Topics:

Binomial & Bernoulli

- Gaussian / Normal distribution (more time) ✓
- Poisson disb
- Exponential disb
- Geometric disb
- Log-normal disb

Pace

various scenarios

100's of disb

Class-speed:

last-class

- slow vs Fast
- feedback @ end of the class also helps

pacing

prob

9% slow ← 1.5X

86% Good ← same

6% fast ← revise 2X

revise

4hr

Problem: DS @ Google HR

T-shirts

100K employees

[XXS, XS, S, M, L, XL, XXL

XXL: >200cm

XL: 180-200cm

⋮

Survey: $\boxed{1000}$ random employees → height

```
[5] employees = pd.read_csv("employees.csv")
```

```
employees.info()
```

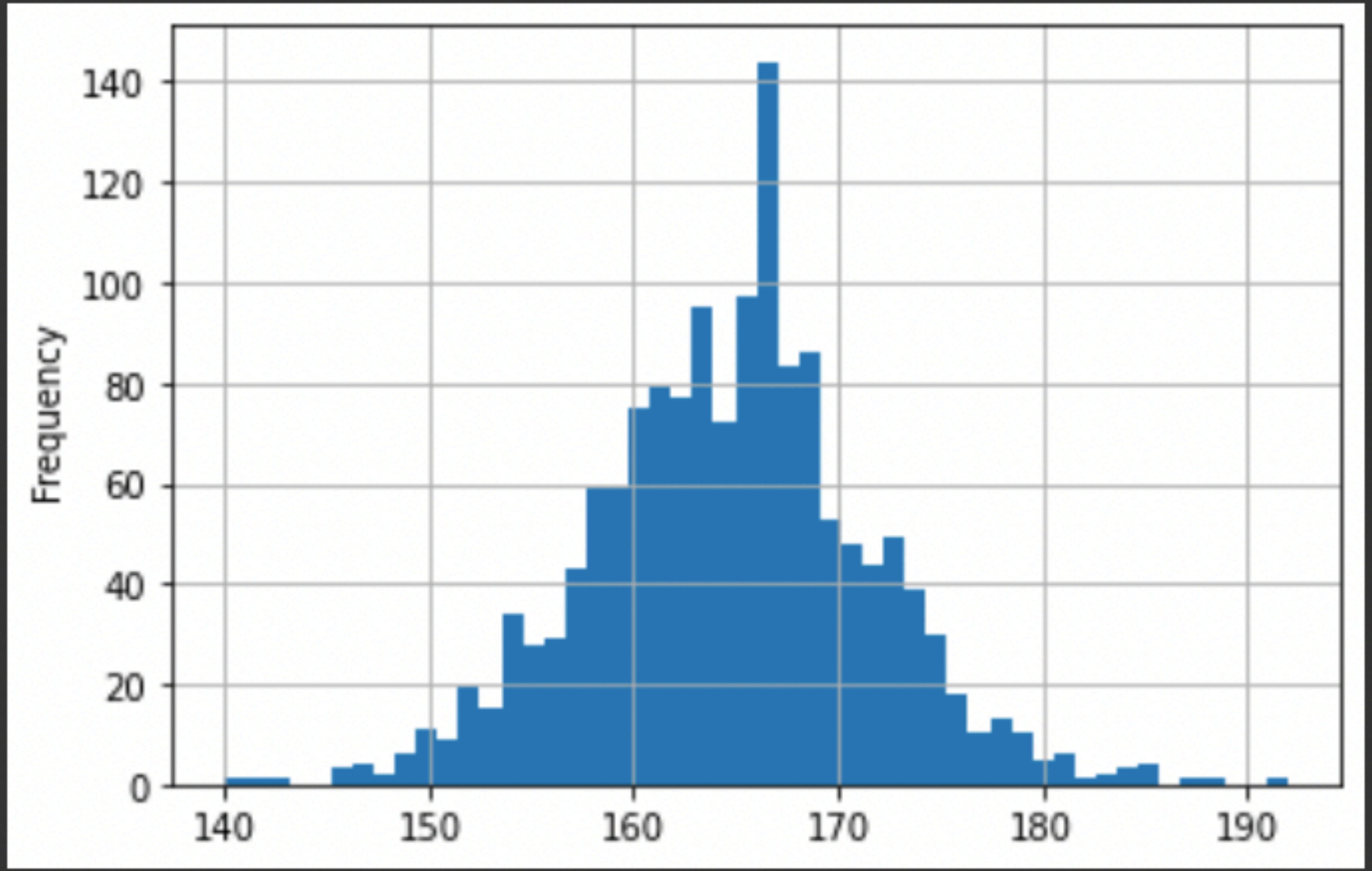
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Age               1470 non-null   int64
 1   Department        1470 non-null   object
 2   DistanceFromHome  1470 non-null   int64
 3   Education         1470 non-null   int64
 4   EmployeeNumber    1470 non-null   int64
 5   Gender            1470 non-null   object
 6   Height            1470 non-null   int64
 7   MaritalStatus     1470 non-null   object
 8   MonthlyIncome     1470 non-null   int64
dtypes: int64(6), object(3)
memory usage: 103.5+ KB
```

```
employees.head()
```

Age Department ... Gender Height MaritalStatus MonthlyIncome

Development

```
employees['Height'].plot(kind='hist', bins=50) # change with the bins value to show
plt.grid()
plt.show()
```

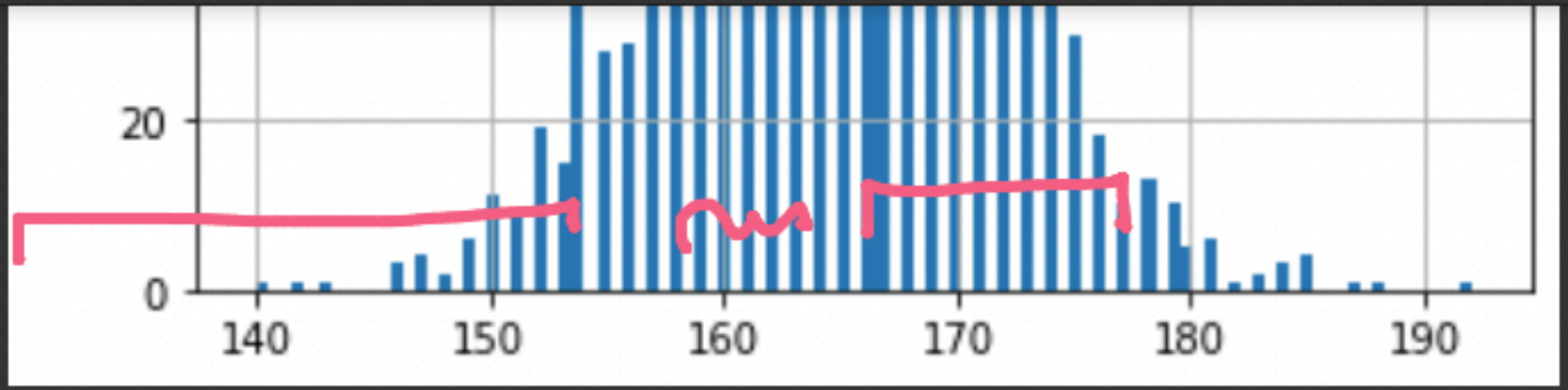


H : discrete or continous

182.5cm real

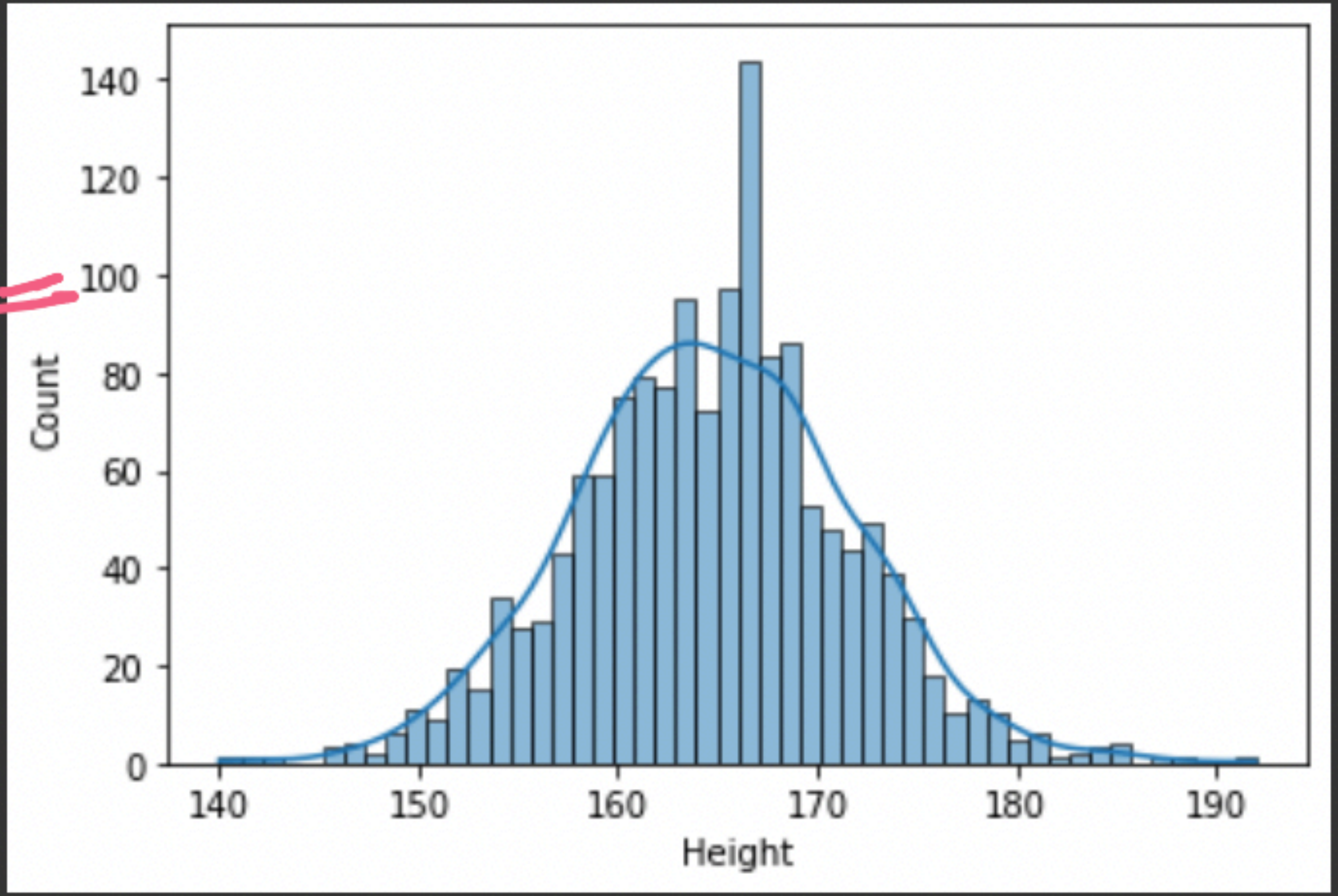
```
# lets get mean and std-dev from the data since we dont know population mean and std-dev
```

[9]




```
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>

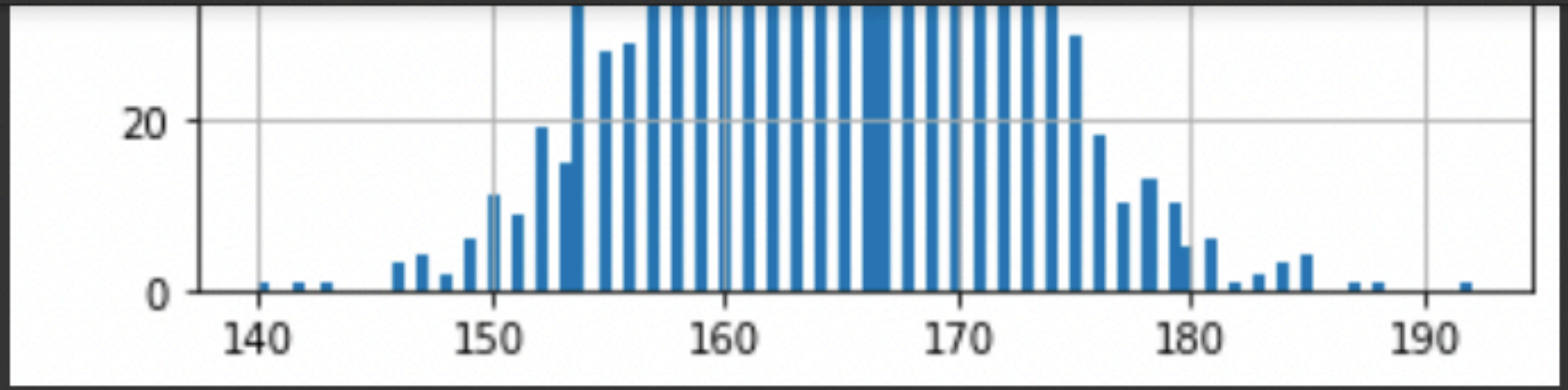


shape histogram ~ shape pdf

freq/count

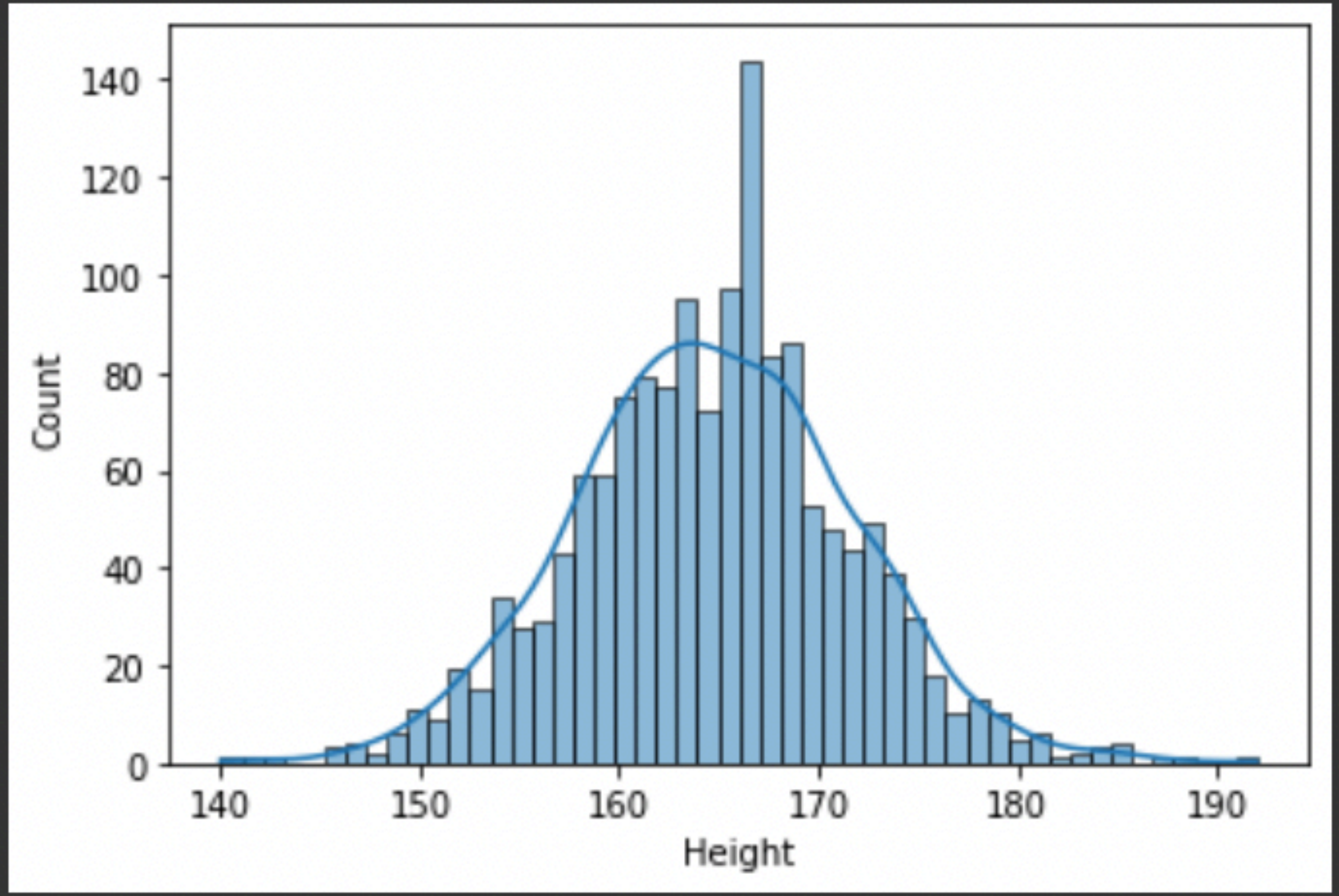
y-axis ?
relative likelihood

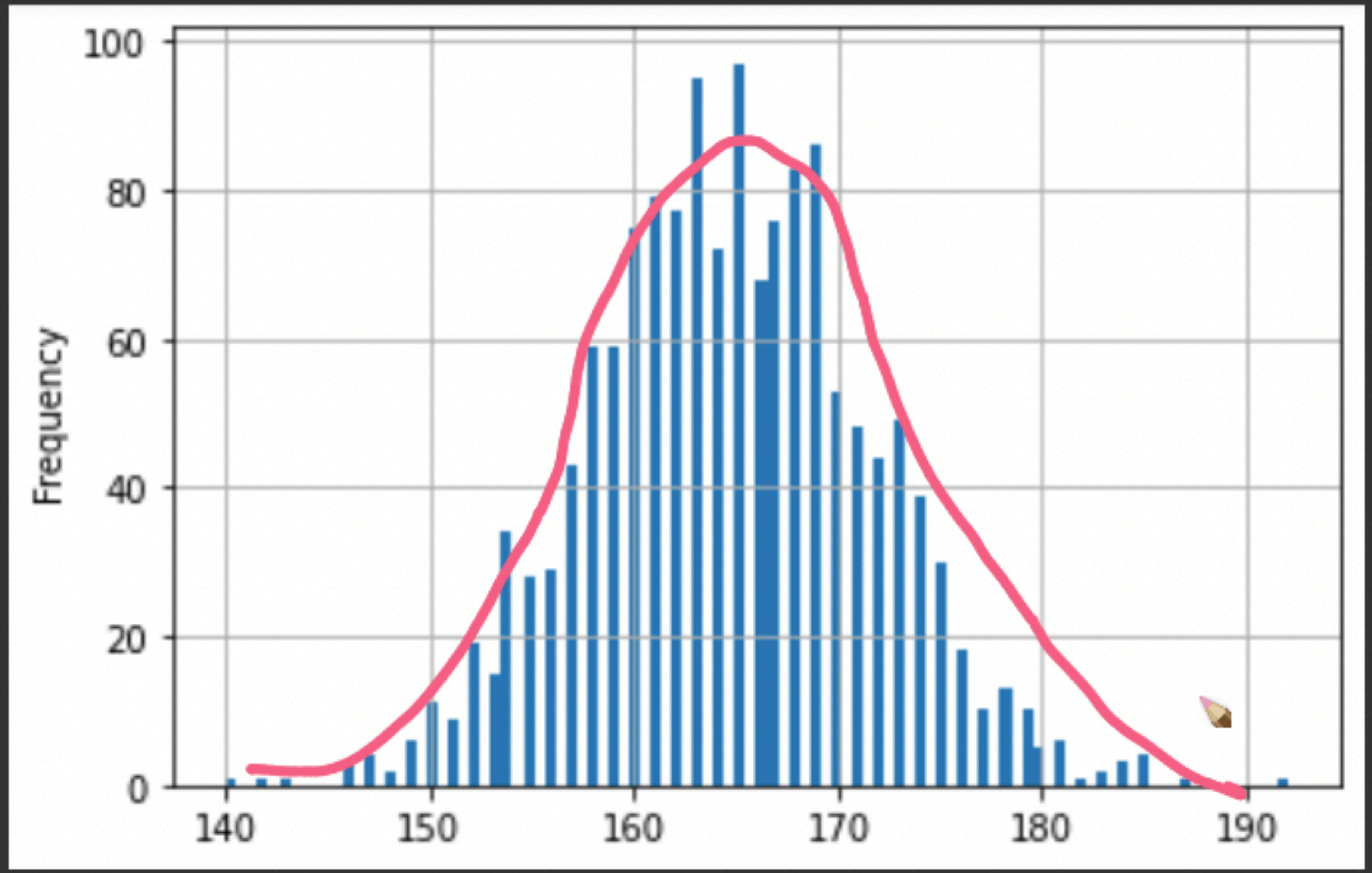
[9]




```python
sns.histplot(employees['Height'], bins=50, kde=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>
```



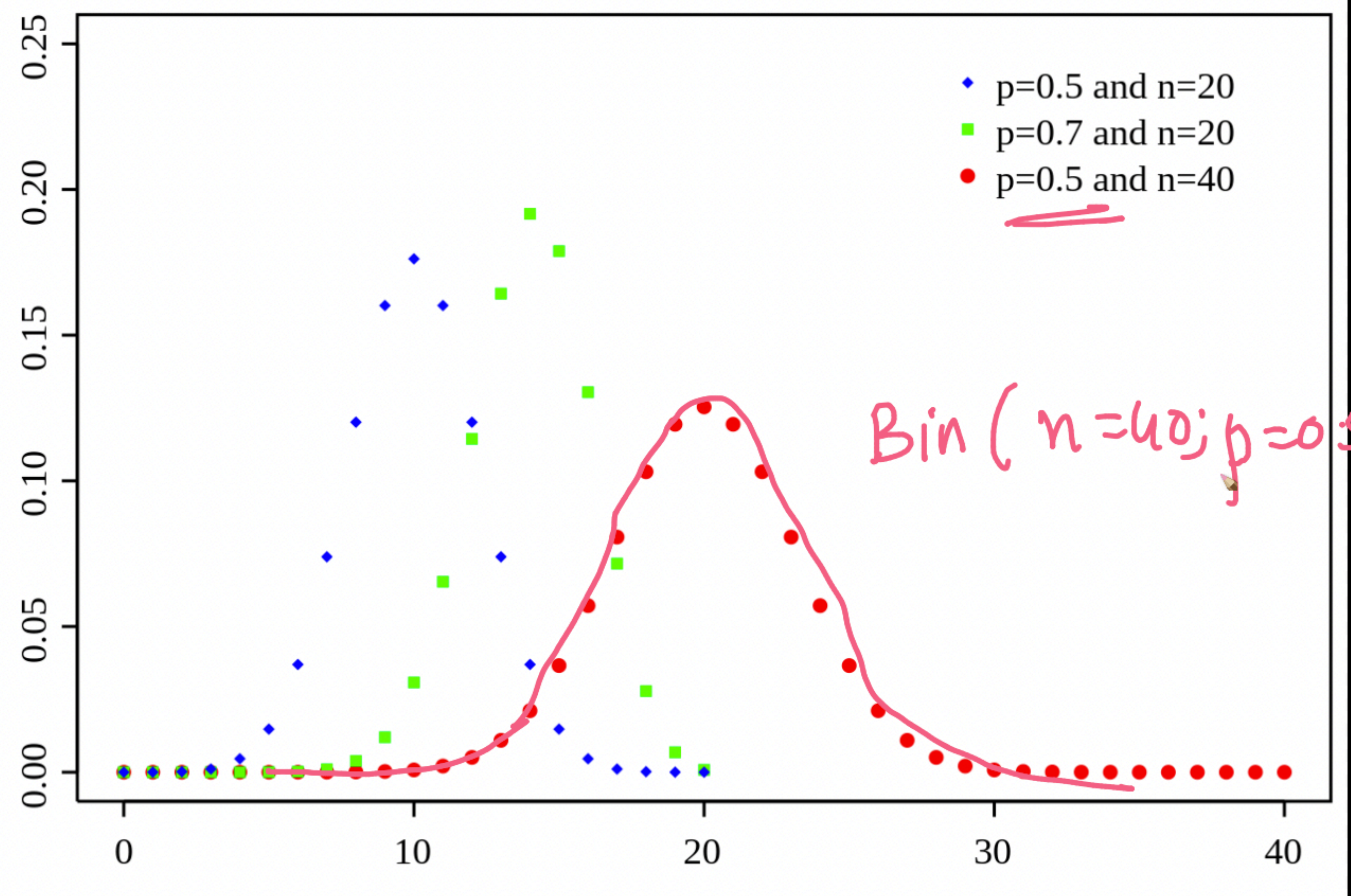



```
# lets get mean and std-dev from the data since we dont know population mean and std-dev
# ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev
employees['Height'].mean()
```

