MGSC 310 - Final Project

Kayelin Santa Elena, Christian Muresan, Nina Valdez, Oliver Brooker

5/22/2020

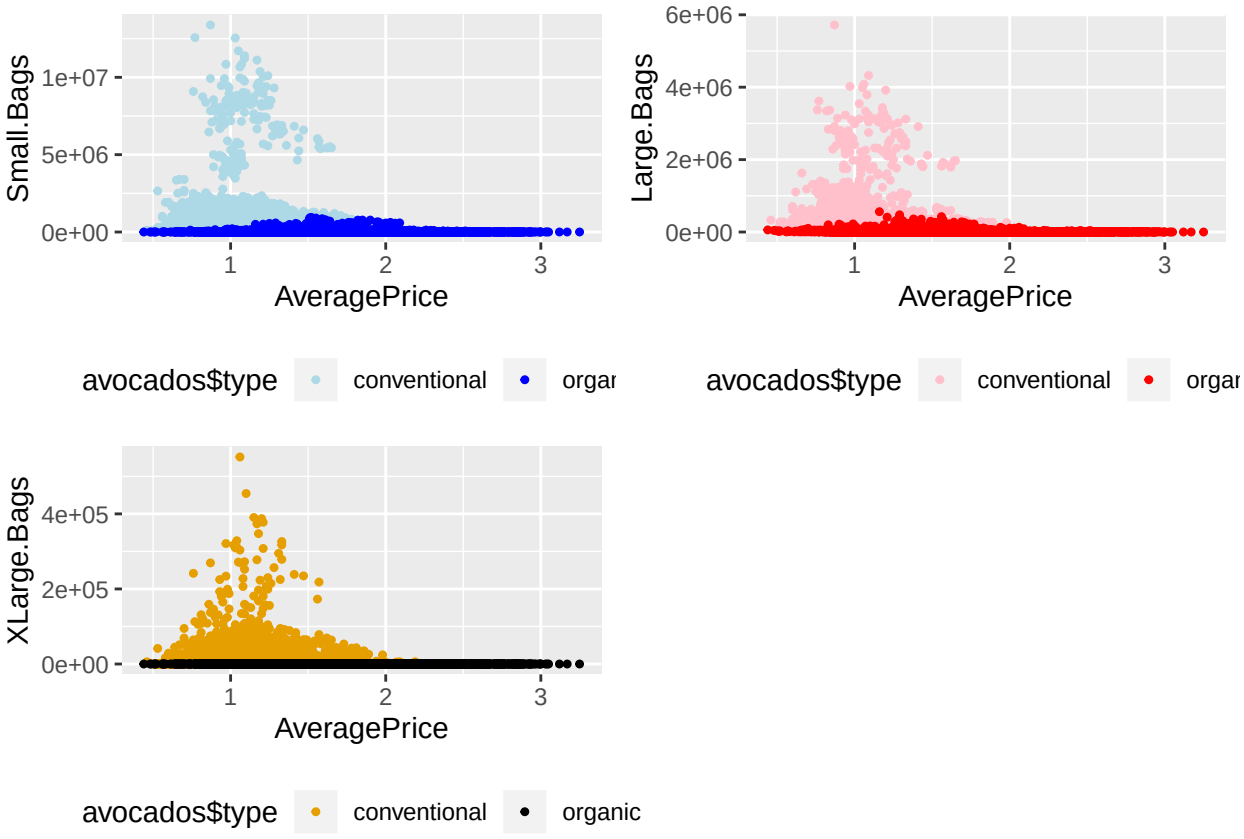
Data Exploration

```
library(cowplot)
library(ggplot2)
```

Scatter Plots comparing Average Price and Bag Sizes

plot for types of bags

```
avocados <- read.csv("/Users/kksizzle/Desktop/MGSC 310/final project/avocado.csv")

Small <- ggplot(avocados,aes(x = AveragePrice, y = Small.Bags))+
  geom_point(aes(color=avocados$type), size = 0.9)+
  scale_color_manual(values = c("lightblue", "blue"))+
  theme(legend.position = "bottom")

### plot for large bags
Large <- ggplot(avocados,aes(x = AveragePrice, y = Large.Bags))+
  geom_point(aes(color=avocados$type), size = 0.9)+
  scale_color_manual(values = c("pink", "red"))+
  theme(legend.position = "bottom")

### plot for Xlarge bags
XL <- ggplot(avocados,aes(x = AveragePrice, y = XLarge.Bags))+
  geom_point(aes(color=avocados$type), size = 0.9)+
  scale_color_manual(values = c("#E69F00", "black"))+
  theme(legend.position = "bottom")

### plots 3 scatterplots at same time
plot_grid(Small, Large, XL)
```



Linear Regression

Import packages

```
library(caret)
library(rsq)
library(olsrr)
```

Import data

```
avocados <- read.csv("/Users/kksizzle/Desktop/MGSC 310/final project/avocado.csv")
```

Basic data cleaning / organizing

```
names(avocados)[5:7] <- c("Small.Size","Medium.Size","Large.Size")  #rename columns
avocados <- avocados[,-1]  #get rid of first column
```

test / train data

```
set.seed(310)
index <- sample(1:nrow(avocados),size=0.75*nrow(avocados),replace=FALSE)
train <- avocados[index,]
test <- avocados[-index,]
```

Linear model

```
mod1_lm_train <- lm(AveragePrice ~ . -Date,
                    data = train)
```

Summary

```
summary(mod1_lm_train)
```

```
##
## Call:
## lm(formula = AveragePrice ~ . - Date, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09396 -0.16077 -0.00958  0.14514  1.47274
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -7.671e+01  5.022e+00 -15.276  < 2e-16 ***
## Total.Volume               -7.773e-05  8.730e-05  -0.890  0.37326
## Small.Size                  7.774e-05  8.730e-05   0.890  0.37324
## Medium.Size                 7.773e-05  8.730e-05   0.890  0.37328
## Large.Size                  7.764e-05  8.730e-05   0.889  0.37384
## Total.Bags                  1.541e-02  3.703e-02   0.416  0.67725
## Small.Bags                 -1.534e-02  3.703e-02  -0.414  0.67879
## Large.Bags                 -1.534e-02  3.703e-02  -0.414  0.67879
## XLarge.Bags                -1.533e-02  3.703e-02  -0.414  0.67882
## typeorganic                 4.897e-01  4.823e-03 101.516  < 2e-16 ***
## year                        3.871e-02  2.491e-03  15.540  < 2e-16 ***
## regionAtlanta              -2.150e-01  2.396e-02  -8.974  < 2e-16 ***
## regionBaltimoreWashington -4.387e-02  2.371e-02  -1.850  0.06434 .
## regionBoise                -2.354e-01  2.418e-02  -9.734  < 2e-16 ***
## regionBoston               -2.526e-02  2.361e-02  -1.070  0.28482
## regionBuffaloRochester     -5.927e-02  2.375e-02  -2.496  0.01257 *
## regionCalifornia           -1.599e-01  2.416e-02  -6.617 3.80e-11 ***
## regionCharlotte             4.096e-02  2.333e-02   1.756  0.07915 .
## regionChicago              -5.515e-03  2.398e-02  -0.230  0.81814
## regionCincinnatiDayton     -3.519e-01  2.356e-02 -14.935  < 2e-16 ***
## regionColumbus             -3.270e-01  2.387e-02 -13.700  < 2e-16 ***
## regionDallasFtWorth        -4.886e-01  2.345e-02 -20.833  < 2e-16 ***
```

