Applied Time Series Analysis

Avocado Prices

Barbara Schmitz & Tiffany Geistkemper

12/9/2023

**Introduction**

For our project, we decided to use a variety of time series modeling techniques from class to explore a relationship with time for avocado prices. Originally, we sourced our dataset from https://hassavocadoboard.com/. This dataset spans four years from 2015 to 2018 (more specifically from 1/4/15 to 3/25/18) with features such as average price, date in a weekly format, and type of avocado, i.e., conventional or organic. We did not include any additional columns relating to total volume, bag sizes, or region for simplicity.

Through analyzing exploratory data analysis, stationarity, seasonality, autocorrelation and partial autocorrelation patterns, and residual analysis, we conclude that although the data was challenging to fit to any model in within the scope of this class, for both conventional and organic avocado prices are best modeled by an ARMA (p=1, q=2) model.   
  
**Exploratory Data Analysis**

**Normality**

First, we investigated the normality of the average price feature for both conventional and organic avocados. We plotted density plots ([Fig 1](#Fig1)), Q-Q plots ([Fig 2](#Fig2)), as well as performed a Shapiro-Wilks test in the code. The conventional avocado density plot looks more positively skewed, and the organic density plot looks slightly bimodal in comparison ([Fig 1](#Fig1)). The Q-Q plots deviate from their respective diagonal line particularly at the tails ([Fig 2](#Fig2)). It is difficult to confidently determine normality with just graphs. Our Shapiro-Wilks test resulted in a p-value of less than 0.05 for conventional and organic. Because our p-values are below our significance value of 0.05, we can reject the null hypothesis of normality.   
  
A graph of an average price

Description automatically generated  
Fig 1. Density Plots  
A graph of different colored lines

Description automatically generated with medium confidence  
Fig 2. Q-Q Plots  
  
Looking at the data as a time series by average price over time ([Fig 3](#Fig3)), we see there is no clear upward or downward trend, and the variation is not very consistent which brings into question its stationarity. However, we do learn conventional avocados are consistently less expensive than organic avocados. Plotting these separating with their respective mean of average price over time, we see they hover somewhat equally above and below their mean lines ([Fig 4](#Fig4), [Fig 5](#Fig5)). The mean price for conventional avocados is $1.16 and for organic avocados it is $1.65.

A graph of avocado price

Description automatically generated  
Fig 3. Time series plot  
A graph with numbers and lines

Description automatically generated  
Fig 4. Conventional Avocado Time Series with Mean line

A graph with blue lines

Description automatically generated  
Fig 5. Conventional Avocado Time Series with Mean line

**Seasonality**

Looking at our data through a yearly lens ([Fig 6](#Fig6), [Fig 7](#Fig7)), we were curious if there were any patterns of cyclical-ness. However, results are inconclusive: there is no clear pattern year to year for either avocado type. However, it is difficult to tell with only four years of data.

A graph of an average avocado price

Description automatically generated  
Fig 6. Yearly Line Plot  
A graph of different colored bars

Description automatically generated  
Fig 7. Yearly Bar Plot  
  
Examining through a monthly lens proves more insightful ([Fig 8](#Fig8), [Fig 9](#Fig9)), particularly around the months of September to November. Prices are consistently higher during these peak periods, likely due to avocados being out of season and volume being the lowest during this time.

A graph showing different colored lines

Description automatically generated  
Fig 8. Monthly Line Plot

A graph of different colored bars

Description automatically generated  
Fig 9. Monthly Bar Plot

There is a similar story for the weekly perspective ([Fig 10](#Fig10), [Fig 11](#Fig11)). This is as granular as we can go as there is not daily or hourly data. There are more peak periods, but the primary one is during the weeks that fall during the September to November months, such as weeks 36 to 46. Also, as we see this through a weekly lens, it looks less stationary.

A graph of different colored lines

Description automatically generated  
Fig 10. Weekly Line Plot (the x axis is 2015.1, 2016.2,2017.3,2018.4, 2015.2, 2016.2… with .x being the week number)  
A graph of avocado prices

Description automatically generated  
Fig 11. Weekly Bar Plot  
  
**Stationarity**

Since the weekly view did not look stationary, we tested this using the augmented Dickey-Fuller test. We tested for each conventional ([Fig 12](#Fig12)), lag 1 conventional ([Fig 13](#Fig13)), organic ([Fig 14](#Fig14)), and lag 1 organic ([Fig 15](#Fig15)). The lag 1 conventional and lag 1 organic will be explained later in this analysis. We can reject the null hypothesis of nonstationarity for each because the p-values are smaller than 0.05, which indicates that each time series is stationary. However, the lag order is at its max value for the function for each, and the ar() function in R suggested this should have been higher, which indicates these results may be unreliable.  
  
A black text on a white background

Description automatically generated  
Fig 12. Conventional Avocado ADF Test  
A close-up of a computer screen

Description automatically generated  
Fig 13. Lag 1 Conventional Avocado ADF Test  
A black text on a white background

Description automatically generated  
Fig 14. Organic Avocado ADF Test  
A close-up of a computer screen

Description automatically generated

Fig 15. Lag 1 Organic Avocado ADF Test

**Models**

Once our initial exploratory analysis was complete, we focused on modeling the time series. We decided to use cosine trends, ARMA, and ARIMA models to attempt to fit our model.

**Cosine Trends Model**

The main features of our cosine trends model include the following:

* + is the zero mean unobserved variance(for all *t*)
  + is the deterministic trend linear model.
* Time is measured by year and fractional year (weekly), i.e., 2015.0 for the first week of 2015, 2015 and (2015.01923) for the second week of 2015, etc.
  + A frequency of 1 (i.e., ) corresponds to an annual or a 52-week period, and frequency of (i.e., ) corresponds to a weekly period.
* The constant term, , is a cosine with frequency zero.

Our cosine models do not fit our data very well as shown in [Fig 16](#Fig16) and [Fig 17](#Fig17) and by the low R-squared values (i.e., far from 1). Our cosine models included:

* (min m)
  + No significant p-values
  + R-squared = 0.01308
  + 1 significant p-value: p-value is 0.0289.
  + R-squared = 0.09524
* (max m)
  + 2 significant p-values: p-value is 0.0335 and p-value is 0.0149.
  + R-squared = 0.2171

For Organic, our cosine models were:

* (min m)
  + No significant p-values.
  + R-squared = 0.02988
  + No significant p-values.
  + R-squared = 0.123
* (max m)
  + 1 significant p-value: p-value was 0.00025
  + R-squared = 0.2743

A graph with numbers and lines

Description automatically generated  
Fig 16. Cosine trend Model Conventional Avocados  
  
A graph with numbers and lines

Description automatically generated  
Fig 17. Cosine Model Organic Avocados.

**ARMA and ARIMA Models**

**ACF and PACF Plots**

We then moved on to AR/MA/ARMA/ARIMA models. To determine which of these models to try, we investigated autocorrelation and partial autocorrelation function plots. The lags in the ACF and PACF plots for the non-differenced conventional ([Fig 18](#Fig18) and [Fig 19](#Fig19)), the lag 1 differenced conventional ([Fig 20](#Fig20) and [Fig 21](#Fig21)), the non-differenced organic ([Fig 22](#Fig22) and [Fig 23](#Fig23)), and the lag 1 differenced organic ([Fig 24](#Fig24) and [Fig 25](#Fig25)) do not go below the threshold after a certain point. In addition, none of the plots have any pattern that we have studied that would suggest a transformation would fix this (i.e., the ACF and PACF plots decay and/or fluctuate for all and do not linearly decline). This suggests that neither the conventional or organic will have strictly an AR(p) component or an MA(q) component. Looking at the plots together, this leads us down the path of testing strictly for an ARMA(p,q) and an ARIMA(p,d,q) model.  
  
A graph with lines and text

Description automatically generated  
Fig 18. Conventional Avocado ACF Plot  
A graph with lines and text

Description automatically generated  
Fig 19. Conventional Avocado PACF Plot  
A graph with lines and text

Description automatically generated  
Fig 20. Lag 1 Differenced Conventional Avocado ACF Plot  
A graph with lines and numbers

Description automatically generated  
Fig 21. Lag 1 Differenced Conventional Avocado PACF Plot  
  
A graph of a graph

Description automatically generated with medium confidence  
Fig 22. Organic Avocado ACF Plot  
A graph with lines and text

Description automatically generated  
Fig 23. Organic Avocado PACF Plot  
A graph with lines and numbers

Description automatically generated  
Fig 24. Lag 1 Differenced Organic Avocado ACF Plot  
A graph with lines and numbers

Description automatically generated  
Fig 25. Lag 1 Differenced Organic Avocado PACF Plot

**Autoregressive Moving Average (ARMA) Model**

The main features of our ARMA model include the following:

* ​ are autoregressive coefficients.
* are moving average coefficients.
* is white noise.

For conventional avocados, using an eacf plot to find suitable p and q values for an ARMA model, we get an ARMA(1,2) model ([Fig 26](#Fig26)). There is no distinct triangle pattern, so there are likely more appropriate models for this data. Using the maximum likelihood to estimate coefficients, our model becomes ARMA(1,2) with ([Fig 27](#Fig27)). The AIC is -2,818.59, we will compare this to the AIC for the Conventional Avocado ARIMA model to see which is more appropriate.  
  
A black and white chart with numbers and x

Description automatically generated with medium confidence   
Fig 26. Conventional Avocado EACF Plot

A screenshot of a computer

Description automatically generated  
Fig 27. Coefficients and AIC for ARMA (1,2) Conventional Avocados  
  
For the organic avocados, the eacf plot also shows that an ARMA(1,2) model could be suitable ([Fig 28](#Fig28)). Using the maximum likelihood estimate of the coefficients makes the model ARMA(1,2) with ([Fig 29](#Fig29)). The AIC is 5900.76, we will compare this to the AIC for the Organic Avocado ARIMA model to see which is more appropriate. Note, we could have also tried an ARMA (2,1) model but for simplicity, we did not use this.  
  
