This is an E-commerce data analysis consists of multiple tables, constructed by Yosua Saputra. (Time Limit: 24-hour project)
Dataset was found on Kaggle: https://www.kaggle.com/olistbr/brazilian-ecommerce?select=olist_order_payments_dataset.csv

---

Context:

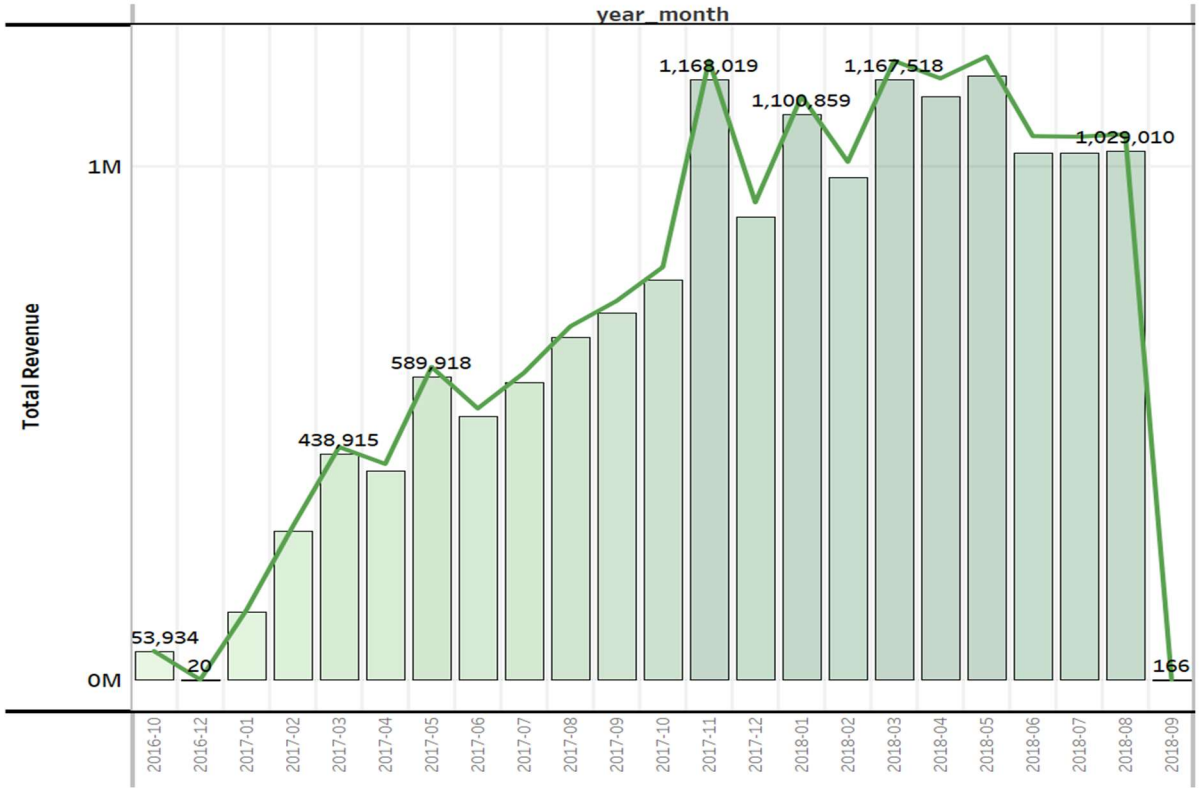
This dataset was generously provided by Olist, the largest department store in Brazilian marketplaces.
Olist connects small businesses from all over Brazil to channels without hassle and with a single contract.
Those merchants are able to sell their products through the Olist Store and ship them directly to the customers
using Olist logistics partners. See more on our website: www.olist.com
After a customer purchases the product from Olist Store a seller gets notified to fulfill that order. Once the customer receives
the product, or the estimated delivery date is due, the customer gets a satisfaction survey by email where he can give a note
for the purchase experience and write down some comments.

Task:

My goal for this project is to provide insights about how the business is going and potentially provide new ideas to improve the market going forward.

