Data 622 - Homework 1

Leticia Salazar

October 8, 2023

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

Load Libraries: Below are the libraries used to complete this assignment

```
library(tidyverse) # data prep
```

```
FALSE -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
FALSE v dplyr     1.1.3     v readr     2.1.4
FALSE v forcats   1.0.0     v stringr   1.5.0
FALSE v ggplot2   3.4.3     v tibble    3.2.1
FALSE v lubridate 1.9.3     v tidyr     1.3.0
FALSE v purrr     1.0.2
FALSE -- Conflicts ------------------------------------------ tidyverse_conflicts() --
FALSE x dplyr::filter() masks stats::filter()
FALSE x dplyr::lag()    masks stats::lag()
FALSE i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become er
```

```
library(skimr) # data prep
#install.packages('rpart.plot') # must install if not already
library(rpart) # decision tree package
library(rpart.plot) # decision tree display package
library(knitr) # kable function for table
```

```r
library(tidyr) # splitting data
library(ggplot2) # graphing
library(hrbrthemes) # chart customization
```

```
FALSE NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
FALSE       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
FALSE       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```r
library(gridExtra) # layering charts
```

```
FALSE
FALSE Attaching package: 'gridExtra'
FALSE
FALSE The following object is masked from 'package:dplyr':
FALSE
FALSE     combine
```

```r
library(stringr) # data prep
library(tidymodels) # predictions
```

```
FALSE -- Attaching packages -------------------------------------- tidymodels 1.1.1 --
FALSE v broom        1.0.5     v rsample      1.2.0
FALSE v dials        1.2.0     v tune         1.1.2
FALSE v infer        1.0.5     v workflows    1.1.3
FALSE v modeldata    1.2.0     v workflowsets 1.0.1
FALSE v parsnip      1.1.1     v yardstick    1.2.0
FALSE v recipes      1.0.8
FALSE -- Conflicts ----------------------------------------- tidymodels_conflicts() --
FALSE x gridExtra::combine() masks dplyr::combine()
FALSE x scales::discard()    masks purrr::discard()
FALSE x dplyr::filter()      masks stats::filter()
FALSE x recipes::fixed()     masks stringr::fixed()
FALSE x dplyr::lag()         masks stats::lag()
FALSE x dials::prune()       masks rpart::prune()
FALSE x yardstick::spec()    masks readr::spec()
FALSE x recipes::step()      masks stats::step()
FALSE * Learn how to get started at https://www.tidymodels.org/start/
```

Load Data: The data chosen from Excel BI Analytics were the 100 sales records for the small and 5000 sales records for the large. The data sets are included in my GitHub and read into R.

The Data:

Both of these data sets contain the same columns with the minor difference of the total of records. The columns are as follows:

- Region: region of sale

- Country: country of sale
- Item Type: item sold
- Sales Channel: online or offline sale
- Order Priority: priority of the order “L”- Low, “M”- Medium, “H”- High, “C”- Critical
- Order Date: date of the order
- Order ID: ID of the order
- Ship Date: date the order was shipped
- Units Sold: amount of units sold
- Unit Cost: cost of the order
- Total Revenue: total revenue of the order
- Total Cost: total cost of the order
- Total Profit: total profit of the order

The small data set:

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Order.ID	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50
Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	224598.75
Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75591.66	56065.84	19525.82
Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50
Australia and Oceania	Solomon Islands	Baby Food	Online	C	2/4/2015	547995746	2/21/2015	2974	255.28	159.42	759202.72	474115.08	285087.64

The large data set:

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Order.ID	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Central America and the Caribbean	Antigua and Barbuda	Baby Food	Online	M	12/20/2013	957081544	1/11/2014	552	255.28	159.42	140914.56	87999.84	52914.72
Central America and the Caribbean	Panama	Snacks	Offline	C	7/5/2010	301644504	7/26/2010	2167	152.58	97.44	330640.86	211152.48	119488.38

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Order.ID	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Europe	Czech Republic	Beverages	Offline	C	9/12/2017	478051030	9/29/2017	4778	47.45	31.79	226716.10	151892.62	74823.48
Asia	North Korea	Cereal	Offline	L	5/13/2010	892599952	6/15/2010	9016	205.70	117.11	1854591.20	1055863.76	798727.44
Asia	Sri Lanka	Snacks	Offline	C	7/20/2015	571902596	7/27/2015	7542	152.58	97.44	1150758.36	734892.48	415865.88
Middle East and North Africa	Morocco	Personal Care	Offline	L	11/8/2010	412882792	11/22/2010	48	81.73	56.67	3923.04	2720.16	1202.88

Data Exploration:

Let's explore the data sets; first the `small_df` data set, using the `skimr` library we can obtain quick summary statistics beyond the `summary()`. We notice that we have 14 variables split into 7 character and 7 numeric. There seems to be no missing values, so this will have a simple preparation before we build our models.

Table 3: Data summary

Name	small_df
Number of rows	100
Number of columns	14
Column type frequency:	
character	7
numeric	7
Group variables	None

Variable type: character

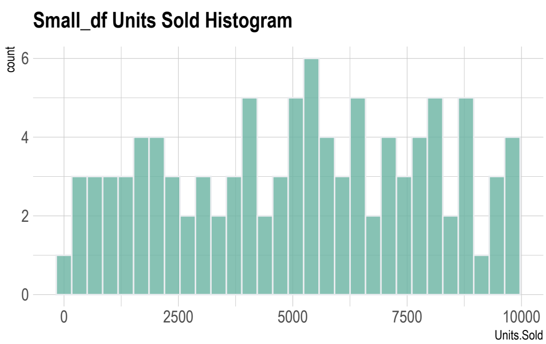
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Region	0	1	4	33	0	7	0
Country	0	1	4	32	0	76	0
Item.Type	0	1	4	15	0	12	0
Sales.Channel	0	1	6	7	0	2	0
Order.Priority	0	1	1	1	0	4	0
Order.Date	0	1	8	10	0	100	0
Ship.Date	0	1	8	10	0	99	0

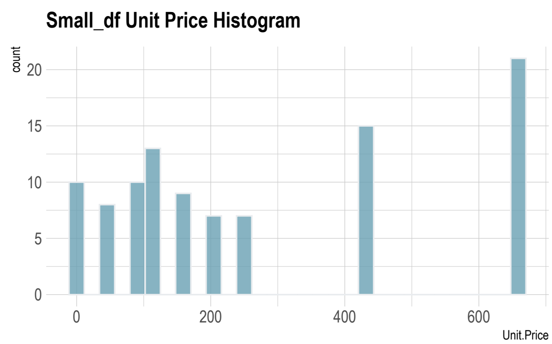
Variable type: numeric

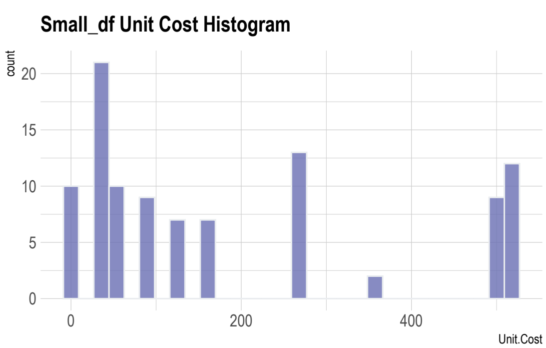
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Order.ID	0	1	555020412.36	260615257.13	114606559.00	338922488.00	557708561.00	790755080.75	994022214.00	
Units.Sold	0	1	5128.71	2794.48	124.00	2836.25	5382.50	7369.00	9925.00	

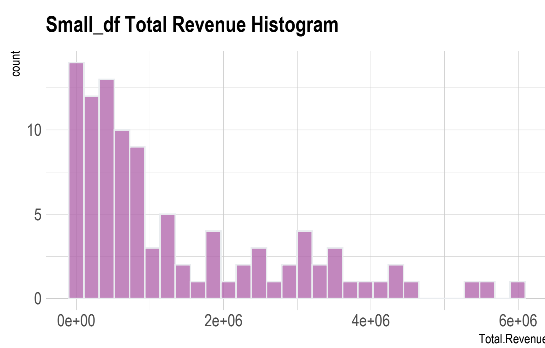
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Unit.Price	0	1	276.76	235.59	9.33	81.73	179.88	437.20	668.27	
Unit.Cost	0	1	191.05	188.21	6.92	35.84	107.28	263.33	524.96	
Total.Revenue	0	1	1373487.68	1460028.71	4870.26	268721.21	752314.36	2212044.68	5997054.98	
Total.Cost	0	1	931805.70	1083938.25	3612.24	168868.03	363566.38	1613869.72	4509793.96	
Total.Profit	0	1	441681.98	438537.91	1258.02	121443.58	290768.00	635828.80	1719922.04	

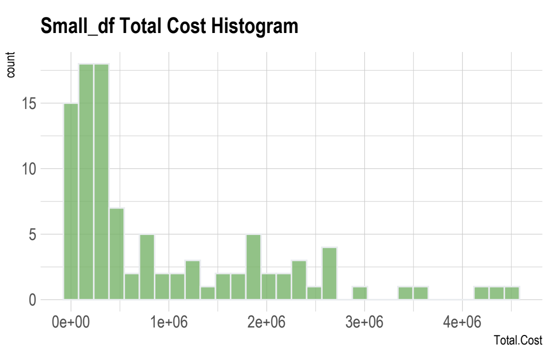
Let's take a look at the distributions of the numeric variables for the small data set:

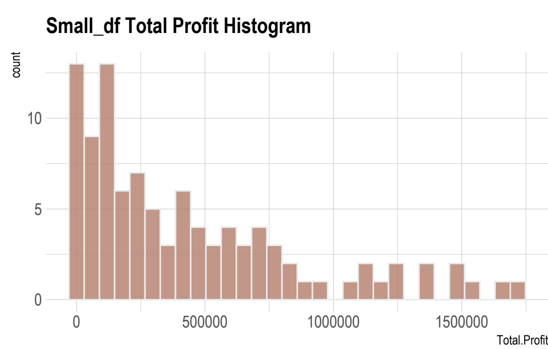




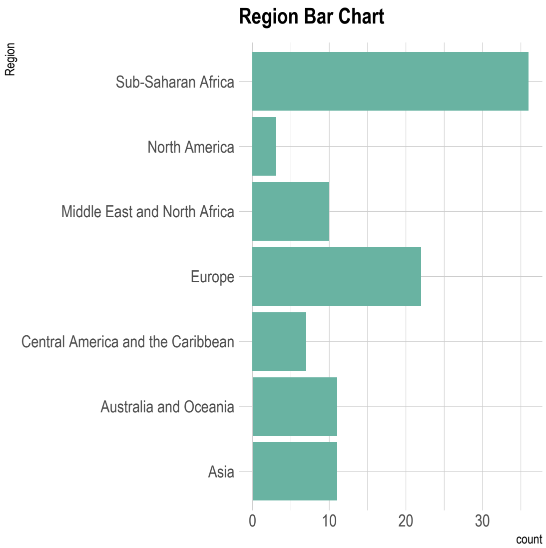


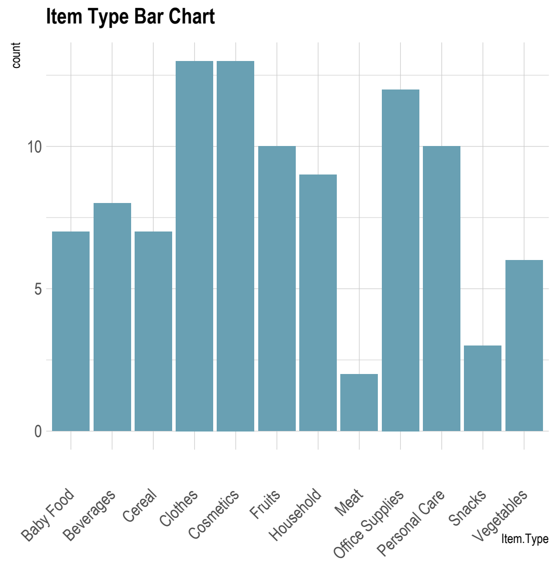


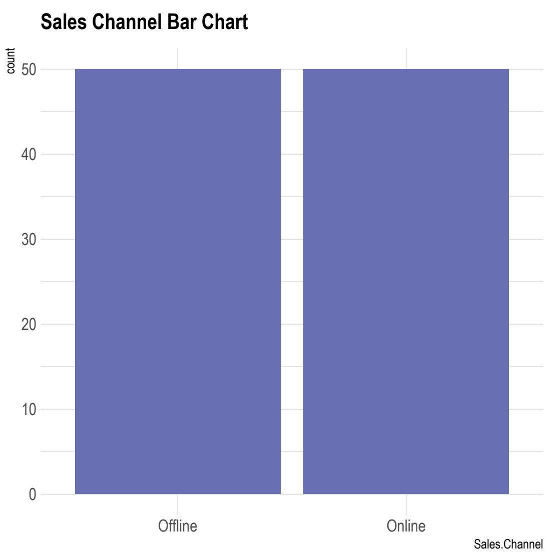


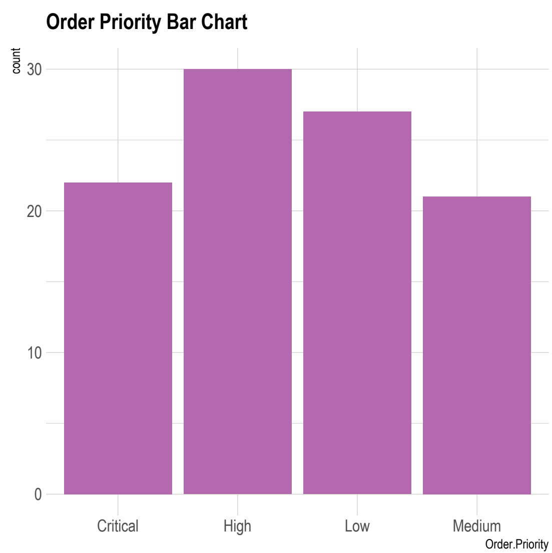


Categorical variables visualization for the small dataset:









Now, the `large_df` dataset is composed of 5000 values of the same 14 variables as the `small` data set. It also has 7 character and 7 numeric variables with no missing values.

Table 6: Data summary

Name	large_df
Number of rows	5000
Number of columns	14

Column type frequency:	
character	7
numeric	7
Group variables	None

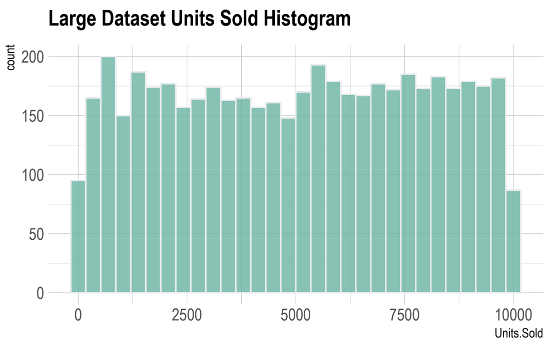
Variable type: character

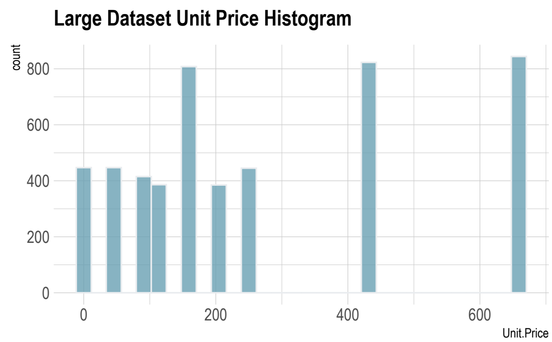
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Region	0	1	4	33	0	7	0
Country	0	1	4	32	0	185	0
Item.Type	0	1	4	15	0	12	0
Sales.Channel	0	1	6	7	0	2	0
Order.Priority	0	1	1	1	0	4	0
Order.Date	0	1	8	10	0	2305	0
Ship.Date	0	1	8	10	0	2320	0

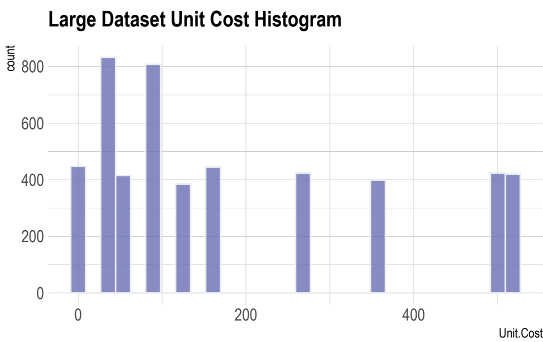
Variable type: numeric

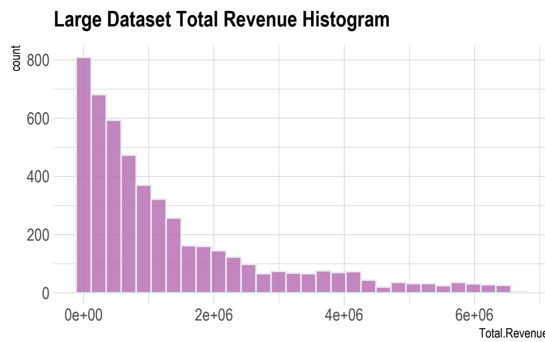
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Order.ID	0	1	54864473[illegible]	[illegible]9467108.[illegible]	[illegible]0090873.[illegible]	[illegible]0104216.[illegible]	[illegible]2314960.[illegible]	[illegible]8770944.[illegible]	[illegible]9879729.00	
Units.Sold	0	1	5030.70	2914.52	2.00	2453.00	5123.00	7576.25	9999.00	
Unit.Price	0	1	265.75	218.72	9.33	81.73	154.06	437.20	668.27	
Unit.Cost	0	1	187.49	176.42	6.92	35.84	97.44	263.33	524.96	
Total.Revenue	0	1	1325737.84	1475374.67	65.31	257416.82	779409.46	1839975.10	6672675.95	
Total.Cost	0	1	933093.20	1150873.22	48.44	154748.02	468180.67	1189577.71	5248025.12	
Total.Profit	0	1	392644.65	382935.15	16.87	85339.25	279095.18	565106.42	1726007.49	

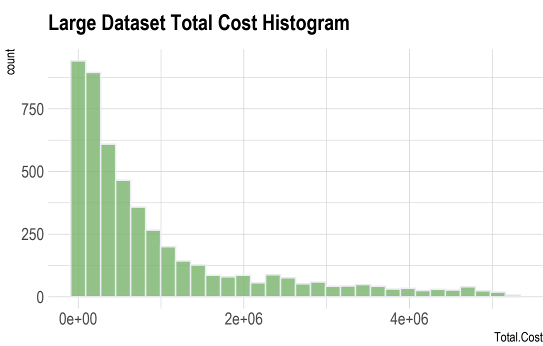
Visualizations of the numeric variable distributions of the large dataset:

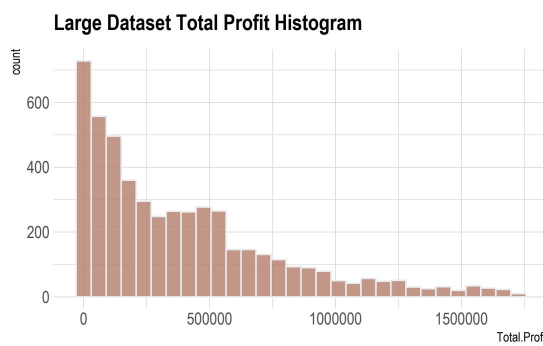




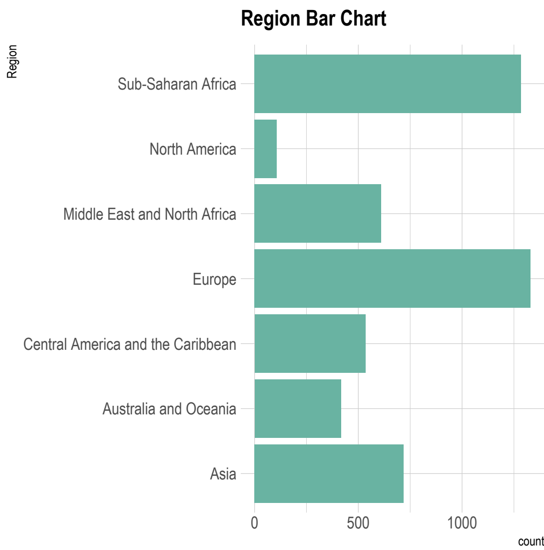


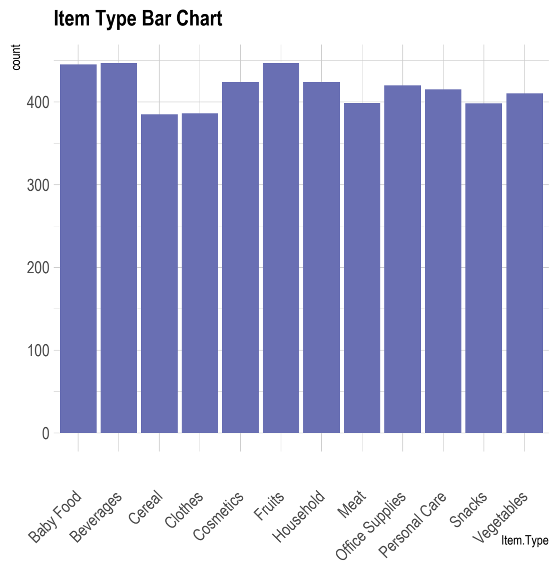


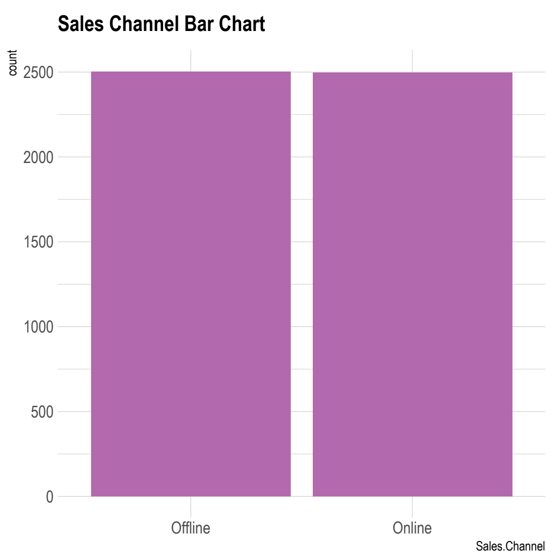


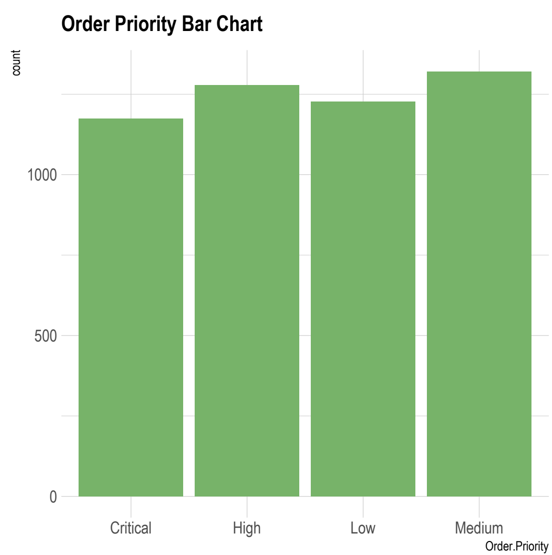


Now let's look at the categorical variables:









Some notes on the visualizations above:

- The distributions for both small and large datasets are fairly similar with the exception of `Units.Sold`. The large data set has a more unform distribution for this variable compared to the small dataset.
- There is no pattern for `Unit.Price` and `Unit.Cost` in both datasets
- Both data sets have the variables `Total.Revenue`, `Total.Cost` and `Total.Profit` histograms right skewed
- For the categotical variables, both Sub-Saharan African and Europe are the top 2 largest Region where the sales are from for both datasets with North American being the region with the lowest sales
- `Sales.Channel` variable are even for both datasets
- The `Item.Type` variable in the small dataset has top 3 items as: Clothes, Cosmetics and Office Supplies while the large dataset has Beverages, Fruits and Baby Food as it's top 3 items.
- In terms of the `Order.Priority`, the small dataset's "High" and "Low" priorities have the highest

frequency count as opposed to the larger dataset which has "Medium" and "High" with the largest frequency count.

Data Preparation:

Now that I've visualized the data it's time to make some changes to the variables. First, convert the categorical values into `as.factor` and convert the two columns containing dates to `as.Date` to be able to manipulate. I'll drop the `Order.ID` column as it is not needed with our model. Below are the results:

Small dataset:

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50
Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36
Europe	Russia	Office Supplies	Offline	L	2014-05-02	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224598.75
Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	2014-07-05	8102	9.33	6.92	75591.66	56065.84	19525.82
Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	2013-02-06	5062	651.21	524.96	3296425.02	2657347.52	639077.50
Australia and Oceania	Solomon Islands	Baby Food	Online	C	2015-02-04	2015-02-21	2974	255.28	159.42	759202.72	474115.08	285087.64

Large dataset:

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Central America and the Caribbean	Antigua and Barbuda	Baby Food	Online	M	2013-12-20	2014-01-11	552	255.28	159.42	140914.56	87999.84	52914.72
Central America and the Caribbean	Panama	Snacks	Offline	C	2010-07-05	2010-07-26	2167	152.58	97.44	330640.86	211152.48	119488.38
Europe	Czech Republic	Beverages	Offline	C	2011-09-12	2011-09-29	4778	47.45	31.79	226716.10	151892.62	74823.48

Region	Country	Item.Type	Sales.Channel	Order.Priority	Order.Date	Ship.Date	Units.Sold	Unit.Price	Unit.Cost	Total.Revenue	Total.Cost	Total.Profit
Asia	North Korea	Cereal	Offline	L	2010-05-13	2010-06-15	9016	205.70	117.11	1854591.20	1055863.76	798727.44
Asia	Sri Lanka	Snacks	Offline	C	2015-07-20	2015-07-27	7542	152.58	97.44	1150758.36	734892.48	415865.88
Middle East and North Africa	Morocco	Personal Care	Offline	L	2010-11-08	2010-11-22	48	81.73	56.67	3923.04	2720.16	1202.88

Model Selection:

While exploring the data I've noticed that my data doesn't have labels by default but more so can be defined based on the analysis being performed. There are two labels I can visualize `Order.Priority` or `Total.Profit`. With `Order.Priority` as my target variable I can predict which category a new sale would fall into "C", "H", "L", or "M". Variables such as `Item.Type`, `Units.Sold` and `Total.Cost` can affect the level in priority of a new sale. With `Total.Profit` as my target variable I can consider all the other variables to see how it affects sales profits.

Decision Trees can be a suitable choice for predicting a categorical target variable like `Order.Priority`. They are a type of supervised machine learning algorithm that can handle both classification and regression tasks. In this case, I chose to classify orders into different priority levels and have opted to use a decision tree model.

Some considerations for using a decision tree model for predicting `Order.Priority`:

- Well-suited for predicting categorical target variables, such as priority levels (critical, low, medium, high).
- Highly interpretable models that can easily visualize the tree structure and understand the rules that lead to a particular priority classification.
- Can provide information about feature importance, helping you identify which factors have the most significant influence on order priority.
- Can capture nonlinear relationships between input features and the target variable, which can be valuable if the relationship between order attributes and priority is not linear.

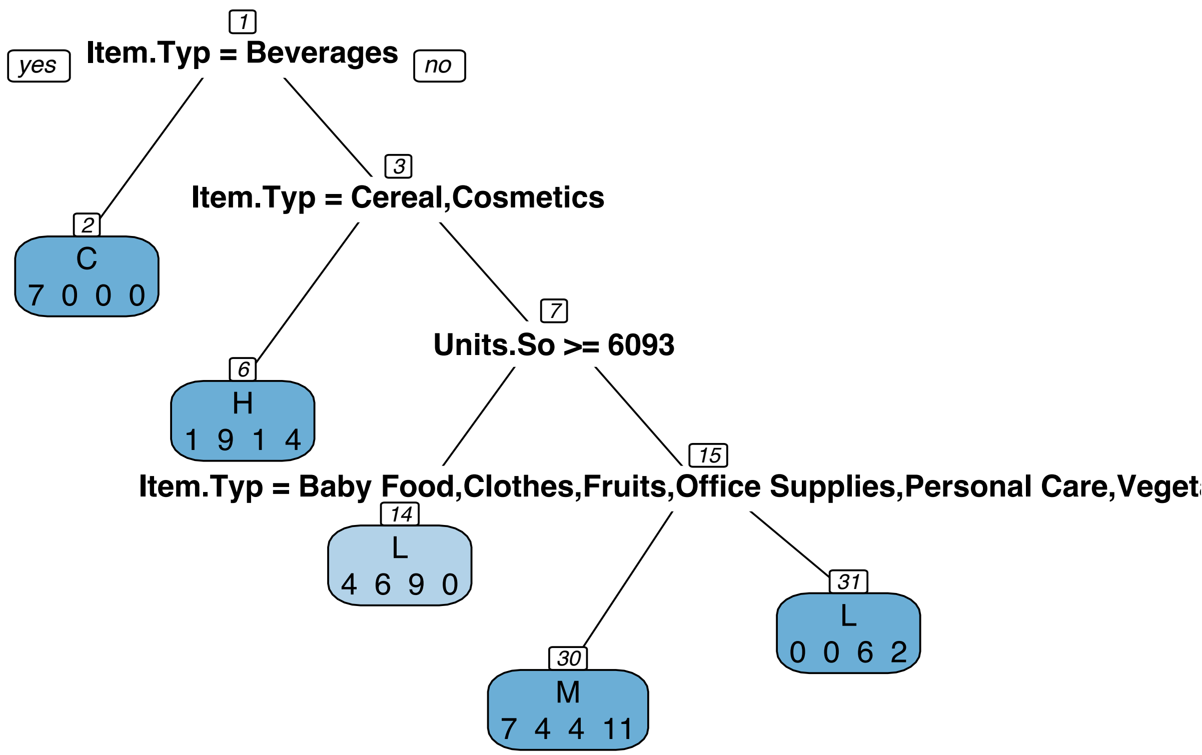
There are also some considerations and potential challenges when using decision trees:

- Decision trees can be prone to overfitting, where the model captures noise in the training data and performs poorly on unseen data.
- If the dataset has imbalanced class distributions for order priorities (e.g., a lot of "low" priority orders and few "high" priority orders), these will need to be addressed during model training and evaluation.
- The quality of your input data, including missing values and outliers, can affect the performance of a decision tree model.
- To achieve the best performance with a decision tree, you may need to tune hyperparameters, such as the maximum depth of the tree or the minimum number of samples required to split a node.

Model Building:

First, we start by splitting both datasets into the standard ratio 75:25

Now we can start the decision tree for the small data set using the `rpart` function and setting `Order.Priority` as our target variable followed by the rest of the variables. The results are below:



To test the above model I used the `small_df` testing data to create the prediction table below:

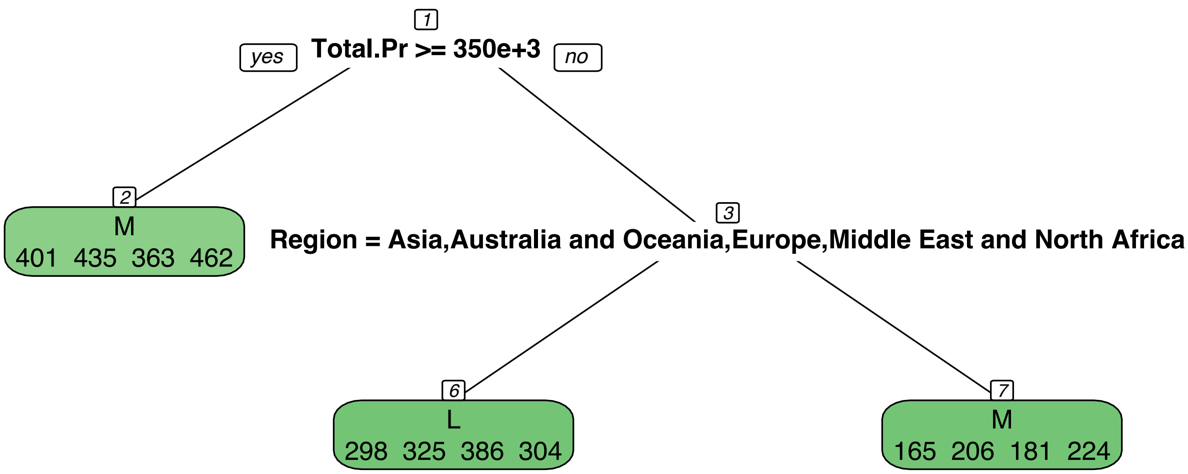
	C	H	L	M
C	0	1	0	2
H	1	4	2	4
L	0	0	4	3
M	0	0	1	3

and checking the accuracy of the model using the predicted values alongside the `small_test` data which is 44%:

Table 12: Accuracy

x
0.44

Now that the `small_df` has been completed it's time to do the `large_df`. Same as before, create the decision tree with `Order.Priority` as my target variable. The results are below:



Testing the model against the `large_test` data:

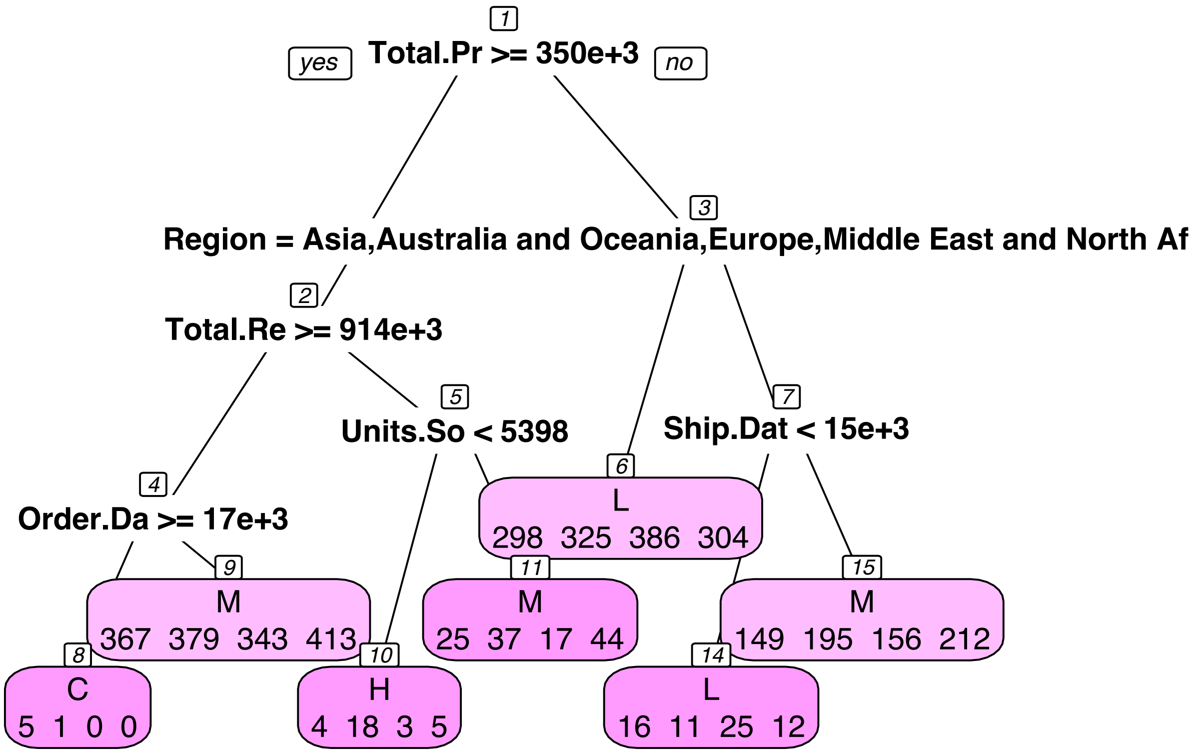
	C	H	L	M
C	0	0	103	207
H	0	0	98	214
L	0	0	105	192
M	0	0	115	216

and now to check the accuracy of the model which is 25.7%:

Table 14: Accuracy

x
0.2568

I did not expect the decision tree for the larger dataset to be this small along with the accuracy compared to the small dataset. After some research I found some parameters I could improve on the `rpart` function to improve the model and it's accuracy. Below are the results:



Now that I have a better decision tree I test the above model using our `large_df_model2` and `large_test` testing data:

	C	H	L	M
C	3	4	107	196
H	1	6	106	199
L	1	0	109	187
M	0	5	124	202

and finally checking the accuracy of the second model; we see the accuracy is only 25.6% which is less than the first model. There wasn't much improvement in accuracy but we note the changes in the nodes of the decision trees.

Table 16: Accuracy

x
0.256

Conclusion:

Based on the results for both `small_df` and `large_df` although the smaller data set has a higher accuracy than the larger dataset it is still not sufficient enough to make business decisions. The models could use some improvements to make it more valuable. For the large dataset, changing the parameters didn't improve the accuracy of the model but it was a lower percentage than the small dataset accuracy. It's safe to assume that using too much or too little data can have it's challenges and lead to some errors.

For instance using too much data can lead to:

- being computationally expensive and time-consuming, especially for complex models like deep neural networks
- there's a risk of overfitting where the model learns to memorize the training data rather than generalizing from it leading to poor performance
- not ensuring data quality, where the larger data set can contain noise and outliers that can affect the model's performance

Too little data can lead to:

- a model struggling to learn complex patterns and generalize effectively therefore, it's performance may not be representative of the underlying relationships
- overfitting to the noise of the dataset resulting in a model that performs well on the training data but poorly on new data introduced
- reduction of the variables used in the analysis to prevent overfitting that can lead to losing important information

By choosing to create decision trees for these two datasets I wanted to predict `Order.Priority` to visualize how the outcomes "Critical", "Low", "Medium" and "High" are affected by the other variables. Based on my findings I conclude that a decision tree was probably not the best route to take for these two datasets and could have used other sizes in small and large datasets to view bigger differences between the models.

References:

- StackOverFlow- Color Nodes in rpart Tree
- DataCamp - Decision Trees R
- StackOverFlow - Display More Nodes in Decision Tree in R
- Guru99 - Decision Trees

For code used | not used in this assignment see GitHub.