Report10

Khaled Hasan

2024-06-03

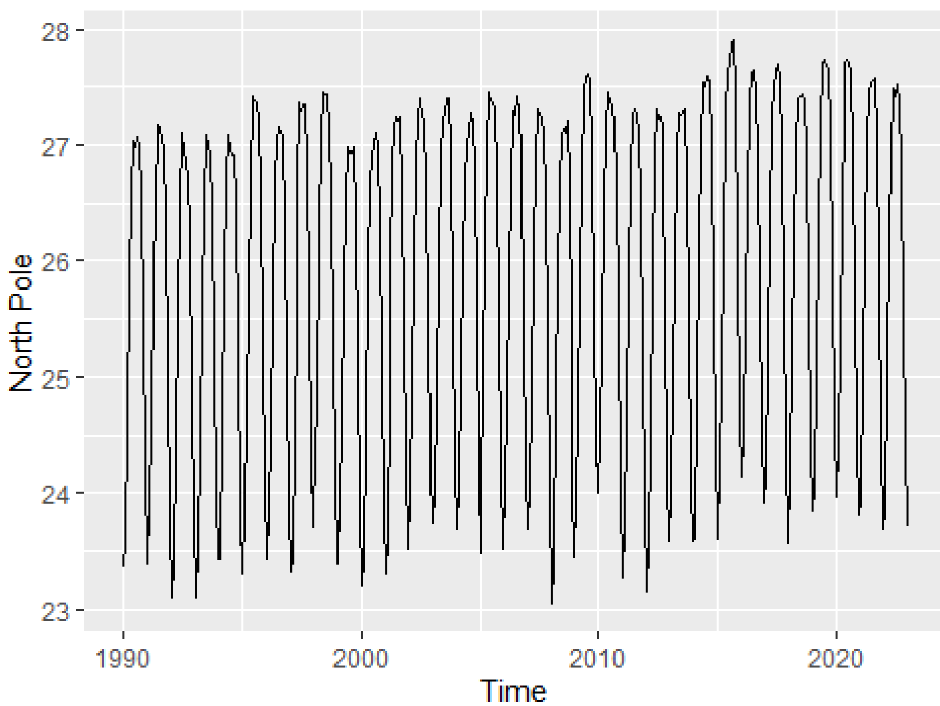
Data

```
GLOBALTEMPERATURE = read.csv(file =
"C:\\Users/ss/Desktop/Time_series_Analysis/MyGlobalTemperetures_incOcean.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/LatittudCuttedTemperetures.csv")
northPole = ts(LattitudinalTemps[, "X3"], start = c(1850, 1), frequency = 12)
north = ts(LattitudinalTemps[, "X2"], start = c(1850, 1), frequency = 12)
trop_north = ts(LattitudinalTemps[, "X1"], start = c(1850, 1), frequency =
12)
trop_south = ts(LattitudinalTemps[, "X4"], start = c(1850, 1), frequency =
12)
south = ts(LattitudinalTemps[, "X5"], start = c(1850, 1), frequency = 12)
southPole = ts(LattitudinalTemps[, "X6"], start = c(1850, 1), frequency = 12)
```

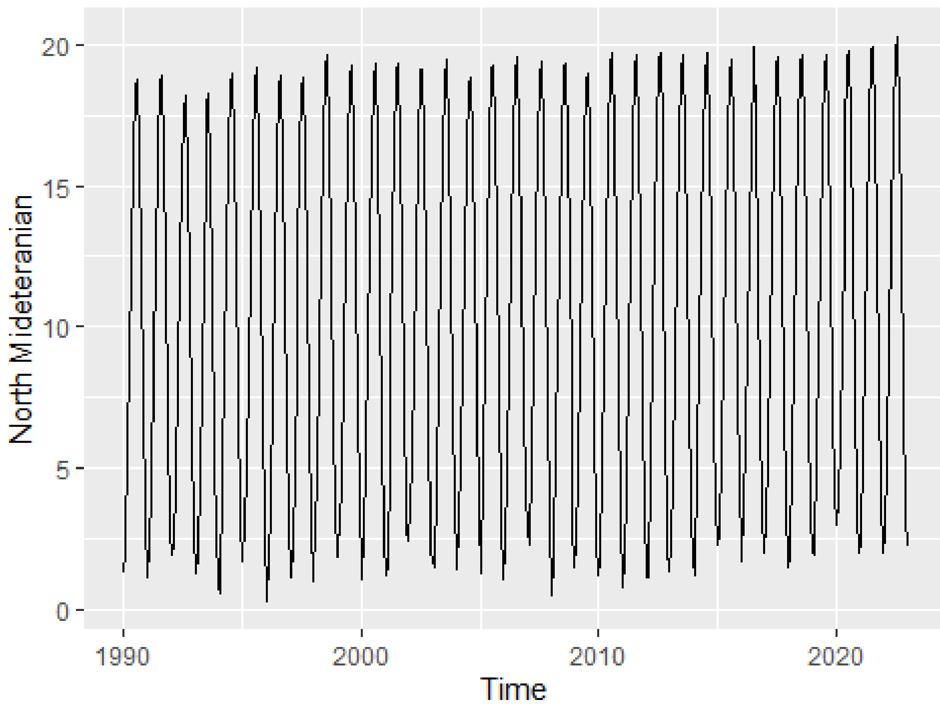
Plots

```
library(ggplot2)
library(gridExtra)



autoplot1 <- autoplot(window(northPole, start = c(1990, 1), freq = 12), ylab
= "North Pole")

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

autoplot3 <- autoplot(window(trop_north, start = c(1990, 1), freq = 12), ylab
= "above, tropical")

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

autoplot5 <- autoplot(window(south, start = c(1990, 1), freq = 12), ylab =
"southern middle strip")

autoplot6 <- autoplot(window(southPole, start = c(1990, 1), freq = 12), ylab
```

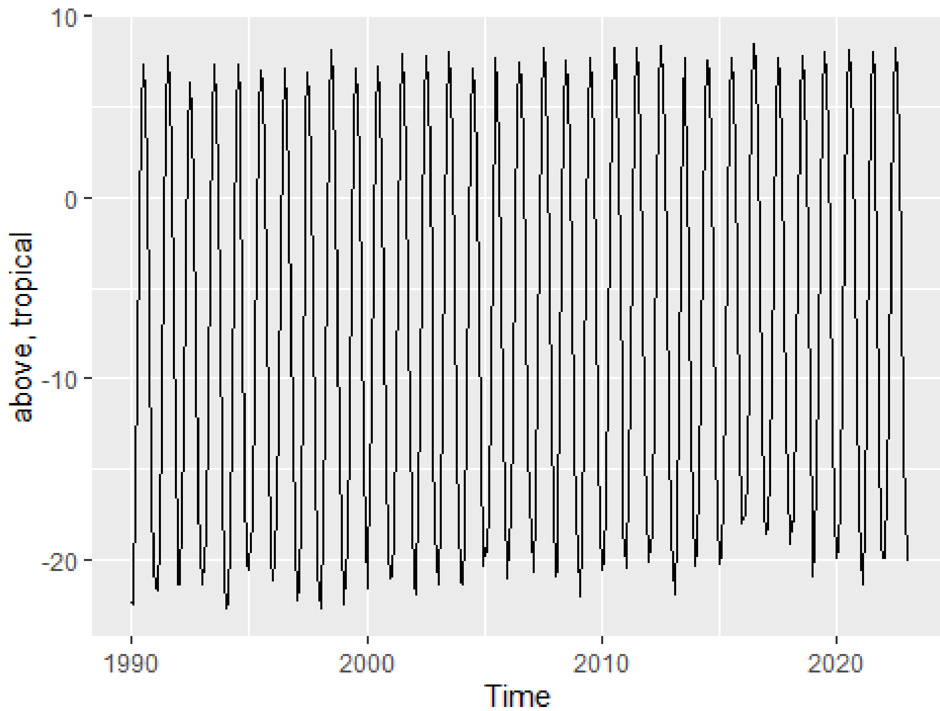
```
= "south pole")


par(mfrow = c(1, 6))
plot(autoplot1)
```



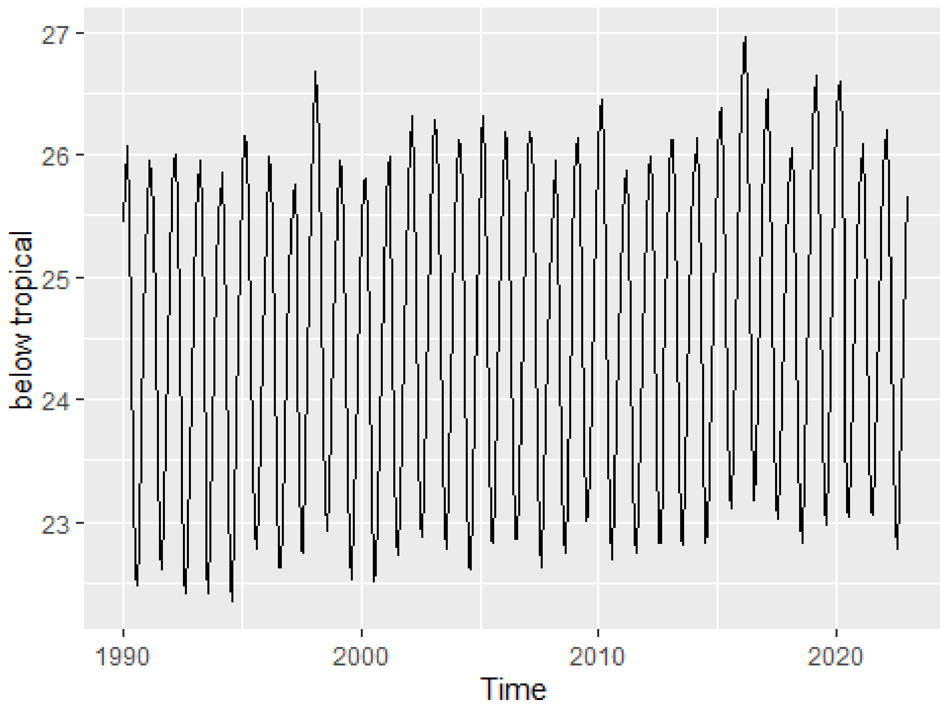

```
plot(autoplot2)
```



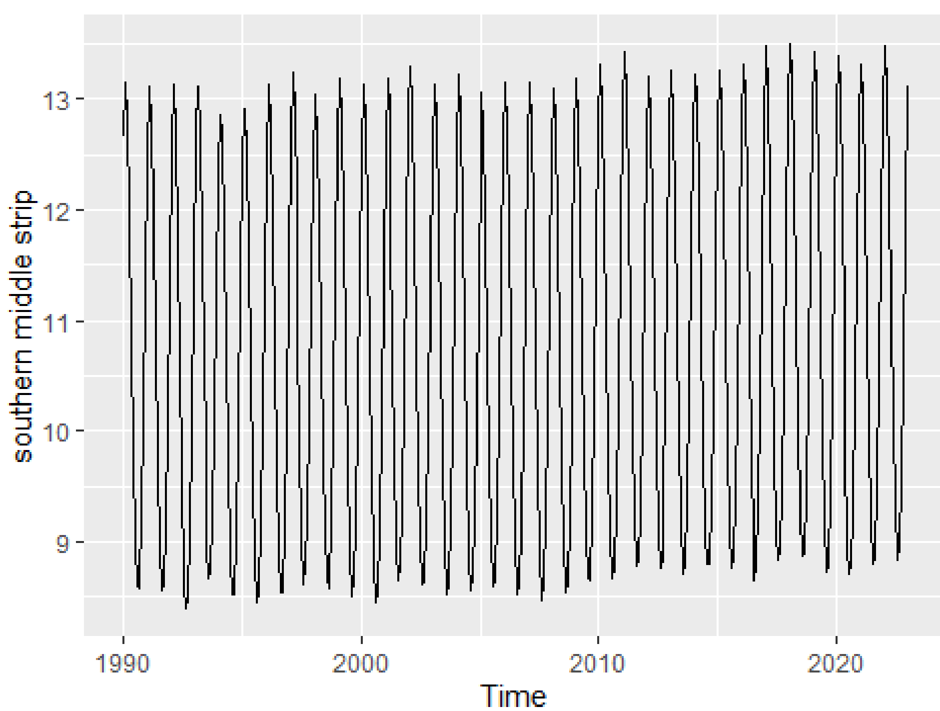

```
plot(autoplot3)
```



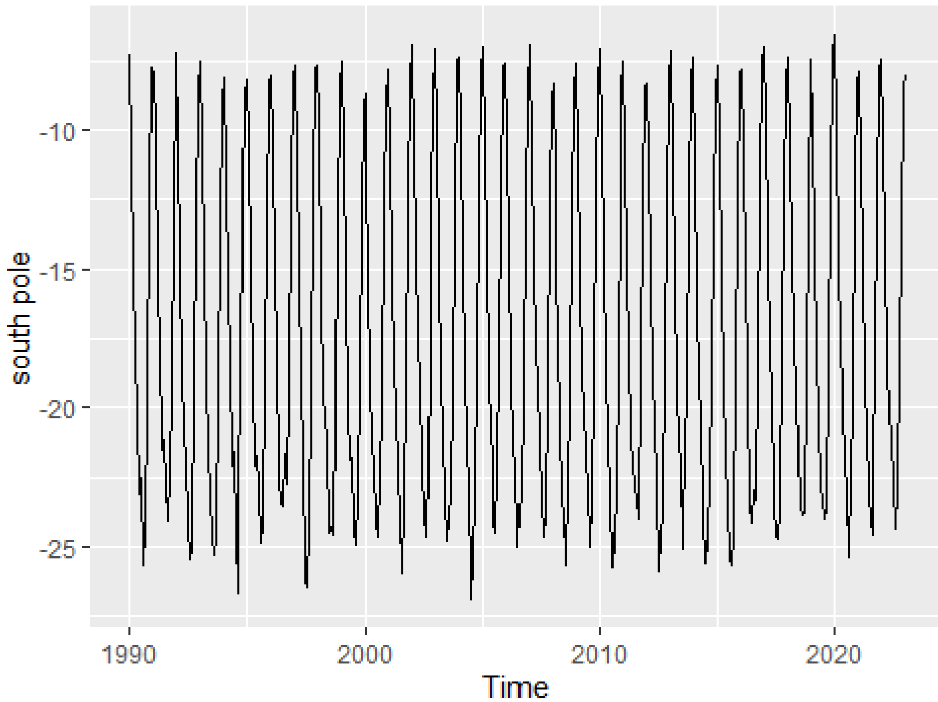

```
plot(autoplot4)
```



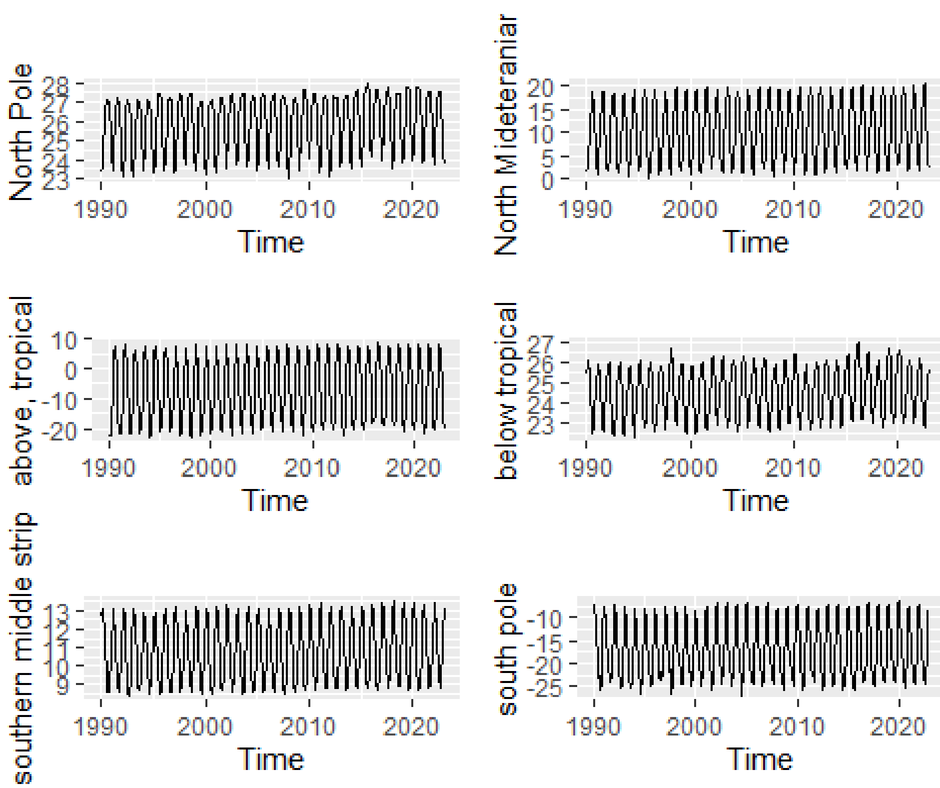

```r
plot(autoplot5)
```




```r
plot(autoplot6)
```




```
layout(matrix(c(1, 2, 3, 4, 5), nrow = 1))
grid.arrange(autoplot1, autoplot2, autoplot3, autoplot4, autoplot5,
autoplot6)
```



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))

summary(global_fitting_Arimareg)

## Series: cutted_data
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##           ar1     ar2     ma1      ma2  intercept
t
##       -0.1544  0.7115  0.4969  -0.3345    14.3399  -0.9728  -1.5103
0.0017
## s.e.   0.0908  0.0912  0.1142   0.1103     0.0395   0.0144   0.0144
0.0002
##
## sigma^2 = 0.02564:  log likelihood = 182.63
## AIC=-347.27   AICc=-346.84   BIC=-310.63
##
## Training set error measures:
##                          ME      RMSE       MAE         MPE      MAPE
MASE
## Training set -0.0002747135 0.1586278 0.1256687 -0.01510298 0.8773863
0.6323278
##                    ACF1
## Training set 0.01626736
```

Different Earth strips

- northPole: latitude from 60^o to 90^o.
- north: latitude from 30^o to 60^o.
- trop_north: latitude from 0^o to 30^o.
- trop_south: latitude from 0^o to -30^o.
- south: latitude from -30^o to -60^o.
- southPole: latitude from -60^o to -90^o.

```
northPole_fitting_Arimareg = Arima_fittng(northPole, startingPoint = c(1980,
1))
```

```r
north_fitting_Arimareg = Arima_fittng(north, startingPoint = c(1980, 1))
trop_north_fitting_Arimareg = Arima_fittng(trop_north, startingPoint =
c(1980, 1))
trop_south_fitting_Arimareg = Arima_fittng(trop_south, startingPoint =
c(1980, 1))
south_fitting_Arimareg = Arima_fittng(south, startingPoint = c(1980, 1))
southPole_fitting_Arimareg = Arima_fittng(southPole, startingPoint = c(1980,
1))

# reorder = 1, 3, 4, 6, 7, 2, 5

model_details <- data.frame(c(northPole_fitting_Arimareg$arma),
c(north_fitting_Arimareg$arma), c(trop_north_fitting_Arimareg$arma),
c(trop_south_fitting_Arimareg$arma),c(south_fitting_Arimareg$arma),
c(southPole_fitting_Arimareg$arma))

model_details

##   c.northPole_fitting_Arimareg.arma. c.north_fitting_Arimareg.arma.
## 1                                  1                              1
## 2                                  2                              0
## 3                                  1                              2
## 4                                  0                              0
## 5                                 12                             12
## 6                                  0                              0
## 7                                  0                              0
##   c.trop_north_fitting_Arimareg.arma. c.trop_south_fitting_Arimareg.arma.
## 1                                   1                                   1
## 2                                   0                                   0
## 3                                   2                                   2
## 4                                   0                                   0
## 5                                  12                                  12
## 6                                   0                                   0
## 7                                   0                                   0
##   c.south_fitting_Arimareg.arma. c.southPole_fitting_Arimareg.arma.
## 1                              5                                  5
## 2                              0                                  1
## 3                              2                                  2
## 4                              0                                  0
## 5                             12                                 12
## 6                              0                                  0
## 7                              0                                  0

summary(northPole_fitting_Arimareg)

## Series: cutted_data
## Regression with ARIMA(1,0,2)(1,0,0)[12] errors
##
## Coefficients:
##          ar1      ma1      ma2     sar1   intercept
t
```

```
##       0.8355  -0.2156  -0.0751  0.8515    25.5179  -1.2014  -1.4957
0.0011
## s.e.  0.0393   0.0620   0.0489  0.0243     0.3406   0.1057   0.1056
0.0011
##
## sigma^2 = 0.0325:  log likelihood = 148
## AIC=-277.99   AICc=-277.64   BIC=-239.76
##
## Training set error measures:
##                         ME      RMSE       MAE          MPE      MAPE
MASE
## Training set -0.0002161595 0.1788706 0.1425854 -0.005758032 0.5606875
0.6746827
##                     ACF1
## Training set 0.004786892
```

```
summary(north_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                         t
##       0.3540  0.2984  0.3187     9.5607  -5.8871  -6.6481  0.0026
## s.e.  0.0412  0.0415  0.0422     0.1436   0.0992   0.0990  0.0005
##
## sigma^2 = 0.2151:  log likelihood = -335.49
## AIC=686.99   AICc=687.27   BIC=720.97
##
## Training set error measures:
##                       ME     RMSE       MAE       MPE     MAPE      MASE
## Training set -0.00134444 0.460656 0.3501063 -4.009222 11.52556 0.7855967
##                     ACF1
## Training set -0.01689206
```

```
summary(trop_north_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1   sar1    sar2  intercept                          t
##       0.2427  0.261  0.3444    -8.6697  -9.2434  -11.3177  0.0044
## s.e.  0.0430  0.042  0.0434     0.2417   0.1737    0.1734  0.0008
##
## sigma^2 = 0.8729:  log likelihood = -697.5
## AIC=1411   AICc=1411.28   BIC=1444.98
##
## Training set error measures:
##                      ME     RMSE       MAE      MPE     MAPE      MASE
## Training set -0.0110331 0.927926 0.6803327 13.01068 24.02486 0.7585858
```

```
##                      ACF1
## Training set -0.01237397
```

```r
summary(trop_south_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                      t
##       0.8580  0.4682  0.3572    24.2721  1.4509  0.8094  7e-04
## s.e.  0.0224  0.0407  0.0415     0.3159  0.0776  0.0775  1e-03
##
## sigma^2 = 0.0158:  log likelihood = 335.27
## AIC=-654.54   AICc=-654.25   BIC=-620.55
##
## Training set error measures:
##                        ME      RMSE        MAE          MPE      MAPE
MASE
## Training set 0.0001207197 0.1248272 0.09950499 -0.00154996 0.4058563
0.4645412
##                      ACF1
## Training set -0.04137371
```

```r
summary(south_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(5,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1      ar2      ar3     ar4     ar5    sar1    sar2  intercept
##       0.7185  -0.0490  -0.0717  0.0208  0.1387  0.4342  0.3668    10.5279
## s.e.  0.0463   0.0627   0.0587  0.0566  0.0523  0.0450  0.0475     0.1099
##                           t
##       2.1043  0.8898  8e-04
## s.e.  0.0360  0.0360  3e-04
##
## sigma^2 = 0.006761:  log likelihood = 557.55
## AIC=-1091.1   AICc=-1090.48   BIC=-1040.13
##
## Training set error measures:
##                        ME      RMSE        MAE          MPE      MAPE
MASE
## Training set -0.001588887 0.0813433 0.06389744 -0.01906295 0.6081312
0.6166872
##                      ACF1
## Training set 0.006445156
```

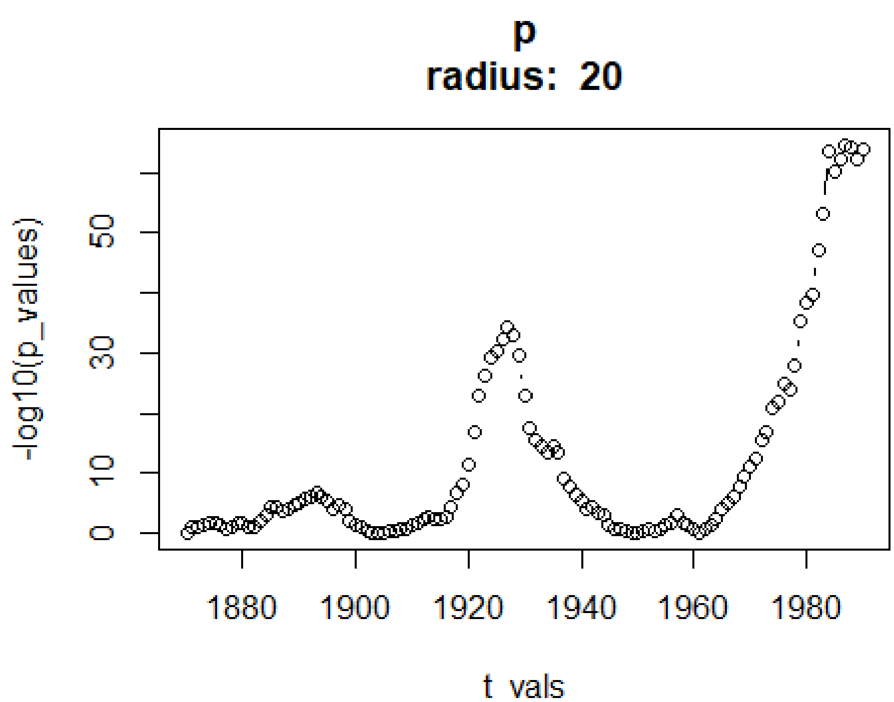
```r
summary(southPole_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(5,0,1)(2,0,0)[12] errors
##
## Coefficients:
##          ar1      ar2      ar3     ar4     ar5      ma1    sar1    sar2
##       0.7339  -0.2415  -0.2912  0.1170  0.2816  -0.3871  0.1637  0.1381
## s.e.  0.0777   0.0634   0.0543  0.0623  0.0463   0.0713  0.0523  0.0528
##       intercept                      t
##        -17.4089  4.2594  7.0029  6e-04
## s.e.     0.1554  0.0644  0.0644  5e-04
##
## sigma^2 = 0.7201:  log likelihood = -644.34
## AIC=1314.68   AICc=1315.4   BIC=1369.9
##
## Training set error measures:
##                         ME      RMSE       MAE        MPE     MAPE
MASE
## Training set -0.008611571 0.8386841 0.6681146 -0.3543008 4.296067
0.8314695
##                     ACF1
## Training set -0.02980657
```

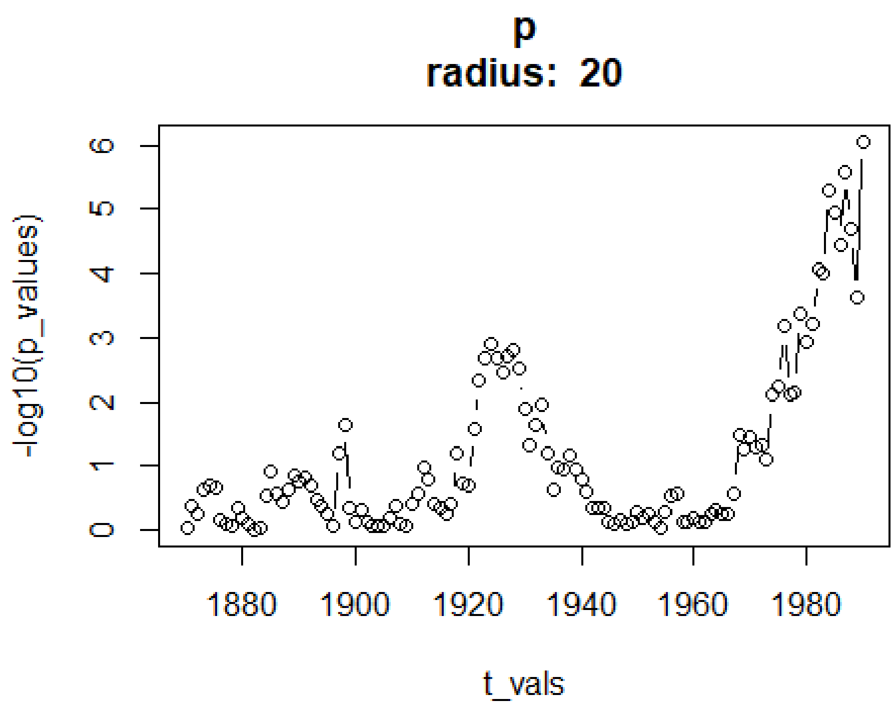
```
# Helping functions

