STAT 443: Assignment 1

Flora Zhang 52135365

31 January, 2022

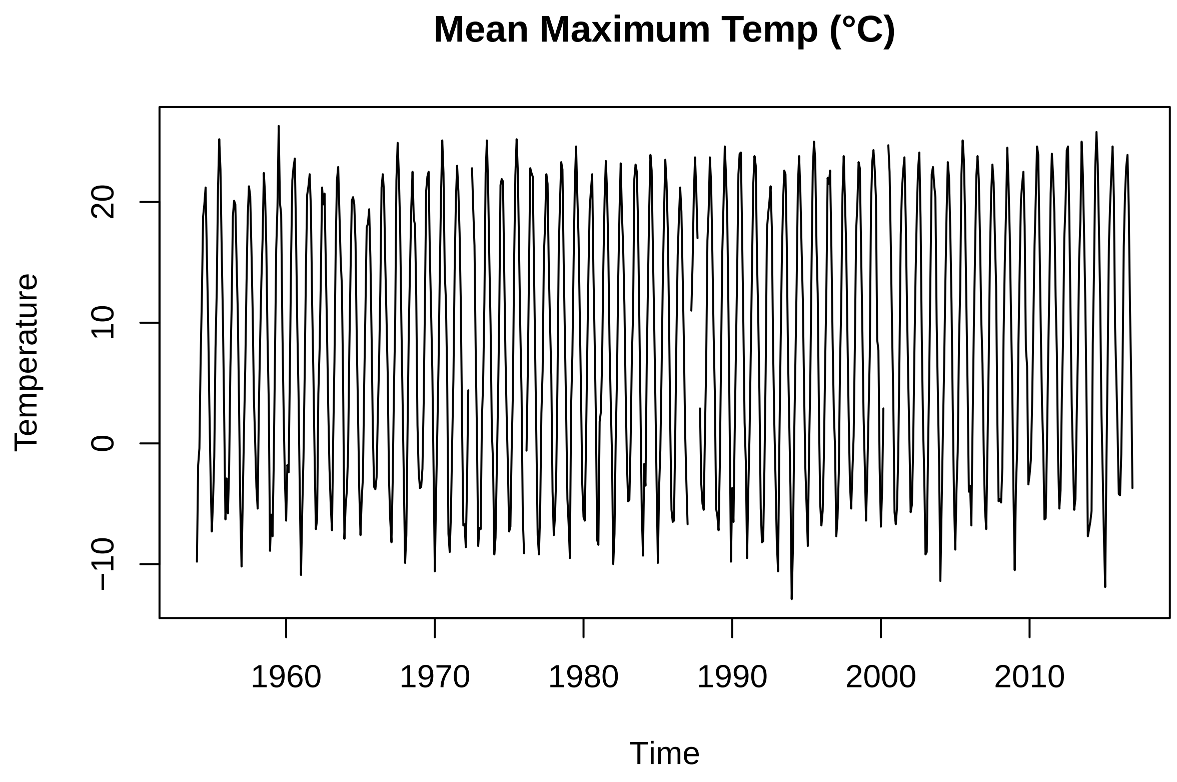
Question 1

a)

```r
#read the data and create time series
assign1_dat <- read_csv("AssignmentQ1.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   `Station Name` = col_character(),
##   `Date/Time` = col_character(),
##   Month = col_character(),
##   `Mean Max Temp Flag` = col_character(),
##   `Mean Min Temp Flag` = col_character(),
##   `Mean Temp Flag` = col_character(),
##   `Extr Max Temp Flag` = col_character(),
##   `Extr Min Temp Flag` = col_character(),
##   `Total Rain Flag` = col_character(),
##   `Total Snow Flag` = col_character(),
##   `Total Precip Flag` = col_character(),
##   `Snow Grnd Last Day Flag` = col_character(),
##   `Dir of Max Gust (10's deg)` = col_logical(),
##   `Dir of Max Gust Flag` = col_logical(),
##   `Spd of Max Gust (km/h)` = col_logical(),
##   `Spd of Max Gust Flag` = col_logical()
## )
## i Use `spec()` for the full column specifications.
```

```r
ts_data = ts(data=assign1_dat$`Mean Max Temp (°C)`, start=c(1952,1), frequency=12)
ts_data = window(ts_data, start=c(1954,1), end=c(2016,12))
plot(ts_data, xlab="Time", ylab="Temperature", main="Mean Maximum Temp (°C)")
```



From the plot, we can see that the series does not show a clear trend as we do not see a long term change in the mean of the series. The series is not stationary because there appears to be seasonality. There appears to be seasonal variation, with a period of 12. An additive model would be more suitable as we do not observe a seasonal variation that increases with the mean, which would correspond to a multiplicative model.

