961" "1971" "1981" "1991"
# select a row
> new.dt[2,]
  YEAR   TYPE
"1961"    "A"
```

If you want delete any data frame / vector / object, “rm()” function is available.

```
# delete single object x
> rm(x)
# delete all objects
> rm(list=ls())
```

Exercise.

A. Plot long-term precipitation anomaly of Japan. Anomaly data is stored in

"JMA_Anomaly.csv". To read data, you should use "read.csv()" with option "header = TRUE".

B. Plot a time series of rice production in Thailand. The rice production data are stored in "Rice_Thailand.csv". The rice production data were obtained from FAO database and "World Rice Statistics" (Palacpac, 1977).

5. How to save graph plot and data frame

In this section, we will introduce method for saving a graph plot. We can save a graph as "PNG", and "PDF".

```
> x <- c(1951,1961,1971,1981,1991)
> y <- c(1:5)
# plot symbol at (x,y)
> plot(x, y)
# plot line at (x, y)
> plot(x, y, type="l")
# to save a graph to PNG file. First set resolution in ppi unit.
> ppi <- 300
# width and height are in inch.
> png("fig_1.png", width = 6 * ppi, height = 3 * ppi, res = ppi)
> plot(nvec,x,type="l")
> dev.off()
>
# to save a PDF file.
> pdf("fig_1.pdf")
> plot(nvec,x,type="l")
> dev.off()
# save data frame
> new.dt <- cbind(x, y)
> colnames(new.dt) <- c("YEAR","TYPE")
> head(new.dt)
> write.csv(new.dt,"test1.csv",row.names=FALSE)
# You can read "test1.csv" in Microsoft Excel.
```

Exercise.

A. Plot a climatological precipitation anomaly time series of Japan, and save it to a PNG file. "JMA_Anomaly.csv" has the precipitation anomaly data as "JP_RAIN".

6. What is running mean (average)

In section 3, we applied an 11-year running mean (average) to the residual temperature anomaly series. What is the running mean? Running mean is a simple weighted average in a fixed time window (Box weight). Let y_i is observed values, and i is time index.

$$\overline{y_i} = \frac{1}{2n+1}\sum_{k=-n}^{n} y_i = w_k \sum_{k=-n}^{n} y_i$$

For the 11-year running mean, window length and n are related as $11 = 2n + 1$. Therefore, $n = 5$. An average was obtained from index i=-5 to index i=5.

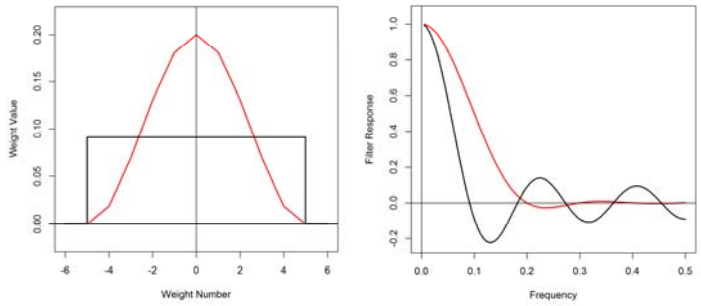

Fig. 7. Weight distribution (left), and its response function (right). Box (cosine) weight is shown in black (red) line.

Weights are $w_k = 1/(2n+1)$, and equal over the window. In figure 7, two example of weight for the 11-year running mean are shown. One is equal weight (box weight), and another is cosine weight.

$$w_k = \frac{1 + cos\,(\pi k/n)}{2n}$$

As shown in response function (right panel of Fig. 7), the observed component with frequency larger than 0.1 (the observed signal less than 10 years periodicity) are almost eliminated from the original observed time series after applied the equally weighted average. However, the simple equally weighted mean has drawback. As shown in Fig. 7, the response function indicate wave-like form around 0. This means that the weighted averaged time series is contaminated. On the other hand, the cosine weighted average shows no wave-like form in response function. So, the cosine weighted mean is better than the simple equally weighted mean. In despite of the drawback in the simple equally weight mean, the simple equally weighted mean has been used in meteorological / climatological study.

5. Correlation between observed data

Correlation coefficient is a measure of linear association between two variables. Correlation coefficient easily calculated in R. In this section, we will use a data file "JMA_Consume.csv" and calculate correlation coefficient between two variables. There is following variables:

"Mon", "Day", "Temp", "Watermelon", "Icecream", and "ChineseNoodle" in "JMA_Consume.csv". "Temp" is average surface air temperature prepared from 15 JMA stations. Watermelon, Ice Cream, and Chinese Noodle are consumer spending in Japan from July to August, 2015. The consumer spending data were obtained from the statistical office of Japanese Government.

