STAT3022/STAT3922 Assignment 1

Your SID: 470345744

08/04/2019

The assignment is out of 20 marks. The presentation of your report (e.g. lack of spelling error; your sentences make grammatical sense; the visual aspect of your report well presented etc) is worth **2 marks**. Load the `html` output in a browser then print it to a PDF file to submit to Turnitin.

Background (3 marks)

Describe the structure of the data (size, variables, missingness etc) and the aim of the analysis.

1. By using skim function in the skim library, we know that the orginal data(before my manipulation) have 18249 observations in total
2. There are 13 variables in total. Among those variables, 3 are character variables(Date, region and type) and 9 are numerical variables(4046, 4225, 4770, AveragePrice, Total Bags, Small Bags, Large Bags, XLarge Bags and year).
3. There is no missingness in the data.
4. There are 54 original regions in total. 45 of them are cities, 8 of them are larger regions such as California and Plains and the remaining 1 is Total US. Also, I noticed that the total avocado sales of larger regions are the same with the avocado sales of Total US, while the total avocado sales of smaller cities are smaller than both of them. Hence it will be more accurate for me to focus on the region analysis in the later part.
5. The aim of our analysis is to predict the state that provides the cheapest avocado in July 2019. Although the customer gives priority to organic avocado, since the customers' price preference is not known, I will recommend states that provide both cheapest conventional and organic avocados and let the customer decide which type of avocado they want based on the given predict prices.

Executive Summary (3 marks)

Write a brief conclusion/recommendations/key points that your client will be interested most. There should be at least 3 key points made to your client.

Conclusion: 1. In general, organic avocados have a much higher price than conventional ones and states in South Central area are supposed to have the lowest avocado prices for both conventional and organic types in July 2019.

2. If the client is willing to pay $1.684271 per organic avocado in South Central area in July 2019, which is the lowest organic avocado price available across 8 larger regions, then I would recommend the client to work in Texas and Houston in particular, if there is an office.

3. If he/she thinks this price is too high and is willing to accept conventional avocados instead($0.9686191 in South Central area), then I would recommend him/her to work in Arizona and Phoenix or Tucson in particular, if there is an office.

4. Otherwise, since we only have information regarding cities and regions rather than state, we might need the client to provide the city information of each office or other information such as the willingness to travel between cities and traveling budgets.

Key Statistical Analysis (6 marks)

Describe 3 key statistcal analysis that you conducted and shown in the appendix that is relevant to answering the question from the client. This should include a *statistical model* learnt within the course (i.e. do not do a time series analysis) and some form of *hypothesis testing*.

The client's question can be summarized as: Which state should I go so I can consume the cheapest avocados (better organic ones) in July 2019?

The client wants to know which state to go. However, all the regions given in the orginal dataset are mostly cities and large regions. Since the city information of each state office is not provided, I chose to conduct the statistical analysis on all 8 large regions first. Also, since the organic and conventional avocados differ with each other in various aspects such as average price and total volume, I decided to analyse both datasets separately.

By visually inspecting the combined boxplot graph in appendix, it is apparent that South Central area and West area in US have the lowest and second lowest median average price for both organic and conventional avocados. However, the client is more interested in the future prices of avocados in July 2019 and in order to predict them, I constructed 2 multiple linear regression models for conventional and organic avocados in South Central area and similar 2 multiple linear regression models for conventional and organic avocados in West area.

In order to obtain an accurate future price, we need a model with good fitness, which can be affected by what variables we included and what transformation or interaction effect we considered.

Take the MLR model of conventional avocados in South Central area as an example. First I added all the potentital predictors which has a high correlation(>0.2) with the average price in the pairwise graph except for those that might collinearity issues (e.g. Total Volume and Total Bags). The transformation effect is also being considered. These predictors form a full model
$sc.AvgPrice\ sc.smallHass + log(sc.largeHass) + log(sc.xlHass) + log(sc.SmallBags) + sc.LargeBags + sc.XLargeBags + sc.Year$
and then I performed the stepwise selections on the full model.

The first term that I dropped from the full model is $sc.LargeBags$. This is done by the hypothesis testing that $H_0 : \beta_{largeBags} = 0$ vs $H_1 : \beta_{largeBags} \neq 0$. The observed f value is 1.4255 and the corresponding p-value is given as 0.2342605 which is much larger than 0.05. Hence our data are consistent with the null hypothesis that $\beta_{largeBags} = 0$ and it is reasonable for me to drop $sc.LargeBags$ term.

Then I kept dropping similar terms until all the coefficients of the remaining terms are statistically signicant, which is $sc.AvgPrice\ sc.small_hass + log(sc.large_hass) + log(sc.xl_hass) + log(sc.SmallBags) + sc.Year$, with the corresponding coefficients listed in the output.

According to the summary output, 66.26% variation of the average price is being explained by all the predictors in the model, with SER being 0.0825. Also, there is some curved pattern in the residual-fitted plot as well as scale-location plot and the normal Q-Q plot is not very linear, indicating homoscedasticity assumption and normality assumption of MLR model are violated. This part will be further analysed in the discussion part.

Assuming there is no seasonal movement for each predictor value, I averaged all the available July values for each predictor between 2015 and 2018 and took that average value as the predicted value for each predictor in July 2019. By substituting those values back to the MLR model, the predicted average price for a conventional avocado of South Central in July 2019 is obtained.

Then I applied the similar analysis to other 3 MLR models. As a result, the predicted average price for a conventional and organic avocado of South Central area in July 2019 are 0.9686191 and 1.684271. The predicted average price for a conventional and organic avocado of West area in July 2019 are 1.450977 and 2.171533.

Clearly the predicted price for both conventional and organic avocados are lower in South Central area than the ones in West area. Hence among all large regions, it is verified that South Central area has the lowest prices for both types of avocados. Also, it is up to the client's choice whether 1.684271 for organic avocados is affordable or not.

Discussion (3 marks)

Discussion should include at least 3 points that shows some insight or understanding of data.

1. In all 4 MLR models it seems there are some curved pattern in the residual plots and the normal QQ plots are not that linear. This could due to the fact that I did not include any interaction effects for the convenience of interpretation. For example, in the MLR model for conventional avocado in South Central area, I did not include the interaction effect smallBag:smallHass,smallBag:LargeHass and smallBag:LargeHass. These missing parts leave some variation of average price unexplained and increase the residuals as a result.
2. In all 4 MLR models, the predicted value of predictor variables in July 2019 were produced based on the assumption that there is no seasonal effect in the dataset. That is the reason why we can take the average July data in the previous years as the predicted value of predictor variable. However, according to the time series plot of average US avocado price, there is an obvious seasonal movement for both conventional and organic avocados. For example, there are two huge spikes for conventional avocado price in 2017. Hence our predicted value for avocado prices might not be accurate enough and it is necessary to include some time series analysis.
3. The client only wants to know which state provides the cheapest avocado. Since the region data set does not include much state information, I performed the statistical analysis mainly on the large regions and concluded that South Central tends to provide the cheapest avocados for both types in July 2019. However, by inspecting the combined boxplot carefully, it seems the some states have a lower minimum average avocado price compared to South Central. For example, Southeast area for conventional avocados and West area for organic avocados. The average price span for each boxplot varies very much from each other during 2015-2018 and it is totally possible for the client to get the cheapest avocado in some states that are not recommended.
4. In general, it seems cities and regions that are close to Mexico provide the cheapest avocados. Regarding large regions, South Central and West area provides the first and second cheapest avocados and both regions are near Mexico. Regarding cities, Houston, Dallas, Denver and Los Angelos are high on the list for both types of avocados and all those cities are also very close to Mexico. It might due to the greater supply and the lower transportation cost.

Appendix (3 marks)

Please ensure that you do NOT use `echo=F` in any of your outputs. Keep your `.Rmd` file in case it is needed for inspection of your results.

```
library(tidyverse)
library(readr)
library(scales)
library(plotly)
library(wesanderson)
library(ggplot2)
library(gridExtra)
library(reshape2)
avocados = read_csv("avocado.csv")
avocados_orgin = read_csv("avocado.csv")
```

Data Cleaning and Preprocessing

```
skimr::skim(avocados_orgin)
```

	variable <chr>	type <chr>	stat <chr>	level <chr>	value <dbl>	formatted <chr>
1	Date	character	missing	.all	0.000000e+00	0
2	Date	character	complete	.all	1.824900e+04	18249
3	Date	character	n	.all	1.824900e+04	18249
4	Date	character	min	.all	6.000000e+00	6
5	Date	character	max	.all	8.000000e+00	8
6	Date	character	empty	.all	0.000000e+00	0
7	Date	character	n_unique	.all	1.690000e+02	169
8	AveragePrice	numeric	missing	.all	0.000000e+00	0
9	AveragePrice	numeric	complete	.all	1.824900e+04	18249
10	AveragePrice	numeric	n	.all	1.824900e+04	18249

1-10 of 131 rows Previous **1** 2 3 4 5 6 … 14 Next

```
#replace the space with _ in
names(avocados) <- gsub(" ", "_", names(avocados))

#rename avocado types as [small_hass, large_hass, xl_hass and other] and store them in the avocado_type column
avocados <- avocados %>%
  rename(small_hass = "4046", large_hass = "4225", xl_hass = "4770") %>%
  mutate(other = Total_Volume - small_hass - large_hass - xl_hass) %>%
  gather(bag_size, bag_total, c(Small_Bags, Large_Bags, XLarge_Bags)) %>%
  gather(avocado_type, avocado_volume, c(small_hass, large_hass, xl_hass,
                                         other))

#divide the original regions(54) into 3 parts: larger regions(8), cities(45) and total US(1).
#1.subset data by larger region(8)
avocados_region <- avocados %>%
  filter(region %in% c("California", "West", "SouthCentral", "GreatLakes",
                       "Midsouth", "Southeast", "Northeast", "Plains"))

#2.subset data by city market(45)
avocados_market <- avocados %>%
  filter(!(region %in% c("California", "West", "SouthCentral", "GreatLakes",
                       "Midsouth", "Southeast", "Northeast", "Plains",
                       "TotalUS")))

#3.dataset for entire US(1)
avocados_total <- avocados %>%
  filter(region == "TotalUS")
```

Exploratory Analysis

```
#check the structure of the data after manipulation
str(avocados)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    218988 obs. of  11 variables:
##  $ Date          : chr  "27/12/15" "20/12/15" "13/12/15" "6/12/15" ...
##  $ AveragePrice  : num  1.33 1.35 0.93 1.08 1.28 1.26 0.99 0.98 1.02 1.07 ...
##  $ Total_Volume  : num  64237 54877 118220 78992 51040 ...
##  $ Total_Bags    : num  8697 9506 8145 5811 6184 ...
##  $ type          : chr  "conventional" "conventional" "conventional" "conventional" ...
##  $ year          : num  2015 2015 2015 2015 2015 ...
##  $ region        : chr  "Albany" "Albany" "Albany" "Albany" ...
##  $ bag_size      : chr  "Small_Bags" "Small_Bags" "Small_Bags" "Small_Bags" ...
##  $ bag_total     : num  8604 9408 8042 5677 5986 ...
##  $ avocado_type  : chr  "small_hass" "small_hass" "small_hass" "small_hass" ...
##  $ avocado_volume: num  1037 674 795 1132 941 ...
```

```
#easier to check variable types and sizes in this way
skimr::skim(avocados)
```

	variable <chr>	type <chr>	stat <chr>	level <chr>	value <dbl>	formatted <chr>
1	Date	character	missing	.all	0.000000e+00	0
2	Date	character	complete	.all	2.189880e+05	218988
3	Date	character	n	.all	2.189880e+05	218988
4	Date	character	min	.all	6.000000e+00	6
5	Date	character	max	.all	8.000000e+00	8
6	Date	character	empty	.all	0.000000e+00	0
7	Date	character	n_unique	.all	1.690000e+02	169
8	AveragePrice	numeric	missing	.all	0.000000e+00	0
9	AveragePrice	numeric	complete	.all	2.189880e+05	218988
10	AveragePrice	numeric	n	.all	2.189880e+05	218988

1-10 of 101 rows Previous **1** 2 3 4 5 6 … 11 Next

Explore each character variable

```
unique(as.character(avocados$avocado_type))
```

```
## [1] "small_hass" "large_hass" "xl_hass"    "other"
```

```
length(unique(as.character(avocados$avocado_type)))
```

```
## [1] 4
```

```
unique(as.character(avocados$bag_size))
```

```
## [1] "Small_Bags"  "Large_Bags"  "XLarge_Bags"
```

```
length(unique(as.character(avocados$bag_size)))
```

```
## [1] 3
```

```
# all the orginial regions
unique(as.character(avocados$region))
```

```
##  [1] "Albany"              "Atlanta"             "BaltimoreWashington"
##  [4] "Boise"               "Boston"              "BuffaloRochester"
##  [7] "California"          "Charlotte"           "Chicago"
## [10] "CincinnatiDayton"    "Columbus"            "DallasFtWorth"
## [13] "Denver"              "Detroit"             "GrandRapids"
## [16] "GreatLakes"          "HarrisburgScranton"  "HartfordSpringfield"
## [19] "Houston"             "Indianapolis"        "Jacksonville"
## [22] "LasVegas"            "LosAngeles"          "Louisville"
## [25] "MiamiFtLauderdale"   "Midsouth"            "Nashville"
## [28] "NewOrleansMobile"    "NewYork"             "Northeast"
## [31] "NorthernNewEngland"  "Orlando"             "Philadelphia"
## [34] "PhoenixTucson"       "Pittsburgh"          "Plains"
## [37] "Portland"            "RaleighGreensboro"   "RichmondNorfolk"
## [40] "Roanoke"             "Sacramento"          "SanDiego"
## [43] "SanFrancisco"        "Seattle"             "SouthCarolina"
## [46] "SouthCentral"        "Southeast"           "Spokane"
## [49] "StLouis"             "Syracuse"            "Tampa"
## [52] "TotalUS"             "West"                "WestTexNewMexico"
```

```
length(unique(as.character(avocados$region)))
```

```
## [1] 54
```

```
# larger regions
unique(as.character(avocados_region$region))
```

```
## [1] "California"   "GreatLakes"   "Midsouth"     "Northeast"
## [5] "Plains"       "SouthCentral" "Southeast"    "West"
```

```
length(unique(as.character(avocados_region$region)))
```

```
## [1] 8
```

```
# smaller cities
unique(as.character(avocados_market$region))
```

```
##  [1] "Albany"              "Atlanta"             "BaltimoreWashington"
##  [4] "Boise"               "Boston"              "BuffaloRochester"
##  [7] "Charlotte"           "Chicago"             "CincinnatiDayton"
## [10] "Columbus"            "DallasFtWorth"       "Denver"
## [13] "Detroit"             "GrandRapids"         "HarrisburgScranton"
## [16] "HartfordSpringfield" "Houston"             "Indianapolis"
## [19] "Jacksonville"        "LasVegas"            "LosAngeles"
## [22] "Louisville"          "MiamiFtLauderdale"   "Nashville"
## [25] "NewOrleansMobile"    "NewYork"             "NorthernNewEngland"
## [28] "Orlando"             "Philadelphia"        "PhoenixTucson"
## [31] "Pittsburgh"          "Portland"            "RaleighGreensboro"
## [34] "RichmondNorfolk"     "Roanoke"             "Sacramento"
## [37] "SanDiego"            "SanFrancisco"        "Seattle"
## [40] "SouthCarolina"       "Spokane"             "StLouis"
## [43] "Syracuse"            "Tampa"               "WestTexNewMexico"
```

```
length(unique(as.character(avocados_market$region)))
```

```
## [1] 45
```

```
# one TotalUS
unique(as.character(avocados_total$region))
```

```
## [1] "TotalUS"
```

```
length(unique(as.character(avocados_total$region)))
```

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

```
unique(as.character(avocados$type))
```

```
## [1] "conventional" "organic"
```

```
length(unique(as.character(avocados$type)))
```

```
## [1] 2
```

