DATA 622: Machine Learning and Big Data HW1

Gabriel Campos

Last edited March 10, 2024

Packages

```
library(readr)
library(tidyverse)
library(tidymodels)
library(psych)
library(caret)
library(rpart)
library(rpart.plot)
library(corrplot)
library(RColorBrewer)
library(labelled)
library(ggplot2)
library(ggforce)
library(kableExtra)
library(gridExtra)
library(Metrics)
```

Instructions

Exploratory analysis and essay

Pre-work

1. Visit the following website and explore the range of sizes of this dataset (from 100 to 5 million records): https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/ or (new) https://www.kaggle.com/datasets
2. Select 2 files to download Based on your computer's capabilities (memory, CPU), select 2 files you can handle (recommended one small, one large)
3. Download the files
4. Review the structure and content of the tables, and think about the data sets (structure, size, dependencies, labels, etc)
5. Consider the similarities and differences in the two data sets you have downloaded
6. Think about how to analyze and predict an outcome based on the datasets available
7. Based on the data you have, think which two machine learning algorithms presented so far could be used to analyze the data

Deliverable

1. Essay (minimum 500 word document) Write a short essay explaining your selection of algorithms and how they relate to the data and what you are trying to do
2. Exploratory Analysis (**EDA**) using R or Python (submit code + errors + analysis as notebook or copy/paste to document) Explore how to analyze and predict an outcome based on the data available. This will be an exploratory exercise, so feel free to show errors and warnings that raise during the analysis. Test the code with both datasets selected and compare the results.

Answer questions such as:

1. Are the columns of your data correlated?
2. Are there labels in your data? Did that impact your choice of algorithm?
3. What are the pros and cons of each algorithm you selected?
4. How your choice of algorithm relates to the datasets (was your choice of algorithm impacted by the datasets you chose)?
5. Which result will you trust if you need to make a business decision?
6. Do you think an analysis could be prone to errors when using too much data, or when using the least amount possible?
7. How does the analysis between data sets compare?

Develop your exploratory analysis of the data and the essay in the following 2 weeks.

DATA

```
url<-"https://raw.githubusercontent.com/GitableGabe/Data624_Data/main/"
df_1k <- read.csv(paste0(url,"1000%20Sales%20Records.csv"))
df_100k <- read.csv(paste0(url,"100000%20Sales%20Records.csv"))
```

EDA

Familiarization with Sales datasets extracted from excelbi analytics requires understanding of dataset composition, dimensions, column types, `NA` or `Null` value count, etc.

Data Composition

```
str(df_1k)
```

```
## 'data.frame':    1000 obs. of  14 variables:
##  $ Region        : chr  "Middle East and North Africa" "North America" "Middle East and North Africa"
##  $ Country       : chr  "Libya" "Canada" "Libya" "Japan" ...
##  $ Item.Type     : chr  "Cosmetics" "Vegetables" "Baby Food" "Cereal" ...
##  $ Sales.Channel : chr  "Offline" "Online" "Offline" "Offline" ...
##  $ Order.Priority: chr  "M" "M" "C" "C" ...
##  $ Order.Date    : chr  "10/18/2014" "11/7/2011" "10/31/2016" "4/10/2010" ...
##  $ Order.ID      : int  686800706 185941302 246222341 161442649 645713555 683458888 679414975 208630
```

```
##  $ Ship.Date     : chr  "10/31/2014" "12/8/2011" "12/9/2016" "5/12/2010" ...
##  $ Units.Sold    : int  8446 3018 1517 3322 9845 9528 2844 7299 2428 4800 ...
##  $ Unit.Price    : num  437.2 154.06 255.28 205.7 9.33 ...
##  $ Unit.Cost     : num  263.33 90.93 159.42 117.11 6.92 ...
##  $ Total.Revenue : num  3692591 464953 387260 683335 91854 ...
##  $ Total.Cost    : num  2224085 274427 241840 389039 68127 ...
##  $ Total.Profit  : num  1468506 190526 145420 294296 23726 ...
```

```
str(df_100k)
```

```
## 'data.frame':    100000 obs. of  14 variables:
##  $ Region        : chr  "Middle East and North Africa" "Central America and the Caribbean" "Sub-Saha
##  $ Country       : chr  "Azerbaijan" "Panama" "Sao Tome and Principe" "Sao Tome and Principe" ...
##  $ Item.Type     : chr  "Snacks" "Cosmetics" "Fruits" "Personal Care" ...
##  $ Sales.Channel : chr  "Online" "Offline" "Offline" "Online" ...
##  $ Order.Priority: chr  "C" "L" "M" "M" ...
##  $ Order.Date    : chr  "10/8/2014" "2/22/2015" "12/9/2015" "9/17/2014" ...
##  $ Order.ID      : int  535113847 874708545 854349935 892836844 129280602 473105037 754046475 772153
##  $ Ship.Date     : chr  "10/23/2014" "2/27/2015" "1/18/2016" "10/12/2014" ...
##  $ Units.Sold    : int  934 4551 9986 9118 5858 1149 7964 6307 8217 2758 ...
##  $ Unit.Price    : num  152.58 437.2 9.33 81.73 668.27 ...
##  $ Unit.Cost     : num  97.44 263.33 6.92 56.67 502.54 ...
##  $ Total.Revenue : num  142510 1989697 93169 745214 3914726 ...
##  $ Total.Cost    : num  91009 1198415 69103 516717 2943879 ...
##  $ Total.Profit  : num  51501 791282 24066 228497 970846 ...
```

```
kable(as.data.frame(table(df_1k$Region)) %>% arrange(desc(Freq)),
      caption = "Frequency Region df_1k")
```

Table 1: Frequency Region df_1k

Var1	Freq
Europe	267
Sub-Saharan Africa	262
Middle East and North Africa	138
Asia	136
Central America and the Caribbean	99
Australia and Oceania	79
North America	19

```
kable(as.data.frame(table(df_100k$Region)) %>% arrange(desc(Freq)),
      caption = "Frequency Region df_100k")
```

Table 2: Frequency Region df_100k

Var1	Freq
Sub-Saharan Africa	26019
Europe	25877
Asia	14547

Var1	Freq
Middle East and North Africa	12580
Central America and the Caribbean	10731
Australia and Oceania	8113
North America	2133

```
kable(as.data.frame(table(df_1k$Item.Type )) %>% arrange(desc(Freq)),
      caption = "Frequency Item.Type df_1k")
```

Table 3: Frequency Item.Type df_1k

Var1	Freq
Beverages	101
Vegetables	97
Office Supplies	89
Baby Food	87
Personal Care	87
Snacks	82
Cereal	79
Clothes	78
Meat	78
Household	77
Cosmetics	75
Fruits	70

```
kable(as.data.frame(table(df_100k$Item.Type )) %>% arrange(desc(Freq)),
            caption = "Frequency Item Type 100k")
```

Table 4: Frequency Item Type 100k

Var1	Freq
Office Supplies	8426
Cereal	8421
Baby Food	8407
Cosmetics	8370
Personal Care	8364
Meat	8320
Snacks	8308
Clothes	8304
Vegetables	8282
Household	8278
Fruits	8262
Beverages	8258

```
kable(as.data.frame(table(df_1k$Sales.Channel )) %>% arrange(desc(Freq)),
      caption = "Frequency Sales Channel 1k")
```

Table 5: Frequency Sales Channel 1k

Var1	Freq
Offline	520
Online	480

```
kable(as.data.frame(table(df_100k$Sales.Channel )) %>% arrange(desc(Freq)),
      caption = "Frequency Sales Channel 100k")
```

Table 6: Frequency Sales Channel 100k

Var1	Freq
Online	50054
Offline	49946

```
var_label(df_1k)
```

```
## $Region
## NULL
##
## $Country
## NULL
##
## $Item.Type
## NULL
##
## $Sales.Channel
## NULL
##
## $Order.Priority
## NULL
##
## $Order.Date
## NULL
##
## $Order.ID
## NULL
##
## $Ship.Date
## NULL
##
## $Units.Sold
## NULL
##
## $Unit.Price
## NULL
##
## $Unit.Cost
## NULL
##
```

```
## $Total.Revenue
## NULL
##
## $Total.Cost
## NULL
##
## $Total.Profit
## NULL
```

```
var_label(df_100k)
```

```
## $Region
## NULL
##
## $Country
## NULL
##
## $Item.Type
## NULL
##
## $Sales.Channel
## NULL
##
## $Order.Priority
## NULL
##
## $Order.Date
## NULL
##
## $Order.ID
## NULL
##
## $Ship.Date
## NULL
##
## $Units.Sold
## NULL
##
## $Unit.Price
## NULL
##
## $Unit.Cost
## NULL
##
## $Total.Revenue
## NULL
##
## $Total.Cost
## NULL
##
## $Total.Profit
## NULL
```

```r
# Dimensions
dim_1k_tmp<-dim(df_1k)
dim_100k_tmp<-dim(df_100k)
# Class
class_1k_tmp<-sapply(df_1k,class)
class_100k_tmp<-sapply(df_100k,class)



column_name_1k_tmp <- "Order.ID"

# Count the number of duplicates in the specified column
num_duplicates_1k_tmp <- sum(duplicated(df_1k[[column_name_1k_tmp]]) |
                             duplicated(df_1k[[column_name_1k_tmp]],
                                        fromLast = TRUE))

column_name_100k_tmp <- "Order.ID"

# Count the number of duplicates in the specified column
num_duplicates_100k_tmp <- sum(duplicated(df_100k[[column_name_100k_tmp]]) |
                             duplicated(df_100k[[column_name_100k_tmp]],
                                        fromLast = TRUE))


na_null_cnt_tmp<-(sum(colSums(is.na(df_1k) | is.null(df_1k)))+
                    sum(colSums(is.na(df_100k) | is.null(df_100k))))

region_tmp<-unique(df_1k$Region)
country_len_tmp<-length(unique(df_1k$Country))
```

The dataset of size 1000 is stored to `df_1k` and the dataset size 100,000 is stored to `df_100k`

- `df_1k` dimensions is 1000 rows and 14 columns.
- `df_100k` dimensions is 100000 rows and 14 columns.
- The column types for df_1k are character, character, character, character, character, character, integer, character, integer, numeric, numeric, numeric, numeric, numeric
- The column types for df_100k are character, character, character, character, character, character, integer, character, integer, numeric, numeric, numeric, numeric, numeric
- Notable categories include
  - `Order.Date` and `Ship.Date` the only date valued columns, but set to type `chr` and may need converting.
  - `Order.ID` is compose of unique values with 0 duplicates found in the `df_1k` data and 0 found in the `df_100k` data.
  - `Region` and `Country` both of which define location
  - `Item.Type` for type of item sold.
  - `Sales.Channel` defines sales method as an online or offline purchase, or e-purchase vs in-store.
  - `Order.Priority` which has a ranking of severity.
  - Attributes labeled with `Total` that are calculated values.
  - Using the `length()` functions we see that 185 countries are listed in the data.
- Using the `table` function we see: -Of the `Regions` listed `Sub-Saharan Africa` and `Europe` is most frequented.

- For `df_1k Beverages` and `Vegetables` is most frequented, however with `df_100k Office Supplies` and `Cereals` is.
- For `df_1k` more purchases are done `Offline` while for `df_100k` more is done `Online` Albiet by a small margin in both cases.

With respect to dependencies, the formulas below highlight the dependency that exists with calculated variables with the label `Total` in there Attribute name.

$Total.Cost = Units.Sold \times Unit.Cost$ making Total.Cost dependent on Units.Sold and Unit Cost
$Total.Revenue = Units.Sold \times Unit.Price$ making Total.Revenue dependent on Units.Sold and Unit.Price
$Total.Profit = Total.Revenue - Total.Cost$ making the subsequent totals above the dependent variables for Total.Profit

The Order.Priority have a dependency based on ranking of M, C, H, L Which is Critical, High, Medium, Low in ascending order.

Date values are dependent in interpretation, with calculation of `Order.Date` and `Ship.Date` being a factor of performance or timeliness.

Data Tranformation

```
df_1k[['Order.Date']] <- as.Date(df_1k[['Order.Date']], "%m/%d/%Y")
df_1k[['Ship.Date']] <- as.Date(df_1k[['Ship.Date']], "%m/%d/%Y")

df_100k[['Order.Date']] <- as.Date(df_100k[['Order.Date']], "%m/%d/%Y")
df_100k[['Ship.Date']] <- as.Date(df_100k[['Ship.Date']], "%m/%d/%Y")

df_1k$Order.Priority <- as.factor(df_1k$Order.Priority)
df_100k$Order.Priority <- as.factor(df_100k$Order.Priority)
```

The most obvious transformations were the date values as noted in *EDA* and factoring the categories in `Order.Priority`

```
df_1k$Sales.Channel <- as.factor(df_1k$Sales.Channel)
df_100k$Sales.Channel <- as.factor(df_100k$Sales.Channel)
df_1k$Item.Type <- as.factor(df_1k$Item.Type)
df_100k$Item.Type <- as.factor(df_100k$Item.Type)
df_1k$Region <- as.factor(df_1k$Region)
df_100k$Region <- as.factor(df_100k$Region)
df_1k$Country <- as.factor(df_1k$Country)
df_100k$Country <- as.factor(df_100k$Country)
```

`Sales.Channel`,`Item.Type` and `Region` were also logical choices, considering the amount of unique values for `Country` and the nature of its relationship with Region, I believe if I make a model with Region, Country would be excluded. Order.ID are just arbitrary, chronological or incremented numbers therefor it was not set as a factor.

```
levels(df_1k$Region)
```

```
## [1] "Asia"                           "Australia and Oceania"
## [3] "Central America and the Caribbean" "Europe"
## [5] "Middle East and North Africa"      "North America"
## [7] "Sub-Saharan Africa"
```

Correlation and Skewness

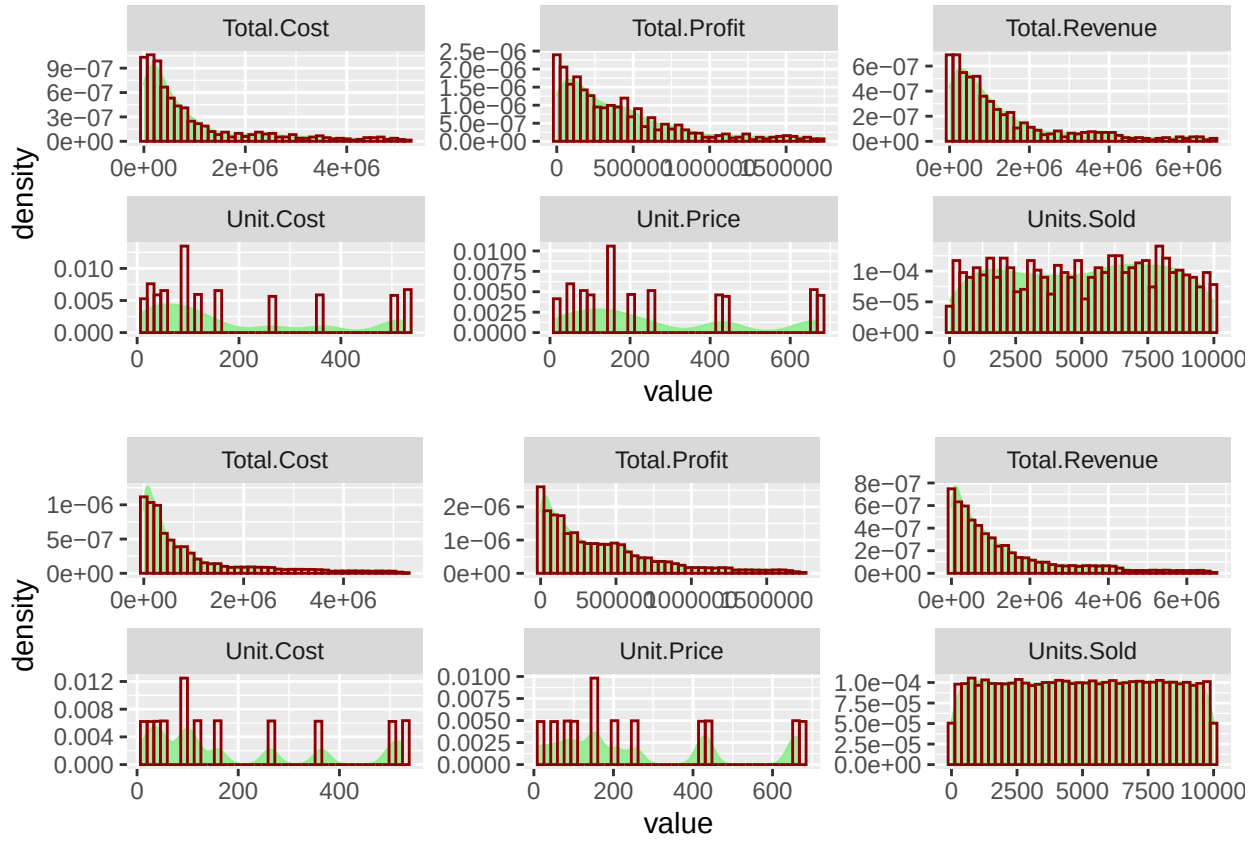
```
describe(df_1k%>%
           dplyr::select(contains("Unit") | contains("Total"))) %>%
  dplyr::select(c(mean,sd,min,max,range,se,skew))
```

```
##                      mean         sd     min        max      range       se
## Units.Sold       5053.99    2901.38   13.00    9998.00    9985.00    91.75
## Unit.Price        262.11     216.02    9.33     668.27     658.94     6.83
## Unit.Cost         184.97     175.29    6.92     524.96     518.04     5.54
## Total.Revenue 1327321.84 1486514.56 2043.25 6617209.54 6615166.29 47007.72
## Total.Cost     936119.23 1162570.75 1416.75 5204978.40 5203561.65 36763.72
## Total.Profit   391202.61  383640.19  532.61 1726181.36 1725648.75 12131.77
##                skew
## Units.Sold    -0.05
## Unit.Price     0.79
## Unit.Cost      0.95
## Total.Revenue  1.63
## Total.Cost     1.79
## Total.Profit   1.40
```

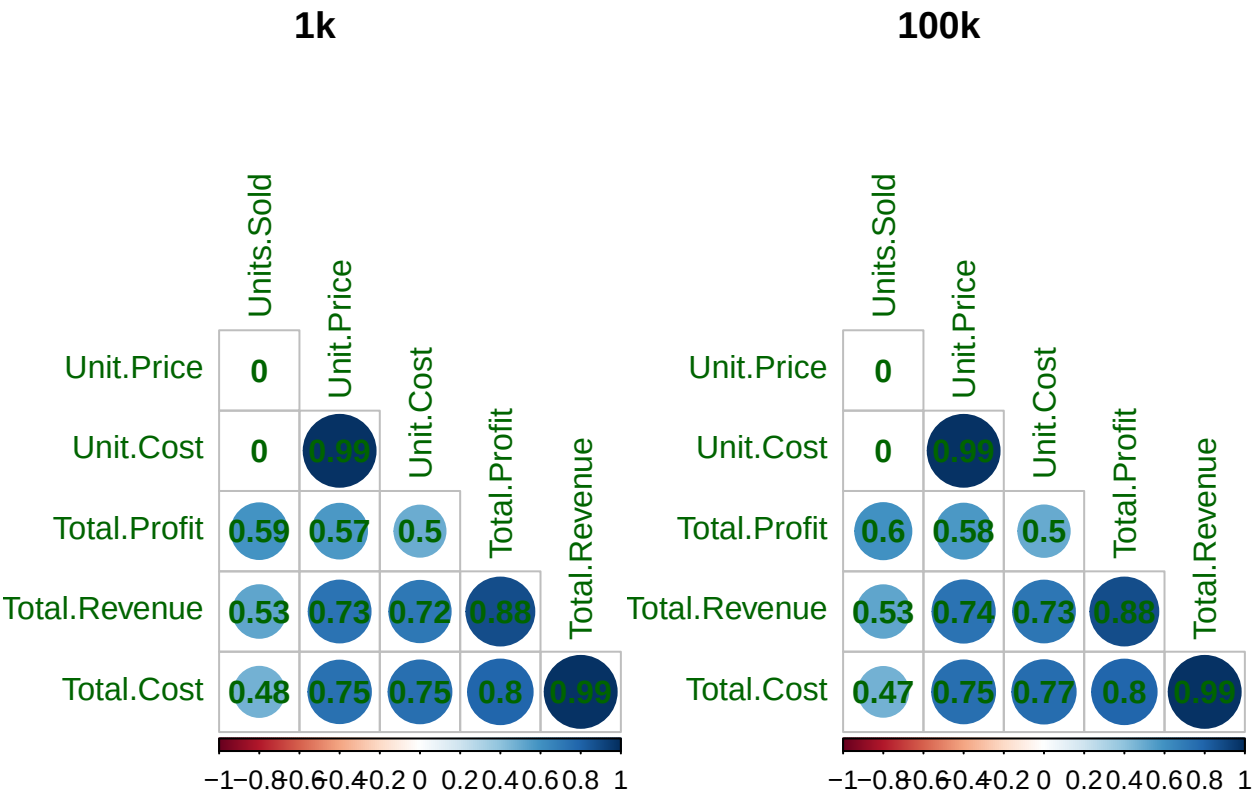
```
describe(df_100k%>%
           dplyr::select(contains("Unit") | contains("Total"))) %>%
  dplyr::select(c(mean,sd,min,max,range,se,skew))
```

```
##                      mean         sd   min        max      range      se skew
## Units.Sold       5001.45    2884.58  1.00   10000.00    9999.00    9.12 0.00
## Unit.Price        266.70     216.94  9.33     668.27     658.94    0.69 0.73
## Unit.Cost         188.02     175.71  6.92     524.96     518.04    0.56 0.89
## Total.Revenue 1336066.73 1471767.59 18.66 6682700.00 6682681.34 4654.14 1.57
## Total.Cost     941975.49 1151828.43 13.84 5249075.04 5249061.20 3642.40 1.74
## Total.Profit   394091.24  379598.60  4.82 1738700.00 1738695.18 1200.40 1.28
```

```
plot_numeric_1k<-df_1k%>%
      dplyr::select(contains("Unit") | contains("Total")) %>%
  gather(variable, value, 1:6) %>%
  ggplot(aes(value)) +
    facet_wrap(~variable, scales = "free") +
    geom_density(fill = "lightgreen", alpha=0.9, color="lightgreen") +
    geom_histogram(aes(y=after_stat(density)), alpha=0.2, fill = "lightblue",
                   color="darkred", position="identity", bins = 40)

plot_numeric_100k<-df_100k%>%
      dplyr::select(contains("Unit") | contains("Total")) %>%
  gather(variable, value, 1:6) %>%
  ggplot(aes(value)) +
    facet_wrap(~variable, scales = "free") +
    geom_density(fill = "lightgreen", alpha=0.9, color="lightgreen") +
    geom_histogram(aes(y=after_stat(density)), alpha=0.2, fill = "pink",
                   color="darkred", position="identity", bins = 40)

grid.arrange(plot_numeric_1k,plot_numeric_100k,ncol=1)
```




```
par(mfrow = c(1, 2), mar = c(0, 0, 3, 0))
plot_corr_1k <- cor(df_1k %>%
                      dplyr::select(contains("Unit") | contains("Total")))
corrplot(plot_corr_1k, tl.col = 'darkgreen', diag = FALSE, type = "lower",
         order = "hclust", addCoef.col = "darkgreen",
         title = "1k",mar=c(0,0,1,0))

# Plot correlation for df_100k
plot_corr_100k <- cor(df_100k %>%
                        dplyr::select(contains("Unit") | contains("Total")))
corrplot(plot_corr_100k, tl.col = 'darkgreen', diag = FALSE, type = "lower",
         order = "hclust", addCoef.col = "darkgreen",
         title = "100k",mar=c(0,0,1,0))
```



Skewness is a measure of symmetry, therefore the values near zero, despite one being negative, did not particularly stand out, however for both size data sets, Total - Revenue, Cost and Profit all are right skewed. Skewness = 0: perfect symmetry. Skewness < 0: Negatively is left skewed or has a tail. Skewness > 0: Positive is right skewed or has a right tail.

Concern is not too big with respect to these values as for our model I can try to normalize it as much as possible.

Correlation does more than just support the obvious relationships noted earlier, rather it help identify if we have multicollinearity. Multicollinearity occurs when two or more independent variables in a data frame have a high correlation with one another, and can cause issues with stability and size of an estimated regression coefficient, which in turn makes unreliable inferences for our predictor variables.

Of our variables, Unit.Cost and Total.Profit have the highest correlation, while Unit.-Cost,Price and Sold show the weakest. The way to interpret the correlation is understanding that the higher the absolute value of a correlation coefficient is, the stronger the relationship.

Because I suspect multicollinearity, I've chosen to not create my second model off the numeric values, rather I am opting to make a decision tree using one of the categorical values, whose variables I've set to factors earlier. For my first I plan to do a simple regression but I suspect normalizing it will not impact the data much.

Model Selection and logic

Simple Linear regression

For my first model I will choose a simple linear regression after normalizing the data.

Normalization

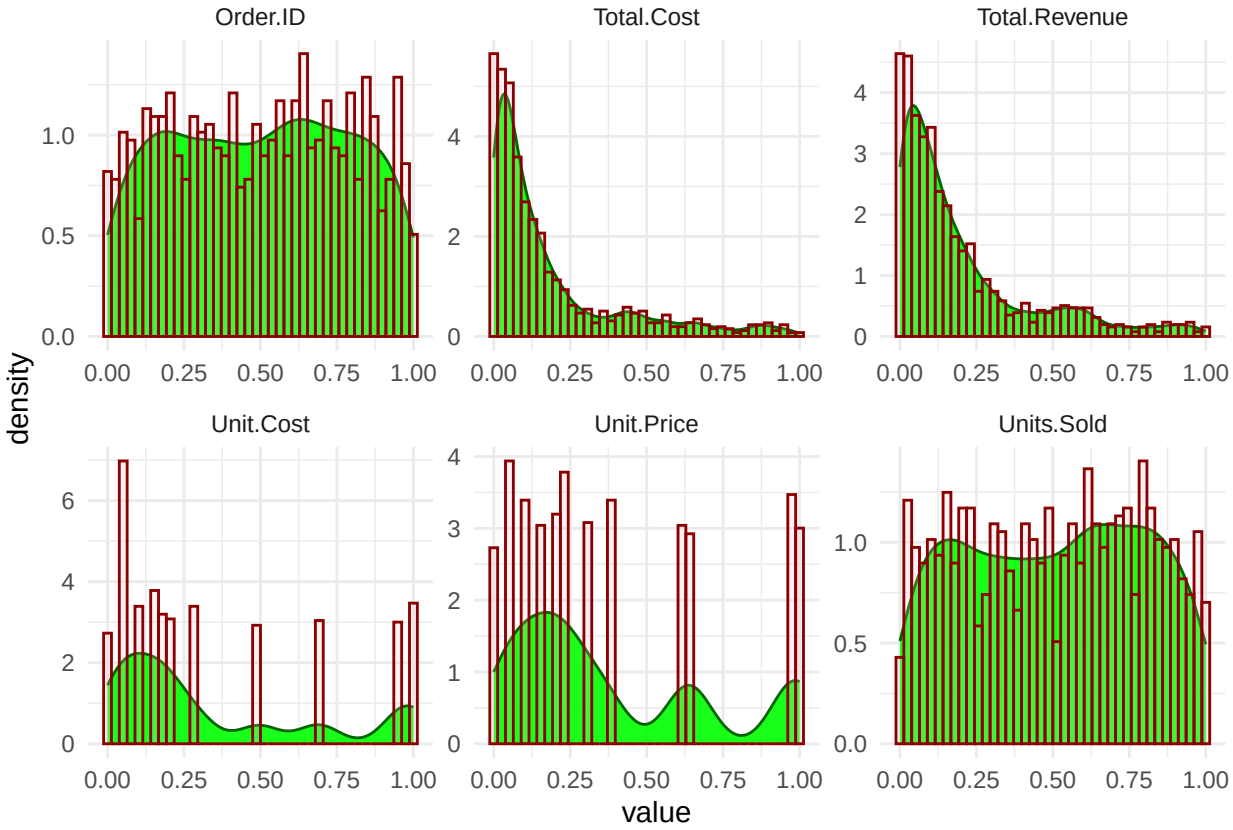
Statology provides a great walk through for normalization. Normalization ensures all variables contribute equally to a model vs having one contribute more because of its value.

```
# Function for normalization
min_max_norm <- function(x) {
    (x - min(x)) / (max(x) - min(x))
}

# Run function using lapply and only with the numeric values
norm_1k <- as.data.frame(lapply(df_1k %>%
                                  keep(is.numeric) , min_max_norm))

norm_100k <- as.data.frame(lapply(df_100k %>%
                                    keep(is.numeric) , min_max_norm))
```

```
#stats
describe(norm_1k, fast=TRUE) %>%
  dplyr::select(c(-vars,-n))
```

```
##                mean   sd min max range   se
## Order.ID       0.50 0.29   0   1     1 0.01
## Units.Sold     0.50 0.29   0   1     1 0.01
## Unit.Price     0.38 0.33   0   1     1 0.01
## Unit.Cost      0.34 0.34   0   1     1 0.01
## Total.Revenue 0.20 0.22   0   1     1 0.01
## Total.Cost     0.18 0.22   0   1     1 0.01
## Total.Profit   0.23 0.22   0   1     1 0.01
```

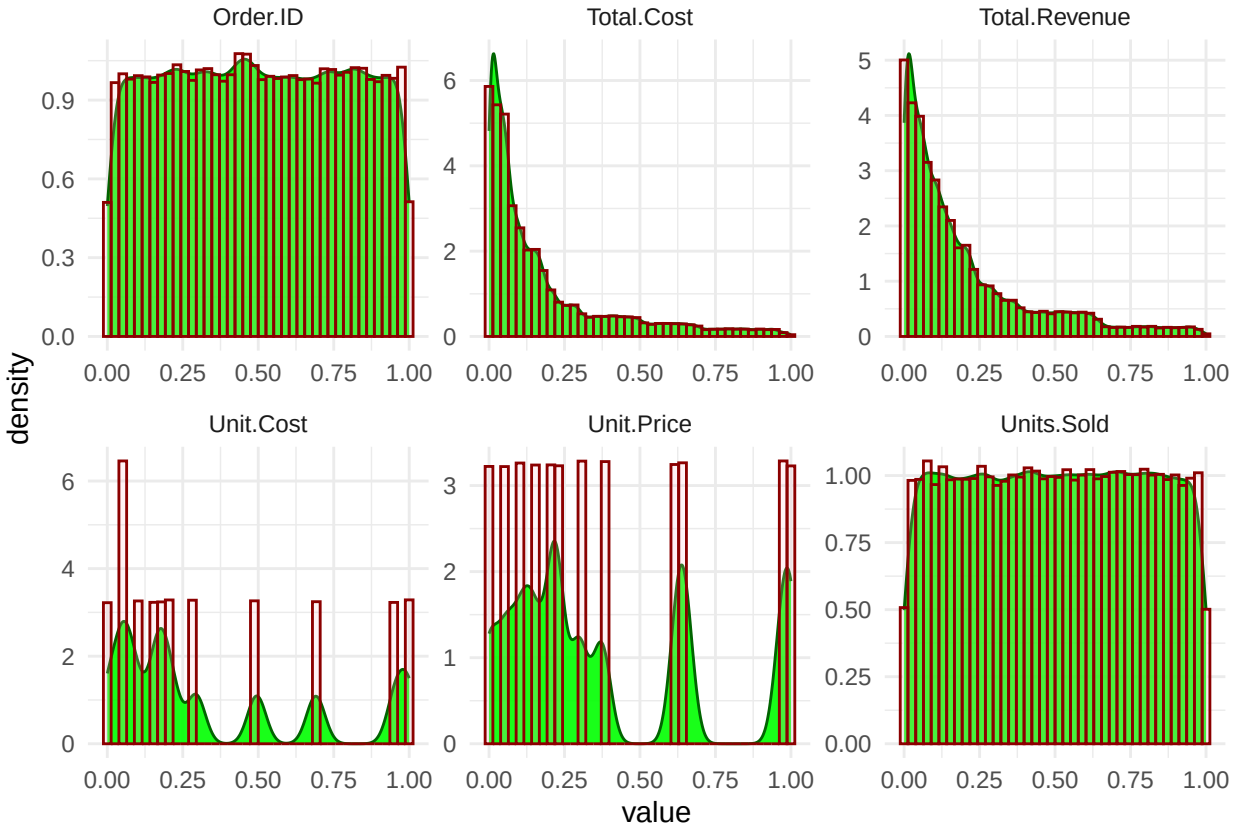
```
#distribution
norm_1k %>%
  gather(variable, value, 1:6) %>%
  ggplot(aes(value)) +
    facet_wrap(~variable, scales = "free") +
    geom_density(fill = "green", alpha=0.9, color="darkgreen") +
    geom_histogram(aes(y=after_stat(density)), alpha=0.2, fill = "pink",
                   color="darkred", position="identity", bins = 40) +
    theme_minimal()
```




```
#stats
describe(norm_100k, fast=TRUE) %>%
  dplyr::select(c(-vars,-n))
```

```
##                mean   sd min max range se
## Order.ID       0.50 0.29   0   1     1  0
## Units.Sold     0.50 0.29   0   1     1  0
## Unit.Price     0.39 0.33   0   1     1  0
## Unit.Cost      0.35 0.34   0   1     1  0
## Total.Revenue 0.20 0.22   0   1     1  0
## Total.Cost     0.18 0.22   0   1     1  0
## Total.Profit   0.23 0.22   0   1     1  0
```

```
#distribution
norm_100k %>%
  gather(variable, value, 1:6) %>%
  ggplot(aes(value)) +
    facet_wrap(~variable, scales = "free") +
    geom_density(fill = "green", alpha=0.9, color="darkgreen") +
    geom_histogram(aes(y=after_stat(density)), alpha=0.2, fill = "pink",
                   color="darkred", position="identity", bins = 40) +
    theme_minimal()
```



Model

```
set.seed(777)

simp_reg_sample_1k <- norm_1k$Total.Revenue %>%
  createDataPartition(p = 0.8, list = FALSE)
simp1k_train  <- norm_1k[simp_reg_sample_1k, ]
simp1k_test <- norm_1k[-simp_reg_sample_1k, ]


simp_reg_sample_100k <- norm_100k$Total.Revenue %>%
  createDataPartition(p = 0.8, list = FALSE)
simp100k_train  <- norm_100k[simp_reg_sample_100k, ]
simp100k_test <- norm_100k[-simp_reg_sample_100k, ]
```

```
simptrain1k_model<- lm(Total.Revenue~Units.Sold, data=simp1k_train )

summary(simptrain1k_model)
```

```
##
## Call:
## lm(formula = Total.Revenue ~ Units.Sold, data = simp1k_train)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38438 -0.11995 -0.02443  0.08641  0.59640
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.007133   0.013649  -0.523    0.601
## Units.Sold   0.411394   0.023203  17.730   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1925 on 798 degrees of freedom
## Multiple R-squared:  0.2826, Adjusted R-squared:  0.2817
## F-statistic: 314.4 on 1 and 798 DF,  p-value: < 2.2e-16
```

```
# Make predictions
prediction <- simptrain1k_model %>% predict(simp1k_test)

class(simp1k_test$Total.Revenue)
```

```
## [1] "numeric"
```

```
# Model performance
data.frame(
  MAE = mae(prediction, simp1k_test$Total.Revenue),
  RMSE = RMSE(prediction, simp1k_test$Total.Revenue),
  R2 = R2(prediction, simp1k_test$Total.Revenue)
)
```

```
##         MAE      RMSE        R2
## 1 0.1345915 0.1846641 0.2601431
```

```
simptrain100k_model<- lm(Total.Revenue~Units.Sold, data=simp100k_train )

summary(simptrain100k_model)
```

```
##
## Call:
## lm(formula = Total.Revenue ~ Units.Sold, data = simp100k_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38676 -0.11840 -0.02697  0.08723  0.59917
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0007499  0.0013246  -0.566    0.571
## Units.Sold   0.4015808  0.0022946 175.009   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1873 on 79999 degrees of freedom
## Multiple R-squared:  0.2769, Adjusted R-squared:  0.2768
## F-statistic: 3.063e+04 on 1 and 79999 DF,  p-value: < 2.2e-16
```

```r
# Make predictions
prediction <- simptrain100k_model %>% predict(simp100k_test)

class(simp100k_test$Total.Revenue)
```

```
## [1] "numeric"
```

```r
# Model performance
data.frame(
  MAE = mae(prediction, simp100k_test$Total.Revenue),
  RMSE = RMSE(prediction, simp100k_test$Total.Revenue),
  R2 = R2(prediction, simp100k_test$Total.Revenue)
)
```

```
##         MAE      RMSE        R2
## 1 0.1393459 0.1876358 0.2723898
```

The steps taken for a simple regression were splitting the normalized data into a train and test only using the numeric values. Using `Units.Sold` as the predictor variable I run my models. The R-squared value of 0.2826 and 0.2769 shows these are terrible models, but that was expected from the EDA. The models accuracy is about 27%-28% which just shows it was not a good model.

Decision Tree

To simplify decision tree, the approach I will use a attribute with a lower number of unique values, but I've chosen **not** to go with Sales.Channel, since this model is very much random and I hope to implement a decision tree with more than 2 possible outcomes for analysis. With this in mind I will make a decision tree model using Region, which I already suspect will create an outcome where Europe and Sub-Saharan Africa are the most likely the regions that will be highlighted in my decision tree, because of its high frequency in the data frames. I will use rpart for my decision tree. NOTE: this will be my first time using rpart, so I am curious on the results.

```r
#split into test/train set

#For df_1k
set.seed(2341)
sample_set <- sample(nrow(df_1k), round(nrow(df_1k)*0.75), replace = FALSE)
df_1k_train <- df_1k[sample_set, ]
df_1k_test <- df_1k[-sample_set, ]

# For df_100k

sample_set <- sample(nrow(df_100k), round(nrow(df_100k)*0.75), replace = FALSE)
df_100k_train <- df_100k[sample_set, ]
df_100k_test <- df_100k[-sample_set, ]

#check class distribution of original, train, and test sets
table_1k<-round(prop.table(table(dplyr::select(df_1k, Region), exclude = NULL)),
            4) * 100
table_1k_train<-round(prop.table(table(dplyr::select(df_1k_train , Region), exclude = NULL)),
            4) * 100
```

```r
table_1k_test<-round(prop.table(table(dplyr::select(df_1k_test, Region), exclude = NULL)),
            4) * 100

table_100k<-round(prop.table(table(dplyr::select(df_100k, Region), exclude = NULL)),
            4) * 100
table_100k_train<-round(prop.table(table(dplyr::select(df_100k_train, Region), exclude = NULL)),
            4) * 100
table_100k_test<-round(prop.table(table(dplyr::select(df_100k_test, Region), exclude = NULL)),
            4) * 100


as.data.frame(table_1k)
```

```
##                              Region Freq
## 1                              Asia 13.6
## 2              Australia and Oceania  7.9
## 3 Central America and the Caribbean  9.9
## 4                            Europe 26.7
## 5      Middle East and North Africa 13.8
## 6                     North America  1.9
## 7                Sub-Saharan Africa 26.2
```

```r
as.data.frame(table_1k_train)
```

```
##                              Region  Freq
## 1                              Asia 13.73
## 2              Australia and Oceania  8.00
## 3 Central America and the Caribbean  9.07
## 4                            Europe 26.93
## 5      Middle East and North Africa 14.53
## 6                     North America  1.87
## 7                Sub-Saharan Africa 25.87
```

```r
as.data.frame(table_1k_test)
```

```
##                              Region Freq
## 1                              Asia 13.2
## 2              Australia and Oceania  7.6
## 3 Central America and the Caribbean 12.4
## 4                            Europe 26.0
## 5      Middle East and North Africa 11.6
## 6                     North America  2.0
## 7                Sub-Saharan Africa 27.2
```

```r
as.data.frame(table_100k)
```

```
##                              Region  Freq
## 1                              Asia 14.55
## 2              Australia and Oceania  8.11
## 3 Central America and the Caribbean 10.73
## 4                            Europe 25.88
```

```
## 5      Middle East and North Africa 12.58
## 6                      North America  2.13
## 7                Sub-Saharan Africa 26.02
```

```
as.data.frame(table_100k_train)
```

```
##                              Region  Freq
## 1                              Asia 14.55
## 2             Australia and Oceania  8.12
## 3 Central America and the Caribbean 10.70
## 4                            Europe 25.93
## 5      Middle East and North Africa 12.58
## 6                      North America  2.18
## 7                Sub-Saharan Africa 25.94
```

```
as.data.frame(table_100k_test)
```

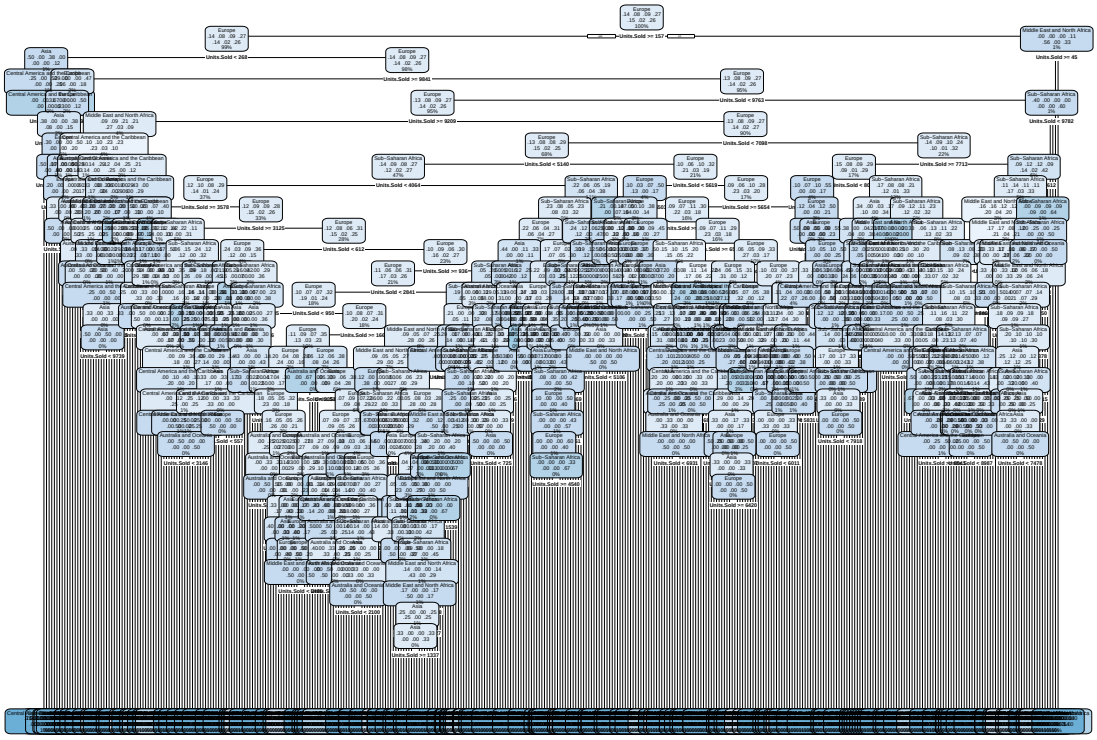
```
##                              Region  Freq
## 1                              Asia 14.54
## 2             Australia and Oceania  8.08
## 3 Central America and the Caribbean 10.83
## 4                            Europe 25.70
## 5      Middle East and North Africa 12.59
## 6                      North America  2.00
## 7                Sub-Saharan Africa 26.25
```

Incorporating `Order.ID` in my model kept causing my file to crash despite it not being made into a factor, therefore I opted to remove it, so that I may see the results.

```
df_1k_train<-df_1k_train%>%
                dplyr::select(-c(Order.ID))
```

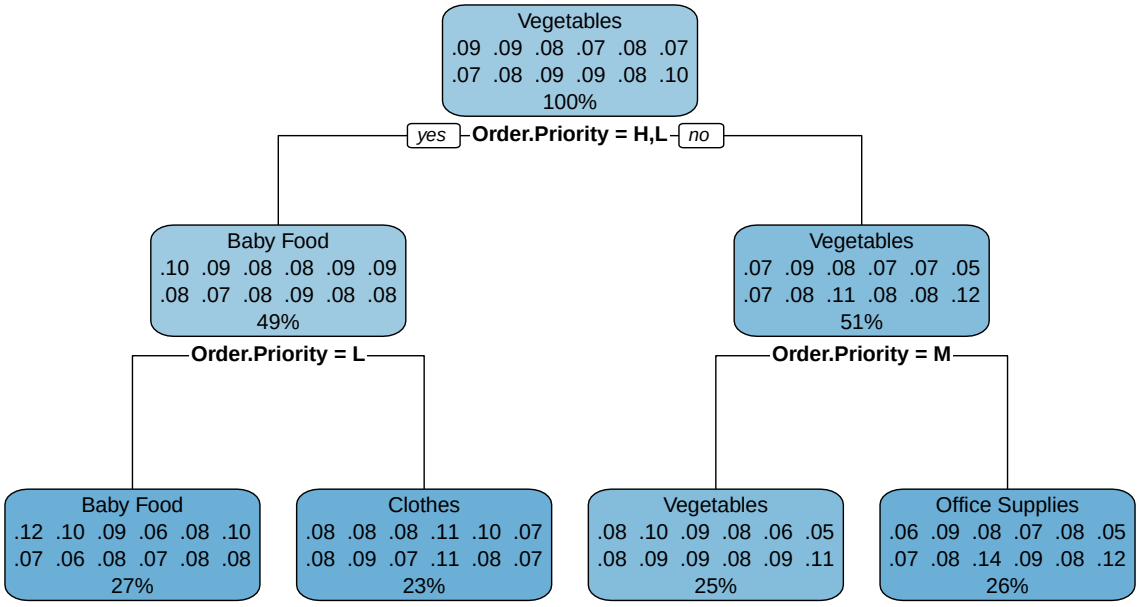
```
#build model via rpart package
model_1k <- rpart(Region ~ Units.Sold,
                          method = "class",
                          data = df_1k_train,
                          control=rpart.control(minsplit=1, minbucket=1, cp=0.001)
                           )

#display decision tree
# rpart.plot(model_100k)
rpart.plot(model_1k, box.palette = "Blues")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

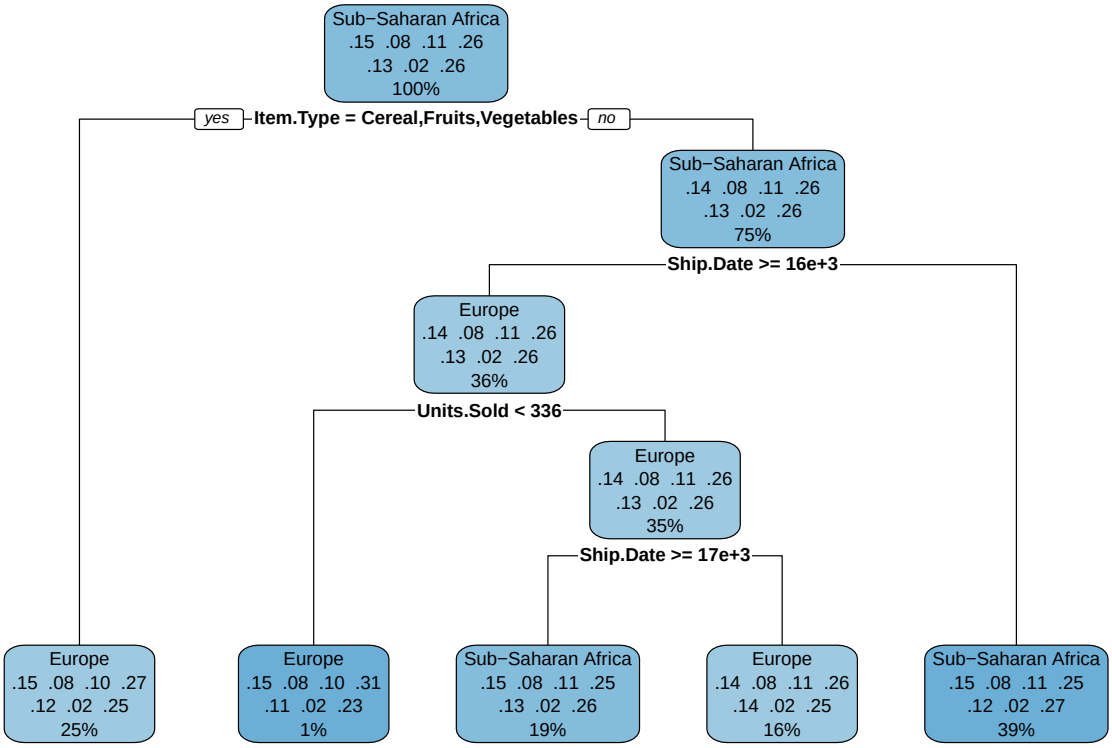


Because the data is undecipherable in this form I opted to make a simpler one with just categorical values.

```
#build model via rpart package
model_1k <- rpart(Item.Type ~ Order.Priority,
                          method = "class",
                          data = df_1k_train,
                          control=rpart.control(minsplit=1, minbucket=1, cp=0.001)
                           )

#display decision tree
# rpart.plot(model_100k)
rpart.plot(model_1k, box.palette = "Blues")
```




```
#build model via rpart package
model_100k <- rpart(Region ~ .-Country,
                        method = "class",
                        data = df_100k_train,
                        control=rpart.control(minsplit=1, minbucket=1, cp=0.001)
                         )

#display decision tree
# rpart.plot(model_100k)
rpart.plot(model_100k, box.palette = "Blues")
```



Response to Questions

1. Are the columns of your data correlated?

Yes they were. Its apparent in just the relationships involved, such as `Country` being categorized in `Region`, and the numeric attributes with label "Total" being derived from their calculations. I also immediately noted the multicollinearity which made it VERY difficult on how I wanted to proceed.

2. Are there labels in your data? Did that impact your choice of algorithm?

No, after checking both data sets, neither had any labels.

3. What are the pros and cons of each algorithm you selected?

The Simple Regression model helped identify the garbage in garbage out data results we wer getting, and because of my familiarity with it I was able to assess and understand the results very easily.

In contrast, this is the first time I'm using a Regression Tree and I am not 100% comfortable with selecting data that is best used for this model. For instance, originally I had decided to select `Region` and `Units.Sold` for my tree, but `R` did not make a useful of even viewable visual. I ended up using to small categories in the 1k data so the result was printable, but in contest with the data, all I can decipher is based on the frequency this is the likelihood of a level of priority based on `Item.Type`, which is still a somewhat confusing assessment for me. I also read through the cran r_project.org documentation for rpart their is limitations

to the amount of factors you may use, forcing me to disregard Country altogether. Using the larger data set I feel a great deal of data was ommitted considersing only 2 regions were represented here.

4. How your choice of algorithm relates to the datasets (was your choice of algorithm impacted by the datasets you chose)?

I chose simple regression when I figured the data had multicollinearity and assumed that my transformations would not do much to make the data a better fit.

5. Which result will you trust if you need to make a business decision?

Simple regression. I would have to circle back to business and explain why the data would not be a suitable fit for prediction or analysis.

6. Do you think an analysis could be prone to errors when using too much data, or when using the least amount possible?

Definitely the Decision Tree, but to be frank operator errors and unfamiliarity with this method is definitely a major factor to account for.

7. How does the analysis between data sets compare?

No. After I assess the lack of usefulness of the numeric values I opted to make this a learning opportunity in using a decision tree and familiarizing myself with it for future use.

```
rm(list = ls(pattern = "_tmp$"))
```