Start coding or generate with AI.

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

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

```
ls drive/MyDrive/DS2024/Data_set.csv
```

```
drive/MyDrive/DS2024/Data_set.csv
```

Data Cleaning

```
import pandas as pd
# READ CSV FILE HERE
df=pd.read_csv('drive/MyDrive/DS2024/Data_set.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	[illegible]	Monday,	[illegible]	7.4	[illegible]	[illegible]	[illegible]

```
# CHECK OUT NULL VALUES IN DATA SET USING FUNCTION
df_null=df.isnull()
df_null
```

	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

```
# DISPLAY THE SUM ON NULL VALUES IN EACH ROWS
df_null_sum=df.isnull().sum()
df_null_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

```
# DROP NULL VALUES
df_dropna=df.isnull().dropna()
df_dropna
```

	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

```
# FILL NULL VALUES WITH CONSTANT VALUE "O"
df_nafill_0=df.fillna(0)
df_nafill_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	20	Monday,	KBS2	7.4	2271.0	100	34660.0

```
# FILL NULL VALUES WITH ffill METHOD
df_ffill=df.ffill()
df_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	20	Monday,	KBS2	7.4	2271.0	100	34660.0

```
# FILL NULL VALUES WITH bfill METHOD
df_bfill=df.bfill()
df_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	20	Monday,	KBS2	7.4	2271.0	100	34660.0

```
# CALCULATE MEAN VALUE OF A COLUMN AND FILL IT WITH NULL VALUES
df_mean1=df['num_episodes'].fillna(df['num_episodes'].mean())
df_mean1
```

	num_episodes
0	16
1	16
2	16
3	25
4	16
...	...
95	16
96	20
97	16
98	20
99	16

100 rows × 1 columns

```
df_mean2=df['rating'].fillna(df['rating'].mean())
df_mean2
```

	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

```
df_mean3=df['current_overall_rank'].fillna(df['current_overall_rank'].mean())
df_mean3
```

	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

```
df_mean4=df['lifetime_popularity_rank'].fillna(df['lifetime_popularity_rank'].mean())
df_mean4
```

	lifetime_popularity_rank
0	1
1	2
2	3
3	4
4	5
...	...
95	99
96	100
97	101
98	102
99	103

100 rows × 1 columns

```
df_mean5=df['watchers'].fillna(df['watchers'].mean())
df_mean5
```

	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

```
# DROP NULL VALUES
df_dropna=df.dropna()
df_dropna
```

	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

Outlier Detection and Removal - IQR

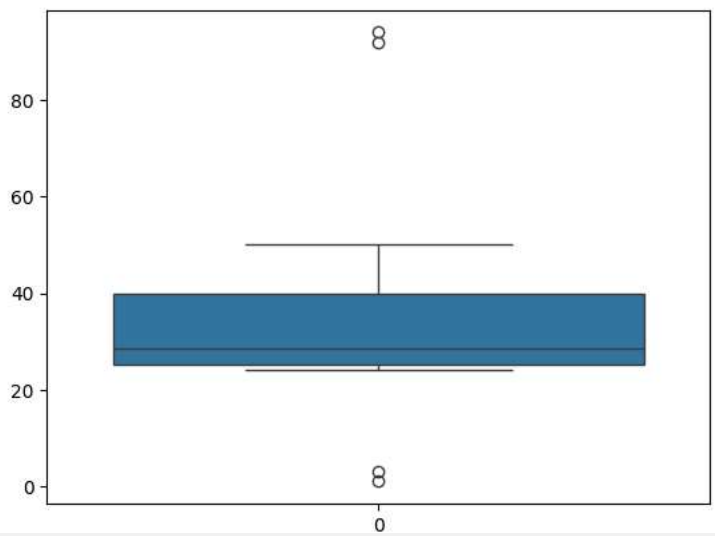
```
import pandas as pd
import seaborn as sns