```
# remove all objects before used.
> rm(list=ls())
# read data
> x <- read.csv("./Data/JMA_Consume.csv",header=TRUE)
# check the data
# Temp = air temp (degC). Watermelon, Icecream, Chinese Noodle are in unit of JPY.
> head(x)
  Mon Day     Temp Watermelon Icecream ChineseNoodle
1   7   1 23.96667       6.90    25.88          9.73
2   7   2 26.61333      10.16    36.90          8.13
3   7   3 25.44667       7.33    33.15         12.61
4   7   4 23.40667      13.34    39.25         25.93
5   7   5 23.74000      12.16    41.02         33.77
6   7   6 23.56667       6.74    18.88          7.80
# pch=16 means plot filled circle
> plot(x$Temp,x$Watermelon, col="red", pch=16)
```

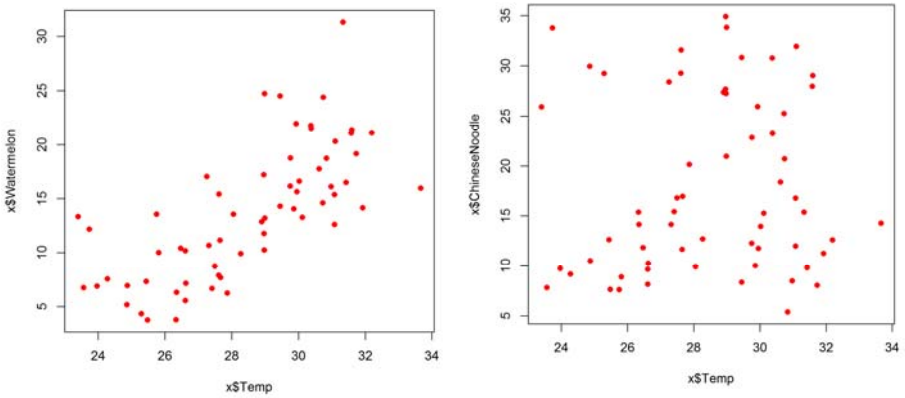

Fig.8. Scatter plot between air temperature and consumption of watermelon (left) and Chinese Noodle (right) in Japan.

We get a scatter plot between air temperature and consumption of watermelon in Japan (Fig. 8, left). It is clearly showed that most of people tend to buy watermelon when air temperature was high. Similar result is also true for Ice Cream. On the contrary, there is no linear relation between air temperature and consumption of Chinese Noodle (Fig. 8, right).

Let's evaluate how linearly associated between watermelon and air temperature using with "cor.test()" function.

```
> cor.test(x$Temp,x$Watermelon)

        Pearson's product-moment correlation

data:  x$Temp and x$Watermelon
t = 7.7753, df = 60, p-value = 1.185e-10
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5572856 0.8141508
sample estimates:
      cor
0.7084439
```

The "cor.test()" calculate "correlation coefficient", and statistical significance also evaluate. Statistical test for correlation coefficient evaluate "correlation coefficient is 0 (zero)" or not. The result of "cor.test()" showed that correlation coefficient between air temperature and consumption of watermelon is 0.708. Further, p-value is 1.185e-10. If we set a significance level of 0.05, the p-value is smaller than 0.05. So, the correlation coefficient is statistically different from 0.

Let's move another example. We will revisit well known relationship between sea surface temperature in the tropical Pacific and sea level pressure index produced from observations at Darwin, Australia and Tahiti Island (Southern Oscillation Index; SOI). Nino 3.4 index is a measure of activity of El Nino/La Nina phenomenon, which is regional average of sea surface temperature 5°S-5°N, 170°W-120°W. When Nino 3.4 is higher (lower) than 0.5 (-0.5) °C and continue 5 months, we consider El Nino (La Nina) event has occurred. SOI is difference between Tahiti and Darwin (Tahiti minus Darwin). When La Nina (El Nino) event occur, SOI is positive (negative).

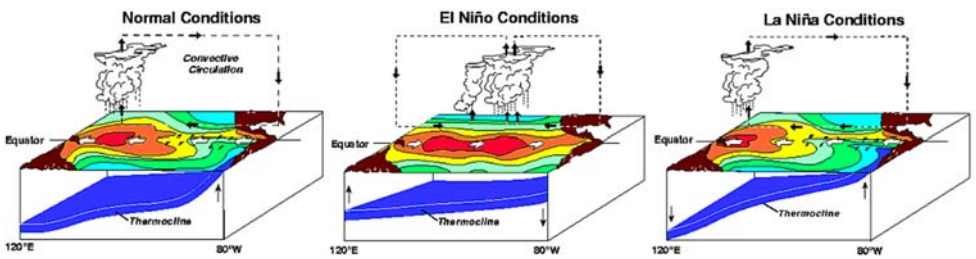

Fig. 9. Schematic figure for El Nino and La Nina condition (NOAA/CPC).

At first, a 5-month running mean applied to “nino 3.4” and SOI time series. Then the boreal winter season (Dec – Feb) mean were prepared. Data files were “./Data/nino_DJF.csv”, “./Data/soi_DJF.csv”, respectively.

```
> x <- read.csv("./Data/nino_DJF.csv", header = TRUE)
> head(x)
# x is a temporal data frame. Nino3.4 index is available from 1951.
# On the contrary, SOI are available from 1952. So, delete Nino3.4 data of 1951.
# delete first row
> nino <- x[-1,]
> head(nino)
  YEAR NINO34  ANOM
2 1952  26.92  0.30
3 1953  26.86  0.23
4 1954  27.08  0.46
5 1955  25.60 -1.02
6 1956  25.29 -1.33
7 1957  26.13 -0.50
> rm(x)
# read SOI index : Dec-Feb mean.
> soi <- read.csv("./Data/soi_DJF.csv", header = TRUE)
> head(soi)
  YEAR         SOI
1 1952 -0.56000000
2 1953 -0.26666667
3 1954  0.02666667
4 1955  0.66666667
5 1956  1.27333333
6 1957  0.50666667
> plot(nino$NINO, soi$SOI, pch=16, col="red")
> cor.test(nino$NINO, soi$SOI)

        Pearson's product-moment correlation

data:  nino$NINO and soi$SOI
t = -16.157, df = 57, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
```

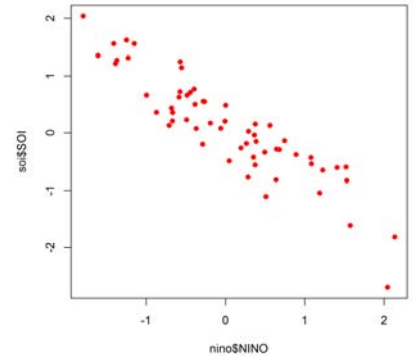

Fig. 10 Scatter plot between Nino3.4 index and SOI index

```
 -0.9432223 -0.8462118
sample estimates:
       cor
-0.9059704
```

It is obvious that there is very strong linear relationship between Nino 3.4 index and SOI index for the period 1952-2010 (Fig. 10). When Nino 3.4 index is strongly low, SOI index is simultaneously high (La Nina condition). On the contrary, Nino 3.4 is high with low SOI (El Nino). Linear relationship between Nino 3.4 and SOI is -0.906 with very low p-value (2.2e-16).

Exercise.

A. Evaluate linear association between rainfall at Darwin and SOI. Monthly rainfall at Darwin are in "Rain_Darwin_2.csv". SOI are in "soi_std_3.csv".

B. Evaluate linear association between monthly average surface air temperature at Tokyo and Nino 3.4 index. Meteorological data at Tokyo are in "JMA_Tokyo_2.txt", and Nino 3.4 are in "ersst3b.nino34_2.csv".

7. Sliding correlation

Sliding correlation (moving window correlation) is used for investigation of temporal variation in linear relationship between two variables. In climatologically, ENSO and other largescale phenomena shows a decadal scale variability. Therefore a linear relationship between ENSO and rainfall at some region may change from an epoch to another epoch. In this section, we will calculate sliding correlation.

In this section we will use an R package "dplyr". To install required R package, setup CRAN mirror site (Fig. 11). Select [Package] in menu bar, window "HTTPS CRAN mirror" will pop up. Select "HTTP mirrors" and push "OK". New window "HTTP CRAN mirror" will open. Select "JAPAN (Tokyo)". Back to R console, please type following command, then push return key.

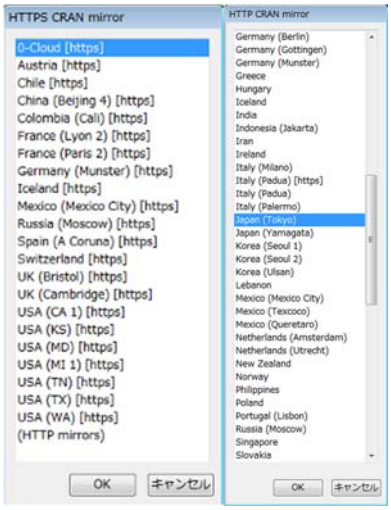

Fig.11 Choose "(HTTP mirrors)", then choose again "Japan (Tokyo)".

```
> install.packages( "dplyr" )
```

"dplyr" is a very useful package for

aggregation / searching large data.
When using “dplyr”, some tricky code will introduce. Other programing language use “y = x”, this means “x” substitute to “y”. However, “dplyr” use another syntax. “x -> y”, this means “x” substitute to “y”. In addition, “%>%” can be use in script. These syntax called as “chain syntax”.

We will use monthly rainfall data at Kohn Kaen, northeast Thailand.

```
> library(dplyr)
> x <- read.table("./Data/Rain_TMD_KhonKaen_2.txt", header = TRUE)
> head(x)
  YEAR MON  RAIN
1 1951   1   9.8
2 1951   2  23.5
3 1951   3  22.9
4 1951   4  80.7
5 1951   5 181.7
6 1951   6 208.1
# pick up rainfall data in Marth and April. Picked data substitute to "y".
# chain syntax %>% will be used.
# first select March/April data, then data after 1952 are picked up.
> filter(x, MON>=3&MON<=4) %>% filter(YEAR>=1952) -> y
# group_by(YEAR) :  It means for "each YEAR"
# summarize(RR_MA = sum(RAIN)) : It will gets rainfall in March + April
# Total rainfall during March/April is stored in data frame  'rr'
> y %>% group_by(YEAR) %>% summarize(RR_MA = sum(RAIN)) -> rr
> head(rr)
Source: local data frame [6 x 2]
   YEAR RR_MA
  (int) (dbl)
1  1952 187.3
2  1953 123.6
3  1954  40.6
4  1955  84.0
5  1956 128.0
6  1957 164.0
# plot time series
> plot(rr$YEAR, rr$RR_MA, type="h")
```

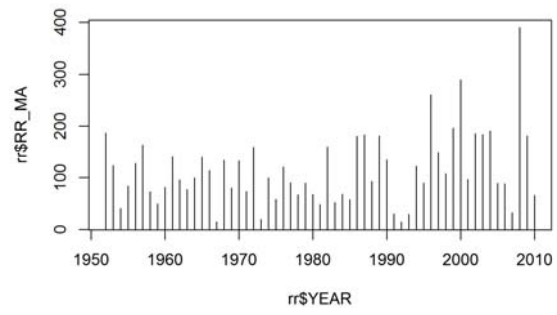

Fig. 12 Time series of RR_MA

```
# write data frame ‘rr’ to CSV file.
> write.csv(rr, "./Rain_KK_MA.csv", row.names = FALSE)
# delete all object
> rm(list=ls())
>
# read Nino3.4 DJF data
> x <- read.csv("./Data/nino_DJF.csv", header = TRUE)
> nino <- x[-1,]
> head(nino)
  YEAR       NINO
2 1952  0.3760000
3 1953  0.1966667
4 1954  0.2913333
5 1955 -0.9933333
6 1956 -1.3673333
7 1957 -0.3813333
> rr <- read.csv("./Rain_KK_MA.csv")
> head(rr)
  YEAR RR_MA
1 1952 187.3
2 1953 123.6
3 1954  40.6
4 1955  84.0
5 1956 128.0
6 1957 164.0
# check length of both data
> length(nino$NINO)
[1] 59
> length(rr$RR_MA)
[1] 59
# include a script which get statistical significant correlation coefficient at 5% point
> source("./MyCorLimit.R")
# include a sliding correlation script
> source("./MySlidCor.R")
# set window size : 13-year window
> window <- 13
# calculation of sliding correlation
> res <- MySlidCor(nino$NINO, rr$RR_MA, window)
```

```
# check first 10 rows in 'res'.  'res'  is matrix
> head(res,n=10)
          corVal      pVal
 [1,]         NA        NA
 [2,]         NA        NA
 [3,]         NA        NA
 [4,]         NA        NA
 [5,]         NA        NA
 [6,]         NA        NA
 [7,] -0.1859529 0.5430217
 [8,] -0.4121778 0.1616505
 [9,] -0.3366898 0.2606415
[10,] -0.1881112 0.5382650
# set significance level at 0.05.
> alpha <- 0.05
# calculate limit of correlation coefficient
> cor.limit <- MyCorLimit(window, alpha)
> cor.limit
[1]  0.5529427 -0.5529427
# If correlation coefficient is larger (smaller) than 0.553 (-0.553),
# correlation coefficient is different from  '0' (statistically significant)
#
# get year
> years <- rr$YEAR
# conbine year and res, then create new data frame  'res2'
> res2 <- as.data.frame(cbind(years,res))
# check  'res2', first 10 rows. NA means  "No data"
# corVal is correlation coefficient
# pVal is p-value
> head(res2,n=10)
   years     corVal      pVal
1   1952         NA        NA
2   1953         NA        NA
3   1954         NA        NA
4   1955         NA        NA
5   1956         NA        NA
6   1957         NA        NA
7   1958 -0.1859529 0.5430217
```

```
8    1959 -0.4121778 0.1616505
9    1960 -0.3366898 0.2606415
10   1961 -0.1881112 0.5382650
# plot time series of correlation.
> plot(res2$years,res2$corVal, type="l", ylim = c(-0.8,0.2), xlab = "YEAR", ylab =
"Correlation Coef.", col = "red", lwd = 2)
# add horizontal zero-line
> abline(h=0)
# add horizontal line of upper limit of correlation coef.
> abline(h=cor.limit[1], col = "blue")
# add horizontal line of lower limit of correlation coef.
> abline(h=cor.limit[2], col = "blue")
# plot save to PNG file
# set resolution in ppi. Width and length are Inch.
> ppi <- 300
> png("fig_KohnKane_cor.png", width = 6 * ppi, height = 3 * ppi, res = ppi)
> plot(res2$years, res2$corVal, type = "l", ylim = c(-0.8,0.2), xlab = "YEAR", ylab =
"Correlation Coef.", col = "red", lwd = 2)
> abline(h=0)
> abline(h=cor.limit[2], col = "blue")
# output to PNG is finished by dev.off(). Do not forget this function.
> dev.off()
# write out the results to CSV file
> write.csv(res2, "./res_KohnKane_cor.csv", row.names = FALSE)
# QUIT R
> quit()
```

In this example, we calculated the correlation coefficient between March-April total rainfall at Kohn Kaen, Thailand and previous DJF mean Nino 3.4 Index with 13-years sliding window. Figure 13 shows the result. A horizontal blue line is limit of correlation coefficient with degree of freedom = 11. Correlation coefficient is statistically different from '0' at 0.05 % level.

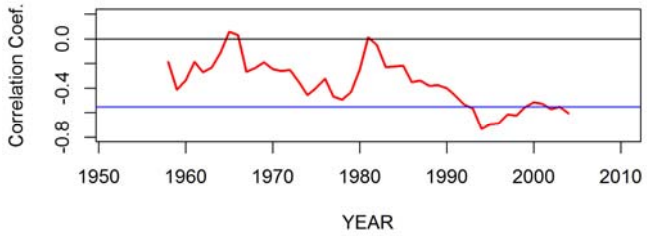

Fig. 13. 13-years window sliding correlation between March-April rainfall at Kohn Kaen, Thailand and previous winter (Dec-Feb) mean Nino 3.4 index.

It is also note that window size can be arbitrarily chosen, but should be odd number.

El Nino / La Nina phenomenon occurred 5 to 9 times for each decade (Table 1). So, we can choose 13-year window. If the phenomena only occur every 15 year, we cannot choose 13-year window size.

You need to consider a characteristics of a phenomena's variability.

Table 1. Number of El Nino / La Nina years defined by NOAA/CPC.

Decades	El Nino	La Nina
1950's	5	3
1960's	3	2
1970's	4	5
1980's	3	2
1990's	3	3
2000's	4	2

Appendix A. List of R scripts.

Name	Description
MyCorLimit.R	Calculate upper/lower limit of correlation coefficient. x <- MyCorLimit(n, alpha) <In> n : length of vector (number of pairs), alpha : significance level, usually uses 0.05 or 0.01. <Return> vector of upper/lower limit of correlation coefficient.
MyRunAve.R	Calculate running average y <- MyRunAve(x, window) <In> x : a vector of observations, window : size of window (e.g. if window=11, 11-years running average.) <Return> y : vector of running averaged observations
MySlidCor.R	Calculate sliding correlation. z <- MySlidCor(x, y, window) <In> x : a vector of observations. y : another vector of observations. window : size of sliding window (e.g. if window=13, 13-years sliding correlation coefficient. <Return> z : matrix of corVal and pVal. corVal are time series of the sliding correlation coefficient. pVal are time series of the calculated p-values.

Appendix B. List of data file.

Name	Data
ersst3b.nino34_2.csv	Monthly Nino3.4 index provided by NOAA/CPC. NINO3.4 : Sea surface temperature. ANOM : Anomaly of NINO3.4 nino.runave : 5-month running average of ANOM.
JMA_Anomaly.csv	Climatological annual anomaly time series compiled by JMA. GL_TEMP: Global surface air temperature anomaly. NH_TEMP: Northern Hemisphere temp. anomaly. SH_TEMP: Southern Hemisphere temp. anomaly. JP_TEMP: Annual mean temperature anomaly in Japan. Compiled from 15 JMA stations. JP_RAIN: Annual total precipitation anomaly in Japan. Compiled from 51 JMA stations.
JMA_Consume.csv	Daily mean temperature and consumption value in Japan during July-August 2015. Daily consumption values were provided by e-stat of Japan. Temp: Daily mean temperature. The data were averaged value of 15 JMA stations. Watermelon: consumption value of watermelon. Icecream: consumption value of ice cream. ChineseNoodle: consumption value of Chinese noodle.
JMA_Kyoto_2.txt	Monthly mean maximum / average / minimum temperature, and monthly total precipitation at JMA Kyoto. TAVE: Monthly mean average temperature. TMAX: Monthly mean maximum temperature. TMIN: Monthly mean minimum temperature. RAIN: Monthly total precipitation.
JMA_Tokyo_2.txt	Same as JMA_Kyoto_2.txt, but for JMA Tokyo.
nino_DJF.csv	Boreal winter (Dec-Feb) mean of 5-month running averaged Nino3.4 anomaly. NINO: Nino3.4 DJF mean.
Rain_Darwin_2.csv	Monthly total precipitation at Darwin, Australia. Precipitation data were obtained from BOM,

	Australia. RAIN: Monthly total precipitation.
Rain_TMD_Ann.csv	All Thailand mean annual rainfall. Original data were obtained from TMD. Mean annual rainfall time series were simple arithmetic average of 47 stations. RR: All Thailand mean annual rainfall.
Rain_TMD_KhonKaen_2.txt	Monthly total precipitation at Khon Kaen, Thailand. Precipitation data were obtained from TMD, Thailand. RAIN: Monthly total precipitation.
Rain_TMD_UbonRatchathani_2.txt	Same as Rain_TMD_KhonKaen_2.txt, but for Ubon Ratchathani, Thailand.
Rice_Japan.csv	Time series of area of paddy field, rice production, and yield in Japan. Data were obtained from e-stat, Japan. Area: unit is ha. Production: unit is t. Yield: unit is kg/10a.
Rice_Thailand.csv	Time series of area of harvested are, rice production, and yield in Thailand. Data were obtained from FAO, and supplemented from "World Rice Statistics" by Palacpac (1977). Area: unit is ha. Production: unit is t. Yield: unit is kg/10a.
soi_DJF.csv	Boreal winter (Dec-Feb) mean SOI. SOI was 5-month running averaged first. SOI was obtained from NOAA/CPC. SOI: Southern Oscillation Index.
soi_std_3.csv	Monthly SOI and 5-month running averaged SOI. SOI was obtained from NOAA/CPC. SOI: Southern Oscillation Index (Standardized) soi.runave: 5-month running averaged SOI.