```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
dict1={
    "rollno":[1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1010,1011,1012],
    "name":["harry","rohan","rishik","shubh","jordan","critiano","ronaldo","messi","tristan","/Andrew","...HONEY SINGH",np.NaN,"nina"],
    "marks":[92,82,100,140,96,78,190,69,np.NaN,89,87,75,77],
    "city":["rampur","kolkata","delhi","jaipur","haryana","uttar pradesh","jharkhand","gujrat","__himachal","/chattisgarh",np.NaN,"goa","punjab"],
    "age":[18,19,17,23,20,21,np.NaN,57,90,22,89,20,16],
    "not usefull":["yes","no,",np.NaN,np.NaN,"no","yes","n","y",np.NaN,"y","rt","no","yes"]

}
```

```
df=pd.DataFrame(dict1)
# data before duplicates
df
```

	rollno	name	marks	city	age	not usefull
0	1001	harry	92.0	rampur	18.0	yes
1	1002	rohan	82.0	kolkata	19.0	no,
2	1003	rishik	100.0	delhi	17.0	NaN
3	1004	shubh	140.0	jaipur	23.0	NaN
4	1005	jordan	96.0	haryana	20.0	no
5	1006	critiano	78.0	uttar pradesh	21.0	yes
6	1007	ronaldo	190.0	jharkhand	NaN	n
7	1008	messi	69.0	gujrat	57.0	y
8	1009	tristan	NaN	__himachal	90.0	NaN
9	1010	/Andrew	89.0	/chattisgarh	22.0	y
10	1010	...HONEY SINGH	87.0	NaN	89.0	rt
11	1011	NaN	75.0	goa	20.0	no
12	1012	nina	77.0	punjab	16.0	yes

Next steps: Generate code with df View recommended plots

```
# this will generate whether there are any null values
df.isnull().sum()
```

```
rollno         0
name           1
marks          1
city           1
age            1
not usefull    3
dtype: int64
```

```
#now taking care of null values in the data frame
# data set after removing duplicates
df=df.drop_duplicates("rollno")
df
```

	rollno	name	marks	city	age	not usefull
0	1001	harry	92.0	rampur	18.0	yes
1	1002	rohan	82.0	kolkata	19.0	no,
2	1003	rishik	100.0	delhi	17.0	NaN
3	1004	shubh	140.0	jaipur	23.0	NaN
4	1005	jordan	96.0	haryana	20.0	no
5	1006	critiano	78.0	uttar pradesh	21.0	yes
6	1007	ronaldo	190.0	jharkhand	NaN	n
7	1008	messi	69.0	gujrat	57.0	y
8	1009	tristan	NaN	__himachal	90.0	NaN
9	1010	/Andrew	89.0	/chattisgarh	22.0	y
11	1011	NaN	75.0	goa	20.0	no
12	1012	nina	77.0	punjab	16.0	yes

Next steps: Generate code with df View recommended plots

```
df=df.drop(columns="not usefull")
df
```

	rollno	name	marks	city	age
0	1001	harry	92.0	rampur	18.0
1	1002	rohan	82.0	kolkata	19.0
2	1003	rishik	100.0	delhi	17.0
3	1004	shubh	140.0	jaipur	23.0
4	1005	jordan	96.0	haryana	20.0
5	1006	critiano	78.0	uttar pradesh	21.0
6	1007	ronaldo	190.0	jharkhand	NaN
7	1008	messi	69.0	gujrat	57.0
8	1009	tristan	NaN	__himachal	90.0
9	1010	/Andrew	89.0	/chattisgarh	22.0
11	1011	NaN	75.0	goa	20.0
12	1012	nina	77.0	punjab	16.0

Next steps: Generate code with df View recommended plots

```
# data before strip
df["city"]
```

```
0           rampur
1          kolkata
2            delhi
3           jaipur
4          haryana
5     uttar pradesh
6         jharkhand
7           gujrat
8        __himachal
9      /chattisgarh
11              goa
12           punjab
Name: city, dtype: object
```

```
# data after striping
df["city"]=df["city"].str.lstrip("__")
df["city"]=df["city"].str.lstrip("/")
df["city"]
```

```
0           rampur
1          kolkata
```

```
2             delhi
3            jaipur
4           haryana
5     uttar pradesh
6         jharkhand
7            gujrat
8          himachal
9       chattisgarh
11              goa
12            punjab
Name: city, dtype: object
```

```
df["name"]=df["name"].str.lstrip("/")
df
```

	rollno	name	marks	city	age
0	1001	harry	92.0	rampur	18.0
1	1002	rohan	82.0	kolkata	19.0
2	1003	rishik	100.0	delhi	17.0
3	1004	shubh	140.0	jaipur	23.0
4	1005	jordan	96.0	haryana	20.0
5	1006	critiano	78.0	uttar pradesh	21.0
6	1007	ronaldo	190.0	jharkhand	NaN
7	1008	messi	69.0	gujrat	57.0
8	1009	tristan	NaN	himachal	90.0
9	1010	Andrew	89.0	chattisgarh	22.0
11	1011	NaN	75.0	goa	20.0
12	1012	nina	77.0	punjab	16.0

Next steps: Generate code with df View recommended plots

```
# chenking null values
from sklearn.impute import SimpleImputer
print(f'Number of null values before imputing:{df.age.isnull().sum()}')
```

```
Number of nullvalues before imputing:1
```

```
#now replacing the null values with mean of other values in the column
imp=SimpleImputer(strategy="mean")
```

```
imp=SimpleImputer(strategy="mean")
df["age"]=imp.fit_transform(df[["age"]])
print(f"the number of NULL values in the age column is: {df.age.isnull().sum()}")
```

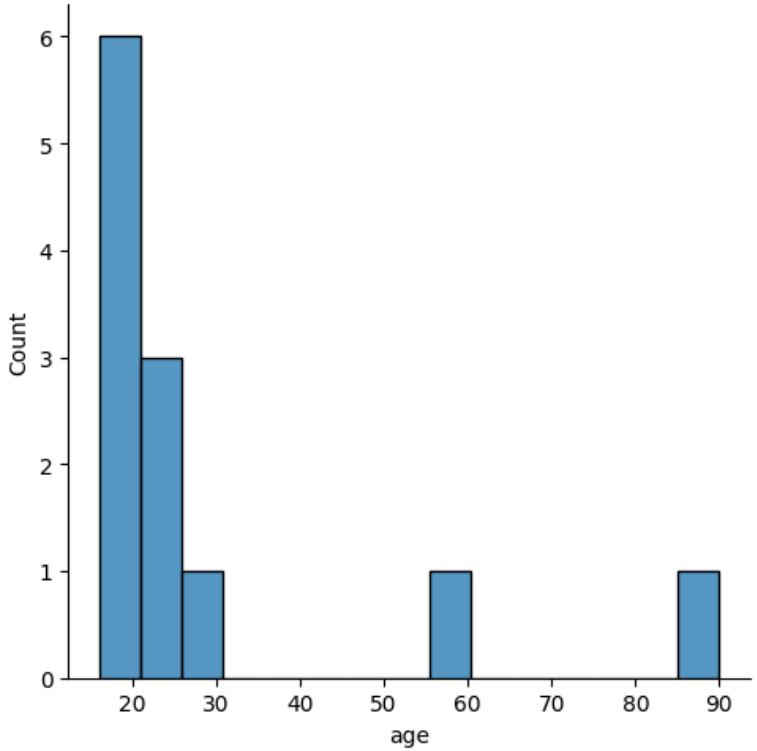
```
the number of NULL values in the age column is: 0
```

```
df.age
```

```
0     18.000000
1     19.000000
2     17.000000
3     23.000000
4     20.000000
5     21.000000
6     29.363636
7     57.000000
8     90.000000
9     22.000000
11    20.000000
12    16.000000
Name: age, dtype: float64
```

```
import seaborn as sns
sns.displot(df["age"])
```

```
<seaborn.axisgrid.FacetGrid at 0x7c546e2c6830>
```

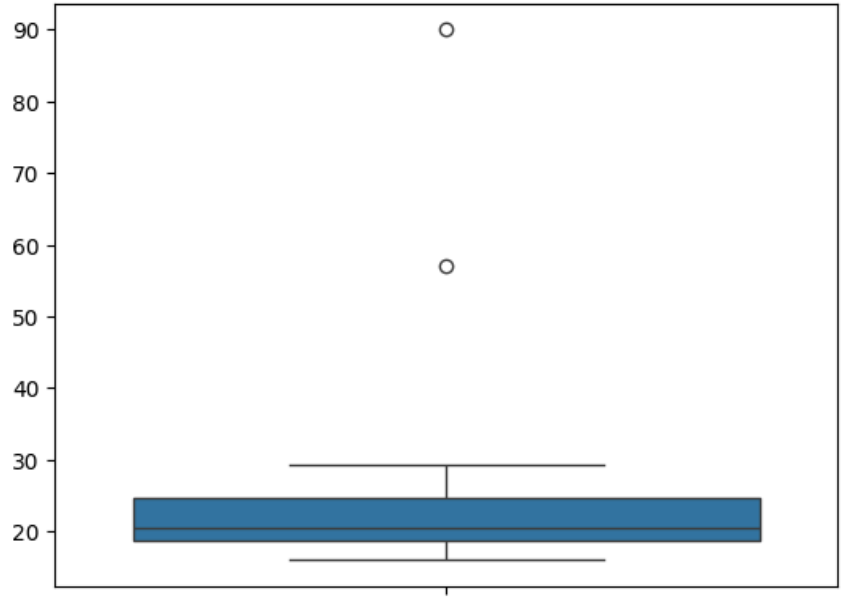



```
dataset=df["age"]
dataset
```

```
0     18.000000
1     19.000000
2     17.000000
3     23.000000
4     20.000000
5     21.000000
6     29.363636
7     57.000000
8     90.000000
9     22.000000
11    20.000000
12    16.000000
Name: age, dtype: float64
```

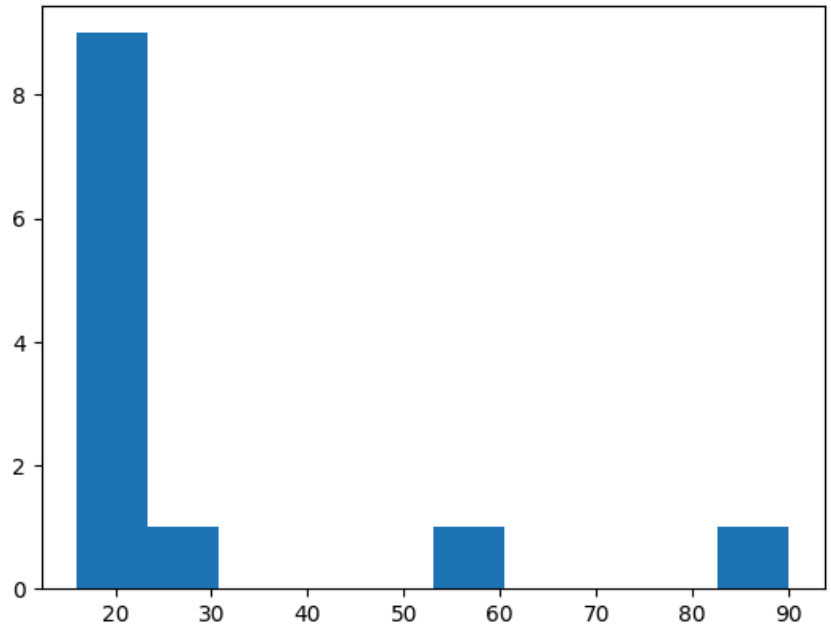
```
import seaborn as sns
sns.boxplot(data=dataset)
```

```
<Axes: >
```




```
# through this we can get to know that whether the dataset contains outliers or not
plt.hist(dataset)
```

```
(array([9., 1., 0., 0., 0., 1., 0., 0., 0., 1.]),
 array([16. , 23.4, 30.8, 38.2, 45.6, 53. , 60.4, 67.8, 75.2, 82.6, 90. ]),
 <BarContainer object of 10 artists>)
```



IQR

1. SORT THE DATASET
2. CALCULATE Q1-->25%
3. IQR(Q.3-Q1)
4. Find the Lower Fence(q1-1.5(iqr))
5. Find the Upper Fence (q.3 +1.5(iqr))

```
# we will find outliers in the box plot method
dataset=sorted(dataset)
data1=[]

for i in dataset:
  data1.append(int(i))
print(data1)
```

```
[16, 17, 18, 19, 20, 20, 21, 22, 23, 29, 57, 90]
```

```
q1,q3=np.percentile(data1,[25,75])
print(q1,q3)
```