Modelling Sollar Radiation in Hanford, California

Linchuan Yang

PSTAT 174

Abstract

This project will focusing on Sollar Radiation in Hanford, CA. By using the Box-Jenkins Method, the time-series relationship between time and solar radiation will be analyzed, and a time-series model will be created to forecast the sollar radiation in Hanford, CA.

Introduction

Solar energy is a renewable energy source. Refers to the sun's thermal radiation energy, the main performance is often said that the sun's rays. It is generally used for power generation in modern times.Since the birth of creatures on earth, they have mainly survived on the thermal radiation provided by the sun. With the decrease of un-renewable energy sources such as fossil fuels, solar energy has become an important part of the energy used by human beings and has been continuously developed.

The dataset is from Earth System Research Laboratory, Global Monitoring Laborator. The dataset contains the solar radiation data of Hanford, CA from Feburary 1st, 2002 to current time.

Source:
Hanford, California, United States (HNX) Continuous in-situ measurements of solar radiation.Earth System Research Laboratory, Global Monitoring Laboratory
URL: https://gml.noaa.gov/aftp/data/radiation/solrad/hnx/

Variables:

var name	var type	var description	var used
year	integer	year, i.e., 2002	x
jday	integer	Julian day (1 through 365 [or 366])	
month	integer	number of the month (1-12)	x
day	integer	day of the month(1-31)	
hour	integer	hour of the day (0-23)	
min	integer	minute of the hour (0-59)	
dt	real	decimal time (hour.decimalminutes),e. g., 23.5 = 2330	
zen	real	solar zenith angle (degrees)	
dw_psp	real	downwelling global solar (Watts m^-2)	x
direct	real	direct solar (Watts m^-2)	
diffuse	real	downwelling diffuse solar (Watts m^-2)	
uvb	real	global UVB (milliWatts m^-2)	
uvb_temp	real	UVB temperature (C) – 25 deg. C is normal	
qc_dwpsp	integer	quality control parameter for downwelling global solar (0=good)	

var name	var type	var description	var used
qc_direct	integer	quality control parameter for direct solar (0=good)	
qc_diffuse	integer	quality control parameter for diffuse solar (0=good)	
qc_uvb	integer	quality control parameter for UVB irradiance (0=good)	
qc_uvb_temp	integer	quality control parameter for the UVB instrument temperature (0=good)	
std_dw_psp	real	standard deviation of the 1-sec. samples for global solar (dw_psp)	
std_direct	real	standard deviation of the 1-sec. samples for direct solar	
std_diffuse	real	standard deviation of the 1-sec. samples for diffuse solar	
std_uvb	real	standard deviation of the 1-sec. samples for uvb	

This project is only focusing on three variables(year, month, dw_psp), therefore, the raw data will be preprocessed to monthly data. After, the data are split to two set, training set(2003.01-2020.12) and testing set(2021.01-2021.12) for model validation.

This project is strictly following the Box-Jenkins Modelling Method.
During model identification, data is differenced first with apporiate differencing times by using Augmented Dickey-Fuller test. By ploting the acf and pacf plot of differenced data about 72 models are nominated.
During model estimation, 72 nominated models are being checked with AICc criterion. Only best three models are nominated to be diagnosed.
During model diagnose, data are fit with three best nominated models. To choose the best model for model forecating, Ljung-Box test and Shapiro-Wilk test are performed and acf, pacf, histigram, qqplot are plotted to test the independence and normality of the residuals. During model forecast, the training data are fit with the best model and predict the solar radiation values in 2021. To validate the data, the compared graph of predicted value and true value are plotted along with confidence interval of predicted value.

Read Raw Data & Preprocessing

```r
setwd("hnx_solar")
# read_list <- list.files(pattern = "\\.dat")

# merge <- read.table(file = "hnx03001.dat", skip = 2, header = FALSE)

# colnames(merge) = c("year","jday","month","day","hour","min",
#                     "dt","zen","dw_psp","qc_dwpsp","direct","qc_direct",
#                     "diffuse","qc_diffuse","uvb","qc_uvb",
#                     "uvb_temp","qc_uvb_temp","std_dw_psp",
#                     "std_direct","std_diffuse","std_uvb")

# merge[merge == -9999.9] <- NA

# merge <- data.frame(merge["year"], merge["month"], merge["day"], merge["dw_psp"])

# for (file in read_list[-which(read_list == "hnx03001.dat")]) {
#     print(paste("processing:", file))
#     new = read.table(file, skip = 2)
#     colnames(new) = c("year","jday","month","day","hour","min",
#                       "dt","zen","dw_psp","qc_dwpsp","direct","qc_direct",
#                       "diffuse","qc_diffuse","uvb","qc_uvb",
#                       "uvb_temp","qc_uvb_temp","std_dw_psp",
#                       "std_direct","std_diffuse","std_uvb")

#     new[new == -9999.9] <- NA
#     new <- data.frame(new["year"], new["month"], new["day"], new["dw_psp"])

#     merge = rbind(merge, new)
# }

# data = merge %>% group_by(year, month) %>%
#                 summarize(monthly_dw_psp = mean(dw_psp, na.rm = TRUE))

# write.csv(data, file = "hnx_solar_monthly.csv", col.names = TRUE)

# read from merged monthly data
data = read.csv("hnx_solar_monthly.csv")$monthly_dw_psp
data.train = data[1:216]
data.test = data[217:228]
```

Since reading from large amount of files will take a long time, for future use, the data has been merged to a single csv file with only needed variables.

There are also bad data that equals to -9999.9, so NA are filled while merging. Since only monthly data were focused, the bad data in monthly downwelling global solar variable where ignored and filled with mean value.

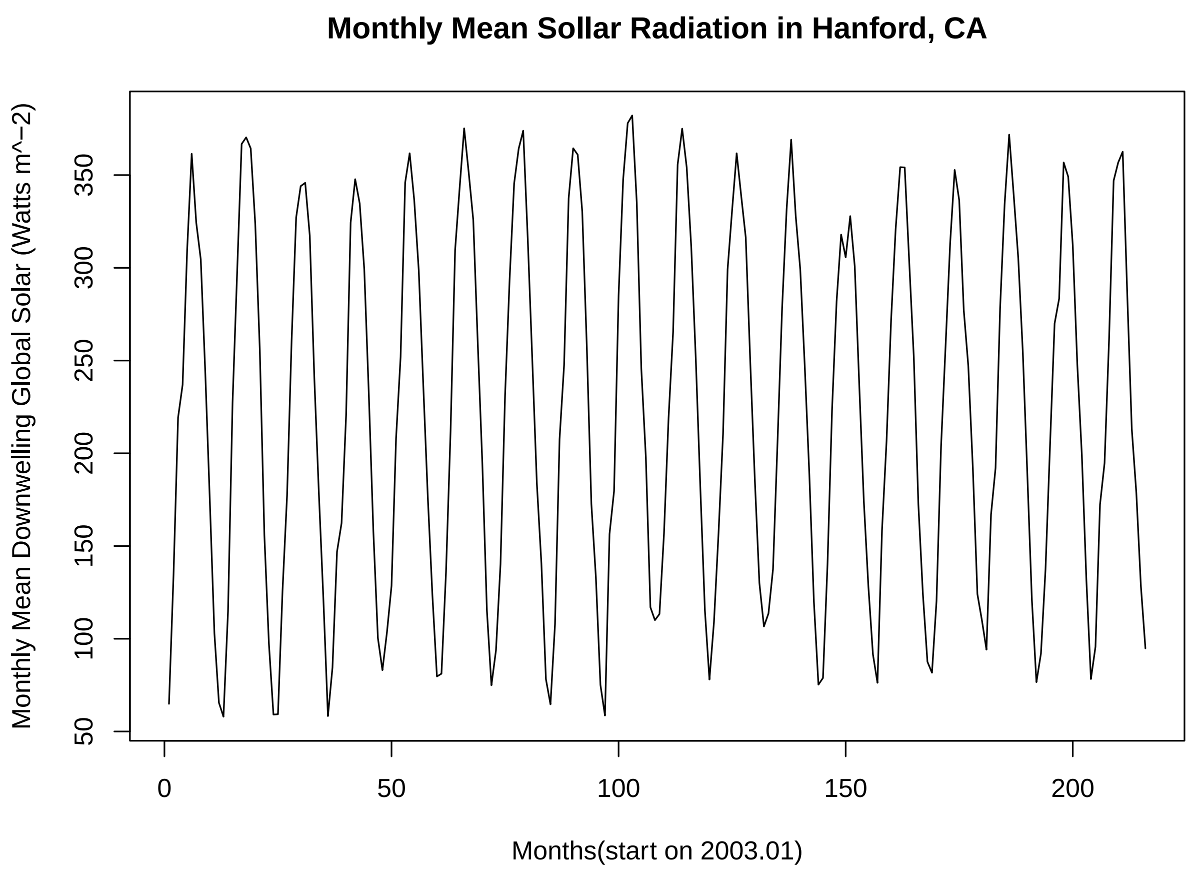
At last, the data are splited into train data(date:2003.01-2020.12) for training and test data(date:2021.01-2021.12) for validating.

Model Identification:

Analysing from the Original Data

In order to start the future analysing, we plot the original time series data first.

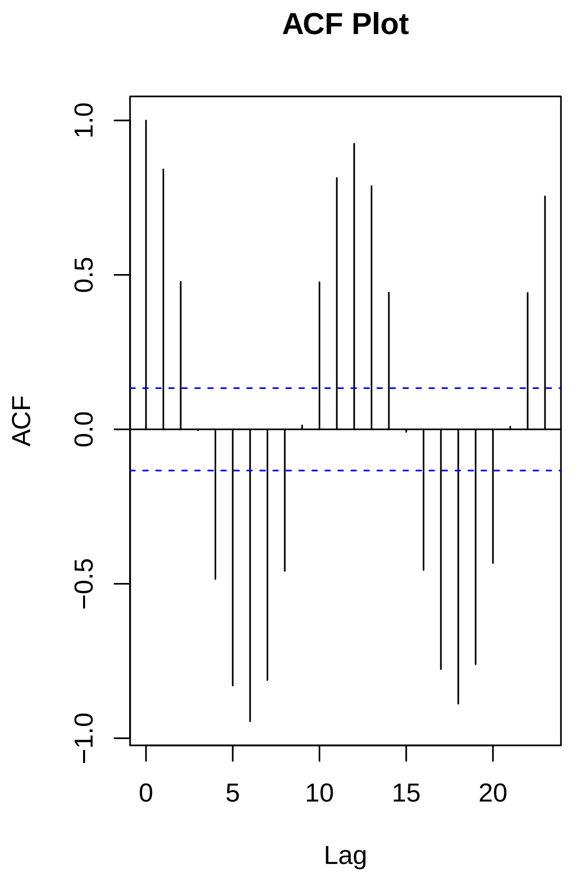
```r
# plot the original data
ts.plot(data.train, ylab = "Monthly Mean Downwelling Global Solar (Watts m^-2)",
                    xlab = "Months(start on 2003.01)")
title("Monthly Mean Sollar Radiation in Hanford, CA")
```

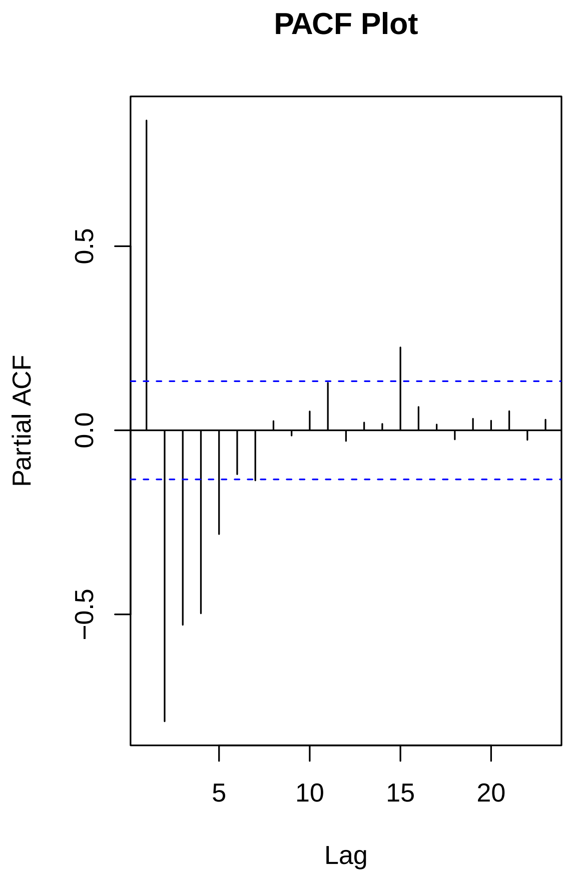


We can see the seasonality pattern is shown in the plot.

Next, we plot the ACF and PACF plot.

```r
par(mfrow=c(1,2))
acf(data.train, main = "ACF Plot")
pacf(data.train, main = "PACF Plot")
```





We can see from the ACF is kept bouncing and PACF is dying out.
Since ACF is bouncing, the original data is not good for modelling.
Therefore, we will need to do differencing on the original data.

Dickey-Fuller Test

In order to known the appropriate number of differencing time to the original data, we will do the Augmented Dickey-Fuller Test.

```
adf.test(data)
```

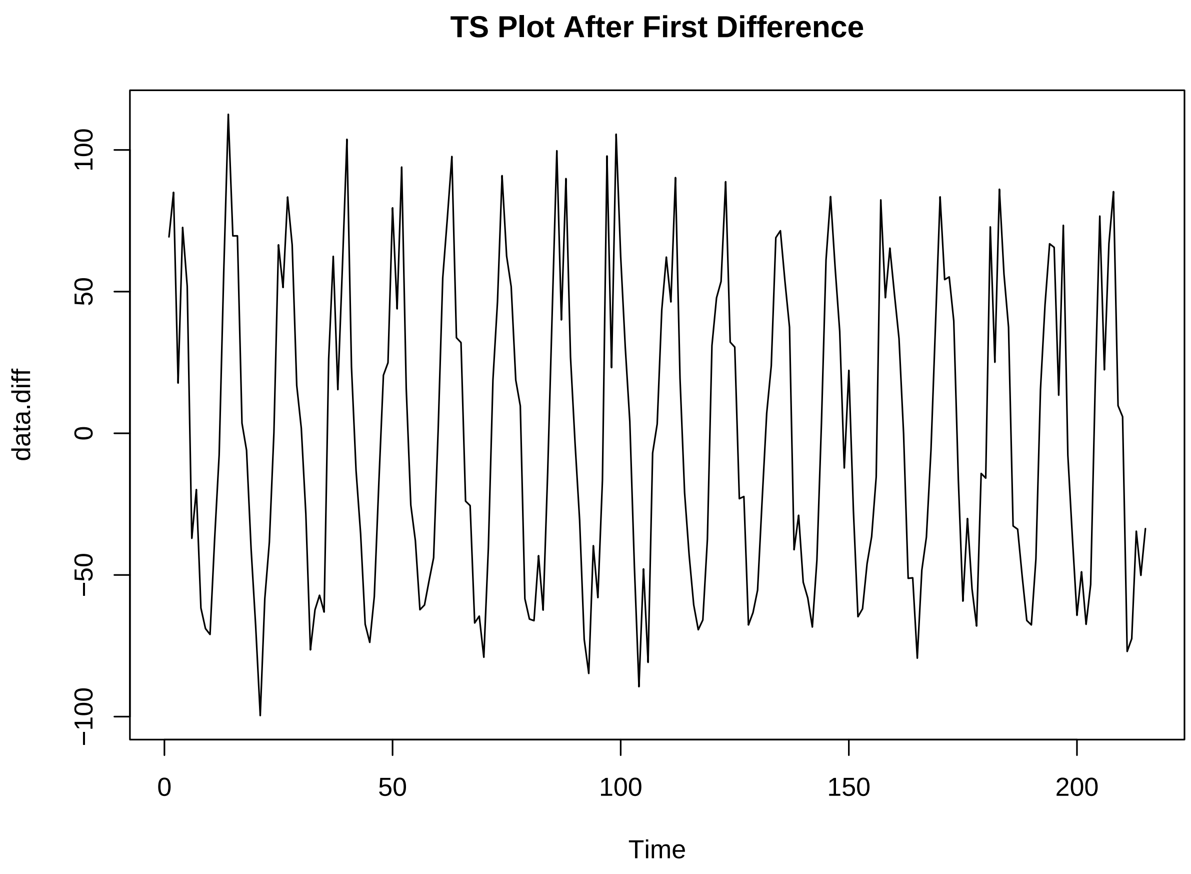
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 -1.67  0.0926
## [2,]   1 -4.37  0.0100
## [3,]   2 -3.78  0.0100
## [4,]   3 -2.17  0.0306
## [5,]   4 -1.24  0.2363
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0  -4.27    0.01
## [2,]   1 -14.45    0.01
## [3,]   2 -20.05    0.01
## [4,]   3 -20.58    0.01
## [5,]   4 -16.97    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0  -4.25    0.01
## [2,]   1 -14.42    0.01
## [3,]   2 -20.04    0.01
## [4,]   3 -20.57    0.01
## [5,]   4 -17.04    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

From the test result, we can see taking difference one time and two times is appropriate based on the p-value.

Take First Difference

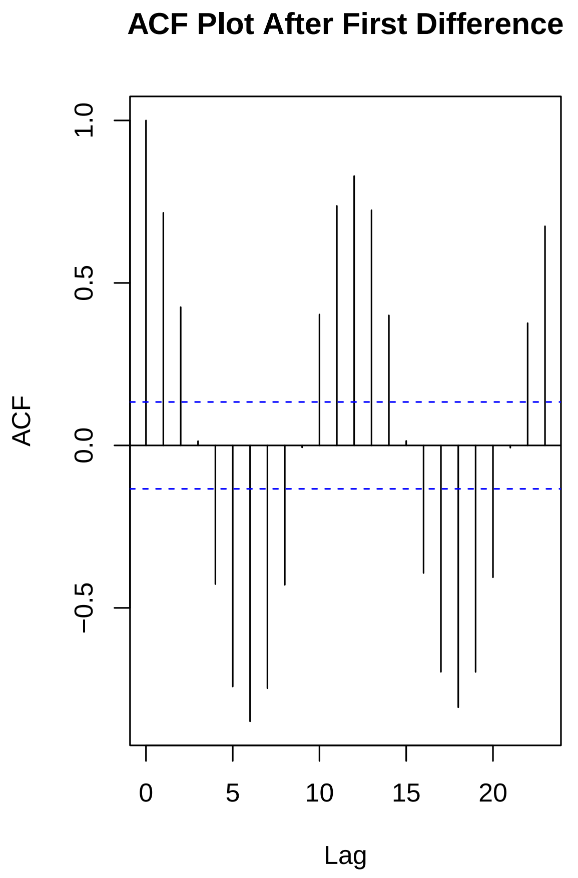
We will try to take one difference on the original data.

