report 9

Khaled Hasan

2024-05-20

Data

```
GLOBALTEMPERATURE = read.csv(file = "C:\\Users/ss/Desktop/Time_series_Analysis/MyGlobalTemperetures.csv
global_temp = ts(GLOBALTEMPERATURE[,1], start = c(1850, 1), frequency = 12)
northernhemisphere_temp = ts(GLOBALTEMPERATURE[,2], start = c(1850, 1), frequency = 12)
southernhemisphere_temp = ts(GLOBALTEMPERATURE[,3], start = c(1850, 1), frequency = 12)
LattitudinalTemps = read.csv(file = "C:\\Users/ss/Desktop/Time_series_Analysis/LatittudCuttedTempereture
north = ts(LattitudinalTemps[, 2], start = c(1850, 1), frequency = 12)
tropical = ts(LattitudinalTemps[, 3], start = c(1850, 1), frequency = 12)
south = ts(LattitudinalTemps[, 4], , start = c(1850, 1), frequency = 12)
south_noPole = ts(LattitudinalTemps[, 5], start = c(1850, 1), frequency = 12)
```

Forecasting from 1960

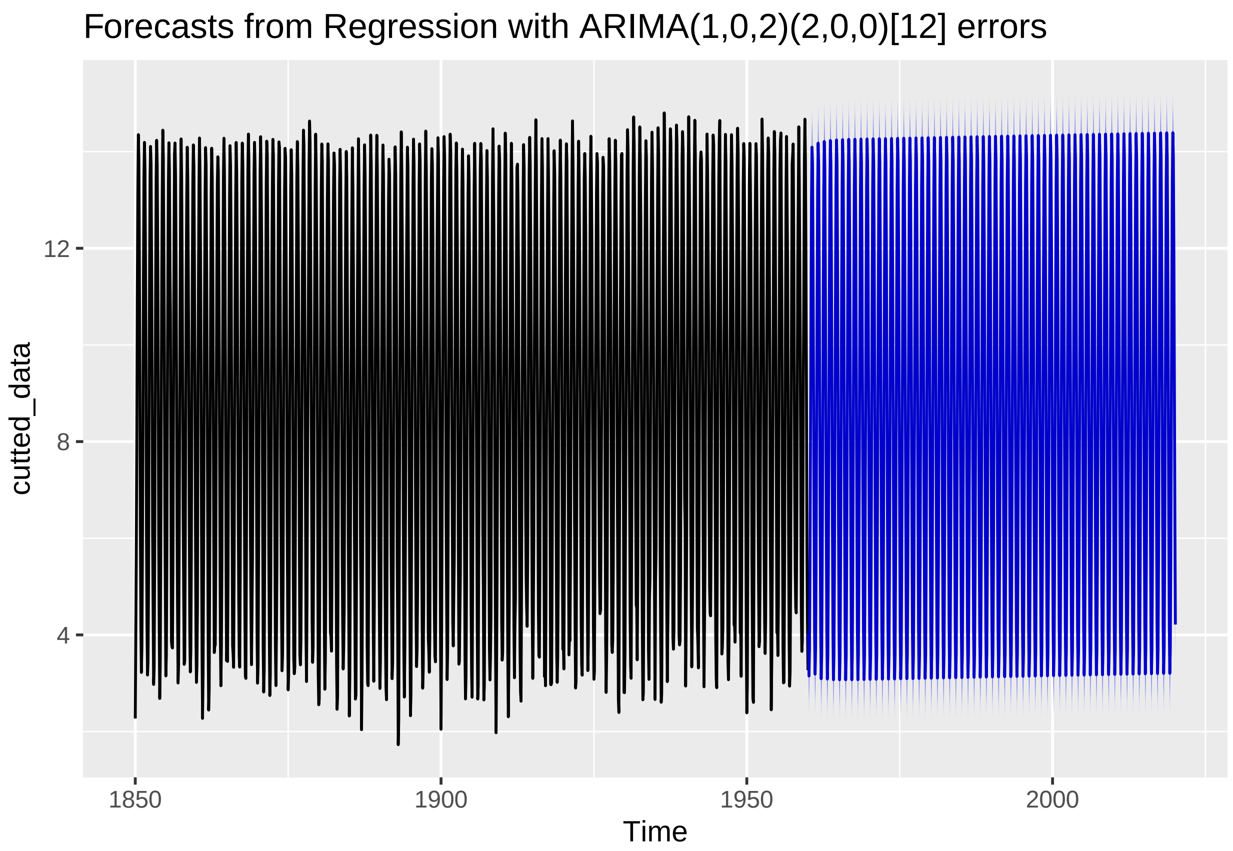
```
Arima_fittng <- function(timeseries, startingPoint = start(timeseries), endingPoint = end(timeseries)){
  cutted_data = window(timeseries, start = startingPoint, end = endingPoint, freq = 12)
  t = seq_along(cutted_data)
  regressors = cbind(sin(pi/6*t), cos(pi/6*t), t)
  arima_fit = auto.arima(cutted_data, xreg = regressors, approximation = FALSE, seasonal = TRUE)
  return(arima_fit)
}

global_fitting_Arimareg = Arima_fittng(global_temp, startingPoint = c(1987, 1))
global_fitting_old = Arima_fittng(global_temp, startingPoint = c(1850, 1), endingPoint = c(1960, 1))

summary(global_fitting_old)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,2)(2,0,0)[12] errors
##
## Coefficients:
##          ar1      ma1      ma2    sar1    sar2  intercept
##       0.8613  -0.5402  -0.1708  0.1636  0.1555     8.6451  -3.2450  -4.5850
## s.e.  0.0447   0.0536   0.0344  0.0278  0.0274     0.0622   0.0262   0.0262
##            t
##       2e-04
## s.e.  1e-04
##
```

```
## sigma^2 = 0.1433:  log likelihood = -587.51
## AIC=1195.02   AICc=1195.18   BIC=1246.88
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.0003400988 0.3773153 0.2798349 -0.8549017 5.088169 0.7281159
##                      ACF1
## Training set -0.003337366
```

```
t_forecast = (0):(720)
forecasting_regressors = cbind(sin(t_forecast*pi/6), cos(t_forecast*pi/6), t_forecast)
global_fitting_old %>% forecast(xreg = forecasting_regressors) %>% autoplot()
```

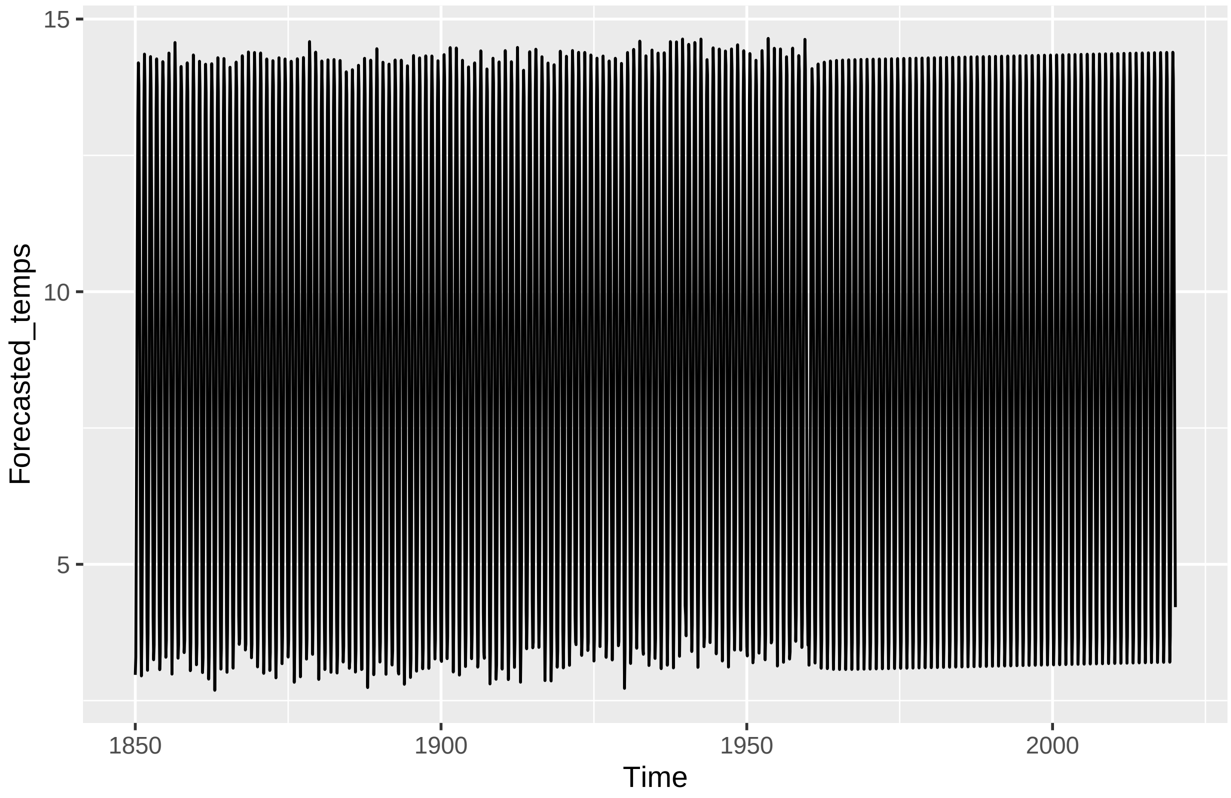
```
## Warning in forecast.forecast_ARIMA(., xreg = forecasting_regressors): xreg
## contains different column names from the xreg used in training. Please check
## that the regressors are in the same order.
```



```
data_up_to_1960 = window(global_temp, start = c(1850, 1), end = c(1960, 1), freq = 12)

Forecasted = forecast(global_fitting_old, xreg = forecasting_regressors)
```

