FINA3295 Final Project

Le Minh Khue

2025-03-15

```
dataset <-
read.csv("C:\\Users\\khuem\\Downloads\\CLMTEMP_HKO_MONTHLY(1).csv")
colnames(dataset) <- c('index', 'year', 'month', 'temp')
dataset$year_standardized <- (1947-1/12) + dataset$index/12
dataset$time <- 1:nrow(dataset)/12
head(dataset)
```

```
##   index year month     temp year_standardized       time
## 1     1 1947     1 16.46129          1947.000 0.08333333
## 2     2 1947     2 13.78214          1947.083 0.16666667
## 3     3 1947     3 17.18710          1947.167 0.25000000
## 4     4 1947     4 20.34333          1947.250 0.33333333
## 5     5 1947     5 24.95806          1947.333 0.41666667
## 6     6 1947     6 27.02667          1947.417 0.50000000
```

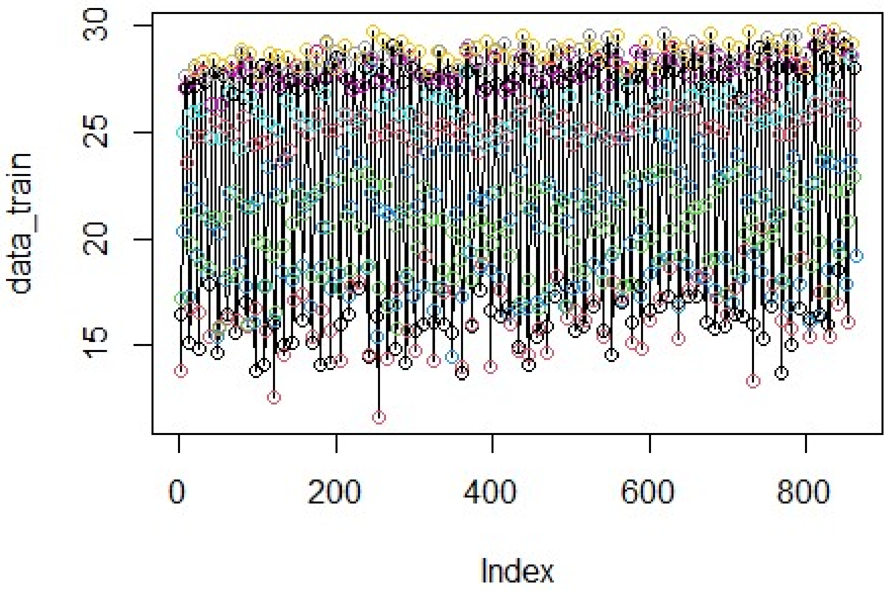
```
m <- length(dataset$year_standardized[dataset$year_standardized >= 2019 &
dataset$year_standardized <2025]) #length of data_test
n <- length(dataset$temp)
data_train <- dataset$temp[1:(n - m)]
data_test <- dataset$temp[(n - m + 1):n]
k <- length(data_train)

dataset_train <- data.frame(
    index = c(1:k),
    time = dataset$time[1:k],
    year_standardized = dataset$year_standardized[1:k],
    month = dataset$month[1:k],
    temp = dataset$temp[1:k],
    stringsAsFactors = FALSE)
head(dataset_train)
```

```
##   index       time year_standardized month     temp
## 1     1 0.08333333          1947.000     1 16.46129
## 2     2 0.16666667          1947.083     2 13.78214
## 3     3 0.25000000          1947.167     3 17.18710
## 4     4 0.33333333          1947.250     4 20.34333
## 5     5 0.41666667          1947.333     5 24.95806
## 6     6 0.50000000          1947.417     6 27.02667
```

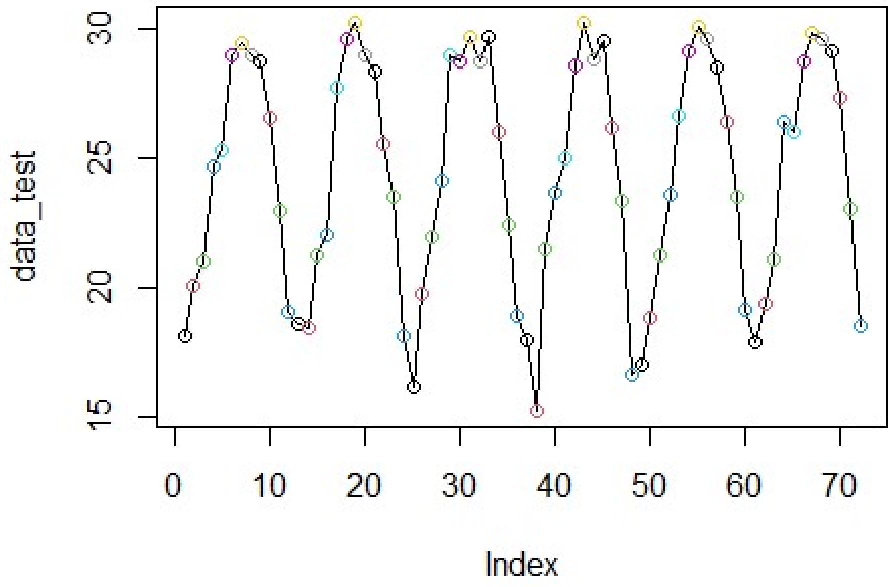
```
dataset_test <- data.frame(
    index = c((k+1):n),
    time = dataset$time[(k+1):n],
    year_standardized = dataset$year_standardized[(k+1):n],
    month = dataset$month[(k+1):n],
    temp = dataset$temp[(k+1):n],
    stringsAsFactors = FALSE)
head(dataset_test)
```

```
##   index     time year_standardized month     temp
## 1   865 72.08333          2019.000     1 18.13548
## 2   866 72.16667          2019.083     2 20.12143
## 3   867 72.25000          2019.167     3 21.03226
## 4   868 72.33333          2019.250     4 24.69667
## 5   869 72.41667          2019.333     5 25.32903
## 6   870 72.50000          2019.417     6 29.00000
```

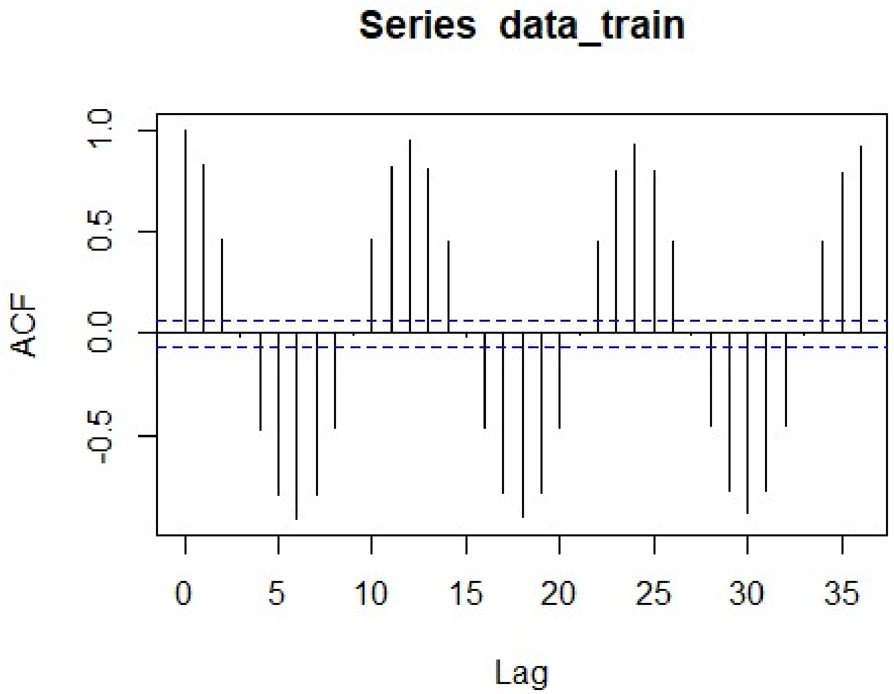
```
plot(data_train, type = 'l')
points(data_train, col = rep(1:12, length.out = length(data_train)))
```



```
plot(data_test, type = 'l')
points(data_test, col = rep(1:12, length.out = length(data_test)))
```



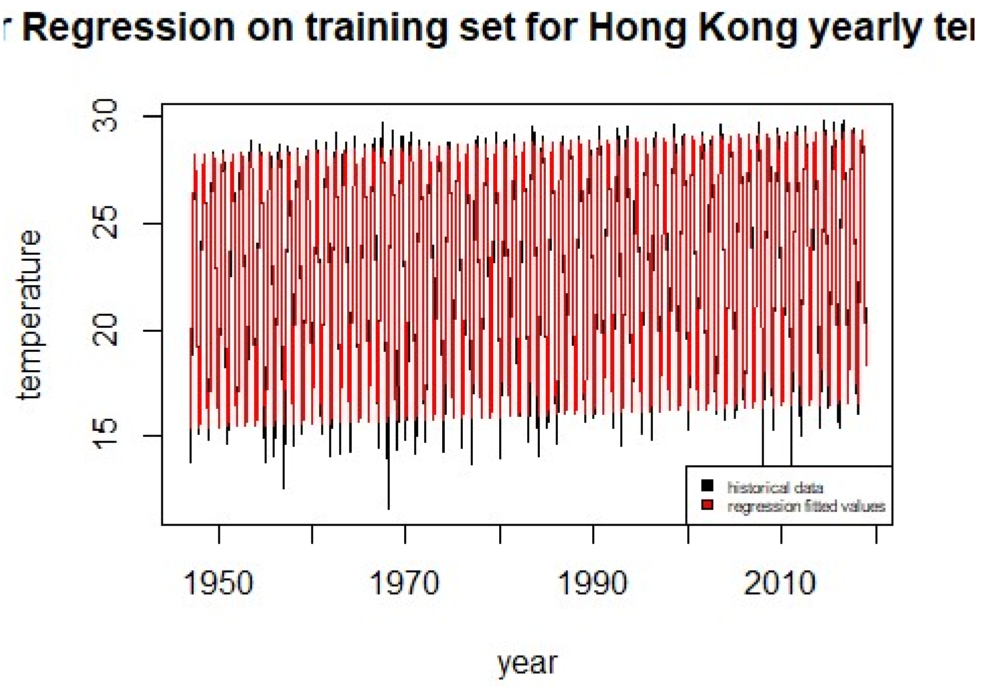
```r
acf(data_train, lag.max = 36)
```



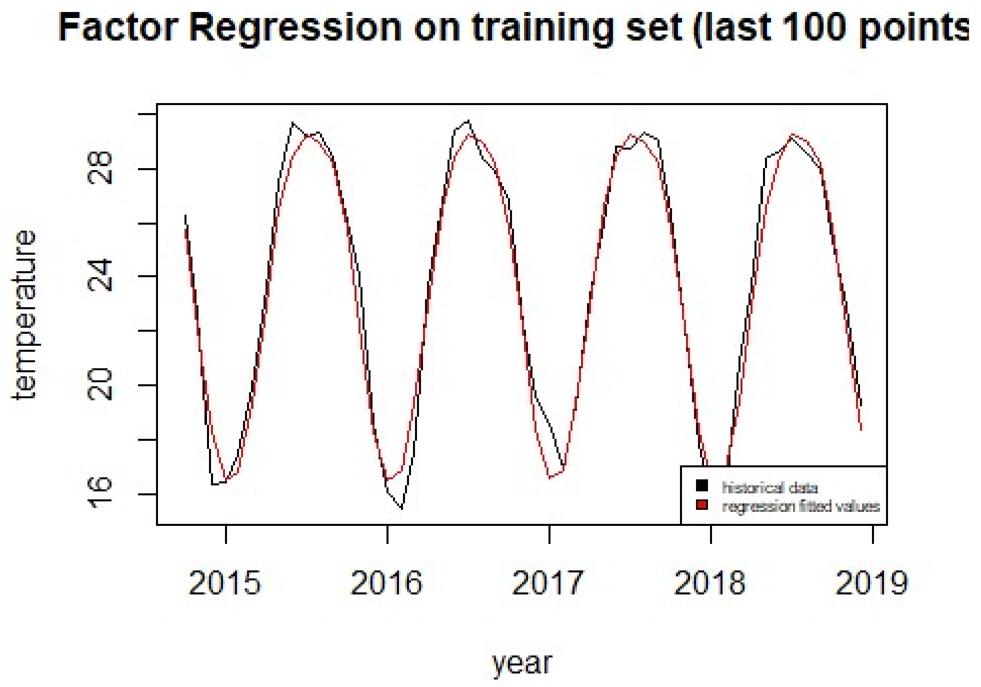
```r
mylm <- lm(temp~time+factor(month), data = dataset_train)
summary(mylm)
```

```
## 
## Call:
## lm(formula = temp ~ time + factor(month), data = dataset_train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.4170 -0.5230 -0.0027  0.5808  3.7774 
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      15.411675   0.127238 121.125   <2e-16 ***
## time              0.016040   0.001584  10.124   <2e-16 ***
## factor(month)2    0.324476   0.161313   2.011   0.0446 *  
## factor(month)3    2.821386   0.161314  17.490   <2e-16 ***
## factor(month)4    6.381053   0.161314  39.557   <2e-16 ***
## factor(month)5    9.998506   0.161314  61.982   <2e-16 ***
## factor(month)6   11.894722   0.161315  73.736   <2e-16 ***
## factor(month)7   12.722580   0.161315  78.868   <2e-16 ***
## factor(month)8   12.434998   0.161316  77.085   <2e-16 ***
## factor(month)9   11.664786   0.161317  72.310   <2e-16 ***
## factor(month)10   9.278068   0.161318  57.514   <2e-16 ***
## factor(month)11   5.619844   0.161319  34.837   <2e-16 ***
## factor(month)12   1.793406   0.161320  11.117   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9679 on 851 degrees of freedom
## Multiple R-squared:  0.9598, Adjusted R-squared:  0.9592 
## F-statistic:  1691 on 12 and 851 DF,  p-value: < 2.2e-16
```

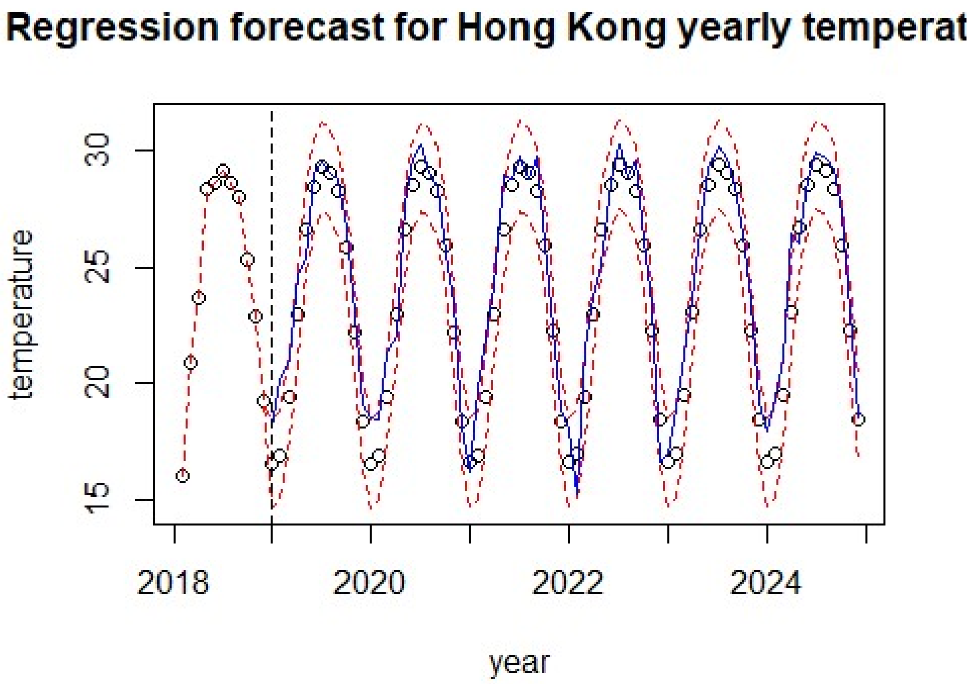
```r
plot(dataset_train$year_standardized, dataset_train$temp, type = 'l',lwd =
1.5, main = 'Factor Regression on training set for Hong Kong yearly
temperature', xlab = 'year', ylab = 'temperature')
lines(dataset_train$year_standardized, mylm$fitted.values, type = 'l', col =
'red')
legend('bottomright', legend=c("historical data", "regression fitted
values"), fill = c("black","red"), cex = 0.5)
```



```
plot(dataset_train$year_standardized[(k-50):k], dataset_train$temp[(k-50):k],
type = 'l', main = 'Factor Regression on training set (last 100 points)',
xlab = 'year', ylab = 'temperature')
lines(dataset_train$year_standardized[(k-50):k], mylm$fitted.values[(k-
50):k], type = 'l', col = 'red')
legend('bottomright', legend=c("historical data", "regression fitted
values"), fill = c("black","red"), cex = 0.5)
```



```r
mypreds1 <- data.frame(predict(mylm, newdata = dataset_test, interval =
'prediction'))
fore1 <- c(data_train[(k-10):k], mypreds1$fit)
foreupper1 <- c(data_train[(k-10):k], mypreds1$upr)
forelower1 <- c(data_train[(k-10):k], mypreds1$lwr)
plot(dataset$year_standardized[(k-10):(k+m)],fore1, ylim =
range(c(foreupper1, forelower1)),lwd = 1.5, main = 'Regression forecast for
Hong Kong yearly temperature', xlab = 'year', ylab = 'temperature')
lines(dataset$year_standardized[(k-10):(k+m)], foreupper1, lty = 2, col =
'red')
lines(dataset$year_standardized[(k-10):(k+m)], forelower1, lty = 2, col =
'red')
lines(dataset$year_standardized[(n - length(mypreds1$fit) + 1):n], data_test,
col = 'blue')
abline(v= 2019 , lty=2)
```

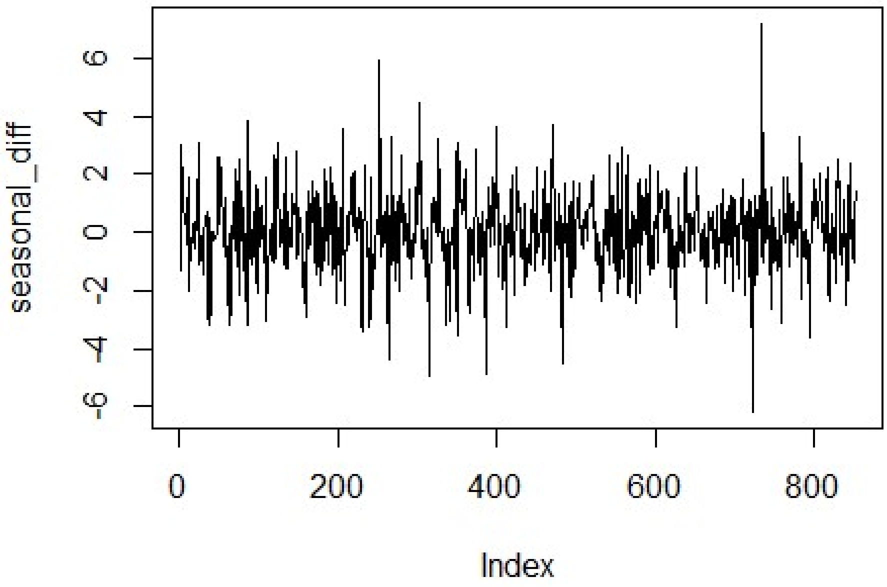


```
# performance metrics for factor regression
print('Performance metrics for factor regression')

## [1] "Performance metrics for factor regression"

mae <- mean(abs(data_test - mypreds1$fit))
mse <- mean((data_test - mypreds1$fit)^2)
rmse <- sqrt(mse)
mape <- mean(abs((data_test - mypreds1$fit) / data_test)) * 100
ss_res <- sum((data_test - mypreds1$fit)^2)
ss_tot <- sum((data_test - mean(data_test))^2)
r_squared <- 1 - (ss_res / ss_tot)

cat("MAE:", mae, "\n")

## MAE: 1.016779

cat("MSE:", mse, "\n")

## MSE: 1.652479

cat("RMSE:", rmse, "\n")

## RMSE: 1.285488

cat("MAPE:", mape, "%\n")

## MAPE: 4.544962 %
```

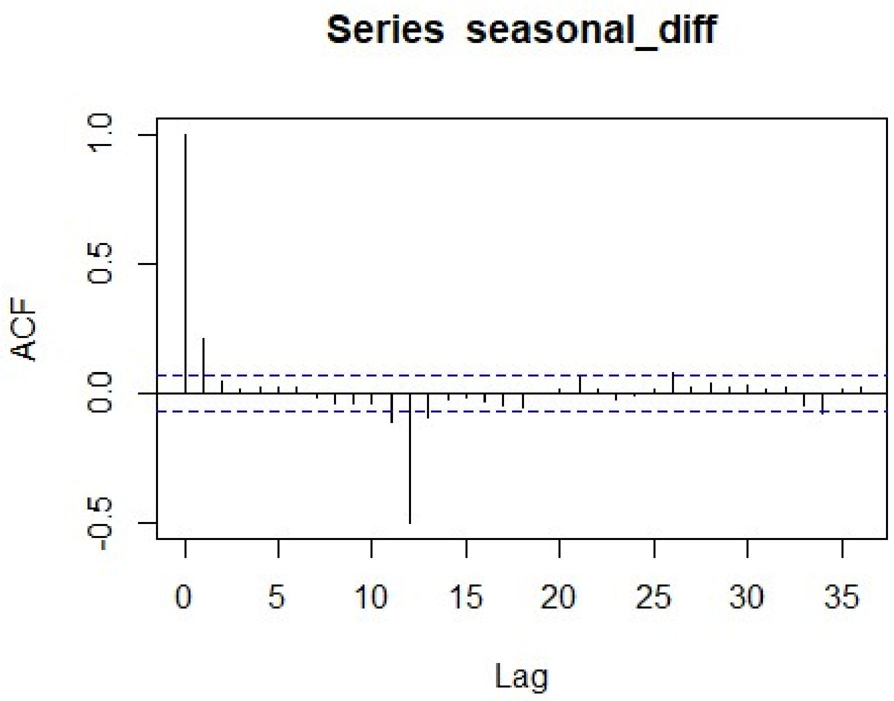
```
cat("R-squared:", r_squared, "\n")

## R-squared: 0.9171673
```

The data seems to follow a clear seasonal pattern, so I will take the seasonal difference

```
seasonal_diff <- diff(data_train, lag = 12)
plot(seasonal_diff, type = 'l')
```

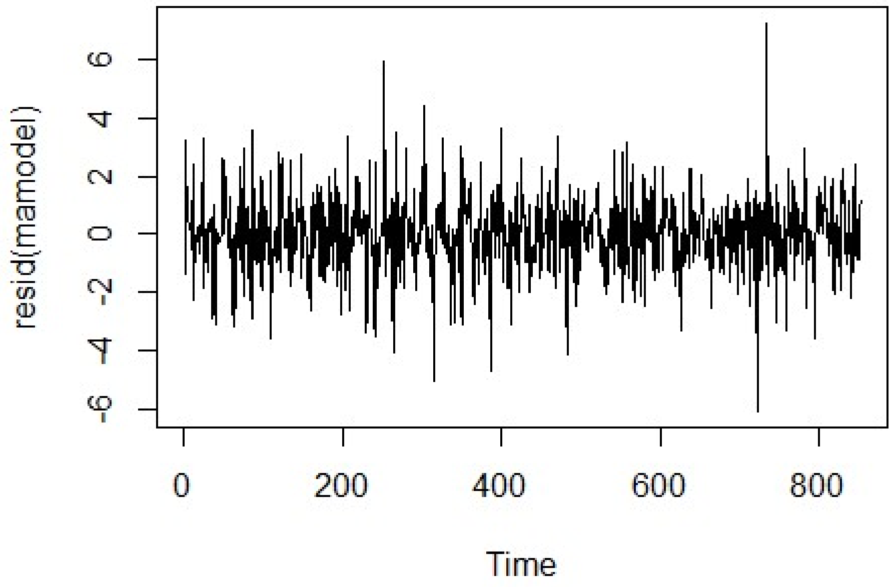


```
acf(seasonal_diff, lag.max = 36)
```

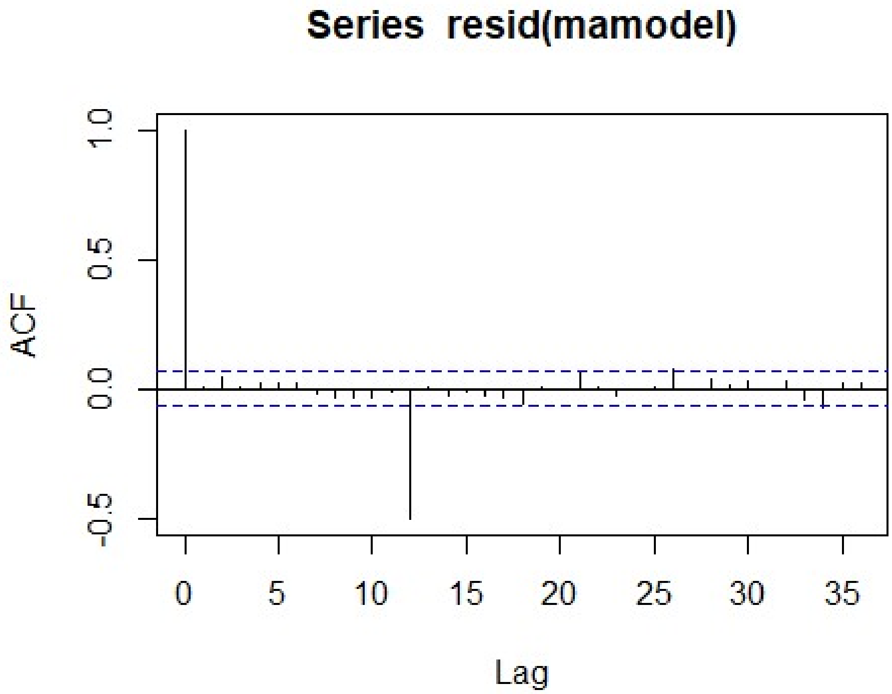


There is a clear spike at lag 12 but no clear spike after. Therefore, a MA model might be used in this case.

```
mamodel <- arima(seasonal_diff, order = c(0, 0, 1))
mamodel

##
## Call:
## arima(x = seasonal_diff, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.2011     0.0266
## s.e.  0.0322     0.0552
##
## sigma^2 estimated as 1.803:  log likelihood = -1460.06,  aic = 2926.12

plot(resid(mamodel), type = 'l')
```



```
acf(resid(mamodel), lag.max = 36)
```

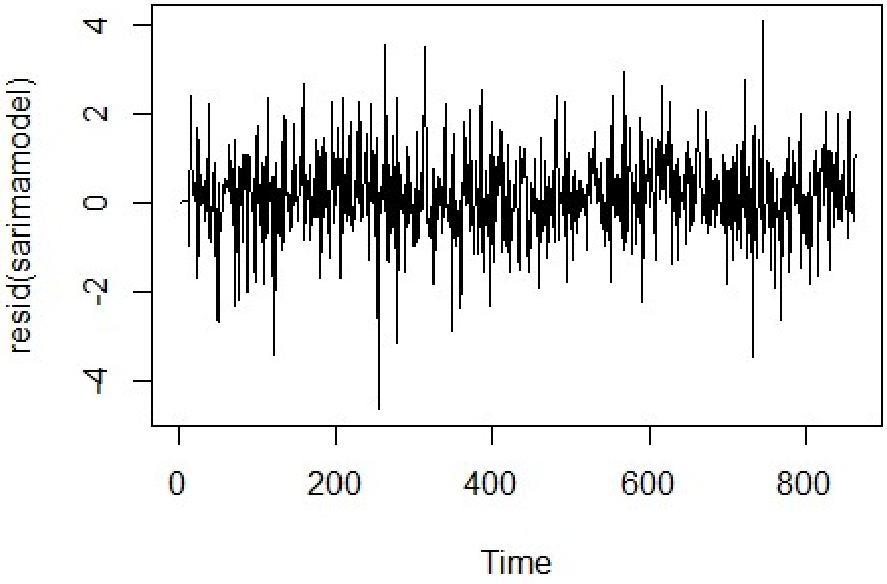


There is still a clear spike at lag 12. Suggesting a Seasonal MA model.

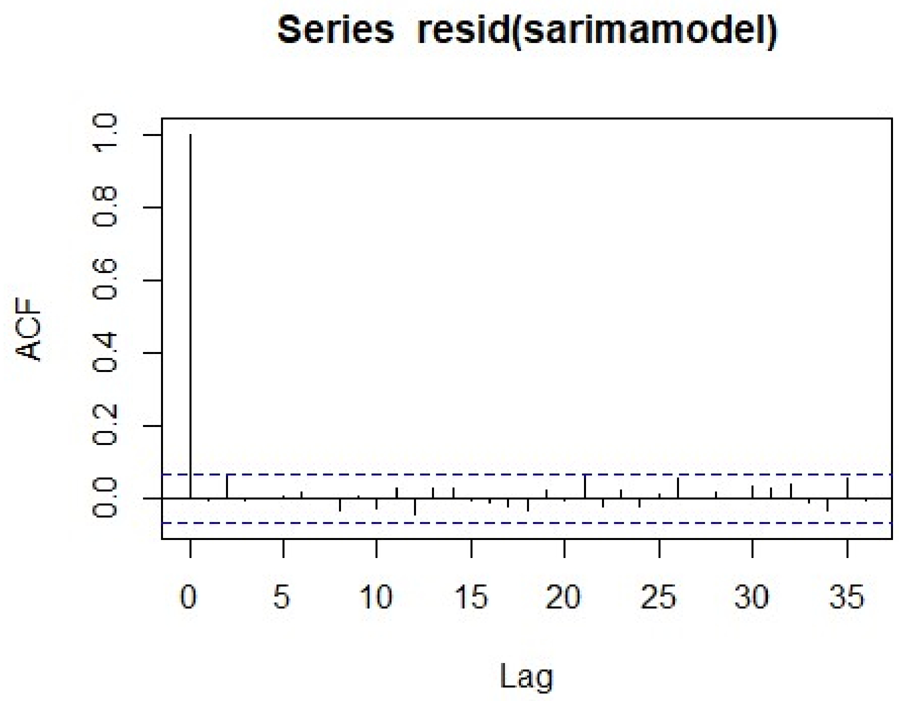
```
sarimamodel <- arima(data_train, order = c(0, 0, 1), seasonal = list(order =
c(0, 1, 1), period = 12))
sarimamodel
```

```
##
## Call:
## arima(x = data_train, order = c(0, 0, 1), seasonal = list(order = c(0, 1,
1),
##     period = 12))
##
## Coefficients:
##          ma1     sma1
##       0.2404  -0.9247
## s.e.  0.0316   0.0140
##
## sigma^2 estimated as 0.9516:  log likelihood = -1199.43,  aic = 2404.86
```

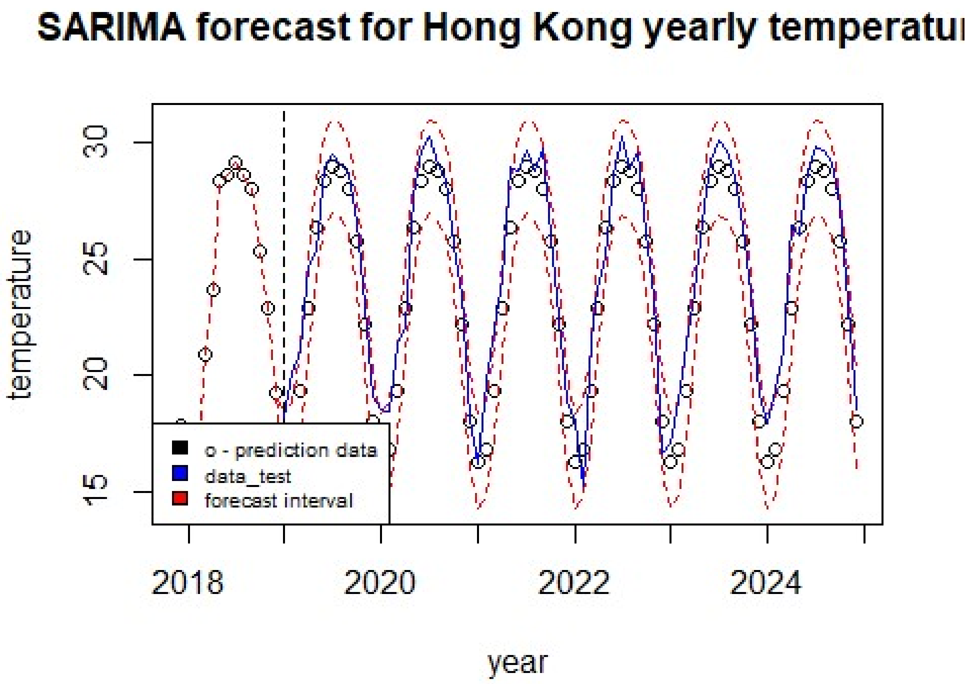
```
plot(resid(sarimamodel), type = 'l')
```



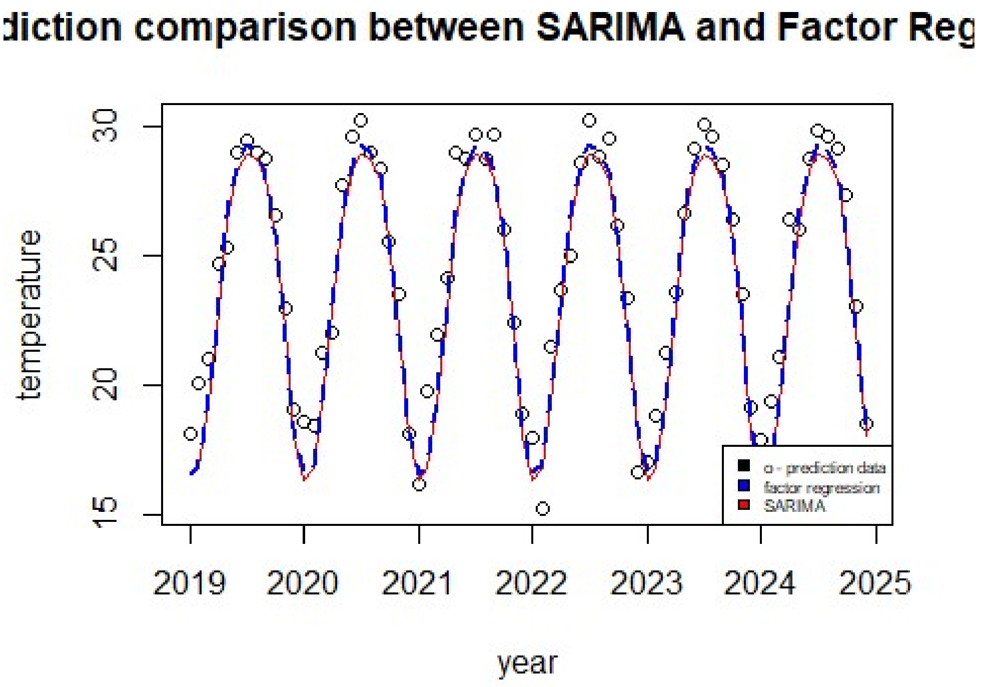
```
acf(resid(sarimamodel), lag.max = 36)
```



```r
mypreds <- predict(sarimamodel, n.ahead = m, se.fit = TRUE)
fore <- c(data_train[(k-12):k], mypreds$pred)
foreupper <- c(data_train[(k-12):k], mypreds$pred + 2*mypreds$se)
forelower <- c(data_train[(k-12):k], mypreds$pred - 2*mypreds$se)

plot(dataset$year_standardized[(k-12):(k+m)],fore, ylim = range(c(foreupper,
forelower)),lwd = 1.5, main = 'SARIMA forecast for Hong Kong yearly
temperature', xlab = 'year', ylab = 'temperature')
lines(dataset$year_standardized[(k-12):(k+m)], foreupper, lty = 2, col =
'red')
lines(dataset$year_standardized[(k-12):(k+m)], forelower, lty = 2, col =
'red')
lines(dataset$year_standardized[(n - length(mypreds$pred) + 1):n],
      data_test, col = 'blue')
abline(v= 2019 , lty=2)
legend('bottomleft', legend=c("o - prediction data", "data_test", "forecast
interval"), fill = c("black", "blue","red"), cex = 0.65)
```



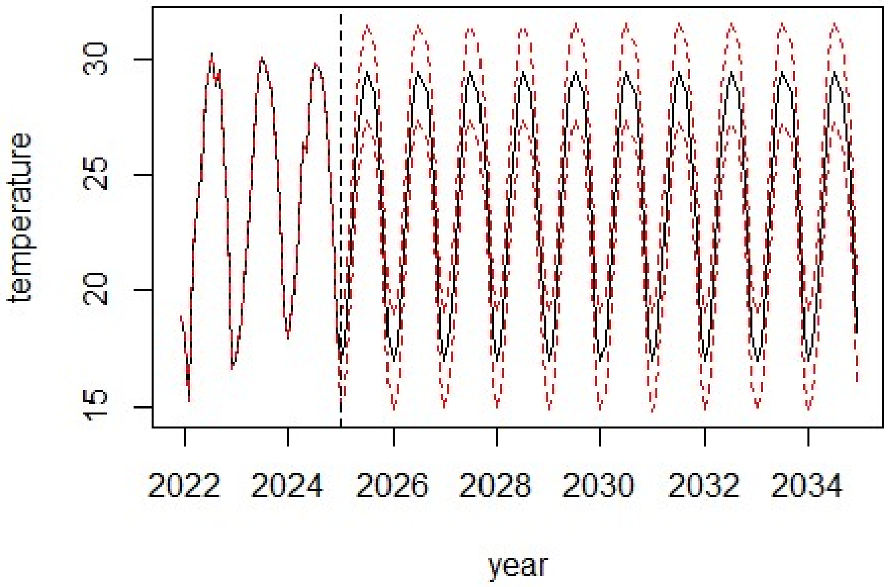
```
plot(dataset_test$year_standardized, dataset_test$temp, main = 'Prediction
comparison between SARIMA and Factor Regression', xlab = 'year', ylab =
'temperature')
lines(dataset_test$year_standardized, mypreds$pred, col = 'red')
lines(dataset_test$year_standardized, mypreds1$fit, lty = 2, col = 'blue',
lwd = 2)
legend('bottomright', legend=c("o - prediction data", "factor regression",
"SARIMA"), fill = c("black", "blue","red"), cex = 0.5)
```



```
# performance metrics for diff sarima model
print('Performance metrics for sarima')

## [1] "Performance metrics for sarima"

mae <- mean(abs(data_test - mypreds$pred))
mse <- mean((data_test - mypreds$pred)^2)
rmse <- sqrt(mse)
mape <- mean(abs((data_test - mypreds$pred) / data_test)) * 100
ss_res <- sum((data_test - mypreds$pred)^2)
ss_tot <- sum((data_test - mean(data_test))^2)
r_squared <- 1 - (ss_res / ss_tot)

cat("MAE:", mae, "\n")

## MAE: 1.167658

cat("MSE:", mse, "\n")

## MSE: 1.976142

cat("RMSE:", rmse, "\n")

## RMSE: 1.405753

cat("MAPE:", mape, "%\n")

## MAPE: 5.141989 %
```

```
cat("R-squared:", r_squared, "\n")

## R-squared: 0.9009433

sarimamodel_full <- arima(dataset$temp, order = c(0, 0, 1), seasonal =
list(order = c(0, 1, 1), period = 12))
sarimamodel_full

##
## Call:
## arima(x = dataset$temp, order = c(0, 0, 1), seasonal = list(order = c(0,
1,
##     1), period = 12))
##
## Coefficients:
##          ma1     sma1
##       0.2346  -0.8971
## s.e.  0.0302   0.0156
##
## sigma^2 estimated as 0.9927:  log likelihood = -1317.53,  aic = 2641.06

mypreds2 <- predict(sarimamodel_full, n.ahead = 120, se.fit = TRUE)
t <- n + 120
year_standardized <- (1947-1/12) +(1:t)/12


fore2 <- c(dataset$temp[(n-36):n], mypreds2$pred)
p <- length(fore2)
foreupper2 <- c(dataset$temp[(n-36):n], mypreds2$pred + 2*mypreds2$se)
forelower2 <- c(dataset$temp[(n-36):n], mypreds2$pred - 2*mypreds2$se)


plot(year_standardized[(t-p+1):t],fore2, ylim = range(c(foreupper2,
forelower2)),type = 'l', lwd = 1.5, main = 'SARIMA forecast for Hong Kong
yearly temperature from 2025 to 2034', xlab = 'year', ylab = 'temperature')
lines(year_standardized[(t-p+1):t], foreupper2, lty = 2, col = 'red')
lines(year_standardized[(t-p+1):t], forelower2, lty = 2, col = 'red')
abline(v = 2025 , lty=2)
```

A forecast for Hong Kong yearly temperature from 20


temperature
year