Project - Data Analysis Python Library project related thier variables

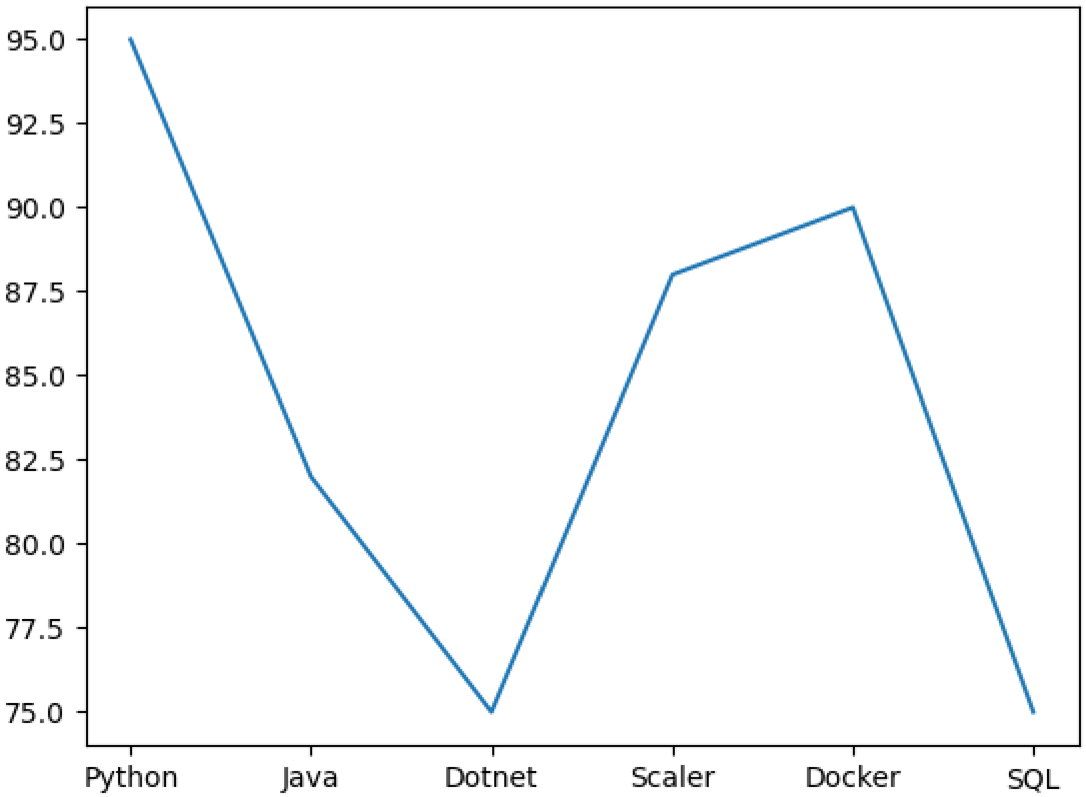
Matplotlib Library

In [1]:

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

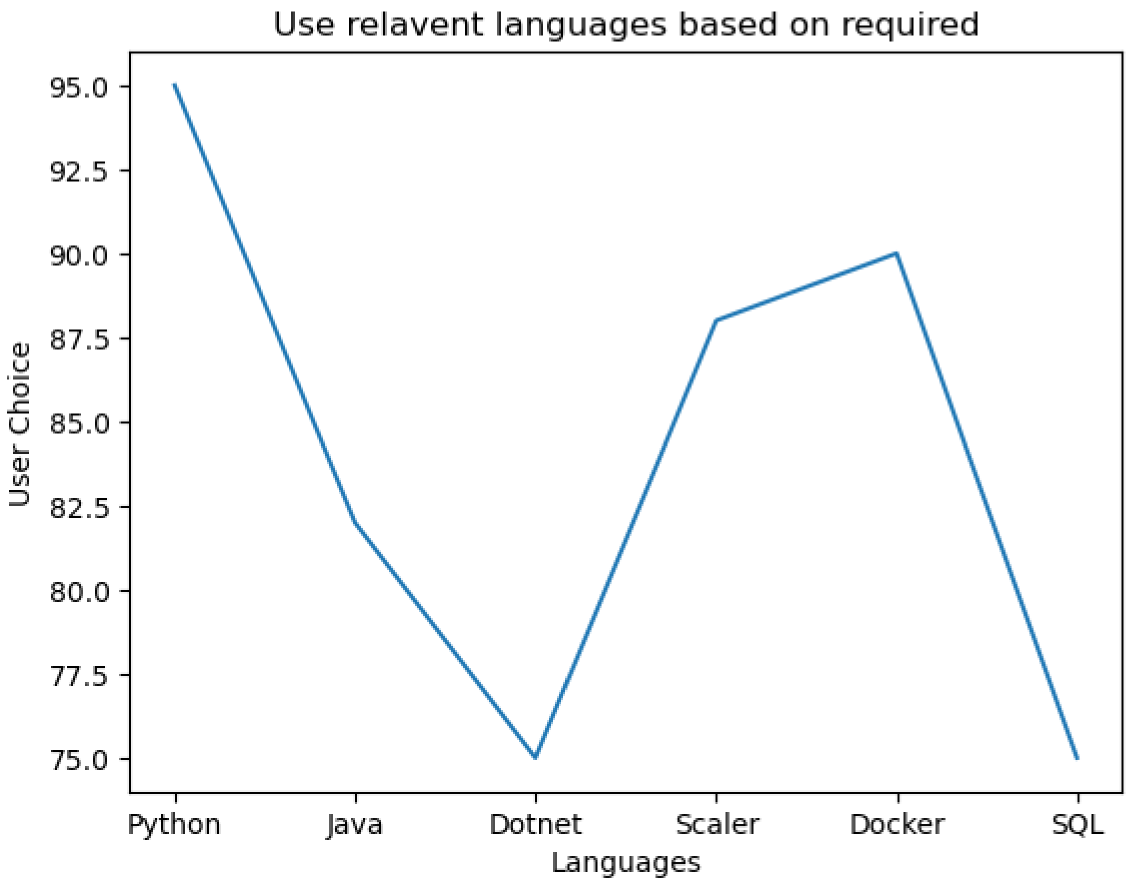
In [2]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.plot(x, y)

plt.show() # show the plot here.
```



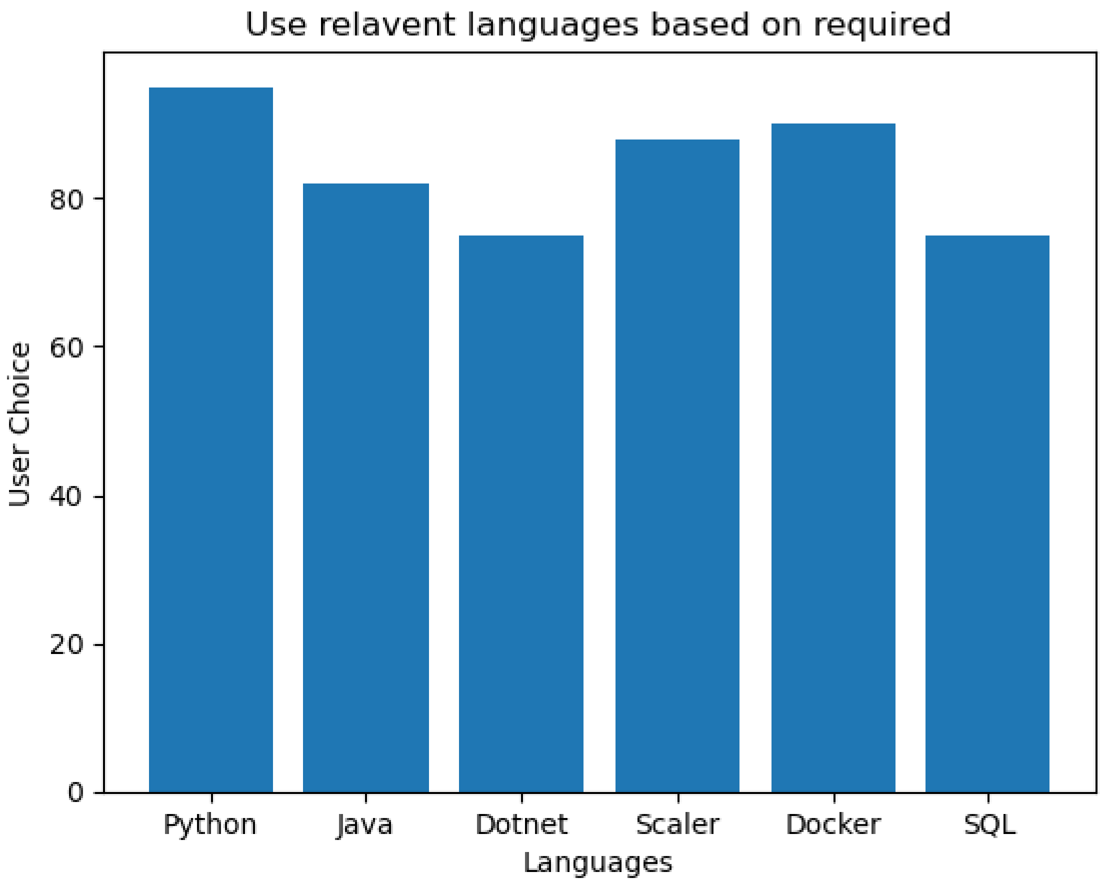
In [3]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages')
plt.ylabel('User Choice')
plt.title('Use relavent languages based on required')

plt.plot(x, y)

plt.show() # show the plot here.
```



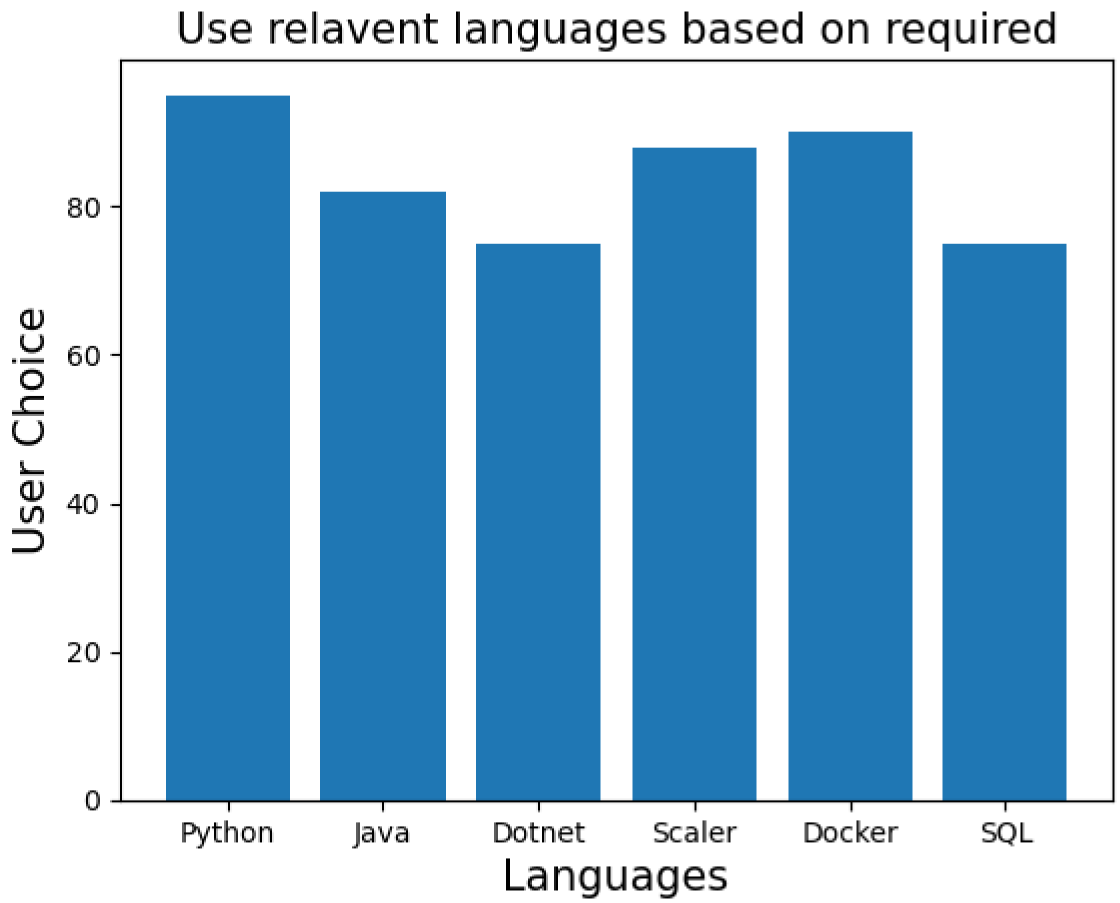
In [4]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages')
plt.ylabel('User Choice')
plt.title('Use relavent languages based on required')

plt.bar(x, y) # use this chart in bar mode.

