CLARUSWAY
WAY TO REINVENT YOURSELF

SEABORN CHEAT-SHEETS

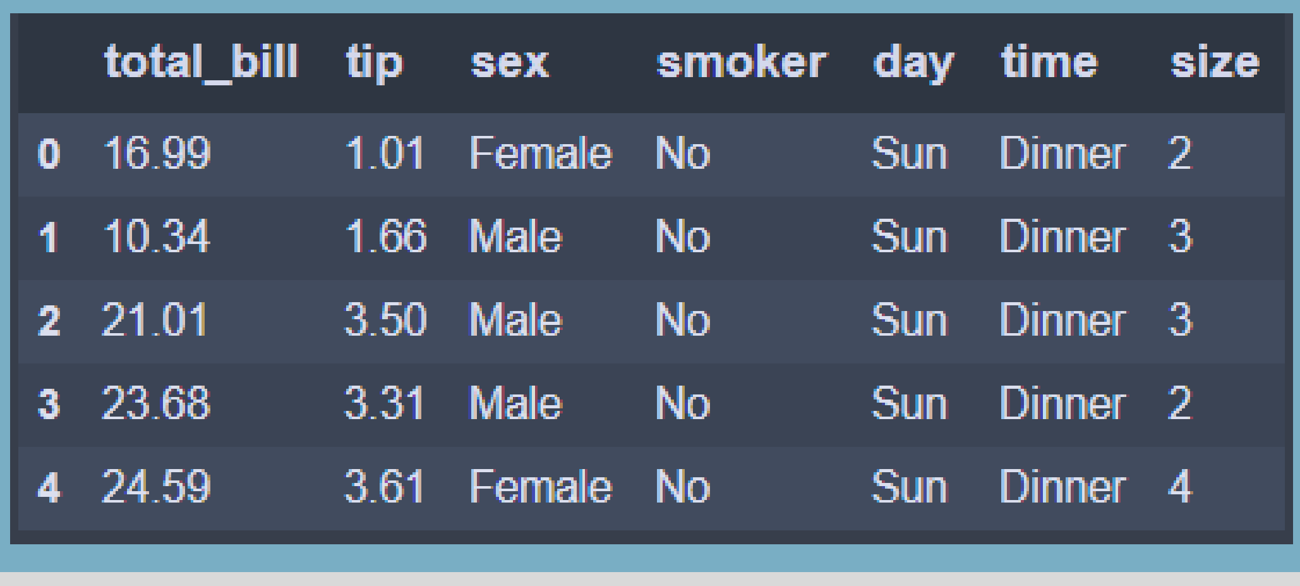
Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

import the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

load the dataset

```
tips = sns.load_dataset('tips')
tips.head()
```

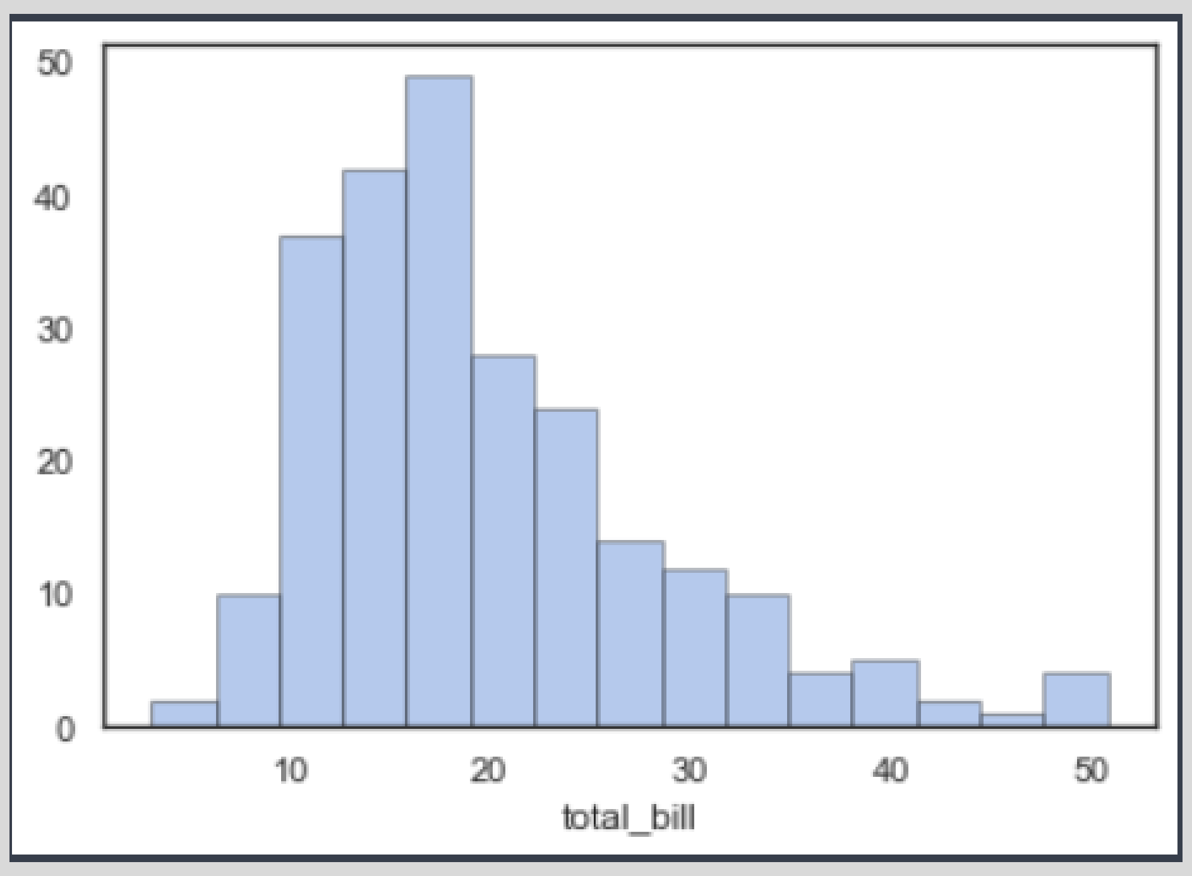


	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

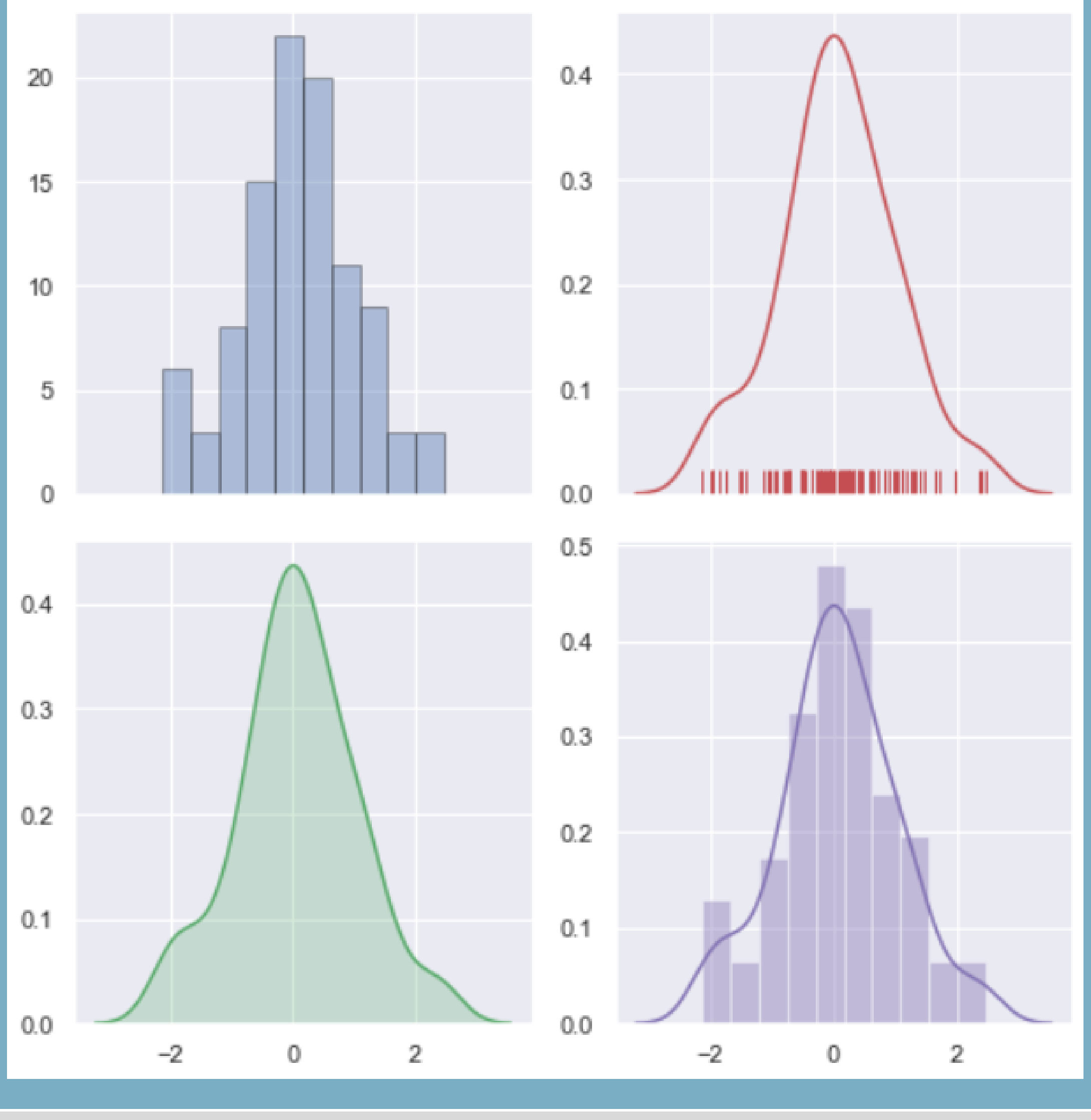
1- DISTRIBUTION PLOT

A) DISPLOT

```
sns.distplot(tips['total_bill'],
hist_kws=dict(edgecolor="k",linewidth=1), bins=15,
kde=False);
```



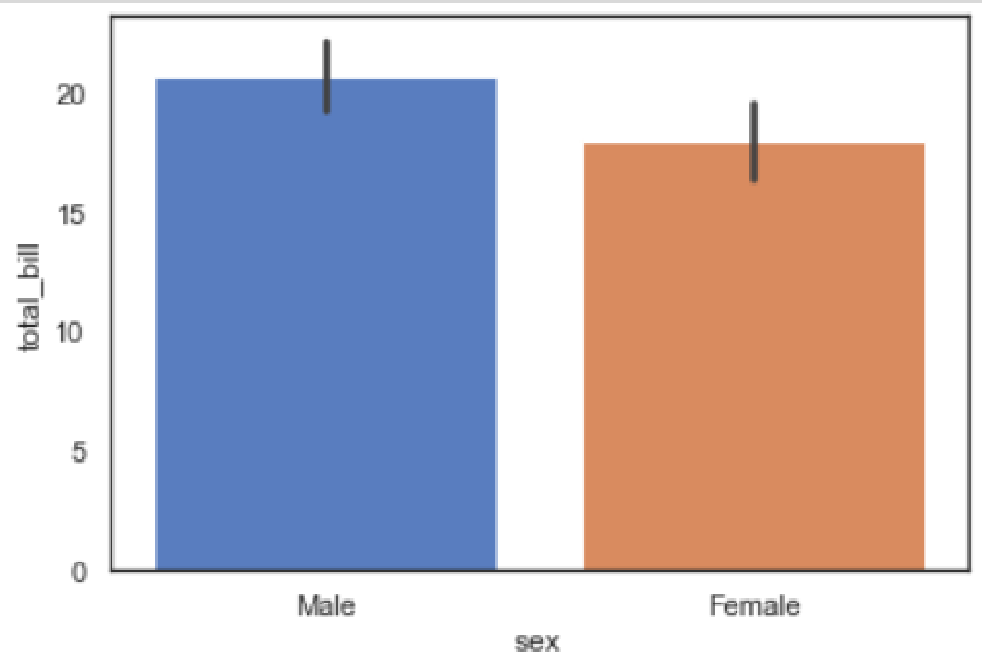
```
sns.set(style="darkgrid")
# Set up the matplotlib figure
f, axes = plt.subplots(2, 2, figsize=(7, 7), sharex=True)
sns.despine()
# Generate a random univariate dataset
rs = np.random.RandomState(10)
d = rs.normal(size=100)
# Plot a simple histogram with binsize determined automatically
sns.distplot(d, kde=False, hist_kws=dict(edgecolor="k", linewidth=1), color="b", ax=axes[0, 0])
# Plot a kernel density estimate and rug plot
sns.distplot(d, hist=False, rug=True, color="r", ax=axes[0, 1])
# Plot a filled kernel density estimate
sns.distplot(d, hist=False, color="g", kde_kws={"shade": True}, ax=axes[1, 0])
# Plot a histogram and kernel density estimate
sns.distplot(d, color="m", ax=axes[1, 1])
plt.tight layout()
```



2- CATEGORICAL PLOT

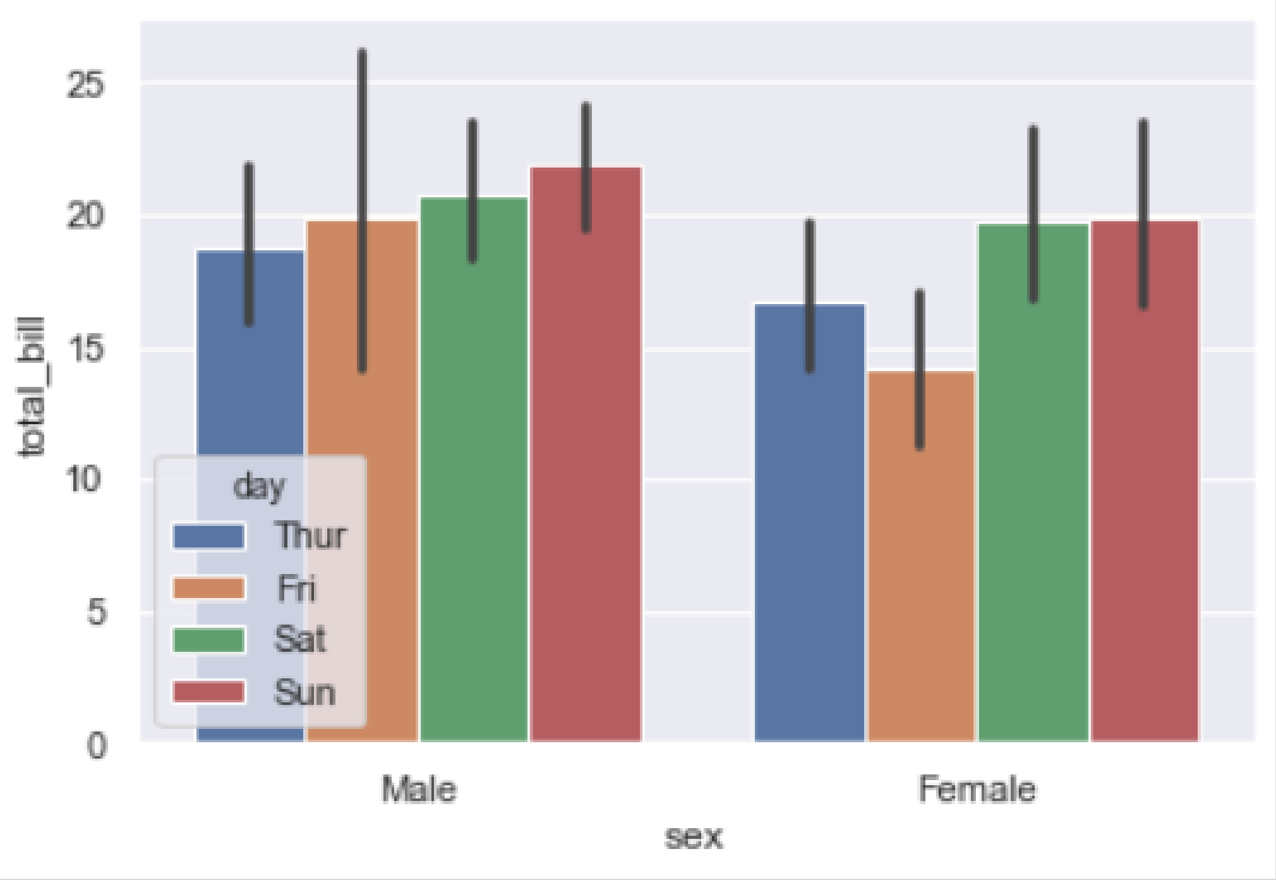
A) BAR PLOT - METHOD 1

```
sns.barplot(tips["sex"], tips["total_bill"]);
```



