R sample of Time Series Analysis

Hsin-Yu,Ku (Matrikel Num. : 3038591)

Goal : explain basic concepts of Time Series(TS) Analysis with several TS datasets.

What is Time Series (TS) ?

TS data is a sequence of data in chronological order.

01 / Build TS data from non-TS data

ts() turns some data into a TS data.
  Use: ts(data , start=YEAR, frequency=how many frequences are there in a year)

```
vector_beforeTS <- c(2, 6, 8, 3, 9, 1, 0, 4)
TS_data <- ts(vector_beforeTS, start=2020, frequency=4)
# print() shows TS_data is a TS data inclding 4 quarters from each year 2020-2021.
print(TS_data)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2020    2    6    8    3
## 2021    9    1    0    4
```

02 / A quick inspection into time Series(TS) data

```
?LakeHuron
# LakeHuron shows annual measurements (in feet) of the level of Lake Huron in 1875-1972.
```

```
# is.ts() checks if a data set is a TS data or not. (TRUE/FALSE)
is.ts(LakeHuron)
```

```
## [1] TRUE
```

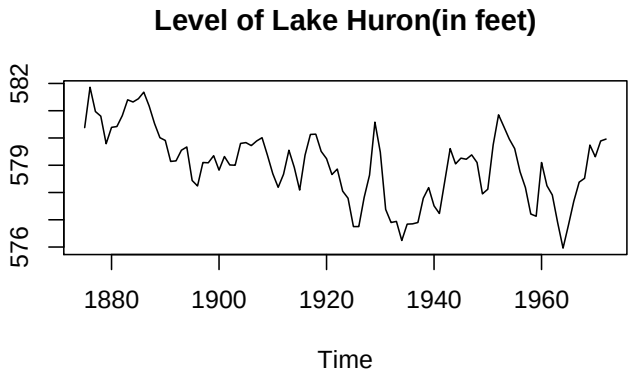
```
# print() shows that it is a TS data with start in 1875, end in 1972, frequency =1.
print(LakeHuron)
```

```
## Time Series:
## Start = 1875
## End = 1972
## Frequency = 1
##  [1] 580.38 581.86 580.97 580.80 579.79 580.39 580.42 580.82 581.40 581.32
## [11] 581.44 581.68 581.17 580.53 580.01 579.91 579.14 579.16 579.55 579.67
## [21] 578.44 578.24 579.10 579.09 579.35 578.82 579.32 579.01 579.00 579.80
## [31] 579.83 579.72 579.89 580.01 579.37 578.69 578.19 578.67 579.55 578.92
## [41] 578.09 579.37 580.13 580.14 579.51 579.24 578.66 578.86 578.05 577.79
## [51] 576.75 576.75 577.82 578.64 580.58 579.48 577.38 576.90 576.94 576.24
## [61] 576.84 576.85 576.90 577.79 578.18 577.51 577.23 578.42 579.61 579.05
## [71] 579.26 579.22 579.38 579.10 577.95 578.12 579.75 580.85 580.41 579.96
## [81] 579.61 578.76 578.18 577.21 577.13 579.10 578.25 577.91 576.89 575.96
## [91] 576.80 577.68 578.38 578.52 579.74 579.31 579.89 579.96
```

```r
# str() shows it is a TS data with 98 observations between 1875-1972
str(LakeHuron)
```

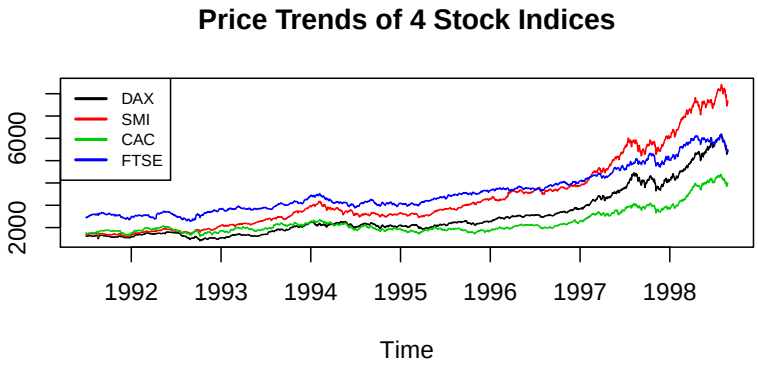
```
##  Time-Series [1:98] from 1875 to 1972: 580 582 581 581 580 ...
```

```r
# ts.plot() gives an overlook of the trend of one or more TS in a single common plot.
ts.plot(LakeHuron, main="Level of Lake Huron(in feet)", ylab=NA)
```



When there are more than 1 TS trends, use argument "col=" to display with distinct colors.

```r
?EuStockMarkets # EuStockMarkets includes daily price of 4 stock indices (1991-1998).
 # Use col() in ts.plot() to displays trends of 4 stock indices (DAX,SMI,CAC,FTSE) together.
ts.plot(EuStockMarkets,col=1:4, main="Price Trends of 4 Stock Indices")
 # Add a legend on the topleft with 0.7 as size, type of line = 1, width =2.
legend("topleft",colnames(EuStockMarkets), cex=0.7, lty=1, lwd=2, col=1:4)
```



03/ More functions to look into a TS data

start() & end() functions return the time index of the first & last observations.

```r
start(AirPassengers)
```

```
## [1] 1949    1
```

```r
end(AirPassengers)
```

```
## [1] 1960   12
```

```r
 # AirPassengers is a TS data starts on 1949 with time index 1 & ends on 1960 with time index 12
```

deltat() function returns the fixed time interval between observations.

```r
deltat(AirPassengers)
```

```
## [1] 0.08333333
```

```r
 # AirPassengers has a time interval of 0.083333 year between observations.
```

frequency() function returns the number of observations per unit time.

```r
frequency(AirPassengers)
```

```
## [1] 12
```

```r
 # AirPassengers has 12 observations per unit time(year)
```

cycle() function returns the position in the cycle of each observation.

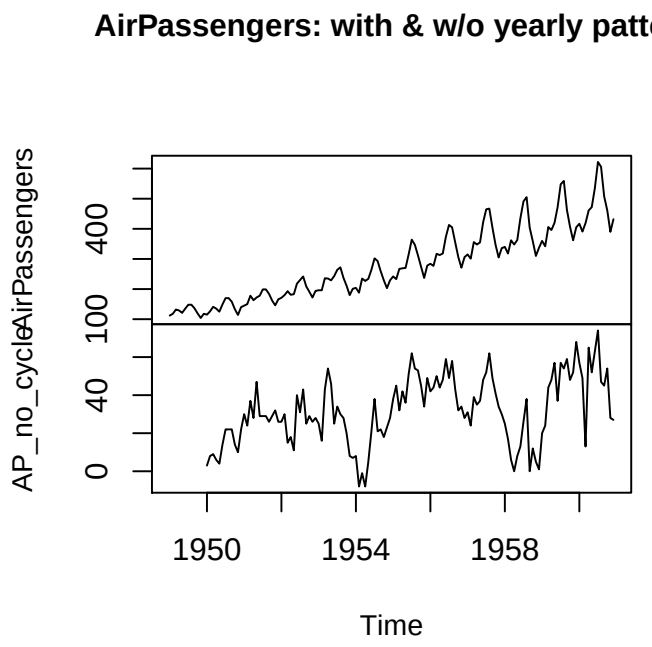
```r
cycle(AirPassengers)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949   1   2   3   4   5   6   7   8   9  10  11  12
## 1950   1   2   3   4   5   6   7   8   9  10  11  12
## 1951   1   2   3   4   5   6   7   8   9  10  11  12
## 1952   1   2   3   4   5   6   7   8   9  10  11  12
## 1953   1   2   3   4   5   6   7   8   9  10  11  12
## 1954   1   2   3   4   5   6   7   8   9  10  11  12
## 1955   1   2   3   4   5   6   7   8   9  10  11  12
## 1956   1   2   3   4   5   6   7   8   9  10  11  12
## 1957   1   2   3   4   5   6   7   8   9  10  11  12
## 1958   1   2   3   4   5   6   7   8   9  10  11  12
## 1959   1   2   3   4   5   6   7   8   9  10  11  12
## 1960   1   2   3   4   5   6   7   8   9  10  11  12
```

04/ Simple transformations of trends in TS data

Transformation by diff() : diff() deletes seasonal trend in a TS data with "lag = cycle of seasonality."

```r
AP_no_cycle <-diff(AirPassengers, lag=12)
plot.ts(cbind(AirPassengers, AP_no_cycle),
        main= " AirPassengers: with & w/o yearly pattern", cex.main=1, cex.lab=0.9)
```

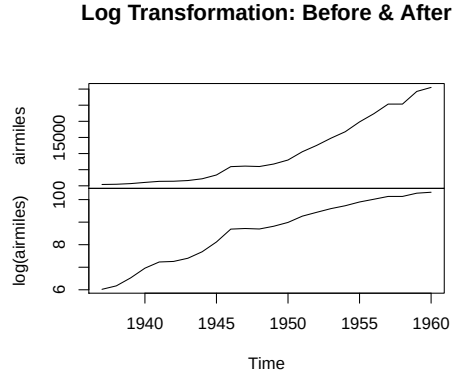



```r
# plot.ts() plots several TS on differnt bases (but with same time frequency).
# Here we see that year 1954 and 1958 have comparably low or even negative growth
  # in the overall growing trend, comparing to the same months of the previous year.
```

Transformation by log()

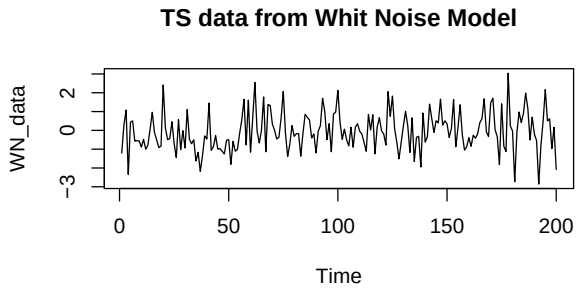
```r
?airmiles # airmiles shows miles flown by commercial airlines in the US from 1937 to 1960.
```

```r
 # log() linarlizes the rapid growth since around 1950.
plot.ts(cbind(airmiles,log(airmiles)),
        main="Log Transformation: Before & After", cex.main=1.3)
```



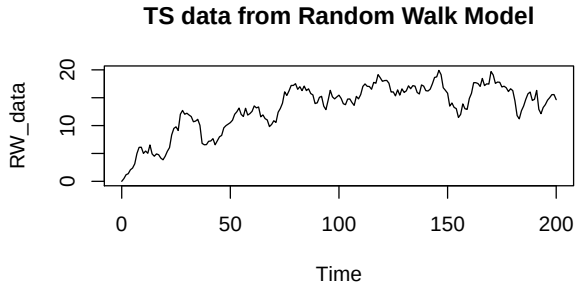
White Noise (WN) model

```
# Generating WN data with arima.sim(): order=c(0,0,0)
WN_data <- arima.sim(model =list(order=c(0,0,0)), n = 200 )
ts.plot(WN_data, main= "TS data from Whit Noise Model")
```



Random Walk (RW) model

```
# Generating RW data with arima.sim(): order=c(0,1,0)
RW_data <- arima.sim(model =list(order=c(0,1,0)), n = 200)
ts.plot(RW_data, main= "TS data from Random Walk Model")
```




```
# Use cumsum() to turn WN model data into RW model data.
RW_data2 <- cumsum(WN_data)
ts.plot(RW_data2)
title(main= list("Random Walk data
generated by cumsum() from White Noise Model",cex=1.2))
```