Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 3

Professor Jeffrey Yau

March 18, 2018

Introduction

Load packages set some formatting preferences

```r
pkg <- c('knitr', 'Hmisc', 'ggcorrplot', 'car',
         'dplyr', 'ggplot2', 'jtools', 'readr')
invisible(lapply(pkg, require, character.only = T))
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Question 1: E-Commerce Retail Sales as a Percent of Total Sales- Build a Seasonal ARIMA model and generate quarterly forecast for 2017

Load data and quality-check the raw data

```r
# Load csv file as df
df_q1 <- read.csv("ECOMPCTNSA.csv", header = TRUE, sep = ",")
str(df_q1)
```

```
## 'data.frame':    69 obs. of  2 variables:
##  $ DATE      : Factor w/ 69 levels "1999-10-01","2000-01-01",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ ECOMPCTNSA: num  0.7 0.8 0.8 0.9 1.1 1.1 1 1 1.3 1.3 ...
```

```r
# View(df_q1) names(df_q1)
head(df_q1)
```

```
##         DATE ECOMPCTNSA
## 1 1999-10-01        0.7
## 2 2000-01-01        0.8
## 3 2000-04-01        0.8
## 4 2000-07-01        0.9
## 5 2000-10-01        1.1
## 6 2001-01-01        1.1
```

```r
describe(df_q1)
```

```
## df_q1
##
##  2  Variables      69  Observations
## ---------------------------------------------------------------------------
## DATE
##        n  missing distinct
##       69        0       69
##
## lowest : 1999-10-01 2000-01-01 2000-04-01 2000-07-01 2000-10-01
## highest: 2015-10-01 2016-01-01 2016-04-01 2016-07-01 2016-10-01
## ---------------------------------------------------------------------------
## ECOMPCTNSA
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       69        0       50        1    3.835    2.524     0.94     1.10
##      .25      .50      .75      .90      .95
##     2.00     3.60     5.30     6.92     7.70
##
## lowest : 0.7 0.8 0.9 1.0 1.1, highest: 7.0 7.5 7.7 8.7 9.5
## ---------------------------------------------------------------------------
```

```
# Create an R time-series object with our quarterly data
# starting with final quarter of 1999
q1_ts <- ts(df_q1$ECOMPCTNSA, frequency = 4, start = c(1999,
    4))
str(q1_ts)
```

```
##  Time-Series [1:69] from 2000 to 2017: 0.7 0.8 0.8 0.9 1.1 1.1 1 1 1.3 1.3 ...
```

```
head(q1_ts)
```

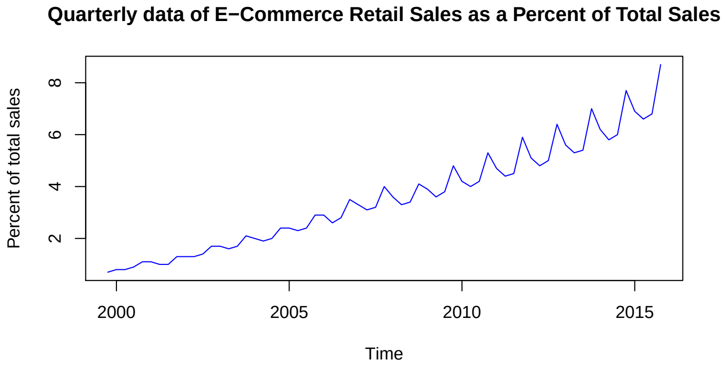
```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1999                 0.7
## 2000  0.8  0.8  0.9  1.1
## 2001  1.1
```

We see that there are no missing values and that we are working with quarterly time series data. No anomalies are detected and there is not potential for top or bottom code. On gross visual inspection of the time series, we see that the starting values are all less than 1, with the later values all being greater than 7. This leads us to already suspect we are not dealing with a stationary series. We cannot make a confident comment on seasonality without further EDA for visualization.

EDA

Plot the data

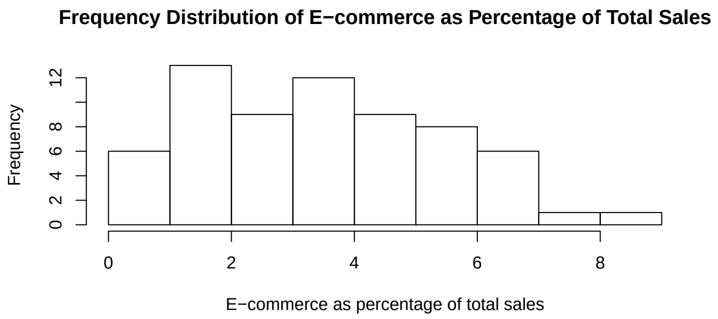
```
# For modeling purposes we keep data between 1999 and 2015 as
# our training data; we hold-out 2016 as test data
q1_ts_train <- q1_ts[time(q1_ts) >= 1999 & time(q1_ts) < 2016]
q1_ts_train <- ts(q1_ts_train, frequency = 4, start = c(1999,
    4))
q1_ts_test <- q1_ts[time(q1_ts) >= 2016]
q1_ts_test <- ts(q1_ts_test, frequency = 4, start = c(2016, 1))
# Plot the training data
plot.ts(q1_ts_train, main = "Quarterly data of E-Commerce Retail Sales as a Percent of Total Sales",
    ylab = "Percent of total sales", col = "blue")
```

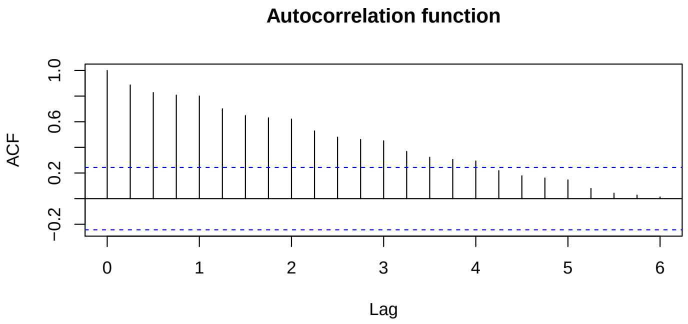


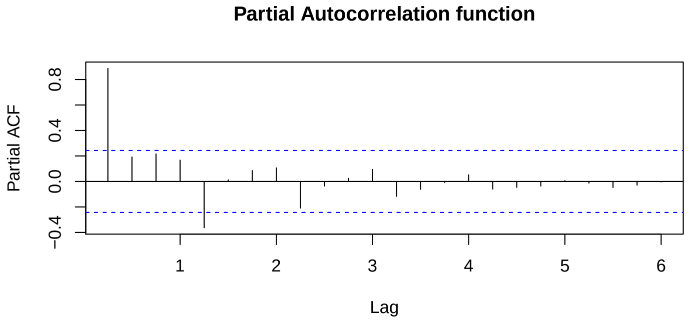
We are now able to clearly see that the e-commernce retail sales time series is not stationary in the mean and exhibits seasonality. We will attempt to stationarize our time series via differencing.

Examine the ACF/PACF to determine if an AR(p) or MA(q) model is appropriate

```
# Plot frequency distribution, ACF, PACF
hist(q1_ts_train, main = "Frequency Distribution of E-commerce as Percentage of Total Sales",
    xlab = "E-commerce as percentage of total sales")
acf(q1_ts_train, main = "Autocorrelation function", lag.max = 24)
pacf(q1_ts_train, main = "Partial Autocorrelation function",
    lag.max = 24)
```





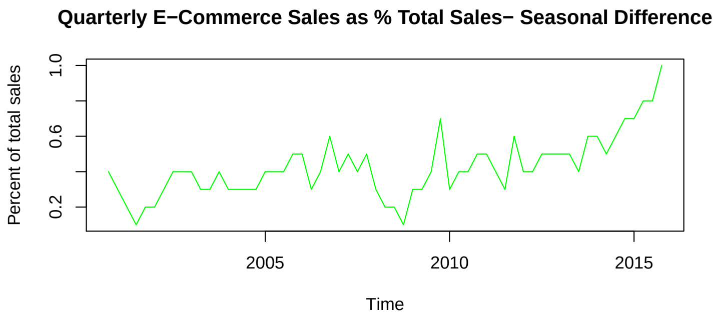


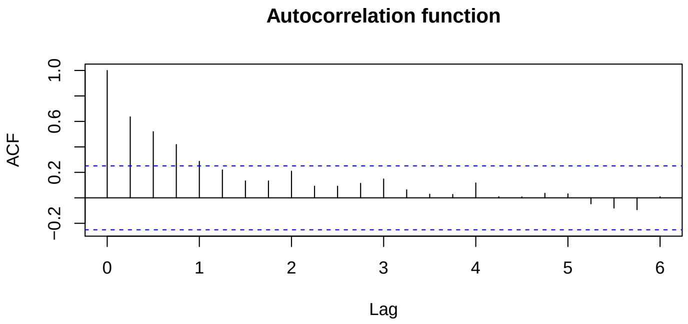
Although it has been dictated that we are to create a SARIMA model, we still need to check that this model is appropriate.We see that the autocorrelations are significant for a large number of lag (16 quarters). This slow decay in the autocorrelations is evidence of a trend in the data, thus telling us that the time series is not stationary. The gradual decay without any spikes at seasonal intervals tells us that we will need a non-seasonal AR term p but will not need a seasonal AR term P. We see that the partial autocorrelation plot has a significant spike at a lag of 1 quarter and at 4 quarters. We see that our PACF has a somewhat abrupt drop-off non-seasonally following lag-1 and but that there are spikes in the PACF at an annual lag (multiples of lag-4 given our quarterly data). This leads us to believe that our model will not require a non-seasonal MA term q but will require a seasonal MA term Q. The histogram of our data shows that the e-commerce as percentage of total sales is fairly normally distributed withpositive skew; however, this tells us nothing about how the data are related in time.

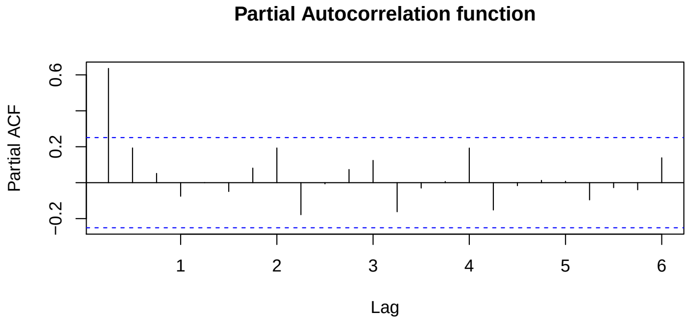
Difference the data to impose stationarity

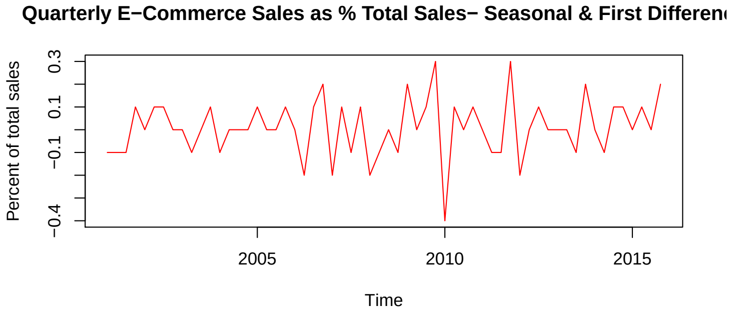
We have demonstrated evidence of both trend and seasonality in our time series. To impose stationarity, we will first apply a seasonal difference to the data and then re-evaluate the trend. If a trend remains, then we will take first differences.

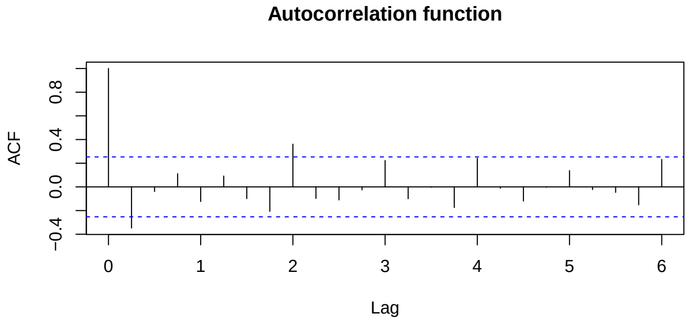
```
# Impose seasonal difference at one-year (4 quarters)
q1_ts_train_seasonal_diff = diff(q1_ts_train, 4)
plot.ts(q1_ts_train_seasonal_diff, main = "Quarterly E-Commerce Sales as % Total Sales- Seasonal Difference",
    ylab = "Percent of total sales", col = "green")
acf(q1_ts_train_seasonal_diff, main = "Autocorrelation function",
    lag.max = 24)
pacf(q1_ts_train_seasonal_diff, main = "Partial Autocorrelation function",
    lag.max = 24)
# Impose additional first-difference
q1_ts_train_seasonal_and_first_diff = diff(q1_ts_train_seasonal_diff,
    1)
plot.ts(q1_ts_train_seasonal_and_first_diff, main = "Quarterly E-Commerce Sales as % Total Sales- Seasonal & First Differenc
    ylab = "Percent of total sales", col = "red")
acf(q1_ts_train_seasonal_and_first_diff, main = "Autocorrelation function",
    lag.max = 24)
pacf(q1_ts_train_seasonal_and_first_diff, main = "Partial Autocorrelation function",
    lag.max = 24)
```

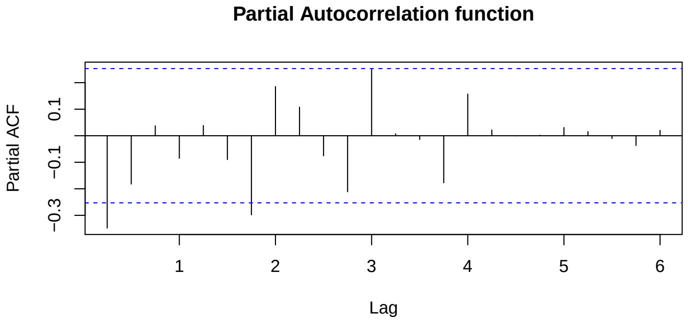












After creating a seasonal-differenced series, the series still appears to be non-stationary. For stationary time series, the ACF drops to zero relatively quickly, while for non-stationary data the ACF decreases slowly. We see improvement here as compared to out initial non-differenced model, but we have still not imposed stationarity. This provides evidence that we need to impose a first-difference. After taking a first-difference we see that the seasonal-differenced and first-differenced series is stationary. Overall we do not see evidence that the volatility is increasing over time, so we do not take a difference in log to stabilize the series.

Order Identification

The spike in the ACF at a lag of 1 quarter suggests a nonseasonal MA(1) (q=1) component and the spikes at intervals of 4 quarters of lag suggest a seasonal MA(1) (Q=1). Additionally, the spike at at a lag of 1 quarter in the PACF and the spikes at intervals of 4 quarters of lag tells us that a a nonseasonal AR(1) (p=1) component and a seasonal AR(1) (P=1) component are appropriate for our initial model. Therefore, our initial model will be of the form $ARIMA(1,1,1)(1,1,1)_4$

Model Creation: Build a Seasonal ARIMA model and generate quarterly forecast for 2017

We first start by building a model with our estimated components from our prior analysis. Next, we will see if an interative method comparing difference combinations of component values as well as the auto.arima function all agree with our model having the lowest AIC/BIC.

```
q1_ts_train_seasonal_and_first_diff.fit <- Arima(q1_ts_train_seasonal_and_first_diff,
    order = c(1, 1, 1), seasonal = c(1, 1, 1))
summary(q1_ts_train_seasonal_and_first_diff.fit)
```

```
## Series: q1_ts_train_seasonal_and_first_diff
## ARIMA(1,1,1)(1,1,1)[4]
##
## Coefficients:
```

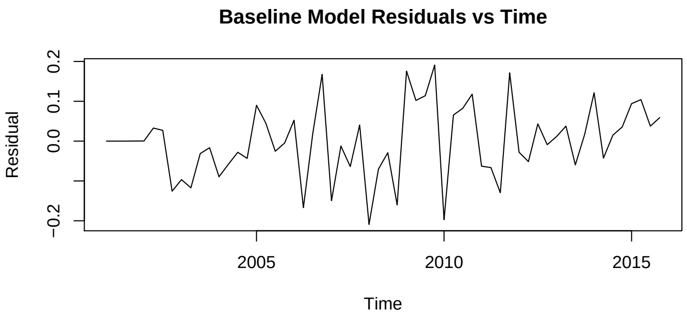
```
##          ar1    ma1   sar1   sma1
##        -0.36  -0.93  -0.45  -0.91
## s.e.   0.13   0.12   0.13   0.22
##
## sigma^2 estimated as 0.00965:  log likelihood=44
## AIC=-78   AICc=-76   BIC=-67
##
## Training set error measures:
##                    ME  RMSE   MAE MPE MAPE MASE   ACF1
## Training set -0.0012 0.091 0.07 NaN  Inf 0.46 -0.019
```

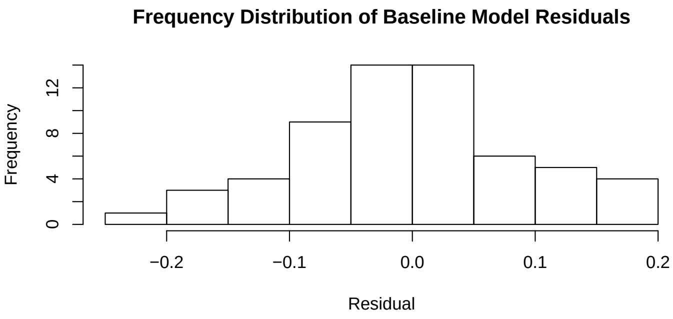
```
plot.ts(q1_ts_train_seasonal_and_first_diff.fit$resid, main = "Baseline Model Residuals vs Time",
    ylab = "Residual", col = "black")
hist(q1_ts_train_seasonal_and_first_diff.fit$resid, main = "Frequency Distribution of Baseline Model Residuals",
    xlab = "Residual")
acf(q1_ts_train_seasonal_and_first_diff.fit$resid, main = "Autocorrelation function",
    lag.max = 24)
pacf(q1_ts_train_seasonal_and_first_diff.fit$resid, main = "Partial Autocorrelation function",
    lag.max = 24)
shapiro.test(q1_ts_train_seasonal_and_first_diff.fit$resid)
```

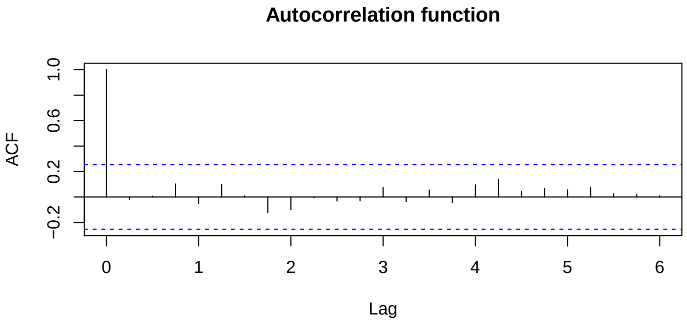
```
##
##  Shapiro-Wilk normality test
##
## data:  q1_ts_train_seasonal_and_first_diff.fit$resid
## W = 1, p-value = 0.8
```

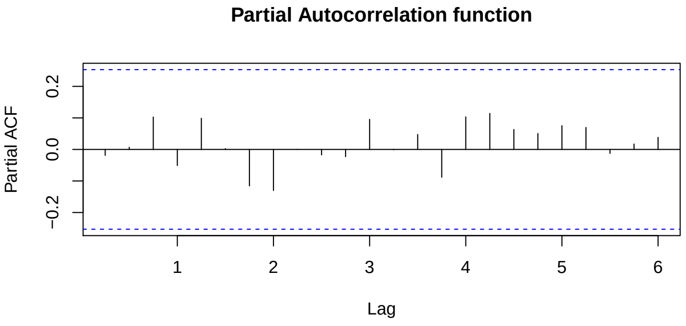
```
qqnorm(q1_ts_train_seasonal_and_first_diff.fit$resid)
qqline(q1_ts_train_seasonal_and_first_diff.fit$resid)
Box.test(q1_ts_train_seasonal_and_first_diff.fit$resid, type = "Ljung-Box")
```

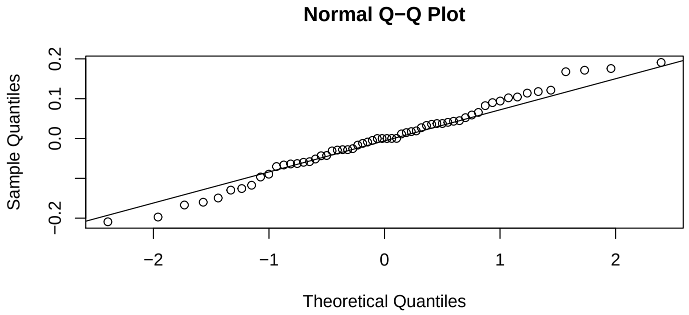
```
##
##  Box-Ljung test
##
## data:  q1_ts_train_seasonal_and_first_diff.fit$resid
## X-squared = 0.02, df = 1, p-value = 0.9
```











Now we will look at other values for p,d,q,P,D,Q via an iterative method.

Lastly, we look at the auto.arima function to see if it agrees with our model thus far.

Fit Evaluation

Question 2: data_2018Spring_MTS.txt

Load data

```
# df_q2 <- read.csv('correlate-flight_prices.csv', header =
# TRUE, sep=',')
```

EDA

Order Identification

Model Creation

Fit Evaluation

Conclusion