Final Project Report - CMDA 3654

Lukas Coffey

8/16/2019

Introduction

The data contains information about online orders from a superstore (e.g. something like Amazon). The idea is to attempt to find new ways to improve efficiency with regards to logistics and shipping. Improving these areas will in turn reduce the amount of time between initial customer order and delivery. Companies that sell merchandise online need to have distribution centers where their merchandise is kept and shipped from. When an order is placed, the distribution center receives the order and then sends that product to the appropriate place via a shipping method. This is what we will focus on in this analysis.

Overview of Dataset

The dataset I am using contains information about online super store orders. I got this data from the Tableau Community Forum: https://community.tableau.com/docs/DOC-1236.

It contains rows corresponding to orders placed by online customers. There are 21 columns and 9994 rows of data. The data is sample data and most likely randomly generated, so it's very normal data with not a lot of dramatic trends. There are a few number columns, including Sales (total sale amount), Profit (profit made from that sale), Discount (percent discount), and Quantity (amount of single product ordered). There was also a second dataset that came with it containing information about returns, which I joined as another column, "Returned."

Interesting note: There are two extra columns that have been added, Latitude and Longitude. I wanted to visualize the distribution of products around the shipping area. The postal code was included in the data so I searched for a mapping of postal codes to latitude/longitude coordinates. I joined the two datasets and created two new columns that represented the lat/lng for each order. By plotting this on an XY plane I was able to visualize how product categories were distributed among the different regions!

Statistical Methods Used

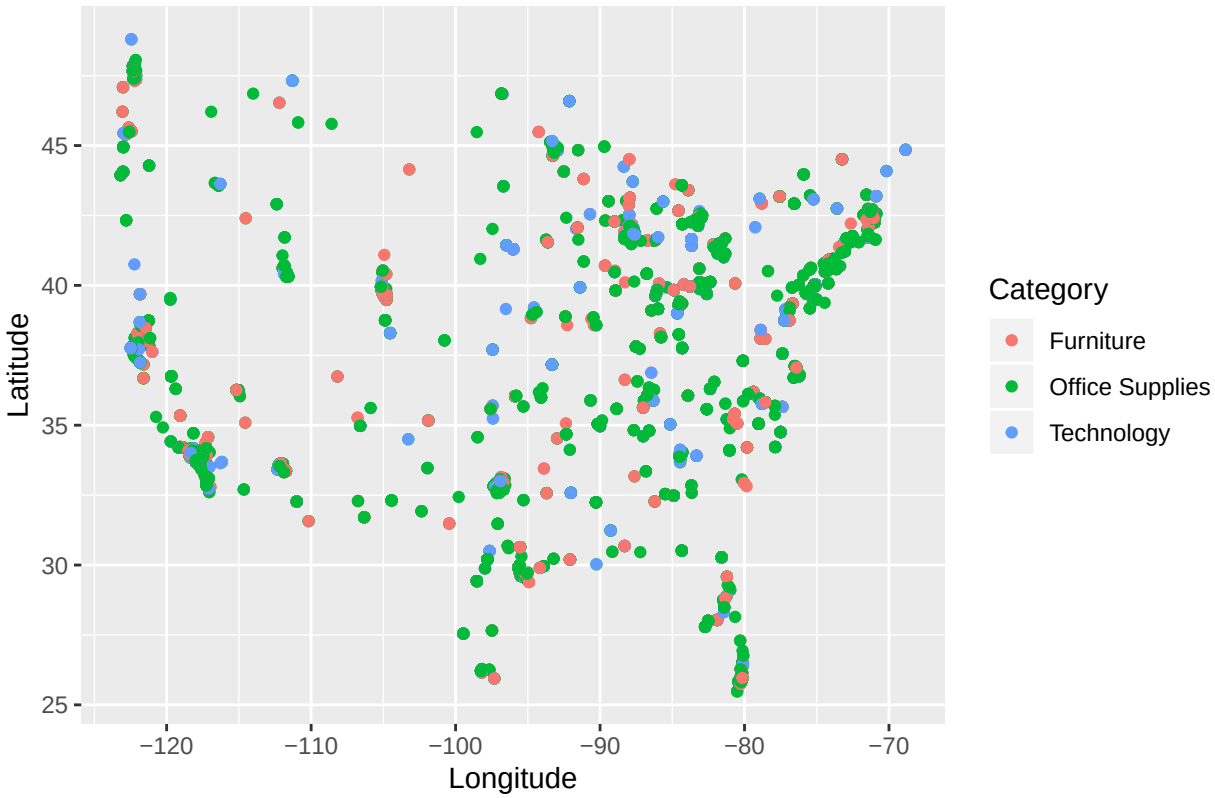
Proposed Business Question: Where should we build product distribution centers to increase delivery efficiency (minimize delivery time)?

There are 4 given regions: East, South, Central, and West. There are 3 product categories: Furniture, Office Supplies and Technology. We can see the different regions illustrated below in **Plot 1**. We want to place distribution centers so that they can minimize delivery time, but some of these regions are very far-reaching and spread out.

Below in **Plot 2**, we can see the distribution of product categories across the service area. You can see distinct clusters of categories in certain areas, especially the Office Supplies category. However the categories are quite intermingled with one another, and it's hard to make a clear distinction between purchases in certain areas. This led me to believe that there may be a more efficient way to cluster orders, based on coordinates and time of year. If they are based on time of year then we can also coordinate WHEN certain products should be placed in those distribution centers to minimize shipping time to the customer.

To do this analysis I will use **hierarchial clustering via agglomeration**. We want to find *new* ideal categories of order areas besides the Region variable, since this may not be the most efficient dispersion of customers and the regions cover large areas.

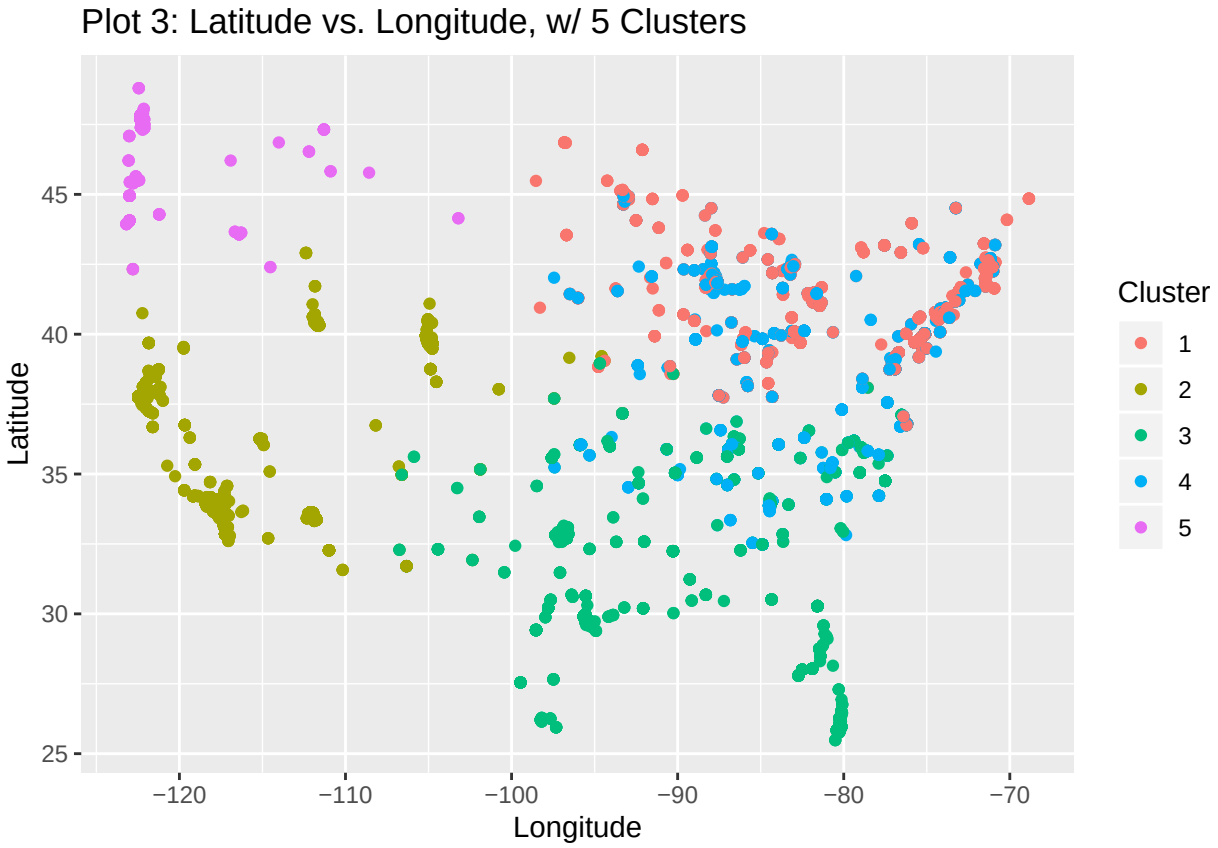
Plot 2: Distribution of Product Categories



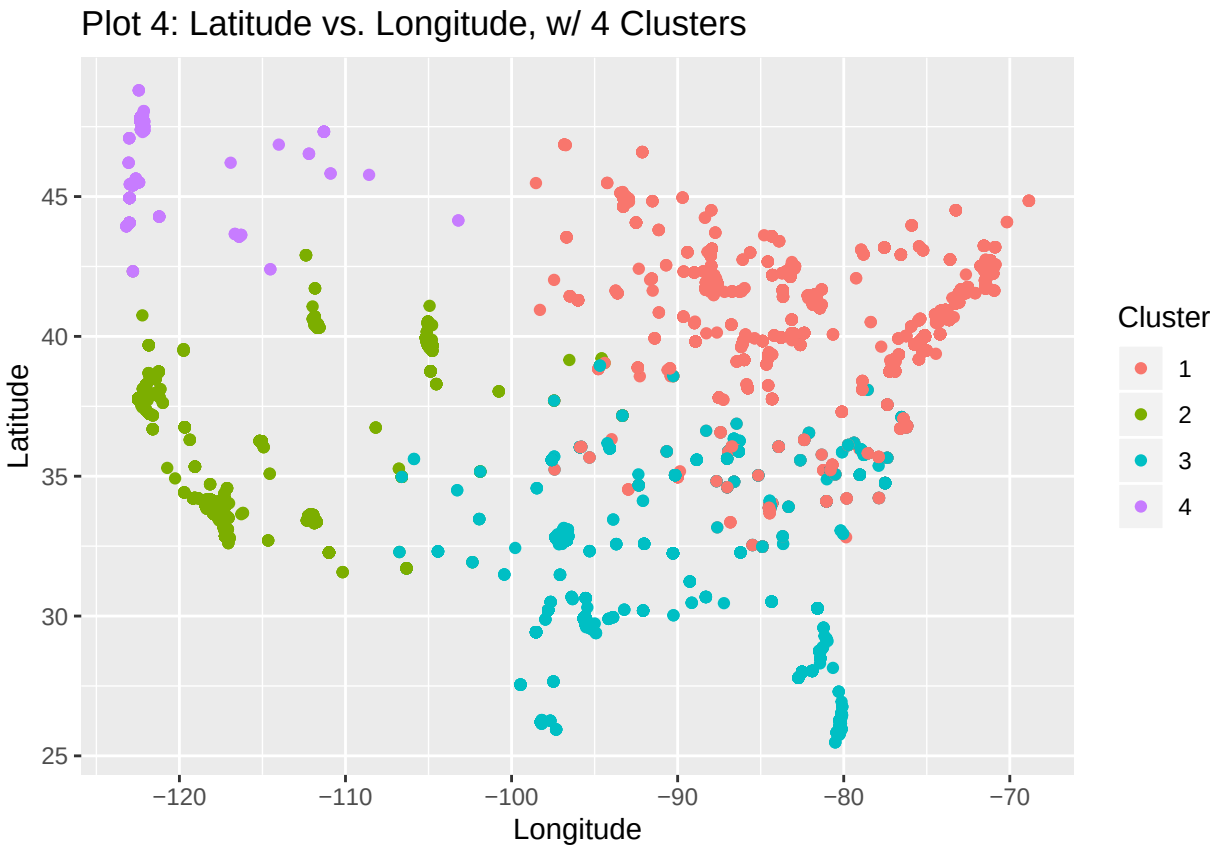
Analysis Results

Using **hierarchial clustering via agglomeration**, I performed clustering with the average linkage dissimilarity measure. I attempted it with complete and single linkages, but average produced the best classification. The optimal number of clusters turned out to be 5 using the elbow method.

The clusters are plotted below in **Plot 3:**. We can see quite a bit of overlap in clusters 1 and 4, and also a little bit in clusters 3 and 4.



Because clusters 1 and 4 overlapped so much, I decided to try reducing the number of clusters to 4 instead of 5. In **Plot 4 below**, this produced a much cleaner delineation between the clusters on the right side of the plot (Eastern seaboard). When looking at these clusters and comparing them to **Plot 1**, they seem very similar; but these clusters are also based on order date, specifically what month they were ordered. This creates a more accurate picture of purchasing activity with respect to time of year.

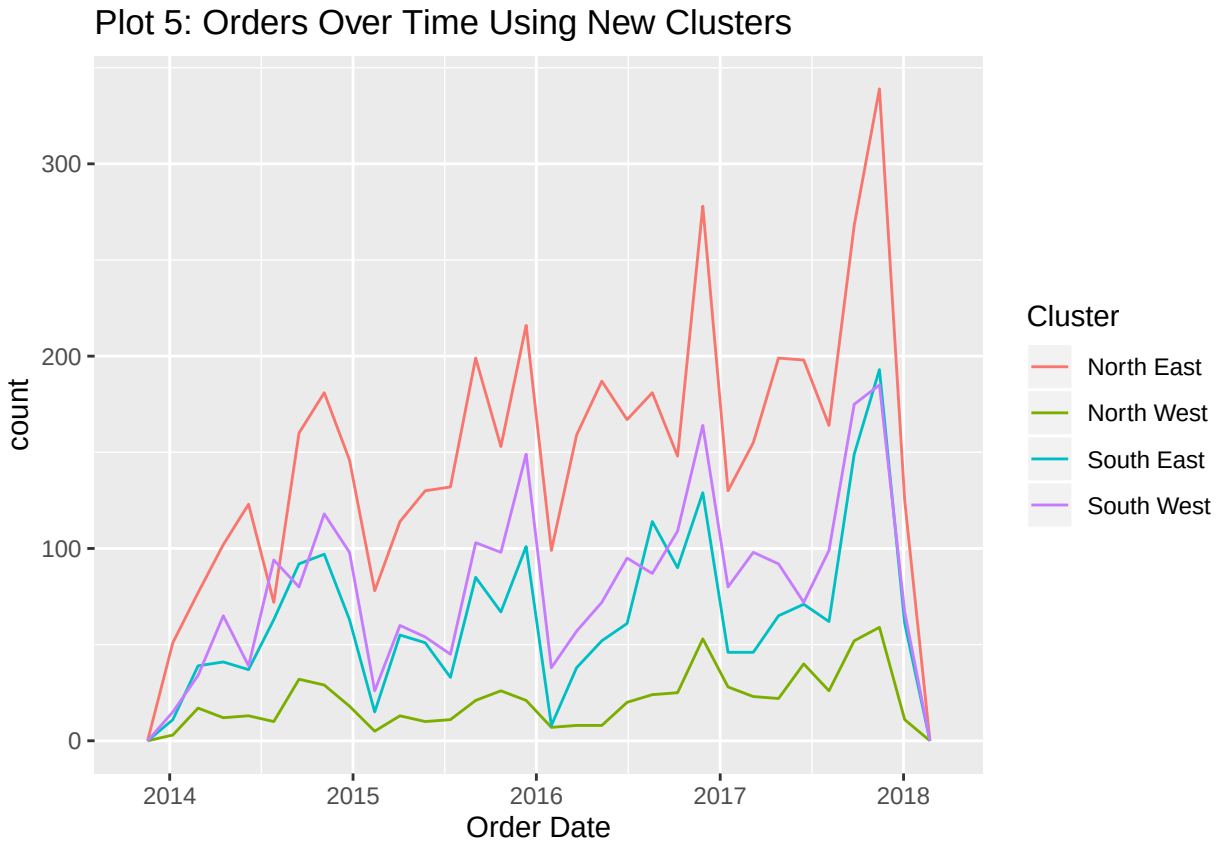

Plot 4: Latitude vs. Longitude, w/ 4 Clusters
Latitude
45
40
35
30
25
−120
−110
−100
−90
−80
−70
Longitude
Cluster
1
2
3
4

Conclusions

Using agglomerative clustering, we've found some new categories of orders we didn't know were there. Instead of relying only upon the built-in Region data, we have used our historical data to find new groupings of customers based on location and time factors. In the table below (**Table 1**), we can see that the number of orders in each cluster are much different than the original Region data. We can rename these categories to be more descriptive, instead of just numbers:

Cluster Name	Number of Orders	Original Cluster Number
North East	4731	1
South West	2568	4
South East	2035	3
North West	647	2

We can also visualize this using these newly-found clusters as a factor, and look at the number of orders from each cluster over time.



In this last table (**Table 3**) we look at the distribution of product categories in each new cluster:

	North East	North West	South East	South West
Furniture	1005	138	397	578
Office Supplies	2861	378	1258	1521
Technology	865	131	380	469

Our original question was, **where should we build product distribution centers to increase delivery efficiency (minimize delivery time)?**

We can definitely see that these new clusters help us prioritize where our products should go, and where we should place our distribution centers. **By placing distribution centers containing more or less merchandise from each product category strategically in these new clusters, we can minimize the shipping time because this merchandise will then be closer to the consumer.**

With further analysis we could try and figure out *what* products need to be in which distribution centers *at what time of the year*. Shipments to distribution centers could then be planned ahead of time throughout the year to minimize overhead and unnecessary merchandise transport.

How?

If you look at the counts of products in the North East cluster, we see very high counts of the Office Supplies category, meaning we should place more distribution centers there with higher quantities of Office Supplies.

This logic applies to each other category. E.g. we need less products from the Technology category than the Furniture category in the North East, and we need much less of every product category in the North West.

Citations

I've used multiple lectures from CMDA 3654, as well as knowledge gained from CMDA 3654 Homework 9. I did not record the resources I used for that homework, thus I don't have all of them, but this was a main one:

Hierarchial Clustering With R - DataCamp: https://www.datacamp.com/community/tutorials/hierarchical-clustering-R

