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

```
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

```
# 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 ...
```

```
# 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
```

```
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
## ---------------------------------------------------------------------------
```

```r
# 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 ...
```

```r
head(q1_ts)
```

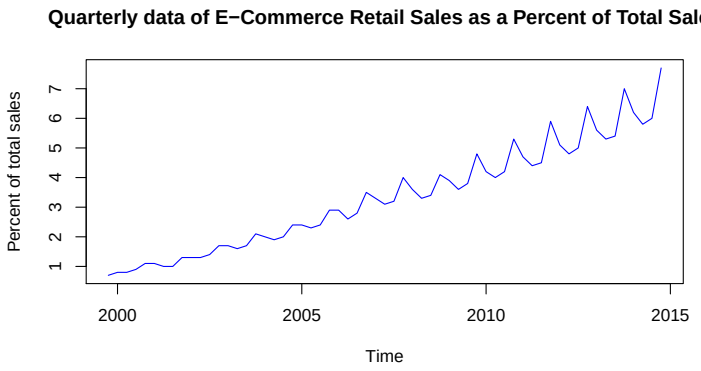
```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1999                 0.7
## 2000  0.8  0.8  0.9  1.1
## 2001  1.1
```

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

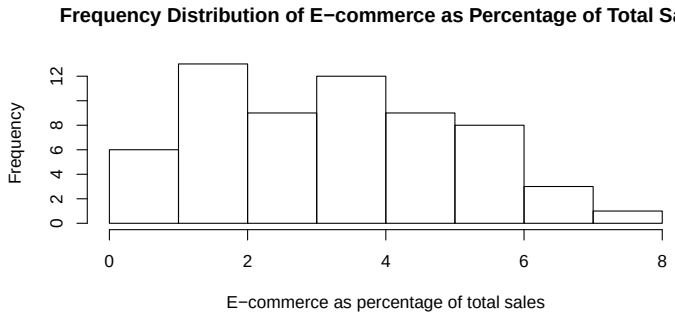
```r
# For modeling purposes we keep data between 1999 and 2015 as
# our training data; we hold-out 2015-2016 as test data
q1_ts_train <- q1_ts[time(q1_ts) >= 1999 & time(q1_ts) < 2015]
q1_ts_train <- ts(q1_ts_train, frequency = 4, start = c(1999,
    4))
q1_ts_test <- q1_ts[time(q1_ts) >= 2015]
q1_ts_test <- ts(q1_ts_test, frequency = 4, start = c(2015, 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")
```

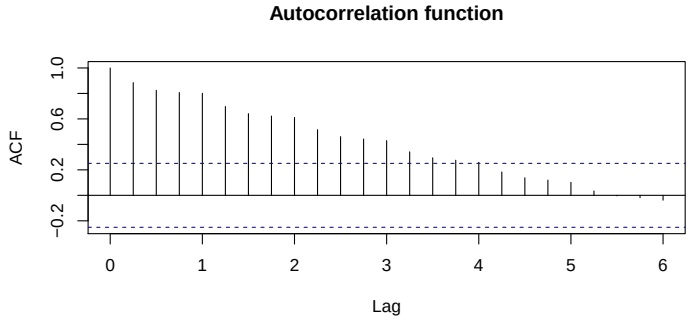


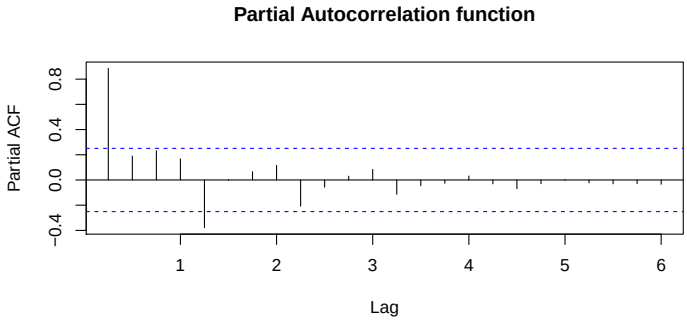
We are now able to clearly see that the e-commernce retail sales time series is not stationary in the mean and exhibits seasonality. We will attempt to stationarize our time series via differencing.

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

```r
# 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)
```





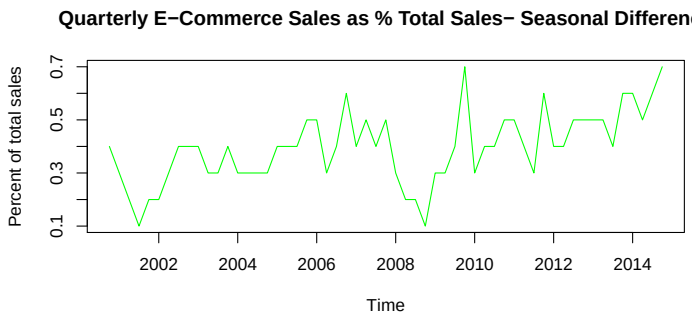


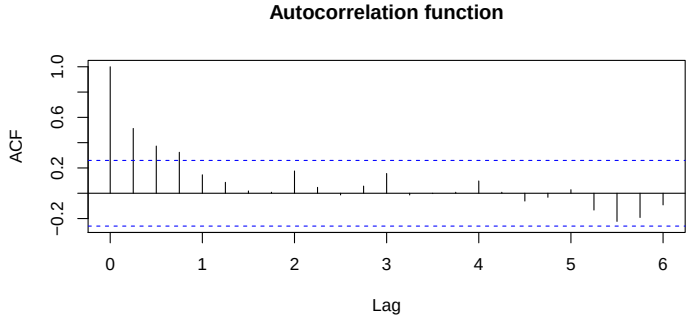
Although it has been dictated that we are to create a SARIMA model, we still need to check that this model is appropriate.We see that the autocorrelations are significant for a large number of lag (16 quarters). This slow decay in the autocorrelations is evidence of a trend in the data, thus telling 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 with positive skew; however, this tells us nothing about how the data are related in time.

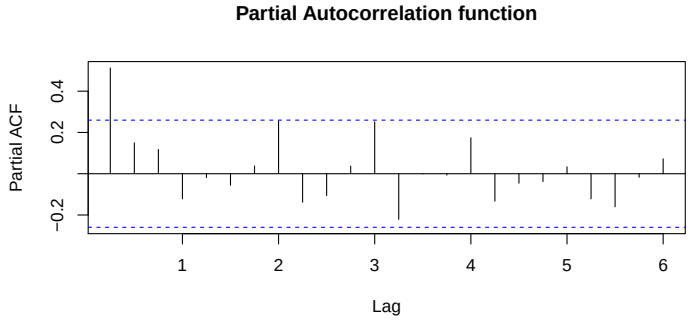
Difference the data to impose stationarity

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)
```

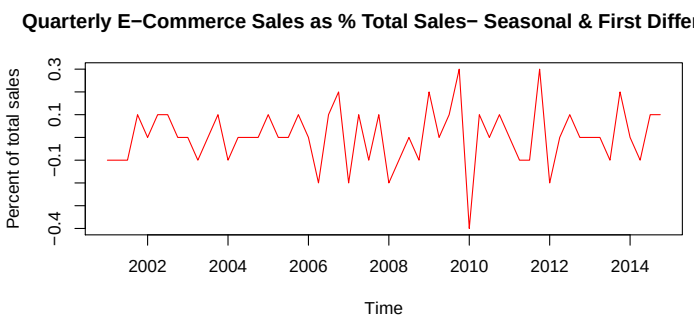


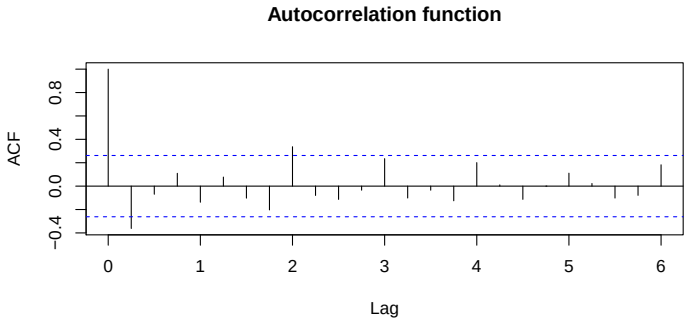


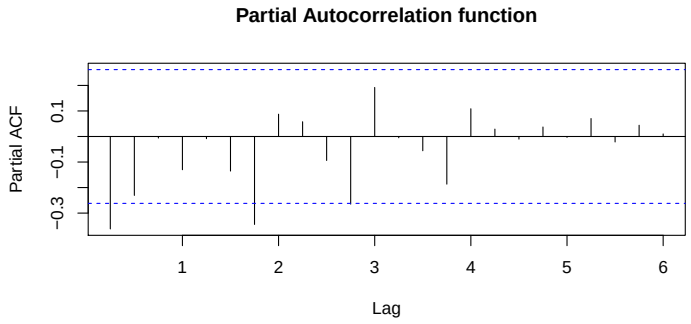


After creating a seasonal-differenced series, the series still appears to be non-stationary. For stationary time series, the ACF drops to zero relatively quickly, while for non-stationary data the ACF decreases slowly. We see improvement here as compared to out initial non-differenced model, but we have still not imposed stationarity. This provides evidence that we need to impose a first-difference.

```
# 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 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 out to approximately lag 16 suggest a seasonal MA(4) (Q=4). 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 $ARIMA(1,1,1)(1,1,4)_4$

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.fit <- Arima(q1_ts_train, order = c(1, 1, 1), seasonal = c(1,
    1, 4))
summary(q1_ts_train.fit)
```

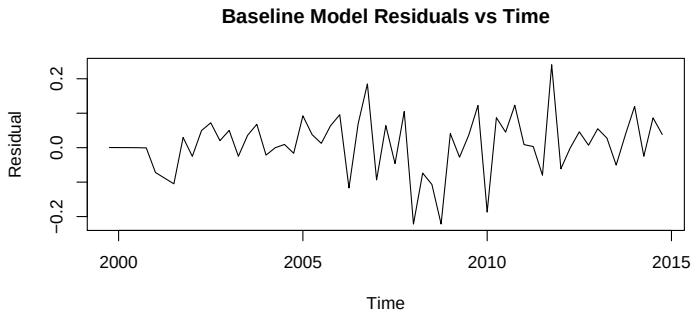
```
## Series: q1_ts_train
## ARIMA(1,1,1)(1,1,4)[4]
##
## Coefficients:
##         ar1     ma1  sar1  sma1  sma2    sma3   sma4
##       0.59  -0.933  1.00  -1.4  0.63  -0.087  -0.11
## s.e.  0.19   0.092  0.01   0.2  0.25   0.220   0.15
##
## sigma^2 estimated as 0.00901:  log likelihood=52
## AIC=-87   AICc=-84   BIC=-71
##
## Training set error measures:
##                     ME  RMSE   MAE   MPE MAPE MASE  ACF1
## Training set 0.0085 0.085 0.063 0.003  2.2 0.16 -0.14
```

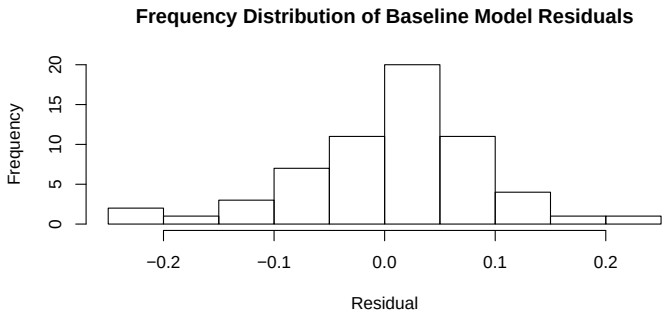
```
plot.ts(q1_ts_train.fit$resid, main = "Baseline Model Residuals vs Time",
    ylab = "Residual", col = "black")
hist(q1_ts_train.fit$resid, main = "Frequency Distribution of Baseline Model Residuals",
    xlab = "Residual")
acf(q1_ts_train.fit$resid, main = "Autocorrelation function",
    lag.max = 24)
shapiro.test(q1_ts_train.fit$resid)
```

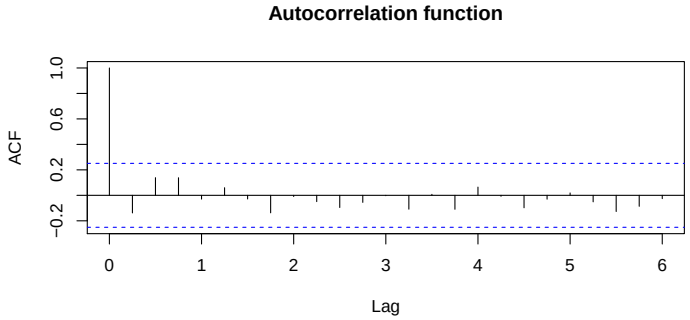
```
##
##  Shapiro-Wilk normality test
##
## data:  q1_ts_train.fit$resid
## W = 1, p-value = 0.08
```

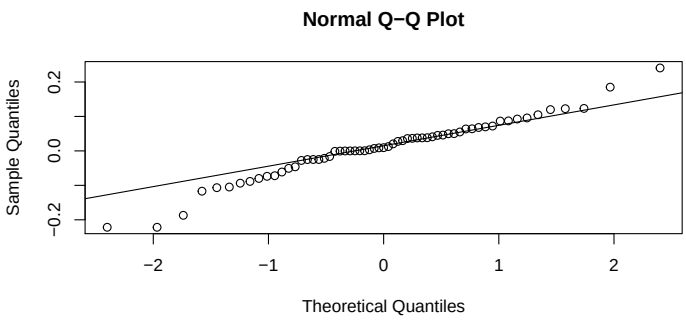
```
qqnorm(q1_ts_train.fit$resid)
qqline(q1_ts_train.fit$resid)
Box.test(q1_ts_train.fit$resid, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  q1_ts_train.fit$resid
## X-squared = 1, df = 1, p-value = 0.3
```









We see that our baseline $ARIMA(1,1,1)(1,1,4)_4$ model generates an AIC of -78. We conduct a Shapiro-Wilk normality test for our residuals, which shows that we fail to reject the null hypothesis that the population from which our residuals are derived is normal. We also conduct a Box-Ljung test. The p-value is large, which means we do not suspect that there is non-zero autocorrelation within the lags. Looking at the plot of the residuals, we see that the variance is increased somewhat at the center of the plot but that overall the variance is not increasing over time. The histogram of our residuals shows a somewhat normal distribution with a positive skew. Looking at the ACF plot, we do not see evidence of autocorrelation in the residuals, which suggests that there is not information that has not been accounted for in the model. Our Q-Q plot supports that our residuals are normally distributed.

Now we will look at other values for p,d,q,P,D,Q via an iterative method. We will impose both first-order non-seasonal and first-order seasonal minimum differencing on our iterative search here, per our prior EDA.

```
mod_AIC <- 0
for (P in 0:2) {
    for (Q in 0:2) {
        for (D in 1:4) {
            for (p in 0:2) {
                for (q in 0:2) {
                  for (d in 1:2) {
                    mod <- Arima(q1_ts_train, order = c(p, d,
                      q), seasonal = list(order = c(P, D, Q),
                      4), method = "ML")
                    if (mod$aic < mod_AIC) {
                      mod_AIC <- mod$aic
                      best_params <- c(p, d, q, P, D, Q)
                    }
                  }
                }
            }
        }
    }
}
print(c(best_params, mod_AIC))
```

```
## [1]   0   1   1   1   1   2 -93
```

Interestingly, our iterative method to determine the values of p,d,q,P,D,Q in our SARIMA model that are associated with the lowest AIC value tells us that the model with the lowest AIC is $ARIMA(0,1,1)(1,1,2)_4$. The AIC for this model is lower than for our baseline model (-93 vs -78). We now look at the residuals for this model.

```
# Residual diagnostics on iterative model
q1_ts_train_it.fit <- Arima(q1_ts_train, order = c(0, 1, 1),
    seasonal = c(1, 1, 2))
summary(q1_ts_train_it.fit)
```

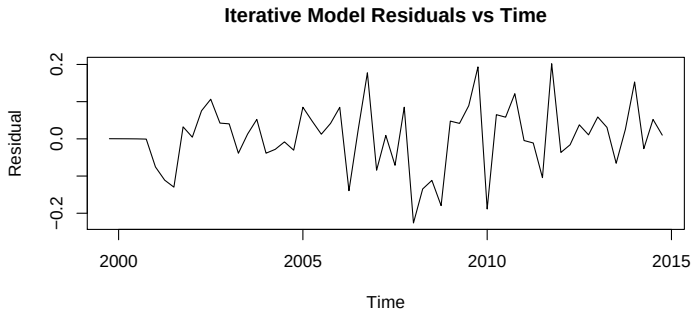
```
## Series: q1_ts_train
## ARIMA(0,1,1)(1,1,2)[4]
##
## Coefficients:
##         ma1   sar1   sma1  sma2
##       -0.44  0.943  -1.33  0.56
## s.e.   0.13  0.071   0.15  0.14
##
## sigma^2 estimated as 0.00897:  log likelihood=52
## AIC=-93   AICc=-92   BIC=-83
##
## Training set error measures:
##                     ME  RMSE   MAE   MPE MAPE MASE   ACF1
## Training set 0.0045 0.087 0.065 -0.15  2.4 0.17 -0.008
```

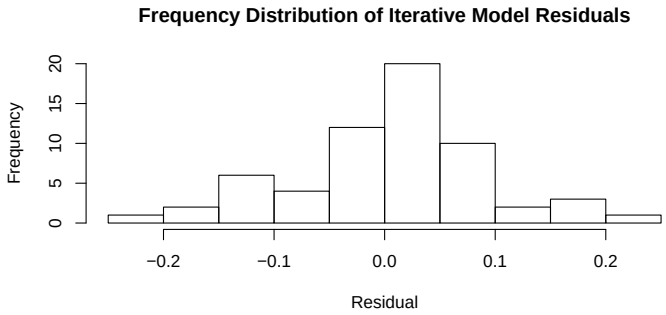
```r
plot.ts(q1_ts_train_it.fit$resid, main = "Iterative Model Residuals vs Time",
    ylab = "Residual", col = "black")
hist(q1_ts_train_it.fit$resid, main = "Frequency Distribution of Iterative Model Residuals",
    xlab = "Residual")
acf(q1_ts_train_it.fit$resid, main = "Autocorrelation function",
    lag.max = 24)
shapiro.test(q1_ts_train_it.fit$resid)
```

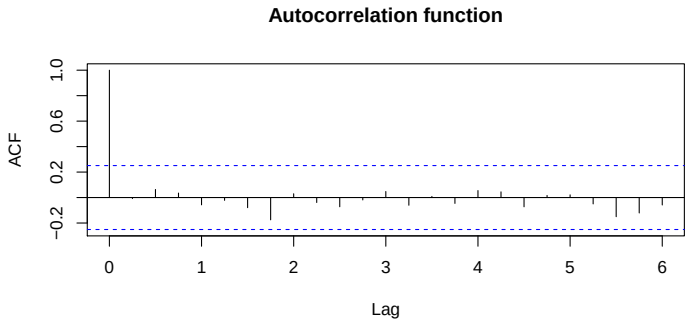
```
##
##  Shapiro-Wilk normality test
##
## data:  q1_ts_train_it.fit$resid
## W = 1, p-value = 0.2
```

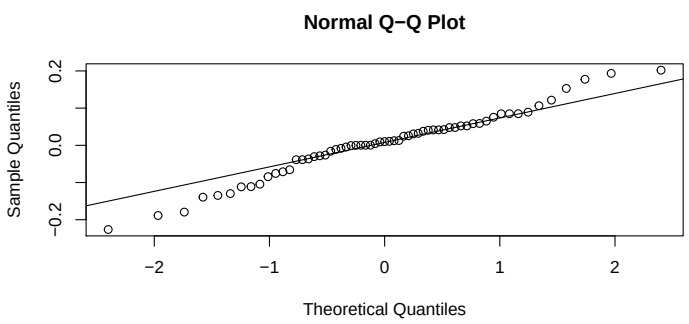
```r
qqnorm(q1_ts_train_it.fit$resid)
qqline(q1_ts_train_it.fit$resid)
Box.test(q1_ts_train_it.fit$resid, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  q1_ts_train_it.fit$resid
## X-squared = 0.004, df = 1, p-value = 0.9
```









Similar to our baseline model, the Shapiro-Wilk normality test shows evidence that our residuals are derived from a normal population. The Box-Ljung test shows that there is evidence of zero autocorrelation within the lags. Overall the variance is not increasing over time. The histogram of our residuals shows a fairly normal distribution. Looking at the ACF plot, we do not see evidence of autocorrelation in the residuals. Our Q-Q plot supports that our residuals are normally distributed.

We will proceed with the auto-arima() function to also provide evidence for the order of the model.

```r
auto.arima(q1_ts_train, seasonal = TRUE)
```

```
## Series: q1_ts_train
## ARIMA(0,1,1)(0,1,0)[4]
##
## Coefficients:
##          ma1
##       -0.57
## s.e.   0.16
##
## sigma^2 estimated as 0.013:  log likelihood=42
## AIC=-81   AICc=-81   BIC=-77
```

We see that the auto.arima function has determined that the model, as evaluated with stepwise argument on, that yields the lowest AIC is different from both out proposed baseline model and from our model generated by the iterative procedure. The auto.arima best model per AIC is $ARIMA(0,1,1)(0,1,0)_4$ We now look at the residuals for the auto.arima generated model.

```
# Residual diagnostics on auto.arima model
q1_ts_train_auto.fit <- Arima(q1_ts_train, order = c(0, 1, 1),
    seasonal = c(0, 1, 0))
summary(q1_ts_train_auto.fit)
```

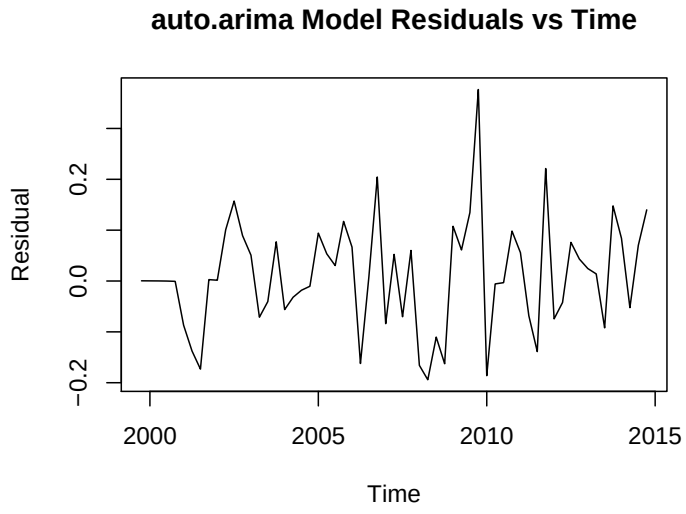
```
## Series: q1_ts_train
## ARIMA(0,1,1)(0,1,0)[4]
##
## Coefficients:
##          ma1
##       -0.57
## s.e.   0.16
##
## sigma^2 estimated as 0.013:  log likelihood=42
## AIC=-81   AICc=-81   BIC=-77
##
## Training set error measures:
##                     ME RMSE   MAE   MPE MAPE MASE  ACF1
## Training set 0.0095 0.11 0.083 -0.13    3 0.21 0.064
```

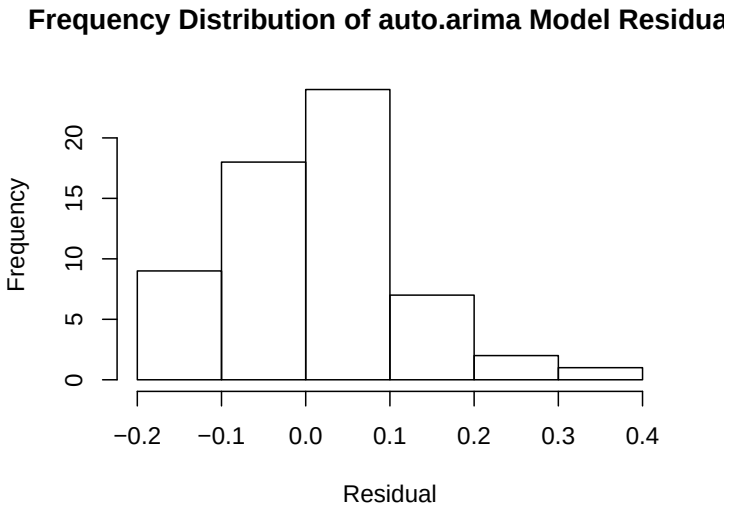
```
plot.ts(q1_ts_train_auto.fit$resid, main = "auto.arima Model Residuals vs Time",
    ylab = "Residual", col = "black")
hist(q1_ts_train_auto.fit$resid, main = "Frequency Distribution of auto.arima Model Residuals",
    xlab = "Residual")
acf(q1_ts_train_auto.fit$resid, main = "Autocorrelation function",
    lag.max = 24)
shapiro.test(q1_ts_train_auto.fit$resid)
```

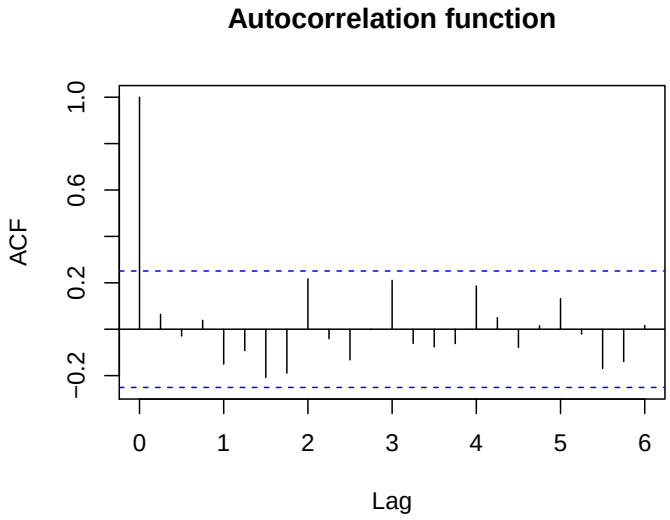
```
##
##  Shapiro-Wilk normality test
##
## data:  q1_ts_train_auto.fit$resid
## W = 1, p-value = 0.2
```

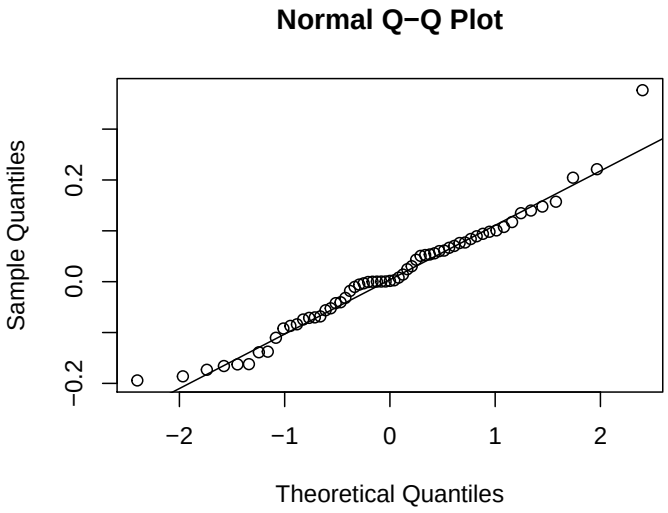
```
qqnorm(q1_ts_train_auto.fit$resid)
qqline(q1_ts_train_auto.fit$resid)
Box.test(q1_ts_train_auto.fit$resid, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  q1_ts_train_auto.fit$resid
## X-squared = 0.3, df = 1, p-value = 0.6
```









Similar to our baseline model and to our model generated by our iterative method, the Shapiro-Wilk normality test shows evidence that our residuals are derived from a normal population. The Box-Ljung test shows that there is evidence of zero autocorrelation within the lags. Overall the variance is not increasing over time. The histogram of our residuals shows a fairly normal distribution with a positive skew. Looking at the ACF plot, we do not see evidence of autocorrelation in the residuals. Our Q-Q plot supports that our residuals are normally distributed.

Fit Evaluation

We already looked at the in-sample performance of our candidate models by looking at their residuals. Our "iterative" SARIMA model $ARIMA(0,1,1)(1,1,2)_4$ had the lowest AIC at -93. Our baseline model $ARIMA(1,1,1)(1,1,4)_4$ had an intermediate AIC at -87. Our auto-arim model $ARIMA(0,1,1)(0,1,0)_4$ had the highest AIC at -81. Now, we look at the out-of-sample performance of our candidate models by forecasting the quarterly retail sales in 2015 and 2016. We will determine which model has the lowest forecasting error.

```
# Out-of-sample performance: Forecasting the quarterly
# E-Commerce retail sales in 2015 and 2016 with our models
# There are 8 observations in the test set (2015-2016), thus
# we generate an 8-step ahead forecast from the training set.
library(forecast)
library(tseries)
forecast_base <- forecast(q1_ts_train.fit, h = 8)
plot(forecast_base)
forecast_it <- forecast(q1_ts_train_it.fit, h = 8)
plot(forecast_it)
forecast_auto <- forecast(q1_ts_train_auto.fit, h = 8)
plot(forecast_auto)
# Calculate RMSE
compare.forecast.df <- data.frame(forecast_base = forecast_base$mean,
    forecast_it = forecast_it$mean, forecast_auto = forecast_auto$mean,
    testdata = q1_ts_test)
# Calculate RMSE
calculate_rmse <- function(fcast, test) {
    rmse <- sqrt(mean((fcast - test)^2))
}
print(calculate_rmse(compare.forecast.df$forecast_base, compare.forecast.df$testdata))
```

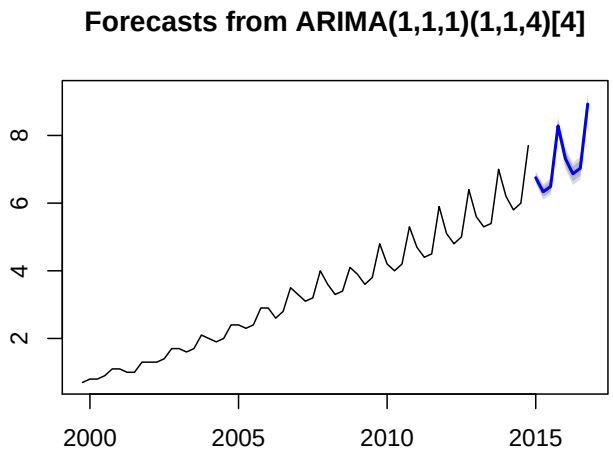
```
## [1] 0.46
```

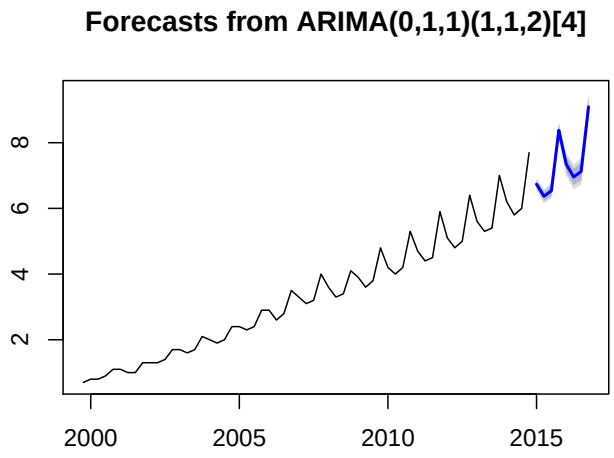
```
print(calculate_rmse(compare.forecast.df$forecast_it, compare.forecast.df$testdata))
```

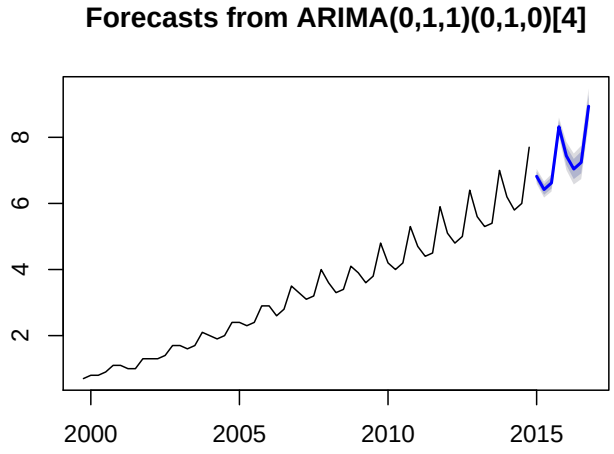
```
## [1] 0.38
```

```
print(calculate_rmse(compare.forecast.df$forecast_auto, compare.forecast.df$testdata))
```

```
## [1] 0.36
```







We see that model dictated by the auto-arim method $ARIMA(0,1,1)(0,1,0)_4$ actually has the lowest RMSE despite having the highest AIC of the three candidate models. Lower values of RMSE indicate better fit, seen here as the lowest forecasting error for the out-of-sample data from 2015-2016. Thus, we choose $ARIMA(0,1,1)(0,1,0)_4$ as our model to forecast for 2017.

```
# Forecast beyond the observed time-period of the series:
# generate quarterly forecast for 2017 Given the AR and MA
# components in our model, we cannot hold-out 2015-2016 in
# predicting 2017.  Therefore we need to use the auto.arima
# again to generate a predictive model for 2017 using our
# entire initial time series through 2016.
auto.arima(q1_ts, seasonal = TRUE)
```

```
## Series: q1_ts
## ARIMA(0,1,1)(0,1,0)[4]
##
## Coefficients:
##          ma1
##       -0.47
## s.e.   0.12
##
## sigma^2 estimated as 0.013:  log likelihood=49
## AIC=-93   AICc=-93   BIC=-89
```

```
q1_ts_auto.fit <- Arima(q1_ts, order = c(0, 1, 1), seasonal = c(0,
    1, 0))
q1_ts_pred_2017 <- predict(q1_ts_auto.fit, n.ahead = 4, ci = 0.95)
q1_ts_pred_2017
```

```
## $pred
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2017  8.5  8.3  8.5 10.3
##
## $se
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2017 0.11 0.13 0.14 0.15
```

```
# Alternate [INCORRECT] method: Extend train model another 4
# quarters into 2017 q1_ts_train_pred_2017 <-
# predict(q1_ts_train_auto.fit, n.ahead = 12, ci = 0.95)
# q1_ts_train_pred_2017
```

We see that the quarterly forecasted percentages of E-commerce sales of total sales are 8.5, 8.3, 8.5, and 10.3.

Question 2: data_2018Spring_MTS.txt

Load data

```
# Load txt file as df
df_q2 <- read.table("data_2018Spring_MTS_v2.txt", header = TRUE)
# View(df_q2)
head(df_q2)
```

```
##   year mon series1 series2 series3 series4
## 1 1947   1      11      22     5.9      14
## 2 1947   2      12      21     5.9      14
## 3 1947   3      12      21     6.0      14
## 4 1947   4      12      21     6.1      14
## 5 1947   5      12      21     6.1      14
## 6 1947   6      12      21     6.1      14
```

```
describe(df_q2)
```

```
## df_q2
##
##  6  Variables      564  Observations
## ---------------------------------------------------------------------------
## year
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0       47        1     1970    15.69     1949     1951
##      .25      .50      .75      .90      .95
##     1958     1970     1982     1989     1991
##
## lowest : 1947 1948 1949 1950 1951, highest: 1989 1990 1991 1992 1993
## ---------------------------------------------------------------------------
## mon
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0       12    0.993      6.5    3.979     1.00     2.00
##      .25      .50      .75      .90      .95
##     3.75     6.50     9.25    11.00    12.00
##
## Value          1     2     3     4     5     6     7     8     9    10
## Frequency     47    47    47    47    47    47    47    47    47    47
## Proportion 0.083 0.083 0.083 0.083 0.083 0.083 0.083 0.083 0.083 0.083
##
## Value         11    12
## Frequency     47    47
## Proportion 0.083 0.083
## ---------------------------------------------------------------------------
## series1
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0      526        1    32.61    16.79    12.14    13.72
##      .25      .50      .75      .90      .95
##    17.57    33.69    44.78    53.26    55.53
##
## lowest : 11 11 11 11 11, highest: 61 61 62 63 63
```

```
## ------------------------------------------------------------------------
## series2
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0      516        1    52.29    22.74    22.04    25.07
##      .25      .50      .75      .90      .95
##    33.23    53.63    68.57    79.10    81.41
##
## lowest : 21 21 21 21 21, highest: 83 83 84 84 84
## ------------------------------------------------------------------------
## series3
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0      526        1    21.31    14.95    6.218    7.308
##      .25      .50      .75      .90      .95
##    9.317   17.389   32.539   43.817   45.689
##
## lowest :  5.0  5.0  5.1  5.2  5.3, highest: 48.8 49.0 49.9 50.2 50.5
## ------------------------------------------------------------------------
## series4
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      564        0      526        1     36.9    17.06    14.08    17.20
##      .25      .50      .75      .90      .95
##    22.04    38.31    50.48    57.17    58.01
##
## lowest : 12 13 13 13 13, highest: 61 61 61 62 62
## ------------------------------------------------------------------------
```

```
# Create R time-series objects with our monthly data starting
# with January 1947
q2_ts_series1 <- ts(df_q2$series1, frequency = 12, start = c(1947,
    1))
q2_ts_series2 <- ts(df_q2$series2, frequency = 12, start = c(1947,
    1))
q2_ts_series3 <- ts(df_q2$series3, frequency = 12, start = c(1947,
    1))
q2_ts_series4 <- ts(df_q2$series4, frequency = 12, start = c(1947,
    1))
```

We see that there are no missing values and that we are working with monthly time series data. No anomalies are detected and there is not potential for top or bottom code.

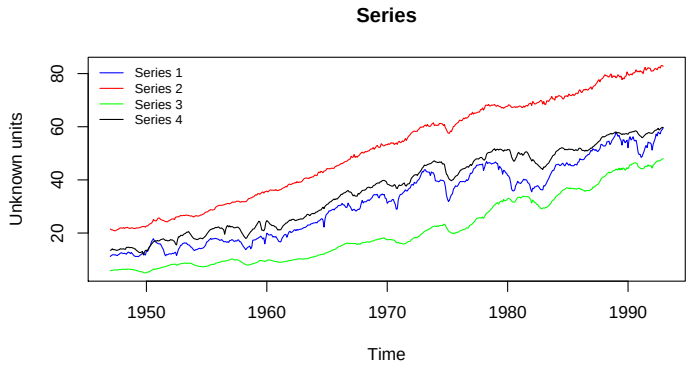
EDA

Generate ts objects, segment train/test sets and plot the data

```
# For modeling purposes we keep data between 1947 and the end
# of 1992 as our training data; we hold-out 1993 as test data
q2_ts_series1_train <- q2_ts_series1[time(q2_ts_series1) >= 1947 &
    time(q2_ts_series1) < 1993]
q2_ts_series2_train <- q2_ts_series2[time(q2_ts_series2) >= 1947 &
    time(q2_ts_series2) < 1993]
q2_ts_series3_train <- q2_ts_series3[time(q2_ts_series3) >= 1947 &
    time(q2_ts_series3) < 1993]
q2_ts_series4_train <- q2_ts_series4[time(q2_ts_series4) >= 1947 &
    time(q2_ts_series4) < 1993]
q2_ts_series1_train <- ts(q2_ts_series1_train, frequency = 12,
    start = c(1947, 1))
q2_ts_series2_train <- ts(q2_ts_series2_train, frequency = 12,
    start = c(1947, 1))
q2_ts_series3_train <- ts(q2_ts_series3_train, frequency = 12,
    start = c(1947, 1))
q2_ts_series4_train <- ts(q2_ts_series4_train, frequency = 12,
    start = c(1947, 1))

# Our test data will span from the start of 1993 to the end
# of 1993
q2_ts_series1_test <- q2_ts_series1[time(q2_ts_series1) >= 1993]
q2_ts_series2_test <- q2_ts_series2[time(q2_ts_series2) >= 1993]
q2_ts_series3_test <- q2_ts_series3[time(q2_ts_series3) >= 1993]
q2_ts_series4_test <- q2_ts_series4[time(q2_ts_series4) >= 1993]
q2_ts_series1_test <- ts(q2_ts_series1_test, frequency = 12,
    start = c(1993, 1))
```

```r
q2_ts_series2_test <- ts(q2_ts_series2_test, frequency = 12,
    start = c(1993, 1))
q2_ts_series3_test <- ts(q2_ts_series3_test, frequency = 12,
    start = c(1993, 1))
q2_ts_series4_test <- ts(q2_ts_series4_test, frequency = 12,
    start = c(1993, 1))

# Plot the training data for all four time series
ts.plot(q2_ts_series1_train, q2_ts_series2_train, q2_ts_series3_train,
    q2_ts_series4_train, gpars = list(main = "Series", ylab = "Unknown units",
        col = c("blue", "red", "green", "black")))
legend("topleft", c("Series 1", "Series 2", "Series 3", "Series 4"),
    lty = 1, col = c("blue", "red", "green", "black"), bty = "n",
    cex = 0.75)

# Show the correlation between ther series
cor(df_q2[3:6])
```

```
##         series1 series2 series3 series4
## series1    1.00    0.98    0.95    0.99
## series2    0.98    1.00    0.95    0.99
## series3    0.95    0.95    1.00    0.95
## series4    0.99    0.99    0.95    1.00
```



We can clearly see that none of the four time series are stationary. There does not appear to be an obvious seasonal trend within any of the four time series. We need to atempt to stationarize our time series via differencing. There also appears to be high correlation between the four time series. This is confirmed with the cor() function, showing that the correlation between any given time series pair is greater than 0.94. All four of the series appear to be a random walk.

Examine the ACF/PACF

```r
# Plot frequency distribution, ACF, PACF for each series
hist(q2_ts_series1_train, main = "Frequency Distribution of Series 1",
    xlab = "Series 1")
acf(q2_ts_series1_train, main = "Autocorrelation function", lag.max = 24)
pacf(q2_ts_series1_train, main = "Partial Autocorrelation function",
    lag.max = 24)
hist(q2_ts_series2_train, main = "Frequency Distribution of Series 2",
    xlab = "Series 2")
acf(q2_ts_series2_train, main = "Autocorrelation function", lag.max = 24)
pacf(q2_ts_series2_train, main = "Partial Autocorrelation function",
    lag.max = 24)
hist(q2_ts_series3_train, main = "Frequency Distribution of Series 3",
    xlab = "Series 3")
acf(q2_ts_series3_train, main = "Autocorrelation function", lag.max = 24)
pacf(q2_ts_series3_train, main = "Partial Autocorrelation function",
    lag.max = 24)
hist(q2_ts_series4_train, main = "Frequency Distribution of Series 4",
    xlab = "Series 4")
acf(q2_ts_series4_train, main = "Autocorrelation function", lag.max = 24)
pacf(q2_ts_series4_train, main = "Partial Autocorrelation function",
    lag.max = 24)
# Unit root tests to confirm non-stationarity
adf.test(q2_ts_series1_train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  q2_ts_series1_train
## Dickey-Fuller = -3, Lag order = 8, p-value = 0.07
## alternative hypothesis: stationary
```

```
adf.test(q2_ts_series2_train)
```

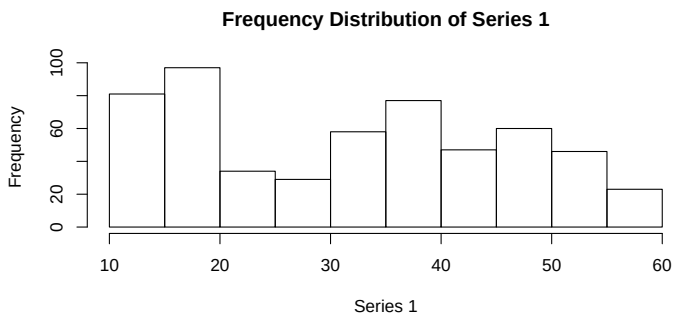
```
##
##  Augmented Dickey-Fuller Test
##
## data:  q2_ts_series2_train
## Dickey-Fuller = -2, Lag order = 8, p-value = 0.5
## alternative hypothesis: stationary
```

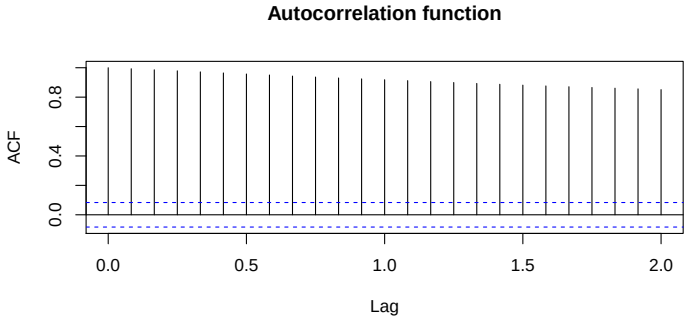
```
adf.test(q2_ts_series3_train)
```

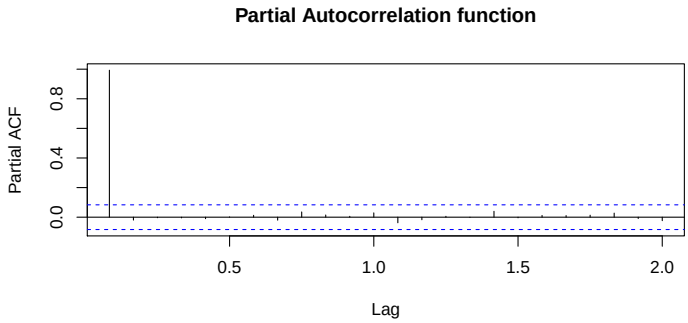
```
##
##  Augmented Dickey-Fuller Test
##
## data:  q2_ts_series3_train
## Dickey-Fuller = -2, Lag order = 8, p-value = 0.5
## alternative hypothesis: stationary
```

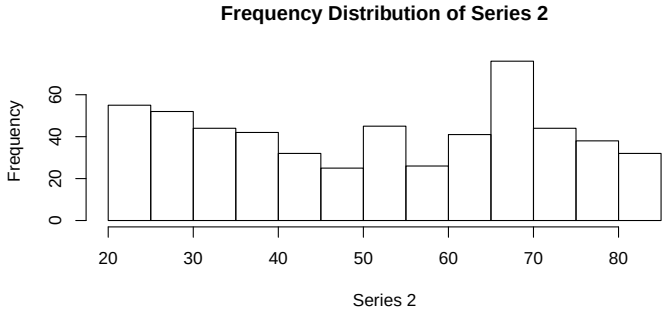
```
adf.test(q2_ts_series4_train)
```

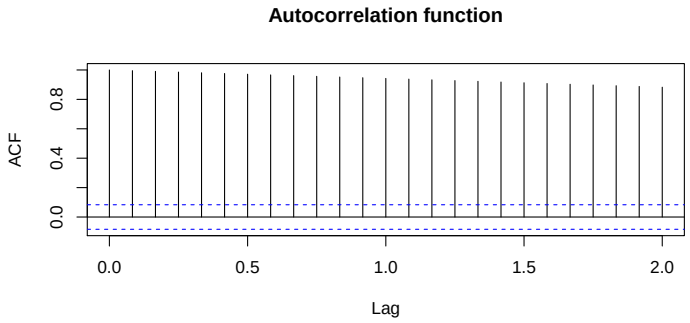
```
##
##  Augmented Dickey-Fuller Test
##
## data:  q2_ts_series4_train
## Dickey-Fuller = -4, Lag order = 8, p-value = 0.03
## alternative hypothesis: stationary
```

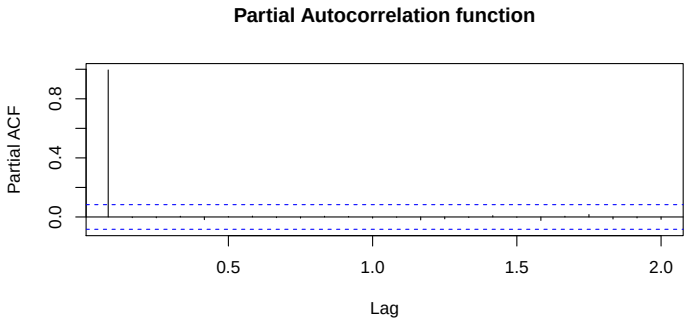


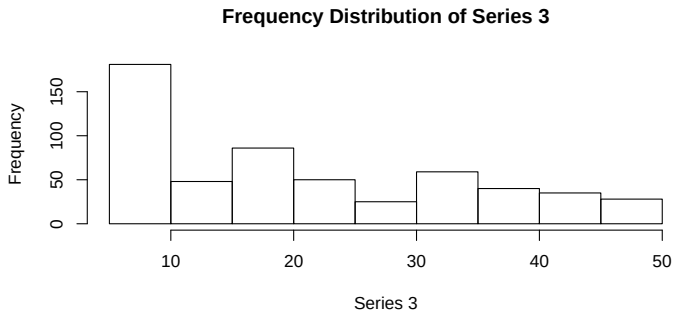


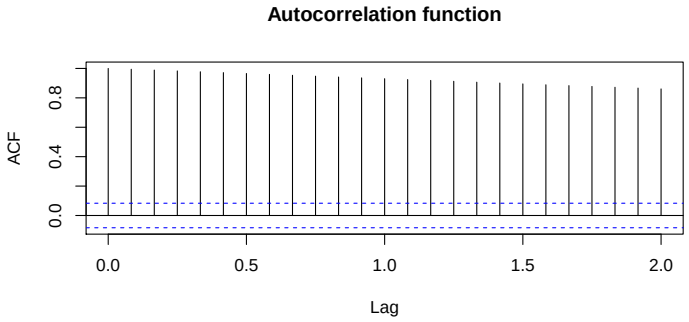


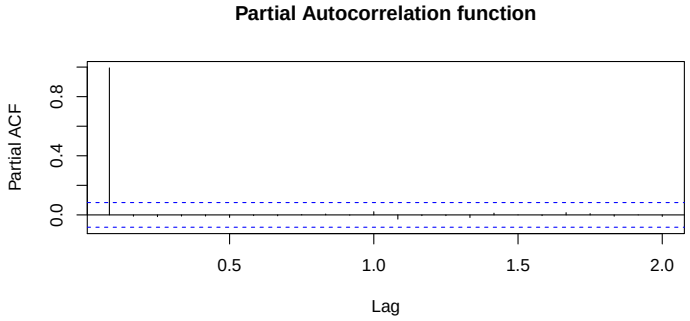


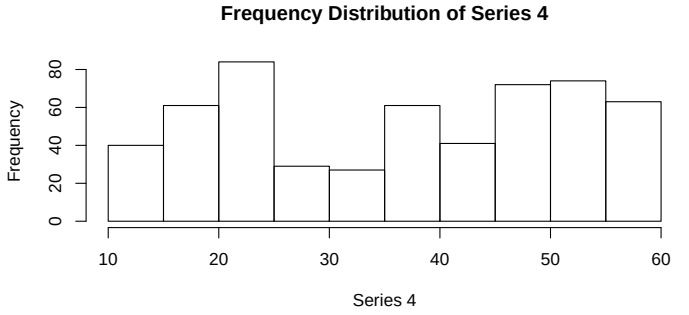


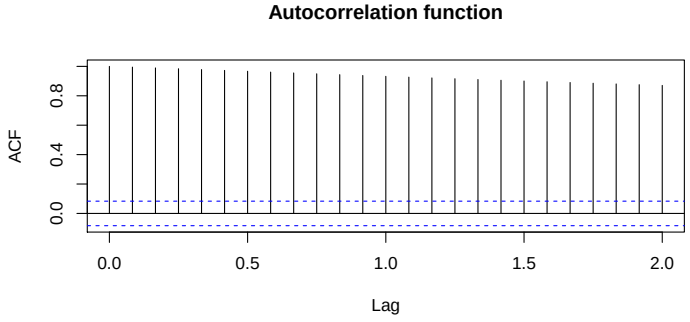


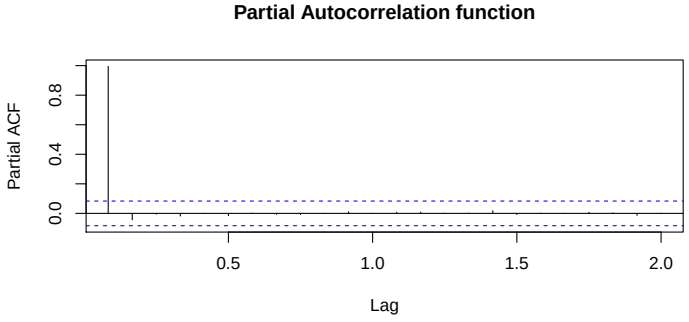












We see that the distributions for the series are fairly uniform except for Series 3, which has a an exponential distribution. All four of the series show autocorrelation that "tails-off" with significant autocorrelation at a high number of lags. Additionally, all four of the series show that the partial autocorrelation cuts-off after lag q = 1, meaning that the autocorrelations at lag 2 and beyondare due to propogation of the autocorrelation at lag 1. This informs our model by telling us that an VAR(1) should be used and that we are unlikely to need a non-seasonal MA components in our model. We also know that the VAR model captures a large amount of model dynamics and that obtaining a VARMA model can be computationally difficult. Thus we will proceed with a VAR model. We perform augmented Dickey-Fuller tests for stationarity on all four series, where the null hypothesis is that the series has a unit root. If we cannot reject the null hypothesis, then it means that there is a unit root and that the series is not stationary. We see that for Series 1-3 we cannot reject the null hypothesis and must treat Series 1-3 as non-stationary. The p-value for Series 1 approaches the level of significance but does not achieve it. We can reject the null hypothesis for Series 4 and treat Series 4 as stationary.

Determination of Cointegration and Differencing the series to impose stationarity

Two non-stationary time series are cointegrated if some linear combination of the two time series is stationary. Here we will test for cointegration within our previously demonstrated non-stationary Series 1-3.

```r
# Phillips-Ouliaris Cointegration Test
q2_ts_mv_1_to_3 <- ts(df_q2[3:5], frequency = 12, start = c(1947,
    1))
po.test(cbind(q2_ts_mv_1_to_3))
```

```
##
##  Phillips-Ouliaris Cointegration Test
##
## data:  cbind(q2_ts_mv_1_to_3)
## Phillips-Ouliaris demeaned = -20, Truncation lag parameter = 5,
## p-value = 0.1
```

```r
# Difference Series 1-3 individually until stationary per
# Dickey-Fuller test
diff_1_q2_ts_series1_train <- diff(q2_ts_series1_train, 1)
diff_1_q2_ts_series2_train <- diff(q2_ts_series2_train, 1)
diff_1_q2_ts_series3_train <- diff(q2_ts_series3_train, 1)
# Dickey-Fuller test for stationarity of final differenced
# series
adf.test(diff_1_q2_ts_series1_train)
```

```
## Warning in adf.test(diff_1_q2_ts_series1_train): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_1_q2_ts_series1_train
## Dickey-Fuller = -8, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(diff_1_q2_ts_series2_train)
```

```
## Warning in adf.test(diff_1_q2_ts_series2_train): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_1_q2_ts_series2_train
## Dickey-Fuller = -8, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(diff_1_q2_ts_series3_train)
```

```
## Warning in adf.test(diff_1_q2_ts_series3_train): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_1_q2_ts_series3_train
## Dickey-Fuller = -6, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

We see that time series 1-3 are not cointegrated per our p-value of 0.1, therefore we will not proceed with a conintegration to attempt to impose stationarity as we cannot reject the null hypothesis that states that the time series are not cointegrated. We should not fit linear regression models of Series 1-3 on each other for modeling. Instead, we proceed with first-differencing within each individual Series 1-3 to impose stationarity. We see per the results of our Dickey-Fuller test on each of the first-differenced Series 1-3 that the first-differenced series is now stationary for Series 1-3.

Order Identification

```r
# Use Varselect to determine lag optimization for VAR model.
# Of note we must take-out the first value from Series 4
# since the other series are all differenced and we need to
# maintain consistent number of rows between the series
q2_ts_series4_train_for_mv <- window(q2_ts_series4_train, start = c(1947,
    2))
series1_as_df <- data.frame(Y = as.matrix(diff_1_q2_ts_series1_train),
    date = time(diff_1_q2_ts_series1_train))
series2_as_df <- data.frame(Y = as.matrix(diff_1_q2_ts_series2_train),
    date = time(diff_1_q2_ts_series2_train))
series3_as_df <- data.frame(Y = as.matrix(diff_1_q2_ts_series3_train),
    date = time(diff_1_q2_ts_series3_train))
series4_as_df <- data.frame(Y = as.matrix(q2_ts_series4_train_for_mv),
    date = time(q2_ts_series4_train_for_mv))
mv_df <- cbind(series1_as_df, series2_as_df, series3_as_df, series4_as_df)
mv_df <- mv_df[c(1, 3, 5, 7, 8)]
VARselect(mv_df, lag.max = 10, type = "both")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      1      2
##
## $criteria
##                1        2        3        4        5        6        7
## AIC(n) -6.2e+01 -6.2e+01 -6.2e+01 -6.1e+01 -6.1e+01 -6.1e+01 -6.1e+01
## HQ(n)  -6.2e+01 -6.2e+01 -6.2e+01 -6.1e+01 -6.1e+01 -6.1e+01 -6.0e+01
## SC(n)  -6.1e+01 -6.1e+01 -6.1e+01 -6.0e+01 -6.0e+01 -6.0e+01 -6.0e+01
## FPE(n)  1.5e-27  1.2e-27  1.3e-27  2.7e-27  2.8e-27  2.9e-27  3.1e-27
##                8        9       10
## AIC(n) -6.1e+01 -6.1e+01 -6.1e+01
## HQ(n)  -6.0e+01 -6.0e+01 -6.0e+01
## SC(n)  -5.9e+01 -5.9e+01 -5.9e+01
## FPE(n)  3.3e-27  3.5e-27  3.7e-27
```

Per VARselect and seeking to minimize our AIC, we should proceed with a VAR(2) model

Model Creation

```r
mv_df <- mv_df[c(1, 2, 3, 4)]
mv_model <- VAR(mv_df[], p = 2, type = "both")
summary(mv_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: Y, Y.1, Y.2, Y.3
## Deterministic variables: both
## Sample size: 549
## Log Likelihood: -964.617
## Roots of the characteristic polynomial:
## 0.952 0.661 0.517 0.399 0.326 0.326 0.308 0.308
## Call:
## VAR(y = mv_df[], p = 2, type = "both")
##
##
## Estimation results for equation Y:
## ==================================
## Y = Y.l1 + Y.1.l1 + Y.2.l1 + Y.3.l1 + Y.l2 + Y.1.l2 + Y.2.l2 + Y.3.l2 + const + trend
##
##        Estimate Std. Error t value Pr(>|t|)
## Y.l1    0.02878    0.05567    0.52  0.60536
## Y.1.l1  0.02194    0.07995    0.27  0.78383
## Y.2.l1 -0.26893    0.15363   -1.75  0.08059 .
## Y.3.l1  0.46992    0.08507    5.52  5.2e-08 ***
## Y.l2   -0.03372    0.05329   -0.63  0.52720
## Y.1.l2  0.07514    0.07942    0.95  0.34452
## Y.2.l2 -0.04078    0.15280   -0.27  0.78967
## Y.3.l2 -0.51868    0.08445   -6.14  1.6e-09 ***
```

```
## const   0.55756    0.16542    3.37  0.00080 ***
## trend   0.00460    0.00119    3.87  0.00012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.73 on 539 degrees of freedom
## Multiple R-Squared: 0.112,   Adjusted R-squared: 0.0967
## F-statistic: 7.52 on 9 and 539 DF,  p-value: 2.08e-10
##
##
## Estimation results for equation Y.1:
## ====================================
## Y.1 = Y.l1 + Y.1.l1 + Y.2.l1 + Y.3.l1 + Y.l2 + Y.1.l2 + Y.2.l2 + Y.3.l2 + const + trend
##
##         Estimate Std. Error t value Pr(>|t|)
## Y.l1    0.040201   0.031270    1.29   0.1991
## Y.1.l1 -0.279480   0.044905   -6.22  9.8e-10 ***
## Y.2.l1 -0.026410   0.086288   -0.31   0.7597
## Y.3.l1  0.092028   0.047783    1.93   0.0546 .
## Y.l2    0.026682   0.029932    0.89   0.3731
## Y.1.l2 -0.108918   0.044608   -2.44   0.0149 *
## Y.2.l2  0.002963   0.085822    0.03   0.9725
## Y.3.l2 -0.106481   0.047435   -2.24   0.0252 *
## const   0.304434   0.092912    3.28   0.0011 **
## trend   0.001317   0.000668    1.97   0.0493 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.41 on 539 degrees of freedom
## Multiple R-Squared: 0.0803,  Adjusted R-squared: 0.0649
## F-statistic: 5.23 on 9 and 539 DF,  p-value: 7.79e-07
##
##
## Estimation results for equation Y.2:
## ====================================
## Y.2 = Y.l1 + Y.1.l1 + Y.2.l1 + Y.3.l1 + Y.l2 + Y.1.l2 + Y.2.l2 + Y.3.l2 + const + trend
##
##         Estimate Std. Error t value Pr(>|t|)
## Y.l1   -0.012490   0.019281   -0.65    0.517
## Y.1.l1 -0.020176   0.027688   -0.73    0.467
## Y.2.l1  0.019868   0.053205    0.37    0.709
## Y.3.l1  0.174366   0.029462    5.92  5.8e-09 ***
## Y.l2   -0.039456   0.018456   -2.14    0.033 *
## Y.1.l2  0.014070   0.027505    0.51    0.609
## Y.2.l2  0.291716   0.052917    5.51  5.5e-08 ***
## Y.3.l2 -0.172592   0.029248   -5.90  6.4e-09 ***
## const  -0.025499   0.057289   -0.45    0.656
## trend   0.000016   0.000412    0.04    0.969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.25 on 539 degrees of freedom
## Multiple R-Squared: 0.189,   Adjusted R-squared: 0.176
## F-statistic:   14 on 9 and 539 DF,  p-value: <2e-16
##
##
## Estimation results for equation Y.3:
## ====================================
## Y.3 = Y.l1 + Y.1.l1 + Y.2.l1 + Y.3.l1 + Y.l2 + Y.1.l2 + Y.2.l2 + Y.3.l2 + const + trend
##
##         Estimate Std. Error t value Pr(>|t|)
## Y.l1   -0.016789   0.031612   -0.53  0.59557
## Y.1.l1  0.022228   0.045397    0.49  0.62459
## Y.2.l1  0.223935   0.087233    2.57  0.01052 *
## Y.3.l1  1.276041   0.048305   26.42  < 2e-16 ***
## Y.l2    0.020174   0.030259    0.67  0.50525
## Y.1.l2  0.080284   0.045096    1.78  0.07559 .
## Y.2.l2  0.138560   0.086761    1.60  0.11085
## Y.3.l2 -0.304707   0.047954   -6.35  4.5e-10 ***
## const   0.368754   0.093929    3.93  9.8e-05 ***
```

```
## trend   0.002503   0.000676    3.70  0.00023 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.41 on 539 degrees of freedom
## Multiple R-Squared: 0.999,   Adjusted R-squared: 0.999
## F-statistic: 7.47e+04 on 9 and 539 DF,  p-value: <2e-16
##
##
##
## Covariance matrix of residuals:
##          Y    Y.1    Y.2    Y.3
## Y   0.5334 0.0807 0.1142 0.1334
## Y.1 0.0807 0.1683 0.0237 0.0421
## Y.2 0.1142 0.0237 0.0640 0.0444
## Y.3 0.1334 0.0421 0.0444 0.1720
##
## Correlation matrix of residuals:
##         Y   Y.1   Y.2   Y.3
## Y   1.000 0.269 0.618 0.440
## Y.1 0.269 1.000 0.228 0.247
## Y.2 0.618 0.228 1.000 0.423
## Y.3 0.440 0.247 0.423 1.000
```

Fit Evaluation

Conclusion