# project_Times_series

Omar

2024-10-16

### Packages

```r
library(TSA)
```

```
##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```r
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method       from
##   fitted.Arima TSA
##   plot.Arima   TSA
```

```r
library(Metrics)
```

```
##
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':
##
##     accuracy
```

### Importing the data

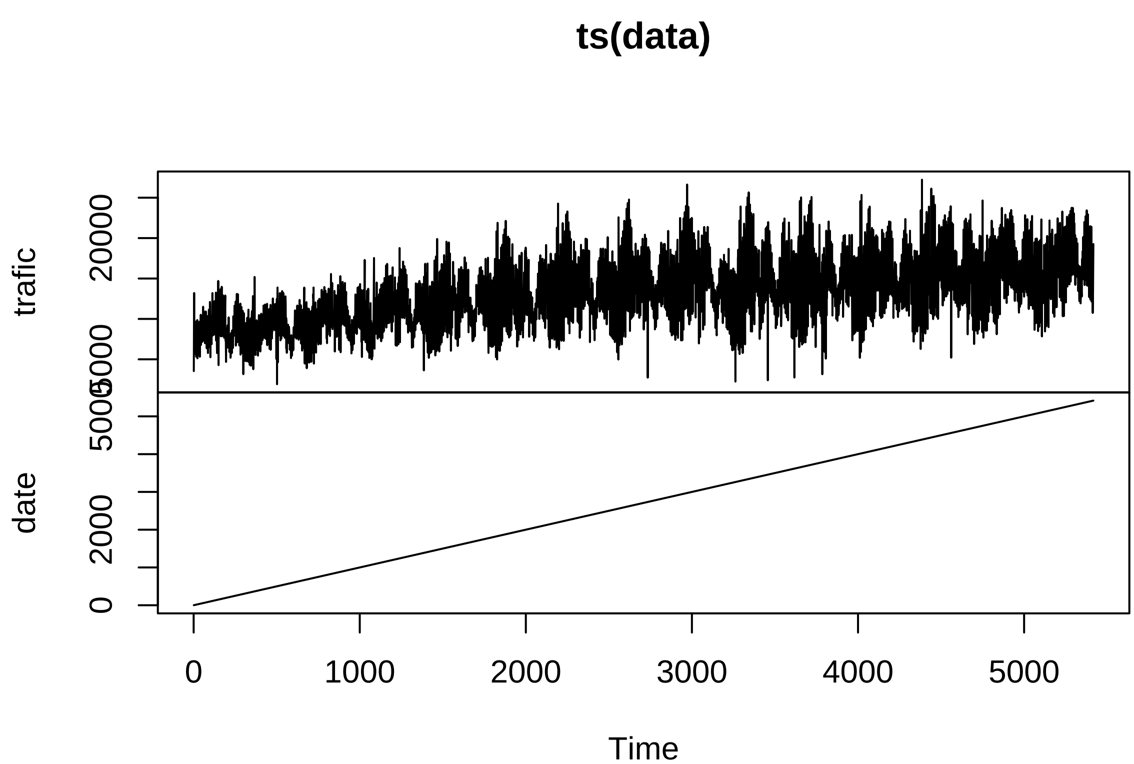
```
data = read.table("trafquoti.txt",
                  header = FALSE,
                  quote ="",
                  colClasses=c('numeric', 'character'),
                  col.names = c('trafic', 'date'))
```

### Data Preparation

```
summary(data)
```

```
##      trafic          date          
##  Min.   : 1915   Length:5417       
##  1st Qu.:10345   Class :character  
##  Median :13115   Mode  :character  
##  Mean   :13174                     
##  3rd Qu.:15729                     
##  Max.   :27231                     
```

```
plot(ts(data))
```



```r
first_date = data$date[1]
print(paste("first date", first_date))
```

```
## [1] "first date 1993-01-01"
```

```r
last_date = data$date[nrow(data)]
print(paste("Last date" , last_date))
```

```
## [1] "Last date 2007-10-31"
```

```r
#which date has the lowest number of passenger
print(paste(" date with the lowest number of passenger",
            data$date[data$trafic == 1915]))
```

```
## [1] " date with the lowest number of passenger 1994-05-17"
```

```r
print(paste(" number of passenger  the day of 09/11",
            data$trafic[data$date == "2001-09-11" ]))
```

```
## [1] " number of passenger  the day of 09/11 15622"
```

```r
data[data$date == "2001-09-14", ]
```

```
##      trafic       date
## 3179  17185 2001-09-14
```

```r
nrow(data)
```

```
## [1] 5417
```

```r
#transform the string date to actual date and see if we have some missing valu

date.1 = as.Date(data$date)
date.2 = seq(from = as.Date(first_date), to = as.Date(last_date), by = "day")
c(length(date.1), length(date.2)) # no missing value
```

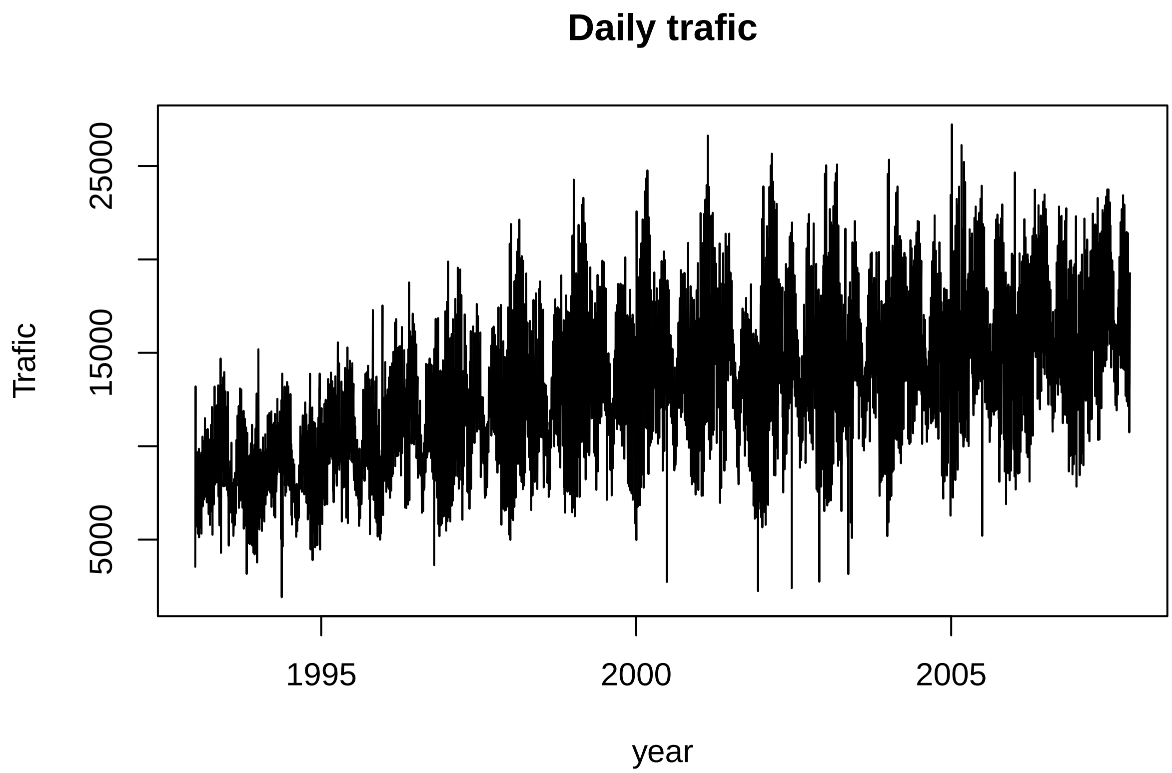
```
## [1] 5417 5417
```

```r
sum(any(is.na(data)))
```

```
## [1] 0
```

there is no missing value

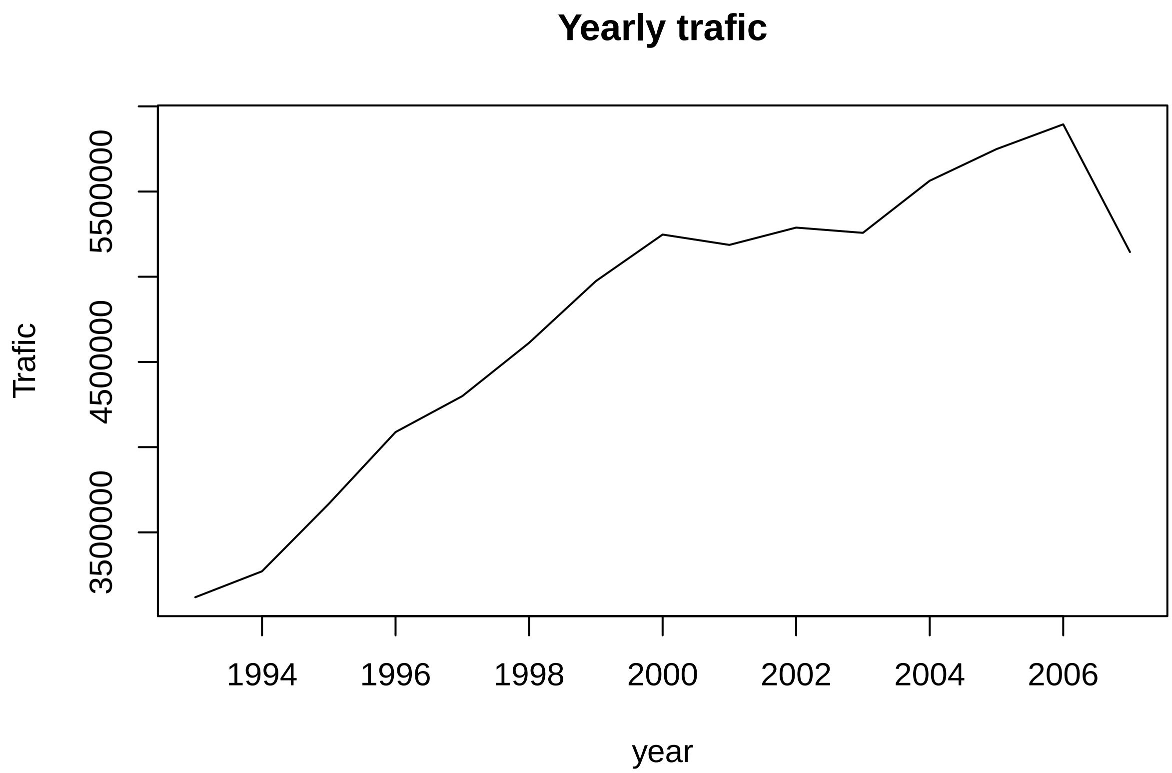
```r
#Daily time series
trafic_daily <- ts(data$trafic, start = c(1993, 1), frequency = 365)
plot(trafic_daily, main = "Daily trafic", xlab="year", ylab = "Trafic")
```




```r
#yearly time series
year  = substr(data$date, 1, 4)
trafic_year = aggregate(data$trafic, list(year = year), sum)
trafic_year = ts(trafic_year$x, start = 1993, frequency = 1)
str(trafic_year)
```

```
##  Time-Series [1:15] from 1993 to 2007: 3119216 3271363 3667372 4088480 4299624 ...
```

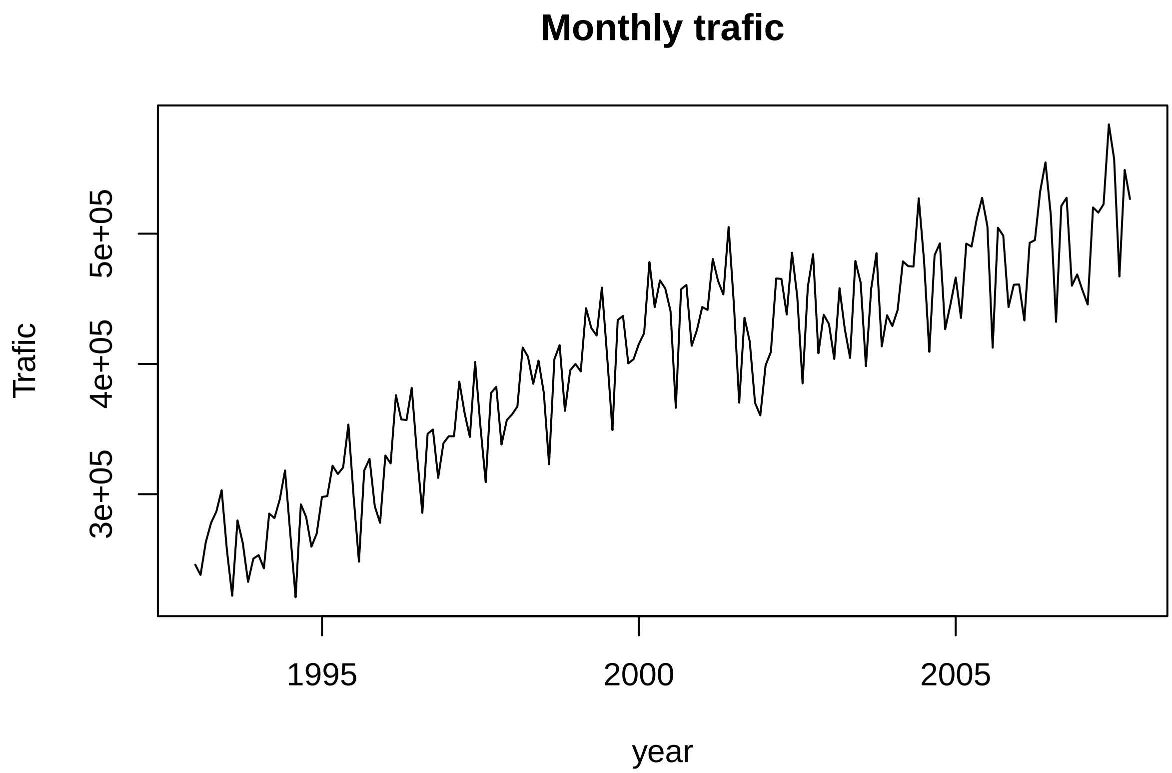
```r
plot(trafic_year, main = "Yearly trafic", xlab = "year", ylab = "Trafic")
```




```r
#monthly time series
month = substr(data$date, 6, 7)
month.year = as.numeric(paste(year,month,sep ="."))
trafic_month = aggregate(data$trafic, list (month = month.year), sum)
trafic_month = ts(trafic_month$x, start = c(1993,1), frequency = 12)
str(trafic_month)
```

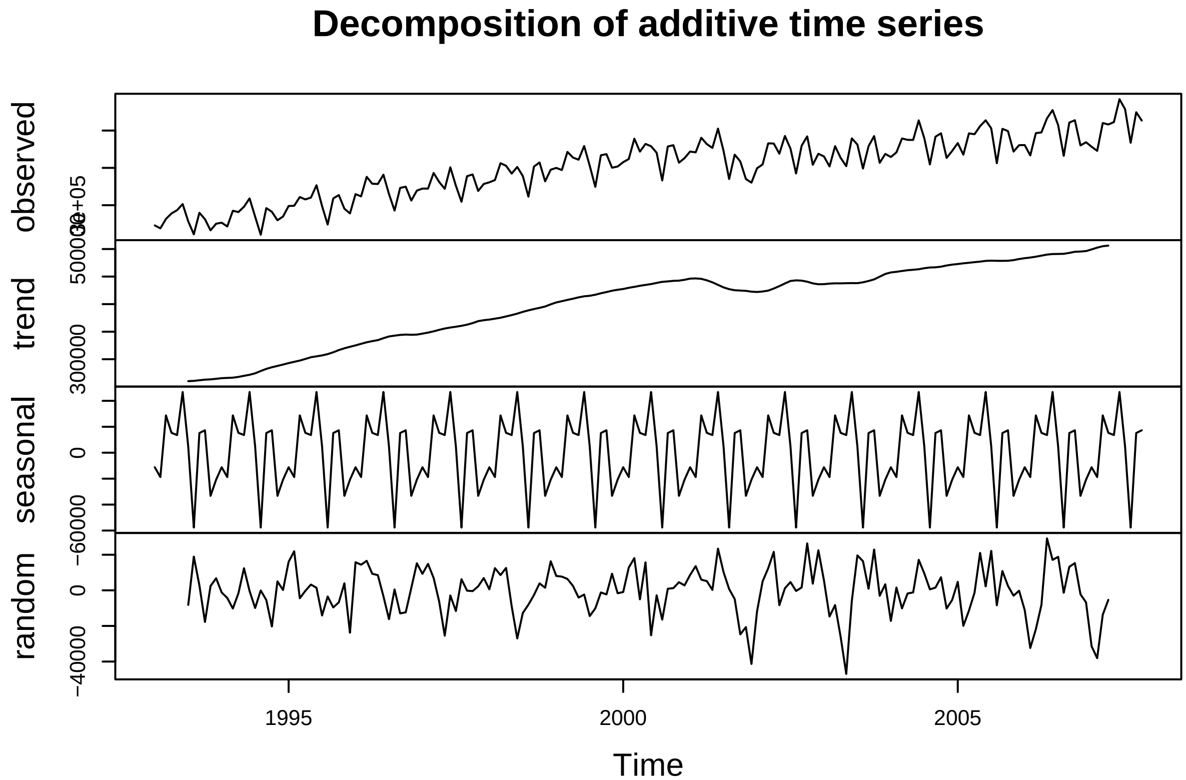
```
##  Time-Series [1:178] from 1993 to 2008: 245872 238014 263227 277991 286691 ...
```

```r
plot(trafic_month, main = "Monthly trafic", xlab = "year", ylab = "Trafic")
```

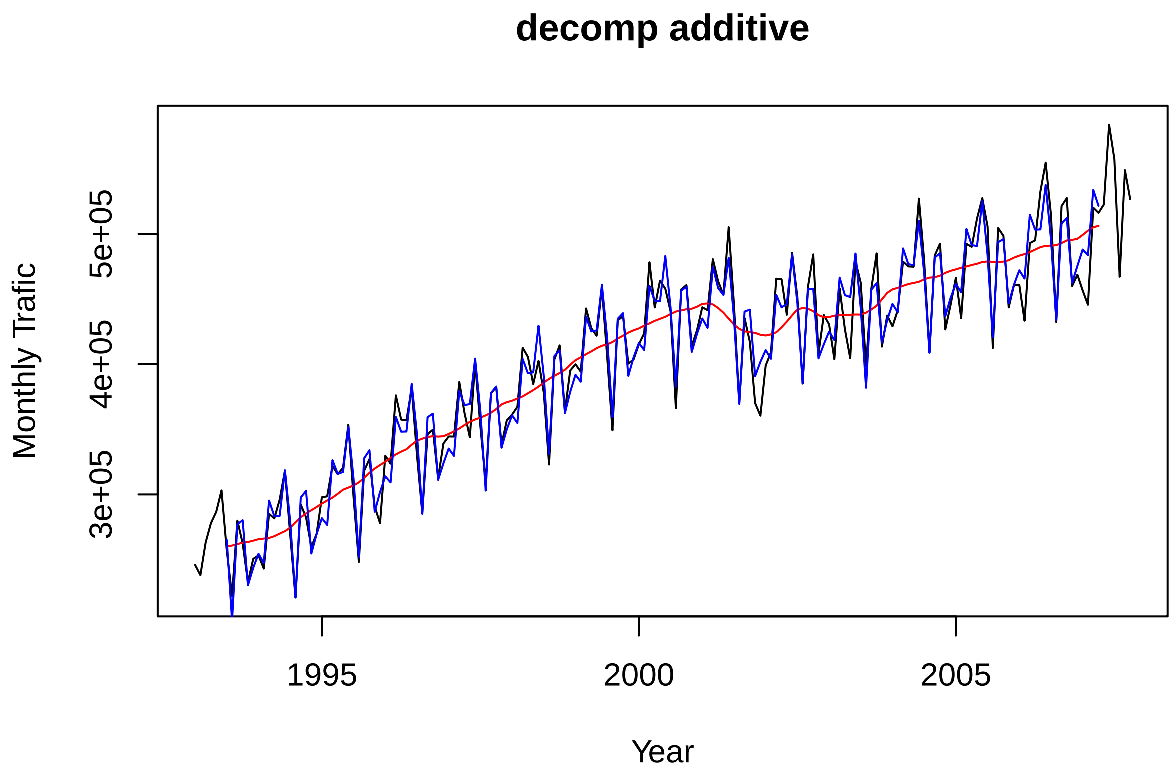



### Data exploration

```
# additive  decomposition
decomp.add = decompose(trafic_month, type = "additive")
plot(decomp.add)
```




```r
# Plot of the time series with additive trend and saisonnality
plot(trafic_month, xlab = "Year", ylab = "Monthly Trafic", main = "decomp additive")
points(decomp.add$trend, type = "l", col = "red")
points(decomp.add$trend + decomp.add$seasonal, type = "l", col = "blue")
```

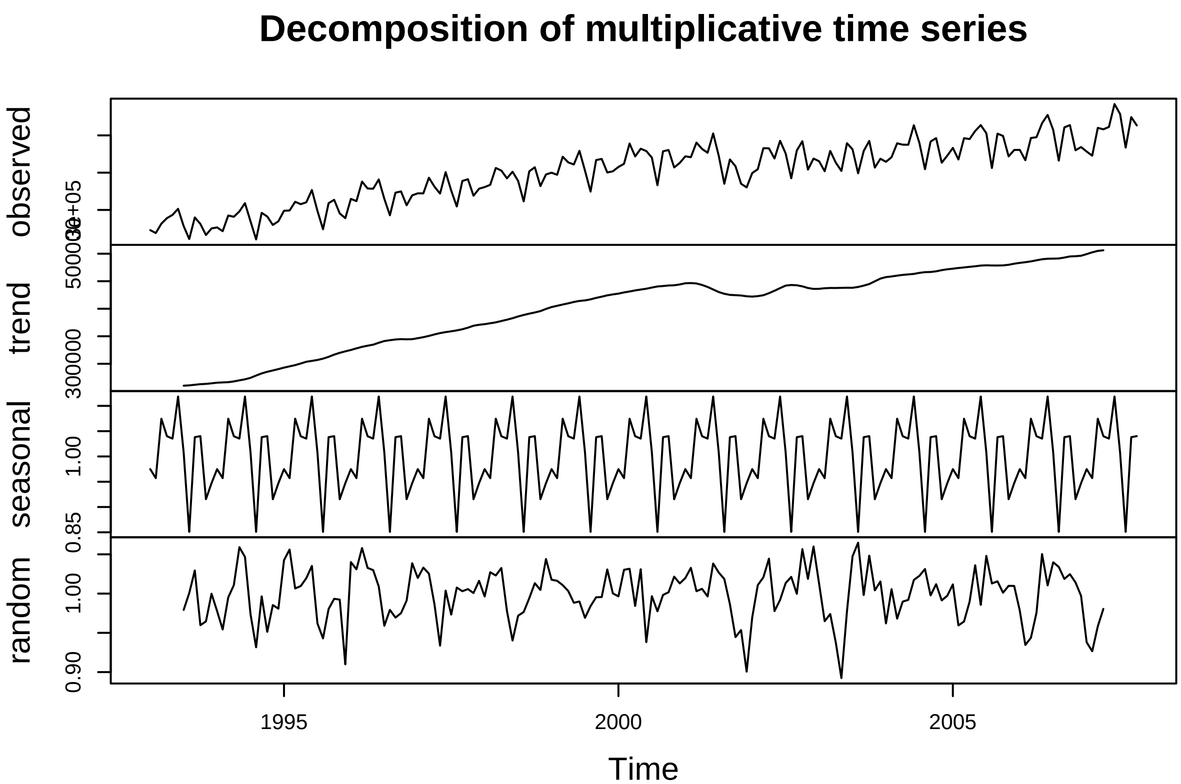



```r
ljung_box_test <- Box.test(decomp.add$random, lag = 5, type = "Ljung-Box")
ljung_box_test
```

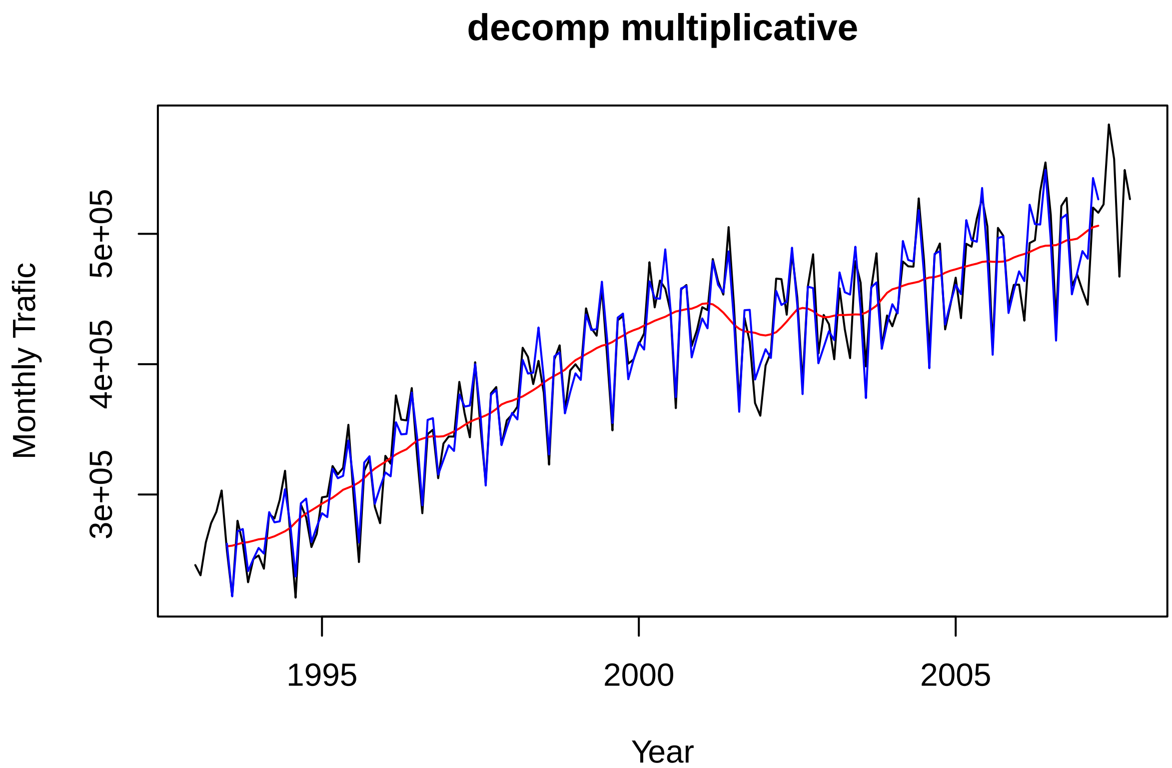
```
##
##  Box-Ljung test
##
## data:  decomp.add$random
## X-squared = 65.557, df = 5, p-value = 8.588e-13
```

the p_value is very small than 0.05 that mean the noise is correlate therefore this decomposition is not accurate

```r
# multiplicative decomposition
decomp.multi = decompose(trafic_month, type = "multiplicative")
plot(decomp.multi)
```




```r
# Plot of the time series with multicative trend and saisonnality
plot(trafic_month, xlab = "Year", ylab = "Monthly Trafic", main = "decomp multiplicative")
points(decomp.multi$trend, type = "l", col = "red")
points(decomp.multi$trend * decomp.multi$seasonal, type = "l", col = "blue")
```




```
ljung_box_test_multi <- Box.test(decomp.multi$random, lag = 5, type = "Ljung-Box")
ljung_box_test_multi
```

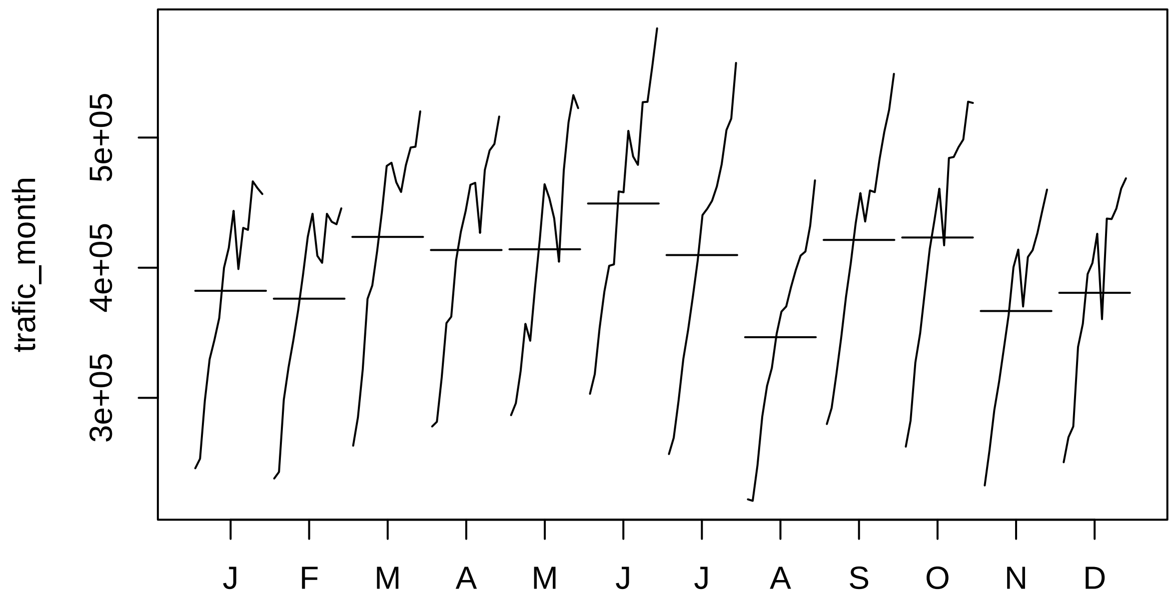
```
##
##  Box-Ljung test
##
## data:  decomp.multi$random
## X-squared = 72.491, df = 5, p-value = 3.109e-14
```

the p_value is very small (<0.05) that mean the noise is correlate therefore this decomposition is not accurate.

If the two decomposition is not accurate that means we need to explore more our data. let's do some monthplot and lagplot

## Monthplot

```
monthplot(trafic_month)
```




We can see that our time series is increasing over time. Also, the graph for each month is similar, which indicates that we have seasonality. We can see that the month of June has the highest traffic, while August has the lowest activity. We notice a drop in traffic, shown by downturns in the series, which are consecutive to the 9/11 attacks.

```
trafic_00_02 = window(trafic_month, start = c(2000,1), end = c(2002,12))
trafic_00_02
```

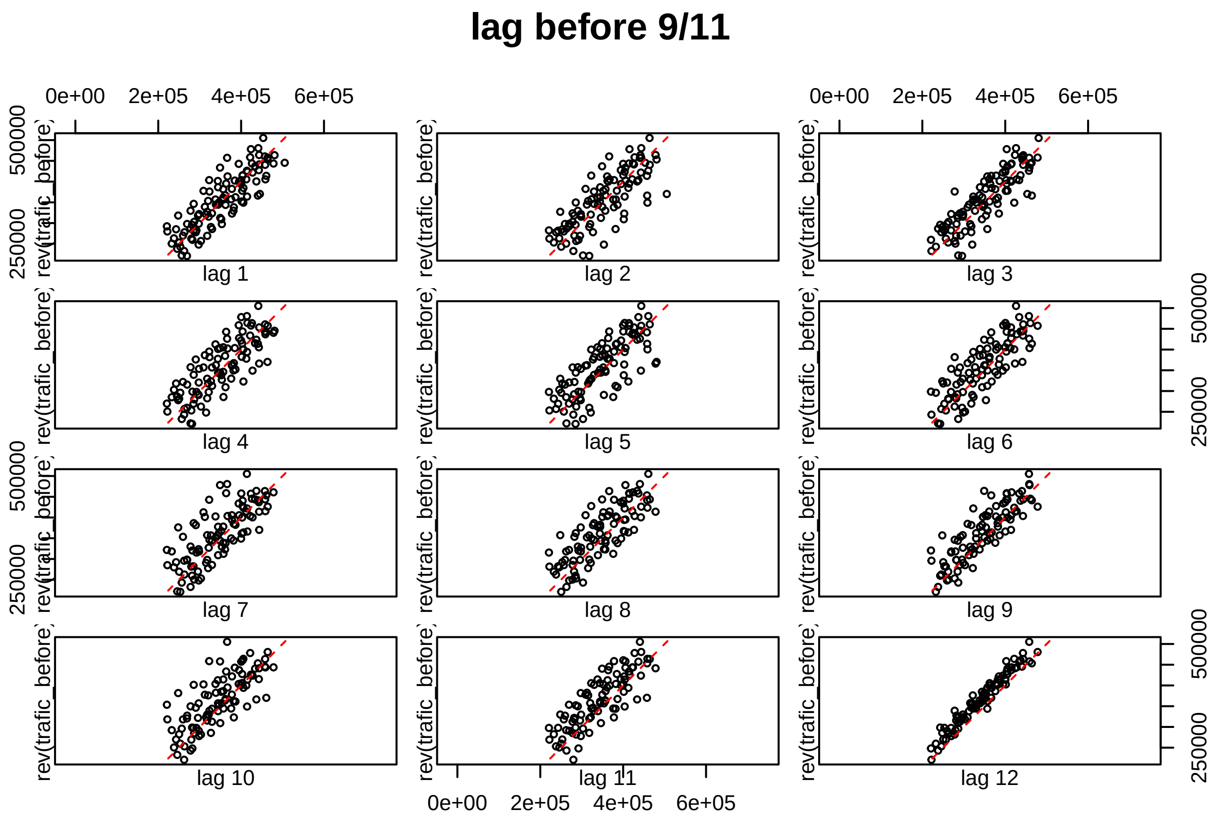
```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2000 415292 423665 478207 443548 464162 457944 440436 366272 457318 460735
## 2001 443700 441499 480649 463680 453372 505190 445332 370211 435473 417169
## 2002 398975 409142 465646 465236 437930 485439 451417 385078 459356 484329
##         Nov    Dec
## 2000 413933 426097
## 2001 370169 360457
## 2002 408187 437763
```

we can see that after 09/11 the trafic drop every month until July 2002 when the trafic began to stablize and the trafic begin to take the increasing trend again

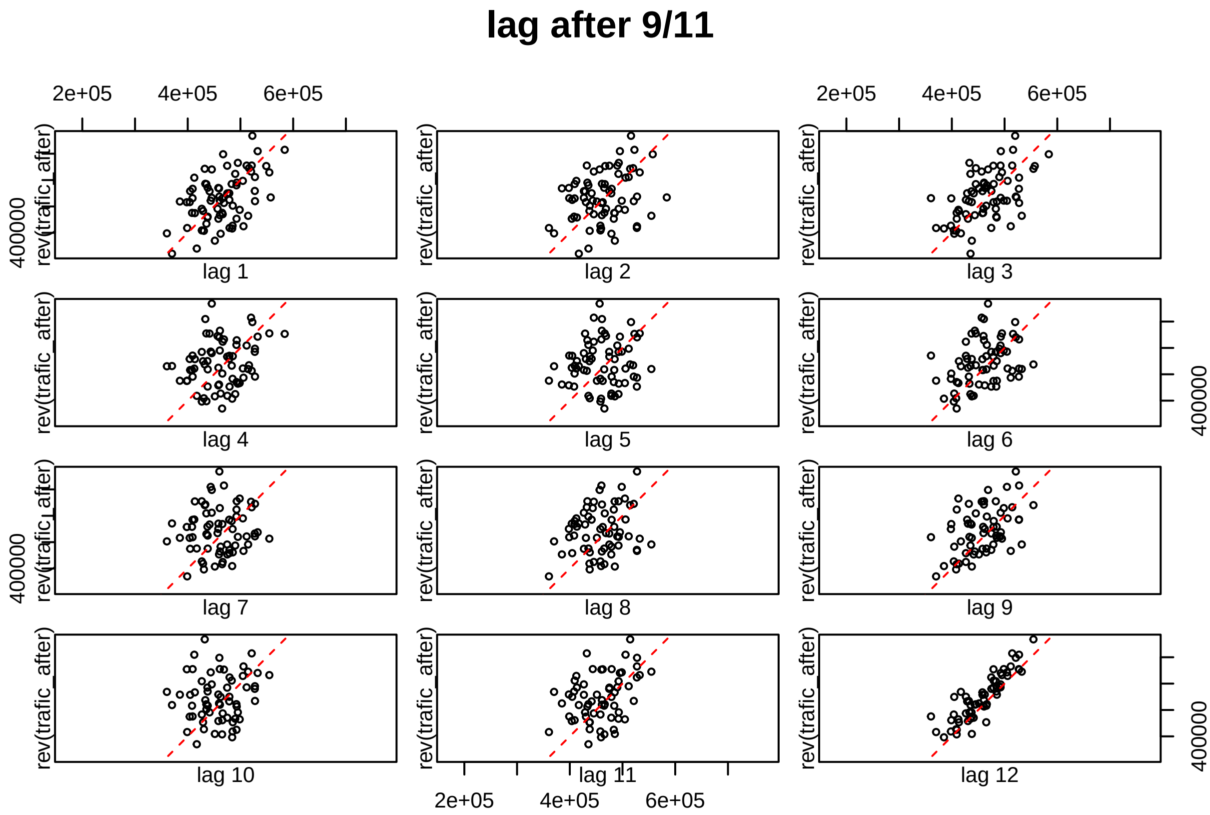
### Lagplot

```
trafic_before = window(trafic_month, start  = c(1993,1), end = c(2001,8))
trafic_after = window(trafic_month, start = c(2001,9), end = c(2007,10))
```

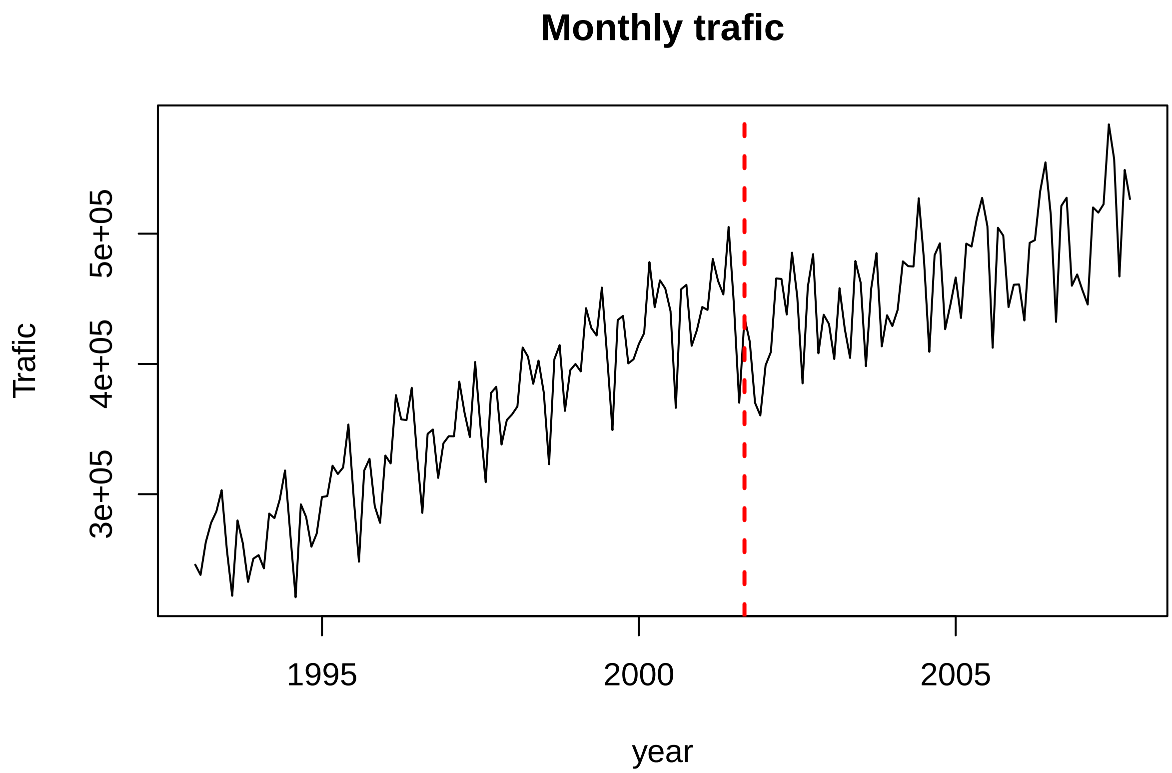
```r
lag.plot(rev(trafic_before), set.lags = 1:12, asp=1, diag = TRUE, diag.col = "red",
         type ="p", do.lines = FALSE, main = "lag before 9/11")
```




```r
lag.plot(rev(trafic_after), set.lags = 1:12, asp=1, diag = TRUE, diag.col = "red",
         type ="p", do.lines = FALSE, main = "lag after 9/11")
```



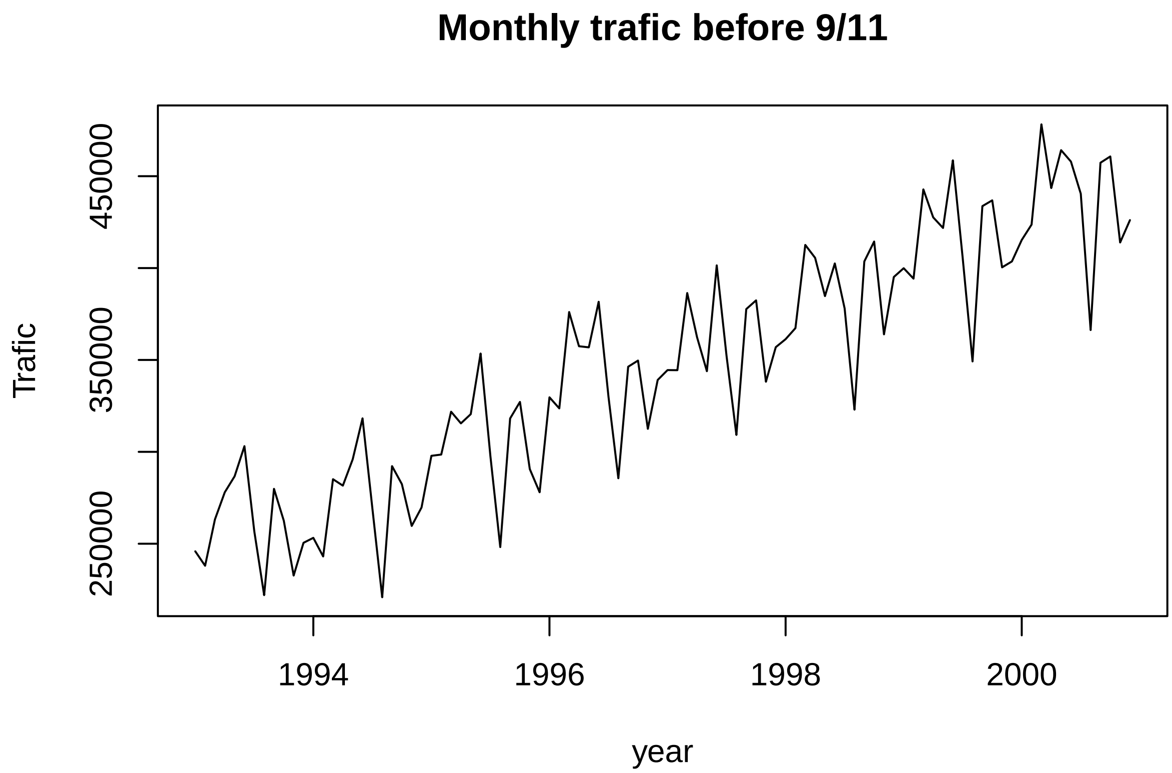

```r
plot(trafic_month, main = "Monthly trafic", xlab = "year", ylab = "Trafic")
abline(v=2001 + 8/12, col="red", lwd=2, lty=2)
```



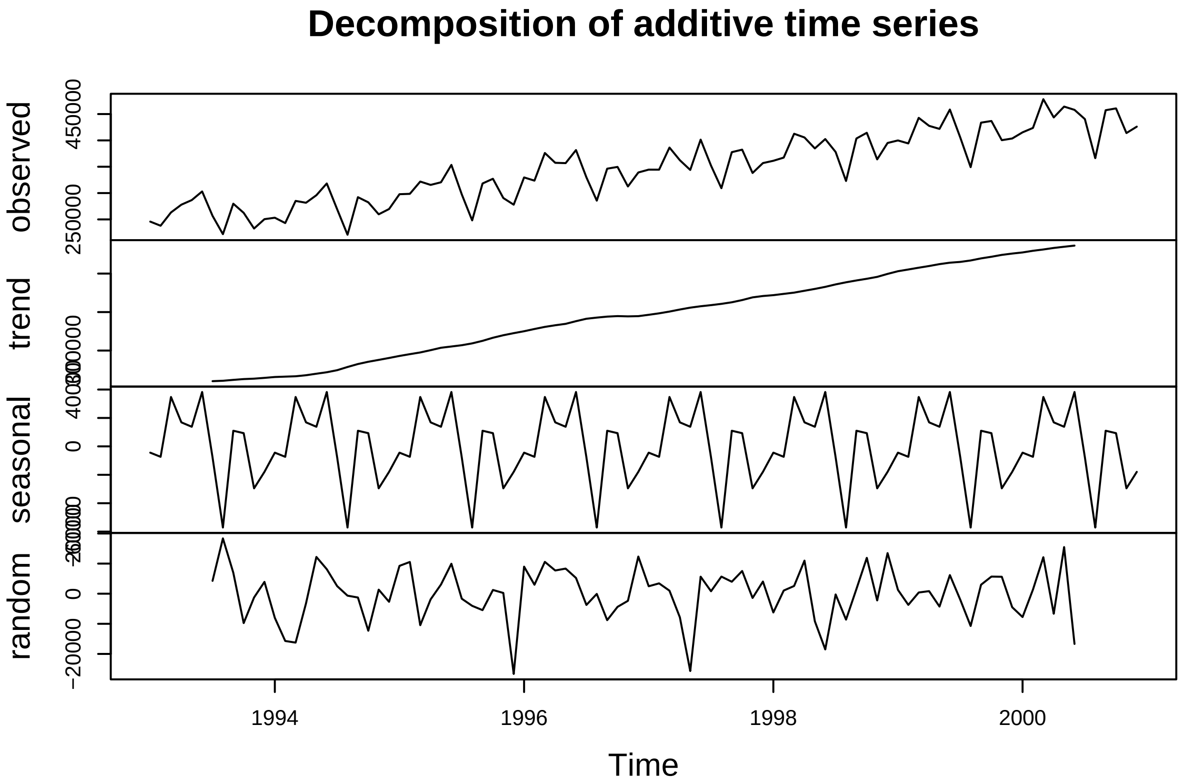

We can observe that before 09/11 for each month the time series is very corelated and with a strong corelation between the month with lag 12 (January 2002 wwith Jannuary 2023). And after 09/11 the serie is much less corelated and we only have one strong corelation with lag 12. the times series before 09/11 and after doesn't behave the same therefore we will consider two times series one before 09/11 and the other aafter 09/11

### Time series before 9/11

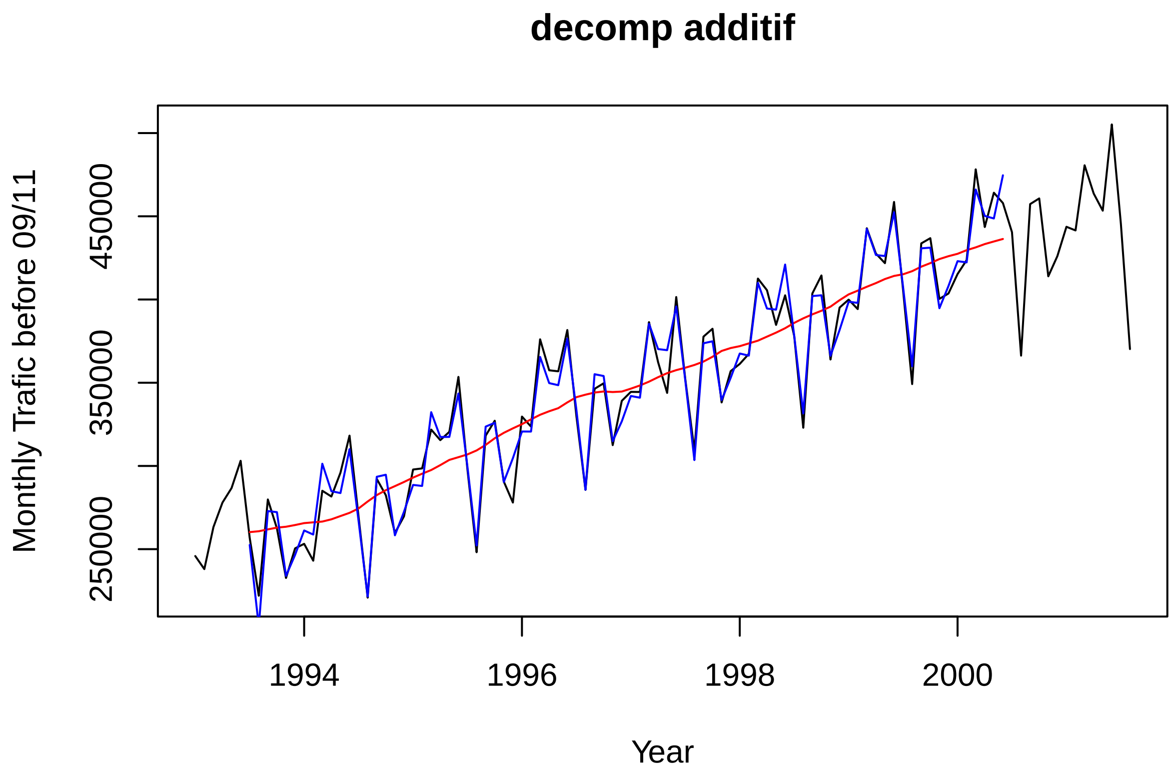
```
train_data_before = window(trafic_month, start  = c(1993,1), end = c(2000,12))
test_data_before = window(trafic_before, start = c(2001,1), end = c(2001,8))

plot(train_data_before,
     main = "Monthly trafic before 9/11",
     xlab = "year", ylab = "Trafic")
```




```
# additive  decomposition
decomp.add = decompose(train_data_before, type = "additive")
plot(decomp.add)
```




```r
# Plot of the time series with additive trend and saisonnality
plot(trafic_before,
     xlab = "Year",
     ylab = "Monthly Trafic before 09/11",
     main = "decomp additif")

points(decomp.add$trend, type = "l", col = "red")
points(decomp.add$trend + decomp.add$seasonal, type = "l", col = "blue")
```