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

	0
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

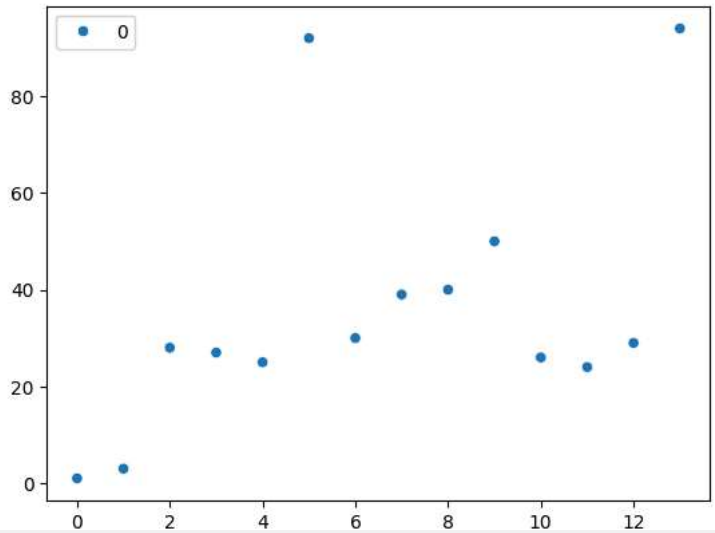
```
# USE BOXPLOT FUNCTION HERE TO DETECT OUTLIER
sns.boxplot(af)
```

```
<Axes: >
```




```
sns.scatterplot(af)
```

```
<Axes: >
```




```
q1=af.quantile(0.25)
q2=af.quantile(0.5)
q3=af.quantile(0.75)
```

```
iqr=q3-q1
iqr
```

	0
0	14.5

```
import numpy as np
```

```
Q1=np.percentile(af,25)
Q2=np.percentile(af,50)
Q3=np.percentile(af,75)
```

```
IQR=Q3-Q1
```

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

```
outliers = [x for x in age if x < lower_bound or x > upper_bound]
```

```
print('Q1:',Q1)
print('Q3:',Q3)
print('IQR:',IQR)
print('Lower bound:',lower_bound)
print('Upper bound:',upper_bound)
print('Outliers:',outliers)
```

```
Q1: 25.25
Q3: 39.75
IQR: 14.5
Lower bound: 3.5
Upper bound: 61.5
Outliers: [1, 3, 92, 94]
```

```
af=af[((af>=lower_bound)&(af<=upper_bound))]
af
```

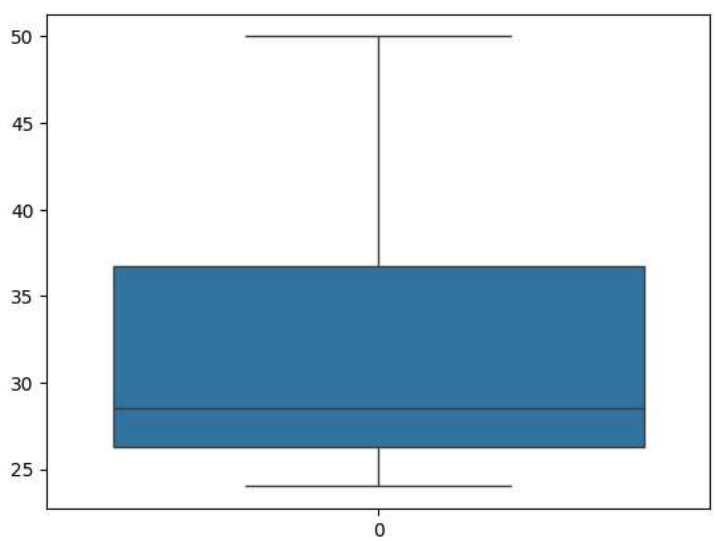
	0
0	NaN
1	NaN
2	28.0
3	27.0
4	25.0
5	NaN
6	30.0
7	39.0
8	40.0
9	50.0
10	26.0
11	24.0
12	29.0
13	NaN

```
af.dropna()
```

	0
2	28.0
3	27.0
4	25.0
6	30.0
7	39.0
8	40.0
9	50.0
10	26.0
11	24.0
12	29.0

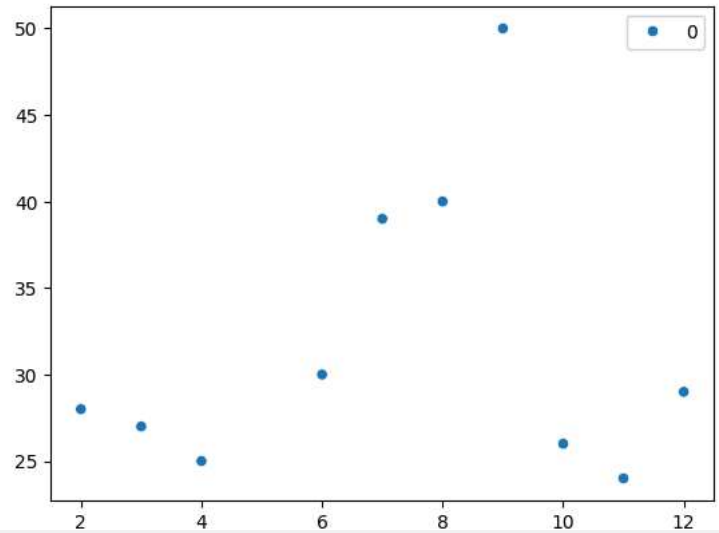
```
sns.boxplot(af)
```

```
<Axes: >
```




```
sns.scatterplot(af)
```

```
<Axes: >
```



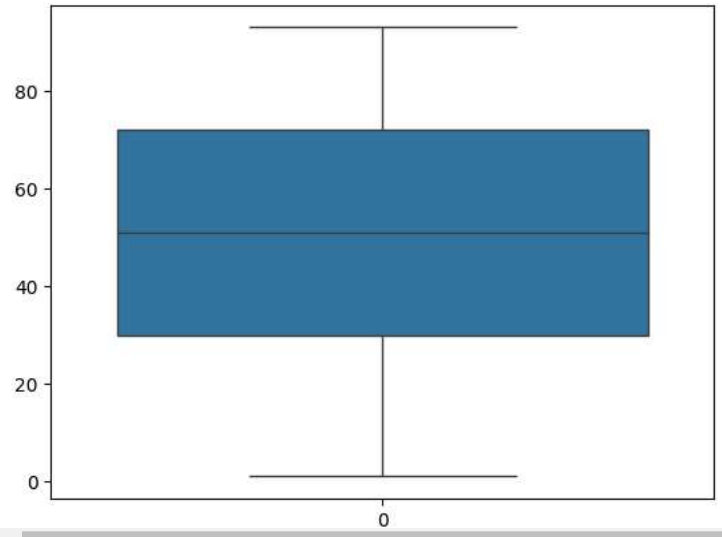
Z Score

```
from scipy import stats #STATS METHOD IS USED TO IMPLEMENT Z SCORE METHOD
import numpy as np
import pandas as pd
import seaborn as sns
```

```
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]
df=pd.DataFrame(data)
```

```
# USE BOXPLOT FUNCTION HERE TO DETECT OUTLIER
sns.boxplot(df)
```

```
<Axes: >
```




```
mean=np.mean(data)
mean
```

```
50.724137931034484
```

```
std=np.std(data)
std
```

```
25.59889080534025
```

```
# PERFORM Z SCORE METHOD AND DETECT OUTLIER VALUES
z=np.abs(stats.zscore(df))
z
```

	0
0	1.942433
1	1.512727
2	1.395535
3	1.278342
4	1.161149
5	1.043957
6	0.926764
7	0.809572
8	0.692379
9	0.575187
10	0.457994
11	0.340801
12	0.223609
13	0.106416
14	0.010776
15	0.127969
16	0.245161
17	0.362354
18	0.479547
19	0.596739
20	0.713932
21	0.831124
22	0.948317
23	1.065510
24	1.182702
25	1.299895
26	1.417087
27	1.534280
28	1.651472

```
threshold=3
outliers = df[abs(df) > 3]
print("Outliers:")
print(outliers)
```

```
Outliers:
        0
0     NaN
1    12.0
2    15.0
3    18.0
4    21.0
5    24.0
6    27.0
7    30.0
8    33.0
9    36.0
10   39.0
11   42.0
12   45.0
13   48.0
14   51.0
15   54.0
```

```
16  57.0
17  60.0
18  63.0
19  66.0
20  69.0
21  72.0
22  75.0
23  78.0
24  81.0
25  84.0
26  87.0
27  90.0
28  93.0
```

```
# Remove outliers
df_cleaned = df[(z <= threshold)]
df_cleaned
```

	0
0	1
1	12
2	15
3	18
4	21
5	24
6	27
7	30
8	33
9	36
10	39