```
117 / 18
```

```
    6.5
```

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

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/130/original/sehwag.csv?1684996594 -O sehwag.csv
```

```
    --2023-08-17 17:04:30--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/130/original/sehwag.csv?1684
    Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.173, 108.157.172.176, 108.157.172.
    Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.173|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 18584 (18K) [text/plain]
    Saving to: ‘sehwag.csv’

    sehwag.csv          100%[===================>]  18.15K  --.-KB/s    in 0s

    2023-08-17 17:04:30 (71.7 MB/s) - ‘sehwag.csv’ saved [18584/18584]
```

```
sehwag = pd.read_csv("sehwag.csv")
sehwag
```

	Runs	Mins	BF	4s	6s	SR	Pos	Dismissal	Inns	Unnamed: 9	Opposition	Ground	Start Date	Unnamed: 13
0	1	5	2	0	0	50.00	7	lbw	1	NaN	v Pakistan	Mohali	1 Apr 1999	ODI # 1427
1	19	18	24	0	1	79.16	6	caught	1	NaN	v Zimbabwe	Rajkot	14 Dec 2000	ODI # 1660
2	58	62	54	8	0	107.40	6	bowled	1	NaN	v Australia	Bengaluru	25 Mar 2001	ODI # 1696
3	2	7	7	0	0	28.57	6	caught	2	NaN	v Zimbabwe	Bulawayo	27 Jun 2001	ODI # 1730
4	11	19	16	1	0	68.75	6	not out	2	NaN	v West Indies	Bulawayo	30 Jun 2001	ODI # 1731
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
240	15	21	15	2	0	100.00	2	caught	1	NaN	v Sri Lanka	Hambantota	24 Jul 2012	ODI # 3292
241	3	6	6	0	0	50.00	2	caught	2	NaN	v Sri Lanka	Colombo (RPS)	28 Jul 2012	ODI # 3293
242	34	46	29	6	0	117.24	2	caught	2	NaN	v Sri Lanka	Colombo (RPS)	31 Jul 2012	ODI # 3294
243	4	20	11	1	0	36.36	2	bowled	1	NaN	v Pakistan	Chennai	30 Dec 2012	ODI # 3314
244	31	70	43	3	0	72.09	2	lbw	2	NaN	v Pakistan	Kolkata	3 Jan 2013	ODI # 3315

245 rows × 14 columns

```
sehwag['Runs'].describe()
```

```
    count    245.000000
    mean      33.767347
    std       34.809419
    min        0.000000
    25%        8.000000
    50%       23.000000
    75%       46.000000
    max      219.000000
    Name: Runs, dtype: float64
```

```
## 25th percentile
p_25 = np.percentile(sehwag['Runs'], 25)
p_25
```

```
    8.0
```

```
## 75th percentile
p_75 = np.percentile(sehwag['Runs'], 75)
p_75
```

```
    46.0
```

```
## 50th percentile
p_50 = np.percentile(sehwag['Runs'], 50)
p_50
```

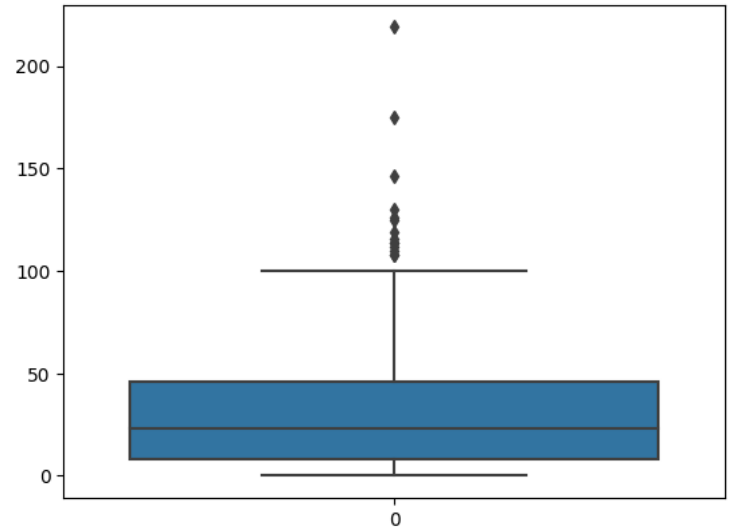
```
    23.0
```

