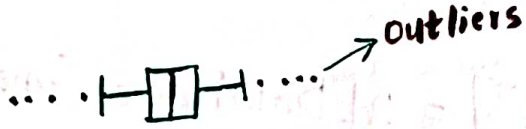
3/4/22 2:08 AM

Outliers → # Applicable Only For Continous Data

1Que :- What is outlier ?

Ans :- An Outlier is a data point in a data set. that is distant From all other observation which is significantly different From The Remaing Data



(or)

- The data point that lies outside the over all Distribution of the dataset.

2 Que :- What are Impacts having Outliers in a DataSet ?

Ans :- it will cause Various Problems in Statistical Analysis. (it may cause a significant impact on mean and Standard deviation)

Ex :- $\bar{x}$ = mean (affect)

$\frac{\Sigma[x-\bar{x}]}{n-1}$ = varience → $\sqrt{\frac{\Sigma(x-\bar{x})^2}{n-1}}$ (Standard deviation)

all are Effected by Outliers

* In addition, Some Machine learning models are Sensitive to outliers, which may Decrease There Performance.

3 Que : Reasons For Outliers ?

Ans :
1. Data Entry Errors (Ex:- Salary = 1,00,000, But 10,000 Entering)
2. Measurement Errors (Ex :- Measuring in meters instead of KM)
3. Instrumental Error

4 Que :- Types of Outliers ?

Ans :- Univariate Outlier ----> Identifying Outlier For Single Variable

* Bivariate Outlier ----> Identifying as Outlier by Analyzing 2 Variables at a Time

Importing Libraries

```
# Import numpy as np
# Import pandas as pd
# Import matplotlib as plt.
# Import Seaborn as Sns.
```

Real Time Case study

boston - House Price Dataset

```
# boston = pd.read_csv("D:\\datascience\\shivaramakrishna class\\6. Data cleaning\\boston.csv")
```

```
boston.head()
```

out

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRTIO	B	LSTAT	PRICE
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

```
# boston.info()
```

506 rows, 14 columns

out

0	CRIM	Float 64
1	ZN	Float 64
2	INDUS	Float 64
3	CHAS	→ int 64
4	NOX	Float 64
5	RM	Float 64
6	AGE	Float 64
7	DIS	
8	RAD	→ Int 64
9	TAX	→ Int 64
10	PTRTIO	Float 64
11	B	Float 64
12	LSTAT	Float 64
13	PRICE	Float 64

(Float 64 (11), int 64 (3))

Various ways of finding The Outlier

Changing Every Value to Zscore.

Ex:-

X	Zscore
6	$\frac{6-5.6}{1}$
4	$\frac{4-5.6}{1}$
9	$\frac{9-5.6}{1}$
7	$\frac{7-5.6}{1}$
2	$\frac{2-5.6}{1}$

$Z = \frac{X-\mu}{\sigma}$

$\therefore \mu = \frac{6+4+9+7+2}{5}$

$\mu = 28/5 = 5.6$

Ex:- $\sigma = 1$

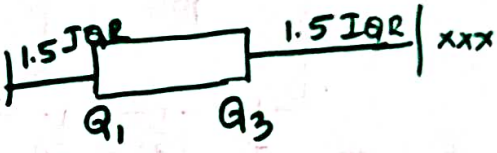
Option (1) **Z score**

$|Z\text{score}| > 2$

$[\because Z < -2, \; Z > +2]$ |modulus|

Option (2) **IQR** (Inter Quartile Range)

- any Value $< Q_1 - 1.5\,IQR$
- any value $> Q_3 + 1.5\,IQR$



any value Less than $Q_1 - 1.5\,IQR$ is Outlier.

any value More Than $Q_3 - 1.5\,IQR$ is Outlier

Option (3) **visualization**

* Box plot, Histogram For Univarte Outliers
* Scatter plot For bivariate Outliers.

