Adhish Krishna S
23N206
BE CSE(AI & ML)

23N411 Machine Learning Laboratory

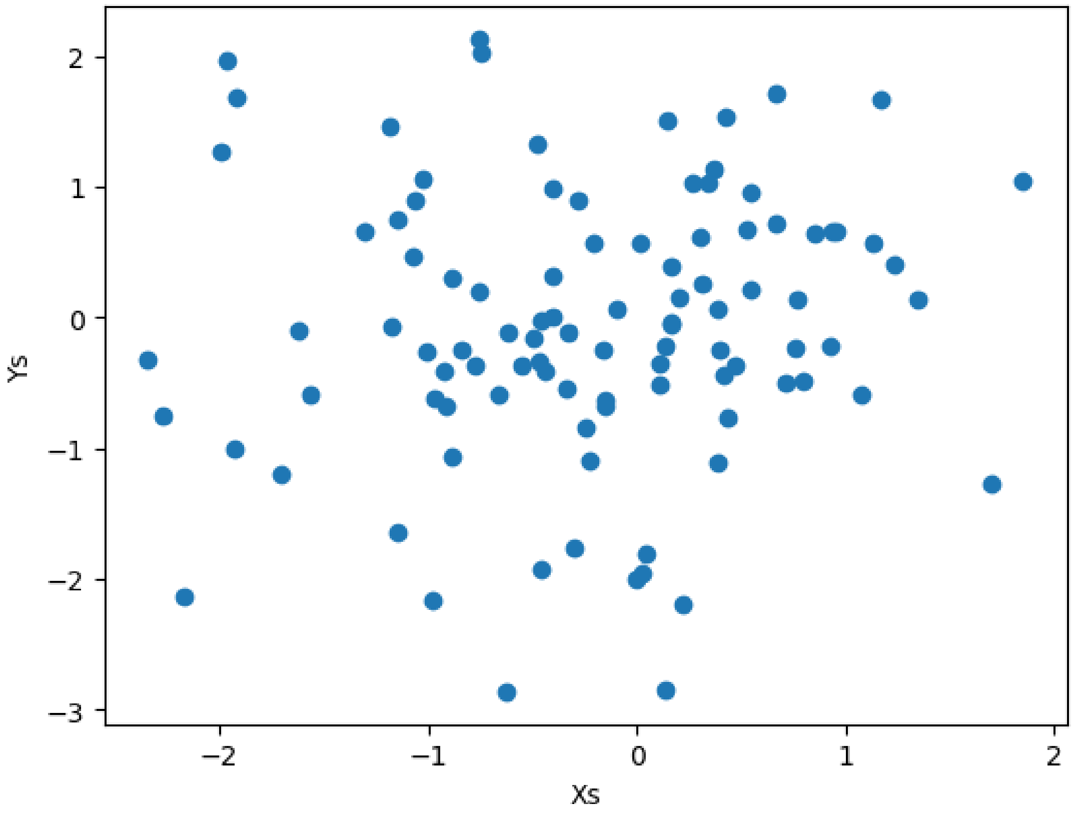
Matplotlib Exercises

Scatter Plot

1. Write a Python program to draw a scatter graph taking a random distribution in X and Y and plotted against each other.

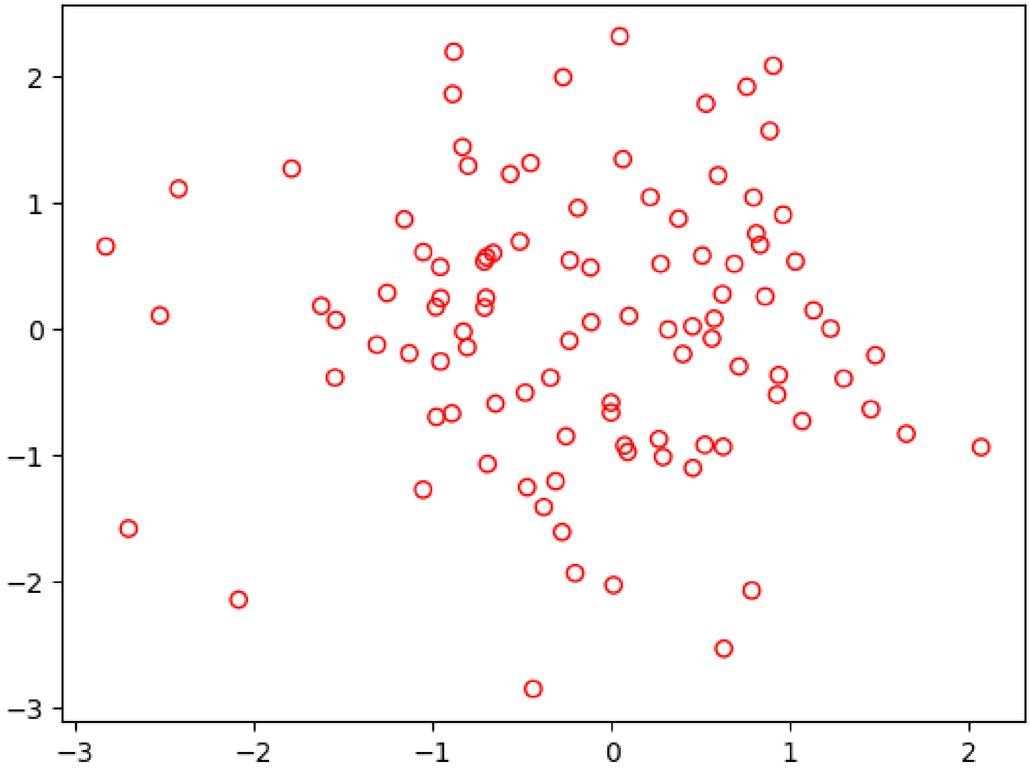
```
import matplotlib.pyplot as plt
import numpy as np
import random
import math
Xs = np.random.standard_normal(100)
Ys=  np.random.standard_normal(100)
plt.scatter(Xs, Ys , cmap='blue')
plt.xlabel("Xs")
plt.ylabel("Ys")
plt.show()
```

```
C:\Users\strea\AppData\Local\Temp\ipykernel_2544\1549478528.py:7:
UserWarning: No data for colormapping provided via 'c'. Parameters
'cmap' will be ignored
  plt.scatter(Xs, Ys , cmap='blue')
```



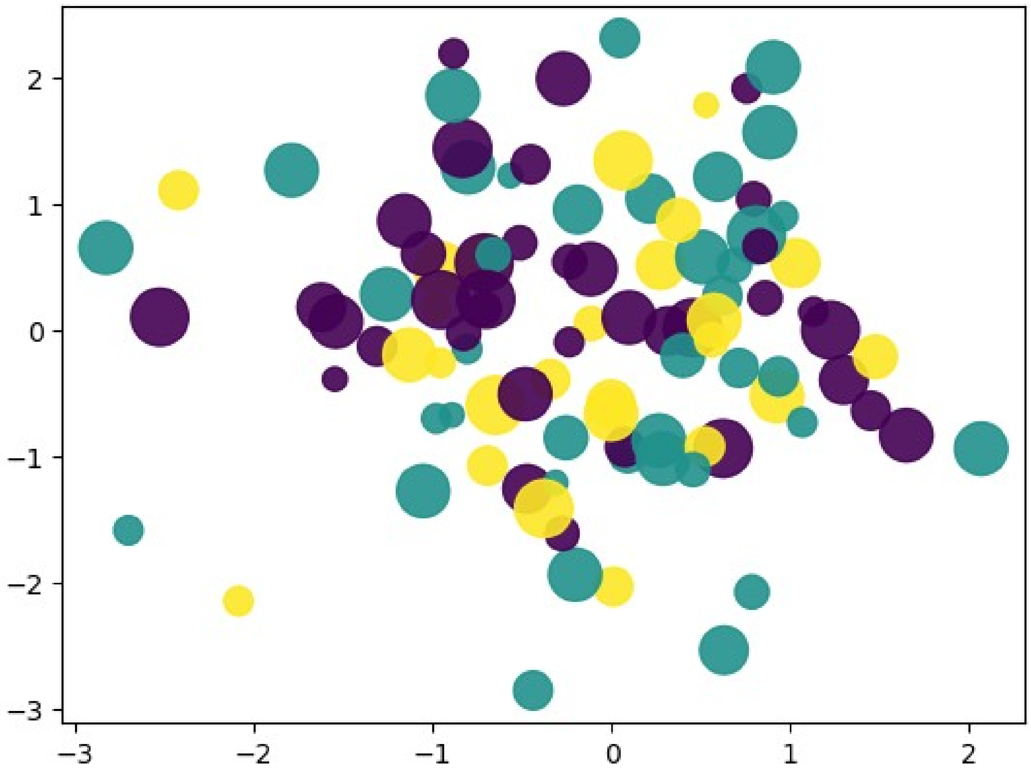
1. Write a Python program to draw a scatter plot with empty circles taking a random distribution in X and Y and plotted against each other.

```
plt.scatter(Xs , Ys,facecolors = "none" ,edgecolors="r")

<matplotlib.collections.PathCollection at 0x1b2f279eb10>
```



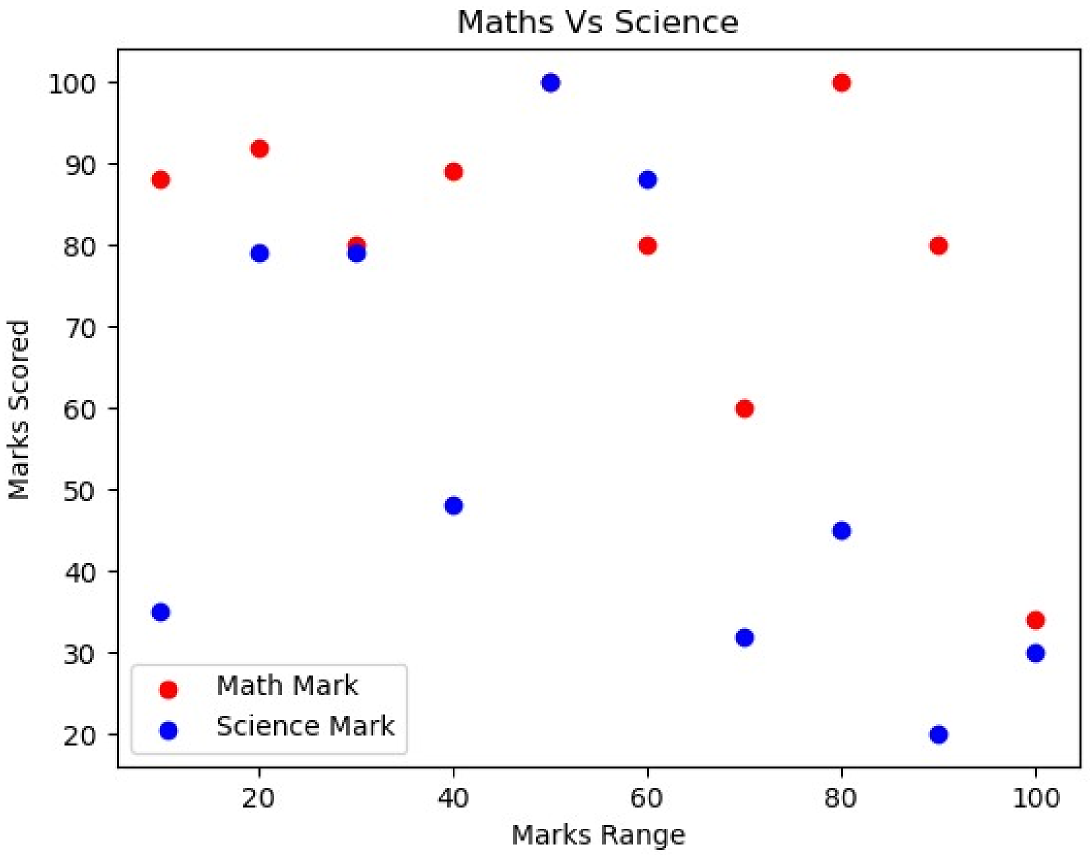
1. Write a Python program to draw a scatter plot using random distributions to generate balls of different sizes.

```
colors = [np.random.randint(1,4) for i in range(100)]
areas = [math.pi * random.randint(5, 12)**2 for i in range(100)]
plt.scatter(Xs, Ys ,s = areas ,c= colors , alpha= .9 )

<matplotlib.collections.PathCollection at 0x1b2f2731190>
```

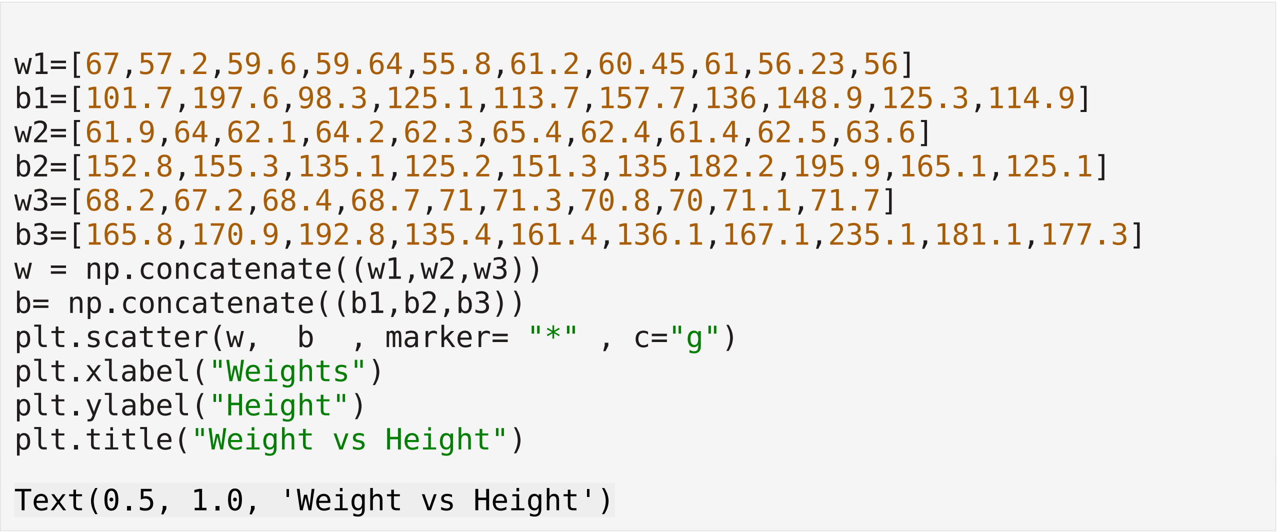


1. Write a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science

```
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

plt.scatter(marks_range , math_marks ,c = "r"  , label = "Math Mark")
plt.scatter(marks_range, science_marks,c = "b" ,  label = "Science 
Mark")
plt.title("Maths Vs Science")
plt.xlabel("Marks Range")
plt.ylabel('Marks Scored')
plt.legend()

<matplotlib.legend.Legend at 0x1b2f27acad0>
```

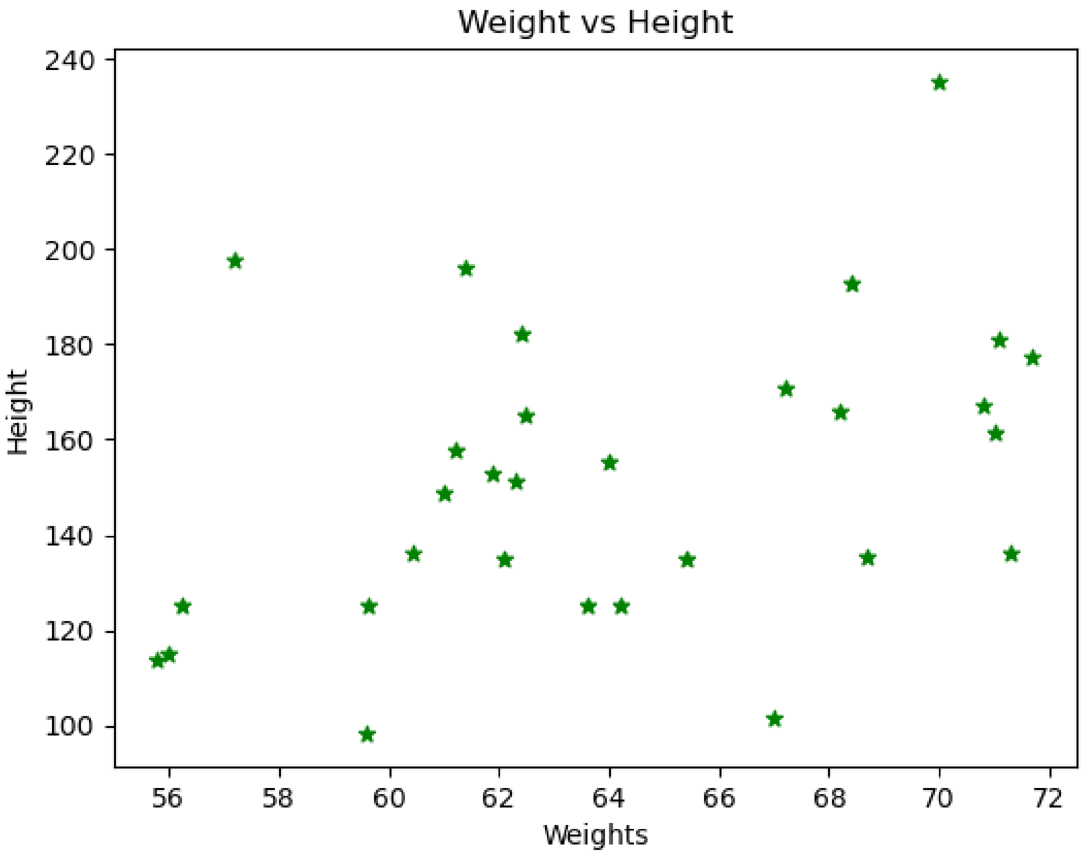


1. Write a Python program to draw a scatter plot for three different groups comparing weights and heights



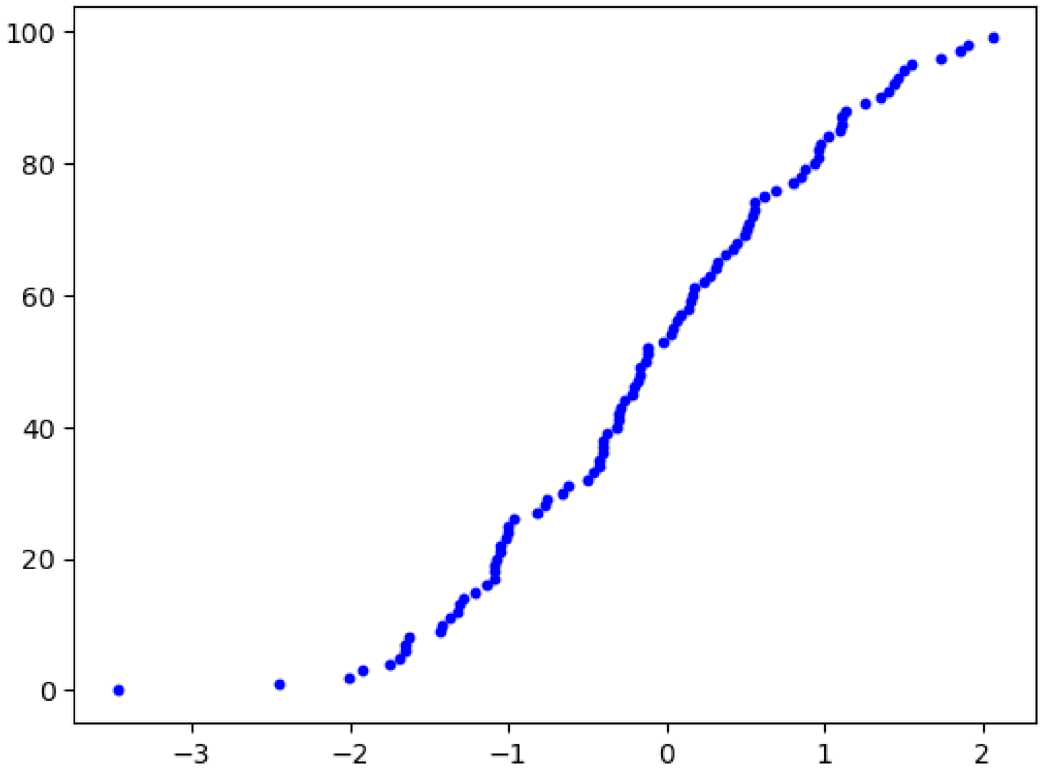
```
w1=[67,57.2,59.6,59.64,55.8,61.2,60.45,61,56.23,56]
b1=[101.7,197.6,98.3,125.1,113.7,157.7,136,148.9,125.3,114.9]
w2=[61.9,64,62.1,64.2,62.3,65.4,62.4,61.4,62.5,63.6]
b2=[152.8,155.3,135.1,125.2,151.3,135,182.2,195.9,165.1,125.1]
w3=[68.2,67.2,68.4,68.7,71,71.3,70.8,70,71.1,71.7]
b3=[165.8,170.9,192.8,135.4,161.4,136.1,167.1,235.1,181.1,177.3]
w = np.concatenate((w1,w2,w3))
b= np.concatenate((b1,b2,b3))
plt.scatter(w,  b  , marker= "*" , c="g")
plt.xlabel("Weights")
plt.ylabel("Height")
plt.title("Weight vs Height")
```

```
Text(0.5, 1.0, 'Weight vs Height')
```



1. Write a Python program to draw a scatter plot to find sea level rise in past 100 years.

```
years = np.arange(100)
sea_l = sorted([ np.random.standard_normal() for _ in range(100)])
plt.scatter(sea_l , years , c="b" , s=10)

<matplotlib.collections.PathCollection at 0x1b2f2891190>
```



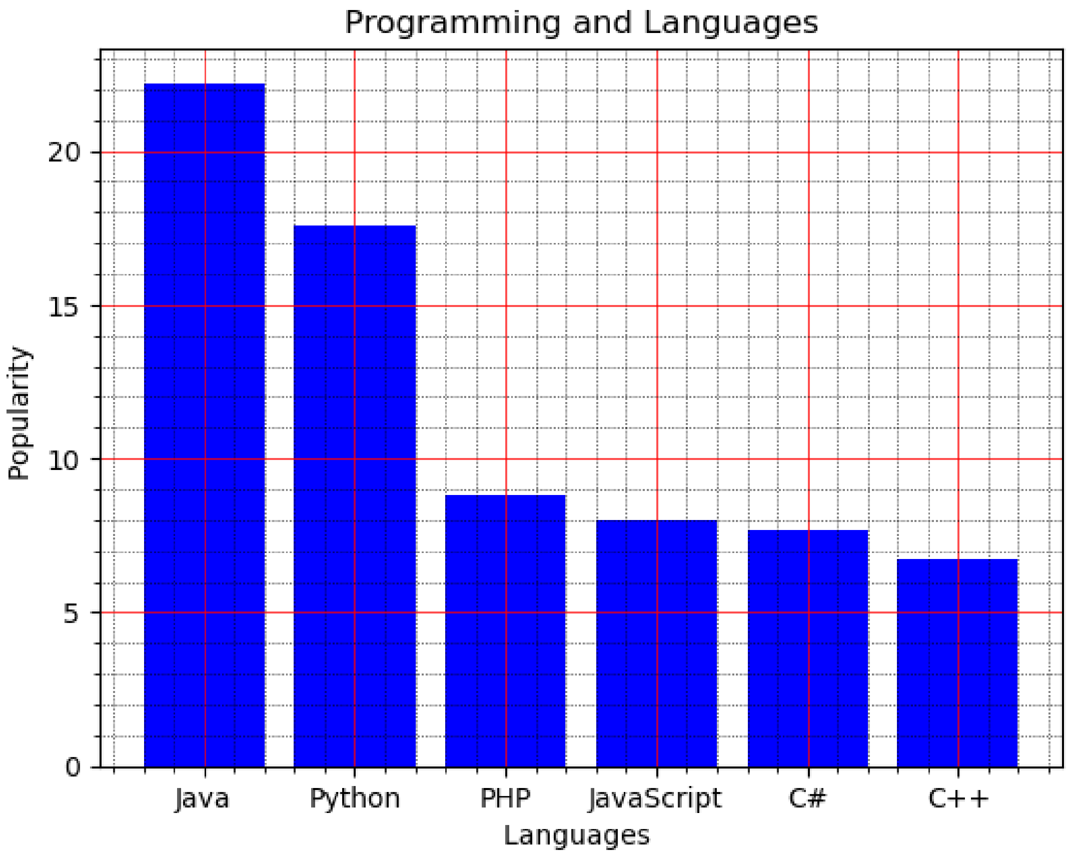
Barchart

```
import matplotlib.pyplot as plt
```

1. Write a Python programming to display a bar chart of the popularity of programming Languages.

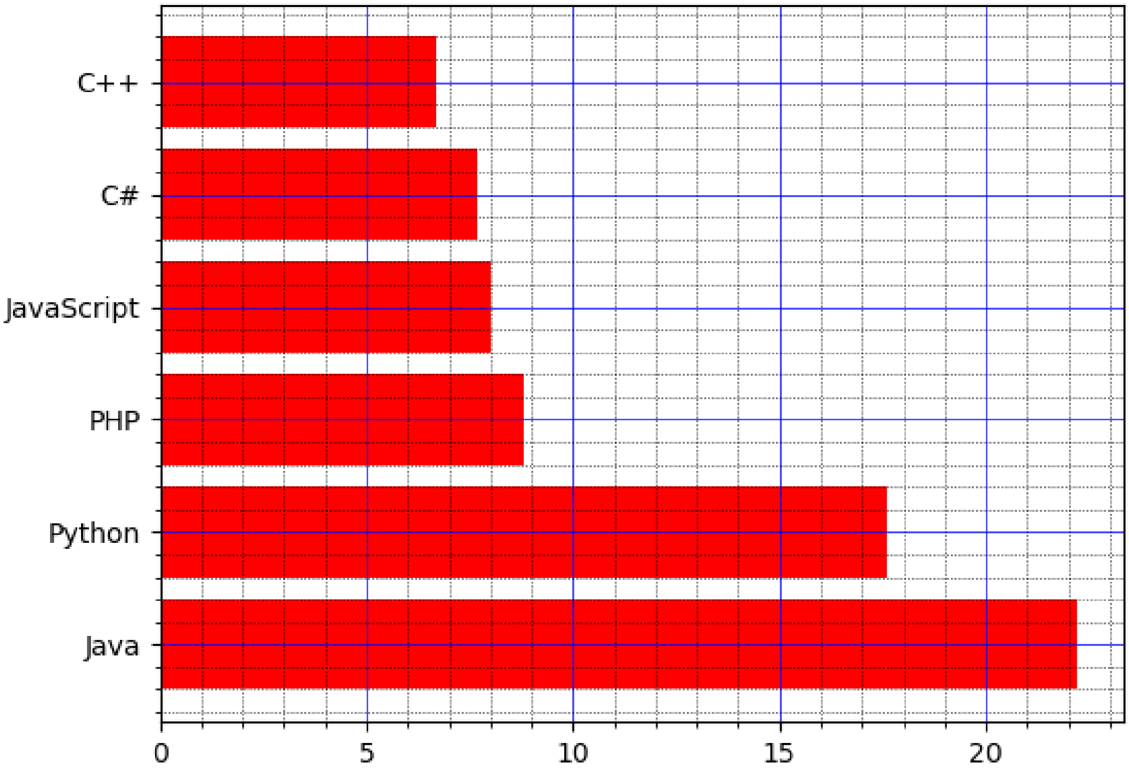
```
Programming  = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
Popularity = [ 22.2, 17.6, 8.8, 8, 7.7, 6.7]
x=  [ i for i, _ in enumerate(Programming)]
plt.bar(x, Popularity , color ="blue")
plt.xticks(x, Programming)

plt.minorticks_on()
plt.grid(which='major' ,linestyle = "-" ,linewidth = '.5' , color =
'red')
plt.grid(which='minor' ,linestyle = ":" ,linewidth = '.5' , color =
'black')

plt.xlabel("Languages ")
plt.ylabel("Popularity ")
plt.title("Programming and Languages")
```

```
Text(0.5, 1.0, 'Programming and Languages')
```



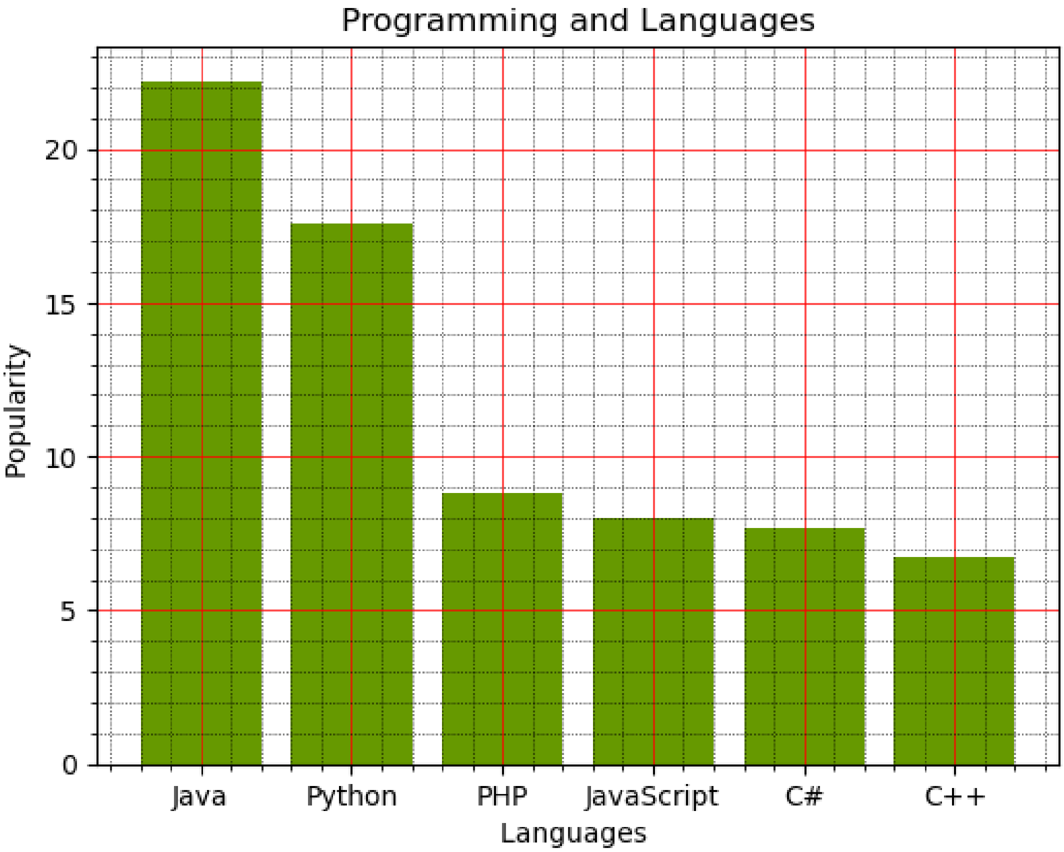
1. Write a Python programming to display a horizontal bar chart of the popularity of programming Languages

```
plt.barh(x, Popularity , color = "red")
plt.yticks(x, Programming)
plt.minorticks_on()
plt.grid(which= "minor" , linestyle = ":" , linewidth = ".5" , color =
"black")

plt.grid(which= "major" , linestyle = "-" , linewidth = ".5" , color =
"blue")
```



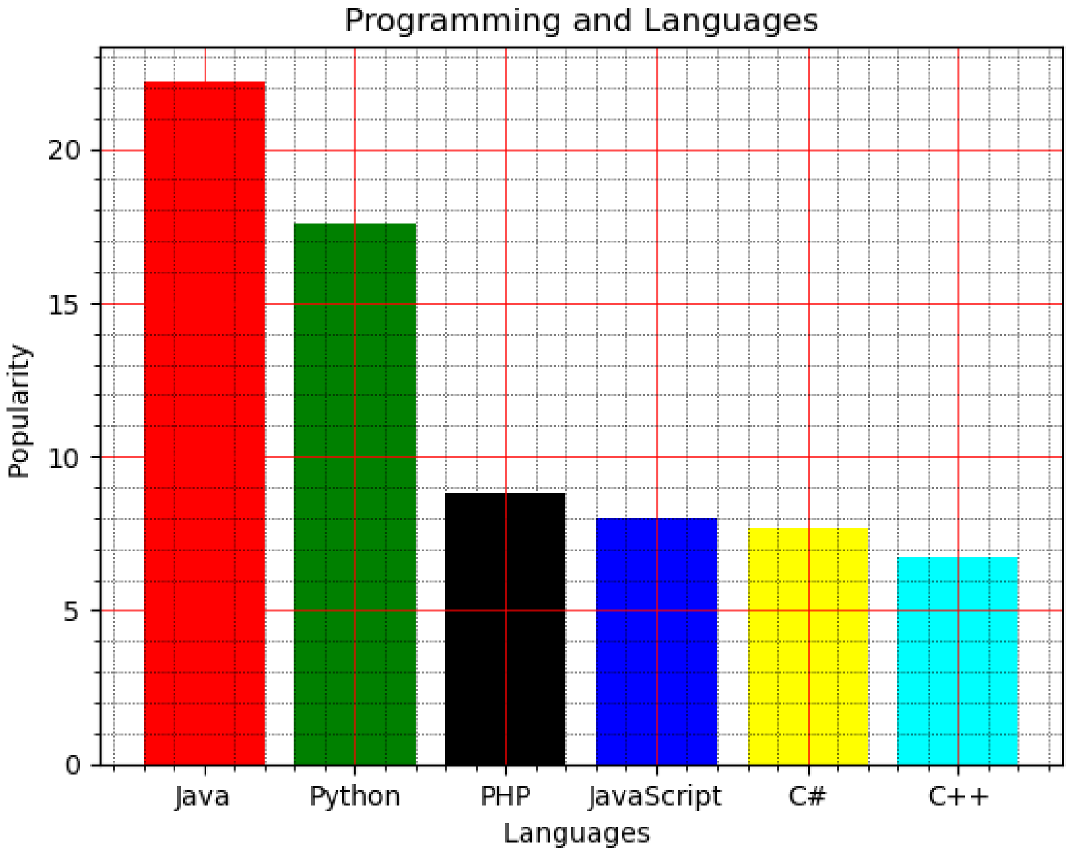
1. Write a Python programming to display a bar chart of the popularity of programming Languages. Use uniform color.

```
Programming  = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
Popularity = [ 22.2, 17.6, 8.8, 8, 7.7, 6.7]
x=  [ i for i, _ in enumerate(Programming)]
plt.bar(x, Popularity , color =(.4, .6 , 0,1))
plt.xticks(x, Programming)

plt.minorticks_on()
plt.grid(which='major' ,linestyle = "-" ,linewidth = '.5' , color = 
'red')
plt.grid(which='minor' ,linestyle = ":" ,linewidth = '.5' , color = 
'black')

plt.xlabel("Languages ")
plt.ylabel("Popularity ")
plt.title("Programming and Languages")

Text(0.5, 1.0, 'Programming and Languages')
```



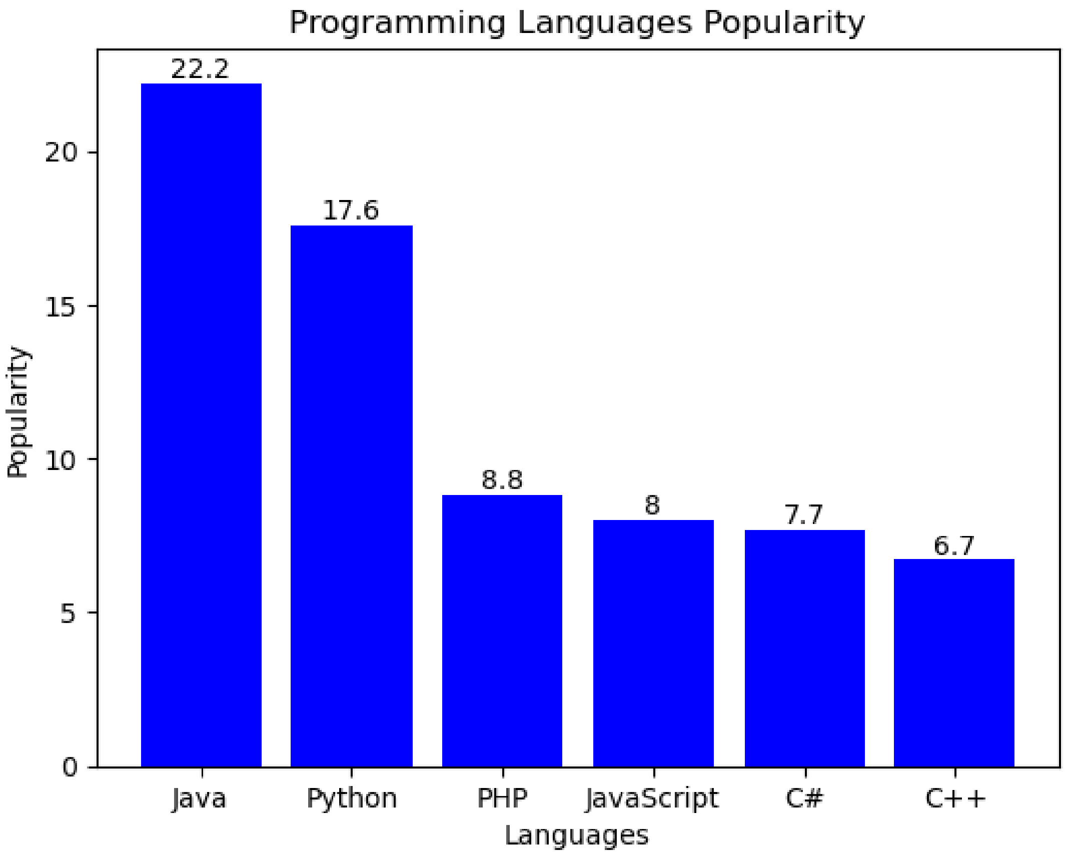
```
Programming  = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
colors = [ "red" , "green", "black" , "blue",  "yellow" , "cyan"]
Popularity = [ 22.2, 17.6, 8.8, 8, 7.7, 6.7]
x=  [ i for i, _ in enumerate(Programming)]
plt.bar(x, Popularity , color =colors)
plt.xticks(x, Programming)

plt.minorticks_on()
plt.grid(which='major' ,linestyle = "-" ,linewidth = '.5' , color =
'red')
plt.grid(which='minor' ,linestyle = ":" ,linewidth = '.5' , color =
'black')

plt.xlabel("Languages ")
plt.ylabel("Popularity ")
plt.title("Programming and Languages")
```

```
Text(0.5, 1.0, 'Programming and Languages')
```



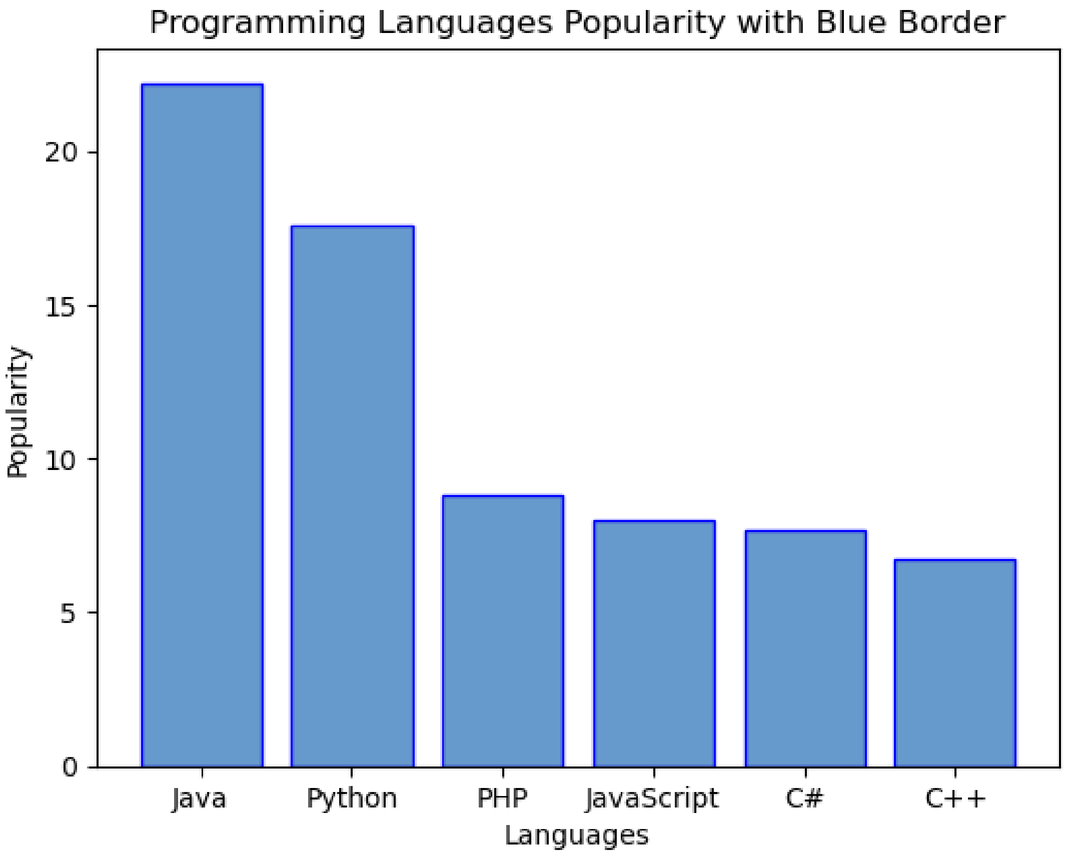
1. Write a Python programming to display a bar chart of the popularity of programming Languages. Attach a text label above each bar displaying its popularity (float value).

```
Programming  = ["Java", "Python", "PHP", "JavaScript", "C#", "C++"]
Popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
x = [i for i, _ in enumerate(Programming)]

plt.bar(x, Popularity, color="blue")
plt.xticks(x, Programming)

for i, v in enumerate(Popularity):
    plt.text(i, v , str(v), ha='center', va='bottom')

plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("Programming Languages Popularity")
plt.show()
```



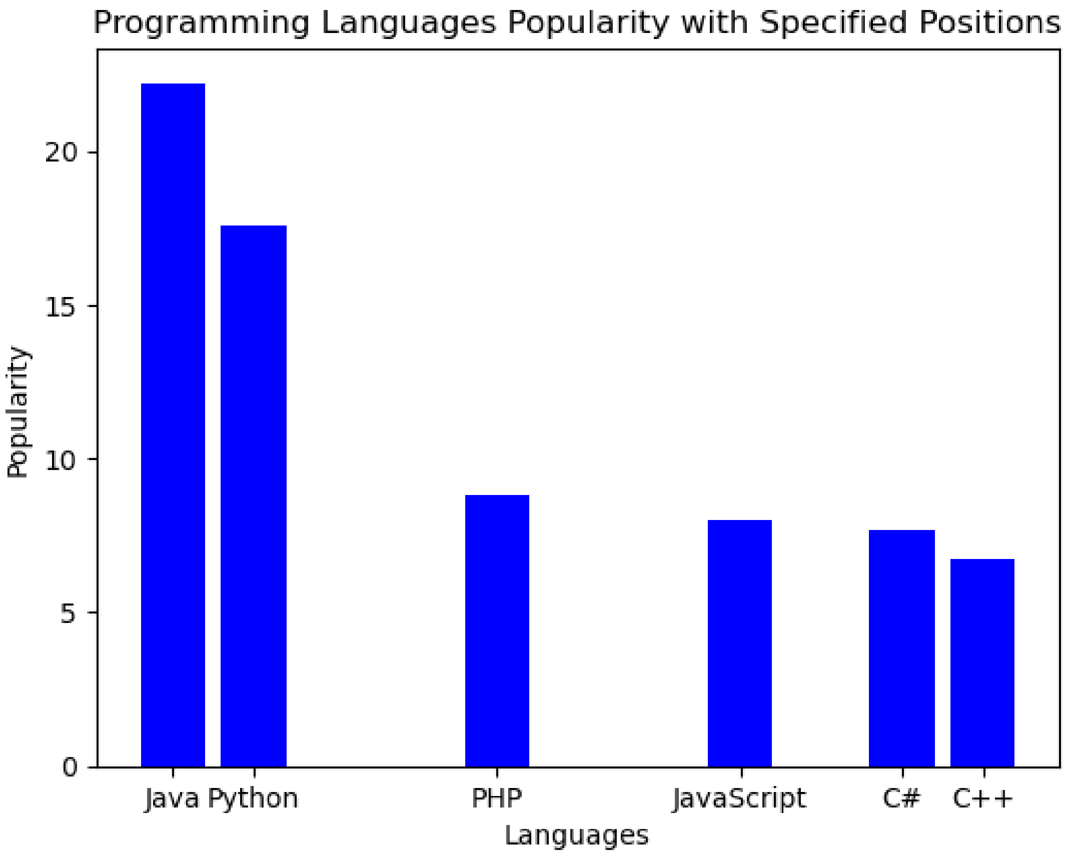
1. Write a Python programming to display a bar chart of the popularity of programming Languages. Make blue border to each bar.

```
plt.bar(x, Popularity, color= (.4,.6 ,.8 ,1), edgecolor="blue")
plt.xticks(x, Programming)

plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("Programming Languages Popularity with Blue Border")
plt.show()
```

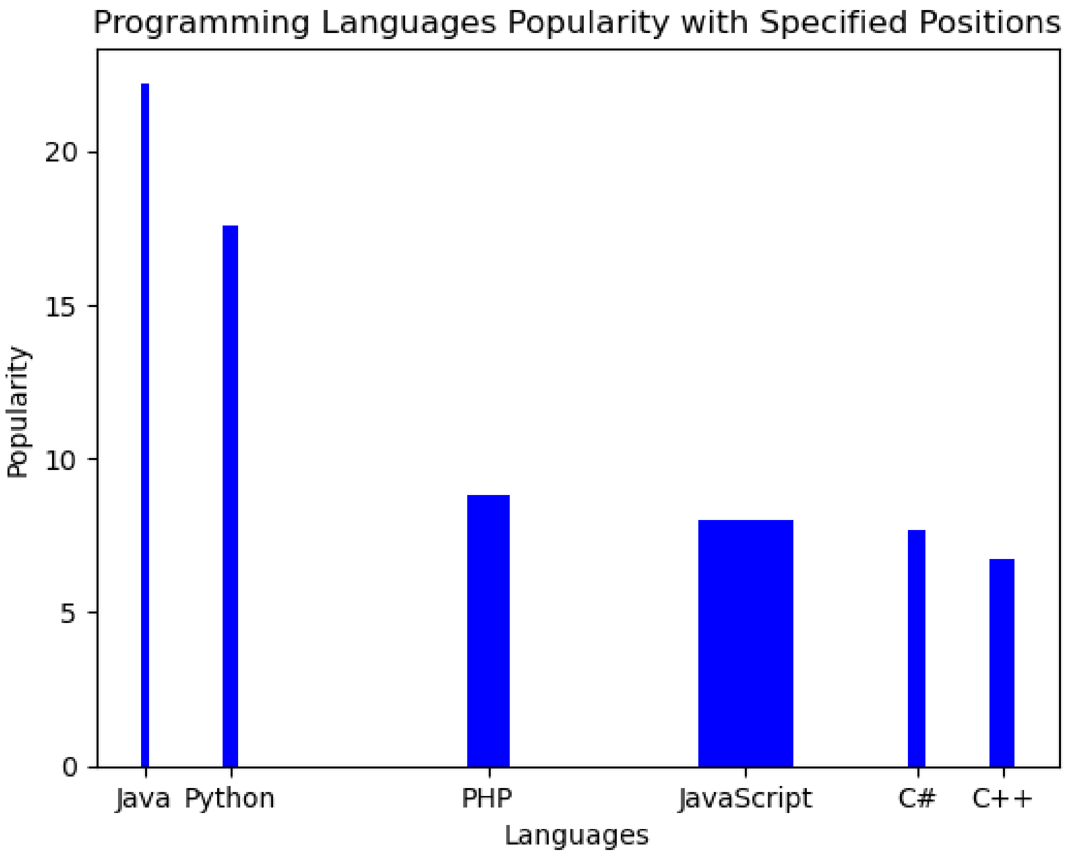


1. Write a Python programming to display a bar chart of the popularity of programming Languages. Specify the position of each bar plot.

```
positions = [0,1,4,7,9,10]

plt.bar(positions, Popularity, color="blue")
plt.xticks(positions, Programming)

plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("Programming Languages Popularity with Specified Positions")
plt.show()
```



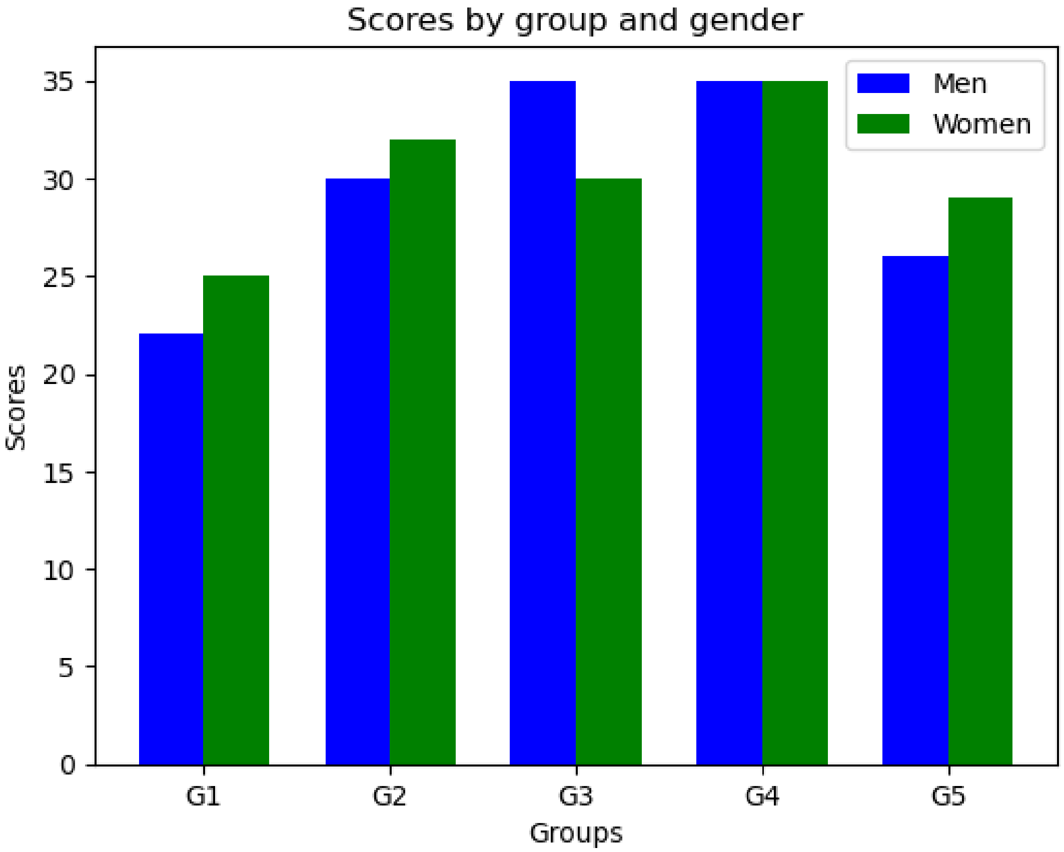
```
positions = [0,1,4,7,9,10]
width = [0.1,0.2,0.5,1.1,0.2,0.3]

plt.bar(positions, Popularity, color="blue", width= width)
plt.xticks(positions, Programming)

plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("Programming Languages Popularity with Specified Positions")
plt.show()
```



1. Write a Python program to create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.

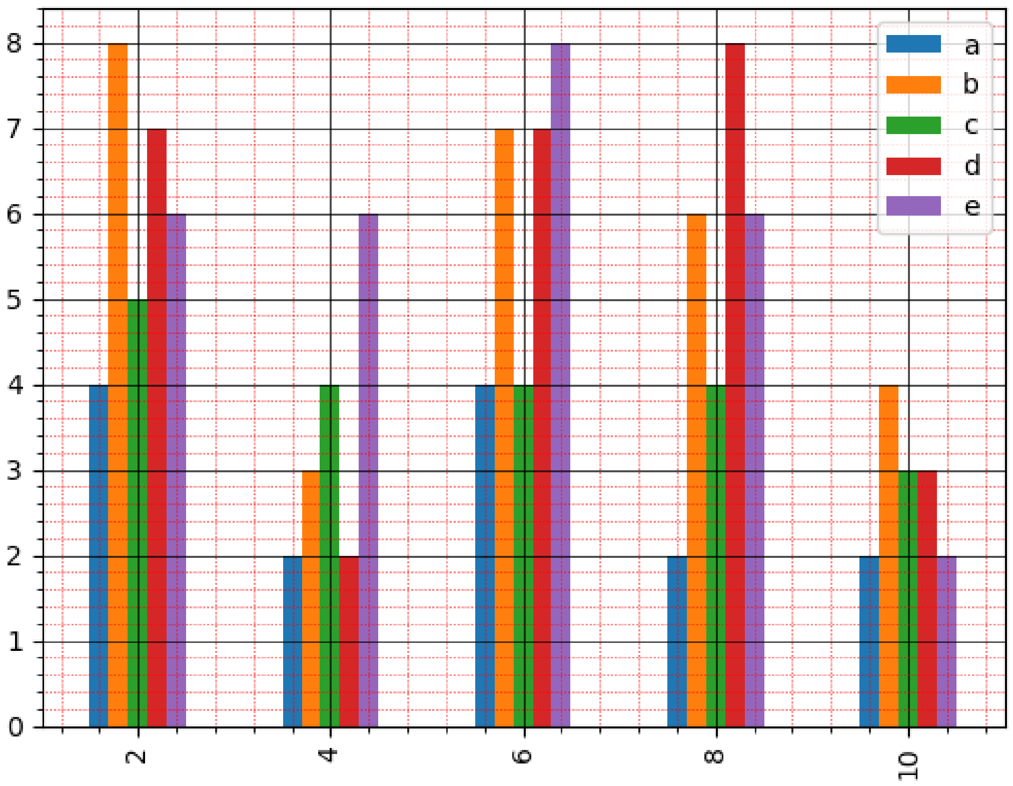
Sample Data: Means (men) = (22, 30, 35, 35, 26) Means (women) = (25, 32, 30, 35, 29)

```
means_men = [22, 30, 35, 35, 26]

means_women = [25, 32, 30, 35, 29]

groups = np.arange(len(means_men))


width = 0.35

fig, ax = plt.subplots()
bars1 = ax.bar(groups-width/2, means_men,width, label ='Men',color
="blue")
bars2 = ax.bar(groups + width/2,means_women, width,label='Women',color
="green")

ax.set_xlabel('Groups')
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(groups)
ax.set_xticklabels(['G1', 'G2', 'G3', 'G4', 'G5'])
ax.legend()

plt.show()
```



11 Write a Python program to create bar plot from a DataFrame.

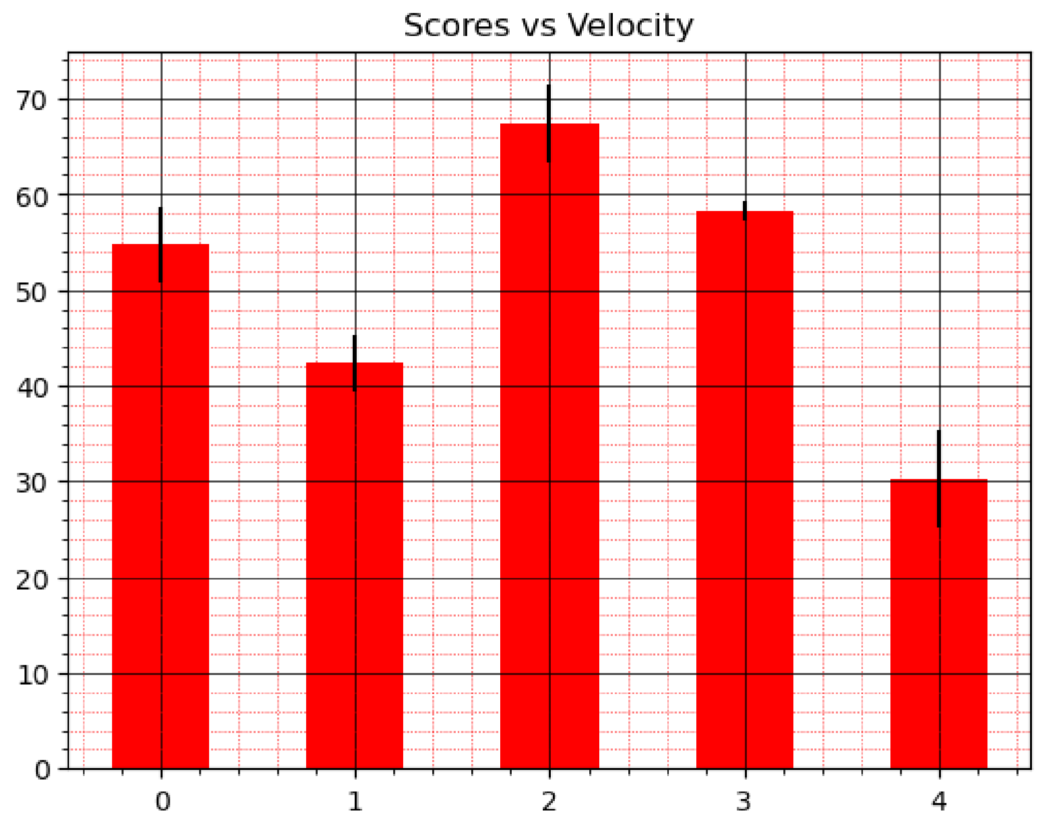
```
from turtle import color
import pandas as pd
a=np.array([[4,8,5,7,6],[2,3,4,2,6],[4,7,4,7,8],[2,6,4,8,6],
[2,4,3,3,2]])
df=pd.DataFrame(a, columns=['a','b','c','d','e'], index=[2,4,6,8,10])
df.plot(kind="bar")
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='black')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='red')

plt.show()
```



12 Write a Python program to create bar plots with errorbars on the same figure.

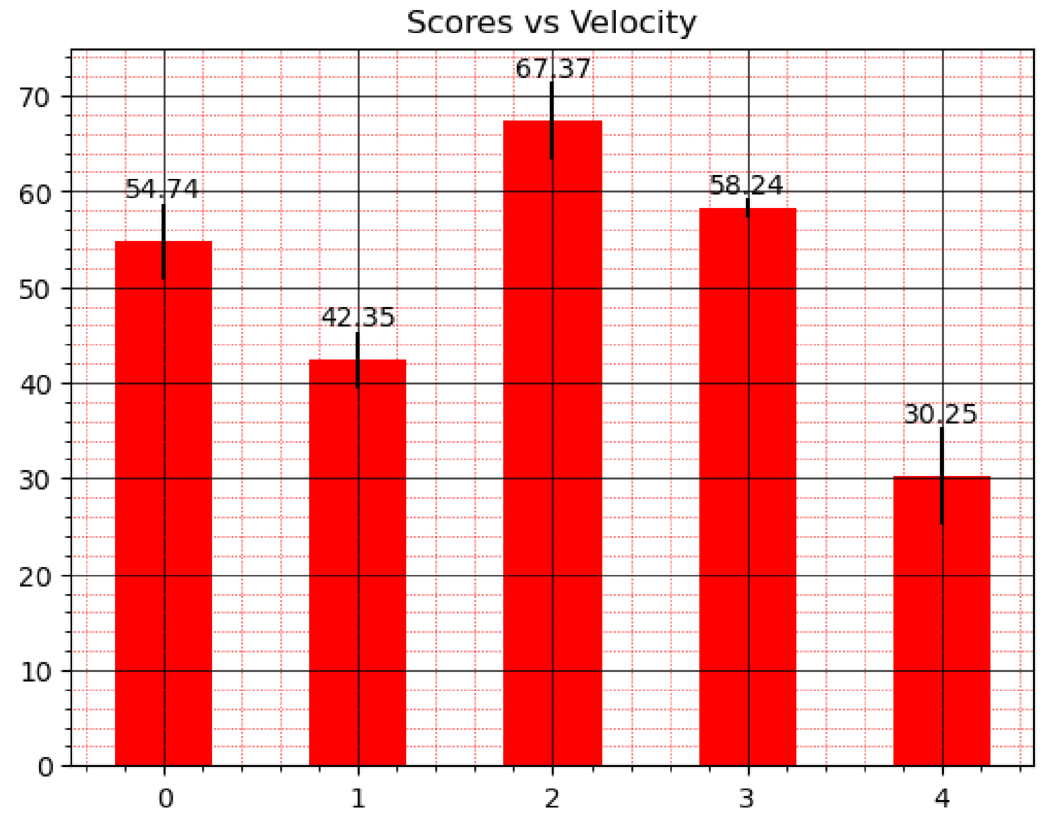
```
N = 5
menMeans = (54.74, 42.35, 67.37, 58.24, 30.25)
menStd = (4, 3, 4, 1, 5)
x_ticks =np.arange(N)
plt.bar(x_ticks ,menMeans  , width=.5 , yerr = menStd, color = 'red')
plt.minorticks_on()
plt.grid(which= 'minor' , linestyle= ':' , linewidth = '.5' ,color
='red')
plt.grid(which= 'major'  ,  linestyle = "-" , linewidth = ".5" , color
= "black")
plt.title("Scores vs Velocity")
```

```
Text(0.5, 1.0, 'Scores vs Velocity')
```



1. Write a Python program to create bar plots with errorbars on the same figure. Attach a text label above each bar displaying men means (integer value).

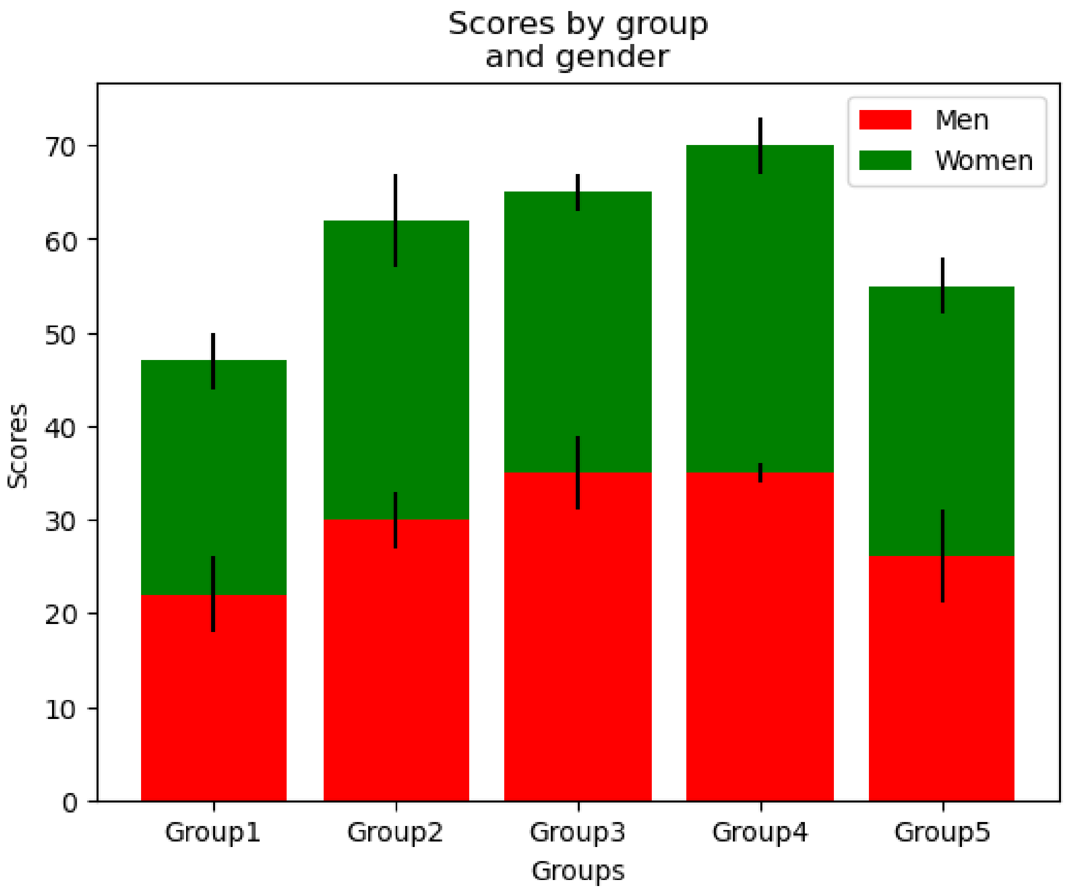
```
N = 5
menMeans = (54.74, 42.35, 67.37, 58.24, 30.25)
menStd = (4, 3, 4, 1, 5)
x_ticks =np.arange(N)
plt.bar(x_ticks ,menMeans  , width=.5 , yerr = menStd, color = 'red')
plt.minorticks_on()
plt.grid(which= 'minor' , linestyle= ':' , linewidth = '.5' ,color
='red')
plt.grid(which= 'major'  ,  linestyle = "-" , linewidth = ".5" , color
= "black")
for i , v in enumerate(menMeans):
    plt.text(i, v +menStd[i], str(v) , ha= "center" , va = "bottom" )
plt.title("Scores vs Velocity")

Text(0.5, 1.0, 'Scores vs Velocity')
```



14.Write a Python program to create a stacked bar plot with error bars.

```
Means_m = (22, 30, 35, 35, 26)
Means_w = (25, 32, 30, 35, 29)
Men_sd = (4, 3, 4, 1, 5)
Women_sd = (3, 5, 2, 3, 3)
i = np.arange(5)


p1 = plt.bar(i , Means_m , color = "red" ,yerr = Men_sd)
p2 = plt.bar(i , Means_w , color = "green" , bottom=Means_m,yerr =
Women_sd)
plt.legend((p1[0] , p2[0] ), ("Men", "Women"))

plt.ylabel('Scores')
plt.xlabel('Groups')
plt.title('Scores by group\n' + 'and gender')
plt.xticks(i, ('Group1', 'Group2', 'Group3', 'Group4', 'Group5'))
```

```
([<matplotlib.axis.XTick at 0x1b2f6aef770>,
  <matplotlib.axis.XTick at 0x1b2f6799c10>,
  <matplotlib.axis.XTick at 0x1b2f69af740>,
  <matplotlib.axis.XTick at 0x1b2f69ac6e0>,
  <matplotlib.axis.XTick at 0x1b2f69acfe0>],
 [Text(0, 0, 'Group1'),
  Text(1, 0, 'Group2'),
  Text(2, 0, 'Group3'),
  Text(3, 0, 'Group4'),
  Text(4, 0, 'Group5')])
```

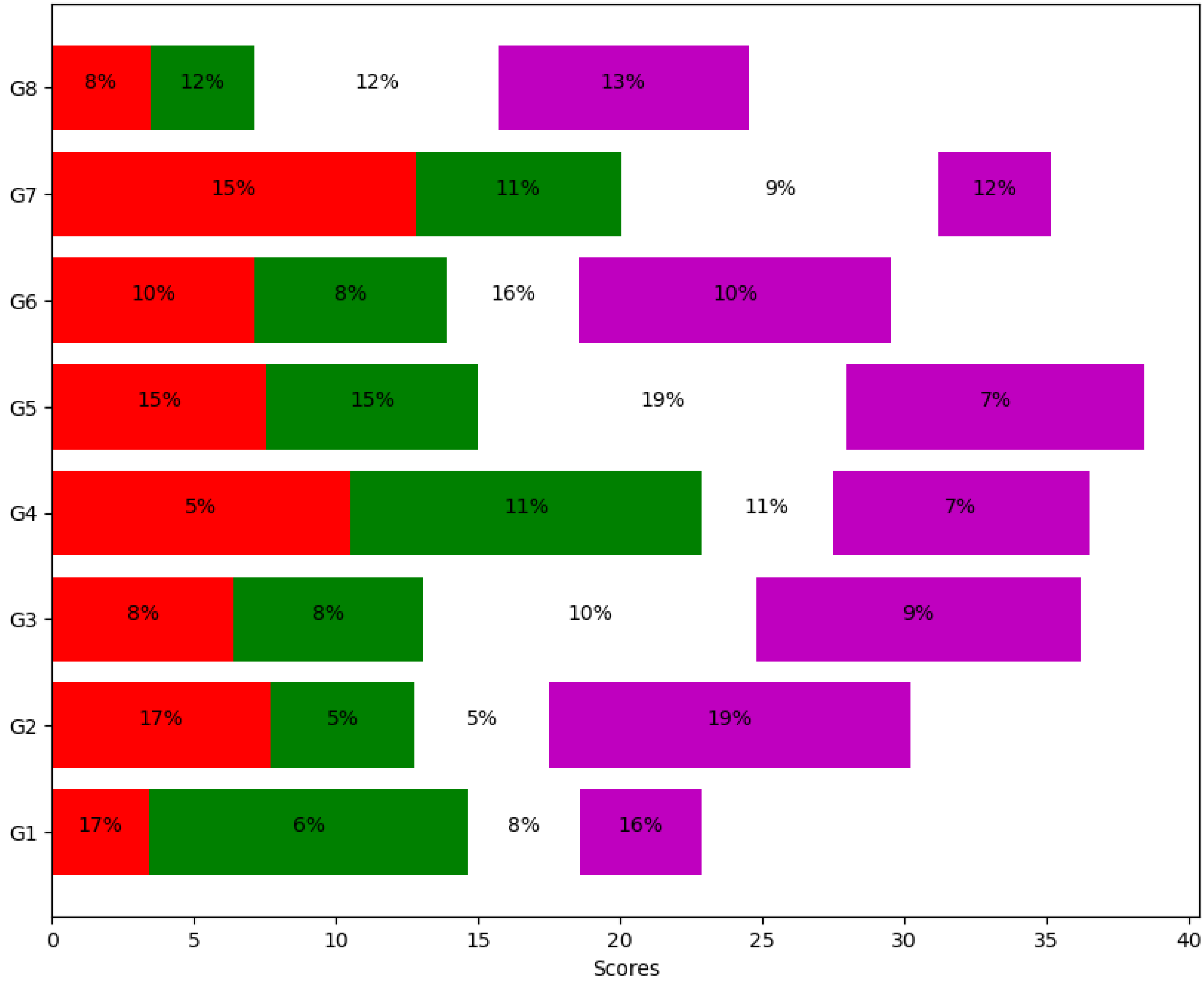


```
languages = [['Language', 'Science', 'Math'],
             ['Science', 'Math', 'Language'],
             ['Math', 'Language', 'Science']]
num_groups = len(numbers)

numbers = [{'Language': 75, 'Science': 88, 'Math': 96},
           {'Language': 71, 'Science': 95, 'Math': 92},
           {'Language': 75, 'Science': 90, 'Math': 89}]

color = {'Language': 'red', 'Science': 'blue', 'Math': 'green'}
fig, ax = plt.subplots(figsize=(10, 6))

for i , (l,d) in enumerate(zip(languages , numbers)):
    bottom  = 0

    for lang in l :
        ax.barh(y = f'Student-{num_groups - i}',width= d[lang] , color
= color[lang], edgecolor = "black", left = bottom , label=lang if i ==
0 else "")
        bottom += d[lang]

# for i, (language_order, data) in enumerate(zip(languages, numbers)):
#     bottom = 0  # Initialize stacking from the base (0)
#     for lang in language_order:
#         # Plot the horizontal bar
#         ax.barh(y=f'Student-{num_groups - i}', width=data[lang],
color=colors[lang], edgecolor='black', left=bottom, label=lang if i ==
0 else "")

#         bottom += data[lang]  # Increment the bottom for stacking


# Add a legend
ax.legend(loc='upper right')

# Add labels and title
ax.set_xlabel('Scores')
ax.set_title('Stacked Horizontal Bar Chart')

# Add y-axis ticks
ax.set_yticks(range(num_groups))
ax.set_yticklabels([f'Student-{i+1}' for i in range(num_groups)])

# Adjust layout and display
plt.tight_layout()
plt.show()
```

```
----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[21], line 4
      1 languages = [['Language', 'Science', 'Math'],
      2              ['Science', 'Math', 'Language'],
      3              ['Math', 'Language', 'Science']]
----> 4 num_groups = len(numbers)
      6 numbers = [{'Language': 75, 'Science': 88, 'Math': 96},
      7            {'Language': 71, 'Science': 95, 'Math': 92},
      8            {'Language': 75, 'Science': 90, 'Math': 89}]
     10 color = {'Language': 'red', 'Science': 'blue', 'Math':
'green'}

NameError: name 'numbers' is not defined
```

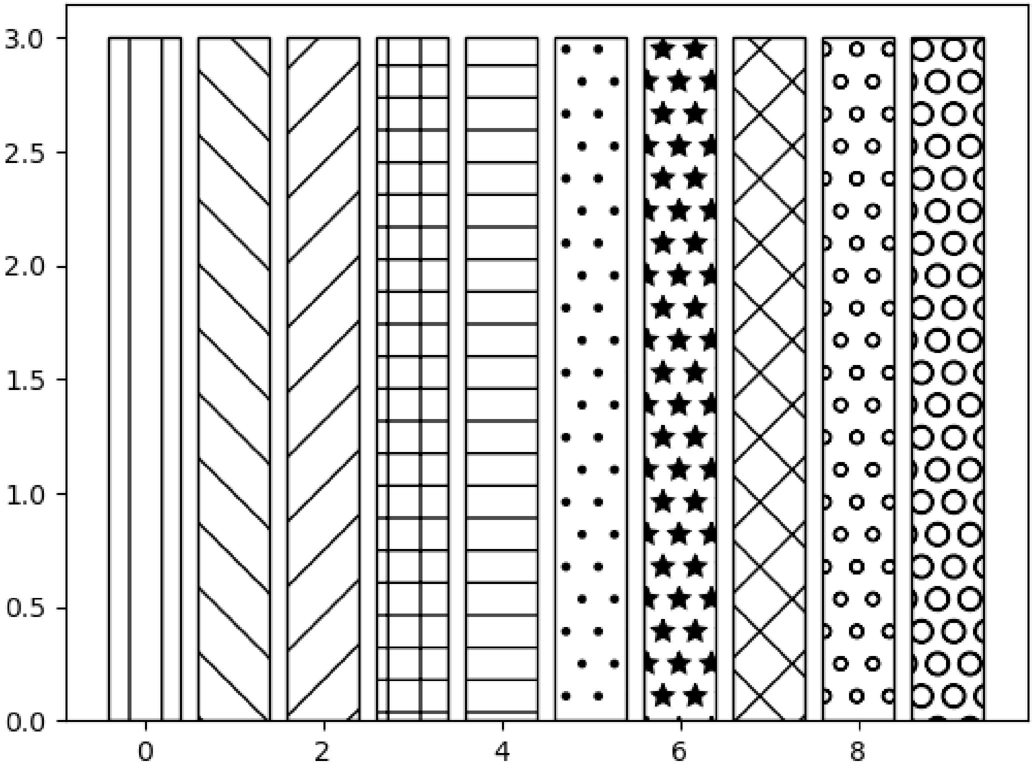
16 Write a Python program to create stack bar plot and add label to each section.

```
import numpy as np
import matplotlib.pyplot as plt

people = ('G1','G2','G3','G4','G5','G6','G7','G8')
segments = 4
data = [[  3.40022085,   7.70632498,   6.4097905,   10.51648577,
7.5330039,
    7.1123587,   12.77792868,   3.44773477],
 [ 11.24811149,   5.03778215,   6.65808464,  12.32220677,
7.45964195,
    6.79685302,   7.24578743,   3.69371847],
 [  3.94253354,   4.74763549,  11.73529246,   4.6465543,   12.9952182,
    4.63832778,  11.16849999,   8.56883433],
 [  4.24409799,  12.71746612,  11.3772169,    9.00514257,
10.47084185,
   10.97567589,   3.98287652,   8.80552122]]
percentages = (np.random.randint(5,20, (len(people), segments)))
y_pos = np.arange(len(people))

fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111)

colors ='rgwm'
patch_handles = []
left = np.zeros(len(people))
for i, d in enumerate(data):
    patch_handles.append(ax.barh(y_pos, d,
      color=colors[i%len(colors)], align='center',
      left=left))
    left += d

for j in range(len(patch_handles)):
    for i, patch in enumerate(patch_handles[j].get_children()):
        bl = patch.get_xy()
        x = 0.5*patch.get_width() + bl[0]
        y = 0.5*patch.get_height() + bl[1]
        ax.text(x,y, "%d%%" % (percentages[i,j]), ha='center')

ax.set_yticks(y_pos)
ax.set_yticklabels(people)
ax.set_xlabel('Scores')
plt.show()
```



17 Write a Python program to add textures (black and white) to bars and wedges.

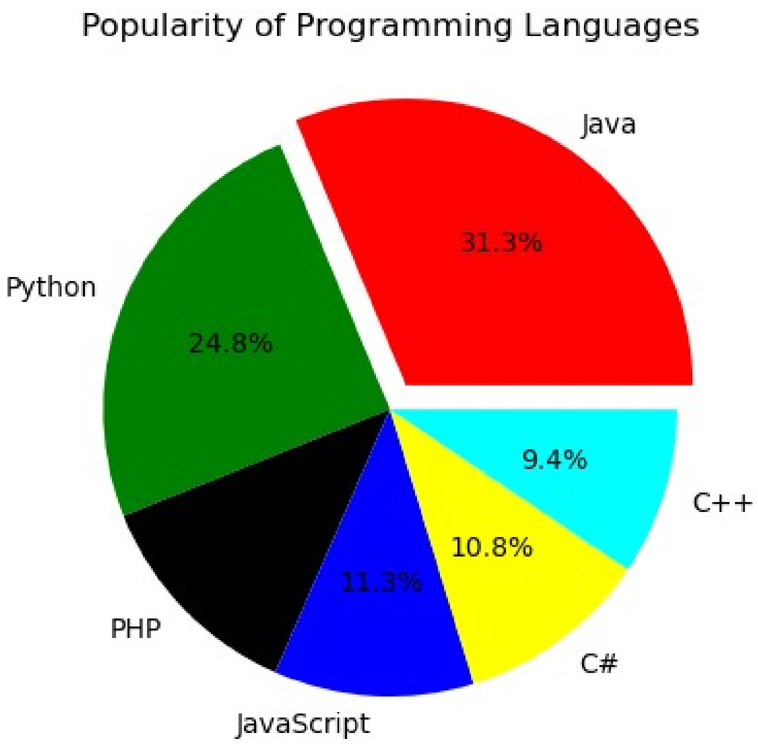
```
from matplotlib import hatch


patterns = [ "|" , "\\" , "/" , "+" , "-", ".", "*","x", "o", "O" ]

for i in range(len(patterns)):
    plt.bar(i , 3 , hatch = patterns[i]  , edgecolor = "black" ,color=
"white")
```



Pie Chart

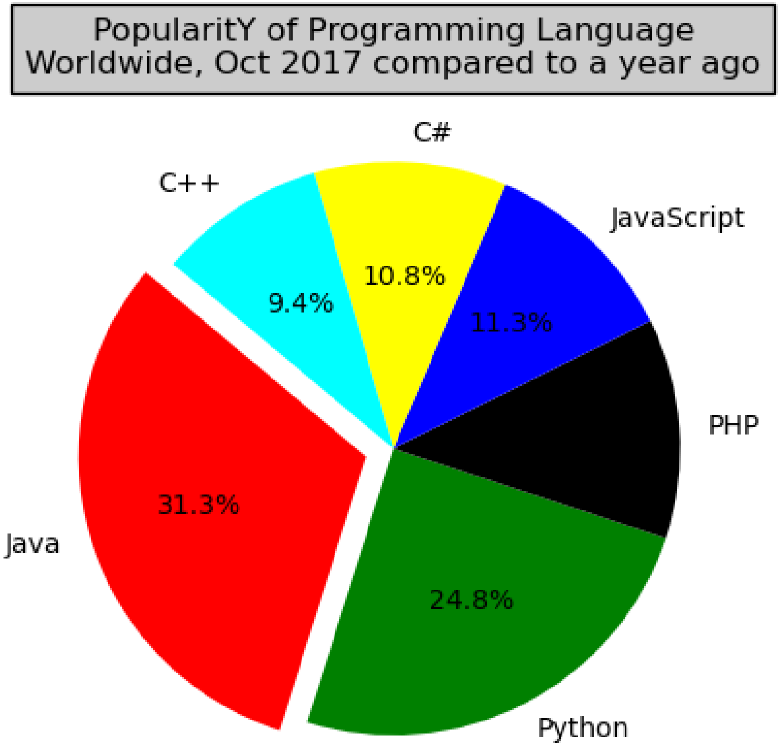
1. Write a Python programming to create a pie chart of the popularity of programming Languages. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

```
explode = (0.1, 0, 0, 0,0,0)

plt.pie(Popularity, labels=Programming, colors=colors, autopct='%1.1f%
%', explode= explode, )
plt.title("Popularity of Programming Languages")
plt.show()
```



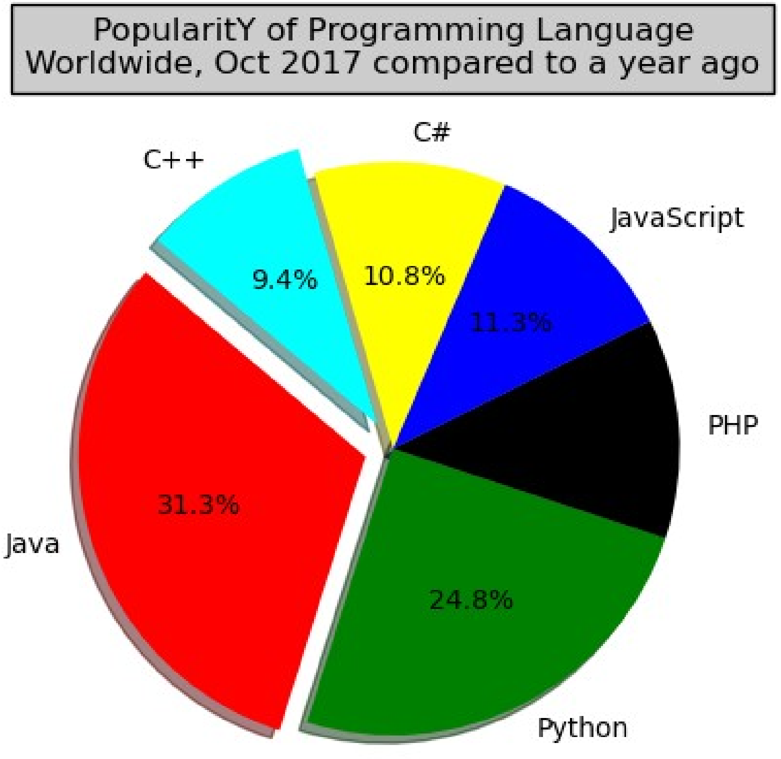
2 Write a Python programming to create a pie chart with a title of the popularity of programming Languages.

```
explode = (0.1, 0, 0, 0,0,0)

plt.pie(Popularity, labels=Programming, colors=colors, autopct='%1.1f%
%', explode= explode, startangle=140)
plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct
2017 compared to a year ago", bbox={'facecolor':'0.8', 'pad':5})
plt.show()
```



1. Write a Python programming to create a pie chart with a title of the popularity of programming Languages.Make multiple wedges of the pie.

```
explode = (0.1, 0, 0, 0,0,.1)

plt.pie(Popularity, labels=Programming, colors=colors, autopct='%1.1f%
%', explode= explode, shadow= True , startangle=140)
plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct
2017 compared to a year ago", bbox={'facecolor':'0.8', 'pad':5})
plt.show()
```

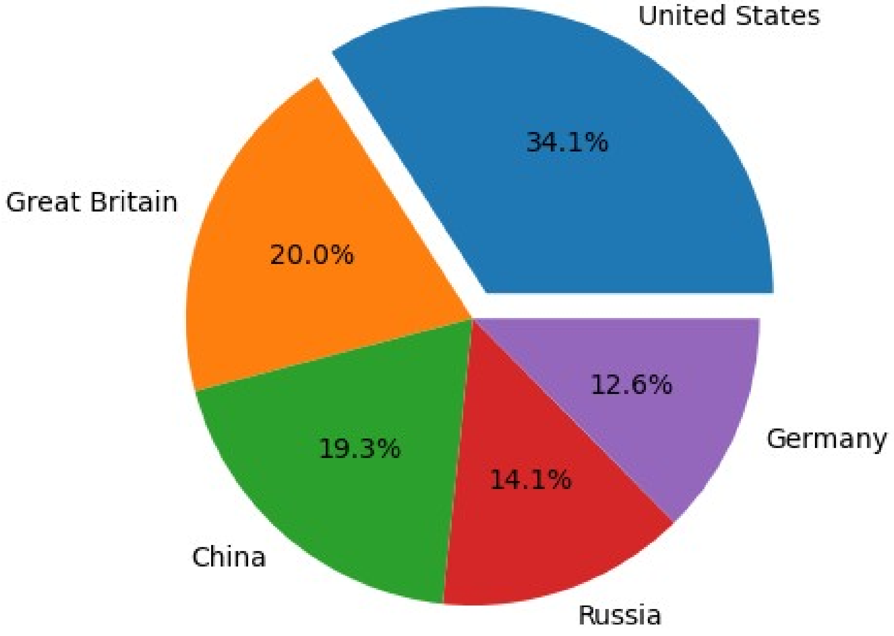


1. Write a Python programming to create a pie chart of gold medal achievements of five most successful countries in 2016 Summer Olympics. Read the data from a csv file.

```
import pandas as pd
df = pd.read_csv("medals.csv")
plt.pie(df["gold_medal"], explode=(.1,0,0,0,0) ,labels= df["country"]
, autopct= '%1.1f%%')
plt.title("Gold medal achievements of five most successful\
n"+"countries in 2016 Summer Olympics" ,bbox = {"facecolor": '.7' ,
"pad" :.5})
```

```
Text(0.5, 1.0, 'Gold medal achievements of five most successful\
ncountries in 2016 Summer Olympics')
```

Gold medal achievements of five most successful countries in 2016 Summer Olympics


United States
Great Britain
China
Russia
Germany