# Doing Data Science: Case Study 2 - Forecasting Chulwalar Total Etel Exports

### Claire Chu & Joseph Stoffa

### July 19, 2016

## Introduction

Chulwalar is part of the island group Urbano in the northern hemisphere. They are famous for their plants which flower in winter. There are three main plants that Chulwalar exports: Efak is a leafy bush with white flowers, Wuge is a grass like plant with tiny pink flowers and Etel is a flowering tree. Etel comes in two varieties: red flowers and blue flowers. Due to the nature of the products, exports generally are higher towards the end of the year. Chulwalar celebrates its independence on the 1st of December each year. On this day it is custom to give presents to family and friends. Chulwalar also celebrates the March Equinox as a time of rebirth in the northern hemisphere.

The Prime Minister of Chulwalar has asked [Wheeler et al] to help him in forecasting the exports. In order to do this [Wheeler et al] were given as is data and plan data as well as a list of indicators which may affect exports. The Wheeler group's job is to find out the best way to forecast Chulwalar's exports in 2014 based on data collected before this year

## Background

This case study builds on the data analysis and modeling of plant exports from Chulwalar performed by Wheeler et al (outlined in Introduction) in order to find the best model to forecast total Etel exports.

## Table of Contents

#### 1.0 [Importing Data](#id-section1)

#### 2.0 [Cleaning Modifying Data](#id-section2)

#### 3.0 [Exploring Data](#id-section3)

#### ---3.1 [Preliminary Analysis](#id-section3.1)

#### ---3.2 [Preliminary Time Series Analysis](#id-section3.2)

#### 4.0 [Correlation of Different External Indicators](#id-section4)

#### ---4.1 [4.1 Definition of the Indicators and Their Correlation with the Basic Data](#id-section4.1)

#### ---4.2 [4.2 Correlation of the Indicators with a Time Offset](#id-section4.2)

#### ---4.3 [Correlation of the Indicators with Each Other](#id-section4.3)

#### 5.0 [Development of Forecasting Models](#id-section5)

#### ---5.1 [ModelWithAlllIndicators and with Each Indicator Individually](#id-section5.1)

#### ---5.2 [Models](#id-section5.2)

#### 6.0 [Forecasts with the models](#id-section6)

#### ---6.1 [Shorten the Time Series in Order to Test the Forecasts](#id-section6.1)

#### ---6.2 [Forecasting and Testing the Models](#id-section6.2)

#### 7.0 [Forecast for 2014](#id-section7)

#### 8.0 [Developing Forecasting Models with Alternative Model Approaches](#id-section8)

#### ---8.1 [Exponential Smoothing](#id-section8.1)

#### ---8.2 [ARIMA](#id-section8.2)

#### ---8.3 [Dynamic Regression Models](#id-section8.3)

#### 9.0 [Conclusion](#id-section9)

#### 10.0 [Acknowledgements](#id-section10)

## 1.0 Importing Data

The script *loadData.r* loads the the data for this case study into dataframes *ImportedAsISData*, *ImportedIndicators*, and *ImportedPlanData* from the files *ImportedAsIsDataChulwalar.csv*, *ImportedIndicatorshulwalar.csv*, and *ImportedPlanDataChulwalar.csv* respectively.

## [1] CSV files successfully read. Dataframes loaded:   
## [1] ImportedAsIsData ImportedIndicators ImportedPlanData

## 2.0 Cleaning/Modifying Data

In order to conduct our analysis the data needs to be subsetted into vectors and time series. For this case study we are only interested in the Total Etel exports. Data on other individual plants will be ignored. The script *cleanData.r* takes the data frames for As Is, Planned, and Indicator data and splits them into vectors which are then converted into time series. Below are the resultant time series outputted by the script.

source("Scripts/cleanData.r")

## [1] TotalAsIs:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## 2013 4119526 3535744 3560974 3760065 2959933 2787898 2828744 3084113  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673  
## 2013 5107775 4562144 4729313 4372181  
## [1]   
## [1] TotalEtelAsIs:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 1279668 1053325 1367520 1090725 873568 644479 772658 806741  
## 2009 1583216 1407388 1420801 1141100 919860 858876 910134 843050  
## 2010 1637464 1676161 1549560 813469 1198401 1140024 551268 1012542  
## 2011 1595267 1473528 1469728 1034650 952553 819303 802076 1222812  
## 2012 1519748 1812897 1607280 1008022 1291983 940158 945929 1235146  
## 2013 2109497 1738197 1633944 1745092 1039449 1054201 1003166 1154675  
## Sep Oct Nov Dec  
## 2008 1715265 1795751 1518288 1601324  
## 2009 1981563 1647934 1857836 1615091  
## 2010 2335488 1856264 1678123 1699063  
## 2011 2303271 1591584 1960675 1713991  
## 2012 2330334 2177895 2306324 1618147  
## 2013 3000929 2305605 2284672 2062160  
## [1]   
## [1] YearAsIs:   
## Jan Feb Mar Apr May Jun Jul  
## 2008 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## 2009 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## 2010 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## 2011 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## 2012 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## 2013 26280011 29609916 32726772 37215503 40629676 45408410 26280011  
## Aug Sep Oct Nov Dec  
## 2008 29609916 32726772 37215503 40629676 45408410  
## 2009 29609916 32726772 37215503 40629676 45408410  
## 2010 29609916 32726772 37215503 40629676 45408410  
## 2011 29609916 32726772 37215503 40629676 45408410  
## 2012 29609916 32726772 37215503 40629676 45408410  
## 2013 29609916 32726772 37215503 40629676 45408410  
## [1]   
## [1] TotalAsIs\_2014:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 4308161 4155378 3924332 3659121 3898758 3313891 3595106 3502426  
## Sep Oct Nov Dec  
## 2014 5619059 5274287 4841693 4664854  
## [1]   
## [1] TotalPlan:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2243103 2162705 2720911 2011182 1877757 1819924 1682196 1893171  
## 2009 2547980 2247049 2731156 2020158 2098038 1927995 1783692 1907705  
## 2010 2965885 2751170 2906493 2383358 2246893 1992851 2023434 2244997  
## 2011 3113110 2883766 2957893 2601648 2370949 2339881 2105328 2341623  
## 2012 3895396 3588151 3787240 3036434 2907891 2707822 2619486 3784557  
## 2013 3580325 3863212 3606083 3213575 3139128 2998610 2785453 3083654  
## Sep Oct Nov Dec  
## 2008 3325711 2662148 2909966 2574633  
## 2009 3124040 3102251 3154669 2742367  
## 2010 3257717 3536338 3358206 3112906  
## 2011 4086297 3640827 3502334 3280476  
## 2012 4987460 4367319 4205772 4059533  
## 2013 5143757 4149334 4495212 4093664  
## [1]   
## [1] TotalEtelPlan:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 1263613 1231125 1489621 1051346 933392 932047 855520 923070  
## 2009 1546801 1378217 1563799 1166229 1057223 983279 913751 980703  
## 2010 1648769 1490577 1538493 1208636 1104777 931127 916160 1096933  
## 2011 1781991 1564272 1455531 1257528 1134418 1018200 843336 974375  
## 2012 2070256 1731099 1663266 1232994 1164076 1018137 932241 1800576  
## 2013 1864733 1837228 1663834 1305603 1172373 1089115 1074687 1217930  
## Sep Oct Nov Dec  
## 2008 2080877 1575579 1561956 1515127  
## 2009 1974166 1886971 1839155 1727567  
## 2010 1832882 2103588 1877929 1862684  
## 2011 2435674 1972649 1873075 1684766  
## 2012 2823873 2224655 2025003 1955509  
## 2013 2916115 2043888 2199880 2133214  
## [1]   
## [1] YearPlan:   
## Jan Feb Mar Apr May Jun Jul  
## 2008 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## 2009 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## 2010 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## 2011 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## 2012 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## 2013 27883407 29387100 32780247 35224132 43947063 44152007 27883407  
## Aug Sep Oct Nov Dec  
## 2008 29387100 32780247 35224132 43947063 44152007  
## 2009 29387100 32780247 35224132 43947063 44152007  
## 2010 29387100 32780247 35224132 43947063 44152007  
## 2011 29387100 32780247 35224132 43947063 44152007  
## 2012 29387100 32780247 35224132 43947063 44152007  
## 2013 29387100 32780247 35224132 43947063 44152007  
## [1]   
## [1] TotalPlan\_2014:   
## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 4474000 4185565 4278119 3985542 3605973 3515173 3269444 3656112  
## Sep Oct Nov Dec  
## 2014 5637391 5157781 5353458 4703185

## 3.0 Exploring Data

### 3.1 Preliminary Analysis

For the preliminary Analysis, the time series have been plotted and the correlation between the As Is and Plan data has been tested in order to ascertain how exact the planning is. These results show a very high planning accuracy and are therefore suitable for modeling. (Wheeler et al.)

source("Scripts/preliminaryAnalysis.r")

## [1] Correlation between Total As Is and Total Plan:   
## [1] 0.9183402  
## [1] Correlation between Total Etel As Is and Total Etel Plan:   
## [1] 0.9159505  
## [1] Correlation between Year As Is and Year Plan:   
## [1] 0.9627401  
##   
## Call:  
## lm(formula = TotalAsIs ~ TotalPlan, data = TotalAsIs)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -770214 -196776 26017 182579 672705   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.959e+04 1.521e+05 0.589 0.558   
## TotalPlan 9.627e-01 4.959e-02 19.413 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 332600 on 70 degrees of freedom  
## Multiple R-squared: 0.8433, Adjusted R-squared: 0.8411   
## F-statistic: 376.9 on 1 and 70 DF, p-value: < 2.2e-16  
##   
##   
## Call:  
## tslm(formula = TotalAsIs ~ TotalPlan)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -770214 -196776 26017 182579 672705   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.959e+04 1.521e+05 0.589 0.558   
## TotalPlan 9.627e-01 4.959e-02 19.413 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 332600 on 70 degrees of freedom  
## Multiple R-squared: 0.8433, Adjusted R-squared: 0.8411   
## F-statistic: 376.9 on 1 and 70 DF, p-value: < 2.2e-16

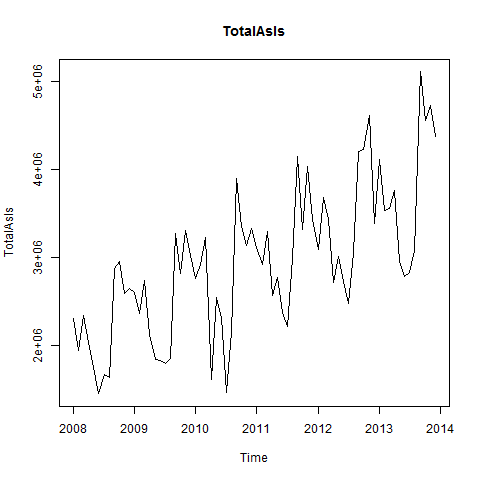


Figure 1

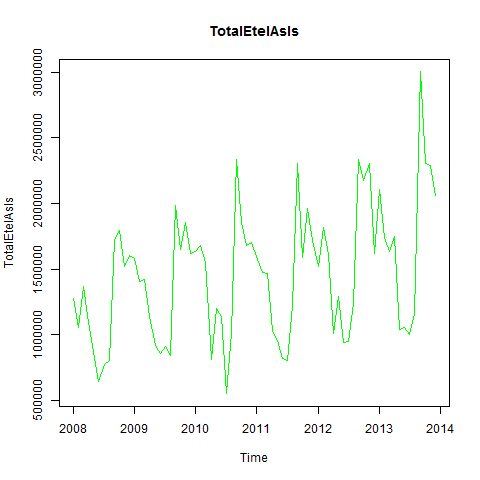


Figure 2

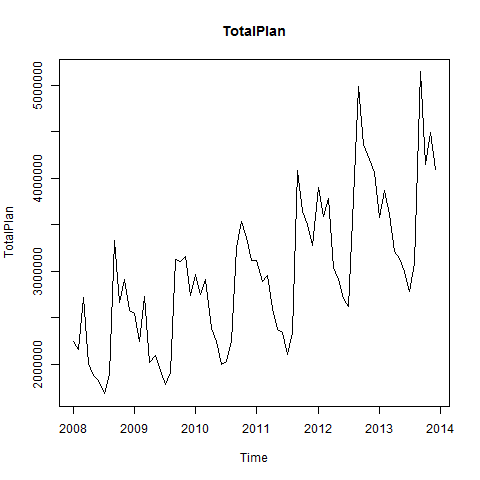


Figure 3

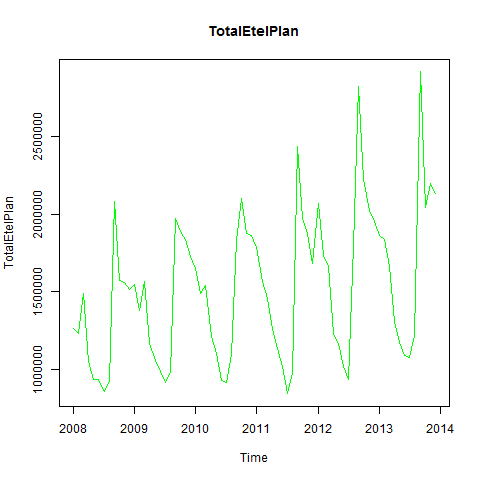


Figure 4

### 3.2 Preliminary Time Series Analysis

The time series can be analyzed using the *stl* function in order to seperate the trend, seasonality, and remainder (remaining) coincidental) components from one another. Thus the individual time series can be shown graphically and tabularly. The trend of the total exports is almost linear. A relatively uniform seasonality can be seen in Figure 5. (Wheeler et al).

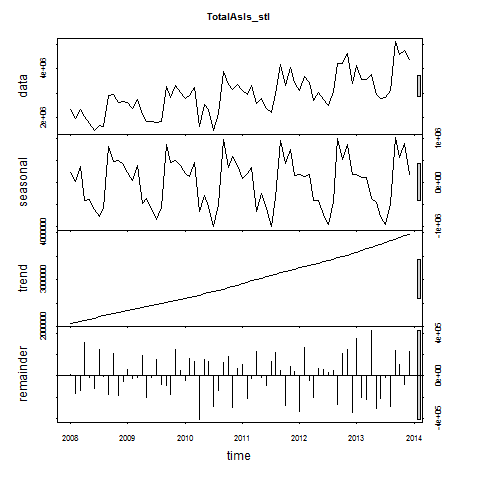


Figure 5

print(TotalAsIs\_stl)

## Call:  
## stl(x = TotalAsIs, s.window = 5)  
##   
## Components  
## seasonal trend remainder  
## Jan 2008 223320.67 2074233 15667.16  
## Feb 2008 17036.99 2096208 -163113.80  
## Mar 2008 361473.74 2118182 -133021.18  
## Apr 2008 -410834.24 2140157 310464.16  
## May 2008 -391831.93 2162114 -13317.80  
## Jun 2008 -608564.13 2184070 -117204.25  
## Jul 2008 -777993.52 2206027 251603.49  
## Aug 2008 -583615.66 2228213 -4927.72  
## Sep 2008 810939.36 2250400 -178453.09  
## Oct 2008 474131.86 2272586 212998.05  
## Nov 2008 488504.52 2294373 -186383.79  
## Dec 2008 395452.58 2316160 -55045.03  
## Jan 2009 217151.38 2337948 55473.99  
## Feb 2009 39716.91 2359168 -27558.10  
## Mar 2009 378507.21 2380389 -15109.96  
## Apr 2009 -467522.18 2401609 191220.87  
## May 2009 -371597.89 2425515 -203844.26  
## Jun 2009 -595724.45 2449421 -17474.54  
## Jul 2009 -827029.12 2473327 151013.28  
## Aug 2009 -567342.69 2495885 -76573.99  
## Sep 2009 843160.68 2518443 -90432.21  
## Oct 2009 447562.71 2541000 -169675.09  
## Nov 2009 497312.47 2562364 251099.75  
## Dec 2009 388265.67 2583727 50520.14  
## Jan 2010 201133.54 2605091 -45536.12  
## Feb 2010 122776.46 2628120 167436.40  
## Mar 2010 442825.47 2651150 133065.83  
## Apr 2010 -652923.75 2674179 -407367.50  
## May 2010 -301149.68 2698691 152615.46  
## Jun 2010 -543850.29 2723203 138292.09  
## Jul 2010 -985987.99 2747715 -287583.18  
## Aug 2010 -487941.31 2774544 -138081.68  
## Sep 2010 972415.73 2801373 124782.46  
## Oct 2010 343206.82 2828202 177544.55  
## Nov 2010 573281.74 2858572 -295909.05  
## Dec 2010 375326.75 2888943 68616.25  
## Jan 2011 84179.43 2919314 109367.89  
## Feb 2011 190940.11 2949475 -213752.60  
## Mar 2011 339598.68 2979637 -24451.98  
## Apr 2011 -661193.66 3009799 228473.57  
## May 2011 -252299.73 3037669 -11300.88  
## Jun 2011 -597799.74 3065538 -89511.39  
## Jul 2011 -1002974.31 3093408 132466.66  
## Aug 2011 -345401.48 3120526 216662.97  
## Sep 2011 951339.44 3147643 52548.18  
## Oct 2011 418464.54 3174761 -274541.80  
## Nov 2011 749466.48 3200972 86637.11  
## Dec 2011 166063.96 3227184 36595.48  
## Jan 2012 173825.10 3253395 -334131.81  
## Feb 2012 131526.89 3279250 268531.13  
## Mar 2012 171949.25 3305105 -43690.50  
## Apr 2012 -412193.90 3330961 -203867.63  
## May 2012 -414897.17 3358540 68124.29  
## Jun 2012 -723606.43 3386119 63515.20  
## Jul 2012 -957183.71 3413699 27319.12  
## Aug 2012 -438041.15 3441507 52189.27  
## Sep 2012 998725.79 3469315 -267244.98  
## Oct 2012 523934.85 3497123 207665.66  
## Nov 2012 847979.72 3527674 242886.44  
## Dec 2012 172550.29 3558224 -347101.49  
## Jan 2013 184195.89 3588775 346555.55  
## Feb 2013 114297.14 3623803 -202355.91  
## Mar 2013 121000.80 3658831 -218857.78  
## Apr 2013 -360531.42 3693859 426737.22  
## May 2013 -462506.26 3728897 -306457.92  
## Jun 2013 -759940.89 3763935 -216096.28  
## Jul 2013 -951772.71 3798973 -18456.45  
## Aug 2013 -468011.67 3834192 -282067.53  
## Sep 2013 1004335.28 3869411 234028.47  
## Oct 2013 554713.70 3904630 102800.01  
## Nov 2013 873598.66 3940742 -85027.53  
## Dec 2013 169104.03 3976853 226223.51

It is interesting to note that the almost linear trend is not seen in the individual segment for Total Etel (Figure 6). The individual trends for all flower exports run in partially opposite directions in the middle of the time scale, which causes the total As Is data trend to be linear. (Wheeler et al).

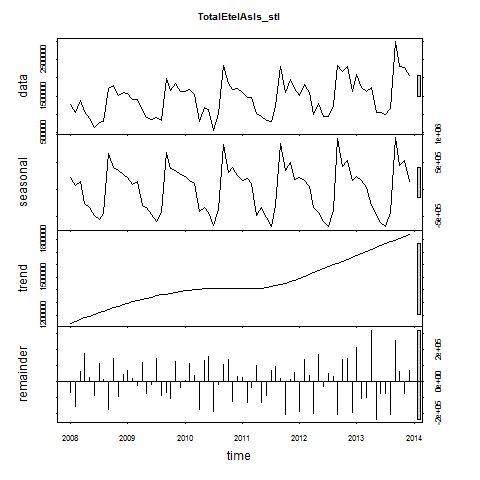


Figure 6

print(TotalEtelAsIs\_stl)

## Call:  
## stl(x = TotalEtelAsIs, s.window = 5)  
##   
## Components  
## seasonal trend remainder  
## Jan 2008 212543.90 1135393 -68268.686  
## Feb 2008 64724.69 1149373 -160772.697  
## Mar 2008 141647.42 1163353 62519.359  
## Apr 2008 -264999.95 1177333 178391.504  
## May 2008 -339776.29 1191010 22334.080  
## Jun 2008 -473456.92 1204687 -86751.057  
## Jul 2008 -560479.22 1218364 114773.472  
## Aug 2008 -439963.23 1231709 14995.207  
## Sep 2008 649360.43 1245054 -179149.730  
## Oct 2008 394715.11 1258400 142636.314  
## Nov 2008 343804.99 1270718 -96234.645  
## Dec 2008 272635.02 1283036 45653.230  
## Jan 2009 216526.66 1295354 71335.502  
## Feb 2009 85366.77 1304840 17181.430  
## Mar 2009 134597.06 1314326 -28121.833  
## Apr 2009 -299576.10 1323812 116864.362  
## May 2009 -339425.04 1333273 -73988.443  
## Jun 2009 -463451.32 1342735 -20407.903  
## Jul 2009 -588086.05 1352197 146023.090  
## Aug 2009 -428334.97 1359570 -88185.300  
## Sep 2009 684000.28 1366944 -69380.862  
## Oct 2009 378107.74 1374317 -104490.629  
## Nov 2009 350361.28 1380061 127414.131  
## Dec 2009 269269.49 1385804 -39982.767  
## Jan 2010 237811.58 1391548 8104.442  
## Feb 2010 164697.76 1396069 115393.780  
## Mar 2010 108887.92 1400591 40081.142  
## Apr 2010 -414588.58 1405112 -177054.835  
## May 2010 -341339.97 1406966 132774.500  
## Jun 2010 -426196.92 1408821 157400.399  
## Jul 2010 -672372.03 1410675 -187034.545  
## Aug 2010 -377863.24 1410092 -19686.420  
## Sep 2010 818966.77 1409509 107012.482  
## Oct 2010 310866.57 1408926 136471.594  
## Nov 2010 396380.37 1408474 -126731.546  
## Dec 2010 260838.84 1408023 30201.636  
## Jan 2011 162908.25 1407571 24787.879  
## Feb 2011 200345.42 1408473 -135290.178  
## Mar 2011 98058.73 1409375 -37705.374  
## Apr 2011 -475991.41 1410277 100364.893  
## May 2011 -326861.16 1414388 -134973.648  
## Jun 2011 -508609.13 1418499 -90586.968  
## Jul 2011 -690240.90 1422610 69706.513  
## Aug 2011 -303950.41 1431899 94863.512  
## Sep 2011 841755.54 1441187 20328.047  
## Oct 2011 348453.95 1450476 -207345.878  
## Nov 2011 482953.99 1463581 14139.587  
## Dec 2011 177522.68 1476687 59781.396  
## Jan 2012 219295.41 1489792 -189339.836  
## Feb 2012 167343.82 1504763 140790.260  
## Mar 2012 47565.92 1519733 39980.664  
## Apr 2012 -327162.02 1534704 -199519.887  
## May 2012 -429104.97 1550797 170291.408  
## Jun 2012 -592640.70 1566889 -34090.529  
## Jul 2012 -686742.76 1582982 49689.871  
## Aug 2012 -391971.46 1597629 29488.941  
## Sep 2012 926501.74 1612275 -208442.888  
## Oct 2012 411978.67 1626922 138994.561  
## Nov 2012 520321.23 1641216 144786.748  
## Dec 2012 157545.15 1655510 -194908.427  
## Jan 2013 228279.54 1669805 211412.939  
## Feb 2013 159093.45 1685663 -106559.227  
## Mar 2013 32529.79 1701521 -100106.827  
## Apr 2013 -295207.15 1717379 322919.852  
## May 2013 -454478.79 1733056 -239128.325  
## Jun 2013 -615888.66 1748733 -78643.269  
## Jul 2013 -688356.21 1764410 -72887.536  
## Aug 2013 -417329.54 1779986 -207981.755  
## Sep 2013 944212.59 1795563 261153.561  
## Oct 2013 431257.90 1811139 63207.703  
## Nov 2013 530407.87 1827035 -72771.039  
## Dec 2013 148706.84 1842931 70522.228

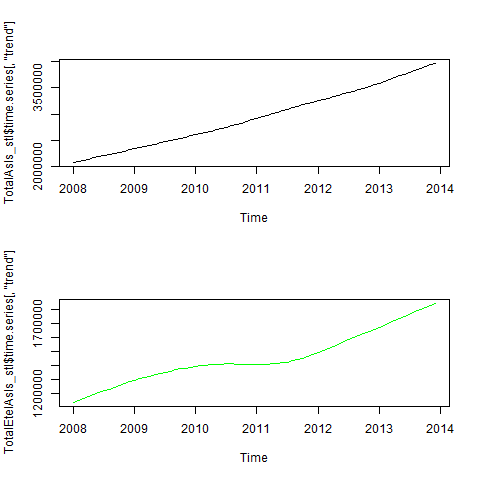


Figure 7

The modification of the seasonality component can also be changed into a montly view. It only makes sense to do this if the seasonality component as the trend looks almost identical and the remainder is then randomly spread. (Wheeler et al.)

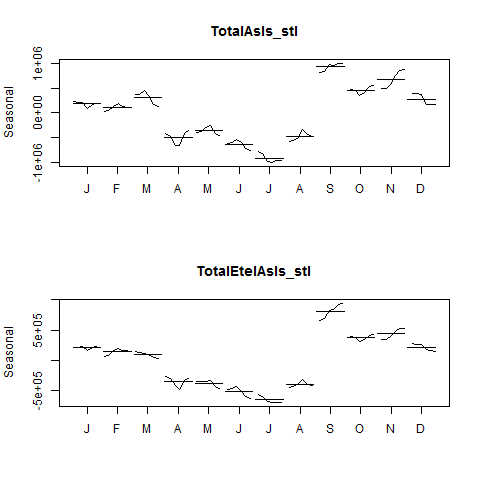


Figure 8

## 4.0 Correlation of Different External Indicators

### 4.1 Definition of the Indicators and Their Correlation with the Basic Data

The following indicators are to be tested:

1. Monthly Change in Export Price Index (CEPI)
2. Monthly Satisfaction Index (SI) government based data
3. Average monthly temperatures in Chulwalar
4. Monthly births in Chulwalar
5. Monthly Satisfaction Index (SI) external index
6. Yearly Exports from Urbano
7. Yearly number of Globalisation Party members in Chulwalar
8. Monthly Average Export Price Index for Chulwalar
9. Monthly Producer Price Index (PPI) for Etel in Chulwalar
10. National Holidays
11. Chulwalar Index (Total value of all companies in Chulwalar)
12. Monthly Inflation rate in Chulwalar
13. Proposed spending for National Holidays
14. Influence of National Holiday

The indicators will be converted into individual vectors and subsequently converted into time series. The correlation of the indicators will then be tested against the As Is exports for Chulwalar.

source("Scripts/correlateIndicators.r")

#### Monthly Changine in Export Price (CEPI):

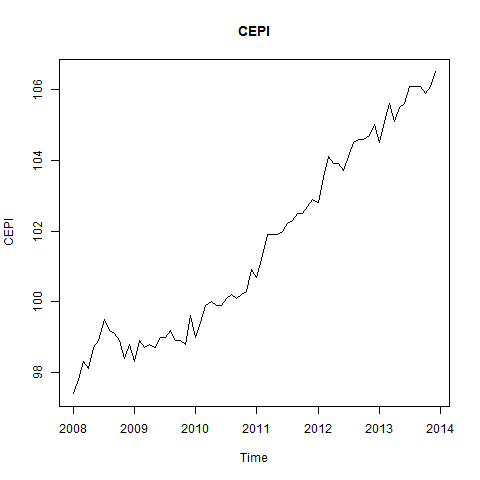


Figure 9

#check correlation of CEPI   
print(cor(TotalAsIs, CEPI))

## [1] 0.663925

print(cor(TotalEtelAsIs, CEPI))

## [1] 0.339713

The CEPI correlates very well with the efak exports.

#### Monthly Satisfaction Index (SI) Government Based Data:

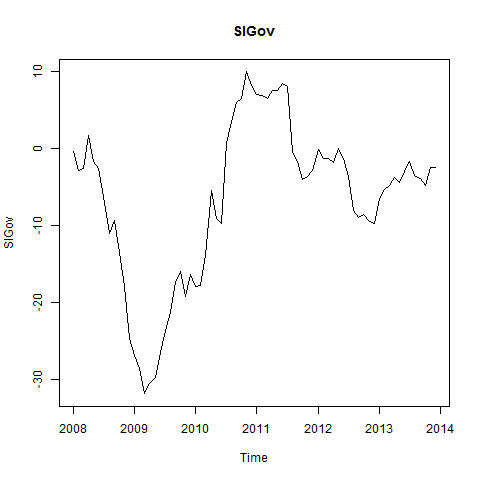


Figure 10

#check correlation of SIGov  
print(cor(TotalAsIs, SIGov))

## [1] 0.2007768

print(cor(TotalEtelAsIs, SIGov))

## [1] 0.002556094

The Satisfaction Index does not show any particular correlation with any of the exports data.

#### Average Monthly Temperatures in Chulwalar:

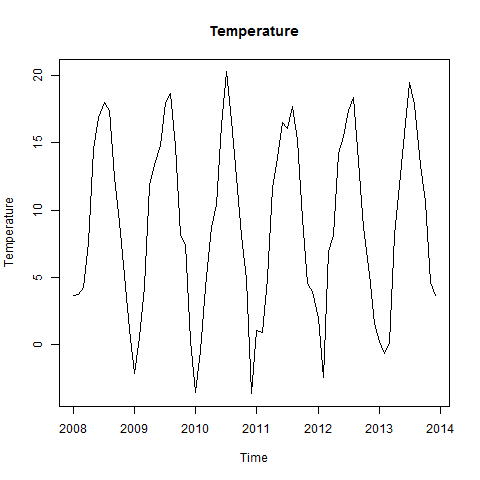


Figure 11

#check correlation of Temperature  
print(cor(TotalAsIs, Temperature))

## [1] -0.3429684

print(cor(TotalEtelAsIs, Temperature))

## [1] -0.453138

The temperatures have a negative correlation, exports increase in the colder months. However, the relationship is only stronger with blue Etels.

#### Monthly Births in Chulwalar:

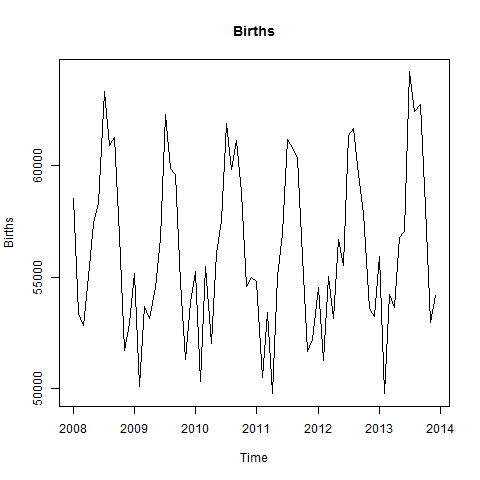


Figure 12

#check correlation of Births  
print(cor(TotalAsIs, Births))

## [1] -0.1190228

print(cor(TotalEtelAsIs, Births))

## [1] -0.1504242

The consideration by Chulwalar's experts was that expecting new parents to try to export more products to pay for the cost of a new child. However, this could not be confirmed.

#### Monthly Satisfaction Index (SI) External Index:

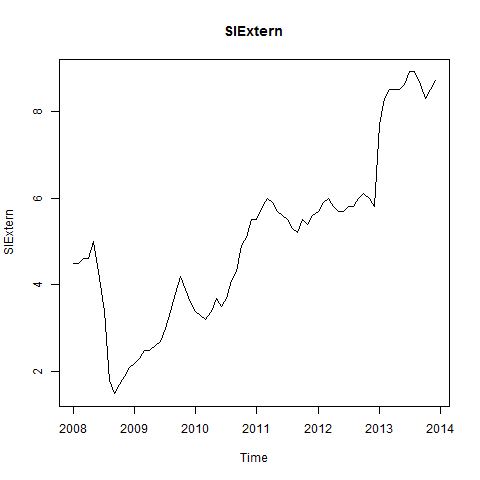


Figure 13

#check correlation of SIExtern  
print(cor(TotalAsIs, SIExtern))

## [1] 0.5883122

print(cor(TotalEtelAsIs, SIExtern))

## [1] 0.2865672

This indicator also has a high correlation with Efak exports.

#### Yearly Exports from Urbano:

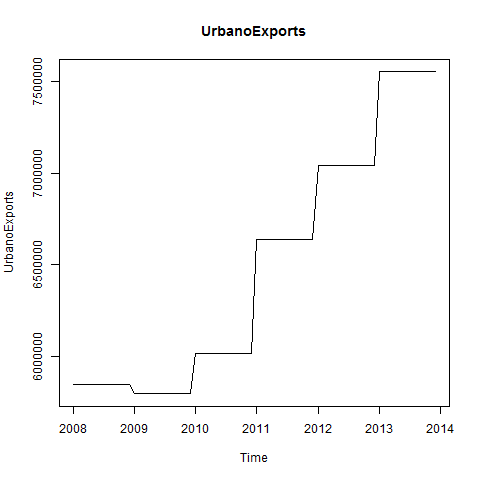


Figure 14

#check correlation of SIExtern  
print(cor(TotalAsIs, UrbanoExports))

## [1] 0.638178

print(cor(TotalEtelAsIs, UrbanoExports))

## [1] 0.3182532

This indicator also has a high correlation with Efak exports. The Wuge exports also show a correlation. Unfortunatly it was not possible to find other useful indicators based on exports from Urbano, due to possible informers being very restrictive with information.

#### Yearly Number of Globalisation Party Members in Chulwalar:

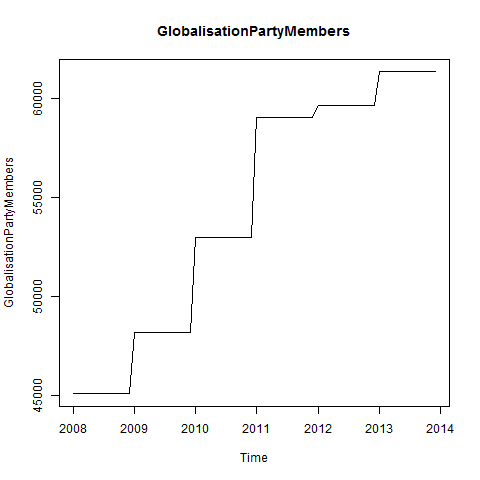


Figure 15

#check correlation of GlobalisationPartyMembers  
print(cor(TotalAsIs, GlobalisationPartyMembers))

## [1] 0.630084

print(cor(TotalEtelAsIs, GlobalisationPartyMembers))

## [1] 0.2994635

There is a similar picture here to that of Urbano Exports.It should however be noted that there is a continuos growth here and that the yearly view could lead to the data appearing to correlate, although this could just be due to an increase in trend. Although this could also be true for the Urbano Exports, the trend seems logical due to the Chulwalar's exports growing in accordance with the Urbano's Exports.

#### Monthly Average Export Price Index for Chulwalar:

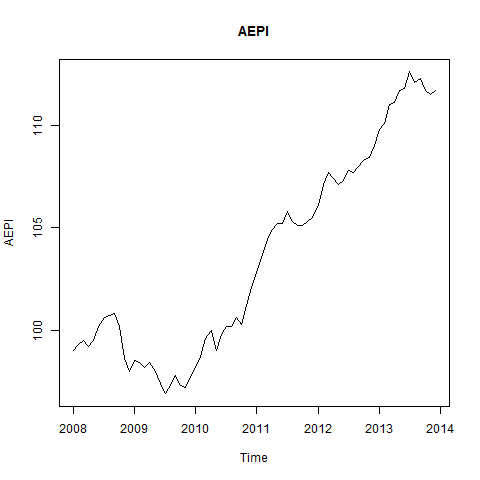


Figure 16

#check correlation of AEPI  
print(cor(TotalAsIs, AEPI))

## [1] 0.625232

print(cor(TotalEtelAsIs, AEPI))

## [1] 0.3035506

The continuous growth leads to a good correlation here too. See Above.

#### Monthly Producer Price Index (PPI) for Etel in Chulwalar:

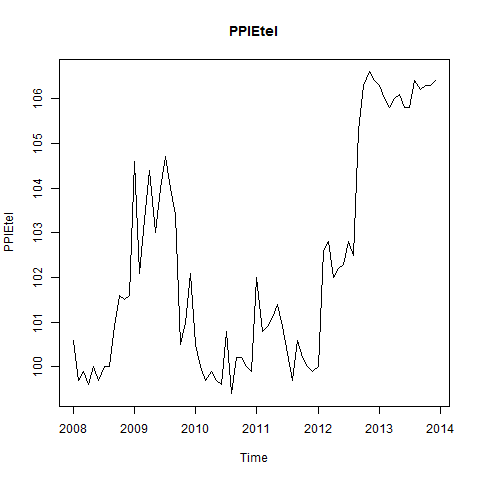


Figure 17

#check correlation of PPIEtel  
print(cor(TotalAsIs, PPIEtel))

## [1] 0.4836129

print(cor(TotalEtelAsIs, PPIEtel))

## [1] 0.3374707

This indicator does not give the expected results. It does not show any correlation worth mentioning, not even with the Etel segment.

#### National Holidays:

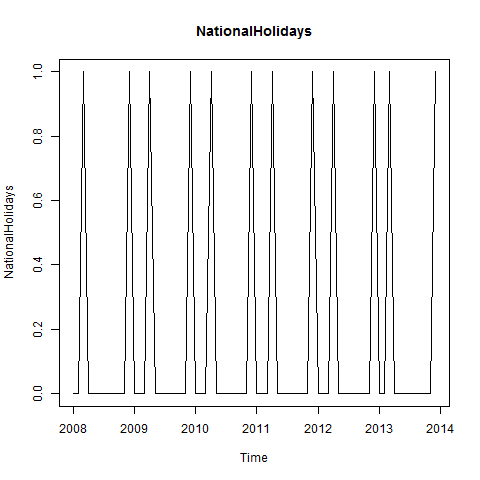


Figure 18

#check correlation of NationalHolidays  
print(cor(TotalAsIs, NationalHolidays))

## [1] -0.007883708

print(cor(TotalEtelAsIs, NationalHolidays))

## [1] -0.01081446

The months April and December do not correlate well with the exports data. However later tests will show that these are worth considering. The missing correlation is just due to the sparse structure of the NationalHolidays time series.

#### Chulwalar Index (Total Value of all Companies in Chulwalar):

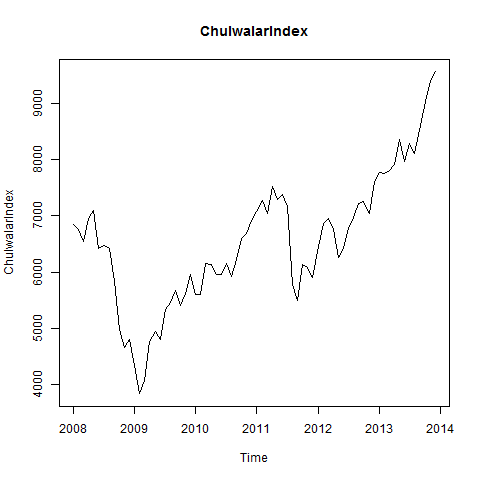


Figure 19

#check correlation of ChulwalarIndex  
print(cor(TotalAsIs, ChulwalarIndex))

## [1] 0.4837017

print(cor(TotalEtelAsIs, ChulwalarIndex))

## [1] 0.2209171

No particular findings.

#### Monthly Inflation Rate in Chulwalar:

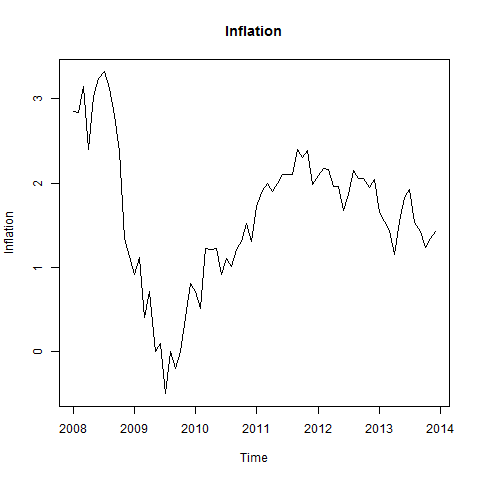


Figure 20

#check correlation of Inflation  
print(cor(TotalAsIs, Inflation))

## [1] 0.002438708

print(cor(TotalEtelAsIs, Inflation))

## [1] -0.08378282

No particular findings.

#### Proposed Spending for Independence Day Presents:

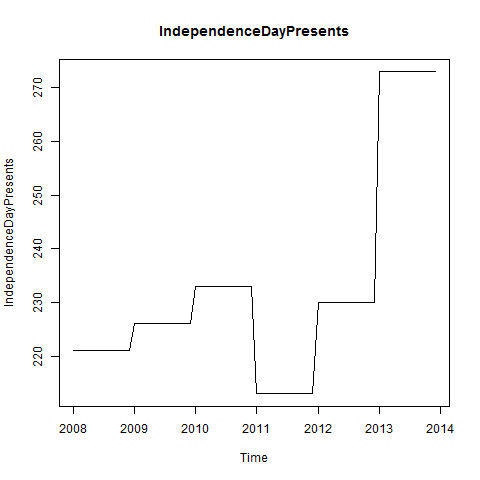


Figure 21

#check correlation of IndependenceDayPresents  
cor(TotalAsIs, IndependenceDayPresents)

## [1] 0.4359522

cor(TotalEtelAsIs, IndependenceDayPresents)

## [1] 0.2872013

No particular findings.

#### Influence of National Holidays:

This indicator is an experiment where the influence of National Holidays is extended into the months leading up to the holiday. However later tests show that this indicator is no better for forecasting than the orignial National Holidays indicator.

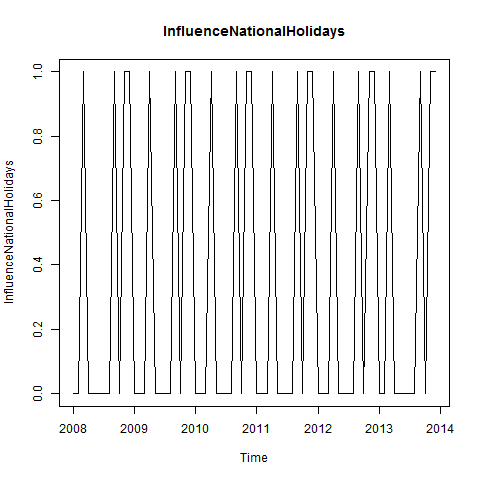


Figure 22

#check correlation of InfluenceNationalHolidays  
print(cor(TotalAsIs, InfluenceNationalHolidays))

## [1] 0.3717463

print(cor(TotalEtelAsIs, InfluenceNationalHolidays))

## [1] 0.4535836

#check that the data import has worked  
head(CEPIVector)

## [1] 97.4 97.8 98.3 98.1 98.7 98.9

head(SIGovVector)

## [1] -0.4 -2.9 -2.7 1.7 -1.7 -2.6

head(TemperatureVector)

## [1] 3.6 3.7 4.2 7.6 14.5 16.9

head(BirthsVector)

## [1] 58519 53370 52852 55048 57398 58313

head(SIExternVector)

## [1] 4.5 4.5 4.6 4.6 5.0 4.3

head(UrbanoExportsVector)

## [1] 5850000 5850000 5850000 5850000 5850000 5850000

head(GlobalisationPartyMembersVector)

## [1] 45089 45089 45089 45089 45089 45089

head(AEPIVector)

## [1] 99.0 99.3 99.5 99.2 99.5 100.2

head(PPIEtelVector)

## [1] 100.6 99.7 99.9 99.6 100.0 99.7

head(NationalHolidaysVector)

## [1] 0 0 1 0 0 0

head(ChulwalarIndexVector)

## [1] 6851.75 6748.13 6534.97 6948.82 7096.79 6418.32

head(InflationVector)

## [1] 2.85 2.84 3.15 2.40 3.03 3.24

head(IndependenceDayPresentsVector)

## [1] 221 221 221 221 221 221

### 4.2 Correlation of the Indicators with a Time Offset

The External Satisfaction Index indicator is to be offset by one month, to see if the index change makes itself first noticeable on exports in the following months.

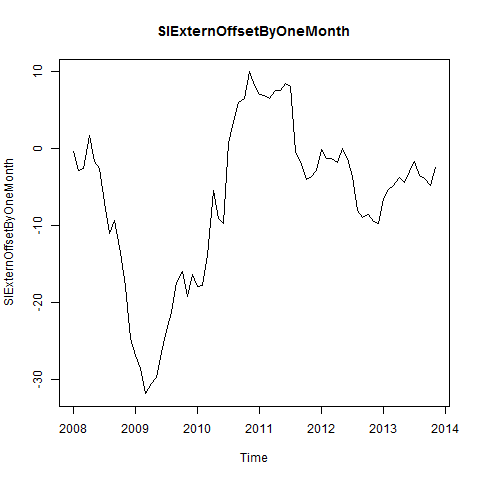


Figure 23

#### Here December 2013 has been deleted from the time series:

print(cor(TotalAsIsWithoutDec12013, SIExternOffsetByOneMonth))

## [1] 0.1952995

print(cor(TotalEtelAsIsWithoutDec12013, SIExternOffsetByOneMonth))

## [1] -0.004445279

print(summary(TotalAsIsWithoutDec2013\_lm))

##   
## Call:  
## lm(formula = TotalAsIsWithoutDec12013 ~ SIExternOffsetByOneMonth,   
## data = TotalAsIsWithoutDec12013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1560602 -560765 246 437927 2142998   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3025619 114816 26.352 <2e-16 \*\*\*  
## SIExternOffsetByOneMonth 15211 9196 1.654 0.103   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 812500 on 69 degrees of freedom  
## Multiple R-squared: 0.03814, Adjusted R-squared: 0.0242   
## F-statistic: 2.736 on 1 and 69 DF, p-value: 0.1026

#### Offsett SIGov Indicator by two months:

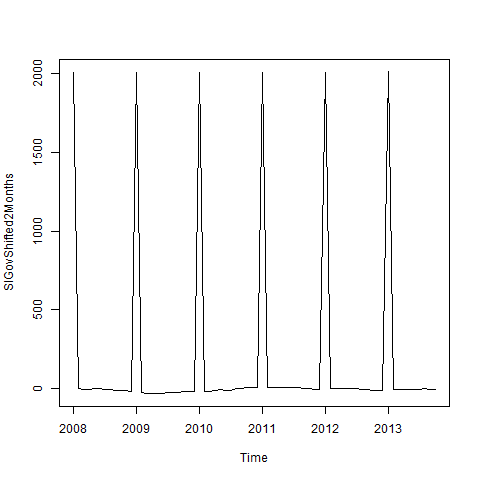


Figure 24

#### Delete November and December 2013 from the time series:

print(cor(TotalAsIsWithoutNovDec2013, SIGovShifted2Months))

## [1] 0.0446355

print(cor(TotalEtelAsIsWithoutNovDec2013, SIGovShifted2Months))

## [1] 0.1173295

print(summary(TotalAsIsWithoutNovDec2013))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1458000 2353000 2901000 2897000 3329000 5108000

The correlation of the indicators has not really been improved by the offsets, so we will not continue with this approach.

### 4.3 Correlation of the Indicators with Each Other

In order to test which indicators could be used in a model with eachother, we need to look at the correlation of the indicators with eachother. All thirteen indicators will be compared with eachother in a correlation coefficient matrix. First of all it is necessary to summarise all indicators in a matrix. ####Establish the standardised data matrix:

print(head(IndicatorsmatrixStandardised))

## CEPIVector SIGovVector TemperatureVector BirthsVector SIGovVector  
## [1,] -1.5292232 0.6019786 -0.8015830 0.6357158 0.6019786  
## [2,] -1.3821236 0.3638332 -0.7867017 -0.7548522 0.3638332  
## [3,] -1.1982491 0.3828849 -0.7122954 -0.8947462 0.3828849  
## [4,] -1.2717989 0.8020208 -0.2063327 -0.3016821 0.8020208  
## [5,] -1.0511494 0.4781430 0.8204739 0.3329722 0.4781430  
## [6,] -0.9775996 0.3924107 1.1776240 0.5800823 0.3924107  
## UrbanoExportsVector GlobalisationPartyMembersVector AEPIVector  
## [1,] -0.9637871 -1.507017 -0.9164668  
## [2,] -0.9637871 -1.507017 -0.8561509  
## [3,] -0.9637871 -1.507017 -0.8159403  
## [4,] -0.9637871 -1.507017 -0.8762562  
## [5,] -0.9637871 -1.507017 -0.8159403  
## [6,] -0.9637871 -1.507017 -0.6752032  
## PPIEtel NationalHolidaysVector ChulwalarIndexVector  
## [1,] -0.6937760 -0.4440971 0.25840362  
## [2,] -1.0625757 -0.4440971 0.17119839  
## [3,] -0.9806202 2.2204854 -0.00819425  
## [4,] -1.1035535 -0.4440971 0.34009645  
## [5,] -0.9396425 -0.4440971 0.46462605  
## [6,] -1.0625757 -0.4440971 -0.10636535  
## InflationVector IndependenceDayPresentsVector  
## [1,] 1.4687496 -0.6048427  
## [2,] 1.4569353 -0.6048427  
## [3,] 1.8231792 -0.6048427  
## [4,] 0.9371052 -0.6048427  
## [5,] 1.6814073 -0.6048427  
## [6,] 1.9295080 -0.6048427

#### The dimensions of the matrix are determined by the number of indicators:

print(NumberOfIndicators)

## [1] 72

#### Produce the IndicatorsCorrelationCoefficientMatrix.

print(IndicatorsCorrelationCoefficientMatrix)

## CEPIVector SIGovVector TemperatureVector  
## CEPIVector 1.00000000 0.38443508 0.061196862  
## SIGovVector 0.38443508 1.00000000 0.088109231  
## TemperatureVector 0.06119686 0.08810923 1.000000000  
## BirthsVector 0.08872676 0.12753378 0.744270853  
## SIGovVector 0.38443508 1.00000000 0.088109231  
## UrbanoExportsVector 0.97660022 0.40700264 -0.001244458  
## GlobalisationPartyMembersVector 0.91557949 0.49433954 -0.009695828  
## AEPIVector 0.97697428 0.45955807 0.055196145  
## PPIEtel 0.65446147 -0.23602751 -0.013959906  
## NationalHolidaysVector 0.04830482 -0.02025819 -0.316148237  
## ChulwalarIndexVector 0.76208613 0.63652935 0.036317166  
## InflationVector 0.16379793 0.55866085 0.054966975  
## IndependenceDayPresentsVector 0.64887003 0.03237405 -0.040110690  
## BirthsVector SIGovVector  
## CEPIVector 0.08872676 0.38443508  
## SIGovVector 0.12753378 1.00000000  
## TemperatureVector 0.74427085 0.08810923  
## BirthsVector 1.00000000 0.12753378  
## SIGovVector 0.12753378 1.00000000  
## UrbanoExportsVector 0.03139251 0.40700264  
## GlobalisationPartyMembersVector -0.01768274 0.49433954  
## AEPIVector 0.09673808 0.45955807  
## PPIEtel 0.05960084 -0.23602751  
## NationalHolidaysVector -0.37785553 -0.02025819  
## ChulwalarIndexVector 0.11795545 0.63652935  
## InflationVector 0.11231574 0.55866085  
## IndependenceDayPresentsVector 0.10063892 0.03237405  
## UrbanoExportsVector  
## CEPIVector 9.766002e-01  
## SIGovVector 4.070026e-01  
## TemperatureVector -1.244458e-03  
## BirthsVector 3.139251e-02  
## SIGovVector 4.070026e-01  
## UrbanoExportsVector 1.000000e+00  
## GlobalisationPartyMembersVector 9.121013e-01  
## AEPIVector 9.827920e-01  
## PPIEtel 6.521194e-01  
## NationalHolidaysVector -1.876433e-17  
## ChulwalarIndexVector 7.856783e-01  
## InflationVector 1.985267e-01  
## IndependenceDayPresentsVector 6.699996e-01  
## GlobalisationPartyMembersVector AEPIVector  
## CEPIVector 9.155795e-01 0.97697428  
## SIGovVector 4.943395e-01 0.45955807  
## TemperatureVector -9.695828e-03 0.05519615  
## BirthsVector -1.768274e-02 0.09673808  
## SIGovVector 4.943395e-01 0.45955807  
## UrbanoExportsVector 9.121013e-01 0.98279202  
## GlobalisationPartyMembersVector 1.000000e+00 0.88225030  
## AEPIVector 8.822503e-01 1.00000000  
## PPIEtel 4.583532e-01 0.62229942  
## NationalHolidaysVector 1.250956e-17 0.01886347  
## ChulwalarIndexVector 6.647301e-01 0.80958140  
## InflationVector 9.009471e-02 0.30646256  
## IndependenceDayPresentsVector 4.606363e-01 0.64313387  
## PPIEtel NationalHolidaysVector  
## CEPIVector 0.65446147 4.830482e-02  
## SIGovVector -0.23602751 -2.025819e-02  
## TemperatureVector -0.01395991 -3.161482e-01  
## BirthsVector 0.05960084 -3.778555e-01  
## SIGovVector -0.23602751 -2.025819e-02  
## UrbanoExportsVector 0.65211942 -1.876433e-17  
## GlobalisationPartyMembersVector 0.45835315 1.250956e-17  
## AEPIVector 0.62229942 1.886347e-02  
## PPIEtel 1.00000000 2.896317e-02  
## NationalHolidaysVector 0.02896317 1.000000e+00  
## ChulwalarIndexVector 0.45429124 5.430333e-02  
## InflationVector -0.25048037 -9.384951e-03  
## IndependenceDayPresentsVector 0.71474813 0.000000e+00  
## ChulwalarIndexVector InflationVector  
## CEPIVector 0.76208613 0.163797927  
## SIGovVector 0.63652935 0.558660851  
## TemperatureVector 0.03631717 0.054966975  
## BirthsVector 0.11795545 0.112315739  
## SIGovVector 0.63652935 0.558660851  
## UrbanoExportsVector 0.78567826 0.198526676  
## GlobalisationPartyMembersVector 0.66473014 0.090094706  
## AEPIVector 0.80958140 0.306462559  
## PPIEtel 0.45429124 -0.250480368  
## NationalHolidaysVector 0.05430333 -0.009384951  
## ChulwalarIndexVector 1.00000000 0.341955823  
## InflationVector 0.34195582 1.000000000  
## IndependenceDayPresentsVector 0.62615921 -0.185842679  
## IndependenceDayPresentsVector  
## CEPIVector 0.64887003  
## SIGovVector 0.03237405  
## TemperatureVector -0.04011069  
## BirthsVector 0.10063892  
## SIGovVector 0.03237405  
## UrbanoExportsVector 0.66999963  
## GlobalisationPartyMembersVector 0.46063633  
## AEPIVector 0.64313387  
## PPIEtel 0.71474813  
## NationalHolidaysVector 0.00000000  
## ChulwalarIndexVector 0.62615921  
## InflationVector -0.18584268  
## IndependenceDayPresentsVector 1.00000000

The Correlation Coefficient Matrix shows that CEPI has a high correlation with SIExtern, UrbanoExports, GlobalisationPartyMembers and AEPI. These will become the set of indicators used later, although we are aware of the dangers of multicollinearity.However it is interesting to note that NationalHolidays, UrbanoExports, GlobalisationPartyMembers have a very low correlation with one another. Therefore these will also become a set of indicators used later.

## 5.0 Development of Forecasting Models

With help of the tslm function, we will produce a model based on the time series. Possible inputs could be Trend and Seasonality as well as the time series of the indicators.

### 5.1 ModelWithAlllIndicators and with Each Indicator Individually

source("Scripts/developModels.r")

#### All Indiators in one model:

print(summary(ModelWithAlllIndicators))

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + CEPI + SIGov + Temperature +   
## Births + SIExtern + UrbanoExports + GlobalisationPartyMembers +   
## AEPI + PPIEtel + NationalHolidays + ChulwalarIndex + Inflation +   
## IndependenceDayPresents)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -458389 -119426 1119 165463 342741   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.982e+05 2.301e+07 -0.039 0.96904   
## trend 3.176e+03 3.849e+04 0.083 0.93458   
## season2 3.146e+05 2.678e+05 1.175 0.24624   
## season3 5.172e+05 2.649e+05 1.953 0.05695 .   
## season4 2.972e+05 3.413e+05 0.871 0.38836   
## season5 -7.277e+04 3.661e+05 -0.199 0.84333   
## season6 -2.597e+05 4.199e+05 -0.618 0.53932   
## season7 -7.550e+05 5.225e+05 -1.445 0.15525   
## season8 -2.869e+05 4.990e+05 -0.575 0.56809   
## season9 1.066e+06 4.225e+05 2.523 0.01517 \*   
## season10 8.033e+05 3.352e+05 2.396 0.02068 \*   
## season11 1.226e+06 3.555e+05 3.449 0.00122 \*\*  
## season12 9.734e+05 3.645e+05 2.670 0.01044 \*   
## CEPI -3.551e+04 2.516e+05 -0.141 0.88838   
## SIGov -1.506e+04 9.150e+03 -1.646 0.10657   
## Temperature -3.108e+04 2.069e+04 -1.502 0.14003   
## Births 8.045e+01 3.894e+01 2.066 0.04448 \*   
## SIExtern 3.706e+04 5.872e+04 0.631 0.53109   
## UrbanoExports 5.323e-01 5.675e-01 0.938 0.35317   
## GlobalisationPartyMembers 7.324e+01 6.583e+01 1.113 0.27163   
## AEPI -6.003e+04 7.476e+04 -0.803 0.42612   
## PPIEtel 7.799e+03 3.622e+04 0.215 0.83048   
## NationalHolidays -3.192e+05 1.718e+05 -1.858 0.06963 .   
## ChulwalarIndex 6.102e+01 7.545e+01 0.809 0.42284   
## Inflation 7.058e+04 1.555e+05 0.454 0.65213   
## IndependenceDayPresents 4.211e+01 6.187e+03 0.007 0.99460   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 249400 on 46 degrees of freedom  
## Multiple R-squared: 0.9421, Adjusted R-squared: 0.9106   
## F-statistic: 29.94 on 25 and 46 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9106

#### CEPI:

The CEPI Indicator correlated best with total exports. Indeed the multiple R² improved the model slighltly compared to the simple ModelWithTrendAndSeasonalityOnly; however, the adjusted R² remains the same.

summary(ModelWithCEPI)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + CEPI)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -670684 -142117 7024 168664 495366   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2946424 5153463 -0.572 0.569710   
## trend 19698 6926 2.844 0.006145 \*\*   
## season2 -153665 153683 -1.000 0.321523   
## season3 8677 156732 0.055 0.956039   
## season4 -634082 154130 -4.114 0.000124 \*\*\*  
## season5 -648875 154240 -4.207 9.09e-05 \*\*\*  
## season6 -906108 153943 -5.886 2.10e-07 \*\*\*  
## season7 -1112258 155872 -7.136 1.73e-09 \*\*\*  
## season8 -755527 155490 -4.859 9.34e-06 \*\*\*  
## season9 683382 154129 4.434 4.18e-05 \*\*\*  
## season10 287071 153168 1.874 0.065940 .   
## season11 465878 152885 3.047 0.003474 \*\*   
## season12 50523 154712 0.327 0.745176   
## CEPI 53135 53376 0.995 0.323636   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263300 on 58 degrees of freedom  
## Multiple R-squared: 0.9187, Adjusted R-squared: 0.9004   
## F-statistic: 50.39 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9004

##### SIGov:

The Satisfaction Index (gov) hardly changes the function of the model.

summary(ModelWithSIGov)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + SIGov)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -697126 -157160 22782 161382 486711   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2154993 126151 17.083 < 2e-16 \*\*\*  
## trend 26826 1656 16.196 < 2e-16 \*\*\*  
## season2 -133003 152843 -0.870 0.387782   
## season3 44751 152866 0.293 0.770763   
## season4 -606128 152952 -3.963 0.000205 \*\*\*  
## season5 -622634 152935 -4.071 0.000143 \*\*\*  
## season6 -881666 153013 -5.762 3.35e-07 \*\*\*  
## season7 -1075681 153183 -7.022 2.69e-09 \*\*\*  
## season8 -726089 153194 -4.740 1.43e-05 \*\*\*  
## season9 705690 153291 4.604 2.31e-05 \*\*\*  
## season10 297924 153457 1.941 0.057071 .   
## season11 468770 153659 3.051 0.003439 \*\*   
## season12 68494 153977 0.445 0.658095   
## SIGov -2003 3274 -0.612 0.543174   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 264700 on 58 degrees of freedom  
## Multiple R-squared: 0.9178, Adjusted R-squared: 0.8994   
## F-statistic: 49.81 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8994

#### Temperature:

summary(ModelWithTemperature)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + Temperature)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -706803 -154965 23511 160215 483373   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2185999 118102 18.509 < 2e-16 \*\*\*  
## trend 26367 1526 17.278 < 2e-16 \*\*\*  
## season2 -130163 152875 -0.851 0.39803   
## season3 91513 171443 0.534 0.59553   
## season4 -504879 236159 -2.138 0.03675 \*   
## season5 -476774 296010 -1.611 0.11268   
## season6 -703539 345717 -2.035 0.04643 \*   
## season7 -873818 386156 -2.263 0.02740 \*   
## season8 -524053 378812 -1.383 0.17184   
## season9 858772 305542 2.811 0.00673 \*\*   
## season10 401142 232466 1.726 0.08974 .   
## season11 530742 183985 2.885 0.00549 \*\*   
## season12 85552 155077 0.552 0.58329   
## Temperature -11344 19587 -0.579 0.56473   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 264800 on 58 degrees of freedom  
## Multiple R-squared: 0.9177, Adjusted R-squared: 0.8993   
## F-statistic: 49.78 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8993

#### Births:

summary(ModelWithBirths)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + Births)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -648252 -106586 23124 166173 443675   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.485e+06 1.452e+06 -1.023 0.310779   
## trend 2.633e+04 1.449e+03 18.163 < 2e-16 \*\*\*  
## season2 1.856e+05 1.918e+05 0.968 0.337199   
## season3 1.510e+05 1.512e+05 0.998 0.322286   
## season4 -4.181e+05 1.639e+05 -2.551 0.013402 \*   
## season5 -6.484e+05 1.459e+05 -4.444 4.04e-05 \*\*\*  
## season6 -9.698e+05 1.496e+05 -6.482 2.16e-08 \*\*\*  
## season7 -1.518e+06 2.265e+05 -6.704 9.20e-09 \*\*\*  
## season8 -1.068e+06 1.992e+05 -5.364 1.48e-06 \*\*\*  
## season9 3.721e+05 1.966e+05 1.893 0.063345 .   
## season10 2.114e+05 1.502e+05 1.407 0.164622   
## season11 6.744e+05 1.666e+05 4.049 0.000155 \*\*\*  
## season12 2.147e+05 1.565e+05 1.372 0.175458   
## Births 6.589e+01 2.601e+01 2.533 0.014026 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 251900 on 58 degrees of freedom  
## Multiple R-squared: 0.9255, Adjusted R-squared: 0.9088   
## F-statistic: 55.43 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9088

#### SIExtern:

summary(ModelWithSIExtern)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + SIExtern)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -667444 -154044 -5891 162628 473612   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2124425 137869 15.409 < 2e-16 \*\*\*  
## trend 24163 3191 7.572 3.20e-10 \*\*\*  
## season2 -133767 152487 -0.877 0.383979   
## season3 43156 152535 0.283 0.778243   
## season4 -609825 152516 -3.998 0.000183 \*\*\*  
## season5 -624208 152569 -4.091 0.000134 \*\*\*  
## season6 -877941 152767 -5.747 3.55e-07 \*\*\*  
## season7 -1071287 153027 -7.001 2.92e-09 \*\*\*  
## season8 -710173 153873 -4.615 2.22e-05 \*\*\*  
## season9 722059 154265 4.681 1.76e-05 \*\*\*  
## season10 312879 153885 2.033 0.046617 \*   
## season11 486780 154278 3.155 0.002542 \*\*   
## season12 88661 154442 0.574 0.568139   
## SIExtern 26522 32881 0.807 0.423187   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 264000 on 58 degrees of freedom  
## Multiple R-squared: 0.9182, Adjusted R-squared: 0.8998   
## F-statistic: 50.07 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8998

#### UrbanoExports:

Indicator with adjusted R² shows a better result than the reference model (ModelWithTrendAndSeasonalityOnly). The individual months are also very significant.

summary(ModelWithTotalUrbanoExports)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + UrbanoExports)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -651323 -145654 7297 172919 469753   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.232e+06 9.485e+05 1.299 0.199178   
## trend 2.118e+04 5.414e+03 3.912 0.000243 \*\*\*  
## season2 -1.259e+05 1.521e+05 -0.828 0.411030   
## season3 5.708e+04 1.524e+05 0.375 0.709261   
## season4 -5.934e+05 1.528e+05 -3.882 0.000267 \*\*\*  
## season5 -6.025e+05 1.535e+05 -3.925 0.000232 \*\*\*  
## season6 -8.568e+05 1.544e+05 -5.551 7.40e-07 \*\*\*  
## season7 -1.048e+06 1.554e+05 -6.741 7.96e-09 \*\*\*  
## season8 -6.879e+05 1.566e+05 -4.392 4.82e-05 \*\*\*  
## season9 7.477e+05 1.580e+05 4.732 1.47e-05 \*\*\*  
## season10 3.473e+05 1.596e+05 2.176 0.033640 \*   
## season11 5.246e+05 1.613e+05 3.252 0.001913 \*\*   
## season12 1.317e+05 1.632e+05 0.807 0.423118   
## UrbanoExports 1.717e-01 1.700e-01 1.010 0.316698   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263200 on 58 degrees of freedom  
## Multiple R-squared: 0.9187, Adjusted R-squared: 0.9005   
## F-statistic: 50.41 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9005

#### GlobalisationPartyMembers:

summary(ModelWithGlobalisationPartyMembers)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + GlobalisationPartyMembers)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -696019 -161848 22345 172443 478347   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.629e+06 9.653e+05 2.724 0.008517 \*\*   
## trend 2.928e+04 6.311e+03 4.640 2.04e-05 \*\*\*  
## season2 -1.340e+05 1.531e+05 -0.875 0.385097   
## season3 4.087e+04 1.535e+05 0.266 0.791010   
## season4 -6.177e+05 1.542e+05 -4.006 0.000178 \*\*\*  
## season5 -6.350e+05 1.551e+05 -4.094 0.000133 \*\*\*  
## season6 -8.973e+05 1.562e+05 -5.744 3.59e-07 \*\*\*  
## season7 -1.096e+06 1.576e+05 -6.955 3.49e-09 \*\*\*  
## season8 -7.447e+05 1.593e+05 -4.676 1.79e-05 \*\*\*  
## season9 6.829e+05 1.611e+05 4.238 8.18e-05 \*\*\*  
## season10 2.743e+05 1.632e+05 1.681 0.098191 .   
## season11 4.435e+05 1.655e+05 2.680 0.009573 \*\*   
## season12 4.252e+04 1.680e+05 0.253 0.801132   
## GlobalisationPartyMembers -9.840e+00 2.111e+01 -0.466 0.642806   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 265000 on 58 degrees of freedom  
## Multiple R-squared: 0.9176, Adjusted R-squared: 0.8991   
## F-statistic: 49.67 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8991

#### AEPI:

summary(ModelWithAEPI)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + AEPI)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -668980 -141696 1689 169009 482621   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 839421 1642691 0.511 0.611288   
## trend 23291 4116 5.658 4.95e-07 \*\*\*  
## season2 -134830 152491 -0.884 0.380247   
## season3 38792 152745 0.254 0.800419   
## season4 -615165 152666 -4.029 0.000165 \*\*\*  
## season5 -625294 152554 -4.099 0.000131 \*\*\*  
## season6 -884504 152617 -5.796 2.95e-07 \*\*\*  
## season7 -1082577 152748 -7.087 2.09e-09 \*\*\*  
## season8 -723603 152794 -4.736 1.45e-05 \*\*\*  
## season9 706895 152908 4.623 2.16e-05 \*\*\*  
## season10 308319 153364 2.010 0.049049 \*   
## season11 485176 154001 3.150 0.002578 \*\*   
## season12 85919 154027 0.558 0.579115   
## AEPI 14065 17159 0.820 0.415759   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 264000 on 58 degrees of freedom  
## Multiple R-squared: 0.9182, Adjusted R-squared: 0.8999   
## F-statistic: 50.09 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8999

#### PPIEtel:

summary(ModelWithPPIEtel)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + PPIEtel)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -670282 -185589 19856 172554 468929   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 593668 1640506 0.362 0.718756   
## trend 25282 1919 13.172 < 2e-16 \*\*\*  
## season2 -122617 152330 -0.805 0.424141   
## season3 53107 152246 0.349 0.728486   
## season4 -603022 152264 -3.960 0.000207 \*\*\*  
## season5 -614727 152459 -4.032 0.000163 \*\*\*  
## season6 -872851 152619 -5.719 3.94e-07 \*\*\*  
## season7 -1073314 152456 -7.040 2.51e-09 \*\*\*  
## season8 -711389 153051 -4.648 1.98e-05 \*\*\*  
## season9 707996 152568 4.641 2.03e-05 \*\*\*  
## season10 307412 152867 2.011 0.048984 \*   
## season11 479843 153028 3.136 0.002692 \*\*   
## season12 80433 153124 0.525 0.601390   
## PPIEtel 15872 16347 0.971 0.335606   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263400 on 58 degrees of freedom  
## Multiple R-squared: 0.9186, Adjusted R-squared: 0.9003   
## F-statistic: 50.34 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9003

#### NationalHolidays:

Indicator with the best adjusted R². The months remain very significant and the indicator itself has a p-value of 0,00636\*\*

summary(ModelWithNationalHolidays)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + NationalHolidays)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -555545 -153976 4 150487 404837   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2182435 110867 19.685 < 2e-16 \*\*\*  
## trend 26427 1431 18.465 < 2e-16 \*\*\*  
## season2 -131168 143696 -0.913 0.36512   
## season3 190430 152432 1.249 0.21658   
## season4 -321411 176034 -1.826 0.07302 .   
## season5 -623539 143803 -4.336 5.86e-05 \*\*\*  
## season6 -883072 143867 -6.138 8.06e-08 \*\*\*  
## season7 -1079124 143945 -7.497 4.29e-10 \*\*\*  
## season8 -724693 144037 -5.031 5.02e-06 \*\*\*  
## season9 705716 144144 4.896 8.18e-06 \*\*\*  
## season10 300019 144265 2.080 0.04199 \*   
## season11 472099 144400 3.269 0.00182 \*\*   
## season12 505461 210051 2.406 0.01932 \*   
## NationalHolidays -431536 152405 -2.832 0.00636 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 248900 on 58 degrees of freedom  
## Multiple R-squared: 0.9273, Adjusted R-squared: 0.911   
## F-statistic: 56.92 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9110

#### ChulwalarIndex:

summary(ModelWithChulwalarIndex)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + ChulwalarIndex)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -689635 -153608 9444 166039 495113   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.013e+06 2.262e+05 8.898 1.96e-12 \*\*\*  
## trend 2.506e+04 2.176e+03 11.515 < 2e-16 \*\*\*  
## season2 -1.295e+05 1.523e+05 -0.850 0.398630   
## season3 4.684e+04 1.523e+05 0.308 0.759534   
## season4 -6.157e+05 1.525e+05 -4.036 0.000161 \*\*\*  
## season5 -6.281e+05 1.525e+05 -4.119 0.000122 \*\*\*  
## season6 -8.809e+05 1.525e+05 -5.776 3.18e-07 \*\*\*  
## season7 -1.082e+06 1.526e+05 -7.092 2.05e-09 \*\*\*  
## season8 -7.182e+05 1.528e+05 -4.699 1.65e-05 \*\*\*  
## season9 7.115e+05 1.529e+05 4.653 1.95e-05 \*\*\*  
## season10 3.049e+05 1.530e+05 1.993 0.050965 .   
## season11 4.779e+05 1.532e+05 3.120 0.002817 \*\*   
## season12 7.433e+04 1.532e+05 0.485 0.629364   
## ChulwalarIndex 3.339e+01 3.805e+01 0.878 0.383723   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263800 on 58 degrees of freedom  
## Multiple R-squared: 0.9184, Adjusted R-squared: 0.9001   
## F-statistic: 50.18 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9001

#### Inflation:

summary(ModelWithInflation)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + Inflation)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -694867 -148205 9248 156635 501218   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2160745 132862 16.263 < 2e-16 \*\*\*  
## trend 26414 1526 17.313 < 2e-16 \*\*\*  
## season2 -131511 153141 -0.859 0.394009   
## season3 45633 153184 0.298 0.766848   
## season4 -607707 153249 -3.966 0.000204 \*\*\*  
## season5 -623065 153258 -4.065 0.000146 \*\*\*  
## season6 -882807 153322 -5.758 3.41e-07 \*\*\*  
## season7 -1078758 153407 -7.032 2.59e-09 \*\*\*  
## season8 -724536 153503 -4.720 1.53e-05 \*\*\*  
## season9 706375 153627 4.598 2.36e-05 \*\*\*  
## season10 301603 153808 1.961 0.054698 .   
## season11 474428 154026 3.080 0.003160 \*\*   
## season12 76824 154261 0.498 0.620359   
## Inflation 13335 37358 0.357 0.722422   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 265200 on 58 degrees of freedom  
## Multiple R-squared: 0.9174, Adjusted R-squared: 0.8989   
## F-statistic: 49.58 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8989

#### IndependenceDayPresents:

summary(ModelWithIndependenceDayPresents)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + IndependenceDayPresents)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -704113 -161955 23265 169241 468613   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1925395 469903 4.097 0.000131 \*\*\*  
## trend 25706 1986 12.944 < 2e-16 \*\*\*  
## season2 -130448 152891 -0.853 0.397053   
## season3 48026 152930 0.314 0.754620   
## season4 -606940 152994 -3.967 0.000203 \*\*\*  
## season5 -620657 153084 -4.054 0.000152 \*\*\*  
## season6 -879470 153200 -5.741 3.63e-07 \*\*\*  
## season7 -1074801 153342 -7.009 2.83e-09 \*\*\*  
## season8 -719650 153509 -4.688 1.72e-05 \*\*\*  
## season9 711480 153702 4.629 2.12e-05 \*\*\*  
## season10 306503 153919 1.991 0.051162 .   
## season11 479303 154163 3.109 0.002907 \*\*   
## season12 81850 154431 0.530 0.598127   
## IndependenceDayPresents 1201 2125 0.565 0.574184   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 264800 on 58 degrees of freedom  
## Multiple R-squared: 0.9177, Adjusted R-squared: 0.8993   
## F-statistic: 49.76 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8993

#### InfluenceNationalHolidays:

Indicator with the best adjusted R². The months remain very significant and the indicator itself has a p-value of 0,00636\*\*

summary(ModelWithInfluenceNationalHolidays)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + InfluenceNationalHolidays)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -555545 -153976 4 150487 404837   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2182435 110867 19.685 < 2e-16 \*\*\*  
## trend 26427 1431 18.465 < 2e-16 \*\*\*  
## season2 -131168 143696 -0.913 0.36512   
## season3 190430 152432 1.249 0.21658   
## season4 -321411 176034 -1.826 0.07302 .   
## season5 -623539 143803 -4.336 5.86e-05 \*\*\*  
## season6 -883072 143867 -6.138 8.06e-08 \*\*\*  
## season7 -1079124 143945 -7.497 4.29e-10 \*\*\*  
## season8 -724693 144037 -5.031 5.02e-06 \*\*\*  
## season9 1137252 209773 5.421 1.20e-06 \*\*\*  
## season10 300019 144265 2.080 0.04199 \*   
## season11 903635 209949 4.304 6.53e-05 \*\*\*  
## season12 505461 210051 2.406 0.01932 \*   
## InfluenceNationalHolidays -431536 152405 -2.832 0.00636 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 248900 on 58 degrees of freedom  
## Multiple R-squared: 0.9273, Adjusted R-squared: 0.911   
## F-statistic: 56.92 on 13 and 58 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9110

### 5.2 Models

#### ModelWithHighCorrelatingIndicators:

In this model only the indicators that correlate well with eachother have been used.

summary(ModelWithHighCorrelatingIndicators)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + CEPI + SIExtern +   
## UrbanoExports + GlobalisationPartyMembers + AEPI)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -651383 -159842 14275 171424 489393   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.625e+06 1.240e+07 -0.373 0.71054   
## trend 1.446e+04 1.650e+04 0.876 0.38477   
## season2 -1.584e+05 1.724e+05 -0.919 0.36213   
## season3 7.086e+03 1.984e+05 0.036 0.97164   
## season4 -6.221e+05 1.862e+05 -3.341 0.00152 \*\*   
## season5 -6.417e+05 1.944e+05 -3.302 0.00171 \*\*   
## season6 -8.872e+05 1.983e+05 -4.473 4.01e-05 \*\*\*  
## season7 -1.088e+06 2.218e+05 -4.904 8.99e-06 \*\*\*  
## season8 -7.287e+05 2.260e+05 -3.225 0.00214 \*\*   
## season9 7.236e+05 2.261e+05 3.201 0.00230 \*\*   
## season10 3.199e+05 2.231e+05 1.434 0.15741   
## season11 4.997e+05 2.246e+05 2.225 0.03027 \*   
## season12 7.986e+04 2.585e+05 0.309 0.75853   
## CEPI 9.245e+04 1.672e+05 0.553 0.58252   
## SIExtern 2.378e+04 4.559e+04 0.522 0.60401   
## UrbanoExports 1.504e-01 5.104e-01 0.295 0.76934   
## GlobalisationPartyMembers 3.463e+00 2.546e+01 0.136 0.89233   
## AEPI -3.307e+04 5.992e+04 -0.552 0.58327   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 271600 on 54 degrees of freedom  
## Multiple R-squared: 0.9194, Adjusted R-squared: 0.8941   
## F-statistic: 36.25 on 17 and 54 DF, p-value: < 2.2e-16

# Adjusted R²: 0.8941

It can be seen that the addition of these indicators causes the seasonality to be weakened. The individual indicators are not very significant either. Is this a multicollinearity effect? Or is it just because we have chose irrelevant indicators? An experimental idea comes from the next section:

#### ModelWithLowCorrelatingIndicators:

In this model only the indicators that hardly correlate at all with eachother have been used.

summary(ModelWithLowCorrelatingIndicators)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + NationalHolidays +   
## UrbanoExports + GlobalisationPartyMembers)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -508755 -122676 7119 173089 403964   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.467e+06 1.517e+06 0.967 0.337647   
## trend 2.264e+04 9.148e+03 2.474 0.016399 \*   
## season2 -1.274e+05 1.450e+05 -0.878 0.383528   
## season3 1.980e+05 1.546e+05 1.281 0.205562   
## season4 -3.100e+05 1.794e+05 -1.728 0.089424 .   
## season5 -6.084e+05 1.493e+05 -4.075 0.000146 \*\*\*  
## season6 -8.641e+05 1.518e+05 -5.693 4.78e-07 \*\*\*  
## season7 -1.056e+06 1.548e+05 -6.824 6.75e-09 \*\*\*  
## season8 -6.982e+05 1.583e+05 -4.411 4.72e-05 \*\*\*  
## season9 7.360e+05 1.622e+05 4.538 3.05e-05 \*\*\*  
## season10 3.341e+05 1.665e+05 2.007 0.049635 \*   
## season11 5.100e+05 1.712e+05 2.979 0.004276 \*\*   
## season12 5.471e+05 2.338e+05 2.341 0.022838 \*   
## NationalHolidays -4.315e+05 1.535e+05 -2.811 0.006794 \*\*   
## UrbanoExports 1.622e-01 1.692e-01 0.959 0.341873   
## GlobalisationPartyMembers -4.032e+00 2.086e+01 -0.193 0.847464   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 250700 on 56 degrees of freedom  
## Multiple R-squared: 0.9288, Adjusted R-squared: 0.9097   
## F-statistic: 48.69 on 15 and 56 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9097

It can be seen that the addition of these indicators causes the seasonality to be weakened. The individual indicators are not very significant either. Thus we should continue with trend and *seasonality*; the comparison of 4.3 and 4.4 confirms this:

#### ModelWithTrendAndSeasonalityOnly:

summary(ModelWithTrendAndSeasonalityOnly)

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -699390 -154210 17753 150363 495430   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2182435 117276 18.609 < 2e-16 \*\*\*  
## trend 26427 1514 17.456 < 2e-16 \*\*\*  
## season2 -131168 152001 -0.863 0.391663   
## season3 46585 152024 0.306 0.760356   
## season4 -609102 152062 -4.006 0.000176 \*\*\*  
## season5 -623539 152114 -4.099 0.000129 \*\*\*  
## season6 -883072 152182 -5.803 2.74e-07 \*\*\*  
## season7 -1079124 152265 -7.087 1.93e-09 \*\*\*  
## season8 -724693 152363 -4.756 1.31e-05 \*\*\*  
## season9 705716 152476 4.628 2.07e-05 \*\*\*  
## season10 300019 152603 1.966 0.054009 .   
## season11 472099 152746 3.091 0.003045 \*\*   
## season12 73925 152903 0.483 0.630546   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263300 on 59 degrees of freedom  
## Multiple R-squared: 0.9173, Adjusted R-squared: 0.9004   
## F-statistic: 54.51 on 12 and 59 DF, p-value: < 2.2e-16

# Adjusted R²: 0.9004

This model remains one of the best when looking at total exports.

#### ModelWithoutTrendAndSeasonality:

summary(ModelWithoutTrendAndSeasonality)

##   
## Call:  
## tslm(formula = TotalAsIs ~ CEPI + SIExtern + UrbanoExports +   
## GlobalisationPartyMembers + AEPI)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1246553 -546934 -10272 433938 1304765   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.101e+07 1.024e+07 -2.052 0.0442 \*  
## CEPI 3.277e+05 1.591e+05 2.059 0.0434 \*  
## SIExtern 4.274e+04 9.598e+04 0.445 0.6575   
## UrbanoExports -7.051e-04 7.794e-01 -0.001 0.9993   
## GlobalisationPartyMembers 1.126e+01 3.341e+01 0.337 0.7372   
## AEPI -9.807e+04 9.917e+04 -0.989 0.3263   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 638100 on 66 degrees of freedom  
## Multiple R-squared: 0.4562, Adjusted R-squared: 0.415   
## F-statistic: 11.07 on 5 and 66 DF, p-value: 9.091e-08

# Adjusted R²: 0.415

#### ModelTotalEtel

The model for the etel segment, including both subcategories, work best with trend and seasonality. An attempt to improve the model by adding Temperature showed no improvement worth mentioning.

summary(ModelTotalEtel)

##   
## Call:  
## tslm(formula = TotalEtelAsIs ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -352676 -105634 5934 107814 481013   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1370632 81168 16.886 < 2e-16 \*\*\*  
## trend 8070 1048 7.702 1.75e-10 \*\*\*  
## season2 -101964 105202 -0.969 0.3364   
## season3 -128812 105218 -1.224 0.2257   
## season4 -506178 105244 -4.810 1.08e-05 \*\*\*  
## season5 -607122 105281 -5.767 3.14e-07 \*\*\*  
## season6 -751654 105327 -7.136 1.59e-09 \*\*\*  
## season7 -838360 105385 -7.955 6.51e-11 \*\*\*  
## season8 -631474 105452 -5.988 1.35e-07 \*\*\*  
## season9 592436 105531 5.614 5.60e-07 \*\*\*  
## season10 202397 105619 1.916 0.0602 .   
## season11 232807 105718 2.202 0.0316 \*   
## season12 8713 105827 0.082 0.9347   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 182200 on 59 degrees of freedom  
## Multiple R-squared: 0.8905, Adjusted R-squared: 0.8683   
## F-statistic: 40 on 12 and 59 DF, p-value: < 2.2e-16

# Adjusted R²: 0,8683

## 6.0 Forecast with Models

### 6.1 Shorten the Time Series in Order to Test the Forecasts

Shorten the exports data in the Time Series in order to be able to compare the produced forecasts with the As Is data.

source("Scripts/forecastModels.r")

The first part of the *forecastModels.r* script:

* Shortens the indicators by the same amount
* Seperates the As Is and Plan data for 2013 in order to be able to compare the forecast to this date
* Seperate the indicator data for 2013 and 2014 in order to use these in the forecasts. First as a vector and then as a time series.

### 6.2 Forecasting and Testing the Models

#### Forecast ModelWithHighCorrelatingIndicators:

Shorten ModelWithHighCorrelatingIndicators by one year in order to be able to produce a forecast for 2013.

summary(ModelWithHighCorrelatingIndicators\_2012)

##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + CEPI\_2012 +   
## SIExtern\_2012 + UrbanoExports\_2012 + GlobalisationPartyMembers\_2012 +   
## AEPI\_2012)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -590682 -148874 23944 148648 423243   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.542e+07 1.461e+07 -1.056 0.297224   
## trend 2.827e+02 2.096e+04 0.013 0.989301   
## season2 -1.071e+05 1.777e+05 -0.603 0.549946   
## season3 6.219e+04 2.031e+05 0.306 0.760993   
## season4 -7.186e+05 1.935e+05 -3.715 0.000595 \*\*\*  
## season5 -5.757e+05 1.978e+05 -2.910 0.005752 \*\*   
## season6 -8.241e+05 1.994e+05 -4.134 0.000167 \*\*\*  
## season7 -1.083e+06 2.186e+05 -4.955 1.23e-05 \*\*\*  
## season8 -6.963e+05 2.236e+05 -3.113 0.003325 \*\*   
## season9 6.649e+05 2.219e+05 2.996 0.004572 \*\*   
## season10 3.046e+05 2.223e+05 1.370 0.177909   
## season11 5.136e+05 2.230e+05 2.303 0.026314 \*   
## season12 4.974e+04 2.530e+05 0.197 0.845057   
## CEPI\_2012 2.248e+05 1.922e+05 1.169 0.248882   
## SIExtern\_2012 2.369e+04 4.590e+04 0.516 0.608451   
## UrbanoExports\_2012 -1.522e-01 5.208e-01 -0.292 0.771535   
## GlobalisationPartyMembers\_2012 3.142e+01 3.844e+01 0.817 0.418370   
## AEPI\_2012 -4.974e+04 6.236e+04 -0.798 0.429556   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 256600 on 42 degrees of freedom  
## Multiple R-squared: 0.9145, Adjusted R-squared: 0.8798   
## F-statistic: 26.41 on 17 and 42 DF, p-value: < 2.2e-16

Add "newdata" to the 2013 indicator values for the forecast

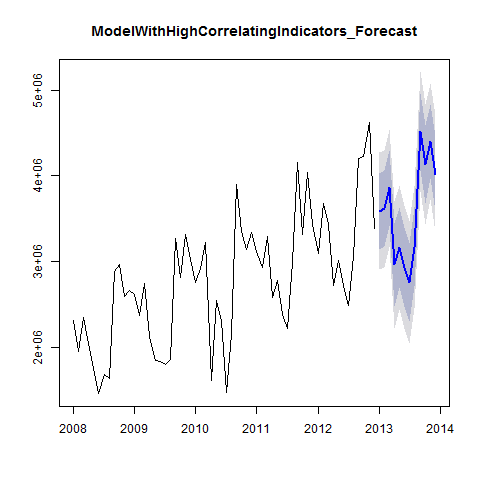


Figure 25

print(head(ModelWithHighCorrelatingIndicators\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + CEPI\_2012 +   
## SIExtern\_2012 + UrbanoExports\_2012 + GlobalisationPartyMembers\_2012 +   
## AEPI\_2012)  
##   
## Coefficients:  
## (Intercept) trend   
## -1.542e+07 2.827e+02   
## season2 season3   
## -1.071e+05 6.219e+04   
## season4 season5   
## -7.186e+05 -5.757e+05   
## season6 season7   
## -8.241e+05 -1.083e+06   
## season8 season9   
## -6.963e+05 6.649e+05   
## season10 season11   
## 3.046e+05 5.136e+05   
## season12 CEPI\_2012   
## 4.974e+04 2.248e+05   
## SIExtern\_2012 UrbanoExports\_2012   
## 2.369e+04 -1.522e-01   
## GlobalisationPartyMembers\_2012 AEPI\_2012   
## 3.142e+01 -4.974e+04   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3588409 3615751 3857685 2959830 3163014 2934791 2755659 3167751  
## Sep Oct Nov Dec  
## 2013 4512206 4129962 4398869 4019999  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3145307 2901625  
## Feb 2013 3174780 2932271  
## Mar 2013 3418316 3176687  
## Apr 2013 2478742 2214171  
## May 2013 2698603 2443204  
## Jun 2013 2472811 2218748  
## Jul 2013 2296827 2044495  
## Aug 2013 2707972 2455119  
## Sep 2013 4061044 3812930  
## Oct 2013 3677062 3427993  
## Nov 2013 3963060 3723389  
## Dec 2013 3580235 3338390  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4031511 4275193  
## Feb 2013 4056722 4299231  
## Mar 2013 4297055 4538684  
## Apr 2013 3440918 3705489  
## May 2013 3627424 3882824  
## Jun 2013 3396771 3650835  
## Jul 2013 3214490 3466822  
## Aug 2013 3627531 3880384  
## Sep 2013 4963367 5211481  
## Oct 2013 4582861 4831931  
## Nov 2013 4834678 5074349  
## Dec 2013 4459762 4701608  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673

In order to be able to correlate the Forecast with the As Is Data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data.

print(cor(ModelWithHighCorrelatingIndicators\_PointForecast, TotalAsIs\_2013))

## [1] 0.9028604

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

A Comparison with linear regression also supports the result.

print(summary(ModelWithHighCorrelatingIndicators\_forecast\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ ModelWithHighCorrelatingIndicators\_PointForecast,   
## data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -540546 -227283 12596 157487 731430   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -5.082e+05 6.545e+05  
## ModelWithHighCorrelatingIndicators\_PointForecast 1.195e+00 1.799e-01  
## t value Pr(>|t|)   
## (Intercept) -0.776 0.455   
## ModelWithHighCorrelatingIndicators\_PointForecast 6.641 5.78e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 357000 on 10 degrees of freedom  
## Multiple R-squared: 0.8152, Adjusted R-squared: 0.7967   
## F-statistic: 44.1 on 1 and 10 DF, p-value: 5.776e-05

print(summary(TotalAsIs\_2013\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ TotalPlan\_2013, data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -441885 -227385 -43470 184761 466401   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.972e+04 4.930e+05 -0.182 0.859   
## TotalPlan\_2013 1.053e+00 1.318e-01 7.987 1.2e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 305700 on 10 degrees of freedom  
## Multiple R-squared: 0.8645, Adjusted R-squared: 0.8509   
## F-statistic: 63.78 on 1 and 10 DF, p-value: 1.195e-05

Forecast ModelWithLowCorrelatingIndicators Shorten ModelWithLowCorrelatingIndicators by one year in order to be able to produce a forecast for 2013.

print(summary(ModelWithLowCorrelatingIndicators\_2012))

##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + NationalHolidays\_2012 +   
## UrbanoExports\_2012 + GlobalisationPartyMembers\_2012)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -508808 -130098 11746 177748 466017   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.211e+06 1.549e+06 0.782 0.438428   
## trend 2.004e+04 1.028e+04 1.950 0.057515 .   
## season2 -2.898e+04 1.573e+05 -0.184 0.854714   
## season3 2.520e+05 1.631e+05 1.545 0.129609   
## season4 -3.800e+05 2.242e+05 -1.695 0.097100 .   
## season5 -4.697e+05 1.623e+05 -2.894 0.005897 \*\*   
## season6 -7.350e+05 1.652e+05 -4.450 5.79e-05 \*\*\*  
## season7 -9.668e+05 1.687e+05 -5.732 8.35e-07 \*\*\*  
## season8 -5.809e+05 1.727e+05 -3.364 0.001602 \*\*   
## season9 7.426e+05 1.772e+05 4.190 0.000132 \*\*\*  
## season10 3.765e+05 1.822e+05 2.066 0.044712 \*   
## season11 5.612e+05 1.876e+05 2.991 0.004541 \*\*   
## season12 4.716e+05 2.756e+05 1.711 0.094062 .   
## NationalHolidays\_2012 -3.051e+05 1.962e+05 -1.554 0.127234   
## UrbanoExports\_2012 1.218e-01 1.838e-01 0.663 0.510937   
## GlobalisationPartyMembers\_2012 5.692e+00 2.753e+01 0.207 0.837165   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 248200 on 44 degrees of freedom  
## Multiple R-squared: 0.9162, Adjusted R-squared: 0.8876   
## F-statistic: 32.05 on 15 and 44 DF, p-value: < 2.2e-16

Add "newdata" to the 2013 indicator values for the forecast

print(head(ModelWithLowCorrelatingIndicators\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + NationalHolidays\_2012 +   
## UrbanoExports\_2012 + GlobalisationPartyMembers\_2012)  
##   
## Coefficients:  
## (Intercept) trend   
## 1.211e+06 2.004e+04   
## season2 season3   
## -2.898e+04 2.520e+05   
## season4 season5   
## -3.800e+05 -4.697e+05   
## season6 season7   
## -7.350e+05 -9.668e+05   
## season8 season9   
## -5.809e+05 7.426e+05   
## season10 season11   
## 3.765e+05 5.612e+05   
## season12 NationalHolidays\_2012   
## 4.716e+05 -3.051e+05   
## UrbanoExports\_2012 GlobalisationPartyMembers\_2012   
## 1.218e-01 5.692e+00   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3703214 3694280 3690209 3383361 3313734 3068413 2856693 3262648  
## Sep Oct Nov Dec  
## 2013 4606119 4260121 4464894 4090225  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3306827 3089217  
## Feb 2013 3297893 3080283  
## Mar 2013 3244288 2999485  
## Apr 2013 2937440 2692637  
## May 2013 2917347 2699736  
## Jun 2013 2672026 2454415  
## Jul 2013 2460306 2242696  
## Aug 2013 2866261 2648651  
## Sep 2013 4209732 3992121  
## Oct 2013 3863734 3646123  
## Nov 2013 4068507 3850897  
## Dec 2013 3693837 3476227  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4099602 4317212  
## Feb 2013 4090668 4308278  
## Mar 2013 4136130 4380933  
## Apr 2013 3829282 4074085  
## May 2013 3710121 3927731  
## Jun 2013 3464800 3682410  
## Jul 2013 3253081 3470691  
## Aug 2013 3659036 3876646  
## Sep 2013 5002506 5220117  
## Oct 2013 4656508 4874119  
## Nov 2013 4861282 5078892  
## Dec 2013 4486612 4704222  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673

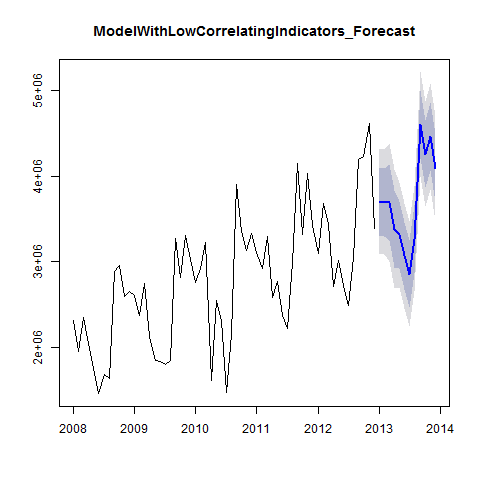


Figure 26

In order to be ableCorrelation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data. to correlate the Forecast with the As Is Data, it is necessary to convert the Point Estimator into a time series.

print(cor(ModelWithLowCorrelatingIndicators\_PointForecast, TotalAsIs\_2013))

## [1] 0.9590162

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

A Comparison with linear regression also supports the result.

print(summary(ModelWithLowCorrelatingIndicators\_forecast\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ ModelWithLowCorrelatingIndicators\_PointForecast,   
## data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -299026 -155463 -40768 115237 406333   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -1.252e+06 4.754e+05  
## ModelWithLowCorrelatingIndicators\_PointForecast 1.361e+00 1.272e-01  
## t value Pr(>|t|)   
## (Intercept) -2.633 0.025 \*   
## ModelWithLowCorrelatingIndicators\_PointForecast 10.703 8.5e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 235300 on 10 degrees of freedom  
## Multiple R-squared: 0.9197, Adjusted R-squared: 0.9117   
## F-statistic: 114.6 on 1 and 10 DF, p-value: 8.5e-07

print(summary(TotalAsIs\_2013\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ TotalPlan\_2013, data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -441885 -227385 -43470 184761 466401   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.972e+04 4.930e+05 -0.182 0.859   
## TotalPlan\_2013 1.053e+00 1.318e-01 7.987 1.2e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 305700 on 10 degrees of freedom  
## Multiple R-squared: 0.8645, Adjusted R-squared: 0.8509   
## F-statistic: 63.78 on 1 and 10 DF, p-value: 1.195e-05

#### Forecast ModelWithTrendAndSeasonalityOnly:

Shorten ModelWithTrendAndSeasonalityOnly by one year in order to be able to produce a forecast for 2013.

print(summary(ModelWithTrendAndSeasonalityOnly\_2012))

##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -600304 -116717 -7864 163111 473692   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2147793 120530 17.820 < 2e-16 \*\*\*  
## trend 25212 1887 13.363 < 2e-16 \*\*\*  
## season2 -34146 156872 -0.218 0.828632   
## season3 180612 156906 1.151 0.255518   
## season4 -639529 156962 -4.074 0.000176 \*\*\*  
## season5 -490327 157042 -3.122 0.003068 \*\*   
## season6 -760860 157144 -4.842 1.43e-05 \*\*\*  
## season7 -997792 157268 -6.345 8.09e-08 \*\*\*  
## season8 -617048 157415 -3.920 0.000286 \*\*\*  
## season9 701211 157585 4.450 5.26e-05 \*\*\*  
## season10 330001 157777 2.092 0.041907 \*   
## season11 509563 157991 3.225 0.002292 \*\*   
## season12 109681 158227 0.693 0.491603   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 248000 on 47 degrees of freedom  
## Multiple R-squared: 0.9106, Adjusted R-squared: 0.8878   
## F-statistic: 39.89 on 12 and 47 DF, p-value: < 2.2e-16

Add "newdata" to the 2013 indicator values for the forecast.

print(head(ModelWithTrendAndSeasonalityOnly\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season)  
##   
## Coefficients:  
## (Intercept) trend season2 season3 season4   
## 2147793 25212 -34146 180612 -639529   
## season5 season6 season7 season8 season9   
## -490327 -760860 -997791 -617048 701211   
## season10 season11 season12   
## 330001 509563 109681   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3685709 3676775 3916745 3121815 3296229 3050908 2839188 3245143  
## Sep Oct Nov Dec  
## 2013 4588614 4242616 4447389 4072720  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3321691 3122318  
## Feb 2013 3312757 3113384  
## Mar 2013 3552727 3353354  
## Apr 2013 2757797 2558424  
## May 2013 2932211 2732838  
## Jun 2013 2686890 2487517  
## Jul 2013 2475170 2275797  
## Aug 2013 2881125 2681752  
## Sep 2013 4224596 4025223  
## Oct 2013 3878598 3679225  
## Nov 2013 4083371 3883998  
## Dec 2013 3708702 3509329  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4049727 4249100  
## Feb 2013 4040793 4240166  
## Mar 2013 4280763 4480136  
## Apr 2013 3485833 3685206  
## May 2013 3660247 3859620  
## Jun 2013 3414926 3614299  
## Jul 2013 3203206 3402579  
## Aug 2013 3609161 3808534  
## Sep 2013 4952632 5152005  
## Oct 2013 4606634 4806007  
## Nov 2013 4811407 5010780  
## Dec 2013 4436737 4636110  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673

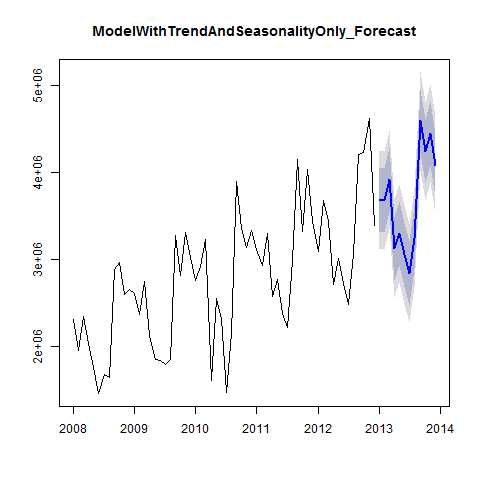


Figure 27

In order to be able to correlate the Forecast with the As Is Data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data.

print(cor(ModelWithTrendAndSeasonalityOnly\_PointForecast, TotalAsIs\_2013) )

## [1] 0.9138049

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

A Comparison with linear regression also supports the result

print(summary(ModelWithTrendAndSeasonalityOnly\_Forecast\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ ModelWithTrendAndSeasonalityOnly\_PointForecast,   
## data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -516239 -216450 33683 123007 675607   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -8.142e+05 6.536e+05  
## ModelWithTrendAndSeasonalityOnly\_PointForecast 1.249e+00 1.755e-01  
## t value Pr(>|t|)   
## (Intercept) -1.246 0.241   
## ModelWithTrendAndSeasonalityOnly\_PointForecast 7.115 3.24e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 337300 on 10 degrees of freedom  
## Multiple R-squared: 0.835, Adjusted R-squared: 0.8185   
## F-statistic: 50.62 on 1 and 10 DF, p-value: 3.238e-05

print(summary(TotalAsIs\_2013\_lm))

##   
## Call:  
## lm(formula = TotalAsIs\_2013 ~ TotalPlan\_2013, data = TotalAsIs\_2013)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -441885 -227385 -43470 184761 466401   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.972e+04 4.930e+05 -0.182 0.859   
## TotalPlan\_2013 1.053e+00 1.318e-01 7.987 1.2e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 305700 on 10 degrees of freedom  
## Multiple R-squared: 0.8645, Adjusted R-squared: 0.8509   
## F-statistic: 63.78 on 1 and 10 DF, p-value: 1.195e-05

#### Forecast ModelTotalEtel

Shorten the variables in ModelTotalEtel by one year in order to be able to produce a forecast for 2013.

print(summary(ModelTotalEtel\_2012))

##   
## Call:  
## tslm(formula = TotalEtelAsIs\_2012 ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -299816 -89175 -2539 108720 287047   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1361585 78100 17.434 < 2e-16 \*\*\*  
## trend 6460 1223 5.284 3.20e-06 \*\*\*  
## season2 -44872 101648 -0.441 0.66091   
## season3 -53014 101671 -0.521 0.60452   
## season4 -524858 101707 -5.160 4.88e-06 \*\*\*  
## season5 -501638 101759 -4.930 1.07e-05 \*\*\*  
## season6 -674802 101825 -6.627 3.01e-08 \*\*\*  
## season7 -765417 101905 -7.511 1.38e-09 \*\*\*  
## season8 -544231 102001 -5.336 2.68e-06 \*\*\*  
## season9 558436 102111 5.469 1.70e-06 \*\*\*  
## season10 232677 102235 2.276 0.02745 \*   
## season11 276582 102374 2.702 0.00957 \*\*   
## season12 55396 102527 0.540 0.59154   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 160700 on 47 degrees of freedom  
## Multiple R-squared: 0.9008, Adjusted R-squared: 0.8755   
## F-statistic: 35.58 on 12 and 47 DF, p-value: < 2.2e-16

#### Forecast:

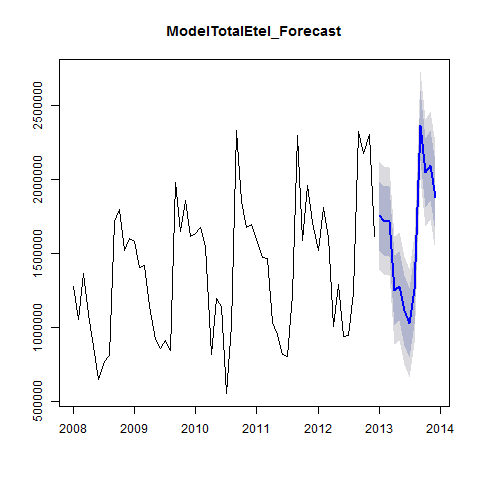


Figure 28

print(head(ModelTotalEtel\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalEtelAsIs\_2012 ~ trend + season)  
##   
## Coefficients:  
## (Intercept) trend season2 season3 season4   
## 1361585 6460 -44872 -53014 -524858   
## season5 season6 season7 season8 season9   
## -501638 -674802 -765417 -544231 558436   
## season10 season11 season12   
## 232677 276582 55396   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 1755615 1717202 1715520 1250135 1279815 1113110 1028955 1256600  
## Sep Oct Nov Dec  
## 2013 2365726 2046428 2096791 1882065  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 1519741.4 1390553.3  
## Feb 2013 1481328.6 1352140.5  
## Mar 2013 1479646.6 1350458.5  
## Apr 2013 1014262.0 885073.9  
## May 2013 1043941.8 914753.7  
## Jun 2013 877236.8 748048.7  
## Jul 2013 793081.8 663893.7  
## Aug 2013 1020727.0 891538.9  
## Sep 2013 2129853.0 2000664.9  
## Oct 2013 1810554.4 1681366.3  
## Nov 2013 1860918.0 1731729.9  
## Dec 2013 1646192.0 1517003.9  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 1991488 2120676  
## Feb 2013 1953076 2082264  
## Mar 2013 1951394 2080582  
## Apr 2013 1486009 1615197  
## May 2013 1515689 1644877  
## Jun 2013 1348984 1478172  
## Jul 2013 1264829 1394017  
## Aug 2013 1492474 1621662  
## Sep 2013 2601600 2730788  
## Oct 2013 2282301 2411489  
## Nov 2013 2332665 2461853  
## Dec 2013 2117939 2247127  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 1279668 1053325 1367520 1090725 873568 644479 772658 806741  
## 2009 1583216 1407388 1420801 1141100 919860 858876 910134 843050  
## 2010 1637464 1676161 1549560 813469 1198401 1140024 551268 1012542  
## 2011 1595267 1473528 1469728 1034650 952553 819303 802076 1222812  
## 2012 1519748 1812897 1607280 1008022 1291983 940158 945929 1235146  
## Sep Oct Nov Dec  
## 2008 1715265 1795751 1518288 1601324  
## 2009 1981563 1647934 1857836 1615091  
## 2010 2335488 1856264 1678123 1699063  
## 2011 2303271 1591584 1960675 1713991  
## 2012 2330334 2177895 2306324 1618147

In order to be able to correlate the Forecast with the As Is Data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data.

print(cor(ModelTotalEtel\_PointForecast, TotalEtelAsIs\_2013))

## [1] 0.9392717

print(cor(TotalEtelPlan\_2013, TotalEtelAsIs\_2013))

## [1] 0.9602983

#### Forecast ModelWithTotalUrbanoExports

Shorten the variables in ModelWithTotalUrbanoExports by one year in order to be able to produce a forecast for 2013.

print(summary(ModelWithTotalUrbanoExports\_2012))

##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + UrbanoExports\_2012)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -569149 -128266 8067 181935 457990   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.450e+06 1.039e+06 1.396 0.169437   
## trend 2.187e+04 5.296e+03 4.129 0.000152 \*\*\*  
## season2 -3.080e+04 1.579e+05 -0.195 0.846161   
## season3 1.873e+05 1.581e+05 1.184 0.242307   
## season4 -6.295e+05 1.586e+05 -3.970 0.000250 \*\*\*  
## season5 -4.770e+05 1.592e+05 -2.996 0.004395 \*\*   
## season6 -7.441e+05 1.600e+05 -4.651 2.80e-05 \*\*\*  
## season7 -9.777e+05 1.609e+05 -6.075 2.23e-07 \*\*\*  
## season8 -5.936e+05 1.621e+05 -3.663 0.000643 \*\*\*  
## season9 7.280e+05 1.634e+05 4.456 5.31e-05 \*\*\*  
## season10 3.601e+05 1.648e+05 2.185 0.034032 \*   
## season11 5.430e+05 1.664e+05 3.263 0.002084 \*\*   
## season12 1.465e+05 1.682e+05 0.871 0.388352   
## UrbanoExports\_2012 1.246e-01 1.843e-01 0.676 0.502194   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 249500 on 46 degrees of freedom  
## Multiple R-squared: 0.9115, Adjusted R-squared: 0.8864   
## F-statistic: 36.43 on 13 and 46 DF, p-value: < 2.2e-16

Add "newdata" to the 2013 indicator values for the forecast.

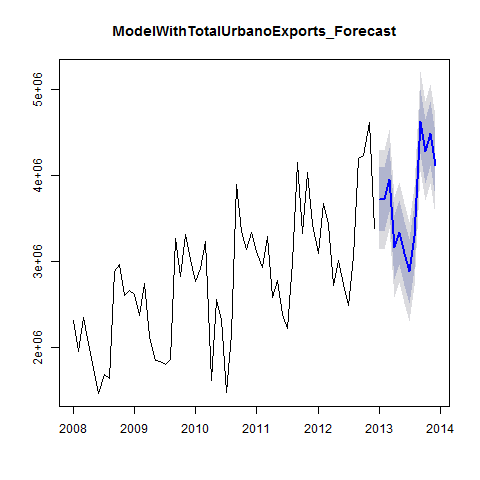


Figure 29

print(head(ModelWithTotalUrbanoExports\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + UrbanoExports\_2012)  
##   
## Coefficients:  
## (Intercept) trend season2   
## 1.450e+06 2.187e+04 -3.080e+04   
## season3 season4 season5   
## 1.873e+05 -6.295e+05 -4.770e+05   
## season6 season7 season8   
## -7.441e+05 -9.777e+05 -5.936e+05   
## season9 season10 season11   
## 7.280e+05 3.601e+05 5.430e+05   
## season12 UrbanoExports\_2012   
## 1.465e+05 1.246e-01   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3724840 3715906 3955876 3160946 3335360 3090039 2878319 3284274  
## Sep Oct Nov Dec  
## 2013 4627745 4281747 4486520 4111851  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3350944 3146008  
## Feb 2013 3342010 3137074  
## Mar 2013 3581980 3377044  
## Apr 2013 2787050 2582114  
## May 2013 2961464 2756527  
## Jun 2013 2716143 2511206  
## Jul 2013 2504423 2299487  
## Aug 2013 2910378 2705442  
## Sep 2013 4253849 4048913  
## Oct 2013 3907851 3702915  
## Nov 2013 4112624 3907688  
## Dec 2013 3737954 3533018  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4098736 4303672  
## Feb 2013 4089802 4294739  
## Mar 2013 4329772 4534708  
## Apr 2013 3534842 3739778  
## May 2013 3709256 3914192  
## Jun 2013 3463935 3668871  
## Jul 2013 3252215 3457151  
## Aug 2013 3658170 3863106  
## Sep 2013 5001641 5206577  
## Oct 2013 4655643 4860579  
## Nov 2013 4860416 5065352  
## Dec 2013 4485747 4690683  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673

In order to be able to correlate the Forecast with the As Is data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data.

print(cor(ModelWithTotalUrbanoExports\_PointForecast, TotalAsIs\_2013) )

## [1] 0.9138049

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

#### Forecast ModelWithNationalHolidays

Shorten the variables in ModelWithNationalHolidays by one year in order to be able to produce a forecast for 2013.

print(summary(ModelWithNationalHolidays\_2012))

##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + NationalHolidays\_2012)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -539294 -116717 -7864 163111 473692   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2147793 118654 18.101 < 2e-16 \*\*\*  
## trend 25212 1857 13.574 < 2e-16 \*\*\*  
## season2 -34146 154431 -0.221 0.825988   
## season3 241623 159215 1.518 0.135962   
## season4 -395488 218453 -1.810 0.076768 .   
## season5 -490327 154598 -3.172 0.002699 \*\*   
## season6 -760860 154698 -4.918 1.16e-05 \*\*\*  
## season7 -997792 154821 -6.445 6.22e-08 \*\*\*  
## season8 -617048 154966 -3.982 0.000241 \*\*\*  
## season9 701211 155133 4.520 4.31e-05 \*\*\*  
## season10 330001 155322 2.125 0.039022 \*   
## season11 509563 155532 3.276 0.002005 \*\*   
## season12 414732 248034 1.672 0.101299   
## NationalHolidays\_2012 -305051 193024 -1.580 0.120873   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 244200 on 46 degrees of freedom  
## Multiple R-squared: 0.9152, Adjusted R-squared: 0.8912   
## F-statistic: 38.19 on 13 and 46 DF, p-value: < 2.2e-16

Add "newdata" to the 2013 indicator values for the forecast.

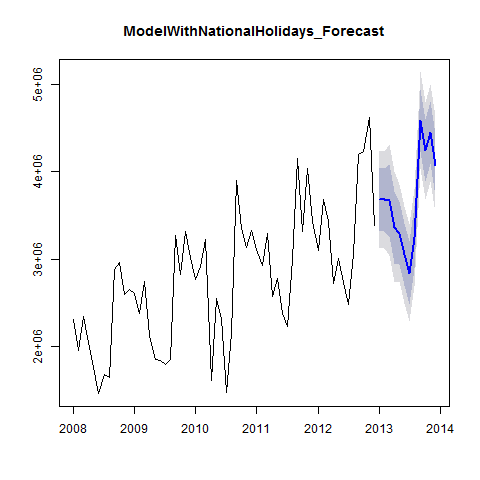


Figure 30

print(head(ModelWithNationalHolidays\_Forecast))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + NationalHolidays\_2012)  
##   
## Coefficients:  
## (Intercept) trend season2   
## 2147793 25212 -34146   
## season3 season4 season5   
## 241623 -395488 -490327   
## season6 season7 season8   
## -760860 -997791 -617048   
## season9 season10 season11   
## 701211 330001 509563   
## season12 NationalHolidays\_2012   
## 414732 -305051   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3685709 3676775 3672704 3365856 3296229 3050908 2839188 3245143  
## Sep Oct Nov Dec  
## 2013 4588614 4242616 4447389 4072720  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3327245 3130767  
## Feb 2013 3318311 3121833  
## Mar 2013 3261840 3036641  
## Apr 2013 2954992 2729793  
## May 2013 2937764 2741286  
## Jun 2013 2692443 2495965  
## Jul 2013 2480724 2284246  
## Aug 2013 2886679 2690201  
## Sep 2013 4230149 4033671  
## Oct 2013 3884151 3687673  
## Nov 2013 4088925 3892447  
## Dec 2013 3714255 3517777  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4044174 4240652  
## Feb 2013 4035240 4231718  
## Mar 2013 4083568 4308767  
## Apr 2013 3776720 4001919  
## May 2013 3654693 3851171  
## Jun 2013 3409372 3605850  
## Jul 2013 3197653 3394131  
## Aug 2013 3603608 3800086  
## Sep 2013 4947079 5143556  
## Oct 2013 4601081 4797558  
## Nov 2013 4805854 5002332  
## Dec 2013 4431184 4627662  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673

In order to be able to correlate the Forecast with the As Is data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is data for 2013 with the Plan Data.

print(cor(ModelWithNationalHolidays\_PointForecast, TotalAsIs\_2013) )

## [1] 0.9590162

#### Forecast ModelWithInfluenceNationalHolidays

Shorten the variables in ModelWithInfluenceNationalHolidays by one year in order to be able to produce a forecast for 2013. { r ModelWithInfluenceNationalHolidays\_2012} print(summary(ModelWithInfluenceNationalHolidays\_2012))

Add "newdata" to the 2013 indicator values for the forecast.

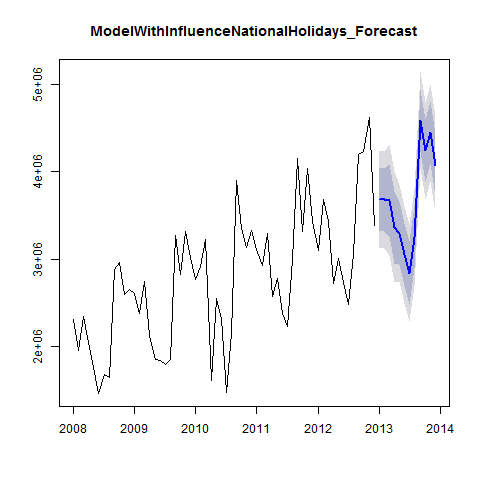


Figure 31

print(head(ModelWithInfluenceNationalHolidays\_Forecast, 10))

## $model  
##   
## Call:  
## tslm(formula = TotalAsIs\_2012 ~ trend + season + InfluenceNationalHolidays\_2012)  
##   
## Coefficients:  
## (Intercept) trend   
## 2147793 25212   
## season2 season3   
## -34146 241623   
## season4 season5   
## -395488 -490327   
## season6 season7   
## -760860 -997791   
## season8 season9   
## -617048 1006262   
## season10 season11   
## 330001 814614   
## season12 InfluenceNationalHolidays\_2012   
## 414732 -305051   
##   
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2013 3685709 3676775 3672704 3365856 3296229 3050908 2839188 3245143  
## Sep Oct Nov Dec  
## 2013 4588614 4242616 4447389 4072720  
##   
## $lower  
## [,1] [,2]  
## Jan 2013 3327245 3130767  
## Feb 2013 3318311 3121833  
## Mar 2013 3261840 3036641  
## Apr 2013 2954992 2729793  
## May 2013 2937764 2741286  
## Jun 2013 2692443 2495965  
## Jul 2013 2480724 2284246  
## Aug 2013 2886679 2690201  
## Sep 2013 4230149 4033671  
## Oct 2013 3884151 3687673  
## Nov 2013 4088925 3892447  
## Dec 2013 3714255 3517777  
##   
## $upper  
## [,1] [,2]  
## Jan 2013 4044174 4240652  
## Feb 2013 4035240 4231718  
## Mar 2013 4083568 4308767  
## Apr 2013 3776720 4001919  
## May 2013 3654693 3851171  
## Jun 2013 3409372 3605850  
## Jul 2013 3197653 3394131  
## Aug 2013 3603608 3800086  
## Sep 2013 4947079 5143556  
## Oct 2013 4601081 4797558  
## Nov 2013 4805854 5002332  
## Dec 2013 4431184 4627662  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2313221 1950131 2346635 2039787 1756964 1458302 1679637 1639670  
## 2009 2610573 2371327 2743786 2125308 1850073 1836222 1797311 1851968  
## 2010 2760688 2918333 3227041 1613888 2550157 2317645 1474144 2148521  
## 2011 3112861 2926663 3294784 2577079 2774068 2378227 2222900 2991787  
## 2012 3093088 3679308 3433364 2714899 3011767 2726028 2483834 3055655  
## Sep Oct Nov Dec  
## 2008 2882886 2959716 2596494 2656568  
## 2009 3271171 2818888 3310776 3022513  
## 2010 3898571 3348953 3135945 3332886  
## 2011 4151531 3318684 4037076 3429843  
## 2012 4200796 4228724 4618540 3383673  
##   
## $method  
## [1] "Linear regression model"  
##   
## $newdata  
## InfluenceNationalHolidays\_2012  
## 1 0  
## 2 0  
## 3 1  
## 4 0  
## 5 0  
## 6 0  
## 7 0  
## 8 0  
## 9 1  
## 10 0  
## 11 1  
## 12 1  
##   
## $residuals  
## Jan Feb Mar Apr May  
## 2008 140216.750 -213939.450 186635.850 186635.850 -26559.850  
## 2009 135027.775 -95284.425 -23805.250 274667.000 -235991.825  
## 2010 -17398.200 149180.600 156908.775 -539293.975 161551.200  
## 2011 32233.825 -145030.375 -77889.200 121356.050 82921.225  
## 2012 -290080.150 305073.650 -241850.175 -43364.925 18079.250  
## Jun Jul Aug Sep Oct  
## 2008 -79900.850 353153.750 -92768.250 -193023.050 229804.950  
## 2009 -4521.825 168286.775 -183011.225 -107279.025 -213564.025  
## 2010 174360.200 -457421.200 -188999.200 217580.000 13960.000  
## 2011 -67598.775 -11206.175 351725.825 167999.025 -318849.975  
## 2012 -22338.750 -52813.150 113052.850 -85276.950 288649.050  
## Nov Dec  
## 2008 -338190.250 96553.350  
## 2009 73550.775 159957.375  
## 2010 -403821.200 167789.400  
## 2011 194768.825 -37794.575  
## 2012 473691.850 -386505.550  
##   
## $fitted  
## Jan Feb Mar Apr May Jun Jul Aug  
## 2008 2173004 2164070 2159999 1853151 1783524 1538203 1326483 1732438  
## 2009 2475545 2466611 2767591 1850641 2086065 1840744 1629024 2034979  
## 2010 2778086 2769152 3070132 2153182 2388606 2143285 1931565 2337520  
## 2011 3080627 3071693 3372673 2455723 2691147 2445826 2234106 2640061  
## 2012 3383168 3374234 3675214 2758264 2993688 2748367 2536647 2942602  
## Sep Oct Nov Dec  
## 2008 3075909 2729911 2934684 2560015  
## 2009 3378450 3032452 3237225 2862556  
## 2010 3680991 3334993 3539766 3165097  
## 2011 3983532 3637534 3842307 3467638  
## 2012 4286073 3940075 4144848 3770179

In order to be able to correlate the Forecast with the As Is Data, it is necessary to convert the Point Estimator into a time series. Correlation of the forecasts and As Is Data. As a comparison, the correlation of the As Is Data for 2013 with the Plan Data.

print(cor(ModelWithInfluenceNationalHolidays\_PointForecast, TotalAsIs\_2013))

## [1] 0.9590162

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

print(cor(TotalAsIs\_2013, TotalPlan\_2013))

## [1] 0.929769

## 7.0 Forecast for 2014

The script *forecast2014.r* procuces a 2014 export forecast and alternate forecast. It also outputs both forecasts as CSV files. As ModelWithLowCorrelatingIndicators was the one of best fitting model for a forecast, the exports data for 2014 will be forecast based on trend and seasonality and NationalHolidays, UrbanoExports and GlobalisationPartyMembers.

source("Scripts/forecast2014.r")

print(summary(ModelWithLowCorrelatingIndicators))

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + NationalHolidays +   
## UrbanoExports + GlobalisationPartyMembers)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -508755 -122676 7119 173089 403964   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.467e+06 1.517e+06 0.967 0.337647   
## trend 2.264e+04 9.148e+03 2.474 0.016399 \*   
## season2 -1.274e+05 1.450e+05 -0.878 0.383528   
## season3 1.980e+05 1.546e+05 1.281 0.205562   
## season4 -3.100e+05 1.794e+05 -1.728 0.089424 .   
## season5 -6.084e+05 1.493e+05 -4.075 0.000146 \*\*\*  
## season6 -8.641e+05 1.518e+05 -5.693 4.78e-07 \*\*\*  
## season7 -1.056e+06 1.548e+05 -6.824 6.75e-09 \*\*\*  
## season8 -6.982e+05 1.583e+05 -4.411 4.72e-05 \*\*\*  
## season9 7.360e+05 1.622e+05 4.538 3.05e-05 \*\*\*  
## season10 3.341e+05 1.665e+05 2.007 0.049635 \*   
## season11 5.100e+05 1.712e+05 2.979 0.004276 \*\*   
## season12 5.471e+05 2.338e+05 2.341 0.022838 \*   
## NationalHolidays -4.315e+05 1.535e+05 -2.811 0.006794 \*\*   
## UrbanoExports 1.622e-01 1.692e-01 0.959 0.341873   
## GlobalisationPartyMembers -4.032e+00 2.086e+01 -0.193 0.847464   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 250700 on 56 degrees of freedom  
## Multiple R-squared: 0.9288, Adjusted R-squared: 0.9097   
## F-statistic: 48.69 on 15 and 56 DF, p-value: < 2.2e-16

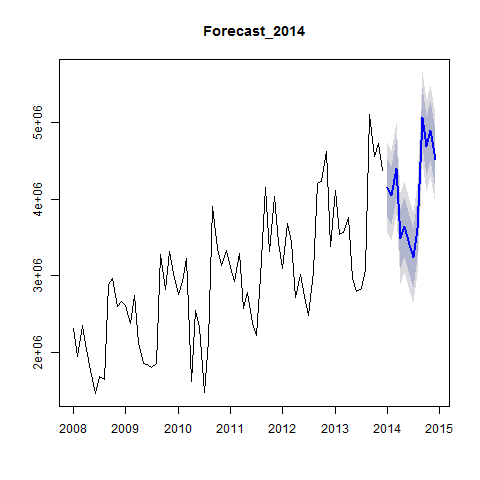


Figure 32

print(PointForecast\_ModelWithLowCorrelatingIndicators\_2014)

## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 4154873 4050131 4398156 3481206 3637040 3403934 3234308 3615166  
## Sep Oct Nov Dec  
## 2014 5072002 4692732 4891237 4519491

print(cor(TotalAsIs\_2014,TotalPlan\_2014))

## [1] 0.9448221

print(cor(TotalAsIs\_2014,PointForecast\_ModelWithLowCorrelatingIndicators\_2014))

## [1] 0.9178468

As ModelWithTrendAndSeasonalityOnly also gave a well fitting model for a forecast, the exports data for 2014 will be forecast based on trend and seasonality.

print(summary(ModelWithTrendAndSeasonalityOnly) )

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -699390 -154210 17753 150363 495430   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2182435 117276 18.609 < 2e-16 \*\*\*  
## trend 26427 1514 17.456 < 2e-16 \*\*\*  
## season2 -131168 152001 -0.863 0.391663   
## season3 46585 152024 0.306 0.760356   
## season4 -609102 152062 -4.006 0.000176 \*\*\*  
## season5 -623539 152114 -4.099 0.000129 \*\*\*  
## season6 -883072 152182 -5.803 2.74e-07 \*\*\*  
## season7 -1079124 152265 -7.087 1.93e-09 \*\*\*  
## season8 -724693 152363 -4.756 1.31e-05 \*\*\*  
## season9 705716 152476 4.628 2.07e-05 \*\*\*  
## season10 300019 152603 1.966 0.054009 .   
## season11 472099 152746 3.091 0.003045 \*\*   
## season12 73925 152903 0.483 0.630546   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 263300 on 59 degrees of freedom  
## Multiple R-squared: 0.9173, Adjusted R-squared: 0.9004   
## F-statistic: 54.51 on 12 and 59 DF, p-value: < 2.2e-16

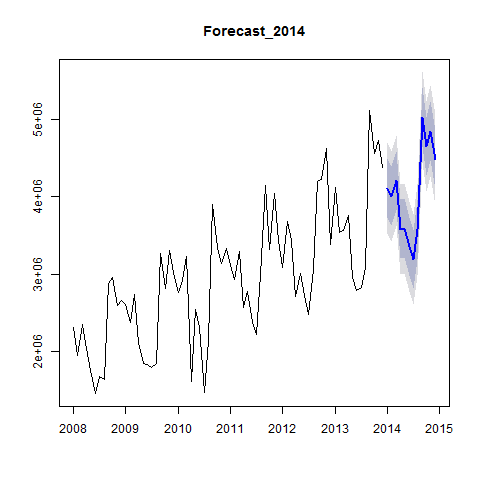


Figure 33

cor(TotalAsIs\_2014,TotalPlan\_2014)

## [1] 0.9448221

cor(TotalAsIs\_2014,PointForecast\_TrendAndSeasonality\_2014)

## [1] 0.9349765

#### Alternative

As the indiators NationalHolidays delievered a good result, but could not convince in the 2013 forecast,it could be possible that the data for 2013 was to blame. Therefore there is another Forecast using the ModelWithNationalHolidays

print(summary(ModelWithNationalHolidays))

##   
## Call:  
## tslm(formula = TotalAsIs ~ trend + season + NationalHolidays)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -555545 -153976 4 150487 404837   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2182435 110867 19.685 < 2e-16 \*\*\*  
## trend 26427 1431 18.465 < 2e-16 \*\*\*  
## season2 -131168 143696 -0.913 0.36512   
## season3 190430 152432 1.249 0.21658   
## season4 -321411 176034 -1.826 0.07302 .   
## season5 -623539 143803 -4.336 5.86e-05 \*\*\*  
## season6 -883072 143867 -6.138 8.06e-08 \*\*\*  
## season7 -1079124 143945 -7.497 4.29e-10 \*\*\*  
## season8 -724693 144037 -5.031 5.02e-06 \*\*\*  
## season9 705716 144144 4.896 8.18e-06 \*\*\*  
## season10 300019 144265 2.080 0.04199 \*   
## season11 472099 144400 3.269 0.00182 \*\*   
## season12 505461 210051 2.406 0.01932 \*   
## NationalHolidays -431536 152405 -2.832 0.00636 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 248900 on 58 degrees of freedom  
## Multiple R-squared: 0.9273, Adjusted R-squared: 0.911   
## F-statistic: 56.92 on 13 and 58 DF, p-value: < 2.2e-16

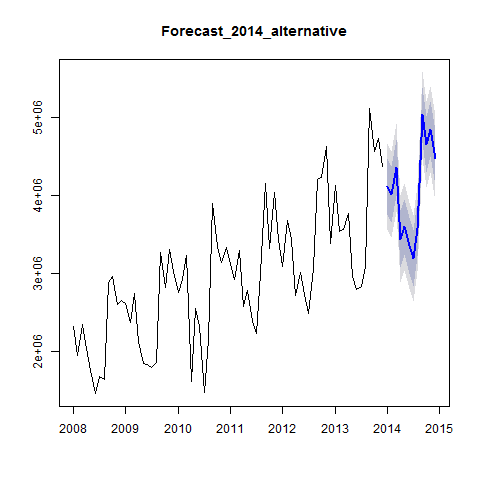


Figure 34

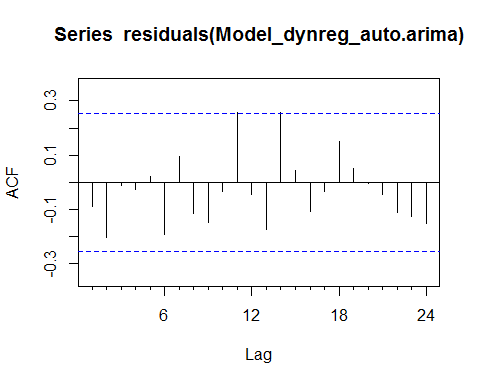
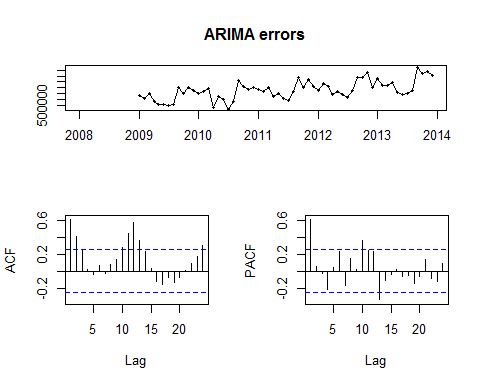
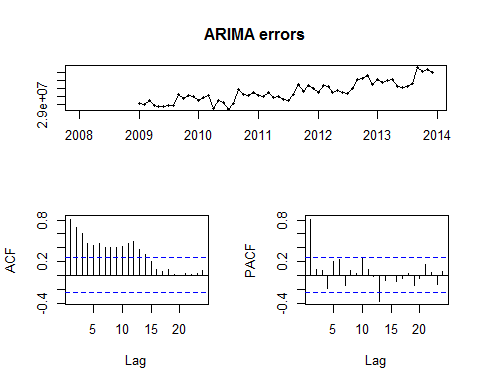
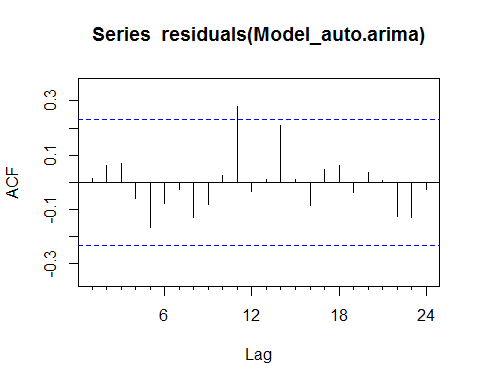
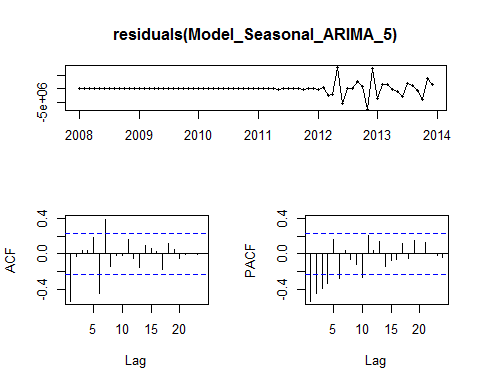
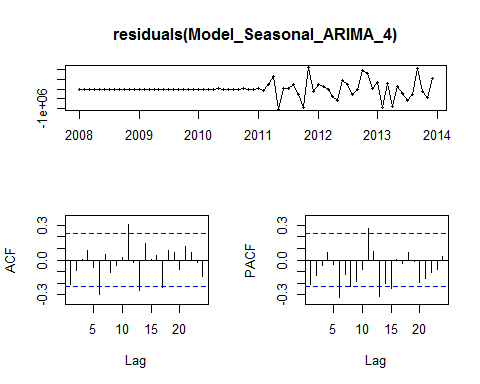
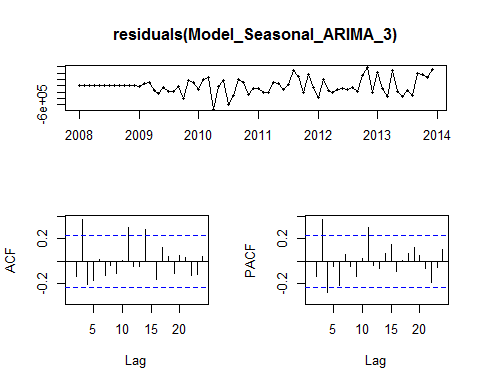
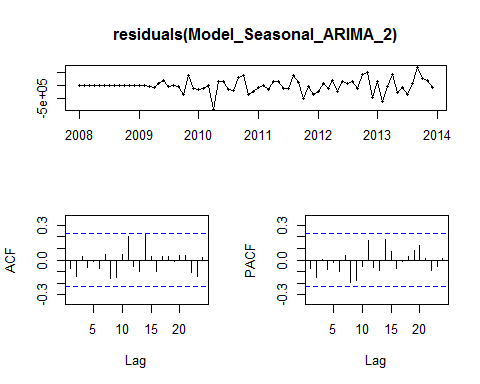
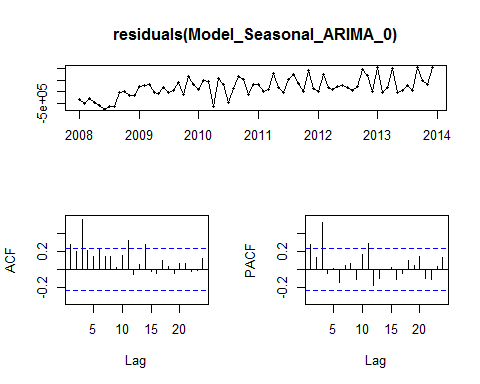
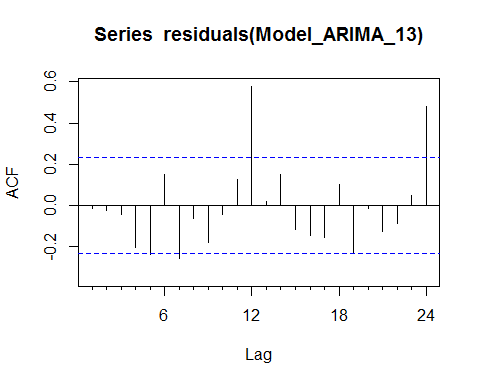
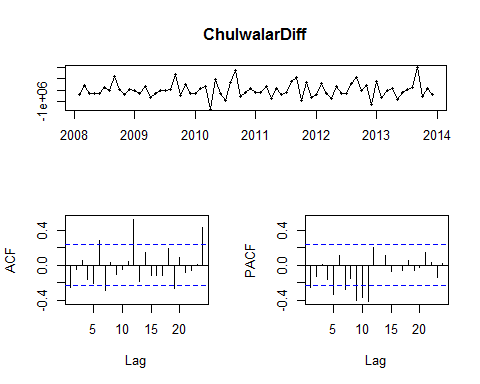
print(head(PointForecast\_2014\_alternative))

## [1] 4111576 4006834 4354859 3437909 3593744 3360637

## 8.0 Developing Forecasting Models with Alternative Model Approaches

### 8.1 Exponential Smoothing

source("Scripts/alternativeForecastModels.r")



Exponential Smoothing uses past values to calculate a forecast. The strength with which each value influences the forecast is weakened with help of a smoothing parameter. Thus we are dealing with a weighted average, whose values fade out the longer ago they were in the past.

#### Simple expontential smoothing

Formula: ses(). It must be decided if alpha (the smoothing parameter should be automatically calculated. If initial=simple, the alpha value canbe set to any chosen value, if initial=optimal (or nothing, as this is the default), alpha will be set to the optimal value based on ets(). h=12 gives the number of cycles for the forecast.

print(summary(Model\_ses))

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(x = TotalAsIs, h = 12)   
##   
## Smoothing parameters:  
## alpha = 0.671   
##   
## Initial states:  
## l = 2173226.7433   
##   
## sigma: 609507  
##   
## AIC AICc BIC   
## 2230.058 2230.232 2234.612   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 47469.84 609507 429997.1 -1.511008 15.02336 1.172074  
## ACF1  
## Training set 0.02384493  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4466448 3685333 5247562 3271836 5661059  
## Feb 2014 4466448 3525801 5407094 3027853 5905042  
## Mar 2014 4466448 3389650 5543245 2819628 6113267  
## Apr 2014 4466448 3268880 5664015 2634926 6297969  
## May 2014 4466448 3159220 5773675 2467215 6465680  
## Jun 2014 4466448 3058072 5874823 2312524 6620371  
## Jul 2014 4466448 2963718 5969177 2168221 6764674  
## Aug 2014 4466448 2874947 6057948 2032458 6900437  
## Sep 2014 4466448 2790873 6142022 1903878 7029017  
## Oct 2014 4466448 2710821 6222074 1781448 7151447  
## Nov 2014 4466448 2634263 6298632 1664363 7268532  
## Dec 2014 4466448 2560778 6372117 1551977 7380918  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4466448 3685333 5247562 3271836 5661059  
## Feb 2014 4466448 3525801 5407094 3027853 5905042  
## Mar 2014 4466448 3389650 5543245 2819628 6113267  
## Apr 2014 4466448 3268880 5664015 2634926 6297969  
## May 2014 4466448 3159220 5773675 2467215 6465680  
## Jun 2014 4466448 3058072 5874823 2312524 6620371  
## Jul 2014 4466448 2963718 5969177 2168221 6764674  
## Aug 2014 4466448 2874947 6057948 2032458 6900437  
## Sep 2014 4466448 2790873 6142022 1903878 7029017  
## Oct 2014 4466448 2710821 6222074 1781448 7151447  
## Nov 2014 4466448 2634263 6298632 1664363 7268532  
## Dec 2014 4466448 2560778 6372117 1551977 7380918

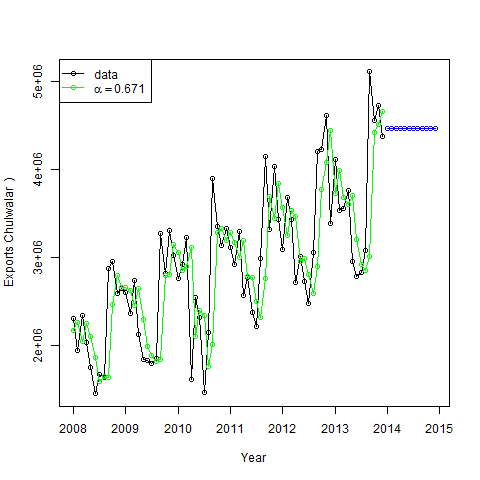


Figure 35

The Akaike's Information Criterion(AIC/AICc) or the Bayesian Information Criterion (BIC) should be at minimum.

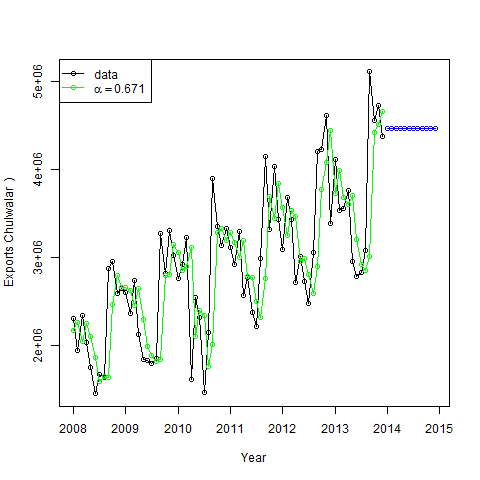


Figure 36

##### Holt's linear trend method:

Holt added to the model in order to forecast using trends as well. For this it is necessary to add a beta, which determines the trend. If neither alpha nor beta is stated, both parameters will be optimised using ets().

print(summary(Model\_holt\_1))

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(x = TotalAsIs, h = 12)   
##   
## Smoothing parameters:  
## alpha = 0.6571   
## beta = 1e-04   
##   
## Initial states:  
## l = 2040390.7764   
## b = 45050.7514   
##   
## sigma: 608119.1  
##   
## AIC AICc BIC   
## 2233.730 2234.327 2242.837   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -16586.9 608119.1 441110.7 -3.88925 15.75307 1.202367  
## ACF1  
## Training set 0.03462672  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4536367 3757031 5315703 3344475 5728259  
## Feb 2014 4581298 3648703 5513894 3155016 6007580  
## Mar 2014 4626230 3562188 5690271 2998918 6253541  
## Apr 2014 4671161 3490181 5852141 2865008 6477314  
## May 2014 4716092 3428721 6003463 2747228 6684956  
## Jun 2014 4761024 3375378 6146669 2641862 6880185  
## Jul 2014 4805955 3328531 6283379 2546429 7065480  
## Aug 2014 4850886 3287035 6414738 2459182 7242591  
## Sep 2014 4895818 3250047 6541588 2378829 7412807  
## Oct 2014 4940749 3216925 6664573 2304387 7577111  
## Nov 2014 4985680 3187164 6784196 2235088 7736273  
## Dec 2014 5030612 3160363 6900860 2170314 7890909  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4536367 3757031 5315703 3344475 5728259  
## Feb 2014 4581298 3648703 5513894 3155016 6007580  
## Mar 2014 4626230 3562188 5690271 2998918 6253541  
## Apr 2014 4671161 3490181 5852141 2865008 6477314  
## May 2014 4716092 3428721 6003463 2747228 6684956  
## Jun 2014 4761024 3375378 6146669 2641862 6880185  
## Jul 2014 4805955 3328531 6283379 2546429 7065480  
## Aug 2014 4850886 3287035 6414738 2459182 7242591  
## Sep 2014 4895818 3250047 6541588 2378829 7412807  
## Oct 2014 4940749 3216925 6664573 2304387 7577111  
## Nov 2014 4985680 3187164 6784196 2235088 7736273  
## Dec 2014 5030612 3160363 6900860 2170314 7890909

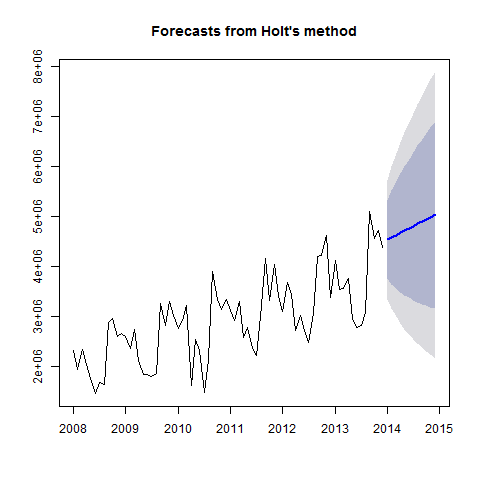


Figure 37

The trend is exponential if the intercepts(level) and the gradient (slope) are multiplied with eachother. The values are worse. As the Beta was very low in the optimisation, the forecast is very similar to the ses() model.

print(summary(Model\_holt\_2))

##   
## Forecast method: Holt's method with exponential trend  
##   
## Model Information:  
## Holt's method with exponential trend   
##   
## Call:  
## holt(x = TotalAsIs, h = 12, exponential = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.6637   
## beta = 1e-04   
##   
## Initial states:  
## l = 2041538.9468   
## b = 1.0029   
##   
## sigma: 0.2438  
##   
## AIC AICc BIC   
## 2251.010 2251.607 2260.116   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 37825.61 609787.5 433018.9 -1.838214 15.18487 1.180311  
## ACF1  
## Training set 0.02918287  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4488281 3088115 5925250 2420273 6707636  
## Feb 2014 4502175 2873726 6345456 2148120 7359507  
## Mar 2014 4516113 2689016 6591169 2014450 7948409  
## Apr 2014 4530094 2554728 6879901 1834292 8440387  
## May 2014 4544118 2356797 7090057 1706424 9163399  
## Jun 2014 4558186 2258191 7396963 1537299 9692519  
## Jul 2014 4572297 2132702 7604930 1426237 10161853  
## Aug 2014 4586452 2024565 7767881 1347217 10834671  
## Sep 2014 4600650 1977137 7830548 1268811 11238423  
## Oct 2014 4614893 1865592 8091651 1221065 11791332  
## Nov 2014 4629180 1809284 8264089 1184871 12138552  
## Dec 2014 4643510 1734926 8367264 1088540 12321867  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4488281 3088115 5925250 2420273 6707636  
## Feb 2014 4502175 2873726 6345456 2148120 7359507  
## Mar 2014 4516113 2689016 6591169 2014450 7948409  
## Apr 2014 4530094 2554728 6879901 1834292 8440387  
## May 2014 4544118 2356797 7090057 1706424 9163399  
## Jun 2014 4558186 2258191 7396963 1537299 9692519  
## Jul 2014 4572297 2132702 7604930 1426237 10161853  
## Aug 2014 4586452 2024565 7767881 1347217 10834671  
## Sep 2014 4600650 1977137 7830548 1268811 11238423  
## Oct 2014 4614893 1865592 8091651 1221065 11791332  
## Nov 2014 4629180 1809284 8264089 1184871 12138552  
## Dec 2014 4643510 1734926 8367264 1088540 12321867

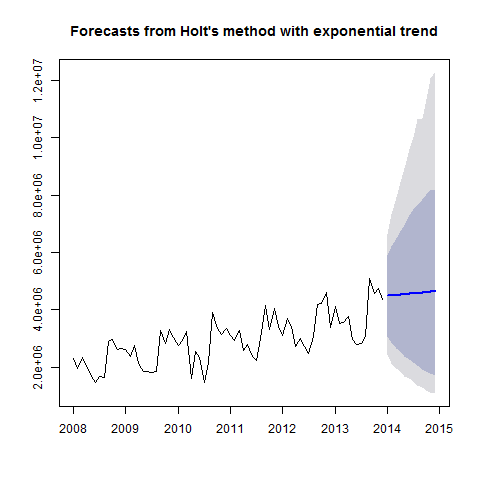


Figure 38

As such simple trends tend to forecast the future to positively, we have added a dampener. Similar values to that of Model\_holt\_1

summary(Model\_holt\_3)

##   
## Forecast method: Damped Holt's method  
##   
## Model Information:  
## Damped Holt's method   
##   
## Call:  
## holt(x = TotalAsIs, h = 12, damped = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.6613   
## beta = 2e-04   
## phi = 0.98   
##   
## Initial states:  
## l = 2040392.5761   
## b = 45053.25   
##   
## sigma: 608787.2  
##   
## AIC AICc BIC   
## 2235.888 2236.797 2247.272   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 15578.94 608787.2 436909.7 -2.797612 15.46526 1.190916  
## ACF1  
## Training set 0.03351419  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4483618 3703426 5263811 3290417 5676819  
## Feb 2014 4493914 3558436 5429391 3063224 5924603  
## Mar 2014 4504003 3435520 5572486 2869899 6138107  
## Apr 2014 4513891 3327168 5700614 2698955 6328827  
## May 2014 4523581 3229332 5817829 2544198 6502963  
## Jun 2014 4533077 3139534 5926619 2401837 6664316  
## Jul 2014 4542383 3056128 6028638 2269352 6815413  
## Aug 2014 4551503 2977955 6125051 2144969 6958036  
## Sep 2014 4560440 2904162 6216719 2027381 7093499  
## Oct 2014 4569199 2834101 6304298 1915595 7222803  
## Nov 2014 4577783 2767264 6388301 1808834 7346732  
## Dec 2014 4586195 2703249 6469141 1706477 7465913

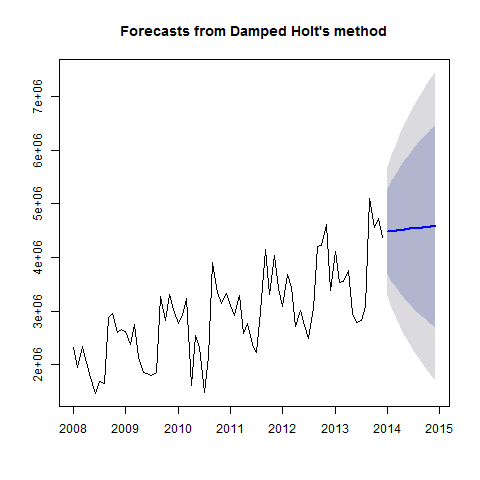


Figure 39

This also works for exponential trends. The values remain worse.

print(summary(Model\_holt\_4))

##   
## Forecast method: Damped Holt's method with exponential trend  
##   
## Model Information:  
## Damped Holt's method with exponential trend   
##   
## Call:  
## holt(x = TotalAsIs, h = 12, damped = TRUE, exponential = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.6679   
## beta = 1e-04   
## phi = 0.9799   
##   
## Initial states:  
## l = 2041541.9705   
## b = 1.0019   
##   
## sigma: 0.2449  
##   
## AIC AICc BIC   
## 2253.216 2254.125 2264.600   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 46119.56 609906.7 432069.1 -1.549114 15.11987 1.177722  
## ACF1  
## Training set 0.0254941  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4470648 3093312 5915594 2391206 6598493  
## Feb 2014 4473164 2827055 6262884 2169066 7305926  
## Mar 2014 4475630 2640821 6554810 1888594 7977975  
## Apr 2014 4478047 2457711 6732160 1808000 8295808  
## May 2014 4480418 2313808 7054261 1652860 8975078  
## Jun 2014 4482742 2248118 7192296 1553133 9507856  
## Jul 2014 4485020 2108341 7320856 1440358 9891682  
## Aug 2014 4487253 1965887 7501251 1377621 10271249  
## Sep 2014 4489443 1929469 7715232 1295809 10571906  
## Oct 2014 4491589 1858922 7921688 1237930 11417652  
## Nov 2014 4493694 1786954 8094435 1137834 11756944  
## Dec 2014 4495757 1702482 8159257 1079300 12355368  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4470648 3093312 5915594 2391206 6598493  
## Feb 2014 4473164 2827055 6262884 2169066 7305926  
## Mar 2014 4475630 2640821 6554810 1888594 7977975  
## Apr 2014 4478047 2457711 6732160 1808000 8295808  
## May 2014 4480418 2313808 7054261 1652860 8975078  
## Jun 2014 4482742 2248118 7192296 1553133 9507856  
## Jul 2014 4485020 2108341 7320856 1440358 9891682  
## Aug 2014 4487253 1965887 7501251 1377621 10271249  
## Sep 2014 4489443 1929469 7715232 1295809 10571906  
## Oct 2014 4491589 1858922 7921688 1237930 11417652  
## Nov 2014 4493694 1786954 8094435 1137834 11756944  
## Dec 2014 4495757 1702482 8159257 1079300 12355368

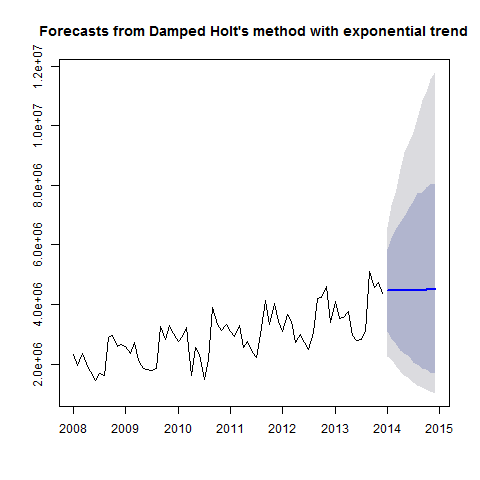


Figure 40

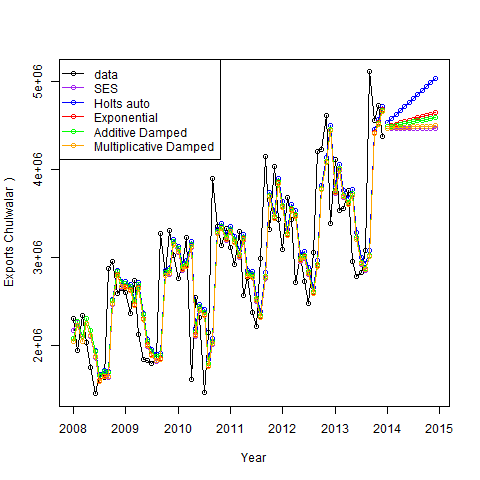


Figure 41

As these forecasts are not very convincing at the moment, there is no need to export the data.

#### Holt-Winter's Seasonal Method

Holt and Winters have expanded Holt's model further to include the seasonality aspect. The parameter gamma, which is for smoothing the seasonality, was added to achieve this. The values are better than the models without seasonality. This logical matches our results from the regression approaches, the data is strongly influenced by seasonality. In the following model, none of the parameters are given so that they will be optimised automatically. There are two models: one using an additive error model method and one using a multiplicative error model.

print(summary(Model\_hw\_1))

##   
## Forecast method: Holt-Winters' additive method  
##   
## Model Information:  
## Holt-Winters' additive method   
##   
## Call:  
## hw(x = TotalAsIs, h = 12, seasonal = "additive")   
##   
## Smoothing parameters:  
## alpha = 0.0087   
## beta = 0.0087   
## gamma = 1e-04   
##   
## Initial states:  
## l = 2047375.0884   
## b = 22509.7631   
## s=259168.3 654942.6 474529.8 876025.2 -475155 -852844  
## -664662.5 -412596.7 -438677.3 273215 138077.9 167976.7  
##   
## sigma: 241685  
##   
## AIC AICc BIC   
## 2124.856 2134.747 2161.283   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 21615.43 241685 202218.5 -0.08252109 7.329458 0.5512016  
## ACF1  
## Training set -0.2819072  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4141204 3831472 4450936 3667510 4614898  
## Feb 2014 4147309 3837472 4457147 3673453 4621165  
## Mar 2014 4318537 4008512 4628563 3844394 4792680  
## Apr 2014 3642744 3332425 3953063 3168153 4117335  
## May 2014 3704865 3394124 4015605 3229628 4180102  
## Jun 2014 3488859 3177546 3800173 3012746 3964973  
## Jul 2014 3336738 3024677 3648799 2859482 3813994  
## Aug 2014 3750478 3437474 4063482 3271780 4229176  
## Sep 2014 5137771 4823607 5451935 4657298 5618244  
## Oct 2014 4772337 4456775 5087900 4289726 5254949  
## Nov 2014 4988809 4671591 5306028 4503665 5473953  
## Dec 2014 4629097 4309943 4948252 4140992 5117202  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4141204 3831472 4450936 3667510 4614898  
## Feb 2014 4147309 3837472 4457147 3673453 4621165  
## Mar 2014 4318537 4008512 4628563 3844394 4792680  
## Apr 2014 3642744 3332425 3953063 3168153 4117335  
## May 2014 3704865 3394124 4015605 3229628 4180102  
## Jun 2014 3488859 3177546 3800173 3012746 3964973  
## Jul 2014 3336738 3024677 3648799 2859482 3813994  
## Aug 2014 3750478 3437474 4063482 3271780 4229176  
## Sep 2014 5137771 4823607 5451935 4657298 5618244  
## Oct 2014 4772337 4456775 5087900 4289726 5254949  
## Nov 2014 4988809 4671591 5306028 4503665 5473953  
## Dec 2014 4629097 4309943 4948252 4140992 5117202

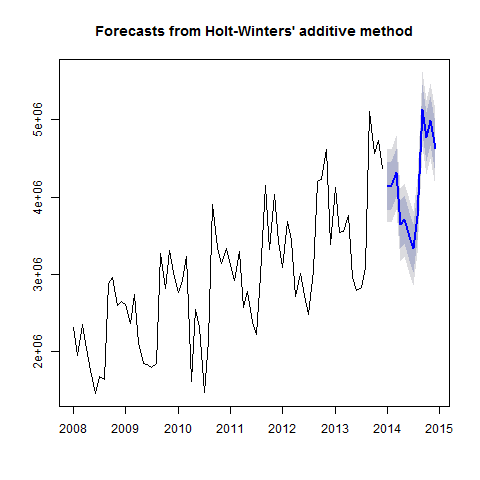


Figure 42

# AIC AICc BIC   
#2127.984 2137.875 2164.411

print(summary(Model\_hw\_2))

##   
## Forecast method: Holt-Winters' multiplicative method  
##   
## Model Information:  
## Holt-Winters' multiplicative method   
##   
## Call:  
## hw(x = TotalAsIs, h = 12, seasonal = "multiplicative")   
##   
## Smoothing parameters:  
## alpha = 0.025   
## beta = 0.0062   
## gamma = 1e-04   
##   
## Initial states:  
## l = 2026247.531   
## b = 25395.1259   
## s=1.0933 1.232 1.1763 1.3086 0.8384 0.699  
## 0.7653 0.8502 0.8596 1.0793 1.0316 1.0665  
##   
## sigma: 0.0877  
##   
## AIC AICc BIC   
## 2128.303 2138.194 2164.729   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 17434.11 235296.6 191805.3 -0.3292809 7.213472 0.5228175  
## ACF1  
## Training set -0.3514421  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4226941 3751624 4702258 3500006 4953876  
## Feb 2014 4123665 3659738 4587591 3414151 4833179  
## Mar 2014 4350808 3860995 4840620 3601704 5099911  
## Apr 2014 3494208 3100476 3887940 2892046 4096370  
## May 2014 3484738 3091618 3877858 2883513 4085963  
## Jun 2014 3162774 2805463 3520085 2616314 3709234  
## Jul 2014 2912399 2582802 3241996 2408324 3416474  
## Aug 2014 3521645 3122278 3921013 2910865 4132425  
## Sep 2014 5540988 4911109 6170867 4577671 6504304  
## Oct 2014 5020487 4448200 5592775 4145249 5895725  
## Nov 2014 5299729 4693715 5905743 4372911 6226547  
## Dec 2014 4740169 4196230 5284108 3908286 5572052  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4226941 3751624 4702258 3500006 4953876  
## Feb 2014 4123665 3659738 4587591 3414151 4833179  
## Mar 2014 4350808 3860995 4840620 3601704 5099911  
## Apr 2014 3494208 3100476 3887940 2892046 4096370  
## May 2014 3484738 3091618 3877858 2883513 4085963  
## Jun 2014 3162774 2805463 3520085 2616314 3709234  
## Jul 2014 2912399 2582802 3241996 2408324 3416474  
## Aug 2014 3521645 3122278 3921013 2910865 4132425  
## Sep 2014 5540988 4911109 6170867 4577671 6504304  
## Oct 2014 5020487 4448200 5592775 4145249 5895725  
## Nov 2014 5299729 4693715 5905743 4372911 6226547  
## Dec 2014 4740169 4196230 5284108 3908286 5572052

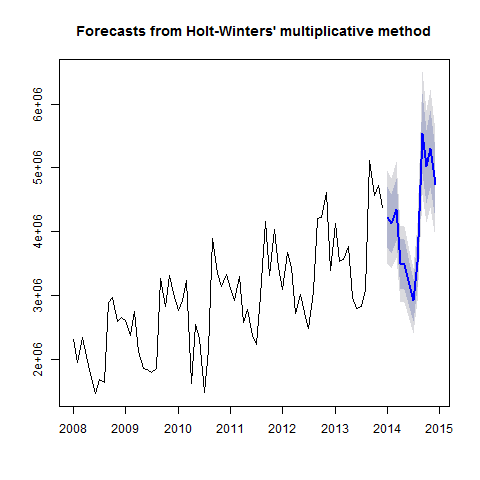


Figure 43

# AIC AICc BIC   
#2137.673 2147.564 2174.100

The additive model gives slightly better results than the multiplicative model.

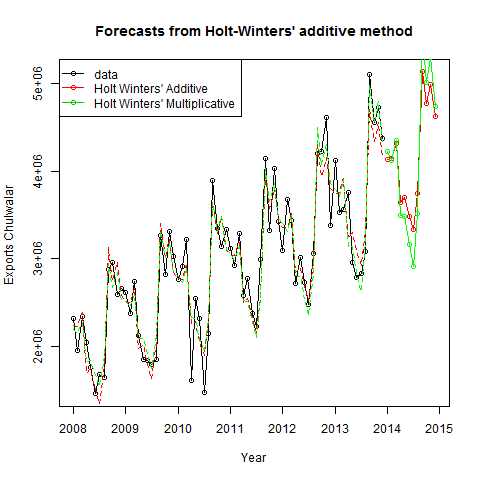


Figure 44

In order to use the results later, they need to be converted into point forcasts.

print(head(Model\_hw\_1\_PointForecast))

## [1] 4141204 4147309 4318537 3642744 3704865 3488859

print(head(Model\_hw\_2\_PointForecast))

## [1] 4226941 4123665 4350808 3494208 3484738 3162774

#### Innovations state space models for exponential smoothing

The funktion ets() produces a model with the same values as Model\_hw\_1. The reason for this is that all of the parameters in this model were optimised using the ets() function. The results are a ets(A,A,A) model which is anadditive method for trend, seasonality and errors. The previous models also showed the type of ets() model in their summary. In this case the user parameters were either accepted or rejected. As the model has been set to "ZZZ", the best model will be automatically chosen.

print(summary(Model\_ets))

## ETS(A,A,A)   
##   
## Call:  
## ets(y = TotalAsIs, model = "ZZZ", damped = NULL, alpha = NULL,   
##   
## Call:  
## beta = NULL, gamma = NULL, phi = NULL, additive.only = FALSE,   
##   
## Call:  
## lambda = NULL, lower = c(rep(1e-04, 3), 0.8), upper = c(rep(0.9999,   
##   
## Call:  
## 3), 0.98), opt.crit = c("lik", "amse", "mse", "sigma",   
##   
## Call:  
## "mae"), nmse = 3, bounds = c("both", "usual", "admissible"),   
##   
## Call:  
## ic = c("aicc", "aic", "bic"), restrict = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.0087   
## beta = 0.0087   
## gamma = 1e-04   
##   
## Initial states:  
## l = 2047375.0885   
## b = 22509.7629   
## s=259168.3 654942.6 474529.8 876025.2 -475155 -852844  
## -664662.5 -412596.7 -438677.3 273215 138077.9 167976.7  
##   
## sigma: 241685.1  
##   
## AIC AICc BIC   
## 2124.856 2134.747 2161.283   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 21587.58 241685.1 202221.1 -0.08332944 7.329631 0.5512087  
## ACF1  
## Training set -0.2818997  
## ME RMSE MAE MPE MAPE MASE  
## Training set 21587.58 241685.1 202221.1 -0.08332944 7.329631 0.5512087  
## ACF1  
## Training set -0.2818997

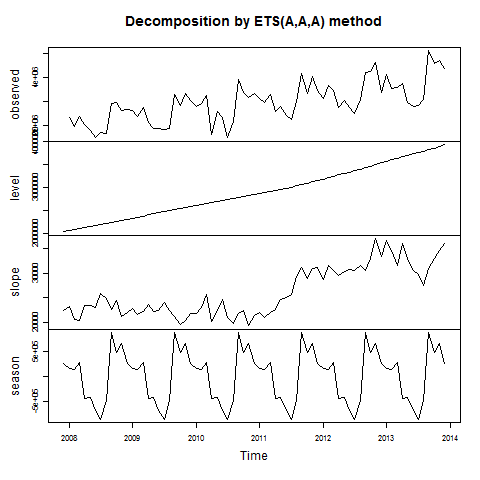


Figure 45

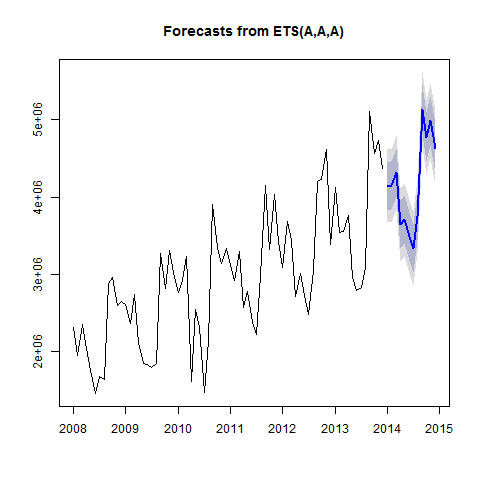


Figure 46

# AIC AICc BIC   
#2127.984 2137.875 2164.411

In order to use the results later, they need to be converted into point forcasts.

head(Model\_ets\_PointForecast)

## [1] 4141260 4147361 4318585 3642788 3704905 3488896

### 8.2 ARIMA

#### AR = Autoregression

A Regression of a variable with itself. The autoregressive model specifies that the output variable depends linearly on its own previous values.

#### MA = Moving Average

The rolling average of past forecast errors. This model should not be confused with moving average smoothing, which is used for establishing trends and is based on past values.

#### ARIMA = AutoRegressive Integrated Moving Average model A combination of Differencing, Autoregression and Moving Average. Integration is the opposite of differencing.

#### Differencing

In order to make the time series stationary, it is necessary to difference. Firstly, we need to check if the data are already stationary. This can be done with help of the Augmented Dickey-Fuller Test

print(adf.test(TotalAsIs, alternative = "stationary"))

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

##   
## Augmented Dickey-Fuller Test  
##   
## data: TotalAsIs  
## Dickey-Fuller = -5.8915, Lag order = 4, p-value = 0.01  
## alternative hypothesis: stationary

The p-value is less than 0,05. This means that the data is stationary, as the 0-Hypothesis of the test is "The data are not stationary". Another possibility is the Kwiatkowski-Phillips-Schmidt-Shin Test

print(kpss.test(TotalAsIs))

## Warning in kpss.test(TotalAsIs): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: TotalAsIs  
## KPSS Level = 2.0548, Truncation lag parameter = 1, p-value = 0.01

This test swaps the hypothesis so that a low p-value means that it is necessary to difference. The p-value here is under 0,01 and a warning is shown. As the test failed to deliver a clear result, the data will be differencedand then retested.

print(adf.test(ChulwalarDiff, alternative = "stationary"))

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

##   
## Augmented Dickey-Fuller Test  
##   
## data: ChulwalarDiff  
## Dickey-Fuller = -6.2758, Lag order = 4, p-value = 0.01  
## alternative hypothesis: stationary

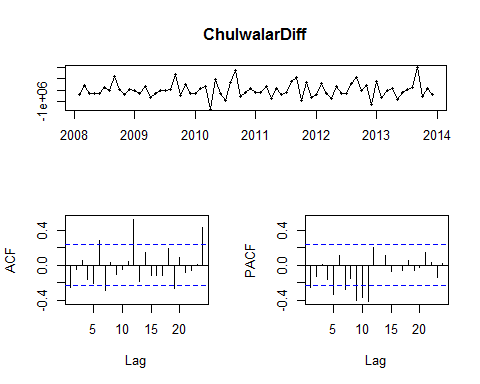
print(kpss.test(ChulwalarDiff))

## Warning in kpss.test(ChulwalarDiff): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ChulwalarDiff  
## KPSS Level = 0.024179, Truncation lag parameter = 1, p-value = 0.1

The kpss.test now has a p-value of more than 0,1, which hints that the data is stationary.

print(tsdisplay(ChulwalarDiff))



## NULL

However this plot shows that the months correlate stongly with the values from the previous year. This plot shows a ACF (autocorrelation function) and a PACF (partial autocorrelation function). The folling is a test method to distinguish the number of "normal" differencing rounds and seasonal differencing rounds. Seasonal differencing is used for data which is dominated by seasonality. The time series has been assigned a lag.

nd # Number of normal differencing rounds

## [1] 0

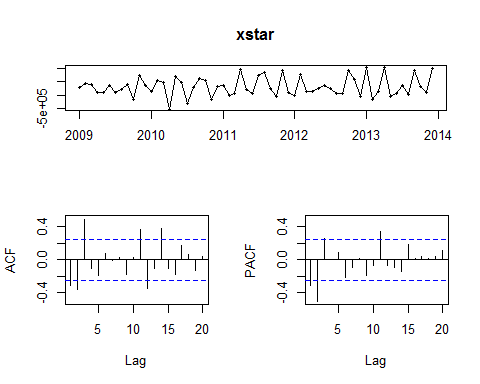
ns # Number of seasonal differencing rounds

## [1] 1

xstar # The output "xstar" has been differenced correctly.

## Jan Feb Mar Apr May Jun Jul Aug  
## 2009 297352 421196 397151 85521 93109 377920 117674 212298  
## 2010 150115 547006 483255 -511420 700084 481423 -323167 296553  
## 2011 352173 8330 67743 963191 223911 60582 748756 843266  
## 2012 -19773 752645 138580 137820 237699 347801 260934 63868  
## 2013 1026438 -143564 127610 1045166 -51834 61870 344910 28458  
## Sep Oct Nov Dec  
## 2009 388285 -140828 714282 365945  
## 2010 627400 530065 -174831 310373  
## 2011 252960 -30269 901131 96957  
## 2012 49265 910040 581464 -46170  
## 2013 906979 333420 110773 988508

print(tsdisplay(xstar))



## NULL

If "lag" is set to 12, this is equivalent to 1\* seasonal differencing

print(adf.test(ChulwalarDiff\_lag, alternative = "stationary"))

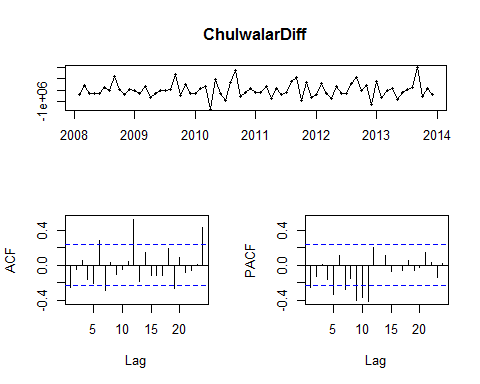
##   
## Augmented Dickey-Fuller Test  
##   
## data: ChulwalarDiff\_lag  
## Dickey-Fuller = -4.0902, Lag order = 3, p-value = 0.01175  
## alternative hypothesis: stationary

print(kpss.test(ChulwalarDiff\_lag))

## Warning in kpss.test(ChulwalarDiff\_lag): p-value greater than printed p-  
## value

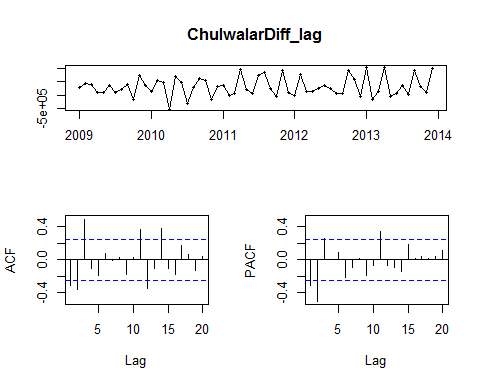
##   
## KPSS Test for Level Stationarity  
##   
## data: ChulwalarDiff\_lag  
## KPSS Level = 0.13387, Truncation lag parameter = 1, p-value = 0.1

print(tsdisplay(ChulwalarDiff))



## NULL

print(tsdisplay(ChulwalarDiff\_lag))



## NULL

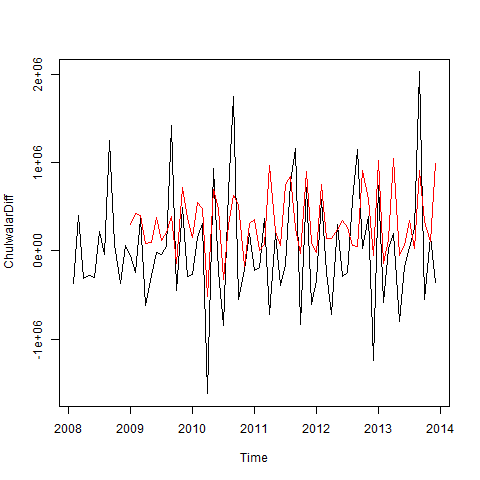


Figure 47

The time series appears much more "stationary".

#### ARIMA modelling

The values for AIC, AICc and BIC should be minimised. We wil try a range of combinations. R uses the maximum likelihood estimation (MLE) in order to decide how good a certain model is. The parameters, which give the most likely model based on the given data, are chosen. Furthermore, R gives the log-likelihood, which should be maximised.

print(summary(Model\_ARIMA\_1))

## Series: TotalAsIs   
## ARIMA(0,1,0)   
##   
## sigma^2 estimated as 4.097e+11: log likelihood=-1049.96  
## AIC=2101.93 AICc=2101.99 BIC=2104.19  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 28628.79 635581.2 472471.7 -1.605296 16.44154 1.28785  
## ACF1  
## Training set -0.2573856  
## ME RMSE MAE MPE MAPE MASE  
## Training set 28628.79 635581.2 472471.7 -1.605296 16.44154 1.28785  
## ACF1  
## Training set -0.2573856

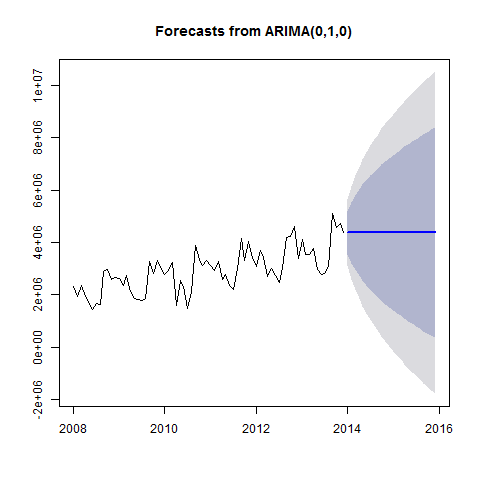


Figure 48

#AIC=2101.93 AICc=2101.99 BIC=2104.19

print(summary(Model\_ARIMA\_2))

## Series: TotalAsIs   
## ARIMA(1,1,0)   
##   
## Coefficients:  
## ar1  
## -0.2531  
## s.e. 0.1144  
##   
## sigma^2 estimated as 3.884e+11: log likelihood=-1047.6  
## AIC=2099.2 AICc=2099.38 BIC=2103.72  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 37287.11 614478.1 430163.6 -1.639433 15.09104 1.172528  
## ACF1  
## Training set -0.03792494  
## ME RMSE MAE MPE MAPE MASE  
## Training set 37287.11 614478.1 430163.6 -1.639433 15.09104 1.172528  
## ACF1  
## Training set -0.03792494

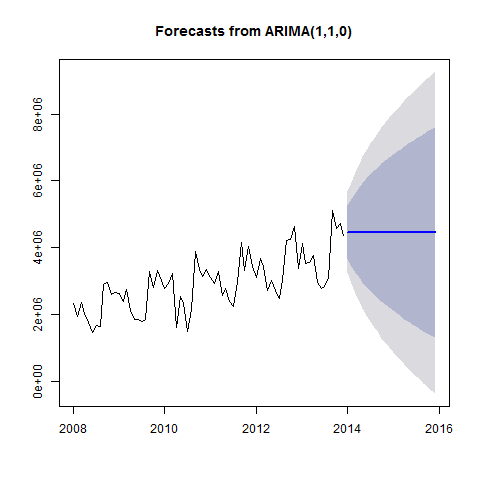


Figure 49

#AIC=2099.2 AICc=2099.38 BIC=2103.72

print(summary(Model\_ARIMA\_3))

## Series: TotalAsIs   
## ARIMA(1,1,1)   
##   
## Coefficients:  
## ar1 ma1  
## 0.5013 -0.8940  
## s.e. 0.1225 0.0523  
##   
## sigma^2 estimated as 3.483e+11: log likelihood=-1043.55  
## AIC=2093.09 AICc=2093.45 BIC=2099.88  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 108896.1 577776.4 418339 0.1597019 14.59994 1.140297  
## ACF1  
## Training set -0.04946106  
## ME RMSE MAE MPE MAPE MASE  
## Training set 108896.1 577776.4 418339 0.1597019 14.59994 1.140297  
## ACF1  
## Training set -0.04946106

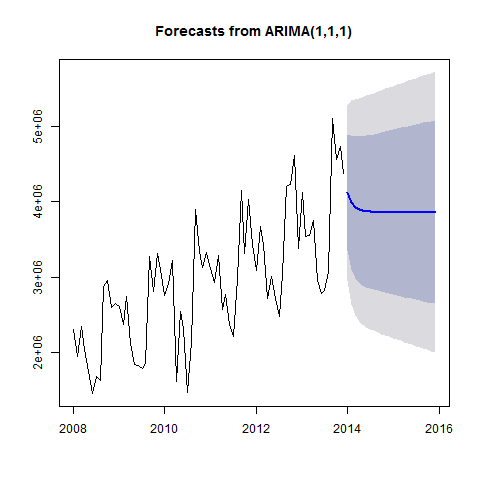


Figure 50

#AIC=2093.09 AICc=2093.45 BIC=2099.88

print(summary(Model\_ARIMA\_4))

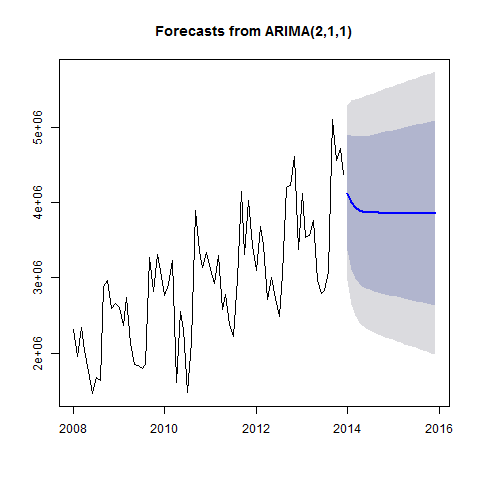


Figure 51

#AIC=2095.08 AICc=2095.68 BIC=2104.13

print(summary(Model\_ARIMA\_5))

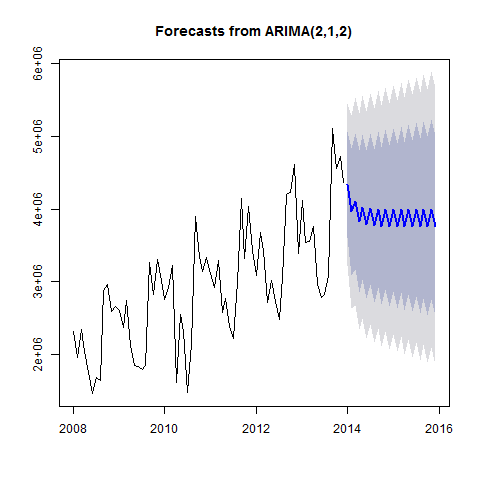


Figure 52

#AIC=2091.07 AICc=2092 BIC=2102.39

print(summary(Model\_ARIMA\_6))

## Series: TotalAsIs   
## ARIMA(3,1,2)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2  
## -0.3893 0.5013 -0.1092 0.1083 -0.8830  
## s.e. 0.1284 0.1385 0.1231 0.0569 0.0566  
##   
## sigma^2 estimated as 3.207e+11: log likelihood=-1040.15  
## AIC=2092.3 AICc=2093.61 BIC=2105.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 104678.5 542170.4 409442.9 0.3156703 14.41613 1.116048  
## ACF1  
## Training set -0.03533504  
## ME RMSE MAE MPE MAPE MASE  
## Training set 104678.5 542170.4 409442.9 0.3156703 14.41613 1.116048  
## ACF1  
## Training set -0.03533504

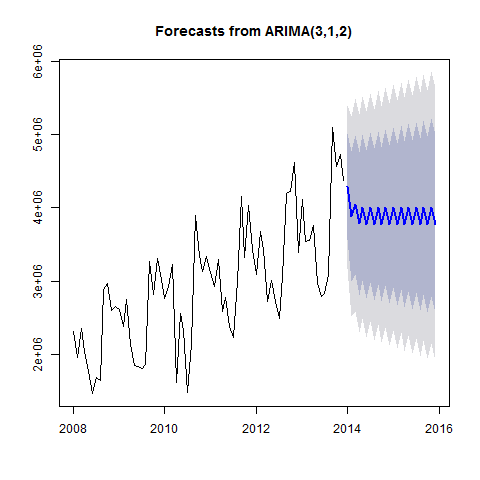


Figure 53

#AIC=2092.3 AICc=2093.61 BIC=2105.87

print(summary(Model\_ARIMA\_7))

## Series: TotalAsIs   
## ARIMA(3,1,3)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2 ma3  
## -0.3899 -0.3992 0.5784 0.0121 0.1473 -0.8863  
## s.e. 0.2719 0.2546 0.1245 0.5756 0.2820 0.6757  
##   
## sigma^2 estimated as 3.243e+11: log likelihood=-1040.01  
## AIC=2094.03 AICc=2095.81 BIC=2109.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 98929.36 541073 383116.9 0.2717968 13.13481 1.044289  
## ACF1  
## Training set 0.006450566  
## ME RMSE MAE MPE MAPE MASE  
## Training set 98929.36 541073 383116.9 0.2717968 13.13481 1.044289  
## ACF1  
## Training set 0.006450566

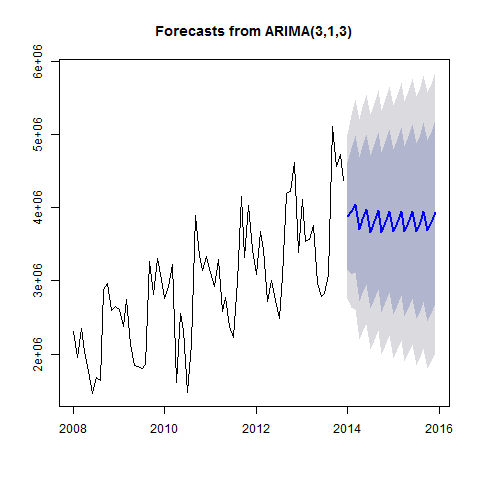


Figure 54

#AIC=2094.03 AICc=2095.81 BIC=2109.87

print(summary(Model\_ARIMA\_8))

## Series: TotalAsIs   
## ARIMA(3,1,1)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1  
## 0.4861 0.0501 -0.0876 -0.8861  
## s.e. 0.1284 0.1321 0.1224 0.0558  
##   
## sigma^2 estimated as 3.557e+11: log likelihood=-1043.29  
## AIC=2096.57 AICc=2097.5 BIC=2107.89  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 112288.1 575349.1 416026.3 0.2735252 14.58788 1.133993  
## ACF1  
## Training set -0.06722377  
## ME RMSE MAE MPE MAPE MASE  
## Training set 112288.1 575349.1 416026.3 0.2735252 14.58788 1.133993  
## ACF1  
## Training set -0.06722377

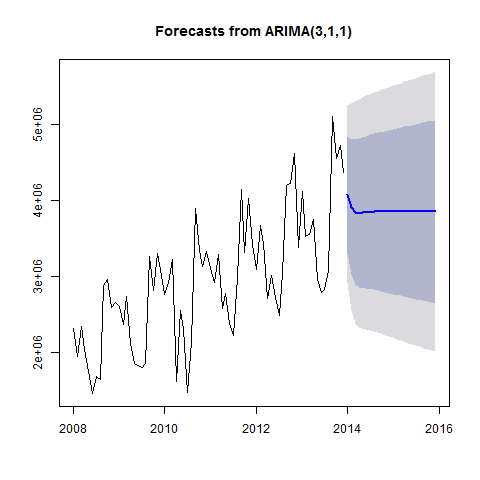


Figure 55

#AIC=2096.57 AICc=2097.5 BIC=2107.89

print(summary(Model\_ARIMA\_9))

## Series: TotalAsIs   
## ARIMA(3,1,2)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2  
## -0.3893 0.5013 -0.1092 0.1083 -0.8830  
## s.e. 0.1284 0.1385 0.1231 0.0569 0.0566  
##   
## sigma^2 estimated as 3.207e+11: log likelihood=-1040.15  
## AIC=2092.3 AICc=2093.61 BIC=2105.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 104678.5 542170.4 409442.9 0.3156703 14.41613 1.116048  
## ACF1  
## Training set -0.03533504  
## ME RMSE MAE MPE MAPE MASE  
## Training set 104678.5 542170.4 409442.9 0.3156703 14.41613 1.116048  
## ACF1  
## Training set -0.03533504

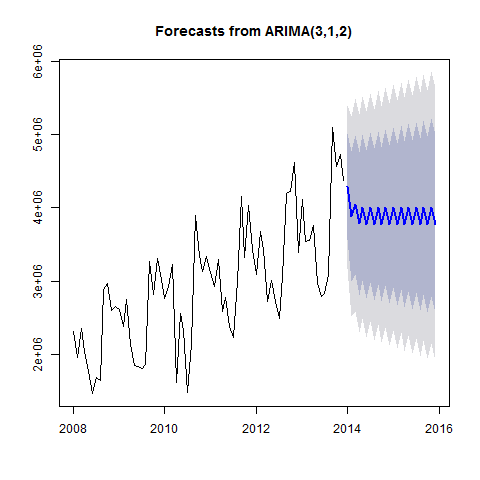


Figure 56

#AIC=2092.3 AICc=2093.61 BIC=2105.87

print(summary(Model\_ARIMA\_10))

## Series: TotalAsIs   
## ARIMA(1,1,3)   
##   
## Coefficients:  
## ar1 ma1 ma2 ma3  
## 0.4527 -0.8779 0.0965 -0.1023  
## s.e. 0.2410 0.2408 0.2460 0.1621  
##   
## sigma^2 estimated as 3.567e+11: log likelihood=-1043.34  
## AIC=2096.69 AICc=2097.61 BIC=2108  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 108260.8 576125 411814.8 0.1681833 14.39459 1.122513  
## ACF1  
## Training set -0.02687555  
## ME RMSE MAE MPE MAPE MASE  
## Training set 108260.8 576125 411814.8 0.1681833 14.39459 1.122513  
## ACF1  
## Training set -0.02687555

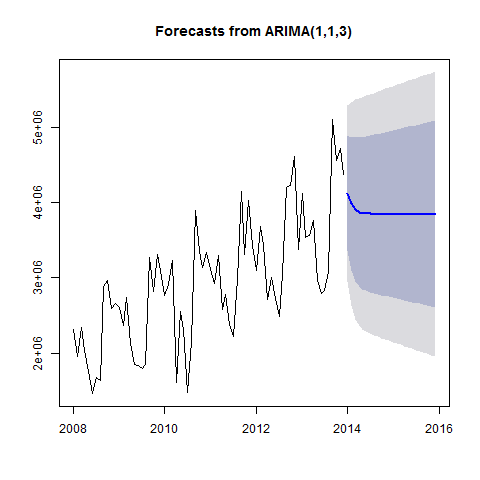


Figure 57

#AIC=2096.69 AICc=2097.61 BIC=2108

print(summary(Model\_ARIMA\_11))

## Series: TotalAsIs   
## ARIMA(2,1,3)   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3  
## -1.1981 -0.1982 1.1439 -0.6940 -0.8420  
## s.e. 0.1350 0.1350 0.1239 0.1394 0.1099  
##   
## sigma^2 estimated as 2.64e+11: log likelihood=-1036.61  
## AIC=2085.22 AICc=2086.53 BIC=2098.8  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 96418 491927 381028.7 0.4152085 13.44772 1.038597 0.003938262  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 96418 491927 381028.7 0.4152085 13.44772 1.038597 0.003938262

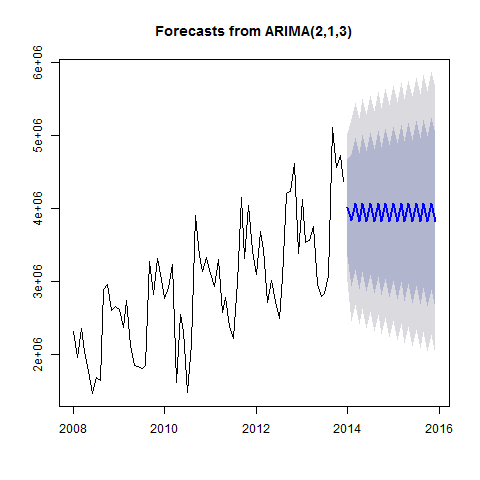


Figure 58

#AIC=2085.22 AICc=2086.53 BIC=2098.8

print(summary(Model\_ARIMA\_12))

## Series: TotalAsIs   
## ARIMA(2,2,3)   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3  
## -0.4575 0.5423 -0.9992 -0.9842 0.9938  
## s.e. 0.1097 0.1097 0.0600 0.0468 0.0596  
##   
## sigma^2 estimated as 2.933e+11: log likelihood=-1026.69  
## AIC=2065.39 AICc=2066.72 BIC=2078.88  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 32207.69 514572.7 390086.2 -1.782274 13.95983 1.063286  
## ACF1  
## Training set 0.0523144  
## ME RMSE MAE MPE MAPE MASE  
## Training set 32207.69 514572.7 390086.2 -1.782274 13.95983 1.063286  
## ACF1  
## Training set 0.0523144

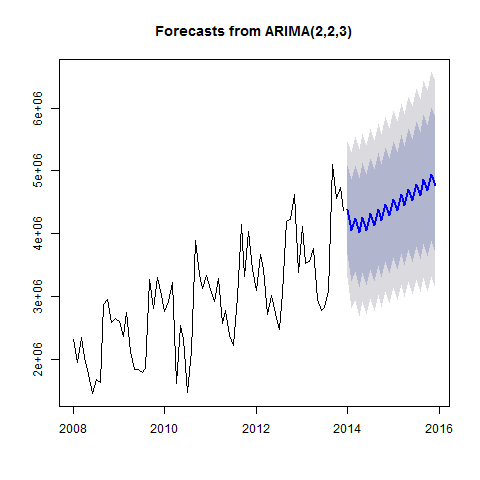


Figure 59

#AIC=2065.39 AICc=2066.72 BIC=2078.88

print(summary(Model\_ARIMA\_13))

## Series: TotalAsIs   
## ARIMA(2,3,2)   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2  
## -0.2556 -0.0955 -1.9949 0.9998  
## s.e. 0.1213 0.1203 0.0568 0.0565  
##   
## sigma^2 estimated as 4.154e+11: log likelihood=-1025.64  
## AIC=2061.27 AICc=2062.22 BIC=2072.44  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -18789.17 612404.9 442164 -3.108129 15.40378 1.205238  
## ACF1  
## Training set -0.0160389  
## ME RMSE MAE MPE MAPE MASE  
## Training set -18789.17 612404.9 442164 -3.108129 15.40378 1.205238  
## ACF1  
## Training set -0.0160389

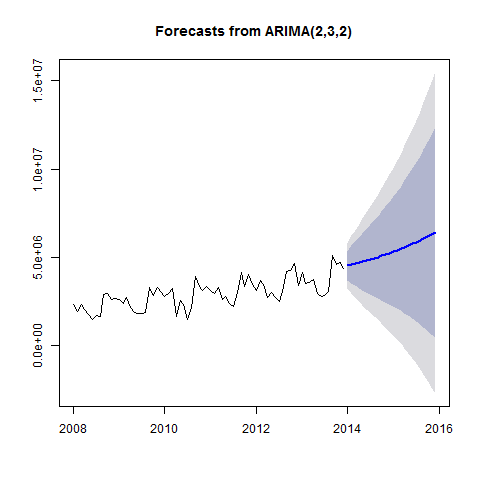
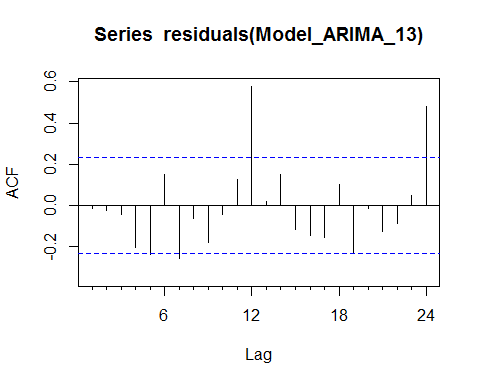


Figure 60

#AIC=2061.27 AICc=2062.22 BIC=2072.44

print(Acf(residuals(Model\_ARIMA\_13)))



##   
## Autocorrelations of series 'residuals(Model\_ARIMA\_13)', by lag  
##   
## 0 1 2 3 4 5 6 7 8 9   
## 1.000 -0.016 -0.025 -0.043 -0.201 -0.240 0.151 -0.256 -0.065 -0.181   
## 10 11 12 13 14 15 16 17 18 19   
## -0.044 0.128 0.578 0.020 0.153 -0.115 -0.146 -0.154 0.100 -0.234   
## 20 21 22 23 24   
## -0.013 -0.125 -0.088 0.049 0.479

print(Box.test(residuals(Model\_ARIMA\_13), lag=12, fitdf=4, type="Ljung"))

##   
## Box-Ljung test  
##   
## data: residuals(Model\_ARIMA\_13)  
## X-squared = 49.513, df = 8, p-value = 5.068e-08

The Ljung-Box Test has H0: The data are independently distributed und Ha: The data are not independently distributed. Just like the remainder showed before, there is a definite coherence.

#### Seasonal ARIMA modelling

This model integrates the seasonal aspect into the ARIMA model. As the previous models all had a peak in lag 12, it seems viable that the data are seasonal.

summary(Model\_Seasonal\_ARIMA\_0)

## Series: TotalAsIs   
## ARIMA(0,0,0)(1,0,0)[12] with non-zero mean   
##   
## Coefficients:  
## sar1 intercept  
## 0.8670 2972908.7  
## s.e. 0.0496 230693.3  
##   
## sigma^2 estimated as 2.211e+11: log likelihood=-1049.89  
## AIC=2105.79 AICc=2106.14 BIC=2112.62  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 178628.1 463638.5 354855 3.056776 12.76901 0.967254 0.2749878

#AIC=2105.79 AICc=2106.14 BIC=2112.62

print(summary(Model\_Seasonal\_ARIMA\_1))

## Series: TotalAsIs   
## ARIMA(0,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ma1 sma1  
## -0.9999 -0.7599  
## s.e. 0.1180 0.3144  
##   
## sigma^2 estimated as 8.71e+10: log likelihood=-833.44  
## AIC=1672.88 AICc=1673.31 BIC=1679.11  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 27419.53 262591.5 185283.1 -0.2676759 6.296359 0.5050396  
## ACF1  
## Training set -0.2913248  
## ME RMSE MAE MPE MAPE MASE  
## Training set 27419.53 262591.5 185283.1 -0.2676759 6.296359 0.5050396  
## ACF1  
## Training set -0.2913248

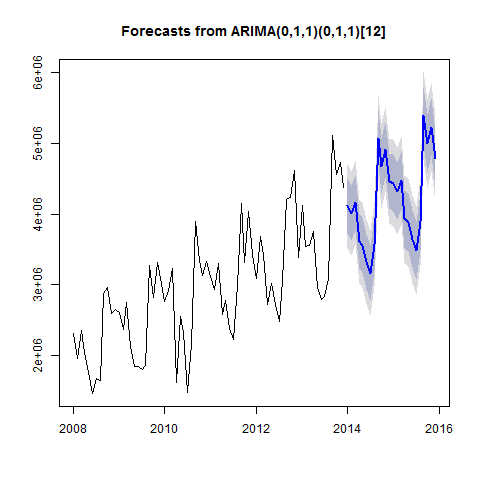


Figure 61

#AIC=1672.88 AICc=1673.31 BIC=1679.11

Insert the values from the previous chapter for the non-seasonal values.

print(summary(Model\_Seasonal\_ARIMA\_2))

## Series: TotalAsIs   
## ARIMA(2,3,2)(1,1,1)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sar1 sma1  
## -0.8870 -0.7259 -1.9692 0.9977 -0.1289 -0.5091  
## s.e. 0.0876 0.0907 0.0699 0.0697 0.2692 0.3059  
##   
## sigma^2 estimated as 9.302e+10: log likelihood=-808.11  
## AIC=1630.23 AICc=1632.51 BIC=1644.53  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -5252.206 256693.1 176056.7 -0.8935257 5.910925 0.4798906  
## ACF1  
## Training set -0.07108056  
## ME RMSE MAE MPE MAPE MASE  
## Training set -5252.206 256693.1 176056.7 -0.8935257 5.910925 0.4798906  
## ACF1  
## Training set -0.07108056

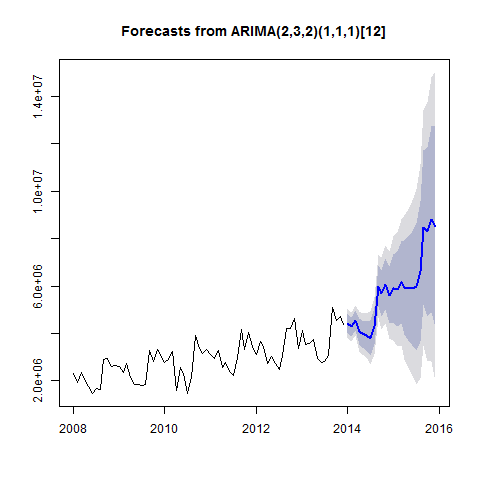


Figure 62

# AIC=1630.23 AICc=1632.51 BIC=1644.53

Good results when using drift.

print(summary(Model\_Seasonal\_ARIMA\_3))

## Series: TotalAsIs   
## ARIMA(1,0,1)(1,1,1)[12] with drift   
##   
## Coefficients:  
## ar1 ma1 sar1 sma1 drift  
## 0.0432 -0.4498 0.1916 -0.775 26229.159  
## s.e. 0.2973 0.2700 0.3590 0.528 1029.248  
##   
## sigma^2 estimated as 8.173e+10: log likelihood=-840  
## AIC=1691.99 AICc=1693.58 BIC=1704.56  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -12740.01 249858 187104 -1.804816 6.490842 0.5100028  
## ACF1  
## Training set -0.004832396  
## ME RMSE MAE MPE MAPE MASE  
## Training set -12740.01 249858 187104 -1.804816 6.490842 0.5100028  
## ACF1  
## Training set -0.004832396

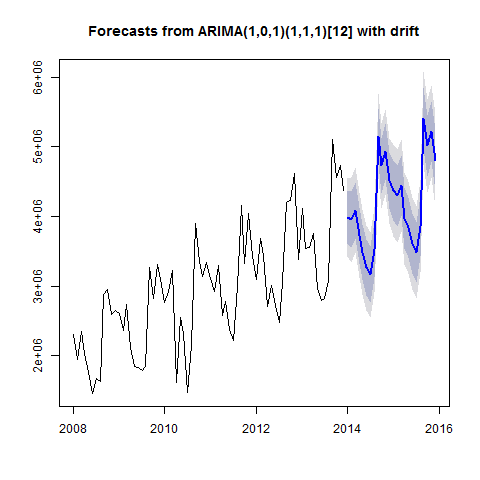


Figure 63

# AIC=1355.99 AICc=1357.58 BIC=1368.56

print(summary(Model\_Seasonal\_ARIMA\_4))

## Series: TotalAsIs   
## ARIMA(2,3,2)(1,3,2)[12]   
##   
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

## ar1 ar2 ma1 ma2 sar1 sma1 sma2  
## -0.7734 -0.7922 -1.9661 0.9954 -0.639 -0.613 0.0136  
## s.e. 0.1009 0.1204 0.1040 0.1017 NaN NaN NaN  
##   
## sigma^2 estimated as 4.148e+11: log likelihood=-502.44  
## AIC=1020.89 AICc=1026.89 BIC=1032.86  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4082.089 387044.8 209718 -0.2681636 6.055173 0.5716435  
## ACF1  
## Training set -0.2075146  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4082.089 387044.8 209718 -0.2681636 6.055173 0.5716435  
## ACF1  
## Training set -0.2075146

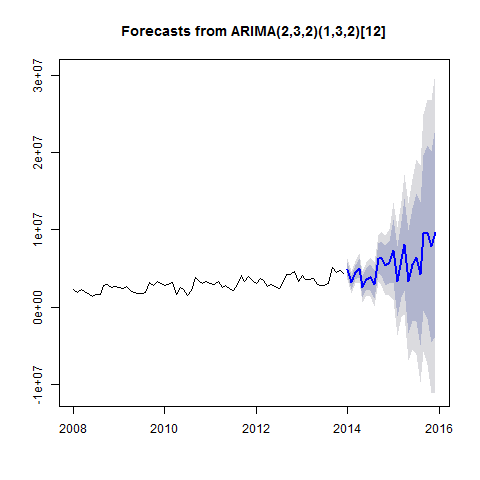


Figure 64

# AIC=1630.23 AICc=1632.51 BIC=1644.53

The stronger the seasonality is differenced, the better the results are. However the plot shows that the data are being increasingly changed.

print(summary(Model\_Seasonal\_ARIMA\_5))

## Series: TotalAsIs   
## ARIMA(2,3,2)(1,4,2)[12]   
##   
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

## ar1 ar2 ma1 ma2 sar1 sma1 sma2  
## -0.7594 -0.6887 0.0437 -0.0219 -0.552 -0.2248 0.0119  
## s.e. NaN 0.1077 NaN NaN NaN NaN NaN  
##   
## sigma^2 estimated as 1.994e+13: log likelihood=-374.5  
## AIC=765 AICc=777 BIC=773.36  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 23417.82 1969200 835437.3 0.9676692 23.85754 2.277212  
## ACF1  
## Training set -0.5364577  
## ME RMSE MAE MPE MAPE MASE  
## Training set 23417.82 1969200 835437.3 0.9676692 23.85754 2.277212  
## ACF1  
## Training set -0.5364577

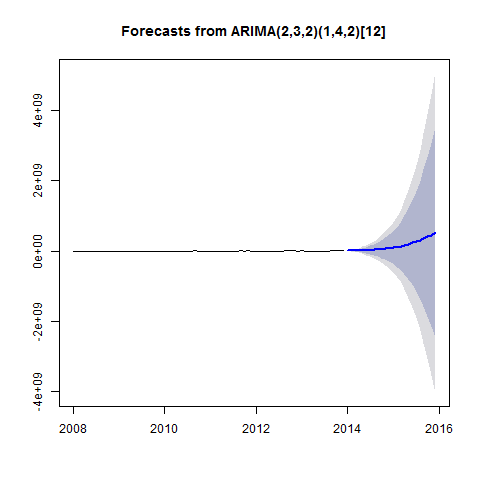


Figure 65

# AIC=765 AICc=777 BIC=773.36

The more the seasonal aspect is changed, the better the results based on AIC, AICc and BIC. Theoretically the models should more and more suitable for the forecast. However, a look at the plot of the forecasts shows that the changes are making the data less and less convincing and thus unuseable.

##### Auto-ARIMA modelling

The automatic establishment of an ARIMA model shows that (2,0,1)(0,1,1) with drift delivers the best results.

# AIC=1344.04 AICc=1345.62 BIC=1356.6

For comparison, here are the results of ModelWithTrendAndSeasonalityOnly with tslm():

# CV AIC AICc BIC AdjR2   
# 8.472378e+10 1810.912 1818.281 1842.786 0.9004392

summary(Model\_auto.arima)

## Series: TotalAsIs   
## ARIMA(2,0,1)(0,1,1)[12] with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 sma1 drift  
## -0.9408 -0.5723 0.6947 -0.5312 26298.315  
## s.e. 0.1532 0.1109 0.1945 0.1845 1118.294  
##   
## sigma^2 estimated as 6.63e+10: log likelihood=-832.52  
## AIC=1677.05 AICc=1678.63 BIC=1689.61  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -9481.321 225048.2 165219.8 -1.492866 5.660998 0.4503516  
## ACF1  
## Training set 0.01340082

CV(ModelWithTrendAndSeasonalityOnly)

## CV AIC AICc BIC AdjR2   
## 8.472378e+10 1.810912e+03 1.818281e+03 1.842786e+03 9.004392e-01

Box.test(residuals(Model\_auto.arima), lag=12, fitdf=4, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(Model\_auto.arima)  
## X-squared = 12.591, df = 8, p-value = 0.1267

The Ljung-Box Test has H0: The data are independently distributed and Ha: The data are not independently distributed. The results show: White noise

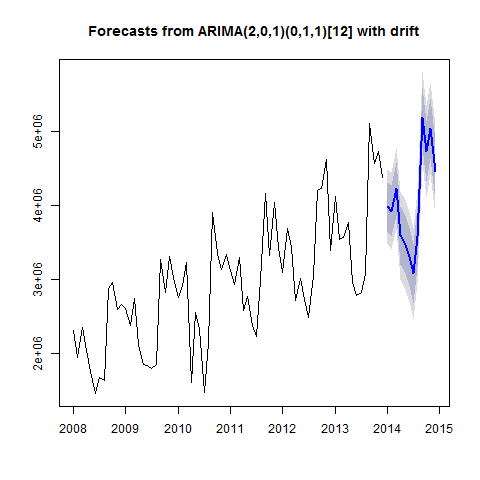


Figure 66

print(Model\_auto.arima\_PointForecast)

## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 3981722 3921507 4220286 3591865 3497687 3307882 3073575 3574462  
## Sep Oct Nov Dec  
## 2014 5179185 4726394 5033430 4460765

### 8.3 Dynamic Regression Models

Regression models are combined with ARIMA models on order to make sure that external factors are included and that the time series are not only forecasted based on past values. A regression of the ARIMA errors should be aspired for. We have to diffentiate, as the time series and the SIGov Indicator are not stationary. So that a forecast can be produced, the indicator has to be lagged so that we have values for 2014.

print(summary(Model\_dynreg))

## Series: TotalAsIs   
## ARIMA(2,2,0)   
##   
## Coefficients:  
## ar1 ar2 CEPI\_lagged  
## -0.8375 -0.4848 -272894.3  
## s.e. 0.1133 0.1165 285739.4  
##   
## sigma^2 estimated as 5.131e+11: log likelihood=-872.28  
## AIC=1752.56 AICc=1753.17 BIC=1761.55  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -6115.637 756954.4 568807.6 -1.554716 18.4045 1.55044  
## ACF1  
## Training set -0.09154476  
## ME RMSE MAE MPE MAPE MASE  
## Training set -6115.637 756954.4 568807.6 -1.554716 18.4045 1.55044  
## ACF1  
## Training set -0.09154476

print(summary(Model\_dynreg\_auto.arima))

## Series: TotalAsIs   
## ARIMA(2,1,0)(1,1,0)[12]   
##   
## Coefficients:  
## ar1 ar2 sar1 CEPI\_lagged  
## -0.9138 -0.7234 -0.5714 12819.6  
## s.e. 0.0981 0.0986 0.1323 108321.3  
##   
## sigma^2 estimated as 7.409e+10: log likelihood=-661.69  
## AIC=1333.39 AICc=1334.52 BIC=1343.77  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10997.11 260603.8 192254.9 -0.3570223 6.202412 0.524043  
## ACF1  
## Training set -0.08855174  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10997.11 260603.8 192254.9 -0.3570223 6.202412 0.524043  
## ACF1  
## Training set -0.08855174

ARIMA(2,0,1)(0,1,1)[12] with drift

# AIC=1343.61 AICc=1345.76 BIC=1358.27

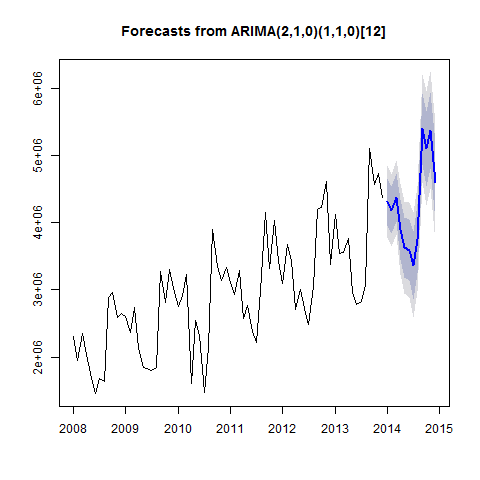
The Ljung-Box Test has H0: The data are independently distributed and Ha: The data are not independently distributed. The results show: White noise.

print(summary(Model\_dynreg\_auto.arima))

## Series: TotalAsIs   
## ARIMA(2,1,0)(1,1,0)[12]   
##   
## Coefficients:  
## ar1 ar2 sar1 CEPI\_lagged  
## -0.9138 -0.7234 -0.5714 12819.6  
## s.e. 0.0981 0.0986 0.1323 108321.3  
##   
## sigma^2 estimated as 7.409e+10: log likelihood=-661.69  
## AIC=1333.39 AICc=1334.52 BIC=1343.77  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10997.11 260603.8 192254.9 -0.3570223 6.202412 0.524043  
## ACF1  
## Training set -0.08855174  
## ME RMSE MAE MPE MAPE MASE  
## Training set 10997.11 260603.8 192254.9 -0.3570223 6.202412 0.524043  
## ACF1  
## Training set -0.08855174

Box.test(residuals(Model\_dynreg\_auto.arima), lag=12, fitdf=4, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(Model\_dynreg\_auto.arima)  
## X-squared = 14.198, df = 8, p-value = 0.07674

 if(!checkPlot("./Model\_dynreg\_auto\_arima\_forecast\_plot")) { #checks to see if plot exists and is up to date source("Plots/Model\_dynreg\_auto\_arima\_forecast\_plot.r") }

print(Model\_dynreg\_auto.arima\_forecast)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2014 4311394 3962566 4660222 3777908 4844880  
## Feb 2014 4182765 3832643 4532886 3647299 4718230  
## Mar 2014 4376006 4019152 4732860 3830245 4921767  
## Apr 2014 3903181 3459270 4347091 3224278 4582084  
## May 2014 3630475 3182836 4078114 2945870 4315080  
## Jun 2014 3593592 3133773 4053411 2890359 4296825  
## Jul 2014 3369718 2863778 3875659 2595949 4143488  
## Aug 2014 3756013 3243800 4268226 2972651 4539375  
## Sep 2014 5395076 4868274 5921878 4589401 6200750  
## Oct 2014 5099481 4543257 5655706 4248809 5950154  
## Nov 2014 5376694 4812136 5941251 4513278 6240110  
## Dec 2014 4590587 4010944 5170230 3704100 5477074

print(Model\_dynreg\_auto.arima\_PointForecast)

## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 4311394 4182765 4376006 3903181 3630475 3593592 3369718 3756013  
## Sep Oct Nov Dec  
## 2014 5395076 5099481 5376694 4590587

## 9.0 Conclusion

##### SES model

AIC(Model\_sesEtel$model)

## [1] 2188.81

accuracy(Model\_sesEtel)

## ME RMSE MAE MPE MAPE MASE  
## Training set 17461.77 457693.8 297227.3 -4.846428 21.14544 1.453283  
## ACF1  
## Training set 0.02266816

#### Holt model

AIC(Model\_holt\_1Etel$model)

## [1] 2193.032

accuracy(Model\_holt\_1Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set -23016.69 458401.4 311051.9 -7.993243 22.56079 1.520877  
## ACF1  
## Training set 0.02160216

##### Holt Model with Exponential Trend

AIC(Model\_holt\_2Etel$model)

## [1] 2226.746

accuracy(Model\_holt\_2Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set 51092.46 457072 290407.9 -2.258191 20.3863 1.419939  
## ACF1  
## Training set 0.004350746

##### Dampened Holt Model

AIC(Model\_holt\_3Etel$model)

## [1] 2194.944

accuracy(Model\_holt\_3Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set 14074.53 458121.2 299379.3 -5.181928 21.35676 1.463805  
## ACF1  
## Training set 0.02356552

##### Dampened Holt Model with Exponential Trend

AIC(Model\_holt\_4Etel$model)

## [1] 2228.461

accuracy(Model\_holt\_4Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set 46720.45 457308.5 292360.8 -2.502549 20.54122 1.429488  
## ACF1  
## Training set 0.007568185

##### Holt Winters Additive Model

AIC(Model\_hw\_1Etel$model)

## [1] 2076.452

accuracy(Model\_hw\_1Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set -19817 172689.4 137259 -3.310116 10.74612 0.6711232  
## ACF1  
## Training set -0.1543248

##### Holt Winters Multiplicative Model

AIC(Model\_hw\_2Etel$model)

## [1] 2071.694

accuracy(Model\_hw\_2Etel)

## ME RMSE MAE MPE MAPE MASE  
## Training set 2570.544 154506.4 126050.4 -1.274176 9.741218 0.6163191  
## ACF1  
## Training set -0.2582845

##### Minimizing the AIC and RMSE from the models, it is clear that the best model for the "TotalEtel" forecast is the Holt-Winters model. We can see that the Holt-Winters Multiplicative Model is marginally better than the Additive model. We assume this is the case since this is the only model that takes into account the seasonality changing proportionally to the level of the data. Whereas the Additive model would have been better if the "TotalEtel" seasonality would have been roughly constant throughout the data. The Total Etel exports were found to be highly seasonal so this makes sense. This may be attributed to the Winter blooming flower and the demonstrated correlation between the Total Etel Exports and the Influential National Holidays. Our complete predicted forecast for 2014 Total Etel Exports, based on the Holt-Winters Multiplicative Model, is as follows:

Model\_hw\_2\_PointForecastEtel

## Jan Feb Mar Apr May Jun Jul Aug  
## 2014 2134821 2018272 1990076 1517542 1384653 1228015 1105178 1401176  
## Sep Oct Nov Dec  
## 2014 3067063 2547465 2613831 2325692

## 10.0 Acknowledgements

##### While this document details the specific models and data for the "TotalEtel" flowers all data and code are based off of information provided in the document entitled "Forecasting Exports Chulwalar\_0.8a.R". This document can be found in this repository in the data folder.

[/Analysis/Data](https://github.com/clairecDS/DoingDataScience_CaseStudy2/blob/master/Data/Forecasting%20Exports%20Chulwalar_0.8a.R) #####We would like to credit the following for their contributions to this document. #####Amy Wheeler #####Nina Weitkamp #####Patrick Berlekamp #####Johannes Brauer #####Andreas Faatz #####Hans-Ulrich Holst #####Designed and coded at Hochschule Osnabrück, Germany #####Contact: [faatz@wi.hs-osnabrueck.de](mailto:faatz@wi.hs-osnabrueck.de) #####Additionally, we would like to thank: Rob Hyndman for the forecasting libraries in R