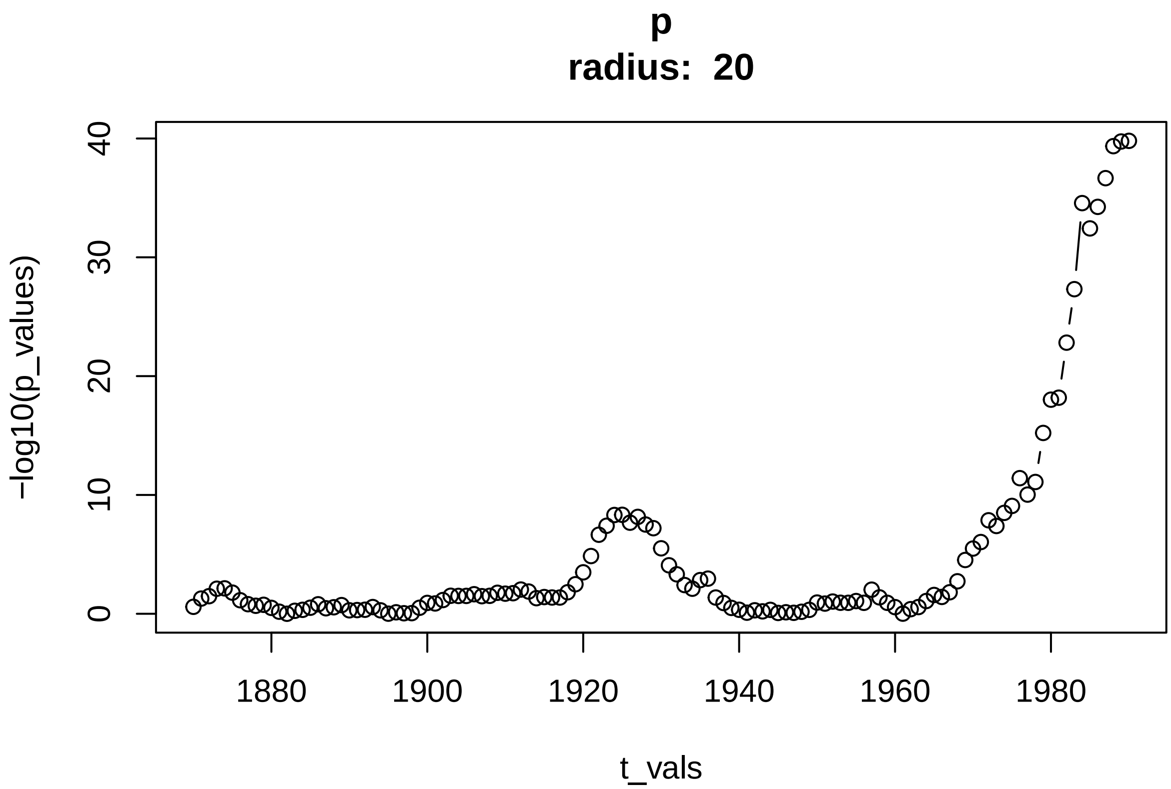
```
## Warning in forecast.forecast_ARIMA(global_fitting_old, xreg =
## forecasting_regressors): xreg contains different column names from the xreg
## used in training. Please check that the regressors are in the same order.
```

```r
Forecasted_temps= ts(c(Forecasted$fitted, Forecasted$mean), start = c(1850, 1), frequency=12)
autoplot(Forecasted_temps)
```



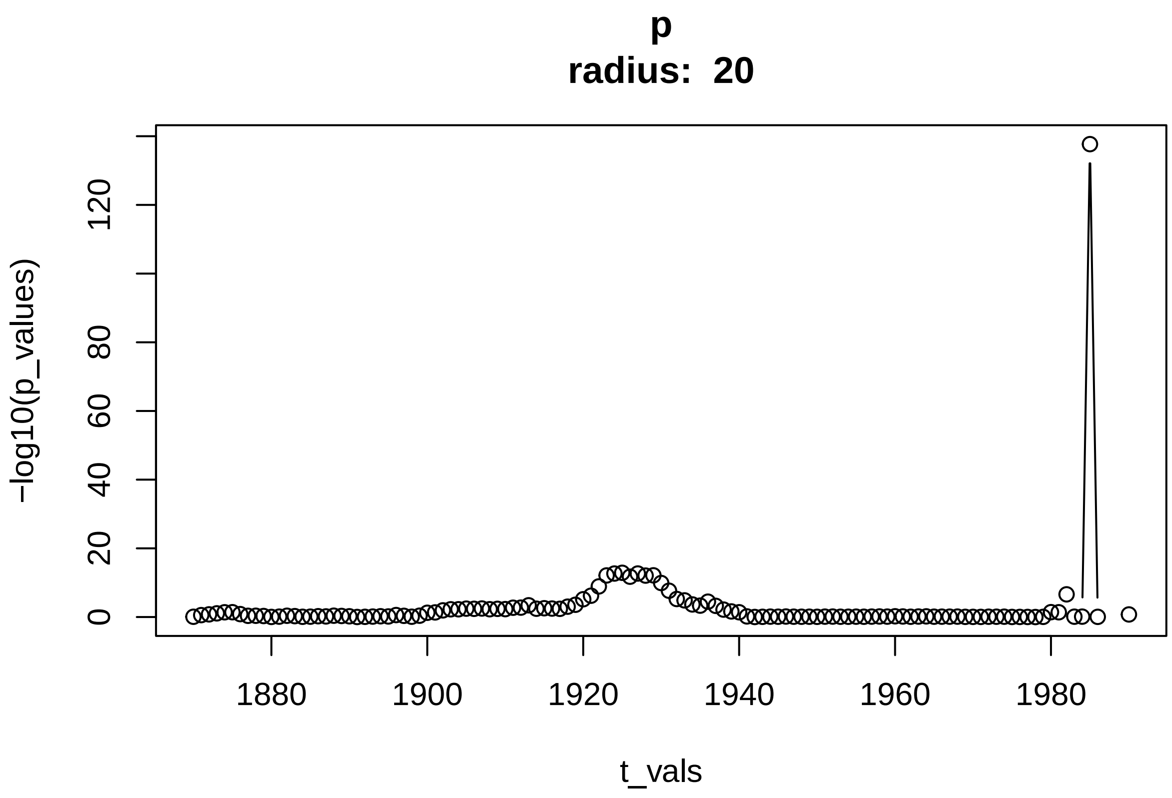
```r
linear_coef <- function(DATA, x, Ord, sOrd, radius = 2){

  temporary_data = window(DATA, start = c(x-radius, 1), end = c(x+radius, 1))

  new_t <- seq_along(temporary_data)

  temporary_xreg = cbind(sin(new_t*pi/6), cos(new_t*pi/6), new_t)

  temporary_model = arima(temporary_data, order = Ord, seasonal = sOrd, xreg = temporary_xreg)

  std_error <- sqrt(diag(vcov(temporary_model)))

  return(c(as.numeric(temporary_model$coef["new_t"]), as.numeric(sqrt(diag(vcov(temporary_model)))["new_
}


plot_p_vals <- function(timeseries, arima_fit, r){


  ord = arima_fit$arma

  p = ord[1]; q = ord[2]; P = ord[3]; Q = ord[4]; period = ord[5]; d = ord[6]; D = ord[7];
  parameters =c()

  errors = c()

  p_values = c()

  rad = r

  sp = 1850

  fp = 2010

  for (i in (sp + rad):(fp - rad)){
      u = linear_coef(timeseries,  i, c(p, d, q), c(P, D, Q) , rad)
      parameters <- cbind(parameters, u[1])
      errors <- cbind(errors, u[2])
      p = 2*pnorm(min(0, 2*u[1]), mean = u[1], sd = u[2], lower.tail = TRUE)
      p_values = cbind(p_values, p)
  }

  t_vals = c((sp + rad):(fp - rad))

  plot(t_vals, -log10(p_values), type='b', main=paste("p\nradius: ", toString(rad)))

}

plot_p_vals(global_temp, arima_fit = global_fitting_Arimareg, 20)
```

```
## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced
```



```r
plot_p_vals(Forecasted_temps, arima_fit = global_fitting_old, 20)
```

```
## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced
## Warning in log(s2): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced

## Warning in sqrt(diag(vcov(temporary_model))): NaNs produced
```