164.6734693877551

```
employees['Height'].std()
```

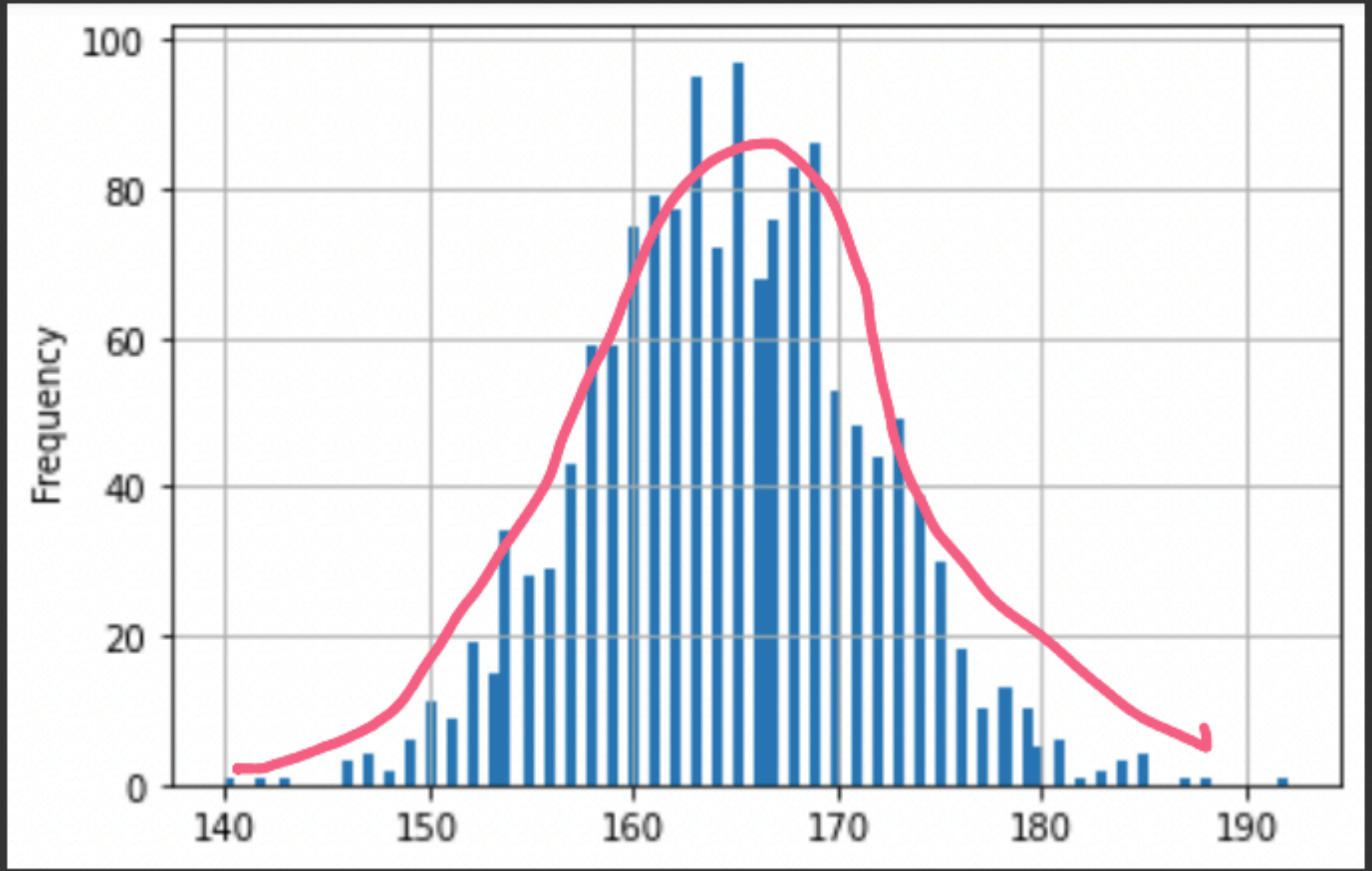
6.887961959078209



Probability mass function for the binomial distribution

More details

Tayste - Own work



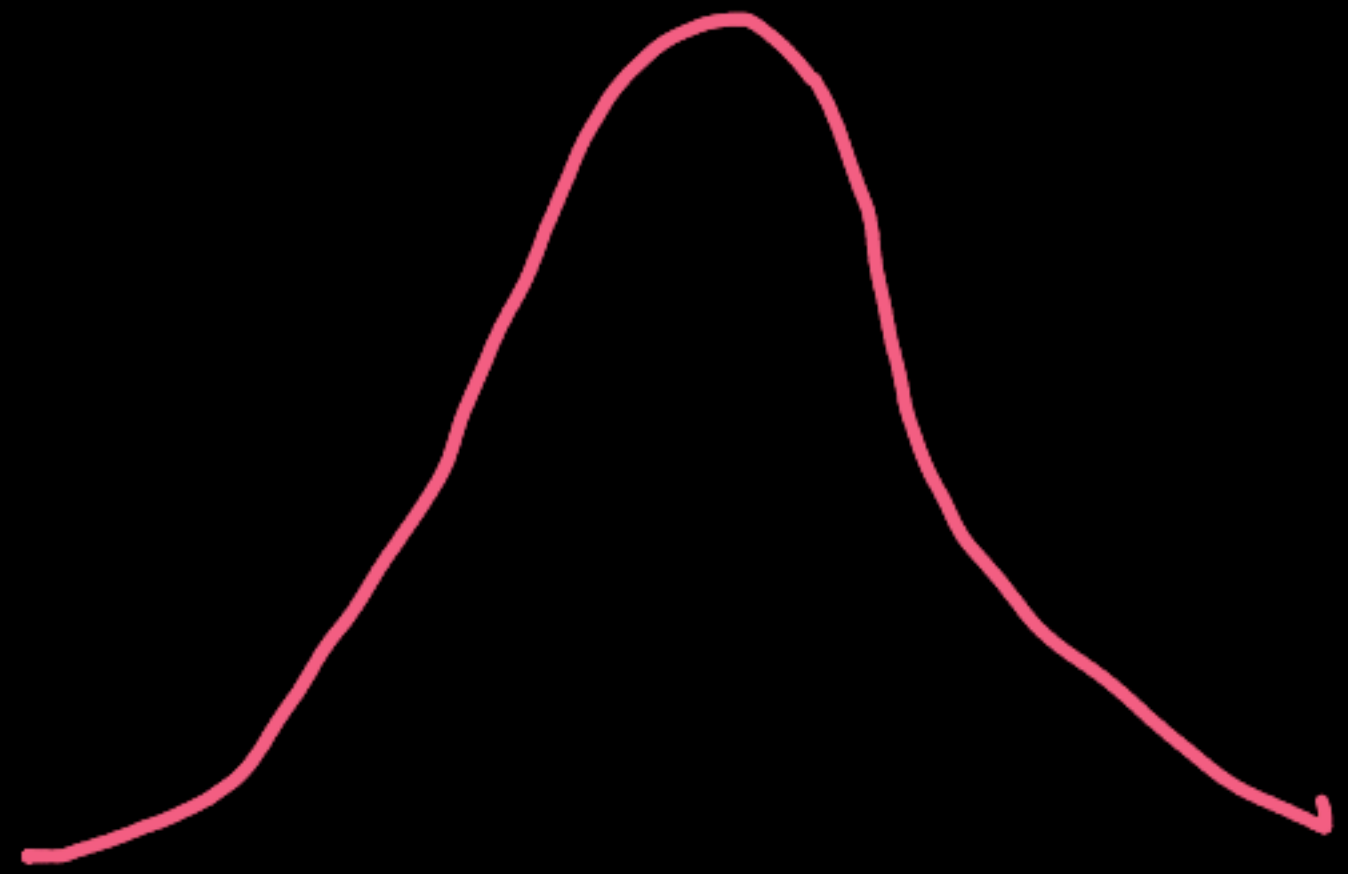
H ~ Binomial r.v (?)

```
# lets get mean and std-dev from the data since we dont know population mean and std-dev
# ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev
employees['Height'].mean()
```

164.6734693877551

```
employees['Height'].std()
```

6.887961959078209

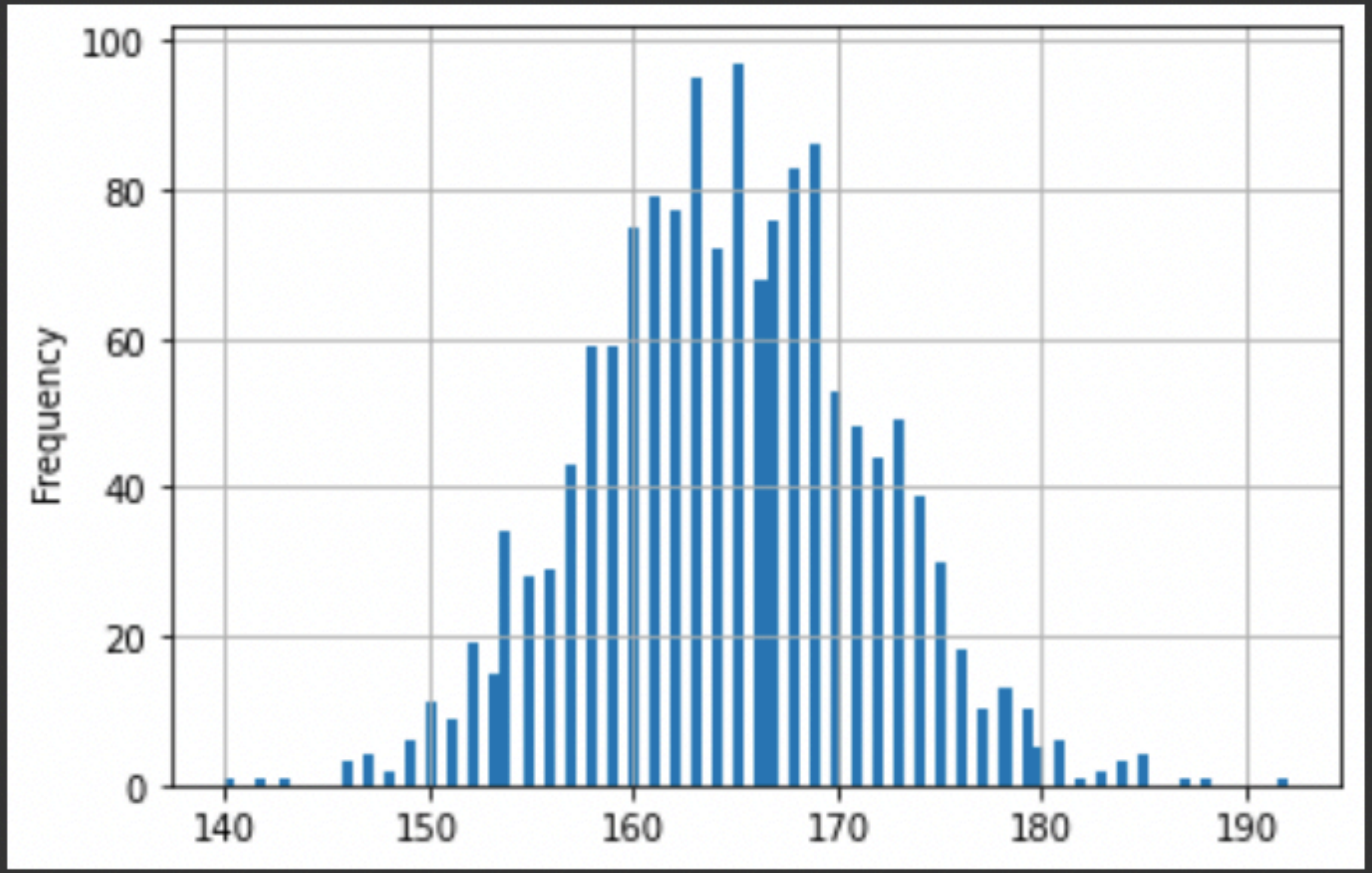


H ~ Binomial r.v (?)

- discrete
- $\boxed{\text{count}}$ of #success in n indep bernouli trials

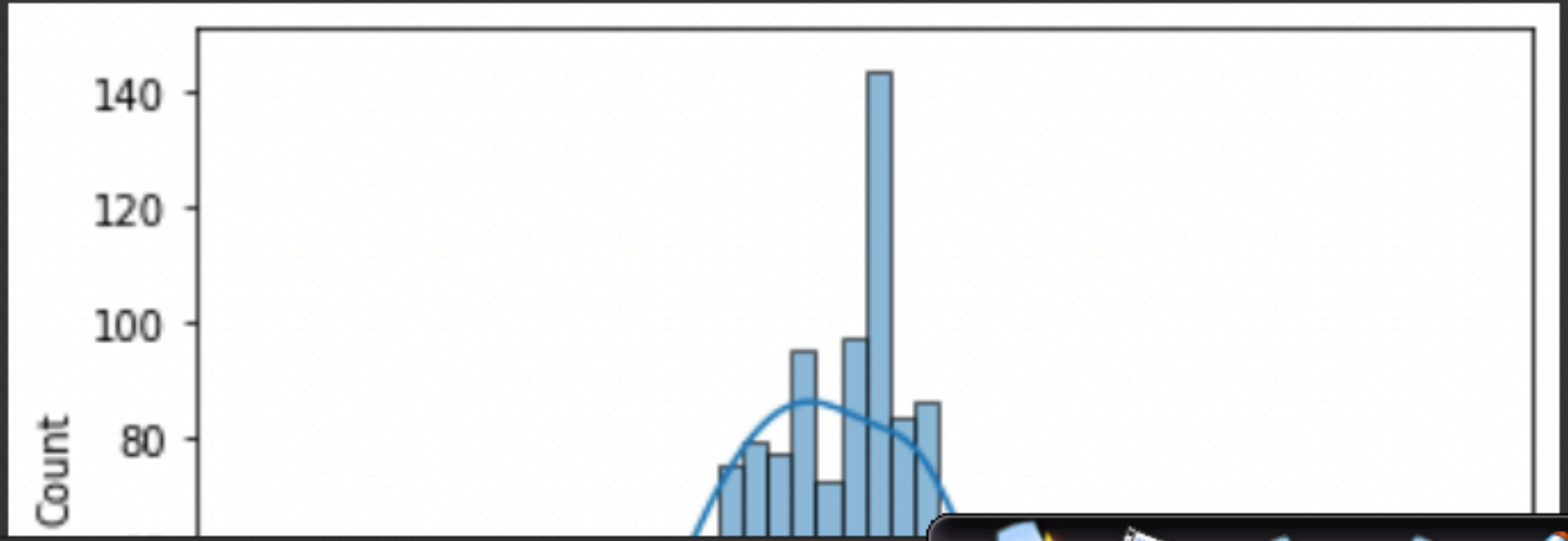
p: 0.5

[9]



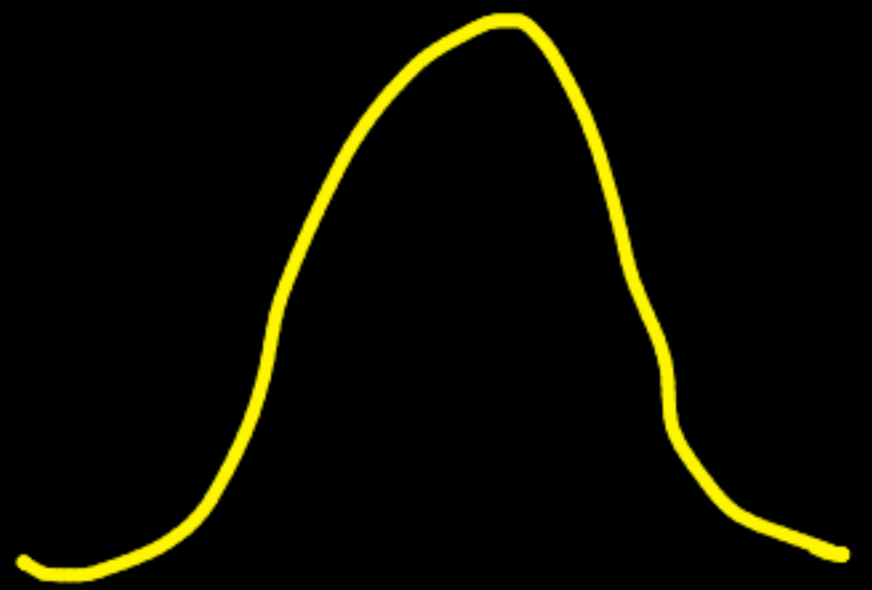

```python
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>

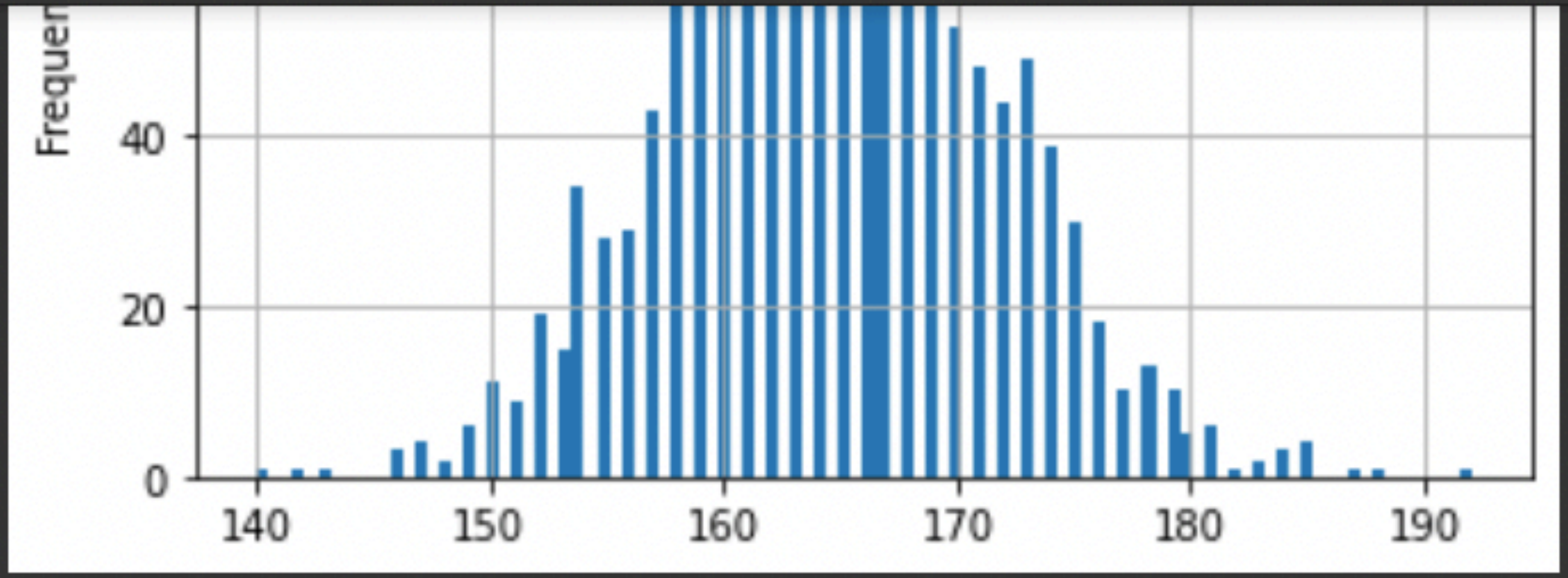


lesson: shape looks the same

the dist matches

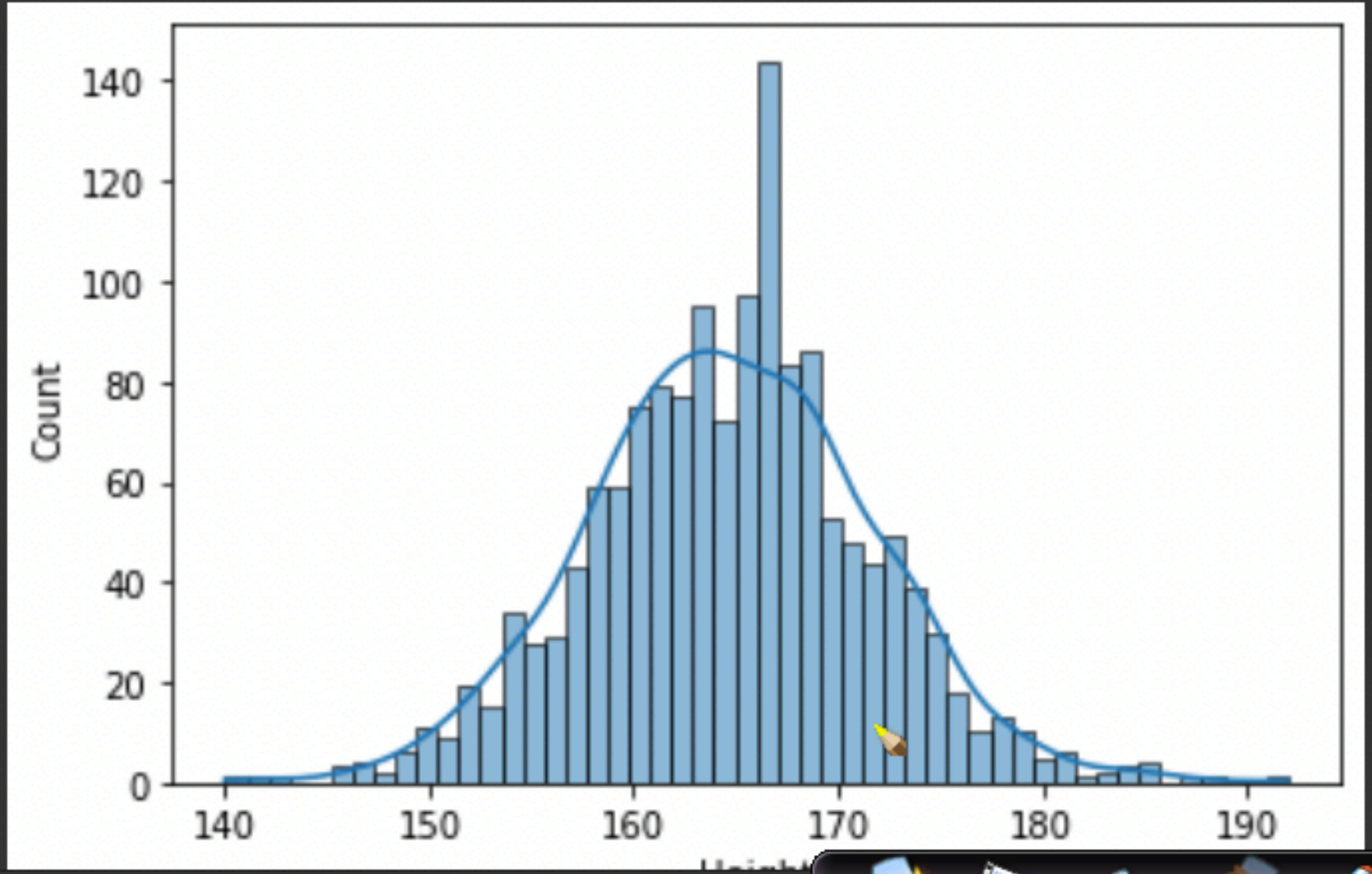


[9]




```
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>



Chrome File Edit View History Bookmarks Profiles Tab Window Help 21:37

Two simple ways to determine the distb

1 → Check for the conditions defining a distb

binomial : counts

2 → PMF/PDF or CDFs → Statistical Tests (KS-test : later)

→ Visual : QQ plot (later)

15 / 18

Chrome File Edit View History Bookmarks Profiles Tab Window Help 21:30

$X \sim$ Bernouli (n, p)

Disb: concise model of the disb. of data from a natural phenomenon

Stats

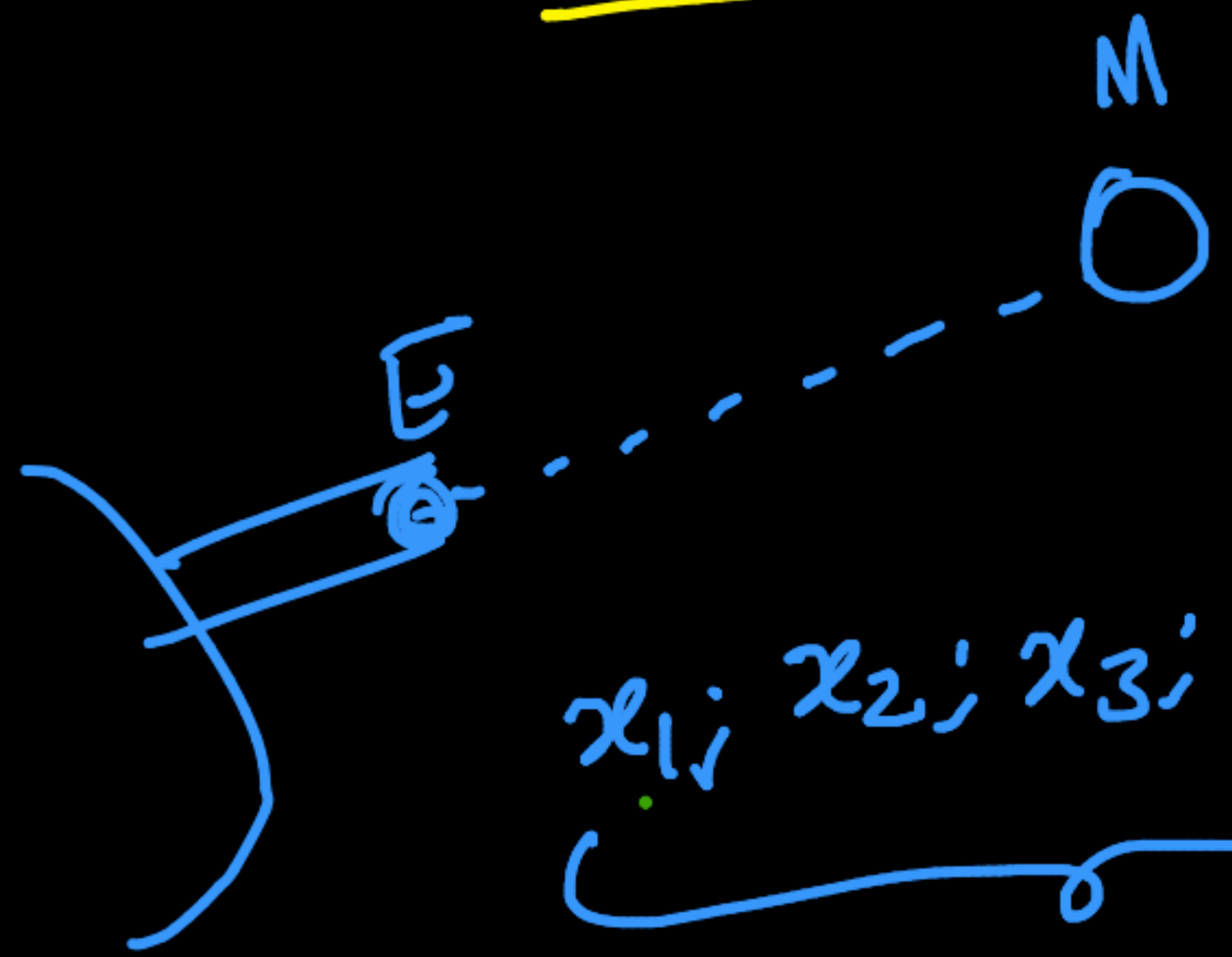
PDF & CDF

data

$x_1, x_2, \ldots x_{100}$

16 / 16

Normal
Gaussian



$x_1; x_2; x_3; \dots x_n$

x_{act}

heights

{Birth weights

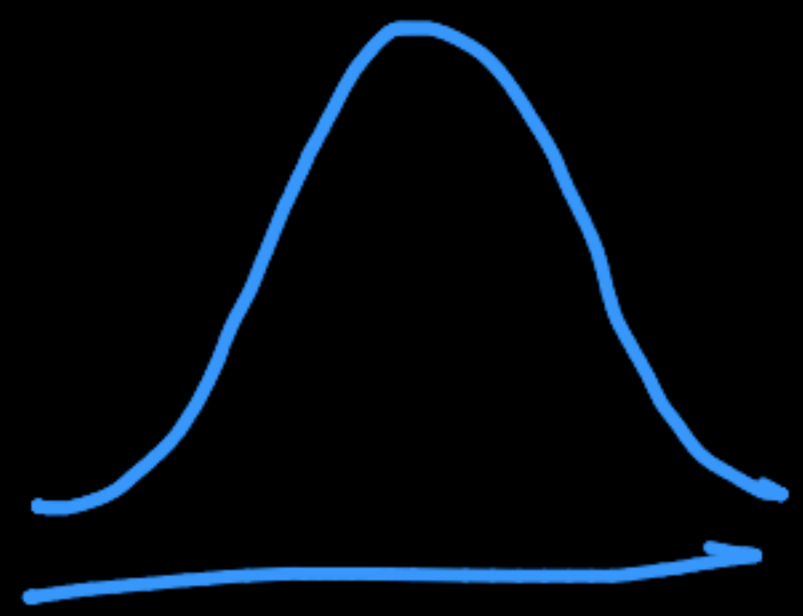
weights

shoe-sizes → feet-size

size of leaves/petals

measurement-errors

$e_1, e_2, \dots e_n$



Conditions: → Bernoulli ✓ → derived ...

Gaussian-dist → also deriv

intuition:

→ many random things

loosely

Practical: $x_1, x_2, \dots x_{100}$

↓

KS-test

AD-test

⋮

Stats-tests

e.g: { blood-pressure: Normal

$H \sim \text{Normal}(\mu, \sigma)$ → population

Binomial (n, p)

Bernouli (p)

X → $[x_1, x_2, x_3 \dots x_{100}]$: sample

mean of sample $= m$

std-dev of sample $= s$

sample size ↑

$m \approx \mu$

$s \approx \sigma$

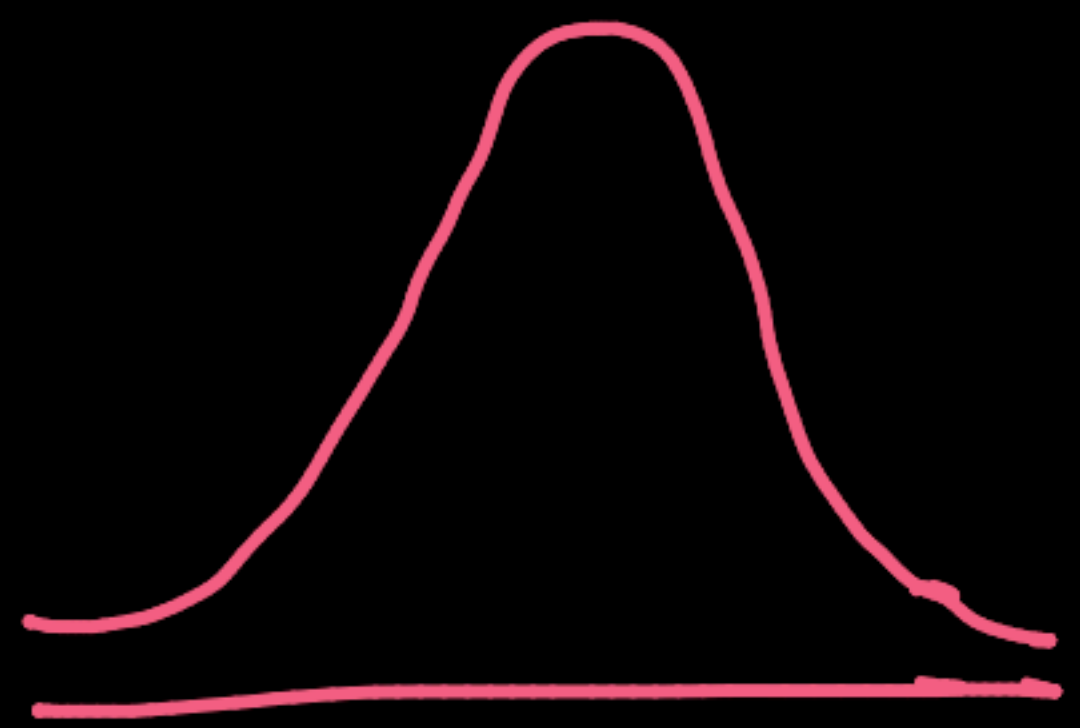
T-shirts

1000 - employees - heights

✓ $H \sim N(\underset{m}{\mu}, \underset{s}{\sigma})$ already verified by experts

Shape of PDF of Normal(μ, σ)

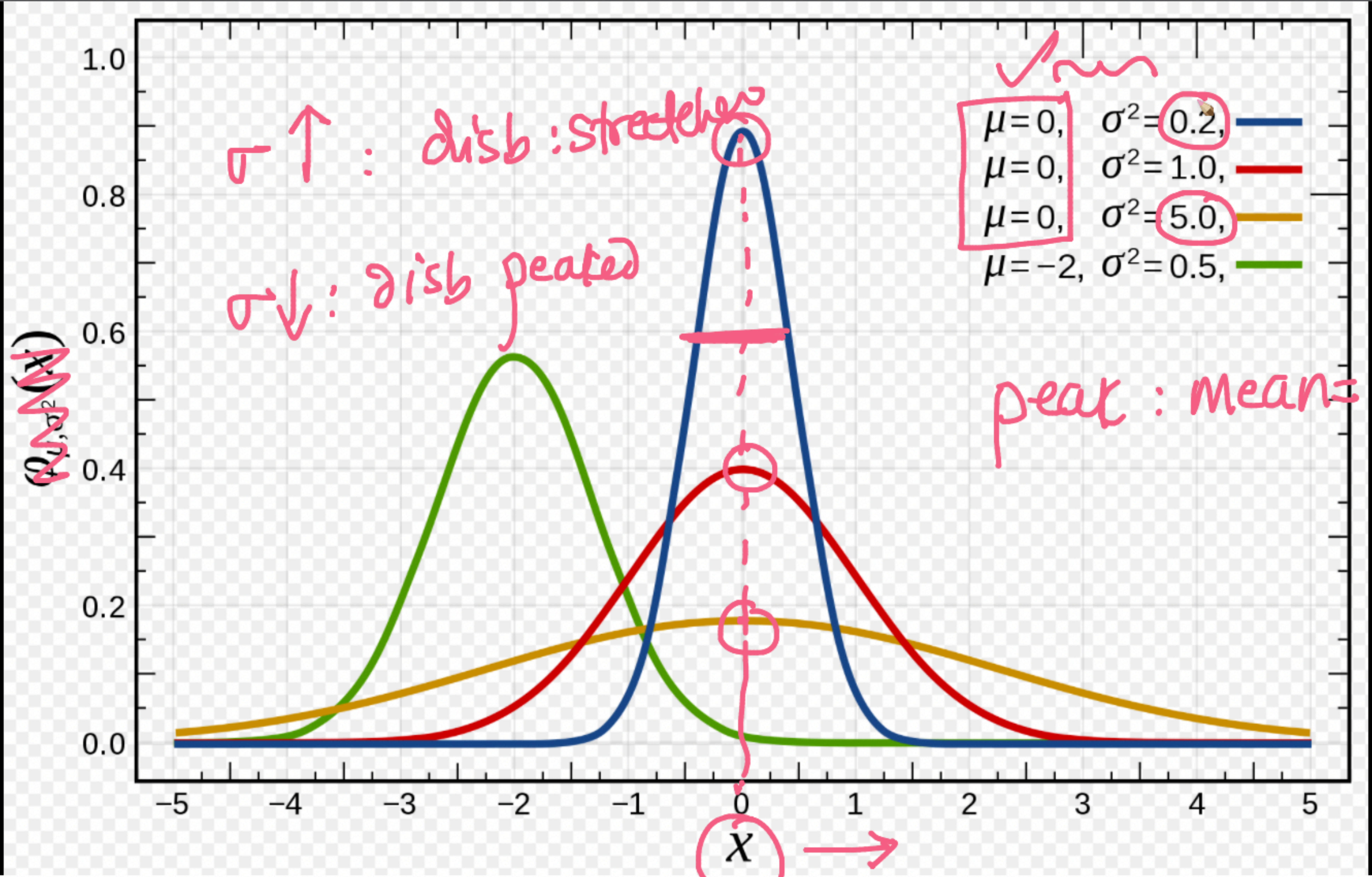
bell-shaped



$H \sim N(\mu, \sigma)$

continous

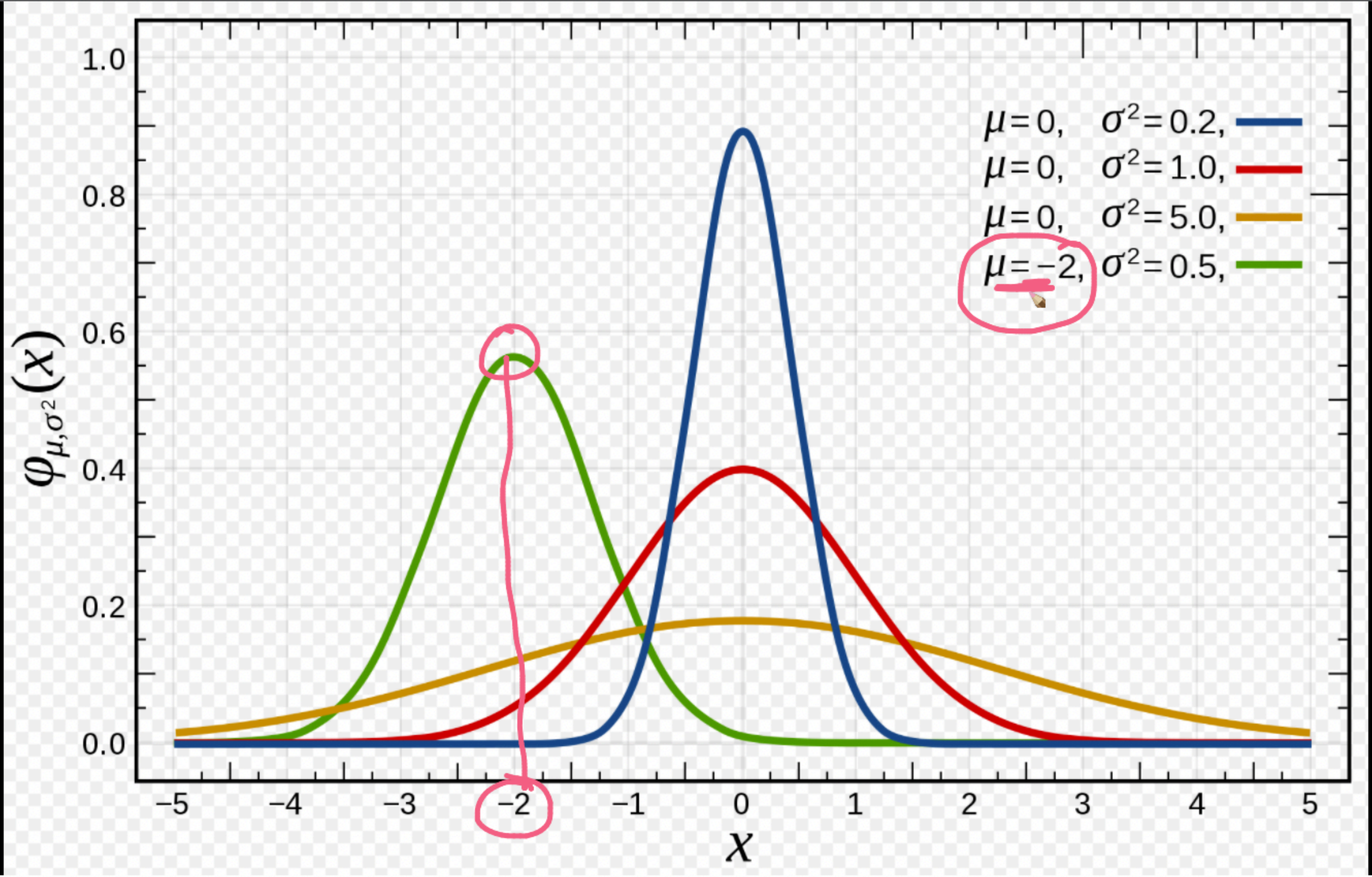
PDF



A selection of Normal Distribution Probability Density Functions (PDFs). Both the mean, μ, and variance, σ^2, are varied. The key is given on the graph.

More details

Inductiveload - self-made, Mathematica, Inkscape

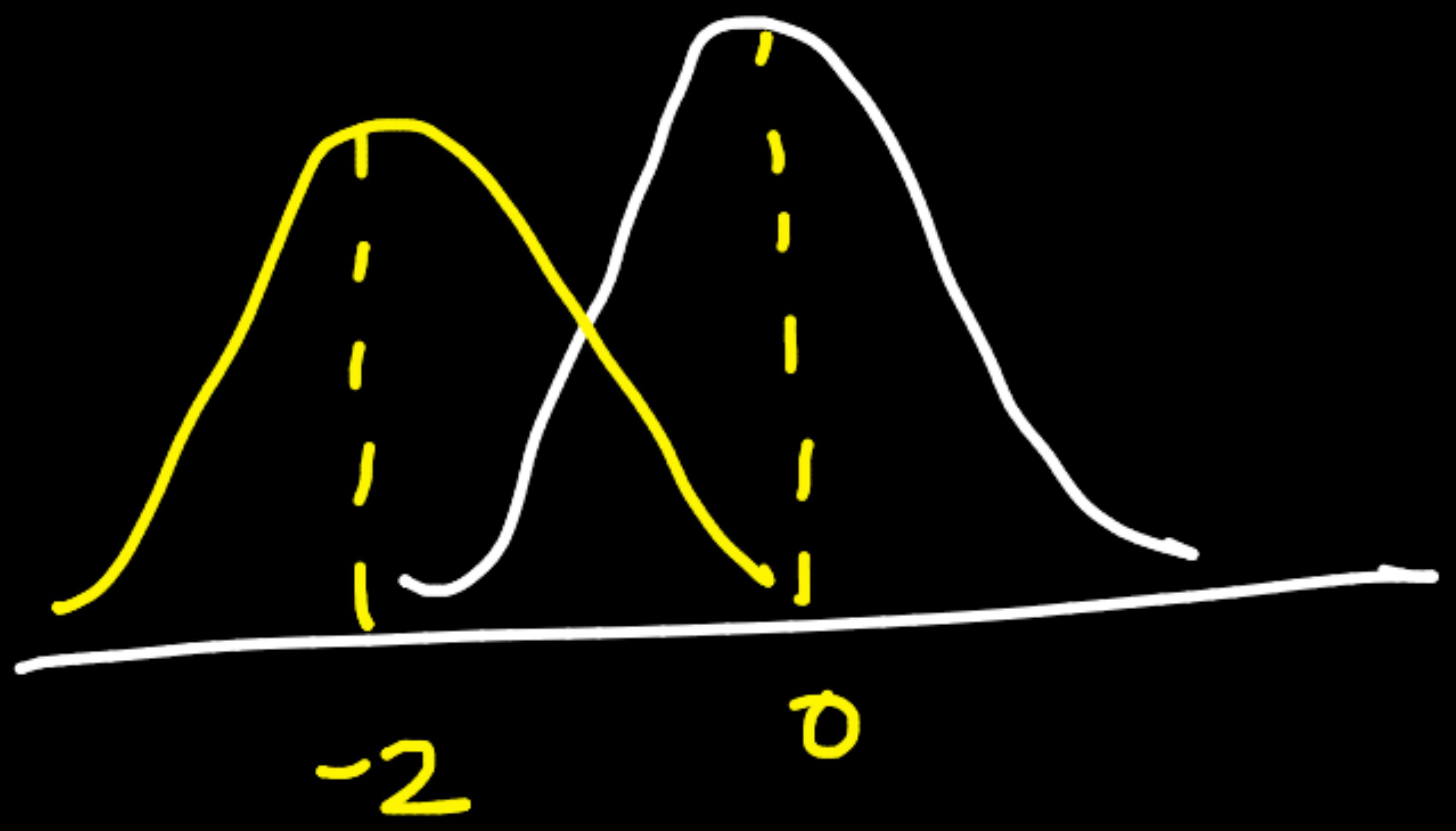


A selection of Normal Distribution Probability Density Functions (PDFs). Both the mean, μ, and variance, σ^2, are varied. The key is given on the graph.

More details

$N(\mu, \boxed{\sigma})$

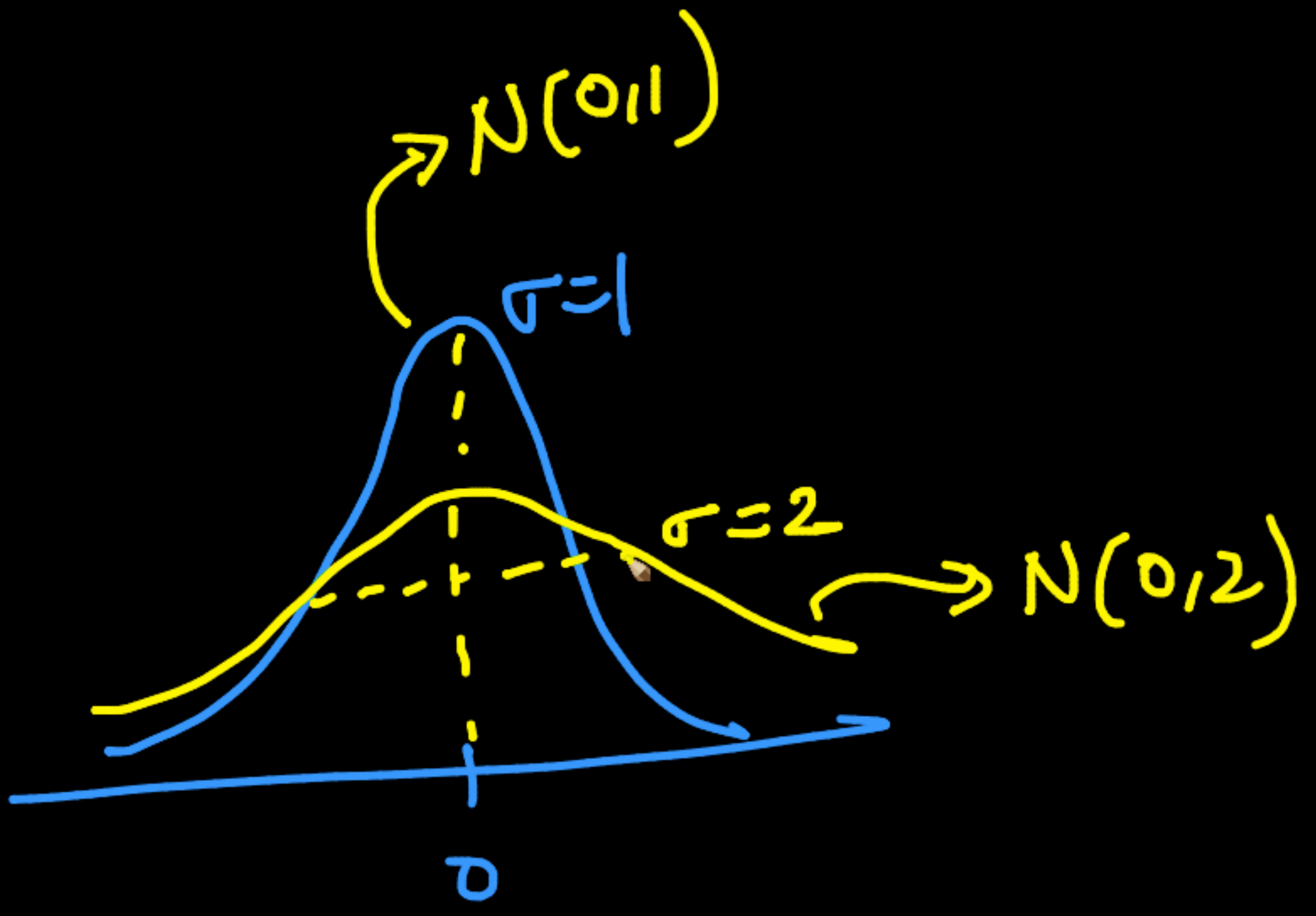
peak



$N(\mu=0, \sigma=1)$ → std-dev

$N(\mu=0; \sigma=2)$

spread / variability



INTUITION

- peak @ μ
- σ determines peakedness or spread
- curves falling exponentially
- Symmetry

$\boxed{Q}$

Sample

reasonable size

non-random

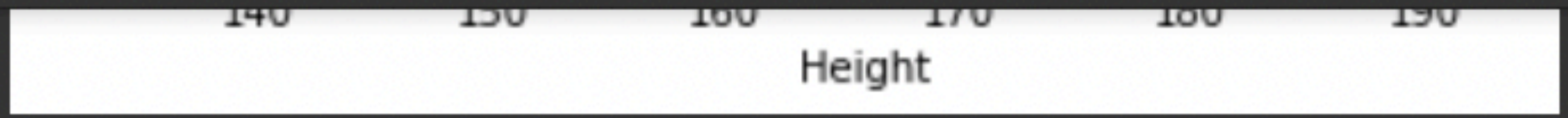
$m, s \neq \mu, \sigma$

$x_1\ x_2 \dots x_{100} \rightarrow \boxed{\text{disb}} \rightarrow$ outliers better

disb does not fr an ML model

(RF/GDBT later)

not-needed




```
[ ] # lets get mean and std-dev from the data since we dont know population mean and std-dev

    # ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev

    employees['Height'].mean()
```

```
164.6734693877551
```

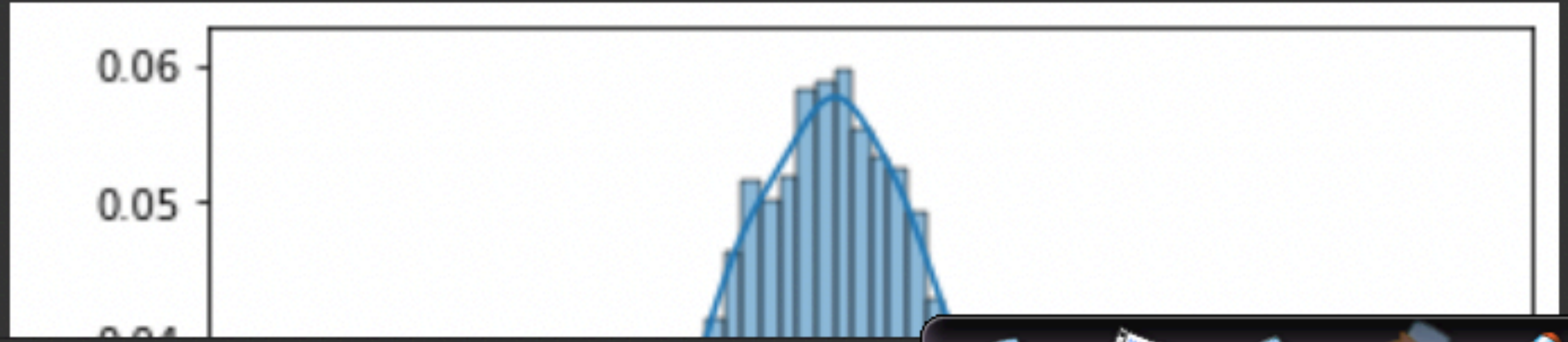
```
[ ] employees['Height'].std()
```

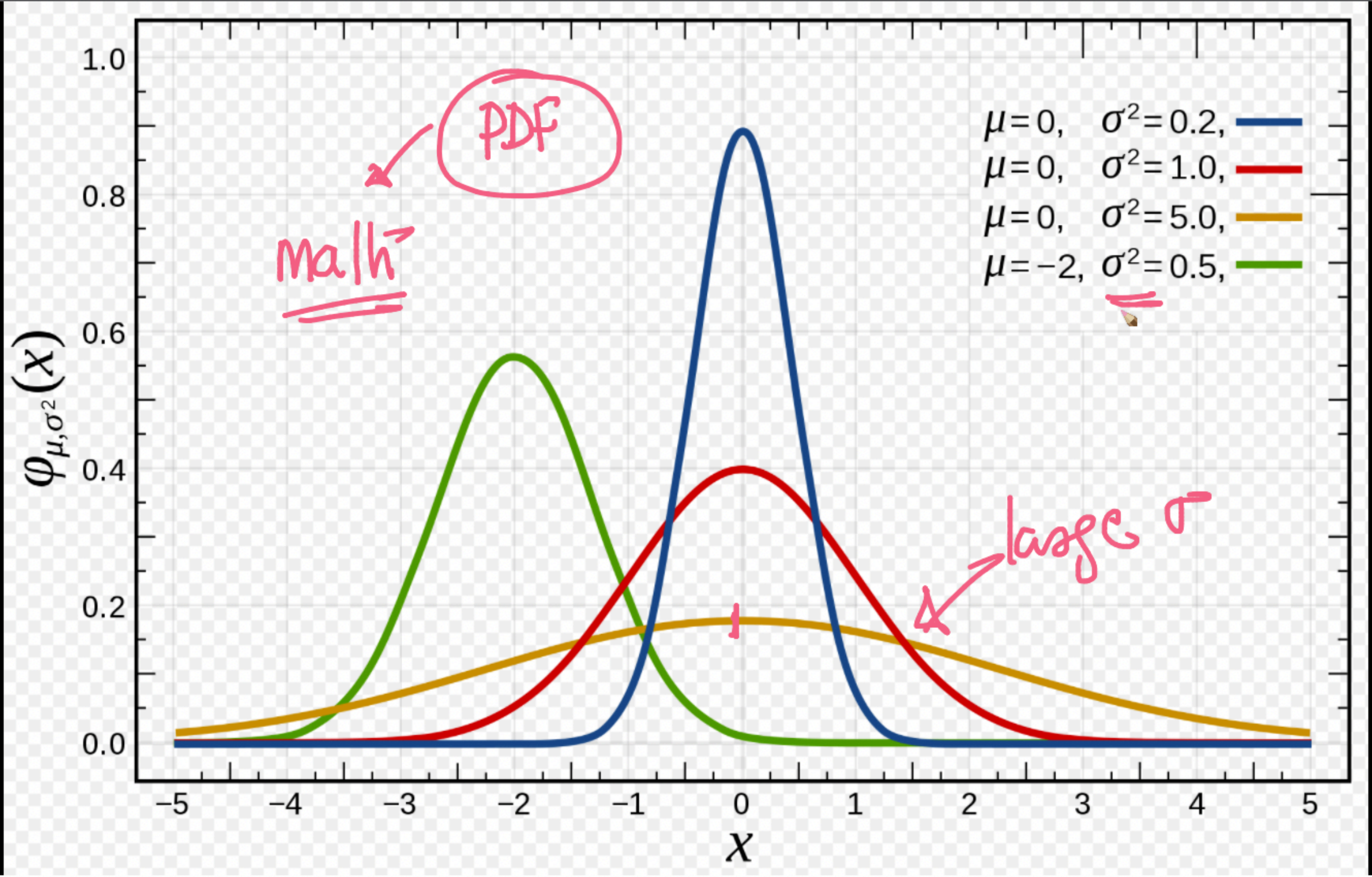
```
6.887961959078209
```

```
[ ] height_dist = stats.norm(loc=employees["Height"].mean(), scale = employees["Height"].std())
```

```
sns.histplot(height_dist.rvs(10000), kde=True , stat='density')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6c28c31710>
```





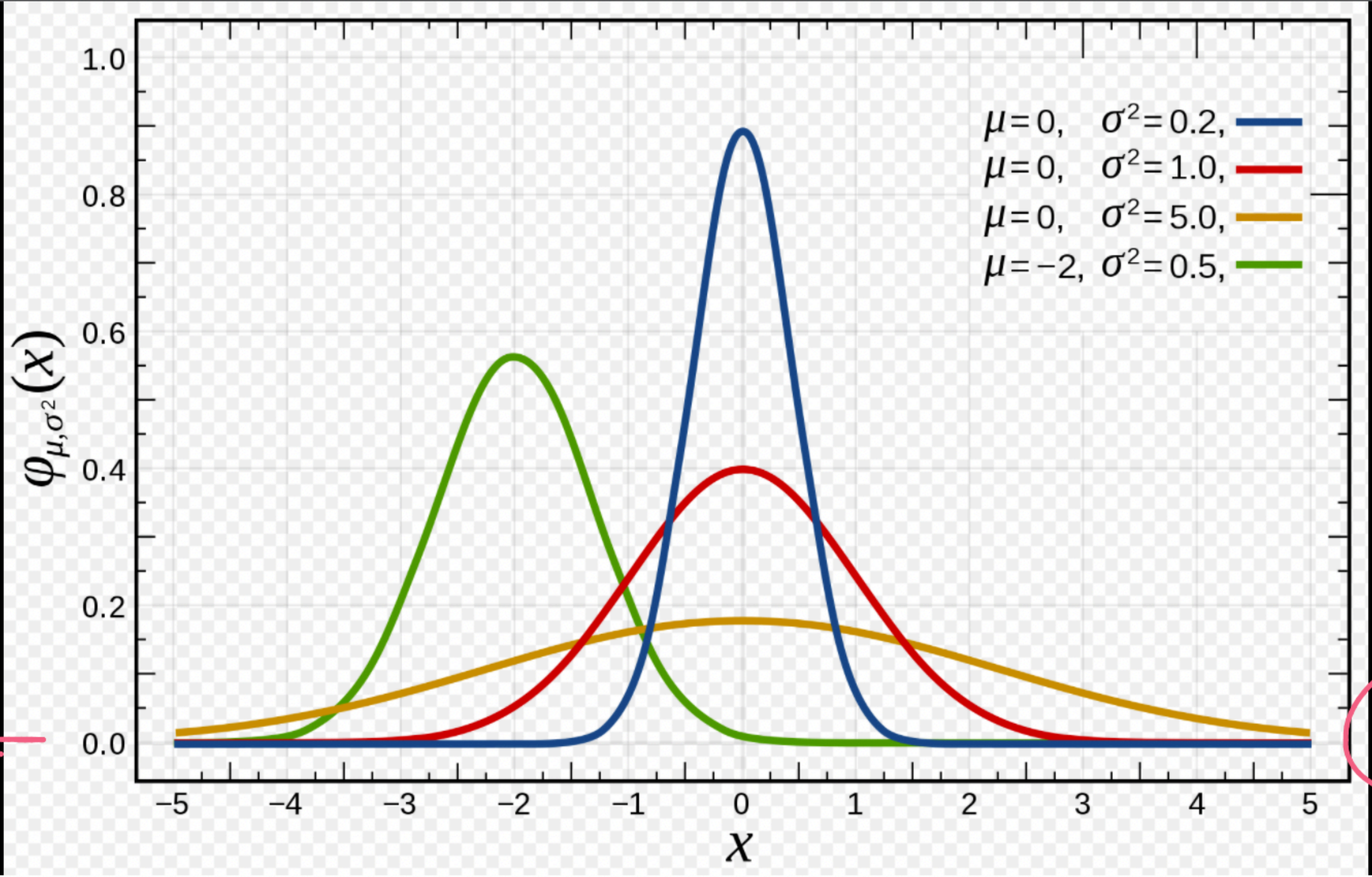
A selection of Normal Distribution Probability Density Functions (PDFs). Both the mean, μ, and variance, σ^2, are varied. The key is given on the graph.

More details

Inductiveload - self-made, Mathematica, Inkscape

en.wikipedia.org/wiki/Normal_distribution#/media/File:Normal_Distribution_PDF.svg

$-\infty$



∞

A selection of Normal Distribution Probability Density Functions (PDFs). Both the mean, μ, and variance, σ^2, are varied. The key is given on the graph.

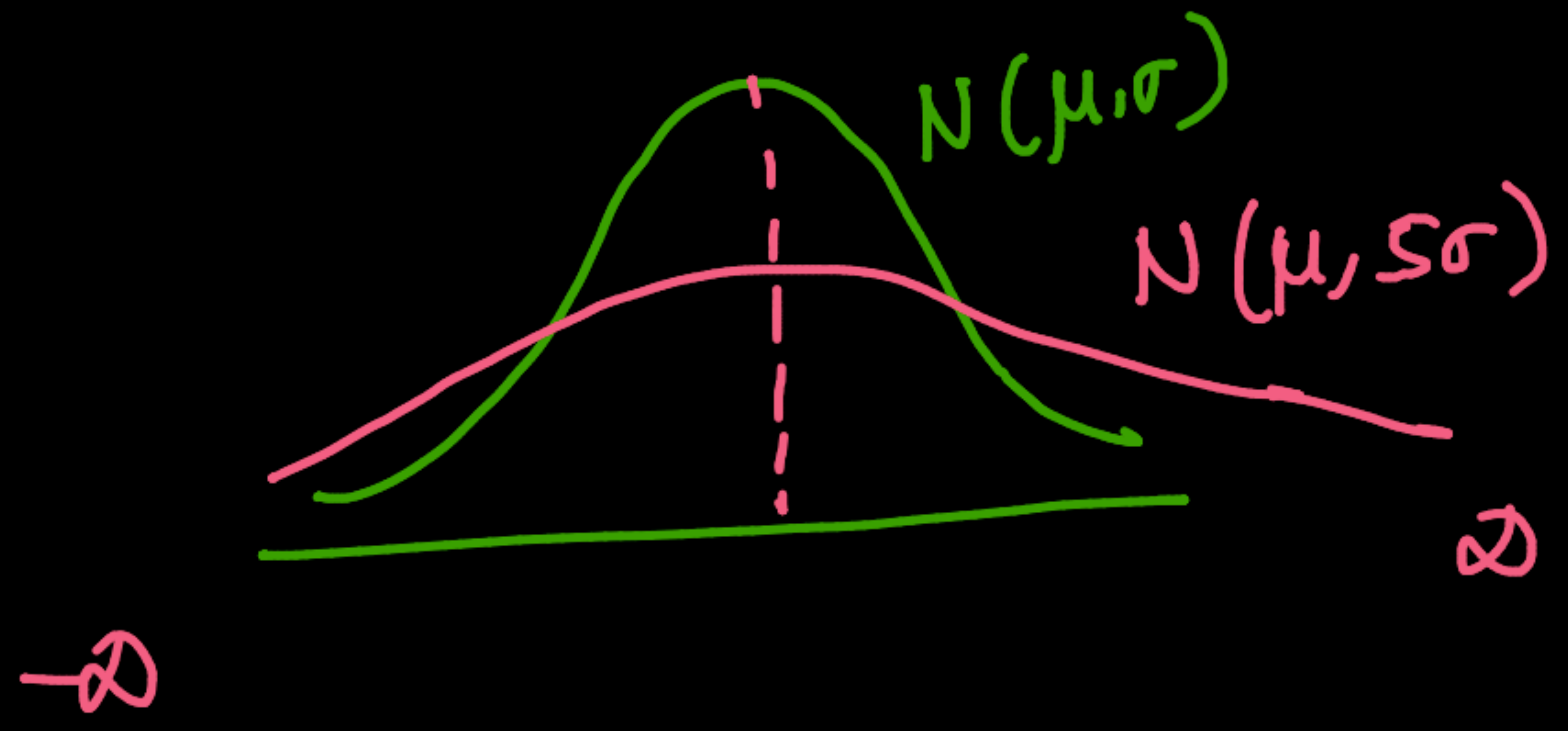
More details

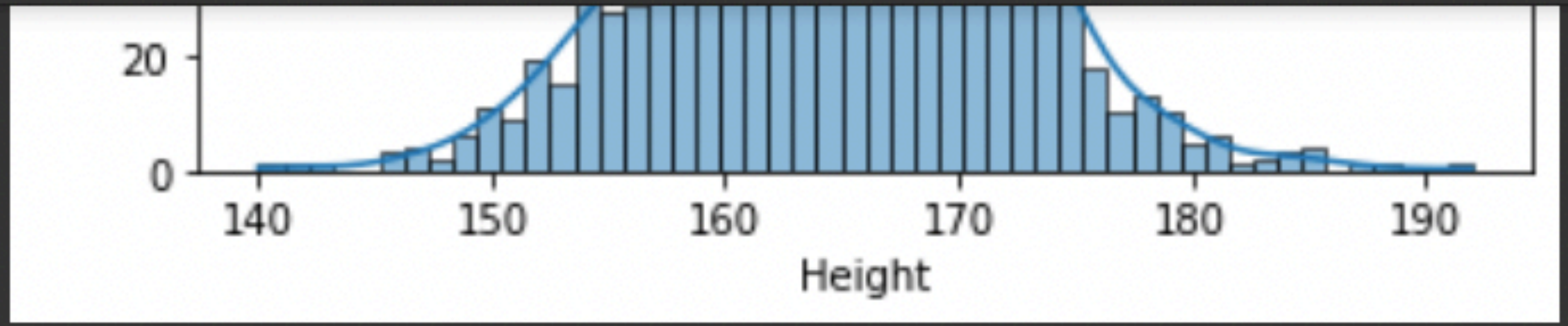
Inductiveload - self-made, Mathematica, Inkscape

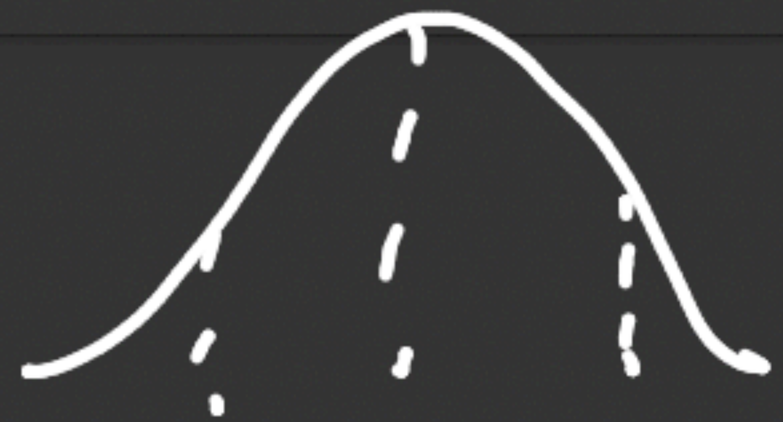
$\boxed{Q}$

area under the curve

$\boxed{1}$ ✓







```
[ ] # lets get mean and std-dev from the data since we dont know population mean and std-dev

    # ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev

    employees['Height'].mean()
```

164.6734693877551

Model

$$H \sim \boxed{N}(\mu = 164.67;\ \sigma = 6.89)$$

PDF & CDF

```
[ ] employees['Height'].std()
```

6.887961959078209

```
[ ] height_dist = stats.norm(loc=employees["Height"].mean(), scale = employees["Height"].std())
```

```
sns.histplot(height_dist.rvs(10000), kde=True , stat='density')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6c28c31710>

164.6734693877551

scipy.stats

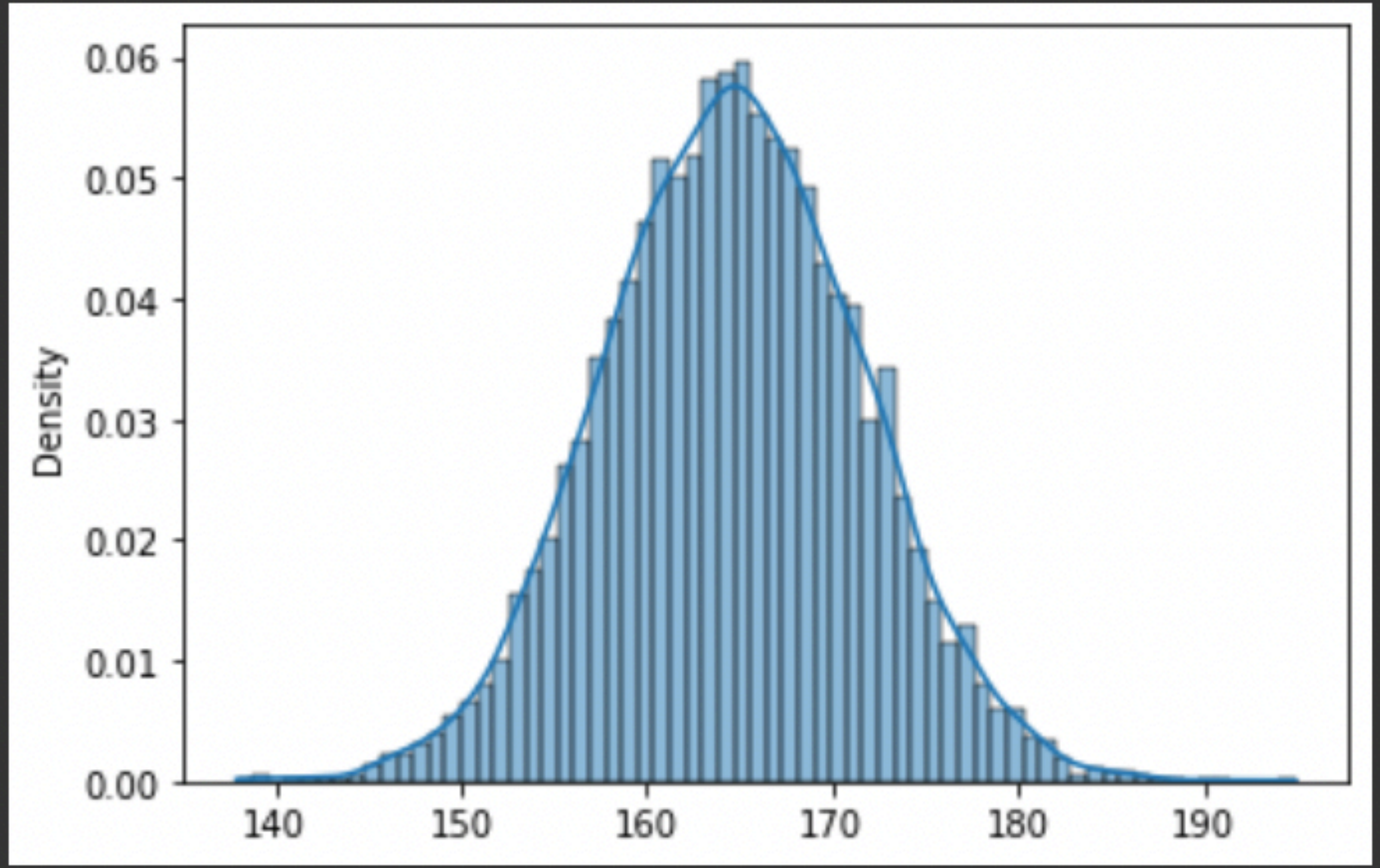
```
[ ] employees['Height'].std()
```

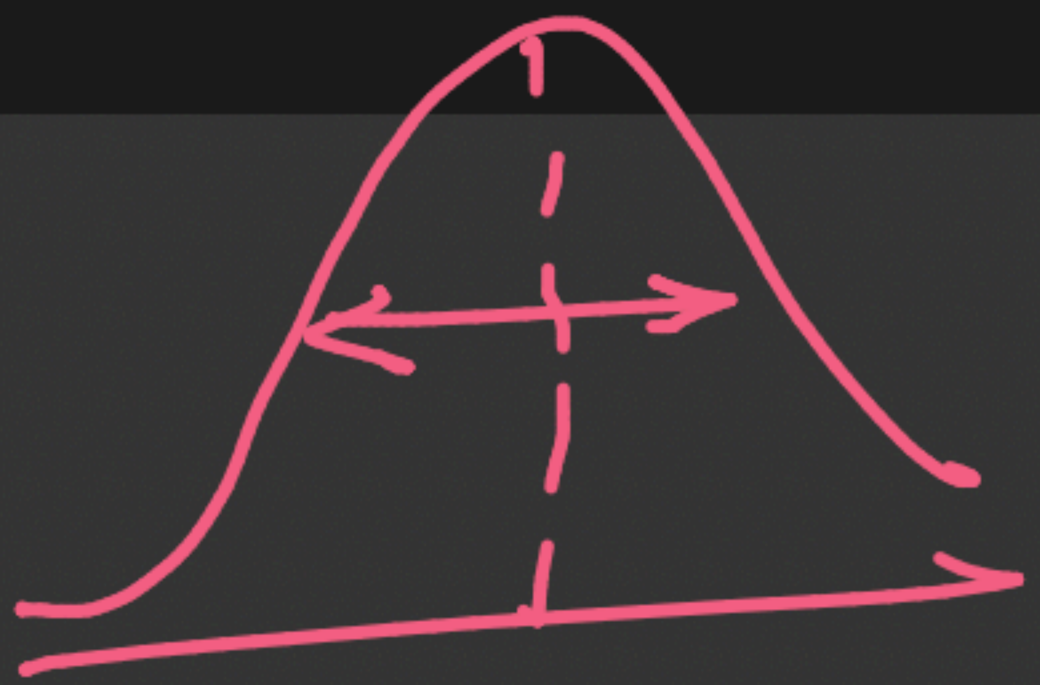
6.887961959078209

```
[ ] height_dist = stats.norm(loc=employees["Height"].mean(), scale = employees["Height"].std())
```

```
sns.histplot(height_dist.rvs(10000), kde=True , stat='density')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6c28c31710>





```
164.6734693877551
```

$164.67, 6.89$

$N(\mu, \sigma)$

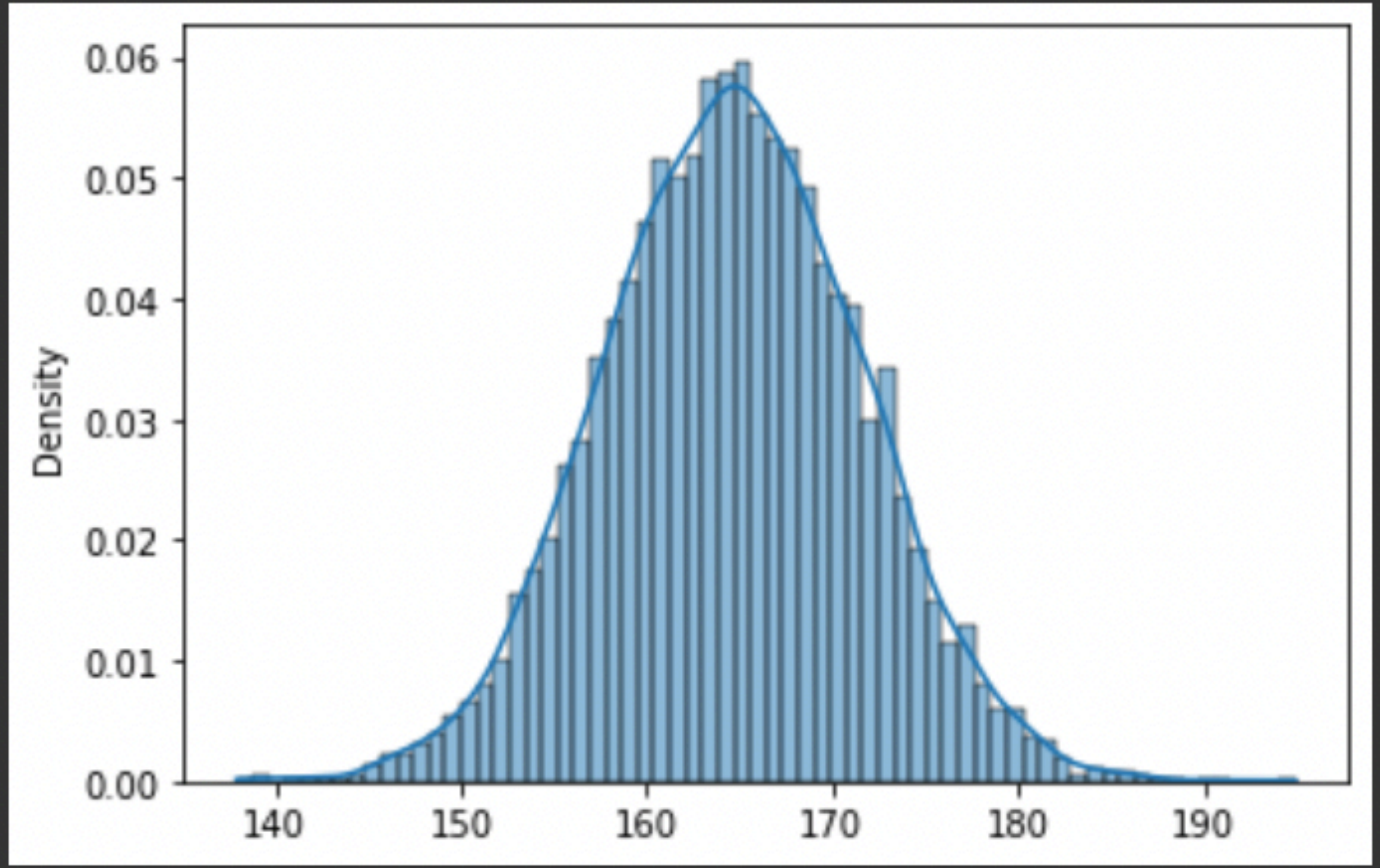
```
employees['Height'].std()
```

```
6.887961959078209
```

```
height_dist = stats.norm(loc=employees["Height"].mean(), scale = employees["Height"].std())
```

```
sns.histplot(height_dist.rvs(10000), kde=True , stat='density')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6c28c31710>
```



$\underbrace{h_1, h_2, \ldots h_{10,000}}$

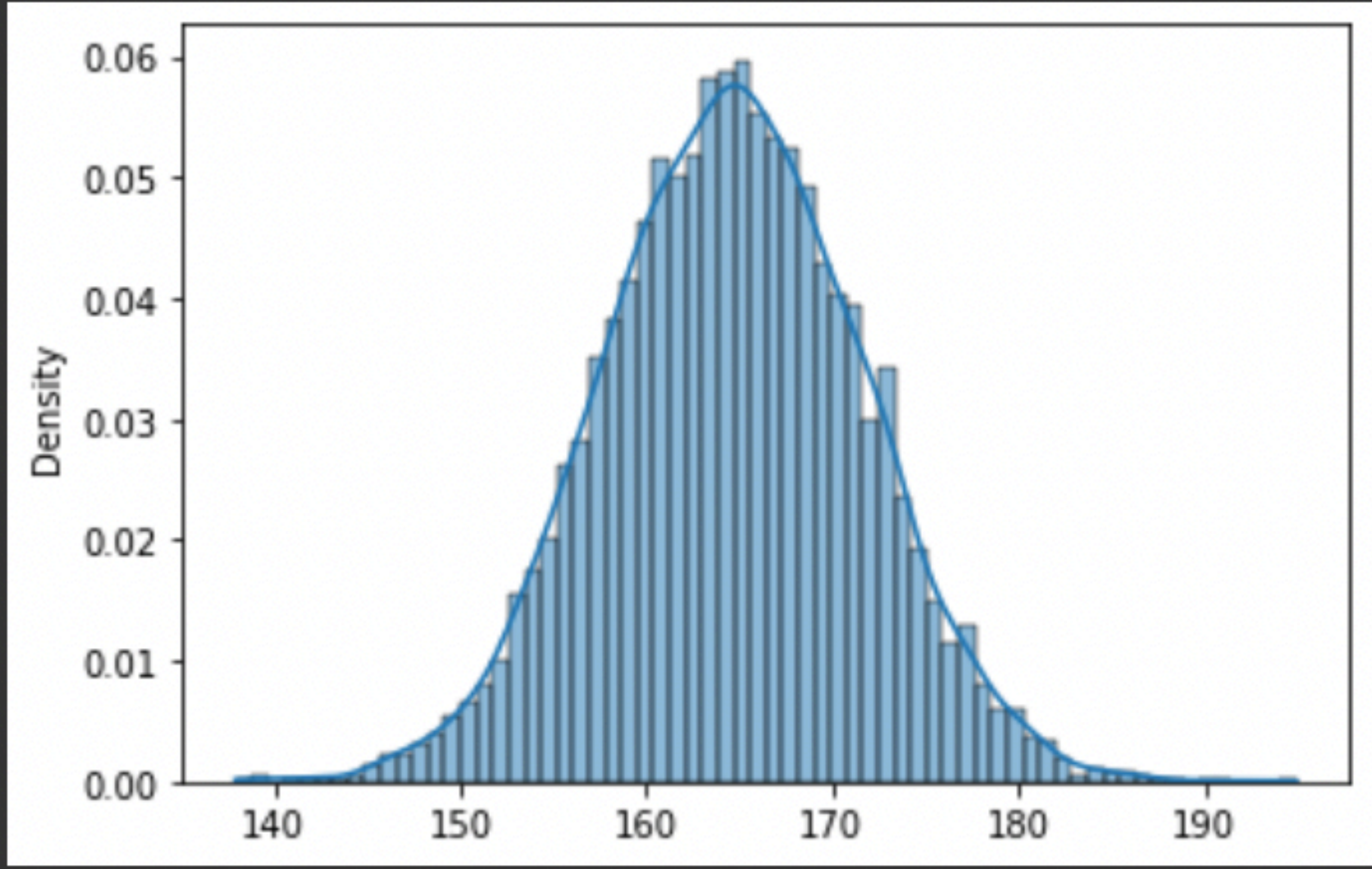
```
employees['Height'].std()
```

6.887961959078209

```
height_dist = stats.norm(loc=employees["Height"].mean(), scale = employees["Height"].std())
```

```
sns.histplot(height_dist.rvs(10000), kde=True , stat='density')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6c28c31710>




