Time Series - Basics

```
setwd("C:/Users/reshed001/Desktop/OMSA_7.26.2020/ISYE 6402 - Time Series/Module 1/M1 HW")
if (!require(TSA)) install.packages("TSA")
```

```
## Loading required package: TSA
```

```
## Warning: package 'TSA' was built under R version 4.0.3
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(TSA)
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

```
library(dynlm)
```

```
## Warning: package 'dynlm' was built under R version 4.0.3
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
#Choose the 'LA Temp Monthly.csv' data set, wherever it is located on your computer
#Additionally, just skip this step and replace ‘fname’ with the files direct location

data = read.csv('LA Temp Monthly.csv',header = TRUE)
head(data)
```

```
##     Date Temp
## 1 195001 46.8
## 2 195002 51.9
## 3 195003 54.1
## 4 195004 58.1
## 5 195005 58.1
## 6 195006 61.2
```

```
data = data[,2]

#Convert to TS data in proper frame
temp = ts(data,start=c(1950,1),freq=12)
```

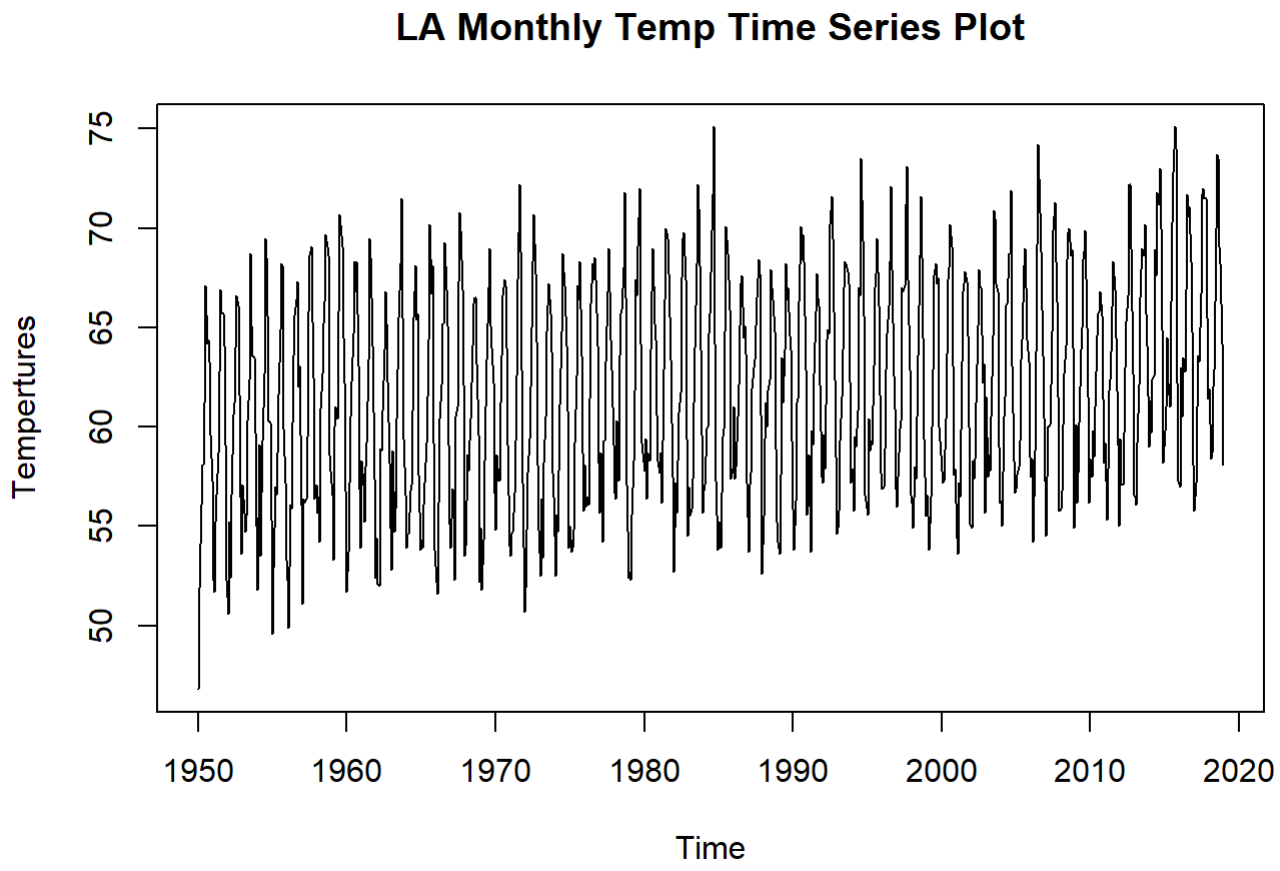
Question 1: Temperature Analysis

Question 1a: Exploratory Data Analysis

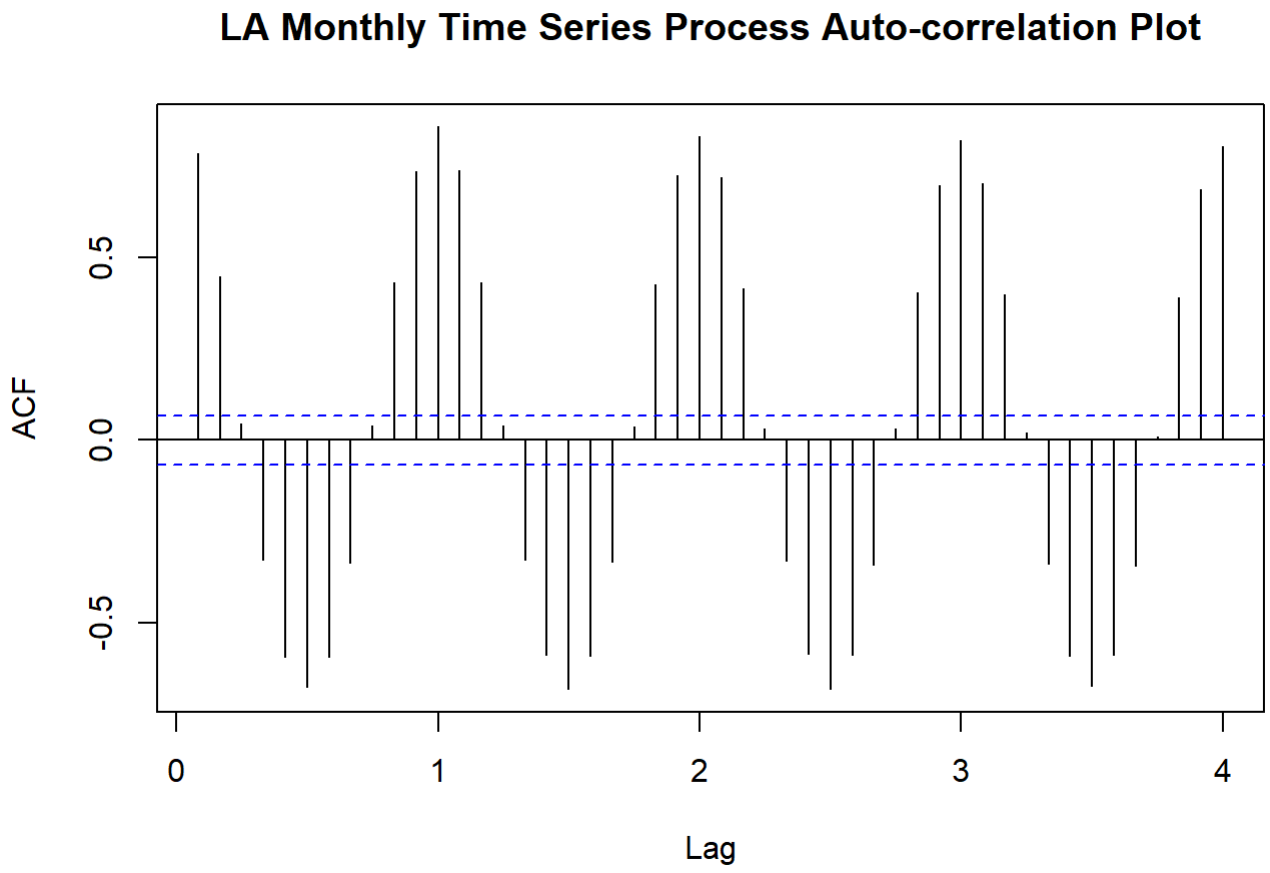
Plot the Time Series and ACF plots. Comment on the main features, and identify what (if any) assumptions of stationarity are violated. Hint: Before plotting, can you infer anything from the nature of the data?

On its own, which type of model do you think will fit the data best: trend or seasonality fitting?

```
ts.plot(temp, ylab = "Tempertures", main = "LA Monthly Temp Time Series Plot")
```




```
acf(temp,lag.max=12*4,main="LA Monthly Time Series Process Auto-correlation Plot")
```




```
print(paste("The lowest Temp within our time interval is: ", min(temp), "and the highest is: ",
 max(temp)))
```

```
## [1] "The lowest Temp within our time interval is:  46.8 and the highest is:  75.1"
```

Answer

We are analyzing the Los Angeles monthly average temperature from January 1950 through December 2018. Due to the fact that this is a temperature data, we should expect a seasonality component (e.g. higher temperatures during the summer seasons and lower during the winter seasons). However, when it comes to the trend component in temperature-based time series process, this is a long debate between the global warming/ climate change believers and non-believers. Since I believe that global warming is occurring, I would expect to see an upward trend.

After plotting, we can see based on the time series plot that there is seasonality and an overall slight upward trend. The lowest monthly average temperature within our time interval is below 50 degrees and the highest is ~ 75 degrees. A stationary process is described by the behavior of its auto-covariance or auto-correlation function. Here we are plotting the auto-correlation (ACF) plot of the time series process (before removing trend and seasonality components). The ACF plot shows a clear seasonality pattern, which repeats for each block of lags (i.e. 1, 2,.. on the x-axis). As indicated in the time series process plot, there is a slight upward trend and since the trend slowly changes over the years, we cannot observe it in the ACF plot. Due to the fact that the sample auto-correlation is large and outside of the significance bands, we can conclude that the time series process is non-stationary.

I believe that a seasonality model will fit the model better since the seasonality component seems to be greater than the trend's component (however, seasonality AND trend would probably be even better).

Question 1b: Trend Estimation

Fit the following trend estimation models:

Moving average

Parametric quadratic polynomial

Local Polynomial

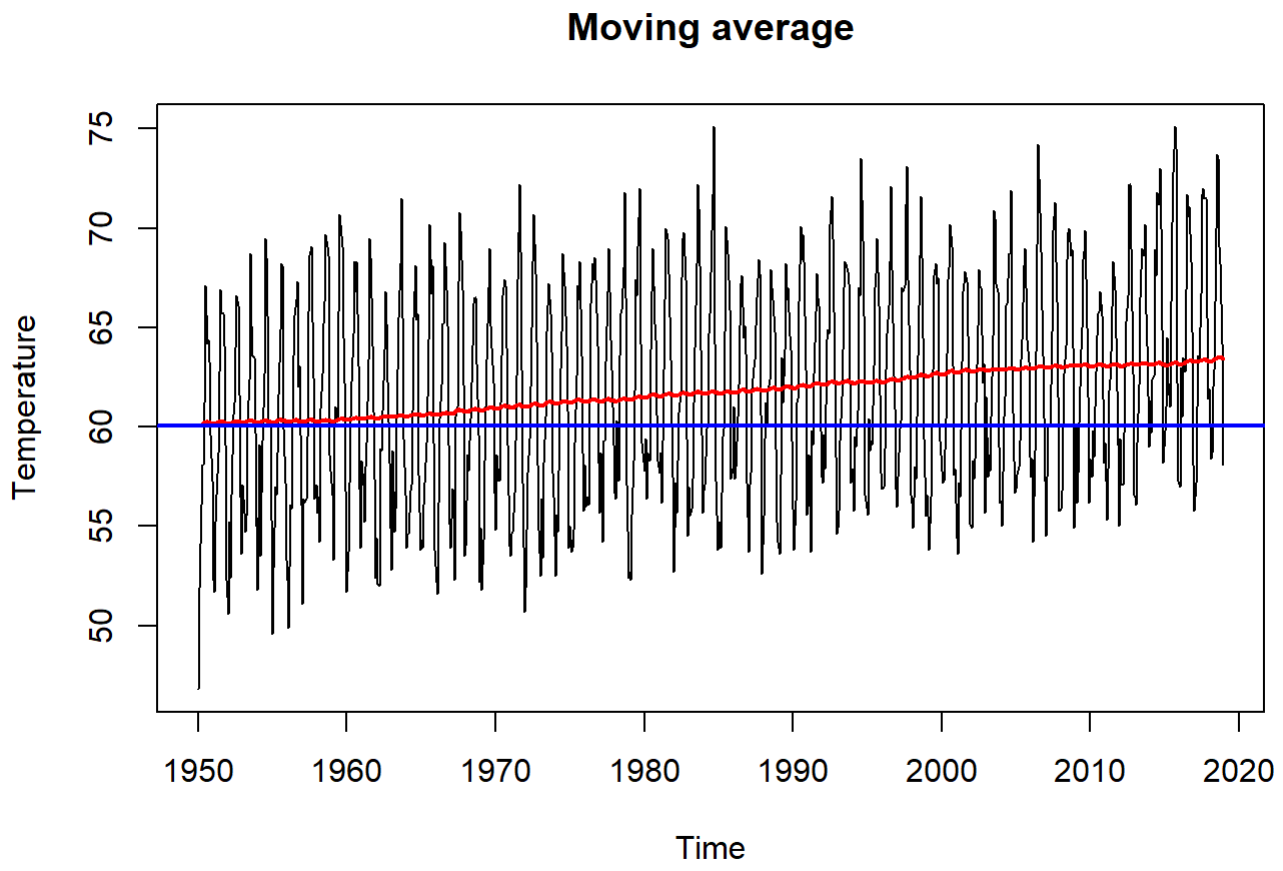
Splines

Overlay the fitted values on the original time series. Construct and plot the residuals with respect to time and ACF of residuals. Comment on the four models fit and on the appropriateness of the stationarity assumption of the residuals.

```
# Create equally spaced time points for fitting trends
time.pts = c(1:length(temp))
time.pts = c(time.pts - min(time.pts))/ max(time.pts)

# Fit a Moving Average
mav.fit = ksmooth(time.pts, temp, kernel = "box")
summary(mav.fit)
```

```
##   Length Class  Mode
## x 828    -none- numeric
## y 828    -none- numeric
```

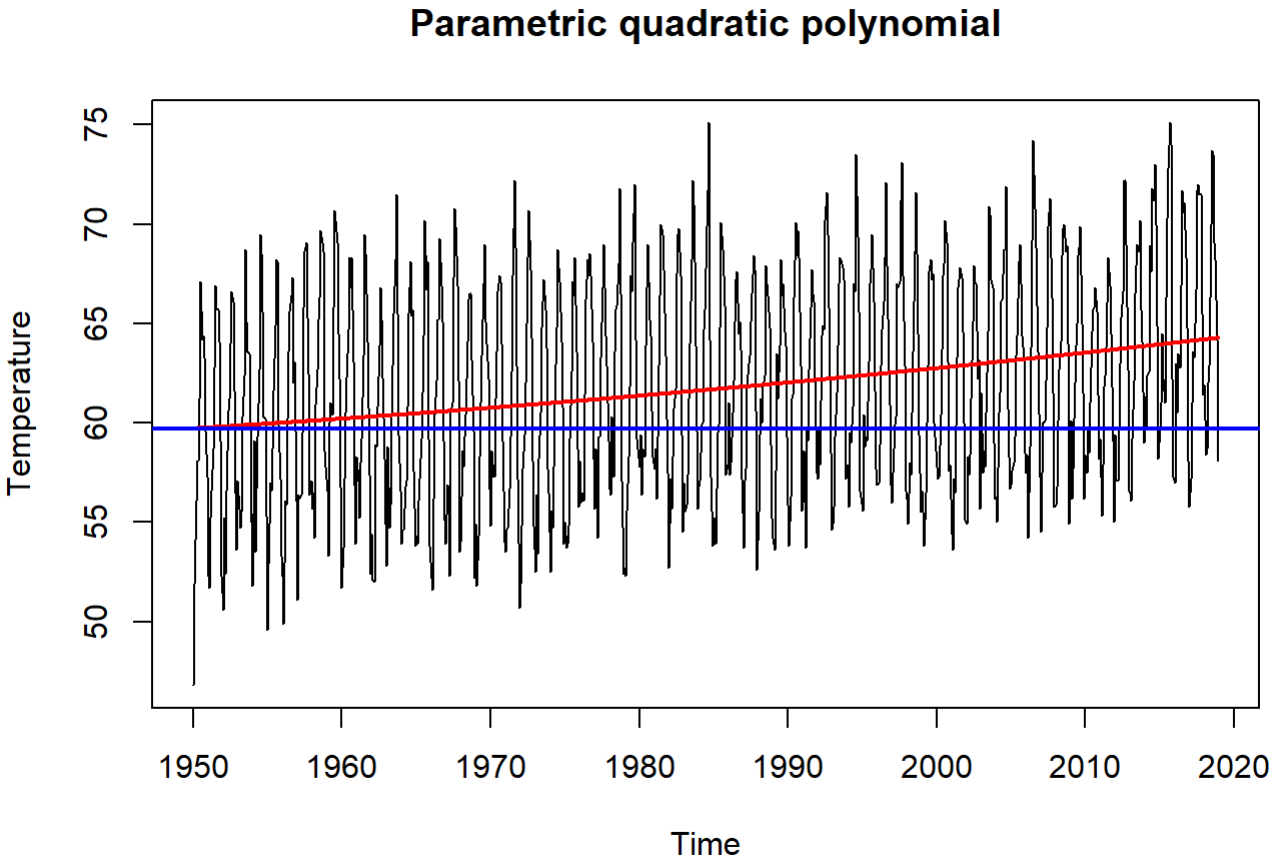
```
temp.fit.mav = ts(mav.fit$y, start = 1950, frequency = 12)
ts.plot(temp,ylab="Temperature", main="Moving average")
lines(temp.fit.mav, lwd=2, col="red")
abline(temp.fit.mav[1], 0, lwd=2, col="blue")
```




```
# Fit Parametric quadratic polynomial
x1 = time.pts
x2 = time.pts^2
lm.fit = lm(temp ~ x1+x2)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = temp ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9226  -4.4207  -0.2526   4.6385  13.4337
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59.7226     0.5489 108.809   <2e-16 ***
## x1            3.1639     2.5384   1.246    0.213
## x2            1.4030     2.4606   0.570    0.569
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.277 on 825 degrees of freedom
## Multiple R-squared:  0.05925,    Adjusted R-squared:  0.05697
## F-statistic: 25.98 on 2 and 825 DF,  p-value: 1.143e-11
```

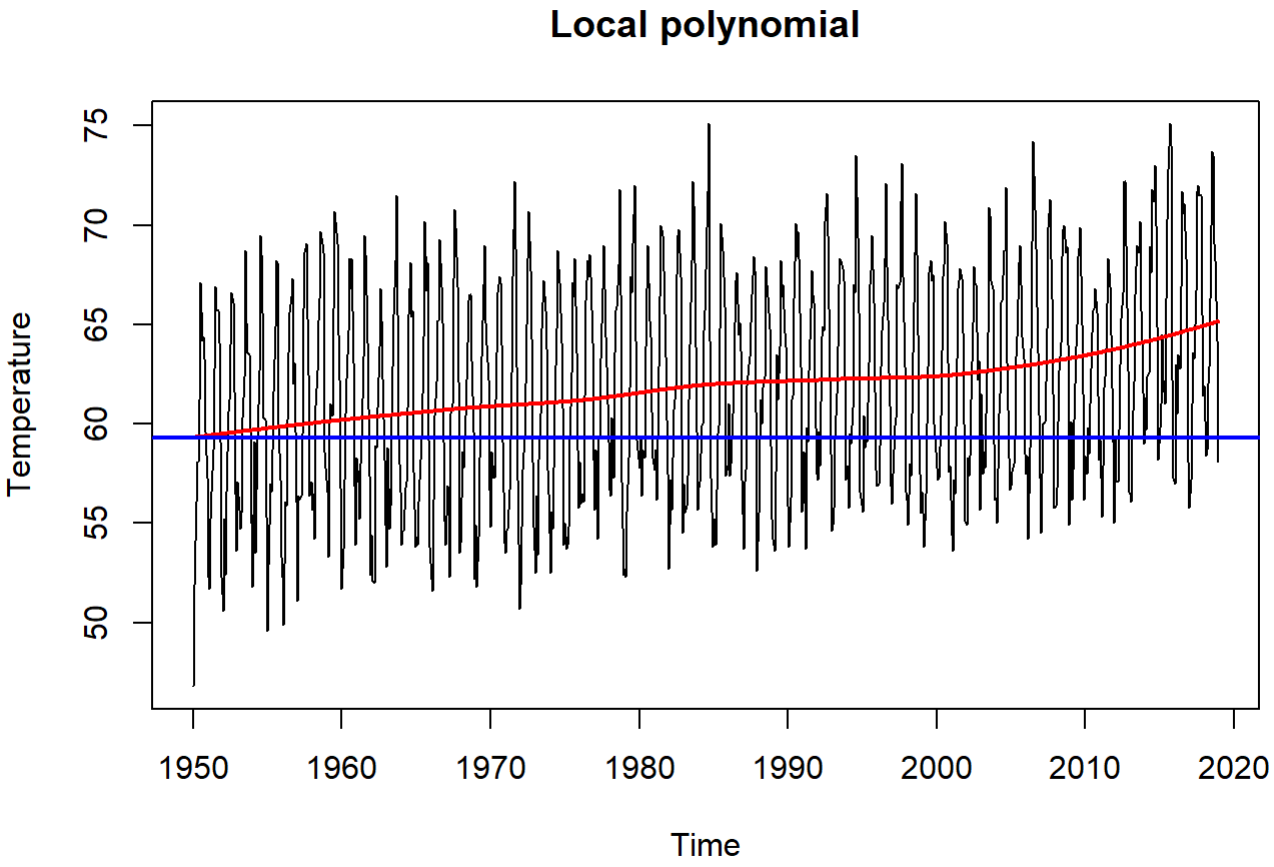
```
temp.fit.lm = ts(fitted(lm.fit), start = 1950, frequency = 12)
ts.plot(temp, ylab = "Temperature", main="Parametric quadratic polynomial")
lines(temp.fit.lm, lwd=2, col="red")
abline(temp.fit.lm[1], 0, lwd=2, col="blue")
```




```
# Fit Local Polynomial
loc.fit = loess(temp ~ time.pts)
summary(loc.fit)
```

```
## Call:
## loess(formula = temp ~ time.pts)
##
## Number of Observations: 828
## Equivalent Number of Parameters: 4.34
## Residual Standard Error: 5.27
## Trace of smoother matrix: 4.72  (exact)
##
## Control settings:
##   span     :  0.75
##   degree   :  2
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```

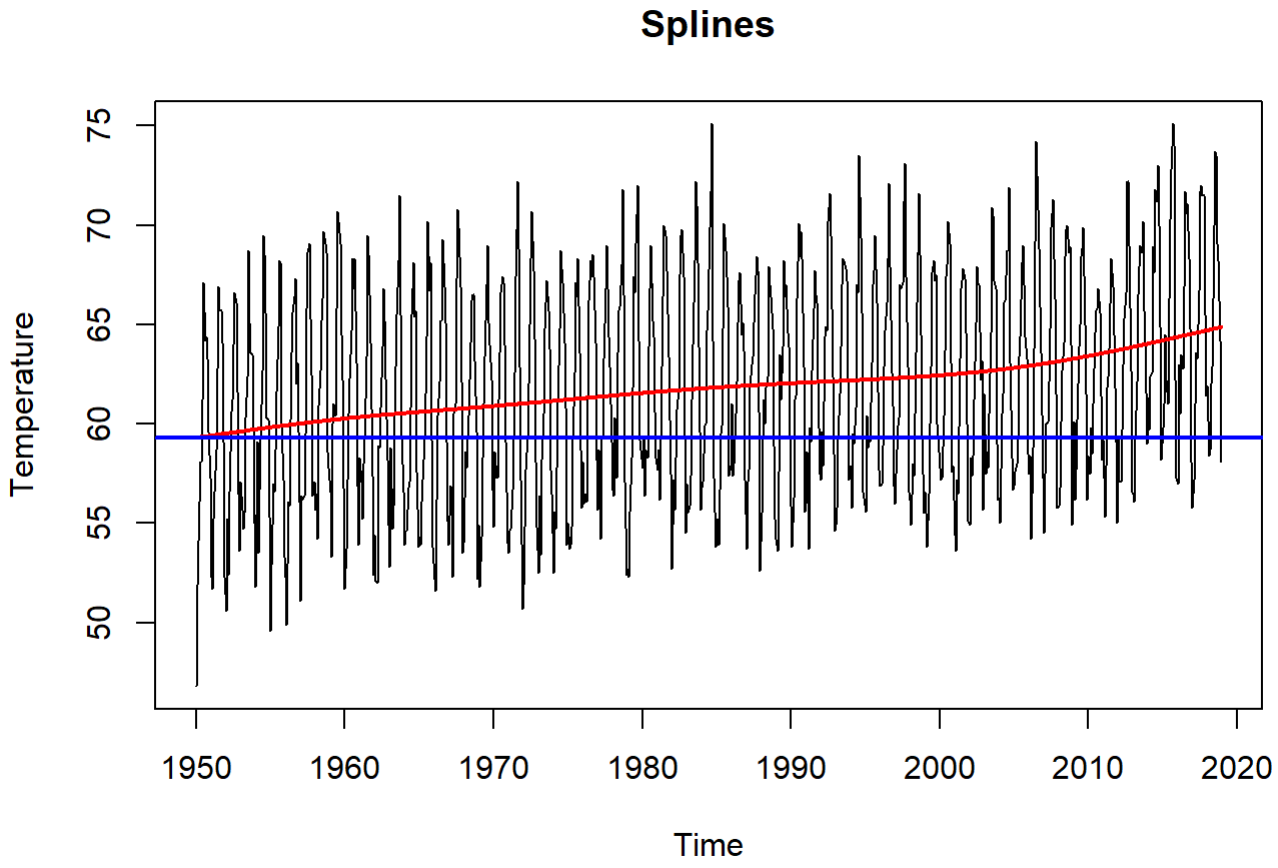
```
temp.fit.loc = ts(fitted(loc.fit), start = 1950, frequency = 12)
ts.plot(temp, ylab = "Temperature", main="Local polynomial")
lines(temp.fit.loc, lwd=2, col="red")
abline(temp.fit.loc[1], 0, lwd=2, col="blue")
```



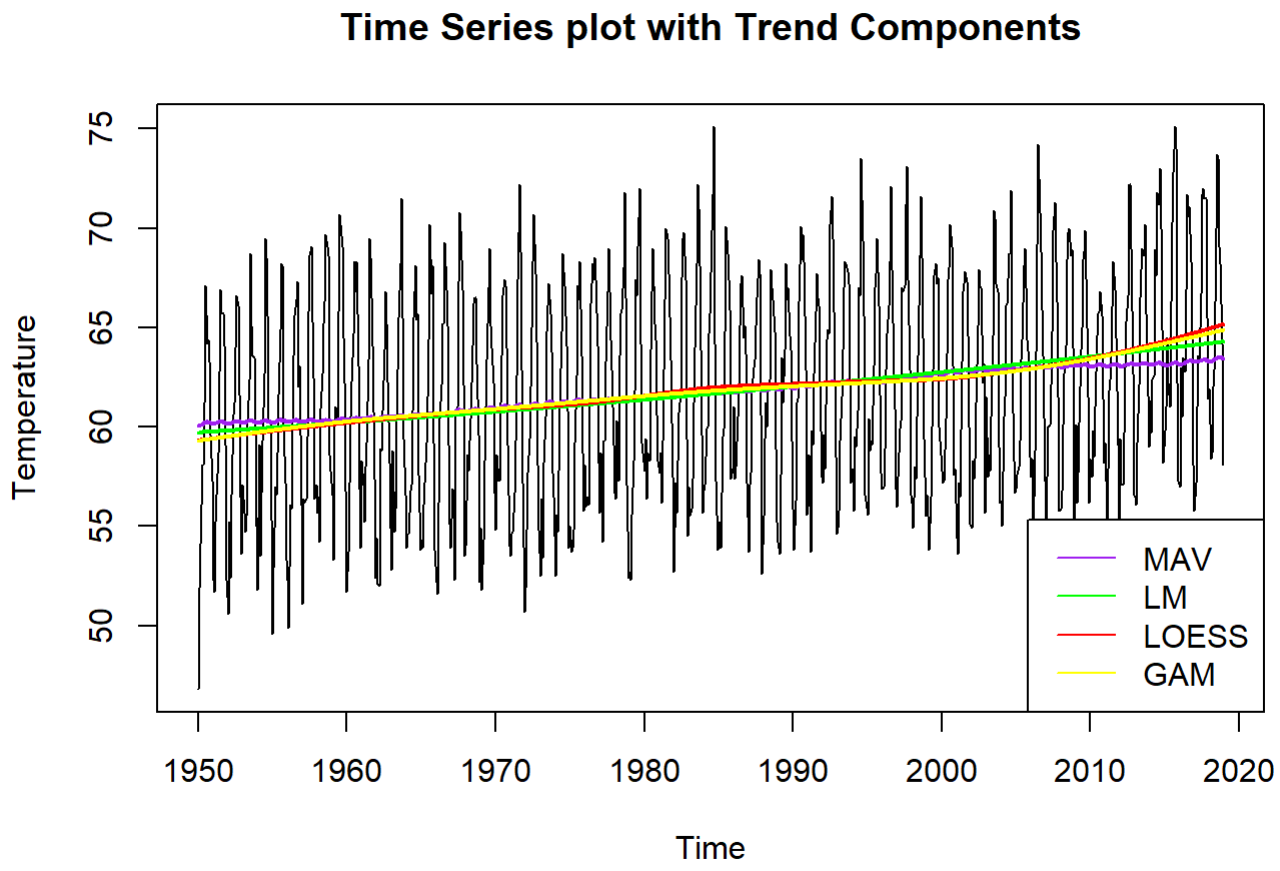
```
# Fit Splines
gam.fit = gam(temp ~ s(time.pts))
summary(gam.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## temp ~ s(time.pts)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.7694     0.1831   337.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F  p-value
## s(time.pts) 3.408  4.228 12.73 2.07e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0599   Deviance explained = 6.38%
## GCV = 27.913  Scale est. = 27.765    n = 828
```

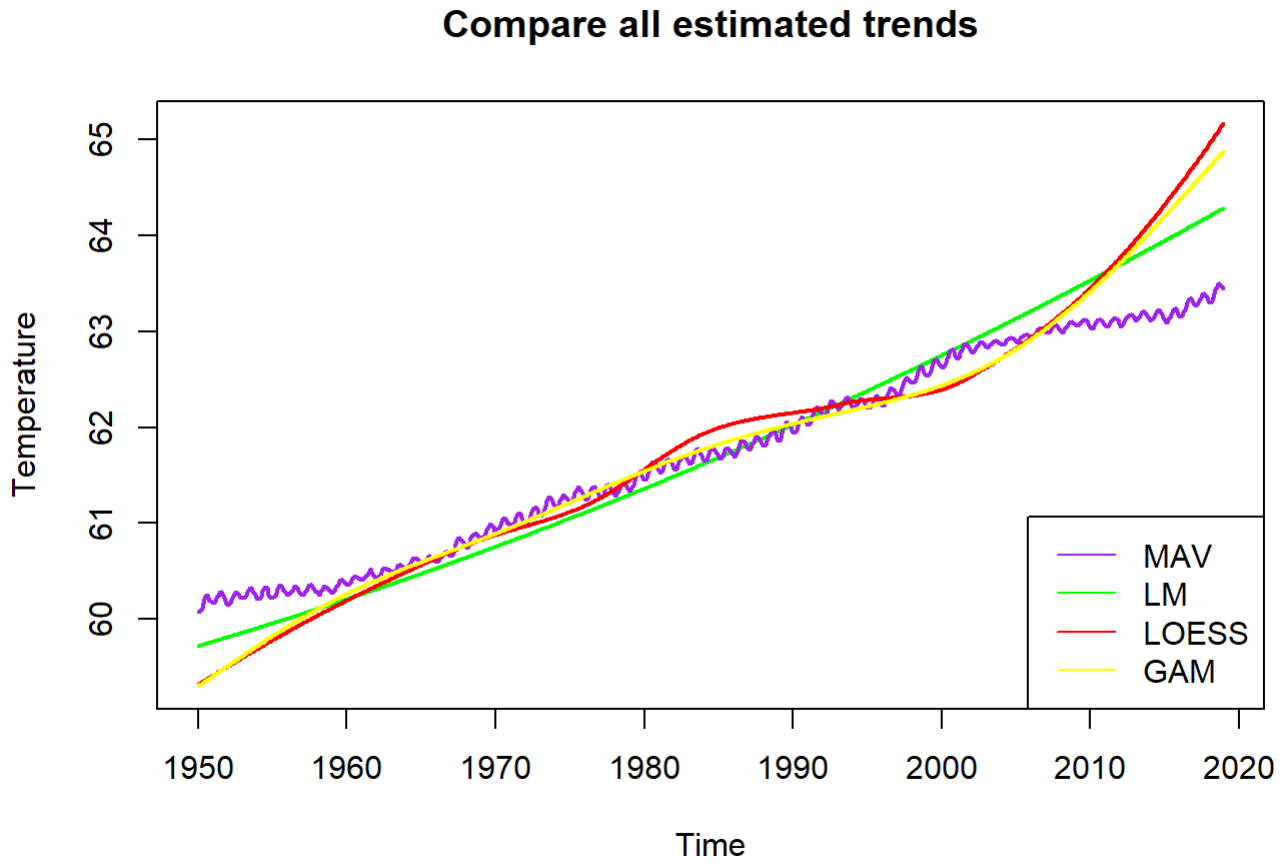
```
temp.fit.gam = ts(fitted(gam.fit), start = 1950, frequency = 12)
ts.plot(temp, ylab = "Temperature", main="Splines")
lines(temp.fit.gam, lwd=2, col="red")
abline(temp.fit.gam[1], 0, lwd=2, col="blue")
```



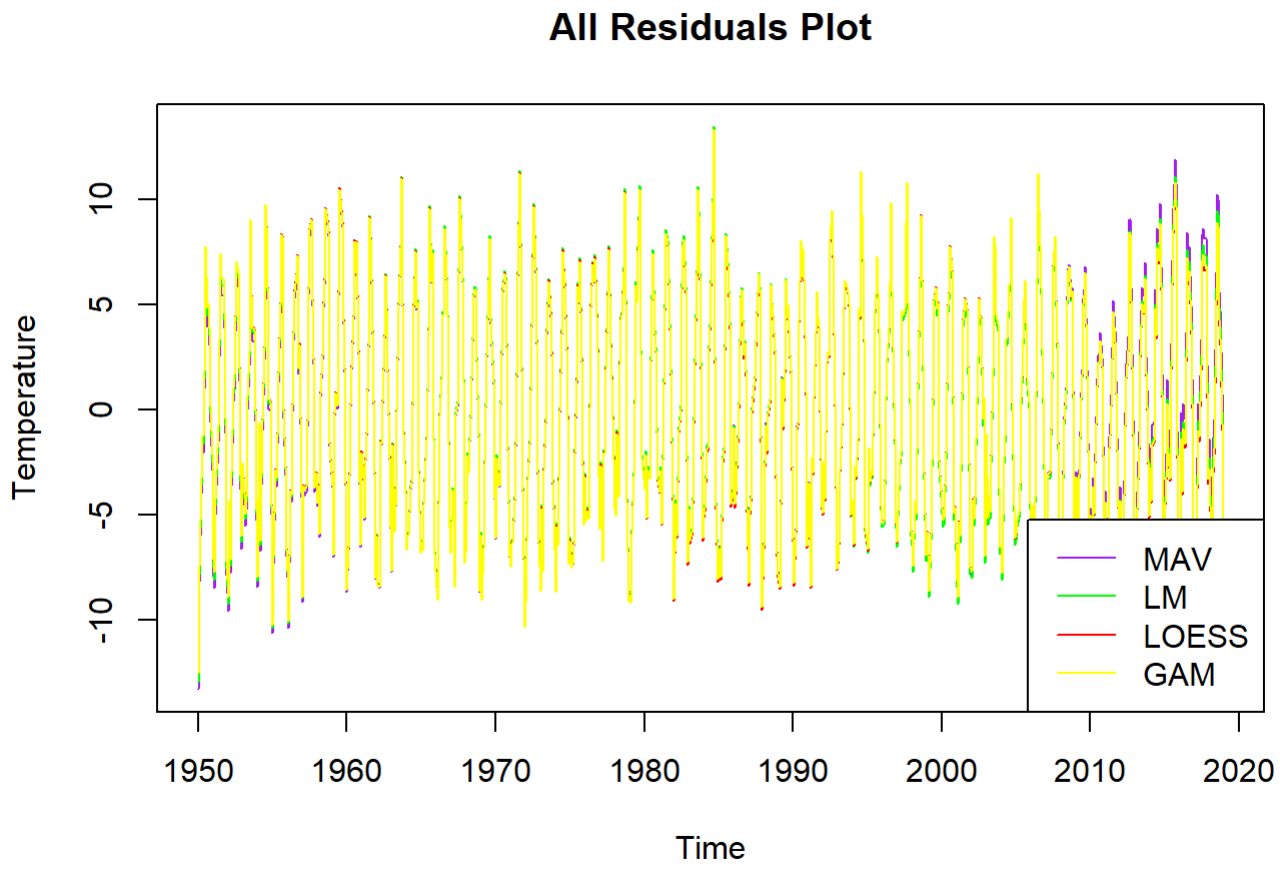

```
# Overlay the fitted values on the original time series
ts.plot(temp, ylab="Temperature", main="Time Series plot with Trend Components")
lines(temp.fit.mav, lwd=2, col="purple")
lines(temp.fit.lm, lwd=2, col="green")
lines(temp.fit.loc, lwd=2, col="red")
lines(temp.fit.gam, lwd=2, col="yellow")
legend("bottomright",legend=c("MAV","LM","LOESS","GAM"),
       lty = 1,col=c(" purple","green","red","yellow"))
```



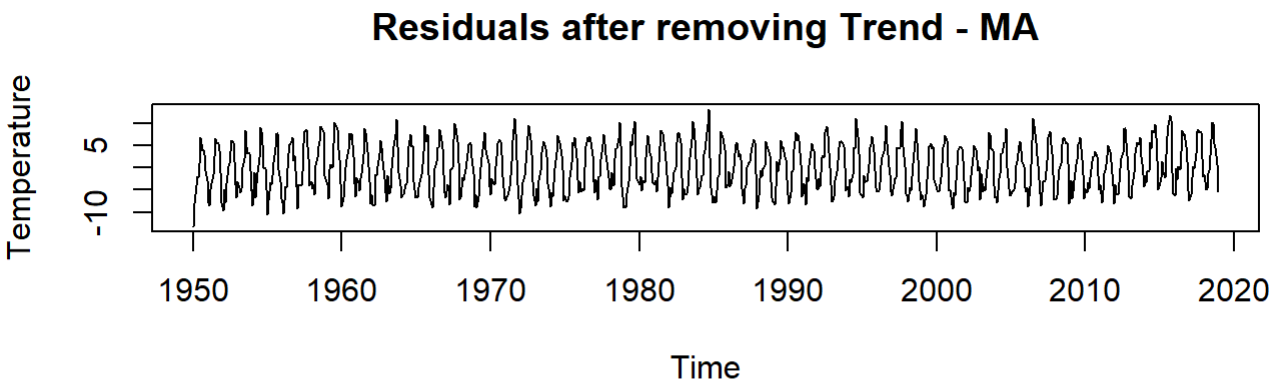

```
# Compare all estimated trends
all.val = c(temp.fit.mav,temp.fit.lm,temp.fit.loc,temp.fit.gam)
ylim = c( min(all.val), max(all.val))
ts.plot(temp.fit.lm,lwd =2,col="green", ylim=ylim,ylab ="Temperature", main="Compare all estimat
ed trends")
lines(temp.fit.mav, lwd=2, col="purple")
lines(temp.fit.loc, lwd=2, col="red")
lines(temp.fit.gam, lwd=2, col="yellow")
legend("bottomright",legend=c("MAV","LM","LOESS","GAM"),
       lty = 1,col=c(" purple","green","red","yellow"))
```

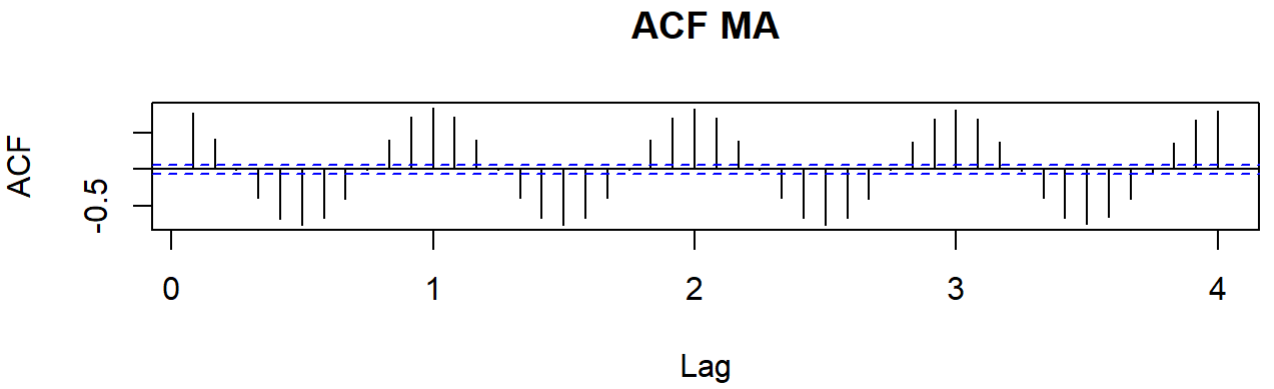



```
# Residuals
res_mav = temp - temp.fit.mav
res_lm = temp - temp.fit.lm
res_loc = temp - temp.fit.loc
res_gam = temp - temp.fit.gam

# Plot all residuals
ts.plot(res_mav, res_lm, res_loc, res_gam, gpars=list(ylab="Temperature", main="All Residuals Pl
ot", col=c("purple", "green", "red", "yellow")))
legend("bottomright",legend=c("MAV","LM","LOESS","GAM"),
       lty = 1,col=c(" purple","green","red","yellow"))
```

