Tutorial 2

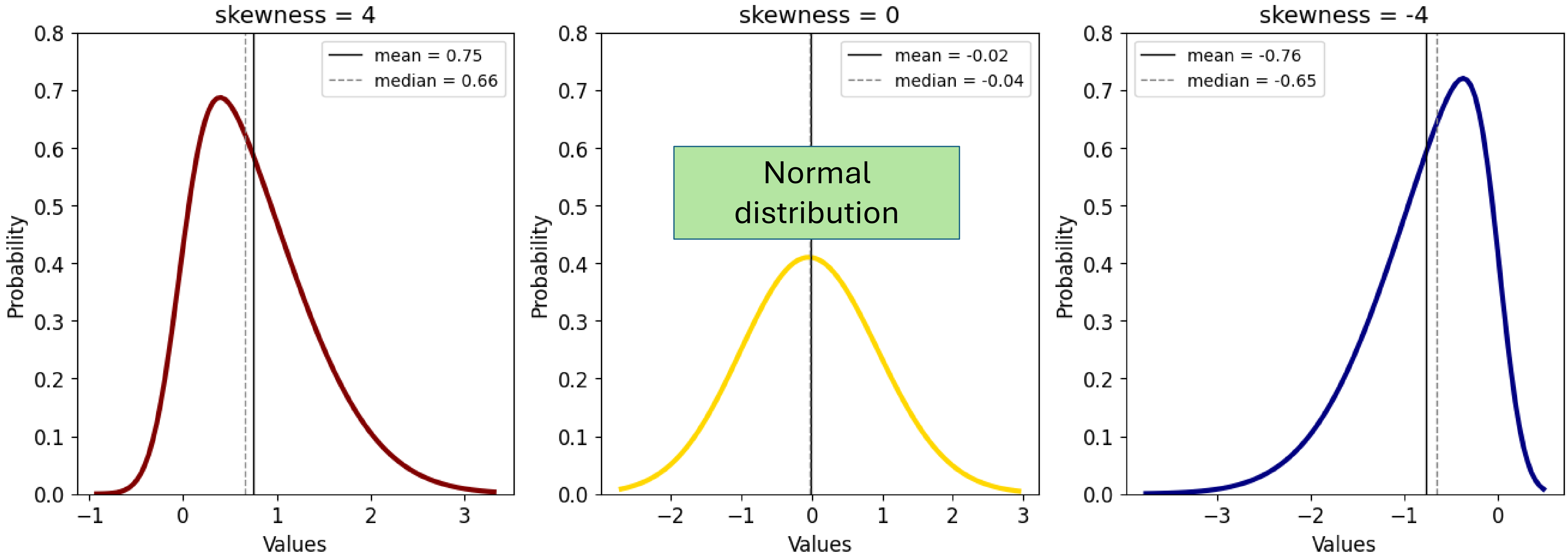
Statistical Computation and Analysis

Spring 2024

Skewness

- How symmetric is the data?

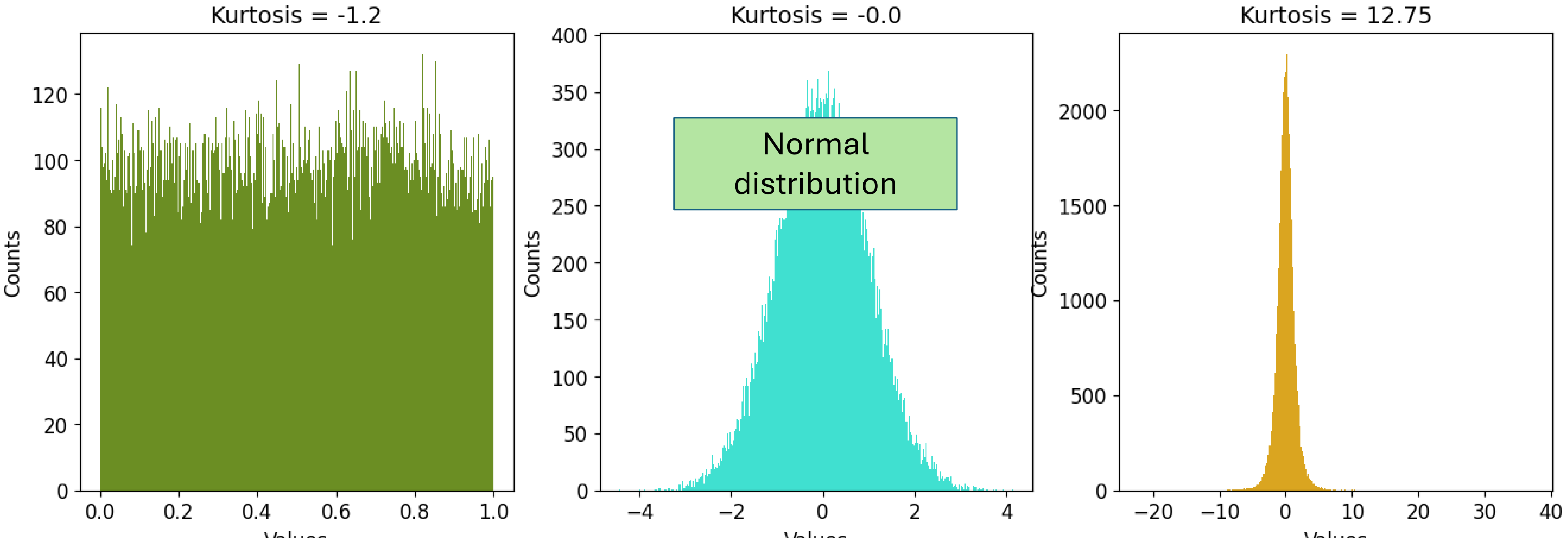
$$\text{Skewness} = \frac{1}{\sigma^3} E\left(\left(x_i - \mu\right)^3\right)$$



Kurtosis

- Size of tails
- Pointiness of peak

$$\text{Kurtosis} = \frac{1}{\sigma^4} E\left(\left(x - \mu\right)^4\right) - 3$$



Tutorial Outline

- Models
- Examining where data comes from
- Validity and reliability
- Simulation

Models

- A model is a specific idea about the shape of the population
  - A particular distribution
    - With specific parameters
- We can use models to:
  - Express ideas about the population.
  - Check ideas about the population.
  - Provide alternative ideas about the population.
- A good model:
  - Fits the data
  - Simple
  - Expressive

Models

heights = np.array([165, 161, 157, 165, 162, 164, 161, 169, 160, 174, 165, 166, 163, 160, 154, 150, 162, 171,

166, 172, 157, 175, 158, 168, 163, 164, 164, 167, 161, 169, 170, 168, 166, 163, 163, 163,

166, 159, 166, 164, 174, 173, 171, 158, 165, 172, 164, 155, 163, 169, 164, 161, 153, 167,

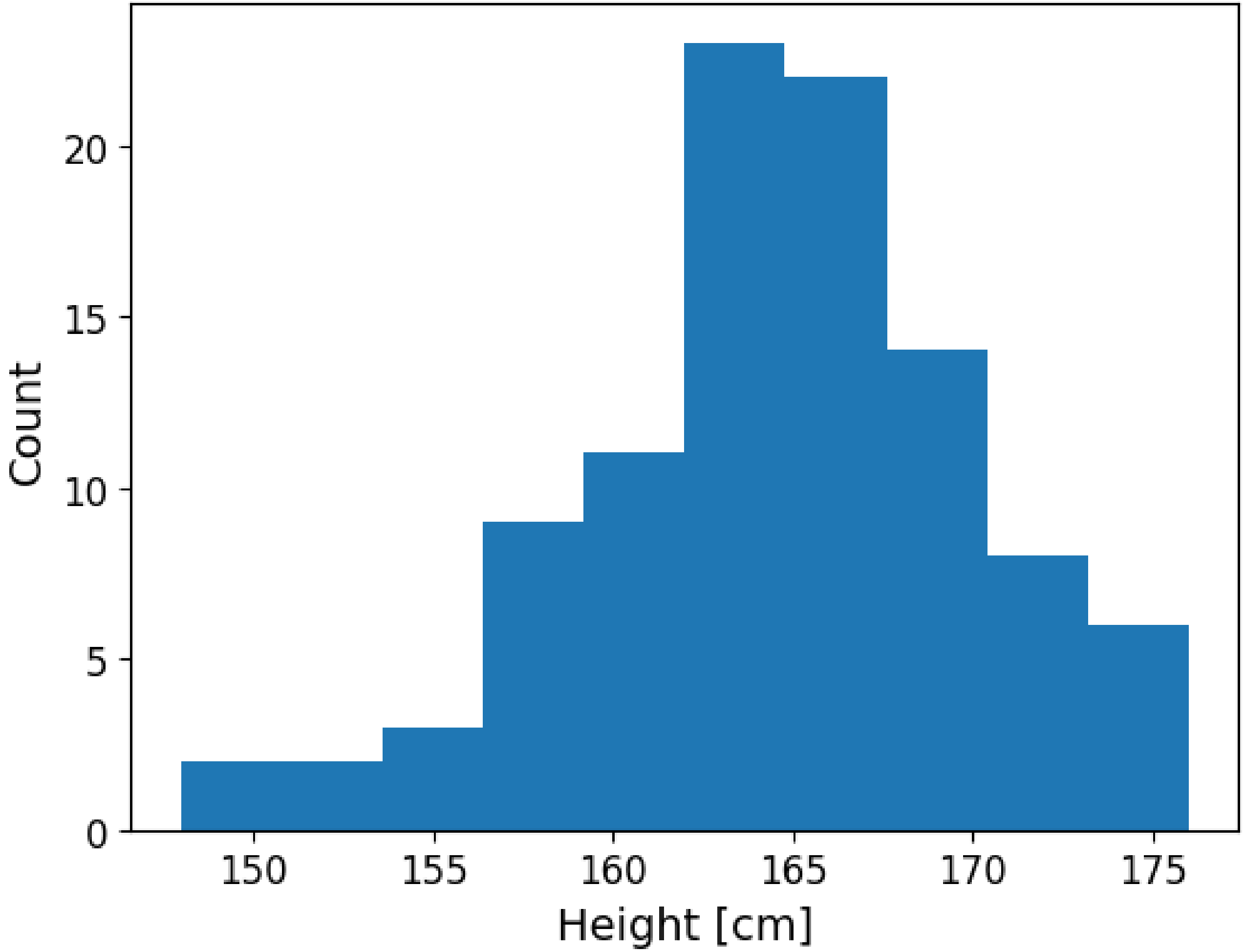
160, 162, 159, 163, 175, 161, 167, 172, 160, 173, 163, 176, 148, 162, 165, 160, 170, 166,

167, 166, 165, 157, 171, 170, 168, 162, 165, 169, 165, 166, 162, 165, 157, 175, 169, 156,

157, 164, 169, 169, 162, 161, 152, 168, 167, 163])

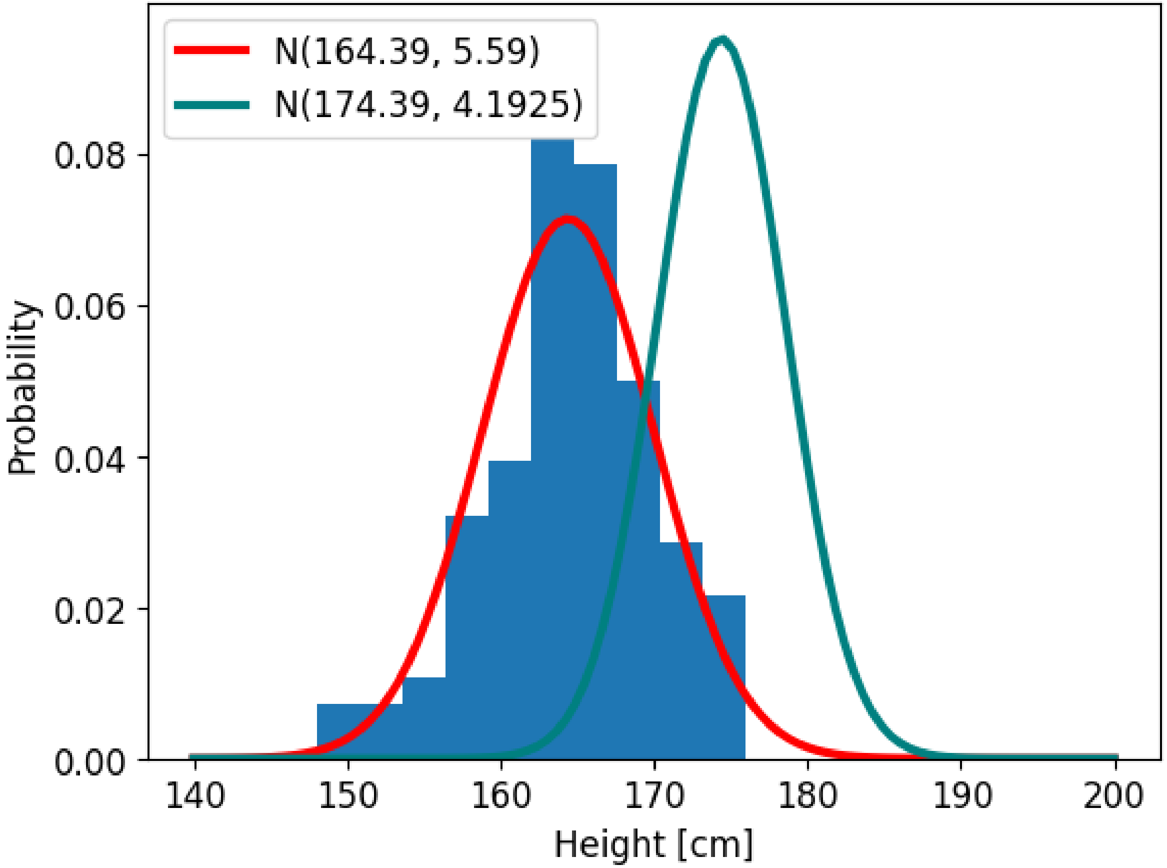
Models

- Examine the data



Models

- Does it make sense that this data came from each model?



Models

```
plt.hist(heights, density = True, bins = 10)


#models
pdf_f = stats.norm.pdf(np.linspace(140, 200, 100), np.mean(heights), np.std(heights))
plt.plot(np.linspace(140, 200, 100), pdf_f, color = 'red', lw = 3, label =
f'N({round(np.mean(heights), 2)}, {round(np.std(heights), 2)})')

pdf_f2 = stats.norm.pdf(np.linspace(140, 200, 100), np.mean(heights) + 10,
np.std(heights)*0.75)
plt.plot(np.linspace(140, 200, 100), pdf_f2, color = 'teal', lw = 3, label =
f'N({round(np.mean(heights), 2) + 10}, {round(np.std(heights), 2)*0.75})')

plt.ylabel('Probability', fontsize = 12)
plt.xlabel('Height [cm]', fontsize = 12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(fontsize = 12)
```

Example Dataset

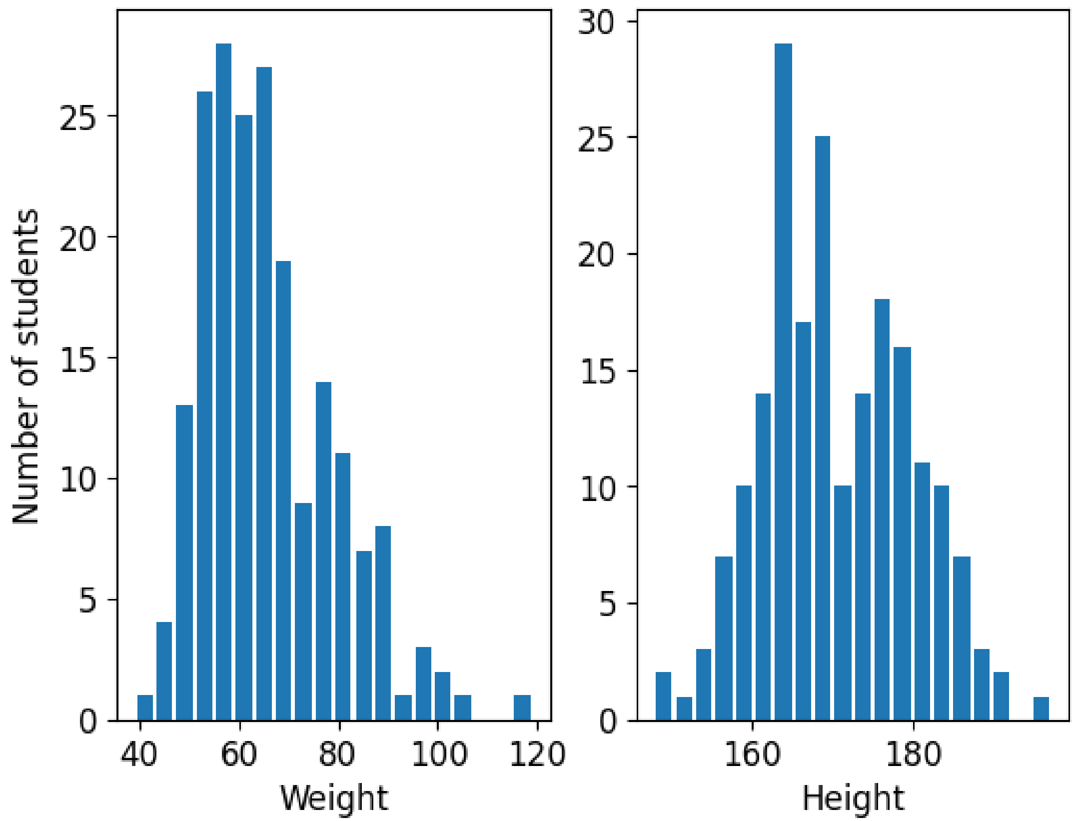
Self-reported and real weight and height of 200 students – 88 males and 112 females.

	subject	sex	weight	height	repwt	repht
0	1	M	77	182	77.0	180.0
1	2	F	58	161	51.0	159.0
2	3	F	53	161	54.0	158.0
3	4	M	68	177	70.0	175.0
4	5	F	59	157	59.0	155.0
..	...	..	...	...	...	...
195	196	M	74	175	71.0	175.0
196	197	M	83	180	80.0	180.0
197	198	M	81	175	NaN	NaN
198	199	M	90	181	91.0	178.0
199	200	M	79	177	81.0	178.0

[200 rows x 6 columns]