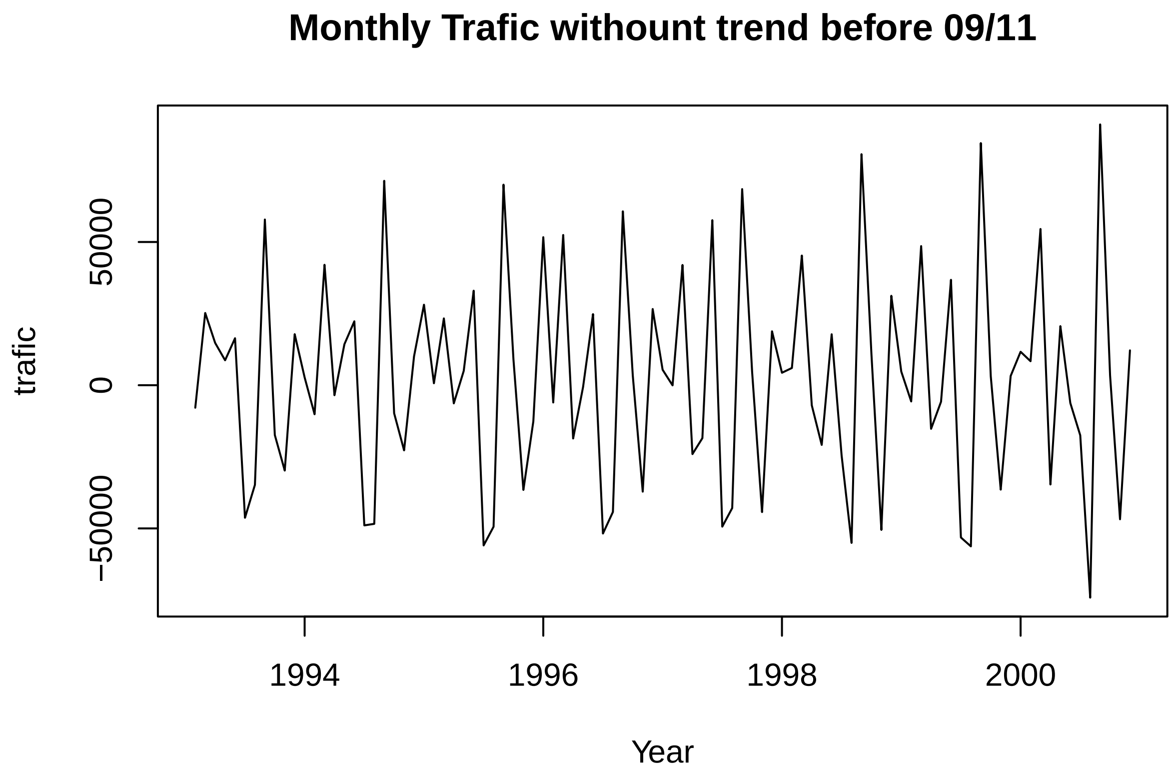

```
ljung_box_test <- Box.test(decomp.add$random, lag = 5, type = "Ljung-Box")
ljung_box_test
```

```
##
##  Box-Ljung test
##
## data:  decomp.add$random
## X-squared = 8.346, df = 5, p-value = 0.1382
```

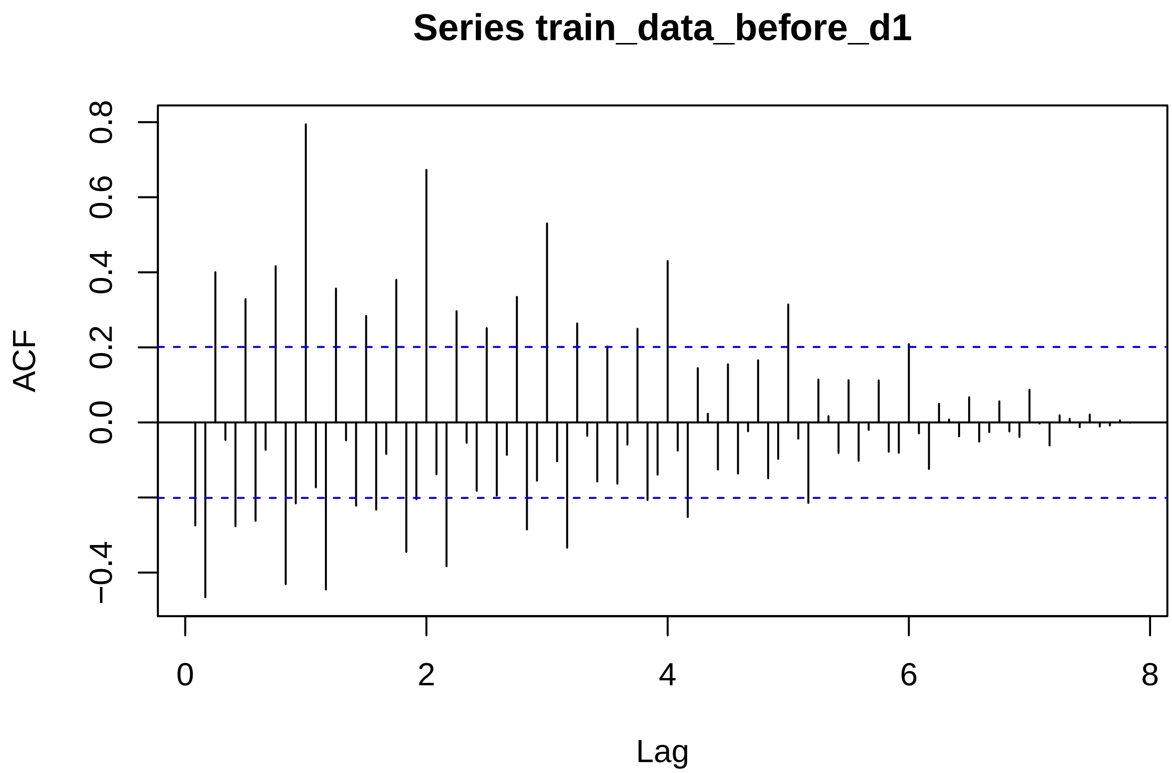
This decomposition is accurate because the pvalue is greater than 0.05

### ACF and PACF

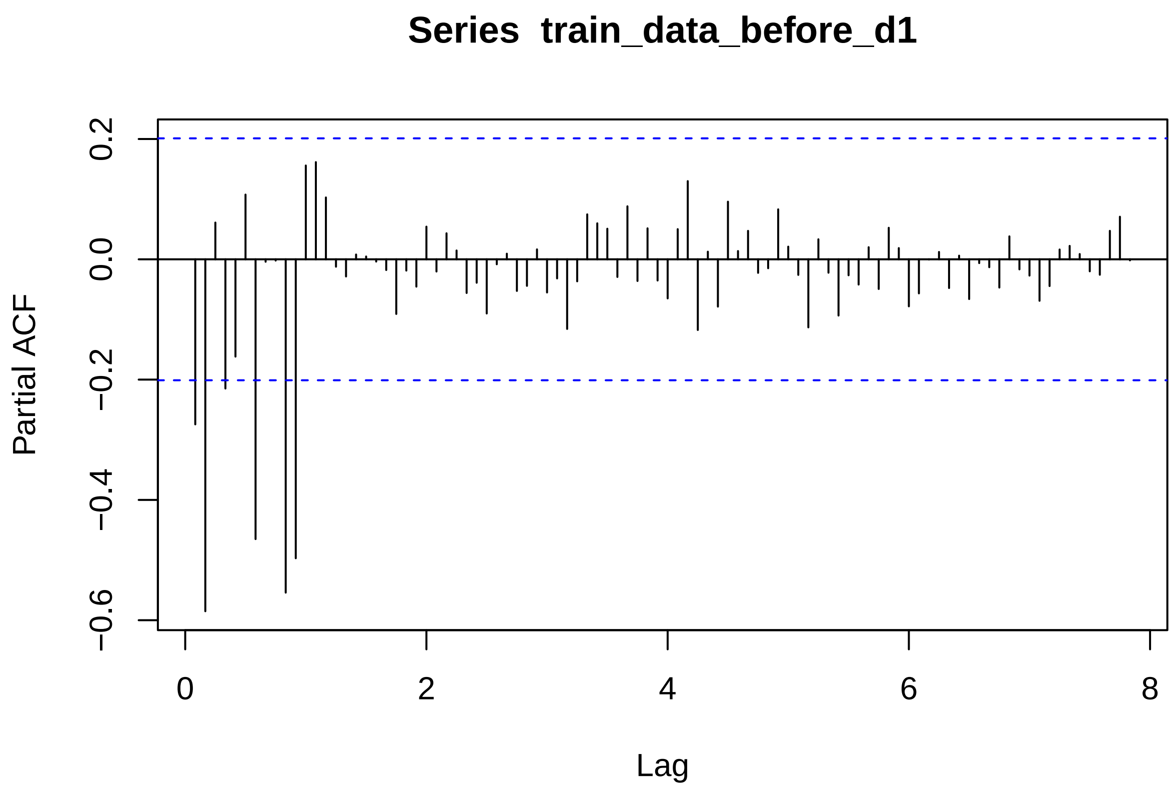
```
# first difference to remove the trend
train_data_before_d1 = diff(train_data_before, differences = 1)
plot(train_data_before_d1, xlab = "Year", ylab = "trafic",
     main = "Monthly Trafic withount trend before 09/11")
```



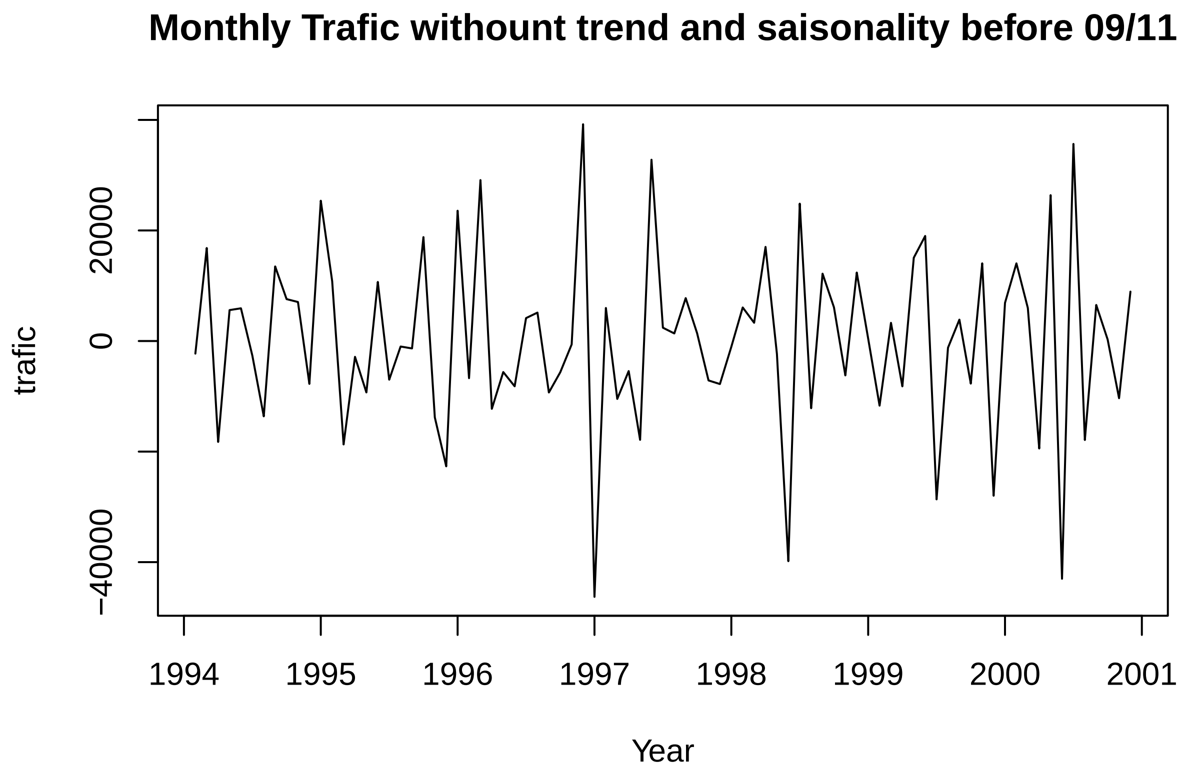

```r
acf(train_data_before_d1, lag.max = 100)
```



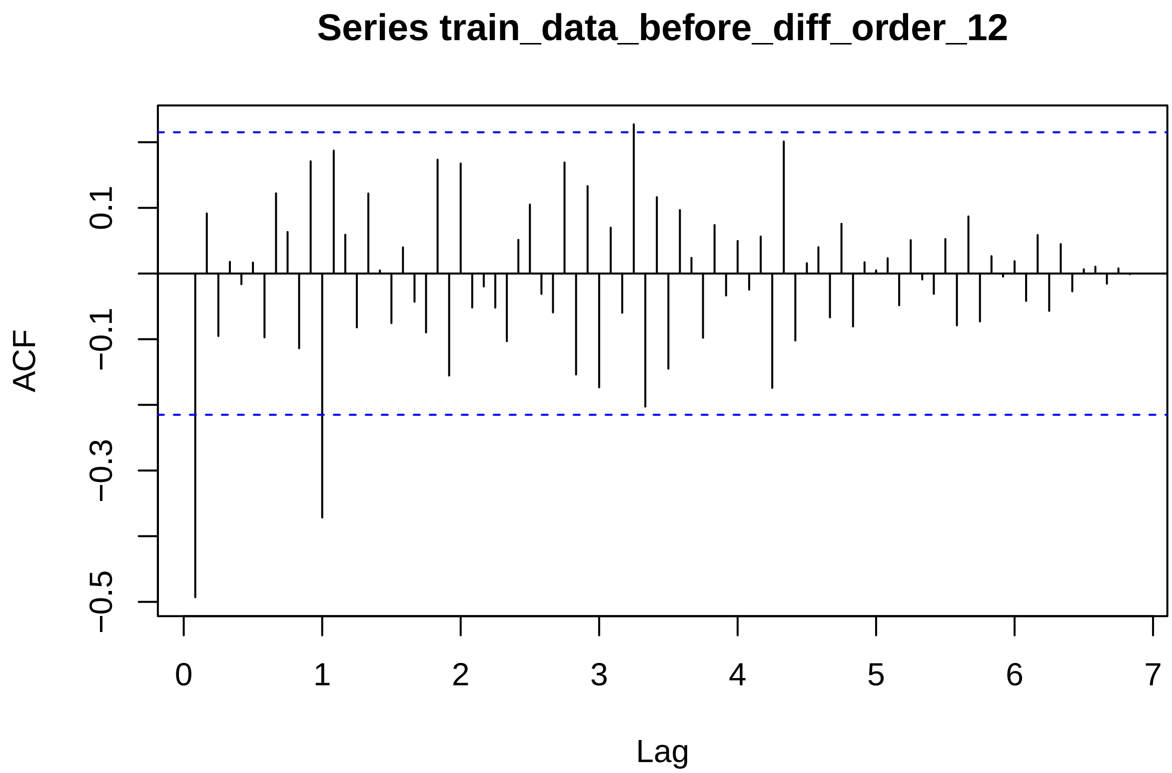

```r
pacf(train_data_before_d1, lag.max = 100)
```



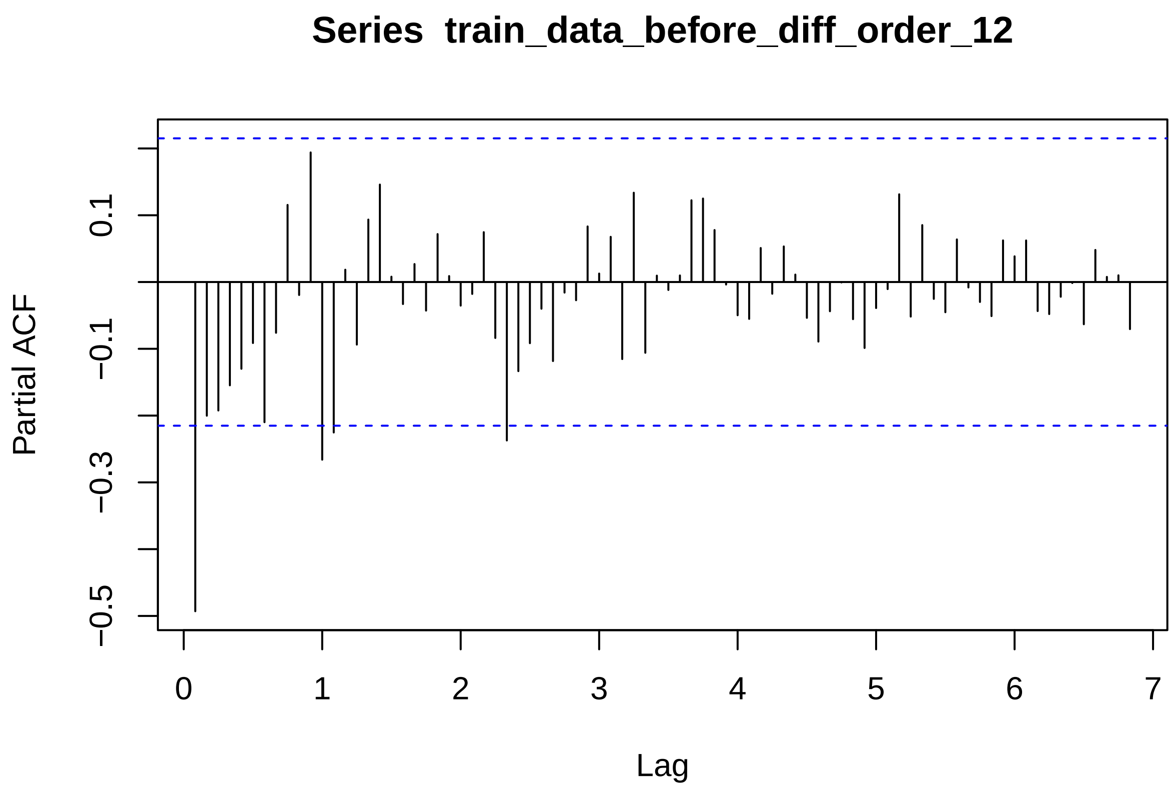

```r
# remove saisonality
train_data_before_diff_order_12 = diff(train_data_before_d1,
                                       lag = 12,
                                       differences = 1)

plot(train_data_before_diff_order_12, xlab = "Year", ylab = "trafic",
     main = "Monthly Trafic withount trend and saisonality before 09/11")
```




```
acf(train_data_before_diff_order_12, lag.max = 100)
```




```
pacf(train_data_before_diff_order_12, lag.max = 100)
```




```r
# Stationary test

adf.test(train_data_before_diff_order_12)
```

```
## Warning in adf.test(train_data_before_diff_order_12): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_data_before_diff_order_12
## Dickey-Fuller = -5.9347, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

the pvalue is smaller than 0.01 therefore we reject the null hypothesis therefore this series is stationary.

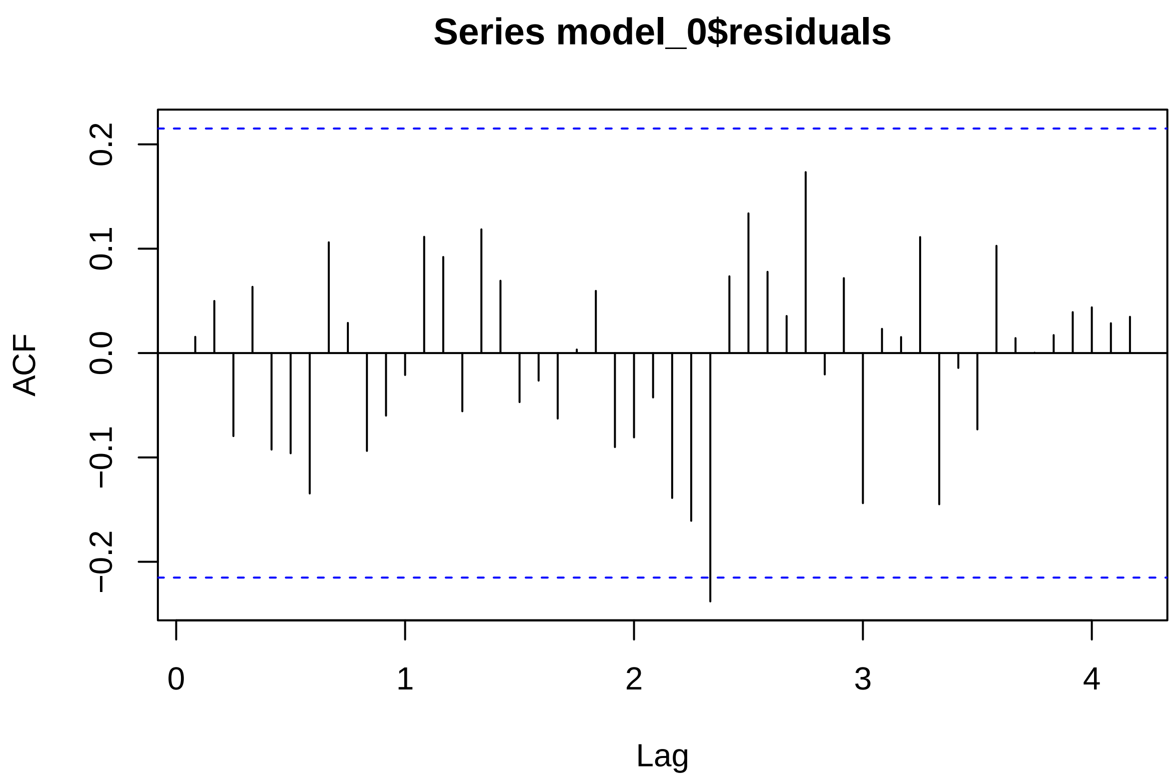
### Manually research

```r
model_0 = Arima(train_data_before_diff_order_12,
                order = c(0,0,1),
                seasonal=list(order=c(1,0,0),
                period=12,
                include.drift=TRUE))
```

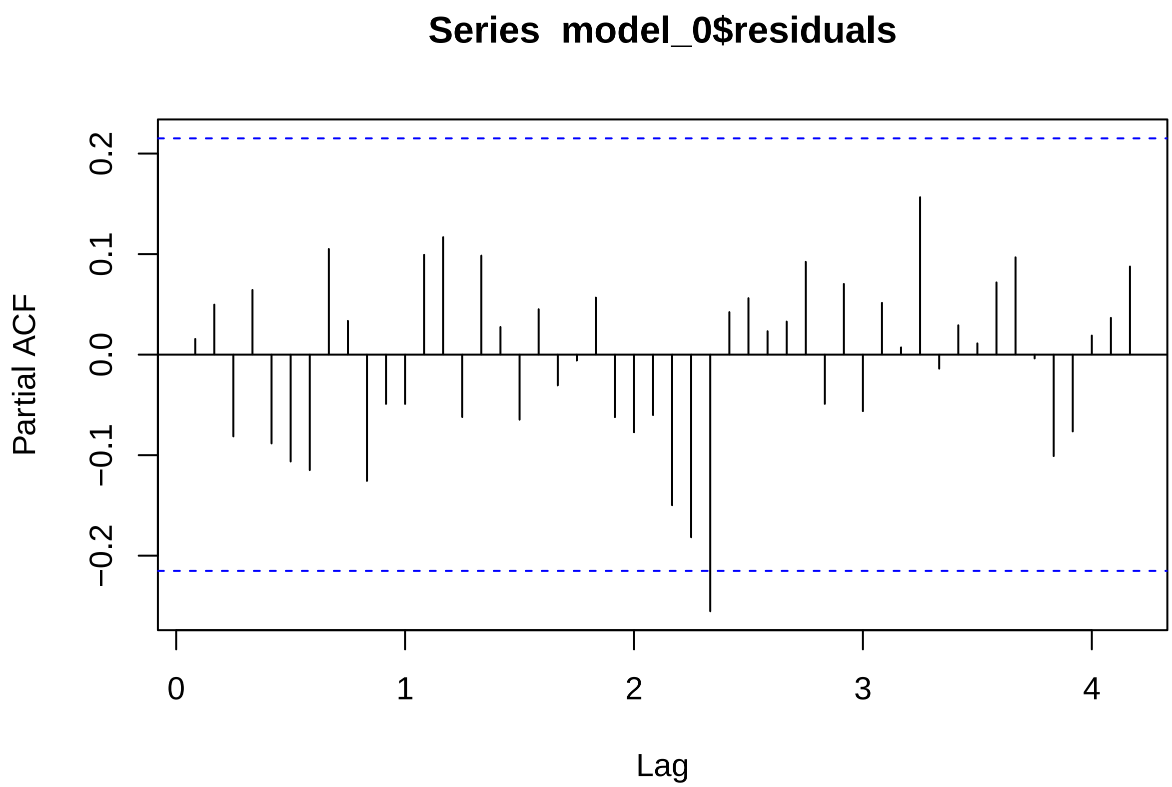
```r
summary(model_0)
```

```
## Series: train_data_before_diff_order_12
## ARIMA(0,0,1)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##           ma1     sar1      mean
##       -0.7097  -0.4139   41.6182
## s.e.   0.0980   0.1046  292.4611
##
## sigma^2 = 148631807:  log likelihood = -898.63
## AIC=1805.27   AICc=1805.78   BIC=1814.94
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE      ACF1
## Training set 428.3273 11969.11 9591.085 134.0264 185.1751 0.4768747 0.0155552
```

```r
acf(model_0$residuals, lag.max = 50)
```




```r
pacf(model_0$residuals, lag.max = 50)
```




```r
#  Ljung-Box on the residus
Box.test(model_0$residuals, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  model_0$residuals
## X-squared = 6.4454, df = 10, p-value = 0.7766
```

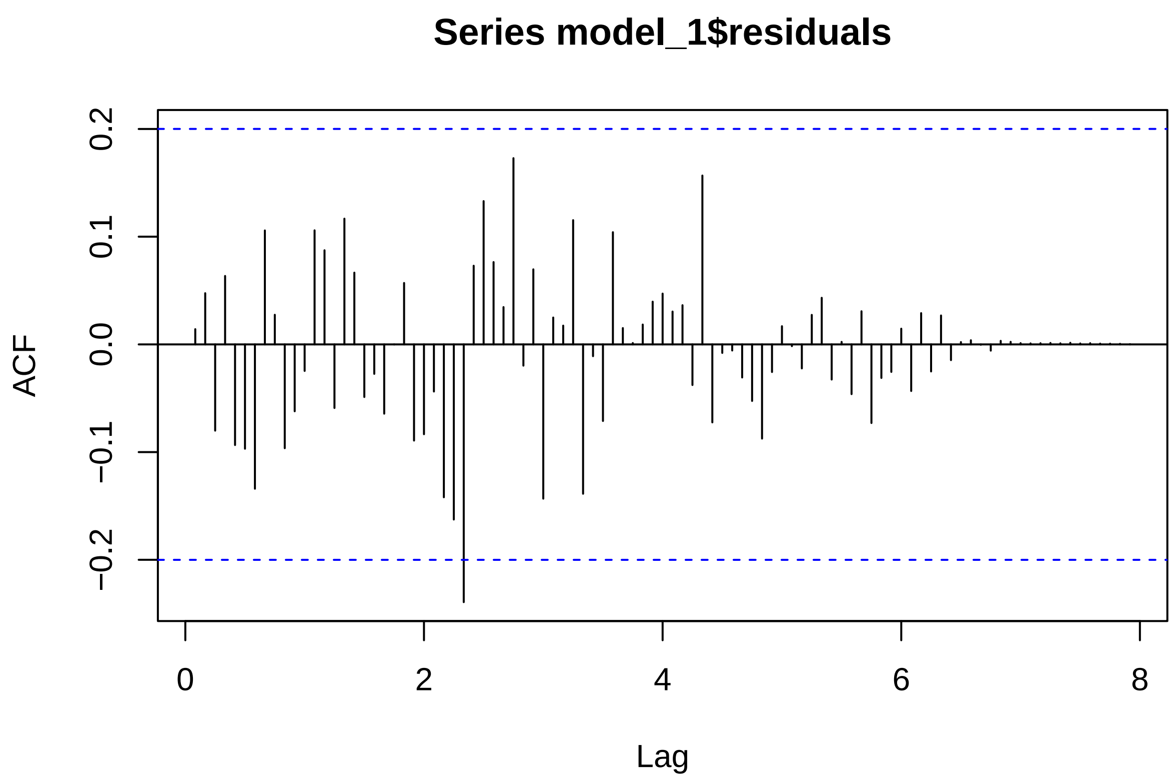
The pvalue is greater than 0.05 therefore this model is accurate but we can improve it

```r
model_1 = Arima(train_data_before, order = c(0,1,1),
                seasonal=list(order=c(1,1,0),method='ML',
                period=12,
                include.drift=TRUE))
summary(model_1)
```

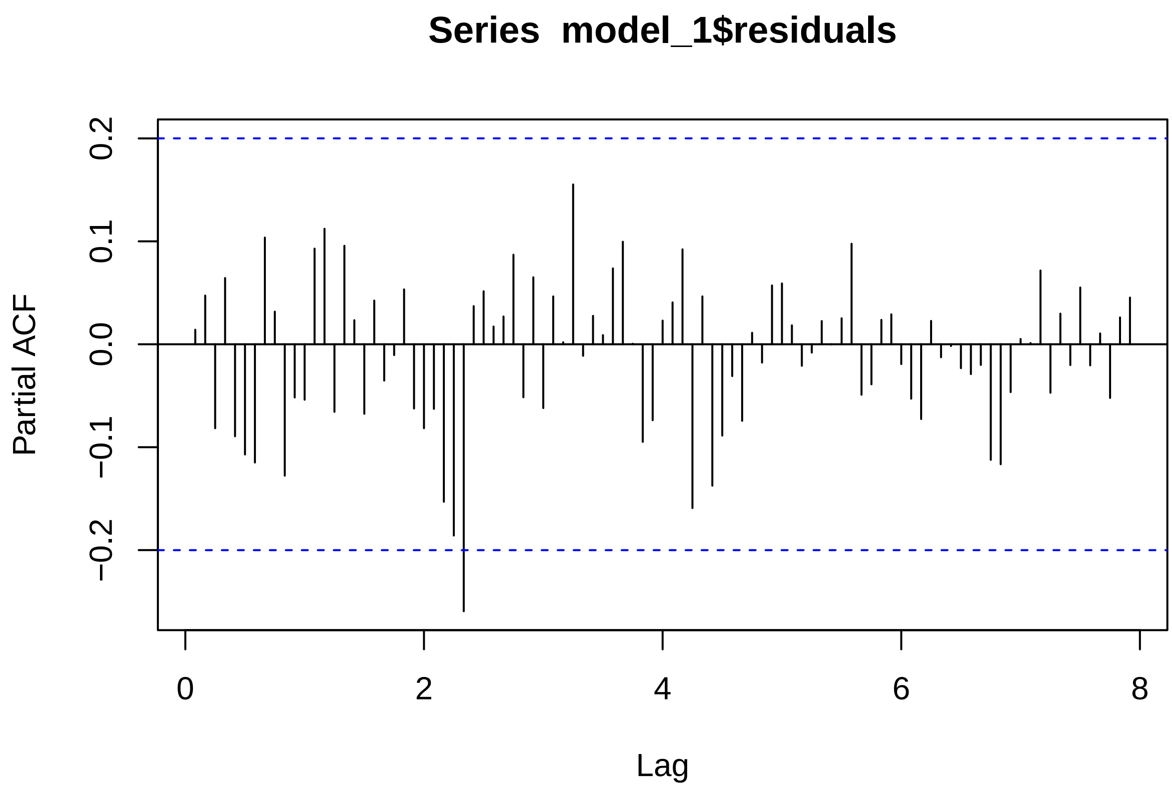
```
## Series: train_data_before
## ARIMA(0,1,1)(1,1,0)[12]
##
## Coefficients:
##           ma1     sar1
##       -0.7084  -0.4151
## s.e.   0.0976   0.1044
```

```
##
## sigma^2 = 146822037:  log likelihood = -898.64
## AIC=1803.29   AICc=1803.59   BIC=1810.54
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 519.6383 11130.19 8302.327 0.1353545 2.371724 0.3231712 0.01414065
```

```r
acf(model_1$residuals, lag.max =100)
```




```r
pacf(model_1$residuals, lag.max = 100)
```




```
Box.test(model_1$residuals, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  model_1$residuals
## X-squared = 7.3847, df = 10, p-value = 0.6887
```

the Pvalue is greater than 0.05 therefore this model is accurate also his AIC is smaller than the previous model. Now let's see what model the auto arima model will give us.