First, how do we measure the success of this e-commerce market.
I want to know based off each year which months has the highest revenue.

```
WITH cte AS (
    SELECT o.order_id, customer_id, order_status, order_approved_at, payment_value,
        FORMAT(order_approved_at, 'yyyy-MM') AS 'year_month'
        --FORMAT(order_approved_at, 'MM') AS 'month'
    FROM olist_orders_dataset o
    JOIN olist_order_payments_dataset p
    ON o.order_id = p.order_id
    WHERE order_status != 'canceled'
    AND order_approved_at IS NOT NULL)
SELECT year_month , ROUND(SUM(payment_value),2) AS total_revenue
FROM cte
GROUP BY year_month
ORDER BY 2 DESC
```



As we can see the revenue has an upward trend as the year progresses since the data we have for 2016. This is an indicator of success over the period

Let’s take a deeper dive into month-by-month revenue percentage:

```
WITH cte1 AS (
    SELECT *,
        LAG(total_revenue) OVER(ORDER BY year_month) AS previous_month
    FROM(
        SELECT year_month, ROUND(SUM(payment_value), 2) AS total_revenue
        FROM(
            SELECT payment_value,
                FORMAT(order_approved_at, 'yyyy-MM') AS 'year_month'
            FROM olist_orders_dataset o
            JOIN olist_order_payments_dataset p
            ON o.order_id = p.order_id
            WHERE order_status != 'canceled'
            AND order_approved_at IS NOT NULL) x
        GROUP BY year_month) y)
SELECT *,
    ROUND((total_revenue-previous_month)/previous_month * 100, 2) AS monthly_pct
FROM cte1
WHERE ROUND((total_revenue-previous_month)/previous_month * 100, 2) < 0
```

	year_month	total_revenue	previous_month	monthly_pct
1	2016-12	19.62	53934.25	-99.96
2	2017-04	407008.31	438915.25	-7.27
3	2017-06	511659.09	589918.11	-13.27
4	2017-12	900447.53	1168019.23	-22.91
5	2018-02	976835.36	1100858.64	-11.27
6	2018-04	1134366.45	1167517.57	-2.84
7	2018-06	1025357.36	1175578.52	-12.78
8	2018-07	1024297.03	1025357.36	-0.1
9	2018-09	166.46	1029010.04	-99.98

After showing year_months that have slowed down from its previous months in terms of revenue, we see that December of 2016 and September of 2018 sees a drop in total revenue by nearly 100%. Given that these are the first and latest months recorded on our original data, we want to assume not all days of these months were tracked which led to an inflated result of monthly loss.

```
SELECT year_month, COUNT(payment_value) AS total_countof_payments
FROM(
    SELECT o.order_id, customer_id, order_approved_at, payment_value,
        FORMAT(order_approved_at, 'yyyy-MM') AS year_month
    FROM olist_orders_dataset o
    JOIN olist_order_payments_dataset p
    ON o.order_id = p.order_id
    WHERE order_status != 'canceled'
    AND order_approved_at IS NOT NULL) a
GROUP BY year_month
ORDER BY 2
```

	year_month	total_countof_payments
1	2016-12	1
2	2018-09	1
3	2016-10	318
4	2017-01	807
5	2017-02	1857
6	2017-04	2528
7	2017-03	2813
8	2017-06	3424
9	2017-05	3917
10	2017-07	4233
11	2017-09	4525
12	2017-08	4531

	year_month	total_countof_payments
12	2017-08	4531
13	2017-10	4793
14	2017-12	6057
15	2018-07	6346
16	2018-06	6398
17	2018-08	6778
18	2018-02	6854
19	2018-04	7021
20	2018-05	7310
21	2018-01	7448
22	2018-03	7567
23	2017-11	7676

Surely enough there were only 1 payment recorded on both those months.

So far in our analysis we conclude that Olist, the largest department store in Brazilian marketplaces, is heading towards the right direction in terms of generating payment values especially in the year 2017. However, we also notice that the growth rate is slowly diminishing in 2018. Olist may need additional solutions to continue its market's success.

Metrics suggested to improve Olist's success:

How most e-commerce sites make money is through advertisement campaigns and commission from its sellers. Therefore, my proposal would be to look at how engaged our sellers are in the market, improving sellers/buyer's experience, and improving ways in implementing our advertisements.

1. User's experience
    - Buyers and Sellers p.o.v:
        - How many of our sellers would be considered active sellers?
        - Are buyers satisfied with their experience through olist?

2. Marketing/advertisement campaigns:
    - Although we do not have data about this topic, we can think about when the right time would be to implement our marketing campaigns/ads.
    - Suggested location for the campaign.

3. Additional features that could potentially help the experience/success of the business

---

1. USER'S EXPERIENCE

Customer Engagement (Active sellers, buyer's experience)

To determine whether a seller is active or not we will assume:
- Really Active: Seller sold an approved order from at least 500 different dates
- Active: Seller sold an approved order from 250-499 different dates
- Somewhat Active: Seller sold an approved order from 50-249 different dates
- Somewhat Not Active: Seller sold an approved order from 10-49 different dates
- Not Active: Seller sold an approved order less than 10 different dates

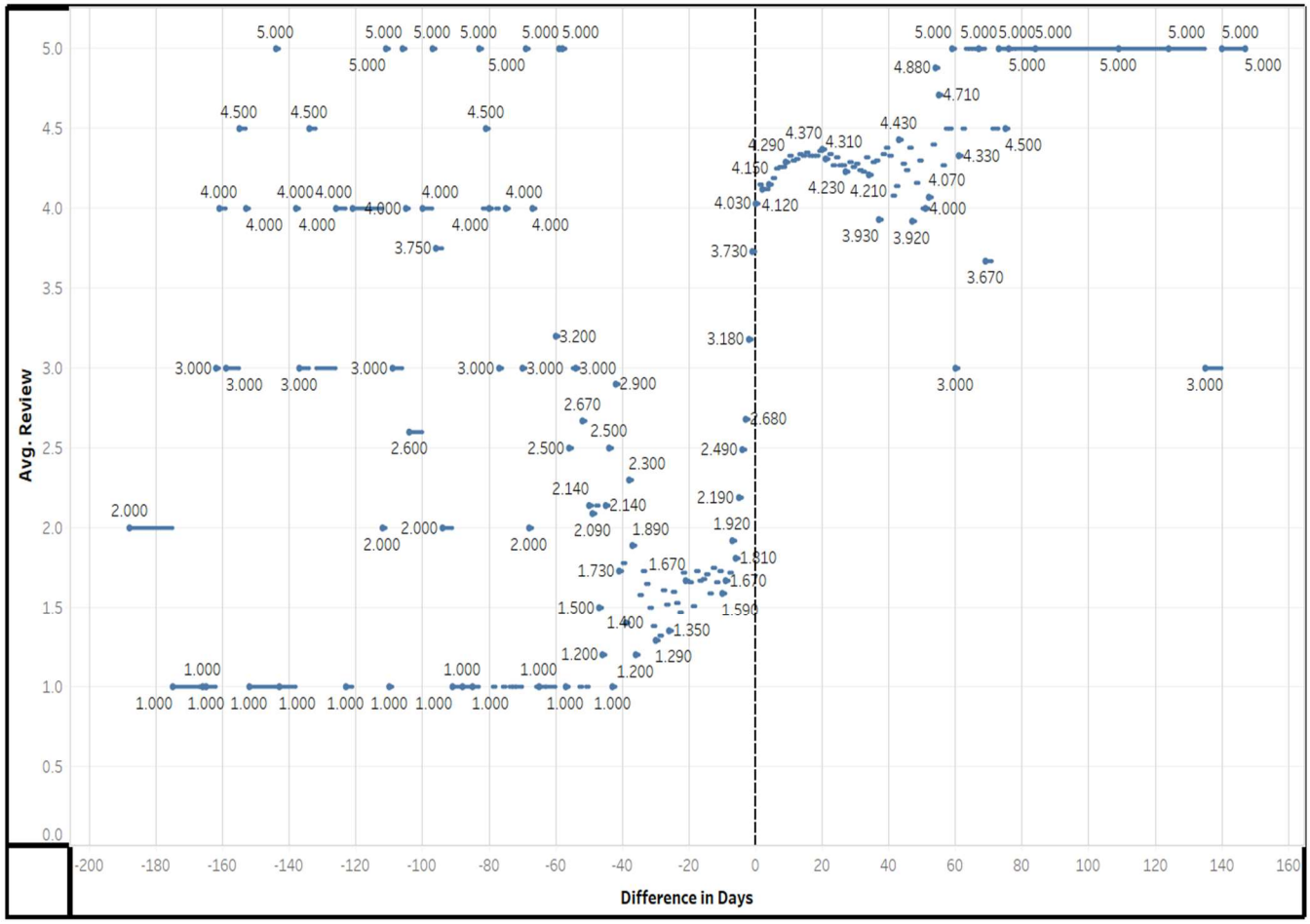
```
WITH total_engaged_seller AS (
SELECT seller_id,
    CASE WHEN distinct_order_dates >= 500 THEN 'Really_Active'
        WHEN distinct_order_dates BETWEEN 250 AND 499 THEN 'Active'
        WHEN distinct_order_dates BETWEEN 50 AND 249 THEN 'Somewhat_Active'
        WHEN distinct_order_dates BETWEEN 10 AND 49 THEN 'Somewhat_not_Active'
        WHEN distinct_order_dates < 10 THEN 'Not_Active'
        END AS engagement
FROM (
SELECT seller_id, COUNT(DISTINCT order_approved_at) AS distinct_order_dates
FROM olist_order_items_dataset oi
JOIN olist_orders_dataset o
ON oi.order_id = o.order_id
GROUP BY seller_id) seller_activity
)
SELECT engagement, COUNT(*) AS total_engaged_results
FROM total_engaged_seller
GROUP BY engagement
ORDER BY 2
```

	engagement	total_engaged_results
1	Really_Active	26
2	Active	40
3	Somewhat_Active	362
4	Somewhat_not_Active	843
5	Not_Active	1824

This query shows us that most of the sellers are inactive. For one it could mean most sellers are individuals and not already a business selling partner with olist.

Next, we will analyze the buyers and their experience. Given their review_score I will want to know the average based on the difference between their estimated shipping and how long it actually took, as well as the average rating in which products/product-category scored the highest average reviews. This will also give valuable insights to sellers as well.

```
WITH shipping_difference AS (
	SELECT order_id, review_score,
		DATEDIFF(day, order_approved_at, order_estimated_delivery_date) AS
est_duration,
		DATEDIFF(day, order_approved_at, order_delivered_customer_date) AS
actual_duration,
		DATEDIFF(day, order_approved_at, order_estimated_delivery_date) -
DATEDIFF(day, order_approved_at, order_delivered_customer_date)
		AS diff
	FROM (
		SELECT review_id, r.order_id, review_score, order_approved_at,
order_delivered_customer_date, order_estimated_delivery_date
		FROM olist_order_reviews_dataset r
		JOIN olist_orders_dataset o
		ON r.order_id = o.order_id
		WHERE order_status != 'canceled'
		AND order_approved_at IS NOT NULL) x)
SELECT diff, ROUND(AVG(review_score), 2) AS avg_review
FROM shipping_difference
GROUP BY diff
ORDER BY diff
```



A few reviews scored well even though the shipping process was delayed tremendously. However, we can safely conclude that those who's orders were on time or even days or months ahead of its estimated schedule received the highest reviews.

For the products:

```
SELECT COALESCE(product_category_name, 'Other') AS product_category_name,
COUNT(product_category_name) AS amount,
	AVG(DATEDIFF(day, order_approved_at, order_estimated_delivery_date) -
DATEDIFF(day, order_approved_at, order_delivered_customer_date))
	AS shipping_diff,
	ROUND(AVG(review_score), 2) AS avg_review
FROM(
	SELECT r.order_id, review_score, order_approved_at, order_delivered_customer_date,
order_estimated_delivery_date,
		oi.product_id, product_category_name
	FROM olist_order_reviews_dataset r
	JOIN olist_orders_dataset o
	ON r.order_id = o.order_id
```

```
        JOIN olist_order_items_dataset oi
        ON o.order_id = oi.order_id
        JOIN olist_products_dataset p
        ON oi.product_id = p.product_id
        WHERE order_status != 'canceled'
        AND order_approved_at IS NOT NULL) category_reviews
GROUP BY product_category_name
ORDER BY 4 DESC
```

	product_category_name	amount	shipping_diff	avg_review
1	cds_dvds_musicais	14	16	4.64
2	fashion_roupa_infanto_juvenil	8	15	4.5
3	livros_interesse_geral	542	12	4.48
4	construcao_ferramentas_ferramentas	99	12	4.44
5	flores	31	12	4.42
6	livros_importados	60	11	4.4
7	livros_tecnicos	266	11	4.37
8	alimentos_bebidas	278	11	4.33
9	malas_acessorios	1084	12	4.33
10	portateis_casa_forno_e_cafe	76	11	4.3
11	fashion_esporte	31	12	4.26
12	fashion_calcados	259	14	4.26
13	alimentos	494	9	4.22
14	cine_foto	73	11	4.21
15	musica	38	14	4.21
16	papelaria	2496	12	4.2
17	pet_shop	1934	12	4.19
18	instrumentos_musicais	664	11	4.19
19	eletrodomesticos	804	12	4.18
20	eletroportateis	669	13	4.18
21	brinquedos	4057	12	4.18
22	dvds_blu_ray	61	13	4.18
23	perfumaria	3406	12	4.17
24	pcs	200	11	4.17
25	eletrodomesticos_2	236	12	4.16
26	cool_stuff	3756	12	4.16
27	fashion_bolsas_e_acessorios	2033	13	4.15
28	beleza_saude	9609	12	4.15
29	artes_e_artesanato	24	6	4.13
30	esporte_lazer	8588	12	4.12
31	moveis_quarto	110	13	4.12

Query executed successfully.

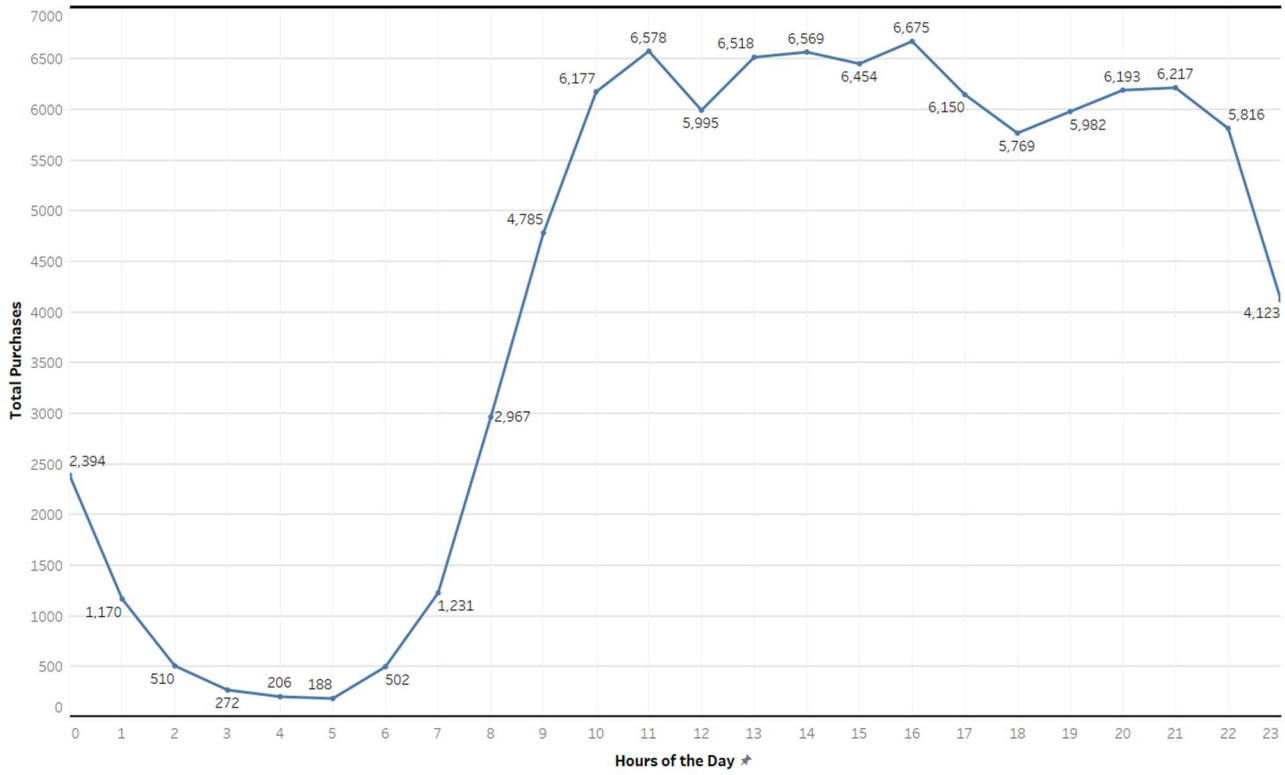
From our output we see that average reviews by category name does not correlate with how much the shipping date deviates from the estimated shipping date. Also, to take into consideration the amount ordered per each product's category.

2. MARKETING/ADVERTISEMENT CAMPAIGN:

Our data does not give us values about revenue associating with ads or marketing campaigns. However, many of the largest-commerce sites make money through advertisement campaigns and in this section I will analyze the right place or time to implement these additional measures.

First, we want to look at what times during the given day orders are being placed.

```
SELECT hour_day, COUNT(*) AS total_purchases
FROM(
	SELECT order_purchase_timestamp, FORMAT(order_purchase_timestamp,'HH') AS hour_day
	FROM olist_orders_dataset) hours_purchased
GROUP BY hour_day
ORDER BY total_purchases DESC
```



Most products are bought around 4:00 pm, 11:00 am, 2:00pm, 1:00pm, 3:00 pm, 9:00 pm, 8:00pm, respectively. Least active purchases around 5:00 am, 4:00 am, 3:00 am, 6:00 am , assuming when most people are usually not awake. This could possibly show the times during the days when online traffic spike as well. Why could this be helpful? Given this data, product specific ads can be implemented during the peak hours when most people are browsing the web. Keep in mind the cost to produce the ongoing marketing campaign can affect the profit margins for the better or worse.

Now, we will look at the location which would best be appropriate to implement potential ads to maximize profit. For this example, I will write a query for busiest hours during the day, of the top ten state-city and its total number purchases.

```
WITH top_10 AS (
	SELECT customer_state, customer_city
	FROM(
		SELECT *, RANK() OVER(ORDER BY total_purchases_count DESC) AS ranking
		FROM (
			SELECT customer_state, customer_city, COUNT(*) AS
total_purchases_count
			FROM olist_customers_dataset c
			JOIN olist_orders_dataset o
			ON c.customer_id = o.customer_id
			WHERE order_status != 'canceled'
			AND order_approved_at IS NOT NULL
			GROUP BY customer_state, customer_city) ranks) rankss
			WHERE ranking BETWEEN 1 AND 10)

SELECT customer_state, customer_city, hour_day, COUNT(hour_day) AS total
FROM
	(SELECT customer_state, customer_city, order_purchase_timestamp,
		FORMAT(order_purchase_timestamp,'HH') AS hour_day
	FROM olist_customers_dataset c
	JOIN olist_orders_dataset o
	ON c.customer_id = o.customer_id) location_time
	WHERE EXISTS (
		SELECT 1
		FROM top_10
		WHERE location_time.customer_state = top_10.customer_state
		AND location_time.customer_city = top_10.customer_city)
		GROUP BY customer_state, customer_city, hour_day
		ORDER BY total DESC
```

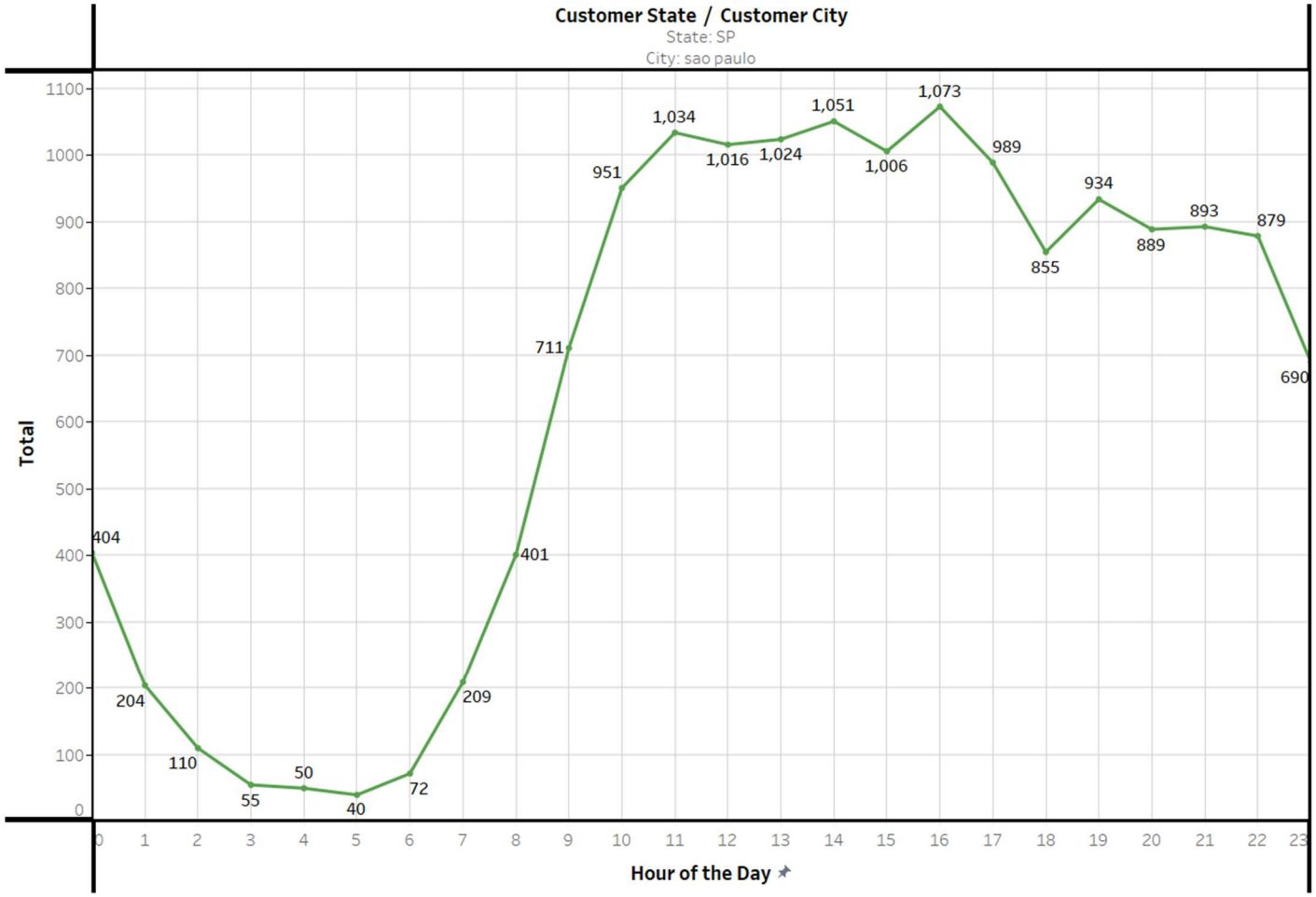
	customer_state	customer_city	hour_day	total
1	SP	sao paulo	16	1073
2	SP	sao paulo	14	1051
3	SP	sao paulo	11	1034
4	SP	sao paulo	13	1024
5	SP	sao paulo	12	1016
6	SP	sao paulo	15	1006
7	SP	sao paulo	17	989
8	SP	sao paulo	10	951
9	SP	sao paulo	19	934
10	SP	sao paulo	21	893
11	SP	sao paulo	20	889
12	SP	sao paulo	22	879
13	SP	sao paulo	18	855
14	SP	sao paulo	09	711
15	SP	sao paulo	23	690
16	RJ	rio de janeiro	14	499
17	RJ	rio de janeiro	11	475
18	RJ	rio de janeiro	16	461
19	RJ	rio de janeiro	15	459
20	RJ	rio de janeiro	18	451
21	RJ	rio de janeiro	17	438
22	RJ	rio de janeiro	13	436
23	RJ	rio de janeiro	21	424
24	RJ	rio de janeiro	12	420
25	RJ	rio de janeiro	19	413
26	RJ	rio de janeiro	10	411
27	SP	sao paulo	00	404
28	SP	sao paulo	08	401
29	RJ	rio de janeiro	20	387
30	RJ	rio de janeiro	22	372
31	RJ	rio de janeiro	09	301
32	RJ	rio de janeiro	23	267
33	SP	sao paulo	07	209
34	MG	belo horizo...	16	207
35	SP	sao paulo	01	204
36	MG	belo horizo...	17	200
37	MG	belo horizo...	14	199
38	MG	belo horizo...	11	197
39	RJ	rio de janeiro	00	190
40	MG	belo horizo...	21	176
41	MG	belo horizo...	10	175
42	MG	belo horizo...	13	175

	customer_state	customer_city	hour_day	total
43	MG	belo horizo...	18	173
44	RJ	rio de janeiro	08	168
45	MG	belo horizo...	15	166
46	MG	belo horizo...	12	166
47	MG	belo horizo...	19	162
48	DF	brasilia	11	161
49	DF	brasilia	16	159
50	MG	belo horizo...	20	158
51	DF	brasilia	14	150
52	DF	brasilia	21	144
53	MG	belo horizo...	22	144
54	DF	brasilia	12	142
55	MG	belo horizo...	09	138
56	DF	brasilia	20	137
57	DF	brasilia	17	135
58	DF	brasilia	15	132
59	DF	brasilia	10	127
60	DF	brasilia	13	126
61	DF	brasilia	19	125
62	PR	curitiba	15	125
63	DF	brasilia	22	119
64	DF	brasilia	18	115
65	SP	sao paulo	02	110
66	PR	curitiba	13	109
67	PR	curitiba	20	108
68	RS	porto alegre	14	107
69	MG	belo horizo...	08	107
70	PR	curitiba	22	106
71	SP	campinas	10	105
72	SP	campinas	13	100
73	SP	guarulhos	21	100
74	PR	curitiba	11	100
75	SP	campinas	11	98
76	SP	campinas	14	97
77	RJ	rio de janeiro	01	97
78	DF	brasilia	09	97
79	PR	curitiba	16	96
80	SP	campinas	20	96
81	RS	porto alegre	22	95
82	PR	curitiba	14	95
83	PR	curitiba	10	95
84	PR	curitiba	18	94

	customer_state	customer_city	hour_day	total
85	RS	porto alegre	15	94
86	MG	belo horizo...	23	94
87	RS	porto alegre	13	92
88	SP	campinas	16	92
89	SP	campinas	22	91
90	SP	campinas	15	91
91	SP	campinas	12	91
92	SP	guarulhos	17	89
93	SP	campinas	19	88
94	PR	curitiba	19	88
95	PR	curitiba	21	87
96	RJ	rio de janeiro	07	87
97	DF	brasilia	23	86
98	RS	porto alegre	20	86
99	SP	guarulhos	15	86
100	RS	porto alegre	12	83
101	RS	porto alegre	10	82
102	RS	porto alegre	18	82
103	SP	guarulhos	10	81
104	PR	curitiba	17	81
105	BA	salvador	14	81
106	BA	salvador	21	81
107	BA	salvador	22	81
108	BA	salvador	16	81
109	RS	porto alegre	16	80
110	RS	porto alegre	19	80
111	RS	porto alegre	21	79
112	PR	curitiba	12	79
113	BA	salvador	17	79
114	BA	salvador	09	79
115	RS	porto alegre	09	78
116	SP	campinas	17	78
117	SP	guarulhos	16	77
118	RS	porto alegre	17	77
119	RS	porto alegre	11	76
120	SP	campinas	18	76
121	SP	guarulhos	13	75
122	PR	curitiba	09	75
123	BA	salvador	13	75
124	BA	salvador	20	74
125	BA	salvador	11	74
126	SP	guarulhos	11	74

	customer_state	customer_city	hour_day	total
127	SP	campinas	21	74
128	SP	campinas	09	74
129	SP	guarulhos	12	73
130	SP	guarulhos	22	73
131	BA	salvador	18	73
132	SP	guarulhos	19	72
133	SP	sao paulo	06	72
134	SP	sao bernard...	20	70
135	BA	salvador	12	70
136	BA	salvador	23	68
137	PR	curitiba	23	67
138	SP	sao bernard...	18	67
139	SP	sao bernard...	16	66
140	SP	guarulhos	14	66
141	BA	salvador	10	65
142	BA	salvador	19	64
143	SP	guarulhos	18	64
144	SP	sao bernard...	12	64
145	SP	sao bernard...	17	62
146	SP	sao bernard...	14	62
147	BA	salvador	15	62
148	SP	sao bernard...	15	60
149	SP	campinas	23	60
150	RS	porto alegre	23	59
151	DF	brasilia	08	59
152	SP	guarulhos	20	58
153	SP	sao bernard...	21	57
154	SP	sao bernard...	22	56
155	SP	sao bernard...	13	56
156	SP	sao bernard...	11	56
157	SP	guarulhos	23	56
158	SP	sao paulo	03	55
159	SP	guarulhos	09	53
160	SP	campinas	08	50
161	SP	sao paulo	04	50
162	RJ	rio de janeiro	02	47
163	RS	porto alegre	00	47
164	BA	salvador	00	46
165	DF	brasilia	00	45
166	SP	sao bernard...	10	45
167	SP	sao bernard...	09	44
168	SP	sao bernard...	19	43

Here is a graph of hours during the day, of the number one state-city and its total number purchases:



The query shows the top 10 most engaged location by state and city based off total count of orders that have been purchased. Then outputs the busiest hours during the 24-hour day. The results shows that the city of Sao Paulo which is in the state 'SP' is the busiest location and within this exact location at 4:00 pm is where online traffic is at its peak.

This could give olist an idea now of where or where not to implement its market strategies with an addition to its given time during the day.

3. ADDITIONAL FEATURES THAT COULD HELP INCREASE LEVEL OF EXPERIENCE/SUCCESS

This last section is to provide additional assumptions on improving olist moving forward. As we saw earlier in our analysis, the customer's ratings have a correlation with how far off its order's shipping estimate is from the actual shipping date. Most being days, weeks, or sometimes even months keeping the customers waiting.

To improve the customer's experience even further, olist can look to consider e-commerce subscription plans. For example, the largest and most successful e-commerce company, Amazon, has its very own subscription program where they charge its users a monthly or annual fee. By subscribing and becoming a member they get access to faster shipping, exclusive deals, and many more perks. This could potentially enhance customers' experience and in addition to generating more profits for olist as a whole with a fair and strategized price.