Appendix - Forecasting Arctic Sea Ice

TSA FINAL PROJECT - Time Series Analysis (MATH1318)

Sean Smyth (s3704987), Daniel Enoka (s3686502) and Charles Galea (s3688570)

04/06/2019

Contents

1 Data pre-processing

- Data Source: https://sites.google.com/site/arcticseaicegraphs/

1.1 Load libraries

1.2 Load data set

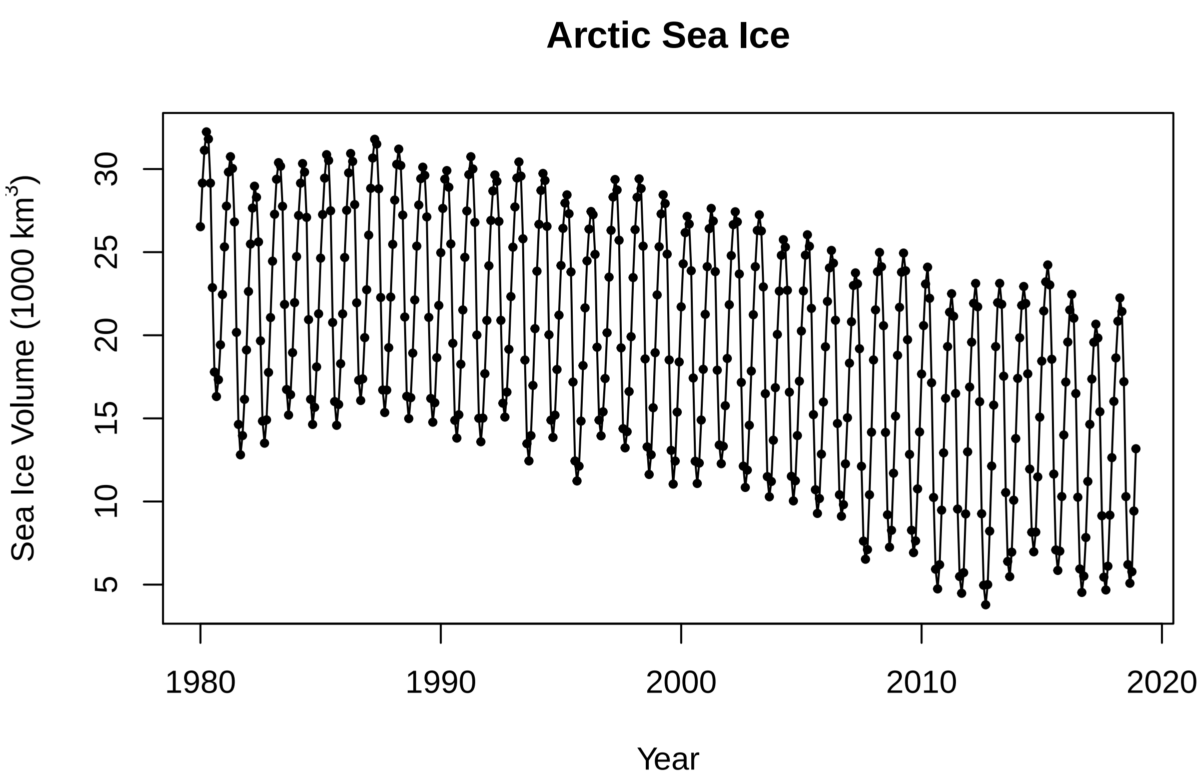
```
sea_ice_orig <-  read.csv("sea_ice_arctic.csv")

colnames(sea_ice_orig) <- c('Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct

sea_ice_orig <- sea_ice_orig %>% gather('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',
```

1.3 Convert to time series

```
sea_ice_orig_ts <- ts(sea_ice_orig$volume, start = 1980, frequency = 12)
```

1.4 Plot data

```
plot(sea_ice_orig_ts, type = "o", pch = 19, cex = 0.5, xlab = "Year", ylab = expression(paste("Sea Ice
```



1.5 Load truncated data set

```
sea_ice_long <-  read.csv("sea_ice_arctic2.csv")

colnames(sea_ice_long) <- c('Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct

sea_ice_long <- sea_ice_long %>% gather('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',
```

1.6 Convert truncated data set to time series

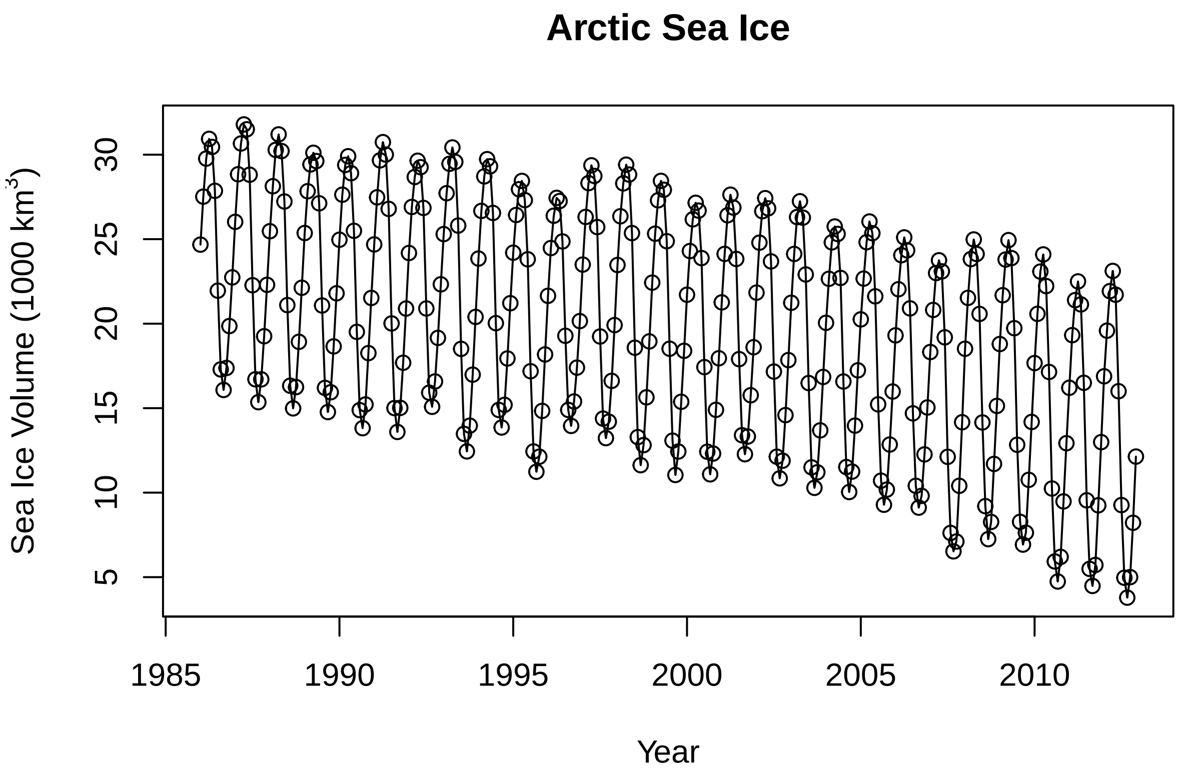
```
sea_ice_long_ts <- ts(sea_ice_long$volume, start = 1986, frequency = 12)
lambda = 0.85
sea_ice_long_BC <- (sea_ice_long_ts^lambda-1)/lambda
```

```
sea_ice <-  read.csv("sea_ice_arctic3.csv")

colnames(sea_ice) <- c('Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'No

sea_ice <- sea_ice %>% gather('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov
```

```
sea_ice_ts <- ts(sea_ice$volume, start = 1986, frequency = 12)
```

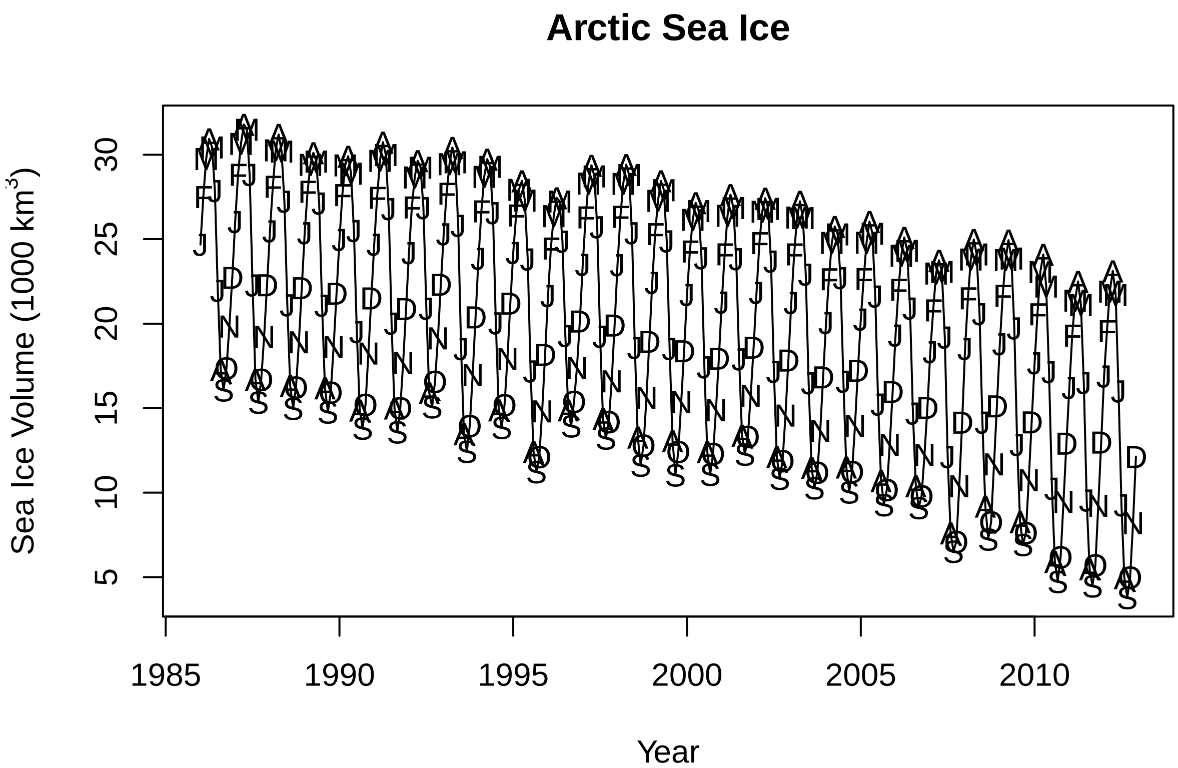
1.7 Plot truncated time series data

```
plot(sea_ice_ts, type = "o", xlab = "Year", ylab = expression(paste("Sea Ice Volume (1000 km"^"3", ")"))
```



1.8 Annotated plot time series of truncated data to examine seasonality

```
plot(sea_ice_ts, ylab=expression(paste("Sea Ice Volume (1000 km"^"3", ")")), xlab = 'Year', type = "l",
points(y=sea_ice_ts,x=time(sea_ice_ts),pch=as.vector(season(sea_ice_ts)))
```



- From the plot of the output we know:

1. Seasonality is present.
2. There is a downward trend (non-stationary).
3. There is no obvious change in variance.
4. It is not possible to determine behaviour due to the presence of seasonality.

2 Harmonic-Trend model

- Set up indicator variables that indicate the month to which each of the data points pertains before estimating parameters.

```
month.=season(sea_ice_ts) # period added to improve table display and this line sets up indicators
model2=lm(sea_ice_ts~month.-1) # -1 removes the intercept term
summary(model2)
```

```
##
## Call:
## lm(formula = sea_ice_ts ~ month. - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.840 -2.387  0.302  2.705  5.265
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## month.January    21.9192     0.6247   35.09   <2e-16 ***
## month.February   24.6609     0.6247   39.48   <2e-16 ***
## month.March      26.6870     0.6247   42.72   <2e-16 ***
## month.April      27.6801     0.6247   44.31   <2e-16 ***
## month.May        26.9158     0.6247   43.09   <2e-16 ***
## month.June       23.5491     0.6247   37.70   <2e-16 ***
## month.July       17.1039     0.6247   27.38   <2e-16 ***
## month.August     12.3081     0.6247   19.70   <2e-16 ***
## month.September  11.0253     0.6247   17.65   <2e-16 ***
## month.October    12.1887     0.6247   19.51   <2e-16 ***
## month.November   14.9781     0.6247   23.98   <2e-16 ***
## month.December   18.2201     0.6247   29.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.246 on 312 degrees of freedom
## Multiple R-squared:  0.9768, Adjusted R-squared:  0.9759
## F-statistic:  1093 on 12 and 312 DF,  p-value: < 2.2e-16
```

All of the parameters corresponding to months are statistically significant at 5% level.

```r
month.=season(sea_ice_ts) # period added to improve table display and this line sets up indicators
model3=lm(sea_ice_ts~month.) # -1 removes the intercept term
summary(model3)
```

```
##
## Call:
## lm(formula = sea_ice_ts ~ month.)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.840 -2.387  0.302  2.705  5.265
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      21.9192     0.6247  35.088  < 2e-16 ***
## month.February    2.7417     0.8834   3.103  0.00209 **
## month.March       4.7678     0.8834   5.397 1.34e-07 ***
## month.April       5.7609     0.8834   6.521 2.81e-10 ***
## month.May         4.9966     0.8834   5.656 3.51e-08 ***
## month.June        1.6300     0.8834   1.845  0.06598 .
## month.July       -4.8153     0.8834  -5.451 1.02e-07 ***
## month.August     -9.6110     0.8834 -10.879  < 2e-16 ***
## month.September -10.8939     0.8834 -12.331  < 2e-16 ***
## month.October    -9.7304     0.8834 -11.014  < 2e-16 ***
## month.November   -6.9411     0.8834  -7.857 6.42e-14 ***
## month.December   -3.6990     0.8834  -4.187 3.68e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.246 on 312 degrees of freedom
## Multiple R-squared:  0.7779, Adjusted R-squared:  0.7701
## F-statistic: 99.33 on 11 and 312 DF,  p-value: < 2.2e-16
```

2.1 Fit of cosine curve to average monthly Arctic sea ice series.

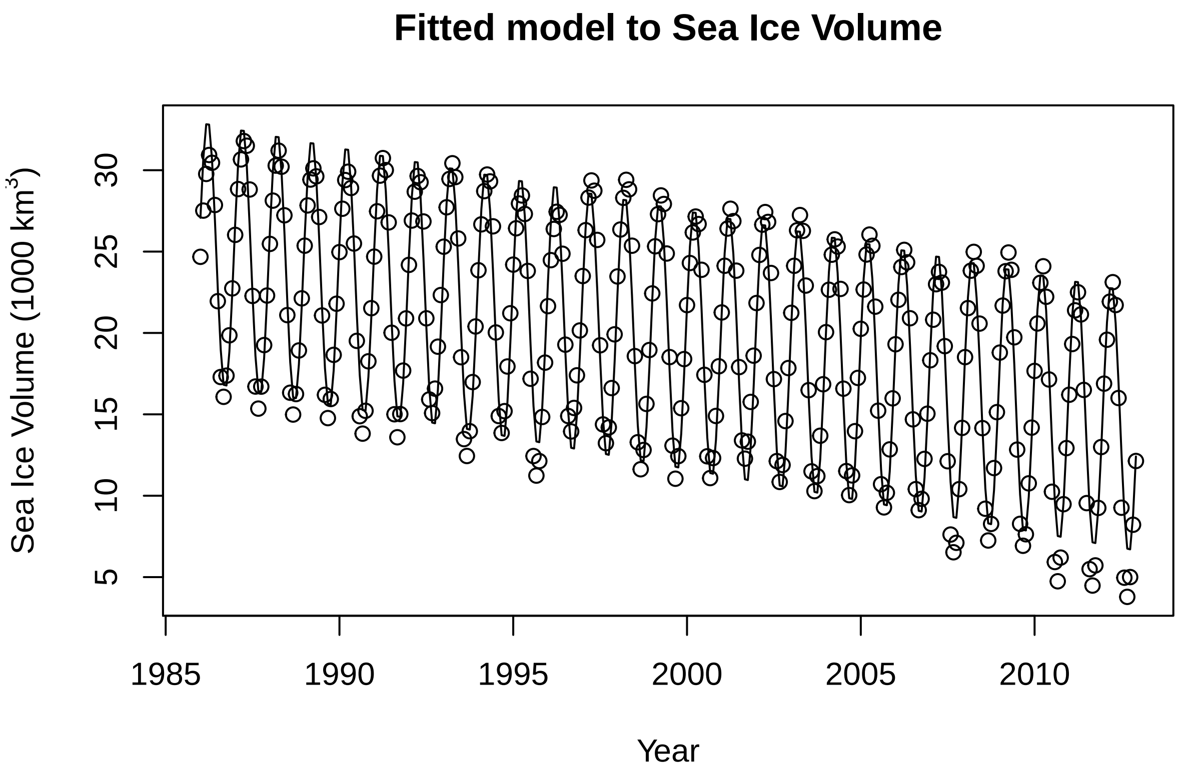
```
har.=harmonic(sea_ice_ts,1) # calculate cos(2*pi*t) and sin(2*pi*t)
t1 <- time(sea_ice_ts)
model4=lm(sea_ice_ts~har. + t1)
summary(model4)
```

```
##
## Call:
## lm(formula = sea_ice_ts ~ har. + t1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9501 -0.9066  0.0295  0.9480  3.0424
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     794.03512   20.30584   39.10   <2e-16 ***
## har.cos(2*pi*t)   2.09440    0.11190   18.72   <2e-16 ***
## har.sin(2*pi*t)   7.91857    0.11194   70.74   <2e-16 ***
## t1               -0.38724    0.01016  -38.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.424 on 320 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9557
## F-statistic:  2326 on 3 and 320 DF,  p-value: < 2.2e-16
```

Low p-values indicate a high degree of significance for the determined parameters. The R-squared value is high (0.96) which is very good but care should be taken to ensure this is not due to overfitting.

2.2 Plot of fitted Harmonic-Trend curve along with observed average monthly Arctic sea ice series.

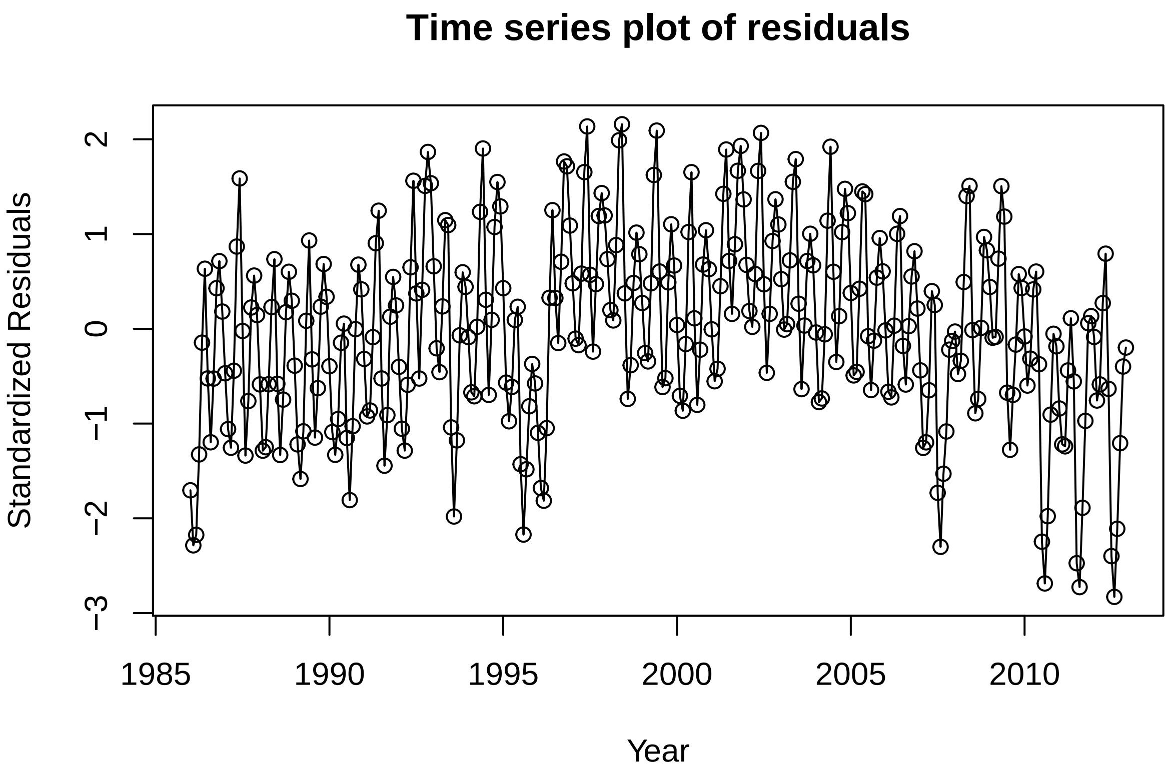
```
plot(ts(fitted(model4),freq=12,start=c(1986,1)),xlab = 'Year', ylab=expression(paste("Sea Ice Volume (10
ylim=range(c(fitted(model4),sea_ice_ts)),main="Fitted model to Sea Ice Volume") # ylim ensures that the
points(sea_ice_ts)
```



* The Harmonic-Trend model appears to fit reasonably well to the average monthly Arctic sea ice data up to 2007 but there is some deviation beyond this point.

2.3 Time series plot of standardized residuals.

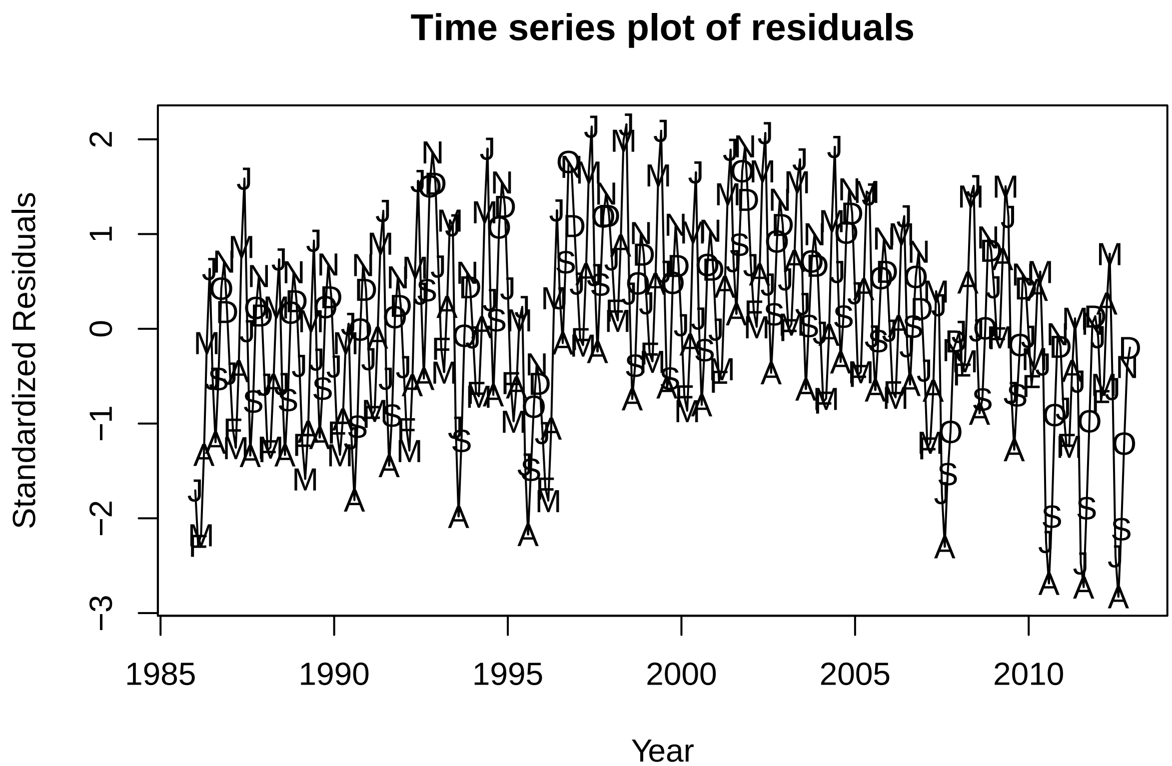
```
plot(y=rstudent(model4),x=as.vector(time(sea_ice_ts)), xlab='Year',ylab='Standardized Residuals',type='
```



* There was a departure from randomness in the plot of the standardized residuals. Therefore, labeled months to determine if there is a trend present.

2.4 Labeled months in plot of standardized residuals.

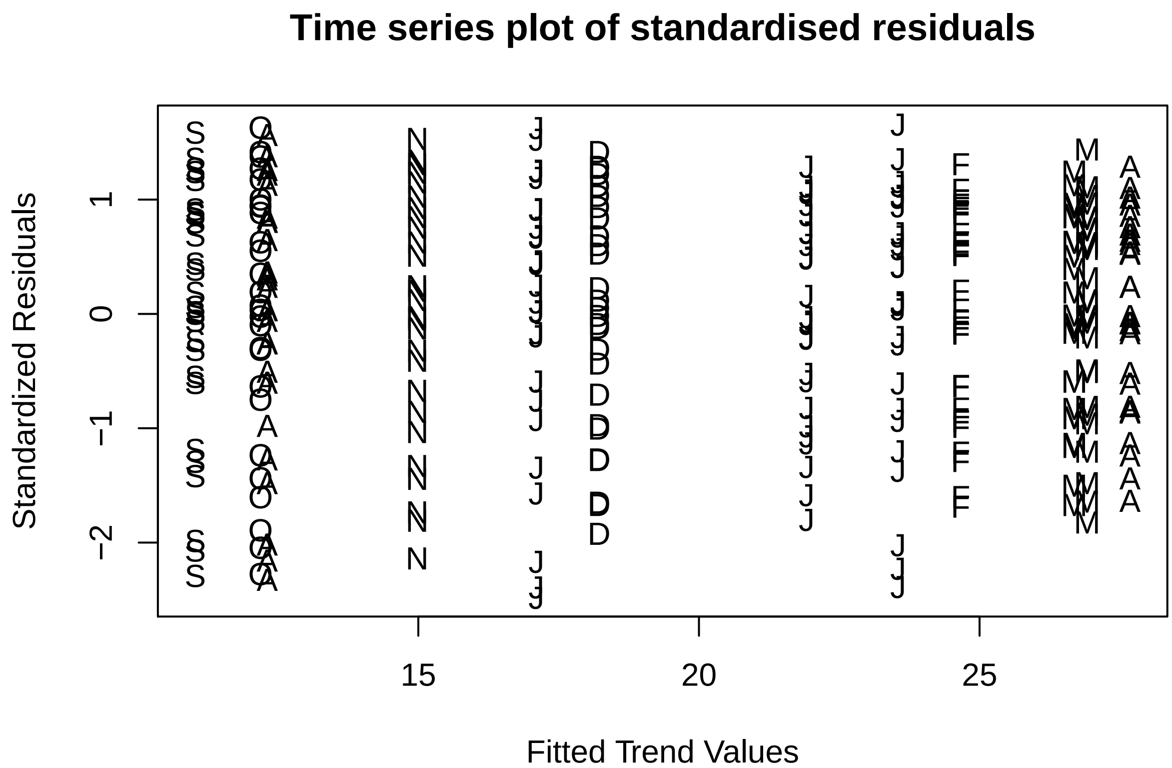
```
plot(y=rstudent(model4),x=as.vector(time(sea_ice_ts)),xlab='Year', ylab='Standardized Residuals',type='l
points(y=rstudent(model4),x=as.vector(time(sea_ice_ts)), pch=as.vector(season(sea_ice_ts)))
```



* Some high and low points but appears more random than the original series.

2.5 Plot of standardized residuals with labels.

```
plot(y=rstudent(model3),x=as.vector(fitted(model3)), xlab='Fitted Trend Values', ylab='Standardized Resi
points(y=rstudent(model3),x=as.vector(fitted(model3)),pch=as.vector(season(sea_ice_ts)))
```

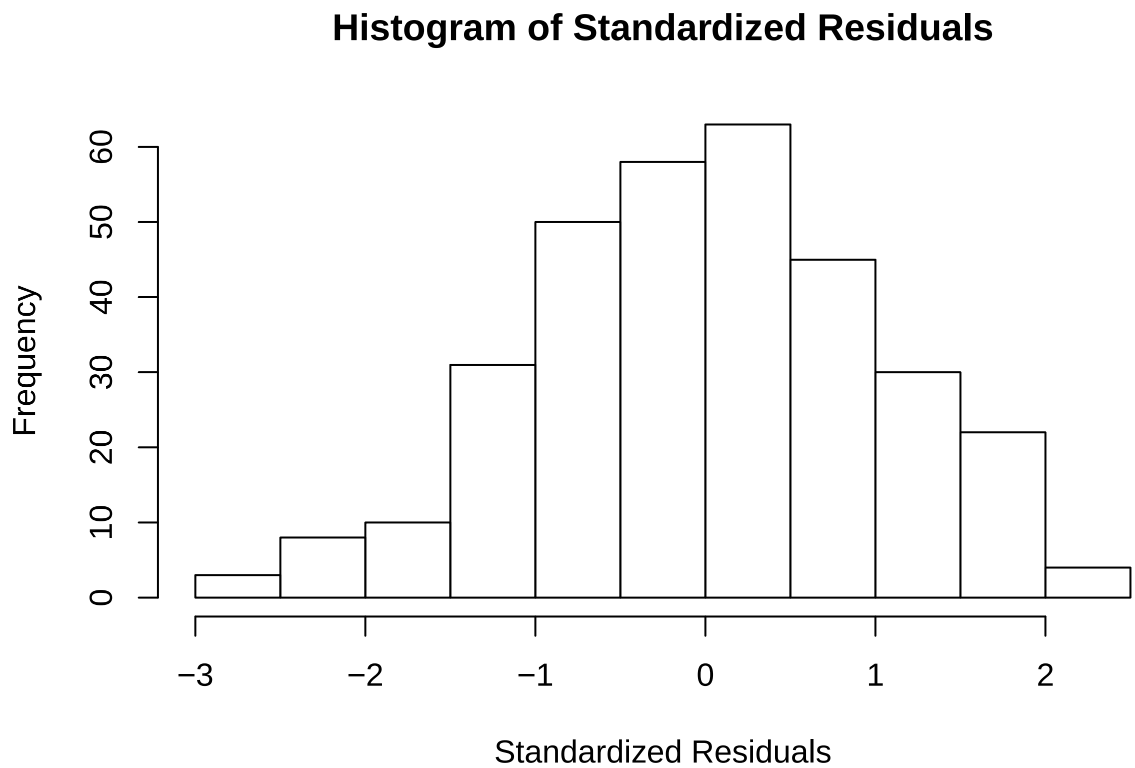


* There was a similar spread of labels across the plot. The plot does not indicate any dramatic patterns that would cause us to doubt the seasonal means model.

2.6 Normality of standardized residuals

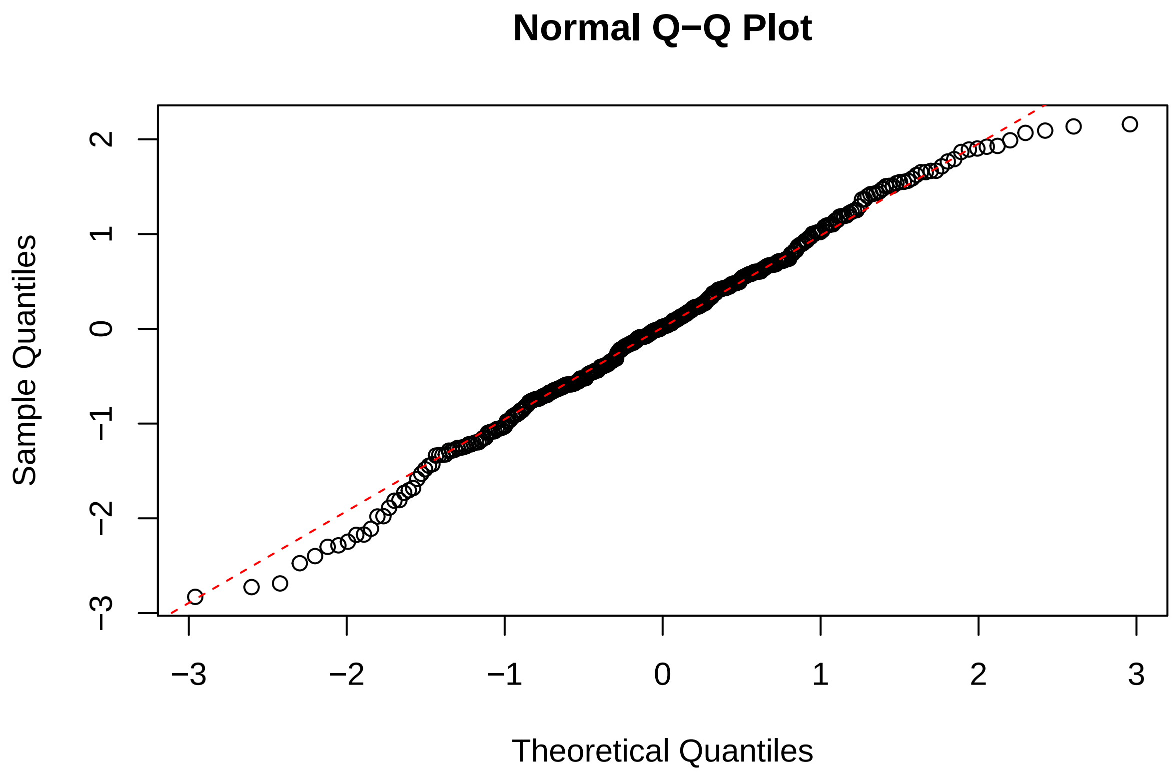
2.6.1 Histogram of standardized residuals for Harmoic-Trend model.

```
hist(rstudent(model4),xlab='Standardized Residuals', main = 'Histogram of Standardized Residuals')
```



2.6.2 Q-Q plot of standardized residuals for Harmoic-Trend model.

```
y = rstudent(model4)
qqnorm(y)
qqline(y, col = 2, lwd = 1, lty = 2)
```

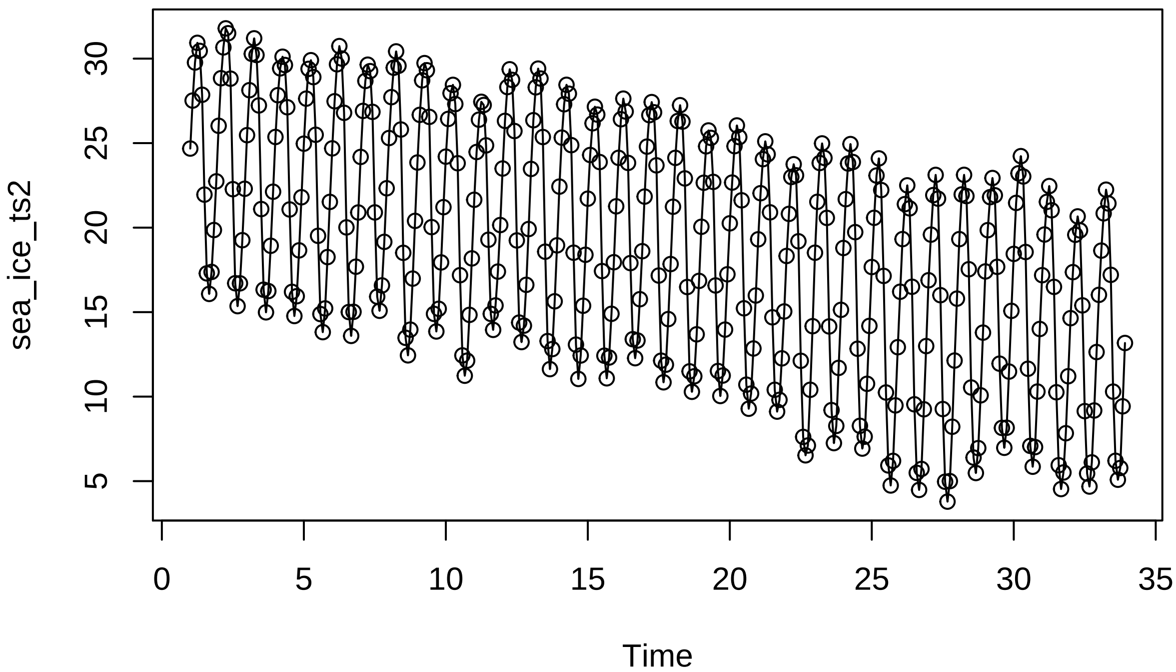



```
shapiro.test(model4$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model4$residuals
## W = 0.99224, p-value = 0.08909
```

- The histogram, Q-Q plot of standardized residuals for the Harmoic-Trend model shows a normal distribution. The Shpiro-Wilk test gave a p-value of 0.089 indicating that we can not reject the null hypothesis that the stocastic component of this model is normally distributed.

```
sea_ice2 = read.csv("sea_ice_arctic2.csv")
colnames(sea_ice2) <- c('Year','Jan', 'Feb', 'Mar', 'Apr', 'May', 'June','July','Aug', 'Sep', 'Oct' , 'I
sea_ice2 <- sea_ice2 %>% gather('Jan', 'Feb', 'Mar', 'Apr', 'May', 'June','July','Aug', 'Sep', 'Oct' ,
sea_ice_ts2 <- ts(sea_ice2$volume, start = 1, frequency = 12)
plot(sea_ice_ts2, type = "o")
```



2.7 Three-year forecast of Arctic sea ice volumes from 2013 to 2015 using the Harmonic-Trend model.

```
har.=harmonic(sea_ice_ts2,1) # calculate cos(2*pi*t) and sin(2*pi*t)
t3 <- time(sea_ice_ts2)
t1 = har.[,1] # To make it easier assign harmonic variables to separate variables
t2 = har.[,2]
model4=lm(sea_ice_ts2~t3+t1+t2) # Fit the model with separate variables
# We need to create continuous time for 12 months starting from the first month of 2013
t = c(34.000, 34.083, 34.167 ,34.250, 34.333, 34.417 ,34.500, 34.583, 34.667, 34.750, 34.833, 34.917, 3
t1 = cos(2*pi*t)
t2 = sin(2*pi*t)
t3 <- t
t3
```

```
##  [1] 34.000 34.083 34.167 34.250 34.333 34.417 34.500 34.583 34.667 34.750
## [11] 34.833 34.917 35.000 35.083 35.167 35.250 35.333 35.417 35.500 35.583
## [21] 35.667 35.750 35.833 35.917 36.000 36.083 36.167 36.250 36.333 36.417
## [31] 36.500 36.583 36.667 36.750 36.833 36.917 37.000
```

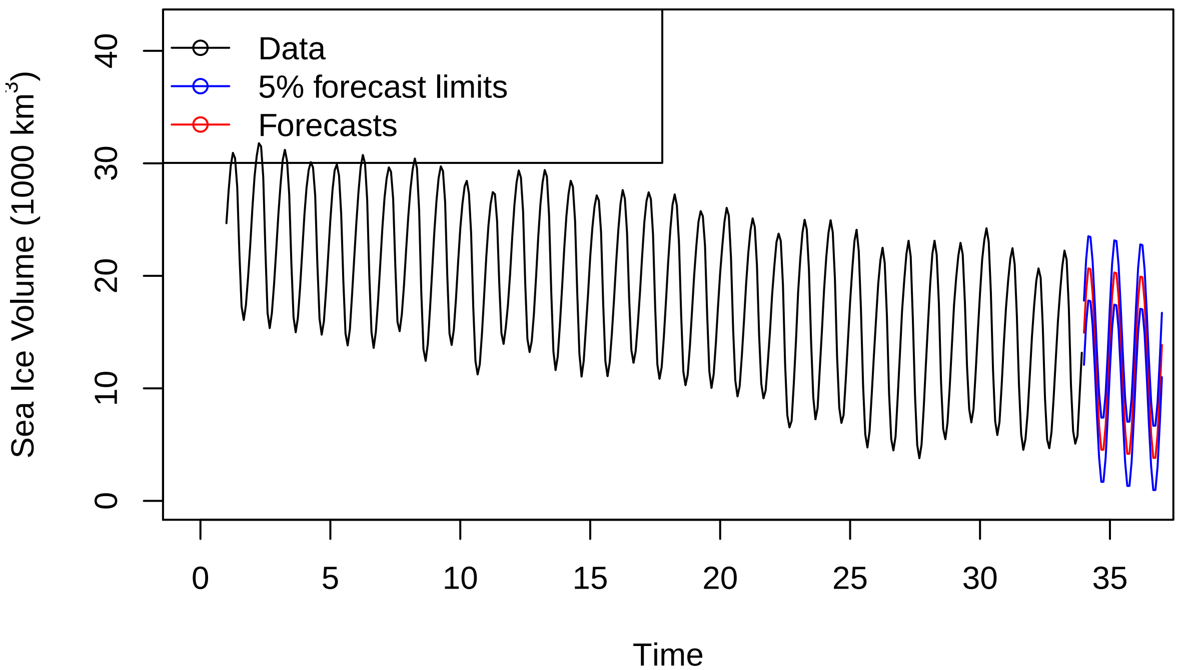
```
new = data.frame(t3, t1 , t2) # Step 1
# Notice here that I'm using the same variable names "t1" and "t2" as in the
# fitted model above, where the name of the variables showing sine and cosine
# components are also "t1" and "t2". To run the predict() function properly,
# the names of variables in fitted model and "new" data frame
```

```r
# must be the same!!!
forecasts = predict(model4, new, interval = "prediction")
print(forecasts)
```

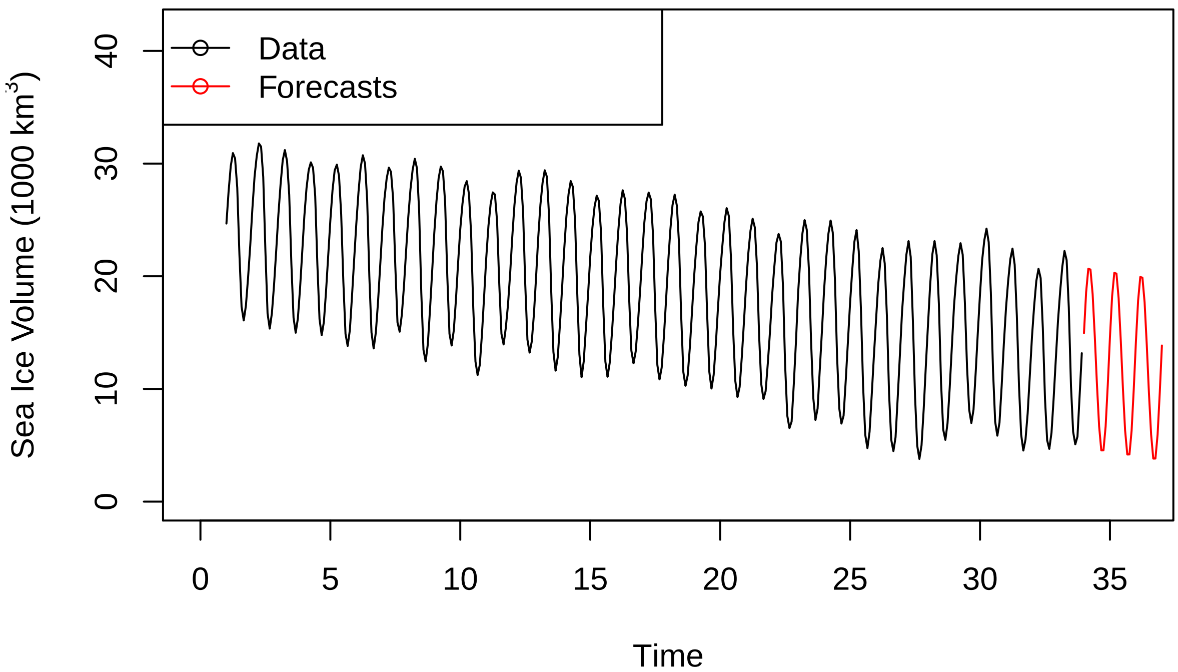
```
##          fit        lwr       upr
## 1  14.946060 12.0910334 17.801086
## 2  18.577035 15.7217116 21.432358
## 3  20.664983 17.8094369 23.520529
## 4  20.598113 17.7424564 23.453770
## 5  18.422072 15.5664148 21.277729
## 6  14.656961 11.8013863 17.512536
## 7  10.383300  7.5278354 13.238764
## 8   6.692016  3.8366324  9.547400
## 9   4.543033  1.6876495  7.398417
## 10  4.549595  1.6941007  7.405089
## 11  6.665328  3.8096123  9.521043
## 12 10.369404  7.5133780 13.225429
## 13 14.582757 11.7263900 17.439123
## 14 18.213732 15.3570505 21.070414
## 15 20.301680 17.4447628 23.158597
## 16 20.234810 17.3777759 23.091845
## 17 18.058769 15.2017343 20.915803
## 18 14.293658 11.4367106 17.150606
## 19 10.019997  7.1631661 12.876828
## 20  6.328713  3.4719678  9.185458
## 21  4.179730  1.3229849  7.036476
## 22  4.186292  1.3294297  7.043154
## 23  6.302025  3.4449284  9.159121
## 24 10.006101  7.1486763 12.863525
## 25 14.219454 11.3616690 17.077238
## 26 17.850429 14.9923119 20.708546
## 27 19.938377 17.0800112 22.796743
## 28 19.871507 17.0130178 22.729997
## 29 17.695466 14.8369763 20.553956
## 30 13.930355 11.0719573 16.788753
## 31  9.656694  6.7984193 12.514969
## 32  5.965410  3.1072257  8.823595
## 33  3.816427  0.9582427  6.674612
## 34  3.822989  0.9646811  6.681297
## 35  5.938722  3.0801669  8.797277
## 36  9.642798  6.7838971 12.501698
## 37 13.856151 10.9968705 16.715431
```

2.8 Plot of 3-year forecast of Arctic sea ice.

```r
plot(sea_ice_ts2, xlim = c(0,36), ylim = c(0, 42), ylab = expression(paste("Sea Ice Volume (1000 km"^"3
# Here we convert the forecasts and prediction limits to monthly time series!
lines(ts(as.vector(forecasts[,1]), start = c(34,1), frequency = 12), col="red", type="l")
lines(ts(as.vector(forecasts[,2]), start = c(34,1), frequency = 12), col="blue", type="l")
lines(ts(as.vector(forecasts[,3]), start = c(34,1), frequency = 12), col="blue", type="l")
legend("topleft", lty=1, pch=1, col=c("black","blue","red"), text.width = 15,
       c("Data","5% forecast limits", "Forecasts"))
```




```
plot(sea_ice_ts2, xlim = c(0,36), ylim = c(0, 42), ylab = expression(paste("Sea Ice Volume (1000 km"^"3
# Here we convert the forecasts and prediction limits to monthly time series!
lines(ts(as.vector(forecasts[,1]), start = c(34,1), frequency = 12), col="red", type="l")
legend("topleft", lty=1, pch=1, col=c("black","red"), text.width = 15,
       c("Data","Forecasts"))
```

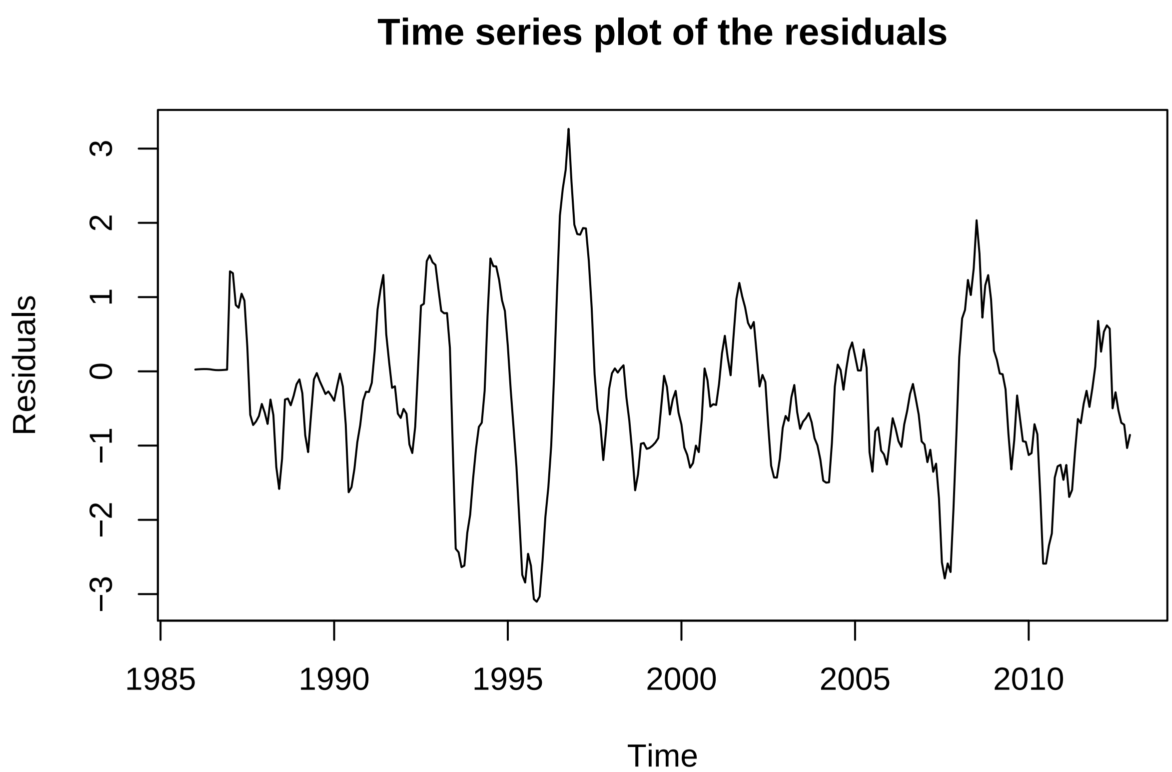


3 Nonstationary seasonal ARIMA (SARIMA) model (Residuals Approach).

3.1 Fit of SARIMA models.

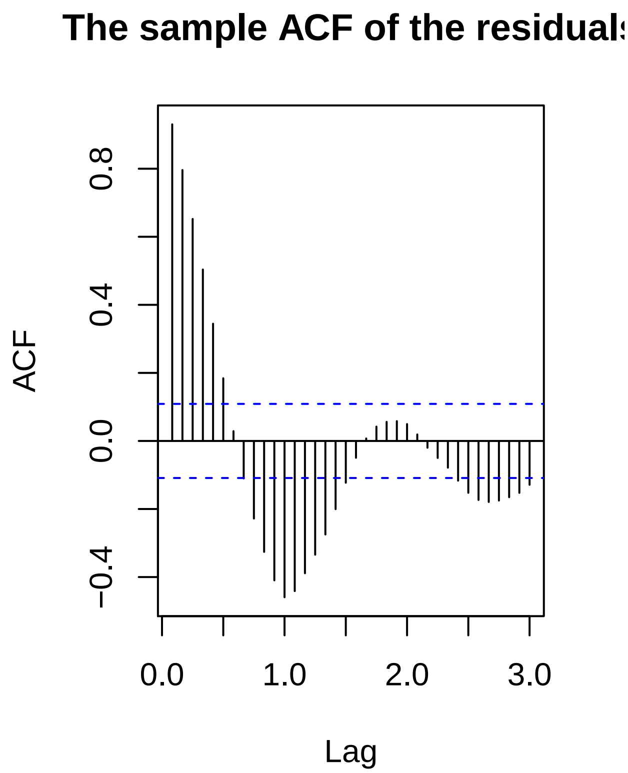
3.1.1 Initial fit of SARIMA(0,0,0)x(0,1,0)$_{12}$ model

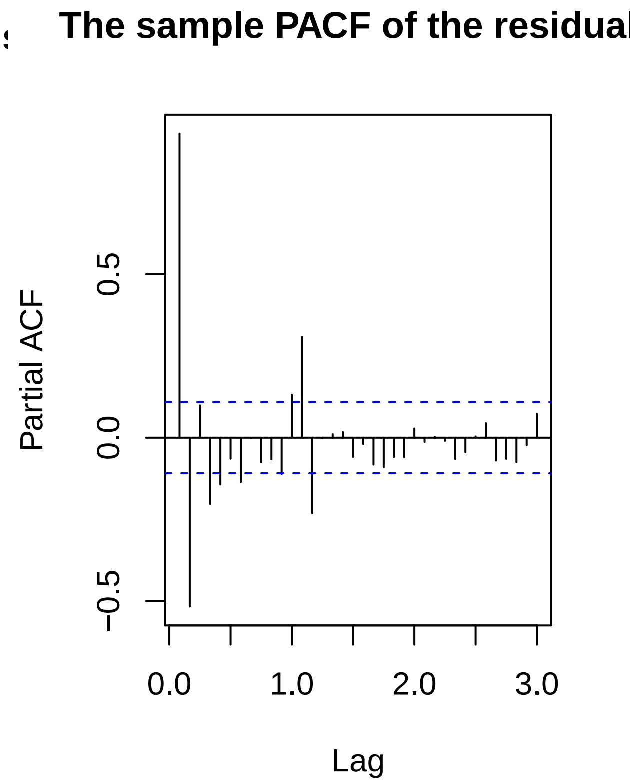
```
m100_sea_ice = arima(sea_ice_ts,order=c(0,0,0),seasonal=list(order=c(0,1,0), period=12))
res_m100 = residuals(m100_sea_ice);
plot(res_m100,xlab='Time', ylab='Residuals',main="Time series plot of the residuals")
```



3.1.2 ACF and PACF plots of residuals for SARIMA(0,0,0)x$(0,1,0)_{12}$ model

```
par(mfrow=c(1,2))
acf(res_m100, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m100, lag.max = 36, main = "The sample PACF of the residuals")
```

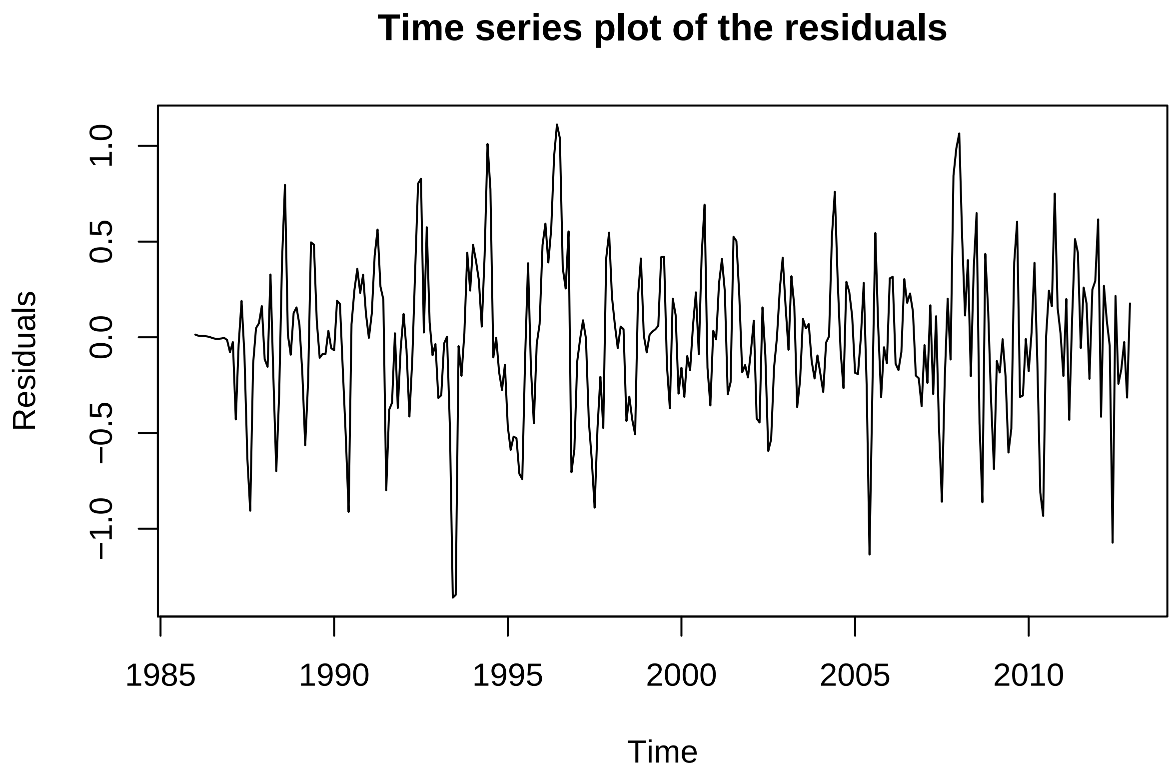




- There is now no pattern implying the existence of a seasonal trend.
- However, the slowly decaying pattern prior to the first period at 1s implies the existence of an ordinary trend in the ACF plot of the residuals.
- We will attempt to remove this ordinary trend by fitting a SARIMA(0,1,0)x$(0,1,0)_{12}$ model.

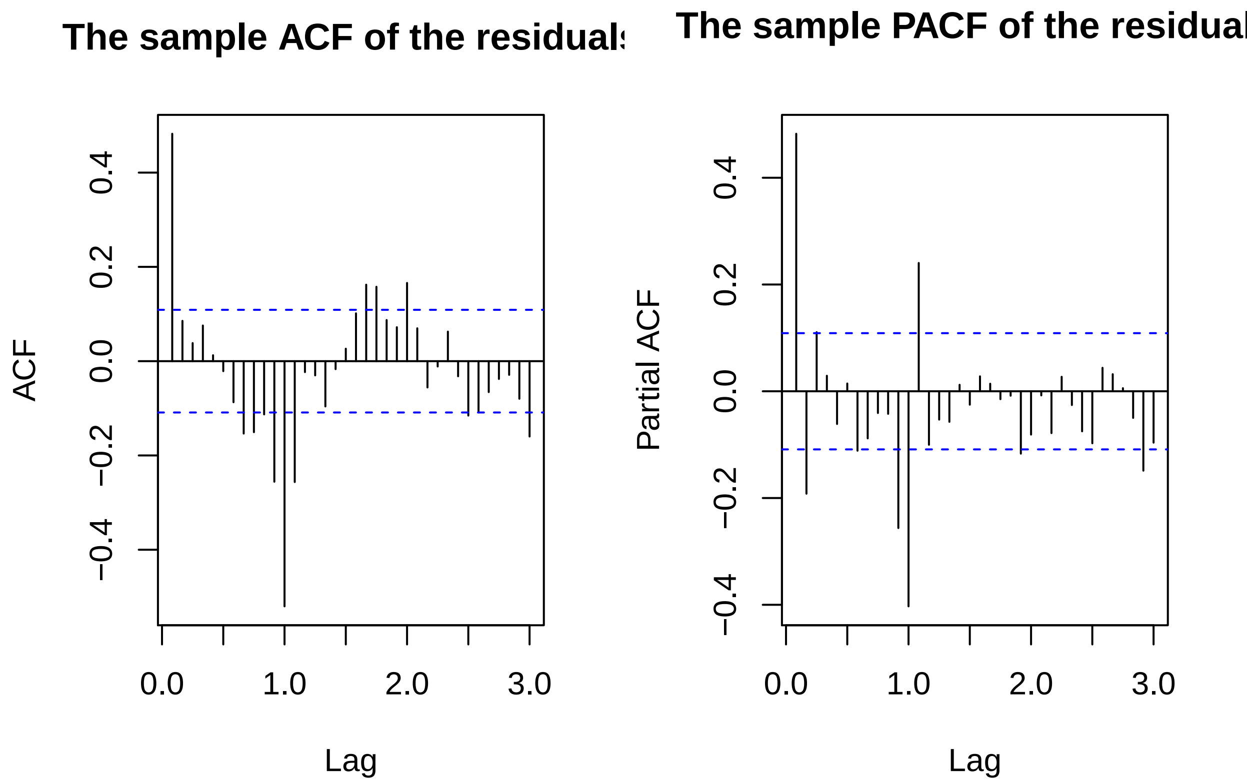
3.1.3 Fit of SARIMA(0,1,0)x$(0,1,0)_{12}$ model

```
m200_sea_ice = arima(sea_ice_ts,order=c(0,1,0),seasonal=list(order=c(0,1,0), period=12))
res_m200 = residuals(m200_sea_ice);
plot(res_m200,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.4 ACF and PACF plots for SARIMA(0,1,0)x$(0,1,0)_{12}$ model.

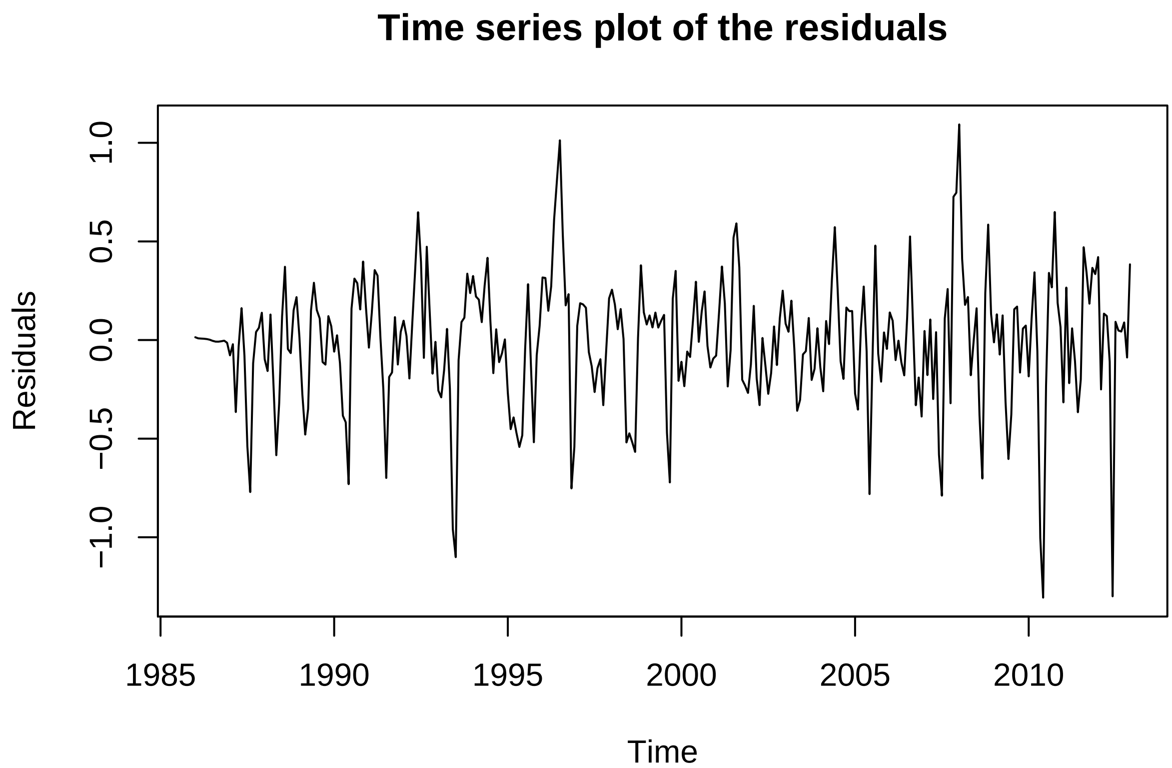
```
par(mfrow=c(1,2))
acf(res_m200, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m200, lag.max = 36, main = "The sample PACF of the residuals")
```



- No evidence of an ordinary trend remaining in the residuals.
- There is a decreasing pattern in lags 1, 2, 3,. . . . in the SARMA conponent of the PACF plot. The correation at lag 1 in the ACF plot is significant. This implies the existence of an SMA(1) component.
- ACF - 1 significant lag at 1s and 1 in first part prior to 1s (q=1, Q=1)
- PACF - 1 significant lag and 1 not so significant lag in first part and 1 significant after 1s (p=1,2, P=1)
- Now tried to fit a SARIMA(0,1,0)x$(0,1,1)_{12}$ model to try to remove the remaining seasonal component in the residuals.

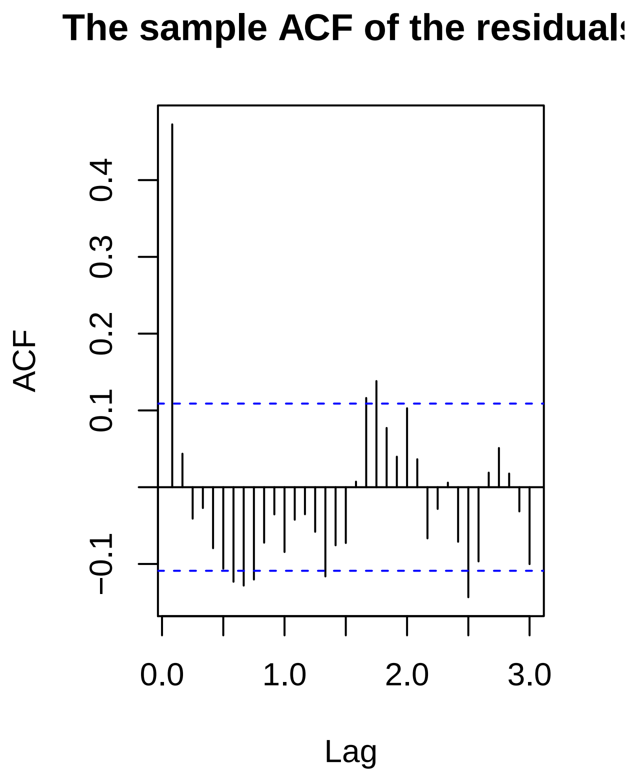
3.1.5 Fit of SARIMA(0,1,0)x$(0,1,1)_{12}$ model.

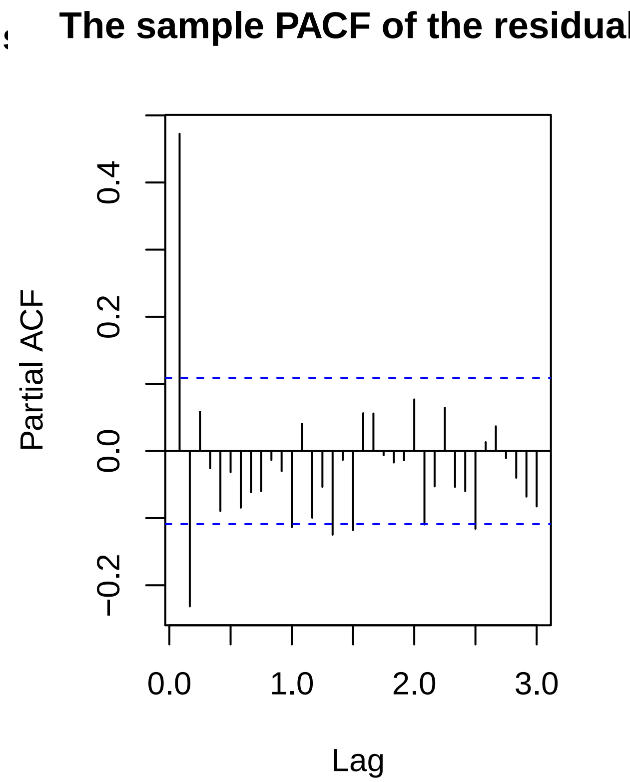
```
m300_sea_ice = arima(sea_ice_ts,order=c(0,1,0),seasonal=list(order=c(0,1,1), period=12))
res_m300 = residuals(m300_sea_ice);
plot(res_m300,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.6 ACF AND PACF plots for SARIMA(0,1,0)x$(0,1,1)_{12}$ model.

```
par(mfrow=c(1,2))
acf(res_m300, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m300, lag.max = 36, main = "The sample PACF of the residuals")
```





- The autocorrelations, especially the first seasonal lag (lag 1) in the ACF plot, become insignificant after adding the seasonal component.
- The ACF and PACF plots can be used to determine the orders of the ARMA component since there are no highly significant correlations at lags s, 2s, 3s, ...
- The ACF plot displays one significant and 3 less significant autocorrelation (q=1,2,3) while the PACF plot has two significant autocorrelations (p=1,2). This suggests a ARMA(1,1), ARMA(1,2), ARMA(2,1), ARMA(1,3) and ARMA(2,3) models.
- We will now fit SARIMA(1,1,1)x(0,1,1)$_{12}$, SARIMA(1,1,2)x(0,1,1)$_{12}$ models, SARIMA(2,1,1)x(0,1,1)$_{12}$, SARIMA(1,1,3)x(0,1,1)$_{12}$ and SARIMA(2,1,3)x(0,1,1)$_{12}$ models.

3.1.6.1 EACF analysis of the residuals.

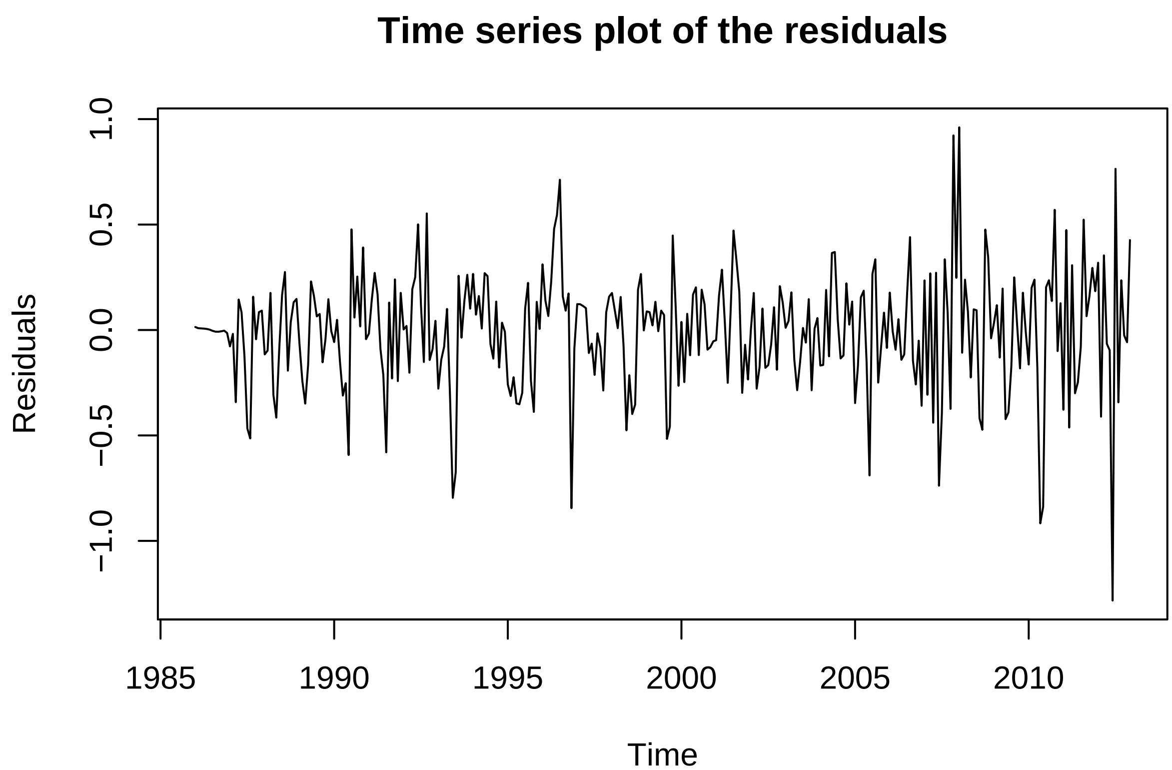
```
eacf(res_m300)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o x x x o o  o  o  o
## 1 x x o o o o o o o o o  o  o  o
## 2 x o x o o o o o o o o  o  o  o
## 3 x o x x o o o o o o o  o  o  o
## 4 x x x x o o o o o o o  o  o  o
## 5 x x o x o o o o o o o  o  o  o
## 6 x x o x o o o o o o o  o  o  o
## 7 x x x x o o o o o o o  o  o  o
```

- The tentative models are specified as
- SARIMA(1,1,1)x(0,1,1)_12
- SARIMA(1,1,2)x(0,1,1)_12
- SARIMA(1,1,3)x(0,1,1)_12
- SARIMA(2,1,2)x(0,1,1)_12
- From the EACF, we will include
- SARIMA(0,1,1)x(0,1,1)_12
- SARIMA(0,1,2)x(0,1,1)_12

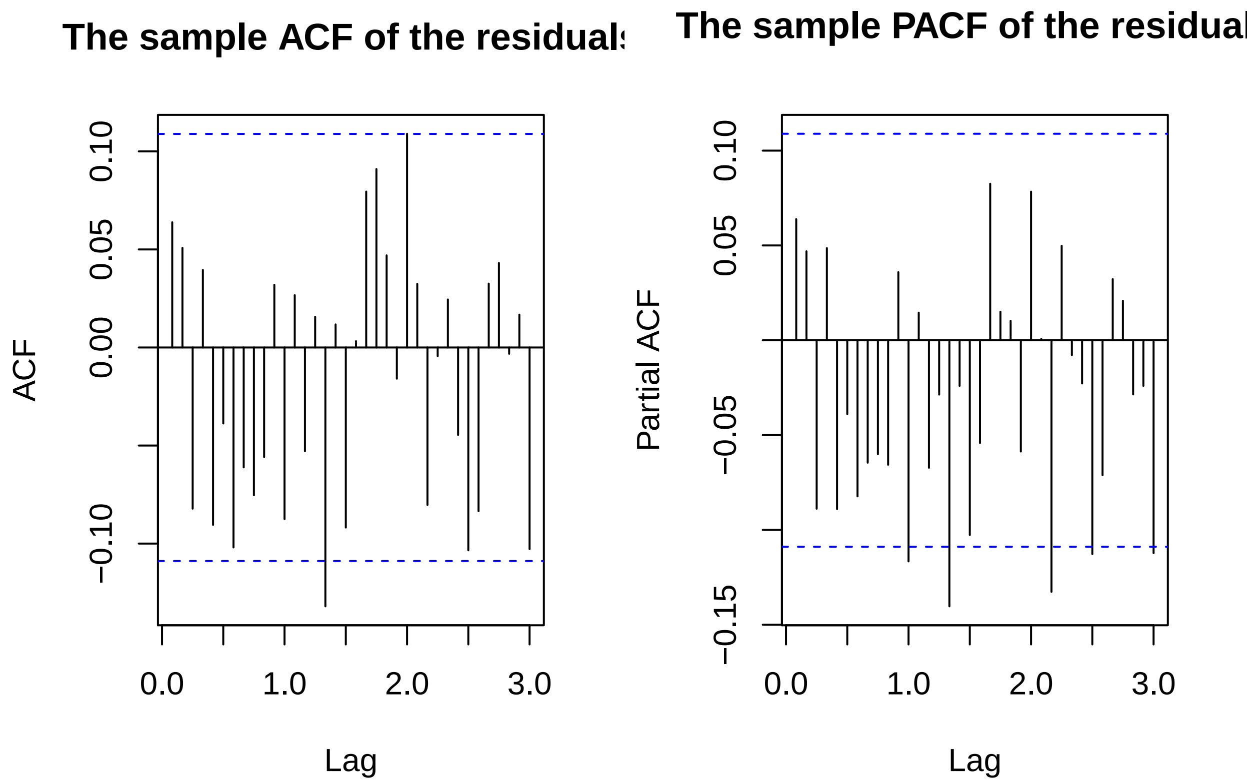
3.1.7 Fit of SARIMA(0,1,1)x(0,1,1)$_{12}$ model

```
m1_sea_ice = arima(sea_ice_ts,order=c(0,1,1),seasonal=list(order=c(0,1,1), period=12))
res_m1 = residuals(m1_sea_ice);
plot(res_m1,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



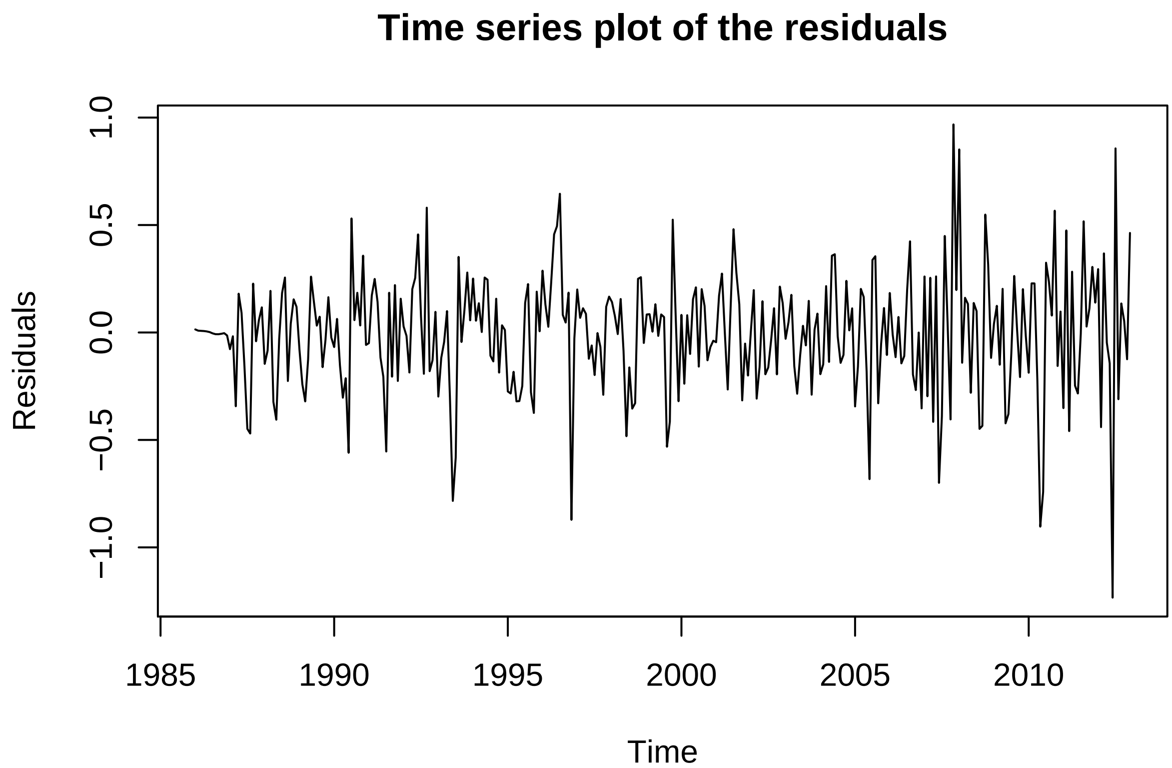
3.1.8 ACF and PACF plots for SARIMA(0,1,1)x(0,1,1)$_{12}$ model

```
par(mfrow=c(1,2))
acf(res_m1, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m1, lag.max = 36, main = "The sample PACF of the residuals")
```



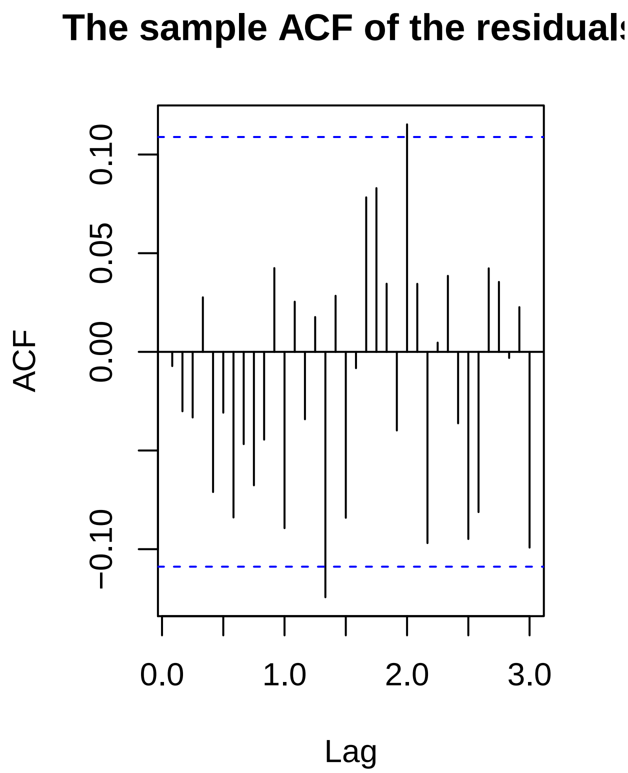
3.1.9 Fit of SARIMA(0,1,2)x$(0,1,1)_{12}$ model

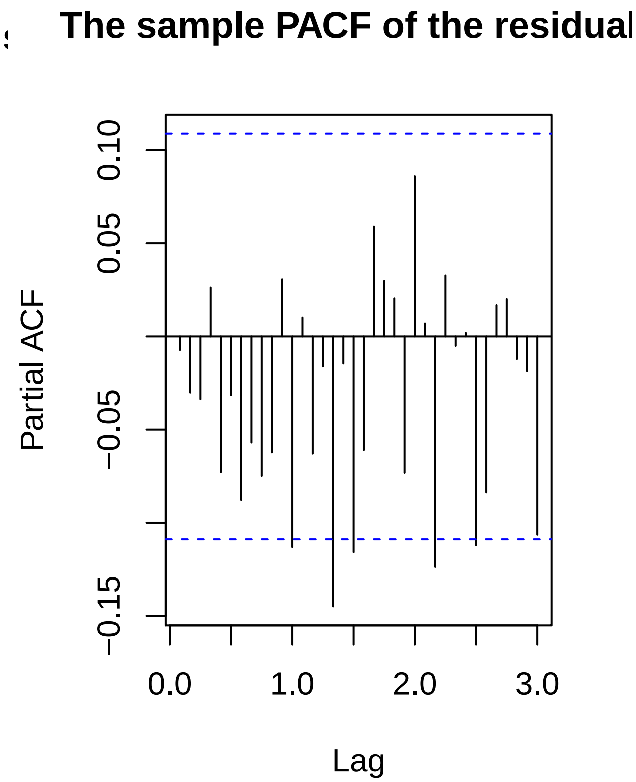
```
m2_sea_ice = arima(sea_ice_ts,order=c(0,1,2),seasonal=list(order=c(0,1,1), period=12))
res_m2 = residuals(m2_sea_ice);
plot(res_m2,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.10 ACF and PACF plots for SARIMA(0,1,2)x(0,1,1)$_{12}$ model

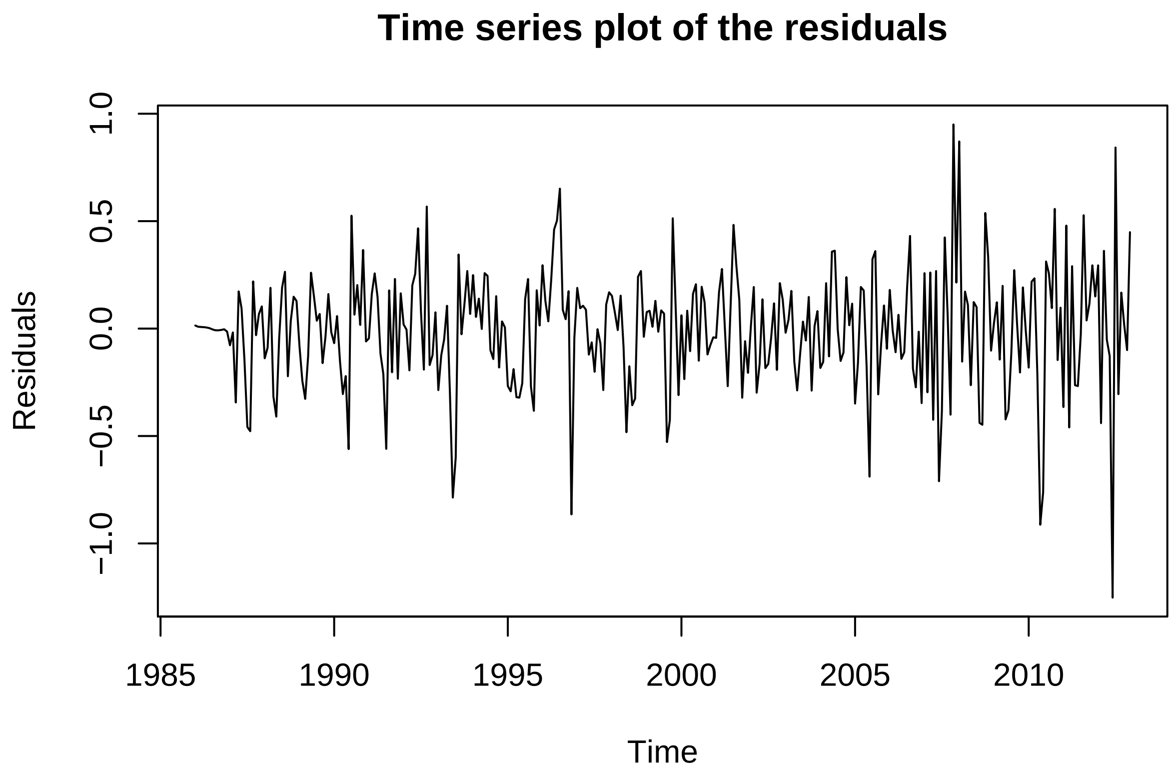
```
par(mfrow=c(1,2))
acf(res_m2, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m2, lag.max = 36, main = "The sample PACF of the residuals")
```





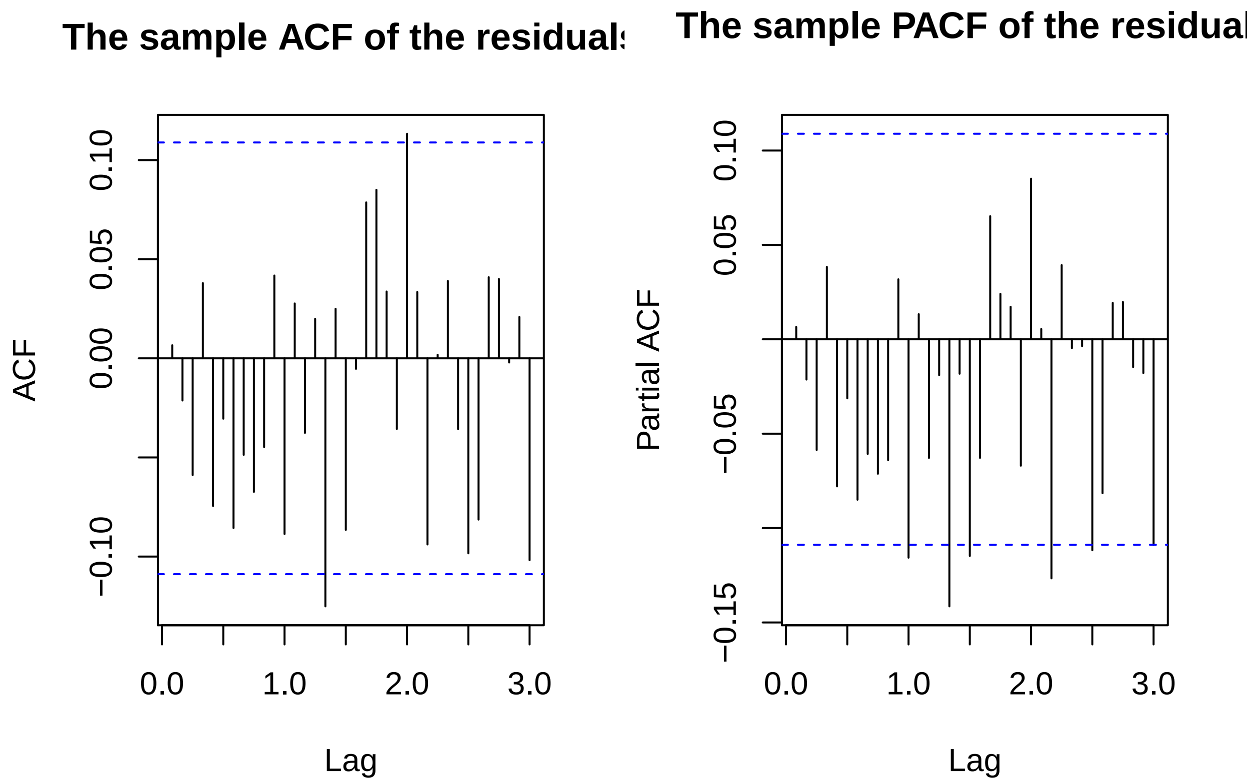
3.1.11 Fit of SARIMA(1,1,1)x(0,1,1)$_{12}$ model

```
m3_sea_ice = arima(sea_ice_ts,order=c(1,1,1),seasonal=list(order=c(0,1,1), period=12))
res_m3 = residuals(m3_sea_ice);
plot(res_m3,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



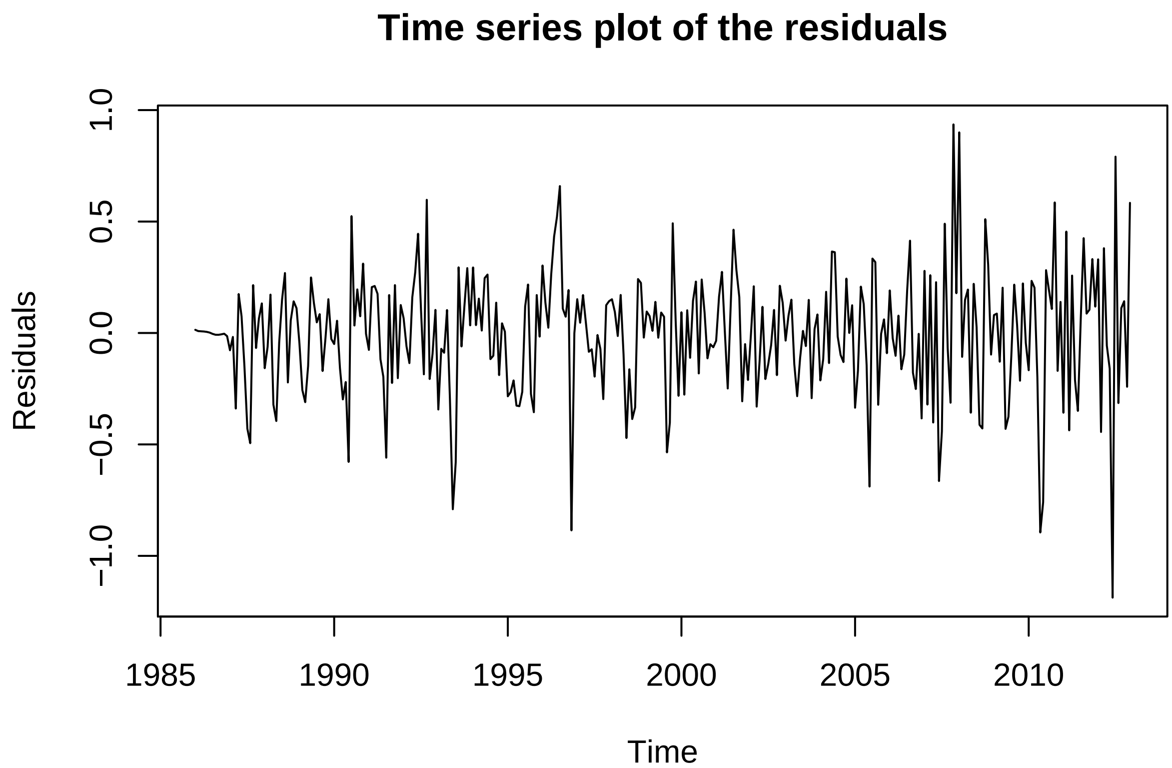
3.1.12 ACF and PACF plots for SARIMA(1,1,1)x(0,1,1)$_{12}$ model

```
par(mfrow=c(1,2))
acf(res_m3, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m3, lag.max = 36, main = "The sample PACF of the residuals")
```



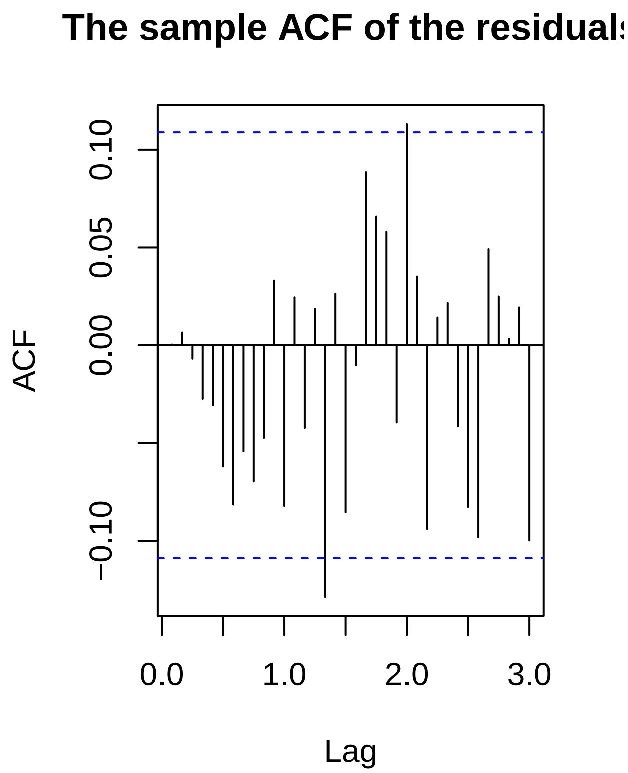
3.1.13 Fit of SARIMA(1,1,2)x$(0,1,1)_{12}$ model

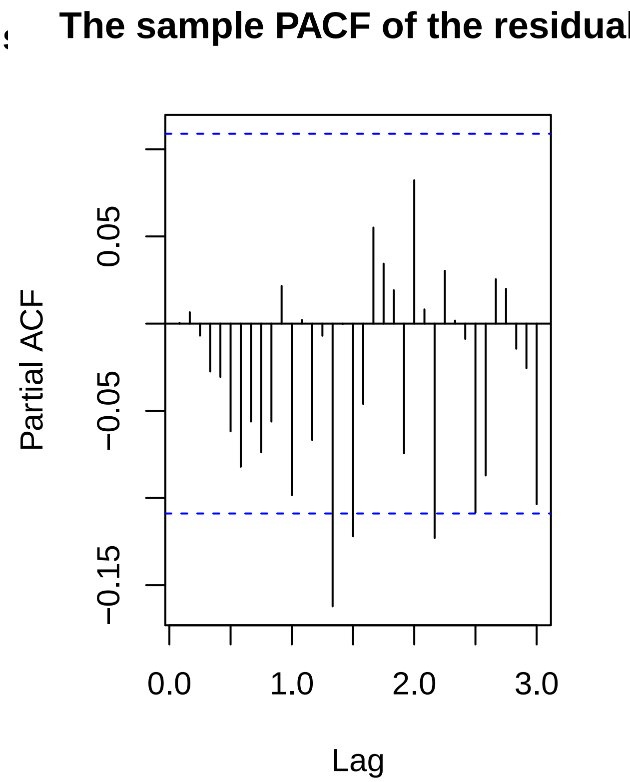
```
m4_sea_ice = arima(sea_ice_ts,order=c(1,1,2),seasonal=list(order=c(0,1,1), period=12))
res_m4 = residuals(m4_sea_ice);
plot(res_m4,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.14 ACF and PACF plots for SARIMA(1,1,2)x$(0,1,1)_{12}$ model

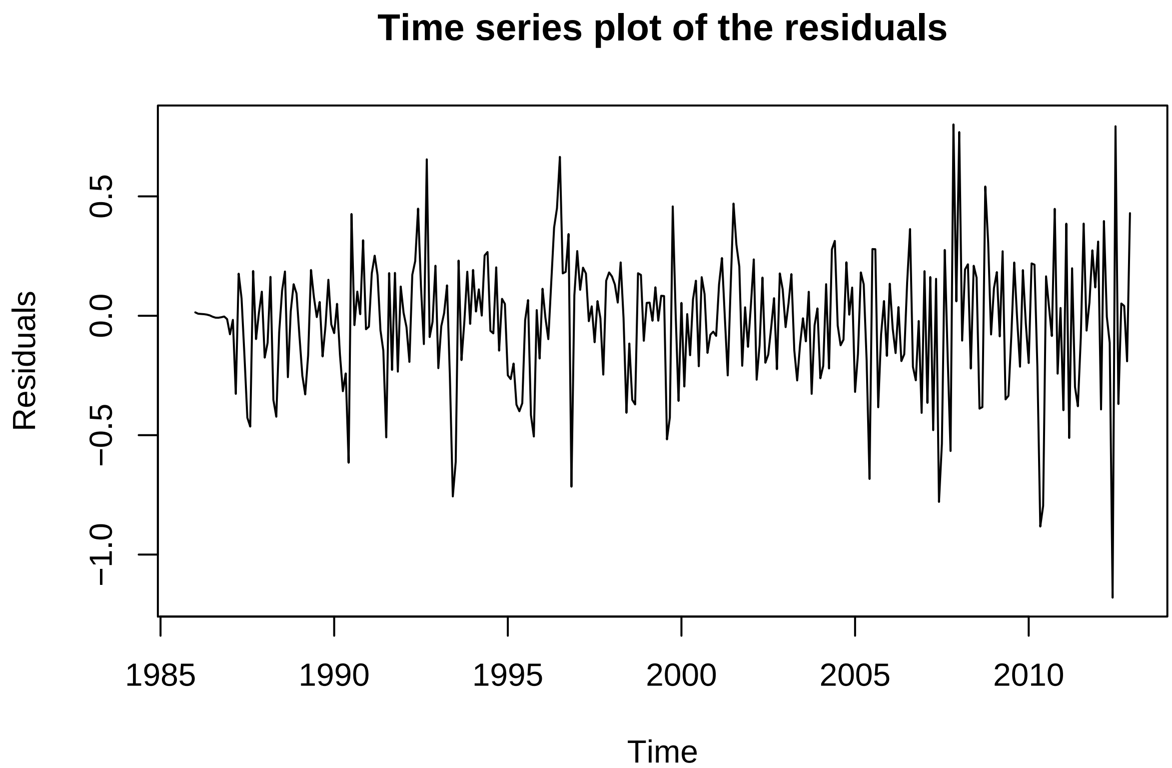
```
par(mfrow=c(1,2))
acf(res_m4, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m4, lag.max = 36, main = "The sample PACF of the residuals")
```





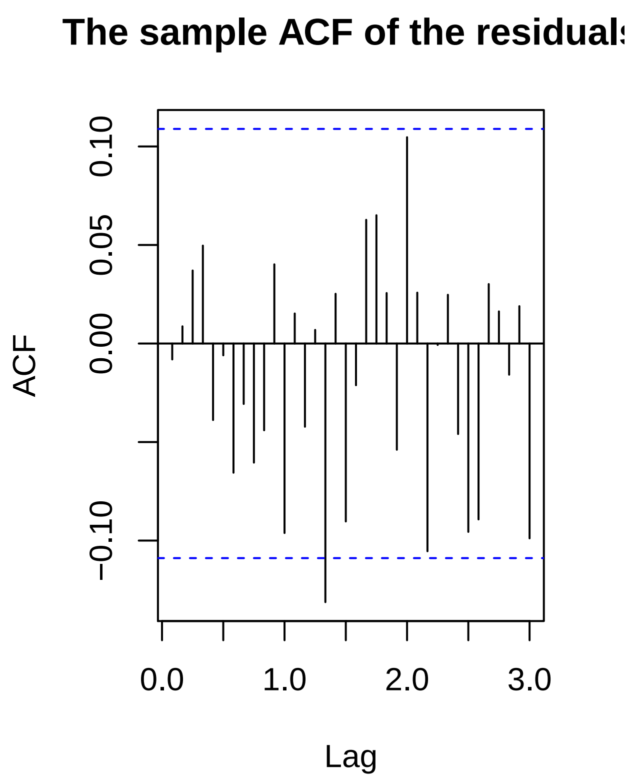
3.1.15 Fit of SARIMA(1,1,3)x$(0,1,1)_{12}$ model

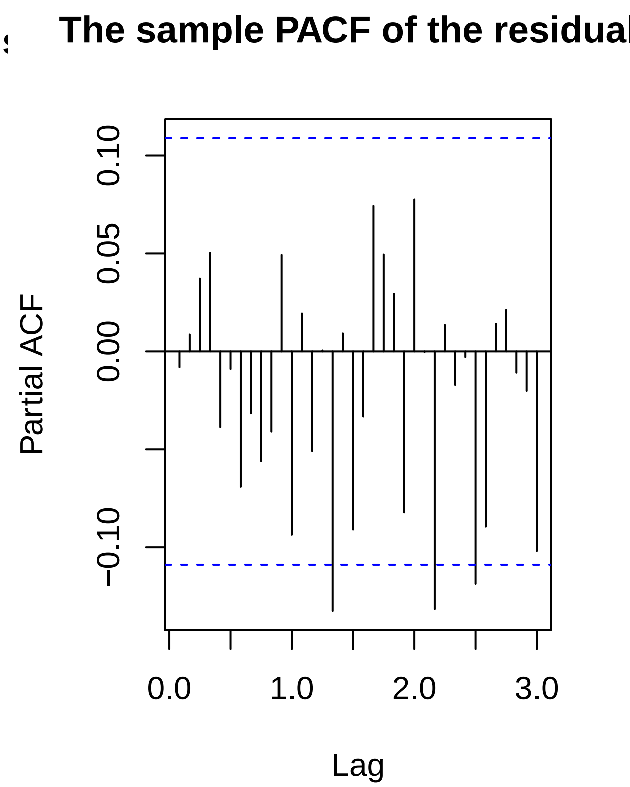
```
m5_sea_ice = arima(sea_ice_ts,order=c(1,1,3),seasonal=list(order=c(0,1,1), period=12))
res_m5 = residuals(m5_sea_ice);
plot(res_m5,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.16 ACF and PACF plots for SARIMA(1,1,3)x(0,1,1)$_{12}$ model

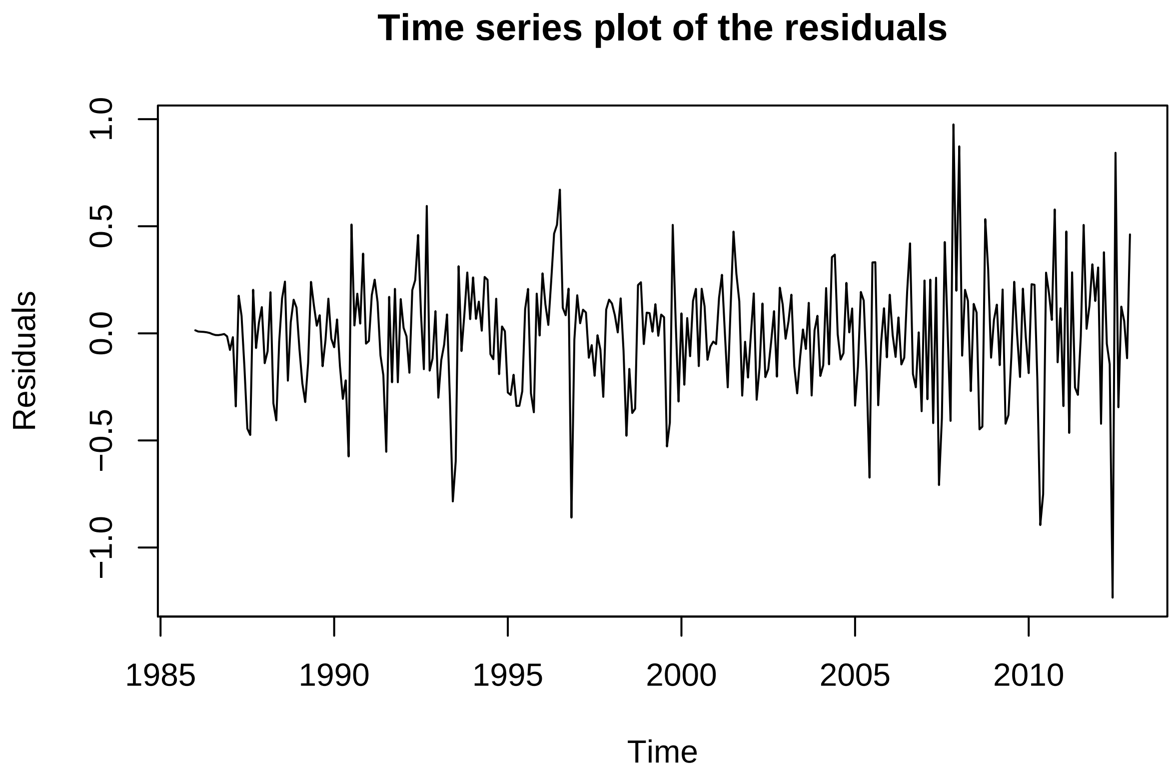
```
par(mfrow=c(1,2))
acf(res_m5, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m5, lag.max = 36, main = "The sample PACF of the residuals")
```





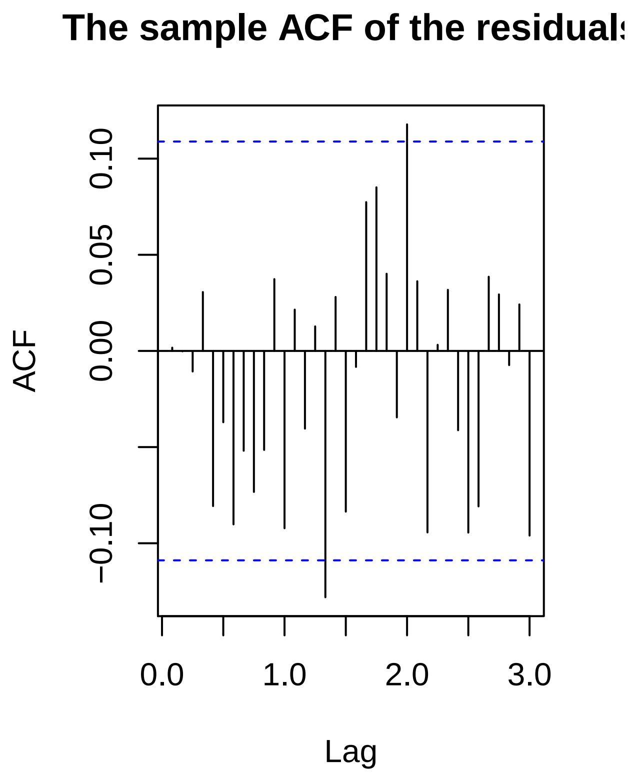
3.1.17 Fit of SARIMA(2,1,1)x(0,1,1)$_{12}$ model

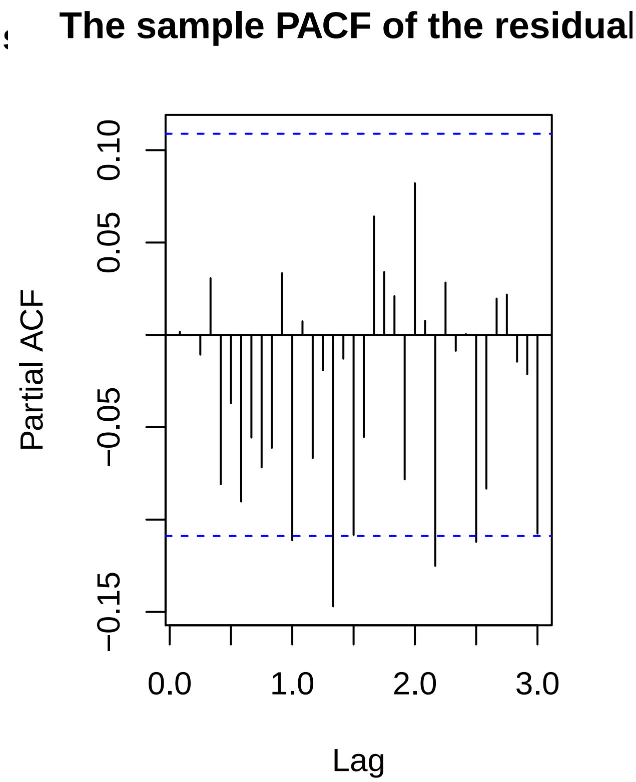
```
m6_sea_ice = arima(sea_ice_ts,order=c(2,1,1),seasonal=list(order=c(0,1,1), period=12))
res_m6 = residuals(m6_sea_ice);
plot(res_m6,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.18 ACF and PACF plots for SARIMA(2,1,1)x(0,1,1)$_{12}$ model

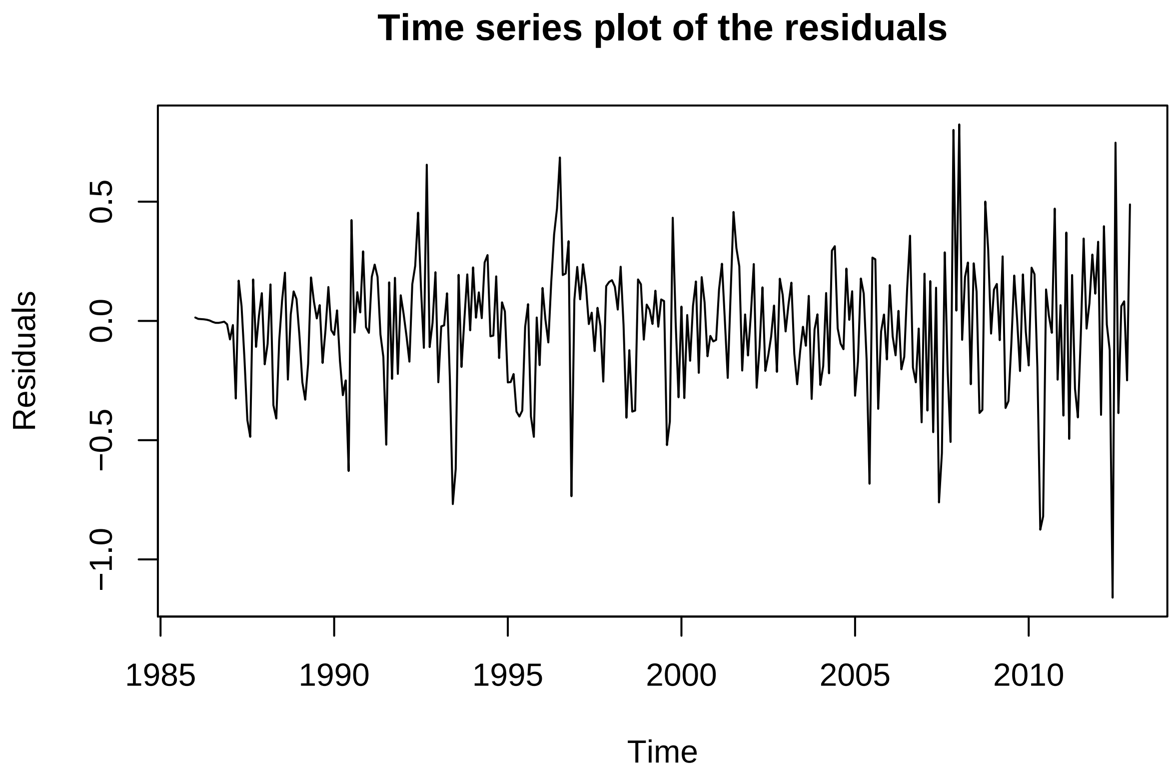
```
par(mfrow=c(1,2))
acf(res_m6, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m6, lag.max = 36, main = "The sample PACF of the residuals")
```





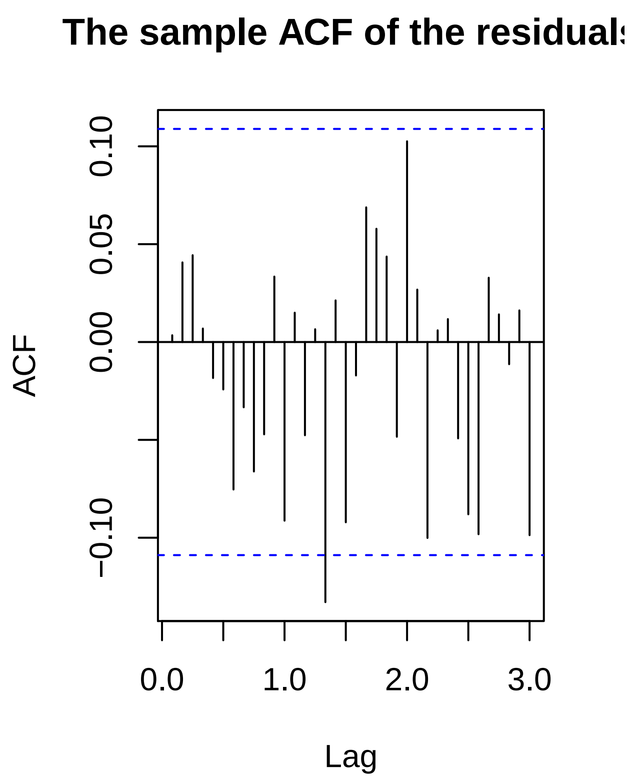
3.1.19 Fit of SARIMA(2,1,3)x(0,1,1)$_{12}$ model

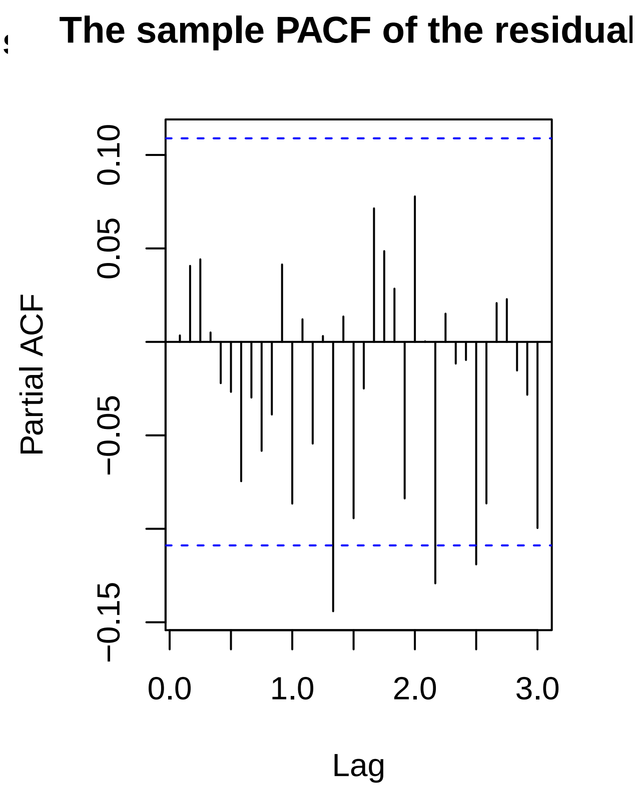
```
m7_sea_ice = arima(sea_ice_ts,order=c(2,1,3),seasonal=list(order=c(0,1,1), period=12))
res_m7 = residuals(m7_sea_ice);
plot(res_m7,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.20 ACF and PACF plots for SARIMA(2,1,3)x(0,1,1)$_{12}$ model

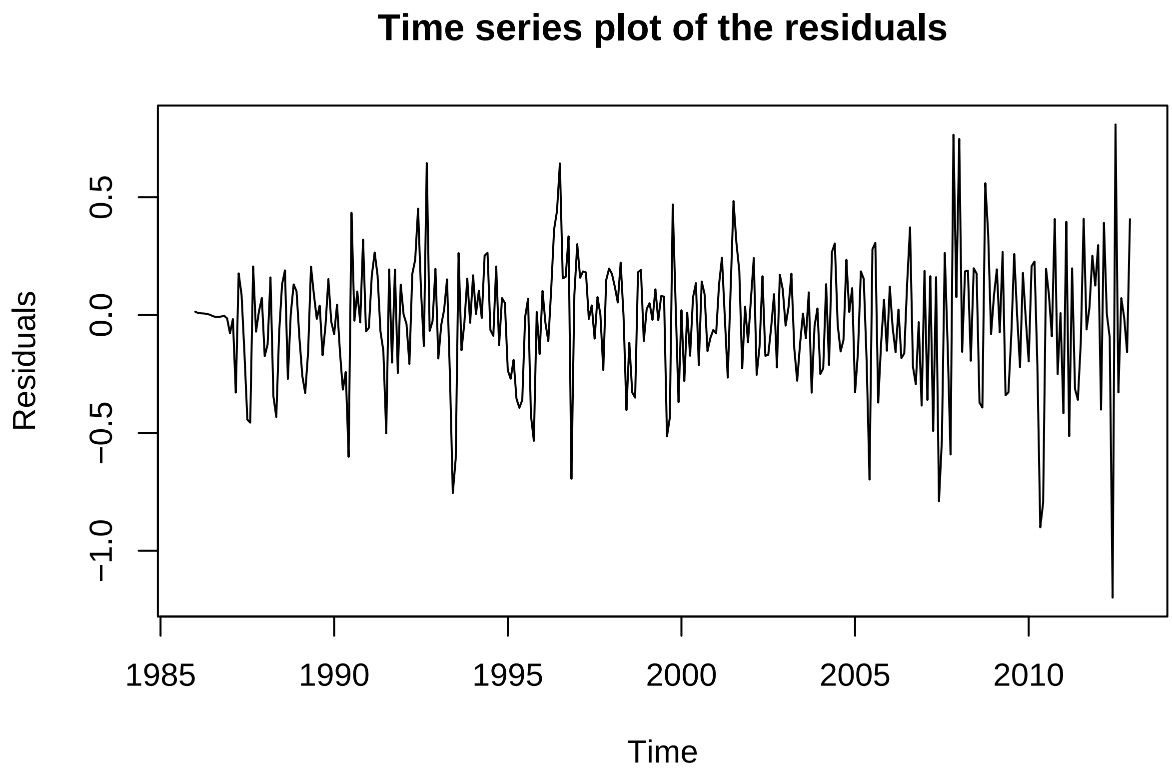
```
par(mfrow=c(1,2))
acf(res_m7, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m7, lag.max = 36, main = "The sample PACF of the residuals")
```





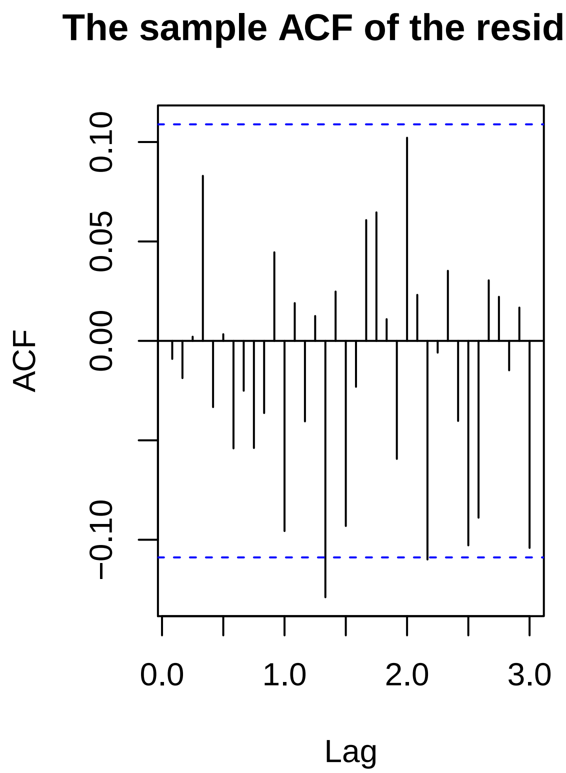
3.1.21 Fit of SARIMA(2,1,2)x$(0,1,1)_{12}$ model

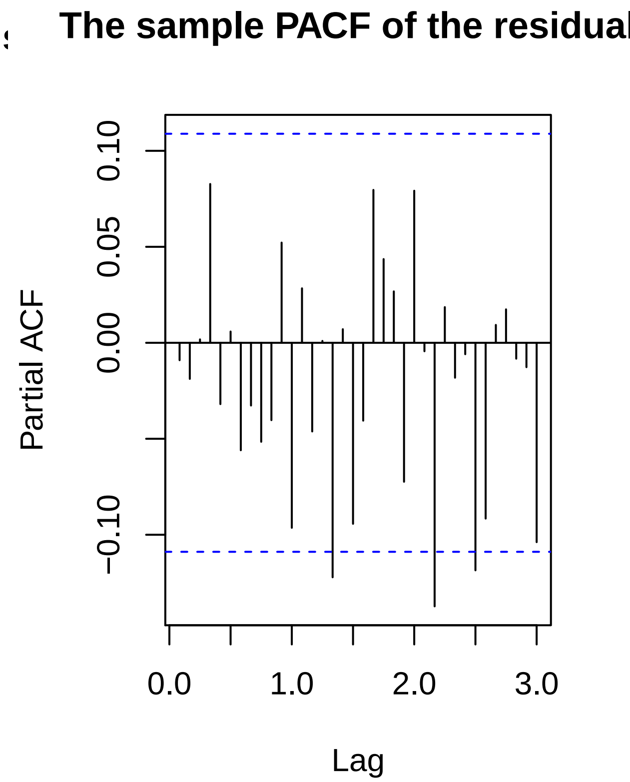
```
m8_sea_ice = arima(sea_ice_ts,order=c(2,1,2),seasonal=list(order=c(0,1,1), period=12))
res_m8 = residuals(m8_sea_ice);
plot(res_m8,xlab='Time',ylab='Residuals',main="Time series plot of the residuals")
```



3.1.22 ACF and PACF plots for SARIMA(2,1,2)x(0,1,1)$_{12}$ model

```
par(mfrow=c(1,2))
acf(res_m8, lag.max = 36, main = "The sample ACF of the residuals")
pacf(res_m8, lag.max = 36, main = "The sample PACF of the residuals")
```





- The residuals for the SARIMA(1,1,2)x(0,1,1)$_{12}$ model are closer to white noise. However, there are a number of significant autocorrelations in the ACF and PACF plots of the residuals which will be considered in more detail latter.
- Therefore we can conclude that the orders are: p=1, d=1, q=2, P=0, D=1, Q=1 and s=12 for the SARIMA(p,d,q)x(P,D,Q)$_s$ model.

3.2 Model Fitting - ML estimates and Conditional Least Squares for SARIMA models.

3.2.1 ML estimates for SARIMA(0,1,1)x(0,1,1)$_{12}$ model

```
m1_sea_ice_ts = arima(sea_ice_ts,order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))
coeftest(m1_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.533392   0.041952  12.714 < 2.2e-16 ***
## sma1 -0.640394   0.044802 -14.294 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.2 ML estimates for SARIMA(0,1,2)x(0,1,1)$_{12}$ model

```
m2_sea_ice_ts = arima(sea_ice_ts,order=c(0,1,2), seasonal=list(order=c(0,1,1), period=12))
coeftest(m2_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ma1   0.613225   0.059311  10.3392  < 2e-16 ***
## ma2   0.124154   0.056944   2.1803  0.02924 *
## sma1 -0.636502   0.045190 -14.0851  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.3 ML estimates for SARIMA(1,1,1)x(0,1,1)$_{12}$ model

```
m3_sea_ice_ts = arima(sea_ice_ts,order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12))
coeftest(m3_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.178782   0.090896   1.9669    0.0492 *
## ma1   0.417993   0.080069   5.2204 1.786e-07 ***
## sma1 -0.636858   0.045063 -14.1327 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.4 ML estimates for SARIMA(1,1,2)x(0,1,1)$_{12}$ model

```
m4_sea_ice_ts = arima(sea_ice_ts,order=c(1,1,2), seasonal=list(order=c(0,1,1), period=12))
coeftest(m4_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.647143   0.197411  -3.2782  0.001045 **
## ma1   1.260159   0.195445   6.4476 1.136e-10 ***
## ma2   0.477516   0.110513   4.3209 1.554e-05 ***
## sma1 -0.639266   0.046287 -13.8109 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.5 ML estimates for SARIMA(1,1,3)x(0,1,1)$_{12}$ model

```
m5_sea_ice_ts = arima(sea_ice_ts,order=c(1,1,3), seasonal=list(order=c(0,1,1), period=12))
coeftest(m5_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.842707   0.040071  21.0305 < 2.2e-16 ***
## ma1  -0.297435   0.067382  -4.4142 1.014e-05 ***
## ma2  -0.487598   0.049078  -9.9352 < 2.2e-16 ***
## ma3  -0.190600   0.059194  -3.2199  0.001282 **
## sma1 -0.636035   0.045983 -13.8319 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.6 ML estimates for SARIMA(2,1,1)x(0,1,1)$_{12}$ model

```
m6_sea_ice_ts = arima(sea_ice_ts,order=c(2,1,1), seasonal=list(order=c(0,1,1), period=12))
coeftest(m6_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1   0.356707   0.187181   1.9057  0.05669 .
## ar2  -0.129080   0.108771  -1.1867  0.23534
## ma1   0.246321   0.184888   1.3323  0.18277
## sma1 -0.638723   0.045102 -14.1616  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.7 ML estimates for SARIMA(2,1,3)x(0,1,1)$_{12}$ model

```
m7_sea_ice_ts = arima(sea_ice_ts,order=c(2,1,3), seasonal=list(order=c(0,1,1), period=12))
coeftest(m7_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1   0.484978   2.111859   0.2296   0.8184
## ar2   0.323860   1.924049   0.1683   0.8663
## ma1   0.055727   2.062558   0.0270   0.9784
## ma2  -0.648039   0.972311  -0.6665   0.5051
## ma3  -0.378331   1.064944  -0.3553   0.7224
## sma1 -0.637609   0.047261 -13.4912   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.8 ML estimates for SARIMA(2,1,2)x(0,1,1)$_{12}$ model

```
m8_sea_ice_ts = arima(sea_ice_ts,order=c(2,1,2), seasonal=list(order=c(0,1,1), period=12))
coeftest(m8_sea_ice_ts)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   1.177434   0.091309  12.8951 < 2.2e-16 ***
## ar2  -0.298959   0.089459  -3.3419 0.0008322 ***
## ma1  -0.638129   0.089610  -7.1212  1.07e-12 ***
## ma2  -0.342867   0.088469  -3.8756 0.0001064 ***
## sma1 -0.635259   0.045760 -13.8824 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

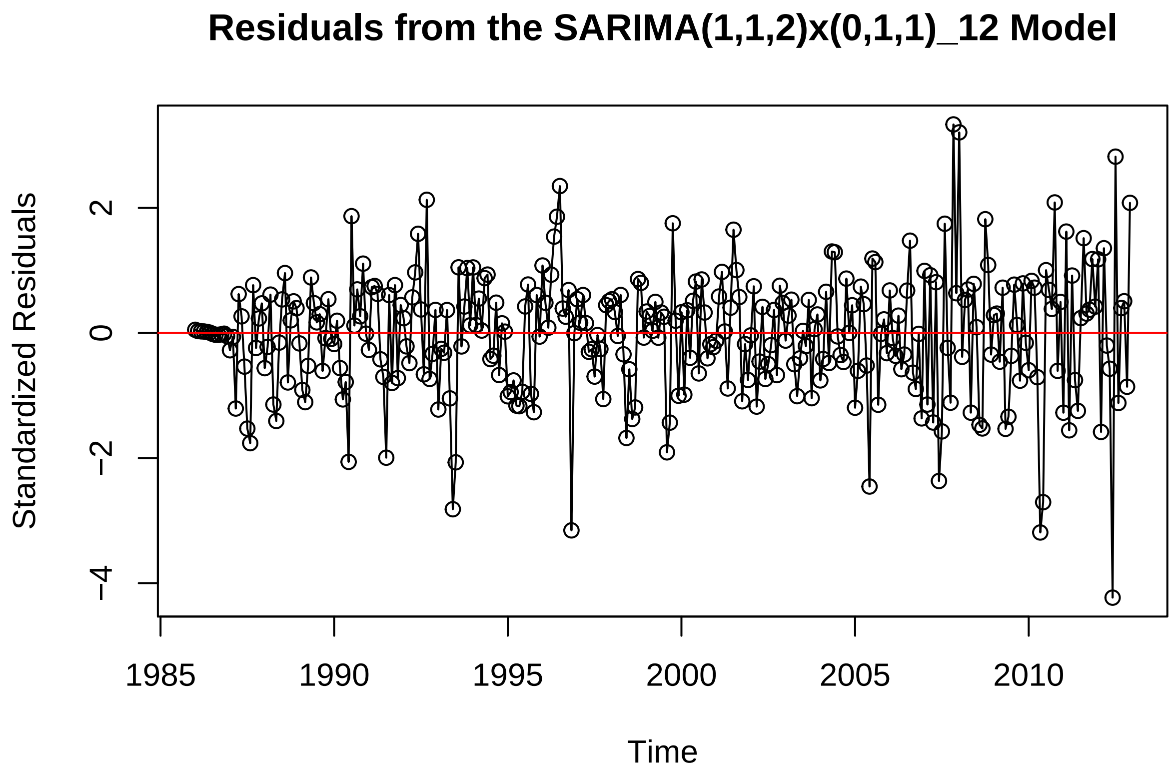
- The Maximum likelihood (ML) estimates were determined to be highly significant for SARIMA(0,1,1)x$(0,1,1)_{12}$, SARIMA(0,1,2)x$(0,1,1)_{12}$, SARIMA(1,1,1)x$(0,1,1)_{12}$, SARIMA(1,1,2)x$(0,1,1)_{12}$ models, SARIMA(1,1,3)x$(0,1,1)_{12}$ models and SARIMA(2,1,2)x$(0,1,1)_{12}$ models.
- The best model was SARIMA(1,1,2)x$(0,1,1)_{12}$. The ACF and PACF plots were mostly white noise but contained some significant residuals.

3.3 Diagnostic Check of models

3.3.1 SARIMA(1,1,2)x$(0,1,1)_{12}$ model

3.3.2 Time series plot for standardized residuals.

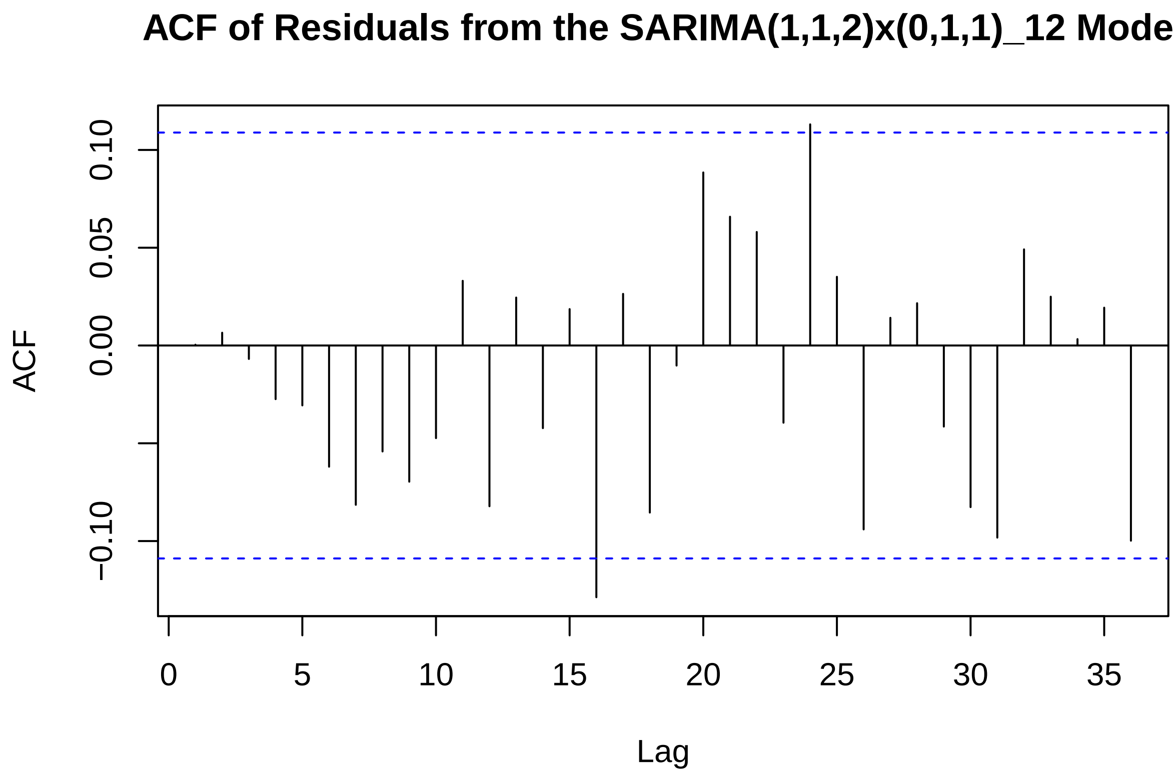
```
plot(window(rstandard(m4_sea_ice),start=c(1986, 1)),
ylab='Standardized Residuals',type='o',
main="Residuals from the SARIMA(1,1,2)x(0,1,1)_12 Model")
abline(h=0, col = 'red')
```



- Plot suggests no major abnormalities with this model(except around 1995) although there are several outliers that may need to be investigated in more detail.

3.3.3 ACF plot of standardized residuals.

```
acf(as.vector(window(rstandard(m4_sea_ice),start=c(1986,1))),
lag.max=36,
main="ACF of Residuals from the SARIMA(1,1,2)x(0,1,1)_12 Model")
```



- Besides the slightly significant autocorrelations at lag 16 there is no sign of violation of the independence of residuals.

3.3.3.1 Box-Ljung test.

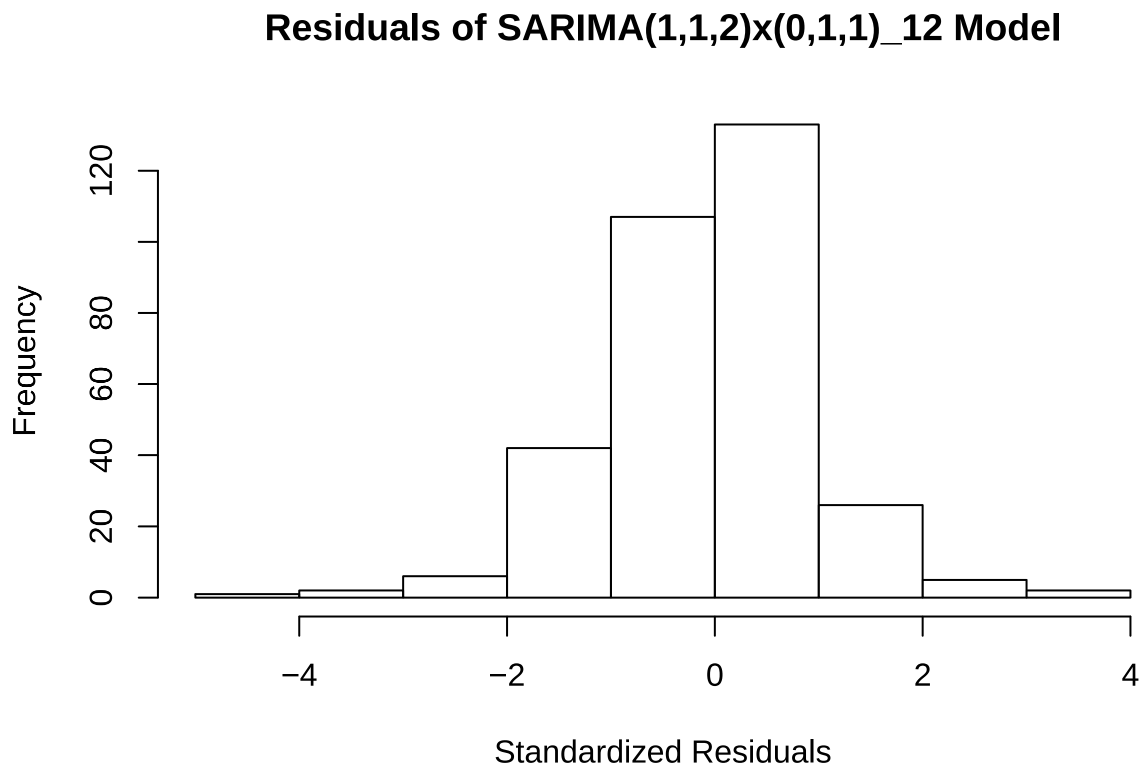
```
Box.test(window(rstandard(m4_sea_ice), start = c(1986,1)), lag = 16, type = "Ljung-Box", fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  window(rstandard(m4_sea_ice), start = c(1986, 1))
## X-squared = 16.708, df = 16, p-value = 0.4047
```

- Overall, the Ljung-Box test indicates that there is no problem in terms of independence of errors.

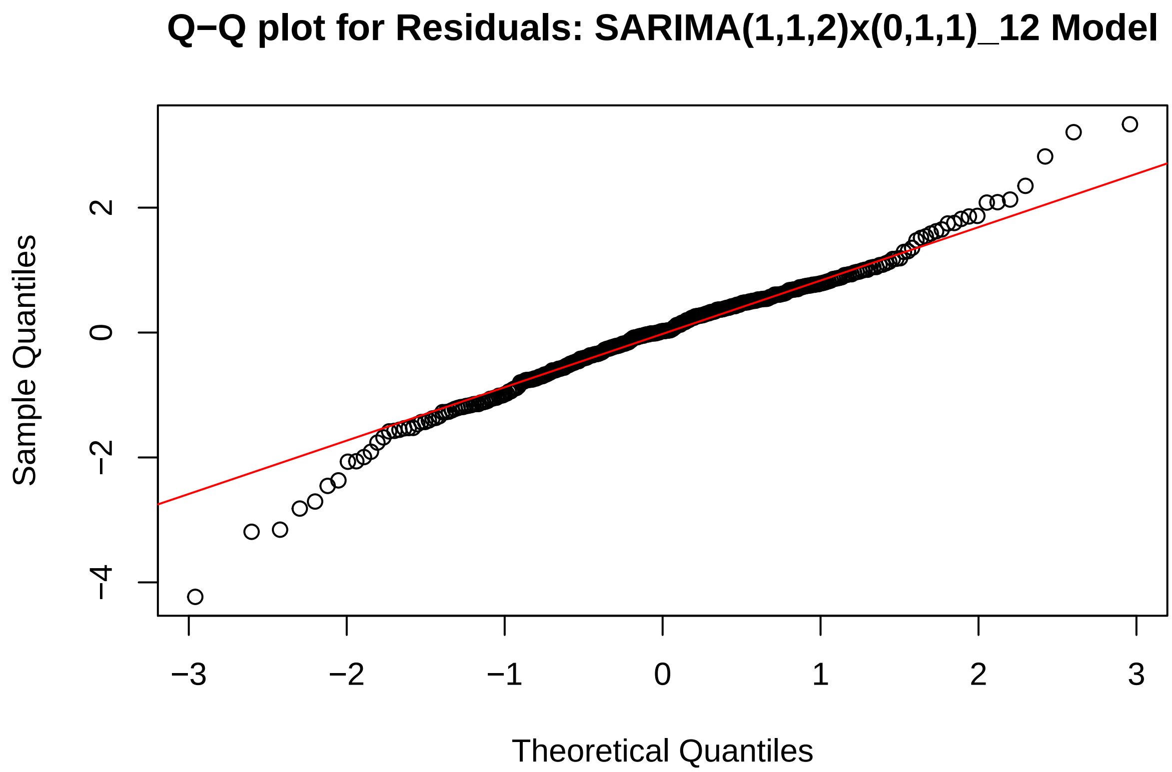
3.3.3.2 Histogram of standardized residuals.

```
hist(window(rstandard(m4_sea_ice),start=c(1986,1)), xlab = 'Standardized Residuals', ylab = 'Frequency'
```



3.3.3.3 Q-Q plot of standardized residuals.

```
qqnorm(window(rstandard(m4_sea_ice),start=c(1986,1)),main="Q-Q plot for Residuals: SARIMA(1,1,2)x(0,1,1)
qqline(window(rstandard(m4_sea_ice),start=c(1986,1)), col=2)
```



3.3.3.4 Shapiro-Wilk test for normality of standardized residuals.

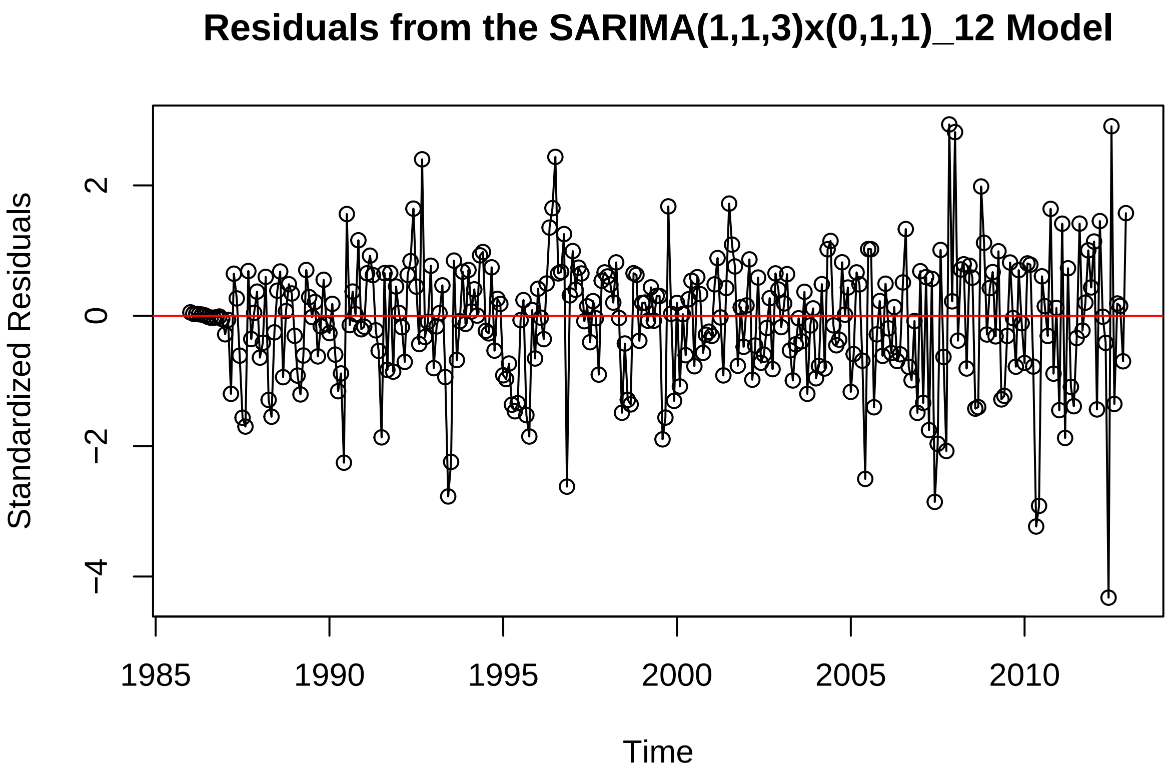
```
shapiro.test(window(rstandard(m4_sea_ice), start=c(1986,1), end=c(2012,12)))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  window(rstandard(m4_sea_ice), start = c(1986, 1), end = c(2012,     12))
## W = 0.97724, p-value = 5.201e-05
```

3.3.4 SARIMA(1,1,3)x(0,1,1)$_{12}$ model

3.3.4.1 Time series plot for standardized residuals.

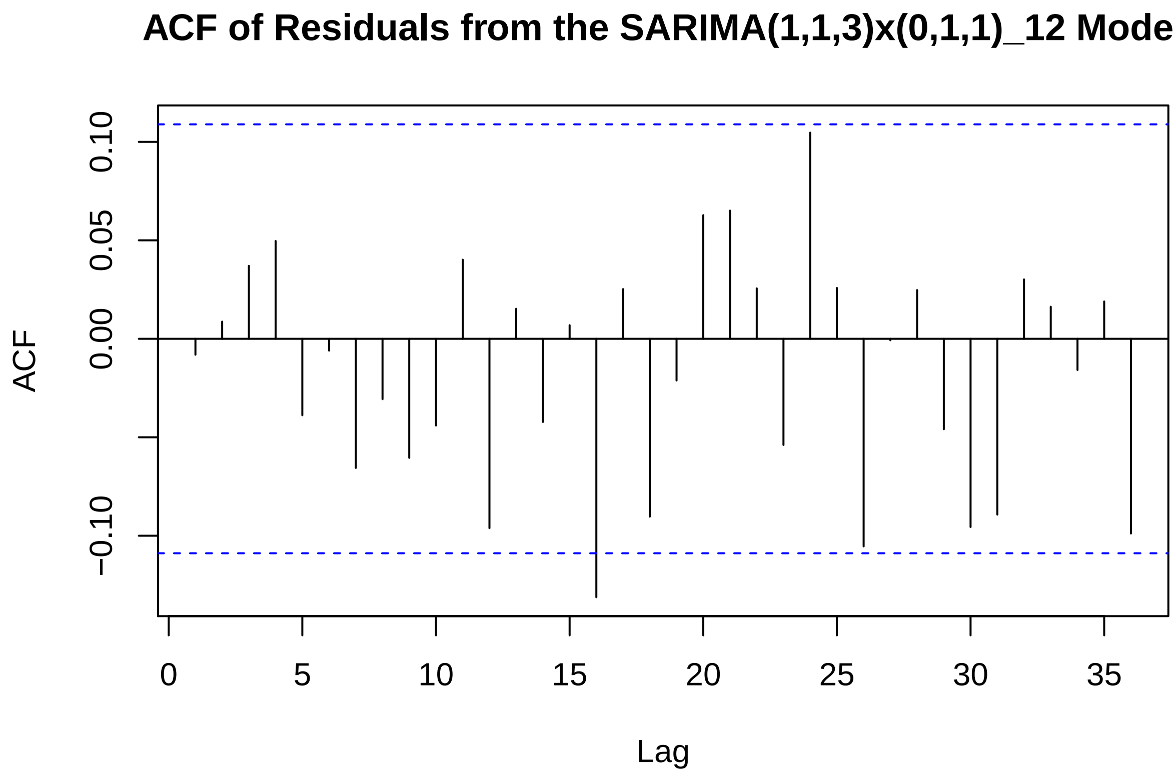
```
plot(window(rstandard(m5_sea_ice),start=c(1986, 1)),
ylab='Standardized Residuals',type='o',
main="Residuals from the SARIMA(1,1,3)x(0,1,1)_12 Model")
abline(h=0, col = 'red')
```



- Plot suggests no major abnormalities with this model(except around 1995) although there are several outliers that may need to be investigated in more detail.

3.3.4.2 ACF plot of standardized residuals.

```
acf(as.vector(window(rstandard(m5_sea_ice),start=c(1986,1))),
lag.max=36,
main="ACF of Residuals from the SARIMA(1,1,3)x(0,1,1)_12 Model")
```



- Besides the slightly significant autocorrelations at lag 16 there is no sign of violation of the independence of residuals.

3.3.4.3 Box-Ljung test.

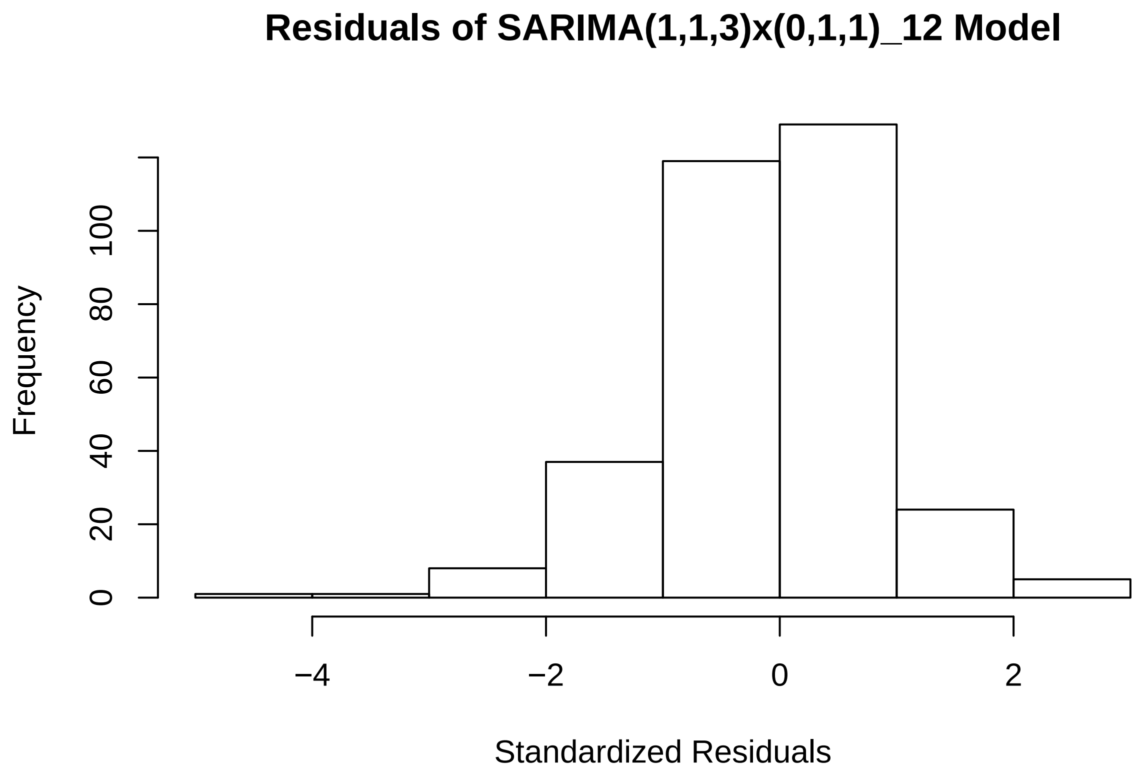
```
Box.test(window(rstandard(m5_sea_ice), start = c(1986,1)), lag = 16, type = "Ljung-Box", fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  window(rstandard(m5_sea_ice), start = c(1986, 1))
## X-squared = 15.733, df = 16, p-value = 0.4717
```

- Overall, the Ljung-Box test indicates that there is no problem in terms of independence of errors.

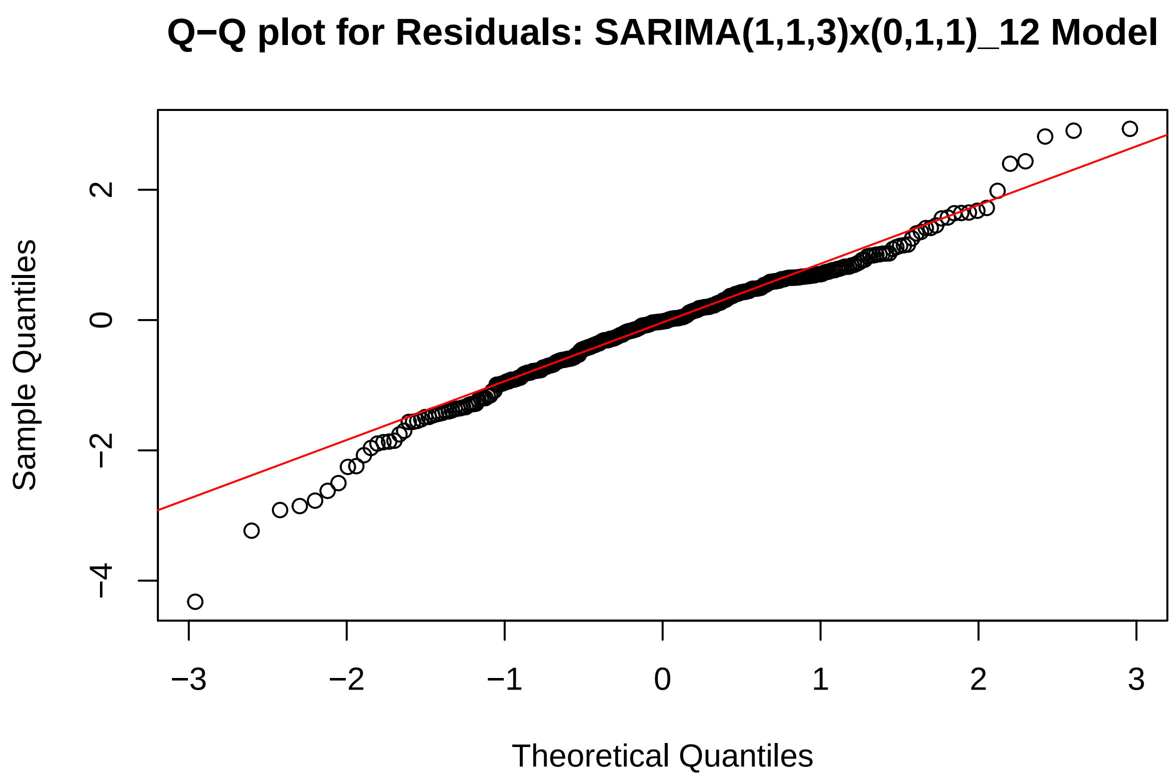
3.3.4.4 Histogram of standardized residuals.

```
hist(window(rstandard(m5_sea_ice),start=c(1986,1)), xlab = 'Standardized Residuals', ylab = 'Frequency'
```



3.3.4.5 Q-Q plot of standardized residuals.

```
qqnorm(window(rstandard(m5_sea_ice),start=c(1986,1)),main="Q-Q plot for Residuals: SARIMA(1,1,3)x(0,1,1
qqline(window(rstandard(m5_sea_ice),start=c(1986,1)), col=2)
```



3.3.4.6 Shapiro-Wilk test for normality of standardized residuals.

```
shapiro.test(window(rstandard(m5_sea_ice), start=c(1986,1), end=c(2012,12)))
```

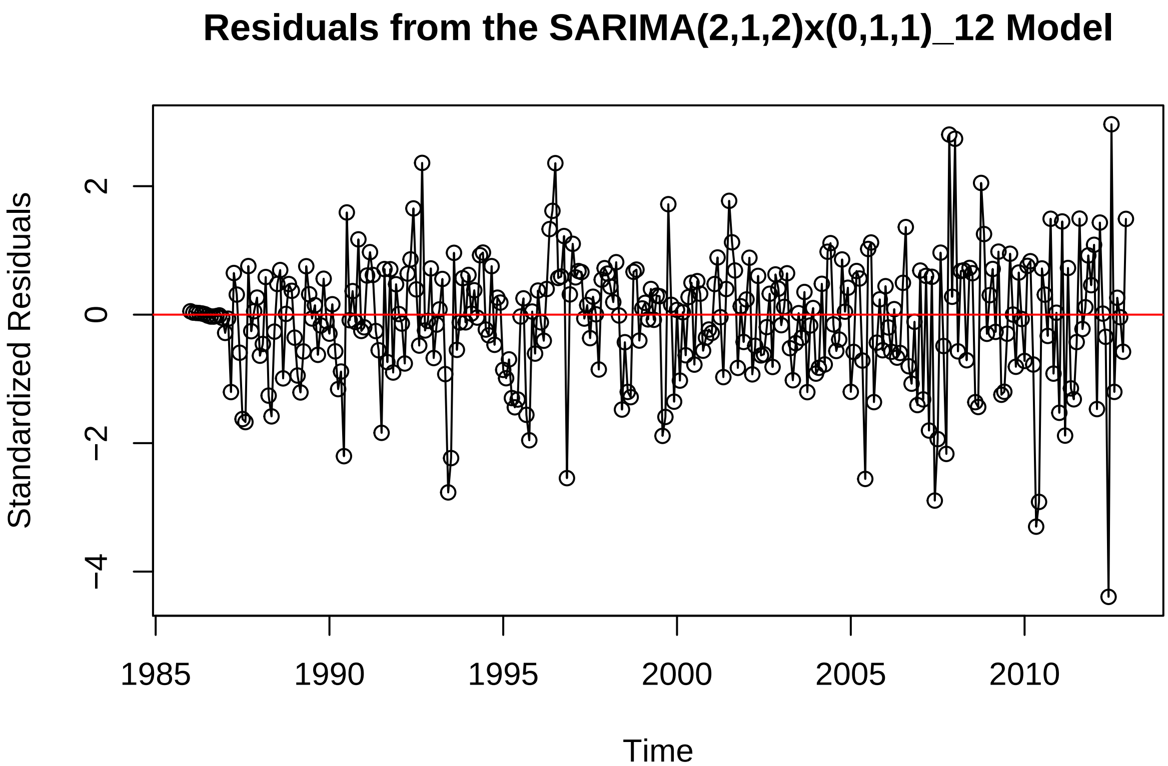
```
##
##  Shapiro-Wilk normality test
##
## data:  window(rstandard(m5_sea_ice), start = c(1986, 1), end = c(2012,     12))
## W = 0.97567, p-value = 2.698e-05
```

- Although we have mostly white noise residuals, the large-valued residuals make it impossible to conclude the normality of residuals by either the Q-Q plot or Shapiro test at 5% level of significance.

3.3.5 SARIMA(2,1,2)x(0,1,1)$_{12}$ model

3.3.5.1 Time series plot for standardized residuals.

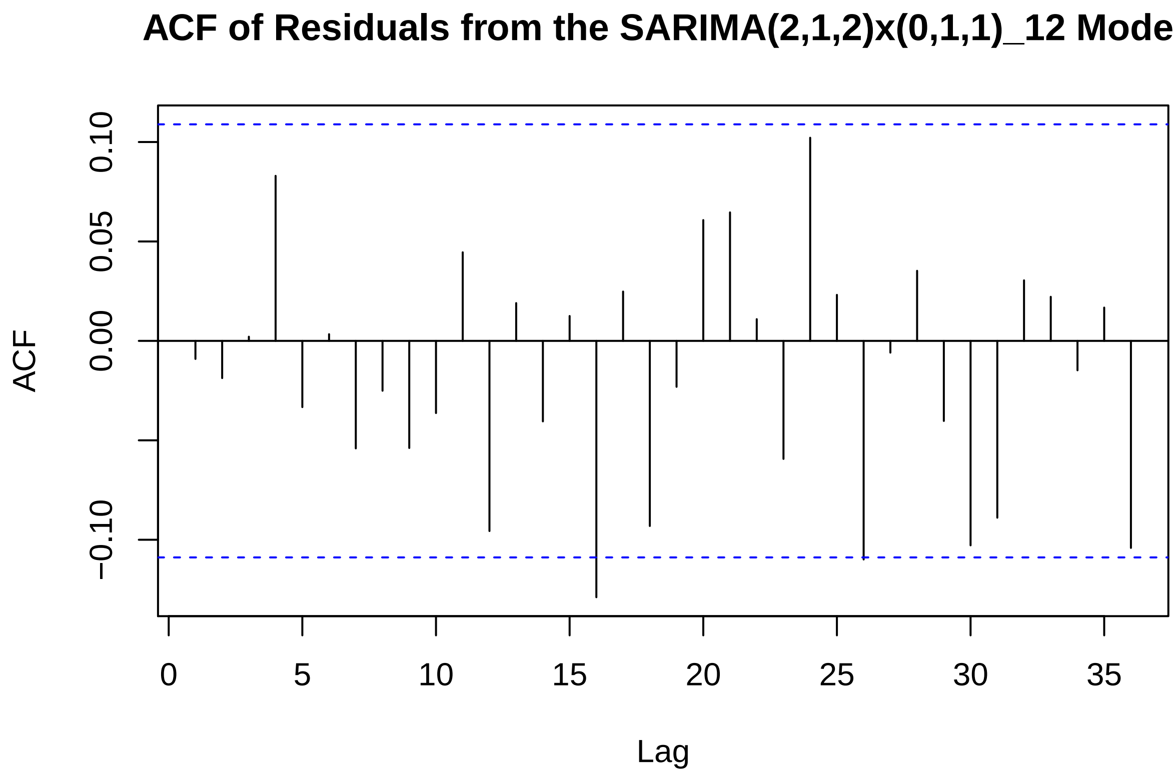
```
plot(window(rstandard(m8_sea_ice),start=c(1986, 1)),
ylab='Standardized Residuals',type='o',
main="Residuals from the SARIMA(2,1,2)x(0,1,1)_12 Model")
abline(h=0, col = 'red')
```



- Plot suggests no major abnormalities with this model(except around 1995) although there are several outliers that may need to be investigated in more detail.

3.3.5.2 ACF plot of standardized residuals.

```
acf(as.vector(window(rstandard(m8_sea_ice),start=c(1986,1))),
lag.max=36,
main="ACF of Residuals from the SARIMA(2,1,2)x(0,1,1)_12 Model")
```



- Besides the slightly significant autocorrelations at lag 16 there is no sign of violation of the independence of residuals.

3.3.5.3 Box-Ljung test.

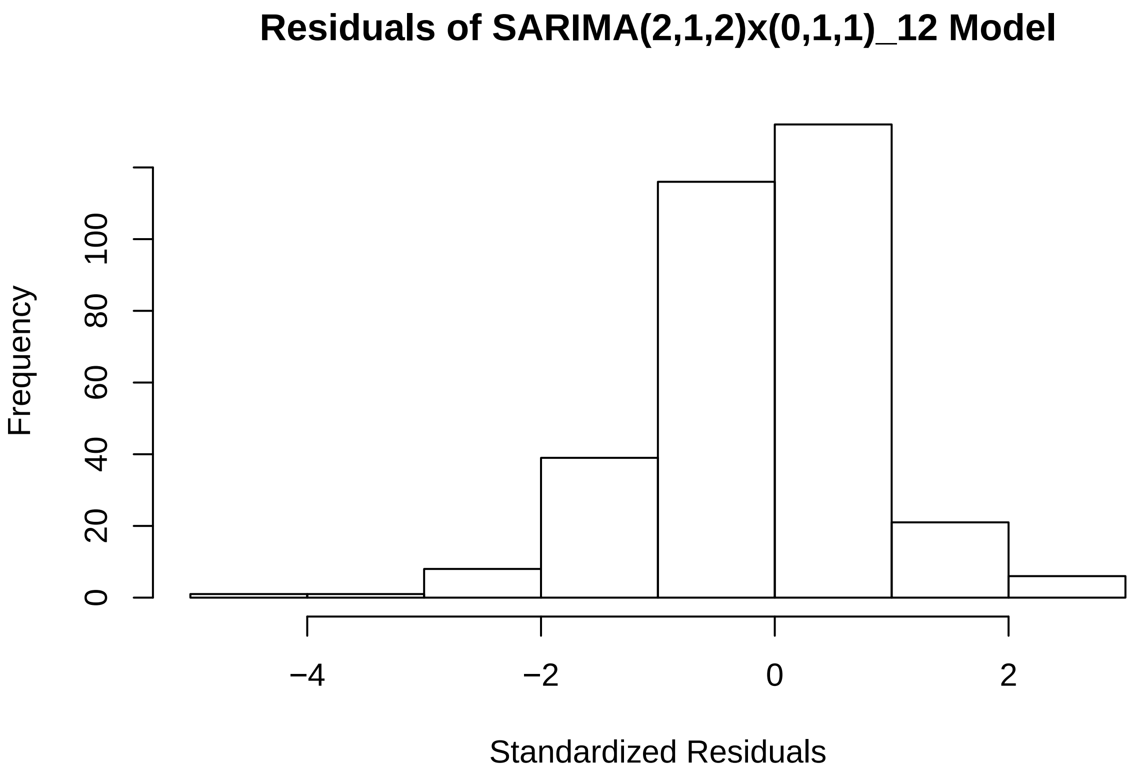
```
Box.test(window(rstandard(m8_sea_ice), start = c(1986,1)), lag = 16, type = "Ljung-Box", fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  window(rstandard(m8_sea_ice), start = c(1986, 1))
## X-squared = 15.587, df = 16, p-value = 0.4821
```

- Overall, the Ljung-Box test indicates that there is no problem in terms of independence of errors.

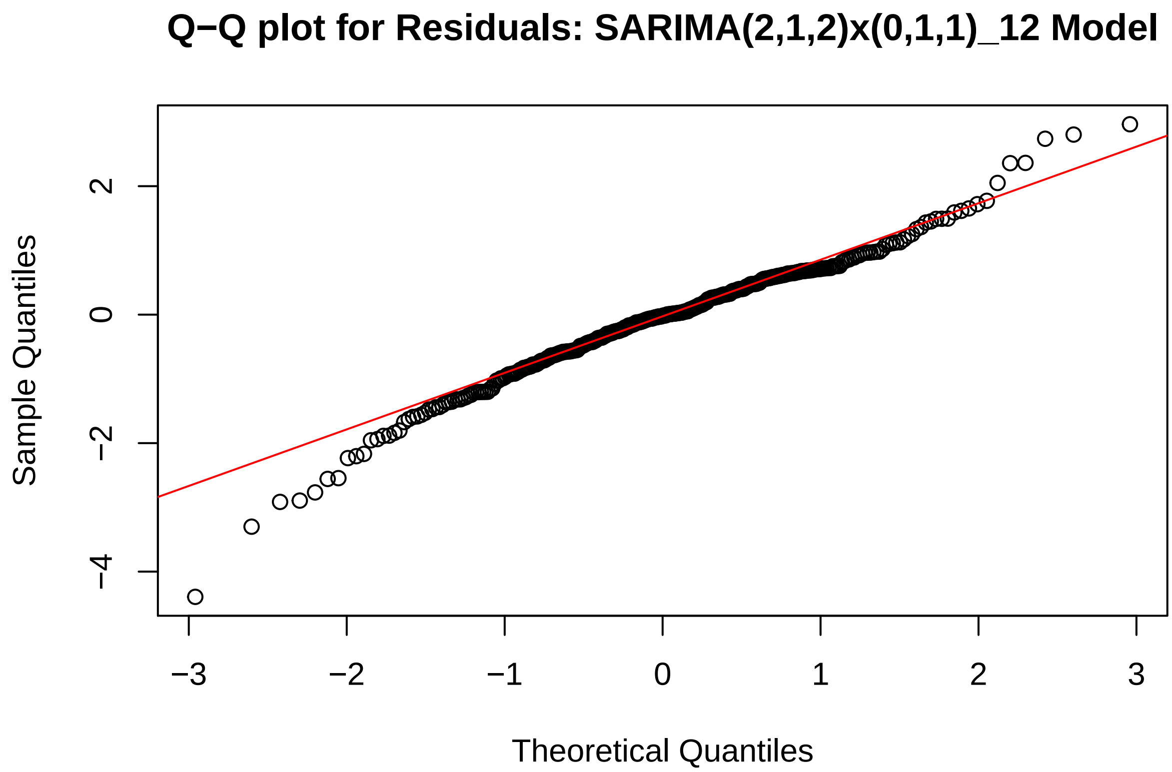
3.3.5.4 Histogram of standardized residuals.

```
hist(window(rstandard(m8_sea_ice),start=c(1986,1)), xlab = 'Standardized Residuals', ylab = 'Frequency'
```



3.3.5.5 Q-Q plot of standardized residuals.

```
qqnorm(window(rstandard(m8_sea_ice),start=c(1986,1)),main="Q-Q plot for Residuals: SARIMA(2,1,2)x(0,1,1)
qqline(window(rstandard(m8_sea_ice),start=c(1986,1)), col=2)
```



3.3.5.6 Shapiro-Wilk test for normality of standardized residuals.

```
shapiro.test(window(rstandard(m8_sea_ice), start=c(1986,1), end=c(2012,12)))
```

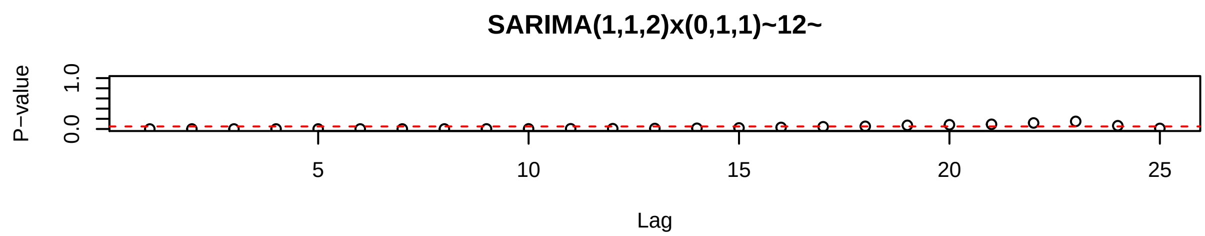
```
##
##  Shapiro-Wilk normality test
##
## data:  window(rstandard(m8_sea_ice), start = c(1986, 1), end = c(2012,     12))
## W = 0.97615, p-value = 3.299e-05
```

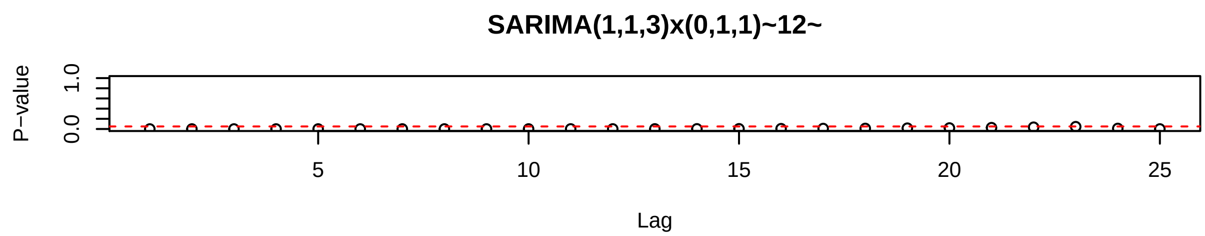
- Although we have mostly white noise residuals, the large-valued residuals make it impossible to conclude the normality of residuals by either the Q-Q plot or Shapiro test at 5% level of significance.
- The identified SARIMA models contained a number of slightly significant autocorrelations for their residuals and the residuals did not appear to normally distributed. Therefore we decided to check for the presence of an ARCH component.

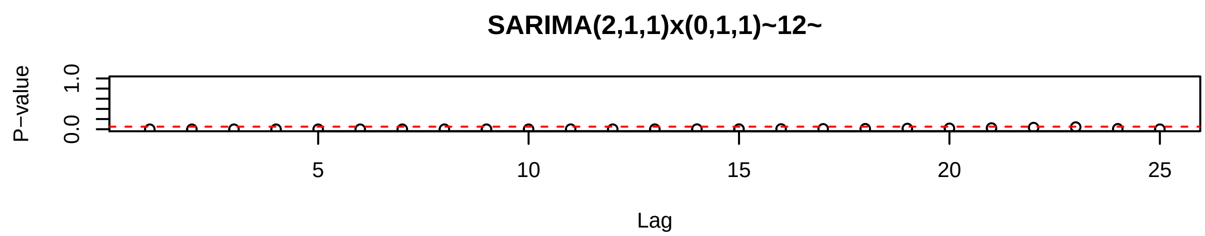
3.3.6 Check for ARCH component in residuals of SARIMA model.

3.3.6.1 McLeod-Li test and Q-Q plot for the SARIMA(1,1,2)x$(0,1,1)_{12}$ model.

```
par(mfrow=c(3,1))
McLeod.Li.test(y=res_m4, main = 'SARIMA(1,1,2)x(0,1,1)~12~')
McLeod.Li.test(y=res_m5, main = 'SARIMA(1,1,3)x(0,1,1)~12~')
McLeod.Li.test(y=res_m8, main = 'SARIMA(2,1,1)x(0,1,1)~12~')
```







- McLeod-Li tests are mostly highly significant and the normality assumption is highly violated.

3.4 Consider overfitting (compare with SARIMA models SARIMA(1,1,3)x$(0,1,1)_{12}$ and SARIMA(2,1,2)x$(0,1,1)_{12}$)

3.4.1 ML estimates for SARIMA(1,1,4)x$(0,1,1)_{12}$ model

```
m9_sea_ice = arima(sea_ice_ts,order=c(1,1,4), seasonal=list(order=c(0,1,1), period=12))
coeftest(m9_sea_ice)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.621139   0.247964  -2.5050  0.012247 *
## ma1   1.235405   0.253429   4.8748 1.089e-06 ***
## ma2   0.467886   0.178312   2.6240  0.008691 **
## ma3   0.016506   0.094981   0.1738  0.862040
## ma4   0.020480   0.071640   0.2859  0.774978
## sma1 -0.638814   0.046483 -13.7430 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.4.2 ML estimates for SARIMA(3,1,3)x(0,1,1)$_{12}$ model

```
m10_sea_ice = arima(sea_ice_ts,order=c(3,1,3), seasonal=list(order=c(0,1,1), period=12))
coeftest(m10_sea_ice)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.359364   0.201405   1.7843  0.074378 .
## ar2   0.555213   0.199476   2.7834  0.005380 **
## ar3  -0.133740   0.110446  -1.2109  0.225932
## ma1   0.193508   0.200443   0.9654  0.334344
## ma2  -0.776139   0.108064  -7.1822 6.858e-13 ***
## ma3  -0.383398   0.131857  -2.9077  0.003641 **
## sma1 -0.636961   0.046565 -13.6790 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- It was not possible to get significant coefficients in the over-fitted model indicating that SARIMA(1,1,2)x(0,1,1)$_{12}$ is the most suitable model.

3.5 Comparison of AIC values for differnt SARIMA models.

- m4_sea_ice_ts SARIMA(1,1,2)x(0,1,1)$_{12}$, m5_sea_ice_ts SARIMA(1,1,3)x(0,1,1)$_{12}$, m8_sea_ice_ts SARIMA(2,1,2)x(0,1,1)$_{12}$

```
AIC(m4_sea_ice_ts, m5_sea_ice_ts, m8_sea_ice_ts)
```

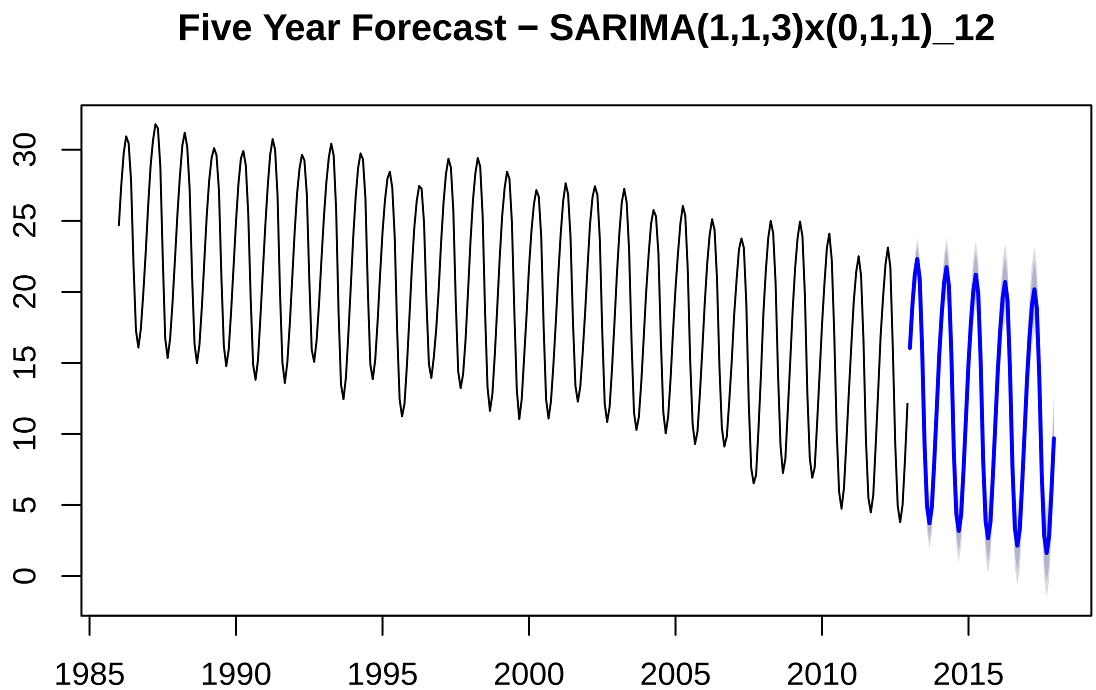
```
##               df       AIC
## m4_sea_ice_ts  5 108.83175
## m5_sea_ice_ts  6  96.29905
## m8_sea_ice_ts  6  96.18930
```

- Values obtained from the AIC analysis indicate that the best models are SARIMA(1,1,3)x(0,1,1)$_{12}$ and SARIMA(2,1,2)x(0,1,1)$_{12}$.

3.6 Prediction of Seasonal Arctic Sea-Ice

3.6.1 Five year forecast for SARIMA(1,1,3)x(0,1,1)$_{12}$.

```
sea_ice_for_113 = Arima(sea_ice_ts,order=c(1,1,3),seasonal=list(order=c(0,1,1), period=12))

future_113 = forecast(sea_ice_for_113, h = 60)
plot(future_113, main = 'Five Year Forecast - SARIMA(1,1,3)x(0,1,1)_12')
```

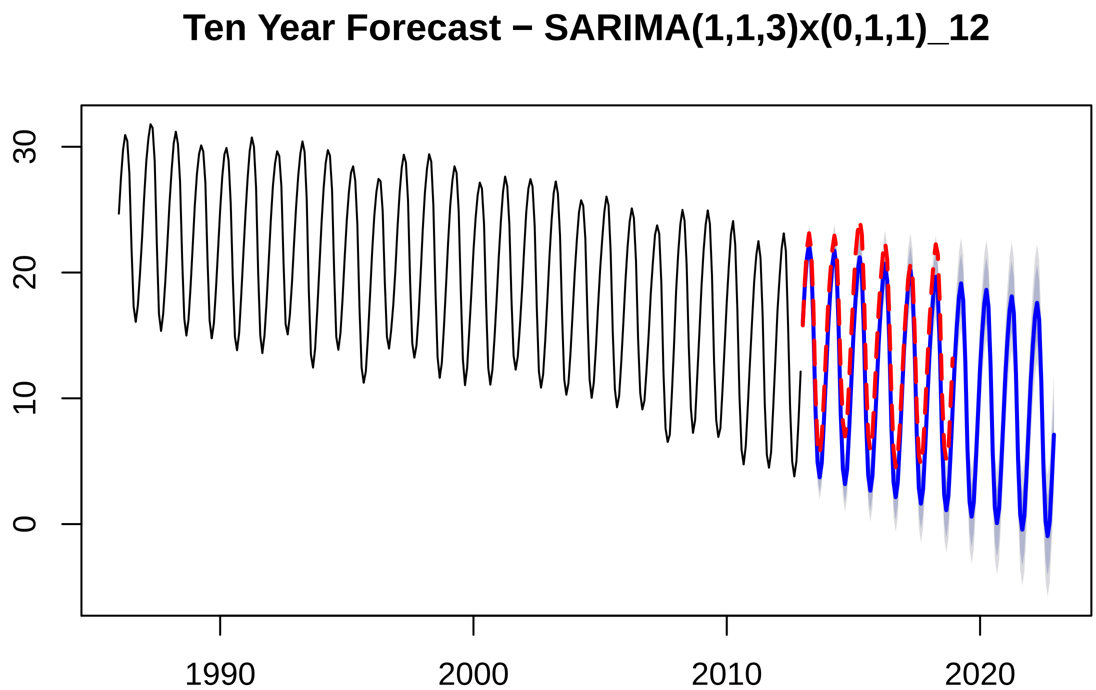


* The 5-year forecast is shown as a blue line and the forecast limits are grey.

3.6.2 Long-term (10 year) forecast for SARIMA(1,1,3)x$(0,1,1)_{12}$.

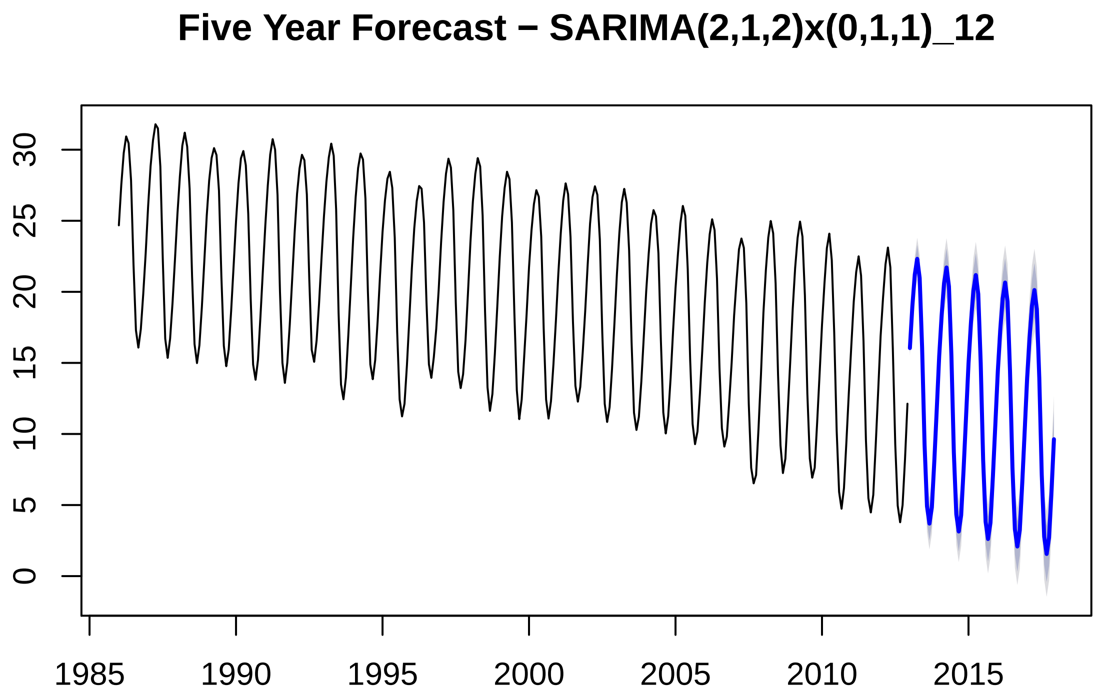
```
sea_ice_for_long_113 = Arima(sea_ice_ts,order=c(1,1,3),seasonal=list(order=c(0,1,1), period=12))

future_long_113 = forecast(sea_ice_for_long_113, h = 120)
```

```
par(mfrow=c(1,1))
plot(future_long_113, main = 'Ten Year Forecast - SARIMA(1,1,3)x(0,1,1)_12')
lines(window(sea_ice_long_ts,start=c(2013, 1)), col = 2, lty = 5, lwd = 2, type = 'l')
```



3.6.3 Five year forecast for SARIMA(2,1,2)x$(0,1,1)_{12}$.

```
sea_ice_for_212 = Arima(sea_ice_ts,order=c(2,1,2),seasonal=list(order=c(0,1,1), period=12))

future_212 = forecast(sea_ice_for_212, h = 60)
plot(future_212, main = 'Five Year Forecast - SARIMA(2,1,2)x(0,1,1)_12')
```

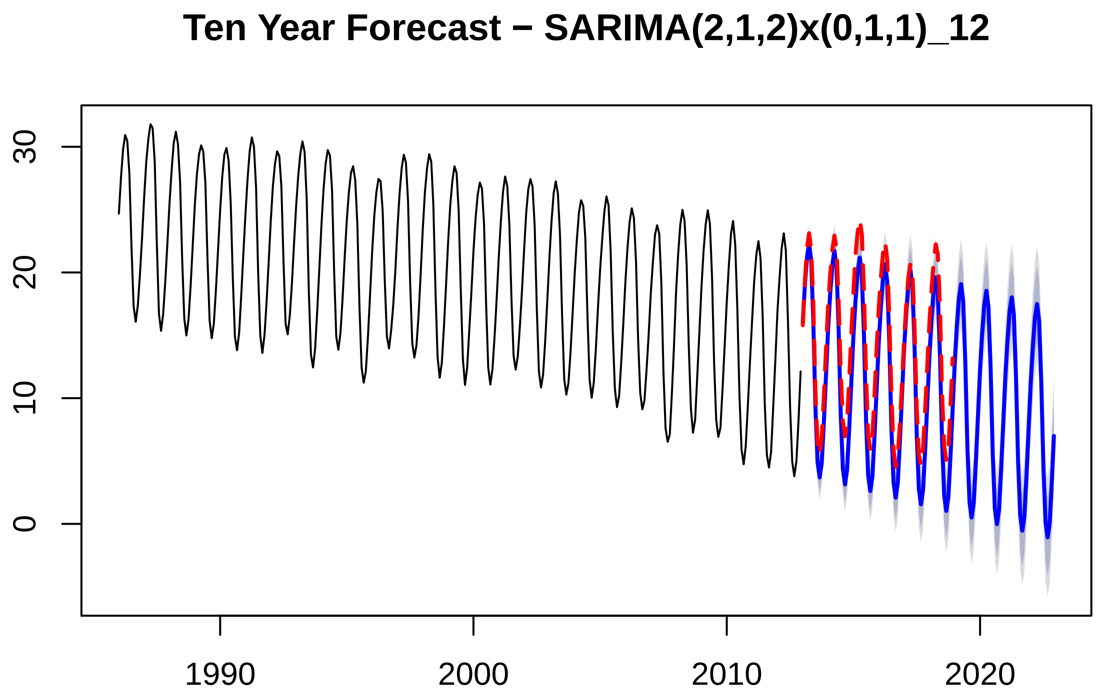


* The 5-year forecast is shown as a blue line and the forecast limits are grey.

3.6.4 Long-term (10 year) forecast for SARIMA(2,1,2)x(0,1,1)$_{12}$.

```
sea_ice_for_long_212 = Arima(sea_ice_ts,order=c(2,1,2),seasonal=list(order=c(0,1,1), period=12))

future_long_212 = forecast(sea_ice_for_long_212, h = 120)
```

```
par(mfrow=c(1,1))
plot(future_long_212, main = 'Ten Year Forecast - SARIMA(2,1,2)x(0,1,1)_12')
lines(window(sea_ice_long_ts,start=c(2013, 1)), col = 2, lty = 5, lwd = 2, type = 'l')
```



- The truncated data (2013-2018) has been overlayed (dashed red line) onto the 10-year forecasts for Arctic sea-ice volumes. The data fit well within the forecast limits for both models and highlights the potential for forecasts to gain a idea of future Arctic sea-ice volumes.