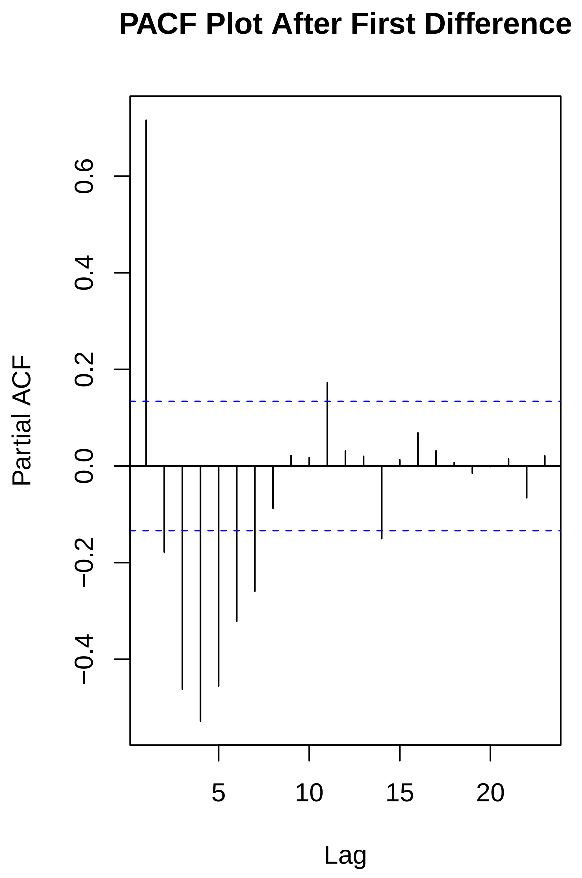
```r
# take first difference
data.diff = diff(data.train)
ts.plot(data.diff)
title("TS Plot After First Difference")
```



Next, we plot the ACF and PACF plot of the first difference data.

```r
par(mfrow=c(1,2))
acf(data.diff, main = "ACF Plot After First Difference")
pacf(data.diff, main = "PACF Plot After First Difference")
```



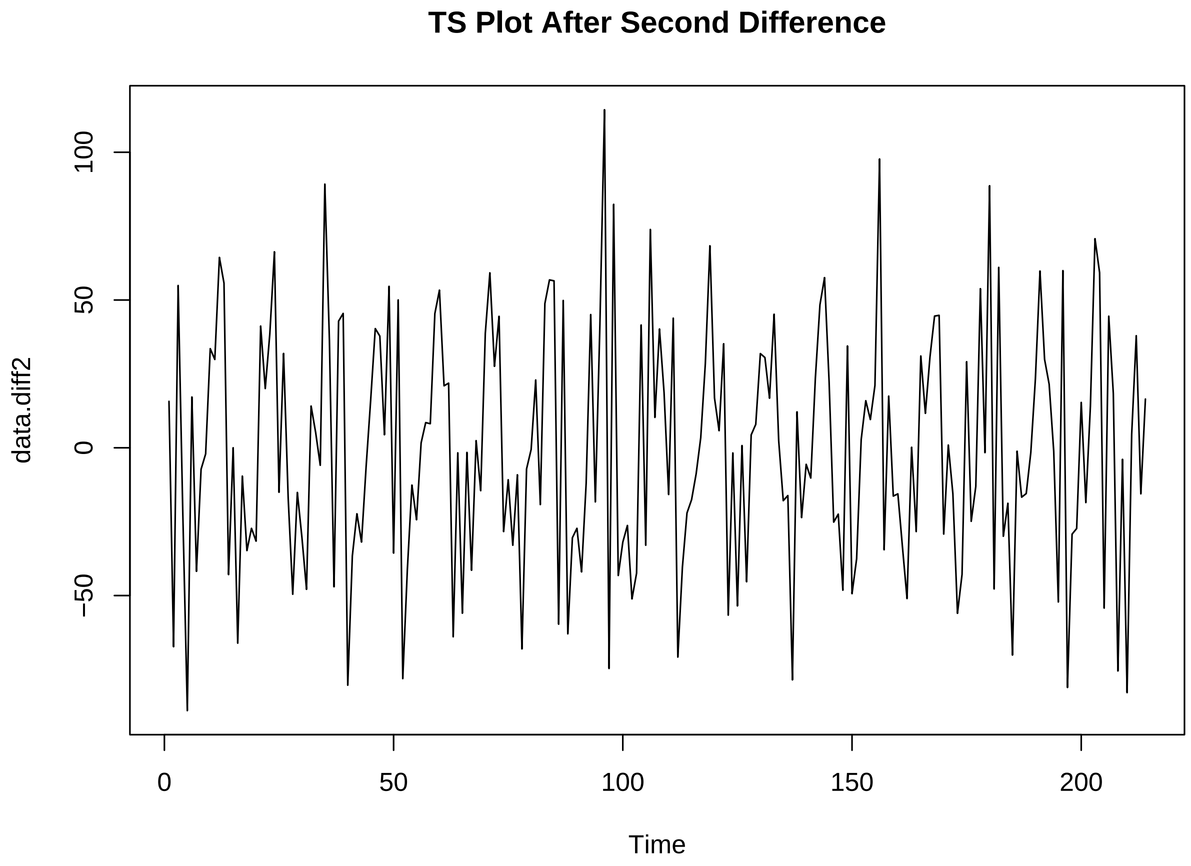


Since there are still seasonal patterns on the first difference data, ACF plot is still showing bouncing pattern, and PACF is still showing dying out pattern.
Therefore, we will try to take another difference on the data.

Take Second Difference

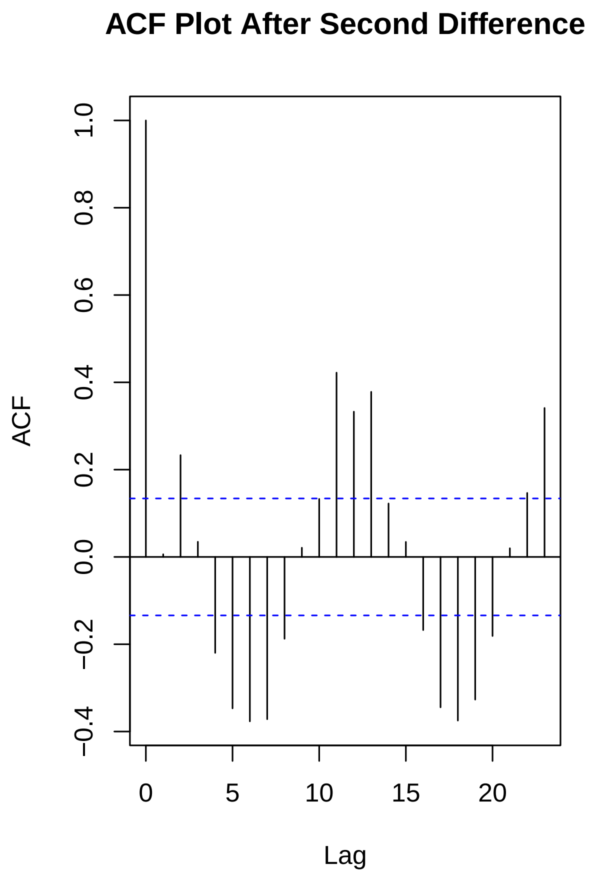
We will try to take the second difference on the original data.

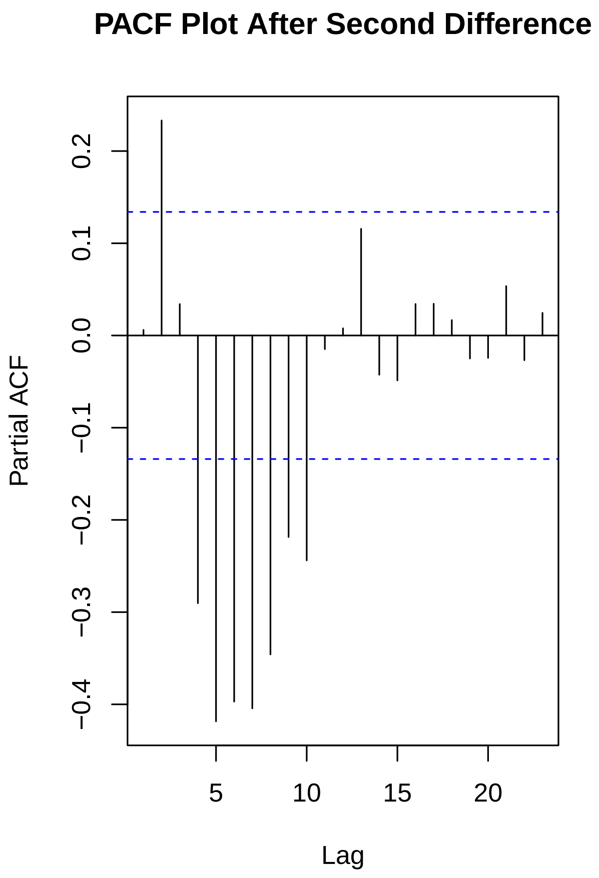
```r
# take second difference
data.diff2 = diff(data.diff)
ts.plot(data.diff2)
title("TS Plot After Second Difference")
```



Next, we plot the ACF and PACF plot of the second difference data.

```r
par(mfrow=c(1,2))
acf(data.diff2, main = "ACF Plot After Second Difference")
pacf(data.diff2, main = "PACF Plot After Second Difference")
```





From the time series plot, we can see the data is turning to a white noise pattern and PACF is still showing a die out pattern which are good for modelling.
However, the ACF plot is still showing the bouncing pattern which is not good for modelling.

Model Nominees

Based on the anlysis above, SARIMA model will be an appropriate model for the data due to its seasonal pattern.

Now, let's decide the parameters for the SARIMA model.
Since the data show seaonality with loop of 1 year, therefore, S should be 12.
Since Q and q is not decidable based on the bouncing patterns of all the ACF plot, so we will try different values(1~12) during the Model Estimation.
Since all PACF plot are showing an expotental decay pattern, therefore, P should be 1.

Since we take first and second difference, d can be either 1 or 2.
For d = 1:
Since PACF plot showing significant after Lag 7, therefore, p should be 7.
For d = 2:
Since PACF plot showing significant after Lag 10, therefore, p should be 10.

Therefore, the nominees are:
For d = 1, p = 7:

$$SARIMA(p=7,d=1,q=1)(P=1,D=1,Q=1)_{S=12}$$
$$SARIMA(p=7,d=1,q=1)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=7,d=1,q=1)(P=1,D=1,Q=3)_{S=12}$$
$$\vdots$$
$$SARIMA(p=7,d=1,q=1)(P=1,D=1,Q=6)_{S=12}$$

$$SARIMA(p=7,d=1,q=2)(P=1,D=1,Q=1)_{S=12}$$
$$SARIMA(p=7,d=1,q=2)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=7,d=1,q=2)(P=1,D=1,Q=3)_{S=12}$$
$$\vdots$$
$$SARIMA(p=7,d=1,q=2)(P=1,D=1,Q=6)_{S=12}$$
$$SARIMA(p=7,d=1,q=3)(P=1,D=1,Q=1)_{S=12}$$
$$\vdots$$
$$\vdots$$
$$SARIMA(p=7,d=1,q=6)(P=1,D=1,Q=6)_{S=12}$$

For d = 2, p = 10:

$$SARIMA(p=10,d=2,q=1)(P=1,D=1,Q=1)_{S=12}$$
$$SARIMA(p=10,d=2,q=1)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=10,d=2,q=1)(P=1,D=1,Q=3)_{S=12}$$
$$\vdots$$
$$SARIMA(p=10,d=2,q=1)(P=1,D=1,Q=6)_{S=12}$$

$$SARIMA(p=10,d=2,q=2)(P=1,D=1,Q=1)_{S=12}$$
$$SARIMA(p=10,d=2,q=2)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=10,d=2,q=2)(P=1,D=1,Q=3)_{S=12}$$
$$\vdots$$
$$SARIMA(p=10,d=2,q=2)(P=1,D=1,Q=6)_{S=12}$$
$$SARIMA(p=10,d=2,q=3)(P=1,D=1,Q=1)_{S=12}$$
$$\vdots$$
$$\vdots$$
$$SARIMA(p=10,d=2,q=6)(P=1,D=1,Q=6)_{S=12}$$

Model Estimation:

Now, for model estimation, we will use AICc value as a criterion to check which model fit the best.

```
# aiccs <- matrix(NA, nr =6, nc = 6)
# dimnames(aiccs) = list(q = c(1:6), Q = c(1:6))
```

For d = 1, p = 7:

```
setwd("hnx_solar")

# Model Estimation for d=1, p=7
# for (q in 1:6) {
#     for (Q in 1:6) {
#         try({
#             aiccs[q,Q] = sarima(xdata = data.train,
#                                 p = 7, d = 1, q = q,
#                                 P = 1, D = 1, Q = Q, S = 12)$AICc
#             print(paste("d1p7: {", " q: ", q, "Q: ", Q, "}"))
#         })
#     }
# }
# write.csv(aiccs, "d1p7_aicc.csv", col.names = TRUE, row.names = TRUE)

read.csv("d1p7_aicc.csv", header = TRUE)
```

```
##   X    X1    X2    X3    X4    X5    X6
## 1 1 8.575 8.548 8.554 8.555 8.559 8.566
## 2 2 8.579 8.549 8.557 8.557 8.562 8.567
## 3 3 8.582 8.543 8.549    NA    NA 8.572
## 4 4 8.594 8.555 8.561 8.595 8.607 8.600
## 5 5 8.595 8.561 8.557 8.559 8.568 8.573
## 6 6 8.607 8.610 8.606 8.602    NA    NA
```

For d = 2, p = 10:

```
setwd("hnx_solar")

# Model Estimation for d=2, p=10
# for (q in c(1:6)) {
#     for (Q in c(1:6)) {
#         try({
#             aiccs[q,Q] = sarima(xdata = data.train,
#                                 p = 10, d = 2, q = q,
#                                 P = 1, D = 1, Q = Q, S = 12)$AICc
#             print(paste("d2p10: {", " q: ", q, "Q: ", Q, "}"))
#         })
#     }
# }
# write.csv(aiccs, "d2p10_aicc.csv", col.names = TRUE, row.names = TRUE)

read.csv("d2p10_aicc.csv", header = TRUE)
```

```
##   X    X1    X2    X3    X4    X5    X6
## 1 1 8.708 8.668 8.675 8.674 8.683 8.689
## 2 2 8.708 8.667 8.678 8.678 8.688 8.695
## 3 3 8.715 8.676 8.691 8.687 8.696 8.705
## 4 4 8.723 8.675 8.701 8.698 8.708 8.716
## 5 5 8.716 8.745 8.692 8.689 8.718 8.728
## 6 6 8.727 8.751 8.736 8.701 8.708 8.716
```

Based on the AICc Values, we find out the following three models are the best models to fit.

$$SARIMA(p=7,d=1,q=1)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=7,d=1,q=2)(P=1,D=1,Q=2)_{S=12}$$
$$SARIMA(p=7,d=1,q=3)(P=1,D=1,Q=2)_{S=12}$$

Model Diagnostics:

$SARIMA(p=7, d=1, q=1)(P=1, D=1, Q=2)_{S=12}$:

```r
# fit the first model
fit.1 = sarima(data.train, p=7, d=1, q=1, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.1
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##         ar1     ar2    ar3     ar4     ar5     ar6     ar7     ma1   sar1
##       0.232  -0.042  0.015  -0.032  -0.066  -0.068  -0.134  -0.862  0.729
## s.e.  0.089   0.081  0.077   0.077   0.080   0.083   0.081   0.067  0.107
##         sma1   sma2
##       -1.922  1.000
## s.e.   0.298  0.309
##
## sigma^2 estimated as 209:  log likelihood = -855,  aic = 1734
##
## $degrees_of_freedom
## [1] 192
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.2317 0.0888   2.6081  0.0098
## ar2   -0.0425 0.0814  -0.5217  0.6025
## ar3    0.0155 0.0772   0.2007  0.8411
## ar4   -0.0317 0.0768  -0.4135  0.6797
## ar5   -0.0664 0.0805  -0.8244  0.4108
## ar6   -0.0679 0.0834  -0.8141  0.4166
## ar7   -0.1345 0.0814  -1.6523  0.1001
## ma1   -0.8622 0.0674 -12.7836  0.0000
## sar1   0.7294 0.1065   6.8473  0.0000
## sma1  -1.9217 0.2984  -6.4390  0.0000
## sma2   1.0000 0.3094   3.2320  0.0014
##
## $AIC
## [1] 8.542
##
## $AICc
## [1] 8.548
##
## $BIC
## [1] 8.738
```

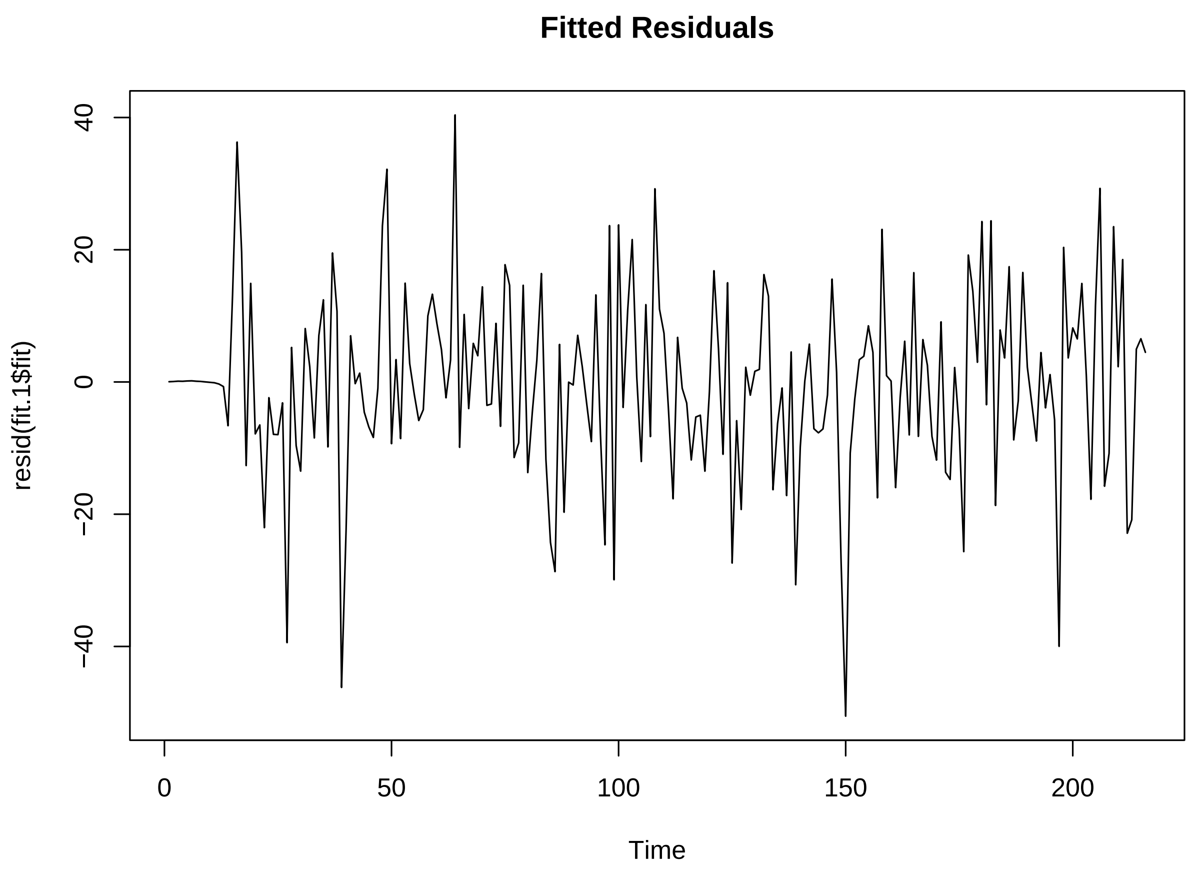
```
# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.1$fit), type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid(fit.1$fit)
## X-squared = 0.032, df = 1, p-value = 0.9
```

```
shapiro.test(resid(fit.1$fit))
```

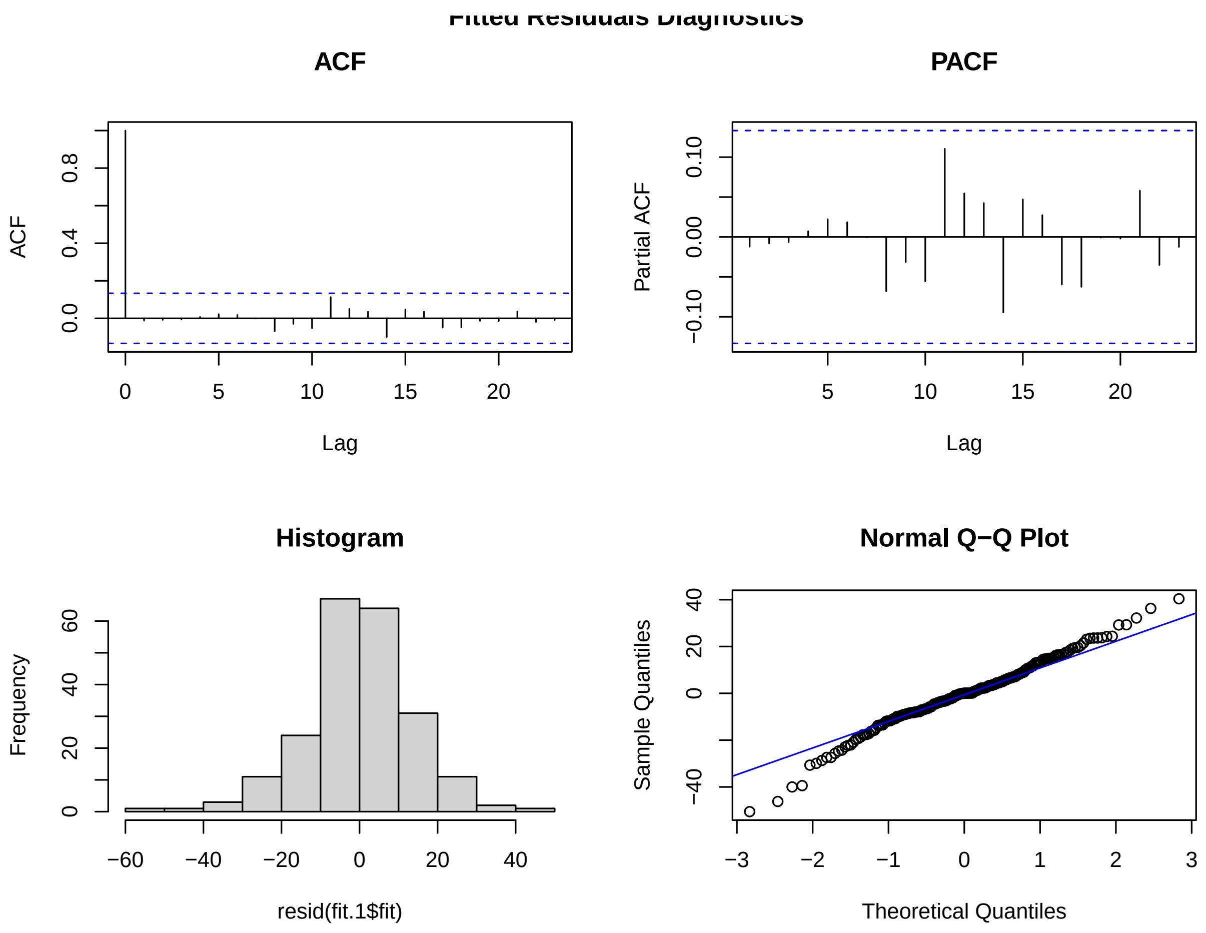
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit.1$fit)
## W = 0.98, p-value = 0.01
```

```
ts.plot(resid(fit.1$fit), main = "Fitted Residuals")
```



```
par(mfrow=c(2,2))
acf(resid(fit.1$fit), main="ACF")
```

```r
pacf(resid(fit.1$fit), main="PACF")
hist(resid(fit.1$fit), main="Histogram")
qqnorm(resid(fit.1$fit))
qqline(resid(fit.1$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
```



We can see for $SARIMA(p = 7, d = 1, q = 2)(P = 1, D = 1, Q = 2)_{S=12}$ model, the Ljung-Box independence test does pass based on its p-value, the Shapiro-Wilk normality test does pass based on its p-value. Therefore, we will leave this model as a great backup model.

$SARIMA(p=7, d=1, q=2)(P=1, D=1, Q=2)_{S=12}$:

```r
# fit the second model
fit.2 = sarima(data.train, p=7, d=1, q=2, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.2
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2    ar3      ar4      ar5      ar6      ar7      ma1    ma2
##        0.731  -0.227  0.000  -0.074  -0.078  -0.055  -0.131  -1.362  0.497
## s.e.   0.215   0.123  0.093   0.093   0.093   0.092   0.094   0.203  0.194
##         sar1     sma1   sma2
##        0.679  -1.914  0.999
## s.e.   0.114   0.411  0.427
##
## sigma^2 estimated as 206:  log likelihood = -853.9,  aic = 1734
##
## $degrees_of_freedom
## [1] 191
##
## $ttable
##      Estimate     SE t.value p.value
## ar1    0.7308 0.2145  3.4062  0.0008
## ar2   -0.2269 0.1230 -1.8447  0.0666
## ar3    0.0002 0.0929  0.0020  0.9984
## ar4   -0.0744 0.0935 -0.7963  0.4268
## ar5   -0.0781 0.0933 -0.8371  0.4036
## ar6   -0.0546 0.0925 -0.5908  0.5553
## ar7   -0.1308 0.0939 -1.3931  0.1652
## ma1   -1.3619 0.2027 -6.7177  0.0000
## ma2    0.4975 0.1938  2.5669  0.0110
## sar1   0.6792 0.1141  5.9523  0.0000
## sma1  -1.9144 0.4106 -4.6631  0.0000
## sma2   0.9986 0.4273  2.3371  0.0205
##
## $AIC
## [1] 8.54
##
## $AICc
## [1] 8.549
##
## $BIC
## [1] 8.753
```

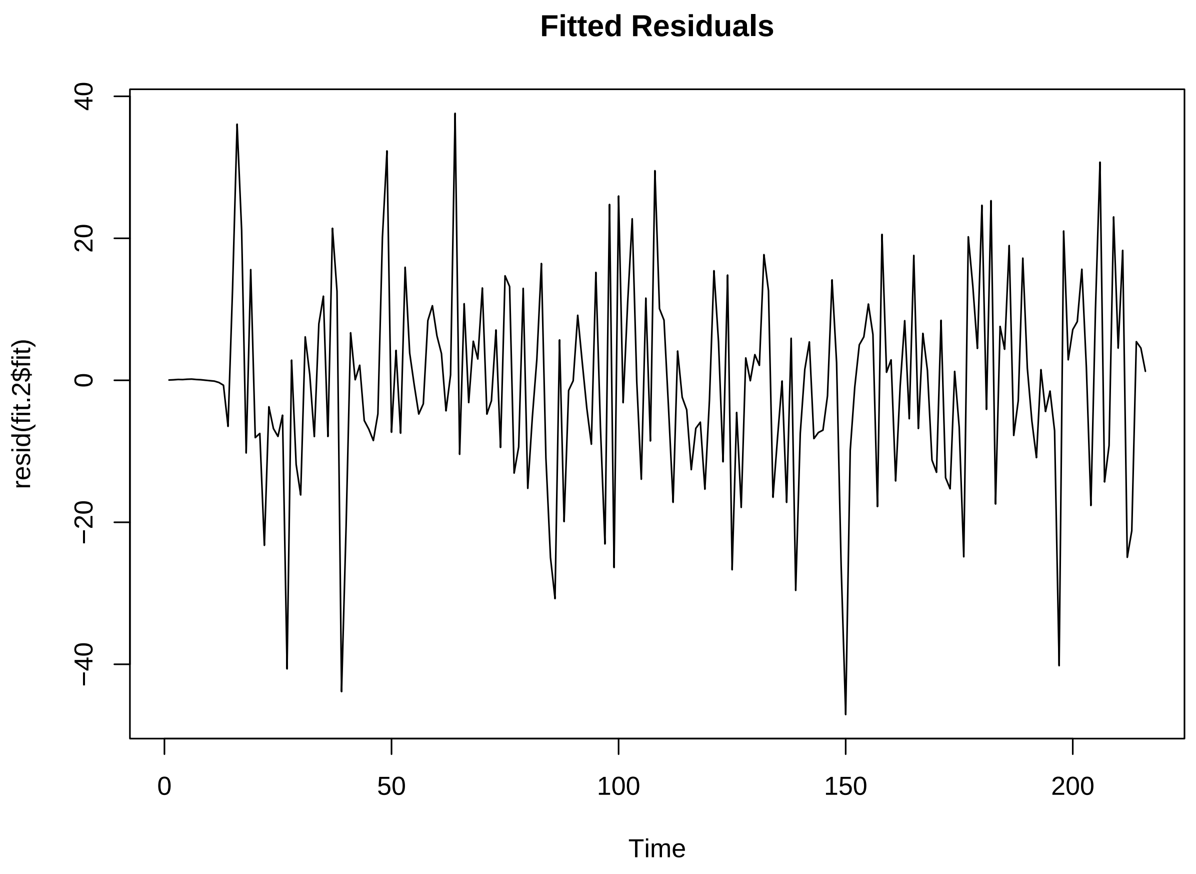
```r
# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.2$fit), type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid(fit.2$fit)
## X-squared = 0.0083, df = 1, p-value = 0.9
```

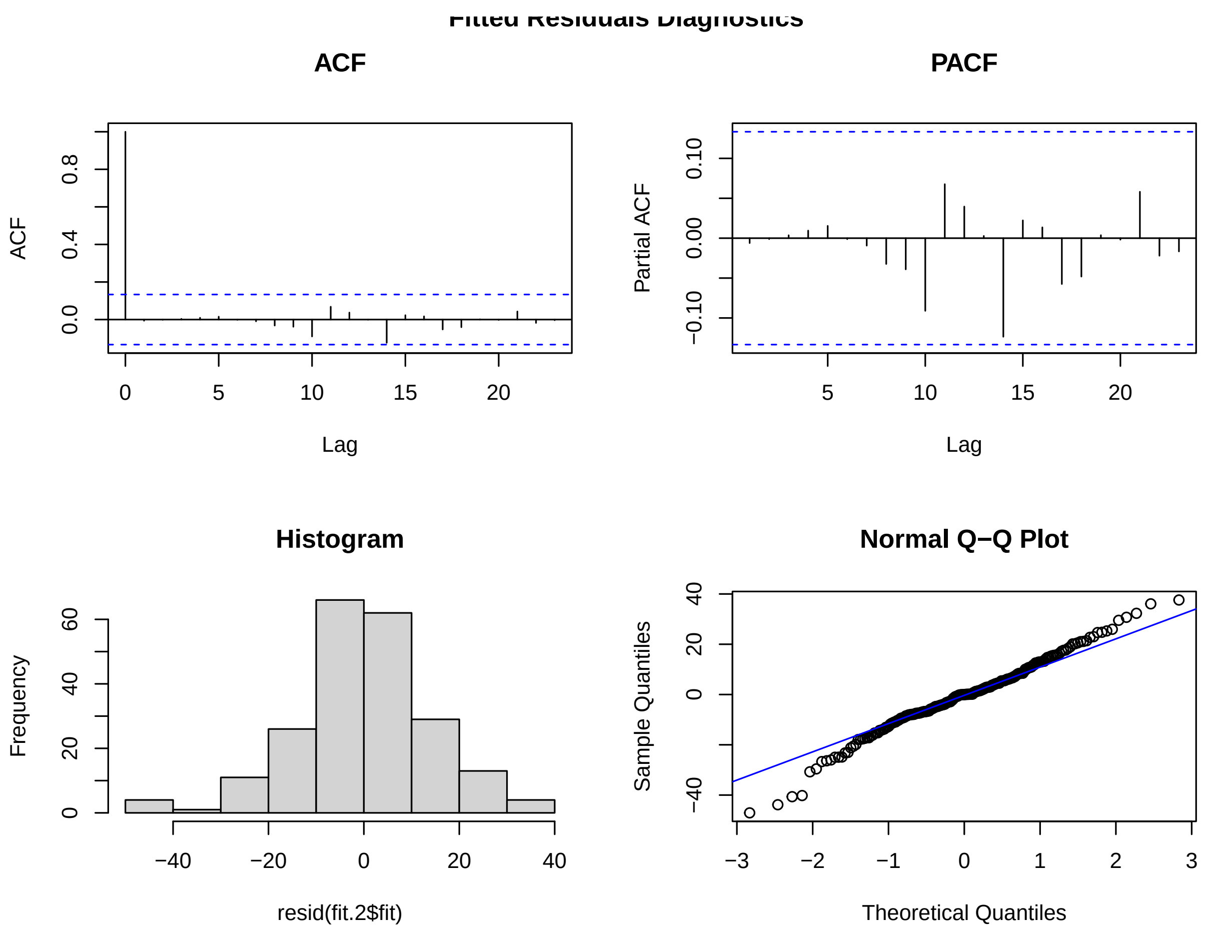
```
shapiro.test(resid(fit.2$fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit.2$fit)
## W = 0.99, p-value = 0.03
```

```
ts.plot(resid(fit.2$fit), main = "Fitted Residuals")
```



```
par(mfrow=c(2,2))
acf(resid(fit.2$fit), main="ACF")
pacf(resid(fit.2$fit), main="PACF")
hist(resid(fit.2$fit), main="Histogram")
qqnorm(resid(fit.2$fit))
qqline(resid(fit.2$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
```



We can see for $SARIMA(p = 7, d = 1, q = 2)(P = 1, D = 1, Q = 2)_{S=12}$ model, same as previous model, the Ljung-Box independence test does pass based on its p-value, the Shapiro-Wilk normality test does pass based on its p-value. Therefore, we will leave this model as a great backup model.

$SARIMA(p = 7, d = 1, q = 3)(P = 1, D = 1, Q = 2)_{S=12}$:

```r
# fit the third model
fit.3 = sarima(data.train, p=7, d=1, q=3, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.3
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ar5      ar6      ar7      ma1     ma2
##        1.747  -1.270  0.296  -0.085  -0.003  -0.022  -0.014  -2.388  2.194
## s.e.  0.194   0.238  0.182   0.170   0.171   0.150   0.090   0.184  0.306
##          ma3    sar1    sma1   sma2
##       -0.758  0.636  -1.911  1.000
## s.e.   0.149  0.104   0.243  0.253
##
## sigma^2 estimated as 202:  log likelihood = -852.2,  aic = 1732
##
## $degrees_of_freedom
## [1] 190
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    1.7470 0.1944   8.9868  0.0000
## ar2   -1.2701 0.2382  -5.3317  0.0000
## ar3    0.2965 0.1822   1.6274  0.1053
## ar4   -0.0852 0.1696  -0.5022  0.6161
## ar5   -0.0030 0.1710  -0.0176  0.9860
## ar6   -0.0222 0.1503  -0.1479  0.8825
## ar7   -0.0137 0.0904  -0.1512  0.8800
## ma1   -2.3880 0.1844 -12.9515  0.0000
## ma2    2.1940 0.3065   7.1587  0.0000
## ma3   -0.7582 0.1489  -5.0923  0.0000
## sar1   0.6364 0.1043   6.1001  0.0000
## sma1  -1.9113 0.2428  -7.8712  0.0000
## sma2   1.0000 0.2528   3.9554  0.0001
##
## $AIC
## [1] 8.534
##
## $AICc
## [1] 8.543
##
## $BIC
## [1] 8.762
```

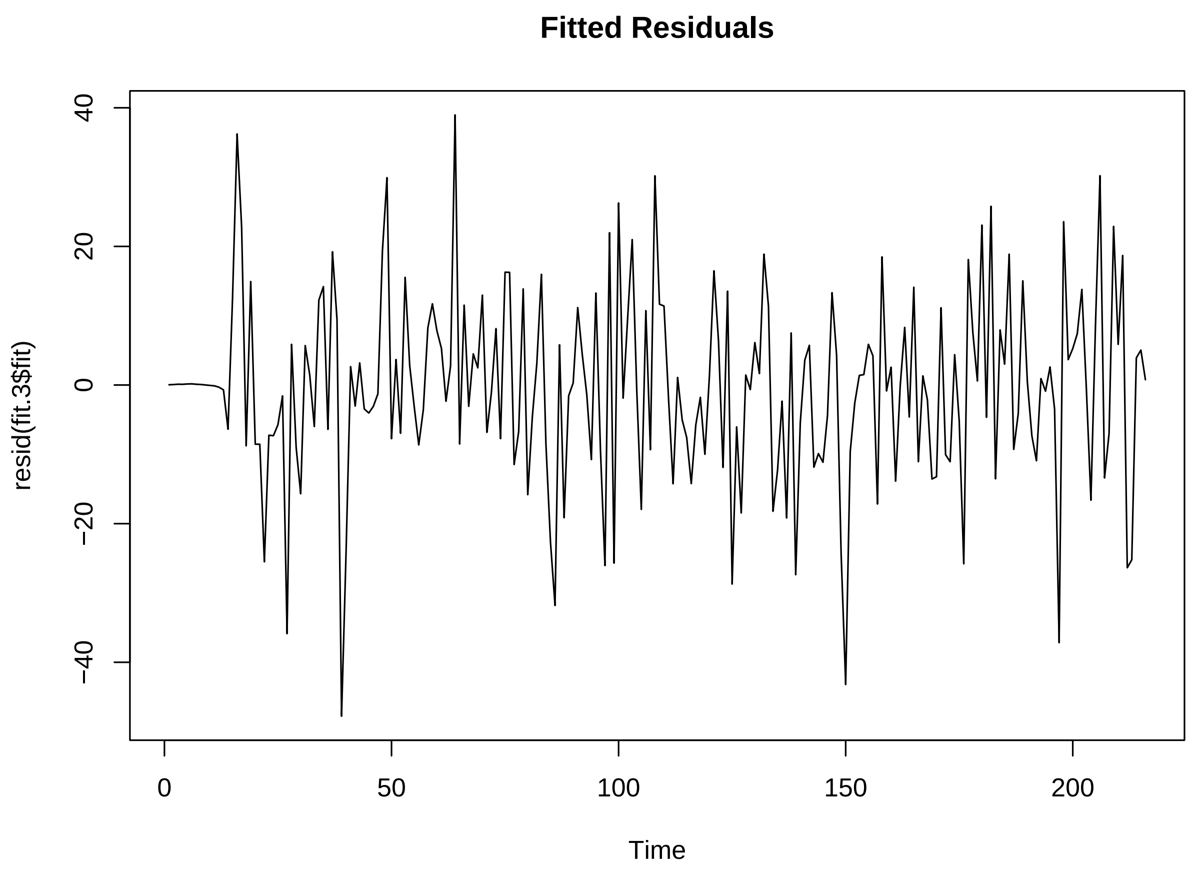
```
# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.3$fit), type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid(fit.3$fit)
## X-squared = 0.00014, df = 1, p-value = 1
```

```
shapiro.test(resid(fit.3$fit))
```

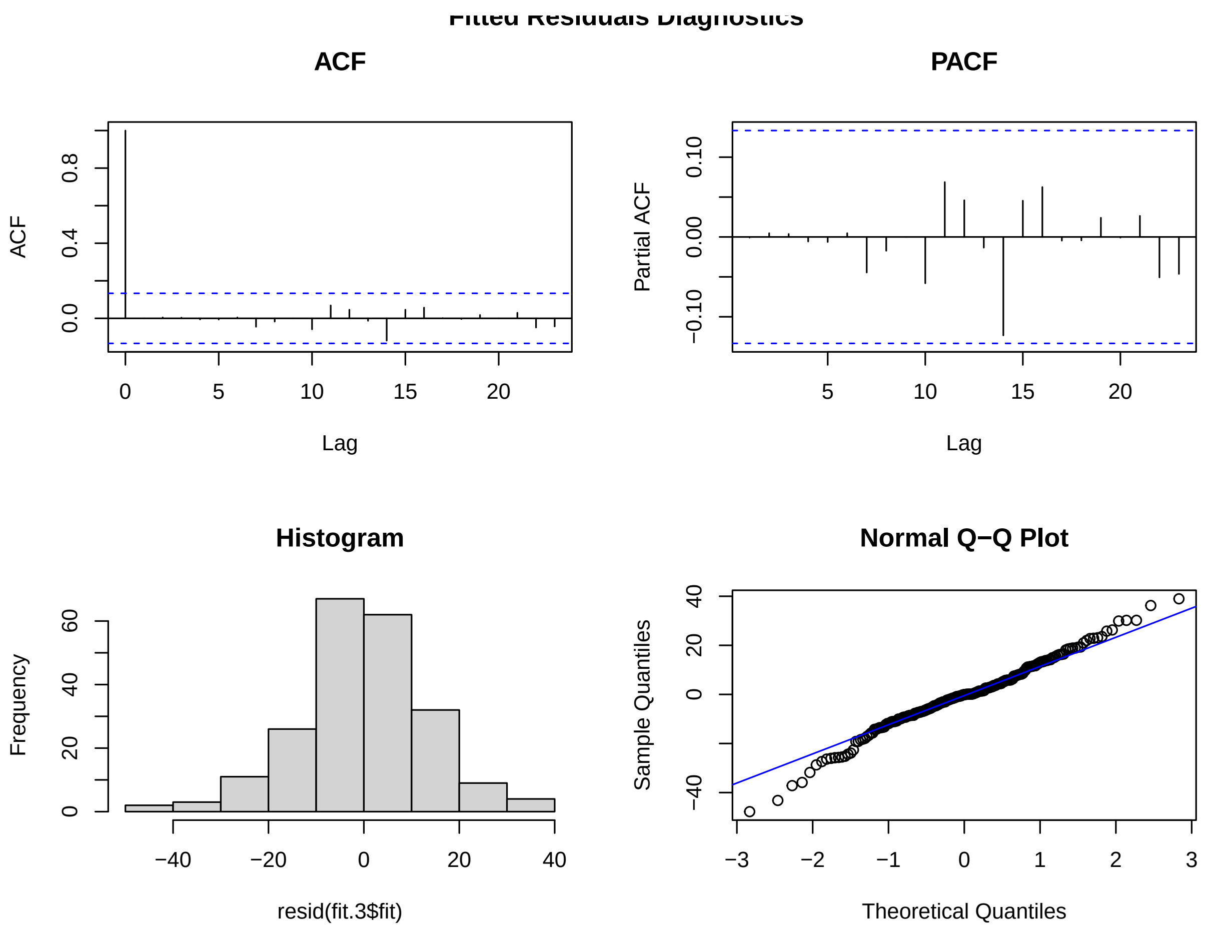
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit.3$fit)
## W = 0.99, p-value = 0.03
```

```
ts.plot(resid(fit.3$fit), main = "Fitted Residuals")
```



```
par(mfrow=c(2,2))
acf(resid(fit.3$fit), main="ACF")
```

```r
pacf(resid(fit.3$fit), main="PACF")
hist(resid(fit.3$fit), main="Histogram")
qqnorm(resid(fit.3$fit))
qqline(resid(fit.3$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
```



We can see for $SARIMA(p = 7, d = 1, q = 3)(P = 1, D = 1, Q = 2)_{S=12}$ model, same as previous model, the Ljung-Box independence test does pass based on its p-value, the Shapiro-Wilk normality test does pass based on its p-value. Therefore, we will leave this model as a backup model.

The Model for Forecasting:

Since all three models passed the independence and normality test, we will use the model with best test statistics. Therefore, we will use $SARIMA(p = 7, d = 1, q = 3)(P = 1, D = 1, Q = 2)_{S=12}$ for the model forecast.

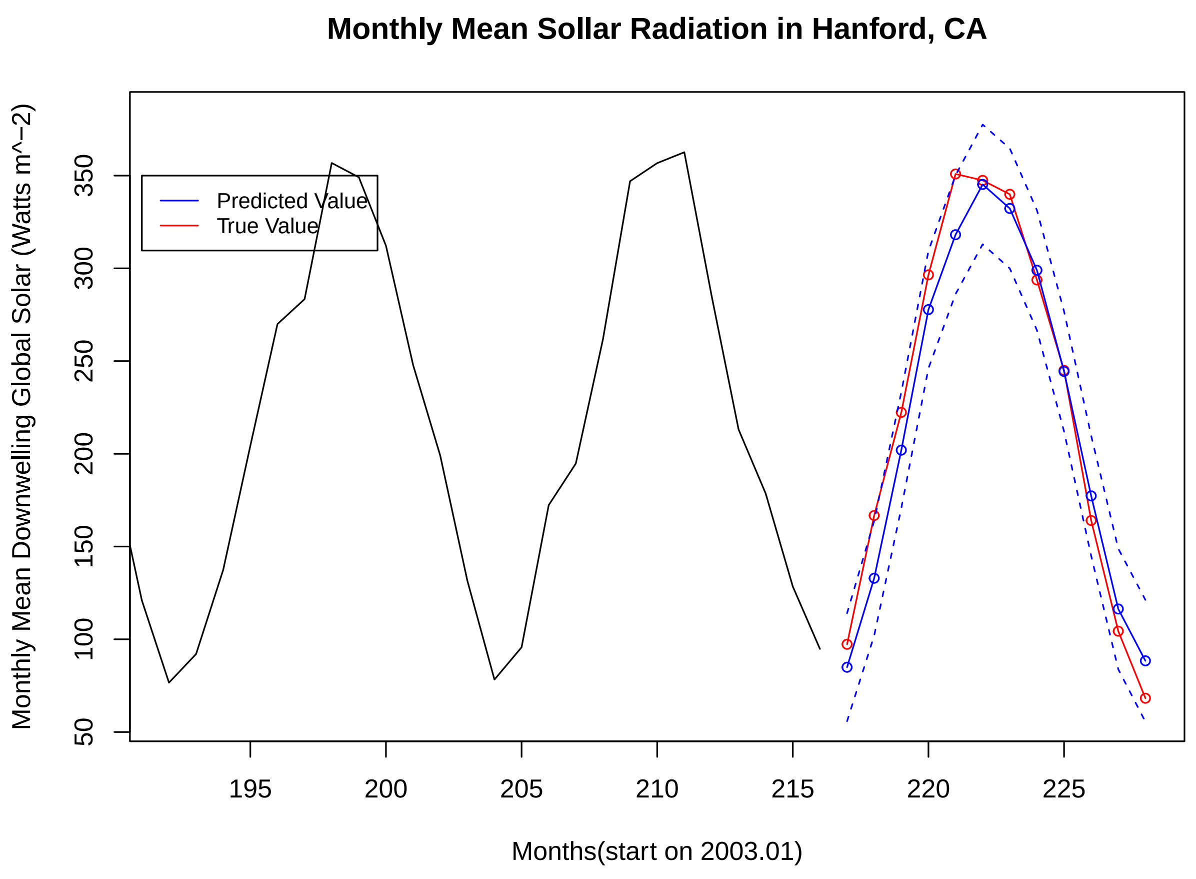
Model Forecast:

```r
# model forecast
data.pred = predict(fit.3$fit, n.ahead=12)
data.frame(pred = data.pred$pred, true = data.test)
```

```
##      pred   true
## 1   84.93  97.32
## 2  132.94 166.74
## 3  201.99 222.34
## 4  277.75 296.49
## 5  318.13 350.88
## 6  345.24 347.44
## 7  332.28 339.93
## 8  298.99 293.75
## 9  244.41 245.00
## 10 177.33 164.05
## 11 116.34 104.38
## 12  88.39  68.23
```

```r
# plot the forecasted values
ts.plot(data.train, ylab = "Monthly Mean Downwelling Global Solar (Watts m^-2)",
                    xlab = "Months(start on 2003.01)",xlim=c(192,228))
points(ts(data.test,start=c(217,1)), cex=0.8, pch=1, col="red")
lines(ts(data.test,start=c(217,1)), cex=0.8, pch=1, col="red")
points(data.pred$pred, cex=0.8, col="blue")
lines(data.pred$pred, cex=0.8, col="blue")
lines(data.pred$pred+1.96*data.pred$se, lty=2, col="blue")
lines(data.pred$pred-1.96*data.pred$se, lty=2, col="blue")

legend(191,350,legend=c("Predicted Value", "True Value"),col=c("blue","red"),lty=1,cex=0.8)
title("Monthly Mean Sollar Radiation in Hanford, CA")
```



Conclusion

We use year 2021 as our test set, and predict 2021 based on our model. We can see the $SARIMA(p = 7, d = 1, q = 3)(P = 1, D = 1, Q = 2)_{S=12}$ model fits really well. All the true values are within the predicted confidence interval. Therefore, this model is a valid model to predict the monthly mean sollar radiation in Hanford, CA.

References

Hanford, California, United States (HNX) Continuous in-situ measurements of solar radiation.Earth System Research Laboratory, Global Monitoring Laboratory, URL: https://gml.noaa.gov/aftp/data/radiation/solrad/hnx/
Rob J Hyndman, Box–Jenkins modelling, URL: http://robjhyndman.com/papers/BoxJenkins.pdf

Appendix

```r
# knit options
knitr::opts_chunk$set(fig.width=8, fig.height=6, message = F, warning = F)
options(digits = 4)

# packages
library(dplyr)
library(aTSA)
library(astsa)
setwd("hnx_solar")
# read_list <- list.files(pattern = "\\.dat")

# merge <- read.table(file = "hnx03001.dat", skip = 2, header = FALSE)

# colnames(merge) = c("year","jday","month","day","hour","min",
#                     "dt","zen","dw_psp","qc_dwpsp","direct","qc_direct",
#                     "diffuse","qc_diffuse","uvb","qc_uvb",
#                     "uvb_temp","qc_uvb_temp","std_dw_psp",
#                     "std_direct","std_diffuse","std_uvb")

# merge[merge == -9999.9] <- NA

# merge <- data.frame(merge["year"], merge["month"], merge["day"], merge["dw_psp"])

# for (file in read_list[-which(read_list == "hnx03001.dat")]) {
#     print(paste("processing:", file))
#     new = read.table(file, skip = 2)
#     colnames(new) = c("year","jday","month","day","hour","min",
#                       "dt","zen","dw_psp","qc_dwpsp","direct","qc_direct",
#                       "diffuse","qc_diffuse","uvb","qc_uvb",
#                       "uvb_temp","qc_uvb_temp","std_dw_psp",
#                       "std_direct","std_diffuse","std_uvb")

#     new[new == -9999.9] <- NA
#     new <- data.frame(new["year"], new["month"], new["day"], new["dw_psp"])

#     merge = rbind(merge, new)
# }

# data = merge %>% group_by(year, month) %>%
#                 summarize(monthly_dw_psp = mean(dw_psp, na.rm = TRUE))

# write.csv(data, file = "hnx_solar_monthly.csv", col.names = TRUE)

# read from merged monthly data
data = read.csv("hnx_solar_monthly.csv")$monthly_dw_psp
data.train = data[1:216]
data.test = data[217:228]
# plot the original data
ts.plot(data.train, ylab = "Monthly Mean Downwelling Global Solar (Watts m^-2)",
                 xlab = "Months(start on 2003.01)")
title("Monthly Mean Sollar Radiation in Hanford, CA")
```

```r
par(mfrow=c(1,2))
acf(data.train, main = "ACF Plot")
pacf(data.train, main = "PACF Plot")
adf.test(data)
# take first difference
data.diff = diff(data.train)
ts.plot(data.diff)
title("TS Plot After First Difference")
par(mfrow=c(1,2))
acf(data.diff, main = "ACF Plot After First Difference")
pacf(data.diff, main = "PACF Plot After First Difference")
# take second difference
data.diff2 = diff(data.diff)
ts.plot(data.diff2)
title("TS Plot After Second Difference")
par(mfrow=c(1,2))
acf(data.diff2, main = "ACF Plot After Second Difference")
pacf(data.diff2, main = "PACF Plot After Second Difference")
# aiccs <- matrix(NA, nr =6, nc = 6)
# dimnames(aiccs) = list(q = c(1:6), Q = c(1:6))
setwd("hnx_solar")

# Model Estimation for d=1, p=7
# for (q in 1:6) {
#     for (Q in 1:6) {
#         try({
#             aiccs[q,Q] = sarima(xdata = data.train,
#                             p = 7, d = 1, q = q,
#                             P = 1, D = 1, Q = Q, S = 12)$AICc
#             print(paste("d1p7: {", " q: ", q, "Q: ", Q, "}"))
#         })
#     }
# }
# write.csv(aiccs, "d1p7_aicc.csv", col.names = TRUE, row.names = TRUE)

read.csv("d1p7_aicc.csv", header = TRUE)
setwd("hnx_solar")

# Model Estimation for d=2, p=10
# for (q in c(1:6)) {
#     for (Q in c(1:6)) {
#         try({
#             aiccs[q,Q] = sarima(xdata = data.train,
#                             p = 10, d = 2, q = q,
#                             P = 1, D = 1, Q = Q, S = 12)$AICc
#             print(paste("d2p10: {", " q: ", q, "Q: ", Q, "}"))
#         })
#     }
# }
# write.csv(aiccs, "d2p10_aicc.csv", col.names = TRUE, row.names = TRUE)

read.csv("d2p10_aicc.csv", header = TRUE)
# fit the first model
```

```r
fit.1 = sarima(data.train, p=7, d=1, q=1, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.1

# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.1$fit), type="Ljung-Box")
shapiro.test(resid(fit.1$fit))

ts.plot(resid(fit.1$fit), main = "Fitted Residuals")
par(mfrow=c(2,2))
acf(resid(fit.1$fit), main="ACF")
pacf(resid(fit.1$fit), main="PACF")
hist(resid(fit.1$fit), main="Histogram")
qqnorm(resid(fit.1$fit))
qqline(resid(fit.1$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
# fit the second model
fit.2 = sarima(data.train, p=7, d=1, q=2, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.2

# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.2$fit), type="Ljung-Box")
shapiro.test(resid(fit.2$fit))

ts.plot(resid(fit.2$fit), main = "Fitted Residuals")
par(mfrow=c(2,2))
acf(resid(fit.2$fit), main="ACF")
pacf(resid(fit.2$fit), main="PACF")
hist(resid(fit.2$fit), main="Histogram")
qqnorm(resid(fit.2$fit))
qqline(resid(fit.2$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
# fit the third model
fit.3 = sarima(data.train, p=7, d=1, q=3, P=1, D=1, Q=2, S=12, detail=FALSE)
fit.3

# Ljung-Box and Shapiro-Wilk test
Box.test(resid(fit.3$fit), type="Ljung-Box")
shapiro.test(resid(fit.3$fit))

ts.plot(resid(fit.3$fit), main = "Fitted Residuals")
par(mfrow=c(2,2))
acf(resid(fit.3$fit), main="ACF")
pacf(resid(fit.3$fit), main="PACF")
hist(resid(fit.3$fit), main="Histogram")
qqnorm(resid(fit.3$fit))
qqline(resid(fit.3$fit), col="blue")
title("Fitted Residuals Diagnostics", outer=TRUE)
# model forecast
data.pred = predict(fit.3$fit, n.ahead=12)
data.frame(pred = data.pred$pred, true = data.test)

# plot the forecasted values
ts.plot(data.train, ylab = "Monthly Mean Downwelling Global Solar (Watts m^-2)",
                    xlab = "Months(start on 2003.01)",xlim=c(192,228))
points(ts(data.test,start=c(217,1)), cex=0.8, pch=1, col="red")
lines(ts(data.test,start=c(217,1)), cex=0.8, pch=1, col="red")
points(data.pred$pred, cex=0.8, col="blue")
lines(data.pred$pred, cex=0.8, col="blue")
lines(data.pred$pred+1.96*data.pred$se, lty=2, col="blue")
lines(data.pred$pred-1.96*data.pred$se, lty=2, col="blue")

legend(191,350,legend=c("Predicted Value", "True Value"),col=c("blue","red"),lty=1,cex=0.8)
title("Monthly Mean Sollar Radiation in Hanford, CA")
```