

In [1]:

```
def add(num1, num2):  
    return num1 + num2  
  
def minus(num1, num2):  
    return num1 - num2  
  
def multiply(num1, num2):  
    return num1 * num2  
  
def div(num1, num2):  
    return num1 / num2  
  
def divint(num1, num2):  
    return num1 // num2
```

In [2]:

```
def calc(num1, num2, op):  
    if op == '+':  
        return add(num1, num2)  
    elif op == '-':  
        return minus(num1, num2)  
    elif op == '*':  
        return multiply(num1, num2)  
    elif op == '/':  
        return div(num1, num2)  
    elif op == '//':  
        return divint(num1, num2)
```

In [3]:

```
print(calc(6, 13, '+'))  
print(calc(6, 13, '-'))  
print(calc(6, 13, '*'))  
print(calc(6, 13, '/'))  
print(calc(6, 13, '//'))
```

```
19  
-7  
78  
0.46153846153846156  
0
```

In [5]:

```
import numpy as np  
  
a = np.array([1, 2, 3])  
b = 2  
  
result = a * b  
print(result)
```

```
[2 4 6]
```

In [6]:

```
a = np.array([[1, 2, 3], [4, 5, 6]])  
b = np.array([1, 2, 3])  
  
result = a + b  
print(result)
```

```
[[2 4 6]  
 [5 7 9]]
```

In [7]:

```
numbers = list(range(1, 101))

multiples_of_three = [x for x in numbers if x % 3 == 0]
print(multiples_of_three)
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]
```

In [11]:

```
def toList(str):
    print("toList Called!!")
    print("    my ret is [{0}]", list(str))
    return list(str)

def makeDict(elem):
    print("makeDict Called!!")
    print("    my ret is [{0}]", [elem[0], 1])
    return [elem[0], 1]

def mapping(lst):
    print("mapping Called!!")
    print("    my lst is [{0}]", list(map(makeDict, lst)))
    return list(map(makeDict, lst))

def shuffle(mapped_result):
    shuffled_result = {}
    for key, val in mapped_result:
        shuffled_result[key] = shuffled_result.get(key, []) + [val]

    print("shuffle Called!!")
    print("    my ret is [{0}]", shuffled_result)

    return shuffled_result

def reduce(shuffled_data):
    ret = {}
    for key in shuffled_data.keys():
        ret[key] = sum(shuffled_data[key])

    print("reduce Called!!")
    print("    my ret is [{0}]", ret)

    return ret;

def repeatprint(reduced_data):
    var = ""

    for key, value in reduced_data.items():
        var += (key + str(value))
        #try var += (key * value)
    return var

repeatprint(reduce(shuffle(mapping(toList("todayweather")))))
```

```
toList Called!!
    my ret is [{0}] ['t', 'o', 'd', 'a', 'y', 'w', 'e', 'a', 't', 'h', 'e', 'r']
mapping Called!!
makeDict Called!!
    my ret is [{0}] ['t', 1]
makeDict Called!!
    my ret is [{0}] ['o', 1]
makeDict Called!!
    my ret is [{0}] ['d', 1]
makeDict Called!!
```

```

    my ret is [{0}] ['a', 1]
makeDict Called!!
    my ret is [{0}] ['y', 1]
makeDict Called!!
    my ret is [{0}] ['w', 1]
makeDict Called!!
    my ret is [{0}] ['e', 1]
makeDict Called!!
    my ret is [{0}] ['a', 1]
makeDict Called!!
    my ret is [{0}] ['t', 1]
makeDict Called!!
    my ret is [{0}] ['h', 1]
makeDict Called!!
    my ret is [{0}] ['e', 1]
makeDict Called!!
    my ret is [{0}] ['r', 1]
    my lst is [{0}] [['t', 1], ['o', 1], ['d', 1], ['a', 1], ['y', 1], ['w', 1], ['e',
1], ['a', 1], ['t', 1], ['h', 1], ['e', 1], ['r', 1]]
makeDict Called!!
    my ret is [{0}] ['t', 1]
makeDict Called!!
    my ret is [{0}] ['o', 1]
makeDict Called!!
    my ret is [{0}] ['d', 1]
makeDict Called!!
    my ret is [{0}] ['a', 1]
makeDict Called!!
    my ret is [{0}] ['y', 1]
makeDict Called!!
    my ret is [{0}] ['w', 1]
makeDict Called!!
    my ret is [{0}] ['e', 1]
makeDict Called!!
    my ret is [{0}] ['a', 1]
makeDict Called!!
    my ret is [{0}] ['t', 1]
makeDict Called!!
    my ret is [{0}] ['h', 1]
makeDict Called!!
    my ret is [{0}] ['e', 1]
makeDict Called!!
    my ret is [{0}] ['r', 1]
shuffle Called!!
    my ret is [{0}] {'t': [1, 1], 'o': [1], 'd': [1], 'a': [1, 1], 'y': [1], 'w': [1],
'e': [1, 1], 'h': [1], 'r': [1]}
reduce Called!!
    my ret is [{0}] {'t': 2, 'o': 1, 'd': 1, 'a': 2, 'y': 1, 'w': 1, 'e': 2, 'h': 1,
'r': 1}

```

Out[11]: 't2o1d1a2y1w1e2h1r1'

```

In [12]: my_array = np.arange(1000000)
my_list = list(range(1000000))

print("np array : ")
%time for _ in range(1000): my_array2 = my_array * 2
print("list : ")
%time for _ in range(1000): my_list2 = my_list * 2

np array :
Wall time: 3.91 s
list :
Wall time: 39.8 s

```

```

In [14]: import matplotlib.pyplot as plt

```

```

labels = ['G1', 'G2', 'G3', 'G4', 'G5']
men_means = [20, 34, 30, 35, 27]
women_means = [25, 32, 34, 20, 25]

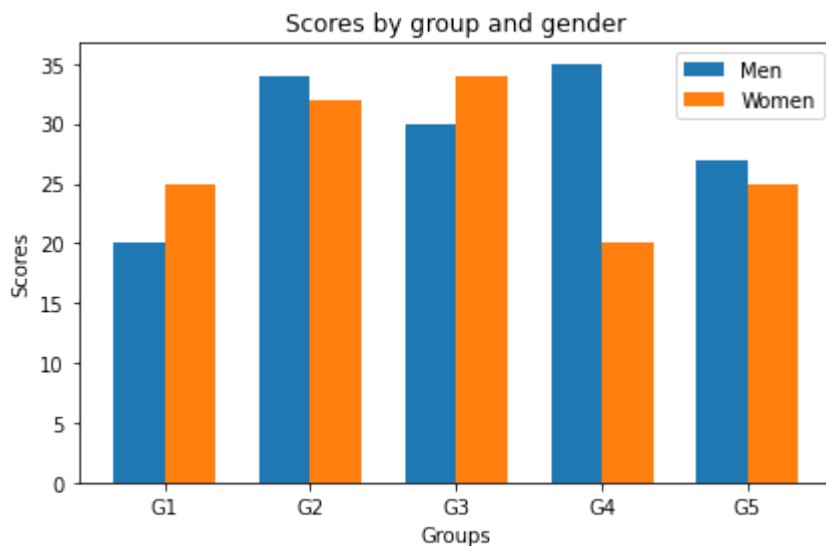
x = np.arange(len(labels))
width = 0.35

fig, ax = plt.subplots()
ax.bar(x - width/2, men_means, width, label='Men')
ax.bar(x + width/2, women_means, width, label='Women')

ax.set_xlabel('Groups')
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

plt.tight_layout()
plt.show()

```



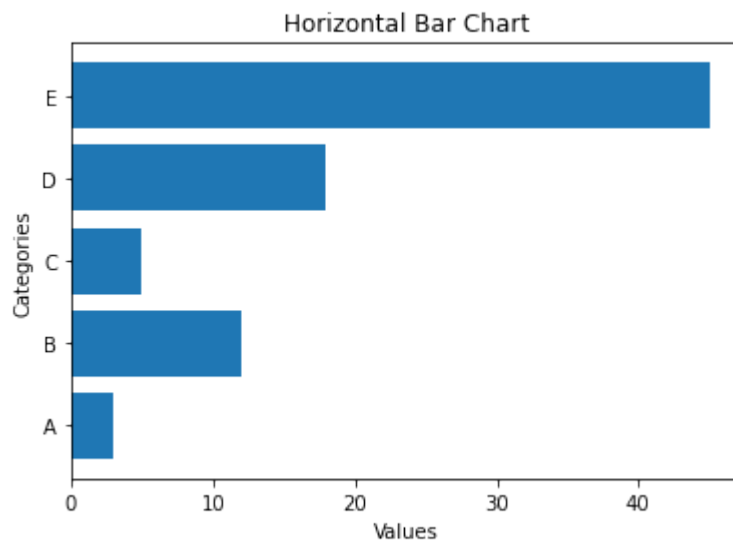
```

In [15]: categories = ['A', 'B', 'C', 'D', 'E']
values = [3, 12, 5, 18, 45]

plt.barh(categories, values)
plt.xlabel('Values')
plt.ylabel('Categories')
plt.title('Horizontal Bar Chart')

plt.show()

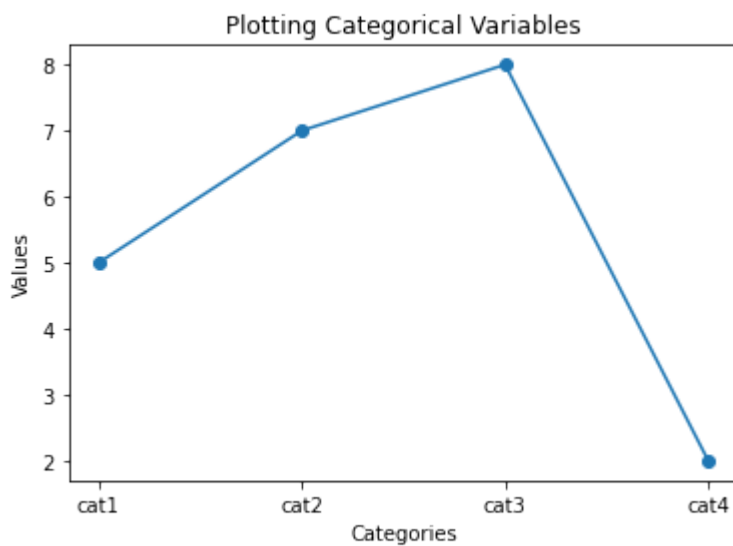
```



```
In [16]: categories = ['cat1', 'cat2', 'cat3', 'cat4']
values = [5, 7, 8, 2]

plt.plot(categories, values, 'o-')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Plotting Categorical Variables')

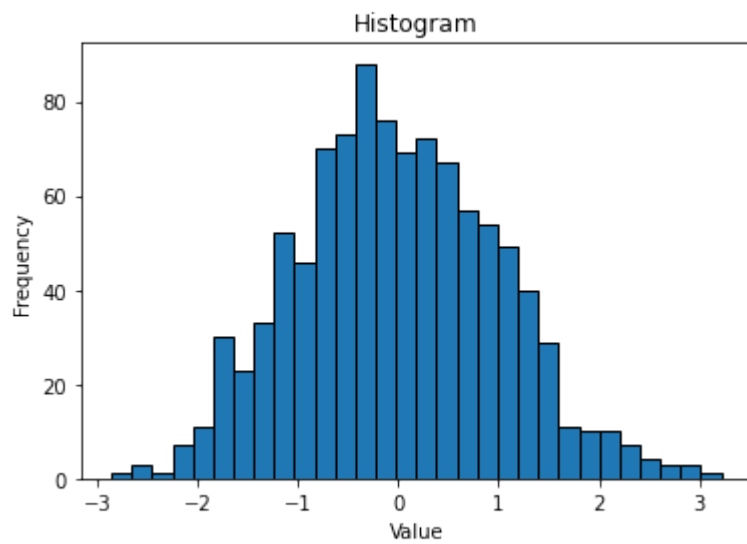
plt.show()
```



```
In [17]: data = np.random.randn(1000)

plt.hist(data, bins=30, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')

plt.show()
```



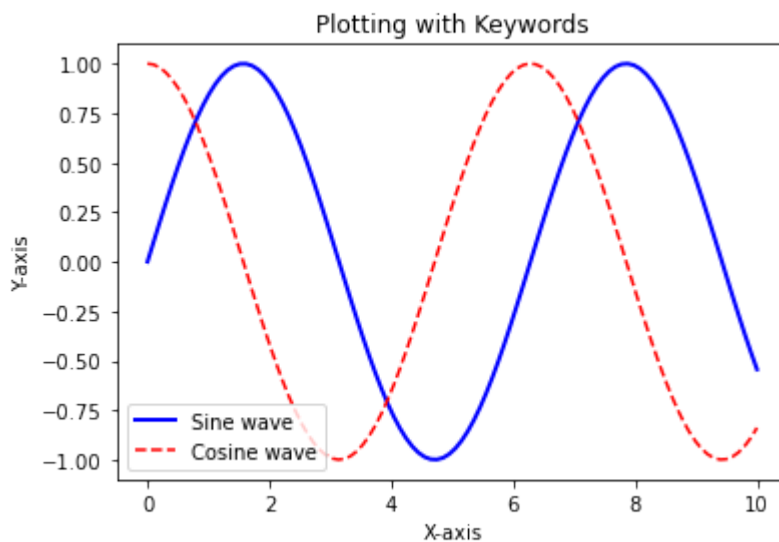
In [18]:

```
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, label='Sine wave', color='blue', linewidth=2)
plt.plot(x, y2, label='Cosine wave', color='red', linestyle='--')

plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Plotting with Keywords')
plt.legend()

plt.show()
```



In []: