

Trimmed Mean

Let's understand what trimmed mean is and what is the importance of it, Trimmed mean actually finds out the average from the given data points by excluding extreme values, as in smallest and largest numbers in the data, how many of these extreme values will be removed is an argument given by us only. Let's take an example

```
from scipy import stats
import pandas as pd
import numpy as np
```

```
nifty = pd.read_csv('data.csv')
nifty.head(2)
```

	Date	Open	High	Low	Close	Shares Traded
0	03-Oct-2022	17102.10	17114.65	16855.55	16887.35	285850587
1	04-Oct-2022	17147.45	17287.30	17117.30	17274.30	231601639

	Turnover (Rs. Cr)
0	21860.54
1	18480.43

```
print(f'the normal mean/average value of shares traded column is {nifty["Shares Traded"].mean()}')
```

```
the normal mean/average value of shares traded column is
257471846.89655173
```

```
# first trimmed mean function looks into the length of the array
trim_len = len(nifty['Shares Traded'])
print(f"the length of array here is {trim_len}")
```

```
the length of array here is 29
```

```
# The next argument comes deciding what percentage of data we want
trimmed
```

```
# let's consider 20% as of now
```

```
trim_per = 0.20
```

```
# let's calculate the number of elements as 20% of our data
```

```
trim_len_20_per = round(trim_len*trim_per)
```

```
print(f'the number of elements after calculating the percentage is {trim_len_20_per}')
```

```
the number of elements after calculating the percentage is 6
```

```
# The number of elements to be removed from top and bottom extreme of
the data
```

```
# Is actually half of the 20% data that we calculated
```

```

ele_to_remove_from_top_nd_bottom = round(int(trim_len_20_per/2))
print(f'from top and bottom extreme we will remove
{ele_to_remove_from_top_nd_bottom} number of data')

```

from top and bottom extreme we will remove 3 number of data

```

# Let's find the top and bottom 2 extreme data points

```

```

s_3 = nifty['Shares Traded'].nsmallest(3)
l_3 = nifty['Shares Traded'].nlargest(3)
print(f'the smallest three data points are {s_3}')
print(f'the largest three data points are {l_3}')

```

```

the smallest three data points are 14      45029477

```

```

11      210480869

```

```

9       212223419

```

```

Name: Shares Traded, dtype: int64

```

```

the largest three data points are 26      378469072

```

```

19      349924808

```

```

16      324649088

```

```

Name: Shares Traded, dtype: int64

```

```

# Let's drop these records

```

```

update_nifty = nifty.drop([14, 11, 9, 26, 19, 16], axis=0)

```

```

update_nifty.head(2)

```

	Date	Open	High	Low	Close	Shares Traded
0	03-Oct-2022	17102.10	17114.65	16855.55	16887.35	285850587
1	04-Oct-2022	17147.45	17287.30	17117.30	17274.30	231601639

	Turnover (Rs. Cr)
0	21860.54
1	18480.43

```

print(f'the normal mean/average value of shares traded column is
{nifty["Shares Traded"].mean()}')
print(f'the trimmed mean value of shares traded column is
{round(update_nifty["Shares Traded"].mean())}')

```

```

the normal mean/average value of shares traded column is
257471846.89655173

```

```

the trimmed mean value of shares traded column is 258517688

```