```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
```

```
df=pd.read_csv("/kaggle/input/tipscsv/tips.csv")
```

```
df.head()
```

	total_bill	tip	sex	smoker	day	time	size	price_per_person	Payer Name	CC Numbe
0	16.99	1.01	Female	No	Sun	Dinner	2	8.49	Christy Cunningham	356032516860341
1	10.34	1.66	Male	No	Sun	Dinner	3	3.45	Douglas Tucker	447807137977923
2	21.01	3.50	Male	No	Sun	Dinner	3	7.00	Travis Walters	601181211297132
3	23.68	3.31	Male	No	Sun	Dinner	2	11.84	Nathaniel Harris	467613764768599
4	24.59	3.61	Female	No	Sun	Dinner	4	6.15	Tonya Carter	483273261863722

```
df.shape
```

```
(244, 11)
```

```
df.info
```

```
<bound method DataFrame.info of      total_bill   tip     sex smoker   day    time  size  
price_per_person  \
0         16.99  1.01  Female     No   Sun  Dinner     2              8.49   
1         10.34  1.66    Male     No   Sun  Dinner     3              3.45   
2         21.01  3.50    Male     No   Sun  Dinner     3              7.00   
3         23.68  3.31    Male     No   Sun  Dinner     2             11.84   
4         24.59  3.61  Female     No   Sun  Dinner     4              6.15   
..          ...   ...     ...    ...   ...     ...   ...               ...   
239       29.03  5.92    Male     No   Sat  Dinner     3              9.68   
240       27.18  2.00  Female    Yes   Sat  Dinner     2             13.59   
241       22.67  2.00    Male    Yes   Sat  Dinner     2             11.34   
242       17.82  1.75    Male     No   Sat  Dinner     2              8.91   
243       18.78  3.00  Female     No  Thur  Dinner     2              9.39   

             Payer Name         CC Number Payment ID  
0    Christy Cunningham  3560325168603410    Sun2959  
1        Douglas Tucker  4478071379779230    Sun4608  
2        Travis Walters  6011812112971322    Sun4458  
3      Nathaniel Harris  4676137647685994    Sun5260  
4          Tonya Carter  4832732618637221    Sun2251  
..                  ...               ...        ...  
239       Michael Avila  5296068606052842    Sat2657  
240      Monica Sanders  3506806155565404    Sat1766  
241          Keith Wong  6011891618747196    Sat3880  
```

```
242        Dennis Dixon     4375220550950      Sat17
243     Michelle Hardin  3511451626698139    Thur672

[244 rows x 11 columns]>
```

```
df.dtypes
```

```
total_bill          float64
tip                 float64
sex                   int64
smoker                int64
day                   int64
time                  int64
size                  int64
price_per_person    float64
Payer Name            int64
CC Number             int64
Payment ID            int64
dtype: object
```

```
df.isnull().sum()
```

```
total_bill          0
tip                 0
sex                 0
smoker              0
day                 0
time                0
size                0
price_per_person    0
Payer Name          0
CC Number           0
Payment ID          0
dtype: int64
```

```
# Display the summary statistics of the dataset
print("\nSummary statistics of the dataset:")
df.describe(include='all')
```

```
Summary statistics of the dataset:
```

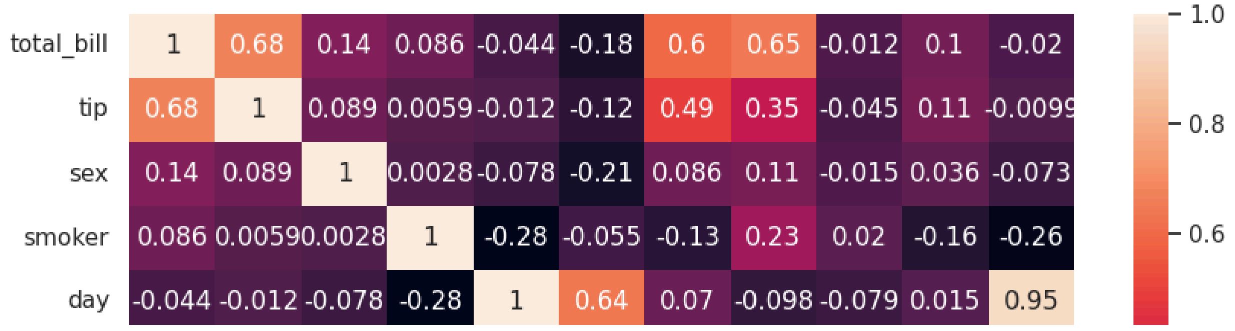
	total_bill	tip	sex	smoker	day	time	size	price_per_person	Payer Name	
count	244.000000	244.000000	244	244	244	244	244.000000	244.000000	244	2
unique	NaN	NaN	2	2	4	2	NaN	NaN	244	
top	NaN	NaN	Male	No	Sat	Dinner	NaN	NaN	Christy Cunningham	
freq	NaN	NaN	157	151	87	176	NaN	NaN	1	
mean	19.785943	2.998279	NaN	NaN	NaN	NaN	2.569672	7.888197	NaN	2
std	8.902412	1.383638	NaN	NaN	NaN	NaN	0.951100	2.914234	NaN	2
min	3.070000	1.000000	NaN	NaN	NaN	NaN	1.000000	2.880000	NaN	6
25%	13.347500	2.000000	NaN	NaN	NaN	NaN	2.000000	5.800000	NaN	3
50%	17.795000	2.900000	NaN	NaN	NaN	NaN	2.000000	7.255000	NaN	3
75%	24.127500	3.562500	NaN	NaN	NaN	NaN	3.000000	9.390000	NaN	4
max	50.810000	10.000000	NaN	NaN	NaN	NaN	6.000000	20.270000	NaN	6

```
df.corr()
```

	total_bill	tip	sex	smoker	day	time	size	price_per_per
total_bill	1.000000	0.675734	0.144877	0.085721	-0.043550	-0.183118	0.598315	0.64
tip	0.675734	1.000000	0.088862	0.005929	-0.011548	-0.121629	0.489299	0.34
sex	0.144877	0.088862	1.000000	0.002816	-0.078292	-0.205231	0.086195	0.10
smoker	0.085721	0.005929	0.002816	1.000000	-0.282721	-0.054921	-0.133178	0.22
day	-0.043550	-0.011548	-0.078292	-0.282721	1.000000	0.638019	0.069510	-0.09
time	-0.183118	-0.121629	-0.205231	-0.054921	0.638019	1.000000	-0.103411	-0.12
size	0.598315	0.489299	0.086195	-0.133178	0.069510	-0.103411	1.000000	-0.17
price_per_person	0.647554	0.347405	0.108485	0.229916	-0.097593	-0.122258	-0.175359	1.00
Payer Name	-0.012311	-0.044563	-0.014759	0.019588	-0.079234	-0.021932	0.022345	-0.01
CC Number	0.104576	0.110857	0.035575	-0.158763	0.015021	-0.038887	-0.030239	0.13
Payment ID	-0.019868	-0.009931	-0.073021	-0.260056	0.949678	0.618591	0.097901	-0.09

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(),annot=True)
```

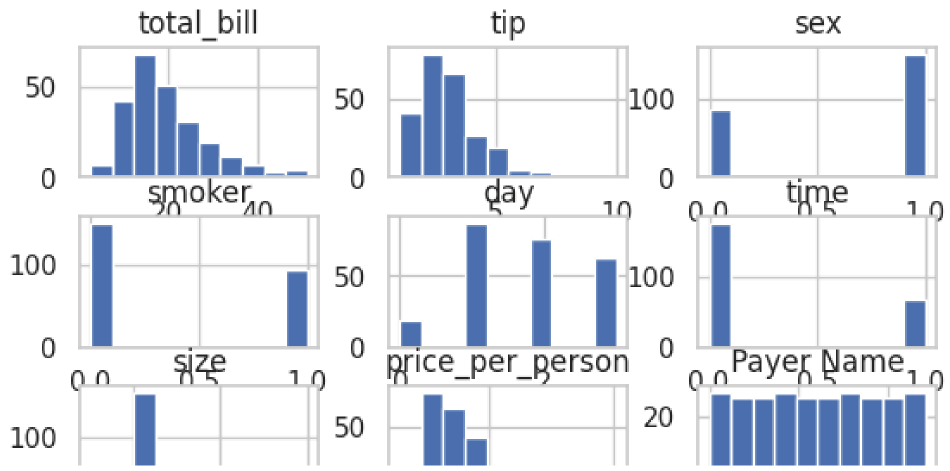
```
<Axes: >
```



total_bill	1	0.68	0.14	0.086	-0.044	-0.18	0.6	0.65	-0.012	0.1	-0.02
tip	0.68	1	0.089	0.0059	-0.012	-0.12	0.49	0.35	-0.045	0.11	-0.0099
sex	0.14	0.089	1	0.0028	-0.078	-0.21	0.086	0.11	-0.015	0.036	-0.073
smoker	0.086	0.0059	0.0028	1	-0.28	-0.055	-0.13	0.23	0.02	-0.16	-0.26
day	-0.044	-0.012	-0.078	-0.28	1	0.64	0.07	-0.098	-0.079	0.015	0.95

```
df.hist()
plt.show
plt.figure(figsize=(20,12))
```

```
<Figure size 2000x1200 with 0 Axes>
```

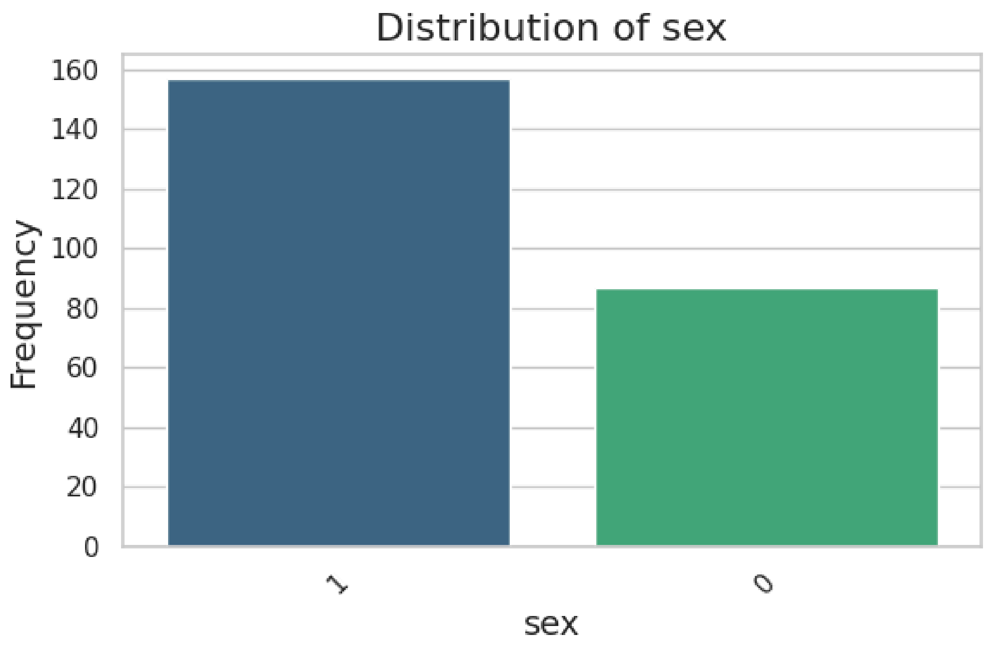


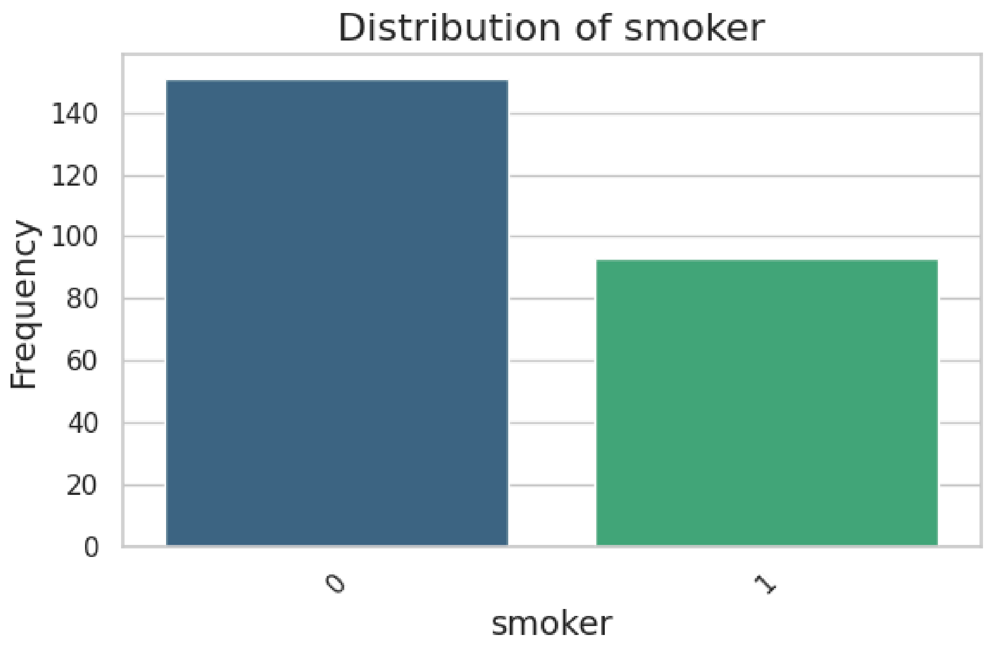
```
# Identify the categorical columns in the dataset
categorical_columns = df.select_dtypes(include=['object', 'category']).columns.tolist()
categorical_columns
```

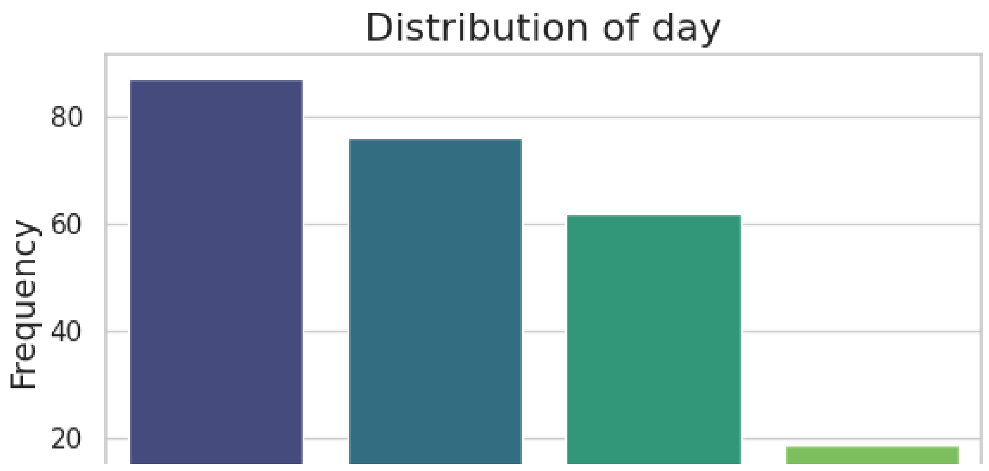
```
['sex', 'smoker', 'day', 'time', 'Payer Name', 'Payment ID']
```

```
# Set the style for the plots
sns.set(style="whitegrid")

# Loop through each categorical column and create a bar plot
for column in categorical_columns:
    plt.figure(figsize=(6,4))
    sns.countplot(data=df, x=column, order=df[column].value_counts().index, palette='viridis')
    plt.title(f'Distribution of {column}', fontsize=16)
    plt.xlabel(column, fontsize=14)
    plt.ylabel('Frequency', fontsize=14)
    plt.xticks(rotation=45)  # Rotate x labels if necessary
    plt.tight_layout()
    plt.show()
```







```
# Initialize the LabelEncoder
le = LabelEncoder()
```

```
# Loop through each categorical column and apply LabelEncoder
for column in categorical_columns:
    df[column] = le.fit_transform(df[column])

# Display the first few rows of the transformed dataframe
print("Transformed dataframe with numeric values for categorical variables:")
df.head()
```

```
Transformed dataframe with numeric values for categorical variables:
```

	total_bill	tip	sex	smoker	day	time	size	price_per_person	Payer Name	CC Number	Payme
0	16.99	1.01	0	0	2	0	2	8.49	40	3560325168603410	1
1	10.34	1.66	1	0	2	0	3	3.45	58	4478071379779230	1
2	21.01	3.50	1	0	2	0	3	7.00	233	6011812112971322	1
3	23.68	3.31	1	0	2	0	2	11.84	180	4676137647685994	1
4	24.59	3.61	0	0	2	0	4	6.15	230	4832732618637221	1

```
# Use pd.get_dummies() to apply One-Hot Encoding to the categorical variables
df_one_hot_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)

# Display the first few rows of the transformed dataframe
print("Transformed dataframe with One-Hot Encoding:")
print(df_one_hot_encoded.head())

# You can also check the shapes of both dataframes to compare
print("\nShape of One-Hot Encoded DataFrame:")
df_one_hot_encoded.shape
```

```
Transformed dataframe with One-Hot Encoding:
   total_bill   tip  size  price_per_person         CC Number  sex_1  \
0       16.99  1.01     2              8.49  3560325168603410  False   
1       10.34  1.66     3              3.45  4478071379779230   True   
2       21.01  3.50     3              7.00  6011812112971322   True   
3       23.68  3.31     2             11.84  4676137647685994   True   
4       24.59  3.61     4              6.15  4832732618637221  False   

   smoker_1  day_1  day_2  day_3  ...  Payment ID_233  Payment ID_234  \
0     False  False   True  False  ...           False           False   
1     False  False   True  False  ...           False           False   
2     False  False   True  False  ...           False           False   
3     False  False   True  False  ...           False           False   
4     False  False   True  False  ...           False           False   

   Payment ID_235  Payment ID_236  Payment ID_237  Payment ID_238  \
0           False           False           False           False   
1           False           False           False           False   
2           False           False           False           False   
3           False           False           False           False   
4           False           False           False           False   

   Payment ID_239  Payment ID_240  Payment ID_241  Payment ID_242
```