plt.show() # show the plot here.
```



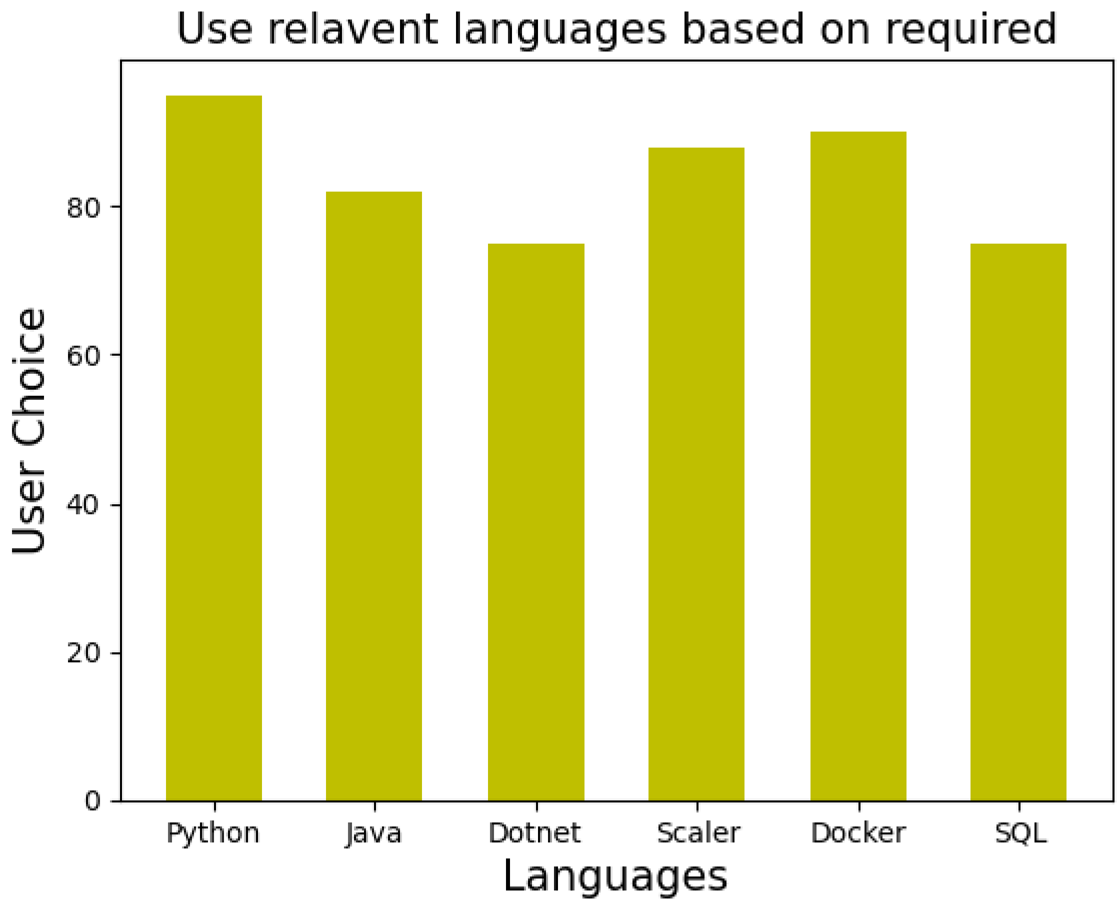
In [5]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

plt.bar(x, y) # use this chart in bar mode.

plt.show() # show the plot here.
```



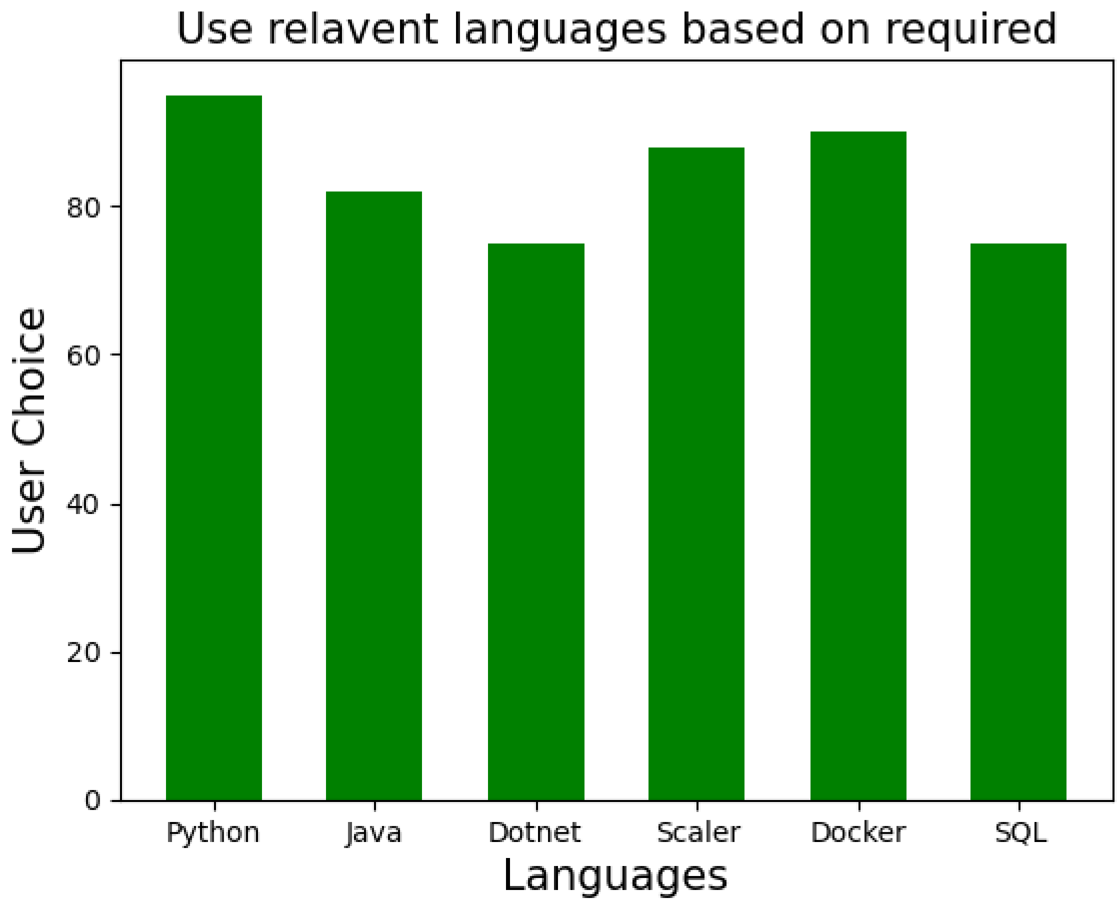
In [6]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

plt.bar(x, y, width = 0.6, color = "y") # Changes the colour of the bar.

plt.show() # show the plot here.
```



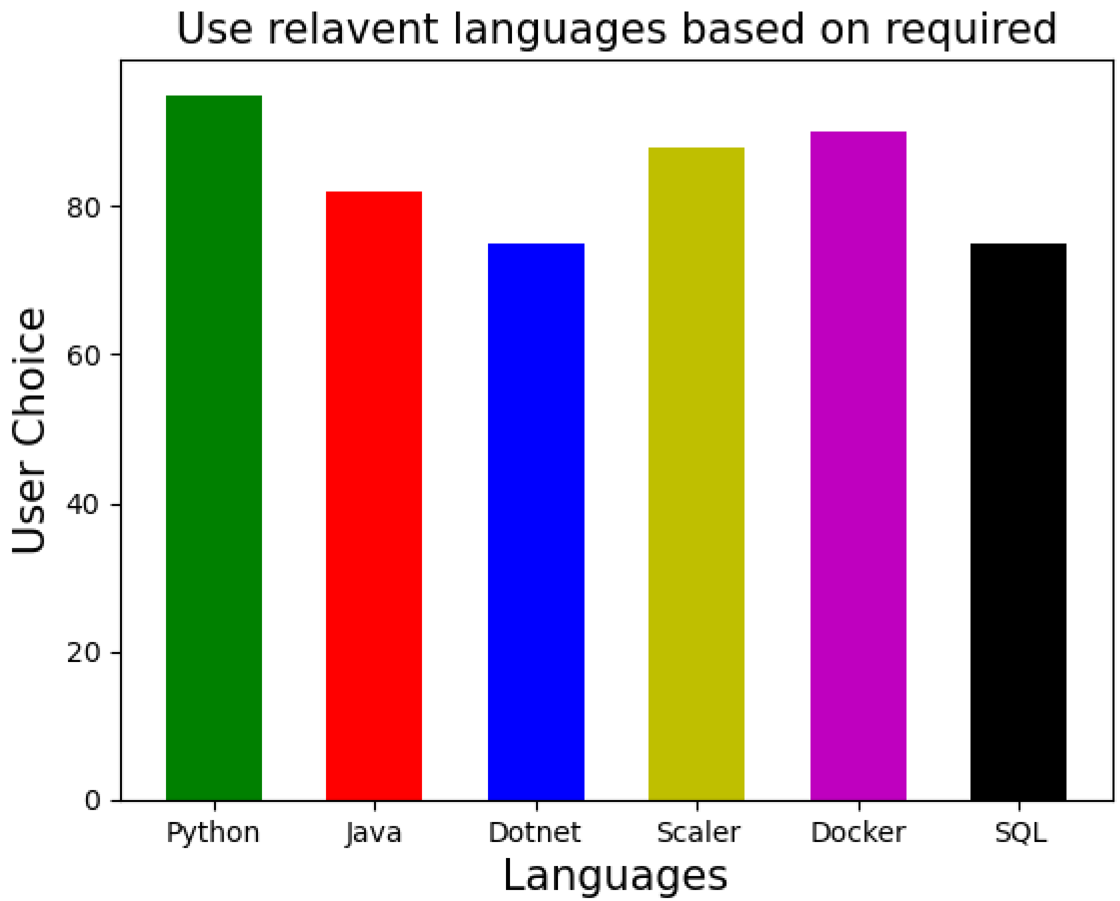
In [7]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

plt.bar(x, y, width = 0.6, color = "g") # Changes the colour of the bar.

plt.show() # show the plot here.
```



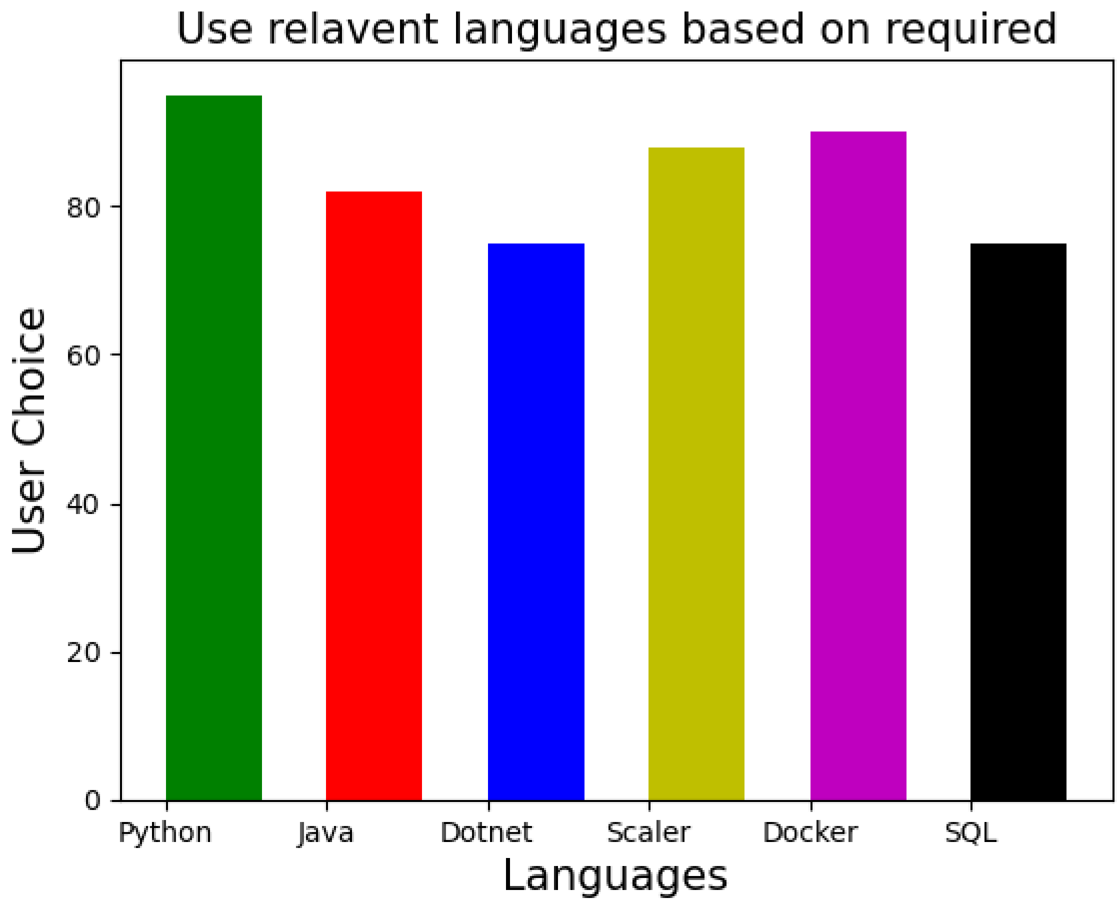
In [8]:

```python
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = c) # Changes the colour of the bar.

plt.show() # show the plot here.
```



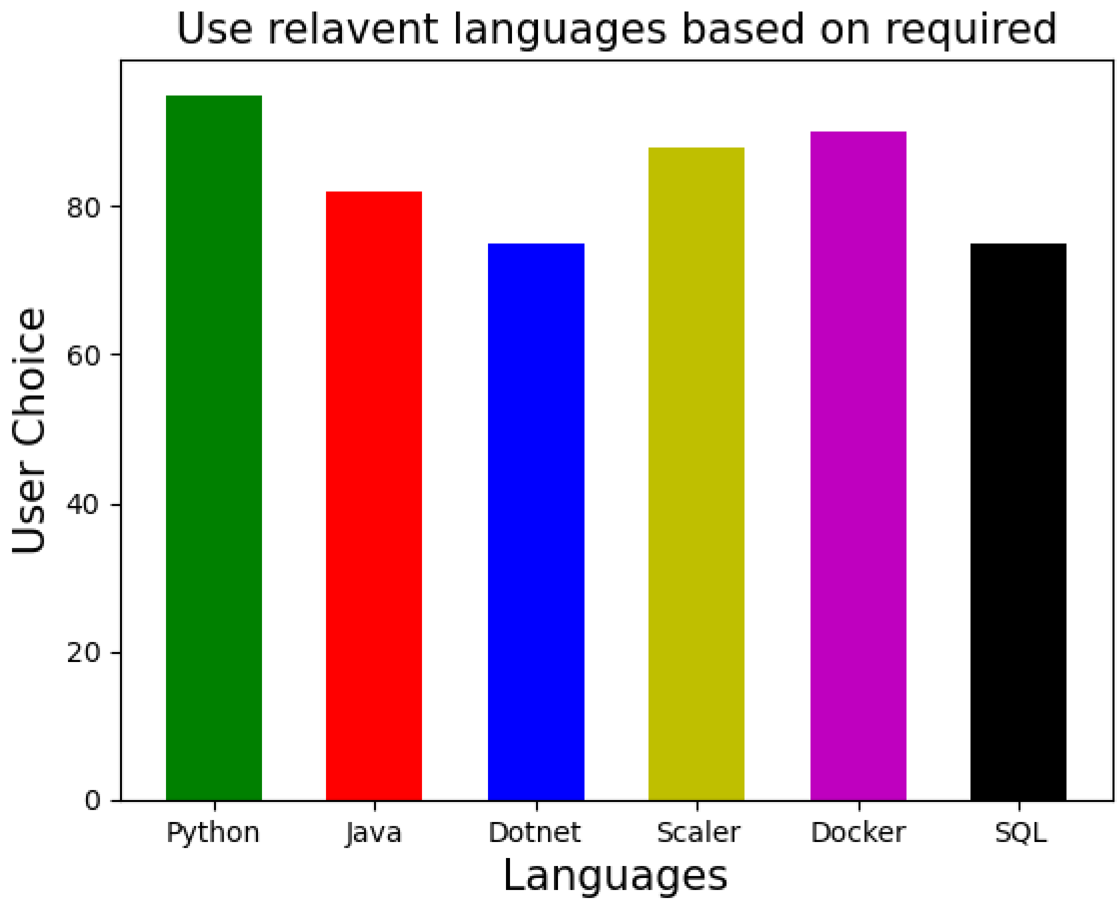
In [9]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = c, align= "edge") # If changes the edge of the languages

plt.show() # show the plot here.
```



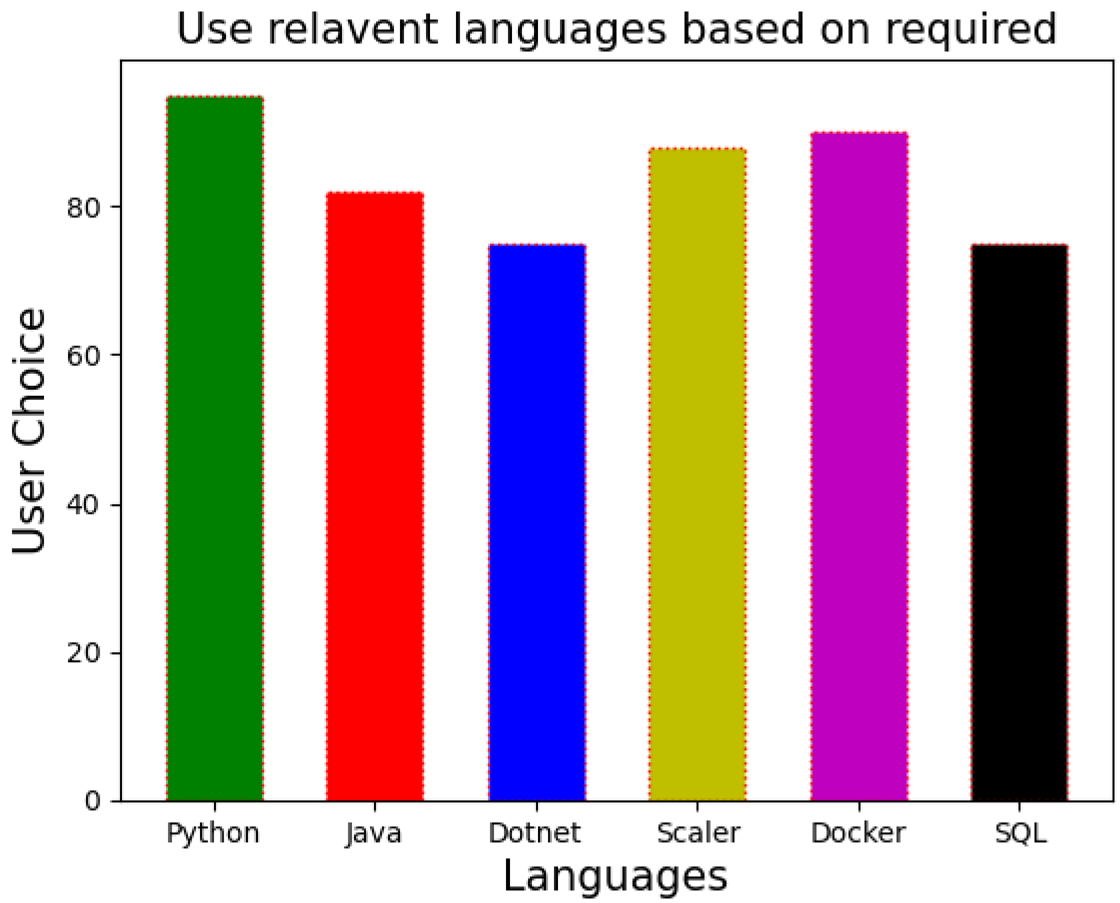
In [10]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = c) # If changes the edges of the boxs.

plt.show() # show the plot here.
```



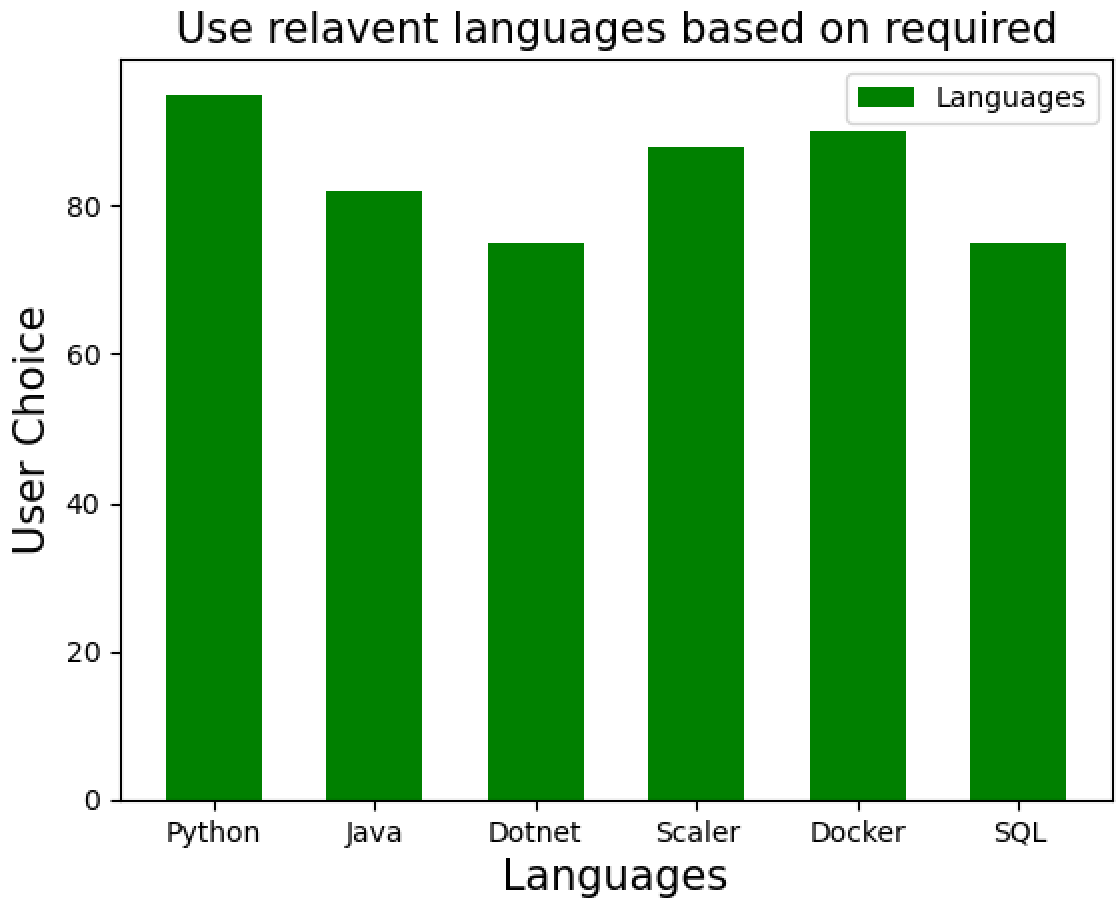
In [11]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = c, edgecolor = "r", linestyle = ":") # If put the  dotted colon of the edges.

plt.show() # show the plot here.
```



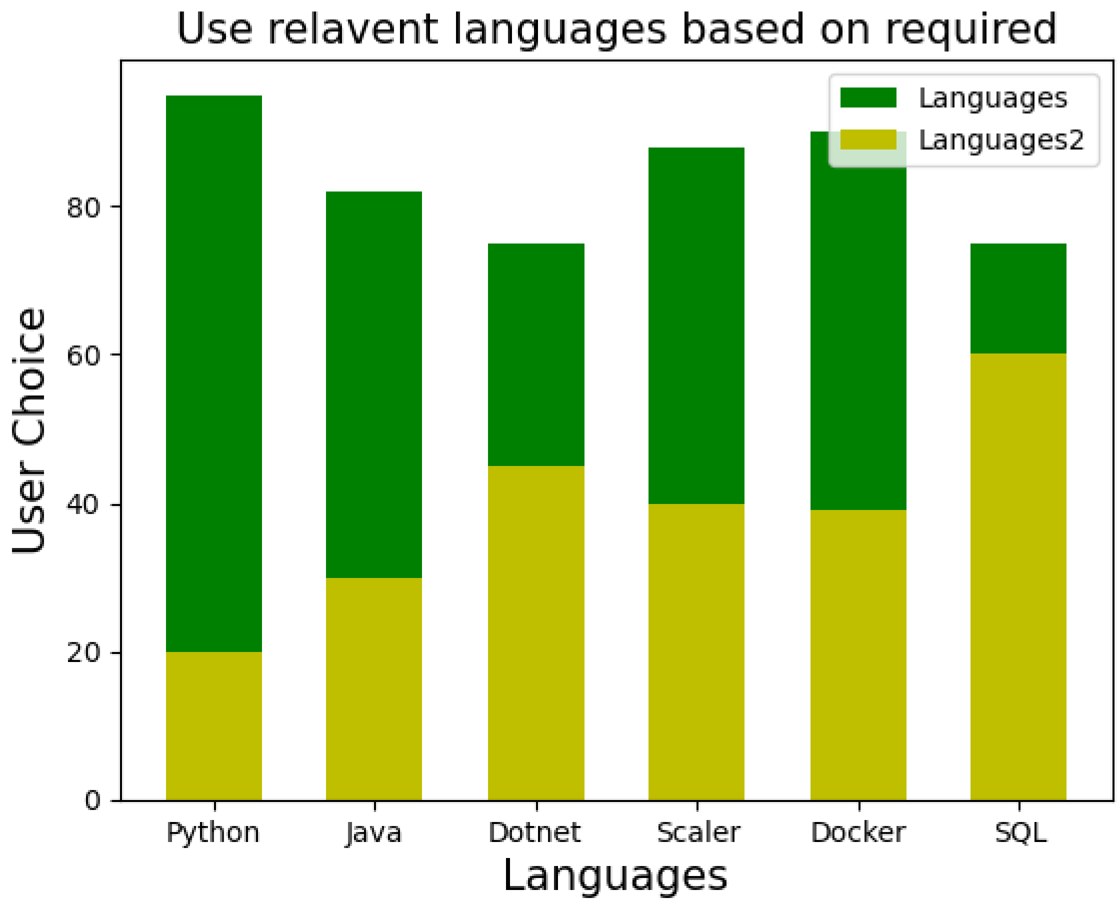
In [12]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size

c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = "g", label = "Languages") # For label

plt.legend() # For show the label
plt.show() # show the plot here.
```



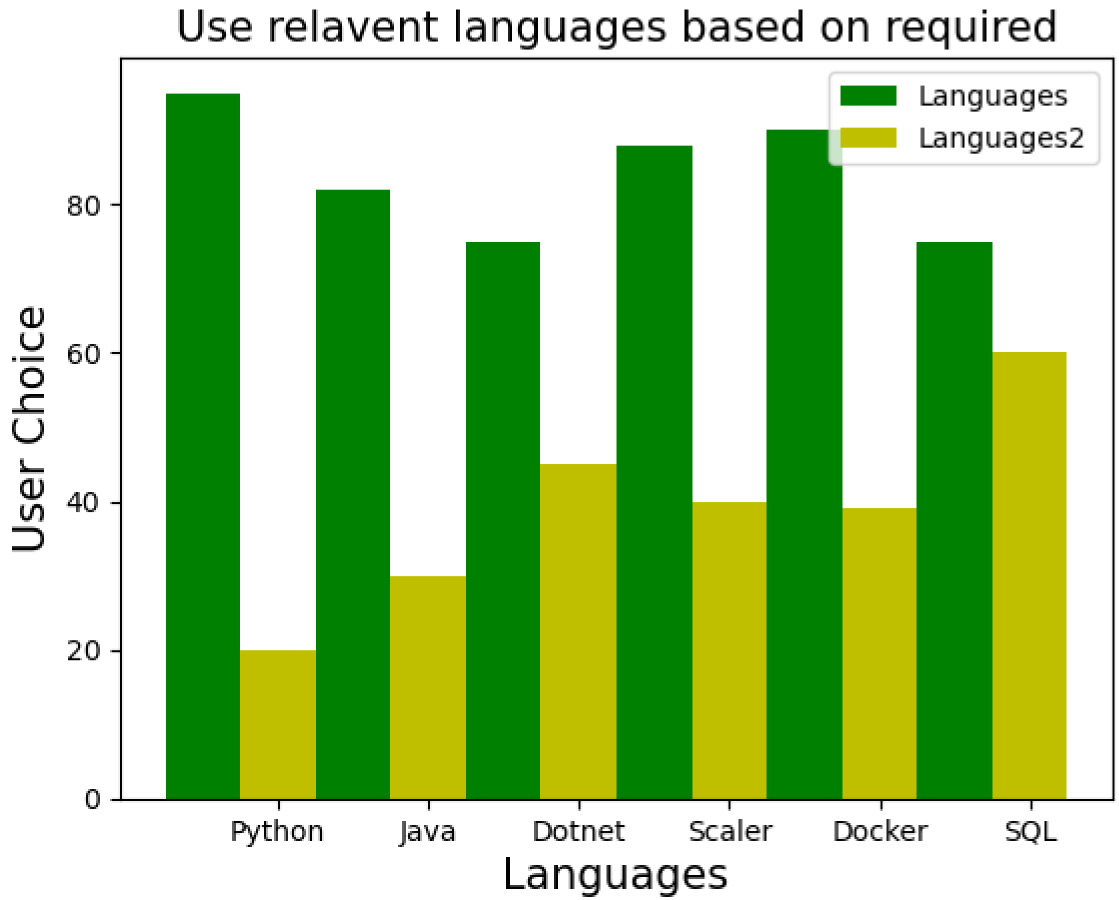
In [13]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
z = [20,30,45,40,39,60]

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size
c = ["g","r","b","y","m","k"] # changes the colour as per languages

plt.bar(x, y, width = 0.6, color = "g", label = "Languages") # For label
plt.bar(x, z, width = 0.6, color = "y", label = "Languages2") # For create multiple bar graph so use same parameter here.

plt.legend() # For show the label
plt.show() # show the plot here.
```



In [14]:

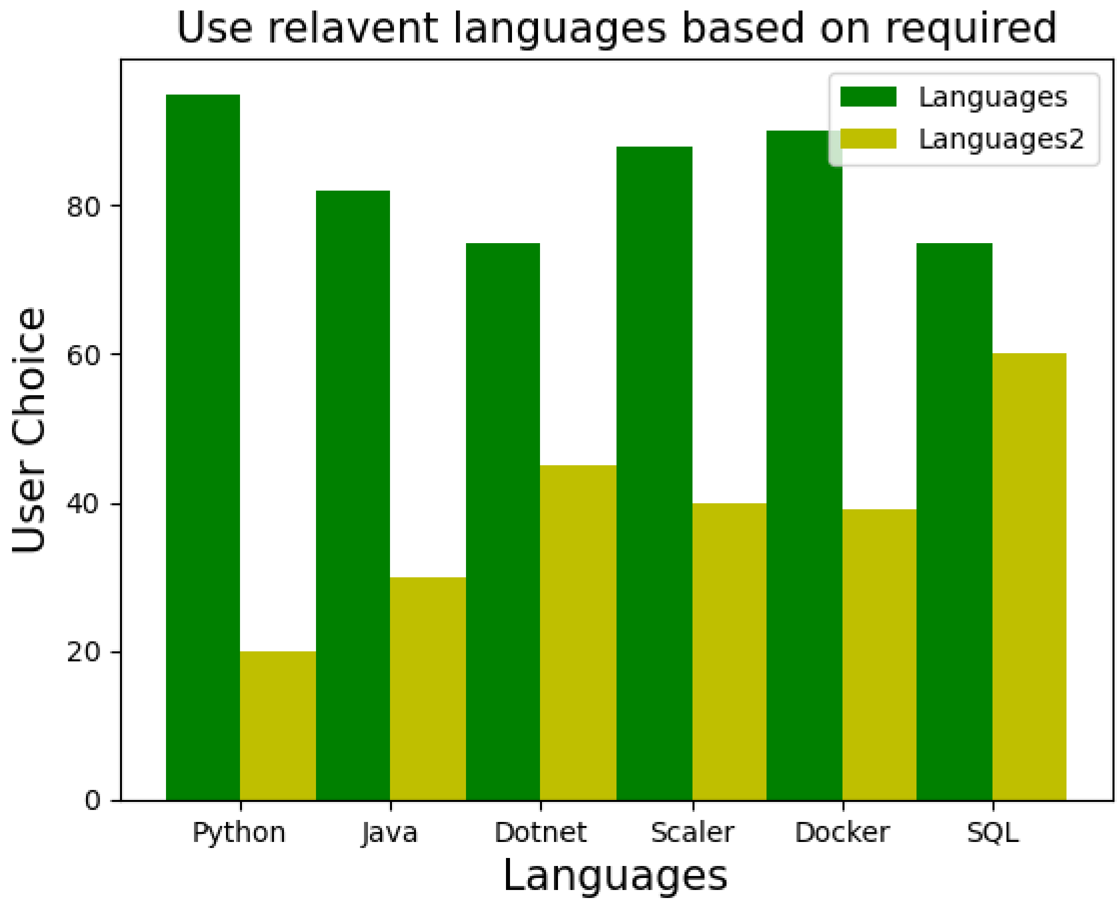
```
import numpy as np # import numpy here for array
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
z = [20,30,45,40,39,60]

width = 0.5
p = np.arange(len(x)) # For make an array in place of list
p1= [j+width for j in p] # For not overlapping the bar so need one new parametr p1 and use width for create two bars.

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size
plt.bar(p, y, width, color = "g", label = "Languages") # For label
plt.bar(p1, z, width, color = "y", label = "Languages2") # For create multiple bar graph so use same parameter here.

plt.xticks(p+width,x) # For see the original name of languages.
plt.legend() # For show the label
plt.show() # show the plot here.
```



In [15]:

```
import numpy as np # import numpy here for array
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
z = [20,30,45,40,39,60]

width = 0.5
p = np.arange(len(x)) # For make an array in place of list
```

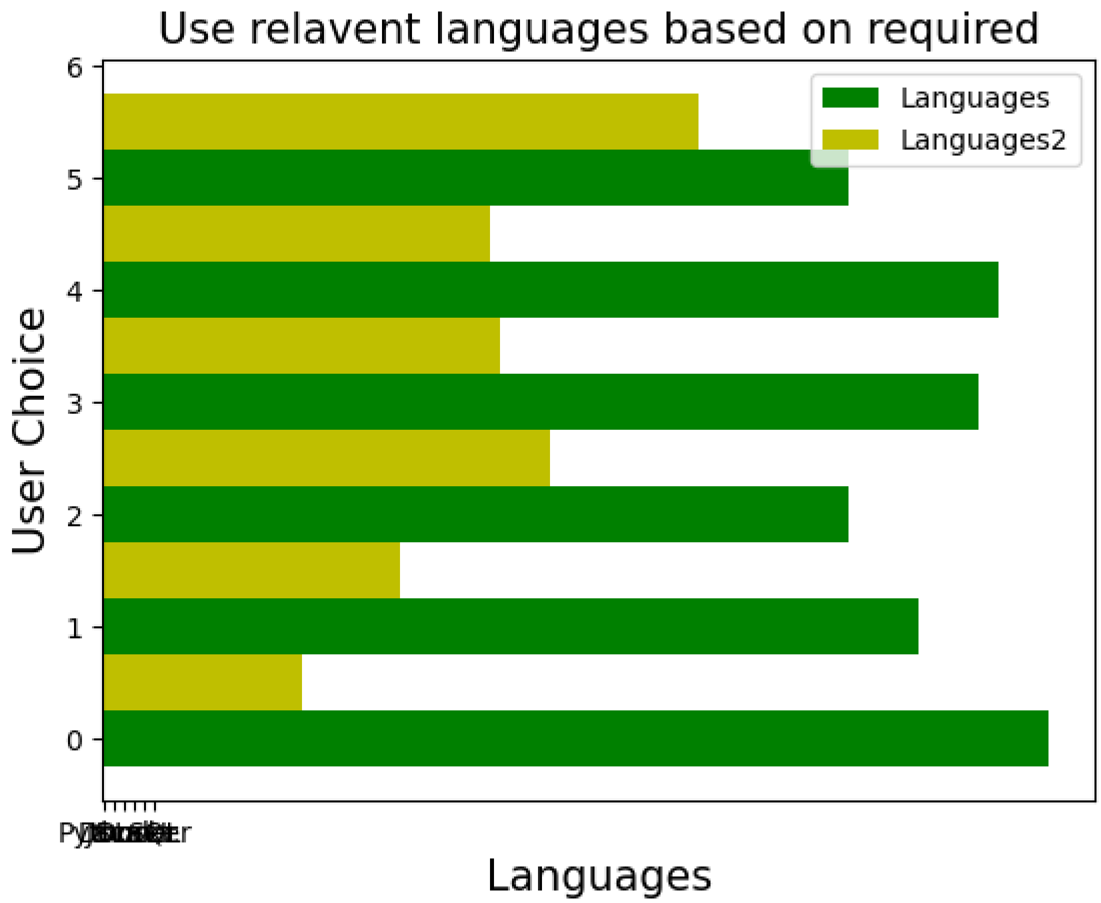
```
p1= [j+width for j in p] # For not overlapping the bar so need one new parametr p1 and use width for create two bars.

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size


plt.bar(p, y, width, color = "g", label = "Languages") # For label
plt.bar(p1, z, width, color = "y", label = "Languages2") # For create multiple bar graph so use same parameter here.

plt.xticks(p+width/2,x) # For see the original name of languages but this in in mid of yellow colour so fit in actual in center.
plt.legend() # For show the label
plt.show() # show the plot here.
```



For Horizontal bar graph

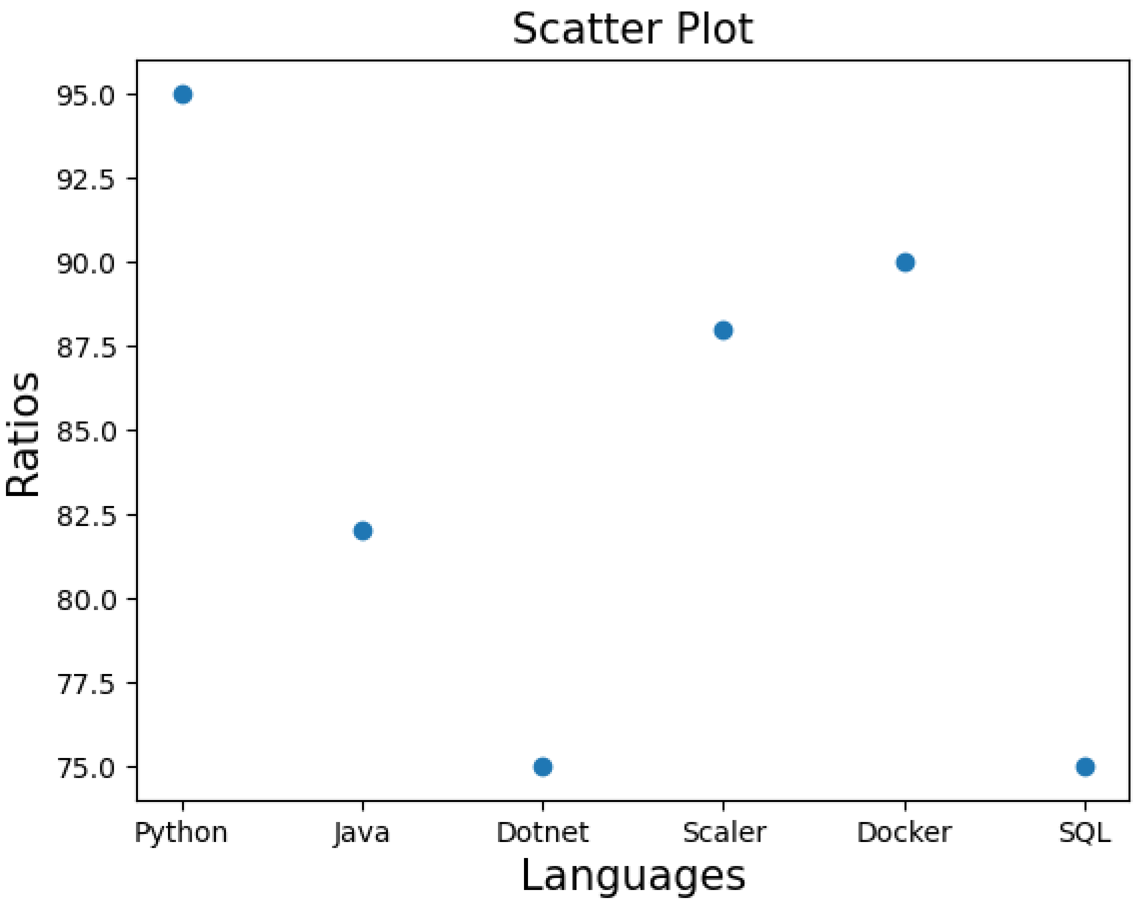
In [16]:

```
import numpy as np # import numpy here for array
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
z = [20,30,45,40,39,60]

width = 0.5
p = np.arange(len(x)) # For make an array in place of list
p1= [j+width for j in p] # For not overlapping the bar so need one new parametr p1 and use width for create two bars.

plt.xlabel('Languages', fontsize = 15) # Changes the font size
plt.ylabel('User Choice',fontsize = 15) # Changes the font size
plt.title('Use relavent languages based on required',fontsize = 15) # Changes the font size


plt.barh(p, y, width, color = "g", label = "Languages") # For adding h here
plt.barh(p1, z, width, color = "y", label = "Languages2") # For create multiple bar graph so use same parameter here.

plt.xticks(p+width/2,x) # For see the original name of languages but this in in mid of yellow colour so fit in actual in center.
plt.legend() # For show the label
plt.show() # show the plot here.
```



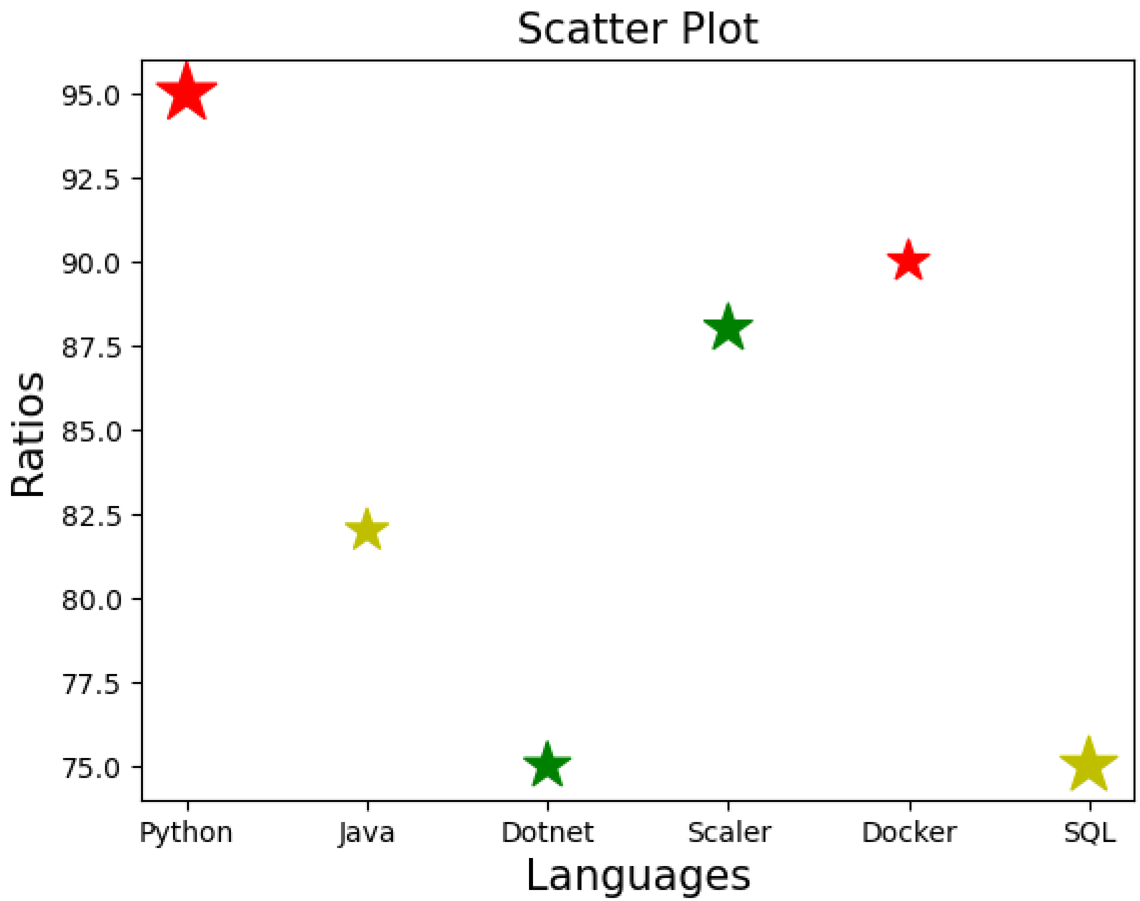
Scatter Plot

```
In [17]: import matplotlib.pyplot as plt
         x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
         y = [95, 82, 75, 88, 90, 75]
         plt.scatter(x,y)
         plt.title("Scatter Plot", fontsize = 15)
         plt.xlabel("Languages",fontsize = 15)
         plt.ylabel("Ratios",fontsize = 15)
         plt.show()
```



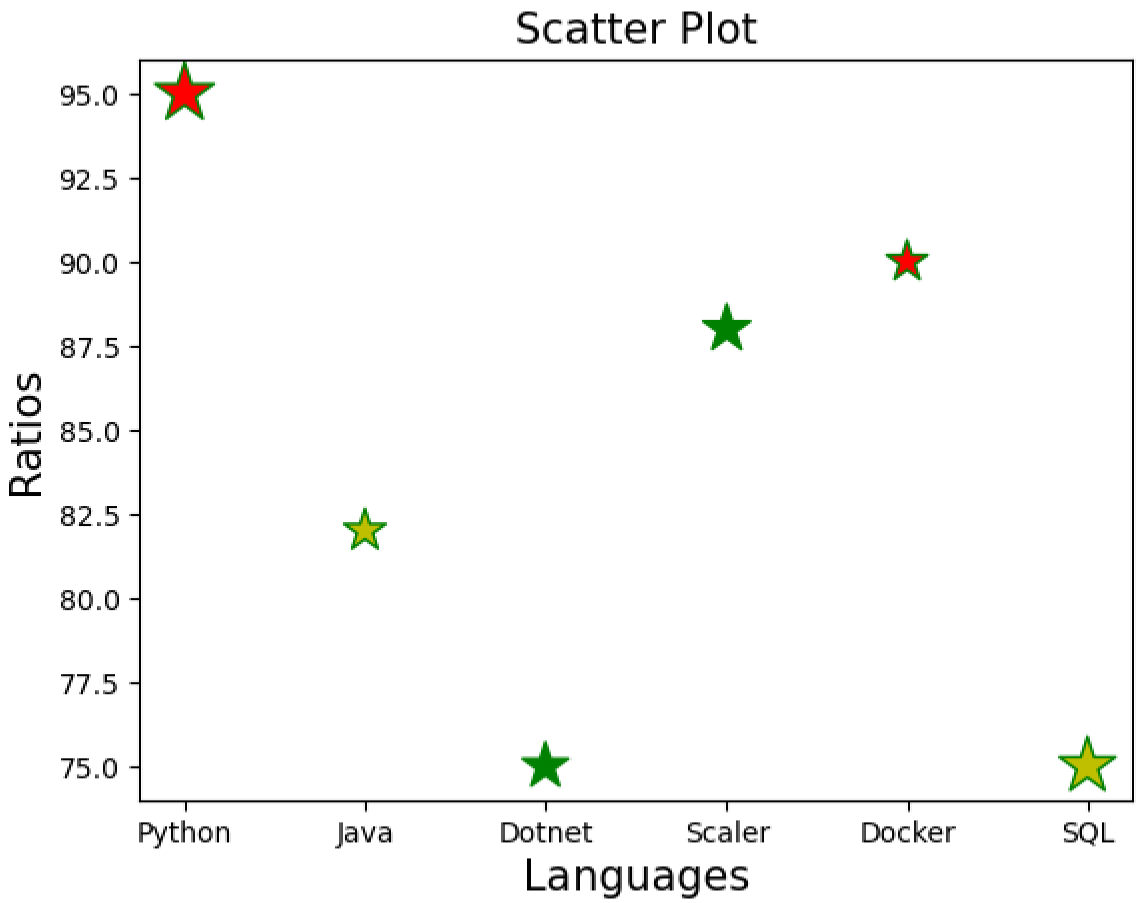
In [18]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
colors = ["r","y", "g", "g", "r", "y"] # For shows as different colour
size = [500,250, 300,320,240,450]
plt.scatter(x,y, c = colors,s = size, marker = "*") # For show the star shape here as marker
plt.title("Scatter Plot", fontsize = 15)
plt.xlabel("Languages",fontsize = 15)
plt.ylabel("Ratios",fontsize = 15)
plt.show()
```

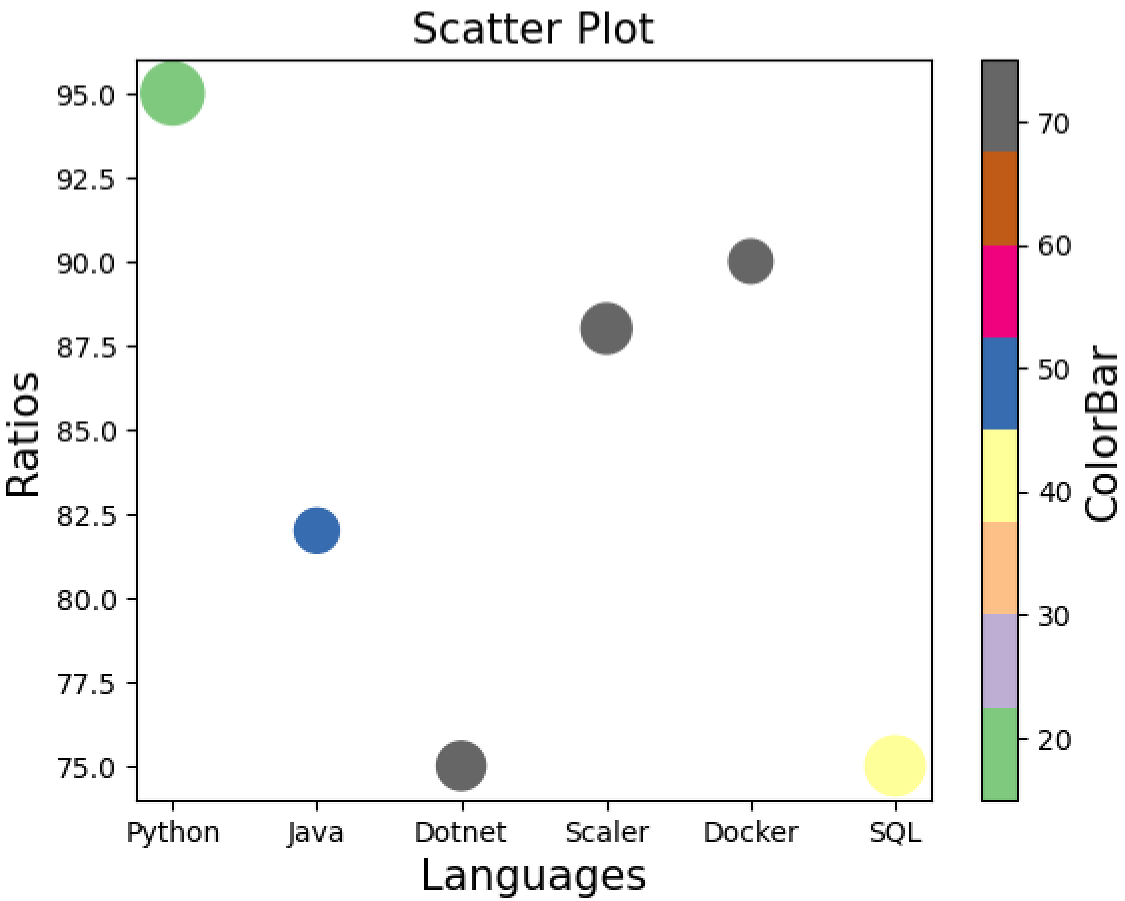


In [19]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
colors = ["r","y", "g", "g", "r", "y"] # For shows as different colour
size = [500,250, 300,320,240,450]
plt.scatter(x,y, c = colors,s = size, marker = "*", edgecolor = "g") # For show the star shape here as marker
plt.title("Scatter Plot", fontsize = 15)
plt.xlabel("Languages",fontsize = 15)
plt.ylabel("Ratios",fontsize = 15)
plt.show()
```

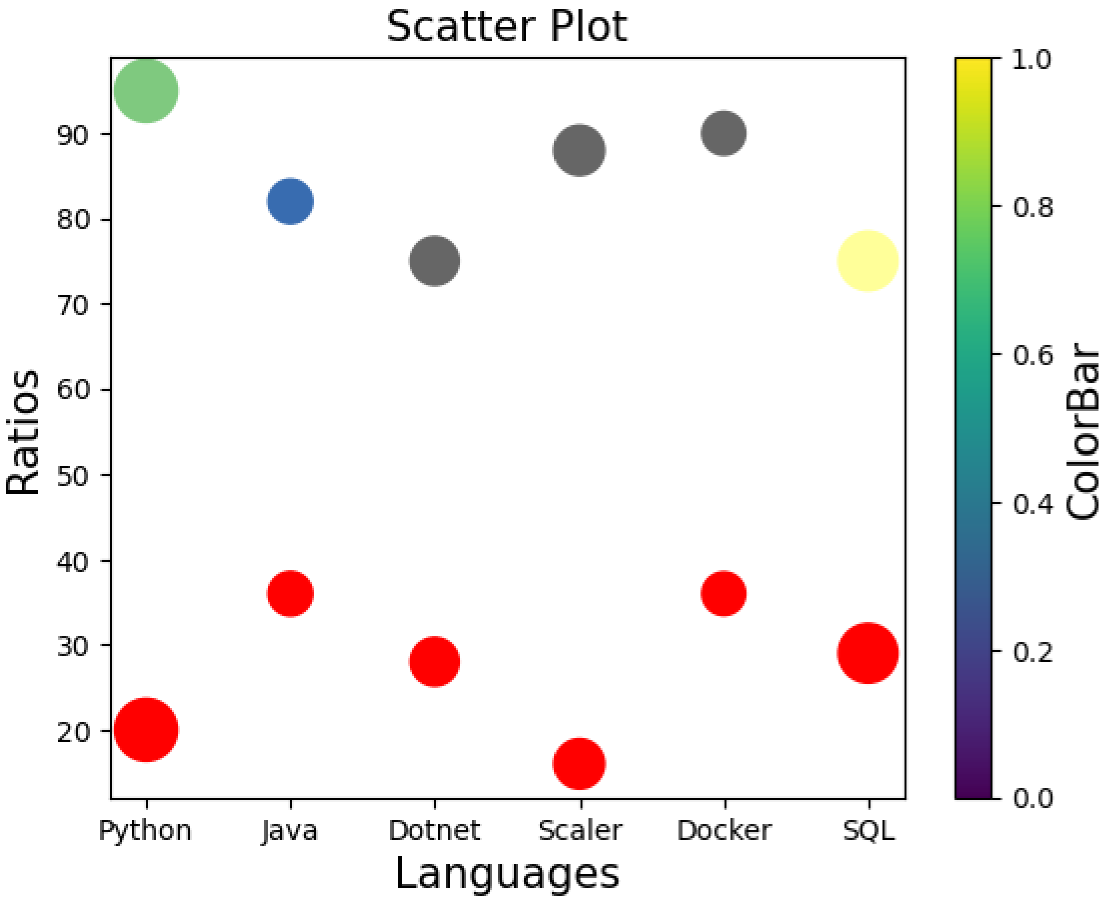


