1. df.shape
2. df.info()
3. df.isnull().sum()
4. df.describe(include='all')
5. df['rating'].value\_counts()
6. df['order\_id'].nunique()
7. df['customer\_id'].nunique()
8. df['restaurant\_name'].nunique()
9. df['customer\_id'].value\_counts().head(3)
10. df['cuisine\_type'].nunique()
11. plt.figure(figsize = (15,5))

sns.countplot(data = df, x = 'cuisine\_type')

1. df['restaurant\_name'].value\_counts()
2. sns.histplot(data=df,x='cost\_of\_the\_order')

plt.show()

sns.boxplot(data=df,x='cost\_of\_the\_order')

plt.show()

1. df\_greater\_than\_20 = df[df['cost\_of\_the\_order']>20]

print('The number of total orders that cost above 20 dollars is:', df\_greater\_than\_20.shape[0])

percentage = (df\_greater\_than\_20.shape[0] / df.shape[0]) \* 100

print("Percentage of orders above 20 dollars:", round(percentage, 2), '%')

1. df['day\_of\_the\_week'].unique()
2. sns.countplot(data = df, x = 'day\_of\_the\_week')
3. df['rating'].unique()
4. sns.countplot(data = df, x = 'rating')
5. sns.histplot(data=df,x='food\_preparation\_time')

plt.show()

sns.boxplot(data=df,x='food\_preparation\_time')

plt.show()

1. sns.histplot(data=df,x='delivery\_time')

plt.show()

sns.boxplot(data=df,x='delivery\_time')

plt.show()

1. df\_weekend = df[df['day\_of\_the\_week'] == 'Weekend']

df\_weekend['cuisine\_type'].unique()

1. plt.figure(figsize=(15,7))

sns.boxplot(x = "cuisine\_type", y = "cost\_of\_the\_order", data = df, palette = 'PuBu')

plt.xticks(rotation = 60)

plt.show()

1. plt.figure(figsize=(15,7))

sns.boxplot(x = "cuisine\_type", y = "food\_preparation\_time", data = df, palette = 'PuBu')

plt.xticks(rotation = 60)

plt.show()

1. plt.figure(figsize=(15,7))

sns.boxplot(x = "day\_of\_the\_week", y = "delivery\_time", data = df, palette = 'PuBu')

plt.show()

1. print('The mean delivery time on weekdays is around',

round(df[df['day\_of\_the\_week'] == 'Weekday']['delivery\_time'].mean()),

'minutes')

print('The mean delivery time on weekends is around',

round(df[df['day\_of\_the\_week'] == 'Weekend']['delivery\_time'].mean()),

'minutes')

1. df.groupby(['restaurant\_name'])['cost\_of\_the\_order'].sum().sort\_values(ascending = False).head(14)
2. plt.figure(figsize=(15, 7))

sns.pointplot(x = 'rating', y = 'delivery\_time', data = df)

plt.show()

1. plt.figure(figsize=(15, 7))

sns.pointplot(x = 'rating', y = 'food\_preparation\_time', data = df)

plt.show()

1. plt.figure(figsize=(15, 7))

sns.pointplot(x = 'rating', y = 'cost\_of\_the\_order', data = df)

plt.show()

1. col\_list = ['cost\_of\_the\_order', 'food\_preparation\_time', 'delivery\_time']

plt.figure(figsize=(15, 7))

sns.heatmap(df[col\_list].corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral")

plt.show()

1. df\_rated = df[df['rating'] != 'Not given'].copy()

df\_rated['rating'] = df\_rated['rating'].astype('int')

df\_rating\_count = df\_rated.groupby(['restaurant\_name'])['rating'].count().sort\_values(ascending = False).reset\_index()

df\_rating\_count.head()

rest\_names = df\_rating\_count[df\_rating\_count['rating'] > 50]['restaurant\_name']

df\_mean\_4 = df\_rated[df\_rated['restaurant\_name'].isin(rest\_names)].copy()

df\_mean\_4.groupby(['restaurant\_name'])['rating'].mean().sort\_values(ascending = False).reset\_index().dropna()

1. def compute\_rev(x):

if x > 20:

return x\*0.25

elif x > 5:

return x\*0.15

else:

return x\*0

df['Revenue'] = df['cost\_of\_the\_order'].apply(compute\_rev)

df.head()

total\_rev = df['Revenue'].sum()

print('The net revenue is around', round(total\_rev, 2), 'dollars')

1. df['total\_time'] = df['food\_preparation\_time'] + df['delivery\_time']

df\_delivery\_greater\_than\_60 = df[df['total\_time']>60]

print('The number of total orders whose delivery time is more than 60 minutes is:', df\_delivery\_greater\_than\_60.shape[0])

percentage = (df\_delivery\_greater\_than\_60.shape[0] / df.shape[0]) \* 100

print("Percentage of orders delivered in more than 60 minutes:", round(percentage, 2), '%')