This is an excerpt of my written report from my senior data analytics capstone project (STAT 4911). Our client, Honda Motor Company, was interested in which products they were sourcing from multiple vendors and which vendors were they sourcing multiple products from. The intent was to optimize their purchasing, reducing duplicate purchases, and making their acquisition practices more efficient. I conducted network analysis and visualizations. Labels have been replaced with fake vendor/item names. This writing corresponds to the sample code submitted.

Network Analysis:

Network analysis is a technique that allows one to visualize the relationships between entities to show the overall relational structure of a community. Although a technique typically applied in the social sciences with social networks, network analysis can be reimagined and repurposed within Honda Procurement by looking at the connections between vendors and items. In doing so, opportunities for recommended sourcing events and cataloging will be revealed through the number of connections between these two entities.

In order to properly design and model this purchasing phenomenon, a two-mode undirected valued asymmetric network was chosen. Two-mode networks show the relationships between two types of entities. In this case, the two different entity types are vendors and items and are represented by nodes. When an item is purchased from a vendor, an edge connects these nodes. This edge does not represent the flow or passing of something, therefore, it is undirected. However, the edge will represent how many times an item is purchased from a vendor, with thicker edges representing more items being bought from that vendor and thinner edges representing less items. This is what makes the network valued. Lastly, there can only exist edges between vendors and items; vendor-vender and item-item relationships cannot exist. Therefore, an asymmetric network is proposed.

In order to build a subset of data that is suitable to build networks, the trimmed dataframe must be put into a wide format. This requires having rows that represent vendors, columns that represent items, and the numeric values that fill the cells of their intersection represent how many times that item was purchased from that vendor.

vendor_id	Std Ground Freight	Warehouse Rental	Fuel, Gas, Diesel	Returnable/Fabricate
2000006428	1	1	0	0
2000021774	0	0	11	0
2000037410	0	0	0	4

Figure 8: Example of wide dataframe

```
#Getting Purchasing counts
purchaseCounts <- completeHondaDataNew %>%
  group_by(vendor_id, `LEVEL 3`) %>%
  summarise(count = n()) %>%
  ungroup()

#This puts the dataframe into wide format
PreNetworkHonda <- purchaseCounts %>%
  pivot_wider(names_from = 'LEVEL 3', values_from = count, values_fill = 0)

#This creates the network
NetworkHonda <- ggnetwork(as.matrix(PreNetworkHonda), layout =
"fruchtermanreingold")
head(NetworkHonda)
```

Using the `ggnetwork` package in RStudio, a simple starting network is able to be created. However, in order to visually create a two mode network (i.e., make vendor nodes one color and item nodes another), the nodes need to be explicitly coded as a vendor or item node, then manually construct the nodes, and finally build the network. This is shown below.

```
#build each node type
vendorNodes <- data.frame(id = unique(purchaseCounts$vendor_id), type =
"vendor_id")
level3Nodes <- data.frame(id = unique(purchaseCounts$`LEVEL 3`), type = "LEVEL 3")
allNodes <- rbind(vendorNodes, level3Nodes)

#Get edges
edges <- purchaseCounts %>%
  mutate(from = vendor_id, to = `LEVEL 3`)

#Build network
NetworkHonda <- ggnetwork(edges, layout = "fruchtermanreingold", nodes = allNodes)
```

The two-mode network can now be visualized as follows:

```
netVisNoLab <- ggplot(NetworkHonda, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(linewidth = 0.35) +
  geom_nodes(size = 0.75, aes(color = ifelse(vertex.names %in%
unique(purchaseCounts$`LEVEL 3`), "LEVEL 3", "vendor_id"))) +
  scale_color_manual(values = c("LEVEL 3" = "blue", "vendor_id" = "red")) +
  labs(caption = "Blue: Item Description, Red: Vendor ID") +
  theme_blank(legend.position = "none")
```

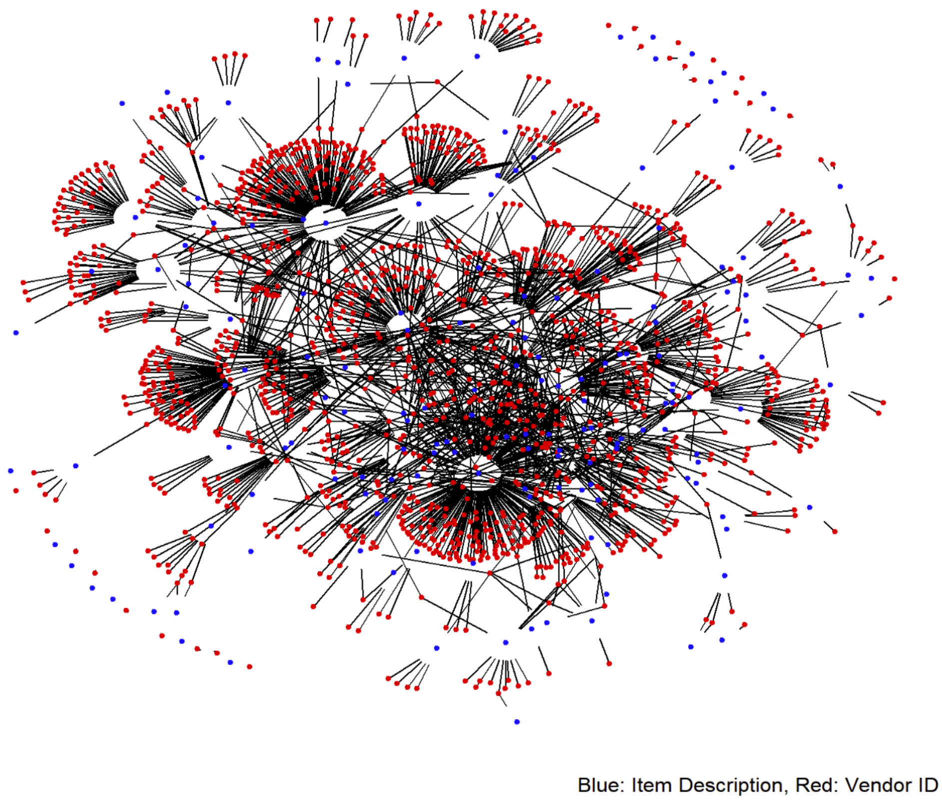

Figure 9: Network Visual

As shown above, the network depicts vendors, items, and which items are being bought from whom. What makes these networks useful is their ability to visualize similar purchasing patterns and reveal opportunities for potential sourcing events and cataloging. When using the “Fruchtermanreingold” method, the distance between nodes is maximized. Maximizing this distance requires the nodes with similar connections to be placed nearer to one another. It’s through this method where nodes with less connections are placed on the outskirts of the visualization, and those nodes with the most connections are shown nearer to the center.

Revealing patterns for potential sourcing events and cataloging is demonstrated through flower-like clusters and knotted clusters respectively. For example, in the top left quadrant, there is an item node surrounded by over 150 vendor nodes, creating a flower-esque visual. This item node is [Fake Item] and would be a great opportunity for a sourcing event as it is a singular item with many vendors. On the contrary, where there is large knotting, this would be an opportunity for a catalog. Knotting is occurring in this space as there is a much larger frequency of vendors than there is items. So, when there is a vendor that is connected to many items, it creates a "black hole" almost as the vendor is trying to connect to many items and in doing so, pulls in other vendors connected to those items. In the bottom right of the network, there exists one of these knotted clusters. Within this knot is [Fake Vendor] which is connected to over 10 distinct items, making it potentially useful for cataloging.

Additionally, data from the network can be extracted and visualized in other formats. The Following charts show the items bought from the most number of vendors, vendors that bought the most distinct items, and vendors that bought the most total items. Only the top 15-18 items or vendors are shown, making those items on the first visualization recommended for sourcing, and those vendors on the latter two visualizations recommended for cataloging. The following visualizations and code is below.