```
x = np.arange(120,200)
```

$\boxed{100K}$

$$H \sim N(\mu = 164.67;\ \sigma = 6.89)$$

$\downarrow$

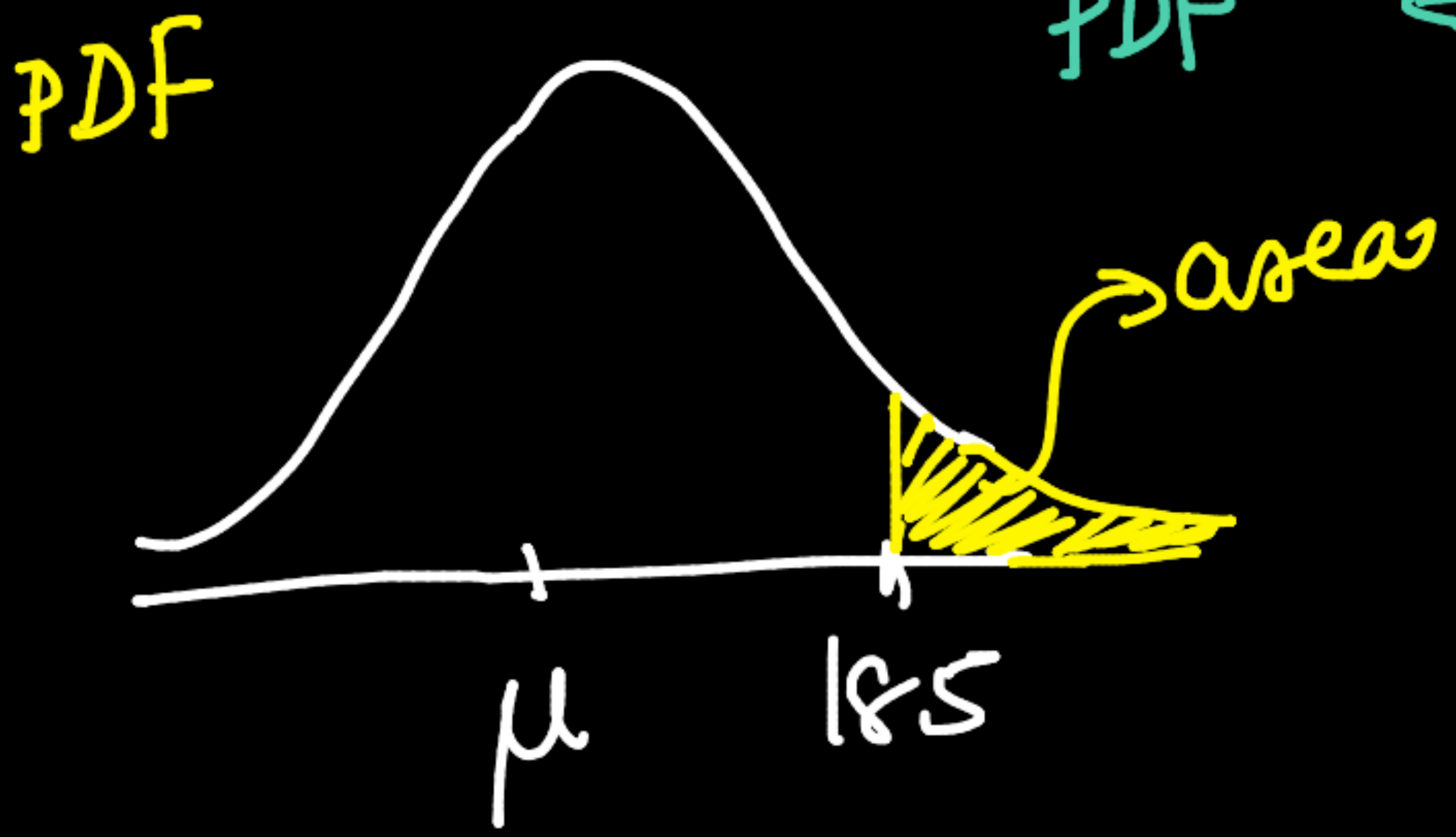
PDF & CDF

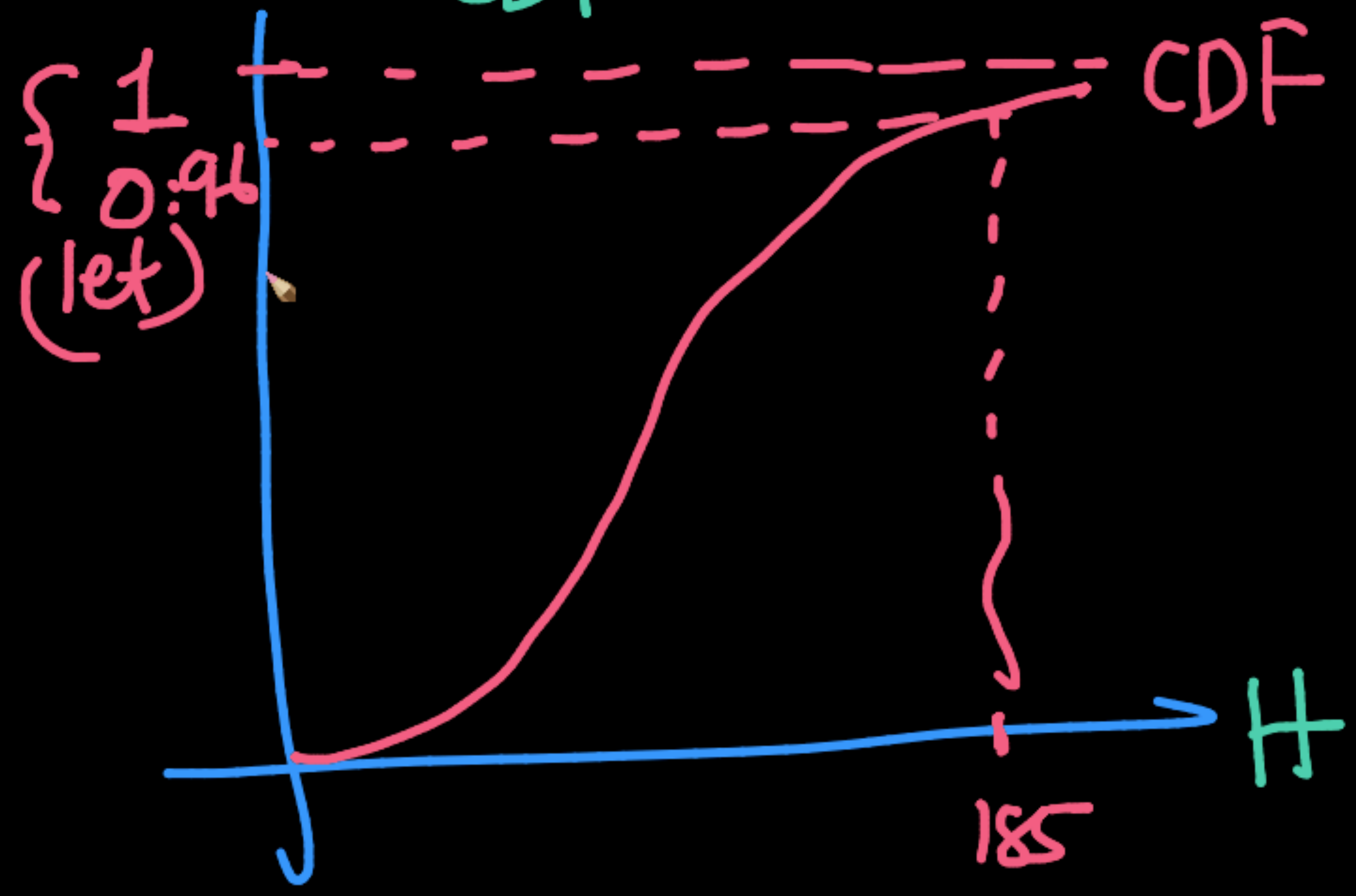
Tshirt sizes

XL: > 185cm

$$1 - P(H \le 185) = P(H > 185)$$

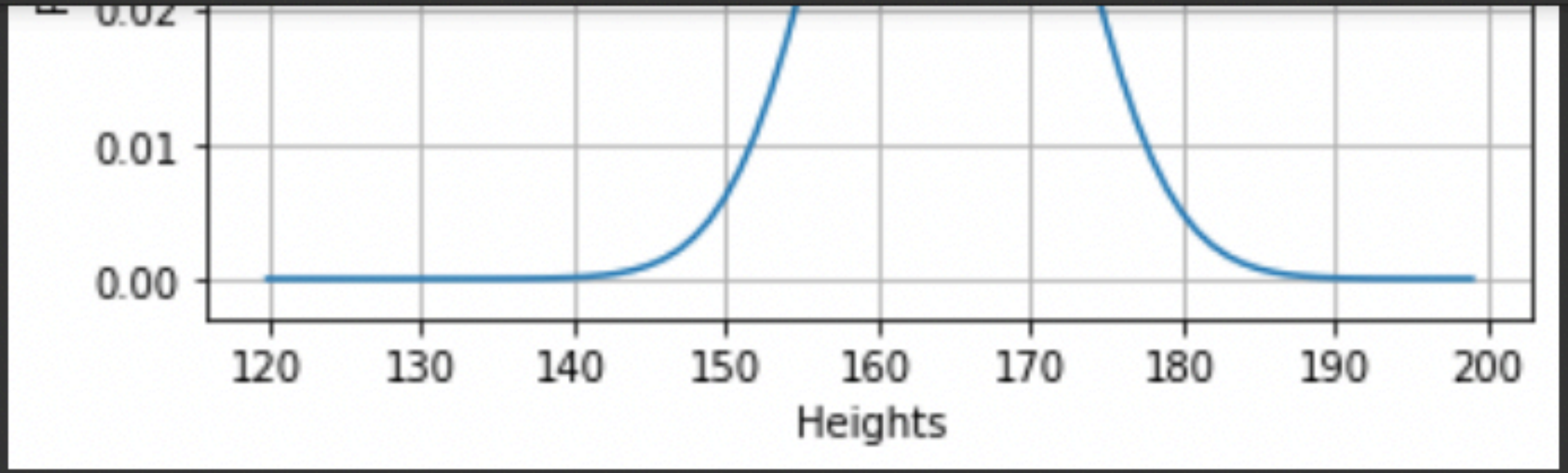
PDF $\longleftrightarrow$ CDF





$$\underline{\underline{0.96}} = P(H \le 185)$$

$$\sim 4\% = XL$$




```python
height_dist.cdf(x = 150)   # P(X<=150)
```

= 1.65% : XXS

```
0.016573163101179463
```

```python
height_dist.cdf(x = 200)    # P(X<=200)
```

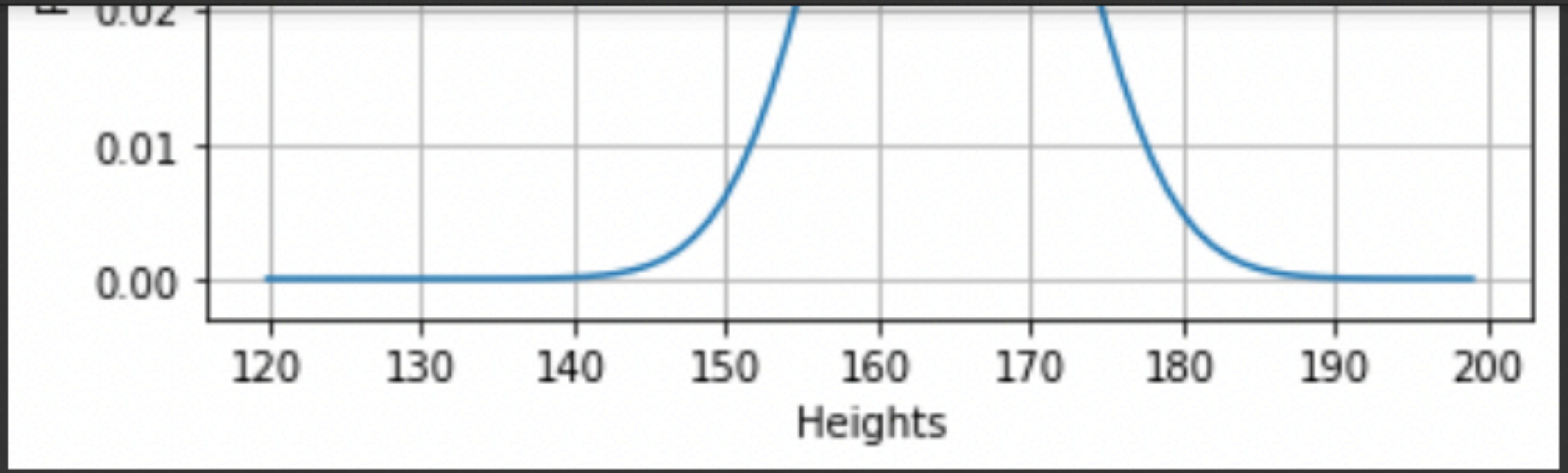
XXL *

```
0.9999998541520864
```

```python
below_150 = height_dist.cdf(x= 150)
below_160 = height_dist.cdf(x= 160)
```

```python
import math
math.ceil((below_160 - below_150)*10000)
```

```
2322
```




```
height_dist.cdf(x = 150)   # P(X<=150)
```

0.016573163101179463

```
height_dist.cdf(x = 200)    # P(X<=200)
```

0.9999998541520864

XS

150 — 160

```
below_150 = height_dist.cdf(x= 150)
below_160 = height_dist.cdf(x= 160)
```

```
import math
math.ceil((below_160 - below_150)*10000)
```

2322

fn

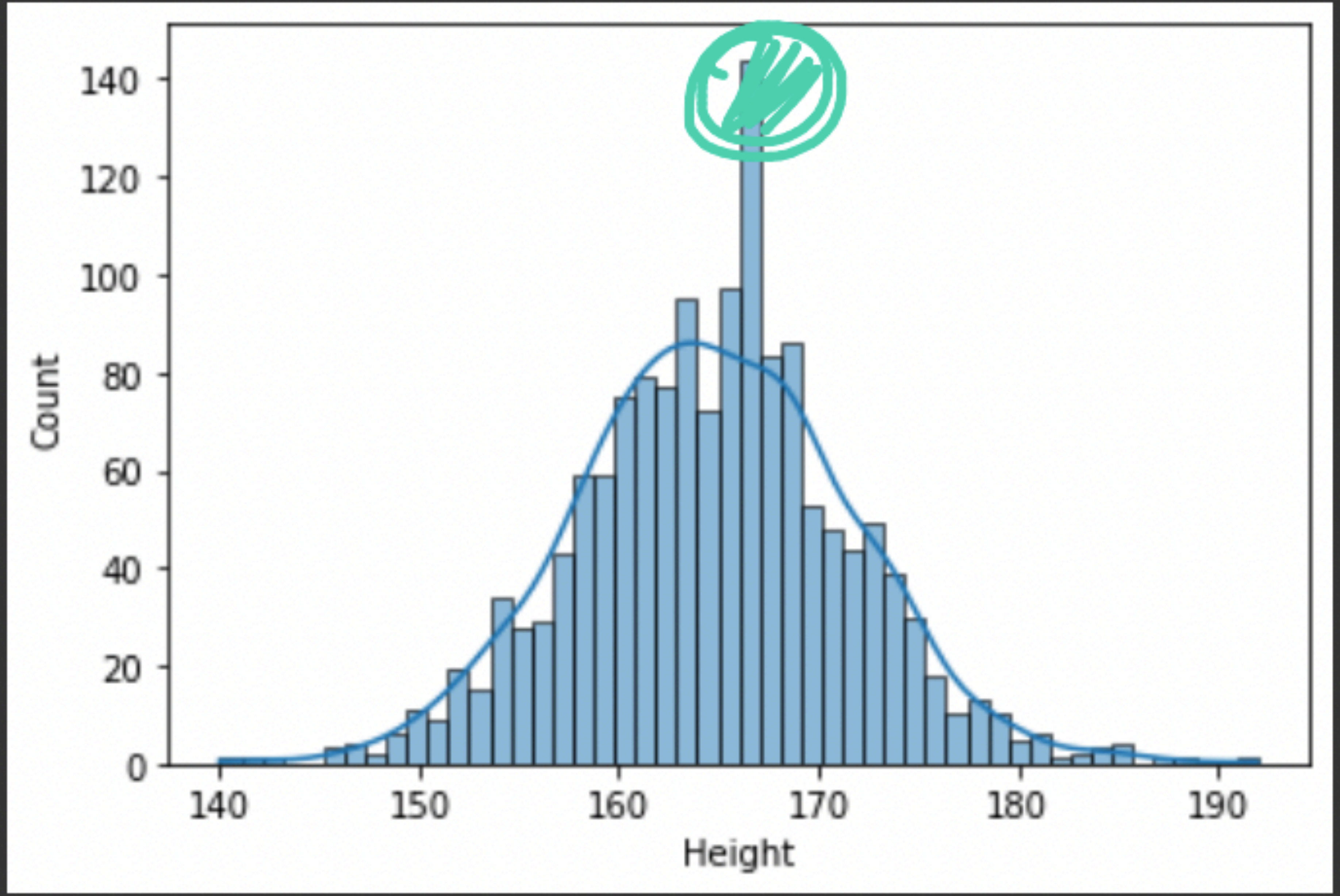
$X \sim \text{Normal}(\mu, \sigma)$

+ Code + Text

$\rightarrow N(\mu, \sigma)$

[10]
```
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>

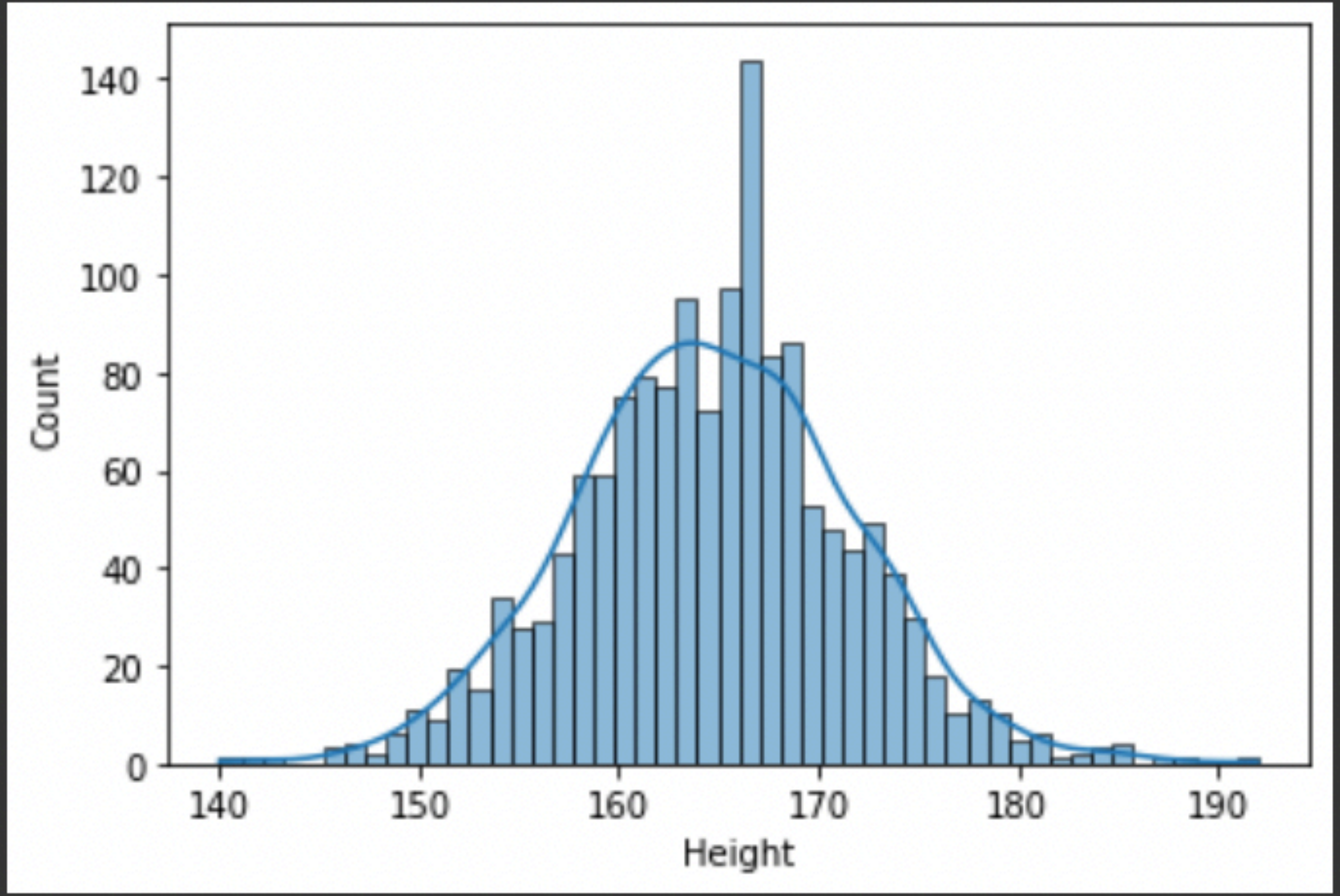


[13]
```
# lets get mean and std-dev from the data since we dont know population mean and std-dev
# ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev
employees['Height'].mean()
```

+ Code + Text

[10]
```
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>

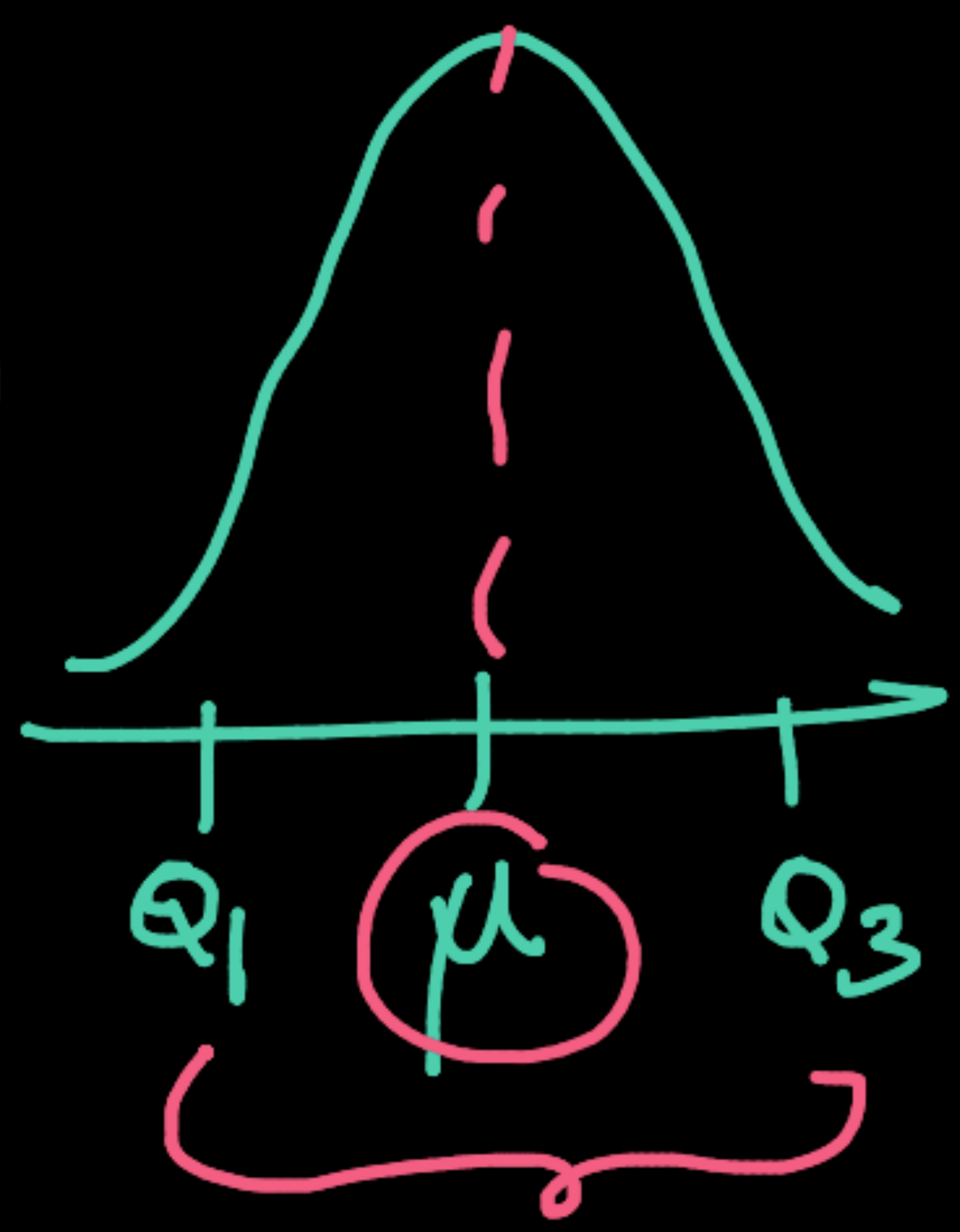


$H \sim N(\mu, \sigma)$

[13]
```
# lets get mean and std-dev from the data since we dont know population mean and std-dev
# ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev
employees['Height'].mean()
```

44 / 44

$\boxed{\text{Math}}$



$$\underbrace{Q_1 \qquad\qquad Q_3}_{50\%}$$

```
0.016573163101179463
```

```python
height_dist.cdf(x = 185)     # P(X<=180)
```

```
0.9984163806989597
```

```python
below_150 = height_dist.cdf(x= 150)
below_160 = height_dist.cdf(x= 160)
```

150 - 160 : XS

