campusx-official / 100-days-of-machine-learning Public

Code Issues 3 Pull requests Actions Projects Security Insights

main

100-days-of-machine-learning / day44-outlier-detection-using-percentiles / day44.ipynb

campusx-official Add files via upload History

1 contributor

553 lines (553 sloc) | 67.2 KB

In [4]:
```
import numpy as np
import pandas as pd
```

In [5]:
```
df = pd.read_csv('weight-height.csv')
```

In [6]:
```
df.head()
```

Out[6]:

	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 [7]:
```
df.shape
```

Out[7]:
```
(10000, 3)
```

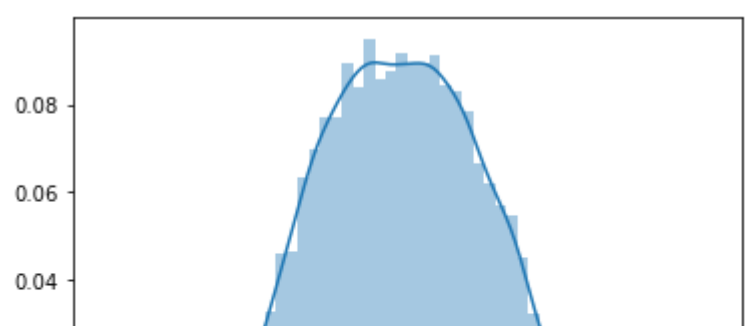
In [8]:
```
df['Height'].describe()
```

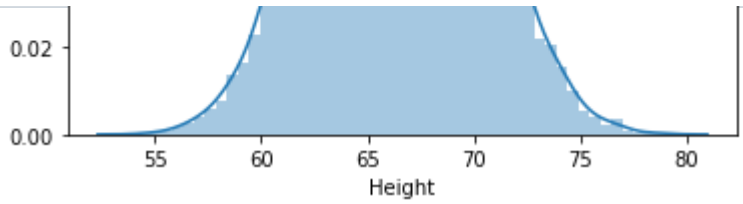
Out[8]:
```
count    10000.000000
mean        66.367560
std          3.847528
min         54.263133
25%         63.505620
50%         66.318070
75%         69.174262
max         78.998742
Name: Height, dtype: float64
```

In [9]:
```
import seaborn as sns
```

In [10]:
```
sns.distplot(df['Height'])
```

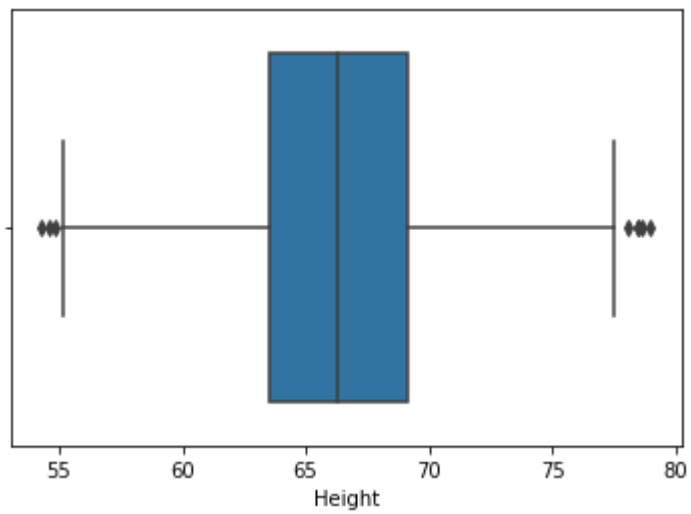
Out[10]:





In [11]:
```
sns.boxplot(df['Height'])
```

Out[11]:



In [12]:
```
upper_limit = df['Height'].quantile(0.99)
upper_limit
```

Out[12]: 74.7857900583366

In [13]:
```
lower_limit = df['Height'].quantile(0.01)
lower_limit
```

Out[13]: 58.134411586716546

In [17]:
```
new_df = df[(df['Height'] <= 74.78) & (df['Height'] >= 58.13)]
```

In [18]:
```
new_df['Height'].describe()
```

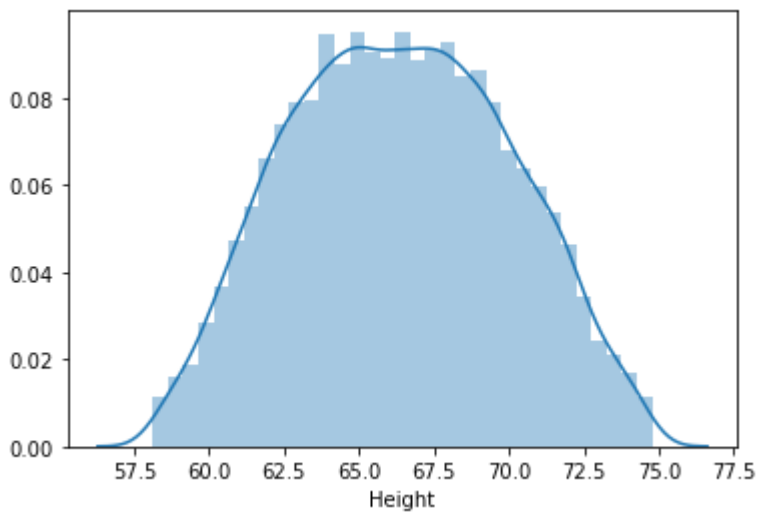
Out[18]:
```
count    9799.000000
mean       66.363507
std         3.644267
min        58.134496
25%        63.577147
50%        66.317899
75%        69.119859
max        74.767447
Name: Height, dtype: float64
```

In [19]:
```
df['Height'].describe()
```

Out[19]:
```
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
```

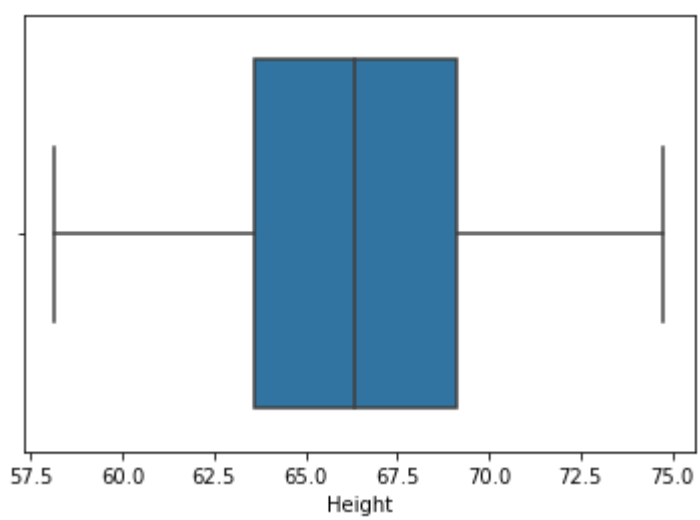
In [20]:
```
sns.distplot(new_df['Height'])
```

Out[20]:



In [21]:
```
sns.boxplot(new_df['Height'])
```

Out[21]:



In [24]:
```
# Capping --> Winsorization
df['Height'] = np.where(df['Height'] >= upper_limit,
        upper_limit,
        np.where(df['Height'] <= lower_limit,
        lower_limit,
```

```
        df['Height']))
```

In [26]:
```
df.shape
```

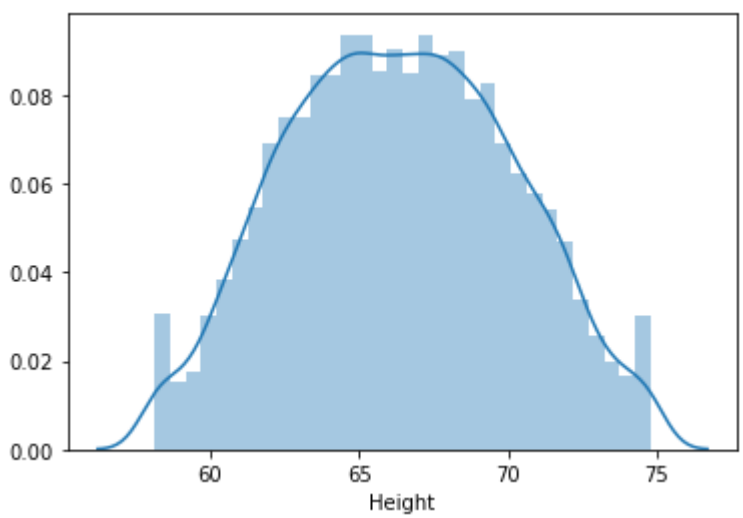
Out[26]:
```
(10000, 3)
```

In [27]:
```
df['Height'].describe()
```

Out[27]:
```
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
```

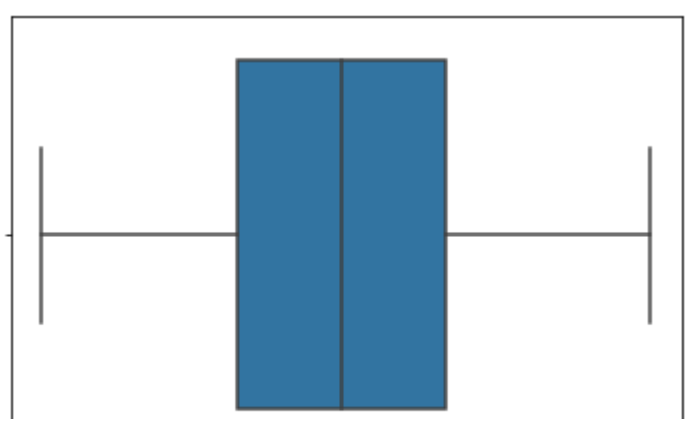
In [28]:
```
sns.distplot(df['Height'])
```

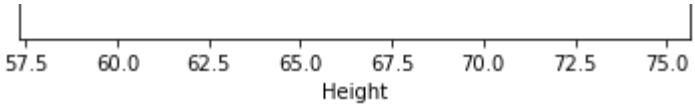
Out[28]:



In [29]:
```
sns.boxplot(df['Height'])
```

Out[29]:





In [ ]: