Demand Prediction using Time Series (Monthly)

11/10/2020

Monthly Data

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2

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

```r
library(ggplot2)
library(zoo)
```

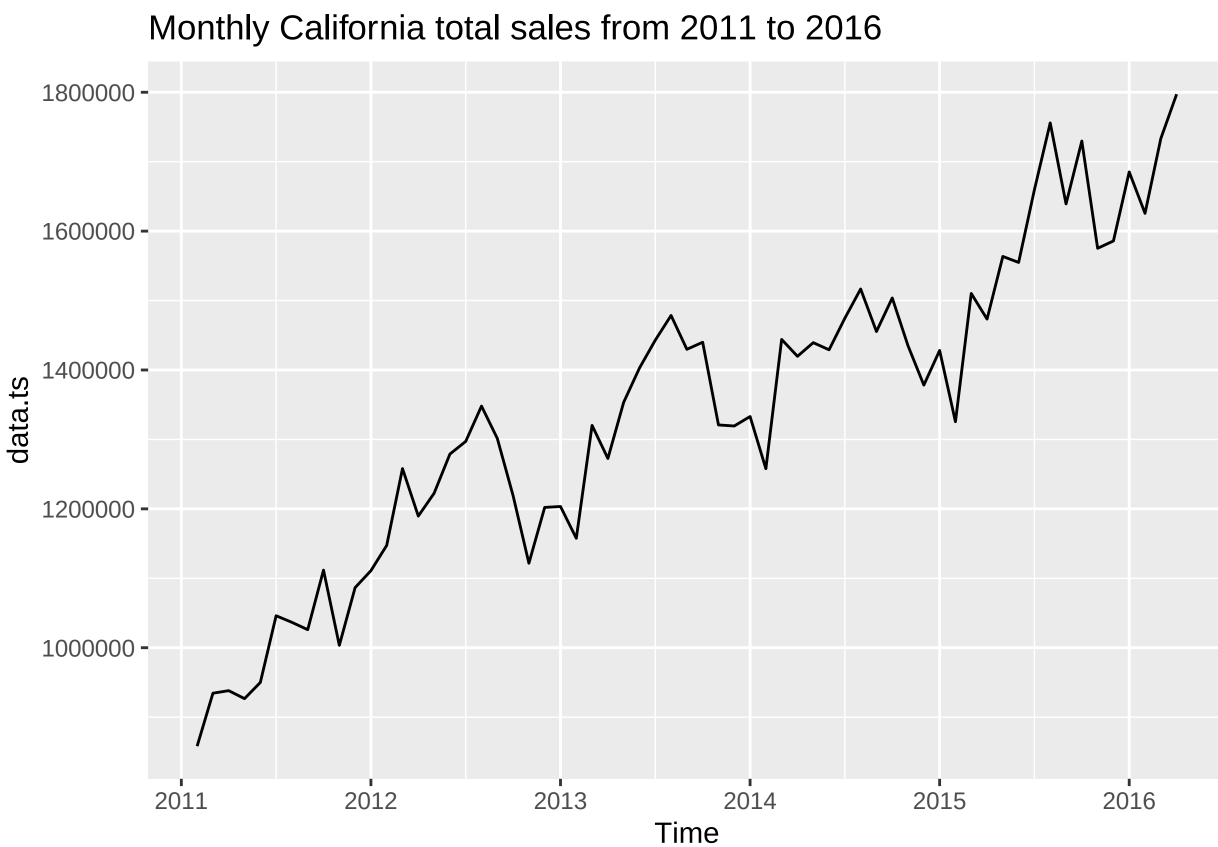
```
## Warning: package 'zoo' was built under R version 3.6.2

##
## Attaching package: 'zoo'

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

```r
data = read.csv("monthly_new.csv", header=T)
head(data)
```

```
##   X   month CA_FOODS CA_HOBBIES CA_HOUSEHOLD TX_FOODS TX_HOBBIES
## 1 0 2011-01  58314.4   10711.10     27581.79  39086.1    7529.58
## 2 1 2011-02 510610.3   97368.76    250162.46 353840.1   56818.03
## 3 2 2011-03 540558.0  109869.51    284058.22 375910.2   61082.42
## 4 3 2011-04 525853.9  117021.07    295246.10 367134.3   68456.65
## 5 4 2011-05 517492.4  120215.81    288950.89 367056.2   72988.33
## 6 5 2011-06 540954.7  118140.65    290853.84 402972.1   68466.57
##   TX_HOUSEHOLD  WI_FOODS WI_HOBBIES WI_HOUSEHOLD Total.Revenue      YM
## 1     17866.28  32050.81    6821.14     18366.89      218328.1 2011-01
## 2    156046.49 297903.28   56844.18    146284.15     1925877.7 2011-02
## 3    166587.36 295634.87   62099.80    157100.04     2052900.5 2011-03
## 4    166757.40 274054.64   65446.56    147443.24     2027413.9 2011-04
## 5    166918.76 260542.03   66472.27    142351.32     2002988.0 2011-05
## 6    159045.94 279551.70   63408.30    148324.17     2071718.0 2011-06
##   event event_weekend Cultural National Religious Sporting  CA_total
## 1     0             0        0        0         0        0  96607.29
## 2     3             1        1        1         0        1 858141.49
## 3     4             1        1        0         3        0 934485.77
## 4     2             1        0        0         2        0 938121.11
## 5     4             1        2        1         0        1 926659.08
## 6     2             2        1        0         0        1 949949.22
##    TX_total  WI_total CA_FOODS_pct CA_HOBBIES_pct CA_HOUSEHOLD_pct
## 1  64481.96  57238.84    0.6036232      0.1108726        0.2855042
## 2 566704.60 501031.61    0.5950187      0.1134647        0.2915166
## 3 603579.99 514834.71    0.5784551      0.1175722        0.3039728
## 4 602348.37 486944.44    0.5605395      0.1247398        0.3147207
## 5 606963.34 469365.62    0.5584496      0.1297304        0.3118201
```

```
## 6 630484.63 491284.17    0.5694565      0.1243652        0.3061783
##   unemployment_rate real_gdp personal_income
## 1              12.1  2086244         1721993
## 2              12.0  2086244         1721993
## 3              11.9  2086244         1721993
## 4              11.8  2092379         1728856
## 5              11.8  2092379         1728856
## 6              11.8  2092379         1728856
```

```r
data = data[2:(dim(data)[1]-1),]

data.ts = ts(data$CA_total, start=c(2011,2), frequency=12)
autoplot(data.ts, main="Monthly California total sales from 2011 to 2016")
```



Train-test split

```r
n = length(data.ts)
nValid=12
nTrain = n - nValid
train.ts = window(data.ts, start=c(2011,2), end=c(2011, nTrain), frequency=12)
test.ts = window(data.ts, start=c(2011, nTrain+1), frequency=12)
```

Naive Forecast

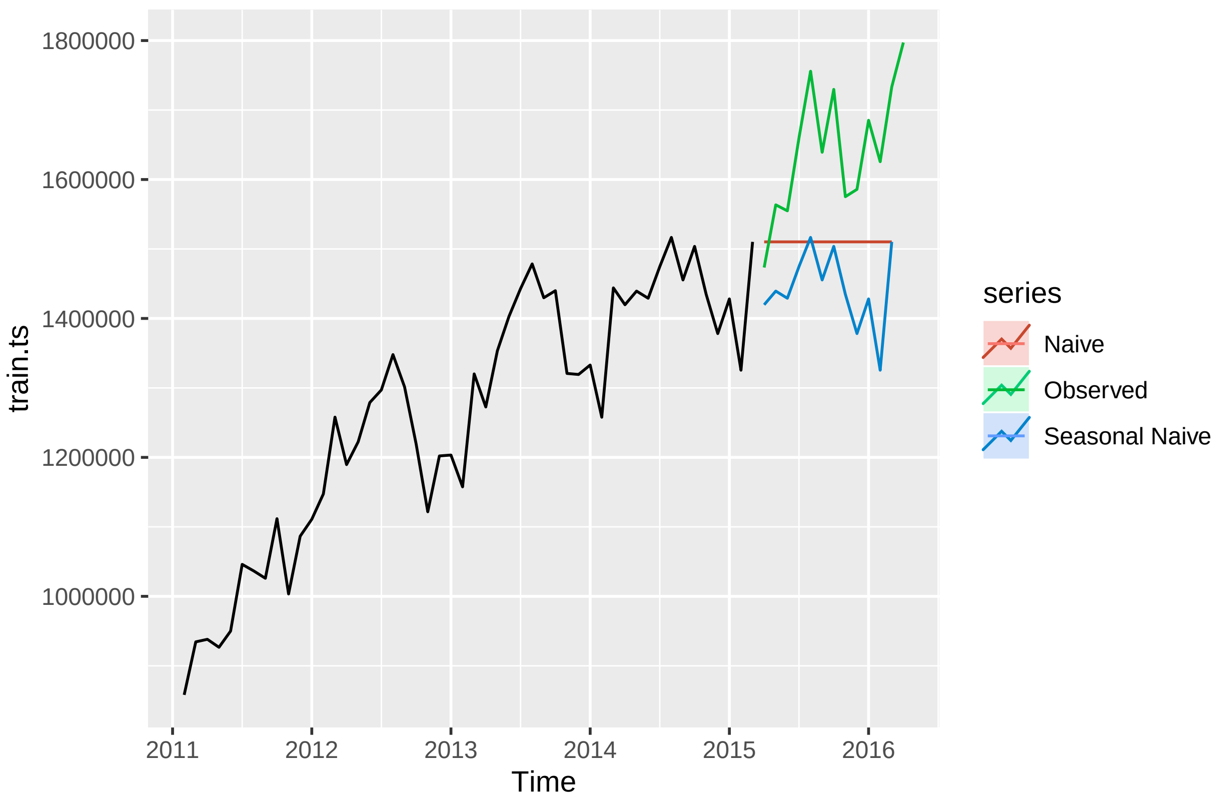
```r
model = naive(train.ts, h=12)
accuracy(model, test.ts)
```

```
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set  13307.68  72684.07  58067.79 0.9864657 4.589060 0.4173773
## Test set     121582.18 146459.96 127734.32 7.2160320 7.633606 0.9181235
##                    ACF1 Theil's U
## Training set -0.3443775        NA
## Test set      0.1582257  1.616549
```

```
model_season = snaive(train.ts, h=frequency(train.ts))
accuracy(model_season, test.ts)
```

```
##                    ME     RMSE      MAE      MPE     MAPE     MASE
## Training set 139125.4 165788.1 139125.4 10.61982 10.61982 1.000000
## Test set     188854.2 199849.7 188854.2 11.44441 11.44441 1.357438
##                   ACF1 Theil's U
## Training set 0.7981618        NA
## Test set     0.4366385  2.208834
```

```
autoplot(train.ts) +
  autolayer(model, series="Naive", PI=FALSE) +
  autolayer(model_season, series="Seasonal Naive", PI=FALSE) +
  autolayer(test.ts, series="Observed")
```



Moving Average

```
w = 12
pred = rep(NA, nValid)
for(i in 1:nValid){
  ntrain.temp = n-nValid+(i-1)
  train.temp = window(data.ts, start=c(2011,2), end=c(2011, ntrain.temp), frquency=12)
```

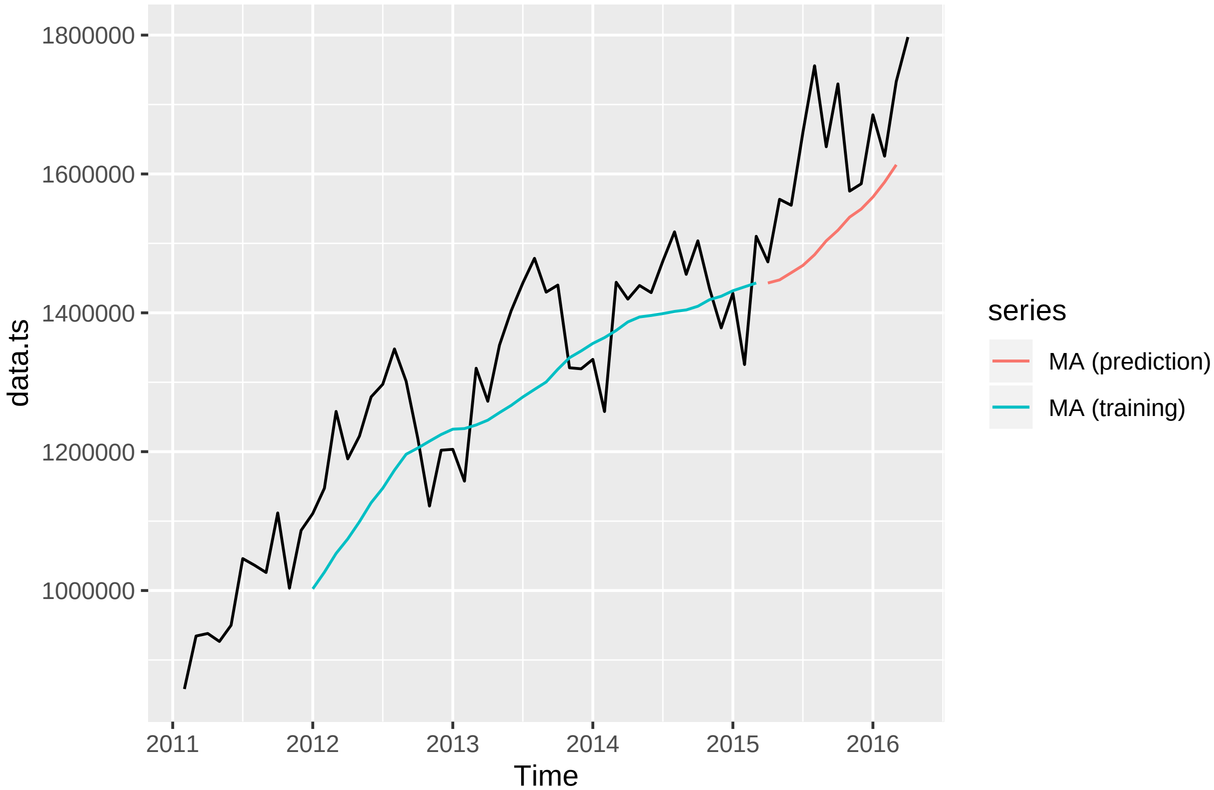
```
  model = rollmean(train.temp, k=w, align="right")
  last.ma = tail(model, 1)
  pred[i] = last.ma
}
pred = ts(pred, start=c(2011, n-nValid+1), frequency=12)
accuracy(pred, test.ts)
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 116963.1 138451.9 116963.1 7.013921 7.013921 0.2015369  1.540191
```

```
accuracy(rollmean(train.ts, k=12, align="right"), train.ts)
```

```
##                ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 58007.2 101401.8 86296.02 4.231504 6.508833 0.6047513  1.322551
```

```
autoplot(data.ts, col="black")+
  autolayer(pred, series="MA (prediction)")+
  autolayer(rollmean(train.ts, k=12, align="right"), series="MA (training)")
```



Exponential Smoothing

```
model = ets(train.ts, model="ZZZ")
summary(model)
```

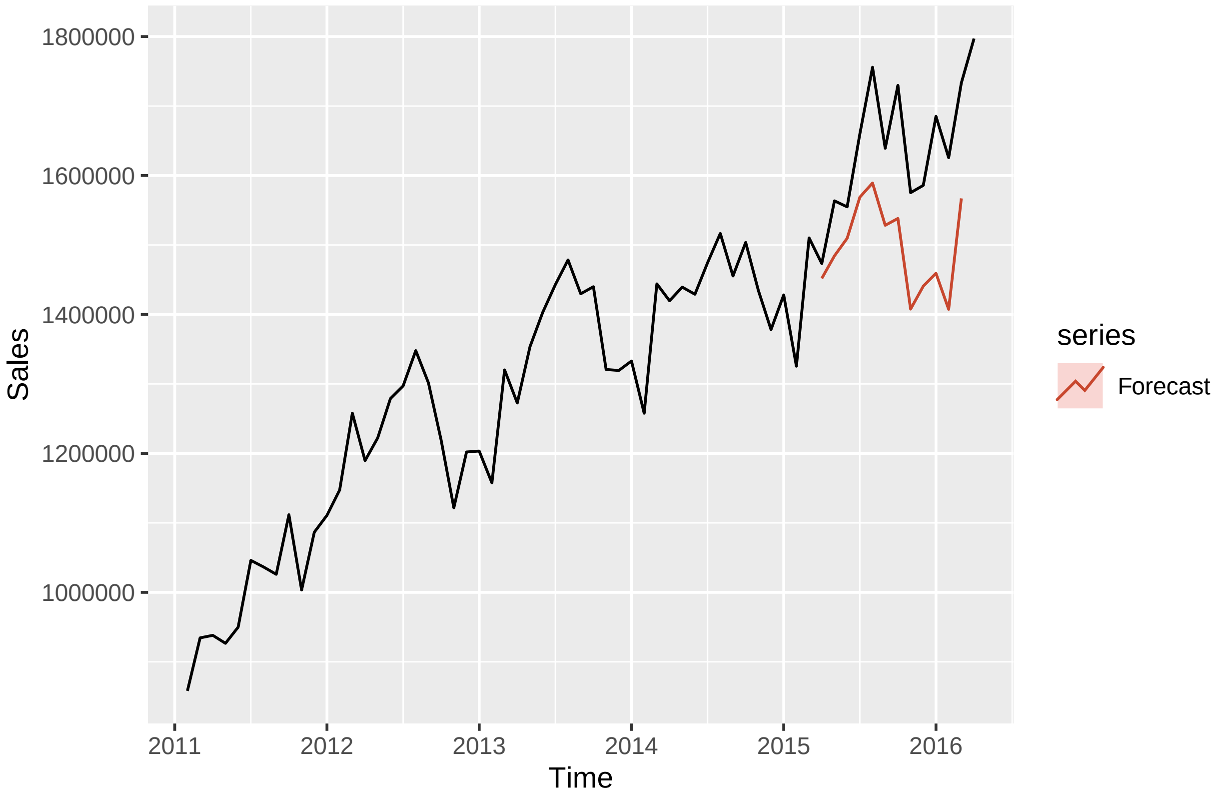
```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = train.ts, model = "ZZZ")
##
```

```
##   Smoothing parameters:
##     alpha = 0.7402
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9756
##
##   Initial states:
##     l = 926725.0738
##     b = 19586.7743
##     s = 0.9641 0.9548 0.9358 1.0258 1.0225 1.0669
##            1.0569 1.0206 1.0072 0.9889 1.0294 0.9272
##
##   sigma:  0.035
##
##      AIC     AICc      BIC
## 1279.320 1301.384 1313.736
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE     MASE
## Training set -465.4573 34531.93 27092.73 -0.09949108 2.218036 0.194736
##                     ACF1
## Training set -0.05486422
```

```r
pred = forecast(model, h=nValid, level=0)
accuracy(pred, test.ts)
```

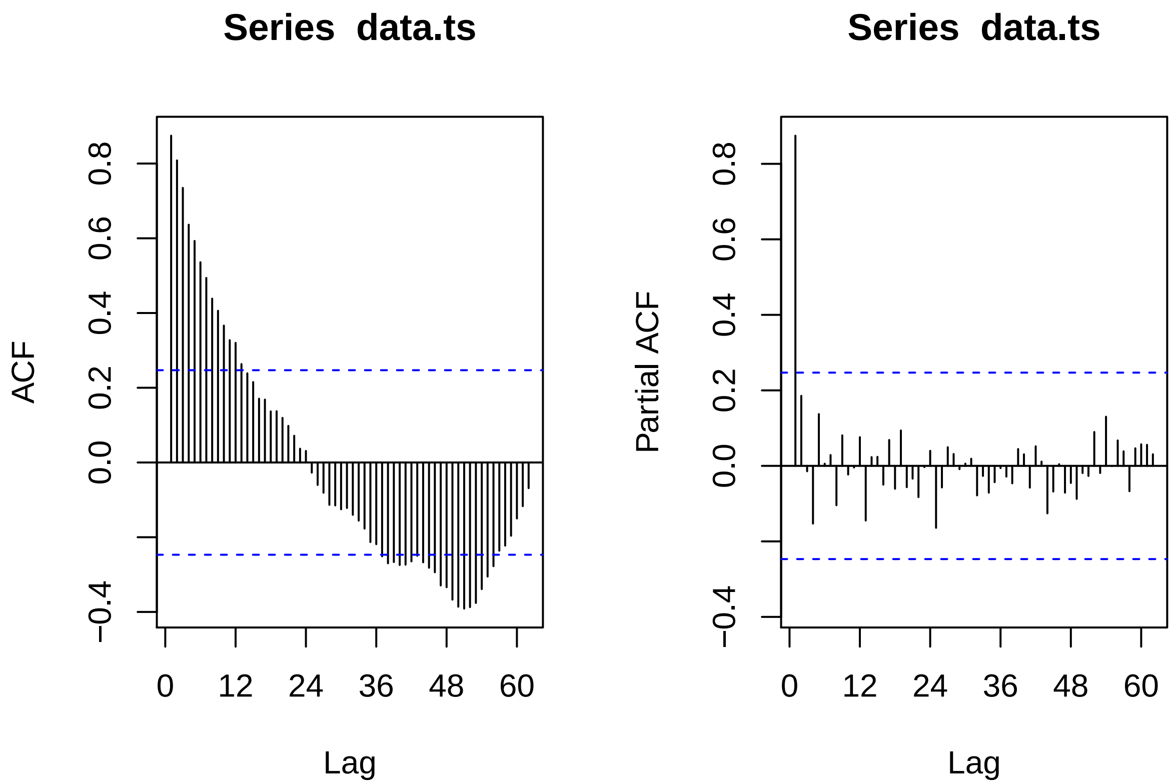
```
##                       ME      RMSE       MAE         MPE     MAPE
## Training set   -465.4573  34531.93  27092.73 -0.09949108 2.218036
## Test set     135745.3452 149790.64 135745.35  8.20381082 8.203811
##                   MASE        ACF1 Theil's U
## Training set 0.1947360 -0.05486422        NA
## Test set     0.9757048  0.51879746  1.651781
```

```r
autoplot(data.ts, ylab="Sales")+
  autolayer(pred, series="Forecast")
```

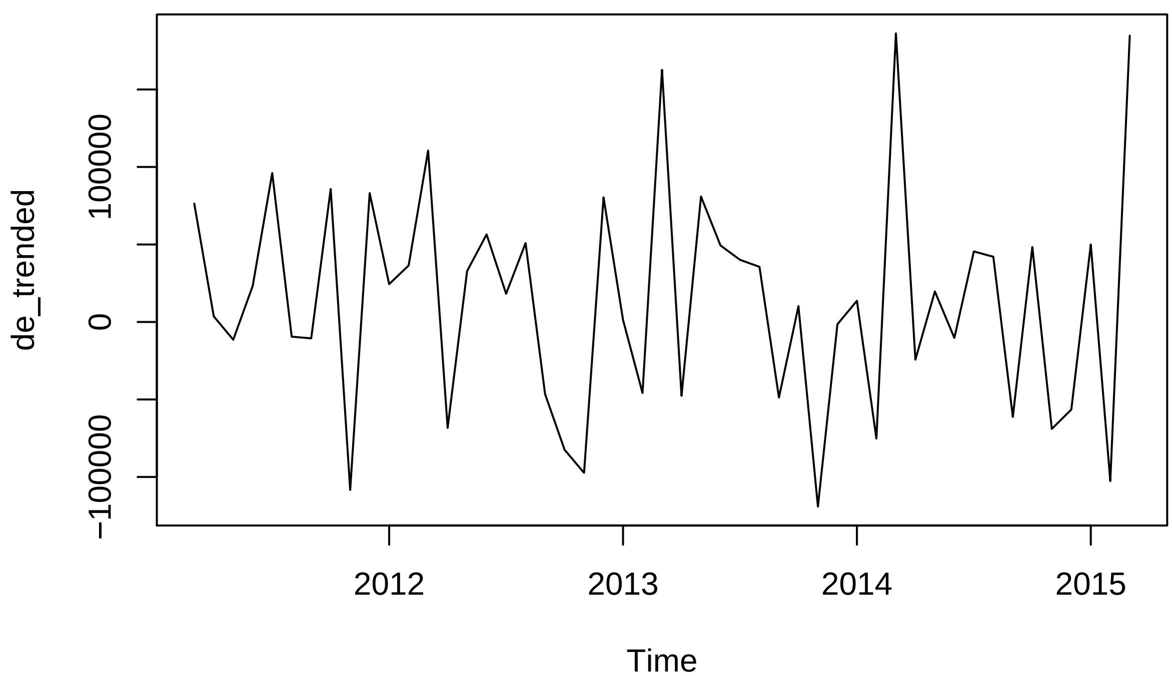


ARIMA

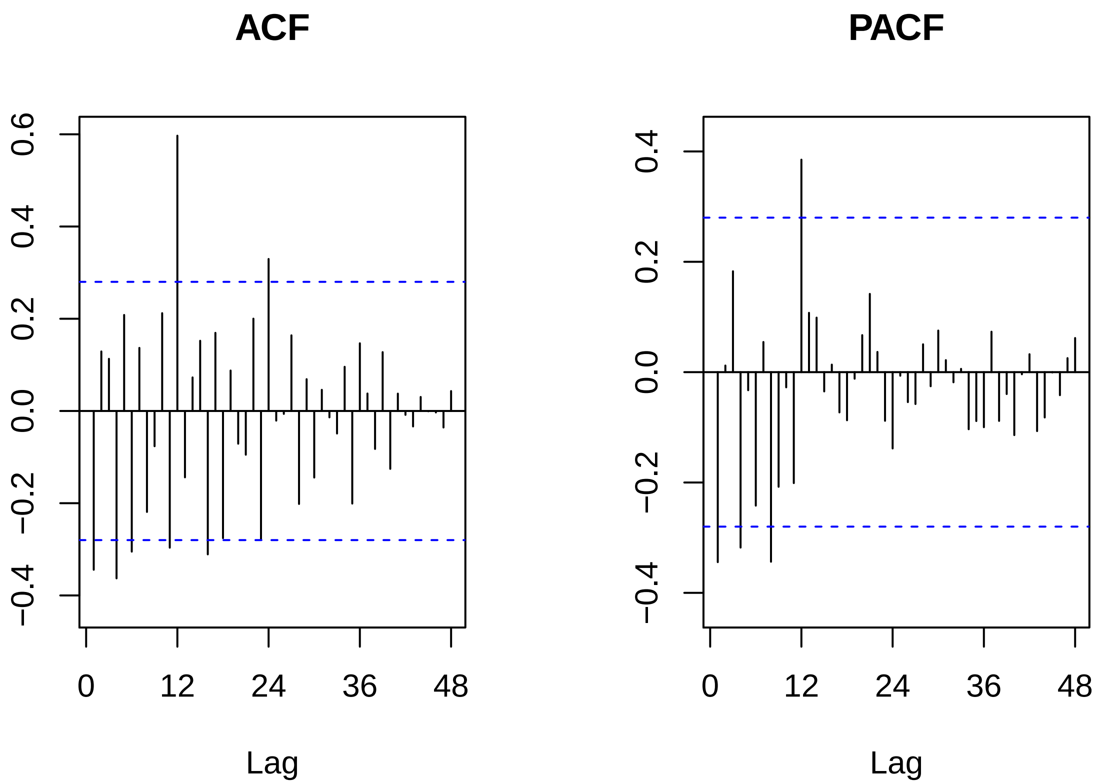
```
par(mfrow=c(1,2))
Acf(data.ts, lag.max=210)
Pacf(data.ts, lag.max=210)
```




```
par(mfrow=c(1,1))

data.diff = diff(train.ts, lag=1)
plot(data.diff, ylab="de_trended")
```




```
par(mfrow=c(1,2))
Acf(data.diff, lag.max=210, main="ACF", ylab="")
Pacf(data.diff, lag.max=210, main="PACF", ylab="")
```




```
par(mfrow=c(1,1))

model = Arima(train.ts, order=c(1,1,1)) # order = c(p(AR),d(diff),q(MA))
summary(model)
```

```
## Series: train.ts
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.4380  0.0851
## s.e.   0.2574  0.2501
##
## sigma^2 estimated as 4.863e+09:  log likelihood=-615.05
## AIC=1236.1   AICc=1236.63   BIC=1241.77
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 15949.76 67607.96 55130.97 1.223197 4.382449 0.3962681
##                     ACF1
## Training set -0.03713552
```

```
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                      ME      RMSE       MAE       MPE      MAPE      MASE
## Training set   15949.76  67607.96  55130.97  1.223197  4.382449 0.3962681
## Test set      170365.90 187745.00 170365.90 10.223250 10.223250 1.2245490
##                     ACF1 Theil's U
## Training set -0.03713552        NA
```

```
## Test set      0.17124703  2.077532
```

```r
model = Arima(train.ts, order=c(2,1,2)) # order = c(p(AR),d(diff),q(MA))
summary(model)
```

```
## Series: train.ts
## ARIMA(2,1,2)
##
## Coefficients:
##           ar1      ar2      ma1     ma2
##       -0.3425  -0.3359  -0.0018  0.5985
## s.e.   0.8839   0.3474   0.8131  0.2345
##
## sigma^2 estimated as 4.616e+09:  log likelihood=-612.89
## AIC=1235.78   AICc=1237.17   BIC=1245.24
##
## Training set error measures:
##                   ME     RMSE     MAE      MPE     MAPE      MASE
## Training set 13155.5 64454.08 55395.8 1.010249 4.401867 0.3981716
##                     ACF1
## Training set -0.02054496
```

```r
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                    ME      RMSE      MAE      MPE     MAPE      MASE
## Training set  13155.5  64454.08  55395.8 1.010249 4.401867 0.3981716
## Test set     149436.4 168250.56 149436.4 8.945530 8.945530 1.0741125
##                     ACF1 Theil's U
## Training set -0.02054496        NA
## Test set      0.15497382  1.860359
```

```r
model = Arima(train.ts, order=c(3,1,3)) # order = c(p(AR),d(diff),q(MA))
summary(model)
```

```
## Series: train.ts
## ARIMA(3,1,3)
##
## Coefficients:
##           ar1      ar2     ar3     ma1     ma2      ma3
##       -1.2133  -0.1622  0.4758  1.3351  0.1670  -0.4631
## s.e.   1.2359   2.0954  1.2123  1.2840  2.3102   1.2840
##
## sigma^2 estimated as 3.218e+09:  log likelihood=-605.92
## AIC=1225.85   AICc=1228.58   BIC=1239.09
##
## Training set error measures:
##                    ME    RMSE      MAE       MPE     MAPE      MASE
## Training set 11567.31 52608.7 39995.72 0.9117436 3.175355 0.2874796
##                     ACF1
## Training set -0.06100199
```

```r
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                     ME     RMSE       MAE       MPE     MAPE      MASE
## Training set  11567.31  52608.7  39995.72 0.9117436 3.175355 0.2874796
```

```
## Test set     151322.62 169381.7 151322.62 9.0741235 9.074124 1.0876705
##                     ACF1 Theil's U
## Training set -0.06100199        NA
## Test set      0.22643482   1.87738
```

```r
model = Arima(train.ts, order=c(1,1,1), seasonal=list(order=c(1,1,0), period=12)) # order = c(p(AR),d(d
summary(model)
```

```
## Series: train.ts
## ARIMA(1,1,1)(1,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1
##       0.1557  -0.3597  -0.1398
## s.e.  0.5482   0.5059   0.2146
##
## sigma^2 estimated as 2.689e+09:  log likelihood=-452.76
## AIC=913.52   AICc=914.77   BIC=919.96
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -6423.416 42763.82 28034.58 -0.5638533 2.141298 0.2015058
##                     ACF1
## Training set -0.02150067
```

```r
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  -6423.416  42763.82  28034.58 -0.5638533 2.141298 0.2015058
## Test set     113315.252 129897.05 115095.21  6.8114882 6.932302 0.8272766
##                     ACF1 Theil's U
## Training set -0.02150067        NA
## Test set      0.45532139  1.434944
```

```r
model = Arima(train.ts, order=c(2,1,1), seasonal=list(order=c(1,1,0), period=12)) # order = c(p(AR),d(d
summary(model)
```

```
## Series: train.ts
## ARIMA(2,1,1)(1,1,0)[12]
##
## Coefficients:
##           ar1      ar2     ma1     sar1
##       -0.6504  -0.3252  0.4990  -0.0589
## s.e.   0.3250   0.1589  0.3153   0.2266
##
## sigma^2 estimated as 2.602e+09:  log likelihood=-451.59
## AIC=913.18   AICc=915.11   BIC=921.23
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -5935.039 41441.23 28270.33 -0.5320367 2.154317 0.2032003
##                     ACF1
## Training set -0.01052402
```

```r
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  -5935.039  41441.23  28270.33 -0.5320367 2.154317 0.2032003
## Test set     110612.090 128654.88 115214.23  6.6371675 6.949536 0.8281321
##                     ACF1 Theil's U
## Training set -0.01052402        NA
## Test set      0.42942135   1.41866
```

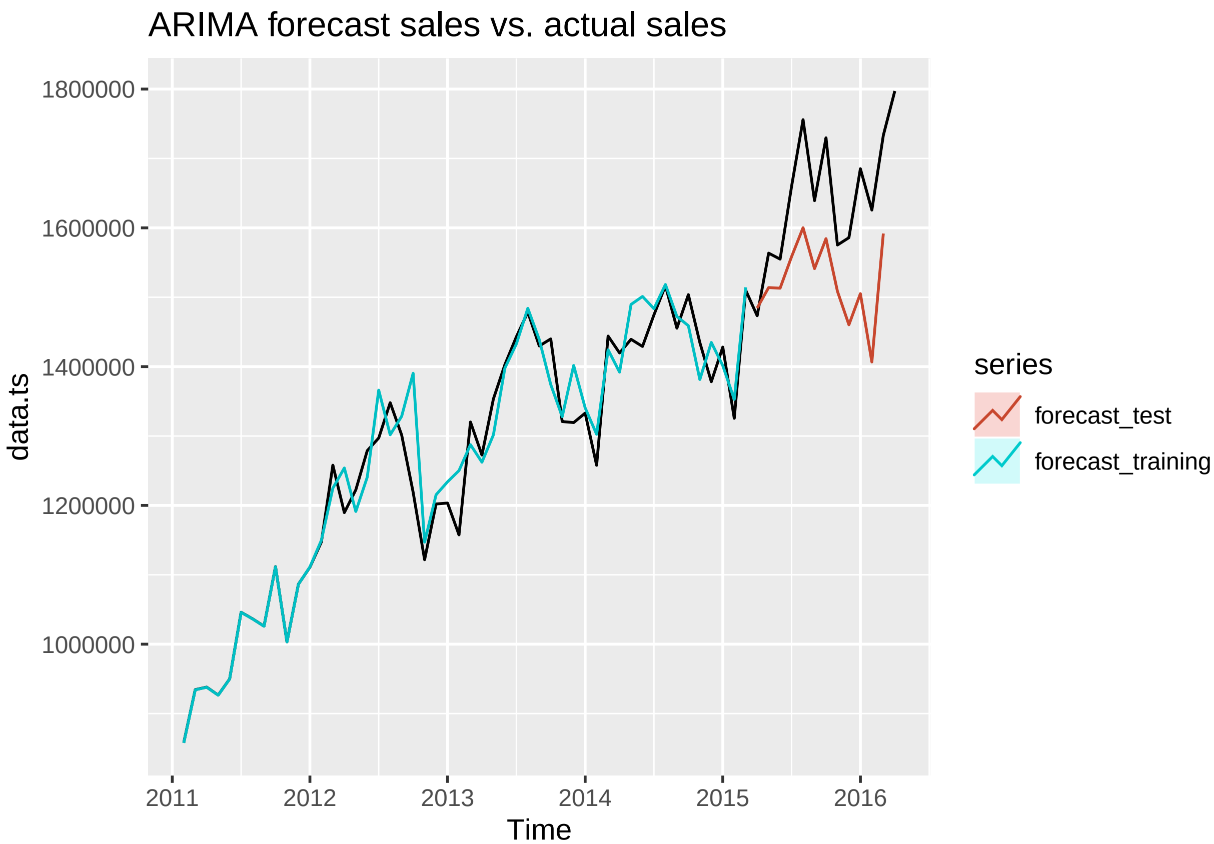
```r
model = Arima(train.ts, order=c(1,1,2), seasonal=list(order=c(1,1,0), period=12)) # order = c(p(AR),d(d
summary(model)
```

```
## Series: train.ts
## ARIMA(1,1,2)(1,1,0)[12]
##
## Coefficients:
##          ar1      ma1     ma2     sar1
##       0.8369  -1.0501  0.1467  -0.1442
## s.e.  0.3544   0.4080  0.2212   0.2145
##
## sigma^2 estimated as 2.76e+09:  log likelihood=-452.73
## AIC=915.45   AICc=917.39   BIC=923.51
##
## Training set error measures:
##                     ME  RMSE      MAE        MPE     MAPE      MASE
## Training set -7836.274 42681 27784.42 -0.6668532 2.124193 0.1997077
##                     ACF1
## Training set -0.03343257
```

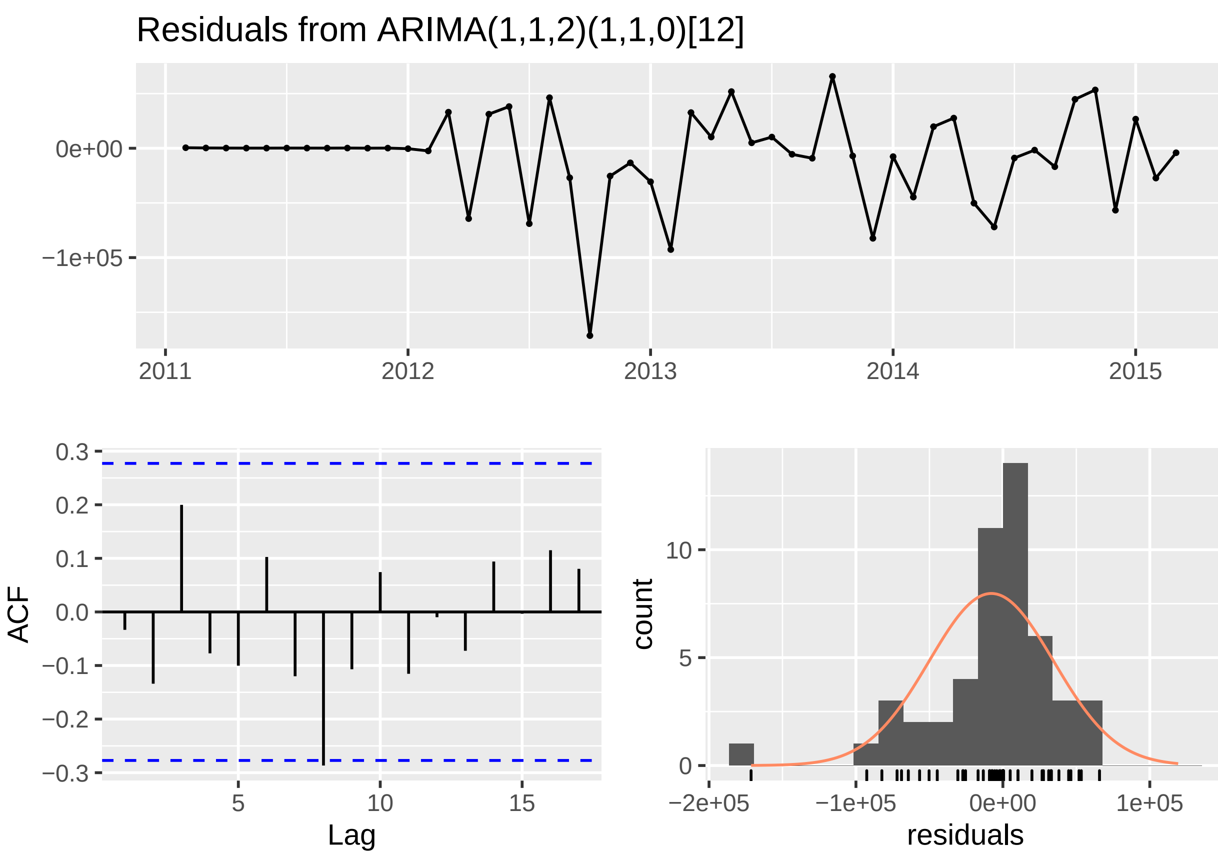
```r
pred = forecast(model, h=nValid)
accuracy(pred, test.ts)
```

```
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set  -7836.274  42681.0  27784.42 -0.6668532 2.124193 0.1997077
## Test set     109444.736 125868.1 111297.20  6.5769550 6.702690 0.7999774
##                     ACF1 Theil's U
## Training set -0.03343257        NA
## Test set      0.44480419  1.390368
```

```r
autoplot(data.ts, main="ARIMA forecast sales vs. actual sales")+
  autolayer(model$fitted, series="forecast_training")+
  autolayer(pred, series="forecast_test", PI=FALSE)
```




```
checkresiduals(pred)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(1,1,0)[12]
## Q* = 11.821, df = 6, p-value = 0.06608
##
## Model df: 4.   Total lags used: 10
```

Final Forecasting model

This is our final forecasting model that should be deployed

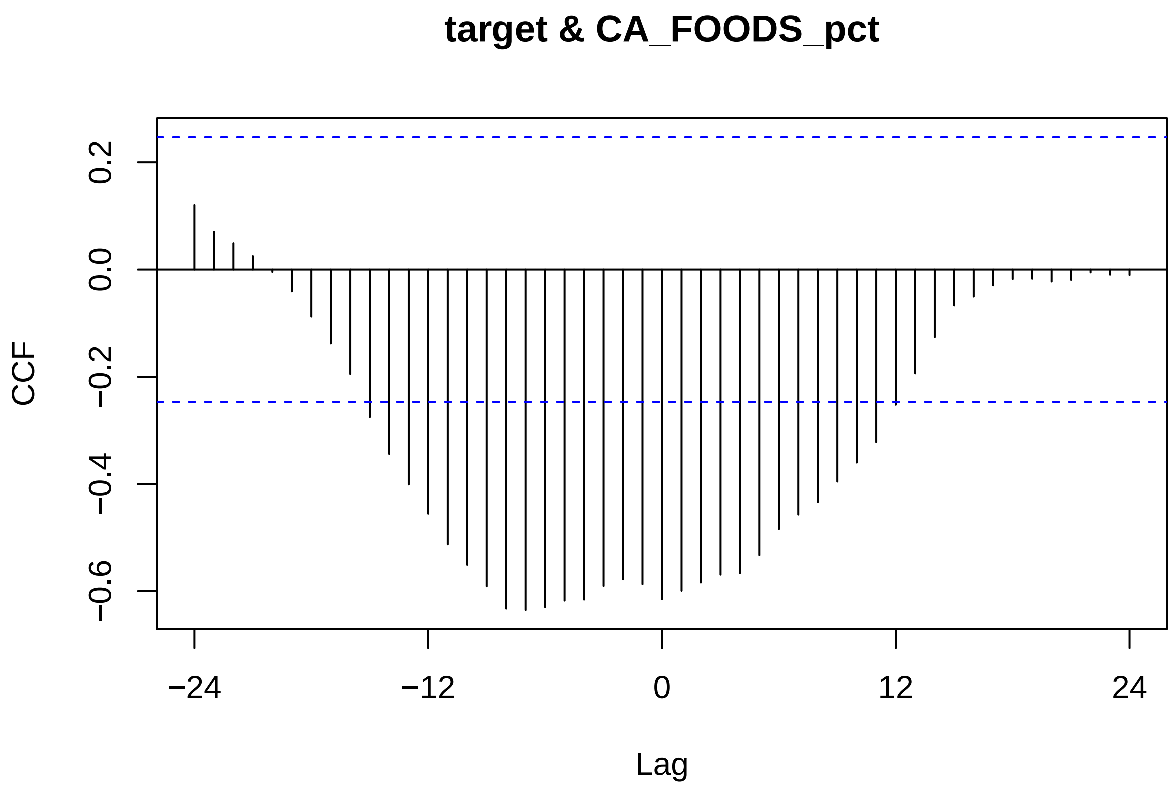
```
model = Arima(data.ts, order=c(1,1,2), seasonal=list(order=c(1,1,0), period=12))
pred = forecast(model, h=12)
pred$mean
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 2016                                 1844941 1839502 1929836 2011885
## 2017 1937087 1865640 1994464 2030718
##          Sep     Oct     Nov     Dec
## 2016 1910781 1989970 1859017 1851223
## 2017
```

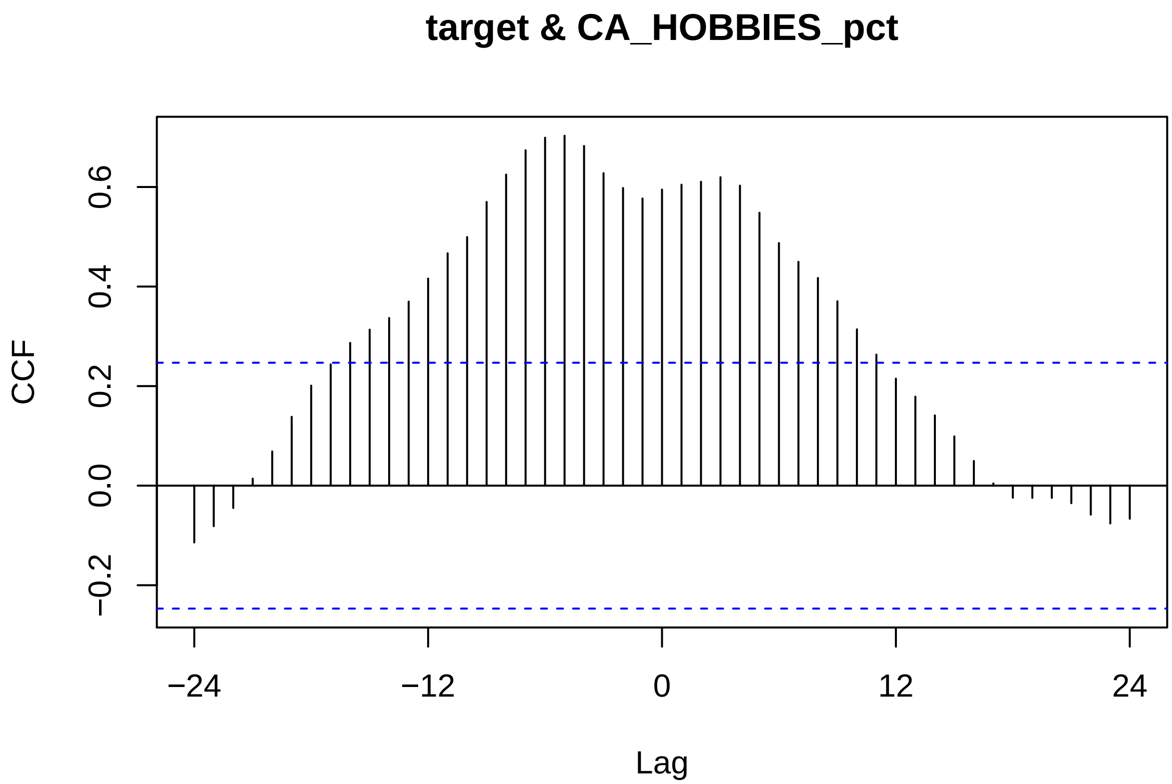
External Variables

```
target = ts(data$CA_total, start=c(2011,2), frequency=12)
trend = time(target)
```

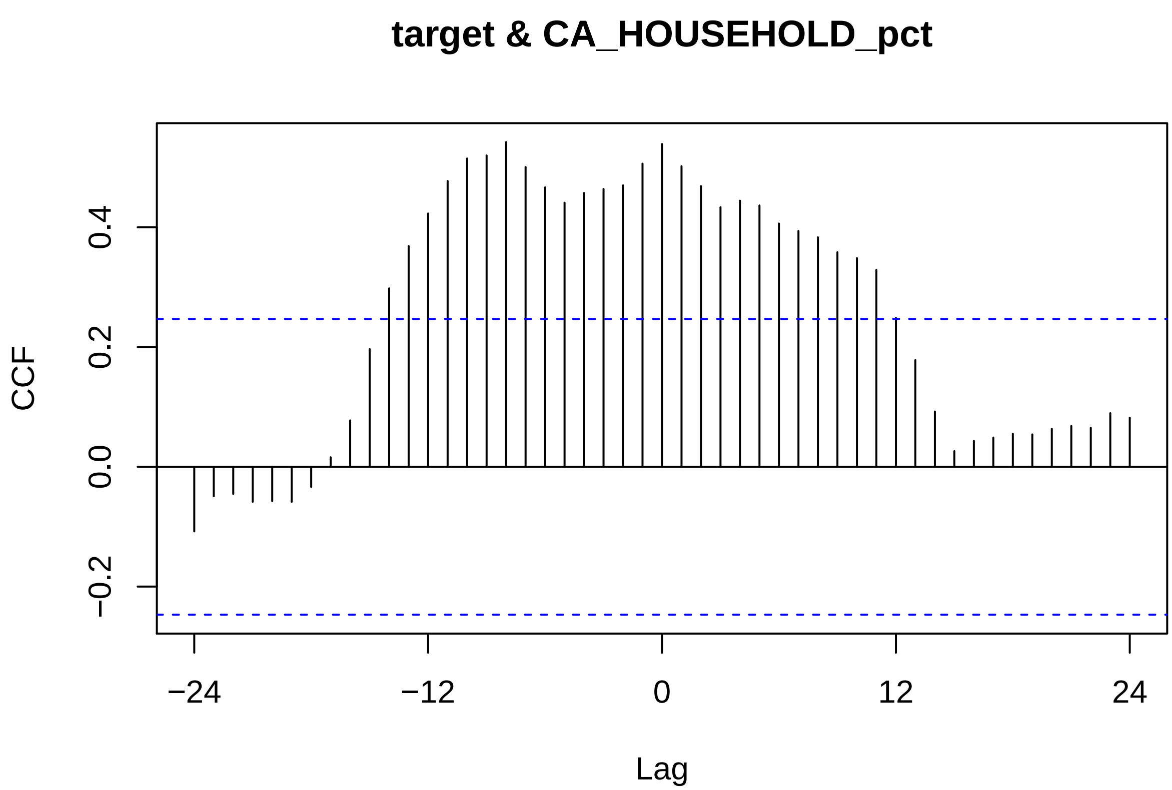
```r
CA_FOODS_pct = ts(data$CA_FOODS_pct, start=c(2011,2), frequency=12)
CA_HOBBIES_pct = ts(data$CA_HOBBIES_pct, start=c(2011,2), frequency=12)
CA_HOUSEHOLD_pct = ts(data$CA_HOUSEHOLD_pct, start=c(2011,2), frequency=12)
event = ts(data$event, start=c(2011,2), frequency=12)
event_weekend = ts(data$event_weekend, start=c(2011,2), frequency=12)
Cultural = ts(data$Cultural, start=c(2011,2), frequency=12)
National = ts(data$National, start=c(2011,2), frequency=12)
Religious = ts(data$Religious, start=c(2011,2), frequency=12)
Sporting = ts(data$Sporting, start=c(2011,2), frequency=12)
TX_total = ts(data$TX_total, start=c(2011,2), frequency=12)
WI_total = ts(data$WI_total, start=c(2011,2), frequency=12)
unemployment = ts(data$unemployment_rate, start=c(2011,2), frequency=12)
income = ts(data$personal_income, start=c(2011,2), frequency=12)
gdp = ts(data$real_gdp, start=c(2011,2), frequency=12)

Ccf(target, CA_FOODS_pct) # lag7
```



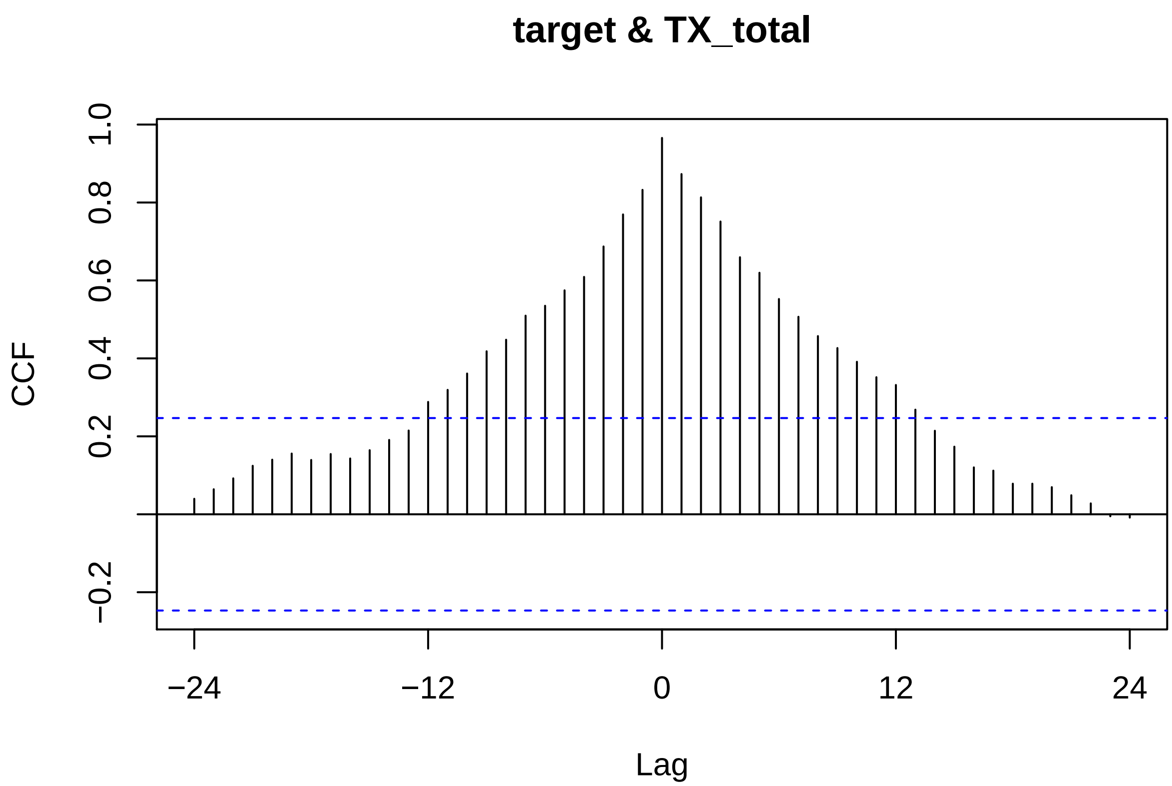

```r
Ccf(target, CA_HOBBIES_pct) # lag5
```



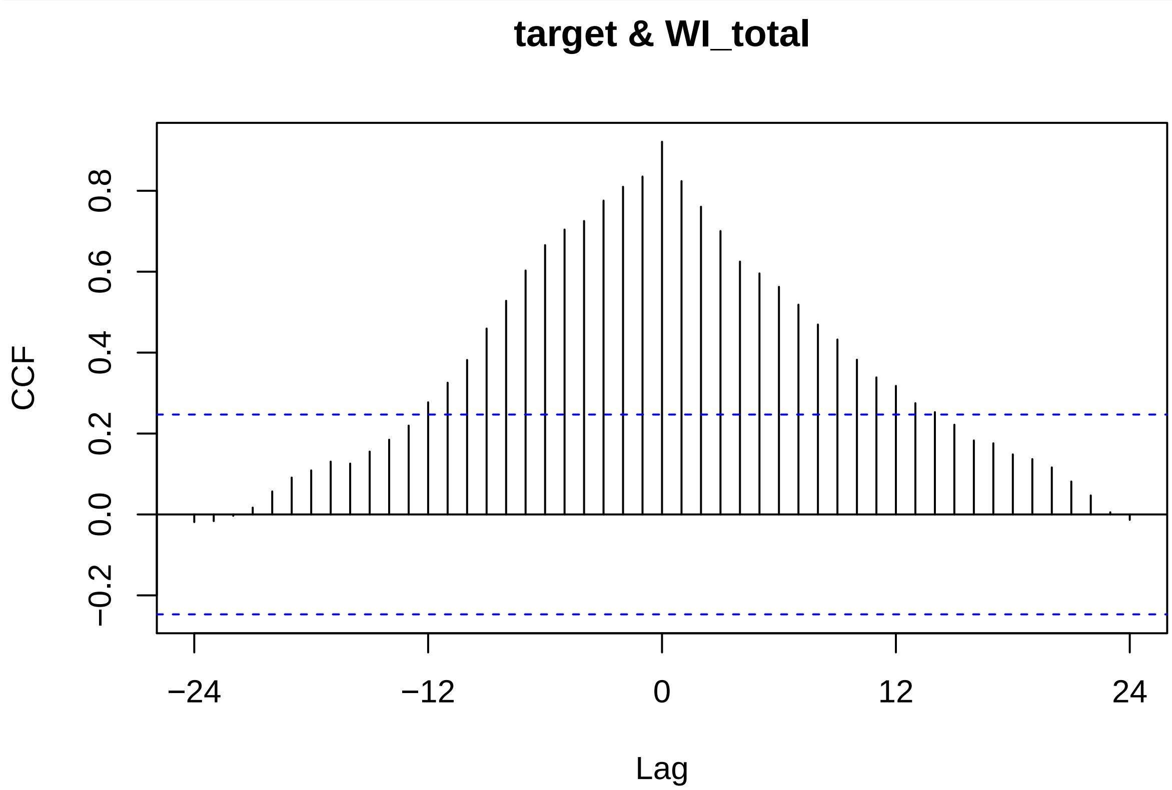

```r
Ccf(target, CA_HOUSEHOLD_pct) # lag8
```



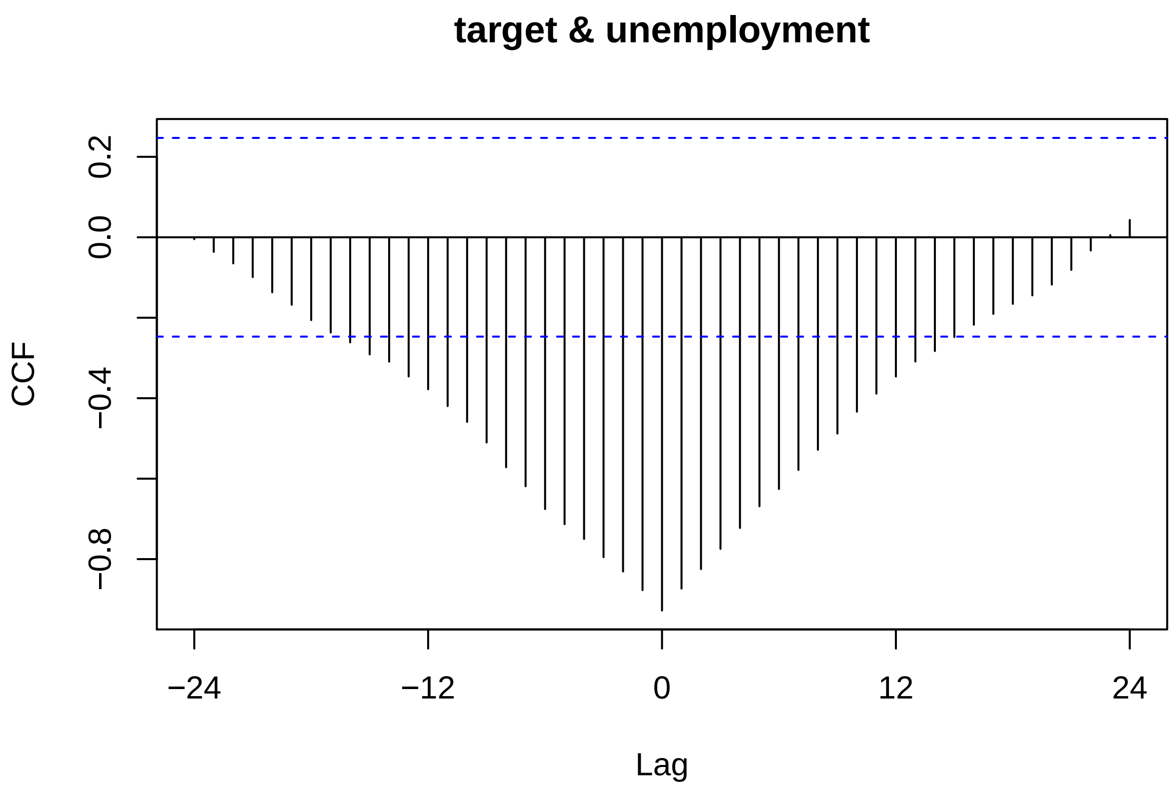

```r
Ccf(target, TX_total) # lag1
```



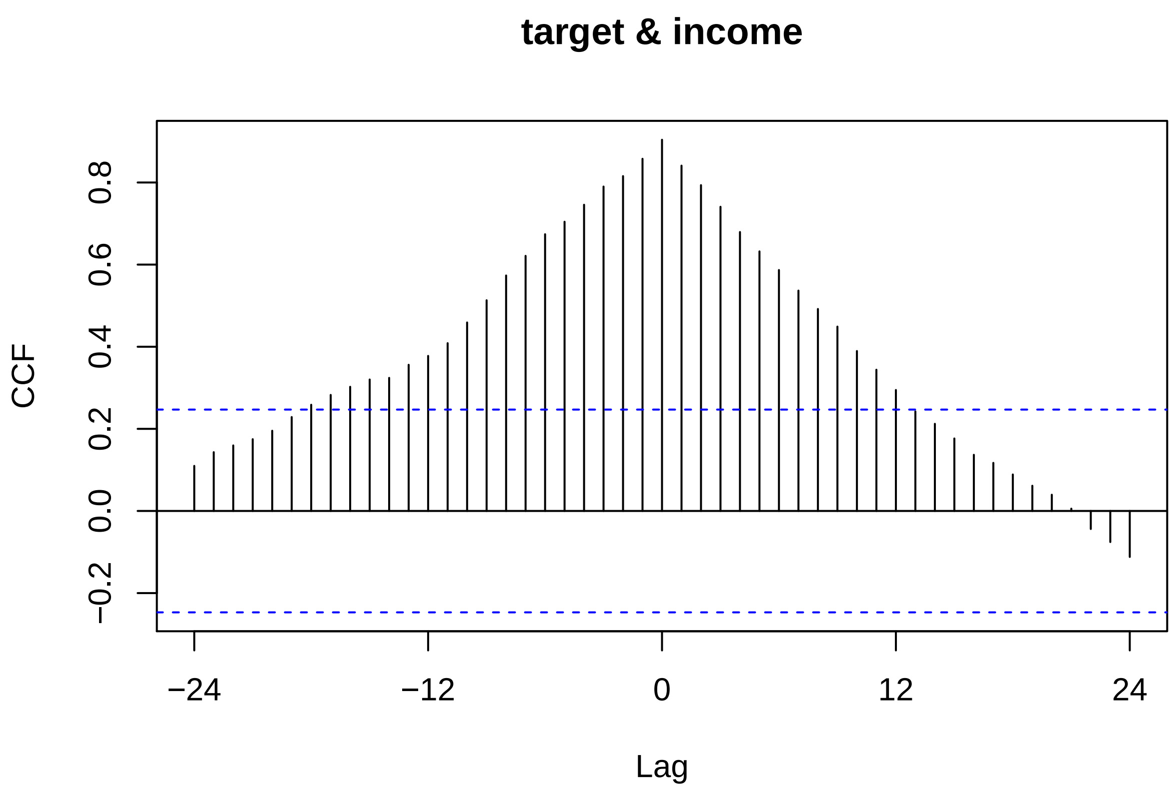

```r
Ccf(target, WI_total) # lag1
```



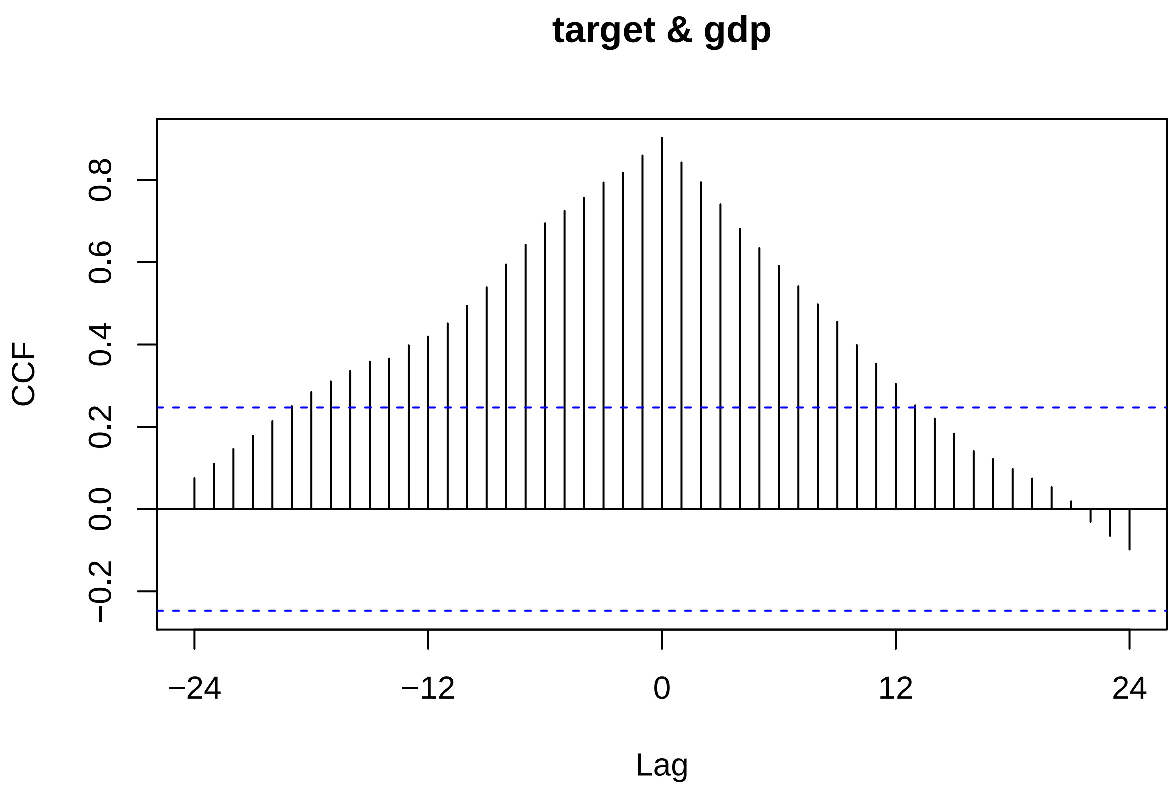

```r
Ccf(target, unemployment) # lag1
```




```r
Ccf(target, income) # lag1
```




```r
Ccf(target, gdp) # lag1
```




```
newdata = ts.intersect(target, trend,
                       food=lag(CA_FOODS_pct,-7), hobbies=lag(CA_HOBBIES_pct,-5), household=lag(CA_HOUSE
                       event, event_weekend, event_culture=Cultural, event_religion=Religious, event_spo
                       WI=lag(WI_total,-1),TX=lag(TX_total,-1),
                       unemployment=lag(unemployment,-1), gdp=lag(gdp,-1), income=lag(income,-1))

model = lm(target ~ trend + food + hobbies + household +
               event + event_weekend + event_culture + event_religion + event_sport +
               WI+TX +
               unemployment + gdp + income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ trend + food + hobbies + household + event +
##     event_weekend + event_culture + event_religion + event_sport +
##     WI + TX + unemployment + gdp + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -131002  -27746    7419   35241  101207
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.131e+09  4.634e+08  -2.442 0.019144 *
## trend           5.630e+05  2.301e+05   2.447 0.018897 *
## food            6.352e+04  8.123e+05   0.078 0.938062
## hobbies         6.931e+05  1.261e+06   0.550 0.585542
```

```
## household      -4.498e+05  1.025e+06  -0.439 0.663033
## event          -5.587e+04  1.456e+04  -3.838 0.000432 ***
## event_weekend  -8.919e+03  1.322e+04  -0.674 0.503893
## event_culture   5.228e+04  1.812e+04   2.885 0.006275 **
## event_religion  3.899e+04  1.814e+04   2.149 0.037703 *
## event_sport     2.639e+04  1.900e+04   1.389 0.172566
## WI             -5.246e-01  2.583e-01  -2.031 0.048912 *
## TX              5.774e-01  2.211e-01   2.612 0.012619 *
## unemployment    2.216e+05  1.221e+05   1.815 0.077025 .
## gdp            -1.336e+00  7.292e-01  -1.832 0.074378 .
## income          2.350e-02  4.620e-01   0.051 0.959687
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61310 on 40 degrees of freedom
## Multiple R-squared:   0.92,  Adjusted R-squared:  0.892
## F-statistic: 32.85 on 14 and 40 DF,  p-value: < 2.2e-16
```

```r
model = lm(target ~ food + hobbies + household +
               event + event_weekend + event_culture + event_religion + event_sport +
               WI+TX +
               unemployment + gdp + income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ food + hobbies + household + event + event_weekend +
##     event_culture + event_religion + event_sport + WI + TX +
##     unemployment + gdp + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -134665  -35975    4993   44793  101766
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.498e+06  1.651e+06   1.513 0.137933
## food           -4.837e+05  8.270e+05  -0.585 0.561857
## hobbies         1.591e+06  1.277e+06   1.246 0.219876
## household       3.912e+05  1.022e+06   0.383 0.703887
## event          -5.628e+04  1.542e+04  -3.651 0.000733 ***
## event_weekend  -1.060e+04  1.399e+04  -0.758 0.452783
## event_culture   5.327e+04  1.919e+04   2.776 0.008246 **
## event_religion  4.373e+04  1.910e+04   2.289 0.027288 *
## event_sport     2.729e+04  2.012e+04   1.356 0.182410
## WI             -1.194e-01  2.099e-01  -0.569 0.572666
## TX              6.357e-01  2.328e-01   2.731 0.009260 **
## unemployment   -6.610e+04  3.483e+04  -1.898 0.064763 .
## gdp            -8.607e-01  7.444e-01  -1.156 0.254233
## income          4.936e-01  4.450e-01   1.109 0.273837
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64930 on 41 degrees of freedom
```

```
## Multiple R-squared:  0.908,  Adjusted R-squared:  0.8788
## F-statistic: 31.13 on 13 and 41 DF,  p-value: < 2.2e-16
```

```r
model = lm(target ~ food + hobbies + household +
               event + event_weekend + event_culture + event_religion + event_sport +
               WI+TX +
               unemployment + income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ food + hobbies + household + event + event_weekend +
##     event_culture + event_religion + event_sport + WI + TX +
##     unemployment + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -132602  -37690    9482   45833  109773
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.045e+06  1.075e+06   0.972 0.336635
## food           -1.832e+05  7.883e+05  -0.232 0.817389
## hobbies         1.533e+06  1.281e+06   1.196 0.238353
## household       2.935e+05  1.023e+06   0.287 0.775511
## event          -5.681e+04  1.547e+04  -3.672 0.000674 ***
## event_weekend  -1.075e+04  1.404e+04  -0.766 0.448012
## event_culture   5.295e+04  1.926e+04   2.749 0.008779 **
## event_religion  4.285e+04  1.916e+04   2.236 0.030725 *
## event_sport     2.840e+04  2.018e+04   1.408 0.166562
## WI             -7.757e-02  2.076e-01  -0.374 0.710507
## TX              5.734e-01  2.273e-01   2.522 0.015536 *
## unemployment   -4.378e+04  2.910e+04  -1.504 0.140019
## income          8.766e-02  2.746e-01   0.319 0.751143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65190 on 42 degrees of freedom
## Multiple R-squared:  0.905,  Adjusted R-squared:  0.8778
## F-statistic: 33.34 on 12 and 42 DF,  p-value: < 2.2e-16
```

```r
model = lm(target ~ food + hobbies + household +
               event + event_weekend + event_culture + event_religion + event_sport +
               WI+TX +
               income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ food + hobbies + household + event + event_weekend +
##     event_culture + event_religion + event_sport + WI + TX +
##     income, data = newdata)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -137865  -43980    1557   40426  111800
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.423e+04  8.002e+05  -0.068 0.946282
## food           -4.085e+05  7.852e+05  -0.520 0.605510
## hobbies         1.906e+06  1.275e+06   1.494 0.142396
## household       5.570e+05  1.022e+06   0.545 0.588657
## event          -5.991e+04  1.556e+04  -3.852 0.000385 ***
## event_weekend  -1.290e+04  1.417e+04  -0.911 0.367625
## event_culture   5.561e+04  1.946e+04   2.858 0.006547 **
## event_religion  4.503e+04  1.939e+04   2.323 0.024994 *
## event_sport     3.115e+04  2.039e+04   1.528 0.133831
## WI              1.026e-01  1.720e-01   0.596 0.554093
## TX              5.316e-01  2.289e-01   2.322 0.025015 *
## income          4.059e-01  1.776e-01   2.285 0.027322 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66140 on 43 degrees of freedom
## Multiple R-squared:  0.8999, Adjusted R-squared:  0.8743
## F-statistic: 35.13 on 11 and 43 DF,  p-value: < 2.2e-16
```

```r
model = lm(target ~ hobbies + household +
               event + event_weekend + event_culture + event_religion + event_sport +
               TX +
               income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ hobbies + household + event + event_weekend +
##     event_culture + event_religion + event_sport + TX + income,
##     data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -141851  -44440    3666   41720  118251
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.090e+05  1.934e+05  -2.632 0.011586 *
## hobbies         2.235e+06  1.028e+06   2.175 0.034897 *
## household       8.532e+05  7.580e+05   1.126 0.266312
## event          -5.938e+04  1.529e+04  -3.885 0.000333 ***
## event_weekend  -1.578e+04  1.346e+04  -1.172 0.247169
## event_culture   5.568e+04  1.912e+04   2.912 0.005577 **
## event_religion  4.836e+04  1.856e+04   2.605 0.012401 *
## event_sport     3.123e+04  2.006e+04   1.557 0.126570
## TX              5.764e-01  1.954e-01   2.949 0.005037 **
## income          4.729e-01  1.488e-01   3.178 0.002679 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65100 on 45 degrees of freedom
## Multiple R-squared:  0.8985, Adjusted R-squared:  0.8782
## F-statistic: 44.25 on 9 and 45 DF,  p-value: < 2.2e-16
```

```
model = lm(target ~ hobbies +
               event + event_weekend + event_culture + event_religion + event_sport +
               TX +
               income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ hobbies + event + event_weekend + event_culture +
##     event_religion + event_sport + TX + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144945  -43597    4086   42493  113527
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.551e+05  1.371e+05  -2.589 0.012841 *
## hobbies         2.825e+06  8.868e+05   3.185 0.002596 **
## event          -5.849e+04  1.531e+04  -3.821 0.000398 ***
## event_weekend  -1.475e+04  1.347e+04  -1.095 0.279252
## event_culture   5.714e+04  1.914e+04   2.986 0.004520 **
## event_religion  4.896e+04  1.861e+04   2.631 0.011544 *
## event_sport     3.104e+04  2.012e+04   1.543 0.129762
## TX              5.371e-01  1.928e-01   2.785 0.007744 **
## income          5.074e-01  1.460e-01   3.475 0.001126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65290 on 46 degrees of freedom
## Multiple R-squared:  0.8956, Adjusted R-squared:  0.8775
## F-statistic: 49.34 on 8 and 46 DF,  p-value: < 2.2e-16
```

```
model = lm(target ~ hobbies +
               event + event_culture + event_religion + event_sport +
               TX +
               income,
             data=newdata)
summary(model)
```

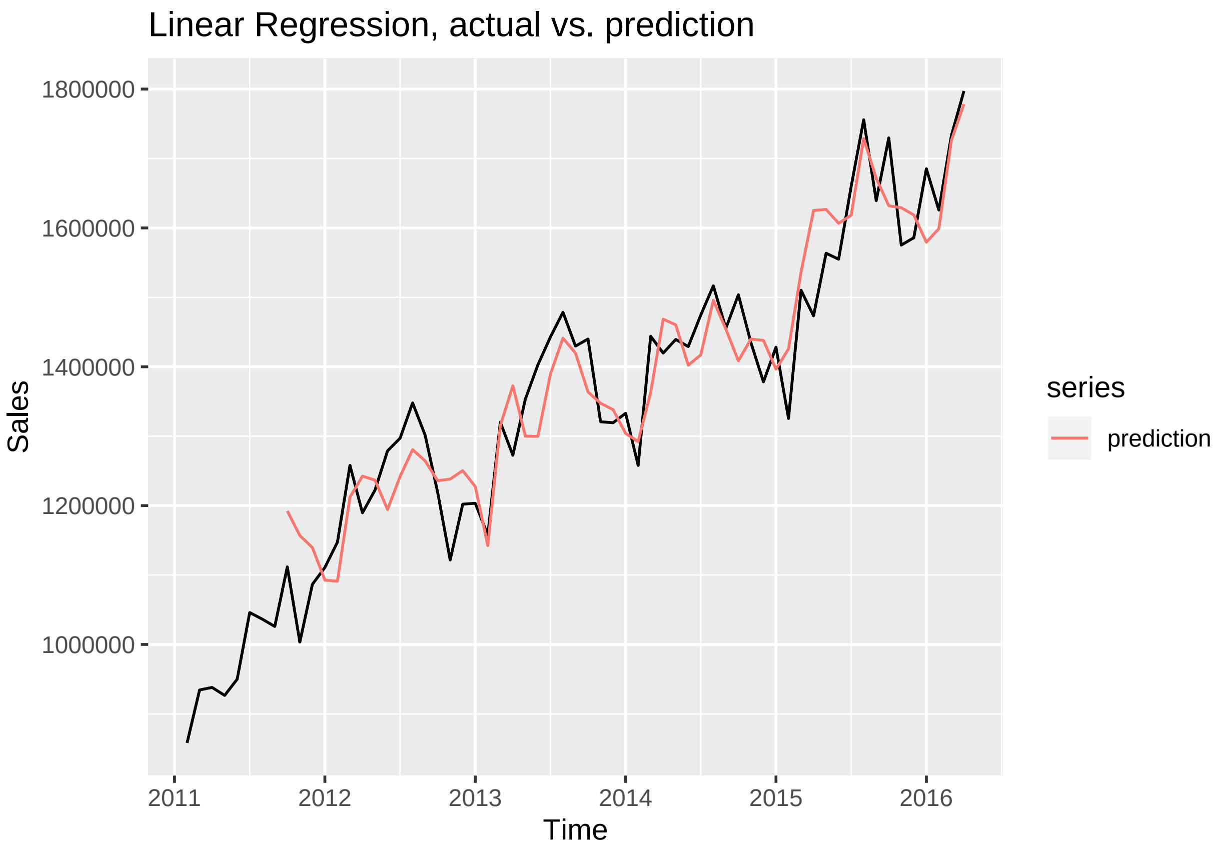
```
##
## Call:
## lm(formula = target ~ hobbies + event + event_culture + event_religion +
##     event_sport + TX + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -145719  -45032    5992   42638  118418
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.615e+05  1.373e+05  -2.633 0.011429 *
## hobbies         2.909e+06  8.853e+05   3.286 0.001924 **
## event          -5.722e+04  1.530e+04  -3.741 0.000498 ***
## event_culture   4.780e+04  1.717e+04   2.784 0.007703 **
## event_religion  4.220e+04  1.759e+04   2.399 0.020466 *
## event_sport     2.602e+04  1.963e+04   1.325 0.191499
## TX              5.401e-01  1.932e-01   2.795 0.007489 **
## income          5.028e-01  1.463e-01   3.437 0.001239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65430 on 47 degrees of freedom
## Multiple R-squared:  0.8929, Adjusted R-squared:  0.8769
## F-statistic: 55.98 on 7 and 47 DF,  p-value: < 2.2e-16
```

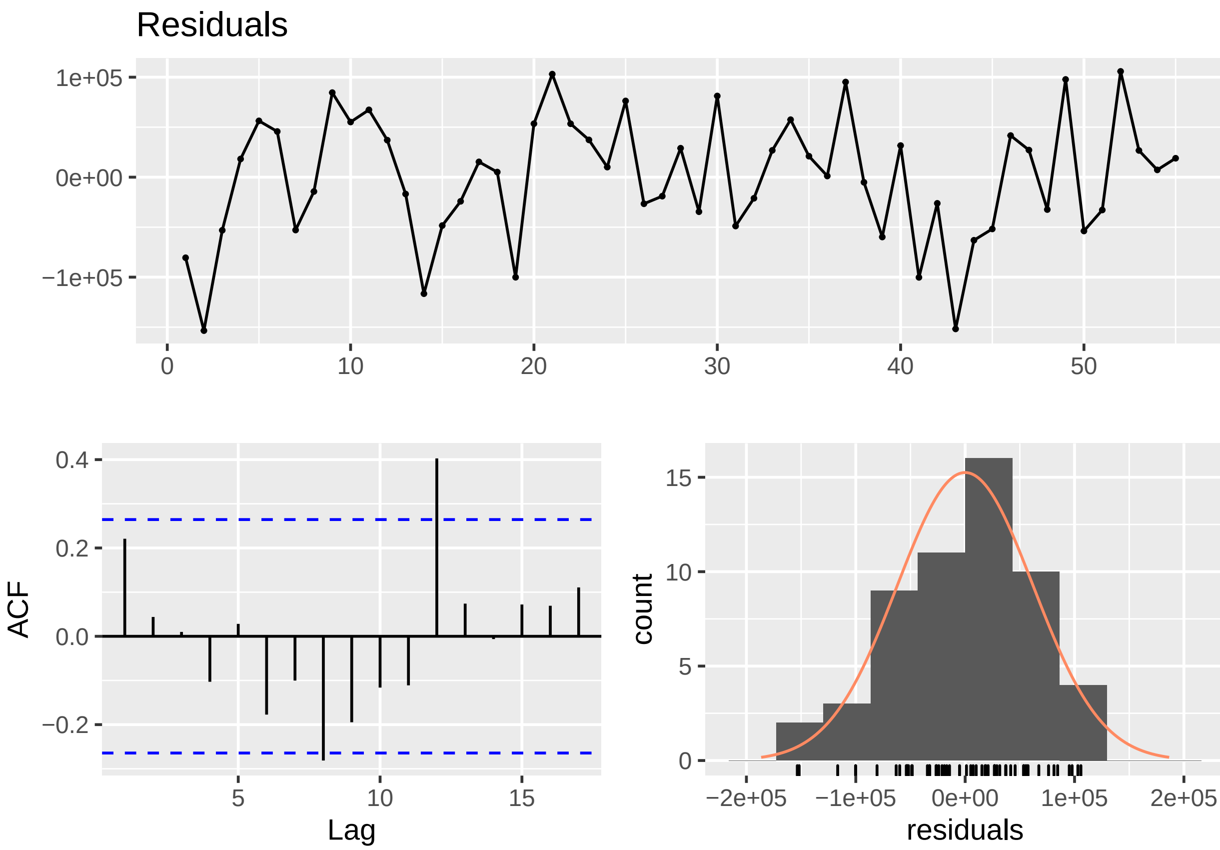
```r
model = lm(target ~ hobbies +
               event + event_culture + event_religion +
               TX +
               income,
             data=newdata)
summary(model)
```

```
##
## Call:
## lm(formula = target ~ hobbies + event + event_culture + event_religion +
##     TX + income, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -153459  -41457    7297   43664  105826
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.492e+05  1.381e+05  -2.529 0.014781 *
## hobbies         2.930e+06  8.921e+05   3.285 0.001912 **
## event          -4.521e+04  1.242e+04  -3.640 0.000665 ***
## event_culture   4.244e+04  1.681e+04   2.524 0.014966 *
## event_religion  3.358e+04  1.647e+04   2.038 0.047033 *
## TX              6.085e-01  1.877e-01   3.243 0.002156 **
## income          4.579e-01  1.434e-01   3.193 0.002486 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65940 on 48 degrees of freedom
## Multiple R-squared:  0.8889, Adjusted R-squared:  0.875
## F-statistic:    64 on 6 and 48 DF,  p-value: < 2.2e-16
```

```r
target_fitted = ts(model$fitted.values, start=c(2011,2+8), frequency=12)
autoplot(target, ylab="Sales", main="Linear Regression, actual vs. prediction")+
  autolayer(target_fitted, series="prediction")
```




```
checkresiduals(model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 13.237, df = 10, p-value = 0.2107
```