Stats 153 Final Project

Peirang Xu

Dec. 8^{th}, 2017

Information Age?

—An Reflection from Computer and software Wholesales

Introduction

The scientific question that I will analyze in this project is whether there exists a time series model to substantially capture the trend of total amount of monthly wholesale of computers, computer peripheral equipment and software in the United States from January,1997 to August, 2017. As we step in the Information Age, the analysis on the wholesale of computers and software can mirror the continuous impacts of digital technologies. Also, since wholesale trades are considered as an important economic indicator, and manufacturing is a large part of GDP in the United States, the analysis of the amount of monthly sales of computers and software can reflect the role of digital industry in economy nationwide. Thus, based on the study of the trends in the past years' data, we can better investigate the future digital market. The time series analysis is valuable for investors, workforce and distributors, and also important to the technology development as well as innovative thinking in the United States.

Data Description

The original data was obtained from the **United States Census Bureau (https://www.census.gov/wholesale/www/historic_releases/monthly_historic_releases.html)**. The original data from U.S. Census Bureau contains the wholesales for different kinds of durable and nondurable goods as well as the total amount for each month from January 1992 to August 2017. Since the data for computer, computer peripheral equipment and software are only available starting from 1997, my project focuses on the monthly sales of these digital goods from January 1997 through August 2017 in the United States. There is a total of 248 data points to be accessed in the analysis, and the project will focus on fitting a model and predicting the monthly wholesales amount for computers, computer peripheral equipment and software in the United States (The data for wholesales of computer, computer peripheral equipment and software are shown in Appendix I).

Methodology

For the analysis, I first import all 248 observations into R, then analyze the original data and do model fitting. After that, I split the observations into two groups: 212 data points (from January 1997 to August 2014) are used as the training set to build model, and the remaining 36 time points (September 2014 to August 2017) are used to check the performance of model forecasting. Finally, I will use all 248 data points to forecast the future two years.

The steps are: 1. Analyzing the original data. 2. Analyzing trends. 3. Checking for stationarity by checking the plots and using the Augmented Dickey-Fuller Test. 4. Using ACF and PACF to obtain the autocorrelation function estimates. 5. Fitting ARIMA model and forecasting using the training set. 6. Spectral Analysis and curve plots. 7.ARCH/GRACH and forecasting. 8. Residual Analysis and model assessing. 9. Forecasting the future.

Original Data

In the original data published on U.S. Census Bureau website, there are 248 observations for wholesales of computer, computer peripheral equipment and software available. To begin with, I called all packages needed for my time series analysis. Then, I import all the data points into R using the *scan()* function and create the data as time series objects via the *ts()* function. The figure of the observations is shown in Appendix I (**Appendix I**). After creating a time series data set, I plot the original data (**Figure 1**). It is obvious to see that as time goes, the wholesale of computers, computer peripheral equipment and software gradually increases.

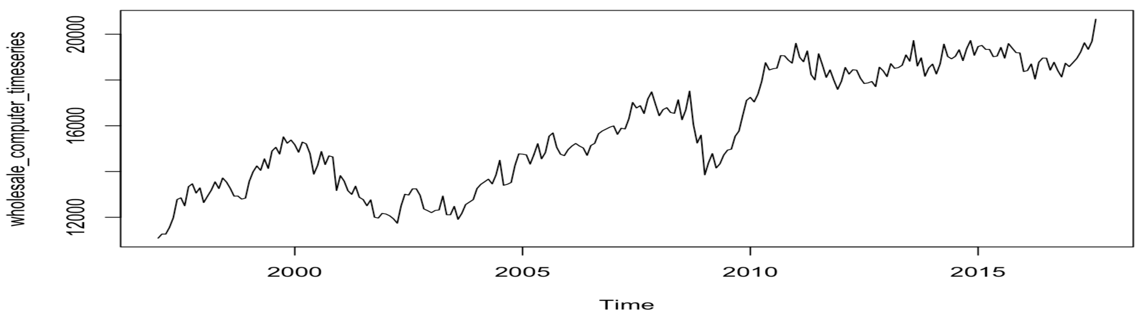


Figure 1

There is a ramp appearing around 2000 followed by a downturn low at around 2003. From 2003 to approximately 2008, the data rebuild and the local mean is steadily increasing over time. However, the sale amounts precipitate around the year of 2008, which could be due to the economic recession at that time.

Then, as I decompose the wholesale data into trend, seasonal and remainders (**Figure 2**), I could get a better understanding of the data set as well as the type of models to use for forecasting future sales.

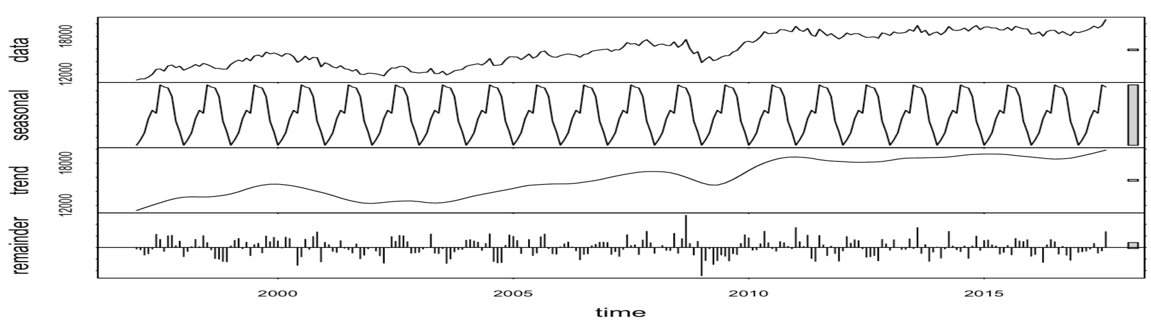


Figure 2

From this plot, it is easy to see that the trend component shows a clear period of increases before and after the major economic events from 2001 to 2008. Looking at the seasonal component, I can see that there is a relatively stable periodic pattern existing and the pattern repeats every 12 months. In the remainder component, the peaks are relatively high around the years 2008 to 2009 (larger variance).

Fitting Model

In order to conduct model fitting, the stationarity of the data needs to be checked, and the ACF and PACF of the data set need to be obtained as well. As observing the plot of the original data (**Figure 1**), I can see that the original time series data are not stationary. To further prove my observation, I used the *adf.test()* function in R to test for stationarity. By setting the alternative hypothesis as stationary, I get a p-value of 0.07274 (> 0.01), meaning that the data set is not stationary. Taking the first difference of the data set, I obtained a plot (**Figure 3**).

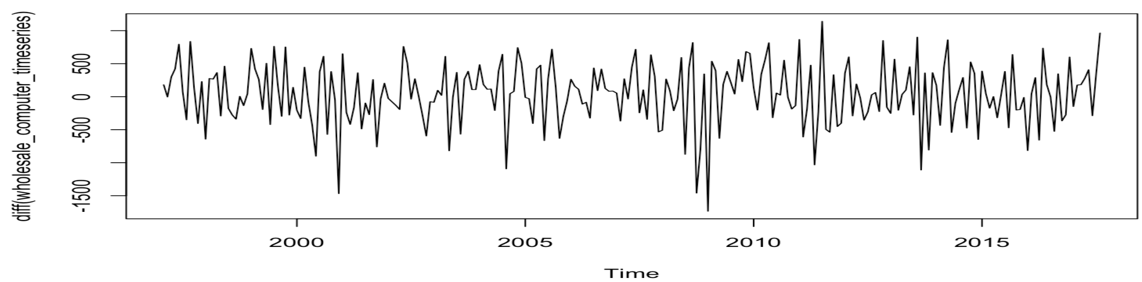
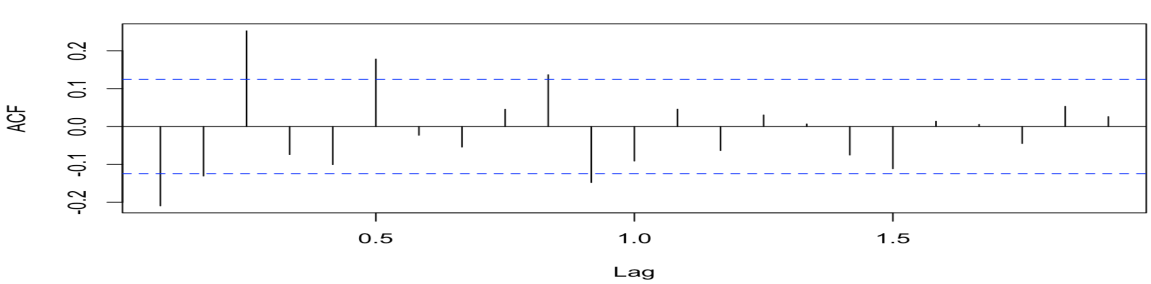


Figure 3

The first difference of the data set looks stationary and has a constant mean. Checking the stationarity using the Augmented Dickey-Fuller Test again, I get a p-value smaller than 0.01, which is small enough to accept the alternative hypothesis and conclude that the first difference of the data set is stationary. Plotting the autocorrelation function and partial autocorrelation function (**Figure 4**), there are several lags that are more significant than the others (exceeding the 95% confidence interval). Moreover, the autocorrelation function follows a trend, which further suggest seasonality of the data set.



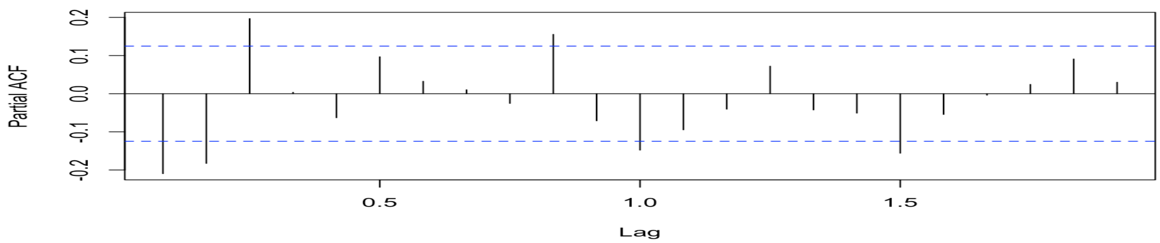


Figure 4

Seasonal ARIMA Model

Using the wholesales data, I will "fit" a seasonal ARIMA model to the data set. This model seeks to minimize the errors of the forecasted sales values using the previous observations and uses past information regarding the seasonality to increase the forecast accuracy. By employing the *eacf()* function and check the AIC values of seasonal ARIMA model with different inputs, and comparing the AIC value with the suggested inputs from *auto.arima()* function, I conclude that an **ARIMA(3,1,2)(1,0,2)[12]** gives the minimum AIC, therefore should be used as the seasonal ARIMA model for forecasting. The following code shows how model fitting is done.

m1_wholesale_computer=arima(wholesale_computer_timeseries,order=c(3,1,2),seasonal=list(order=c(1,0,2), period=12))

As model fitting is done, I next split the data set into training set and testing set. I use the first 212 data points (January 1997 – August 2014) as training set to predict the 36 data points afterwards (September 2014 – August 2017). Then, I compare the predicted values with the actual value of the 36 observations. The forecasting result is shown in the figure below (**Figure 5**). The forecast is shown in blue with the grey area representing the 95% confidence interval. By observing the plot, it is easy to affirm that the forecast matches the historical pattern of the data.

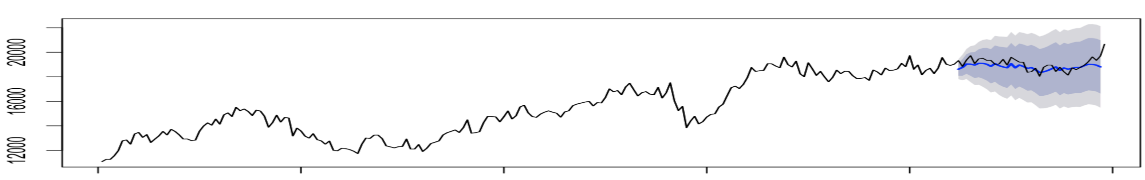


Figure 5

The actual value of the test set (36 observations) are plotted in black. Comparing the forecast and actual values, I can see that they follow a similar pattern, and the actual data are all with in the 95% confidence interval of the forecast, meaning that the seasonal ARIMA model produces reasonable predictions.

ARCH/GARCH

As observing the large volatility in variance in the first difference (**Figure 3**) of the data set, an ARCH/GARCH model also seems reasonable. I perform *McLeod.Li.test()* function and plot the autocorrelation as well as the partial autocorrelation functions of both the absolute value and squared values of the first difference of the data set, and there is pattern leading to fitting an ARCH/GARCH model.

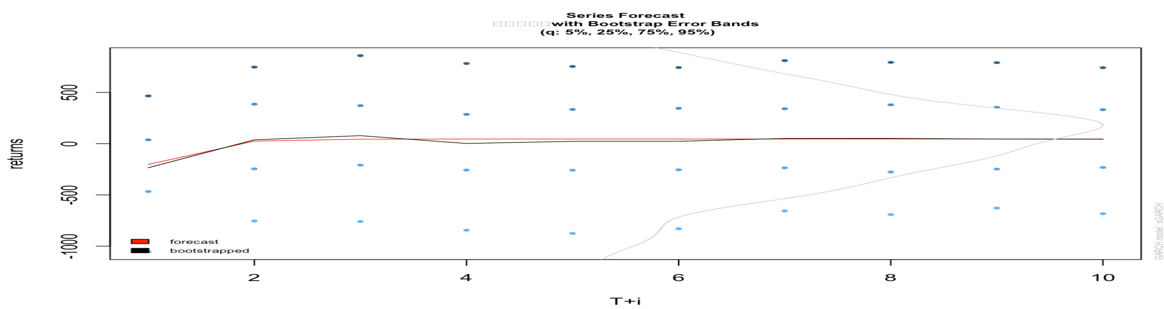


Figure 6

Using the "rugarch" package for garch model fitting, I decided to use GARCH (1,1) model for forecasting the first difference of the data set (**Figure 6**). Looking at the forecast in red and the bootstrapped values in black, since the two lines mostly overlap except for several points, I can say that a GARCH (1,1) model also seems to give reasonable predictions.

Spectral Analysis

Next, I employ Spectral Analysis. Obtaining the spectral density of the time series by a smoothed periodogram (**Figure 7**),

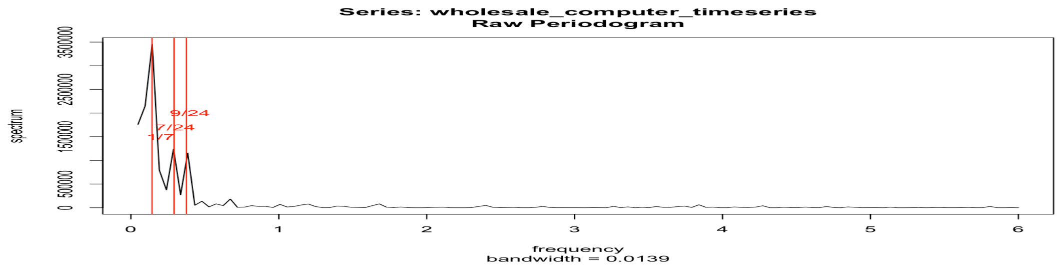


Figure 7

I get frequencies of 1/7, 7/24 and 9/24. Then I construct a general linear model using the first difference of the time series data and the Fourier Series involving three cosines and sines associated with the frequencies in the smoothed periodogram. The general linear model with coefficients is shown below.

$$38.7 - 26.334cos(\frac{2\pi x}{7}) + 10.967sin(\frac{2\pi x}{7}) + 11.275cos(\frac{7\pi x}{12}) - 9.285sin(\frac{7\pi x}{12}) - 2.8\,cos(\frac{3\pi x}{4}) + 68.481sin(\frac{3\pi x}{4})$$

Then, using the *curve()* function for plotting, the seasonality is obvious as shown in the plot (**Figure 8**).

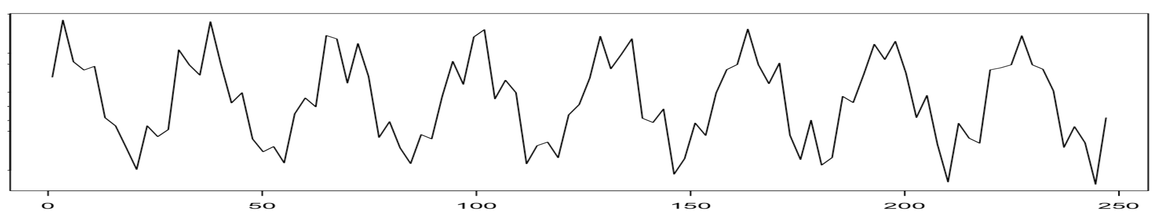


Figure 8

Assessing Model

Then, I would perform more diagnostic checking on the **ARIMA(3,1,2)(1,0,2)[12]** model, **GARCH(1, 1)** model, and the **residuals from spectral analysis**. I first employ the QQ-normal plot in order to investigate the distributions of the residuals from all three models.

In the QQ-normal plot of the residuals for ARIMA model (**Figure 9**), I observed that there are several outliers in the lower tail and one outlier in the upper tail of the plot. But the remaining residuals seem to follow a normal distribution. As I perform the Shapiro-Wilk normality test, I get

W = 0.9864, and a p-value of 0.019. Thus, the outliers in both the lower and upper tail of the QQ-normal plot are not significant enough to reject normality at the usual significance levels. Hence I conclude that the model **ARIMA(3,1,2)(1,0,2)[12]** is a good fit.

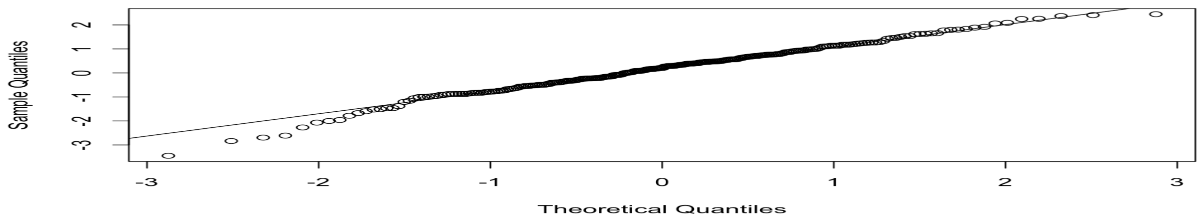


Figure 9

However, looking at the QQ-normal plot of the residuals from **GARCH (1, 1)** model (**Figure 10**), the distribution seems normal except for some outliers at the lower and upper tail. The distribution is not as normal as the residuals of an ARIMA model.

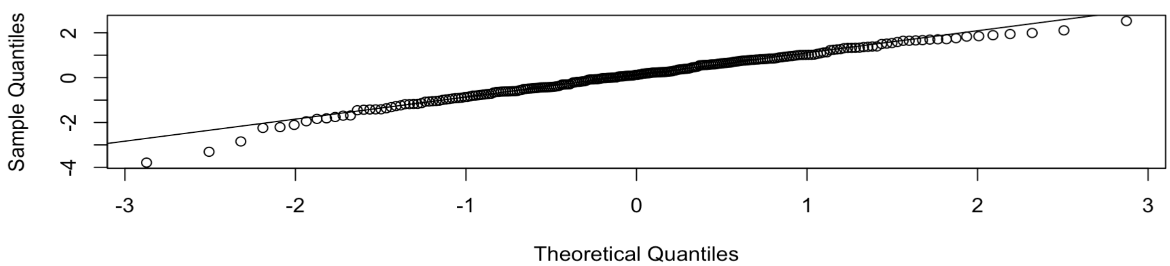


Figure 10

For the residuals from the curve in the spectral analysis part, the QQ-normal plots shows (**Figure 11**), but the normality is not as strong as that of the seasonal ARIMA model. Hence I decide to continue forecasting the future via the seasonal ARIMA model.

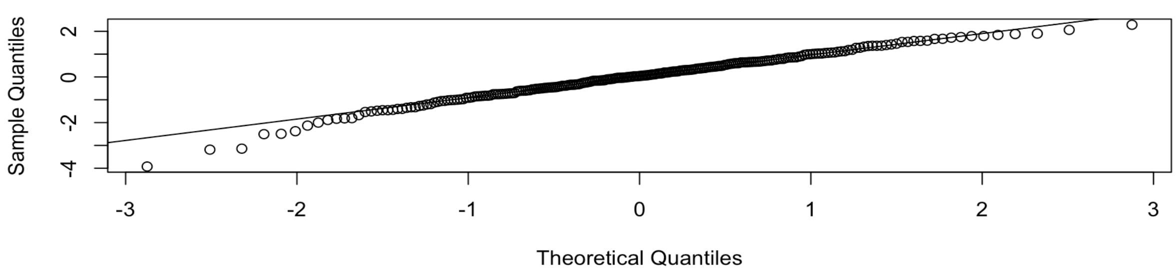


Figure 11

To further assess the seasonal ARIMA model before continuing to the data prediction part, I first take a look at the time series plot of the standardized residuals (**Figure 12**). I can see that other than some strong irregular behavior around the year of 2008, the plot does not suggest any major irregularities with the model.

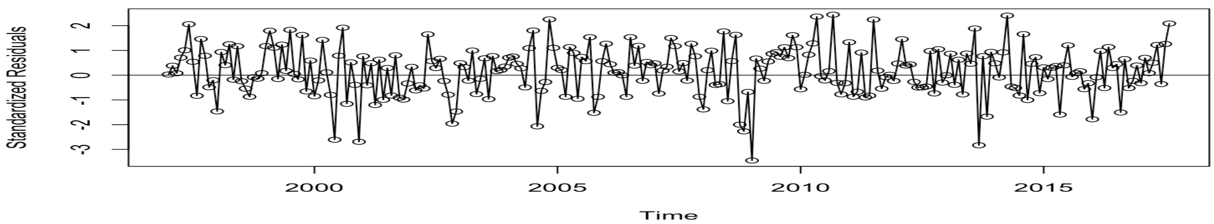


Figure 12

Then, I plot the autocorrelation function of the residuals (**Figure 13**). The only relatively "statistically significant" correlation is at lag 10, yet it does not exceed the 95% confidence interval. Moreover, there is no reason to think of an interpretation for dependence at lag 10. Hence, except for the marginal significance at lag 10, the model seems to be a good fit. Next, I would also check the normality of the residuals.

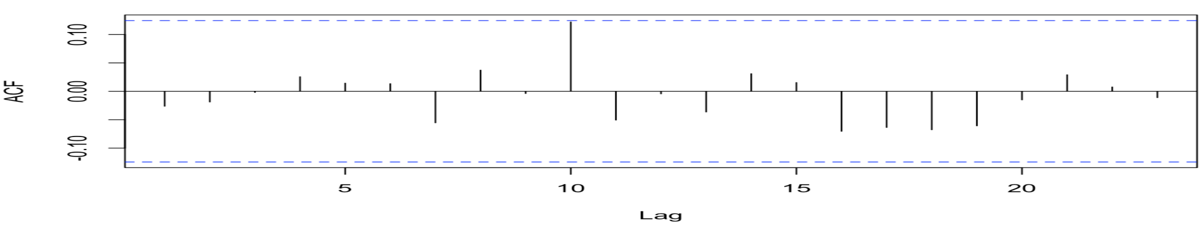


Figure 13

Data Prediction

Next, I use all 248 observations to forecast the future 24 data points (**Figure 14**), which are the wholesales data from September 2017 to August 2019.

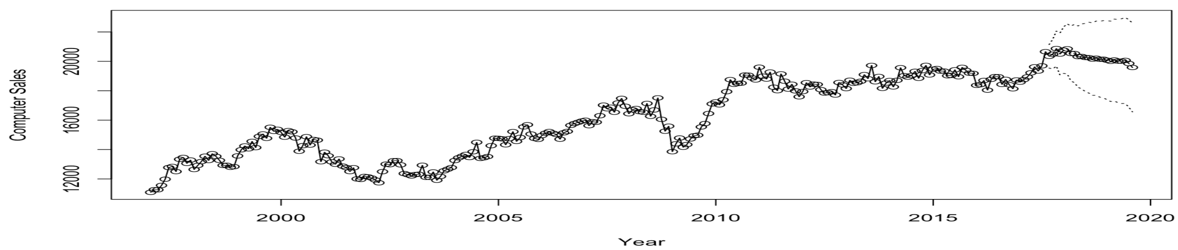


Figure 14

The forecast follows similar trend as the historical pattern, and the dotted lines represent the 95% confidence interval. According to the mean values of forecast data points are shown below (**Figure 15**), I can see that the seasonal ARIMA model forecasts that the wholesales will increase until April 2018, and will reverse to a downward trend afterwards.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2017									20342.95	20485.66	20868.22	20495.46
2018	20759.58	20853.65	20535.43	20532.17	20321.25	20333.16	20262.69	20249.26	20151.69	20213.96	20142.66	20143.02
2019	20075.59	20002.97	20093.43	20015.98	20013.27	20064.48	19854.82	19583.76				

Figure 15

Conclusion

In this project, I investigated 248 observations of the wholesales of computer, computer peripheral equipment and software. By examining the original data, I found a seasonal ARIMA model can substantially capture the trend of the data set. Observing the original data in a time series plot as well as doing forecasting, I can see that the wholesales of computer, computer peripheral equipment and software follows an increasing trend, and the wholesales amounts are under the influence of economic events (for example, the downfall of sales in 2008 due to the economic recession). People are now living in the information age, which could be reflected as the sales of these digital products gradually increases.

Appendix I

Wholesales for Computer, Computer Peripheral Equipment and Software from January 1992 to August 2017

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1997	11087	11267	11263	11561	11982	12776	12851	12501	13337	13464	13058	13286
1998	12643	12918	13184	13548	13256	13717	13540	13264	12924	12929	12794	12836
1999	13566	13984	14244	14050	14555	14133	14895	15059	14762	15514	15236	15377
2000	15171	14841	15287	15209	14785	13884	14264	14875	14304	14684	14639	13171
2001	13820	13581	13161	12999	13360	12872	12775	12503	12762	12001	11966	12169
2002	12143	12064	11932	11738	12498	13008	12971	13241	13245	12960	12365	12288
2003	12203	12299	12319	12929	12110	12109	12476	11909	12172	12556	12663	12771
2004	13255	13439	13551	13667	13458	13849	14490	13397	13442	13521	14264	14771
2005	14760	14729	14323	14743	15218	14553	14825	15544	15686	15055	14755	14693
2006	14956	15114	15226	15113	15028	14704	15138	15230	15648	15773	15854	15939
2007	15993	15624	15894	15859	16303	17020	16776	16878	16535	17169	17480	16946
2008	16437	16703	16791	16579	16542	17135	16262	16696	17515	16055	15246	15589
2009	13854	14390	14785	14154	14339	14721	14936	14976	15539	15767	16450	17106
2010	17242	17040	17380	17940	18756	18438	18493	18517	19068	19058	18871	18736
2011	19602	18992	18796	19272	18239	18005	19146	18650	18112	18443	17991	17589
2012	17946	18548	18257	18448	18430	18076	17845	17872	17935	17710	18560	18400
2013	18147	18714	18509	18543	18644	19096	18819	19720	18611	18970	18162	18525
2014	18696	18262	18707	19567	19026	18926	19032	19322	18847	19372	19725	19077
2015	19463	19511	19337	19338	19019	19044	19425	18954	19592	19390	19196	19184
2016	18368	18412	18700	18042	18776	18959	18955	18430	18775	18409	18133	18733
2017	18584	18760	18942	19220	19627	19339	19693	20657				

Appendix II

```
## Importing Data
wholesale_computer = scan("~/desktop/computerwholesale.csv")
wholesale_computer_timeseries = ts(wholesale_computer, frequency = 12, start = c(1997, 1))

## Calling the packages needed
library(ggplot2)
library(tseries)
library(TSA)
library(forecast)
library(quantmod)
library(rugarch)

## Initial Plot
plot.ts(wholesale_computer_timeseries)

## Decompose data into trend, seasonal and remainder components
fit = stl(wholesale_computer_timeseries, s.window = "periodic")
plot(fit)

## Spectral Analysis
raw_spec = spec.pgram(wholesale_computer_timeseries, taper = 0)
plot(raw_spec)
plot(raw_spec, log = "no")

abline(v = 1/7, col = "red")
abline(v = 7/24, col = "red")
abline(v = 9/24, col = "red")
text(x = c(0.2, 0.3, 0.4), labels = c("1/7", "7/24", "9/24"), y = c(1500000, 1700000, 2000000), col =
"red")

sp = lm(formula = diff(wholesale_computer_timeseries) ~ cos(2*pi*1/7*seq(from = 1, to = 247)) +
sin(2*pi*1/7*seq(from = 1, to = 247)) + cos(2*pi*7/24*seq(from = 1, to = 247)) +
sin(2*pi*7/24*seq(from = 1, to = 247)) +cos(2*pi*9/24*seq(from = 1, to = 247)) +
sin(2*pi*9/24*seq(from = 1, to = 247)), diff(wholesale_computer_timeseries))

curve(38.7 - 26.334*cos(2*pi*1/7*x) +10.967*sin(2*pi*1/7*x) + 11.275*cos(2*pi*7/24*x) -
9.285*sin(2*pi*7/24*x) -2.8 *cos(2*pi*9/24*x) +68.481*sin(2*pi*9/24*x), from = 1, to = 247)

qqnorm(window(rstandard(sp)));qqline(window(rstandard(sp)))

shapiro.test(rstandard(sp))

## ARIMA Model
#adf test
```

```
adf.test(wholesale_computer_timeseries, alternative="stationary", k=0)
adf.test(diff(wholesale_computer_timeseries), alternative = "stationary", k=0)

acf(diff(wholesale_computer_timeseries))
pacf(diff(wholesale_computer_timeseries))
eacf(diff(wholesale_computer_timeseries))

mean(diff(wholesale_computer_timeseries))

#plot to test for stationary
plot.ts(wholesale_computer_timeseries)
plot.ts(diff(wholesale_computer_timeseries))

#ARIMA model
arima(diff(wholesale_computer_timeseries), order = c(2, 1, 2), include.mean = FALSE)
arima(diff(wholesale_computer_timeseries), order = c(0, 1, 3), include.mean = FALSE)
arima(diff(wholesale_computer_timeseries), order = c(1, 1, 3), include.mean = FALSE)
arima(diff(wholesale_computer_timeseries), order = c(2, 1, 3), include.mean = FALSE)
arima(diff(wholesale_computer_timeseries), order = c(3, 1, 3), include.mean = FALSE)
#ARIMA(2, 1, 3) gives the minimum AIC value

tsdiag(arima(diff(wholesale_computer_timeseries), order = c(2, 1, 3), include.mean = FALSE))
qqnorm(residuals(arima(diff(wholesale_computer_timeseries), order = c(2, 1, 3), include.mean =
FALSE)));qqline(residuals(arima(diff(wholesale_computer_timeseries), order = c(2, 1, 3),
include.mean = FALSE)))

## ARIMA Forecast
fit_arima = stats::arima(wholesale_computer_timeseries, order = c(2, 1, 3))
plot(forecast(fit_arima, h = 30), pch = 19, ylab = "computer sales")

hold = window(ts(wholesale_computer_timeseries), start = 224)
fit_no_holdout = stats::arima(ts(wholesale_computer_timeseries[-c(224:248)]), order = c(2, 1, 2))
plot(forecast(fit_no_holdout, h = 24))
lines(ts(wholesale_computer_timeseries))

## Seasonal ARIMA
plot(window(wholesale_computer_timeseries), start = c(1997,1), ylab = "computer sales")
Month = c("J", "F", "M", "A", "M", "J", "J", "A", "S", "O", "N", "D")
points(window(wholesale_computer_timeseries, start = c(1997, 1)), pch = Month)

auto.arima(wholesale_computer_timeseries)
#An ARIMA(3,1,2)(1,0,2)[12] is given by auto.arima

plot(diff(diff(wholesale_computer_timeseries),lag=12),xlab='Time',
  ylab='First and Seasonal Difference of computer sales')

acf(as.vector(diff(diff(wholesale_computer_timeseries),lag=12)))
```

```
#model
m1_wholesale_computer=arima(wholesale_computer_timeseries,order=c(3,1,2),seasonal=list(order
=c(1,0,2),
  period=12))
m1_wholesale_computer

#Residuals
plot(window(rstandard(m1_wholesale_computer)), start=c(1997,1), ylab='Standardized Residuals',
type='o')
abline(h=0)

acf(as.vector(window(rstandard(m1_wholesale_computer),start=c(1997,1))))

qqnorm(window(rstandard(m1_wholesale_computer),start=c(1997,1)))
qqline(window(rstandard(m1_wholesale_computer),start=c(1997,1)))

shapiro.test(rstandard(m1_wholesale_computer))

## Seasonal ARIMA Forecasting
hold = window(ts(wholesale_computer_timeseries), start = 212)
fit_no_holdout = stats::arima(ts(wholesale_computer_timeseries[-c(212:248)]), order = c(3, 1, 2),
seasonal = list(order = c(1, 0, 2), period = 12))
plot(forecast(fit_no_holdout, h = 36))
lines(ts(wholesale_computer_timeseries))

plot(m1_wholesale_computer,n1=c(1997,1),n.ahead=24,xlab='Year',type = 'o',
  ylab='Computer Sales')

fit_seasonal_ARIMA =
forecast(stats::arima(wholesale_computer_timeseries,order=c(3,1,2),seasonal=list(order=c(1,0,2),pe
riod=12)))
fit_seasonal_ARIMA$mean
fit_seasonal_ARIMA$lower
fit_seasonal_ARIMA$upper

plot(fit_seasonal_ARIMA$residuals)
abline(h = 0)

## ARCH/GARCH
diff_wholesales_computer_timeseries = diff(wholesale_computer_timeseries)
plot(diff_wholesales_computer_timeseries);abline(h = 0)

acf(diff_wholesales_computer_timeseries)
pacf(diff_wholesales_computer_timeseries)

acf(abs(diff_wholesales_computer_timeseries))
pacf(abs(diff_wholesales_computer_timeseries))
```

```
acf(diff_wholesales_computer_timeseries^2)
pacf(diff_wholesales_computer_timeseries^2)

McLeod.Li.test(y = diff_wholesales_computer_timeseries)

garch_wholesale<-ugarchspec(variance.model = list(model="sGARCH", garchOrder=c(1,1),
mean.model = list(armaOrder=c(0,0))), distribution.model = "std")
wholesale_garch<-ugarchfit(spec=garch_wholesale, data=diff_wholesales_computer_timeseries)

wholesale_predict<-ugarchboot(wholesale_garch,n.ahead=10, method=c("Partial","Full")[1])
plot(wholesale_predict,which=2)

ga = garch(diff_wholesales_computer_timeseries, order = c(1,1))
qqnorm(ga$residuals);qqline(ga$residuals)
```