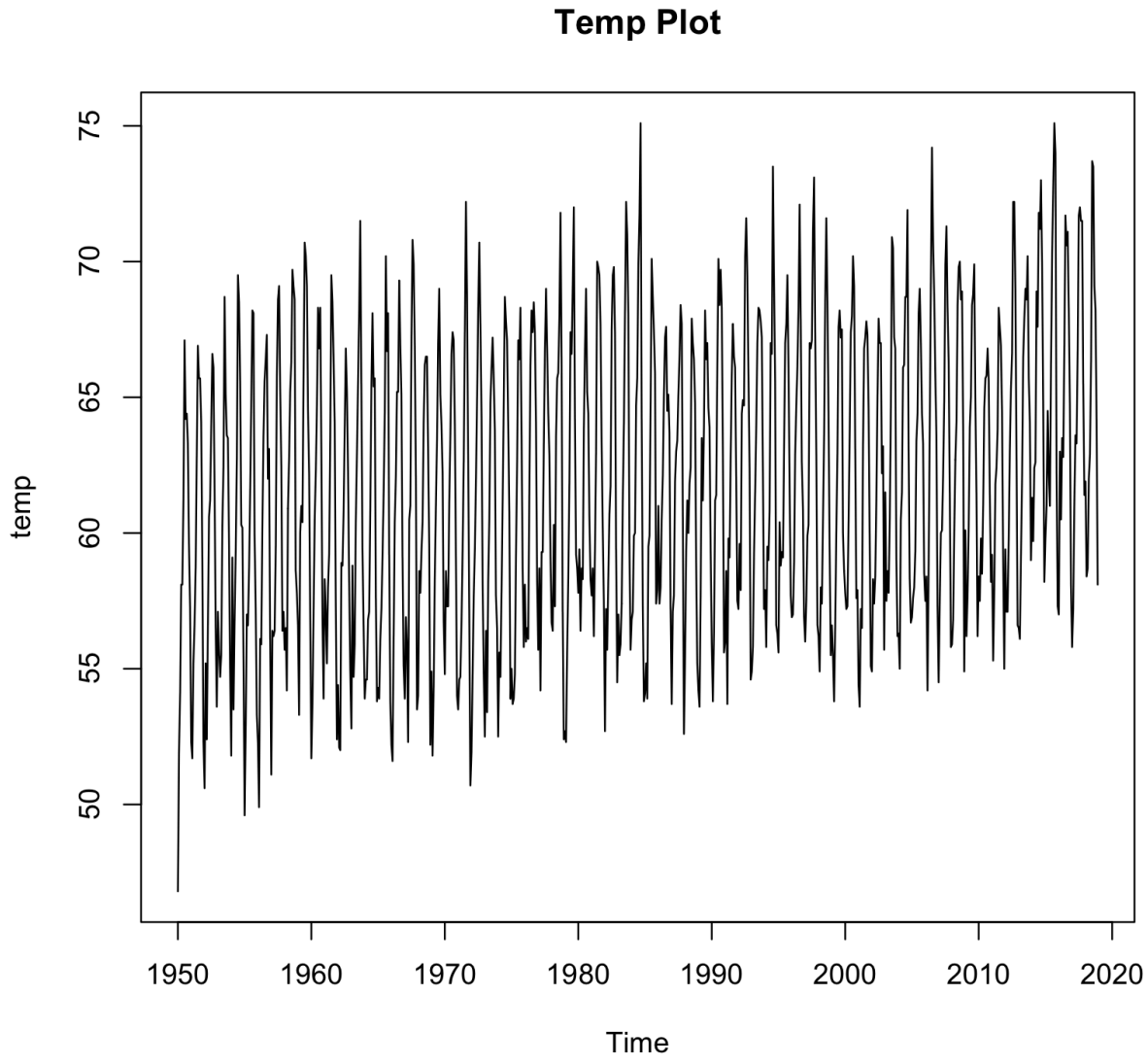
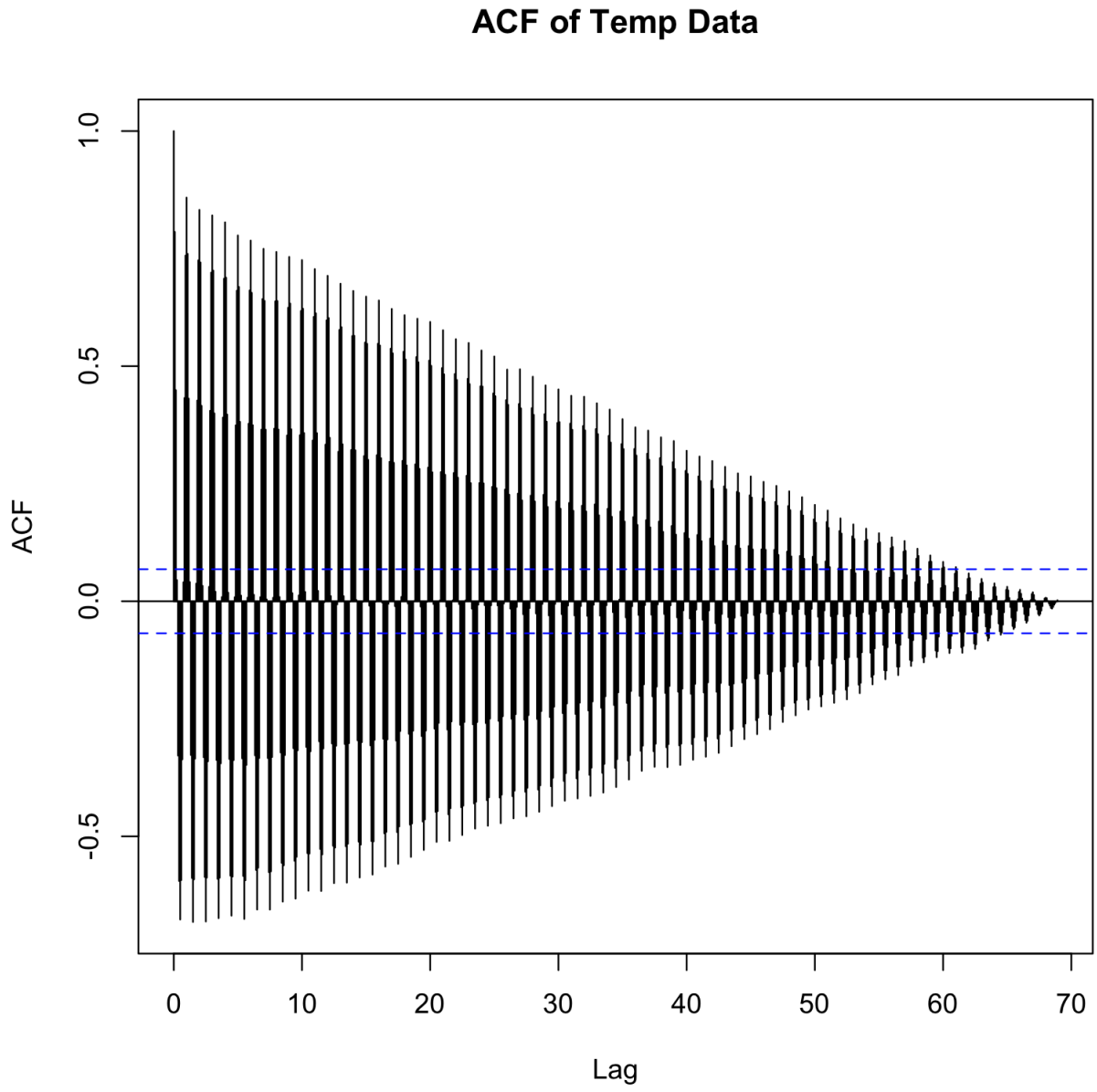
```r
# npark62 01/26/20

# Question 1

library(TSA)
library(mgcv)

#Choose the 'LA Temp Monthly.csv' dataset, wherever it
 is located on your computer
#Additionally, just skip this step and replace ‘fname’
 with the files direct location
# fname <- file.choose("~/Downloads/
 LA_Temp_Monthly.csv")

#Load Data
data <- read.csv("~/Desktop/LA_Temp_Monthly.csv",
 header = FALSE)
tempData <- data[2:829, 2]
tempData <- as.numeric(levels(tempData))[tempData] #
 tempData is in factor. Needs conversion to numeric.
 Super important
#Convert to TS data in proper frame
temp <- ts(tempData, start = c(1950,1), freq = 12)
temp_acf = acf(temp, lag.max = 1000)


# Question 1a

# Before plotting this time series, we can presume that
 the atmospheric temperature would follow a regular
 seasonal pattern due to the eliptical orbit of the
 planet, i.e. summer generally being hot and winter
 generally being cold. We can also assume a linear trend
 of the yearly temperature rising due to global warming.
```

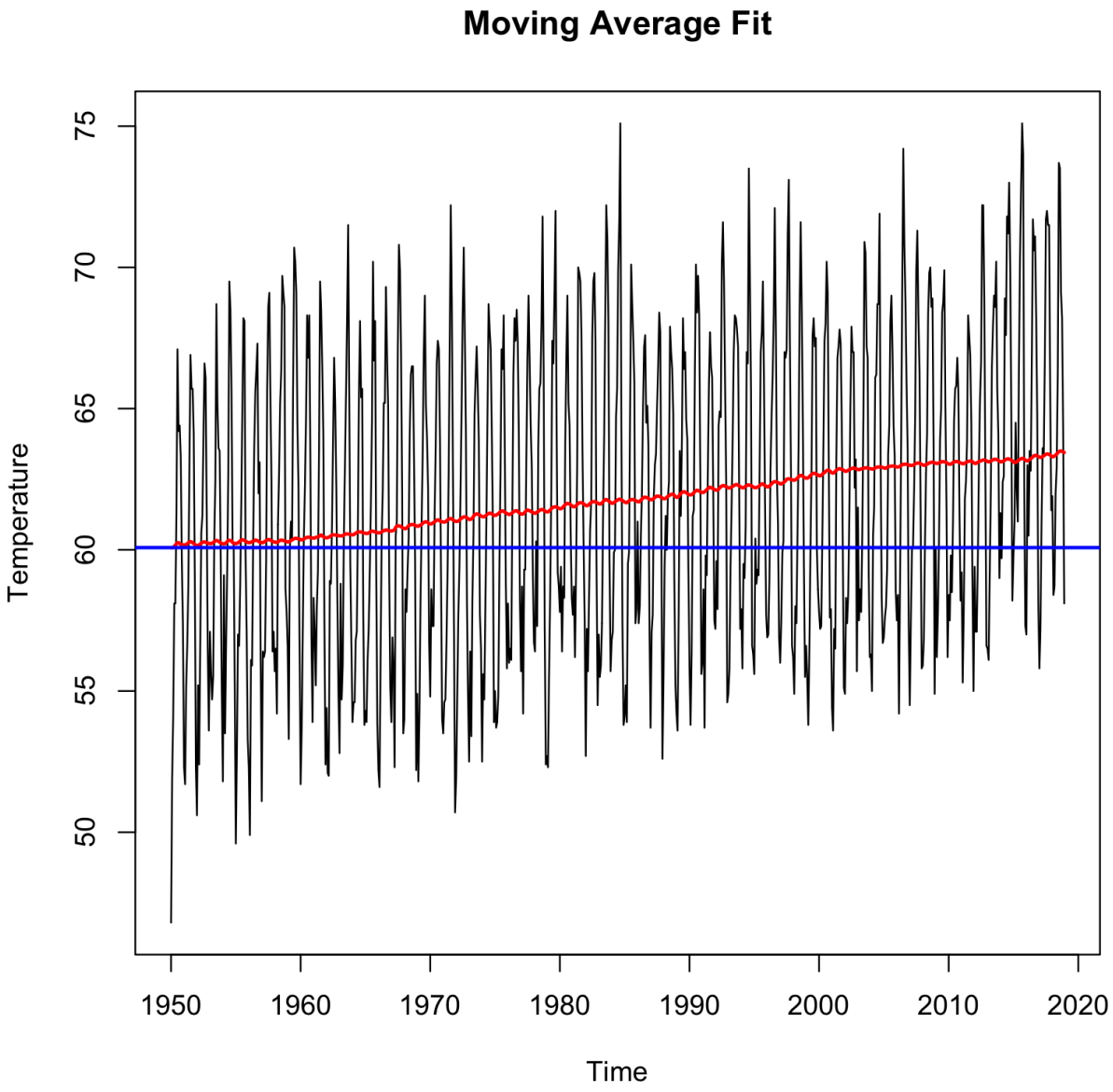
```r
plot(temp, main = "Temp Plot")
plot(temp_acf, main = "ACF of Temp Data",)
```





