Food delivery service, Doordash Exploratory data anlaysis

January 11, 2023

[1]:
```
#Basic libraries
import pandas as pd
pd.set_option('display.max_columns', 100) #Display all columns
import numpy as np

#Visualization libraries
import matplotlib.pyplot as plt
from matplotlib.ticker import FormatStrFormatter
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
from plotly import tools
from plotly.offline import iplot

#Geospatial Analysis Libraries
import geopandas as gpd
import math
import folium
from folium import Choropleth, Circle, Marker
from folium.plugins import HeatMap, MarkerCluster

#NLTK libraries
import nltk
import re
import string
from wordcloud import WordCloud,STOPWORDS
from nltk.corpus import stopwords
from textblob import TextBlob


#Miscellaneous libraries
from collections import defaultdict
import cufflinks as cf
cf.go_offline()
cf.set_config_file(offline=False, world_readable=True)
```

1 Understanding the data

1.0.1 Start : By peeking into using pandas inbuilt functions

[2]:
```
#Reading the data
delivery=pd.read_csv('onlinedeliverydata.csv')

#Printing the information of dataset
print ("The shape of the  data is (row, column):"+ str(delivery.shape)) # shape␣
 ↪shows dimension
print(delivery.info()) #finding types and counting if there are missing values
```

```
The shape of the  data is (row, column):(388, 55)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 55 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Age                                   388 non-null    int64
 1   Gender                                388 non-null    object
 2   Marital Status                        388 non-null    object
 3   Occupation                            388 non-null    object
 4   Monthly Income                        388 non-null    object
 5   Educational Qualifications            388 non-null    object
 6   Family size                           388 non-null    int64
 7   latitude                              388 non-null    float64
 8   longitude                             388 non-null    float64
 9   Pin code                              388 non-null    int64
 10  Medium (P1)                           388 non-null    object
 11  Medium (P2)                           388 non-null    object
 12  Meal(P1)                              388 non-null    object
 13  Meal(P2)                              388 non-null    object
 14  Perference(P1)                        388 non-null    object
 15  Perference(P2)                        388 non-null    object
 16  Ease and convenient                   388 non-null    object
 17  Time saving                           388 non-null    object
 18  More restaurant choices               388 non-null    object
 19  Easy Payment option                   388 non-null    object
 20  More Offers and Discount              388 non-null    object
 21  Good Food quality                     388 non-null    object
 22  Good Tracking system                  388 non-null    object
 23  Self Cooking                          388 non-null    object
 24  Health Concern                        388 non-null    object
 25  Late Delivery                         388 non-null    object
 26  Poor Hygiene                          388 non-null    object
 27  Bad past experience                   388 non-null    object
 28  Unavailability                        388 non-null    object
 29  Unaffordable                          388 non-null    object
```

```
 30  Long delivery time                            388 non-null    object
 31  Delay of delivery person getting assigned     388 non-null    object
 32  Delay of delivery person picking up food      388 non-null    object
 33  Wrong order delivered                         388 non-null    object
 34  Missing item                                  388 non-null    object
 35  Order placed by mistake                       388 non-null    object
 36  Influence of time                             388 non-null    object
 37  Order Time                                    388 non-null    object
 38  Maximum wait time                             388 non-null    object
 39  Residence in busy location                    388 non-null    object
 40  Google Maps Accuracy                          388 non-null    object
 41  Good Road Condition                           388 non-null    object
 42  Low quantity low time                         388 non-null    object
 43  Delivery person ability                       388 non-null    object
 44  Influence of rating                           388 non-null    object
 45  Less Delivery time                            388 non-null    object
 46  High Quality of package                       388 non-null    object
 47  Number of calls                               388 non-null    object
 48  Politeness                                    388 non-null    object
 49  Freshness                                     388 non-null    object
 50  Temperature                                   388 non-null    object
 51  Good Taste                                    388 non-null    object
 52  Good Quantity                                 388 non-null    object
 53  Output                                        388 non-null    object
 54  Reviews                                       388 non-null    object
dtypes: float64(2), int64(3), object(50)
memory usage: 166.8+ KB
None
```

1.1 Content of dataset

1.1.1 1. Demographics of consumers

1.1.2 2. Overall/general purchase decision

1.1.3 3. Time of delivery influencing the purchase decision

1.1.4 4. Rating of Restaurant influencing the purchase decision

1.2 Data-peak Head

[3]:
```
delivery.head()
```

[3]:

	Age	Gender	Marital Status	Occupation	Monthly Income	\
0	20	Female	Single	Student	No Income	
1	24	Female	Single	Student	Below Rs.10000	
2	22	Male	Single	Student	Below Rs.10000	
3	22	Female	Single	Student	No Income	
4	22	Male	Single	Student	Below Rs.10000	

```
  Educational Qualifications  Family size  latitude  longitude  Pin code  \
0              Post Graduate            4   12.9766    77.5993    560001
1                   Graduate            3   12.9770    77.5773    560009
2              Post Graduate            3   12.9551    77.6593    560017
3                   Graduate            6   12.9473    77.5616    560019
4              Post Graduate            4   12.9850    77.5533    560010

          Medium (P1)   Medium (P2)   Meal(P1) Meal(P2)  \
0  Food delivery apps   Web browser  Breakfast    Lunch
1  Food delivery apps   Web browser     Snacks   Dinner
2  Food delivery apps   Direct call      Lunch   Snacks
3  Food delivery apps       Walk-in     Snacks   Dinner
4             Walk-in   Direct call      Lunch   Dinner

                           Perference(P1)  \
0          Non Veg foods (Lunch / Dinner)
1          Non Veg foods (Lunch / Dinner)
2          Non Veg foods (Lunch / Dinner)
3  Veg foods (Breakfast / Lunch / Dinner)
4          Non Veg foods (Lunch / Dinner)

                           Perference(P2) Ease and convenient  \
0                    Bakery items (snacks)             Neutral
1   Veg foods (Breakfast / Lunch / Dinner)      Strongly agree
2                  Ice cream / Cool drinks      Strongly agree
3                    Bakery items (snacks)               Agree
4   Veg foods (Breakfast / Lunch / Dinner)               Agree

      Time saving More restaurant choices Easy Payment option  \
0         Neutral                 Neutral             Neutral
1  Strongly agree          Strongly agree      Strongly agree
2  Strongly agree          Strongly agree             Neutral
3           Agree          Strongly agree               Agree
4           Agree                   Agree               Agree

  More Offers and Discount Good Food quality Good Tracking system  \
0                  Neutral           Neutral              Neutral
1           Strongly agree           Neutral                Agree
2                  Neutral          Disagree              Neutral
3           Strongly agree             Agree                Agree
4                    Agree           Neutral              Neutral

     Self Cooking  Health Concern   Late Delivery    Poor Hygiene  \
0         Neutral         Neutral         Neutral         Neutral
1  Strongly agree  Strongly agree           Agree  Strongly agree
2        Disagree         Neutral         Neutral           Agree
3           Agree  Strongly agree         Neutral           Agree
```

```
4           Agree  Strongly agree  Strongly agree           Agree

  Bad past experience  Unavailability    Unaffordable Long delivery time  \
0             Neutral         Neutral         Neutral              Agree
1      Strongly agree  Strongly agree  Strongly agree     Strongly agree
2               Agree           Agree           Agree              Agree
3            Disagree        Disagree         Neutral              Agree
4      Strongly agree           Agree        Disagree     Strongly agree

  Delay of delivery person getting assigned  \
0                                     Agree
1                           Strongly agree
2                                     Agree
3                                     Agree
4                           Strongly agree

  Delay of delivery person picking up food Wrong order delivered  \
0                                   Agree                  Agree
1                           Strongly agree        Strongly agree
2                                   Agree        Strongly agree
3                                   Agree              Disagree
4                                 Neutral               Neutral

     Missing item Order placed by mistake Influence of time  \
0           Agree                   Agree               Yes
1  Strongly agree          Strongly agree               Yes
2           Agree                 Neutral               Yes
3        Disagree                 Neutral               Yes
4         Neutral                Disagree               Yes

            Order Time Maximum wait time Residence in busy location  \
0  Weekend (Sat & Sun)        30 minutes                      Agree
1    Anytime (Mon-Sun)        30 minutes             Strongly Agree
2    Anytime (Mon-Sun)        45 minutes                      Agree
3    Anytime (Mon-Sun)        30 minutes                   Disagree
4  Weekend (Sat & Sun)        30 minutes                      Agree

  Google Maps Accuracy Good Road Condition Low quantity low time  \
0              Neutral             Neutral               Neutral
1              Neutral            Disagree     Strongly disagree
2       Strongly Agree             Neutral               Neutral
3                Agree               Agree               Neutral
4                Agree               Agree                 Agree

  Delivery person ability Influence of rating    Less Delivery time  \
0                 Neutral                 Yes  Moderately Important
1                   Agree                 Yes        Very Important
```

```
2                   Agree                 Yes             Important
3                   Agree                 Yes        Very Important
4                   Agree                 Yes             Important

  High Quality of package       Number of calls            Politeness  \
0    Moderately Important  Moderately Important  Moderately Important
1          Very Important        Very Important        Very Important
2          Very Important  Moderately Important        Very Important
3               Important  Moderately Important        Very Important
4               Important  Moderately Important             Important

             Freshness            Temperature           Good Taste   \
0  Moderately Important  Moderately Important  Moderately Important
1        Very Important        Very Important        Very Important
2        Very Important             Important        Very Important
3        Very Important        Very Important        Very Important
4             Important             Important        Very Important

          Good Quantity Output  \
0  Moderately Important    Yes
1        Very Important    Yes
2  Moderately Important    Yes
3             Important    Yes
4        Very Important    Yes

                                              Reviews
0                                             Nil\n
1                                               Nil
2  Many a times payment gateways are an issue, so…
3                                               nil
4                                               NIL
```

1.3 Insights of data

- Three more categorical variables and very less continuous variables
- Longitude and latitude variable
- Most of the variables has likert scale classes inside the variable , it needs to be encoded

[4]:
```
delivery.describe()
```

[4]:
```
                Age  Family size    latitude   longitude       Pin code
count   388.000000   388.000000  388.000000  388.000000     388.000000
mean     24.628866     3.280928   12.972058   77.600160  560040.113402
std       2.975593     1.351025    0.044489    0.051354      31.399609
min      18.000000     1.000000   12.865200   77.484200  560001.000000
25%      23.000000     2.000000   12.936900   77.565275  560010.750000
50%      24.000000     3.000000   12.977000   77.592100  560033.500000
```

```
75%      26.000000      4.000000    12.997025    77.630900   560068.000000
max      33.000000      6.000000    13.102000    77.758200   560109.000000
```

1.4 Insights of .describe()

- 388 respondents in dataset
- Age between 18-33 and average of 25
- Family size between 1-6 and average of 3
- Latitude, longitude, pincode should not be considered for the numerical variable as they do not provide much values from such measures.

1.5 Pivot table analysis with Pandas

Demographic Anaylsis 1

[5]:
```
#Pivot table
delivery_pivot1=pd.pivot_table(delivery,index=["Gender","Marital Status"],
                               values=['Age','Family size'],
                               aggfunc=[np.mean,len], margins=True)

#Adding color gradient
cm = sns.light_palette("gray", as_cmap=True)
delivery_pivot1.style.background_gradient(cmap=cm)
```

[5]:
```
<pandas.io.formats.style.Styler at 0x7fcfd10fc460>
```

Insights from Demographis Analysis 1

- Average age of female, male, and single is around 23, AND 27 and 28 for married female and male
- Average family size is 4 for female respondent, 3 for male respondent.
- More male respondents than female respondents.

Demographic Anaylsis 2

[6]:
```
#Pivot table
delivery_pivot2=pd.pivot_table(delivery,index=["Educational␣
 ↪Qualifications","Occupation"],
                               values=['Age','Family size'],
                               aggfunc=[np.mean,len,np.std])

#Adding bar for numbers
delivery_pivot2.style.bar(color='yellow')
```

[6]:
```
<pandas.io.formats.style.Styler at 0x7fcfb205e160>
```

Insights from Demographis Analysis 2

- Female respondents who has only till school education or uneducated are mostly house wives on average and they are not very young.
- Most of them are self-educated if they are uneducated.
- Graduates and Phd's are most employeed and most of graduates and Phd's are still students.
- Higher deviation i n cas of school educated as the uncertainties are high about their career.
- Phd's who are employed also have higher deviation.

Demographic Anaylsis 3

[7]:
```
#Pivot table
delivery_pivot3=pd.pivot_table(delivery,index=["Occupation","Monthly Income"],
                               values=['Age','Family size'],
                               aggfunc=[np.mean,len,np.std])
```

[8]:
```
#Adding style
delivery_pivot3.style\
    .format('{:.2f}')\
    .bar(align='mid',color='lightpink')\
    .set_properties(padding='5px',border='3px solid white',width='200px')
```

[8]:
```
<pandas.io.formats.style.Styler at 0x7fcfb213b340>
```

Insights from Demographis Analysis 2

- Age increases then the wage increases along with the family size
- Employee who received below Rs.10000 have higher deviation in age and family size
- There are nearly 178 students who has no income responded to this data.
- Next highest responses were from employees who's salaryis between Rs.25000-Rs.50000.

Time factor analysis 1

[9]:
```
#Pivot table
delivery_pivot4=pd.pivot_table(delivery,index=["Order Time","Maximum wait␣
 ↪time"],
                               values=['Age','Family size'],columns=['Influence␣
 ↪of time'],
                               aggfunc={'Influence of time':len},
                               fill_value=0)

#Adding color gradient
cm = sns.light_palette("lightblue", as_cmap=True)
delivery_pivot4.style.background_gradient(cmap=cm)
```

[9]:
```
<pandas.io.formats.style.Styler at 0x7fcfb20a0130>
```

Insights from time factor analysis 1

- No influences in time of delivery when people can wait for 60 or more than 60 minutes for an order.

- High influence of time of delivery where the buyer can wait only for 30-45 minutes.
- Not much purchasers who buys during weekdays only.
- During waiting purchasers can wait a max 45 minutes.

1.5.1 Time factor Anlaysis 2

[10]:
```
#Pivot table
delivery_pivot4=pd.pivot_table(delivery,index=["Medium (P1)","Perference(P1)"],
                               columns=['Influence of time'],
                               aggfunc={'Influence of time':len},
                               fill_value=0)
#Adding color gradient
delivery_pivot4.style\
    .format('{:.2f}')\
    .bar(align='mid',color='lightblue')\
    .set_properties(padding='5px',border='3px solid white',width='200px')
```

[10]:
```
<pandas.io.formats.style.Styler at 0x7fcfb2143160>
```

Insights from time factor analysis 2

- 224 people who order Non veg foods on apps are influenced by time.
- # of Non veg foods preference purchasers > # of Veg foods preference purchasers
- Not much data for other types of 'Medium' and influence of time of delivery

1.6 Explatory data analysis

1.6.1 Univariate Analysis

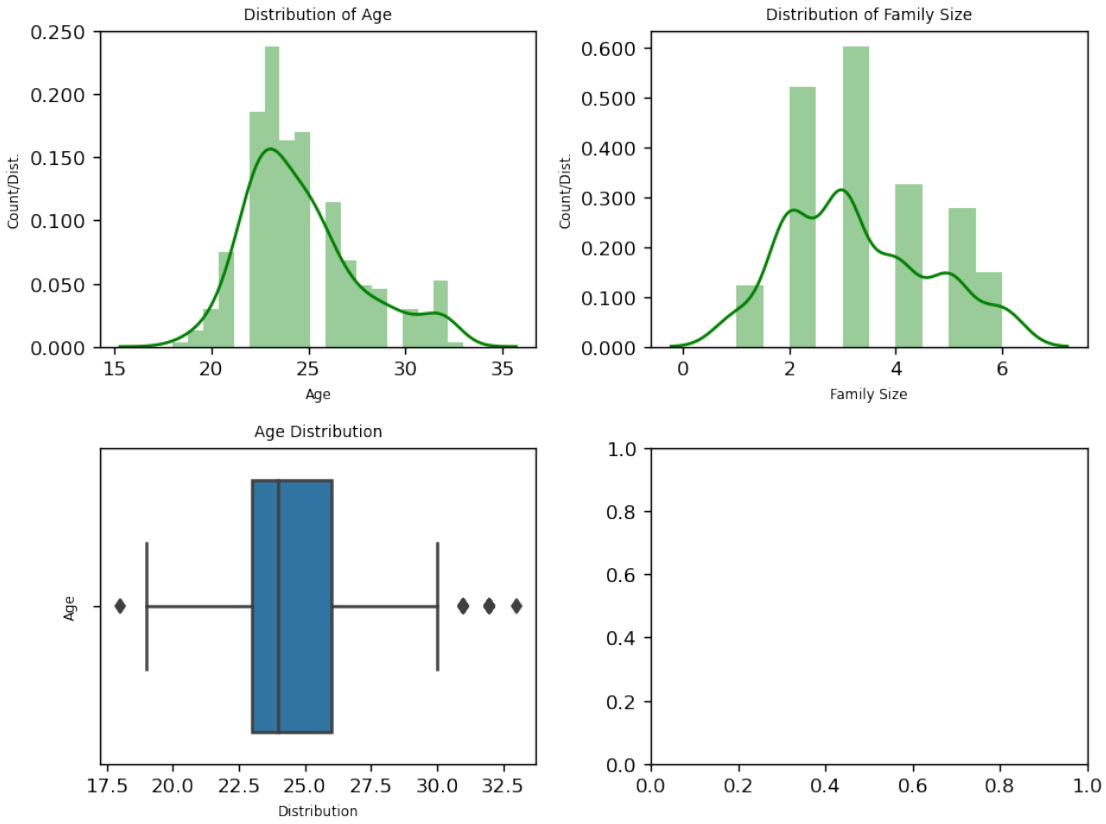
[11]:
```
#Setting up the frame
fig,axes = plt.subplots(nrows=2,ncols=2,dpi=120,figsize = (8,6))

#Distribution of age with displot
plot00=sns.distplot(delivery['Age'],ax=axes[0][0],color='green')
axes[0][0].yaxis.set_major_formatter(FormatStrFormatter('%.3f'))
axes[0][0].set_title('Distribution of Age',fontdict={'fontsize':8})
axes[0][0].set_xlabel('Age',fontdict={'fontsize':7})
axes[0][0].set_ylabel('Count/Dist.',fontdict={'fontsize':7})
plt.tight_layout()

#Distribution of Familysize with displot
plot01=sns.distplot(delivery['Family size'],ax=axes[0][1],color='green')
axes[0][1].set_title('Distribution of Family Size',fontdict={'fontsize':8})
axes[0][1].set_xlabel('Family Size',fontdict={'fontsize':7})
axes[0][1].set_ylabel('Count/Dist.',fontdict={'fontsize':7})
axes[0][1].yaxis.set_major_formatter(FormatStrFormatter('%.3f'))
plt.tight_layout()
```

```
#Age-Boxplot
plot10=sns.boxplot(delivery['Age'],ax=axes[1][0])
axes[1][0].set_title('Age Distribution',fontdict={'fontsize':8})
axes[1][0].set_xlabel('Distribution',fontdict={'fontsize':7})
axes[1][0].set_ylabel(r'Age',fontdict={'fontsize':7})
plt.tight_layout()

#FamilySize-Boxplot
plot11=sns.boxplot(delivery[' Family size'],ax=axes[1][1])
axes[1][1].set_title(r'Numerical Summary (Family Size)',fontdict={'fontsize':8})
axes[1][1].set_ylabel(r'Size of Family',fontdict={'fontsize':7})
axes[1][1].set_xlabel('Family Size',fontdict={'fontsize':7})
plt.tight_layout()

plt.show()
```

```
/Users/hwangtaejin/.local/lib/python3.8/site-
packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

/Users/hwangtaejin/.local/lib/python3.8/site-
packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

/Users/hwangtaejin/.local/lib/python3.8/site-packages/seaborn/_decorators.py:36:
FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-packages/panda
 ↪core/indexes/base.py:3621, in Index.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
```

```
File ~/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-packages/panda /
 ↪_libs/index.pyx:136, in pandas._libs.index.IndexEngine.get_loc()

File ~/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-packages/panda /
 ↪_libs/index.pyx:163, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.
 ↪PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.
 ↪PyObjectHashTable.get_item()

KeyError: ' Family size'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [11], in <cell line: 28>()
     25 plt.tight_layout()
     27 #FamilySize-Boxplot
---> 28 plot11=sns.boxplot(delivery[' Family size'],ax=axes[1][1])
     29 axes[1][1].set_title(r'Numerical Summary (Family␣
 ↪Size)',fontdict={'fontsize':8})
     30 axes[1][1].set_ylabel(r'Size of Family',fontdict={'fontsize':7})

File ~/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-packages/panda /
 ↪core/frame.py:3505, in DataFrame.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-packages/panda /
 ↪core/indexes/base.py:3623, in Index.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: ' Family size'
```



Insights from Univariate Anaylsis

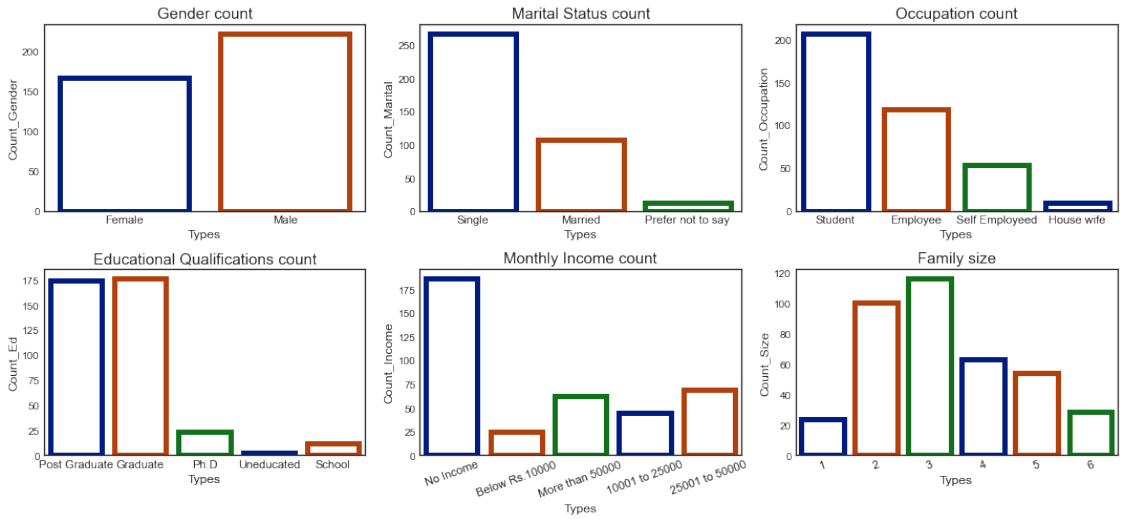
- Age and family size : Normally Distributed
- Age average : between 23 & 25 (some outliers exist)
- Average family size : around 3

1.6.2 Demographic Analysis

[12]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('seaborn-white')

#Gender Countplot
plt.subplot(2,3,1)
ax = sns.countplot(x="Gender", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Gender count',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
```

```
ax.set_ylabel('Count_Gender', fontsize = 12)
plt.tight_layout()

#Marital Status Countplot
plt.subplot(2,3,2)
ax = sns.countplot(x="Marital Status", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Marital Status count',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Marital', fontsize = 12)
plt.tight_layout()

#Occupation Countplot
plt.subplot(2,3,3)
ax = sns.countplot(x="Occupation", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Occupation count',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count_Occupation', fontsize = 12)
plt.tight_layout()

#Education Countplot
plt.subplot(2,3,4)
ax = sns.countplot(x="Educational Qualifications", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Educational Qualifications count',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Ed', fontsize = 12)
plt.tight_layout()

#Income Countplot
plt.subplot(2,3,5)
ax = sns.countplot(x="Monthly Income", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=20)
ax.set_title('Monthly Income count',fontsize = 15)
```

```
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Income', fontsize = 12)
plt.tight_layout()

#Family Size Countplot
plt.subplot(2,3,6)
ax = sns.countplot(x="Family size", data=delivery,
                   facecolor=(0, 0, 0, 0),
                   linewidth=5,
                   edgecolor=sns.color_palette("dark", 3))
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=20)
ax.set_title('Family size',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Size', fontsize = 12)
plt.tight_layout()
```



Insights from Demographic Anaylsis

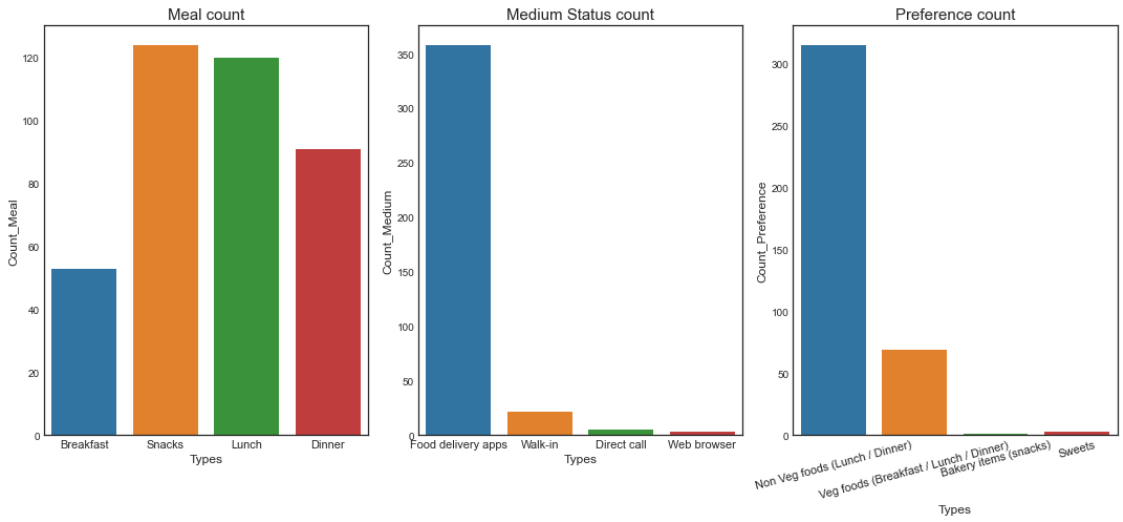
- Normal distribution exists for family distribution where most of the family size recorded were in the 3 people followed by 2.
- Assumption : Respondents are mostly bachelors staying in hostel or PG and generally are less likely to cook food in their room....?
- It can also be seen that nearly 57% of the respondents were male and 43%of the respondents were male and the data is almost balanced here.
- The education qualification of respondents where highly from the young graduates (177) followed by post graduates (174) and it also covers Ph.D. Scholars, School children and Uneducated people.
- Most of the respondents comes under the No income bracket which supports the age variable as most of them are students and they don't have regular income.
- The next highest income bracket is between Rs 25001 to 50000 : nominal salary for a fresher

in corporate industry.

- Most of the respondents are bachelors who are students and there are significantly lesser number from married respondents.

1.6.3 Consumer prefernences

[13]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('seaborn-white')

#Meal Countplot
plt.subplot(1,3,1)
ax = sns.countplot(x="Meal(P1)", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Meal count',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Meal', fontsize = 12)
plt.tight_layout()

#Medium Countplot
plt.subplot(1,3,2)
ax = sns.countplot(x="Medium (P1)", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11)
ax.set_title('Medium Status count',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count_Medium', fontsize = 12)
plt.tight_layout()

#Preference Countplot
plt.subplot(1,3,3)
ax = sns.countplot(x="Perference(P1)", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=15)
ax.set_title('Preference count',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count_Preference', fontsize = 12)
plt.tight_layout()
```



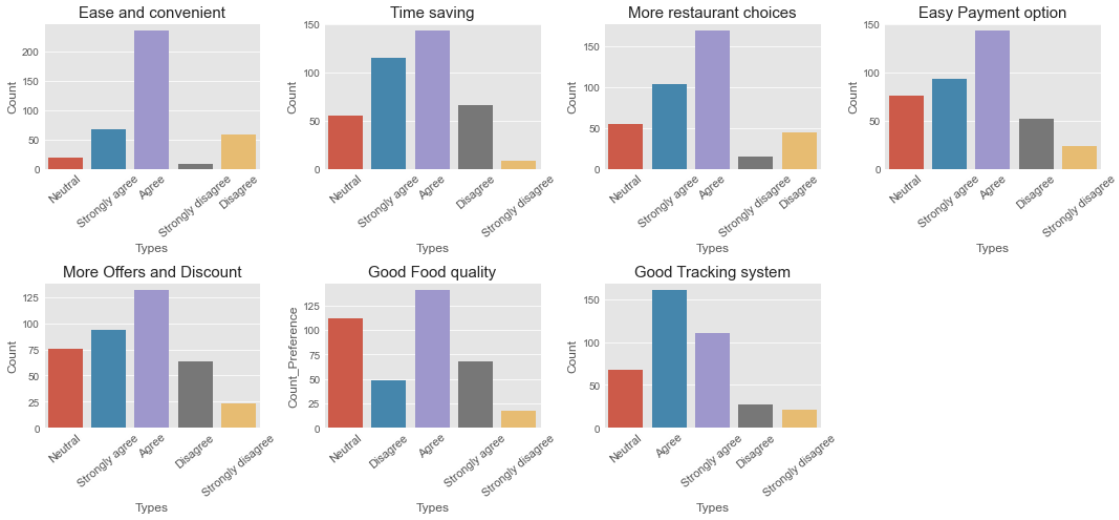
Insights from Consumer Preferences

- The data for consumer preferences on which meal of the day, type of meal and medium to purchase food can be seen.
- Most of the preferences belongs to food delivery applications when it comes to medium of purchase followed by Walking
- Nonveg is the most preferred type of meal followed by veg food and minimum numbers on sweets and bakery items.

1.6.4 Purchase Demand

[14]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('ggplot')

#Ease and convenient Countplot
plt.subplot(2,4,1)
ax = sns.countplot(x="Ease and convenient", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Ease and convenient',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Time Countplot
plt.subplot(2,4,2)
ax = sns.countplot(x="Time saving", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Time saving',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
```

```
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Restaurant Countplot
plt.subplot(2,4,3)
ax = sns.countplot(x="More restaurant choices", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('More restaurant choices',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Payment Countplot
plt.subplot(2,4,4)
ax = sns.countplot(x="Easy Payment option", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Easy Payment option',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Offers Countplot
plt.subplot(2,4,5)
ax = sns.countplot(x="More Offers and Discount", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('More Offers and Discount',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Preference Countplot
plt.subplot(2,4,6)
ax = sns.countplot(x="Good Food quality", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Good Food quality',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count_Preference', fontsize = 12)
plt.tight_layout()

#Tracking Countplot
plt.subplot(2,4,7)
ax = sns.countplot(x="Good Tracking system", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Good Tracking system',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()
```



Insights from Purchase Demand

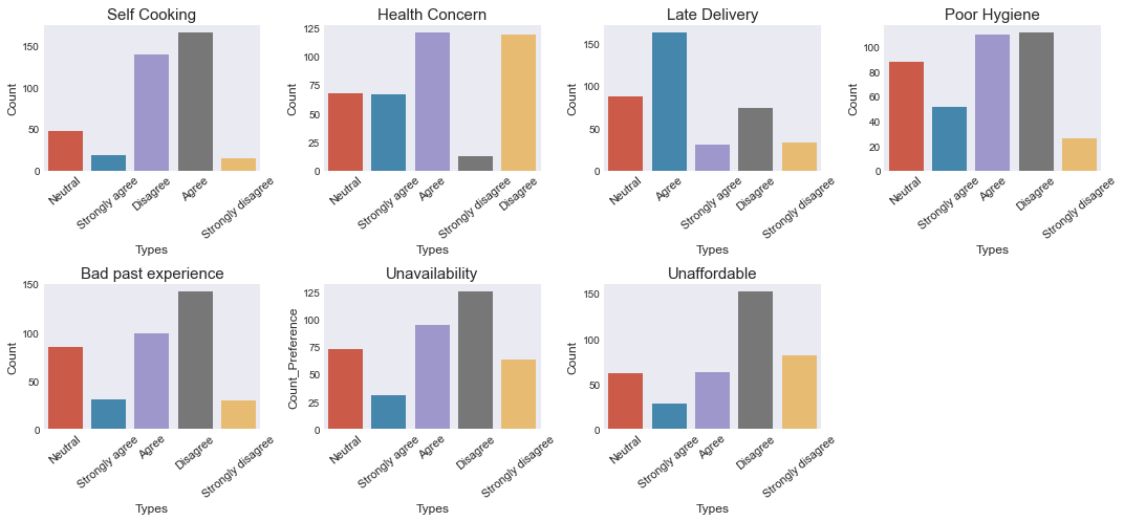
- Can check demand factors which pulls the consumer to purchase an order from application
- Ease and convenient while using an app for ordering are agreed much for influencing them to make an order.
- Time saving factor has most response as nearly 258 respondents agree that using application to order is time saving
- Easy Payment makes a significant contribution as 143 respondents agree that they consider this factor for the order demand

1.6.5 Restriction in Order Demand

[15]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('seaborn-dark')

#Self cooking Countplot
plt.subplot(2,4,1)
ax = sns.countplot(x="Self Cooking", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Self Cooking',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Health Countplot
plt.subplot(2,4,2)
ax = sns.countplot(x="Health Concern", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Health Concern',fontsize = 15)
```

```
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Late delivery Countplot
plt.subplot(2,4,3)
ax = sns.countplot(x="Late Delivery", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Late Delivery',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Hygiene Countplot
plt.subplot(2,4,4)
ax = sns.countplot(x="Poor Hygiene", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Poor Hygiene',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Past Countplot
plt.subplot(2,4,5)
ax = sns.countplot(x="Bad past experience", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Bad past experience',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Unavailability Countplot
plt.subplot(2,4,6)
ax = sns.countplot(x="Unavailability", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Unavailability',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count_Preference', fontsize = 12)
plt.tight_layout()

#Unaffordable Countplot
plt.subplot(2,4,7)
ax = sns.countplot(x="Unaffordable", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Unaffordable',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
```

```
plt.tight_layout()
```



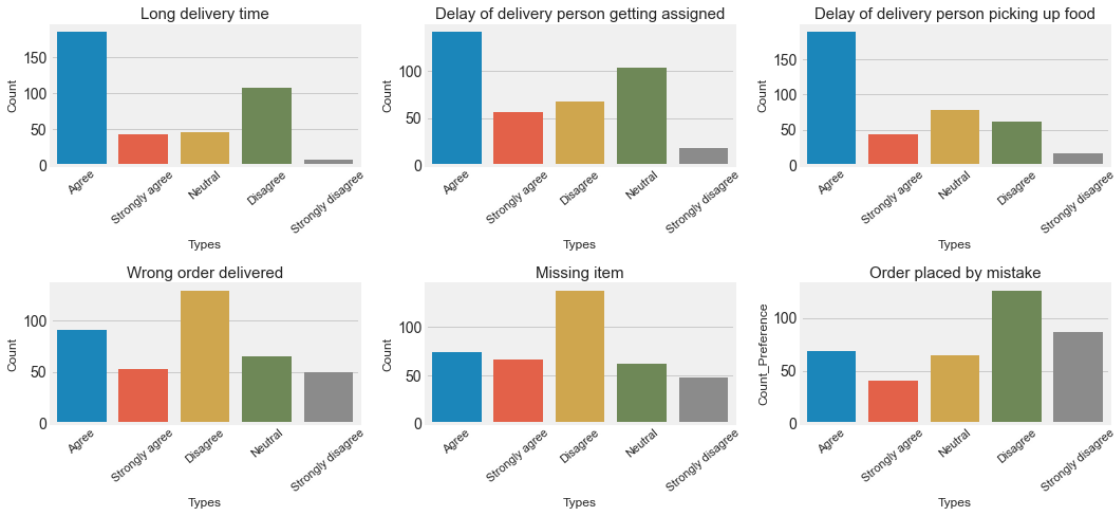
Insights from restriction on order demand

- Majority of the people agree that self-cooking makes them not to order food on applications, while 140 people disagree with that statement as well.
- Majority has agreed that health reason was one of the important criteria which restricts them from ordering
- In case of hygiene factor many disagreed that the food is poor: hygiene doesn't restrict people from ordering
- Majority have disagreed that past food ordering experience doesn't influence their current order demand
- Around 163 respondents have agreed that late delivery does influence their order decision

1.6.6 Influence in Cancellation order

[16]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('fivethirtyeight')


#Long delivery time Countplot
plt.subplot(2,3,1)
ax = sns.countplot(x="Long delivery time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Long delivery time',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()
```

```
#Delay of delivery person getting assigned Countplot
plt.subplot(2,3,2)
ax = sns.countplot(x="Delay of delivery person getting assigned", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Delay of delivery person getting assigned',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Delay of delivery person picking up food Countplot
plt.subplot(2,3,3)
ax = sns.countplot(x="Delay of delivery person picking up food", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Delay of delivery person picking up food',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Wrong order delivered Countplot
plt.subplot(2,3,4)
ax = sns.countplot(x="Wrong order delivered", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Wrong order delivered',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Missing item Countplot
plt.subplot(2,3,5)
ax = sns.countplot(x="Missing item", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Missing item',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Order placed by mistake Countplot
plt.subplot(2,3,6)
ax = sns.countplot(x="Order placed by mistake", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Order placed by mistake',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count_Preference', fontsize = 12)
plt.tight_layout()
```



Insights from Influence in Cancellation Order

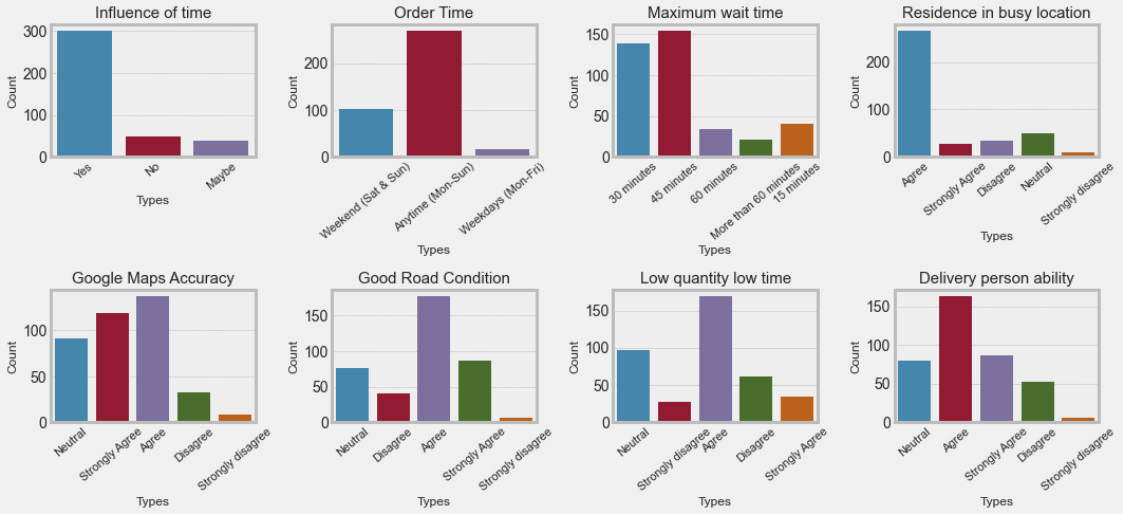
- People tend to feel that long time in the delivery is highly influencing them to cancel their order.
- Delay in delivery person getting assigned to them & delay in delivery person picking up the food affects to people's decision of cancelling the order
- Large amount of people do not cancel order although they have missing item or wrong item being delivered.

1.6.7 Time factors in order demand

[17]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('bmh')

#Influence of time Countplot
plt.subplot(2,4,1)
ax = sns.countplot(x="Influence of time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Influence of time',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Order Time Countplot
plt.subplot(2,4,2)
ax = sns.countplot(x="Order Time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Order Time',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
```

```
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Maximum wait time Countplot
plt.subplot(2,4,3)
ax = sns.countplot(x="Maximum wait time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Maximum wait time',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Good Road Condition Countplot
plt.subplot(2,4,6)
ax = sns.countplot(x="Good Road Condition", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Good Road Condition',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Hygiene Countplot
plt.subplot(2,4,4)
ax = sns.countplot(x="Residence in busy location", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Residence in busy location',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Accuracy Countplot
plt.subplot(2,4,5)
ax = sns.countplot(x="Google Maps Accuracy", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Google Maps Accuracy',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Low quantity low time Countplot
plt.subplot(2,4,7)
ax = sns.countplot(x="Low quantity low time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Low quantity low time',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()
```

```
#Delivery person ability Countplot
plt.subplot(2,4,8)
ax = sns.countplot(x="Delivery person ability", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Delivery person ability',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()
```



Insights from Time factors on order demand

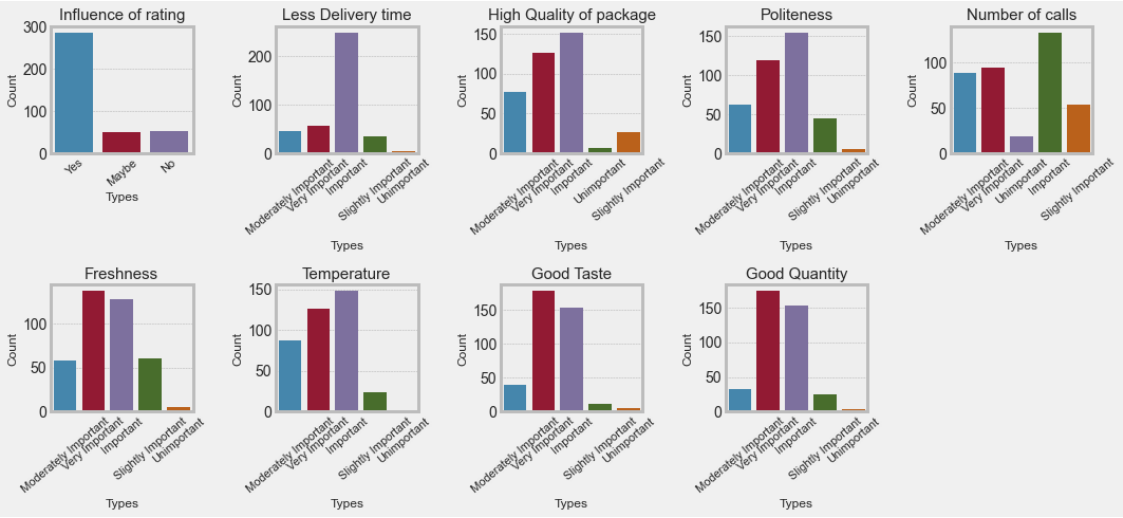
- Majority of respondents are influenced by the time of entire process
- Respondents' maximum wait time rank : 1st) 45 minutes, 2nd) 30 minutes, 3rd) 60 minutes
- Google map accuracy seems to be very accurate in most cases.
- Majority think that it is up to the delivery person to figure out the location and delivery the order at the shortest possible time.

1.6.8 Importance of rating post delivery

[18]:
```
#Setting up the frame
plt.figure(figsize = (15, 7))
plt.style.use('bmh')

#Influence of rating Countplot
plt.subplot(2,5,1)
ax = sns.countplot(x="Influence of rating", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Influence of rating',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
```

```
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Less Delivery time Countplot
plt.subplot(2,5,2)
ax = sns.countplot(x="Less Delivery time", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Less Delivery time',fontsize = 15)
ax.set_xlabel('Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#High Quality of package Countplot
plt.subplot(2,5,3)
ax = sns.countplot(x="High Quality of package", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('High Quality of package',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()


#Politeness Countplot
plt.subplot(2,5,4)
ax = sns.countplot(x="Politeness", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Politeness',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Number of calls Countplot
plt.subplot(2,5,5)
ax = sns.countplot(x="Number of calls", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Number of calls',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Freshness Countplot
plt.subplot(2,5,6)
ax = sns.countplot(x="Freshness ", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Freshness',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
```

```
plt.tight_layout()

#Temperature Countplot
plt.subplot(2,5,7)
ax = sns.countplot(x="Temperature", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Temperature',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Good Taste Contplot
plt.subplot(2,5,8)
ax = sns.countplot(x="Good Taste ", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Good Taste',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()

#Good Quantity Countplot
plt.subplot(2,5,9)
ax = sns.countplot(x="Good Quantity", data=delivery)
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax.set_title('Good Quantity',fontsize = 15)
ax.set_xlabel( 'Types',fontsize = 12)
ax.set_ylabel('Count', fontsize = 12)
plt.tight_layout()
```



Insights from Importance of rating post delivery

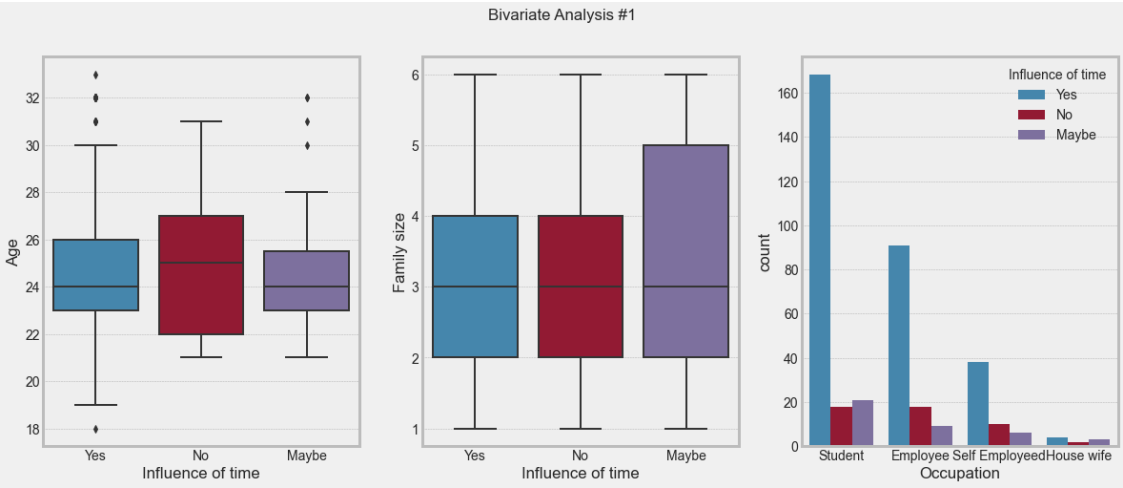
- Around 280 respondents agreed that they consider quality of package as important for the rating of delivery.
- Lesser delivery time is considered to be an important factor while giving rating.
- For delivery person : Consider number of calls made and politeness are considered important factor by majority of respondent.
- Delivery person will make more calls if there are any issue
- Delivery people are less polite if they feel the delivery process is difficult and that also reflects their rating.
- Freshness is the most important factor for rating.

1.7 Bivariate Analysis

1.7.1 Bivariate Analysis #1

[19]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 7))


fig.suptitle('Bivariate Analysis #1')

sns.boxplot(ax=axes[0], data=delivery, x='Influence of time', y='Age')
sns.boxplot(ax=axes[1], data=delivery, x='Influence of time', y='Family size')
sns.countplot(ax=axes[2],data=delivery,x="Occupation", hue="Influence of time")
```

[19]:
```
<AxesSubplot:xlabel='Occupation', ylabel='count'>
```



Insights from Bivariate Analysis #1

- Most of the age groups are influenced by time of delivery.
- In mid-age group, some people are not influenced by time of delivery that much.
- Most of the family size are influenced by time of delivery.

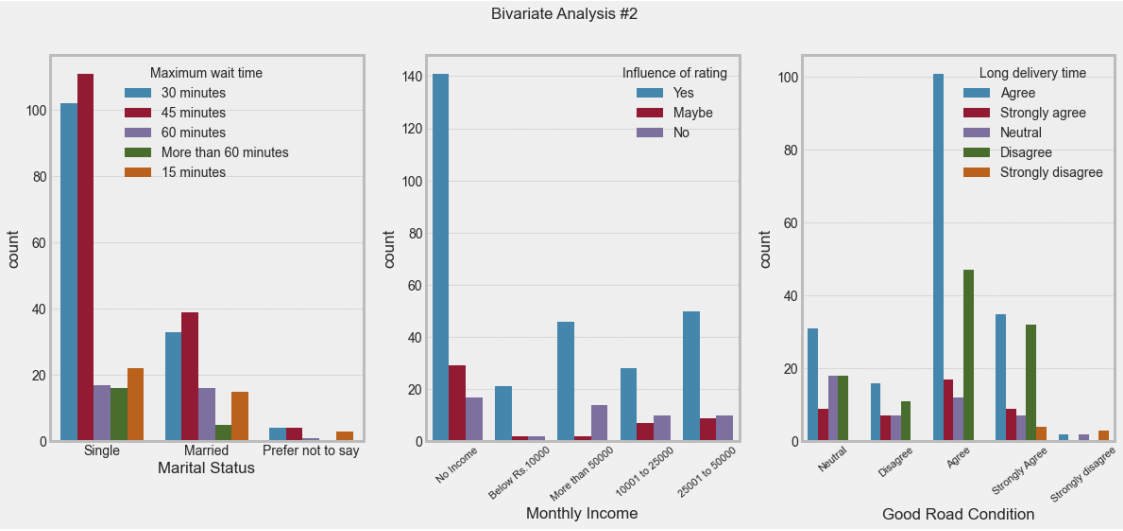
- Student groups are mostly influenced by delivery time and house wives are leaast influenced by delivery time.

1.7.2 Bivariate Analysis #2

1. Marital status and wait time
2. Income and inluence in rating
3. Road condition and long time in delivery

[20]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 7))


fig.suptitle('Bivariate Analysis #2')

sns.countplot(ax=axes[0],data=delivery,x="Marital Status", hue="Maximum wait␣
 ↪time")
ax=sns.countplot(ax=axes[1],data=delivery,x="Monthly Income", hue="Influence of␣
 ↪rating")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax=sns.countplot(ax=axes[2],data=delivery,x="Good Road Condition", hue="Long␣
 ↪delivery time")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
```

[20]:
```
[Text(0, 0, 'Neutral'),
 Text(1, 0, 'Disagree'),
 Text(2, 0, 'Agree'),
 Text(3, 0, 'Strongly Agree'),
 Text(4, 0, 'Strongly disagree')]
```



Insight from Bivariate Analysis #2

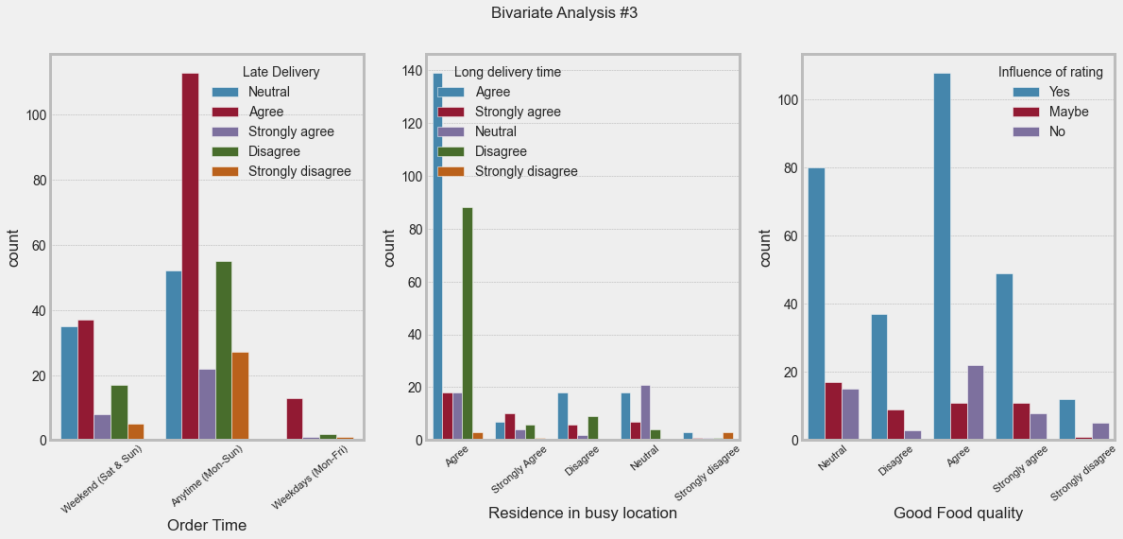
- For all kind of people whether no matter their marital status, normally respondents can wait 30 -45 minutes.

1.7.3 Bivariate Analysis #3

1. Order time and late delivery
2. Busy location and long time in delivery
3. Food quality and influence in rating

[21]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 7))
sns.set_style("white")

fig.suptitle('Bivariate Analysis #3')

ax=sns.countplot(ax=axes[0],data=delivery,x="Order Time", hue="Late Delivery")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax=sns.countplot(ax=axes[1],data=delivery,x="Residence in busy location",␣
 ↪hue="Long delivery time")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax=sns.countplot(ax=axes[2],data=delivery,x="Good Food quality", hue="Influence␣
 ↪of rating")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
```

[21]:
```
[Text(0, 0, 'Neutral'),
 Text(1, 0, 'Disagree'),
 Text(2, 0, 'Agree'),
 Text(3, 0, 'Strongly agree'),
 Text(4, 0, 'Strongly disagree')]
```



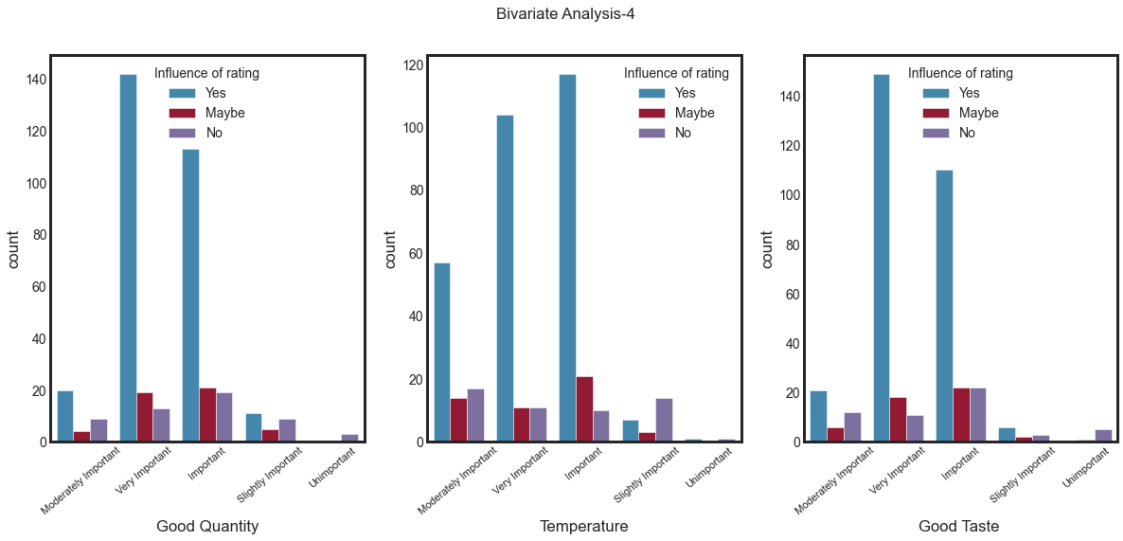
Insight from Bivariate Analysis #3

- People usually feel order time is late no matter weekdays and weekends.
- Large number of people who think they are from busy location feels that delivery time is kind of long.
- No matter the level of the quality of food, people usually think that the quality of food influce their rating.

1.7.4 Bivariate Analysis #4

1. Quantity and influence in rating
2. Food temperature and influence in rating
3. Taste and influence in rating

[22]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 7))


fig.suptitle('Bivariate Analysis-4')

ax=sns.countplot(ax=axes[0],data=delivery,x="Good Quantity", hue="Influence of␣
 ↪rating")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax=sns.countplot(ax=axes[1],data=delivery,x="Temperature", hue="Influence of␣
 ↪rating")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)
ax=sns.countplot(ax=axes[2],data=delivery,x="Good Taste ", hue="Influence of␣
 ↪rating")
ax.set_xticklabels(ax.get_xticklabels(), fontsize=11,rotation=40)

sns.set_style("dark")
```



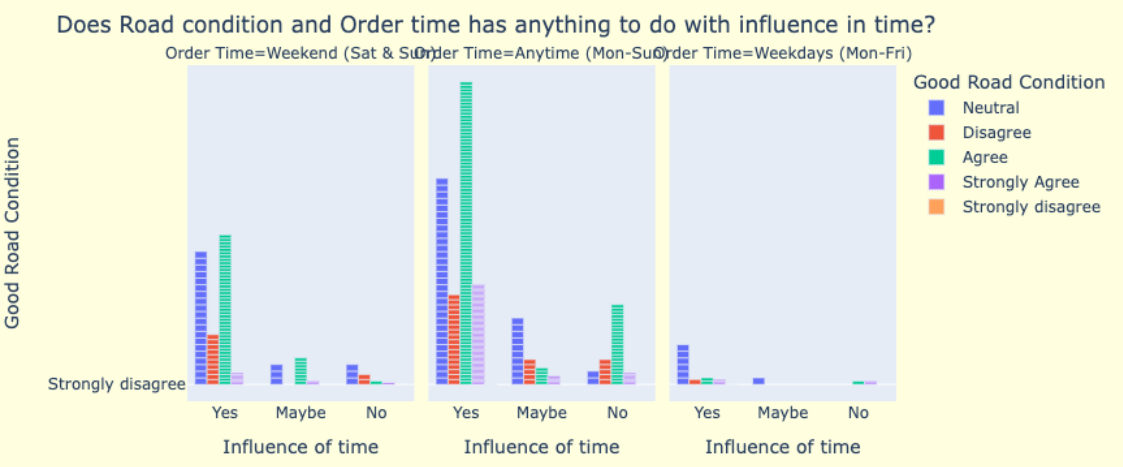
- Food Quantity is an important factor for respondents' rating

- Temperature and taste food is also an important factor for respondents' rating

1.8 Multivariate Analysis

1.8.1 Multivariate Analysis #1

[23]:
```
#Considering 3 variables
fig = px.bar(delivery, x="Influence of time", y="Good Road Condition",
             color="Good Road Condition", barmode="group",facet_col="Order␣
 ↪Time")

#Setting up the title
fig.update_layout(
    margin=dict(l=20, r=20, t=50, b=20),
    paper_bgcolor="LightYellow",
    title_text='Does Road condition and Order time has anything to do with␣
 ↪influence in time?'
)
fig.show()
```



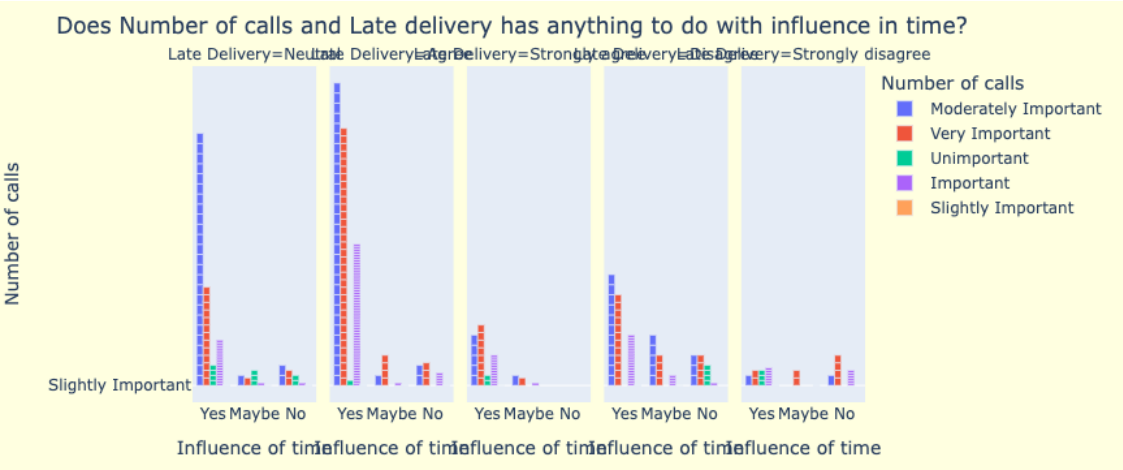
Insight from Multivariate Analysis #1

- High influence of time exist especially during weekdays and when the road conditions are good.

1.8.2 Multivariate Analysis #2

[24]:
```
#Considering 3 variables
fig = px.bar(delivery, x="Influence of time", y="Number of calls",
             color="Number of calls", barmode="group",facet_col="Late Delivery")

#Setting up the title
```

```
fig.update_layout(
    margin=dict(l=20, r=20, t=50, b=20),
    paper_bgcolor="LightYellow",
    title_text='Does Number of calls and Late delivery has anything to do with influence in time?'
)
fig.show()
```

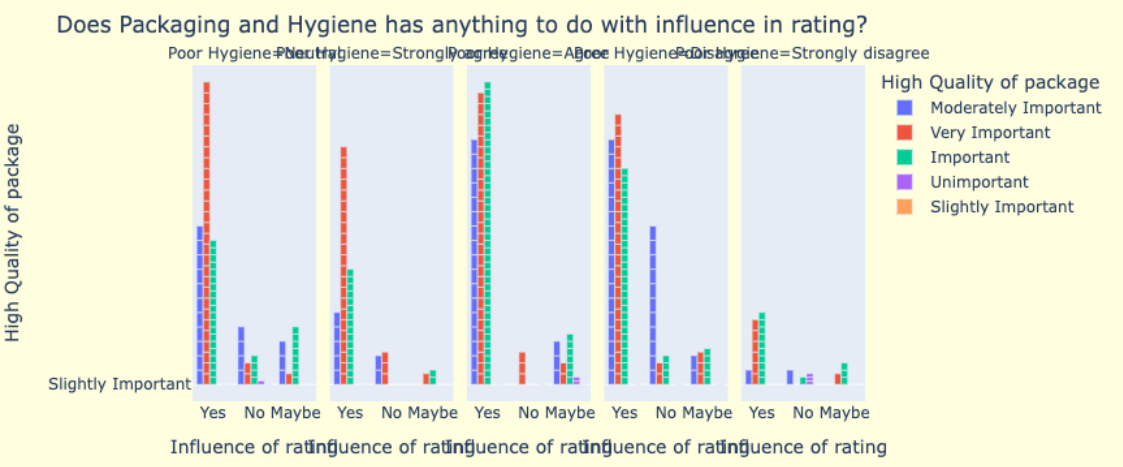


Insight from Multivariate Analysis #2

- A lot of influence in time when the people consider number of calls as an important factor and there is a late delivery.
- Delivery person calls the customer to pass on any information and or when the person couldn't reach the destination which consumes time and the delivery gets delayed.

1.8.3 Multivariate Ananlysis #3

[25]:
```
#Considering 3 variables
fig = px.bar(delivery, x="Influence of rating", y="High Quality of package",
             color="High Quality of package", barmode="group",facet_col="Poor Hygiene")

#Setting up the title
fig.update_layout(
    margin=dict(l=20, r=20, t=50, b=20),
    paper_bgcolor="LightYellow",
    title_text='Does Packaging and Hygiene has anything to do with influence in rating?'
)
fig.show()
```

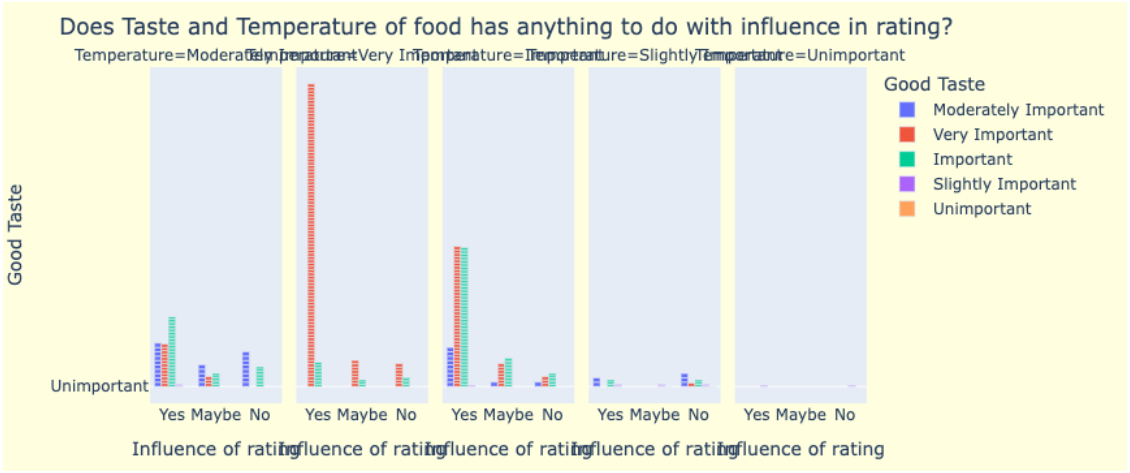


Insight from Multivariate Analysis #3

- Quality of package is very important and hygiene also plays an important role

1.8.4 Multivariate Analysis #4

[26]:
```
#Considering 3 variables
fig = px.bar(delivery, x="Influence of rating", y="Good Taste ",
             color="Good Taste ", barmode="group",facet_col="Temperature")

#Setting up the title
fig.update_layout(
    margin=dict(l=20, r=20, t=50, b=20),
    paper_bgcolor="LightYellow",
    title_text='Does Taste and Temperature of food has anything to do with␣
 ↪influence in rating?'
)
fig.show()
```

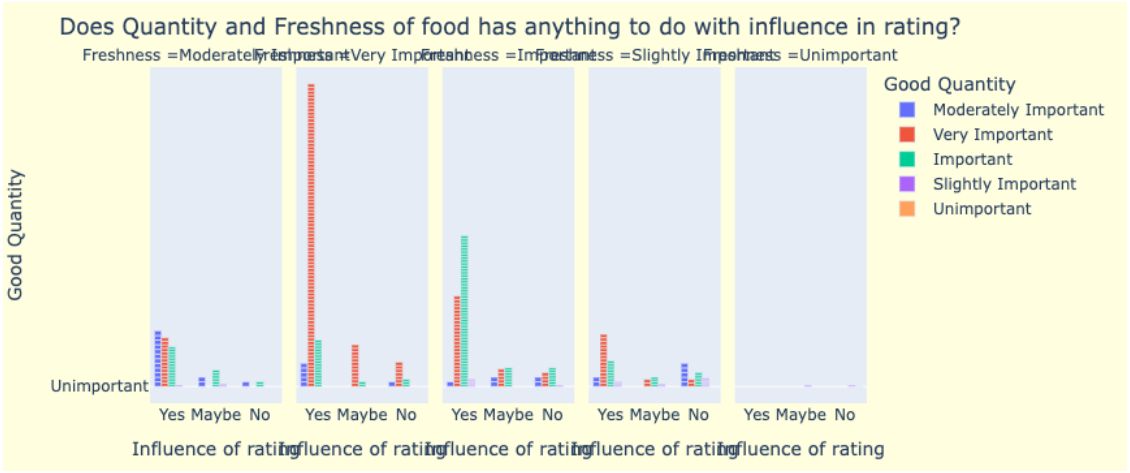


Insight from Multivariate Analysis #4

- People consider both taste and temperature of food important.
- There are significantly a smaller number of people who has voted when they have no influence of rating.

1.8.5 Multivariate Analysis #5

[27]:
```
#Considering 3 variables
fig = px.bar(delivery, x="Influence of rating", y="Good Quantity",
             color="Good Quantity", barmode="group",facet_col="Freshness ")

#Setting up the title
fig.update_layout(
    margin=dict(l=20, r=20, t=50, b=20),
    paper_bgcolor="LightYellow",
    title_text='Does Quantity and Freshness of food has anything to do with␣
 ↪influence in rating?'
)
fig.show()
```



Insight from Multivariate Analysis #5

- People consider both Quality and Freshness are very important for rating a restaurant
- There are significantly a smaller number of people who has voted when they have no influence of rating.

1.9 Geospatial Analysis

1.9.1 Geo Pointers

[28]:
```
Age_band = delivery[(delivery.Age.isin(range(18,40)))]
# Creating a map
m_2 = folium.Map(location=[12.9716,77.5946], tiles='cartodbpositron',␣
 ↪zoom_start=13)

# Adding points to the map
for idx, row in Age_band.iterrows():
    Marker([row['latitude'], row['longitude']]).add_to(m_2)

# Displaying the map
m_2
```

[28]:
```
<folium.folium.Map at 0x7fcfb65b1e80>
```

Insight form Geo Pointers

- ?

1.9.2 Clustered Location

[29]:
```
# Creating the map
m_3 = folium.Map(location=[12.9716,77.5946], tiles='cartodbpositron',␣
 ↪zoom_start=13)

# Adding points to the map
mc = MarkerCluster()
for idx, row in Age_band.iterrows():
    if not math.isnan(row['longitude']) and not math.isnan(row['latitude']):
        mc.add_child(Marker([row['latitude'], row['longitude']]))
m_3.add_child(mc)

# Displaying the map
m_3
```

[29]:
```
<folium.folium.Map at 0x7fcfb76d98b0>
```

Insight from Clustered Location

- Most of the data are concentrated in the heart of city : places like Gandhinagar, Chikkapete, VVPuram, Rajajinagar.

1.10 Text Analysis

For analysing the text - Review column is for an open ended question from the questionnaire where the respondents can pass on any comment about the service they experience through the food delivery startup

1.10.1 Processing Review Column

little processing by removing the records which has nil in their review WHAT IS NIL

[30]:
```
#Creating a copy
delivery_reviews=delivery.copy()

#Removing unecessary reviews- Nil value
delivery_reviews= delivery[~delivery['Reviews'].
 ↪isin(['NIL','nil','Nil','No','Nil\n'])]

# Considering only two columns
delivery_reviews=delivery_reviews[['Reviews','Output']]

#Viewing the data
delivery_reviews
```

[30]:
```
                                              Reviews Output
2    Many a times payment gateways are an issue, so…    Yes
```

```
11   Language barrier is also one major issue. Mosl…    Yes
17          Spillage, bad packaging and missing items     Yes
18   Once my order from kfc got exchanged with some…    Yes
22   I feel Swiggy has a good interface for users a…    Yes
..                                                 …    …
375            I had bad quality order delivered twice     No
376  Bad rating doesn't mean that the food tastes b…    Yes
377            Order delivered to my location are late     No
378  My location is pretty well built for food deli…    Yes
386  Language barrier is also one major issue. Mosl…    Yes

[243 rows x 2 columns]
```

1.10.2 Cleaning Reviews

[31]:
```
#Function for cleaning
def review_cleaning(text):
    '''Make text lowercase, remove text in square brackets,remove links,remove␣
 ↪punctuation
    and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

#Applying the function on reviews column
delivery_reviews['Reviews']=delivery_reviews['Reviews'].apply(lambda x:
 ↪review_cleaning(x))
delivery_reviews.head()
```

[31]:
```
                                              Reviews Output
2   many a times payment gateways are an issue so …    Yes
11  language barrier is also one major issue moslt…    Yes
17          spillage bad packaging and missing items     Yes
18  once my order from kfc got exchanged with some…    Yes
22  i feel swiggy has a good interface for users a…    Yes
```

1.10.3 Deriving new variables

Create polarity, review length and word count

- Polarity: We use Textblob for for figuring out the rate of sentiment . It is between [-1,1] where -1 is negative and 1 is positive polarity

- Review length: length of the review which includes each letters and spaces
- Word length: This measures how many words are there in review

[32]:
```
#Creating new variables
delivery_reviews['polarity'] = delivery_reviews['Reviews'].map(lambda text:␣
 ↪TextBlob(text).sentiment.polarity)
delivery_reviews['review_len'] = delivery_reviews['Reviews'].astype(str).
 ↪apply(len)
delivery_reviews['word_count'] = delivery_reviews['Reviews'].apply(lambda x:␣
 ↪len(str(x).split()))

#Viewing the new variables
delivery_reviews.head()
```

[32]:

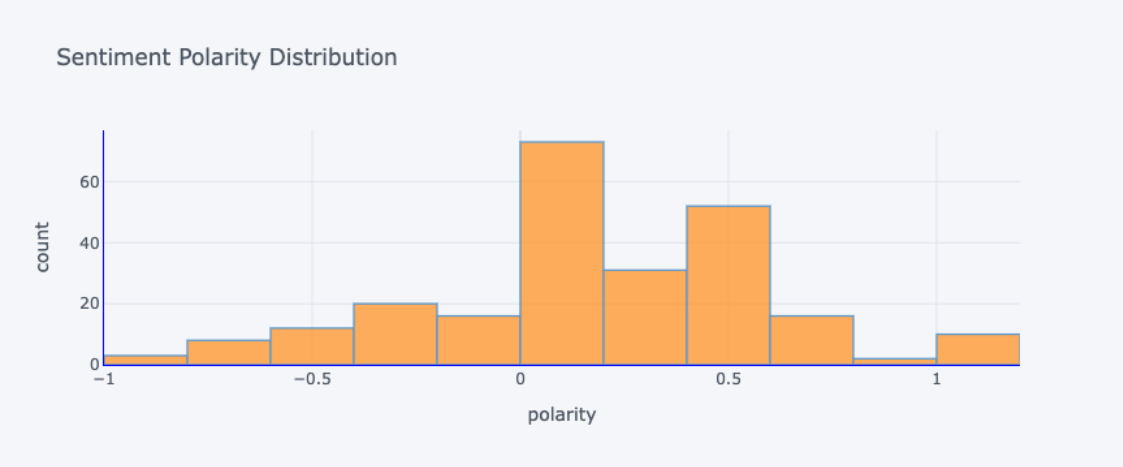
	Reviews	Output	polarity
2	many a times payment gateways are an issue so …	Yes	0.500000
11	language barrier is also one major issue moslt…	Yes	0.145833
17	spillage bad packaging and missing items	Yes	-0.450000
18	once my order from kfc got exchanged with some…	Yes	0.000000
22	i feel swiggy has a good interface for users a…	Yes	0.177778

	review_len	word_count
2	104	18
11	148	24
17	41	6
18	79	14
22	162	30

1.10.4 Polarity of Reviews

Check the distribution of polarity which can give us an idea of how the sentiment are distributed

[33]:
```
delivery_reviews['polarity'].iplot(
    kind='hist',
    bins=15,
    xTitle='polarity',
    linecolor='blue',
    yTitle='count',
    title='Sentiment Polarity Distribution')
```



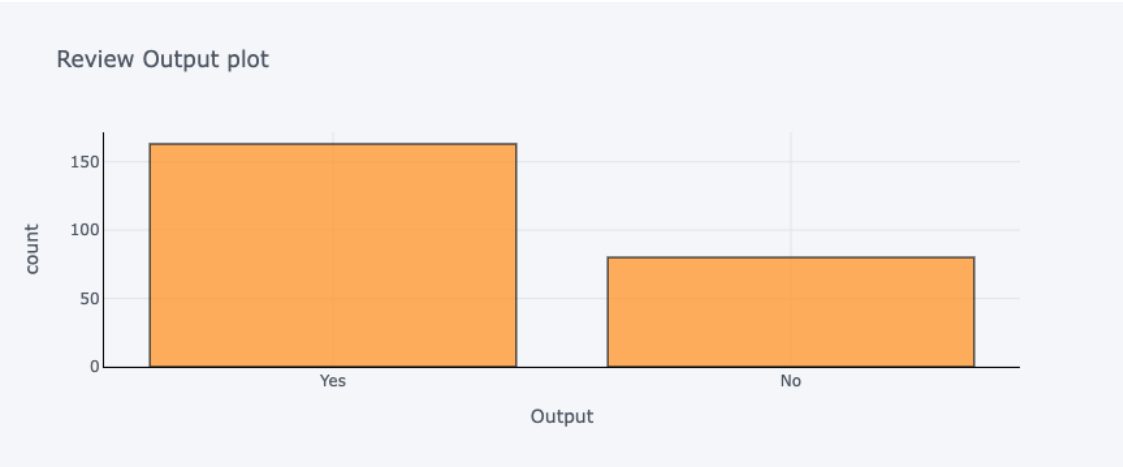
Insight from Polarity of Reviews

- Distribution is normally distributed
- Has an almost balanced proportion of positive and negative reviews
- Most of the them are populated around 0 which expresses neutral sentiment

1.10.5 Output Plot

Plot the output from the data, the output defines that the customer will purchase the product again from the food delivery service if the circumstances of their responses stays same and true.

[34]:
```
delivery_reviews['Output'].iplot(
    kind='hist',
    xTitle='Output',
    linecolor='black',
    yTitle='count',
    title='Review Output plot')
```



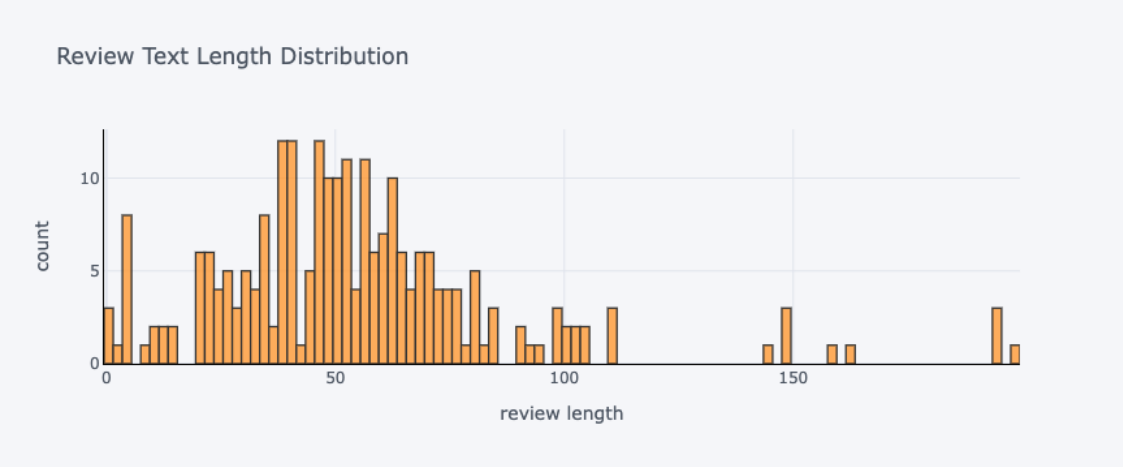
Insight from Output Plot

- The count of 'Yes' is double the times as 'No'
- Most of the respondents are positive about the service and they have declared they will buy from the same food delivery service

1.10.6 Length of Review

Check how long the respondents have typed to express their feedback

[35]:
```
delivery_reviews['review_len'].iplot(
    kind='hist',
    bins=100,
    xTitle='review length',
    linecolor='black',
    yTitle='count',
    title='Review Text Length Distribution')
```



Insight from Length of Review

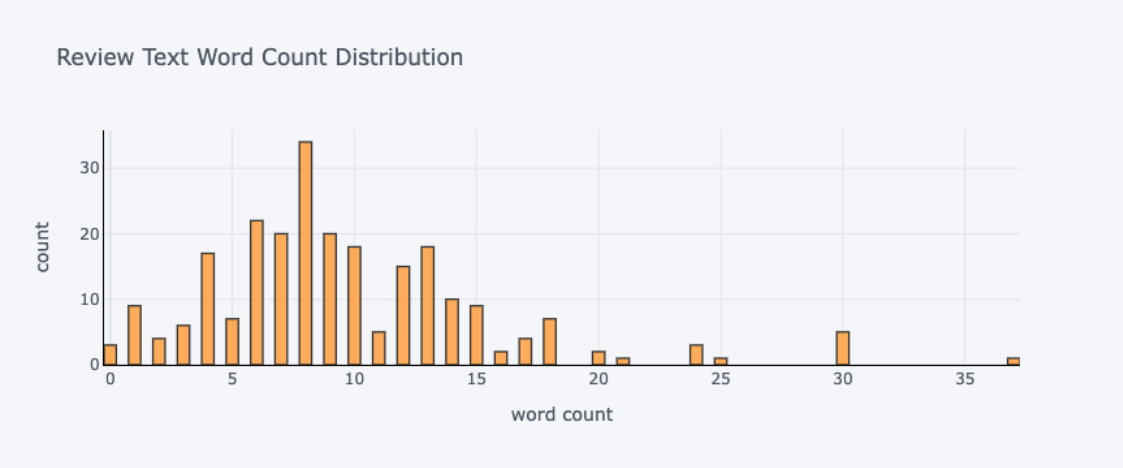
- The length of review has a right skewed distribution
- Most of the reviews are limited around 60 letters
- There are very few reviews which extends around 140 letters

1.10.7 Word Count Distribution

Letters including spaces - Split the sentences into words and calculate the count

[36]:
```
delivery_reviews['word_count'].iplot(
    kind='hist',
```

```
    bins=100,
    xTitle='word count',
    linecolor='black',
    yTitle='count',
    title='Review Text Word Count Distribution')
```



Insight from Word Count Distribution

- The word count also expresses a right skewed distribution
- The word count are highly populated(mode) at 5 and extends upto 10 words
- There were word counts which are nearly around 20.

1.11 N-gram Analysis

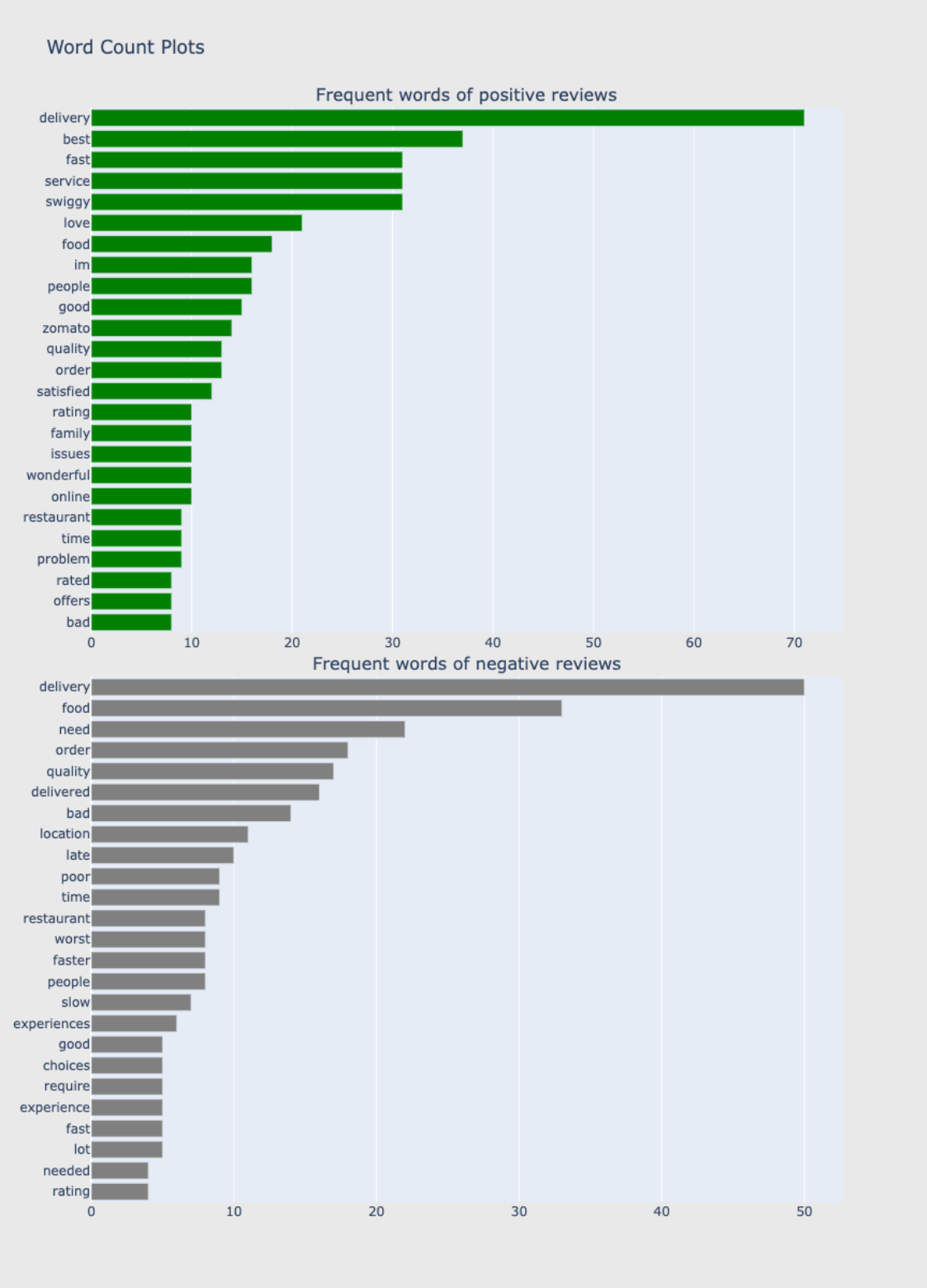
1.11.1 Monogram Analysis

Plot most frequent one word in reviews based on sentiments

[37]:
```
#Filtering data
review_pos = delivery_reviews[delivery_reviews["Output"]=='Yes']
review_neg = delivery_reviews[delivery_reviews["Output"]=='No']

## custom function for ngram generation ##
def generate_ngrams(text, n_gram=1):
    token = [token for token in text.lower().split(" ") if token != "" if token␣
 ↪not in STOPWORDS]
    ngrams = zip(*[token[i:] for i in range(n_gram)])
    return [" ".join(ngram) for ngram in ngrams]

## custom function for horizontal bar chart ##
def horizontal_bar_chart(df, color):
    trace = go.Bar(
```

```
        y=df["word"].values[::-1],
        x=df["wordcount"].values[::-1],
        showlegend=False,
        orientation = 'h',
        marker=dict(
            color=color,
        ),
    )
    return trace

## Get the bar chart from positive reviews ##
freq_dict = defaultdict(int)
for sent in review_pos["Reviews"]:
    for word in generate_ngrams(sent):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace0 = horizontal_bar_chart(fd_sorted.head(25), 'green')

## Get the bar chart from negative reviews ##
freq_dict = defaultdict(int)
for sent in review_neg["Reviews"]:
    for word in generate_ngrams(sent):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace1 = horizontal_bar_chart(fd_sorted.head(25), 'gray')

# Creating two subplots
fig = tools.make_subplots(rows=2, cols=1, vertical_spacing=0.04,
                          subplot_titles=["Frequent words of positive reviews",
                                          "Frequent words of negative reviews"])
fig.append_trace(trace0, 1, 1)
fig.append_trace(trace1, 2, 1)
fig['layout'].update(height=1200, width=900, paper_bgcolor='rgb(233,233,233)',␣
 ↪title="Word Count Plots")
iplot(fig, filename='word-plots')
```

```
/Users/hwangtaejin/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-
packages/plotly/tools.py:460: DeprecationWarning:

plotly.tools.make_subplots is deprecated, please use
plotly.subplots.make_subplots instead
```



Insight from Monogram Analysis

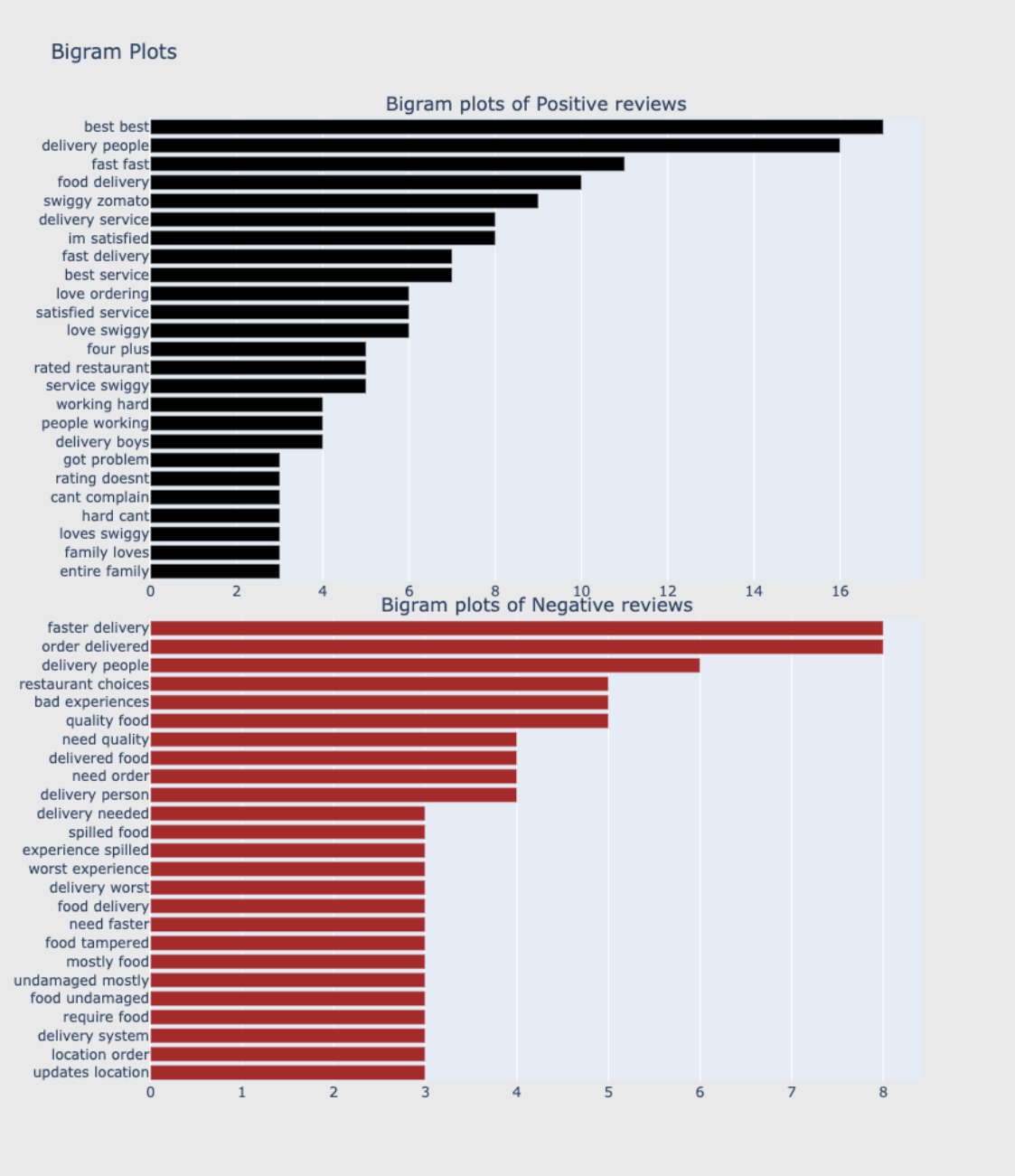
- Most of the good reviews are on swiggy,zomato, faster service etc,
- Bad reviews mostly contain food quality,location and late deliveries

1.11.2 Bigram Analysis

Plot most frequent two words in reviews based on sentiments

[38]:
```
## Get the bar chart from positive reviews ##
freq_dict = defaultdict(int)
for sent in review_pos["Reviews"]:
    for word in generate_ngrams(sent,2):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace0 = horizontal_bar_chart(fd_sorted.head(25), 'black')


## Get the bar chart from negative reviews ##
freq_dict = defaultdict(int)
for sent in review_neg["Reviews"]:
    for word in generate_ngrams(sent,2):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace1 = horizontal_bar_chart(fd_sorted.head(25), 'brown')

# Creating two subplots
fig = tools.make_subplots(rows=2, cols=1, vertical_spacing=0.
 ↪04,horizontal_spacing=0.25,
                          subplot_titles=["Bigram plots of Positive reviews",
                                          "Bigram plots of Negative reviews"
                                          ])
fig.append_trace(trace0, 1, 1)
fig.append_trace(trace1, 2, 1)
fig['layout'].update(height=1000, width=800, paper_bgcolor='rgb(233,233,233)',␣
 ↪title="Bigram Plots")
iplot(fig, filename='word-plots')
```

```
/Users/hwangtaejin/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-
packages/plotly/tools.py:460: DeprecationWarning:

plotly.tools.make_subplots is deprecated, please use
plotly.subplots.make_subplots instead
```



Insight from Bigram Analysis

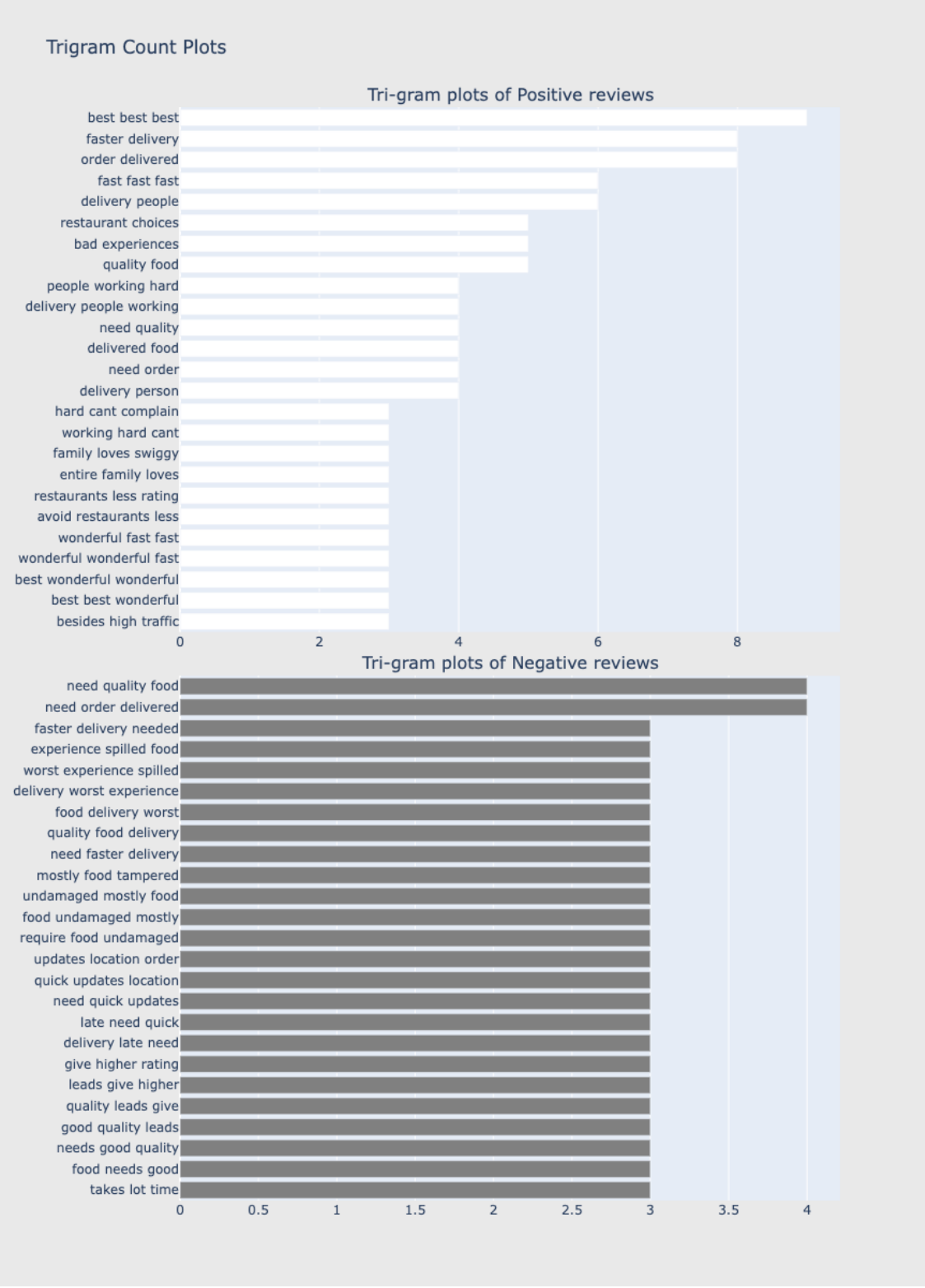
- From positive reviews : Swiggy and Zomato are the most loved food services and people mostly provide 4+ stars for their service
- From negative reviews : about the delivery people and bad experiences & people have complained about damaged and spilled food.

1.11.3 Trigram Analysis

Plot most frequent three words in reviews based on sentiment

[39]:
```
## Get the bar chart from positive reviews ##
for sent in review_pos["Reviews"]:
    for word in generate_ngrams(sent,3):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace0 = horizontal_bar_chart(fd_sorted.head(25), 'white')


## Get the bar chart from negative reviews ##
freq_dict = defaultdict(int)
for sent in review_neg["Reviews"]:
    for word in generate_ngrams(sent,3):
        freq_dict[word] += 1
fd_sorted = pd.DataFrame(sorted(freq_dict.items(), key=lambda x: x[1])[::-1])
fd_sorted.columns = ["word", "wordcount"]
trace1 = horizontal_bar_chart(fd_sorted.head(25), 'gray')


# Creating two subplots
fig = tools.make_subplots(rows=2, cols=1, vertical_spacing=0.04,␣
 ↪horizontal_spacing=0.05,
                          subplot_titles=["Tri-gram plots of Positive reviews",
                                          "Tri-gram plots of Negative reviews"])
fig.append_trace(trace0, 1, 1)
fig.append_trace(trace1, 2, 1)
fig['layout'].update(height=1200, width=1200, paper_bgcolor='rgb(233,233,233)',␣
 ↪title="Trigram Count Plots")
iplot(fig, filename='word-plots')
```

```
/Users/hwangtaejin/Library/jupyterlab-desktop/jlab_server/lib/python3.8/site-
packages/plotly/tools.py:460: DeprecationWarning:

plotly.tools.make_subplots is deprecated, please use
plotly.subplots.make_subplots instead
```

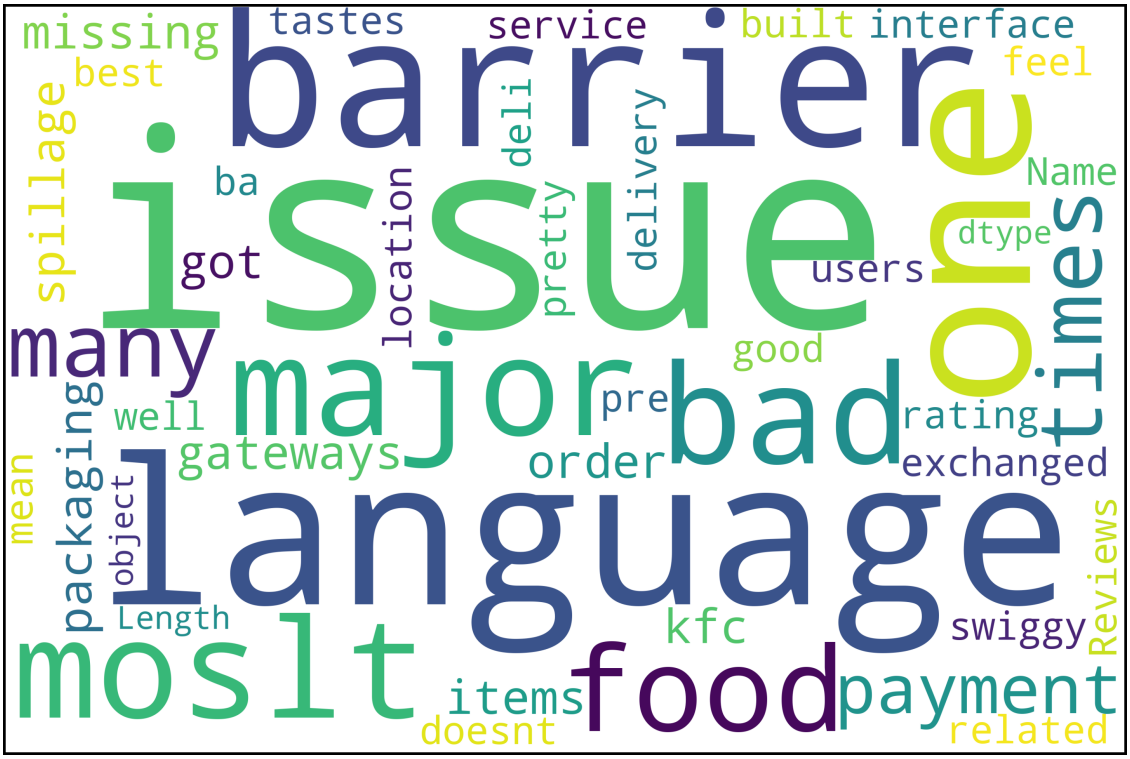


Insight from Trigram Analysis

- Damaged and spilled foods are especially been frowned by the customers.

1.11.4 Wordcloud-Positive reviews

[40]:
```
text = review_pos["Reviews"]
wordcloud = WordCloud(
    width = 3000,
    height = 2000,
    background_color = 'white',
    stopwords = STOPWORDS).generate(str(text))
fig = plt.figure(
    figsize = (40, 30),
    facecolor = 'k',
    edgecolor = 'k')
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

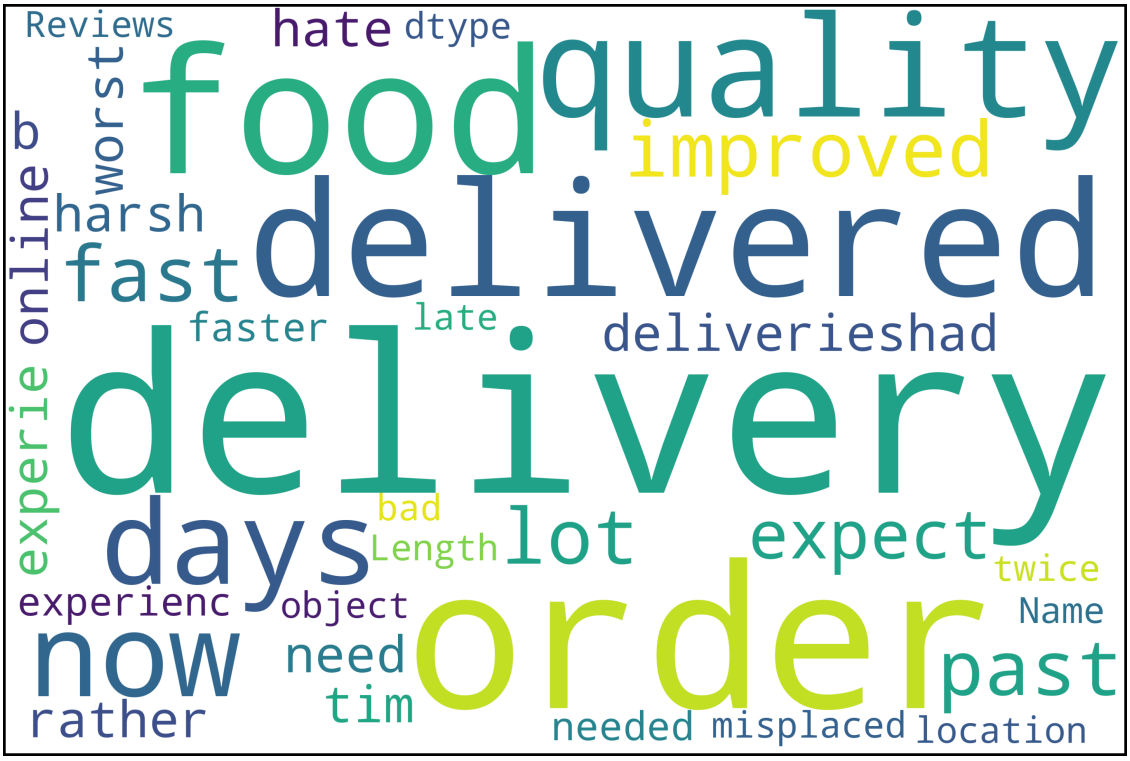


Insight from Wordcloud-Positive Review

- There could be words such as not,no associated before them which could have been removed through stopwords

1.11.5 Wordcloud-Negative Reviews

[41]:
```
text = review_neg["Reviews"]
wordcloud = WordCloud(
    width = 3000,
    height = 2000,
    background_color = 'white',
    stopwords = STOPWORDS).generate(str(text))
fig = plt.figure(
    figsize = (40, 30),
    facecolor = 'k',
    edgecolor = 'k')
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



Insight from Wordcloud-Negative reviews

- Example of Negative words : quality,misplaced, worst

[42]:
```
import nbconvert
```

[43]:
```
pip install nbconvert[webpdf]
```

```
zsh:1: no matches found: nbconvert[webpdf]
Note: you may need to restart the kernel to use updated packages.
```

[44]:
```
pip install xelatex
```

```
ERROR: Could not find a version that satisfies the requirement xelatex
(from versions: none)
ERROR: No matching distribution found for xelatex
Note: you may need to restart the kernel to use updated packages.
```

[ ]:
```

```