```
best_model = auto.arima(train_data_before, stepwise = FALSE, approximation = FALSE)
best_model
```

```
## Series: train_data_before
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1      drift
##       0.2141  -0.6057  2176.2651
## s.e.  0.1080   0.1261    66.5513
##
## sigma^2 = 128350451:  log likelihood = -904.58
## AIC=1817.15   AICc=1817.66   BIC=1826.88
```

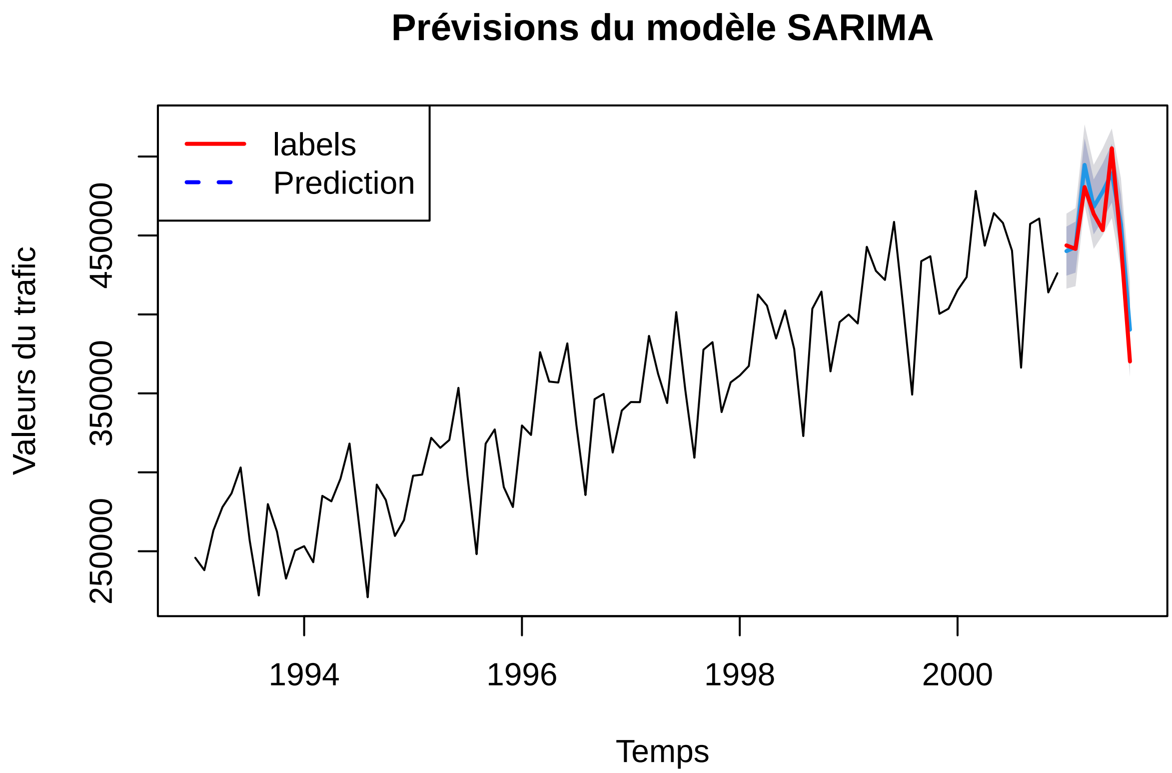
```
Box.test(best_model$residuals, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  best_model$residuals
## X-squared = 6.9957, df = 10, p-value = 0.7259
```

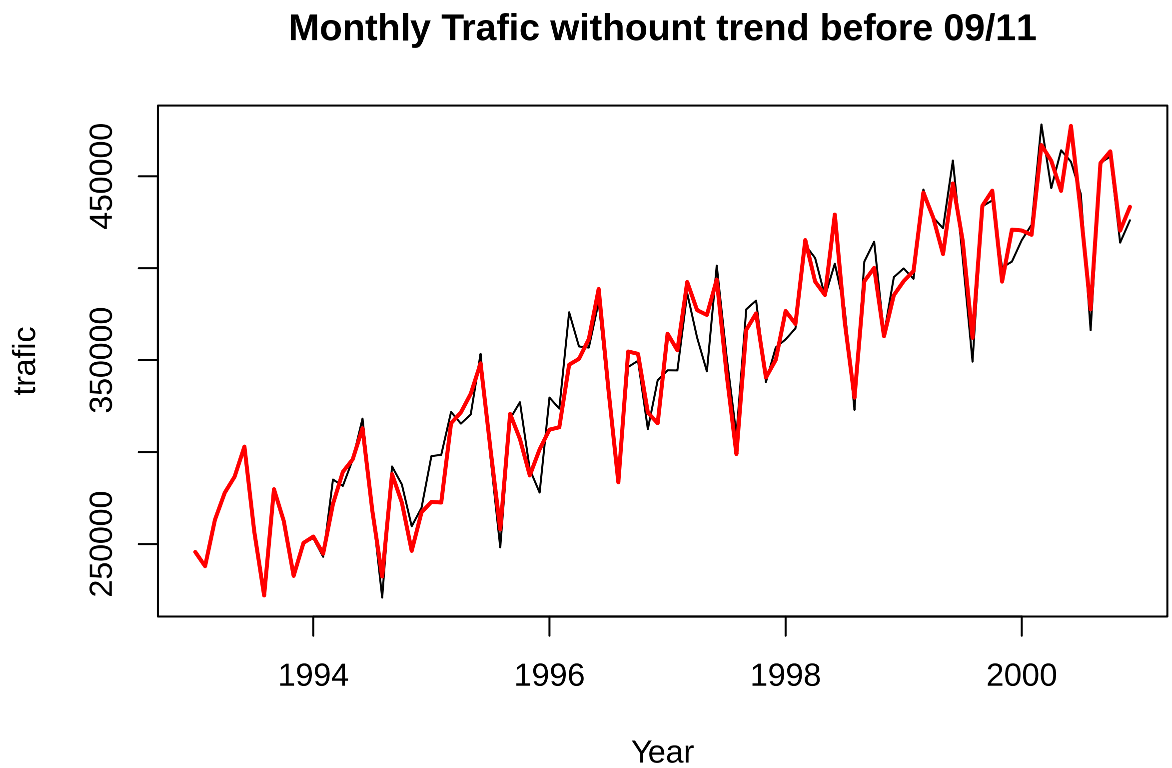
Our manual model is better than the auto arima model because his AIC is smaller and overall our manual model is less complex than the auto arima model

### Forecasting and evaluation of the model

```
n_forecast = 8  # number of month to forecast
forecasted_values = forecast(model_1, h = n_forecast)

# plot forecast
plot(forecasted_values, main = "Prévisions du modèle SARIMA", ylab = "Valeurs du trafic", xlab = "Temps

# Add the data since  1993
lines(test_data_before, col = "red", lwd = 2)

# Add legend
legend("topleft", legend = c("labels", "Prediction"),
       col = c("red", "blue"), lwd = 2, lty = c(1, 2))
```




```r
ec80 = model_1$sigma2^.5 * qnorm(0.90)
vajust = fitted(model_1)
matri = as.ts(cbind(trafic_before,vajust-ec80,vajust+ec80),
              start=c(1993,1),
              frequency=12)
par(oma=rep(0,4))
#plot(matri, plot.type='single', lty=c(1,2,2), xlab="temps",
 #    ylab='trafic',main="",cex.main=0.8 )

plot(train_data_before, xlab = "Year", ylab = "trafic",
     main = "Monthly Trafic withount trend before 09/11")

# Ajouter les valeurs ajustées en rouge
lines(vajust, col = "red", lwd = 2)  # Tracer les valeurs ajustées
```




```r
#legend( par("usr")[1], par("usr")[4], c("Valeur observ´ee","Bande de pr´ediction"),lwd=1,lty=c(1,2))


predicted_values <- forecasted_values$mean

metrics_SARIMA_before = c(
  MAPE = mape(test_data_before, predicted_values),
  MAE  = mae(test_data_before, predicted_values),
  RMSE = rmse(test_data_before, predicted_values)
)

metrics_SARIMA_before
```

```
##         MAPE          MAE         RMSE
## 2.691042e-02 1.190919e+04 1.422557e+04
```

```r
indi = (trafic_before - (vajust-ec80))>0 & (vajust+ec80 - trafic_before) > 0
prop = 100*sum(indi)/length(indi)
prop
```

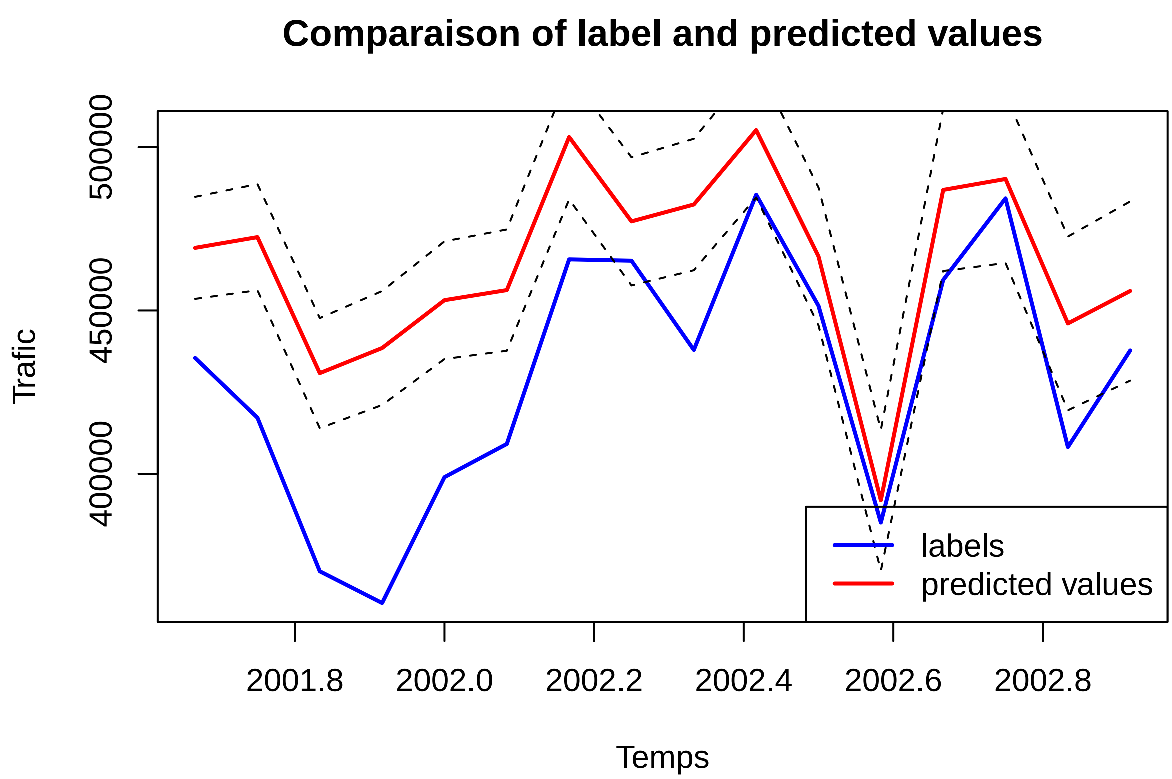
```
## [1] 86.45833
```

86.45% of the predictions are within the confidence interval, which means the model is neither underfitted nor overfitted, which is good.

```r
final_model_before = Arima(trafic_before, order = c(0,1,1),
                          seasonal=list(order=c(1,1,0),method='ML',
                                        period=12,
                                        include.drift=TRUE))
```

### comparing the model with the real value after 09/11 to calculate the loss

```r
data_2002 = window(trafic_month, start  = c(2001,9), end = c(2002,12))

forecasted_values_2002 = forecast(final_model_before, h = 16,level = 80)

plot(data_2002, type = "l", col = "blue", lwd = 2,
     xlab = "Temps", ylab = "Trafic", main = "Comparaison of label and predicted values",
     ylim = range(c(data_2002, forecasted_values_2002$mean)))

lines(forecasted_values_2002$mean, col = "red", lwd = 2)

lines(forecasted_values_2002$lower[,1], col = "black", lty = 2)  # Lower bound
lines(forecasted_values_2002$upper[,1], col = "black", lty = 2)  # upper bound

# plot label and predict

legend("bottomright", legend = c("labels", "predicted values"),
       col = c("blue", "red"), lwd = 2)
```

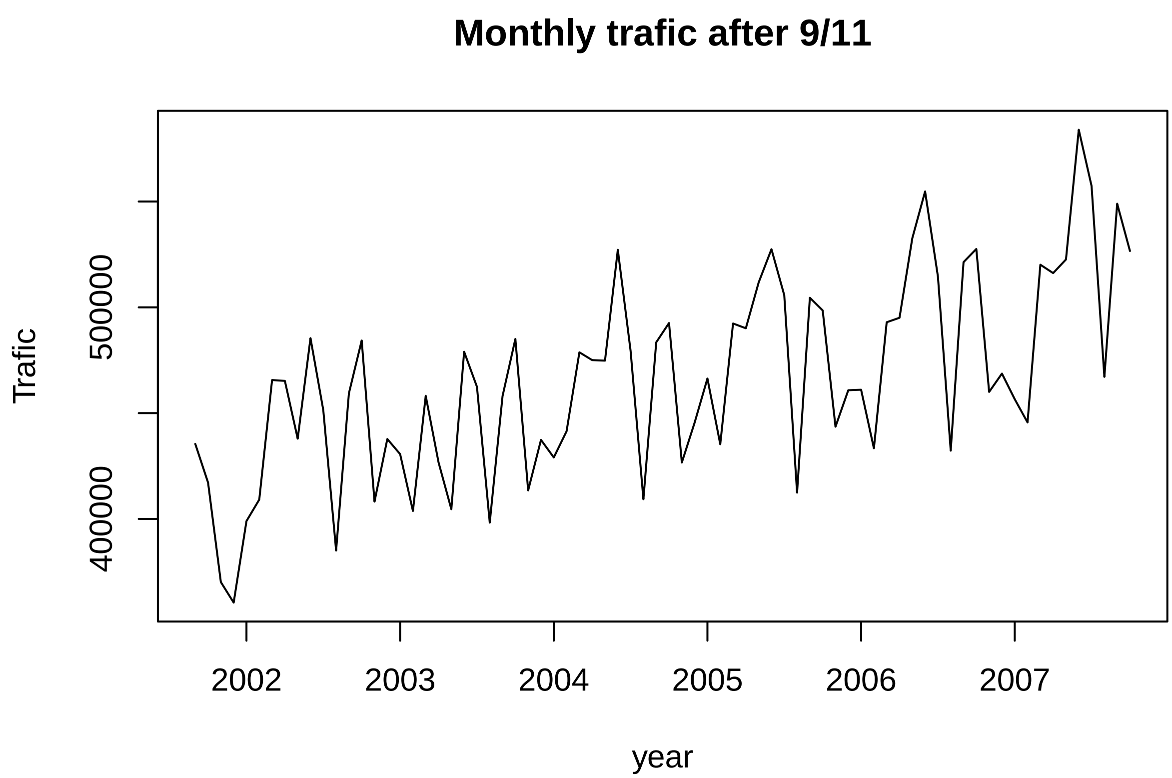


```r
loss_month <- data_2002 - forecasted_values_2002$mean
loss_month
```

```
##              Jan         Feb         Mar         Apr         May         Jun
## 2001
## 2002 -54174.461 -47097.954 -37446.494 -12034.978 -44507.117 -19773.457
##              Jul         Aug         Sep         Oct         Nov         Dec
## 2001                         -33696.001 -55278.383 -60652.778 -78054.479
## 2002 -15130.452  -6827.358 -27547.997  -5922.747 -37848.745 -18201.497
```

### Time series after 9/11

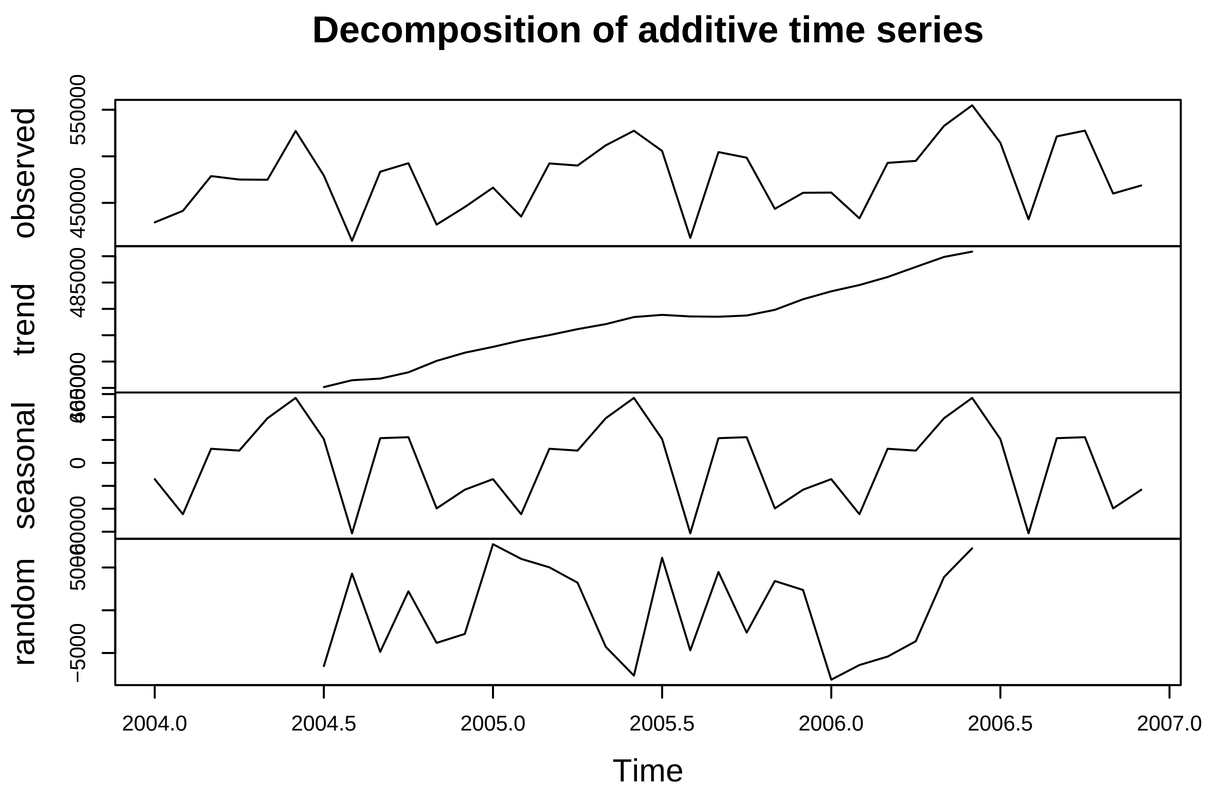
```r
plot(trafic_after,
     main = "Monthly trafic after 9/11",
     xlab = "year", ylab = "Trafic")
```



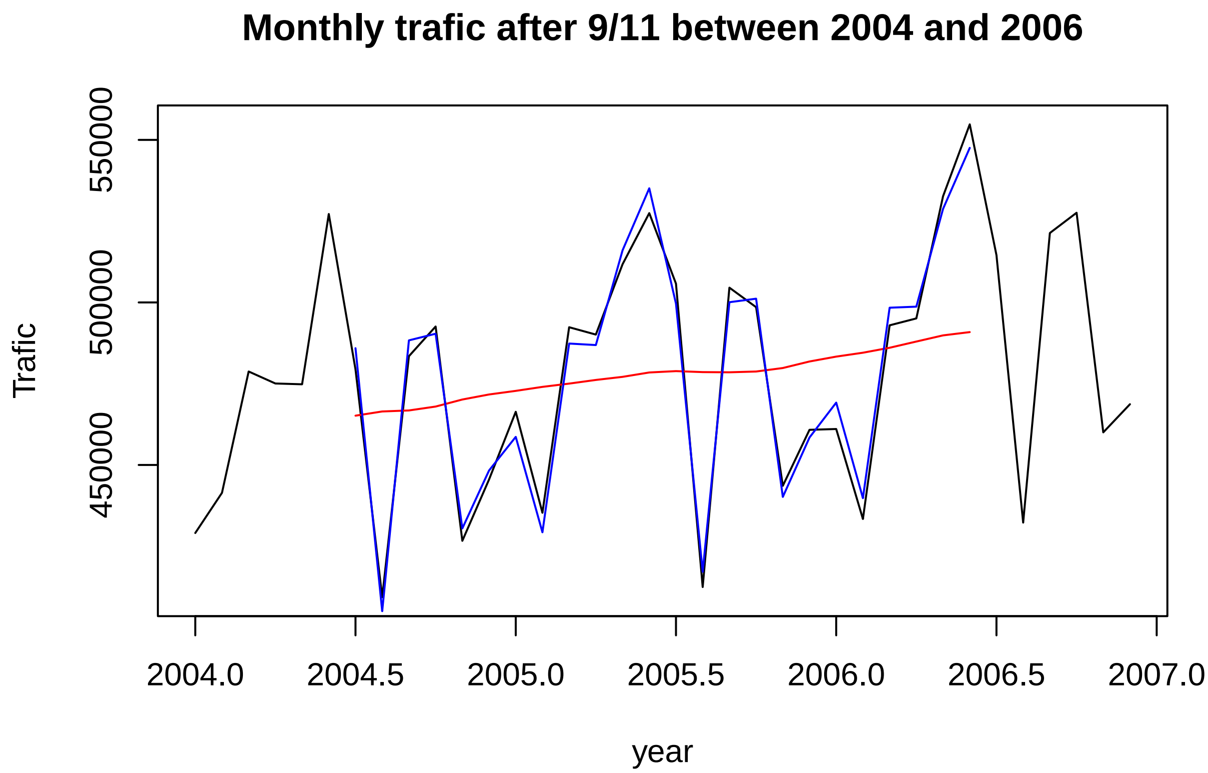

```r
# the time series is very instable before 2004 there fore we will only take the
#serie between 2004 to end of 2007 to build and évaluate out model

train_data_after = window(trafic_after, start = c(2004,1), end = c(2006,12))
test_data_after = window(trafic_after, start = c(2007,1), end = c(2007,10))
```

the time series is very instable before 2004 therefore we will only take the serie between 2004 to end of 2007 to build a model

```
decomp.add = decompose(train_data_after, type = "additive")
plot(decomp.add)
```




```
plot(train_data_after,
     main = "Monthly trafic after 9/11 between 2004 and 2006",
     xlab = "year", ylab = "Trafic")

points(decomp.add$trend, type = "l", col = "red")
points(decomp.add$trend + decomp.add$seasonal, type = "l", col = "blue")
```

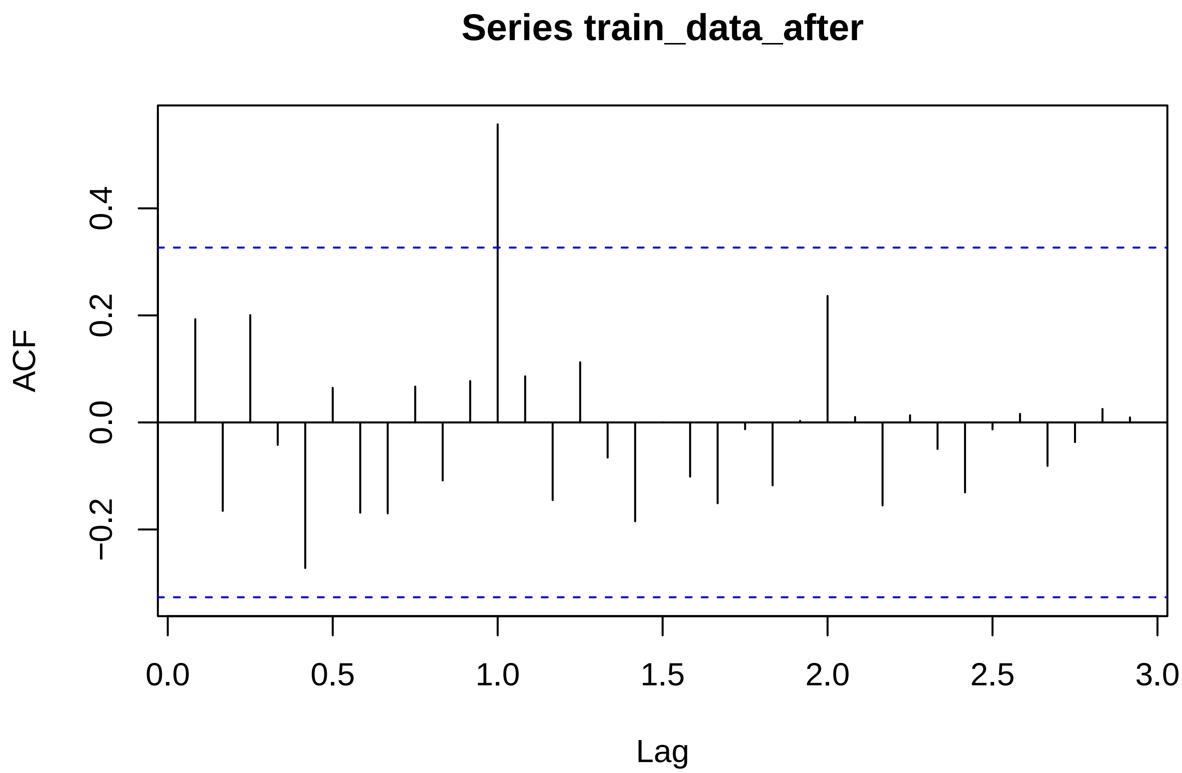



```r
ljung_box_test <- Box.test(decomp.add$random, lag = 5, type = "Ljung-Box")
ljung_box_test
```

```
##
##  Box-Ljung test
##
## data:  decomp.add$random
## X-squared = 5.3682, df = 5, p-value = 0.3726
```

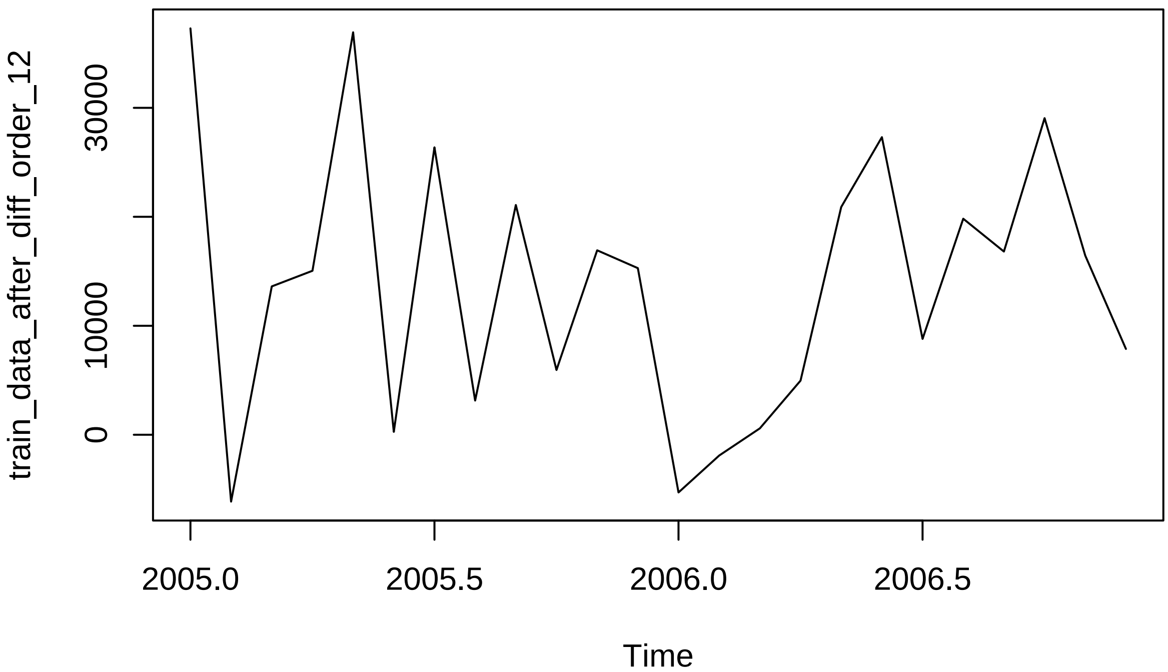
the pval greater than 0.05 therefore this decomposition is accurate

```r
acf(train_data_after, lag.max = 100)
```

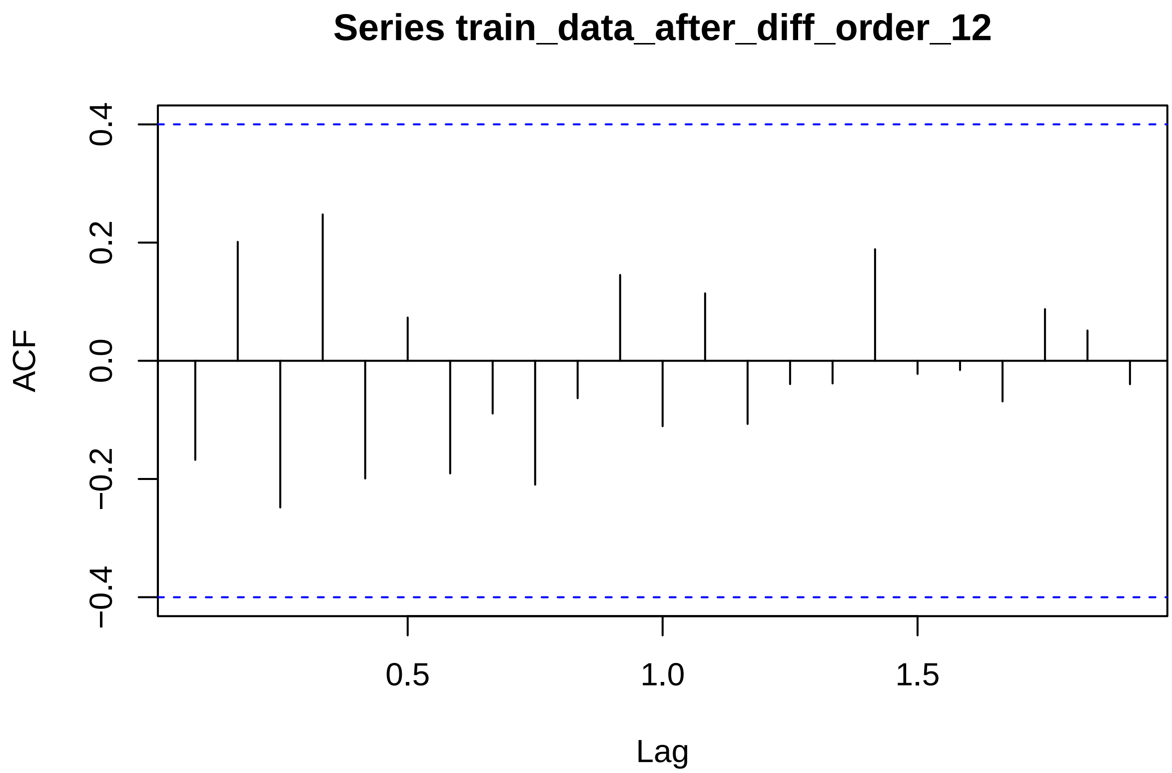



### Building model

```r
# no visible trend we only have saisonnality
train_data_after_diff_order_12 = diff(train_data_after,
                                      lag = 12,
                                      differences = 1)


plot(train_data_after_diff_order_12)
```




```r
acf(train_data_after_diff_order_12, lag.max  = 50)
```




```r
best_model_after = auto.arima(train_data_after, stepwise = FALSE, approximation = FALSE)
best_model_after
```

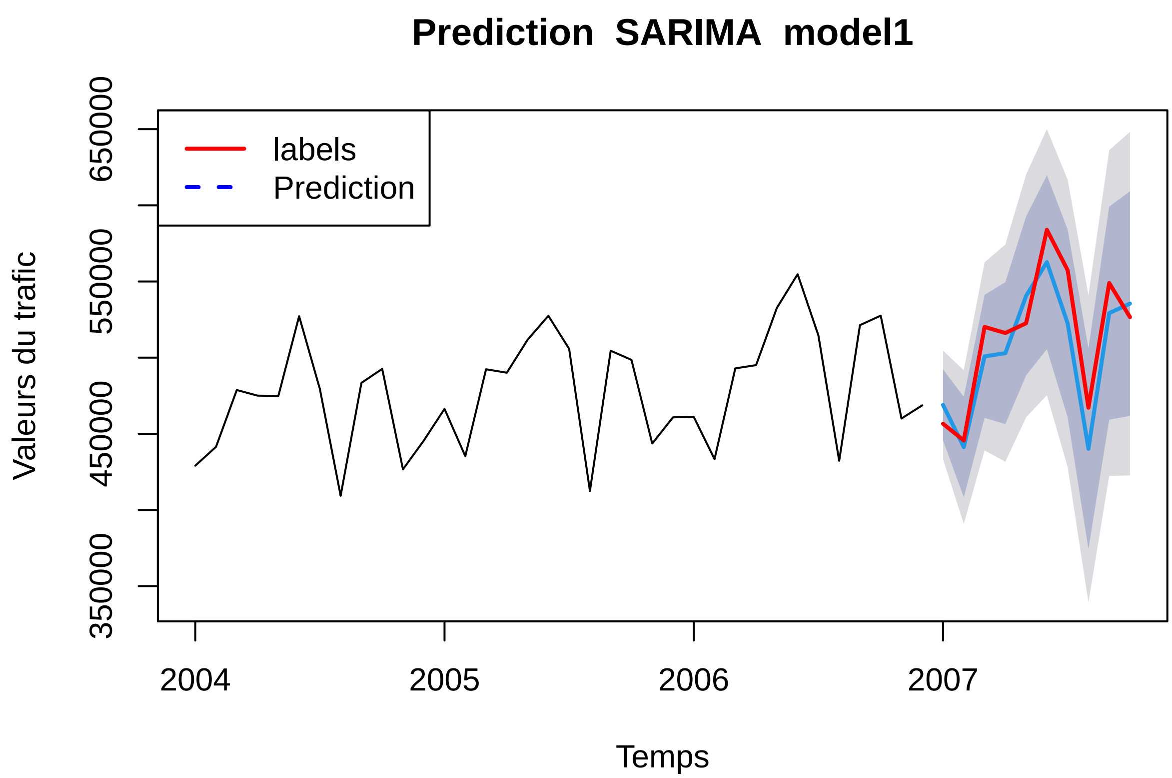
```
## Series: train_data_after
## ARIMA(0,0,0)(0,1,0)[12] with drift
##
## Coefficients:
##           drift
##       1149.7500
## s.e.   205.6977
##
## sigma^2 = 152694933:  log likelihood = -259.66
## AIC=523.33   AICc=523.9   BIC=525.68
```

```r
model_1_after = Arima(train_data_after, order = c(0,1,0),
                seasonal=list(order=c(0,1,0),method='ML',
                              period=12,
                              include.drift=TRUE))
summary(model_1_after)
```

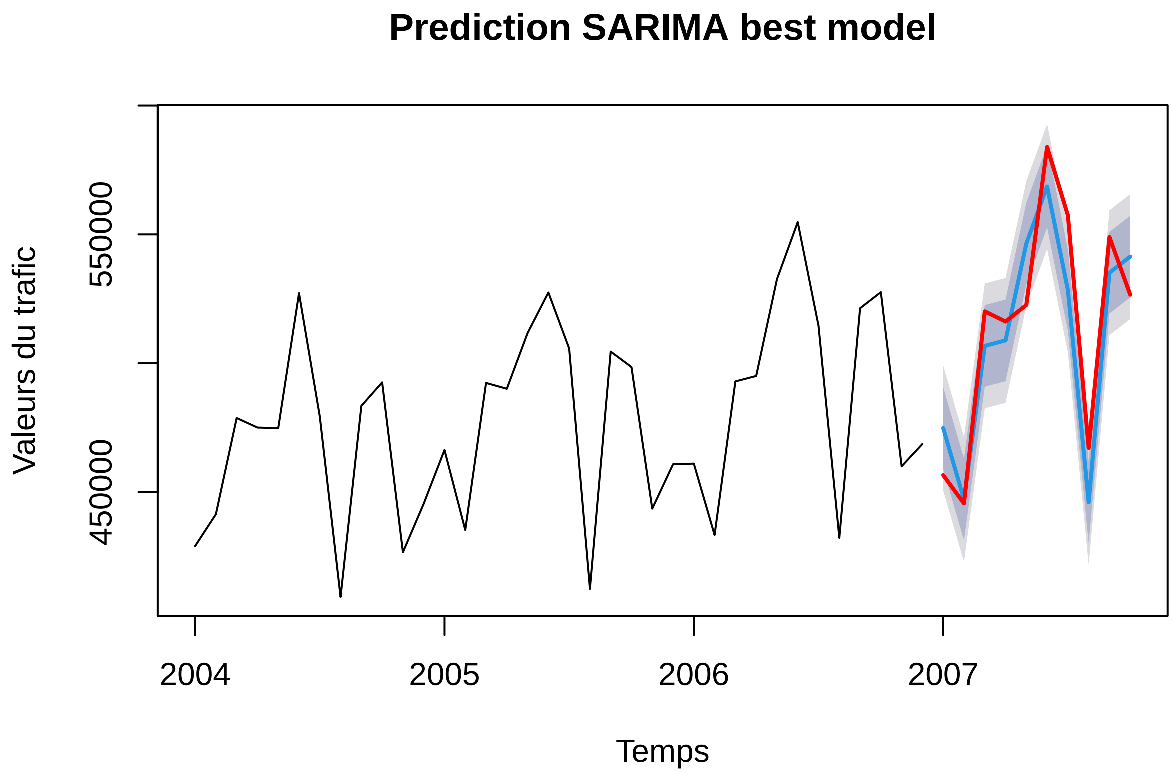
```
## Series: train_data_after
## ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 = 330900847:  log likelihood = -258.23
## AIC=518.46   AICc=518.65   BIC=519.6
```

```
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -842.8088 14539.91 9444.263 -0.233303 1.963777 0.6334818
##                    ACF1
## Training set -0.5637966
```

```r
n_forecast = 10  # number of month to forecast
forecasted_values_model_1_after = forecast(model_1_after, h = n_forecast)

# plot forecast
plot(forecasted_values_model_1_after, main = "Prediction  SARIMA  model1", ylab = "Valeurs du trafic", 

# Add the data since  1993
lines(test_data_after, col = "red", lwd = 2)

# Add legend
legend("topleft", legend = c("labels", "Prediction"),
       col = c("red", "blue"), lwd = 2, lty = c(1, 2))
```

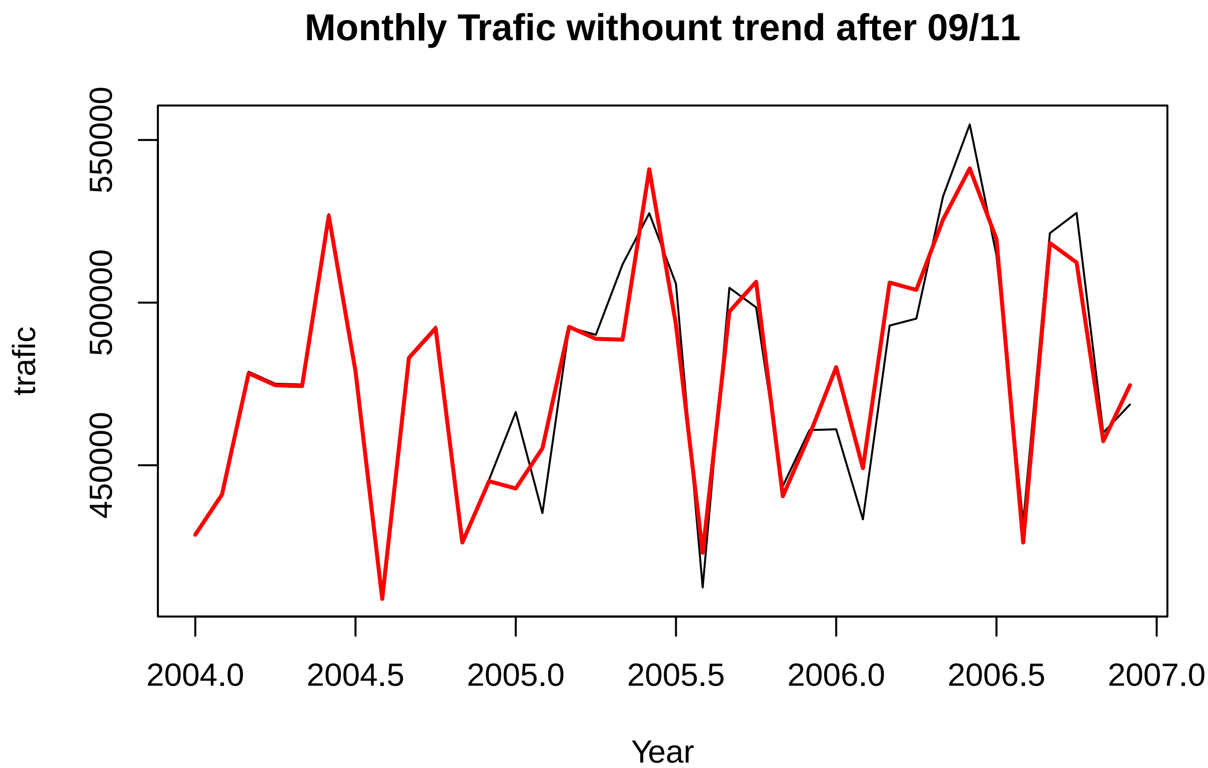



```r
# plot forecast
forecasted_values_best_model_after = forecast(best_model_after, h = n_forecast)

plot(forecasted_values_best_model_after, main = "Prediction SARIMA best model", ylab = "Valeurs du trafi
lines(test_data_after, col = "red", lwd = 2)
```




Even though the AIC of model1 (manual model) is smaller than the AIC of the auto ARIMA model, its confidence interval is much wider. This means model1 is less certain about its forecasts. Therefore, we will prefer the auto ARIMA model.

```
vajust = fitted(best_model_after)
plot(train_data_after, xlab = "Year", ylab = "trafic",
     main = "Monthly Trafic withount trend after 09/11")

# fitted model in red
lines(vajust, col = "red", lwd = 2)  # Tracer les valeurs ajustées
```




```r
# Calcul of MAPE, MAE, et RMSE r
metrics_SARIMA_after = c(
  MAPE = mape(test_data_after, forecasted_values_best_model_after$mean),
  MAE  = mae(test_data_after, forecasted_values_best_model_after$mean),
  RMSE = rmse(test_data_after, forecasted_values_best_model_after$mean)
)

metrics_SARIMA_after
```

```
##         MAPE          MAE         RMSE
## 3.057353e-02 1.584120e+04 1.751597e+04
```

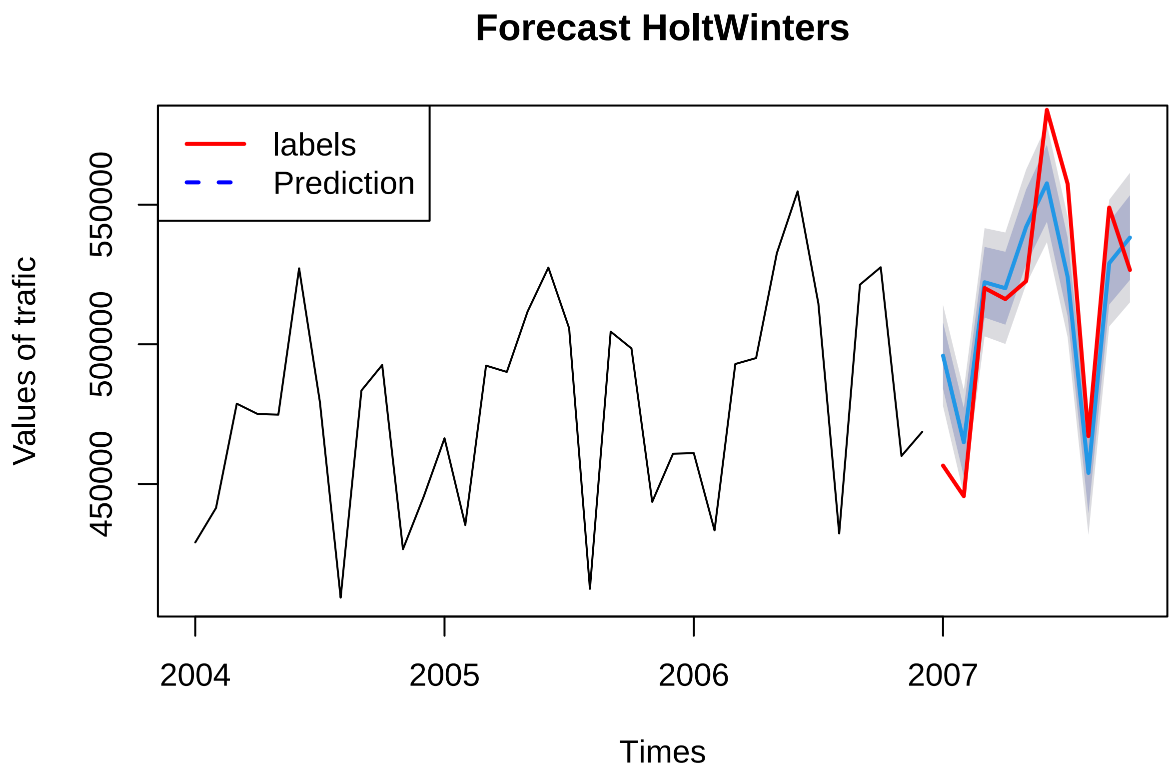
### Holt Winters method

```r
hw_model = HoltWinters(train_data_after)
hw_model
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = train_data_after)
##
## Smoothing parameters:
```

```
##  alpha: 0.2657331
##  beta : 0
##  gamma: 0
##
## Coefficients:
##           [,1]
## a   501033.604
## b     1244.170
## s1   -6313.399
## s2  -38606.024
## s3   17457.809
## s4   14071.642
## s5   34760.684
## s6   49137.351
## s7   14347.351
## s8  -57028.816
## s9   16806.601
## s10  24746.434
## s11 -43345.316
## s12 -26034.316
```

```r
n_forecast = 10  # number of month to forecast
forecasted_values_hw_model = forecast(hw_model, h = n_forecast)

# plot forecast
plot(forecasted_values_hw_model, main = "Forecast HoltWinters", ylab = "Values of trafic", xlab = "Time

# Add the data since  1993
lines(test_data_after, col = "red", lwd = 2)

# Add legend
legend("topleft", legend = c("labels", "Prediction"),
       col = c("red", "blue"), lwd = 2, lty = c(1, 2))
```




```r
# Calcul of MAPE, MAE, et RMSE r
metrics_holtwinters <- c(
  MAPE = mape(test_data_after, forecasted_values_hw_model$mean),
  MAE  = mae(test_data_after, forecasted_values_hw_model$mean),
  RMSE = rmse(test_data_after, forecasted_values_hw_model$mean)
)

metrics_holtwinters
```

```
##         MAPE          MAE         RMSE
## 3.695592e-02 1.883690e+04 2.196323e+04
```

comparing the MAPE, MAE, RMSE of the two models we can see that the SARIMA model is better at forecasting than the HoltWinters models