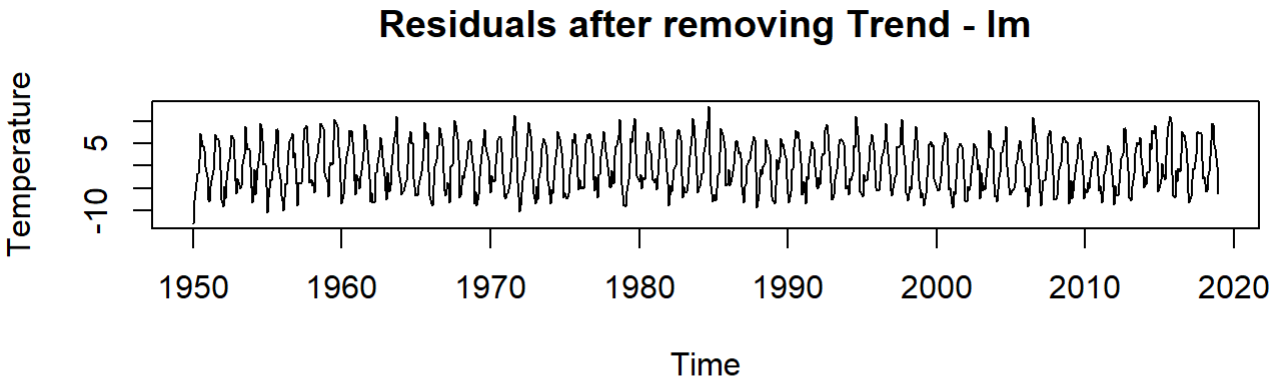


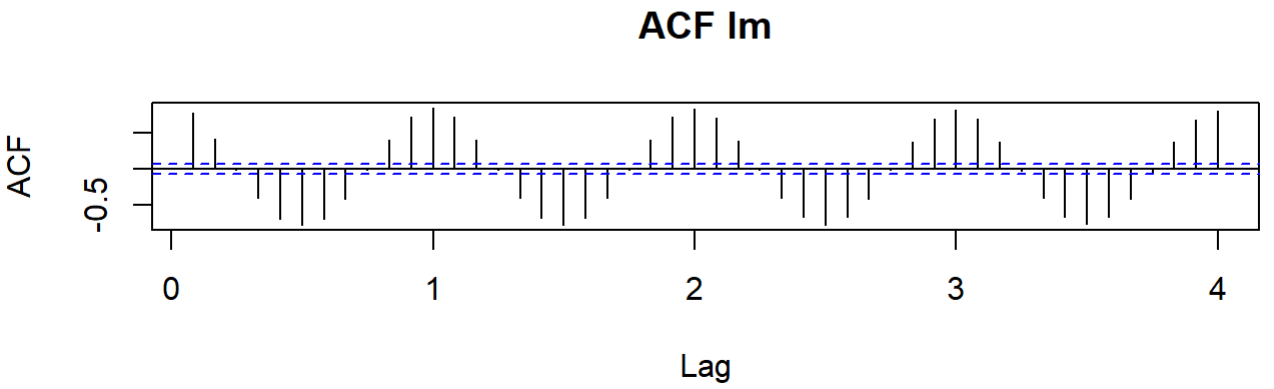

```
par(mfrow = c(2, 1))
ts.plot(res_mav, ylab="Temperature", main="Residuals after removing Trend - MA")
acf(res_mav, lag.max=12*4, main="ACF MA")
```



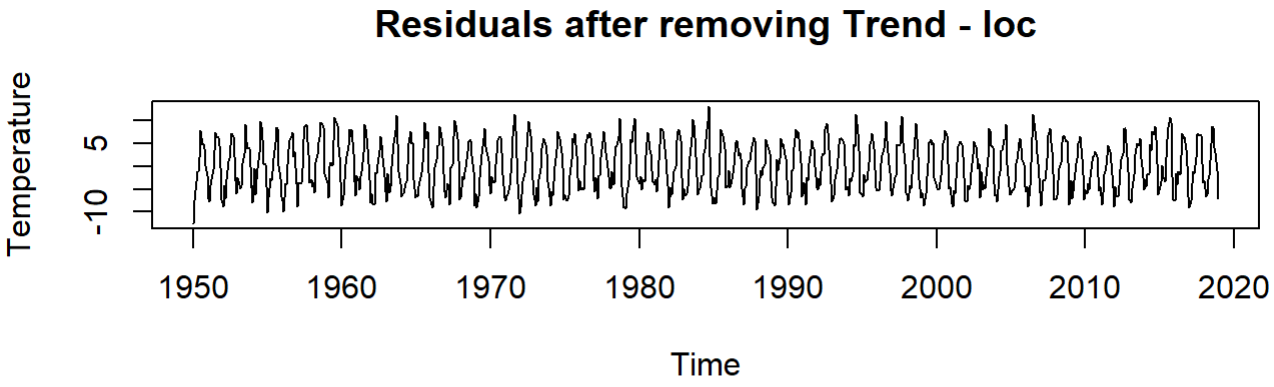


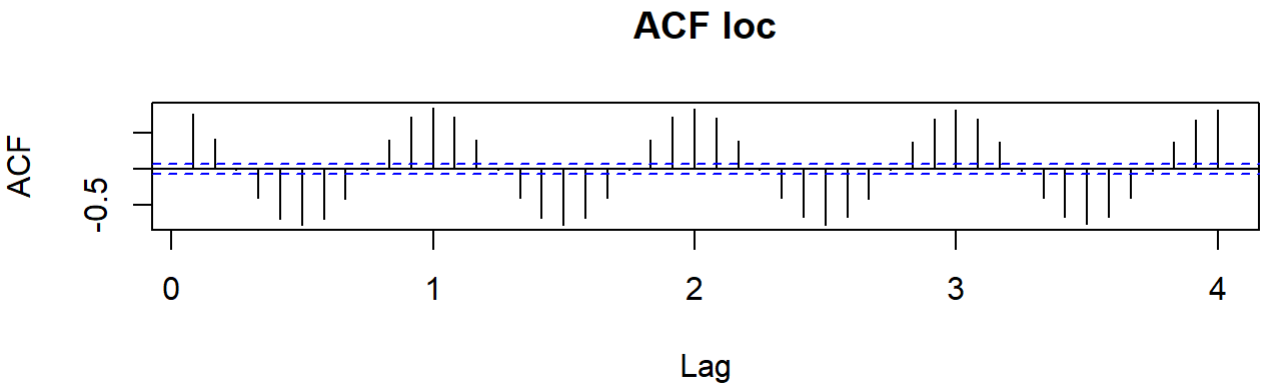

```
par(mfrow = c(2, 1))
ts.plot(res_lm, ylab="Temperature", main="Residuals after removing Trend - lm")
acf(res_lm, lag.max=12*4, main="ACF lm")
```



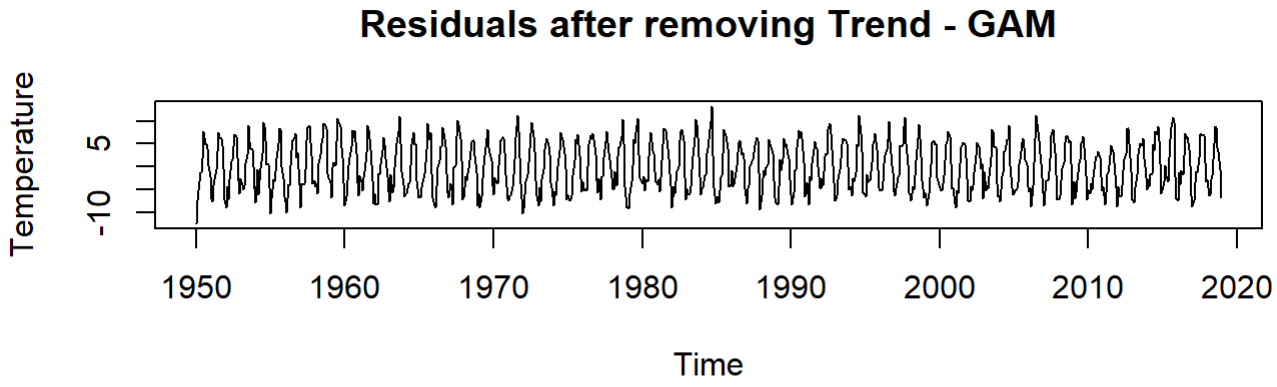


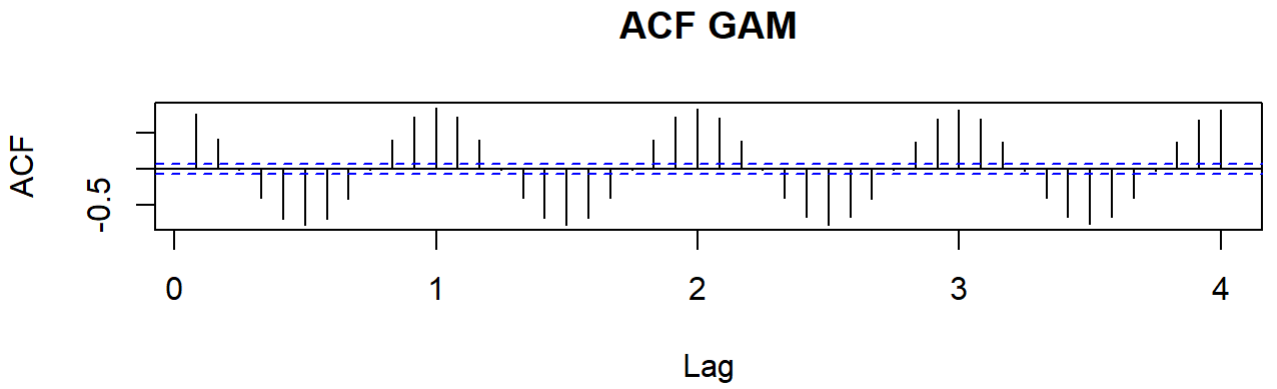

```
par(mfrow = c(2, 1))
ts.plot(res_loc, ylab="Temperature", main="Residuals after removing Trend - loc")
acf(res_loc, lag.max=12*4, main="ACF loc")
```






```
par(mfrow = c(2, 1))
ts.plot(res_gam, ylab="Temperature", main="Residuals after removing Trend - GAM")
acf(res_gam, lag.max=12*4, main="ACF GAM")
```





Answer

Based on the plots above we can see that all four models indicate a slight upward trend. However, we can see that all models' R^2 are extremely low, an indication that the trend does not explain the variability in the monthly average temperature over years. The wiggly line indicated by the moving average incorporates the seasonality effect and hence not a good trend estimate. The trend lines of the local polynomial and splines are very close to each other and the parametric quadratic polynomial, as expected, shows a linear increase. From the "Compare all estimated trends" plot, we can conclude that since 1950 there has been a trend of a temperature increase of approximately 5 degrees. With regards to stationarity, based on the four ACF plots, we can conclude that non of the four plots indicates a stationary process when removing the trend component. We can easily observe the seasonality across all four ACF plots and out-of-band auto-correlation across all lags.

Question 1c: Seasonality Estimation

Seasonality Estimation:

Fit the following seasonality estimation models.

Categorical Linear Regression (ANOVA)

COS-SIN

Overlay the fitted values on the original time series. Construct and plot the residuals with respect to time and ACF plots. Comment on how the two models fit and on the appropriateness of the stationarity assumption of the residuals. Also compare the fits to those in part B and comment if your initial prediction was correct.

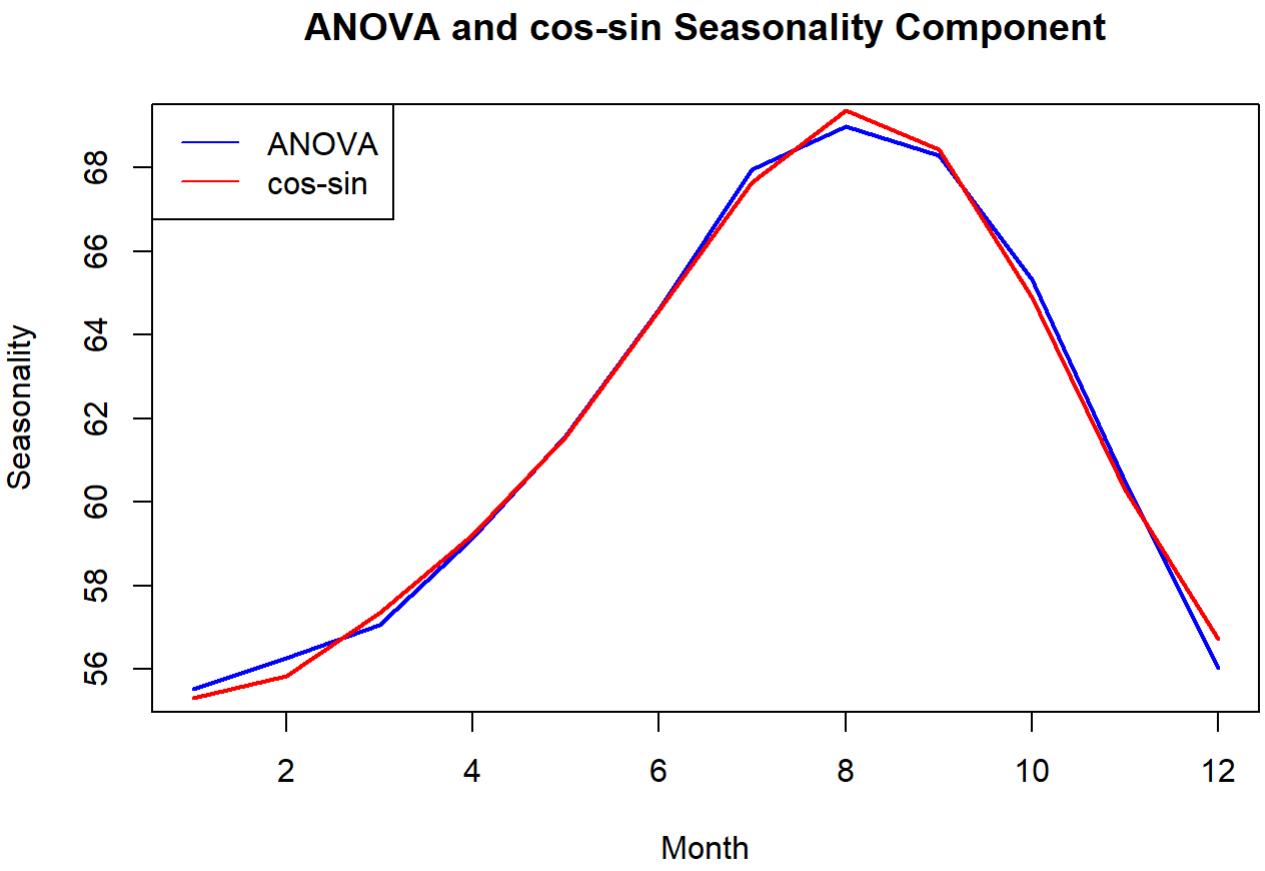
```
# Seasonal mean effects/without intercept
ANOVA = dynlm(temp ~ season(temp) - 1)
summary(ANOVA)
```

```
##
## Time series regression with "ts" data:
## Start = 1950(1), End = 2018(12)
##
## Call:
## dynlm(formula = temp ~ season(temp) - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7145 -1.5855 -0.0319  1.6000  8.6768
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## season(temp)Jan  55.5145     0.2864   193.9   <2e-16 ***
## season(temp)Feb  56.2565     0.2864   196.5   <2e-16 ***
## season(temp)Mar  57.0478     0.2864   199.2   <2e-16 ***
## season(temp)Apr  59.1333     0.2864   206.5   <2e-16 ***
## season(temp)May  61.5942     0.2864   215.1   <2e-16 ***
## season(temp)Jun  64.6159     0.2864   225.7   <2e-16 ***
## season(temp)Jul  67.9565     0.2864   237.3   <2e-16 ***
## season(temp)Aug  68.9826     0.2864   240.9   <2e-16 ***
## season(temp)Sep  68.3000     0.2864   238.5   <2e-16 ***
## season(temp)Oct  65.3232     0.2864   228.1   <2e-16 ***
## season(temp)Nov  60.4754     0.2864   211.2   <2e-16 ***
## season(temp)Dec  56.0333     0.2864   195.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.379 on 816 degrees of freedom
## Multiple R-squared:  0.9985, Adjusted R-squared:  0.9985
## F-statistic: 4.682e+04 on 12 and 816 DF,  p-value: < 2.2e-16
```

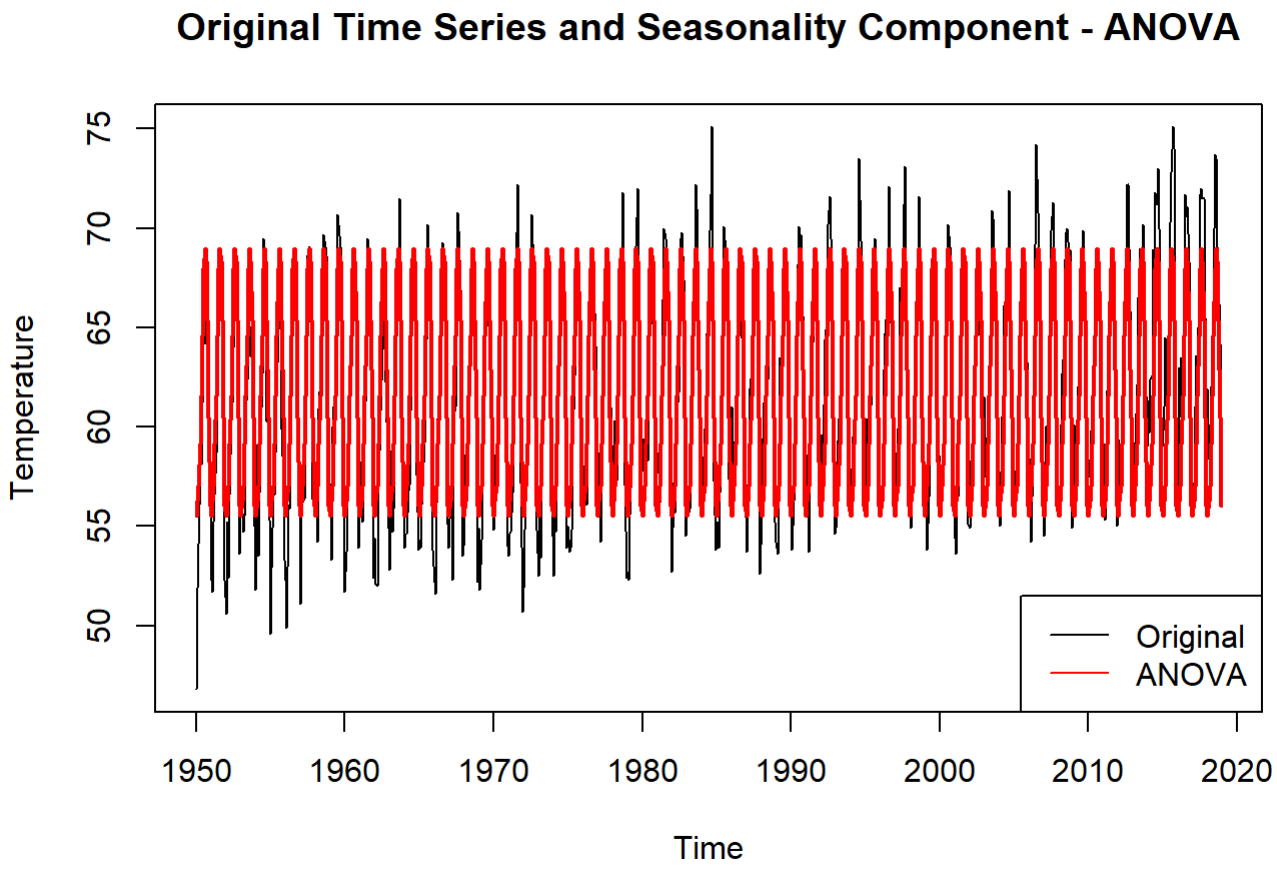
```
st1 = coef (ANOVA)

# Estimate seasonality using cos-sin model
cos_sin = dynlm(temp ~ harmon(temp,2))
summary(cos_sin)
```

```
##
## Time series regression with "ts" data:
## Start = 1950(1), End = 2018(12)
##
## Call:
## dynlm(formula = temp ~ harmon(temp, 2))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.498 -1.653 -0.148  1.572  9.099
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         61.76944    0.08308 743.484   <2e-16 ***
## harmon(temp, 2)cos1 -6.18023    0.11749 -52.600   <2e-16 ***
## harmon(temp, 2)cos2 -0.29167    0.11749  -2.482   0.0132 *
## harmon(temp, 2)sin1 -2.83955    0.11749 -24.168   <2e-16 ***
## harmon(temp, 2)sin2  1.13566    0.11749   9.666   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.391 on 823 degrees of freedom
## Multiple R-squared:  0.8074, Adjusted R-squared:  0.8065
## F-statistic: 862.6 on 4 and 823 DF,  p-value: < 2.2e-16
```

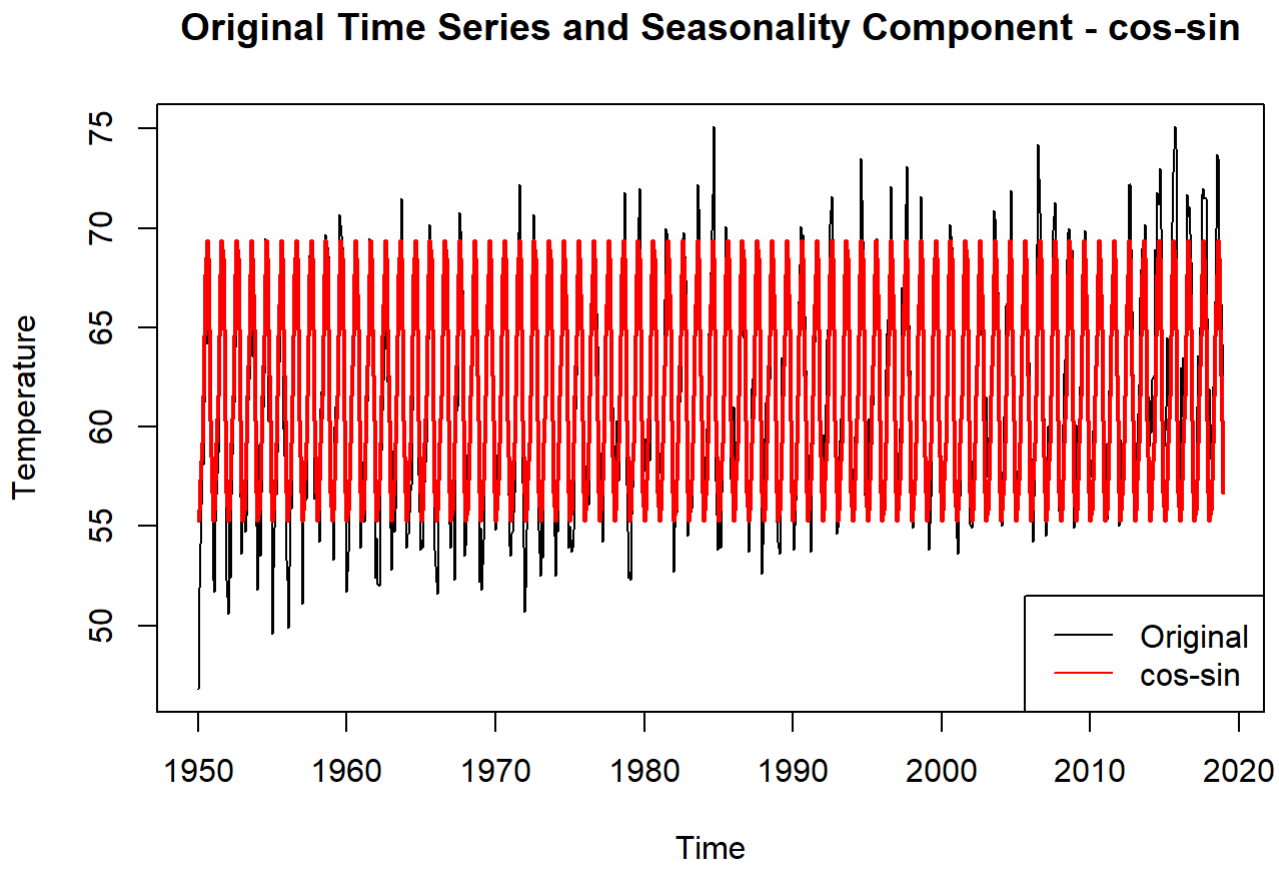
```
st2 = fitted(cos_sin)[1:12]

# Compare Seasonality Estimates
plot(1:12,st1,lwd=2,type="l", xlab ="Month", ylab ="Seasonality", main="ANOVA and cos-sin Season
ality Component", col="blue")
lines(1:12,st2,lwd=2, col="red")
legend("topleft",legend=c("ANOVA", "cos-sin"),
       lty = 1,col=c("blue", "red"))
```



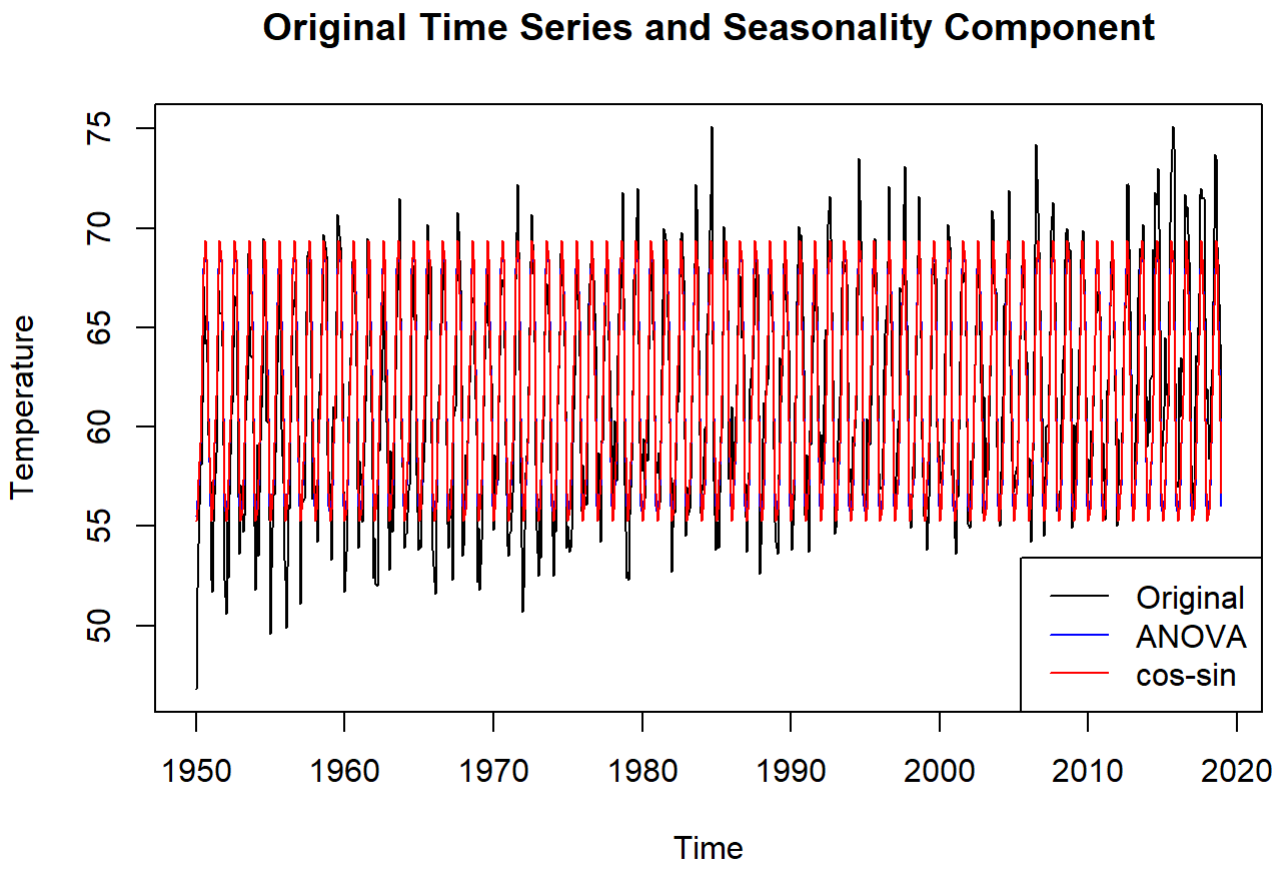

```
st1 = ts(fitted(ANOVA),start=c(1950,1),freq=12)
st2 = ts(fitted(cos_sin),start=c(1950,1),freq=12)

# Overlay the fitted values on the original time series
ts.plot(temp, ylab="Temperature", main="Original Time Series and Seasonality Component - ANOVA")
lines(st1, lwd=2, col="red")
legend("bottomright",legend=c("Original", "ANOVA"),
       lty = 1,col=c("black", "red"))
```



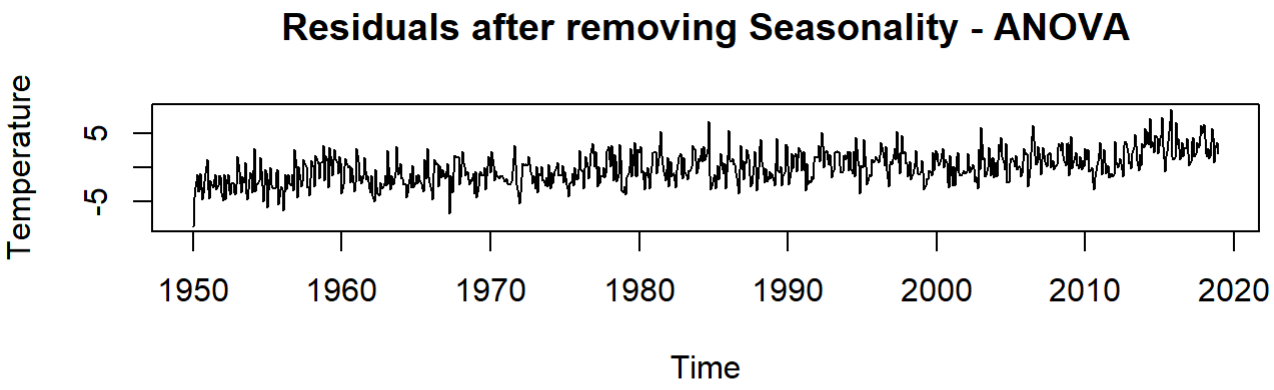

```
ts.plot(temp, ylab="Temperature", main="Original Time Series and Seasonality Component - cos-si
n")
lines(st2, lwd=2, col="red")
legend("bottomright",legend=c("Original", "cos-sin"),
       lty = 1,col=c("black", "red"))
```

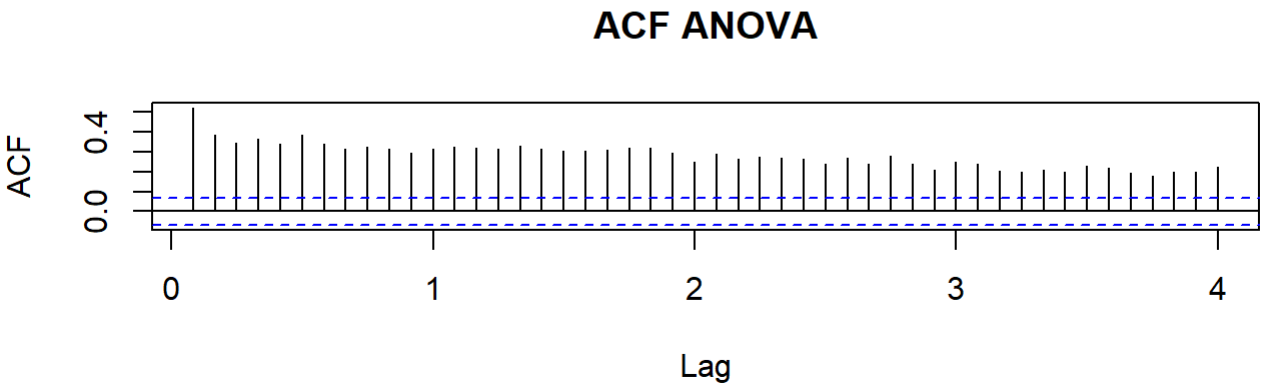



```
ts.plot(temp, st1, st2, gpars=list(ylab="Temperature", main="Original Time Series and Seasonalit
y Component", col=c("black", "blue", "red")))
legend("bottomright",legend=c("Original", "ANOVA", "cos-sin"),
       lty = 1,col=c("black", "blue", "red"))
```

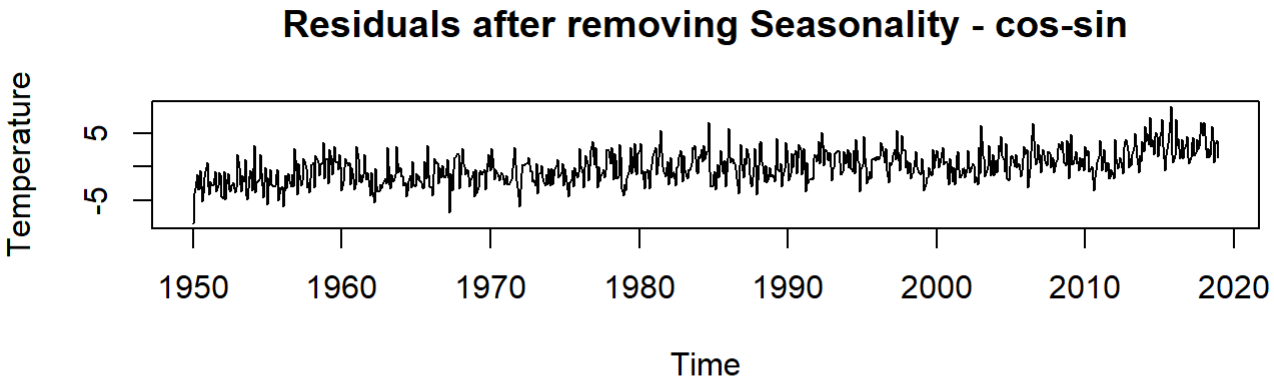


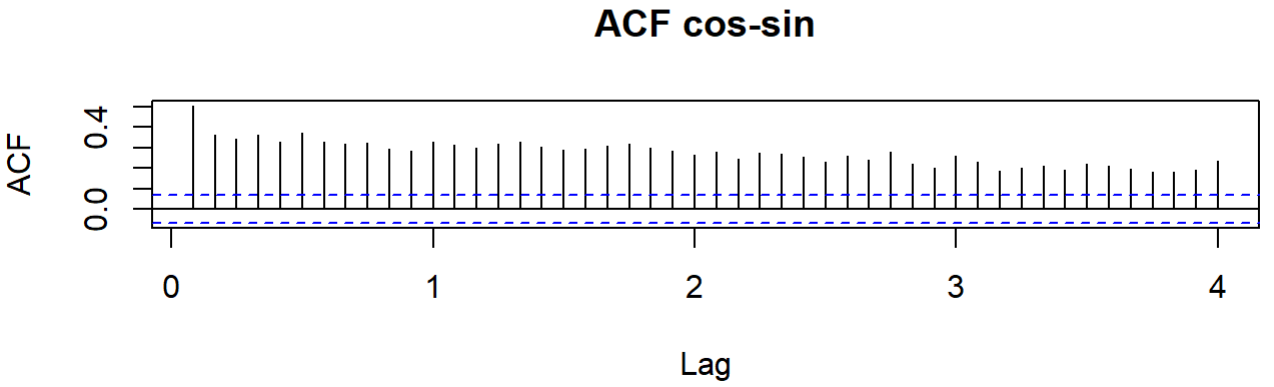

```
# Residuals
res_ANOVA = temp - st1
res_cos_sin = temp - st2

par(mfrow = c(2, 1))
ts.plot(res_ANOVA, ylab="Temperature", main="Residuals after removing Seasonality - ANOVA")
acf(res_ANOVA, lag.max=12*4, main="ACF ANOVA")
```






```
par(mfrow = c(2, 1))
ts.plot(res_cos_sin, ylab="Temperature", main="Residuals after removing Seasonality - cos-sin")
acf(res_cos_sin, lag.max=12*4, main="ACF cos-sin")
```





Answer

We can see in the ANOVA model output the means for each individual seasonal group (i.e. the mean temperature of each month) and that all coefficients are statistically significant. Moreover, the ANOVA R^2 is very large at 99.85%, indicating that the seasonality effects explain most (if not all) of the variability in the monthly average temperature over years. Observe in the cos-sin model output that all coefficients are statistically significant as well, but the model has a lower R^2 of 80.74%. As seen in the “ANOVA and cos-sin Seasonality Component” plot, both are very similar in fitting the seasonality.

Assuming the two seasonality models’ goodness of fit equals, we would normally prefer the model with fewer regression coefficients which is the cos-sin, but despite the fact that the R^2 of the ANOVA model is considerably higher, there isn’t much difference between the two models as seen in the original-seasonality component and residuals plots.

Based on the ACF residuals plots, we can see there is no seasonality pattern in either the ANOVA or the cos-sin models, but there is some cyclical and trend patterns. Moreover, based on the ACF residuals plots, the residuals process does NOT show stationarity. Comparing the results between removing trend or seasonality, no one process resulted in non-stationary residuals. Since the ACF range after removing the seasonality component is smaller than the range after removing the trend component and due to the fact that seasonality explains almost all variability in the process (99.85% per the ANOVA model), we can conclude that as initially expected, the seasonality component is greater than the trend component. However, further analysis of removing both trend and seasonality is required.

Question 2: Currency Conversion Analysis

```
library(locfit)
```

```
## Warning: package 'locfit' was built under R version 4.0.3
```

```
## locfit 1.5-9.4    2020-03-24
```

```
library(mgcv)

#Load data
data = read.csv('USD-EUR Exchange.csv',header = TRUE)
head(data)
```

```
##         DATE DEXUSEU
## 1 2000-01-05 1.01786
## 2 2000-01-12 1.02946
## 3 2000-01-19 1.01585
## 4 2000-01-26 1.00608
## 5 2000-02-02 0.97822
## 6 2000-02-09 0.98412
```

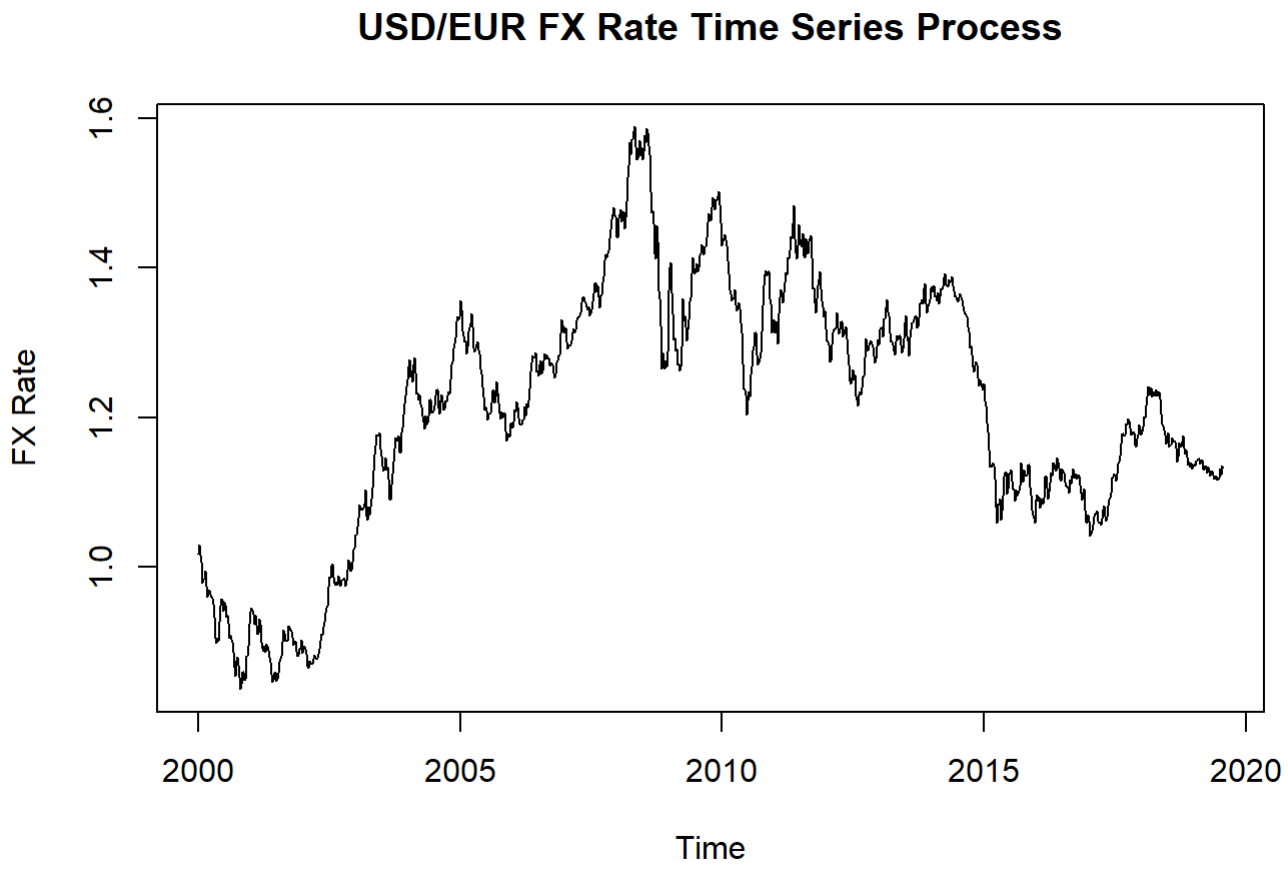
```
data = data[,2]

#Convert to TS data in proper frame
rate = ts(data,start=c(2000,1),freq=52)

#Generate differenced data
rate.dif = diff(rate)
```

Question 2a: Exploratory Data Analysis

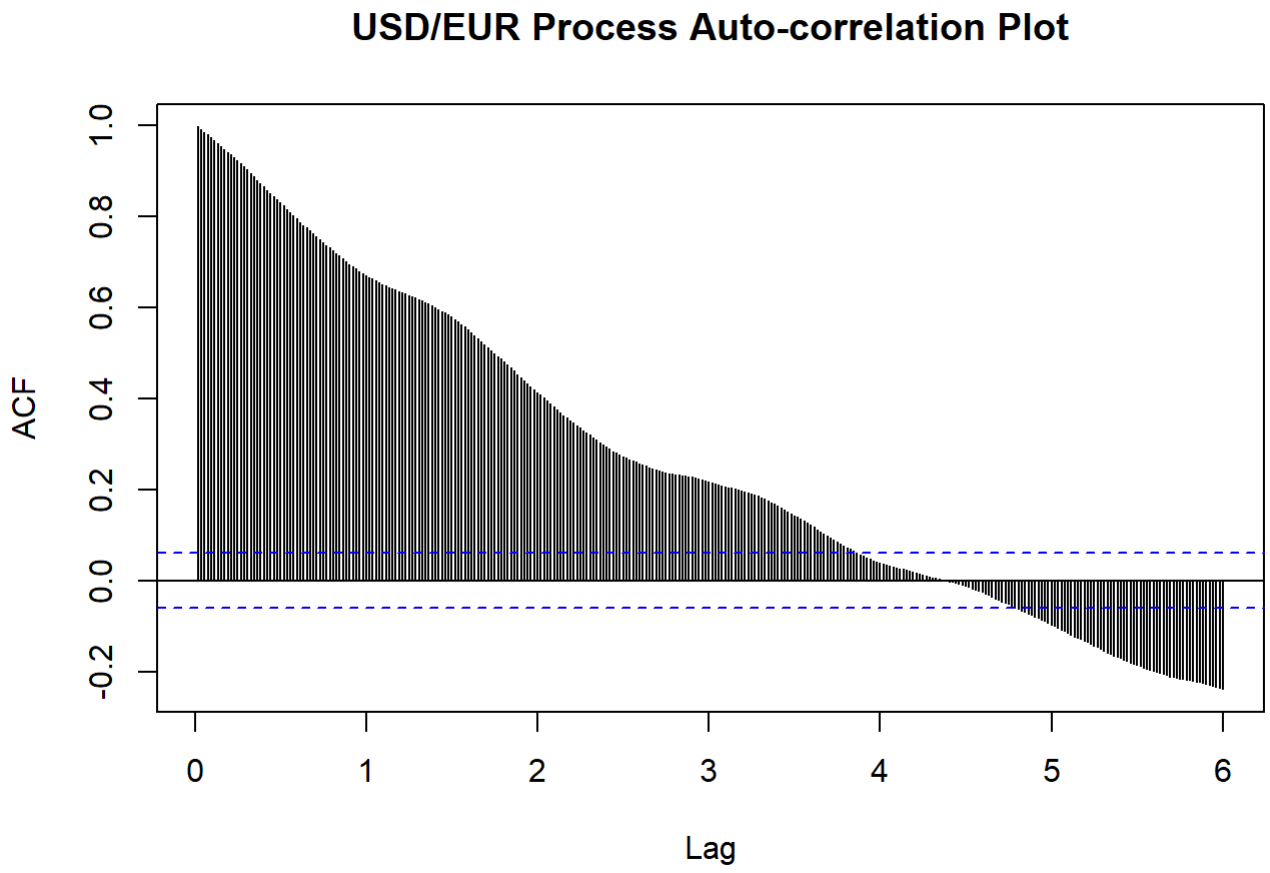
Plot the Time Series and ACF plots. Comment on the main features, and identify what (if any) assumptions of stationarity are violated.

Using the differenced rate data (‘rate.dif’), plot both the Time Series and ACF plots. Comment on the main features, and identify what (if any) assumptions of stationarity are violated. Additionally comment if you believe the differenced data is more appropriate for use in analysis. Support your position with your graphical analysis.

```
ts.plot(rate, ylab = "FX Rate", main = "USD/EUR FX Rate Time Series Process")
```



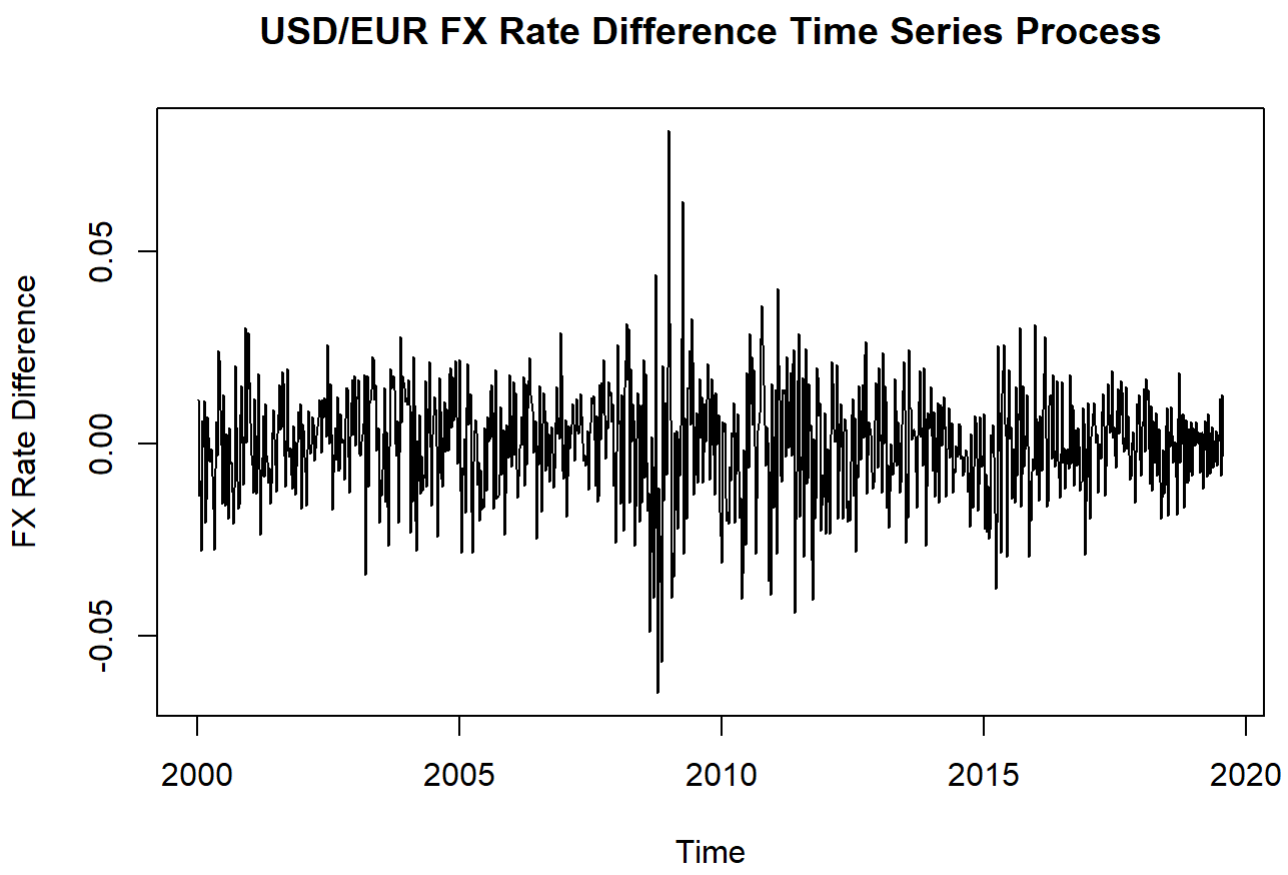

```
acf(rate,lag.max=52*6,main="USD/EUR Process Auto-correlation Plot")
```



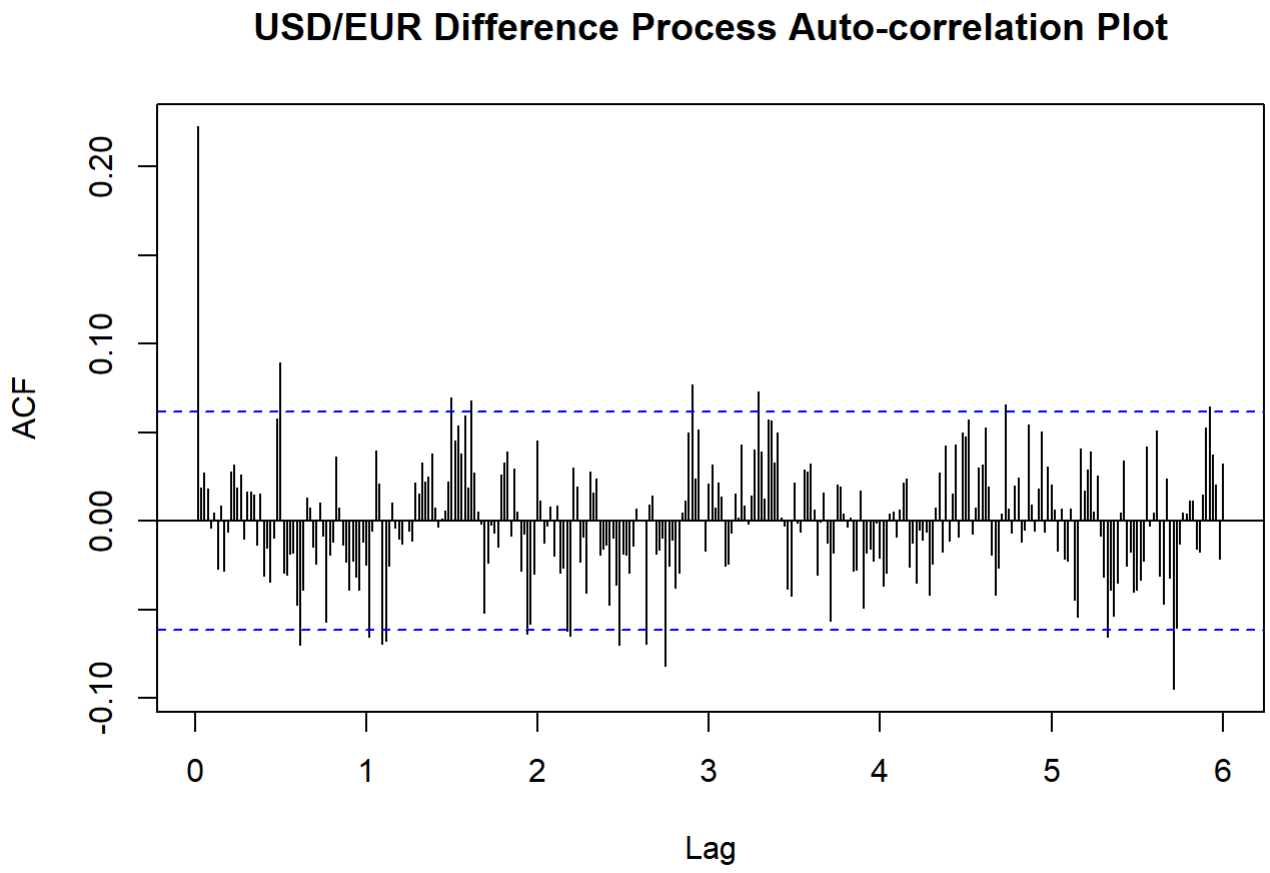
Answer

For this time interval of years 2000-2019 it is obvious that the original time series is not stationary based on both the time series and ACF plots. There is no seasonality, but there is some cyclical pattern based on the time series plot and the “waves” in the ACF plot. There seems to be a slight upward trend for approximately the first half of the time interval and a slight downward trend for the other half. Also, looking at the time series plot we can observe the mean is not constant across time and variability also changes which is another indication of non-stationarity. In addition, beside the fact that the auto-correlation is outside the bands (high auto-correlation), we can see the somewhat linear decay in auto-correlation until lag ~ 4 which indicates non-stationarity.

```
ts.plot(rate.dif, ylab = "FX Rate Difference", main = "USD/EUR FX Rate Difference Time Series Pr
ocess")
```




```
acf(rate.dif,lag.max=52*6,main="USD/EUR Difference Process Auto-correlation Plot")
```



Answer

Looking at the FX difference plots, we can see there is no trend and the values lay around the zero. However, there are a couple of outliers around the year 2009. We can also observe that there is no seasonal pattern but there is some slight cyclical component, but overall the process seems to be stationary as indicated by the fact that the mean and variability in the time series process are for the most part constants and by the ACF plot laying within the significance bands.

The differenced data is more appropriate for use in the analysis, because it indicates (weakly) stationarity while the original process does not.

Question 2b: Trend-Seasonality Estimation

Using the original time series data, fit the following models to estimate both trend and seasonality:

Parametric Polynomial Regression

Non-parametric model

Overlay the fitted values on the original time series. Construct and plot the residuals with respect to time and ACF of residuals. Comment on how the two models fit and on the appropriateness of the stationarity assumption of the residuals. For sake of simplicity, only use Categorical Regression (ANOVA) seasonality modeling.

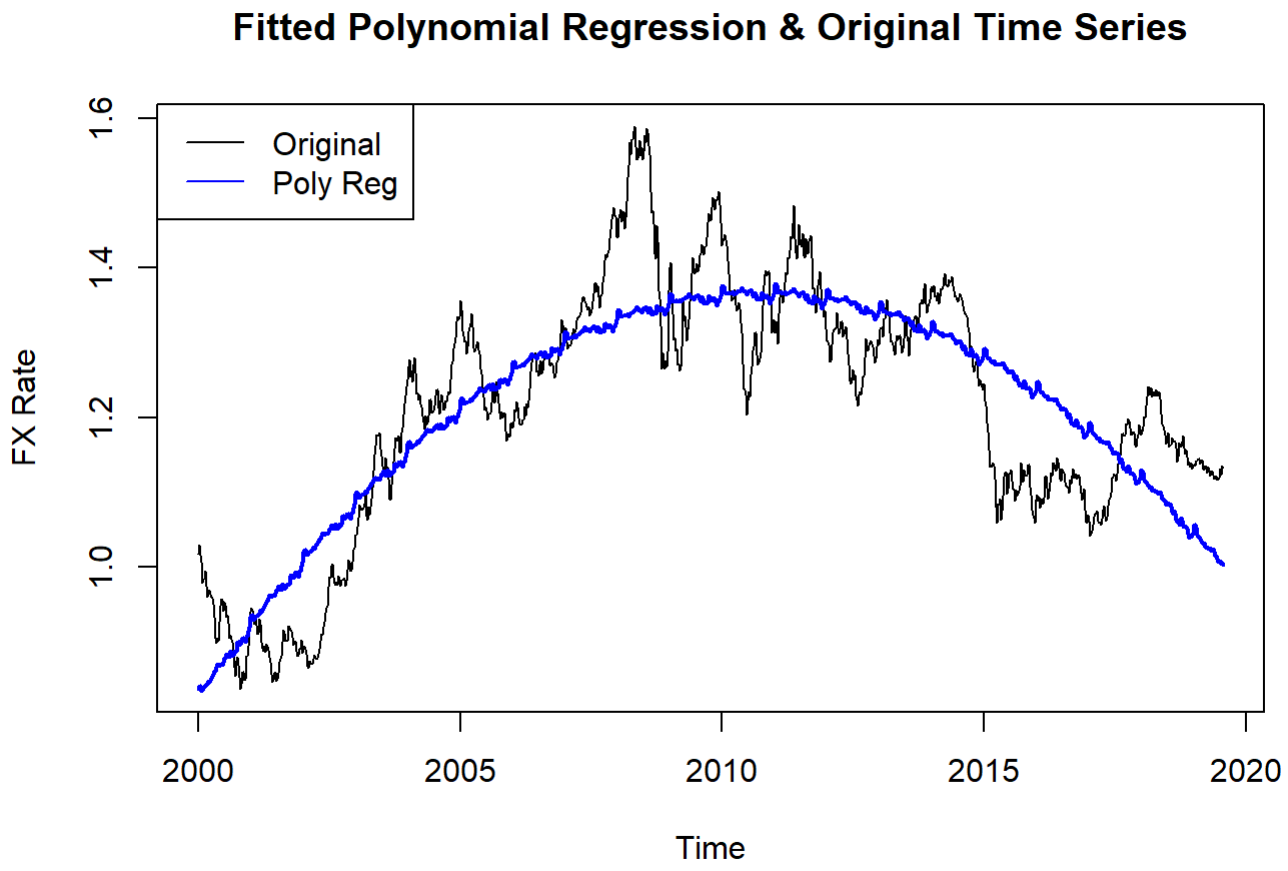
Parametric Polynomial Regression

```
## Seasonality & Trend: Parametric Model
## Fit a parametric model for both trend and seasonality

time.pts = c(1:length(rate))
time.pts = c(time.pts - min(time.pts))/ max(time.pts)

x1 = time.pts
x2 = time.pts^2
lm.fit = dynlm(rate ~ x1+x2+(season(rate)-1))
summary(lm.fit)
```

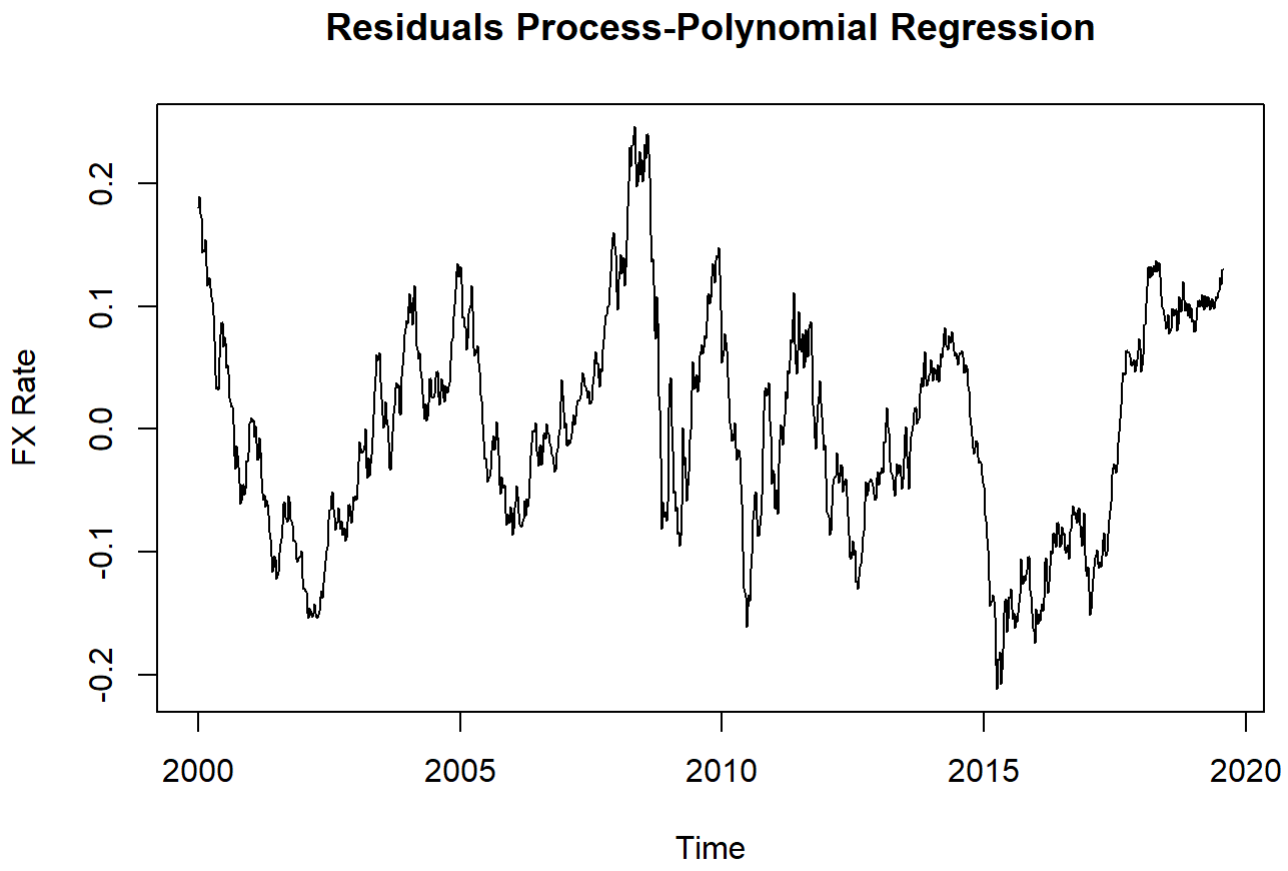
```
##
## Time series regression with "ts" data:
## Start = 2000(1), End = 2019(30)
##
## Call:
## dynlm(formula = rate ~ x1 + x2 + (season(rate) - 1))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.211931 -0.066917 -0.002785  0.062843  0.246424
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## x1              1.97618    0.03892   50.77   <2e-16 ***
## x2             -1.80313    0.03772  -47.80   <2e-16 ***
## season(rate)1   0.83689    0.02147   38.98   <2e-16 ***
## season(rate)2   0.83787    0.02147   39.02   <2e-16 ***
## season(rate)3   0.83608    0.02148   38.93   <2e-16 ***
## season(rate)4   0.82758    0.02148   38.53   <2e-16 ***
## season(rate)5   0.82667    0.02148   38.48   <2e-16 ***
## season(rate)6   0.82792    0.02148   38.54   <2e-16 ***
## season(rate)7   0.82689    0.02149   38.49   <2e-16 ***
## season(rate)8   0.82635    0.02149   38.46   <2e-16 ***
## season(rate)9   0.82569    0.02149   38.42   <2e-16 ***
## season(rate)10  0.82521    0.02149   38.40   <2e-16 ***
## season(rate)11  0.82792    0.02149   38.52   <2e-16 ***
## season(rate)12  0.82396    0.02150   38.33   <2e-16 ***
## season(rate)13  0.82572    0.02150   38.41   <2e-16 ***
## season(rate)14  0.82593    0.02150   38.41   <2e-16 ***
## season(rate)15  0.82734    0.02150   38.48   <2e-16 ***
## season(rate)16  0.82747    0.02151   38.48   <2e-16 ***
## season(rate)17  0.82960    0.02151   38.57   <2e-16 ***
## season(rate)18  0.82834    0.02151   38.51   <2e-16 ***
## season(rate)19  0.83099    0.02151   38.63   <2e-16 ***
## season(rate)20  0.83305    0.02151   38.72   <2e-16 ***
## season(rate)21  0.83079    0.02152   38.61   <2e-16 ***
## season(rate)22  0.82733    0.02152   38.45   <2e-16 ***
## season(rate)23  0.82855    0.02152   38.50   <2e-16 ***
## season(rate)24  0.82628    0.02152   38.39   <2e-16 ***
## season(rate)25  0.82348    0.02152   38.26   <2e-16 ***
## season(rate)26  0.82574    0.02153   38.36   <2e-16 ***
## season(rate)27  0.82844    0.02153   38.48   <2e-16 ***
## season(rate)28  0.82669    0.02153   38.40   <2e-16 ***
## season(rate)29  0.82962    0.02153   38.53   <2e-16 ***
## season(rate)30  0.82774    0.02153   38.44   <2e-16 ***
## season(rate)31  0.82232    0.02206   37.28   <2e-16 ***
## season(rate)32  0.82010    0.02206   37.17   <2e-16 ***
## season(rate)33  0.82631    0.02206   37.45   <2e-16 ***
## season(rate)34  0.82314    0.02206   37.30   <2e-16 ***
## season(rate)35  0.81636    0.02207   36.99   <2e-16 ***
## season(rate)36  0.81616    0.02207   36.98   <2e-16 ***
## season(rate)37  0.81919    0.02207   37.12   <2e-16 ***
## season(rate)38  0.81622    0.02207   36.98   <2e-16 ***
```

```
## season(rate)39  0.81632    0.02208   36.98   <2e-16 ***
## season(rate)40  0.82621    0.02208   37.42   <2e-16 ***
## season(rate)41  0.82494    0.02208   37.36   <2e-16 ***
## season(rate)42  0.81930    0.02208   37.10   <2e-16 ***
## season(rate)43  0.81860    0.02208   37.07   <2e-16 ***
## season(rate)44  0.82133    0.02209   37.19   <2e-16 ***
## season(rate)45  0.82209    0.02209   37.22   <2e-16 ***
## season(rate)46  0.82063    0.02209   37.15   <2e-16 ***
## season(rate)47  0.81605    0.02209   36.94   <2e-16 ***
## season(rate)48  0.81128    0.02209   36.72   <2e-16 ***
## season(rate)49  0.81691    0.02210   36.97   <2e-16 ***
## season(rate)50  0.81633    0.02210   36.94   <2e-16 ***
## season(rate)51  0.82071    0.02210   37.14   <2e-16 ***
## season(rate)52  0.82477    0.02210   37.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08957 on 964 degrees of freedom
## Multiple R-squared:  0.9949, Adjusted R-squared:  0.9946
## F-statistic:  3499 on 54 and 964 DF,  p-value: < 2.2e-16
```

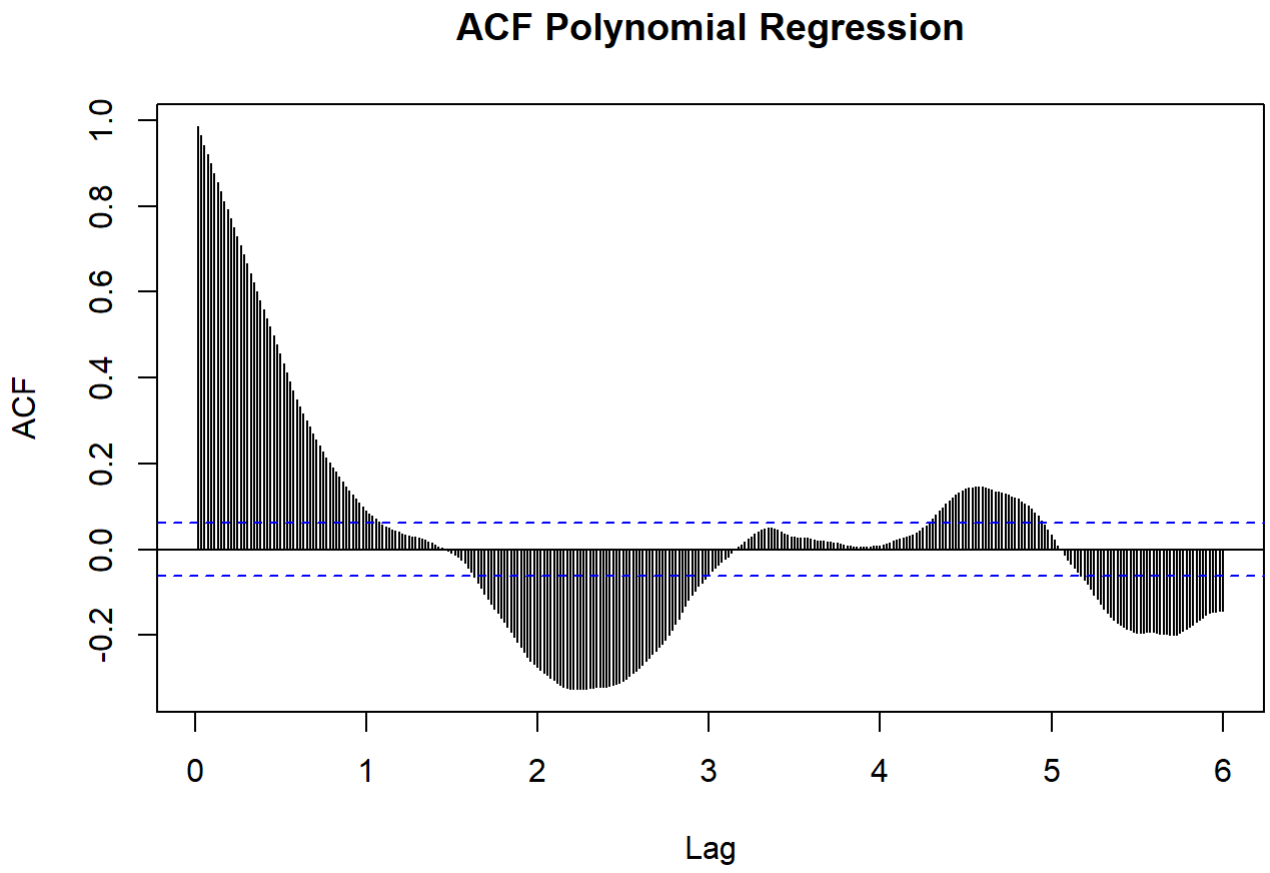
```
lm.fitted = ts(fitted(lm.fit),start=c(2000,1),freq=52)
res_lm = ts((rate-lm.fitted),start=c(2000,1),freq=52)

# Overlay the fitted values on the original time series
ts.plot(rate, ylab = "FX Rate", main = "Fitted Polynomial Regression & Original Time Series")
lines(lm.fitted, lwd=2, col="blue")
legend("topleft",legend=c("Original", "Poly Reg"),
       lty = 1,col=c("black", "blue"))
```




```
# Plot the residuals with respect to time and ACF of residuals
ts.plot(res_lm, ylab="FX Rate", main="Residuals Process-Polynomial Regression")
```




```
acf(res_lm, lag.max = 52*6, main="ACF Polynomial Regression")
```



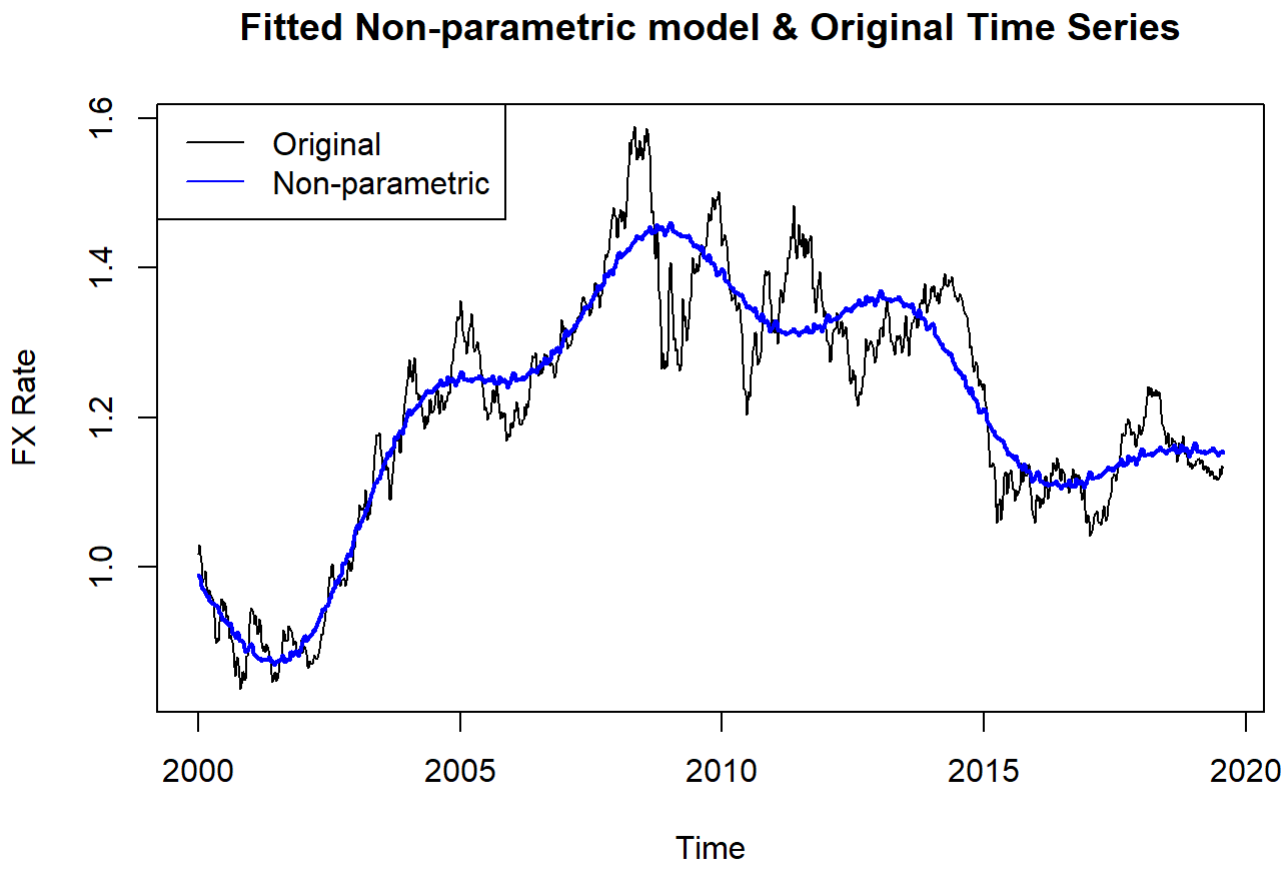
Non-parametric model

```
## Seasonality & Trend: Non-Parametric Model
## Fit a non parametric model for trend and linear model for seasonality

# Trend - Splines
# Seasonality - season/ cos-sin
#har2 = harmonic(rate,2)
#gam.fit = gam(rate ~ s(time.pts)+har2)
gam.fit = gam(rate ~ s(time.pts)+(season(rate)-1))
summary(gam.fit)
```

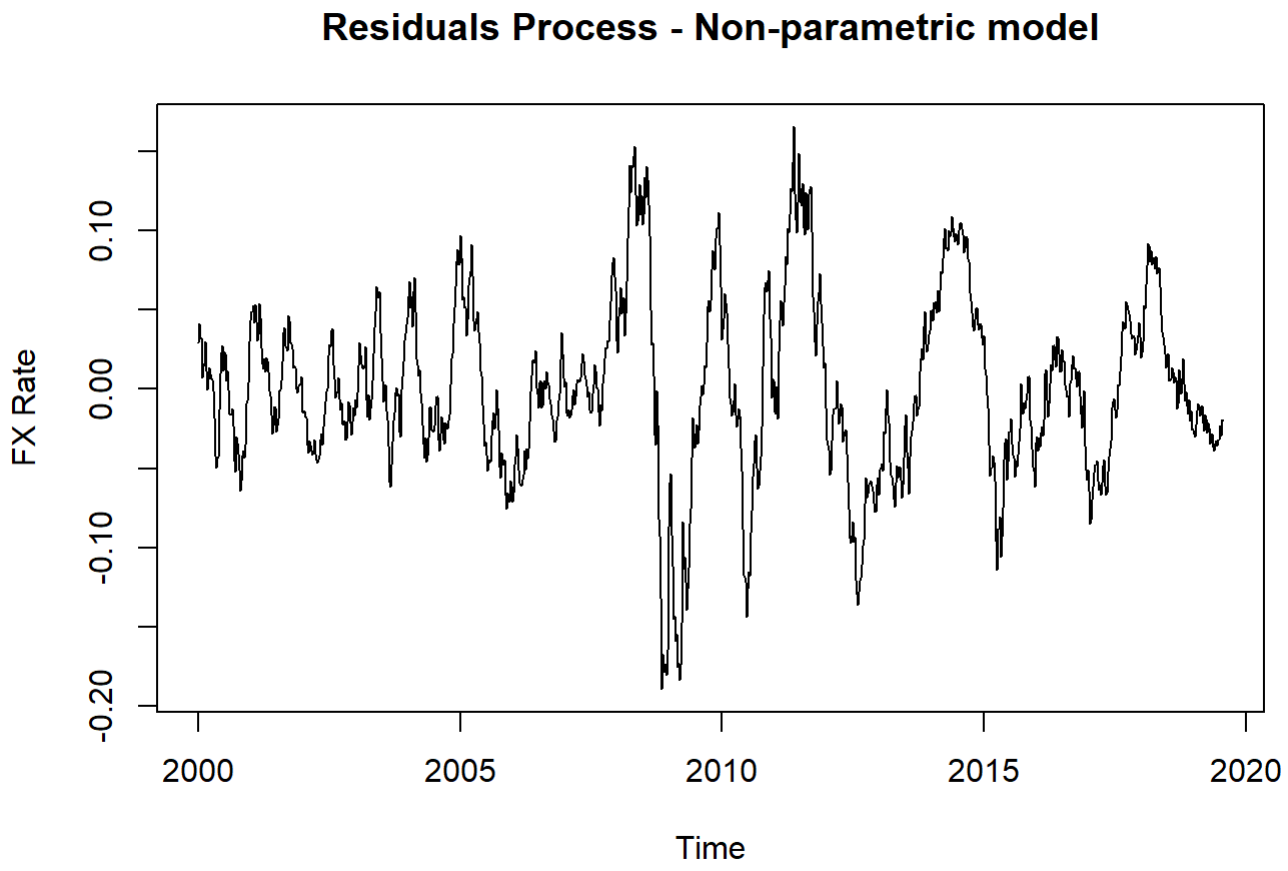
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## rate ~ s(time.pts) + (season(rate) - 1)
##
## Parametric coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## season(rate)Season-1   1.21983    0.01305   93.50   <2e-16 ***
## season(rate)Season-2   1.22091    0.01305   93.58   <2e-16 ***
## season(rate)Season-3   1.21923    0.01305   93.46   <2e-16 ***
## season(rate)Season-4   1.21081    0.01305   92.81   <2e-16 ***
## season(rate)Season-5   1.21000    0.01305   92.75   <2e-16 ***
## season(rate)Season-6   1.21133    0.01304   92.86   <2e-16 ***
## season(rate)Season-7   1.21038    0.01304   92.79   <2e-16 ***
## season(rate)Season-8   1.20991    0.01304   92.75   <2e-16 ***
## season(rate)Season-9   1.20932    0.01304   92.71   <2e-16 ***
## season(rate)Season-10  1.20890    0.01304   92.68   <2e-16 ***
## season(rate)Season-11  1.21166    0.01304   92.89   <2e-16 ***
## season(rate)Season-12  1.20776    0.01304   92.59   <2e-16 ***
## season(rate)Season-13  1.20956    0.01304   92.73   <2e-16 ***
## season(rate)Season-14  1.20981    0.01304   92.75   <2e-16 ***
## season(rate)Season-15  1.21126    0.01304   92.86   <2e-16 ***
## season(rate)Season-16  1.21141    0.01304   92.87   <2e-16 ***
## season(rate)Season-17  1.21357    0.01304   93.04   <2e-16 ***
## season(rate)Season-18  1.21233    0.01304   92.94   <2e-16 ***
## season(rate)Season-19  1.21499    0.01304   93.15   <2e-16 ***
## season(rate)Season-20  1.21706    0.01304   93.31   <2e-16 ***
## season(rate)Season-21  1.21481    0.01304   93.13   <2e-16 ***
## season(rate)Season-22  1.21135    0.01304   92.86   <2e-16 ***
## season(rate)Season-23  1.21256    0.01304   92.96   <2e-16 ***
## season(rate)Season-24  1.21028    0.01304   92.78   <2e-16 ***
## season(rate)Season-25  1.20747    0.01304   92.56   <2e-16 ***
## season(rate)Season-26  1.20970    0.01305   92.73   <2e-16 ***
## season(rate)Season-27  1.21237    0.01305   92.94   <2e-16 ***
## season(rate)Season-28  1.21059    0.01305   92.80   <2e-16 ***
## season(rate)Season-29  1.21349    0.01305   93.02   <2e-16 ***
## season(rate)Season-30  1.21156    0.01305   92.86   <2e-16 ***
## season(rate)Season-31  1.21407    0.01338   90.71   <2e-16 ***
## season(rate)Season-32  1.21187    0.01338   90.55   <2e-16 ***
## season(rate)Season-33  1.21810    0.01338   91.01   <2e-16 ***
## season(rate)Season-34  1.21494    0.01338   90.78   <2e-16 ***
## season(rate)Season-35  1.20816    0.01338   90.27   <2e-16 ***
## season(rate)Season-36  1.20796    0.01338   90.26   <2e-16 ***
## season(rate)Season-37  1.21098    0.01338   90.48   <2e-16 ***
## season(rate)Season-38  1.20800    0.01338   90.26   <2e-16 ***
## season(rate)Season-39  1.20808    0.01338   90.27   <2e-16 ***
## season(rate)Season-40  1.21795    0.01338   91.00   <2e-16 ***
## season(rate)Season-41  1.21665    0.01338   90.91   <2e-16 ***
## season(rate)Season-42  1.21098    0.01338   90.48   <2e-16 ***
## season(rate)Season-43  1.21024    0.01338   90.43   <2e-16 ***
## season(rate)Season-44  1.21292    0.01338   90.63   <2e-16 ***
```

