****

**TCDS - Final Project**

**Forecasting purchases of items**

**per day and store**

**Data Science**

**FAVORITA** **Project Protocol**

**Authors:**

**Ariel Shafir & Tzvika Senderov**

**Feb 2019**

**Introduction**

Grocery stores have always faced the challenge of forecasting procurement and sales of consumable products. High forecasting will result in excess inventory being thrown into the trash. If the prediction is low, it will result in a loss of income mainly in products that are sold a lot (Popular).

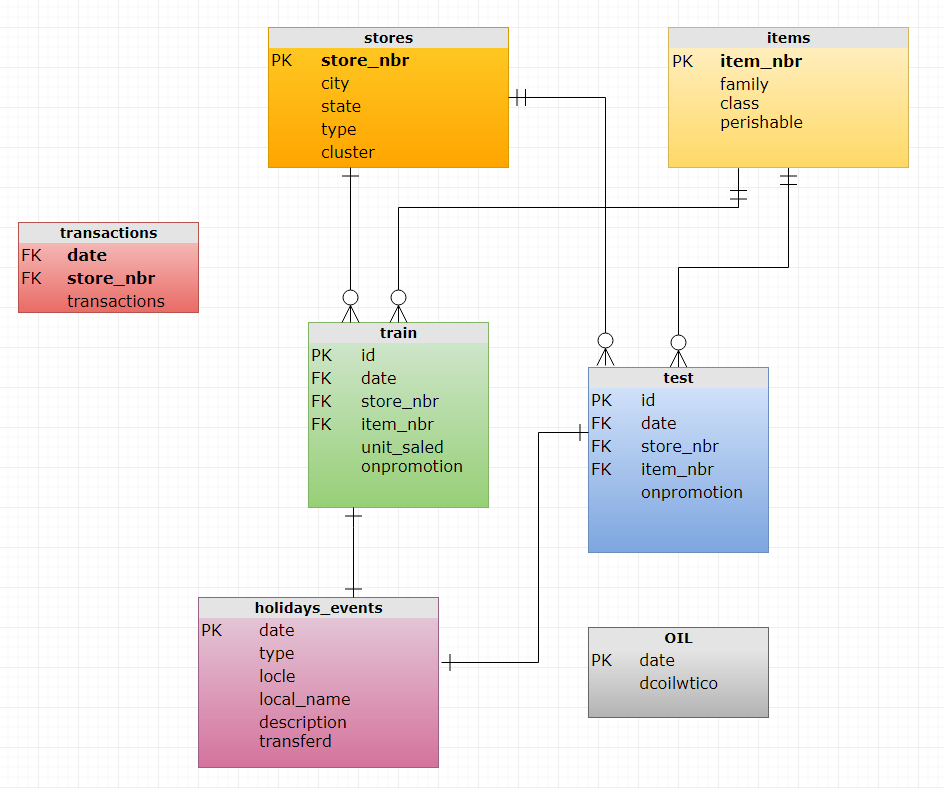
[Corporación Favorita](http://www.corporacionfavorita.com/) is a large Ecuadorian-based grocery retailer.

They operate hundreds of supermarkets, with over 200,000 different products on their shelves tackles this challenge on a daily basis.

The difficulty is greater when it comes to new products and seasonal products, which should be prepared accordingly according to the period of the year, which sometimes also takes into consideration unique storage conditions,

We want to see how using ML machine model will predict the quantity of products sold at a daily level in each store.

The data source originally from a Kaggle challenge. The Dataset has been provided by **Favorita corporation**.



**Methodology (project design)**

**Data**

The data source was taken from Kaggle Competition – *"Corporación Favorita Grocery Sales Forecasting,* the challenge: *can you accurately predict sales for a large grocery chain?"*

These are the csv files we used:

### train.csv

Training data, which includes the target unit\_sales by date, store\_nbr, and item\_nbr and a unique id to label rows onpromotion column tells whether that item\_nbr was on promotion for a specified date and store\_nbr   Negative values of unit\_sales represent returns of that particular item.

stores.csv

Store metadata, including city, state, type, and cluster where   cluster is a grouping of similar stores.

### items.csv

Item metadata, including family, class, and perishable -: Items marked as perishable have a score weight of 1.25.

### oil.csv

Daily oil price. Includes values during both the train *and* test data timeframe. (Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices.)

### holidays\_events.csv

Holidays, Bridge and Events

From Google we took information about the population <https://www.worldatlas.com/articles/biggest-cities-in-ecuador.html>

We build a function that calculate the payment day and payment day +1

We decided to split the data and avoid in advanced some date we had reason to assume it will influence the data. We avoided 4 months of activity from the earthquake and the beginning of the data which a field is missing until Apr 2014. Finally, we decided to divide the Train file to three-hole years- Train, Dev and Test as describe bellow and then we joined the date from the other tables as we will detail later, for the EDA faze.

The **Train** dataset includes activity of one year. Dates between 1/4/2014 and 31/3/2015

The **Dev** dataset includes activity of one year. Dates between 1/4/2015 until31/3 /2016

The **Test** dataset includes activity of one year. Dates between 1/7/2016 until 31/7/2017

Each dataset includes a hole year with all Holidays and events.

\*\* because of our assumption and decision of splitting the date in advanced to train dev and test for now

**EDA- Exploratory data analysis**

**Clear outcome variable definition**

Our prediction will be how many units was sold of each item per day and store over a hole year.

The data contains information about

* Number of units were sold of each item per store and day
* Daily oil price
* Table with holydays and events in Ecuador (national, state and city) actual dates it took place.
* Information of items like category, did it was on promotion? and is it a perishable item?
* Information about the stores like type of store cluster of stores

The original dataset includes more than 4000 items and more than 120,000,000 transaction. In our datasets we decided to forecast only 250 most sailing items. we partitioned the data by years, so each year data set has about 3 million rows.

We know that in April 2014 a major earthquake took place in Ecuador. this information is very important because the behavior of customer probably was abnormal, and it could bias the prediction. (due to buying a lot of products which are necessary for existence and less of other products which are not necessary at times like that)

For that reason, we didn't want to insert this period of few months to our model.

Assumptions and decisions:

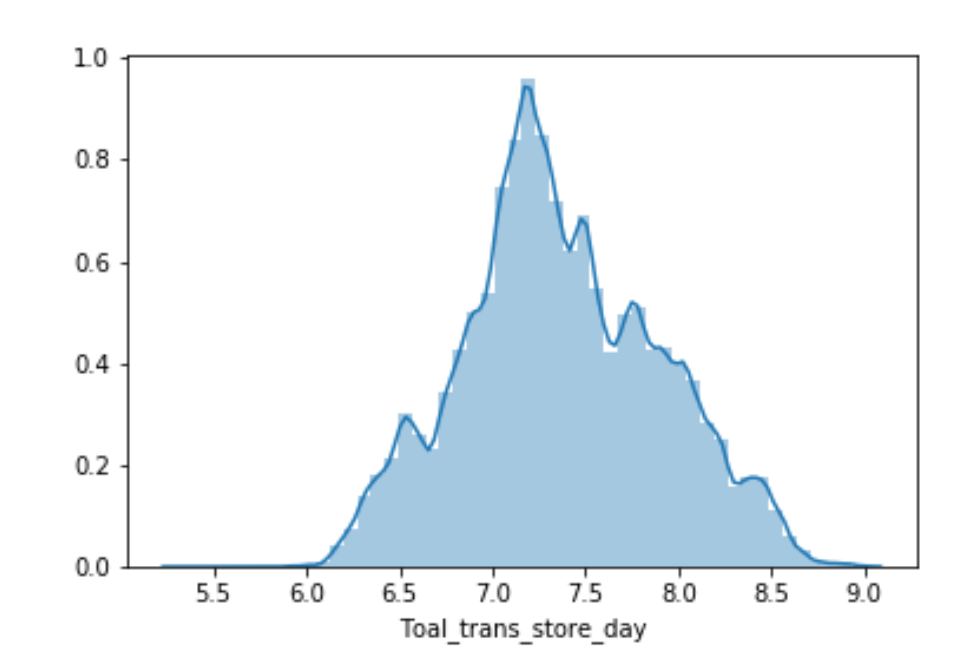
* We assumed that there is a seasonableness in the behavior of customers and items being sold during the year so we defined in advanced that our train data as well as our dev and test data will be as a whole year cycle.
* We believe that it will good enough for prediction of the next years and there is in the data.
* The data before 01/04/2014 does not include a value in **onpromotion** field, and because we assume it is significant, we decided to avoid data without it.
* We assume there is a correlation between the region of the store and types of products people are buying more.
* In Ecuador there is a payday twice a month at the 15th and at end of month, we assume its influence is major, so we add parameters which indicates payday, payday -1, payday -2, payday +1 and payday +2 days.
* We add holydays indication by national, state and city to check for influence.
* oil prices influence very much at Ecuador economy, so we add daily price as a parameter as well as prices 15 and 30 days earlier and the ratio of change between.

The first analyze was done in SQL Server

We build view to concatenate the data that we thought is relevant

In this way we built at the end one view that contain all the relevant information

TOTAL TRANSACTIONS PER DAY AND STORE looks as its normally distributing (after log) – do we need to add a column with a log value to flat file?



I tried to do the same with our main variable **weight\_unit\_sale** with no success so far. We will do it later

**Variable engineering**

We changed all parameters that were as type object into category type

We changed dates into date type

All categories we add as columns as dummies. One hat encoding. Some we changed in the DB view before EDA and the rest such **month** and **day\_of\_**week and **season** we will do later. We will check about **store\_nbr**, **city**, **state** if needs to slit as well

* Type category (there is 5 types of store)
* Family category (there is 32 types of Family products)
* Cluster category (there is 17 types of store clusters)

For the category parameters which we split to columns in the DB view we built a table for decoding.

We considered the type of holydays in our view, there are some types of holydays at Ecuador some are local holydays, region holydays and national holydays

We calculated with SQL function if the date is a payday, one day after the payday one day before payday and two days before payday.

Data taken from the internet:

The population of each city which have a store in our data.

(Python engineering)

We calculate the percentage of city population out of all population to normalize the parameter values.

We calculate the oil price 15 and 30 days before purchasing of the item

We calculate how much (%) the oil price changed compare 15 and 30 days before purchasing of the item

Column "**weight\_unit\_sale**" is a calculated column of units sold, in case it is perishable we multiplied the units by 1.25 else we multiplied the unit by 1

This is the order list for the most popular family category item

Family\_cat\_13

Family\_cat\_9

Family\_cat\_29

Family\_cat\_31

And in those categories:

1,2,3,5,7,14,15,17,18,19,20,22,24,27,28,32,33 no items was sold and we will not take those category as features.

This is the order list for the most popular cluster store category

Cluster 3,10,14,15,13

This is the order list for the most popular store type

Type D, C, A, B, E

**Missing values - determination of MCAR-MAR-MNAR and treatment**

Description function and df.isna().sum()

Showed about 30% nulls in oil price column.

Most of the null are in **MNAR** missing not at random because it is on weekends, so we copied the last known value.

Few days had missing values which wasn’t weekends and it looks like there is no seeing reason, so we believe it is **MCAR** missing completely at random. we treated it the same way as we did for the other missing dates, we copied the last known value.

We extracted the date and oil price columns from our flat file table. Then imported table of oil prices at the month before our data begins and added missing dates manually.

We built a column contains the oil price day before (with shift function) and replaced with the nan values. We ran it three times to replace all nulls.

Then we add columns of the price 15 days ago and 30 days ago and calculated the ratio of change between each one and actual date price and finally merged it with our data.

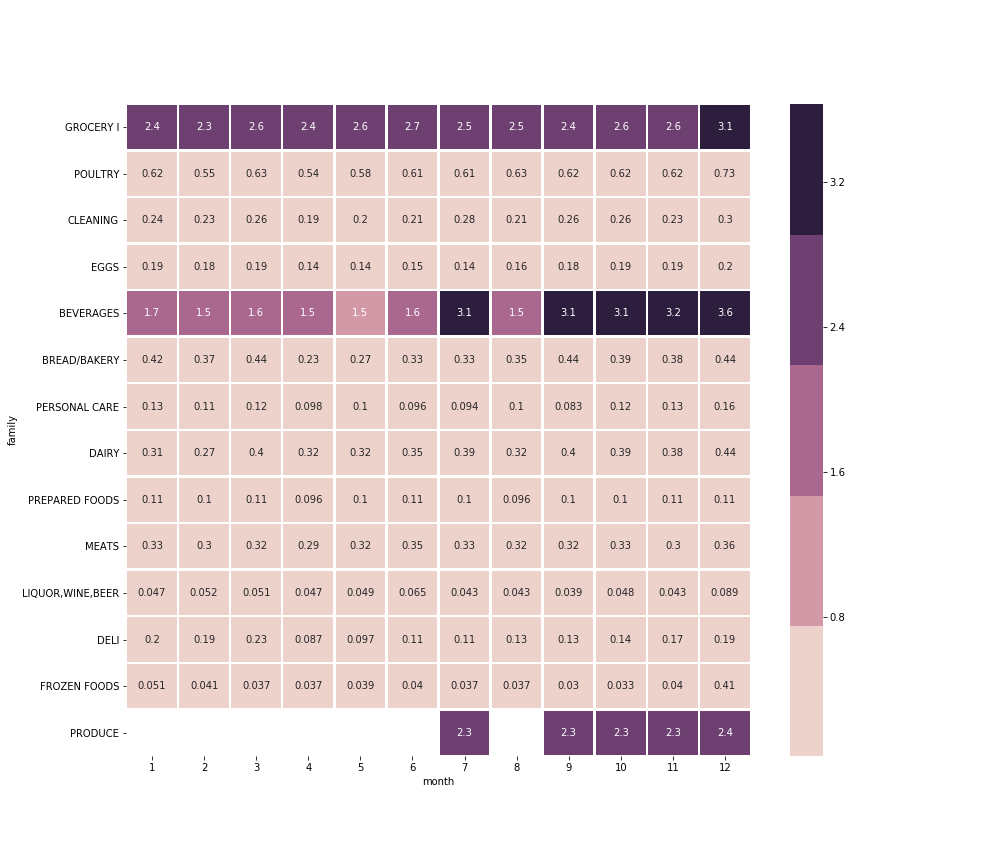
**Outliers determination and treatment**

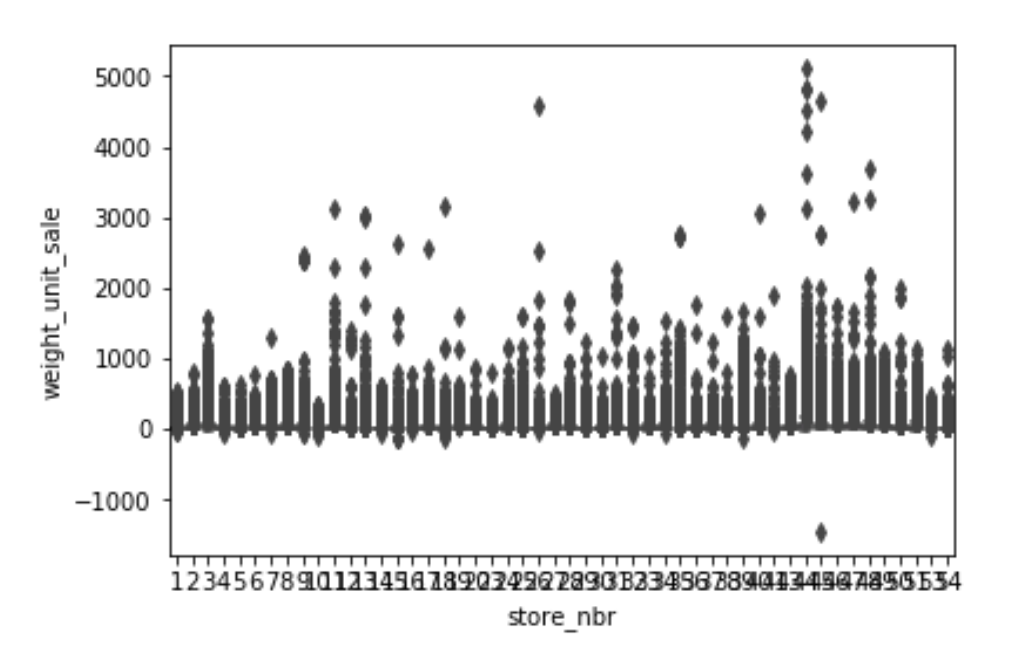
**Here we show some graphs which can implies the distribution of the data and outliers’ values**

To get values on the same range we took the percentage out total of sales

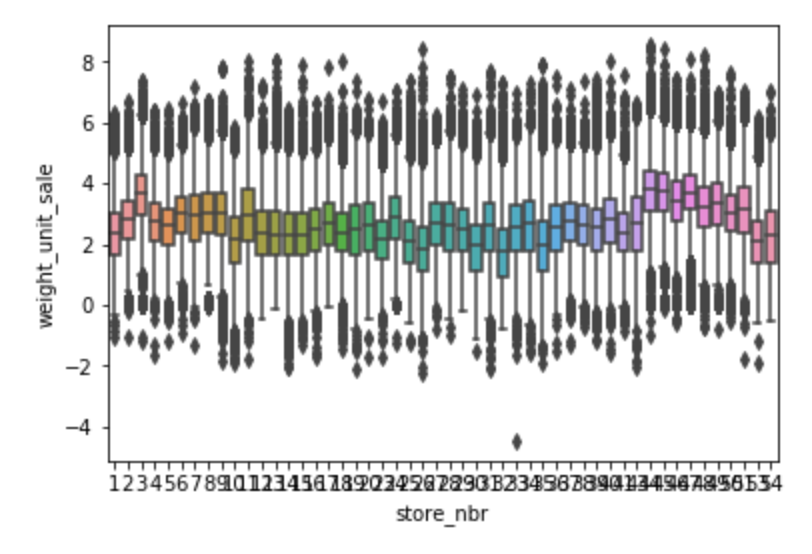
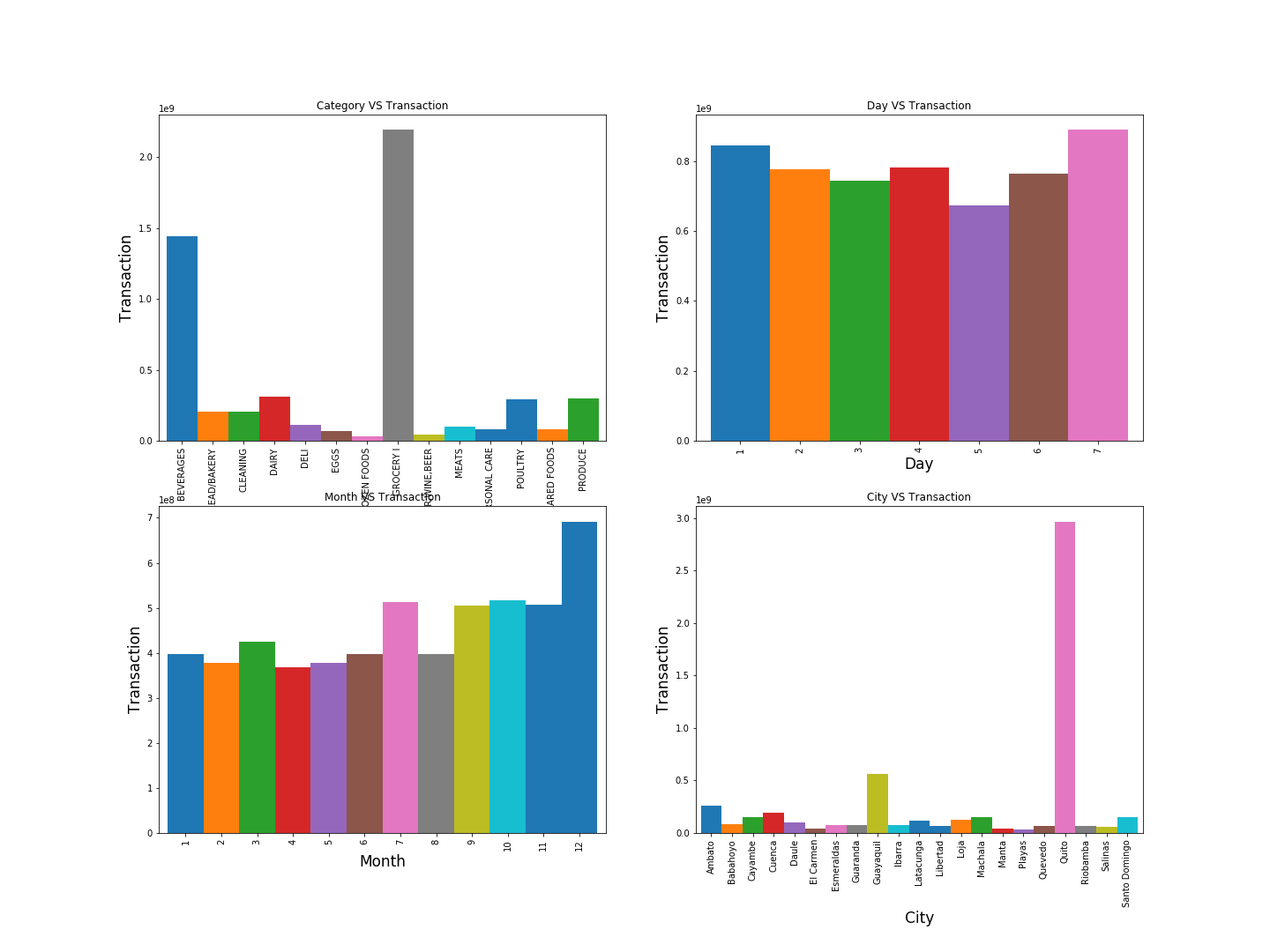
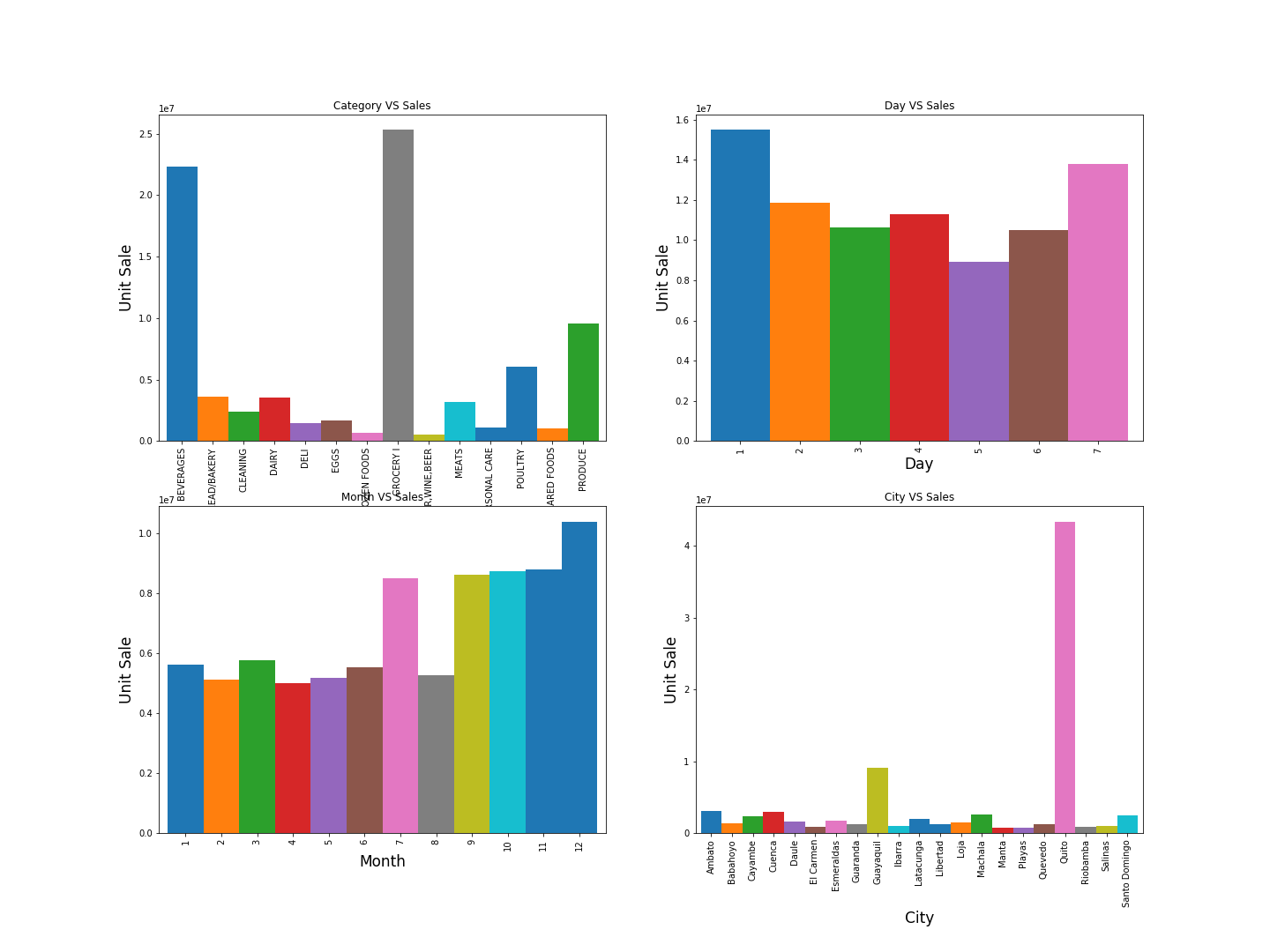
We see in this plot that most of categories are in the same range of value except of grocery category which seem to be higher than the rest

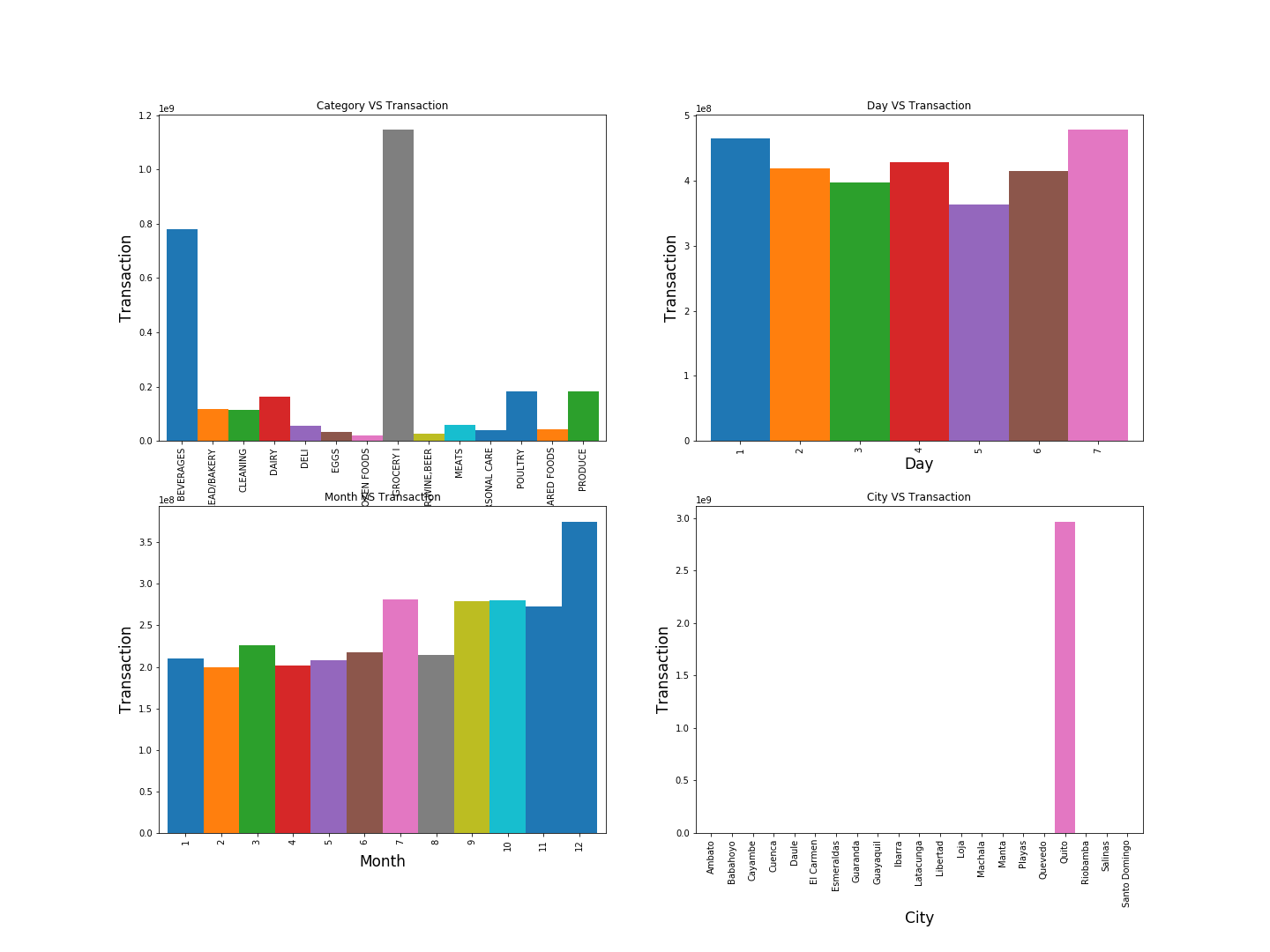
Another thing we see that beverages category seems to be which a lot of variance in their group

****

****in this boxplot graph we can see there are suspicious values for outliers.

After we activated log on it looked better, less points looked as outlier, but we must check values later.

after log function over field **whight\_unit\_sale** we can see now that each store has a little different behavior and still some suspicious points for outliers.  
  
****we can see the behavior is very similar between amount of unit sold by day, category, month and city and the amount of transitions by city, day, month and ****category in Respectively.  
  
  
even when we filtered only a specific city the distribution looks very much a like

****  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**for the multivariant part we encountered some difficulties.**\*\* extensive amount of data which the outlier function couldn’t handle in our computer because of hardware limitations, for now we will show only sample in this section and will fix it asap. We will reduce data until it will work and then we will ran everything over again and update document.

Probably our solution will be reduction data amount by choosing less items for our datasets.

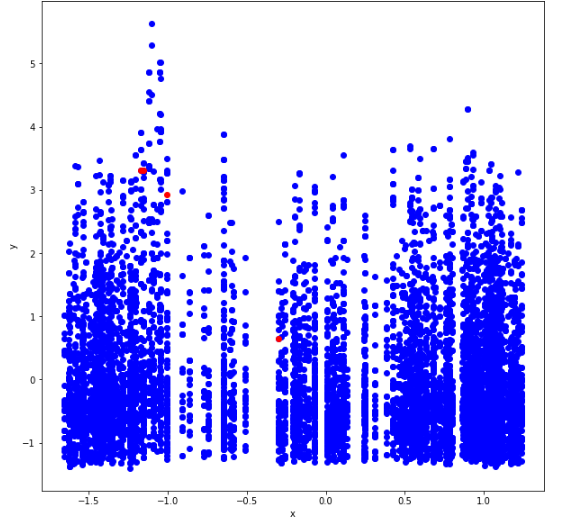
**Most of our variables are categories**

We found the following information

* we have a max value of **weight\_unit\_sale** as 5103 which is extremely different the std is 44

when we manipulated **weight\_unit\_sale** to be log the outliers was much less significant

**our data sample included 10,000 rows and showed only 4 values as outliers’ inputs to dbscan**

****

**TABLE ONE**The rows in red has no true (1) values (after reducing data to 250 items)

|  |  |  |
| --- | --- | --- |
| **Variables** | **Categories** | **Population** |
| Individuals | n | 2988132 |
| month | Mean (SD) | 6.90 ( 3.50) |
| month | Median (IQR) | 7.00 ( 4.00- 10.00) |
| month | Missing (%) | -- |
| day\_of\_week | Mean (SD) | 4.00 ( 2.00) |
| day\_of\_week | Median (IQR) | 4.00 ( 2.00- 6.00) |
| day\_of\_week | Missing (%) | -- |
| season | 3 | 630,708.00 ( 21.10%) |
| season | 2 | 630,708.00 ( 21.10%) |
| season | 4 | 791,575.00 ( 26.50%) |
| season | 1 | 863,878.00 ( 28.90%) |
| store\_nbr | Mean (SD) | 27.10 ( 16.20) |
| store\_nbr | Median (IQR) | 28.00 ( 12.00- 41.00) |
| store\_nbr | Missing (%) | -- |
| item\_nbr | Mean (SD) | 840,738.40 (420,141.00) |
| item\_nbr | Median (IQR) | 903,283.00 (457,928.00-1,157,329.00) |
| item\_nbr | Missing (%) | -- |
| Toal\_trans\_store\_day | Mean (SD) | 1,831.70 (1,041.40) |
| Toal\_trans\_store\_day | Median (IQR) | 1,504.00 (1,122.00-2,323.00) |
| Toal\_trans\_store\_day | Missing (%) | -- |
| city | Quito | 1,156,763.00 ( 38.70%) |
| city | Guayaquil | 402,428.00 ( 13.50%) |
| city | Ambato | 133,796.00 ( 4.50%) |
| city | Cuenca | 128,937.00 ( 4.30%) |
| city | Santo Domingo | 123,404.00 ( 4.10%) |
| city | Machala | 122,992.00 ( 4.10%) |
| city | Other | 811,987.00 ( 27.20%) |
| city\_population | Mean (SD) | 868,121.50 (727,340.10) |
| city\_population | Median (IQR) | 1,399,814.00 (128,190.00-1,399,814.00) |
| city\_population | Missing (%) | -- |
| state | Pichincha | 1,219,328.00 ( 40.80%) |
| state | Guayas | 581,147.00 ( 19.40%) |
| state | Tungurahua | 133,796.00 ( 4.50%) |
| state | Azuay | 128,937.00 ( 4.30%) |
| state | Santo Domingo de los Tsachilas | 123,404.00 ( 4.10%) |
| state | El Oro | 122,992.00 ( 4.10%) |
| state | Other | 560,826.00 ( 18.80%) |
| type | D | 562,801.00 ( 18.80%) |
| type | C | 378,964.00 ( 12.70%) |
| type | A | 756,465.00 ( 25.30%) |
| type | B | 1,108,984.00 ( 37.10%) |
| type | E | 180,918.00 ( 6.10%) |
| cluster | Mean (SD) | 8.80 ( 4.70) |
| cluster | Median (IQR) | 9.00 ( 4.00- 13.00) |
| cluster | Missing (%) | -- |
| family | GROCERY I | 1,228,621.00 ( 41.10%) |
| family | BEVERAGES | 785,057.00 ( 26.30%) |
| family | DAIRY | 171,920.00 ( 5.80%) |
| family | POULTRY | 148,835.00 ( 5.00%) |
| family | PRODUCE | 146,419.00 ( 4.90%) |
| family | BREAD/BAKERY | 111,074.00 ( 3.70%) |
| family | Other | 285,474.00 ( 9.60%) |
| class | Mean (SD) | 1,540.60 ( 734.10) |
| class | Median (IQR) | 1,122.00 (1,040.00-2,116.00) |
| class | Missing (%) | -- |
| LocalHoliday | 0 | 2,973,851.00 ( 99.50%) |
| LocalHoliday | 1 | 14,281.00 ( 0.50%) |
| RegionalHoliday | 0 | 2,987,153.00 ( 100.00%) |
| RegionalHoliday | 1 | 979.00 ( 0.00%) |
| NationalHoliday | 0 | 2,680,054.00 ( 89.70%) |
| NationalHoliday | 1 | 308,078.00 ( 10.30%) |
| weight\_unit\_sale | Mean (SD) | 27.60 ( 44.80) |
| weight\_unit\_sale | Median (IQR) | 15.00 ( 7.00- 32.00) |
| weight\_unit\_sale | Missing (%) | -- |
| onpromotion | 0 | 2,891,283.00 ( 96.80%) |
| perishable | 0 | 2,212,485.00 ( 74.00%) |
| cat\_type\_A | 0 | 2,425,331.00 ( 81.20%) |
| cat\_type\_B | 0 | 2,609,168.00 ( 87.30%) |
| cat\_type\_C | 0 | 2,231,667.00 ( 74.70%) |
