Learn_Seaborn

January 18, 2023

0.1 Setup

[2]:

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

%reload_ext autoreload
%autoreload 2
%matplotlib inline
```

0.2 Import Data

[3]:

```
# df = pd.read_csv()
crash_df = sns.load_dataset('car_crashes')
crash_df.head(10)
```

[3]:

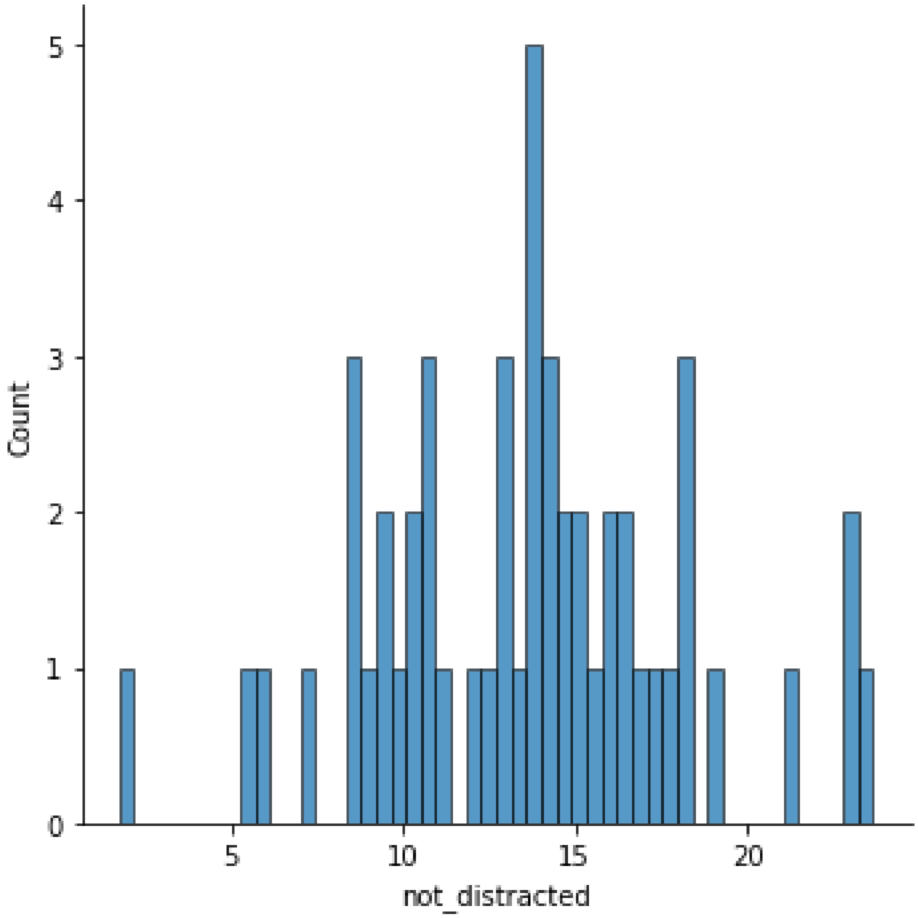
```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium  \
0   18.8     7.332    5.640          18.048       15.040       784.55
1   18.1     7.421    4.525          16.290       17.014      1053.48
2   18.6     6.510    5.208          15.624       17.856       899.47
3   22.4     4.032    5.824          21.056       21.280       827.34
4   12.0     4.200    3.360          10.920       10.680       878.41
5   13.6     5.032    3.808          10.744       12.920       835.50
6   10.8     4.968    3.888           9.396        8.856      1068.73
7   16.2     6.156    4.860          14.094       16.038      1137.87
8    5.9     2.006    1.593           5.900        5.900      1273.89
9   17.9     3.759    5.191          16.468       16.826      1160.13

   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
3      142.39     AR
4      165.63     CA
5      139.91     CO
6      167.02     CT
7      151.48     DE
```