```
In [20]: x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
         y = [95, 82, 75, 88, 90, 75]
         colors = [15, 49, 68, 75, 69, 39] # For shows as different colour
         size = [500,250, 300,320,240,450]
         plt.scatter(x,y, c = colors,s = size, cmap = "Accent") # For use the different color as per choices.
         t = plt.colorbar() # For check the color as per number in a ranges
         t.set_label("ColorBar", fontsize = 15)
         plt.title("Scatter Plot", fontsize = 15)
         plt.xlabel("Languages",fontsize = 15)
         plt.ylabel("Ratios",fontsize = 15)
         plt.show()
```



In [21]:

```
x = ["Python", "Java", "Dotnet", "Scaler", "Docker", "SQL"]
y = [95, 82, 75, 88, 90, 75]
z = [20, 36, 28, 16, 36, 29]

colors = [15, 49, 68, 75, 69, 39] # For shows as different colour
size = [500,250, 300,320,240,450]

plt.scatter(x,y, c = colors,s = size, cmap = "Accent")
plt.scatter(x,z, color = "r",s = size)# For use the different color as per choices.

t = plt.colorbar() # For check the color as per number in a ranges
t.set_label("ColorBar", fontsize = 15)

plt.title("Scatter Plot", fontsize = 15)
plt.xlabel("Languages",fontsize = 15)
plt.ylabel("Ratios",fontsize = 15)
plt.show()
```



Matplotlib Histogram Plot

In [23]:

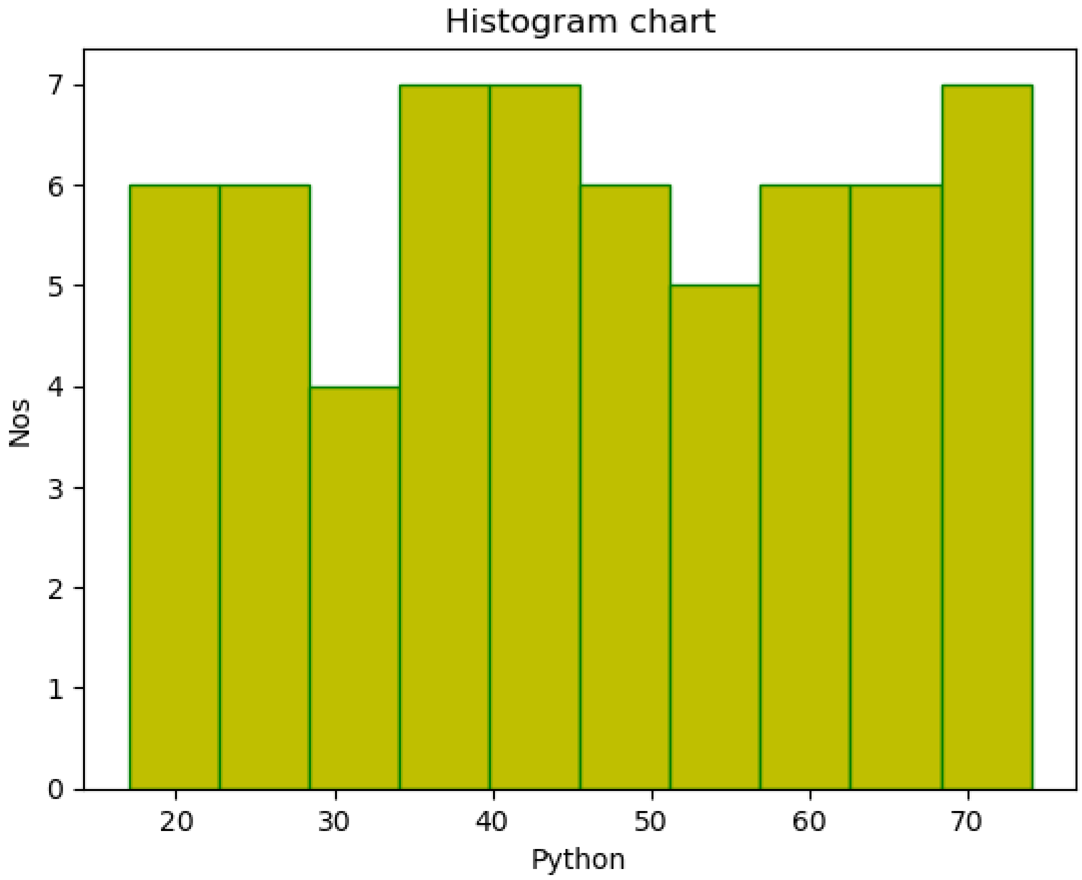
```
import matplotlib.pyplot as plt
import numpy as np
import random
```

```
x = np.random.randint(15,75,(70)) # Need to show 75 data from 15 to till 75 and use here random function
print(x)
```

```
[66 65 62 69 47 36 42 64 21 20 28 38 22 42 27 28 53 50 45 39 47 37 23 61
 74 44 32 70 53 59 42 21 52 50 68 29 45 47 52 57 37 61 17 37 50 26 32 69
 17 64 33 37 72 71 66 53 44 74 60 25 49 59 49 34 19 57 48 28 42 30]
```

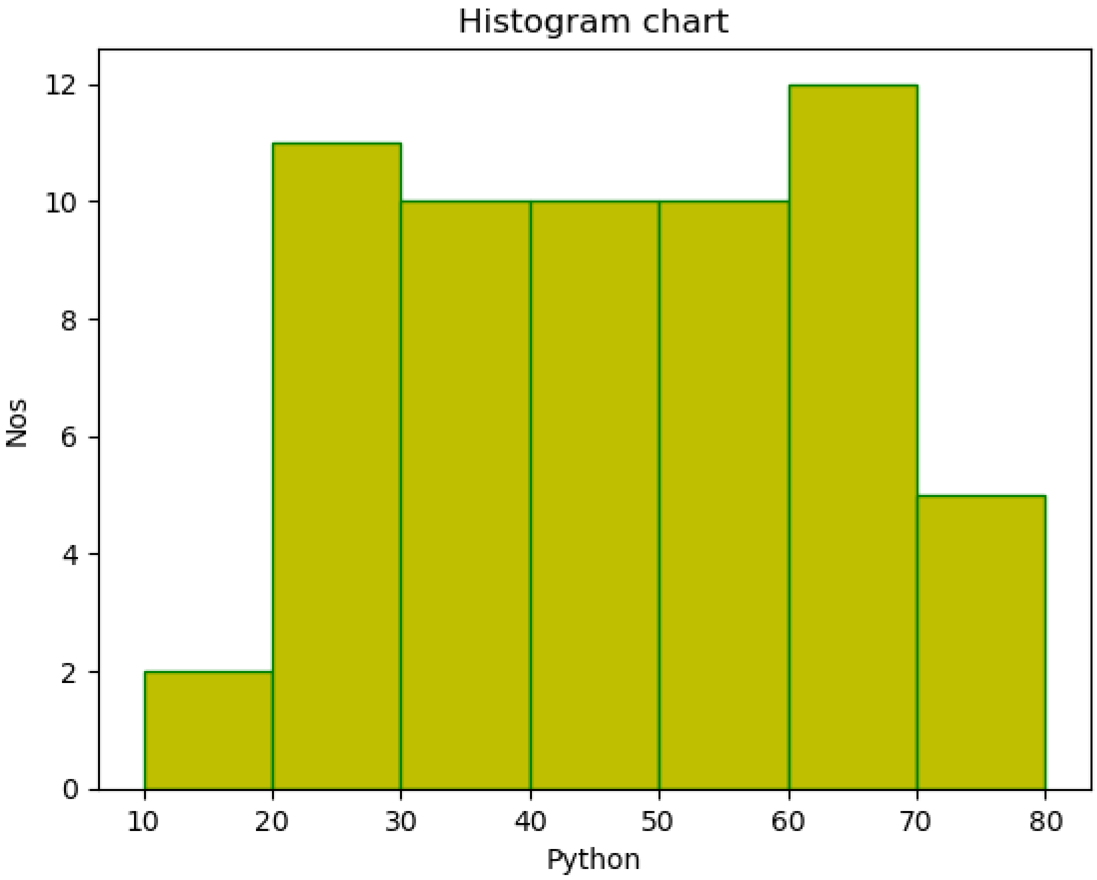
In [31]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]

plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")

plt.hist(number, color = "y", edgecolor = "g") # For changes color and use edge colour for clearly show the color  ranges
plt.show()
```



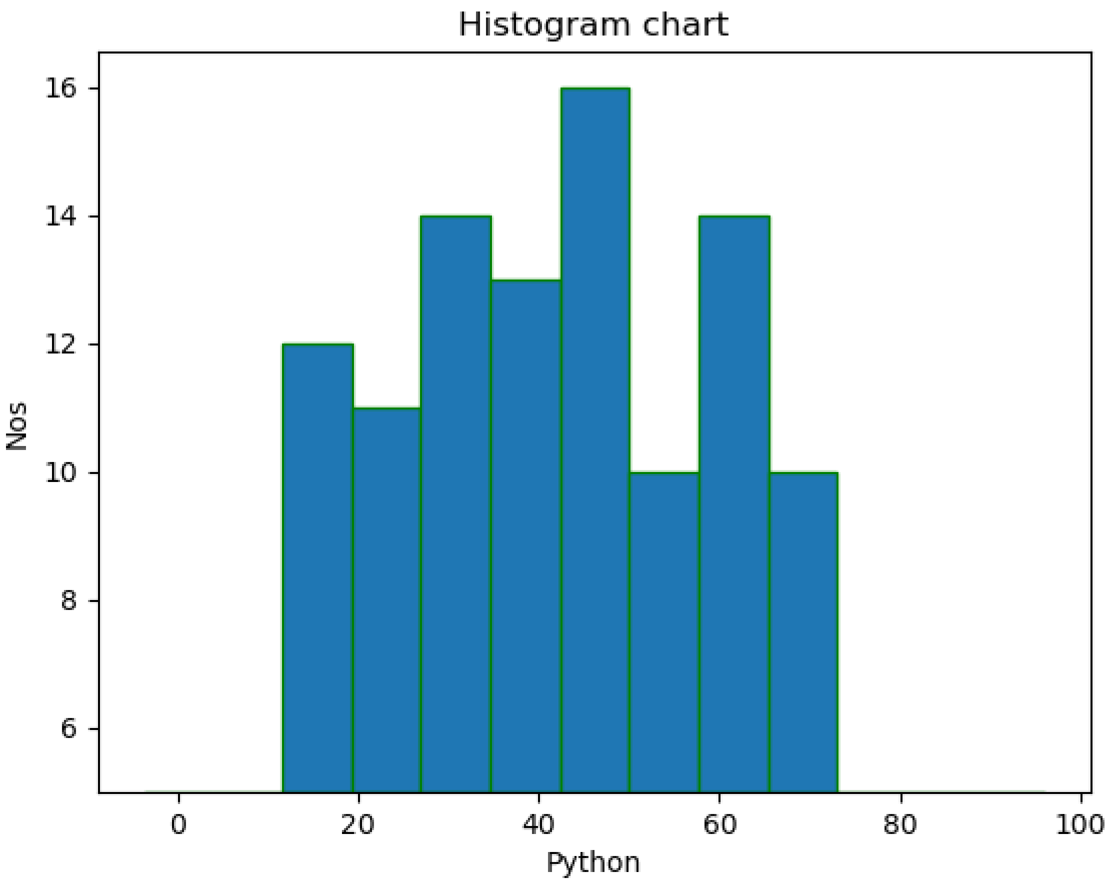
In [44]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]
l = [10,20,30,40,50,60,70,80]
plt.hist(number, color = "y", edgecolor = "g", bins = l) # For changes color and use edge colour for clearly show the color  rang

plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")


plt.show()
```



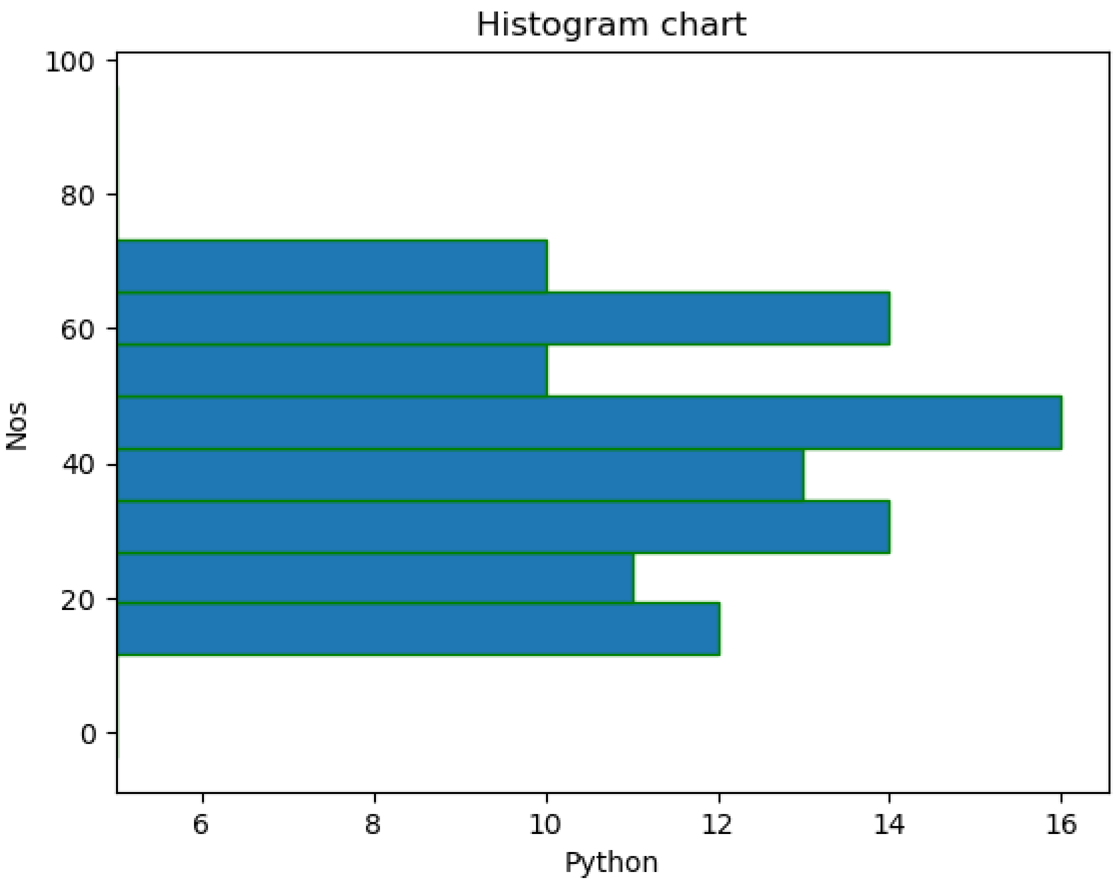
In [37]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]
l = [10,20,30,40,50,60,70,80]
plt.hist(number, "auto", (0,100), edgecolor = "g", bottom = 5, align = "left") # Use bottom parameter here

plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")


plt.show()
```



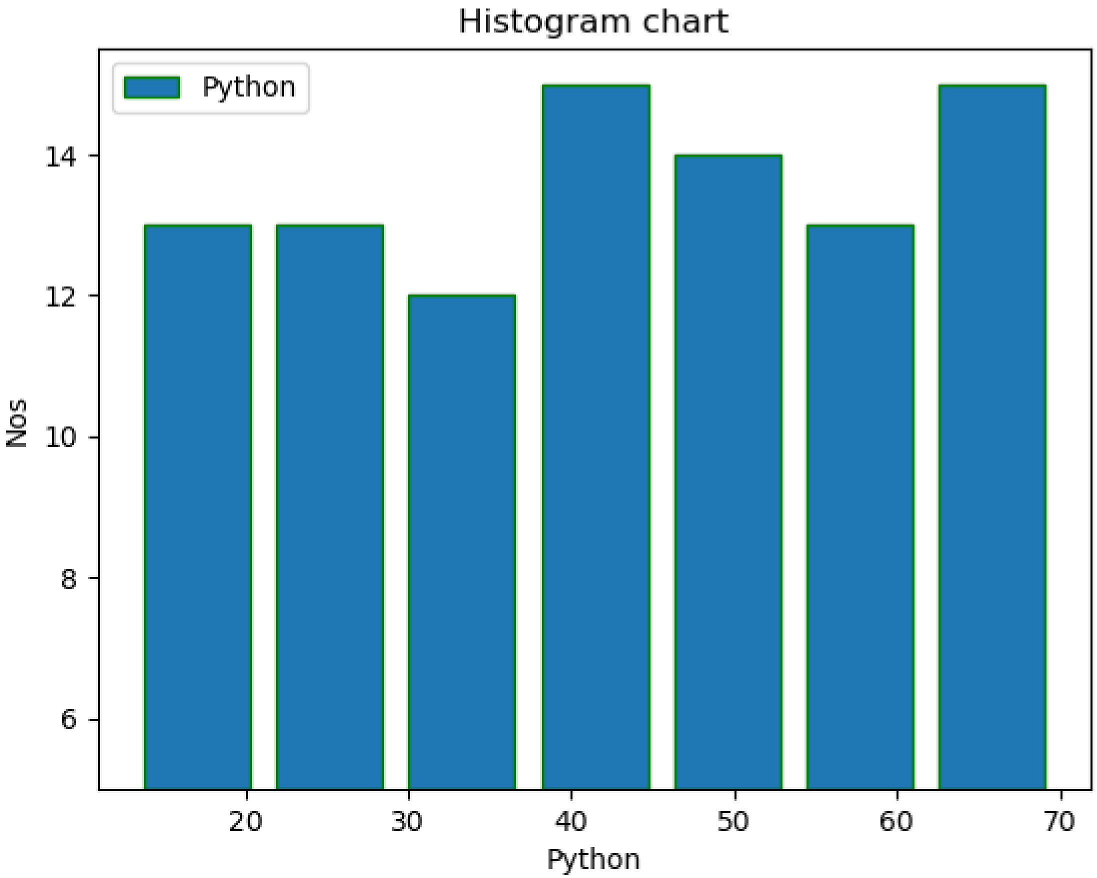
In [38]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]
l = [10,20,30,40,50,60,70,80]
plt.hist(number, "auto", (0,100), edgecolor = "g", bottom = 5, align = "left", orientation = "horizontal") # Use horizontal shape
plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")


plt.show()
```



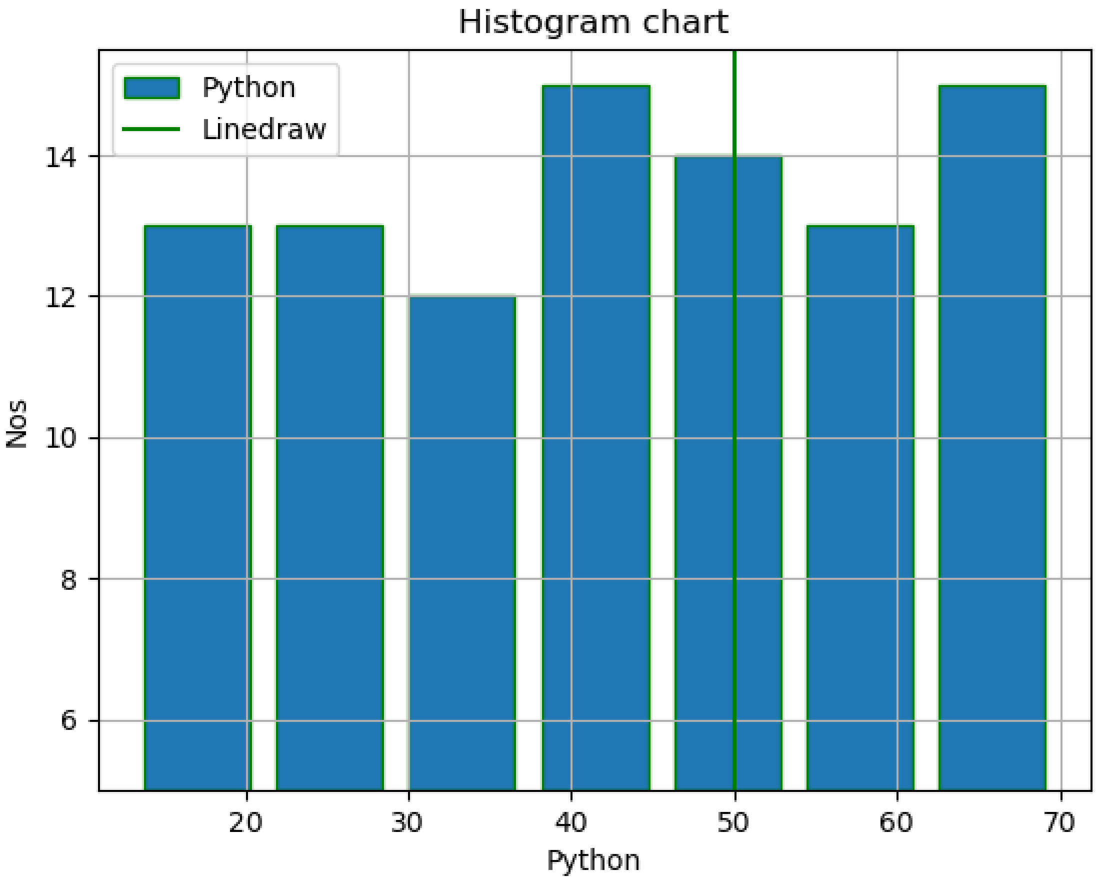
In [45]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]
l = [10,20,30,40,50,60,70,80]
plt.hist(number, "auto", edgecolor = "g", bottom = 5, align = "left", rwidth = 0.8, label = "Python") # If we changes the width c

plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")
plt.legend()


plt.show()
```



In [48]:

```
number = [66, 65, 62, 69, 47, 36, 42, 64, 21, 20, 28, 38, 22, 42, 27, 28, 53, 50, 45, 39, 47, 37, 23, 61,
 74, 44, 32, 70, 53, 59, 42, 21, 52, 50,68, 29, 45, 47, 52,57, 37, 61, 17, 37, 50, 26, 32, 69,
 17, 64, 33, 37, 72, 71, 66, 53, 44, 74, 60, 25]
l = [10,20,30,40,50,60,70,80]

plt.hist(number, "auto", edgecolor = "g", bottom = 5, align = "left", rwidth = 0.8, label = "Python") # If we changes the width c

plt.axvline(50, color = "g", label = "Linedraw")
plt.title("Histogram chart")
plt.xlabel("Python")
plt.ylabel("Nos")
plt.legend()

plt.grid()
plt.show()
```


Histogram chart
Python
Linedraw
Nos
14
12
10
8
6
20
30
40
50
60
70
Python

In [ ]: