Climate Change Time Series

Austin Jia

2022-03-16

###Libraries

library(xts)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

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

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:xts':  
##   
## first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(hrbrthemes)

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

library(forecast)

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

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

###Reading Data

train = read.csv("DailyDelhiClimateTrain.csv")  
test = read.csv("DailyDelhiClimateTest.csv")  
full = rbind(train, test)  
  
nrow(train)

## [1] 1462

nrow(test)

## [1] 114

###Data Wrangling

#Converting to   
train$date = as.Date(train$date, format = "%Y-%m-%d")  
test$date = as.Date(test$date, format = "%Y-%m-%d")  
full$date = as.Date(full$date, format = "%Y-%m-%d")  
  
#Convert first day to day of the year  
dayofyear = as.numeric(format(train[1,1], "%j"))  
train.ts = ts(train$meantemp, start = c(2013, dayofyear), frequency=365)  
test.ts = ts(test$meantemp, start = c(2013, dayofyear), frequency=365)  
full.ts = ts(full$meantemp, start= c(2013, dayofyear), frequency=365)

###Exploratory Data Analysis

max(full$meantemp)

## [1] 38.71429

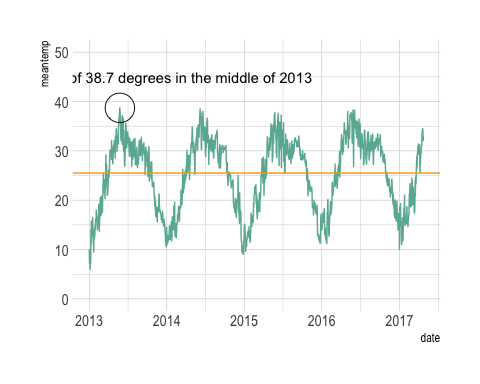
mean(full$meantemp)

## [1] 25.22192

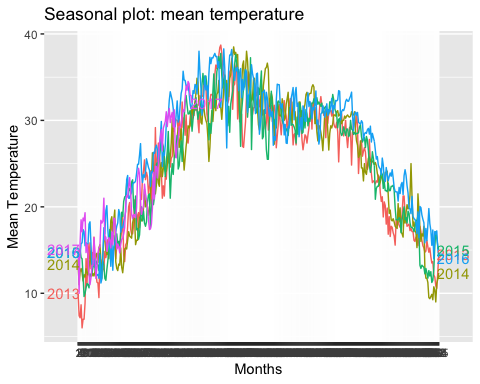
full$date[full$meantemp == max(full$meantemp)]

## [1] "2013-05-25"

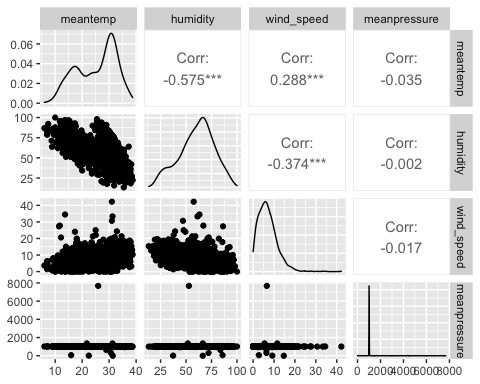
#Overall time-seriesplot  
full %>%  
 ggplot(aes(x=date, y=meantemp)) +  
 geom\_line(color="#69b3a2") +  
 ylim(0, 50) +  
 annotate(geom="text", x=as.Date("2014-01-25"), y=45, label = "Peak of 38.7 degrees in the middle of 2013") +  
 annotate(geom = "point", x=as.Date("2013-05-25"), y=38.71429, size = 10, shape=21, fill = "transparent") +  
 geom\_hline(yintercept = mean(train$meantemp), color = "orange", size = .5) +  
 theme\_ipsum()



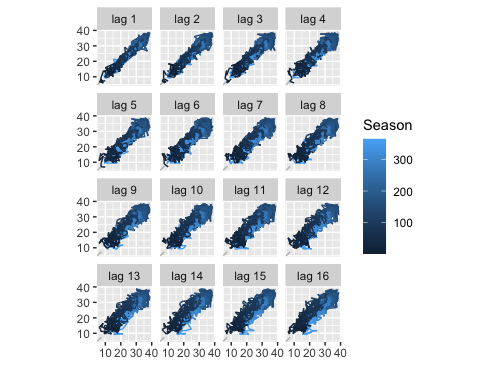
#Seasonal plots  
ggseasonplot(full.ts, year.labels=TRUE, year.labels.left=TRUE) +  
 ylab("Mean Temperature") +  
 xlab("Months") +   
 ggtitle("Seasonal plot: mean temperature")



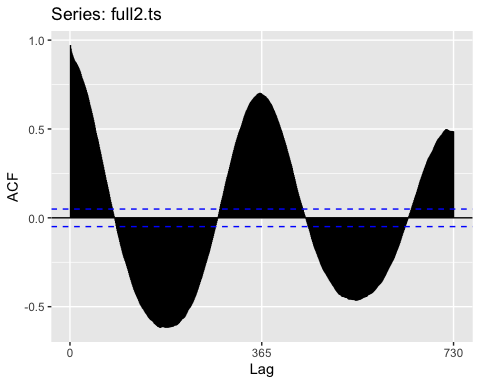
#Distribution plots with other covariates  
GGally::ggpairs(as.data.frame(full[,2:5]))



#Lag plots -- reflects strong seasonality  
full2.ts <- window(full.ts, start=2013)  
gglagplot(full2.ts, h = 12)

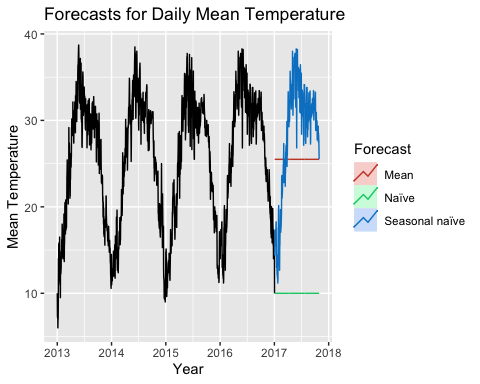


ggAcf(full2.ts)

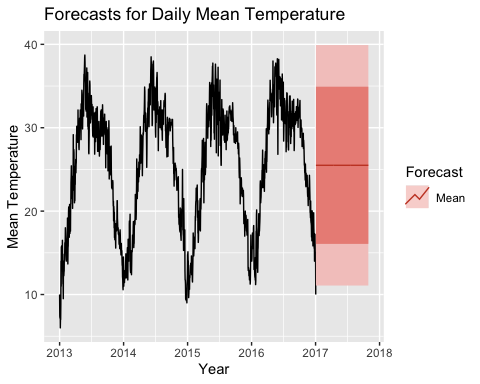


###Forecasting – No Other Predictors ####Forecasting

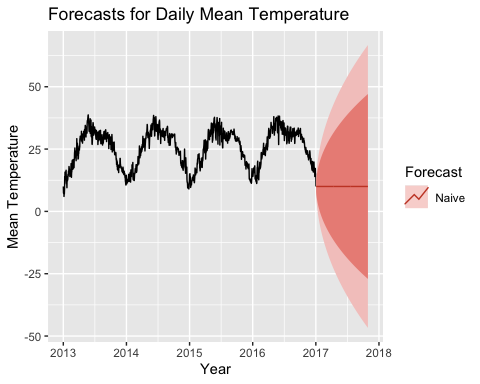
#Benchmarks of mean, naive, and seasonal naive  
autoplot(train.ts) +  
 autolayer(meanf(train.ts, h=300),  
 series="Mean", PI=FALSE) +  
 autolayer(naive(train.ts, h=300),  
 series="Naïve", PI=FALSE) +  
 autolayer(snaive(train.ts, h=300),  
 series="Seasonal naïve", PI=FALSE) +  
 ggtitle("Forecasts for Daily Mean Temperature") +  
 xlab("Year") + ylab("Mean Temperature") +  
 guides(colour=guide\_legend(title="Forecast"))



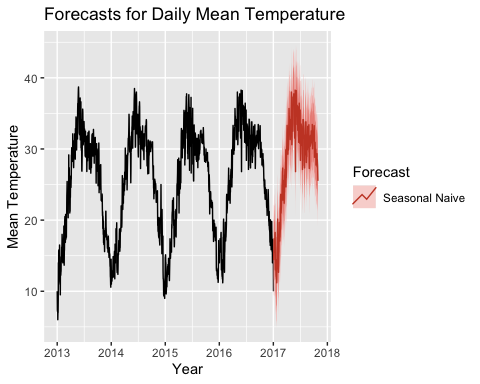
#Benchmark forecasts with intervals  
autoplot(train.ts) + autolayer(meanf(train.ts, h=300),  
 series="Mean", PI=TRUE) +  
 ggtitle("Forecasts for Daily Mean Temperature") +  
 xlab("Year") + ylab("Mean Temperature") +  
 guides(colour=guide\_legend(title="Forecast"))



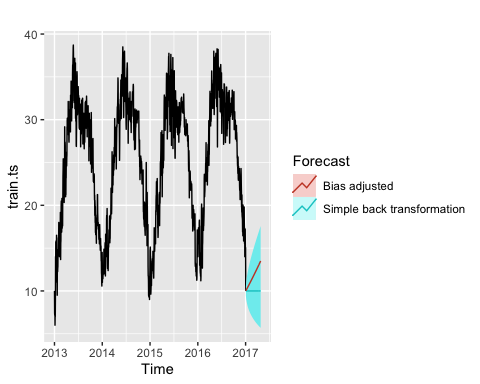
autoplot(train.ts) + autolayer(naive(train.ts, h=300),  
 series="Naive", PI=TRUE) +  
 ggtitle("Forecasts for Daily Mean Temperature") +  
 xlab("Year") + ylab("Mean Temperature") +  
 guides(colour=guide\_legend(title="Forecast"))



autoplot(train.ts) + autolayer(snaive(train.ts, h=300),  
 series="Seasonal Naive", PI=TRUE) +  
 ggtitle("Forecasts for Daily Mean Temperature") +  
 xlab("Year") + ylab("Mean Temperature") +  
 guides(colour=guide\_legend(title="Forecast"))

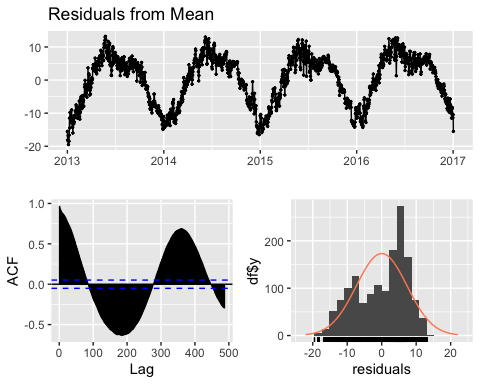


#Drift, with bias-adjustment  
drift1 = rwf(train.ts, drift=TRUE, lambda=0, h=114, level=50)  
drift2 = rwf(train.ts, drift=TRUE, lambda=0, h=114, level=50,  
 biasadj=TRUE)  
autoplot(train.ts) +  
 autolayer(drift1, series="Simple back transformation") +  
 autolayer(drift2, series="Bias adjusted", PI=FALSE) +  
 guides(colour=guide\_legend(title="Forecast"))



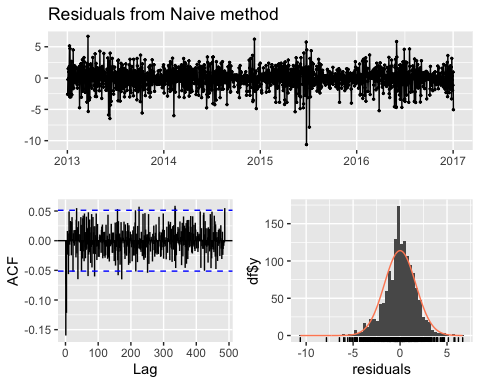
####Residuals

#Mean  
checkresiduals(meanf(train.ts))



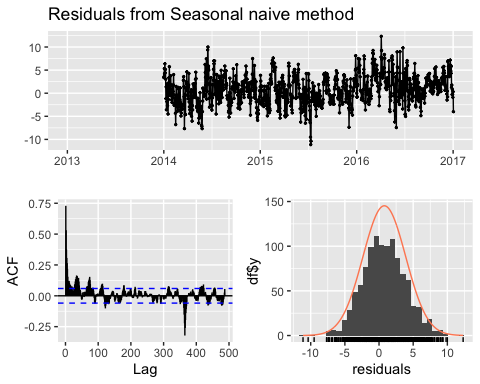
##   
## Ljung-Box test  
##   
## data: Residuals from Mean  
## Q\* = 116833, df = 291, p-value < 2.2e-16  
##   
## Model df: 1. Total lags used: 292

#Naive  
checkresiduals(naive(train.ts))



##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 336.05, df = 292, p-value = 0.03873  
##   
## Model df: 0. Total lags used: 292

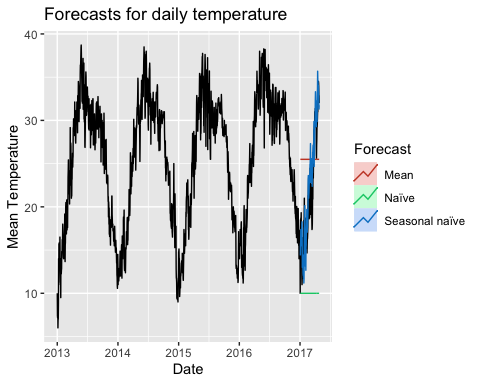
#Seasonal Naive  
checkresiduals(snaive(train.ts))



##   
## Ljung-Box test  
##   
## data: Residuals from Seasonal naive method  
## Q\* = 2364, df = 292, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 292

#### Accuracy

#Visualization of forecast against test set  
meanf <- meanf(train.ts,h=114)  
rfw <- rwf(train.ts,h=114)  
snaive <- snaive(train.ts,h=114)  
autoplot(window(full.ts, start=2013)) +  
 autolayer(meanf, series="Mean", PI=FALSE) +  
 autolayer(rfw, series="Naïve", PI=FALSE) +  
 autolayer(snaive, series="Seasonal naïve", PI=FALSE) +  
 xlab("Date") + ylab("Mean Temperature") +  
 ggtitle("Forecasts for daily temperature") +  
 guides(colour=guide\_legend(title="Forecast"))



#Quantifying errors  
test2.ts <- window(full.ts, start=2017)  
accuracy(meanf, test2.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set -7.577135e-17 7.345589 6.36267 -11.92700 31.61768 2.488531  
## Test set -3.782442e+00 7.375808 6.61238 -27.69661 36.77720 2.586197  
## ACF1 Theil's U  
## Training set 0.9711027 NA  
## Test set 0.9494438 5.140798

accuracy(rfw, test2.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.00000 1.670899 1.239266 -0.2872711 5.354941 0.4846949  
## Test set 11.71308 13.315101 11.713079 49.9141007 49.914101 4.5811529  
## ACF1 Theil's U  
## Training set -0.1598851 NA  
## Test set 0.9494438 6.035655

accuracy(snaive, test2.ts)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.7768687 3.230929 2.556797 2.447009 10.88840 1.0000000  
## Test set -1.0740657 3.092394 2.523022 -5.668860 13.09646 0.9867899  
## ACF1 Theil's U  
## Training set 0.7294180 NA  
## Test set 0.7450584 1.932563

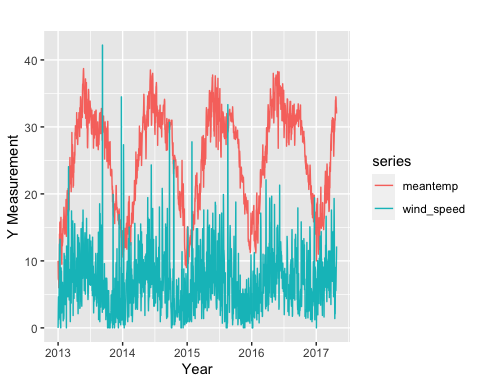
###Time Series Regression ####Data Wrangling

full.ts.predictors = ts(full, start=2013, frequency=365)  
train.ts.predictors = ts(train, start=2013, frequency=365)  
test.ts.predictors = ts(test, start=2017, frequency=365)  
  
colnames(full)

## [1] "date" "meantemp" "humidity" "wind\_speed" "meanpressure"

####EDA

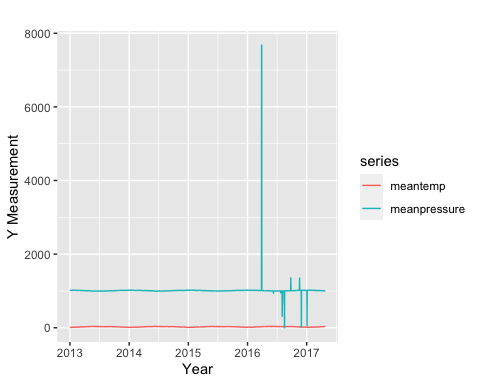
autoplot(full.ts.predictors[,c("meantemp","wind\_speed")]) +  
 ylab("Y Measurement") + xlab("Year")



autoplot(full.ts.predictors[,c("meantemp","humidity")]) +  
 ylab("Y Measurement") + xlab("Year")



autoplot(full.ts.predictors[,c("meantemp","meanpressure")]) +  
 ylab("Y Measurement") + xlab("Year")

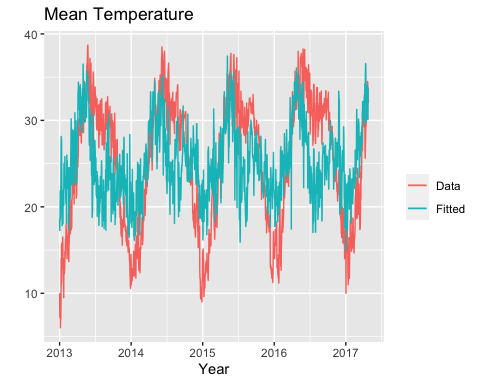


####Modeling

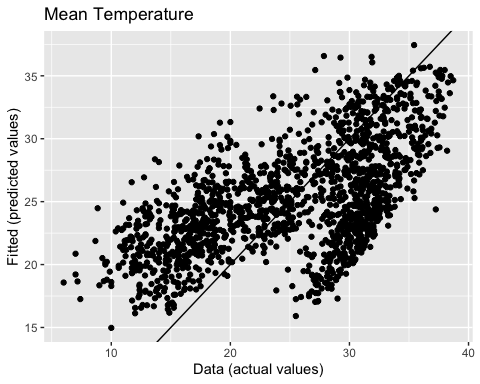
#Fitting the Model  
fit.predictors <- tslm(  
 meantemp ~ humidity + wind\_speed + meanpressure,  
 data=full.ts.predictors)  
summary(fit.predictors)

##   
## Call:  
## tslm(formula = meantemp ~ humidity + wind\_speed + meanpressure,   
## data = full.ts.predictors)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.6184 -4.7132 -0.0895 5.3098 12.8673   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 39.9254587 1.1399796 35.023 < 2e-16 \*\*\*  
## humidity -0.2350271 0.0095711 -24.556 < 2e-16 \*\*\*  
## wind\_speed 0.1376284 0.0360342 3.819 0.000139 \*\*\*  
## meanpressure -0.0014316 0.0008604 -1.664 0.096309 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.982 on 1572 degrees of freedom  
## Multiple R-squared: 0.3379, Adjusted R-squared: 0.3366   
## F-statistic: 267.4 on 3 and 1572 DF, p-value: < 2.2e-16

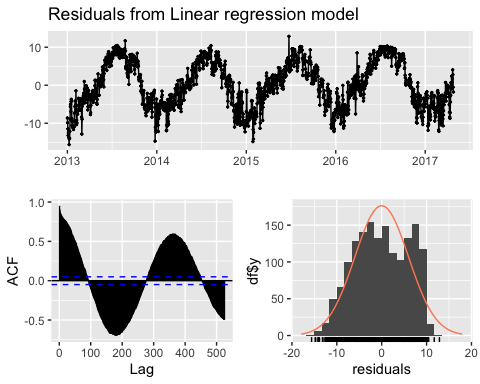
#Time plot of actual temperatures and predicted temperatures  
autoplot(full.ts.predictors[,'meantemp'], series="Data") +  
 autolayer(fitted(fit.predictors), series="Fitted") +  
 xlab("Year") + ylab("") +  
 ggtitle("Mean Temperature") +  
 guides(colour=guide\_legend(title=" "))



#Actual temperature plotted against predicted temperature  
cbind(Data = full.ts.predictors[,"meantemp"],  
 Fitted = fitted(fit.predictors)) %>%  
 as.data.frame() %>%  
 ggplot(aes(x=Data, y=Fitted)) +  
 geom\_point() +  
 ylab("Fitted (predicted values)") +  
 xlab("Data (actual values)") +  
 ggtitle("Mean Temperature") +  
 geom\_abline(intercept=0, slope=1)

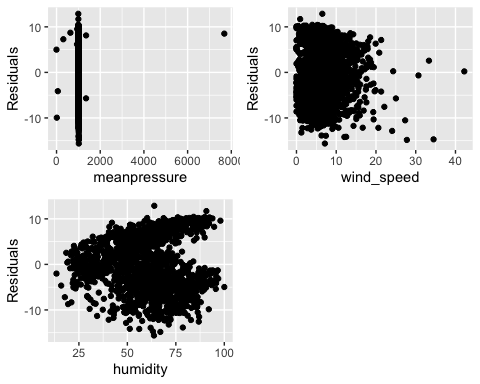


#Checking Residuals  
checkresiduals(fit.predictors)

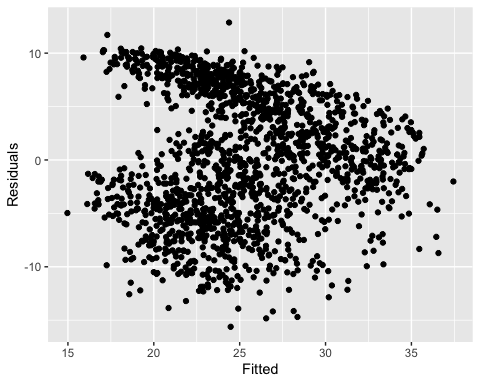


##   
## Breusch-Godfrey test for serial correlation of order up to 315  
##   
## data: Residuals from Linear regression model  
## LM test = 1489.2, df = 315, p-value < 2.2e-16

#Residual plots against predictors -- don't look good!  
df <- as.data.frame(full.ts.predictors)  
df[,"Residuals"] <- as.numeric(residuals(fit.predictors))  
p1 <- ggplot(df, aes(x=meanpressure, y=Residuals)) +  
 geom\_point()  
p2 <- ggplot(df, aes(x=wind\_speed, y=Residuals)) +  
 geom\_point()  
p3 <- ggplot(df, aes(x=humidity, y=Residuals)) +  
 geom\_point()  
gridExtra::grid.arrange(p1, p2, p3, nrow=2)



#Residuals versus fitted values -- violates homoscedasticity  
cbind(Fitted = fitted(fit.predictors),  
 Residuals=residuals(fit.predictors)) %>%  
 as.data.frame() %>%  
 ggplot(aes(x=Fitted, y=Residuals)) + geom\_point()



CV(fit.predictors)

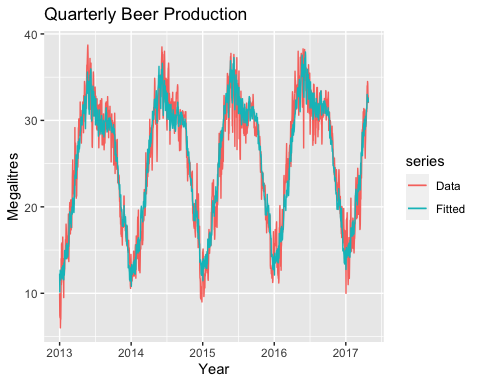
## CV AIC AICc BIC AdjR2   
## 43.0288070 5644.3918197 5644.4300363 5671.2050460 0.3366022

###Time Series Regression with Trend and Season ####Modeling

#Fitting the model  
fit.trend <- tslm(full.ts ~ trend + season)  
summary(fit.trend)

##   
## Call:  
## tslm(formula = full.ts ~ trend + season)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.6233 -1.3294 0.0589 1.3137 5.7640   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12.2707078 0.9930087 12.357 < 2e-16 \*\*\*  
## trend 0.0017889 0.0001235 14.481 < 2e-16 \*\*\*  
## season2 -2.1251723 1.3985075 -1.520 0.128873   
## season3 -0.5660192 1.3985075 -0.405 0.685747   
## season4 0.3995832 1.3985076 0.286 0.775141   
## season5 -0.5216501 1.3985076 -0.373 0.709210   
## season6 -0.2866136 1.3985077 -0.205 0.837652   
## season7 -0.7327676 1.3985077 -0.524 0.600400   
## season8 -0.4808407 1.3985078 -0.344 0.731038   
## season9 0.0120674 1.3985079 0.009 0.993117   
## season10 -0.4326413 1.3985080 -0.309 0.757102   
## season11 0.2358705 1.3985081 0.169 0.866093   
## season12 -0.0758390 1.3985082 -0.054 0.956762   
## season13 0.6701499 1.3985083 0.479 0.631891   
## season14 -0.1895776 1.3985084 -0.136 0.892194   
## season15 -0.5385119 1.3985086 -0.385 0.700259   
## season16 -0.4214072 1.3985087 -0.301 0.763218   
## season17 0.8799509 1.3985089 0.629 0.529333   
## season18 -0.1587945 1.3985091 -0.114 0.909617   
## season19 -0.6447104 1.3985093 -0.461 0.644883   
## season20 -0.7335628 1.3985095 -0.525 0.600005   
## season21 -0.6485659 1.3985097 -0.464 0.642907   
## season22 -0.7416518 1.3985099 -0.530 0.595990   
## season23 1.1980219 1.3985102 0.857 0.391813   
## season24 0.3226388 1.3985104 0.231 0.817585   
## season25 -0.2144540 1.3985107 -0.153 0.878152   
## season26 1.1172929 1.3985109 0.799 0.424496   
## season27 2.6382007 1.3985112 1.886 0.059476 .   
## season28 1.6149833 1.3985115 1.155 0.248406   
## season29 1.6840277 1.3985118 1.204 0.228764   
## season30 1.7638947 1.3985121 1.261 0.207456   
## season31 1.9162833 1.3985124 1.370 0.170869   
## season32 1.8242807 1.3985128 1.304 0.192331   
## season33 2.4248484 1.3985131 1.734 0.083195 .   
## season34 2.6732976 1.3985135 1.912 0.056173 .   
## season35 3.2338464 1.3985138 2.312 0.020926 \*   
## season36 1.6696116 1.3985142 1.194 0.232772   
## season37 3.2222166 1.3985146 2.304 0.021390 \*   
## season38 3.4267435 1.3985150 2.450 0.014416 \*   
## season39 1.9995985 1.3985154 1.430 0.153032   
## season40 1.9664104 1.3985158 1.406 0.159960   
## season41 1.8417278 1.3985163 1.317 0.188116   
## season42 2.4107722 1.3985167 1.724 0.084998 .   
## season43 1.9364833 1.3985171 1.385 0.166409   
## season44 2.6401706 1.3985176 1.888 0.059287 .   
## season45 3.0631437 1.3985181 2.190 0.028694 \*   
## season46 3.2111167 1.3985186 2.296 0.021841 \*   
## season47 3.0678992 1.3985191 2.194 0.028448 \*   
## season48 3.5815865 1.3985196 2.561 0.010558 \*   
## season49 4.6990833 1.3985201 3.360 0.000804 \*\*\*  
## season50 5.9711589 1.3985206 4.270 2.11e-05 \*\*\*  
## season51 6.6000294 1.3985212 4.719 2.64e-06 \*\*\*  
## season52 6.8031106 1.3985217 4.865 1.30e-06 \*\*\*  
## season53 7.0571659 1.3985223 5.046 5.20e-07 \*\*\*  
## season54 6.0649008 1.3985228 4.337 1.57e-05 \*\*\*  
## season55 5.7089453 1.3985234 4.082 4.76e-05 \*\*\*  
## season56 6.1857278 1.3985240 4.423 1.06e-05 \*\*\*  
## season57 6.0278675 1.3985246 4.310 1.76e-05 \*\*\*  
## season58 6.4146500 1.3985252 4.587 4.97e-06 \*\*\*  
## season59 5.9285754 1.3985259 4.239 2.41e-05 \*\*\*  
## season60 5.2363104 1.3985265 3.744 0.000190 \*\*\*  
## season61 6.4892834 1.3985272 4.640 3.86e-06 \*\*\*  
## season62 6.8831355 1.3985278 4.922 9.76e-07 \*\*\*  
## season63 6.8807056 1.3985285 4.920 9.85e-07 \*\*\*  
## season64 7.5714167 1.3985292 5.414 7.43e-08 \*\*\*  
## season65 6.7652231 1.3985299 4.837 1.49e-06 \*\*\*  
## season66 6.8785532 1.3985306 4.918 9.93e-07 \*\*\*  
## season67 7.7547405 1.3985313 5.545 3.61e-08 \*\*\*  
## season68 8.6231898 1.3985320 6.166 9.54e-10 \*\*\*  
## season69 7.5876346 1.3985328 5.425 6.98e-08 \*\*\*  
## season70 8.1606834 1.3985335 5.835 6.89e-09 \*\*\*  
## season71 8.0972279 1.3985343 5.790 8.97e-09 \*\*\*  
## season72 7.2887723 1.3985350 5.212 2.20e-07 \*\*\*  
## season73 7.3094834 1.3985358 5.227 2.03e-07 \*\*\*  
## season74 6.9382501 1.3985366 4.961 8.01e-07 \*\*\*  
## season75 7.3656675 1.3985374 5.267 1.64e-07 \*\*\*  
## season76 9.0082834 1.3985382 6.441 1.71e-10 \*\*\*  
## season77 10.0553041 1.3985390 7.190 1.13e-12 \*\*\*  
## season78 8.7763723 1.3985399 6.275 4.85e-10 \*\*\*  
## season79 11.2110212 1.3985407 8.016 2.55e-15 \*\*\*  
## season80 10.8819612 1.3985416 7.781 1.54e-14 \*\*\*  
## season81 11.3451723 1.3985424 8.112 1.21e-15 \*\*\*  
## season82 11.2642168 1.3985433 8.054 1.90e-15 \*\*\*  
## season83 11.9899279 1.3985442 8.573 < 2e-16 \*\*\*  
## season84 12.4498057 1.3985451 8.902 < 2e-16 \*\*\*  
## season85 13.5832549 1.3985460 9.712 < 2e-16 \*\*\*  
## season86 10.8778946 1.3985469 7.778 1.57e-14 \*\*\*  
## season87 12.1092009 1.3985479 8.658 < 2e-16 \*\*\*  
## season88 11.7776501 1.3985488 8.421 < 2e-16 \*\*\*  
## season89 11.8919724 1.3985498 8.503 < 2e-16 \*\*\*  
## season90 12.8916914 1.3985507 9.218 < 2e-16 \*\*\*  
## season91 13.4534739 1.3985517 9.620 < 2e-16 \*\*\*  
## season92 14.6243041 1.3985527 10.457 < 2e-16 \*\*\*  
## season93 14.1345390 1.3985537 10.107 < 2e-16 \*\*\*  
## season94 13.1591787 1.3985547 9.409 < 2e-16 \*\*\*  
## season95 14.2252470 1.3985557 10.171 < 2e-16 \*\*\*  
## season96 15.0972676 1.3985568 10.795 < 2e-16 \*\*\*  
## season97 14.3996088 1.3985578 10.296 < 2e-16 \*\*\*  
## season98 14.4970629 1.3985589 10.366 < 2e-16 \*\*\*  
## season99 13.8829724 1.3985599 9.927 < 2e-16 \*\*\*  
## season100 14.2161835 1.3985610 10.165 < 2e-16 \*\*\*  
## season101 15.3962994 1.3985621 11.009 < 2e-16 \*\*\*  
## season102 14.2437962 1.3985632 10.185 < 2e-16 \*\*\*  
## season103 14.4941502 1.3985643 10.364 < 2e-16 \*\*\*  
## season104 14.8623613 1.3985654 10.627 < 2e-16 \*\*\*  
## season105 15.7689057 1.3985665 11.275 < 2e-16 \*\*\*  
## season106 15.9587835 1.3985677 11.411 < 2e-16 \*\*\*  
## season107 16.3539391 1.3985688 11.693 < 2e-16 \*\*\*  
## season108 16.2956819 1.3985700 11.652 < 2e-16 \*\*\*  
## season109 16.8716708 1.3985712 12.064 < 2e-16 \*\*\*  
## season110 17.9266280 1.3985723 12.818 < 2e-16 \*\*\*  
## season111 17.2206724 1.3985735 12.313 < 2e-16 \*\*\*  
## season112 17.4230502 1.3985747 12.458 < 2e-16 \*\*\*  
## season113 17.4212613 1.3985760 12.456 < 2e-16 \*\*\*  
## season114 17.3399486 1.3985772 12.398 < 2e-16 \*\*\*  
## season115 17.5087550 1.3985784 12.519 < 2e-16 \*\*\*  
## season116 16.9908947 1.3985797 12.149 < 2e-16 \*\*\*  
## season117 17.8731667 1.4833640 12.049 < 2e-16 \*\*\*  
## season118 18.0878957 1.4833633 12.194 < 2e-16 \*\*\*  
## season119 18.9504612 1.4833626 12.775 < 2e-16 \*\*\*  
## season120 19.3075322 1.4833620 13.016 < 2e-16 \*\*\*  
## season121 19.2945826 1.4833613 13.007 < 2e-16 \*\*\*  
## season122 18.7868414 1.4833607 12.665 < 2e-16 \*\*\*  
## season123 18.8602013 1.4833601 12.715 < 2e-16 \*\*\*  
## season124 19.3048410 1.4833594 13.014 < 2e-16 \*\*\*  
## season125 18.5798378 1.4833588 12.526 < 2e-16 \*\*\*  
## season126 18.2860846 1.4833582 12.327 < 2e-16 \*\*\*  
## season127 19.3954565 1.4833576 13.075 < 2e-16 \*\*\*  
## season128 19.8490247 1.4833571 13.381 < 2e-16 \*\*\*  
## season129 20.2801227 1.4833565 13.672 < 2e-16 \*\*\*  
## season130 19.6819053 1.4833560 13.268 < 2e-16 \*\*\*  
## season131 18.6092831 1.4833554 12.545 < 2e-16 \*\*\*  
## season132 17.3842799 1.4833549 11.720 < 2e-16 \*\*\*  
## season133 16.5342767 1.4833543 11.147 < 2e-16 \*\*\*  
## season134 18.0301244 1.4833538 12.155 < 2e-16 \*\*\*  
## season135 18.3990025 1.4833533 12.404 < 2e-16 \*\*\*  
## season136 19.3503386 1.4833528 13.045 < 2e-16 \*\*\*  
## season137 19.7979545 1.4833524 13.347 < 2e-16 \*\*\*  
## season138 20.4467609 1.4833519 13.784 < 2e-16 \*\*\*  
## season139 20.7408053 1.4833514 13.982 < 2e-16 \*\*\*  
## season140 21.0056831 1.4833510 14.161 < 2e-16 \*\*\*  
## season141 22.0038942 1.4833505 14.834 < 2e-16 \*\*\*  
## season142 22.0916886 1.4833501 14.893 < 2e-16 \*\*\*  
## season143 23.0690664 1.4833497 15.552 < 2e-16 \*\*\*  
## season144 21.0279918 1.4833493 14.176 < 2e-16 \*\*\*  
## season145 21.3631390 1.4833489 14.402 < 2e-16 \*\*\*  
## season146 20.5867976 1.4833485 13.879 < 2e-16 \*\*\*  
## season147 20.6747543 1.4833481 13.938 < 2e-16 \*\*\*  
## season148 21.9803871 1.4833477 14.818 < 2e-16 \*\*\*  
## season149 21.6284224 1.4833474 14.581 < 2e-16 \*\*\*  
## season150 18.7502942 1.4833470 12.641 < 2e-16 \*\*\*  
## season151 18.8078803 1.4833467 12.679 < 2e-16 \*\*\*  
## season152 18.1699307 1.4833463 12.249 < 2e-16 \*\*\*  
## season153 19.6699276 1.4833460 13.261 < 2e-16 \*\*\*  
## season154 20.6400137 1.4833457 13.914 < 2e-16 \*\*\*  
## season155 21.6087605 1.4833454 14.568 < 2e-16 \*\*\*  
## season156 22.6874776 1.4833451 15.295 < 2e-16 \*\*\*  
## season157 21.0508673 1.4833448 14.191 < 2e-16 \*\*\*  
## season158 22.1993760 1.4833446 14.966 < 2e-16 \*\*\*  
## season159 22.2029442 1.4833443 14.968 < 2e-16 \*\*\*  
## season160 23.3981792 1.4833441 15.774 < 2e-16 \*\*\*  
## season161 23.0201998 1.4833438 15.519 < 2e-16 \*\*\*  
## season162 21.3475776 1.4833436 14.392 < 2e-16 \*\*\*  
## season163 20.5408780 1.4833434 13.848 < 2e-16 \*\*\*  
## season164 18.1335832 1.4833432 12.225 < 2e-16 \*\*\*  
## season165 17.3654252 1.4833430 11.707 < 2e-16 \*\*\*  
## season166 18.5064935 1.4833428 12.476 < 2e-16 \*\*\*  
## season167 19.0719665 1.4833426 12.857 < 2e-16 \*\*\*  
## season168 19.2305943 1.4833425 12.964 < 2e-16 \*\*\*  
## season169 19.3444304 1.4833423 13.041 < 2e-16 \*\*\*  
## season170 20.7805879 1.4833422 14.009 < 2e-16 \*\*\*  
## season171 21.3814776 1.4833420 14.414 < 2e-16 \*\*\*  
## season172 19.6973971 1.4833419 13.279 < 2e-16 \*\*\*  
## season173 20.2153998 1.4833418 13.628 < 2e-16 \*\*\*  
## season174 19.4177776 1.4833417 13.091 < 2e-16 \*\*\*  
## season175 20.7465748 1.4833416 13.986 < 2e-16 \*\*\*  
## season176 17.9060153 1.4833415 12.071 < 2e-16 \*\*\*  
## season177 18.8633038 1.4833414 12.717 < 2e-16 \*\*\*  
## season178 19.8490149 1.4833414 13.381 < 2e-16 \*\*\*  
## season179 20.1640415 1.4833413 13.594 < 2e-16 \*\*\*  
## season180 18.4680562 1.4833413 12.450 < 2e-16 \*\*\*  
## season181 19.1969221 1.4833413 12.942 < 2e-16 \*\*\*  
## season182 19.5990023 1.4833412 13.213 < 2e-16 \*\*\*  
## season183 18.5519753 1.4833412 12.507 < 2e-16 \*\*\*  
## season184 16.8823292 1.4833412 11.381 < 2e-16 \*\*\*  
## season185 18.2817804 1.4833412 12.325 < 2e-16 \*\*\*  
## season186 19.8940613 1.4833413 13.412 < 2e-16 \*\*\*  
## season187 17.1028554 1.4833413 11.530 < 2e-16 \*\*\*  
## season188 18.0531499 1.4833413 12.171 < 2e-16 \*\*\*  
## season189 18.8076110 1.4833414 12.679 < 2e-16 \*\*\*  
## season190 17.7662388 1.4833414 11.977 < 2e-16 \*\*\*  
## season191 17.8456999 1.4833415 12.031 < 2e-16 \*\*\*  
## season192 18.2032860 1.4833416 12.272 < 2e-16 \*\*\*  
## season193 16.8421221 1.4833417 11.354 < 2e-16 \*\*\*  
## season194 17.3403332 1.4833418 11.690 < 2e-16 \*\*\*  
## season195 17.7135444 1.4833419 11.942 < 2e-16 \*\*\*  
## season196 18.5711305 1.4833420 12.520 < 2e-16 \*\*\*  
## season197 17.7099666 1.4833422 11.939 < 2e-16 \*\*\*  
## season198 16.3331777 1.4833423 11.011 < 2e-16 \*\*\*  
## season199 15.5501388 1.4833425 10.483 < 2e-16 \*\*\*  
## season200 15.9337666 1.4833426 10.742 < 2e-16 \*\*\*  
## season201 16.1767694 1.4833428 10.906 < 2e-16 \*\*\*  
## season202 17.8603972 1.4833430 12.041 < 2e-16 \*\*\*  
## season203 18.2638166 1.4833432 12.313 < 2e-16 \*\*\*  
## season204 18.5679801 1.4833434 12.518 < 2e-16 \*\*\*  
## season205 17.4054769 1.4833436 11.734 < 2e-16 \*\*\*  
## season206 16.7258607 1.4833438 11.276 < 2e-16 \*\*\*  
## season207 17.4889527 1.4833441 11.790 < 2e-16 \*\*\*  
## season208 16.8264495 1.4833443 11.344 < 2e-16 \*\*\*  
## season209 16.4005535 1.4833446 11.056 < 2e-16 \*\*\*  
## season210 16.1950444 1.4833448 10.918 < 2e-16 \*\*\*  
## season211 16.0599222 1.4833451 10.827 < 2e-16 \*\*\*  
## season212 15.7873000 1.4833454 10.643 < 2e-16 \*\*\*  
## season213 16.5605111 1.4833457 11.164 < 2e-16 \*\*\*  
## season214 16.4786627 1.4833460 11.109 < 2e-16 \*\*\*  
## season215 17.9121416 1.4833463 12.075 < 2e-16 \*\*\*  
## season216 17.6112456 1.4833467 11.873 < 2e-16 \*\*\*  
## season217 16.8504041 1.4833470 11.360 < 2e-16 \*\*\*  
## season218 17.0078166 1.4833474 11.466 < 2e-16 \*\*\*  
## season219 16.0622778 1.4833477 10.828 < 2e-16 \*\*\*  
## season220 16.7294840 1.4833481 11.278 < 2e-16 \*\*\*  
## season221 15.7511545 1.4833485 10.619 < 2e-16 \*\*\*  
## season222 16.3324834 1.4833489 11.011 < 2e-16 \*\*\*  
## season223 16.7259555 1.4833493 11.276 < 2e-16 \*\*\*  
## season224 16.4697024 1.4833497 11.103 < 2e-16 \*\*\*  
## season225 17.3250563 1.4833501 11.680 < 2e-16 \*\*\*  
## season226 16.6551127 1.4833505 11.228 < 2e-16 \*\*\*  
## season227 15.4119548 1.4833510 10.390 < 2e-16 \*\*\*  
## season228 15.6750468 1.4833514 10.567 < 2e-16 \*\*\*  
## season229 16.5619484 1.4833519 11.165 < 2e-16 \*\*\*  
## season230 16.1539095 1.4833524 10.890 < 2e-16 \*\*\*  
## season231 16.9130333 1.4833528 11.402 < 2e-16 \*\*\*  
## season232 17.1964627 1.4833533 11.593 < 2e-16 \*\*\*  
## season233 17.1814463 1.4833538 11.583 < 2e-16 \*\*\*  
## season234 17.3335926 1.4833543 11.685 < 2e-16 \*\*\*  
## season235 17.9472138 1.4833549 12.099 < 2e-16 \*\*\*  
## season236 18.1664500 1.4833554 12.247 < 2e-16 \*\*\*  
## season237 17.5893040 1.4833560 11.858 < 2e-16 \*\*\*  
## season238 18.7353723 1.4833565 12.630 < 2e-16 \*\*\*  
## season239 18.7528889 1.4833571 12.642 < 2e-16 \*\*\*  
## season240 17.2089373 1.4833576 11.601 < 2e-16 \*\*\*  
## season241 18.2000056 1.4833582 12.269 < 2e-16 \*\*\*  
## season242 16.3139905 1.4833588 10.998 < 2e-16 \*\*\*  
## season243 15.9457135 1.4833594 10.750 < 2e-16 \*\*\*  
## season244 16.0165966 1.4833601 10.798 < 2e-16 \*\*\*  
## season245 16.5471358 1.4833607 11.155 < 2e-16 \*\*\*  
## season246 17.2445938 1.4833613 11.625 < 2e-16 \*\*\*  
## season247 16.9290279 1.4833620 11.413 < 2e-16 \*\*\*  
## season248 16.4897270 1.4833626 11.116 < 2e-16 \*\*\*  
## season249 16.6540324 1.4833633 11.227 < 2e-16 \*\*\*  
## season250 17.2297390 1.4833640 11.615 < 2e-16 \*\*\*  
## season251 16.7946167 1.4833647 11.322 < 2e-16 \*\*\*  
## season252 17.5282445 1.4833654 11.817 < 2e-16 \*\*\*  
## season253 16.3528445 1.4833661 11.024 < 2e-16 \*\*\*  
## season254 16.1478352 1.4833668 10.886 < 2e-16 \*\*\*  
## season255 16.9960921 1.4833675 11.458 < 2e-16 \*\*\*  
## season256 17.3633509 1.4833683 11.705 < 2e-16 \*\*\*  
## season257 17.2984668 1.4833690 11.662 < 2e-16 \*\*\*  
## season258 17.5651302 1.4833698 11.841 < 2e-16 \*\*\*  
## season259 17.3850405 1.4833706 11.720 < 2e-16 \*\*\*  
## season260 16.9472668 1.4833713 11.425 < 2e-16 \*\*\*  
## season261 17.0206152 1.4833721 11.474 < 2e-16 \*\*\*  
## season262 17.1999390 1.4833729 11.595 < 2e-16 \*\*\*  
## season263 16.6067811 1.4833737 11.195 < 2e-16 \*\*\*  
## season264 16.0949723 1.4833746 10.850 < 2e-16 \*\*\*  
## season265 16.9179057 1.4833754 11.405 < 2e-16 \*\*\*  
## season266 16.2609713 1.4833762 10.962 < 2e-16 \*\*\*  
## season267 16.1784946 1.4833771 10.907 < 2e-16 \*\*\*  
## season268 15.8483128 1.4833780 10.684 < 2e-16 \*\*\*  
## season269 15.9137755 1.4833788 10.728 < 2e-16 \*\*\*  
## season270 16.1904493 1.4833797 10.915 < 2e-16 \*\*\*  
## season271 16.4363853 1.4833806 11.080 < 2e-16 \*\*\*  
## season272 16.2840144 1.4833815 10.978 < 2e-16 \*\*\*  
## season273 15.9388651 1.4833824 10.745 < 2e-16 \*\*\*  
## season274 16.0231152 1.4833834 10.802 < 2e-16 \*\*\*  
## season275 16.3585287 1.4833843 11.028 < 2e-16 \*\*\*  
## season276 16.7348483 1.4833852 11.282 < 2e-16 \*\*\*  
## season277 15.7799967 1.4833862 10.638 < 2e-16 \*\*\*  
## season278 15.8914557 1.4833872 10.713 < 2e-16 \*\*\*  
## season279 15.6466113 1.4833881 10.548 < 2e-16 \*\*\*  
## season280 16.2597364 1.4833891 10.961 < 2e-16 \*\*\*  
## season281 15.5684502 1.4833901 10.495 < 2e-16 \*\*\*  
## season282 15.6349946 1.4833911 10.540 < 2e-16 \*\*\*  
## season283 15.2370152 1.4833922 10.272 < 2e-16 \*\*\*  
## season284 13.2219922 1.4833932 8.913 < 2e-16 \*\*\*  
## season285 14.0049256 1.4833942 9.441 < 2e-16 \*\*\*  
## season286 14.7040295 1.4833953 9.912 < 2e-16 \*\*\*  
## season287 13.8120290 1.4833963 9.311 < 2e-16 \*\*\*  
## season288 13.1364637 1.4833974 8.856 < 2e-16 \*\*\*  
## season289 12.4939671 1.4833985 8.423 < 2e-16 \*\*\*  
## season290 13.3025288 1.4833996 8.968 < 2e-16 \*\*\*  
## season291 12.6769303 1.4834007 8.546 < 2e-16 \*\*\*  
## season292 12.4685938 1.4834018 8.405 < 2e-16 \*\*\*  
## season293 12.6566198 1.4834029 8.532 < 2e-16 \*\*\*  
## season294 12.7431754 1.4834040 8.590 < 2e-16 \*\*\*  
## season295 12.8994285 1.4834052 8.696 < 2e-16 \*\*\*  
## season296 12.1162002 1.4834063 8.168 7.83e-16 \*\*\*  
## season297 11.6516137 1.4834075 7.855 8.79e-15 \*\*\*  
## season298 11.9480391 1.4834087 8.054 1.90e-15 \*\*\*  
## season299 11.6625002 1.4834098 7.862 8.32e-15 \*\*\*  
## season300 11.2324421 1.4834110 7.572 7.26e-14 \*\*\*  
## season301 10.0169780 1.4834122 6.753 2.25e-11 \*\*\*  
## season302 10.0580132 1.4834135 6.780 1.87e-11 \*\*\*  
## season303 9.9383341 1.4834147 6.700 3.19e-11 \*\*\*  
## season304 9.7667442 1.4834159 6.584 6.81e-11 \*\*\*  
## season305 9.5397751 1.4834172 6.431 1.82e-10 \*\*\*  
## season306 9.2673819 1.4834184 6.247 5.77e-10 \*\*\*  
## season307 9.1325073 1.4834197 6.156 1.01e-09 \*\*\*  
## season308 9.7443169 1.4834210 6.569 7.51e-11 \*\*\*  
## season309 8.3278984 1.4834222 5.614 2.45e-08 \*\*\*  
## season310 8.5857758 1.4834235 5.788 9.08e-09 \*\*\*  
## season311 8.4294793 1.4834248 5.682 1.66e-08 \*\*\*  
## season312 8.2656083 1.4834262 5.572 3.10e-08 \*\*\*  
## season313 8.5556838 1.4834275 5.768 1.02e-08 \*\*\*  
## season314 7.1777503 1.4834288 4.839 1.48e-06 \*\*\*  
## season315 7.0615434 1.4834302 4.760 2.17e-06 \*\*\*  
## season316 6.6824463 1.4834315 4.505 7.29e-06 \*\*\*  
## season317 6.7236656 1.4834329 4.533 6.41e-06 \*\*\*  
## season318 6.8488354 1.4834343 4.617 4.31e-06 \*\*\*  
## season319 6.5470109 1.4834357 4.413 1.11e-05 \*\*\*  
## season320 5.9249336 1.4834371 3.994 6.89e-05 \*\*\*  
## season321 4.9170595 1.4834385 3.315 0.000945 \*\*\*  
## season322 5.2627981 1.4834399 3.548 0.000403 \*\*\*  
## season323 5.0735093 1.4834413 3.420 0.000647 \*\*\*  
## season324 5.6181947 1.4834428 3.787 0.000160 \*\*\*  
## season325 5.4497392 1.4834442 3.674 0.000250 \*\*\*  
## season326 5.5825657 1.4834457 3.763 0.000176 \*\*\*  
## season327 4.8766125 1.4834472 3.287 0.001040 \*\*   
## season328 5.4946443 1.4834486 3.704 0.000222 \*\*\*  
## season329 5.7978468 1.4834501 3.908 9.81e-05 \*\*\*  
## season330 7.2782947 1.4834516 4.906 1.05e-06 \*\*\*  
## season331 6.1687853 1.4834531 4.158 3.43e-05 \*\*\*  
## season332 6.2525579 1.4834547 4.215 2.69e-05 \*\*\*  
## season333 5.7990644 1.4834562 3.909 9.78e-05 \*\*\*  
## season334 5.8364170 1.4834577 3.934 8.82e-05 \*\*\*  
## season335 5.4304916 1.4834593 3.661 0.000262 \*\*\*  
## season336 5.2738114 1.4834609 3.555 0.000392 \*\*\*  
## season337 5.9803559 1.4834624 4.031 5.89e-05 \*\*\*  
## season338 5.4517813 1.4834640 3.675 0.000248 \*\*\*  
## season339 3.9666591 1.4834656 2.674 0.007598 \*\*   
## season340 3.6757468 1.4834672 2.478 0.013355 \*   
## season341 3.4932324 1.4834688 2.355 0.018693 \*   
## season342 3.6827210 1.4834705 2.483 0.013181 \*   
## season343 2.8077178 1.4834721 1.893 0.058641 .   
## season344 4.9886670 1.4834737 3.363 0.000796 \*\*\*  
## season345 3.6188226 1.4834754 2.439 0.014854 \*   
## season346 3.2663392 1.4834771 2.202 0.027867 \*   
## season347 2.6108707 1.4834787 1.760 0.078666 .   
## season348 2.4568091 1.4834804 1.656 0.097959 .   
## season349 2.0769448 1.4834821 1.400 0.161756   
## season350 2.1654337 1.4834838 1.460 0.144634   
## season351 0.0490615 1.4834855 0.033 0.973623   
## season352 0.3701892 1.4834873 0.250 0.802986   
## season353 0.2017337 1.4834890 0.136 0.891855   
## season354 0.1746472 1.4834908 0.118 0.906303   
## season355 0.2835726 1.4834925 0.191 0.848439   
## season356 -0.2182163 1.4834943 -0.147 0.883081   
## season357 -0.4735766 1.4834961 -0.319 0.749607   
## season358 -1.3467941 1.4834979 -0.908 0.364138   
## season359 -0.9586208 1.4834997 -0.646 0.518280   
## season360 -1.4878718 1.4835015 -1.003 0.316087   
## season361 -0.2039464 1.4835033 -0.137 0.890677   
## season362 -0.4351996 1.4835051 -0.293 0.769298   
## season363 -0.0772835 1.4835070 -0.052 0.958461   
## season364 -0.2167536 1.4835088 -0.146 0.883860   
## season365 0.0894933 1.4835107 0.060 0.951906   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.211 on 1210 degrees of freedom  
## Multiple R-squared: 0.9304, Adjusted R-squared: 0.9094   
## F-statistic: 44.3 on 365 and 1210 DF, p-value: < 2.2e-16

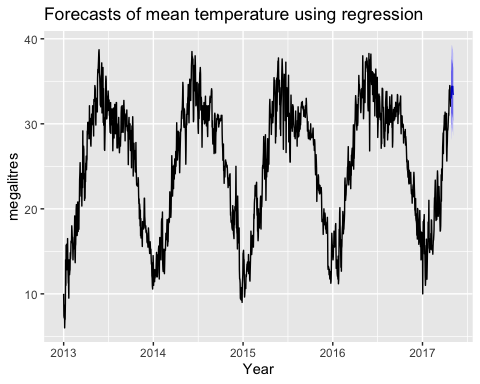
#Visualizing results -- Time plot of actual temperature and predicted temperature  
autoplot(full.ts, series="Data") +  
 autolayer(fitted(fit.trend), series="Fitted") +  
 xlab("Year") + ylab("Megalitres") +  
 ggtitle("Quarterly Beer Production")



CV(fit.trend)

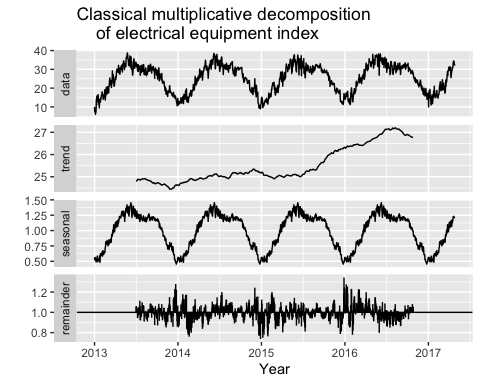
## CV AIC AICc BIC AdjR2   
## 6.3145057 2818.7837437 3042.3863927 4786.8745579 0.9093674

#Visualizing   
fcast.trend <- forecast(fit.trend)  
autoplot(fcast.trend) +  
 ggtitle("Forecasts of mean temperature using regression") +  
 xlab("Year") + ylab("megalitres")

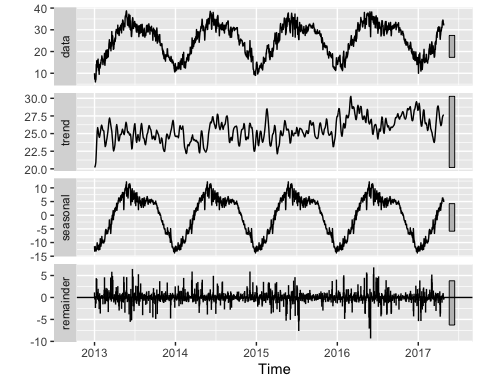


###Forecasting Through Decomposition

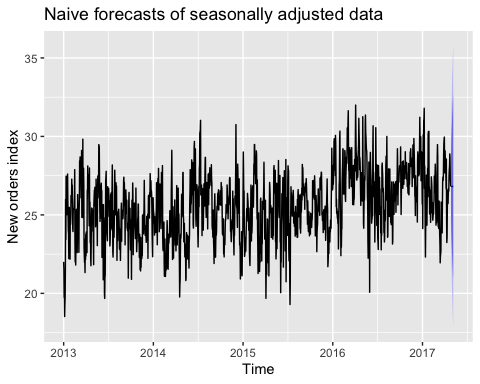
#Classical decomposition  
full.ts %>% decompose(type="multiplicative") %>%  
 autoplot() + xlab("Year") +  
 ggtitle("Classical multiplicative decomposition  
 of electrical equipment index")



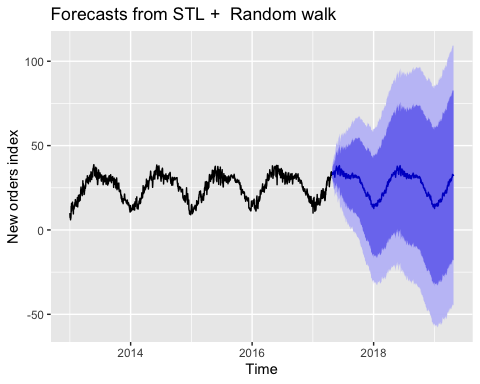
#STL Decomposition  
full.ts %>%  
 stl(t.window=13, s.window="periodic", robust=TRUE) %>%  
 autoplot()



#Modeling through decomposition  
fit.stl <- stl(full.ts, t.window=2, s.window="periodic",  
 robust=TRUE)  
fit.stl %>% seasadj() %>% naive() %>%  
 autoplot() + ylab("New orders index") +  
 ggtitle("Naive forecasts of seasonally adjusted data")



fit.stl %>% forecast(method="naive") %>%  
 autoplot() + ylab("New orders index")

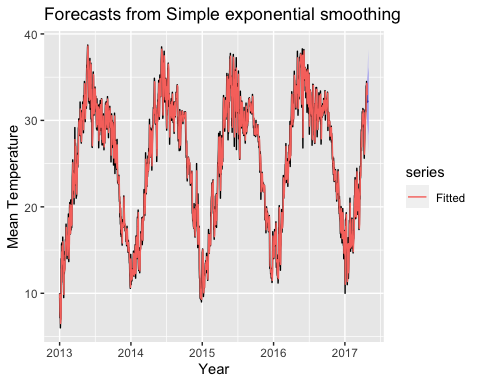


###Exponential Smoothing ####Simple Exponential Smoothing

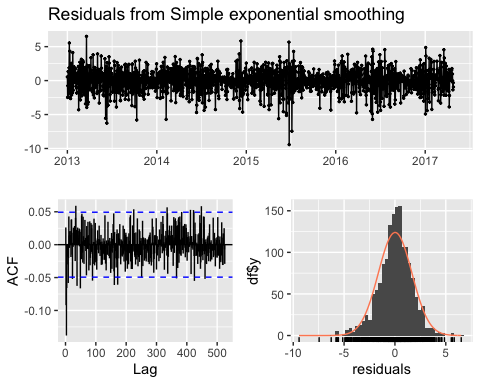
fit.ses <- ses(full.ts, h=5)  
# Accuracy of one-step-ahead training errors  
round(accuracy(fit.ses),2)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.02 1.65 1.24 -0.25 5.44 0.49 0.03

#Visualizing predictions  
autoplot(fit.ses) +  
 autolayer(fitted(fit.ses), series="Fitted") +  
 ylab("Mean Temperature") + xlab("Year")



checkresiduals(ses(full.ts))



##   
## Ljung-Box test  
##   
## data: Residuals from Simple exponential smoothing  
## Q\* = 334.82, df = 313, p-value = 0.1897  
##   
## Model df: 2. Total lags used: 315

####Holt Method

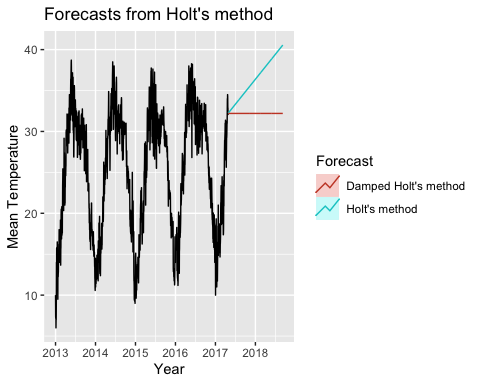
#Fitting  
fit.holt <- holt(full.ts, h=500)  
round(accuracy(fit.holt),2)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0 1.65 1.24 -0.33 5.45 0.49 0.03

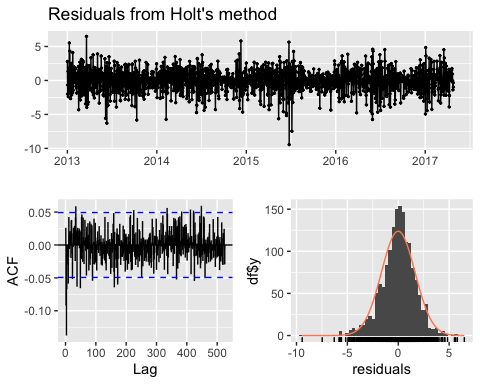
#Dampened Holt  
fit.holt.damp <- holt(full.ts, damped=TRUE, phi = 0.9, h=500)  
round(accuracy(fit.holt.damp), 2)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.02 1.65 1.24 -0.27 5.44 0.49 0.03

#Visualizing Fit  
autoplot(full.ts) +  
 autolayer(fit.holt, series="Holt's method", PI=FALSE) +  
 autolayer(fit.holt.damp, series="Damped Holt's method", PI=FALSE) +  
 ggtitle("Forecasts from Holt's method") + xlab("Year") +  
 ylab("Mean Temperature") +  
 guides(colour=guide\_legend(title="Forecast"))



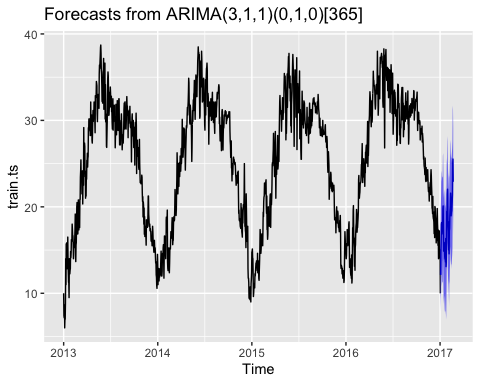
checkresiduals(holt(full.ts))



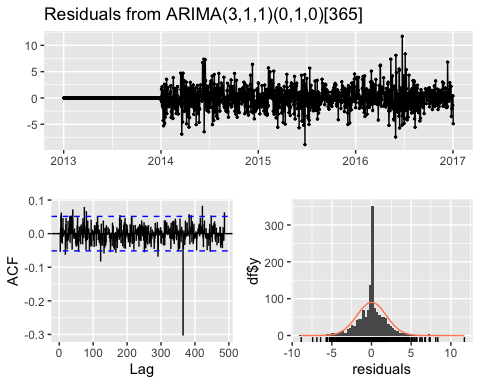
##   
## Ljung-Box test  
##   
## data: Residuals from Holt's method  
## Q\* = 335.09, df = 311, p-value = 0.1663  
##   
## Model df: 4. Total lags used: 315

###ARIMA Fitting

d.arima <- auto.arima(train.ts)  
d.forecast <- forecast(d.arima, level = c(95), h = 50)  
autoplot(d.forecast)



checkresiduals(d.arima)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,1)(0,1,0)[365]  
## Q\* = 367.14, df = 288, p-value = 0.001095  
##   
## Model df: 4. Total lags used: 292

round(accuracy(d.arima),2)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.02 1.85 1.21 -0.28 5.18 0.47 0