# Fetch Rewards Coding Exercise - Data Analyst

Arthur Li-Chuan Lee

9/24/2021

## Hello!

Thank you for considering me for the Data Analyst role at Fetch Rewards! I had a blast working with the data sets that you provided for this assignment! The data that you provided was quite a challenge to work with, but I was able to find solutions to many of the issues surrounding the data set and I'm excited to share the results of the project with you!

### First: Review Existing Unstructured Data and Diagram a New Structured Relational Data Model

The first thing to immediately notice about the data is that its missing syntax for a JSON file! I needed a solution to fix the issue faster than manually adding commas to the files for a couple of hours, so I came up with the idea of using a text replacement method to make mass text edits to each file, then using a JSON beautification tool to validate the data between edits and search for the next set of problematic syntax. With making about 4-6 edits to each file, I was able to reformat all three JSON files and import each to a SQL database.

I chose to use MS SQL Server for this project due to its better support for handling JSON files. My usual SQL flavor of choice, MariaDB on RHEL 8, ended up not being a great choice for handling JSONs. The supposed package for the CONNECT storage engine that would enable importing of JSON files was missing from the Red Hat repository entirely for some unknown reason, and the alternative solution, the JSON_TABLE function, is only available in the current Alpha build of MariaDB, so I had to look elsewhere.

I was able to use the OPENROWSET function in SQL Server to do a bulk import of each JSON file into my database. The schema for the initial import can be found in the **fetchdanalyst-schema.sql** file, and the data import process is located in **fetchdanalyst-openrowset.sql**. If you intend on trying to run the code yourself to follow along, I recommend executing the code in pieces rather than all at once to avoid upsetting SQL Server.

The simplified database schema I came up with can be found in **fetchdanalyst-diagram.jpg**. I oversimplified the receipts table on this diagram by cutting out variables that didn't seem relevant to this project; however, if the intent of the database is for it to be utilized by the team at large, I may have kept additional variables.

### Second: Write a query that directly answers a predetermined question from a business stakeholder

The code for this section can be found in **fetchdanalyst-cleanse.sql** (data cleansing) and **fetchdanalyst-bq.sql** (business question analysis). I started this section by doing some data cleansing. I created two subset tables (brandsTrimmed & usersUnique) where I pulled only the variables I needed for the project, deleted duplicate rows, changed data types for columns that needed to not be NVARCHAR(MAX), and converted dates from UNIX Epoch time to Date format.

In the final part of the cleanse file, I manually annotated the brandCode column for the month of February 2021 to improve the quality of the analysis since much of this column for that month was NULLed.

**What are the top 5 brands by receipts scanned for most recent month?**

- The top 2 brands ranked by receipts scanned for the month of February 2021 are Sargento (4) and Doritos (3). The next 4 brands, Capri Sun (2), Heinz (2), Mission (2), and Suave (2), all had an equal number of receipt scans.

**How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?**

- The top 5 ranking for Jan 2021 is almost entirely different from the top 5 ranking in Feb 2021. The top 5 brands ranked by receipts scanned for the month of January 2021 are Hy-Vee (228), Ben and Jerry's (50), Dole (47), Doritos (41), and Borden (39). Doritos is the only brand that appears in the rankings for both months.

## Third: Evaluate Data Quality Issues in the Data Provided

Here are the data quality issues I identified:

- There are several duplicate barcode entries in the brands data. I worked around this problem by checking these against the receipts table and trimming the unnecessary data (brandsTrimmed), but we should validate the barcode data in the future to avoid the possibility of introducing bad data to our database.
- Just by placing the barcode column in the brands table and the barcode column in the receipts table side-by-side in the, its obvious that the brands table is incomplete and only contains the subset of barcodes that starts with "511111". What happened to the rest of the brands data?

(I used the 'dlookr' package in R to make diagnosing the data set a bit more organized.)

```
library(dlookr)
library(tidyverse)
brandsTrimmed = read.csv("wd/csv/fetchrewards/brandsTrimmed.csv", fileEncoding = "UTF-8-BOM") %>% as.tb
receipts = read.csv("wd/csv/fetchrewards/receipts.csv", fileEncoding = "UTF-8-BOM") %>% as.tbl
usersUnique = read.csv("wd/csv/fetchrewards/usersUnique.csv", fileEncoding = "UTF-8-BOM") %>% as.tbl
```

- Just over half the data in the brands table is missing any entry for categoryCode and topBrand. categoryCode data is available in the category column, it just needs to be interpreted and annotated in categoryCode.

```
brandsTrimmed[brandsTrimmed == "NULL"] = NA
brandsTrimmed$topBrand[brandsTrimmed$topBrand == 0] = F
brandsTrimmed$topBrand[brandsTrimmed$topBrand == 1] = T
brandsTrimmed$topBrand <- as.logical(brandsTrimmed$topBrand)

diagnose(brandsTrimmed)
```

```
## # A tibble: 9 x 6
##   variables    types     missing_count missing_percent unique_count unique_rate
##   <chr>        <chr>             <int>           <dbl>        <int>       <dbl>
## 1 oid          character             0             0           1154     1
## 2 barcode      numeric               0             0           1154     1
## 3 brandCode    character           234            20.3          885     0.767
## 4 category     character           154            13.3           24     0.0208
## 5 categoryCode character           637            55.2           15     0.0130
## 6 cpgoid       character             0             0            194     0.168
## 7 cpgref       character             0             0              2     0.00173
## 8 name         character             0             0           1144     0.991
## 9 topBrand     logical             603            52.3            3     0.00260
```

- The receipts table is missing data for more than half of barcodes (58.14% missing) and brandCodes (60.38% missing). In other words, for more than half of the data we aren't able to determine what brand the item purchased was due to the brandCode data in the receipts table being unannotated, and we aren't able to use the brandCode data in the brands table to support the receipts table either due to the barcode data in the receipts table being missing or unannotated (assuming many of the barcodes weren't missing from the brands table as well). We need an annotator to annotate this data for us so we can rerun our analysis to get more reliable results.

```
receipts[receipts == "NULL"] = NA
receipts$bonusPointsEarned <- as.integer(receipts$bonusPointsEarned)
receipts$totalPointsEarned <- as.integer(receipts$totalPointsEarned)
receipts$purchasedItemCount <- as.integer(receipts$purchasedItemCount)
receipts$finalPrice <- as.double(receipts$finalPrice)
receipts$totalSpent <- as.double(receipts$totalSpent)

diagnose(receipts)
```

```
## # A tibble: 37 x 6
##    variables       types  missing_count missing_percent unique_count unique_rate
##    <chr>           <chr>          <int>           <dbl>        <int>       <dbl>
##  1 oid             chara~             0            0            1119     0.152
##  2 bonusPointsEar~ integ~          1401           19.0            13     0.00176
##  3 bonusPointsEar~ chara~          1401           19.0            10     0.00135
##  4 totalPointsEar~ integ~          1128           15.3           114     0.0154
##  5 purchasedItemC~ integ~           484            6.56           51     0.00691
##  6 item            chara~           440            5.96          460     0.0623
##  7 barcode         chara~          4291           58.1           569     0.0771
##  8 brandCode       chara~          4457           60.4           244     0.0331
##  9 competitorRewa~ chara~          7106           96.3            31     0.00420
## 10 description     chara~           821           11.1          1890     0.256
## # ... with 27 more rows
```

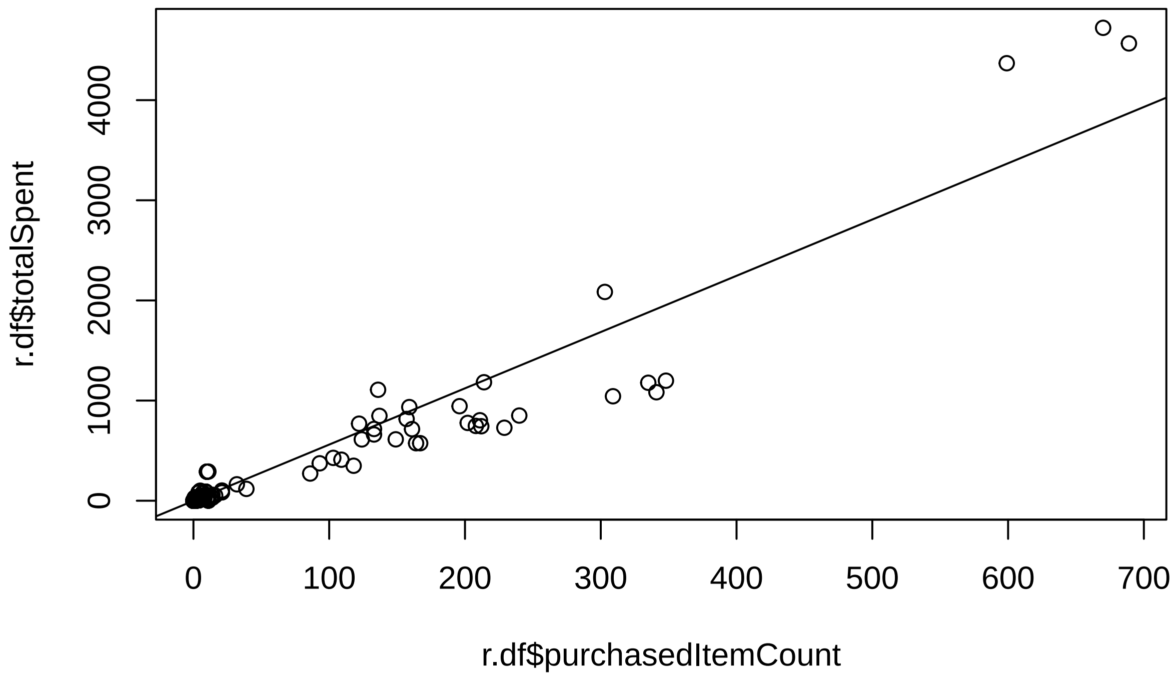
- Diagnosis function is trying to tell us that a significant portion of the receipts table, most notably 1290 results in the totalSpent column, are outliers.

```
diagnose_numeric(receipts)
```

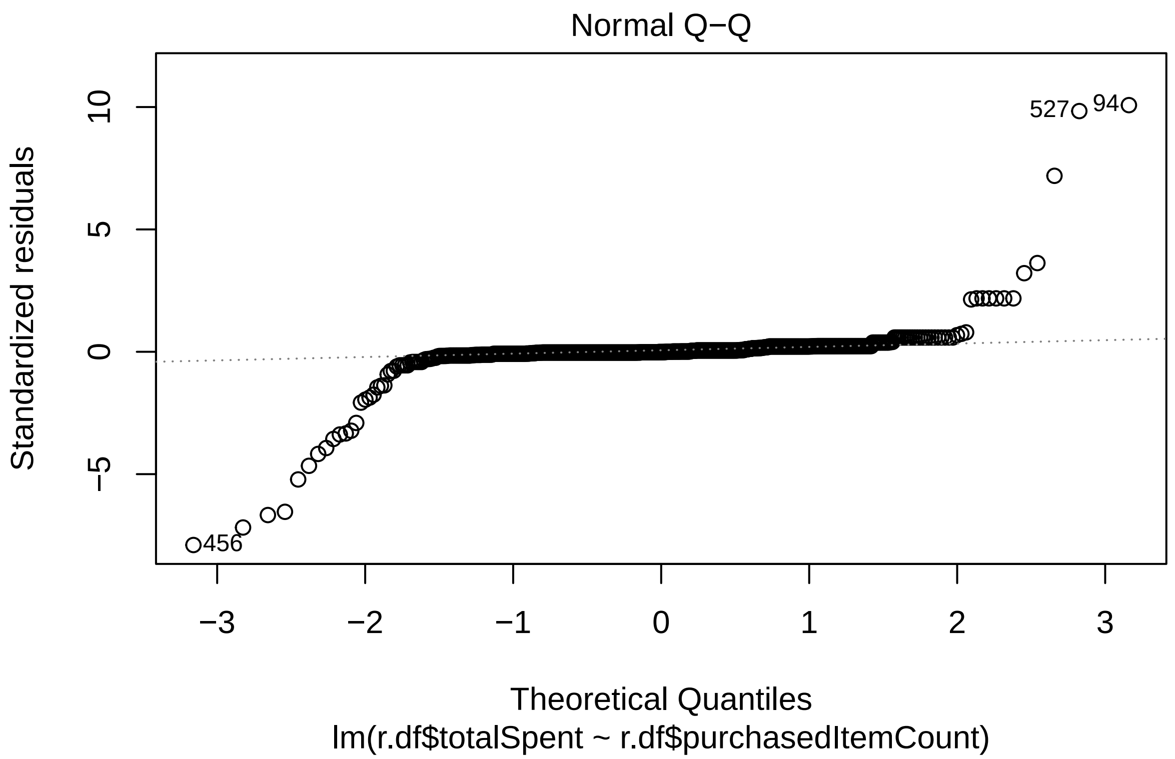
```
## # A tibble: 5 x 10
##   variables            min     Q1   mean median     Q3    max  zero minus outlier
##   <chr>              <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <int> <int>   <int>
## 1 bonusPointsEarned      5 750    6.26e2 7.5 e2 7.5 e2   750      0     0    1375
## 2 totalPointsEarned      0 750    2.18e3 1.45e3 2.68e3 10199      6     0     654
## 3 purchasedItemCount     0  93    2.41e2 1.67e2 3.35e2   689     25     0       0
## 4 finalPrice             0   2.29 7.87e0 4.28e0 9.99e0   442.     4     0     476
## 5 totalSpent             0 374.   1.37e3 7.77e2 1.18e3  4722.    25     0    1290
```

- Plotting the data shows a huge cluster of points for receipts with low purchasedItemCount and what appears to be a jump up to receipts with approximately 100 to 350 purchasedItemCount. The top 3 receipts are another jump up to 600+ purchasedItemCount.

```
r.df <- data.frame(receipts$oid, receipts$totalPointsEarned, receipts$purchasedItemCount, receipts$tota
r.df <- rename(r.df, oid = receipts.oid, totalPointsEarned = receipts.totalPointsEarned, purchasedItemC
r.df <- distinct(r.df)
plot(r.df$purchasedItemCount, r.df$totalSpent)
r.lm <- lm(r.df$totalSpent ~ r.df$purchasedItemCount, na.action = na.omit)
abline(r.lm)
```

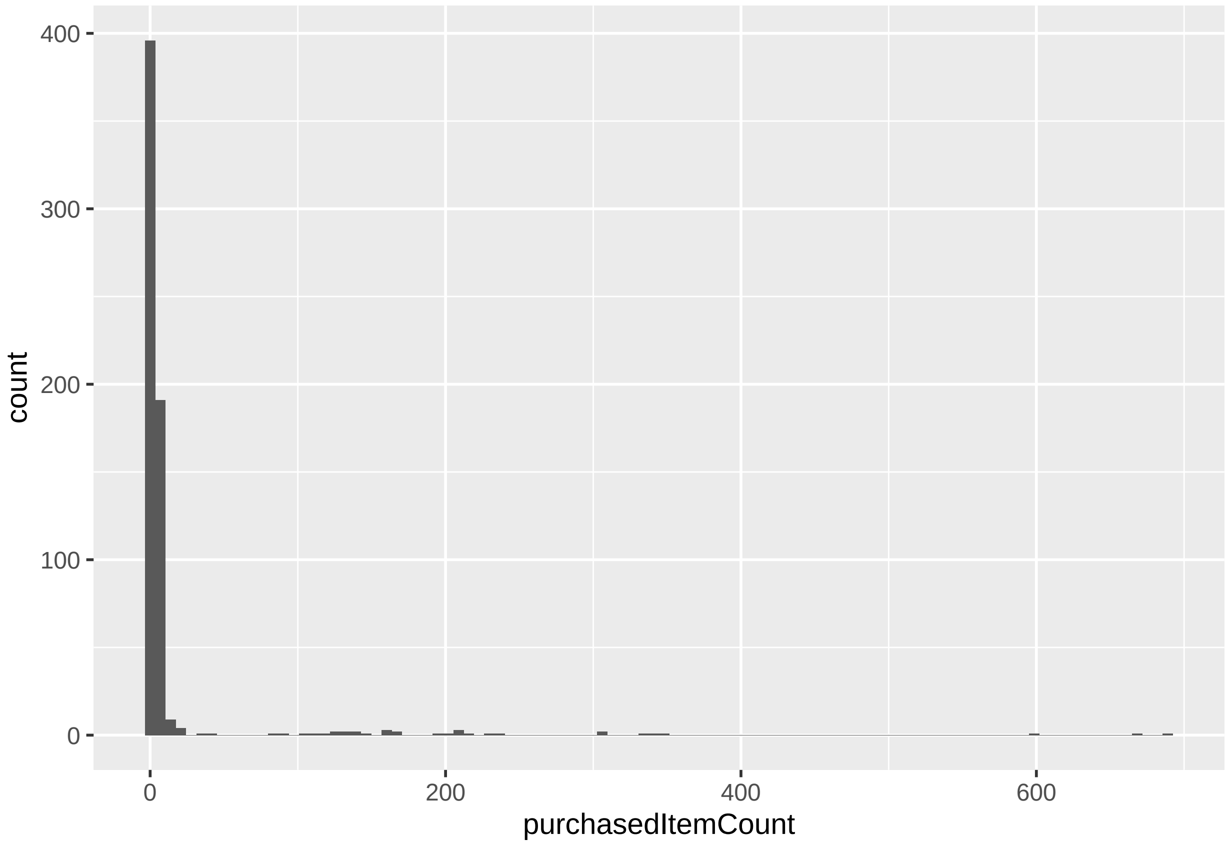


```r
plot(r.lm, which = 2)
```

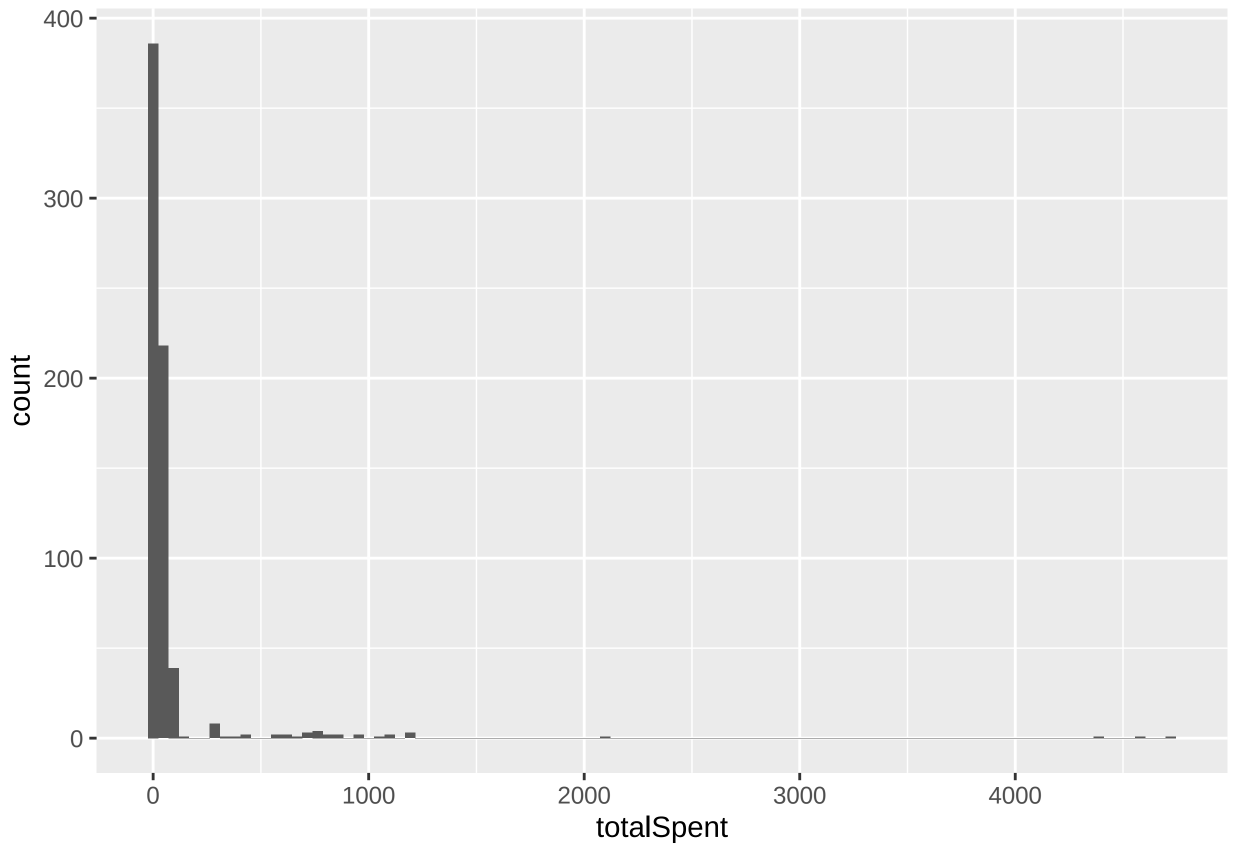


- Data is heavily left skewed.

```r
ggplot(data = r.df, aes(purchasedItemCount)) + geom_histogram(bins = 100)
```



```
ggplot(data = r.df, aes(totalSpent)) + geom_histogram(bins = 100)
```



- The users data shows data for only 8 states (+1 unique_count for NA values) and the majority of the data is Wisconsin users. Are Wisconsin users the primary user base, or is this only a subset of this data set?

```
usersUnique[usersUnique == "NULL"] = NA

diagnose(usersUnique)
```

```
## # A tibble: 7 x 6
##   variables    types     missing_count missing_percent unique_count unique_rate
##   <chr>        <chr>             <int>           <dbl>        <int>       <dbl>
## 1 oid          character             0            0             212     1
## 2 active       integer               0            0               2     0.00943
## 3 createdDate  character             0            0              42     0.198
## 4 lastLogin    character            40           18.9            31     0.146
## 5 role         character             0            0               2     0.00943
## 6 signUpSource character             5            2.36            3     0.0142
## 7 state        character             6            2.83            9     0.0425
```

Here's are the summary statistics for a linear regression to round out this section. Variance on initial glance seems pretty big, but would need to look a little deeper into it to verify. A simple linear model is likely not the right model to represent this data and I would likely need to use a more complex regression method, but this seemed outside the scope of the project so I didn't investigate further than this.

```
summary(r.lm)
```

```
##
## Call:
## lm(formula = r.df$totalSpent ~ r.df$purchasedItemCount, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -831.38   -3.34   -1.25   16.66 1004.61
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.27760    4.40258   -0.29    0.772
## r.df$purchasedItemCount  5.61848    0.07006   80.20   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 107.8 on 633 degrees of freedom
##   (484 observations deleted due to missingness)
## Multiple R-squared:  0.9104, Adjusted R-squared:  0.9103
## F-statistic:  6432 on 1 and 633 DF,  p-value: < 2.2e-16
```

```
anova(r.lm)
```

```
## Analysis of Variance Table
##
## Response: r.df$totalSpent
##                          Df   Sum Sq  Mean Sq F value    Pr(>F)
## r.df$purchasedItemCount   1 74798924 74798924  6431.9 < 2.2e-16 ***
## Residuals               633  7361345    11629
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
paste0(("Variance (purchasedItemCount): "), var(r.df$purchasedItemCount, na.rm = T))
```

```
## [1] "Variance (purchasedItemCount): 3737.39535010805"
```

```
paste0(("Variance (totalSpent): "), var(r.df$totalSpent, na.rm = T))
```

```
## [1] "Variance (totalSpent): 120485.594690691"
```

### Fourth: Communicate with Stakeholders

Here’s the message I drafted to send to the product owner:

Good morning [Product Owner]! Just wanted to reach out again because I had some concerns about the data you forwarded to me for the Jan/Feb 2021 receipt scan project. The data quality for the data set is pretty far from ideal and I wanted to voice this back to you as well as get some additional clarification on some of the elements of the data. Here’s what I noticed while I was putting together the database:

- The brand and barcode data you sent me only contains barcodes starting in “511111”. Can you ask the admin for me where the rest of the barcode data is?
- In the brand and barcode data I found a couple of duplicate barcode entries while I was validating the data. I was able to get around this problem, but we should check this barcode data more vigilantly in the future to keep users from putting bad data into our system.
- The brand and barcode data has a field for what category of product a brand belongs to and that field is full of data, but the category code field right next to it is missing most of its data. We need an annotator to interpret the categories and input the right category code. We also need someone to annotate the column for whether each brand is a “top brand” or not.
- In the receipts data, even though we have a lot of information about what products were purchased on these receipts, the diagnostic I ran is showing that we’re missing the barcodes and brand data for more than half of these products. We also need an annotator to look up these items online and find their barcodes. This should help us get much more accurate results.
- I plotted some of the points to compare how many items users bought to how much their total spend was on their receipts and found that most of the data is for 10 or less items - the data jumps quite a bit after that to a group of big spenders, then it jumps again to a group of 3 receipts with over 600 items. Do you know if there could be unintentional missing data that could account for these jumps or are we certain that this is all of our data?
- The majority of the user account data I received is data from Wisconsin. Is there supposed to be more receipt and user account data than what you sent or is Wisconsin our test market for this project?

I was also interested to know if there was any way for us to know if items purchased were refunded so we could add this to our data set? Perhaps enabling users to also upload refund receipts could allow for this. It may also be useful for us to start including the store name and location for each receipt in our data. This would allow us to analyze data on a store-by-store or chain-by-chain basis.

I also wanted to start a discussion with you about the data migration process for this project. The data that was forwarded to me is in JSON format and I’m storing it in a local SQL Server on my system. When we scale this data project up, if we continue to use JSON to migrate our data we will need a more reliable, faster, and accessible way to ingest it especially if we’re going to be bringing this project to the cloud with AWS. If we will be using Amazon Redshift, it luckily has a solution (called the SUPER data type) that seems to be a perfect fit for our needs, but I just wanted to verify with you whether Redshift is the final destination for our data or if we will using a different solution for our data warehousing needs.

Thanks in advance for taking the time to read my message and answer my questions! Dealing with the complexities of this project has been an interesting experience, but I’m enjoying discovering the solutions and workarounds that will allow us to achieve our goals!