Project1

Jonathan Domingue

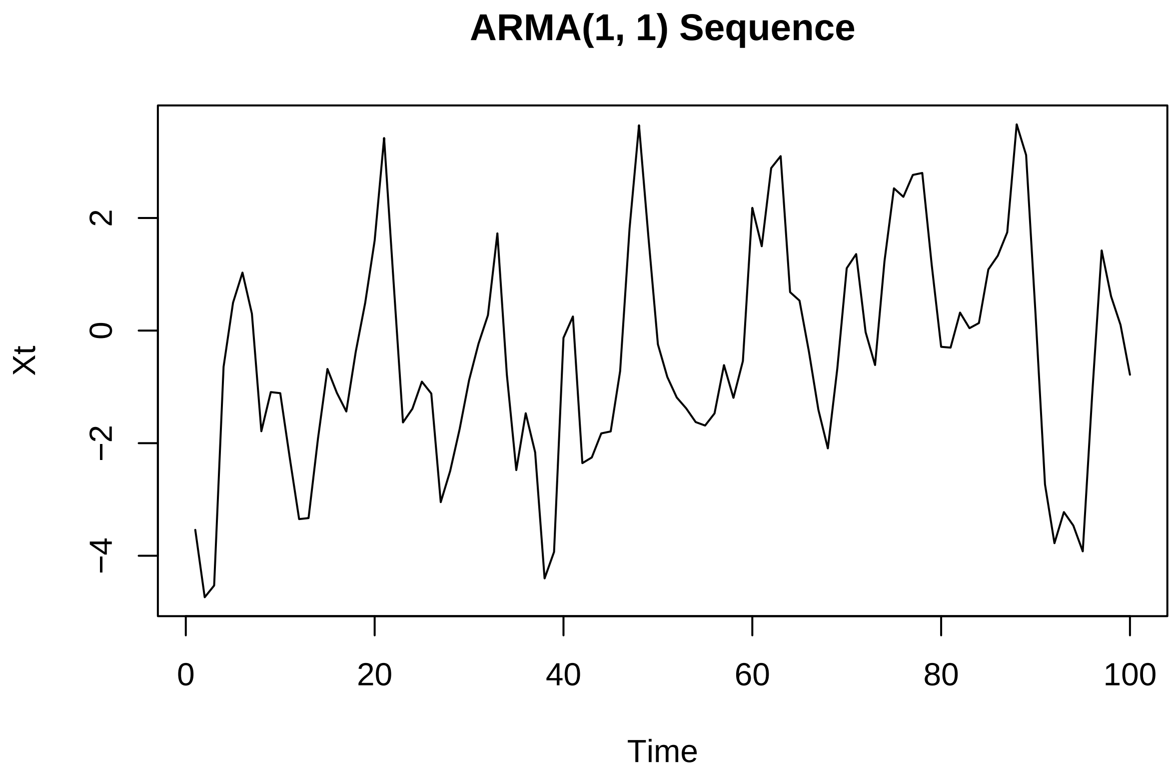
2024-02-24

Question 1

a)Generating ARMA(p,q)

```
MA<-function(n,q,theta=c(rep(1,q))){
IID=rnorm(n+q);
Data=c(rep(0,n+q));
for(i in (q+1):(n+q)){
Data[i]=IID[i]+sum(theta*IID[(i-1):(i-q)])
}
Data=Data[(q+1):(n+q)]
}
```

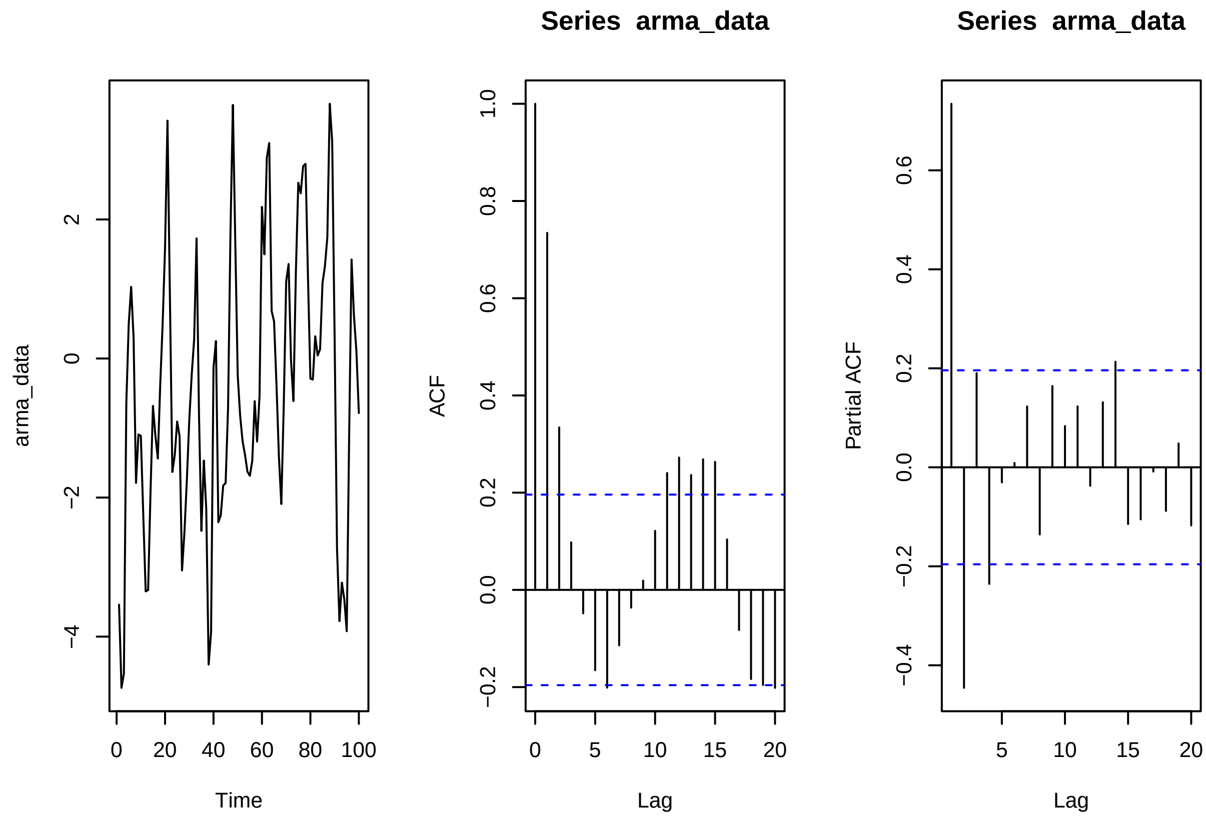
```
ARMA<-function(n,p,q,phi=c(rep(0.1,p)),sigma=1){
m=floor(n/10);
MAnoise=MA(n+m,q);
Data=c(); Data[1:p]=0; begin=p+1; end=n+m;
for(i in begin:end){
Data[i]=sum(phi*Data[(i-p):(i-1)])+MAnoise[i];
}
Data=Data[(m+1):(n+m)];
}

# Parameters for ARMA sequence
n = 100 # Number of observations
p = 1 # AR order
q = 1 # MA order
phi = c(0.5) # AR parameters
theta = c(-0.5) # MA parameters used in the MA function inside ARMA


arma_data = ARMA(n, p, q, phi)

# Plot the generated ARMA sequence
plot.ts(arma_data, main="ARMA(1, 1) Sequence", ylab="Xt")
```




```
#MyTimeSeries2=ARMA(1000,1,2); # phi=0.1 by default
#plot.ts(MyTimeSeries2)
```

b)Identifying the model using ACF and PACF

```
# Plotting the two pictures together
par(mfrow=c(1,3));
plot.ts(arma_data);
acf(arma_data);
pacf(arma_data)
```