```
## IQR
iqr_sehwag = p_75 - p_25
iqr_sehwag
```

```
38.0
```

```
sns.boxplot(data=sehwag['Runs'], orient="v")
```

```
<Axes: >
```



```
# upper limit = Q3 + 1.5 * IQR
upper = p_75 + (1.5 * iqr_sehwag)
upper
```

```
103.0
```

```
# lower limit = Q1 - 1.5 * IQR
lower = p_25 - (1.5 * iqr_sehwag)
lower
```

```
-49.0
```

```
outliers_seh = sehwag[sehwag['Runs'] > upper]
len(outliers_seh)
```

```
14
```

```
14 / 245
```

```
0.05714285714285714
```

COnclusion - Sehwag scored beyond upper limit 6% of the time.

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/131/original/dravid.csv?1684996749 -O dravid.csv
```

```
--2023-08-17 17:20:34--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/131/original/dravid.csv?1684
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.183, 108.157.172.173, 108.157.172.
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.183|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24177 (24K) [text/plain]
Saving to: ‘dravid.csv’

dravid.csv          100%[===================>]  23.61K  --.-KB/s    in 0.001s

2023-08-17 17:20:34 (18.3 MB/s) - ‘dravid.csv’ saved [24177/24177]
```

```
dravid = pd.read_csv("dravid.csv")
```

```
dravid
```

	Runs	Mins	BF	4s	6s	SR	Pos	Dismissal	Inns	Unnamed: 9	Opposition	Ground	Start Date	Unnamed: 13
0	3	-	4	0	0	75.00	4	caught	1	NaN	v Sri Lanka	Singapore	3 Apr 1996	ODI # 1089
1	4	-	7	0	0	57.14	4	run out	1	NaN	v Pakistan	Singapore	5 Apr 1996	ODI # 1091
2	3	-	5	0	0	60.00	5	caught	2	NaN	v Pakistan	Sharjah	12 Apr 1996	ODI # 1094
3	11	28	21	0	0	52.38	8	caught	2	NaN	v South Africa	Sharjah	14 Apr 1996	ODI # 1097
4	22	21	15	3	0	146.66	6	not out	1	NaN	v England	Manchester	26 May 1996	ODI # 1104
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
313	2	8	6	0	0	33.33	3	caught	1	NaN	v England	Chester-le-Street	3 Sep 2011	ODI # 3186
314	32	50	31	2	0	103.22	3	caught	1	NaN	v England	Southampton	6 Sep 2011	ODI # 3187
315	2	19	11	0	0	18.18	3	run out	1	NaN	v England	The Oval	9 Sep 2011	ODI # 3189

```
dravid['Runs'].describe()
```

```
count    318.000000
mean      34.242138
std       29.681822
min        0.000000
25%       10.000000
50%       26.000000
75%       54.000000
max      153.000000
Name: Runs, dtype: float64
```

```
upper = 54 + (1.5 * 44)
upper
```

```
120.0
```

```
len(dravid[dravid['Runs'] > upper])
```

```
3
```

```
3 / 318
```

```
0.009433962264150943
```

Conclusion: Dravid has outliers only 0.1% as compared to Sehwag who has 6% outliers.

Dravid is more consistent in terms of runs scored.

Height

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/126/original/weight-height.csv?1684995383 -O weight-h
```

```
--2023-08-17 17:27:20--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/035/126/original/weight-height.cs
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.10, 108.157.172.173, 108.157.172.1
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 428120 (418K) [text/plain]
Saving to: ‘weight-height.csv’

weight-height.csv   100%[===================>] 418.09K  --.-KB/s    in 0.07s

2023-08-17 17:27:20 (5.63 MB/s) - ‘weight-height.csv’ saved [428120/428120]
```

```
df = pd.read_csv("weight-height.csv")
```

```
df
```

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801
...	...	...	...
9995	Female	66.172652	136.777454

```
df.describe()
```

	Height	Weight
count	10000.000000	10000.000000
mean	66.367560	161.440357
std	3.847528	32.108439
min	54.263133	64.700127
25%	63.505620	135.818051
50%	66.318070	161.212928
75%	69.174262	187.169525
max	78.998742	269.989699

```
len(df[df['Height'] <= 63.505620])
```

```
2500
```

```
len(df[df['Height'] <= 66.318070])
```

```
5000
```

```
len(df[df['Height'] <= 69.174262])
```

```
7500
```

```
min_height = df['Height'].min()
```

```
min_height
```

```
54.2631333250971
```

CDF - describes the probability that a random variable takes on a value less than or equal to a given value.