Appendix

```
# --------------------------
# INITIAL SETUP
# --------------------------
orders <- read_csv("Data/StoreData_CLEAN.csv", col_types = cols(
  "Order Date" = col_date(format = "%Y-%m-%d"),
  "Ship Date" = col_date(format = "%Y-%m-%d"),
  "Ship Mode" = col_factor(),
  "Segment" = col_factor(),
  "Country" = col_factor(),
  "City" = col_factor(),
  "State" = col_factor(),
  "Region" = col_factor(),
  "Category" = col_factor(),
  "Sub-Category" = col_factor(),
  "Returned" = col_factor()
))
```

Plot 1

```
orders %>%
  filter(!is.na(Longitude)) %>%
  ggplot(aes(x = Longitude, y = Latitude, color = Region)) +
    geom_point() +
    labs(title = "Plot 1: Distribution of Regions")
```

Plot 2

```
orders %>%
  filter(!is.na(Longitude)) %>%
  ggplot(aes(x = Longitude, y = Latitude, color = Category)) +
    geom_point() +
    labs(title = "Plot 2: Distribution of Product Categories")
```

Clustering Technique

```
# Cleaning out the NA values to prevent ggplot errors.
filtered_orders <- orders %>%
  filter(!is.na(Longitude), !is.na(Latitude))

# --------------------------
# Clustering Method
# --------------------------
```

```r
# Scaling
scaled_orders <- scale(filtered_orders[,c("OrderMonth", "Longitude", "Latitude")])
# Distance Matrix
dist.mat <- dist(scaled_orders, method = "euclidean")
# Perform the clustering
cluster.method <- hclust(dist.mat, method = "average")
# Cut the clusters at 5
cut.cluster.method.5 <- cutree(cluster.method, 5)

# Average Linkage: Based on the elbow method we should use 5 clusters, using
# this plot below as a visual. NOT INCLUDED because it was unnecessary information
# as described in Dr. Lucero's instructions.
fviz_nbclust(scaled_orders, hcut, method = "wss")

# Turn the clusters into a variable and attach to the data frame
cluster.factors <- as.data.frame(cut.cluster.method.5)
filtered_orders$Cluster <- as.factor(cluster.factors$cut.cluster.method.5)
```

Plot 3

```r
ggplot(filtered_orders,
       aes(x = Longitude,
           y = Latitude,
           color = Cluster)) +
  geom_point() +
  labs(title = "Plot 3: Latitude vs. Longitude, w/ 5 Clusters",
       color = "Cluster")
```

Plot 4

```r
# Cut the clusters at 4 instead of 5
cut.cluster.method.4 <- cutree(cluster.method, 4)

# Turn the new clusters into a variable and attach to the data frame
cluster.factors <- as.data.frame(cut.cluster.method.4)
filtered_orders$Cluster <- as.factor(cluster.factors$cut.cluster.method.4)

ggplot(filtered_orders,
       aes(x = Longitude,
           y = Latitude,
           color = Cluster)) +
  geom_point() +
  labs(title = "Plot 4: Latitude vs. Longitude, w/ 4 Clusters",
       color = "Cluster")
```

Table 1

```r
# Create appropriate names for new clusters
cluster.names <- c(
  "1" = "North East",
  "2" = "South West",
  "3" = "South East",
  "4" = "North West"
)
```

```
filtered_orders$ClusterName <- cluster.names[filtered_orders$Cluster]

# Create table listing cluster and new cluster name
filtered_orders %>%
  count(ClusterName) %>%
  mutate(Cluster = unique(filtered_orders$Cluster)) %>%
  arrange(-n) %>%
  kable(format = "markdown",
      label = "Orders Per Learned Cluster",
      align = c("l", "c", "c"),
      col.names = c("Cluster Name", "Number of Orders", "Original Cluster Number"))
```

Plot 5

```
ggplot(filtered_orders,
       aes(x = `Order Date`,
           color = ClusterName)) +
  geom_freqpoly(bins = 30) +
  labs(title = "Plot 5: Orders Over Time Using New Clusters",
       color = "Cluster")
```

Table 2

```
table(filtered_orders$Category, filtered_orders$ClusterName) %>%
  kable()
```