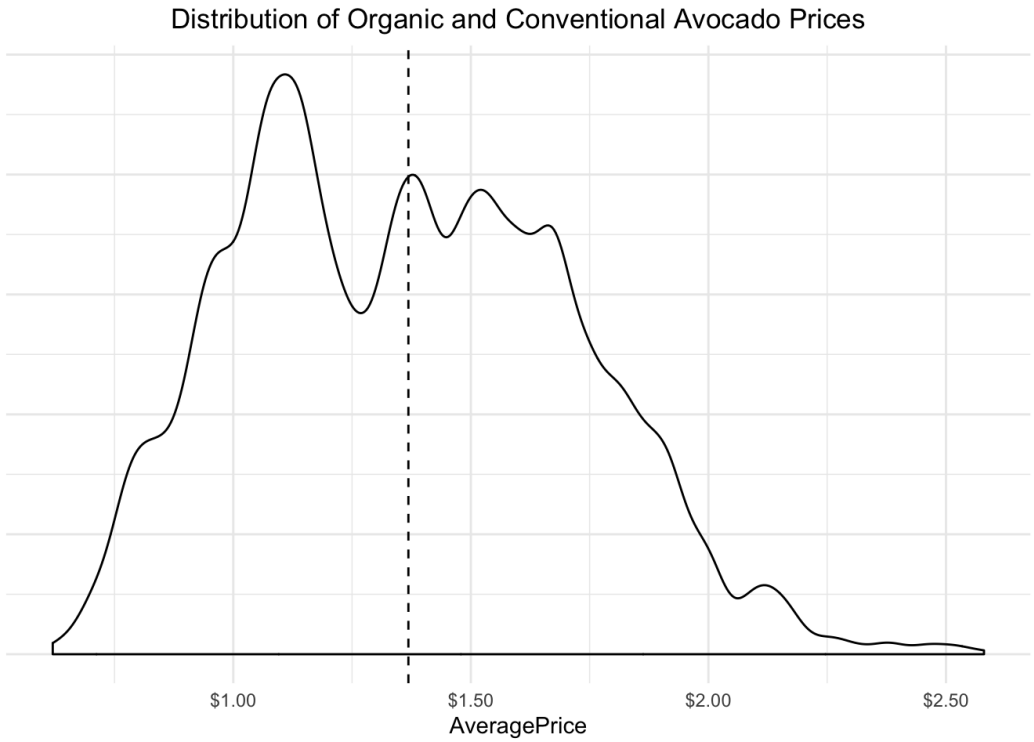
```
#check if there are any missing values
paste(sum(is.na(avocados_market)),
sum(is.na(avocados_region)),
sum(is.na(avocados_total)))
```

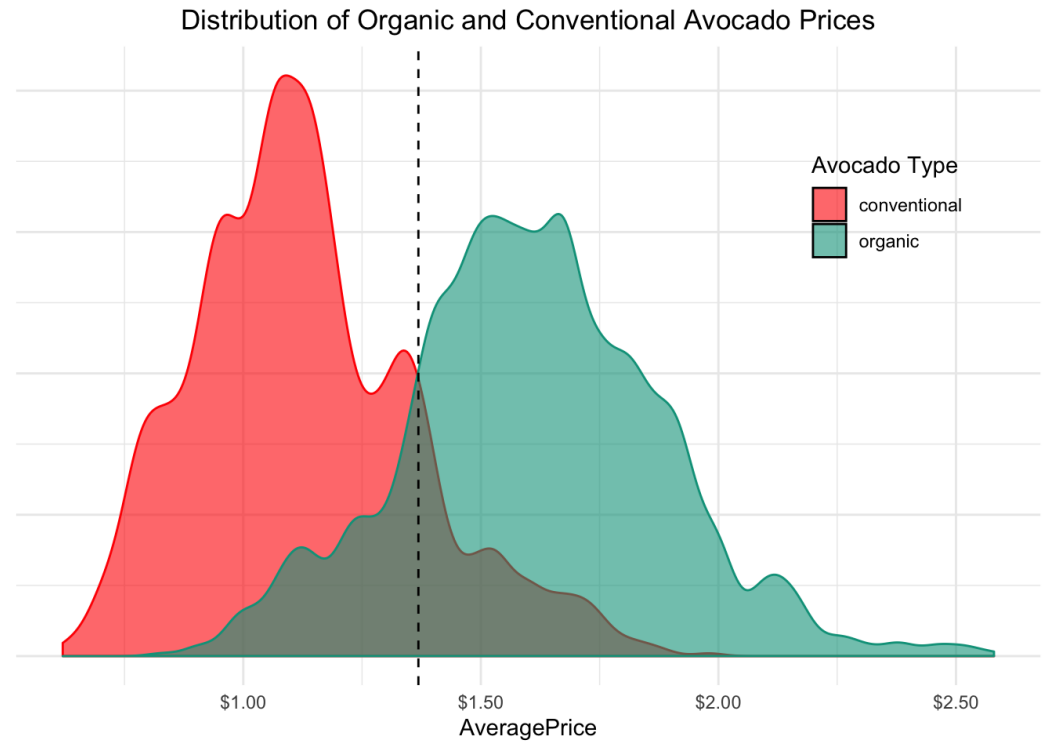
```
## [1] "0 0 0"
```

```
#check if the volume matches
#seems there are some cities' data that havent been included
#region volume matches total volume
cat(paste("Market Volume:",
sum(avocados_market$avocado_volume),"\n"),
paste("Region Volume:",
sum(avocados_region$avocado_volume), "\n"),
paste("Total Volume:",
sum(avocados_total$avocado_volume)))
```

```
## Market Volume: 11381766688.74
##  Region Volume: 17594220546.06
##  Total Volume: 17594220545.4
```

```
ggplot(avocados_region, aes(x = AveragePrice)) +
  geom_density(alpha = .6, bw = .03) +
  geom_vline(aes(xintercept = mean(AveragePrice)), lty = 2) +
  scale_fill_manual(values=wes_palette(n=2, name="Darjeeling1")) +
  scale_color_manual(values=wes_palette(n=2, name="Darjeeling1")) +
  scale_x_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(legend.position = c(.85,.75)) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank(),
        plot.title = element_text(hjust = .5))  +
  labs(fill = "Avocado Type") +
  guides(col = FALSE) +
  ggtitle("Distribution of Organic and Conventional Avocado Prices")
```

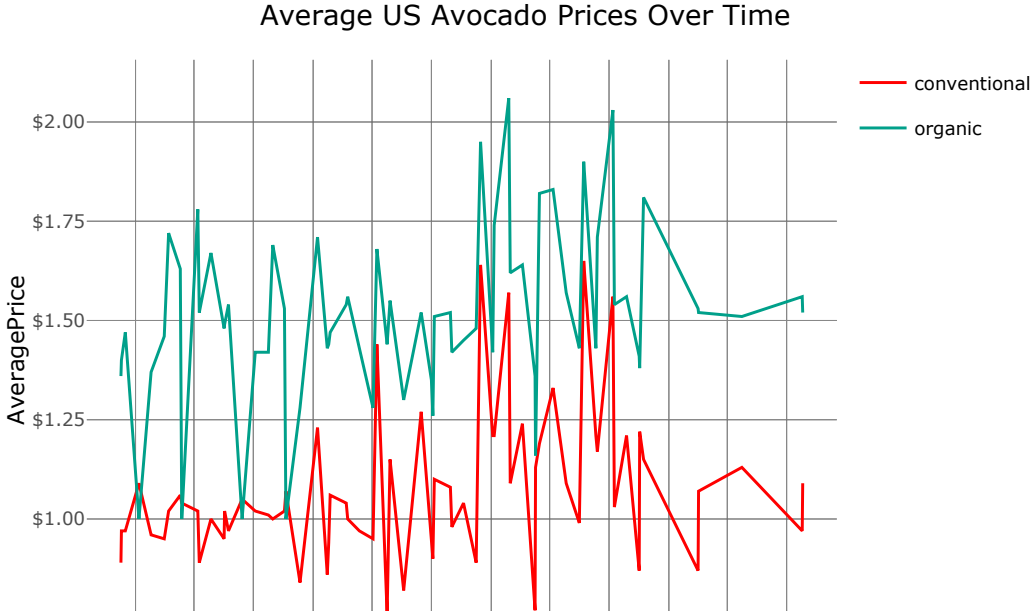


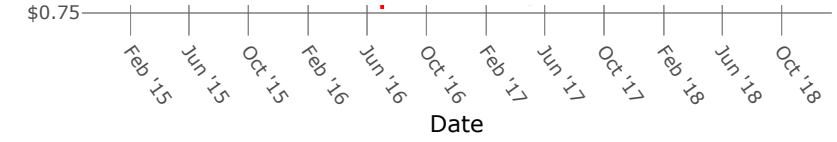



```
avocados_total
```

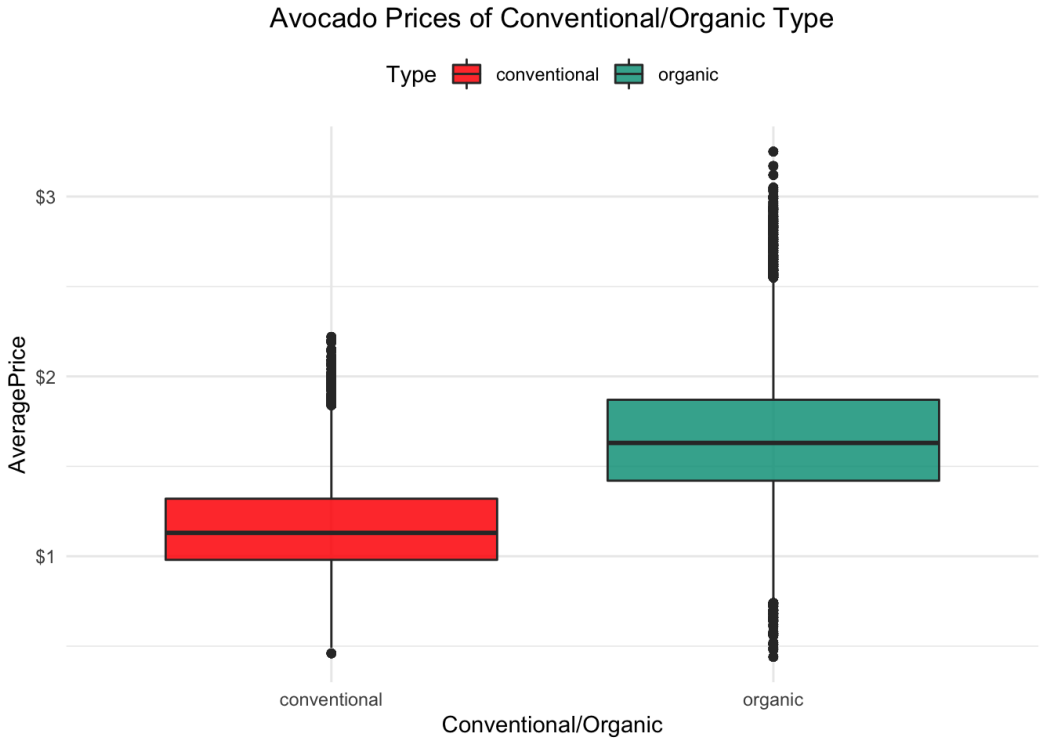
Date <chr>	AveragePrice <dbl>	Total_Volume <dbl>	Total_Bags <dbl>	type <chr>	year <dbl>	region <chr>	bag_size <chr>	bag_total <dbl>
27/12/15	0.95	27297983.7	6288852.4	conventional	2015	TotalUS	Small_Bags	4850404.09
20/12/15	0.98	25083647.2	5842743.5	conventional	2015	TotalUS	Small_Bags	4618389.66
13/12/15	0.93	28041335.4	6364279.6	conventional	2015	TotalUS	Small_Bags	4964462.13
6/12/15	0.89	28800396.6	6302263.0	conventional	2015	TotalUS	Small_Bags	5005077.36
29/11/15	0.99	22617999.4	4789009.0	conventional	2015	TotalUS	Small_Bags	3901953.04
22/11/15	0.96	25114228.1	5347835.9	conventional	2015	TotalUS	Small_Bags	4178583.45
15/11/15	0.92	28597756.3	6174346.0	conventional	2015	TotalUS	Small_Bags	4854619.04
8/11/15	0.97	28485716.0	5151513.7	conventional	2015	TotalUS	Small_Bags	4058500.97
1/11/15	0.97	31047484.3	5268565.4	conventional	2015	TotalUS	Small_Bags	3966597.25
25/10/15	1.04	26240072.1	4957516.0	conventional	2015	TotalUS	Small_Bags	3918658.43

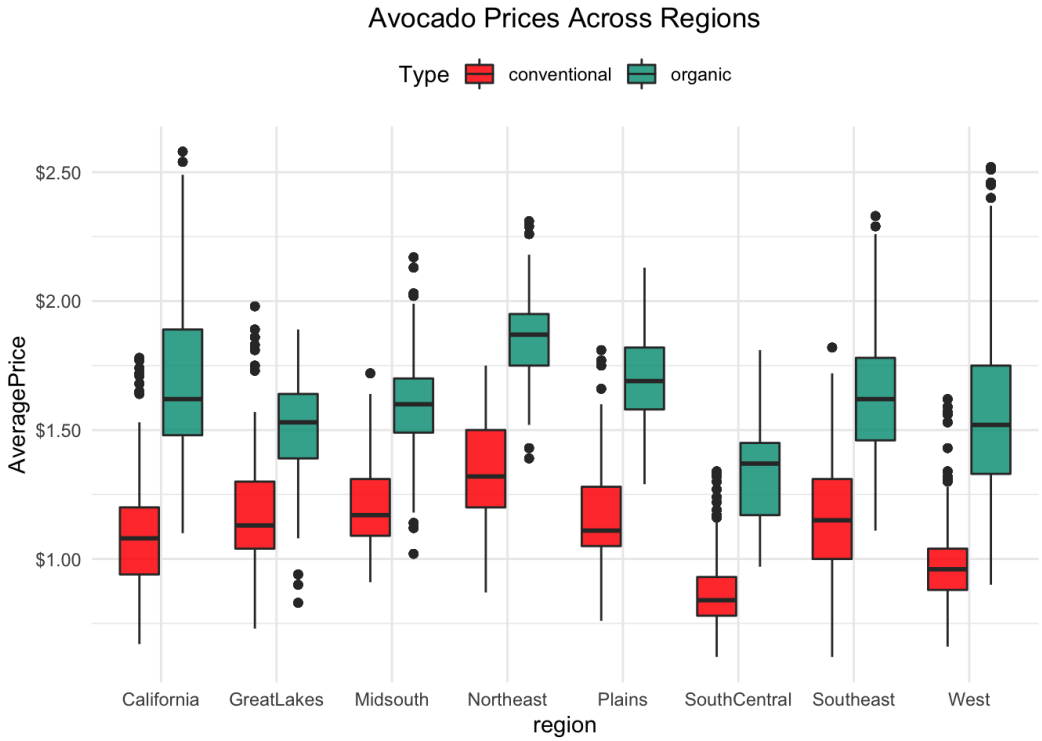
1-10 of 4,056 rows | 1-9 of 11 columns Previous **1** 2 3 4 5 6 … 406 Next



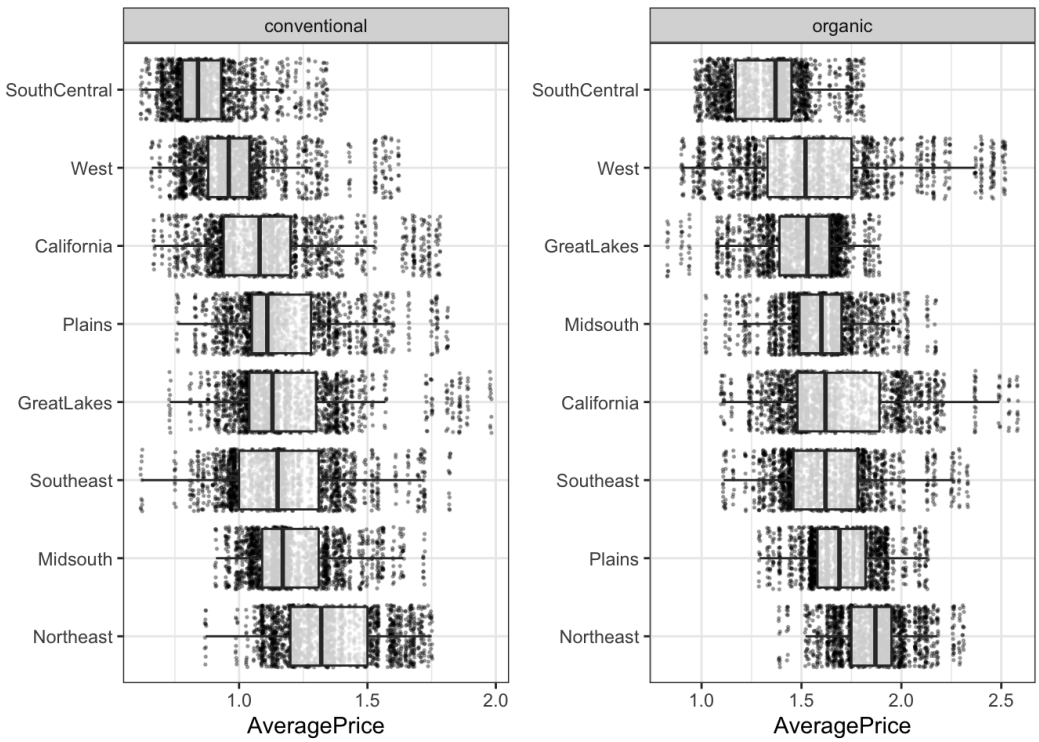



```
ggplot(avocados, aes(x = factor(avocados$type, levels = c('conventional','organic'),ordered = TRUE), y = Averag
ePrice, fill = type)) +
  geom_boxplot(alpha = .85) +
  scale_fill_manual(values=wes_palette(n=2, name="Darjeeling1")) +
  scale_color_manual(values=wes_palette(n=2, name="Darjeeling1")) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  xlab('Conventional/Organic')+
  theme(legend.position = "top",
        plot.title = element_text(hjust = .5)) +
  ggtitle("Avocado Prices of Conventional/Organic Type") +
  labs(fill = "Type")
```

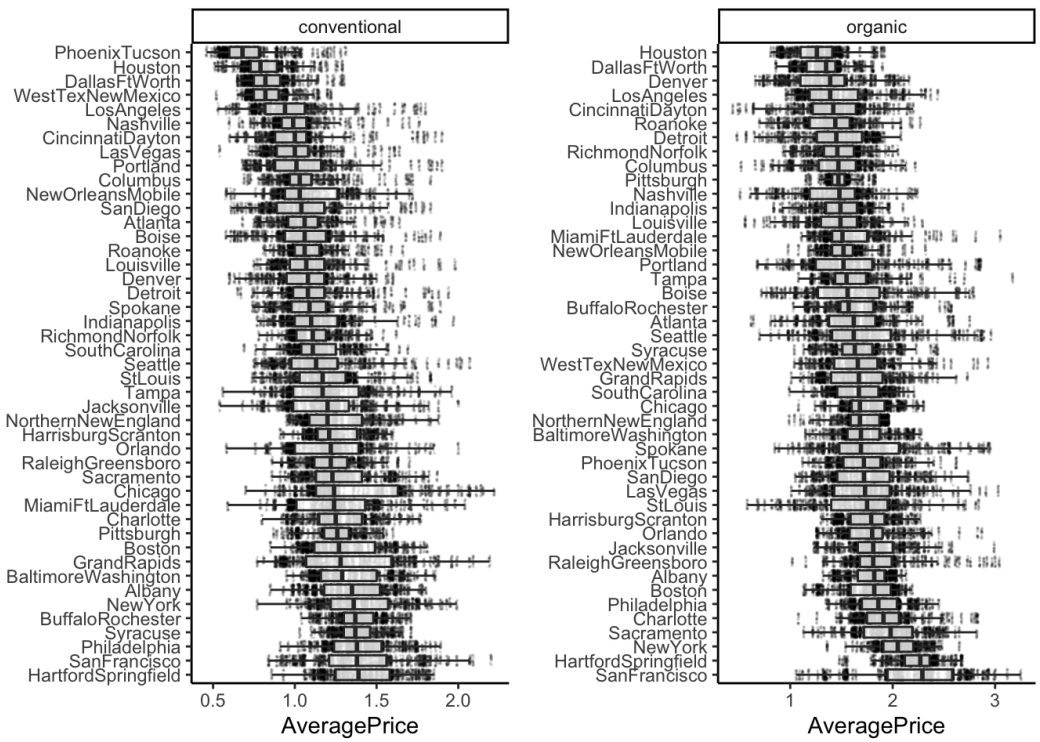




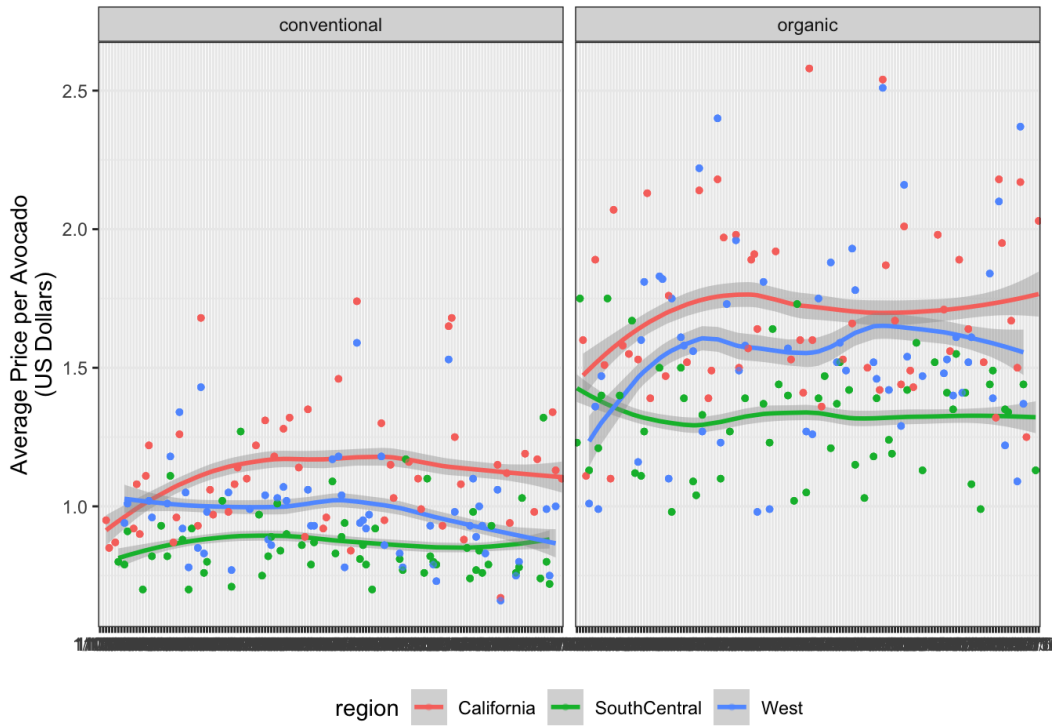
```
# get median average price for organic and conventional avocadoes
conventional <- avocados_region[avocados_region$type == "conventional",]
conventional <- aggregate(conventional$AveragePrice, by = list(conventional$region), FUN = median)
conventional <- conventional[rev(order(conventional$x)),]$Group.1
organic <- avocados_region[avocados_region$type == "organic",]
organic <- aggregate(organic$AveragePrice, by = list(organic$region), FUN = median)
organic <- organic[rev(order(organic$x)),]$Group.1

# visual for conventional avocados
avocados_region$region <- factor(avocados_region$region, levels = conventional)
p1 <- ggplot(data = avocados_region[avocados_region$type == "conventional",], aes(y = AveragePrice, x = regio
n))+
  geom_jitter(alpha = 0.3, size = 0.3) +
  geom_boxplot(alpha = 0.8, outlier.color = NA) +
  facet_grid(. ~ type) +
  coord_flip() +
  xlab("Average Price per Avocado\n(US Dollars)") +
  theme_bw() +
  theme(legend.position = "bottom", axis.title.y=element_blank())

# visual for organic avocados
avocados_region$region <- factor(avocados_region$region, levels = organic)
p2 <- ggplot(data = avocados_region[avocados_region$type == "organic",], aes(y = AveragePrice, x = region))+
  geom_jitter(alpha = 0.3, size = 0.3) +
  geom_boxplot(alpha = 0.8, outlier.color = NA) +
  facet_grid(. ~ type) +
  coord_flip() +
  xlab("Average Price per Avocado\n(US Dollars)") +
  theme_bw() +
  theme(legend.position = "bottom", axis.title.y=element_blank())

# set options for printing figure
options(repr.plot.width = 8, repr.plot.height = 12)

# arrange plots
grid.arrange(p1, p2, nrow = 1)
```



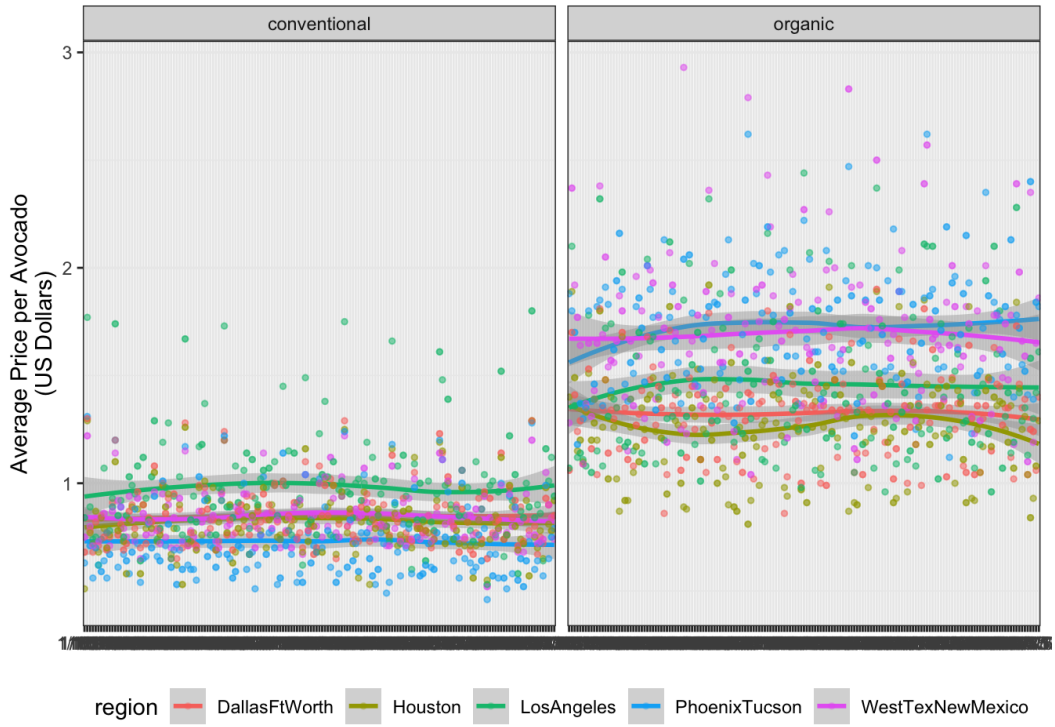
```
# get median average price for organic and conventional avocadoes
conventional <- avocados_market[avocados_market$type == "conventional",]
conventional <- aggregate(conventional$AveragePrice, by = list(conventional$region), FUN = median)
conventional <- conventional[rev(order(conventional$x)),]$Group.1
organic <- avocados_market[avocados_market$type == "organic",]
organic <- aggregate(organic$AveragePrice, by = list(organic$region), FUN = median)
organic <- organic[rev(order(organic$x)),]$Group.1

# visual for conventional avocados
avocados_market$region <- factor(avocados_market$region, levels = conventional)
p1 <- ggplot(data = avocados_market[avocados_market$type == "conventional",], aes(y = AveragePrice, x = regio
n))+
  geom_jitter(alpha = 0.1, size = 0.1) +
  geom_boxplot(alpha = 0.8, outlier.color = NA) +
  facet_grid(. ~ type) +
  coord_flip() +
  xlab("Average Price per Avocado\n(US Dollars)") +
  theme_classic() +
  theme(legend.position = "bottom", axis.title.y=element_blank())

# visual for organic avocados
avocados_market$region <- factor(avocados_market$region, levels = organic)
p2 <- ggplot(data = avocados_market[avocados_market$type == "organic",], aes(y = AveragePrice, x = region))+
  geom_jitter(alpha = 0.1, size = 0.1) +
  geom_boxplot(alpha = 0.8, outlier.color = NA) +
  facet_grid(. ~ type) +
  coord_flip() +
  xlab("Average Price per Avocado\n(US Dollars)") +
  theme_classic() +
  theme(legend.position = "bottom", axis.title.y=element_blank())

# arrange plots
grid.arrange(p1, p2, nrow = 1)
```



```
# set options for printing figure
options(repr.plot.width = 9, repr.plot.height = 4)

# visualize trends in California, New York, and South Carolina
ggplot(
  data = avocados[avocados$region == c("SouthCentral", "West", "California"),],
  aes(x = Date, y = AveragePrice, color = region, group = region)
) +
  geom_smooth(method = "loess") +
  geom_point(size = 1, alpha = 0.3) +
  facet_grid(. ~ type) +
  ylab("Average Price per Avocado\n(US Dollars)") +
  theme_bw() +
  theme(axis.title.x = element_blank(), legend.position = "bottom")
```




```
# set options for printing figure
options(repr.plot.width = 9, repr.plot.height = 4)

# visualize trends in California, New York, and South Carolina
ggplot(
  data = avocados[avocados$region == c("Houston", "DallasFtWorth", "WestTexNewMexico", "LosAngeles", "PhoenixTu
cson"),],
  aes(x = Date, y = AveragePrice, color = region, group = region)
) +
  geom_smooth(method = "loess") +
  geom_point(size = 1, alpha = 0.3) +
  facet_grid(. ~ type) +
  ylab("Average Price per Avocado\n(US Dollars)") +
  theme_bw() +
  theme(axis.title.x = element_blank(), legend.position = "bottom")
```




```
# only interested in small hass for avocado spread
small_hass_volume = avocados$avocado_volume[avocados$avocado_type=="small_hass"]
avocados_orginal = read_csv("avocado.csv")
```

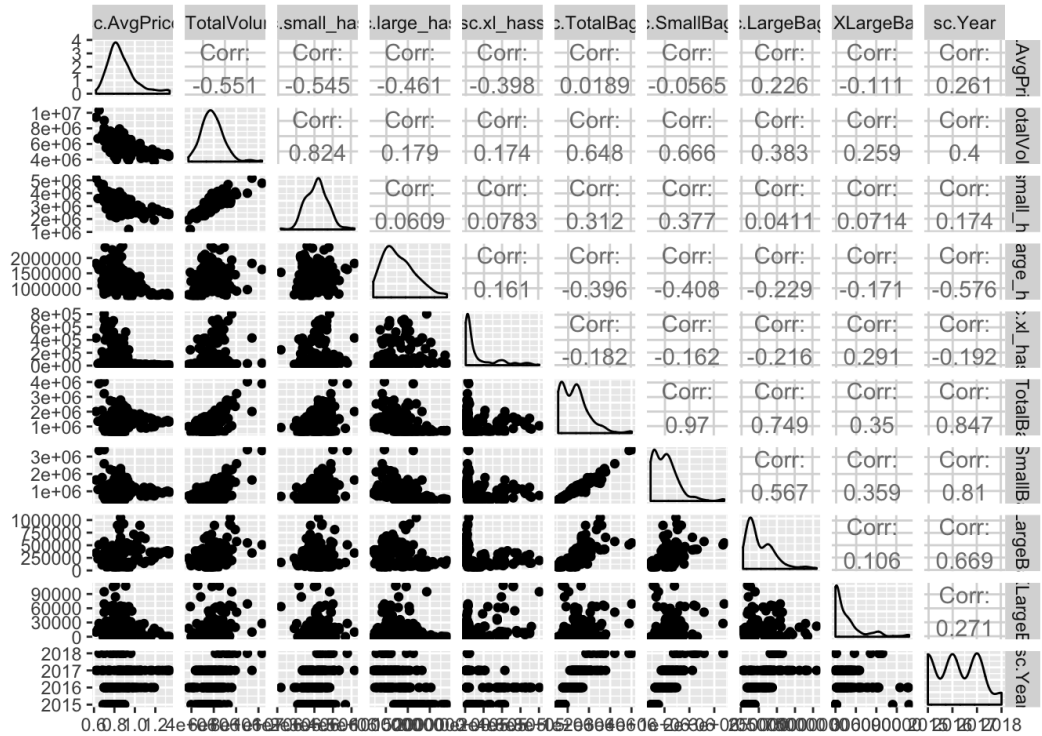
```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   AveragePrice = col_double(),
##   `Total Volume` = col_double(),
##   `4046` = col_double(),
##   `4225` = col_double(),
##   `4770` = col_double(),
##   `Total Bags` = col_double(),
##   `Small Bags` = col_double(),
##   `Large Bags` = col_double(),
##   `XLarge Bags` = col_double(),
##   type = col_character(),
##   year = col_double(),
##   region = col_character()
## )
```

```
# SouthCentral conventional data
avocado_conven = avocados_orginal[avocados_orginal$type=="conventional",]
conven.sc = avocado_conven[avocado_conven$region=="SouthCentral",]
sc.AvgPrice = conven.sc$AveragePrice
sc.TotalVolume = conven.sc$`Total Volume`
sc.small_hass = conven.sc$`4046`
sc.large_hass = conven.sc$`4225`
sc.xl_hass = conven.sc$`4770`
sc.TotalBags = conven.sc$`Total Bags`
sc.SmallBags = conven.sc$`Small Bags`
sc.LargeBags = conven.sc$`Large Bags`
sc.XLargeBags = conven.sc$`XLarge Bags`
sc.Year = conven.sc$year
conven.sc.df = data_frame(sc.AvgPrice , sc.TotalVolume , sc.small_hass, sc.large_hass, sc.xl_hass, sc.TotalBag
s, sc.SmallBags, sc.LargeBags, sc.XLargeBags, sc.Year )
```

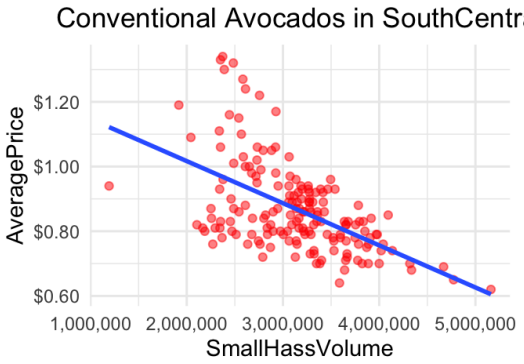
```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

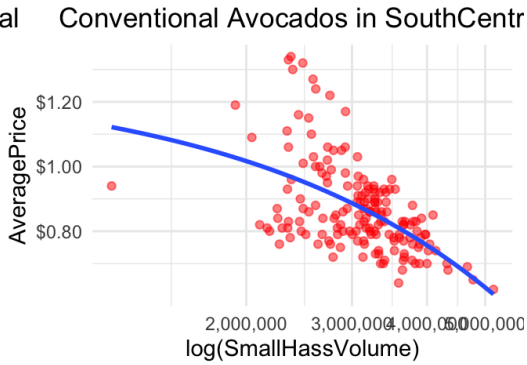
```
GGally::ggpairs(conven.sc.df, progress = F)
```

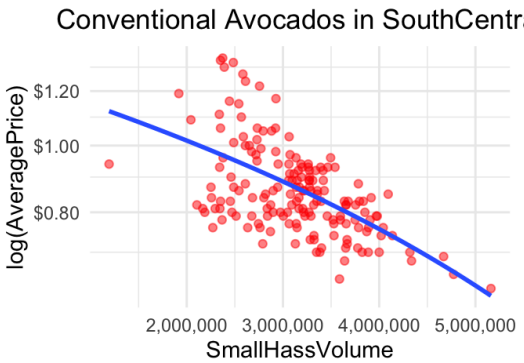
```
## Warning: replacing previous import 'ggplot2::empty' by 'plyr::empty' when
## loading 'GGally'
```

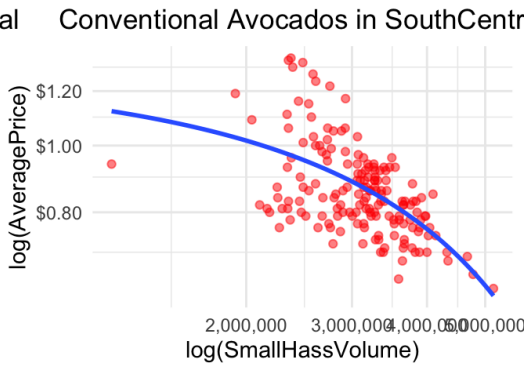


```
#small hass
#avocado volume + 1 then do the log transformation as it could be 0
p1 <- conven.sc %>%
  ggplot(aes(x = `4046`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + ylab("AveragePrice")
 +  xlab("SmallHassVolume")

p2 <- conven.sc %>%
  ggplot(aes(x = `4046`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log")
 + ylab("AveragePrice") +  xlab("log(SmallHassVolume)")

p3 <- conven.sc %>%
  ggplot(aes(x = `4046`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(y="log")
 + ylab("log(AveragePrice)") +  xlab("SmallHassVolume")


p4 <- conven.sc %>%
  ggplot(aes(x = `4046`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log",
 y="log") + ylab("log(AveragePrice)") +  xlab("log(SmallHassVolume)")

cowplot::plot_grid(p1, p2, p3, p4)
```

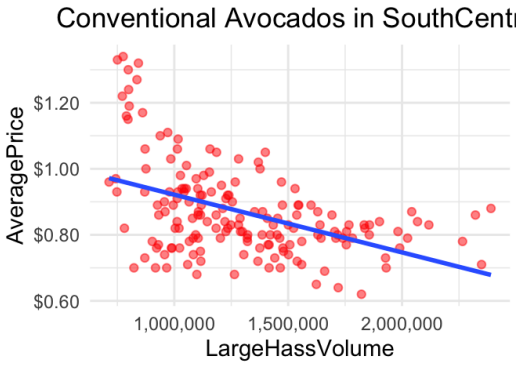

Conventional Avocados in SouthCentral
AveragePrice
$1.20
$1.00
$0.80
$0.60
1,000,000
2,000,000
3,000,000
4,000,000
5,000,000
SmallHassVolume

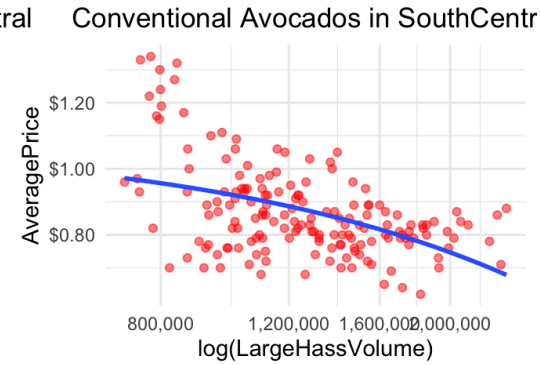

Conventional Avocados in SouthCentr
AveragePrice
$1.20
$1.00
$0.80
2,000,000
3,000,000
4,000,000
5,000,000
log(SmallHassVolume)

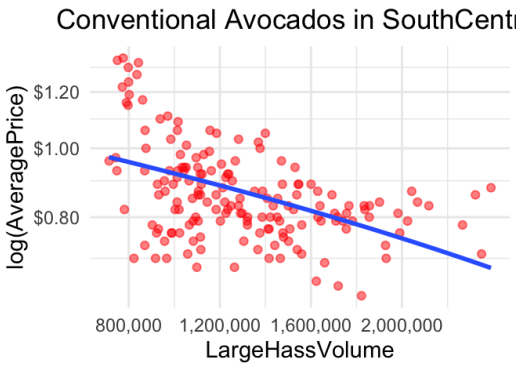

Conventional Avocados in SouthCentral
log(AveragePrice)
$1.20
$1.00
$0.80
2,000,000
3,000,000
4,000,000
5,000,000
SmallHassVolume

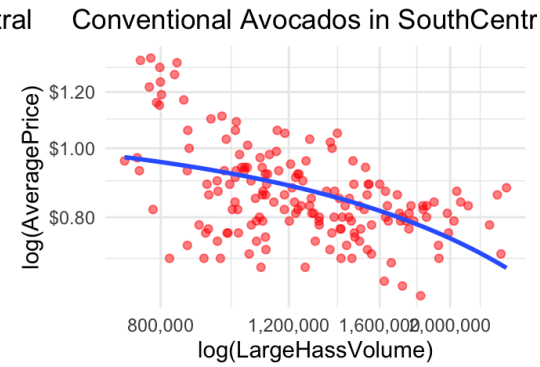

Conventional Avocados in SouthCentr
log(AveragePrice)
$1.20
$1.00
$0.80
2,000,000
3,000,000
4,000,000
5,000,000
log(SmallHassVolume)

```
#large hass
#avocado volume + 1 then do the log transformation as it could be 0
p1 <- conven.sc %>%
  ggplot(aes(x = `sc.large_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + ylab("AveragePrice")
 +  xlab("LargeHassVolume")

p2 <- conven.sc %>%
  ggplot(aes(x = `sc.large_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log")
 + ylab("AveragePrice") +  xlab("log(LargeHassVolume)")

p3 <- conven.sc %>%
  ggplot(aes(x = `sc.large_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(y="log")
 + ylab("log(AveragePrice)") +  xlab("LargeHassVolume")


p4 <- conven.sc %>%
  ggplot(aes(x = `sc.large_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F)  + coord_trans(y="log",
x="log") + ylab("log(AveragePrice)") + xlab("log(LargeHassVolume)")

cowplot::plot_grid(p1, p2, p3, p4)
```

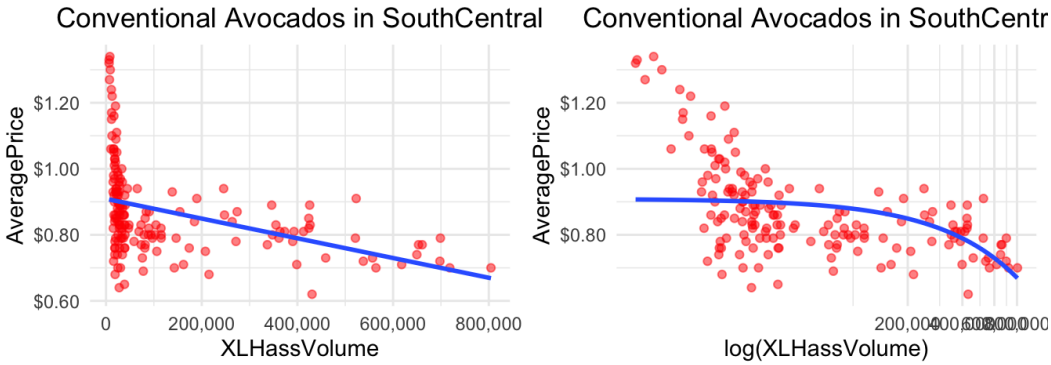

Conventional Avocados in SouthCentral
AveragePrice
$1.20
$1.00
$0.80
$0.60
1,000,000
1,500,000
2,000,000
LargeHassVolume

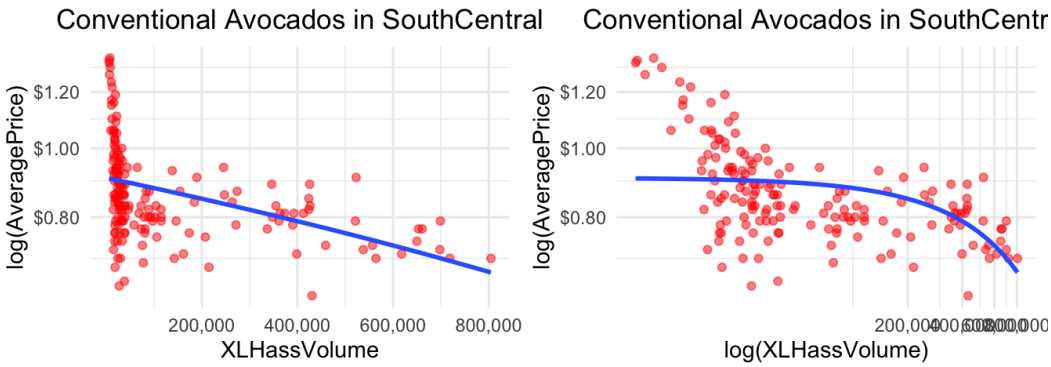

Conventional Avocados in SouthCentr
AveragePrice
$1.20
$1.00
$0.80
800,000
1,200,000
1,600,000
2,000,000
log(LargeHassVolume)

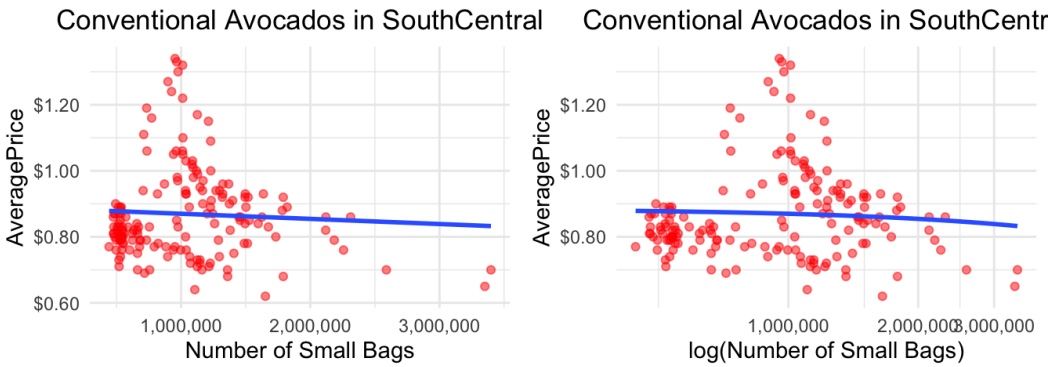

Conventional Avocados in SouthCentral
log(AveragePrice)
$1.20
$1.00
$0.80
800,000
1,200,000
1,600,000
2,000,000
LargeHassVolume

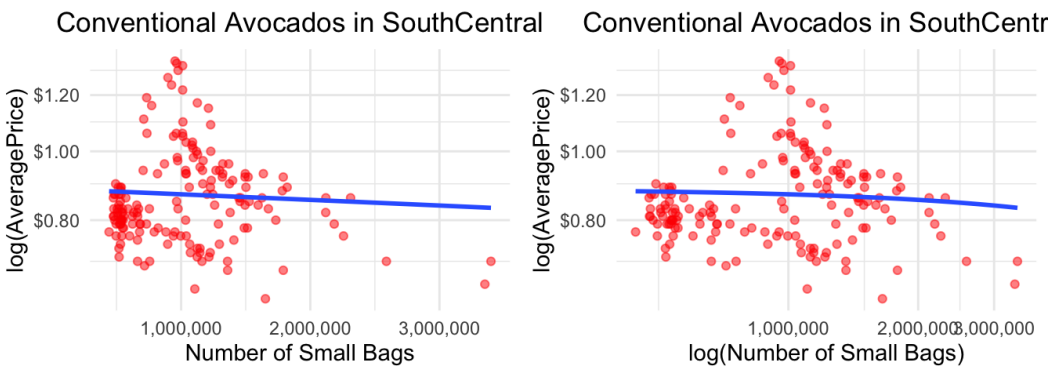

Conventional Avocados in SouthCentr
log(AveragePrice)
$1.20
$1.00
$0.80
800,000
1,200,000
1,600,000
2,000,000
log(LargeHassVolume)

```
#xl hass
#avocado volume + 1 then do the log transformation as it could be 0
p1 <- conven.sc %>%
  ggplot(aes(x = `sc.xl_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + ylab("AveragePrice")
 +  xlab("XLHassVolume")

p2 <- conven.sc %>%
  ggplot(aes(x = `sc.xl_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log")
 + ylab("AveragePrice") +  xlab("log(XLHassVolume)")

p3 <- conven.sc %>%
  ggplot(aes(x = `sc.xl_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(y="log")
 + ylab("log(AveragePrice)") +  xlab("XLHassVolume")


p4 <- conven.sc %>%
  ggplot(aes(x = `sc.xl_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Conventional Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log",
 y ="log") + ylab("log(AveragePrice)") +  xlab("log(XLHassVolume)")

cowplot::plot_grid(p1, p2, p3, p4)
```










```
M0 = lm(sc.AvgPrice ~ 1, data=conven.sc.df)
Mf = lm(sc.AvgPrice ~ sc.small_hass + log(sc.large_hass) + log(sc.xl_hass) + log(sc.SmallBags) + sc.LargeBags +
sc.XLargeBags + sc.Year, data=conven.sc.df)
#MforwardAIC = step(Mf, direction="backward", trace=0, k=2, scope= list(lower=M0, upper=Mf))
drop1(Mf, test="F")
```

	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	1.099557	-834.9135	*NA*	*NA*
sc.small_hass	1	0.3519864612	1.451544	-789.9788	51.53875445	2.444872e-11
log(sc.large_hass)	1	0.2084308637	1.307988	-807.5780	30.51897810	1.303247e-07
log(sc.xl_hass)	1	0.2936035131	1.393161	-796.9167	42.99017444	7.119405e-10
log(sc.SmallBags)	1	0.0911152819	1.190673	-823.4593	13.34133171	3.507494e-04
sc.LargeBags	1	0.0097353566	1.109293	-835.4237	1.42547571	2.342604e-01

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
sc.XLargeBags	1	0.0001683499	1.099726	-836.8876	0.02465022	8.754385e-01
sc.Year	1	0.0420935153	1.141651	-830.5645	6.16343976	1.406716e-02

8 rows

```
MstepFtest = update(Mf, . ~ . - sc.XLargeBags)
drop1(MstepFtest, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.099726	-836.8876	*NA*	*NA*
sc.small_hass	1	0.36526867	1.464994	-790.4200	53.807527	1.004167e-11
log(sc.large_hass)	1	0.20831914	1.308045	-809.5707	30.687378	1.202398e-07
log(sc.xl_hass)	1	0.32668794	1.426414	-794.9303	48.124220	9.082737e-11
log(sc.SmallBags)	1	0.09811338	1.197839	-824.4451	14.453028	2.031309e-04
sc.LargeBags	1	0.00958903	1.109315	-837.4204	1.412555	2.363721e-01
sc.Year	1	0.04203558	1.141761	-832.5482	6.192238	1.384221e-02

7 rows

```
MstepFtest <- update(MstepFtest, . ~ . - sc.LargeBags)
add1(MstepFtest, test="F", scope=Mf)
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.109315	-837.4204	*NA*	*NA*
sc.LargeBags	1	9.589030e-03	1.099726	-836.8876	1.412554758	0.2363721
sc.XLargeBags	1	2.202345e-05	1.109293	-835.4237	0.003216283	0.9548443

3 rows

```
drop1(MstepFtest, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.109315	-837.4204	*NA*	*NA*
sc.small_hass	1	0.40242343	1.511738	-787.1119	59.13111	1.318933e-12
log(sc.large_hass)	1	0.20451744	1.313832	-810.8246	30.05129	1.573180e-07
log(sc.xl_hass)	1	0.37673034	1.486045	-790.0089	55.35583	5.458884e-12
log(sc.SmallBags)	1	0.09128268	1.200597	-826.0564	13.41285	3.374249e-04
sc.Year	1	0.07017961	1.179494	-829.0534	10.31202	1.592413e-03

6 rows

```
summary(MstepFtest)
```

```
##
## Call:
## lm(formula = sc.AvgPrice ~ sc.small_hass + log(sc.large_hass) +
##     log(sc.xl_hass) + log(sc.SmallBags) + sc.Year, data = conven.sc.df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.233442 -0.046180  0.008175  0.048412  0.202650
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -9.144e+01  2.996e+01  -3.052 0.002655 **
## sc.small_hass      -9.538e-08  1.240e-08  -7.690 1.32e-12 ***
## log(sc.large_hass) -1.624e-01  2.963e-02  -5.482 1.57e-07 ***
## log(sc.xl_hass)    -4.147e-02  5.574e-03  -7.440 5.46e-12 ***
## log(sc.SmallBags)  -1.121e-01  3.060e-02  -3.662 0.000337 ***
## sc.Year             4.805e-02  1.496e-02   3.211 0.001592 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0825 on 163 degrees of freedom
## Multiple R-squared:  0.6626, Adjusted R-squared:  0.6522
## F-statistic: 64.01 on 5 and 163 DF,  p-value: < 2.2e-16
```

```
library(lubridate)
```

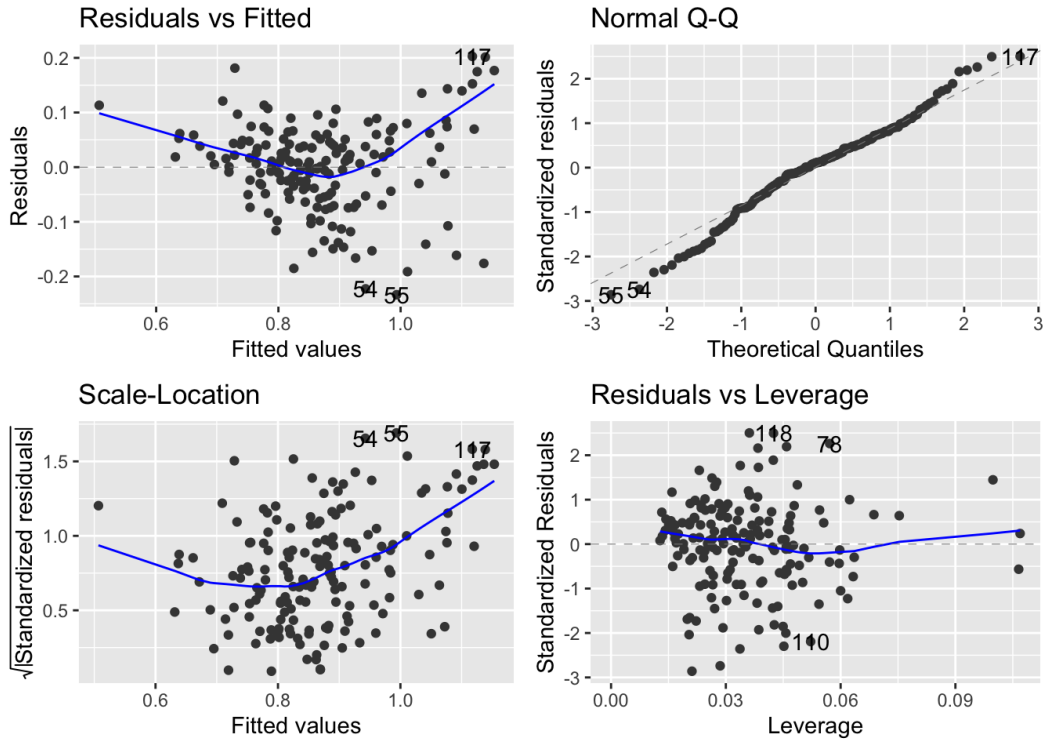
```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
# sc.AvgPrice ~ sc.small_hass + log(sc.large_hass) + log(sc.xl_hass) + log(sc.SmallBags) + sc.Year
conven.sc$Date <- as.Date(conven.sc$Date, format = "%d/%m/%y" )
conven.sc <- conven.sc %>% mutate(month = paste0(month(conven.sc$Date)))
conven.sc_monthly = conven.sc %>% group_by(month)

july_smallhass = mean(conven.sc_monthly$`4046`[conven.sc_monthly$month==7])
july_largehass = mean(conven.sc_monthly$`4225`[conven.sc_monthly$month==7])
july_xlhass = mean(conven.sc_monthly$`4770`[conven.sc_monthly$month==7])
july_smallbags = mean(conven.sc_monthly$`Small Bags`[conven.sc_monthly$month==7])


conventional_predict_sc = MstepFtest$coefficients[1] + MstepFtest$coefficients[2]*july_smallhass + MstepFtest$c
oefficients[3]*log(july_largehass) + MstepFtest$coefficients[4]*log(july_xlhass) + MstepFtest$coefficients[5]*l
og(july_smallbags) + MstepFtest$coefficients[6]*2019
conventional_predict_sc
```

```
## (Intercept)
##   0.9686191
```

```
library(ggfortify)
autoplot(MstepFtest)
```



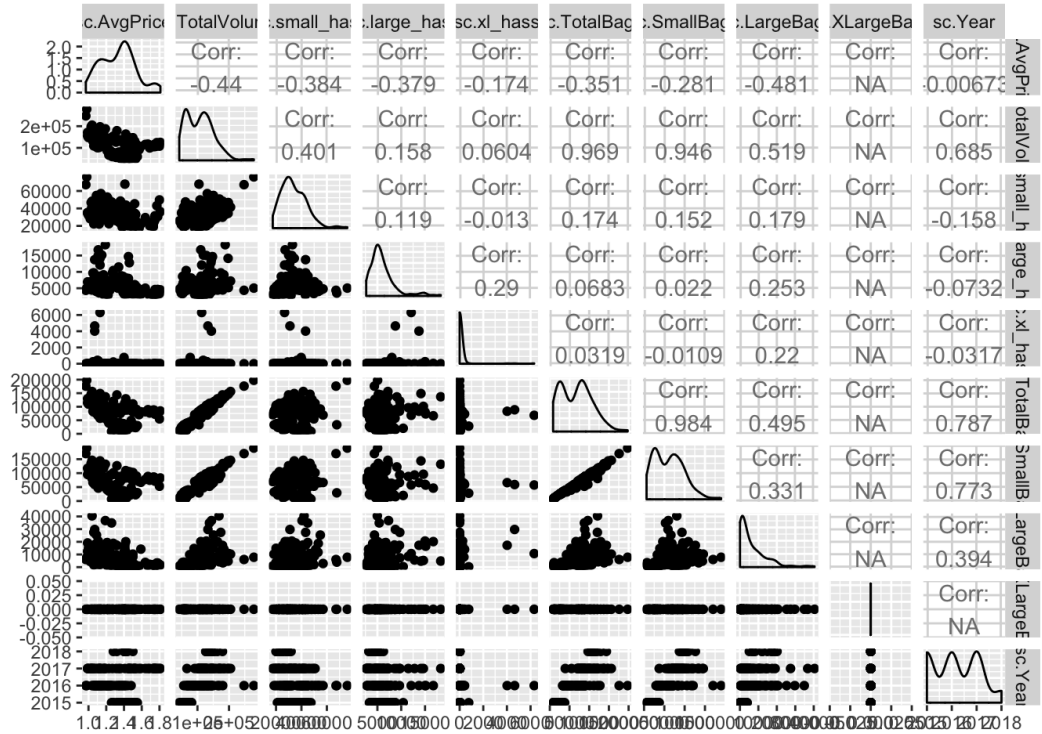

```
deviance (MstepFtest)
```

```
## [1] 1.109315
```

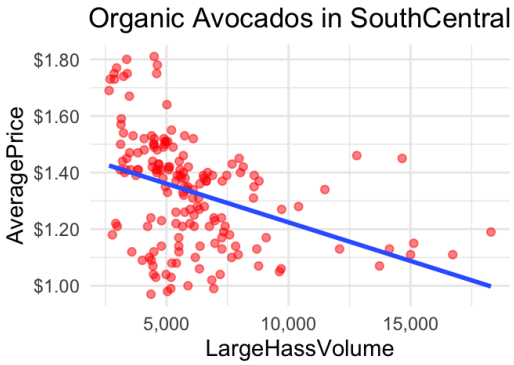
```
sort(lm.influence(MstepFtest)$hat)
```

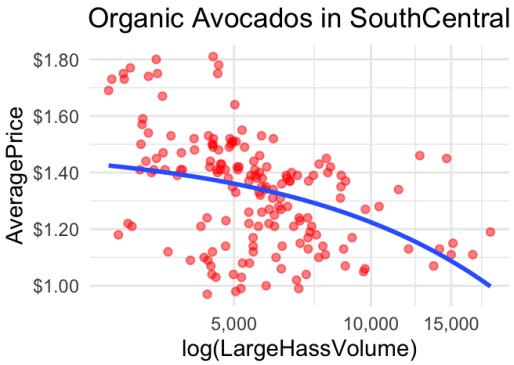
```
##         137         143         145         138          73         146
## 0.01279743 0.01326311 0.01341809 0.01419187 0.01434859 0.01507667
##          74         134         136          46          72          44
## 0.01554002 0.01565699 0.01591962 0.01611326 0.01623191 0.01667318
##         132          45          51         113         135          42
## 0.01667854 0.01705851 0.01741908 0.01758262 0.01861238 0.01889531
##         112           7          56          21         156         106
## 0.01904314 0.01991822 0.02039461 0.02050333 0.02055888 0.02063873
##          82          40          55          50          43          28
## 0.02067915 0.02069239 0.02114077 0.02124553 0.02147786 0.02175895
##          83          75         107         123          19           8
## 0.02187138 0.02196416 0.02222916 0.02306487 0.02309408 0.02317737
##         144          20         127         128          25          18
## 0.02326068 0.02339713 0.02355526 0.02357961 0.02418928 0.02447722
##         105         126          59         148          27          47
## 0.02451639 0.02456696 0.02465355 0.02522521 0.02565185 0.02585256
##         114          52         142          39          10         109
## 0.02610837 0.02623911 0.02629298 0.02637371 0.02652218 0.02658425
##          24          49          66         108          64          80
## 0.02661081 0.02672136 0.02699611 0.02706028 0.02711338 0.02759648
##          71         115         131          54         129         124
## 0.02765880 0.02822475 0.02850517 0.02851738 0.02897723 0.02903153
##           6          11          30         133         150           3
## 0.02925323 0.02941442 0.02977527 0.02977824 0.03016627 0.03033369
##          63          81         160          70          65          58
## 0.03042882 0.03052388 0.03054330 0.03111277 0.03161408 0.03168327
##          93         167          85          41         141         151
## 0.03177656 0.03187996 0.03193357 0.03209343 0.03273967 0.03278152
##          92          76           5          23         121         161
## 0.03303846 0.03351304 0.03368395 0.03373811 0.03377436 0.03403312
##          84          87         162          22          88          26
## 0.03452177 0.03469801 0.03475252 0.03521268 0.03590750 0.03591554
##         117         125          77          36         147          38
## 0.03614659 0.03626386 0.03627796 0.03638577 0.03643680 0.03684135
##           9         139         111          32          60         116
## 0.03709216 0.03757986 0.03816187 0.03827731 0.03829951 0.03839489
##         122         164          91         140          37         166
## 0.03850979 0.03861217 0.03864397 0.03874243 0.03885195 0.03914907
##         159          12          35          86         155          14
## 0.03959337 0.04117615 0.04120574 0.04196203 0.04222045 0.04224073
##         120         118           4         100          53          15
## 0.04240187 0.04246755 0.04266051 0.04269372 0.04357218 0.04403658
##          13         154         157          97         102          57
## 0.04469356 0.04495977 0.04516619 0.04532920 0.04549331 0.04568433
##          89          96         130         119          98         103
## 0.04569677 0.04581120 0.04581987 0.04587231 0.04596312 0.04607814
##          48          94          34           1          17           2
## 0.04652227 0.04654425 0.04663573 0.04688531 0.04693737 0.04735649
##          67          61          79          68         169          69
## 0.04764667 0.04785652 0.04868471 0.04918928 0.05007226 0.05041697
##          95         110         163          99          16          31
## 0.05167646 0.05216285 0.05434078 0.05449693 0.05557856 0.05709209
##          78          33         104         149         158          62
## 0.05714843 0.05962634 0.05977258 0.05999727 0.06184099 0.06234223
##         153          90          29         168         152         101
## 0.06328260 0.06355478 0.06869048 0.07531721 0.09980812 0.10654032
##         165
## 0.10690683
```

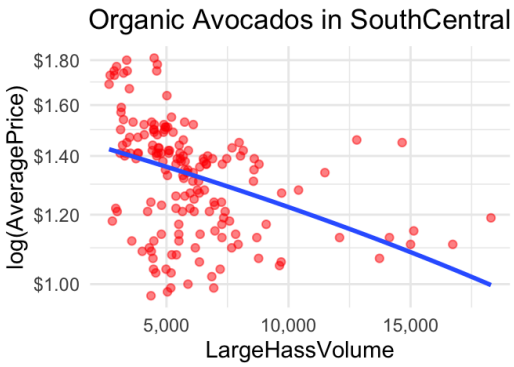
```
# SouthCentral organic data
avocado_organic = avocados_orginal[avocados_orginal$type=="organic",]
organic.sc = avocado_organic[avocado_organic$region=="SouthCentral",]
sc.AvgPrice = organic.sc $AveragePrice

sc.TotalVolume = organic.sc $`Total Volume`
sc.small_hass = organic.sc$`4046`
sc.large_hass = organic.sc$`4225`
sc.xl_hass = organic.sc$`4770`
sc.TotalBags = organic.sc $`Total Bags`
sc.SmallBags = organic.sc $`Small Bags`
sc.LargeBags = organic.sc $`Large Bags`
sc.XLargeBags = organic.sc $`XLarge Bags`
sc.Year = organic.sc $year
organic.sc.df = data_frame(sc.AvgPrice , sc.TotalVolume , sc.small_hass, sc.large_hass, sc.xl_hass, sc.TotalBag
s, sc.SmallBags, sc.LargeBags, sc.XLargeBags, sc.Year )
GGally::ggpairs(organic.sc.df, progress = F)
```

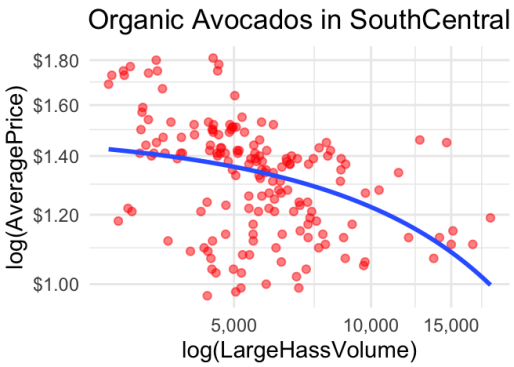


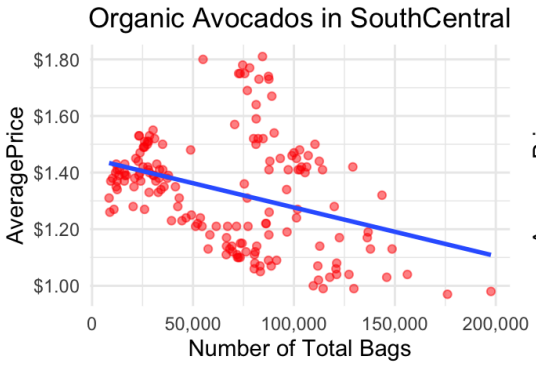
```
#large hass
#avocado volume + 1 then do the log transformation as it could be 0
p1 <- organic.sc %>%
  ggplot(aes(x = `sc.large_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Organic Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + ylab("AveragePrice") +  xl
ab("LargeHassVolume")

p2 <- organic.sc %>%
  ggplot(aes(x = `sc.large_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Organic Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log") + yla
b("AveragePrice") +  xlab("log(LargeHassVolume)")

p3 <- organic.sc %>%
  ggplot(aes(x = `sc.large_hass`, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Organic Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(y="log") + yla
b("log(AveragePrice)") +  xlab("LargeHassVolume")


p4 <- organic.sc %>%
  ggplot(aes(x = `sc.large_hass`+1, y = AveragePrice)) +
  geom_point(alpha = .5, col = "#FF0000" ) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        plot.subtitle = element_text(hjust = .5)) +
  ggtitle("Organic Avocados in SouthCentral") + geom_smooth(method = "lm", se = F) + coord_trans(x="log", y="lo
g") + ylab("log(AveragePrice)") +  xlab("log(LargeHassVolume)")

cowplot::plot_grid(p1, p2, p3, p4)
```

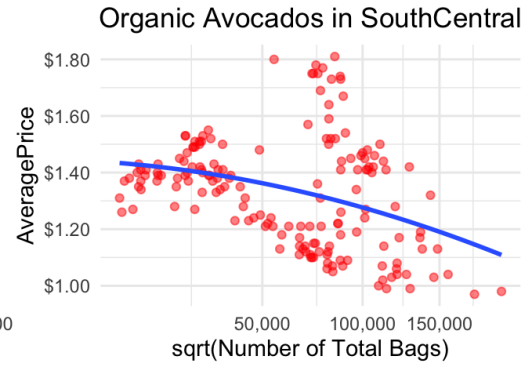

Organic Avocados in SouthCentral
AveragePrice
$1.80
$1.60
$1.40
$1.20
$1.00
5,000
10,000
15,000
LargeHassVolume

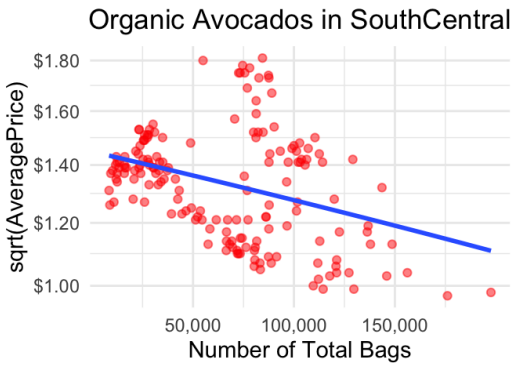

Organic Avocados in SouthCentral
AveragePrice
$1.80
$1.60
$1.40
$1.20
$1.00
5,000
10,000
15,000
log(LargeHassVolume)

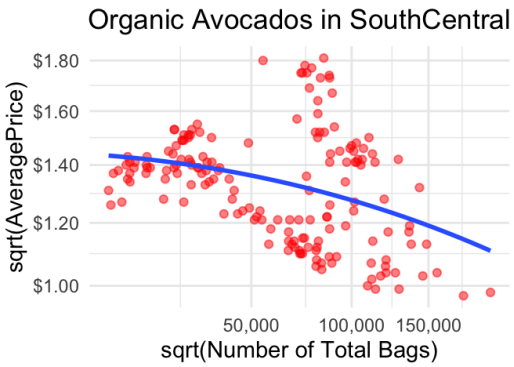

Organic Avocados in SouthCentral
log(AveragePrice)
$1.80
$1.60
$1.40
$1.20
$1.00
5,000
10,000
15,000
LargeHassVolume

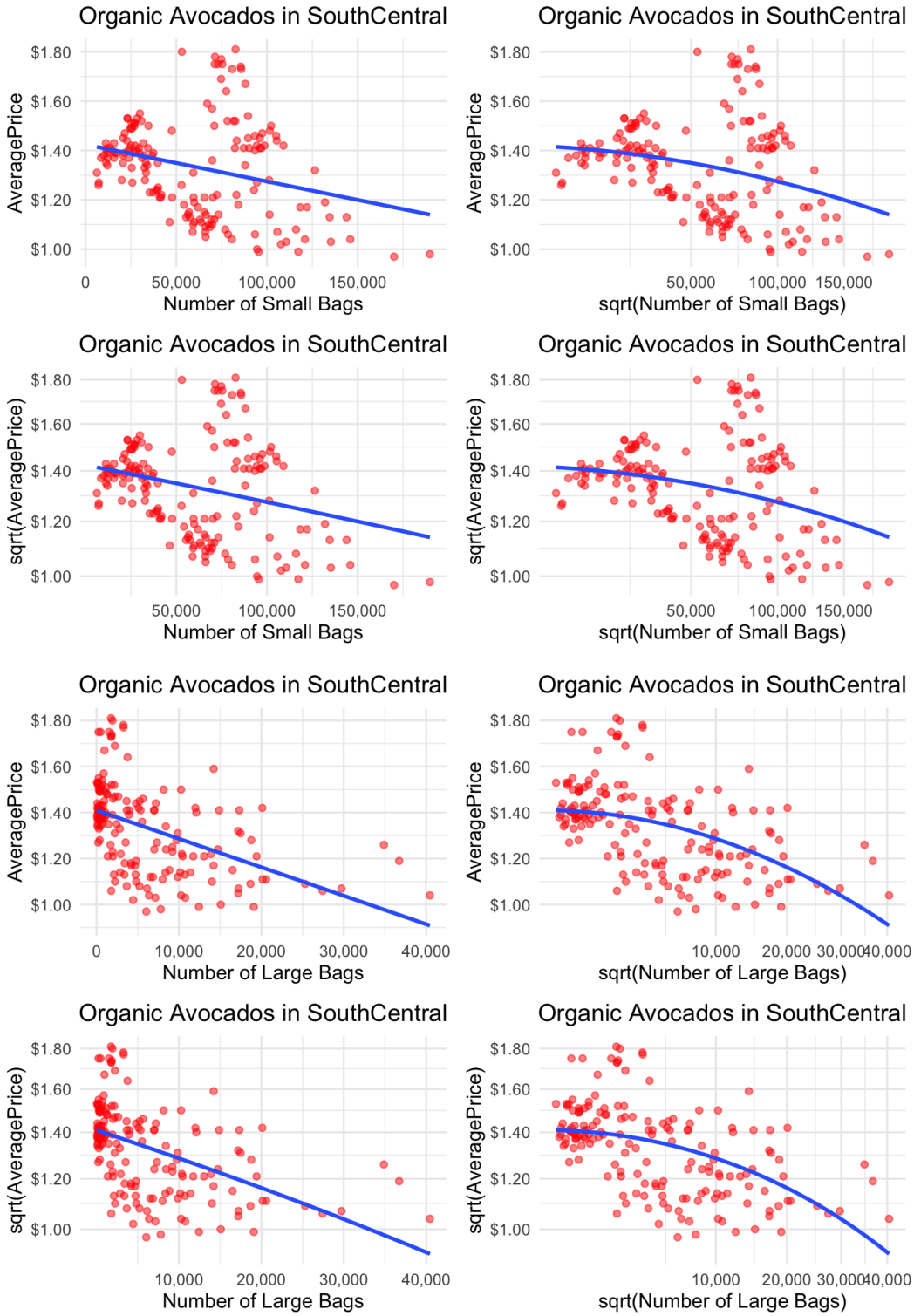

Organic Avocados in SouthCentral
log(AveragePrice)
$1.80
$1.60
$1.40
$1.20
$1.00
5,000
10,000
15,000
log(LargeHassVolume)


Organic Avocados in SouthCentral
AveragePrice
$1.80
$1.60
$1.40
$1.20
$1.00
0
50,000
100,000
150,000
200,000
Number of Total Bags


Organic Avocados in SouthCentral
AveragePrice
$1.80
$1.60
$1.40
$1.20
$1.00
50,000
100,000
150,000
sqrt(Number of Total Bags)


Organic Avocados in SouthCentral
sqrt(AveragePrice)
$1.80
$1.60
$1.40
$1.20
$1.00
50,000
100,000
150,000
Number of Total Bags


Organic Avocados in SouthCentral
sqrt(AveragePrice)
$1.80
$1.60
$1.40
$1.20
$1.00
50,000
100,000
150,000
sqrt(Number of Total Bags)




```
# model for organic + southCentral area
M0 = lm(sc.AvgPrice ~ 1, data=organic.sc.df)
Mf = lm(sc.AvgPrice ~ log(sc.small_hass+1) + log(sc.large_hass+1) + log(sc.xl_hass+1) + sc.TotalBags + sc.Small
Bags + sqrt(sc.LargeBags) + sc.Year, data=organic.sc.df)
drop1(Mf, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	2.643904	-686.6415	*NA*	*NA*
log(sc.small_hass + 1)	1	0.017500064	2.661404	-687.5266	1.0656628	3.034754e-01
log(sc.large_hass + 1)	1	0.234280855	2.878185	-674.2929	14.2664849	2.229162e-04
log(sc.xl_hass + 1)	1	0.008112226	2.652016	-688.1238	0.4939923	4.831668e-01
sc.TotalBags	1	0.118188704	2.762093	-681.2508	7.1970770	8.064473e-03
sc.SmallBags	1	0.173314650	2.817219	-677.9111	10.5539603	1.411337e-03

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
sqrt(sc.LargeBags)	1	0.420615523	3.064520	-663.6913	25.6132966	1.127987e-06
sc.Year	1	0.732768198	3.376672	-647.2984	44.6217702	3.697406e-10

8 rows

```
MstepFtest = update(Mf, . ~ . - log(sc.xl_hass + 1))
drop1(MstepFtest, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	2.652016	-688.1238	*NA*	*NA*
log(sc.small_hass + 1)	1	0.01567445	2.667691	-689.1279	0.9574832	3.292808e-01
log(sc.large_hass + 1)	1	0.27998147	2.931998	-673.1623	17.1028352	5.671645e-05
sc.TotalBags	1	0.12162495	2.773641	-682.5457	7.4295327	7.122147e-03
sc.SmallBags	1	0.17692259	2.828939	-679.2095	10.8074218	1.239759e-03
sqrt(sc.LargeBags)	1	0.44755248	3.099569	-663.7694	27.3390108	5.206163e-07
sc.Year	1	0.75122779	3.403244	-647.9737	45.8891981	2.197889e-10

7 rows

```
MstepFtest <- update(MstepFtest, . ~ . -log(sc.small_hass + 1))
add1(MstepFtest, test="F", scope=Mf)
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	2.667691	-689.1279	*NA*	*NA*
log(sc.small_hass + 1)	1	0.015674450	2.652016	-688.1238	0.9574832	0.3292808
log(sc.xl_hass + 1)	1	0.006286613	2.661404	-687.5266	0.3826669	0.5370482

3 rows

```
drop1(MstepFtest, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	2.667691	-689.1279	*NA*	*NA*
log(sc.large_hass + 1)	1	0.3067357	2.974426	-672.7342	18.742019	2.604340e-05
sc.TotalBags	1	0.1214594	2.789150	-683.6033	7.421359	7.148681e-03
sc.SmallBags	1	0.1807647	2.848455	-680.0476	11.045001	1.098136e-03
sqrt(sc.LargeBags)	1	0.4622013	3.129892	-664.1241	28.241206	3.474302e-07
sc.Year	1	1.0531421	3.720833	-634.8957	64.348601	1.930604e-13

6 rows

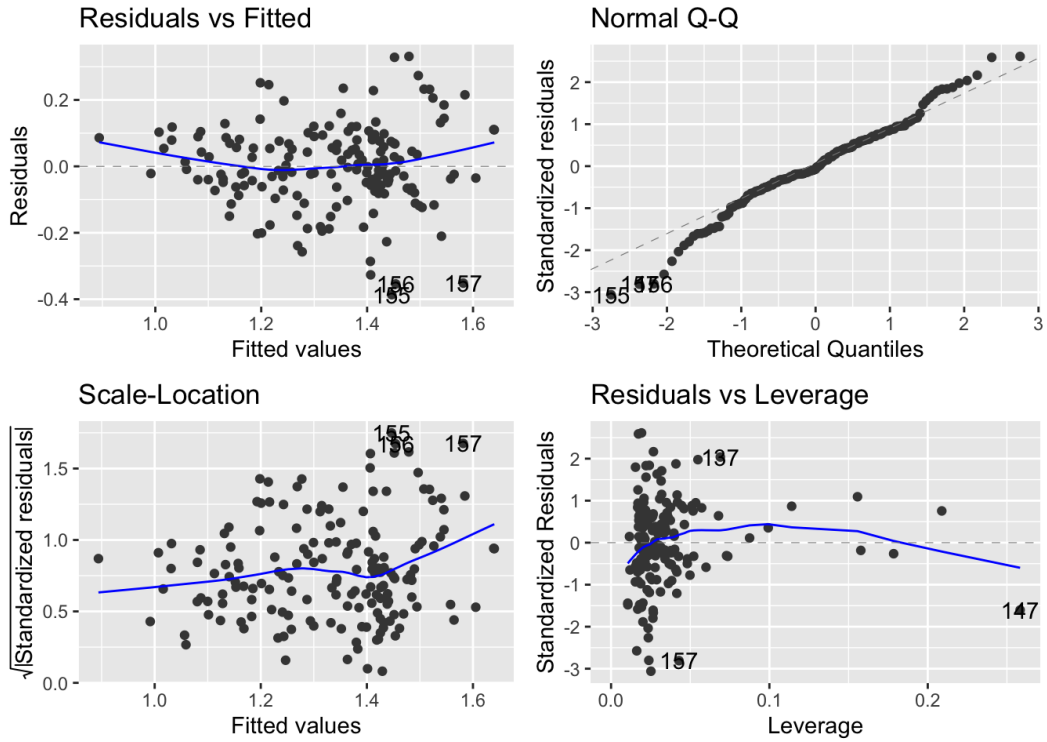
```
summary(MstepFtest)
```

```
##
## Call:
## lm(formula = sc.AvgPrice ~ log(sc.large_hass + 1) + sc.TotalBags +
##     sc.SmallBags + sqrt(sc.LargeBags) + sc.Year, data = organic.sc.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38650 -0.06256 -0.00896  0.07905  0.33102
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.879e+02  3.628e+01  -7.936 3.18e-13 ***
## log(sc.large_hass + 1) -1.220e-01  2.818e-02  -4.329 2.60e-05 ***
## sc.TotalBags            1.338e-05  4.911e-06   2.724  0.00715 **
## sc.SmallBags           -1.602e-05  4.819e-06  -3.323  0.00110 **
## sqrt(sc.LargeBags)     -4.854e-03  9.135e-04  -5.314 3.47e-07 ***
## sc.Year                 1.442e-01  1.797e-02   8.022 1.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1279 on 163 degrees of freedom
## Multiple R-squared:  0.5789, Adjusted R-squared:  0.566
## F-statistic: 44.82 on 5 and 163 DF,  p-value: < 2.2e-16
```

```
#sc.AvgPrice ~ log(sc.large_hass + 1) + sc.TotalBags + sc.SmallBags + sqrt(sc.LargeBags) + sc.Year
organic.sc$Date <- as.Date(organic.sc$Date, format = "%d/%m/%y" )
organic.sc <- organic.sc %>% mutate(month = paste0(month(organic.sc$Date)))
organic.sc_monthly = organic.sc %>% group_by(month)

library(lubridate)
july_largehass = mean(organic.sc_monthly$`4225`[organic.sc_monthly$month==7])
july_totalbag = mean(organic.sc_monthly$`Total Bags`[organic.sc_monthly$month==7])
july_smallbag = mean(organic.sc_monthly$`Small Bags`[organic.sc_monthly$month==7])
july_largebag = mean(organic.sc_monthly$`Large Bags`[organic.sc_monthly$month==7])


organic_predict_sc = MstepFtest$coefficients[1] + MstepFtest$coefficients[2]*log(july_largehass+1)+MstepFtest$c
oefficients[3]*july_totalbag + MstepFtest$coefficients[4]*july_smallbag + MstepFtest$coefficients[5]*sqrt(july_
largebag) + MstepFtest$coefficients[6]*2019
organic_predict_sc
```

```
## (Intercept)
##    1.684271
```

```
library(ggfortify)
autoplot(MstepFtest)
```




```
deviance (MstepFtest)
```

```
## [1] 2.667691
```

```
sort(lm.influence(MstepFtest)$hat)
```

```
##          53          65          61         101         104          79
## 0.01056198 0.01060897 0.01162788 0.01196543 0.01533896 0.01541925
##         124         144         153          99          77           8
## 0.01542743 0.01616394 0.01621311 0.01640454 0.01675980 0.01681089
##         150          12          11         100         122          47
## 0.01684055 0.01701478 0.01701516 0.01712650 0.01729008 0.01731854
##         127          45          17          33          27          54
## 0.01755054 0.01760207 0.01760383 0.01775651 0.01776296 0.01806018
##           9          26         103          42          30          32
## 0.01829338 0.01856211 0.01876162 0.01896151 0.01898314 0.01901762
##          37         151         123          66           7          35
## 0.01905894 0.01917682 0.01924293 0.01953258 0.01964732 0.01966364
##           4         106          34          78         146          67
## 0.01968955 0.01986315 0.02005058 0.02010891 0.02012301 0.02034407
##          38          19          20         107          13          81
## 0.02039458 0.02045180 0.02062974 0.02068485 0.02081324 0.02103107
##         132          22           3          75          14          36
## 0.02123218 0.02139057 0.02139383 0.02167523 0.02175265 0.02182467
##          60         121          16          50          31          80
## 0.02206391 0.02207368 0.02252892 0.02278937 0.02280671 0.02306292
##         154          46          28         152         128          82
## 0.02339360 0.02343387 0.02345677 0.02353526 0.02359539 0.02383366
##         156         116         149          23         155          43
## 0.02384791 0.02414599 0.02419220 0.02456384 0.02516512 0.02551269
##          10         131         105          39         129          56
## 0.02562879 0.02574781 0.02575062 0.02616326 0.02624133 0.02644271
##         148          74         102          94         119          18
## 0.02655189 0.02662424 0.02667307 0.02667894 0.02675748 0.02739590
##          40          92          91          44          41          21
## 0.02748113 0.02765479 0.02769267 0.02780913 0.02800248 0.02827581
##          29          95          24         115          49         126
## 0.02878072 0.02889438 0.02891368 0.02906058 0.02925993 0.02956785
##          90          84           6         112          55          93
## 0.02968194 0.03049602 0.03067587 0.03071089 0.03121547 0.03141615
##         114          96         133         113         120          98
## 0.03161938 0.03170858 0.03174901 0.03194314 0.03194930 0.03299957
##         168          97         110         125          76          62
## 0.03316045 0.03335680 0.03351404 0.03358245 0.03385007 0.03399135
##          52          15         163         108          59         134
## 0.03408401 0.03411962 0.03487939 0.03497531 0.03509434 0.03536819
##         160         140          58          48         161           5
## 0.03554264 0.03604769 0.03622276 0.03670884 0.03742593 0.03752464
##          25          51         165         166         145         167
## 0.03777447 0.03778043 0.03778431 0.03803590 0.03827334 0.03868462
##         111          87         162         159         169         109
## 0.03869878 0.03967348 0.03987810 0.04014893 0.04079686 0.04097494
##          57          68         139         158         164         157
## 0.04152543 0.04156471 0.04171600 0.04195250 0.04262521 0.04289122
##         118           1          83          63          69         130
## 0.04445603 0.04605577 0.04630816 0.04816310 0.04820791 0.04982602
##         117           2         142          89         138          85
## 0.05119884 0.05178650 0.05217144 0.05345849 0.05485124 0.05722451
##          70          88         137          64         143          86
## 0.05997320 0.06785982 0.06899145 0.07293203 0.07368354 0.08749542
##         135         136          73          71         141          72
## 0.09925891 0.11406612 0.15564195 0.15772342 0.17845630 0.20884465
##         147
## 0.25832195
```

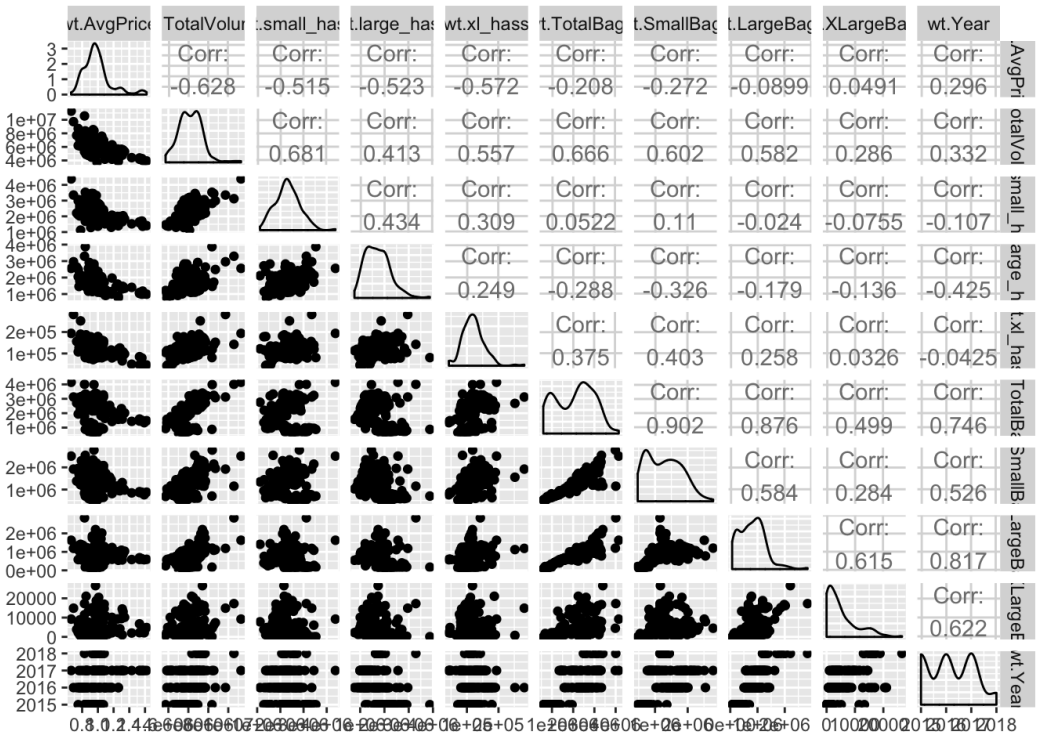
```
# for south central region july prediction
conventional_predict_sc
```

```
## (Intercept)
##   0.9686191
```

```
organic_predict_sc
```

```
## (Intercept)
##    1.684271
```

```
# West conventional data
avocado_conven = avocados_orginal[avocados_orginal$type=="conventional",]
conven.wt = avocado_conven[avocado_conven$region=="West",]
wt.AvgPrice = conven.wt$AveragePrice
wt.TotalVolume = conven.wt$`Total Volume`
wt.small_hass = conven.wt$`4046`
wt.large_hass = conven.wt$`4225`
wt.xl_hass = conven.wt$`4770`
wt.TotalBags = conven.wt$`Total Bags`
wt.SmallBags = conven.wt$`Small Bags`
wt.LargeBags = conven.wt$`Large Bags`
wt.XLargeBags = conven.wt$`XLarge Bags`
wt.Year = conven.wt$year
conven.wt.df = data_frame(wt.AvgPrice , wt.TotalVolume , wt.small_hass, wt.large_hass, wt.xl_hass, wt.TotalBag
s, wt.SmallBags, wt.LargeBags, wt.XLargeBags, wt.Year )
GGally::ggpairs(conven.wt.df, progress = F)
```




```
avocado_wst = avocados_orginal[avocados_orginal$region=="West",]
wst_con = avocado_wst[avocado_wst$type=="conventional",]
wst_org = avocado_wst[avocado_wst$type=="organic",]
```

```
M1 = lm(wst_con$AveragePrice ~ 1, data = wst_con)
Mf = lm(wst_con$AveragePrice ~ wst_con$`4046` + wst_con$`4225` + wst_con$`4770` + wst_con$`Small Bags` + wst_co
n$`Large Bags` + wst_con$`XLarge Bags` + wst_con$year, data = wst_con)
drop1(Mf, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.409602	-792.9340	*NA*	*NA*
wst_con$`4046`	1	0.23450154	1.644103	-768.9269	26.783986	6.698006e-07
wst_con$`4225`	1	0.19090419	1.600506	-773.4689	21.804441	6.328148e-06
wst_con$`4770`	1	0.02226644	1.431868	-792.2853	2.543199	1.127314e-01
wst_con$`Small Bags`	1	0.40410448	1.813706	-752.3350	46.155470	2.007538e-10
wst_con$`Large Bags`	1	0.28243618	1.692038	-764.0701	32.258921	6.157856e-08
wst_con$`XLarge Bags`	1	0.02896763	1.438569	-791.4962	3.308586	7.077686e-02
wst_con$year	1	0.72688661	2.136488	-724.6543	83.022571	3.078324e-16

8 rows

```
MstepFtest = update(Mf, . ~ . - wst_con$`4770`)
drop1(MstepFtest, test="F")
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.431868	-792.2853	*NA*	*NA*
wst_con$`4046`	1	0.26148012	1.693348	-765.9393	29.583579	1.942657e-07
wst_con$`4225`	1	0.21684327	1.648711	-770.4539	24.533414	1.821473e-06
wst_con$`Small Bags`	1	0.58696555	2.018834	-736.2270	66.408650	9.408440e-14
wst_con$`Large Bags`	1	0.38339132	1.815259	-754.1903	43.376481	6.010097e-10
wst_con$`XLarge Bags`	1	0.02894629	1.460814	-790.9029	3.274952	7.219860e-02
wst_con$year	1	0.97491995	2.406788	-706.5214	110.301392	5.255538e-20

7 rows

```
MstepFtest = update(MstepFtest, . ~ . -wst_con$`XLarge Bags`)
drop1(MstepFtest, test="F")
```

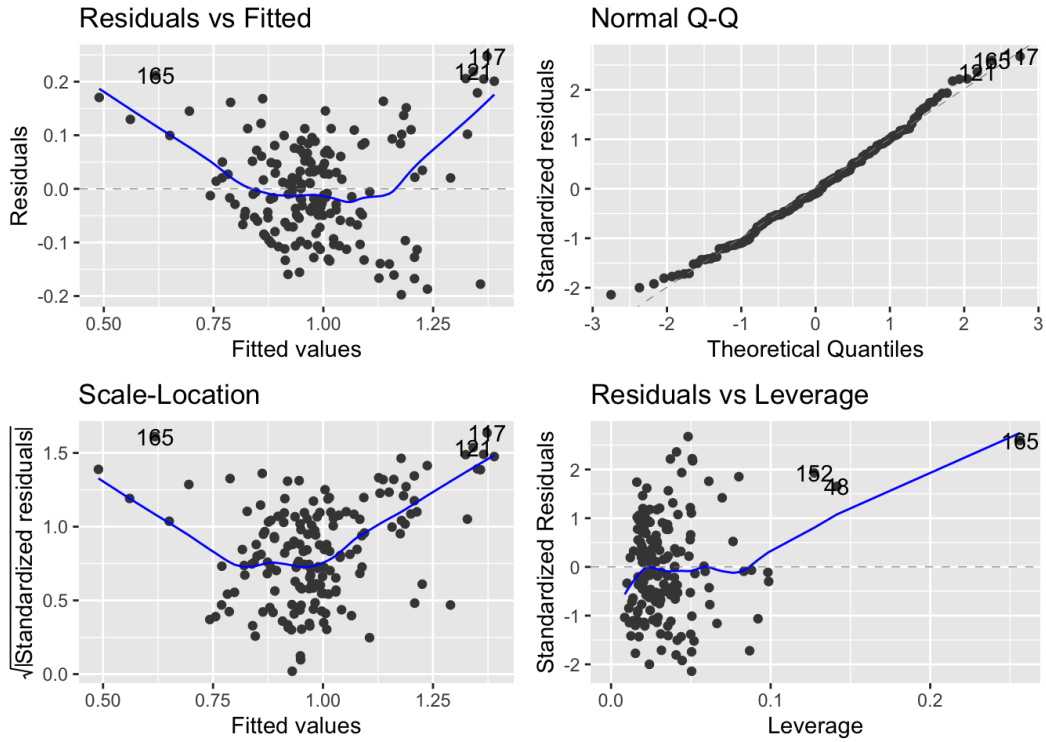
	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
<none>	*NA*	*NA*	1.460814	-790.9029	*NA*	*NA*
wst_con$`4046`	1	0.2520409	1.712855	-766.0036	28.12312	3.659737e-07
wst_con$`4225`	1	0.2349138	1.695728	-767.7019	26.21206	8.536830e-07
wst_con$`Small Bags`	1	0.5667854	2.027600	-737.4948	63.24282	2.889778e-13
wst_con$`Large Bags`	1	0.4498244	1.910639	-747.5359	50.19213	3.976670e-11
wst_con$year	1	0.9598001	2.420614	-707.5533	107.09604	1.308245e-19

6 rows

```
summary(MstepFtest)
```

```
##
## Call:
## lm(formula = wst_con$AveragePrice ~ wst_con$`4046` + wst_con$`4225` +
##     wst_con$`Small Bags` + wst_con$`Large Bags` + wst_con$year,
##     data = wst_con)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.197515 -0.061213 -0.008445  0.064794  0.247092
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3.217e+02  3.125e+01 -10.295  < 2e-16 ***
## wst_con$`4046`       -8.647e-08  1.631e-08  -5.303 3.66e-07 ***
## wst_con$`4225`       -1.015e-07  1.983e-08  -5.120 8.54e-07 ***
## wst_con$`Small Bags` -1.478e-07  1.859e-08  -7.953 2.89e-13 ***
## wst_con$`Large Bags` -2.199e-07  3.104e-08  -7.085 3.98e-11 ***
## wst_con$year          1.604e-01  1.550e-02  10.349  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09467 on 163 degrees of freedom
## Multiple R-squared:  0.7182, Adjusted R-squared:  0.7095
## F-statistic: 83.07 on 5 and 163 DF,  p-value: < 2.2e-16
```

```
library(ggfortify)
autoplot(MstepFtest)
```




```
deviance (MstepFtest)
```

```
## [1] 1.460814
```

```
sort(lm.influence(MstepFtest)$hat)
```

```
##         100          81          94          91         137          88
## 0.008544735 0.010026141 0.011118216 0.011475951 0.012045842 0.012462940
##         112          90         109         113         150         108
## 0.013064040 0.013726047 0.013727684 0.014134690 0.015221452 0.015252443
##         124         136         133         149          93          92
## 0.016151461 0.016373468 0.016625936 0.016635249 0.016651647 0.016785508
##          80          21         148          24          54          77
## 0.016892144 0.016945210 0.017126831 0.017432618 0.017488990 0.017507861
##          18          22          70          53          19          20
## 0.018079882 0.018349980 0.018977606 0.019170062 0.019224806 0.019322144
##          96         141          23          17         126         143
## 0.019559799 0.019665559 0.019676132 0.019760442 0.019795661 0.019910637
##         132          68         131         103         127          25
## 0.019922799 0.019953894 0.019955447 0.020330775 0.020533202 0.020620024
##           8          79          72          95         106          85
## 0.020633487 0.020770379 0.020850367 0.021035937 0.021162694 0.021190265
##          75         146         129          27          16         142
## 0.021276807 0.021277240 0.021284716 0.021331689 0.021720535 0.021810278
##         154          50         125          11         144         135
## 0.021931505 0.022136040 0.022425278 0.022657207 0.022962528 0.023140960
##          43          38          46         107          41         123
## 0.023196865 0.023620817 0.023923247 0.024017598 0.024281409 0.024313207
##          59          83           9          67          84         134
## 0.024392095 0.024436975 0.024503263 0.024505651 0.024723155 0.024884296
##          49         114          73          76          58          98
## 0.024966773 0.025288156 0.025676029 0.025698729 0.025715142 0.026241744
##          51           4          47          74         140          82
## 0.026912661 0.027914040 0.028196132 0.028199835 0.028208055 0.028482652
##         128           1          28          34          56          57
## 0.028737331 0.028783818 0.028785722 0.028985088 0.029141681 0.029428504
##         155         138          64         153           3          97
## 0.030403518 0.030405383 0.030442819 0.031079890 0.031108062 0.031399462
##           7         151          60          89          10          62
## 0.031487855 0.031542581 0.032170601 0.032815021 0.033047445 0.033899692
##          14          29          40          37          33          78
## 0.034479468 0.034697605 0.034703845 0.034818014 0.035327248 0.035411919
##         105          45           2          63          66          71
## 0.035508690 0.035737816 0.035771741 0.035879293 0.036029171 0.036364213
##          52         122          87           6          26          36
## 0.037165536 0.037185941 0.037314728 0.037483227 0.038710424 0.039180384
##          31          32          61          39         145         111
## 0.039667147 0.039744427 0.039924610 0.039945605 0.040186665 0.040766476
##         121         102         147           5          42          86
## 0.041033758 0.041330110 0.041429207 0.041480702 0.041696716 0.041907348
##          69         161         120         110         117         115
## 0.043712717 0.043905530 0.044284920 0.044551603 0.048247496 0.048833728
##          35         139         167          30         169         116
## 0.049577644 0.049650708 0.049657813 0.049896207 0.049932242 0.050338345
##         160         163         158         118         119         166
## 0.050466755 0.050478951 0.050638298 0.050833168 0.051141159 0.052008147
##         104          44          13         130         159         156
## 0.052590650 0.058236653 0.058979972 0.061258409 0.061469363 0.061865380
##         164          99         168          65          15         157
## 0.066373065 0.069622431 0.076440522 0.080082110 0.083207897 0.086826595
##          55         101          12         162         152          48
## 0.087984561 0.092063651 0.098265906 0.098760012 0.127333521 0.141512114
##         165
## 0.256302854
```

```
#wst_con$AveragePrice ~ wst_con$`4046` + wst_con$`4225` + wst_con$`Small Bags` + wst_con$`Large Bags` + wst_con
$year
wst_con$Date <- as.Date(wst_con$Date, format = "%d/%m/%y" )
wst_con <- wst_con %>% mutate(month = paste0(month(wst_con$Date)))
wst_con_monthly = wst_con %>% group_by(month)

library(lubridate)
july_smallhass = mean(wst_con_monthly$`4046`[wst_con_monthly$month==7])
july_largehass = mean(wst_con_monthly$`4225`[wst_con_monthly$month==7])
july_smallbag = mean(wst_con_monthly$`Small Bags`[wst_con_monthly$month==7])
july_largebag = mean(wst_con_monthly$`Large Bags`[wst_con_monthly$month==7])


con_predict_wst = MstepFtest$coefficients[1] + MstepFtest$coefficients[2]*july_smallhass+MstepFtest$coefficient
s[3]*july_largehass + MstepFtest$coefficients[4]*july_smallbag + MstepFtest$coefficients[5]*july_largebag + Mst
epFtest$coefficients[6]*2019
con_predict_wst
```

```
## (Intercept)
##    1.450977
```

```
# west organic avocado
M1 = lm(wst_org$AveragePrice ~ 1, data = wst_org)
Mf = lm(wst_org$AveragePrice ~ wst_org$`4046` + wst_org$`4225` + wst_org$`4770` + wst_org$`Small Bags` + wst_or
g$`Large Bags` + wst_org$`XLarge Bags` + wst_org$year, data = wst_org)
drop1(Mf, test="F")
```

	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	6.848527	-525.7912	*NA*	*NA*
wst_org$`4046`	1	0.062941881	6.911469	-526.2451	1.4796821	2.256046e-01
wst_org$`4225`	1	0.602396816	7.450924	-513.5438	14.1615687	2.346271e-04
wst_org$`4770`	1	0.004880045	6.853407	-527.6708	0.1147235	7.352706e-01
wst_org$`Small Bags`	1	0.405270954	7.253798	-518.0751	9.5273951	2.383961e-03
wst_org$`Large Bags`	1	7.964766043	14.813293	-397.4092	187.2413304	9.087573e-29
wst_org$`XLarge Bags`	1	0.061531475	6.910059	-526.2796	1.4465253	2.308521e-01
wst_org$year	1	0.899679946	7.748207	-506.9319	21.1503099	8.549365e-06

8 rows

```
MstepFtest = update(Mf, . ~ . - wst_org$`4770`)
drop1(MstepFtest, test="F")
```

	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	6.853407	-527.6708	*NA*	*NA*
wst_org$`4046`	1	0.05857759	6.911985	-528.2325	1.384650	2.410368e-01
wst_org$`4225`	1	0.60004353	7.453451	-515.4865	14.183756	2.316433e-04
wst_org$`Small Bags`	1	0.42189690	7.275304	-519.5748	9.972747	1.895420e-03
wst_org$`Large Bags`	1	8.07353995	14.926947	-398.1175	190.841349	3.503682e-29
wst_org$`XLarge Bags`	1	0.05939931	6.912806	-528.2124	1.404074	2.377781e-01
wst_org$year	1	0.96112757	7.814535	-507.4913	22.719016	4.146292e-06

7 rows

```
MstepFtest = update(MstepFtest, . ~ . - wst_org$`4046`)
drop1(MstepFtest, test="F")
```

	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	6.911985	-528.2325	*NA*	*NA*
wst_org$`4225`	1	1.28458036	8.196565	-501.4250	30.293267	1.416024e-07
wst_org$`Small Bags`	1	0.44350897	7.355494	-519.7223	10.458930	1.477721e-03
wst_org$`Large Bags`	1	8.03293335	14.944918	-399.9141	189.434465	4.306812e-29
wst_org$`XLarge Bags`	1	0.06969503	6.981680	-528.5369	1.643564	2.016576e-01
wst_org$year	1	1.05010302	7.962088	-506.3301	24.763769	1.633523e-06

6 rows

```
MstepFtest = update(MstepFtest, . ~ . - wst_org$`XLarge Bags`)
drop1(MstepFtest, test="F")
```

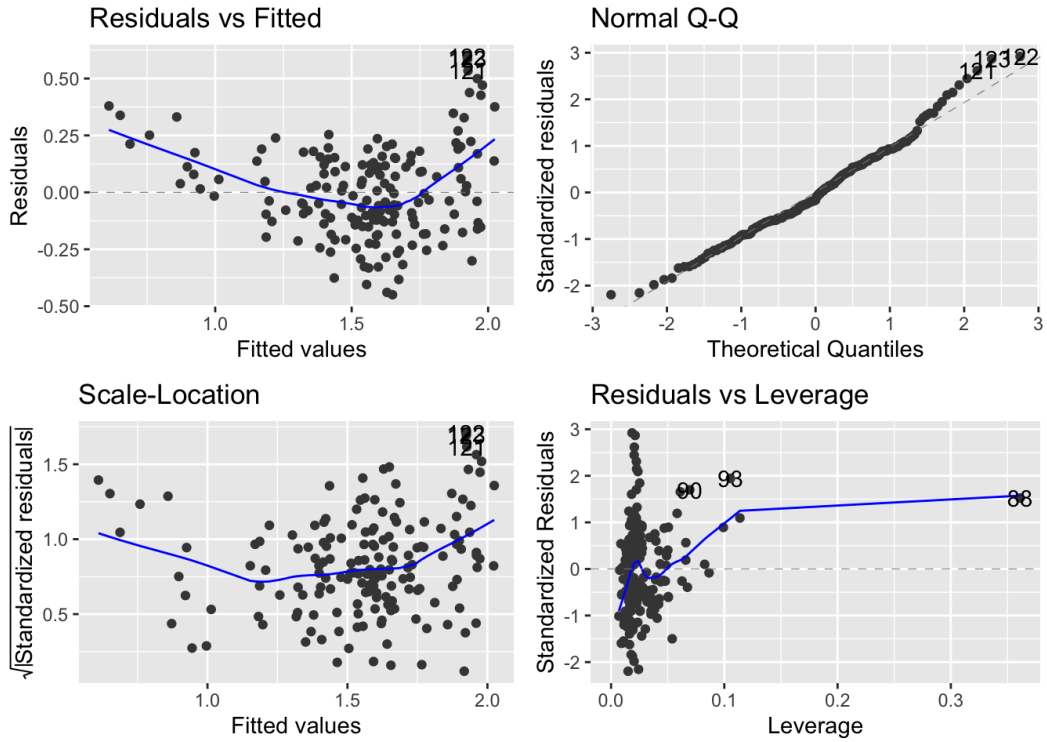
	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	6.981680	-528.5369	*NA*	*NA*
wst_org$`4225`	1	1.2601889	8.241869	-502.4935	29.60190	1.900201e-07
wst_org$`Small Bags`	1	0.4769125	7.458592	-519.3699	11.20270	1.012894e-03
wst_org$`Large Bags`	1	8.0582702	15.039950	-400.8429	189.28916	3.959405e-29
wst_org$year	1	1.4966651	8.478345	-497.7128	35.15674	1.746625e-08

5 rows

```
summary(MstepFtest)
```

```
##
## Call:
## lm(formula = wst_org$AveragePrice ~ wst_org$`4225` + wst_org$`Small Bags` +
##     wst_org$`Large Bags` + wst_org$year, data = wst_org)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44962 -0.12946 -0.03344  0.13406  0.59708
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3.289e+02  5.583e+01  -5.890 2.13e-08 ***
## wst_org$`4225`       -3.048e-06  5.602e-07  -5.441 1.90e-07 ***
## wst_org$`Small Bags` -2.183e-06  6.521e-07  -3.347  0.00101 **
## wst_org$`Large Bags` -5.406e-06  3.930e-07 -13.758  < 2e-16 ***
## wst_org$year          1.642e-01  2.769e-02   5.929 1.75e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2063 on 164 degrees of freedom
## Multiple R-squared:  0.6588, Adjusted R-squared:  0.6505
## F-statistic: 79.17 on 4 and 164 DF,  p-value: < 2.2e-16
```

```
library(ggfortify)
autoplot(MstepFtest)
```




```
deviance (MstepFtest)
```

```
## [1] 6.98168
```

```
sort(lm.influence(MstepFtest)$hat)
```

```
##          91          74          59         101          69          92
## 0.006940315 0.007653236 0.008691594 0.009053860 0.009342220 0.009560519
##         100          78          99          73         105          97
## 0.010028260 0.010062376 0.010181928 0.010789377 0.010815669 0.011119006
##         102         150          58         104          60         109
## 0.011422429 0.011774728 0.012578502 0.012599061 0.012877185 0.012885810
##          63         110          75          68          89          76
## 0.013169316 0.013495179 0.013682315 0.013935962 0.013991506 0.014064834
##          96          71          77         154          34          39
## 0.014151550 0.014418795 0.014897881 0.015041470 0.015204320 0.015435596
##          48         151          65          20          57          26
## 0.015571060 0.015614069 0.015642567 0.015740139 0.015905747 0.016005955
##          33          52          37          54          55          29
## 0.016148390 0.016250015 0.016279484 0.016288521 0.016316214 0.016378451
##          19          32          30          64          13          70
## 0.016444931 0.016475234 0.016475377 0.016744511 0.016826778 0.016869998
##          28           8          46         126         155          45
## 0.016915894 0.017016341 0.017151929 0.017641249 0.017766521 0.017824230
##          72          23          49         125         127         141
## 0.017859884 0.017994993 0.018091078 0.018122339 0.018297865 0.018339544
##         122         111          66          25         153          44
## 0.018529176 0.018602926 0.018804405 0.018895371 0.018906169 0.019232311
##          12         106          62         152         124         149
## 0.020020906 0.020213766 0.020243323 0.020319779 0.020407821 0.020464865
##         121          21          17          22          42         130
## 0.020532834 0.020762803 0.020810264 0.020882619 0.020920117 0.020938610
##         112         123          18          11          14          79
## 0.021099023 0.021218502 0.021500746 0.021828075 0.022050216 0.022157875
##         114         118          47         129         113         117
## 0.022287383 0.022338693 0.022365994 0.022393761 0.022515847 0.022560611
##          51         128          16          81          61         120
## 0.022731377 0.022910844 0.023449526 0.023555604 0.023657625 0.023682310
##          10           4         157         131         140         133
## 0.023704314 0.024071744 0.024244936 0.024351752 0.024978412 0.025250361
##         119           9          35           7          67           1
## 0.025284142 0.025388627 0.025518604 0.025540785 0.025643496 0.025783086
##          24         116           2         115          56         146
## 0.025946854 0.026111961 0.026545564 0.026641219 0.026861119 0.027036002
##          15          27         132         108          31           5
## 0.027050965 0.027100451 0.027224552 0.027226073 0.027641859 0.027922178
##           3         147         162          86          83         103
## 0.028697118 0.030235736 0.031330400 0.031558412 0.031831573 0.032009891
##         169           6         142         164         148          82
## 0.033674376 0.034090305 0.034601048 0.034640068 0.035217247 0.035358245
##         166         167         160         159         134          84
## 0.036294388 0.036655847 0.036957391 0.038406965 0.038660895 0.039333856
##         135         137         156         168          94         107
## 0.039600282 0.039965826 0.040697284 0.041956641 0.042337417 0.042562233
##          41         158          50          38         165          95
## 0.043394944 0.043871622 0.046296707 0.046964055 0.047886900 0.048632063
##          98         136          85          80          40         163
## 0.049029430 0.050129956 0.050497919 0.050572082 0.050599315 0.053986733
##         161         139          36          53          87         143
## 0.058342877 0.061081725 0.062634144 0.065958059 0.066045401 0.067401757
##          90          43         144         138          93         145
## 0.069523515 0.082648504 0.086533514 0.099187940 0.105446970 0.113780105
##          88
## 0.361155081
```

```
#wst_org$AveragePrice ~ wst_org$`4225` + wst_org$`Small Bags` + wst_org$`Large Bags` + wst_org$year
wst_org$Date <- as.Date(wst_org$Date, format = "%d/%m/%y" )
wst_org <- wst_org %>% mutate(month = paste0(month(wst_org$Date)))
wst_org_monthly = wst_org %>% group_by(month)

library(lubridate)
july_largehass = mean(wst_org_monthly$`4225`[wst_org_monthly$month==7])
july_smallbag = mean(wst_org_monthly$`Small Bags`[wst_org_monthly$month==7])
july_largebag = mean(wst_org_monthly$`Large Bags`[wst_org_monthly$month==7])

org_predict_wst = MstepFtest$coefficients[1] + MstepFtest$coefficients[2]*july_largehass+MstepFtest$coefficient
s[3]*july_smallbag + MstepFtest$coefficients[4]*july_largebag + MstepFtest$coefficients[5]*2019
org_predict_wst
```

```
## (Intercept)
##    2.171533
```

```
conventional_predict_sc
```

```
## (Intercept)
##   0.9686191
```

```
organic_predict_sc
```

```
## (Intercept)
##    1.684271
```

```
con_predict_wst
```

```
## (Intercept)
##    1.450977
```

```
org_predict_wst
```

```
## (Intercept)
##    2.171533
```