```
# CDF
x_values = np.linspace(50,80, 100)
y_values = []
```

```
x_values
```

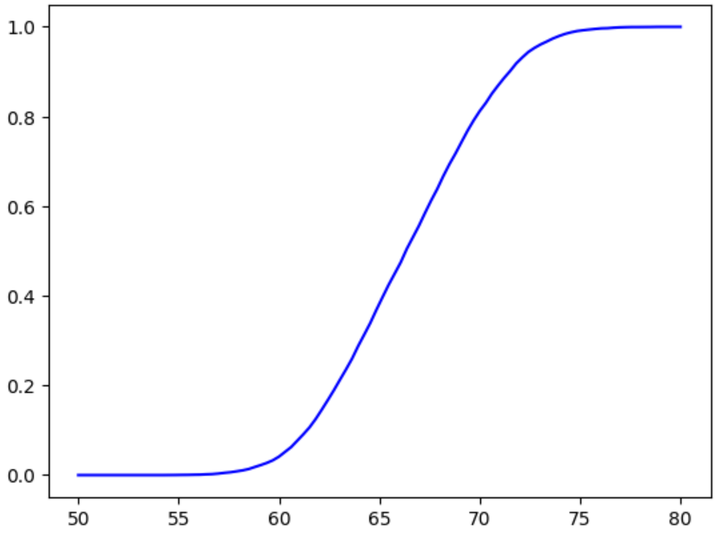
```
array([50.        , 50.3030303 , 50.60606061, 50.90909091, 51.21212121,
       51.51515152, 51.81818182, 52.12121212, 52.42424242, 52.72727273,
       53.03030303, 53.33333333, 53.63636364, 53.93939394, 54.24242424,
       54.54545455, 54.84848485, 55.15151515, 55.45454545, 55.75757576,
       56.06060606, 56.36363636, 56.66666667, 56.96969697, 57.27272727,
       57.57575758, 57.87878788, 58.18181818, 58.48484848, 58.78787879,
       59.09090909, 59.39393939, 59.6969697 , 60.        , 60.3030303 ,
       60.60606061, 60.90909091, 61.21212121, 61.51515152, 61.81818182,
       62.12121212, 62.42424242, 62.72727273, 63.03030303, 63.33333333,
       63.63636364, 63.93939394, 64.24242424, 64.54545455, 64.84848485,
       65.15151515, 65.45454545, 65.75757576, 66.06060606, 66.36363636,
       66.66666667, 66.96969697, 67.27272727, 67.57575758, 67.87878788,
       68.18181818, 68.48484848, 68.78787879, 69.09090909, 69.39393939,
       69.6969697 , 70.        , 70.3030303 , 70.60606061, 70.90909091,
       71.21212121, 71.51515152, 71.81818182, 72.12121212, 72.42424242,
       72.72727273, 73.03030303, 73.33333333, 73.63636364, 73.93939394,
       74.24242424, 74.54545455, 74.84848485, 75.15151515, 75.45454545,
       75.75757576, 76.06060606, 76.36363636, 76.66666667, 76.96969697,
       77.27272727, 77.57575758, 77.87878788, 78.18181818, 78.48484848,
       78.78787879, 79.09090909, 79.39393939, 79.6969697 , 80.        ])
```

```
total = 10000

for x in x_values:
    people_shorter_than_x = df[df['Height'] < x]
    num_people_shorter_than_x = len(people_shorter_than_x)
    fraction_people_shorter_than_x = num_people_shorter_than_x / total
    y_values.append(fraction_people_shorter_than_x)


plt.plot(x_values, y_values, c='b')
```

```
[<matplotlib.lines.Line2D at 0x788022a5f580>]
```


1.0
0.8
0.6
0.4
0.2
0.0
50
55
60
65
70
75
80