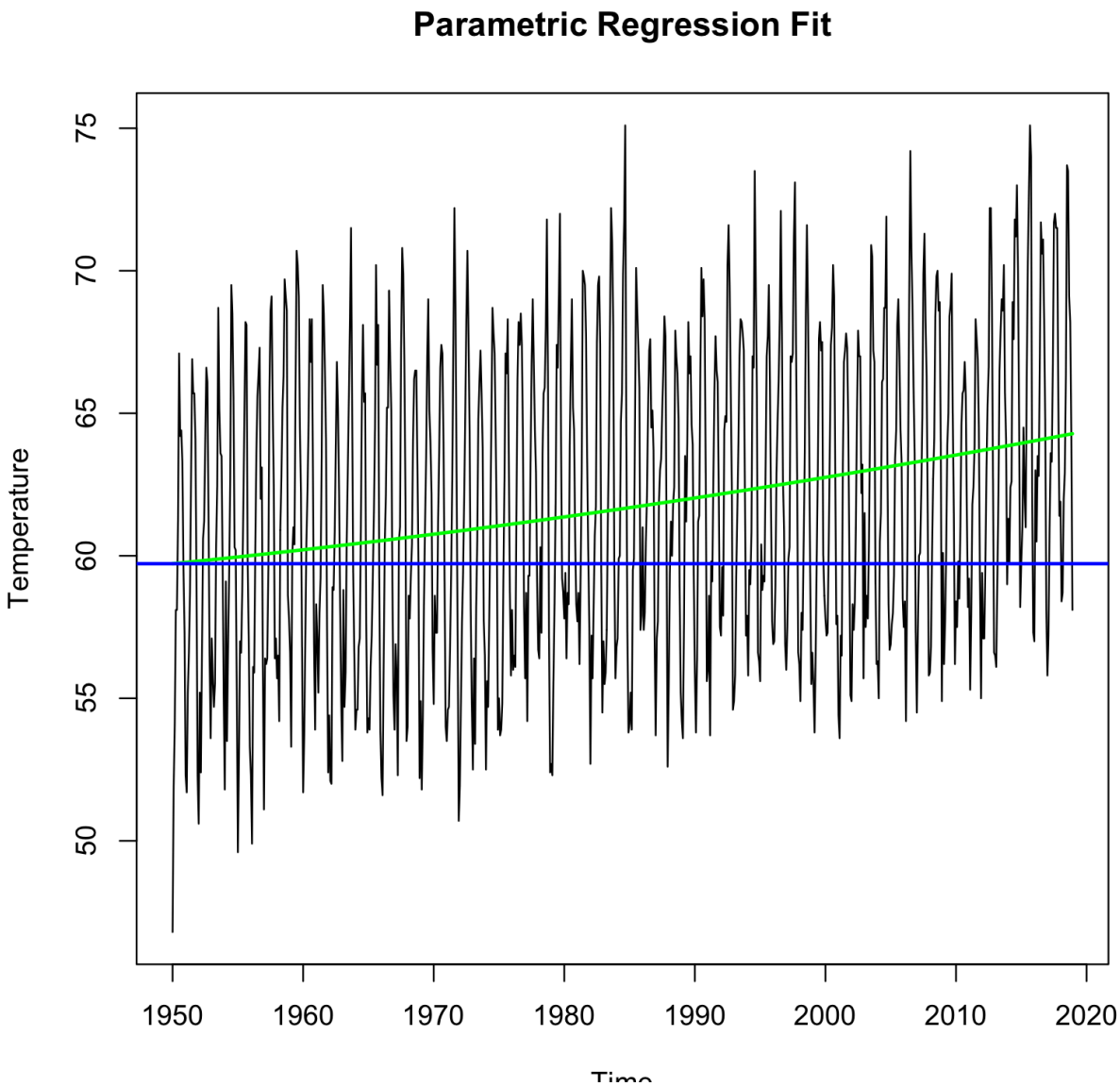

```
# As identified by above assumptions, the time series
 plot shows an upward linear trend as well as a
 seasonality within each year.
# Assumptions for stationary time series: 1) constant
 mean, 2) finite variance, and 3) time independent
 autocovariance
```

```r
# Constant mean is violated. Whether the time
 independent autocovariance is violated is uncertain.
 The ACF plot shows significant autocovariance until k =
 60 and also that it does not converge to very quickly 0
 as k increases.
# Seasonality fitting is assumed to be better fitting
 to the data due to the high seasonality of weather we
 physically experience each year.

# Question 1b

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

# Moving Average Fit

mav.fit = ksmooth(time.pts, temp, kernel = "box")
temp.fit.mav = ts(mav.fit$y, start = 1950, freq = 12)
ts.plot(temp, ylab = "Temperature", main = "Moving
 Average Fit")
lines(temp.fit.mav, lwd = 2, col = "red")
abline(temp.fit.mav[1], 0 , lwd = 2, col = "blue")
```



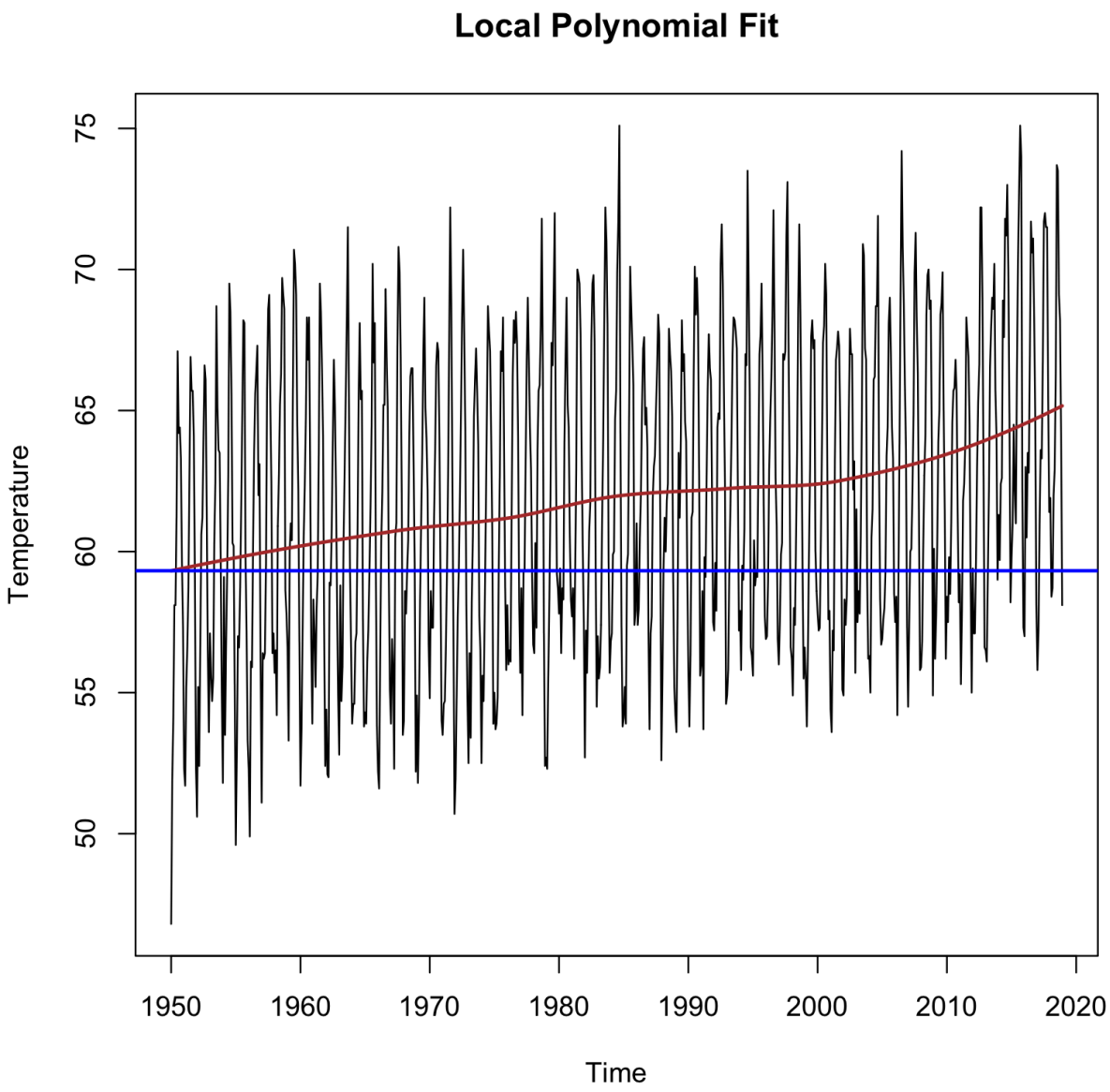

```
# Parametric Quadratic Regression Fit

x1 = time.pts
x2 = time.pts^2
lm.fit = lm(temp~x1+x2)
summary(lm.fit)
temp.fit.lm = ts(fitted(lm.fit), start = 1950, freq =
```

```
 12)
ts.plot(temp, ylab = "Temperature", main = "Parametric
 Regression Fit")
lines(temp.fit.lm, lwd =2 , col= "green")
abline(temp.fit.lm[1], 0 , lwd = 2, col = "blue")
```

