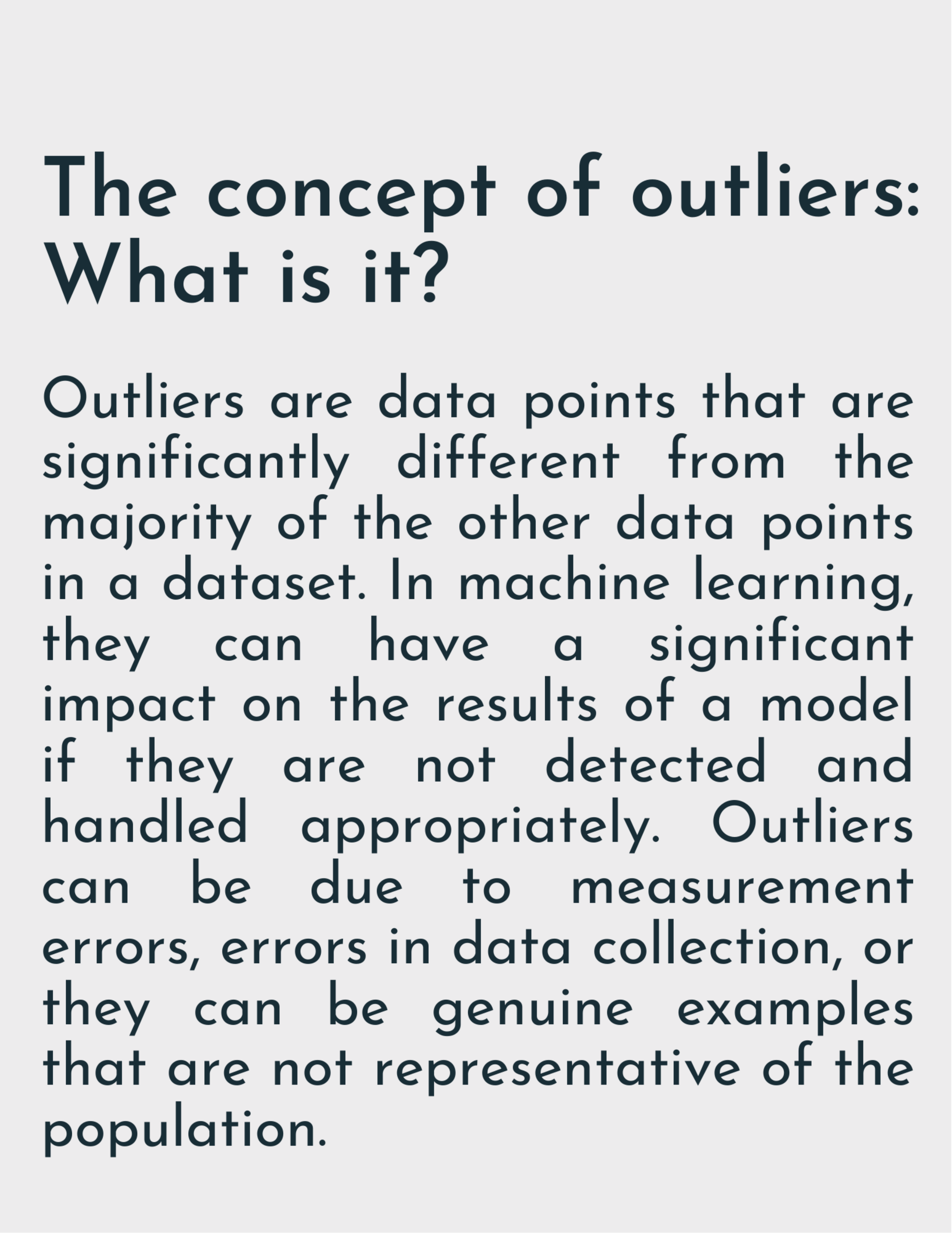
Feature Engineering 101

Topic - 10

Outliers

Outliers Detection Method

1. Z-Score
2. IQR
3. Winsorization or Percentile



The concept of outliers: What is it?

Outliers are data points that are significantly different from the majority of the other data points in a dataset. In machine learning, they can have a significant impact on the results of a model if they are not detected and handled appropriately. Outliers can be due to measurement errors, errors in data collection, or they can be genuine examples that are not representative of the population.

Z-Score

In [1]:

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

In [2]:

```
df = pd.read_csv('placement.csv')
```

In [3]:

```
df.shape
```

Out[3]:

```
(1000, 3)
```

In [4]:

```
df.sample(5)
```

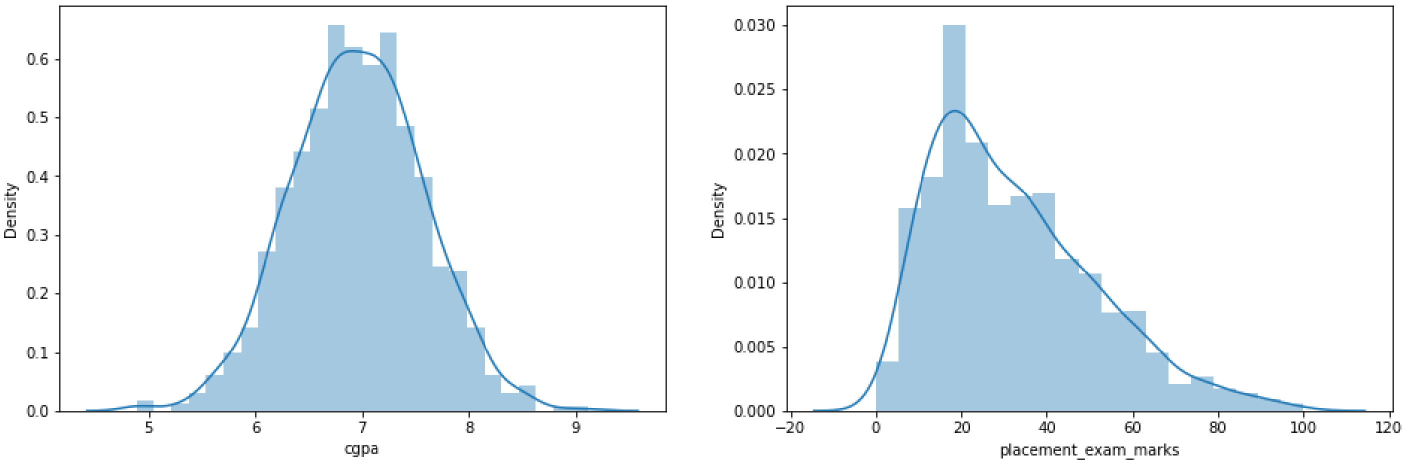
Out[4]:

	cgpa	placement_exam_marks	placed
719	7.17	26.0	0
457	6.58	20.0	0
542	7.06	22.0	0
733	7.07	10.0	0
770	7.33	67.0	1

In [5]:

```
plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df['cgpa'])

plt.subplot(1,2,2)
sns.distplot(df['placement_exam_marks'])

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [6]:

```
df['placement_exam_marks'].skew()
```

Out[6]:

```
0.8356419499466834
```

In [7]:

```
print("Mean value of cgpa",df['cgpa'].mean())
print("Std value of cgpa",df['cgpa'].std())
print("Min value of cgpa",df['cgpa'].min())
print("Max value of cgpa",df['cgpa'].max())
```

```
Mean value of cgpa 6.96124000000001
Std value of cgpa 0.6158978751323894
Min value of cgpa 4.89
Max value of cgpa 9.12
```

In [8]:

```
# Finding the boundary values
print("Highest allowed",df['cgpa'].mean() + 3*df['cgpa'].std())
print("Lowest allowed",df['cgpa'].mean() - 3*df['cgpa'].std())
```

```
Highest allowed 8.808933625397177
Lowest allowed 5.113546374602842
```

In [9]:

```
# Finding the outliers
df[(df['cgpa'] > 8.80) | (df['cgpa'] < 5.11)]
```

Out[9]:

	cgpa	placement_exam_marks	placed
485	4.92	44.0	1
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
999	4.90	10.0	1

Trimming

In [10]:

```
# Trimming

new_df = df[(df['cgpa'] < 8.80) & (df['cgpa'] > 5.11)]
new_df
```

Out[10]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
991	7.04	57.0	0
992	6.26	12.0	0
993	6.73	21.0	1
994	6.48	63.0	0
998	8.62	46.0	1

995 rows × 3 columns

In [11]:

```
# Approach 2

# Calculating the Zscore

df['cgpa_zscore'] = (df['cgpa'] - df['cgpa'].mean())/df['cgpa'].std()
```

In [12]:

```
df.head()
```

Out[12]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371

In [13]:

```
df[df['cgpa_zscore'] > 3]
```

Out[13]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062

In [14]:

```
df[df['cgpa_zscore'] < -3]
```

Out[14]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
997	4.89	34.0	0	-3.362960
999	4.90	10.0	1	-3.346724

In [15]:

```
df[(df['cgpa_zscore'] > 3) | (df['cgpa_zscore'] < -3)]
```

Out[15]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062
997	4.89	34.0	0	-3.362960
999	4.90	10.0	1	-3.346724

In [16]:

```
# Trimming
new_df = df[(df['cgpa_zscore'] < 3) & (df['cgpa_zscore'] > -3)]
```

In [17]:

```
new_df
```

Out[17]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371
...	...	...	...	...
991	7.04	57.0	0	0.127878
992	6.26	12.0	0	-1.138565
993	6.73	21.0	1	-0.375452
994	6.48	63.0	0	-0.781363
998	8.62	46.0	1	2.693239

995 rows × 4 columns

In [18]:

```
new_df['cgpa'].describe()
```

Out[18]:

```
count    995.000000
mean       6.963357
std        0.600082
min        5.230000
25%        6.550000
50%        6.960000
75%        7.365000
max        8.620000
Name: cgpa, dtype: float64
```

Capping

In [19]:

```
upper_limit = df['cgpa'].mean() + 3*df['cgpa'].std()
lower_limit = df['cgpa'].mean() - 3*df['cgpa'].std()
```

In [20]:

```
upper_limit
```

Out[20]:

```
8.808933625397177
```

In [21]:

```
lower_limit
```

Out[21]:

```
5.113546374602842
```

In [22]:

```
#Capping fun
df['cgpa'] = np.where(
    df['cgpa']>upper_limit,
    upper_limit,
    np.where(
        df['cgpa']<lower_limit,
        lower_limit,
        df['cgpa']
    )
)
```

In [23]:

```
df.shape
```

Out[23]:

```
(1000, 4)
```

In [24]:

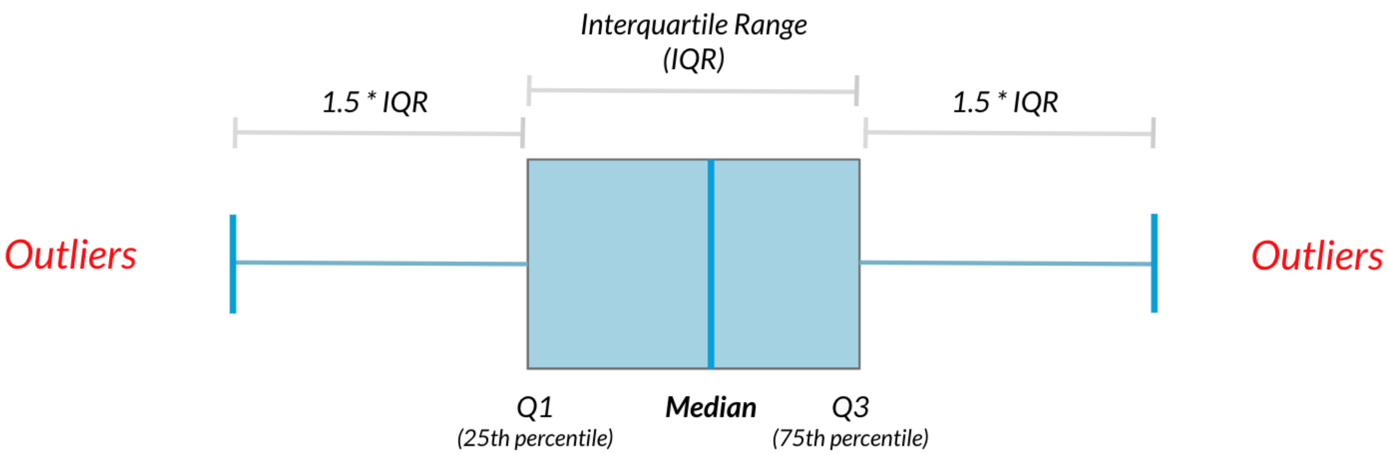
```
df['cgpa'].describe()
```

Out[24]:

```
count    1000.000000
mean        6.961499
std         0.612688
min         5.113546
25%         6.550000
50%         6.960000
75%         7.370000
max         8.808934
Name: cgpa, dtype: float64
```

In [ ]:

IQR (Inter-quartile range)



In [1]:

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

In [2]:

```
df = pd.read_csv('placement.csv')
```

In [3]:

```
df.head()
```

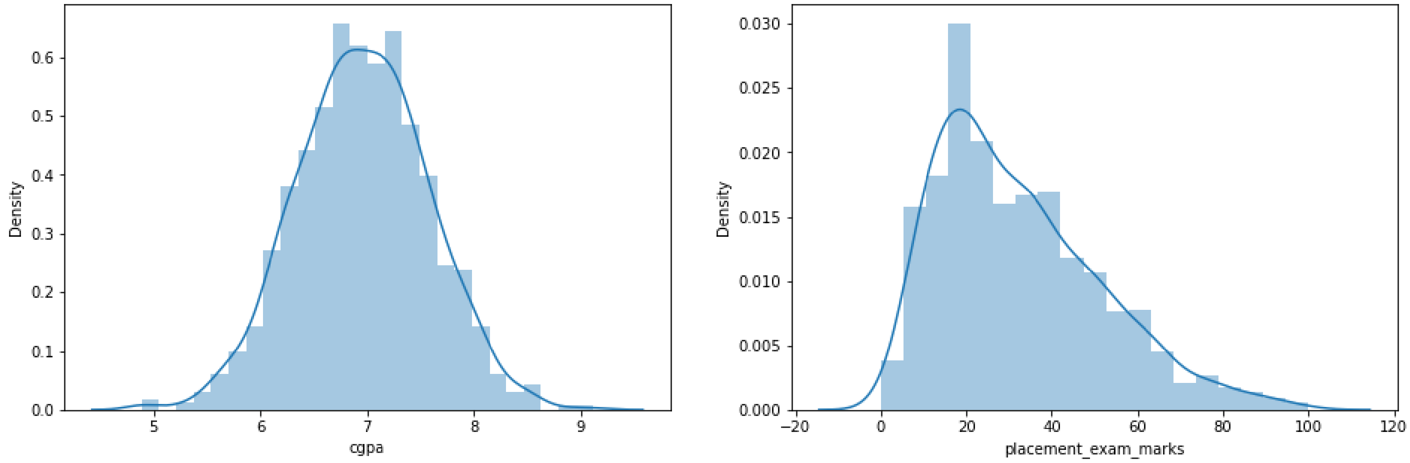
Out[3]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0

In [4]:

```
plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df['cgpa'])

plt.subplot(1,2,2)
sns.distplot(df['placement_exam_marks'])

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [5]:

```
df['placement_exam_marks'].describe()
```

Out[5]:

```
count    1000.000000
mean       32.225000
std        19.130822
min         0.000000
25%        17.000000
50%        28.000000
75%        44.000000
max       100.000000
Name: placement_exam_marks, dtype: float64
```

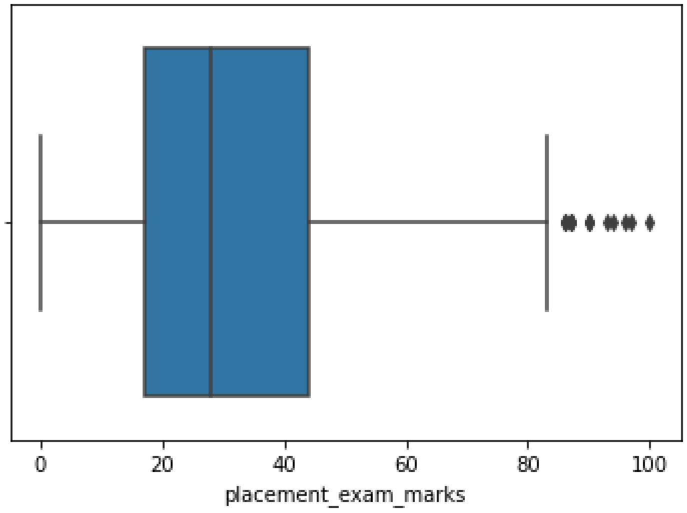
In [6]:

```
sns.boxplot(df['placement_exam_marks'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```

Out[6]:

```
<AxesSubplot:xlabel='placement_exam_marks'>
```



In [7]:

```
# Finding the IQR
percentile25 = df['placement_exam_marks'].quantile(0.25)
percentile75 = df['placement_exam_marks'].quantile(0.75)
```

In [8]:

```
percentile75
```

Out[8]:

```
44.0
```

In [9]:

```
iqr = percentile75 - percentile25
```

In [10]:

```
iqr
```

Out[10]:

```
27.0
```

In [11]:

```
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr
```

In [12]:

```
print("Upper limit",upper_limit)
print("Lower limit",lower_limit)
```

```
Upper limit 84.5
Lower limit -23.5
```

Finding Outliers

In [13]:

```
df[df['placement_exam_marks'] > upper_limit]
```

Out[13]:

	cgpa	placement_exam_marks	placed
9	7.75	94.0	1
40	6.60	86.0	1
61	7.51	86.0	0
134	6.33	93.0	0
162	7.80	90.0	0
283	7.09	87.0	0
290	8.38	87.0	0
311	6.97	87.0	1
324	6.64	90.0	0
630	6.56	96.0	1
685	6.05	87.0	1
730	6.14	90.0	1
771	7.31	86.0	1
846	6.99	97.0	0
917	5.95	100.0	0

In [14]:

```
df[df['placement_exam_marks'] < lower_limit]
```

Out[14]:

cgpa	placement_exam_marks	placed

Trimming

In [15]:

```
new_df = df[df['placement_exam_marks'] < upper_limit]
```

In [16]:

```
new_df.shape
```

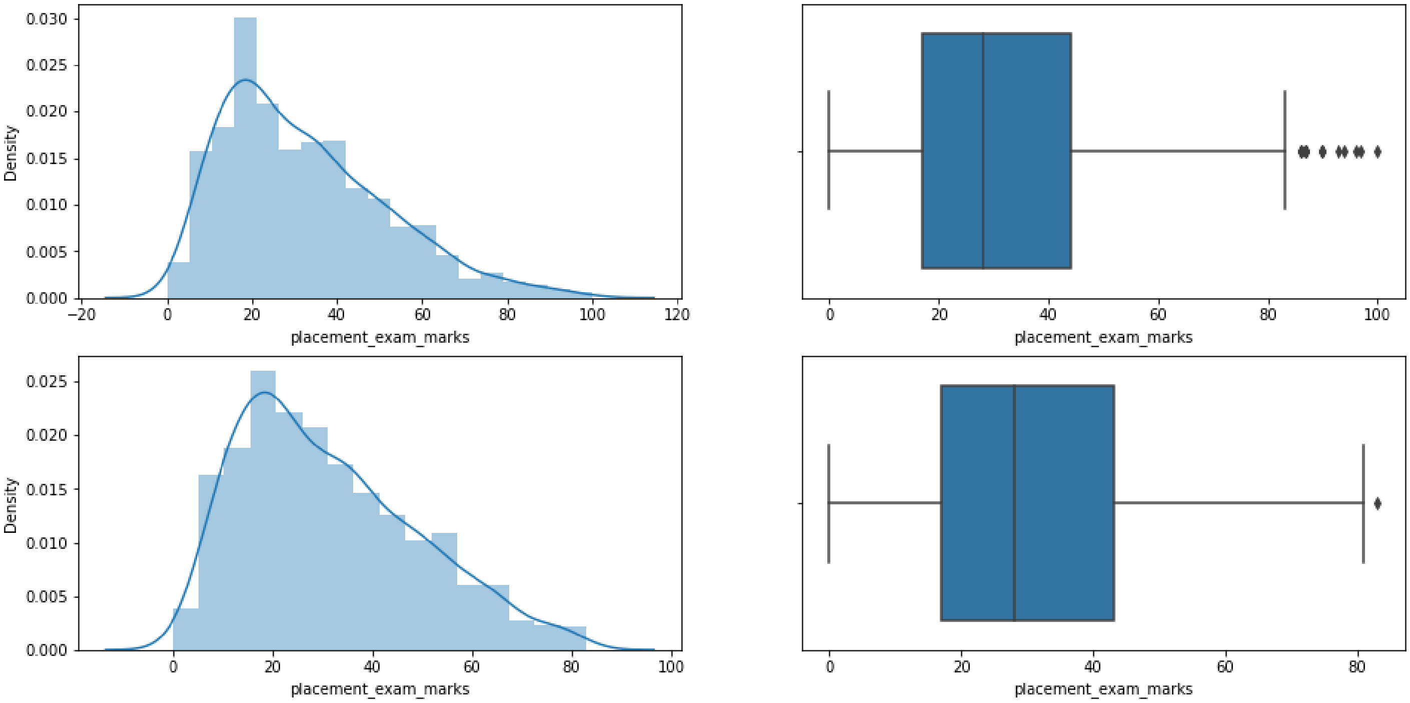
Out[16]:

```
(985, 3)
```

In [17]:

```
# Comparing

plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['placement_exam_marks'])

plt.subplot(2,2,2)
sns.boxplot(df['placement_exam_marks'])

plt.subplot(2,2,3)
sns.distplot(new_df['placement_exam_marks'])

plt.subplot(2,2,4)
sns.boxplot(new_df['placement_exam_marks'])

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



Capping

In [20]:

```
new_df_cap = df.copy()

new_df_cap['placement_exam_marks'] = np.where(
    new_df_cap['placement_exam_marks'] > upper_limit,
    upper_limit,
    np.where(
        new_df_cap['placement_exam_marks'] < lower_limit,
        lower_limit,
        new_df_cap['placement_exam_marks']
    )
)
```

In [22]:

```
new_df_cap.shape
```

Out[22]:

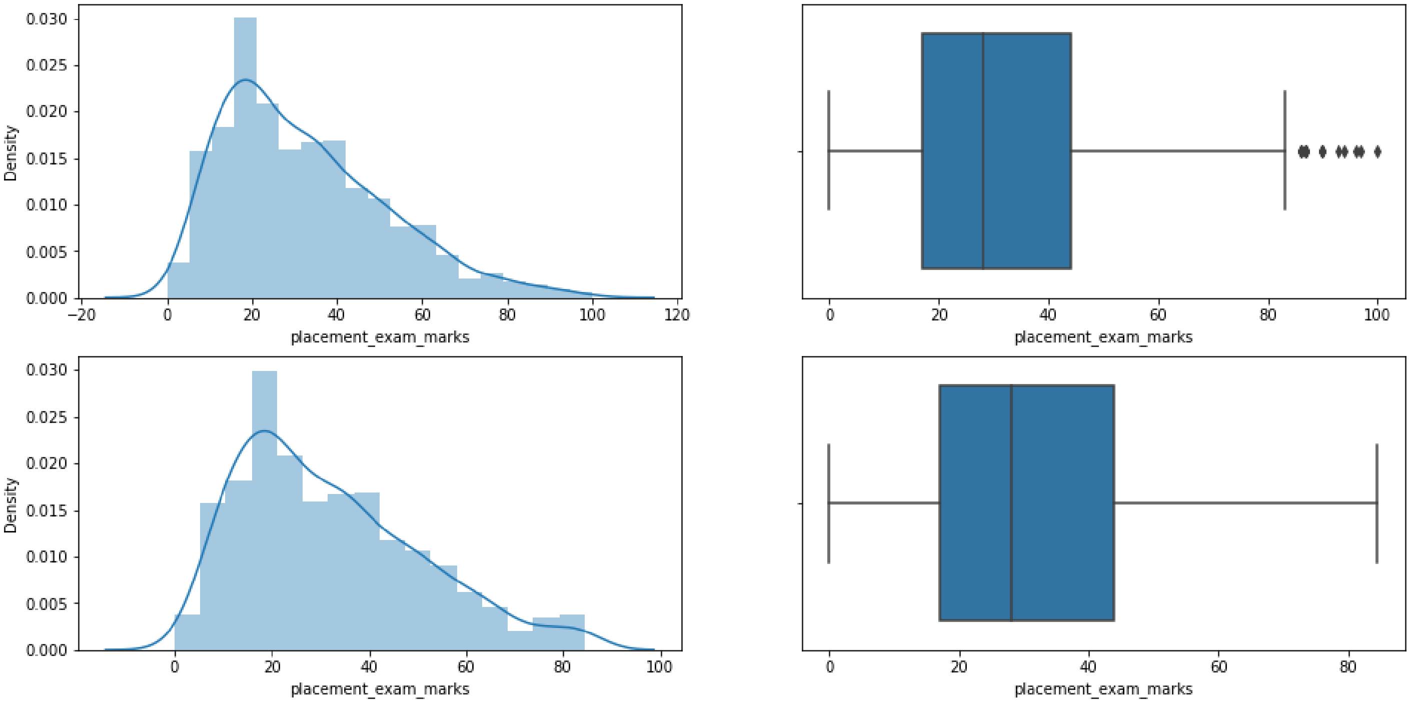
```
(1000, 3)
```

In [23]:

```
# Comparing

plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['placement_exam_marks'])

plt.subplot(2,2,2)
sns.boxplot(df['placement_exam_marks'])

plt.subplot(2,2,3)
sns.distplot(new_df_cap['placement_exam_marks'])

plt.subplot(2,2,4)
sns.boxplot(new_df_cap['placement_exam_marks'])

plt.show()
```

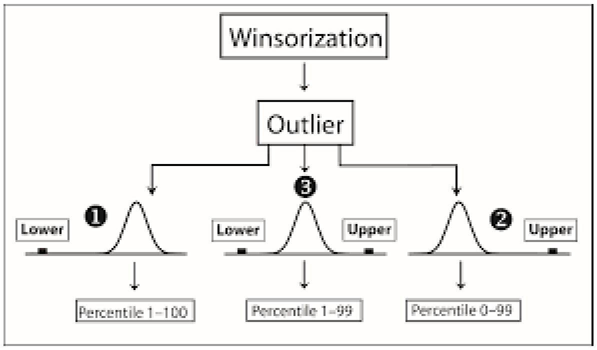
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
```

```
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



```
In [ ]:
```

Winsorization or Persentile



In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

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

In [3]:

```
df.head()
```

Out[3]:

	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

In [4]:

```
df.shape
```

Out[4]:

```
(10000, 3)
```

In [5]:

```
df['Height'].describe()
```

Out[5]:

```
count    10000.000000
mean        66.367560
std          3.847528
min         54.263133
25%         63.505620
50%         66.318070
75%         69.174262
max         78.998742
Name: Height, dtype: float64
```

In [6]:

```
import seaborn as sns
```

In [7]:

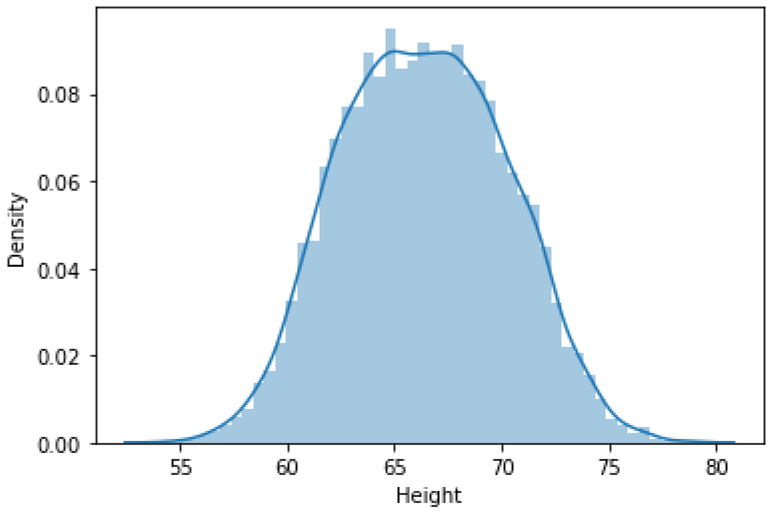
```
sns.distplot(df['Height'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
```

```
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[7]:

```
<AxesSubplot:xlabel='Height', ylabel='Density'>
```



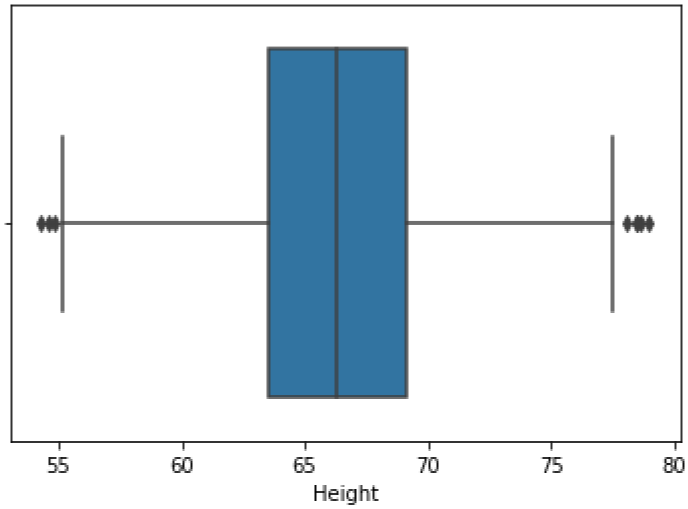
In [8]:

```
sns.boxplot(df['Height'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```

Out[8]:

```
<AxesSubplot:xlabel='Height'>
```



In [9]:

```
upper_limit = df['Height'].quantile(0.99)
upper_limit
```

Out[9]:

```
74.7857900583366
```

In [10]:

```
lower_limit = df['Height'].quantile(0.01)
lower_limit
```

Out[10]:

```
58.13441158671655
```

In [11]:

```
new_df = df[(df['Height'] <= 74.78) & (df['Height'] >= 58.13)]
```

```
In [12]:  new_df['Height'].describe()
```

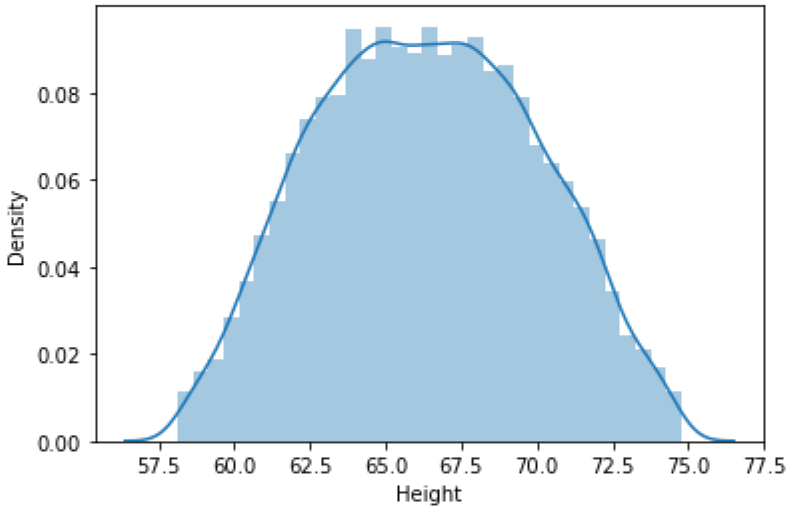
```
Out[12]: count    9799.000000
         mean       66.363507
         std         3.644267
         min        58.134496
         25%        63.577147
         50%        66.317899
         75%        69.119859
         max        74.767447
         Name: Height, dtype: float64
```

```
In [13]:  df['Height'].describe()
```

```
Out[13]: count    10000.000000
         mean        66.367560
         std          3.847528
         min         54.263133
         25%         63.505620
         50%         66.318070
         75%         69.174262
         max         78.998742
         Name: Height, dtype: float64
```

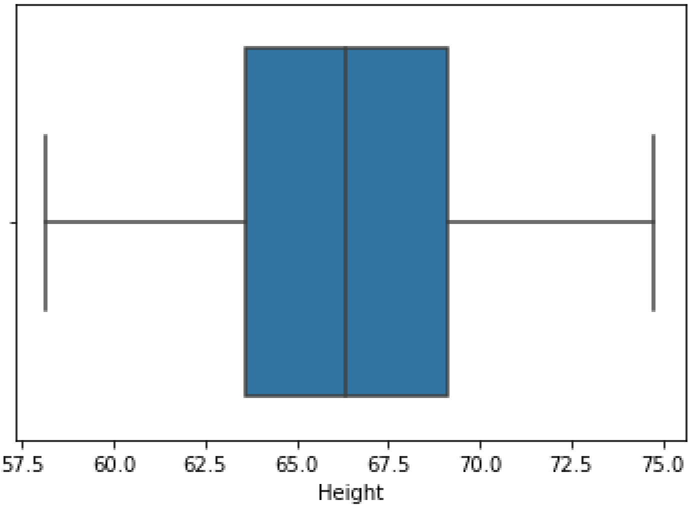
```
In [14]:  sns.distplot(new_df['Height'])
```

```
         C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
         distplot` is a deprecated function and will be removed in a future version. Please adapt y
         our code to use either `displot` (a figure-level function with similar flexibility) or `hi
         stplot` (an axes-level function for histograms).
           warnings.warn(msg, FutureWarning)
Out[14]: <AxesSubplot:xlabel='Height', ylabel='Density'>
```



```
In [15]:  sns.boxplot(new_df['Height'])
```

```
         C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
         the following variable as a keyword arg: x. From version 0.12, the only valid positional a
         rgument will be `data`, and passing other arguments without an explicit keyword will resul
         t in an error or misinterpretation.
           warnings.warn(
Out[15]: <AxesSubplot:xlabel='Height'>
```



In [16]:

```
# Capping --> Winsorization
df['Height'] = np.where(df['Height'] >= upper_limit,
        upper_limit,
        np.where(df['Height'] <= lower_limit,
        lower_limit,
        df['Height']))
```

In [17]:

```
df.shape
```

Out[17]:

```
(10000, 3)
```

In [18]:

```
df['Height'].describe()
```

Out[18]:

```
count    10000.000000
mean        66.366281
std          3.795717
min         58.134412
25%         63.505620
50%         66.318070
75%         69.174262
max         74.785790
Name: Height, dtype: float64
```

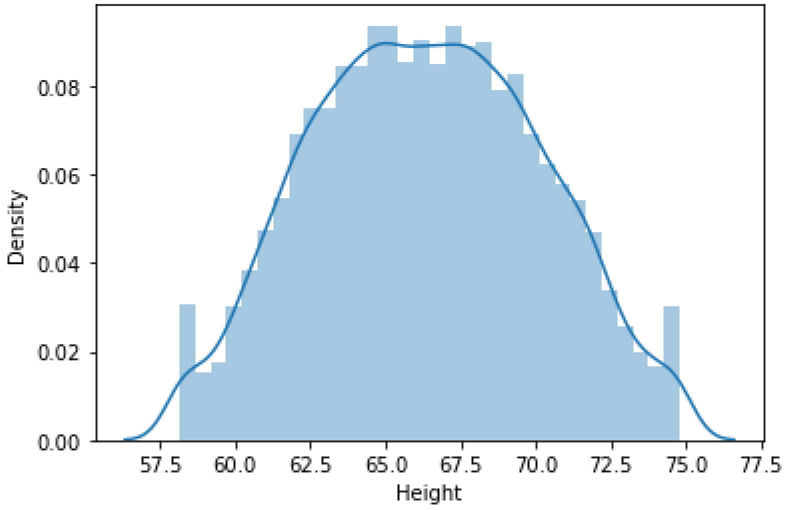
In [19]:

```
sns.distplot(df['Height'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[19]:

```
<AxesSubplot:xlabel='Height', ylabel='Density'>
```



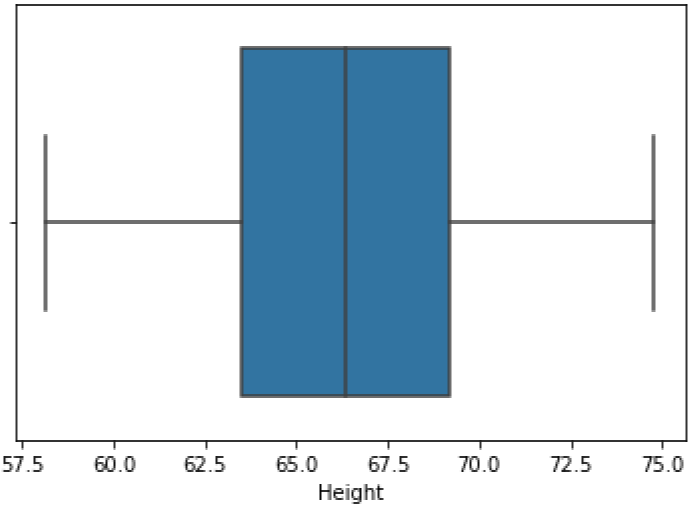
In [20]:

```
sns.boxplot(df['Height'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```

Out[20]:

```
<AxesSubplot:xlabel='Height'>
```



In [ ]: