Cleaning Data

Danielle Lamb

12/15/2021

Project One: Ocean Microplastic Trend Prediction

```
library(DataExplorer)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(rsample)
library(knitr)
library(readxl)
library(forecast)
```

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

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(fUnitRoots)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
##
## Attaching package: 'timeSeries'
```

```
## The following object is masked from 'package:zoo':
##
##     time<-
```

```
## Loading required package: fBasics
```

```
library(tseries)
```

Reading Data

```
mic <- read.csv("C:\\Users\\datre\\OneDrive\\Documents\\Graduate School\\Winter '21\\Project 1\\micropla
```

```
##     OBJECTID    Sample_Date           Lat_deg_       Long_deg_
##  Min.   :   1   Length:5772        Min.   : 7.99   Min.   :-86.72
##  1st Qu.:3074   Class :character   1st Qu.:18.34   1st Qu.:-74.83
##  Median :4684   Mode  :character   Median :25.73   Median :-69.13
##  Mean   :4576                      Mean   :26.96   Mean   :-69.38
##  3rd Qu.:6216                      3rd Qu.:34.84   3rd Qu.:-63.76
##  Max.   :7695                      Max.   :45.09   Max.   :-55.02
```

```
##    Pieces_KM2       Normalized
##  Min.   :     0   Min.   :0
##  1st Qu.:     0   1st Qu.:0
##  Median :  1080   Median :0
##  Mean   :  6918   Mean   :0
##  3rd Qu.:  4970   3rd Qu.:0
##  Max.   :577214   Max.   :0
```

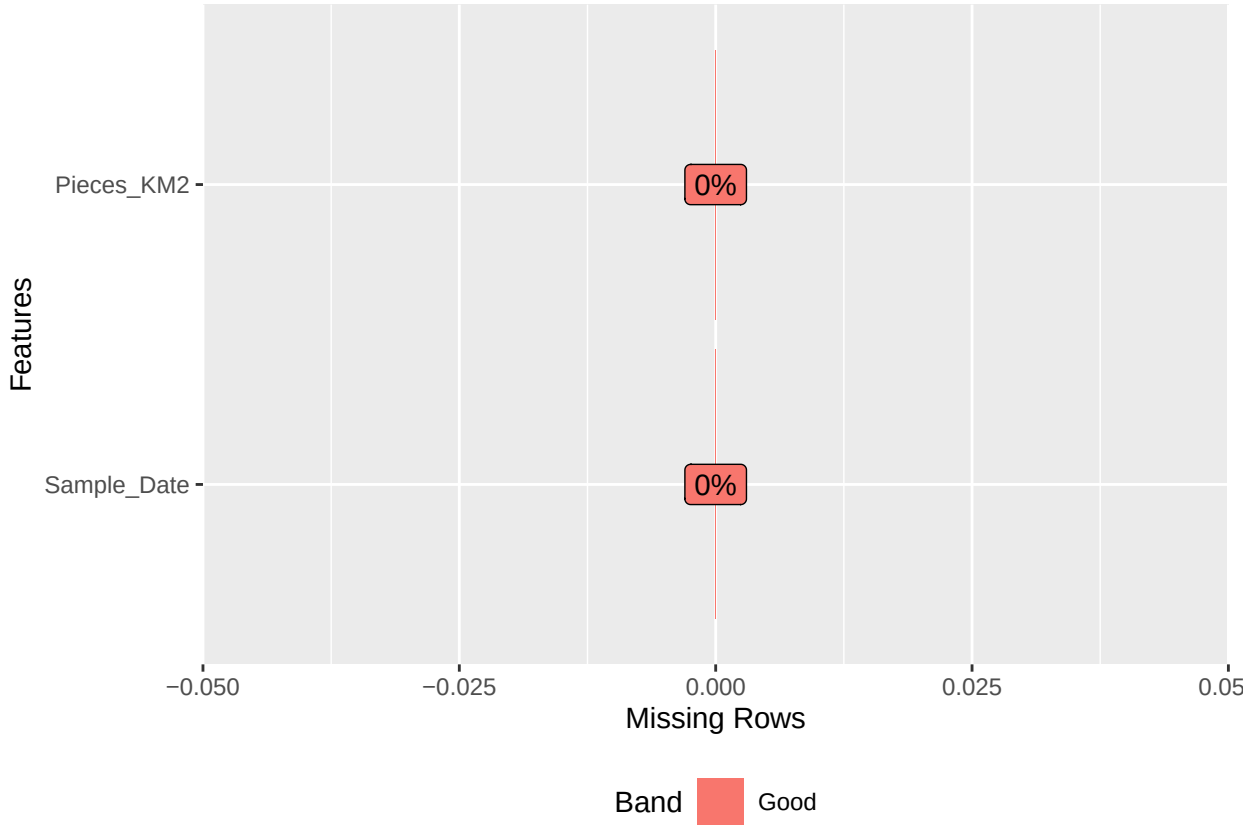
```
plastic$Sample_Date <- as.Date(plastic$Sample_Date, format = "%m/%d/%Y")
head(plastic)
```

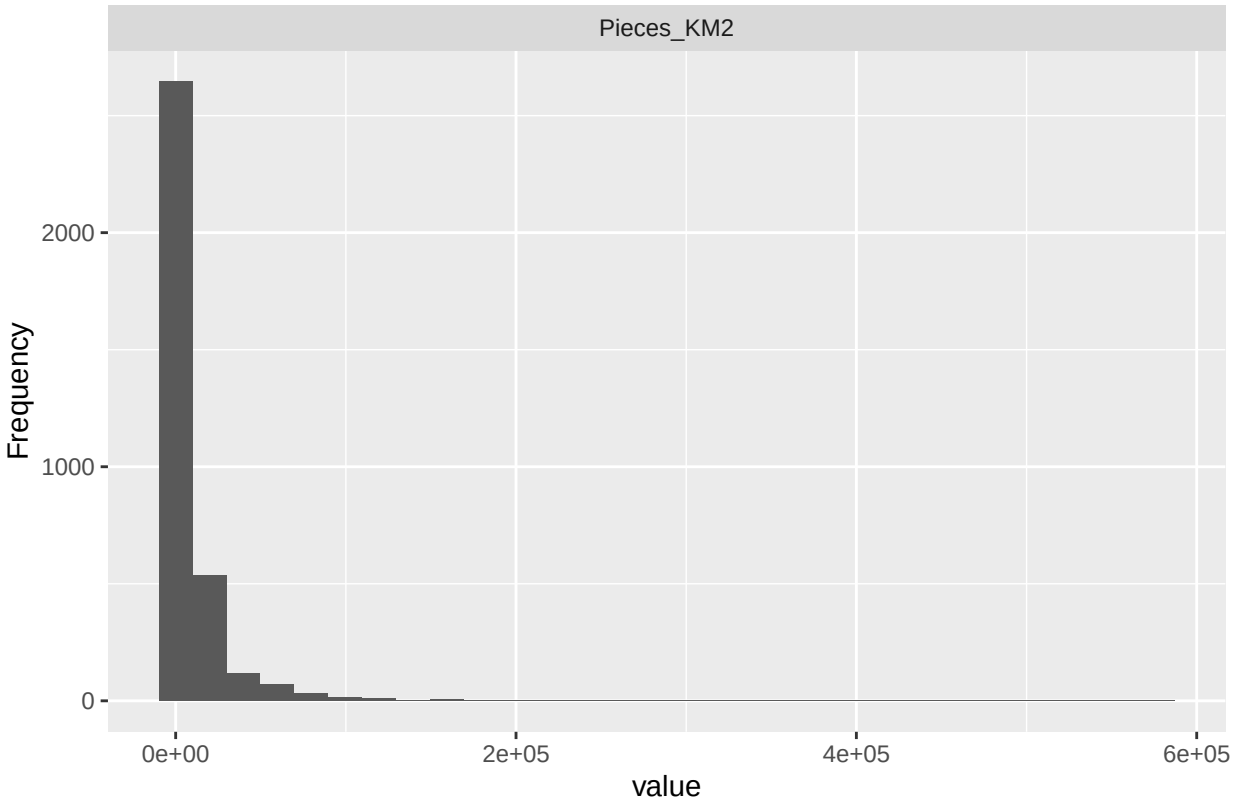
```
##   OBJECTID Sample_Date Lat_deg_ Long_deg_ Pieces_KM2
## 1        1  2004-11-08    13.88    -61.71          0
## 2        2  2004-11-03    14.54    -60.61          0
## 3        3  1997-11-10    16.07    -61.95          0
## 4        4  1996-12-24    18.10    -78.53          0
## 5        5  2004-10-23    28.63    -58.22          0
## 6        6  2004-10-17    39.71    -67.97          0
```