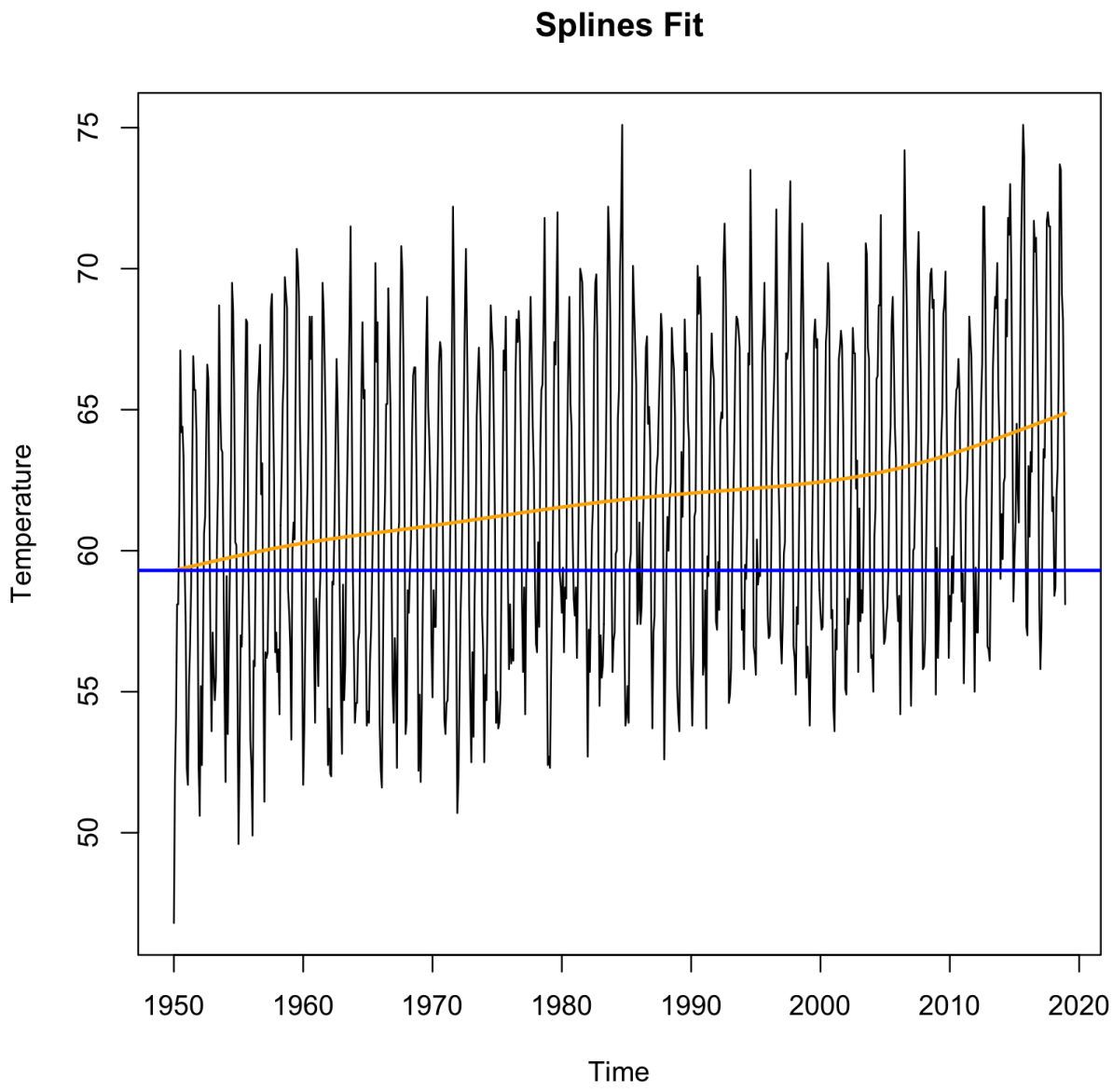



```
# Local Polynomial Fit

loc.fit = loess(temp~time.pts)
temp.fit.loc = ts(fitted(loc.fit),start=1950,freq=12)
ts.plot(temp,ylab="Temperature", main = "Local
```

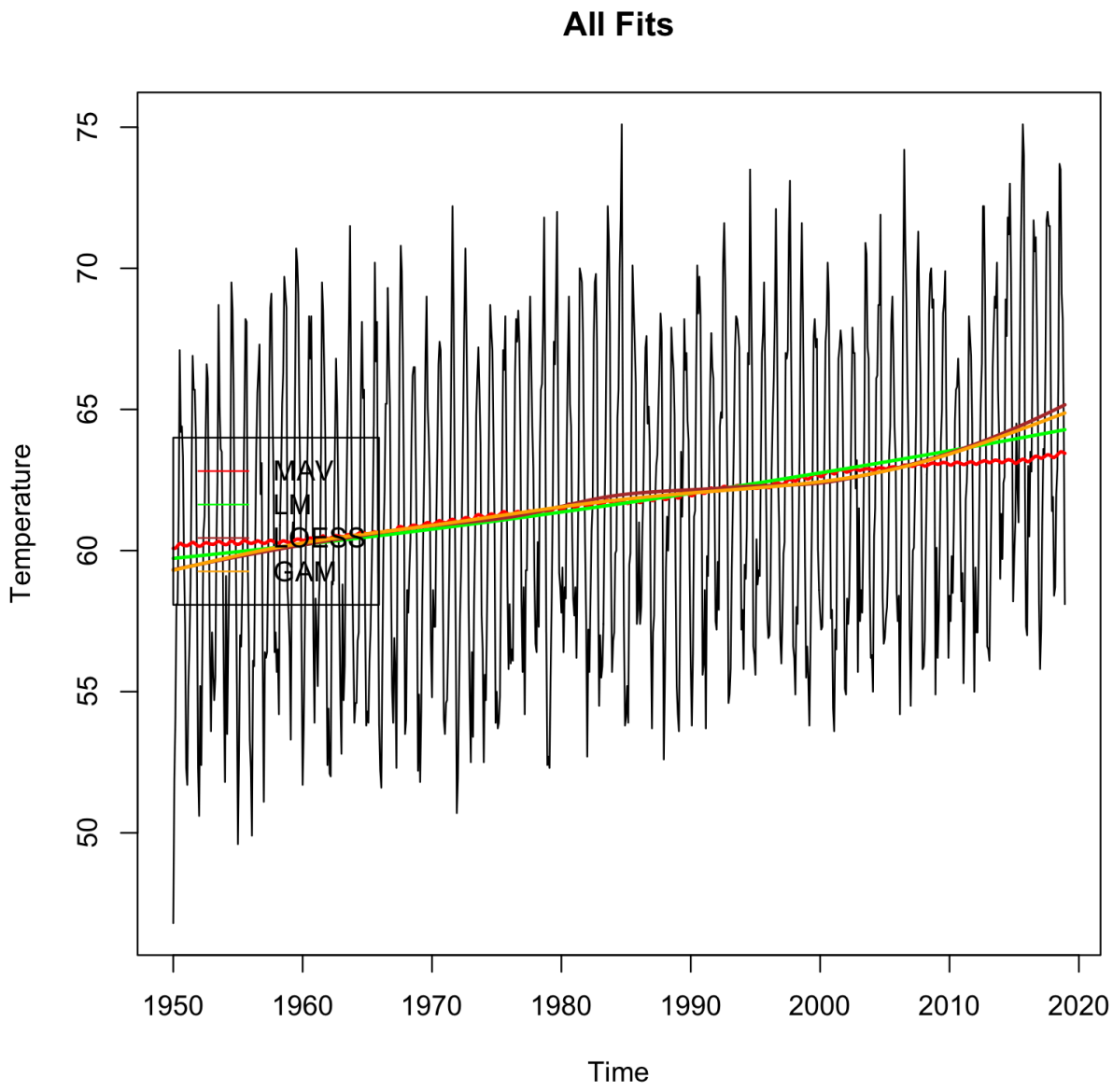
```
 Polynomial Fit")
lines(temp.fit.loc,lwd=2,col="brown")
abline(temp.fit.loc[1],0,lwd=2,col="blue")
```



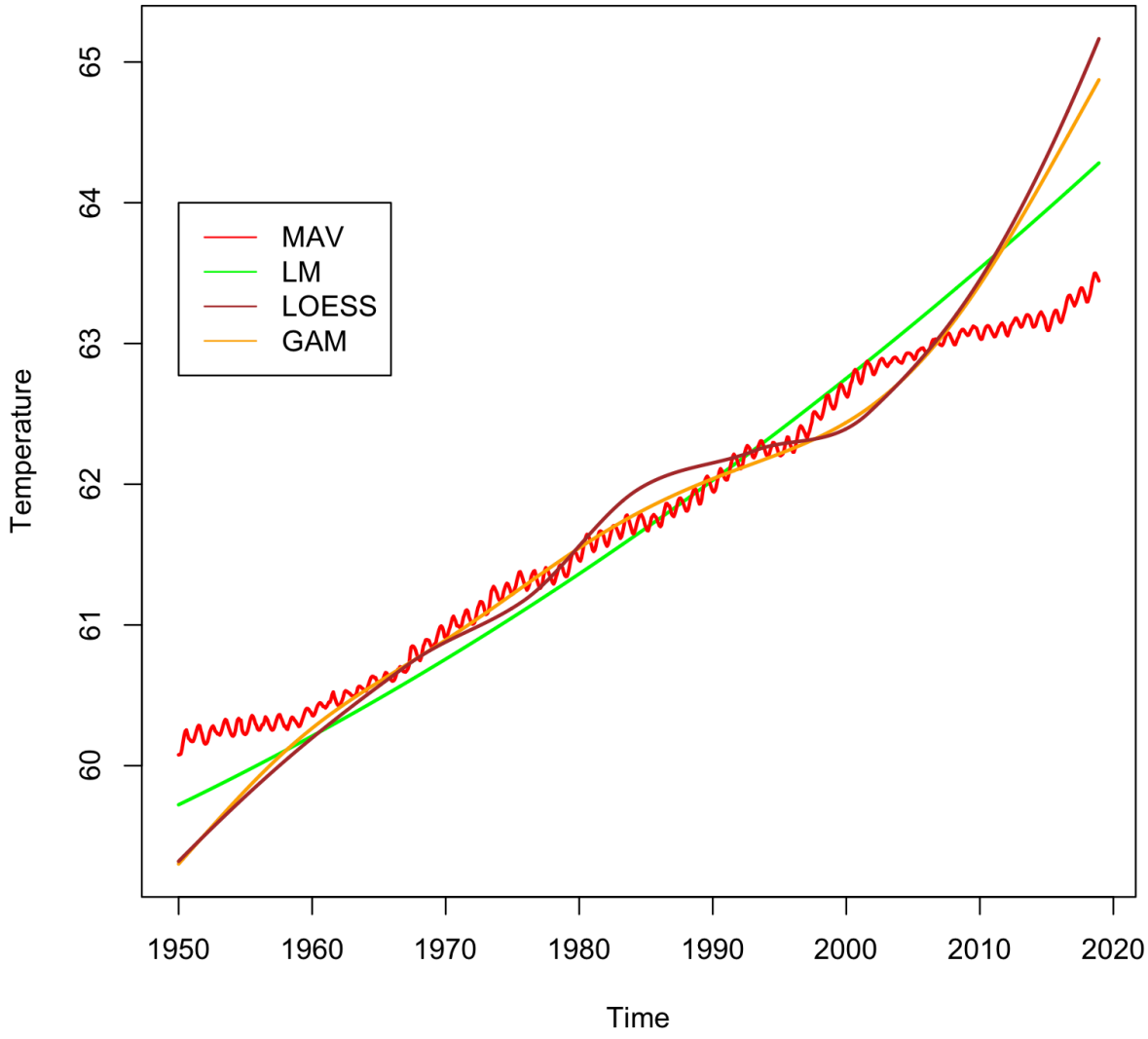

```
# Splines Fit

gam.fit = gam(temp~s(time.pts))
temp.fit.gam = ts(fitted(gam.fit),start=1950,freq=12)
ts.plot(temp,ylab="Temperature", main = "Splines Fit")
lines(temp.fit.gam,lwd=2,col="orange")
abline(temp.fit.gam[1],0,lwd=2,col="blue")
```



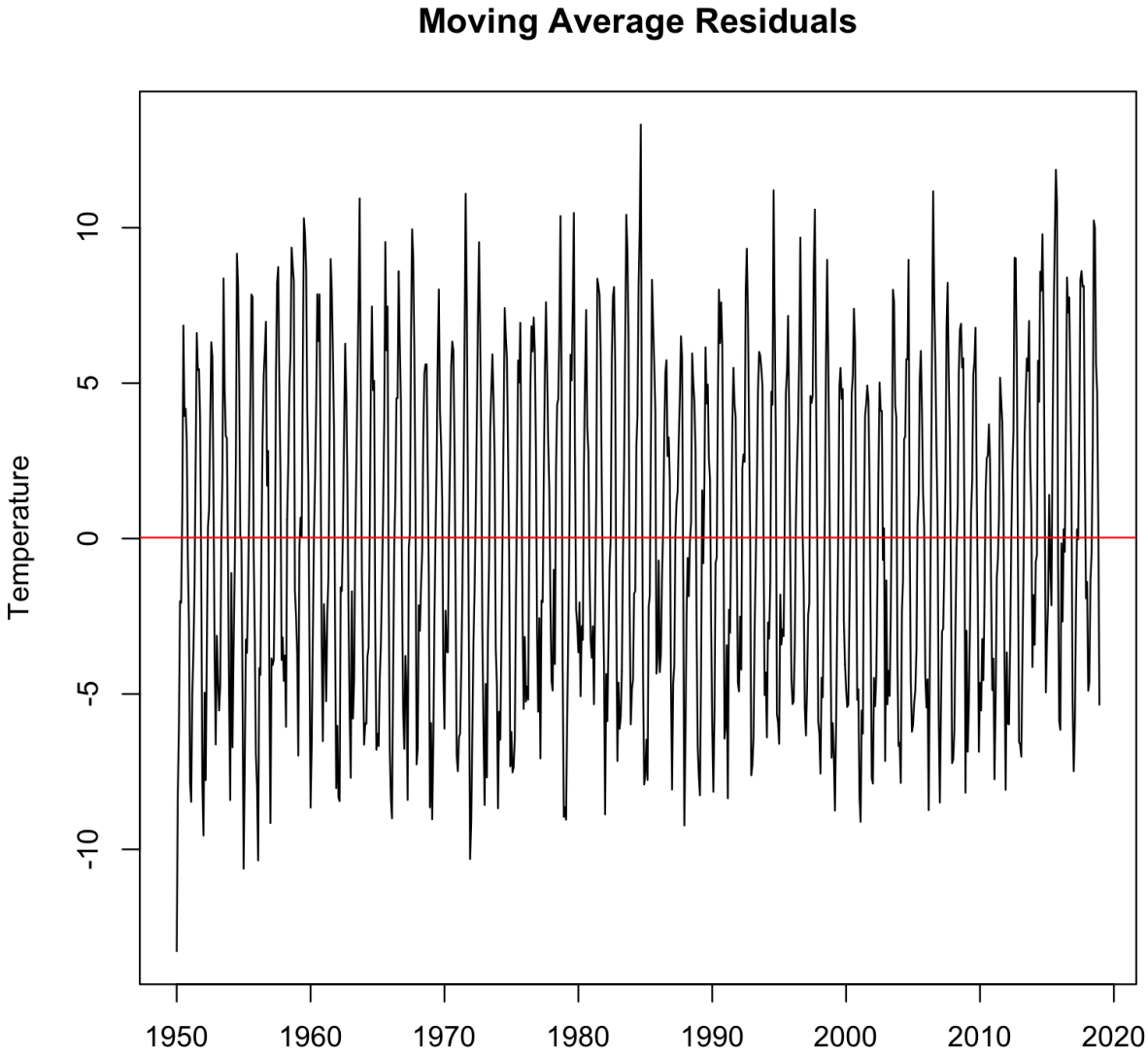

```
# Overlay altogether

ts.plot(temp,ylab="Temperature", main = "All Fits")
lines(temp.fit.mav, lwd = 2, col = "red")
lines(temp.fit.lm, lwd =2 , col= "green")
lines(temp.fit.loc,lwd=2,col="brown")
lines(temp.fit.gam,lwd=2,col="orange")
legend(x=1950,y=64,legend=c("MAV","LM","LOESS","GAM"),l
 ty = 1, col=c("red","green","brown","orange"))
```

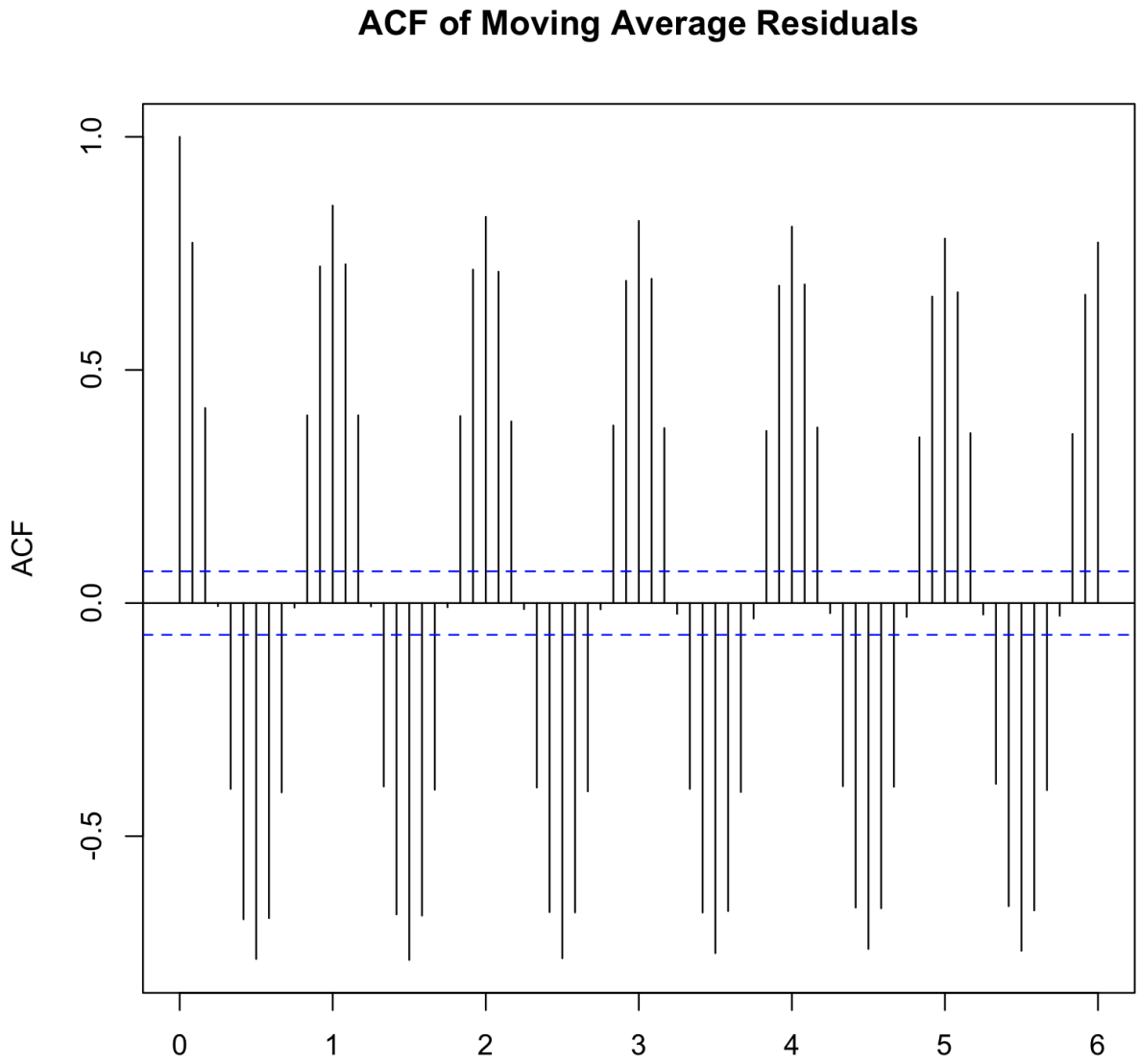



```
all.val =
 c(temp.fit.mav,temp.fit.lm,temp.fit.loc,temp.fit.gam)
ylim= c(min(all.val),max(all.val))
ts.plot(temp.fit.lm,lwd=2,col="green",ylim=ylim,ylab="T
 emperature")
lines(temp.fit.mav,lwd=2,col="red")
lines(temp.fit.gam,lwd=2,col="orange")
lines(temp.fit.loc,lwd=2,col="brown")
legend(x=1950,y=64,legend=c("MAV","LM","LOESS","GAM"),l
 ty = 1, col=c("red","green","brown","orange"))
```

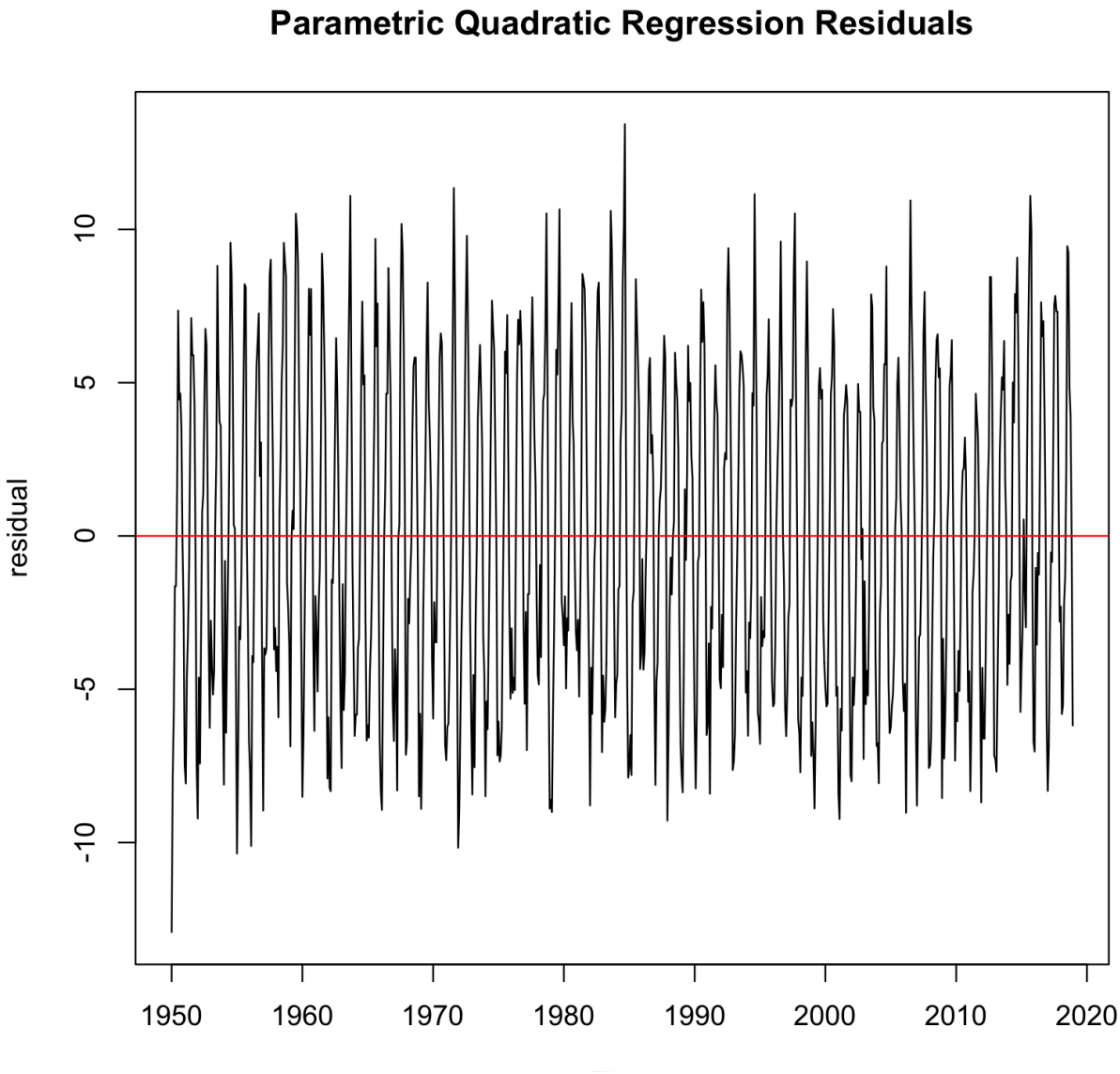


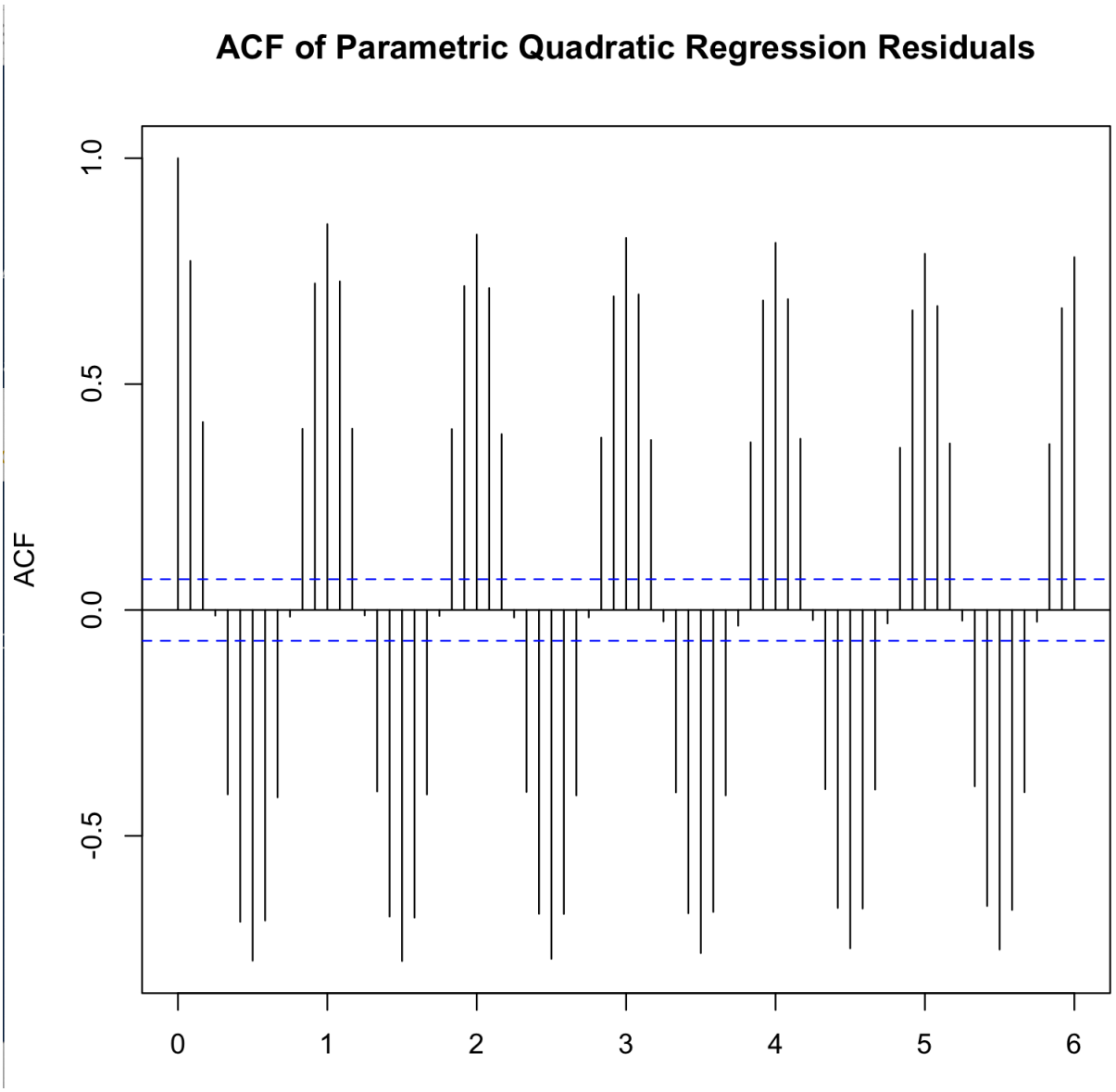

```
# Plotting Residuals with respect to time and ACF of
 residuals

residual = temp-temp.fit.mav
ts.plot(residual, ylab = "Temperature",  main= "Moving
 Average Residuals")
abline(a=mean(residual),b=0,col='red')
acf(residual,main='ACF of Moving Average Residuals',
 lag.max = 12*6)
```


Moving Average Residuals
Temperature
10
5
0
-5
-10
1950
1960
1970
1980
1990
2000
2010
2020

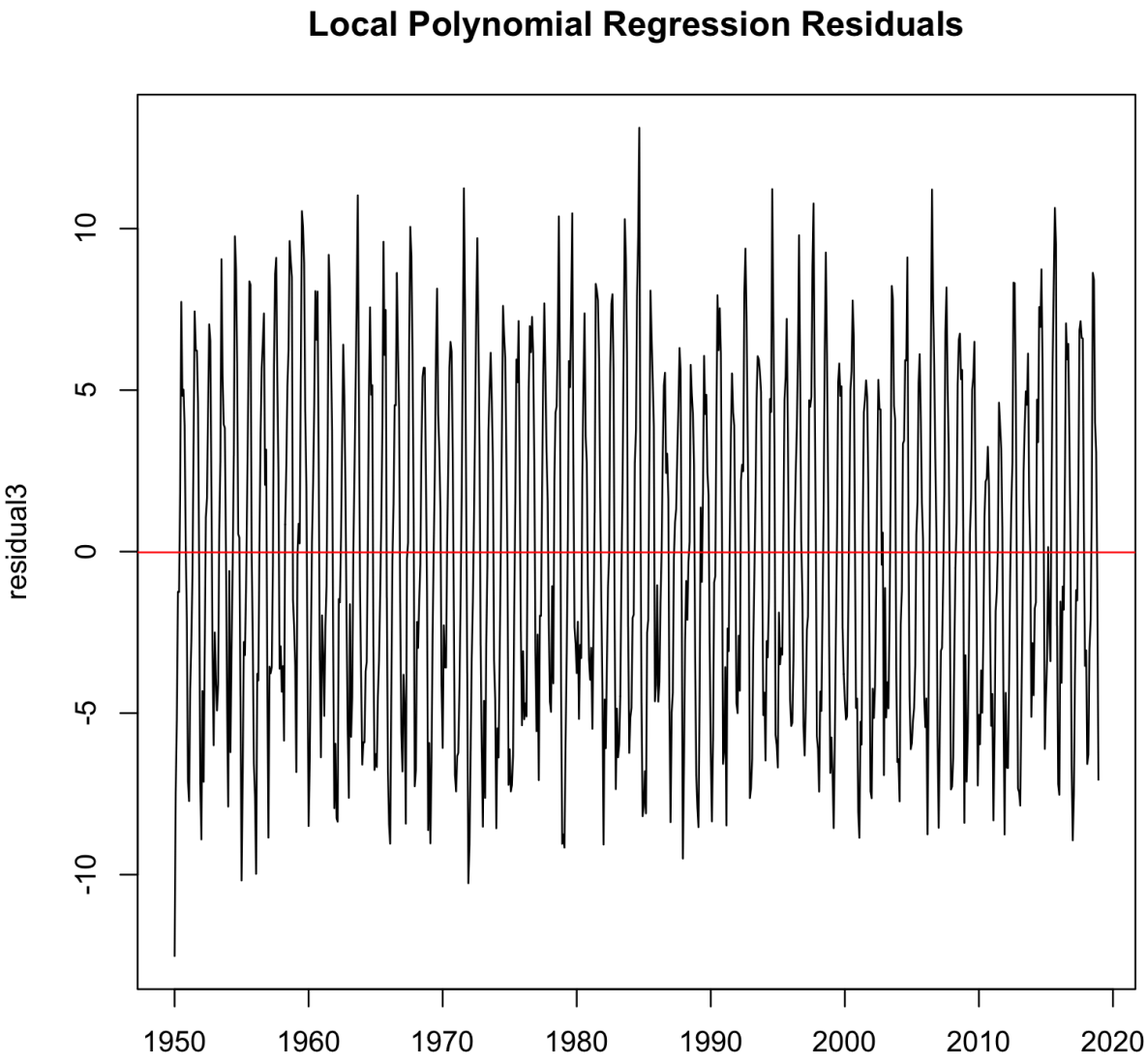


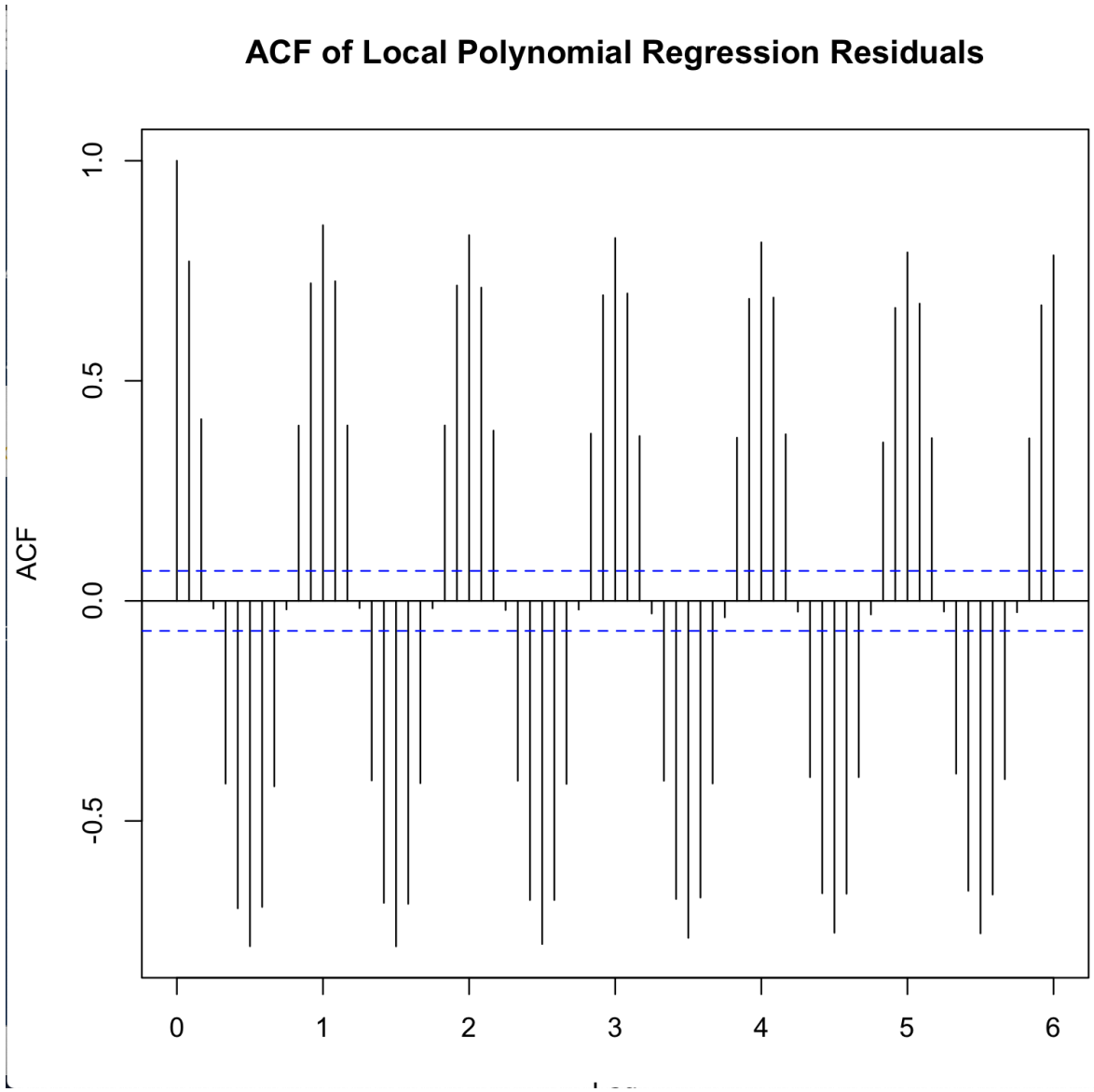

```
residual = ts(residuals(lm.fit), start = 1950, freq =
 12)
ts.plot(residual, main = "Parametric Quadratic
 Regression Residuals")
abline(a=mean(residual), b= 0, col = "red")
acf(residual, main = "ACF of Parametric Quadratic
 Regression Residuals", lag.max= 12*6)
```


Parametric Quadratic Regression Residuals
residual
10
5
0
-5
-10
1950
1960
1970
1980
1990
2000
2010
2020

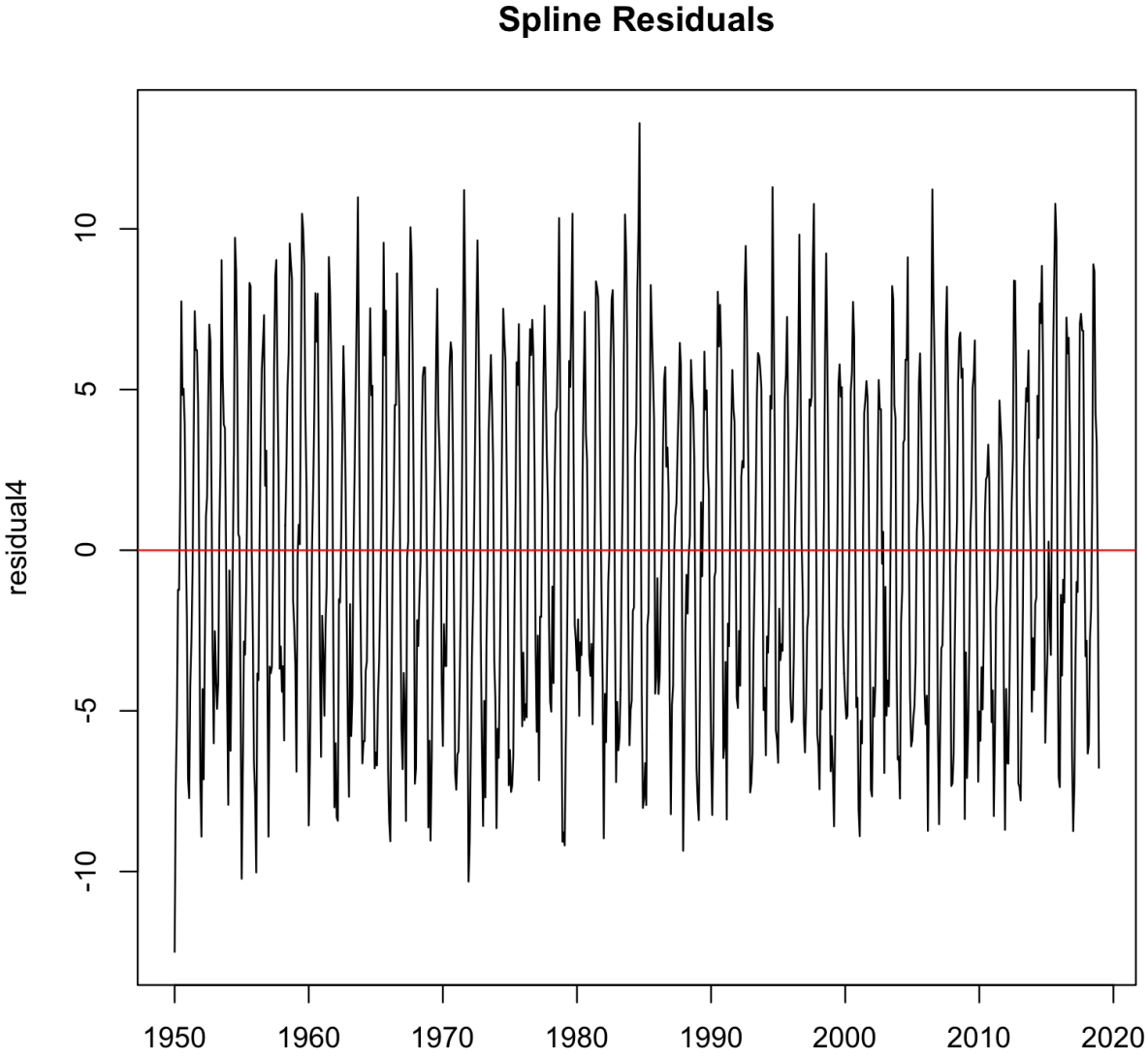


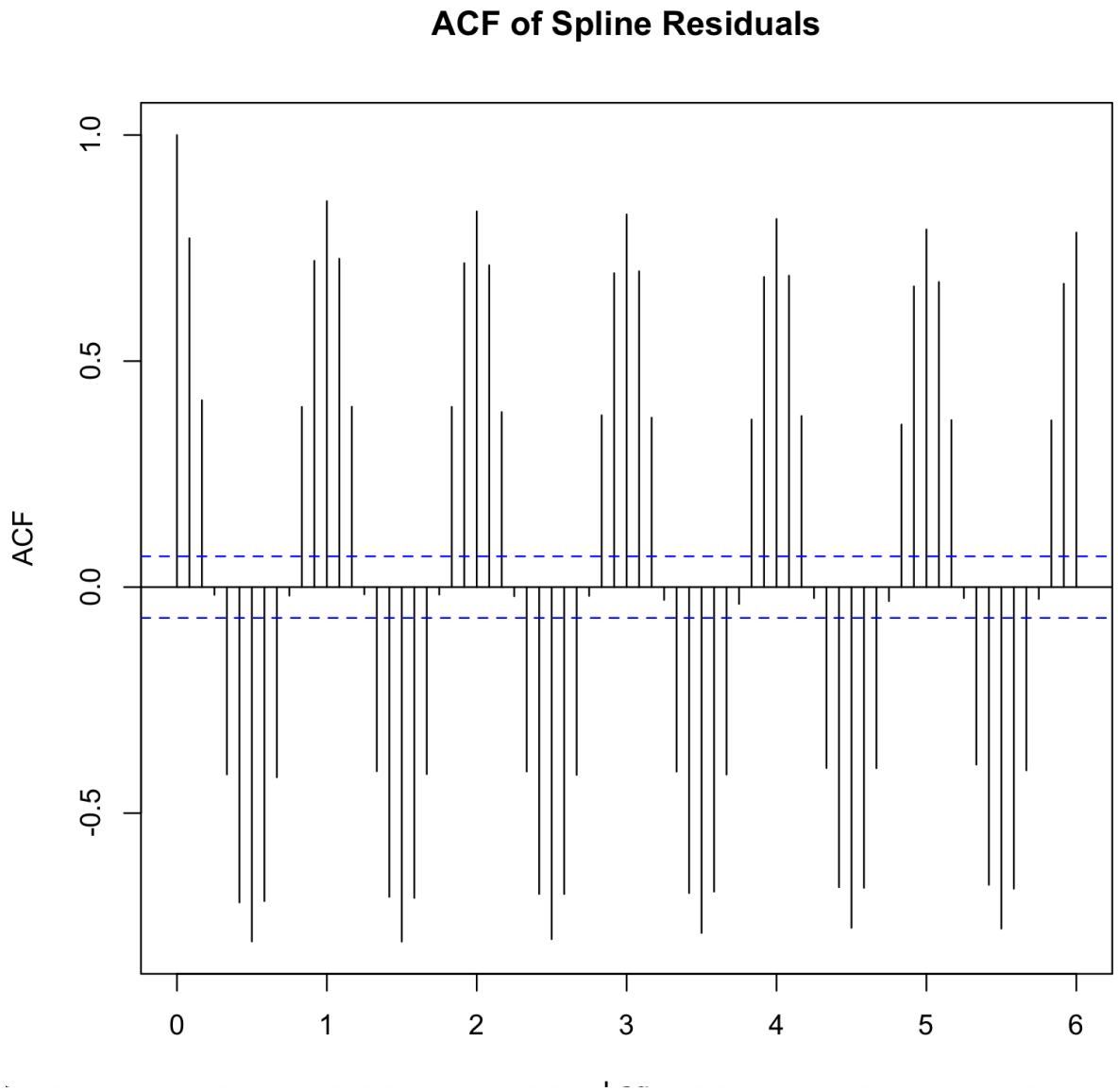

```
residual = ts(residuals(loc.fit), start = 1950, freq =
 12)
ts.plot(residual, main = "Local Polynomial Regression
 Residuals")
abline(a=mean(residual), b= 0, col = "red")
acf(residual, main = "ACF of Local Polynomial
 Regression Residuals", lag.max= 12*6)
```


Local Polynomial Regression Residuals
residual3
10
5
0
-5
-10
1950
1960
1970
1980
1990
2000
2010
2020



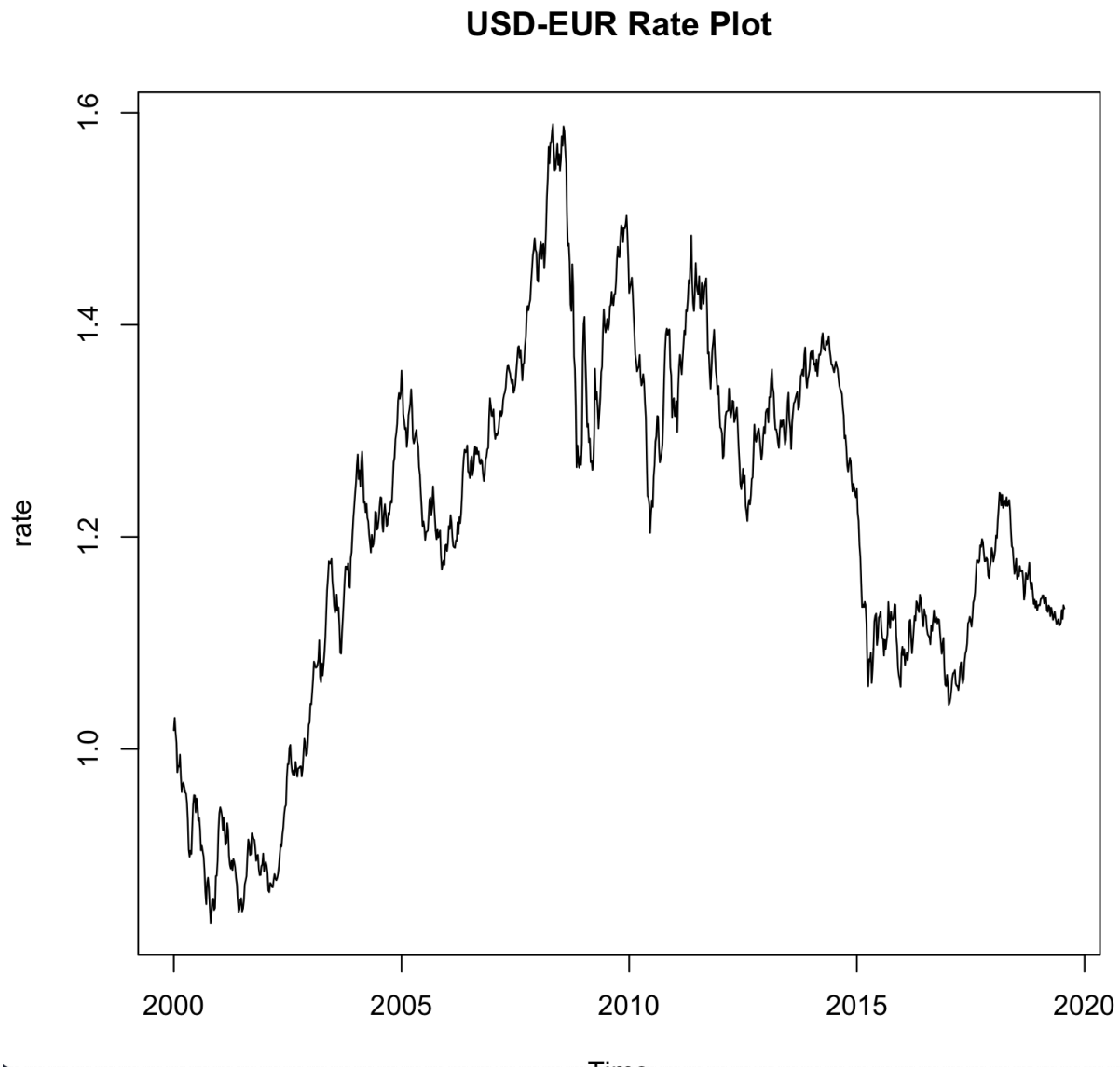

```
residual = ts(residuals(gam.fit), start = 1950, freq =
 12)
ts.plot(residual, main = "Spline Residuals")
abline(a=mean(residual), b= 0, col = "red")
acf(residual, main = "ACF of Spline Residuals",
 lag.max= 12*6)
```


Spline Residuals
residual4
10
5
0
-5
-10
1950
1960
1970
1980
1990
2000
2010
2020



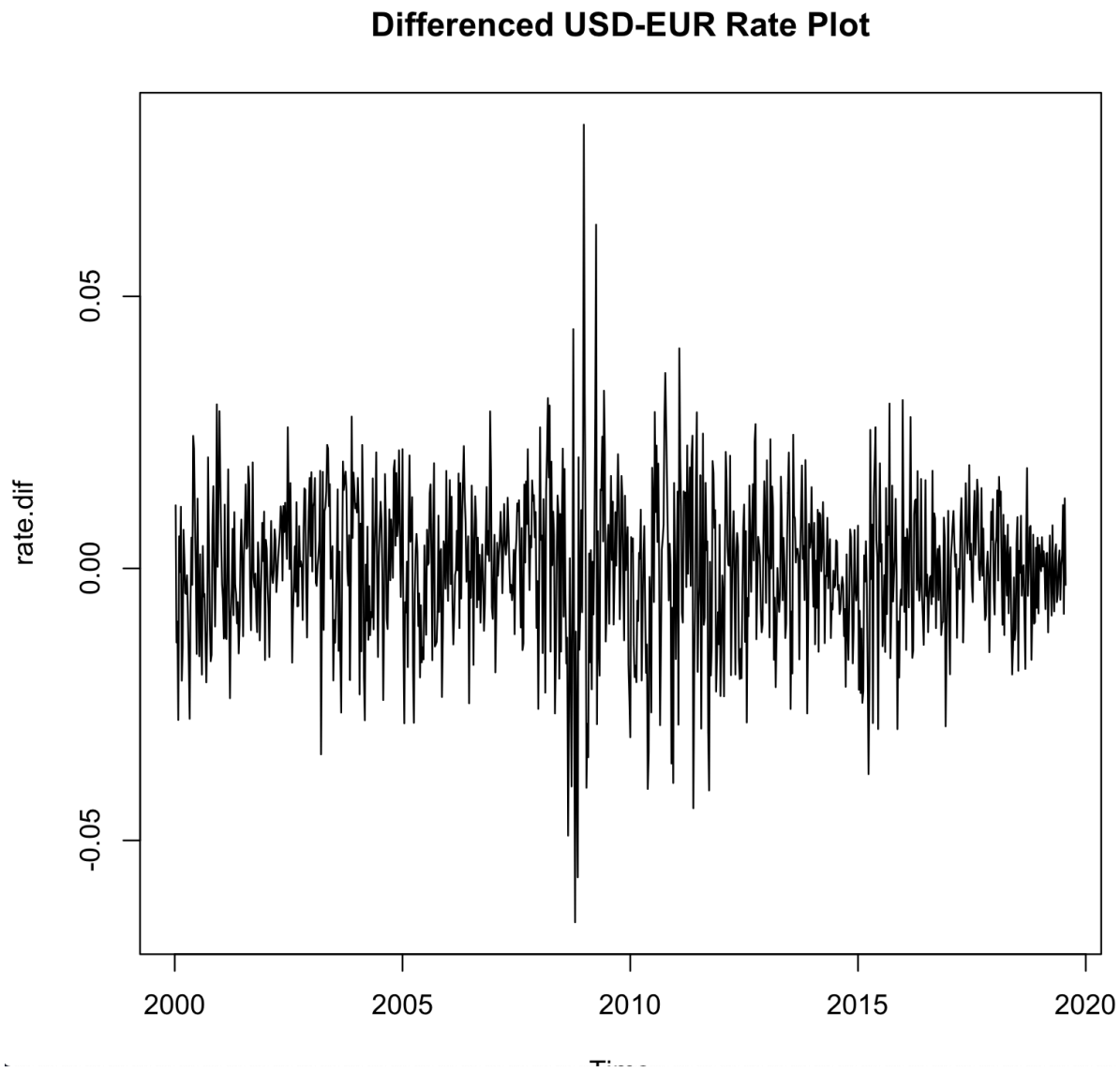
It is seen from the residual plots of the time series in each model that 1) the mean is stationary as well as 2) the variance. As for 3) the time independent autocovariance, we are uncertain by given graphs but from the ACF plots of the residuals from each model we can clearly see a high seasonality in the time series, which implies a lack of stationarity.

```
# Question 1c

<Blank>


# Question 2

library(TSA)
library(mgcv)

#Choose the 'USD-EUR Exchange.csv' dataset, wherever it
 is located on your computer
# fname <- file.choose()
#Load data
data2 <- read.csv("~/Desktop/USD_EUR_Exchange.csv",
 header = FALSE)
rateData <- data2[2:1019, 2]
rateData <- as.numeric(levels(rateData))[rateData] #
 tempData is in factor. Needs conversion to numeric.
 Super important

#Convert to TS data in proper frame
rate <- ts(rateData, start = c(2000, 1), freq = 52)
#Generate differenced data
rate.dif <- diff(rate)

# Question 2a
plot(rate, main = "USD-EUR Rate Plot")
# Exchange rate of USD-EUR seems to increase until 2010
 since 2000 and decreases around 2015 with a small re-
 increase around 2017~2018. # Constant mean is violated.
 Finite variance is violated.
```

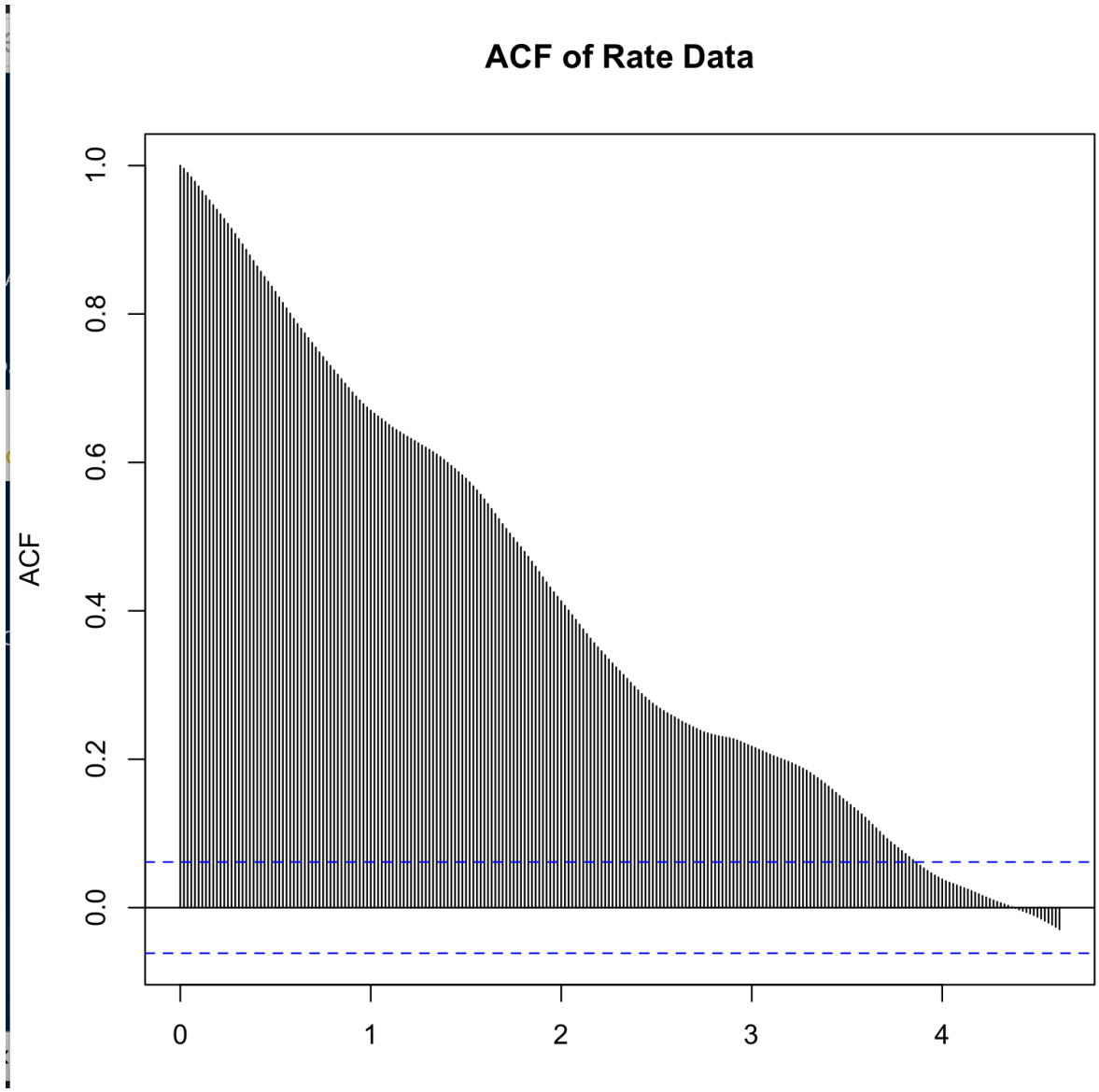



```
plot(rate.dif, main = "Differenced USD-EUR Rate Plot")
# Constant mean is not violated as the differenced rate
 is staying around 0. Though during 2009 the variance
 spike unusually high, variance is usually constant
 throughout the 20 years high.
# The differenced rate's plot seems to follow a
 constant mean and somewhat constant variance, which is
 a good sign for stationarity. Also, the differenced
 rate data ACF converges to zero relatively quickly
 compared to the non-differenced, which is a trait of
```

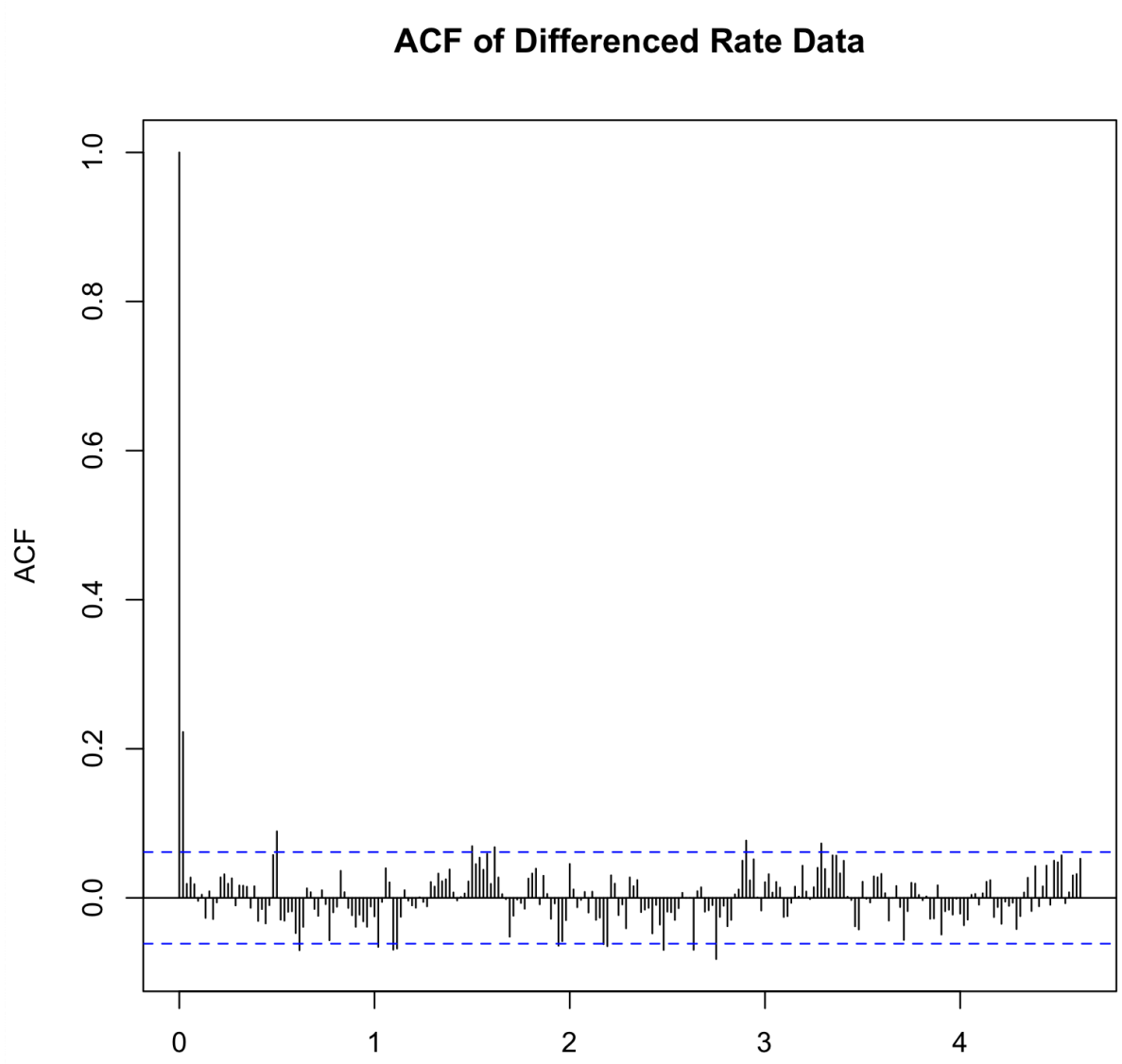
stationarity. Thus, the differenced data is a better candidate for use in analysis as we can assume stationarity.




```
acf(x= rate, lag.max = 12*20,main = "ACF of Rate Data")
acf(x= rate.dif, lag.max = 12*20, main = "ACF of
 Differenced Rate Data")
```



X-axis is lag. The ACF does not converge to 0 as lag increases which means the time series is far from stationary.



X-axis is lag. There is no clear seasonality shown in the graph. Some stationarity is shown in the residual process as ACF coverages to 0 relatively quickly as lag increases.

Question 2b

TSA package repository was having issues during download in my second laptop. I was unable to produce

plots in this section to insert in my homework. The codes are as follows:

```
time.pts = c(1:length(rate))
# set time vector similar to question 1
time.pts = c(time.pts - min(time.pts))/max(time.pts)

# Seasonality estimation using ANOVA approach

week = season(rate)
model1 = lm(rate~week)
summary(model1)
model2 = lm(rate~week-1)
summary(model2)

x1 = time.pts
x2 = time.pts^2

har2=harmonic(rate,2)

# Parametric quadratic regression fit
lm.fit = lm(rate~x1+x2+har2)
dif.fit.lm = ts((rate-
 fitted(lm.fit)),start=2000,frequency=52)
ts.plot(dif.fit.lm,ylab="Residual Process")

# Spline fit
gam.fit = gam(rate~s(time.pts)+har2)
dif.fit.gam = ts((rate-
 fitted(gam.fit)),start=2000,frequency=52)
ts.plot(dif.fit.gam,ylab="Residual Process")

# Compare
ts.plot(dif.fit.lm,ylab="Residual Process",col="brown")
```

```r
lines(dif.fit.gam,col="blue")

acf(rate,lag.max=1000,main="")
acf(dif.fit.lm,lag.max= 1000,main="")
acf(dif.fit.gam,lag.max= 1000,main="")


# Question 2c

<Blank>
```