

```
In [1]: # VIVEK-CHAUHAN-ADVANCED-DATA-ANALYTICS-SEABORN-KDEPLOT-HEATMAP
```

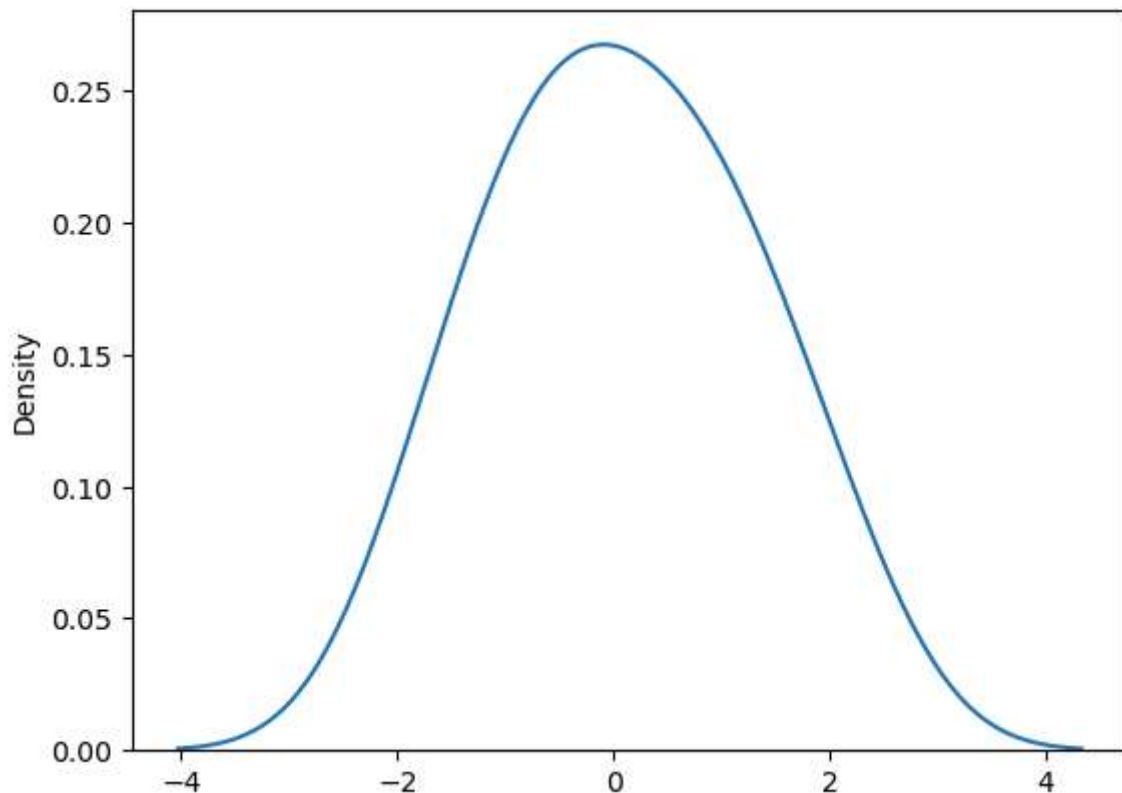
```
In [3]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [13]: x = np.random.randn(10) # generate the random numbers upto 10 times
y = np.random.randn(10)
x
```

```
Out[13]: array([ 2.06646348,  0.25840548,  1.32496136,  1.03331445, -0.67867695,
 0.54581526, -0.1527473 , -1.22180492, -0.59873952, -1.76789785])
```

```
In [15]: # Let's create the kde plot
sns.kdeplot(x)
```

```
Out[15]: <Axes: ylabel='Density'>
```



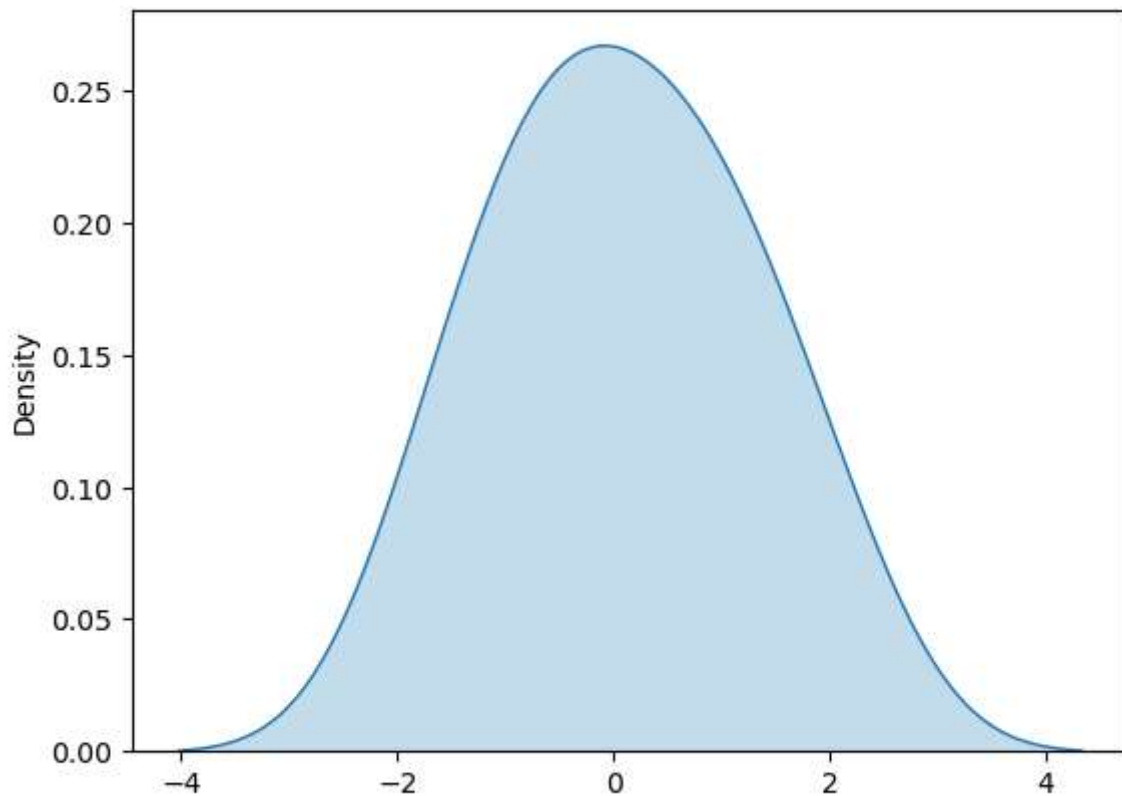
```
In [19]: # if you want to fill the color in shade then use shade=True
sns.kdeplot(x,shade=True)
```

C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel\_34156\647766624.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.  
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(x,shade=True)
```

Out[19]: <Axes: ylabel='Density'>



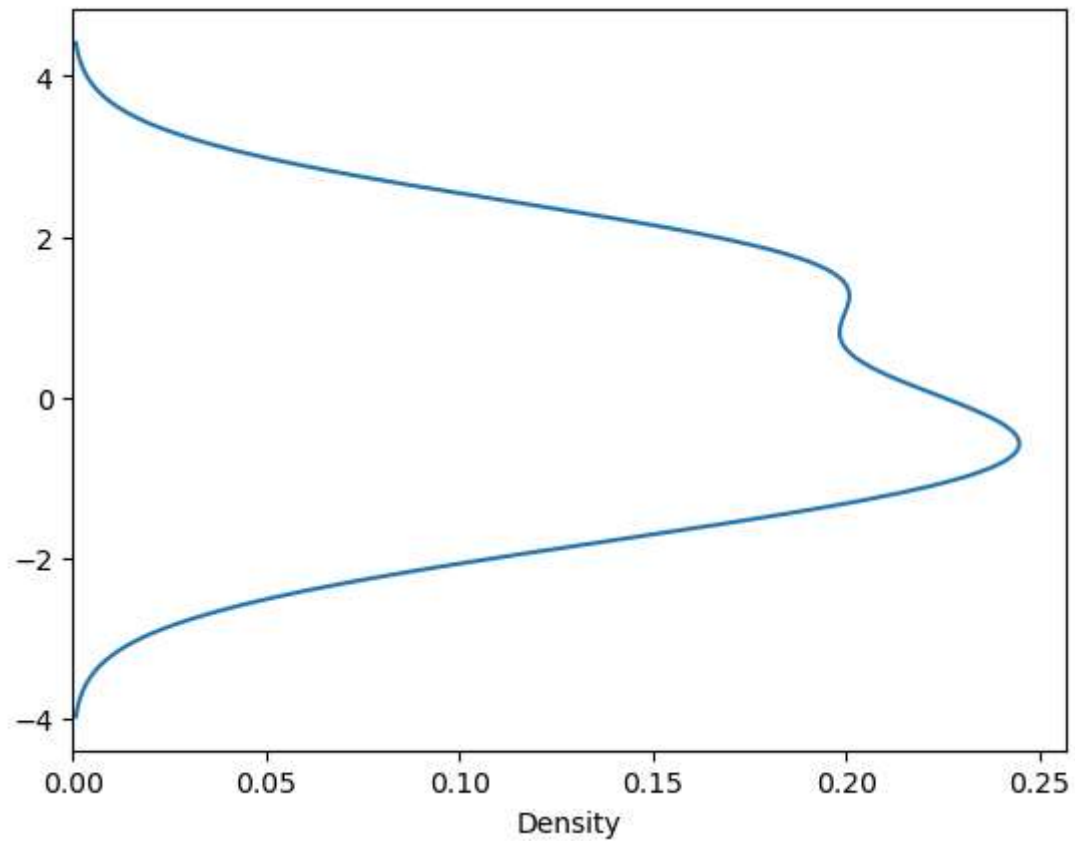
```
In [21]: # if you want vertical plot then vertical=True
sns.kdeplot(y,vertical=True)
```

C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel\_34156\2508444410.py:2: UserWarning:

The `vertical` parameter is deprecated; assigning data to `y`.  
This will become an error in seaborn v0.14.0; please update your code.

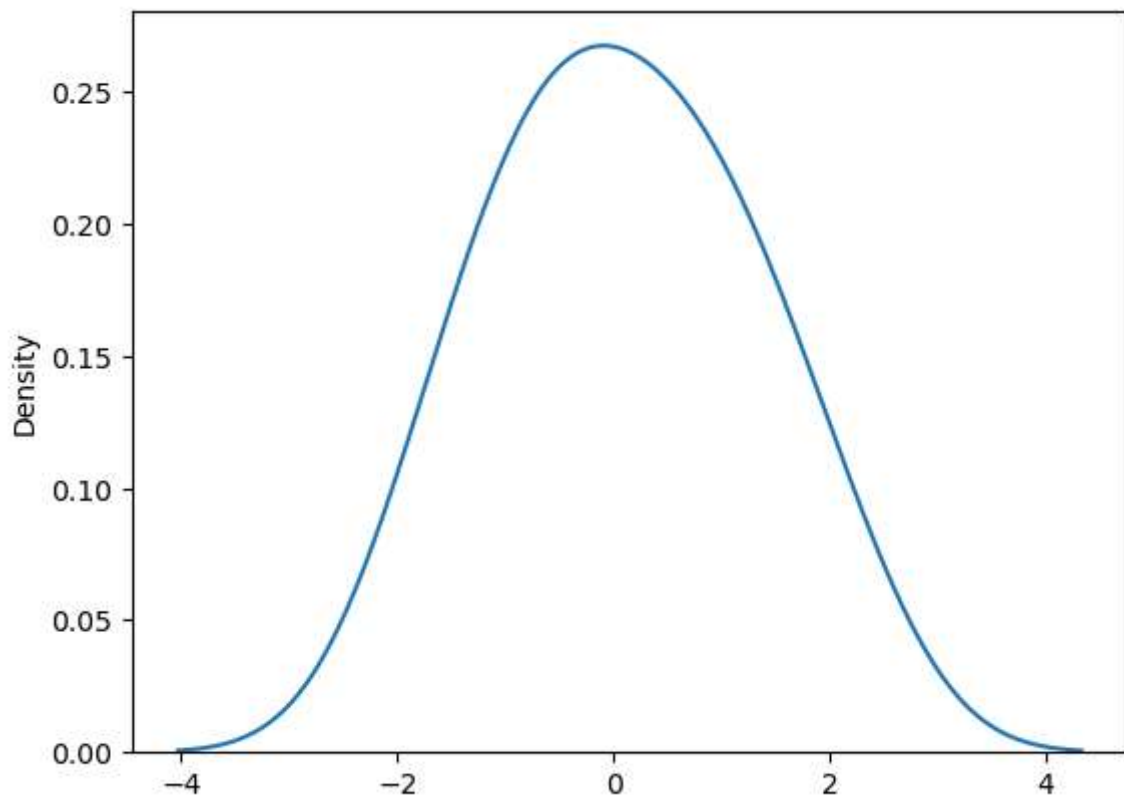
```
sns.kdeplot(y,vertical=True)
```

Out[21]: <Axes: xlabel='Density'>



```
In [23]: sns.kdeplot(x)
```

```
Out[23]: <Axes: ylabel='Density'>
```



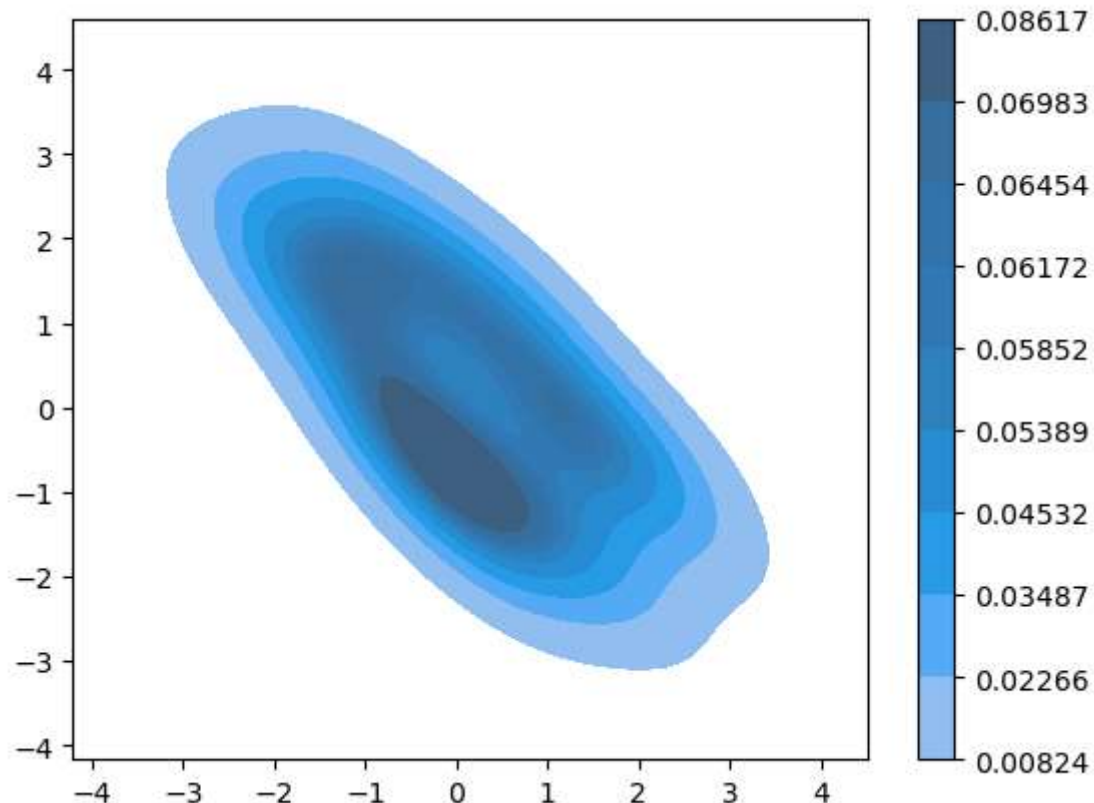
```
In [33]: # bivariate kde plot for two variables
# cbar means coloumn bar for better clarification for relations
sns.kdeplot(x=x,y=y,shade=True,cbar=True)
```

C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel\_34156\1920451080.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.  
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(x=x,y=y,shade=True,cbar=True)
```

Out[33]: <Axes: >



```
In [29]: df = sns.load_dataset("iris")
df
```

Out[29]:

	sepal_length	sepal_width	petal_length	petal_width	species
<b>0</b>	5.1	3.5	1.4	0.2	setosa
<b>1</b>	4.9	3.0	1.4	0.2	setosa
<b>2</b>	4.7	3.2	1.3	0.2	setosa
<b>3</b>	4.6	3.1	1.5	0.2	setosa
<b>4</b>	5.0	3.6	1.4	0.2	setosa
<b>...</b>	...	...	...	...	...
<b>145</b>	6.7	3.0	5.2	2.3	virginica
<b>146</b>	6.3	2.5	5.0	1.9	virginica
<b>147</b>	6.5	3.0	5.2	2.0	virginica
<b>148</b>	6.2	3.4	5.4	2.3	virginica
<b>149</b>	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

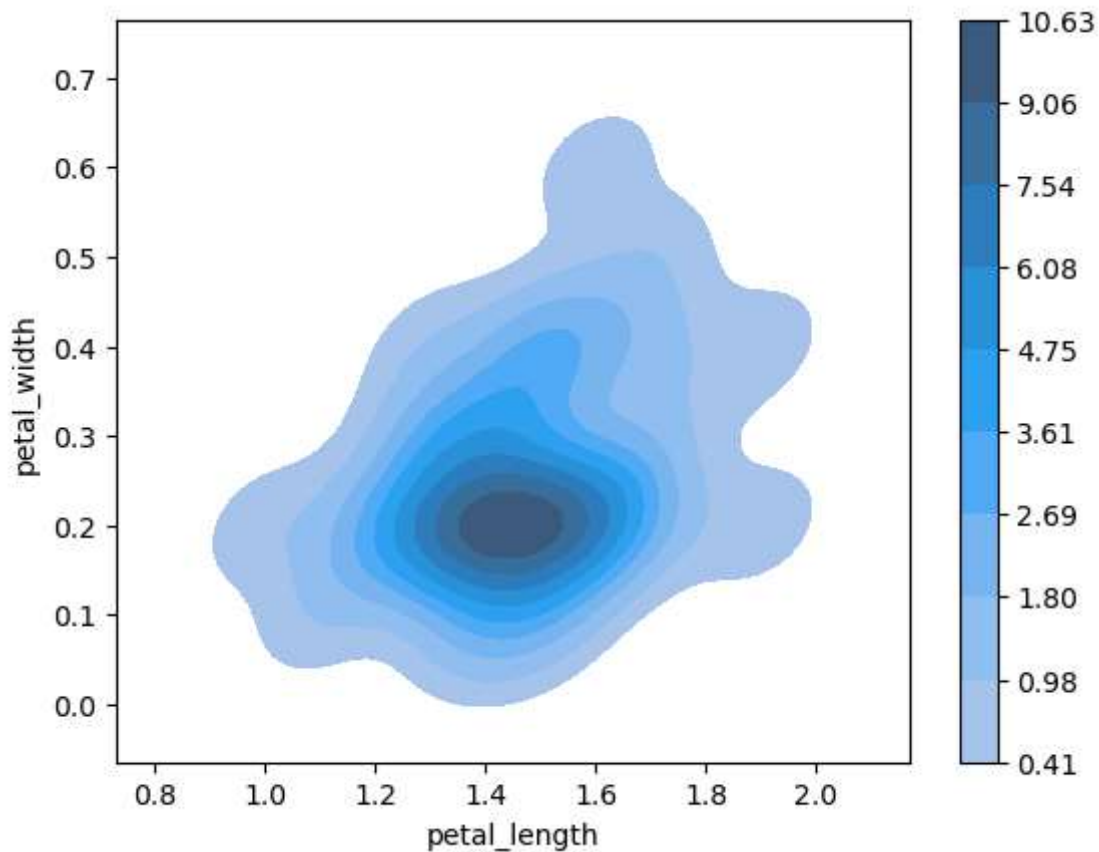
```
In [39]: setosa = df.loc[df.species=="setosa"]
virginica = df.loc[df.species=="virginica"]
sns.kdeplot(x=setosa.petal_length,y=setosa.petal_width,cbar=True,shade=True)
```

C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel\_34156\3050502438.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.  
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(x=setosa.petal_length,y=setosa.petal_width,cbar=True,shade=True)
```

```
Out[39]: <Axes: xlabel='petal_length', ylabel='petal_width'>
```



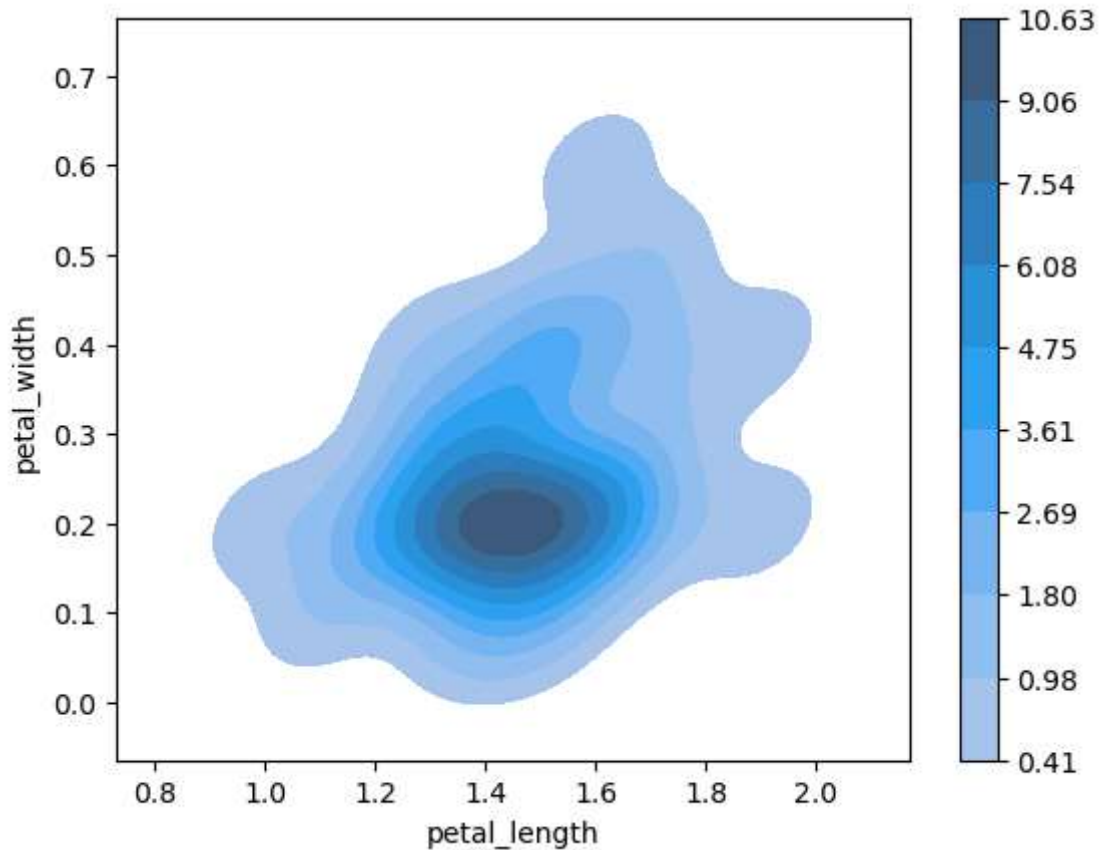
```
In [51]: # if you want to change the color then use cmap = name
setosa = df.loc[df.species=="setosa"]
virginica = df.loc[df.species=="virginica"]
sns.kdeplot(x=setosa.petal_length,y=setosa.petal_width,cbar=True,shade=True)
```

C:\Users\fv8.DESKTOP-N5HA3AQ\AppData\Local\Temp\ipykernel\_34156\1825217355.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.  
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(x=setosa.petal_length,y=setosa.petal_width,cbar=True,shade=True)
```

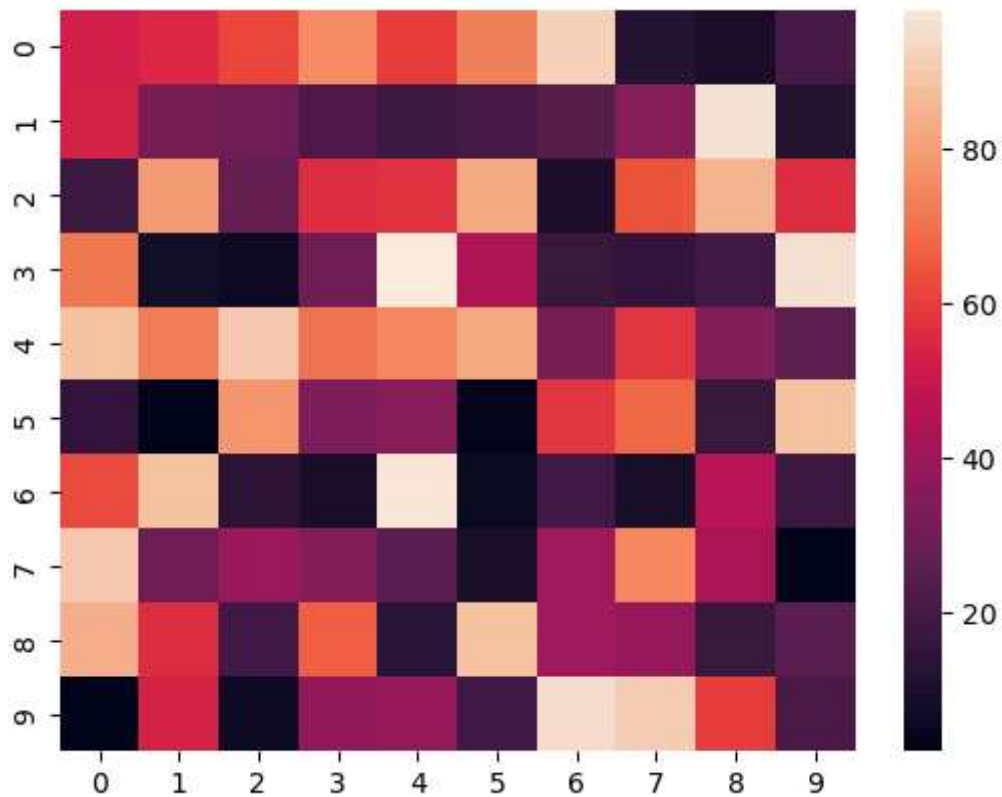
```
Out[51]: <Axes: xlabel='petal_length', ylabel='petal_width'>
```



```
In [67]: # Let's create the heatmap first of all create/load the data
data = np.random.randint(low=1,high=100,size=(10,10)) # size means x,y matrix size
data
```

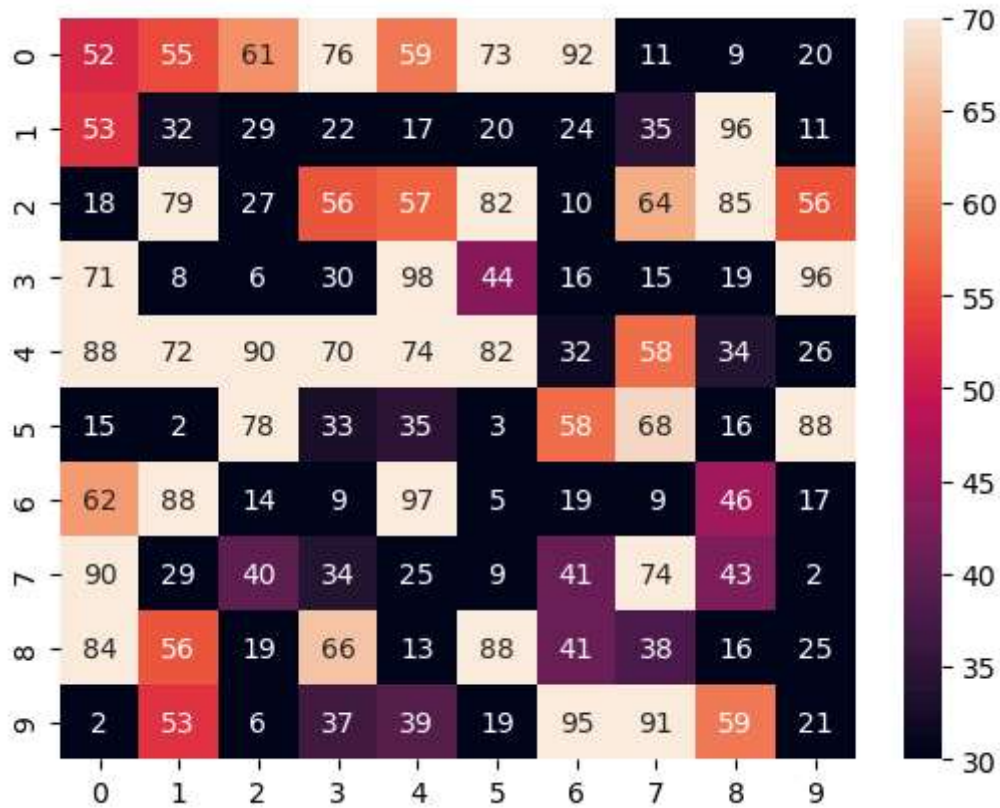
```
Out[67]: array([[52, 55, 61, 76, 59, 73, 92, 11,  9, 20],
 [53, 32, 29, 22, 17, 20, 24, 35, 96, 11],
 [18, 79, 27, 56, 57, 82, 10, 64, 85, 56],
 [71,  8,  6, 30, 98, 44, 16, 15, 19, 96],
 [88, 72, 90, 70, 74, 82, 32, 58, 34, 26],
 [15,  2, 78, 33, 35,  3, 58, 68, 16, 88],
 [62, 88, 14,  9, 97,  5, 19,  9, 46, 17],
 [90, 29, 40, 34, 25,  9, 41, 74, 43,  2],
 [84, 56, 19, 66, 13, 88, 41, 38, 16, 25],
 [ 2, 53,  6, 37, 39, 19, 95, 91, 59, 21]])
```

```
In [69]: # Let's create the heatmap
hm = sns.heatmap(data=data)
```

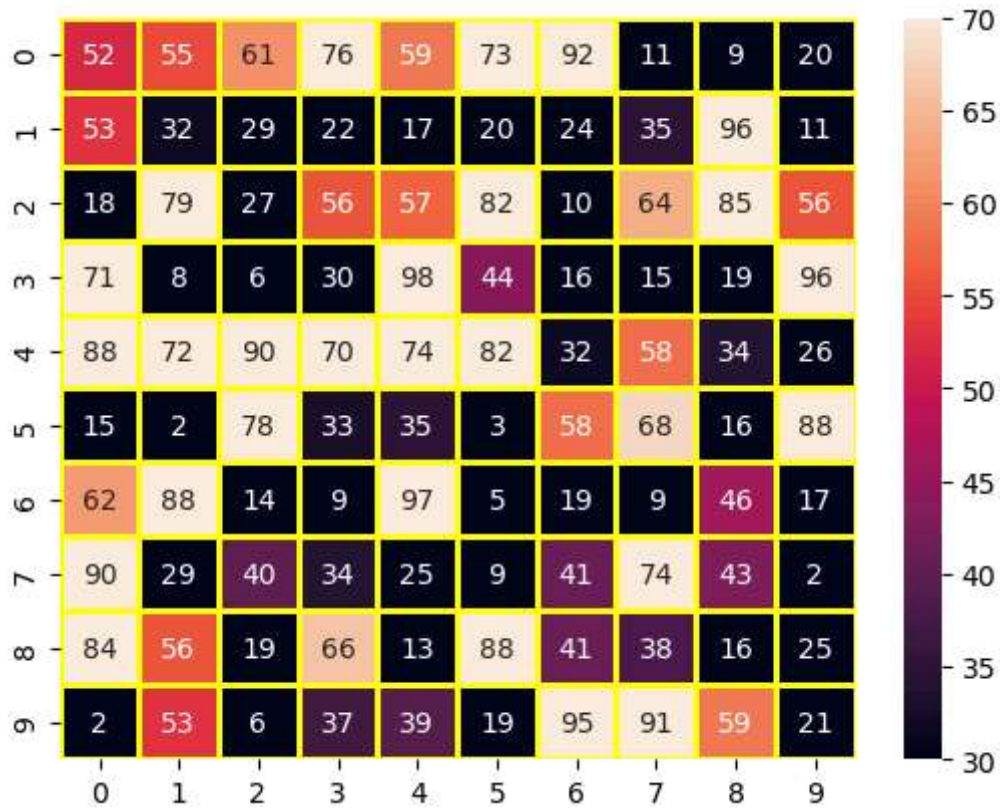


```
In [71]: # we can give vertical min & max
vmin = 30
vmax = 70
# plotting the heatmap
# If annot set to True, the value in each cell is annotated on the heatmap
hm = sns.heatmap(data=data, vmin=vmin, vmax=vmax, annot=True)
```

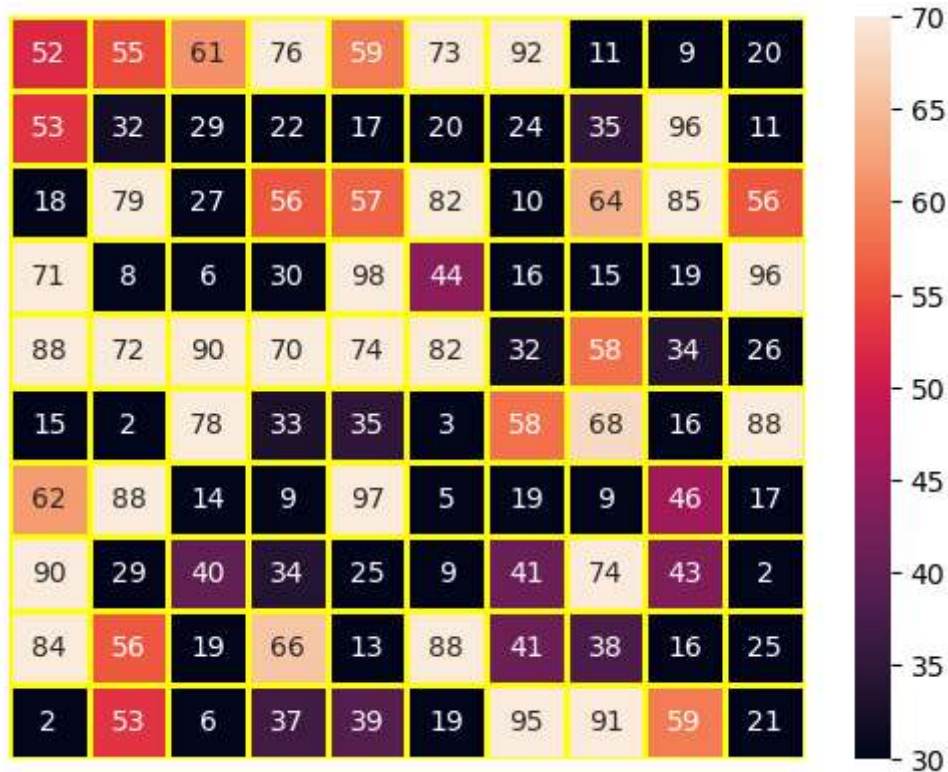




```
In [131... # we can give the linecolor & linewidth of each cell
# we can give vertical min & max
vmin = 30
vmax = 70
# plotting the heatmap
# If annot set to True, the value in each cell is annotated on the heatmap
hm = sns.heatmap(data=data, vmin=vmin, vmax=vmax, annot=True, linecolor="yellow", linewi
```



```
In [93]: # we can also remove the x,y ticklabels
# we can give the linecolor & linewidth of each cell
# we can give vertical min & max
vmin = 30
vmax = 70
# plotting the heatmap
# If annot set to True, the value in each cell is annotated on the heatmap
hm = sns.heatmap(data=data, vmin=vmin, vmax=vmax, annot=True, linecolor="yellow", linewidth=1,
                  xticklabels=False, yticklabels=False)
```



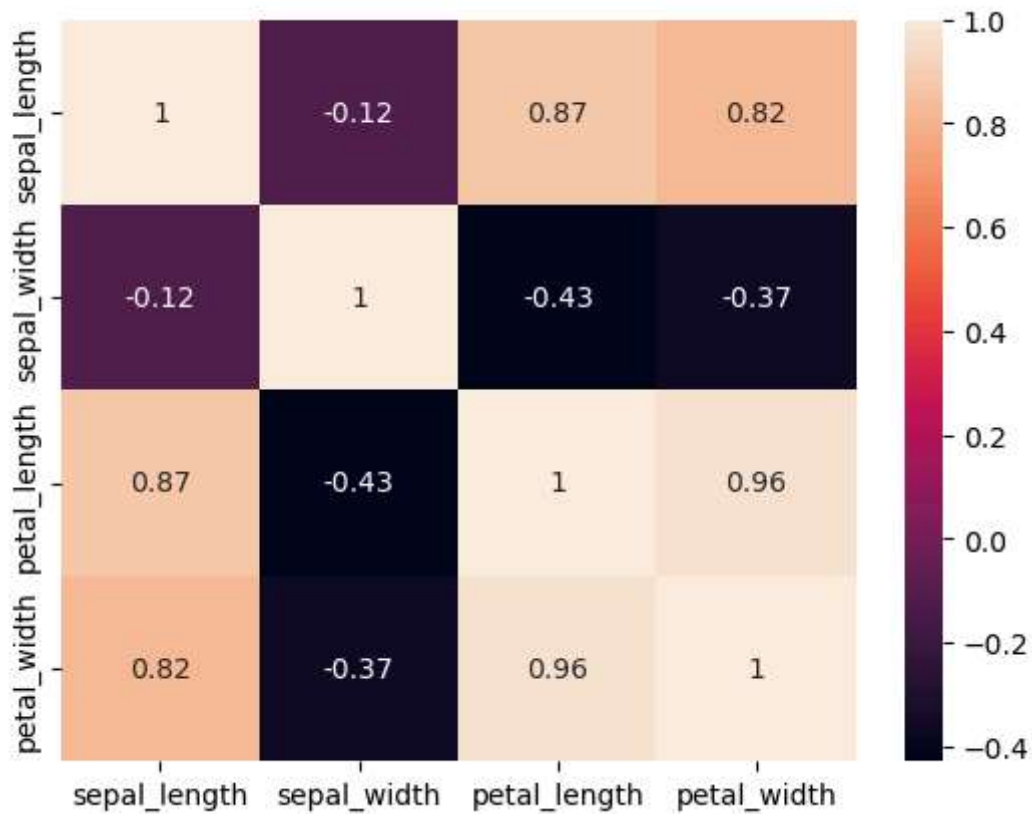
```
In [31]: # Let's create the correlation heatmap so first of all we need to remove the string
crrdata = df.corr(numeric_only=True) # correlation between column & rows
crrdata
```

```
Out[31]:
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

```
In [33]: # we create the heatmap for correlation data
sns.heatmap(crrdata,annot=True)
```

```
Out[33]: <Axes: >
```



```
In [9]: # we can give our own color style to the heatmap by declaring cmap
colormap = sns.color_palette("Greens")
colormap
```



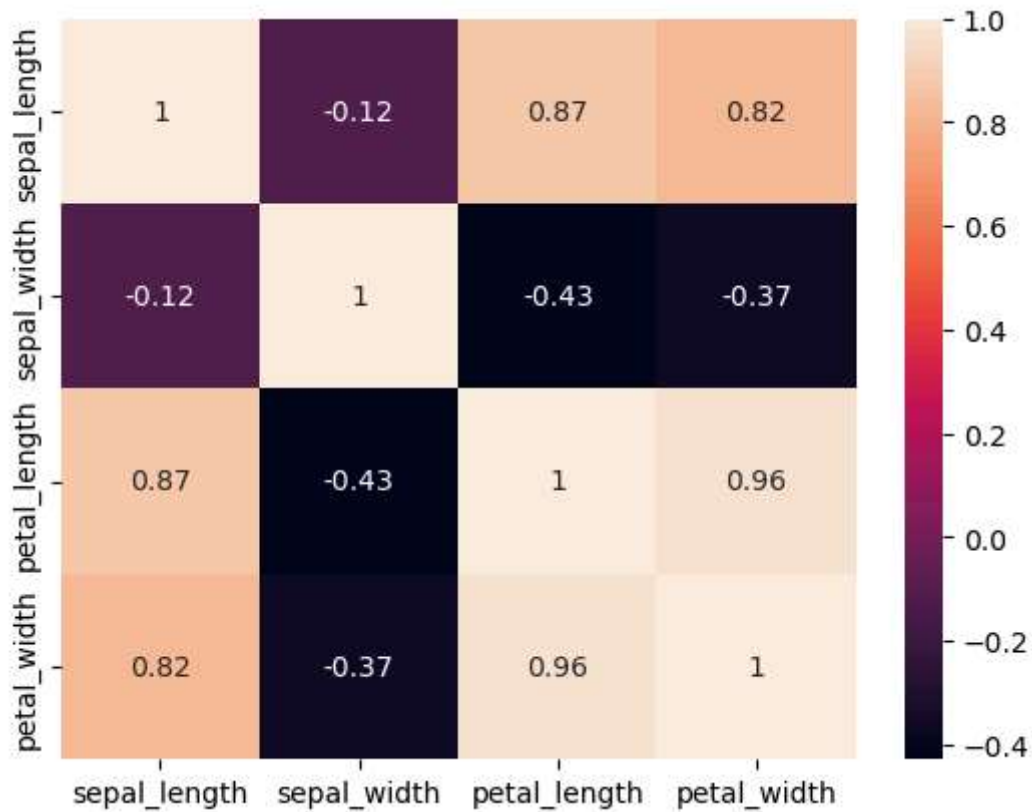
```
In [125... # Let's create the heatmap using color_palette style
# we create the heatmap for correlation data
sns.heatmap(crrdata, cmap=colormap, annot=True)
```

Out[125... <Axes: >



```
In [35]: # we can increase the annot number size by using annot_kws {"size":number}
sns.heatmap(crrdata,annot=True,annot_kws={"size":10})
```

Out[35]: <Axes: >



In [ ]: