

## Scenario 5

Nipun

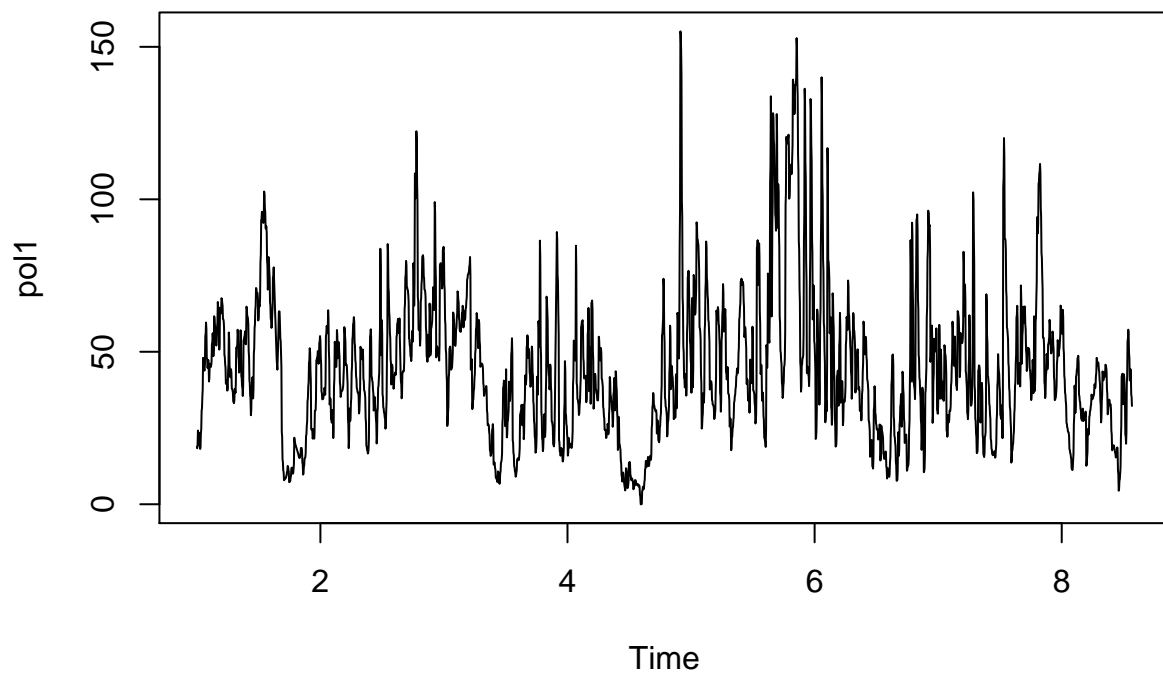
23/04/2021

```
library("forecast")  
library("fpp2")  
library("tseries")
```

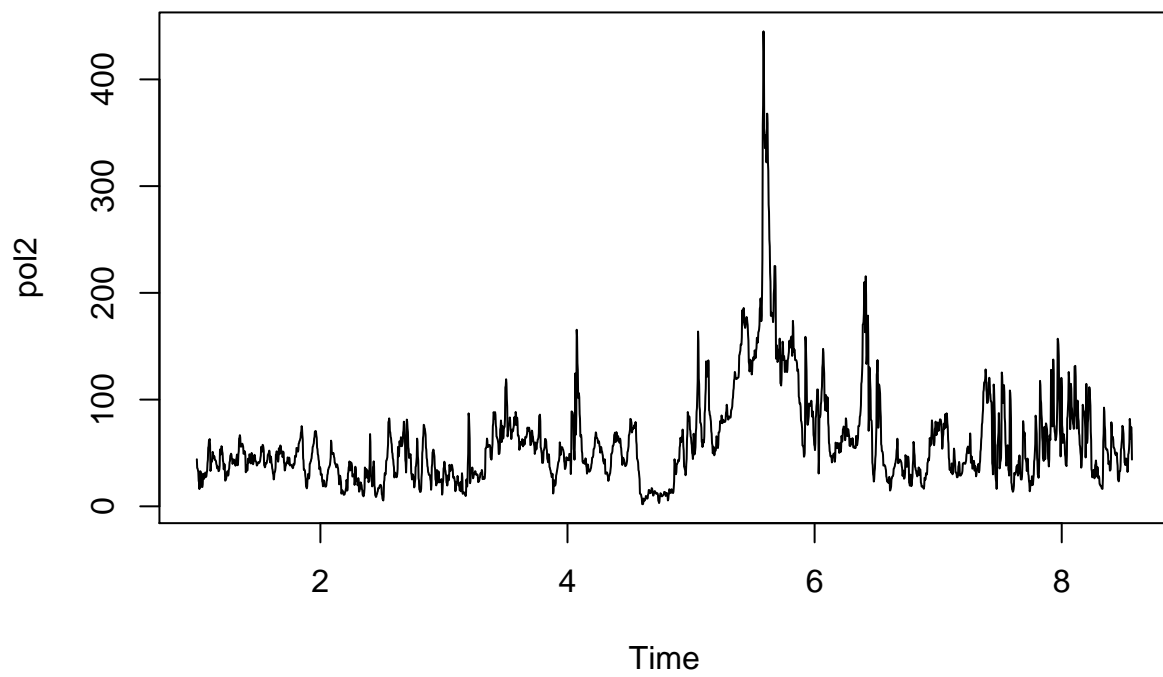
```
pol1_table <- read.table("pollutionCity1.txt", header = TRUE, stringsAsFactors = FALSE, sep=",")  
pol2_table <- read.table("pollutionCity2.txt", header = TRUE, stringsAsFactors = FALSE, sep=",")  
pol3_table <- read.table("pollutionCity3.txt", header = TRUE, stringsAsFactors = FALSE, sep=",")  
pol1 <- ts(pol1_table$x, frequency=336)  
pol2 <- ts(pol2_table$x, frequency=336)  
pol3 <- ts(pol3_table$x, frequency=336)
```