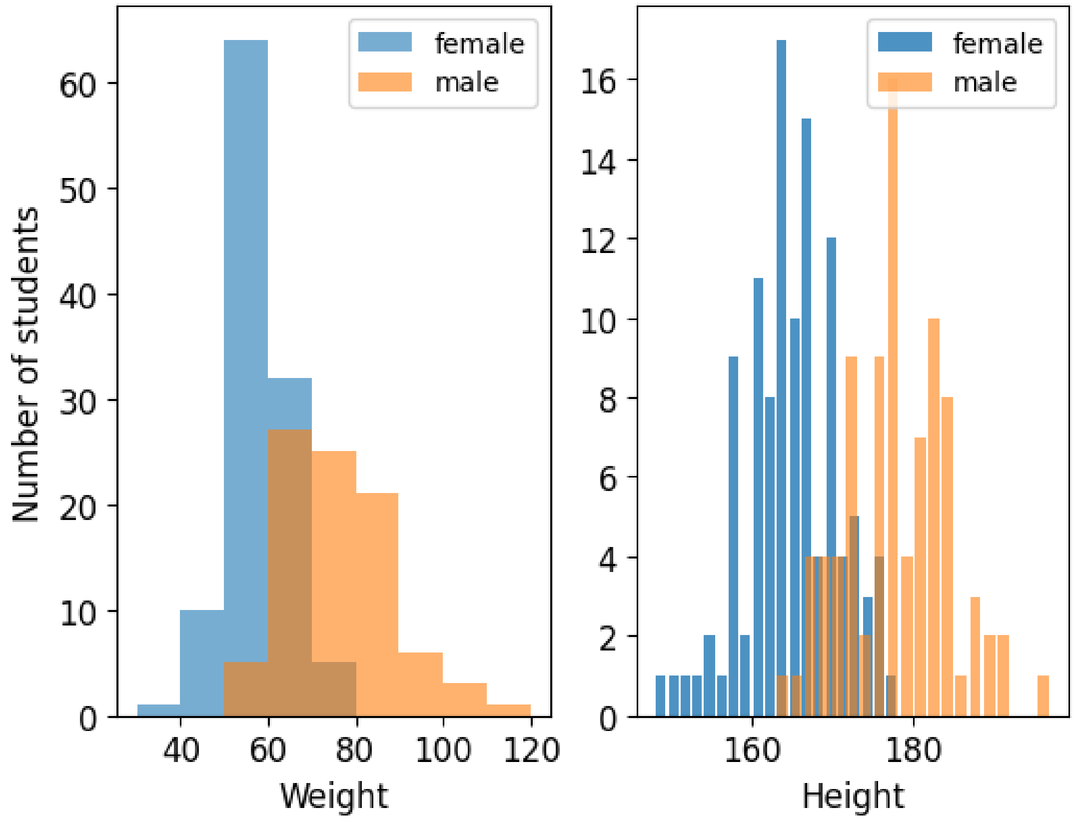
Example Dataset

Examine data



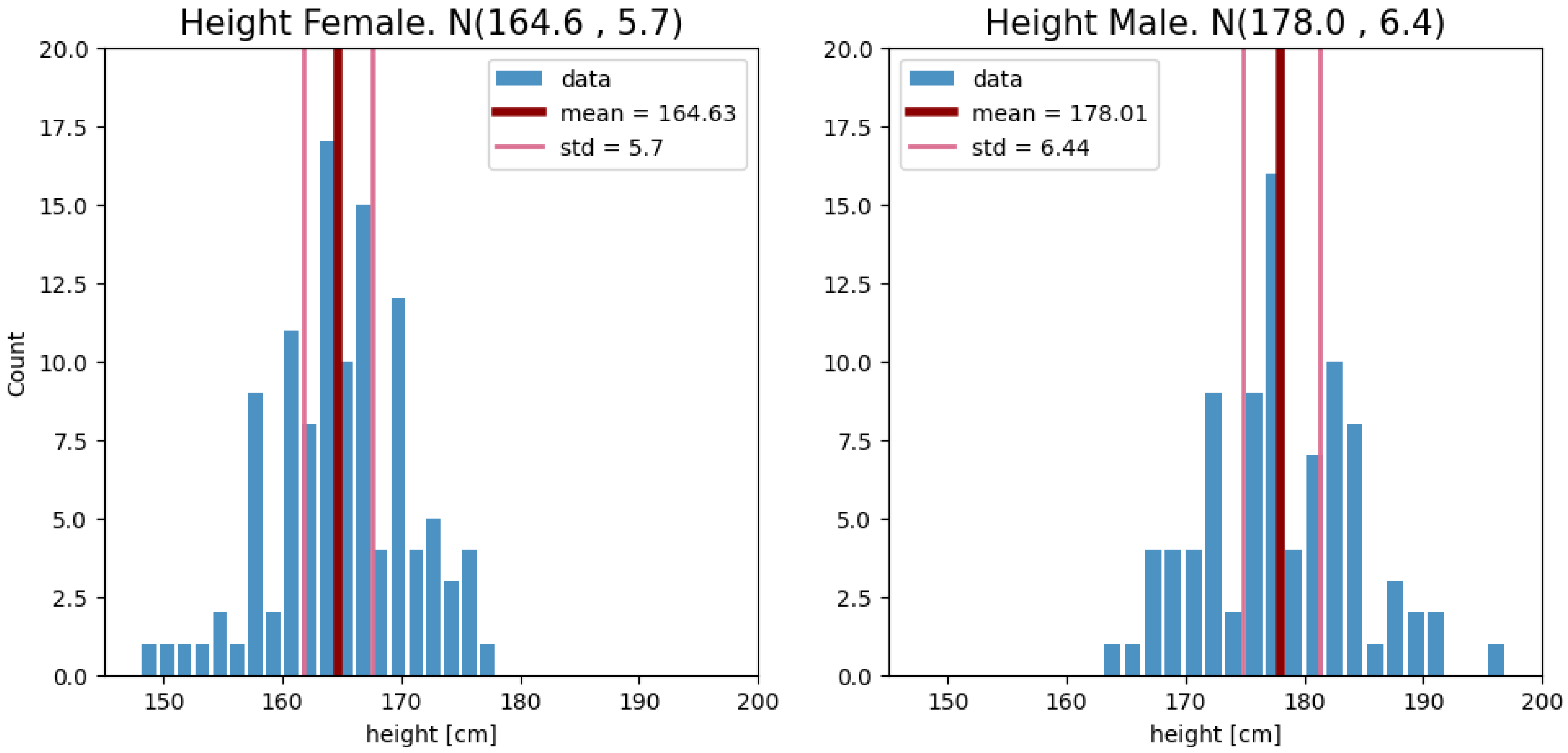
Example Dataset

Examine data



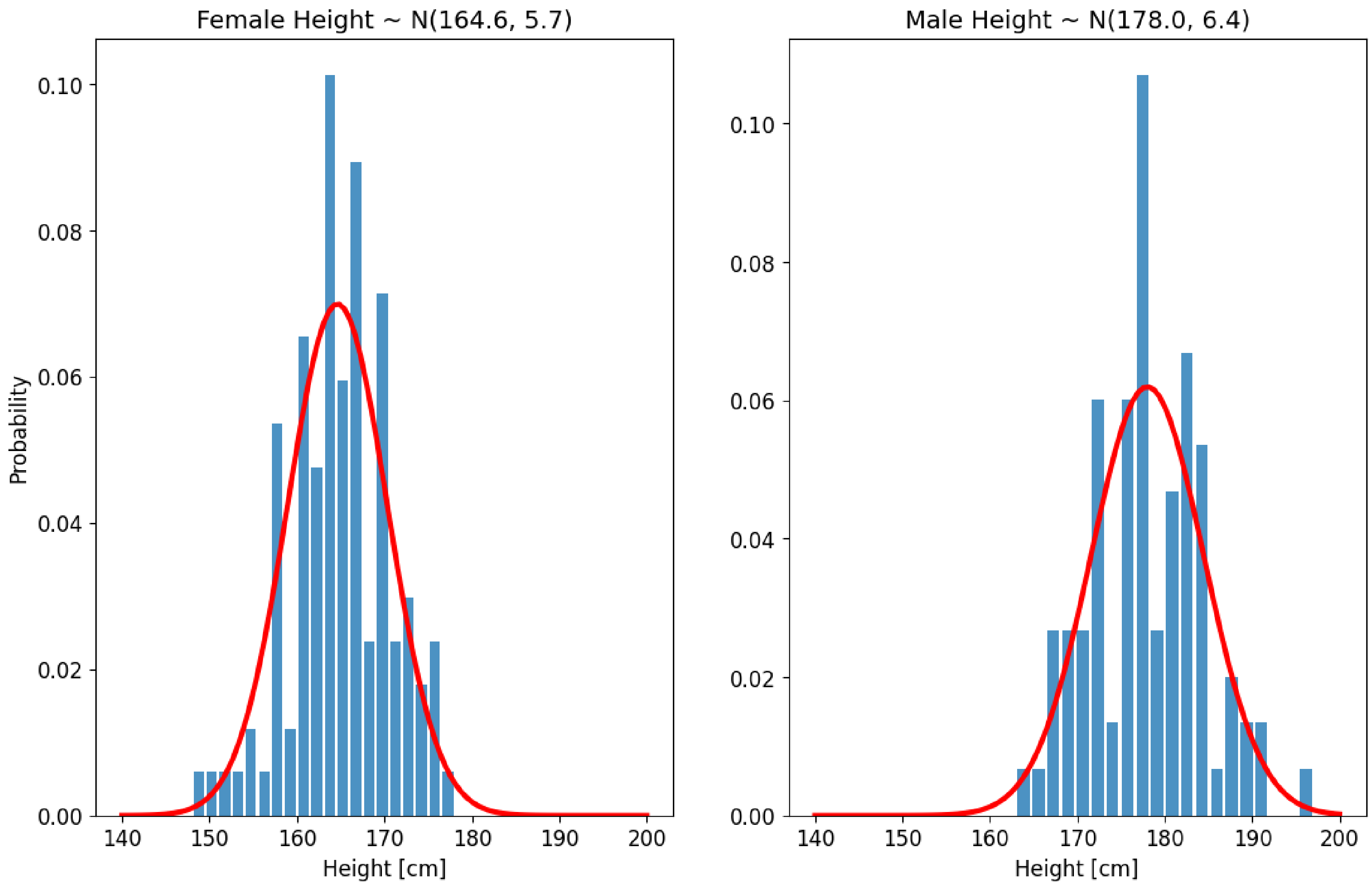
Example Dataset

Compute statistics



Example Dataset

Model data



Example Dataset

Model data:

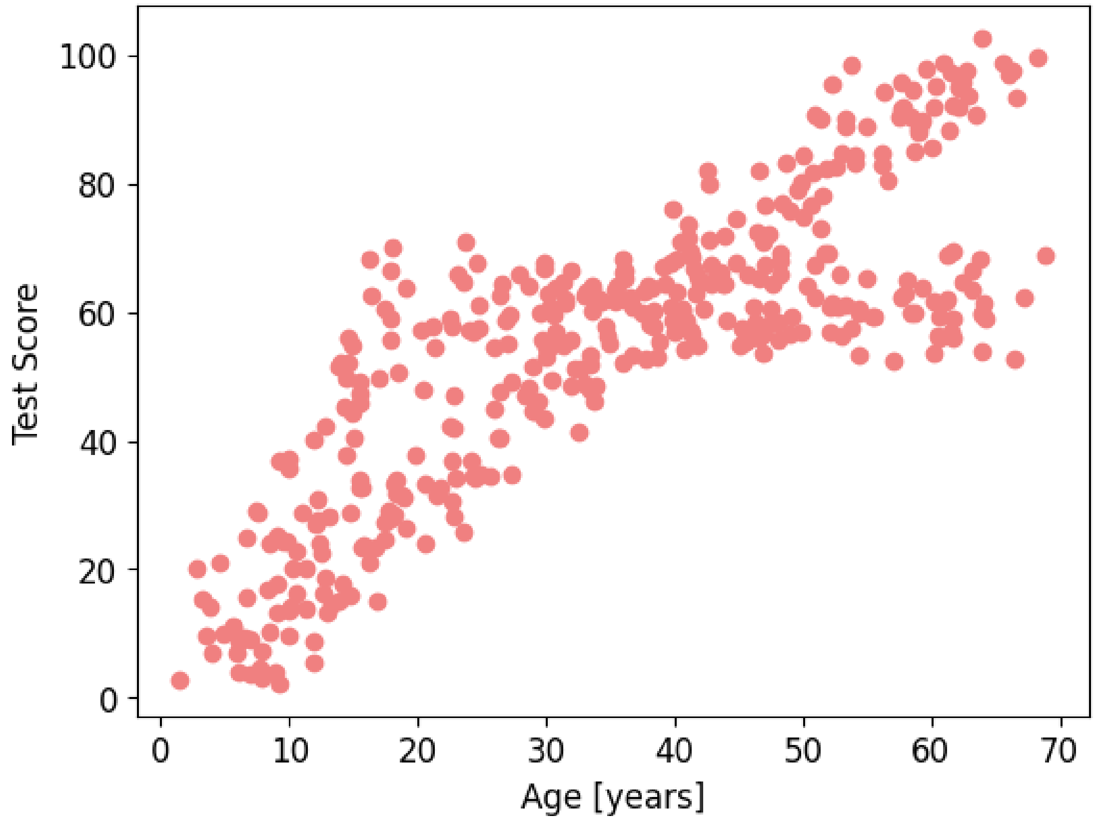
- Distribution that fits heights based on sample:
  - female ~ N(165 , 5.7)
  - male ~ N(178 , 6.4)

Examining Where Data Came From

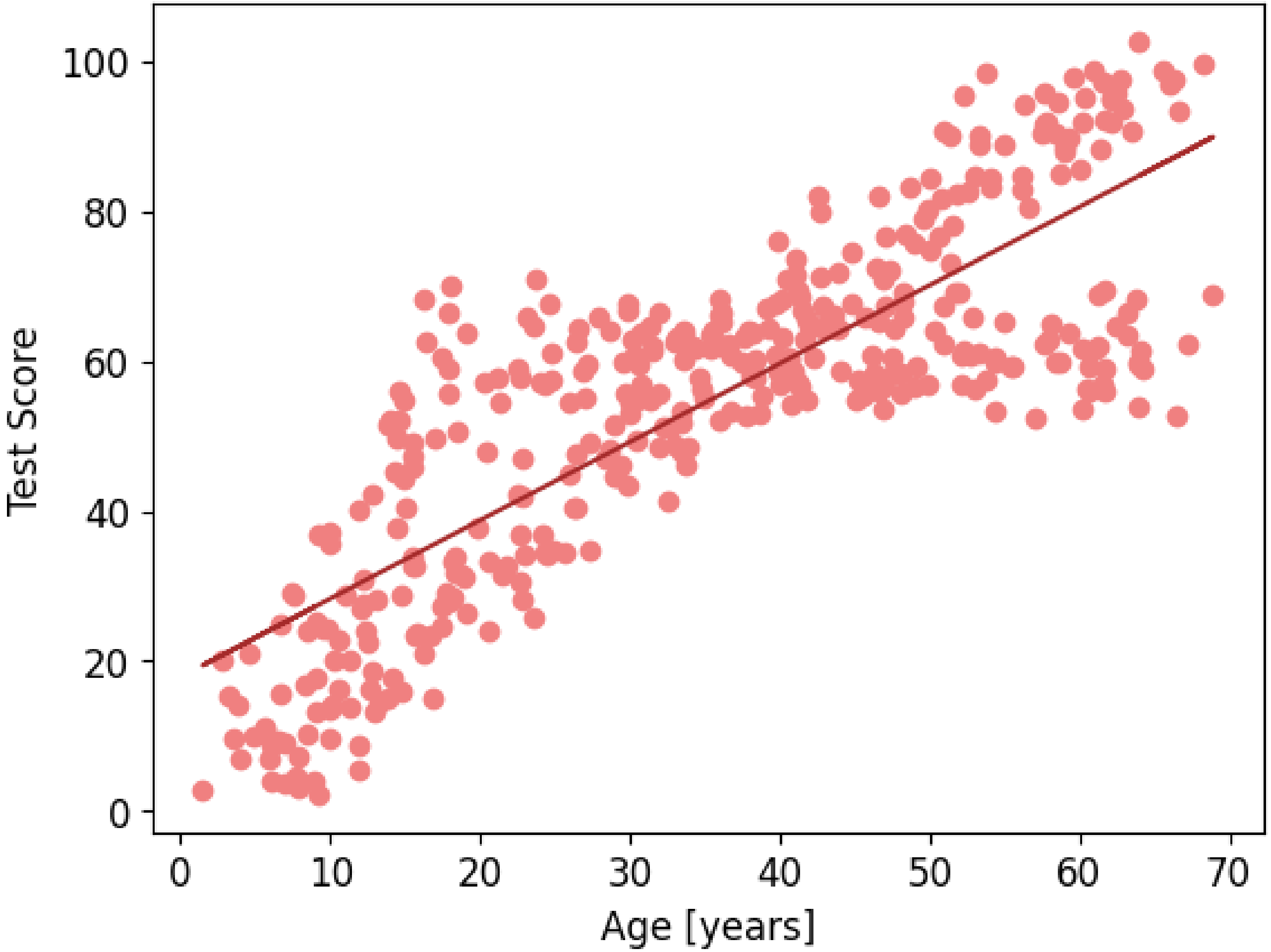
- Is our data measured in appropriate units (precision)?
  - Weight in kg is good for measuring human weight,
  - Weight in mg is needed for measuring medicine weight.
- What does our data mean?
  - Example: measure effect of medicine – what does the measurement mean? Does a high or low measurement value indicate higher success?
  - Examine the data.
- Does vocabulary improve with age?

Does vocabulary improve with age?

