```
import pandas as pd
```

```
a=pd.read_csv("/content/drive/MyDrive/DSBDA/NHANES Weight and Height.csv")
```

```
a.head(5)
```

	Unnamed: 0	Weight (kg)	Height (cm)	BMI(kg/m**2)
0	0	97.1	160.2	37.8
1	1	98.8	182.3	29.7
2	2	74.3	184.2	21.9
3	3	103.7	NaN	30.2
4	4	83.3	177.1	26.6

Next steps: Generate code with a View recommended plots

```
a.columns
```

```
Index(['Unnamed: 0', 'Weight (kg)', 'Height (cm)', 'BMI(kg/m**2)'], dtype='object')
```

```
a.shape
```

```
(8388, 4)
```

```
a=a.drop("Weight (kg)",axis=1)
```

```
a.shape
```

```
(8388, 3)
```

```
a.isnull().sum()
```

```
Unnamed: 0       0
Height (cm)     10
BMI(kg/m**2)     0
dtype: int64
```

```
a['Height (cm)']=a['Height (cm)'].fillna(a['Height (cm)'].mean())
```

```
a.isnull().sum()
```

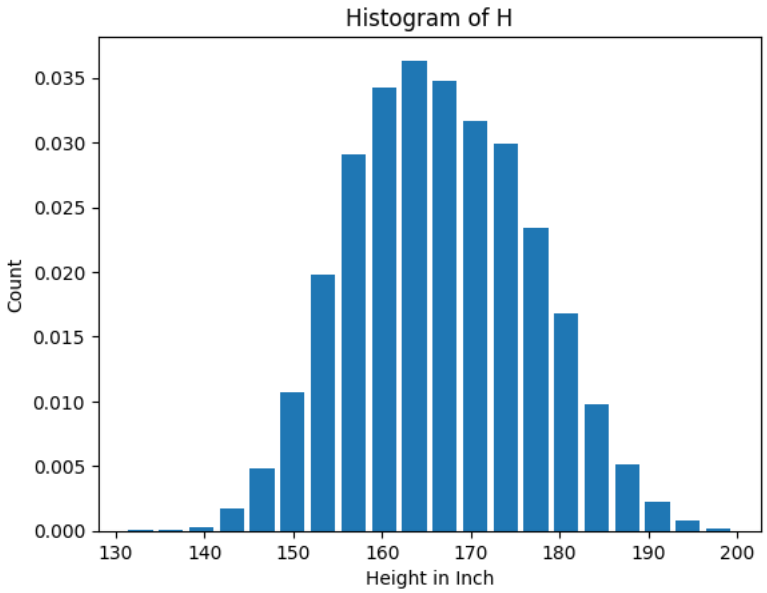
```
Unnamed: 0      0
Height (cm)     0
BMI(kg/m**2)    0
dtype: int64
```

```
import matplotlib.pyplot as plt
```

```
from scipy.stats import norm
```

```
plt.hist(a['Height (cm)'], bins=20, rwidth=0.8, density=True)
plt.title("Histogram of H")
plt.xlabel("Height in Inch")
plt.ylabel("Count")
```

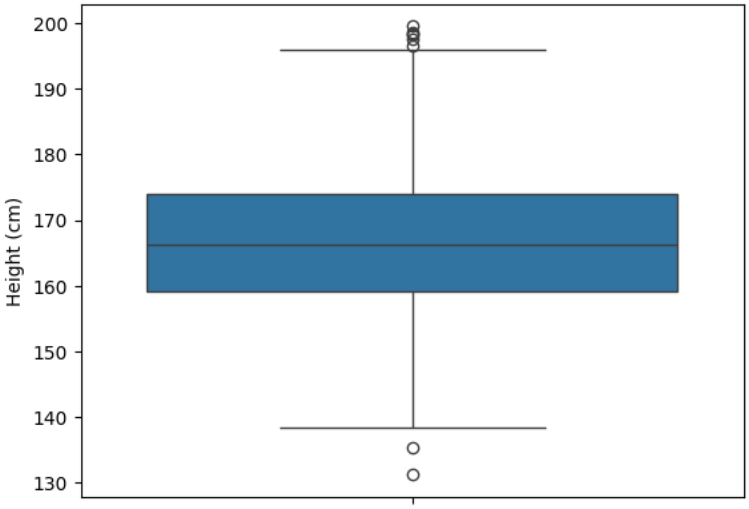
```
Text(0, 0.5, 'Count')
```




```
import seaborn as sb
```

```
sb.boxplot(a['Height (cm)'])
```

```
<Axes: ylabel='Height (cm)'>
```




```
a.describe()
```

	Unnamed: 0	Height (cm)	BMI(kg/m**2)
count	8388.000000	8388.000000	8388.000000
mean	4193.500000	166.644020	30.034859
std	2421.551362	10.070006	7.565376
min	0.000000	131.100000	14.200000
25%	2096.750000	159.100000	24.900000
50%	4193.500000	166.300000	28.800000
75%	6290.250000	173.900000	33.800000
max	8387.000000	199.600000	92.300000

```
########  Z-Score  ###########
#upper limit
ul=a['Height (cm)'].mean()+3*a['Height (cm)'].std()
```

```
ll=a['Height (cm)'].mean()-3*a['Height (cm)'].std()
```

```
print(ul)
```

```
196.85403761950448
```

```
print(ll)
```

```
136.43400248553252
```

```
a.loc[(a['Height (cm)']>=ul) | (a['Height (cm)']<=ll)]
```

	Unnamed: 0	Height (cm)	BMI(kg/m**2)
60	60	198.7	27.1
1906	1906	135.3	29.4
2165	2165	131.1	35.1
3379	3379	197.7	24.9
4026	4026	198.4	23.8
5815	5815	198.3	27.7
7576	7576	199.6	29.5

```
#trimming
a1=a.loc[(a['Height (cm)']<=ul) & (a['Height (cm)']>=ll)]
```

```
a1
```

	Unnamed: 0	Height (cm)	BMI(kg/m**2)
0	0	160.20000	37.8
1	1	182.30000	29.7
2	2	184.20000	21.9
3	3	166.64402	30.2
4	4	177.10000	26.6
...	...	...	...
8383	8383	178.80000	29.5
8384	8384	147.80000	37.9
8385	8385	168.70000	38.2
8386	8386	176.40000	25.5
8387	8387	167.50000	21.3

8381 rows × 3 columns

Next steps: Generate code with a1 View recommended plots

```
print("Before Trim :",len(a))
```

```
Before Trim : 8388
```

```
print("After Trim:",len(a1))
```

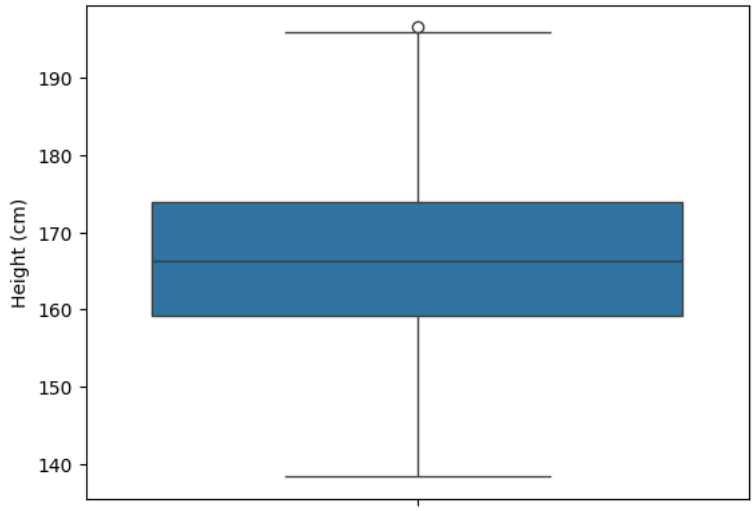
```
After Trim: 8381
```

```
print("No of outliers :",len(a)-len(a1))
```

```
No of outliers : 7
```

```
sb.boxplot(a1['Height (cm)'])
```

```
<Axes: ylabel='Height (cm)'>
```



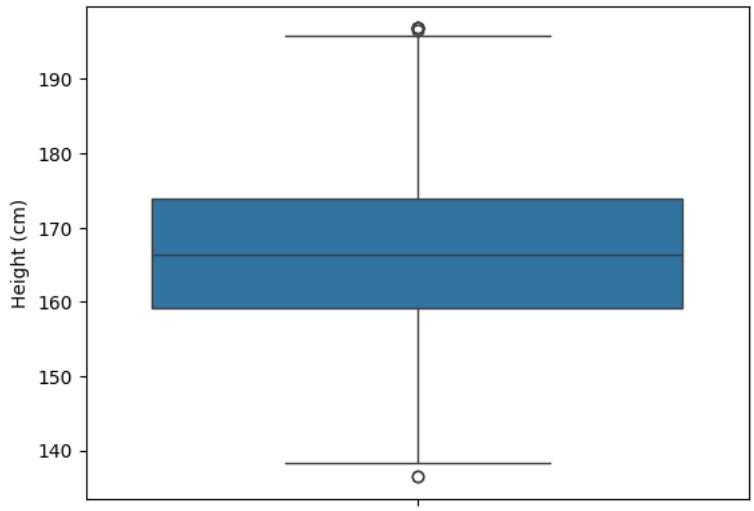

```
#capping
a2=a.copy()

a2.loc[(a2['Height (cm)']>=ul),'Height (cm)']=ul

a2.loc[(a2['Height (cm)']<=ll),'Height (cm)']=ll

sb.boxplot(a2['Height (cm)'])
```

```
<Axes: ylabel='Height (cm)'>
```




```
print("Before Trim :",len(a))
```

```
Before Trim : 8388
```

```
print("After Trim :",len(a2))
```

```
After Trim : 8388
```

```
#####  IQR  #####
import numpy as np

Q1 = np.percentile(a['Height (cm)'], 25)
Q2 = np.percentile(a['Height (cm)'], 50)  # Median
Q3 = np.percentile(a['Height (cm)'], 75)

IQR = Q3 - Q1

IQR
```

```
14.800000000000011
```

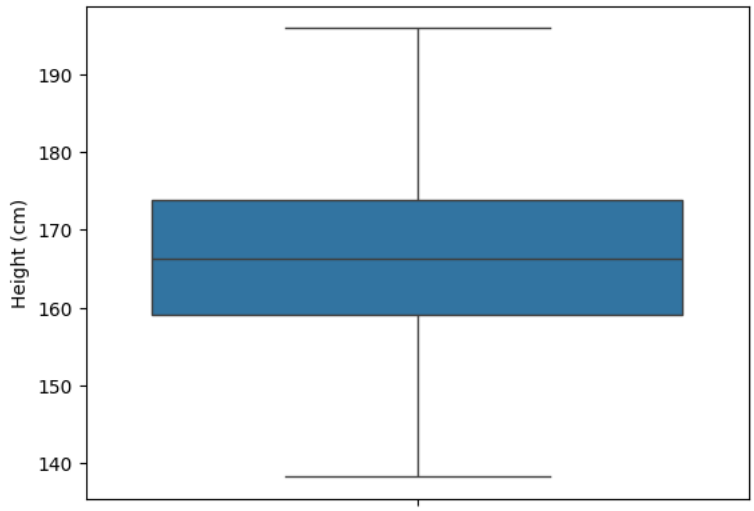
```
lower_limit=Q1-1.5*IQR
uper_limit=Q3+1.5*IQR


#trimming
aa=a.loc[(a['Height (cm)']<=uper_limit) & (a['Height (cm)']>=lower_limit)]



print("After Trimming")
sb.boxplot(aa['Height (cm)'])
```

```
After Trimming
<Axes: ylabel='Height (cm)'>
```




```
#capping
a.loc[(a['Height (cm)']>=uper_limit),'Height (cm)']=uper_limit
a.loc[(a['Height (cm)']<=lower_limit),'Height (cm)']=lower_limit


sb.boxplot(a['Height (cm)'])
```

```
<Axes: ylabel='Height (cm)'>
```