We are given half-hourly measurements of the concentration of an air pollutant in three different cities over 53 days which are pol1, pol2 and pol3 time series. First, we will plot the data with weekly time period to get a bigger picture of how each time series looks and what type of time series models can we fit to them

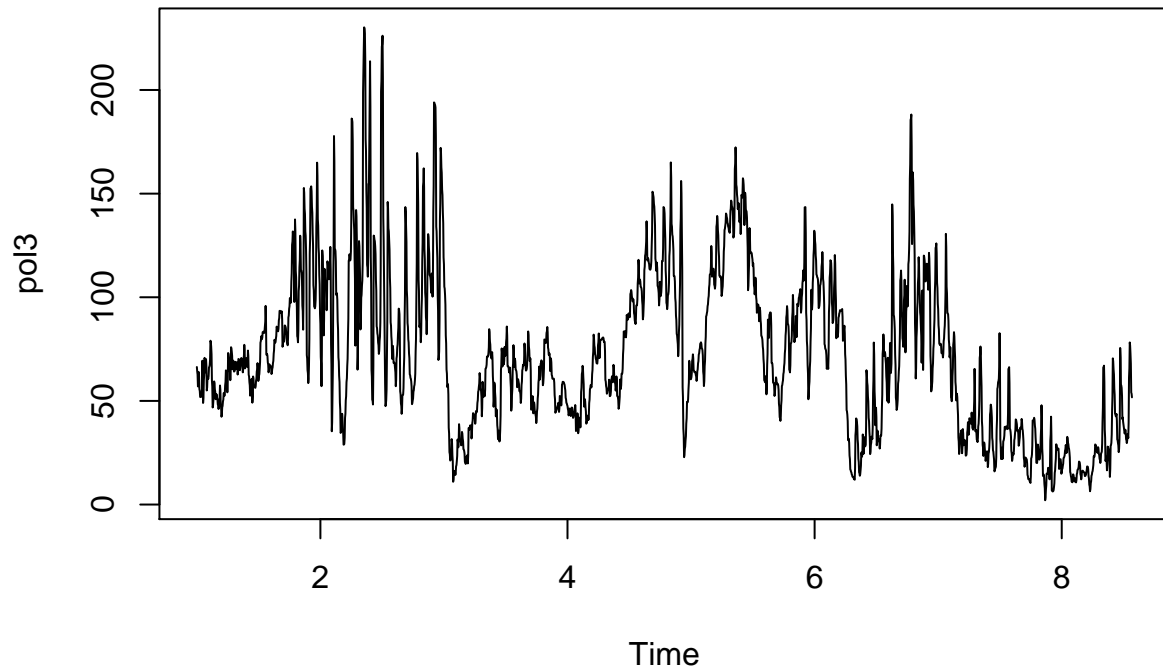
```
plot(pol1)
```



```
plot(pol12)
```