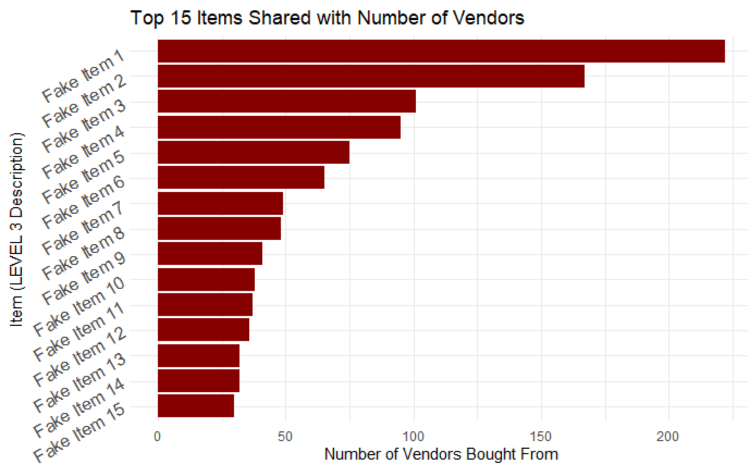

Figure 10: Items bought from many vendors

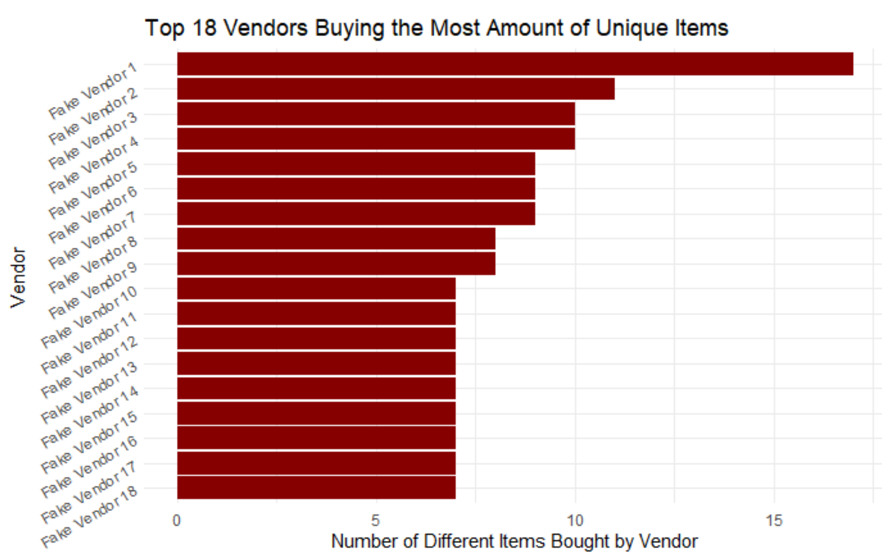

Figure 11: Vendors buying most distinct items

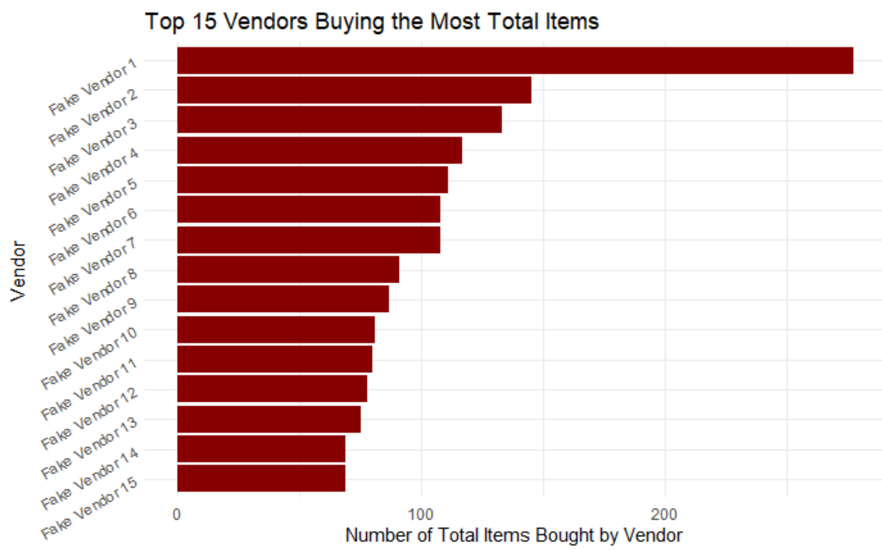

Figure 12: Vendors buying the most total items

```
#Extract relevant counts
same_vendor_many_items <- NetworkHonda %>%
  group_by(from) %>%
  summarize(same_vendor_many_items = n_distinct(to))

same_item_many_vendor <- NetworkHonda %>%
  group_by(to) %>%
  summarize(same_item_many_vendor = n_distinct(from))

#Same items many vendors
ggplot(data = same_item_many_vendor %>% top_n(15,same_item_many_vendor),
aes(reorder(to, same_item_many_vendor), y = same_item_many_vendor)) +
  geom_col(fill = "darkred") +
  ggtitle("Top 15 Items Shared with Number of Vendors") +
  xlab("Item (LEVEL 3 Description)") +
  ylab("Number of Vendors Bought From") +
  coord_flip() +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 30, size = 12), axis.title.x =
element_text(size = )) +
  #scale_x_discrete(labels = c("Put new labels here")) #Use this line to change
labels

#Same vendor many unique items
ggplot(data = same_vendor_many_items %>% top_n(15,same_vendor_many_items),
aes(reorder(from, same_vendor_many_items), y = same_vendor_many_items)) +
  geom_col(fill = "darkred") +
```

```
  ggtitle("Top 18 Vendors Buying the Most Amount of Unique Items") +
  xlab("Vendor") +
  ylab("Number of Different Items Bought by Vendor") +
  coord_flip() +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 30)) +
  #scale_x_discrete(labels = c("Put new labels here")) #Use this line to change
labels

#Same vendor many total items
ggplot(data = total_products_bought %>% top_n(15,total_products_bought), aes( x =
reorder(vendor_id, total_products_bought), y = total_products_bought)) +
  geom_col(fill = "darkred") +
  ggtitle("Top 15 Vendors Buying the Most Total Items") +
  xlab("Vendor") +
  ylab("Number of Total Items Bought by Vendor") +
  coord_flip() +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 30)) +
  scale_x_discrete(labels = c("Put new labels here")) # Use this line to change
labels
```

Although the network approach is able to show both "same vendor many items" and "same item many vendors" relationships with ease, there are some limitations. The most notable being the difficulty in identifying the names of the nodes of interest within the visual. This graph is not interactive, meaning to get this information, one must dive into the network dataframe and extract it almost manually, as opposed to just looking at the graph. Additionally, there is difficulty in including new characteristics within these networks. Because the primary format of this data is wide, including just vendor and items, when wanting to look at other characteristics one must manually construct the nodes as done above.

To build upon this network, community analysis is recommended as it would be able to more formally classify vendors and items with similar purchasing patterns, allowing for more general and wide scale cataloging and sourcing recommendations.