```
8       136.05      DC
9       144.18      FL
```

1 Distribution Plots

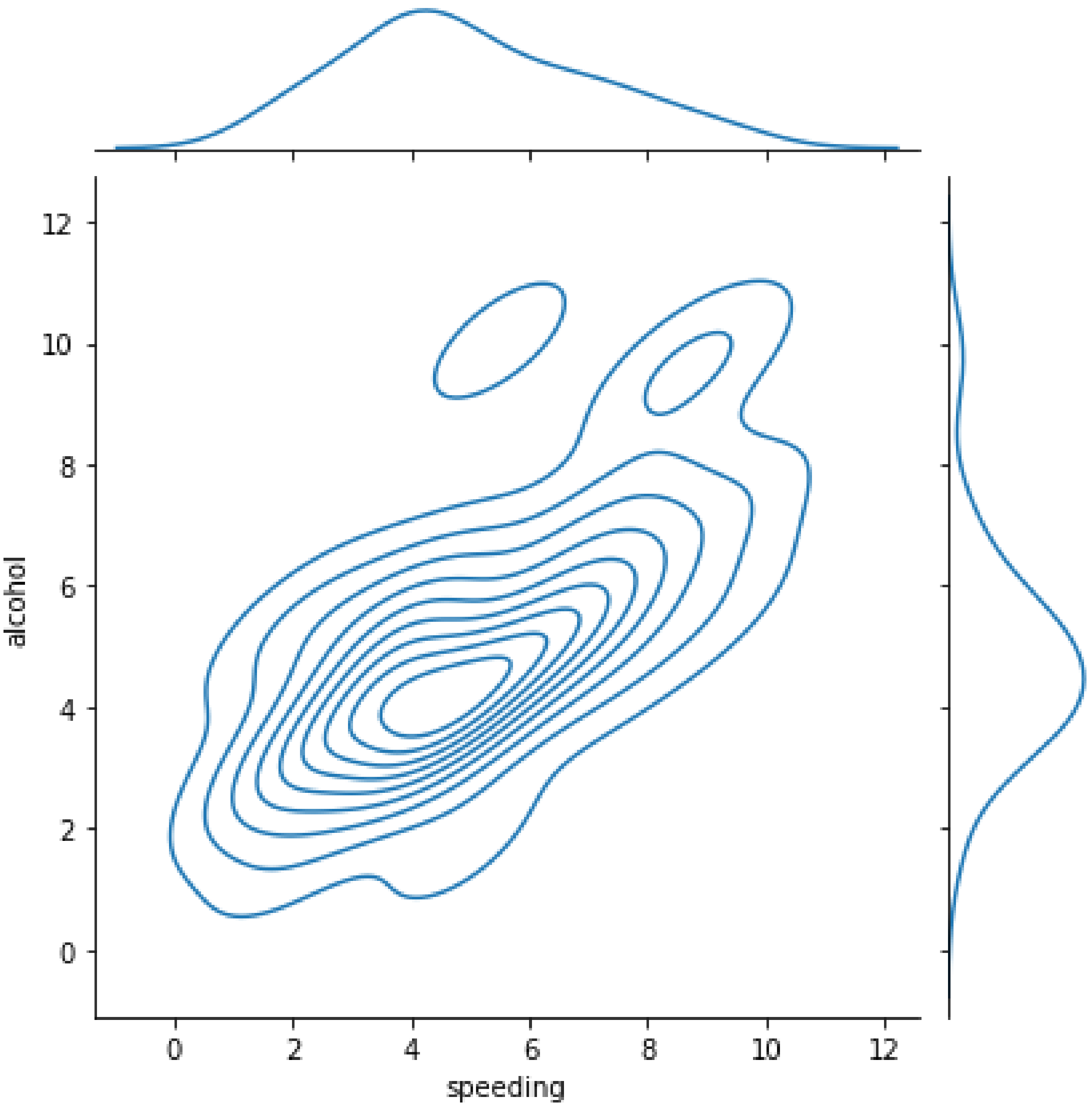
1.1 Distribution Plot

[4]:
```
# sns.distplot(crash_df['not_distracted'])
sns.displot(crash_df['not_distracted'],kde=False,bins=50)
plt.show()
```



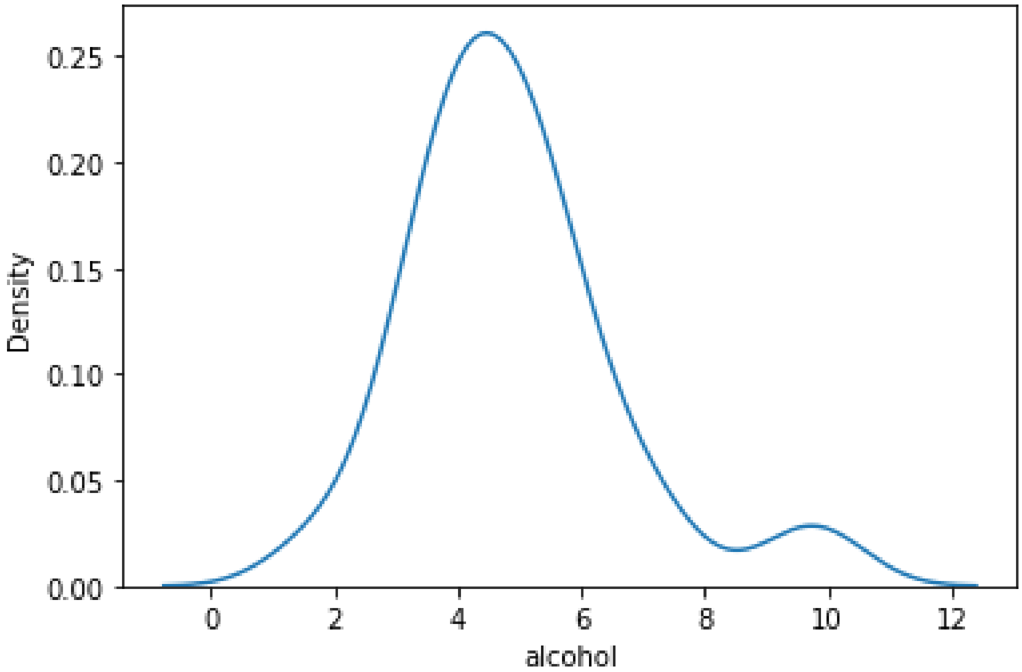
1.2 Joint Plot

[5]:
```
# compare two distribution and plot a scatter plot by default
sns.jointplot(x='speeding',y='alcohol',data=crash_df,kind='kde') # king =␣
  ↪['kde','hex','reg']
plt.show()
```



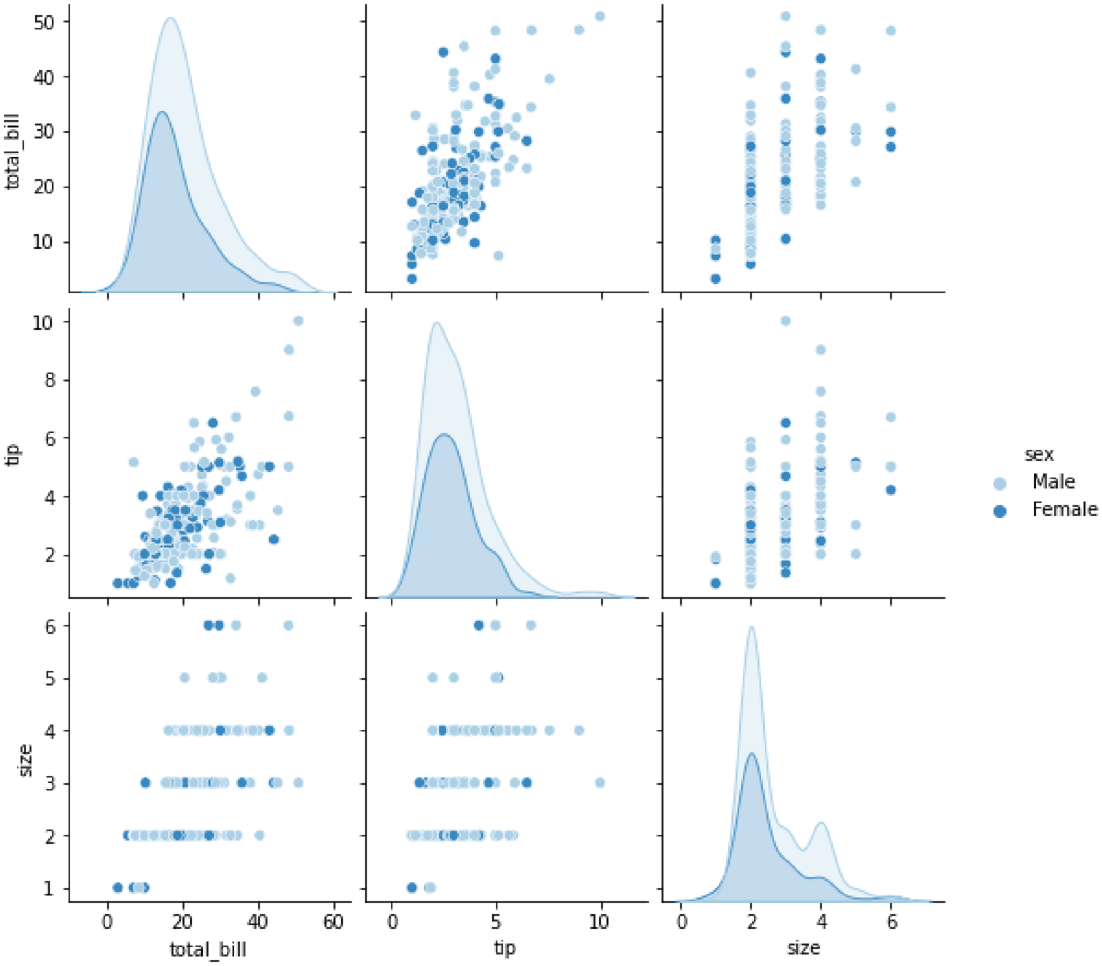
1.3 KDE Plot

[6]:
```
sns.kdeplot(crash_df['alcohol'])
plt.show()
```



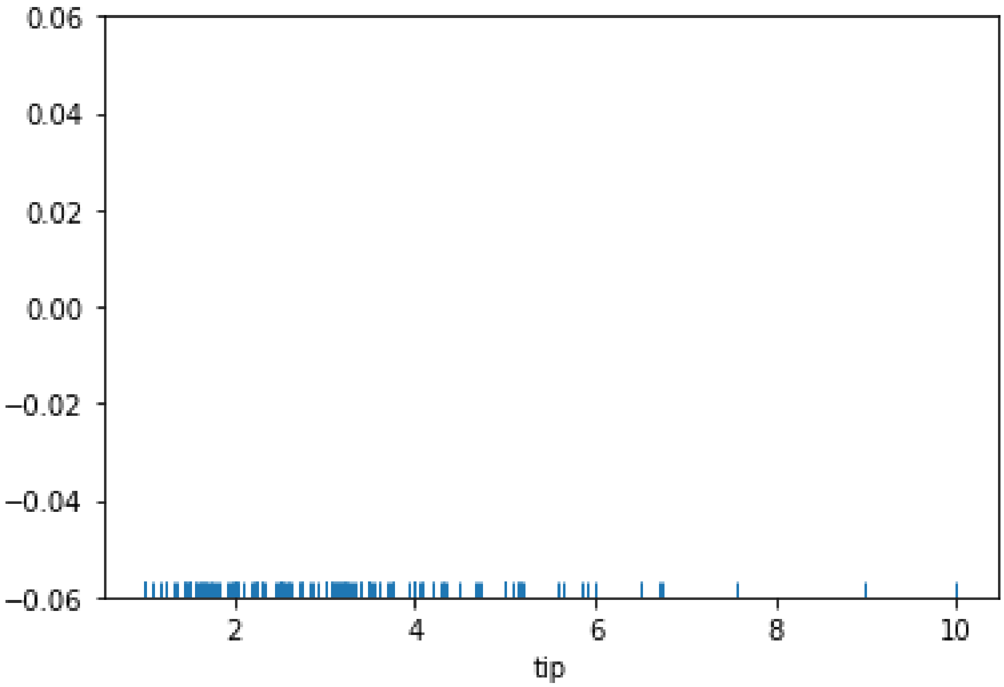
1.4 Pair Plot

[7]:
```
# It plot the the relationship across entire dataframe and  numerical values
# sns.pairplot(crash_df)
tips_df = sns.load_dataset('tips')
sns.pairplot(tips_df,hue='sex',palette='Blues') # hue :-> with hue we can pass␣
 ↪the categorial data and chart will be colorised based on that data
plt.show()
```



1.5 Rug Plot

```
[8]: # It plots the a sigle column of datapoints in a dataframe as sticks on the x␣
     ↪axis, it show more dense number of lines where the amount is more common
     sns.rugplot(tips_df['tip'])
     plt.show()
```

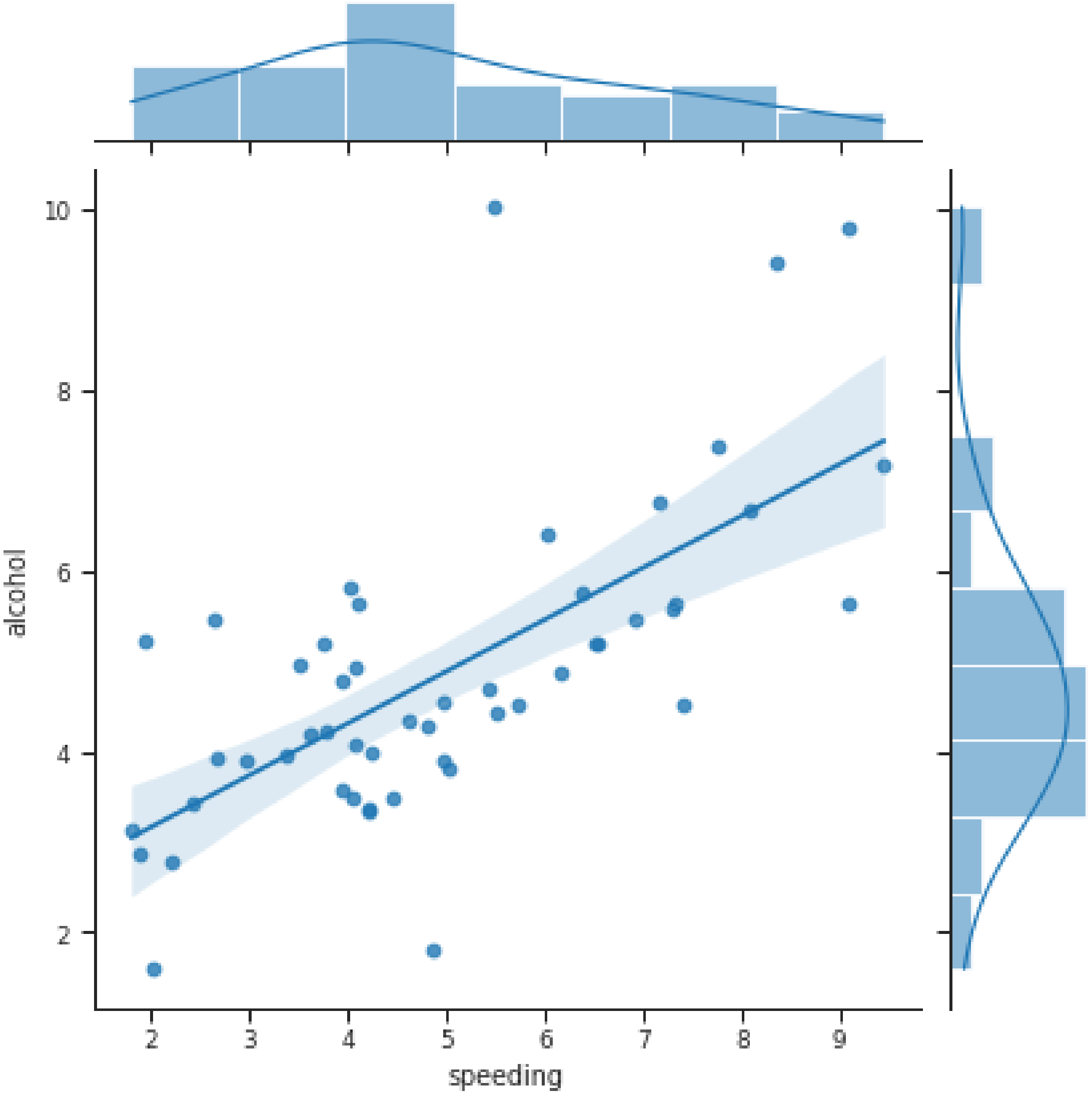


1.6 Styling

[9]:

