Jumpan23

On-demand delivery platform

Sarnia Sulaiman

Table of Contents

5,214

orders placed on the platform

3,192

unique customers

898

merchant partners

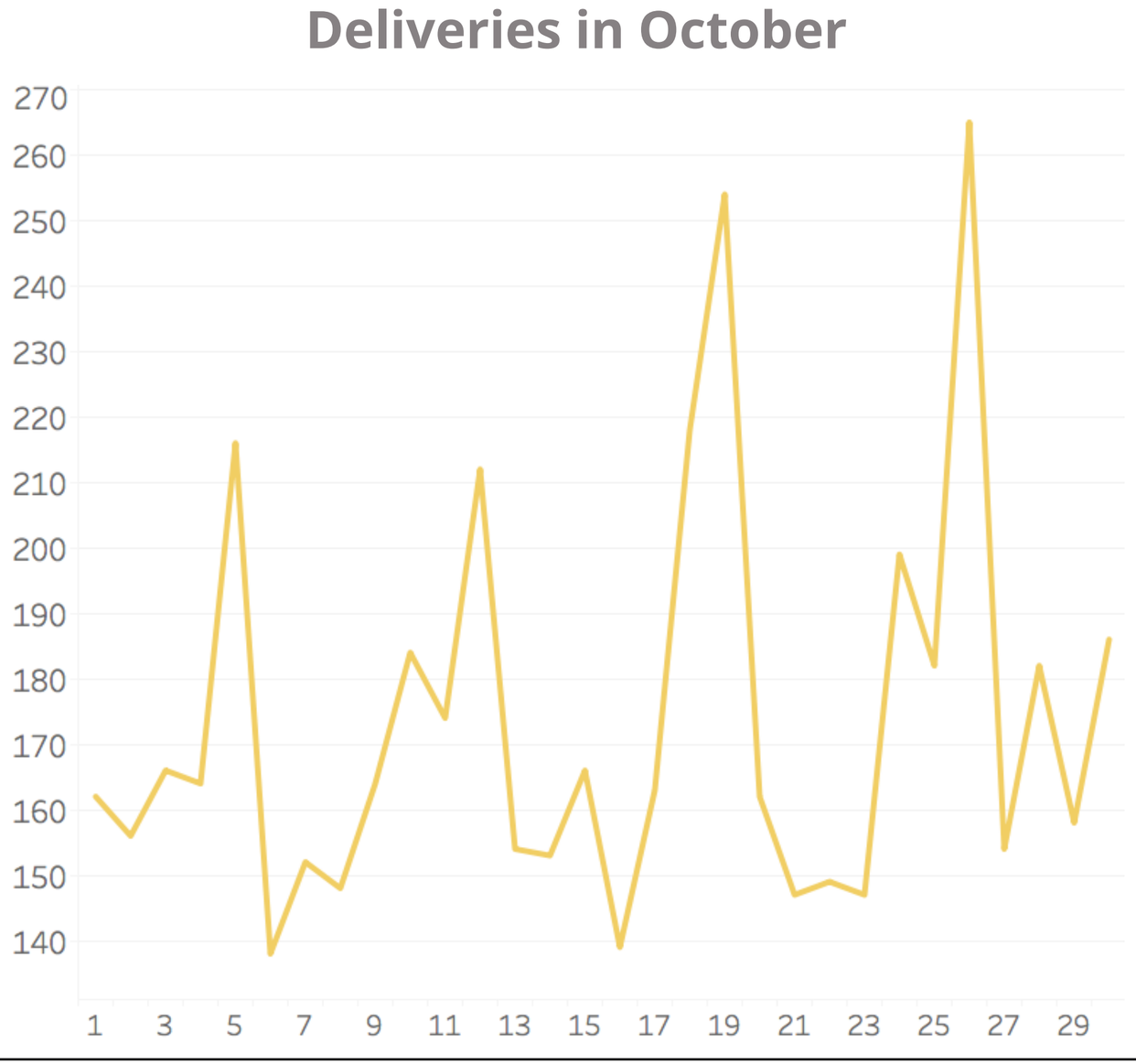
Delivery Trends

Key Trends

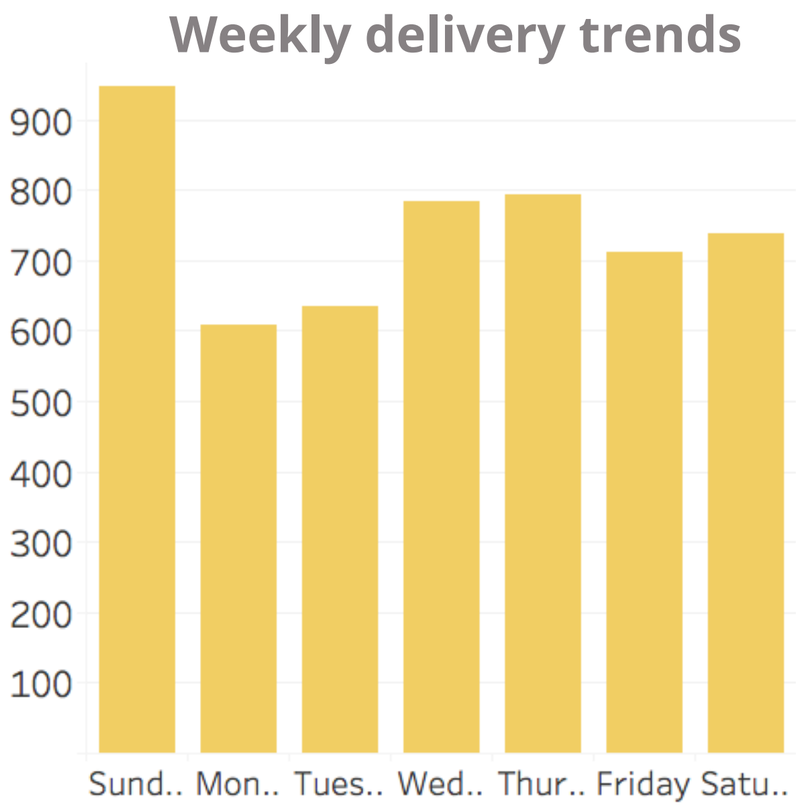
The graph illustrates the number of orders per day in October

A cyclical pattern in the data shows a peak in the number of deliveries on Sundays

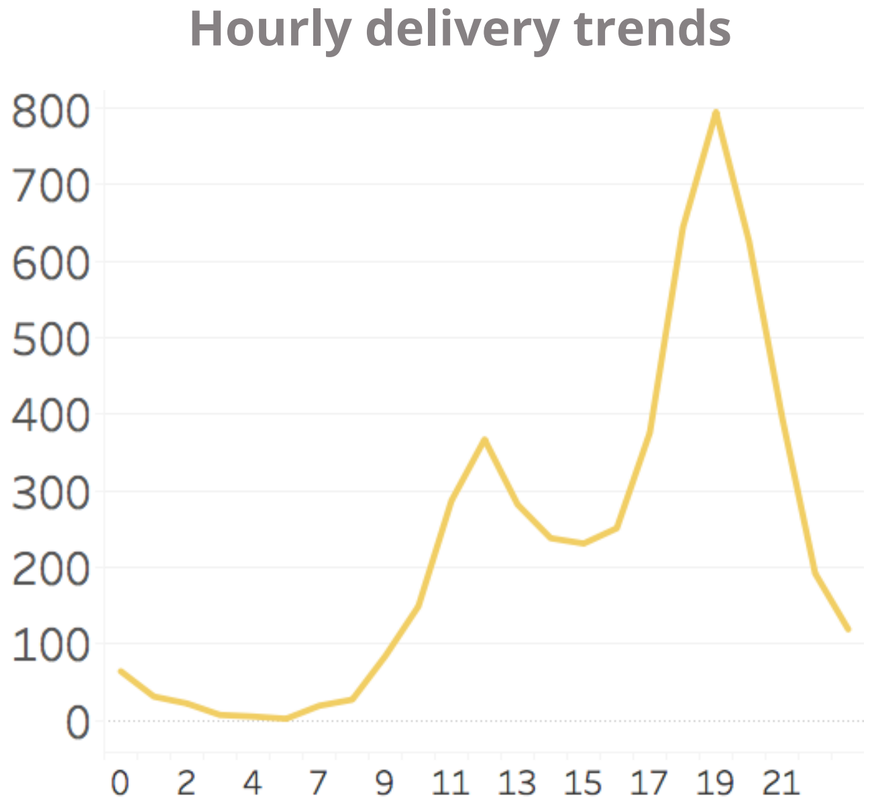
The increase in deliveries in the last half of October suggest that Jumpman23 may be growing in NYC



Delivery Trends



Sunday has the highest number of deliveries, followed by Thursday and Wednesday.



The two peak hours for delivery are at 12 pm and at 7 pm

Delivery Trends

Delivery time & transportation

Transportation

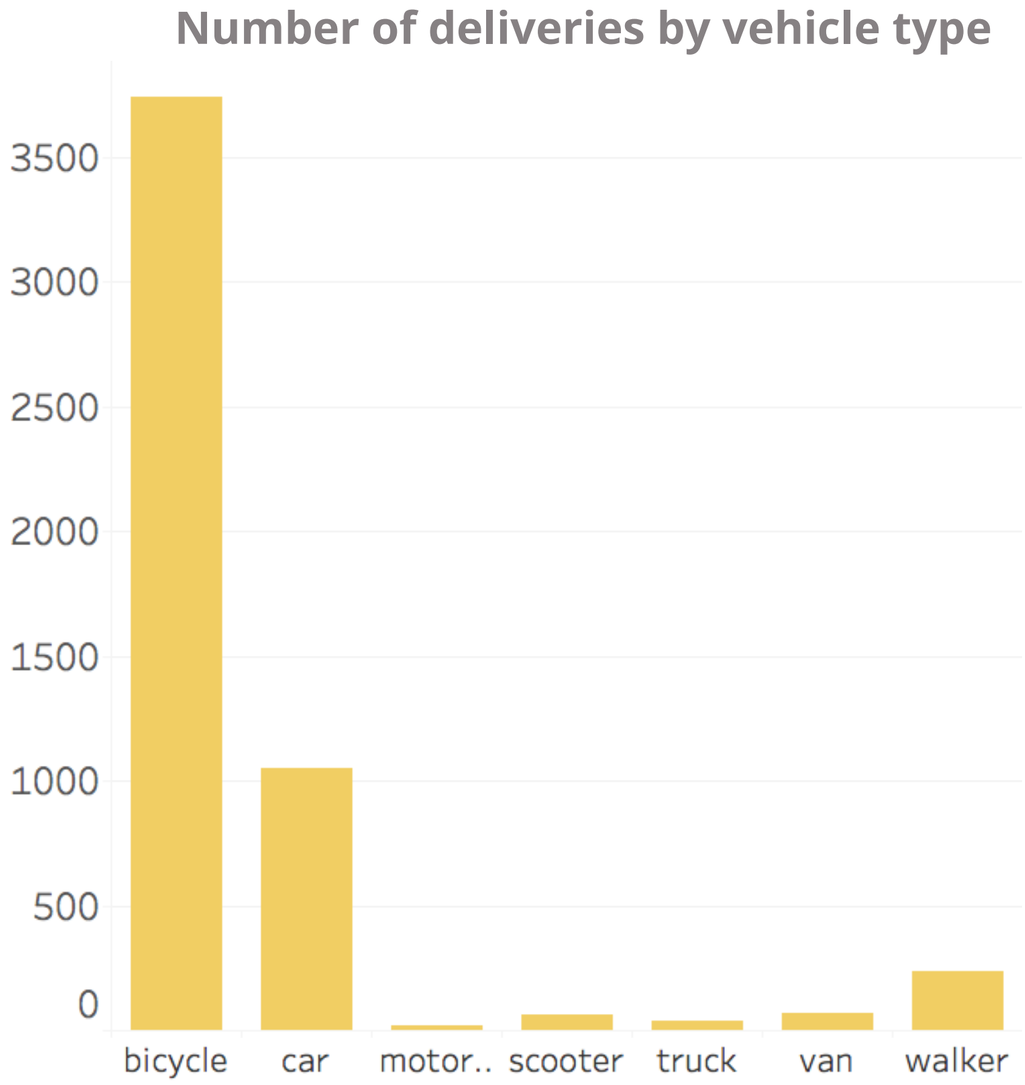
The most popular type of transportation for jumpman are bicycles, followed by cars.

Delivery time

Avg time the customer took to order: 7.7 mins
Avg prep time*: 18 mins
Avg transit time: 14 mins
Avg delivery time: 45 mins

Distance

Avg distance: 0.81 miles



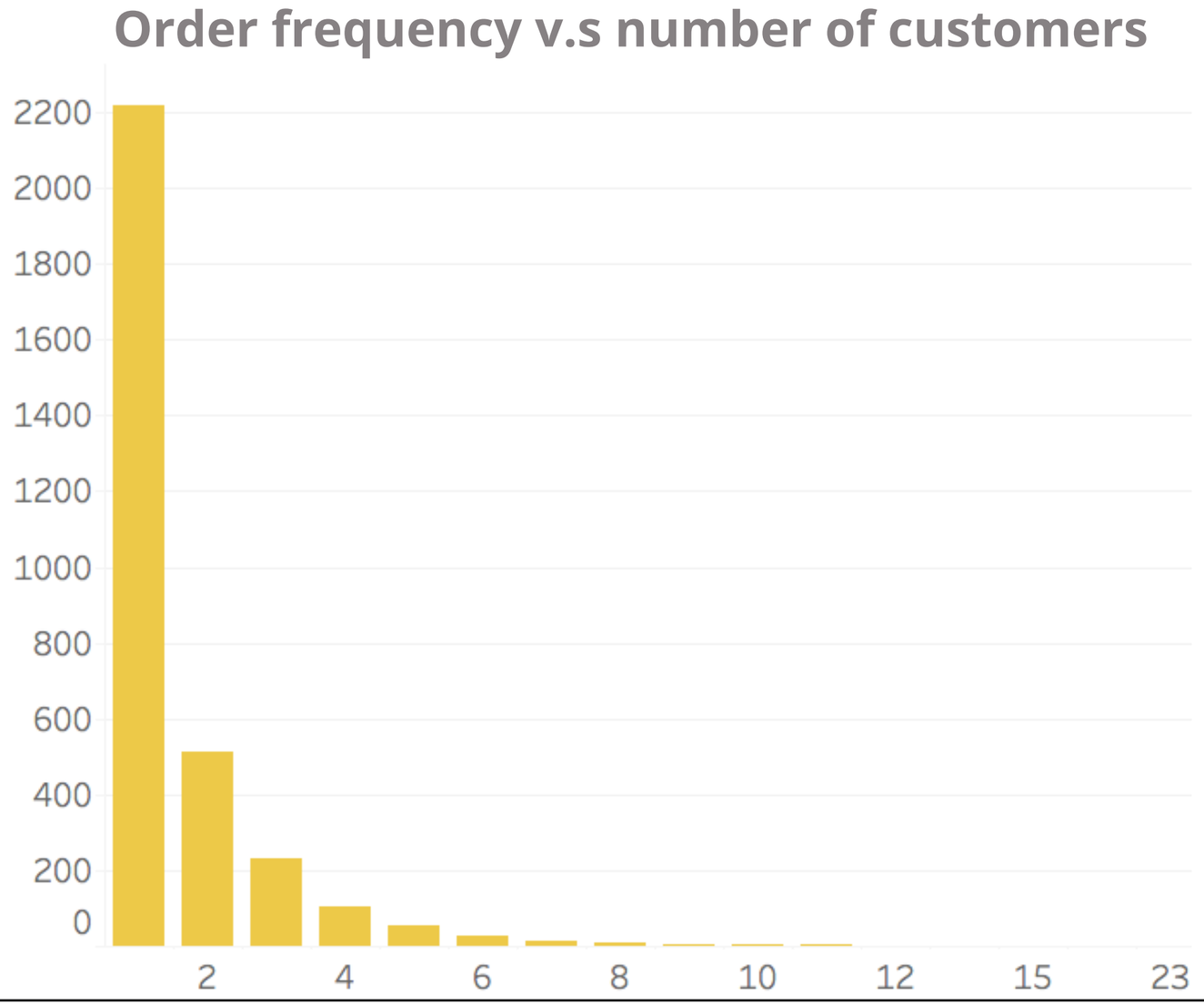
*prep time is defined as the time the jumpman waited at the restaurant

Customer Insights

Customer Order Frequency

According to the Customer Order Frequency graph:

- 30% of customers order more than once
- 15% of customers order more than two times
- 7% of customers order more than 3 times
- <4 % of customers order more than 4 times

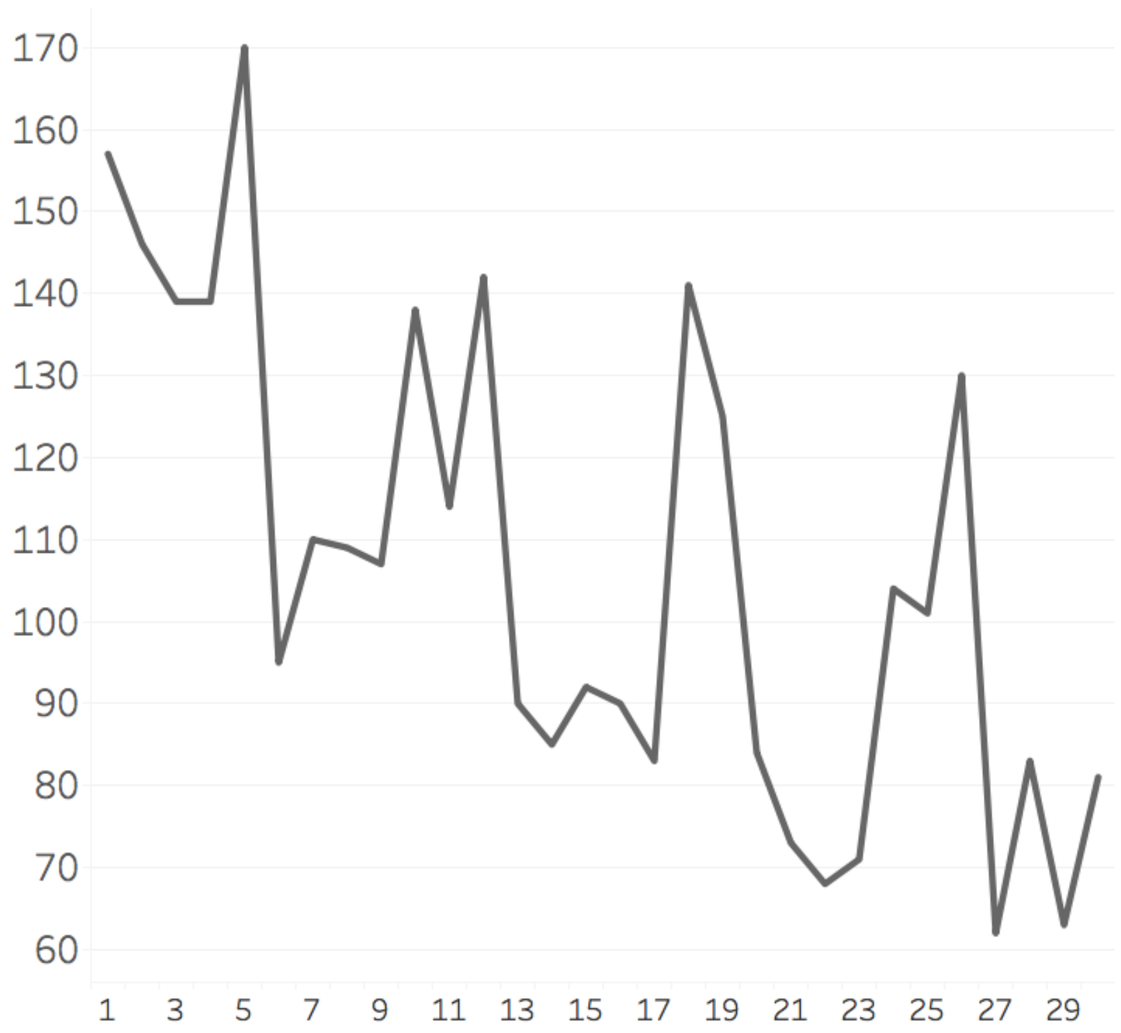


Customer Insights

New customers

The average number of new customers per day in October is ~106

The rate of new customers per day appears to be decreasing over time



Merchant Insights

Top Categories and Merchants

Top 10 Category

Italian
Burger
American
Japanese
Dessert
Chinese
Sushi
Salad
Mexican
Grocery Store

Top 10 Merchants

Shake Shack
Momofuku Milk Bar
The Meatball Shop
sweetgreen
Blue Ribbon Fried Chicken
Blue Ribbon Sushi
Parm
Whole Foods Market
Chipotle Mexica Grill
Mighty Quinn's BBQ

53%

of all orders are from the top 10 category

24%

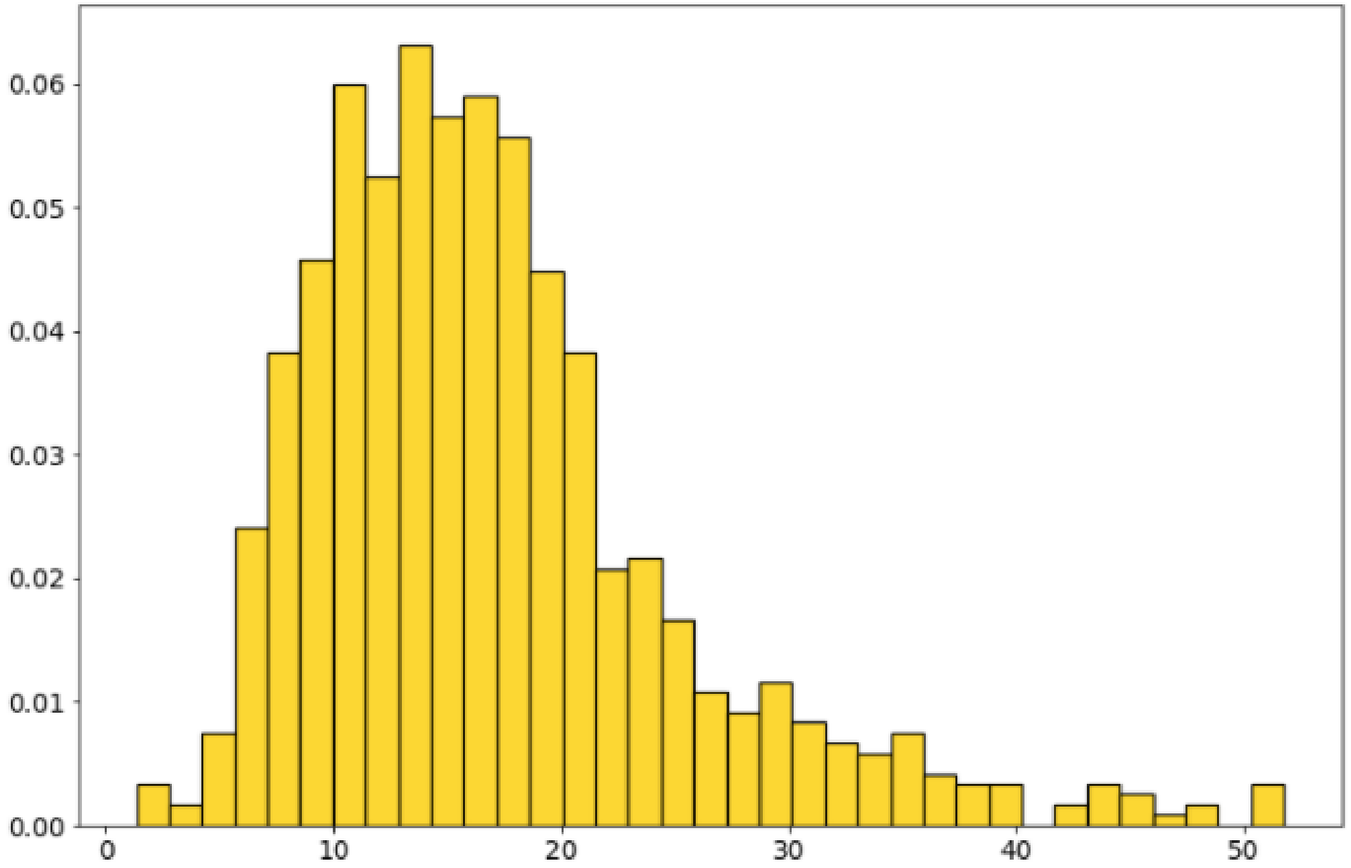
of all orders are from the top 10 merchants

Merchant Insights

Top Categories and Merchants

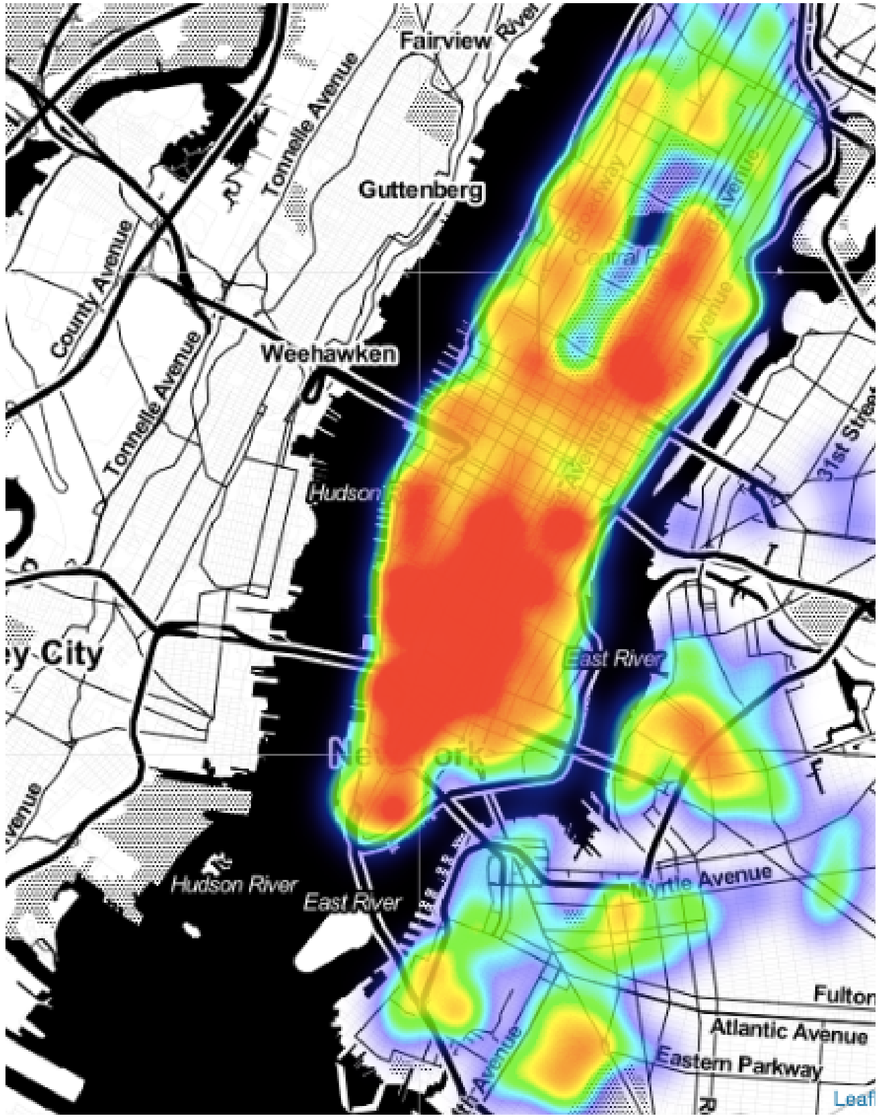
The average prep time is ~18 minutes; however, most restaurants take between 10-22 minutes

