ISYE 6402 Homework 2 Solutions: Question 1

# Background

In this problem, we will study fluctuations in currency exchange rate over time.

File EURUSDCurrency.csv download contains the daily exchange rate of USD/EUR from January 1999 through December 31st 2020. We will aggregate the data on a weekly basis, by taking the average rate within each week. We will also analyze the first order difference of the weekly currency exchange.

# Instructions on reading the data

To read the data in R, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the R function read.csv()

library(readr)  
df <- read\_csv("~/Desktop/EURUSDCurrency.csv")

Here we upload the libraries needed the this data analysis:

library(mgcv)  
library(lubridate)  
library(dplyr)  
library(stats)  
library(tseries)  
library(graphics)  
library(dynlm)

To prepare the data, run the following code snippet. First, aggregate by week:

df$date <- as.Date(df$Date, format='%m/%d/%Y')  
df$week <- floor\_date(df$date, "week")  
df$eur <- df$EUR  
  
df <- df[, c("week", "eur")]

We now form the weekly aggrgated time series to use for data exploration! Please note that we will analyze the weekly aggregated data not the original (daily) data.

agg <- aggregate(x = df$eur, by = list(df$week), FUN = mean)  
colnames(agg) <- c("week", "eur")  
  
price <- ts(agg$eur, start = 1999, freq = 52)

#### Please use the price series to code and answer the following questions.

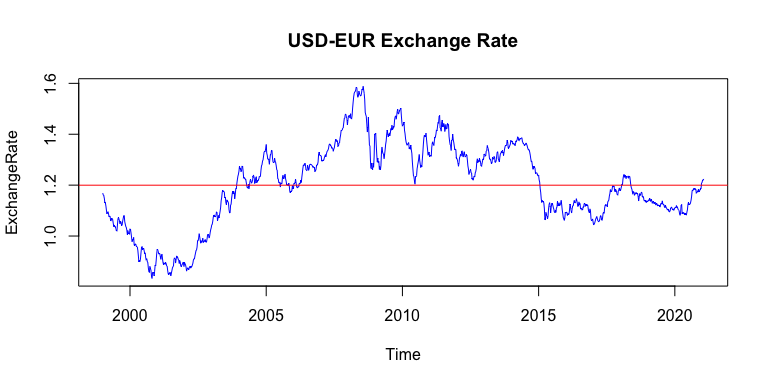
# Question 1a: Exploratory Data Analysis

Before exploring the data, can you infer the data features from what you know about the USD-EUR currency exchange? Next plot the Time Series and ACF plots of the weekly data. Comment on the main features, and identify what (if any) assumptions of stationarity are violated.

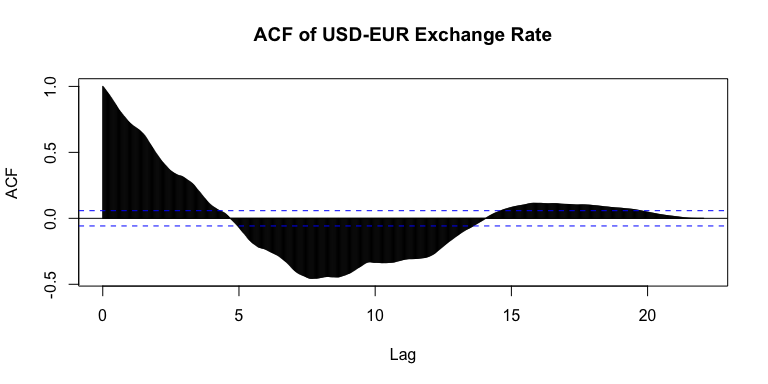
Which type of model do you think will fit the data better: the trend or seasonality fitting model? Provide details for your response.

*Response: General Insights on the USD-EUR Currency Rate*

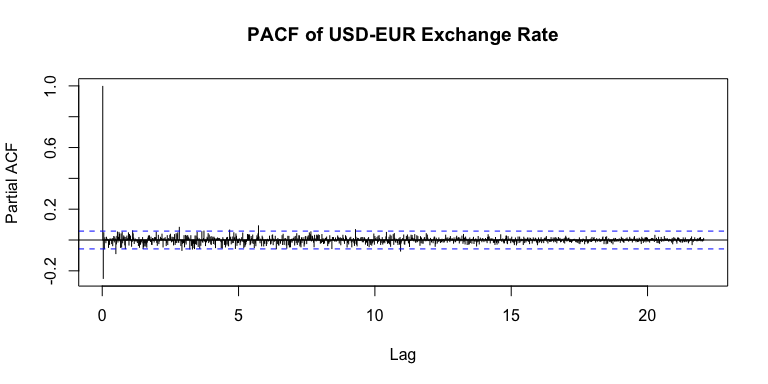
ts.plot(price, main="USD-EUR Exchange Rate", ylab="ExchangeRate", col="blue")  
abline(a=mean(price),b=0,col="red")



acf(price,main="ACF of USD-EUR Exchange Rate",lag=3650)



pacf(price,main="PACF of USD-EUR Exchange Rate",lag=3650)



*Response: General Insights from the Graphical Analysis*

By observing the USD-EUR exchange plot, I find a cyclic pattern but it does not show strong seasonality or trend pattern since it the pattern does not repeat at the exact period of time, and the variance changes over time. By plotting the ACF, it shows that the data autocorrelations are strong and decay slowly but increase slowly later, therefore it is less likely the model is stationary.

# Question 1b: Trend Estimation

Next we fit the following trend estimation models:

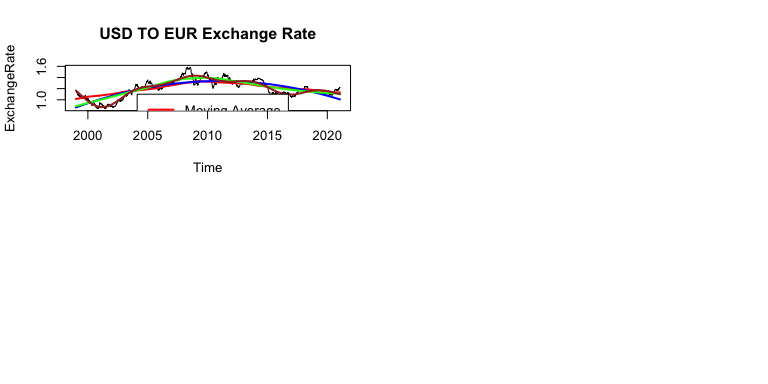
* Moving Average
* Parametric Quadratic Polynomial
* Local Polynomial
* Splines Smoothing

Overlay the fitted values on the time series of interst with the fitted trend models. Plot the residuals with respect to time for each model. Plot the ACF of the residuals for each model also. Comment on the four models fit and on the appropriateness of the stationarity assumption of the residuals.

