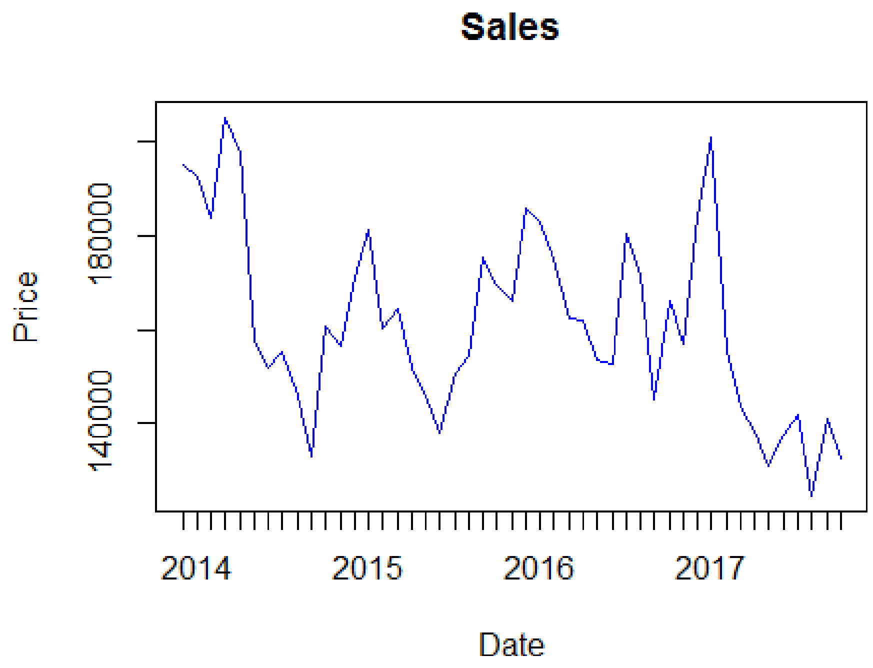
ARIMA – Sales Forecast (Fast Food)

Pedro Schneider

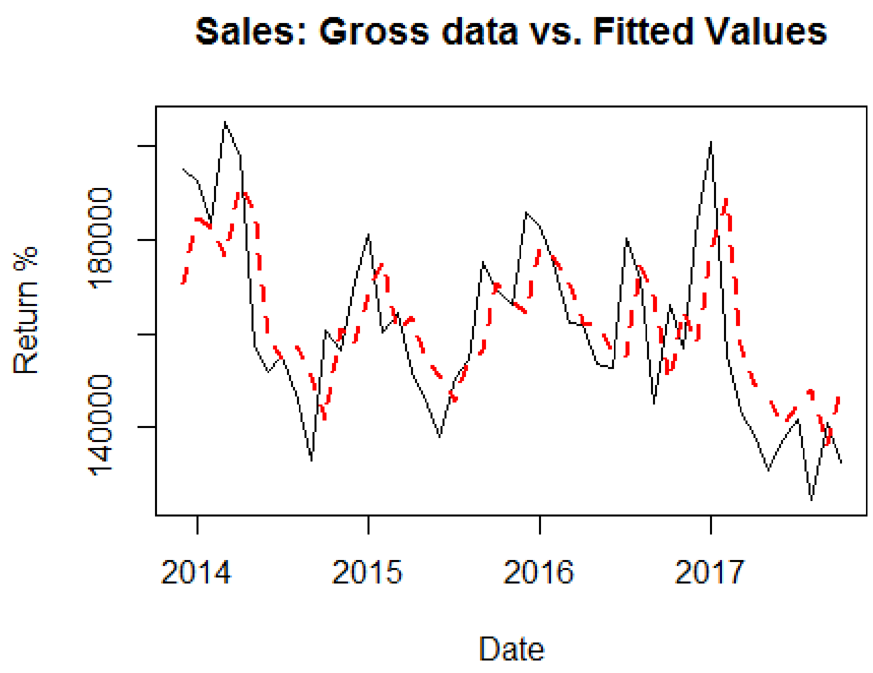
7 de dezembro de 2017

```r
# Time Series Analysis
# forecast, ARIMA

#install.packages("forecast")
library("forecast")

library("zoo")

# Loading the monthly sales data from 2014 to October 2017 in a zoo time
series
# The FUN parameter indicates that the time series is monthly (as.yearmon
)

precosM <- read.zoo("Vendas_Mes_2014_2017.csv", sep = ";", header = TRUE,
format = "%b/%y", FUN = as.yearmon)
head(precosM)

## dez 2013 jan 2014 fev 2014 mar 2014 abr 2014 mai 2014
##   195239   192402   183673   205154   197678   157795

View(precosM)

# Plot

plot(precosM, main = "Sales", col = "blue", xlab = "Date", ylab = "Price"
)
```




```
# Checking the number of periods in the data series
frequency(precosM)

## [1] 12

# Model Identification and estimation

# ARIMA models are sophisticated models that use correlations between obs
ervations at various times.
# ARIMA models show good results when the series is relatively long.
# We use the auto.arima function provided by the forecast package to iden
tify the optimal model and estimate the coefficients
# in a single step.
# stationary = TRUE, restrict search to stationary models.
# seasonal = FALSE restricts the search to non-seasonal models.
# Akaike information criteria as the relative quality measure to be used
in the template selection.
modM <- auto.arima(precosM, stationary = TRUE, seasonal = TRUE, ic = "aic
")


# Viewing the template
modM

## Series: precosM
## ARIMA(1,0,0) with non-zero mean
##
```

```
## Coefficients:
##          ar1        mean
##       0.6857  161669.022
## s.e.  0.1116    6582.729
##
## sigma^2 estimated as 229504101:  log likelihood=-518.39
## AIC=1042.79   AICc=1043.35   BIC=1048.34

# If the model contains coefficients that are insignificant, we can estim
ate the new model using the arima function
# with the fixed argument.
confint(modM)

##                  2.5 %       97.5 %
## ar1       4.670892e-01 9.044087e-01
## intercept 1.487671e+05 1.745709e+05

# To evaluate how well the model represents the sample data, we can plot
the gross monthly returns
# (the fine black solid line) versus the adjusted values (dashed red dash
ed line).

plot(modM$x, lty = 1, main = "Sales: Gross data vs. Fitted Values", ylab
= "Return %", xlab = "Date")
lines(fitted(modM), lty = 2,lwd = 2, col = "red")
```



```
# Calculating other measures of accuracy
accuracy(modM)

##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -685.7843 14823.56 11838.37 -1.246786 7.373249 0.6470892
##                      ACF1
## Training set -0.001787769

# Predicting monthly return 3 months ahead
# To trace the forecast with standard errors, we can use the following co
mmand:
predict(modM, n.ahead = 3)

## $pred
##           Jan Feb Mar Apr May Jun Jul Aug Sep Oct      Nov      Dec
## 2017                                              141716.5 147986.6
## 2018 152286.3
##
## $se
##           Jan Feb Mar Apr May Jun Jul Aug Sep Oct      Nov      Dec
## 2017                                              15149.39 18369.23
## 2018 19702.30

# Plot
plot(forecast(modM))
```