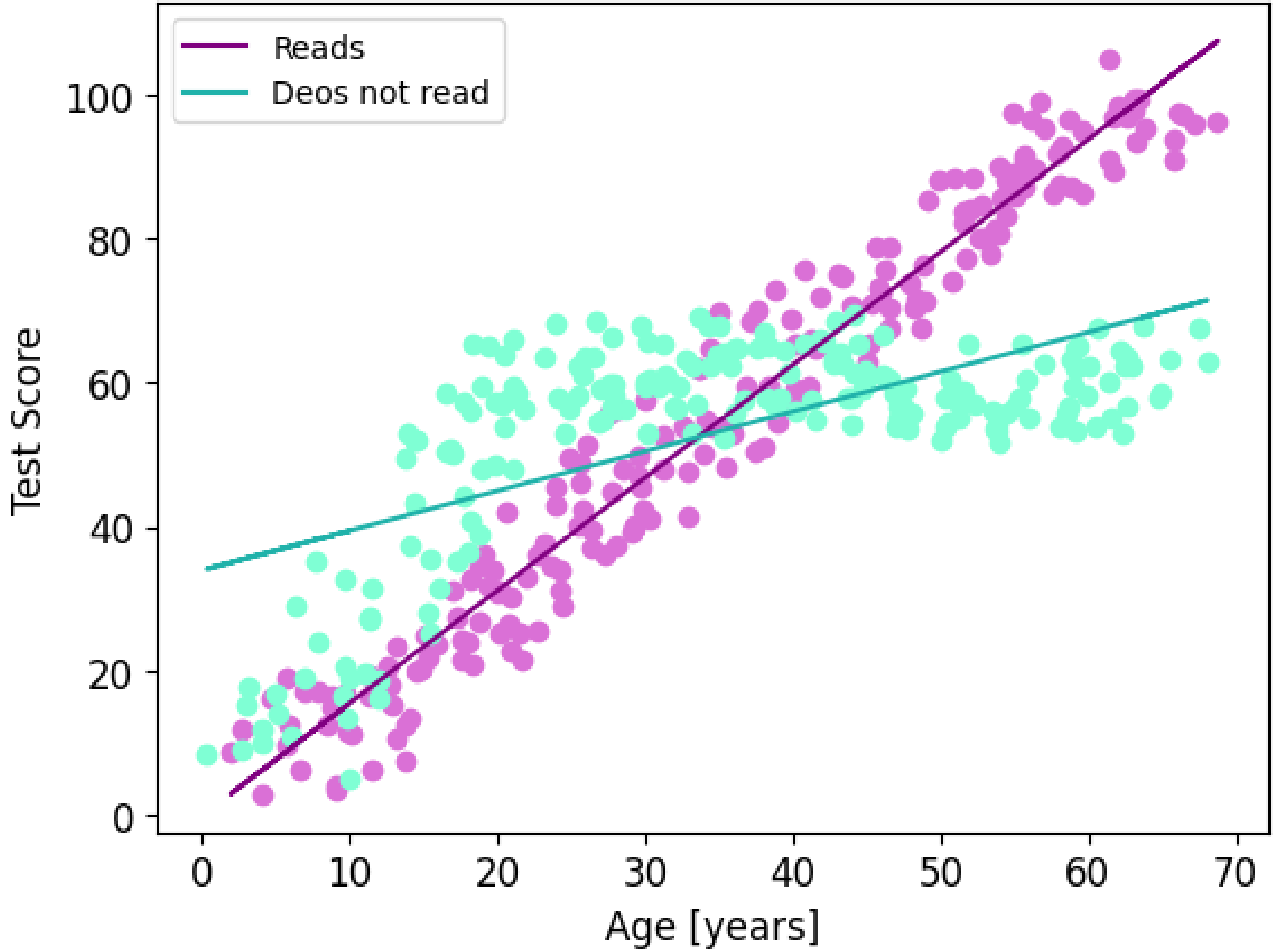
- Age (years)
- Vocabulary test score (0 – 100)



Does vocabulary improve with age?



Does vocabulary improve with age?



Data Creation

- Read:

  Ages = np.linspace(0, 60, 200)

  Scores = np.linspace(0, 100, 200)

  Add noise

- Does not read:

  Ages = np.linspace(0, 60, 200)

  Scores = np.concatenate((np.linspace(0, 50, 50), 50*np.ones(150))

  Add noise

Data Creation

```
#define function to add noise to simulated data
def noise(k, noise_factor):
   return k + rd.random()*noise_factor

#create simulated data to examine the relation between age and grade on vocubulary test
#reads
Ages1 = np.linspace(0, 60, 200)
Scores1 = np.linspace(0, 100, 200)


Ages11 = np.reshape(np.vectorize(noise)(Ages1, 10), (Ages1.shape[0], 1))
Scores11 = np.reshape(np.vectorize(noise)(Scores1, 12), (Scores1.shape[0], 1))
```

Data Creation

```
#correcting scores above 100
Scores11[np.flatnonzero(Scores11 > 100)] =
Scores11[np.flatnonzero(Scores11 > 100)] -
np.vectorize(noise)(5*np.ones_like(Scores11[np.flatnonzero
(Scores11 > 100)]), 10)

#regression model for this data
regr_model = LinearRegression().fit(Ages11, Scores11)
b0_1 = regr_model.intercept_
b1_1 = regr_model.coef_
```

Data Creation

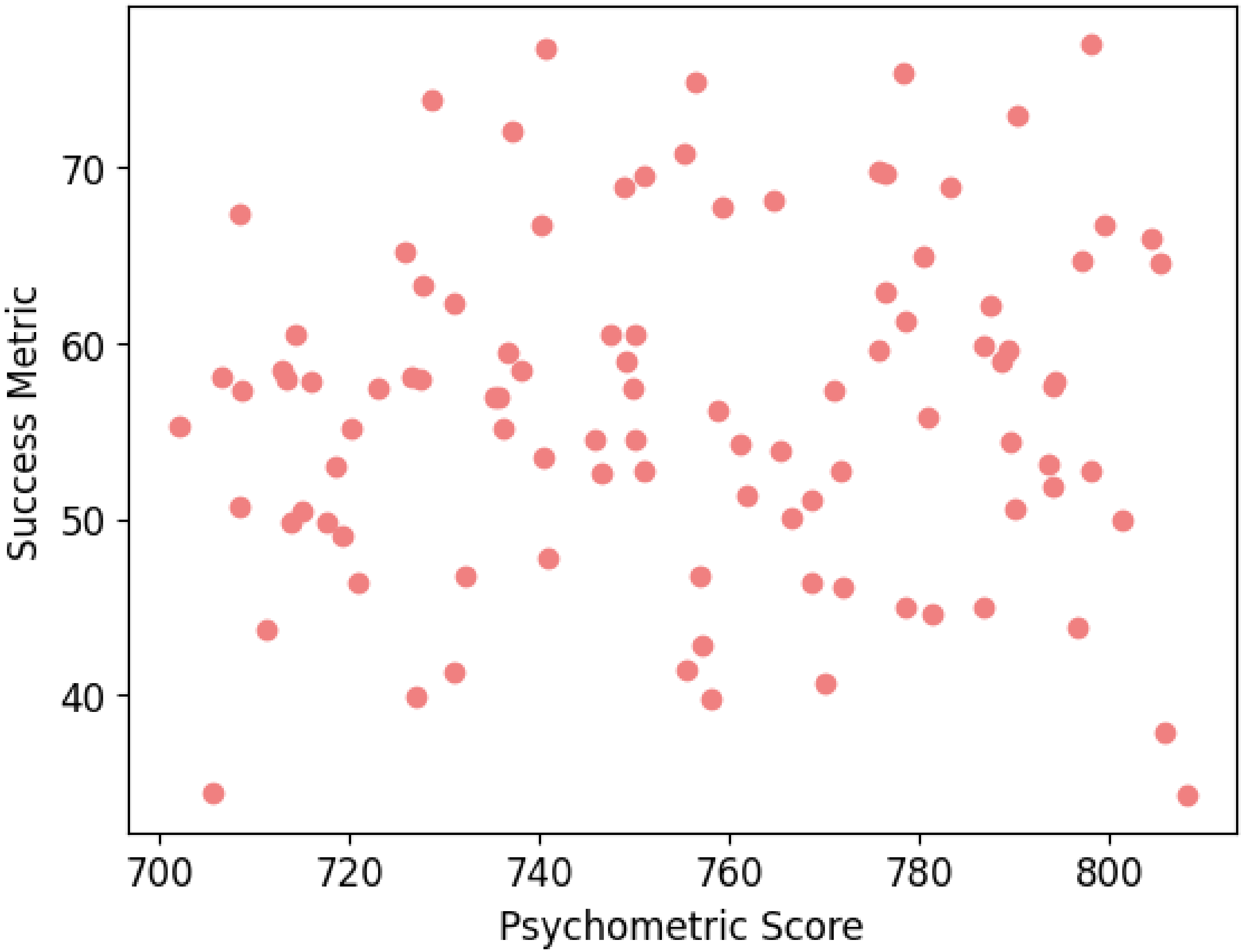
```
#doesn't read
Scores2 = np.concatenate((np.linspace(0, 50, 50),
np.vectorize(noise)(50*np.ones(150,), 6)))
Scores22 = np.reshape(np.vectorize(noise)(Scores2, 15),
(Scores2.shape[0], 1))
Ages22 = np.reshape(np.vectorize(noise)(Ages1, 10),
(Ages1.shape[0], 1))

#regression model for this data
regr_model = LinearRegression().fit(Ages22, Scores22)
b0_2 = regr_model.intercept_
b1_2 = regr_model.coef_
```

Validity

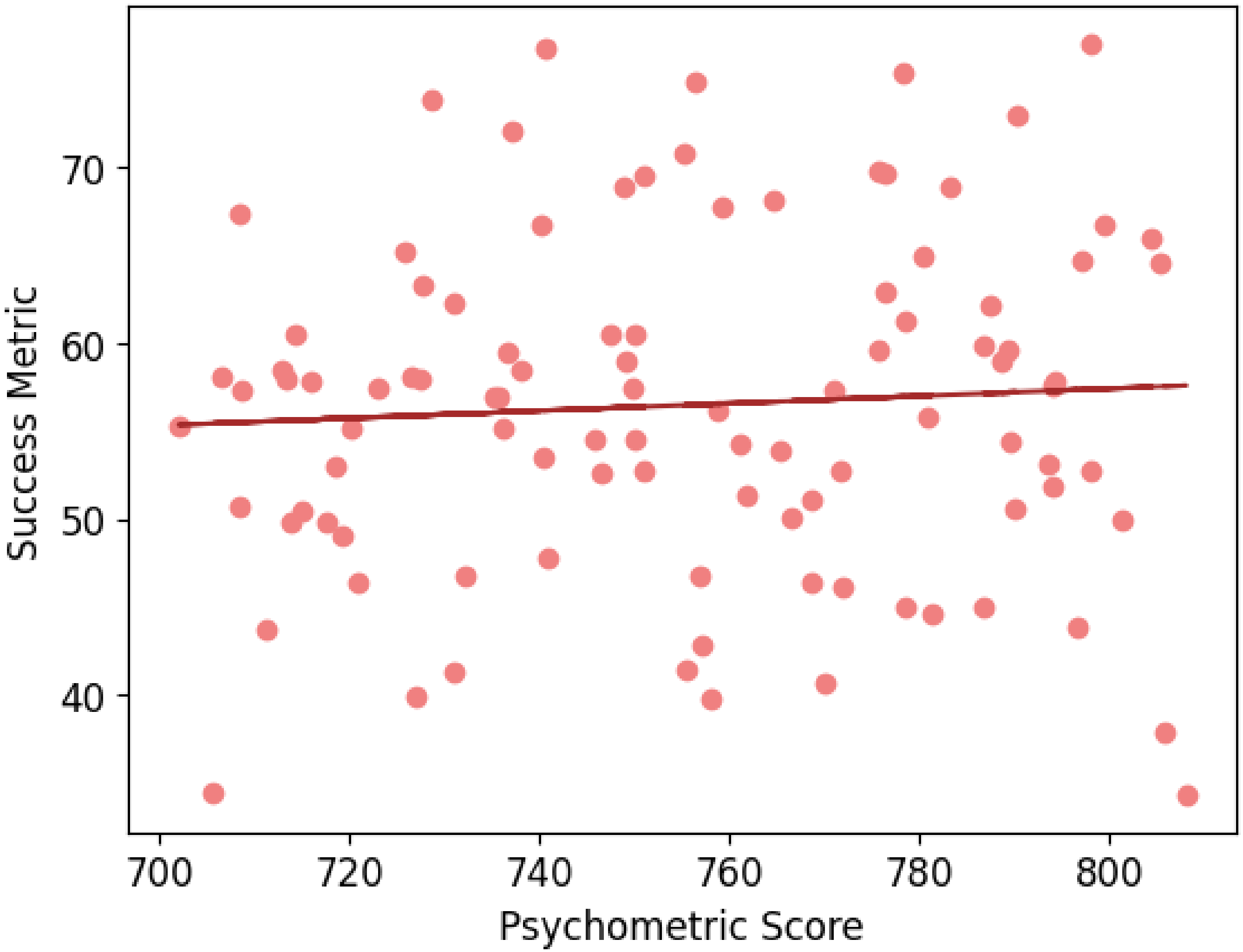
- Can the data answer our question?
  - Can grade on reading test be indicative of music ability?
    - Probably not.
  - Is psychometric grade indicative of who would be a good doctor?
    - Maybe.

- What study can test this?
  - Take a group of doctors:
    - Psychometric scores
    - Success metric (e.g., number of successful surgeries, cases solved….)

Validity


Success Metric
70
60
50
40
700
720
740
760
780
800
Psychometric Score

Validity



Success Metric
Psychometric Score
70
60
50
40
700
720
740
760
780
800

Data Creation

- Psychometric scores

  psychometric = np.linspace(700, 800, 100)

  Add noise

- Success metric:

  success = 50+np.random.normal(0, 10, 100)

  Add noise

Reliability

- If we take a measurement, and then we have occasion to do it again, how much would the value change?
- The variability in our sample is due to real differences among people or things, and not due to random error incurred during the measurement process.
  - Run experiment twice.
  - Give test twice.
  - For subjective ratings – have several judges and examine variability between their scores.

Simulation – Number of Girls Born

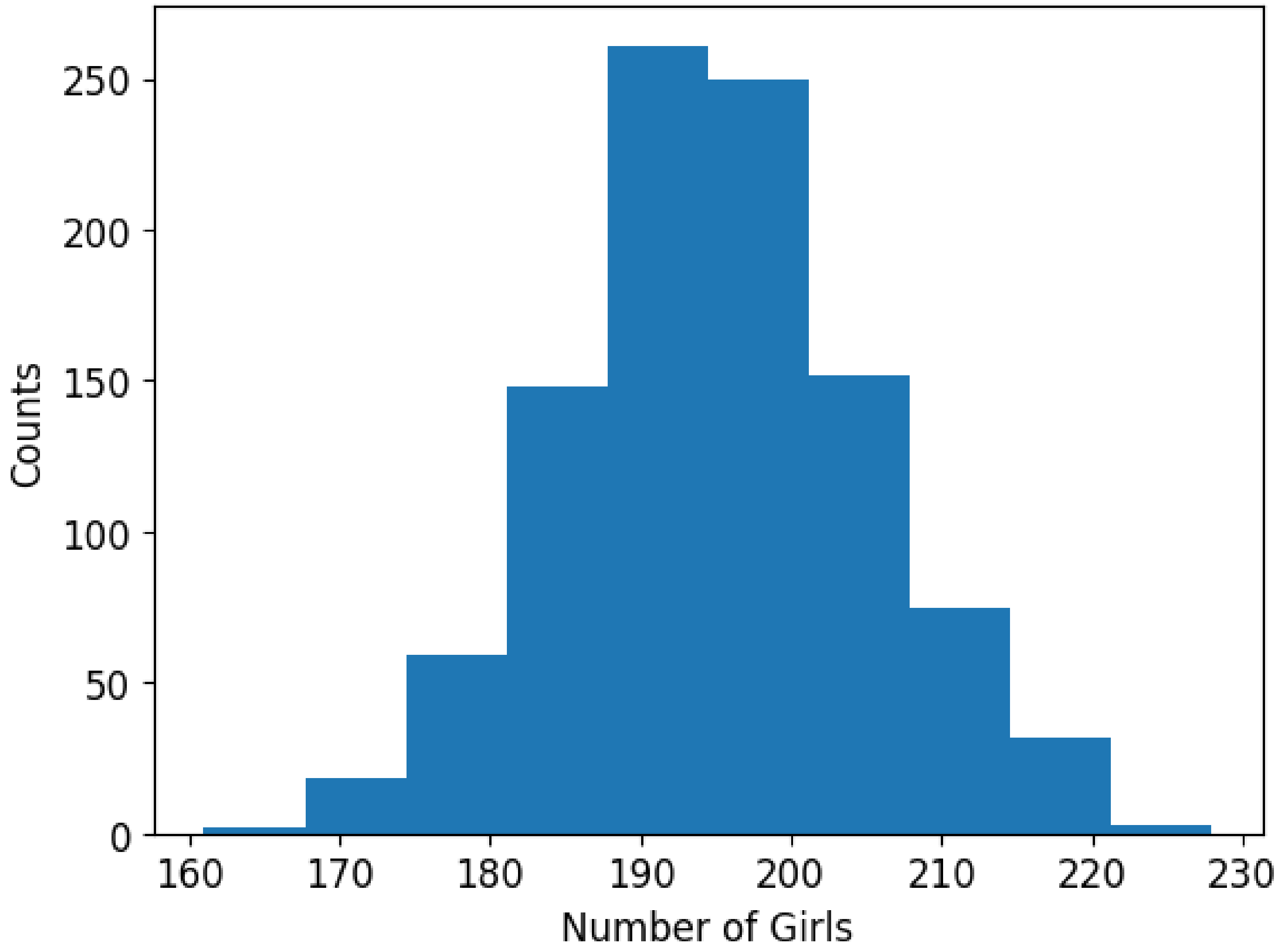
The probability that a baby is a girl or boy is approximately 48.8% or 51.2%, respectively, and these do not vary much across the world. Suppose that 400 babies are born in a hospital in a given year. How many will be girls?

- What distribution should we use?
  - Binomial

$$P(X = k) = \binom{n}{k} \mathrm{p}^{\mathrm{k}}(1 - \mathrm{p})^{\mathrm{n-k}}$$

```
girls = np.random.binomial(400, 0.488, 1000)
```

Simulation – Number of Girls Born



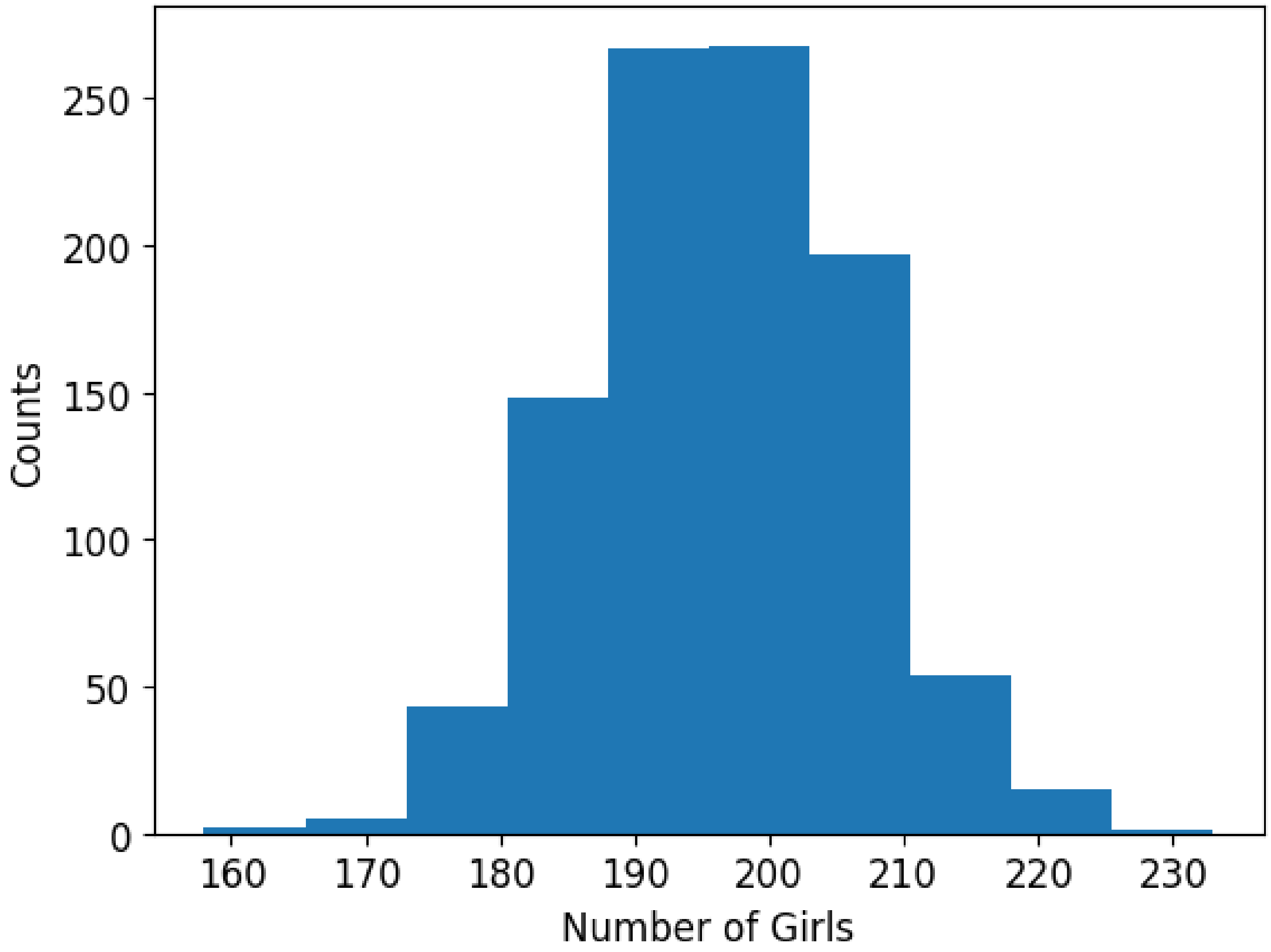
Simulation – Number of Girls Born

There is a 1/125 chance that a birth event results in fraternal twins, of which each has an approximate 49.5% chance of being a girl, and a 1/300 chance of identical twins, which have an approximate 49.5% chance of being a pair of girls. Simulate 400 birth events.

Simulation – Number of Girls Born

```
#brith types (twins and single births)
birth_types = np.array([0, 1, 2]) #fraternal twins, identical twins, single births
NumGirls = np.zeros((1000, 1))
for i in range(1000): #1000 repetitions
  #400 births in each simulation
  births = np.random.choice(birth_types, 400, replace = True, p = np.array([1/125,
1/300, 1 - 1/125 - 1/300]))
  girlboy = np.zeros((births.shape[0], 1)) #1 = girl, 0 = boy
  for j in range(births.shape[0]):
    if births[j] == 0: #fraternal twins
      girlboy[j] = np.random.binomial(2, 0.495, 1) #there are two born
    if births[j] == 1: #identical twins
      girlboy[j] = 2*np.random.binomial(1, 0.495, 1) #either both are twins or
neither are
    else:
      girlboy[j] = np.random.binomial(1, 0.488, 1)
  NumGirls[i] = np.sum(girlboy)
```

Simulation – Number of Girls Born



Homework submission guidelines

- Zip file named ex#_######### (exercise number + ID number) must be submitted to the course website
- Zip file should contain:
  1. word document with your solution
  2. pdf of this document
  3. code file – .py or .ipynb (with comments)
  4. data files
- Every claim or conclusion you make should have an explanation
- Every graph must have a title, axis labels, and legend if there is more than one plot on a graph
- When you display the graph explain its content below
- Code should be annotated
- The homework and the project must be written in English.