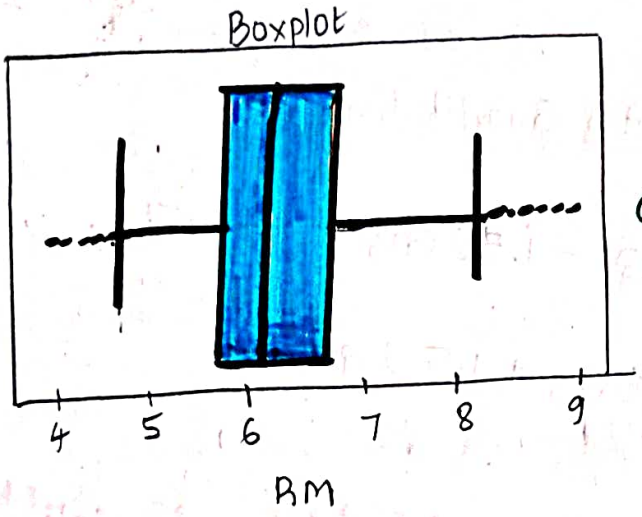
Option ③

Detection of Outliers of "RM" column (Based on Boxplot)

```
# sns.boxplot(boston.RM)
  plt.title("Boxplot")
  plt.show()
```

* Over all view of Outlier

Out :



Outliers in Both Tails of "RM".

Option ①

Detection of Outliers of "RM" column (Based on Zscore)

* For Exact Outliers in Data

$$Z\text{score} = \frac{X - \mu}{\sigma}$$

```
# boston["RM"].mean(), boston["RM"].std()
```

Out (6.284, 0.702)

```
# boston["RM_Zscore"] = (boston["RM"] - boston["RM"].mean()) / boston["RM"].std()
```

creating new column and storing Zscores

X → boston["RM"]; μ → boston["RM"].mean(); σ → boston["RM"].std()

Output :-

Clarity :-

$$\frac{\text{boston["RM"]} - \text{boston["RM"].mean()}}{\text{boston["RM"].std()}}$$

(x − μ) / σ

E Answers Ammi

boston["RM_Zscore"] = ↑ Endulo store ayethaye.

Zscore

original

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	...	RM SCORE
0	0.0632	18.0	2.31	.	.	6.575	.	.	.	.	.	.		0.413263
1	0.0273	0.0	7.07	.	.	6.421	.	.	.	.	.	.		0.194082
2	0.0272	0.0	2.18	.	.	7.185	.	.	.	.	.	.		1.281446
...														
505	0.0474	0.0	11.93	.	.	6.030	.	.	.	.	.	.		-0.362408

506 x 15 columns

Now, Identify Outliers in this column ↑

Outlier = boston[(boston["RM_Zscore"] > +2) OR (boston["RM_Zscore"] < -2)]

2 conditions

Greater Than +2

Less Than -2

Out :- Outlier

	CRIM	...	RM SCORE
97			2.539
98			2.185
162			3.47
163			-2.127
66			-3.055

3/4/22 4:30 Am

Time
1:00pm

Outlier. Shape

Out :: (31, 15)

option (2)
Detection of Outlier of RM (based on IQR)

- Calculate First (q_1) and third quartile (q_3)
- Find interquartile range ($q_3 - q_1$)
- Find Lower bound $q_1 1.5$ & Find upper bond $q_3 1.5$

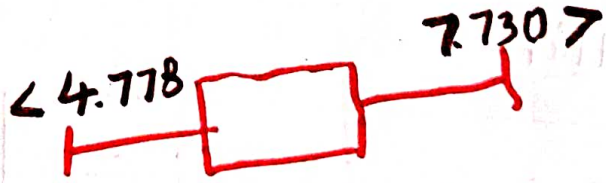
for Exact no. of range + - of outliers

```
# Q3 = boston["RM"].quantile(0.75)
  Q1 = boston["RM"].quantile(0.25)

  IQR = Q3 - Q1
  Lower_limit = Q1 - (IQR * 1.5)
  Upper_limit = Q3 + (IQR * 1.5)

  Lower_limit, upper_Limit
```

$Q_1 = 0.25$ (ol)
$Q_3 = 0.75$ (ol)

$IQR = Q_3 - Q_1$

$\Rightarrow Q_3 + IQR * 1.5$
$\Rightarrow Q_1 - IQR * 1.5$



Out :- 4.778 , 7.730
Min Max

* Which is having less than 4.778 is outlier
* which is having Greater than 7.730 is Outlier

Methods of deal The Outliers

3R Techinque
* Remove → Deleting The outliers
* Replace → changing values / Data Manipulation
* Retain → Treat Them Seperately.
Work without outlier / with outlier "Work" seperately.

Outliers should be detected and "REMOVED" Only From Training data set, NOT from The Test set. So we Should first divide our data set into train and tests. and remove Outliers in the train set, but keep those in test set, and Measure how well our model is doing.

option (1) Removing * (Let's trimm The dataset) → 4.77

```
# boston_trimmed = boston[(boston["RM"] > Lower_limit) &
                          (boston["RM"] < upper_limit)]
                                          → 7.73
```

```
boston_trimmed
```

Out:

Which ever Data is Greater than 4.77 and Less Than 7.73 Consider That data Only. Remove Remaining data.

		RM		RM_Zsco
0		6.575		0.413263
1		6.421		0.194082
2		7.185		1.28 14416
...		...	...	...
504		6.794		0.724955
505		6.030		-0.362408

476 x 15 columns

∴ From 506 records, Removed 30 records

That Means 476 Having (or) there.

⇒ it's about 6% of the data is Removed

⇒ it is only in One Variabe (1 column) = 6%.

⇒ If Total Variable (1,2,3,4..... all columns (14) → [10] continous Data) → Remove → May Delete 100 rows ↓ 20% of Data

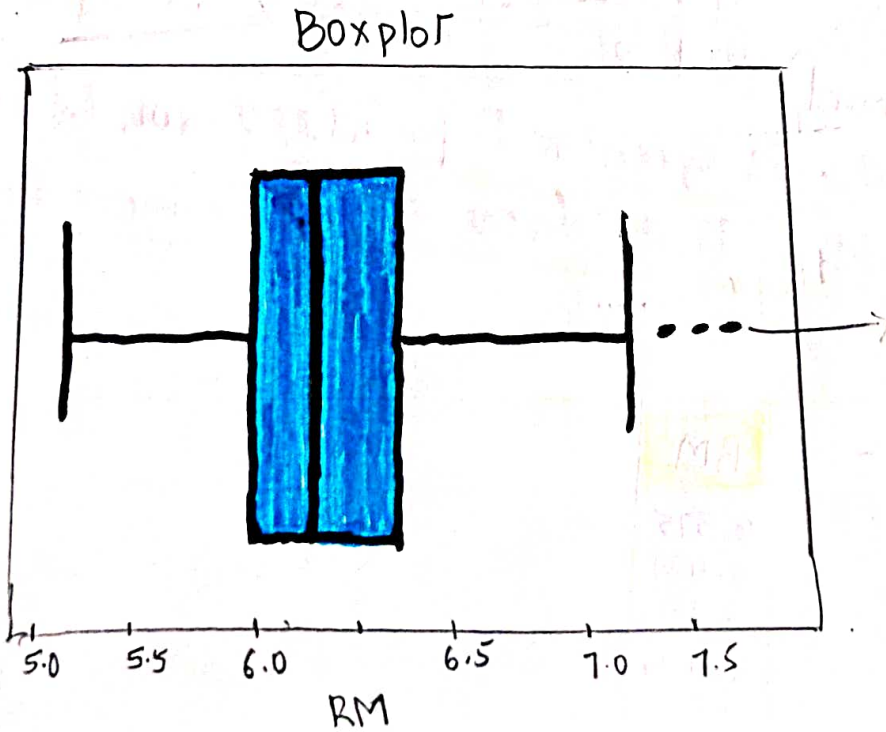
So, In real Life, we don't use outliers "Remove"

Even, Though, If we want Remove, still it contains Outliers.

⇒ Outliers in The Trimmed dataset

```
# sns.boxplot (boston_trimmed . RM)
plt. title ("Boxplot")
plt. show ().
```

Out



still, we have Outliers, after Removing The Outlier records.

Q:- Why ? it showing Outliers after Removing records of outliers

A:- When we remove datapoints From Our dataset, all the Parameters of The distribution are re-calculated.

