Introduction

In this dataset, there are 4622 records of Chipotle orders. Each order shows how many quantities the order contained, what were the items of the order, and the total price. Lets do some Data Exploration and see what we can find

Data Wrangling

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [66]:
```
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_rows', None)
```

In [2]:
```
df = pd.read_csv('Chipotle Sales.csv')
df
```

Out[2]:

	Order_ID	Quantity	Item_Name	Choice_Description	Item_Price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
...	...	...	...	...	...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$8.75

4622 rows × 5 columns

In [5]:
```
df.info()
#Need to convert Order_ID into string
#Need to convert Item_Price into float
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Order_ID            4622 non-null   int64 
 1   Quantity            4622 non-null   int64 
 2   Item_Name           4622 non-null   object
 3   Choice_Description  3376 non-null   object
 4   Item_Price          4622 non-null   object
dtypes: int64(2), object(3)
memory usage: 180.7+ KB
```

In [8]:
```
# After Data Cleaning
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Order_ID            4622 non-null   object
 1   Quantity            4622 non-null   int64 
 2   Item_Name           4622 non-null   object
 3   Choice_Description  3376 non-null   object
 4   Item_Price          4622 non-null   object
dtypes: int64(1), object(4)
memory usage: 180.7+ KB
```

In [15]:
```
df.describe()
#there are 4622 individual orders here, the average order contains 1.1 quantity(rounded)
#OK there is an order with 15 quantities. Need to check it out, and ask if it is real

#order price averages around $7.50
```

Out[15]:

	Quantity	Item_Price
count	4622.000000	4622.000000
mean	1.075725	7.464336
std	0.410186	4.245557
min	1.000000	1.090000
25%	1.000000	3.390000
50%	1.000000	8.750000
75%	1.000000	9.250000
max	15.000000	44.250000

In [23]:
```
df[df['Quantity'] >= 5]
#Someone ordered 15 only chips and Salsa
#And another person ordered 10 bottles of water
# will decide in exploratory analysis if these two should be removed
```

Out[23]:

	Order_ID	Quantity	Item_Name	Choice_Description	Item_Price
2441	970	5	Bottled Water	NaN	7.50
3598	1443	15	Chips and Fresh Tomato Salsa	NaN	44.25
3599	1443	7	Bottled Water	NaN	10.50
3887	1559	8	Side of Chips	NaN	13.52
4152	1660	10	Bottled Water	NaN	15.00

In [26]:
```
df.isna().sum()
#29% of the data has empty Choice_Description
```

Out[26]:
```
Order_ID                0
Quantity                0
Item_Name               0
Choice_Description    1246
Item_Price              0
dtype: int64
```

In [32]:
```
len(df[df.duplicated()])
#there are 59 duplicate rows
```

Out[32]:
```
59
```

In [40]:
```
#df[df.duplicated()]

df.query('Order_ID == "103" ')

#Yes there is duplicate Order_IDs, but these look like different orders to me
```

Out[40]:

	Order_ID	Quantity	Item_Name	Choice_Description	Item_Price
234	103	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Rice, Black Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	11.75
235	103	2	Chips and Tomatillo-Green Chili Salsa	NaN	5.90
236	103	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Rice, Black Beans, Cheese]]	9.25
237	103	1	Carnitas Soft Tacos	[Tomatillo Green Chili Salsa, [Fajita Vegetables, Pinto Beans, Cheese]]	9.25
238	103	1	Steak Burrito	[Tomatillo Red Chili Salsa, [Rice, Black Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	11.75

Data Cleaning

In [12]:
```
df['Order_ID'] = df['Order_ID'].astype(str)
df['Item_Price'] = df['Item_Price'].replace('[\$,]', '', regex=True).astype(float)
```

In [41]:
```
df
```

Out[41]:

	Order_ID	Quantity	Item_Name	Choice_Description	Item_Price
0	1	1	Chips and Fresh Tomato Salsa	NaN	2.39
1	1	1	Izze	[Clementine]	3.39
2	1	1	Nantucket Nectar	[Apple]	3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans, Rice, Cheese, Sour Cream]]	16.98
...	...	...	...	...	...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour Cream, Cheese, Lettuce, Guacamole]]	11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese, Lettuce, Guacamole]]	11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto Beans, Guacamole, Lettuce]]	11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettuce]]	8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto Beans, Lettuce]]	8.75

4622 rows × 5 columns

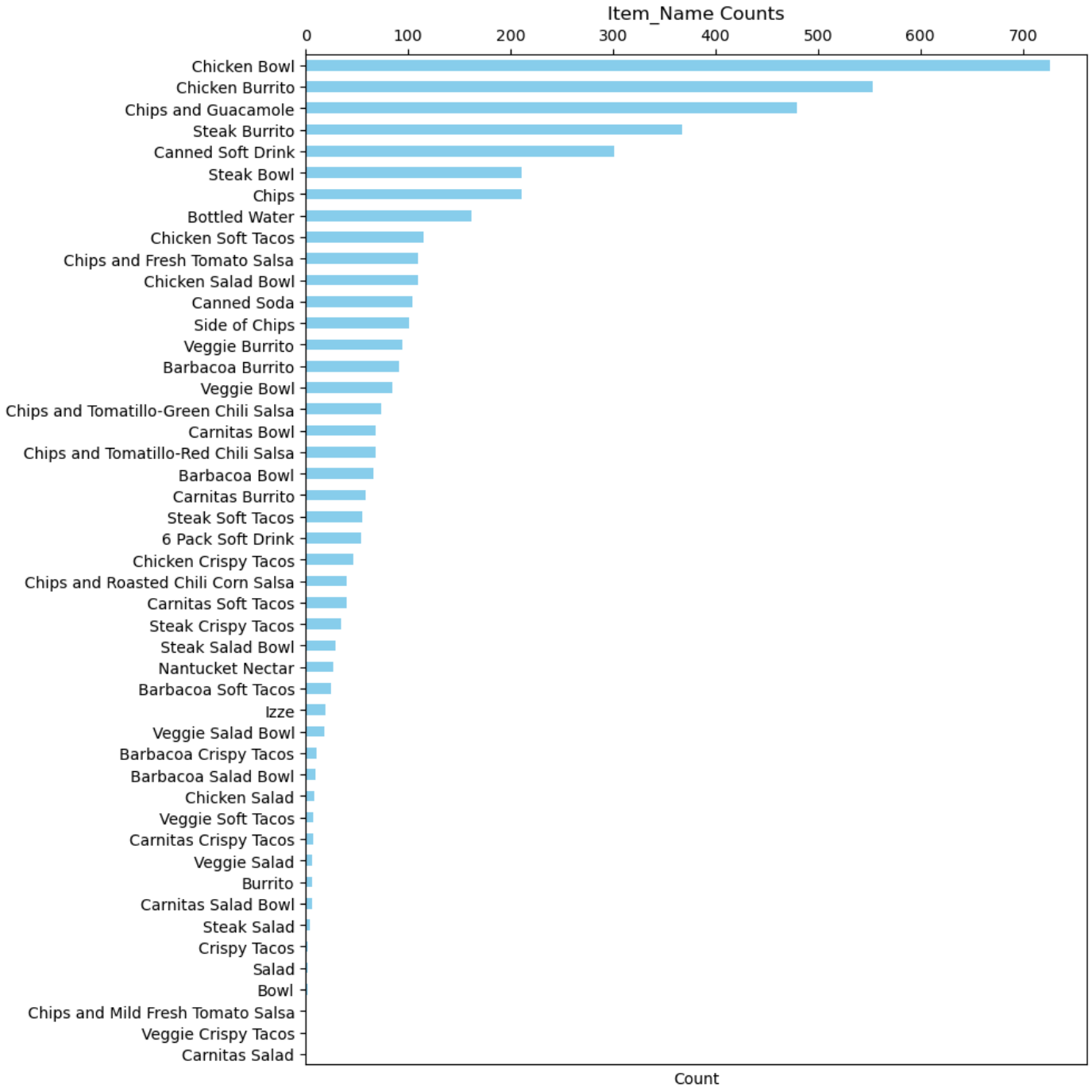
In [86]:
```
#Create Order_Type column, i.e. is it Bowl, Burrito or Tacos
df['Order_Type'] = np.where(df['Item_Name'].str.contains('Burrito'), 'Burrito',
                   np.where(df['Item_Name'].str.contains('Salad'), 'Salad',
                   np.where(df['Item_Name'].str.contains('Bowl'), 'Bowl',
                   np.where(df['Item_Name'].str.contains('Tacos'), 'Tacos',
                   np.where(df['Item_Name'].str.contains('Izze|Nantucket Nectar|Canned Soda|Bott
                   np.where(df['Item_Name'].str.contains('Salsa'), 'Salsa',
                   np.where(df['Item_Name'].str.contains('Chips and Guacamole'), 'Chips and Guac
                   np.where(df['Item_Name'].str.contains('Chips'), 'Chips',       'n/a')

                )))))))

#df
```

In [94]:
```
#Create Order_Type column, i.e. is it Bowl, Burrito or Tacos
df['Meat_Type'] = np.where(df['Item_Name'].str.contains('Steak'), 'Steak',
                  np.where(df['Item_Name'].str.contains('Chicken'), 'Chicken',
                  np.where(df['Item_Name'].str.contains('Barbacoa'), 'Barbacoa',
                  np.where(df['Item_Name'].str.contains('Carnitas'), 'Carnitas',
                  np.where(df['Item_Name'].str.contains('Veggie'), 'Veggie',  'other')

                ))))

#df
```

In [114…
```
meat = ['Chicken', 'Steak']
chicken_steak_df = df.query('Meat_Type == "Chicken" or Meat_Type == "Steak" ')
#Exclude outliers to show the true price distribution >> Back to Data Cleaning exclude those abo

chicken_steak_df = chicken_steak_df.query('Item_Price <= 15.0')
```

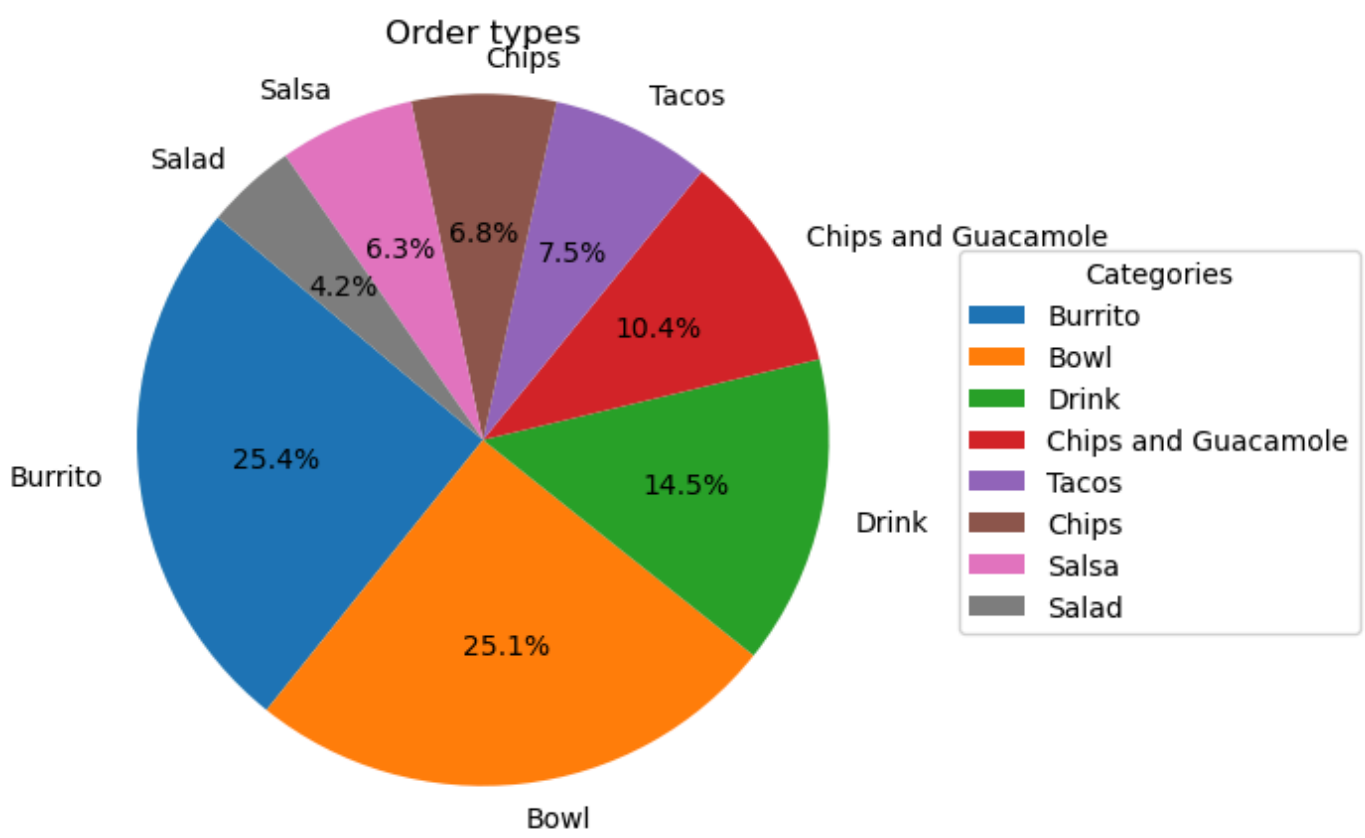
Exploratory Data Analysis

In [53]:
```
# What is Number 1 Item_Name mostly sold?
item_counts = df['Item_Name'].value_counts().sort_values(ascending=True)

plt.figure(figsize=(10, 10))
item_counts.plot(kind='barh', color='skyblue')

plt.title('Item_Name Counts')
plt.xlabel('Count')
#plt.ylabel('Item_Name')

plt.tick_params(axis='x', which='both', bottom=False, top=True, labelbottom=False, labeltop=True

plt.tight_layout()
plt.show()

#Chicken Bowl and Chicken Borrito are the top orders, then the Steak Borrito and Bowls
#Now I want to know if customers buy bowl, burrito, or tacos mostly? Back to Data Cleaning

#Second question I have is, which meat has most orders: Chicken or Steak or Barbacoa or Carnitas
# and also, is one pricer then the other?
```



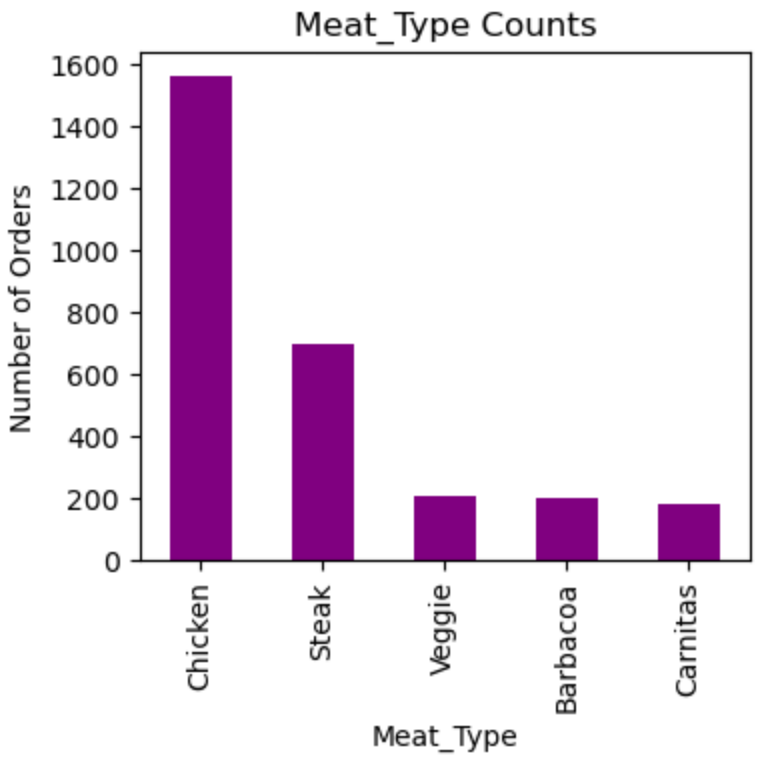
In [89]:

```
#to answer above questions, lets do pie graph
order_type_counts = df['Order_Type'].value_counts()

plt.figure(figsize=(6, 5))
plt.pie(order_type_counts, labels=order_type_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Order types')

plt.legend(title='Categories', loc='center left', bbox_to_anchor=(1, 0.5))

plt.axis('equal')
plt.show()

#OK there is equal spread of customers that order bowl and burrito
```



In [102…

```
meat_counts = df[df['Meat_Type'] != 'other']['Meat_Type'].value_counts().sort_values(ascending=F

plt.figure(figsize=(4, 4))
meat_counts.plot(kind='bar', color='purple')

plt.title('Meat_Type Counts')
plt.xlabel('Meat_Type')
plt.ylabel('Number of Orders')

plt.tight_layout()
plt.show()

print(meat_counts)

#Well, more than half of the orders chicken is ordered, I was expecting steak and chicken to be
#Now lets see what is the price difference
```

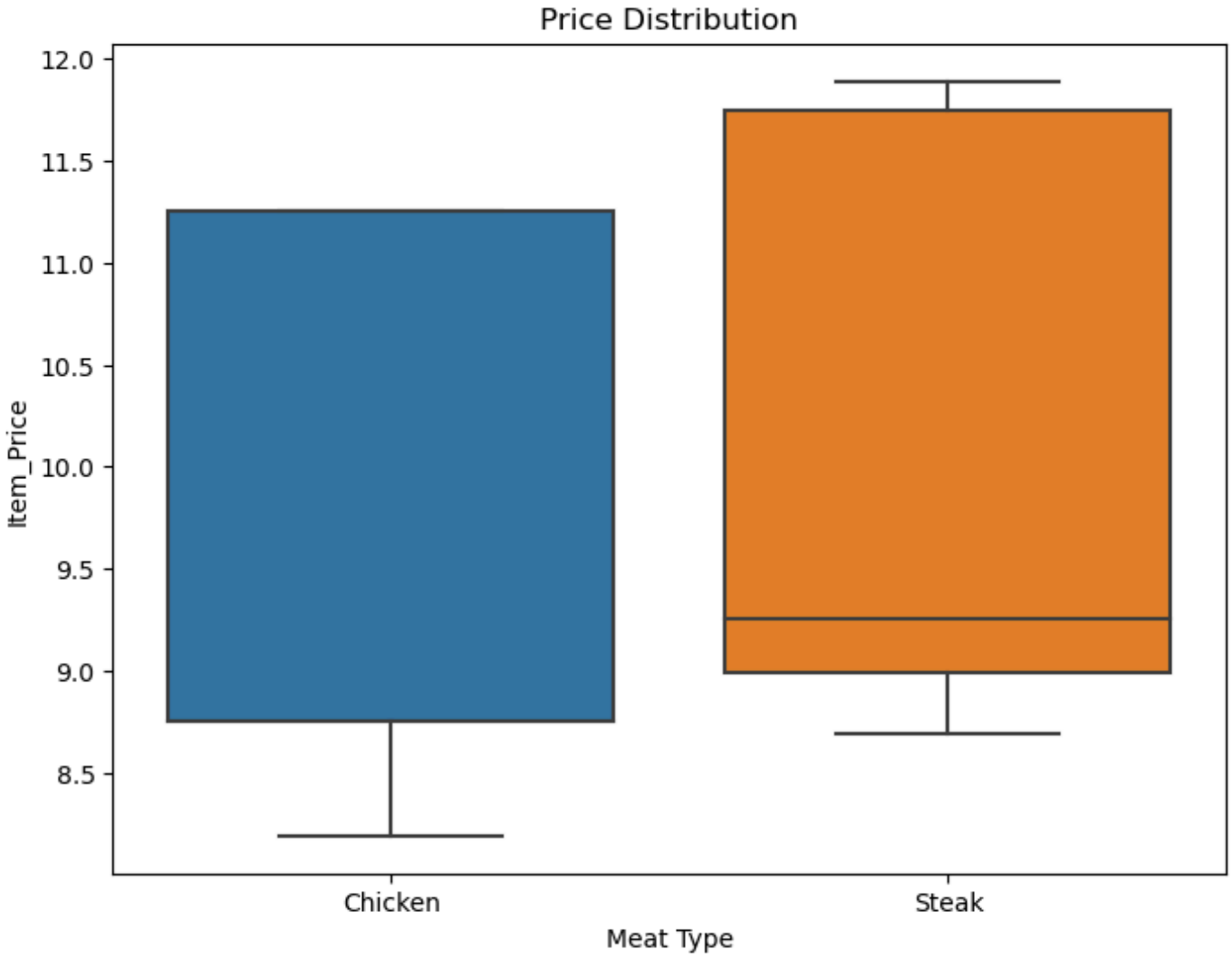



```
Chicken     1560
Steak        702
Veggie       212
Barbacoa     203
Carnitas     181
Name: Meat_Type, dtype: int64
```