A black and white grid with numbers and letters

Description automatically generated with medium confidence  
Fig 28. Organic Avocado EACF Plot

A white background with black text

Description automatically generated  
Fig 29. Coefficients and AIC for ARMA (1,2) Organic Avocados

**Autoregressive Integrated Moving Average (ARIMA) Model:**

The main features of our ARIMA model include the following:

* are autoregressive coefficients.
* are moving average coefficients.
* are the lagged avocado prices.
* is white noise.
* are the differences errors up to order d.

For conventional avocados, using an eacf plot to find suitable p and q values for an ARIMA model for our lag 1 differenced data, we get an ARIMA (0,1,2) model ([Fig 30](#Fig30)). There is no distinct triangle pattern so there are likely more appropriate models for this data. Using the maximum likelihood to estimate coefficients, our model becomes ARIMA (p=0, d=1, q=2) with ([Fig 31](#Fig31)). The AIC is -2788.6, which is greater than the AIC of -2818.59 from the non-differenced conventional avocado ARMA (1,2) model.   
**A black and white grid with numbers and letters

Description automatically generated with medium confidence**  
Fig 30. Lag 1 Conventional Avocado EACF Plot  
**A white background with black text

Description automatically generated**  
Fig 31. Coefficients and AIC for ARIMA (0,1,2) Conventional Avocados  
  
For organic avocados, using an eacf plot to find suitable p and q values for an ARIMA model for our lag 1 differenced data, we get an ARIMA(0,1,2) model ([Fig 32](#Fig32)). There is no distinct triangle pattern so there are likely more appropriate models for this data. Using the maximum likelihood to estimate coefficients, our model becomes ARIMA(p=0, d=1, q=2) with ([Fig 33](#Fig33)). The AIC is 6021.09, which is greater than the AIC of 5900.76 from the non-differenced organic avocado ARMA(1,2) model.  
  
**A black and white grid with letters and numbers

Description automatically generated**  
Fig 32. Lag 1 Organic Avocado EACF Plot  
**A white background with black text

Description automatically generated**  
Fig 33. Coefficients and AIC for ARIMA (0,1,2) Organic Avocados

**Residual Analysis**

To further investigate the fit of the ARMA (1,2) and the ARIMA(0,1,2) model, we can use residual analysis.  
  
**ARMA(1,2)**  
Performing a runs test for conventional avocados, we can reject the null hypothesis of independence for the ARMA for Conventional (p-value is 4.07e-09).  
For organic avocados, we fail to reject the null hypothesis of independence for the ARMA Organic (p-value is 0.21).  In both cases our level of significance was 0.05. Both of these runs can be found in the code provided at the end.

Performing the Shapiro-Wilks test for conventional avocados, we reject the null hypothesis of normality for the ARMA (p-value is 2.30e-07).  
For organic avocados, we reject the null hypothesis of normality for the ARMA (p-value is 2.20e-16). In both cases our level of significance was 0.05. Both of these runs can be found in the code provided at the end.

Plotting Q-Q plots for the residuals, visually we can see this data struggles with normality as the tails deviate from the middle diagonal line ([Fig 34](#Fig34) and [Fig 35](#Fig35)).   
 A graph of a line

Description automatically generated  
Fig 34. Q-Q Plot Conventional Avocado ARMA (1,2) Residuals   
A graph of a graph

Description automatically generated  
Fig 35. Q-Q Plot Organic Avocado ARMA (1,2) Residuals  
  
**ARIMA (0,1,2)**  
Using a runs test for lag 1 differenced conventional avocados, we can reject the null hypothesis of independence for the ARIMA for Conventional (p-value is 6.26e-07).  
For lag 1 differenced organic avocados, we can reject the null hypothesis of independence for the ARIMA Organic (p-value is 2.86-15).  In both cases our level of significance was 0.05. Both of these runs can be found in the code provided at the end.

Using the Shapiro-Wilks test for conventional avocados, we reject the null hypothesis of normality for the ARMA (p-value is 2.25e-07).  
For organic avocados, we reject the null hypothesis of normality for the ARMA (p-value is 2.2e-16). In both cases our level of significance was 0.05. Both of these runs can be found in the code provided at the end.

Plotting Q-Q plots for the residuals, visually we can see this data struggles with normality as the tails deviate from the middle diagonal line, even more so than in the previous ARMA (1,2) models ([Fig 36](#Fig36) and [Fig 37](#Fig37)). This is especially true for organic avocados.

A graph of a line

Description automatically generated  
Fig 36. Q-Q Plot Conventional Avocado ARIMA (0,1,2) Residuals  
A graph with a line

Description automatically generated  
Fig 37. Q-Q Plot Organic Avocado ARIMA (0,1,2) Residuals  
  
**Conclusion**  
We modeled a time series relationship for both conventional and organic avocados using a cosine trends model, an ARMA (1, 2) model, and an ARIMA (0, 1, 2) model. Our findings show that both conventional and organic avocados are best modeled by the ARMA (1,2) model when comparing AIC and Q-Q plots. We are confident that there are better models for this complex time series data, however using the methods learned in the scope of this class, the ARMA (1,2) fit the best.

For future improvements, we could include the column “total volume” in our analysis and see what influence this feature has on the response variable “average price”. In addition, we could test using subset selection and see which combination of p and q yield the best RMSE for ARMA models. We could also use machine learning methods or statistical learning methods such as support vector machines, decision trees, random forest, gradient boosting methods, or neural network architecture such as a long short term memory (LSTM).

Avocado Prices Code

Barbara Schmitz & Tiffany Geistkemper

2023-12-09

##Loading Libraries

library(tidyverse)  
library(scales)  
library(lubridate)  
library(TSA)  
library(tseries)  
library(RColorBrewer)  
library(gridExtra)  
library(stats)

##Settings

#options(scipen = 999)

##Loading Dataset

df <- read.csv("avocado.csv")

head(df)

## X Date AveragePrice Total.Volume X4046 X4225 X4770 Total.Bags  
## 1 0 2015-12-27 1.33 64236.62 1036.74 54454.85 48.16 8696.87  
## 2 1 2015-12-20 1.35 54876.98 674.28 44638.81 58.33 9505.56  
## 3 2 2015-12-13 0.93 118220.22 794.70 109149.67 130.50 8145.35  
## 4 3 2015-12-06 1.08 78992.15 1132.00 71976.41 72.58 5811.16  
## 5 4 2015-11-29 1.28 51039.60 941.48 43838.39 75.78 6183.95  
## 6 5 2015-11-22 1.26 55979.78 1184.27 48067.99 43.61 6683.91  
## Small.Bags Large.Bags XLarge.Bags type year region  
## 1 8603.62 93.25 0 conventional 2015 Albany  
## 2 9408.07 97.49 0 conventional 2015 Albany  
## 3 8042.21 103.14 0 conventional 2015 Albany  
## 4 5677.40 133.76 0 conventional 2015 Albany  
## 5 5986.26 197.69 0 conventional 2015 Albany  
## 6 6556.47 127.44 0 conventional 2015 Albany

#Cleaning Data

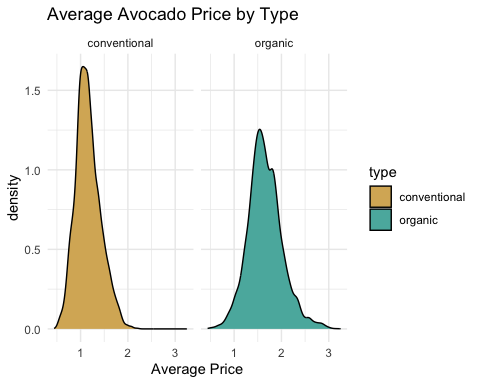
df <- df %>%  
 select(Date, AveragePrice, type) %>%  
 mutate(Date = ymd(Date),  
 type = factor(type),  
 month = month(Date),  
 year = year(Date)) %>%  
 arrange(Date)  
  
head(df)

## Date AveragePrice type month year  
## 1 2015-01-04 1.22 conventional 1 2015  
## 2 2015-01-04 1.00 conventional 1 2015  
## 3 2015-01-04 1.08 conventional 1 2015  
## 4 2015-01-04 1.01 conventional 1 2015  
## 5 2015-01-04 1.02 conventional 1 2015  
## 6 2015-01-04 1.40 conventional 1 2015

##Checking for Normality

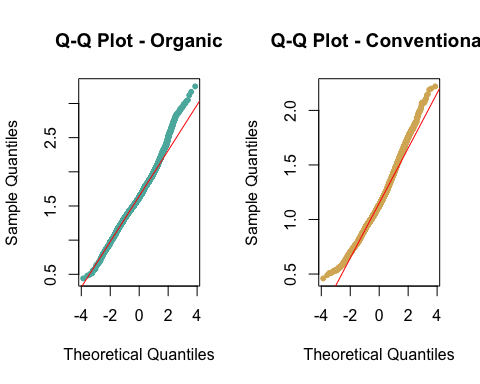
Both organic and conventional types are normally distributed, slightly more positive/right skew for the conventional avocado type.

densityplot <- df%>%  
 ggplot(aes(x=AveragePrice, fill = type))+  
 geom\_density()+  
 facet\_wrap(~type)+  
 labs(x = "Average Price",  
 y="density",  
 title = "Average Avocado Price by Type")+  
 scale\_fill\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365"))+  
 theme\_minimal()  
  
print(densityplot)

 We can inspect with qqplots to confirm. They look mostly normal except at the tail ends.

organic\_df <- df%>%  
 filter(type=="organic")  
conventional\_df <-df%>%  
 filter(type=="conventional")

par(mfrow = c(1, 2))  
qqnorm(organic\_df$AveragePrice, main = "Q-Q Plot - Organic", col = "#5ab4ac", pch = 20)  
qqline(organic\_df$AveragePrice, col = "red")  
  
  
qqnorm(conventional\_df$AveragePrice, main = "Q-Q Plot - Conventional", col = "#d8b365", pch = 20)  
qqline(conventional\_df$AveragePrice, col = "red")



#Creating sample dfs for shapiro wilks function

sampled\_organic <- organic\_df %>% #sampling to 5000 because apparently shapiro.wilks won't work for greater than 5000.   
 sample\_n(5000, replace = FALSE)  
  
sampled\_conventional <- conventional\_df %>% #sampling to 5000 because apparently shapiro.wilks won't work for greater than 5000.   
 sample\_n(5000, replace = FALSE)