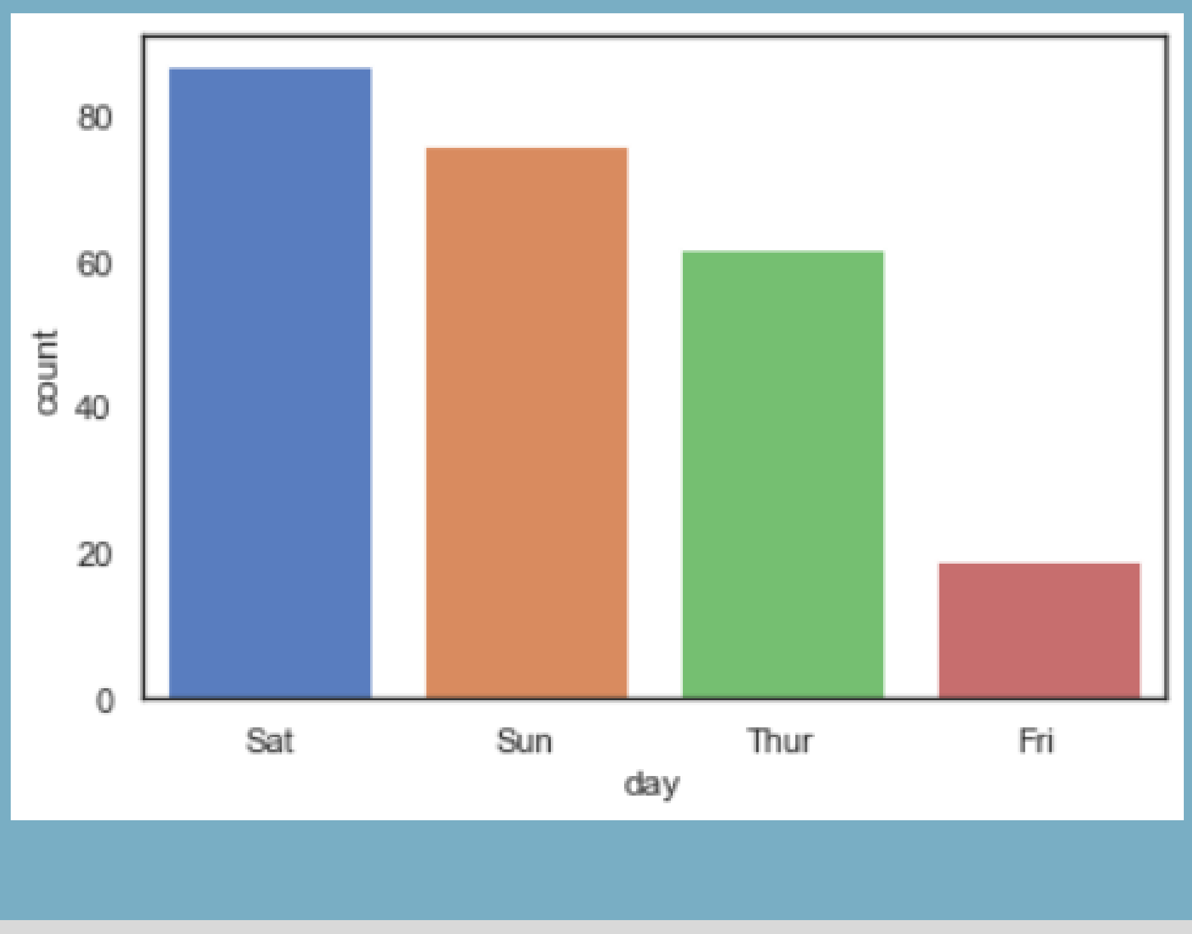
METHOD 2

```
sns.barplot(x="sex", y="total_bill", hue="day", data=tips);
```



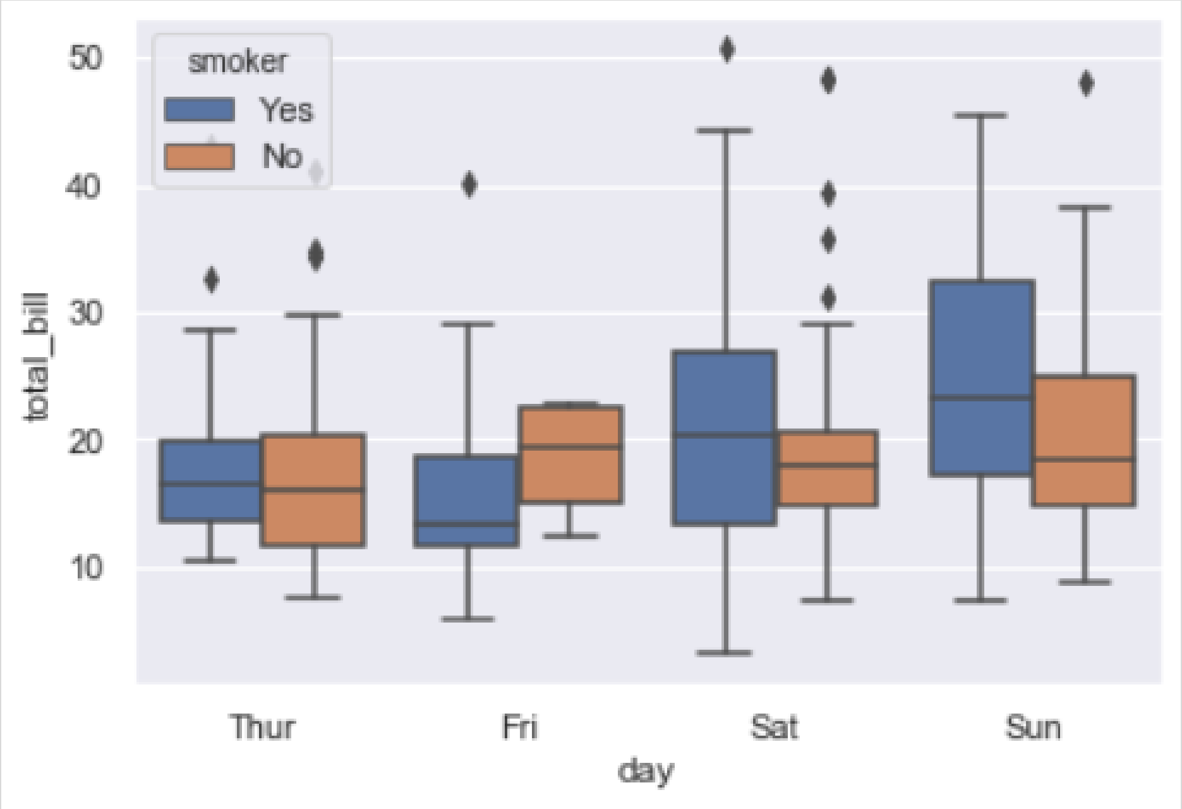
B) COUNTPLOT

```
sns.countplot(x='day',order=tips.day.value_counts().index,data=tips);
```



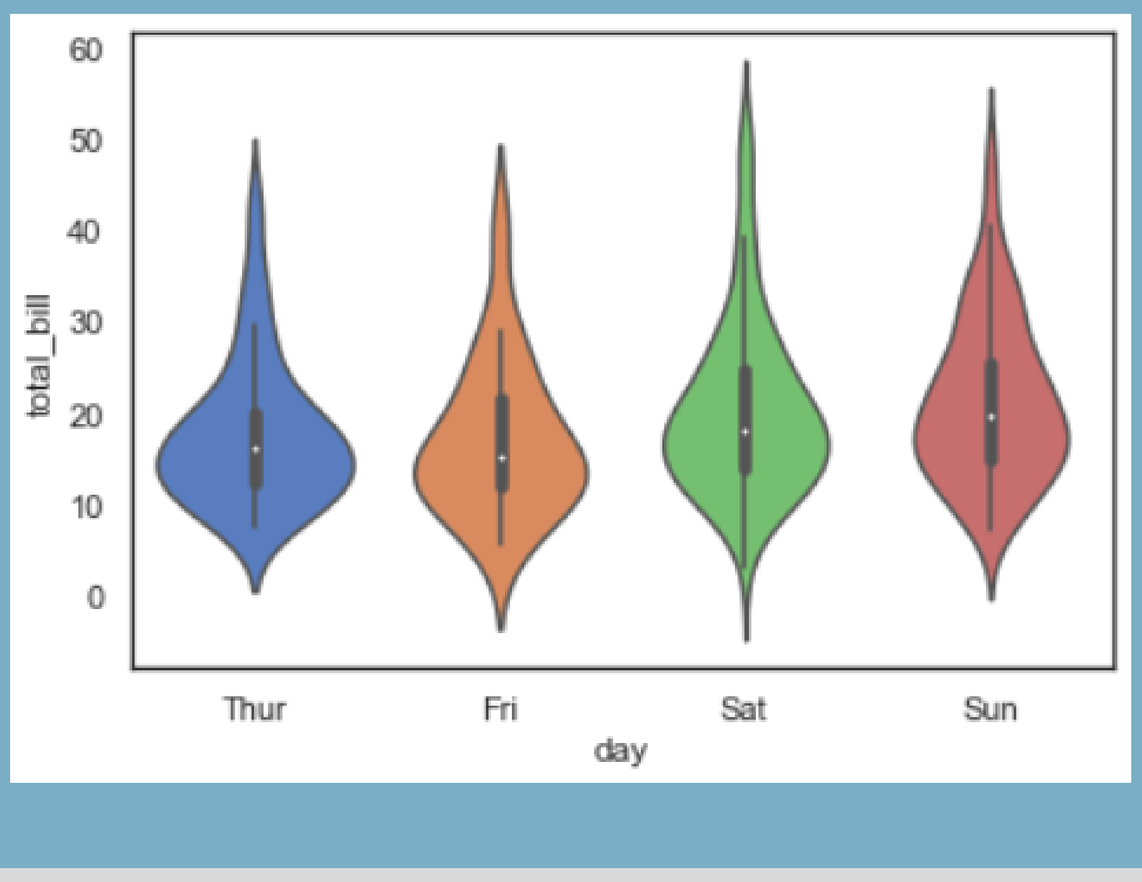
C) BOXPLOT

```
sns.boxplot(x="day", y="total_bill", hue="smoker", data=tips);
```



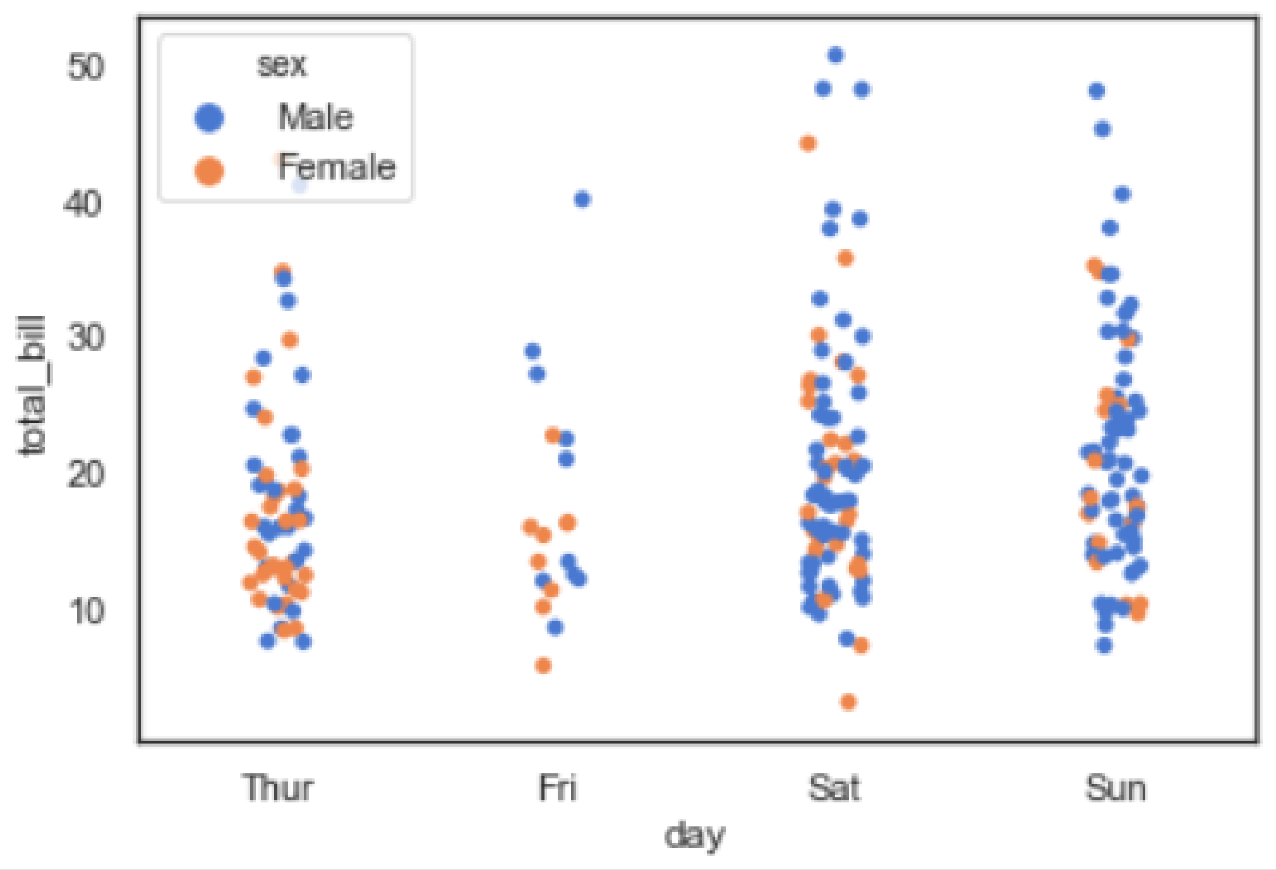
D) VIOLINPLOT

```
sns.violinplot(x="day", y="total_bill", data=tips);
```



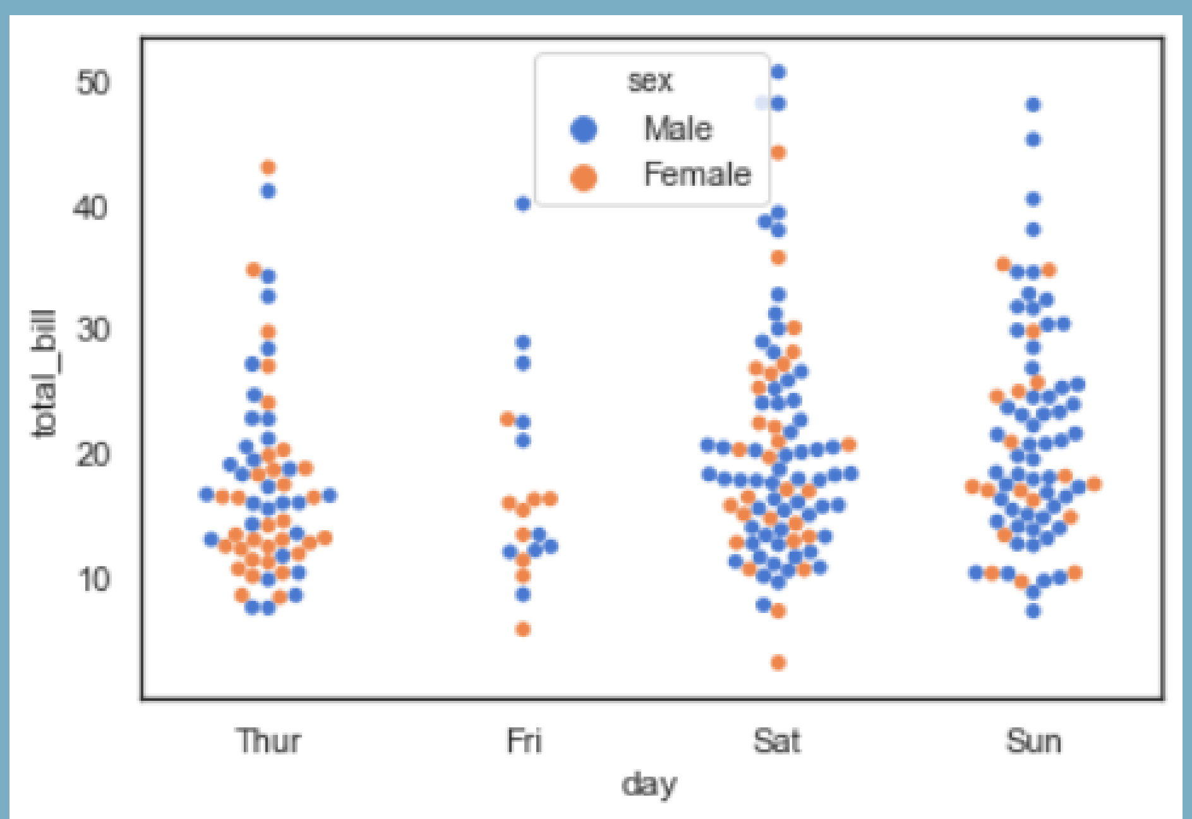
E) STRIPPLOT

```
sns.stripplot(x="day", y="total_bill", hue="sex", data=tips);
```



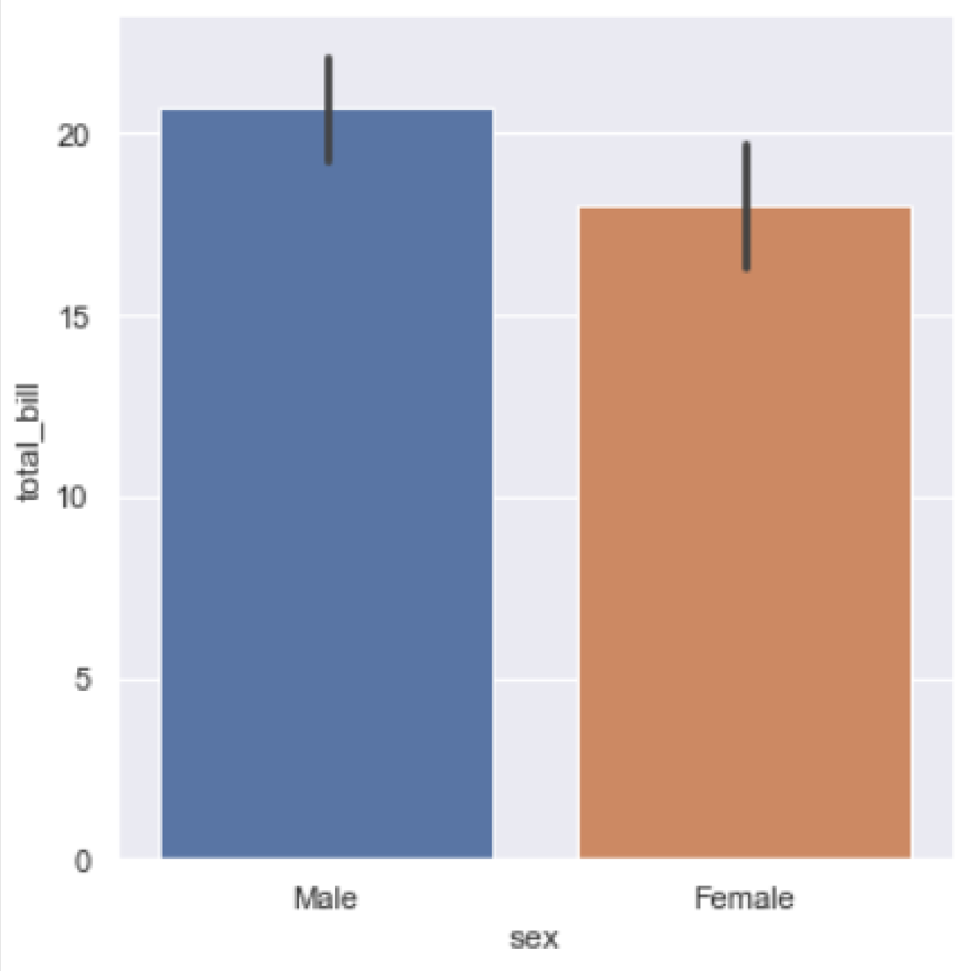
F) SWARMPLOT (VIOLINPLOT + STRIPPLOT)

```
sns.swarmplot(x="day", y="total_bill", hue="sex", data=tips);
```



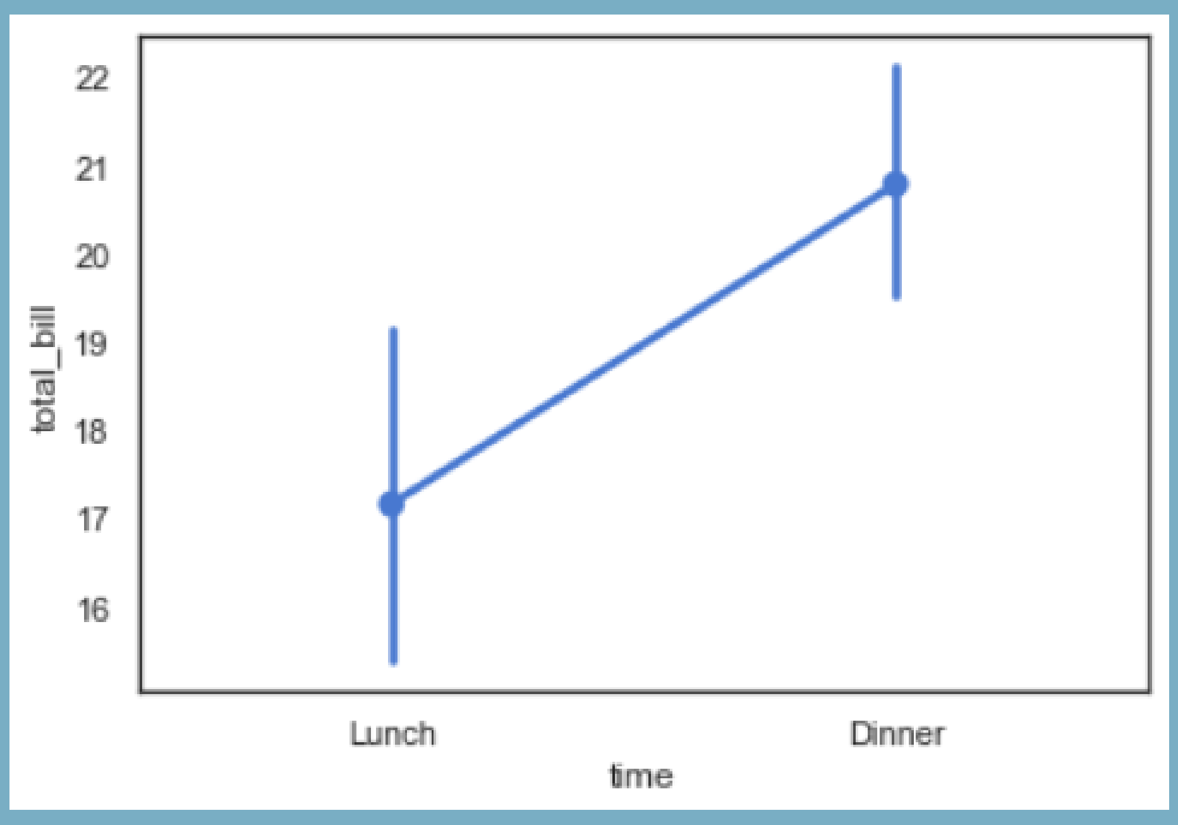
G) CATPLOT (FORMER NAME: FACTOR PLOT)

```
sns.catplot(x='sex',y='total_bill', data=tips, kind='bar')
# kind options: bar, swarm, strip(default), box, violin, point andcount.
```



H) POINTPLOT

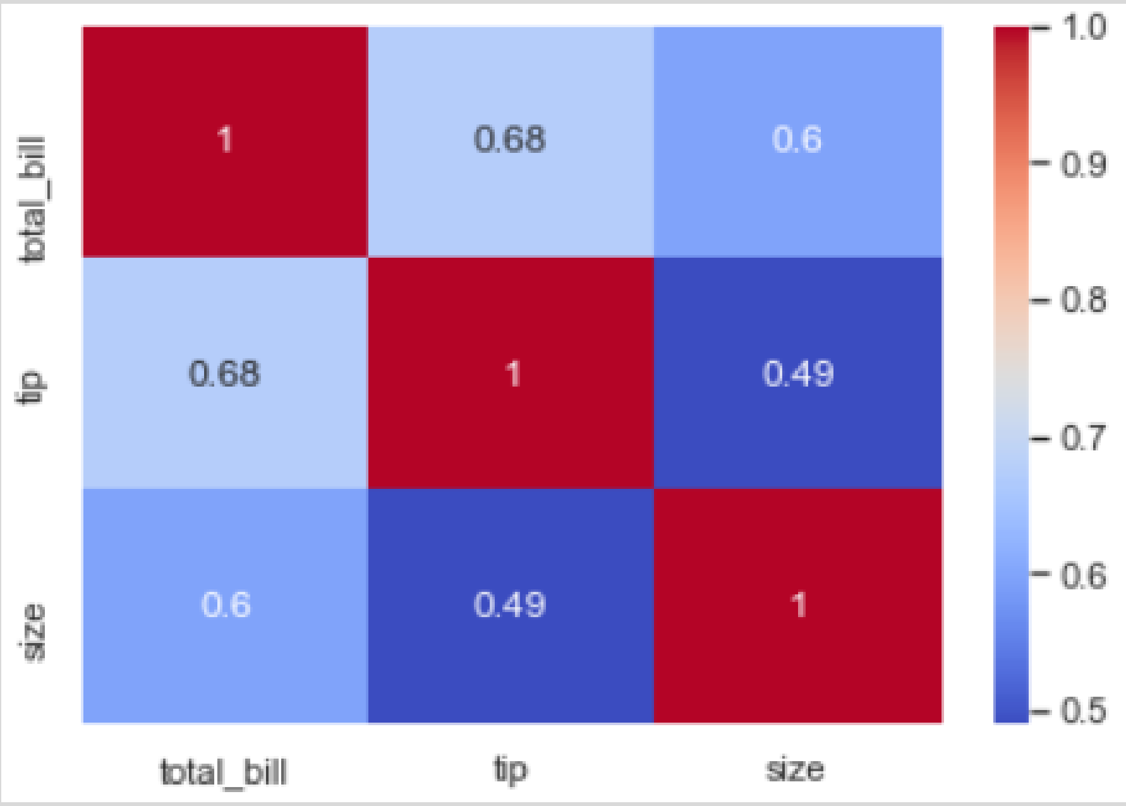
```
sns.pointplot(x="time", y="total_bill", data=tips)
```



3- Matrix and Grid Plots

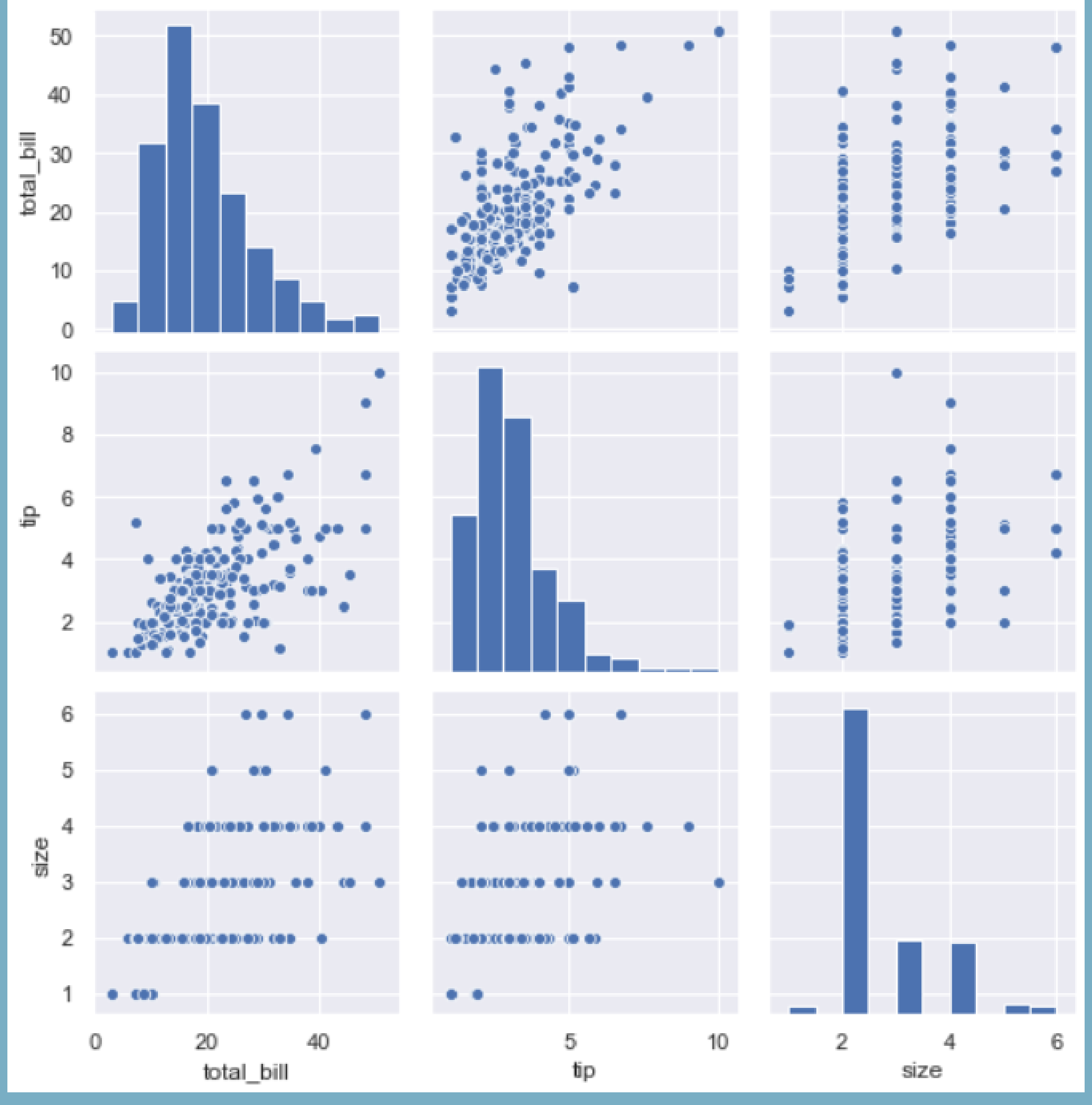
A-Heat Map

```
sns.heatmap(tips.corr(), cmap="coolwarm", annot=True);
```



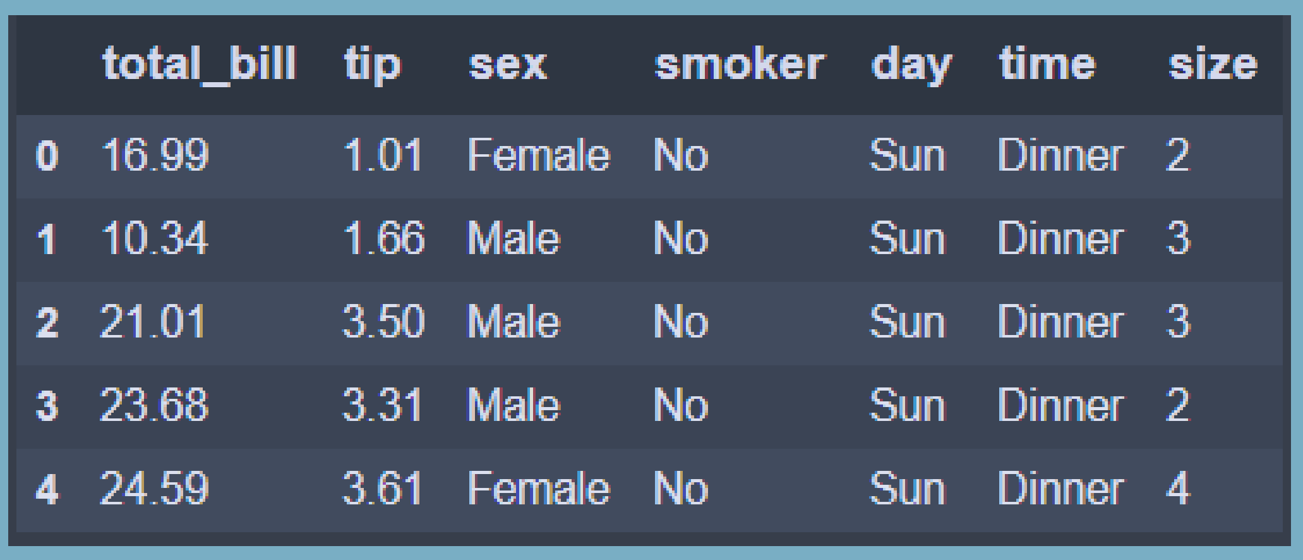
B) PAIR PLOT

```
sns.pairplot(tips);
```



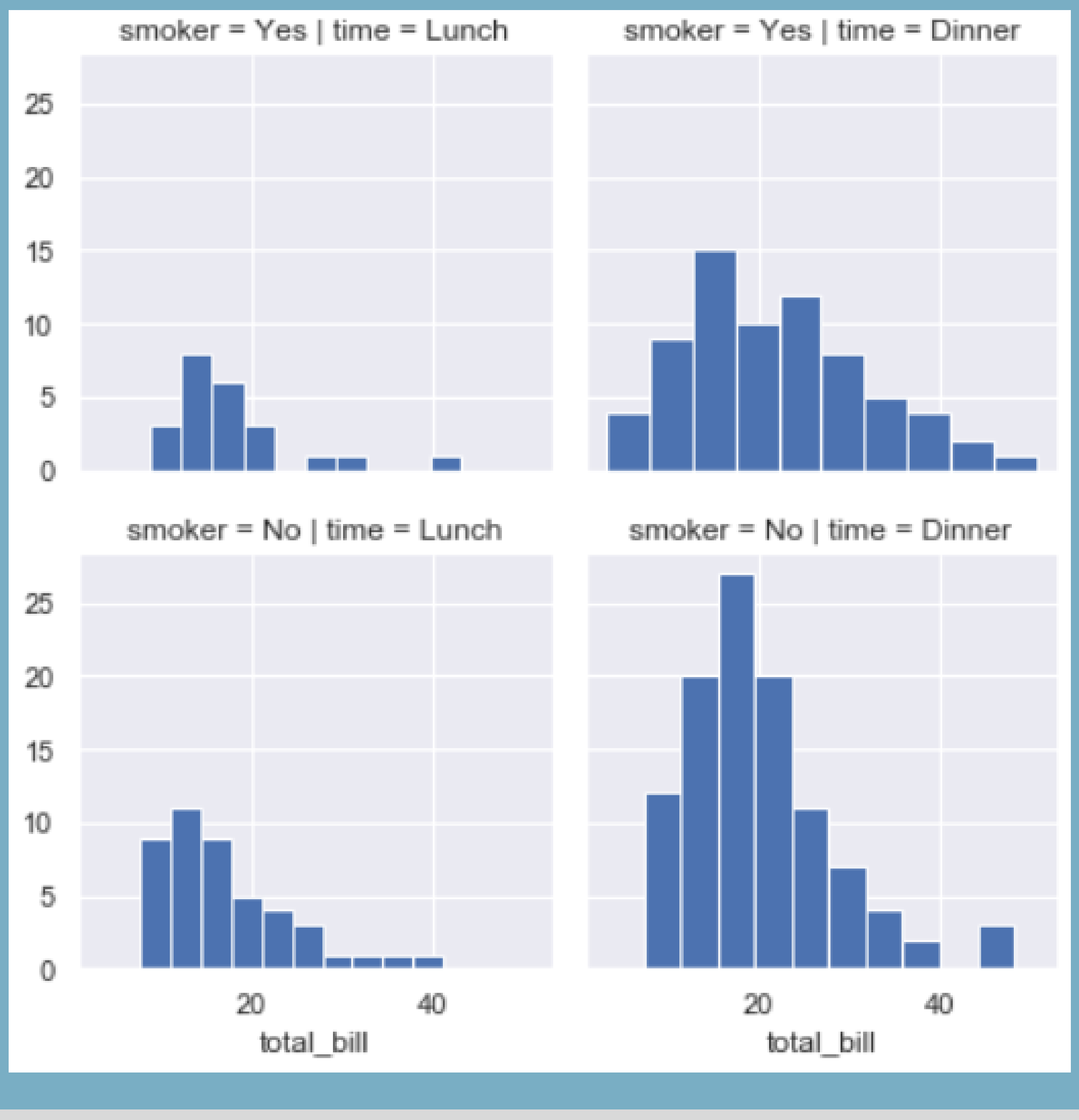
C) FACETGRID

```
tips.head()
```



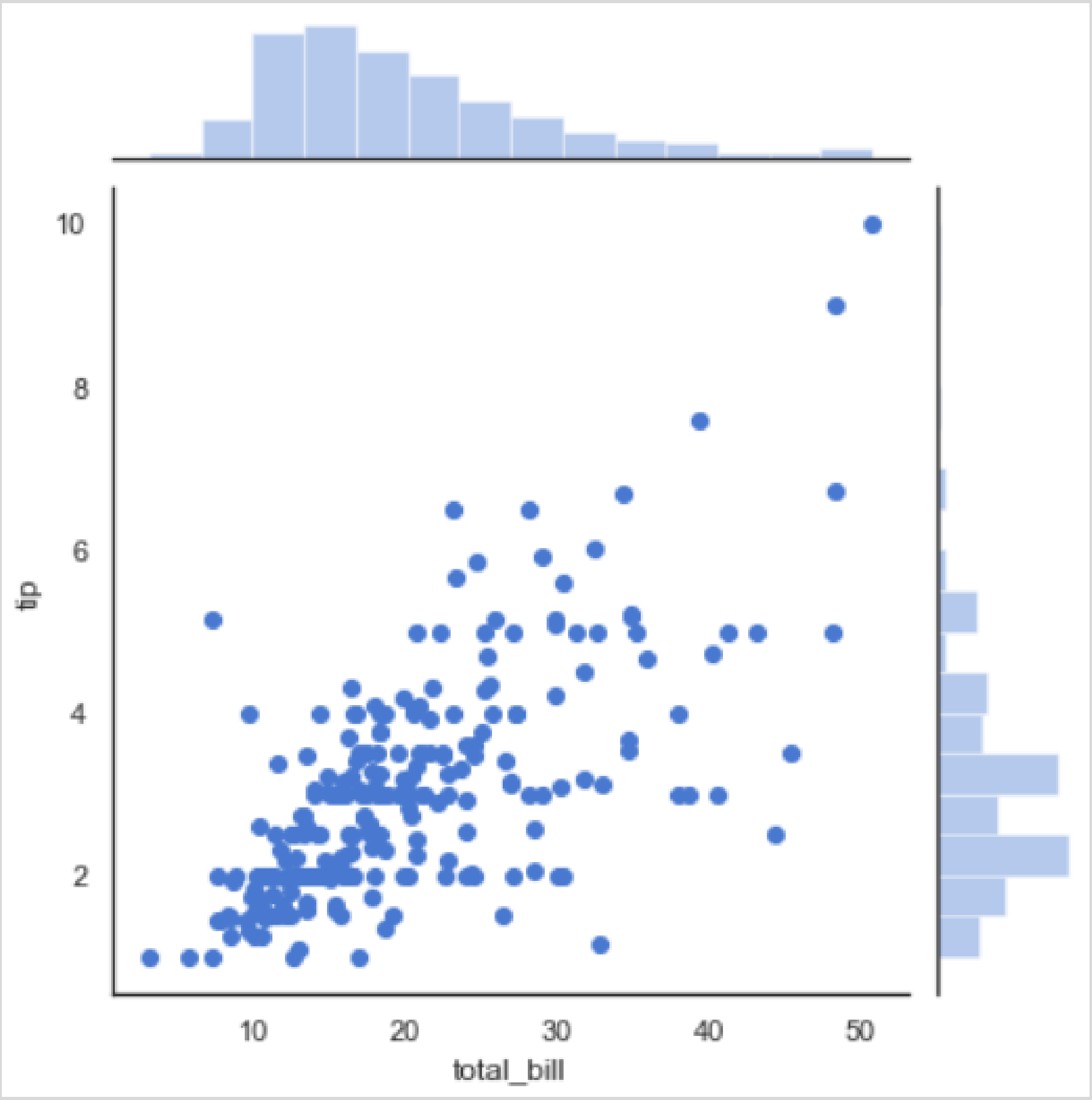
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
g = sns.FacetGrid(tips, col="time", row="smoker")g.map(plt.hist,
"total_bill")
```



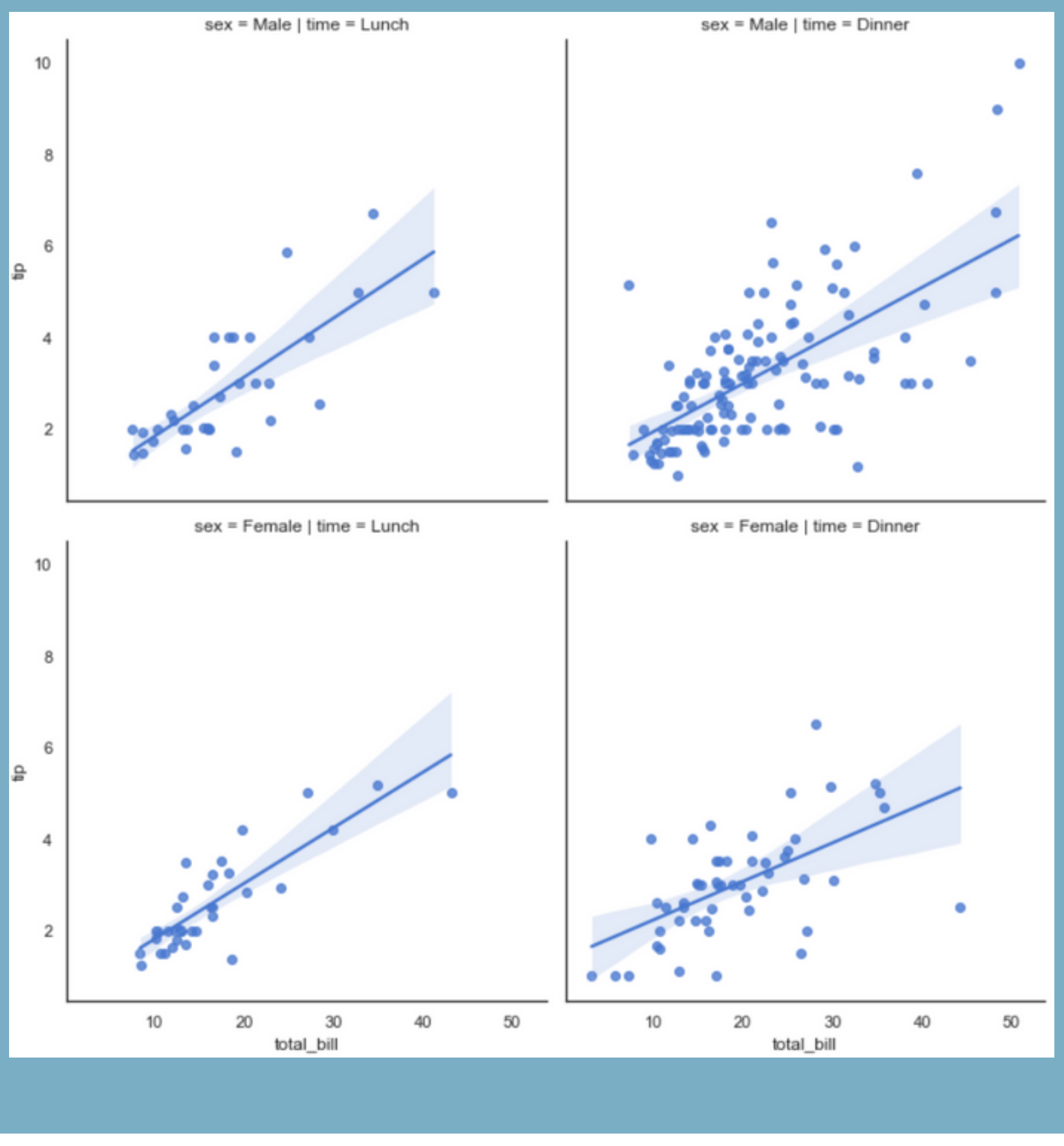
4- JOINPLOT

```
sns.jointplot(x="total_bill", y="tip", kind="scatter", data=tips);
# kind options =scatter (default), reg, resid, kde, hex.
```



5- LMPLOT

```
sns.lmplot(x="total_bill", ci=95, y="tip", data=tips,
row="sex",col="time", aspect=1, height=5)
```



CLARUSWAY

WAY TO REINVENT YOURSELF