Statistics 153: Introduction to Time Series

Professor: Dr. David Brillinger

Student: Audrey Webb

Examining Health and Personal Care Store Sales from a Time Series Perspective

Question

The scientific question motivating my work is: do the sales of health and personal care stores in the United States follow a seasonal pattern?

Introduction

Health and personal care stores make prescriptions drugs and other health-related products readily available to the public. In-store pharmacies, which account for about 67% of total sales, are in demand with over 3.4 billion retail prescriptions being filled nationally since 2005 and have resulted in more patient-pharmacist interaction. Improving upon this relationship, as well as understanding drug sales and spending, will lead to an assessment of how best to lower healthcare costs as well as improve quality of life. By examining this data, my hopes are it will indicate future growth, showing that there is high demand and economic value in health and personal care stores, and aid stores in identifying and measuring seasonal variations within their market to help them plan for the future.

Data Description

The unadjusted data, "Health and Personal Care Stores", is provided by the Monthly and Annual Retail Trade Report on the website, *United States Census Bureau*. The available data is the monthly sales and ranges from January 1992 to October 2017 (approximately 297 observations with two missing values) and sales values are in millions of dollars.

Methods

The methods I will be using are Exponential Smoothing Space Model, TBATS Model, Seasonal ARIMA Model, ARCH/GARCH Model, ARIMA + GARCH Model, Spectral Analysis (Periodogram, Spectral Density), and Time Series Forecast.

Data Cleaning

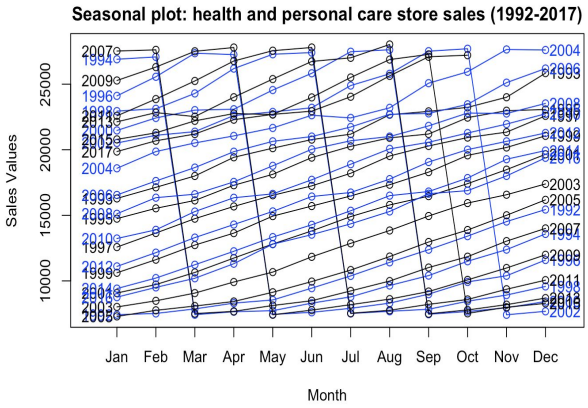
The data is raw and unadjusted. Seasonal adjustment is very important for analyzing data, but the majority of government statistics do not seasonally adjust their own data. Thus, my data being from a government website resulted in me spending a lot of time cleaning and adjusting it in order to account for its seasonality and prepare for future analysis. The data downloads as an unorganized data frame. After using function **unlist()** to turn the data frame into a vector, I am able to remove the missing values and turn the data into a time series object using **ts()**. After understanding my time series data through functions such as **start(), end(), tsdisplay(),** and numerous plots, I see that the timestamps now include integers after a decimal place (i.e. instead of 2017 it's now 2017.833). This is due to removing missing values, making my data exact. The final part of my cleaning process is splitting the data set into training and test sets to carry out future model fits, prediction, and forecasting future sales. The training set is from 1992 to 2014 and the test set is from 2014 to 2017. I will fit numerous time series models on the training set, and use these fitted models to make a prediction on the test set. From this I will show the corresponding forecast and residual plots.

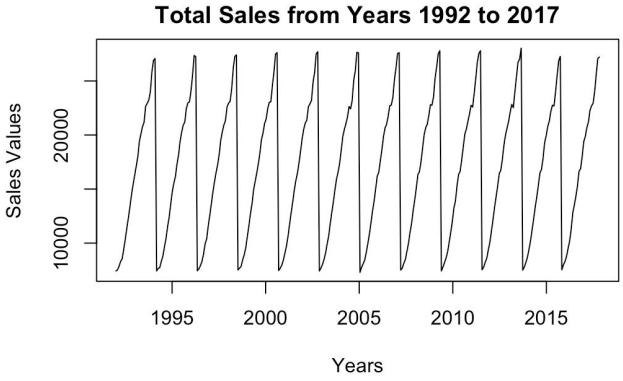
Analysis

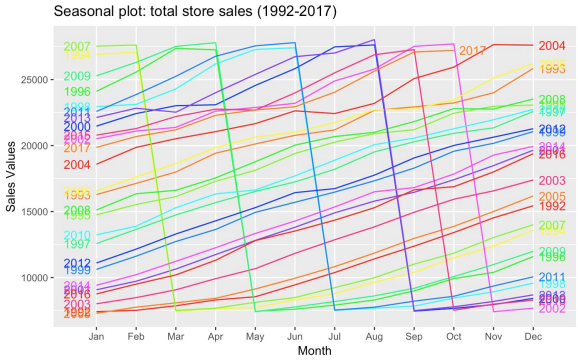
Throughout analysis I will be making the assumption that residuals are normally distributed.

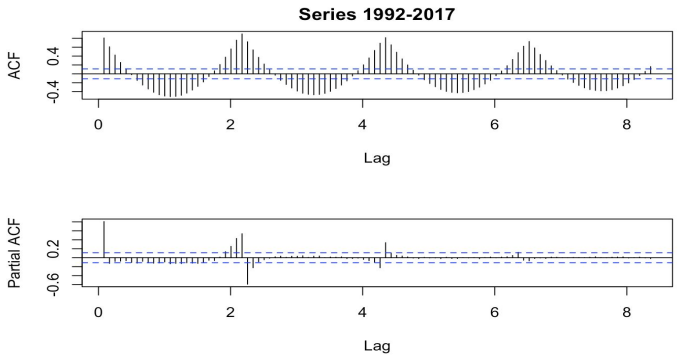
<u>Exploratory Data Analysis (EDA)</u>

Our exploratory data analysis (EDA) begins with us plotting our time series data in order to









identify seasonality, trend, and correlation (ACF and PACF plots with lag.max=100).

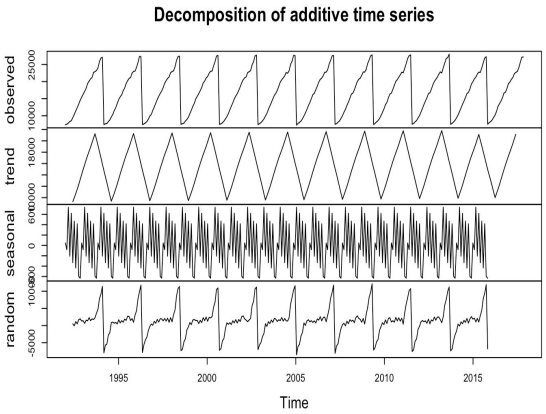
The seasonality plot shown above is the data plotted against the individual "seasons" in which the data is observed.[1] The plots here are the autocorrelation function (ACF), the partial autocorrelation function (PACF), and seasonal plots. ACF plots display correlation between a series and its lags, and in addition to suggesting the order of differencing, it assists in determining the order of an MA(q) model. PACF displays correlation between a variable and the lags not explained by previous lags, and is useful when determining the order of an AR(p) model. From the plots, I am able to make the conclusion that the data is very seasonal. Note the peaks occur at lags of 12 months, showing the months are correlated (in other words January of 2013 correlates to January of 2014 and so on and so forth). Additionally, the large spike

[1] In our case, "season" is a month. This is similar to the general time plot above, except that the data from each season is overlapped. This enables the underlying seasonal pattern to be seen more clearly, and also allows any substantial departures from the seasonal pattern to be easily identified.

at lag 1 followed by a decreasing wave that alternates between positive and negative correlations show that our data is seasonal.

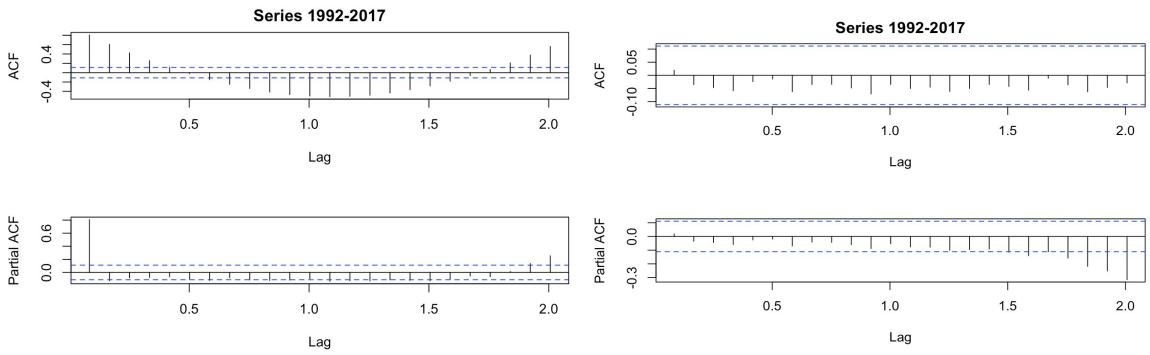
Decomposition:

Deconstructing a time series is important because it can help you understand its behavior, smooth trend and seasonality, and prepare a foundation for building a forecasting model. For my data, I will decompose it using **decompose().** My data has additive seasonal components which is seen in the plot as seasonal variation appears relatively constant over time. Thus why **decompose()** is the correct function for for my data as it deals with additive seasonal components, and decomposes data into seasonal, trend, and irregular components using moving averages. You can see a visual comparison of seasonal and subseries plots, ACF, and PACF plots before and after decomposition below and in the Appendix. Looking at our decomposed data, we see that the seasonal pattern is strong and there is a stable and constant trend. The remainder, the data obtained after seasonal adjustment, and removing any trend, suggests that some differencing may be needed.



Stationarity:

I first take a seasonality difference to account for constant mean as well as a log transformation of my data to account for variance. My data is monthly, so my period is 12. The plot of the log transformation applied to the data (see Appendix) suggests that log is not needed for our dataset as the plots do not show any change before and after the log transformation. ACF plots of our data before and after differencing is seen below. The plots suggest the data is



Before After

stationary after taking one difference. Next, I use the Adjusted Dickey Fuller (ADF) Test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test to double-check for stationarity. The null hypothesis for the ADF test is that data is non-stationary. I am using the 5% significance level, thus a p-value above 0.05 means there is no evidence that the data is stationary. On the other hand, the null hypothesis for the KPSS test is that the data is level or trend stationary. Thus, a p-value above 0.05 means that there is evidence that the data is level or trend stationary. Generally, be careful when choosing the lag length for both tests, because if lag length is too small then the remaining correlation will potentially be biased. A large lag length, on the contrary, will hurt the power of the test. I notice that the length of my data is not significant enough for this to be an issue, thus I proceed with the default inputs in R. The results are:

```
        Augmented Dickey-Fuller Test

data:  u.decomp$x
Dickey-Fuller = -8.3743, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary
```

```
       KPSS Test for Level Stationarity

data:  u.decomp$x
KPSS Level = 0.093326, Truncation lag parameter = 4, p-value =
0.1
```

```
        KPSS Test for Trend Stationarity

data:  u.decomp$x
KPSS Trend = 0.019838, Truncation lag parameter = 4, p-value =
0.1
```

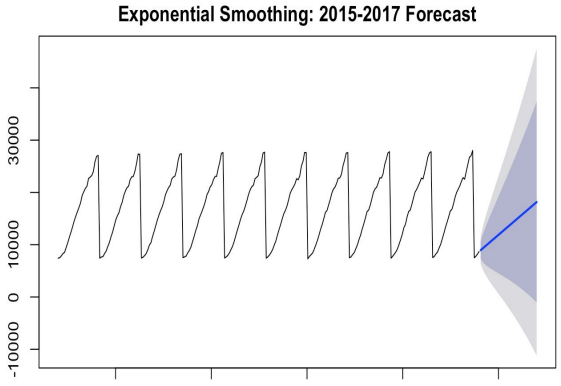
ADF gives a p-value of 0.01 and KPSS level and trend tests give a p-value of 0.1. Both results suggests that my data is stationary. Combining test results with my ACF interpretation, I conclude that the data is stationary and only one difference is needed. This result means that my stationary time series mean, variance, and autocovariance are time invariant; all important factors for using models like ARIMA,

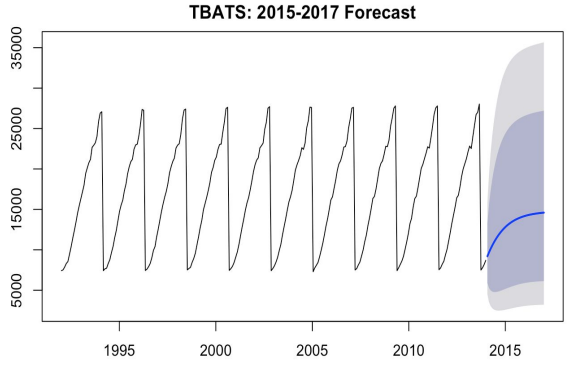
which uses previous lags of stable series to model its behavior with consistent properties involving less uncertainty.

Model Fitting

Now, that the data is decomposed and we have confirmed its constant variance and made it stationary, we can move on to model fitting. I will be splitting the decomposed data into a training set and test set. The training set is made up of the first twenty-two years (1992 to 2014), and the test set is made up of the last three years (2015 to 2017). A number of time series models will be fit on our training set, and this will be used to make predictions on the test set. From this I can carry out a forecast as well as examine error through plots of residuals. For context, the forecast plots will show the forecast in blue within the grey area representing the 95% confidence interval.

We begin our analysis by fitting an exponential smoothing state space model, and carrying out a corresponding forecast. This model produces a smoothed time series, and assigns exponentially decreasing weights as the observation gets older (i.e. recent observations are given relatively more weight in forecasting than the older observations). From our exponential smoothing forecast, we see that this model does not follow the historical pattern well.





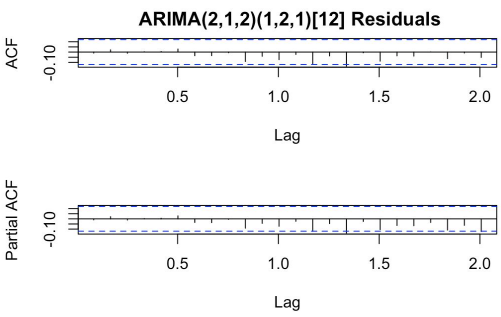
The next model is TBATS, a model for time series that exhibit multiple complex seasonalities. I use the **tbats()** function in the {forecast} package, as it is a generalization of the BATS model with new salient features (trigonometric regressors to model multiple seasonalities, Box-Cox transformations, ARMA errors,

trend, and seasonality). From our TBATS forecast, we see that this model is a worse fit than the exponential smoothing model as this forecast deviates quite a bit from the historical data pattern.

The next model I fit is the seasonal ARIMA model. ARIMA model focuses on analyzing time series linearly and it does not reflect recent changes as new information is available. Therefore, in order to update the model, users need to incorporate new data and estimate parameters again. The variance in ARIMA model is unconditional variance and remains constant. I was not able to use the **sarima()** function from the {sarima} package to analyze my seasonal data as it would not download for the version of R that I have (see Appendix for error message). Instead, I used **auto.arima()** from the {forecast} package to fit a seasonal ARIMA model. This is just to give a preliminary idea on the fit, since we cannot fully rely on it because the number of seasonal differences is sometimes poorly chosen. I then proceed to fitting different seasonal ARIMA models using the **Arima()** function from the {forecast} package based off the information I have from the ACF and PACF plots of my stationary data. I check each seasonal ARIMA fit by looking at the ACF and PACF plots of the respective residuals. A good fit implies that residuals are white noise, so I have to observe one major spike in the ACF plot at lag 1 and the rest of the data should lie within the blue dashed lines (5% interval). I fit eight different seasonal ARIMA models and include the top five below (includes auto.arima). The table of fitted models and barplots representing their AIC, AICc, and BIC can be see in the Appendix.

Model Selection

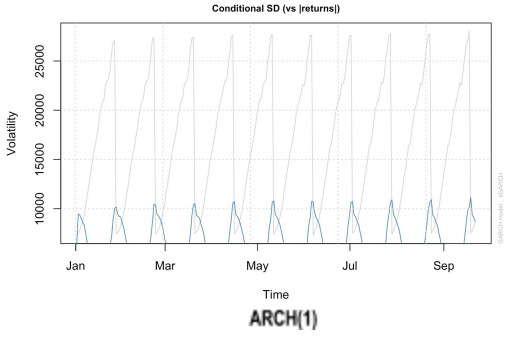
I select the optimal model based on which has the smallest AIC, AICc, and BIC values. I choose AIC because asymptotically minimizing the AIC is equivalent to minimizing the leave-one-out cross-validation MSE for cross-sectional data, and equivalent to minimizing the out-of-sample one-step forecast MSE for time series models. This property is what makes it such an attractive criterion for use in selecting models for forecasting. From my analysis, I see that the model “ARIMA4”, which is



ARIMA(2,1,2)(1,2,1)[12], is the optimal model for my data. From the histogram and qqplot of my optimal model's residuals, I see that my residual normality assumption is incorrect (see Appendix). The ACF and PACF of my model's residuals show that our residuals are now like white noise, which supports the claim that "ARIMA4" is the best model to fit my data compared to that produced by **auto.arima()** and those I tried. We will use this optimal model for future forecasting.

Volatility Modeling

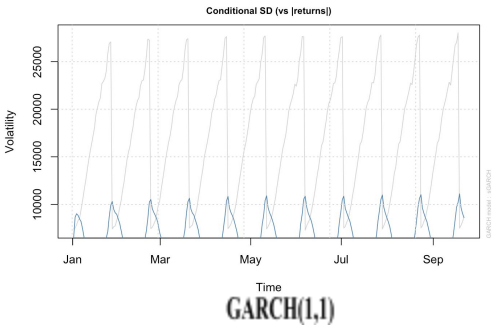
The residuals of my chosen model does not have a normal distribution, meaning there may be variance issues or outliers. In addition, it is found to be serially uncorrelated but admit a higher-order dependence structure, namely volatility clustering, and a heavy-tailed distribution. In order to model variance, I will be fitting an ARCH(1) model (equivalent to ARMA(0,0)-GARCH(0,1)) and a GARCH(1,1) model (equivalent to ARMA(0,0)-GARCH(1,1)) on my data as these are the generally preferred orders for ARCH/GARCH models. ARCH/GARCH is a method to measure volatility of the series, or more specifically, to model the noise term of ARIMA model. ARCH/GARCH incorporates new information and analyzes the series based on conditional variances where users can forecast future values with up-to-date information. I am making the assumption that because my best fitted model is seasonal, the ARMA-GARCH model will not fit my data well. So, I will also be modelling my data using an ARMA-GARCH model with an order that's equivalent to doing an ARIMA(2,1,2)-GARCH(1,1) model, if that was possible. I will be using the 5% significance level, and all plots will be shown below as well as in the Appendix.



I first fit an ARCH(1) model. Weighted Ljung-Box Test on Standardized Residuals and Standardized Squared Residuals have a p-value of zero, so I reject the null hypothesis and

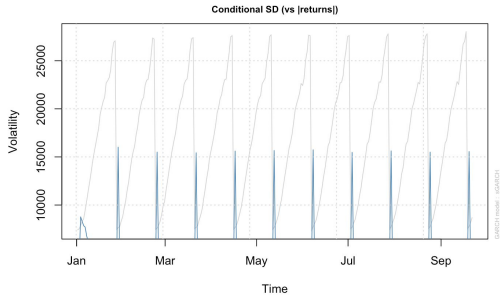
conclude there is serial correlation. The p-value for the Weighted ARCH LM Tests are 0.6034, 0.3098, and 0.2747 which are all larger than my significance level so I conclude that there is no ARCH effect. The Sign Bias test has a p-value of 1.921e-15 which leads me to conclude that there is a significant negative and positive reaction shock meaning there likely exists an apARCH type model.

Next, I fit a GARCH(1,1) model. Weighted Ljung-Box Test on Standardized Residuals and Standardized Squared Residuals have a p-value of zero, so I reject the null hypothesis and conclude there is a serial correlation. The p-value for the Weighted ARCH LM Tests are 0.22109, 0.17241, and 0.06091 which are all larger than my significance level so I conclude that there is no ARCH effect. The Sign Bias Test has a p-value of 1.911e-44 which leads me to conclude that there is a significant negative and positive reaction shock meaning there likely exists an apARCH type model.



GARCH(1,1)

Finally, I fit an ARMA(2,2)-GARCH(1,1) model which is equivalent to an ARIMA(2,1,2) + GARCH(1,1) model. Weighted Ljung-Box Test on Standardized Residuals has a p-values of 2.667e-11 and zero, so I reject the null hypothesis and conclude there is a serial correlation. The p-value for the Weighted ARCH LM Tests are 0.6918, 0.5807, 0.4764 which are all larger than my significance level so I conclude that there is no ARCH effect. The Sign Bias Test has a p-value of 1.189e-01 which leads me to conclude that there is a significant negative and positive reaction shock meaning there likely exists an apARCH type model. From the values, we're able to see that ARIMA+GARCH best models the volatility compared to the other two models. Also, even though ARCH/GARCH is not made specifically for forecasting, the ARIMA+GARCH model has the most accurate forecast compared to ARCH(1) and GARCH(1,1) (see Appendix).



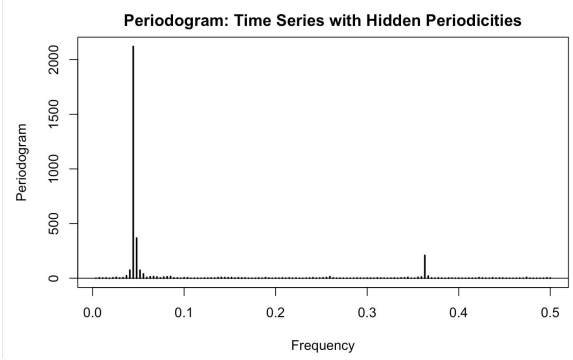
ARMA(2,2)+GARCH(1,1)

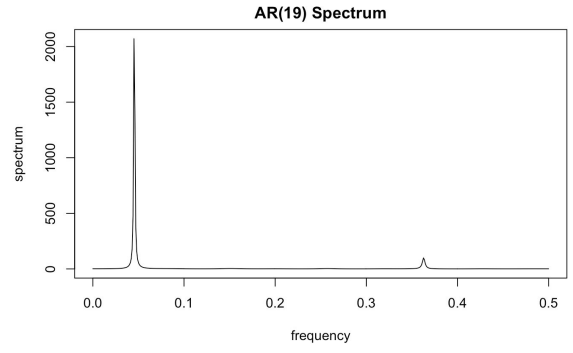
Spectral Analysis

A periodogram calculates the significance of different frequencies in time-series data to identify any intrinsic periodic signals. A periodogram is similar to the Fourier Transform, but is optimized for unevenly time-sampled data, and for different shapes in periodic signals. From my reading and understanding of the function of periodograms, I am using periodogram more as a helpful tool for identifying the dominant cyclical behavior in a series, if any. Additionally, I want to know where there are cosines in my time series, as well as if it contains additional "noise".

Knowing my data has four dominant frequencies, I calculate the linear combination of the two cosine curves, and the hidden periodicities. This calculation is based off this model (often described as a signal plus noise model, where the signal could be deterministic with unknown parameters or stochastic):

$$Y_t = A_1 \cos(2\pi f_1 t) + B_1 \sin(2\pi f_1 t) + A_2 \cos(2\pi f_2 t) + B_2 \sin(2\pi f_2 t) + W_t \ .$$



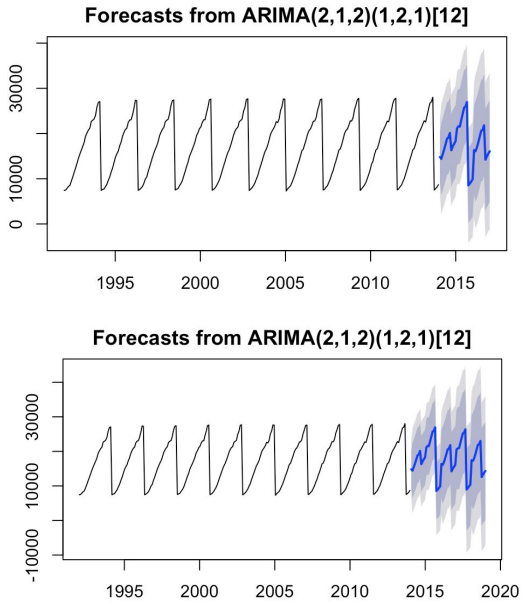


After plotting for visualization purposes, I went on to analyze the respective periodograms. From my periodogram of the time series with hidden periodicities, I see that the series contains two cosine-sine pairs at frequencies of about 0.05 and 0.37, and that the lower-frequency component is much stronger.

This is a rough sample estimate of the population spectral density. The estimate is "rough", in part, because we only use the discrete fundamental harmonic frequencies for the periodogram whereas the spectral density is defined over a continuum of frequencies. The approach we will use is smoothing the periodogram is a parametric estimation approach based on the fact that any stationary time series can be approximated by an AR model of some order (although it might be a high order). In this

approach a suitable AR model is found, and then the spectral density is estimated as the spectral density for that estimated AR model.This method is supported by a theorem which says that the spectral density of any time series process can be approximated by the spectral density of an AR model. I do this using the **spec.ar()** function, and I see that the spectral density gives an alternative view of my stationary time series data by visualizing an AR(19) spectrum.

Forecast



The final step is predicting or forecasting the next three years (2015 to 2017) and the next five years (2015-2020). I use the **forecast()** function from the {forecast} package. The two plots contains the historical time series as well as my ARIMA model's predicted forecast. The plot also includes the 80% and 95% prediction intervals. The 95% prediction intervals are shaded light grey, the 80% prediction intervals are shaded dark grey, and my model's prediction is colored a dark blue. The forecasts suggests continued seasonal behavior. If interested in seeing the other model's forecasts, please see Appendix.

Conclusion

The answer to my question is that health and personal care stores do follow a seasonal pattern. Knowing this information conclusively, as was done in this report, will help prepare them for the temporary increases or decreases in inventory as demand for their products and services fluctuates over certain periods. Additionally, knowing our data follows a seasonal pattern is a big contribution towards the forecasting and prediction of future trends. It can be concluded that health and personal care stores are going to steadily continue being utilized by society.

References

https://www.statista.com/statistics/269555/percentage-of-product-class-sales-of-walgreens-in-the-us-since-2005/

https://www.walgreens.com/images/pdfs/state.pdf

https://www.census.gov/retail/marts/www/timeseries.html

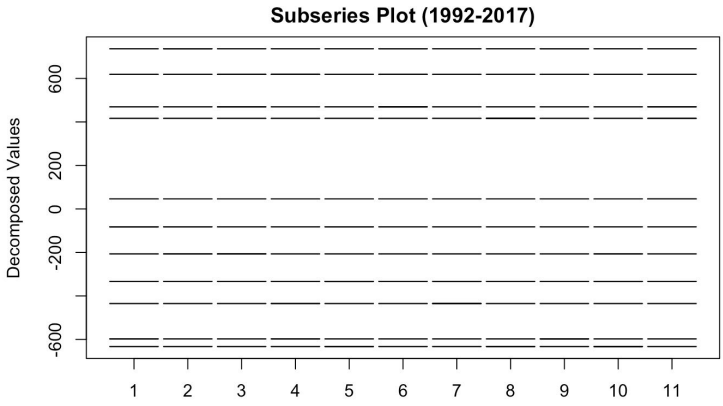
APPENDIX

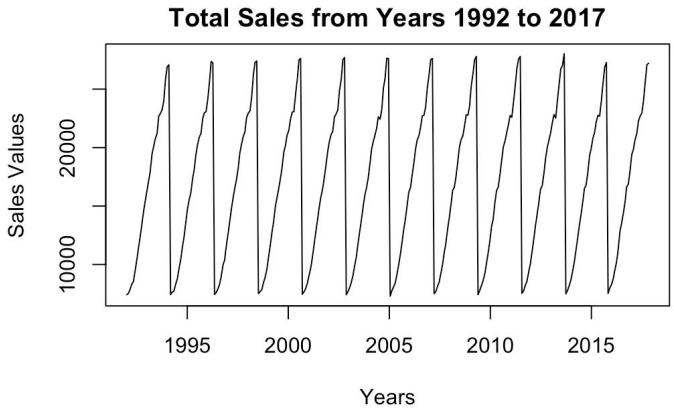
Additional Plots and Outputs

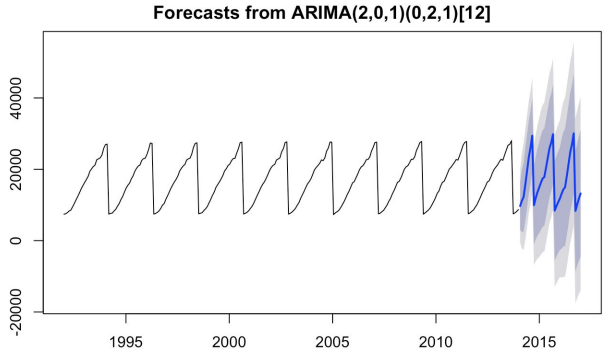
```
> library(sarima)
Error: package or namespace load failed for 'sarima' in dyn.load(file, DLLpath = DLLpath, ...):
 unable to load shared object '/Library/Frameworks/R.framework/Versions/3.4/Resources/library/FitAR/libs/FitAR.so':
  `maximal number of DLLs reached...
In addition: Warning message:
package 'sarima' was built under R version 3.4.2
```

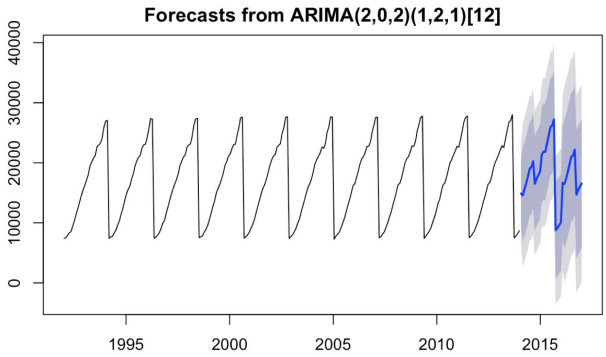
```
BATS(0.449, {0,0}, 0.904, -)

Call: tbats(y = train.ts)

Parameters
  Lambda: 0.449321
  Alpha: 1.021859
  Beta: -0.1204093
  Damping Parameter: 0.903513

Seed States:
         [,1]
[1,] 65.37473
[2,] 11.92180

Sigma: 18.88624
AIC: 5839.824
```

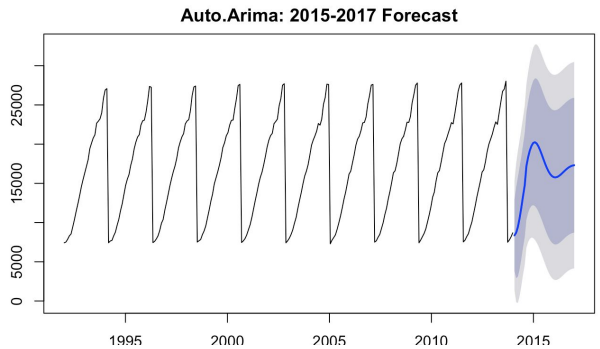
```
ETS(M,A,N)

Call:
 ets(y = train.ts)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0034

  Initial states:
    l = 6154.5416
    b = 682.2821

  sigma:  0.1468

     AIC     AICc      BIC
5578.205 5578.438 5596.085
```

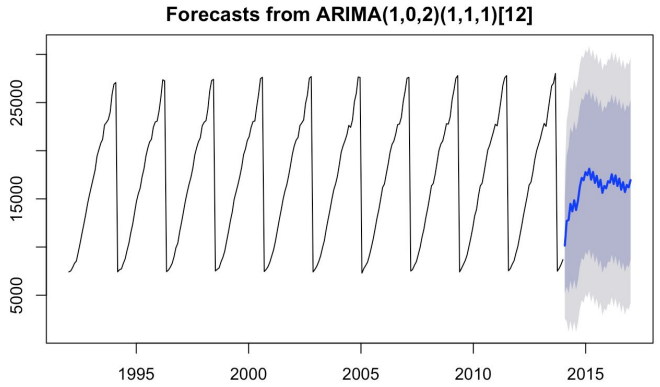


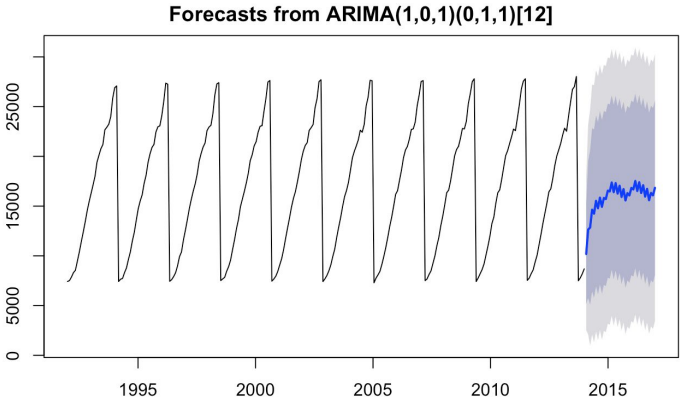




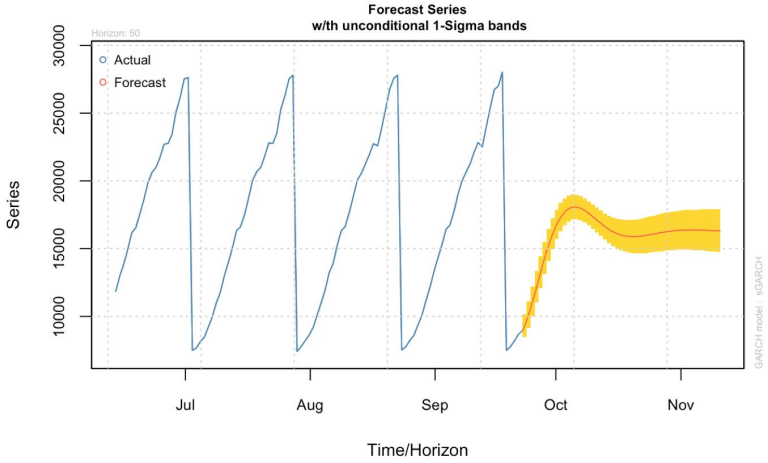


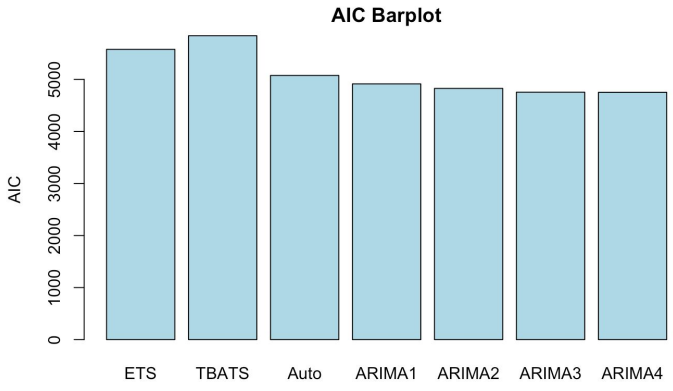


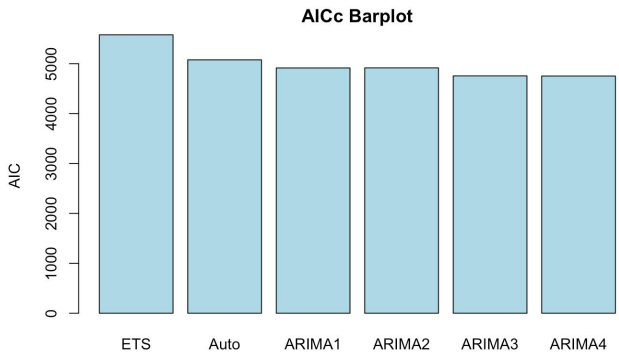


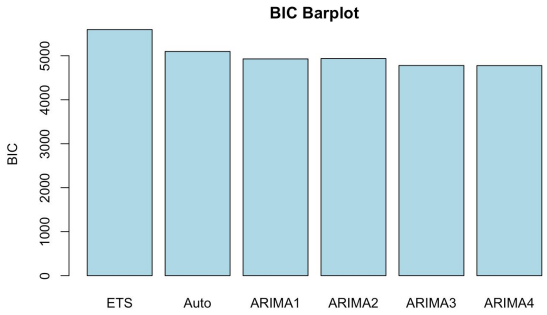


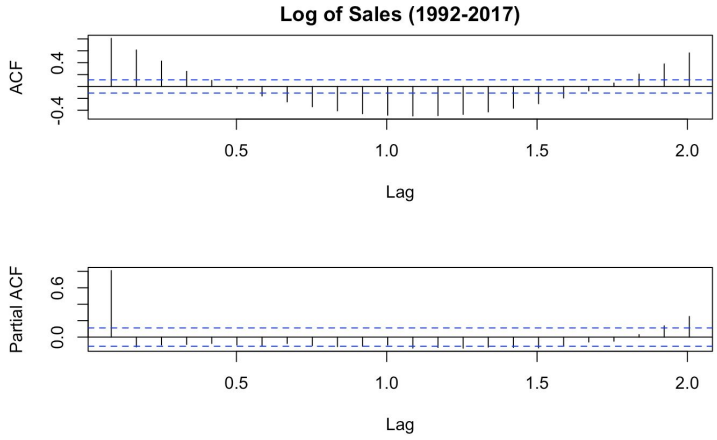
Model <fctr>	ARIMA <fctr>	Seasonal <fctr>	AIC <fctr>
Auto.Arima	(2,0,1)	(0,0,1)	5076.6
ARIMA1	(1,0,1)	(0,1,1)	4914.304
ARIMA2	(2,0,1)	(0,2,1)	4827.634
ARIMA3	(2,0,2)	(1,2,1)	4754.82
ARIMA4	(2,1,2)	(1,2,1)	4752.304

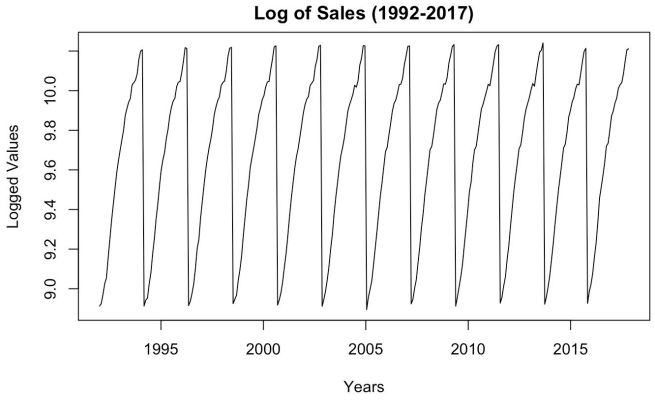


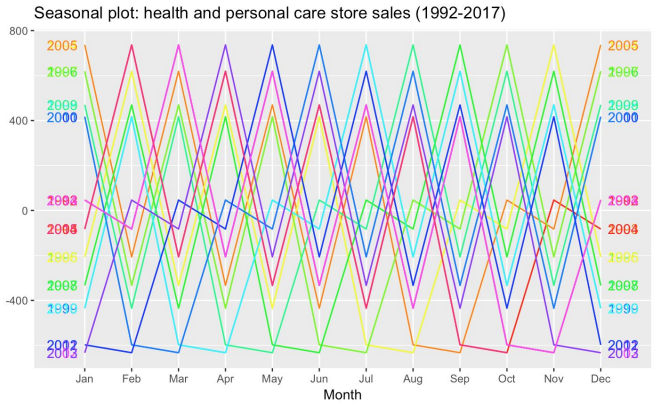




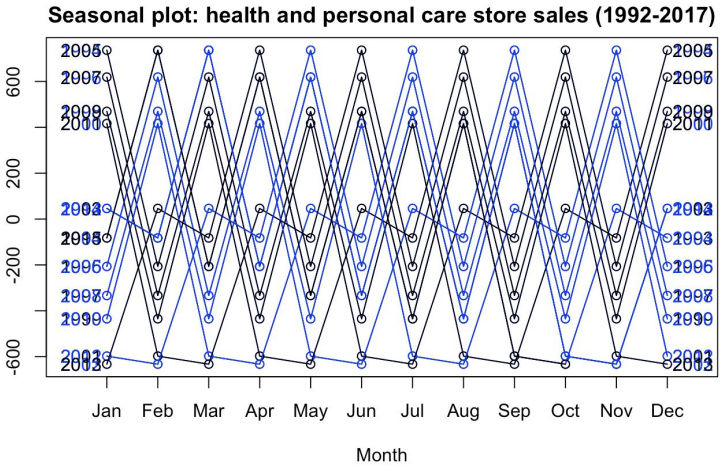




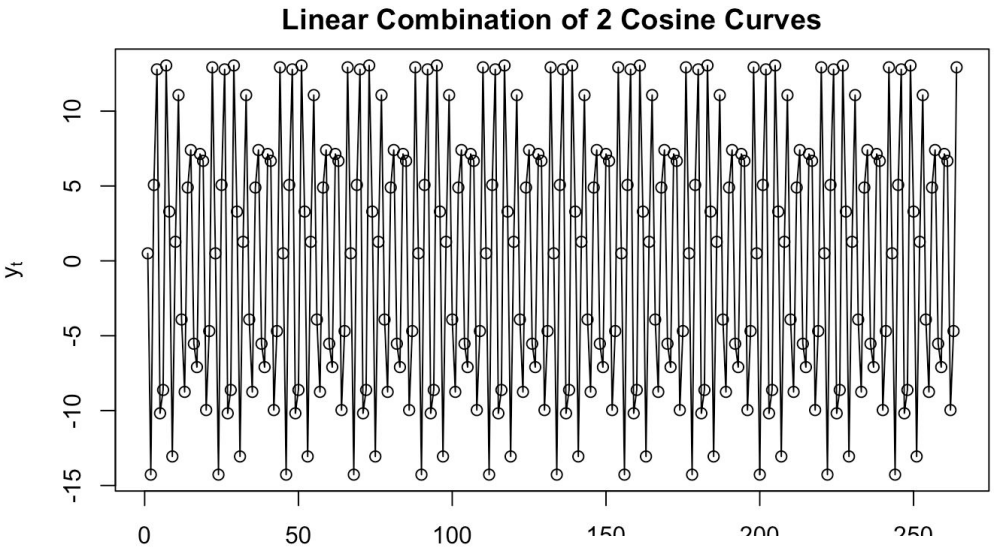


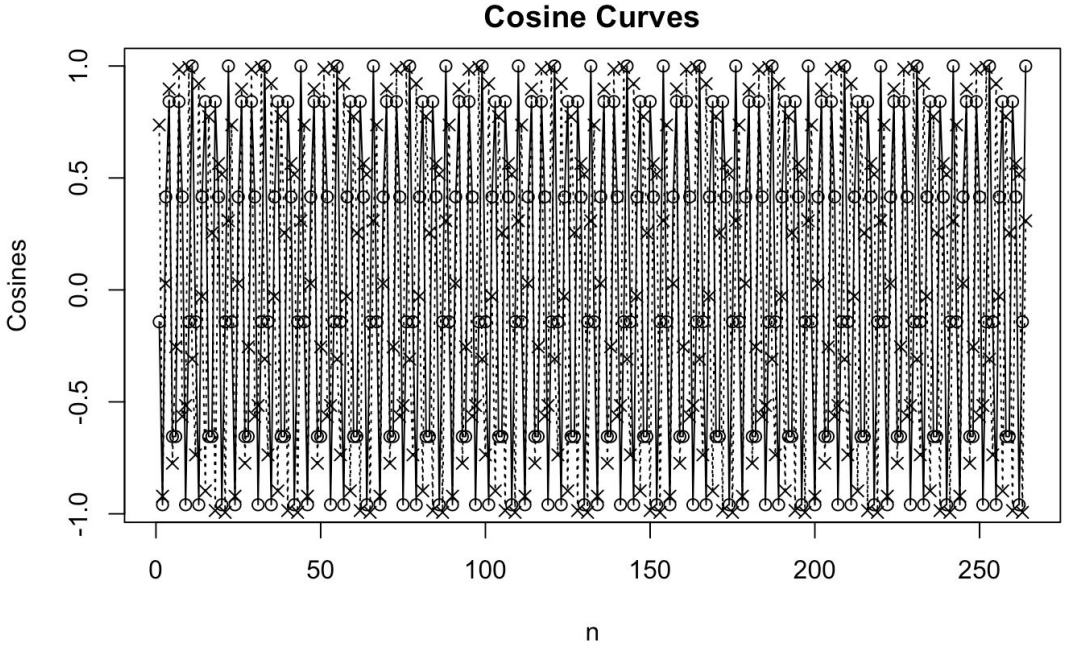


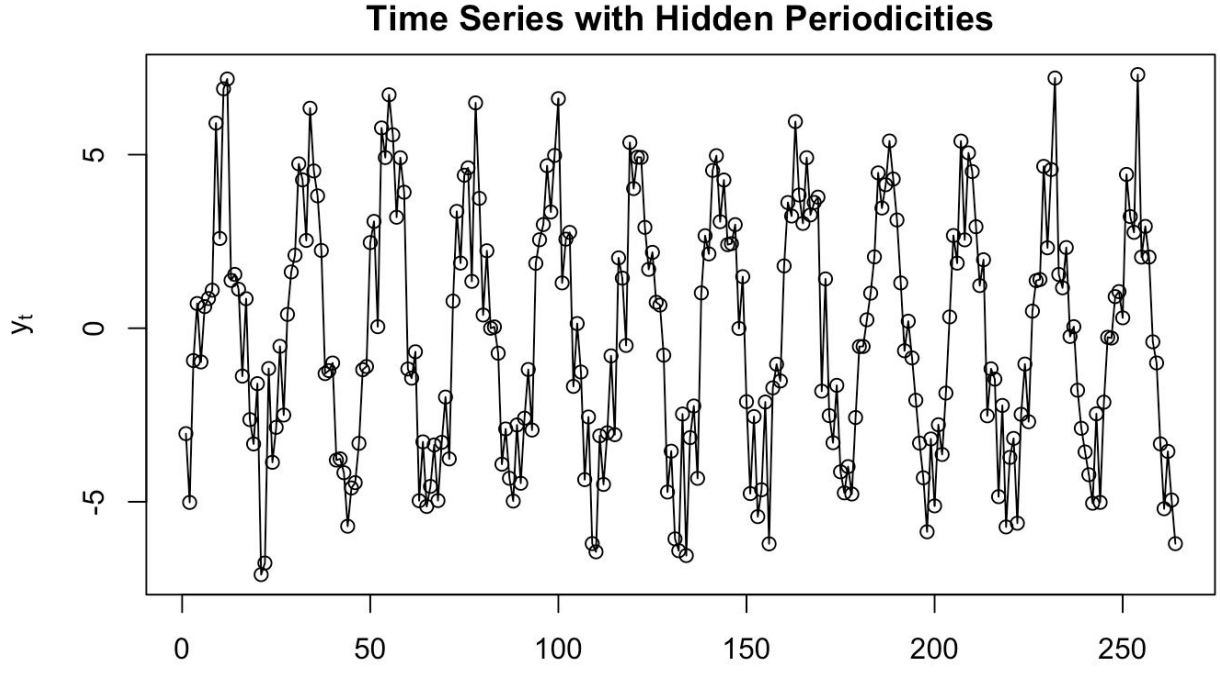
After Decomposition



After Decomposition


Linear Combination of 2 Cosine Curves
y_t
10
5
0
-5
-10
-15
0
50
100
150
200
250


Cosine Curves
Cosines
1.0
0.5
0.0
-0.5
-1.0
0
50
100
150
200
250
n


Time Series with Hidden Periodicities
y_t
5
0
-5
0
50
100
150
200
250