

AIT 580 BIG DATA ANALYTICS PROJECT

FINAL PROJECT REPORT ON

INSTACART

BY

THARUN KANAMNENI

G01235383



The dataset I have selected is **Instacart** which is about 800mb of data which includes data of aisles, departments, orders and products. Some of the data types are strings and integer values.

ABOUT: Instacart is US Software firm which delivers groceries which are ordered online and values at \$8 billion and operates as same-day grocery delivery and pickup service in the U.S and Canada.[3] [4]

NEED: The data was collected to understand and analyze which products the consumers will purchase again based on past orders, what are some of the most reordered items and some more interesting findings about the orders and the products.[4]

PRIVACY :All the customer data is anonymized for privacy, and as it is posted by Instacart itself, it doesn't have any privacy issues and ethical issues.[4]

A structured collection of data representing customer orders over time is the dataset and it was posted by the Instacart as open source of 3 million grocery orders from more than 200,000 orders.[3][4] It also includes the data of the week and hour of the day order was placed.

TYPES OF DATATYPES :There are various attributes present in the dataset, they are order hour of the day, order day of the week, eval_set, days since prior order, product_id, product_type, reordered.

order_id, product_id, order_day_of_week, days_since_prior_order :**Numeric**

product_name, department_name :**Nominal**

eval_set :**Categorical**

The dataset of orders has about 3.4 million rows, and order_products has about 30 million rows.

Some of the questions i got to solve are :

- 1) what are the products that are sold the most?
- 2) what are the most reordered items?
- 3) How many orders are placed in the morning around 10am to 12pm?

Some of the software resources needed to study this data are :

- R analytics for data exploration and visualization
- Python analytics for data exploration
- SQL Work Bench

The Hardware I am using :

Processor: 2.6 GHz 6-Core Intel Core i7, 9th Gen

Memory : 16 GB 2400 MHz DDR4

I have previously used ggplot2, SQL, R and Python in my assignments and did various visualizations, so in order to improve more and work with more large data, I take this as an opportunity to learn more and improve my skills.

Visualizations have been done using python, R studio, Tableau.

Visualizations are done for few major attributes where we can draw many insights and which help to answer our driving questions.

The dataset Orders is :

Similarly, I have used various datasets of Instacart like products, aisles, departments, order_products_prior, order_products_train to draw conclusions and analyze as much as possible

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	
2398795	1	prior	2	3	7	15.0
473747	1	prior	3	3	12	21.0
2254736	1	prior	4	4	7	29.0
431534	1	prior	5	4	15	28.0
3367565	1	prior	6	2	7	19.0
550135	1	prior	7	1	9	20.0
3108588	1	prior	8	1	14	14.0
2295281	1	prior	9	1	16	0.0
2550362	1	prior	10	4	8	30.0
1187899	1	train	11	4	8	14.0
2168274	2	prior	1	2	11	
1501582	2	prior	2	5	10	10.0
1901567	2	prior	3	1	10	3.0
738281	2	prior	4	2	10	8.0
1673511	2	prior	5	3	11	8.0
1199898	2	prior	6	2	9	13.0
3194192	2	prior	7	2	12	14.0
788338	2	prior	8	1	15	27.0
1718559	2	prior	9	2	9	8.0
1447487	2	prior	10	1	11	6.0
1402090	2	prior	11	1	10	30.0
3186735	2	prior	12	1	9	28.0
3268552	2	prior	13	4	11	30.0
839880	2	prior	14	3	10	13.0
1492625	2	train	15	1	11	30.0
1374495	3	prior	1	1	14	
444309	3	prior	2	3	19	9.0
3002854	3	prior	3	3	16	21.0
2037211	3	prior	4	2	18	20.0
2710558	3	prior	5	0	17	12.0
1972919	3	prior	6	0	16	7.0
1839752	3	prior	7	0	15	7.0
3225766	3	prior	8	0	17	7.0
3140960	3	prior	9	0	16	7.0

1. To view the Orders, Prior Dataset

```
>>> orders.head()  
...  
   order_id  user_id  ...  order_hour_of_day  days_since_prior_order  
0    2539329      1  ...            8                 NaN  
1    2398795      1  ...            7             15.0  
2    473747       1  ...           12             21.0  
3    2254736      1  ...            7             29.0  
4    431534       1  ...           15             28.0  
[5 rows x 7 columns]
```

```
>>> prior.head()  
   order_id  product_id  add_to_cart_order  reordered  
0          2        33120                  1          1  
1          2        28985                  2          1  
2          2         9327                  3          0  
3          2        45918                  4          1  
4          2        30035                  5          0
```

To Print number of Aisles.

```
>>> print(aisles.shape)
(134, 2)
```

```
>>> mt.head(20)
   order_id  product_id  ...  days_since_prior_order  aisle
0          2        33120  ...                8.0    eggs
1         26        33120  ...                7.0    eggs
2        120        33120  ...               10.0    eggs
3         327        33120  ...                8.0    eggs
4         390        33120  ...                9.0    eggs
5         537        33120  ...                3.0    eggs
6         582        33120  ...               10.0    eggs
7         608        33120  ...               12.0    eggs
8         623        33120  ...                3.0    eggs
9         689        33120  ...                3.0    eggs
10        689        35921  ...                3.0    eggs
11        726        33120  ...                7.0    eggs
12        726        32655  ...                7.0    eggs
13        771        33120  ...                7.0    eggs
14        800        33120  ...                3.0    eggs
15        901        33120  ...                8.0    eggs
16       1005        33120  ...                6.0    eggs
17       1097        33120  ...                8.0    eggs
18       1193        33120  ...                9.0    eggs
19       1300        33120  ...               20.0    eggs

[20 rows x 14 columns]
```

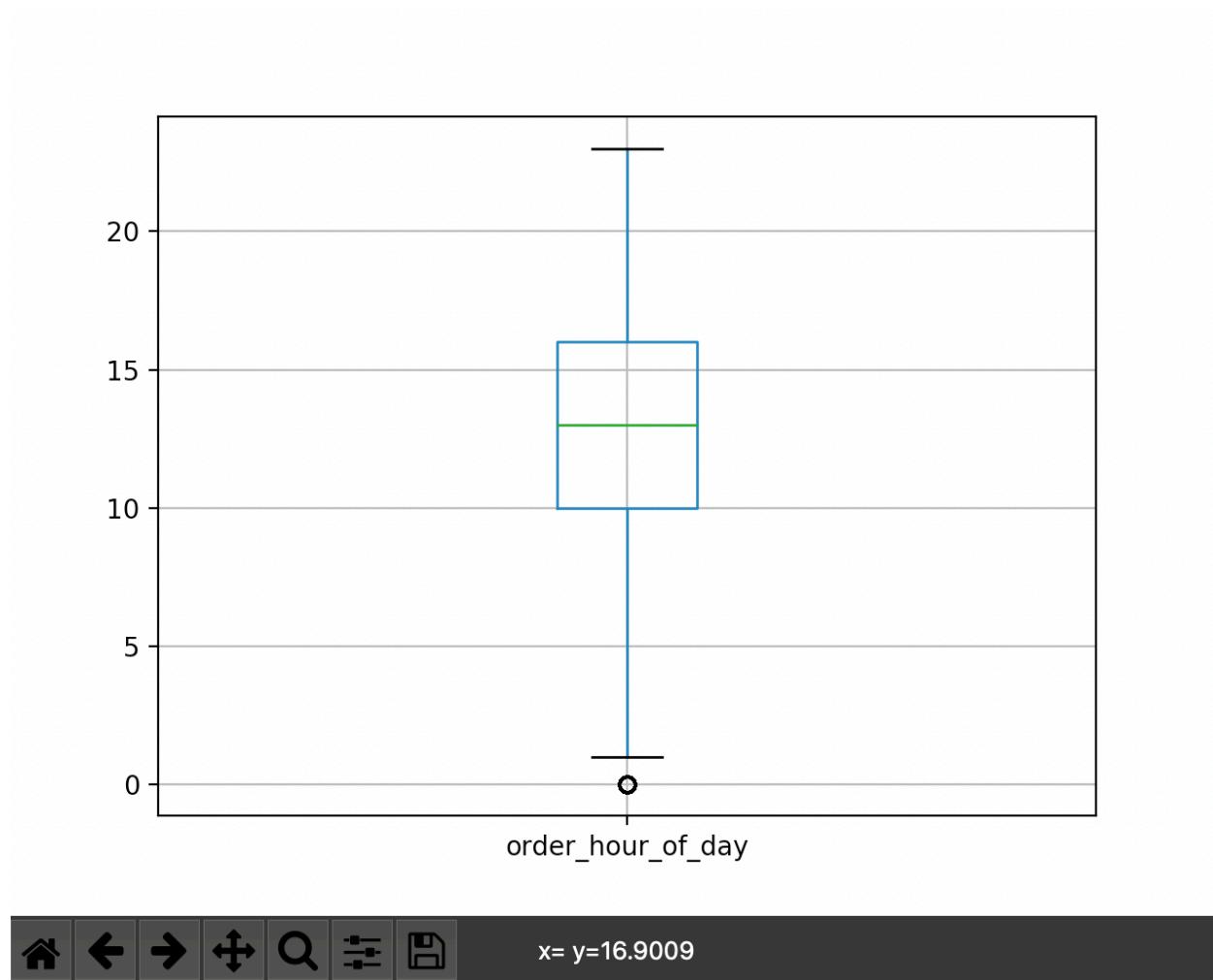
Count of Fresh fruits, fresh vegetables and the best selling goods.

```
>>> mt['aisle'].value_counts()[0:10]
fresh fruits                3642188
fresh vegetables             3418021
packaged vegetables fruits   1765313
yogurt                      1452343
packaged cheese              979763
milk                         891015
water seltzer sparkling water 841533
chips pretzels               722470
soy lactosefree              638253
bread                        584834
Name: aisle, dtype: int64
```

