UROP Project
Spring 2024
Regression and Time Series Modeling and Forecasting

Final Report

Work done by:
Khaled Hasan

Birzeit University

Supervised by:
Dr. Hani Kabajah(Birzeit University)
and
Dr. Peter Kempthorne (MIT)

June 30, 2024

Final Report

Khaled Hasan

2024-06-22

Contents

1 Introduction

In this project, I started with the aim of getting introduced to time series analysis, and their possible applications in physics, I have therefore started with exploring various concepts such as seasonality, stationarity, correlation and the many associated tests (KPSS, Augmented Dicky-Fuller and Ljung-Box tests). Different types of models were also invistegated such as regression, ARIMA, sARIMA and a mixing of the two (regression with arima errors). The second half of the project was putting these models in physics-related applications, namely global warming and earths temperature dynamics. Which was an interesting case to studey, due to the seasonality of the temperatures data, and the various ways of clustering the temperatures based on the spatial dimensions. Different regions provided different results. in terms of seasonality, trending and fitting coefficients.

2 Data

The data in consideration was a cross sectional temperature data anomaly (with the climatology provided) with different resolutions from Berkeley Earth website (https://berkeleyearth.org/high-resolution-data-access-page/).

2.1 Stats

The frequency is monthly, (12 points per year), starting from Jan-1850 lasting until Jan-2022.In terms of spatial part, the set I used had a small resulotion ($5^o \times 5^o$), since a higher resulotion sets like ($1^o \times 1^o$) ($0.25^o \times 0.25^o$) used too much data 1GB and 6GB respictively, making it almost impossible to takle with my humble 8GB RAM device. additionally, the higher resolution data were subject to more data pre-processing, which is generelly unfavourable in our models.

2.2 Code

```
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)
```

2.3 Averaging Scheme

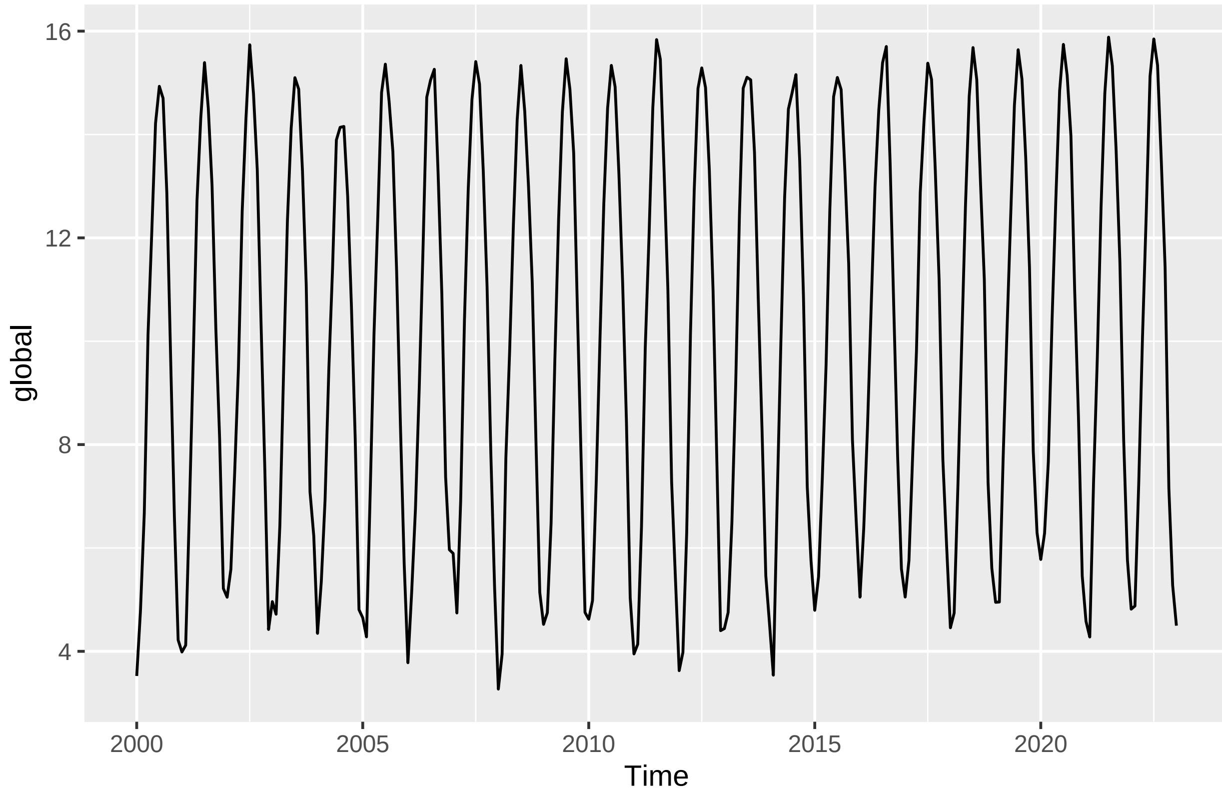
In order to represent a meaningful time series, the data was averaged whether Globally or on different regions. Eitherways, the averaging was corrected (weighted) by the area of the different grid elements and done in Python. It should also be noted that the temperatures in the south pole had many NANs, these were handled automatically by Pyhtons Numpy library. The averaging also excluded the water masses, these are expected to have a smaller temperature variations due to waters high specific heat capacity. The exclusion of water massses was done using a provided land mask from the dataset.

3 Seasonality

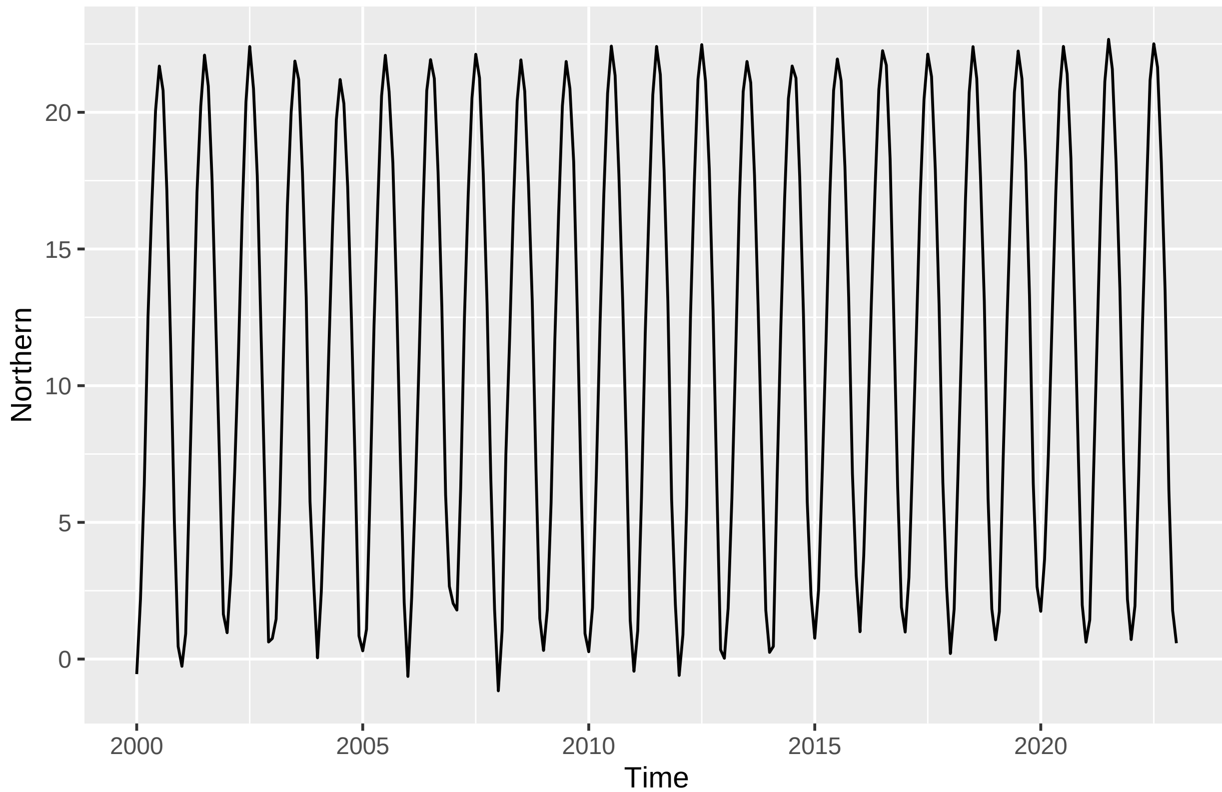
There is a clear seasonality in the data with a 12 months period, however, the phase between the lower and upper hemisphere differs (Northern summers are Southern winters and vice versa) as shown in the figure below. Notice that the global temperature is dominated by the northern hemisphere, this is most likely because the north has more landmass compared to the south.

3.1 Plots

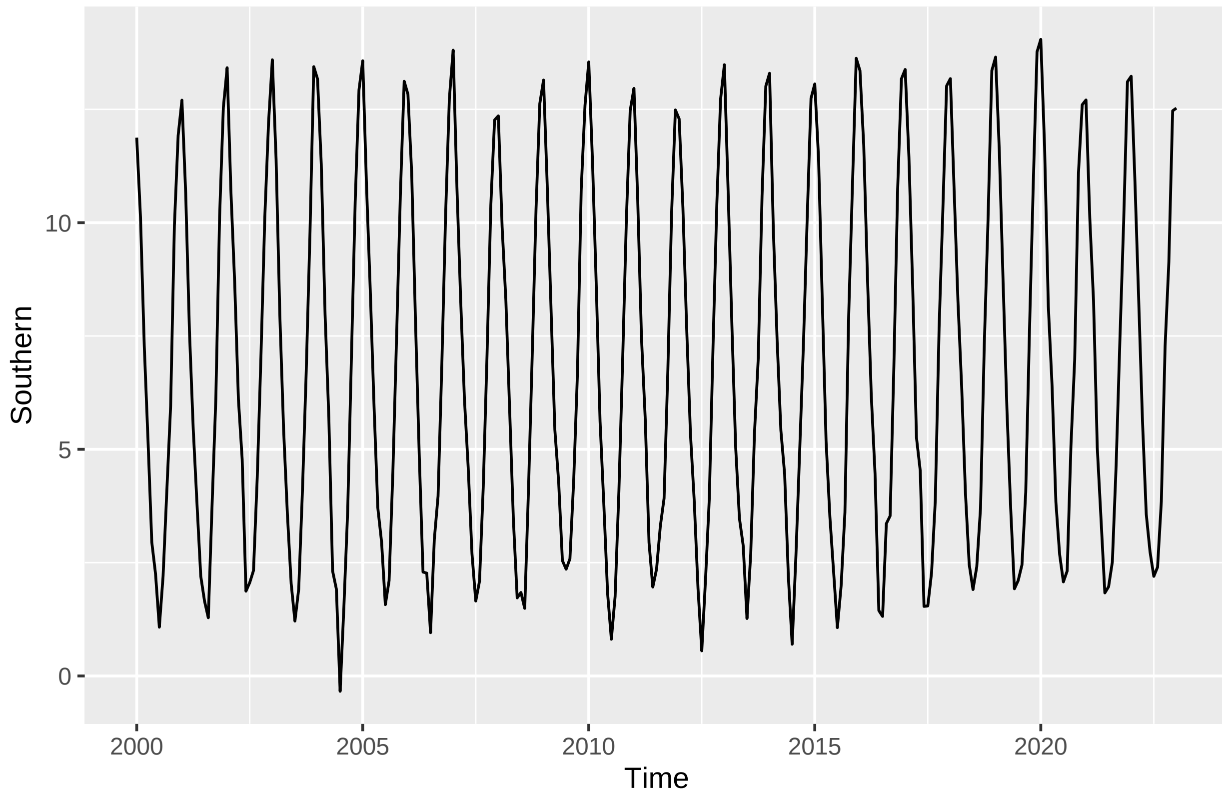
```
library(ggplot2)
library(gridExtra)

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

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

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

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



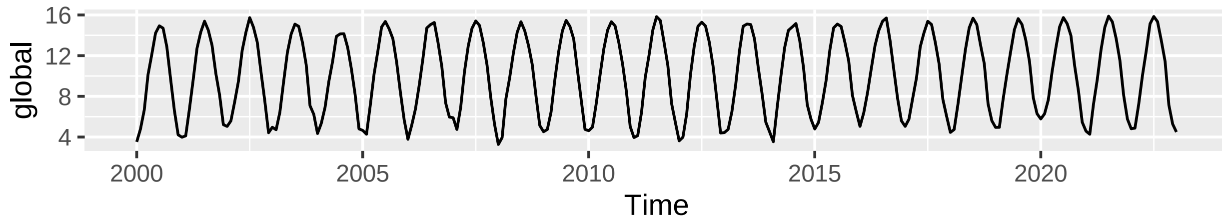

```
plot(autoplot2)
```

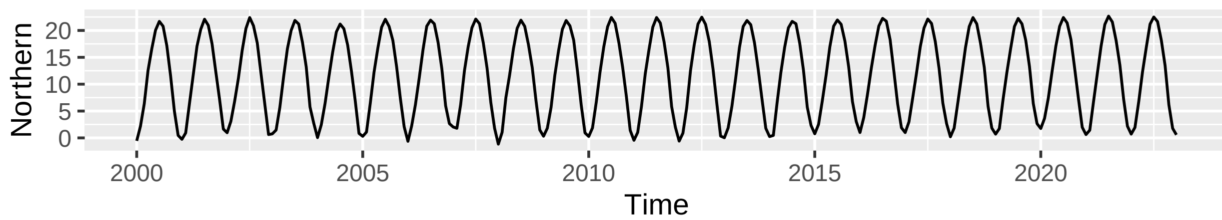


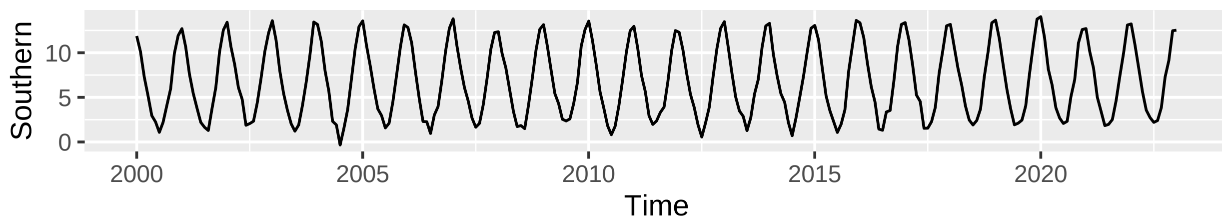

```
plot(autoplot3)
```



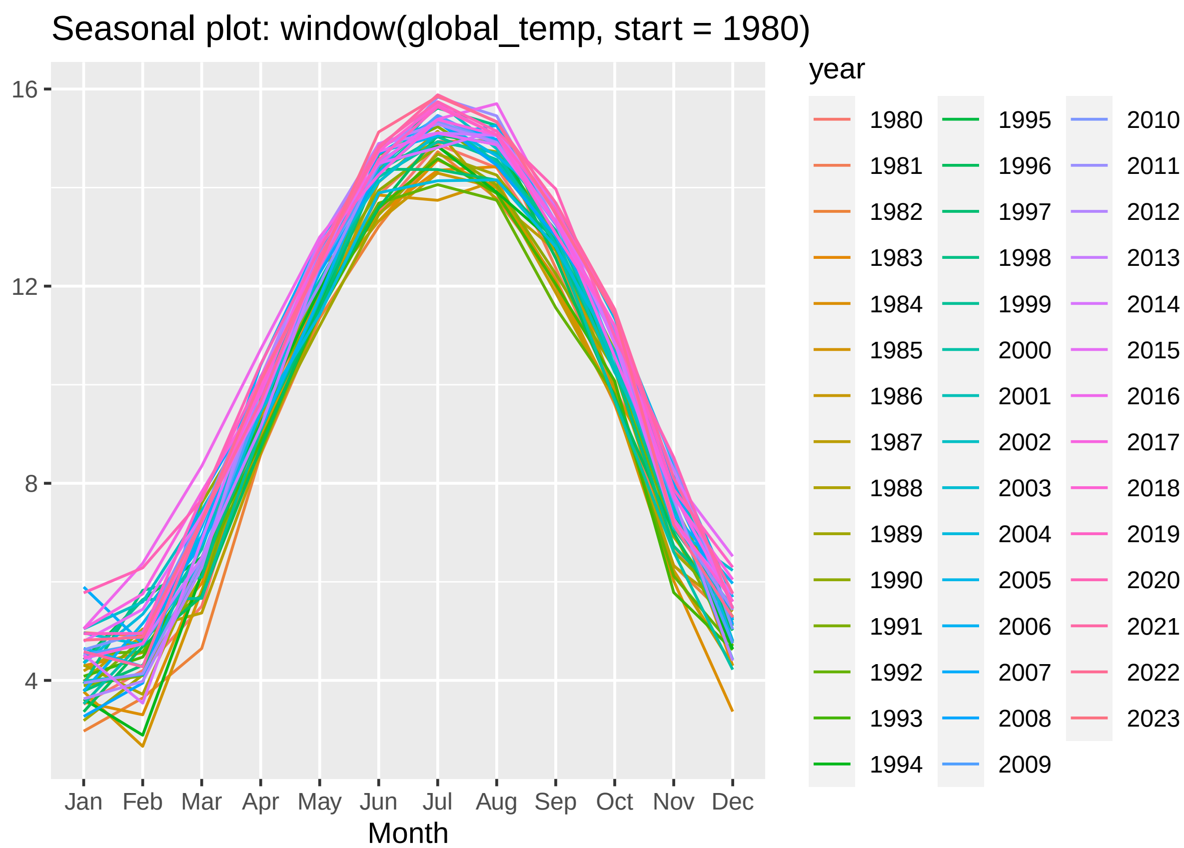

```
layout(matrix(c(1, 2, 3), nrow = 1))
grid.arrange(autoplot1, autoplot2, autoplot3)
```








```r
ggseasonplot(window(global_temp, start=1980))
```



3.2 Removing seasonality

The seasonality can be due to many factors, It can include a trending and cyclic behaviour, therefore, it is not always easy to separate the seasonality from a model, however, in this case, I could separate all seasonality from the global temperature data by simply includeing a sinusoidal element with a similar period in the regression part (i.e. $\sin(\frac{2\pi}{12})$ and $\cos(\frac{2\pi}{12})$).

3.2.1 With regression:

```
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
```

However, this was only possible with the global temperature data, when narrower regions are considered, it can no longer be eliminated with ease.

```
northern_fitting_Arimareg = Arima_fittng(northernhemisphere_temp, startingPoint = c(1987, 1))

summary(northern_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##         ar1    sar1    sar2  intercept                          t
##       0.290  0.2259  0.3069    10.8058  -5.7363  -9.2814  0.0029
## s.e.  0.046  0.0456  0.0461     0.1468   0.1036   0.1034  0.0006
##
## sigma^2 = 0.3291:  log likelihood = -372.17
## AIC=760.35   AICc=760.69   BIC=792.91
##
## Training set error measures:
##                         ME      RMSE      MAE       MPE     MAPE      MASE
## Training set -0.0008339219 0.5690337 0.434276 -1.526649 28.03759 0.7606312
##                     ACF1
## Training set -0.01118138
```

```
southern_fitting_Arimareg = Arima_fittng(southernhemisphere_temp, startingPoint = c(1987, 1))

summary(southern_fitting_Arimareg)
```

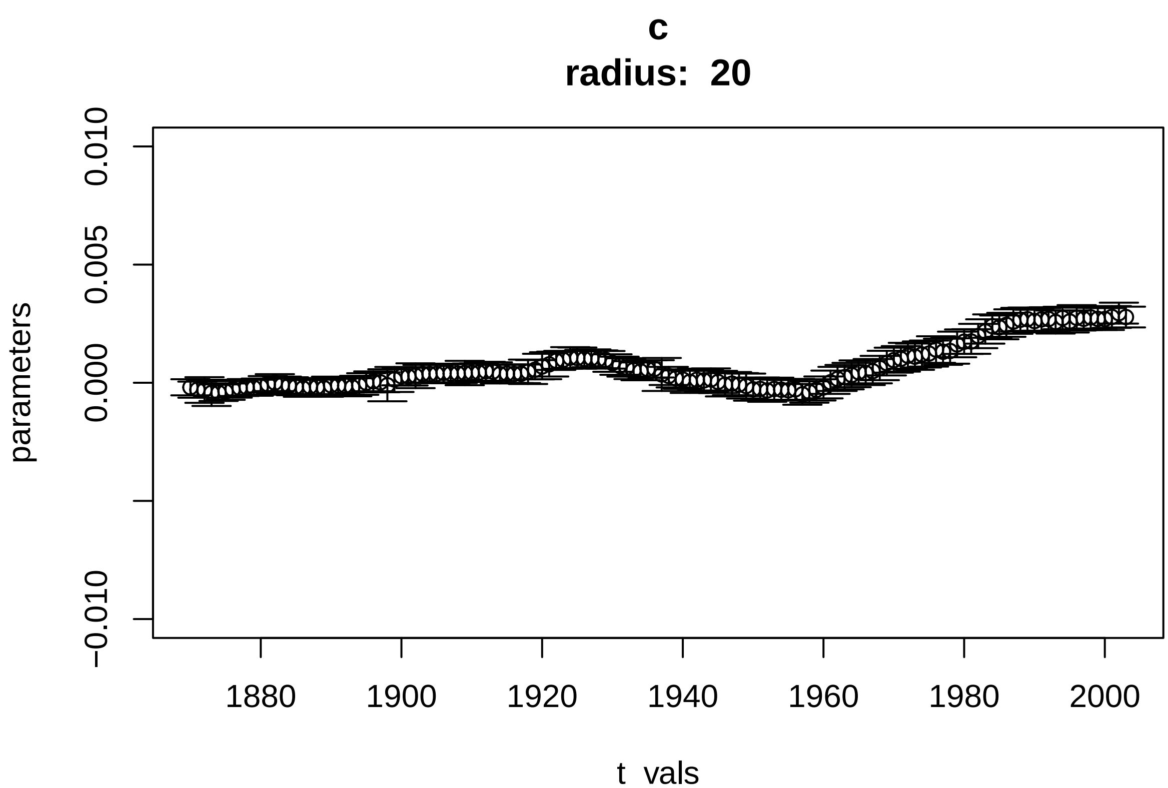
```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                        t
##       0.3019  0.3397  0.3300      6.357  2.0949  5.2165  0.0014
## s.e.  0.0460  0.0458  0.0468      0.199  0.1453  0.1451  0.0008
##
## sigma^2 = 0.3359:  log likelihood = -377.92
## AIC=771.84   AICc=772.18   BIC=804.41
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.002690256 0.5748943 0.4512634 -2.562593 15.35262 0.7727937
##                      ACF1
## Training set -0.004790468
```

4 Global Warming

One of the interesting and important topics to study in environmental sciences that is related to temperature is the global warming, as it has a huge impact on our everyday lives and on the near future of the planet. The warming phenomena is an example of trending behavior in the global temperatures, this can be measured using either drift or a linear term in the regressors. I will be using the factor of such linear coefficient term to represent warming, I will also consider a window to examine the global warming in different times throughout the last 170 years. Therefore I made two functions (linear_coef) which should determine the global warming coefficient based on a given model type (regression with arima errors of a given orders). And plot_Global_warming which uses the same model, or produce a new model for different windows of given radius (the latter option takes too much time) and plots the global warming factor for windows centered at the time coordinate.

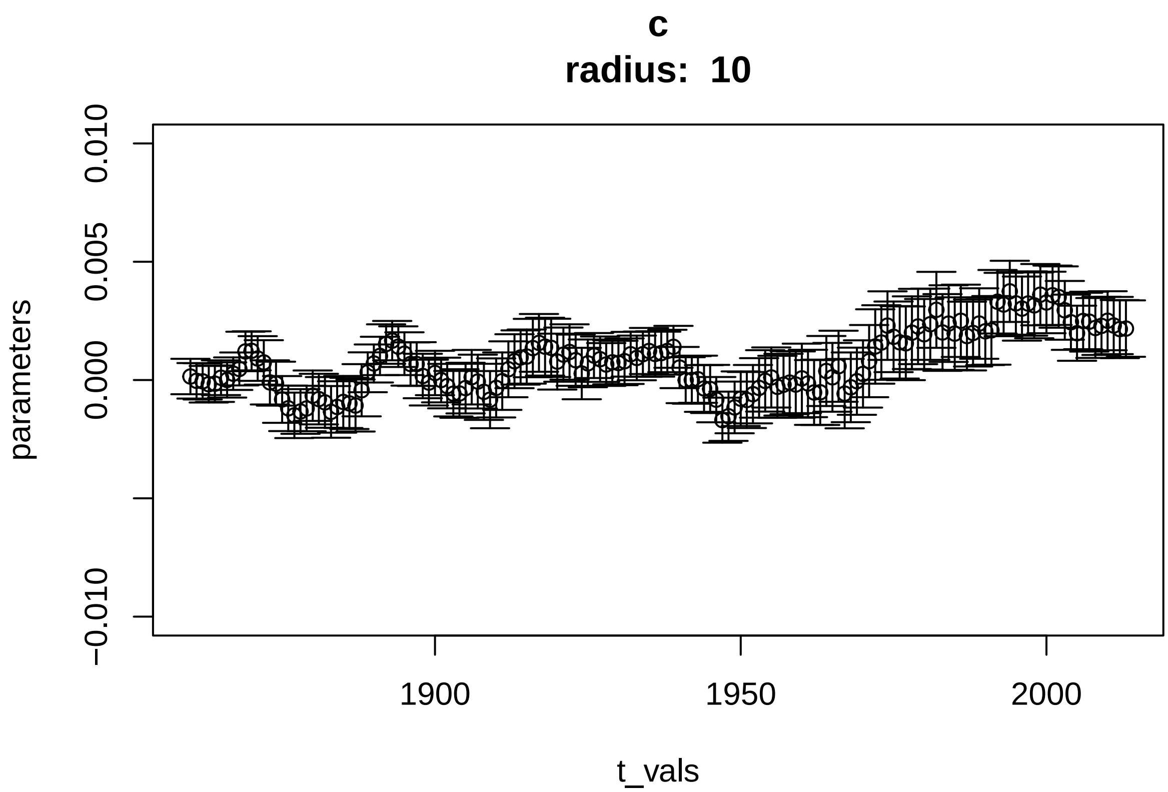
```
linear_coef <- function(DATA, x, Ord, sOrd, radius = 2){

  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_Global_warming <- function(timeseries, arima_fit = NULL, r = 20){

  if(is.null(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 = r

  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, l

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

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

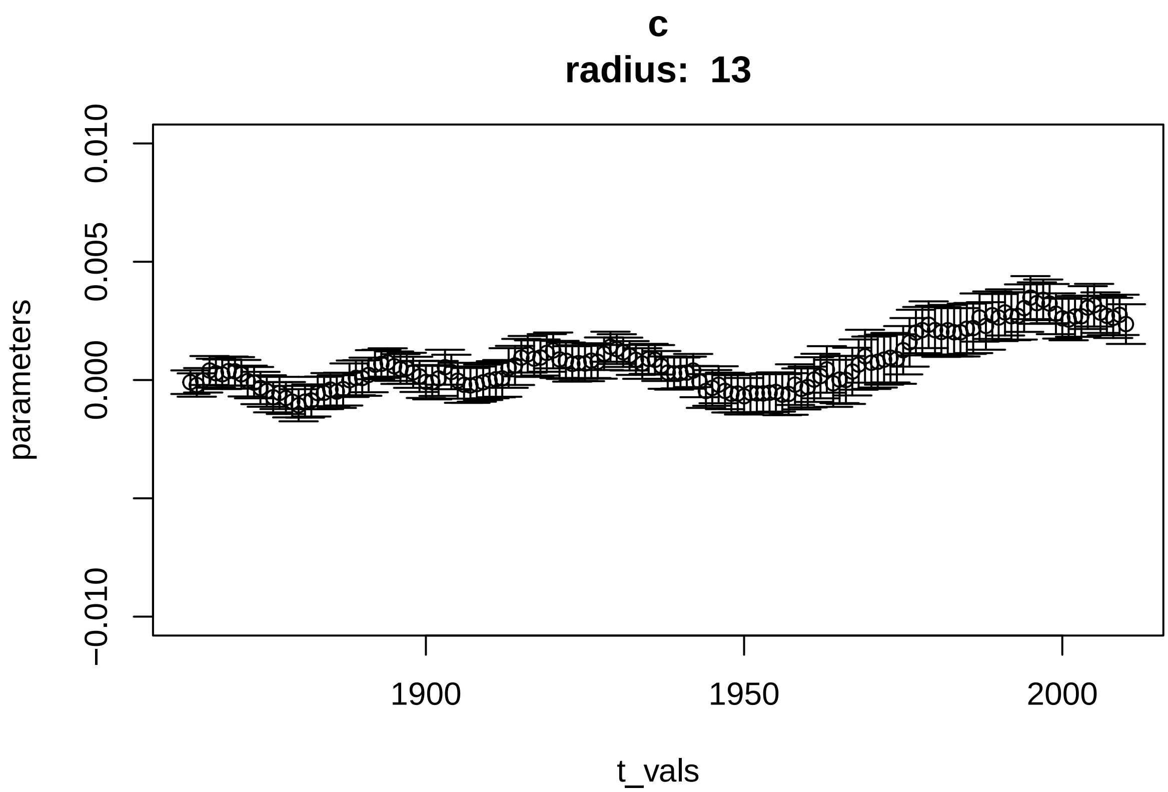


Tuning the radius of the window is essential to insure stability of the model parameters, for small windows the parameters

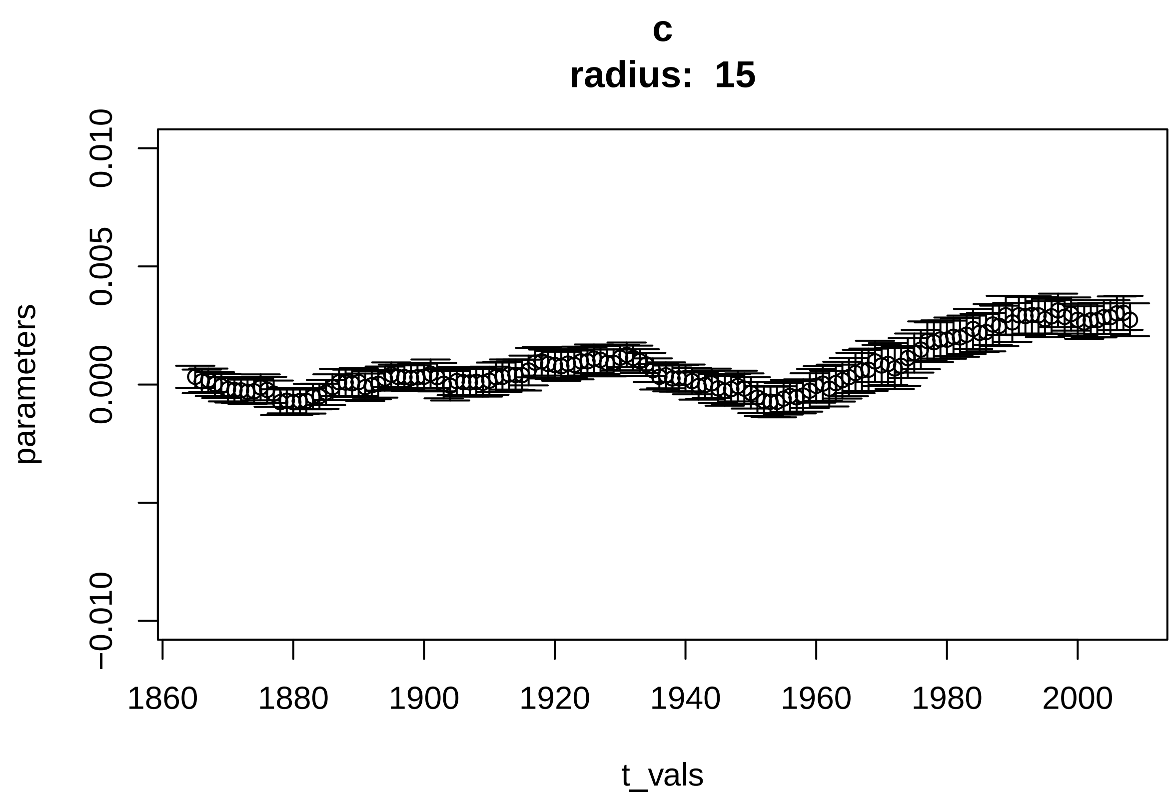
```
t1 = plot_Global_warming(global_temp, r = 10)
```




```
t2 = plot_Global_warming(global_temp, r = 13)
```




```
t3 = plot_Global_warming(global_temp, r = 15)
```

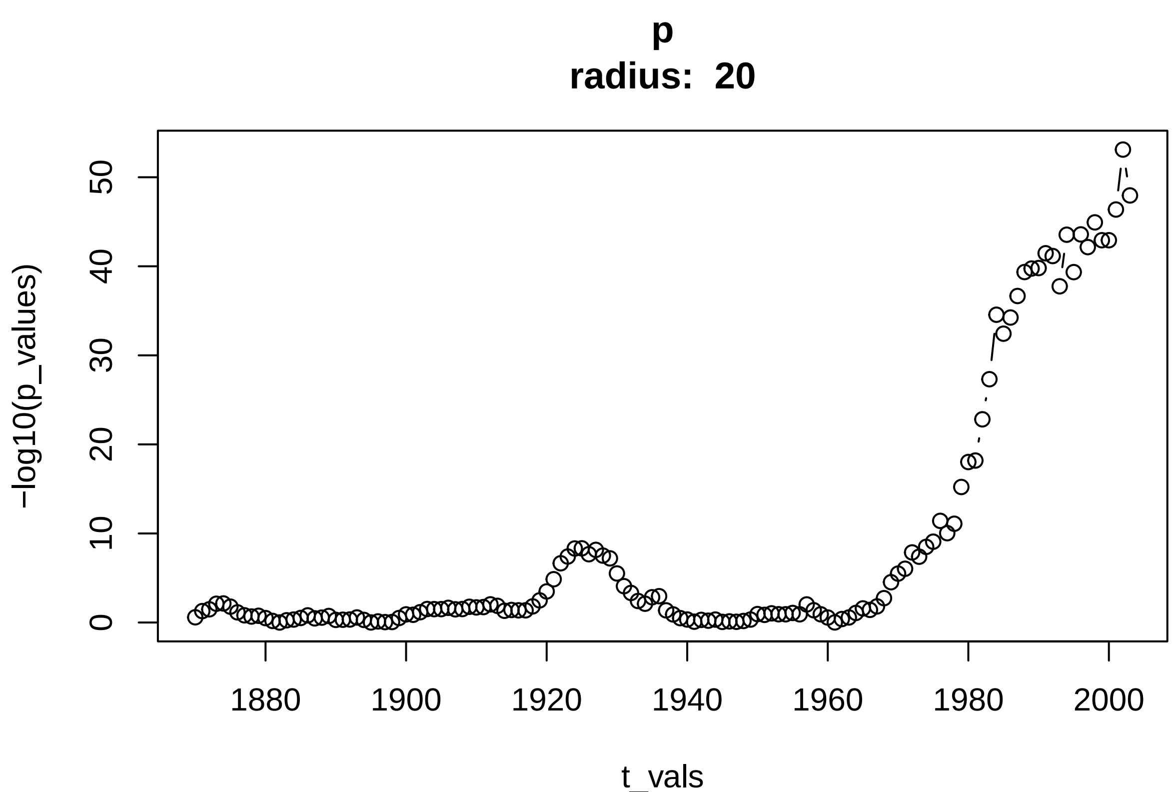


Notice that the global warming coefficient confidence interval might include 0, which implies that there is no significant warming. warming is more significant when the p values are small, therefore I plotted $-\log(p)$

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

  #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()

  p_values = c()

  rad = r

  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("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
```



By looking at the above plot, we find that the global warming increase started to appear in windows centered at 1960, those include year from 1940 onward.

5 Splitting lattiudinal strips

by splitting the earth into 6 geographic zones, based on their latitudinal coordinates, the splitting was made into the following regions: - *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.

```r
LattitudinalTemps = read.csv(file ="C:\\Users/ss/Desktop/Time_series_Analysis/LatittudCuttedTempereture
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)
```

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

the splitting had to insure no mixing between northern and southern regions, since these two regions behave differently. The analysis performed on the different regions has shown that there is stronger warming coeffecient, but this most likely due to the smaller southern land mass. Another important result is the instability

```r
summary(northPole_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(1,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1  intercept                         t
##       0.4060  0.7724    25.9194  -0.8574  -2.9040  0.0018
## s.e.  0.0402  0.0274     0.2634   0.1807   0.1804  0.0008
##
## sigma^2 = 0.2362:  log likelihood = -363.1
## AIC=740.19   AICc=740.41   BIC=769.93
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.003698974 0.4831931 0.3730615 -0.05472521 1.460727 0.8738155
##                     ACF1
## Training set -0.01099241
```

```r
summary(north_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                          t
##       0.2928  0.1855  0.2528     7.0432  -7.2412  -11.4979  3e-03
## s.e.  0.0422  0.0425  0.0432     0.1582   0.1083    0.1082  5e-04
##
## sigma^2 = 0.5928:  log likelihood = -596.1
## AIC=1208.21   AICc=1208.49   BIC=1242.19
##
## Training set error measures:
##                        ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.002197748 0.7646846 0.5786993 11.22289 30.63854 0.7690733
```

```
##                       ACF1
## Training set -0.01235676
```

```r
summary(trop_north_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,1)(2,0,0)[12] errors
##
## Coefficients:
##          ar1      ma1    sar1   sar2  intercept                           t
##       0.6672  -0.5062  0.2903  0.392   -11.6720  -10.4936  -14.0311  0.0042
## s.e.  0.1507   0.1743  0.0408  0.043     0.4336    0.2617    0.2614  0.0014
##
## sigma^2 = 1.599:  log likelihood = -854.52
## AIC=1727.05   AICc=1727.4   BIC=1765.28
##
## Training set error measures:
##                        ME     RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01420732 1.254799 0.9225076 -0.5332841 22.52774 0.7617368
##                      ACF1
## Training set 0.00314081
```

```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
##       0.5771  0.4348  0.3794    23.9467  0.6998  2.0017  0.0015
## s.e.  0.0360  0.0403  0.0411     0.3081  0.1909  0.1906  0.0010
##
## sigma^2 = 0.1428:  log likelihood = -233.03
## AIC=482.05   AICc=482.33   BIC=516.03
##
## Training set error measures:
##                          ME      RMSE      MAE         MPE     MAPE      MASE
## Training set 0.0002757467 0.3752943 0.302247 -0.02259995 1.241574 0.7132813
##                       ACF1
## Training set -0.07714231
```

```r
summary(south_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(1,0,0)(2,0,0)[12] errors
##
## Coefficients:
##          ar1    sar1    sar2  intercept                      t
##       0.2979  0.2822  0.2502    16.9605  3.1644  5.2073  0.0014
## s.e.  0.0422  0.0427  0.0434     0.1252  0.0864  0.0863  0.0004
##
## sigma^2 = 0.2638:  log likelihood = -387.37
## AIC=790.74   AICc=791.02   BIC=824.72
##
## Training set error measures:
##                          ME      RMSE      MAE        MPE     MAPE      MASE
```

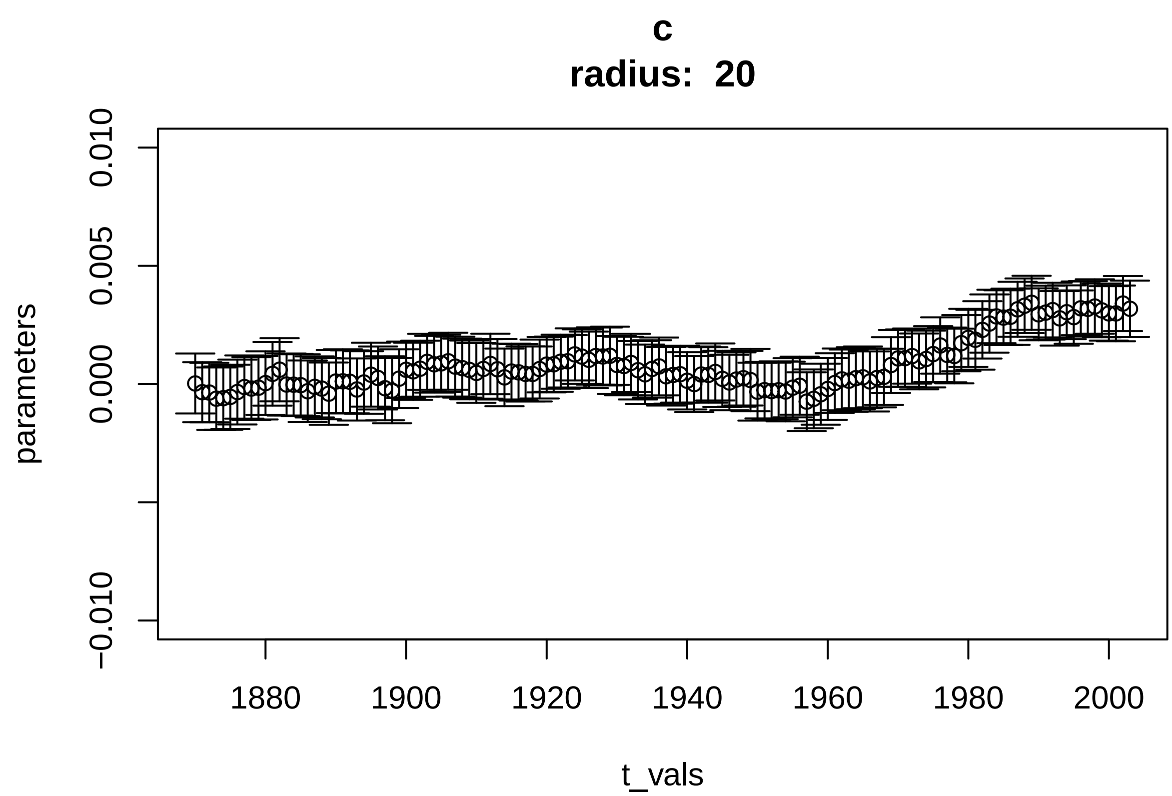
```
## Training set -0.004360253 0.5101558 0.404103 -0.1422314 2.506581 0.7687455
##                      ACF1
## Training set -0.003904747
```

```
summary(southPole_fitting_Arimareg)
```

```
## Series: cutted_data
## Regression with ARIMA(3,0,0)(1,0,0)[12] errors
##
## Coefficients:
##          ar1      ar2      ar3    sar1  intercept                          t
##       0.3714  -0.3368  -0.4903  0.1524   -32.8767  3.9483  10.9408  0.0013
## s.e.  0.0385   0.0402   0.0394  0.0526     0.1170  0.1161   0.1163  0.0004
##
## sigma^2 = 2.743:  log likelihood = -991.84
## AIC=2001.68   AICc=2002.04   BIC=2039.92
##
## Training set error measures:
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.004086165 1.643431 1.296531 -0.3846442 4.133181 0.8963679
##                     ACF1
## Training set -0.01221421
```

These models are less stable, and the have in general a bigger standard error.

```
warming_north = plot_Global_warming(north)
```



The south pole models are highly unstable, due to the missing data, especially in early years

```
tryCatch({warming_southPole = plot_Global_warming(southPole)}, error = function(e) {
  cat("An error occurred: ", e$message, "\n")
  NA
})
```

```
## An error occurred:  non-stationary seasonal AR part from CSS
```

```
## [1] NA
```

#Months Clustering:

For this analysis I only used the northern hemisphere

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

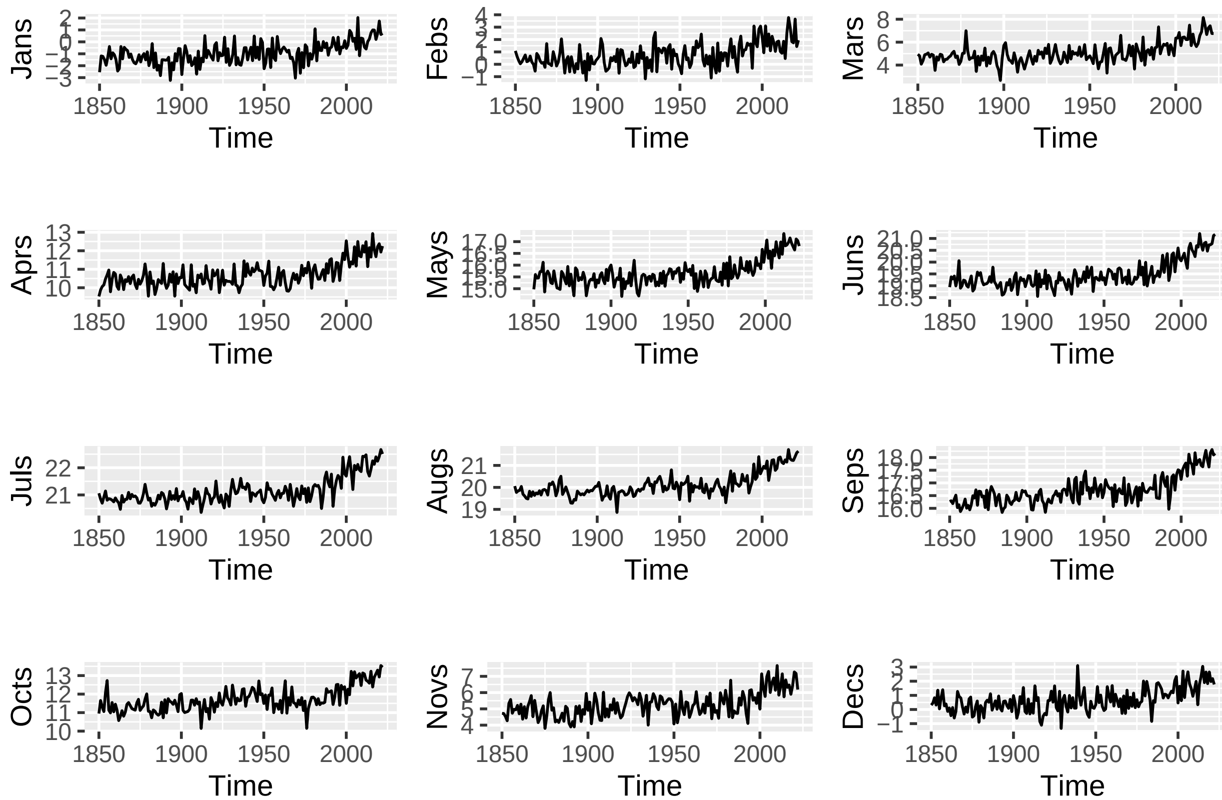
5.1 Plots

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

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

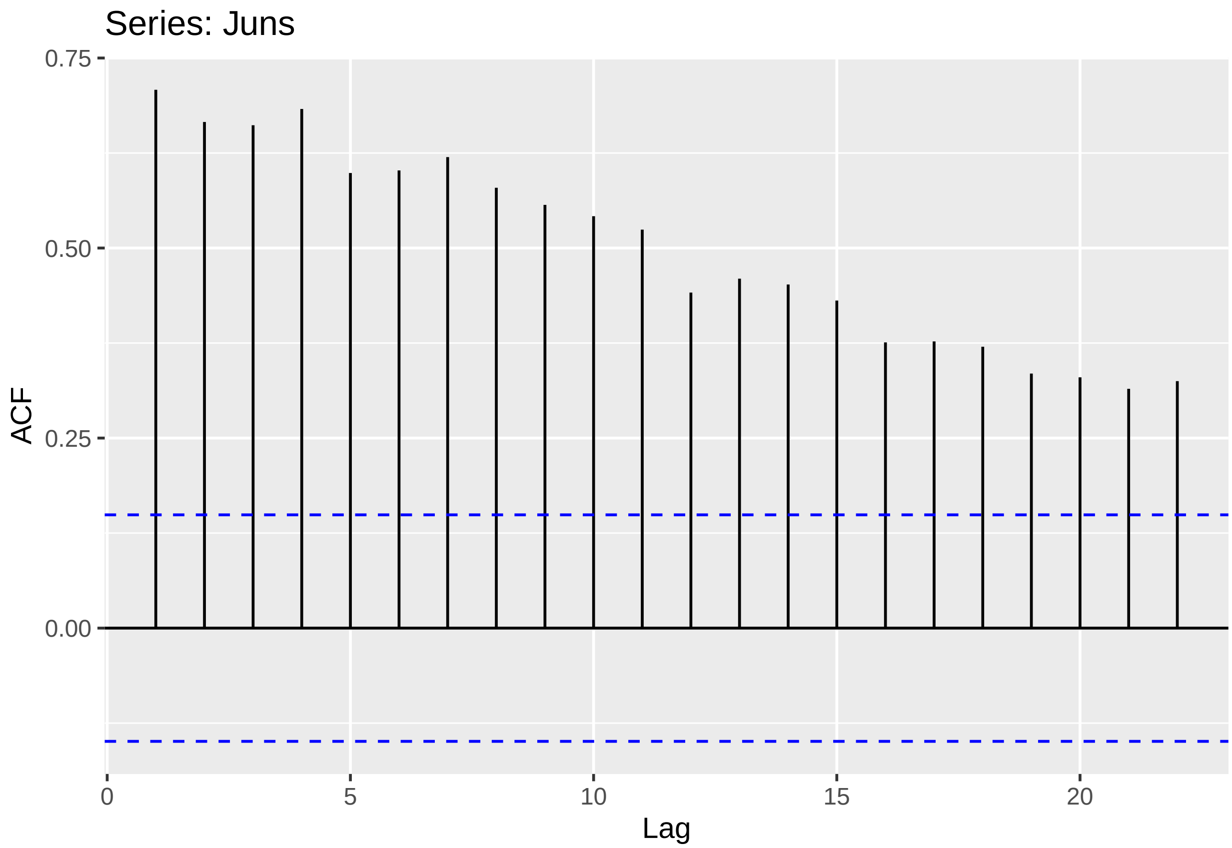
```
plot1 <- autoplot(Jans)

plot2 <- autoplot(Febs)

plot3 <- autoplot(Mars)

plot4 <- autoplot(Aprs)

plot5 <- autoplot(Mays)

plot6 <- autoplot(Juns)

plot7 <-autoplot(Juls)

plot8 <- autoplot(Augs)

plot9 <- autoplot(Seps)

plot10 <- autoplot(Octs)

plot11 <- autoplot(Novs)
```

```r
plot12 <- autoplot(Decs)



par(mfrow = c(2, 6))

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)
```



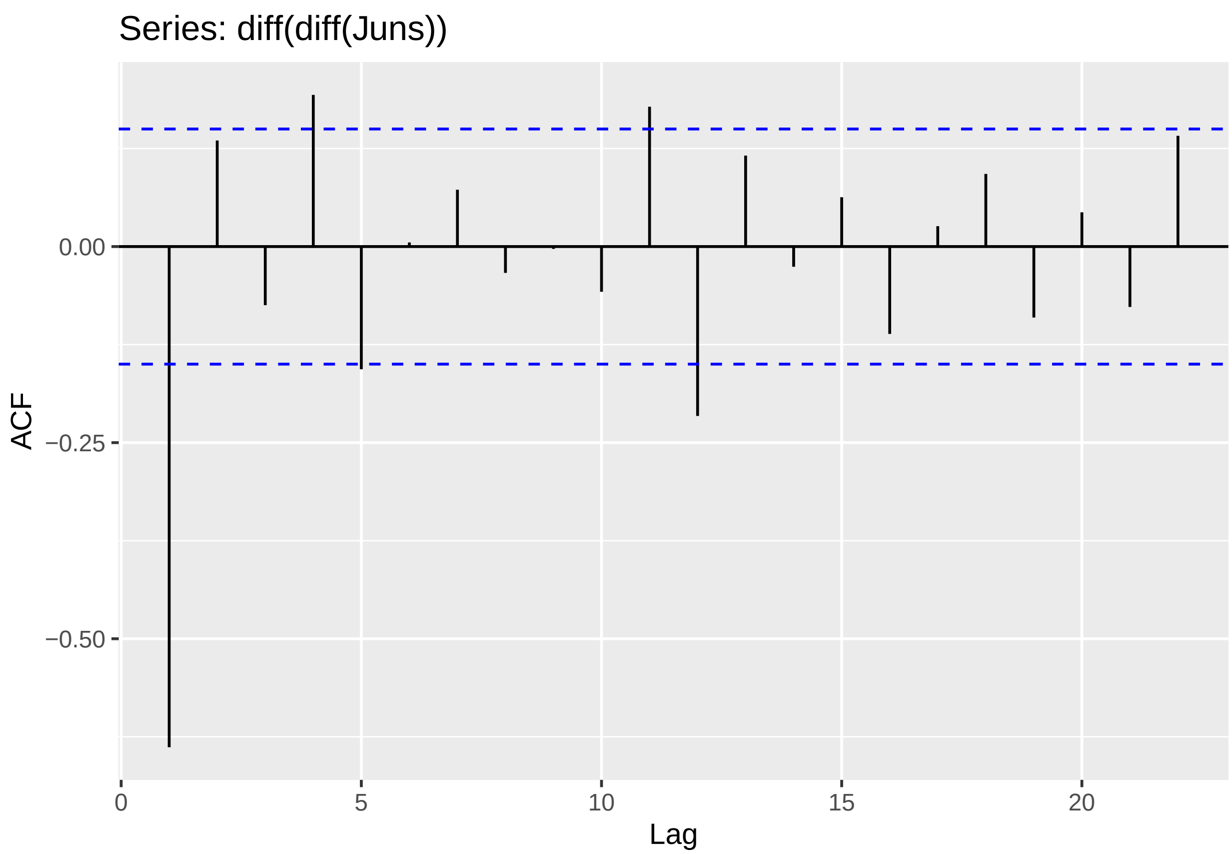
5.2 Seasonality

```r
ggAcf(Juns)
```



There is no apparent seasonality in this figure.

```
ggAcf(diff(diff(Juns)))
```



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

5.3 Arima Models

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

summary(JansArimaFit)
```

```
## Series: Jans
## ARIMA(1,1,1) with drift
##
## Coefficients:
##           ar1      ma1   drift
##       -0.1493  -0.8583  0.0128
## s.e.   0.0871   0.0510  0.0074
##
## sigma^2 = 0.5859:  log likelihood = -197.37
## AIC=402.75   AICc=402.99   BIC=415.34
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.0110921 0.7565671 0.5858128 -36.01011 159.0192 0.6905222
##                      ACF1
## Training set 0.007772005
```

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

summary(FebsArimaFit)
```

```
## Series: Febs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.9114  0.0103
## s.e.   0.0340  0.0060
##
## sigma^2 = 0.6975:  log likelihood = -212.95
## AIC=431.91   AICc=432.05   BIC=441.35
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.01571603 0.8278734 0.6613522 -14.84503 267.8702 0.7310104
##                     ACF1
## Training set -0.01836561
```

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

summary(MarsArimaFit)
```

```
## Series: Mars
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8703  0.0132
## s.e.   0.0335  0.0070
##
## sigma^2 = 0.4677:  log likelihood = -178.4
## AIC=362.81   AICc=362.95   BIC=372.25
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE    MAPE      MASE
## Training set -0.004834004 0.6779242 0.5199041 -2.143945 10.7592 0.7522966
##                    ACF1
## Training set 0.02629492
```

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

summary(AprsArimaFit)
```

```
## Series: Aprs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8476  0.0117
## s.e.   0.0384  0.0057
##
## sigma^2 = 0.2268:  log likelihood = -116.1
```

```
## AIC=238.21   AICc=238.35   BIC=247.65
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.008983057 0.4721324 0.3804977 -0.1125769 3.557137 0.7598155
##                     ACF1
## Training set 0.005058108
```

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

summary(MaysArimaFit)
```

```
## Series: Mays
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8269  0.0088
## s.e.   0.0361  0.0047
##
## sigma^2 = 0.1217:  log likelihood = -62.47
## AIC=130.94   AICc=131.09   BIC=140.39
##
## Training set error measures:
##                     ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.00413058 0.3457741 0.2742007 -0.0269963 1.753041 0.7181208
##                     ACF1
## Training set -0.08545753
```

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

summary(JunsArimaFit)
```

```
## Series: Juns
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8155  0.0099
## s.e.   0.0376  0.0046
##
## sigma^2 = 0.1032:  log likelihood = -48.32
## AIC=102.64   AICc=102.78   BIC=112.08
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set 0.001558295 0.3185169 0.2572943 -0.02350401 1.320322 0.7826297
##                     ACF1
## Training set -0.02033545
```

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

summary(JulsArimaFit)
```

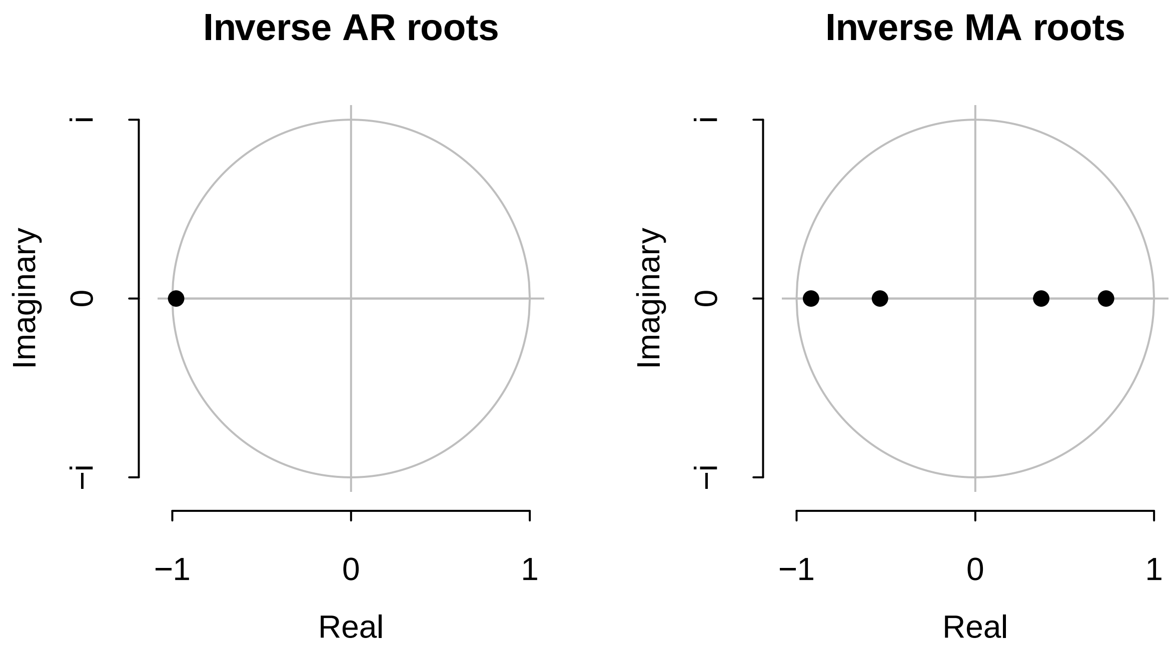
```
## Series: Juls
## ARIMA(2,1,3) with drift
```

```
##
## Coefficients:
##           ar1      ar2      ma1     ma2      ma3   drift
##       -0.5647  -0.7565  -0.0465  0.1807  -0.6642  0.0087
## s.e.   0.1970   0.1001   0.2045  0.1405   0.0860  0.0042
##
## sigma^2 = 0.07248:  log likelihood = -15.97
## AIC=45.94   AICc=46.62   BIC=67.97
##
## Training set error measures:
##                        ME      RMSE      MAE         MPE      MAPE      MASE
## Training set -0.00239044 0.2637244 0.202831 -0.02939569 0.9569008 0.7733554
##                      ACF1
## Training set -0.005294799
```

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

summary(AugsArimaFit)
```

```
## Series: Augs
## ARIMA(1,1,4) with drift
##
## Coefficients:
##           ar1     ma1      ma2      ma3     ma4   drift
##       -0.9786  0.3522  -0.8382  -0.1476  0.1322  0.0095
## s.e.   0.0206  0.0801   0.0842   0.0773  0.0780  0.0056
##
## sigma^2 = 0.08564:  log likelihood = -30.27
## AIC=74.55   AICc=75.23   BIC=96.58
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.002424953 0.2866674 0.2192102 -0.03478599 1.090164 0.8227108
##                      ACF1
## Training set 0.0006727474
```

```r
plot(AugsArimaFit)
```




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

summary(SepsArimaFit)
```

```
## Series: Seps
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.7947  0.0110
## s.e.   0.0426  0.0052
##
## sigma^2 = 0.1059:  log likelihood = -50.5
## AIC=106.99   AICc=107.14   BIC=116.44
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.001484166 0.3226627 0.2570461 -0.05032407 1.538747 0.8237334
##                    ACF1
## Training set 0.04863152
```

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

summary(OctsArimaFit)
```

```
## Series: Octs
## ARIMA(0,1,1)
```

```
##
## Coefficients:
##           ma1
##       -0.7759
## s.e.   0.0465
##
## sigma^2 = 0.2024:  log likelihood = -106.62
## AIC=217.25   AICc=217.32   BIC=223.54
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.05010242 0.4472642 0.3494763 0.2816909 2.981887 0.7910138
##                     ACF1
## Training set -0.01846638
```

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

summary(NovsArimaFit)
```

```
## Series: Novs
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8608  0.0105
## s.e.   0.0373  0.0067
##
## sigma^2 = 0.3725:  log likelihood = -158.81
## AIC=323.62   AICc=323.76   BIC=333.06
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.001950608 0.6050494 0.4847957 -1.319531 9.423161 0.7428419
##                     ACF1
## Training set -0.01841711
```

```r
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
##       0.9183  0.0542  -0.2259  -1.8372  0.8716  0.0093
## s.e.  0.1021  0.1015   0.0811   0.0746  0.0732  0.0071
##
## sigma^2 = 0.4744:  log likelihood = -178.08
## AIC=370.16   AICc=370.84   BIC=392.19
##
## Training set error measures:
##                       ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.005130545 0.6746876 0.5367095 3.581654 150.2769 0.7018703
##                     ACF1
```

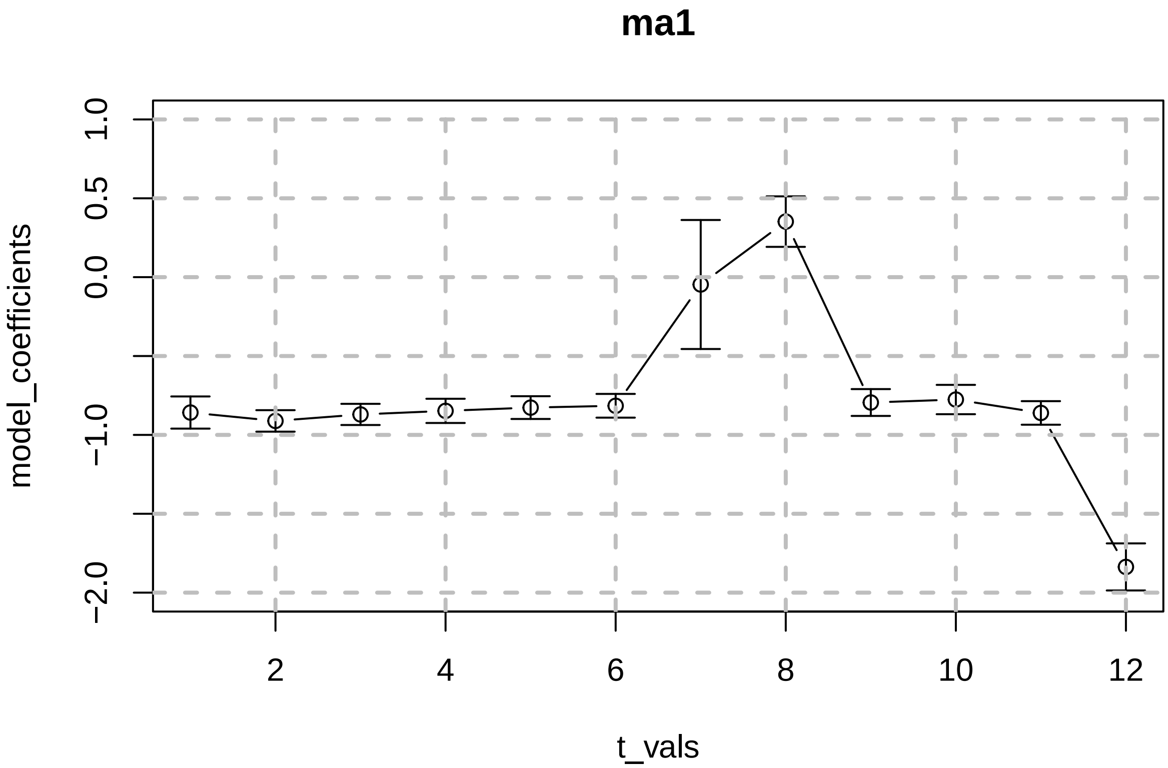
```
## Training set -0.01895019
```

Plotting ma coefficients

```r
months_models = cbind(c(JansArimaFit), c(FebsArimaFit), c(MarsArimaFit),
                  c(AprsArimaFit), c(MaysArimaFit), c(JunsArimaFit),
                  c(JulsArimaFit), c(AugsArimaFit), c(SepsArimaFit),
                  c(OctsArimaFit), c(NovsArimaFit), c(DecsArimaFit))
```

```
## Warning in cbind(c(JansArimaFit), c(FebsArimaFit), c(MarsArimaFit),
## c(AprsArimaFit), : number of rows of result is not a multiple of vector length
## (arg 10)
```

```r
plot_arima_coef <- function(models, c){

  model_coefficients <- c()
  errors <- c()


  for (model in 1:length(models[1, ])){
      model_coefficients <- cbind(model_coefficients, models[, model]$coef[c])
      std_error <- sqrt(diag(models[, model]$var.coef))[c]
      errors <- cbind(errors, std_error)
  }

  t_vals<-seq_along(models[1,])

  MyPlot = (plot(t_vals, model_coefficients, type='b', ylim = c(-2, 1), main = c) +
    arrows(x0=t_vals, y0=model_coefficients-2*errors, x1 = t_vals, y1=model_coefficients+2*errors, code=
    grid(nx = NULL, ny = NULL,
     lty = 2,      # Grid line type
     col = "gray", # Grid line color
     lwd = 2)      # Grid line width
}

plot_arima_coef(months_models, "ma1")
```



the negative ma1 coefficients suggest a restoring force behavior, that is a warm year is likely to be followed by a colder years. and the positive drift suggests that there is a general upward trend.