Data Visualization using Matplotlib and Seaborn

In [96]:
```
# Importing libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import style
```

In [97]:
```
randomNumbers=np.random.rand(10)
randomNumbers2=np.random.rand(10)
```

In [98]:
```
randomNumbers
```

Out[98]:
```
array([0.38754223, 0.17993719, 0.81599334, 0.57791852, 0.61037073,
       0.9203188 , 0.02759025, 0.75622235, 0.60651726, 0.51124737])
```

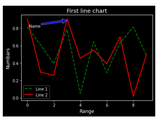
In [99]:
```
randomNumbers
```

Out[99]:
```
array([0.38754223, 0.17993719, 0.81599334, 0.57791852, 0.61037073,
       0.9203188 , 0.02759025, 0.75622235, 0.60651726, 0.51124737])
```

Line Chart

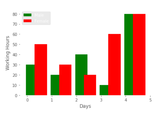
In [75]:
```
# Line Chart
# select the style
style.use('dark_background')
# Plot the random numers
plt.plot(randomNumbers,'green',label='Line 1', linewidth=2,linestyle='--')
plt.plot(randomNumbers2,"r",label='Line 2',linewidth=2)
# Label X axis and Y axis
plt.xlabel('Range')
plt.ylabel('Numbers')
# Title of the chart
plt.title('First line chart')
# plt.legend()
plt.legend()
# to remove grid line
plt.grid(b=None)
# Annotate
plt.annotate('Name',ha='right',va='bottom',xytext=(1,0.8), xy=(3,0.89),arrowprops={"facecolor":"blue"})
plt.show
```

Out[75]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Bar Plot

In [76]:
```
# select the style
style.use('ggplot')
axe = plt.axes()
axe.set(facecolor = 'white') # Removing the back ground
bplot1 = plt.bar([0.20, 1.20, 2.20, 3.20, 4.20], [30, 29, 40, 10, 80], label = 'Male', color = 'g', width = .5)
bplot2 = plt.bar([0.55, 1.55, 2.55, 3.55, 4.55], [50, 30, 20, 60, 89], label = 'Female', color = 'r', width = .5)
def nameLabel(bplot):# Function to define the label on top of bars
    for bar in bplot:
        bar_height = bar.get_height()
        plt.annotate(bar.get_height(),
                     xy=(bar.get_x() + bar.get_width() / 2, bar_height),
                     xytext=(0, 3),textcoords='offset points',ha='center',va='bottom')

nameLabel(bplot1)
nameLabel(bplot2)
plt.legend()
plt.title('Count of Male and Female')
plt.xlabel('Days')
plt.ylabel('Working Hours')
plt.show()
```

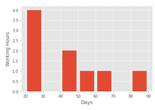


Pie Chart

In [100]:
```
slices = [12,25,50,38]
names_of_slices = ['Slice1','Slice2','Slice3', 'Slice4' ]
cols = ['c','g','k','w']
plt.pie(slices, labels=names_of_slices,
        colors= cols, startangle=90,
        shadow=True,explode = (0,0.1,0,0.1),
        autopct = '%1.1f%%')
plt.title('Pie chart')
plt.show()
```

Histogram

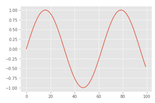
In [79]:
```
# Histogram

ages=[23,24,23,20,45,67,89,45,56]
binss=[20,30,40,50,60,70,80,90]
plt.hist(ages,bins, histtype ='bar', rwidth=0.8)
plt.xlabel("Days")
plt.ylabel("Working Hours")

# Title of the chart
plt.title('Count of Male and Female')

plt.show()
```



Time-Amplitude graph

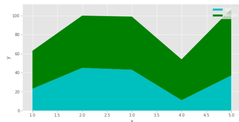
In [78]:
```
time = np.arange(0,10,0.1)
amplitude = np.sin(time)
plt.plot(amplitude)
```

Out[78]:
```
[<matplotlib.lines.Line2D at 0x2218c3d8400>]
```



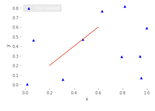
Area Plot

In [104]:
```
days = [1,2,3,4,5]
age = [23,45,43,11,37]
weights = [40, 55, 56, 45, 70]
plt.figure(figsize=(5,5))
plt.plot([], [], color = 'r', label = 'age', linewidth = 5)
plt.plot([],[], color = 'g', label = 'weight', linewidth = 5)
plt.stackplot(days, age, weights, colors=['r', 'g'])
plt.title('Area plot')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



Scatter Plot

In [90]:
```
axe = plt.axes()
axe.set(facecolor = 'white') # Removing the back ground
plt.scatter(randomNumbers,randomNumbers2, label= 'first dataset', color = 'b', marker='^', )
plt.plot([0.2,0.4, 0.5, 0.6 ],[0.2,0.4, 0.5, 0.6])
plt.title('Scatter plot')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

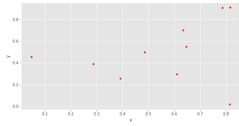


Sea born

In [82]:
```
import warnings
warnings.filterwarnings('ignore')
```

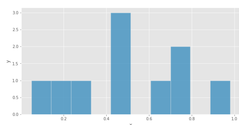
Scatter Plot

In [83]:
```
# Scatterplot using Seaborn

plt.figure(figsize=(10,5))
sns.scatterplot(randomNumbers,randomNumbers2)
plt.title('Area Plot')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Histogram

In [66]:
```
# Use of Sea born in Histogram


plt.figure(figsize=(10,5))
sns.histplot(randomNumbers, bins=10)
plt.title('Histogram')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Distribution Plot

In [67]:
```
# distribution plot


plt.figure(figsize=(10,5))
sns.distplot(randomNumbers, bins=10)
plt.title('Distribution')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

In [ ]:
```

```

In [ ]:
```

```