# Task 3 - Time Series

Benjamin Wiriyapong

## Data Pre-processing

In task 3, time series models of weather station data from **Hurn** and **Sheffield** in the UK is fitted and compared. In this task we will make an attempt to try to fit time series model to predict **sun** variable. The datasets contain information about the following variables:

- yyyy - Year
- mm - Month
- tmax - Mean daily maximum temperature
- tmin - Mean daily minimum temperature
- af - Days of air frost
- rain - Total rainfall
- sun - Total sunshine duration

First of all, let us import library as well as reading in datasets to our workspace. Since we will work with time series on **sun** variable for both Hurn and Sheffield, **month** and **year** columns are combined into **dates** column. Then, **dates** is turned to date datatype.

```
# install and import all libraries via pacman
if (!require("pacman")) install.packages("pacman")
pacman::p_load(stringr, lubridate, splitstackshape, ggplot2, dplyr, rtsplot,
               forecast, extrafont, latex2exp, tseries)

# read in dataset and fix column names
hurn_df <- read.csv("hurn.txt", header = TRUE)
hurn_df_colnames <- strsplit(gsub(x = names(hurn_df), pattern = "\\.+",
                    replacement = " "), " ")
hurn_df <- cSplit(hurn_df, colnames(hurn_df), " ")
colnames(hurn_df) <- hurn_df_colnames[[1]]
hurn_df$sun <- gsub("[^0-9.]", "", hurn_df$sun)
hurn_df$sun <- as.numeric(hurn_df$sun)
hurn_df[hurn_df==""] <- NA
hurn_df <- na.omit(hurn_df)

sheffield_df <- read.csv("sheffield.txt", header = TRUE)
sheffield_df_colnames <- strsplit(gsub(x = names(sheffield_df),
                         pattern = "\\.+", replacement = " "), " ")
sheffield_df <- cSplit(sheffield_df, colnames(sheffield_df), " ")
colnames(sheffield_df) <- sheffield_df_colnames[[1]]
sheffield_df$sun <- as.numeric(sheffield_df$sun)

# create ts objects
hurn_df$dates <- as.Date(paste(hurn_df$yyyy, hurn_df$mm, "01", sep = "-"),
                 format = ("%Y-%m-%d"))
hurn_df <- hurn_df[order(hurn_df$dates), ]
```

```
hurn_ts <- ts(hurn_df[,"sun"], start = c(1968,12),frequency = 12)
sheffield_df$dates <- as.Date(paste(sheffield_df$yyyy, sheffield_df$mm, "01",
                      sep = "-"), format = ("%Y-%m-%d"))
sheffield_df <- sheffield_df[order(sheffield_df$dates), ]
sheffield_ts <- ts(sheffield_df[,"sun"], start = c(1957,1), frequency = 12)
```

## 1. Plot the time series data

Time series plot will help illustrate the effects in the series for further analysis. The effects that will be inspected are:

- Moving average models are the last n past forecast errors which still affect the current time value. We refer to this as a MA(q) model, a Moving Average model of order q is a number of past error terms we are considering in the model setting.
- Autoregressive models are linear combination of past values of the variable used to predict tyhe current time value. We refer to this as AR(p) model, an autoregressive model of order p is a number of past values of the variable of interest.
- Trend is the overall pattern of the series whether the it is increasing or decreasing.
- Seasonality refers to fluctuations in the data related to calendar cycles.

```
tsdisplay(hurn_ts)
```

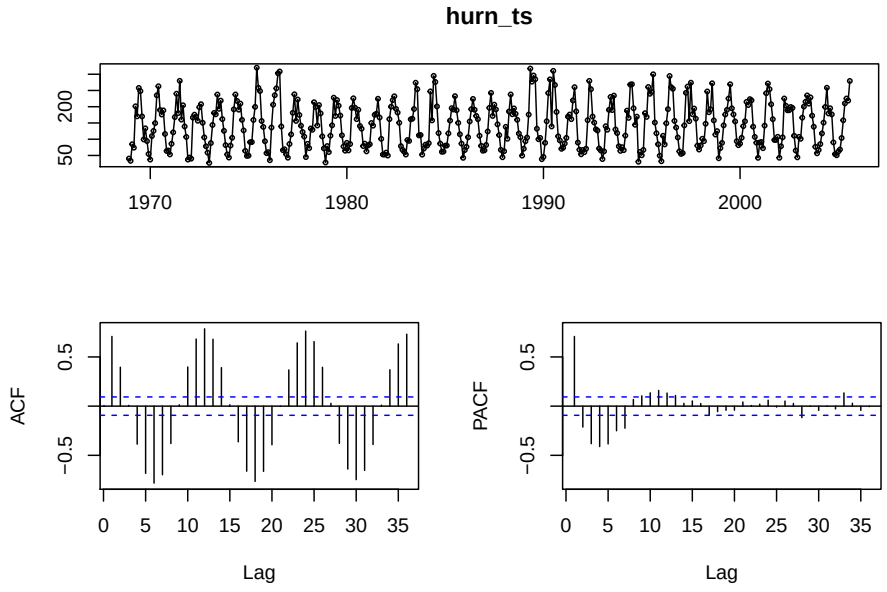


Figure 1: Hurn time series plot

Hurn data are not stationary because although there is no trend in the dataset, seasonal effects still exist. Hurn time series plot shows systematic fluctuations over time *(figure 1)*. Further, ACF plot shows systematic correlation every 12 lags. Similarly, PACF also shows the effect of seasonality.

```
tsdisplay(sheffield_ts)
```

For Sheffield plot, it also shows seasonality in the series even though there is no trend *(figure 2)*. The time series plot shows systematic fluctuation. Moreover, ACF illustrates the periodic effect of data for every 12 lags. Additionally, PACF shows similar pattern. As a result, this series is not stationary.

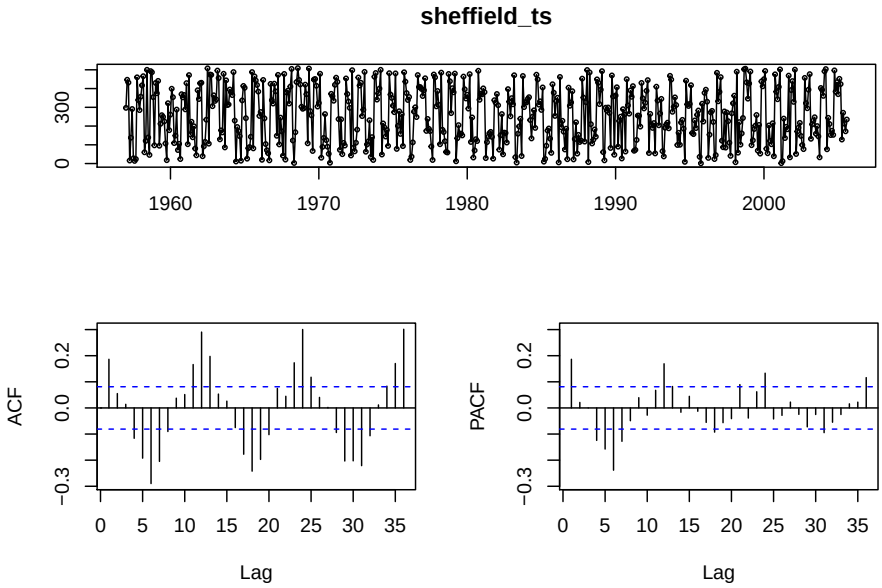


Figure 2: Sheffield time series plot

## 2 Examine seasonality in data

Seasonality usually causes the series to be non-stationary because the average values at some particular times within the seasonal span is different than the average values at other times. As seen from previous section, the datasets are not stationary because there are seasonality effects in the datasets. Therefore, in this section we will attempt to decpmpose the seasonal effect from out data.

Seasonal differencing is a difference between a value at time t and a value with lag that is a multiple of seasonality(S). This is a technique used to remove the effect of seasonality. In this report. we consider 12 to be the period of seasonality based upon figure 1 and 2.

```
no_season_hurn_ts = diff(hurn_ts, differences = 12)
plot.ts(no_season_hurn_ts)
no_season_sheffield_ts = diff(sheffield_ts, differences = 12)
plot.ts(no_season_sheffield_ts)
```

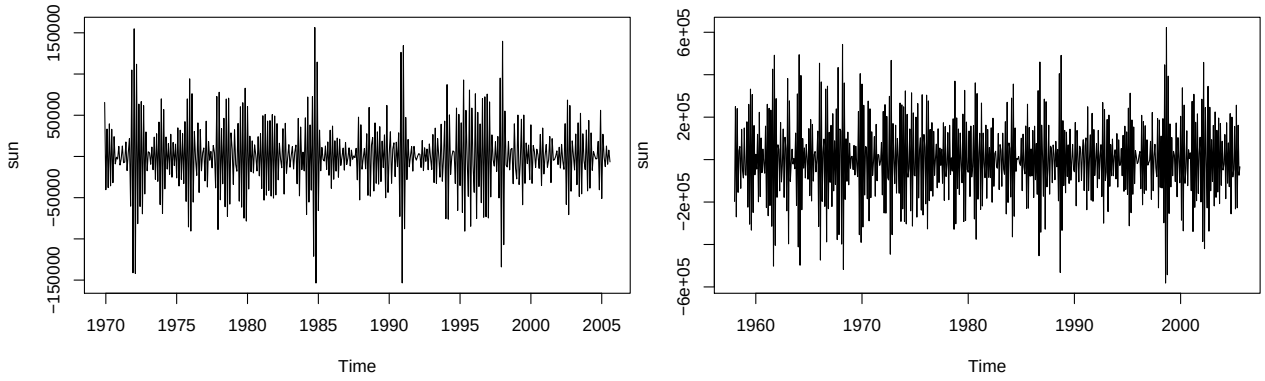


Figure 3: Seasonal difference plot

After doing 12 lags difference, both series *(figure 3)* look stationary enough for both mean and variance. Now they are ready for modeling.

## 3. Estimate model parameters of (p,d,q) and (P,D,Q)S

```
acf(no_season_hurn_ts)
pacf(no_season_hurn_ts)
```

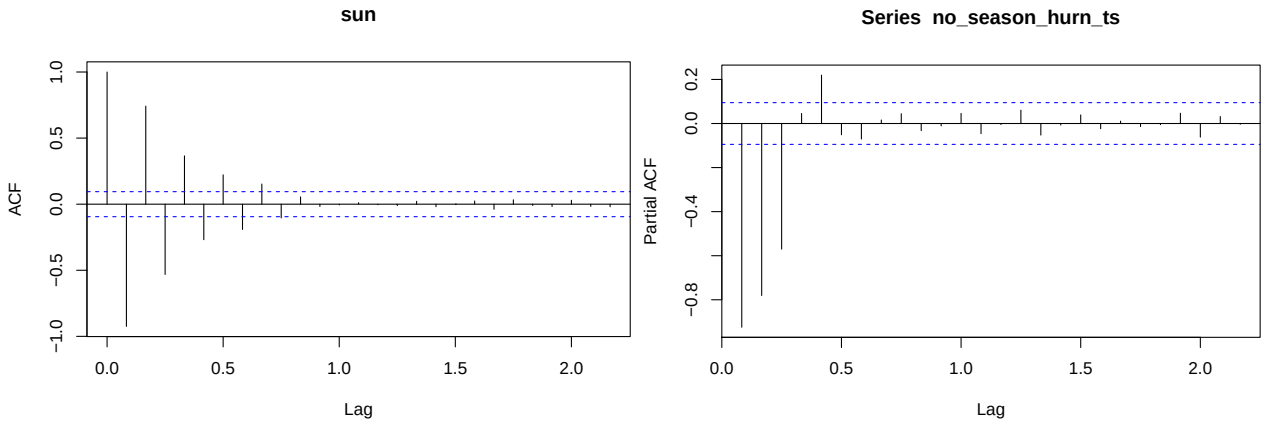


Figure 4: Hurn ACF(left) and PACF(right) plots

**Non-seasonal behavior:** The first 9 lags of ACF are still significant *(figure 4 left)*. Therefore, it is possible that this signal can have the effect of nine white noise back which can indicate MA(9) process. This means that the time series may depend on the effect of nine past errors back. However, this seems to be too complex parameter for the model. PACF, on the other hand, are cut at lag 2 which means this is mostly an AR(2) process *(figure 4 right)*. This means that the current time series may rely on the past previous two values in the series.

**Seasonal behavior:** When looking at every 12 lags *(figure 4)*, there is no sign of seasonality effect. Hence, the effects are seasonal MA(0) and AR(0).

The model for Hurn is SARIMA(2,0,9)(0,1,0)12

```
acf(no_season_sheffield_ts)
pacf(no_season_sheffield_ts)
```

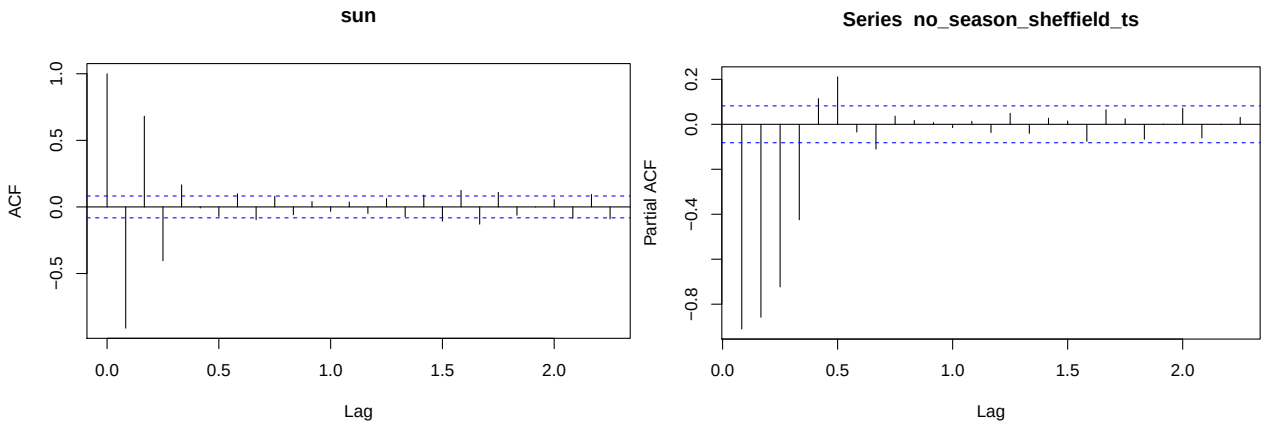


Figure 5: Sheffield ACF(left) and PACF(right) plots

**Non-seasonal behavior:** ACF of Sheffield show significance of autocorrelation until lag 4 which indicates MA(4) process *(figure 5 left)*. This indicates that the four past error are affected the current time series value. On the other hand, PACF are significant until lag 5 which indicate AR(5) process *(figure 5 right)*.

**Seasonal behavior:** There is no significant lag at every 12 data *(figure 5)*. Hence, it is assumed that effects are seasonal MA(0) and AR(0).

The model for Sheffield is SARIMA(4,0,5)(0,1,0)12

## 4 Discussion

***Hurn AIC***

```
AIC(arima(no_season_hurn_ts, order=c(2, 0, 9),
      seasonal=list(order=c(0,1,0), period=12)))
```

```
## [1] 6953.777
```

***Sheffield AIC***

```
AIC(arima(no_season_sheffield_ts, order=c(4, 0, 5),
      seasonal=list(order=c(0,1,0), period=12), method="ML"))
```

```
## [1] 10811.23
```

The model we fitted for **Hurn** is **SARIMA(2,0,9)(0,1,0)12** while **Sheffield** model is **SARIMA(4,0,5)(0,1,0)12**. The past error that affect Sun duration in Hurn is last two past error while four past error affects Sun duration in Sheffield. Neither Hurn nor Sheffield shows visible trend. As a consequence, regression is not fitted in order to remove trend. Hurn parameters show that the past nine values in the series still affects current value while merely past five value affect the current sun duration in Sheffield.

Now let us look at the seasonality effect. According to figure 1 and figure 2, it can be seen that the datasets show correlation every 12 lags. Therefore, 12 difference is applied to remove the effect of seasonality. However, both ACF and PACF for both datasets shows no significant seasonality effect from moving average and autoregressive.

AIC is used to compute the performance of the models. Hurn AIC yields 6953.77 while Sheffield AIC is 10811.23. From using ACF and PACF to fit the model, it is concluded that Hurn model perform slightly better than Sheffield. However, it seems that the AICs are still relatively high. There might be better model than these if we run **auto.arima()** to search on other models parameters. Then we can compare the result against our current model to see which models perform better. In addition, we may also try to apply ARCH (autoregressive conditionally heteroscedastic) or GARCH (generalised autoregressive conditionally heteroscedastic) to describe a change, possibly volatile variance in the datasets. Lastly, it is recommended to check the residuals as they should be uncorrelated and appear to be a white noise.