Histogram of Prep Time

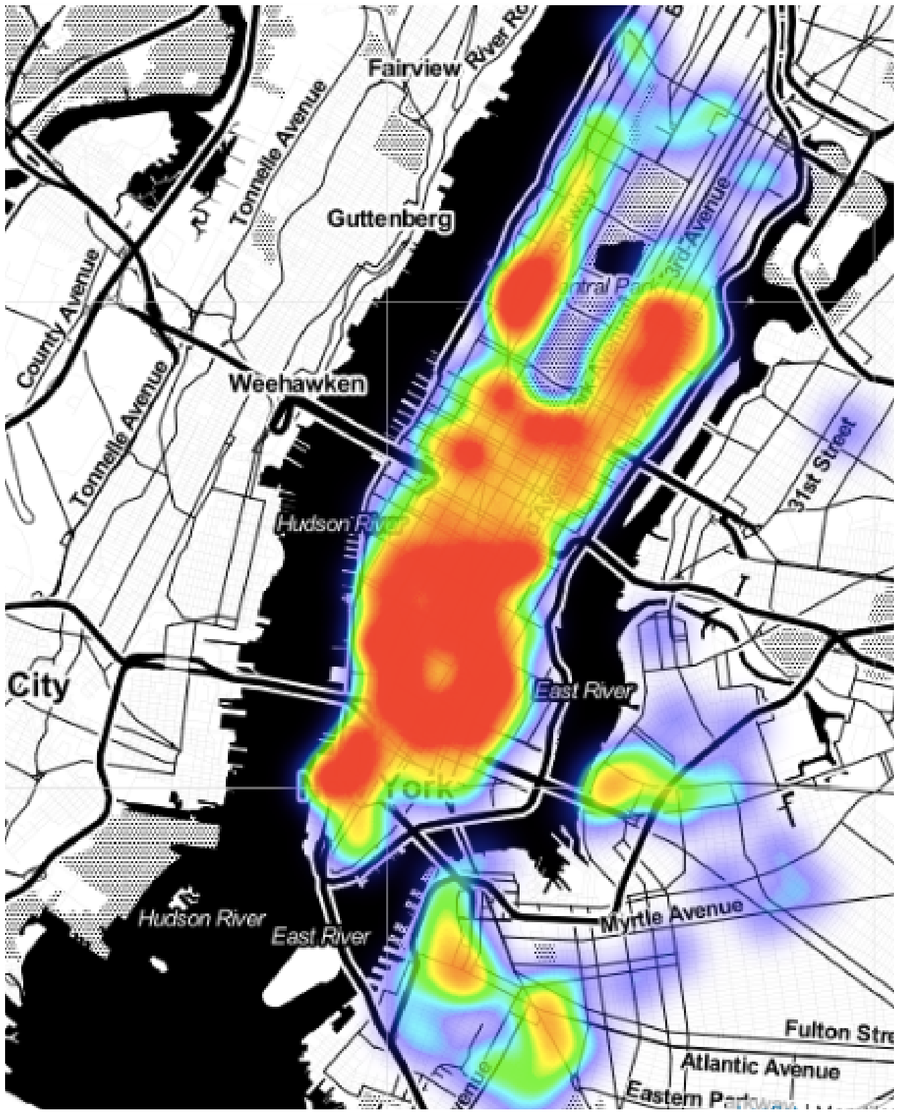


Geographical Distribution

Drop-off locations



Pickup locations

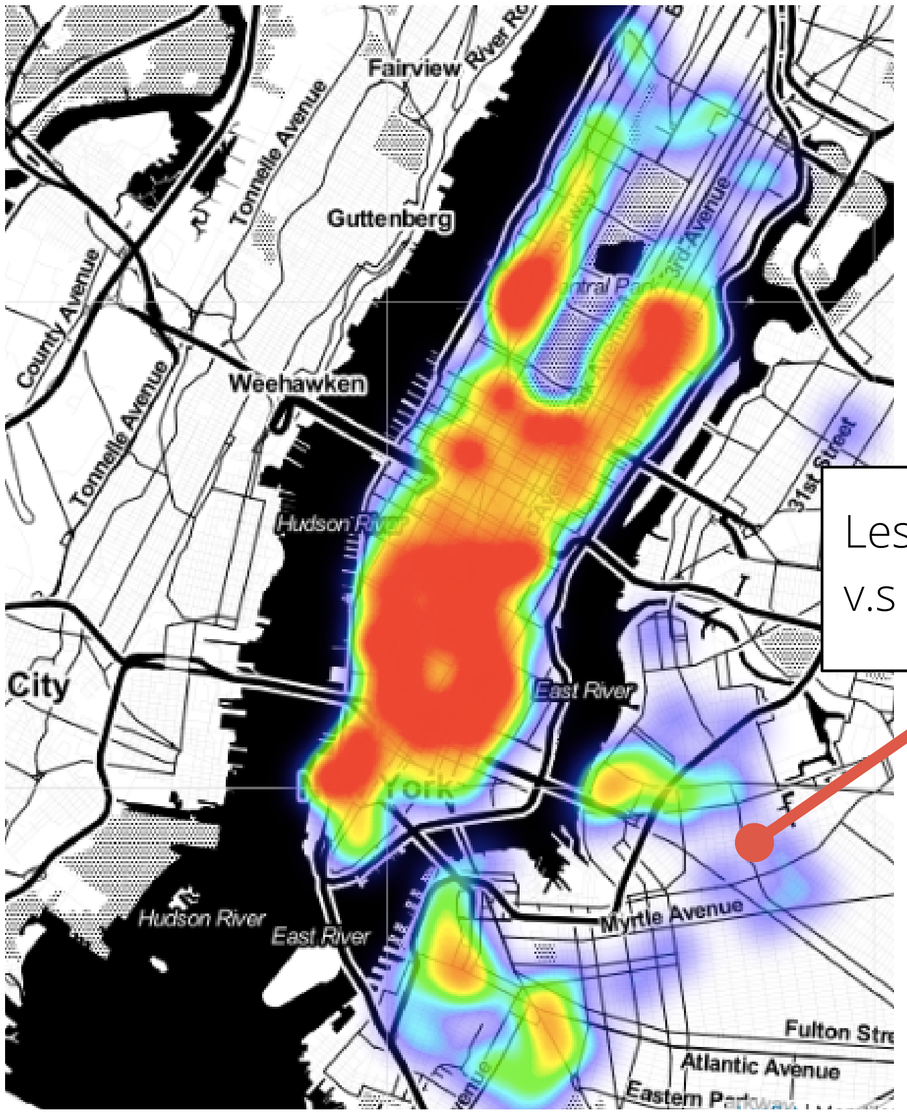


Geographical Distribution

Drop-off locations

Drop-off locations are more spread out in Upper Manhattan and Brooklyn/Queens

Pickup locations



Less merchants v.s customers

Geographical Distribution

Borough Analysis

Number of Deliveries

Brooklyn	205
Manhattan	5004
Queens	5

Number of Customers

Brooklyn	143
Manhattan	3045
Queens	4

Number of Merchants

Brooklyn	64
Manhattan	833
Queens	1

Customers ordering more than once:

Brooklyn	28%
Manhattan	31%

Geographical Distribution

Delivery Analysis by Borough

<1% of all orders delivered to Manhattan are from Brooklyn

~20% of orders delivered to Brooklyn are from Manhattan, and can take 20 minutes longer to arrive

Queens has the longest delivery times due to having only 1 merchant location in Queens

Drop-off Location	Pickup Location	Avg Prep Time	Avg Transit Time	Avg Delivery Time
Brooklyn	Brooklyn	33	12	45
	Manhattan	38	27	66
Manhattan	Brooklyn	43	35	86
	Manhattan	31	14	45
Queens	Brooklyn	45	20	65
	Manhattan	29	42	71
	Queens	28	8	36

How can we grow the market by 20% in 2 months?

Growth Strategy

In October, there were 5,214 orders placed through the delivery platform.
To grow by 20% in the next two months, Jumpman23 needs to gain 1043 orders or 522 orders per month for the next two months. In other words, Jumpman23 must meet the monthly target of 5,736 orders for November and December.

Jumpman23 should focus on the following two strategies:

1 - Customer Acquisition
2 - Customer Retention

1 - Customer Acquisition

Existing Customers

Before calculating how many new customers Jumpman23 needs to acquire, we must first determine how many orders we can expect from the existing customer base. To do this, the customers will be divided into two classes:

Class 1- Customers who have only ordered once. We are going to make an assumption that 30% of these customers will order again next month based off the pattern seen in October.

Class 2+ - Customer who have ordered more than once. On average, customers from Class 2+ order 3.1 times per month, which we will assume will remain consistent next month.

Class	Number of customers	Expected orders next month
A	2216	665
B	976	3026

1 - Customer Acquisition

New Customers

We have determined that we can expect 3,691 orders (Class 1 + Class 2+) from our existing customer base. To meet our goal, we still need 2045 orders, which we can get by acquiring new customers.

According to the data from October, Jumpman23 acquired an average of 106.4 new customers per day (slide 8). At this rate, we can expect 3,192 new customers to use the platform next month. Assuming a similar 30/70 ratio of Class 1 and Class 2+ type customers, we can expect that 2,234 customers (from Class 1) will order only once, accounting for 2,234 orders, and 958 customers (from Class 2+) will order an average of 3.1 times, accounting for 2,969 orders.

Summing the order contributions from new customers and existing customers, we get 8,894 total orders, which is much greater than our goal. Acquiring customers can be expensive, so in order to meet the minimum target of 5,736 total orders, Jumpan23 only needs to acquire **41.8 new customers per day** or 1,254 customers per month.

2 - Customer Retention

Class 1 vs Class 2+

Acquiring new customers can become very expensive over time, and it may be more cost-effective to retain existing customers. According to the data, only 30% of customers order more than once, which is very low!

What factors lead to customers ordering more than once? When comparing Class 1 and Class 2+ customers directly, the data did not show any significant differences.

- The **average delivery time** for Class 1 (ordered once) and Class 2+ customers (ordered more than once) was 45.6 and 45 mins respectively.
- The **average transit time** for both classes was ~ 14 mins (i.e food temperature was not an affect)
- Percentage of customers ordering more than once were similar for Manhattan and Brooklyn
- The most common **vehicle type** for jumpman in both classes was bikes, followed by cars and walking

2 - Customer Retention

Class 1 vs Class 3+

What if we compared Class 1 customers to customers who ordered at least 3 times or 4 times (i.,e Class 3 and Class 4+) ? Customers from these classes are ordering roughly once a week, which suggests that income may be an important factor.

The next slide illustrates the geographical locations of customers who ordered multiple times, and compares their location to a map of New York City depicting median household income by neighbourhood.

It appears that customers who order almost once a week are clustered in the "blue" zones (where there is a high mean household income).

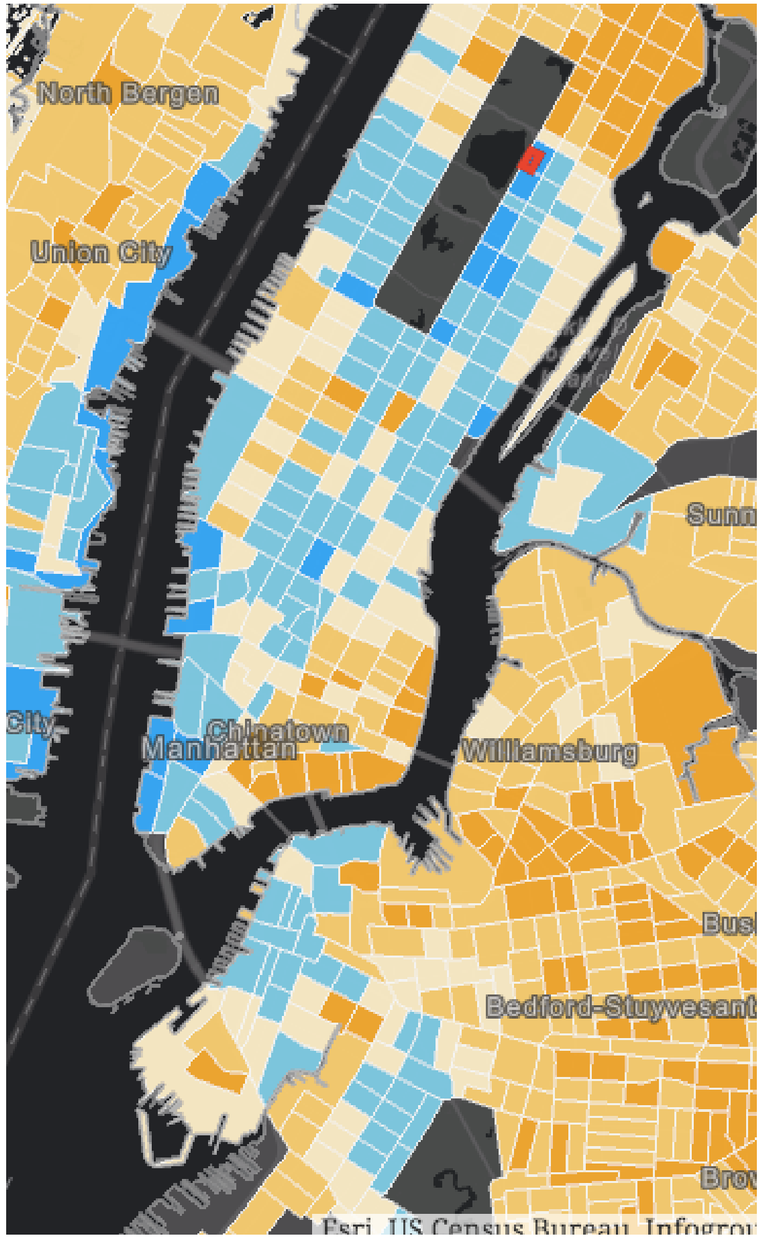
Median Household Income

Less than $35,000
$35,000 - $75,000
$70,000 - $100,000
$100,000 - $150,000
More than $150,000

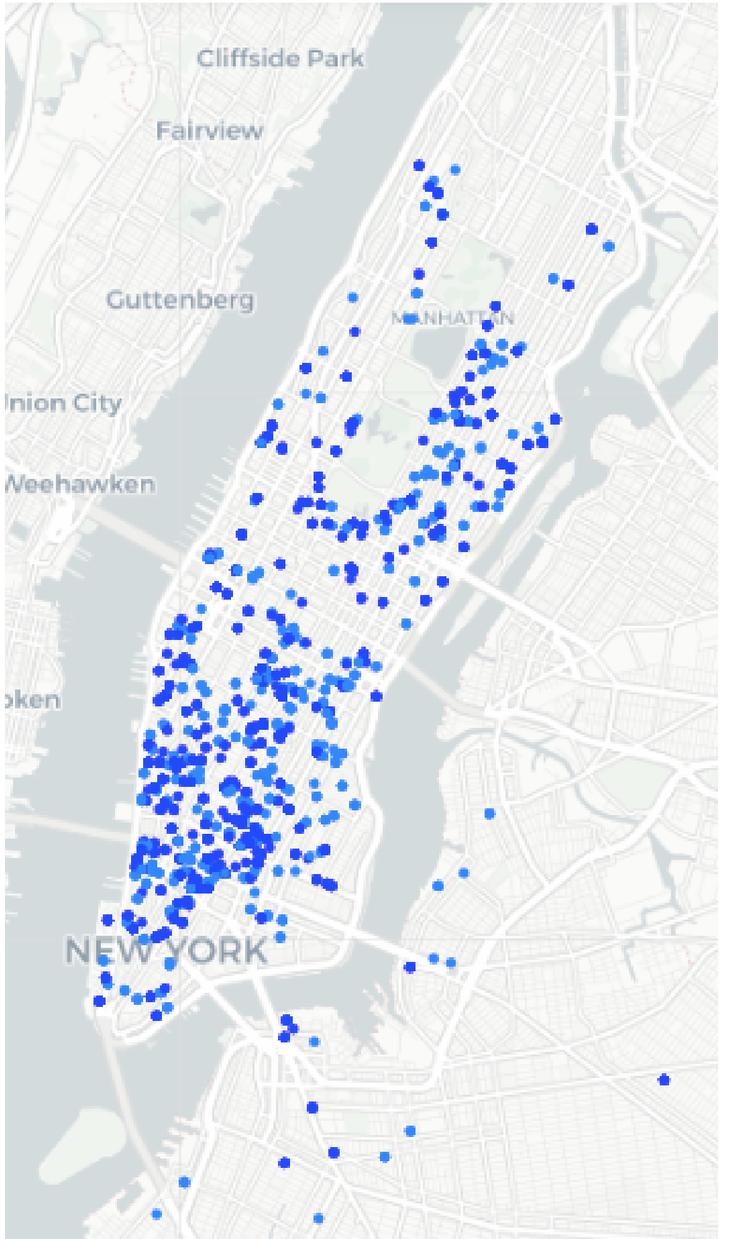
Ordered 3 times

Ordered more than 4 times

NYC household income



Location of customers who order frequently



2 - Customer Retention

Class 3+ Customer Insights

Do customers who order 3 or 4 times a month order differently than customers who order only once?
The top 10 merchants for both classes are shown along with the price range pulled from Yelp.

Class 1

1 - Shake Shack - $$
2 - Momofuku Milk Bar - $$
3 - The Meatball Shop - $$
4 - sweetgreen - $$
5 - RedFarm Broadway -$$$
6 - Blue Ribbon Fried Chicken - $$
7 - Chipotle Mexican Grill - $
8 - Whole Foods Market - $$$
9 - Momofuku Noodle Bar - $$
10 - Blue Ribbon Sushi - $$

Class 3+

1 - Shake Shack - $$
2 - sweetgreen - $$
3 - Blue Ribbon Fried Chicken - $$
4 - Blue Ribbon Sushi - $$
5 - The Meatball Shop - $$
6 - Whole Foods Market - $$$
7 - McDonald's - $
8 - Hu Kitchen - $$
9 - Parm - $$
10 - Mighty Quinn's BBQ - $$

Price range does not appear to differ between the two classes; however, we see more healthy food options, such sweetgreen and Whole Foods move up the list in Class 3+. Extravagant meal options like Suhi and BBQ also moved up in Class 3+.

Interestingly, we see that McDonald's is a popular option in Class 3+, which may suggest that busy individuals with lower budgets are also ordering frequently.

Growth Strategy

Next Steps

Customer Acquisition

- Run marketing campaigns to acquire ~42 new customers per day to meet target orders
- Good days to run campaigns are Sundays and Thursdays; best times are around noon and dinner
- Run promotional offers for popular restaurants like Shake Shack

Customer Retention

- Focus retention in areas with high median household income
- Partner with more merchants offering healthy food options (prices in '$$' range are most popular)

Improving Service Efficiency

- Order most popular meals ahead of time, for merchants in the top 10 list, to cut down on delivery time
- Partner with more merchants in locations where there are higher drop-offs v.s pickups (see heatmap)

Data Integrity

Data Integrity

There were a couple data quality issues in the data set which were accounted for during the analysis:

- Time violations - some rows showed pickup time stamps earlier than the order start time, and showed jumpman leaving the pickup location before the order start time

- NULL values - found in place_category, item_name, item_category_name, how_long_it_took_to_order, when_the jumpman_arrived_at_pickup, when_the_jumpman_left_pickup

- Customer orders split into multiple rows - if a customer ordered more than once item, than the order was split into multiple rows with the same delivery_id

APPENDIX

1 - Code - Performance Analysis

```
:
1  #total jumpman
2  print('total jumpmen: ', df['jumpman_id'].drop_duplicates().count())
3
4  #total merchants
5  print('total merchants: ', df['pickup_place'].drop_duplicates().count())
6
7  #total_customers
8  print('total customers: ', df['customer_id'].drop_duplicates().count())
9
```

```
1  #groub by viechle type
2  df_clean.groupby(['vehicle_type'])['delivery_id'].count().sort_values(ascending=False)
```

```
vehicle_type
bicycle       3740
car           1050
walker         234
van             69
scooter         64
truck           38
motorcycle      19
Name: delivery_id, dtype: int64
```

1 - Code - Performance Analysis

```
#Top 10 Categories
df_clean.groupby(['place_category'])['delivery_id'].count().sort_values(ascending=False).head(10)

# Top 10 Merchants
df_clean.groupby(['pickup_place'])['delivery_id'].count().sort_values(ascending=False).head(10)

```

```
#number of new customers per day
df = df_clean.copy()
df.sort_values('when_the_delivery_started',inplace=True)
df.drop_duplicates(subset = ['customer_id'], keep = 'first', inplace = True)
new_customers_day = (df.groupby(pd.Grouper(key='when_the_delivery_started', freq='D'))
                     ['customer_id'].count()
                     .to_frame(name = 'new_customers')
                     .reset_index()
                    )

#MEAN new customers per day
mean_new_customers_day = new_customers_day['new_customers'].mean()


#plot new customers per day in October
plt.figure(figsize=(12,8))
plt.plot(new_customers_day['when_the_delivery_started'], new_customers_day['new_customers'])

```

1 - Code - Performace Analysis

```
#Add customer order frequency to the dataframe
#groupby customer_id to get order counts data
customer_counts = df.groupby(['customer_id'])['delivery_id'].count().to_frame(name = 'order_counts').reset_index()

#create a new df which contains the original df and the customer order counts data
result = pd.merge(customer_counts, df, on="customer_id", how='left')

```

```
#Plot the location of customers who ordered 3 or more times
#using Folium

map_class = folium.Map(location=[40.744607, -73.990742],tiles='cartodbpositron', zoom_start=12)

for index, row in result.iterrows():

        if row['order_counts']==3:
        folium.CircleMarker([row['dropoff_lat'], row['dropoff_lon']],
                        radius=1,
                        fill_color="3db7e4", # divvy color
                       ).add_to(map_class)

        if row['order_counts']>3:
        folium.CircleMarker([row['dropoff_lat'], row['dropoff_lon']],
                        radius=1,
                        color="blue", # divvy color
                       ).add_to(map_class)
```

2 - Code - Analysis by borough

```
#read the boroughs data pulled from NYC open data
boroughs = gpd.read_file('boroughs.geojson')
boroughs = boroughs.drop([boroughs.index[1]]) #removes Staten Island row

#create geo points from lat and lon info for BOTH pickup and dropoff locations
gdf_pickups = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.pickup_lon, df.pickup_lat))
df['geometry_pickup'] = df['geometry']
gdf_dropoffs = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.dropoff_lon, df.dropoff_lat))
df['geometry_dropoff'] = df['geometry']


# Function that seperates data by BOROUGH
def bourough_check(series,label):
    global boroughs
    if series[label].within(boroughs.loc[boroughs['boro_name']=="Bronx",'geometry'].iloc[0]):
        return "Bronx"
    elif series[label].within(boroughs.loc[boroughs['boro_name']=="Manhattan",'geometry'].iloc[0]):
        return "Manhattan"
    elif series[label].within(boroughs.loc[boroughs['boro_name']=="Brooklyn",'geometry'].iloc[0]):
        return "Brooklyn"
    elif series[label].within(boroughs.loc[boroughs['boro_name']=="Queens",'geometry'].iloc[0]):
        return "Queens"
    else:
        return "Undefined"


#create a column in the df that contains the borough name of the dropoff and pickup location
df['borough_pickup']=df.apply(lambda x: bourough_check(x,'geometry_pickup'),axis=1)
df['borough_dropoff']=df.apply(lambda x: bourough_check(x,'geometry_dropoff'),axis=1)
```

2 - Code - Analysis by borough

```
1  ##### Analysis by borough ##########
2
3  #total orders by borough
4  df.groupby(['borough_dropoff'])['delivery_id'].count()
```

```
borough_dropoff
Brooklyn       205
Manhattan     5004
Queens           5
Name: delivery_id, dtype: int64
```

```
1  df.groupby(['borough_dropoff', 'borough_pickup'])['delivery_id'].count()
```

```
borough_dropoff  borough_pickup
Brooklyn         Brooklyn          166
                 Manhattan          39
Manhattan        Brooklyn           19
                 Manhattan        4985
Queens           Brooklyn            1
                 Manhattan           3
                 Queens              1
Name: delivery_id, dtype: int64
```

3 - Code - Heatmap of pickup and drop-off locations

```
#Heatmap of pickup locations
map_1 = folium.Map(location=[40.744607, -73.990742],tiles='stamentoner', zoom_start=12)


# convert to (n, 2) nd-array format for heatmap
merchants = df[['pickup_lat', 'pickup_lon']].as_matrix()

# plot heatmap
map_1.add_children(plugins.HeatMap(merchants, radius=15))

map_1
```

```
#Heatmap of dropoff locations
map_2 = folium.Map(location=[40.744607, -73.990742],tiles='stamentoner',zoom_start=12)

# convert to (n, 2) nd-array format for heatmap
customers = df[['dropoff_lat', 'dropoff_lon']].as_matrix()

# plot heatmap
map_2.add_children(plugins.HeatMap(customers, radius=15))

map_2
```