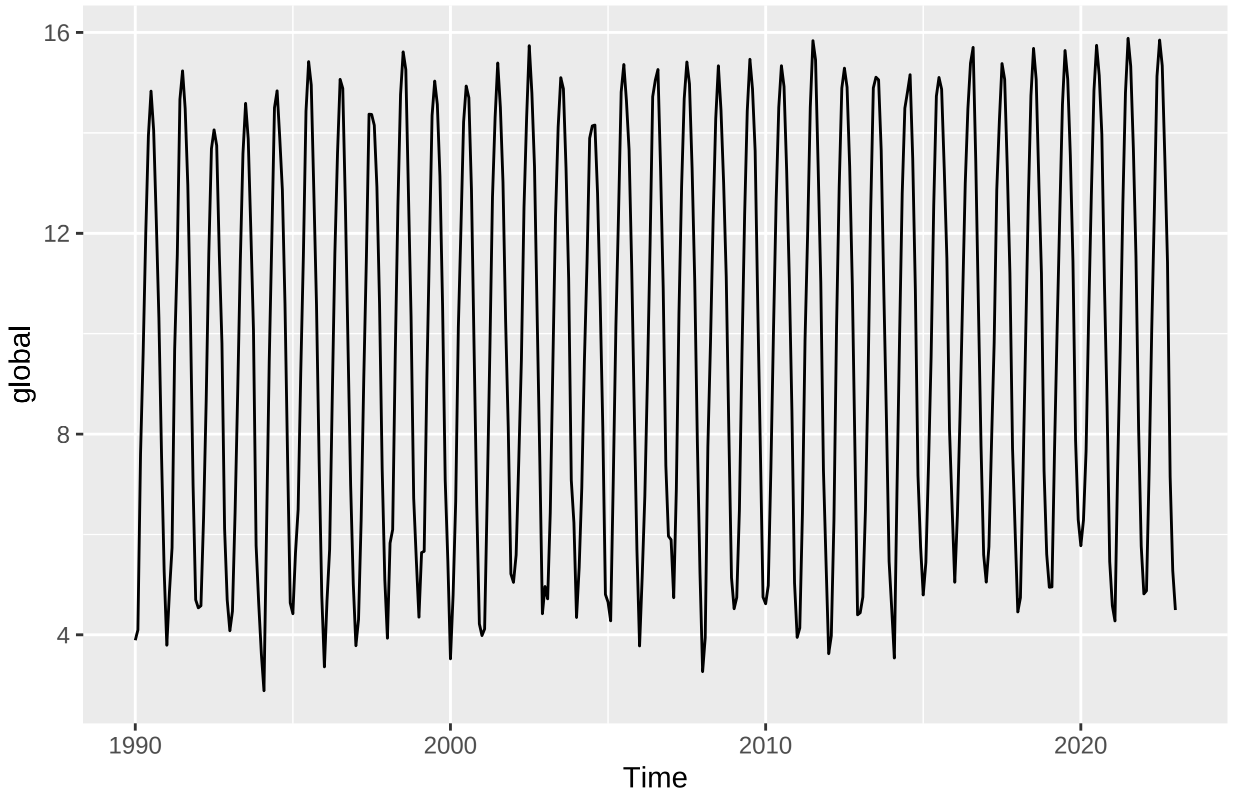


plots

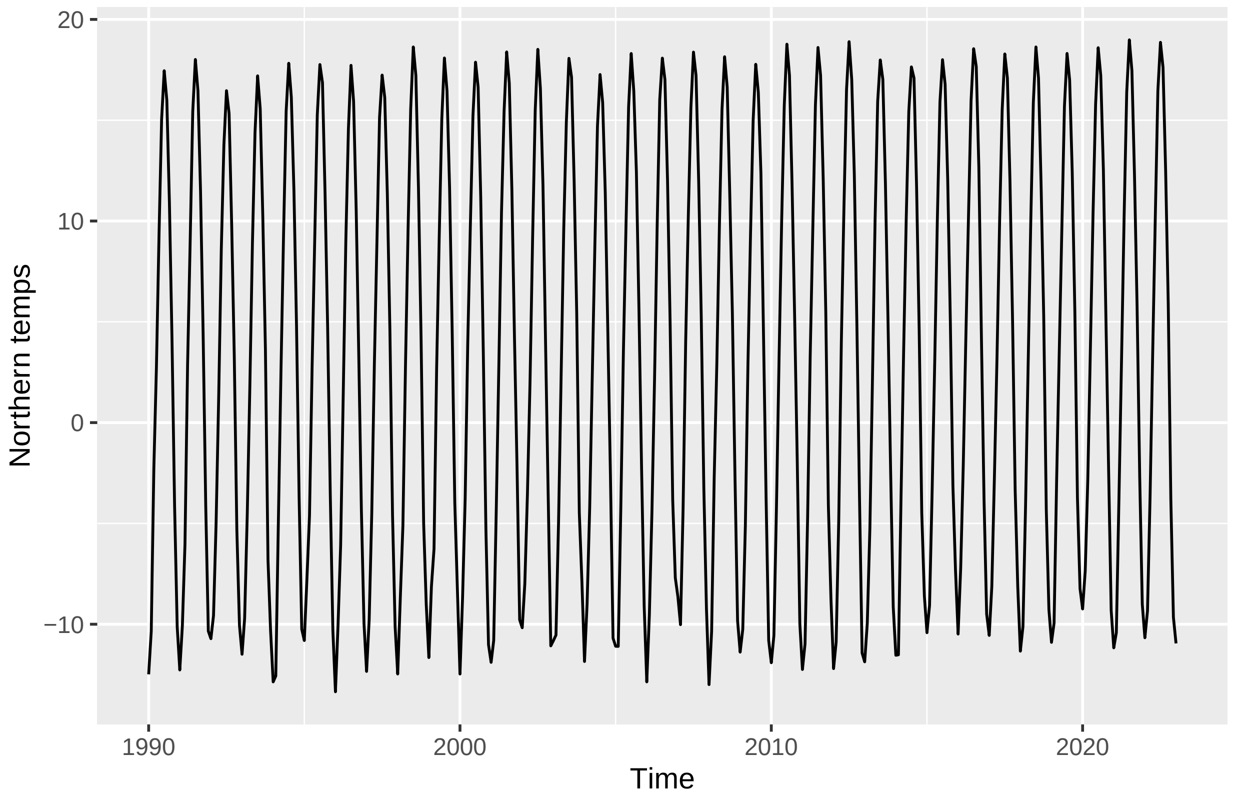
```
library(ggplot2)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.3.3
```

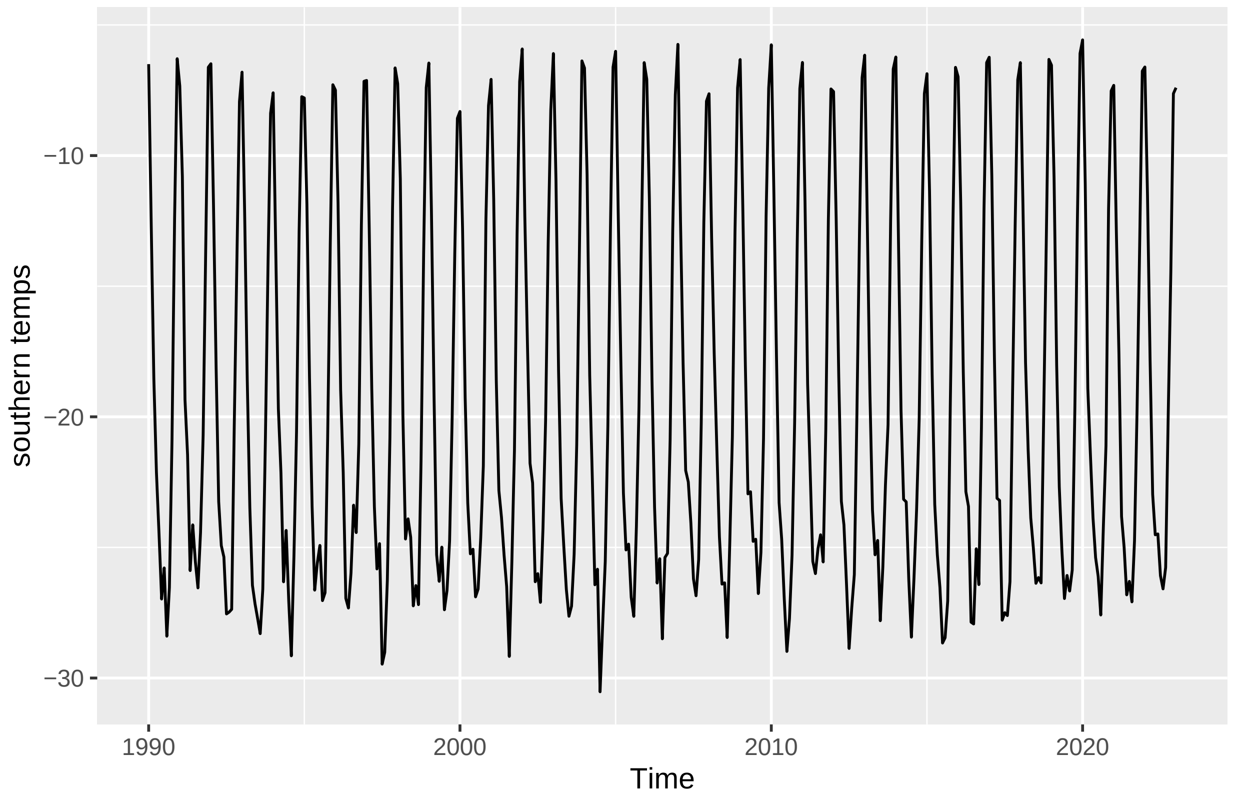
```
autoplot1 <- autoplot(window(global_temp, start = c(1990, 1), freq = 12), ylab = "global")

autoplot2 <- autoplot(window(north, start = c(1990, 1), freq = 12), ylab = "Northern temps")

autoplot3 <- autoplot(window(south, start = c(1990, 1), freq = 12), ylab = "southern temps")

autoplot4 <- autoplot(window(tropical, start = c(1990, 1), freq = 12), ylab = "tropical temps")

autoplot5 <- autoplot(window(south_noPole, start = c(1990, 1), freq = 12), ylab = "southern ex. Pole te
par(mfrow = c(1, 5))
plot(autoplot1)
```



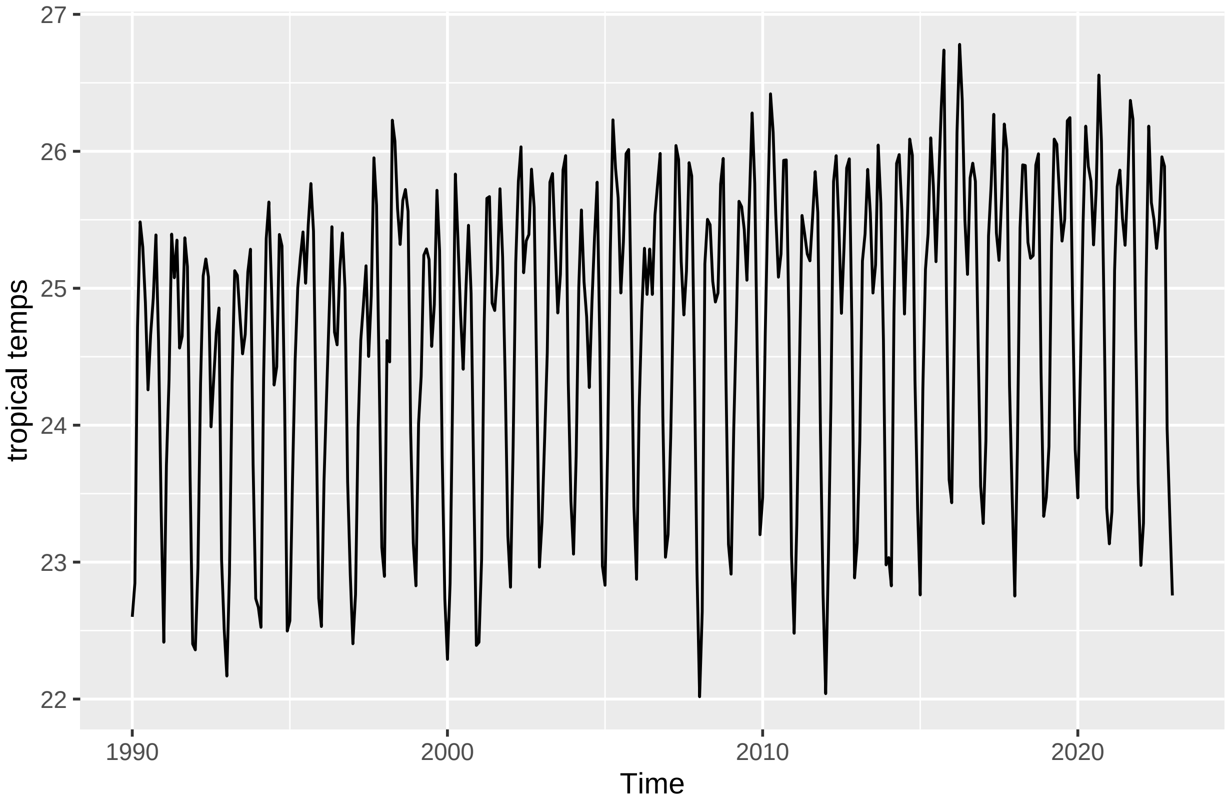
```
plot(autoplot2)
```



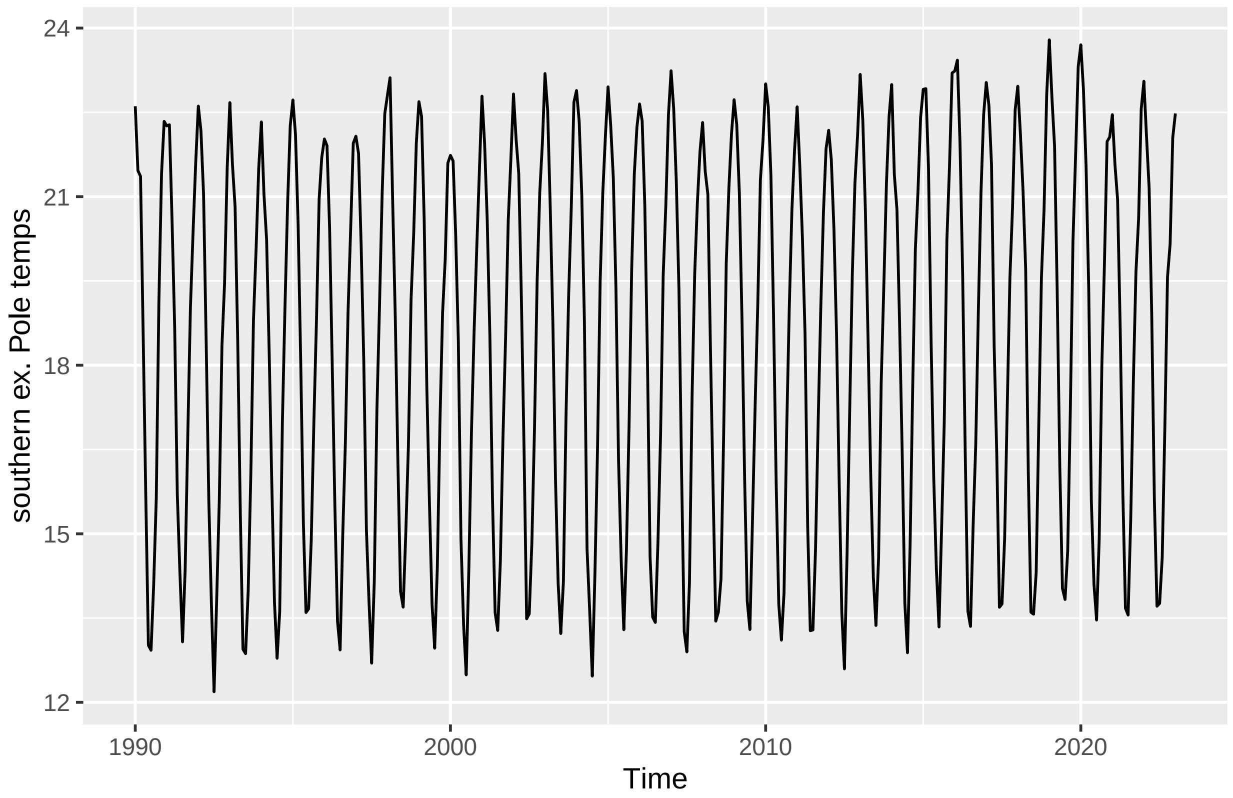
```r
plot(autoplot3)
```



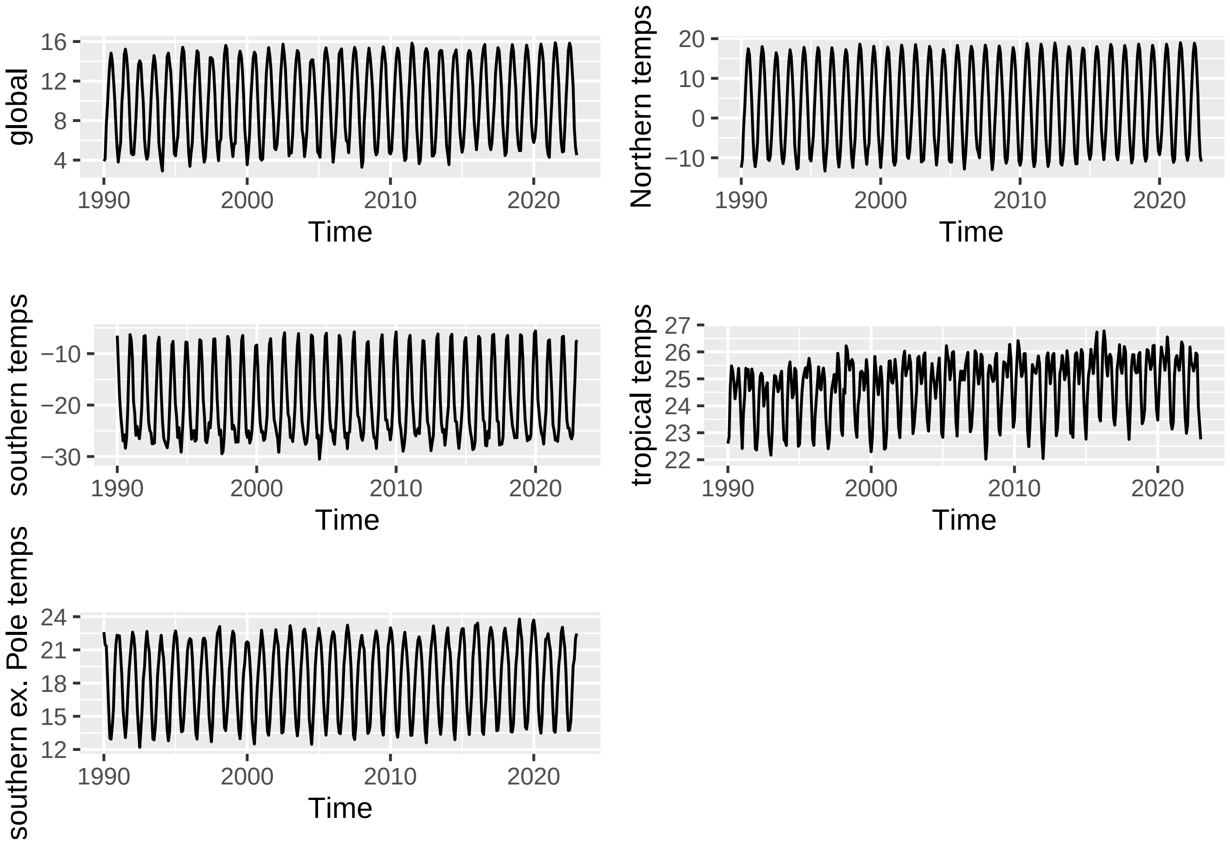
```r
plot(autoplot4)
```



```r
plot(autoplot5)
```



```
layout(matrix(c(1, 2, 3, 4, 5), nrow = 1))
grid.arrange(autoplot1, autoplot2, autoplot3, autoplot4, autoplot5)
```



There are few interesting points about the above data. First, we notice that excluding north-pole did not seem to shift the time phase, but it significantly shifted the data upwards, we also notice that the tropical data seems to agree with the northern phase, suggesting that the region between latitudes -30^o and 30^o has more mass in the northern hemisphere, but what is rather more interesting, is the seasonal patters that defies the usual sinusoidal pattern, this behavior is to be studied.

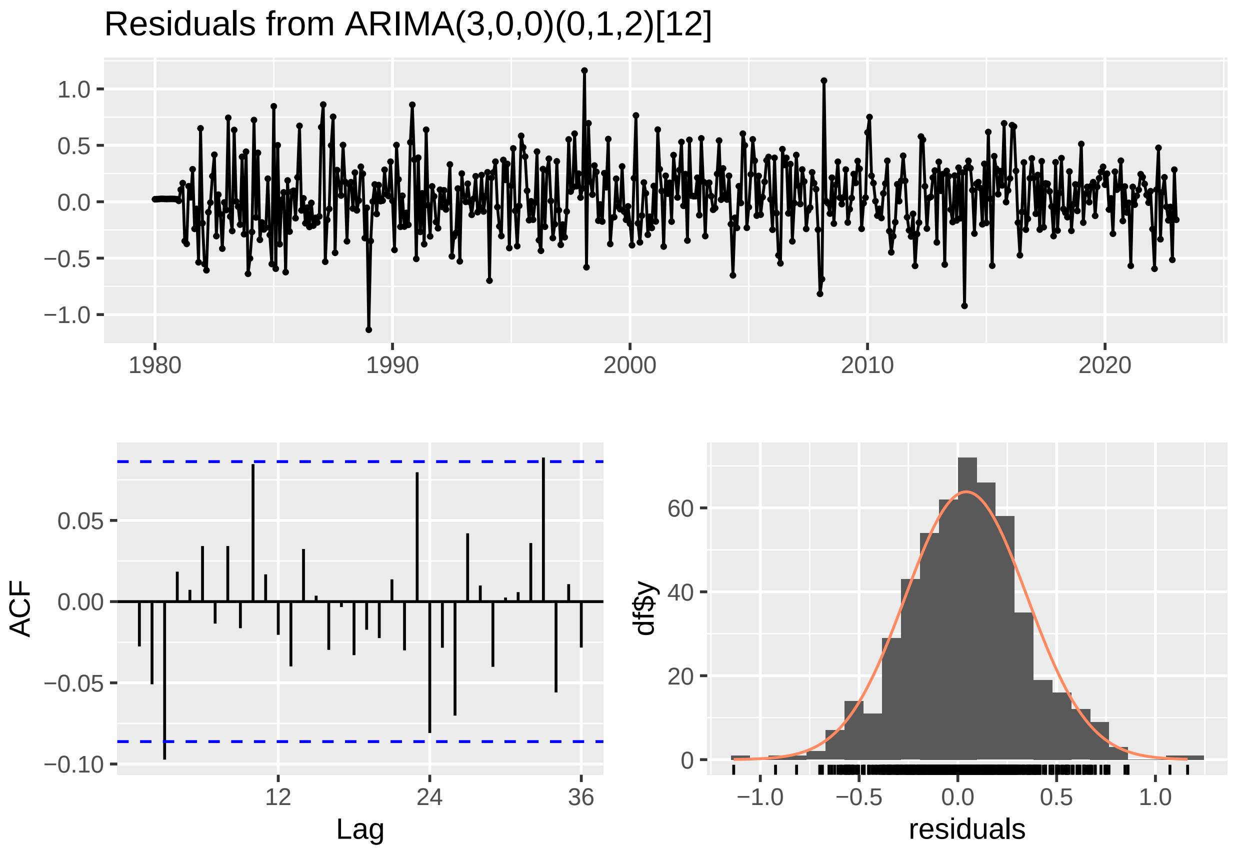
Arima Fitting

```
tropical_autoArima_fit = auto.arima(window(tropical, start=c(1980, 1), freq=12), approximation = FALSE,

summary(tropical_autoArima_fit)
```

```
## Series: window(tropical, start = c(1980, 1), freq = 12)
## ARIMA(3,0,0)(0,1,2)[12]
##
## Coefficients:
##          ar1     ar2     ar3     sma1    sma2
##       0.4125  0.2153  0.0855  -0.9328  0.0507
## s.e.  0.0456  0.0478  0.0462   0.0498  0.0475
##
## sigma^2 = 0.1006:  log likelihood = -143.67
## AIC=299.34   AICc=299.51   BIC=324.69
##
## Training set error measures:
```

```
##                        ME      RMSE       MAE       MPE      MAPE      MASE
## Training set 0.04242536 0.3119142 0.2412032 0.1557792 0.9874523 0.6378395
##                     ACF1
## Training set -0.02759831
```

```
checkresiduals(tropical_autoArima_fit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0)(0,1,2)[12]
## Q* = 23.042, df = 19, p-value = 0.2355
##
## Model df: 5.   Total lags used: 24
```