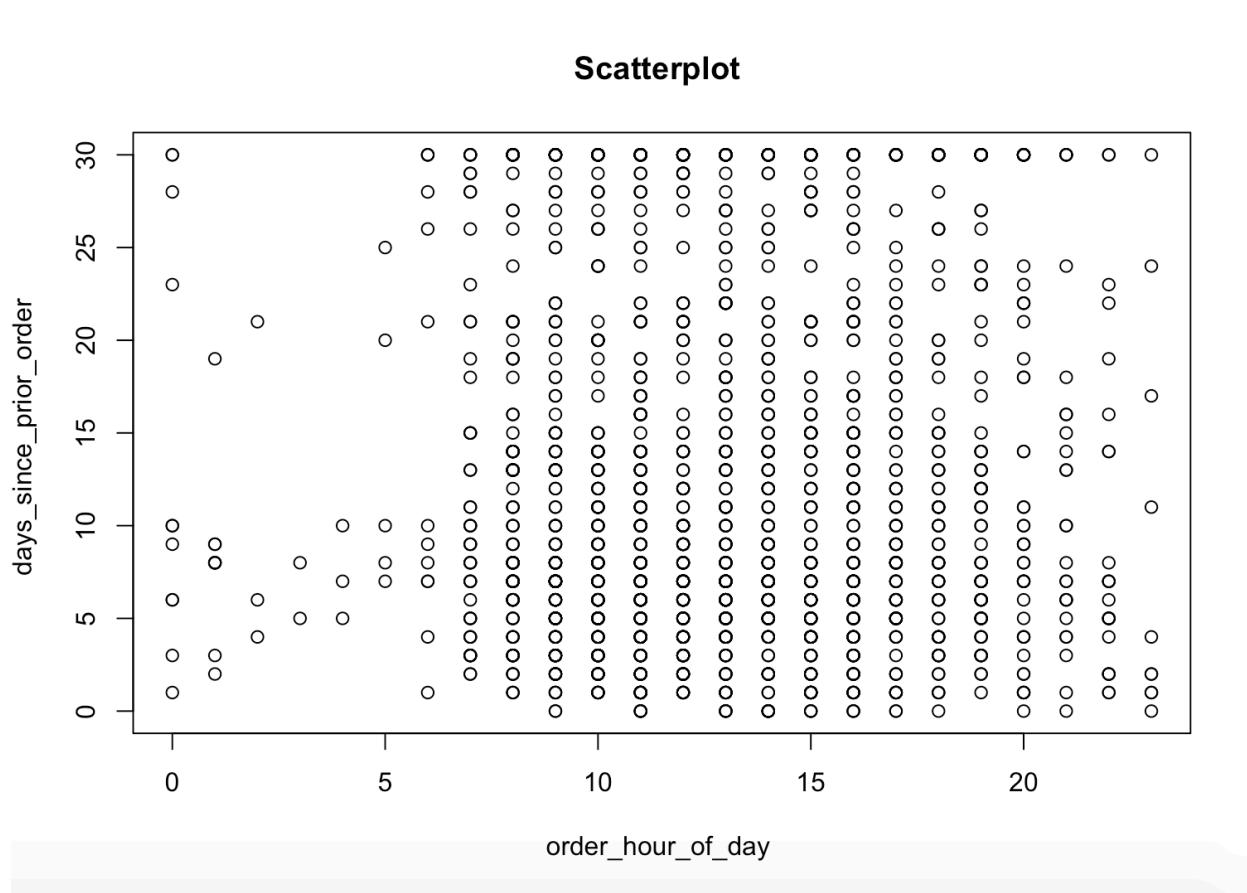
Most number of reorders:

	product_id	Total_reorders	product_name
24848	24852	472565	Banana
13172	13176	379450	Bag of Organic Bananas
21133	21137	264683	Organic Strawberries
21899	21903	241921	Organic Baby Spinach
47198	47209	213584	Organic Hass Avocado
47755	47766	176815	Organic Avocado
47615	47626	152657	Large Lemon
16793	16797	142951	Strawberries
26204	26209	140627	Limes
27839	27845	137905	Organic Whole Milk

1. Total Number of Orders in 24 Hour scale



The above visualization is done in Pycharm using Python to understand that the maximum orders that were done are between 10:00 and 16:00 .

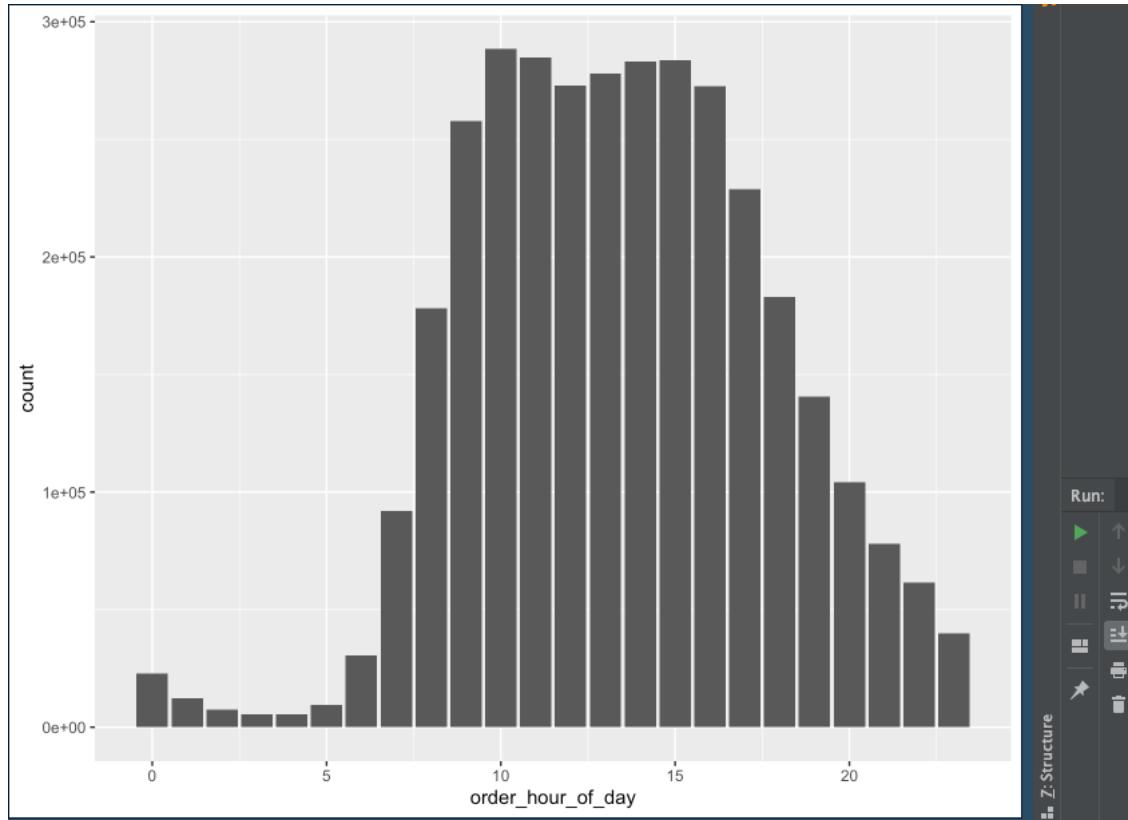
2. This Visualization is a scatter plot made using R studio.

I used R to visualize this,

As I don't have many attributes which relates to a good scatter plot, I don't think this helps me to get information for me as there are no attributes where I can draw a scatter plot to get insights.

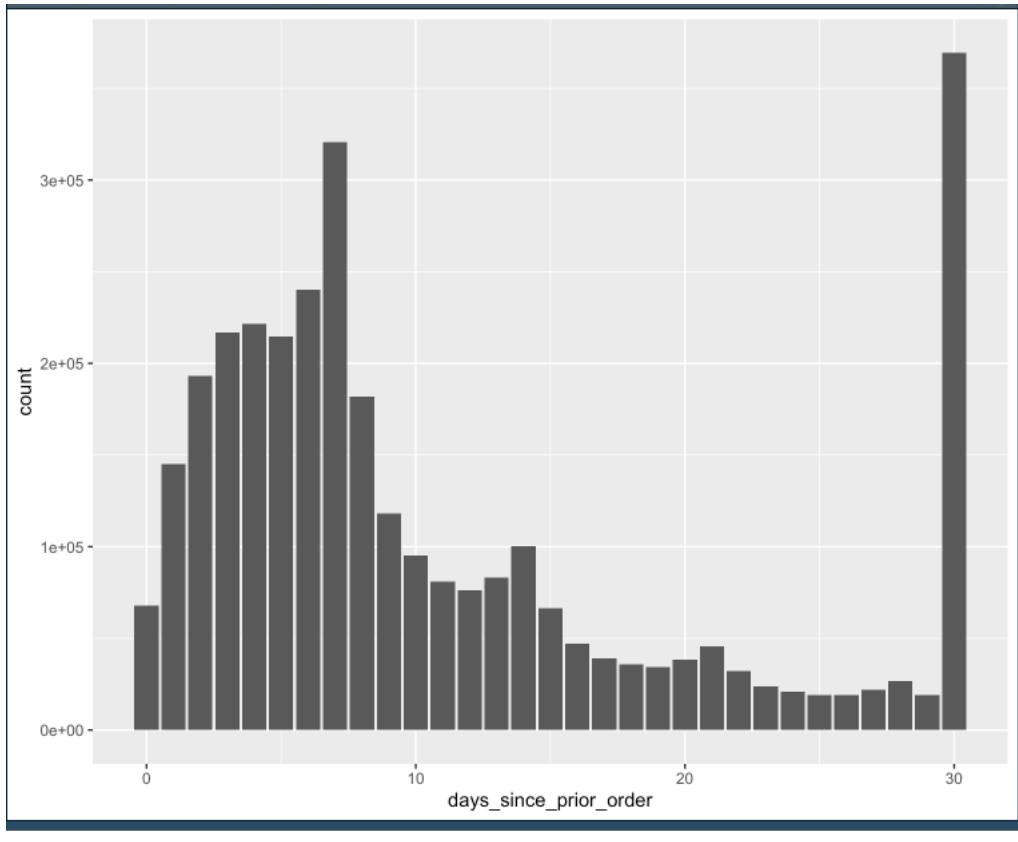
3. Total number of orders in according with the order hour of the day.

The below visualization helps me in finding out the number of items ordered in 24 hour scale where the highest number of items that were sold were between 9:00 to 15:00.



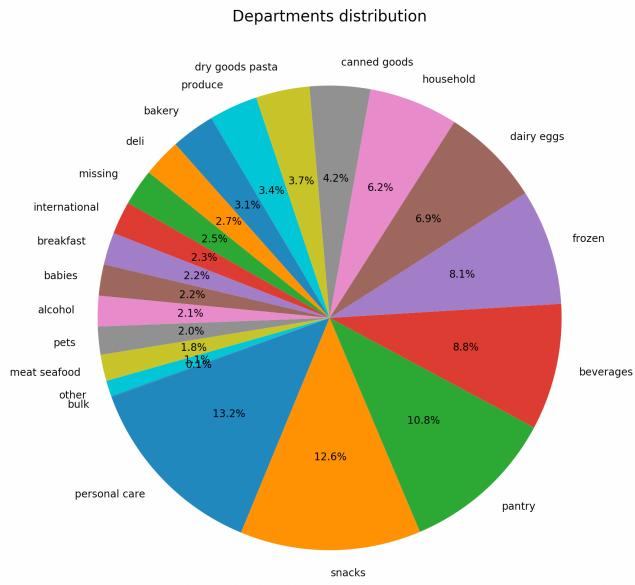
4. Total Number of orders done with respect to days since prior order

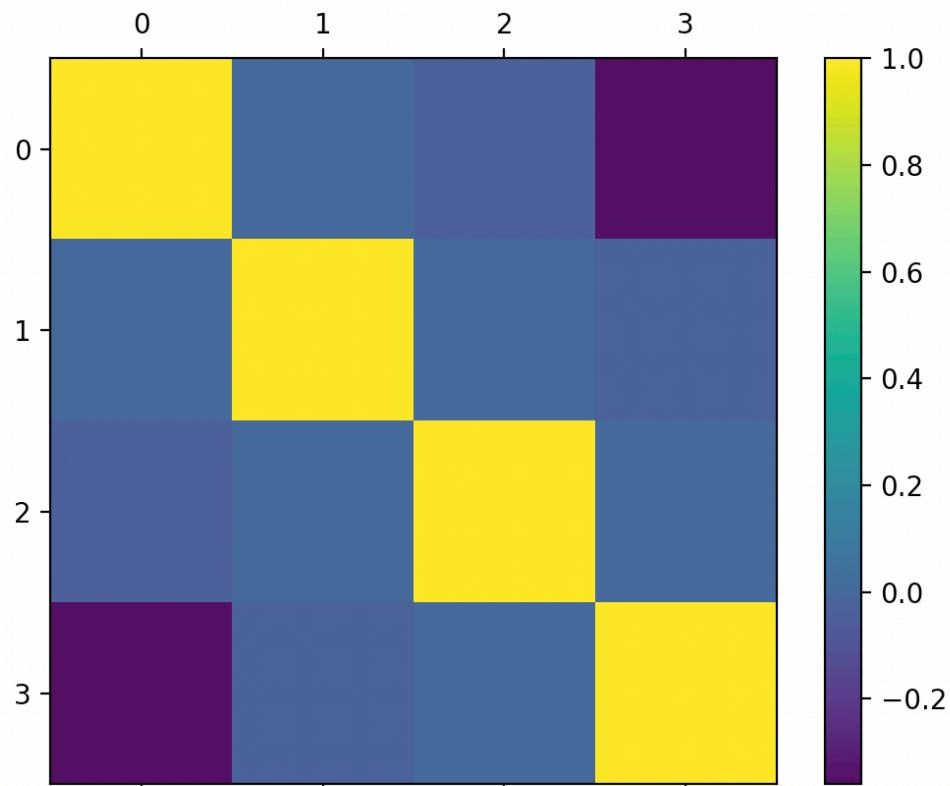
The below visualization describes the customers who order again since their prior order. The highest number of customers who order after 30 days are more.



5. Percentage of items sold in each department

The below visualization helps us to understand which goods from the department are sold the most and how much percentage of goods are sold in each department.



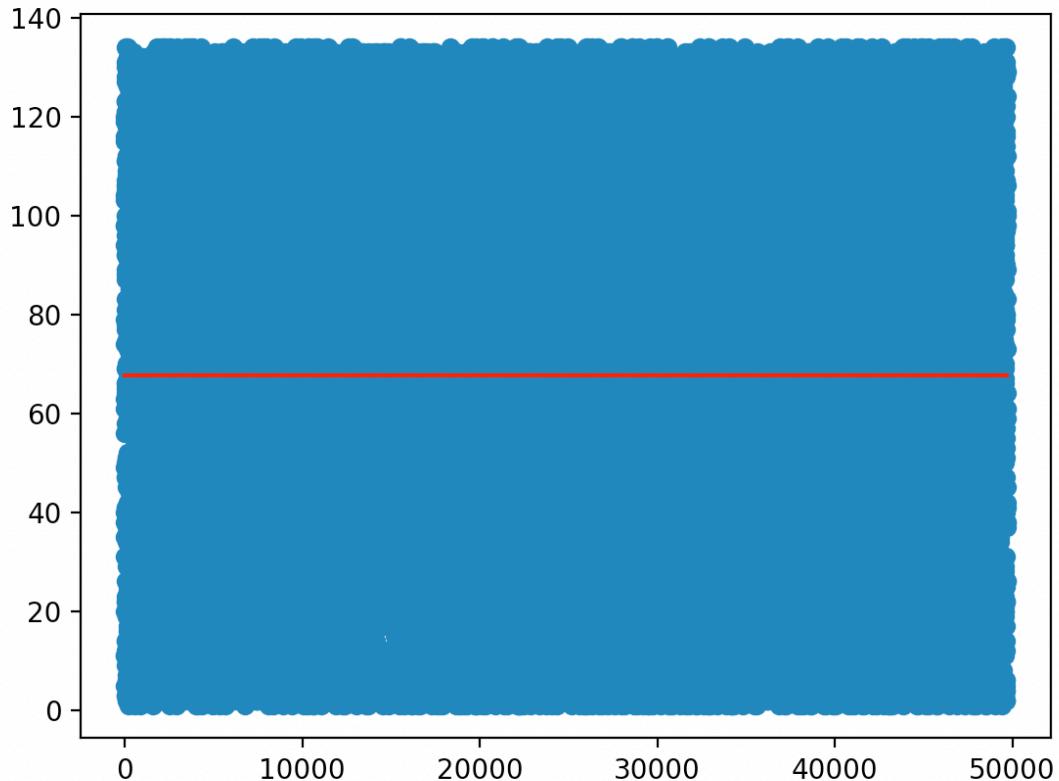
6. Correlation between order_number, order_dow, order_hour_of_day, days_since_prior_order

Where order_number, order_dow, order_hour_of_day, days_since_prior_order is 0,1,2,3 respectively.

By observing the Graph, we can observe that there is no **correlation**.

Null Hypothesis

```
/Users/tharun/PycharmProjects/project01/venv/bin/python /Users/tharun/PycharmProjects/project01/tharun/script_01.py
13.45201534134074
p - values 0.0
we are rejecting null hypothesis
Process finished with exit code 0
```

Regression Analysis

SQL Schema:**Show Tables:**

The screenshot shows the MySQL Workbench interface with a query editor and results grid.

Query Editor (Query 1):

```
Local Instance 3306
Administration Schemas basicQueries
Server Status Client Connections Users and Privileges Status and System Variables Data Export Data Import/Restore
Startup / Shutdown Server Logs Options File
Dashboard Performance Reports Performance Schema Setup
MANAGEMENT INSTANCE PERFORMANCE
Object Info Session
No object selected

51    order_id int,
52    product_id int,
53    add_to_cart_order int,
54    reordered int,
55    FOREIGN KEY (order_id) REFERENCES orders(order_id),
56    FOREIGN KEY (product_id) REFERENCES products(product_id)
57  ;
58
59  show tables;
60

Tables_in_instacart
order_products__prior
order_products__train
orders
products
```

Result Grid:

Tables_in_instacart
order_products__prior
order_products__train
orders
products

Action Output:

Time	Action	Response	Duration / Fetch Time
13:21:49	use instacart	0 row(s) affected	0.00016 sec
13:21:59	show tables	6 row(s) returned	0.0011 sec / 0.00000...

Result 1 Read Only

Query Completed

Products Names and Count Where Reordered=1

Local Instance 3306

Administration Schemas Query 1 basicQueries

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Query 1

basicQueries

Limit to 1000 rows

```
54     reordered int,
55     FOREIGN KEY (order_id) REFERENCES orders(order_id),
56     FOREIGN KEY (product_id) REFERENCES products(product_id)
57   );
58
59   show tables;
60
61   select product_name, count(*) AS count from order_products_prior, products where reordered = 1 and order_products_prior.product_id = products.product_id GROUP BY order_products_prior.product_id
```

100% 219/62

Result Grid Filter Rows: Search Export Fetch rows:

product_name	count
Organic Strawberries	6233
Organic Baby Spinach	5795
Organic Hass Avocado	5190
Organic Avocado	4135
Organic Whole Milk	2441
Large Lemon	3287
Organic Raspberries	3224
Strawberries	3071
Limes	2950
Organic Yellow Onion	2381
Organic Garlic	2281
Organic Zucchini	2234
Cucumber Kirby	2045
Organic Blueberries	1955
Apples Fuji Organic Orga...	1873
Organic Honeydew Melon	1825
Organic Fuji Apple	1820
Organic Lemon	1915
Honeydew Apple	1781
Organic Red Seedless Grap...	1763
Organic Large Extra F...	1751
Organic Cucumber	1731
Organic Gala Apples	1706
Seedless Red Grapes	1692

Object info Session

No object selected

Result 2

Action Output

Time	Action	Response	Duration / Fetch Time
13:21:49	use instacart	0 row(s) affected	0.00016 sec
13:21:59	show tables	6 row(s) returned	0.0011 sec / 0.00000...
13:22:24	select product_name, count(*) AS count from order_products_prior, products where reordered = 1 and order_products_prior.product_id = products.product_id GROUP BY order_products_prior.product_id	1000 row(s) returned	0.948 sec / 0.0054 sec

Query Completed

Product Names and product name with aisle and department ID where reordered =0

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:**

```

62 • select product_name, count(*) AS count from order_products_prior, products where reordered = 1 and order_products_prior.product_id = products.product_id GROUP BY order_product
63
64
65 • select * from products, order_products_train where products.product_id = order_products_train.product_id and reordered = 0;
66
67
68 • SELECT order_hour_of_day, count(*) as count FROM orders GROUP BY order_hour_of_day;
69
70
71 • SELECT COUNT(*) FROM orders WHERE order_hour_of_day > 10 and order_hour_of_day < 12;
    
```
- Result Grid:**

product_id	product_name	aisle_id	department_id	order_id	product_id	add_to_cart_order	reordered
13176	Bag of Organic Bananas	24	4	1	13176	6	0
47209	Organic Hass Avocado	24	4	1	47209	7	0
234	Organic Mozzarella Cheese	2	18	36	234	1	0
48235	Organic Half & Half	53	18	36	48235	3	0
11913	Shelled Pistachios	117	19	36	11913	1	0
18159	Organic Biologique Limes	123	4	36	18159	2	0
4481	Organic Raw Unfiltered Apple Cider Vinegar	13	1	36	4481	3	0
1932	Organic Hot House Tomato	63	4	36	1932	5	0
32433	Green Peas	116	1	36	32433	6	0
28842	Bunched Cilantro	16	4	36	28842	7	0
42625	Flat Parsley, Bunch	16	4	36	42625	8	0
3692	Fresh Cilantro	16	4	36	3692	9	0
10591	Organic Cucumber	83	4	96	10591	2	0
25610	Organic Pomegranate Kernels	116	1	96	25610	4	0
36364	Organic Corn Starch	17	13	98	36364	44	0
32483	Olive Oil & Aloe Vera Hand Soap	25	11	98	32483	45	0
31056	Mulberry Feta	53	17	98	31056	47	0
25359	Organic Coconut Milk	91	16	98	25359	49	0
41860	Sea Salt Baked Potato Chips	107	19	112	41860	3	0
38273	Marinara Pasta Sauce	9	9	112	38273	4	0
47209	Organic Hass Avocado	24	4	112	47209	5	0
234	Organic Mozzarella	2	17	112	234	7	0
9047	Premium Epsom Salt	133	11	112	9047	8	0
4549	Umcka Elderberry Intensive Cold + Flu Ber...	11	11	112	4549	9	0
- Action Output:**

Time	Action	Response	Duration / Fetch Time
1 13:21:49	use instacart	0 row(s) affected	0.00016 sec
2 13:21:59	show tables	0 row(s) returned	0.0011 sec / 0.0000...
3 13:22:24	select product_name, count(*) AS count from order_products_prior, products where reordered = 1 and order_products_prior.product_id = products.product_id GROUP BY order_products_prior.product_id	1000 row(s) returned	0.948 sec / 0.0054 sec
4 13:22:51	select * from products, order_products_train where products.product_id = order_products_train.product_id and reordered = 0 LIMIT 0, 1000	1000 row(s) returned	0.0014 sec / 0.0018 sec

Orders are grouped by order hour of the day

The screenshot shows the MySQL Workbench interface with a query editor and results grid.

Query Editor:

```

64 select * from products, order_products__train where products.product_id = order_products__train.product_id and reordered = 0;
65
66
67
68 SELECT order_hour_of_day, count(*) as count FROM orders GROUP BY order_hour_of_day;
69
70
71 SELECT COUNT(*) FROM orders WHERE order_hour_of_day > 10 and order_hour_of_day < 12;
72
73 select * from aisles;
74
    
```

Result Grid:

order_hour_of_day	count
10	298418
9	257812
17	228795
16	2175033
12	217441
14	203042
6	30529
19	140569
8	178501
15	20339
11	284728
20	104292
18	182912
13	277999
0	2273
22	61468
21	76109
7	91868
1	11658
23	40043
2	7539
4	5827
5	9569
3	5474

Action Output:

Time	Action	Response	Duration / Fetch Time
13:21:49	use instacart	0 row(s) affected	0.00016 sec
13:21:59	show tables	0 row(s) returned	0.0011 sec / 0.00000 sec
13:22:24	select product_name, count(*) AS count from order_products__prior, products where reordered = 1 and order_products__prior.product_id = products.product_id GROUP BY product_name	1000 row(s) returned	0.948 sec / 0.0054 sec
13:22:51	select * from products, order_products__train where products.product_id = order_products__train.product_id and reordered = 0 LIMIT 0, 1000	1000 row(s) returned	0.0014 sec / 0.0018 sec
13:23:27	SELECT order_hour_of_day, count(*) as count FROM orders GROUP BY order_hour_of_day LIMIT 0, 1000	24 row(s) returned	1.220 sec / 0.00008 sec

Order Count done between 10:00 in the morning and 12:00

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

61 -- select product_name, count(*) AS count from order_products__prior, products where reordered = 1 and order_products__prior.product_id = products.product_id GROUP BY order_product
62
63 --
64
65 select * from products, order_products__train where products.product_id = order_products__train.product_id and reordered = 0;
66
67 --
68 SELECT order_hour_of_day, count(*) as count FROM orders GROUP BY order_hour_of_day;
69
70 --
71 SELECT COUNT(*) FROM orders WHERE order_hour_of_day > 10 and order_hour_of_day < 12;
72
73 select * from aisles;
74 select * from departments;
    
```
- Result Grid:** Shows the result of the final query: COUNT(*) = 264728.
- Action Output:** Displays the history of actions taken during the session:

Time	Action	Response	Duration / Fetch Time
13:21:49	use instacart	0 row(s) affected	0.00016 sec
13:21:59	show tables	6 row(s) returned	0.0011 sec / 0.0000...
13:22:24	select product_name, count(*) AS count from order_products__prior, products where reordered = 1 and order_products__prior.product_id = products.product_id GROUP BY order_product	1000 row(s) returned	0.948 sec / 0.0054 sec
13:22:51	select * from products, order_products__train where products.product_id = order_products__train.product_id and reordered = 0 LIMIT 0, 1000	1000 row(s) returned	0.0014 sec / 0.0018 s...
13:23:27	SELECT order_hour_of_day, count(*) as count FROM orders GROUP BY order_hour_of_day LIMIT 0, 1000	24 row(s) returned	1.220 sec / 0.00008...
13:23:47	SELECT COUNT(*) FROM orders WHERE order_hour_of_day > 10 and order_hour_of_day < 12 LIMIT 0, 1000	1 row(s) returned	0.637 sec / 0.000006...
- Status Bar:** Shows "Query Completed".

CONCLUSION:

From above visualizations, we can observe that the most products sold, and interesting insights, which are drawn from graphs. It is good to understand which are the products the customer need more, and what times do the customers order more so that more employees can be employed in that stipulated time. It also shows how many customers reordered the items at least once and how many customer didn't reorder.

REFERENCES:

References:

- [1] “The Instacart Online Grocery Shopping Dataset 2017”, Accessed from <https://www.instacart.com/datasets/grocery-shopping-2017> [Accessed: 21-Nov-2019]
- [2] “Instacart Market Basket Analysis.” [Online]. Available: <https://kaggle.com/c/instacart-market-basket-analysis>. [Accessed: 21-Nov-2019]
- [3] “What Is Instacart? A Guide on How to Use the Grocery Delivery App,” Digital Trends, 19-Sep-2018. [Online]. Available: <https://www.digitaltrends.com/home/what-is-instacart/>. [Accessed: 21-Nov-2019]
- [4] J. Stanley, “3 Million Instacart Orders, Open Sourced,” *Medium*, 04-May-2017. [Online]. Available: <https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2>. [Accessed: 21-Nov-2019]