```
pl1 <- subset(plastic, select = -c(OBJECTID, Lat_deg_, Long_deg_))
head(pl1)
```

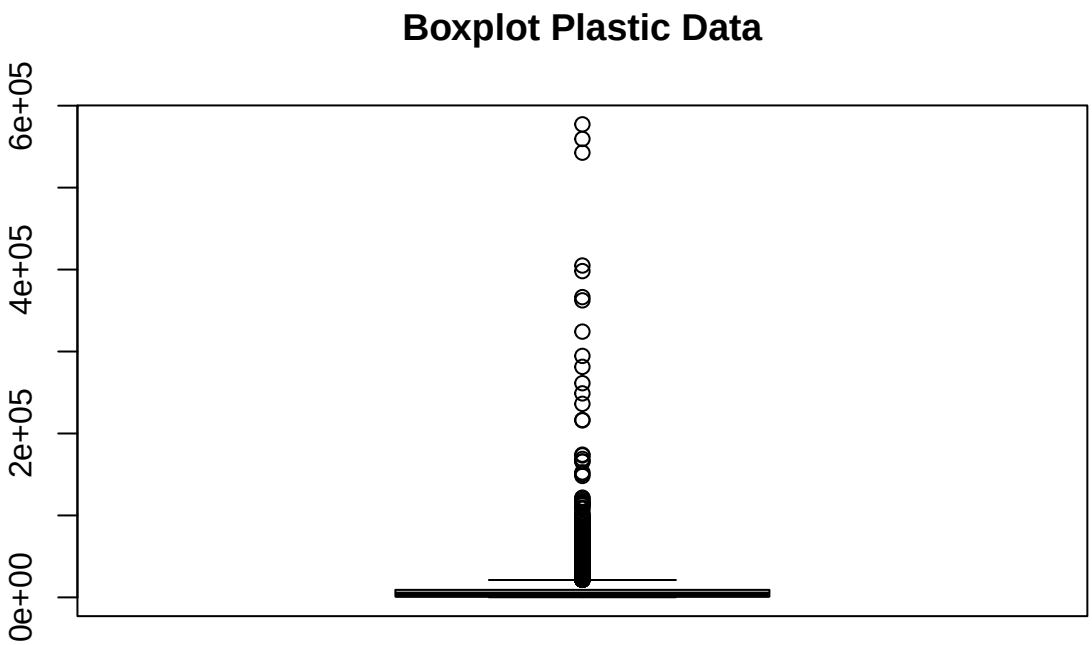
```
##   Sample_Date Pieces_KM2
## 1  2004-11-08          0
## 2  2004-11-03          0
## 3  1997-11-10          0
## 4  1996-12-24          0
## 5  2004-10-23          0
## 6  2004-10-17          0
```

```
pl2 <- subset(pl1, Pieces_KM2 > 0)
pl2 <- pl2[pl2$Sample_Date >= "1990-12-31",]
```


Pieces_KM2
Sample_Date
0%
0%
Features
−0.050
−0.025
0.000
0.025
0.05
Missing Rows
Band
Good




```
boxplot(pl2$Pieces_KM2, main = "Boxplot Plastic Data")
```



Removing Outliers

```
out <- boxplot.stats(pl2$Pieces_KM2)$out
out_ind <- which(pl2$Pieces_KM2 %in% c(out))

pl_clean <- pl2[-out_ind,]
pl_clean = pl_clean[order(pl_clean$Sample_Date),]
head(pl_clean)
```

```
##      Sample_Date Pieces_KM2
## 1656  1991-01-14       6479
## 1453  1991-02-15       2160
## 284   1991-02-16        540
## 279   1991-02-17        540
## 758   1991-02-18        540
## 655   1991-02-19        540
```

```
pl_month <- subset(pl_clean)
pl_month$month <- as.numeric(format(pl_month$Sample_Date, "%m"))
pl_month$year <- as.numeric(format(pl_month$Sample_Date, "%Y"))
head(pl_month)
```

```
##      Sample_Date Pieces_KM2 month year
```

```
## 1656  1991-01-14       6479     1 1991
## 1453  1991-02-15       2160     2 1991
## 284   1991-02-16        540     2 1991
## 279   1991-02-17        540     2 1991
## 758   1991-02-18        540     2 1991
## 655   1991-02-19        540     2 1991
```

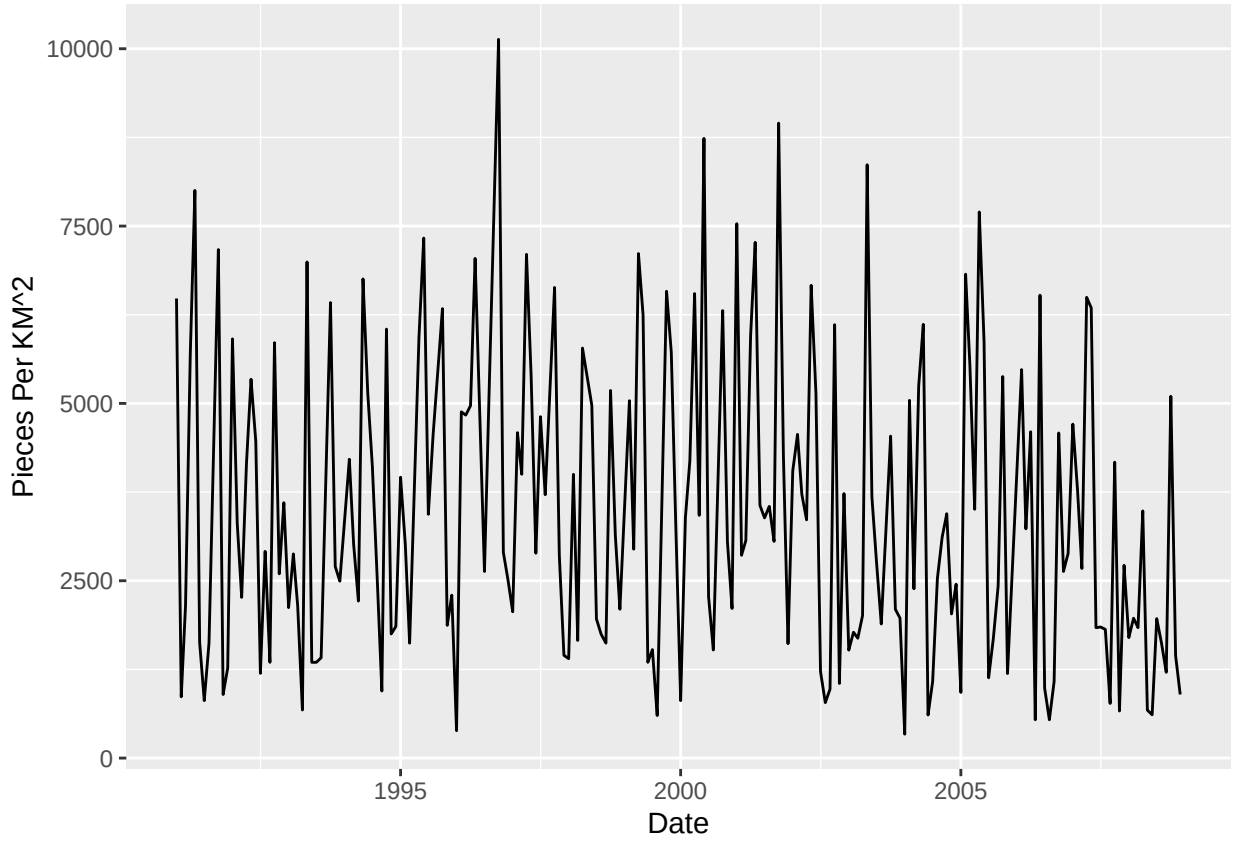
Montly Averages

```
month_avg <- aggregate(Pieces_KM2 ~ month + year, pl_month, FUN = mean)
month_avg$Date <- paste(month_avg$year, month_avg$month, sep = "-")
month_avg$Date <- as.Date(as.yearmon(month_avg$Date))
month_avg <- subset(month_avg, select = -c(month, year))
head(month_avg)
```

```
##   Pieces_KM2       Date
## 1   6479.000 1991-01-01
## 2    864.000 1991-02-01
## 3   2152.100 1991-03-01
## 4   5690.875 1991-04-01
## 5   8003.185 1991-05-01
## 6   1632.714 1991-06-01
```

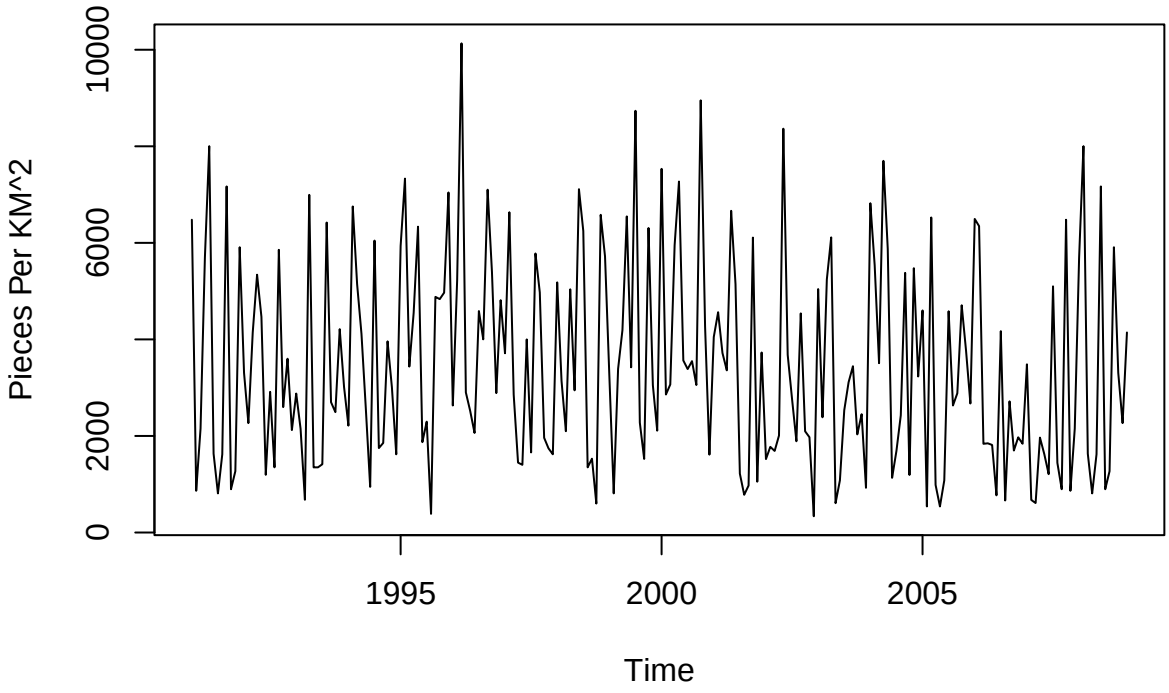
Exploring Data

```
ggplot(month_avg, aes(Date, Pieces_KM2)) + geom_line() + scale_x_date("Date")  + ylab("Pieces Per KM^2")
           xlab("")
```

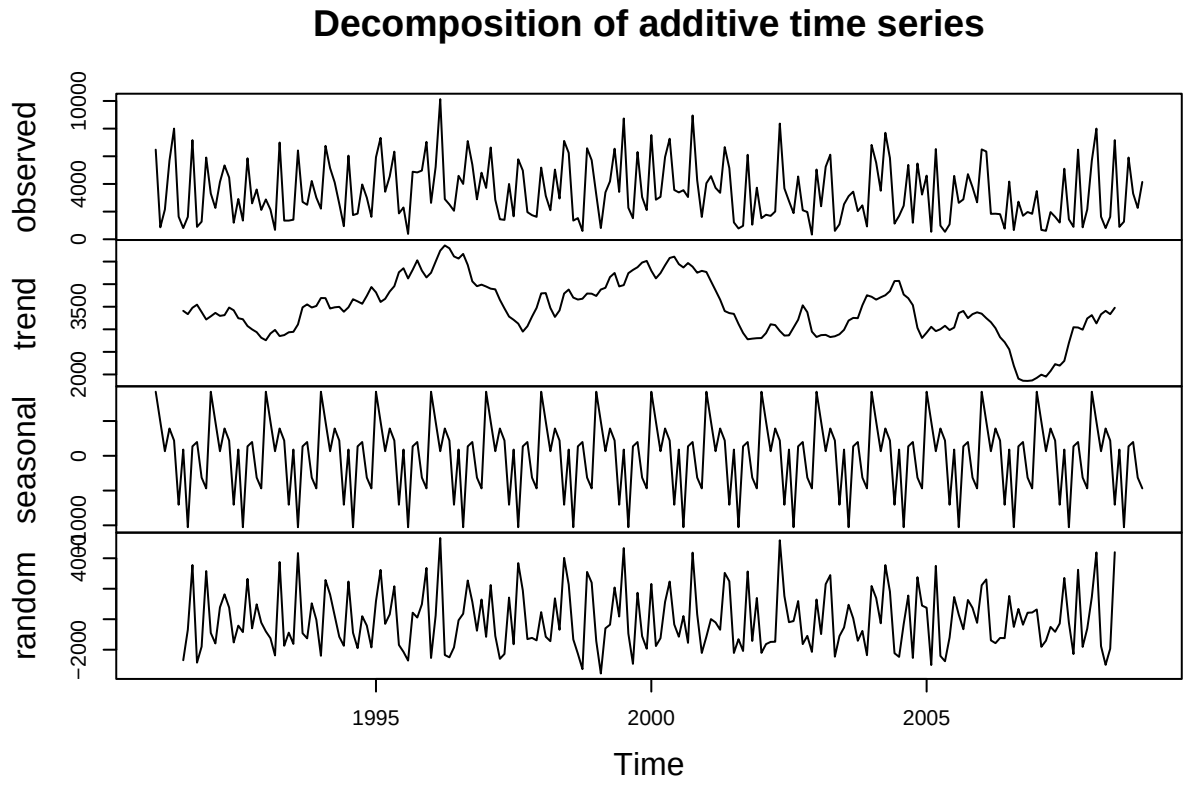


Creating Time Series

```
pl_ts = ts(month_avg$Pieces_KM2, start = c(1991, 1), end = c(2008, 12), frequency = 12)
plot.ts(pl_ts, ylab = "Pieces Per KM^2")
```




```
decomp <- decompose(pl_ts)
plot(decomp)
```



Checking Stationarity

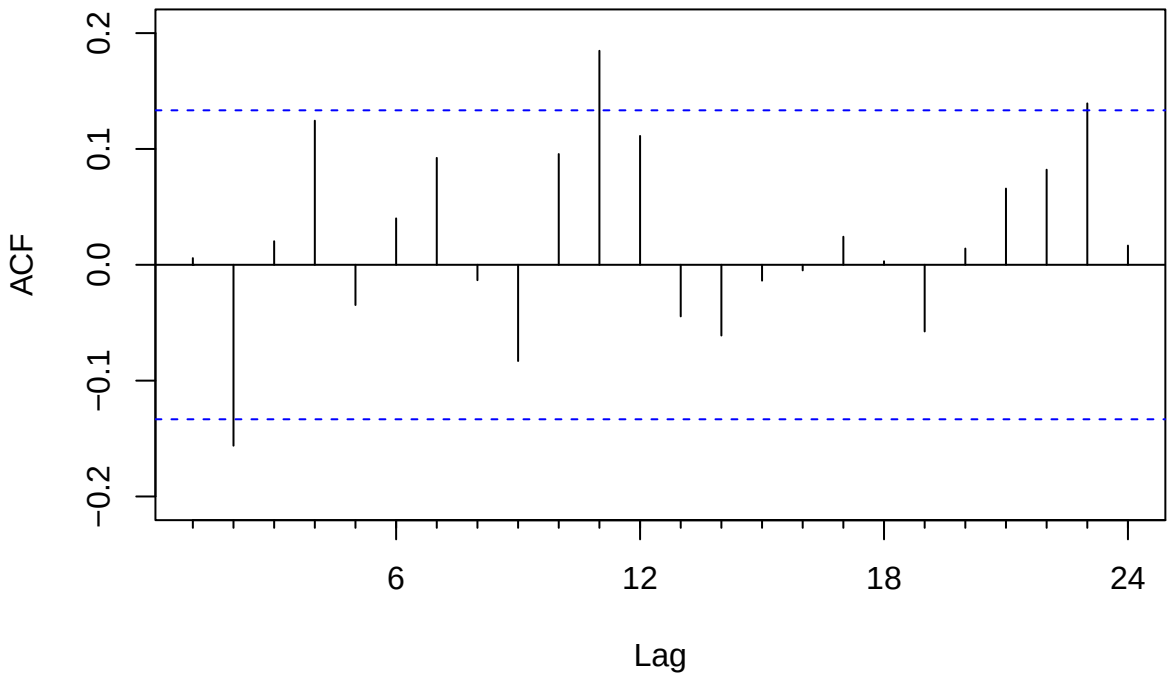
```
adf.test(pl_ts, alternative = "stationary")
```

```
## Warning in adf.test(pl_ts, alternative = "stationary"): p-value smaller than
## printed p-value
```

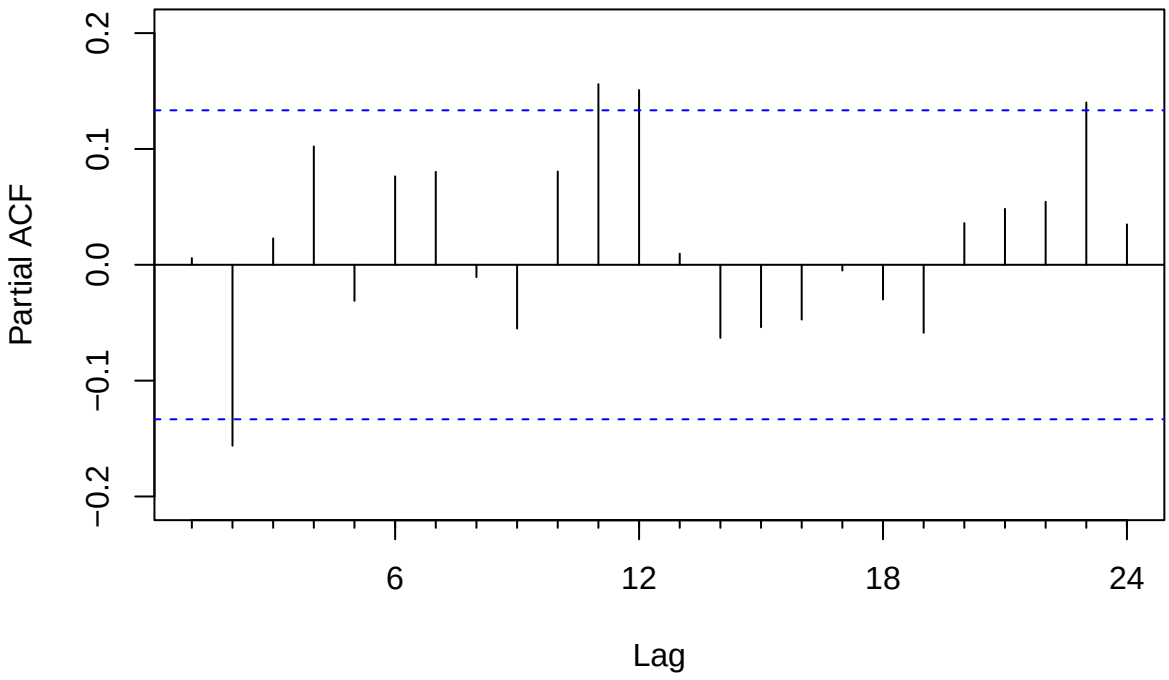
```
##
##  Augmented Dickey-Fuller Test
##
## data:  pl_ts
## Dickey-Fuller = -5.5905, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Evaluating Hyperparameters

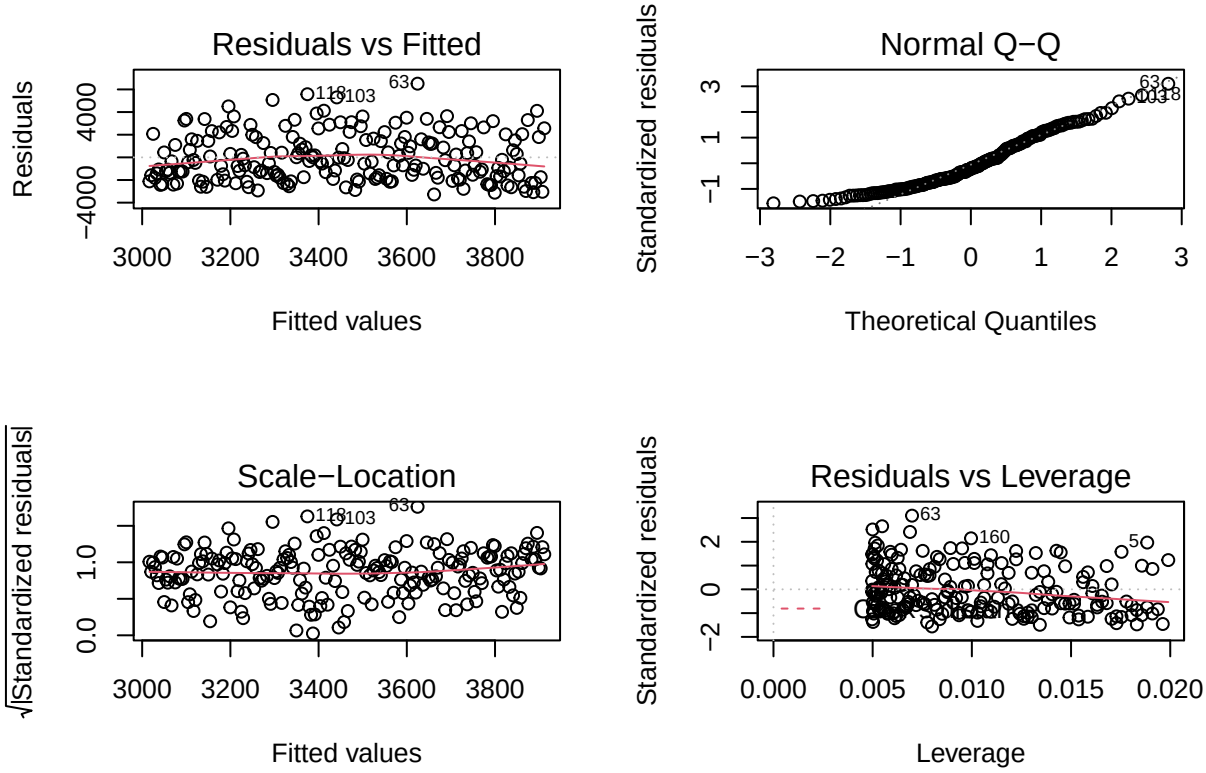
```
Acf(pl_ts, main='')
```




```
Pacf(pl_ts, main='')
```




```
lmMod <- lm(Pieces_KM2 ~ Date, data = month_avg)
par(mfrow = c(2,2))
plot(lmMod)
```

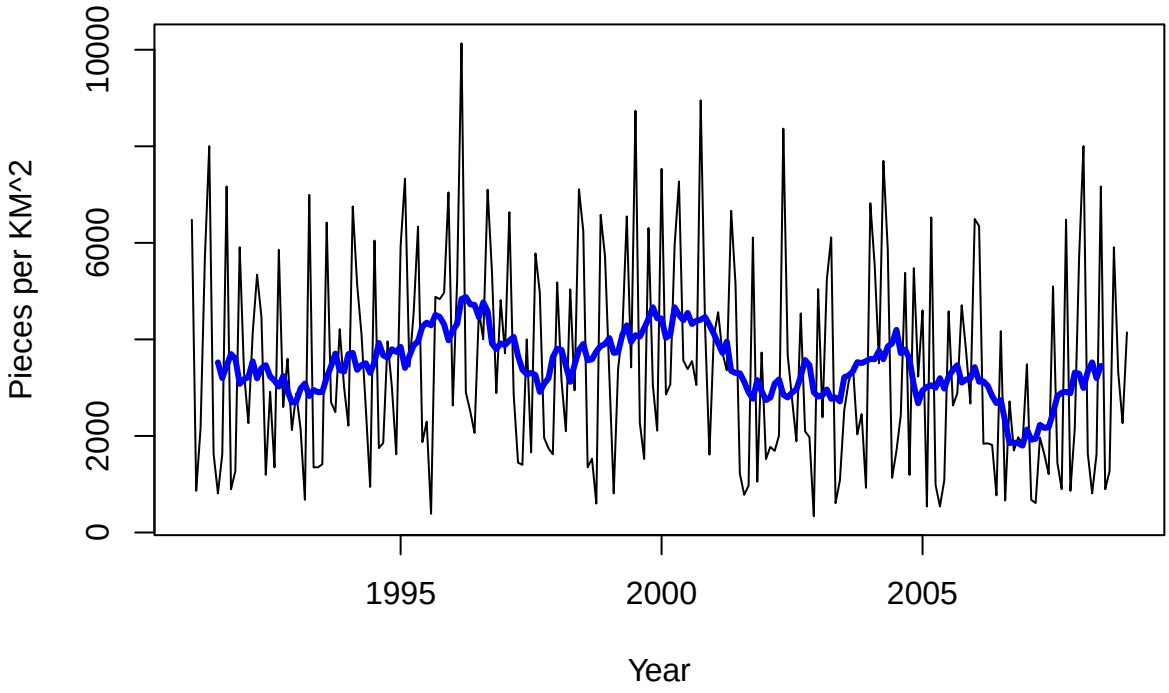



```
lmtest::bptest(lmMod)
```

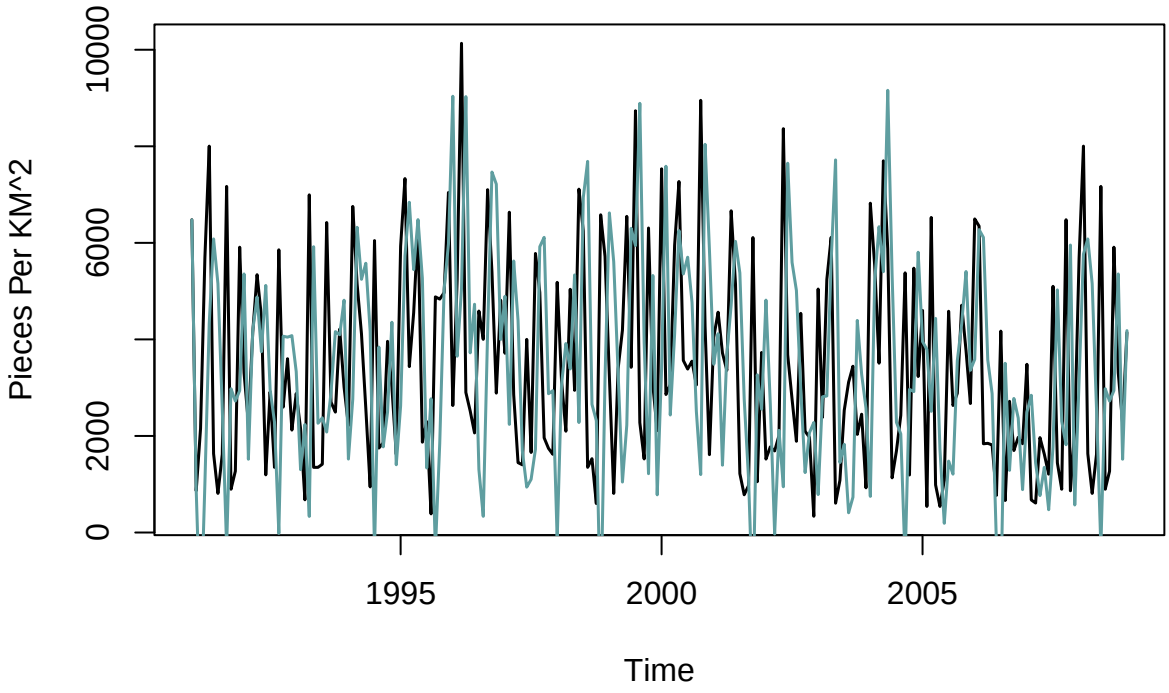
```
##
##  studentized Breusch-Pagan test
##
## data:  lmMod
## BP = 0.7419, df = 1, p-value = 0.3891
```

Moving Average Modeling

```
MA1 = forecast::ma(pl_ts, order = 13, centre = TRUE)
plot(pl_ts, xlab = "Year", ylab = "Pieces per KM^2")
lines(MA1, col = "blue", lwd = 3)
```




```
MA2 = forecast::Arima(pl_ts, c(3,2,0))
plot(pl_ts, ylab = "Pieces Per KM^2", lwd = 1.5)
lines(fitted(MA2), col = "cadetblue", lwd = 1.5)
```



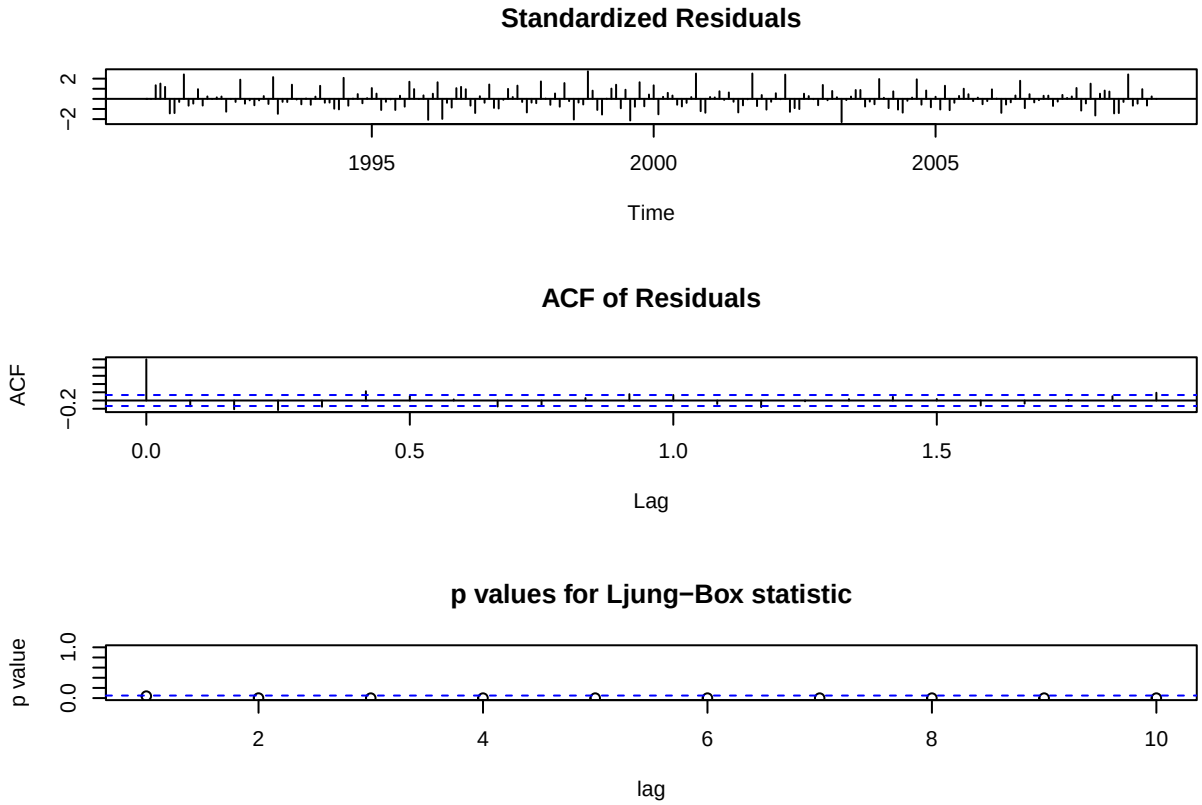

```
summary(MA2)
```

```
## Series: pl_ts
## ARIMA(3,2,0)
##
## Coefficients:
##           ar1      ar2      ar3
##       -1.1360  -0.9509  -0.5052
## s.e.   0.0597   0.0736   0.0595
##
## sigma^2 estimated as 9594482:  log likelihood=-2023.3
## AIC=4054.59   AICc=4054.78   BIC=4068.06
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
## Training set 38.60654 3061.435 2402.027 -58.92221 108.7294 1.068034 -0.1393152
```

```
confint(MA2)
```

```
##          2.5 %     97.5 %
## ar1 -1.2530950 -1.0189900
## ar2 -1.0951531 -0.8066251
## ar3 -0.6219339 -0.3885276
```

```
tsdiag(MA2)
```




```
Box.test(pl_ts, lag = 1, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  pl_ts
## X-squared = 0.0072526, df = 1, p-value = 0.9321
```

```
Box.test(pl_ts, lag = 12, type = "Ljung")
```

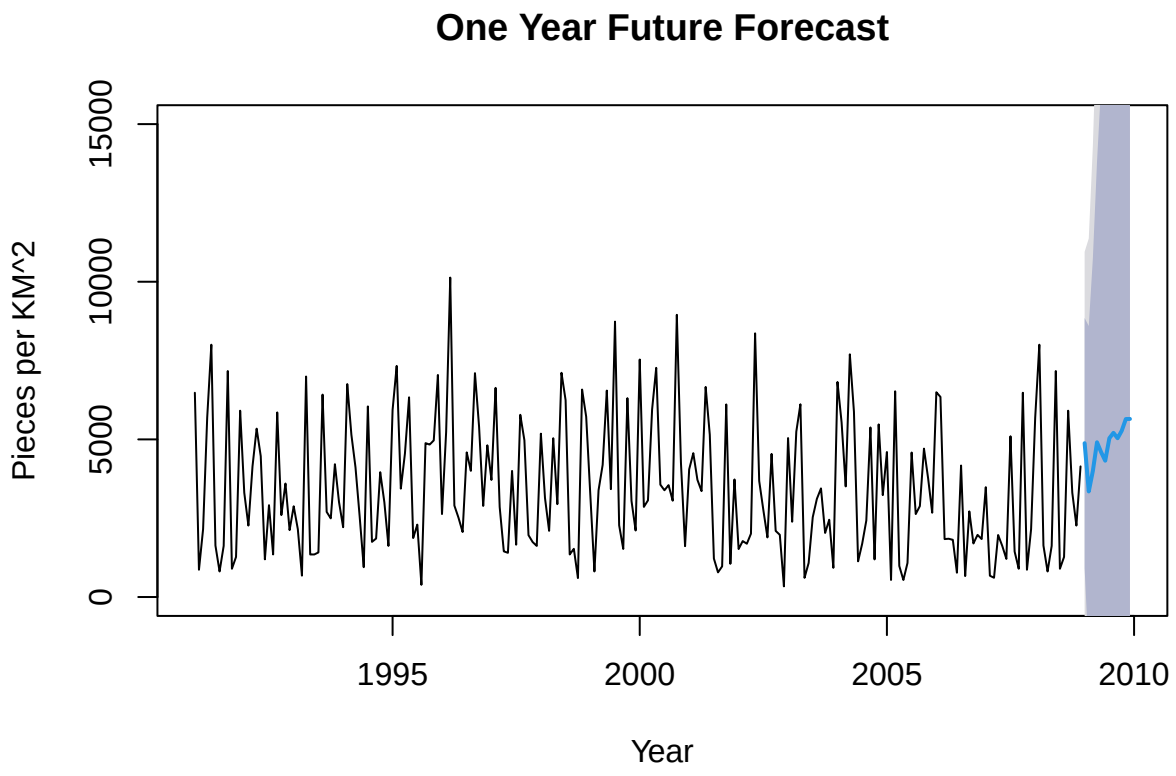
```
##
##  Box-Ljung test
##
## data:  pl_ts
## X-squared = 25.842, df = 12, p-value = 0.0113
```

```
fit <- forecast::Arima(pl_ts, c(3,2,0))
forecast_val <- forecast(fit, 12)
head(forecast_val)
```

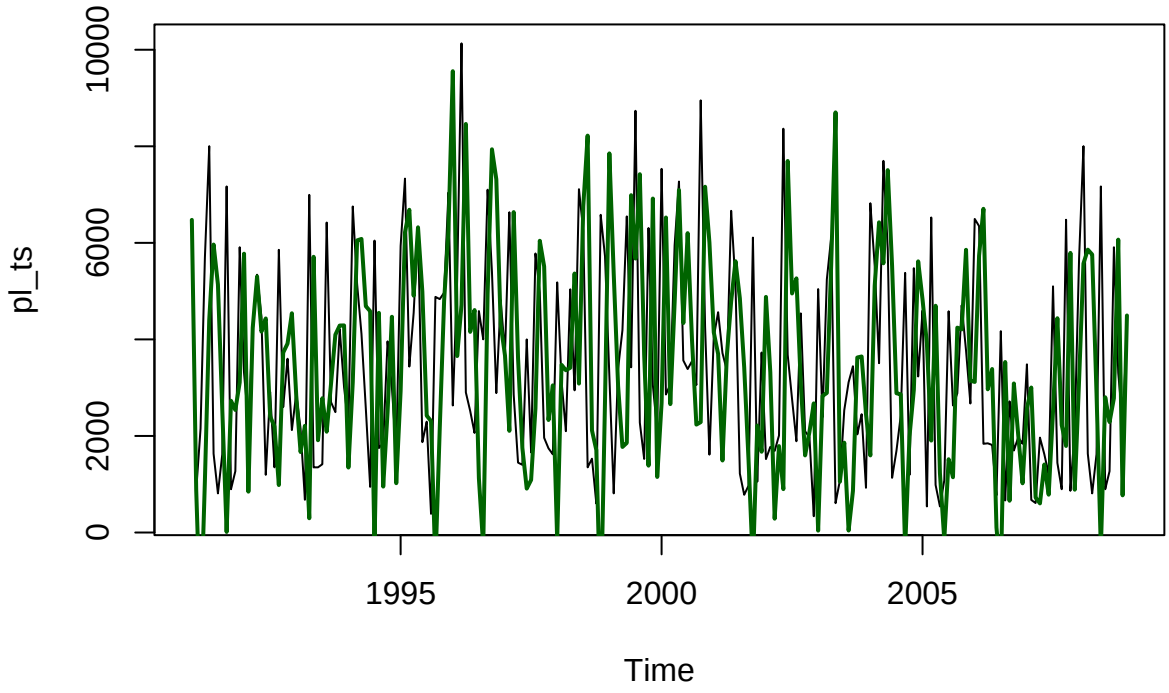
```
## $method
```

```
## [1] "ARIMA(3,2,0)"
##
## $model
## Series: pl_ts
## ARIMA(3,2,0)
##
## Coefficients:
##           ar1      ar2      ar3
##       -1.1360  -0.9509  -0.5052
## s.e.   0.0597   0.0736   0.0595
##
## sigma^2 estimated as 9594482:  log likelihood=-2023.3
## AIC=4054.59   AICc=4054.78   BIC=4068.06
##
## $level
## [1] 80 95
##
## $mean
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2009 4879.848 3348.491 4003.563 4907.880 4595.505 4323.689 5036.809 5207.143
##           Sep      Oct      Nov      Dec
## 2009 5037.048 5272.204 5644.920 5648.008
##
## $lower
##                   80%        95%
## Jan 2009    910.2468  -1191.133
## Feb 2009  -1897.4268  -4674.448
## Mar 2009  -2740.3575  -6310.373
## Apr 2009  -3985.2184  -8692.940
## May 2009  -7037.6938 -13195.937
## Jun 2009  -9596.2240 -16964.981
## Jul 2009 -11562.5674 -20349.746
## Aug 2009 -14394.8433 -24771.508
## Sep 2009 -17649.5881 -29659.168
## Oct 2009 -20504.9201 -34150.505
## Nov 2009 -23516.0150 -38952.881
## Dec 2009 -27025.7296 -44322.162
##
## $upper
##                 80%      95%
## Jan 2009  8849.449 10950.83
## Feb 2009  8594.409 11371.43
## Mar 2009 10747.484 14317.50
## Apr 2009 13800.979 18508.70
## May 2009 16228.704 22386.95
## Jun 2009 18243.601 25612.36
## Jul 2009 21636.185 30423.36
## Aug 2009 24809.128 35185.79
## Sep 2009 27723.684 39733.26
## Oct 2009 31049.327 44694.91
## Nov 2009 34805.855 50242.72
## Dec 2009 38321.746 55618.18
```

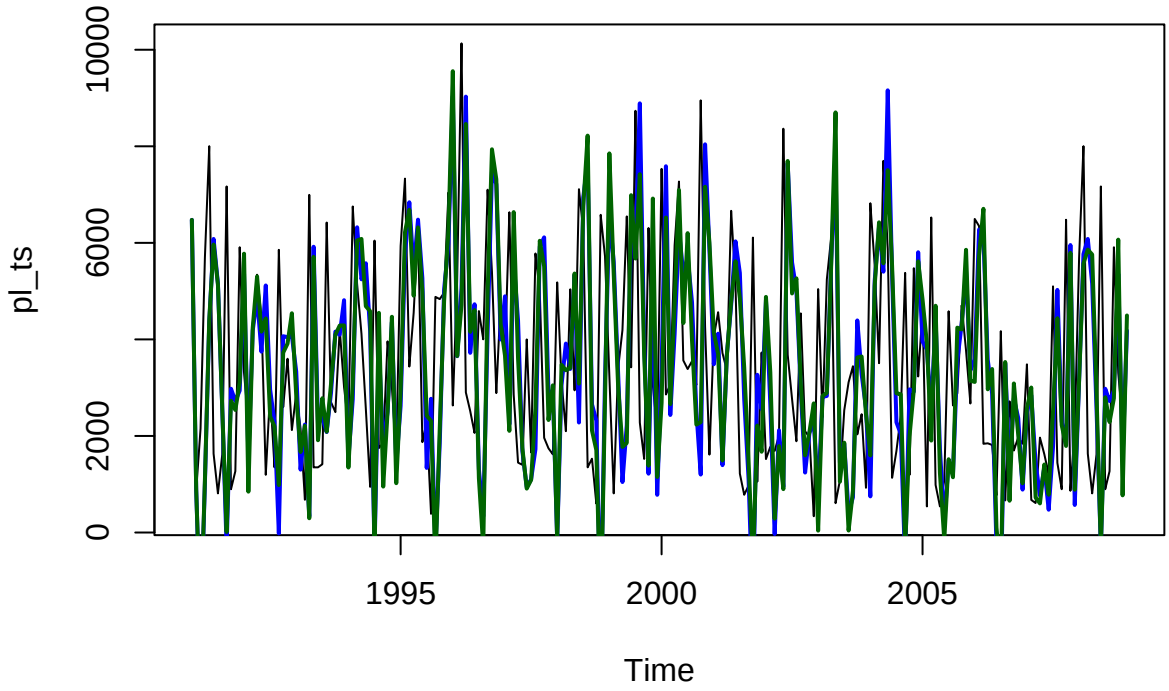
```
plot(forecast_val, main = "One Year Future Forecast", ylim = c(0, 15000), ylab = "Pieces per KM^2", xlab
```




```
season_MA2 = forecast::Arima(pl_ts, c(3,2,0), seasonal = c(3,0,0))
plot(pl_ts)
lines(fitted(season_MA2), col = "dark green", lwd = 2)
```




```
plot(pl_ts)
lines(fitted(MA2), col = "blue", lwd = 2)
lines(fitted(season_MA2), col = "dark green", lwd = 2)
```


pl_ts
0
2000
6000
10000
1995
2000
2005
Time