```
## season(rate)Season-45  1.21363    0.01338   90.68   <2e-16 ***
## season(rate)Season-46  1.21211    0.01338   90.57   <2e-16 ***
## season(rate)Season-47  1.20747    0.01338   90.22   <2e-16 ***
## season(rate)Season-48  1.20264    0.01338   89.86   <2e-16 ***
## season(rate)Season-49  1.20820    0.01338   90.27   <2e-16 ***
## season(rate)Season-50  1.20754    0.01338   90.22   <2e-16 ***
## season(rate)Season-51  1.21184    0.01338   90.54   <2e-16 ***
## season(rate)Season-52  1.21581    0.01338   90.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(time.pts) 8.985      9 829.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.879   Deviance explained = 99.8%
## GCV = 0.0036191  Scale est. = 0.0034023  n = 1018
```

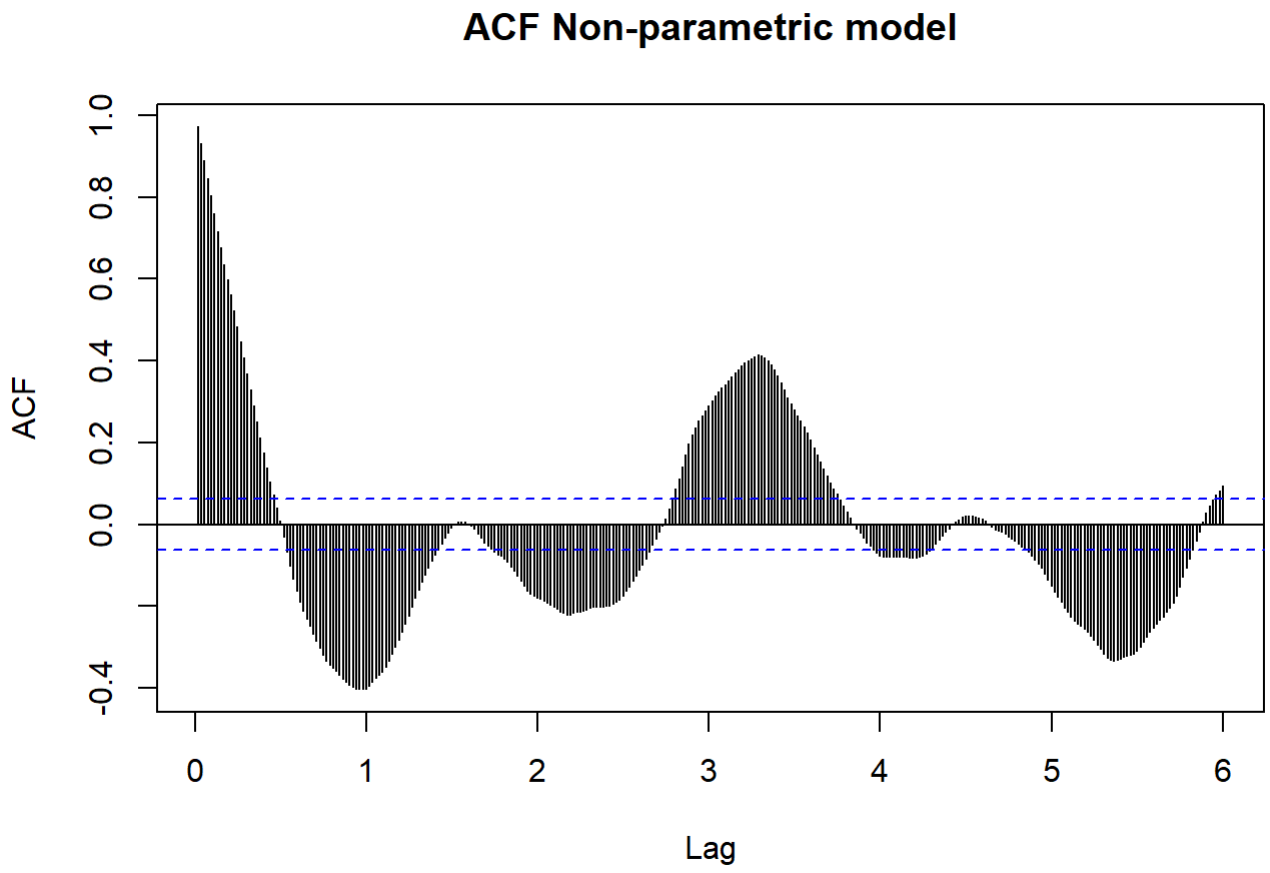
```
gam.fitted = ts(fitted(gam.fit), start=c(2000,1), freq=52)
res_gam = ts((rate-gam.fitted), start=c(2000,1), frequency=52)

# Overlay the fitted values on the original time series
ts.plot(rate, ylab = "FX Rate", main = "Fitted Non-parametric model & Original Time Series")
lines(gam.fitted, lwd=2, col="blue")
legend("topleft",legend=c("Original", "Non-parametric"),
       lty = 1,col=c("black", "blue"))
```




```
# Plot the residuals with respect to time and ACF of residuals
ts.plot(res_gam, ylab="FX Rate", main="Residuals Process - Non-parametric model")
```




```
acf(res_gam, lag.max = 52*6, main="ACF Non-parametric model")
```



Answer

All the coefficients of both the parametric and non-parametric models are statistically significant. We can see in the parametric Polynomial Regression that the model captures the overall trend and the cyclical pattern is incorporated in the line being wiggly. However, the trend captured by the (non-parametric) splines model is more accurate. In addition, the R^2 values suggest that the non-parametric model fits the process (slightly) better with R^2 equals to 99.8% (Deviance explained = 99.8%) while the parametric's R^2 is ~ 99.5%.

Looking at the "Residuals Process - Non-parametric model" plot, we see the residuals process' mean is centered around 0 which is expected with a stationary process, but the variability is not constant. The "Residuals Process - Polynomial Regression" plot shows non constant mean or variance. The ACF plots of both models indicate that the residuals processes are non-stationary and that the cyclical pattern has not been removed.

Question 2c: Trend-Seasonality Estimation with Differenced Data

Now using the differenced time series data, construct the same type of models as you did above. Overlay the fitted values on the original time series.Construct and plot the residuals with respect to time and ACF of residuals. Comment on the two models fit and on the appropriateness of the stationarity assumption of the residuals. Additionally, comment if models built with original or differenced data appear to differ in quality of fit; which (if any) is better?

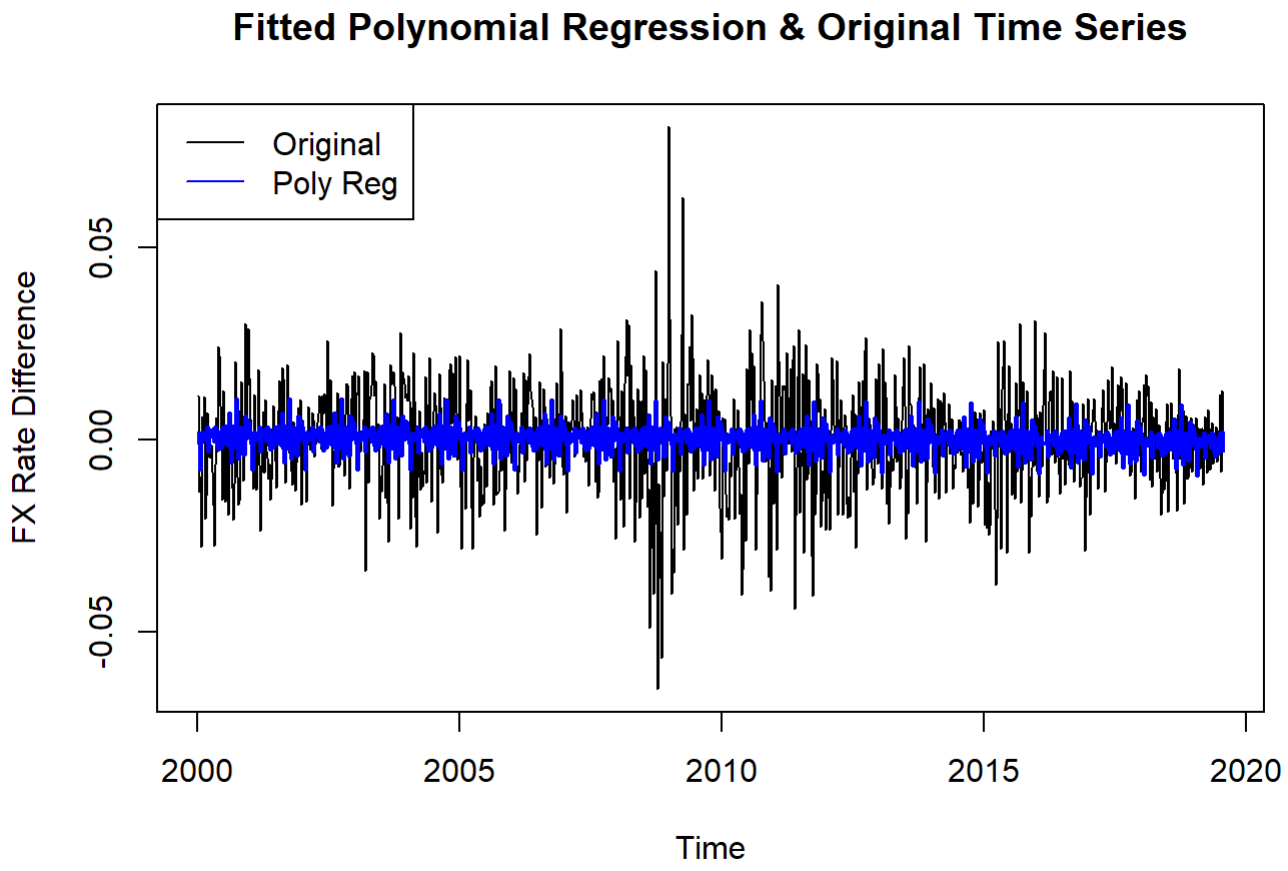
Parametric Polynomial Regression

```
## Seasonality & Trend: Parametric Model
## Fit a parametric model for both trend and seasonality

time.pts = c(1:length(rate.dif))
time.pts = c(time.pts - min(time.pts))/ max(time.pts)

x1 = time.pts
x2 = time.pts^2
lm.fit = dynlm(rate.dif ~ x1+x2+(season(rate.dif)-1))
summary(lm.fit)
```

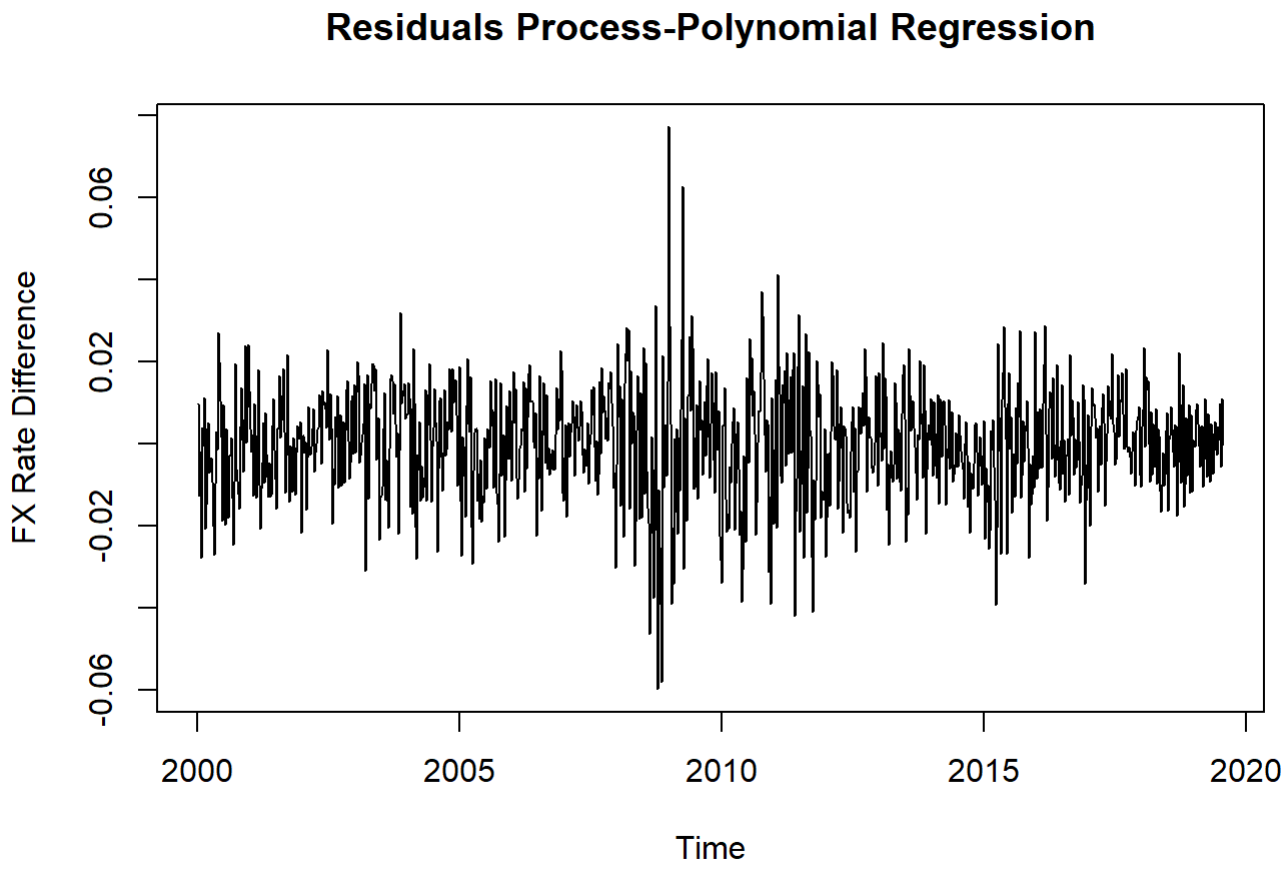
```
##
## Time series regression with "ts" data:
## Start = 2000(2), End = 2019(30)
##
## Call:
## dynlm(formula = rate.dif ~ x1 + x2 + (season(rate.dif) - 1))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.059778 -0.008413  0.001074  0.008683  0.077160
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## x1                  -1.117e-04  5.999e-03  -0.019  0.98514
## x2                  -1.550e-03  5.814e-03  -0.267  0.78983
## season(rate.dif)1    3.307e-03  3.405e-03   0.971  0.33173
## season(rate.dif)2    1.756e-03  3.308e-03   0.531  0.59558
## season(rate.dif)3   -1.010e-03  3.308e-03  -0.305  0.76022
## season(rate.dif)4   -7.737e-03  3.309e-03  -2.338  0.01957 *
## season(rate.dif)5   -1.319e-04  3.309e-03  -0.040  0.96821
## season(rate.dif)6    2.013e-03  3.309e-03   0.608  0.54311
## season(rate.dif)7   -2.622e-04  3.310e-03  -0.079  0.93687
## season(rate.dif)8    2.249e-04  3.310e-03   0.068  0.94585
## season(rate.dif)9    1.010e-04  3.310e-03   0.031  0.97567
## season(rate.dif)10   2.796e-04  3.311e-03   0.084  0.93271
## season(rate.dif)11   3.464e-03  3.311e-03   1.046  0.29572
## season(rate.dif)12  -3.195e-03  3.311e-03  -0.965  0.33482
## season(rate.dif)13   2.510e-03  3.312e-03   0.758  0.44861
## season(rate.dif)14   9.591e-04  3.312e-03   0.290  0.77221
## season(rate.dif)15   2.165e-03  3.312e-03   0.654  0.51359
## season(rate.dif)16   8.753e-04  3.313e-03   0.264  0.79166
## season(rate.dif)17   2.878e-03  3.313e-03   0.869  0.38525
## season(rate.dif)18  -5.124e-04  3.313e-03  -0.155  0.87714
## season(rate.dif)19   3.390e-03  3.314e-03   1.023  0.30653
## season(rate.dif)20   2.803e-03  3.314e-03   0.846  0.39791
## season(rate.dif)21  -1.518e-03  3.314e-03  -0.458  0.64696
## season(rate.dif)22  -2.720e-03  3.315e-03  -0.821  0.41204
## season(rate.dif)23   1.951e-03  3.315e-03   0.589  0.55625
## season(rate.dif)24  -1.537e-03  3.315e-03  -0.464  0.64304
## season(rate.dif)25  -2.058e-03  3.316e-03  -0.621  0.53504
## season(rate.dif)26   2.982e-03  3.316e-03   0.899  0.36872
## season(rate.dif)27   3.431e-03  3.316e-03   1.035  0.30113
## season(rate.dif)28  -1.021e-03  3.317e-03  -0.308  0.75819
## season(rate.dif)29   3.659e-03  3.317e-03   1.103  0.27030
## season(rate.dif)30  -1.162e-03  3.317e-03  -0.350  0.72625
## season(rate.dif)31   2.189e-03  3.398e-03   0.644  0.51959
## season(rate.dif)32  -1.467e-03  3.398e-03  -0.432  0.66613
## season(rate.dif)33   6.966e-03  3.399e-03   2.050  0.04066 *
## season(rate.dif)34  -2.419e-03  3.399e-03  -0.712  0.47678
## season(rate.dif)35  -6.033e-03  3.399e-03  -1.775  0.07623 .
## season(rate.dif)36   5.476e-04  3.400e-03   0.161  0.87207
## season(rate.dif)37   3.774e-03  3.400e-03   1.110  0.26722
## season(rate.dif)38  -2.222e-03  3.400e-03  -0.654  0.51356
```

```
## season(rate.dif)39  8.406e-04  3.401e-03   0.247  0.80482
## season(rate.dif)40  1.062e-02  3.401e-03   3.124  0.00184 **
## season(rate.dif)41 -5.282e-04  3.401e-03  -0.155  0.87662
## season(rate.dif)42 -4.901e-03  3.402e-03  -1.441  0.15002
## season(rate.dif)43  3.292e-05  3.402e-03   0.010  0.99228
## season(rate.dif)44  3.455e-03  3.402e-03   1.015  0.31020
## season(rate.dif)45  1.498e-03  3.403e-03   0.440  0.65991
## season(rate.dif)46 -7.374e-04  3.403e-03  -0.217  0.82849
## season(rate.dif)47 -3.849e-03  3.403e-03  -1.131  0.25833
## season(rate.dif)48 -4.047e-03  3.404e-03  -1.189  0.23469
## season(rate.dif)49  6.356e-03  3.404e-03   1.867  0.06215 .
## season(rate.dif)50  1.400e-04  3.404e-03   0.041  0.96721
## season(rate.dif)51  5.100e-03  3.405e-03   1.498  0.13446
## season(rate.dif)52  4.776e-03  3.405e-03   1.403  0.16101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0138 on 963 degrees of freedom
## Multiple R-squared:  0.05732,    Adjusted R-squared:  0.004456
## F-statistic: 1.084 on 54 and 963 DF,  p-value: 0.3182
```

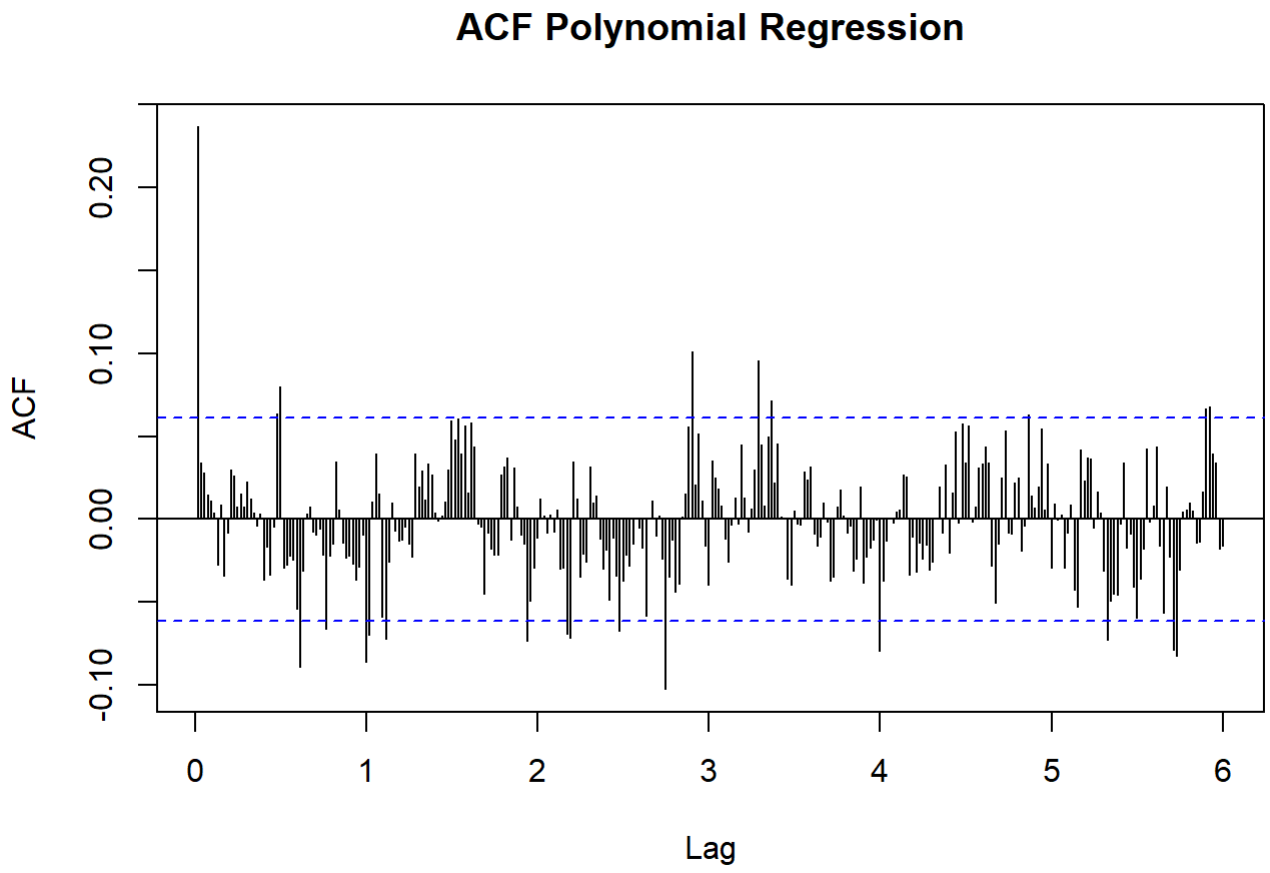
```
lm.fitted = ts(fitted(lm.fit),start=c(2000,2),freq=52)
res_lm = ts((rate.dif-lm.fitted),start=c(2000,2),freq=52)

# Overlay the fitted values on the original time series
ts.plot(rate.dif, ylab = "FX Rate Difference", main = "Fitted Polynomial Regression & Original T
ime Series")
lines(lm.fitted, lwd=2, col="blue")
legend("topleft",legend=c("Original", "Poly Reg"),
       lty = 1,col=c("black", "blue"))
```




```
# Plot the residuals with respect to time and ACF of residuals
ts.plot(res_lm, ylab="FX Rate Difference", main="Residuals Process-Polynomial Regression")
```




```
acf(res_lm, lag.max = 52*6, main="ACF Polynomial Regression")
```



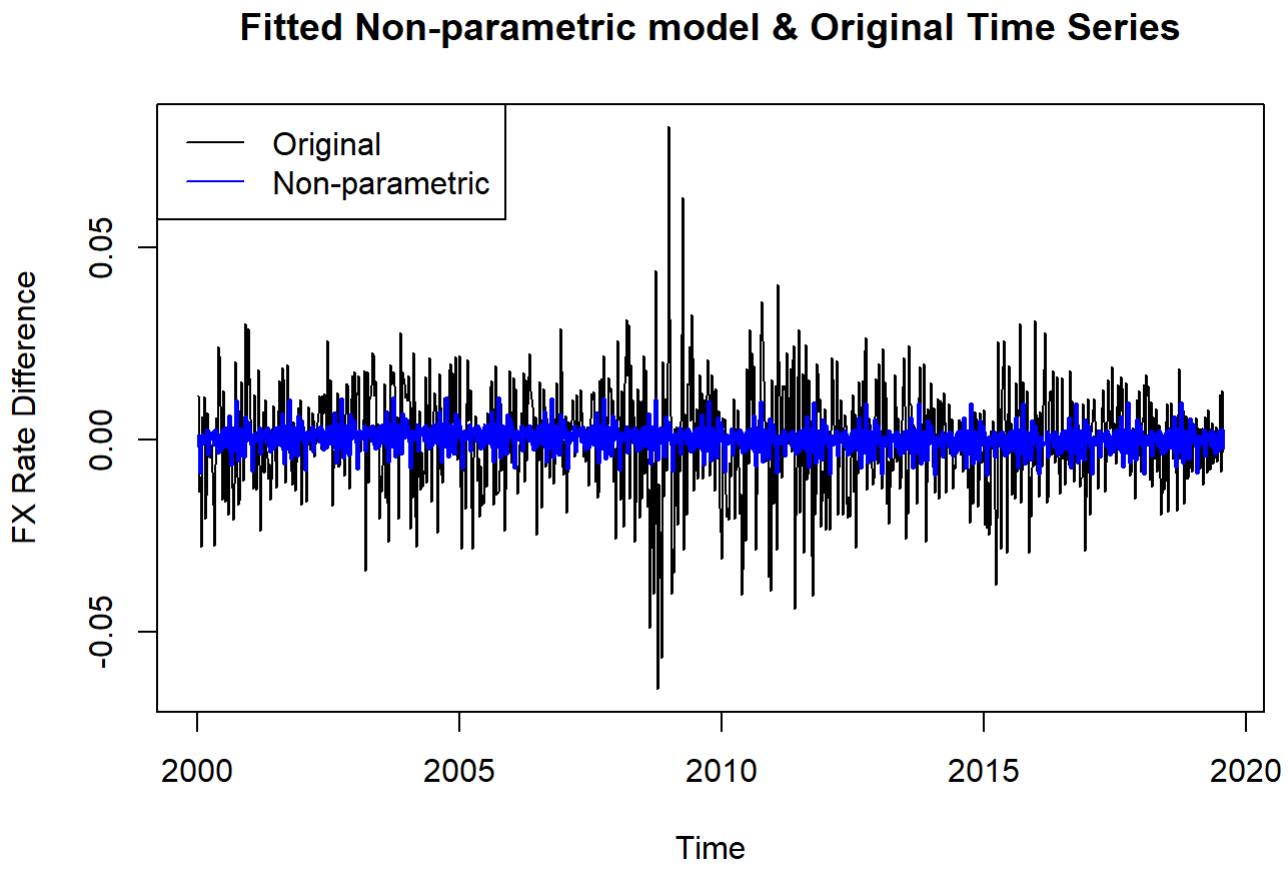
Non-parametric model

```
## Seasonality & Trend: Non-Parametric Model
## Fit a non parametric model for trend and linear model for seasonality

# Trend - Splines
# Seasonality - season/ cos-sin
#har2 = harmonic(rate.dif,2)
#gam.fit = gam(rate.dif ~ s(time.pts)+har2)
gam.fit = gam(rate.dif ~ s(time.pts)+(season(rate.dif)-1))
summary(gam.fit)
```

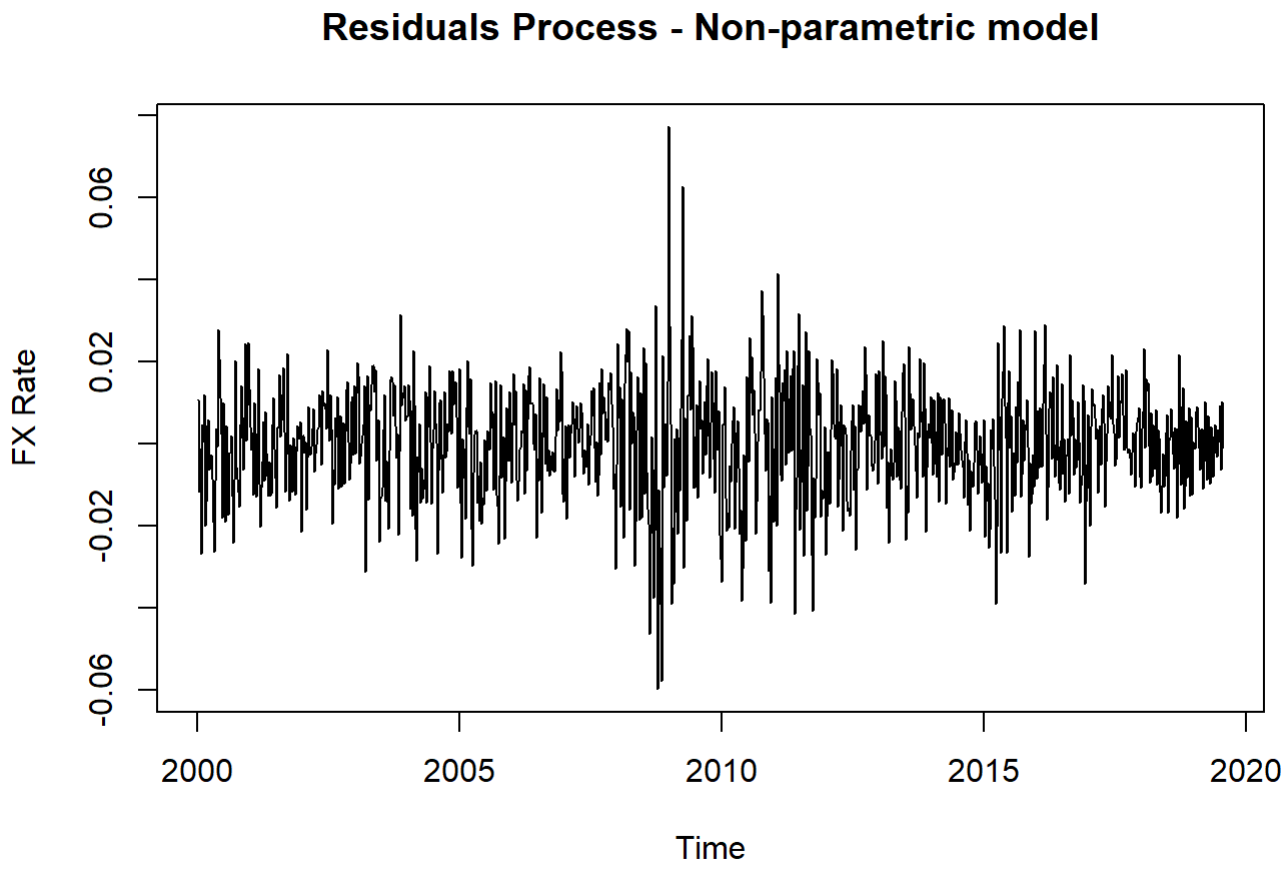
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## rate.dif ~ s(time.pts) + (season(rate.dif) - 1)
##
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## season(rate.dif)Season-1   2.717e-03  3.162e-03   0.859  0.39039
## season(rate.dif)Season-2   1.210e-03  3.082e-03   0.393  0.69473
## season(rate.dif)Season-3  -1.558e-03  3.082e-03  -0.505  0.61333
## season(rate.dif)Season-4  -8.286e-03  3.082e-03  -2.688  0.00730 **
## season(rate.dif)Season-5  -6.835e-04  3.082e-03  -0.222  0.82456
## season(rate.dif)Season-6   1.460e-03  3.082e-03   0.474  0.63585
## season(rate.dif)Season-7  -8.172e-04  3.082e-03  -0.265  0.79096
## season(rate.dif)Season-8  -3.318e-04  3.082e-03  -0.108  0.91429
## season(rate.dif)Season-9  -4.574e-04  3.082e-03  -0.148  0.88205
## season(rate.dif)Season-10 -2.805e-04  3.082e-03  -0.091  0.92750
## season(rate.dif)Season-11  2.902e-03  3.082e-03   0.942  0.34660
## season(rate.dif)Season-12 -3.759e-03  3.082e-03  -1.220  0.22293
## season(rate.dif)Season-13  1.945e-03  3.082e-03   0.631  0.52810
## season(rate.dif)Season-14  3.921e-04  3.082e-03   0.127  0.89878
## season(rate.dif)Season-15  1.596e-03  3.082e-03   0.518  0.60468
## season(rate.dif)Season-16  3.050e-04  3.082e-03   0.099  0.92119
## season(rate.dif)Season-17  2.306e-03  3.082e-03   0.748  0.45454
## season(rate.dif)Season-18 -1.086e-03  3.082e-03  -0.352  0.72463
## season(rate.dif)Season-19  2.815e-03  3.082e-03   0.913  0.36132
## season(rate.dif)Season-20  2.226e-03  3.082e-03   0.722  0.47036
## season(rate.dif)Season-21 -2.097e-03  3.082e-03  -0.680  0.49639
## season(rate.dif)Season-22 -3.301e-03  3.082e-03  -1.071  0.28447
## season(rate.dif)Season-23  1.369e-03  3.082e-03   0.444  0.65696
## season(rate.dif)Season-24 -2.121e-03  3.082e-03  -0.688  0.49157
## season(rate.dif)Season-25 -2.643e-03  3.082e-03  -0.858  0.39137
## season(rate.dif)Season-26  2.395e-03  3.082e-03   0.777  0.43732
## season(rate.dif)Season-27  2.842e-03  3.082e-03   0.922  0.35668
## season(rate.dif)Season-28 -1.612e-03  3.082e-03  -0.523  0.60115
## season(rate.dif)Season-29  3.067e-03  3.082e-03   0.995  0.32003
## season(rate.dif)Season-30 -1.755e-03  3.082e-03  -0.570  0.56912
## season(rate.dif)Season-31  1.631e-03  3.162e-03   0.516  0.60612
## season(rate.dif)Season-32 -2.026e-03  3.162e-03  -0.641  0.52185
## season(rate.dif)Season-33  6.406e-03  3.162e-03   2.026  0.04308 *
## season(rate.dif)Season-34 -2.982e-03  3.162e-03  -0.943  0.34597
## season(rate.dif)Season-35 -6.597e-03  3.162e-03  -2.086  0.03722 *
## season(rate.dif)Season-36 -1.767e-05  3.162e-03  -0.006  0.99554
## season(rate.dif)Season-37  3.208e-03  3.162e-03   1.014  0.31065
## season(rate.dif)Season-38 -2.791e-03  3.162e-03  -0.882  0.37776
## season(rate.dif)Season-39  2.710e-04  3.162e-03   0.086  0.93172
## season(rate.dif)Season-40  1.005e-02  3.162e-03   3.179  0.00152 **
## season(rate.dif)Season-41 -1.101e-03  3.162e-03  -0.348  0.72785
## season(rate.dif)Season-42 -5.474e-03  3.162e-03  -1.731  0.08373 .
## season(rate.dif)Season-43 -5.424e-04  3.162e-03  -0.172  0.86384
## season(rate.dif)Season-44  2.878e-03  3.162e-03   0.910  0.36303
```

```
## season(rate.dif)Season-45  9.196e-04  3.162e-03   0.291  0.77127
## season(rate.dif)Season-46 -1.317e-03  3.162e-03  -0.416  0.67714
## season(rate.dif)Season-47 -4.430e-03  3.162e-03  -1.401  0.16154
## season(rate.dif)Season-48 -4.630e-03  3.162e-03  -1.464  0.14350
## season(rate.dif)Season-49  5.773e-03  3.162e-03   1.825  0.06824 .
## season(rate.dif)Season-50 -4.453e-04  3.162e-03  -0.141  0.88805
## season(rate.dif)Season-51  4.513e-03  3.162e-03   1.427  0.15383
## season(rate.dif)Season-52  4.188e-03  3.162e-03   1.324  0.18568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(time.pts) 2.706  3.369 0.732   0.485
##
## R-sq.(adj) =  0.00758   Deviance explained = 6.01%
## GCV = 0.00020078  Scale est. = 0.00018998  n = 1017
```

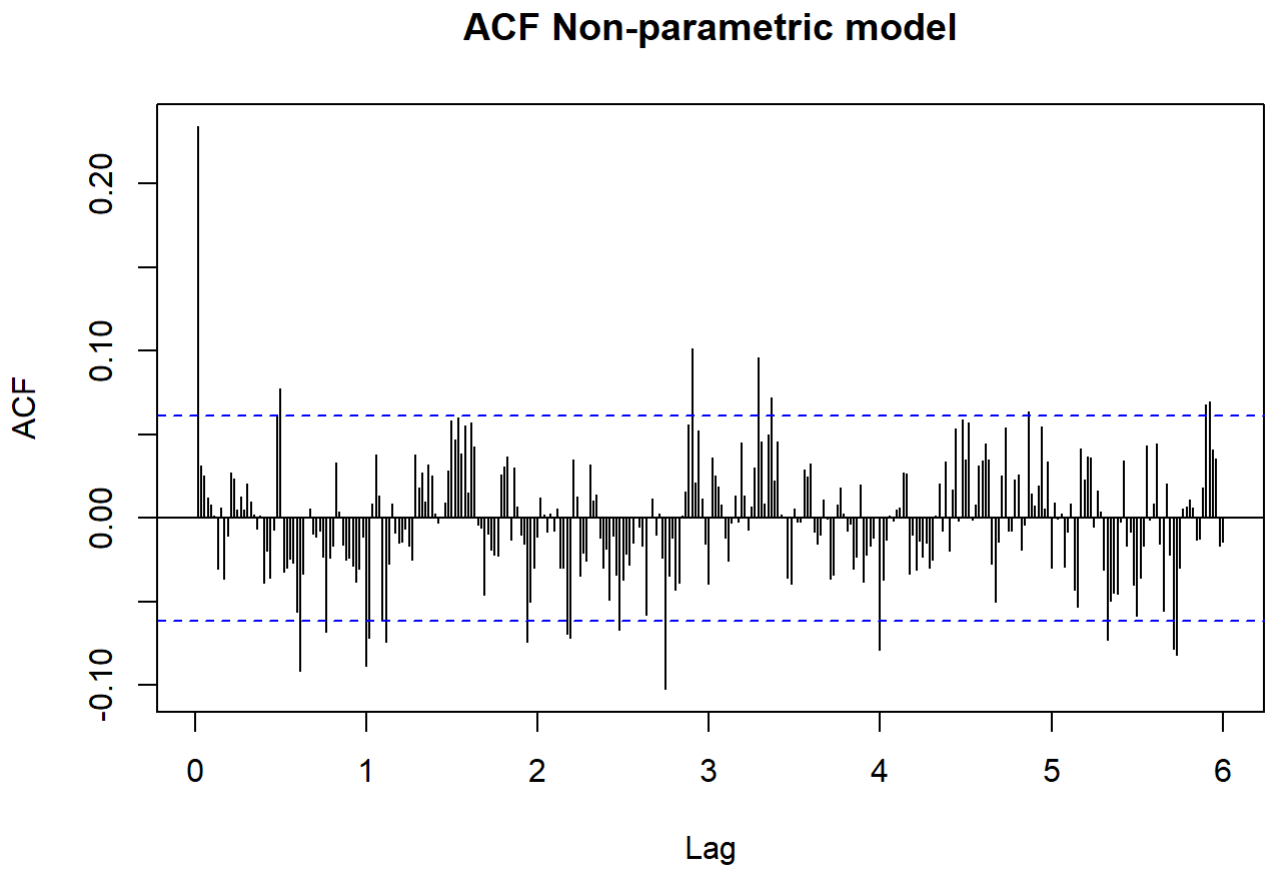
```
gam.fitted = ts(fitted(gam.fit), start=c(2000,2), freq=52)
res_gam = ts((rate.dif-gam.fitted), start=c(2000,2), frequency=52)

# Overlay the fitted values on the original time series
ts.plot(rate.dif, ylab = "FX Rate Difference", main = "Fitted Non-parametric model & Original Ti
me Series")
lines(gam.fitted, lwd=2, col="blue")
legend("topleft",legend=c("Original", "Non-parametric"),
       lty = 1,col=c("black", "blue"))
```




```
# Plot the residuals with respect to time and ACF of residuals
ts.plot(res_gam, ylab="FX Rate", main="Residuals Process - Non-parametric model")
```




```
acf(res_gam, lag.max = 52*6, main="ACF Non-parametric model")
```



Answer

Almost all of the two models' coefficients are statistically insignificant and both R^2 are extremely low at approximately 6%, indicating the models do not account for the variability presented in the process. With that being said, we can see in the residuals plots that the residuals processes' mean is around 0 and the variability is somewhat constant, excluding of course the few outliers around the year 2009. Supported by the ACF plots, we conclude that the residuals processes of both models are stationary with some random cyclical pattern.

With regards to quality of fit, models built with the original data yielded statistically significant coefficients and extremely high R^2s while models built with the differenced data yielded statistically insignificant coefficients and extremely low R^2s. However, removing trend and seasonality from the original data did not produce stationary process while removing trend and seasonality from the differenced data did result in a stationary process. These results make sense because we basically already removed both trend and seasonality when we differenced the original data. In other words, by differencing the data we basically removing the trend and seasonality components so it make sense that the trend and seasonality models will have statistically insignificant coefficients and low R^2, because there are no trend and seasonality components. And since the residuals of the differenced data gave a stationary process and the residuals of the original data did not, we conclude that working with the differenced data is better.