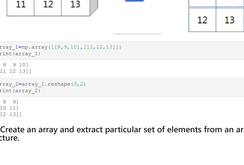
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Write a python program to create an array and covert to a desired dimension, as shown in picture.



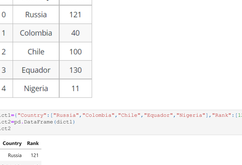
In [2]:
```
array1=np.array([[8,9,10],[11,12,13]])
print(array_1)
```

```
[[ 8  9 10]
 [11 12 13]]
```

In [3]:
```
array_2=array_1.reshape(3,2)
print(array_2)
```

```
[[ 8  9]
 [10 11]
 [12 13]]
```

2. Create an array and extract particular set of elements from an array as shown in the picture.



In [4]:
```
arr1=np.array([[8, 9], [10,11], [12, 13]])
print(arr1)
```

```
[[ 8  9]
 [10 11]
 [12 13]]
```

In [5]:
```
print(arr1[0:2,1])
```

```
[ 9 11]
```

3. Create a dataframe as shown below.

	Country	Rank
0	Russia	121
1	Colombia	40
2	Chile	100
3	Equador	130
4	Nigeria	11

4.Read the given data set and answer the following questions. Show the steps taken to produce your final answer.

In [7]:
```
data=pd.read_csv("C:/Users/rakhi/Downloads/diabetes_model.csv")
data
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	5	77	82	41	42	35.8	0.156	35	0
1	9	122	56	0	0	33.3	1.114	33	1
2	0	113	76	0	0	33.3	0.278	23	1
3	1	139	62	41	480	40.7	0.536	21	0
4	10	161	68	23	132	25.5	0.326	47	1
...	...	...	...	...	...	...	...	...	...
609	4	114	64	0	0	28.9	0.126	24	0
610	2	175	88	0	0	22.9	0.326	22	0
611	3	121	52	0	0	36.0	0.127	25	1
612	7	136	74	26	135	26.0	0.647	51	0
613	4	156	75	0	0	48.3	0.238	32	1

614 rows × 9 columns

1. Display the Glucose level and the Age for the samples having the index ranging from 45 to 60.

In [8]:
```
data1=data.loc[45:60,["Glucose","Age"]]
data1
```

Out[8]:

	Glucose	Age
45	137	21
46	145	70
47	113	22
48	96	27
49	131	26
50	113	21
51	120	34
52	95	24
53	155	46
54	100	46
55	184	49
56	90	29
57	142	61
58	125	27
59	101	26
60	188	22

2. How many patients in the data are having the blood pressure level above 120?

In [9]:
```
data2=data["BloodPressure"]>120
data2.sum()
```

Out[9]:
```
1
```

The number of patients having blood pressure level above 120 is one.

3. Select the rows the age is between 24 and 29 (inclusive)

In [10]:
```
mask=(data["Age"]>=24) & (data["Age"]<=29)
data[mask]
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
5	2	68	70	32	66	25.0	0.187	25	0
7	2	122	52	43	158	36.2	0.816	28	0
11	0	120	74	18	63	30.5	0.285	26	0
22	3	182	74	0	0	30.5	0.345	29	1
23	3	163	70	18	105	31.6	0.268	28	1
...	...	...	...	...	...	...	...	...	...
602	0	93	60	0	0	35.3	0.263	25	0
604	2	100	70	52	57	40.5	0.677	25	0
605	5	86	68	28	71	30.2	0.364	24	0
609	4	114	64	0	0	28.9	0.126	24	0
611	3	121	52	0	0	36.0	0.127	25	1

180 rows × 9 columns

4.Find the minimum and maximum value of insulin level in the data.

In [11]:
```
x=data["Insulin"].max() #maximum value
print("The maximum value of insulin level in the data is",x)
```

```
The maximum value of insulin level in the data is 846
```

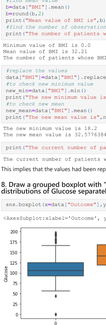
In [12]:
```
y=data["Insulin"].min() #minimum value
print("The minimum value of insulin level in the data is",y)
```

```
The minimum value of insulin level in the data is 0
```

5. Create a histogram of "Age".

In [13]:
```
plt.hist(data["Age"],color="green",bins=14)
plt.title("Histogram of Age")
plt.xlabel("Age")
plt.ylabel("No. of people")
```

Out[13]:
```
Text(0, 0.5, 'No. of people')
```



6. Create a scatter plot of "BMI" vs "Glucose".

In [14]:
```
plt.scatter(data["BMI"],data["Glucose"],color="purple",s=5)
plt.title("Scatterplot of BMI vs Glucose")
plt.xlabel("BMI")
plt.ylabel("Glucose")
```

Out[14]:
```
Text(0, 0.5, 'Glucose')
```


7. Replace the min value of "BMI" with mean value of BMI

In [15]:
```
#find minimum value
a=data["BMI"].min()
print("Minimum value of BMI is",a)
#find mean value
b=data["BMI"].mean()
b=round(b,2)
print("Mean value of BMI is",b)
#find the number of observations that equals minimum value
print("The number of patients whose BMI = 0 is",(data["BMI"]==0.0).sum())

Minimum value of BMI is 0.0
Mean value of BMI is 32.21
The number of patients whose BMI = 0 is 7
```

In [16]:
```
#replace the values
data["BMI"]=data["BMI"].replace(a,b)
#to check new minimum value
new_min=data["BMI"].min()
print("The new minimum value is",new_min)
#to check new mean
new_mean=data["BMI"].mean()
print("The new mean value is",new_mean)

The new minimum value is 18.2
The new mean value is 32.577634364208204
```

In [17]:
```
print("The current number of patients whose BMI =0 is",(data["BMI"]==0.0).sum())
```

```
The current number of patients whose BMI =0 is 0
```

This implies that the values had been replaced.

8. Draw a grouped boxplot with "Outcome" on x-axis and "Glucose" on the y-axis such that we see distributions of Glucose separately for each outcome.

In [18]:
```
sns.boxplot(x=data["Outcome"],y=data["Glucose"])
```

Out[18]:
```
<AxesSubplot:xlabel='Outcome', ylabel='Glucose'>
```


5.

- Create a numpy array (first) with 5 x 5 size and all elements are 0 in that.
- Create a numpy array (second) with 5 x 5 and all elements are 1 in that.
- Combine the arrays first and second vertically, the final array size should be 10 x 5

In [19]:
```
#numpy array with all elements 0
arr2=np.zeros((5,5))
print(arr2)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [20]:
```
#numpy array with all elements 1
arr4=np.ones((5,5))
print(arr4)
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

In [21]:
```
#combining
arr3=np.vstack((arr2,arr4))
print(arr3)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

In [22]:
```
s=arr3.shape
print("The final array size is",s)
```

```
The final array size is (10, 5)
```

In [ ]: