Tempature Forecasting via Time Series Analysis

Xinyu Zhang

2022/2/20

```
cherry <- read.csv("../data/FinalizedData/FinalData/DC.csv")
# cherry <- read.csv("../data/washingtondc.csv") %>%
#   bind_rows(read.csv("../data/liestal.csv")) %>%
#   bind_rows(read.csv("../data/kyoto.csv"))
str(cherry)
```

```
## 'data.frame':    31208 obs. of  15 variables:
##  $ location  : Factor w/ 1 level "washingtondc": 1 1 1 1 1 1 1 1 1 1 ...
##  $ lat       : num  38.9 38.9 38.9 38.9 38.9 ...
##  $ long      : num  -77 -77 -77 -77 -77 ...
##  $ alt       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ year      : int  1936 1936 1936 1936 1936 1936 1936 1936 1936 1936 ...
##  $ bloom_date: Factor w/ 86 levels "1936-04-07","1937-04-14",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ bloom_doy : int  98 98 98 98 98 98 98 98 98 98 ...
##  $ DATE      : Factor w/ 31208 levels "1936-09-01","1936-09-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ PRCP      : num  -0.261 -1.162 4.665 7.307 -1.383 ...
##  $ SNWD      : logi  NA NA NA NA NA NA ...
##  $ TAVG      : num  24.5 22.4 23.3 23.9 23.3 22.8 24.8 28.2 29.9 25.6 ...
##  $ TMAX      : num  30.5 27.7 28.6 29.4 28.8 ...
##  $ TMIN      : num  19.3 17.9 18.1 18.8 17.4 ...
##  $ Status    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ CDD       : num  24.5 46.9 70.2 94.1 117.4 ...
```

```
precip = cherry$PRCP
temp = cherry$TAVG
date = (as.Date(as.character(cherry$DATE)))
year = cherry$year
data = data.frame(precip, temp, date, year)
str(data)
```

```
## 'data.frame':    31208 obs. of  4 variables:
##  $ precip: num  -0.261 -1.162 4.665 7.307 -1.383 ...
##  $ temp  : num  24.5 22.4 23.3 23.9 23.3 22.8 24.8 28.2 29.9 25.6 ...
##  $ date  : Date, format: "1936-09-01" "1936-09-02" ...
##  $ year  : int  1936 1936 1936 1936 1936 1936 1936 1936 1936 1936 ...
```

```
head(data,2)
```

```
##       precip temp       date year
## 1 -0.2614977 24.5 1936-09-01 1936
## 2 -1.1622284 22.4 1936-09-02 1936
```

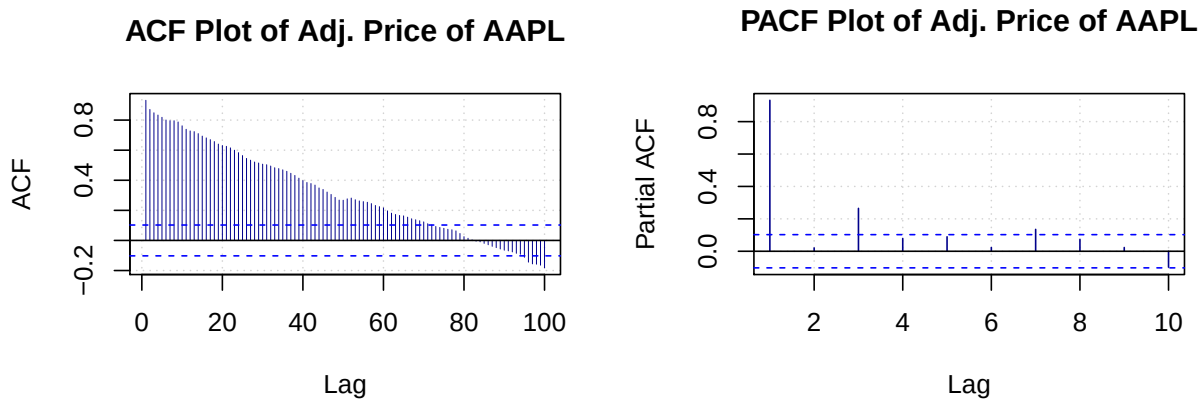
```
tail(data,2)
```

```
##       precip temp       date year
```

```
## 31207      0  4.5 2022-02-08 2022
## 31208      0  4.4 2022-02-09 2022
```

```r
data_year = data %>% filter(year==2000)
```

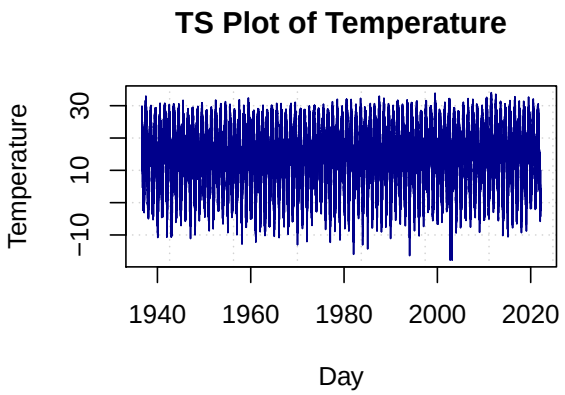
```r
par(mfrow=c(1,2))
#### Step 1: Preview
# Use ACF to identify the type of time series
acf(ts(data_year$temp), lag.max = 100, col = 'dark blue', bg = 'dark blue',
    panel.first = grid(), main = 'ACF Plot of Adj. Price of AAPL', lwd=.01)
# Based on acf plot, we know that it is more likely to be an AR process
pacf(ts(data_year$temp), lag.max = 10, col = 'dark blue', bg = 'dark blue',
     panel.first = grid(), main = 'PACF Plot of Adj. Price of AAPL')
```

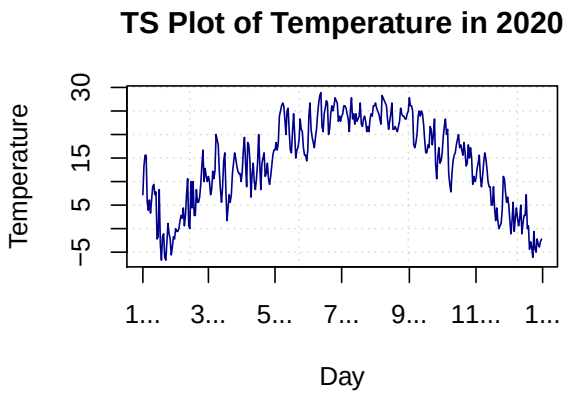



```r
# Since PACF damp out for AR(p) process when lag > p, we could guess that
# it follows AR(p) process with p around 1 (or maybe 3? Or maybe ARMA)
adf.test(ts(data_year$temp))
```

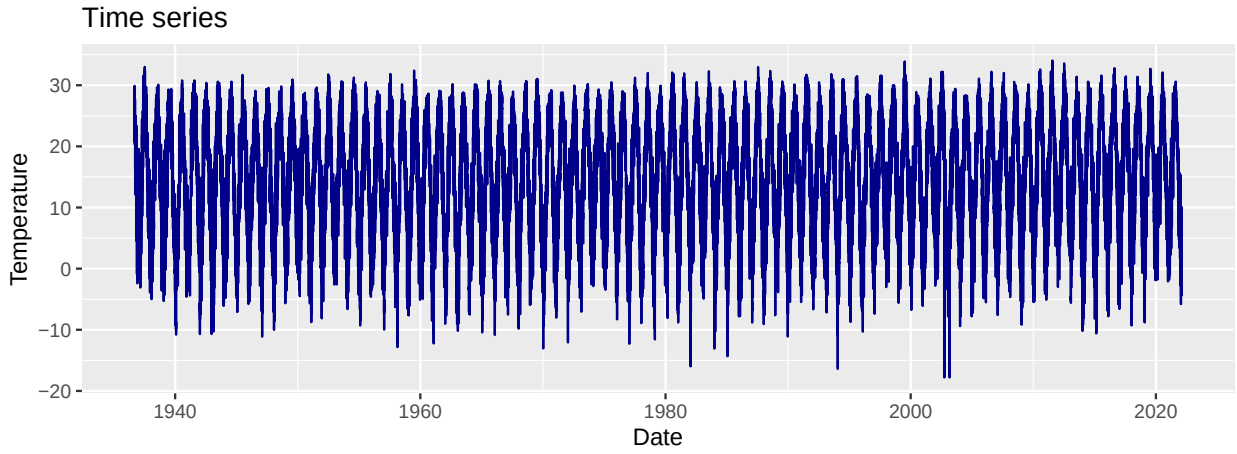
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts(data_year$temp)
## Dickey-Fuller = -0.56144, Lag order = 7, p-value = 0.9788
## alternative hypothesis: stationary
```

```r
# This means the trend has to be estimated

plot(x=data$date, y=ts(data$temp), main = "TS Plot of Temperature", type="l",
     col = 'dark blue', bg = 'dark blue',  panel.first = grid(), lty=1, ylab="Temperature", xlab="Day")

plot(x=data_year$date, y=ts(data_year$temp), main = "TS Plot of Temperature in 2020", type="l",
     col = 'dark blue', bg = 'dark blue',  panel.first = grid(), lty=1, ylab="Temperature", xlab="Day")
```






```
data_ts = as_tsibble(data, index=date)
data_ts %>%
  autoplot(temp, col="dark blue") +
  labs(x="Date", y = "Temperature", title = "Time series")
```




```
#'  Trend decomposition using additive model since the The additive
#'  decomposition is the most appropriate if the magnitude of the
#'  seasonal fluctuations, or the variation around the trend-cycle,
#'   does not vary with the level of the time series.
```

```
#### Step 2 STL Decomposition
dcmp <- data_ts %>%
  model(stl = STL(temp ~ trend(window=7), robust=TRUE)) %>%
  components() %>%
  select(-.model)
head(dcmp)
```

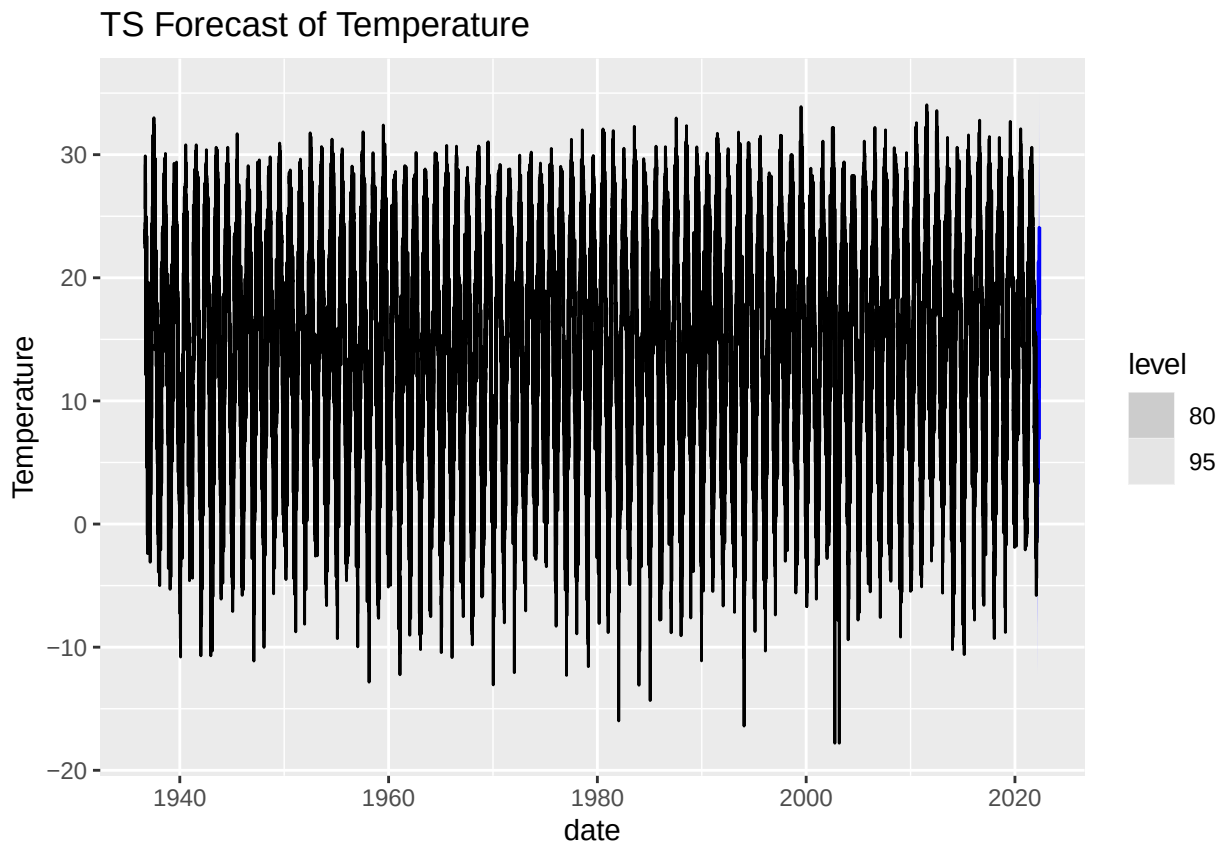
```
## # A tsibble: 6 x 7 [1D]
##   date        temp trend season_year season_week remainder season_adjust
##   <date>     <dbl> <dbl>       <dbl>       <dbl>     <dbl>          <dbl>
## 1 1936-09-01  24.5  12.6       10.9         1.15    -0.211           12.4
## 2 1936-09-02  22.4  12.7        9.78        1.31    -1.40            11.3
## 3 1936-09-03  23.3  12.8       11.1        -0.747    0.139           13.0
## 4 1936-09-04  23.9  13.3       10.9        -0.701    0.403           13.7
```

```
## 5 1936-09-05  23.3  14.1        9.99       0.744    -1.54           12.6
## 6 1936-09-06  22.8  15.1        9.11      -1.99      0.635          15.7
```

```
#### Step 3 Forecasting
h = 100
fcst = dcmp %>%
  model(SNAIVE(temp~lag("year"))) %>%
  forecast(h=h)
```

```
## Warning: Non-integer lag orders for random walk models are not supported.
## Rounding to the nearest integer.

## Warning: Non-integer lag orders for random walk models are not supported.
## Rounding to the nearest integer.
```

```
fcst %>% autoplot(dcmp) +
  labs(y = "Temperature",
       title = "TS Forecast of Temperature")
```




```
temprature_fcst = fcst$.mean


date.f = seq(from=tail(data$date,1), to=tail(data$date,1)+h, length.out=h)
data.forecast = data.frame(temp=temprature_fcst, date=date.f)
data_total = rbind(data[,c('temp','date')],data.forecast)
str(data.forecast)
```

```
## 'data.frame':    100 obs. of  2 variables:
```

```
##  $ temp: num  3.6 1.6 -0.5 -0.8 -0.1 3.1 3.8 -0.6 -1.1 0.5 ...
##  $ date: Date, format: "2022-02-09" "2022-02-10" ...
```

```
write.csv(data.forecast,"../data/temp_forecast.csv")
```