Exponential Smoothing

For the Exponential smoothing, we will be using the function that we previously defined:

```
# Define the exponential smoothing function
ExpSmooth <- function(x, alpha) {
  n <- length(x)
  Data <- c(rep(0, n))
  Data[1] <- x[1]
  for (i in 2:n) {
    Data[i] <- alpha * x[i] + (1 - alpha) * Data[i-1]
  }
  return(Data)
}
```

```
# Define the moving average smoothing function
MASmooth <- function(x, Q) {
  n <- length(x)
  Smooth <- c(rep(0, n))
  for (i in (Q+1):(n-Q)) {
    Smooth[i] <- mean(x[(i-Q):(i+Q)])
  }
  for (i in 1:Q) {
    Smooth[i] <- Smooth[Q+1]
  }
```

```
  for (i in (n-Q+1):n) {
    Smooth[i] <- Smooth[(n-Q)]
  }
  out <- Smooth
  return(out)
}
```

c)

Generating a Linear Trend

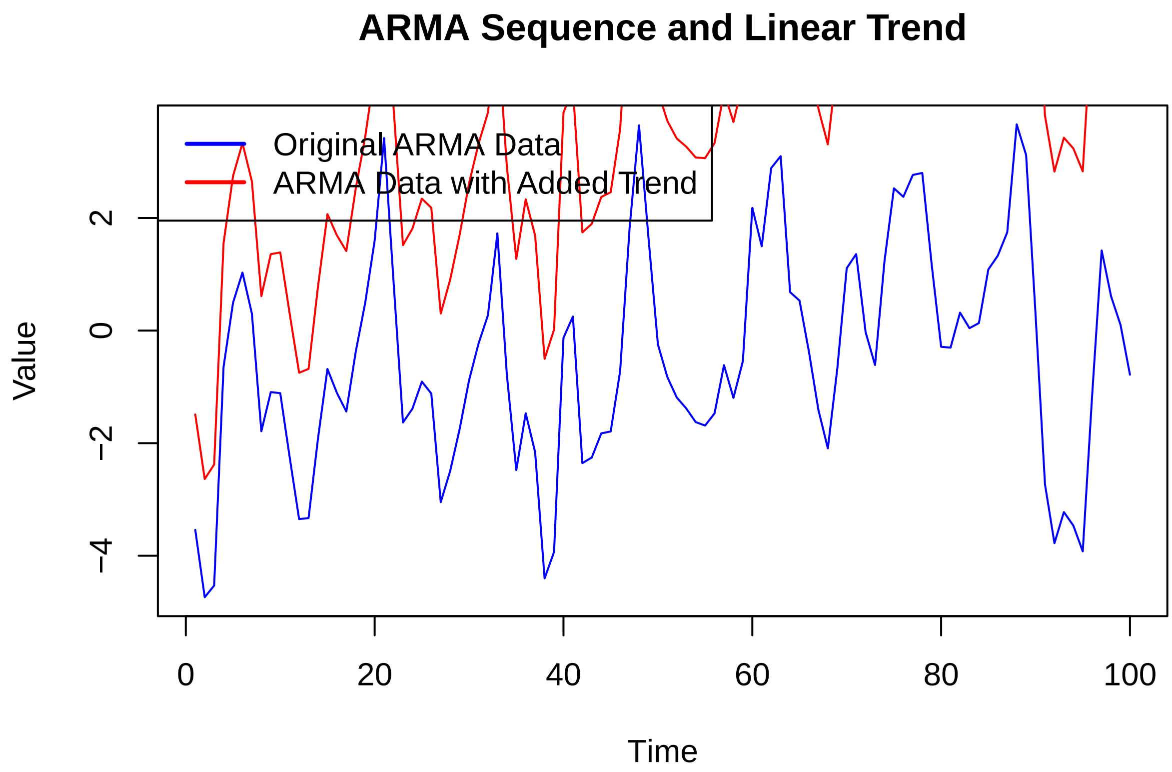
```
n <- length(arma_data)  # Number of observations in arma_data
Time <- 1:n  # Time index

# Parameters for the linear trend
a <- 2  # Intercept of the linear trend
b <- 0.05  # Slope of the linear trend

# Generate the linear trend
linear_trend <- a + b * Time
```

Adding the Linear Trend to the ARMA Sequence

```
# Add the linear trend to the ARMA sequence
Yt <- arma_data + linear_trend
```

Visualization

```
# Plot the original ARMA sequence
plot(Time, arma_data, type="l", col="blue", main="ARMA Sequence and Linear Trend", ylab="Value", xlab="
# Add the new sequence with the trend
lines(Time, Yt, col="red")
legend("topleft", legend=c("Original ARMA Data", "ARMA Data with Added Trend"), col=c("blue", "red"), l
```



d) Estimate mt using all three Methods

Parametric Approach using a Simple Linear Regression

```r
# Estimate the trend using linear regression
linear_model <- lm(Yt ~ Time)
mt_parametric <- linear_model$fitted.values
```

Exponential Smoothing

```r
# Exponential smoothing to estimate the trend
alpha <- 0.1  # Smoothing parameter
mt_exp <- ExpSmooth(Yt, alpha)
```

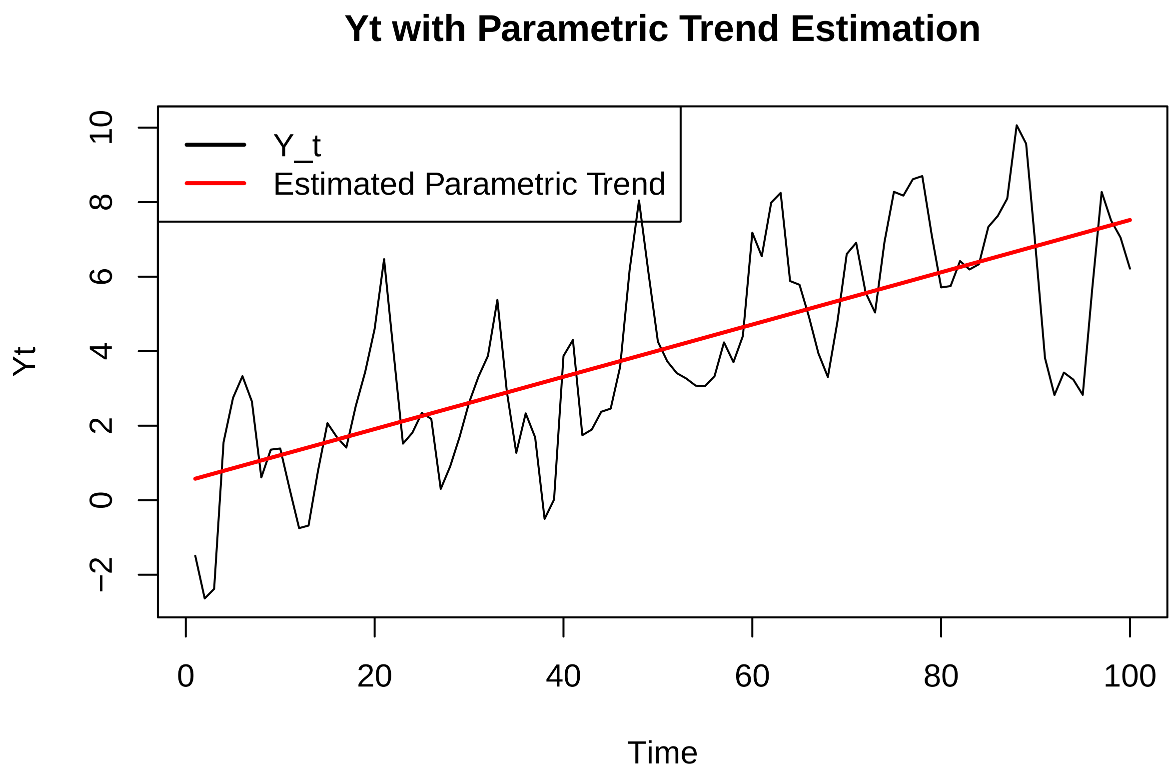
Moving Average Smoothing

```r
Q <- 5  # Window size for smoothing
mt_ma <- MASmooth(Yt, Q)
```

e) Plot Yt and the Estimated Trend on the same Graph

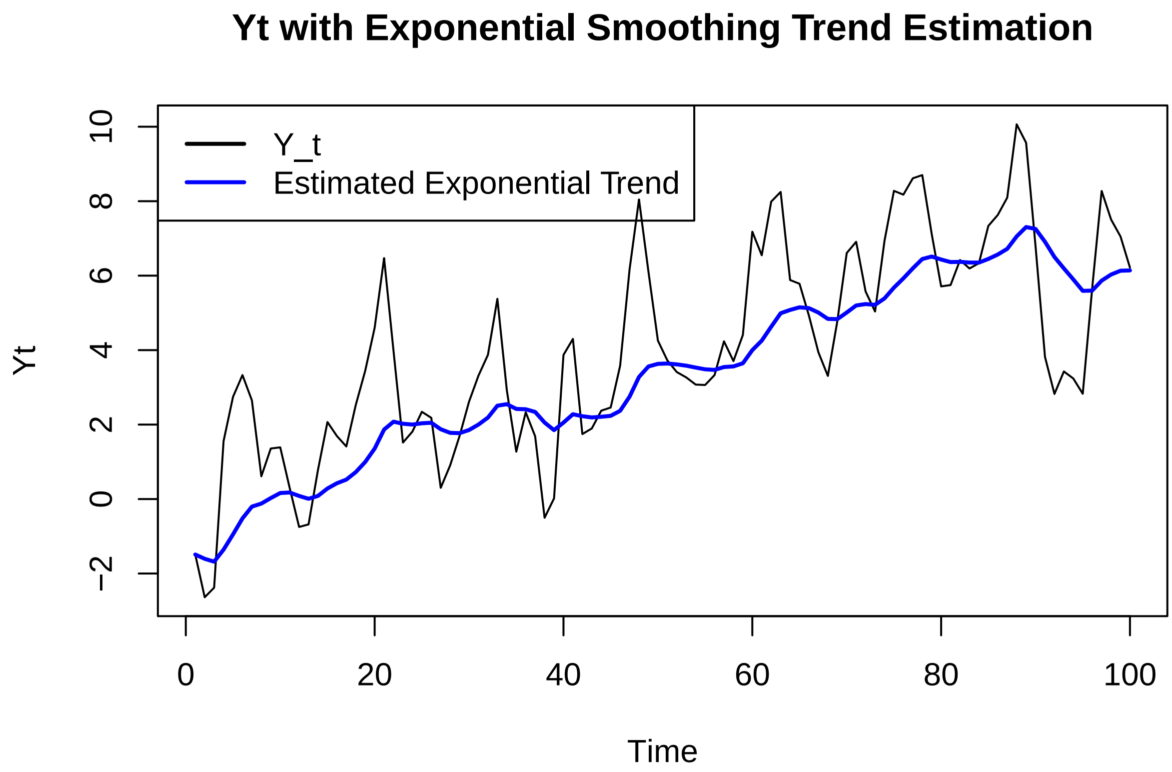
Parametric Method Plot

```
plot(Time, Yt, type="l", col="black", main="Yt with Parametric Trend Estimation")
lines(Time, mt_parametric, col="red", lwd=2)
legend("topleft", legend=c("Y_t", "Estimated Parametric Trend"), col=c("black", "red"), lty=1, lwd=2)
```



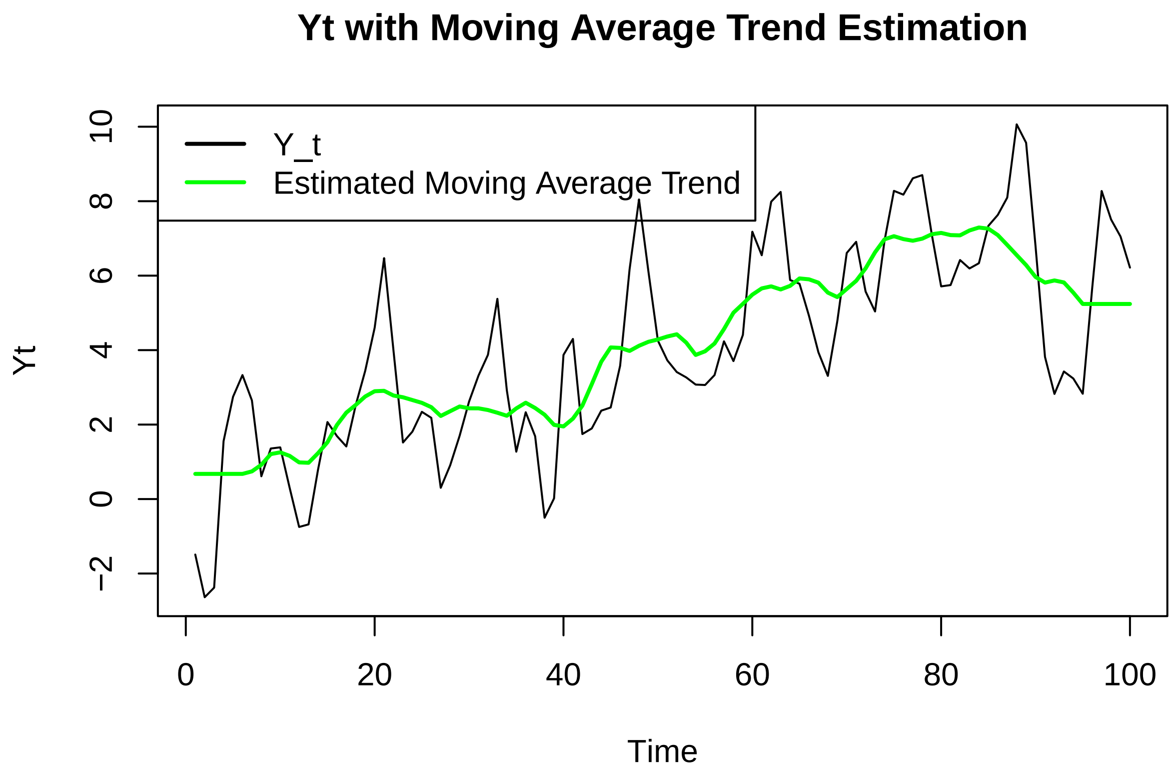
Exponential Smoothing Plot

```
plot(Time, Yt, type="l", col="black", main="Yt with Exponential Smoothing Trend Estimation")
lines(Time, mt_exp, col="blue", lwd=2)
legend("topleft", legend=c("Y_t", "Estimated Exponential Trend"), col=c("black", "blue"), lty=1, lwd=2)
```

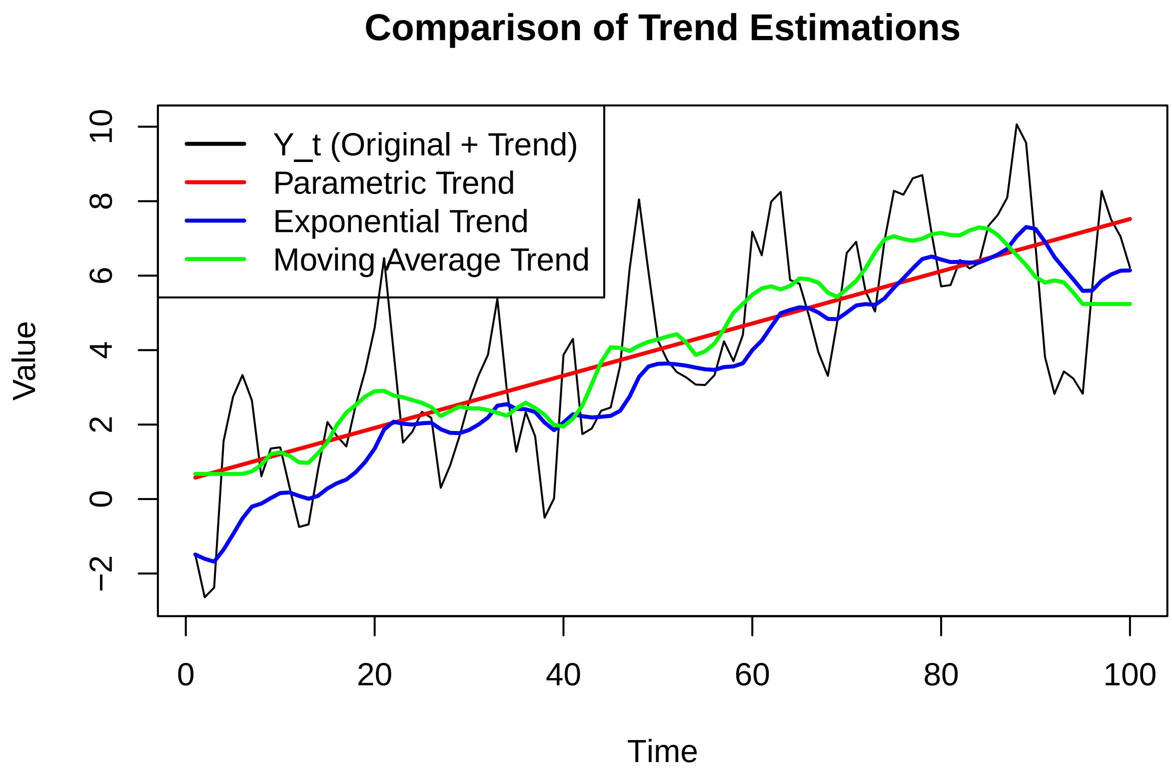


Moving Average Smoothing Plot

```
plot(Time, Yt, type="l", col="black", main="Yt with Moving Average Trend Estimation")
lines(Time, mt_ma, col="green", lwd=2)
legend("topleft", legend=c("Y_t", "Estimated Moving Average Trend"), col=c("black", "green"), lty=1, lw
```




```
# Plot the original series Yt with the added trend
plot(Time, Yt, type="l", col="black", main="Comparison of Trend Estimations", ylab="Value", xlab="Time")
# Add the estimated trend from the Parametric Method
lines(Time, mt_parametric, col="red", lwd=2)
# Add the estimated trend from Exponential Smoothing
lines(Time, mt_exp, col="blue", lwd=2)
# Add the estimated trend from Moving Average Smoothing
lines(Time, mt_ma, col="green", lwd=2)

# Add a legend to differentiate the methods
legend("topleft",
       legend=c("Y_t (Original + Trend)", "Parametric Trend", "Exponential Trend", "Moving Average Tren
       col=c("black", "red", "blue", "green"),
       lty=1, lwd=2)
```



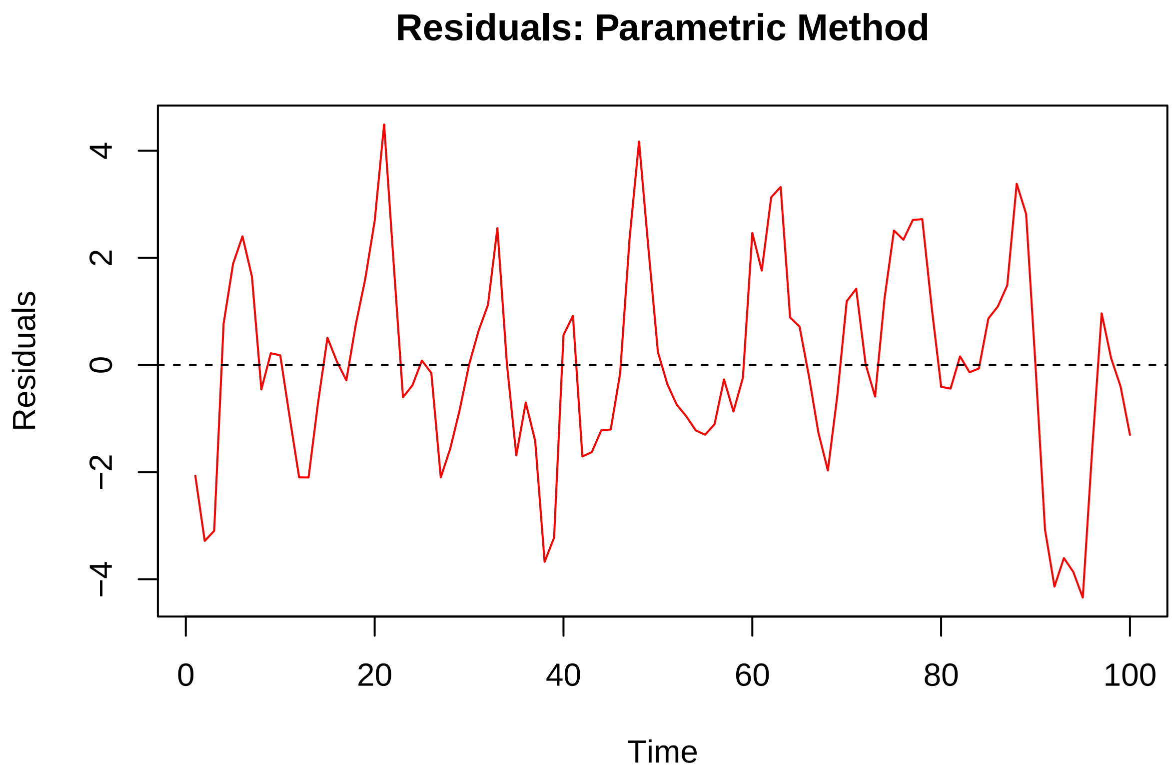
f) Estimate and plot the Residuals for the Three Methods

Computing Residuals for each Method

```
X_hat_t_parametric <- Yt - mt_parametric #Parametric Method Residuals
X_hat_t_exp <- Yt - mt_exp # Exponential Smoothing Method Residuals
X_hat_t_ma <- Yt - mt_ma # Moving Average Smoothing Residuals
```

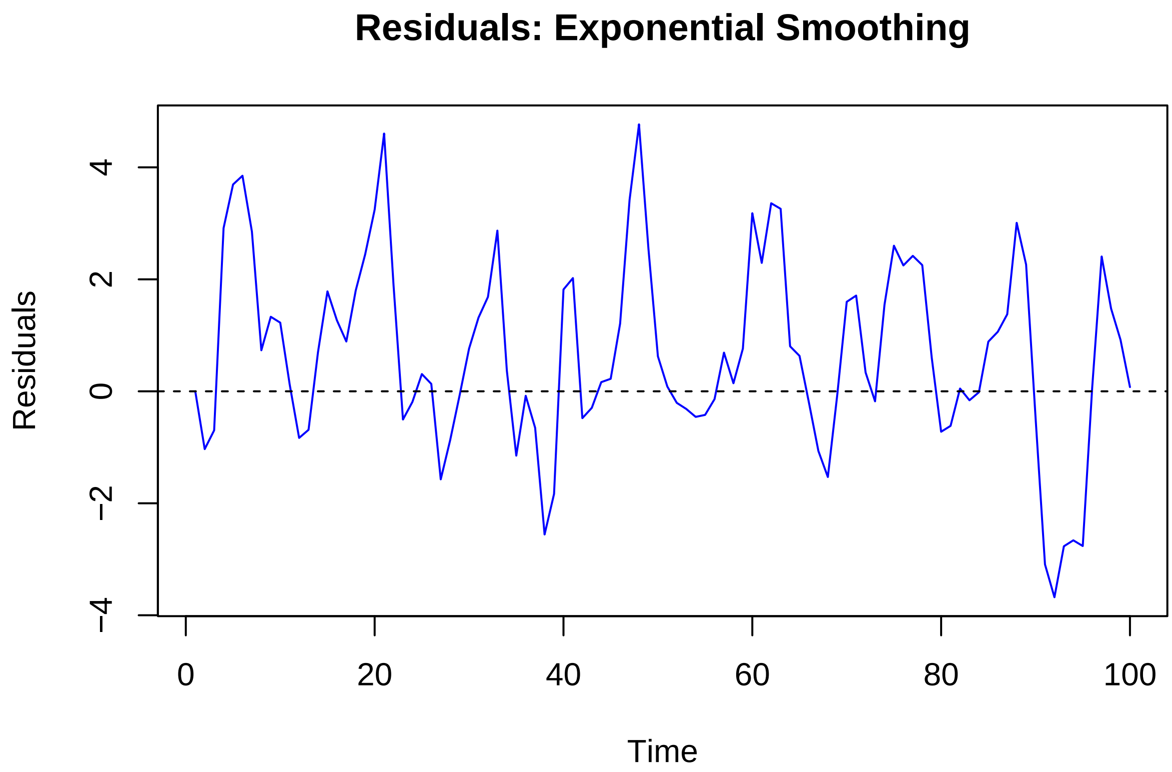
Plotting Residuals for Parametric Method

```
plot(Time, X_hat_t_parametric, type="l", col="red", main="Residuals: Parametric Method", ylab="Residual
abline(h=0, lty=2)
```



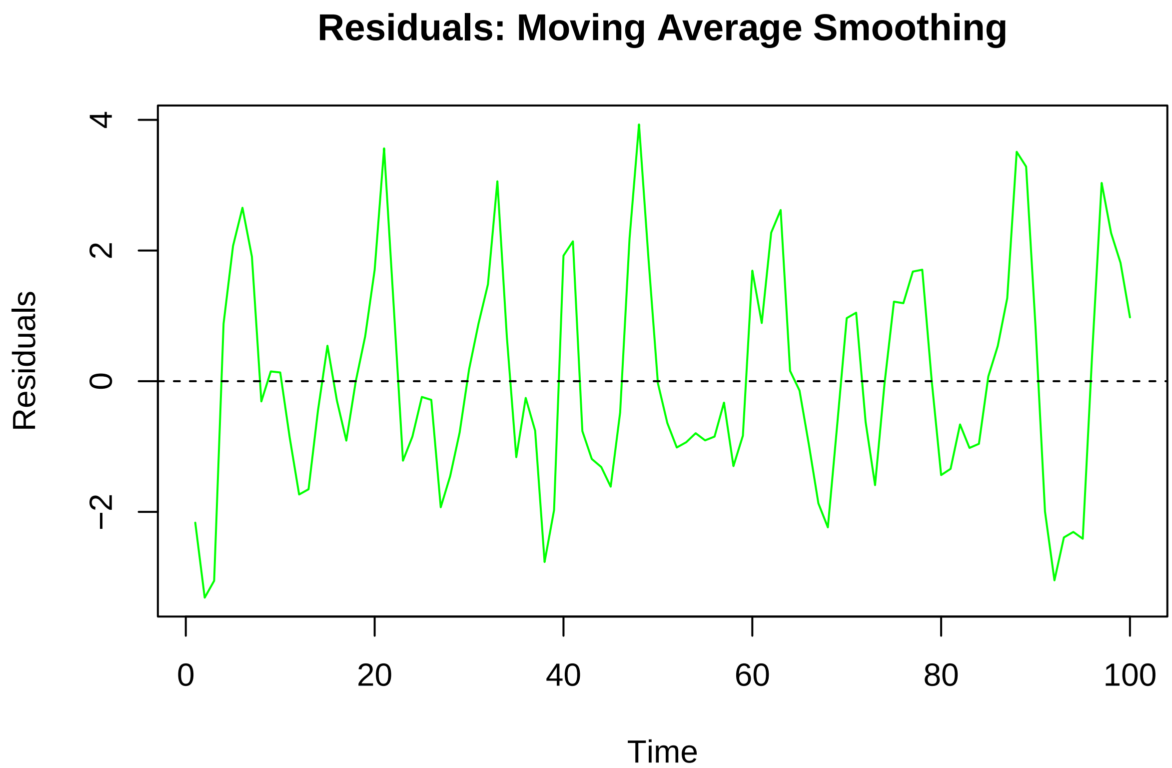
Plotting Residuals for Exponential Smoothing Method

```
plot(Time, X_hat_t_exp, type="l", col="blue", main="Residuals: Exponential Smoothing", ylab="Residuals"
abline(h=0, lty=2)
```



Plotting Residuals for Moving Average Smoothing Method

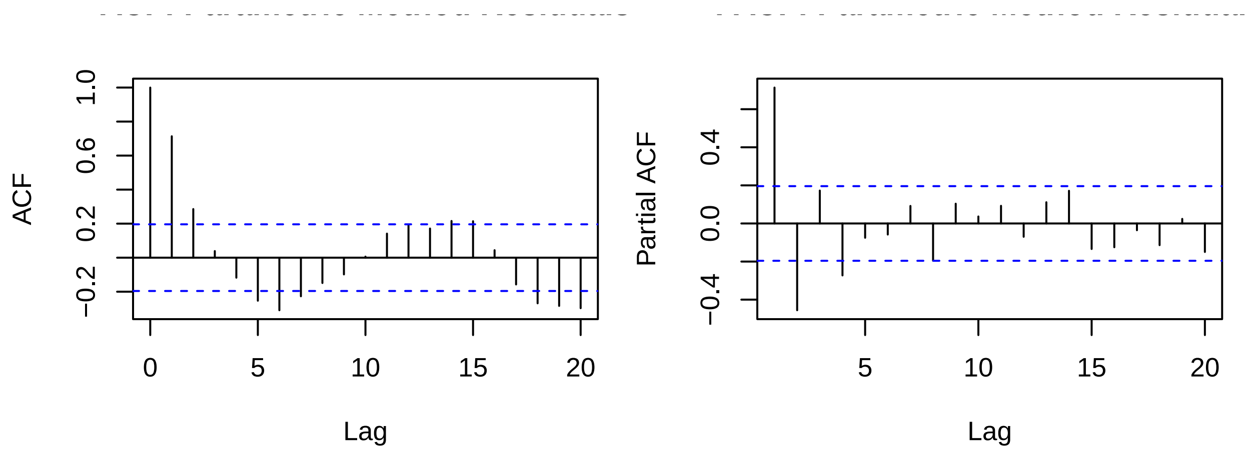
```
plot(Time, X_hat_t_ma, type="l", col="green", main="Residuals: Moving Average Smoothing", ylab="Residual
abline(h=0, lty=2)
```



g) Analysing and comparing with ACF/PACF in c)

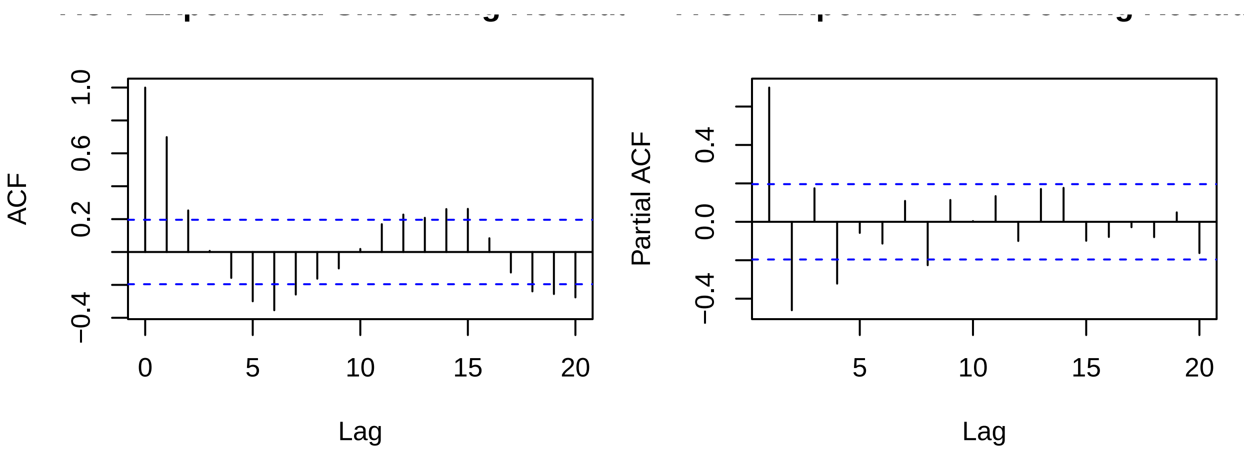
Parametric Method Residuals

```
# ACF and PACF for Parametric Method Residuals
par(mfrow=c(2,2), mar=c(4,4,2,1))
acf(X_hat_t_parametric, main="ACF: Parametric Method Residuals")
pacf(X_hat_t_parametric, main="PACF: Parametric Method Residuals")
```



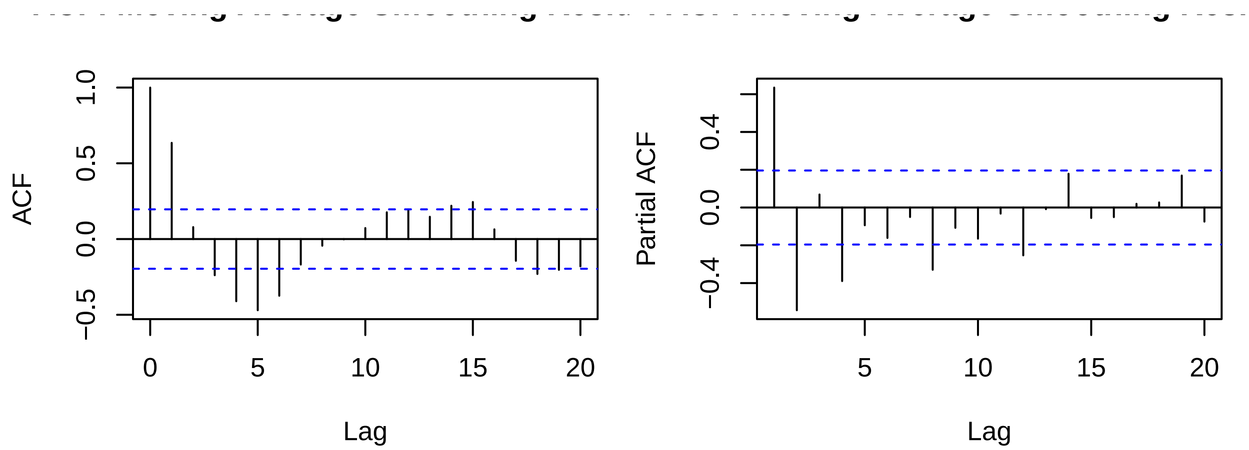
Exponential Smoothing Residuals

```r
# ACF and PACF for Exponential Smoothing Residuals
par(mfrow=c(2,2), mar=c(4,4,2,1))
acf(X_hat_t_exp, main="ACF: Exponential Smoothing Residuals")
pacf(X_hat_t_exp, main="PACF: Exponential Smoothing Residuals")
```



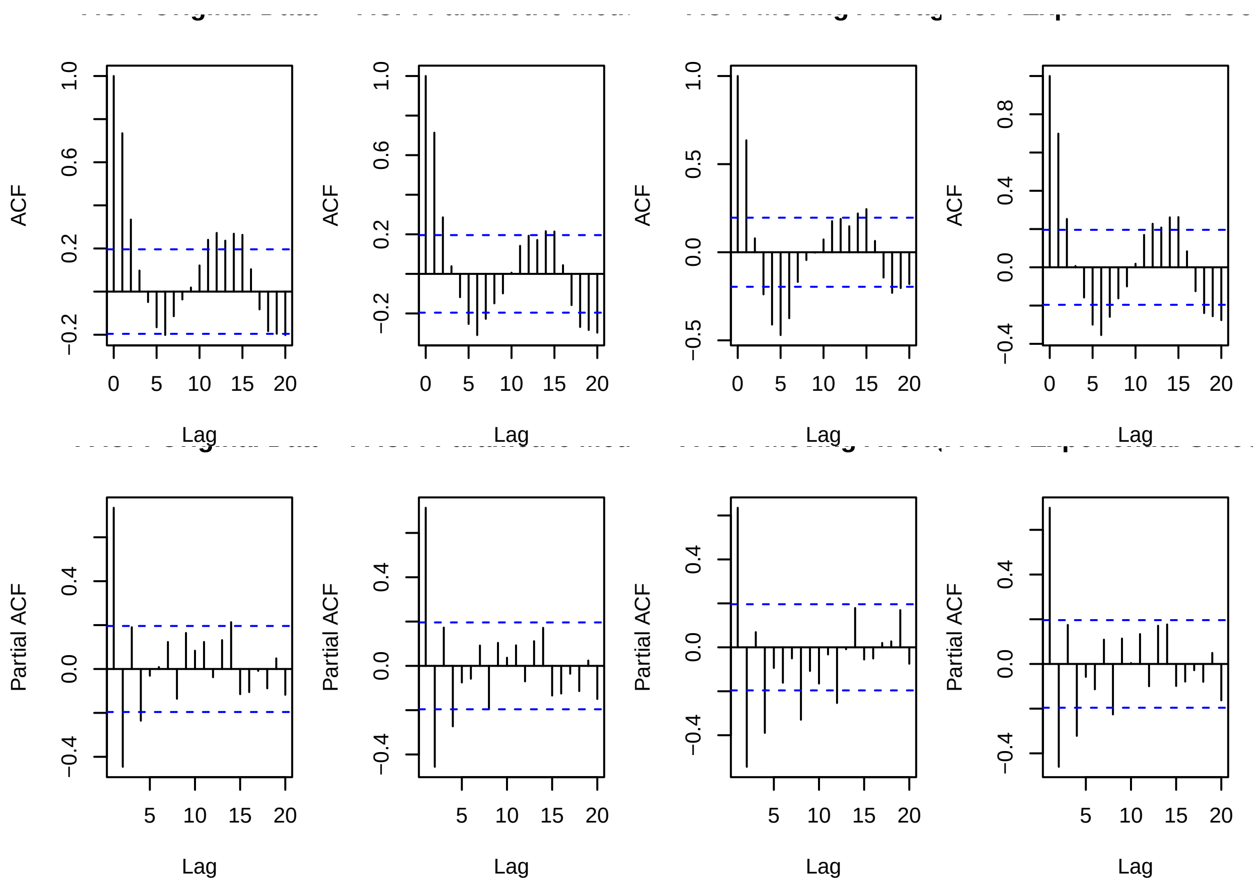
Moving Average Smoothing Residuals

```r
# ACF and PACF for Moving Average Smoothing Residuals
par(mfrow=c(2,2), mar=c(4,4,2,1))
acf(X_hat_t_ma, main="ACF: Moving Average Smoothing Residuals")
pacf(X_hat_t_ma, main="PACF: Moving Average Smoothing Residuals")
```



Comparison with original ACF and PACF

```
# Adjust layout for ACF comparison
par(mfrow=c(2, 4), mar=c(4,4,2,1)) # 2 rows for ACF & PACF, 4 columns for each series

# ACF Plots: Original and Methods
acf(arma_data, main="ACF: Original Data", lag.max=20)
acf(X_hat_t_parametric, main="ACF: Parametric Method", lag.max=20)
acf(X_hat_t_ma, main="ACF: Moving Average", lag.max=20)
acf(X_hat_t_exp, main="ACF: Exponential Smoothing", lag.max=20)

# PACF Plots: Original and Methods
pacf(arma_data, main="PACF: Original Data", lag.max=20)
pacf(X_hat_t_parametric, main="PACF: Parametric Method", lag.max=20)
pacf(X_hat_t_ma, main="PACF: Moving Average", lag.max=20)
pacf(X_hat_t_exp, main="PACF: Exponential Smoothing", lag.max=20)
```




```r
# Reset plot parameters to default after plotting
par(mfrow=c(1, 1), mar=c(5,4,4,2) + 0.1)
```

Conclusion: Upon comparing the ACF and PACF plots from the original time series data with those obtained after applying various smoothing techniques, we can clearly see that s that the underlying auto-correlation structure of the data has been mostly preserved.

Minor discrepancies observed with the moving average smoothing, but they are still within acceptable limits, indicating that the trend removal has been effective without significantly distorting the original ARMA characteristics of the series.

Question 2

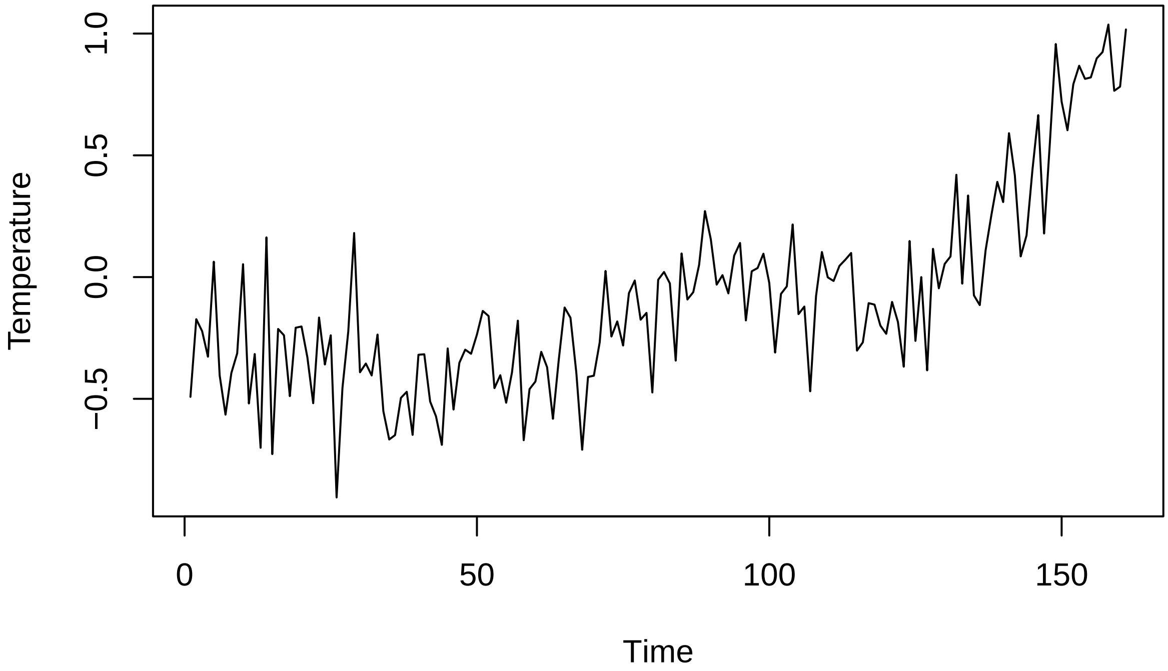
A) Downloading a dataset

The NorthernHemisphere dataset was selected.

```r
Temperature=scan("NorthernHemisphere.txt")
```

Data Visualization

```r
plot.ts(Temperature)
```



The sequence is clearly not stationary. We need to remove trend. To remove trend, I will apply exponential smoothing. Exponential Smoothing seems to be a good choice from the plot that we can observe

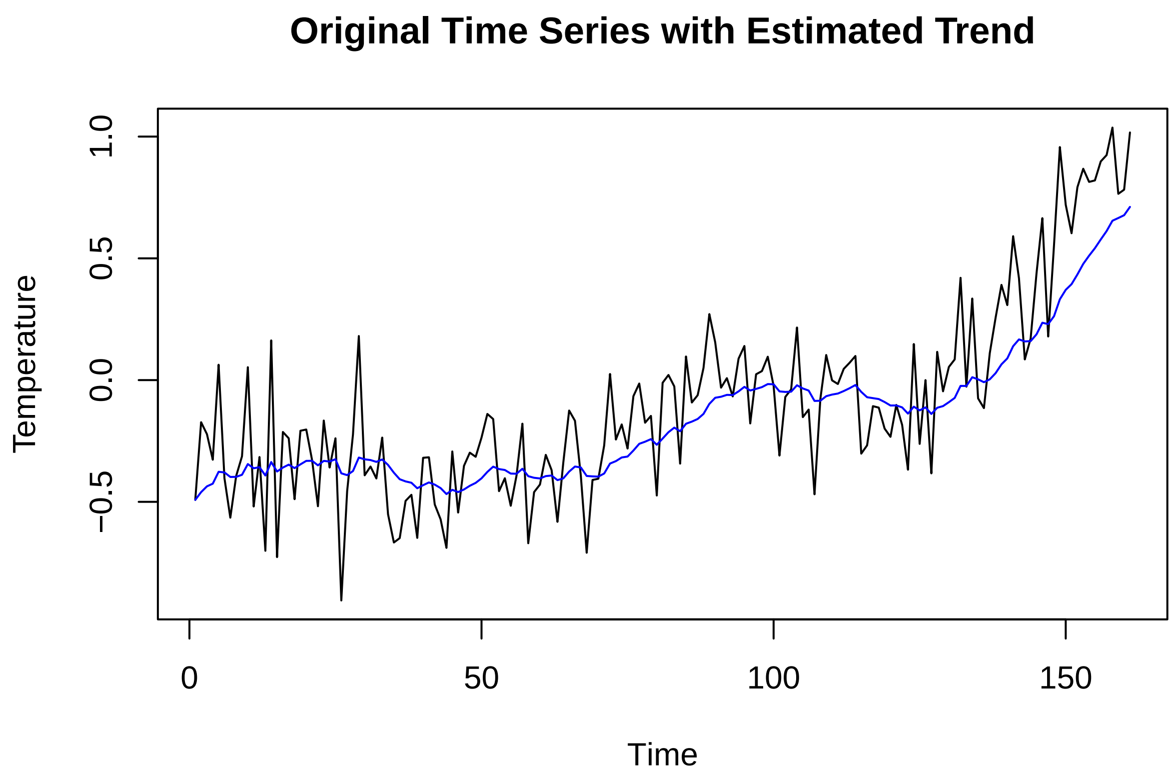
```r
ExpSmooth<-function(x,alpha){
#x- data;
# alpha - smoothing parameter;
n=length(x);
Data=c(rep(0,n));
Data[1]=x[1];
for(i in 2:n){
Data[i]=alpha*x[i]+(1-alpha)*Data[i-1]
};
out<-Data;
}
```

b) Removing the Trend to Obtain a Stationary Time Series

```r
# Using alpha = 0.1 as it provides the most smoothing
alpha <- 0.1
MySmoothedTS <- ExpSmooth(Temperature, alpha)
TS_1 <- Temperature - MySmoothedTS
```

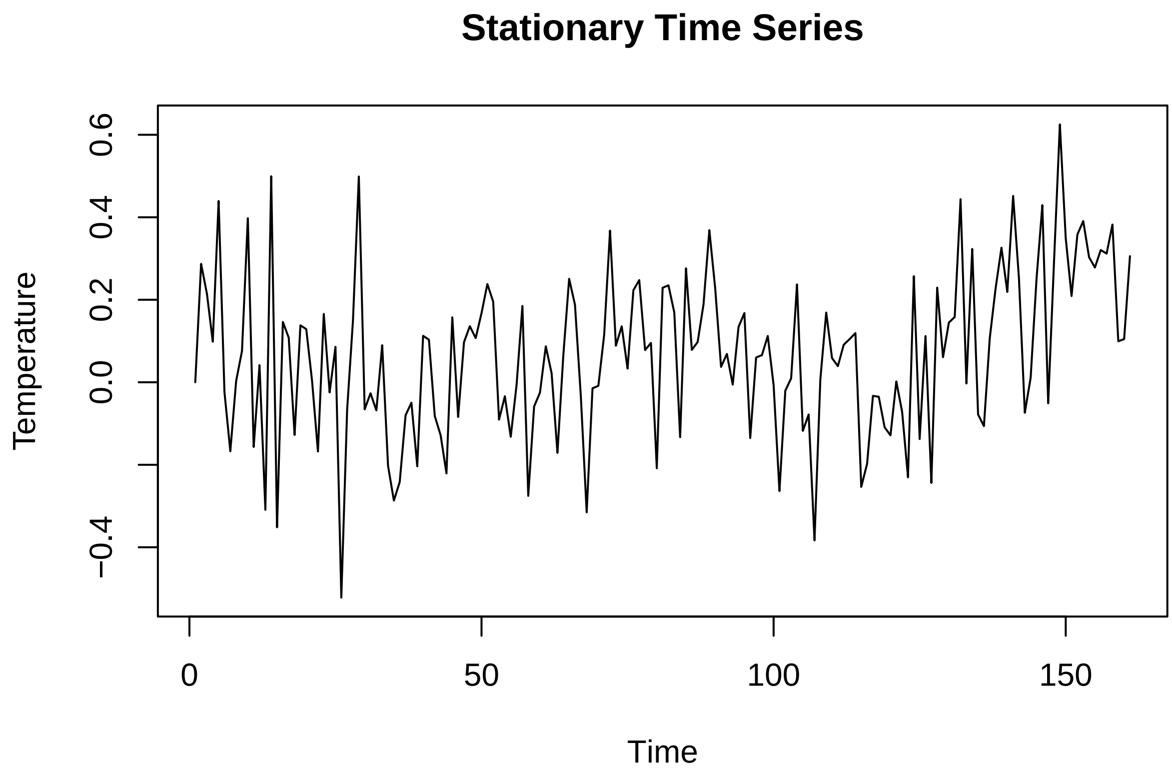
C) Plotting Together

```r
# Plot the original time series with the estimated trend
plot.ts(Temperature, main = "Original Time Series with Estimated Trend", ylab = "Temperature")
lines(MySmoothedTS, col = "blue", type = "l")  # Add the smoothed trend line
```

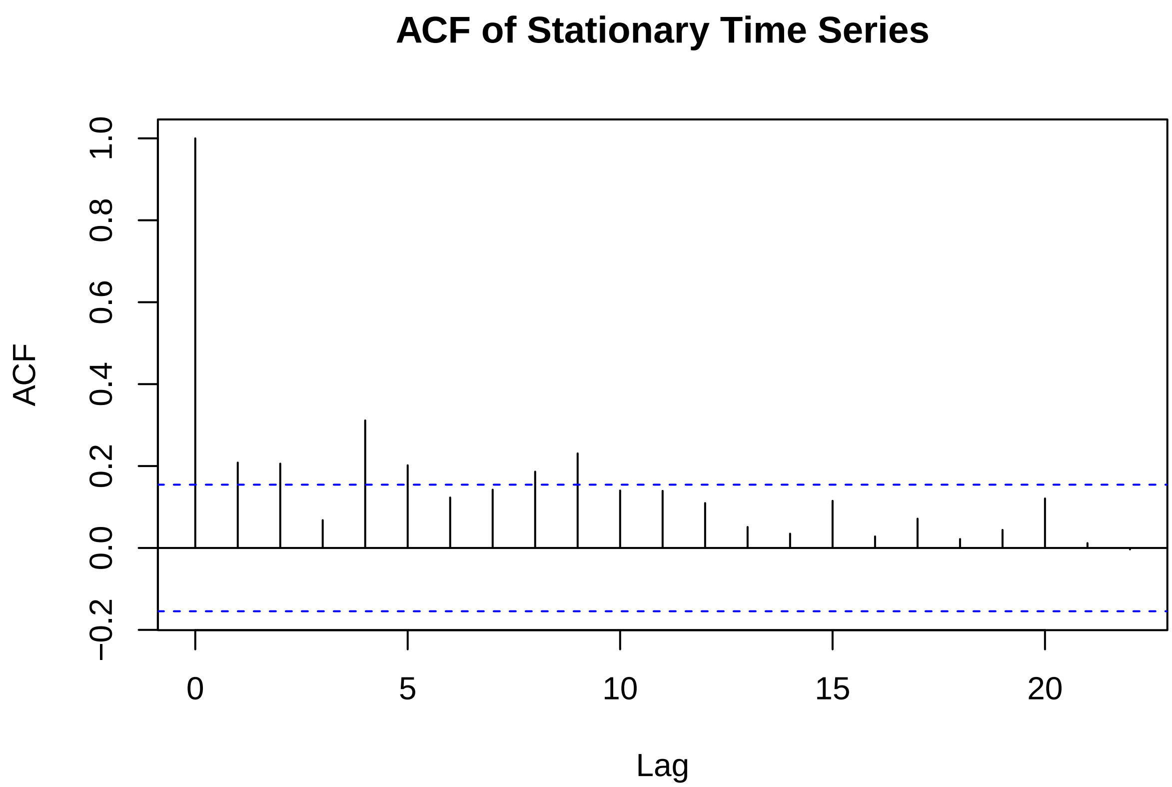


d) Plot the Stationary Part, then ACF and then the PACF

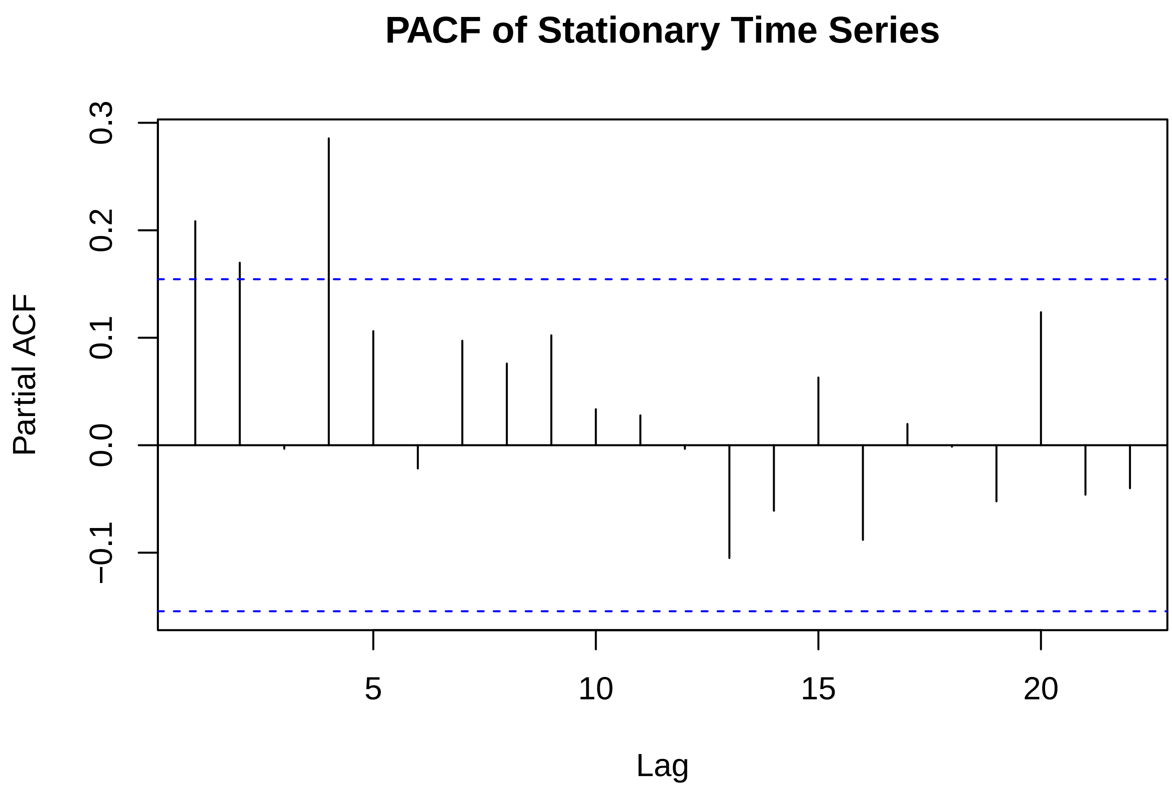
```r
# Plot the stationary time series
plot.ts(TS_1, main = "Stationary Time Series", ylab = "Temperature")
```




```r
# Plot the ACF of the stationary time series
acf(TS_1, main = "ACF of Stationary Time Series")
```




```r
# Plot the PACF of the stationary time series
pacf(TS_1, main = "PACF of Stationary Time Series")
```



Interpretation: The ACF plot indicates a gradual decay in correlation as the lag increases, which is typical for an autoregressive Model. The fact that the autocorrelations are significant for several lags before decreasing also suggests that the time series has an AR component, where past values have a lingering effect on future values.

The PACF plot shows a sharp cut-off after the first lag, with subsequent lags falling within the confidence interval, indicating that they are not statistically significant. This sharp decline in the PACF after the first lag is a characteristic of an AR(1) process(where only the first lag has a significant partial correlation).

Therefore, the ACF's gradual decrease and PACF's sharp cut-off after the firsst lag suggest that we are possibly dealing with a Time Series AR(1) model