```
sns.set_style('ticks') # style= ['white','whitegrid',dark,ticks,'darkgrid']
sns.set_context('paper',font_scale=1.0) # [paper,talk,poster]
plt.figure(figsize=(2,2))
sns.jointplot(x='speeding',y='alcohol',data=crash_df,kind='reg')

# sns.despine(left=True,bottom=True) # [right, top]
plt.show()
```

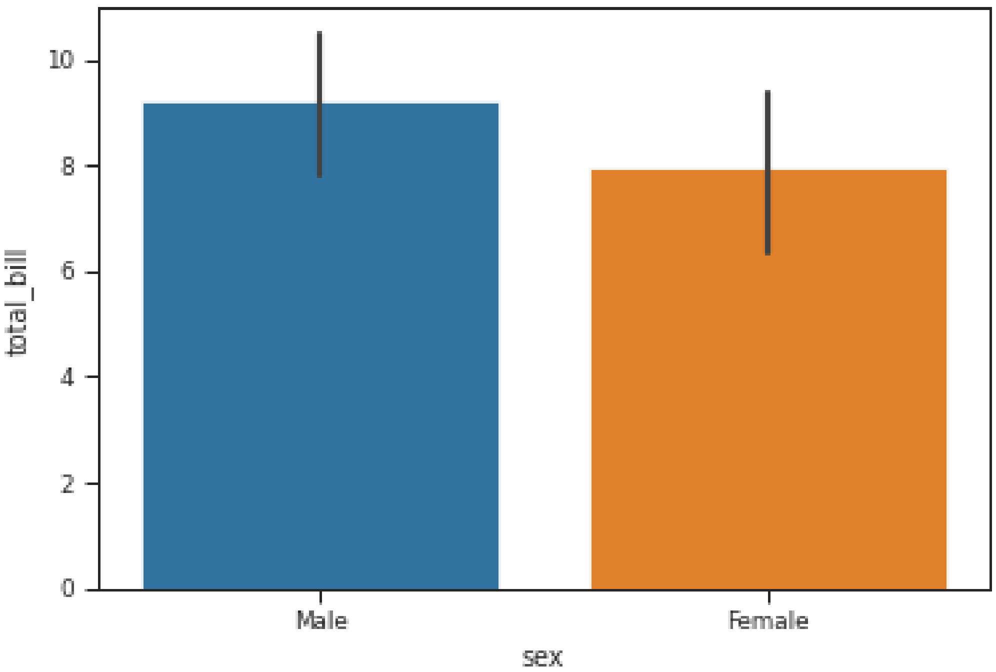
```
<Figure size 144x144 with 0 Axes>
```



2 Categorial Plots

2.1 Bar Plot

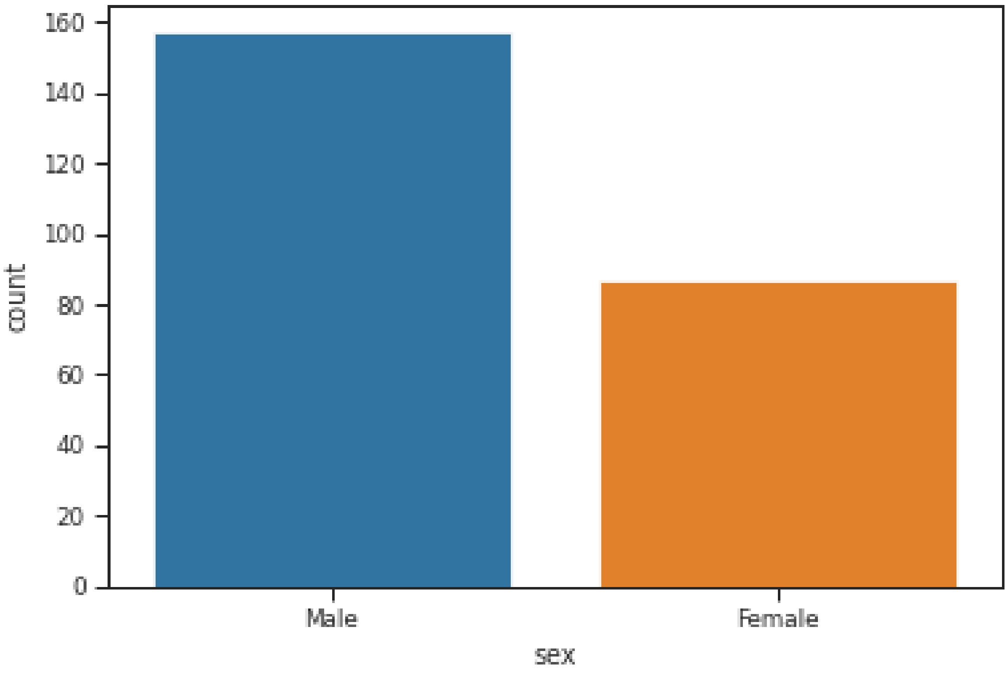
```
[10]: sns.barplot(x='sex',y='total_bill',data=tips_df,estimator=np.std) #␣
       ↪estimator=[median,mean(default),np.std(StandardDeviation), np.
       ↪var(varience),np.cov(coverience),custom]
      plt.show()
```



2.2 Count Plot

```
[11]: # kind of bar plot but estimator is just count the number of occurances
      sns.countplot(x='sex',data=tips_df)
```

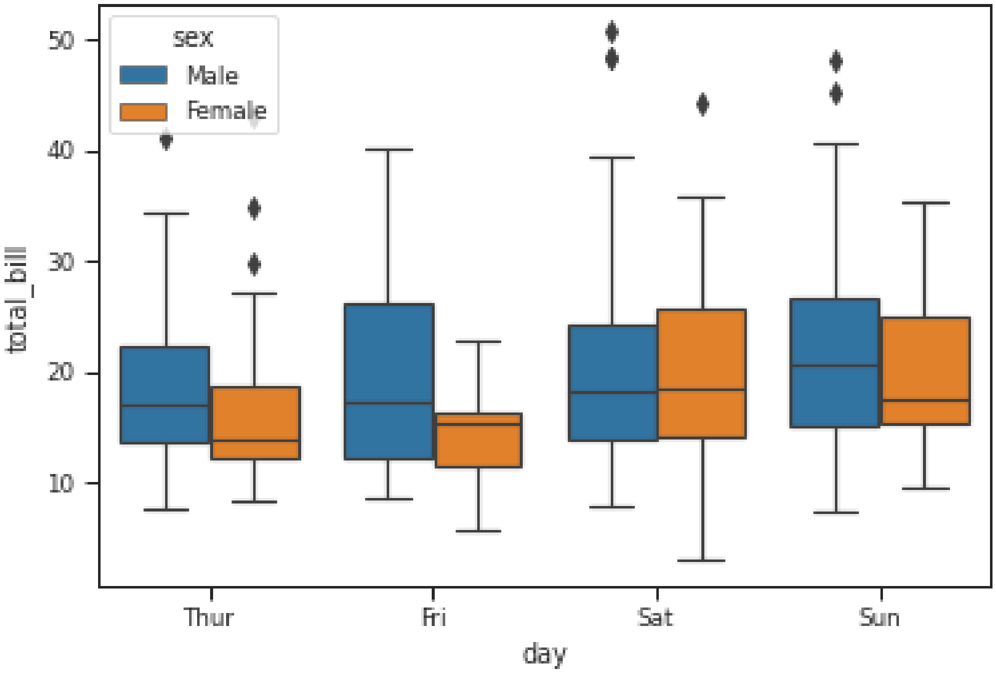
```
[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff4add963a0>
```



2.3 Box Plot