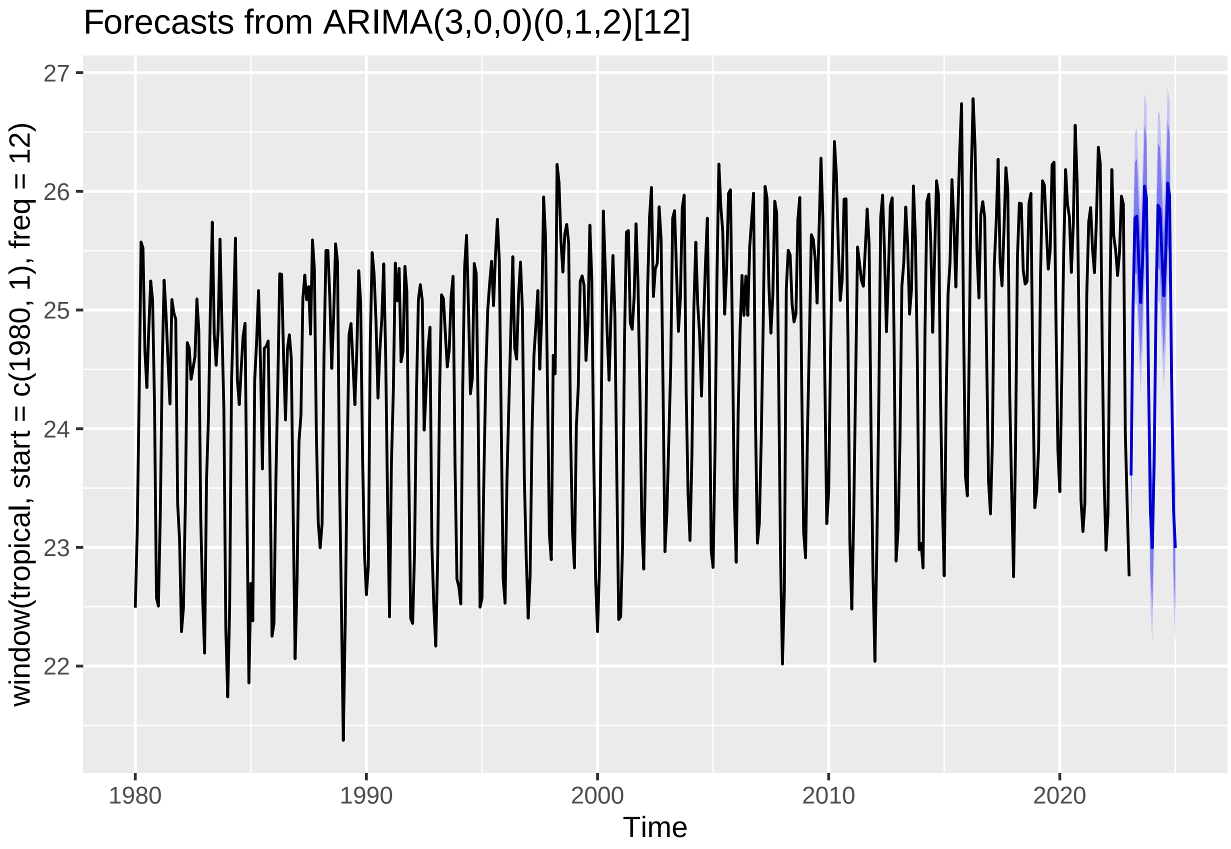
doing the same with different latitudinal cuts:

```
global_autoArima_fit = auto.arima(window(global_temp, start=c(1980, 1), freq=12), approximation = FALSE

north_autoArima_fit = auto.arima(window(north, start=c(1980, 1), freq=12), approximation = FALSE, seaso


south_autoArima_fit = auto.arima(window(south, start=c(1980, 1), freq=12), approximation = FALSE, seaso
south_np_autoArima_fit = auto.arima(window(south_noPole, start=c(1980, 1), freq=12), approximation = FA
northernhemisphereArimaFit = auto.arima(window(northernhemisphere_temp, start=c(1980, 1), freq=12), app
southernhemisphereArimaFit = auto.arima(window(southernhemisphere_temp, start=c(1980, 1), freq=12), app
```

```
{
  print(global_autoArima_fit$arma)
  print(north_autoArima_fit$arma)
  print(south_autoArima_fit$arma)
  print(south_np_autoArima_fit$arma)
  print(northernhemisphereArimaFit$arma)
  print(southernhemisphereArimaFit$arma)
}
```

```
## [1]  2  2  1  1 12  0  1
## [1]  2  0  1  1 12  0  1
## [1]  1  0  2  0 12  0  1
## [1]  1  1  2  0 12  0  1
## [1]  1  1  0  2 12  0  1
## [1]  2  0  0  1 12  0  1
```

```
tropical_autoArima_fit %>% forecast(h = 24) %>% autoplot()
```



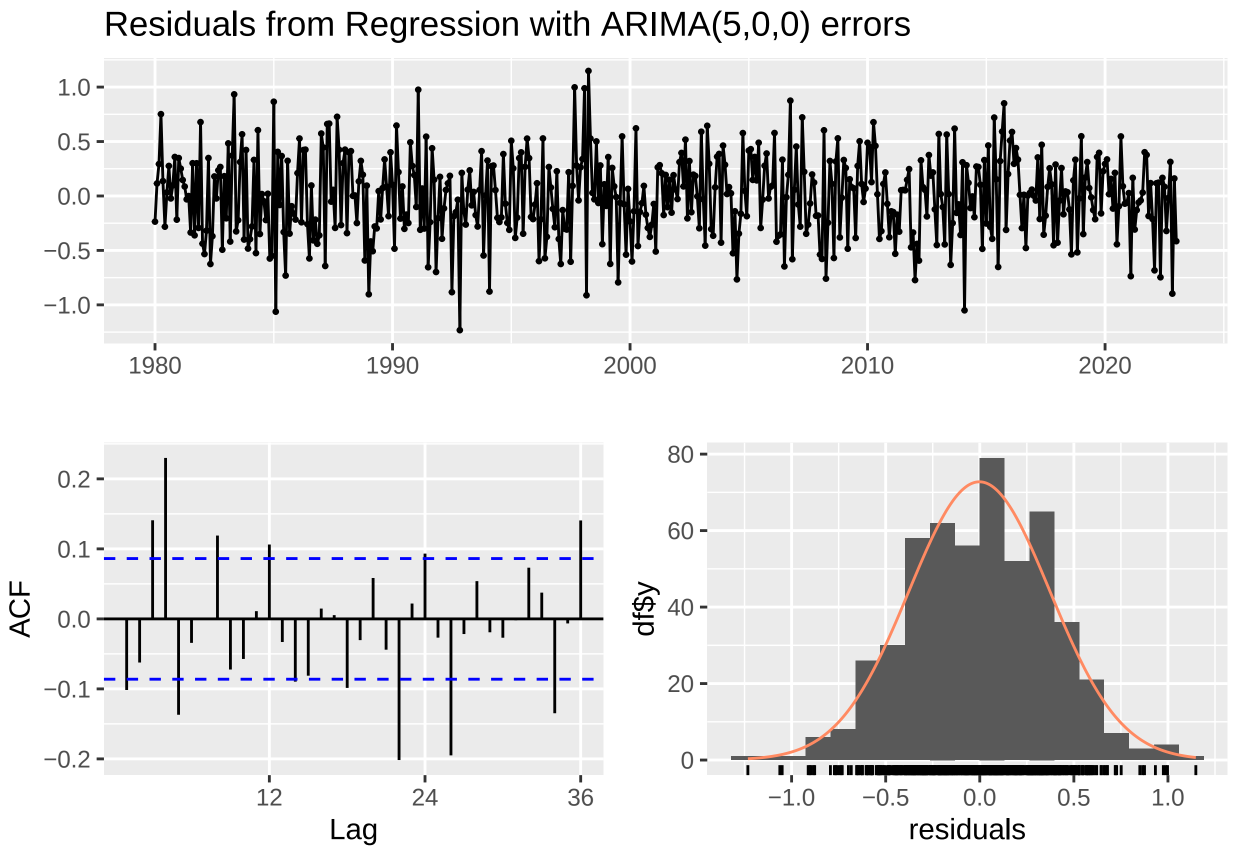
```
{
  print(global_autoArima_fit$aicc)
  print(north_autoArima_fit$aicc)
  print(tropical_autoArima_fit$aicc)
  print(south_autoArima_fit$aicc)
  print(south_np_autoArima_fit$aicc)
  print(northernhemisphereArimaFit$aicc)
  print(southernhemisphereArimaFit$aicc)
}
```

```
## [1] 698.0507
## [1] 1158.212
## [1] 299.5073
## [1] 1560.526
## [1] 604.3467
## [1] 835.0459
## [1] 830.8596
```

```
Arima_fittng <- function(timeseries, startingPoint = start(timeseries), endingPoint = end(timeseries)){
  cutted_data = window(timeseries, start = startingPoint, end = endingPoint, freq = 12)
  t = seq_along(cutted_data)
  regressors = cbind(sin(pi/6*t), cos(pi/6*t), t)
  arima_fit = auto.arima(cutted_data, xreg = regressors, approximation = FALSE, stepwise = FALSE, seasor
  return(arima_fit)
}

global_fitting_Arimareg = Arima_fittng(global_temp, startingPoint = c(1980, 1))

north_reg = Arima_fittng(north, startingPoint = c(1980, 1))

tropical_reg = Arima_fittng(tropical, startingPoint = c(1980, 1))

south_reg = Arima_fittng(south, startingPoint = c(1980, 1))

south_np_reg = Arima_fittng(south_noPole, startingPoint = c(1980, 1))


summary(north_reg)
```

```
## Series: cutted_data
## Regression with ARIMA(2,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1     ar2    sar1    sar2  intercept                          t
##       0.2510  0.1117  0.0788  0.2104     2.3072  -8.0905  -12.4163  0.0034
## s.e.  0.0438  0.0451  0.0431  0.0447     0.1407   0.0855    0.0854  0.0005
##
## sigma^2 = 0.5779:  log likelihood = -588.48
## AIC=1194.95   AICc=1195.31   BIC=1233.19
##
## Training set error measures:
##                       ME      RMSE      MAE        MPE     MAPE      MASE
## Training set -0.0023746 0.7543063 0.571757 -0.4562573 9.325695 0.7164933
##                     ACF1
## Training set 0.003511724
```

```
checkresiduals(tropical_reg)
```

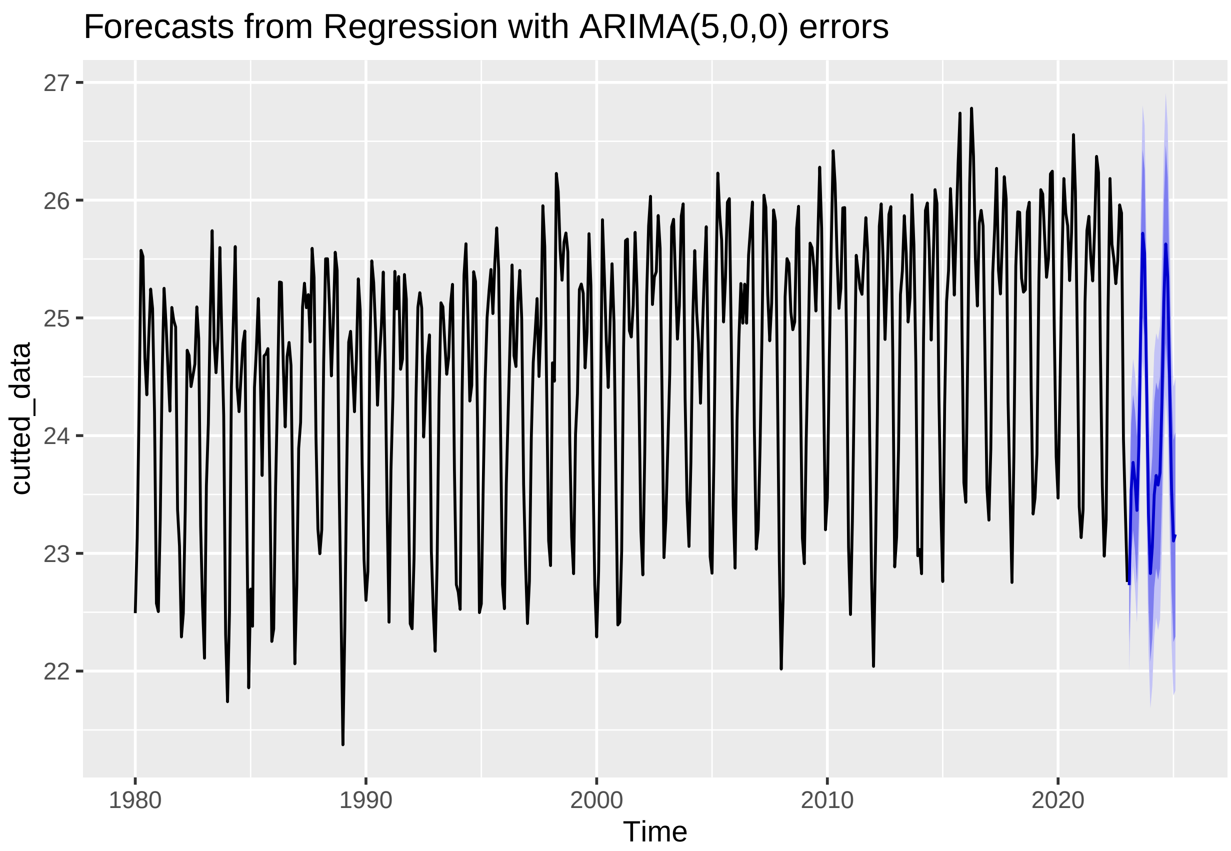


```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(5,0,0) errors
## Q* = 118.03, df = 19, p-value = 2.22e-16
##
## Model df: 5.   Total lags used: 24
```

Notice that the regression models here did not fit the data as well as the global case, despite using periodic fits to handle seasonality.

```
forecast_reg <- function(model, h){
  t_forecast = (0):(h)
  forecasting_regressors = cbind(sin(t_forecast*pi/6), cos(t_forecast*pi/6), t_forecast)
  model %>% forecast(xreg = forecasting_regressors) %>% autoplot() %>% return()
}

forecast_reg(tropical_reg, 24)
```

```
## Warning in forecast.forecast_ARIMA(., xreg = forecasting_regressors): xreg
## contains different column names from the xreg used in training. Please check
## that the regressors are in the same order.
```



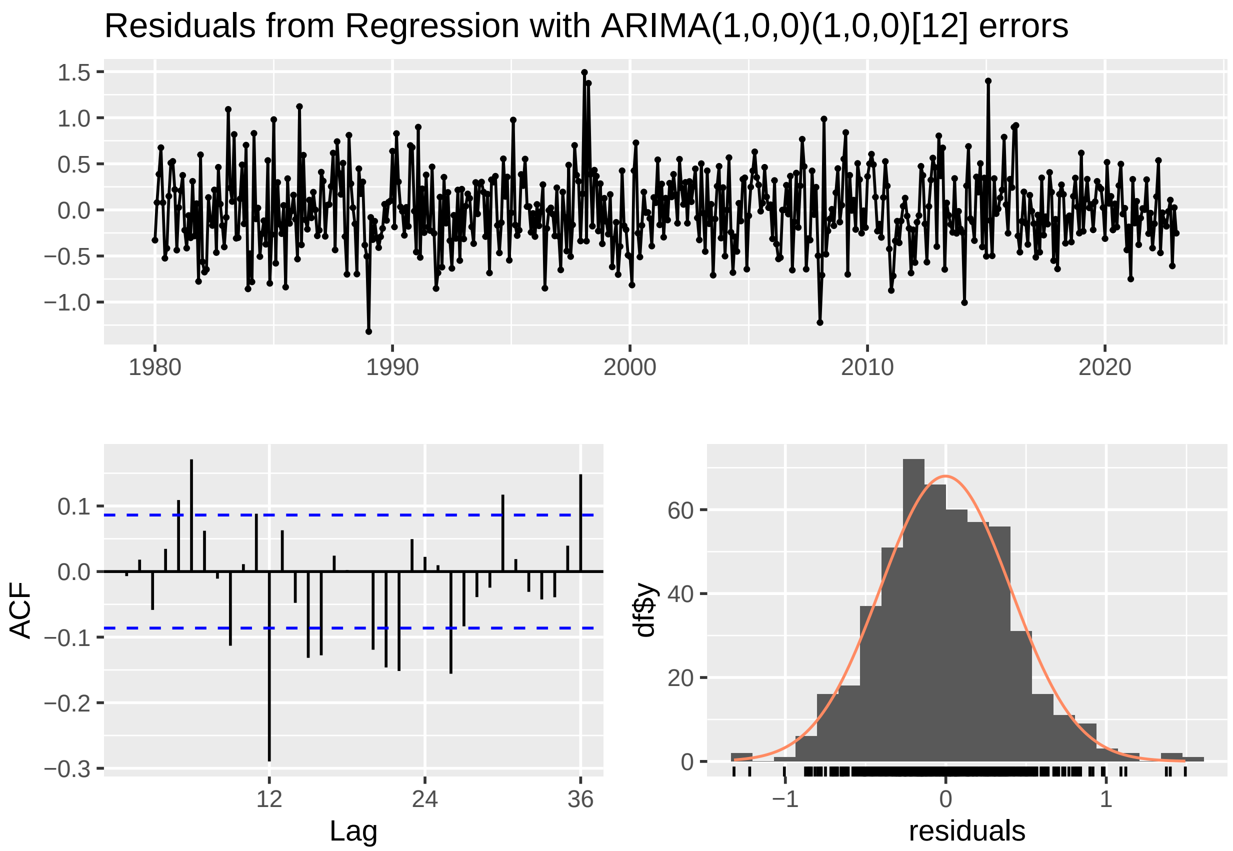
Triying finer frequencies:

```
Arima_fittng2 <- function(timeseries, startingPoint = start(timeseries), endingPoint = end(timeseries))
  cutted_data = window(timeseries, start = startingPoint, end = endingPoint, freq = 12)
  t = seq_along(cutted_data)
  regressors = cbind(sin(pi/6*t), cos(pi/6*t), sin(pi/12*t), cos(pi/12*t), t)
  arima_fit = auto.arima(cutted_data, xreg = regressors, approximation = FALSE, seasonal = TRUE)
  return(arima_fit)
}

tropical_reg2 = Arima_fittng2(tropical, startingPoint = c(1980, 1))

summary(tropical_reg2)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(1,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1  intercept                                          t
##       0.4626  0.7682    24.1284  -0.4559  -1.0216  -0.0133  0.0595  0.0018
## s.e.  0.0389  0.0277     0.2408   0.1579   0.1576   0.0255  0.0254  0.0008
##
## sigma^2 = 0.1674:  log likelihood = -272.93
## AIC=563.87   AICc=564.22   BIC=602.1
```

```
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.002282352 0.4059176 0.3210868 -0.0378739 1.316632 0.8490843
##                      ACF1
## Training set -0.006908302
```

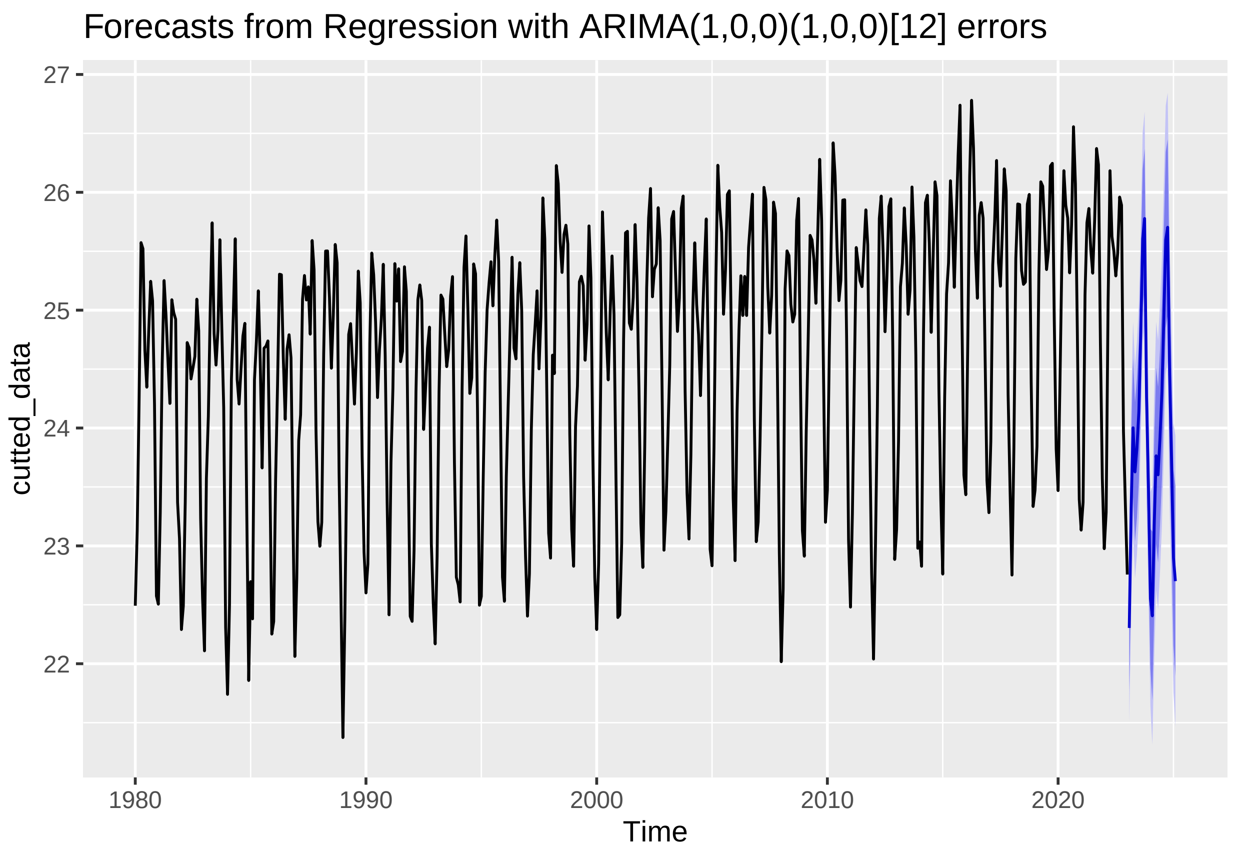
```
checkresiduals(tropical_reg2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0)(1,0,0)[12] errors
## Q* = 136.73, df = 22, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 24
```

```
t_forecast = (0):(24)
forecasting_regressors = cbind(sin(t_forecast*pi/6), cos(t_forecast*pi/6), sin(t_forecast*pi/12), cos(t
tropical_reg2 %>% forecast(xreg = forecasting_regressors) %>% autoplot() %>% return()
```

```
## Warning in forecast.forecast_ARIMA(., xreg = forecasting_regressors): xreg
## contains different column names from the xreg used in training. Please check
## that the regressors are in the same order.
```

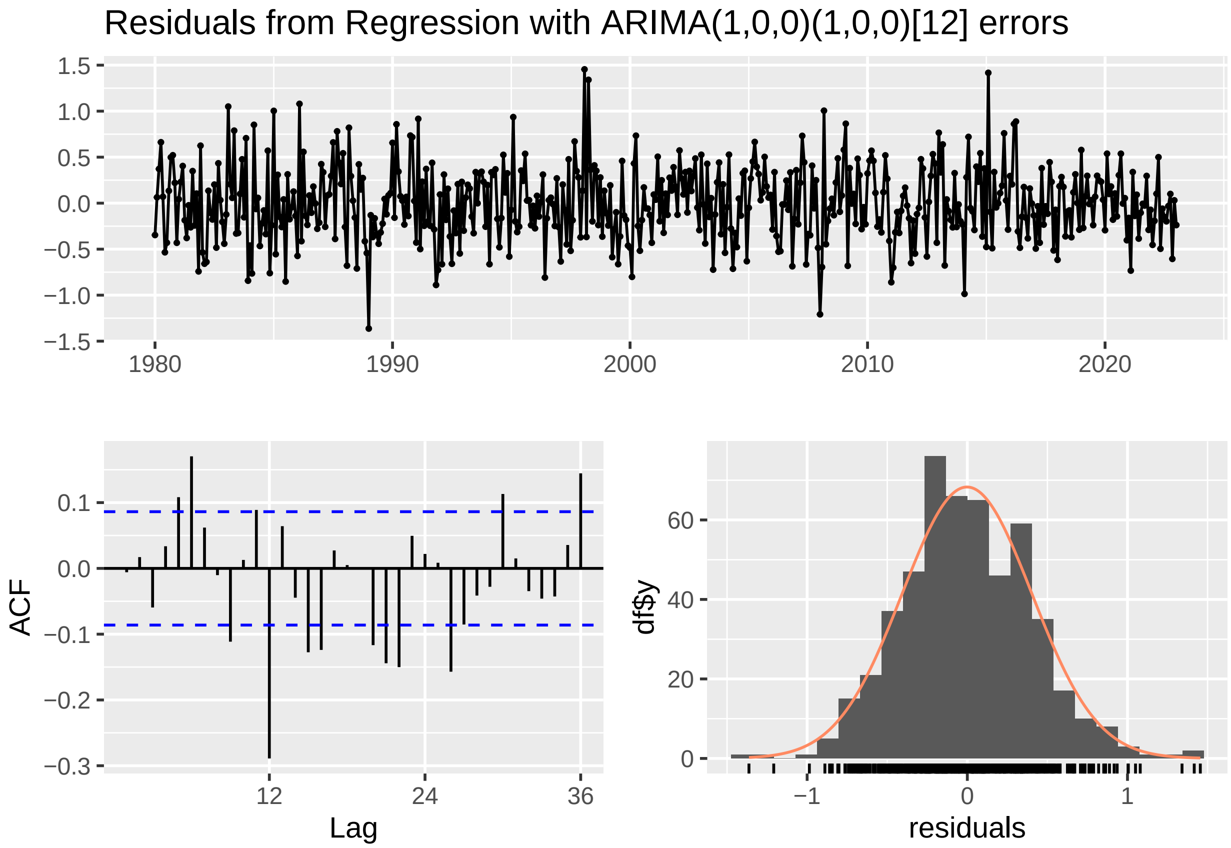


```
Arima_fittng3 <- function(timeseries, startingPoint = start(timeseries), endingPoint = end(timeseries))
  cutted_data = window(timeseries, start = startingPoint, end = endingPoint, freq = 12)
  t = seq_along(cutted_data)
  regressors = cbind(sin(pi/6*t), cos(pi/6*t), sin(pi/12*t), cos(pi/12*t), sin(pi/18*t), cos(pi/18*t), 
  arima_fit = auto.arima(cutted_data, xreg = regressors, approximation = FALSE, seasonal = TRUE)
  return(arima_fit)
}

tropical_reg3 = Arima_fittng3(tropical, startingPoint = c(1980, 1))

summary(tropical_reg3)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(1,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1  intercept
##       0.4562  0.7694    24.1222  -0.4561  -1.0199  -0.0134  0.0594  0.0372
## s.e.  0.0391  0.0276     0.2383   0.1572   0.1570   0.0252  0.0251  0.0298
##                    t
##       0.0293  0.0018
## s.e.  0.0297  0.0008
##
## sigma^2 = 0.1672:  log likelihood = -271.69
## AIC=565.37   AICc=565.9   BIC=612.1
```

```
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE    MAPE      MASE
## Training set -0.001834675 0.4049228 0.3211163 -0.03595401 1.31673 0.8491623
##                      ACF1
## Training set -0.005847378
```

```
checkresiduals(tropical_reg3)
```

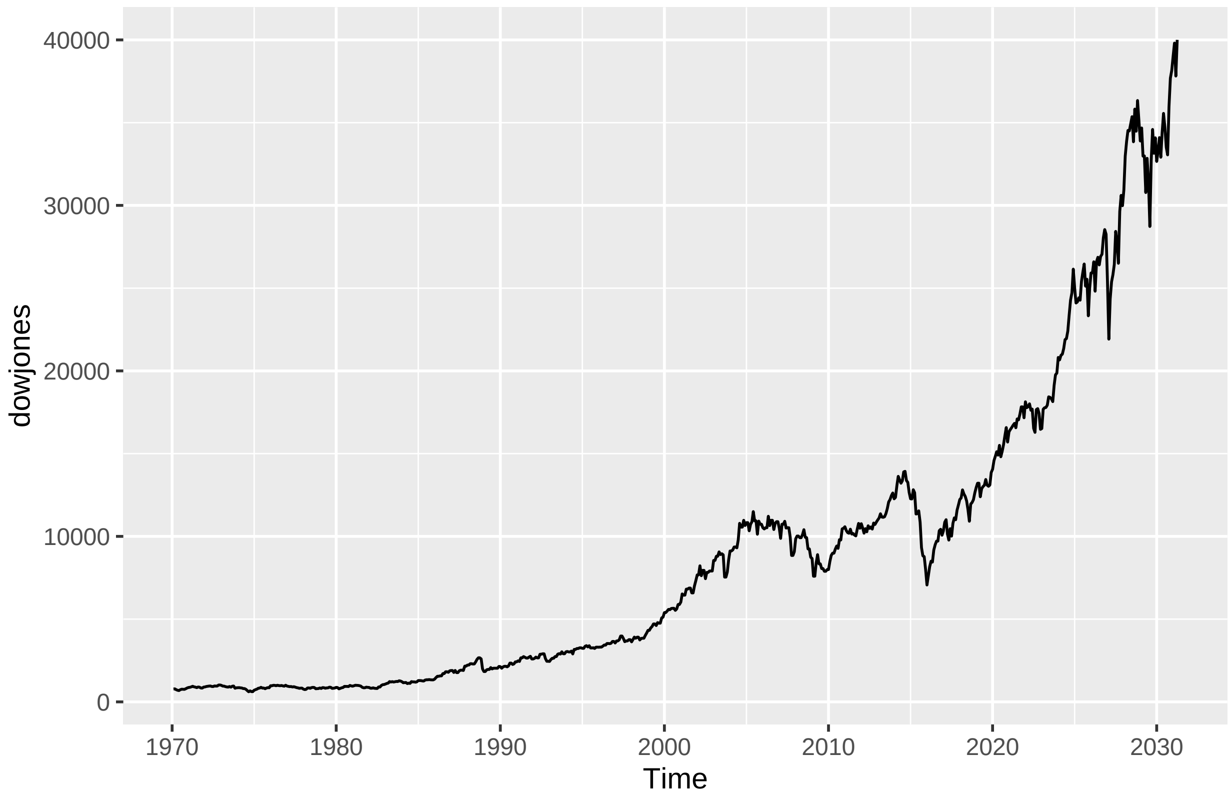


```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0)(1,0,0)[12] errors
## Q* = 134.22, df = 22, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 24
```

Dow Jones data:

src: https://www.investing.com/indices/us-30-historical-data

```
DOWJONES = read.csv(file= "C:\\Users/ss/Desktop/Time_series_Analysis/Dow Jones Industrial Average Histo
dowjones = ts(DOWJONES[, 2], start=c(1970, 2), frequency = 12)
autoplot(dowjones)
```



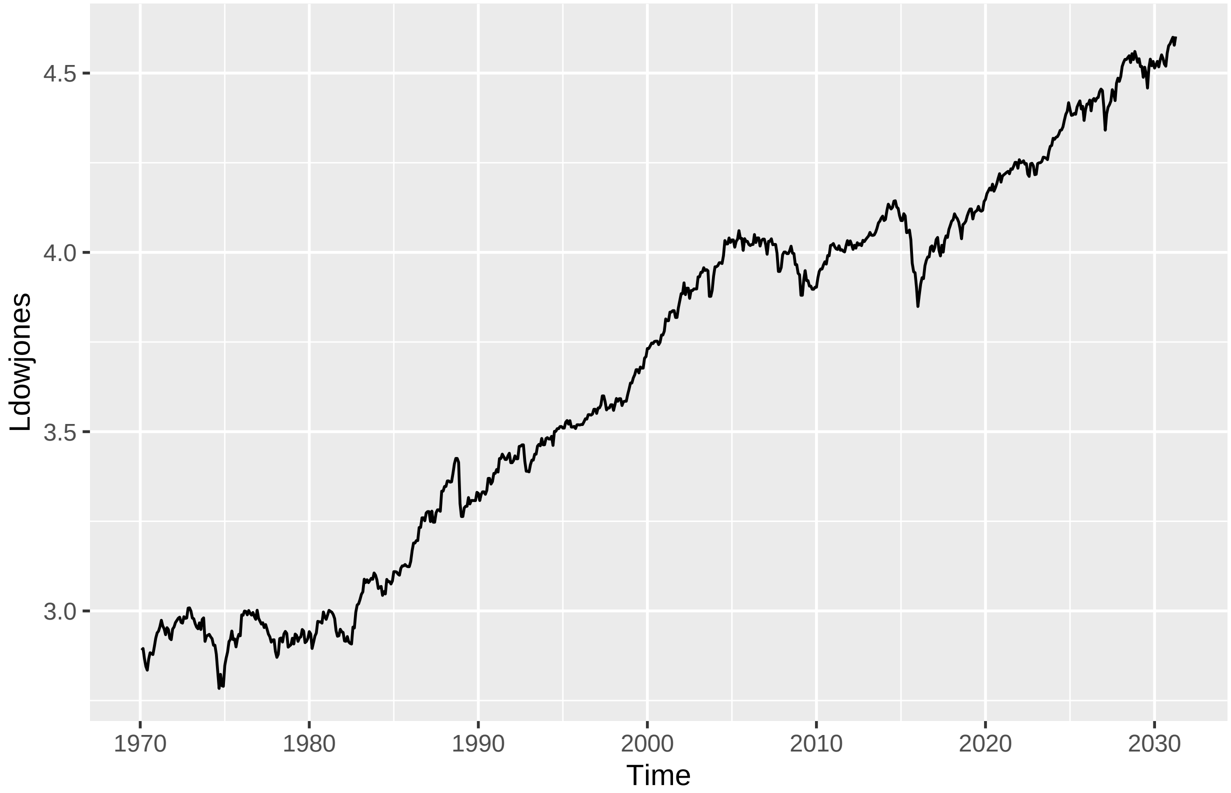
Now I will fit it to autoArima

```
dowjones_autoArima_fit = auto.arima(dowjones, approximation = FALSE, seasonal = TRUE, stepwise = FALSE,

summary(dowjones_autoArima_fit)
```

```
## Series: dowjones
## ARIMA(1,2,3)
##
## Coefficients:
##           ar1      ma1      ma2     ma3
##       -0.6725  -0.4261  -0.7751  0.2175
## s.e.   0.0988   0.0999   0.0878  0.0379
##
## sigma^2 = 293999:  log likelihood = -5654.95
## AIC=11319.89   AICc=11319.97   BIC=11342.88
##
## Training set error measures:
##                   ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 25.3025 539.9988 278.2725 0.1578803 3.068817 0.2765872
##                      ACF1
## Training set -0.002058391
```

try logarithmic transformation:

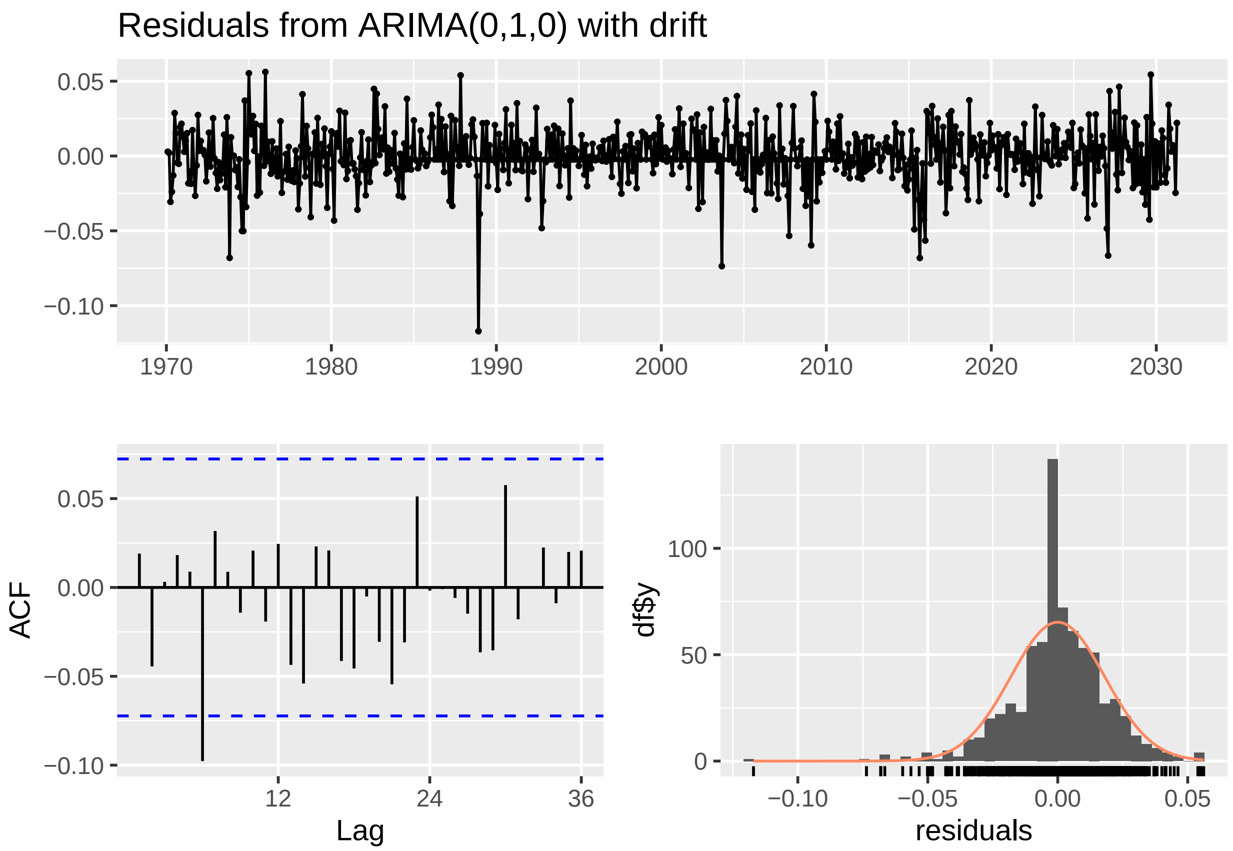
```
Ldowjones = log10(dowjones)
autoplot(Ldowjones)
```



```
Ldowjones_autoArima_fit = auto.arima(Ldowjones, approximation = FALSE, seasonal = TRUE, stepwise = FALSE


summary(Ldowjones_autoArima_fit)
```

```
## Series: Ldowjones
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.0023
## s.e.  0.0007
##
## sigma^2 = 0.0003281:  log likelihood = 1903.19
## AIC=-3802.38   AICc=-3802.36   BIC=-3793.18
##
## Training set error measures:
##                        ME      RMSE        MAE          MPE      MAPE      MASE
## Training set 3.92982e-06 0.0180881 0.01292583 -0.002732865 0.3610894 0.2404342
##                     ACF1
## Training set 0.01905663
```

```r
checkresiduals(Ldowjones_autoArima_fit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0) with drift
## Q* = 24.05, df = 24, p-value = 0.4588
##
## Model df: 0.   Total lags used: 24
```

```r
summary(auto.arima(Ldowjones, approximation = FALSE, seasonal = TRUE, stepwise = FALSE, allowdrift = FA
```

```
## Series: Ldowjones
## ARIMA(0,1,0)
##
## sigma^2 = 0.0003331:  log likelihood = 1897.15
## AIC=-3792.3   AICc=-3792.29   BIC=-3787.7
##
## Training set error measures:
##                       ME       RMSE        MAE        MPE      MAPE      MASE
## Training set 0.002332309 0.01823755 0.01293023 0.06206203 0.3603578 0.2405161
##                    ACF1
## Training set 0.01903661
```