Quantitatively, we can check using a Shapiro-Wilks test. We reject the null hypothesis of normality.

shapiro\_organic <- shapiro.test(sampled\_organic$AveragePrice)  
shapiro\_conventional <- shapiro.test(sampled\_conventional$AveragePrice)  
  
print(paste("Sampled Organic W =", shapiro\_organic$statistic))

## [1] "Sampled Organic W = 0.990047042492909"

print(paste("Sampled Organic p-value =", shapiro\_organic$p.value))

## [1] "Sampled Organic p-value = 2.69726639277983e-18"

print(paste("Sampled Conventional W =", shapiro\_conventional$statistic))

## [1] "Sampled Conventional W = 0.98330261749376"

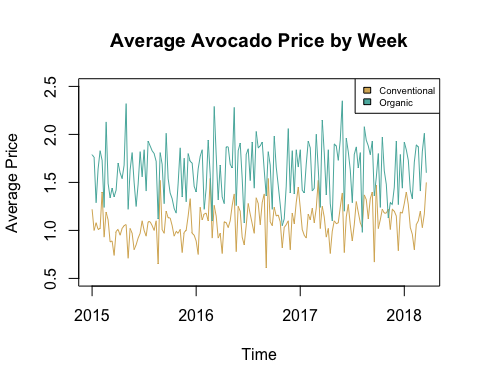
print(paste("Sampled Conventional p-value =", shapiro\_conventional$p.value))

## [1] "Sampled Conventional p-value = 7.44903163904067e-24"

##Time Series

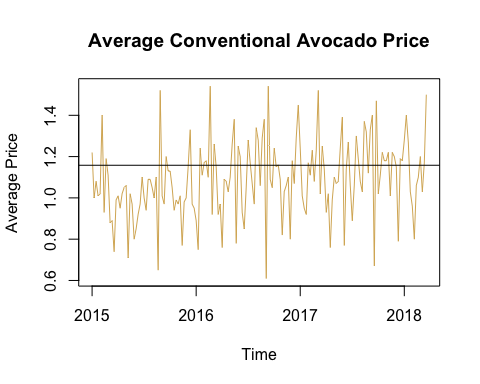
Over time the organic type avocados are more expensive than conventional avocados.

conventional\_ts\_data <- ts(conventional\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52  
plot(conventional\_ts\_data, ylim = c(0.5, 2.5), ylab = 'Average Price', main = 'Average Avocado Price by Week', xaxp = c(2015, 2018, 3), col = "#d8b365")  
organic\_ts\_data <- ts(organic\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52  
lines(organic\_ts\_data, col = "#5ab4ac")  
legend("topright", legend=c("Conventional", "Organic"),   
 fill = c("#d8b365","#5ab4ac"), cex = 0.6)

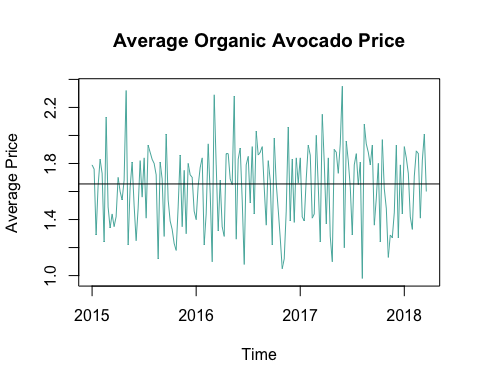


Separating the two series, it is more obvious they seem stationary around the mean value per avocado type.

conventional\_ts\_data <- ts(conventional\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52  
plot(conventional\_ts\_data, ylab = 'Average Price', main = 'Average Conventional Avocado Price', xaxp = c(2015, 2018, 3), col = "#d8b365")  
abline(h = mean(conventional\_df$AveragePrice))



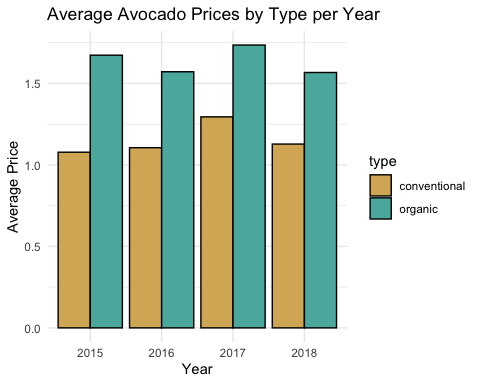
organic\_ts\_data <- ts(organic\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52  
plot(organic\_ts\_data, ylab = 'Average Price', main = 'Average Organic Avocado Price', xaxp = c(2015, 2018, 3), col = "#5ab4ac")  
abline(h = mean(organic\_df$AveragePrice))



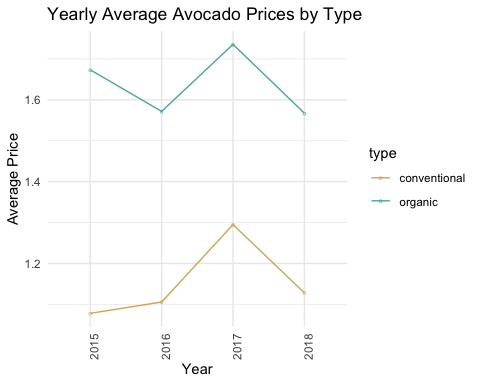
We can further examine these yearly. At the yearly level, neither types follow the same pattern. Not conclusive enough to identify as cyclical.

# Calculate the average of average prices per year per type  
avg\_yearly\_prices <- df %>%  
 group\_by(year = year(Date), type) %>%  
 summarize(avg\_price = mean(AveragePrice))  
  
yearly\_plot <- avg\_yearly\_prices %>%  
 ggplot(aes(x = factor(year), y = avg\_price, fill = type)) +  
 geom\_bar(stat = "identity", position = "dodge", color = "black") +  
 labs(x = "Year", y = "Average Price", title = "Average Avocado Prices by Type per Year") +  
 scale\_fill\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme\_minimal()  
  
line\_plot <- avg\_yearly\_prices %>%  
 ggplot(aes(x = factor(year), y = avg\_price, group = type, color = type)) +  
 geom\_line() +  
 geom\_point(shape = 'o') +  
 labs(x = "Year", y = "Average Price", title = "Yearly Average Avocado Prices by Type") +  
 scale\_color\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))

print(yearly\_plot)



print(line\_plot)



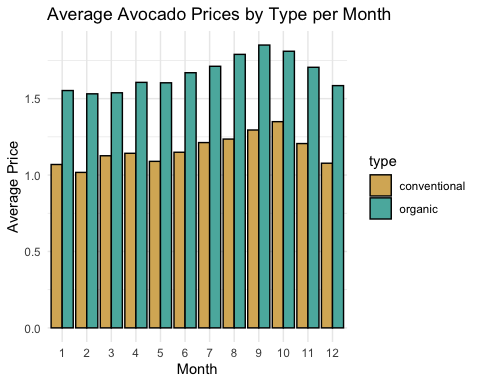
print(avg\_yearly\_prices)

## # A tibble: 8 × 3  
## # Groups: year [4]  
## year type avg\_price  
## <dbl> <fct> <dbl>  
## 1 2015 conventional 1.08  
## 2 2015 organic 1.67  
## 3 2016 conventional 1.11  
## 4 2016 organic 1.57  
## 5 2017 conventional 1.29  
## 6 2017 organic 1.74  
## 7 2018 conventional 1.13  
## 8 2018 organic 1.57

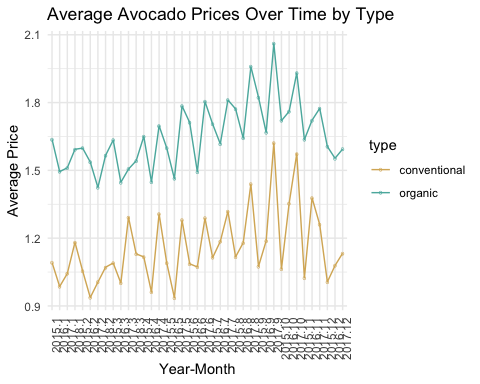
Examining through a monthly lens, there is a pattern. There appears to be seasonality particularly in months September to November. I suspect this is because avocados are in season during spring and summer.

df <- df %>%  
 mutate(month = month(Date), year = year(Date))  
  
# Calculate the average of average prices per month per year and per type  
avg\_prices\_monthly <- df %>%  
 group\_by(month, type) %>%  
 summarize(avg\_price = mean(AveragePrice))  
  
monthly\_plot <- ggplot(data = avg\_prices\_monthly, aes(x = factor(month), y = avg\_price, fill = type)) +  
 geom\_bar(stat = "identity", position = "dodge", color = "black") +  
 labs(x = "Month", y = "Average Price", title = "Average Avocado Prices by Type per Month") +  
 scale\_fill\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme\_minimal()  
  
line\_plot\_monthly <- df %>%  
 group\_by(year, month, type) %>%  
 summarize(avg\_price = mean(AveragePrice)) %>%  
 ggplot(aes(x = interaction(year, month), y = avg\_price, group = type, color = type)) +  
 geom\_line() +  
 geom\_point(shape = 'o') +  
 labs(x = "Year-Month", y = "Average Price", title = "Average Avocado Prices Over Time by Type") +  
 scale\_x\_discrete(labels = function(x) gsub("(\\d+)-(\\d+)", "\\2-\\1", x)) +  
 theme\_minimal() +  
 scale\_color\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))

print(monthly\_plot)



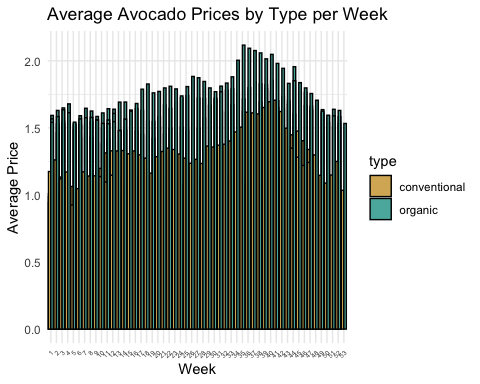
print(line\_plot\_monthly)



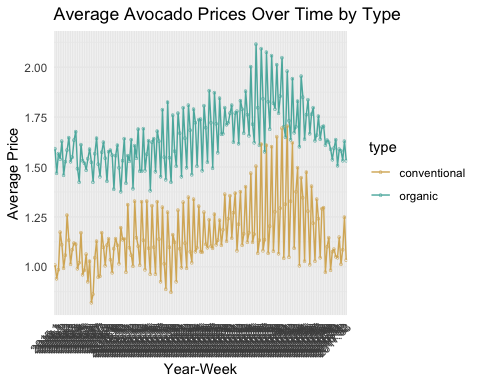
print(avg\_prices\_monthly)

## # A tibble: 24 × 3  
## # Groups: month [12]  
## month type avg\_price  
## <dbl> <fct> <dbl>  
## 1 1 conventional 1.07  
## 2 1 organic 1.55  
## 3 2 conventional 1.02  
## 4 2 organic 1.53  
## 5 3 conventional 1.13  
## 6 3 organic 1.54  
## 7 4 conventional 1.14  
## 8 4 organic 1.61  
## 9 5 conventional 1.09  
## 10 5 organic 1.60  
## # ℹ 14 more rows

# Create week and year variables  
df <- df %>%  
 mutate(week = week(Date), year = year(Date))  
  
# Calculate the average of average prices per week per year and per type  
avg\_prices\_weekly <- df %>%  
 group\_by(week, year, type) %>%  
 summarize(avg\_price = mean(AveragePrice))  
  
# Weekly bar plot  
weekly\_bar\_plot <- ggplot(data = avg\_prices\_weekly, aes(x = factor(week), y = avg\_price, fill = type)) +  
 geom\_bar(stat = "identity", position = "dodge", color = "black") +  
 labs(x = "Week", y = "Average Price", title = "Average Avocado Prices by Type per Week") +  
 scale\_fill\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1, size = 5))  
  
# Weekly line plot over time  
line\_plot\_weekly <- df %>%  
 group\_by(year, week, type) %>%  
 summarize(avg\_price = mean(AveragePrice)) %>%  
 ggplot(aes(x = interaction(year, week), y = avg\_price, group = type, color = type)) +  
 geom\_line() +  
 geom\_point(shape = 'o') +  
 labs(x = "Year-Week", y = "Average Price", title = "Average Avocado Prices Over Time by Type") +  
 scale\_x\_discrete(labels = function(x) gsub("(\\d+)-(\\d+)", "\\2-\\1", x)) +  
 theme\_minimal() +  
 scale\_color\_manual(values = c("organic" = "#5ab4ac", "conventional" = "#d8b365")) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
  
# Print the plots  
print(weekly\_bar\_plot)



print(line\_plot\_weekly)



print(avg\_prices\_weekly)

## # A tibble: 338 × 4  
## # Groups: week, year [169]  
## week year type avg\_price  
## <dbl> <dbl> <fct> <dbl>  
## 1 1 2015 conventional 1.01   
## 2 1 2015 organic 1.59   
## 3 1 2016 conventional 0.942  
## 4 1 2016 organic 1.47   
## 5 1 2017 conventional 0.986  
## 6 1 2017 organic 1.57   
## 7 1 2018 conventional 1.17   
## 8 1 2018 organic 1.54   
## 9 2 2015 conventional 1.11   
## 10 2 2015 organic 1.63   
## # ℹ 328 more rows

##Augmented Dickey Fuller Tests

We compute the augmented Dickey-Fuller (ADF) test statistic for each of the conventional, differenced conventional, organic, and differenced organic avocado datasets and come to the same conclusions for each: We can reject the null hypothesis of nonstationarity because the p-value is smaller than 0.05, which indicates that the time series is stationary. However, the order selected is 20 for the time series, which indicates this test statistic isn’t very meaningful and shouldn’t be taken into consideration as much as other test statistics. In addition, according to the previous function (ar(diff(df))), the order selected could have been around double, making this test statistic even less meaningful.

#Conventional  
ar(conventional\_df$AveragePrice)

##   
## Call:  
## ar(x = conventional\_df$AveragePrice)  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## 0.0472 0.0718 -0.0049 -0.0146 -0.0677 -0.0112 0.0410 0.1018   
## 9 10 11 12 13 14 15 16   
## -0.0183 -0.0032 0.1217 0.3184 -0.0670 0.0424 0.0568 -0.0887   
## 17 18 19 20 21 22 23 24   
## 0.0366 -0.0321 0.0272 0.1070 0.0181 0.0054 -0.1087 0.1167   
## 25 26 27 28 29 30 31 32   
## 0.0585 0.0428 0.1396 0.0702 -0.0084 0.1845 -0.0760 -0.0229   
## 33 34 35 36 37 38   
## 0.0128 0.1040 0.0471 -0.0941 -0.0410 -0.1504   
##   
## Order selected 38 sigma^2 estimated as 0.03167

adf.test(conventional\_df$AveragePrice)

##   
## Augmented Dickey-Fuller Test  
##   
## data: conventional\_df$AveragePrice  
## Dickey-Fuller = -5.1525, Lag order = 20, p-value = 0.01  
## alternative hypothesis: stationary

#Differenced Conventional   
ar(diff(conventional\_df$AveragePrice))

##   
## Call:  
## ar(x = diff(conventional\_df$AveragePrice))  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## -0.9471 -0.8588 -0.8564 -0.8596 -0.9271 -0.9424 -0.8989 -0.7927   
## 9 10 11 12 13 14 15 16   
## -0.8024 -0.8173 -0.6929 -0.3780 -0.4557 -0.4159 -0.3628 -0.4603   
## 17 18 19 20 21 22 23 24   
## -0.4146 -0.4463 -0.4190 -0.3193 -0.3030 -0.2947 -0.4065 -0.2830   
## 25 26 27 28 29 30 31 32   
## -0.2295 -0.1913 -0.0470 -0.0025 -0.0220 0.1609 0.0841 0.0524   
## 33 34 35 36 37 38 39   
## 0.0612 0.1650 0.2164 0.1229 0.0826 -0.0723 -0.0771   
##   
## Order selected 39 sigma^2 estimated as 0.03168

adf.test(diff(conventional\_df$AveragePrice))

##   
## Augmented Dickey-Fuller Test  
##   
## data: diff(conventional\_df$AveragePrice)  
## Dickey-Fuller = -36.571, Lag order = 20, p-value = 0.01  
## alternative hypothesis: stationary

#Organic  
ar(organic\_df$AveragePrice)

##   
## Call:  
## ar(x = organic\_df$AveragePrice)  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## 0.0783 0.0067 0.0325 0.0468 0.0040 -0.0660 0.0425 -0.0063   
## 9 10 11 12 13 14 15 16   
## 0.0726 0.0535 0.0884 0.1596 0.0507 0.0773 0.0182 0.0062   
## 17 18 19 20 21 22 23 24   
## -0.0519 -0.0042 0.0216 0.0392 0.0282 -0.0595 -0.0896 0.0001   
## 25 26 27 28 29 30 31 32   
## 0.0416 0.0495 0.0502 0.0567 0.0549 0.0385 -0.0556 -0.0253   
## 33 34 35 36 37   
## 0.0740 0.0467 0.0417 -0.0086 -0.0574   
##   
## Order selected 37 sigma^2 estimated as 0.1046

adf.test(organic\_df$AveragePrice)

##   
## Augmented Dickey-Fuller Test  
##   
## data: organic\_df$AveragePrice  
## Dickey-Fuller = -8.6863, Lag order = 20, p-value = 0.01  
## alternative hypothesis: stationary

#Differenced Organic   
ar(diff(organic\_df$AveragePrice))

##   
## Call:  
## ar(x = diff(organic\_df$AveragePrice))  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## -0.9183 -0.9069 -0.8655 -0.8151 -0.8120 -0.8792 -0.8384 -0.8377   
## 9 10 11 12 13 14 15 16   
## -0.7571 -0.7042 -0.6178 -0.4605 -0.4119 -0.3365 -0.3189 -0.3086   
## 17 18 19 20 21 22 23 24   
## -0.3496 -0.3468 -0.3260 -0.2882 -0.2592 -0.3153 -0.3983 -0.3967   
## 25 26 27 28 29 30 31 32   
## -0.3552 -0.3109 -0.2646 -0.2199 -0.1702 -0.1337 -0.1920 -0.2132   
## 33 34 35 36 37 38 39   
## -0.1379 -0.0821 -0.0381 -0.0477 -0.1048 -0.0993 -0.0882   
##   
## Order selected 39 sigma^2 estimated as 0.1043

adf.test(diff(organic\_df$AveragePrice))

##   
## Augmented Dickey-Fuller Test  
##   
## data: diff(organic\_df$AveragePrice)  
## Dickey-Fuller = -32.168, Lag order = 20, p-value = 0.01  
## alternative hypothesis: stationary

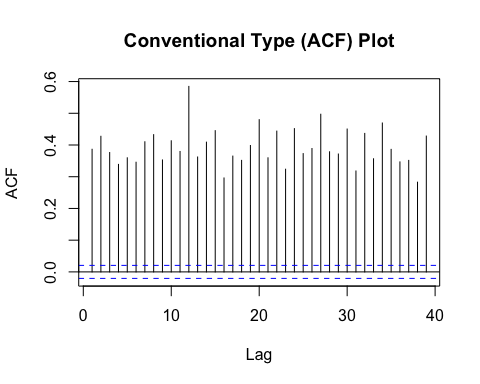
##Autocorrelation and Partial Autocorrelation Analysis

Looking holistically for both conventional and organic avocados, the ACF and PACF plots below.

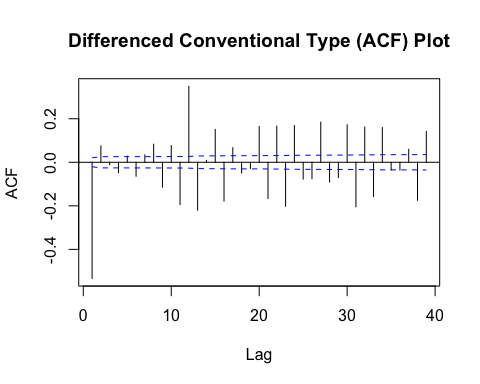
ts\_data <- ts(df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52, start is the first week in 2015, end is the 12th week in 2018  
  
# ACF Analysis  
#acf\_result <- acf(df$AveragePrice, main = "Autocorrelation Function (ACF) Plot")  
#acf\_result <- acf(diff(df$AveragePrice), ci.type = 'ma', main = "Differenced Autocorrelation Function (ACF) Plot") #Note this includes with adjusted ma bounds

# PACF Analysis  
#pacf\_result <- pacf(df$AveragePrice, main = "Partial Autocorrelation Function (PACF) Plot")  
#pacf(diff(df$AveragePrice), main = "Differenced Partial Autocorrelation Function (PACF) Plot")

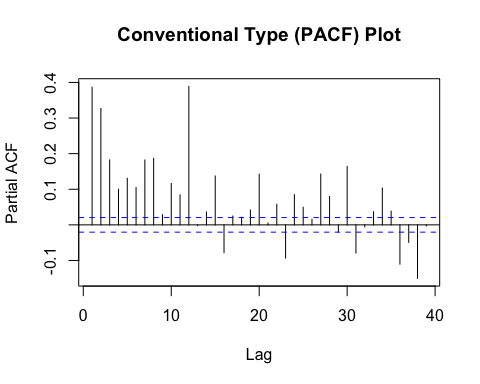
conventional\_ts\_data <- ts(conventional\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52, start is the first week in 2015, end is the 12th week in 2018  
organic\_ts\_data <- ts(organic\_df$AveragePrice, frequency = 52, start=c(2015,1), end = c(2018, 12)) #looking at the data it seems to be weekly so the frequency is 52, start is the first week in 2015, end is the 12th week in 2018  
  
  
# Plot ACF for Conventional Type  
acf(conventional\_df$AveragePrice, main = "Conventional Type (ACF) Plot")



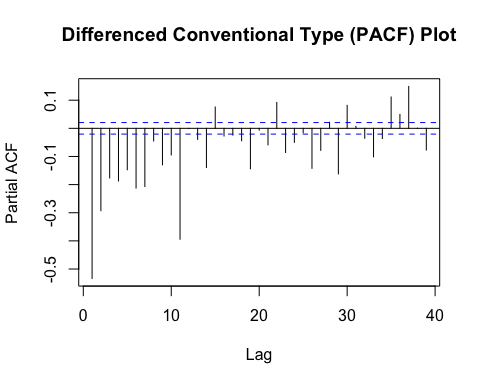
acf(diff(conventional\_df$AveragePrice), ci.type = 'ma', main = "Differenced Conventional Type (ACF) Plot") #Note this includes with adjusted ma bounds



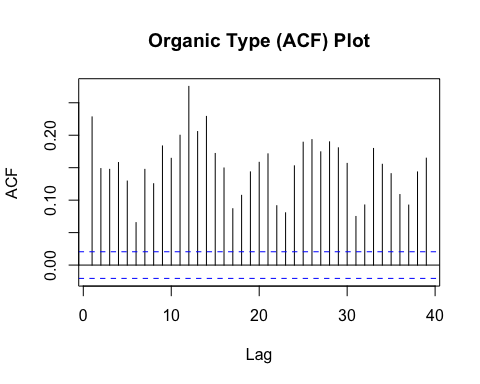
# Plot PACF for Conventional Type  
pacf(conventional\_df$AveragePrice, main = "Conventional Type (PACF) Plot")



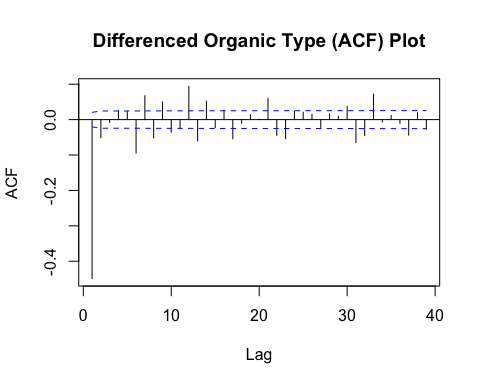
pacf(diff(conventional\_df$AveragePrice), main = "Differenced Conventional Type (PACF) Plot")



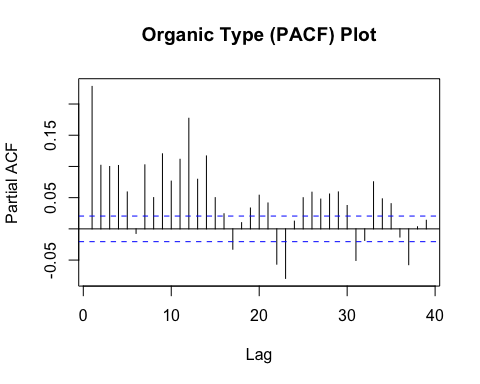
# Plot ACF for Organic Type  
acf(organic\_df$AveragePrice, main = "Organic Type (ACF) Plot")



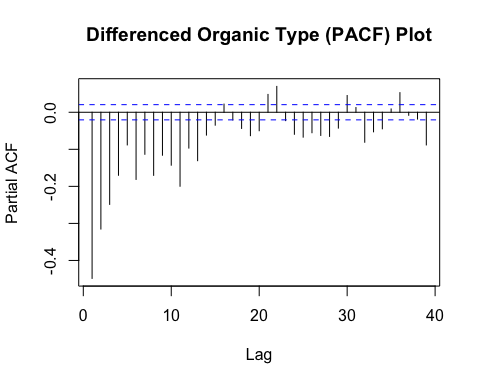
acf(diff(organic\_df$AveragePrice), ci.type='ma', main = "Differenced Organic Type (ACF) Plot") #Note this includes with adjusted ma bounds



# Plot PACF for Organic Type  
pacf(organic\_df$AveragePrice, main = "Organic Type (PACF) Plot")



pacf(diff(organic\_df$AveragePrice), main = "Differenced Organic Type (PACF) Plot")



EACF for Conventional Type

eacf(conventional\_df$AveragePrice)

## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x x x x x x x x x x x x x   
## 1 x x o x x x x x x x x x x o   
## 2 x x o x x x x x x x o x x o   
## 3 x x x x o x o x x x o x x x   
## 4 x x x o o x o x x x x x x x   
## 5 x x x o x o x o o o o x x o   
## 6 x x x x x o x x o o o x x x   
## 7 x x x x x x x o o o o x x x

EACF for Differenced Conventional Type

eacf(diff(conventional\_df$AveragePrice))

## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x o x x x x x x x x x x o   
## 1 x o o x x x o x x x x x x o   
## 2 x o x x o x o x x x o x x o   
## 3 x x x x o x o x x x x x o x   
## 4 x x x o x o x o o o o x o o   
## 5 x x x x x o x o o o o x x x   
## 6 x x x x x x x o o o o x x x   
## 7 x x x x x x x o o o o x x x

EACF for Organic Type

eacf(organic\_df$AveragePrice)

## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x x x x x x x x x x x x x   
## 1 x x o x x x x x x x o x x x   
## 2 x o o o o x o o o x o x x x   
## 3 x x x o o x o o o o o x x o   
## 4 x x x x o x x o o o o x x o   
## 5 x x x x x x x x o o o x x x   
## 6 x x x x x x x o o o x x x x   
## 7 x x x o x x x x x o o x o x

EACF for Differenced Type

eacf(diff(organic\_df$AveragePrice))

## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x o x x x x x x x x x x x   
## 1 x o o o o x o o o x o x x x   
## 2 x x x o o x x o o o o x x o   
## 3 x x x x o x x x o o o x x o   
## 4 x x x x x x o x o o o x x x   
## 5 x x x x x x o x o o x x x x   
## 6 x x x o x x x x x o x x o x   
## 7 x x x o x x x x o o x x o o

##Model Fitting

##Cosine Trends Model

#conventional  
har\_df1=harmonic(conventional\_ts\_data, m = 1)  
model\_df1=lm(conventional\_ts\_data~har\_df1)  
summary(model\_df1)

##   
## Call:  
## lm(formula = conventional\_ts\_data ~ har\_df1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.49834 -0.10846 -0.00815 0.10231 0.46496   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.08865 0.01422 76.543 <2e-16 \*\*\*  
## har\_df1cos(2\*pi\*t) 0.01120 0.01999 0.561 0.576   
## har\_df1sin(2\*pi\*t) -0.02811 0.02020 -1.391 0.166   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1836 on 165 degrees of freedom  
## Multiple R-squared: 0.01308, Adjusted R-squared: 0.001114   
## F-statistic: 1.093 on 2 and 165 DF, p-value: 0.3376

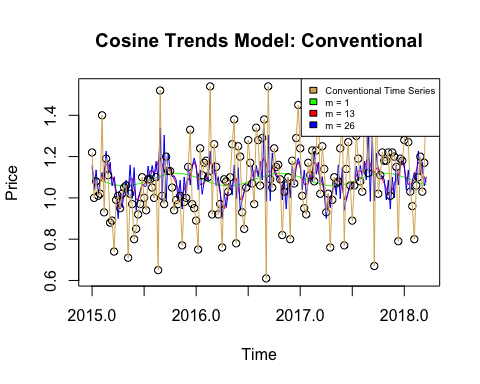
ts\_model\_df1 <- ts(fitted(model\_df1),freq=52, start=c(2015,1), end = c(2018, 12))  
  
har\_df2=harmonic(conventional\_ts\_data, m = 13)  
model\_df2=lm(conventional\_ts\_data~har\_df2)  
summary(model\_df2)

##   
## Call:  
## lm(formula = conventional\_ts\_data ~ har\_df2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.55474 -0.09024 -0.00320 0.10023 0.42483   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.0865053 0.0147758 73.533 <2e-16 \*\*\*  
## har\_df2cos(2\*pi\*t) 0.0087739 0.0207909 0.422 0.6737   
## har\_df2cos(4\*pi\*t) 0.0078402 0.0209402 0.374 0.7087   
## har\_df2cos(6\*pi\*t) 0.0173872 0.0208020 0.836 0.4047   
## har\_df2cos(8\*pi\*t) -0.0125688 0.0209231 -0.601 0.5490   
## har\_df2cos(10\*pi\*t) 0.0127517 0.0208201 0.612 0.5412   
## har\_df2cos(12\*pi\*t) 0.0180048 0.0209017 0.861 0.3905   
## har\_df2cos(14\*pi\*t) -0.0074141 0.0208439 -0.356 0.7226   
## har\_df2cos(16\*pi\*t) 0.0076516 0.0208751 0.367 0.7145   
## har\_df2cos(18\*pi\*t) 0.0049056 0.0208628 0.235 0.8144   
## har\_df2cos(20\*pi\*t) 0.0002102 0.0208536 0.010 0.9920   
## har\_df2cos(22\*pi\*t) 0.0008705 0.0208768 0.042 0.9668   
## har\_df2cos(24\*pi\*t) -0.0018266 0.0208189 -0.088 0.9302   
## har\_df2cos(26\*pi\*t) 0.0024882 0.0208284 0.119 0.9051   
## har\_df2sin(2\*pi\*t) -0.0314765 0.0210015 -1.499 0.1362   
## har\_df2sin(4\*pi\*t) 0.0460194 0.0208532 2.207 0.0289 \*   
## har\_df2sin(6\*pi\*t) 0.0071890 0.0209901 0.342 0.7325   
## har\_df2sin(8\*pi\*t) -0.0162127 0.0208680 -0.777 0.4385   
## har\_df2sin(10\*pi\*t) -0.0399711 0.0209707 -1.906 0.0587 .   
## har\_df2sin(12\*pi\*t) -0.0054695 0.0208897 -0.262 0.7938   
## har\_df2sin(14\*pi\*t) -0.0018469 0.0209452 -0.088 0.9299   
## har\_df2sin(16\*pi\*t) 0.0014088 0.0209098 0.067 0.9464   
## har\_df2sin(18\*pi\*t) -0.0036790 0.0209200 -0.176 0.8607   
## har\_df2sin(20\*pi\*t) -0.0035825 0.0209296 -0.171 0.8643   
## har\_df2sin(22\*pi\*t) 0.0083322 0.0208964 0.399 0.6907   
## har\_df2sin(24\*pi\*t) -0.0141901 0.0209146 -0.678 0.4986   
## har\_df2sin(26\*pi\*t) -0.0040866 0.0208284 -0.196 0.8447   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1902 on 141 degrees of freedom  
## Multiple R-squared: 0.09524, Adjusted R-squared: -0.0716   
## F-statistic: 0.5708 on 26 and 141 DF, p-value: 0.9519

ts\_model\_df2 <- ts(fitted(model\_df2),freq=52, start=c(2015,1), end = c(2018, 12))  
  
har\_df3=harmonic(conventional\_ts\_data, m = 26)  
model\_df3=lm(conventional\_ts\_data~har\_df3)  
summary(model\_df3)

##   
## Call:  
## lm(formula = conventional\_ts\_data ~ har\_df3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.4367 -0.1042 0.0000 0.1100 0.3975   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.0864904 0.0151572 71.681 <2e-16 \*\*\*  
## har\_df3cos(2\*pi\*t) 0.0088304 0.0213275 0.414 0.6796   
## har\_df3cos(4\*pi\*t) 0.0080043 0.0214839 0.373 0.7101   
## har\_df3cos(6\*pi\*t) 0.0174893 0.0213397 0.820 0.4141   
## har\_df3cos(8\*pi\*t) -0.0125894 0.0214666 -0.586 0.5587   
## har\_df3cos(10\*pi\*t) 0.0127797 0.0213613 0.598 0.5508   
## har\_df3cos(12\*pi\*t) 0.0181981 0.0214421 0.849 0.3978   
## har\_df3cos(14\*pi\*t) -0.0072136 0.0213874 -0.337 0.7365   
## har\_df3cos(16\*pi\*t) 0.0076819 0.0214161 0.359 0.7205   
## har\_df3cos(18\*pi\*t) 0.0049069 0.0214119 0.229 0.8191   
## har\_df3cos(20\*pi\*t) 0.0004633 0.0213946 0.022 0.9828   
## har\_df3cos(22\*pi\*t) 0.0013259 0.0214293 0.062 0.9508   
## har\_df3cos(24\*pi\*t) -0.0015234 0.0213824 -0.071 0.9433   
## har\_df3cos(26\*pi\*t) 0.0025641 0.0214355 0.120 0.9050   
## har\_df3cos(28\*pi\*t) 0.0004764 0.0213824 0.022 0.9823   
## har\_df3cos(30\*pi\*t) 0.0068052 0.0214293 0.318 0.7514   
## har\_df3cos(32\*pi\*t) -0.0119591 0.0213946 -0.559 0.5773   
## har\_df3cos(34\*pi\*t) -0.0013913 0.0214119 -0.065 0.9483   
## har\_df3cos(36\*pi\*t) -0.0130065 0.0214161 -0.607 0.5448   
## har\_df3cos(38\*pi\*t) 0.0069061 0.0213874 0.323 0.7473   
## har\_df3cos(40\*pi\*t) 0.0047164 0.0214421 0.220 0.8263   
## har\_df3cos(42\*pi\*t) -0.0303635 0.0213613 -1.421 0.1579   
## har\_df3cos(44\*pi\*t) 0.0117940 0.0214666 0.549 0.5838   
## har\_df3cos(46\*pi\*t) 0.0236443 0.0213397 1.108 0.2702   
## har\_df3cos(48\*pi\*t) -0.0117752 0.0214839 -0.548 0.5847   
## har\_df3cos(50\*pi\*t) 0.0041331 0.0213275 0.194 0.8467   
## har\_df3cos(52\*pi\*t) 0.0201122 0.0151572 1.327 0.1871   
## har\_df3sin(2\*pi\*t) -0.0315691 0.0215430 -1.465 0.1455   
## har\_df3sin(4\*pi\*t) 0.0460053 0.0213871 2.151 0.0335 \*   
## har\_df3sin(6\*pi\*t) 0.0072938 0.0215309 0.339 0.7354   
## har\_df3sin(8\*pi\*t) -0.0161596 0.0214044 -0.755 0.4518   
## har\_df3sin(10\*pi\*t) -0.0400568 0.0215095 -1.862 0.0651 .   
## har\_df3sin(12\*pi\*t) -0.0055278 0.0214289 -0.258 0.7969   
## har\_df3sin(14\*pi\*t) -0.0017328 0.0214836 -0.081 0.9359   
## har\_df3sin(16\*pi\*t) 0.0015432 0.0214549 0.072 0.9428   
## har\_df3sin(18\*pi\*t) -0.0037487 0.0214591 -0.175 0.8616   
## har\_df3sin(20\*pi\*t) -0.0037493 0.0214764 -0.175 0.8617   
## har\_df3sin(22\*pi\*t) 0.0083937 0.0214418 0.391 0.6962   
## har\_df3sin(24\*pi\*t) -0.0138893 0.0214886 -0.646 0.5193   
## har\_df3sin(26\*pi\*t) -0.0039423 0.0214355 -0.184 0.8544   
## har\_df3sin(28\*pi\*t) -0.0050542 0.0214886 -0.235 0.8145   
## har\_df3sin(30\*pi\*t) -0.0007657 0.0214418 -0.036 0.9716   
## har\_df3sin(32\*pi\*t) -0.0025767 0.0214764 -0.120 0.9047   
## har\_df3sin(34\*pi\*t) -0.0115052 0.0214591 -0.536 0.5929   
## har\_df3sin(36\*pi\*t) -0.0080298 0.0214549 -0.374 0.7089   
## har\_df3sin(38\*pi\*t) 0.0056190 0.0214836 0.262 0.7941   
## har\_df3sin(40\*pi\*t) -0.0144585 0.0214289 -0.675 0.5012   
## har\_df3sin(42\*pi\*t) -0.0337152 0.0215095 -1.567 0.1197   
## har\_df3sin(44\*pi\*t) 0.0059977 0.0214044 0.280 0.7798   
## har\_df3sin(46\*pi\*t) 0.0249256 0.0215309 1.158 0.2494   
## har\_df3sin(48\*pi\*t) -0.0193970 0.0213871 -0.907 0.3663   
## har\_df3sin(50\*pi\*t) 0.0532709 0.0215430 2.473 0.0149 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.195 on 116 degrees of freedom  
## Multiple R-squared: 0.2171, Adjusted R-squared: -0.1271   
## F-statistic: 0.6307 on 51 and 116 DF, p-value: 0.9671

ts\_model\_df3 <- ts(fitted(model\_df3),freq=52, start=c(2015,1), end = c(2018, 12))  
  
plot(ts\_model\_df1,ylab='Price',type='l',  
 ylim=range(c(ts\_model\_df1,conventional\_ts\_data)), col = 'green', main = 'Cosine Trends Model: Conventional') # the ylim option ensures that the   
# y axis has a range that fits the raw data and the fitted values  
lines(ts\_model\_df2, col = 'red')  
lines(ts\_model\_df3, col = 'blue')  
points(conventional\_ts\_data)  
lines(conventional\_ts\_data, col = "#d8b365")  
legend("topright", legend=c("Conventional Time Series", "m = 1", "m = 13", "m = 26"),   
 fill = c("#d8b365", "green", "red", "blue"), cex = 0.6)



#Organic  
har\_df1=harmonic(organic\_ts\_data, m = 1)  
model\_df1=lm(organic\_ts\_data~har\_df1)  
summary(model\_df1)

##   
## Call:  
## lm(formula = organic\_ts\_data ~ har\_df1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.67335 -0.22000 0.03908 0.20969 0.64876   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.63023 0.02206 73.902 <2e-16 \*\*\*  
## har\_df1cos(2\*pi\*t) -0.05751 0.03100 -1.855 0.0653 .   
## har\_df1sin(2\*pi\*t) 0.04262 0.03133 1.360 0.1757   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2848 on 165 degrees of freedom  
## Multiple R-squared: 0.02988, Adjusted R-squared: 0.01812   
## F-statistic: 2.541 on 2 and 165 DF, p-value: 0.08188

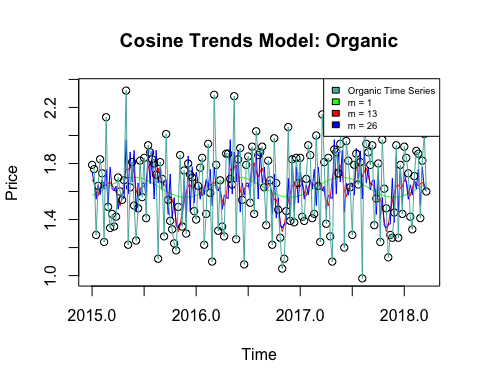
ts\_model\_df1 <- ts(fitted(model\_df1),freq=52, start=c(2015,1), end = c(2018, 12))  
  
har\_df2=harmonic(organic\_ts\_data, m = 13)  
model\_df2=lm(organic\_ts\_data~har\_df2)  
summary(model\_df2)

##   
## Call:  
## lm(formula = organic\_ts\_data ~ har\_df2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72284 -0.20899 0.03501 0.17194 0.62379   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.628e+00 2.276e-02 71.534 <2e-16 \*\*\*  
## har\_df2cos(2\*pi\*t) -6.173e-02 3.202e-02 -1.928 0.0559 .   
## har\_df2cos(4\*pi\*t) 4.801e-02 3.225e-02 1.489 0.1388   
## har\_df2cos(6\*pi\*t) 6.226e-02 3.204e-02 1.943 0.0540 .   
## har\_df2cos(8\*pi\*t) -2.994e-02 3.223e-02 -0.929 0.3544   
## har\_df2cos(10\*pi\*t) 3.497e-03 3.207e-02 0.109 0.9133   
## har\_df2cos(12\*pi\*t) 4.856e-04 3.219e-02 0.015 0.9880   
## har\_df2cos(14\*pi\*t) -1.833e-02 3.210e-02 -0.571 0.5689   
## har\_df2cos(16\*pi\*t) 7.401e-03 3.215e-02 0.230 0.8183   
## har\_df2cos(18\*pi\*t) 2.075e-02 3.213e-02 0.646 0.5195   
## har\_df2cos(20\*pi\*t) 9.871e-03 3.212e-02 0.307 0.7590   
## har\_df2cos(22\*pi\*t) 4.180e-03 3.215e-02 0.130 0.8967   
## har\_df2cos(24\*pi\*t) -6.726e-03 3.207e-02 -0.210 0.8342   
## har\_df2cos(26\*pi\*t) 1.400e-02 3.208e-02 0.437 0.6631   
## har\_df2sin(2\*pi\*t) 4.133e-02 3.235e-02 1.278 0.2034   
## har\_df2sin(4\*pi\*t) 5.057e-02 3.212e-02 1.574 0.1176   
## har\_df2sin(6\*pi\*t) -1.152e-02 3.233e-02 -0.356 0.7220   
## har\_df2sin(8\*pi\*t) -1.676e-02 3.214e-02 -0.522 0.6028   
## har\_df2sin(10\*pi\*t) -5.967e-02 3.230e-02 -1.847 0.0668 .   
## har\_df2sin(12\*pi\*t) 1.774e-02 3.217e-02 0.551 0.5822   
## har\_df2sin(14\*pi\*t) 8.673e-03 3.226e-02 0.269 0.7885   
## har\_df2sin(16\*pi\*t) 6.697e-03 3.221e-02 0.208 0.8356   
## har\_df2sin(18\*pi\*t) -3.322e-03 3.222e-02 -0.103 0.9180   
## har\_df2sin(20\*pi\*t) -2.285e-05 3.224e-02 -0.001 0.9994   
## har\_df2sin(22\*pi\*t) 7.891e-03 3.218e-02 0.245 0.8067   
## har\_df2sin(24\*pi\*t) -8.135e-03 3.221e-02 -0.253 0.8010   
## har\_df2sin(26\*pi\*t) 2.565e-02 3.208e-02 0.800 0.4252   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2929 on 141 degrees of freedom  
## Multiple R-squared: 0.123, Adjusted R-squared: -0.03877   
## F-statistic: 0.7603 on 26 and 141 DF, p-value: 0.7901

ts\_model\_df2 <- ts(fitted(model\_df2),freq=52, start=c(2015,1), end = c(2018, 12))  
  
har\_df3=harmonic(organic\_ts\_data, m = 26)  
model\_df3=lm(organic\_ts\_data~har\_df3)  
summary(model\_df3)

##   
## Call:  
## lm(formula = organic\_ts\_data ~ har\_df3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.56667 -0.18167 -0.00417 0.18312 0.54500   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.628e+00 2.283e-02 71.304 < 2e-16 \*\*\*  
## har\_df3cos(2\*pi\*t) -6.218e-02 3.212e-02 -1.936 0.05532 .   
## har\_df3cos(4\*pi\*t) 4.738e-02 3.236e-02 1.464 0.14583   
## har\_df3cos(6\*pi\*t) 6.170e-02 3.214e-02 1.920 0.05735 .   
## har\_df3cos(8\*pi\*t) -3.029e-02 3.233e-02 -0.937 0.35073   
## har\_df3cos(10\*pi\*t) 3.137e-03 3.217e-02 0.097 0.92251   
## har\_df3cos(12\*pi\*t) -9.465e-05 3.230e-02 -0.003 0.99767   
## har\_df3cos(14\*pi\*t) -1.898e-02 3.221e-02 -0.589 0.55695   
## har\_df3cos(16\*pi\*t) 6.966e-03 3.226e-02 0.216 0.82940   
## har\_df3cos(18\*pi\*t) 2.043e-02 3.225e-02 0.633 0.52769   
## har\_df3cos(20\*pi\*t) 9.326e-03 3.222e-02 0.289 0.77277   
## har\_df3cos(22\*pi\*t) 3.497e-03 3.228e-02 0.108 0.91392   
## har\_df3cos(24\*pi\*t) -6.955e-03 3.221e-02 -0.216 0.82940   
## har\_df3cos(26\*pi\*t) 1.436e-02 3.229e-02 0.445 0.65733   
## har\_df3cos(28\*pi\*t) -1.521e-02 3.221e-02 -0.472 0.63771   
## har\_df3cos(30\*pi\*t) 1.553e-02 3.228e-02 0.481 0.63124   
## har\_df3cos(32\*pi\*t) 1.595e-02 3.222e-02 0.495 0.62148   
## har\_df3cos(34\*pi\*t) 4.230e-03 3.225e-02 0.131 0.89588   
## har\_df3cos(36\*pi\*t) 6.272e-03 3.226e-02 0.194 0.84618   
## har\_df3cos(38\*pi\*t) -1.128e-02 3.221e-02 -0.350 0.72676   
## har\_df3cos(40\*pi\*t) -2.257e-02 3.230e-02 -0.699 0.48607   
## har\_df3cos(42\*pi\*t) -1.109e-02 3.217e-02 -0.345 0.73104   
## har\_df3cos(44\*pi\*t) 2.002e-03 3.233e-02 0.062 0.95074   
## har\_df3cos(46\*pi\*t) -7.443e-04 3.214e-02 -0.023 0.98156   
## har\_df3cos(48\*pi\*t) 5.763e-02 3.236e-02 1.781 0.07750 .   
## har\_df3cos(50\*pi\*t) 2.014e-02 3.212e-02 0.627 0.53193   
## har\_df3cos(52\*pi\*t) 5.288e-04 2.283e-02 0.023 0.98156   
## har\_df3sin(2\*pi\*t) 4.152e-02 3.245e-02 1.279 0.20328   
## har\_df3sin(4\*pi\*t) 5.067e-02 3.221e-02 1.573 0.11844   
## har\_df3sin(6\*pi\*t) -1.159e-02 3.243e-02 -0.357 0.72140   
## har\_df3sin(8\*pi\*t) -1.677e-02 3.224e-02 -0.520 0.60399   
## har\_df3sin(10\*pi\*t) -5.944e-02 3.240e-02 -1.835 0.06913 .   
## har\_df3sin(12\*pi\*t) 1.804e-02 3.228e-02 0.559 0.57733   
## har\_df3sin(14\*pi\*t) 8.783e-03 3.236e-02 0.271 0.78655   
## har\_df3sin(16\*pi\*t) 6.750e-03 3.231e-02 0.209 0.83491   
## har\_df3sin(18\*pi\*t) -3.019e-03 3.232e-02 -0.093 0.92574   
## har\_df3sin(20\*pi\*t) 4.557e-04 3.235e-02 0.014 0.98878   
## har\_df3sin(22\*pi\*t) 8.140e-03 3.229e-02 0.252 0.80145   
## har\_df3sin(24\*pi\*t) -8.039e-03 3.237e-02 -0.248 0.80428   
## har\_df3sin(26\*pi\*t) 2.651e-02 3.229e-02 0.821 0.41333   
## har\_df3sin(28\*pi\*t) -1.273e-02 3.237e-02 -0.393 0.69475   
## har\_df3sin(30\*pi\*t) 1.288e-03 3.229e-02 0.040 0.96827   
## har\_df3sin(32\*pi\*t) 1.141e-02 3.235e-02 0.353 0.72481   
## har\_df3sin(34\*pi\*t) -1.253e-02 3.232e-02 -0.388 0.69903   
## har\_df3sin(36\*pi\*t) 1.445e-03 3.231e-02 0.045 0.96442   
## har\_df3sin(38\*pi\*t) 2.950e-02 3.236e-02 0.912 0.36383   
## har\_df3sin(40\*pi\*t) 2.608e-04 3.228e-02 0.008 0.99357   
## har\_df3sin(42\*pi\*t) -1.813e-02 3.240e-02 -0.560 0.57689   
## har\_df3sin(44\*pi\*t) 1.179e-02 3.224e-02 0.366 0.71526   
## har\_df3sin(46\*pi\*t) 1.225e-01 3.243e-02 3.779 0.00025 \*\*\*  
## har\_df3sin(48\*pi\*t) -1.917e-02 3.221e-02 -0.595 0.55301   
## har\_df3sin(50\*pi\*t) 5.874e-02 3.245e-02 1.810 0.07282 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2937 on 116 degrees of freedom  
## Multiple R-squared: 0.2743, Adjusted R-squared: -0.04479   
## F-statistic: 0.8596 on 51 and 116 DF, p-value: 0.7248

ts\_model\_df3 <- ts(fitted(model\_df3),freq=52, start=c(2015,1), end = c(2018, 12))  
  
plot(ts\_model\_df1,ylab='Price',type='l',  
 ylim=range(c(ts\_model\_df1,organic\_ts\_data)), col = 'green', main = 'Cosine Trends Model: Organic') # the ylim option ensures that the   
# y axis has a range that fits the raw data and the fitted values  
lines(ts\_model\_df2, col = 'red')  
lines(ts\_model\_df3, col = 'blue')  
points(organic\_ts\_data)  
lines(organic\_ts\_data, col = "#5ab4ac")  
legend("topright", legend=c("Organic Time Series", "m = 1", "m = 13", "m = 26"),   
 fill = c("#5ab4ac", "green", "red", "blue"), cex = 0.6)



##Model Fitting

arima(conventional\_df$AveragePrice, order = c(1,0,2), method = 'ML') #ARMA(1,2), this one is better via AIC

##   
## Call:  
## arima(x = conventional\_df$AveragePrice, order = c(1, 0, 2), method = "ML")  
##   
## Coefficients:  
## ar1 ma1 ma2 intercept  
## 0.9989 -0.9846 0.0192 1.1579  
## s.e. 0.0004 0.0100 0.0099 0.0645  
##   
## sigma^2 estimated as 0.04294: log likelihood = 1413.3, aic = -2818.59

arima(diff(conventional\_df$AveragePrice), order = c(0,1,2), method = 'ML') #IMA(2,1)

##   
## Call:  
## arima(x = diff(conventional\_df$AveragePrice), order = c(0, 1, 2), method = "ML")  
##   
## Coefficients:  
## ma1 ma2  
## -1.9633 0.9634  
## s.e. 0.0026 0.0026  
##   
## sigma^2 estimated as 0.04304: log likelihood = 1396.3, aic = -2788.6

arima(organic\_df$AveragePrice, order = c(1,0,2), method = 'ML') #ARMA(1,2) this one is better via AIC

##   
## Call:  
## arima(x = organic\_df$AveragePrice, order = c(1, 0, 2), method = "ML")  
##   
## Coefficients:  
## ar1 ma1 ma2 intercept  
## 0.9991 -0.9021 -0.0800 1.6537  
## s.e. 0.0004 0.0106 0.0105 0.0626  
##   
## sigma^2 estimated as 0.1117: log likelihood = -2946.38, aic = 5900.76

arima(diff(organic\_df$AveragePrice), order = c(0,1,2), method = 'ML') #IMA(1,2)

##   
## Call:  
## arima(x = diff(organic\_df$AveragePrice), order = c(0, 1, 2), method = "ML")  
##   
## Coefficients:  
## ma1 ma2  
## -1.9738 0.9741  
## s.e. 0.0029 0.0029  
##   
## sigma^2 estimated as 0.1131: log likelihood = -3008.54, aic = 6021.09

##Residual Analysis

We can reject the null hypothesis of independence for the ARMA(1,2) Conventional, IMA(1,2) Conventional and IMA(1,2) Organic. We fail to reject the null hypothesis of independence for ARMA(1,2) Organic.

arma\_conv <-arima(conventional\_df$AveragePrice, order = c(1,0,2), method = 'ML')  
arma\_org <-arima(organic\_df$AveragePrice, order = c(1,0,2), method = 'ML')  
  
ima\_conv <-arima(diff(conventional\_df$AveragePrice), order = c(0,1,2), method = 'ML')  
ima\_org <-arima(diff(organic\_df$AveragePrice), order = c(0,1,2), method = 'ML')

# Extract residuals  
residuals\_arma\_conv <- resid(arma\_conv)  
residuals\_arma\_org <- resid(arma\_org)  
residuals\_ima\_conv <- resid(ima\_conv)  
residuals\_ima\_org <- resid(ima\_org)

# Convert residuals to factors because otherwise it breaks  
factor\_residuals\_arma\_conv <- as.factor(sign(residuals\_arma\_conv))  
factor\_residuals\_arma\_org <- as.factor(sign(residuals\_arma\_org))  
factor\_residuals\_ima\_conv <- as.factor(sign(residuals\_ima\_conv))  
factor\_residuals\_ima\_org <- as.factor(sign(residuals\_ima\_org))  
  
# Perform runs tests  
runs\_test\_result\_arma\_conv <- runs.test(factor\_residuals\_arma\_conv)  
runs\_test\_result\_arma\_org <- runs.test(factor\_residuals\_arma\_org)  
runs\_test\_result\_ima\_conv <- runs.test(factor\_residuals\_ima\_conv)  
runs\_test\_result\_ima\_org <- runs.test(factor\_residuals\_ima\_org)  
  
  
print("ARMA Conventional:")

## [1] "ARMA Conventional:"

print(runs\_test\_result\_arma\_conv)

##   
## Runs Test  
##   
## data: factor\_residuals\_arma\_conv  
## Standard Normal = -5.8813, p-value = 4.071e-09  
## alternative hypothesis: two.sided

print("ARMA Organic:")

## [1] "ARMA Organic:"

print(runs\_test\_result\_arma\_org)

##   
## Runs Test  
##   
## data: factor\_residuals\_arma\_org  
## Standard Normal = -1.2562, p-value = 0.209  
## alternative hypothesis: two.sided

print("IMA Conventional:")

## [1] "IMA Conventional:"

print(runs\_test\_result\_ima\_conv)

##   
## Runs Test  
##   
## data: factor\_residuals\_ima\_conv  
## Standard Normal = -4.7183, p-value = 2.379e-06  
## alternative hypothesis: two.sided

print("IMA Organic:")

## [1] "IMA Organic:"

print(runs\_test\_result\_ima\_org)

##   
## Runs Test  
##   
## data: factor\_residuals\_ima\_org  
## Standard Normal = -6.1838, p-value = 6.259e-10  
## alternative hypothesis: two.sided

# Extract residuals  
residuals\_arma\_conv <- resid(arma\_conv)  
sampled\_residuals\_arma\_conv <- as.data.frame(residuals\_arma\_conv) %>%  
 sample\_n(5000, replace = TRUE)  
shapiro.test(sampled\_residuals\_arma\_conv$x)

##   
## Shapiro-Wilk normality test  
##   
## data: sampled\_residuals\_arma\_conv$x  
## W = 0.99827, p-value = 2.441e-05

residuals\_arma\_org <- resid(arma\_org)  
sampled\_residuals\_arma\_org <- as.data.frame(residuals\_arma\_org) %>%  
 sample\_n(5000, replace = TRUE)  
shapiro.test(sampled\_residuals\_arma\_org$x)

##   
## Shapiro-Wilk normality test  
##   
## data: sampled\_residuals\_arma\_org$x  
## W = 0.98929, p-value < 2.2e-16

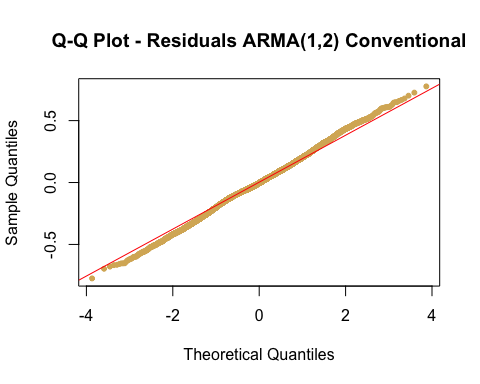
residuals\_ima\_conv <- resid(ima\_conv)  
sampled\_residuals\_ima\_conv <- as.data.frame(residuals\_ima\_conv) %>%  
 sample\_n(5000, replace = TRUE)  
shapiro.test(sampled\_residuals\_ima\_conv$x)

##   
## Shapiro-Wilk normality test  
##   
## data: sampled\_residuals\_ima\_conv$x  
## W = 0.9973, p-value = 9.001e-08

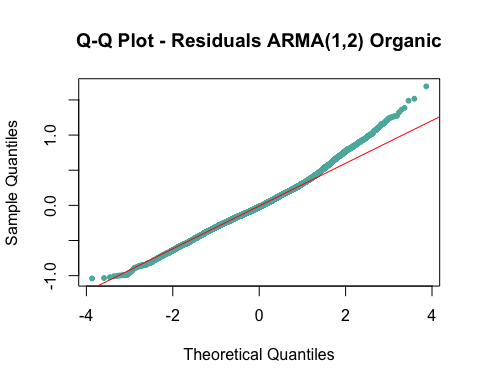
residuals\_ima\_org <- resid(ima\_org)  
sampled\_residuals\_ima\_org <- as.data.frame(residuals\_ima\_org) %>%  
 sample\_n(5000, replace = TRUE)  
shapiro.test(sampled\_residuals\_ima\_org$x)

##   
## Shapiro-Wilk normality test  
##   
## data: sampled\_residuals\_ima\_org$x  
## W = 0.99171, p-value < 2.2e-16

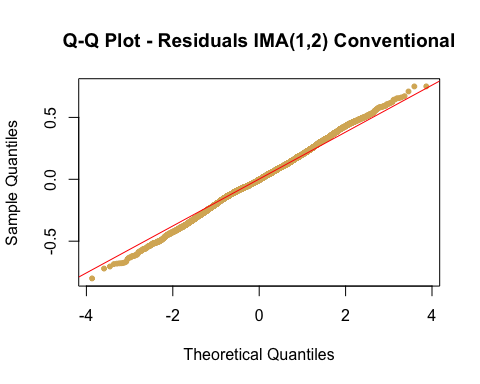
qqnorm(residuals\_arma\_conv, main = "Q-Q Plot - Residuals ARMA(1,2) Conventional", col = "#d8b365", pch = 20)  
qqline(residuals\_arma\_conv, col = "red")



qqnorm(residuals\_arma\_org, main = "Q-Q Plot - Residuals ARMA(1,2) Organic", col = "#5ab4ac", pch = 20)  
qqline(residuals\_arma\_org, col = "red")



qqnorm(residuals\_ima\_conv, main = "Q-Q Plot - Residuals IMA(1,2) Conventional", col = "#d8b365", pch = 20)  
qqline(residuals\_ima\_conv, col = "red")



qqnorm(residuals\_ima\_conv, main = "Q-Q Plot - Residuals IMA(1,2) Organic", col = "#5ab4ac", pch = 20)  
qqline(residuals\_ima\_org, col = "red")

