EDA.NEW_DS

September 6, 2024

1 EDA

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

[2]:
```
df=pd.read_csv("tips.csv")
```

[3]:
```
df.head(5)
```

[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

[4]:
```
df.tail(5)
```

[4]:

	total_bill	tip	sex	smoker	day	time	size
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    object
```

```
 3   smoker      244 non-null    object
 4   day         244 non-null    object
 5   time        244 non-null    object
 6   size        244 non-null    int64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.5+ KB
```

[6]:
```
df.isna().sum()
```

[6]:
```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

[7]:
```
df.describe()
```

[7]:
```
        total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

[8]:
```
df.describe(include='object')
```

[8]:
```
          sex smoker  day    time
count     244    244  244     244
unique      2      2    4       2
top      Male     No  Sat  Dinner
freq      157    151   87     176
```

[9]:
```
df.columns
```

[9]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[10]:
```
df['sex'].value_counts()
```

[10]:
```
sex
Male      157
```

```
Female     87
Name: count, dtype: int64
```

[11]:
```
df_female=df[df['sex']=='Female']
```

[12]:
```
df_female.head(5)
```

[12]:
```
     total_bill   tip     sex smoker  day    time  size
0         16.99  1.01  Female     No  Sun  Dinner     2
4         24.59  3.61  Female     No  Sun  Dinner     4
11        35.26  5.00  Female     No  Sun  Dinner     4
14        14.83  3.02  Female     No  Sun  Dinner     2
16        10.33  1.67  Female     No  Sun  Dinner     3
```

[13]:
```
df[df['sex']=='Male']
```

[13]:
```
      total_bill   tip   sex smoker  day    time  size
1          10.34  1.66  Male     No  Sun  Dinner     3
2          21.01  3.50  Male     No  Sun  Dinner     3
3          23.68  3.31  Male     No  Sun  Dinner     2
5          25.29  4.71  Male     No  Sun  Dinner     4
6           8.77  2.00  Male     No  Sun  Dinner     2
..           …   …   …    …  …      …   …
236        12.60  1.00  Male    Yes  Sat  Dinner     2
237        32.83  1.17  Male    Yes  Sat  Dinner     2
239        29.03  5.92  Male     No  Sat  Dinner     3
241        22.67  2.00  Male    Yes  Sat  Dinner     2
242        17.82  1.75  Male     No  Sat  Dinner     2

[157 rows x 7 columns]
```

[14]:
```
df.columns
```

[14]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[15]:
```
df.head(5)
```

[15]:
```
    total_bill   tip     sex smoker  day    time  size
0        16.99  1.01  Female     No  Sun  Dinner     2
1        10.34  1.66    Male     No  Sun  Dinner     3
2        21.01  3.50    Male     No  Sun  Dinner     3
3        23.68  3.31    Male     No  Sun  Dinner     2
4        24.59  3.61  Female     No  Sun  Dinner     4
```

[16]:
```
df.tail(5)
```

[16]:

	total_bill	tip	sex	smoker	day	time	size
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[17]:
```
df['smoker'].value_counts()
```

[17]:
```
smoker
No     151
Yes     93
Name: count, dtype: int64
```

[18]:
```
df[df['smoker']=='No']
```

[18]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
..	…	…	…	…	…	…	…
235	10.07	1.25	Male	No	Sat	Dinner	2
238	35.83	4.67	Female	No	Sat	Dinner	3
239	29.03	5.92	Male	No	Sat	Dinner	3
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[151 rows x 7 columns]

[19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    object
 3   smoker      244 non-null    object
 4   day         244 non-null    object
 5   time        244 non-null    object
 6   size        244 non-null    int64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.5+ KB
```

[20]:
```
df.describe()
```

[20]:
```
       total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

[21]:
```
df.describe(include='object')
```

[21]:
```
          sex smoker  day    time
count     244    244  244     244
unique      2      2    4       2
top      Male     No  Sat  Dinner
freq      157    151   87     176
```

[22]:
```
df.isnull().sum()
```

[22]:
```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

[23]:
```
df.isna().sum()
```

[23]:
```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

[24]:
```
df.columns
```

[24]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[25]:
```
df['tip'].mean().round(2)
```

[25]:
```
3.0
```

[26]:
```
df.describe()
```

[26]:
```
       total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

[27]:
```
df['tip'].min()
```

[27]:
```
1.0
```

[28]:
```
df['tip'].var()
```

[28]:
```
1.9144546380624725
```

[29]:
```
df.describe(include='object')
```

[29]:
```
          sex smoker  day    time
count     244    244  244     244
unique      2      2    4       2
top      Male     No  Sat  Dinner
freq      157    151   87     176
```

[30]:
```
df['sex'].value_counts()
```

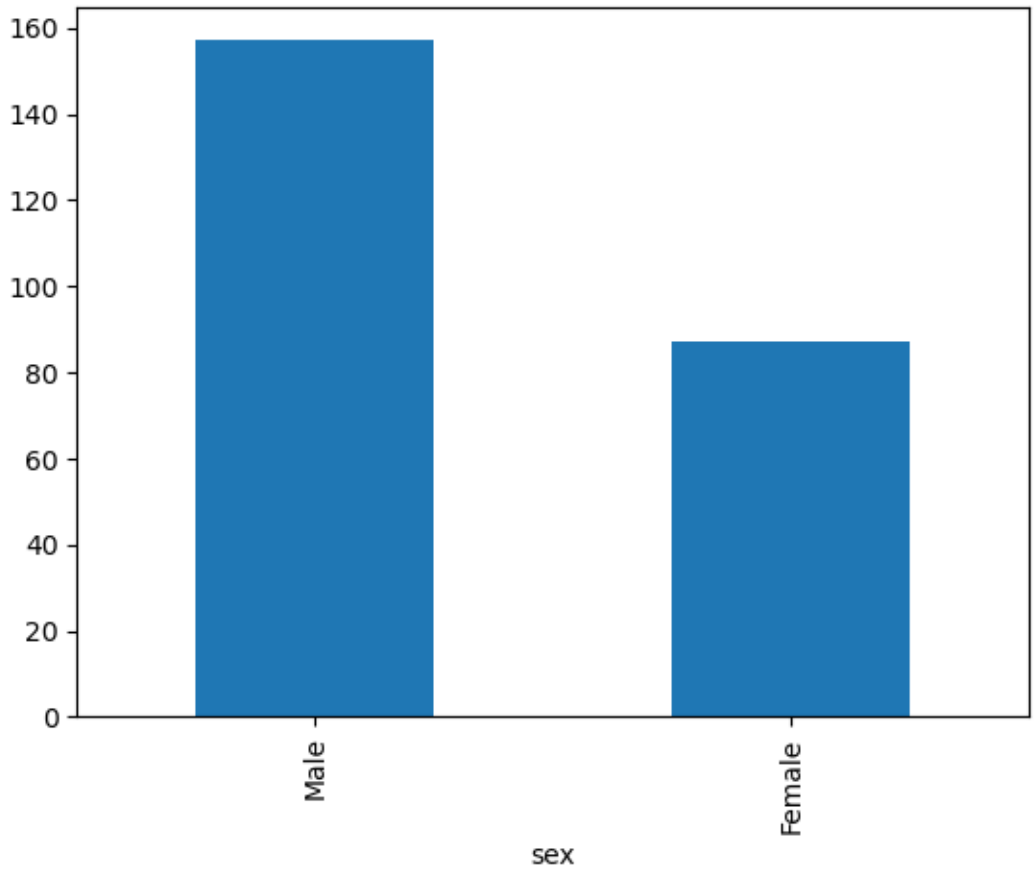
[30]:
```
sex
Male      157
Female     87
Name: count, dtype: int64
```

[31]:
```
df['smoker'].value_counts()
```

[31]:
```
smoker
No     151
Yes     93
Name: count, dtype: int64
```

[32]:
```
df['sex'].value_counts().plot.bar()
```

[32]: <Axes: xlabel='sex'>



[33]:
```
df['sex'].value_counts()
```

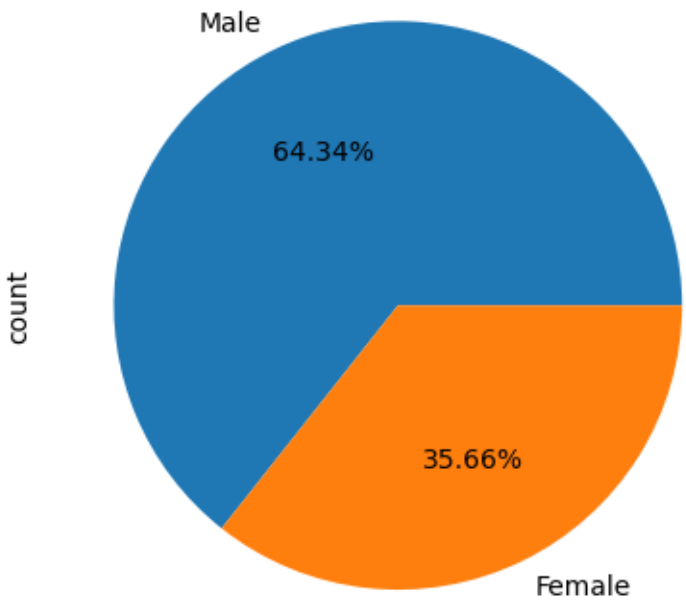
[33]:
```
sex
Male      157
Female     87
Name: count, dtype: int64
```

[34]:
```
df.shape
```

[34]: (244, 7)

[35]:
```
(df['sex'].value_counts()/df.shape[0]*100).plot.pie(autopct="%1.2f%%")
```

[35]: <Axes: ylabel='count'>



[36]:
```
df['day'].value_counts()
```

[36]:
```
day
Sat     87
Sun     76
Thur    62
Fri     19
Name: count, dtype: int64
```

[37]:
```
df['day'].unique()
```

[37]:
```
array(['Sun', 'Sat', 'Thur', 'Fri'], dtype=object)
```

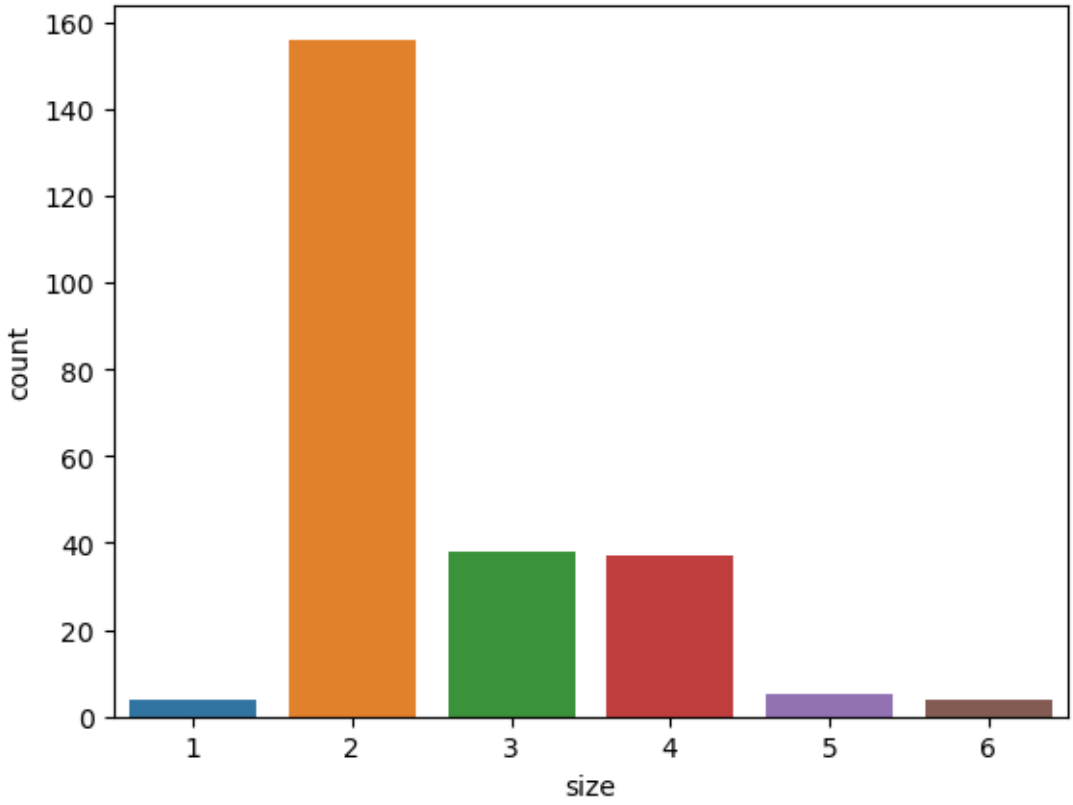
[38]:
```
df.nunique()
```

[38]:
```
total_bill    229
tip           123
sex             2
smoker          2
day             4
time            2
size            6
dtype: int64
```

[39]:
```
df.columns
```

[39]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[40]:
```
sns.countplot(df,x="size")
plt.show()
```

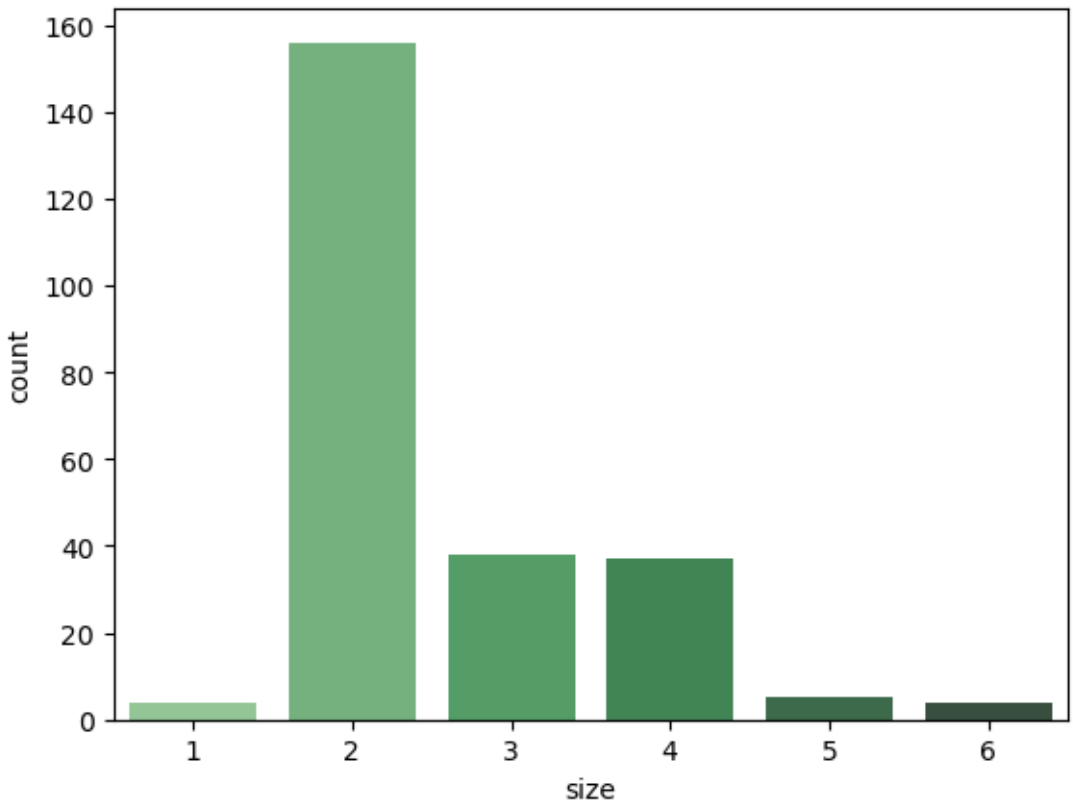


[41]:
```
df.columns
```

[41]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[42]:
```
sns.countplot(x="size",data=df,palette="Greens_d")
```

[42]:
```
<Axes: xlabel='size', ylabel='count'>
```

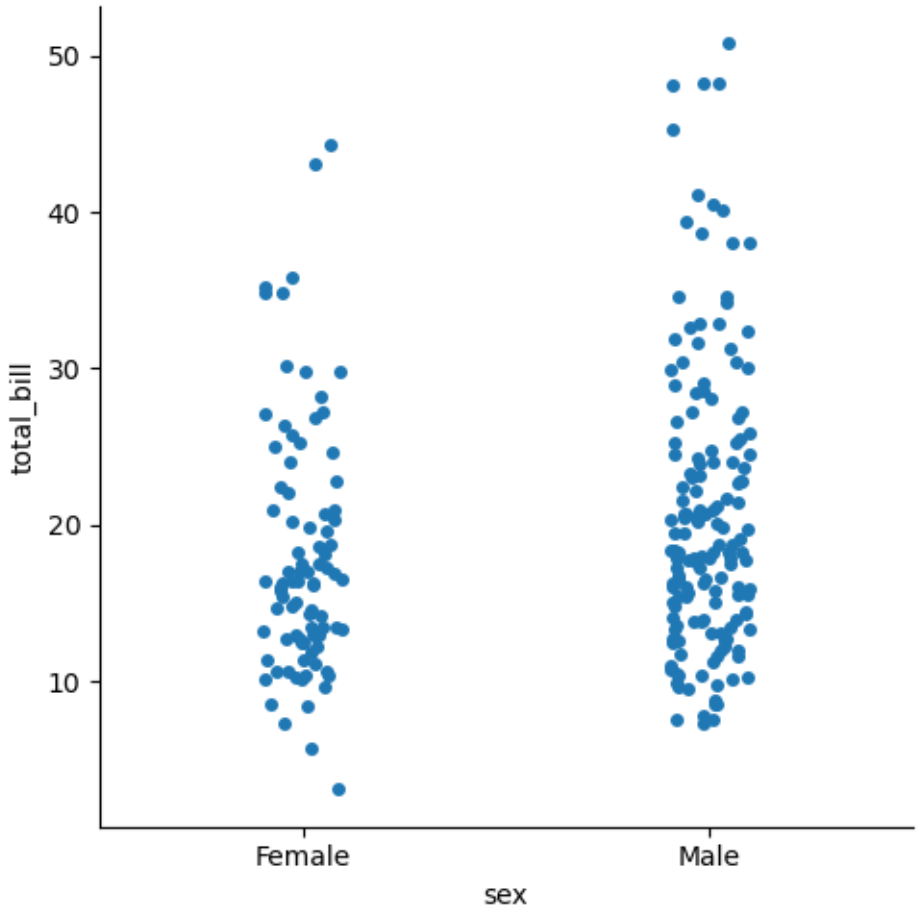


[43]:
```
df.columns
```

[43]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

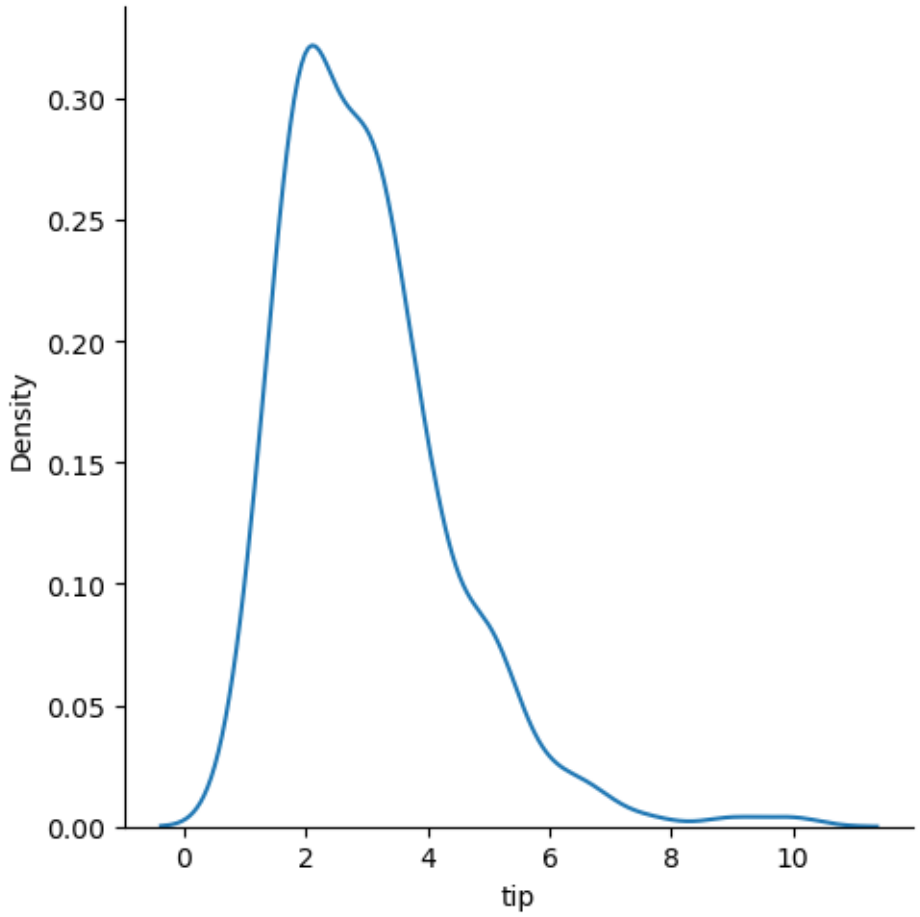
[44]:
```
sns.catplot(df,x='sex',y='total_bill')
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```



[45]:
```
sns.displot(df,x='tip',kind='kde')
plt.show()
```

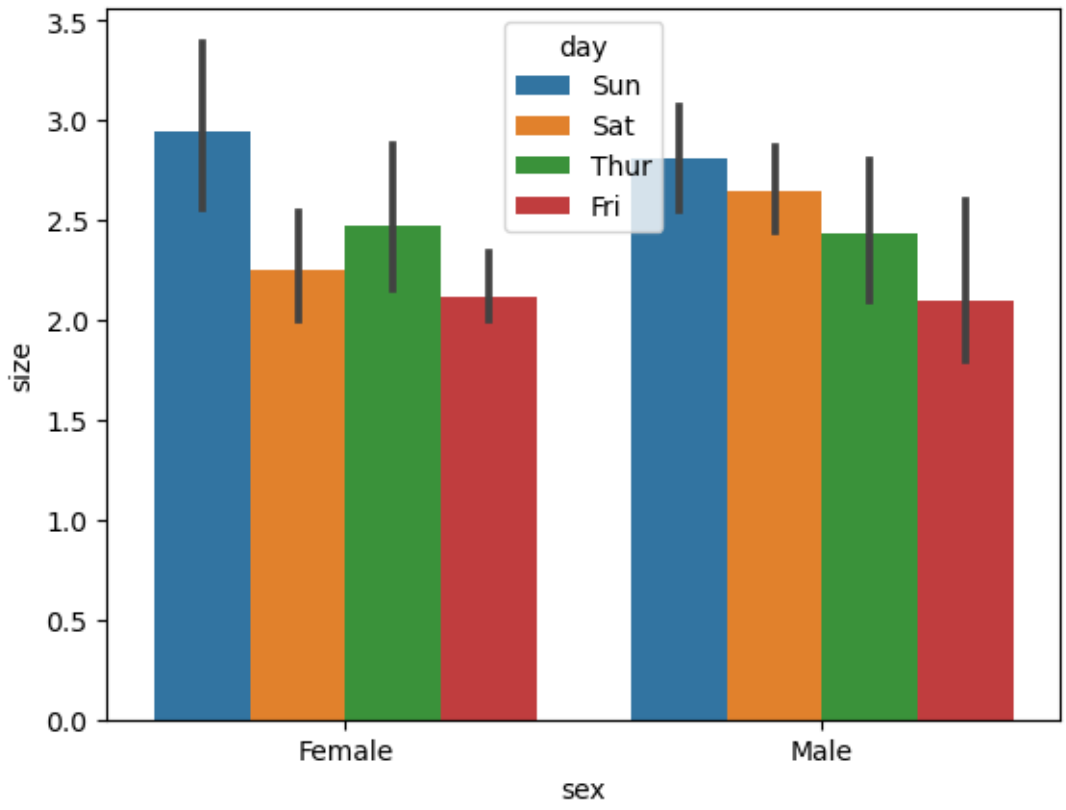
```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```



[46]:
```
df.columns
```

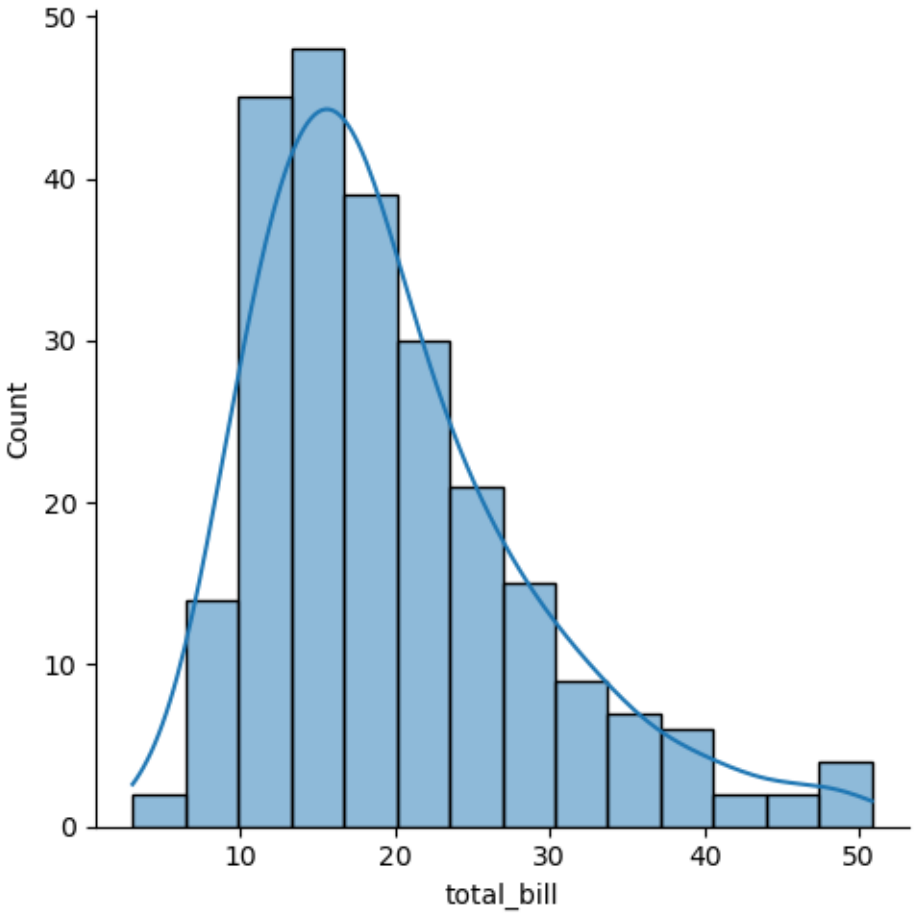
[46]:
```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'],
dtype='object')
```

[47]:
```
sns.barplot(x="sex",
                y="size",
                hue="day",
                data=df)
plt.show()
```



[48]:
```
sns.displot( data=df["total_bill"], kde=True )
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```

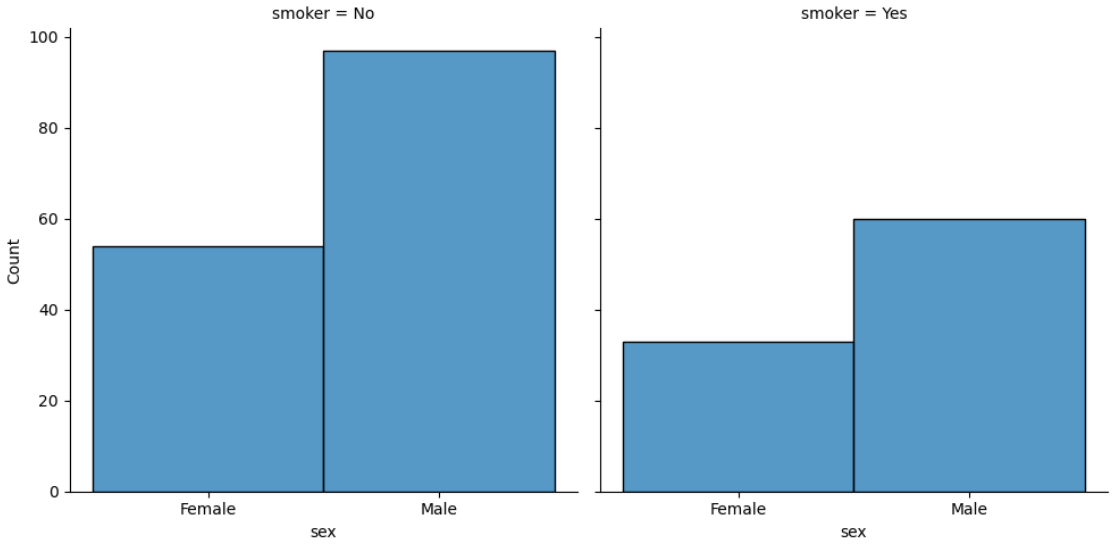


[49]:
```
import plotly.express as px
```

[50]:
```
fig = px.histogram(df, x="total_bill", nbins=30,barmode='relative')
fig.update_layout(
    title="Distribution of Total Bill Amount",
    xaxis_title="tip",
    yaxis_title="Frequency",
)
```

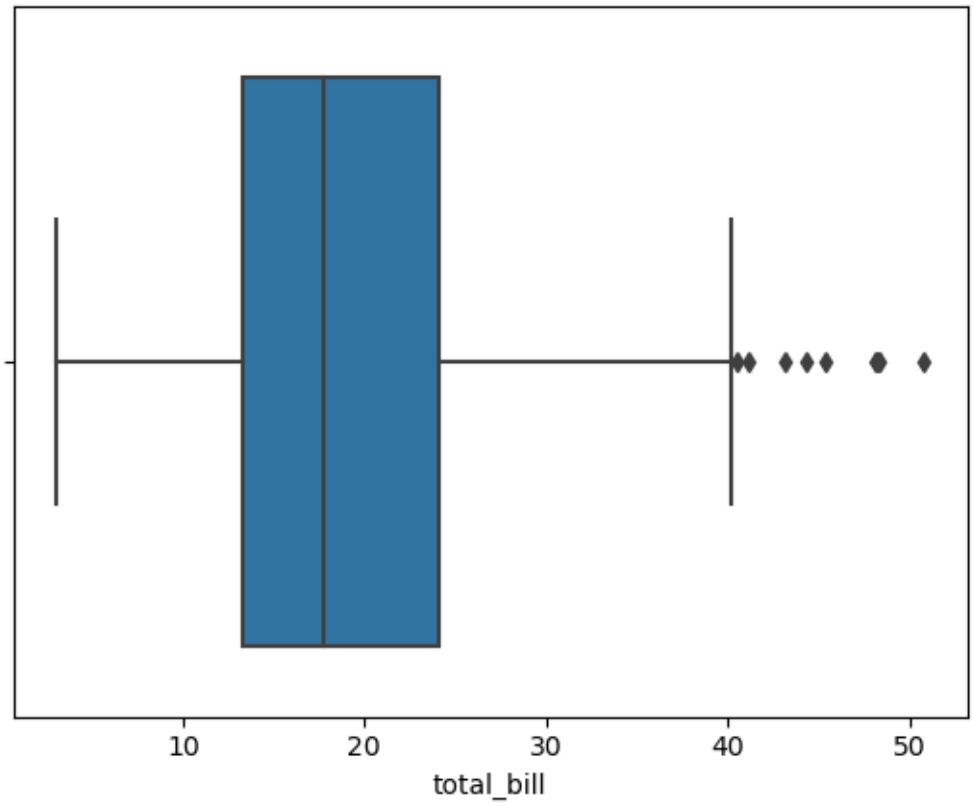
[51]:
```
sns.displot(df,x='sex',col='smoker')
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



[52]:
```
sns.boxplot(x=df['total_bill'])
```

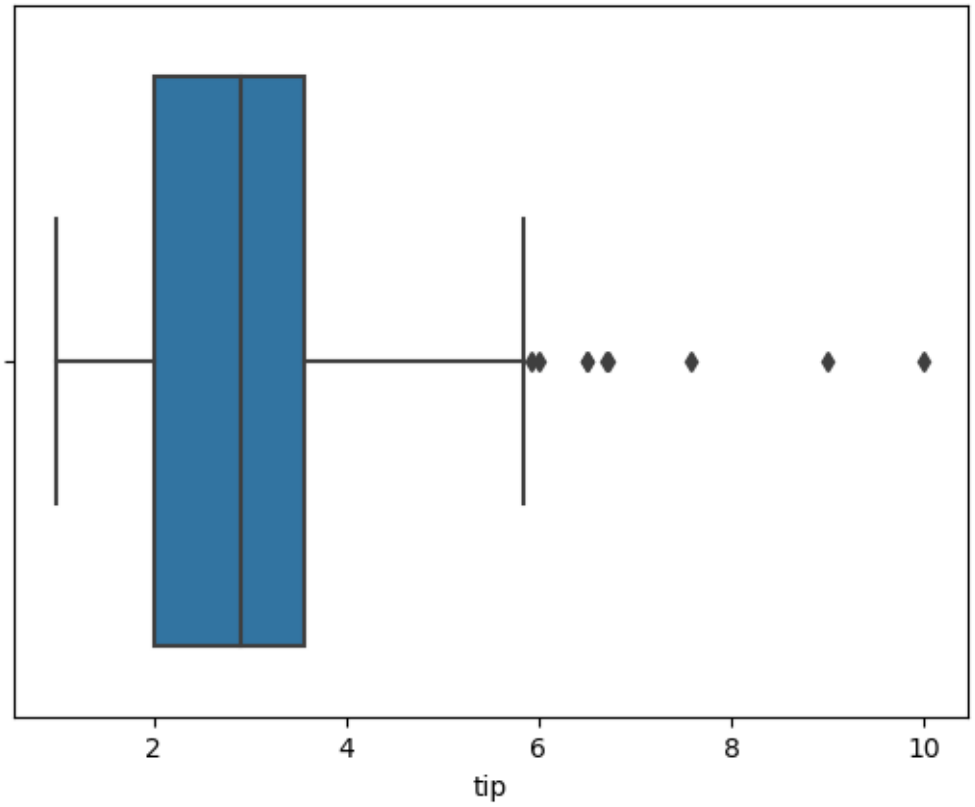
[52]: <Axes: xlabel='total_bill'>



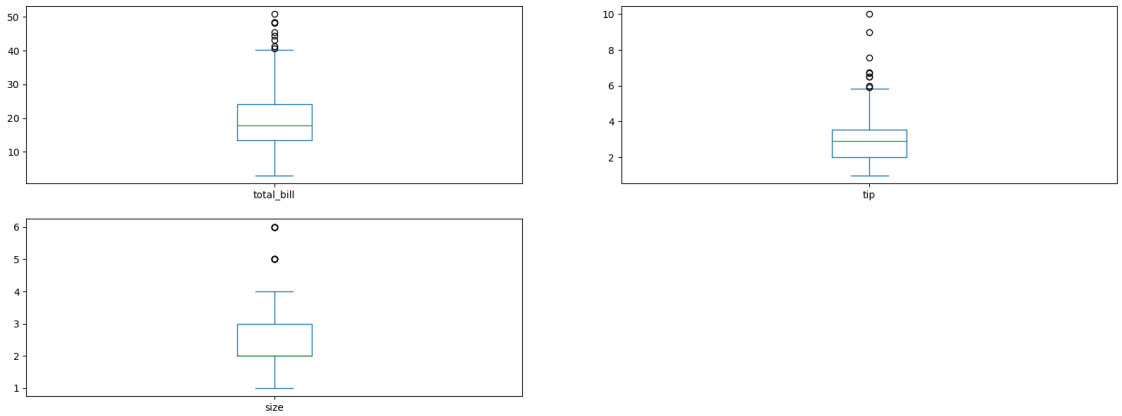
[53]:
```
import plotly.express as px
df = px.data.tips()
fig = px.box(df, y="total_bill")
fig.show()
```

[54]:
```
sns.boxplot(x=df['tip'])
```

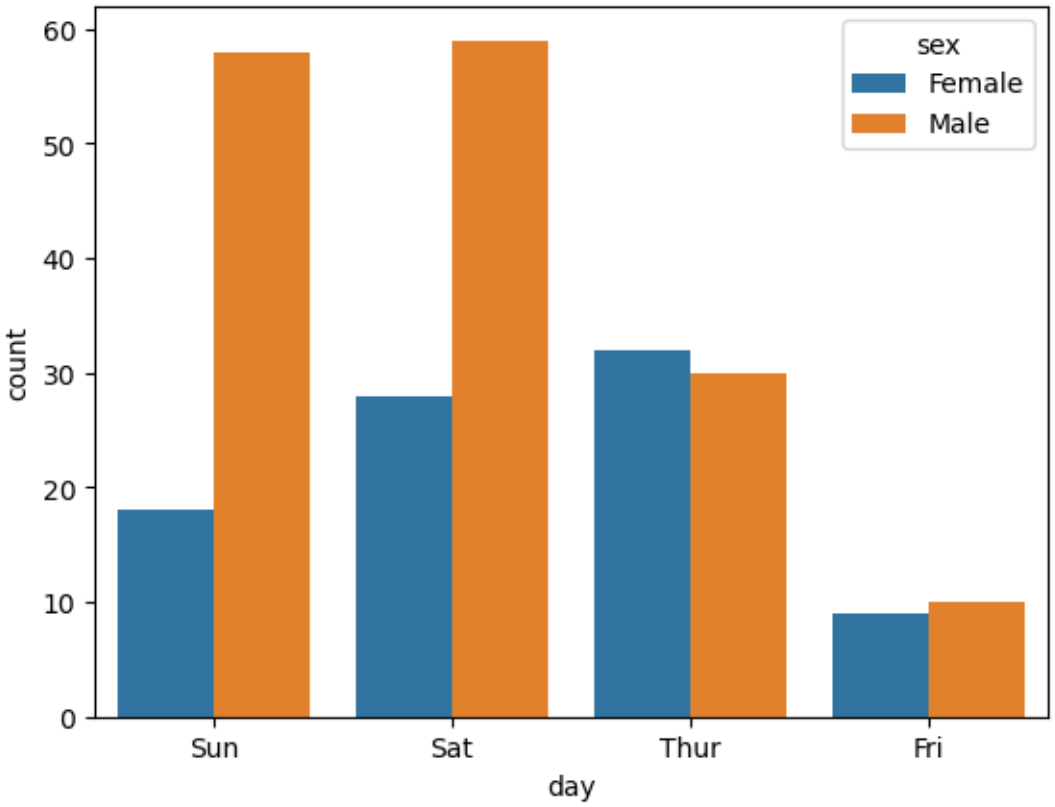
[54]:
```
<Axes: xlabel='tip'>
```



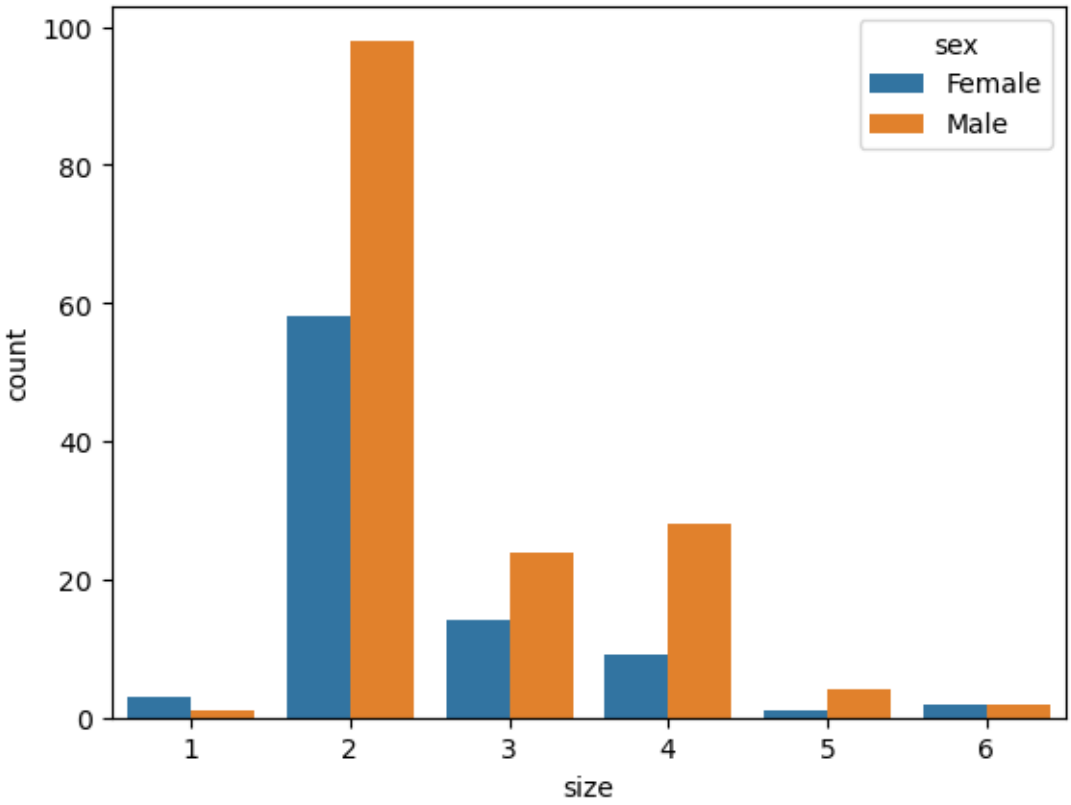
[55]:
```
df.plot(kind="box",subplots=True,figsize=(20,15),layout=(4,2))
plt.show()
```



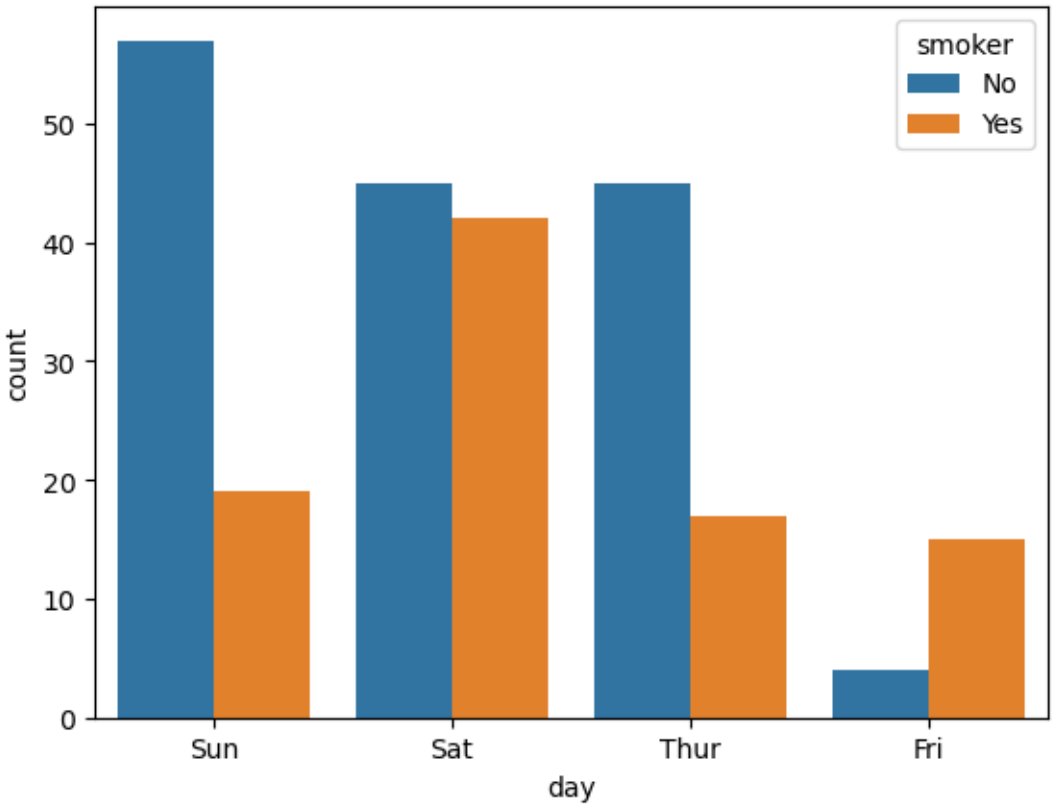
[56]:
```
sns.countplot(x='day',data=df,hue='sex')
plt.show()
```



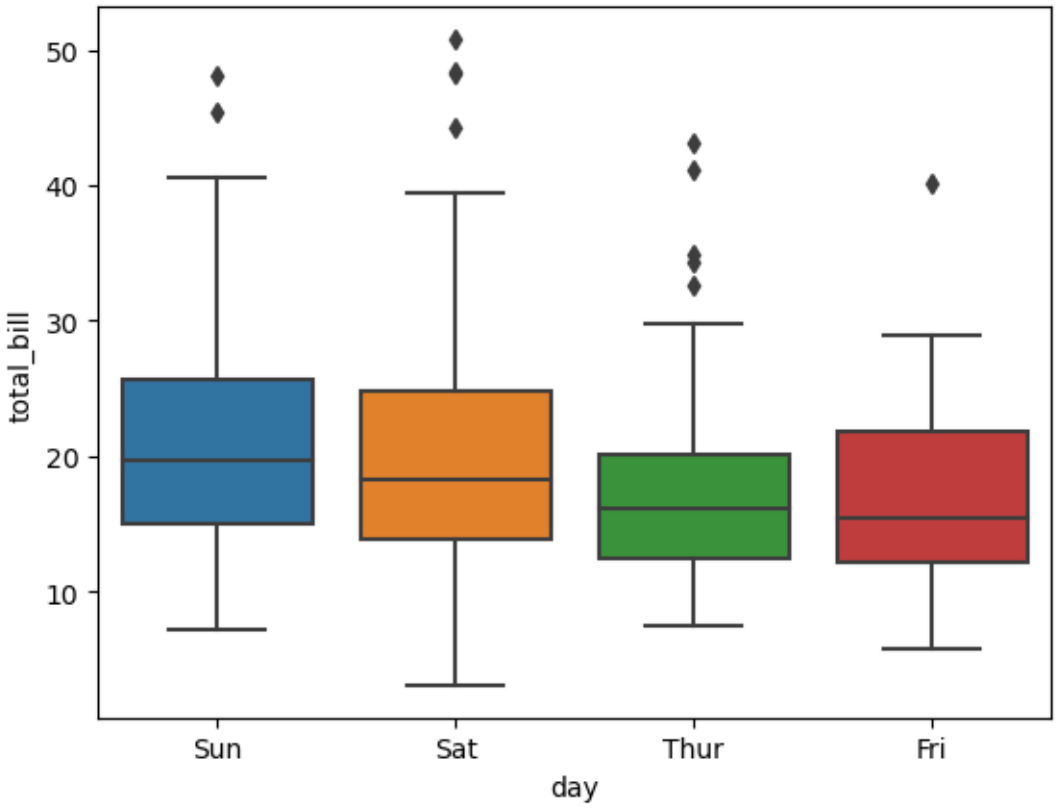
[57]:
```
sns.countplot(x='size',data=df,hue='sex')
plt.show()
```



[58]:
```
sns.countplot(x='day',data=df,hue='smoker')
plt.show()
```



[59]:
```
sns.boxplot(x='day',y='total_bill',data=df)
plt.show()
```



[60]:
```
q1=df['total_bill'].quantile(0.25)
q2=df['total_bill'].quantile(0.5)
q3=df['total_bill'].quantile(0.75)
```

[61]:
```
print("Q1:",q1)
print("Q2:",q2)
print("Q3:",q3)
```

```
Q1: 13.3475
Q2: 17.795
Q3: 24.127499999999998
```

[62]:
```
IQR=q3-q1
IQR
```

[62]: 10.779999999999998

[63]:
```
UL=q3+(IQR)*(1.5)
UL
```

[63]: 40.29749999999999

[64]:
```
LL=q1-(IQR)*(1.5)
LL
```

[64]:
```
-2.8224999999999945
```

[65]:
```
df['total_bill'].max()
```

[65]:
```
50.81
```

[66]:
```
df_num= df.select_dtypes(include=['number'])
```

[67]:
```
df_num
```

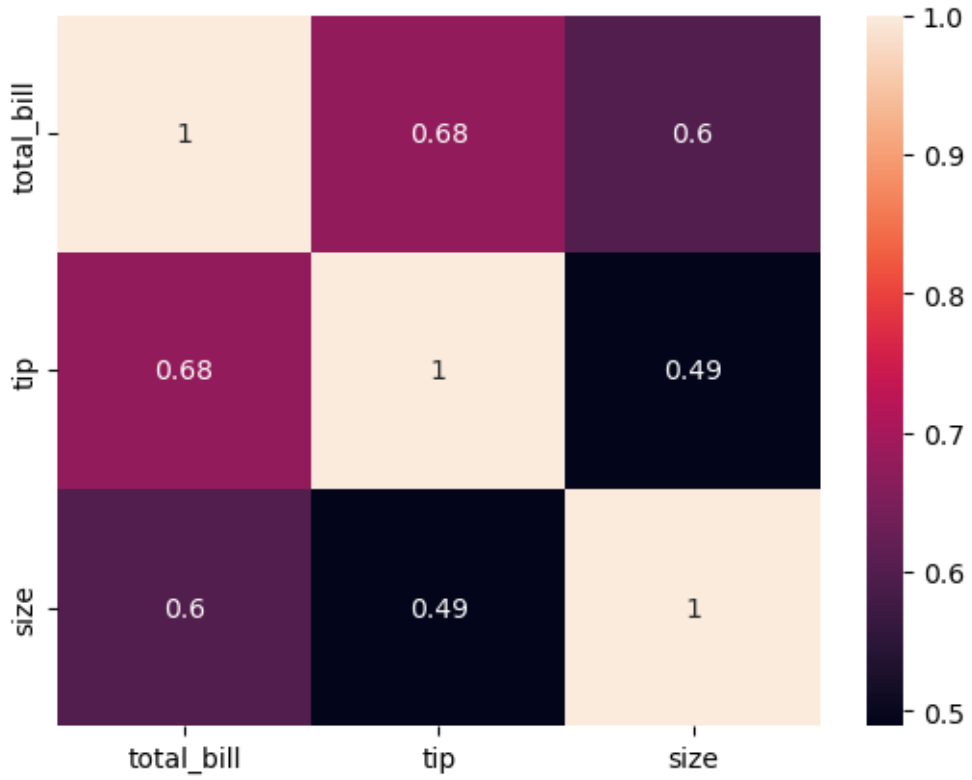
[67]:

	total_bill	tip	size
0	16.99	1.01	2
1	10.34	1.66	3
2	21.01	3.50	3
3	23.68	3.31	2
4	24.59	3.61	4
..	…	…	…
239	29.03	5.92	3
240	27.18	2.00	2
241	22.67	2.00	2
242	17.82	1.75	2
243	18.78	3.00	2

[244 rows x 3 columns]

[68]:
```
sns.heatmap(df_num.corr(),annot=True)
```

[68]:
```
<Axes: >
```



[69]:
```
df[['size']]
```

[69]:
```
     size
0       2
1       3
2       3
3       2
4       4
..    ...
239     3
240     2
241     2
242     2
243     2

[244 rows x 1 columns]
```

[70]:
```
df.head(2)
```

[70]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3

[71]:
```
df['size'].value_counts()
```

[71]:
```
size
2    156
3     38
4     37
5      5
1      4
6      4
Name: count, dtype: int64
```

[72]:
```
def map_size_to_label(size):
    if size == 1:
        return 'Regular'
    elif size == 2:
        return 'Medium'
    elif size >= 3:
        return 'Gold'
    else:
        return 'Other'
```

[73]:
```
df.head(2)
```

[73]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3

[74]:
```
# Apply the function to create a new 'label' column
df['type'] = df['size'].apply(map_size_to_label)
```

[75]:
```
df.head()
```

[75]:

	total_bill	tip	sex	smoker	day	time	size	type
0	16.99	1.01	Female	No	Sun	Dinner	2	Medium
1	10.34	1.66	Male	No	Sun	Dinner	3	Gold
2	21.01	3.50	Male	No	Sun	Dinner	3	Gold
3	23.68	3.31	Male	No	Sun	Dinner	2	Medium
4	24.59	3.61	Female	No	Sun	Dinner	4	Gold

[76]:
```
df.to_excel("Ds 401.xlsx",index=False)
```

[77]:
```
gender_label_counts = df.groupby(['sex', 'type']).size().
↪reset_index(name='count')
```

[78]:
```
gender_label_counts
```

[78]:
```
      sex     type  count
0  Female     Gold     26
1  Female   Medium     58
2  Female  Regular      3
3    Male     Gold     58
4    Male   Medium     98
5    Male  Regular      1
```

[79]:
```
gender_label_counts.sort_index(ascending=False)
```

[79]:
```
      sex     type  count
5    Male  Regular      1
4    Male   Medium     98
3    Male     Gold     58
2  Female  Regular      3
1  Female   Medium     58
0  Female     Gold     26
```

[80]:
```
pivot_table = gender_label_counts.pivot(index='type', columns='sex',␣
↪values='count')
```

[81]:
```
pivot_table
```

[81]:
```
sex      Female  Male
type
Gold         26    58
Medium       58    98
Regular       3     1
```

[82]:
```
pivot_table.sort_index(ascending=False)
```

[82]:
```
sex      Female  Male
type
Regular       3     1
Medium       58    98
Gold         26    58
```

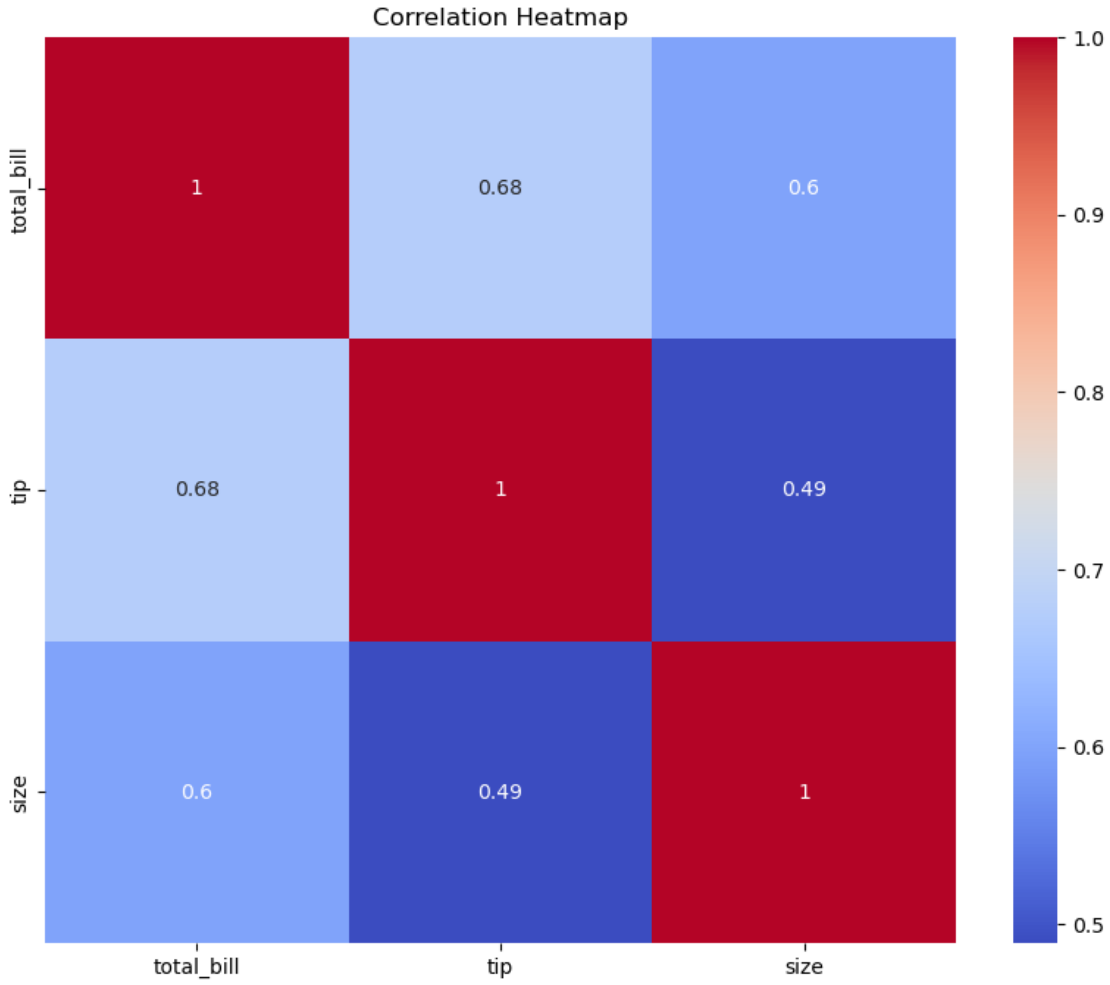
[83]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

def perform_eda(df):
    # Check for missing values
    missing_values = df.isnull().sum()
    print("Missing Values:")
    print(missing_values[missing_values > 0])  # Display columns with missing␣
↪values
```

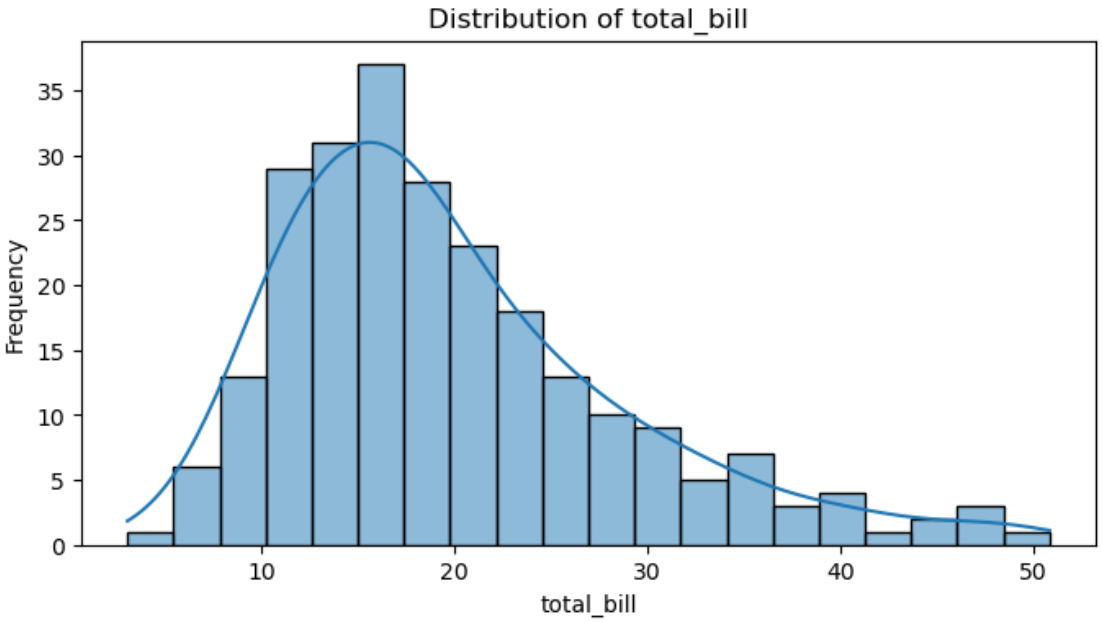
```
    # Summary statistics
    summary_stats = df.describe()
    print("\nSummary Statistics:")
    print(summary_stats)

    # Correlation heatmap for numeric columns
    numeric_df = df.select_dtypes(include=['number'])
    plt.figure(figsize=(10, 8))
    sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
    plt.title("Correlation Heatmap")
    plt.show()

    # Distribution of numeric columns
    for column in numeric_df.columns:
        plt.figure(figsize=(8, 4))
        sns.histplot(df[column], bins=20, kde=True)
        plt.title(f"Distribution of {column}")
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.show()

    # Countplot for categorical columns
    categorical_df = df.select_dtypes(exclude=['number'])
    for column in categorical_df.columns:
        plt.figure(figsize=(8, 4))
        sns.countplot(data=df, x=column)
        plt.title(f"Countplot of {column}")
        plt.xlabel(column)
        plt.ylabel("Count")
        plt.xticks(rotation=45)
        plt.show()
```

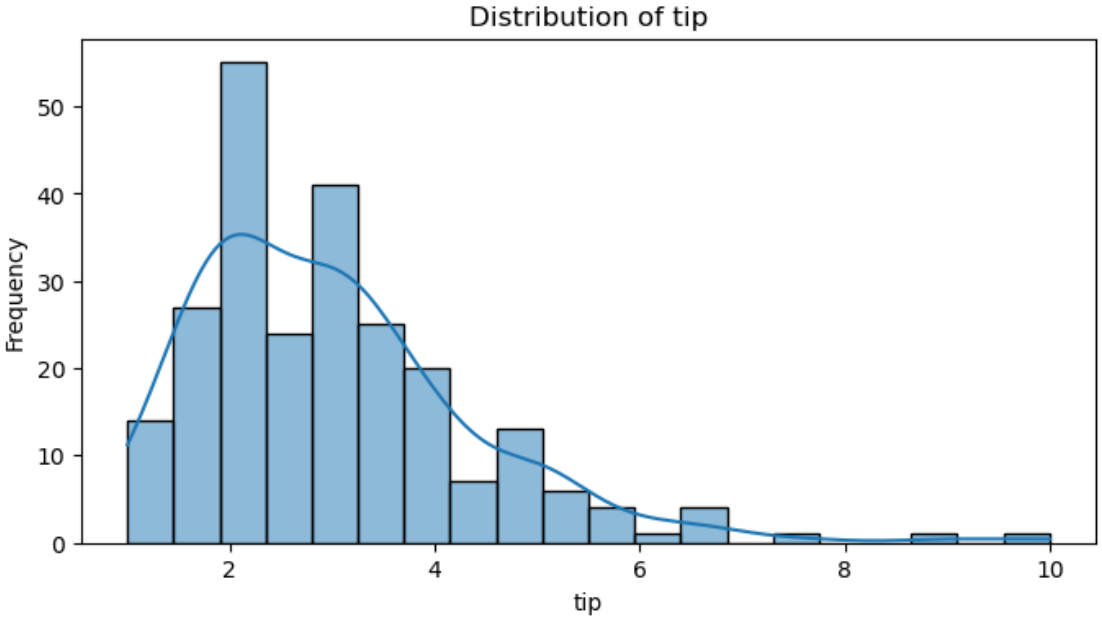
[84]:
```
perform_eda(df)
```

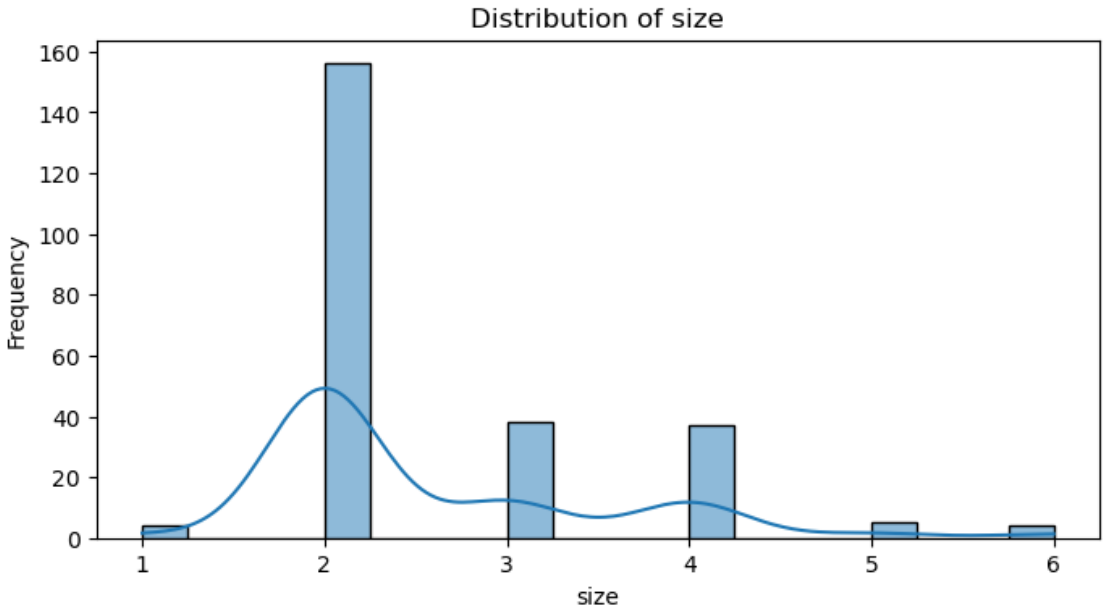
```
Missing Values:
Series([], dtype: int64)

Summary Statistics:
       total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
```

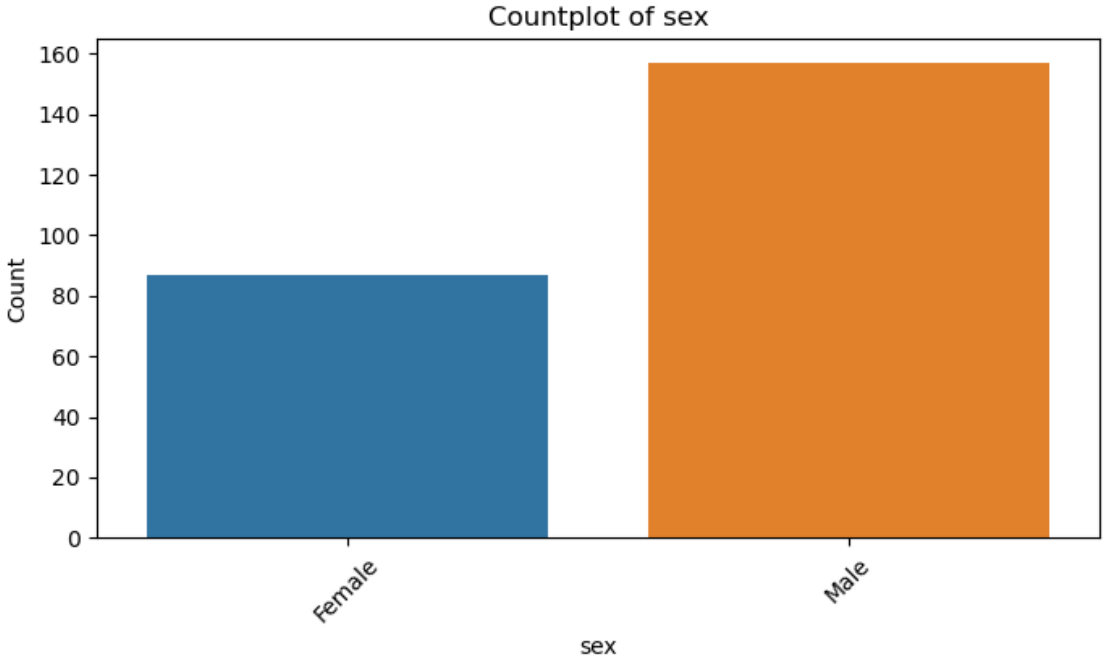
```
max       50.810000    10.000000     6.000000
```

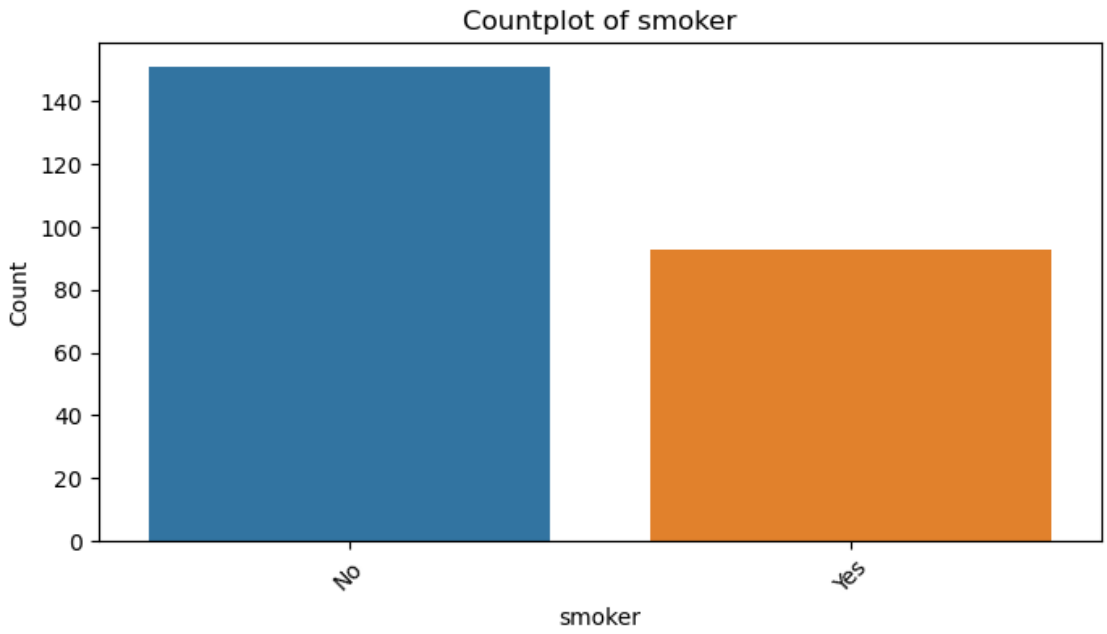


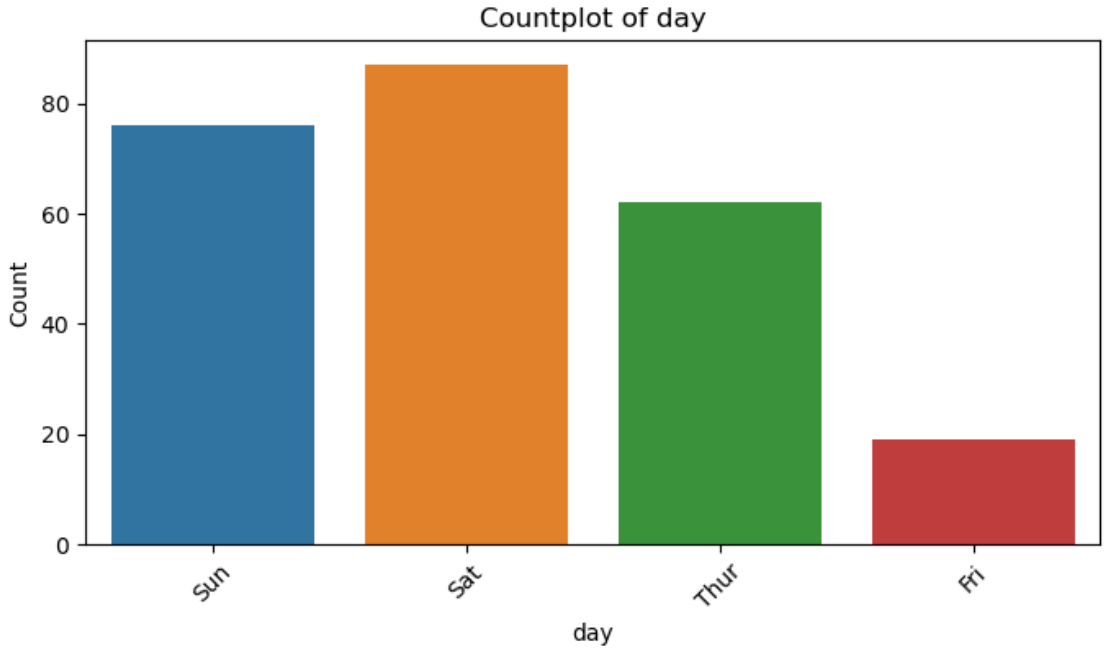

Distribution of total_bill
Frequency
total_bill
0
5
10
15
20
25
30
35
10
20
30
40
50

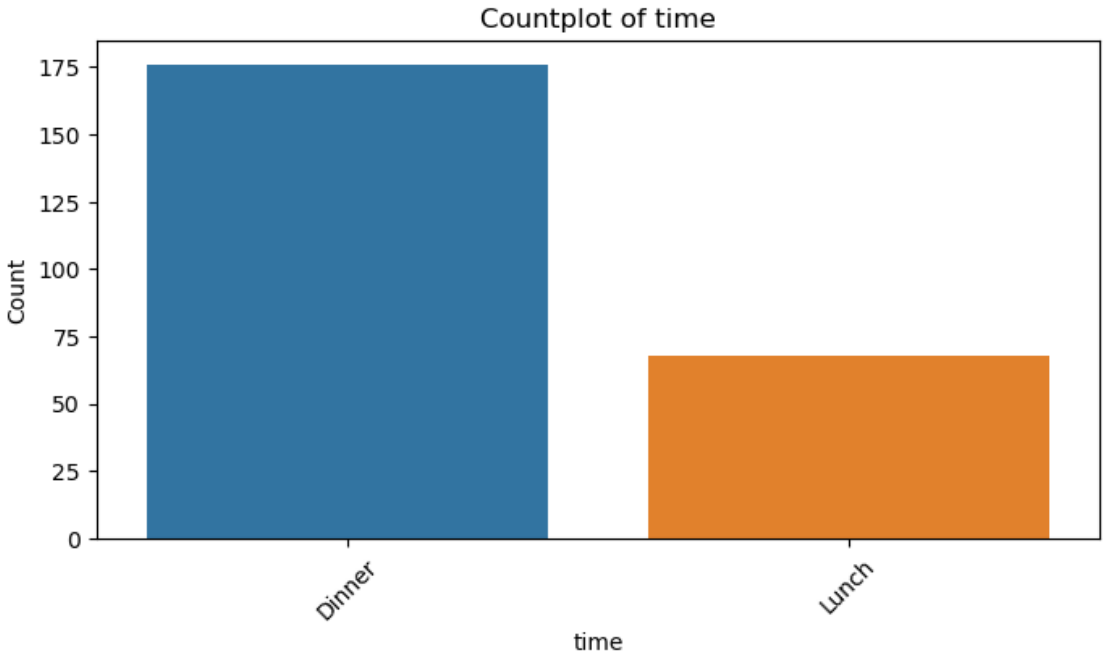

Distribution of tip
Frequency
tip
0
10
20
30
40
50
2
4
6
8
10

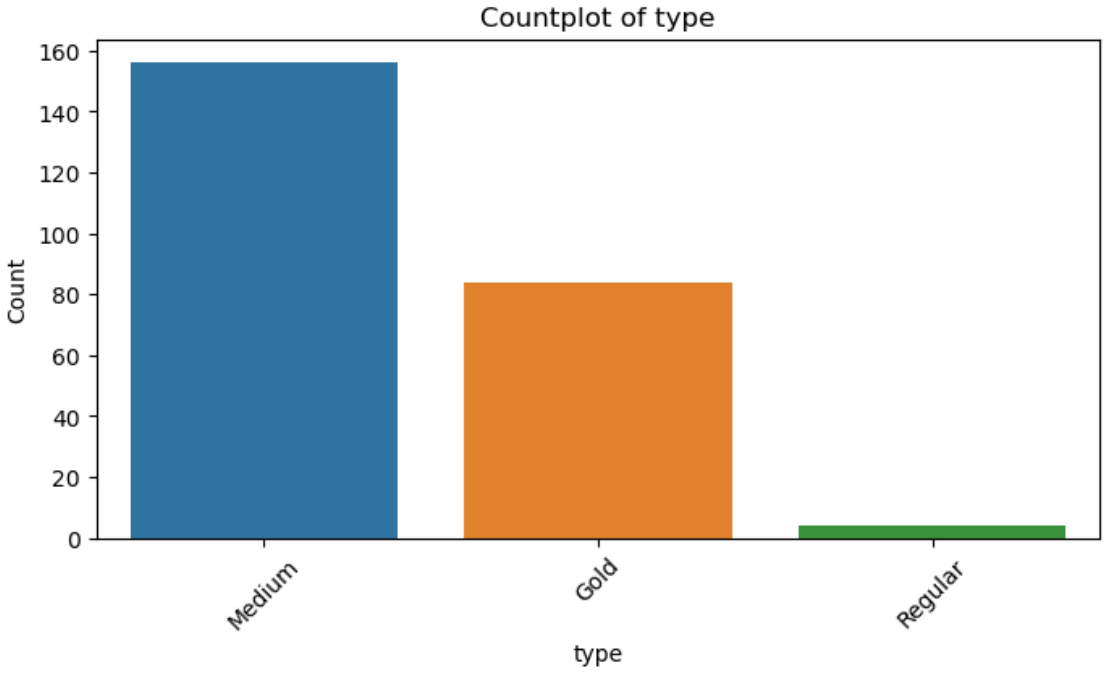

Distribution of size
Frequency
size
0
20
40
60
80
100
120
140
160
1
2
3
4
5
6


Countplot of sex
Count
sex
0
20
40
60
80
100
120
140
160
Female
Male


Countplot of smoker
Count
0
20
40
60
80
100
120
140
No
Yes
smoker


Countplot of day
Count
0
20
40
60
80
Sun
Sat
Thur
Fri
day


Countplot of time
Count
175
150
125
100
75
50
25
0
Dinner
Lunch
time


Countplot of type
Count
160
140
120
100
80
60
40
20
0
Medium
Gold
Regular
type