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_t"])))
}


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(-0.010, 0.010)) +
    arrows(x0=t_vals, y0=parameters-2*errors, x1 = t_vals,
y1=parameters+2*errors, code=3, angle = 90, length = 0.1))

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


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
```

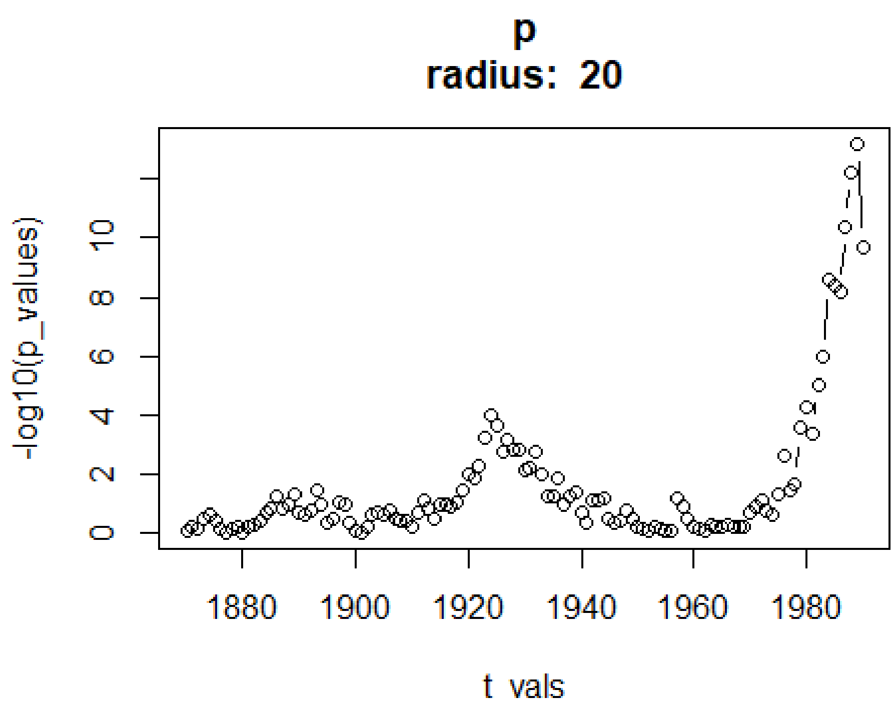
```
  fp = 2010

  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)
```



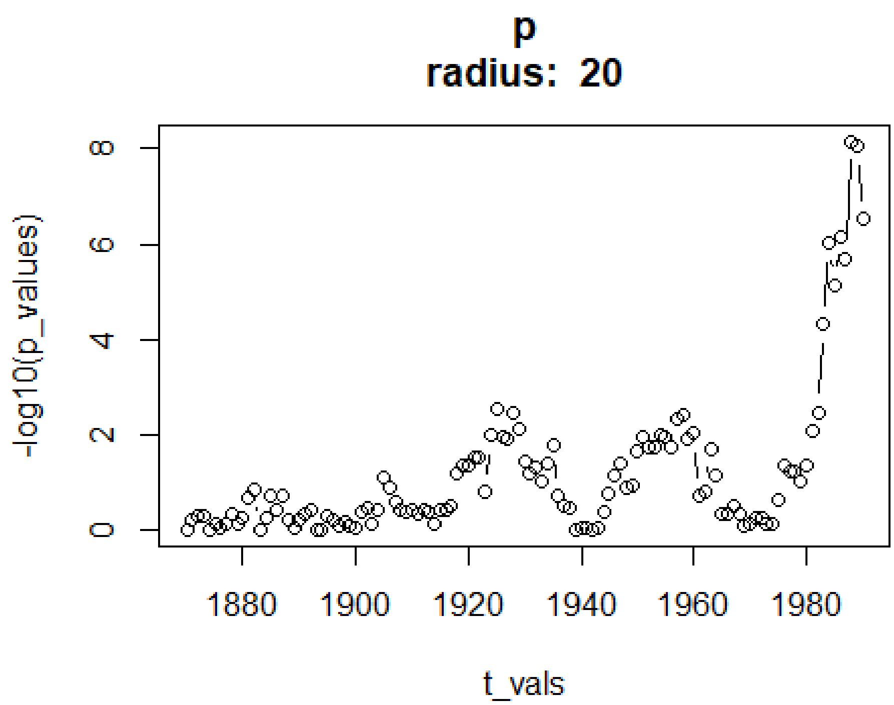

```
plot_p_vals(northPole, arima_fit = northPole_fitting_Arimareg, 20)
```



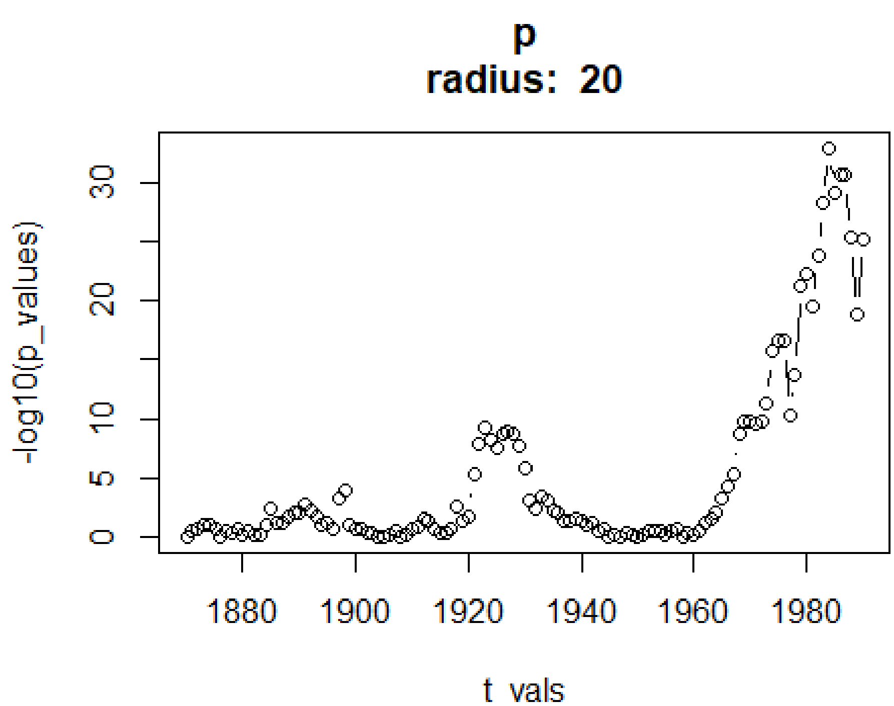

```
plot_p_vals(north, arima_fit = north_fitting_Arimareg, 20)
```



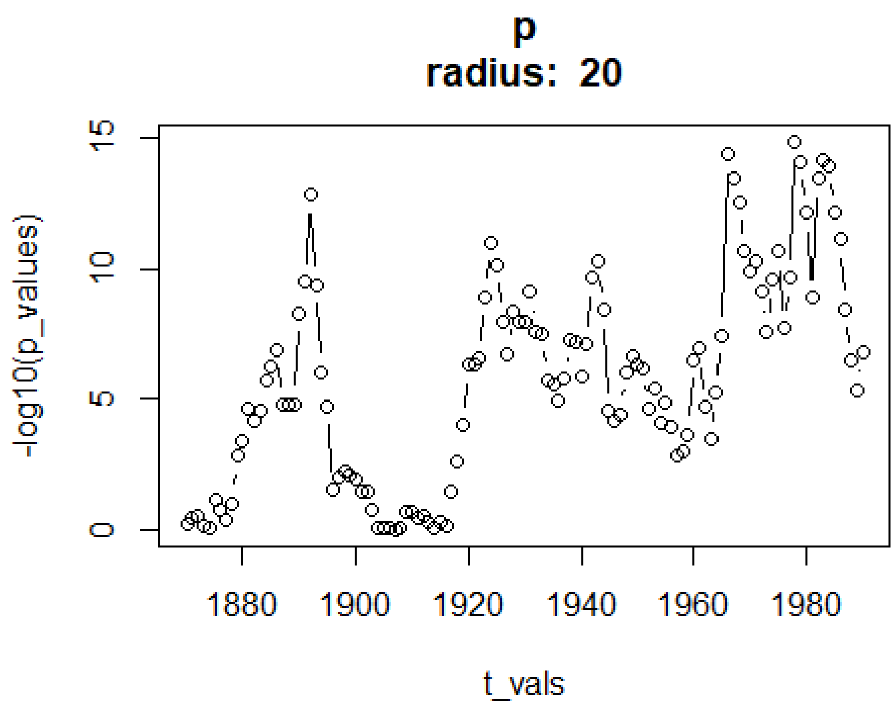

