


```
import pandas as pd
```

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

```
import pandas as pd
da = pd.read_csv('/content/nhanes_2015_2016.csv')
```


da



	SEQN	ALQ101	ALQ110	ALQ130	SMQ020	RIAGENDR	RIDAGEYR	RIDRETH1	DMDCITZN	DMD
0	83732	1.0	NaN	1.0	1	1	62	3	1.0	
1	83733	1.0	NaN	6.0	1	1	53	3	2.0	
2	83734	1.0	NaN	NaN	1	1	78	3	1.0	
3	83735	2.0	1.0	1.0	2	2	56	3	1.0	
4	83736	2.0	1.0	1.0	2	2	42	4	1.0	
...
5730	93695	2.0	2.0	NaN	1	2	76	3	1.0	
5731	93696	2.0	2.0	NaN	2	1	26	3	1.0	
5732	93697	1.0	NaN	1.0	1	2	80	3	1.0	
5733	93700	NaN	NaN	NaN	1	1	35	3	2.0	
5734	93702	1.0	NaN	2.0	2	2	24	3	1.0	

5735 rows x 28 columns

```
da.head()
```




	SEQN	ALQ101	ALQ110	ALQ130	SMQ020	RIAGENDR	RIDAGEYR	RIDRETH1	DMDCITZN	DMDEDU
0	83732	1.0	NaN	1.0	1	1	62	3	1.0	5
1	83733	1.0	NaN	6.0	1	1	53	3	2.0	3
2	83734	1.0	NaN	NaN	1	1	78	3	1.0	3
3	83735	2.0	1.0	1.0	2	2	56	3	1.0	5
4	83736	2.0	1.0	1.0	2	2	42	4	1.0	4

5 rows x 28 columns

```
da.shape
```

(5735, 28)

```
da['BMXHT']
```



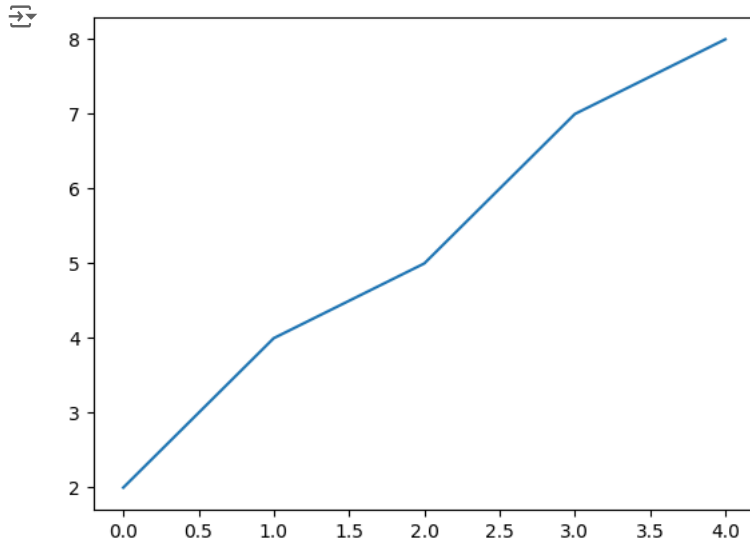
0	184.5
1	171.4
2	170.1
3	160.9
4	164.9
...	
5730	165.8
5731	182.2
5732	152.2
5733	173.3
5734	165.0

Name: BMXHT, Length: 5735, dtype: float64

```
y = [2 , 4 , 5 , 7, 8]
```

```
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
plt.plot(y)
plt.show()
```

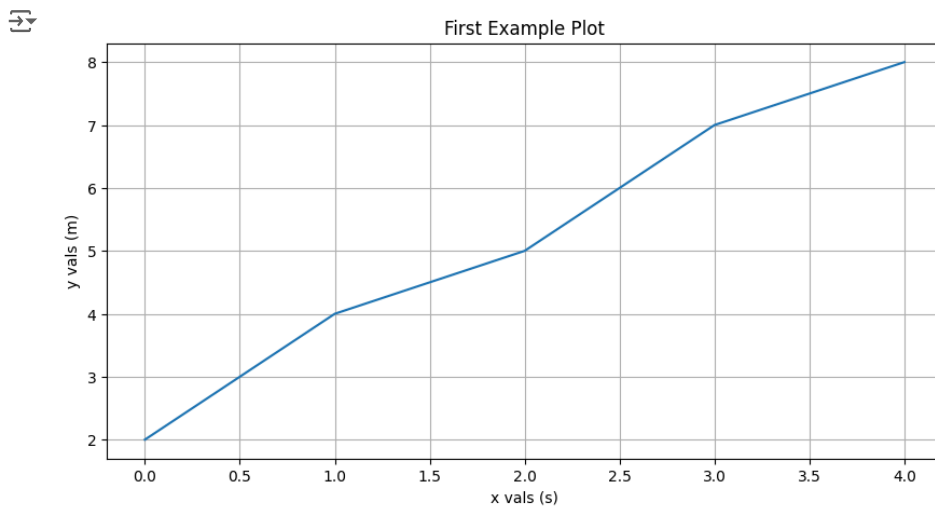


```
# create a figure
plt.figure(figsize=(10, 5))

# add a line renderer with legend and line thickness
plt.plot(y)

# also need to make the figure readable -- very important from stats perspective!
plt.title("First Example Plot")
plt.xlabel("x vals (s)")
plt.ylabel("y vals (m)")
plt.grid()

plt.show()
```



```

## Plotting Weights
# create a figure
plt.figure(figsize=(20, 5))

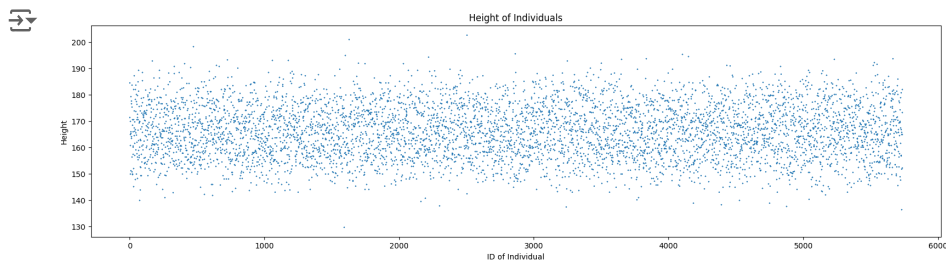
# add a line renderer with legend and line thickness
x = range(da['BMXHT'].size)
y = da['BMXHT']

plt.scatter(x, y, marker='x', s=0.5)

# also need to make the figure readable -- very important from stats perspective!
plt.title("Height of Individuals")
plt.xlabel("ID of Individual")
plt.ylabel("Height")

plt.show()

```



```

plt.figure(figsize=(20, 5))

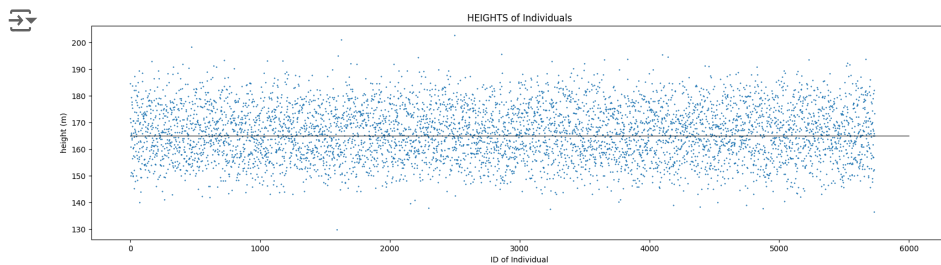
x = range(da['BMXHT'].size)
y = da['BMXHT']

plt.scatter(x, y, marker='x', s=0.5)

plt.title("HEIGHTS of Individuals")
plt.xlabel("ID of Individual")
plt.ylabel("height (m)")

# plot the estimate line
p1 = (0, 165)
p2 = (6000, 165)
x_c, y_c = zip(p1, p2)
plt.plot(x_c, y_c, color='black', linewidth=0.75)
plt.show()

```



```
#Calculate the mean of the 'BMXHT' column
mean_height = da['BMXHT'].mean()

print("Mean height:", mean_height)
```

```
Mean height: 166.1428344791116
```

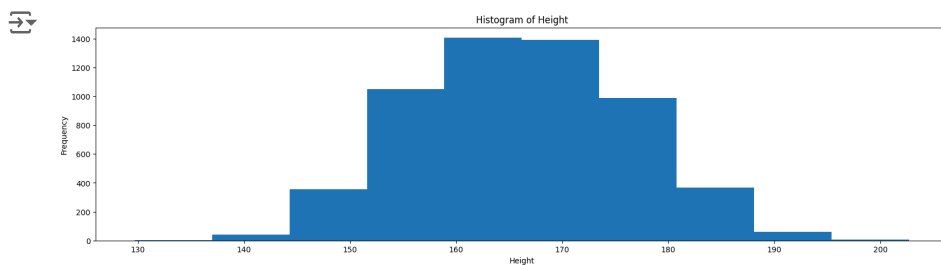
```
# create a histogram
plt.figure(figsize=(20, 5))

# add a line renderer with legend and line thickness
x = da['BMXHT']

plt.hist(x)

# also need to make the figure readable -- very important from stats perspective!
plt.title("Histogram of Height")
plt.xlabel("Height")
plt.ylabel("Frequency")

plt.show()
```



```
import numpy as np
```

```
nums = np.array([12,15,22,27,34,39,23,67,9,46,95,74])
```

```
np.median(nums)
```

```
↩ 30.5
```

```
import statistics
```

```
nums = np.array([12,15,22,27,34,39,23,67,9,46,95,74])
```

```
statistics.quantiles(nums, n=4)
```

```
↩ [16.75, 30.5, 61.75]
```

```
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import numpy as np
```

```
import seaborn as sns
sns.set(color_codes=True)
sns.set_style("black")
```

```
np.random.uniform(low=0.2, high=2.0)
```

```
↩ 1.4887396949573257
```

```
np.random.uniform(low=0.2, high=2.0, size = 12)
```

```
# All values equally likely. Sort of like [ 0.0 0.2 0.4 0.6 0.8 1.0 ]
```

```
↩ array([1.11958697, 1.89661044, 1.451976 , 1.74783604, 0.21379935,
        1.63694187, 1.48580176, 1.01710703, 1.72649207, 1.0995039 ,
        1.22961246, 0.42403102])
```

```
num_samples = 10000 # get rid of 'magic numbers'
```

```
import seaborn as sns
```

```
uniform_vals = np.random.uniform(low=0.8, high=5.0, size = num_samples)
```

```
sns.distplot(uniform_vals, bins=20, kde=False)
plt.ylabel('Frequency')
plt.xlabel('Value');
plt.title("Unform Values")
sns.despine(offset=10, trim=True); # move axes away
plt.show()
```



<ipython-input-34-9792b910a761>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(uniform_vals, bins=20, kde=False)
```

Unform Values