For 476 records,

Again, it's calculating (Q_1, Q_3, IQR)

for 506

$Q_1 = 4.77$, Q_1 = New Value (again)

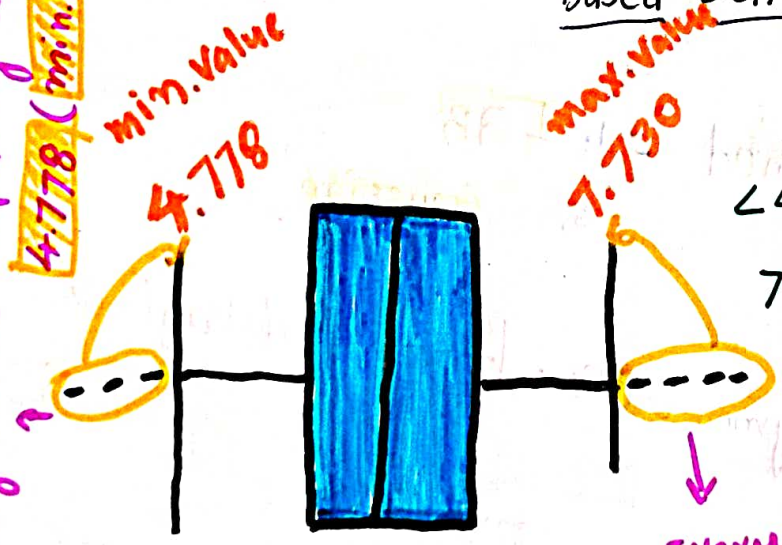
$Q_3 = 7.730$, Q_3 = New value (again)

* Therefore, In The new-trimmed variables values that before were not considered Outliers.

That's why, it showing new Outliers after removing some records also.

option (2). **Replace**

Replace with Upper Limit & Lower Limit (Calculated Based IQR)



$< 4.778 = Q_1 - [IQR * 1.5]$

$> 7.730 = Q_3 + [IQR * 1.5]$

Every Outiler Replacing with 7.730 (max.value)

⇒ More DeTailness

Replacing with Q_1, Q_3 Values

	CRIM	.....	RM	..... RAD
97		More Than 7.730, so, They are outliers. Reduce with Q_3 = 7.730	8.069 7.730	
98			7.820 7.730	
162			7.802 7.730	
163			8.375 7.730	

Q_3 - IQR * 1.5 < [4.778] (Lower_value) → less Than 4.778 = outlier

Q_1 IQR * 1.5 > [7.730] (Upper_value) → greater Than 7.730 = outlier

* Winsorizer

PIP install feature_engine

Winsorzier

From feature_engine.Outliers (Module) Import Winsozier Function

(Library)

In Winsozier Function (capping method argument)

* capping method = 1. IQR
2. "Gaussian"

In feature_engine.Outliers Import ArbitaryOutlierCapper Function

(Library) (module)

We can give our own Lower, upper Values.

1. By IQR method

In Depth of IQR.

Firstly,

 # We Calculate Q_1 Value

```
# Q1 = boston["RM"].quantile(0.25)
```

Out :- 5.855

 # We calculate Q_3 value.

```
# Q3 = boston["RM"].quantile(0.75)
```

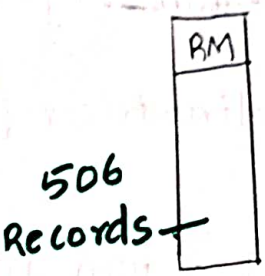
Out :- 6.623

 # IQR Value

```
# IQR = Q3 - Q1
```

IQR

Out 0.737



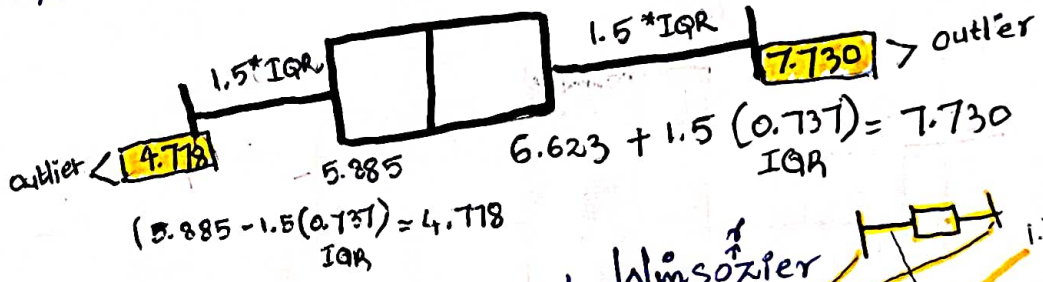
$\therefore Q_1 = 5.885$

$Q_3 = 6.623$

$IQR = Q_3 - Q_1$

$= 0.737$

Distance between Q_3 and Q_1



CODE :-

```
# From feature_engine.Outliers Import Winsozier
# Win = Winsozier(capping_method="iqr", tail='both', fold=1.5, Variables=["RM"])
# boston_t = Win.fit_transform(boston[["RM"]])
# Print(win.left_tail_caps_, Win.right_tail_caps_)
# Sns.boxplot(boston_t.RM)
# Plt.title("boxplot")
# Plt.show()
```

Win → Random name

store in → boston_t

boston[["RM"]] → original data

4.778 ← win.left_tail_caps_

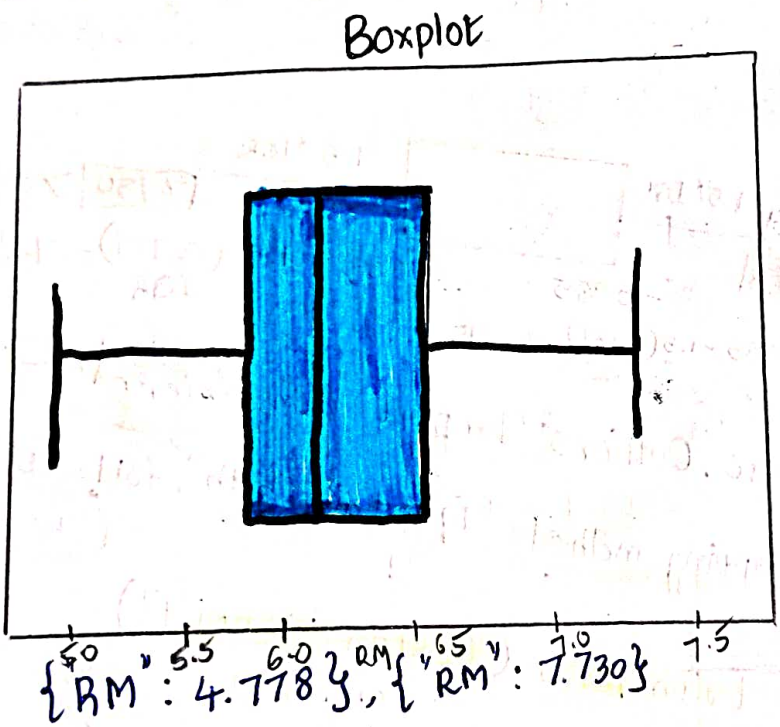
7.730 ← Win.right_tail_caps_

Edited (Replaced IQR).RM

again

```
From feature_engine.Outliers import Winsorizer

win = Winsorizer (capping_method = "iqr", fold = 1.5, Tail = "both",
                  Variables = ["RM"])
boston_t = win.fit_transform (boston [["RM"]]

Print (win.left_tail_caps_ , win.right_tail_caps_)

Sns.boxplot (boston_t.RM)
plt.title ("boxplot")
plt.show()
```

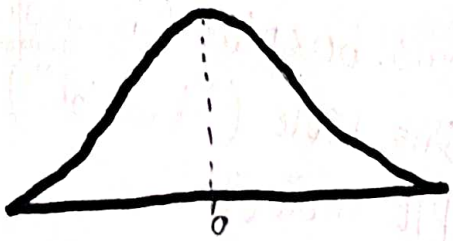
Out



2) Replacing using Winsorizer (min, max, automatically Taken by "Gaussian method")

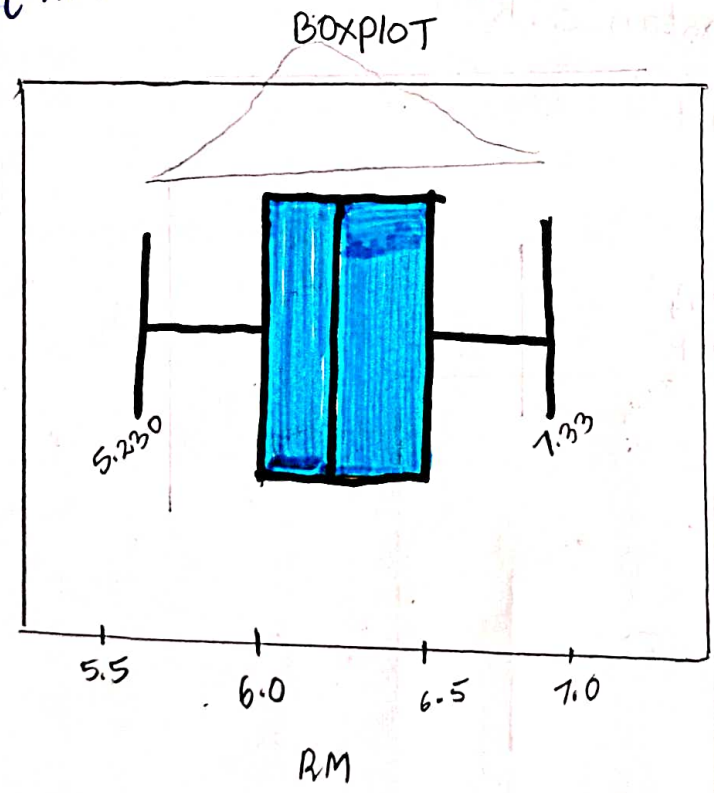
Gaussian method = "Normal Distribution method"

same



```
# from feature_engine.Outliers import Winsorizer

# win = Winsozier (capping method = "gaussian", tail = "both",
                   fold = 1.5, variables = ['RM'])

boston_t = win.fit_transform (boston[["RM"]])

print (win.left_tail_caps_, win.right_tail_caps)

sns.boxplot (boston_t, RM)
plt.title ("Boxplot")
plt.show()
```

Out {'RM': 5.230} {"RM": 7.33}



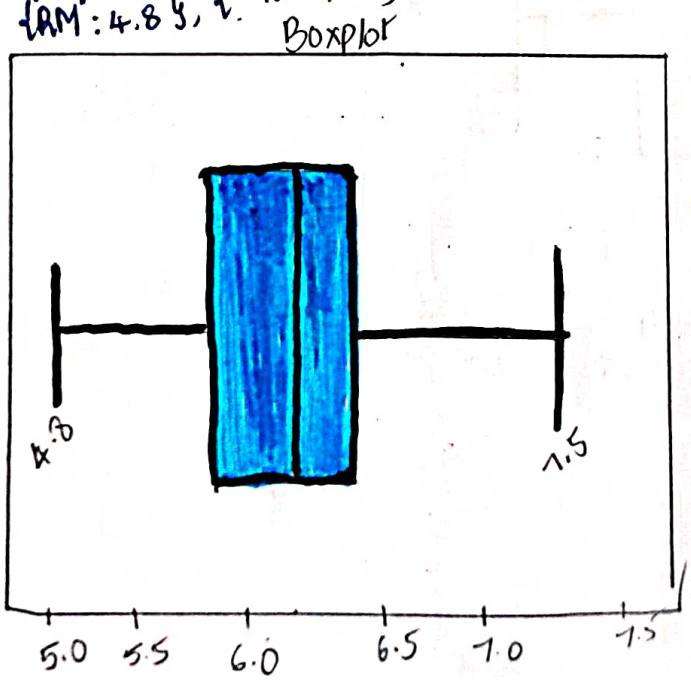
3. Replace arbitary Outlier Capper (The min, max, by user)

we get the min and max values based on "Domain Expert"
(or) by our Own "Research"

```
# From feature_engine.Outliers Import Arbitrary Outlier Capper

Capper = ArbitraryOutlierCapper (max_capping_dict = {"RM": 7.5},
                                 min_capping_dict = {"RM": 4.8})

boston_c = Capper.fit_transform (boston[["RM"]])

Print (capper.right_tail_caps_, Capper.left_tail_caps_)

Sns.boxplot (boston_c.RM)
plt.title ("Boxplot")
plt.show()
```

Out

{'RM': 4.8}, {'RM': 7.5}



3/4/22 5:20 pm