Project: Forecast of Grocery Stores Sales in the U.S. Using Exponential Smoothing and ARIMA Models

7-6-2018

Author:
Simon Sudora

Contents

Introduction

The goal of the project is to create and compare performance of exponential smoothing model and ARIMA model in particular case of forecasting monthly grocery stores sales in the U.S.

Data comes from the portal https://www.census.gov. On this portal are published open data sets of various economical measurements of the U.S.

Data set of monthly grocery stores [1] starts in the January 1992 and ends in the December 2017. This time series has 312 data points. Each data point represents monthly summed sales in all grocery stores in U.S.

Analysis, modelling and forecasting of time series will be used software R with help of packages: “forecast”, ”zoo”, “astsa”.

Time Series Analysis

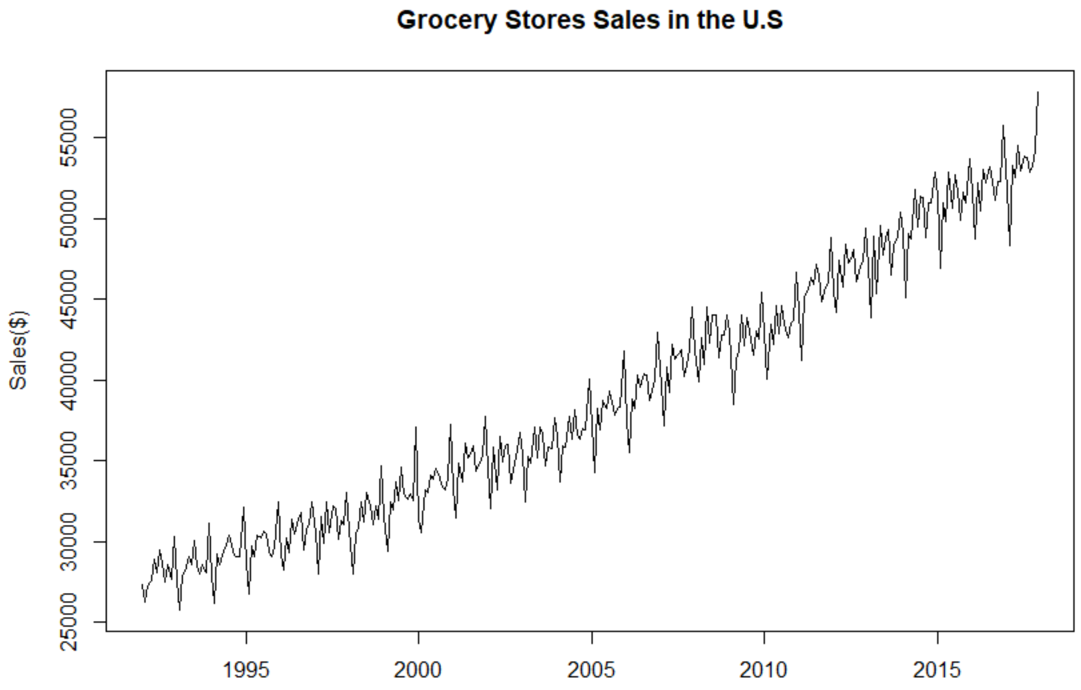

Fig. 1

By visually exploring the plotted time series in the Fig .1 we can conclude:

- It has positive linear trend. Time series had its minimum in the February 1993. The highest sales were achieved in December 2017
- It has 12 months (yearly) seasonality. This seasonality can be spotted mainly because of the pattern on the end and beginnings of the years. There are significant peaks in December and following big decreases in the first 2 months of the following year. This has very natural origin in consumer behavior. December is a period of the Christmas holidays when people eat lot of food. However, in consequent months they tend to not spend that much and rest a little.

- We can identify also another period of the year when the sales are higher. During the summer consumers tends to spend more because of the vacations. However, this pattern is not that clear and there are years when it slightly differs.
- We can also see anomalies of lower sales in the years 2009, 2010, 2011 caused by financial crisis.

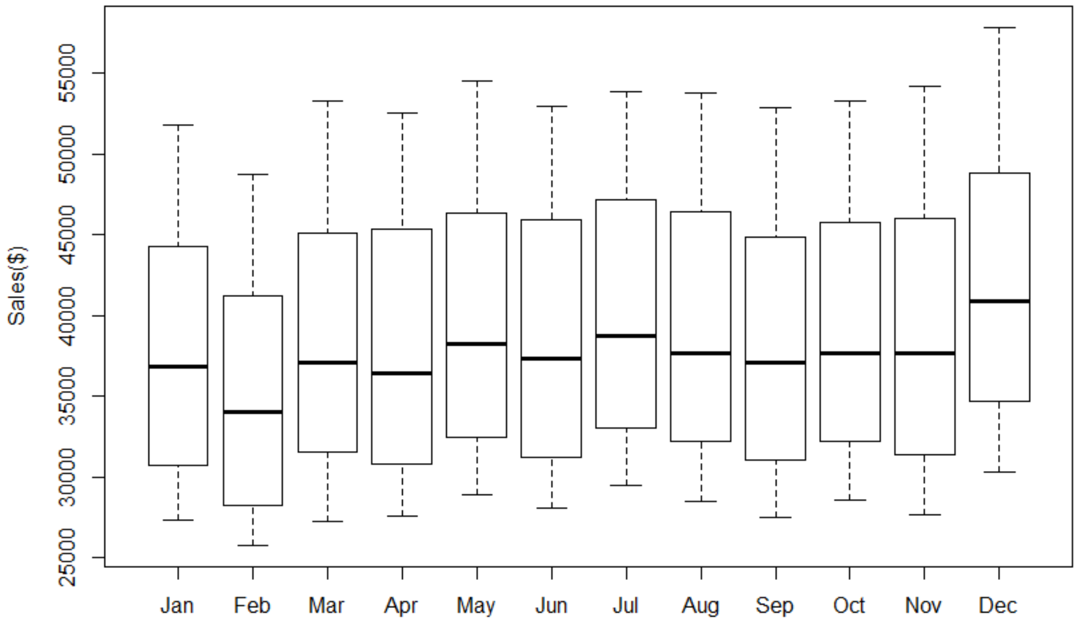

Fig. 2

From the boxplot in the Fig. 2 we can confirm what we explored earlier:

- Time series has seasonal pattern and sales in the months follow the normal distribution without any outliers.
- Highest sales are in December and the lowest in February
- Differences among the other months are not that big.
- In general, slightly higher sales are also in July

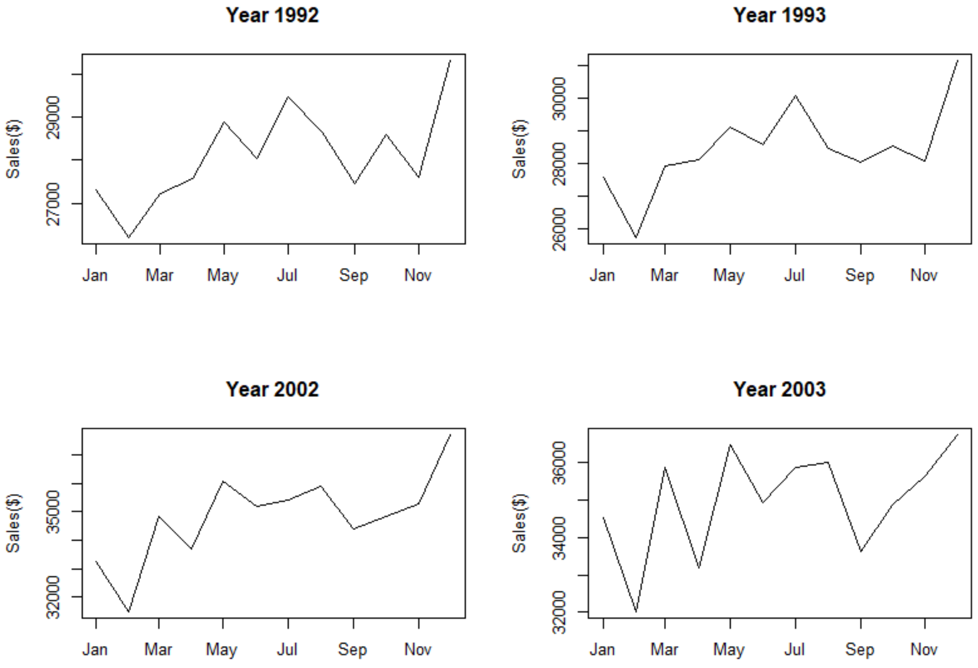

Fig. 3

By zooming on some of the years of the time series we can say:

- Seasonal pattern slightly changed over time. There is apparent peak in July in 1992 and 1993. However, in 2002 and 2003 summer peak is shifted to August and sales in July and August are more balanced.

Exponential Smoothing

Decomposition

From the Fig. 1 we can assume that time series has additive seasonality, since amplitude of seasonal changes is constant over time. Seasonal changes are not related to trend.

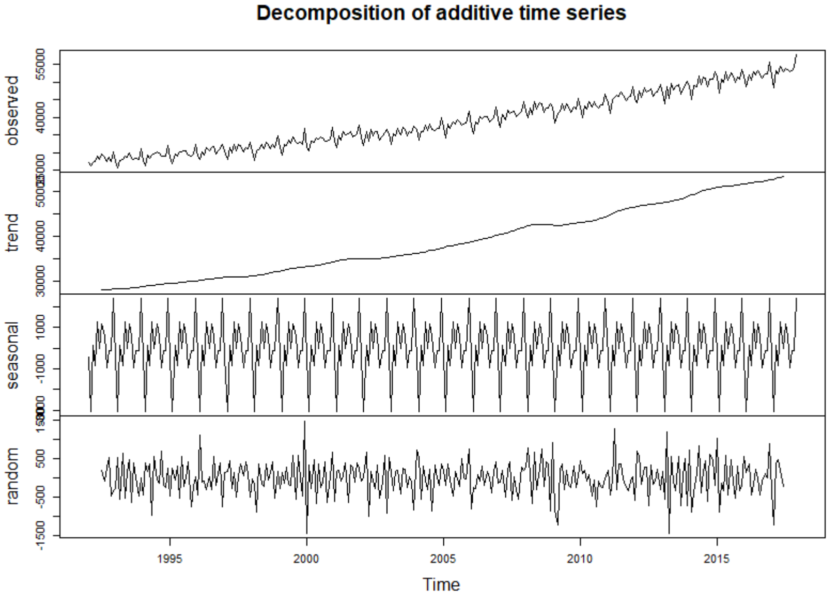

Fig. 4

Plot of additive decomposition of time series in the Fig. 4 proved that time series has all 3 components (trend, seasonality, and random noise).

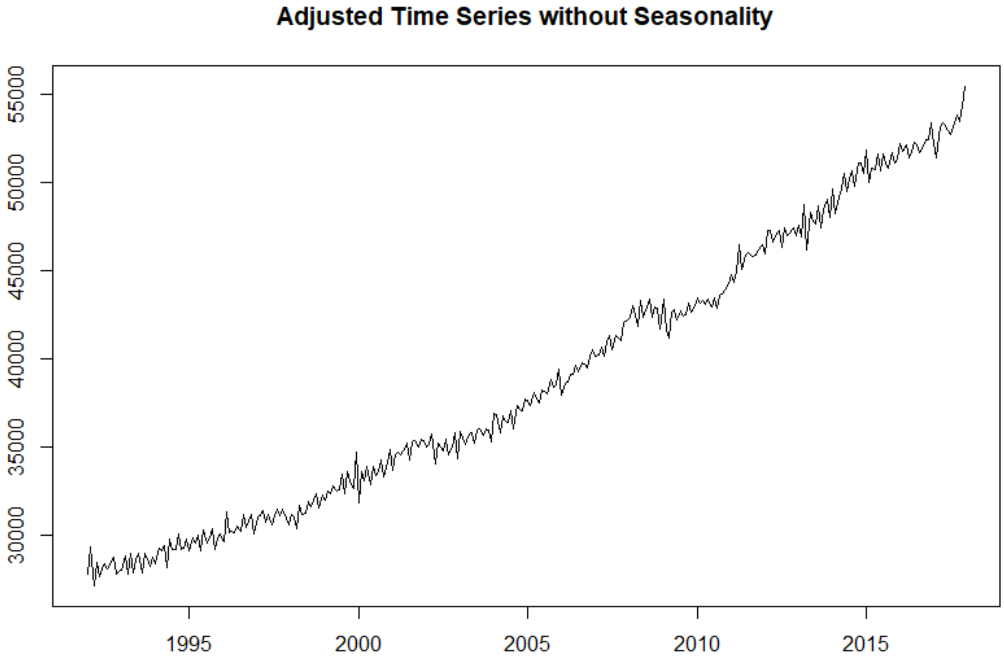

Fig. 5 - Adjusted time series was obtained by subtracting the seasonal component from the time series

ES Model

We identified that our time series has trend and additive seasonality, therefore the suitable exponential smoothing method for this case is additive Holt-Winters. This exponential smoothing method can deal with trend as well as with seasonality.

To create additive Holt-Winters I used function “ets” from the library “forecast” with following parameters ets(ts.train, model = "AAA"). Abbreviation “AAA” stands for additive error, additive trend and additive seasonality.

Model was trained on time series between years 1992 and 2016 and then the validation forecast was made on the 2017.

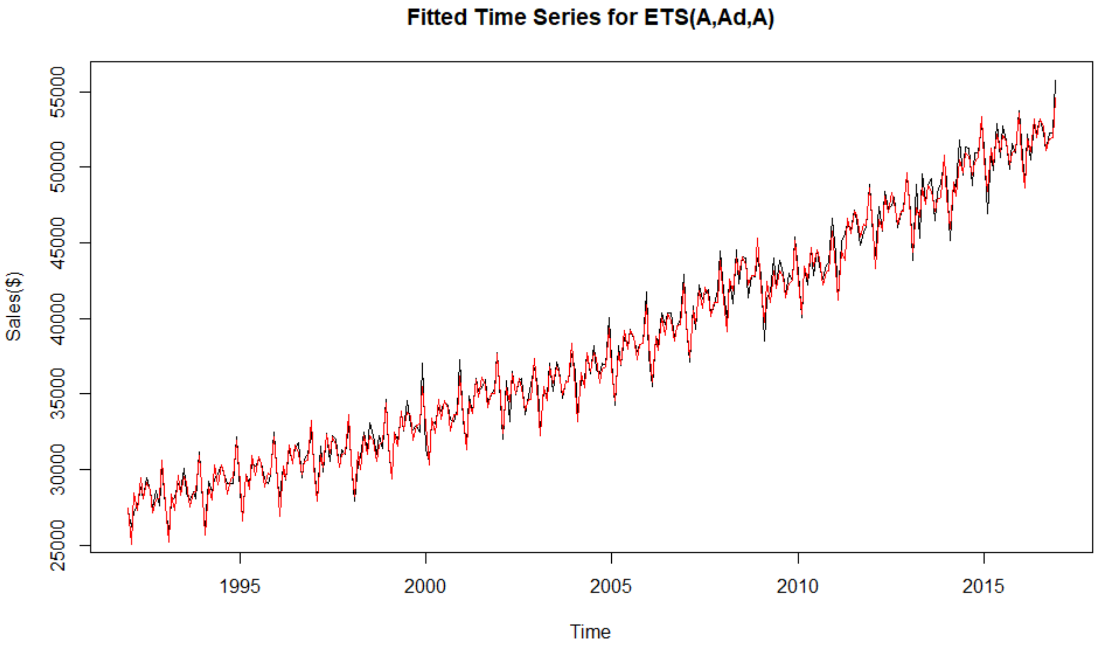

Fig. 6

On Fig. 6 we can see that the model (red line) nicely fitted original time series (black line). It has MAPE = 1.16% on the training set.

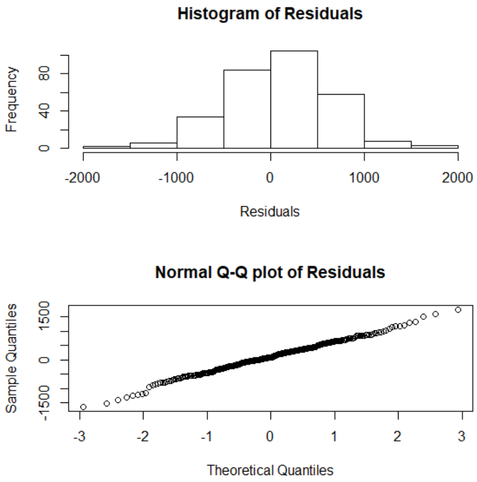

Fig. 7

Plots in the Fig. 7 proves that residuals of the fitted model follow normal distribution and model can be used for forecasting.

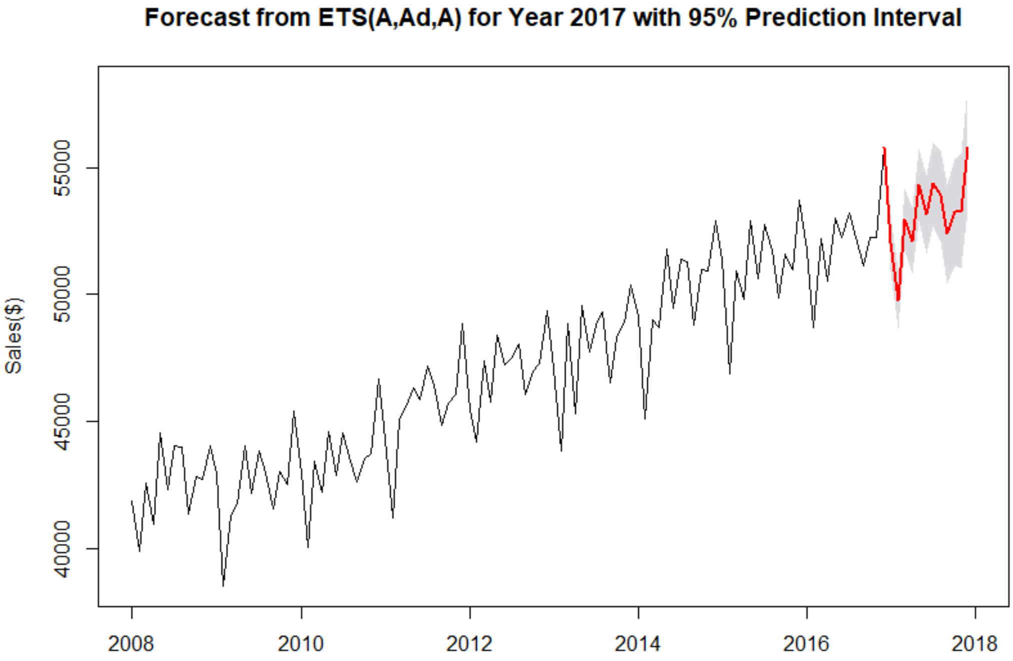

Fig. 8

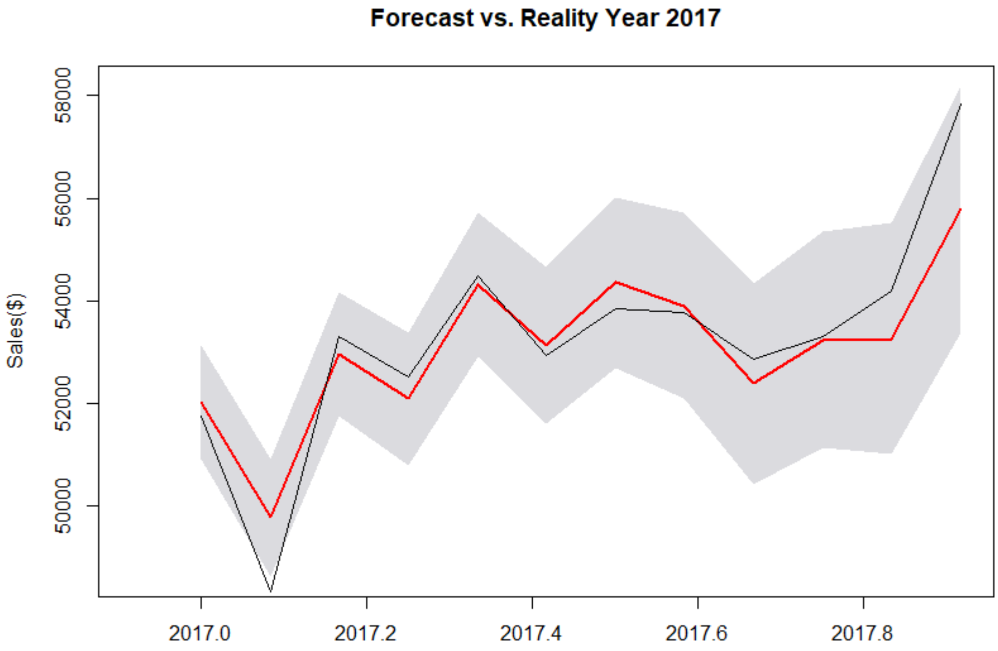

Fig. 9

Fig. 9 shows that ES(additive Holt-Winters) performs also good on the validation set with MAPE = 1.08% which is even better than the accuracy on the training set. Only prediction in February made problems to model. Sale in February 2017 were unexpectedly too low.

ARIMA

For creating and testing Arima model identical training and validation sets were used as for exponential smoothing model.

Stationarity

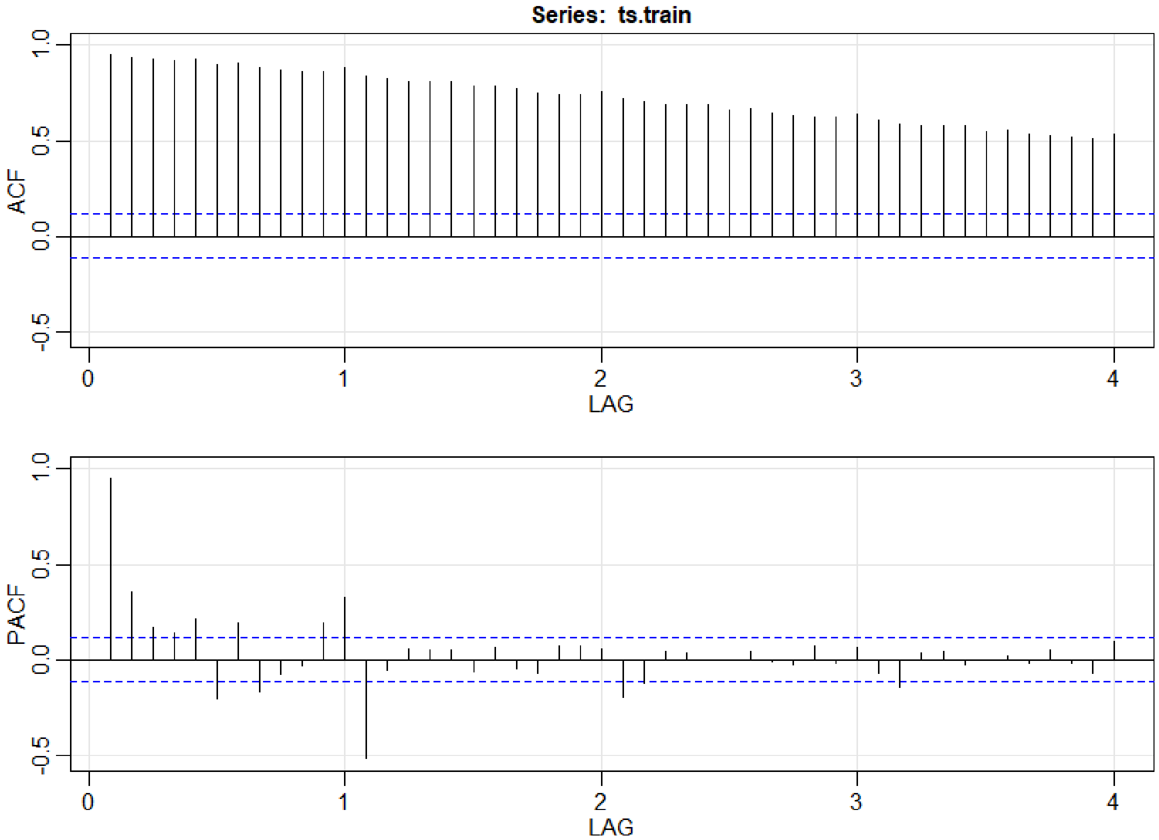

Fig. 10

From the earlier plots (Fig. 1, Fig. 4) we have already known that time series has trend and seasonality which implies non-constant mean and non-constant variance. ACF plot only confirmed this. The ACF decays very slowly, this means that time series has positive trend. We can also see peaks at 1,2,3 value of lags. This indicates seasonality.

To estimate number of difference needed I used function which "nsdiffs" from the library "forecast". This function uses Wang, Smith & Hyndman unit root test which can also deal with seasonal time series. Dickey-Fuller test can't be used because i tis not suitable for seasonal time series. The outcome of the estimation was that only 1 differencing is needed to make our time series stationary.

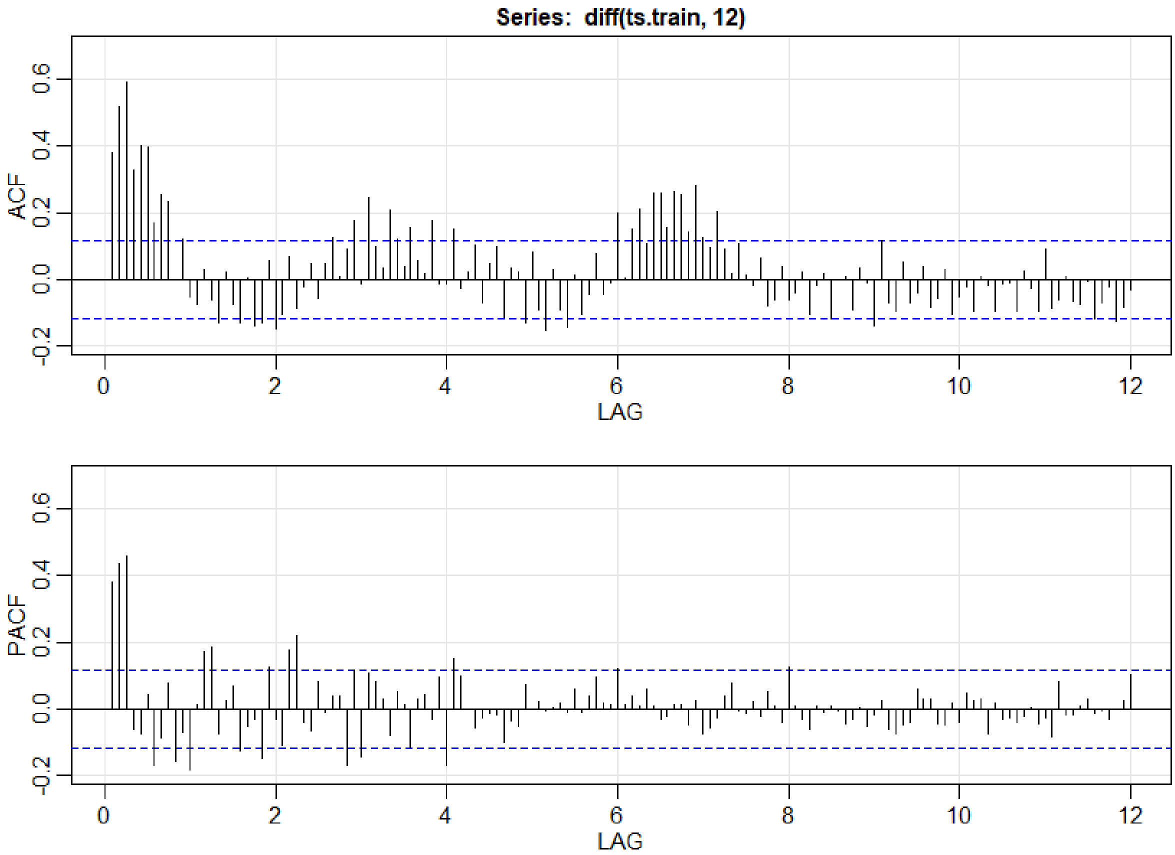

Fig. 11

From Fig. 11 we can see that one differencing helped and both ACF and PACF slowly decays to zero. For this type of time series ARMA model is suitable.

In this case, it is difficult to determine p and q parameters and it's suitable to use estimation method. I used function "auto.arima" from the library "forecast" which uses AIC, AICc and BIC value to find the best parameters for ARIMA. Function estimated following values for ARIMA(2,1,0)(2,1,2)[12]

To create ARIMA model I used function "arima" from the native library "stats" with aforementioned parameters.

ARIMA model

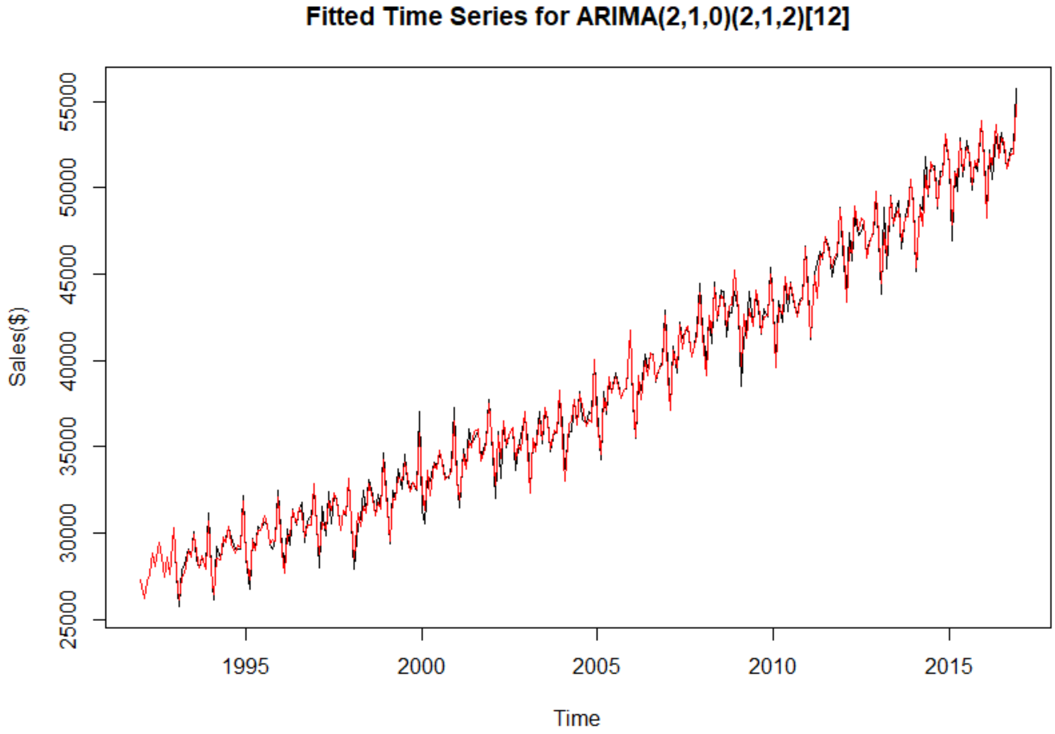

Fig. 12

Plot on Fig. 12 represent fitted time series and again as in the case of exponential smoothing, trained ARIMA model fits original time series very nicely. It has MAPE = 0.85% on the training set. This is even less than MAPE of ES model.

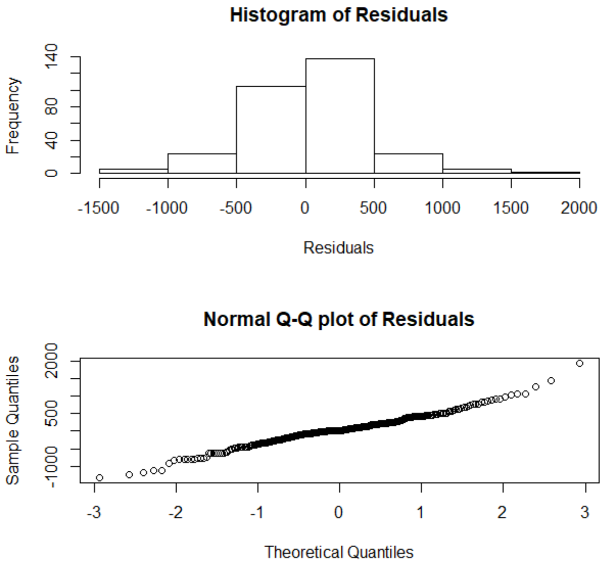

Fig. 13

Histogram and Q-Q plot in the Fig. 13 that residuals of the fitted Arima model have normal distribution which means good fit of ARIMA model.

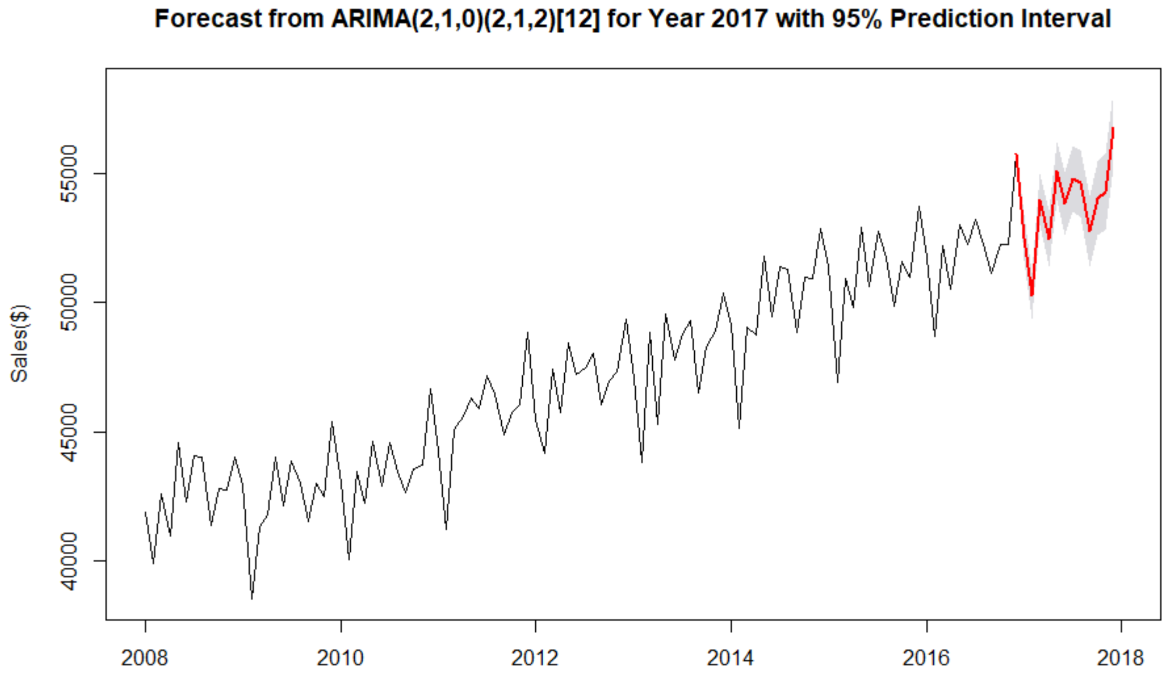

Fig. 14

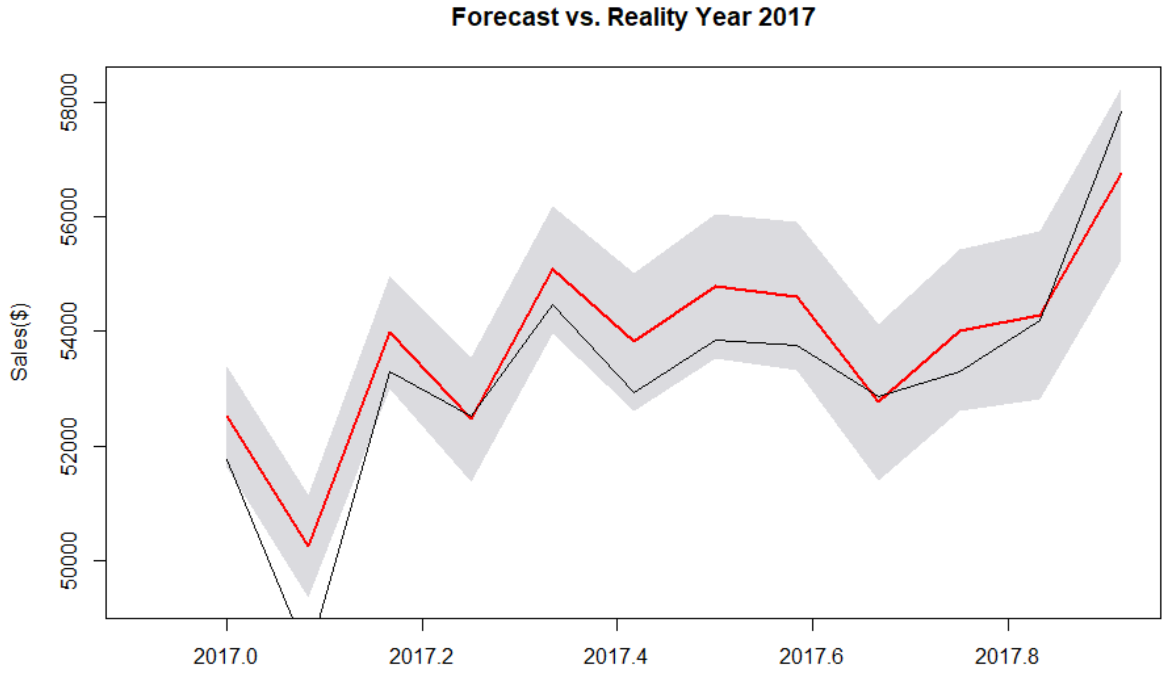

Fig. 15

In the Fig. 15 we can see that ARIMA model was not that successful on the validation set. MAPE was 1.37%. It performs slightly worse than ES. Again, forecast for February sales is the biggest problem.

Comparison and Conclusion

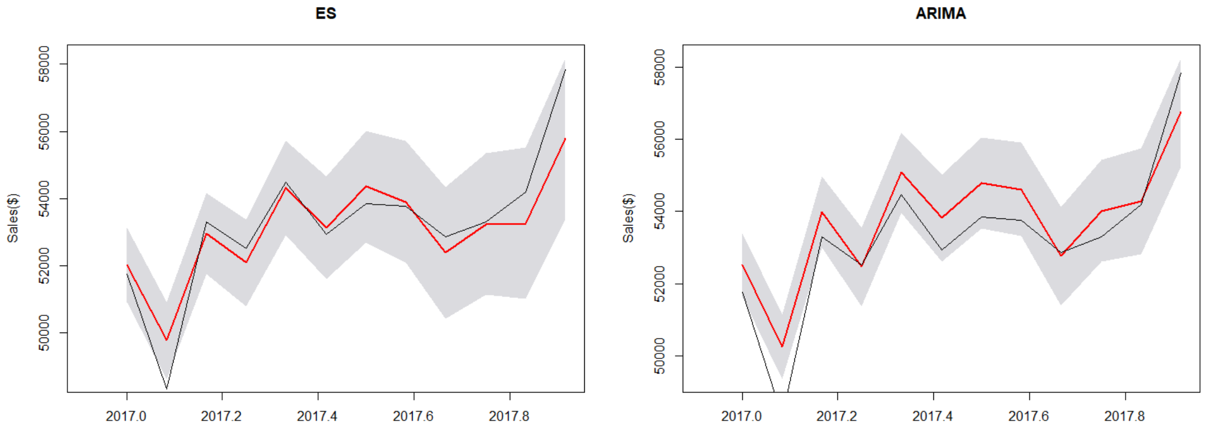

Fig. 16

We can conclude that both models performed in forecasting of grocery stores sales very good. Even thought that ARIMA fitted training time series better than ES, in the particular case of prediction for the year 2017 ES performed better. Interesting is also fact that ARIMA is more optimistic in its forecast.

References

[1] https://www.census.gov/econ/currentdata/dbsearch?program=MARTS&startYear=1992&endYear=2017&categories%5B%5D=4451&dataType=SM&geoLevel=US&notAdjusted=1&submit=GET+DATA&releaseScheduleId=

Appendix

```
library(forecast)
library(zoo)
library(astsa)

mape <- function(y, yhat)
  mean(abs((y - yhat)/y))

ts.data <- ts(scan("./dataset/dataset.txt"),start = c(1992,1),deltat =
1/12)

Sys.setlocale("LC_TIME", "English") #because of english name of days
ts.data.df <- data.frame(date = as.Date(as.yearmon(time(ts.data))),sales =
as.numeric(ts.data))

#whole time series chart
#basic chart
```

```
plot(ts.data,xlab="", ylab="Sales($)",main = "Grocery Stores Sales in the
U.S")

#part of the time series
#basic chart
par(mfrow=c(2,2))
#year 1992
plot(ts.data.df$date[1:12],ts.data.df$sales[1:12],type = "l",xlab="",
ylab="Sales($)", main="Year 1992")

#year 1993
plot(ts.data.df$date[13:24],ts.data.df$sales[13:24],type = "l",xlab="",
ylab="Sales($)", main="Year 1993")

#year 2002
plot(ts.data.df$date[109:120],ts.data.df$sales[109:120],type = "l",xlab="",
ylab="Sales($)", main="Year 2002")
#year 2003
plot(ts.data.df$date[121:132],ts.data.df$sales[121:132],type = "l",xlab="",
ylab="Sales($)", main="Year 2003")

par(mfrow=c(1))

#boxplot for months
ts.data.df$month <- factor(format(ts.data.df$date, format =
"%b"),month.abb, ordered = T)

#basic chart
boxplot(ts.data.df$sales~ts.data.df$month,ylab="Sales($)")

#decomposition
ts.components <- decompose(ts.data,"additive")
ts.components
plot(ts.components)

#removed seasonality
plot(ts.data-ts.components$seasonal,xlab="", ylab="",main = "Adjusted Time
Series without Seasonality")

#training/validation set split
ts.train <- window(ts.data,c(1992,1),c(2016,12))
ts.validation <- window(ts.data,c(2017,1),c(2017,12))

# #ES model
es.model <- ets(ts.train, model = "AAA")
es.forecast <- forecast(es.model, h=12, level =c(95))

#fitted values
plot(ts.train,ylab="Sales($)",main = "Fitted Time Series for ETS(A,Ad,A)")
lines(es.model$fitted,col="red")

#train set error
accuracy(es.model)

# Normality of residuals
par(mfrow=c(2,1))
hist(es.model$residuals,xlab="Residuals", main="Histogram of Residuals")
qqnorm(es.model$residuals,main="Normal Q-Q plot of Residuals")

#plot forecast with 95% prediction intervals - only last 10 years values
plot(es.forecast, showgap = F, include = 108, fcol = "red",
ylab="Sales($)", main = "Forecast from ETS(A,Ad,A) for Year 2017 with 95%
Prediction Interval")

#forecast vs reality validation set
plot(es.forecast, include = 1, fcol = "red", ylab="Sales($)", main =
"Forecast vs. Reality Year 2017")
lines(ts.validation, col="black")
```

```
#Validation set error
mape(ts.validation,es.forecast$mean)


#--------------------------------------------------
#ARIMA

#ACF plot for stationarity check stationary
acf2(ts.train)

# order of differencing needed
nsdiffs(ts.train, test = c("seas"))

acf2(diff(ts.train,12),144)

#help to estimate the best parametres for arima
auto.arima(ts.train)

#Arima
arima.model <- arima(ts.train,order = c(2,1,0), seasonal = list(order =
c(2,1,2), period = 12))


#fitted values
plot(ts.train,ylab="Sales($)",main = "Fitted Time Series for
ARIMA(2,1,0)(2,1,2)[12]")
lines(fitted(arima.model),col="red")

#train set error
accuracy(arima.model)

par(mfrow=c(2,1))
hist(arima.model$residuals,xlab="Residuals", main="Histogram of Residuals")
qqnorm(arima.model$residuals,main="Normal Q-Q plot of Residuals")

arima.forecast <- forecast(arima.model, h=12, level =c(95))

#plot forecast with 95% prediction intervals - only last 100 values
plot(arima.forecast, showgap = F, include = 108, fcol = "red",
ylab="Sales($)", main = "Forecast from ARIMA(2,1,0)(2,1,2)[12] for Year
2017 with 95% Prediction Interval")

#forecast vs reality validation set
plot(arima.forecast, include = 1, fcol = "red", ylab="Sales($)", main =
"Forecast vs. Reality Year 2017")
lines(ts.validation, col="black")

mape(ts.validation,arima.forecast$mean)

par(mfrow=c(1,2))
plot(es.forecast, include = 1, fcol = "red", ylab="Sales($)", main = "ES")
lines(ts.validation, col="black")
plot(arima.forecast, include = 1, fcol = "red", ylab="Sales($)", main =
"ARIMA")
lines(ts.validation, col="black")
```