Report 8

Khaled Hasan

2024-05-12

Data

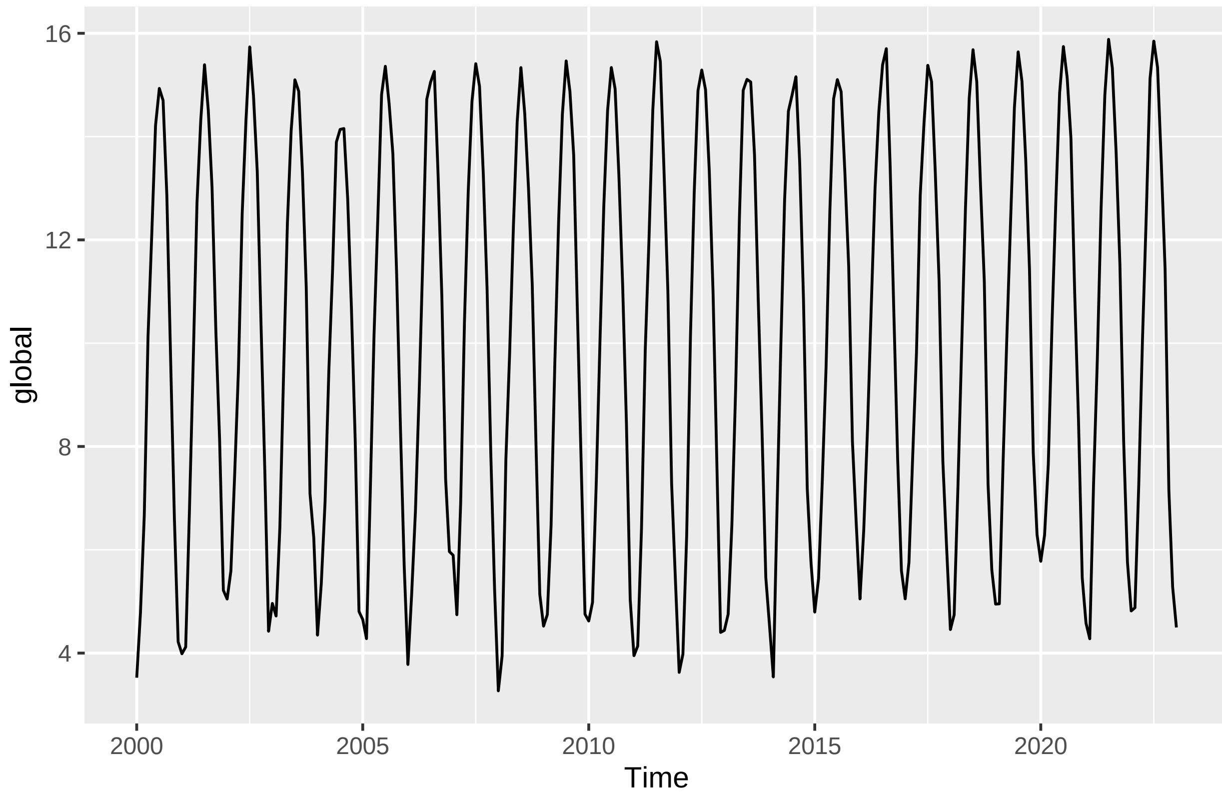
```
GLOBALTEMPERATURE = read.csv(file = "C:\\Users/ss/Desktop/Time_series_Analysis/MyGlobalTemperetures.csv
global_temp = ts(GLOBALTEMPERATURE[,1], start = c(1850, 1), frequency = 12)
northern_temp = ts(GLOBALTEMPERATURE[,2], start = c(1850, 1), frequency = 12)
southern_temp = ts(GLOBALTEMPERATURE[,3], start = c(1850, 1), frequency = 12)
Regional_temp = read.csv(file = "C:/Users/ss/Desktop/Time_series_Analysis/Regional_temperetures_data.csv
time = ts(Regional_temp[, 1], start = c(1850, 1), frequency = 12)
```

Plots

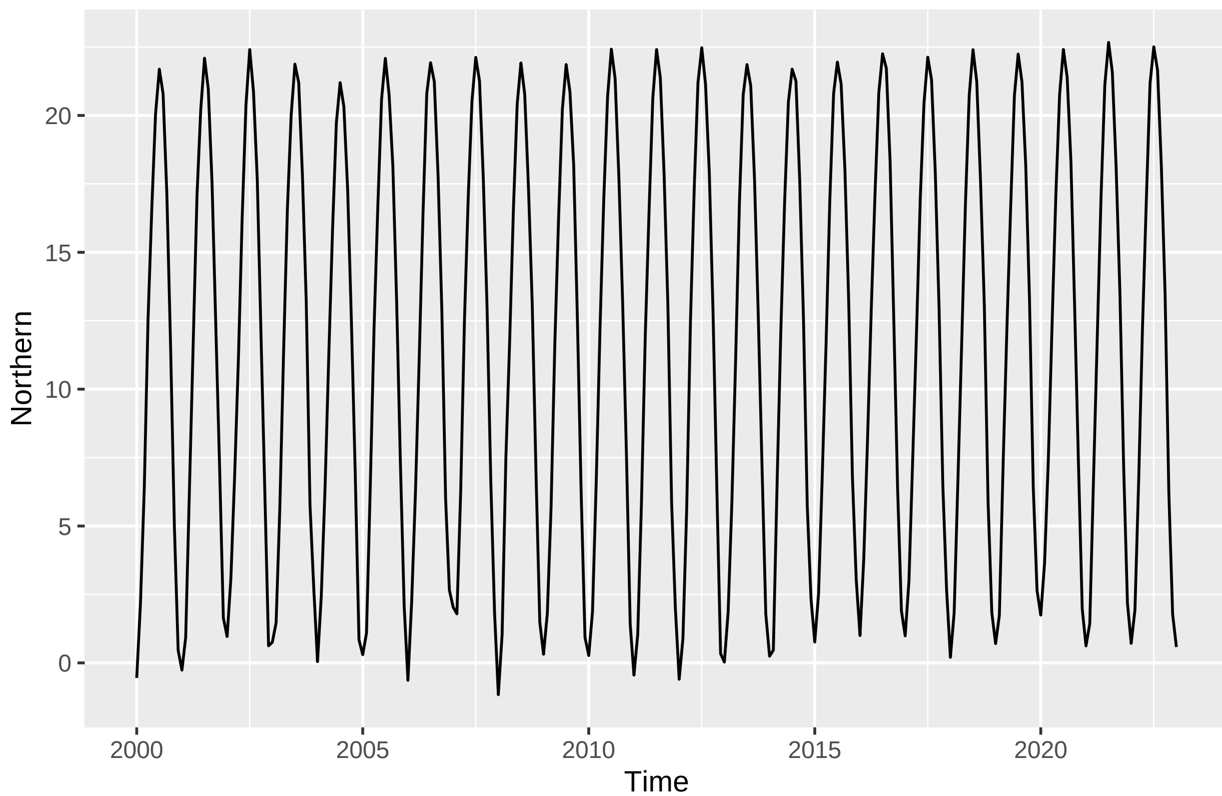
```
library(ggplot2)
library(gridExtra)
```

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

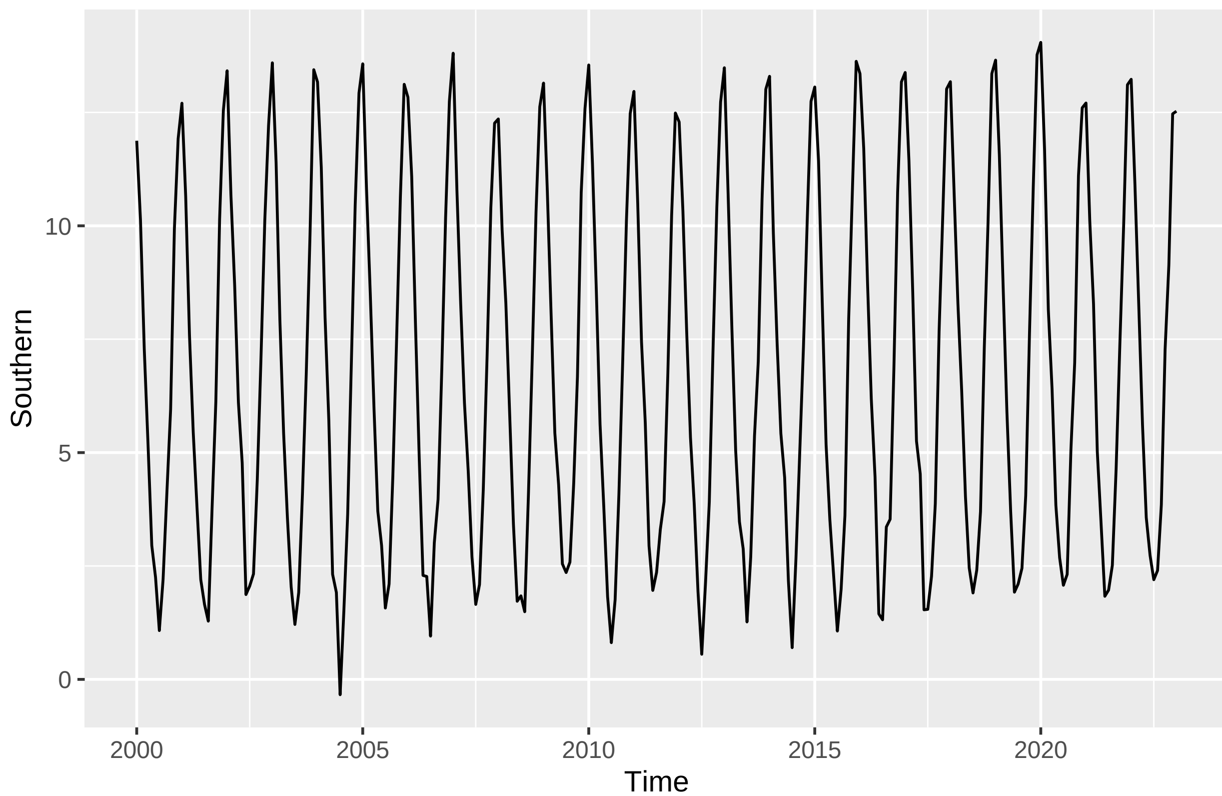
```
autoplot1 <- autoplot(window(global_temp, start = c(2000, 1), freq = 12), ylab = "global")

autoplot2 <- autoplot(window(northern_temp, start = c(2000, 1), freq = 12), ylab = "Northern")

autoplot3 <- autoplot(window(southern_temp, start = c(2000, 1), freq = 12), ylab = "Southern")

par(mfrow = c(1, 3))
plot(autoplot1)
```



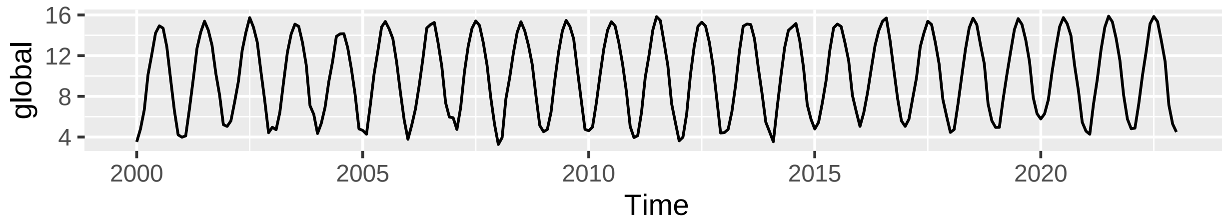

```r
plot(autoplot2)
```

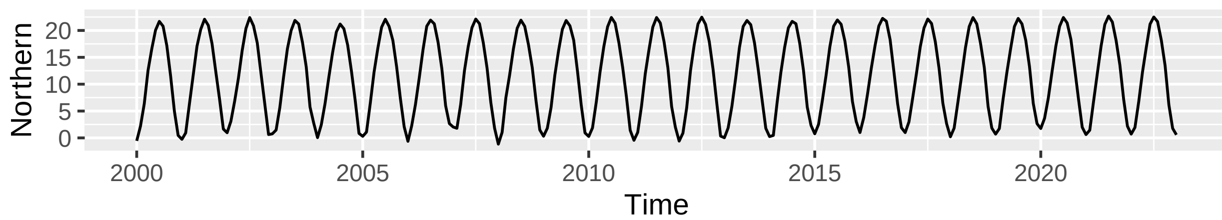


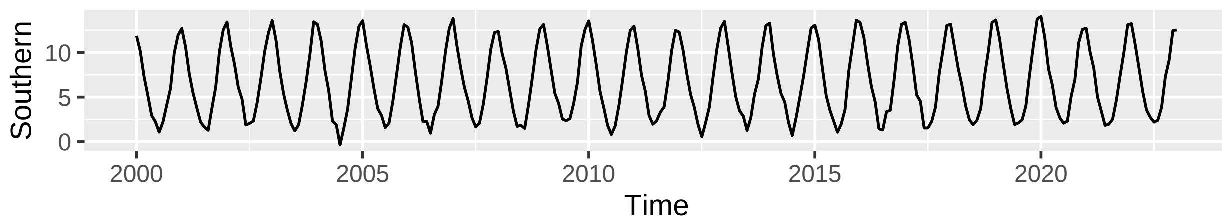

```r
plot(autoplot3)
```




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







Notice that the temperature averages in the southern hemi-sphere are out of phase with those from the northern Hemisphere, and that the global average is dominated by the northern hemisphere (northern hemisphere has more land mass).

plotting p-value

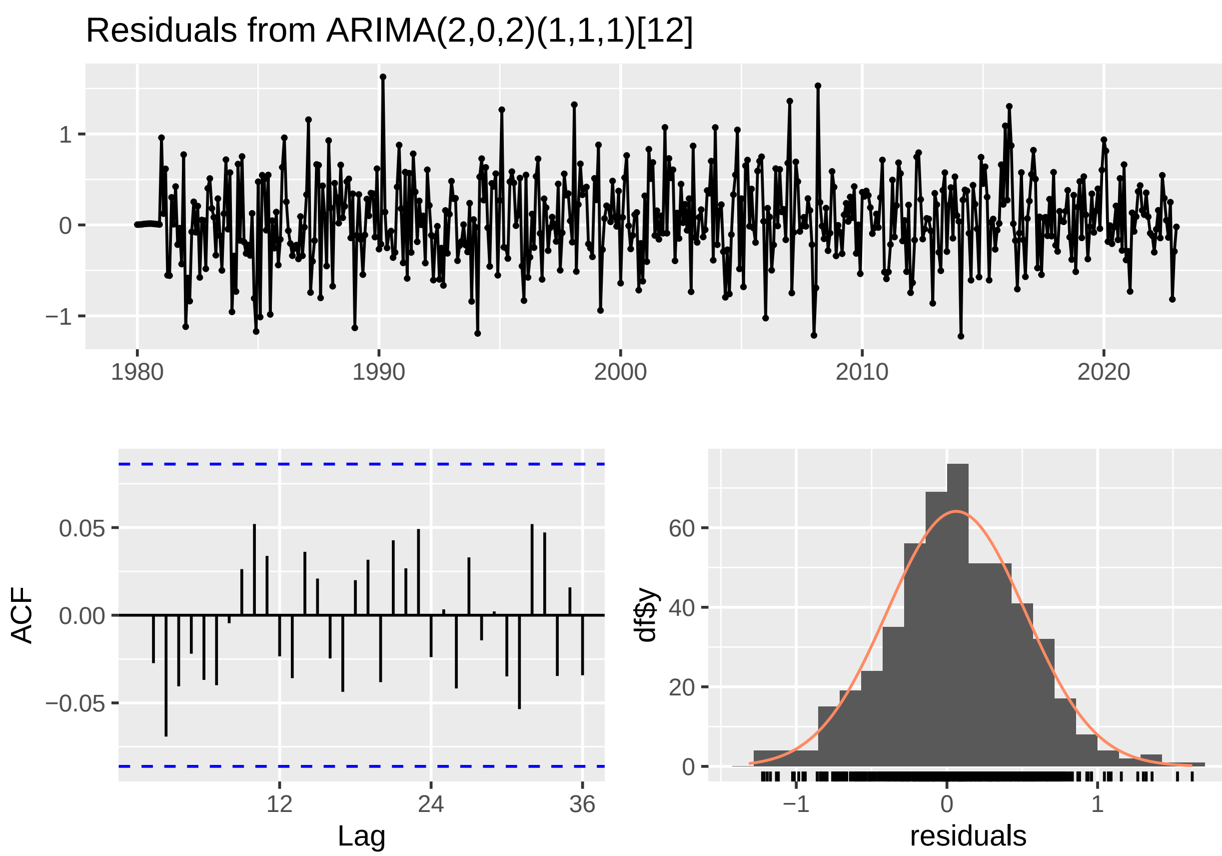
Finding best fitting model:

```
temp_1977_afterwards = window(global_temp, start=c(1987, 1), freq=12)

global_fitting_sarima = auto.arima(window(global_temp, start=c(1980, 1), freq=12), approximation = FALS

summary(global_fitting_sarima)
```

```
## Series: window(global_temp, start = c(1980, 1), freq = 12)
## ARIMA(2,0,2)(1,1,1)[12]
##
## Coefficients:
##           ar1     ar2     ma1      ma2     sar1     sma1
##       -0.0283  0.7152  0.2870  -0.4169  -0.0832  -0.8624
## s.e.   0.1128  0.1198  0.1323   0.1336   0.0511   0.0386
##
## sigma^2 = 0.2214:  log likelihood = -341.91
## AIC=697.83   AICc=698.05   BIC=727.4
```

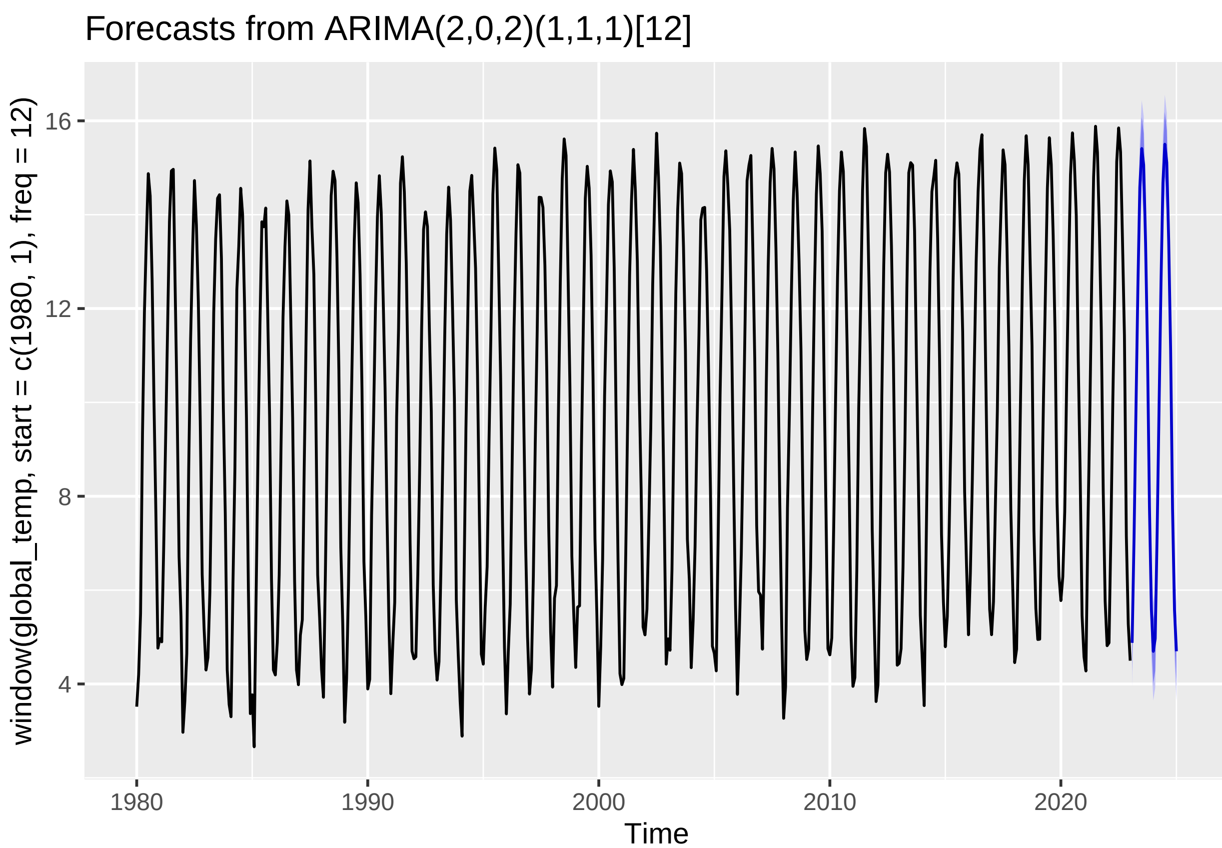
```
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.06138175 0.4623172 0.3587063 0.1622547 5.140828 0.6865682
##                      ACF1
## Training set 0.0006093937
```

```r
checkresiduals(global_fitting_sarima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2)(1,1,1)[12]
## Q* = 15.772, df = 18, p-value = 0.6085
##
## Model df: 6.   Total lags used: 24
```

```r
global_fitting_sarima %>% forecast(h = 24) %>% autoplot()
```




```r
minimum = global_fitting_sarima$aic
ord = c(2, 0, 2)
for (i in 0:5) {
  for (j in 0:5) {
    tord = c(i, 0, j)
    if(Arima(temp_1977_afterwards, tord, c(1, 1, 1), include.drift = TRUE)$aic < minimum){
      minimum = Arima(temp_1977_afterwards, tord, c(1, 1, 1), include.drift = TRUE)$aic
      ord = tord
    }
  }
}

model = Arima(temp_1977_afterwards, ord, c(1, 1, 1), include.drift = TRUE)
summary(model)
```

```
## Series: temp_1977_afterwards
## ARIMA(2,0,0)(1,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2     sar1     sma1   drift
##       0.2416  0.1595  -0.0652  -0.9419  0.0026
## s.e.  0.0483  0.0487   0.0541   0.0394  0.0003
##
## sigma^2 = 0.1983:  log likelihood = -268.08
## AIC=548.16   AICc=548.36   BIC=572.41
```

```
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.004285522 0.4364627 0.3305569 -0.642293 4.681702 0.6482557
##                     ACF1
## Training set 0.004505375
```

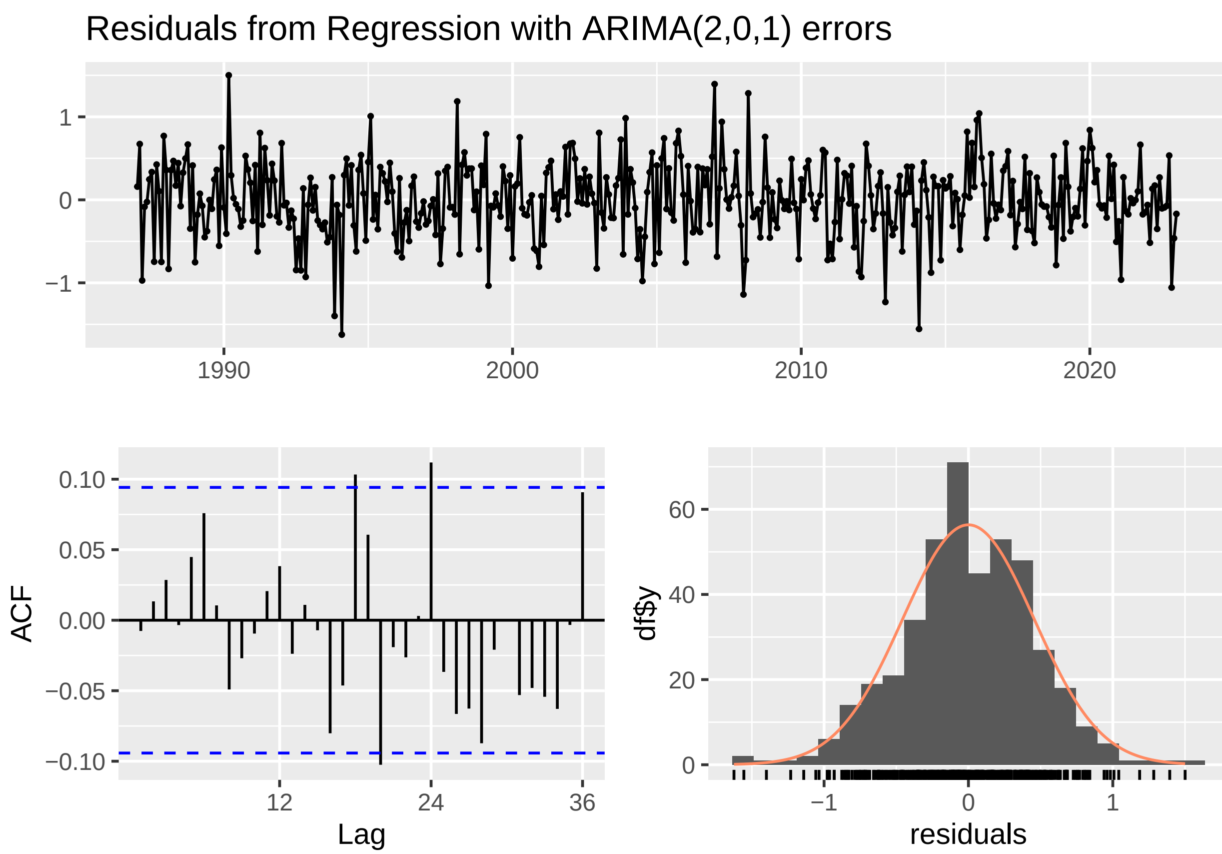
did not converge for aic measures (reached the max)

Fitting with regression:

```r
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))

summary(global_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(2,0,1) errors
##
## Coefficients:
##           ar1     ar2     ma1  intercept                          t
##       -0.5406  0.2813  0.7924     9.3065  -3.1557  -4.5193  0.0026
## s.e.   0.1670  0.0490  0.1702     0.0623   0.0404   0.0403  0.0003
##
## sigma^2 = 0.211:  log likelihood = -274.1
## AIC=564.2   AICc=564.54   BIC=596.77
##
## Training set error measures:
##                         ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.0002543913 0.4556364 0.354646 -0.662263 4.962195 0.6954969
##                     ACF1
## Training set -0.00761437
```

```r
checkresiduals(global_fitting_Arimareg)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,1) errors
## Q* = 27.979, df = 21, p-value = 0.1408
##
## Model df: 3.   Total lags used: 24
```

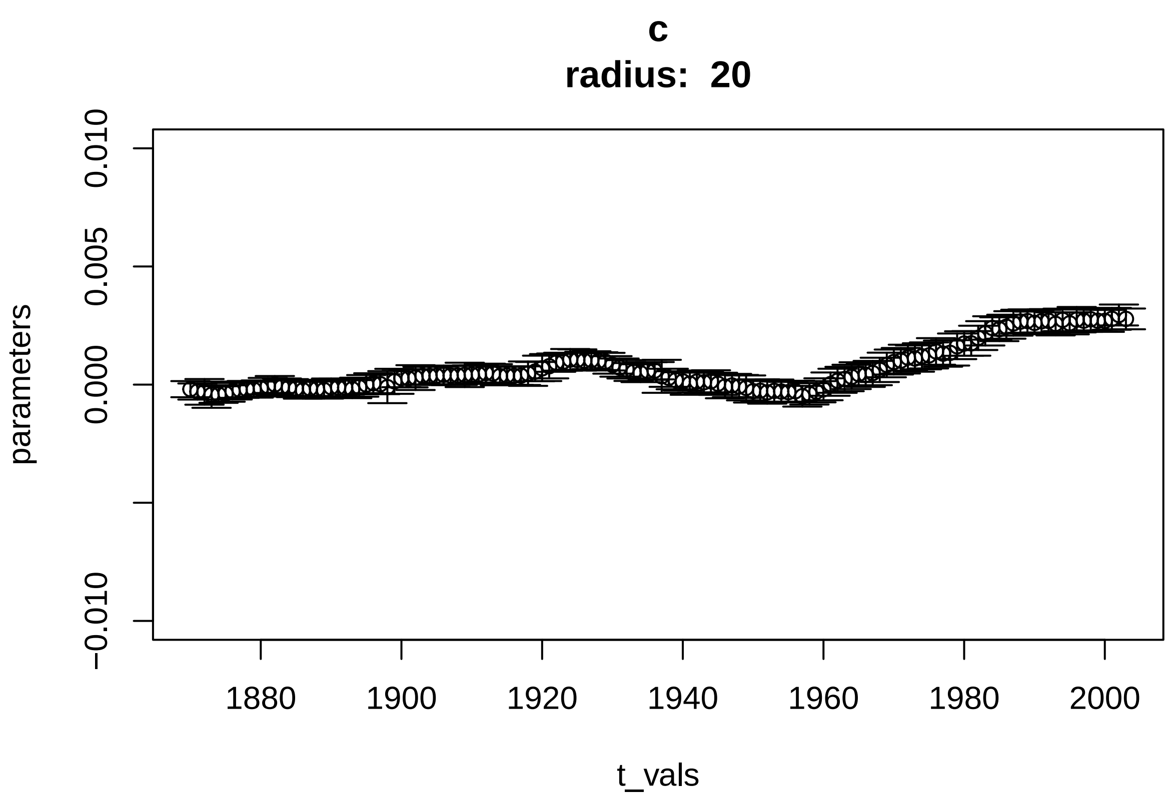
Plotting global warming coefficients

```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_
}
```

```r
plot_Global_warming <- function(timeseries, arima_fit){

  #arima_fit = Arima_fittng(timeseries, startingPoint = c(1980, 1))


  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()

  rad = 20

  sp = 1850

  fp = 2023

  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])
  }

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

  MyPlot = (plot(t_vals, parameters, type='b', main=paste("c\nradius: ", toString(rad)), ylim = c(-0.01
    arrows(x0=t_vals, y0=parameters-2*errors, x1 = t_vals, y1=parameters+2*errors, code=3, angle = 90, 

  return(list(arima_fit, parameters, errors, MyPlot))
}

Results_for_global = plot_Global_warming(global_temp, arima_fit = global_fitting_Arimareg)
```

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

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



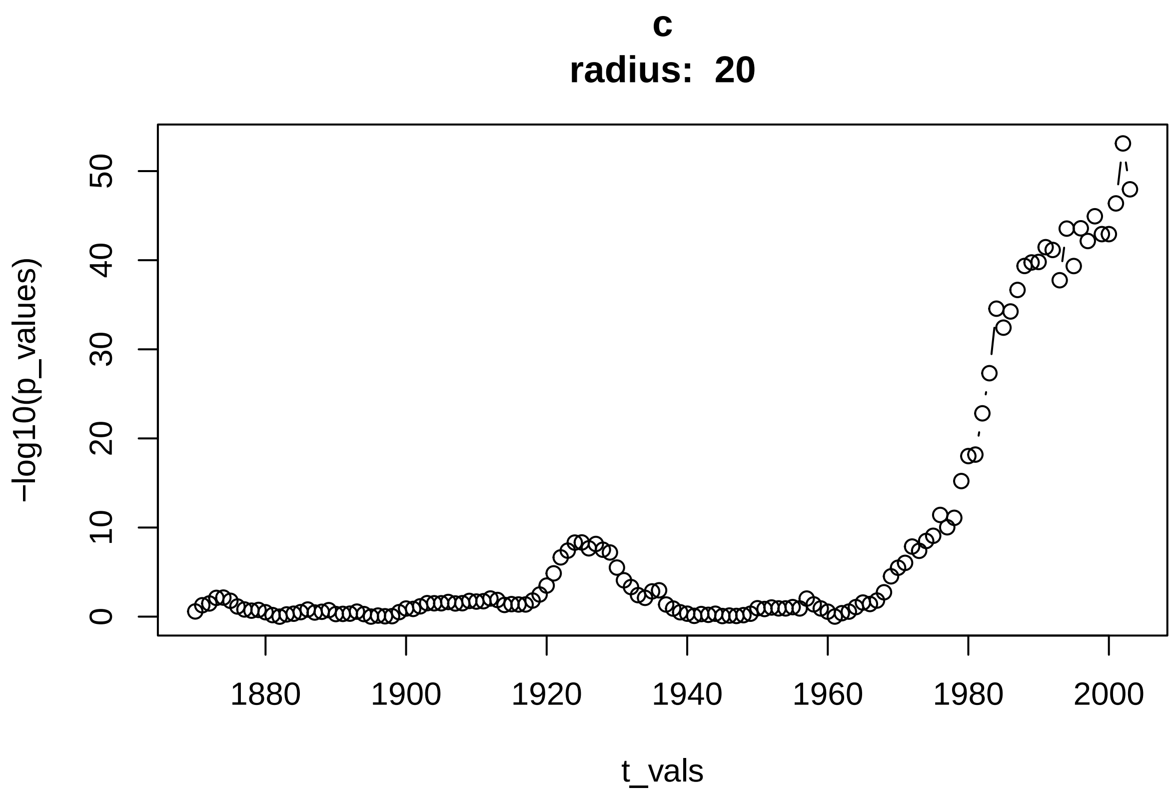
Plotting p-values

using the standard errors and assuming normal distribution (due to the big number of data points)

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

  #arima_fit = Arima_fittng(timeseries, startingPoint = c(1980, 1))

  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()
```

```r
  errors = c()

  p_values = c()

  rad = 20

  sp = 1850

  fp = 2023

  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("c\nradius: ", toString(rad)))

}

plot_p_vals(global_temp, arima_fit = global_fitting_Arimareg)
```

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

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


c
radius: 20
-log10(p_values)
0
10
20
30
40
50
1880
1900
1920
1940
1960
1980
2000
t_vals