Feature Engineering

1. Univariate analysis

For numerical column - histogram(Distplot | histogram)

For Categorical column - Barplot | Countplot

2. Bivariate analysis : Predictor vs Target Variable

Num - Num | Scatterplot, lmplot

Cat - Num | Boxplot | Violinplot

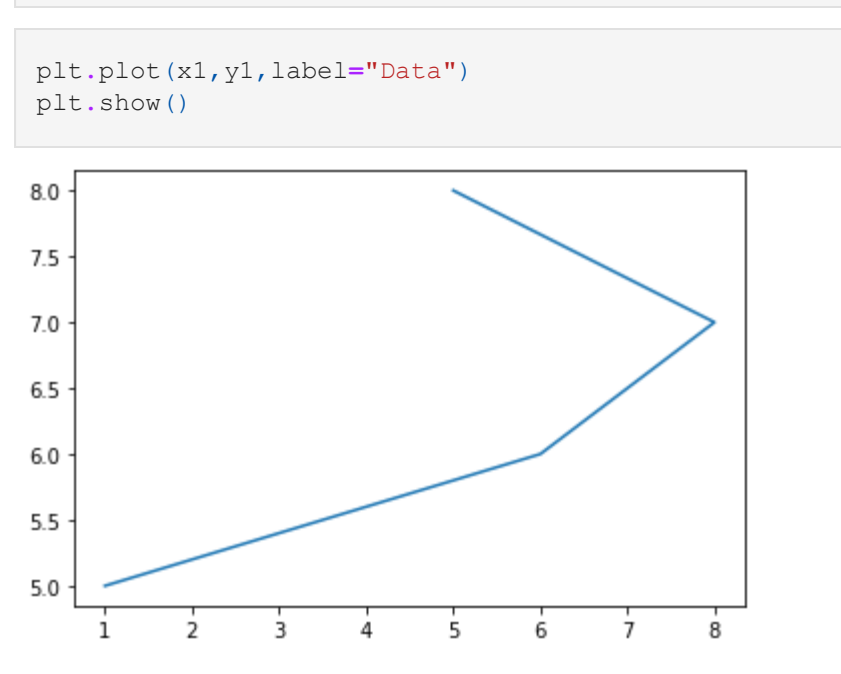
Cat - Cat | pd.crosstab | Table

3. Missing Values : Mean and Median Imputation | Model based Imputation

In [1]:
```
import matplotlib.pyplot as plt
```

In [2]:
```
x1=[1,6,8,5]
y1=[5,6,7,8]
```

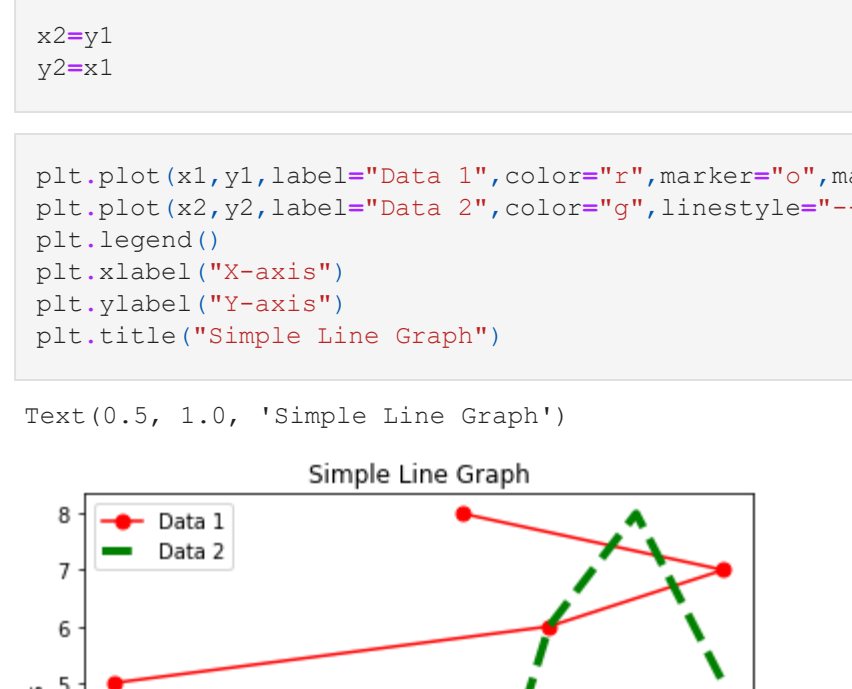
In [3]:
```
plt.plot(x1,y1,label="Data")
plt.show()
```



In [4]:
```
x2=y1
y2=x1
```

In [5]:
```
plt.plot(x1,y1,label="Data 1",color="r",marker="o",markersize=7) #marker: to mark the position 's' = square,'o
plt.plot(x2,y2,label="Data 2",color="g",linestyle="--",linewidth=4) #linestyle  toget line in diff shape
plt.legend()
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Simple Line Graph")
```

Out[5]: `Text(0.5, 1.0, 'Simple Line Graph')`



In [6]:
```
import seaborn as sns
```

In [7]:
```
d1=sns.load_dataset("tips")
```

In [8]:
```
d1.head() #Head is used to get an idea abt the dataset like how the data is? To know what all feature we have
```

Out[8]:

	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

In [9]:
```
d1.tail() #Maybe this is an data of some restaurant
```

Out[9]:

	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

In [10]:
```
#To check what values are their in the data like in time
d1['time'].unique() #So, data has lunch values also
```

Out[10]:
```
['Dinner', 'Lunch']
Categories (2, object): ['Dinner', 'Lunch']
```

In [11]:
```
d1['day'].unique()
```

Out[11]:
```
['Sun', 'Sat', 'Thur', 'Fri']
Categories (4, object): ['Sun', 'Sat', 'Thur', 'Fri']
```

In [12]:
```
d1.nunique() #No. of unique values in all columns
```

Out[12]:
```
total_bill    229
tip           123
sex             2
smoker          2
day             4
time            2
size            6
dtype: int64
```

In [13]:
```
d1.count() #Out of tot 244 we hve 229 unique values for total bill
```

Out[13]:
```
total_bill    244
tip           244
sex           244
smoker        244
day           244
time          244
size          244
dtype: int64
```

In [ ]:
```
#Question : Are this restaurent is prefer by more male or females?
```

In [ ]:
```
# Are more smokers come to restaurent? give some officer for smoker sell ciggretts
```

In [ ]:
```
# Are restaurent i more crowded  sat aand sun only? if yes give off to staff on friday
```

In [ ]:
```
# Was is it lunch or dinner place?
```

In [ ]:
```
# Is this is high cost restaurentor low cost?
```

In [ ]:
```
# How to attract customers who pay higest bill?
```

So in order to get answers to all these questions we need to identify the data `Distributions`

How the data are distributed accross... We need plot and data to `infer` the details

We should know `which plot to be used for which type of work`.

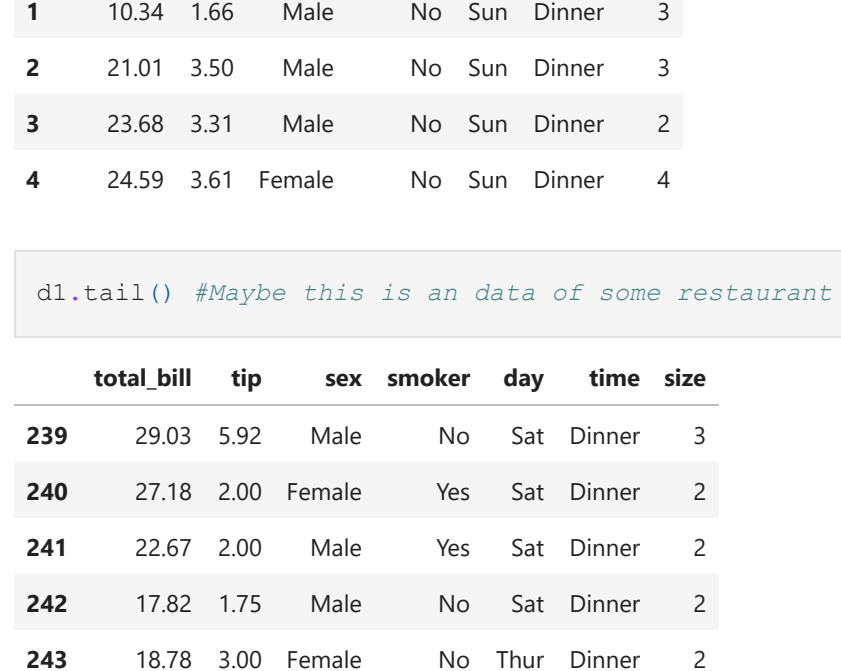
In [ ]:
```
## Distribution : histogram , distplot
```

Question: Need to know the `relationship b/w the total bill and tip`?

In [14]:
```
# So, for that we need to do bi-varite analysis : Use Scatter Plot
```

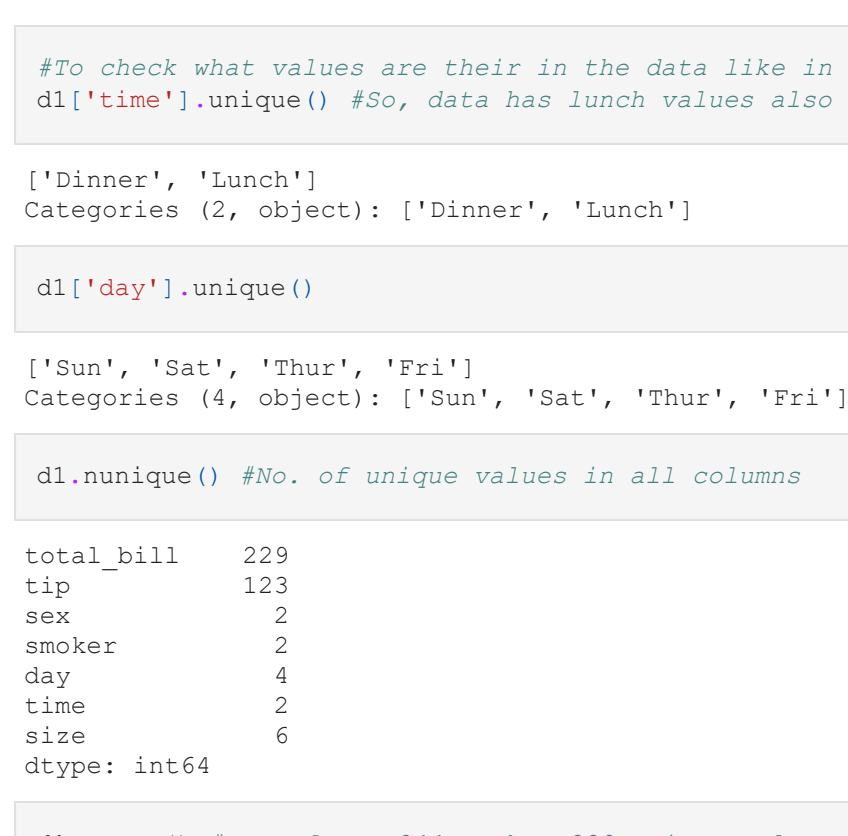
In [15]:
```
plt.scatter(x="total_bill",y="tip",data=d1)
```

Out[15]: `<matplotlib.collections.PathCollection at 0x2367da94220>`



In [16]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=True) #Grap used fr ML purposes.
#'lm' stands for 'Linear Model' plot, 'fit_reg' = fitting the regression line
```

Out[16]: `<seaborn.axisgrid.FacetGrid at 0x2367dab3340>`



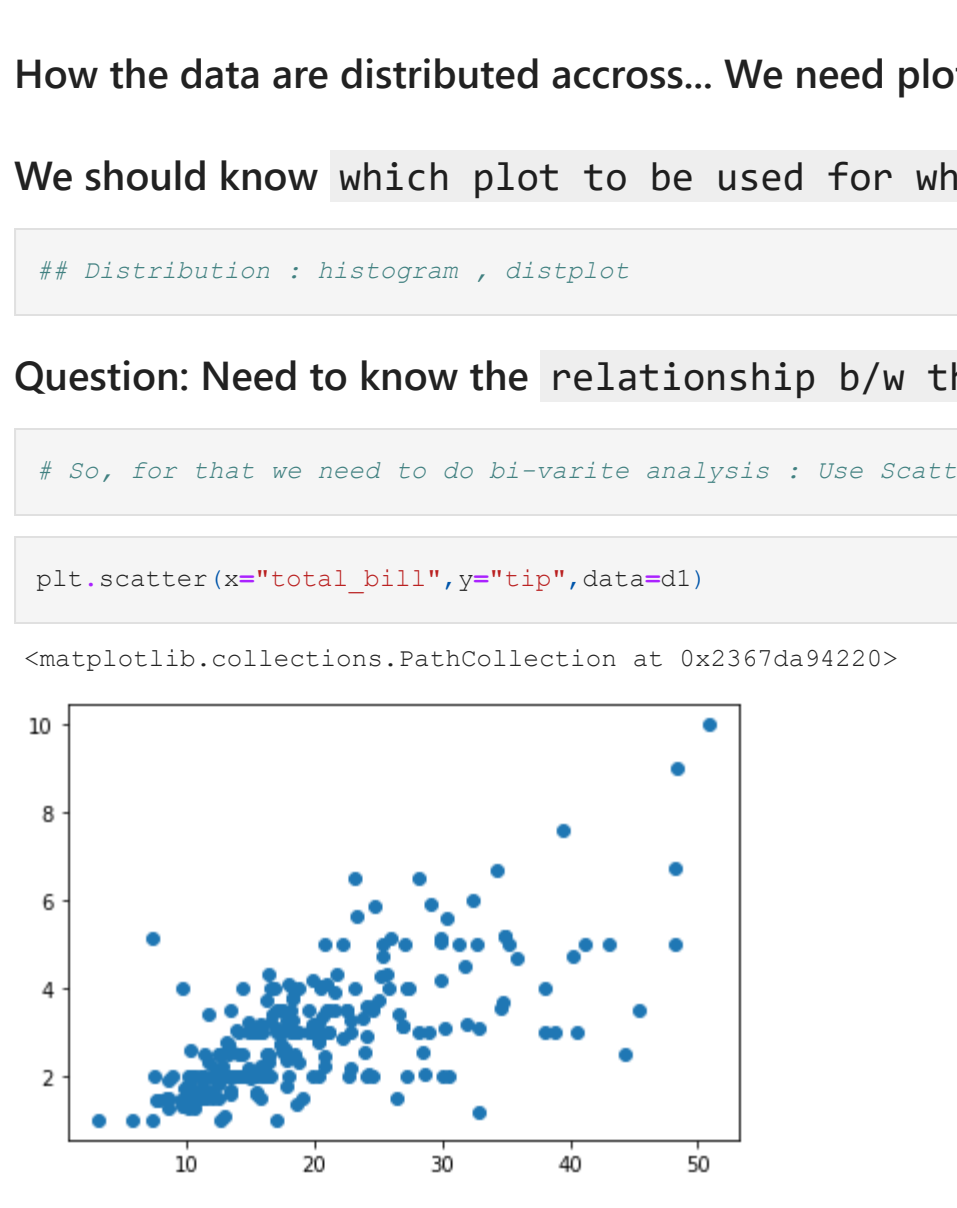
In [17]:
```
# To classify the data based upon which day is more traffic in the restaurent?
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="day")
```

Out[17]: `<seaborn.axisgrid.FacetGrid at 0x2367da945e0>`

1. More traffic on saturday and sunday
2. More tip on saturday and sunday

In [18]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="sex")
```

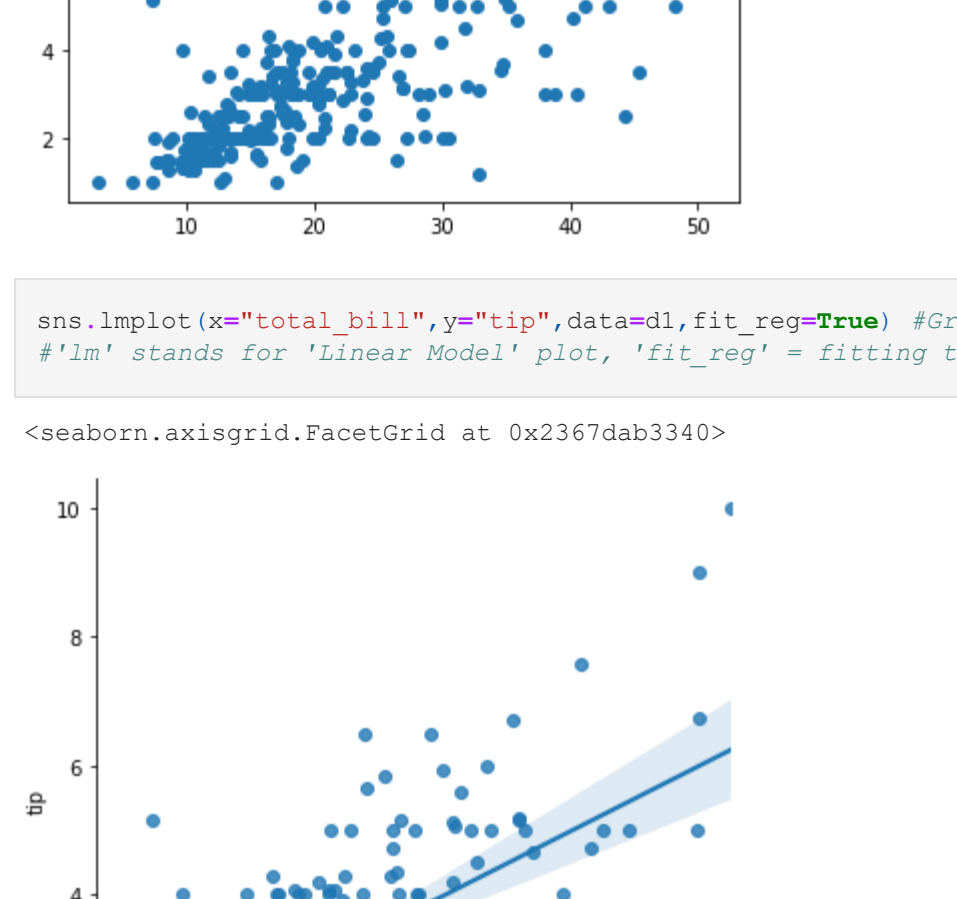
Out[18]: `<seaborn.axisgrid.FacetGrid at 0x23600076c10>`



1. Highest tip was given by "Male'

In [19]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="smoker")
```

Out[19]: `<seaborn.axisgrid.FacetGrid at 0x236000feb50>`

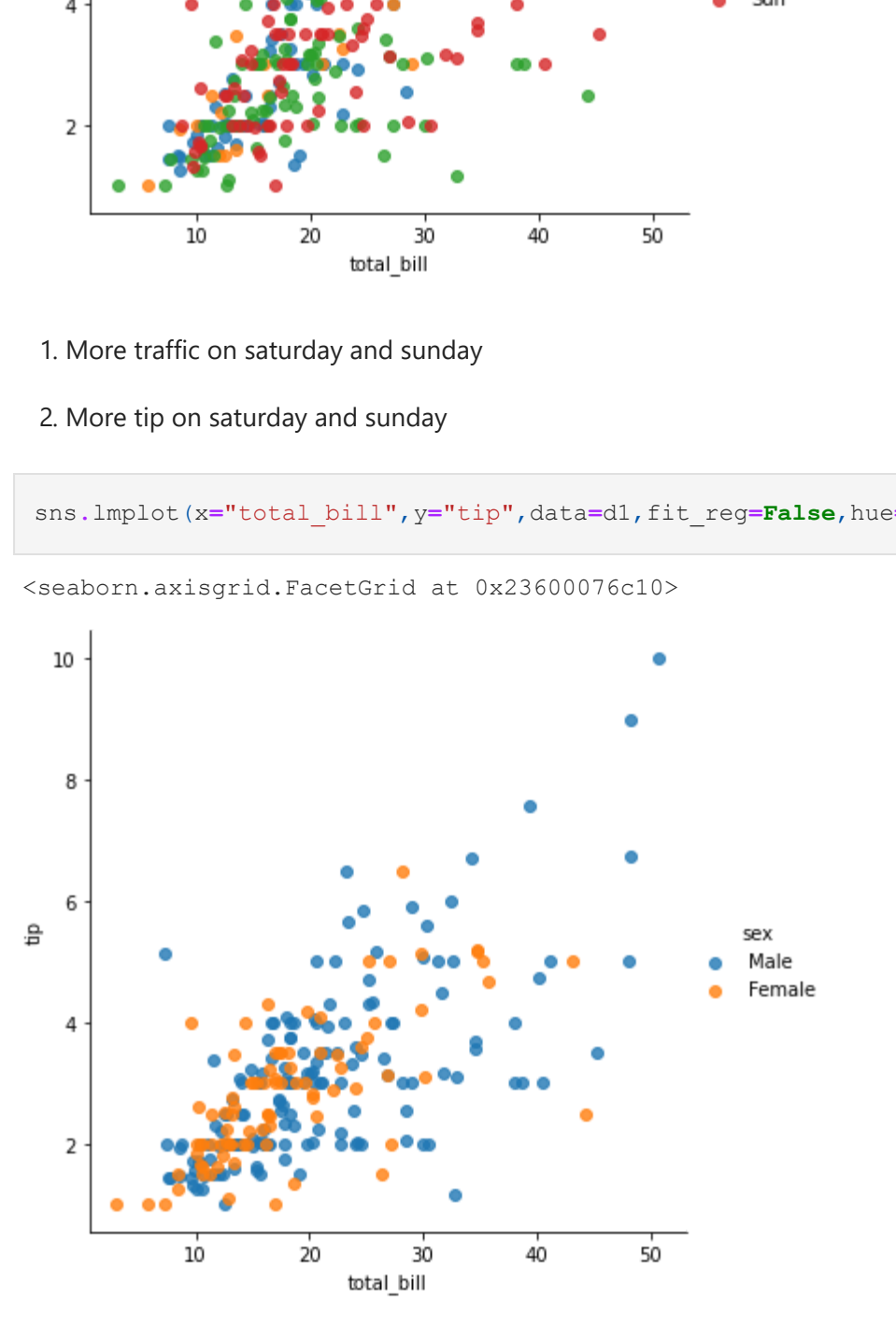


In [20]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="sex",col="smoker")
```

Out[20]: `<seaborn.axisgrid.FacetGrid at 0x236000d77f0>`

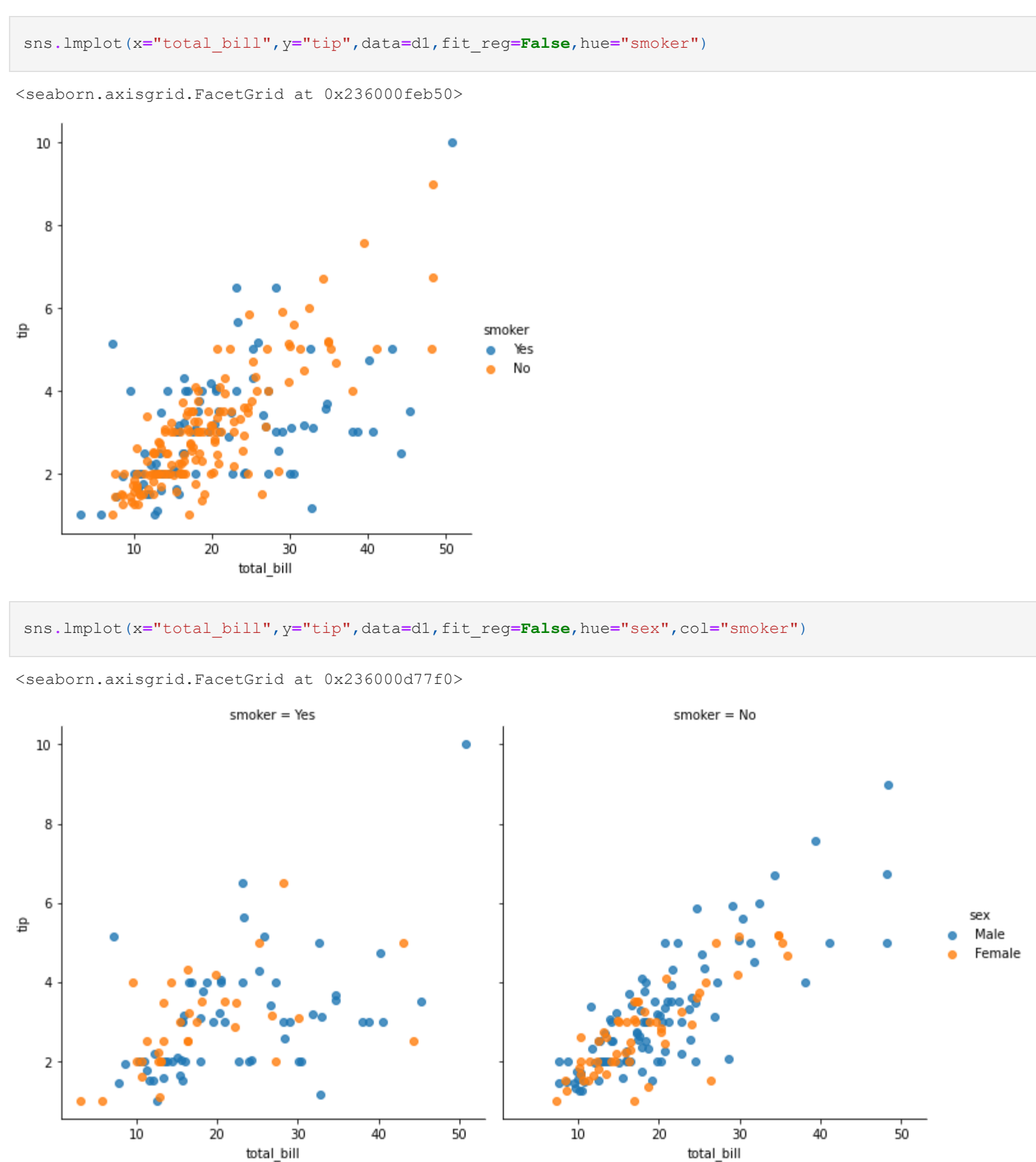
In [21]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="sex",row="smoker")
```

Out[21]: `<seaborn.axisgrid.FacetGrid at 0x2360021c670>`



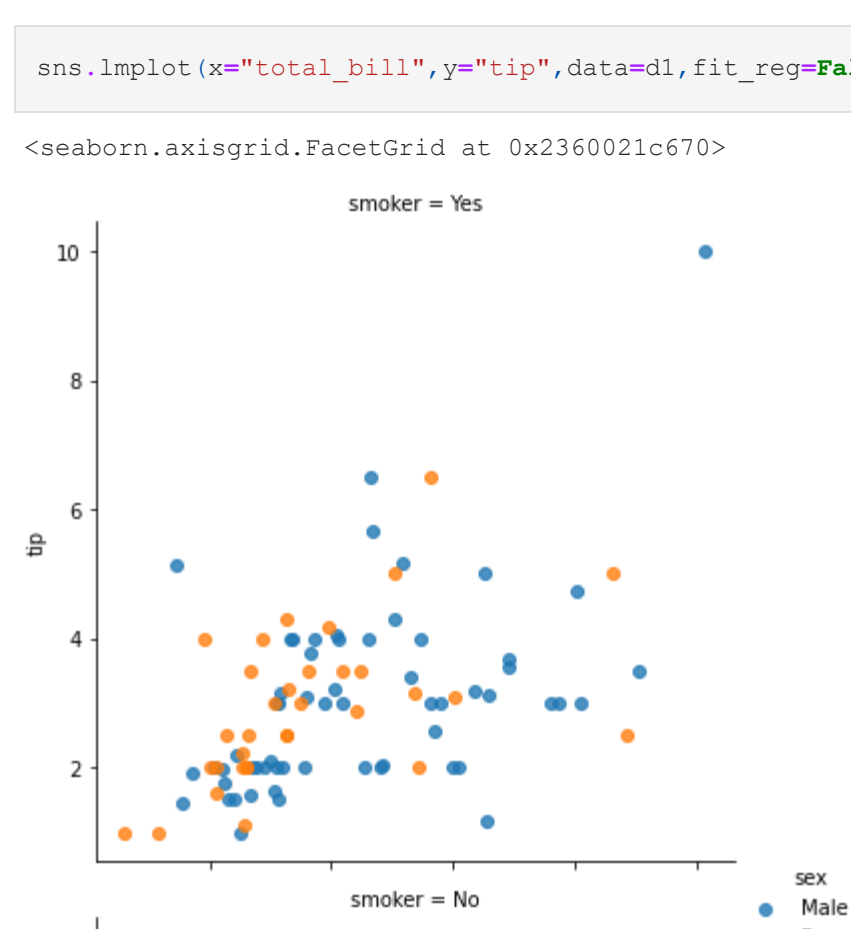
In [22]:
```
sns.lmplot(x="total_bill",y="tip",data=d1,fit_reg=False,hue="sex",row="smoker",col="time")
```

Out[22]: `<seaborn.axisgrid.FacetGrid at 0x236002d1940>`



In [23]:
```
#A violin plot plays a similar role as a box and whisker plot.
#It shows the distribution of quantitative data across several levels of one (or more) categorical variables su
sns.violinplot(x="day",y="total_bill",data=d1)
```

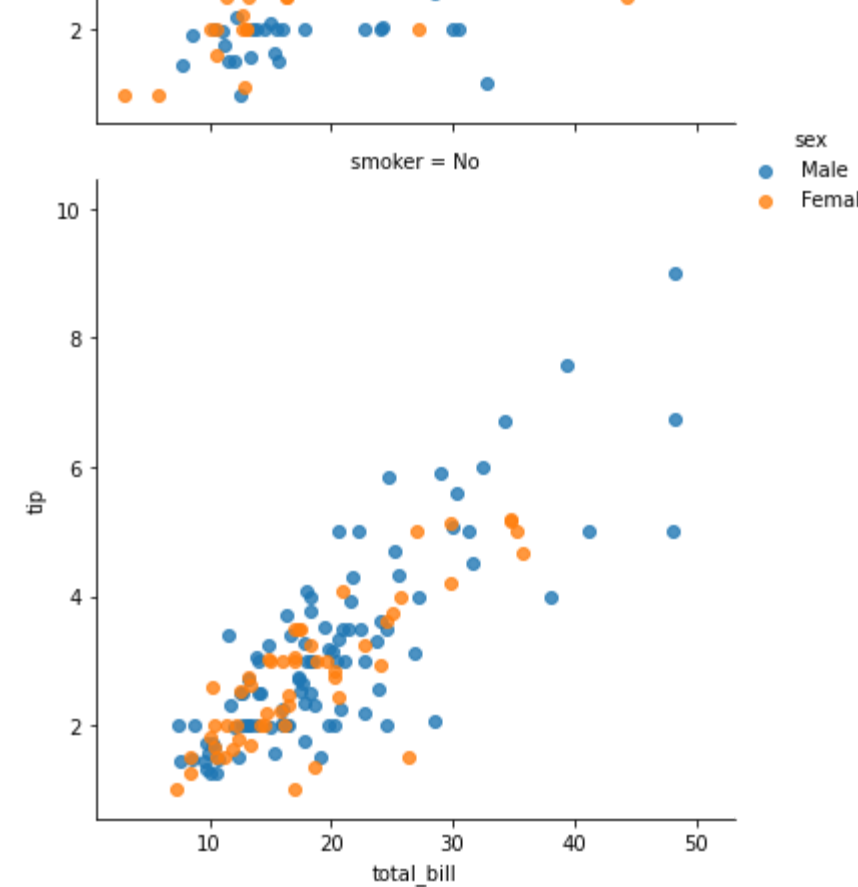
Out[23]: `<AxesSubplot:xlabel='day', ylabel='total_bill'>`



In [24]:
```
# Map density on the basis of total_bill w.r.t to each day as in on which denomination each entry are their

# We used this when we have categorical adn discreate data
sns.stripplot(x="day",y="total_bill",data=d1,jitter=True)
```

Out[24]: `<AxesSubplot:xlabel='day', ylabel='total_bill'>`

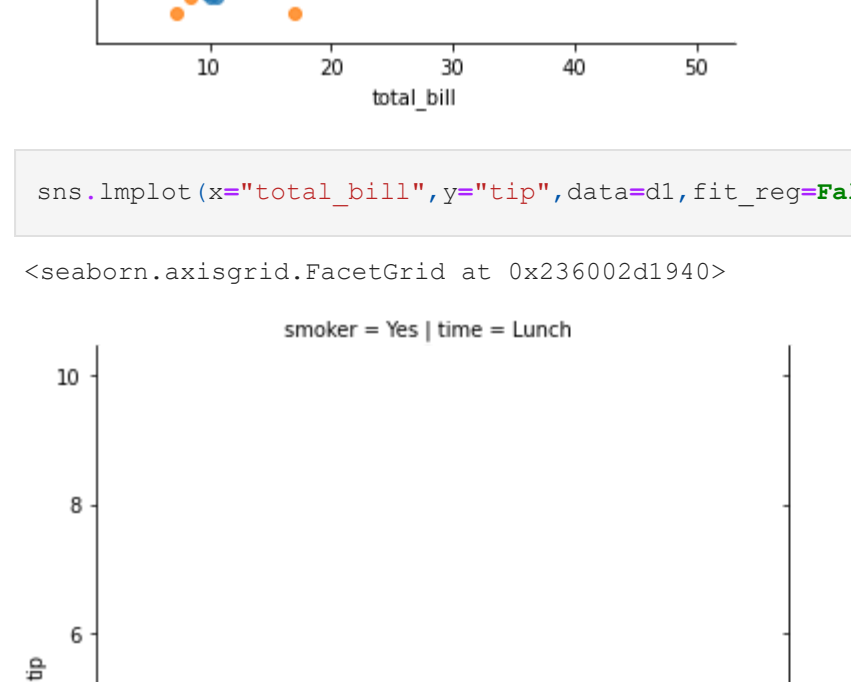


1. At 20 dollar we can see the count in the restaurent for each day.

Probem with this plot is that we cannot identiy which day has more count of 20 dolloar bill

In [25]:
```
sns.stripplot(x="day",y="total_bill",data=d1)
```

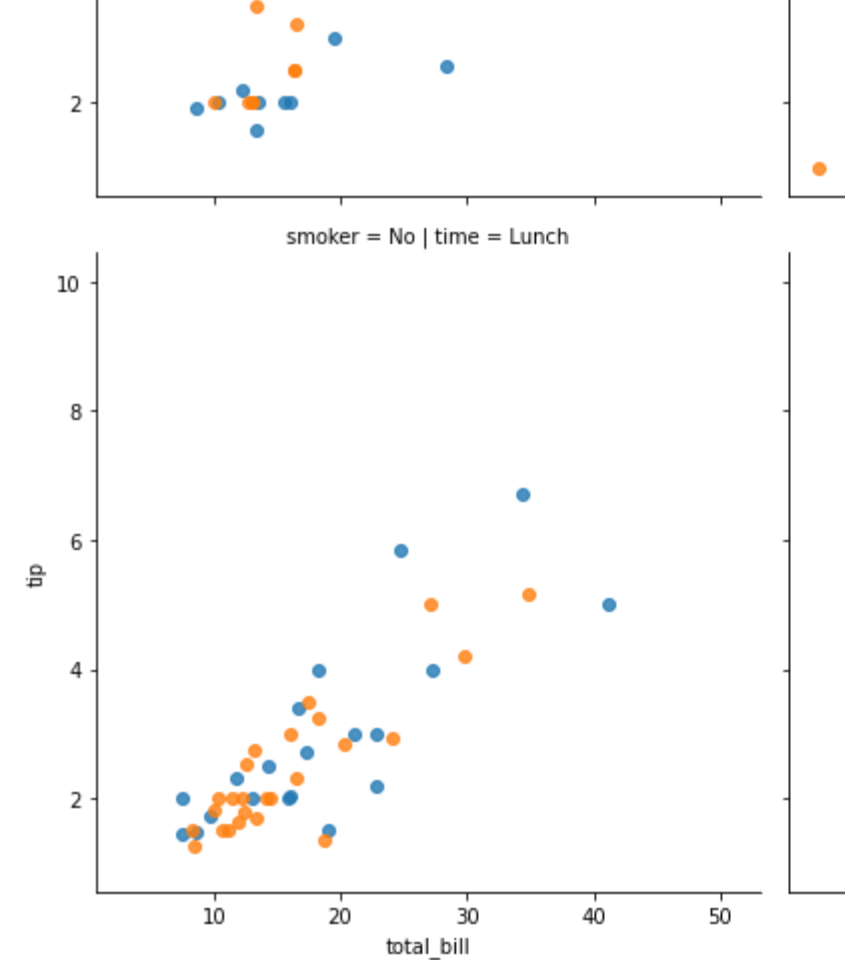
Out[25]: `<AxesSubplot:xlabel='day', ylabel='total_bill'>`



To get the count of each day w.r.t to `20 dollar` bill

In [26]:
```
sns.swarmplot(x="day",y="total_bill",data=d1)
```

Out[26]: `<AxesSubplot:xlabel='day', ylabel='total_bill'>`

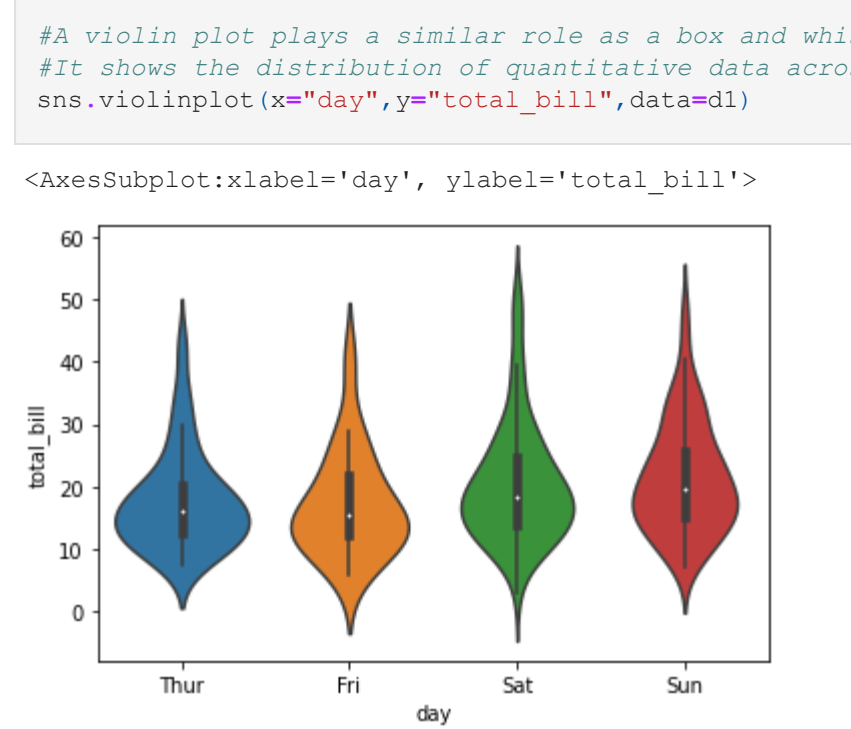



```
Points are not getting overlapped i.e. on `20 dollar` we can map how many entries are their so that we
can see which day can win as `Saturday` has more width as compare to other days
```

In [27]:
```
sns.distplot(d1["total_bill"],kde= False) ## Used for Un-variate data
```

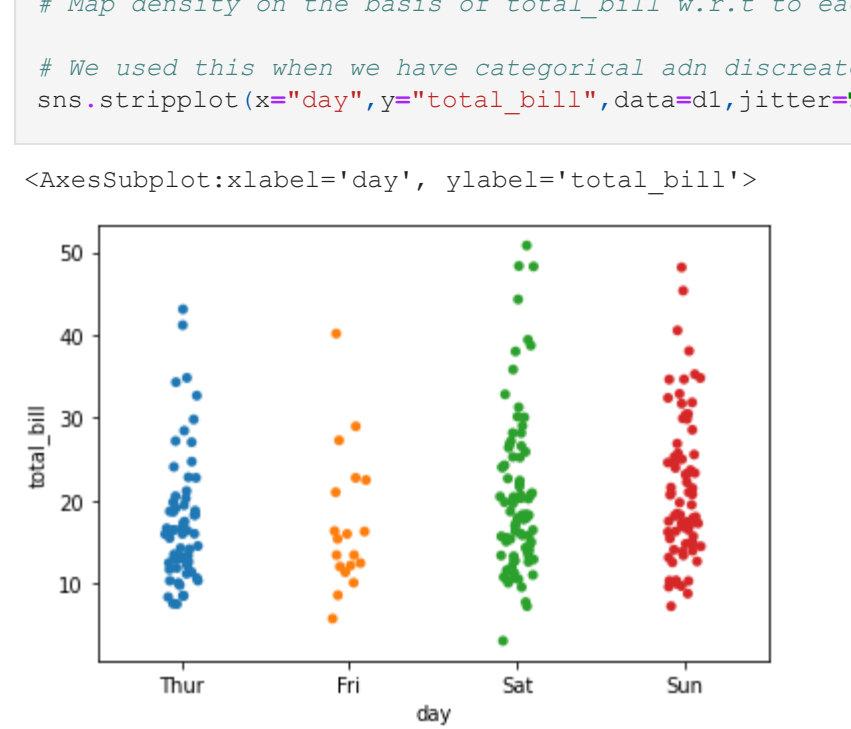
```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depreca
ted function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-
level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[27]: `<AxesSubplot:xlabel='total_bill'>`



In [28]:
```
sns.distplot(d1["total_bill"]) ## Used for Un-variate data
```

Out[28]: `<AxesSubplot:xlabel='total_bill', ylabel='Density'>`



Tells us about the shape of the data or how the data is distributed

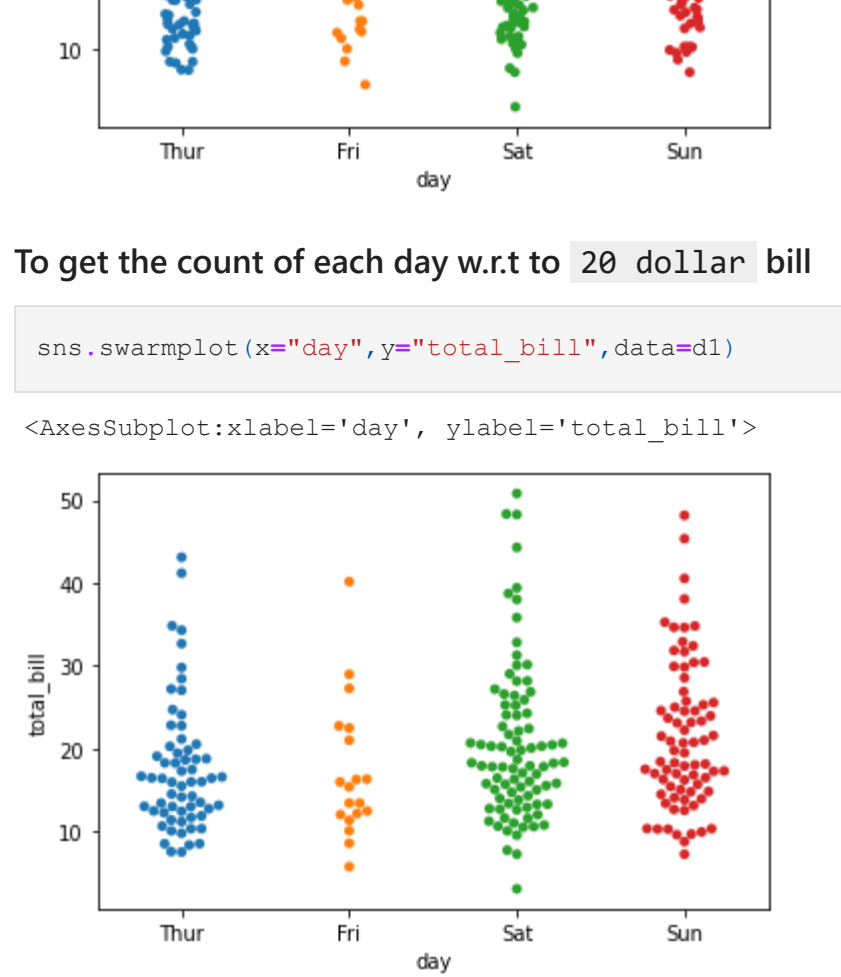
Line in the plot is made w.r.t `KDE` function i.e. `Kernal Density Estimator`

`KDE` Area Under the Curveor Probability

In [29]:
```
sns.distplot(d1["tip"])
```

```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depreca
ted function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-
level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[29]: `<AxesSubplot:xlabel='tip', ylabel='Density'>`



In [30]:
```
d1.skew()
```

Out[30]:
```
total_bill    1.133213
tip           1.465451
size          1.447882
dtype: float64
```

Heat Graph

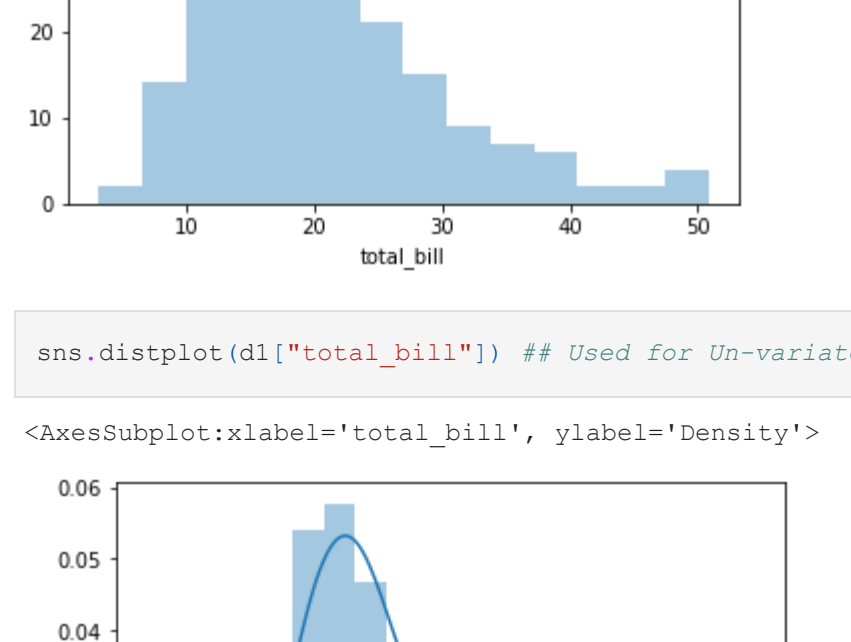
Heat Map is used whene used to get the correlation in the data

In [31]:
```
import numpy as np

a=np.random.randint(1,100,25).reshape(5,5)
a
```

Out[31]:
```
array([[70, 52, 51, 47, 88],
       [23, 11, 59, 15, 24],
       [54, 42,  7, 20, 34],
       [21, 69, 42, 46, 21],
       [91, 91, 23, 48, 52]])
```

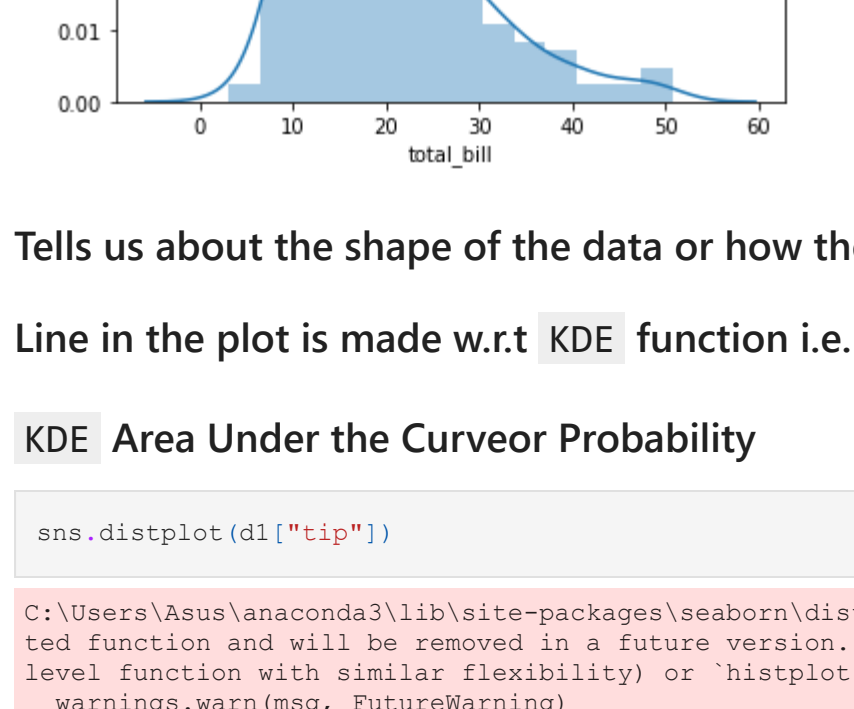
In [32]:
```
sns.heatmap(a,annot=True) #'annot' to get annotation ie. numbers
plt.show()
```



Here color intensity depands upon the value of the cell i.e. if the value is `lowest then color in darkest`

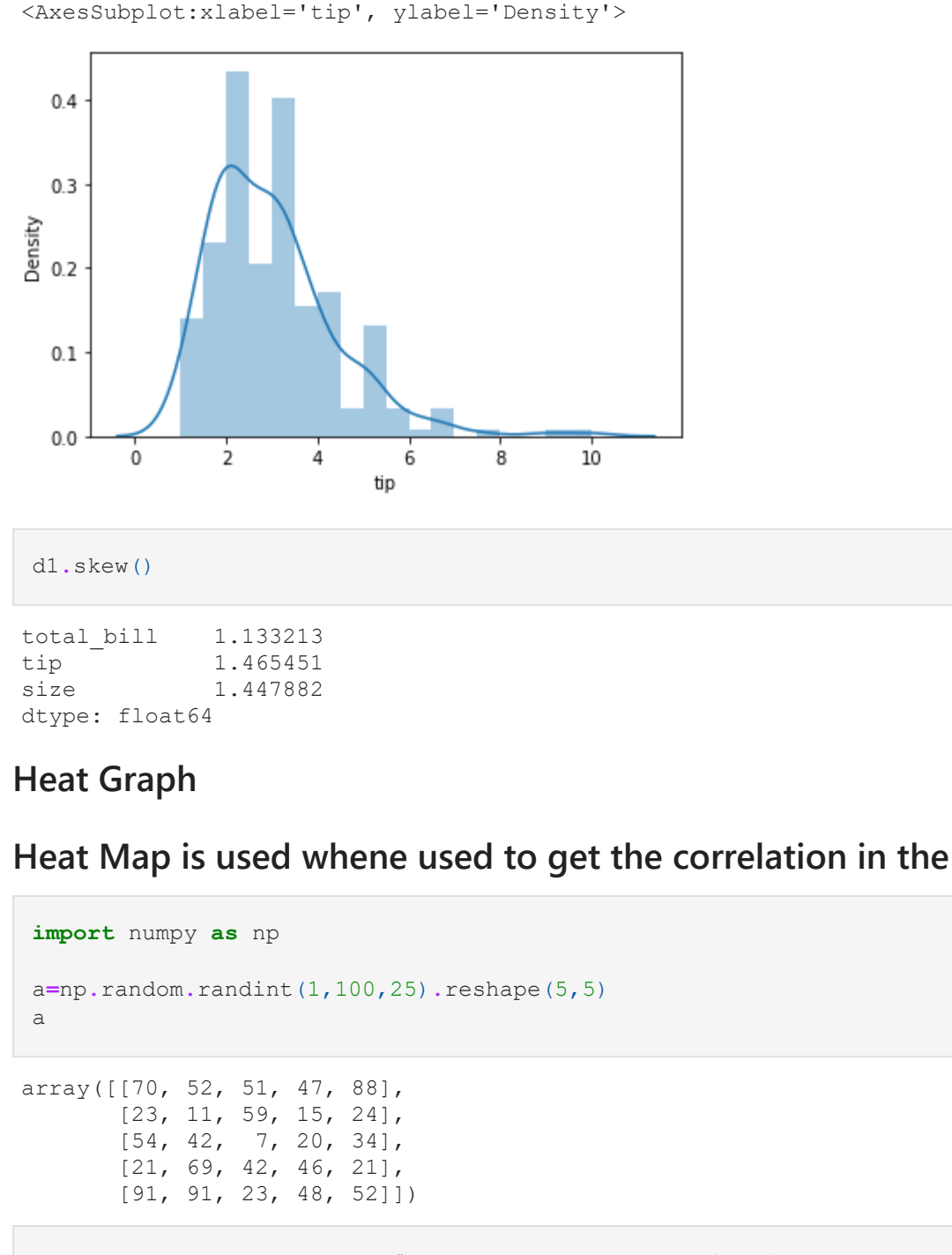
In [33]:
```
sns.boxplot(y="total_bill",x="day",data=d1)
```

Out[33]: `<AxesSubplot:xlabel='day', ylabel='total_bill'>`



In [34]:
```
# To show the uni-variate vs Bi-varaite Analysis under one plot
sns.jointplot(x="total_bill",y="tip",data=d1)
```

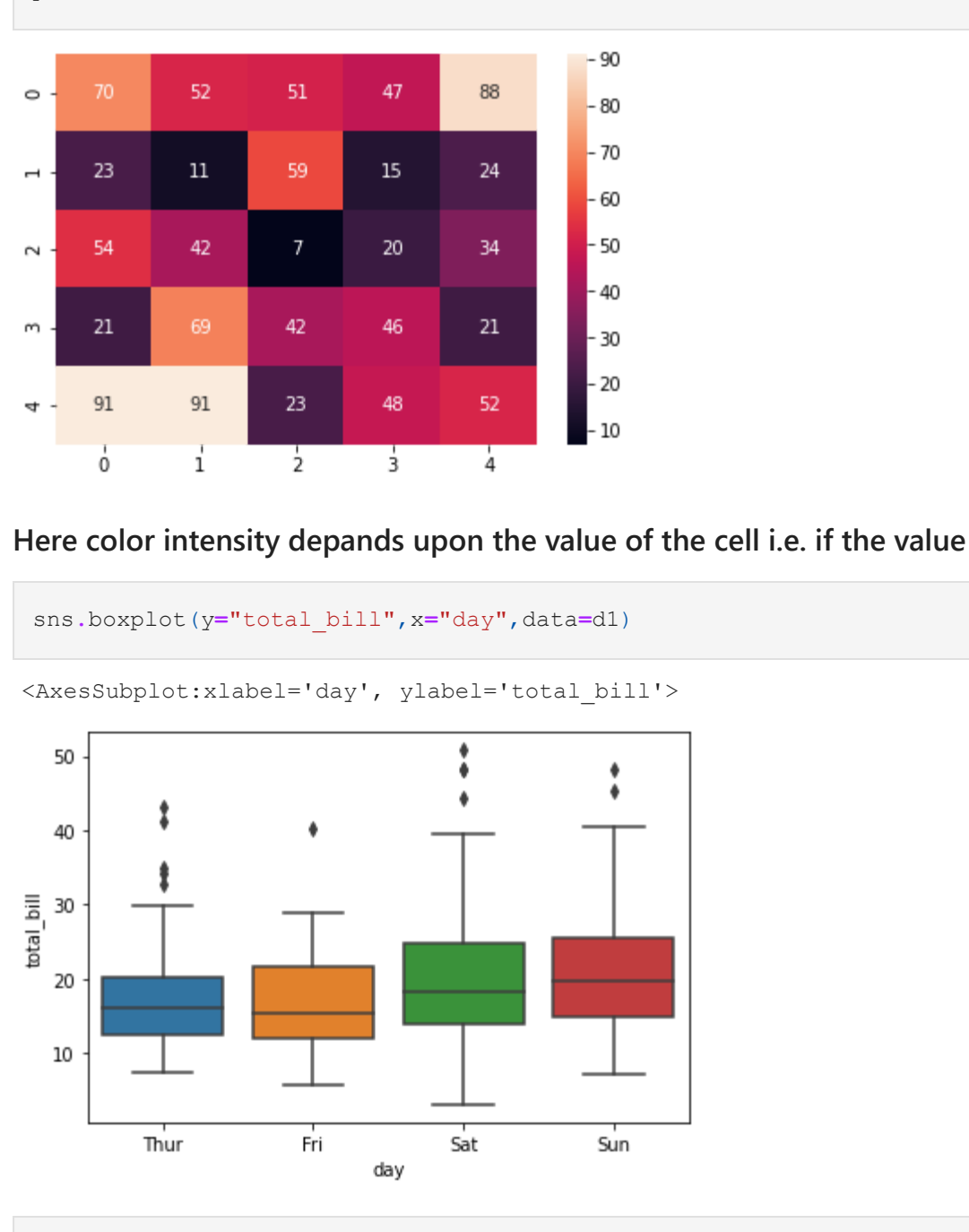
Out[34]: `<seaborn.axisgrid.JointGrid at 0x23601f6f4c0>`



1. Relationship b/w total bill and tip using scatter plot
2. Top one is the histogram for total bill
3. Right one is the histogram for tip

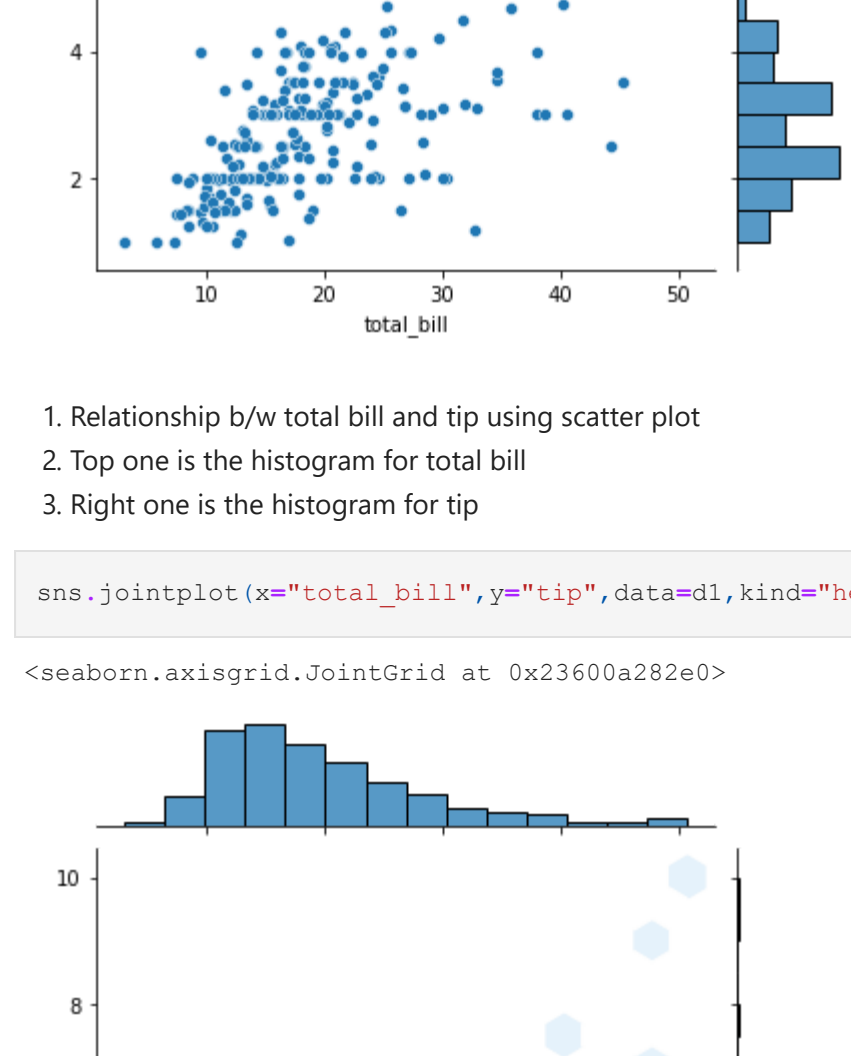
In [35]:
```
sns.jointplot(x="total_bill",y="tip",data=d1,kind="hex") #'hex' = hexagons with dark area with highest density
```

Out[35]: `<seaborn.axisgrid.JointGrid at 0x23600a282e0>`



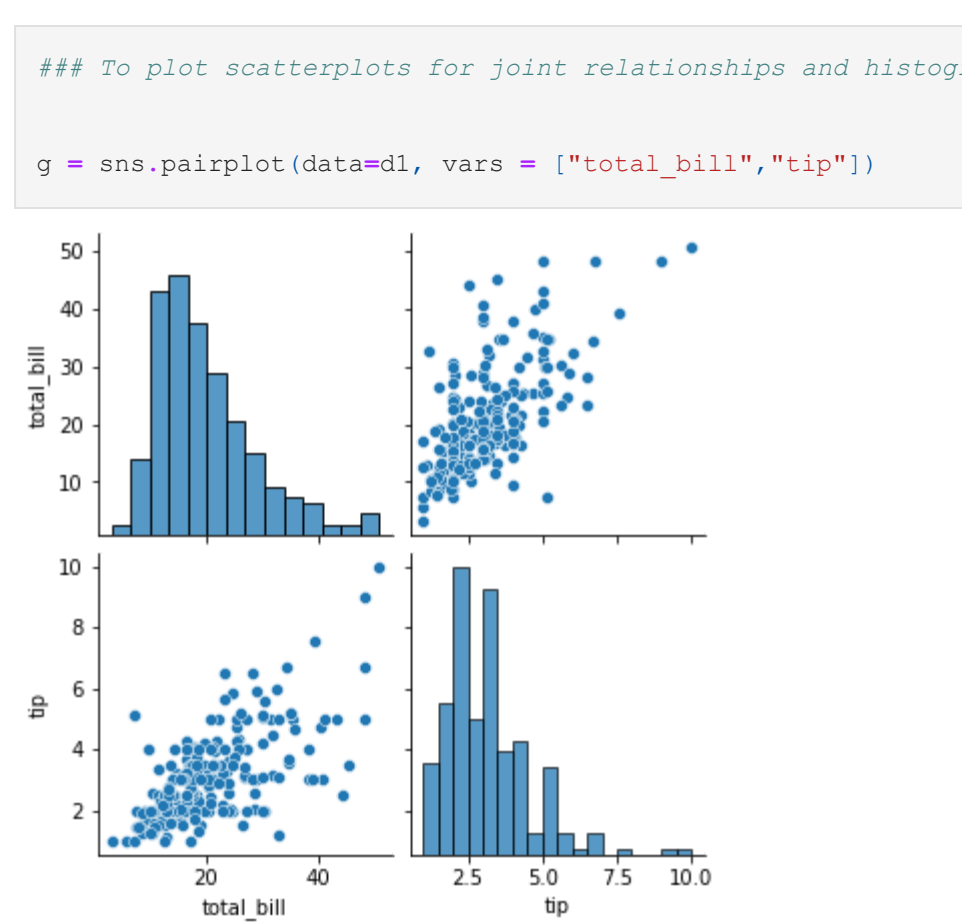
In [36]:
```
### To plot scatterplots for joint relationships and histograms for univariate distributions:


g = sns.pairplot(data=d1, vars = ["total_bill","tip"])
```



In [37]:
```
# Count the frequency of categorical values w.r.t to other var
sns.countplot(x="sex",data=d1,hue="smoker")
plt.show()

#'hue' to compare smokers w.r.t to male and female
```



Question: Avg bll paid by `smokers and non-smokers`

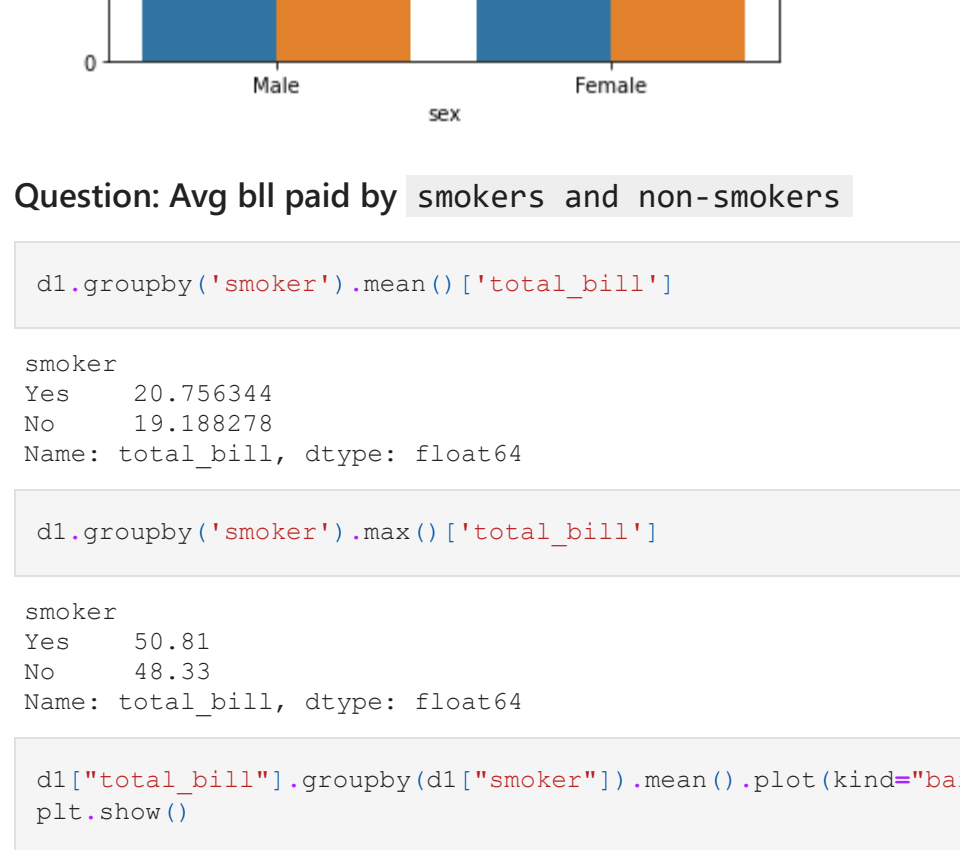
In [38]:
```
d1.groupby('smoker').mean()['total_bill']
```

Out[38]:
```
smoker
Yes    20.756344
No     19.188278
Name: total_bill, dtype: float64
```

In [39]:
```
d1.groupby('smoker').max()['total_bill']
```

Out[39]:
```
smoker
Yes    50.81
No     48.33
Name: total_bill, dtype: float64
```

In [40]:
```
d1["total_bill"].groupby(d1["smoker"]).mean().plot(kind="bar")
plt.show()
```



In [41]:
```
t=d1["total_bill"].groupby(d1["smoker"]).max()
```

```
plt.figure(figsize=(6,4))
sns.countplot(x='Gender', data=df)
plt.title('Gender Distribution')
plt.show()
```

```
<ipython-input-11-1e1e1e1e1e1e>:2: FutureWarning: 

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x='Gender', data=df)
```