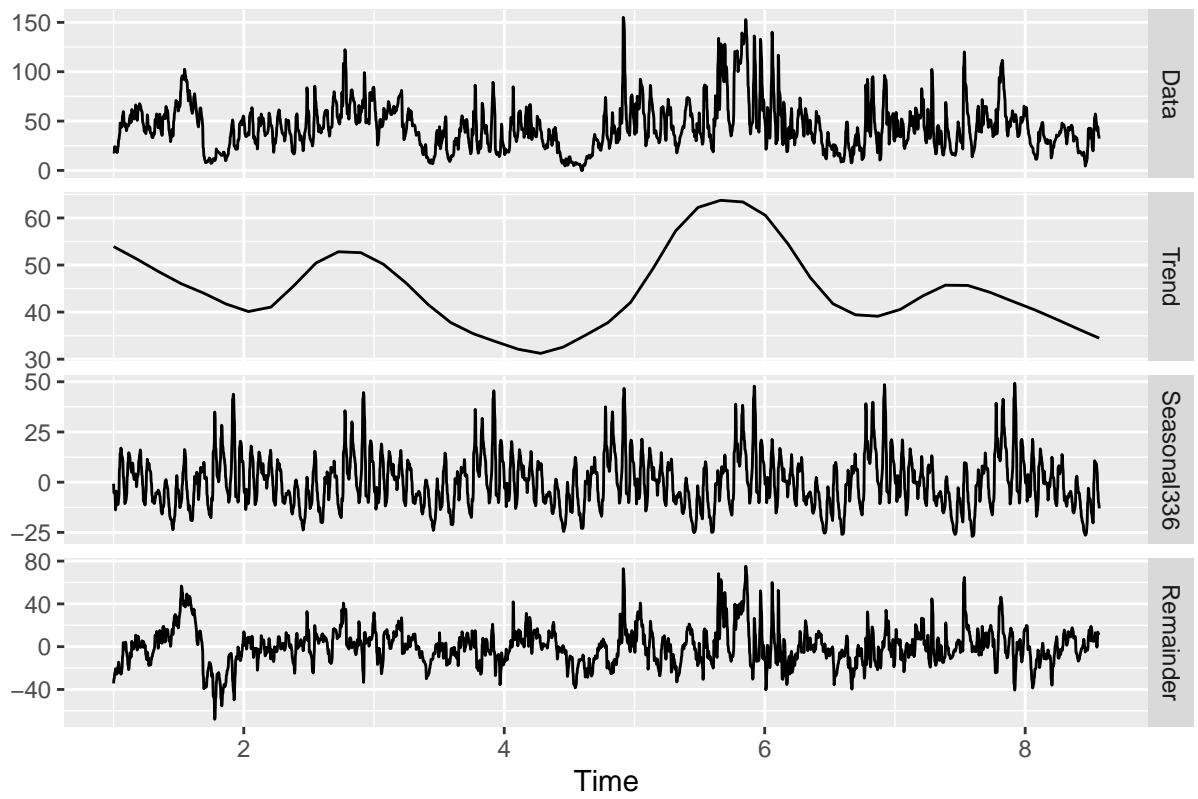
```
plot(pol13)
```



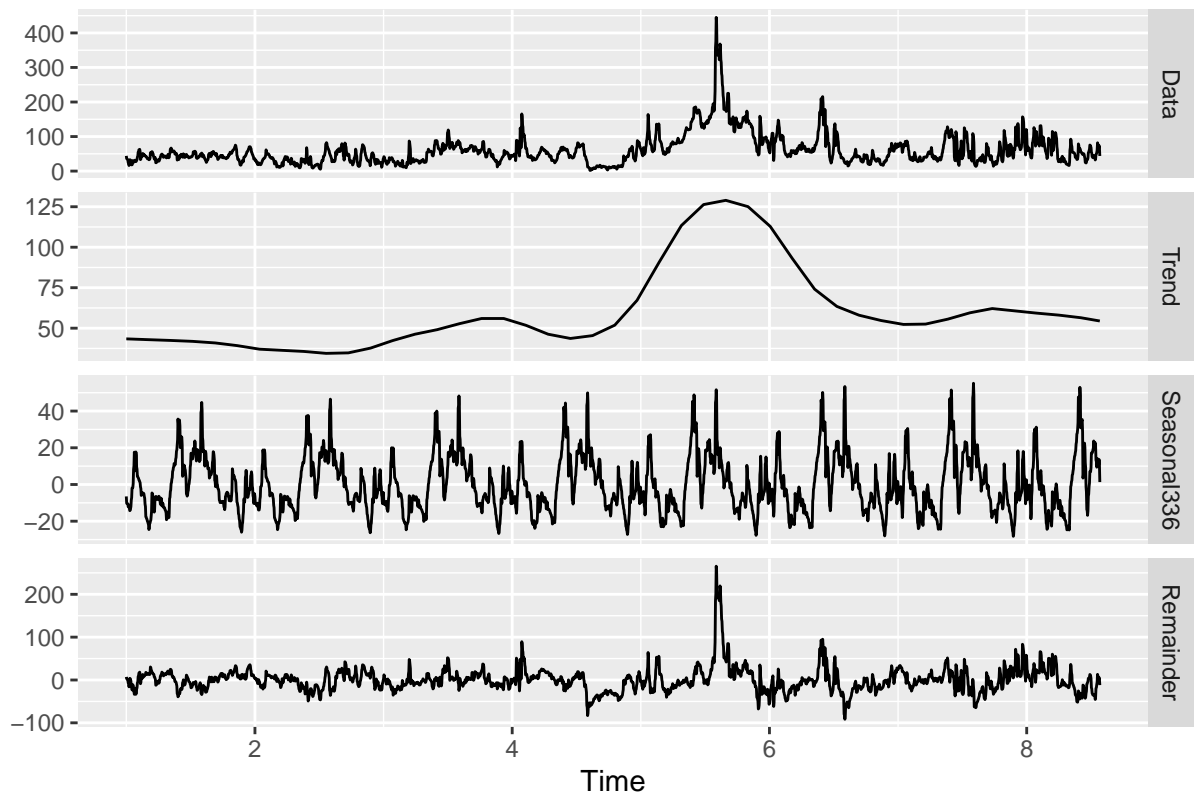
After looking at the plots of pollution time series, we know that hourly data usually has three types of seasonality which are: daily pattern, weekly pattern and annual pattern. But since, we are given the data for only 53 days, we won't be able to account for annual pattern. Hence, we might have a daily and weekly pattern for our models to account for.

We will examine our assumption of multiple seasonal patterns in the data by fitting a STL with multiple seasonal time periods method to these time series. This method will decompose the time series into season, trend and remainder components.

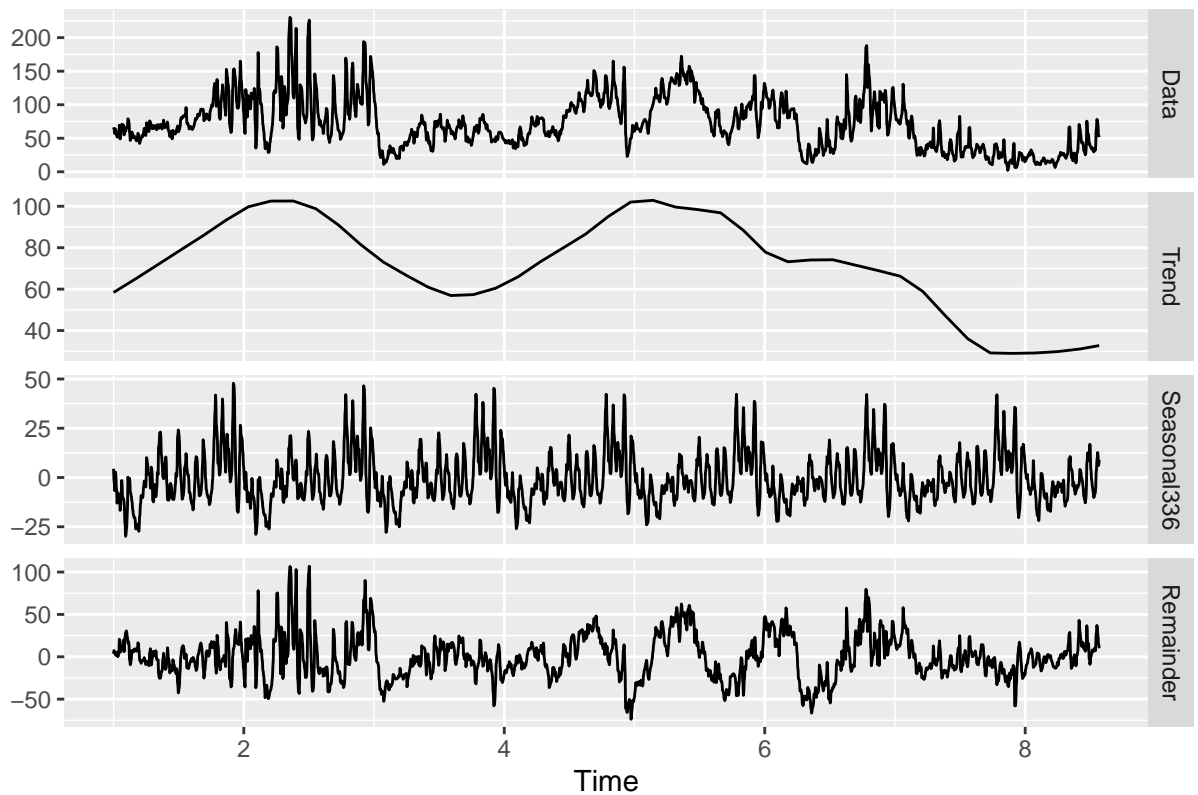
```
autoplot(mstl(pol1))
```



```
autoplot(mstl(pol2))
```



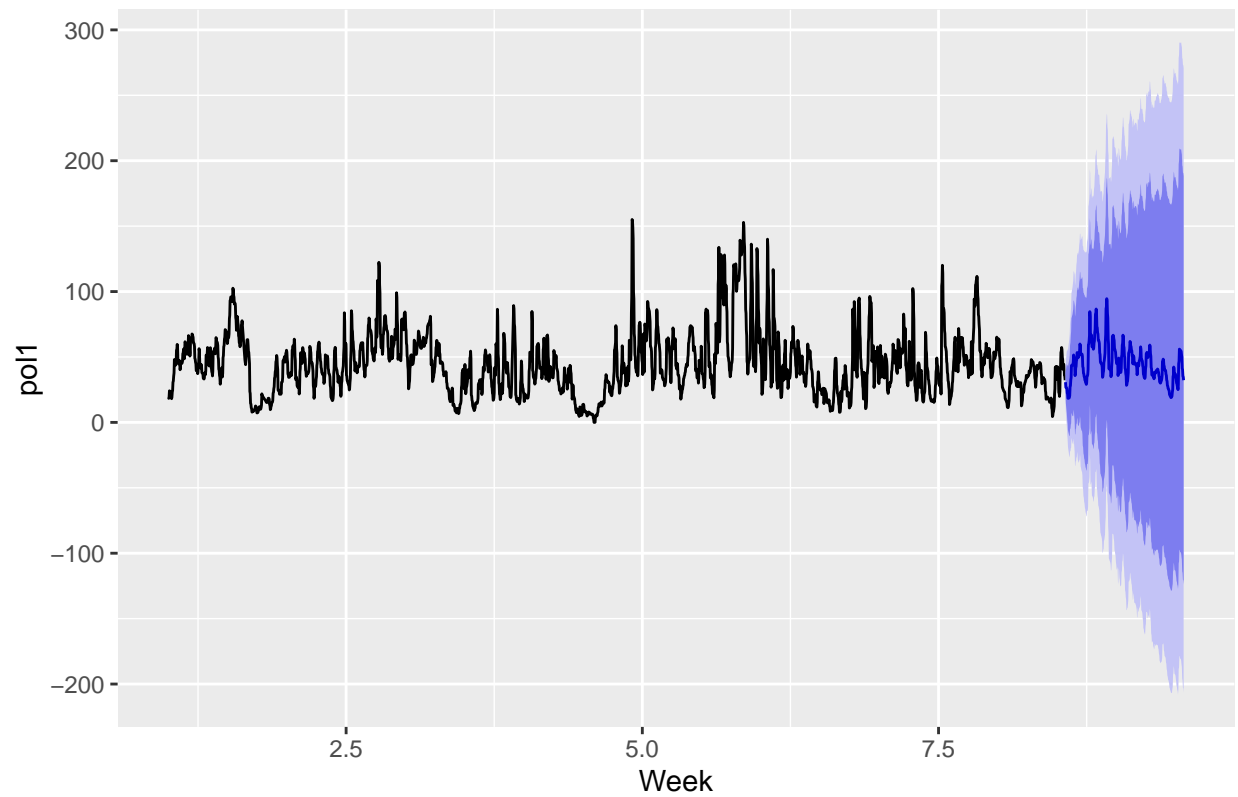
```
autoplot(mstl(pol3))
```



From the time series decomposition, we observe that there is a weekly pattern in the time series. Now we will use this decomposition in forecasting the time series using the ETS model

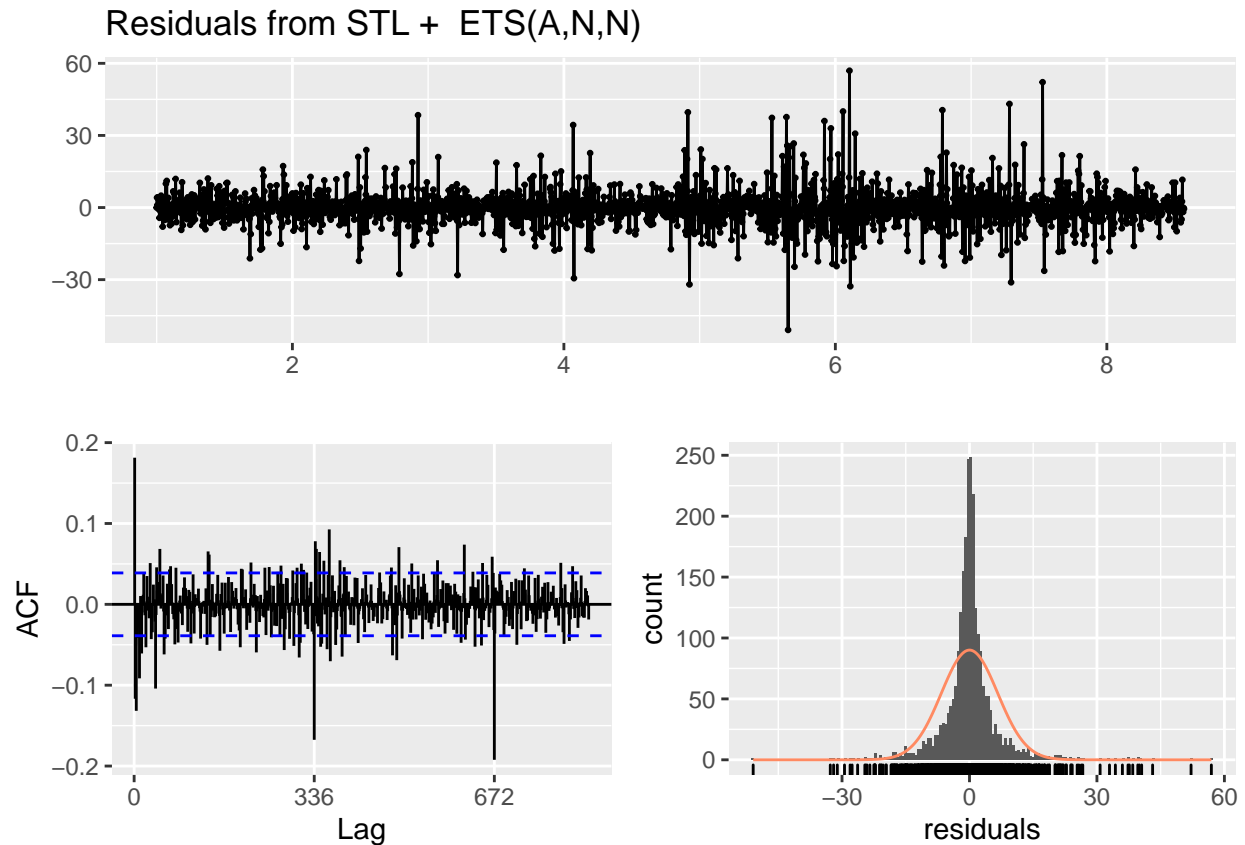
```
pol1.for.ets <- stlf(pol1, h=336, method="ets")
autoplot(pol1.for.ets, xlab="Week")
```

Forecasts from STL + ETS(A,N,N)



```
checkresiduals(pol1.for.ets)
```



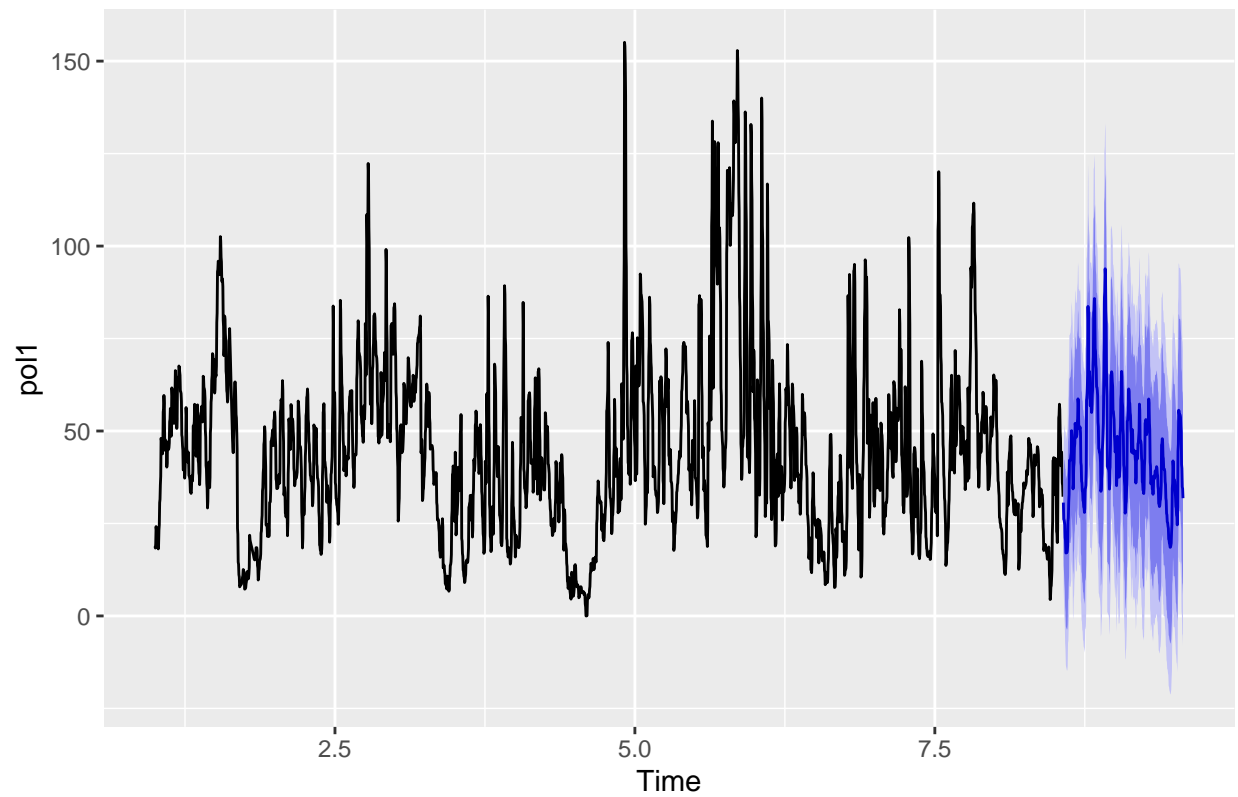


```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(A,N,N)
## Q* = 1091.8, df = 507, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 509
```

From looking at the forecasts of STL+ETS(A,N,N) model, we observe that 336 steps ahead forecasts look plausible given the past observations of the time series. The 95% prediction intervals also look plausible but they seem to be too wide given the variations in the past observations of the time series. Additionally, by looking at the histogram of the residuals, the residuals look gaussian. The ACF plot of residuals has large values specially at lag=336 and lag=672 which indicates that there is weekly seasonality present and additionally, the residuals dont seem to be a white sequence

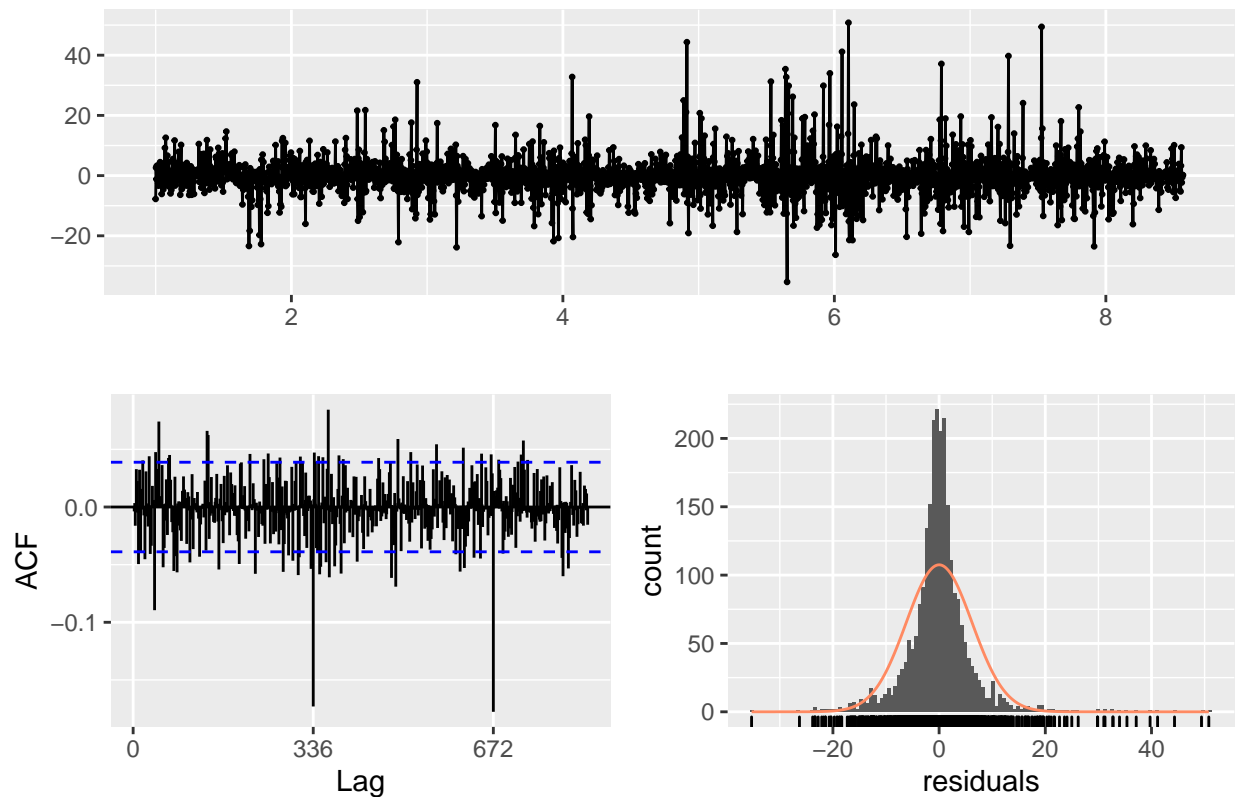
```
pol1.for.ar <- stlf(pol1, h=336, method="arima")
autoplot(pol1.for.ar)
```

Forecasts from STL + ARIMA(4,0,2) with non-zero mean



```
checkresiduals(pol1.for.ar)
```

## Residuals from STL + ARIMA(4,0,2) with non-zero mean



```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ARIMA(4,0,2) with non-zero mean
## Q* = 733.38, df = 502, p-value = 6.364e-11
##
## Model df: 7.   Total lags used: 509
```

From looking at the forecasts of STL+ARIMA(4,0,2) model, we observe that 336 steps ahead forecasts look plausible given the past observations of the time series. The 95% prediction intervals also look plausible as they are not too wide but wide enough to capture the variations in the time series. Additionally, by looking at the histogram of the residuals, the residuals look gaussian. The ACF plot of residuals has large values specially at lag=336 and lag=672 which indicates that there is weekly seasonality present and additionally, the residuals don't seem to be a white sequence.

```
fETS <- function(x, h) {
  stlf(x, h=h, method="ets")
}

fAR <- function(x, h) {
  stlf(x, h=h, method="arima")
}

cvETS <- tsCV(pol1, fETS, h = 336)
cvAR <- tsCV(pol1, fAR, h = 336)
```

```

cvETS = na.remove(cvETS)
cvAR = na.remove(cvAR)

B = round(0.25*length(pol1))

ETS.test= cvETS[(length(cvETS)-B):length(cvETS)]
AR.test= cvAR[ (length(cvAR)-B):length(cvAR)]

mean(ETS.test^2)

```

```
## [1] 1058.083
```

```
mean(AR.test^2)
```

```
## [1] 846.593
```

From the above results of cross-validation, we observe that STL+ARIMA model has a lower cross-validation mean squared error than STL+ETS(A,N,N) model. Additionally, by looking at the forecasts of STL+ARIMA model, we can conclude that they fit the population1 time series data better than STL+ETS(A,N,N) model. Hence, we choose STL+ARIMA model as our final model for population1

Now, we will perform the same procedure on pop2 and pop3 time series, and choose the model based on the lowest mean squared error

```

cvETS2 <- tsCV(pol2, fETS, h = 336)
cvAR2 <- tsCV(pol2, fAR, h = 336)
cvETS3 <- tsCV(pol3, fETS, h = 336)
cvAR3 <- tsCV(pol3, fAR, h = 336)

cvETS2 = na.remove(cvETS2)
cvAR2 = na.remove(cvAR2)
cvETS3 = na.remove(cvETS3)
cvAR3 = na.remove(cvAR3)

B2 = round(0.25*length(pol2))
B3 = round(0.25*length(pol3))

ETS.test2= cvETS2[(length(cvETS2)-B2):length(cvETS2)]
AR.test2= cvAR2[ (length(cvAR2)-B2):length(cvAR2)]
ETS.test3= cvETS3[(length(cvETS3)-B3):length(cvETS3)]
AR.test3= cvAR3[ (length(cvAR3)-B3):length(cvAR3)]

mean(ETS.test2^2)

```

```
## [1] 2666.268
```

```
mean(AR.test2^2)
```

```
## [1] 2911.312
```

```
mean(ETS.test3^2)
```

```
## [1] 2526.785
```

```
mean(AR.test3^2)
```

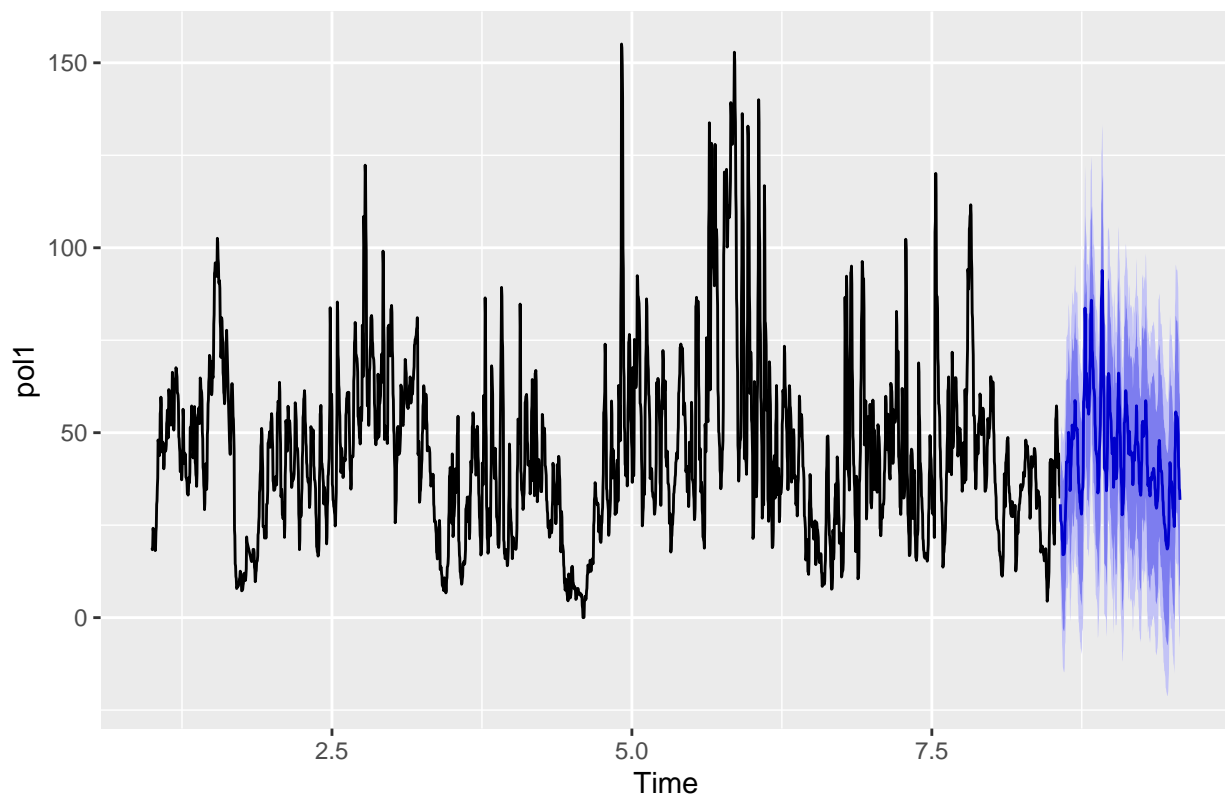
```
## [1] 2320.562
```

By looking at the results of cross-validation mean square error, we conclude that STL+ETS model fits pol2 better than STL+ARIMA model but for pol3 time series, STL+ARIMA model fits better than STL+ETS model

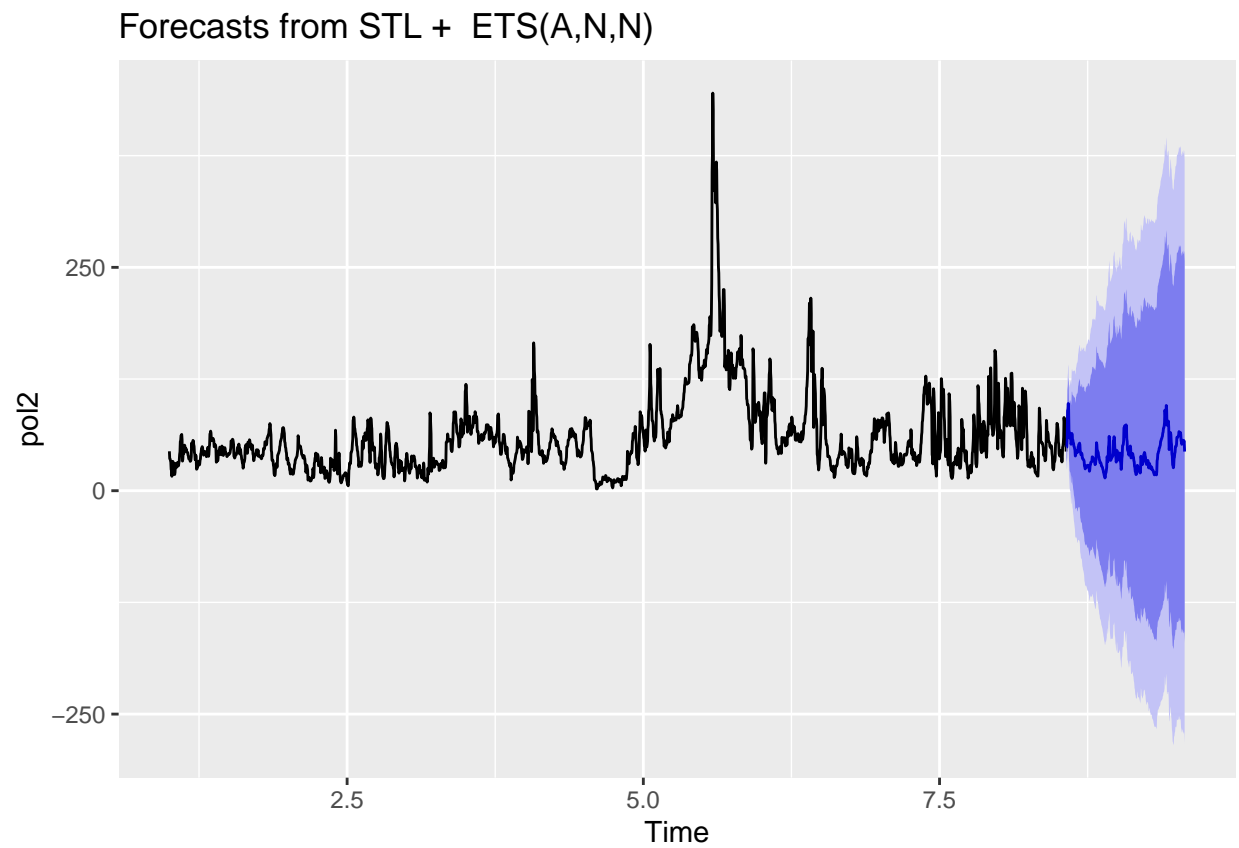
The 95% prediction intervals for pol1, pol2 and pol3 time series are

```
pol2.for.ets <- stlf(pol2, h=336, method="ets")  
pol3.for.ar <- stlf(pol3, h=336, method="arima")  
autoplot(pol1.for.ar)
```

Forecasts from STL + ARIMA(4,0,2) with non-zero mean

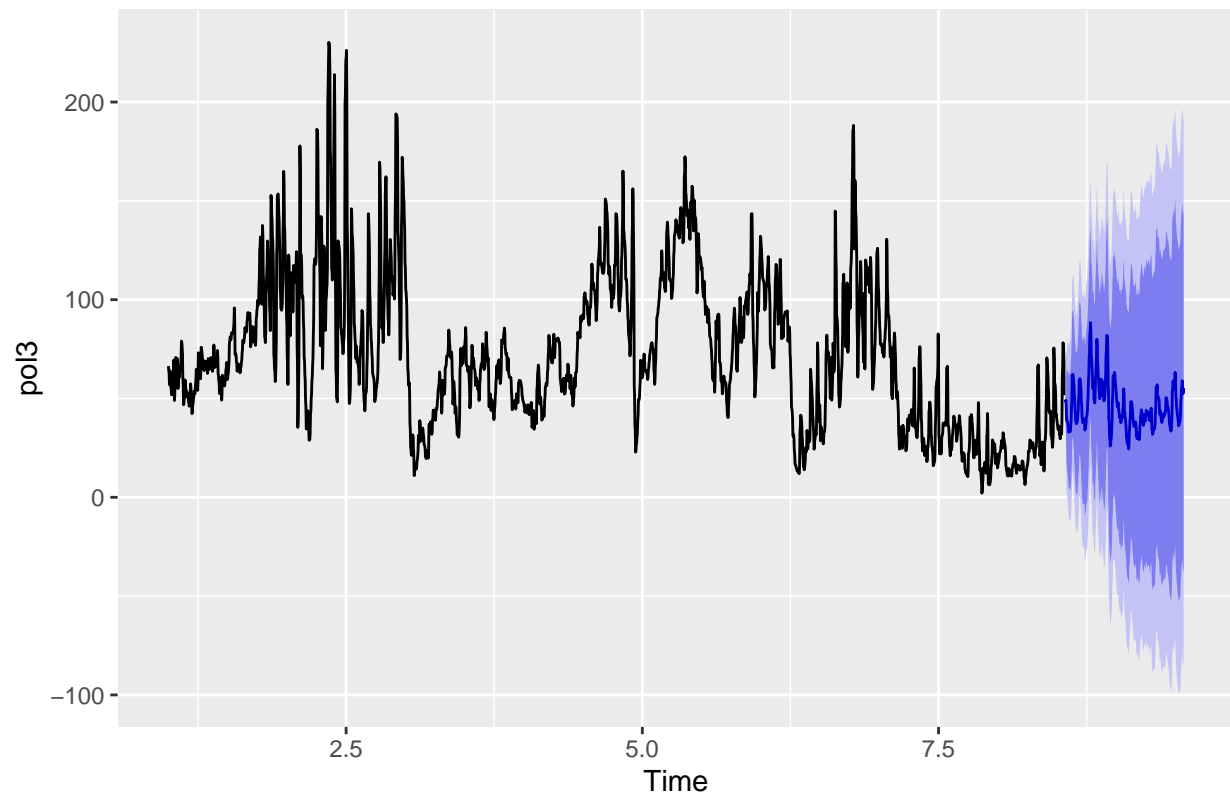


```
autoplot(pol2.for.ets)
```



```
autoplot(pol3.for.ar)
```

Forecasts from STL + ARIMA(5,1,3)



```
#Scenario 5
forecast5 = matrix(1:(3*336),336,3) #forecasts for cities 1-3 should be stored in the columns of a 336x3 matrix
forecast5[,1] = pol1.for.ar$mean
forecast5[,2] = pol2.for.ets$mean
forecast5[,3] = pol3.for.ar$mean

write.table(forecast5, file = paste("Scenario5_", "Lamba", "20751692", ".txt", sep = ""), sep = ",", col.names = FALSE)
```