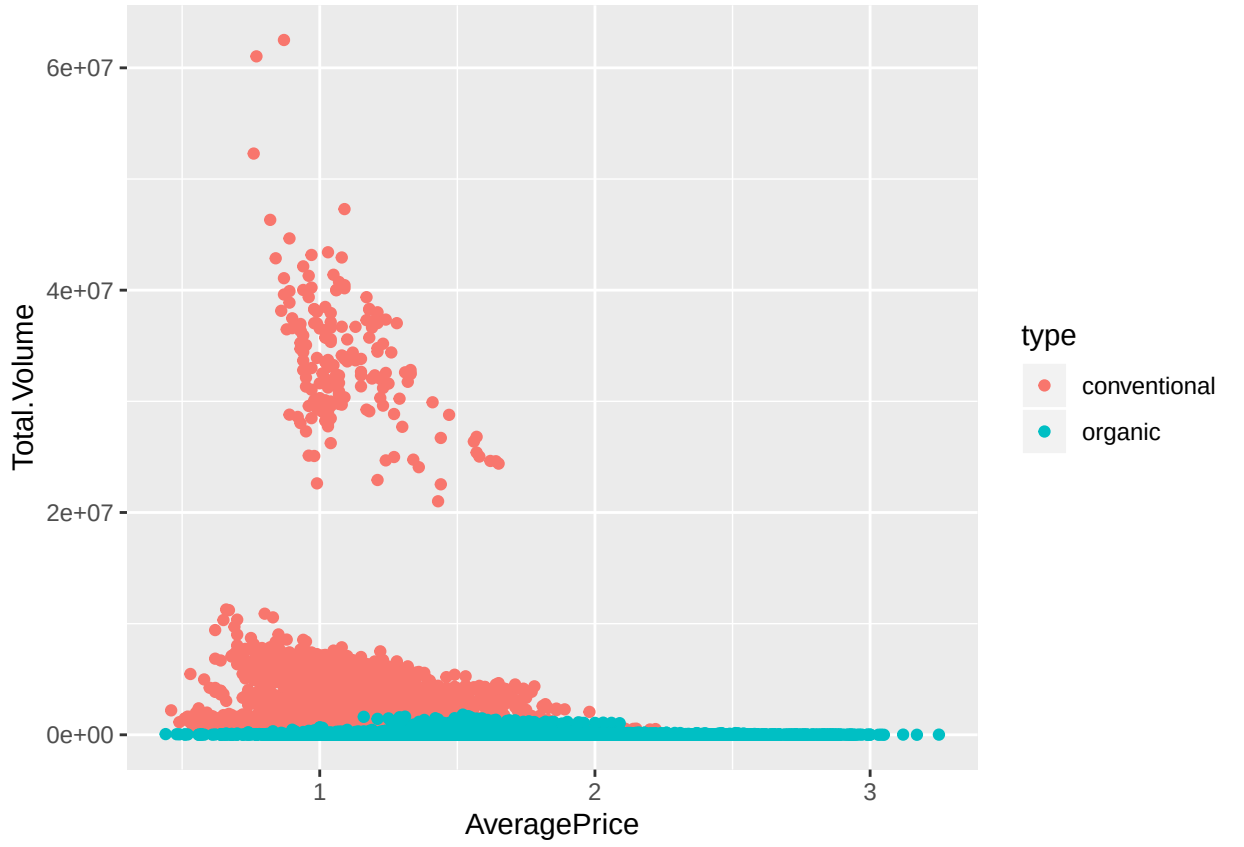
```
## regionDenver              -3.496e-01  2.398e-02 -14.578  < 2e-16 ***
## regionDetroit             -2.932e-01  2.344e-02 -12.506  < 2e-16 ***
## regionGrandRapids         -5.842e-02  2.370e-02  -2.464  0.01374 *
## regionGreatLakes          -2.246e-01  2.451e-02  -9.166  < 2e-16 ***
## regionHarrisburgScranton  -6.386e-02  2.352e-02  -2.716  0.00662 **
## regionHartfordSpringfield  2.376e-01  2.395e-02   9.921  < 2e-16 ***
## regionHouston             -5.127e-01  2.353e-02 -21.790  < 2e-16 ***
## regionIndianapolis        -2.565e-01  2.337e-02 -10.973  < 2e-16 ***
## regionJacksonville        -5.212e-02  2.352e-02  -2.216  0.02672 *
## regionLasVegas            -1.904e-01  2.359e-02  -8.070 7.63e-16 ***
## regionLosAngeles          -3.669e-01  2.393e-02 -15.333  < 2e-16 ***
## regionLouisville          -3.016e-01  2.368e-02 -12.741  < 2e-16 ***
## regionMiamiFtLauderdale   -1.617e-01  2.375e-02  -6.808 1.03e-11 ***
## regionMidsouth            -1.633e-01  2.391e-02  -6.833 8.69e-12 ***
## regionNashville           -3.556e-01  2.352e-02 -15.119  < 2e-16 ***
## regionNewOrleansMobile    -2.658e-01  2.359e-02 -11.266  < 2e-16 ***
## regionNewYork              1.558e-01  2.413e-02   6.460 1.08e-10 ***
## regionNortheast            3.550e-02  2.564e-02   1.385  0.16612
## regionNorthernNewEngland  -1.015e-01  2.371e-02  -4.282 1.87e-05 ***
## regionOrlando             -5.424e-02  2.352e-02  -2.306  0.02114 *
## regionPhiladelphia         7.007e-02  2.348e-02   2.984  0.00285 **
## regionPhoenixTucson       -3.353e-01  2.394e-02 -14.005  < 2e-16 ***
## regionPittsburgh          -2.079e-01  2.365e-02  -8.791  < 2e-16 ***
## regionPlains              -1.406e-01  2.363e-02  -5.949 2.77e-09 ***
## regionPortland            -2.406e-01  2.365e-02 -10.171  < 2e-16 ***
## regionRaleighGreensboro   -1.095e-02  2.357e-02  -0.465  0.64228
## regionRichmondNorfolk     -2.807e-01  2.349e-02 -11.946  < 2e-16 ***
## regionRoanoke             -3.093e-01  2.370e-02 -13.050  < 2e-16 ***
## regionSacramento           4.603e-02  2.390e-02   1.926  0.05407 .
## regionSanDiego            -1.767e-01  2.363e-02  -7.476 8.11e-14 ***
## regionSanFrancisco         2.423e-01  2.369e-02  10.224  < 2e-16 ***
## regionSeattle             -1.292e-01  2.421e-02  -5.337 9.60e-08 ***
## regionSouthCarolina       -1.670e-01  2.385e-02  -7.000 2.68e-12 ***
## regionSouthCentral        -4.658e-01  2.440e-02 -19.093  < 2e-16 ***
## regionSoutheast           -1.681e-01  2.441e-02  -6.884 6.06e-12 ***
## regionSpokane             -1.351e-01  2.397e-02  -5.637 1.76e-08 ***
## regionStLouis             -1.363e-01  2.356e-02  -5.785 7.43e-09 ***
## regionSyracuse            -5.427e-02  2.397e-02  -2.264  0.02357 *
## regionTampa               -1.524e-01  2.376e-02  -6.414 1.46e-10 ***
## regionTotalUS             -1.820e-01  3.011e-02  -6.045 1.53e-09 ***
## regionWest                -2.557e-01  2.457e-02 -10.407  < 2e-16 ***
## regionWestTexNewMexico    -3.051e-01  2.365e-02 -12.900  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2665 on 13622 degrees of freedom
## Multiple R-squared:  0.5604, Adjusted R-squared:  0.5584
## F-statistic: 275.7 on 63 and 13622 DF,  p-value: < 2.2e-16
```

Predictions

```r
#train
preds_train1 <- predict(mod1_lm_train)
preds_train_df1 <- data.frame(true = train$AveragePrice, pred = preds_train1,
                              resid = mod1_lm_train$residuals)

#test
preds_test1 <-  predict(mod1_lm_train, newdata = test)
preds_test_df1 <- data.frame(true = test$AveragePrice, pred = preds_test1)
```

Graphs

```r
ggplot(avocados, aes(AveragePrice, Total.Volume,color = type)) +
  geom_point()
```



Model accuracy: RMSE and R2

There is not much of an overfitting issue since there is no big difference between the RSMEs. R2 is very low, the model didn't score too well.

```r
# train RMSE
RMSE(preds_train_df1$pred, preds_train_df1$true)
```

```
## [1] 0.2658955
```

```
# test RMSE
RMSE(preds_test_df1$pred, preds_test_df1$true)
```

```
## [1] 0.2736647
```

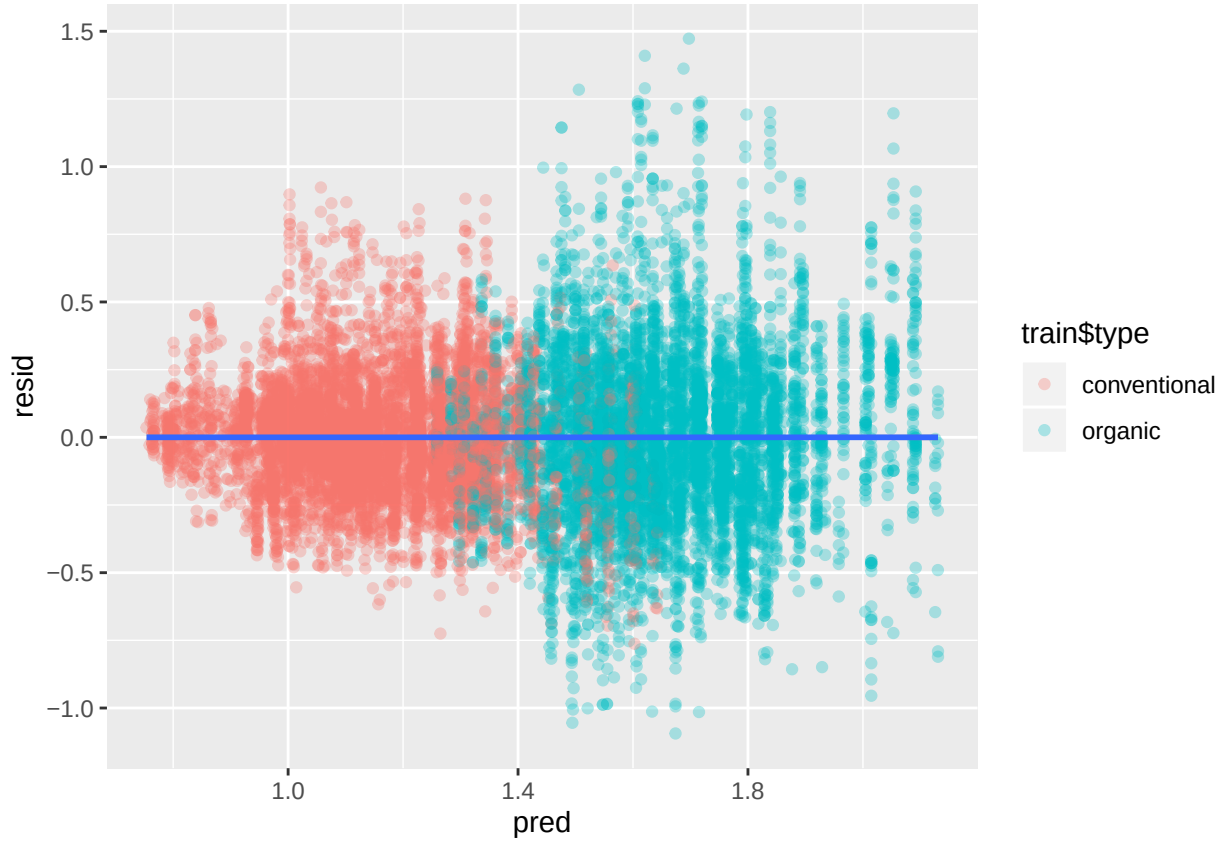
```
# R2
rsq(mod1_lm_train)
```

```
## [1] 0.5604371
```

Heteroskedasticity

There are signs of heteroskedasticity which may contribute to the low R2.

```
ggplot(preds_train_df1, aes(pred, resid)) +
  geom_point(alpha = 0.3, aes(color = train$type)) +
  geom_smooth(method = "lm")
```



Collinearity

VIF > 10 indicates problematic level of multicollinearity. There seems to be no collinearity issue.

```
ols_vif_tol(mod1_lm_train)
```

```
##                    Variables    Tolerance          VIF
## 1               Total.Volume 5.392709e-11 1.854356e+10
## 2                 Small.Size 4.051424e-10 2.468268e+09
## 3                Medium.Size 4.446898e-10 2.248758e+09
## 4                 Large.Size 5.743809e-08 1.741005e+07
## 5                 Total.Bags 3.663736e-15 2.729454e+14
## 6                 Small.Bags 6.439294e-15 1.552965e+14
## 7                 Large.Bags 5.995204e-14 1.668000e+13
## 8                XLarge.Bags 1.169320e-11 8.551977e+10
## 9                typeorganic 8.923599e-01 1.120624e+00
## 10                      year 9.471892e-01 1.055755e+00
## 11            regionAtlanta 5.203385e-01 1.921826e+00
## 12 regionBaltimoreWashington 5.086607e-01 1.965947e+00
## 13              regionBoise 5.302679e-01 1.885839e+00
## 14             regionBoston 5.052161e-01 1.979351e+00
## 15   regionBuffaloRochester 5.132606e-01 1.948328e+00
## 16         regionCalifornia 4.879901e-01 2.049222e+00
## 17          regionCharlotte 4.949103e-01 2.020568e+00
## 18            regionChicago 5.132842e-01 1.948238e+00
## 19   regionCincinnatiDayton 5.035653e-01 1.985840e+00
## 20           regionColumbus 5.180393e-01 1.930355e+00
## 21      regionDallasFtWorth 4.969933e-01 2.012100e+00
## 22             regionDenver 5.113436e-01 1.955632e+00
## 23            regionDetroit 4.956058e-01 2.017733e+00
## 24        regionGrandRapids 5.110553e-01 1.956735e+00
## 25         regionGreatLakes 4.568345e-01 2.188977e+00
## 26  regionHarrisburgScranton 5.036244e-01 1.985607e+00
## 27 regionHartfordSpringfield 5.207587e-01 1.920275e+00
## 28            regionHouston 5.011003e-01 1.995608e+00
## 29       regionIndianapolis 4.967238e-01 2.013191e+00
## 30       regionJacksonville 5.033724e-01 1.986601e+00
## 31           regionLasVegas 5.061565e-01 1.975674e+00
## 32         regionLosAngeles 4.862271e-01 2.056652e+00
## 33         regionLouisville 5.103055e-01 1.959611e+00
## 34   regionMiamiFtLauderdale 5.109159e-01 1.957269e+00
## 35           regionMidsouth 5.005004e-01 1.998000e+00
## 36          regionNashville 5.034215e-01 1.986407e+00
## 37   regionNewOrleansMobile 5.061816e-01 1.975576e+00
## 38            regionNewYork 5.113124e-01 1.955751e+00
## 39          regionNortheast 4.386631e-01 2.279654e+00
## 40  regionNorthernNewEngland 5.107441e-01 1.957928e+00
## 41            regionOrlando 5.032495e-01 1.987086e+00
## 42       regionPhiladelphia 5.012831e-01 1.994881e+00
## 43      regionPhoenixTucson 5.150539e-01 1.941544e+00
## 44         regionPittsburgh 5.093778e-01 1.963180e+00
## 45             regionPlains 5.024045e-01 1.990428e+00
## 46          regionPortland 5.072859e-01 1.971275e+00
## 47   regionRaleighGreensboro 5.050335e-01 1.980067e+00
## 48     regionRichmondNorfolk 5.026489e-01 1.989460e+00
## 49            regionRoanoke 5.112625e-01 1.955942e+00
## 50         regionSacramento 5.191165e-01 1.926350e+00
```

```
## 51            regionSanDiego 5.081710e-01 1.967842e+00
## 52       regionSanFrancisco 5.094993e-01 1.962711e+00
## 53            regionSeattle 5.292712e-01 1.889390e+00
## 54      regionSouthCarolina 5.168932e-01 1.934635e+00
## 55       regionSouthCentral 4.732168e-01 2.113196e+00
## 56          regionSoutheast 4.837962e-01 2.066986e+00
## 57            regionSpokane 5.221964e-01 1.914988e+00
## 58            regionStLouis 5.054379e-01 1.978482e+00
## 59           regionSyracuse 5.222998e-01 1.914609e+00
## 60              regionTampa 5.127234e-01 1.950369e+00
## 61            regionTotalUS 3.083367e-01 3.243208e+00
## 62               regionWest 4.683278e-01 2.135256e+00
## 63    regionWestTexNewMexico 5.054141e-01 1.978576e+00
```

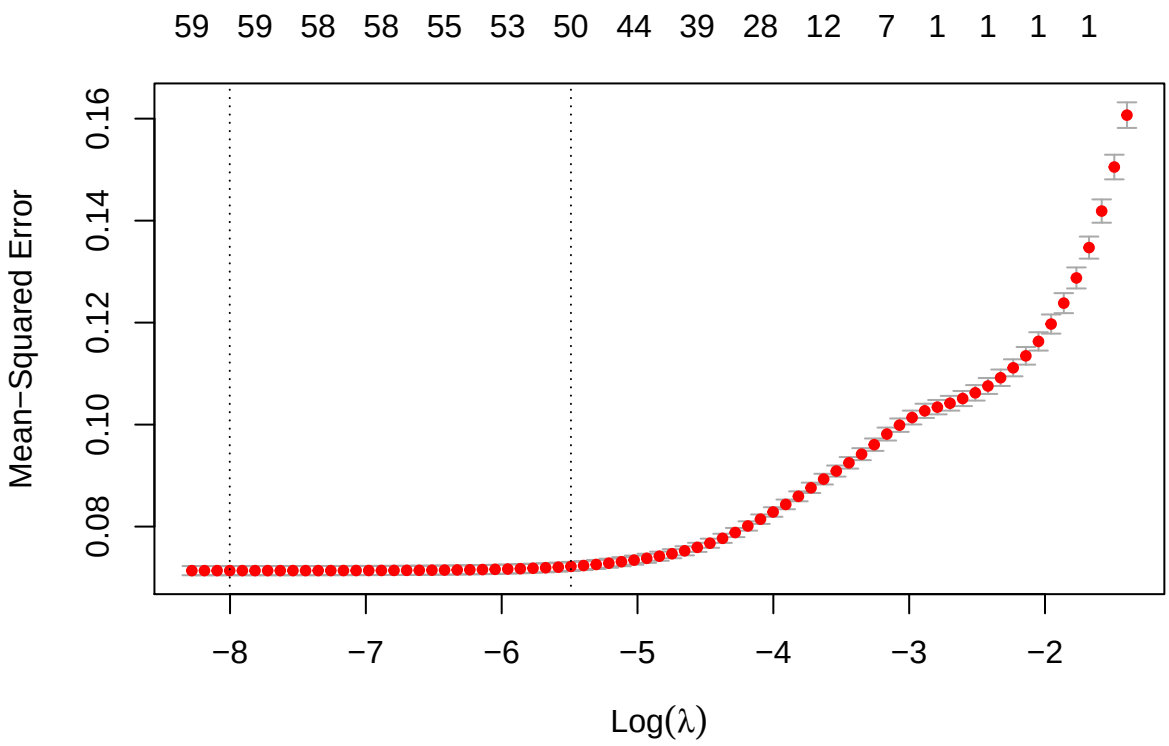
Lasso Regression

Import packages

```
library(glmnet)
library(glmnetUtils)
library(caret)
```

CV Lasso Model

```
set.seed(310)
lasso_mod_cv <- cv.glmnet(AveragePrice~.-Date, alpha = 1,data = train)

plot(lasso_mod_cv) #training MSE as a function of lambda
```



Lambdas

```
lasso_mod_cv$lambda.min
```

```
## [1] 0.0003347769
```

```
lasso_mod_cv$lambda.1se
```

```
## [1] 0.004127286
```

Coefs

Lambda.min removed 6 variables and lambda.1se removed 15 variables.

```
coef(lasso_mod_cv, s = lasso_mod_cv$lambda.min)
```

```
## 66 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)               -7.525218e+01
## Total.Volume               .
## Small.Size                 .
```

```
## Medium.Size               -3.073567e-09
## Large.Size                -8.916901e-08
## Total.Bags                -4.066771e-09
## Small.Bags                -3.345209e-09
## Large.Bags                -5.685130e-08
## XLarge.Bags                1.446389e-06
## typeconventional          -4.890774e-01
## typeorganic                .
## year                       3.814377e-02
## regionAlbany               1.625916e-01
## regionAtlanta             -4.707852e-02
## regionBaltimoreWashington  1.179523e-01
## regionBoise               -6.773851e-02
## regionBoston               1.367828e-01
## regionBuffaloRochester     1.031282e-01
## regionCalifornia           1.875293e-03
## regionCharlotte            2.034452e-01
## regionChicago              1.564418e-01
## regionCincinnatiDayton    -1.844220e-01
## regionColumbus            -1.594799e-01
## regionDallasFtWorth       -3.208473e-01
## regionDenver              -1.817837e-01
## regionDetroit             -1.258542e-01
## regionGrandRapids          1.043767e-01
## regionGreatLakes          -6.062031e-02
## regionHarrisburgScranton   9.852479e-02
## regionHartfordSpringfield  4.000541e-01
## regionHouston             -3.449352e-01
## regionIndianapolis        -8.949542e-02
## regionJacksonville         1.107801e-01
## regionLasVegas            -2.269757e-02
## regionLosAngeles          -1.999221e-01
## regionLouisville          -1.340794e-01
## regionMiamiFtLauderdale    1.585677e-03
## regionMidsouth             .
## regionNashville           -1.881566e-01
## regionNewOrleansMobile    -9.804134e-02
## regionNewYork              3.169753e-01
## regionNortheast            1.935505e-01
## regionNorthernNewEngland   6.085436e-02
## regionOrlando              1.087695e-01
## regionPhiladelphia         2.322293e-01
## regionPhoenixTucson       -1.670267e-01
## regionPittsburgh          -4.044606e-02
## regionPlains               2.201746e-02
## regionPortland            -7.375729e-02
## regionRaleighGreensboro    1.513993e-01
## regionRichmondNorfolk     -1.132942e-01
## regionRoanoke             -1.418338e-01
## regionSacramento           2.086266e-01
## regionSanDiego            -9.369502e-03
## regionSanFrancisco         4.047523e-01
## regionSeattle              3.251125e-02
## regionSouthCarolina        .
```

```
## regionSouthCentral        -2.969763e-01
## regionSoutheast            .
## regionSpokane              2.738315e-02
## regionStLouis              2.641717e-02
## regionSyracuse             1.082322e-01
## regionTampa                1.063372e-02
## regionTotalUS             -2.232117e-02
## regionWest                -8.864719e-02
## regionWestTexNewMexico    -1.370976e-01
```

```
coef(lasso_mod_cv, s = lasso_mod_cv$lambda.1se)
```

```
## 66 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)               -6.857487e+01
## Total.Volume               .
## Small.Size                -4.870868e-09
## Medium.Size                .
## Large.Size                -4.298413e-08
## Total.Bags                 .
## Small.Bags                 .
## Large.Bags                -3.762326e-08
## XLarge.Bags                4.750552e-07
## typeconventional          -4.813550e-01
## typeorganic                .
## year                       3.482924e-02
## regionAlbany               1.366263e-01
## regionAtlanta             -1.714937e-02
## regionBaltimoreWashington  9.032962e-02
## regionBoise               -3.776561e-02
## regionBoston               1.095982e-01
## regionBuffaloRochester     7.661812e-02
## regionCalifornia           .
## regionCharlotte            1.773876e-01
## regionChicago              1.270684e-01
## regionCincinnatiDayton    -1.565493e-01
## regionColumbus            -1.295628e-01
## regionDallasFtWorth       -2.915353e-01
## regionDenver              -1.555933e-01
## regionDetroit             -9.434471e-02
## regionGrandRapids          7.994545e-02
## regionGreatLakes          -2.844909e-02
## regionHarrisburgScranton   7.289308e-02
## regionHartfordSpringfield  3.726825e-01
## regionHouston             -3.168936e-01
## regionIndianapolis        -6.109691e-02
## regionJacksonville         8.486892e-02
## regionLasVegas             .
## regionLosAngeles          -1.626937e-01
## regionLouisville          -1.052206e-01
## regionMiamiFtLauderdale    .
## regionMidsouth             .
## regionNashville           -1.595910e-01
## regionNewOrleansMobile    -6.753141e-02
```

```
## regionNewYork              2.879265e-01
## regionNortheast            1.608090e-01
## regionNorthernNewEngland   3.391874e-02
## regionOrlando              8.358801e-02
## regionPhiladelphia         2.055552e-01
## regionPhoenixTucson       -1.376403e-01
## regionPittsburgh          -1.154339e-02
## regionPlains               .
## regionPortland            -4.636691e-02
## regionRaleighGreensboro    1.247482e-01
## regionRichmondNorfolk     -8.466392e-02
## regionRoanoke             -1.128133e-01
## regionSacramento           1.819790e-01
## regionSanDiego             .
## regionSanFrancisco         3.783739e-01
## regionSeattle              3.517823e-03
## regionSouthCarolina        .
## regionSouthCentral        -2.654789e-01
## regionSoutheast            .
## regionSpokane              1.679750e-04
## regionStLouis              1.213098e-03
## regionSyracuse             8.145444e-02
## regionTampa                .
## regionTotalUS              .
## regionWest                -7.208522e-02
## regionWestTexNewMexico    -1.085385e-01
```

Best lambda

```
best_lam <- lasso_mod_cv$lambda.min
```

Predictions using best lambda

```
#train
lasso_pred_train <- predict(lasso_mod_cv, s = best_lam, newdata = train)

#test
lasso_pred_test <- predict(lasso_mod_cv, s = best_lam, newdata = test)
```

Model accuracy: RMSE and R2

There is not much of an overfitting issue since there is no big difference between the RSMEs. R2 is very low, the model didn't score too well.

```
# MSE train
RMSE(lasso_pred_train, train$AveragePrice)
```

```
## [1] 0.2659225
```

```r
# MSE train
RMSE(lasso_pred_test, test$AveragePrice)
```

```
## [1] 0.273549
```

```r
# R2
residual <- test$AveragePrice - lasso_pred_test
sse <- sum((residual)^2) #sum of sqr distances bwt actual & predicted
tss <- sum((test$AveragePrice-mean(test$AveragePrice))^2) #sum sqr dist bwt actual & their mean
r2 <- 1 - (sse/tss)
r2
```

```
## [1] 0.5492873
```

K-Means Clustering

Import packages

```r
library(tidyverse)
library(forcats)
library(zoo)
library(factoextra)
library(cluster)
library(scales)
library(mondate)
library(RColorBrewer)
library(ggplot2)
```

Import data

```r
avocados <- read.csv("/Users/kksizzle/Desktop/MGSC 310/final project/avocado.csv")
```

Basic data cleaning / organizing

```r
names(avocados)[5:7] <- c("Small.Size","Medium.Size","Large.Size")  #rename columns
avocados <- avocados[,-1]  # drop first column
avocados <- avocados[,-12] #drop year column
```

Subset data

```r
set.seed(310)
avo_indx <-  sample(1:nrow(avocados), 0.06*nrow(avocados), replace=FALSE)
avocados_subset <- avocados[avo_indx,]
```

```r
# create revenue column
avocados_subset$revenue <- avocados_subset$AveragePrice * avocados_subset$Total.Volume

# re order columns
avocados_subset <- avocados_subset[, c(2,3,6,4,9,5,7,8,13,10,11,12,1)]
dim(avocados_subset)
```

```
## [1] 1094   13
```

Perform K-means clustering with 3 clusters

```r
kmeans3 <- kmeans(avocados_subset[,-c(11:13)],
                  centers = 3,
                  nstart = 25)
```

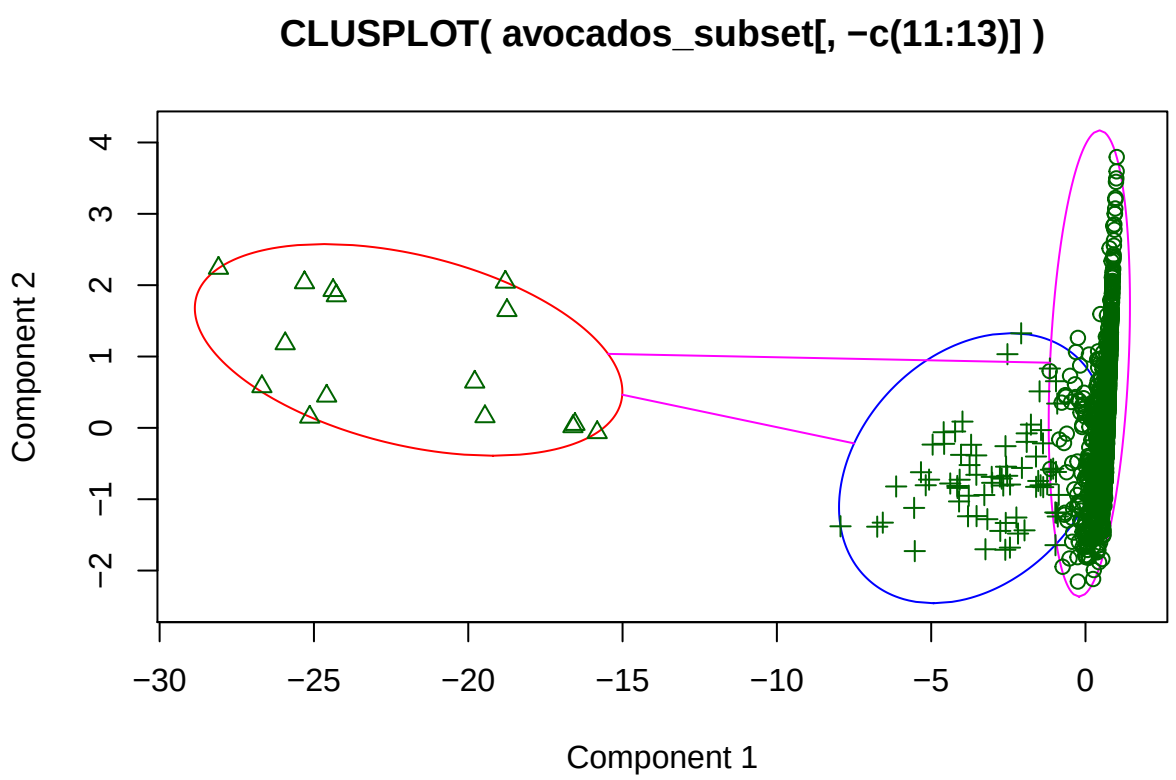
Average feature for each cluster

```r
kmeans3$centers
```

```
##   AveragePrice Total.Volume Large.Size  Small.Size Large.Bags Medium.Size
## 1     1.448779     232918.4   5626.838    67475.02   18040.48    86232.56
## 2     1.110000   31606403.8 903627.093 11383434.34 1635043.11 11361368.44
## 3     1.050833    4793211.2 146027.255  1866430.24  323670.00  1438908.27
##   Total.Bags Small.Bags    revenue XLarge.Bags
## 1   73579.55   54866.74   281248.7    672.3331
## 2 7957973.91 6221435.58 34647017.2 101495.2300
## 3 1341845.41  999344.00  4758280.6  18831.3953
```

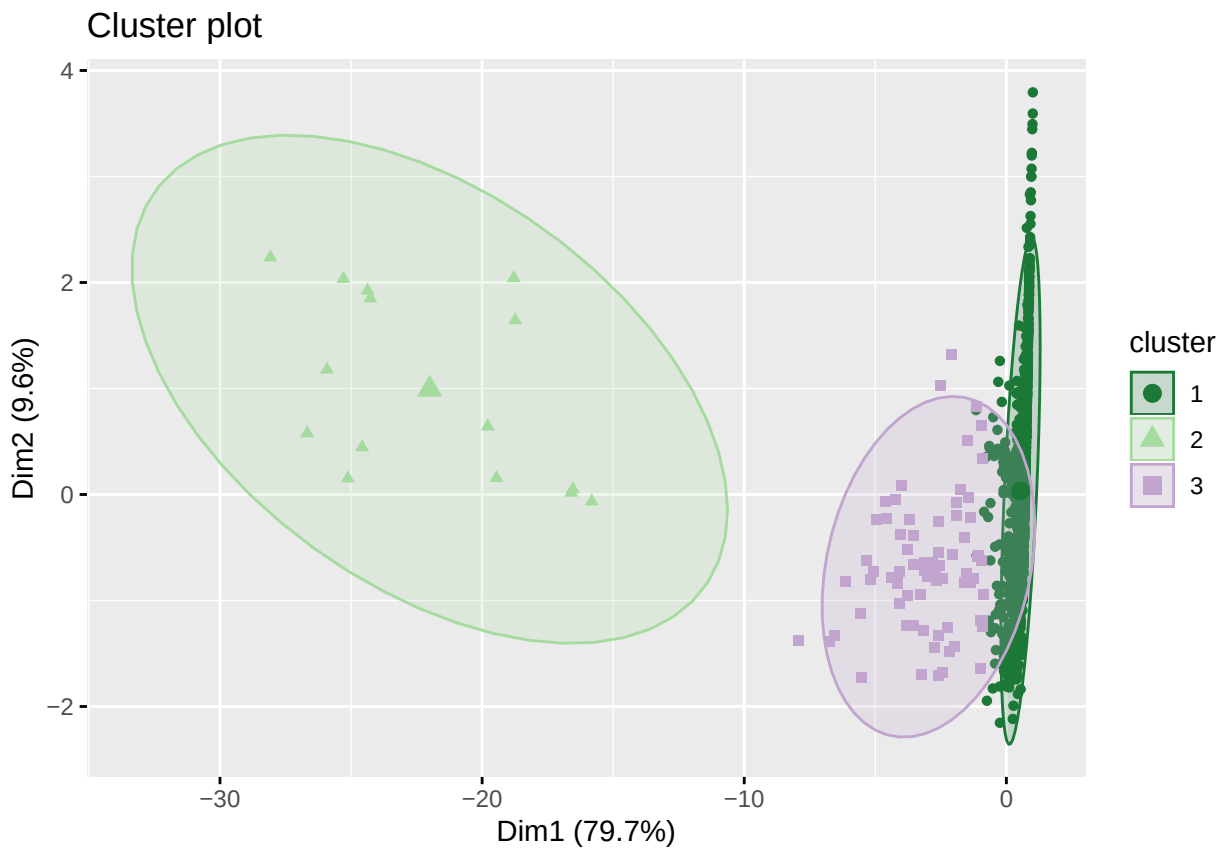
Cluster plot

```r
clusplot(avocados_subset[,-c(11:13)],
         kmeans3$cluster,
         color = TRUE,
         shade = FALSE)
```



Cluster plot

```
#colors for graphs
mycolor <-  c("#1B7837","#A6DBA0","#C2A5CF")

fviz_cluster(kmeans3, avocados_subset[,-c(11:13)], geom = "point", ellipse.type = "norm") +
  scale_color_manual(values = mycolor)+
  scale_fill_manual(values = mycolor)
```



Add cluster and quarter columns + lump regions

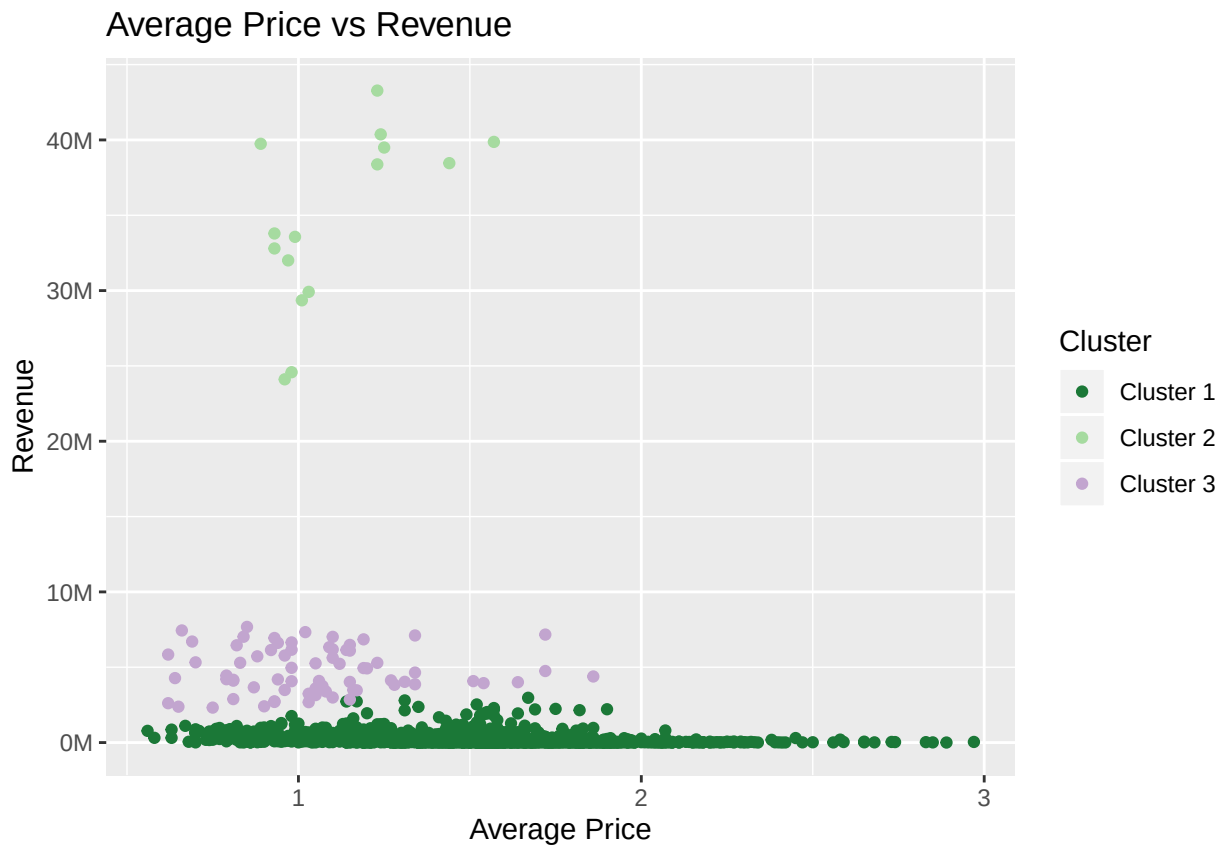
```
#add cluster columns
FinalDF <- data.frame(avocados_subset, factor(kmeans3$cluster))
FinalDF <- transform(FinalDF, cluster_name = paste("Cluster",kmeans3$cluster))

#add quarter columns
FinalDF$avocados_Q <- as.yearqtr(FinalDF$Date, format = "%Y-%m-%d")
FinalDF$avocados_Q <- quarters(FinalDF$avocados_Q)

#lump regions
FinalDF$region_simple <- fct_lump_min(FinalDF$region, min = 26)
```

Graphs

```
options(scipen = 999)
# function to turn axis to millions
ks <- function (x) { number_format(accuracy = 1,
                                   scale = 1/1000000,
                                   suffix = "M",
                                   big.mark = ",")(x) }
```

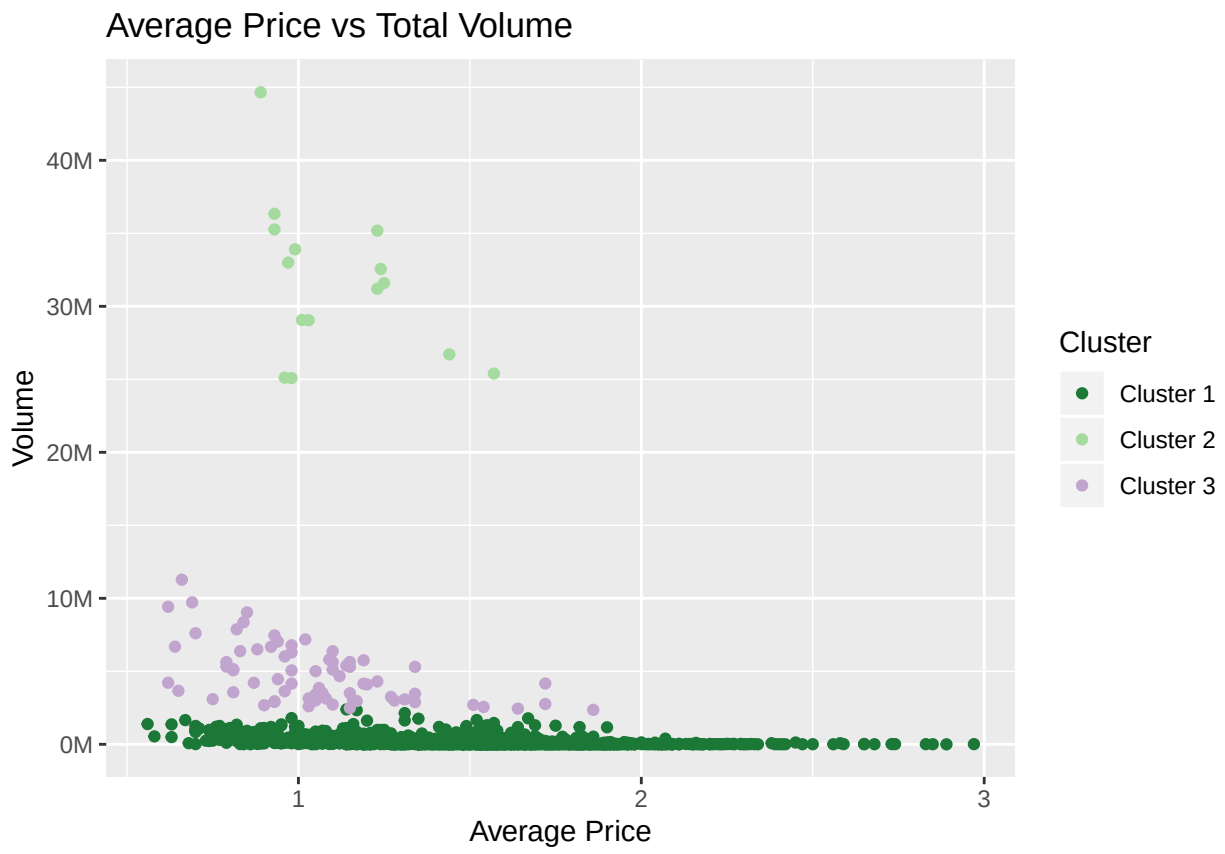
Average Price vs Revenue

```
ggplot(FinalDF, aes(AveragePrice, revenue, color = cluster_name)) +
  geom_point() +
  scale_color_manual(values = mycolor) +
  labs(x = "Average Price", y = "Revenue", title = "Average Price vs Revenue",
       color = "Cluster") +
  scale_y_continuous(labels = ks)
```



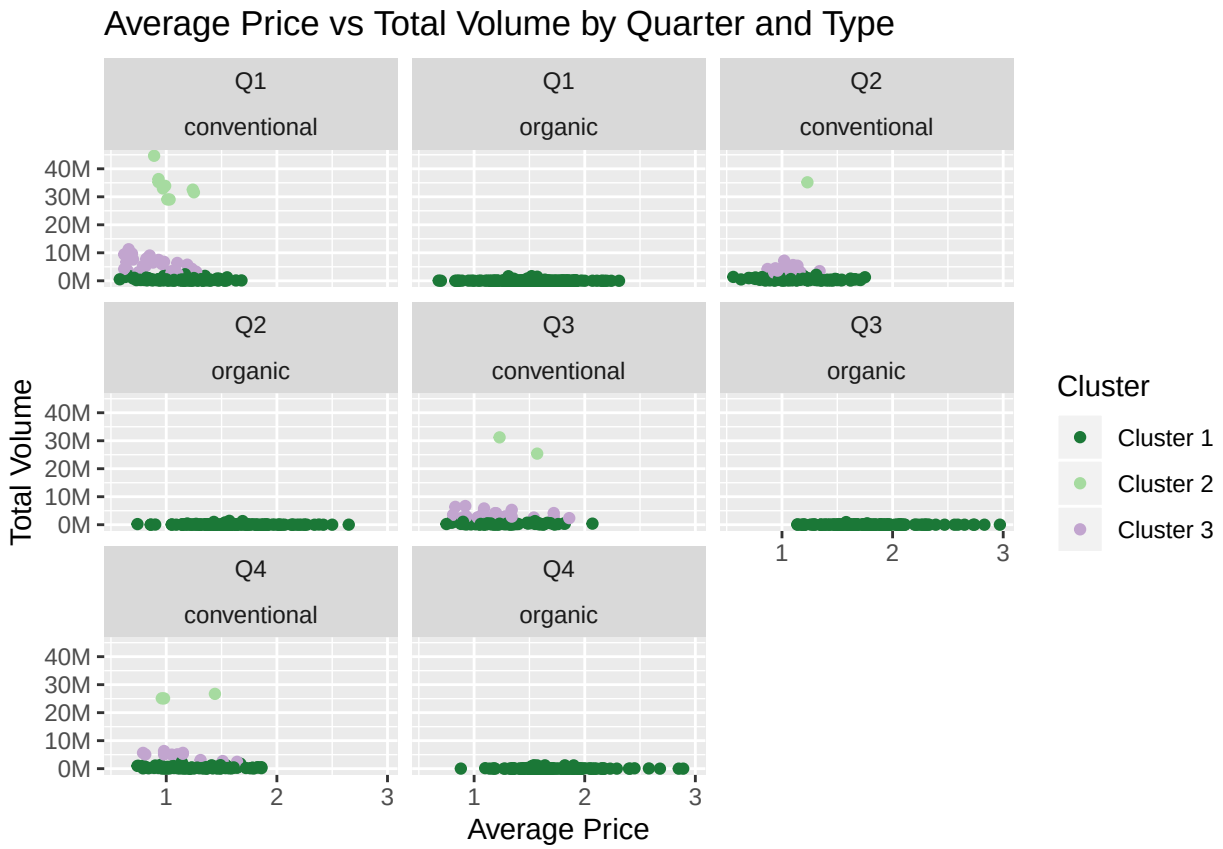
Average Price vs Total Volume

```
ggplot(FinalDF, aes(AveragePrice, Total.Volume, color = cluster_name)) +
  geom_point() +
  scale_color_manual(values = mycolor) +
  labs(x = "Average Price", y = "Volume", title = "Average Price vs Total Volume",
       color = "Cluster") +
  scale_y_continuous(labels = ks)
```



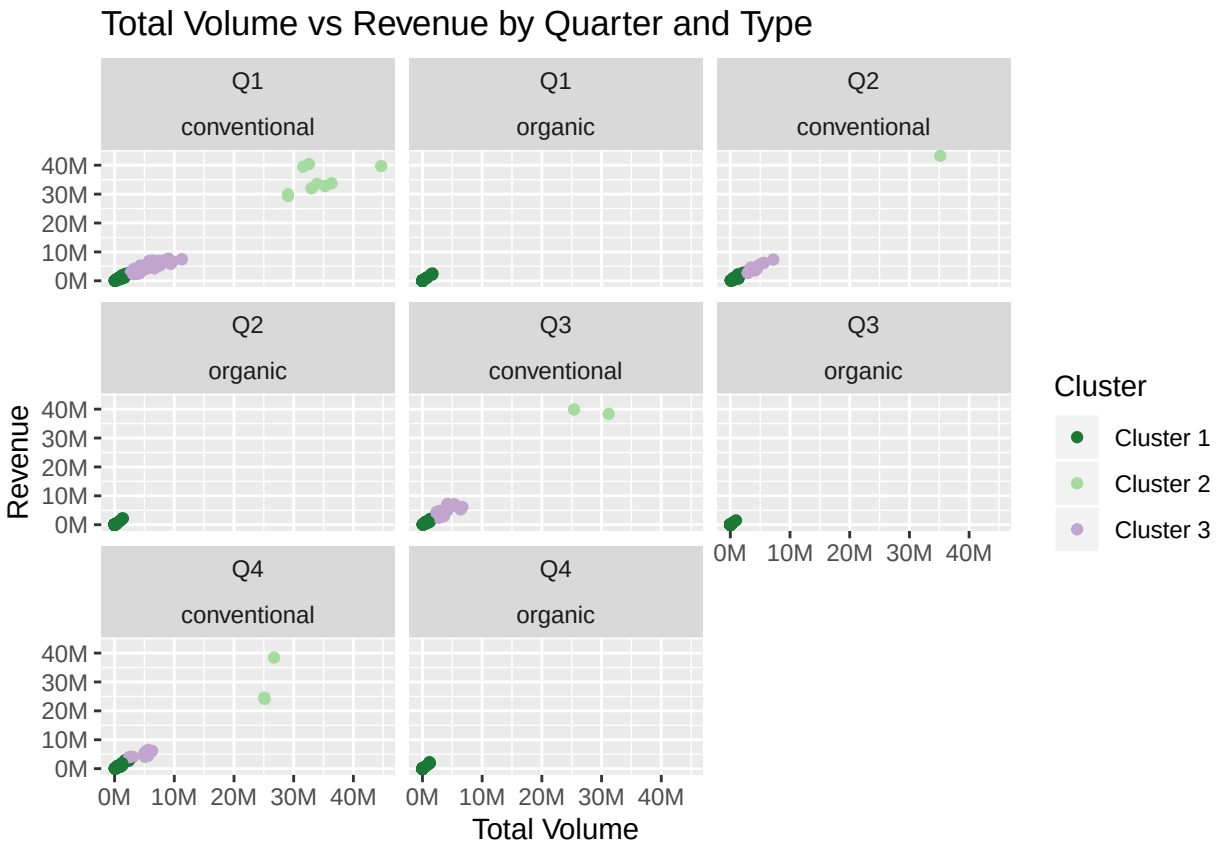
Average Price vs Total Volume by Quarter and Type

```
ggplot(FinalDF, aes(AveragePrice, Total.Volume, color = cluster_name)) +
  geom_point() +
  facet_wrap(~avocados_Q+type) +
  scale_color_manual(values = mycolor) +
  labs(x = "Average Price", y = "Total Volume",
       title = "Average Price vs Total Volume by Quarter and Type",
       color = "Cluster") +
  scale_y_continuous(labels = ks)
```



Total Volume vs Revenue by Quarter and Type

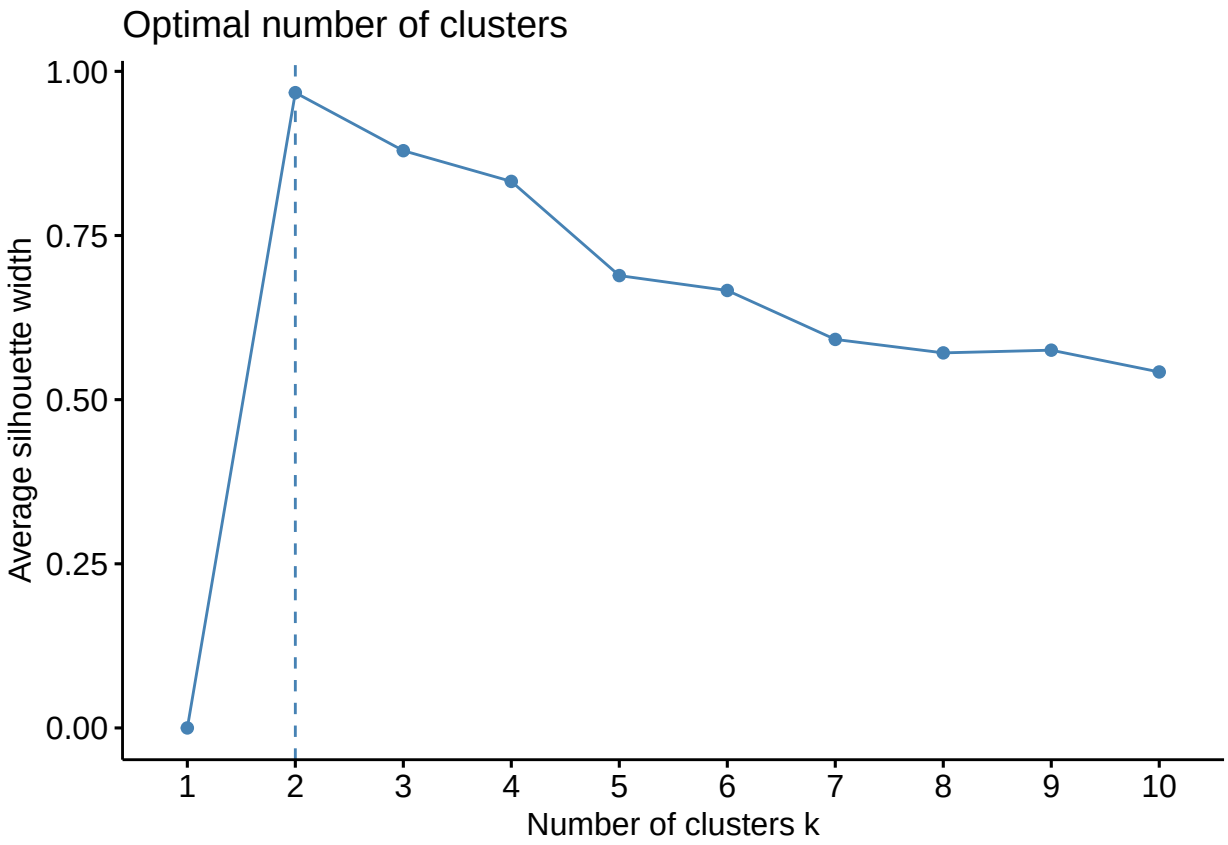
```
ggplot(FinalDF, aes(Total.Volume, revenue, color = cluster_name)) +
  geom_point() +
  facet_wrap(~avocados_Q+type) +
  scale_color_manual(values = mycolor) +
  labs(x = "Total Volume", y = "Revenue",
       title = "Total Volume vs Revenue by Quarter and Type",
       color = "Cluster") +
  scale_x_continuous(labels = ks) +
  scale_y_continuous(labels = ks)
```



Evaluating the model

The model received a silhouette score of about 0.85. This means that our clusters pretty cohesive (clusters members are similar) and separated (clusters are different from each other).

```
fviz_nbclust(avocados_subset[,-c(11:13)],
             kmeans,
             method="silhouette")
```



Random Forest Model

Import packages

```
library(randomForest)
library(randomForestExplainer)
```

Subsetting data

```
avocados_subset <- avocados_subset[!(avocados_subset$region == "West"),]
avocados_subset$region <- droplevels(avocados_subset$region, "West")

set.seed(310)
index2 <- sample(1:nrow(avocados_subset),size=0.75*nrow(avocados_subset),replace=FALSE)
mod4_rf_train <- avocados_subset[index2,]
mod4_rf_test <- avocados_subset[-index2,]
```

Random forest

```
set.seed(310)
mod4_rf <- randomForest(AveragePrice ~ . - Date,
                        data = mod4_rf_train,
                        mtry = 5,
                        ntree = 500,
                        importance = TRUE)
mod4_rf
```

```
##
## Call:
##  randomForest(formula = AveragePrice ~ . - Date, data = mod4_rf_train,      mtry = 5, ntree = 500, i
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 0.05688351
##                     % Var explained: 62.61
```

Model accuracy: RMSE adn R2

```
#train
preds_rf_train <- predict(mod4_rf)
RMSE(preds_rf_train, mod4_rf_train$AveragePrice)
```

```
## [1] 0.2385026
```

```
#test
preds_rf_test <- predict(mod4_rf, newdata = mod4_rf_test)
RMSE(preds_rf_test, mod4_rf_test$AveragePrice)
```

```
## [1] 0.2298039
```

```
#R2
residual2 <- mod4_rf_test$AveragePrice - preds_rf_test
sse2 <- sum((residual2)^2) #sum of sqr distances bwt actual & predicted
tss2 <- sum((mod4_rf_test$AveragePrice-mean(mod4_rf_test$AveragePrice))^2) #sum of sqr distances bwt ac
r22 <- 1 - (sse2/tss2)
r22
```

```
## [1] 0.7023052
```

Visualizing Random Forest

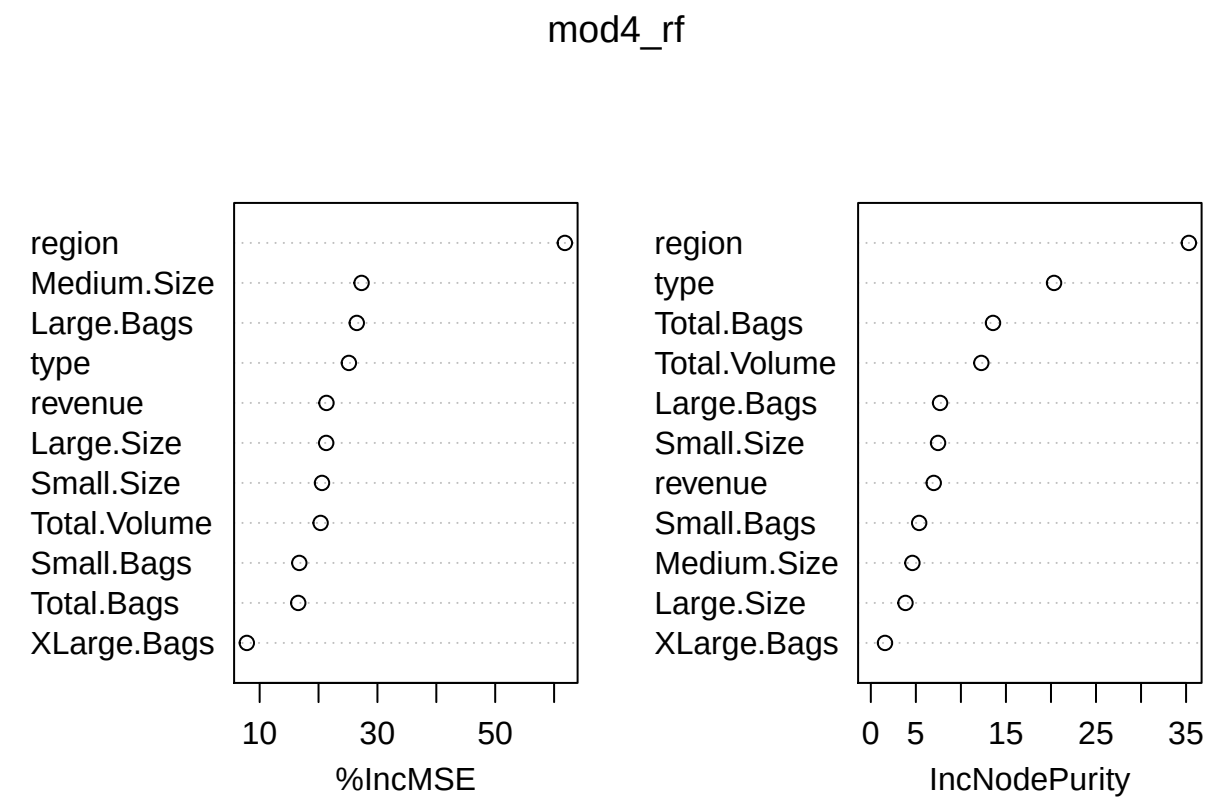
Importance variable chart

```
importance(mod4_rf)
```

```
##                %IncMSE IncNodePurity
## Total.Volume 20.351905     12.268694
## Large.Size   21.300398      3.845804
## Small.Size   20.586241      7.462916
## Large.Bags   26.495003      7.688424
## Medium.Size  27.306424      4.615749
## Total.Bags   16.551566     13.563949
## Small.Bags   16.734393      5.367196
## revenue      21.342137      6.984168
## XLarge.Bags   7.824467      1.572677
## type         25.151084     20.341343
## region       61.830193     35.310781
```

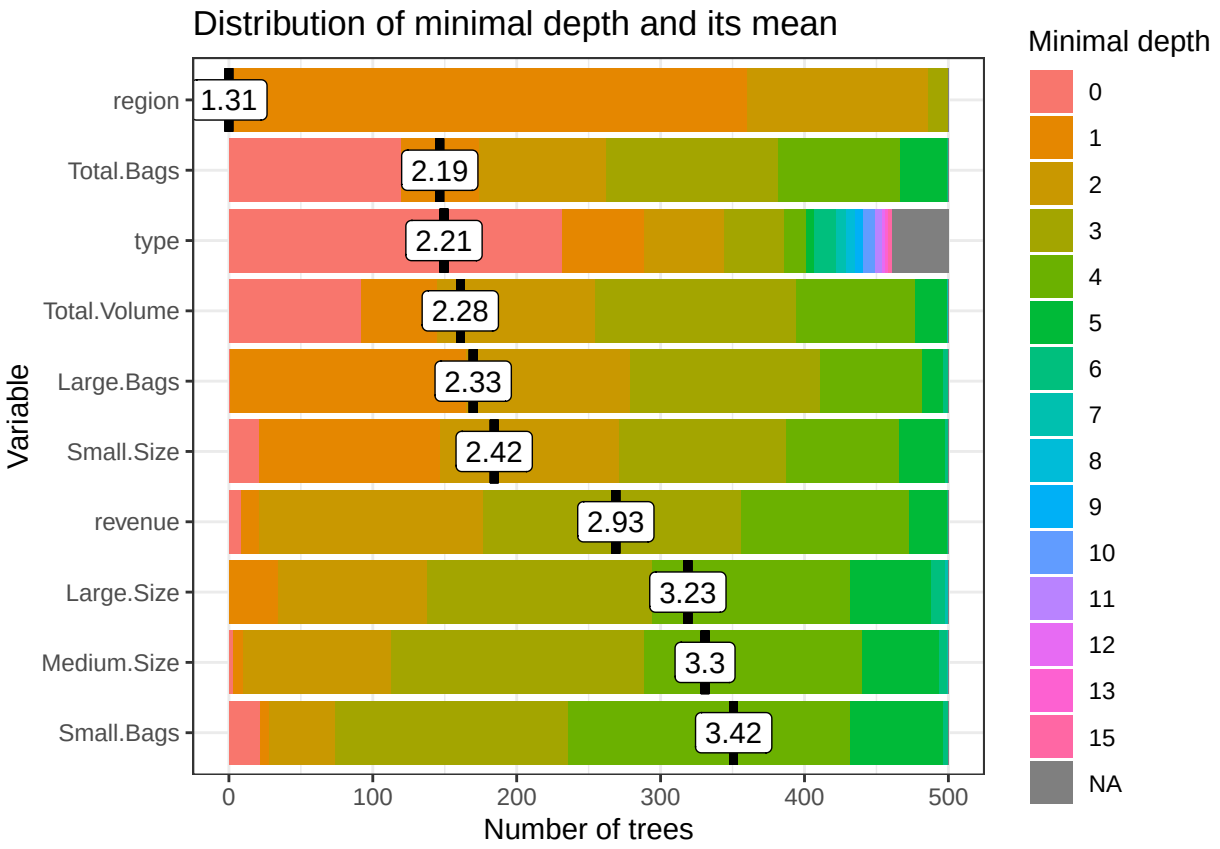
Importance variable plot

```
varImpPlot(mod4_rf)
```



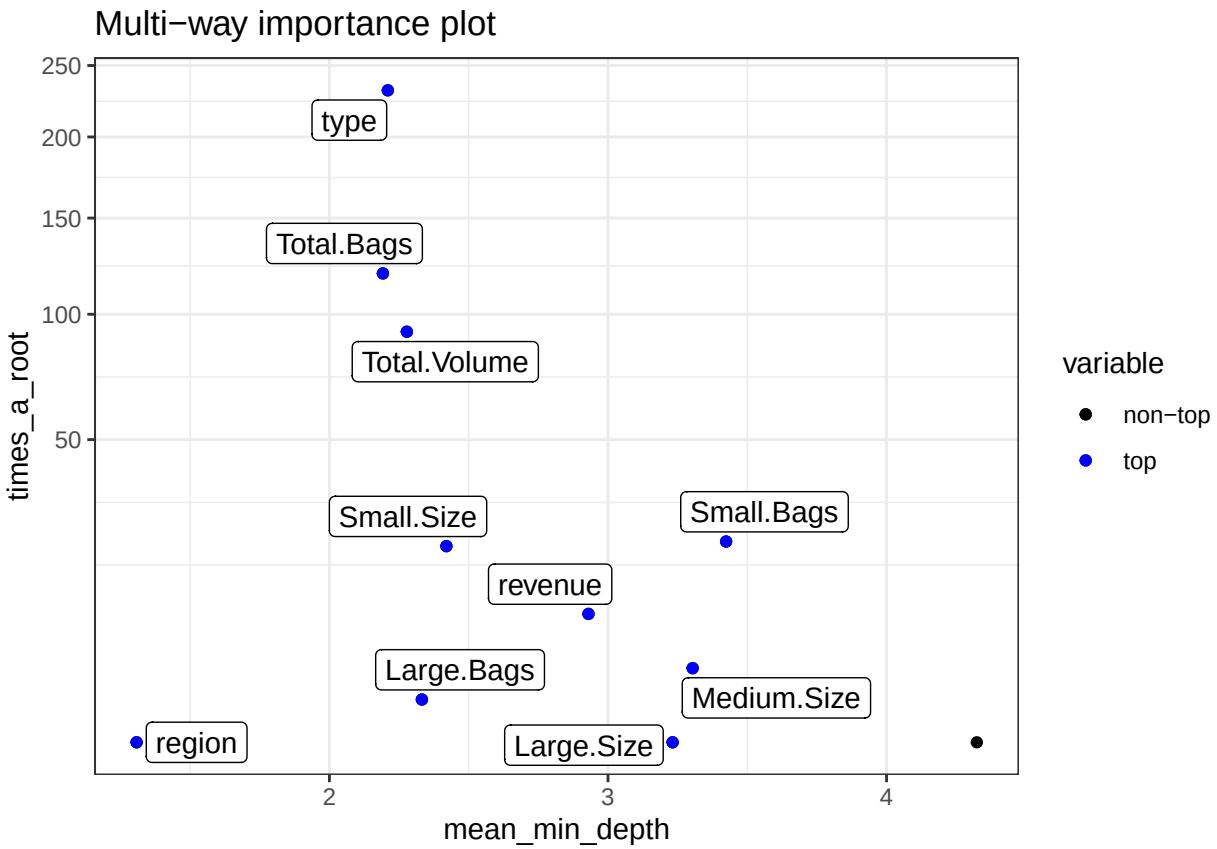
Plot min depth distribution

```
plot_min_depth_distribution(mod4_rf)
```



Plot variable two-way importance measure

```
plot_multi_way_importance(mod4_rf)
```


Multi−way importance plot
250
200
150
100
50
times_a_root
type
Total.Bags
Total.Volume
Small.Size
Small.Bags
revenue
Large.Bags
Medium.Size
region
Large.Size
2
3
4
mean_min_depth
variable
non−top
top