Q10. Make visual representations of data using library Matplotlib and apply basic principles of data graphics to create rich analytic graphs for available datasets.

We will be analysing the following datasets mainly,

- Food
- Meal

Dataset - (Food Demand Forecasting)

https://www.kaggle.com/kannanaikkal/food-demand-forecasting

In [1]:
```
# Importing essential libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Understanding the dataset

In [2]:
```
meal_df = pd.read_csv('meal_info.csv')
meal_df.head()
```

Out[2]:

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

In [3]:
```
food_df = pd.read_csv('train.csv')
food_df.head()
```

Out[3]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage
0	1379560	1	55	1885	136.83	152.29	0	
1	1466964	1	55	1993	136.83	135.83	0	
2	1346989	1	55	2539	134.86	135.86	0	
3	1338232	1	55	2139	339.50	437.53	0	
4	1448490	1	55	2631	243.50	242.50	0	

In [4]:
```
# Merging the above two datasets
df = pd.merge(meal_df,food_df,on='meal_id')
print(df.shape)
```

```
(456548, 11)
```

Visualising the datasets

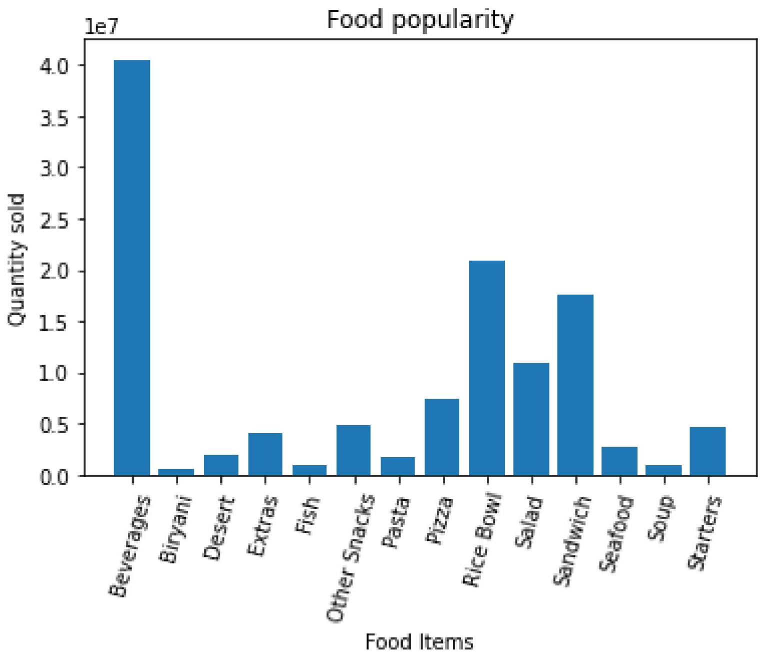
In [5]:
```
table = pd.pivot_table(data=df,index='category',values='num_orders',aggfunc=np.sum)
table
```

Out[5]:

	num_orders
category	
Beverages	40480525
Biryani	631848
Desert	1940754
Extras	3984979
Fish	871959
Other Snacks	4766293
Pasta	1637744
Pizza	7383720
Rice Bowl	20874063
Salad	10944336
Sandwich	17636782
Seafood	2715714
Soup	1039646
Starters	4649122

In [6]:
```
# Bar plot
plt.bar(table.index,table['num_orders'])
plt.xticks(rotation=75)
plt.xlabel('Food Items')
plt.ylabel('Quantity sold')
plt.title('Food popularity')
```

Out[6]: Text(0.5, 1.0, 'Food popularity')

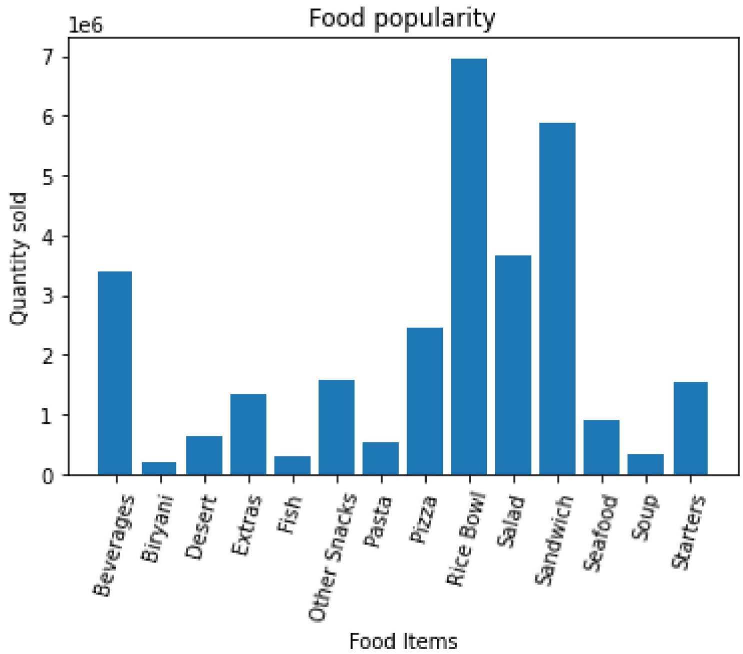


In [7]:
```
# Bar plot with unique meals
count = {}
```

```
for i in range(table.index.nunique()):
    count[table.index[i]] = table.num_orders[i]/meal_df[meal_df['category']==table.i

plt.bar(count.keys(),count.values())
plt.xticks(rotation=75)
plt.xlabel('Food Items')
plt.ylabel('Quantity sold')
plt.title('Food popularity')
```

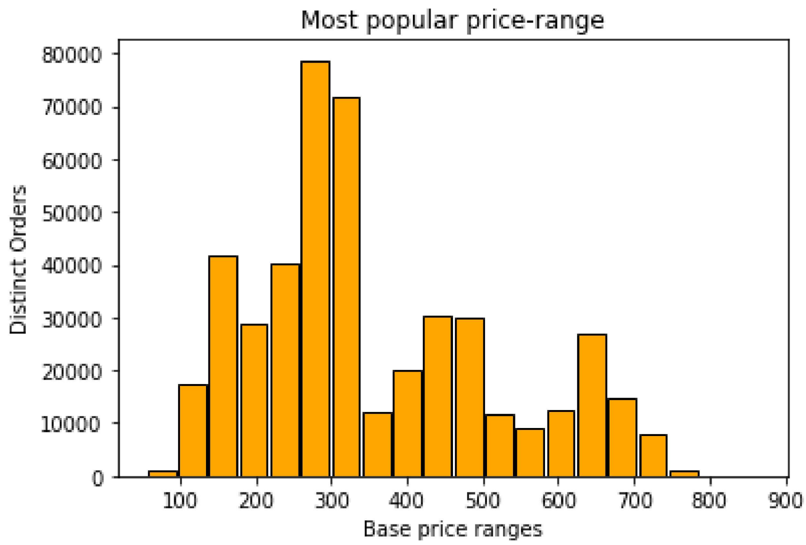
Out[7]: Text(0.5, 1.0, 'Food popularity')



Here, above plots show that **Rice Bowl** is the most popular food item.

In [8]:
```
# Histogram
plt.hist(df['base_price'],rwidth=0.9,color='orange',bins=20,edgecolor='black')
plt.xlabel('Base price ranges')
plt.ylabel('Distinct Orders')
plt.title('Most popular price-range')
```

Out[8]: Text(0.5, 1.0, 'Most popular price-range')



Therefore, there are most orders for base price range of round 300.

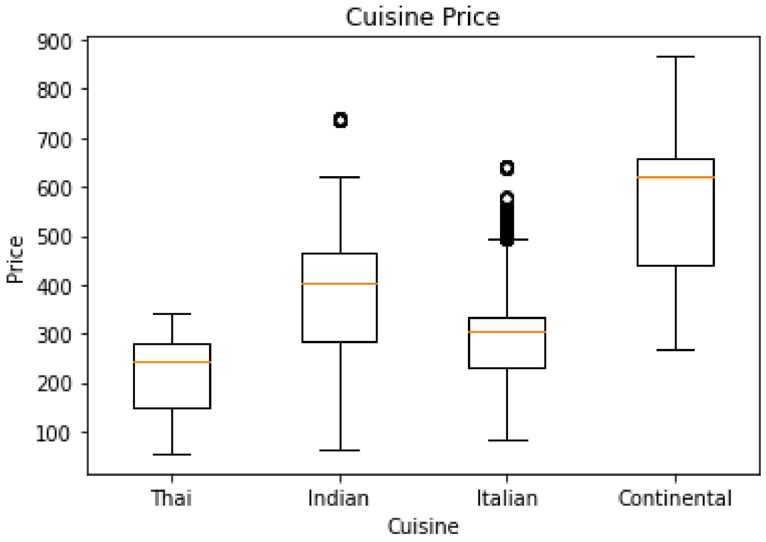
In [9]:
```
# finding base price and cuisine pair
```

```
cuisine_price = {}
for i in df['cuisine'].unique():
    cuisine_price[i] = df[df['cuisine']==i].base_price
```

In [10]:

```
#Boxplot
plt.boxplot(cuisine_price.values(),labels=cuisine_price.keys())
plt.xlabel('Cuisine')
plt.ylabel('Price')
plt.title('Cuisine Price')
```

Out[10]: Text(0.5, 1.0, 'Cuisine Price')



Therefore, we can say that Continental is the most expensive cuisine out of cuisines in the dataset. After that comes the Indian.

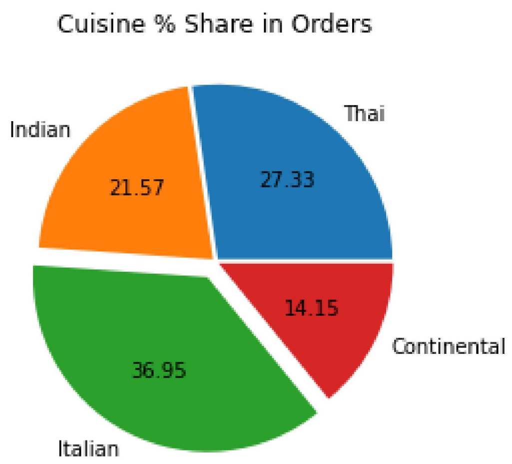
Let's visualise the share of each quisine in total number of orders.

In [11]:

```
cuisine_orders = {}
total = df['num_orders'].sum()
for i in range(df['cuisine'].nunique()):
    c = df['cuisine'].unique()[i]
    c_order = df[df['cuisine']==c]['num_orders'].sum()
    cuisine_orders[c] = c_order/total

#pie plot
plt.pie([x*100 for x in cuisine_orders.values()],labels=cuisine_orders.keys(),autopc
plt.title('Cuisine % Share in Orders')
```

Out[11]: Text(0.5, 1.0, 'Cuisine % Share in Orders')



Now, lets take another datset,

Boston Housing Dataset from sklearn

In [12]:
```
# Importing the dataset
from sklearn.datasets import load_boston
boston_dataset = load_boston()
print(boston_dataset.keys())
```

```
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])
```

In [13]:
```
# Converting it to a DataFrame
boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
boston['MEDV'] = boston_dataset.target
boston.head()
```

Out[13]:

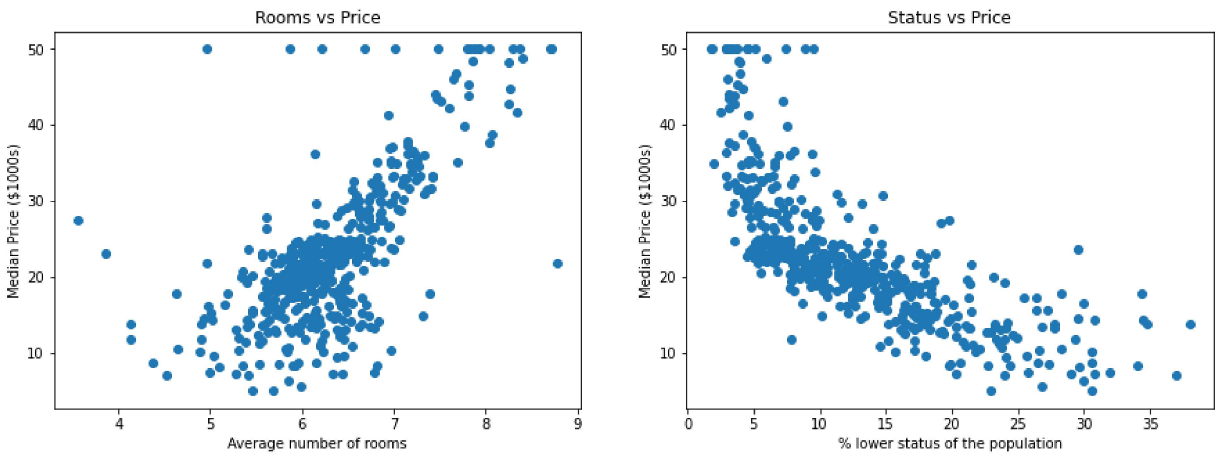
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	I
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	

In [14]:
```
fig,ax=plt.subplots(nrows=1,ncols=2,figsize=(15,5))

ax[0].scatter(boston['RM'], boston['MEDV'], marker='o')
ax[0].set_title('Rooms vs Price')
ax[0].set_xlabel('Average number of rooms')
ax[0].set_ylabel('Median Price ($1000s)')

ax[1].scatter(boston['LSTAT'], boston['MEDV'], marker='o')
ax[1].set_title('Status vs Price')
ax[1].set_xlabel('% lower status of the population')
ax[1].set_ylabel('Median Price ($1000s)')
```

Out[14]: `Text(0, 0.5, 'Median Price ($1000s)')`



Iris Dataset from sklearn

In [15]:
```
from sklearn import datasets
# load dataset
iris = datasets.load_iris()
```

```
X_iris = iris.data[:, :2]  # only take the first two features
Y_iris = iris.target
n_classes = 3

for i in range(n_classes):
    index = np.where(Y_iris == i)
    plt.scatter(X_iris[index, 0], X_iris[index, 1],
    label=iris.target_names[i])

plt.legend()
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
```

Out[15]: `Text(0, 0.5, 'sepal width (cm)')`