```
[12]: # A box plot is going to compare the different variables qurtile of the data
      sns.boxplot(x='day',y='total_bill',data=tips_df,hue='sex')
```

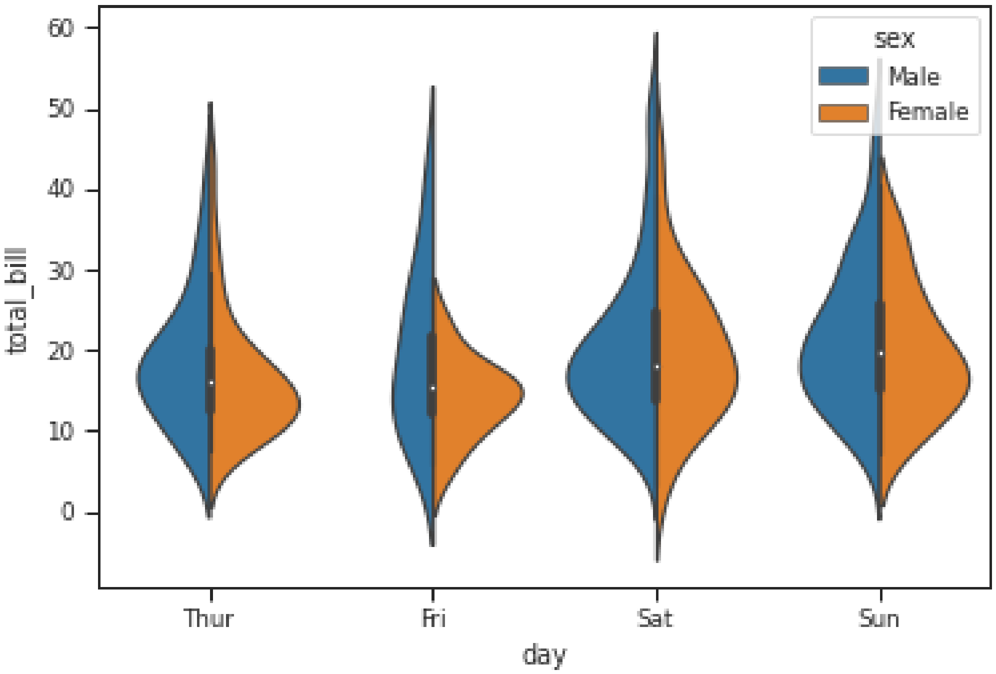
```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff4add9f460>
```



2.4 Violin Plot

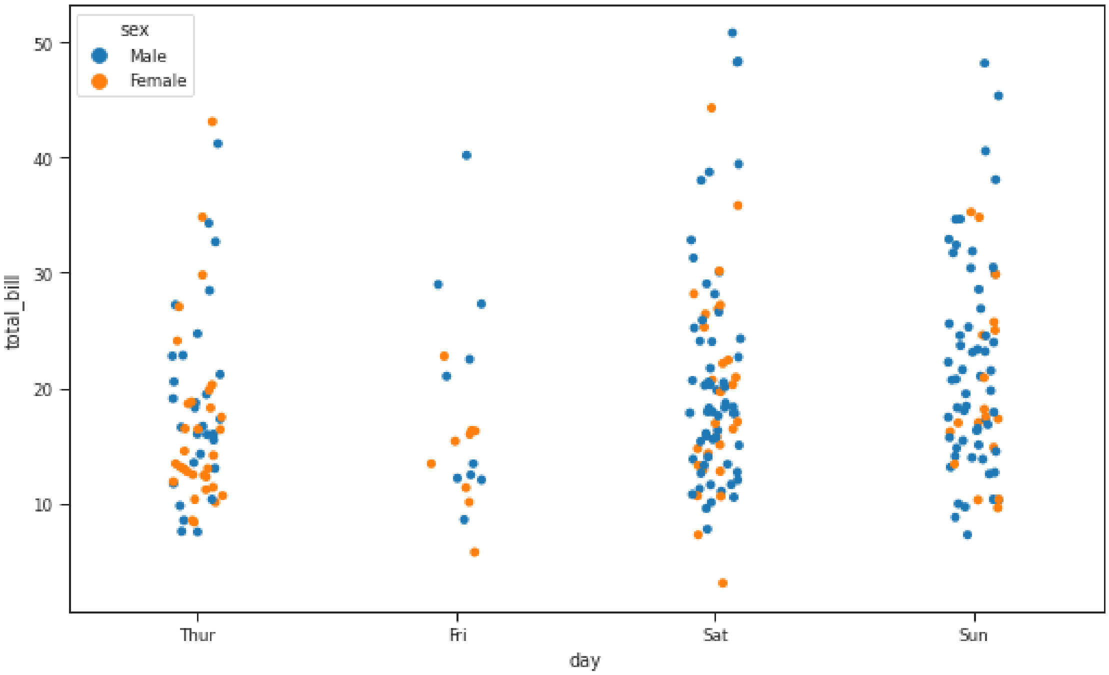
```
[13]: # it is combination of box plot and a kde plot
      # a violin plot uses the kde estimation of data points
      sns.violinplot(x='day',y='total_bill',data=tips_df,hue='sex',split=True)
```

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff4b1b82280>
```



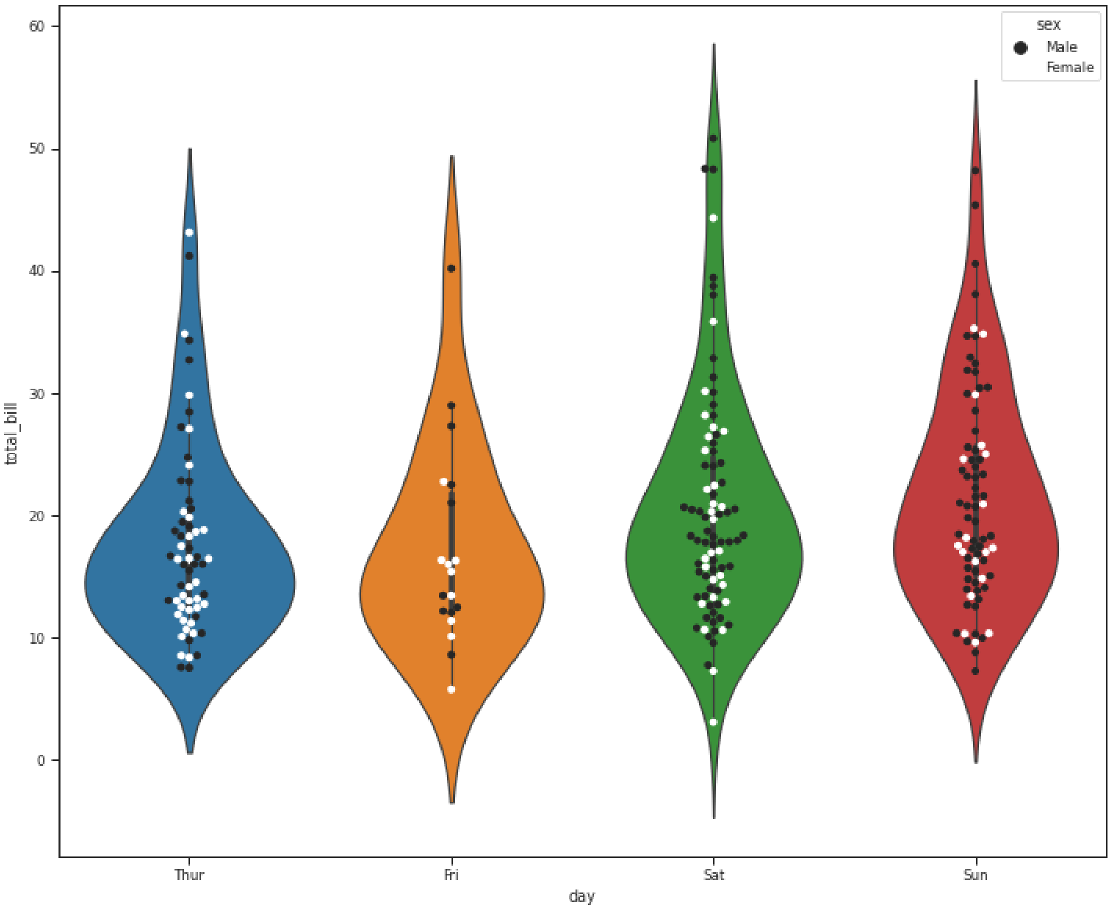
2.5 Strip Plot

[14]:
```
# strip plot draws the scatter plot represents all data points where one␣
 ↪variable is categorical
plt.figure(figsize=(10,6))
sns.stripplot(x='day',y='total_bill',data=tips_df,jitter=True,hue='sex') #␣
 ↪,dodge=True :-> seprates the data
plt.show()
```



2.6 Swarm Plot

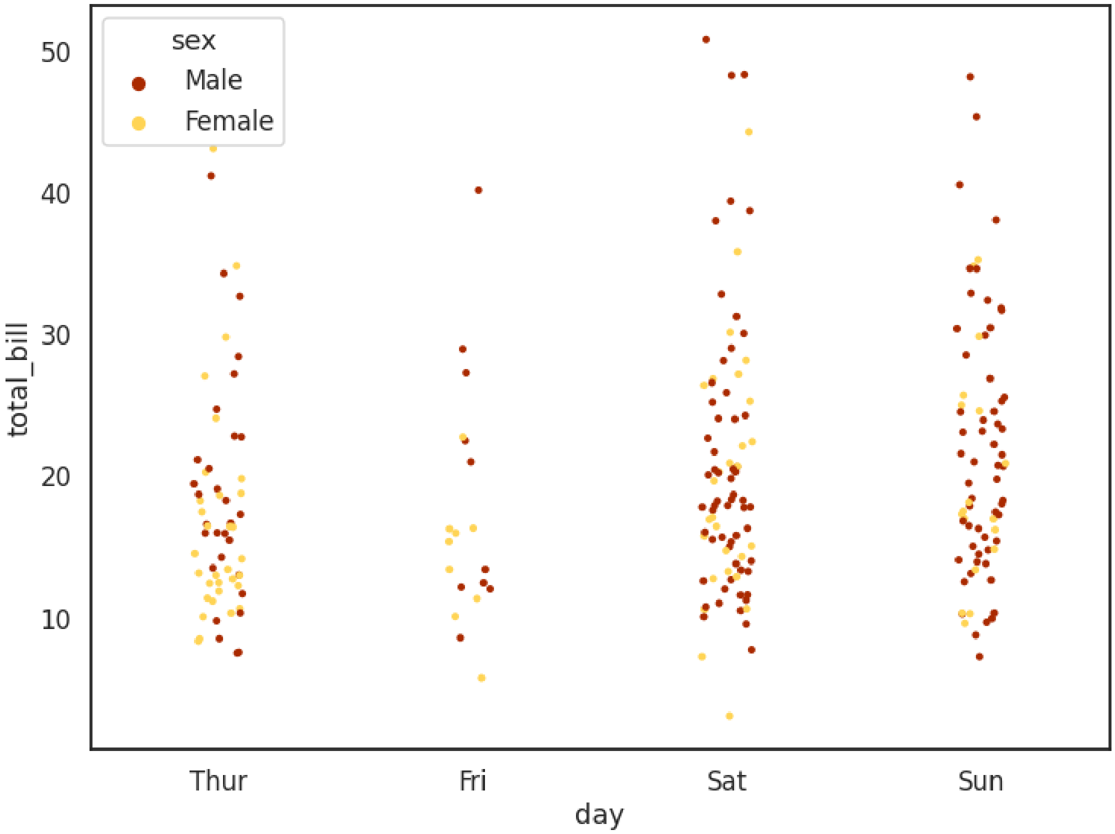
[15]:
```
# It is a combination of violin and a strip plot
plt.figure(figsize=(12,10))
sns.violinplot(x='day',y='total_bill',data=tips_df)
sns.swarmplot(x='day',y='total_bill',data=tips_df,color='white',hue='sex')
plt.show()
```



2.7 Palettes

- by setting the palettes you can change the default coloring of the any plot
- to see more detail about the color palette click here

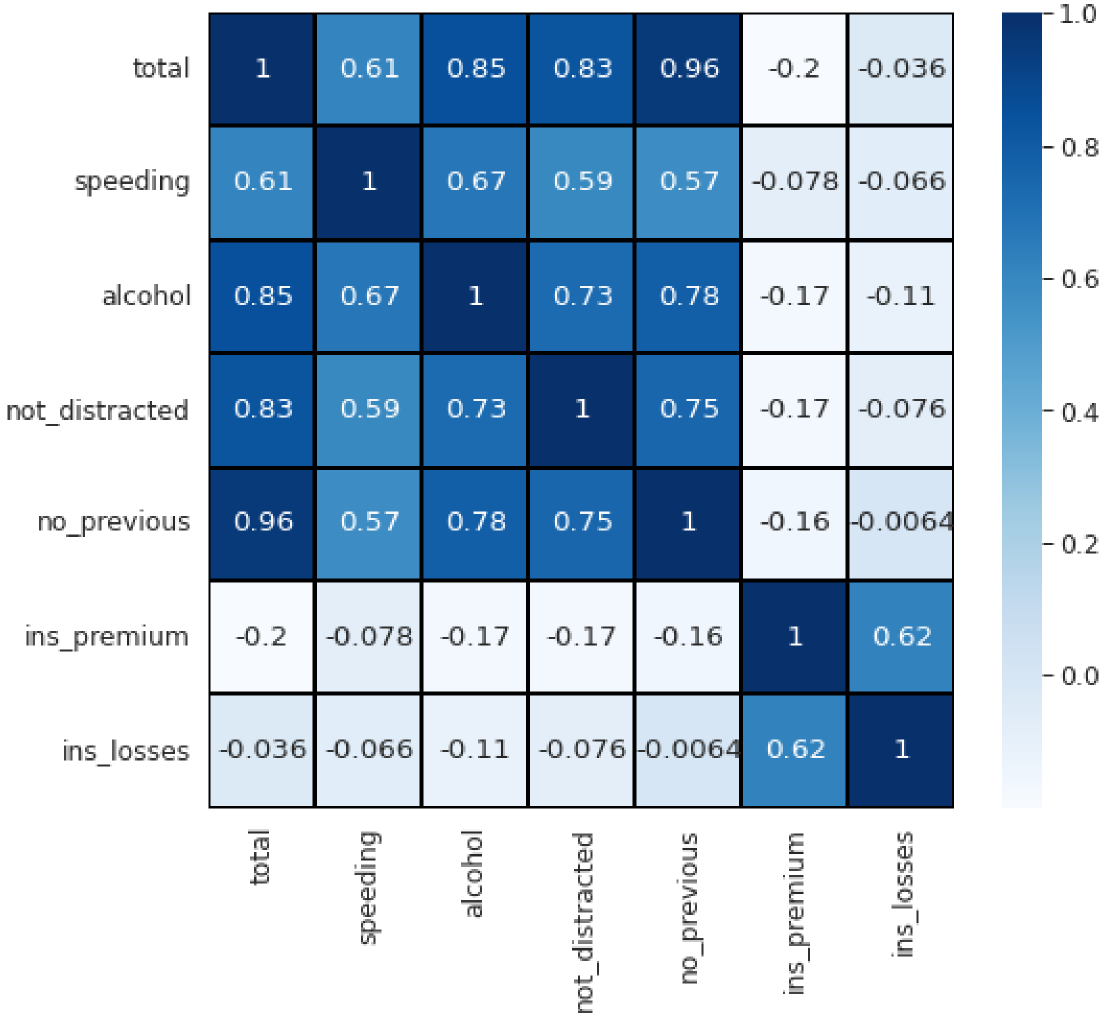
[16]:
```
# by setting the palettes you can change the default coloring of the any plot
plt.figure(figsize=(12,9))
sns.set_style('white')
sns.set_context('talk')
sns.stripplot(x='day',y='total_bill',data=tips_df,hue='sex',palette='afmhot')
# plt.legend(loc=1)
plt.show()
```



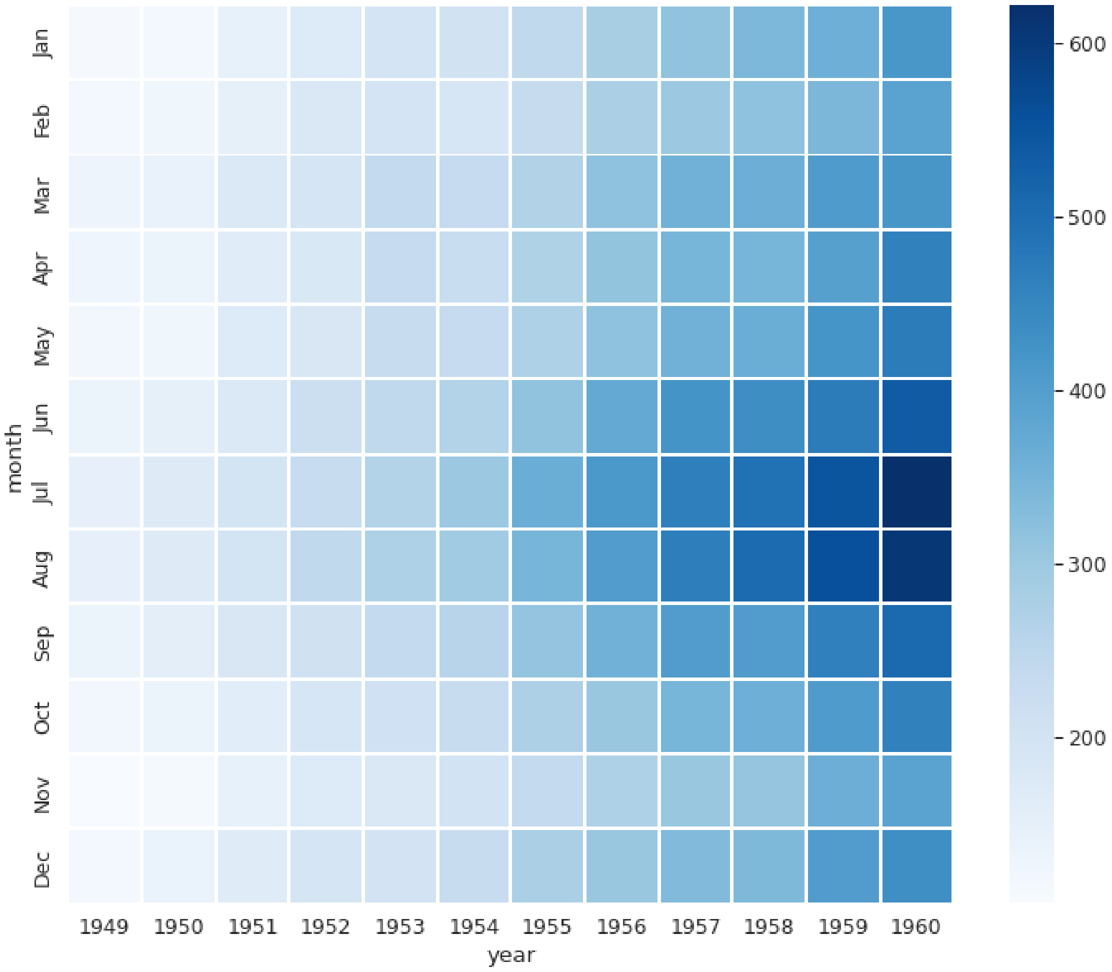
3 Matrix Plot

3.1 Heatmaps

```
[17]: #
plt.figure(figsize=(8,7))
sns.set_context('paper',font_scale=1.4)

crash_mx = crash_df.corr()

sns.heatmap(crash_mx,annot=True,cmap='Blues',linewidth=1,linecolor='black')
plt.show()
```



[18]:
```
plt.figure(figsize=(12,10))
flights_df = sns.load_dataset('flights')
flights = flights_df.
  ↪pivot_table(index='month',columns='year',values='passengers')
sns.heatmap(flights,cmap='Blues',linecolor='white',linewidth=1)
plt.show()
```

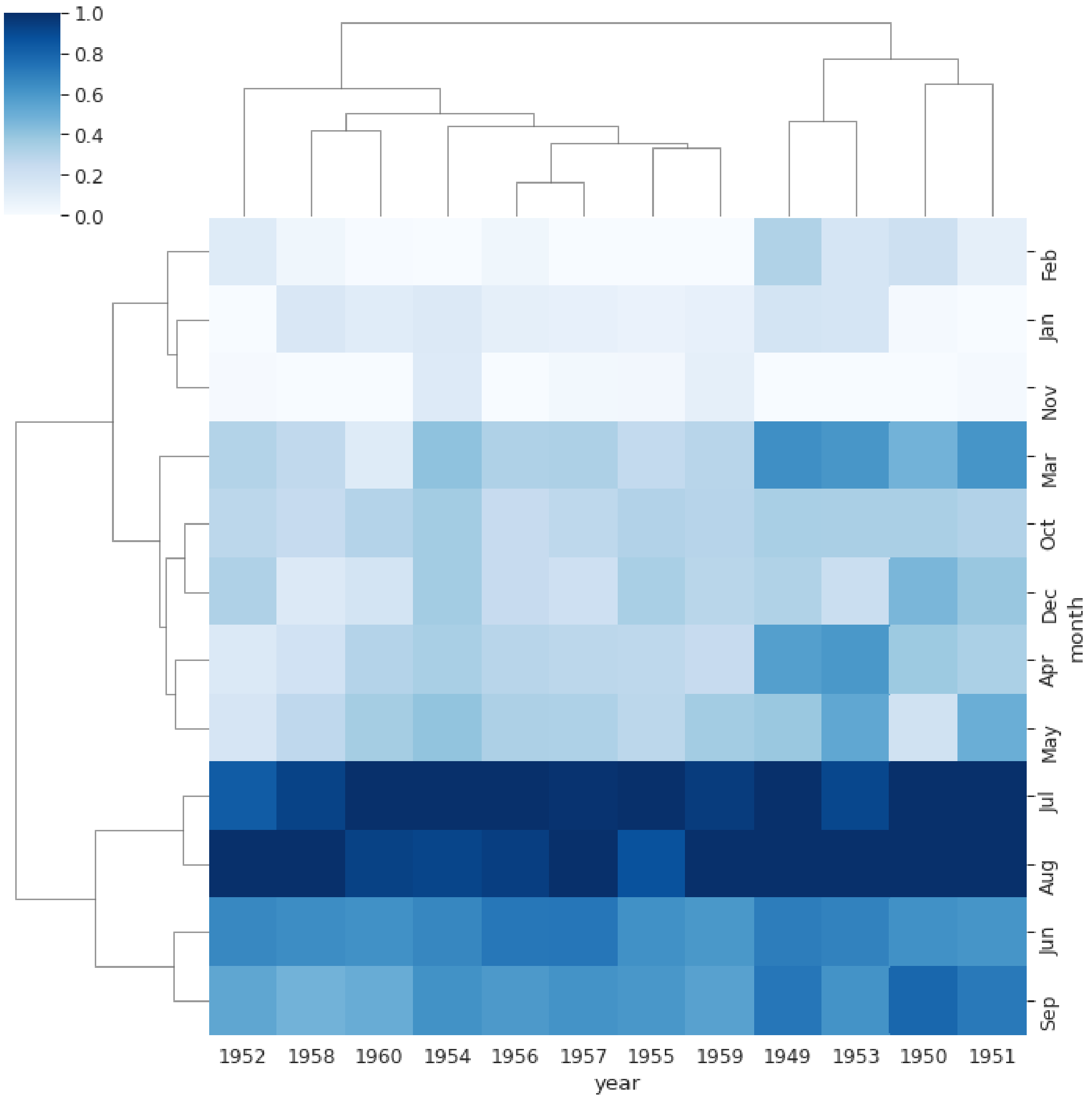


3.2 Cluster Map

```
[19]: # it gives the cluster data by finding the closest distance between various␣
        ↪points
      iris = sns.load_dataset('iris')
      # species = iris.pop('species')
      # sns.clustermap(iris)
```

```
[20]: sns.clustermap(flights,cmap='Blues',standard_scale=1)
```

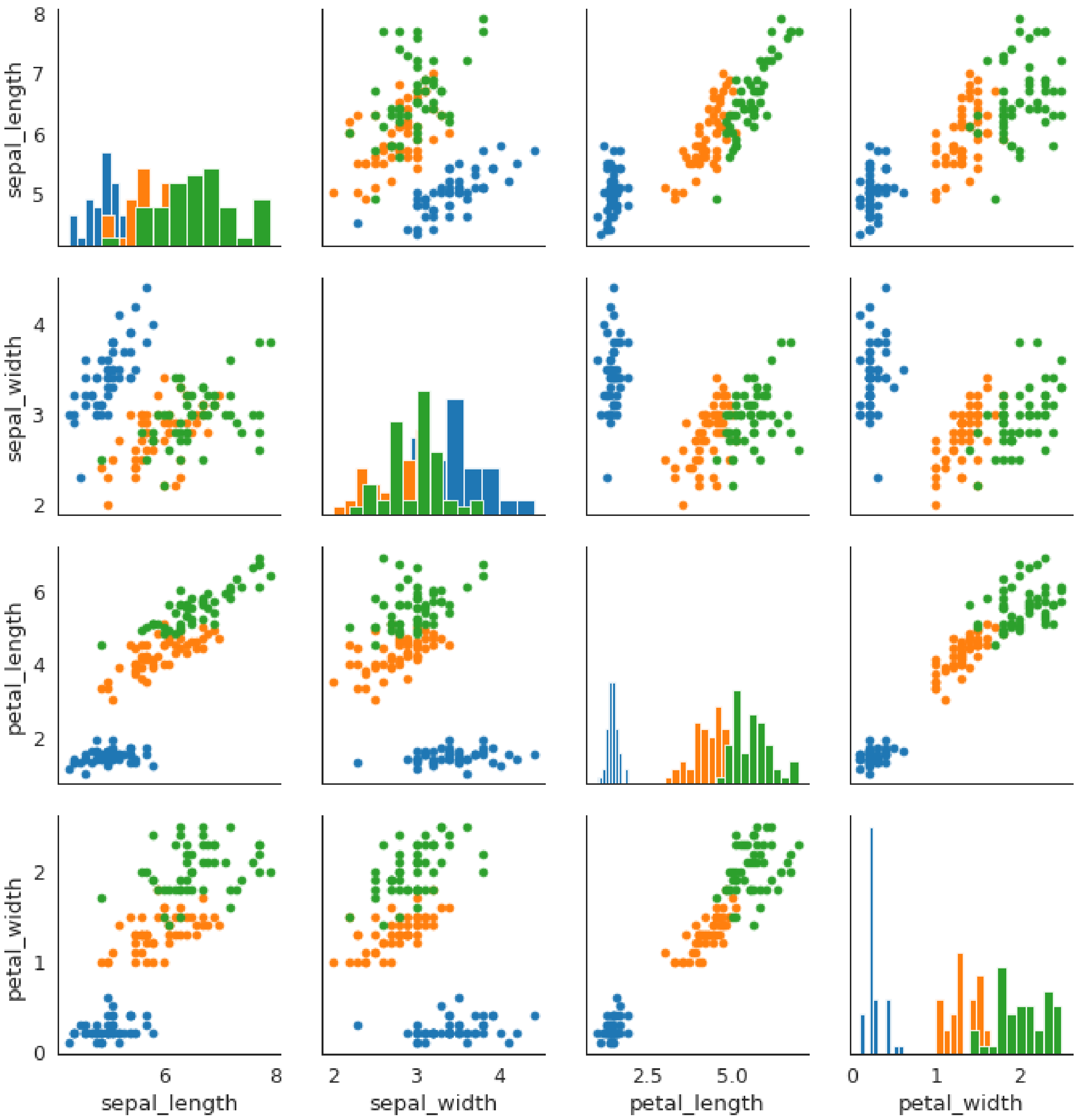
```
[20]: <seaborn.matrix.ClusterGrid at 0x7ff4adc3b670>
```



3.3 Pair Grid

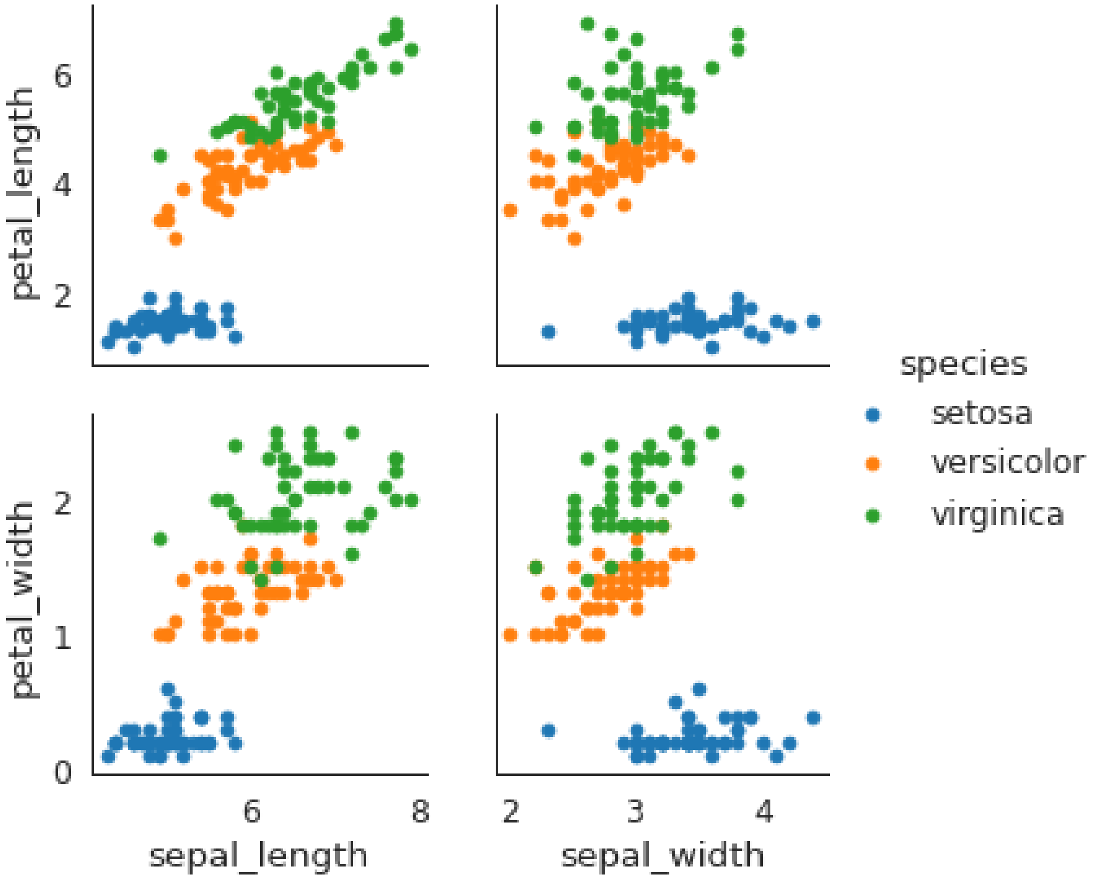
[21]:
```
# to plot or change the default plots in the different position use the␣
↪'shift+tab' [iris_g.map_upper,.map_lower,.map_offdiag,...]
iris_g = sns.PairGrid(iris,hue='species')
# iris_g.map(plt.scatter)
iris_g.map_diag(plt.hist)
iris_g.map_offdiag(plt.scatter)
```

[21]:
```
<seaborn.axisgrid.PairGrid at 0x7ff4b1c02c70>
```



```
[22]: # for our custom grid plot we can define the x and y variables with dataset
      iris_g = sns.
       ↪PairGrid(iris,hue='species',x_vars=['sepal_length','sepal_width'],y_vars=['petal_length','p
      iris_g.map(plt.scatter)
      iris_g.add_legend()
```

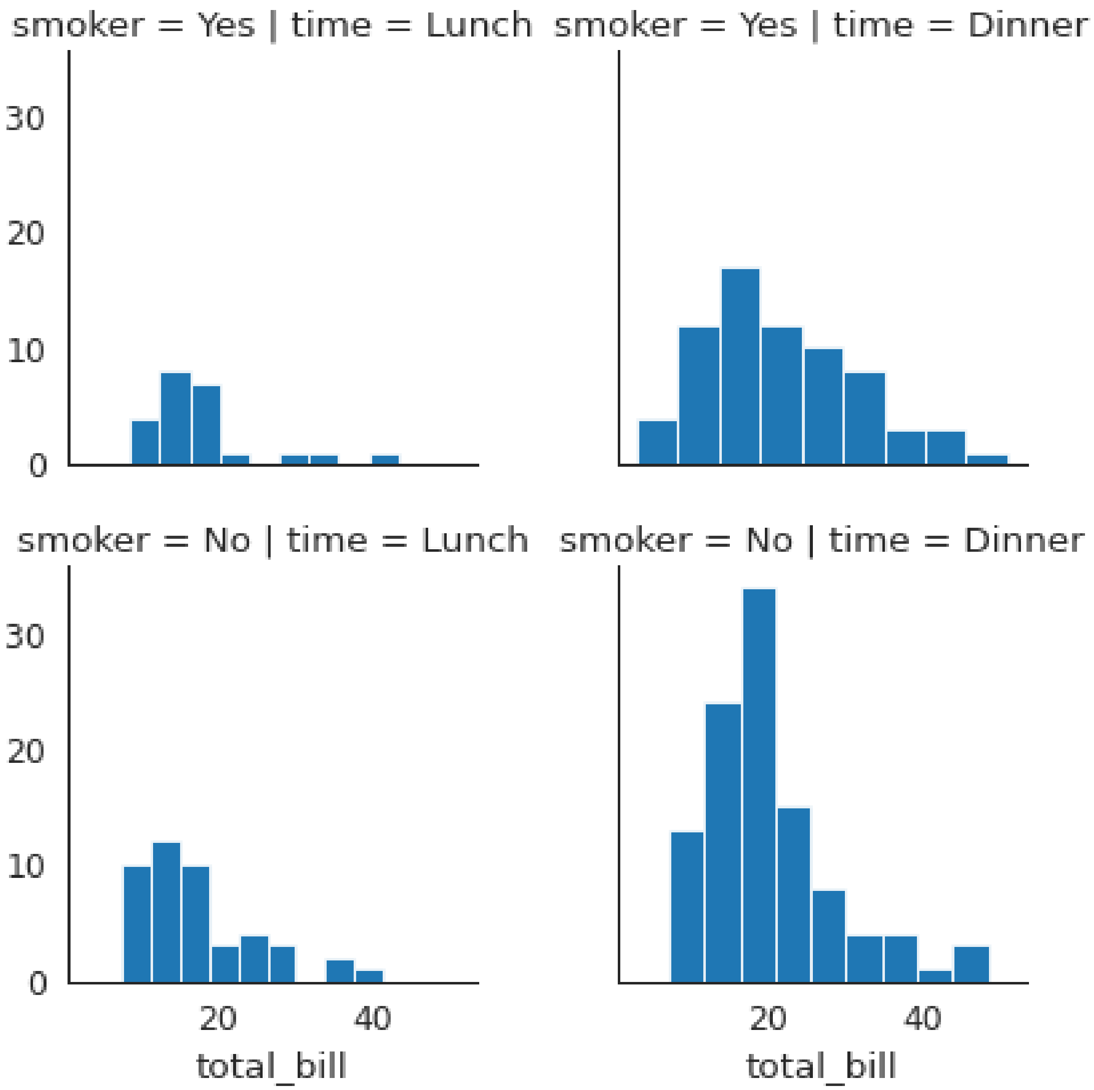
```
[22]: <seaborn.axisgrid.PairGrid at 0x7ff4addad880>
```



3.4 Facet Grid

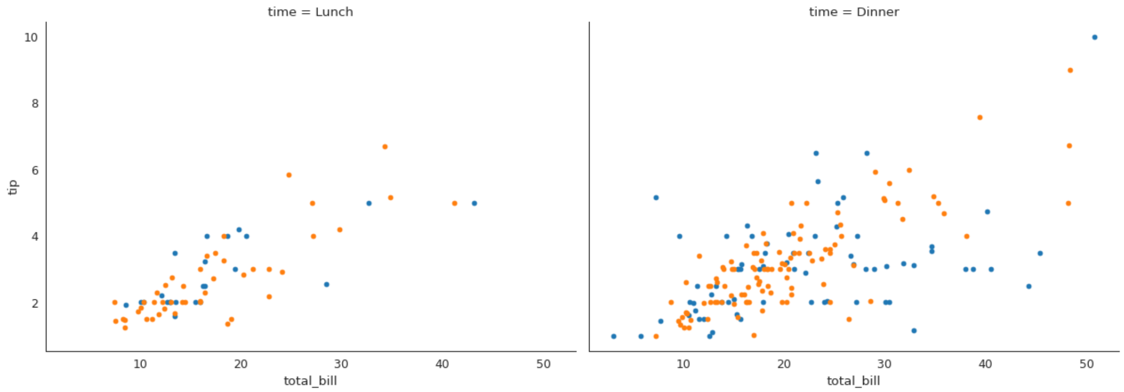
```
[23]: import seaborn as sns
      # this can plot multiple plot in a grid and in this we can define the columns␣
       ↪and rows
      tips_fg = sns.FacetGrid(tips_df,col='time',row='smoker') # to draw the plot fig␣
       ↪with only axis no data in it
      # sns.set_context('paper',font_scale=1.2)
      tips_fg.map(plt.hist, 'total_bill',bins=9) # ['time','smoder',]
      # tips_fg.map(plt.scatter, 'total_bill','tip')
```

```
[23]: <seaborn.axisgrid.FacetGrid at 0x7ff4ad7d3b20>
```



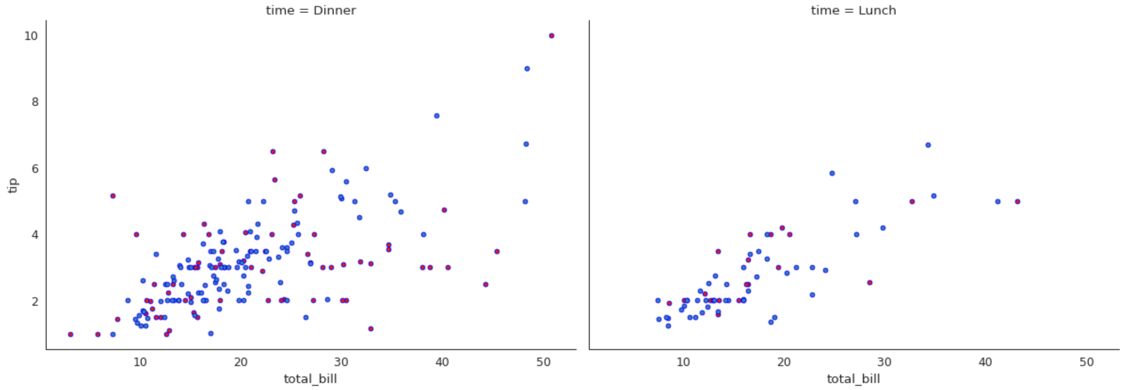
```
[24]: tips_fg1 = sns.FacetGrid(tips_df,col='time',hue='smoker',height=6,aspect=1.4)
      tips_fg1.map(plt.scatter,'total_bill','tip')
```

```
[24]: <seaborn.axisgrid.FacetGrid at 0x7ff4adf1a7c0>
```



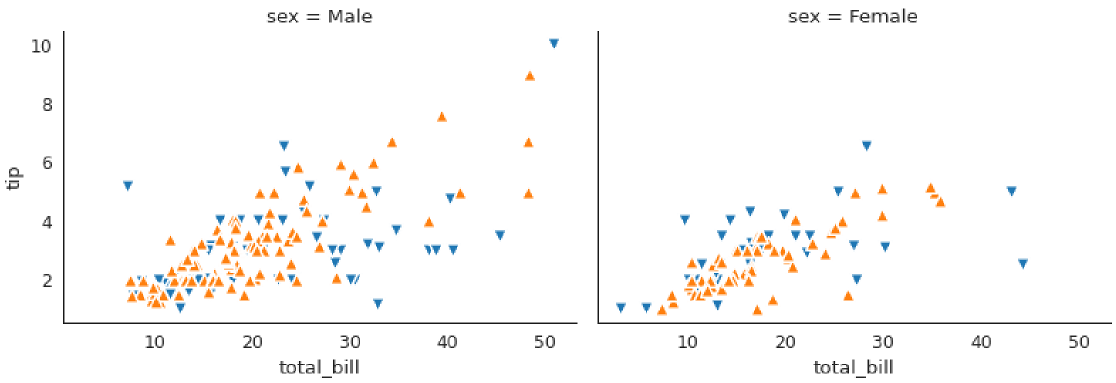
```
[25]: tips_fg2 = sns.FacetGrid(tips_df,col='time',hue='smoker',height=6,aspect=1.
        ↪4,col_order=['Dinner','Lunch'],palette='Set1')
      tips_fg2.map(plt.scatter,'total_bill','tip',edgecolor='b')
```

```
[25]: <seaborn.axisgrid.FacetGrid at 0x7ff4ad5e6e80>
```



```
[26]: kws = dict(s=90,linewidth=1.5,edgecolor='w')
      tips_fg2 = sns.FacetGrid(tips_df,col='sex',hue='smoker',height=4,aspect=1.
        ↪4,hue_order=['Yes','No'],hue_kws=dict(marker=['v','^']))
      tips_fg2.map(plt.scatter,'total_bill','tip',**kws)
```

```
[26]: <seaborn.axisgrid.FacetGrid at 0x7ff4ad553df0>
```



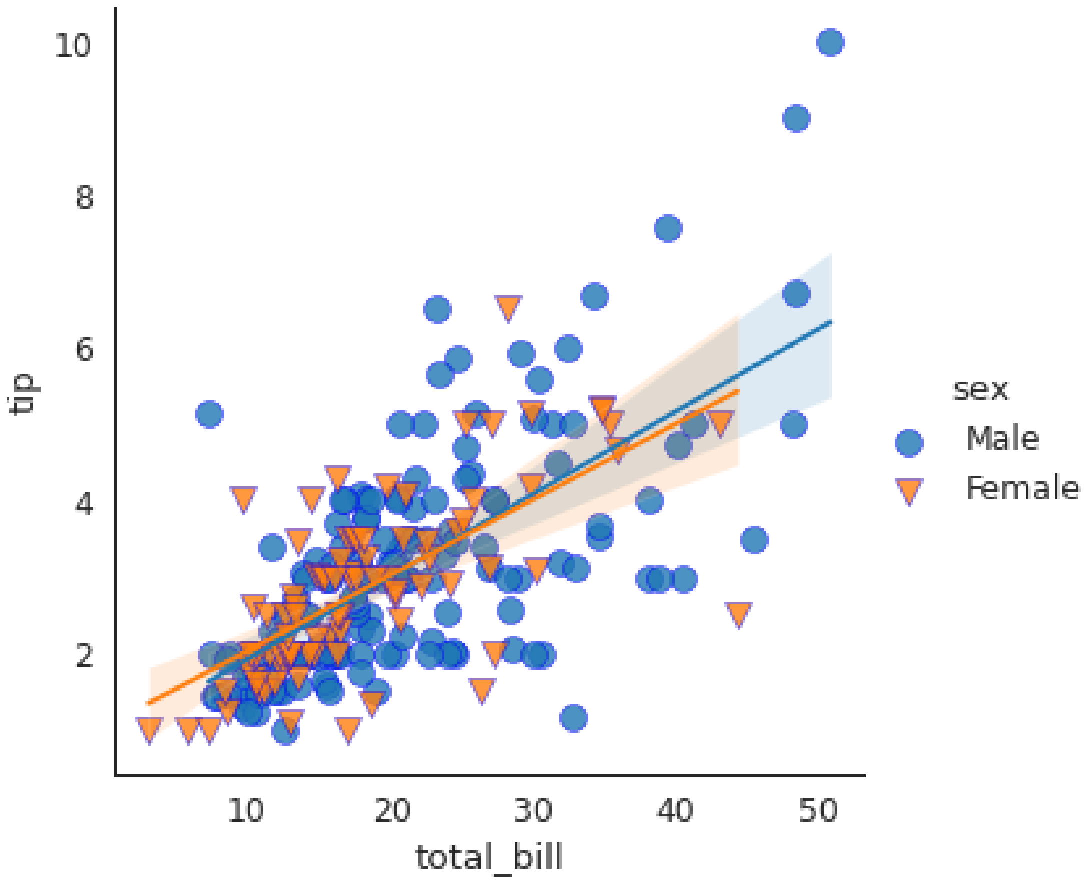
3.5 Regression Plot

[27]:

```
plt.figure(figsize=(8,6))
sns.set_context('paper',font_scale=1.4)
sns.
  ↪lmplot(x='total_bill',y='tip',hue='sex',data=tips_df,markers=['o','v'],scatter_kws={'s':
  ↪100,'linewidth':0.5,'edgecolor':'b'})
```

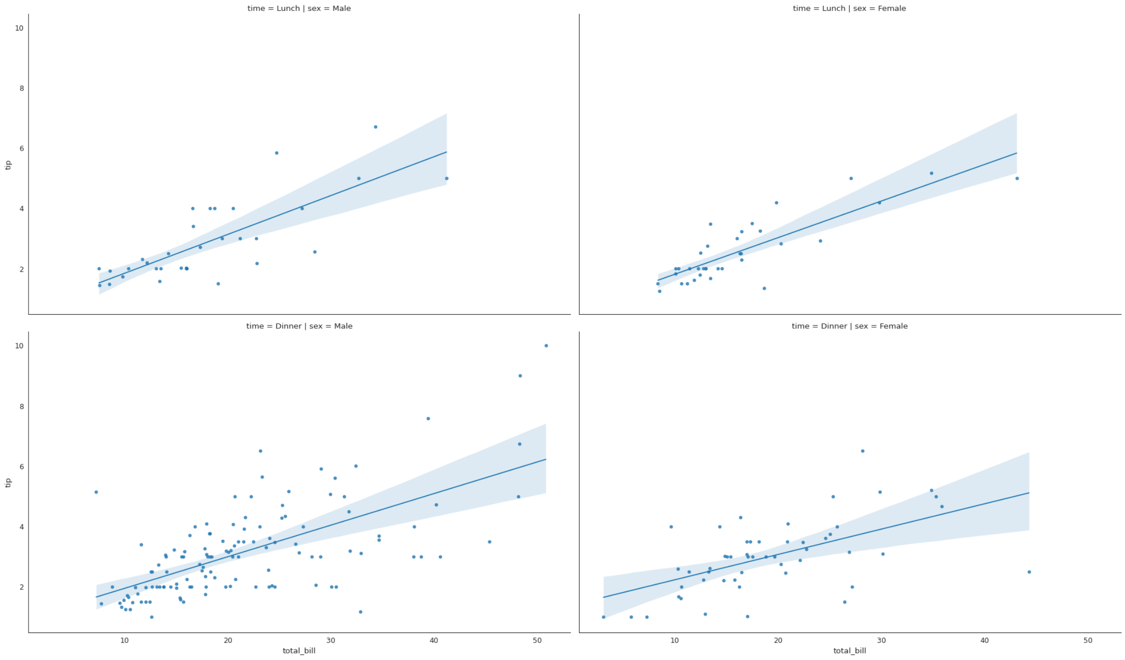
[27]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff4ad5ba460>

<Figure size 576x432 with 0 Axes>
```



[28]:
```
sns.
↪lmplot(x='total_bill',y='tip',col='sex',row='time',data=tips_df,height=8,aspect=1.
↪7)
```

[28]:
```
<seaborn.axisgrid.FacetGrid at 0x7ff4ad4743d0>
```


time = Lunch | sex = Male
time = Lunch | sex = Female
time = Dinner | sex = Male
time = Dinner | sex = Female
tip
tip
10
8
6
4
2
10
8
6
4
2
10
20
30
40
50
10
20
30
40
50
total_bill
total_bill