```
plot_p_vals(trop_north, arima_fit = trop_north_fitting_Arimareg, 20)
```




```
plot_p_vals(trop_south, arima_fit = trop_south_fitting_Arimareg, 20)
```




```
plot_p_vals(south, arima_fit = south_fitting_Arimareg, 20)
```




```
# plot_p_vals(southPole, arima_fit = southPole_fitting_Arimareg, 20)
```

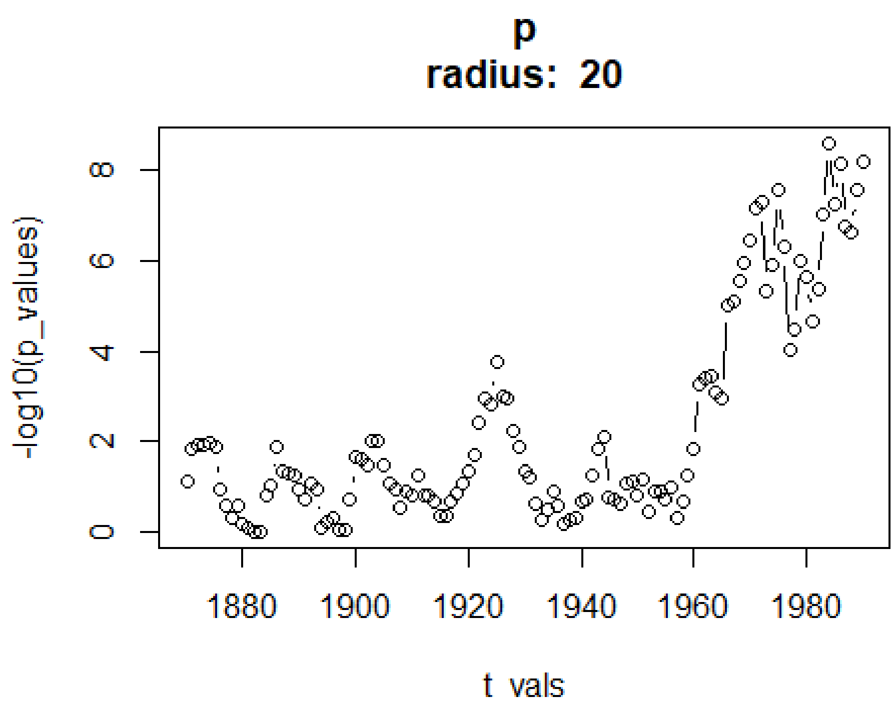
error, did not work

The “south” region shows no indication of Global warming in terms of p-value, therefore, I will try to use the land Only date.

```
south_land_only =
ts(data=read.csv("C:/users/ss/Desktop/Time_series_Analysis/MidSouthRegion.csv
")[, 1], start = c(1850, 1), frequency = 12)

arima_for_south_land_only = Arima_fittng(timeseries = south_land_only,
startingPoint = c(1980))

summary(arima_for_south_land_only)

## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                          t
##       0.2876  0.2532  0.2217    14.8584  3.1531  4.6047  0.0014
## s.e.  0.0423  0.0431  0.0438     0.0960  0.0658  0.0657  0.0003
##
## sigma^2 = 0.1943:  log likelihood = -307.82
## AIC=631.64   AICc=631.92   BIC=665.62
##
```

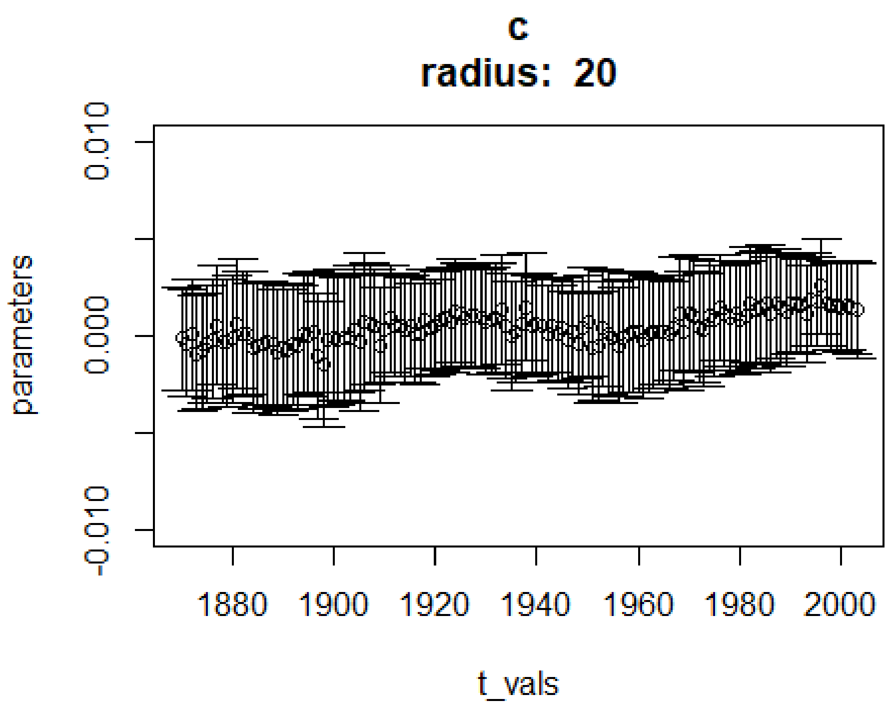
```
## Training set error measures:
##                          ME      RMSE       MAE        MPE     MAPE
MASE
## Training set -0.004253721 0.4377541 0.3454516 -0.1254056 2.483875
0.7636581
##                        ACF1
## Training set -0.003261445
```

```
plot_p_vals(south_land_only, arima_fit = arima_for_south_land_only, 20)
```

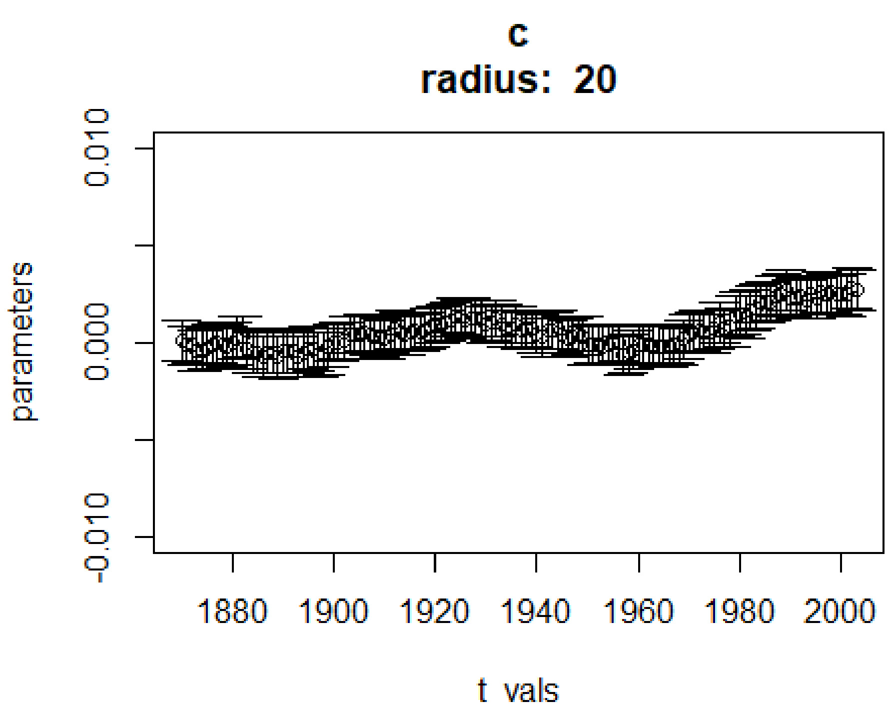


Plotting global warming Coeffiecients

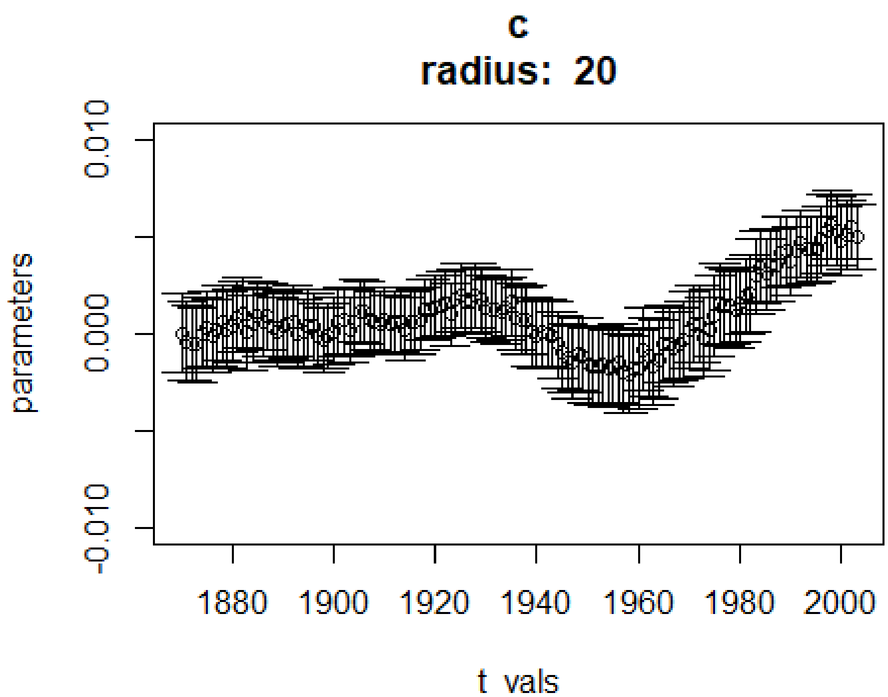
```
plot_Global_warming(global_temp, arima_fit = global_fitting_Arimareg)

## Error in arima(temporary_data, order = Ord, seasonal = sOrd, xreg =
temporary_xreg): non-stationary AR part from CSS

plot_Global_warming(northPole, arima_fit = northPole_fitting_Arimareg)
```




```
plot_Global_warming(north, arima_fit = north_fitting_Arimareg)
```

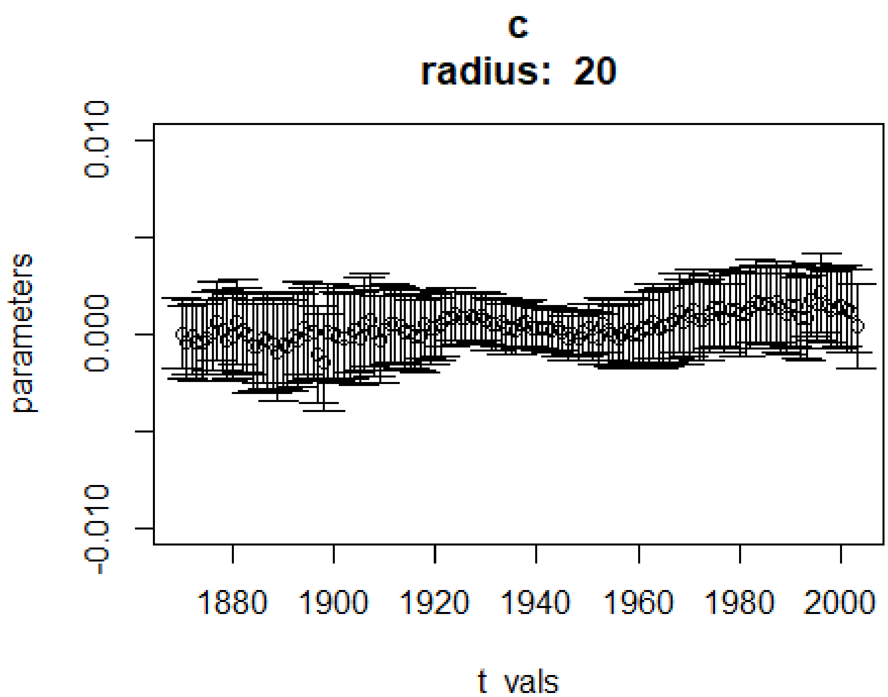


```
## [[1]]
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                         t
##       0.3540  0.2984  0.3187     9.5607  -5.8871  -6.6481  0.0026
## s.e.  0.0412  0.0415  0.0422     0.1436   0.0992   0.0990  0.0005
##
## sigma^2 = 0.2151:  log likelihood = -335.49
## AIC=686.99   AICc=687.27   BIC=720.97
##
plot_Global_warming(trop_north, arima_fit = trop_north_fitting_Arimareg)
```



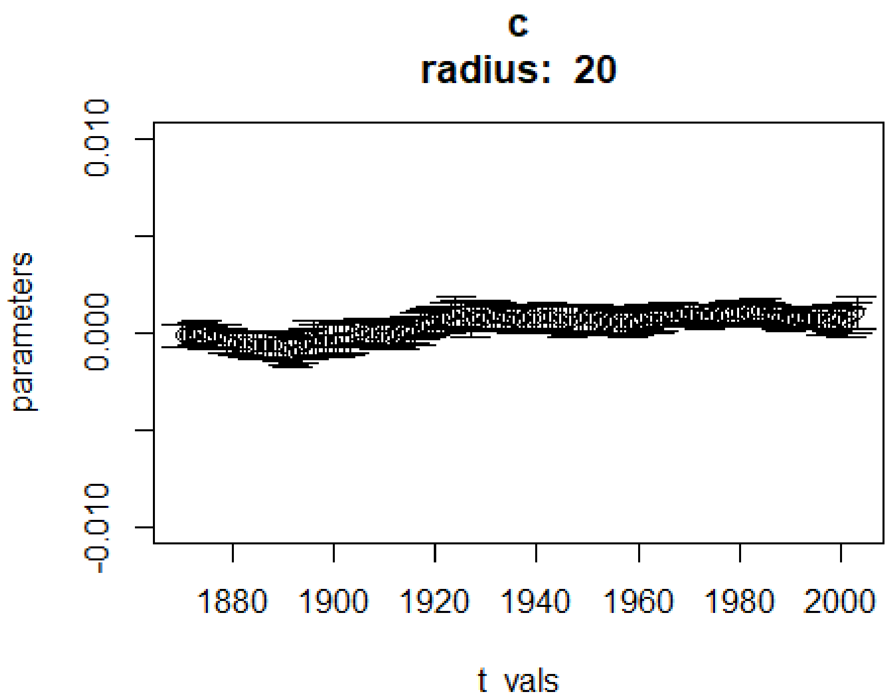

```
## [[1]]
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1   sar1    sar2  intercept                          t
##       0.2427  0.261  0.3444    -8.6697  -9.2434  -11.3177  0.0044
## s.e.  0.0430  0.042  0.0434     0.2417   0.1737    0.1734  0.0008
##
## sigma^2 = 0.8729:  log likelihood = -697.5
## AIC=1411   AICc=1411.28   BIC=1444.98
##
```

```r
plot_Global_warming(trop_south, arima_fit = trop_south_fitting_Arimareg)
```



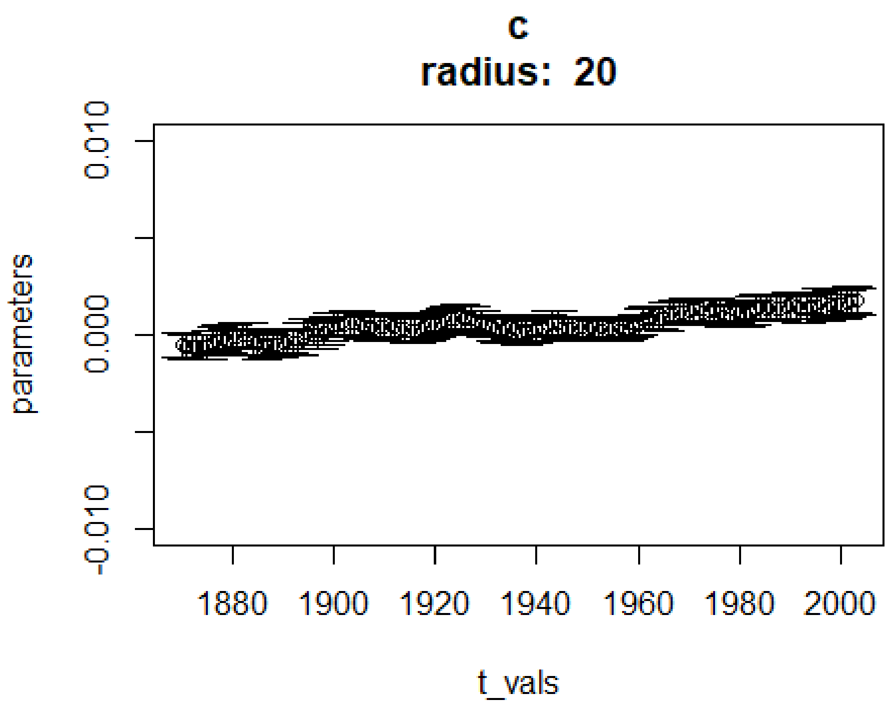

```
## [[1]]
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                          t
##       0.8580  0.4682  0.3572    24.2721  1.4509  0.8094  7e-04
## s.e.  0.0224  0.0407  0.0415     0.3159  0.0776  0.0775  1e-03
##
## sigma^2 = 0.0158:  log likelihood = 335.27
## AIC=-654.54   AICc=-654.25   BIC=-620.55
##
```

```r
plot_Global_warming(south, arima_fit = south_fitting_Arimareg)
```




```
## [[1]]
## Series: cutted_data
## Regression with ARIMA(5,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1      ar2      ar3     ar4     ar5    sar1    sar2  intercept
##       0.7185  -0.0490  -0.0717  0.0208  0.1387  0.4342  0.3668    10.5279
## s.e.  0.0463   0.0627   0.0587  0.0566  0.0523  0.0450  0.0475     0.1099
##                             t
##       2.1043  0.8898  8e-04
## s.e.  0.0360  0.0360  3e-04
##
## sigma^2 = 0.006761:  log likelihood = 557.55
## AIC=-1091.1   AICc=-1090.48   BIC=-1040.13
##
## [[2]]
##                [,1]         [,2]          [,3]          [,4]         [,5]
## [1,] -0.0001491422 -0.000146809 -0.0001278082 -4.204515e-05 3.694682e-05
##                [,6]          [,7]          [,8]          [,9]         [,10]
## [1,] -0.0002509047 -0.0002015537 -0.0001644547 -0.0002131142 -0.0004256456
##               [,11]       [,12]         [,13]         [,14]         [,15]
## [1,] -0.0004876243 -0.00059613 -0.0005643029 -0.0005785155 -0.0006506932
```

```
##                [,16]          [,17]          [,18]          [,19]          [,20]
plot_Global_warming(south_land_only, arima_fit = arima_for_south_land_only)
```




```
## [[1]]
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                        t
##       0.2876  0.2532  0.2217    14.8584  3.1531  4.6047  0.0014
## s.e.  0.0423  0.0431  0.0438     0.0960  0.0658  0.0657  0.0003
##
## sigma^2 = 0.1943:  log likelihood = -307.82
## AIC=631.64   AICc=631.92   BIC=665.62
##
```

Conclusions:

The ocean data significantly decreases the warming trend coefficient. South pole data is highly unstable non stationary.

Separating Months:

```
MonthsSeparatedData =
read.csv("C:/Users/ss/Desktop/Time_series_Analysis/MyGlobalTemperetures_incOc
```

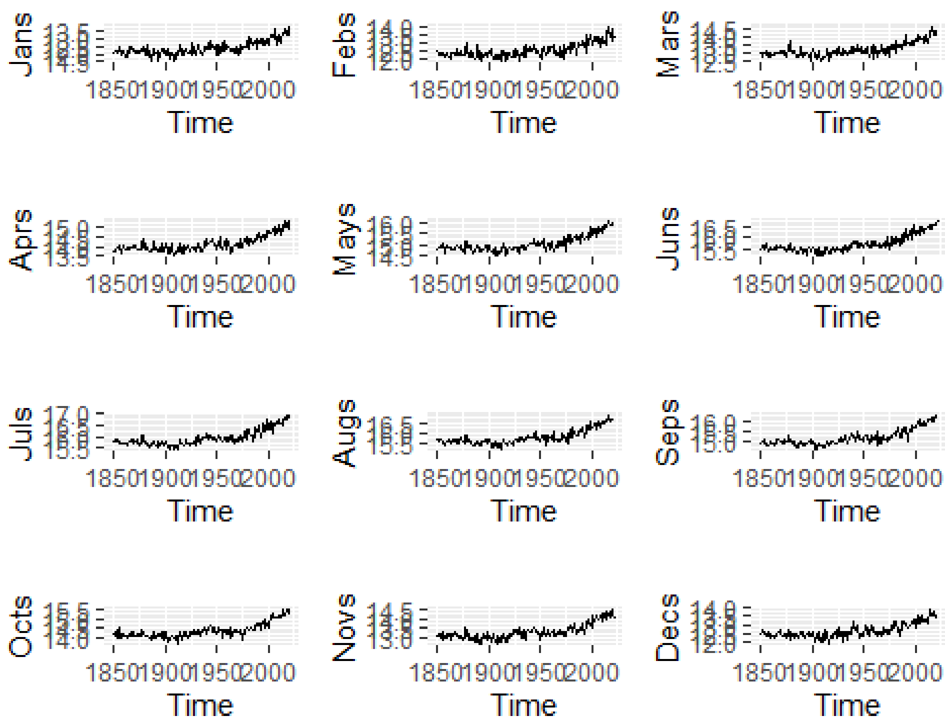
```
ean_sepByMonth.csv")
Jans = ts(data = MonthsSeparatedData[, 1], start = c(1850), end = c(2022),
frequency = 1)
Febs = ts(data = MonthsSeparatedData[, 2], start = c(1850), end = c(2022),
frequency = 1)
Mars = ts(data = MonthsSeparatedData[, 3], start = c(1850), end = c(2022),
frequency = 1)
Aprs = ts(data = MonthsSeparatedData[, 4], start = c(1850), end = c(2022),
frequency = 1)
Mays = ts(data = MonthsSeparatedData[, 5], start = c(1850), end = c(2022),
frequency = 1)
Juns = ts(data = MonthsSeparatedData[, 6], start = c(1850), end = c(2022),
frequency = 1)
Juls = ts(data = MonthsSeparatedData[, 7], start = c(1850), end = c(2022),
frequency = 1)
Augs = ts(data = MonthsSeparatedData[, 8], start = c(1850), end = c(2022),
frequency = 1)
Seps = ts(data = MonthsSeparatedData[, 9], start = c(1850), end = c(2022),
frequency = 1)
Octs = ts(data = MonthsSeparatedData[, 10], start = c(1850), end = c(2022),
frequency = 1)
Novs = ts(data = MonthsSeparatedData[, 11], start = c(1850), end = c(2022),
frequency = 1)
Decs = ts(data = MonthsSeparatedData[, 12], start = c(1850), end = c(2022),
frequency = 1)
```

Plotting

```
library(ggplot2)
library(gridExtra)


plot1 <- autoplot(Jans)

plot2 <- autoplot(Febs)

plot3 <- autoplot(Mars)

plot4 <- autoplot(Aprs)

plot5 <- autoplot(Mays)

plot6 <- autoplot(Juns)

plot7 <- autoplot(Juls)

plot8 <- autoplot(Augs)

plot9 <- autoplot(Seps)
```

```r
plot10 <- autoplot(Octs)

plot11 <- autoplot(Novs)

plot12 <- autoplot(Decs)



par(mfrow = c(2, 6))
plot(plot1)
```

```r
layout(matrix(c(c(1, 2, 3, 4, 5, 6), c(1, 2, 3, 4, 5, 6)), nrow = 2))
grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10, plot11, plot12)
```

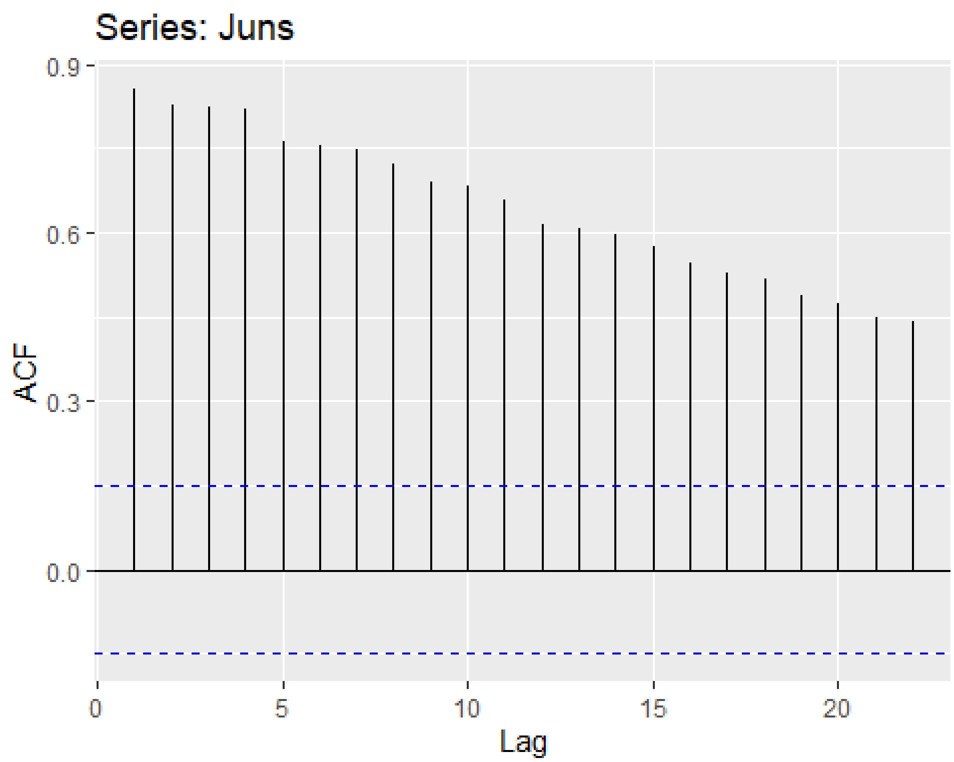


Analysing "June"s

Seasonality checks

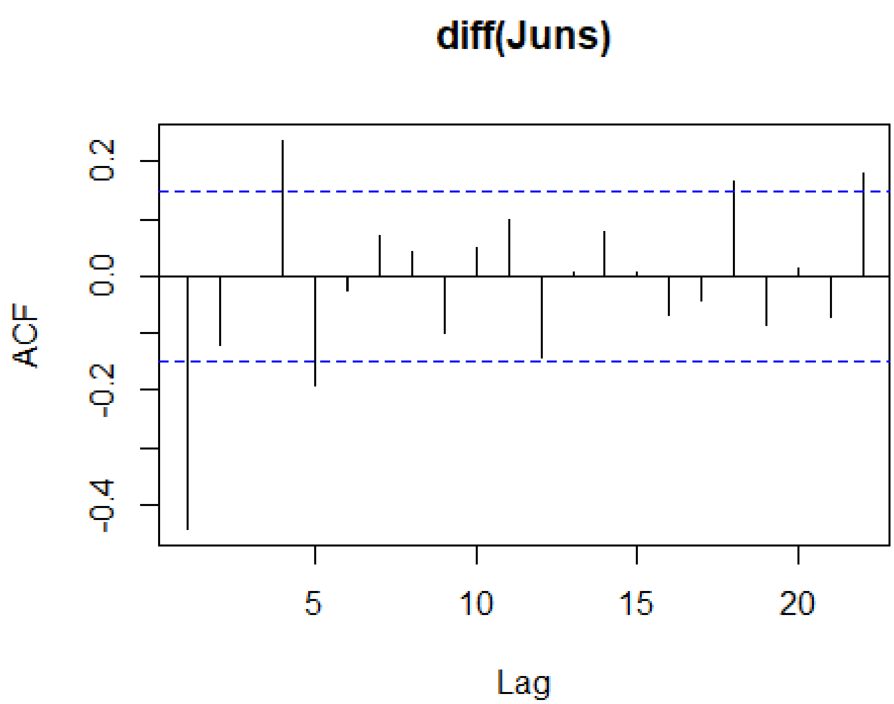
there is no clear given period in the given data, we cannot use functions like ggseasonplot or ggsubseriesplot. I will therefore use Acf to try to find a plausible seasonality

```r
ggAcf(Juns)
```



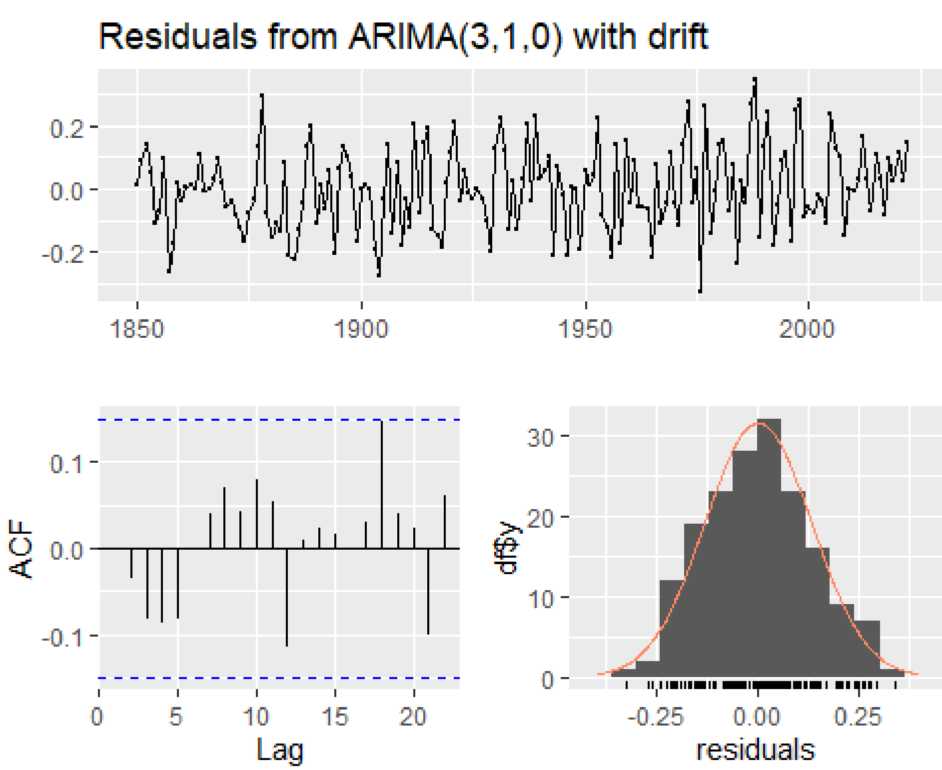
There is no apparent seasonality in this figure.

```
Acf(diff(Juns))
```



the acf of the differenced data seems to suggest cyclic rather than seasonal behavior.

```
JunsArimaFit = auto.arima(Juns, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(JunsArimaFit)

## Series: Juns
## ARIMA(3,1,0) with drift
##
## Coefficients:
##           ar1      ar2      ar3   drift
##       -0.7751  -0.6353  -0.3759  0.0064
## s.e.   0.0709   0.0783   0.0704  0.0036
##
## sigma^2 = 0.01794:  log likelihood = 103.26
## AIC=-196.52   AICc=-196.15   BIC=-180.78
##
## Training set error measures:
##                      ME      RMSE       MAE          MPE      MAPE
MASE
## Training set 0.00102424 0.1319777 0.1068652 -0.001886671 0.6796933
0.7583076
##                      ACF1
## Training set -0.001682539

checkresiduals(JunsArimaFit)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0) with drift
## Q* = 6.6357, df = 7, p-value = 0.4678
##
## Model df: 3.   Total lags used: 10
```

```
JansArimaFit = auto.arima(Jans, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(JansArimaFit)
```

```
## Series: Jans
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8065  0.0073
## s.e.   0.0458  0.0036
##
## sigma^2 = 0.05626:  log likelihood = 3.92
## AIC=-1.84   AICc=-1.69   BIC=7.61
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE
MASE
## Training set 0.001743216 0.2351183 0.1878805 -0.02574302 1.517952
0.7840379
##                     ACF1
## Training set -0.04242369
```

```
FebsArimaFit = auto.arima(Febs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(FebsArimaFit)
```

```
## Series: Febs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8148  0.0067
## s.e.   0.0472  0.0037
##
## sigma^2 = 0.06737:  log likelihood = -11.61
## AIC=29.21   AICc=29.35   BIC=38.65
##
## Training set error measures:
##                        ME      RMSE        MAE         MPE     MAPE
MASE
```

```
## Training set -0.002488931 0.2572962 0.2065518 -0.06508957 1.640744
0.7920133
##                      ACF1
## Training set 0.03585119
```

```r
MarsArimaFit = auto.arima(Mars, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(MarsArimaFit)
```

```
## Series: Mars
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7866  0.0077
## s.e.   0.0467  0.0037
##
## sigma^2 = 0.05091:  log likelihood = 12.55
## AIC=-19.09   AICc=-18.95   BIC=-9.65
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE
MASE
## Training set -0.0005930178 0.223673 0.1711759 -0.03785236 1.296788
0.7750107
##                      ACF1
## Training set 0.04442989
```

```r
AprsArimaFit = auto.arima(Aprs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(AprsArimaFit)
```

```
## Series: Aprs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7625  0.0073
## s.e.   0.0486  0.0033
##
## sigma^2 = 0.03247:  log likelihood = 51.28
## AIC=-96.55   AICc=-96.41   BIC=-87.11
##
## Training set error measures:
##                       ME     RMSE       MAE          MPE     MAPE
MASE
## Training set 0.001178241 0.178624 0.1429521 -0.009931994 1.009385
0.8183043
```

```
##                     ACF1
## Training set 0.0648635
```

```
MaysArimaFit = auto.arima(Mays, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(MaysArimaFit)
```

```
## Series: Mays
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7679  0.0064
## s.e.   0.0418  0.0028
##
## sigma^2 = 0.02511:  log likelihood = 73.39
## AIC=-140.78   AICc=-140.63   BIC=-131.33
##
## Training set error measures:
##                        ME      RMSE       MAE          MPE      MAPE
MASE
## Training set 0.0008522446 0.1570682 0.1238632 -0.007638265 0.8228065
0.7922527
##                     ACF1
## Training set -0.07357558
```

```
JulsArimaFit = auto.arima(Juls, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(JulsArimaFit)
```

```
## Series: Juls
## ARIMA(2,1,3) with drift
##
## Coefficients:
##           ar1      ar2      ma1     ma2      ma3   drift
##       -0.4472  -0.8512  -0.1471  0.3773  -0.5222  0.0064
## s.e.   0.1311   0.1348   0.1903  0.2554   0.0827  0.0030
##
## sigma^2 = 0.01693:  log likelihood = 109.26
## AIC=-204.52   AICc=-203.84   BIC=-182.49
##
## Training set error measures:
##                         ME      RMSE       MAE          MPE      MAPE
MASE
## Training set -0.0001468261 0.1274635 0.1007673 -0.009245452 0.6291836
0.8341799
##                    ACF1
## Training set 0.02563936
```

```r
AugsArimaFit = auto.arima(Augs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(AugsArimaFit)

## Series: Augs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7120  0.0066
## s.e.   0.0524  0.0034
##
## sigma^2 = 0.02286:  log likelihood = 81.54
## AIC=-157.08   AICc=-156.94   BIC=-147.64
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE      MAPE
MASE
## Training set -0.0003328075 0.1498779 0.1158668 -0.01313285 0.7339526
0.8799596
##                   ACF1
## Training set 0.0349142

SepsArimaFit = auto.arima(Seps, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(SepsArimaFit)

## Series: Seps
## ARIMA(1,1,4) with drift
##
## Coefficients:
##           ar1    ma1      ma2      ma3     ma4   drift
##       -0.9050  0.433  -0.7960  -0.2557  0.2323  0.0070
## s.e.   0.0497  0.086   0.0854   0.0744  0.0840  0.0033
##
## sigma^2 = 0.01787:  log likelihood = 104.51
## AIC=-195.02   AICc=-194.34   BIC=-172.99
##
## Training set error measures:
##                         ME     RMSE        MAE          MPE      MAPE
MASE
## Training set -1.558472e-05 0.130935 0.09834443 -0.009458911 0.6471333
0.8246637
##                    ACF1
## Training set -0.0197987

OctsArimaFit = auto.arima(Octs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)
```

```r
summary(OctsArimaFit)

## Series: Octs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7157  0.0070
## s.e.   0.0489  0.0035
##
## sigma^2 = 0.02541:  log likelihood = 72.44
## AIC=-138.87   AICc=-138.73   BIC=-129.43
##
## Training set error measures:
##                        ME      RMSE      MAE          MPE      MAPE
MASE
## Training set 0.0007767577 0.1580188 0.122532 -0.009431547 0.8529203
0.8115004
##                    ACF1
## Training set 0.01390877

NovsArimaFit = auto.arima(Novs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(NovsArimaFit)

## Series: Novs
## ARIMA(0,1,2) with drift
##
## Coefficients:
##           ma1      ma2   drift
##       -0.6425  -0.1192  0.0064
## s.e.   0.0795   0.0774  0.0034
##
## sigma^2 = 0.03495:  log likelihood = 45.5
## AIC=-83.01   AICc=-82.77   BIC=-70.42
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE
MASE
## Training set -0.0005283856 0.1847644 0.1506092 -0.02579027 1.130994
0.8461334
##                     ACF1
## Training set -0.01017421

DecsArimaFit = auto.arima(Decs, stepwise = FALSE, approximation = FALSE,
allowdrift = TRUE)

summary(DecsArimaFit)
```

```
## Series: Decs
## ARIMA(3,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2   drift
##       1.1237  -0.0593  -0.2112  -1.8909  0.9313  0.0071
## s.e.  0.0933   0.1130   0.0815   0.0634  0.0560  0.0044
##
## sigma^2 = 0.04525:  log likelihood = 24.12
## AIC=-34.23   AICc=-33.55   BIC=-12.2
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE
MASE
## Training set 2.242205e-05 0.2083781 0.1623593 -0.03035991 1.285315
0.7558742
##                      ACF1
## Training set -0.004656246
```

Most models had Arima 0 1 1 non seasonal order with close drifts