Data Analytics Project: Instacart

Japneet S. Kohli

George Mason University

AIT 580 001

Data Analytics Project: Instacart

The data set selected for this project is an anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. This data was released directly by Instacart through a public release in 2017 (Stanley, 2017). The domain of this data is business as it is related to eCommerce.

The data was collected by Instacart, a tech eCommerce startup company that was established in 2012. It is a service company that provides online grocery delivery to customers ("AWS Case Study: Instacart", n.d.). The data was collected as part of operations at Instacart since it is a listing of orders by certain customers. It is a subset of the orders that were fulfilled by Instacart in 2017. Since the identifier columns in the dataset have been anonymized, there are no relevant ethical or privacy concerns associated. Also, this is a biased dataset and not representative of the actual data related to orders at Instacart. This further alleviates any concerns with respect to privacy that may have emerged as a result of making inferences from the dataset itself (Stanley, 2017).

The data set is a listing of the details associated with customer orders, the products ordered, the aisles and departments related to the products, as well as the details of interrelatedness between the orders and the products. The data is organized in comma separated value (CSV) files, with the following categories and columns ("The Instacart Online Grocery Shopping Dataset 2017", 2017):

`orders` (3.4m rows, 206k users):

* `order_id`: order identifier

* `user_id`: customer identifier

* `eval_set`: which evaluation set this order belongs in (see `SET` described below)

* `order_number`: the order sequence number for this user (1 = first, n = nth)

* `order_dow`: the day of the week the order was placed on

* `order_hour_of_day`: the hour of the day the order was placed on

* `days_since_prior`: days since the last order, capped at 30 (with NAs for `order_number` = 1)

`products` (50k rows):

* `product_id`: product identifier

* `product_name`: name of the product

* `aisle_id`: foreign key

* `department_id`: foreign key

`aisles` (134 rows):

* `aisle_id`: aisle identifier

* `aisle`: the name of the aisle

`deptartments` (21 rows):

* `department_id`: department identifier

* `department`: the name of the department

`order_products__SET` (30m+ rows):

* `order_id`: foreign key

* `product_id`: foreign key

* `add_to_cart_order`: order in which each product was added to cart

* `reordered`: 1 if this product has been ordered by this user in the past, 0 otherwise

where `SET` is one of the four following evaluation sets (`eval_set` in `orders`):

* `"prior"`: orders prior to that users most recent order (~3.2m orders)

* `"train"`: training data supplied to participants (~131k orders)

* `"test"`: test data reserved for machine learning competitions (~75k orders)

The above metadata structure shows that the dataset is a multi-million row large dataset. The actual size of the data files, combined, is 680 megabytes. Therefore, using a personal computer to analyze this amount of data is a feasible option. Apart from the large size of the dataset, there is also a lot of variety in the dataset in terms of the complexity of the data types as there are categorical, ordinal, and ratio/interval data types, all available in this data set.

This dataset can be used to find patterns within the dataset that can lend insights into the behaviors of the customers. For example, the following questions could be asked of the dataset:

- Which products are more likely to be ordered at particular times of the day, such as at night time?
- Which aisles are most likely to be reordered from?
- Which are the top product items to be ordered?
- Which products are likely to be ordered in combination?

Through this project, an attempt will be made to answer the above questions.

Update After Working on the Dataset: *The following questions were actually answered during the dataset analysis:*

- *Which are the top product items to be ordered?*

- *What times of day are the order volumes high?*
- *Which days of the week are the busiest in terms of order volume?*
- *Are there any predictive relationships we can find between the various variables in the dataset?*

For the purpose of this project, we will use the capabilities of the AIT 580 instance of the Big Data Lite VM, which has been authorized to be used for this class. The relational database PostgreSQL, installed in the VM, will be used to house the data as it is already structured similarly. Python and R are the intended software of choice for conducting any analysis on the data. For visualizations, Tableau will be the product of choice.

Update After Working on the Dataset: *The Big Data Lite VM was not used for this project as several issues with PostgreSQL database hampered the progress of the project. Instead, a standalone version of Oracle SQL Developer was installed in the actual machine of the student. This was connected with Tableau to allow visualization capabilities linked with the dataset. In combination of the above two software, R was used for conducting various analyses.*

As part of the public release of this dataset, some relationships within the dataset were explored by the engineers at Instacart. Some such relationships that have been mentioned include answers to the first two questions asked above. While the answers are provided, the analysis to back these answers is not available; therefore, through this project, an attempt will be made to verify the existence of these relationships (Stanley, 2017).

Data Exploration and Analysis

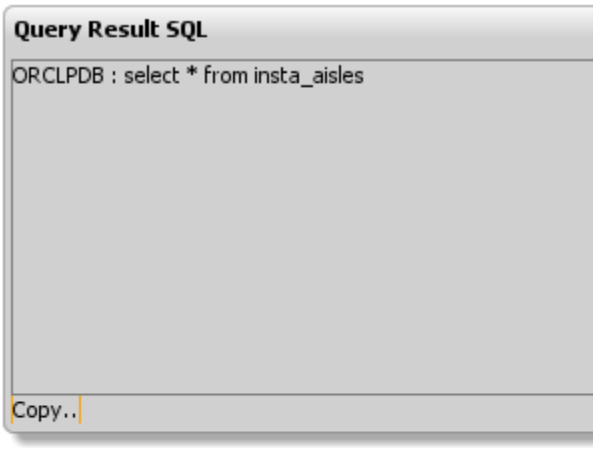
The granular data at the lowest level was found to have more than 32 million records. All of this data was inserted in an Oracle database using Oracle SQL Developer. The tables were created as shown below.

```
-- Create tables to house the project data

create table insta_aisles (aisle_id number,
                           aisle varchar2(4000 byte)
                           );

create table insta_department (department_id number,
                               department varchar2(4000 byte)
                               );

create table insta_department (department_id number,
                               department varchar2(4000 byte)
                               );

create table insta_order_products_prior (order_id number,
                                         product_id number,
                                         add_to_cart_order number,
                                         reordered number
                                         );

create table insta_order_products_train (order_id number,
                                         product_id number,
                                         add_to_cart_order number,
                                         reordered number
                                         );

create table insta_orders (order_id number,
                           user_id number,
                           eval_set varchar2(4000 byte),
                           order_number number,
                           order_dow number,
                           order_hour_of_day number,
                           days_since_prior_order number
                           );

```

```
36  create table insta_products (product_id number,
37                               product_name   varchar2(4000 byte),
38                               aisle_id   number,
39                               department_id number
40                               );
41
42  -- Data inserted in above tables using the "Import" command. Verify results below.
43
44  select * from insta_aisles;
45  select * from insta_department;
46  select * from insta_order_products_prior;
47  select * from insta_order_products_train;
48  select * from insta_orders;
49  select * from insta_products;
```

The SELECT statements from the above queries confirm that the data was loaded correctly. The results of these queries are shown below.

	AISLE_ID	AISLE
1	1	prepared soups salads
2	2	specialty cheeses
3	3	energy granola bars
4	4	instant foods
5	5	marinades meat preparation
6	6	other
7	7	packaged meat
8	8	bakery desserts
9	9	pasta sauce
10	10	kitchen supplies
11	11	cold flu allergy
12	12	fresh pasta
13	13	prepared meals
14	14	tofu meat alternatives
15	15	packaged seafood
16	16	fresh herbs
17	17	baking ingredients
18	18	bulk dried fruits vegetables
19	19	oils vinegars



	DEPARTMENT_ID	DEPARTMENT
1	1	frozen
2	2	other
3	3	bakery
4	4	produce
5	5	alcohol
6	6	international
7	7	beverages
8	8	pets
9	9	dry goods pasta
10	10	bulk
11	11	personal care
12	12	meat seafood
13	13	pantry
14	14	breakfast

Query Result 1 SQL

ORCLPDB : select * from insta_department

Copy..

	ORDER_ID	PRODUCT_ID	ADD_TO_CART_ORDER	REORDERED
1	2	33120	1	1
2	2	28985	2	1
3	2	9327	3	0
4	2	45918	4	1
5	2	30035	5	0
6	2	17794	6	1
7	2	40141	7	1
8	2	1819	8	1
9	2	43668	9	0
10	3	33754	1	1
11	3	24838	2	1
12	3	17704	3	1
13	3	21903	4	1
14	3	17668	5	1
15	3	46667	6	1
16	3	17461	7	1
17	3	32665	8	1
18	4	46842	1	0
19	4	26434	2	1
20	4	39758	3	1
21	4	27761	4	1

Query Result 2 SQL

ORCLPDB : select * from insta_order_products_prior

Copy..

	ORDER_ID	PRODUCT_ID	ADD_TO_CART_ORDER	REORDERED
1	1	49302	1	1
2	1	11109	2	1
3	1	10246	3	0
4	1	49683	4	0
5	1	43633	5	1
6	1	13176	6	0
7	1	47209	7	0
8	1	22035	8	1
9	36	39612	1	0
10	36	19660	2	1
11	36	49235	3	0
12	36	43086	4	1
13	36	46620	5	1
14	36	34497	6	1
15	36	48679	7	1
16	36	46979	8	1
17	38	11913	1	0
18	38	18159	2	0

Query Result 3 SQL

ORCLPDB : select * from insta_order_products_train

Copy..

	ORDER_ID	USER_ID	EVAL_SET	ORDER_NUMBER	ORDER_DOW	ORDER_HOUR_OF_DAY	DAYS_SINCE_PRIOR_ORDER
1	2539329	1	prior	1	2	8	(null)
2	2398795	1	prior	2	3	7	15
3	473747	1	prior	3	3	12	21
4	2254736	1	prior	4	4	7	29
5	431534	1	prior	5	4	15	28
6	3367565	1	prior	6	2	7	19
7	550135	1	prior	7	1	9	20
8	3108588	1	prior	8	1	14	14
9	2295261	1	prior	9	1	16	0
10	2550362	1	prior	10	4	8	30
11	1187899	1	train	11	4	8	14
12	2168274	2	prior	1	2	11	(null)
13	1501582	2	prior	2	5	10	10
14	1901567	2	prior	3	1	10	3
15	738281	2	prior	4	2	10	8
16	1673511	2	prior	5	3	11	8
17	1199898	2	prior	6	2	9	13
18	3194192	2	prior	7	2	12	14
19	788338	2	prior	8	1	15	27

	PRODUCT_ID	PRODUCT_NAME	AISLE_ID	DEPARTMENT_ID
1	1	Chocolate Sandwich Cookies	61	19
2	2	All-Seasons Salt	104	13
3	3	Robust Golden Unsweetened Oolong Tea	94	7
4	4	Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce	38	1
5	5	Green Chile Anytime Sauce	5	13
6	6	Dry Nose Oil	11	11
7	7	Pure Coconut Water With Orange	98	7
8	8	Cut Russet Potatoes Steam N' Mash	116	1
9	9	Light Strawberry Blueberry Yogurt	120	16
10	10	Sparkling Orange Juice & Prickly Pear Beverage	115	7
11	11	Peach Mango Juice	31	7
12	12	Chocolate Fudge Layer Cake	119	1
13	13	Saline Nasal Mist	11	11
14	14	Fresh Scent Dishwasher Cleaner	74	17
15	15	Overnight Diapers Size 6	56	18
16	16	Mint Chocolate Flavored Syrup	103	19

Various queries were written that were saved in views. These views were then used for creating visualizations in Tableau, which was connected with the Oracle database that allowed the views to be accessible in Tableau. Some of these queries and visualizations are shown below.

Number of Orders by Day of Week

The query below shows the total number of orders by day of week. The query is ordered by descending number of orders. It is clear that most of the orders come in on Sundays, while the least are seen on Thursdays.

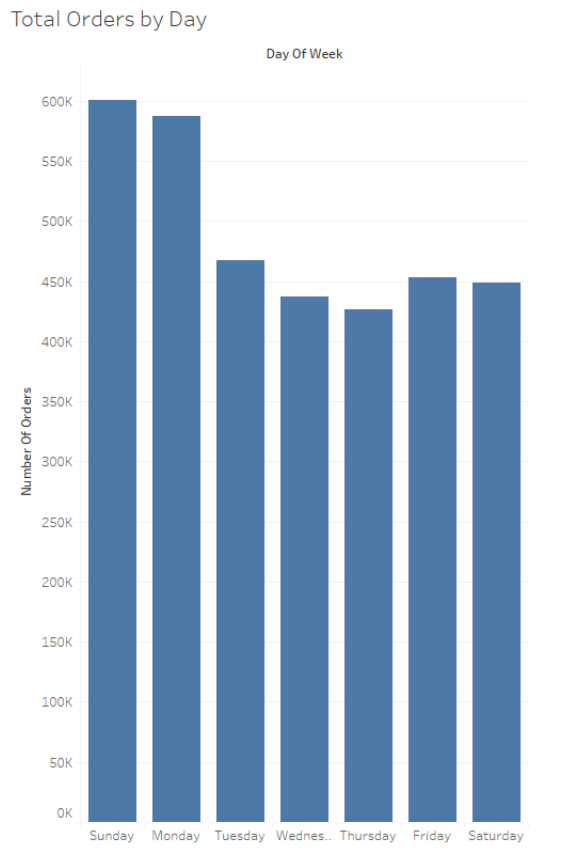
```
53  create or replace view order_total_dow_vw as
54  select
55    count(order_id) as number_of_orders,
56    decode(order_dow,
57           '0', 'Sunday',
58           '1', 'Monday',
59           '2', 'Tuesday',
60           '3', 'Wednesday',
61           '4', 'Thursday',
62           '5', 'Friday',
63           '6', 'Saturday',
64           'Other Day') as day_of_week
65    from insta_orders
66   group by order_dow
67   order by number_of_orders desc;
68
69  select * from order_total_dow_vw;
70
71
72
```

Script Output × Query Result ×

SQL | All Rows Fetched: 7 in 0.207 seconds

	NUMBER_OF_ORDERS	DAY_OF_WEEK
1	600905	Sunday
2	587478	Monday
3	467260	Tuesday
4	453368	Friday
5	448761	Saturday
6	436972	Wednesday
7	426339	Thursday

The above query can be visualized using Tableau, as shown below. The one thing that stands out from this visualization (bar plot) is that the number of orders placed on Sundays and Mondays are significantly higher than the other days in the week. Also notice that the days of the week here are in chronological order, from Sunday to Saturday, which makes reading the data from left to right easier as the next column is the next day.



Orders by Hour of Day

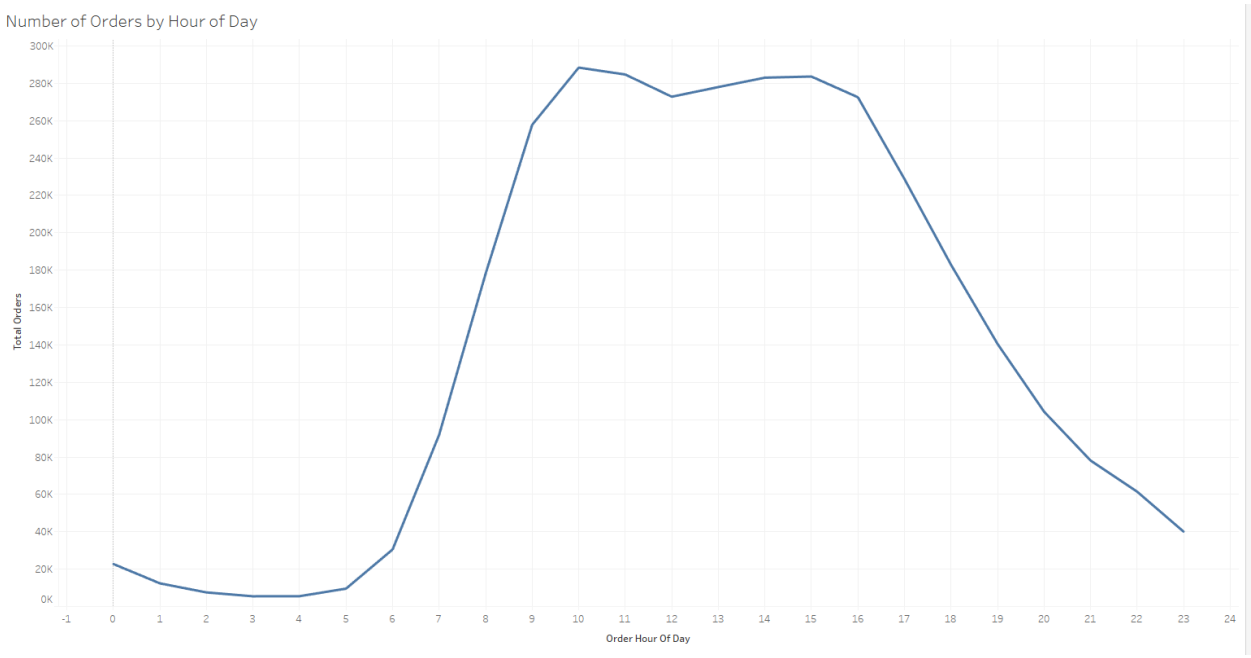
The below query compiles the number of orders by the hour of the day. It is clear that few orders are placed in the night time.

```
73  CREATE OR REPLACE VIEW ORDER_BY_HOUR_VW AS
74  select
75    count(ORDER_ID) as TOTAL_ORDERS,
76    ORDER_HOUR_OF_DAY
77    from INSTA_ORDERS
78   group by ORDER_HOUR_OF_DAY
79   order by ORDER_HOUR_OF_DAY;
80
81  select * from ORDER_BY_HOUR_VW;
82
83
84
```

Script Output | Query Result | SQL | All Rows Fetched: 24 in 0.433 seconds

	TOTAL_ORDERS	ORDER_HOUR_OF_DAY
2	12398	1
3	7539	2
4	5474	3
5	5527	4
6	9569	5
7	30529	6
8	91868	7
9	178201	8
10	257812	9
11	288418	10
12	284728	11
13	272841	12
14	277999	13
15	283042	14
16	283639	15
17	272553	16
18	228795	17
19	182912	18
20	140569	19
21	104292	20
22	78109	21
23	61468	22
24	40043	23

The number of orders is noticed to be consistently high between 9 am and 4 pm. This trend can be observed in the Tableau line chart below, made from the data obtained from this query.



Top Product Names by Number of Orders

The below query finds out which products are the most ordered items based on the number of orders that a product is included in, a higher number of which signifies that a product is among the popular items.

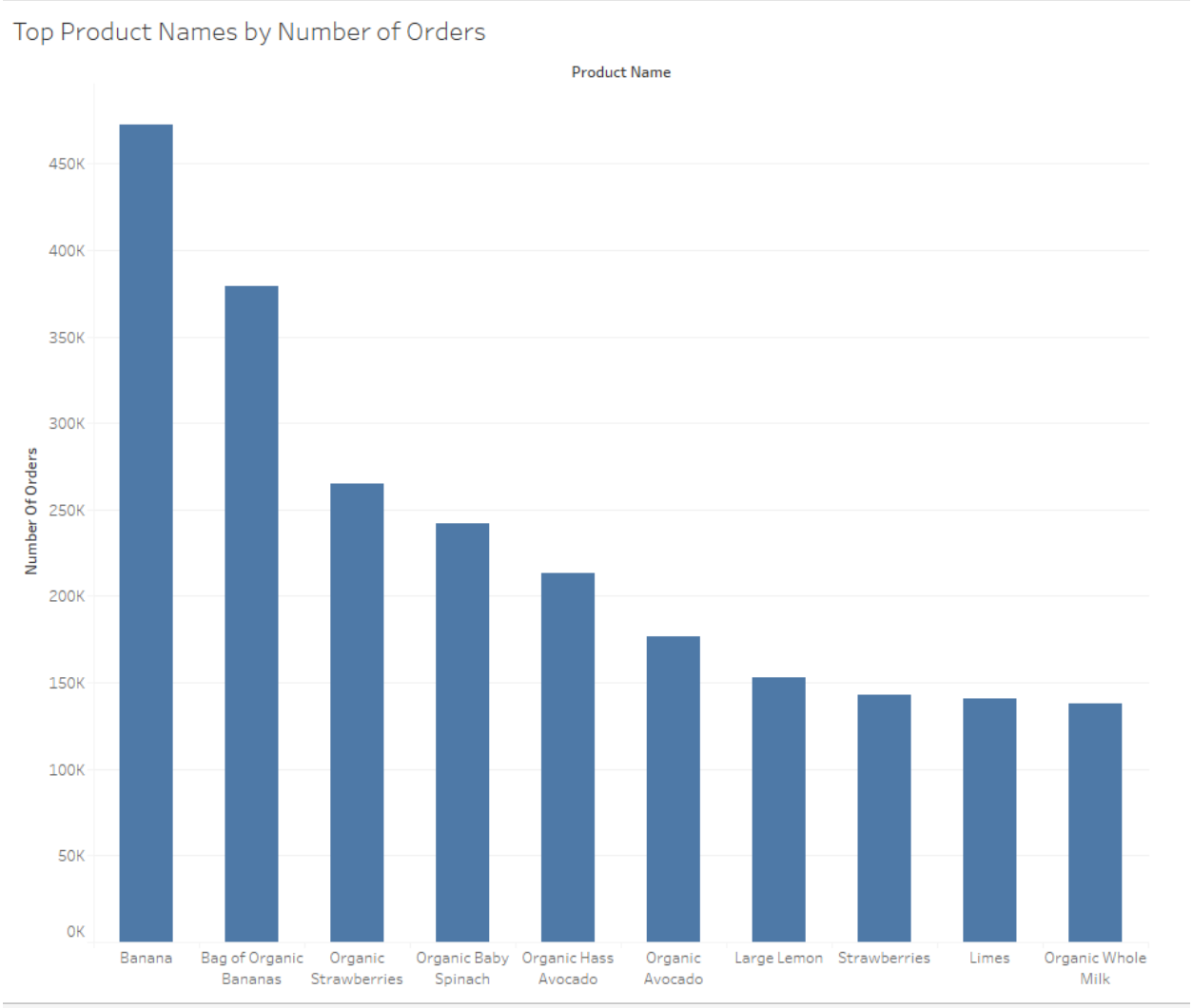
```
35 CREATE OR REPLACE VIEW ORDER_BY_PRODUCT_NAME_VW AS
36
37 select COUNT(PROD_PRIOR.ORDER_ID) as NUMBER_OF_ORDERS,
38        PROD.PRODUCT_ID,
39        PROD.PRODUCT_NAME
40 from INSTA_ORDER_PRODUCTS_PRIOR PROD_PRIOR,
41      INSTA_PRODUCTS PROD
42 where PROD_PRIOR.PRODUCT_ID = PROD.PRODUCT_ID
43  group by PROD.PRODUCT_ID, PROD.PRODUCT_NAME
44  order by NUMBER_OF_ORDERS desc;
45
46 select * from ORDER_BY_PRODUCT_NAME_VW;
```

Script Output | Query Result

SQL | Fetched 50 rows in 1.788 seconds

	NUMBER_OF_ORDERS	PRODUCT_ID	PRODUCT_NAME
1	472565	24852	Banana
2	379450	13176	Bag of Organic Bananas
3	264683	21137	Organic Strawberries
4	241921	21903	Organic Baby Spinach
5	213584	47209	Organic Hass Avocado
6	176815	47766	Organic Avocado
7	152657	47626	Large Lemon
8	142951	16797	Strawberries
9	140627	26209	Limes
10	137905	27845	Organic Whole Milk
11	137057	27966	Organic Raspberries
12	113426	22935	Organic Yellow Onion
13	109778	24964	Organic Garlic
14	104823	45007	Organic Zucchini
15	100060	39275	Organic Blueberries
16	97315	49683	Cucumber Kirby

The below Tableau bar chart shows the top 10 products as ranked by number of orders per product. As you can see, bananas, organic and non-organic both, far outnumber any other product ordered by users. The top 10 list is comprised of fresh fruit and dairy items only.



Exporting Data for Analysis in R

The following query was used to create a dataset with various details regarding the Instacart Orders. The results from this query were exported into a CSV file, which were then read into a dataframe in R for further analysis.

```
create or replace view ORDER_DETAILS_VW as

SELECT  PRIOR_ORDERS.ORDER_ID,
        PRIOR_ORDERS.PRODUCT_ID,
        PRIOR_ORDERS.ADD_TO_CART_ORDER,
        PRIOR_ORDERS.REORDERED,
        ORDERS.USER_ID,
        ORDERS.ORDER_NUMBER,
        ORDERS.ORDER_DOW,
        ORDERS.ORDER_HOUR_OF_DAY,
        ORDERS.DAYS_SINCE_PRIOR_ORDER

FROM INSTA_ORDER_PRODUCTS_PRIOR PRIOR_ORDERS,
     INSTA_ORDERS ORDERS

WHERE PRIOR_ORDERS.ORDER_ID = ORDERS.ORDER_ID;

select * from order_details_vw;
```

Script Output | Query Result | SQL | Fetched 50 rows in 2.723 seconds

	ORDER_ID	PRODUCT_ID	ADD_TO_CART_ORDER	REORDERED	USER_ID	ORDER_NUMBER	ORDER_DOW	ORDER_HOUR_OF_DAY	DAYS_SINCE_PRIOR_ORDER
1	2	33120	1	1	202279	3	5	9	8
2	2	28985	2	1	202279	3	5	9	8
3	2	9327	3	0	202279	3	5	9	8
4	2	45918	4	1	202279	3	5	9	8
5	2	30035	5	0	202279	3	5	9	8
6	2	17794	6	1	202279	3	5	9	8
7	2	40141	7	1	202279	3	5	9	8
8	2	1819	8	1	202279	3	5	9	8
9	2	43668	9	0	202279	3	5	9	8
10	3	33754	1	1	205970	16	5	17	12
11	3	24838	2	1	205970	16	5	17	12
12	3	17704	3	1	205970	16	5	17	12
13	3	21903	4	1	205970	16	5	17	12
14	3	17668	5	1	205970	16	5	17	12
15	3	46667	6	1	205970	16	5	17	12
16	3	17461	7	1	205970	16	5	17	12

The above data was read in an R data frame as shown below.

```
> # set working directory to location where exported file is saved
> setwd("C:/Users/Japneet/Documents/GMU Coursework/Fall 2019/AIT 580 Analytics Big Data to Information/Data Analysis Project")
>
> order_details <- read.csv("export.csv")
>
> # get structure of data
> str(order_details)
'data.frame':   32434489 obs. of  9 variables:
 $ ORDER_ID             : int  13 13 13 13 13 13 13 13 13 13 ...
 $ PRODUCT_ID           : int  17330 27407 35419 196 44635 26878 25783 41290 33198 23020 ...
 $ ADD_TO_CART_ORDER    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ REORDERED            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ USER_ID              : int  45082 45082 45082 45082 45082 45082 45082 45082 45082 45082 ...
 $ ORDER_NUMBER         : int  2 2 2 2 2 2 2 2 2 2 ...
 $ ORDER_DOW            : int  6 6 6 6 6 6 6 6 6 6 ...
 $ ORDER_HOUR_OF_DAY    : int  17 17 17 17 17 17 17 17 17 17 ...
 $ DAYS_SINCE_PRIOR_ORDER: int  1 1 1 1 1 1 1 1 1 1 ...
>
> summary(order_details)
    ORDER_ID         PRODUCT_ID    ADD_TO_CART_ORDER   REORDERED         USER_ID        ORDER_NUMBER     ORDER_DOW     ORDER_HOUR_OF_DAY DAYS_SINCE_PRIOR_ORDER
 Min.   :      2   Min.   :    1   Min.   :  1.000   Min.   :0.0000   Min.   :     1   Min.   : 1.00   Min.   :0.000   Min.   : 0.00     Min.   : 0.0
 1st Qu.: 855943   1st Qu.:13530   1st Qu.:  3.000   1st Qu.:0.0000   1st Qu.: 51421   1st Qu.: 5.00   1st Qu.:1.000   1st Qu.:10.00     1st Qu.: 5.0
 Median :1711048   Median :25256   Median :  6.000   Median :1.0000   Median :102611   Median :11.00   Median :3.000   Median :13.00     Median : 8.0
 Mean   :1710749   Mean   :25576   Mean   :  8.351   Mean   :0.5897   Mean   :102937   Mean   :17.14   Mean   :2.739   Mean   :13.42     Mean   :11.1
 3rd Qu.:2565514   3rd Qu.:37935   3rd Qu.: 11.000   3rd Qu.:1.0000   3rd Qu.:154391   3rd Qu.:24.00   3rd Qu.:5.000   3rd Qu.:16.00     3rd Qu.:15.0
 Max.   :3421083   Max.   :49688   Max.   :145.000   Max.   :1.0000   Max.   :206209   Max.   :99.00   Max.   :6.000   Max.   :23.00     Max.   :30.0
                                                                                                                                        NA's   :2078068
>
```

Using the above data, the following analyses were performed.

Correlation Analysis

The correlation between the various columns can be seen in the below matrix. It is evident from this chart that the best

```
> # select only numeric columns for finding correlation
> order_details_numeric <- order_details[,sapply(order_details,is.numeric)]
> # find pairwise correlation between all numeric columns
> cor(order_details_numeric, use = "complete.obs")
                         ORDER_ID    PRODUCT_ID ADD_TO_CART_ORDER     REORDERED       USER_ID  ORDER_NUMBER    ORDER_DOW ORDER_HOUR_OF_DAY DAYS_SINCE_PRIOR_ORDER
ORDER_ID              1.000000e+00  1.889816e-05     -0.0005055264 -0.0002600507 -0.0002804022 -0.0005673946  0.001262461      0.0005115518           0.0007265988
PRODUCT_ID            1.889816e-05  1.000000e+00      0.0057945413  0.0042430389  0.0001243542 -0.0019036062 -0.002254161      0.0009758589           0.0007999135
ADD_TO_CART_ORDER    -5.055264e-04  5.794541e-03      1.0000000000 -0.1452324752  0.0009411534 -0.0049212613 -0.008969544     -0.0149719694           0.0539514873
REORDERED            -2.600507e-04  4.243039e-03     -0.1452324752  1.0000000000 -0.0008661519  0.2509734040 -0.008800684     -0.0211415028          -0.1328139218
USER_ID              -2.804022e-04  1.243542e-04      0.0009411534 -0.0008661519  1.0000000000 -0.0007632385 -0.001935985     -0.0008650765           0.0005563963
ORDER_NUMBER         -5.673946e-04 -1.903606e-03     -0.0049212613  0.2509734040 -0.0007632385  1.0000000000  0.015291453     -0.0394779706          -0.3584215701
ORDER_DOW             1.262461e-03 -2.254161e-03     -0.0089695435 -0.0088006844 -0.0019359850  0.0152914526  1.000000000      0.0127080563          -0.0300024745
ORDER_HOUR_OF_DAY     5.115518e-04  9.758589e-04     -0.0149719694 -0.0211415028 -0.0008650765 -0.0394779706  0.012708056      1.0000000000           0.0038782700
DAYS_SINCE_PRIOR_ORDER 7.265988e-04  7.999135e-04      0.0539514873 -0.1328139218  0.0005563963 -0.3584215701 -0.030002475      0.0038782700           1.0000000000
> |
```

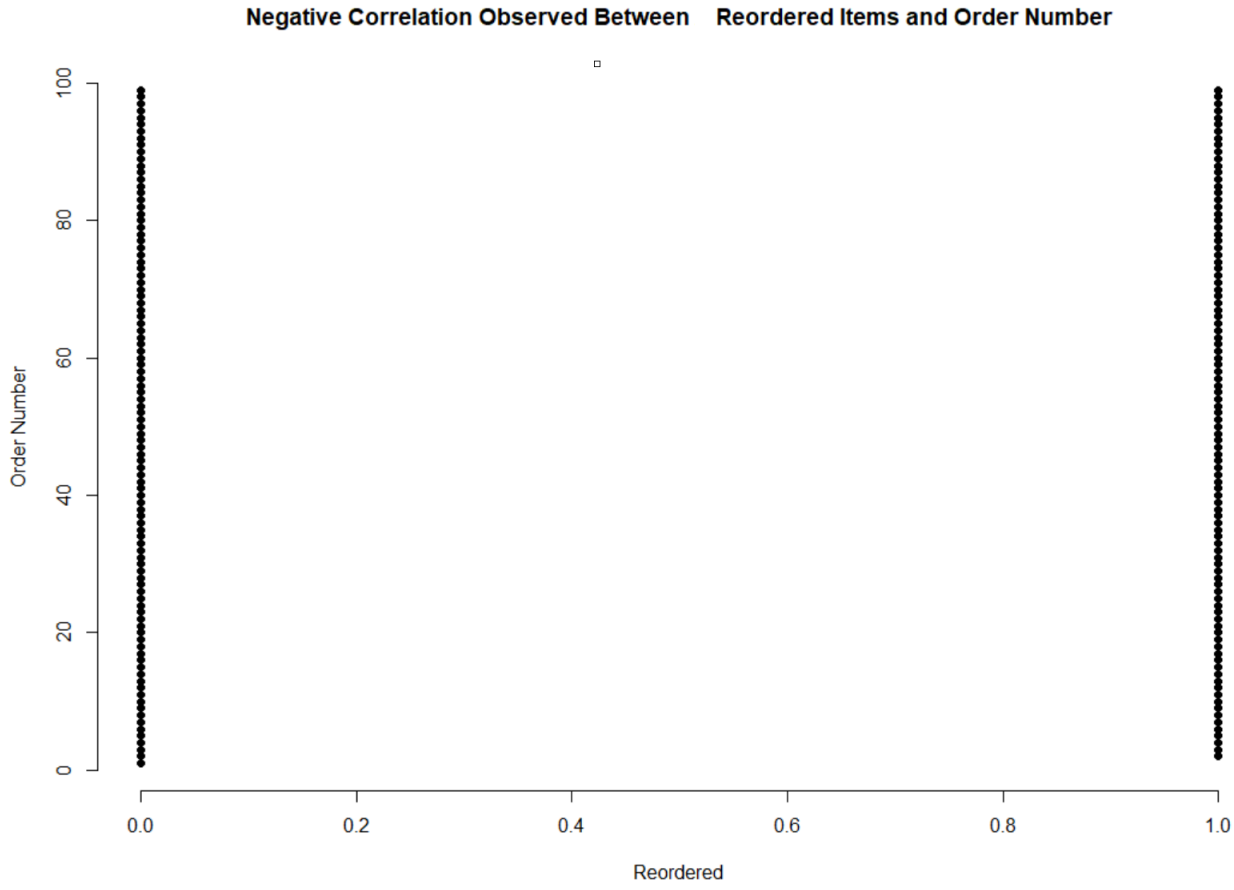
From the above chart, it is clear that most variables have negligible, if any, correlation between them. The strongest relationship that we can find is between DAYS_SINCE_PRIOR_ORDER and ORDER_NUMBER. There exists a negative correlation with a correlation coefficient of roughly -0.36 between the variables. It signifies that as the number of days between placing orders increases for a user, it becomes highly unlikely that they will have multiple orders in a given day as the order number for a new order after a gap of few days would be “1” since it would be their first order of the day. Logically, this is not a surprising outcome.

The other correlation that stands out is between the variables REORDERED and ORDER_NUMBER. The correlation coefficient of 0.25 signifies that whether an item is reordered or not is slightly positively correlated with the order number for a particular user in a day. This could be explained by a theorizing that a user that places multiple orders in a day, thereby increasing the ORDER_NUMBER value, would be more likely to reorder an item.

Scatter Plot Analysis

A scatter plot constructed between REORDERED and ORDER_NUMBER looks like the below graphic, as constructed in R.

```
x <- order_details_numeric$REORDERED
y <- order_details_numeric$ORDER_NUMBER
plot(x, y, main = "Negative Correlation Observed Between Reordered Items and Order Number",
     xlab = "Reordered", ylab = "Order Number",
     pch = 19, frame = FALSE)
```

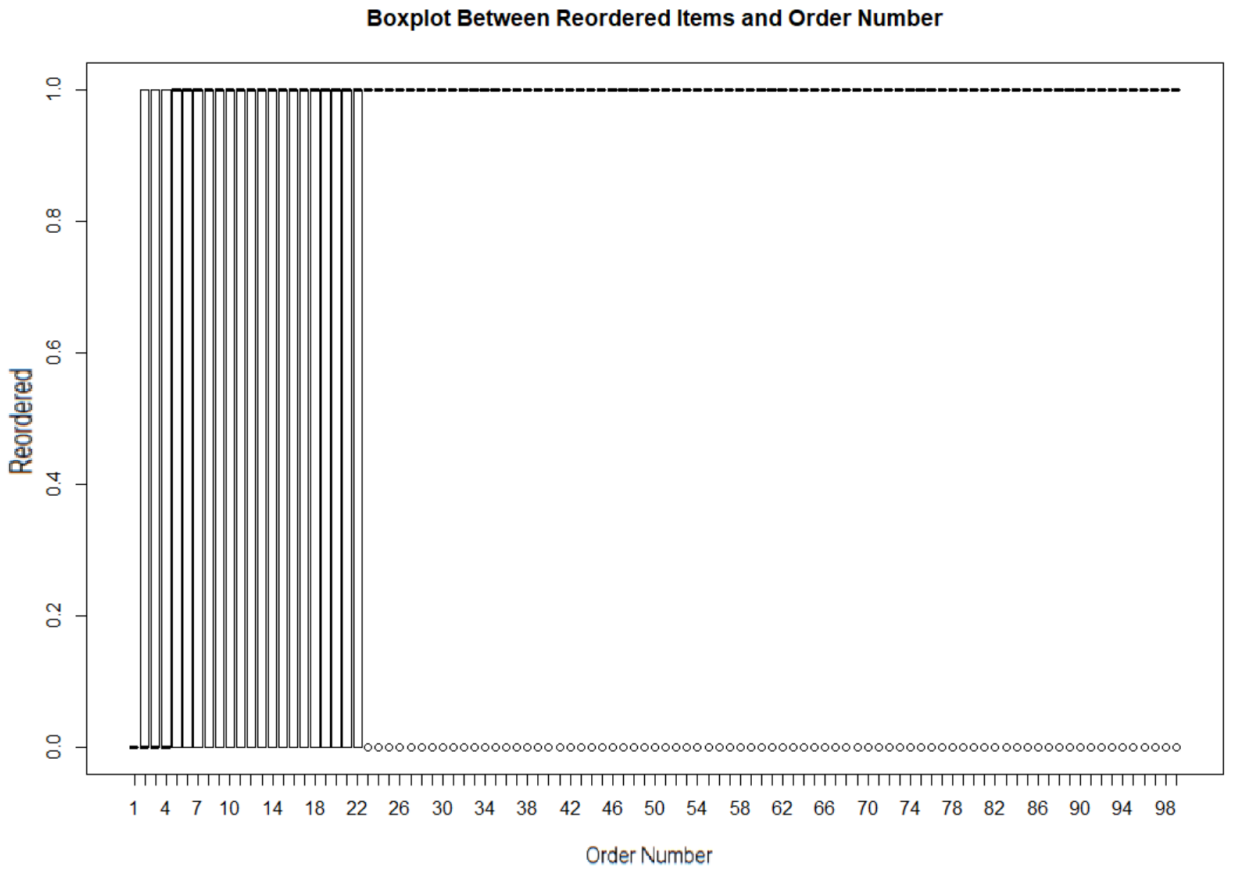


As you can see, it is difficult to see any sort of relationship between the two variables because: 1.) there are only two values for REORDERED (0 for “No” and 1 for “Yes”); and 2.) the ORDER_NUMBER values seem to be spread equally well through the two values for REORDERED. The fact that we have more than 32 million records for this data does hamper our visual ability to see any kind of relationship between the variables through a scatter plot chart as shown above.

Box Plot Analysis

A box plot construction between REORDERED and ORDER_NUMBER is presented below.

```
51  boxplot(x~y, data=order_details_numeric,
52          xlab = "Order Number", ylab = "Reordered", main = "Boxplot Between Reordered Items and Order Number")
```



As can be observed visually, the data is concentrated between the range of ORDER_NUMBER between 1 and 22. The small black tick marks at the 0 and 1 level of y-axis indicate the concentration of the data, i.e., the median values. As can be seen, for Order Numbers 1, 2, 3, and 4, the median Reordered value is close to 0, which means that an item was not reordered for these lower order numbers. However, for all Order Number values greater than 4, the box plot chart shows that the median value is close to 1, which means that an item was

reordered with an increasing number of orders per day. This demonstrates the positive correlation between these two variables, as was determined in the correlation matrix plot shown previously.

Linear Regression and Hypothesis Testing

The findings with regards to the correlation observed between Reordered items and a user's Order Number during a day are further corroborated through a linear regression model, set up as shown below. Here, REORDERED is set up as a function of ORDER_NUMBER, which means that whether a product is reordered or not may be determined by the order number, if such a relationship exists.

```
> lregression <- lm(x ~ y, data = order_details_numeric)
> summary(lregression)

Call:
lm(formula = x ~ y, data = order_details_numeric)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2943 -0.4766  0.2135  0.4460  0.5406

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.421e-01  1.150e-04    3846   <2e-16 ***
y           8.607e-03  4.688e-06    1836   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4682 on 32434487 degrees of freedom
Multiple R-squared:  0.09415,   Adjusted R-squared:  0.09415
F-statistic: 3.371e+06 on 1 and 32434487 DF,  p-value: < 2.2e-16

>
```

By calling a summary of the linear regression model, we can observe that the "y", which refers to the Order Number, is an extremely significant variable to determine the Reordered status of a product.

Moreover, the summary statistics for the linear regression model can also help us conduct hypothesis testing. In this case, our null hypothesis would be that the Reordered status of a product is not determined by or correlated to the Order Number. In other words, it means that data that shows the Reordered status and the Ordered Number is a random distribution. In the above scenario, we can reject this null hypothesis using the p-value statistic. If the p-value is lesser than 5% or 0.05, then we have a 95% confidence in stating that the distribution is not random. In the above case, the pa-value is a much smaller number, 2.2e-16, which means that we can say with greater than 95% confidence (in fact, close to 100% confidence since p-value is extremely small) that this is not a random distribution and there exists a positive correlation between Reordered status and Order Number.

Interpretation of the Study

The results from the project can be used to describe various aspects of user behavior. We have observed that the volume of orders placed through Instacart can vary greatly depending on the time of day or the day of the week. These results indicate when the users are most active on the Instacart platform and when they actually seek the services related to grocery delivery. High order volumes were observed on Monday, which tapered a little bit as the week moved on. Such usage pattern indicates that Instacart may be successful in helping working professionals get grocery deliveries, a chore for which they may not find sufficient time outside of their work life and other matters pertaining to personal life. However, having said this, the fact that the highest order volumes were observed for Sunday, traditionally an off day in the work week, indicates that customers are also attracted to the convenience that Instacart brings.

Similarly, the fact that most of the orders were placed during the hours between 9 am and 4 pm indicates that Instacart may bring a lot of value to working professionals, who find themselves ill-positioned to make grocery runs during the middle of the work day.

Another interesting observation made from this study is that the top ranked products all comprised of fresh fruits, with one dairy item (milk) making it to the top 10 list. This list is noticeably void of any entries from frozen or packaged foods, or other food brackets, such as meats. Clearly, the customers see value in ordering fresh food items from Instacart, which indicates that Instacart has been successful in filling a void in the food delivery space that previously was not met through restaurant or other delivery services.

Lastly, rudimentary analysis of some of the variables in the dataset indicates that there may be underlying predictive capabilities within the dataset itself. We were able to find there exists a significant correlation between Reordered products and Order Number. Similarly, there may be other such relationships between various variables from the other tables in the dataset that were not explored in this study. This could be an interesting topic of future study, which could possibly conclude with creating some sort of a model that could predict certain user behaviors or future order volume.

While a lot more comprehensive analyses would need to be conducted to explore the predictive capabilities of this dataset, it should also be kept in mind that this dataset may not be representative of actual user behavior for Instacart customers. This is because the dataset itself was provided by Instacart for public use. It is obviously not a comprehensive dataset, even with limitations such as location or dates, because these values are not provided in the dataset. It is completely possible that the dataset itself may be biased on purpose to prevent giving away any actual user trends that could be used by Instacart's competitors.

References

AWS Case Study: Instacart. (n.d.). Retrieved October 22, 2019, from https://aws.amazon.com/solutions/case-studies/instacart/.

Stanley, J. (2017, May 4). 3 Million Instacart Orders, Open Sourced. Retrieved October 22, 2019, from https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2.

The Instacart Online Grocery Shopping Dataset 2017, Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on October 29, 2019.