```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
ls drive/MyDrive/
```

```
'10 th marksheet .pdf'
'12 th marksheet .pdf'
'aadhar card .pdf'
'bank book.pdf'
 bank_train.csv
'Colab Notebooks'/
'community certificate .pdf'
'Data_set (1).csv'
'IMG20240909164329~2 (1).jpg'
 IMG20240909164329~2.jpg
 IMG20241021103250.jpg
 IMG-20241021-WA0004.jpg
 IMG20241029152859.jpg
 IMG-20241115-WA0005~2.jpg
'Nativity certificate .pdf'
'PDF Reader.pdf'
 Screenshot_2024-09-09-20-38-44-18_ccbe52b0e23c52d29f1b024e2f6eecaa.jpg
'web assignment.pdf'
```

```
import pandas as pd
df=pd.read_csv("/content/drive/MyDrive/Data_set (1).csv")
```

```
df
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	NaN	South Korea	16	Friday, Saturday	tvN	8.9	33.0	1	111706.0
1	NaN	South Korea	16	Friday, Saturday	jTBC	8.7	89.0	2	100950.0
2	Descendants of the Sun	South Korea	16	Wednesday, Thursday	KBS2	8.7	77.0	3	96318.0
3	Boys Over Flowers	South Korea	25	Monday, Tuesday	KBS2	7.7	2249.0	4	94228.0
4	W	South Korea	16	Wednesday, Thursday	MBC	8.5	201.0	5	92121.0
...	...	...	...	...	...	...	...	...	...
95	Shut Up: Flower Boy Band	South Korea	16	Monday, Tuesday	tvN	8.1	806.0	99	34668.0
96	Blood	South Korea	20	Monday, Tuesday	KBS2	7.4	3271.0	100	34666.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   show_name                 96 non-null     object
 1   country                   100 non-null    object
 2   num_episodes              100 non-null    int64
 3   aired_on                  99 non-null     object
 4   original_network          99 non-null     object
 5   rating                    96 non-null     float64
 6   current_overall_rank      97 non-null     float64
 7   lifetime_popularity_rank  100 non-null    int64
 8   watchers                  97 non-null     float64
dtypes: float64(3), int64(2), object(4)
memory usage: 7.2+ KB
```

```
df.describe()
```

	num_episodes	rating	current_overall_rank	lifetime_popularity_rank	watchers
count	100.000000	96.000000	97.000000	100.000000	97.000000
mean	18.980000	8.293750	731.247423	51.650000	52994.907216
std	6.846041	0.424714	857.597007	30.133164	17551.028458
min	8.000000	7.300000	2.000000	1.000000	34523.000000
25%	16.000000	8.100000	194.000000	25.750000	39545.000000
50%	16.000000	8.300000	441.000000	51.500000	46963.000000
75%	20.000000	8.600000	806.000000	77.250000	63140.000000
max	50.000000	9.100000	3788.000000	103.000000	111706.000000

```
df.isnull()
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	True	False	False	False	False	False	False	False	False
1	True	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
95	False	False	False	False	False	False	False	False	False
96	False	False	False	False	False	False	False	False	False
97	False	False	False	False	False	False	False	False	True
98	False	False	False	False	False	False	False	False	False
99	False	False	False	False	False	False	False	False	False

100 rows × 9 columns

```
df.notnull()
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	False	True	True	True	True	True	True	True	True
1	False	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...
95	True	True	True	True	True	True	True	True	True
96	True	True	True	True	True	True	True	True	True
97	True	True	True	True	True	True	True	True	False
98	True	True	True	True	True	True	True	True	True
99	True	True	True	True	True	True	True	True	True

100 rows × 9 columns

```
df.isnull().sum()
```

	0
show_name	4
country	0
num_episodes	0
aired_on	1
original_network	1
rating	4
current_overall_rank	3
lifetime_popularity_rank	0
watchers	3

dtype: int64

```
df.dropna(axis=1)
```

	country	num_episodes	lifetime_popularity_rank
0	South Korea	16	1
1	South Korea	16	2
2	South Korea	16	3
3	South Korea	25	4
4	South Korea	16	5
...	...	...	...
95	South Korea	16	99
96	South Korea	20	100
97	South Korea	16	101
98	South Korea	20	102
99	South Korea	16	103

100 rows × 3 columns

```
df.dropna(axis=0)
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
2	Descendants of the Sun	South Korea	16	Wednesday, Thursday	KBS2	8.7	77.0	3	96318.0
3	Boys Over Flowers	South Korea	25	Monday, Tuesday	KBS2	7.7	2249.0	4	94228.0
4	W	South Korea	16	Wednesday, Thursday	MBC	8.5	201.0	5	92121.0
5	You Who Came from the Stars	South Korea	21	Wednesday, Thursday	SBS	8.6	112.0	6	91360.0
6	Weightlifting Fairy Kim Bok Joo	South Korea	16	Wednesday, Thursday	MBC	8.8	40.0	7	91330.0
...	...	...	...	...	...	...	...	...	...
94	Flower of Evil	South Korea	16	Wednesday, Thursday	tvN	9.1	4.0	98	34901.0
	Shut Up:	South		Monday					

```
df.fillna(0)
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	0	South Korea	16	Friday, Saturday	tvN	8.9	33.0	1	111706.0
1	0	South Korea	16	Friday, Saturday	jTBC	8.7	89.0	2	100950.0
2	Descendants of the Sun	South Korea	16	Wednesday, Thursday	KBS2	8.7	77.0	3	96318.0
3	Boys Over Flowers	South Korea	25	Monday, Tuesday	KBS2	7.7	2249.0	4	94228.0
4	W	South Korea	16	Wednesday, Thursday	MBC	8.5	201.0	5	92121.0
...	...	...	...	...	...	...	...	...	...
95	Shut Up: Flower Boy Band	South Korea	16	Monday, Tuesday	tvN	8.1	806.0	99	34668.0
96	Blood	South Korea	20	Monday, Tuesday	KBS2	7.4	3271.0	100	34666.0

```
df.fillna(method='ffill')
```

```
<ipython-input-15-5c0beae7dc1e>:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use c
  df.fillna(method='ffill')
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	NaN	South Korea	16	Friday, Saturday	tvN	8.9	33.0	1	111706.0
1	NaN	South Korea	16	Friday, Saturday	jTBC	8.7	89.0	2	100950.0
2	Descendants of the Sun	South Korea	16	Wednesday, Thursday	KBS2	8.7	77.0	3	96318.0
3	Boys Over Flowers	South Korea	25	Monday, Tuesday	KBS2	7.7	2249.0	4	94228.0
4	W	South Korea	16	Wednesday, Thursday	MBC	8.5	201.0	5	92121.0
...	...	...	...	...	...	...	...	...	...
95	Shut Up: Flower Boy Band	South Korea	16	Monday, Tuesday	tvN	8.1	806.0	99	34668.0
96	Blood	South Korea	20	Monday, Tuesday	KBS2	7.4	3271.0	100	34666.0

```
df.fillna(method='bfill')
```

```
<ipython-input-16-b823574c06e2>:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use c
  df.fillna(method='bfill')
```

	show_name	country	num_episodes	aired_on	original_network	rating	current_overall_rank	lifetime_popularity_rank	watchers
0	Descendants of the Sun	South Korea	16	Friday, Saturday	tvN	8.9	33.0	1	111706.0
1	Descendants of the Sun	South Korea	16	Friday, Saturday	jTBC	8.7	89.0	2	100950.0
2	Descendants of the Sun	South Korea	16	Wednesday, Thursday	KBS2	8.7	77.0	3	96318.0
3	Boys Over Flowers	South Korea	25	Monday, Tuesday	KBS2	7.7	2249.0	4	94228.0
4	W	South Korea	16	Wednesday, Thursday	MBC	8.5	201.0	5	92121.0
...	...	...	...	...	...	...	...	...	...
95	Shut Up: Flower Boy Band	South Korea	16	Monday, Tuesday	tvN	8.1	806.0	99	34668.0
96	Blood	South Korea	20	Monday, Tuesday	KBS2	7.4	3271.0	100	34666.0

```
df['rating'].fillna(value=df['rating'].mean())
```

	rating
0	8.9
1	8.7
2	8.7
3	7.7
4	8.5
...	...
95	8.1
96	7.4
97	8.8
98	8.2
99	8.5

100 rows × 1 columns

dtype: float64

```
df['current_overall_rank'].fillna(value=df['current_overall_rank'].mean())
```

```
     current_overall_rank
0                    33.0
1                    89.0
2                    77.0
3                  2249.0
4                   201.0
...                   ...
95                  806.0
96                 3271.0
97                   51.0
98                  605.0
99                  238.0
100 rows × 1 columns

dtype: float64
```

```
df['watchers'].fillna(value=df['watchers'].mean())
```

```
           watchers
0     111706.000000
1     100950.000000
2      96318.000000
3      94228.000000
4      92121.000000
...             ...
95     34668.000000
96     34666.000000
97     52994.907216
98     34615.000000
99     34523.000000
100 rows × 1 columns

dtype: float64
```

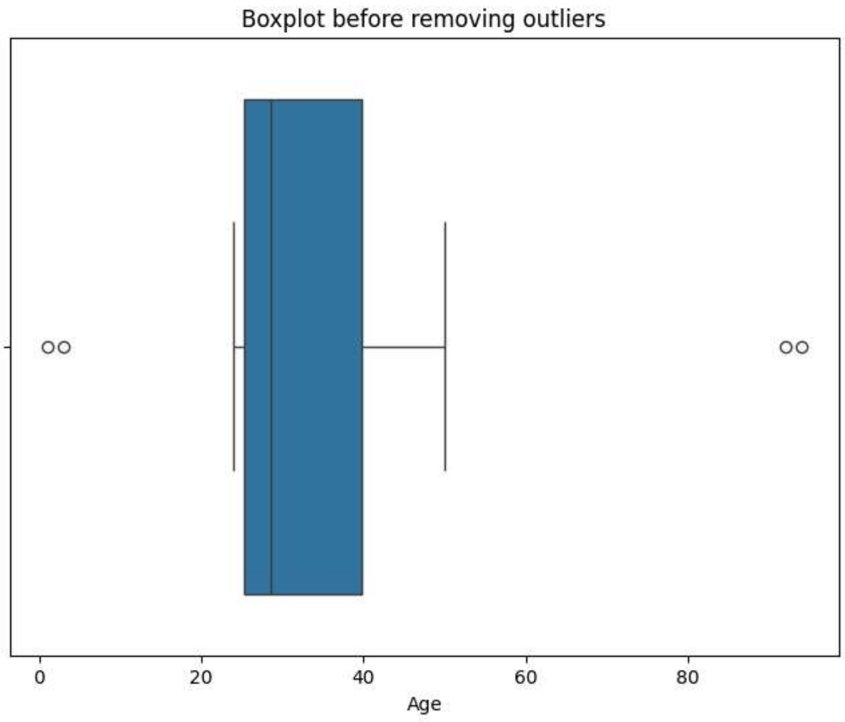
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
age = [1, 3, 28, 27, 25, 92, 30, 39, 40, 50, 26, 24, 29, 94]
af = pd.DataFrame(age, columns=["Age"])
```

```
af
```

	Age
0	1
1	3
2	28
3	27
4	25
5	92
6	30
7	39
8	40
9	50
10	26
11	24
12	29
13	94

```
plt.figure(figsize=(8, 6))
sns.boxplot(x=af["Age"])
plt.title("Boxplot before removing outliers")
plt.show()
```



```
Q1 = af["Age"].quantile(0.25)
Q3 = af["Age"].quantile(0.75)
IQR = Q3 - Q1
```

```
IQR
```

```
14.5
```

```
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
lower_bound
```

```
3.5
```

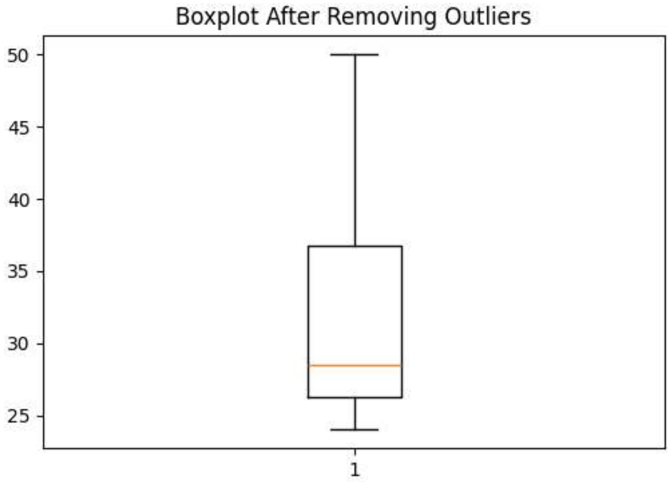
```
upper_bound
```

```
61.5
```

```
outliers = af[(af['Age'] < lower_bound) | (af['Age'] > upper_bound)]
print("Outliers detected:", outliers['Age'].tolist())
```

```
Outliers detected: [1, 3, 92, 94]
```

```
plt.figure(figsize=(6,4))
plt.boxplot(af_cleaned['Age'])
plt.title("Boxplot After Removing Outliers")
plt.show()
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

```
data = [1, 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, 158]
df = pd.DataFrame(data, columns=['Values'])
```

```
plt.figure(figsize=(6,4))
plt.boxplot(df['Values'])
plt.title("Boxplot Before Removing Outliers")
plt.show()
```

```
z_scores = stats.zscore(df['Values'])
threshold = 3
outliers = df[np.abs(z_scores) > threshold]
print("Outliers detected:", outliers['Values'].tolist())
```

```
Outliers detected: [158]
```

```
df_cleaned = df[np.abs(z_scores) <= threshold]
df_cleaned
```

	Values
0	1
1	12
2	15
3	18
4	21
5	24
6	27
7	30
8	33
9	36
10	39
11	42
12	45
13	48
14	51
15	54
16	57
17	60