In [142…

```python
def calculate_quartiles(group):
    return group.quantile([0, 0.25, 0.5, 0.75, 1])

summary = chicken_steak_df.groupby('Meat_Type')['Item_Price'].apply(calculate_quartiles).unstack
summary.columns = ['min', 'Q1', 'median', 'Q3', 'max']
summary.loc['difference'] = summary.loc['Chicken'] - summary.loc['Steak']
print(summary)

plt.figure(figsize=(8, 6))
sns.boxplot(x='Meat_Type', y='Item_Price', data=chicken_steak_df)
plt.title('Price Distribution')
plt.xlabel('Meat Type')
plt.ylabel('Item_Price')
plt.show()

#Steak definitely runs higher price, on average by 50 cents
```

```
             min    Q1  median     Q3    max
Meat_Type
Chicken     8.19  8.75    8.75  11.25  11.25
Steak       8.69  8.99    9.25  11.75  11.89
difference -0.50 -0.24   -0.50  -0.50  -0.64
```



In [148…

```
#Lastly I want to know what is the number one Item in Choice_Description. One that is most often

from wordcloud import WordCloud

all_choices = ' '.join(df['Choice_Description'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_choices)

# Display the word cloud using matplotlib
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()

#Tomatoe Salsa, Sour Creak, Black Beans, Fresh Tomatoe amongst the most choices used in the orde
```

In [146…

```
# the order with most item_price
top_expensive_orders = df.sort_values(by='Item_Price', ascending=False).head(5)
top_expensive_orders
```

Out[146]:

	Order_ID	Quantity	Item_Name	Choice_Description	Item_Price	Order_Type	Meat_Type
3598	1443	15	Chips and Fresh Tomato Salsa	NaN	44.25	Salsa	other
3480	1398	3	Carnitas Bowl	[Roasted Chili Corn Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream, Guacamole, Lettuce]]	35.25	Bowl	Carnitas
1254	511	4	Chicken Burrito	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Lettuce]]	35.00	Burrito	Chicken
3602	1443	4	Chicken Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Cheese, Sour Cream]]	35.00	Burrito	Chicken
3601	1443	3	Veggie Burrito	[Fresh Tomato Salsa, [Fajita Vegetables, Rice, Black Beans, Cheese, Sour Cream, Guacamole]]	33.75	Burrito	Veggie

Conclusion

The following was discovered:

- Chicken Bowl and Chicken Borrito are the top orders, then the Steak Borrito and Bowls
- 25.1% of the orders are Bowl and 25.4% are Burrito
- more than half of the orders are Chicken meat (1560 orders are for Chicken and 702 are for Steak)
- though the price for Steak runs on average higher by $0.50
- Given the available choices: Tomato Salsa, Sour Creak, Black Beans and Fresh Tomato are the top 4 choices
- Order_ID 1443 is for 15 items of Chips and Fresh Tomatoe Salsa

In [ ]: