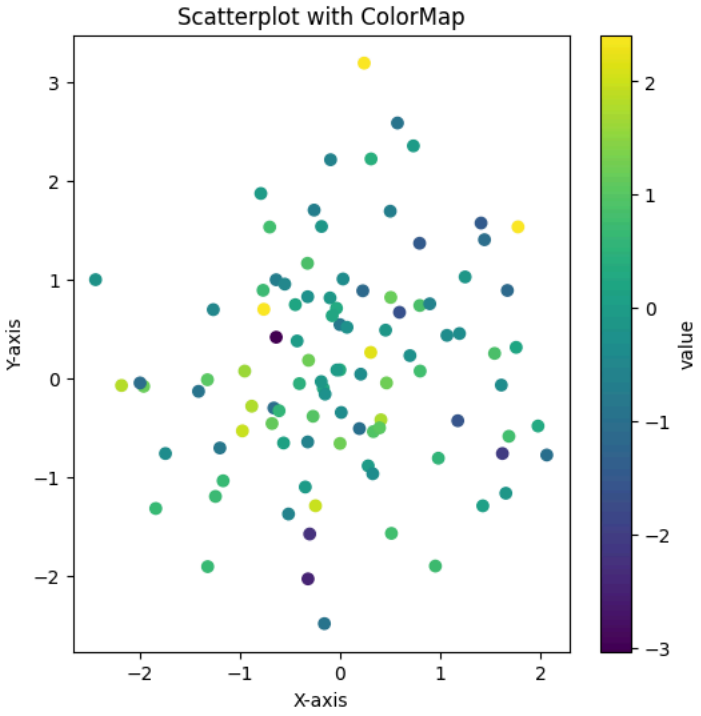
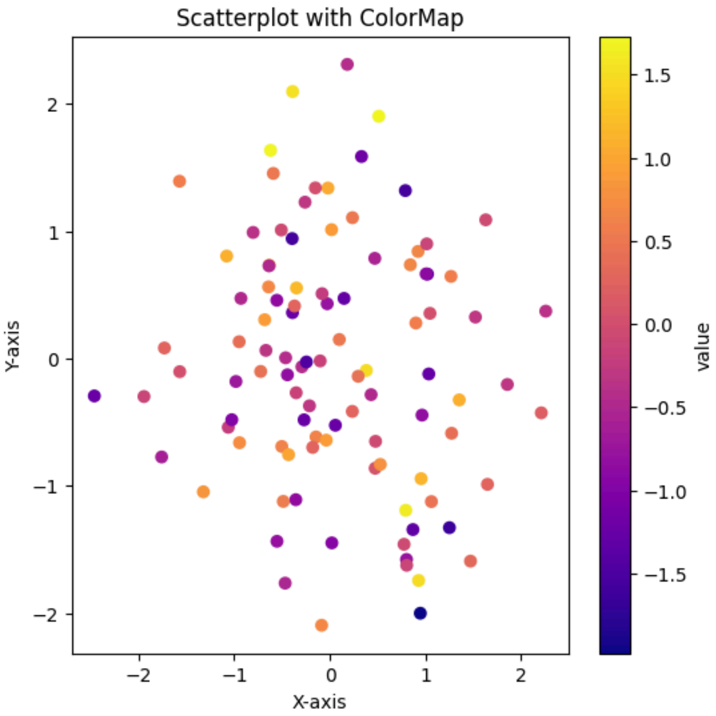
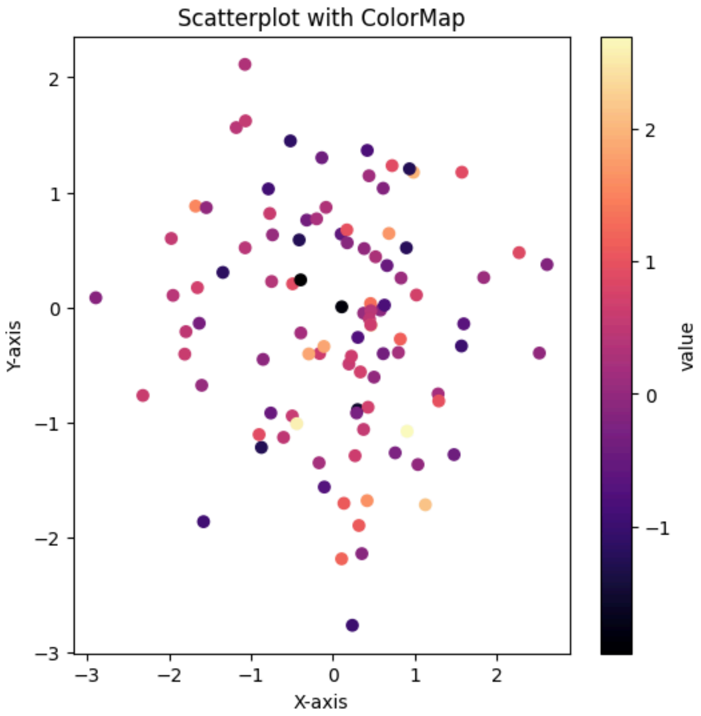
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#Sample dataframe with multiple columns
data=pd.DataFrame({
    "x":np.random.randn(100),
    "y":np.random.randn(100),
    "value":np.random.randn(100)
})
#Define the colormap and alpha values
cmap="viridis"
alpha=1
#create the scatterplot
plt.figure(figsize=(6,6))
plt.scatter(data["x"],data["y"],c=data["value"],cmap=cmap,alpha=alpha)
#customise the plot
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Scatterplot with ColorMap")
plt.colorbar(label="value")
#show the plot
plt.show()
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#Sample dataframe with multiple columns
data=pd.DataFrame({
    "x":np.random.randn(100),
    "y":np.random.randn(100),
    "value":np.random.randn(100)
})
#Define the colormap and alpha values
cmap="plasma"
alpha=1
#create the scatterplot
plt.figure(figsize=(6,6))
plt.scatter(data["x"],data["y"],c=data["value"],cmap=cmap,alpha=alpha)
#customise the plot
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Scatterplot with ColorMap")
plt.colorbar(label="value")
#show the plot
plt.show()
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#Sample dataframe with multiple columns
data=pd.DataFrame({
    "x":np.random.randn(100),
    "y":np.random.randn(100),
    "value":np.random.randn(100)
})
#Define the colormap and alpha values
cmap="magma"
alpha=1
#create the scatterplot
plt.figure(figsize=(6,6))
plt.scatter(data["x"],data["y"],c=data["value"],cmap=cmap,alpha=alpha)
#customise the plot
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Scatterplot with ColorMap")
plt.colorbar(label="value")
#show the plot
plt.show()
```

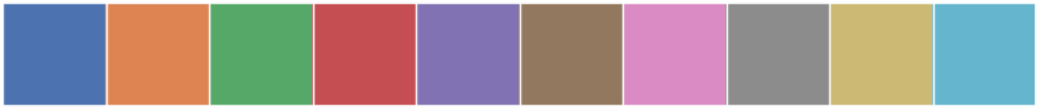


Seaborn

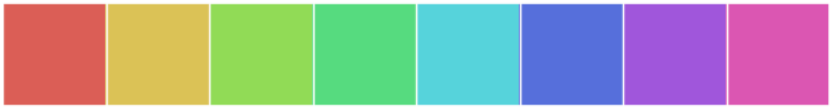
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
#Setting a figure size for all plots we shall be drawing
sns.set(rc={"figure.figsize":(6,6)})
```

building color palettes

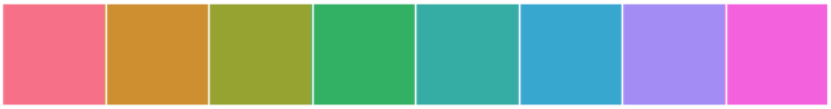
```
current_palette=sns.color_palette()
sns.palplot(current_palette)
```



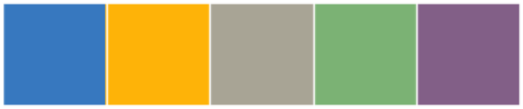
```
sns.palplot(sns.color_palette("hls",8))
```



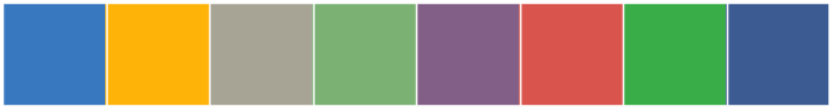
```
sns.palplot(sns.color_palette("husl",8))
```



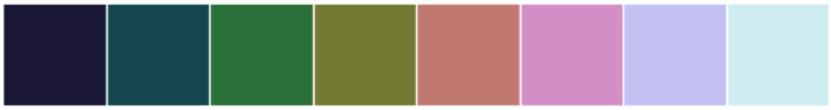
```
sample_colors=["windows blue","amber","greyish","faded green","dusty purple"]
sns.palplot(sns.xkcd_palette(sample_colors))
```



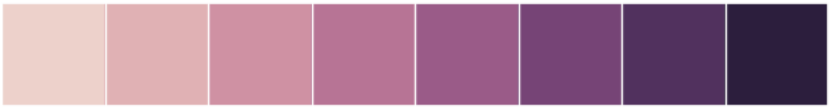
```
sample_colors=["windows blue","amber","greyish","faded green","dusty purple","pale red","medium green",'denim blue']
sns.palplot(sns.xkcd_palette(sample_colors))
```



```
#Default Matplotlib cubelix version:
sns.palplot(sns.color_palette("cubehelix",8))
```



```
#Default seaborn Cubehelix version
sns.palplot(sns.cubehelix_palette(8))
```



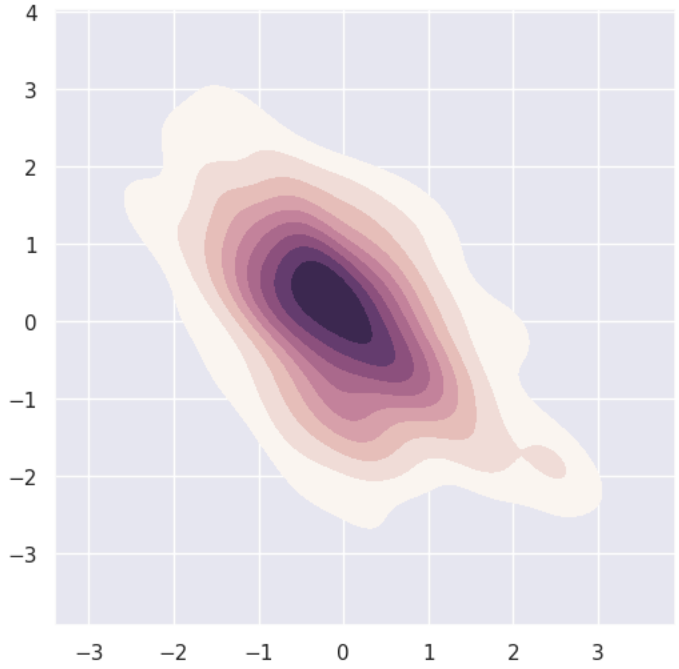
```
#Density Plot with Seaborn defaults:
x,y=np.random.multivariate_normal([0,0],[[1,-.5],[-.5,1]],size=300).T
```

```
sample_cmap=sns.cubehelix_palette(light=1,as_cmap=True)
sns.kdeplot(x=x,y=y,cmap=sample_cmap,shade=True)
```

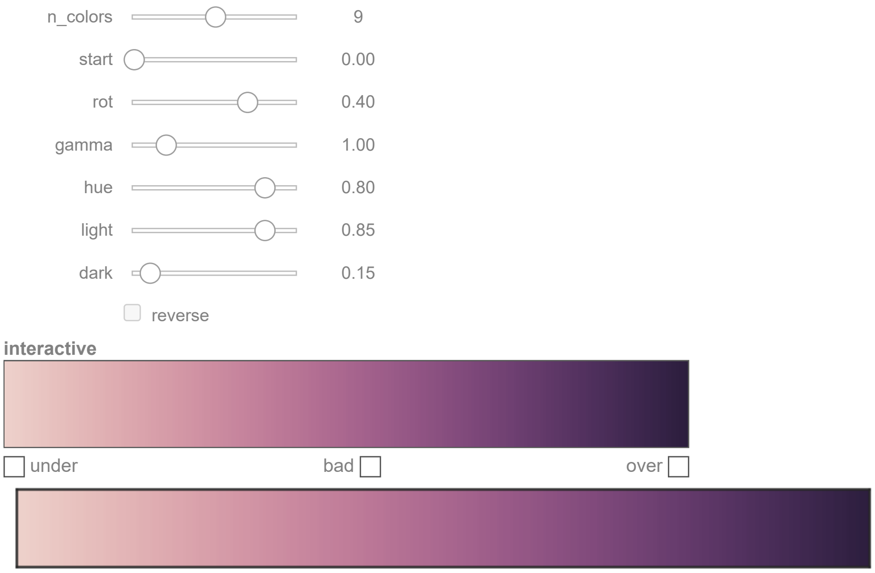
```
<ipython-input-28-59c5e391be82>:2: FutureWarning:

`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=x,y=y,cmap=sample_cmap,shade=True)
<Axes: >
```



```
sns.choose_cubehelix_palette(as_cmap=True)
```

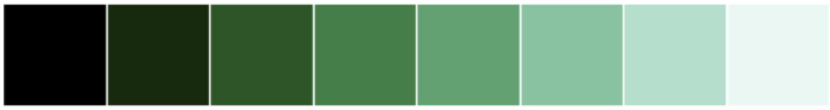


```
n_colors  9
start     0.00
rot       0.40
gamma     1.00
hue       0.80
light     0.85
dark      0.15
□ reverse
```

interactive

□ under bad □ over □

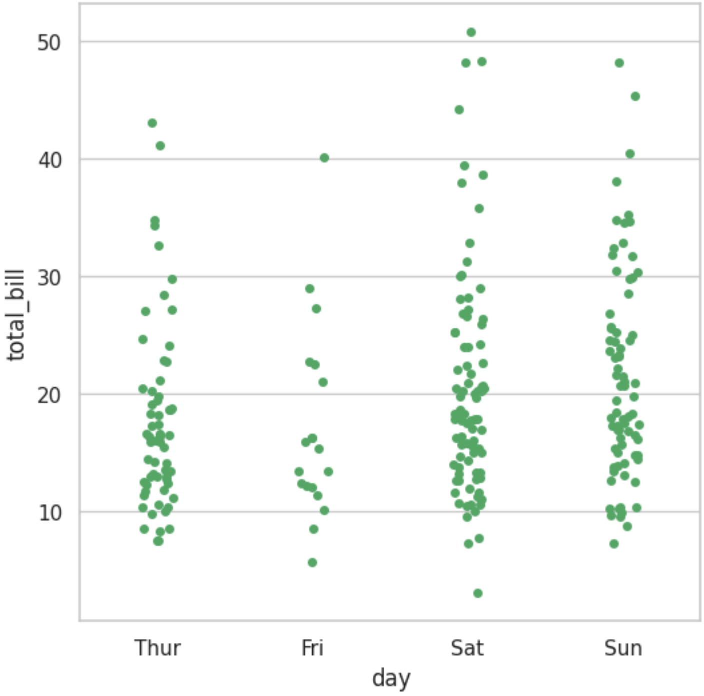
```
sns.palplot(sns.cubehelix_palette(n_colors=8,start=1.7,rot=0.2,dark=0,light=.95,reverse=True))
```



```
#Loading up built-in dataset
tips=sns.load_dataset("tips")
```

```
#Creating Strip plot for day-wise revenue:
sns.stripplot(x="day",y="total_bill",data=tips,color='g')
```

```
<Axes: xlabel='day', ylabel='total_bill'>
```



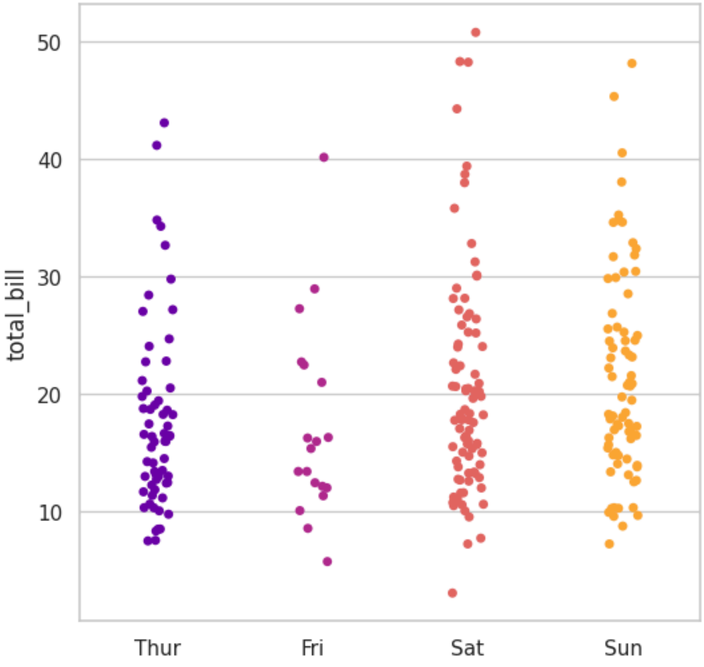
```
#Set Theme:
sns.set_style('whitegrid')
```

```
#Creating a Strip Plot
sns.stripplot(x="day",y="total_bill",data=tips,palette="plasma")
```

```
<ipython-input-36-5ec11283f2a8>: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

  sns.stripplot(x="day",y="total_bill",data=tips,palette="plasma")
<Axes: xlabel='day', ylabel='total_bill'>
```


total_bill
Thur
Fri
Sat
Sun