SBPA-CA-Q3.R

#Load packages  
 pacman::p\_load(forecast, tseries, fUnitRoots, tidyverse, fastDummies, lmtest, lubridate, tidyverse)  
  
#Set working directory  
 setwd("E:/NUS/EBAC/EBA5002 Business Analytics Practice/2 PA Predictive Analytics/CA Doc/CA/data")  
  
##################################################################################  
#====Amtrak====  
 #Load file  
 Rider = read.csv('AmtrakBig\_CA\_Question-3.csv', check.names = FALSE)  
 head(Rider, n = 4)

## Month Ridership t Season  
## 1 Jan-05 1709 1 Jan  
## 2 Feb-05 1621 2 Feb  
## 3 Mar-05 1973 3 Mar  
## 4 Apr-05 1812 4 Apr

#Data cleaning and preparation  
 class(Rider) #Check for data frame

## [1] "data.frame"

sapply(Rider, class) #Check class of each column

## Month Ridership t Season   
## "factor" "integer" "integer" "factor"

glimpse(Rider)

## Observations: 159  
## Variables: 4  
## $ Month <fct> Jan-05, Feb-05, Mar-05, Apr-05, May-05, Jun-05, Jul-...  
## $ Ridership <int> 1709, 1621, 1973, 1812, 1975, 1862, 1940, 2013, 1596...  
## $ t <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...  
## $ Season <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, No...

summary(Rider)

## Month Ridership t Season   
## Apr-05 : 1 Min. :1361 Min. : 1.00 Feb :14   
## Apr-06 : 1 1st Qu.:1698 1st Qu.: 40.50 Jan :14   
## Apr-07 : 1 Median :1831 Median : 80.00 Mar :14   
## Apr-08 : 1 Mean :1822 Mean : 79.99 Apr :13   
## Apr-09 : 1 3rd Qu.:1967 3rd Qu.:119.50 Aug :13   
## Apr-10 : 1 Max. :2223 Max. :159.00 Dec :13   
## (Other):153 (Other):78

#Check N/A  
 Rider %>%  
 filter(complete.cases(.))

## Month Ridership t Season  
## 1 Jan-05 1709 1 Jan  
## 2 Feb-05 1621 2 Feb  
## 3 Mar-05 1973 3 Mar  
## 4 Apr-05 1812 4 Apr  
## 5 May-05 1975 5 May  
## 6 Jun-05 1862 6 Jun  
## 7 Jul-05 1940 7 Jul  
## 8 Aug-05 2013 8 Aug  
## 9 Sep-05 1596 9 Sep  
## 10 Oct-05 1725 10 Oct  
## 11 Nov-05 1676 11 Nov  
## 12 Dec-05 1814 12 Dec  
## 13 Jan-06 1615 13 Jan  
## 14 Feb-06 1557 14 Feb  
## 15 Mar-06 1891 15 Mar  
## 16 Apr-06 1956 16 Apr  
## 17 May-06 1885 17 May  
## 18 Jun-06 1623 18 Jun  
## 19 Jul-06 1903 19 Jul  
## 20 Aug-06 1997 20 Aug  
## 21 Sep-06 1704 21 Sep  
## 22 Oct-06 1810 22 Oct  
## 23 Nov-06 1862 23 Nov  
## 24 Dec-06 1875 24 Dec  
## 25 Jan-07 1705 25 Jan  
## 26 Feb-07 1619 26 Feb  
## 27 Mar-07 1837 27 Mar  
## 28 Apr-07 1957 28 Apr  
## 29 May-07 1917 29 May  
## 30 Jun-07 1882 30 Jun  
## 31 Jul-07 1933 31 Jul  
## 32 Aug-07 1996 32 Aug  
## 33 Sep-07 1673 33 Sep  
## 34 Oct-07 1753 34 Oct  
## 35 Nov-07 1720 35 Nov  
## 36 Dec-07 1734 36 Dec  
## 37 Jan-08 1563 37 Jan  
## 38 Feb-08 1574 38 Feb  
## 39 Mar-08 1903 39 Mar  
## 40 Apr-08 1834 40 Apr  
## 41 May-08 1831 41 May  
## 42 Jun-08 1776 42 Jun  
## 43 Jul-08 1868 43 Jul  
## 44 Aug-08 1907 44 Aug  
## 45 Sep-08 1686 45 Sep  
## 46 Oct-08 1779 46 Oct  
## 47 Nov-08 1776 47 Nov  
## 48 Dec-08 1783 48 Dec  
## 49 Jan-09 1548 49 Jan  
## 50 Feb-09 1497 50 Feb  
## 51 Mar-09 1798 51 Mar  
## 52 Apr-09 1733 52 Apr  
## 53 May-09 1772 53 May  
## 54 Jun-09 1761 54 Jun  
## 55 Jul-09 1792 55 Jul  
## 56 Aug-09 1875 56 Aug  
## 57 Sep-09 1571 57 Sep  
## 58 Oct-09 1647 58 Oct  
## 59 Nov-09 1673 59 Nov  
## 60 Dec-09 1657 60 Dec  
## 61 Jan-10 1382 61 Jan  
## 62 Feb-10 1361 62 Feb  
## 63 Mar-10 1559 63 Mar  
## 64 Apr-10 1608 64 Apr  
## 65 May-10 1697 65 May  
## 66 Jun-10 1693 66 Jun  
## 67 Jul-10 1836 67 Jul  
## 68 Aug-10 1943 68 Aug  
## 69 Sep-10 1551 69 Sep  
## 70 Oct-10 1687 70 Oct  
## 71 Nov-10 1576 71 Nov  
## 72 Dec-10 1700 72 Dec  
## 73 Jan-11 1397 73 Jan  
## 74 Feb-11 1372 74 Feb  
## 75 Mar-11 1708 75 Mar  
## 76 Apr-11 1655 76 Apr  
## 77 May-11 1763 77 May  
## 78 Jun-11 1776 78 Jun  
## 79 Jul-11 1934 79 Jul  
## 80 Aug-11 2008 80 Aug  
## 81 Sep-11 1616 81 Sep  
## 82 Oct-11 1774 82 Oct  
## 83 Nov-11 1732 83 Nov  
## 84 Dec-11 1797 84 Dec  
## 85 Jan-12 1570 85 Jan  
## 86 Feb-12 1413 86 Feb  
## 87 Mar-12 1755 87 Mar  
## 88 Apr-12 1825 88 Apr  
## 89 May-12 1843 89 May  
## 90 Jun-12 1826 90 Jun  
## 91 Jul-12 1968 91 Jul  
## 92 Aug-12 1922 92 Aug  
## 93 Sep-12 1670 93 Sep  
## 94 Oct-12 1791 94 Oct  
## 95 Nov-12 1817 95 Nov  
## 96 Dec-12 1847 96 Dec  
## 97 Jan-13 1599 97 Jan  
## 98 Feb-13 1549 98 Feb  
## 99 Mar-13 1832 99 Mar  
## 100 Apr-13 1840 100 Apr  
## 101 May-13 1846 101 May  
## 102 Jun-13 1865 102 Jun  
## 103 Jul-13 1966 103 Jul  
## 104 Aug-13 1949 104 Aug  
## 105 Sep-13 1607 105 Sep  
## 106 Oct-13 1804 106 Oct  
## 107 Nov-13 1850 107 Nov  
## 108 Dec-13 1836 108 Dec  
## 109 Jan-14 1542 109 Jan  
## 110 Feb-14 1617 110 Feb  
## 111 Mar-14 1920 111 Mar  
## 112 Apr-14 1971 112 Apr  
## 113 May-14 1992 113 May  
## 114 Jun-14 2010 114 Jun  
## 115 Jul-14 2054 115 Jul  
## 116 Aug-14 2097 116 Aug  
## 117 Sep-14 1824 117 Sep  
## 118 Oct-14 1977 118 Oct  
## 119 Nov-14 1981 119 Nov  
## 120 Dec-14 2000 120 Dec  
## 121 Jan-15 1683 121 Jan  
## 122 Feb-15 1663 122 Feb  
## 123 Mar-15 2008 123 Mar  
## 124 Apr-15 2024 124 Apr  
## 125 May-15 2047 125 May  
## 126 Jun-15 2073 126 Jun  
## 127 Jul-15 2127 126 Jul  
## 128 Aug-15 2203 128 Aug  
## 129 Sep-15 1708 129 Sep  
## 130 Oct-15 1951 130 Oct  
## 131 Nov-15 1974 131 Nov  
## 132 Dec-15 1985 132 Dec  
## 133 Jan-16 1760 133 Jan  
## 134 Feb-16 1771 134 Feb  
## 135 Mar-16 2020 134 Mar  
## 136 Apr-16 2048 136 Apr  
## 137 May-16 2069 137 May  
## 138 Jun-16 1994 138 Jun  
## 139 Jul-16 2075 139 Jul  
## 140 Aug-16 2027 140 Aug  
## 141 Sep-16 1734 141 Sep  
## 142 Oct-16 1917 142 Oct  
## 143 Nov-16 1858 143 Nov  
## 144 Dec-16 1996 144 Dec  
## 145 Jan-17 1778 145 Jan  
## 146 Feb-17 1749 146 Feb  
## 147 Mar-17 2066 147 Mar  
## 148 Apr-17 2099 148 Apr  
## 149 May-17 2105 149 May  
## 150 Jun-17 2130 150 Jun  
## 151 Jul-17 2223 151 Jul  
## 152 Aug-17 2174 152 Aug  
## 153 Sep-17 1931 153 Sep  
## 154 Oct-17 2121 154 Oct  
## 155 Nov-17 2076 155 Nov  
## 156 Dec-17 2141 156 Dec  
## 157 Jan-18 1832 157 Jan  
## 158 Feb-18 1838 158 Feb  
## 159 Mar-18 2132 159 Mar

#Edit error in t  
 Rider <- Rider %>%  
 mutate(t = 1:nrow(Rider)) %>%  
 mutate(Ridership = (Ridership))  
   
 #Check if type is Time Series  
 is.ts(Rider$Ridership)

## [1] FALSE

#Select window  
 # Rider <- Rider %>%  
 # filter(t > 60)  
   
 #Convert to Time Series (frequency = 12 months)  
 RiderTS = ts(Rider$Ridership, frequency = 12, start = c(2005, 1))  
 RiderTS

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2005 1709 1621 1973 1812 1975 1862 1940 2013 1596 1725 1676 1814  
## 2006 1615 1557 1891 1956 1885 1623 1903 1997 1704 1810 1862 1875  
## 2007 1705 1619 1837 1957 1917 1882 1933 1996 1673 1753 1720 1734  
## 2008 1563 1574 1903 1834 1831 1776 1868 1907 1686 1779 1776 1783  
## 2009 1548 1497 1798 1733 1772 1761 1792 1875 1571 1647 1673 1657  
## 2010 1382 1361 1559 1608 1697 1693 1836 1943 1551 1687 1576 1700  
## 2011 1397 1372 1708 1655 1763 1776 1934 2008 1616 1774 1732 1797  
## 2012 1570 1413 1755 1825 1843 1826 1968 1922 1670 1791 1817 1847  
## 2013 1599 1549 1832 1840 1846 1865 1966 1949 1607 1804 1850 1836  
## 2014 1542 1617 1920 1971 1992 2010 2054 2097 1824 1977 1981 2000  
## 2015 1683 1663 2008 2024 2047 2073 2127 2203 1708 1951 1974 1985  
## 2016 1760 1771 2020 2048 2069 1994 2075 2027 1734 1917 1858 1996  
## 2017 1778 1749 2066 2099 2105 2130 2223 2174 1931 2121 2076 2141  
## 2018 1832 1838 2132

is.ts(RiderTS)

## [1] TRUE

#Check Time Series  
 start(RiderTS) #Start year

## [1] 2005 1

end(RiderTS) #End year

## [1] 2018 3

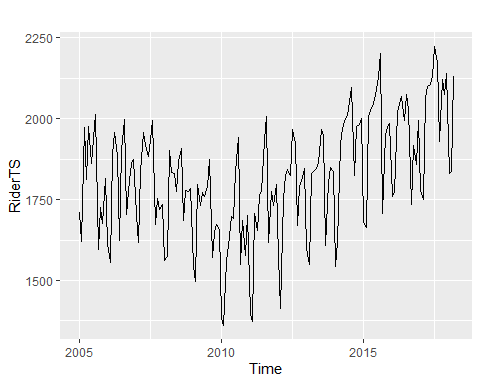
frequency(RiderTS) #period

## [1] 12

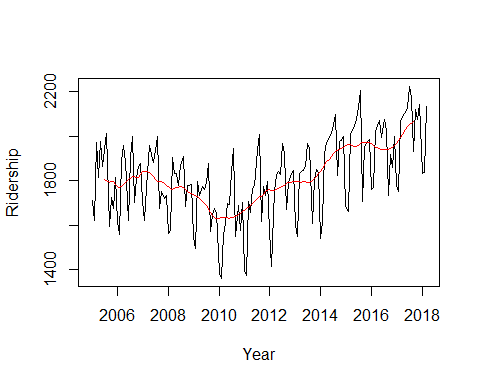
cycle(RiderTS) #Print 2D table of time period

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2005 1 2 3 4 5 6 7 8 9 10 11 12  
## 2006 1 2 3 4 5 6 7 8 9 10 11 12  
## 2007 1 2 3 4 5 6 7 8 9 10 11 12  
## 2008 1 2 3 4 5 6 7 8 9 10 11 12  
## 2009 1 2 3 4 5 6 7 8 9 10 11 12  
## 2010 1 2 3 4 5 6 7 8 9 10 11 12  
## 2011 1 2 3 4 5 6 7 8 9 10 11 12  
## 2012 1 2 3 4 5 6 7 8 9 10 11 12  
## 2013 1 2 3 4 5 6 7 8 9 10 11 12  
## 2014 1 2 3 4 5 6 7 8 9 10 11 12  
## 2015 1 2 3 4 5 6 7 8 9 10 11 12  
## 2016 1 2 3 4 5 6 7 8 9 10 11 12  
## 2017 1 2 3 4 5 6 7 8 9 10 11 12  
## 2018 1 2 3

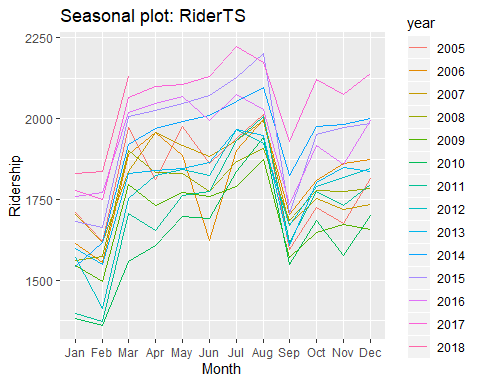
#Plot Time Series  
 autoplot(RiderTS)



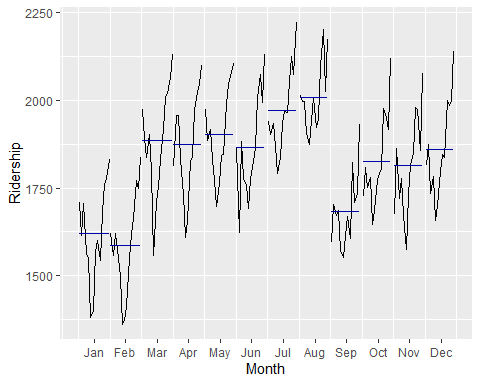
ts.plot(RiderTS, ma(RiderTS, order = 12), lty = c(1:1), col = c('black','red'), xlab = "Year", ylab = "Ridership") #Plot moving average with month = 12



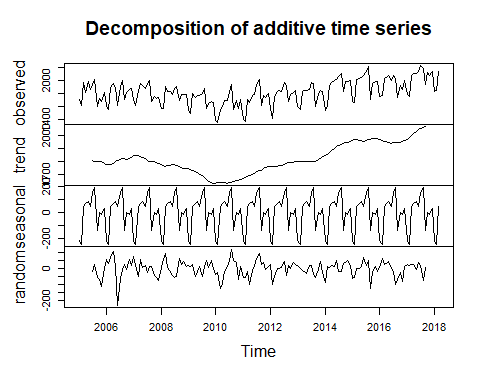
#Plot decomposition  
 ggseasonplot(RiderTS, ylab= "Ridership") #Plot to see if there is increase year-on-year



ggmonthplot(RiderTS, ylab = "Ridership") #Plot to see if there is increase year-on-year



plot(decompose(RiderTS)) #Plot decomposition #Default: additive



#====Forecasting====  
 #Set training and test data based on time  
 forecast\_time = 6 #Split training and test data wrt training data  
 RiderTS\_test = subset(RiderTS, start = length(RiderTS) - forecast\_time + 1)  
 RiderTS\_trg = subset(RiderTS, end = length(RiderTS) - (forecast\_time))  
 cycle(RiderTS\_trg)

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2005 1 2 3 4 5 6 7 8 9 10 11 12  
## 2006 1 2 3 4 5 6 7 8 9 10 11 12  
## 2007 1 2 3 4 5 6 7 8 9 10 11 12  
## 2008 1 2 3 4 5 6 7 8 9 10 11 12  
## 2009 1 2 3 4 5 6 7 8 9 10 11 12  
## 2010 1 2 3 4 5 6 7 8 9 10 11 12  
## 2011 1 2 3 4 5 6 7 8 9 10 11 12  
## 2012 1 2 3 4 5 6 7 8 9 10 11 12  
## 2013 1 2 3 4 5 6 7 8 9 10 11 12  
## 2014 1 2 3 4 5 6 7 8 9 10 11 12  
## 2015 1 2 3 4 5 6 7 8 9 10 11 12  
## 2016 1 2 3 4 5 6 7 8 9 10 11 12  
## 2017 1 2 3 4 5 6 7 8 9

cycle(RiderTS\_test)

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2017 10 11 12  
## 2018 1 2 3

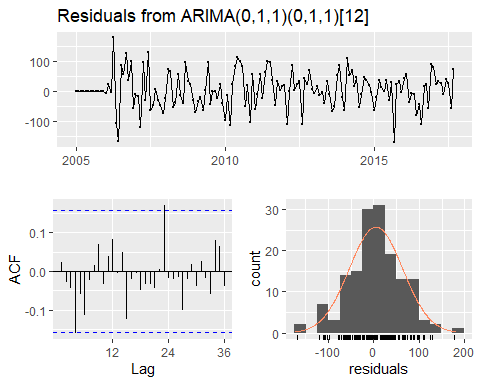
#----Auto-ARIMA----  
 RiderTSaa = auto.arima(RiderTS\_trg, stepwise = FALSE, parallel = TRUE)  
 coeftest(RiderTSaa)

##   
## z test of coefficients:  
##   
## Estimate Std. Error z value Pr(>|z|)   
## ma1 -0.451025 0.091201 -4.9454 7.6e-07 \*\*\*  
## sma1 -0.745009 0.085067 -8.7579 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(RiderTSaa)

## Series: RiderTS\_trg   
## ARIMA(0,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ma1 sma1  
## -0.4510 -0.7450  
## s.e. 0.0912 0.0851  
##   
## sigma^2 estimated as 3833: log likelihood=-780.21  
## AIC=1566.42 AICc=1566.6 BIC=1575.25  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4.538592 58.80038 44.7849 0.1838869 2.510556 0.5635081  
## ACF1  
## Training set 0.02439759

checkresiduals(RiderTSaa)

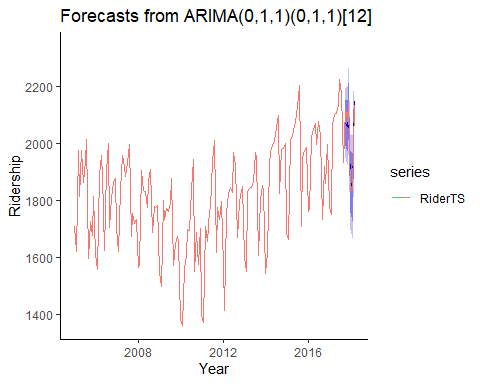


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)(0,1,1)[12]  
## Q\* = 19.337, df = 22, p-value = 0.6245  
##   
## Model df: 2. Total lags used: 24

RiderTSaa\_test = forecast(RiderTSaa, h = forecast\_time)  
 #RiderAcc <- accuracy(RiderTSaa\_test) #Record accuracy of model for comparison  
 summary(RiderTSaa\_test)

##   
## Forecast method: ARIMA(0,1,1)(0,1,1)[12]  
##   
## Model Information:  
## Series: RiderTS\_trg   
## ARIMA(0,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ma1 sma1  
## -0.4510 -0.7450  
## s.e. 0.0912 0.0851  
##   
## sigma^2 estimated as 3833: log likelihood=-780.21  
## AIC=1566.42 AICc=1566.6 BIC=1575.25  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4.538592 58.80038 44.7849 0.1838869 2.510556 0.5635081  
## ACF1  
## Training set 0.02439759  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Oct 2017 2070.318 1990.958 2149.679 1948.947 2191.690  
## Nov 2017 2059.826 1969.293 2150.358 1921.368 2198.284  
## Dec 2017 2112.899 2012.429 2213.370 1959.243 2266.555  
## Jan 2018 1864.718 1755.209 1974.227 1697.239 2032.197  
## Feb 2018 1844.943 1727.087 1962.798 1664.698 2025.187  
## Mar 2018 2146.003 2020.354 2271.652 1953.840 2338.167

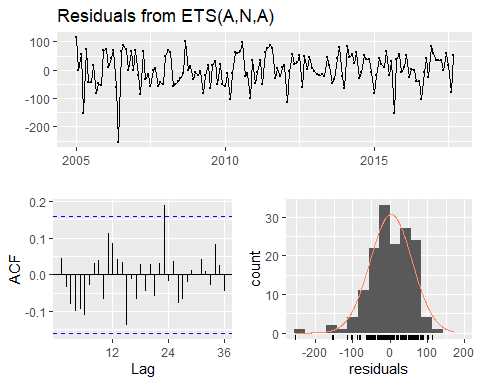
RiderTS\_testsummary <- tibble(x = "Auto-ARIMA", y = accuracy(RiderTSaa\_test))  
  
 RiderTSaa %>%  
 forecast(h = forecast\_time) %>%  
 autoplot(xlab = "Year", ylab = "Ridership") + autolayer(RiderTS) +   
 theme\_classic()



#----ETS----  
 RiderTSets = ets(RiderTS\_trg)  
 summary(RiderTSets)

## ETS(A,N,A)   
##   
## Call:  
## ets(y = RiderTS\_trg)   
##   
## Smoothing parameters:  
## alpha = 0.5608   
## gamma = 1e-04   
##   
## Initial states:  
## l = 1806.6294   
## s = 28.5224 -16.0454 0.0981 -138.9838 189.1324 152.0783  
## 49.17 84.6901 62.5704 48.1823 -247.6577 -211.7572  
##   
## sigma: 58.9789  
##   
## AIC AICc BIC   
## 2032.592 2036.095 2078.048   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 2.810915 56.21584 44.2314 0.075623 2.478808 0.5565436  
## ACF1  
## Training set 0.04339839

checkresiduals(RiderTSets)

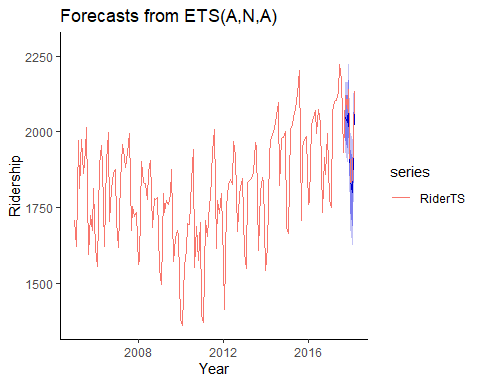


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,A)  
## Q\* = 23.739, df = 10, p-value = 0.008325  
##   
## Model df: 14. Total lags used: 24

RiderTSets\_test = forecast(RiderTSets, h = forecast\_time)  
 summary(RiderTSets\_test)

##   
## Forecast method: ETS(A,N,A)  
##   
## Model Information:  
## ETS(A,N,A)   
##   
## Call:  
## ets(y = RiderTS\_trg)   
##   
## Smoothing parameters:  
## alpha = 0.5608   
## gamma = 1e-04   
##   
## Initial states:  
## l = 1806.6294   
## s = 28.5224 -16.0454 0.0981 -138.9838 189.1324 152.0783  
## 49.17 84.6901 62.5704 48.1823 -247.6577 -211.7572  
##   
## sigma: 58.9789  
##   
## AIC AICc BIC   
## 2032.592 2036.095 2078.048   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 2.810915 56.21584 44.2314 0.075623 2.478808 0.5565436  
## ACF1  
## Training set 0.04339839  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Oct 2017 2047.902 1972.317 2123.487 1932.305 2163.499  
## Nov 2017 2031.766 1945.108 2118.424 1899.234 2164.298  
## Dec 2017 2076.329 1979.860 2172.798 1928.793 2223.865  
## Jan 2018 1836.060 1730.690 1941.430 1674.910 1997.209  
## Feb 2018 1800.150 1686.574 1913.726 1626.451 1973.849  
## Mar 2018 2095.994 1974.767 2217.221 1910.593 2281.395

RiderTSets %>%  
 forecast(h = forecast\_time) %>%  
 autoplot(xlab = "Year", ylab = "Ridership") + autolayer(RiderTS) +   
 theme\_classic()



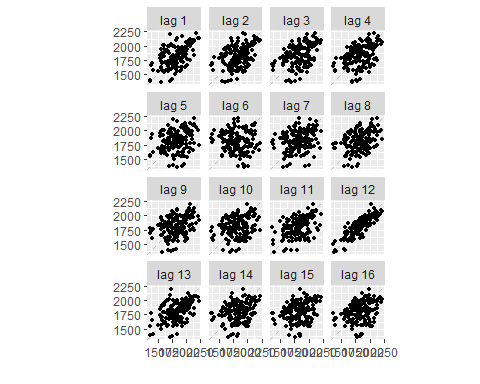
RiderTS\_testsummary <- add\_row(RiderTS\_testsummary, x = "ETS", y = accuracy(RiderTSets\_test))  
   
 #RiderAcc <- rbind(RiderAcc, accuracy(RiderTSets\_test)) #Record accuracy of model for comparison  
  
   
 #----ARIMA----  
 #Augmented Dickey-Fuller Test p < 0.05 for stationarity (H0, p>0.05): non-stationarity)  
 adfTest(RiderTS)

##   
## Title:  
## Augmented Dickey-Fuller Test  
##   
## Test Results:  
## PARAMETER:  
## Lag Order: 1  
## STATISTIC:  
## Dickey-Fuller: -0.1713  
## P VALUE:  
## 0.5598   
##   
## Description:  
## Sun Sep 08 12:26:55 2019 by user: KE

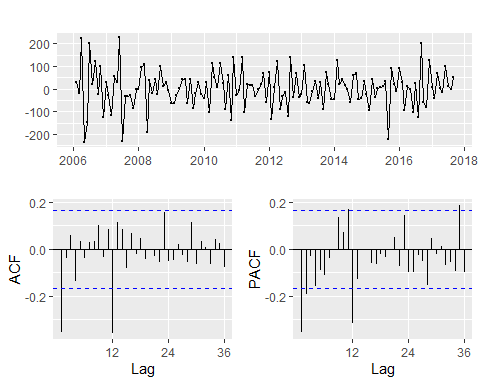
#Lag plot  
 RiderTS\_lag = window(RiderTS, start = 1991) #Select window

## Warning in window.default(x, ...): 'start' value not changed

gglagplot(RiderTS\_lag, do.lines = FALSE, colour = FALSE) #Plot points instead of lines and colours



#Differencing  
 nonseasonal\_diff = 1  
 seasonal\_diff = 1  
   
 RiderTS\_trg %>%   
 diff(differences = nonseasonal\_diff) %>% #non-seasonal differencing  
 diff(lag = 12, differences = seasonal\_diff) %>% #seasonal differencing  
 ggtsdisplay() #Display ACF, PACF



RiderTS\_trg %>%   
 diff(differences = nonseasonal\_diff) %>% #non-seasonal differencing  
 diff(lag = 12, differences = seasonal\_diff) %>% #seasonal differencing  
 adfTest() #Display ADF test statistics

## Warning in adfTest(.): p-value smaller than printed p-value

##   
## Title:  
## Augmented Dickey-Fuller Test  
##   
## Test Results:  
## PARAMETER:  
## Lag Order: 1  
## STATISTIC:  
## Dickey-Fuller: -11.6302  
## P VALUE:  
## 0.01   
##   
## Description:  
## Sun Sep 08 12:26:56 2019 by user: KE

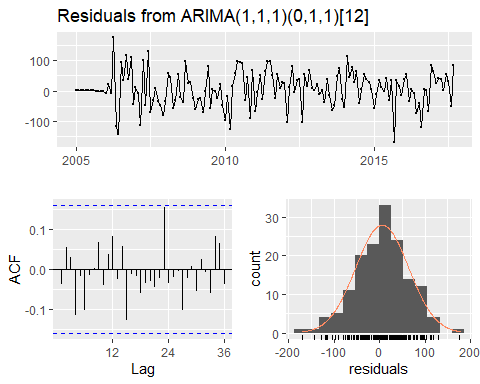
#Create ARIMA model  
 RiderTSa = Arima(RiderTS\_trg, order = c(1, 1, 1), seasonal = c(0, 1, 1))  
 coeftest(RiderTSa)

##   
## z test of coefficients:  
##   
## Estimate Std. Error z value Pr(>|z|)   
## ar1 0.321950 0.170798 1.8850 0.05943 .   
## ma1 -0.735275 0.130111 -5.6511 1.594e-08 \*\*\*  
## sma1 -0.759310 0.084279 -9.0095 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(RiderTSa)

## Series: RiderTS\_trg   
## ARIMA(1,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ar1 ma1 sma1  
## 0.3219 -0.7353 -0.7593  
## s.e. 0.1708 0.1301 0.0843  
##   
## sigma^2 estimated as 3785: log likelihood=-779.19  
## AIC=1566.37 AICc=1566.67 BIC=1578.14  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 5.892061 58.21653 44.64026 0.2572698 2.495726 0.5616881  
## ACF1  
## Training set -0.03674693

checkresiduals(RiderTSa)

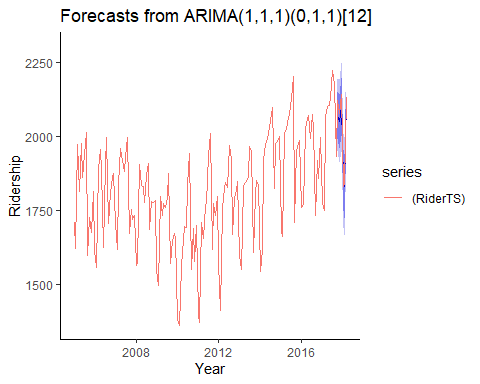


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1)(0,1,1)[12]  
## Q\* = 16.57, df = 21, p-value = 0.7368  
##   
## Model df: 3. Total lags used: 24

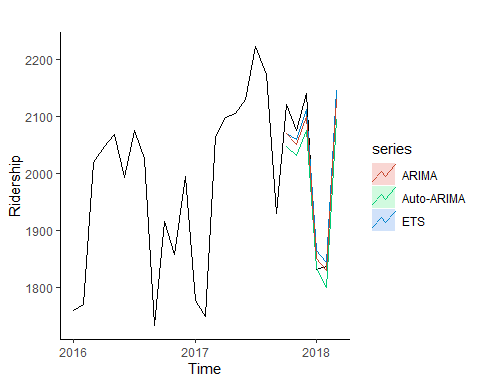
RiderTSa\_test = forecast(RiderTSa, h = forecast\_time)  
 #RiderTSa\_test = Arima(RiderTS\_test, model = RiderTSa, h = forecast\_time) #Same thing as above  
 summary(RiderTSa\_test)

##   
## Forecast method: ARIMA(1,1,1)(0,1,1)[12]  
##   
## Model Information:  
## Series: RiderTS\_trg   
## ARIMA(1,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ar1 ma1 sma1  
## 0.3219 -0.7353 -0.7593  
## s.e. 0.1708 0.1301 0.0843  
##   
## sigma^2 estimated as 3785: log likelihood=-779.19  
## AIC=1566.37 AICc=1566.67 BIC=1578.14  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 5.892061 58.21653 44.64026 0.2572698 2.495726 0.5616881  
## ACF1  
## Training set -0.03674693  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Oct 2017 2070.965 1992.098 2149.831 1950.349 2191.580  
## Nov 2017 2051.023 1959.586 2142.460 1911.182 2190.864  
## Dec 2017 2100.111 2001.925 2198.297 1949.949 2250.274  
## Jan 2018 1851.140 1747.749 1954.530 1693.018 2009.262  
## Feb 2018 1830.094 1722.069 1938.120 1664.884 1995.305  
## Mar 2018 2131.015 2018.642 2243.388 1959.156 2302.874

RiderTSa %>%  
 forecast(h = forecast\_time) %>%  
 autoplot(xlab = "Year", ylab = "Ridership") + autolayer((RiderTS)) +   
 theme\_classic()



RiderTS\_testsummary <- add\_row(RiderTS\_testsummary, x = "ARIMA", y = accuracy(RiderTSa\_test))  
   
 #RiderAcc <- rbind(RiderAcc, accuracy(RiderTSa\_test)) #Record accuracy of model for comparison  
   
   
 #Plot all models on 1 graph  
 autoplot(window(RiderTS, start = 2016), ylab = "Ridership") +   
 autolayer(RiderTSaa\_test, series = "ETS", PI = FALSE) +  
 autolayer(RiderTSets\_test, series = "Auto-ARIMA", PI = FALSE) +  
 autolayer(RiderTSa\_test, series = "ARIMA", PI = FALSE) +  
 theme\_classic()



#Compare accuracy of models  
 #RiderAcc = as.data.frame(RiderAcc)  
 #RiderAcc$Model <- c("Auto-ARIMA", "ETS", "ARIMA")  
   
 #RiderAcc %>%  
 # select(Model, everything()) #Place t as first column  
  
 RiderTS\_testsummary

## # A tibble: 3 x 2  
## x y[,"ME"] [,"RMSE"] [,"MAE"] [,"MPE"] [,"MAPE"] [,"MASE"] [,"ACF1"]  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Auto-~ 4.54 58.8 44.8 0.184 2.51 0.564 0.0244  
## 2 ETS 2.81 56.2 44.2 0.0756 2.48 0.557 0.0434  
## 3 ARIMA 5.89 58.2 44.6 0.257 2.50 0.562 -0.0367