| cat\_type\_D | 0 | 1,879,148.00 ( 62.90%) |
| cat\_type\_E | 0 | 2,807,214.00 ( 93.90%) |
| Family\_cat\_1 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_2 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_3 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_4 | 0 | 2,203,075.00 ( 73.70%) |
| Family\_cat\_4 | 1 | 785,057.00 ( 26.30%) |
| Family\_cat\_5 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_6 | 1 | 111,074.00 ( 3.70%) |
| Family\_cat\_7 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_8 | 1 | 110,732.00 ( 3.70%) |
| Family\_cat\_9 | 1 | 171,920.00 ( 5.80%) |
| Family\_cat\_10 | 1 | 63,670.00 ( 2.10%) |
| Family\_cat\_11 | 1 | 39,409.00 ( 1.30%) |
| Family\_cat\_12 | 1 | 13,508.00 ( 0.50%) |
| Family\_cat\_13 | 1 | 1,228,621.00 ( 41.10%) |
| Family\_cat\_14 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_15 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_17 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_18 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_19 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_20 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_21 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_22 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_23 | 1 | 25,370.00 ( 0.80%) |
| Family\_cat\_24 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_25 | 1 | 51,801.00 ( 1.70%) |
| Family\_cat\_26 | 1 | 49,197.00 ( 1.60%) |
| Family\_cat\_27 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_28 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_29 | 1 | 148,835.00 ( 5.00%) |
| Family\_cat\_30 | 1 | 42,519.00 ( 1.40%) |
| Family\_cat\_31 | 1 | 146,419.00 ( 4.90%) |
| Family\_cat\_32 | 0 | 2,988,132.00 ( 100.00%) |
| Family\_cat\_33 | 0 | 2,988,132.00 ( 100.00%) |
| cat\_cluster\_1 | 1 | 180,975.00 ( 6.10%) |
| cat\_cluster\_2 | 1 | 66,350.00 ( 2.20%) |
| cat\_cluster\_3 | 1 | 372,478.00 ( 12.50%) |
| cat\_cluster\_4 | 1 | 199,166.00 ( 6.70%) |
| cat\_cluster\_5 | 1 | 71,301.00 ( 2.40%) |
| cat\_cluster\_6 | 1 | 257,049.00 ( 8.60%) |
| cat\_cluster\_7 | 1 | 48,136.00 ( 1.60%) |
| cat\_cluster\_8 | 1 | 208,343.00 ( 7.00%) |
| cat\_cluster\_9 | 1 | 132,626.00 ( 4.40%) |
| cat\_cluster\_10 | 1 | 300,140.00 ( 10.00%) |
| cat\_cluster\_11 | 1 | 142,315.00 ( 4.80%) |
| cat\_cluster\_12 | 1 | 57,763.00 ( 1.90%) |
| cat\_cluster\_13 | 1 | 262,629.00 ( 8.80%) |
| cat\_cluster\_14 | 1 | 281,634.00 ( 9.40%) |
| cat\_cluster\_15 | 1 | 278,088.00 ( 9.30%) |
| cat\_cluster\_16 | 1 | 61,588.00 ( 2.10%) |
| cat\_cluster\_17 | 1 | 67,551.00 ( 2.30%) |
| is\_pay\_date\_m2 | 1 | 213,870.00 ( 7.20%) |
| is\_pay\_date\_m1 | 1 | 193,704.00 ( 6.50%) |
| is\_pay\_date | 1 | 195,991.00 ( 6.60%) |
| is\_pay\_date\_p1 | 1 | 193,092.00 ( 6.50%) |
| is\_pay\_date\_p2 | 1 | 185,298.00 ( 6.20%) |
| perc\_city\_pop | Mean (SD) | 0.20 ( 0.10) |
| perc\_city\_pop | Median (IQR) | 0.30 ( 0.00- 0.30) |
| perc\_city\_pop | Missing (%) | -- |
| dcoilwtico | Mean (SD) | 80.60 ( 22.10) |
| dcoilwtico | Median (IQR) | 88.90 ( 54.60- 101.50) |
| dcoilwtico | Missing (%) | -- |
| dcoilwtico\_15 | Mean (SD) | 83.10 ( 21.20) |
| dcoilwtico\_15 | Median (IQR) | 92.40 ( 63.10- 101.60) |
| dcoilwtico\_15 | Missing (%) | -- |
| dcoilwtico\_30 | Mean (SD) | 85.50 ( 20.20) |
| dcoilwtico\_30 | Median (IQR) | 93.60 ( 74.10- 101.90) |
| dcoilwtico\_30 | Missing (%) | -- |
| ratio\_curr\_15 | Mean (SD) | 1.00 ( 0.10) |
| ratio\_curr\_15 | Median (IQR) | 1.00 ( 0.90- 1.00) |
| ratio\_curr\_15 | Missing (%) | -- |
| ratio\_curr\_30 | Mean (SD) | 0.90 ( 0.10) |
| ratio\_curr\_30 | Median (IQR) | 1.00 ( 0.90- 1.00) |
| ratio\_curr\_30 | Missing (%) | -- |