b)

```
#determine month and year of missing values.
date = which(is.na(assign1_dat$`Mean Max Temp (°C)`))
for (i in date) {
  if(i>24){print(assign1_dat$`Date/Time`[i])}
}
```

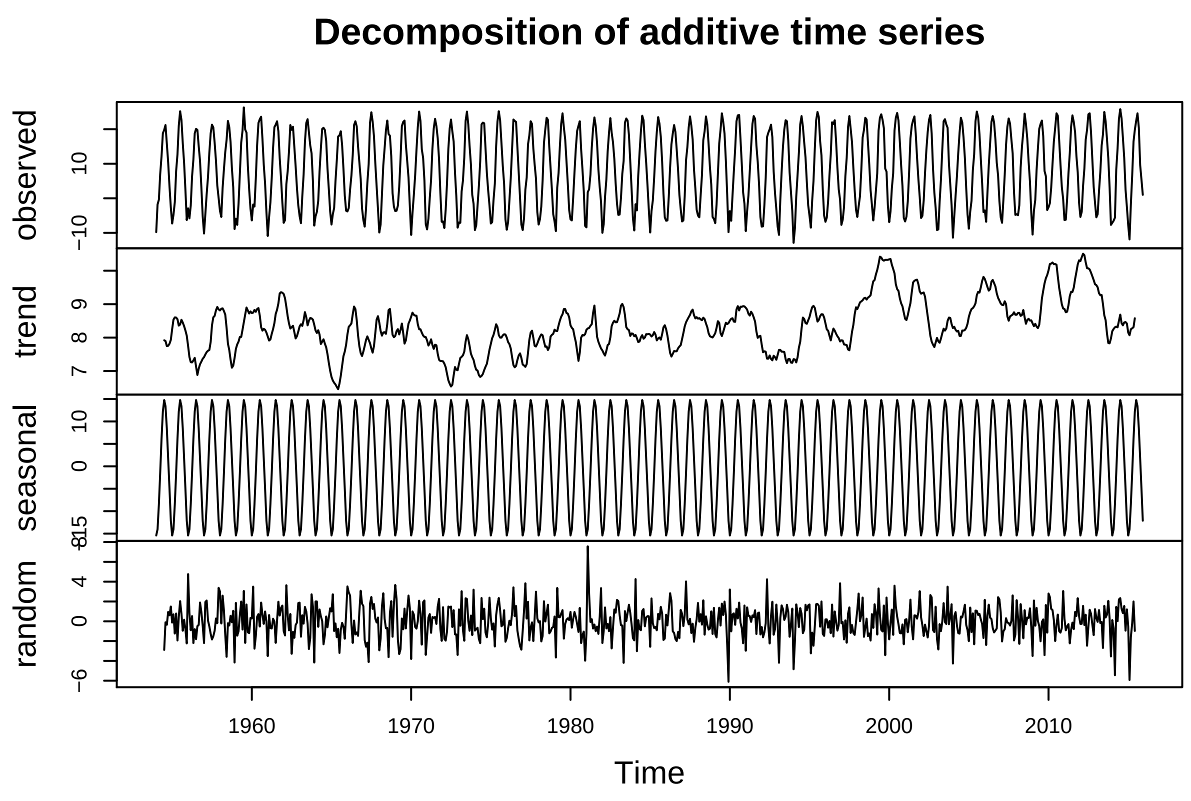
```
## [1] "1972-05"
## [1] "1972-06"
## [1] "1976-02"
## [1] "1987-02"
## [1] "1987-03"
## [1] "1987-10"
## [1] "2000-04"
## [1] "2000-05"
## [1] "2000-06"
```

```r
month = which(is.na(ts_data))
#imputing the NAs
for (i in month) {
 ts_data[i] = ts_data[i - 12]
}
```

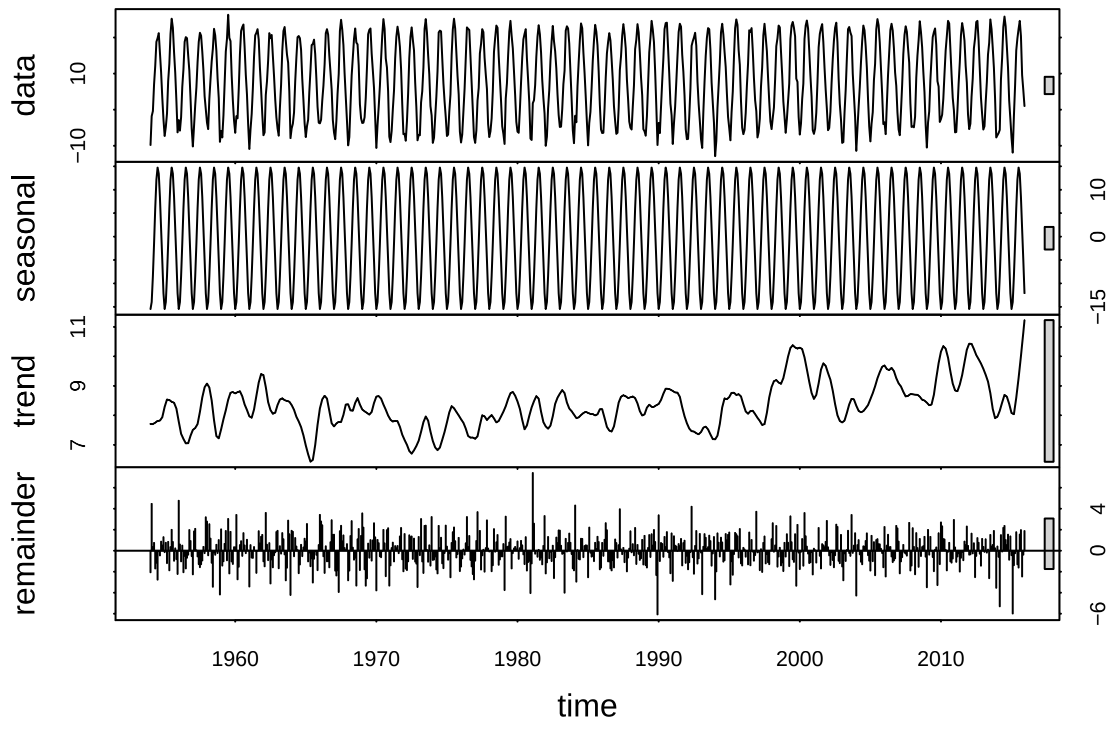
Yes, there are missing values. The NAs are: May 1972, June 1972, Feb 1976, Feb 1987, March 1987, Oct 1987, April 2000, May 2000, June 2000 In this case, the LVCF would not be appropriate since there is seasonality in the data, and imputing the last observed value would not account for that. Instead, we could set the missing observation i equal to that at index i - 12. This method applies since the data does not have a trend.

(c)

```r
#training dataset: include all observations from 1954 to 2015
training = window(ts_data, start=c(1954,1), end=c(2015, 12), frequency=12)

#test dataset: observations of 2016
test_data = window(ts_data, start=c(2016,1), end=c(2016, 12), frequency=12)

#apply decomposition to training set
#decompose method
training.decom <- decompose(training, type="additive")
plot(training.decom)
```



```
trend = training.decom$trend
seas <- training.decom$seasonal
error <- training.decom$random


#loess method
training.stl <- stl(training, s.window="periodic")
plot(training.stl)
```



```
trend.stl <- training.stl$time.series[,"trend"]
seas.stl <- training.stl$time.series[,"seasonal"]
error.stl <- training.stl$time.series[,"remainder"]
```

(d)

```
trend_subset = window(trend, start=c(1954,7), end=c(2015, 6)) #create subset to remove NAs
t = c(1:length(trend_subset))
lm.trend = lm(formula = trend_subset ~ t)
summary(lm.trend)
```

```
##
## Call:
## lm(formula = trend_subset ~ t)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53499 -0.46119 -0.02852  0.46585  1.78017
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.6907688  0.0496039     155   <2e-16 ***
## t           0.0017591  0.0001173      15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6703 on 730 degrees of freedom
## Multiple R-squared:  0.2357, Adjusted R-squared:  0.2346
## F-statistic: 225.1 on 1 and 730 DF,  p-value: < 2.2e-16
```

```
#newdata = data.frame(waiting=732)
#predict(lm.trend, newdata, interval="confidence")
```

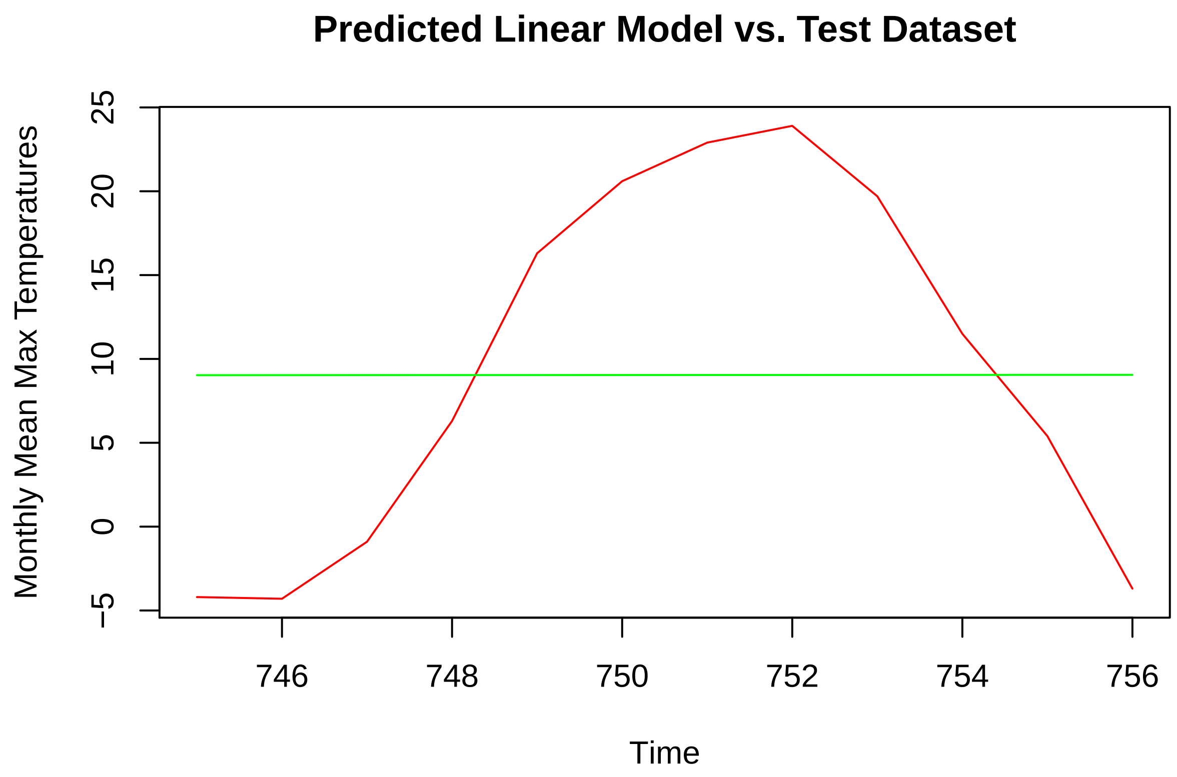
Our linear model is Yt = 7.69 + 0.0018t, where Yt is the trend component extracted from the moving average decomposition, and t is the months. The model's t-value is very small, meaning the linear line has a very minimal slope. From the trend component plot, we can see that there is variation that the linear model is not capturing. Therefore, we should not use the trend component to make predictions.

(e)

```
test_data = window(ts_data, start=c(2016,1), end=c(2016, 12), frequency=12)
test_data
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 2016 -4.2 -4.3 -0.9  6.3 16.3 20.6 22.9 23.9 19.7 11.5  5.4 -3.7
```

```
#model: Yt = 7.69 + 0.0018t
t_predict = c(745:756)  #t values
model = 7.69 + 0.0018*t_predict #the model shows a slight linear trend.

plot(t_predict, test_data, type="l", col="red", xlab = "Time", ylab="Monthly Mean Max Temperatures", ma
lines(t_predict, model, col="green")
```

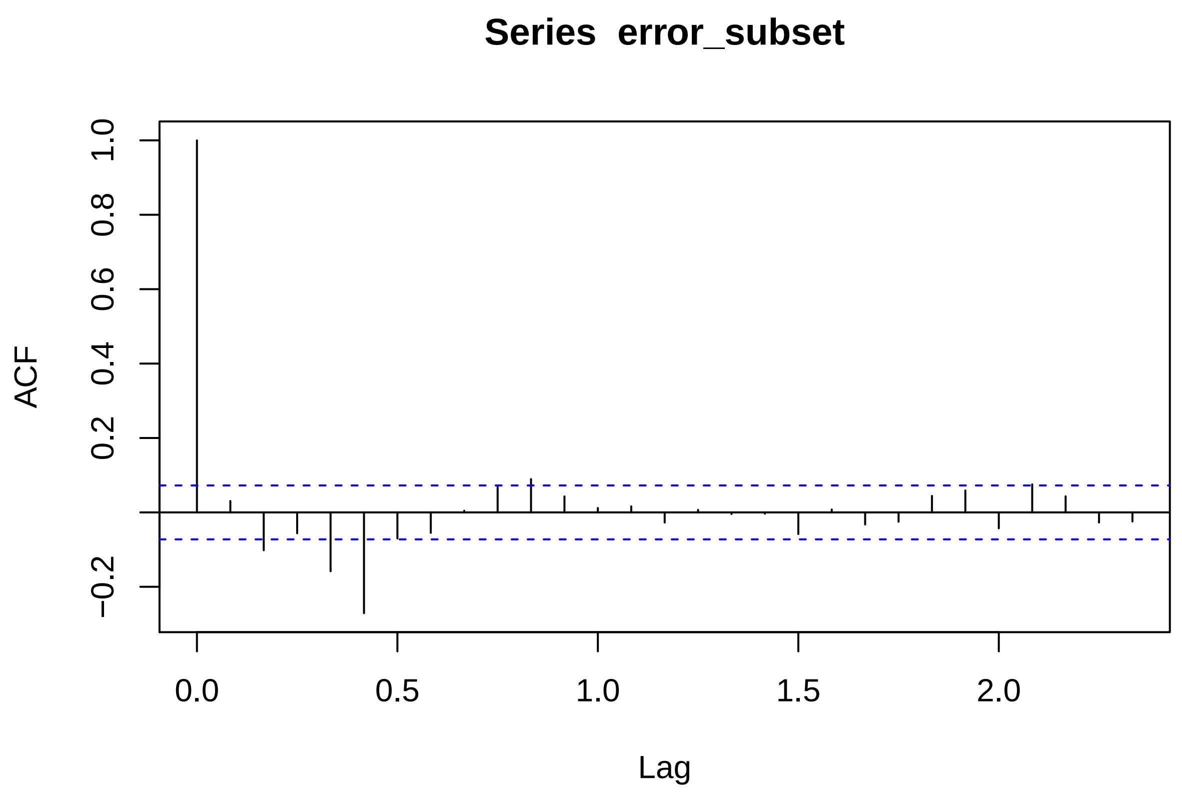


```r
#calculate MSPE = average of the squared distances between the predictions and observations in the test
MSPE = mean((model - test_data)^2)
```

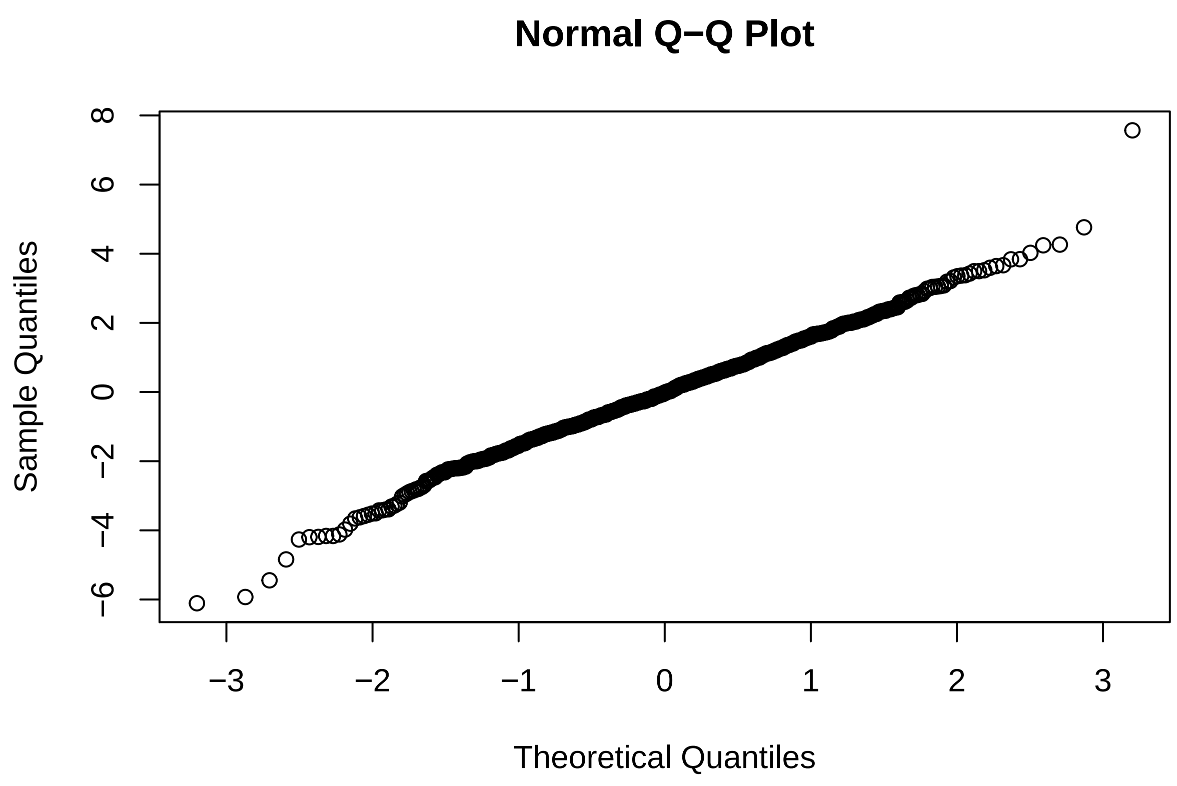
We see that a linear model does not fit the model well, as it disregards the seasonality.

(f)

```r
error_subset = window(error, start=c(1954,7), end=c(2015, 6)) #create subset to remove NAs

#correlogram
acf(error_subset)
```



```
#Q-Q plot
qqnorm(error_subset)
```



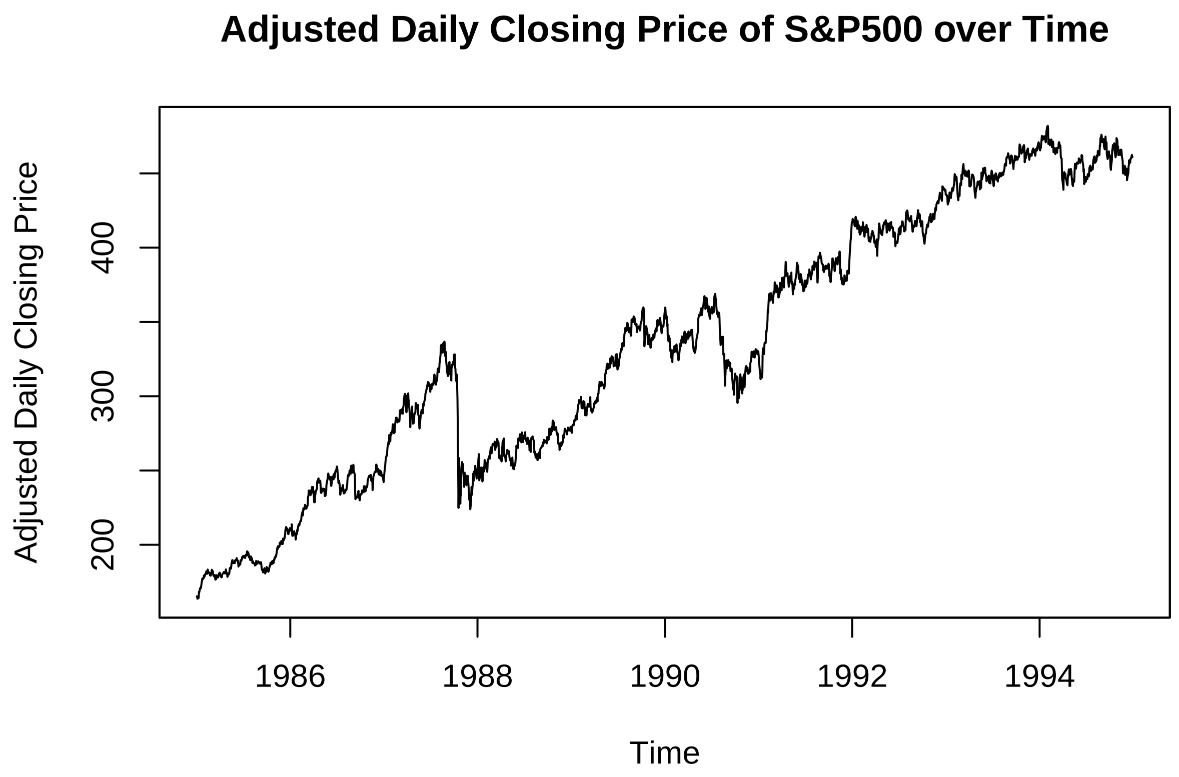
We know that white noise has variables which are independent and identically distributed, with a mean equal to zero. Additionally, all variables have the same variance of sigma^2, and each value has a zero correlation with all other values in the series. From the correlogram, we see that there is no obvious autocorrelation pattern. The values are mostly contained in the confidence interval bounds. The Q-Q plot allows us to check if the error terms are normally distributed. In this Q-Q plot, we observe a straight line, and thus can reasonably determine that the errors are normally distributed. Therefore, the Gaussian white noise assumption is appropriate.

Question 2 (a)

```
a1_q2 <- read_csv("GSPC.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Date = col_date(format = ""),
##   GSPC.Open = col_double(),
##   GSPC.High = col_double(),
##   GSPC.Low = col_double(),
##   GSPC.Close = col_double(),
##   GSPC.Volume = col_double(),
##   GSPC.Adjusted = col_double()
## )
```

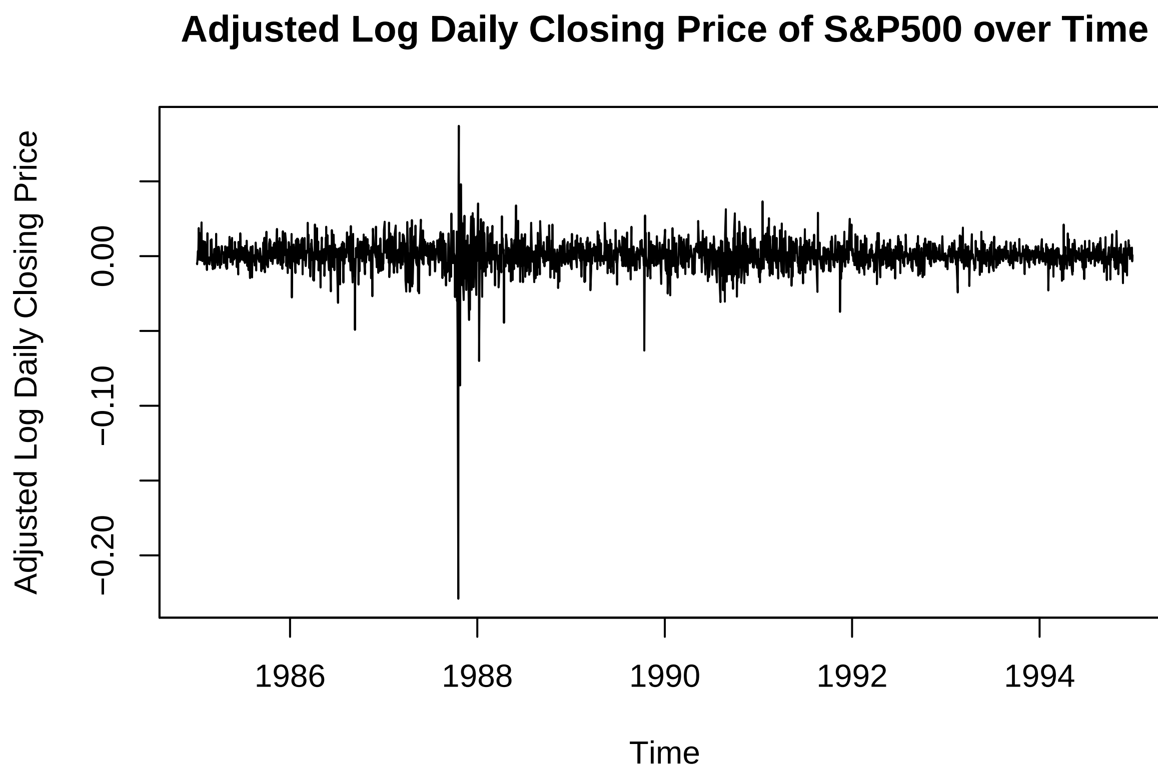
```
q2 = zoo(x=a1_q2$GSPC.Adjusted, order.by = a1_q2$Date)
plot(q2, xlab="Time", ylab = "Adjusted Daily Closing Price", main="Adjusted Daily Closing Price of S&P5
```



From this graph, we can see that there is an upwards trend in the data. There does not appear to be a seasonal effect. Since there is a trend, this data is non-stationary.

(b)

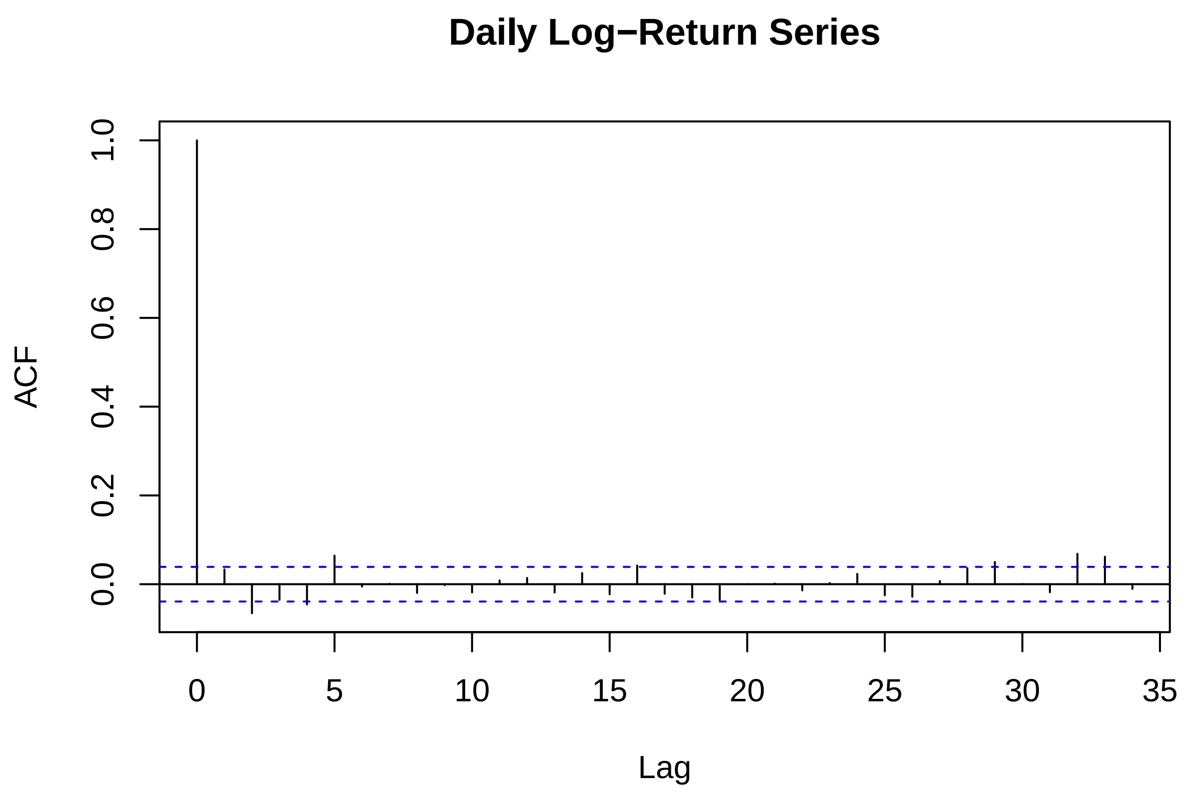
```
log_q2 = diff(log(q2), lag=1)
plot(log_q2, xlab="Time", ylab = "Adjusted Log Daily Closing Price", main="Adjusted Log Daily Closing Pr
```



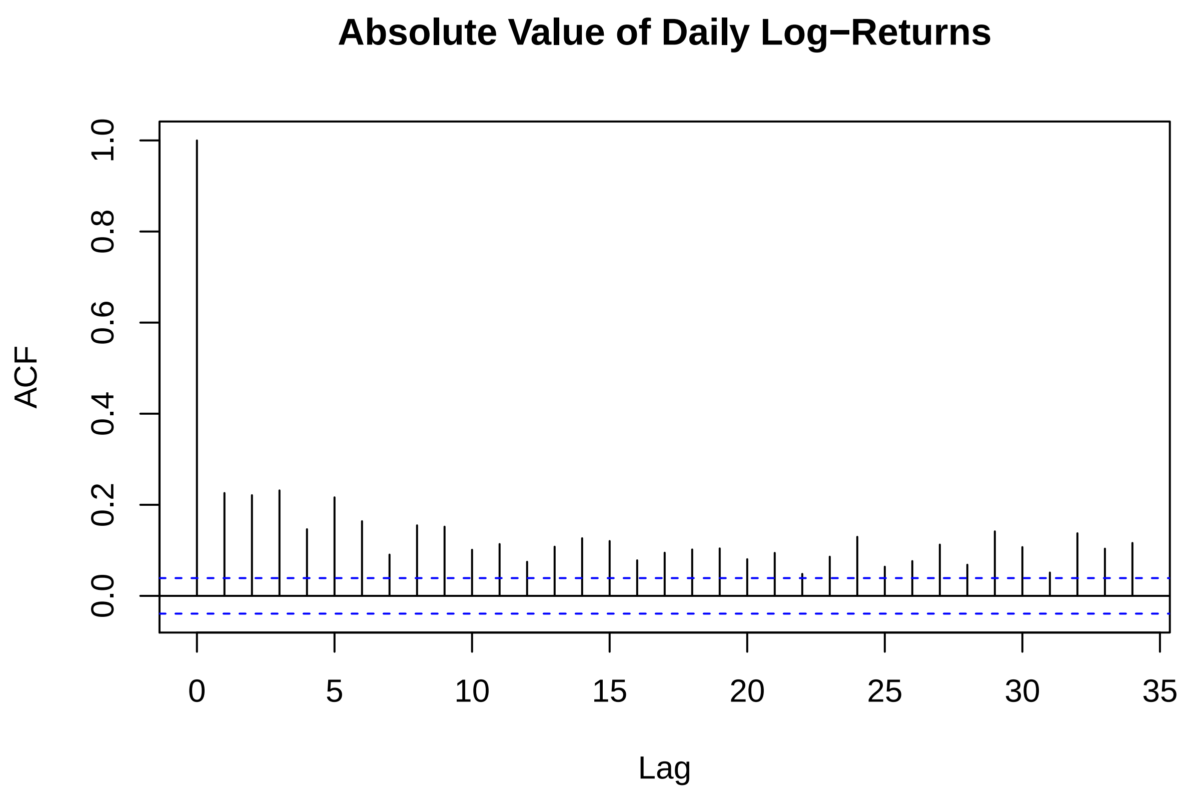
By taking the logarithmic difference and transforming the data into daily log-returns, we can see that this removed the upwards trend previously observed. The data appears like white noise, and looks to have a constant mean, with a finite variance. Therefore, it appears to be stationary. This makes the data more convenient to work with as stationary time series is easier to model. It also ensures that the past is informative about the future, making predictions possible.

(c)

```r
#daily log-return series
log_ts = as.vector(log_q2)
acf(log_ts, main="Daily Log-Return Series")
```



```r
#absolute value of daily log-returns
acf(abs(log_ts), main="Absolute Value of Daily Log-Returns")
```



We can see that the daily-log return series correlogram has a sharp decay, with the ACF fluctuating between small positive and negative numbers. Most of the coefficients are within the confidence band and are not statistically significant. The absolute value daily-log return series also has a sharp decay, with all positive values. It appears that some coefficients are outside of the confidence band, and are statistically significant, demonstrating that the absolute value graph has a slower decay compared to the daily-log return series.