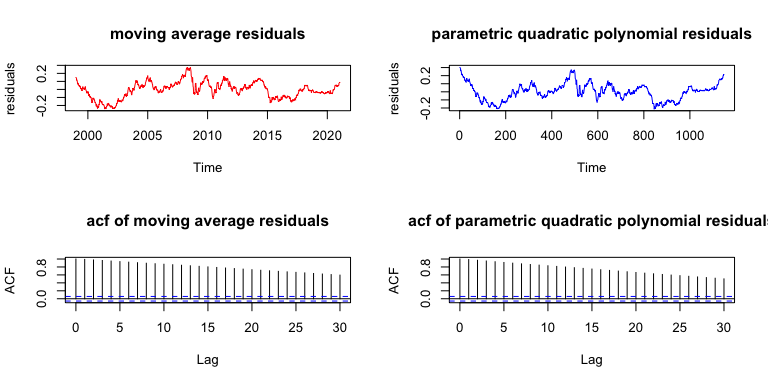
#moving average  
time.pts = c(1:length(price))  
time.pts=c(time.pts-min(time.pts))/max(time.pts)  
mav.fit=ksmooth(time.pts, price, kernel="box")  
mav.price.fit=ts(mav.fit$y, start=c(1999,1),frequency=52)  
#parametric quadratic polynomial  
x1=time.pts  
x2=time.pts^2  
lm.fit = lm(price~x1+x2)  
lm.price.fit=ts(fitted(lm.fit),start=c(1999,1),frequency = 52)  
#local polynomial  
loc.fit=loess(price~time.pts)  
loc.price.fit=ts(fitted(loc.fit),start=c(1999,1),frequency = 52)  
#Splines  
library(mgcv)  
gam.fit=gam(price~s(time.pts))  
gam.price.fit=ts(fitted(gam.fit),start=c(1999,1),frequency=52)  
summary(gam.fit)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## price ~ s(time.pts)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.199777 0.001692 709.3 <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) 8.985 9 899 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.876 Deviance explained = 87.7%  
## GCV = 0.0033138 Scale est. = 0.003285 n = 1148

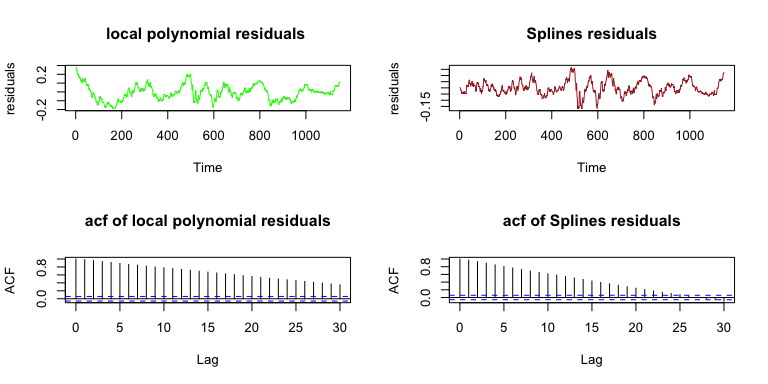
#overlay the data  
par(mfrow=c(2,2))  
ts.plot(price,main="USD TO EUR Exchange Rate", ylab="ExchangeRate")  
lines(mav.price.fit, lwd=2, col="red")  
lines(lm.price.fit,lwd=2,col="blue")  
lines(loc.price.fit,lwd=2,col="green")  
lines(gam.price.fit,lwd=2,col="brown")  
#adding note on the graph  
d=par("usr")  
legend(d[1]+6,1.1,c("Moving Average","Para-Polynomial","Loc-Polynomial","Splines"),lty=c(1,1),lwd=c(3,3), col=c("red","blue","green","brown"))  
  
#residual analysis  
#moving average  
mav.resid = price - mav.price.fit  
#Para-polynomial  
lm.resid = residuals(lm.fit)  
#loc-polynomial  
loc.resid = residuals(loc.fit)  
#Spline  
gam.resid = residuals(gam.fit)  
  
#residuals and ACF  
par(mfcol=c(2,2))



ts.plot(mav.resid, main="moving average residuals", xlab="Time", ylab="residuals",col="red")  
acf(as.numeric(mav.resid), main= "acf of moving average residuals")  
  
ts.plot(lm.resid, main="parametric quadratic polynomial residuals", xlab="Time", ylab="residuals",col="blue")  
acf(lm.resid, main= "acf of parametric quadratic polynomial residuals")



ts.plot(loc.resid, main="local polynomial residuals ",xlab="Time",ylab="residuals",col="green")  
acf(loc.resid, main= "acf of local polynomial residuals")  
  
ts.plot(gam.resid, main="Splines residuals",xlab="Time" ,ylab="residuals",col="brown")  
acf(gam.resid, main= "acf of Splines residuals")



*Response: Comparison of the fitted trend models:*

All four fitted models capture the variety of exchange rates over time, but none shows substantial cyclic patterns as the original time series graph shows. Based on the magnitude of the residuals, the Spline model captures the majority of the variety and some of the cyclical patterns. The p value of the smooth term of the trend indicates a statistically non-constant trend. The adjusted R square is 0.876, implying that about 87.6% of the variability is explained by the trend alone. One possible explanation is that the data does not have a fixed seasonality pattern over a 52-week period, so fitting it into a seasonality model is not suitable to observe the data. It also shows that the fluctuation showing in the original model may have less causation relationship to the seasonality.

As showing in ACF plots, all of the residual models has no trend but shows autocorrelations that decreases with the increase of lags. The variance is non-constant with time too. The model is less likely to be suitable for prediction or inference.

kpss.test(mav.resid, null="Trend")

##   
## KPSS Test for Trend Stationarity  
##   
## data: mav.resid  
## KPSS Trend = 1.3342, Truncation lag parameter = 7, p-value = 0.01

kpss.test(lm.resid, null="Trend")

##   
## KPSS Test for Trend Stationarity  
##   
## data: lm.resid  
## KPSS Trend = 0.55209, Truncation lag parameter = 7, p-value = 0.01

kpss.test(loc.resid, null="Trend")

##   
## KPSS Test for Trend Stationarity  
##   
## data: loc.resid  
## KPSS Trend = 0.22839, Truncation lag parameter = 7, p-value = 0.01

kpss.test(gam.resid, null="Trend")

##   
## KPSS Test for Trend Stationarity  
##   
## data: gam.resid  
## KPSS Trend = 0.047252, Truncation lag parameter = 7, p-value = 0.1

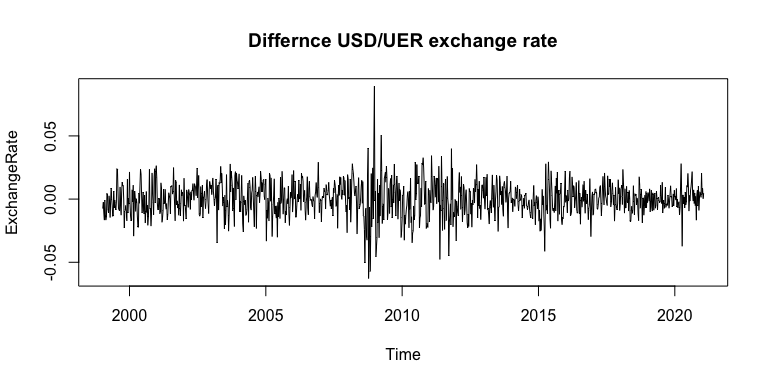
*Response: Appropriateness of the trend model for stationarity*

In general, the stationarity assumption is violated in all fitted models. The p values of all models are less than 0.05, showing the residuals are nonstationary. # Question 1c: Differenced Data Modeling

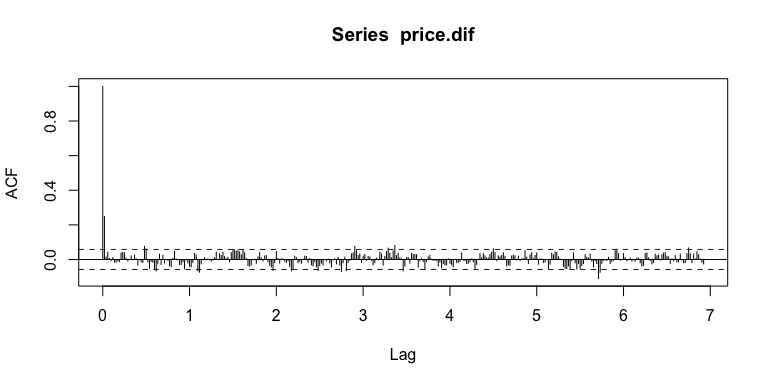
Now plot the difference time series and its ACF plot. Apply the four trend models in Question 1b to the differenced time series. What can you conclude about the difference data in terms of stationarity?

**Hint:** When TS data are differenced, the resulting dataset will have an NA in the first data element due to the differencing.

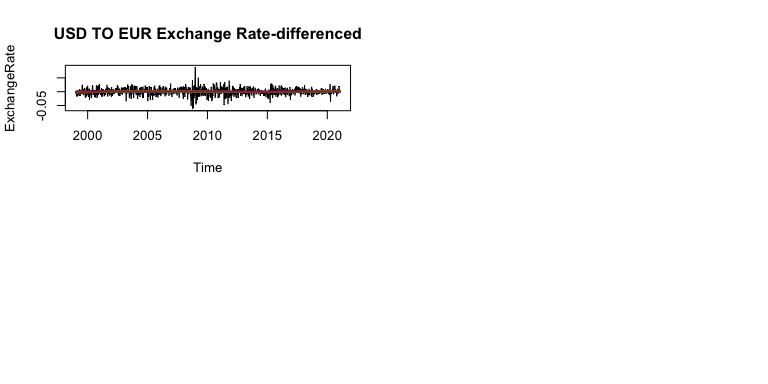
### Plot the difference time series  
price.dif <- diff(price)  
ts.plot(price.dif, ylab="ExchangeRate", main="Differnce USD/UER exchange rate")



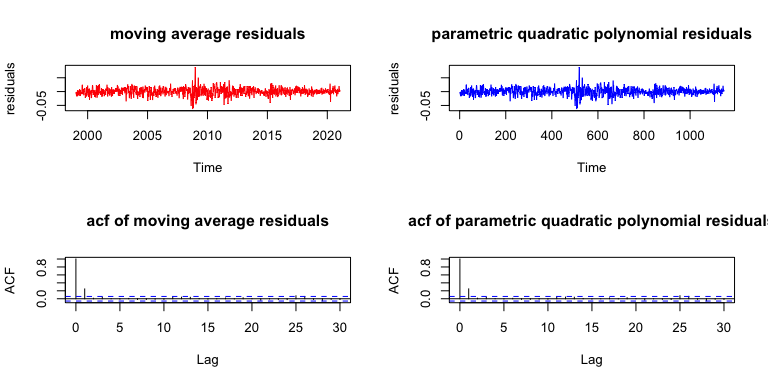
acf(price.dif, lag.max=360)



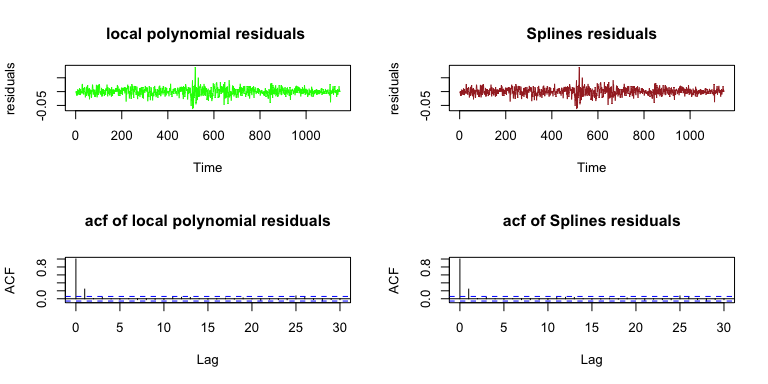
#moving average  
time.pts1 = c(1:length(price.dif))  
time.pts1=c(time.pts1-min(time.pts1))/max(time.pts1)  
price.mav.fit=ksmooth(time.pts1, price.dif, kernel="box")  
dif.mav.price.fit=ts(price.mav.fit$y, start=c(1999,2),frequency=52)  
  
#parametric quadratic polynomial  
x1=time.pts1  
x2=time.pts1^2  
price.lm.fit = lm(price.dif~x1+x2)  
dif.lm.price.fit=ts(fitted(price.lm.fit),start=c(1999,2),frequency = 52)  
  
#local polynomial  
price.loc.fit=loess(price.dif~time.pts1)  
dif.loc.price.fit=ts(fitted(price.loc.fit),start=c(1999,2),frequency = 52)  
#Splines  
library(mgcv)  
price.gam.fit=gam(price.dif~s(time.pts1))  
dif.gam.price.fit=ts(fitted(price.gam.fit),start=c(1999,2),frequency=52)  
  
#overlay the data  
par(mfrow=c(2,2))  
ts.plot(price.dif,main="USD TO EUR Exchange Rate-differenced", ylab="ExchangeRate")  
lines(dif.mav.price.fit, lwd=2, col="red")  
lines(dif.lm.price.fit,lwd=2,col="blue")  
lines(dif.loc.price.fit,lwd=2,col="green")  
lines(dif.gam.price.fit,lwd=2,col="brown")  
#adding note on the graph  
d=par("usr")  
legend(d[1]+6,1.1,c("Moving Average","Para-Polynomial","Loc-Polynomial","Splines"),lty=c(1,1),lwd=c(3,3), col=c("red","blue","green","brown"))  
  
#residual analysis  
#moving average  
mav.resid = price.dif-dif.mav.price.fit  
#Para-polynomial  
lm.resid = residuals(price.lm.fit)  
#loc-polynomial  
loc.resid = residuals(price.loc.fit)  
#Spline  
gam.resid = residuals(price.gam.fit)  
  
#residuals and ACF  
par(mfcol=c(2,2))



ts.plot(mav.resid, main="moving average residuals", xlab="Time", ylab="residuals",col="red")  
acf(as.numeric(mav.resid), main= "acf of moving average residuals")  
  
ts.plot(lm.resid, main="parametric quadratic polynomial residuals", xlab="Time", ylab="residuals",col="blue")  
acf(lm.resid, main= "acf of parametric quadratic polynomial residuals")



ts.plot(loc.resid, main="local polynomial residuals ",xlab="Time",ylab="residuals",col="green")  
acf(loc.resid, main= "acf of local polynomial residuals")  
  
ts.plot(gam.resid, main="Splines residuals",xlab="Time" ,ylab="residuals",col="brown")  
acf(gam.resid, main= "acf of Splines residuals")



```

*Response: Comments about the stationarity of the difference data:*

The differenced data shows the mean value is relatively constant. There’s no cyclical or seasonal pattern showing in the model, the variance still appears to change slightly over time. The ACF value mostly remain within 95% threshold. All these observations are showing the difference data is more consistent with stationarity assumption and hence more suitable for the analysis.

ISYE 6402 Homework 2 Solutions: Question 2

# Background

In this problem, we will analyze aggregated temperature data.

File *LA Temp Monthly.csv* download contains the monthly average temperature of Los Angeles from January 1950 through December 2018.

## Instructions on reading the data

To read the data in R, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the R function read.csv()

You will perform the analysis and modelling on the Temp data column.

fpath <- "~/Desktop/LA Temp Monthly.csv"  
df <- read.csv(fpath, head = TRUE)

Here are the libraries you will need:

library(mgcv)  
library(TSA)  
library(dynlm)

Run the following code to prepare the data for analysis:

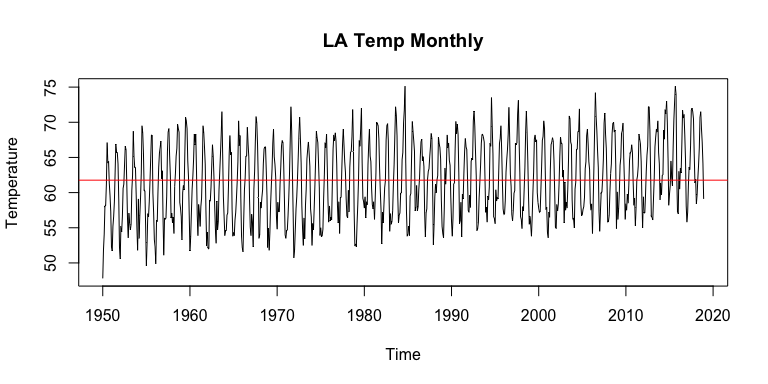
df$Date <- as.Date(paste0(df$Date, "01"), format = "%Y%m%d")  
temp <- ts(df$Temp, start = 1950, freq = 12)  
  
datenum <- ts(df$Date)

# Question 2a: Exploratory Data Analysis

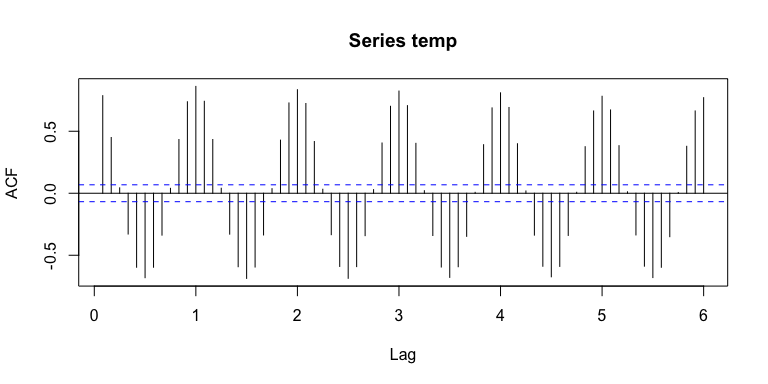
Plot both the Time Series and ACF plots. Comment on the main features, and identify what (if any) assumptions of stationarity are violated. Additionally, comment if you believe the differenced data is more appropriate for use in fitting the data. Support your response with a graphical analysis.

**Hint:** Make sure to use the appropriate differenced data.

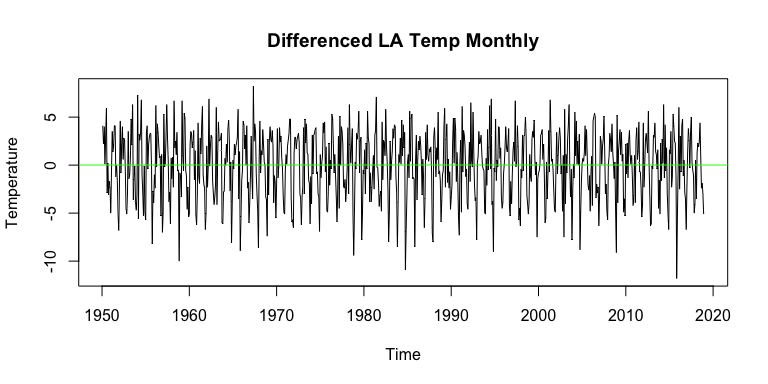
#original time series  
ts.plot(temp,ylab="Temperature",main="LA Temp Monthly")  
abline(a=mean(temp),b=0,col="red")



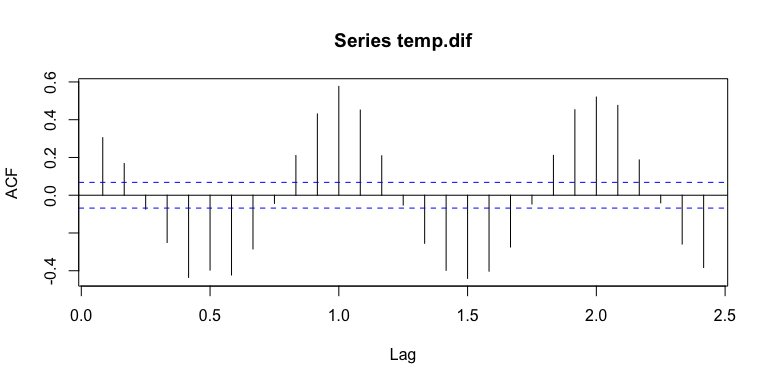
acf(temp,lag.max=72)

 *Response: Comments about the time series and ACF plots of the original time series* In the original time series analysis graph, the data shows a strong seasonal cyclical pattern with a slight upward trend; it looks like there’s a non-constant mean and a upward trend in the original graph hence the stationarity assumption is violated. The seasonality pattern is appearant and stronger than the trend, hence a seasonality model would fit better than the forecasting trend model alone when it comes analyzing the data.

#differenced data analysis  
temp.dif <- diff(temp,1)  
ts.plot(temp.dif, ylab="Temperature",main="Differenced LA Temp Monthly")  
abline(a=mean(temp.dif),b=0,col="green")



acf(temp.dif,lax.max=72)



*Response: Comments about the time series and ACF plots of the difference time series* After differences the data, the mean is 0 and it eliminated the slight upward of the trend. Given that we already observed that the data shows strong seasonality and it’s non-stationary, the difference time series graph reflects a better fitted model.

# Question 2b: Seasonality Estimation

On the temperature data, separately fit a seasonality harmonic model and the ANOVA seasonality model. Evaluate the quality of each fit with residual analysis. Does one model perform better than the other? Which model would you select to fit the seasonality in the data?

library(dynlm)  
month = as.factor(format(df$Dates, "%b"))  
#ANOVA model  
seasonano <- dynlm(temp~season(temp))  
summary(seasonano)

##   
## Time series regression with "ts" data:  
## Start = 1950(1), End = 2018(12)  
##   
## Call:  
## dynlm(formula = temp ~ season(temp))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.729 -1.581 -0.037 1.625 8.735   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 55.5290 0.2841 195.476 < 2e-16 \*\*\*  
## season(temp)Feb 0.7275 0.4017 1.811 0.070511 .   
## season(temp)Mar 1.5623 0.4017 3.889 0.000109 \*\*\*  
## season(temp)Apr 3.6043 0.4017 8.972 < 2e-16 \*\*\*  
## season(temp)May 6.0812 0.4017 15.137 < 2e-16 \*\*\*  
## season(temp)Jun 9.1159 0.4017 22.691 < 2e-16 \*\*\*  
## season(temp)Jul 12.3841 0.4017 30.826 < 2e-16 \*\*\*  
## season(temp)Aug 13.4246 0.4017 33.417 < 2e-16 \*\*\*  
## season(temp)Sep 12.7710 0.4017 31.790 < 2e-16 \*\*\*  
## season(temp)Oct 9.7362 0.4017 24.235 < 2e-16 \*\*\*  
## season(temp)Nov 4.9464 0.4017 12.313 < 2e-16 \*\*\*  
## season(temp)Dec 0.5188 0.4017 1.291 0.196897   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.36 on 816 degrees of freedom  
## Multiple R-squared: 0.8124, Adjusted R-squared: 0.8098   
## F-statistic: 321.2 on 11 and 816 DF, p-value: < 2.2e-16

seasonano1<- dynlm(temp~season(temp)-1)  
summary(seasonano1)

##   
## 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   
## -7.729 -1.581 -0.037 1.625 8.735   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## season(temp)Jan 55.5290 0.2841 195.5 <2e-16 \*\*\*  
## season(temp)Feb 56.2565 0.2841 198.0 <2e-16 \*\*\*  
## season(temp)Mar 57.0913 0.2841 201.0 <2e-16 \*\*\*  
## season(temp)Apr 59.1333 0.2841 208.2 <2e-16 \*\*\*  
## season(temp)May 61.6101 0.2841 216.9 <2e-16 \*\*\*  
## season(temp)Jun 64.6449 0.2841 227.6 <2e-16 \*\*\*  
## season(temp)Jul 67.9130 0.2841 239.1 <2e-16 \*\*\*  
## season(temp)Aug 68.9536 0.2841 242.7 <2e-16 \*\*\*  
## season(temp)Sep 68.3000 0.2841 240.4 <2e-16 \*\*\*  
## season(temp)Oct 65.2652 0.2841 229.8 <2e-16 \*\*\*  
## season(temp)Nov 60.4754 0.2841 212.9 <2e-16 \*\*\*  
## season(temp)Dec 56.0478 0.2841 197.3 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.36 on 816 degrees of freedom  
## Multiple R-squared: 0.9986, Adjusted R-squared: 0.9986   
## F-statistic: 4.757e+04 on 12 and 816 DF, p-value: < 2.2e-16

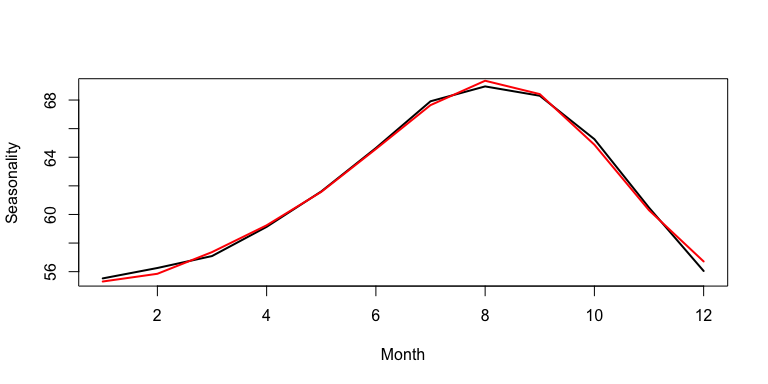
#Harmonic model  
seasonhar <- dynlm(temp~harmon(temp))  
summary(seasonhar)

##   
## Time series regression with "ts" data:  
## Start = 1950(1), End = 2018(12)  
##   
## Call:  
## dynlm(formula = temp ~ harmon(temp))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.8022 -1.7405 -0.0916 1.5677 9.4140   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 61.76836 0.08714 708.81 <2e-16 \*\*\*  
## harmon(temp)cos -6.16619 0.12324 -50.03 <2e-16 \*\*\*  
## harmon(temp)sin -2.81769 0.12324 -22.86 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.508 on 825 degrees of freedom  
## Multiple R-squared: 0.7858, Adjusted R-squared: 0.7853   
## F-statistic: 1513 on 2 and 825 DF, p-value: < 2.2e-16

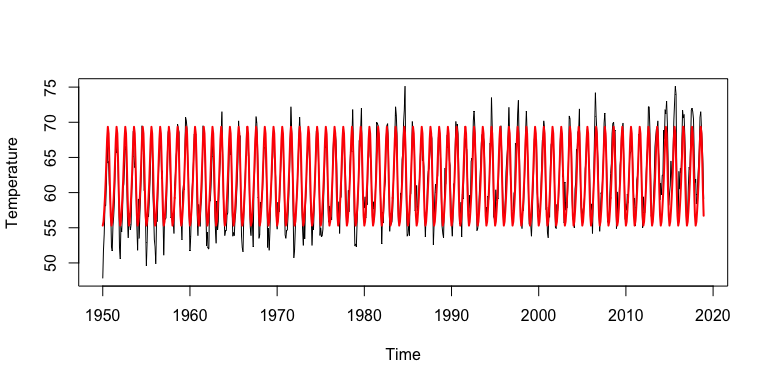
seasonhar1 <- dynlm(temp~harmon(temp,2))  
summary(seasonhar1)

##   
## 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   
## -7.5116 -1.6499 -0.1294 1.5971 9.1234   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 61.76836 0.08238 749.797 <2e-16 \*\*\*  
## harmon(temp, 2)cos1 -6.16619 0.11650 -52.927 <2e-16 \*\*\*  
## harmon(temp, 2)cos2 -0.29058 0.11650 -2.494 0.0128 \*   
## harmon(temp, 2)sin1 -2.81769 0.11650 -24.186 <2e-16 \*\*\*  
## harmon(temp, 2)sin2 1.12918 0.11650 9.692 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.37 on 823 degrees of freedom  
## Multiple R-squared: 0.809, Adjusted R-squared: 0.8081   
## F-statistic: 871.6 on 4 and 823 DF, p-value: < 2.2e-16

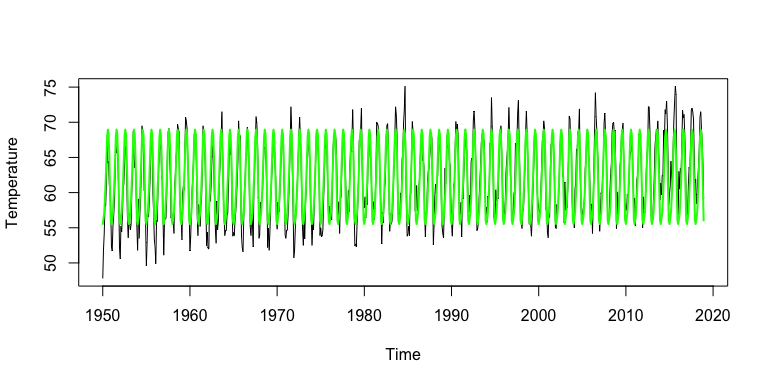
#Given that season-har.1 and season.anov.1 have higher R-squared, they are used in seasonality models below: comparing seasonality estimates  
st1 <- coef(seasonano1)  
st2 <- fitted(seasonhar1)[1:12]  
plot(1:12,st1,lwd=2,type="l",xlab="Month",ylab="Seasonality")  
lines(1:12,st2,lwd=2, col="red")



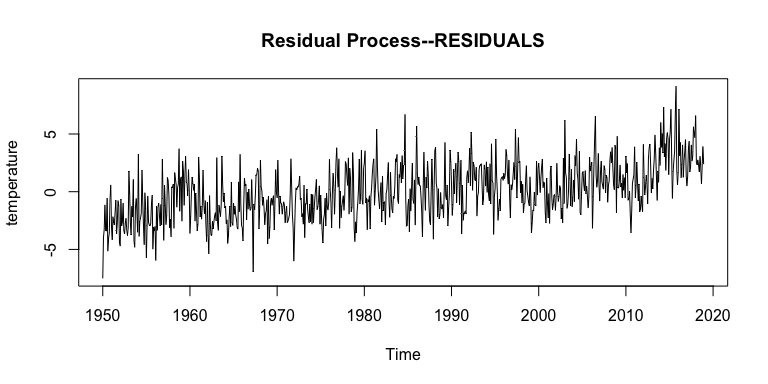
#plotting seasonality fits:  
temp.season=ts(fitted(seasonhar1), start=1950, frequency=12)  
ts.plot(temp, ylab="Temperature")  
lines(temp.season,lwd=2, col="red")



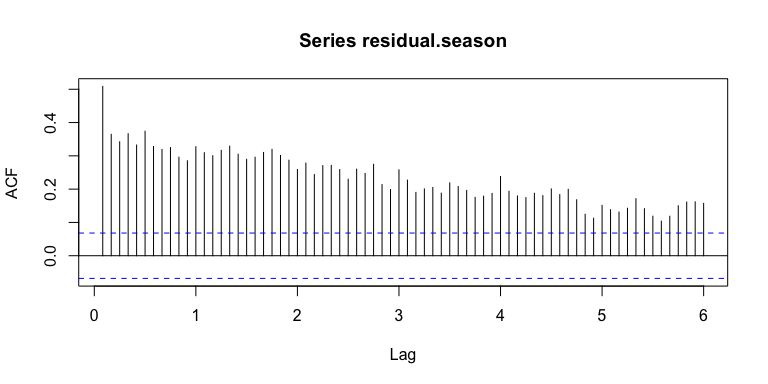
temp.season2=ts(fitted(seasonano1),start=1950, frequency=12)  
ts.plot(temp, ylab="Temperature")  
lines(temp.season2,lwd=2,col="green")



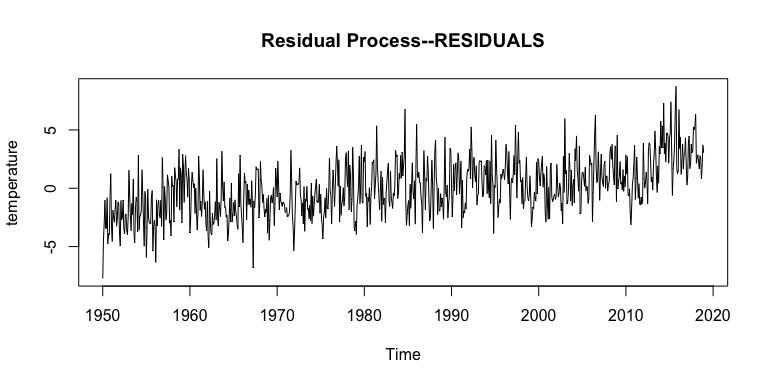
#running a residual analysis  
residual.season <- ts((temp-fitted(seasonhar1)), start=1950, frequency=12)  
ts.plot(residual.season,ylab= "temperature", main="Residual Process--RESIDUALS")



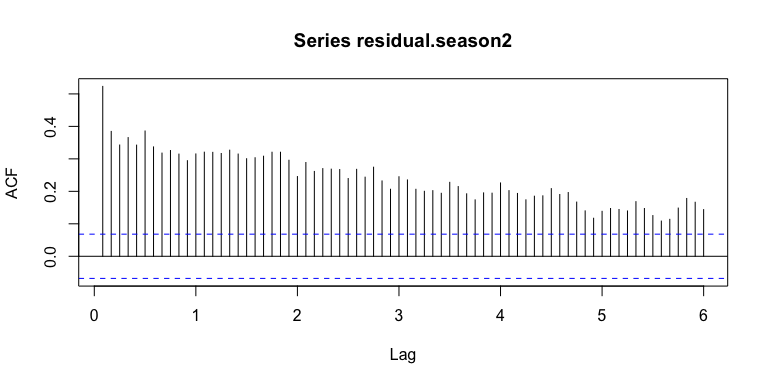
acf(residual.season,lag.max=72)



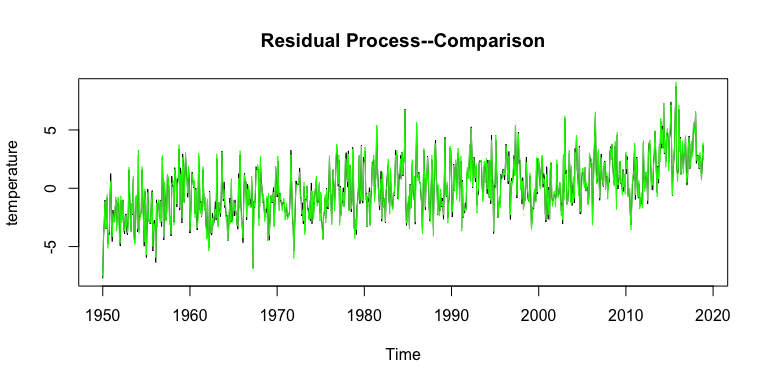
residual.season2 <- ts((temp-fitted(seasonano1)), start=1950, frequency=12)  
ts.plot(residual.season2,ylab= "temperature", main="Residual Process--RESIDUALS")



acf(residual.season2,lag.max=72)



#The residual analysis shows a clear trend, the ACF plot still shows a significant but declining autocorrelation.   
  
# comparing residuals of two models  
ts.plot(residual.season2,ylab= "temperature", main="Residual Process--Comparison")  
lines(residual.season, col="green")



*Response: Compare Seasonality Models*

Both models capture the majority of seasonal variation yet neither captures the trend. When looking at the residuals plot, we can tell that the residuals are not stationary and is changing over time: it moves up through the given time. This violates the constant mean assumption. In addition, both models exhibit significant autocorrelation in the residuals. The cos-sin model failed in capturing some seasonal components due to the fact that the observed seasonality cannot be perfectly fitted into sinusoidal curve. The ANOVA model, on the other hand, shows the remaining autocorrelation might mostly from the trend. We can tell from the comparison that the ANNOVA and cos-sin models have similar pattern but we are using cos-sin model since it has fewer regression coefficients. After fitting the data into the seasonality (cos-sin model), we see it fits well in the original data with no trend. # Question 2c: Trend-Seasonality Estimation

Using the time series data, fit the following models to estimate the trend with seasonality fitted using ANOVA:

* Parametric Polynomial Regression
* Non-parametric model

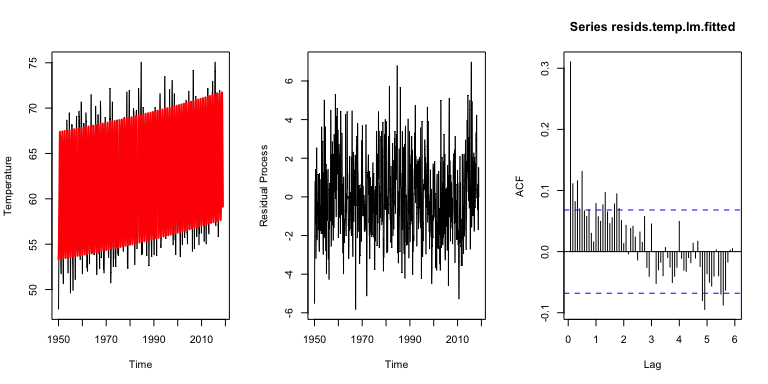
Overlay the fitted values on the original time series. Plot the residuals with respect to time. Plot the ACF of the residuals. Comment on how the two models fit and on the appropriateness of the stationarity assumption of the residuals.

What form of modelling seems most appropriate and what implications might this have for how one might expect long term temperature data to behave? Provide explicit conclusions based on the data analysis.

par(mfrow=(c(1,3)))  
time.pts <- c(1:length(temp))  
time.pts <- c(time.pts -min(time.pts))/max(time.pts)  
x1 <- time.pts  
x2 <- time.pts^2  
#Parametric Polynomial Regression analysis  
lm.fit <- dynlm(temp ~x1+x2+harmon(temp,2))  
summary(lm.fit)

##   
## Time series regression with "ts" data:  
## Start = 1950(1), End = 2018(12)  
##   
## Call:  
## dynlm(formula = temp ~ x1 + x2 + harmon(temp, 2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.8276 -1.3518 -0.1118 1.2889 6.9882   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 59.77924 0.20766 287.871 < 2e-16 \*\*\*  
## x1 3.11941 0.96035 3.248 0.00121 \*\*   
## x2 1.29623 0.93090 1.392 0.16416   
## harmon(temp, 2)cos1 -6.16088 0.09813 -62.785 < 2e-16 \*\*\*  
## harmon(temp, 2)cos2 -0.28525 0.09813 -2.907 0.00375 \*\*   
## harmon(temp, 2)sin1 -2.79779 0.09813 -28.511 < 2e-16 \*\*\*  
## harmon(temp, 2)sin2 1.13842 0.09813 11.601 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.997 on 821 degrees of freedom  
## Multiple R-squared: 0.8649, Adjusted R-squared: 0.8639   
## F-statistic: 875.6 on 6 and 821 DF, p-value: < 2.2e-16

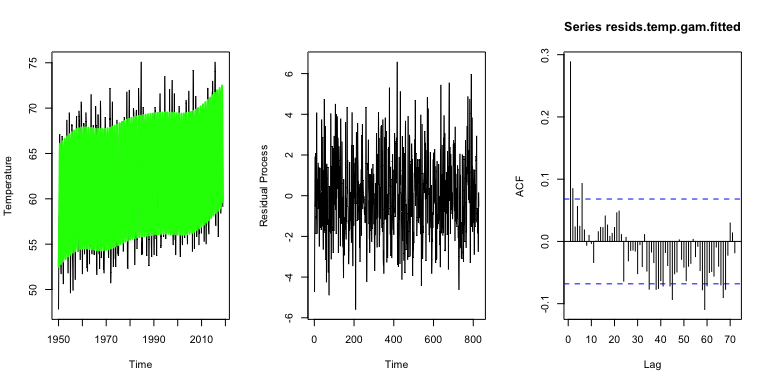
temp.lm.fitted <- ts(fitted(lm.fit), start=1950,frequency=12)  
#plot the fitted model and the original data  
ts.plot(temp,ylab="Temperature")  
lines(temp.lm.fitted,lwd=2,col="red")  
#The fitted model reflects the trend and seasonality of the original data with the exception of several extremes.   
#Plot the residuals and ACF  
resids.temp.lm.fitted <- residuals(lm.fit)  
ts.plot(resids.temp.lm.fitted, ylab="Residual Process")  
acf(resids.temp.lm.fitted,lag.max=72)



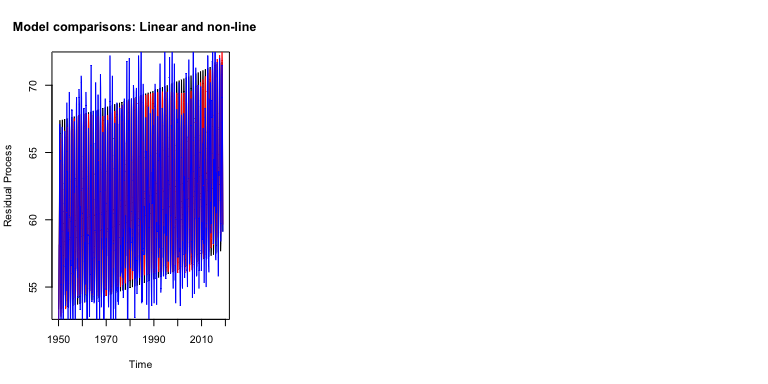
#We observe there's a significant autocorrelation before lag 2 and in between lag5 to lag 6, it shows the data is non stationary.   
  
  
#non-parametric Analysis   
month <- season(ts(temp,start=1950,frequency=12))  
gam.fit <- gam(temp~s(x1)+month)  
temp.gam.fitted <- ts(fitted(gam.fit),start=1950, frequency=12)  
summary(gam.fit)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## temp ~ s(x1) + month  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 55.5735 0.2323 239.203 < 2e-16 \*\*\*  
## monthFebruary 0.7194 0.3285 2.190 0.0288 \*   
## monthMarch 1.5461 0.3285 4.706 2.97e-06 \*\*\*  
## monthApril 3.5801 0.3285 10.897 < 2e-16 \*\*\*  
## monthMay 6.0488 0.3285 18.411 < 2e-16 \*\*\*  
## monthJune 9.0755 0.3285 27.624 < 2e-16 \*\*\*  
## monthJuly 12.3355 0.3285 37.546 < 2e-16 \*\*\*  
## monthAugust 13.3680 0.3286 40.687 < 2e-16 \*\*\*  
## monthSeptember 12.7063 0.3286 38.672 < 2e-16 \*\*\*  
## monthOctober 9.6634 0.3286 29.410 < 2e-16 \*\*\*  
## monthNovember 4.8655 0.3286 14.808 < 2e-16 \*\*\*  
## monthDecember 0.4299 0.3286 1.308 0.1912   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(x1) 7.271 8.282 49.16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.873 Deviance explained = 87.6%  
## GCV = 3.8122 Scale est. = 3.7235 n = 828

#plot the fitted model and the original data  
ts.plot(temp,ylab="Temperature")  
lines(temp.gam.fitted,lwd=2,col="green")  
#plot the residuals and ACF  
resids.temp.gam.fitted <- residuals(gam.fit)  
ts.plot(resids.temp.gam.fitted, ylab="Residual Process")  
acf(resids.temp.gam.fitted,lag.max=72)



#We observe significant autocorrelation aroud lag 3 and in between lag5 to lag 6, it shows the data is non stationary.   
  
#comparing two models  
ts.plot(temp.lm.fitted, ylab="Residual Process", main="Model comparisons: Linear and non-linear")  
lines(temp.gam.fitted, col="red")  
lines(temp,col="blue")  
  
#we observe that the differences are small between the two different approaches



*Response: Model Comparison* Both graphs show fitted data obtained by linear and non-linear models capture the data trend and seasonality well. The non-parametric model as it captures more variety in the fitted vs original data comparisons. When comparing the standard error of the regression for two models, the non-parametric model is smaller, which indicates that the data points are closer to the fitted line. Also, the ACF of non-parametric model (fitted) displays the randomness that is desirable in an unbiased model. When comparing the ACF plots of the residuals, the residuals of fitted Parametric Polynomial Regression shows more frequent lags that are beyond the 0.95% range, i.e., in between lag 1 to 2, and lag 5 to 6; while the residuals of fitted non-parametric model shows that significant variance after lag 40. To sum, we can conclude that the non-parametric model provides a better fit because it is unbiased and produces smaller residuals. We can predict that long term data follows a seasonal pattern and is going to be slightly increasing over years.