```python
import math
math.ceil((below_160 - below_150)*10000)
```

```
2322
```

$P(H \le 160) - P(H \le 150) \Rightarrow P(150 < H < 160) \times 10000$

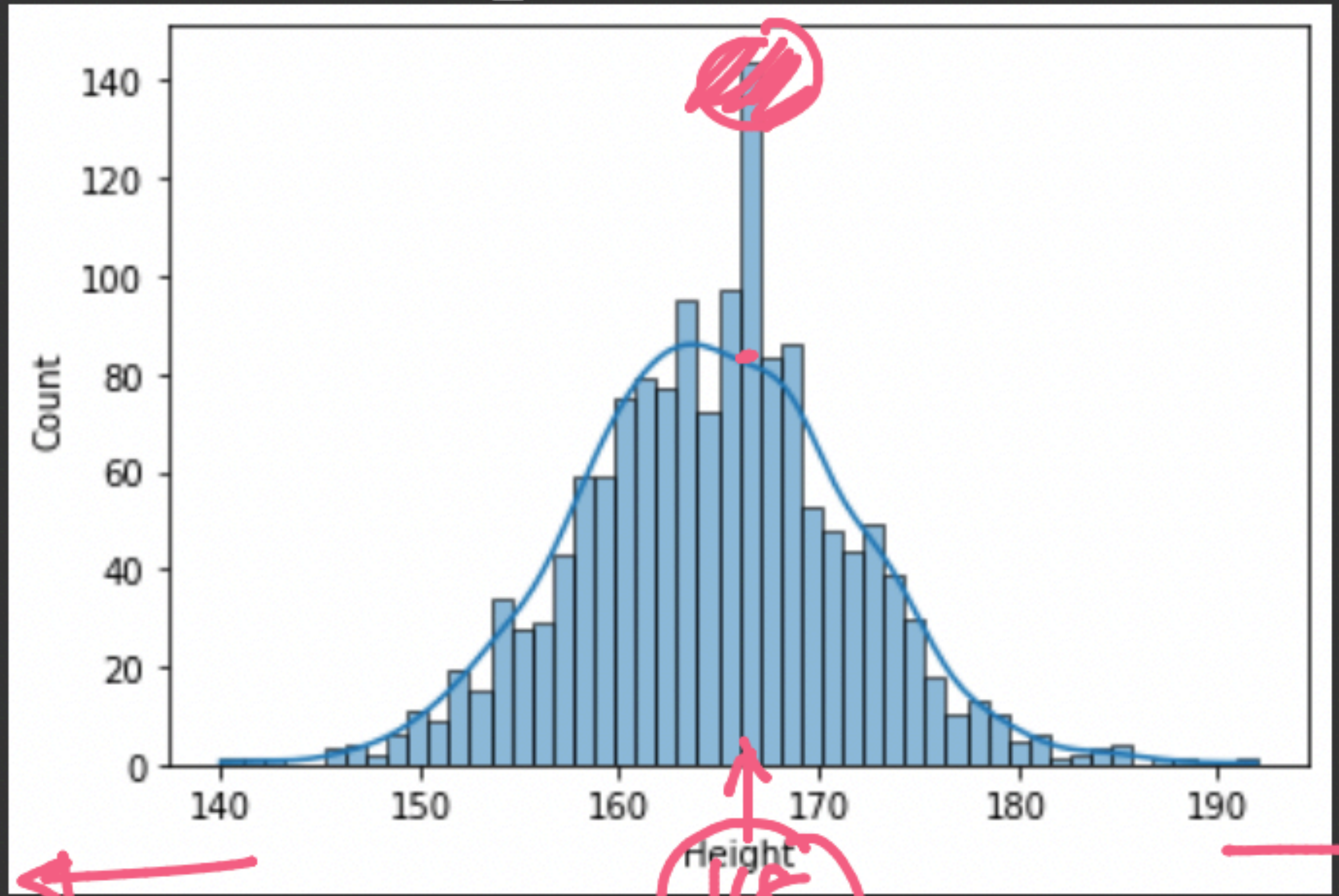
m, s $\to$ μ, σ approx...

Ans

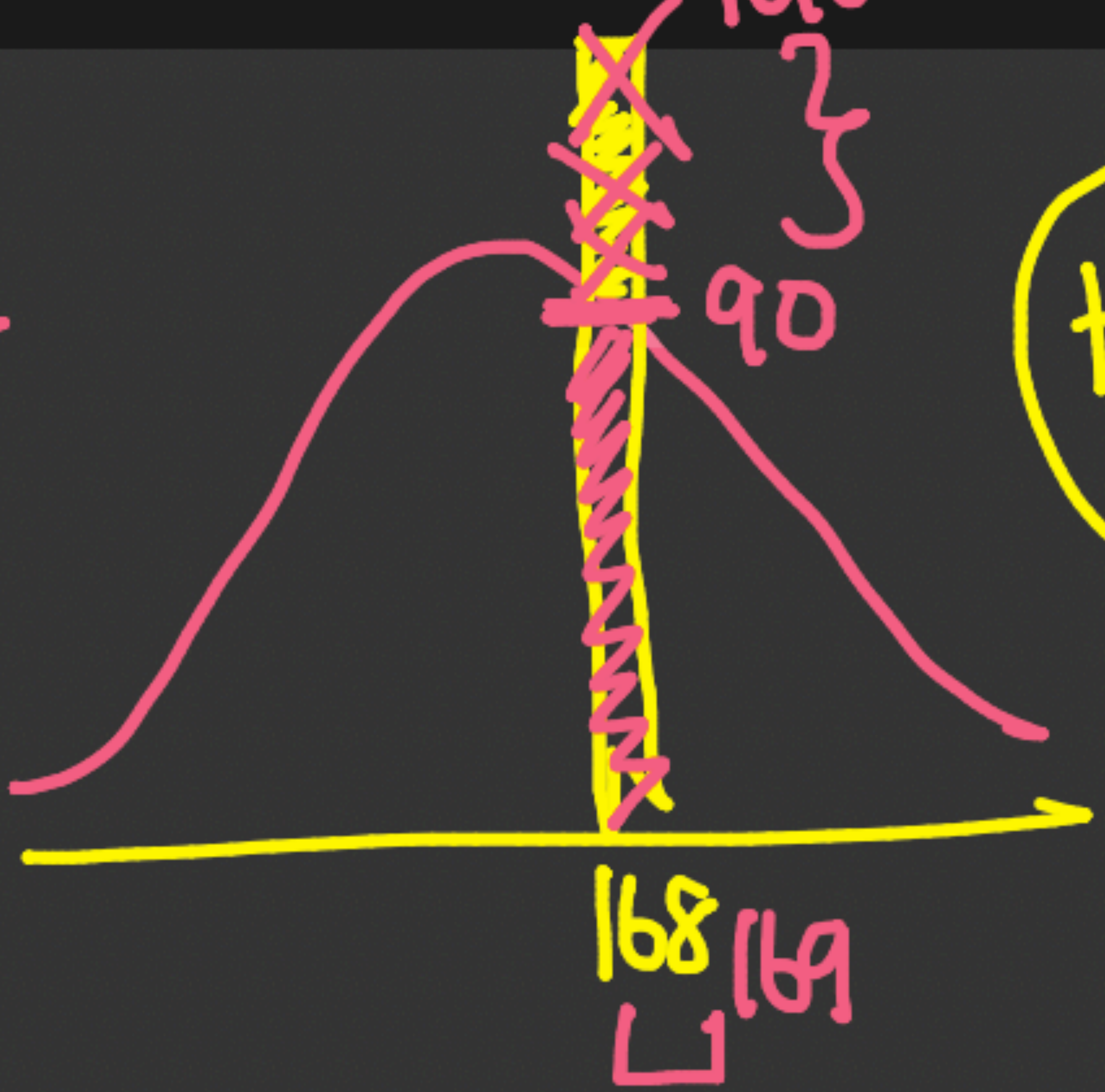
density-based outlier removal

```
[10] sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>



counts



H ~ Normally distb

```
[13] # lets get mean and std-dev from the data since we dont know population mean and std-dev
     # ASSUMPTION: sample mean and std-dev are good approximations of popoulation means and std-dev
     employees['Height'].mean()
```

Summarize:

Data → heights
(sample)

↓

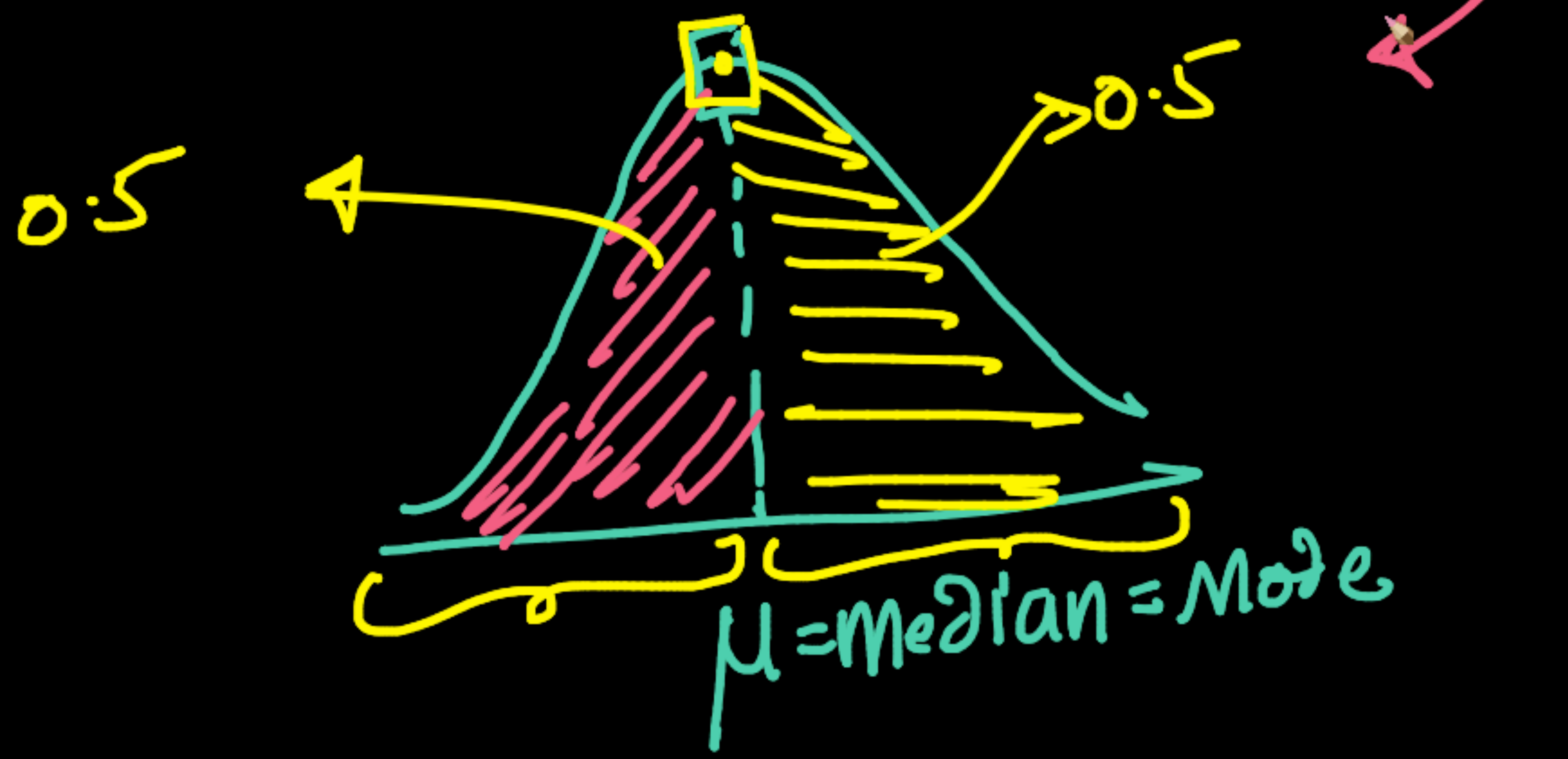
$H \sim N(\mu, \sigma)$ estimate

↓

CDF → # T-shirts
each size

⊛ outliers removed using density based approach → re-estimate μ & σ

Gaussian: mean = median = mode



Q

mean = Median = Mode

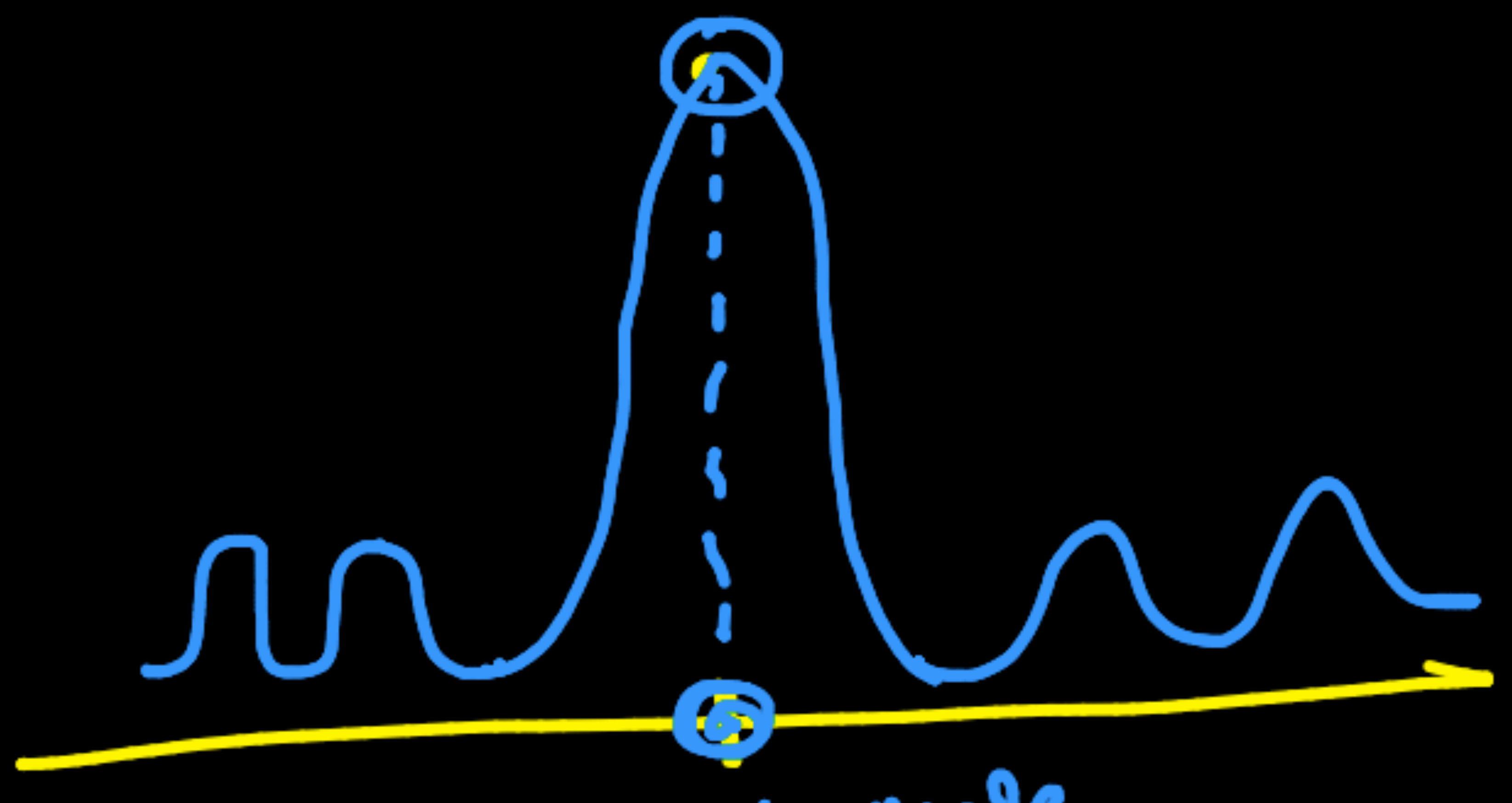
$\not\Rightarrow$ Gaussian disb ?

Stat tests

→ KS-tests

→ AD test

QQ-plot



μ=mode

box-cox

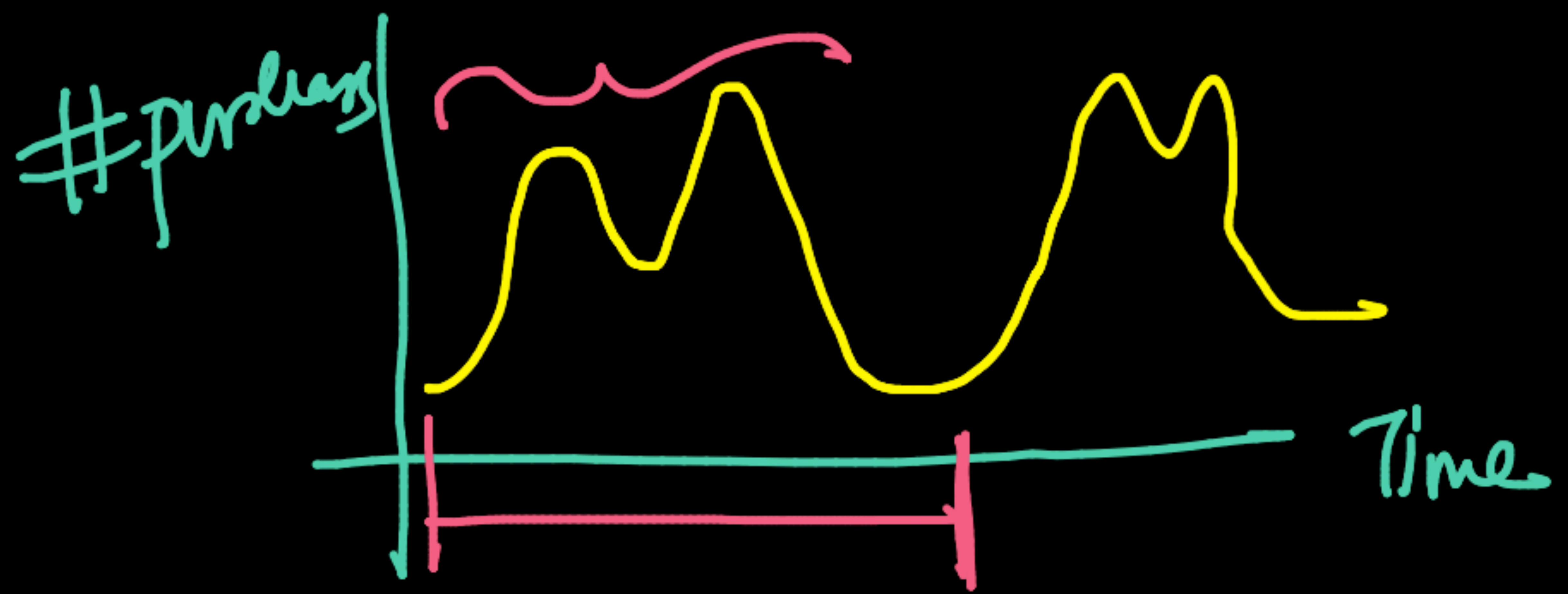
non-gaussian → transform

Gaussian

random

sample

population



(later) Maxima & minima

Max $f(x)$

PDF

Max @ μ

Symm

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}$$

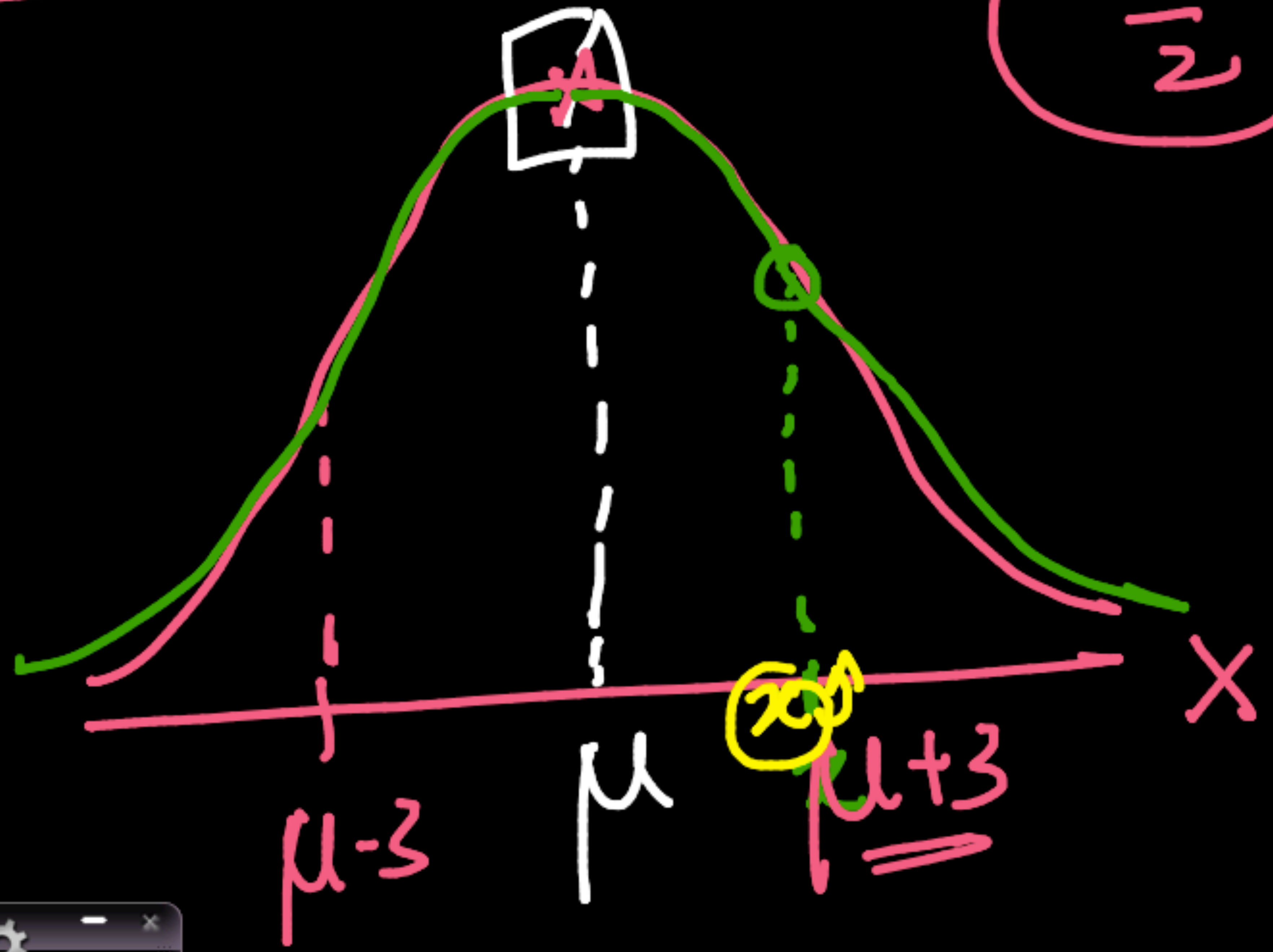
$-\frac{1}{2}\cdot\frac{3^3}{\sigma^2} =$

$-\frac{9}{2\sigma^2}$

$-\frac{1}{2}$

$x = \mu+3$

$x = \mu-3$



dataset

$$\mathrm{Var}: \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

dist

Normal-dist

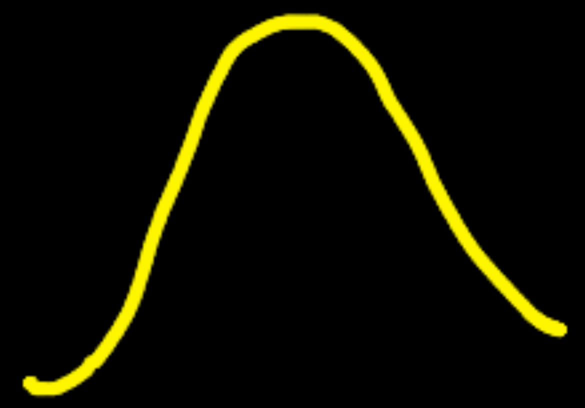
$f(x) =$

PDF

emperically

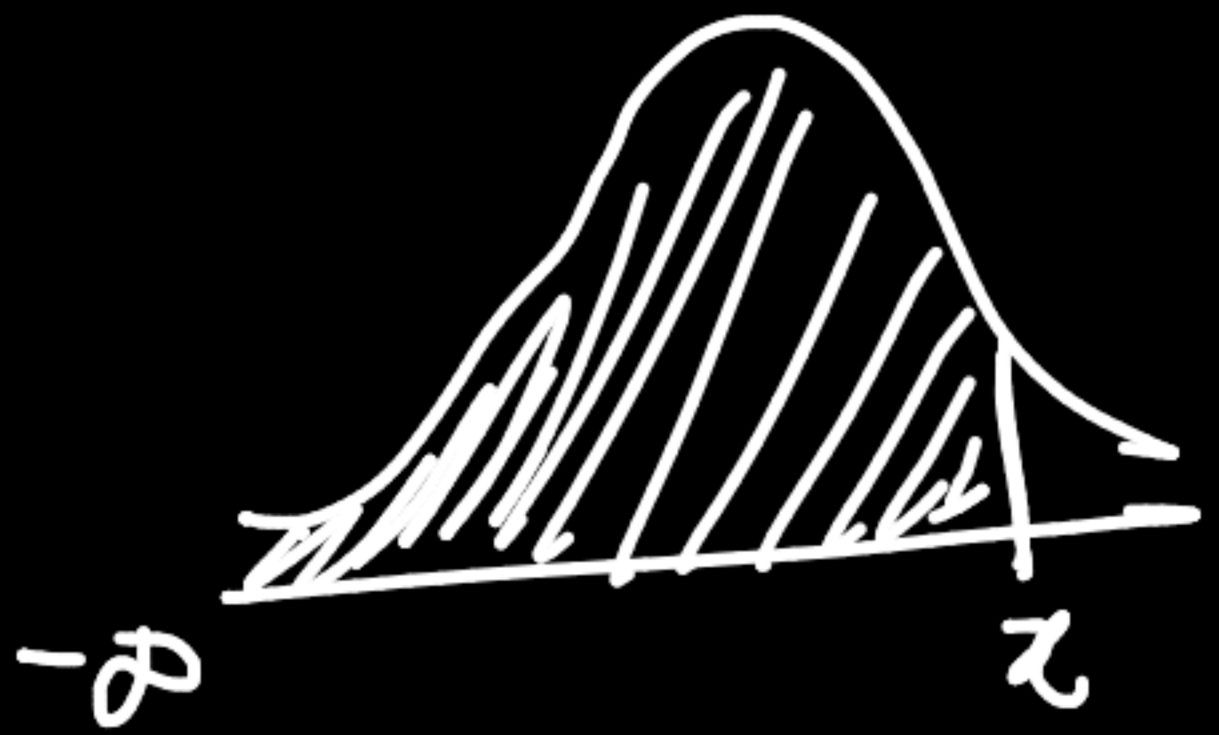
trial & error

$x_1, x_2, \dots x_n$



Rigorous
Math ...

$X \sim \text{Normal}(\mu, \sigma)$



later calculus

PDF: $f(x) =$

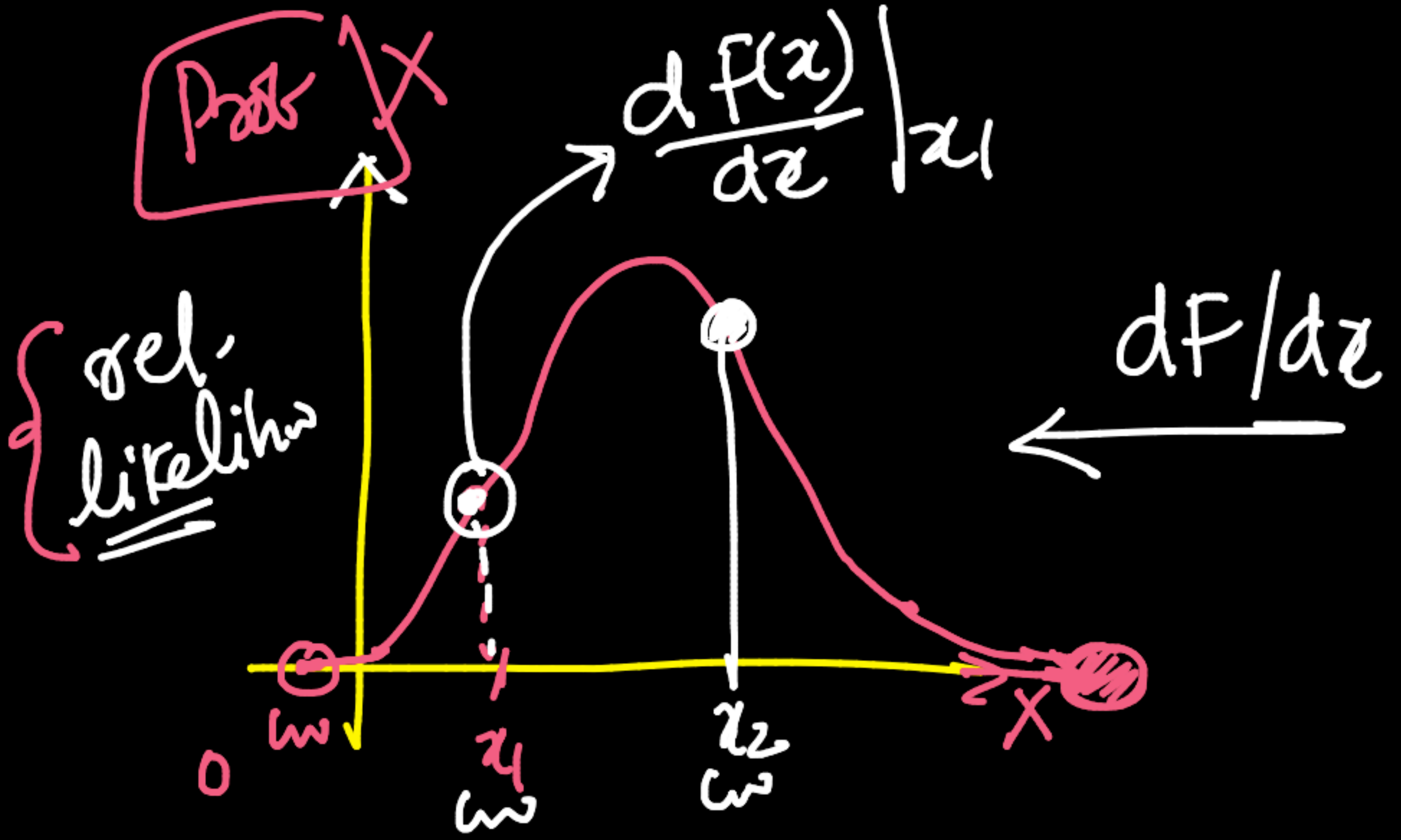
CDF: $F(x) = P(X \le x) = \int_{-\infty}^{x} f(x)\,dx$

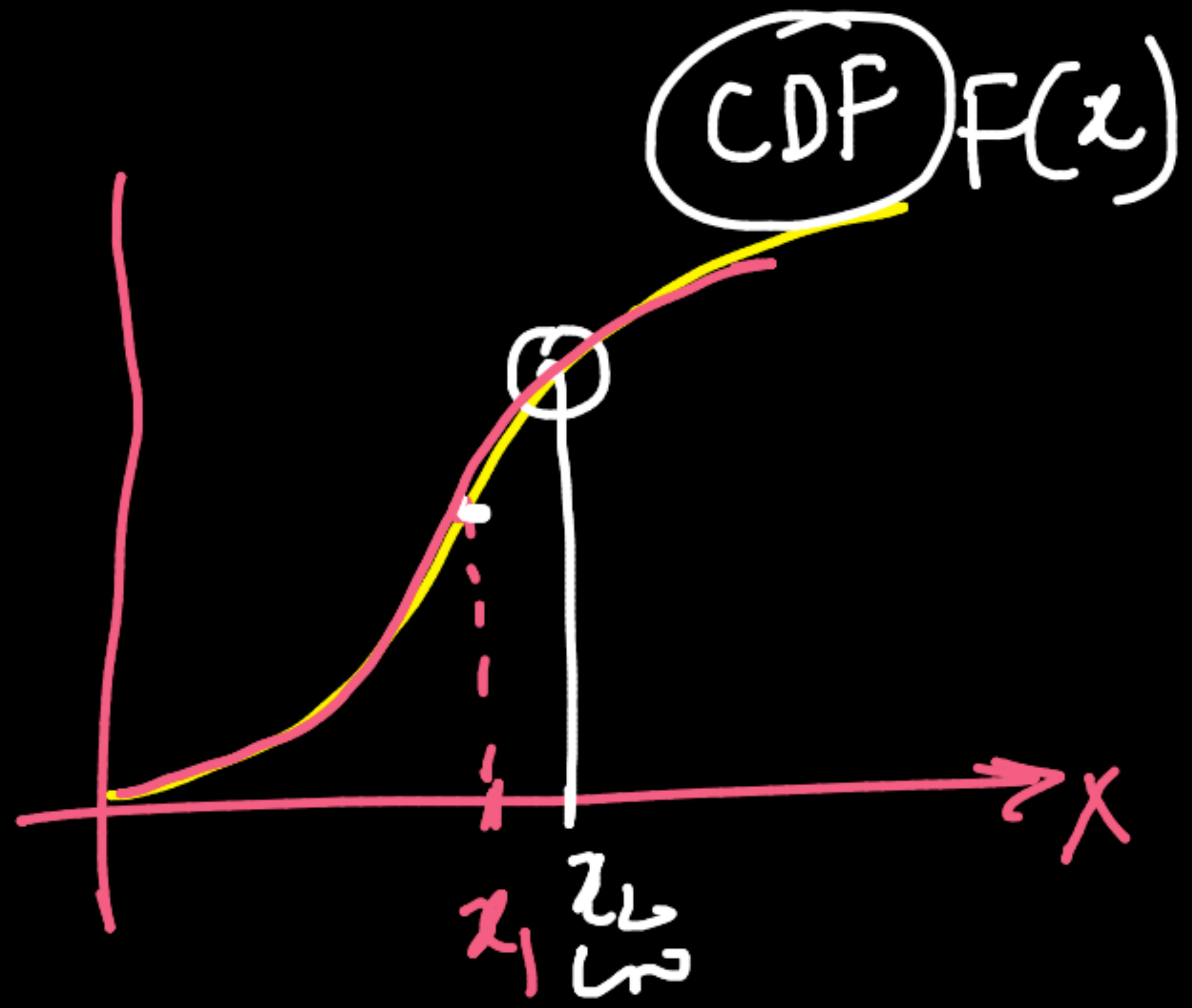
area

$f(x) = \frac{dF(x)}{dx}$: ∞-small difference

Continous r.V

$$\begin{tikzcd} \text{PDF} \arrow[r, "\int"] & \text{CDF} \arrow[l, bend left, "\frac{d}{dx}"] \end{tikzcd}$$





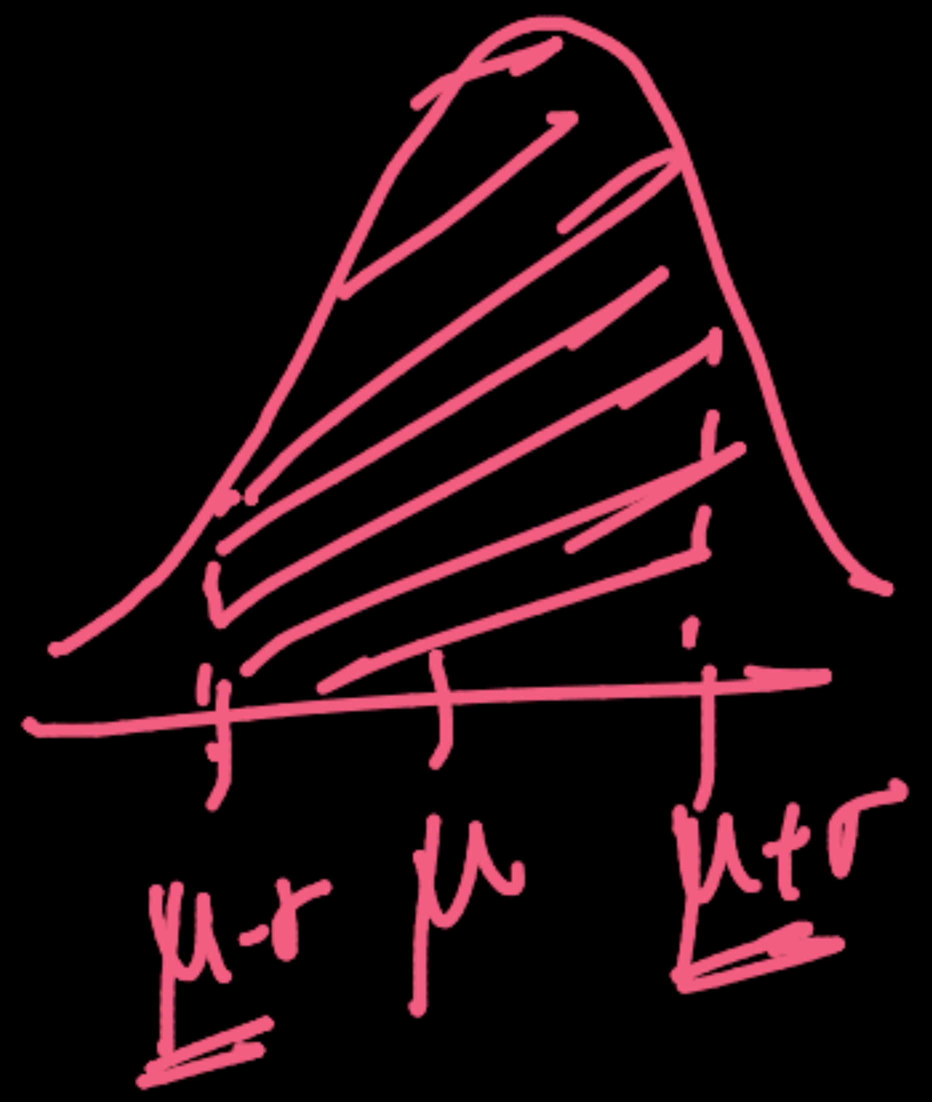
PDF;

$\boxed{x_1} = 0$

$x_2 = 0.1$

$x_3 = 0.2$

$X \sim \text{Normal}$



$[\mu-\sigma \, , \, \mu+\sigma] \rightarrow \sim 68\%$

$[\mu-2\sigma \, , \, \mu+2\sigma] \rightarrow \sim 95\%$

$[\mu-3\sigma \, , \, \mu+3\sigma] \rightarrow \sim 99\%$

68-95-99 rule

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Q

birth-weights

$\hookrightarrow N(\mu, \sigma)$

$\mu = 2.7$ kgs

$\sigma = 0.2$ kgs

what percentage of newborns have a wght [2.5 & 2.9 kgs]

$\mu - \sigma$ $\quad$ $\mu + \sigma$

$\sim 68\%$

Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:03

$\boxed{Q}$ $X = x_1, x_2, \ldots x_{1000}$ (μ, σ)

$\boxed{NO}$

interview

27%

$\sim 68\%$ $[\mu - \sigma \quad \mu + \sigma]$

$\sim 95\%$ $[\mu - 2\sigma \quad \mu + 2\sigma]$

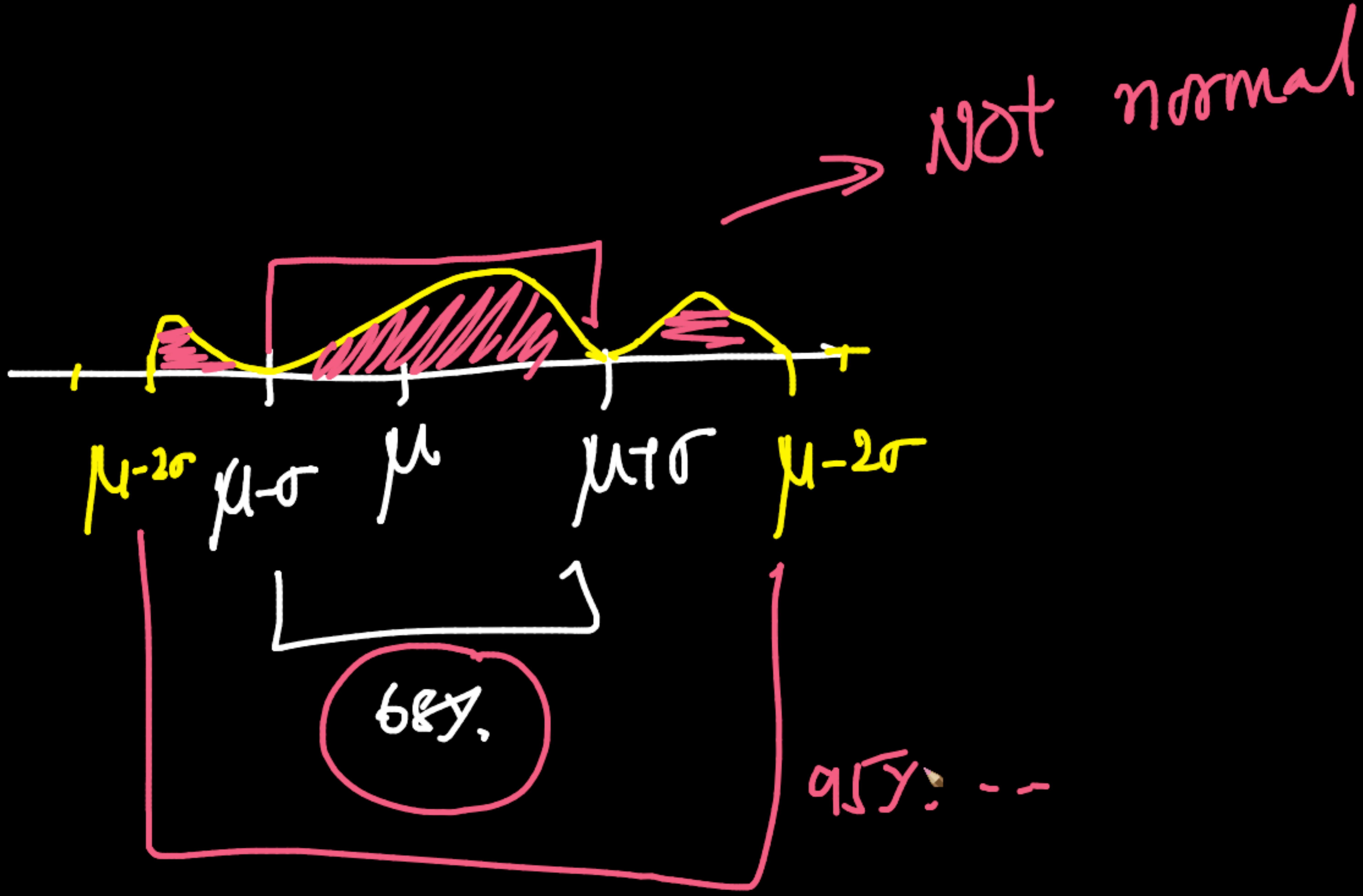
4%

$\sim 99\%$ $[\mu - 3\sigma \quad \mu + 3\sigma]$

Can we conclude $X \sim$ Normal ?

62 / 63

$$\frac{27}{2} = 13.5$$



Standard normal variate

Transformation

$\mu = 0$
$\sigma = 1$

$$X \sim N(\overset{100}{\mu}, \overset{5}{\sigma})$$

$x_1, x_2, \ldots x_i \ldots x_n$

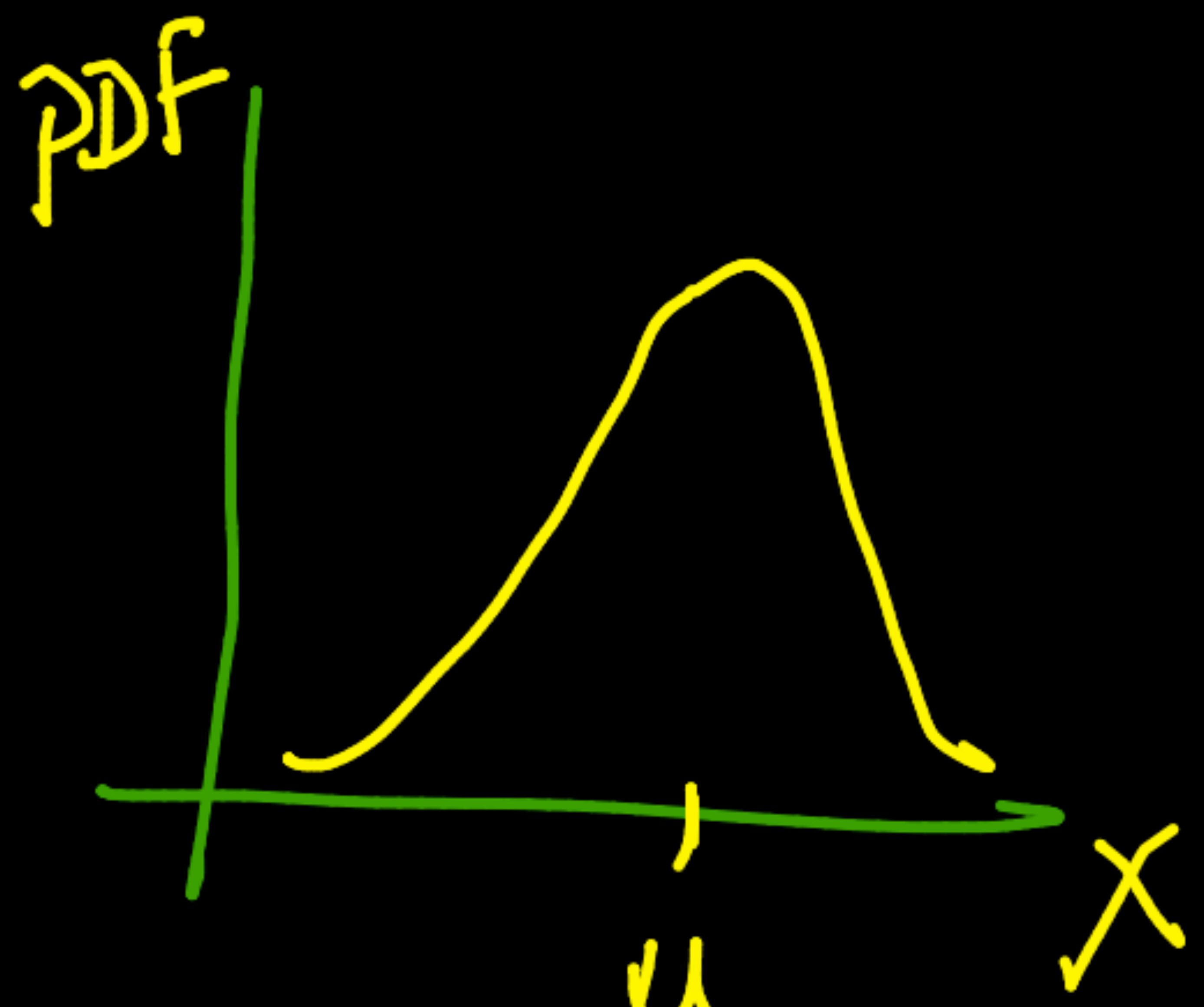
$$z_i = \frac{x_i - \mu}{\sigma}$$

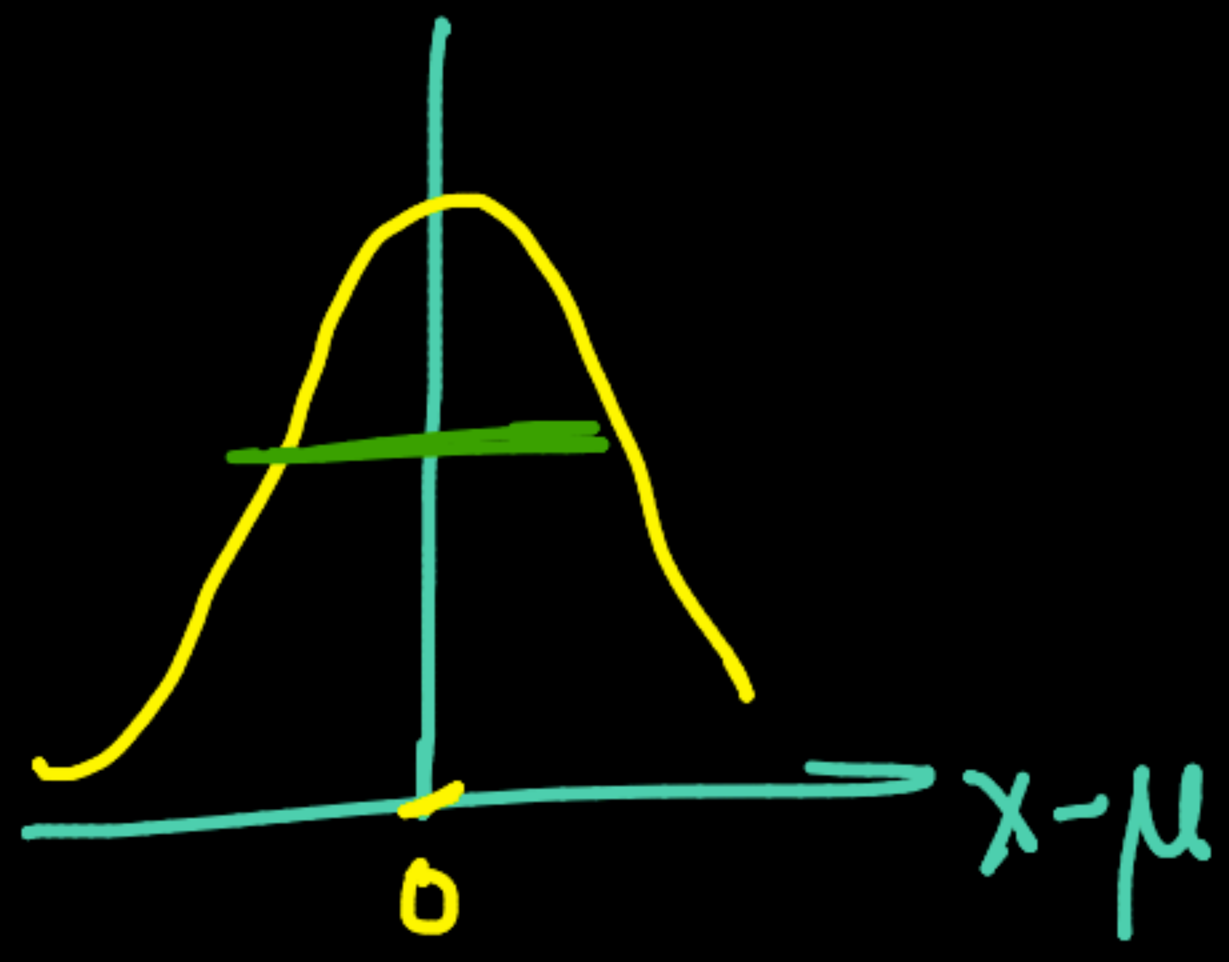
$z_1, z_2, \ldots z_n$ Z

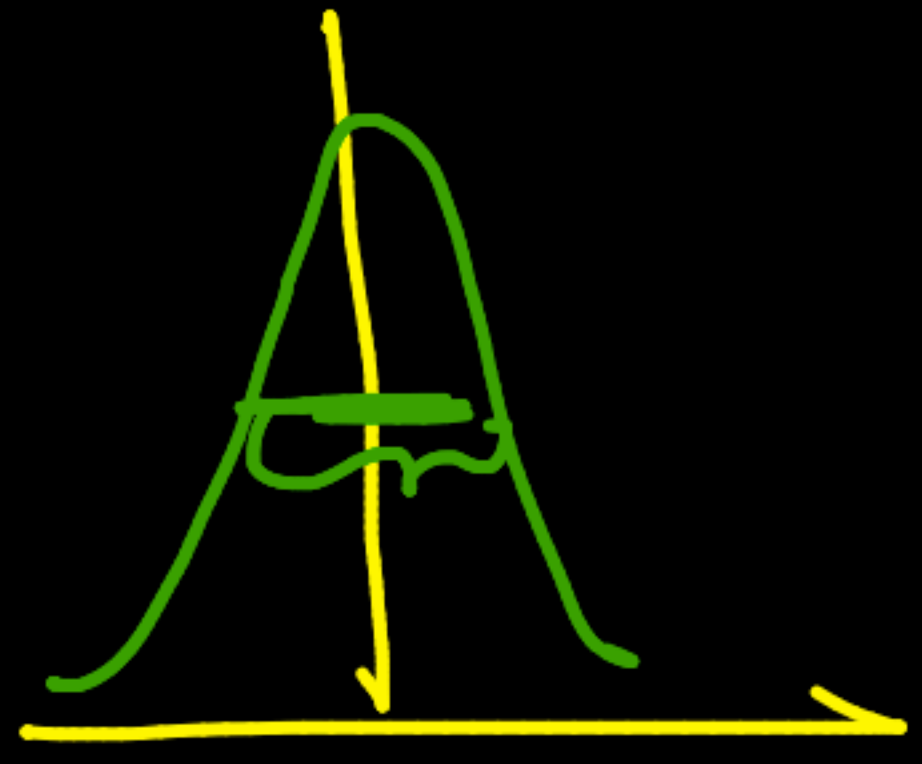
$\sim N(0, 1)$

SNV

$$Z = \frac{X - \mu_x}{\sigma_x}$$





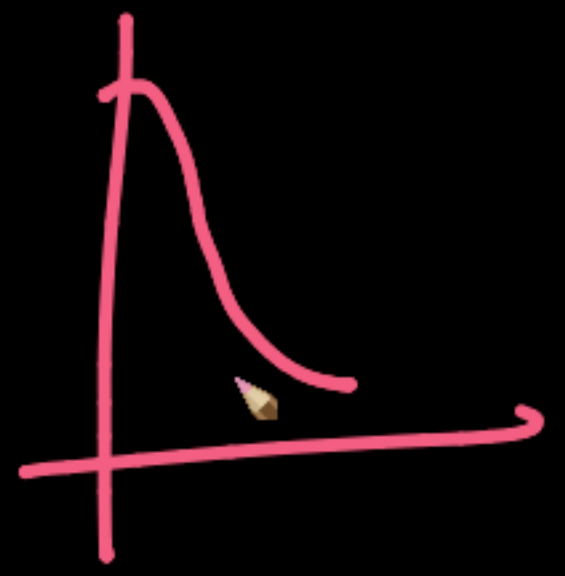


x_i

$x_i - \mu$

$\frac{x_i - \mu}{\sigma}$

$X \left| \frac{x_i - \mu}{\sigma} \right|^{abs}$



Chrome File Edit View History Bookmarks Profiles Tab Window Help

$$X \sim N(\mu, \sigma)$$

Prove!

$$z_i = \frac{x_i - \mu}{\sigma} \sim N(0,1)$$

Standardizati

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right) =$$

$$\frac{\frac{1}{n}\sum_{i=1}^{n} x_i - \mu}{\sigma} = \frac{\mu-\mu}{\sigma}$$

$$\boxed{=0}$$

66 / 66

Application:

(h_1, b_1)

$$H \sim N(\overset{\mu}{50cm} ; \overset{\sigma}{10cm}) \xrightarrow{\text{Standardize}} H' \sim N(0,1)$$

$\mu = 50 ; \sigma = 10$

$$B \sim N(\underset{\text{kgs}}{\mu = 2.7} ; \underset{\text{kgs}}{\sigma = 0.2}) \xrightarrow{\text{Standardize}} B' \sim N(0,1)$$

$\mu = 2.7$

$\sigma = 0.2$

no units

Baby: (h_1, w_1) $\xrightarrow[h \,\&\, w]{\text{standardized}}$

$H' \sim N(0,1)$
$W' \sim N(0,\boxed{1})$

units

h'	w'

Case 1: $(0, -1.6)$ → underweight baby

Case 2: $(-1.2, -1.8)$ → premature baby

h' w'

Case 3: (h', w') : $(1.5, 3)$

little overweight

99%

(h, w, hc)

1.5 1.3

-0.5

$N(0,1)$

$$(h', w') = (\underline{\underline{2.5}}, \underline{\underline{2.5}})$$

$X \nsim N(\mu, \sigma)$

Standardizati $\left\{ \dfrac{X-\mu}{\sigma} = Z \sim N(0,1) \right.$?

$\boxed{\text{NO}}$

50%

Poisson . . . .

Expo

Geometric

Log-normal

Gaussian : ~90% of Speed

$(h, w) = (75\ \text{cm}, 4\ \text{kgs})$

Standardization

H: μ, σ

W: μ, σ

$$Z = \frac{X - \text{mean}}{SD}$$

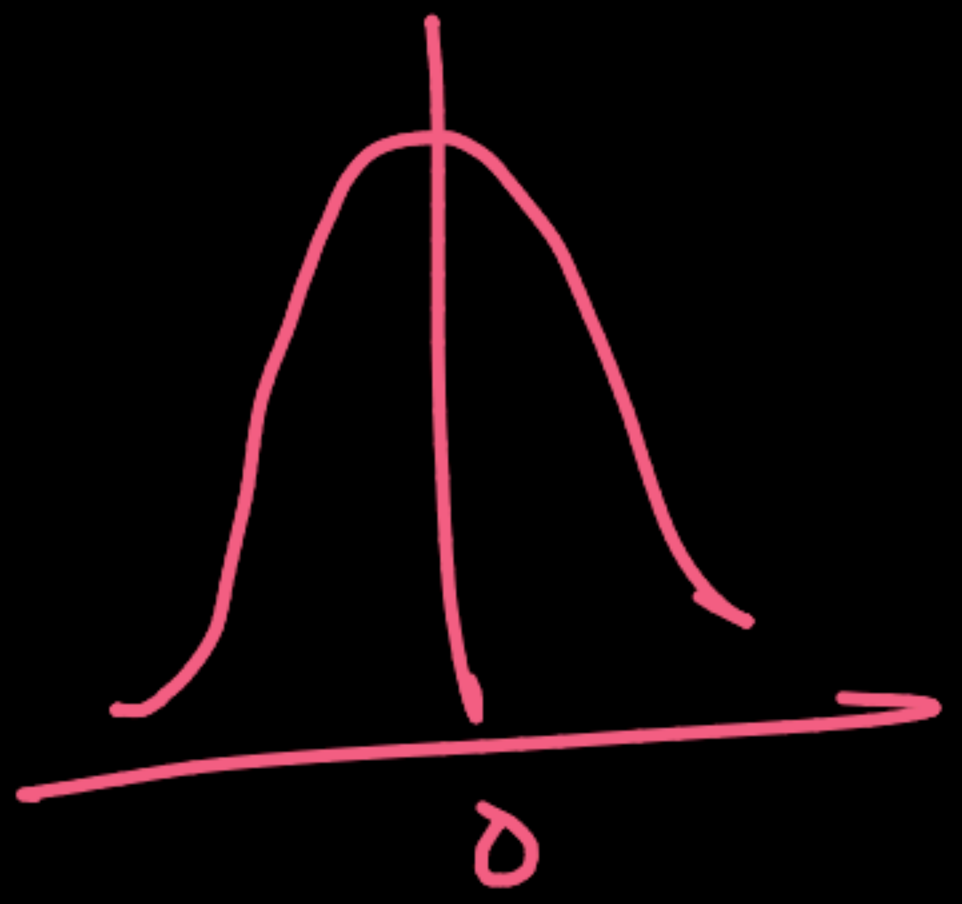
F-test: ANOVA

p-val: Hyp-testing

$\rightarrow$ last topics P&S

$$H \sim N(\overset{\mu}{50}\,\cancel{cm},\ \overset{\sigma}{10}\,\cancel{cm}) \longrightarrow N(0,1)$$

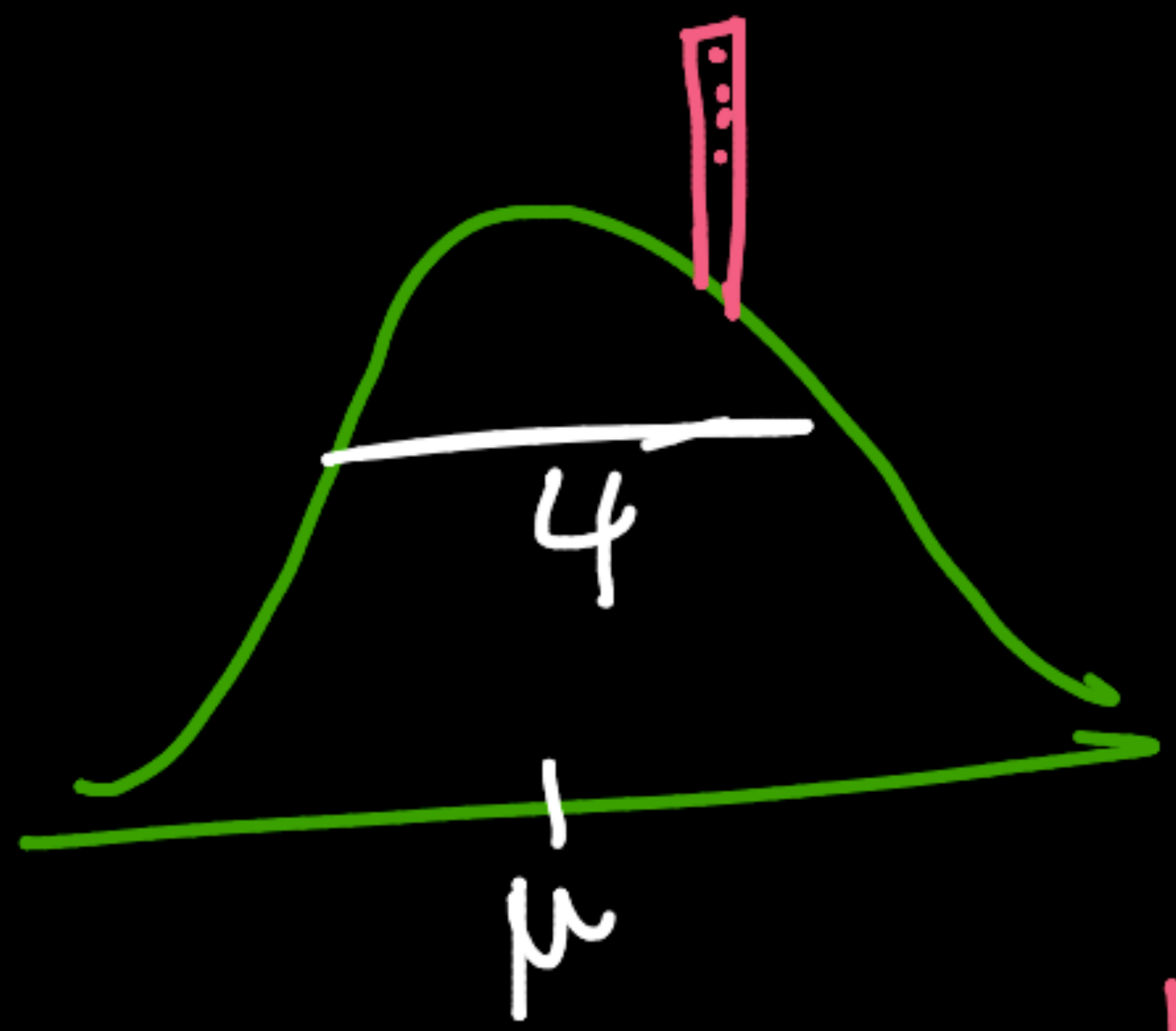
$$\sim N(\mu = 2\,\cancel{ft};\ \tfrac{1}{2}\,\cancel{ft})$$



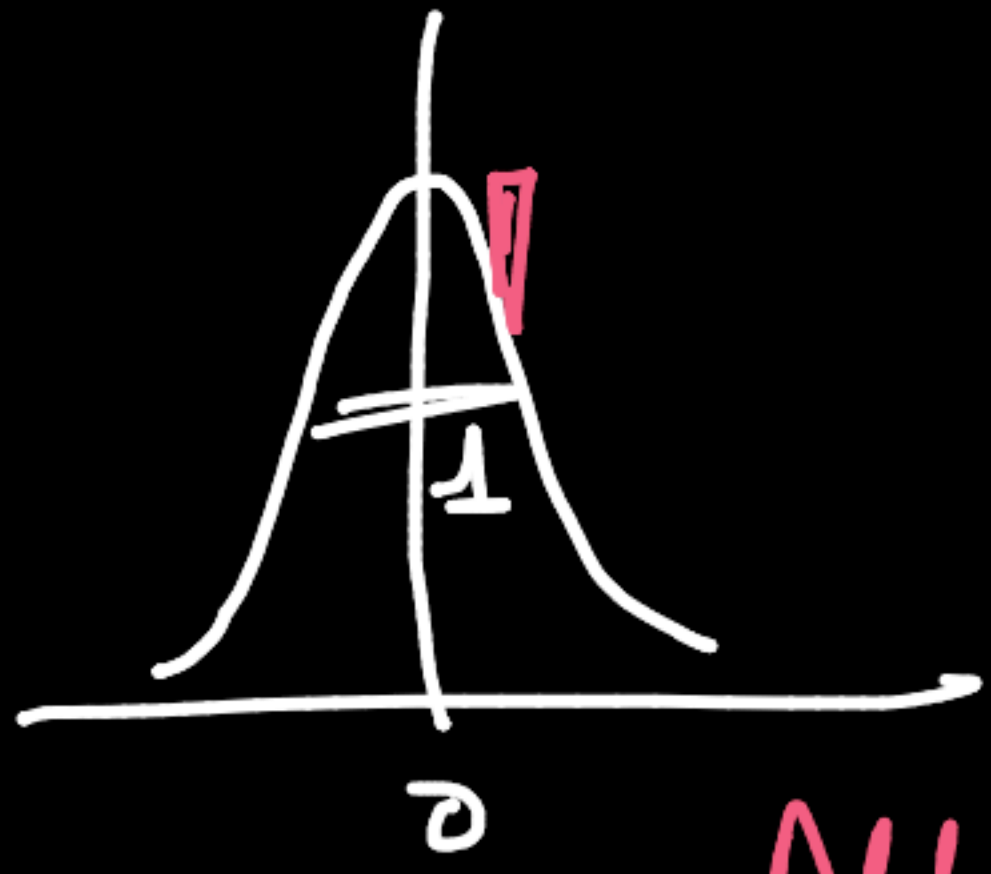
$$\frac{x_i - \mu}{\sigma}$$

→ mean-centering

variance scaling



not-removed



After std

faster $\rightarrow$ 14 : 9

1.2X

Speed $\rightarrow$ 20 : 4

Same-Speed

$\downarrow$

incr 1.1X

$\boxed{\text{scipy}}$ — vast

$\boxed{\text{needed}}$

$\boxed{\text{scipy.stats.norm}}$ ✓

Standardization $\leftrightarrow$ Normalization

$$\frac{x_i - \mu}{\sigma}$$

[mean-centring & Var-scaling]

widely used

x_i : x_{max} x_{min}

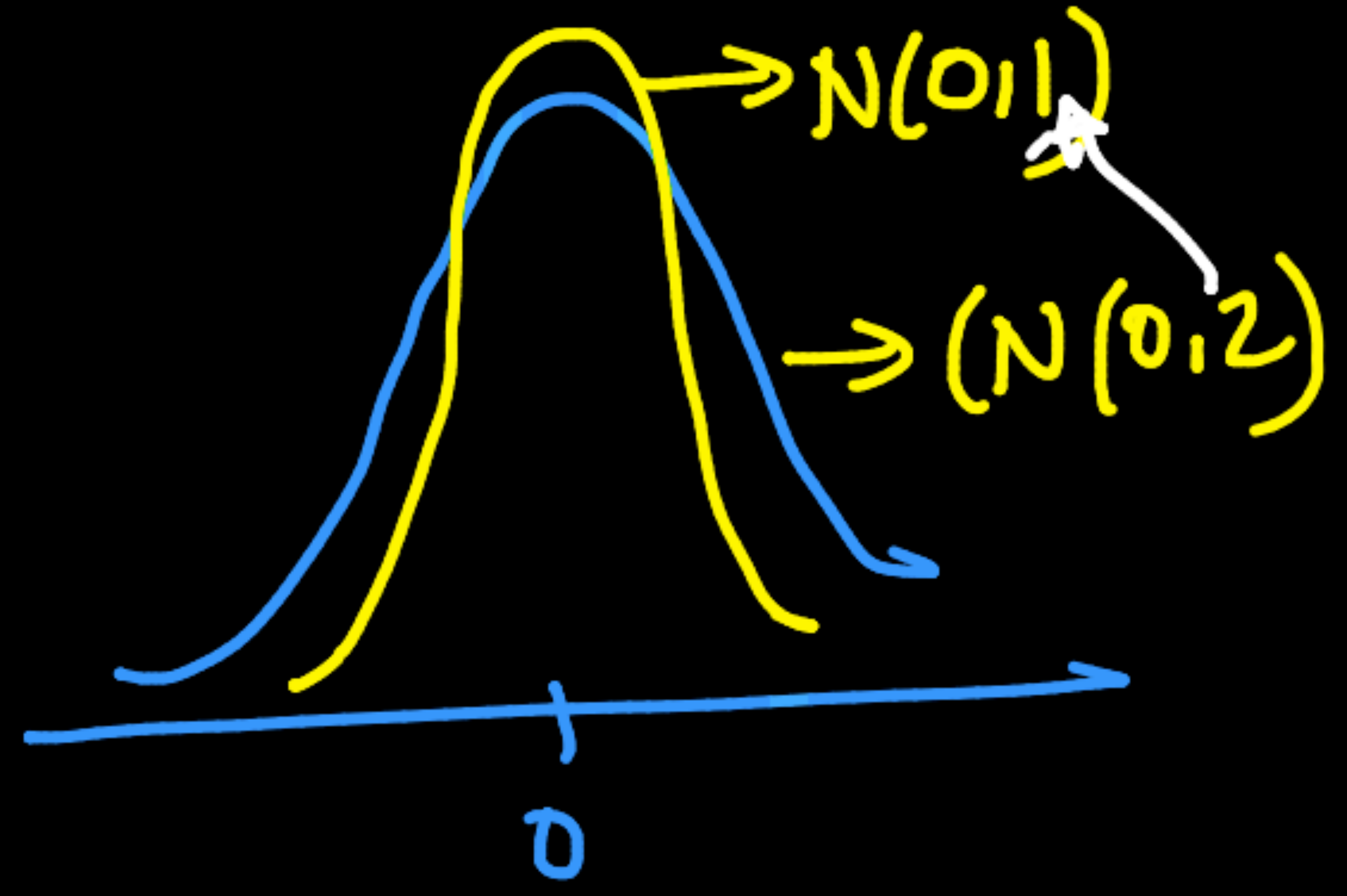
$$\frac{x_i - x_{min}}{x_{max} - x_{min}}$$

min-max scaling (rarely used)

0 —— 1

min max

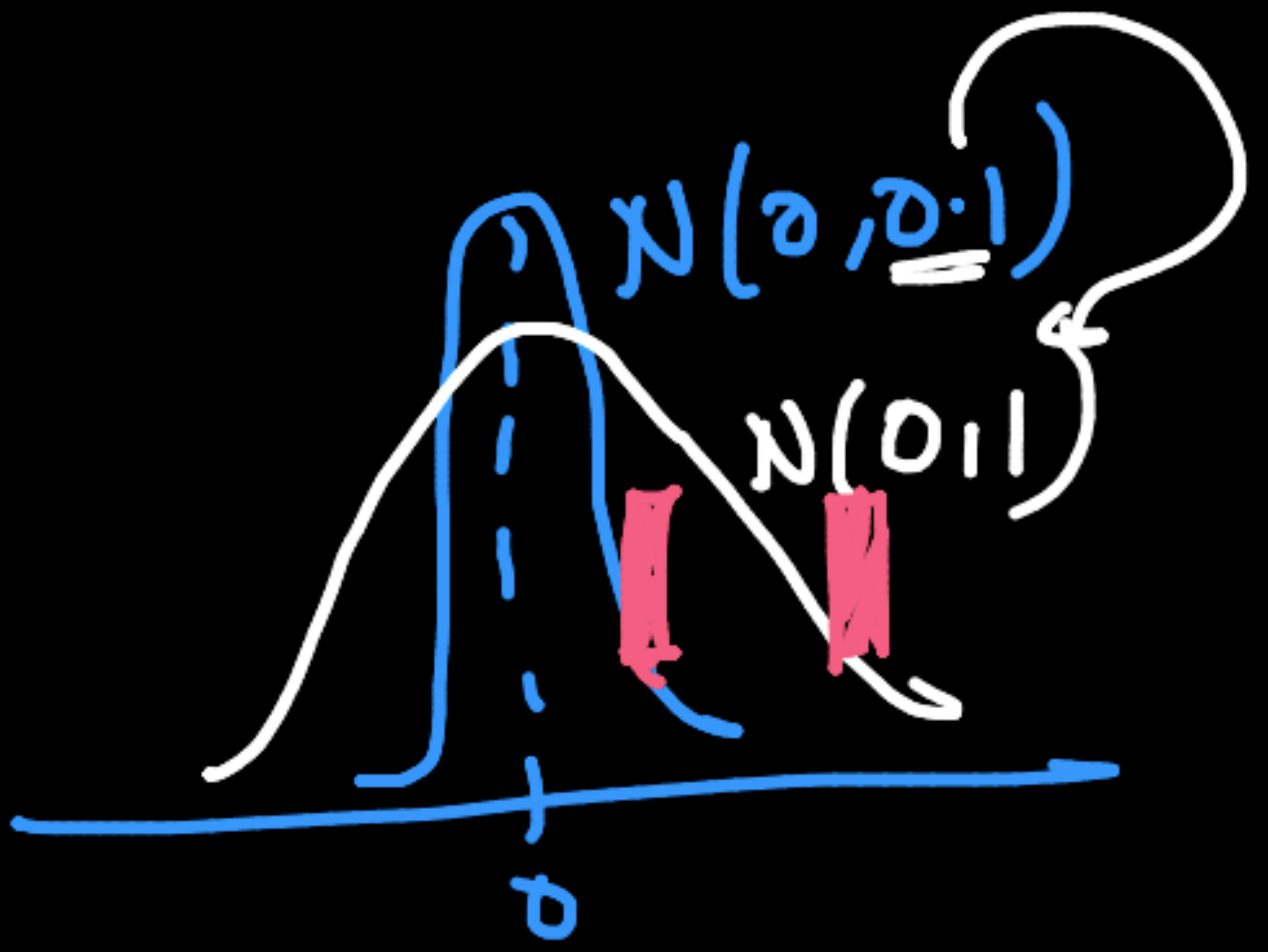
$N(0, 2)$



$X \sim N(0, 0.1)$

SD

$X' \sim N(0,1)$



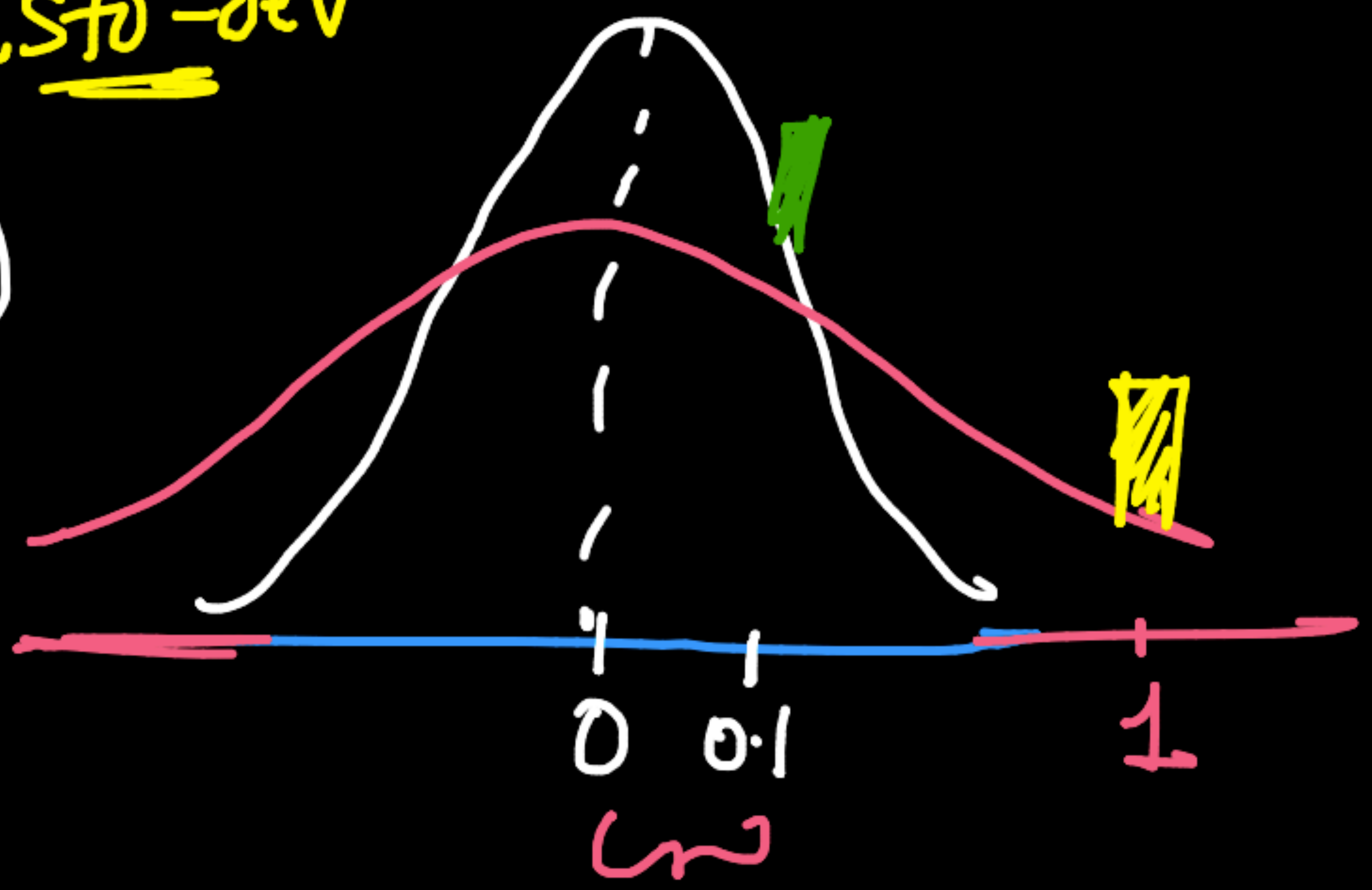
1 std-dev

$\cancel{X} \sim N(0, 0.1)$

std

$X' \sim N(0, 1)$

1 std-dev



→ case-study : $\boxed{\text{ask}}$ / confusions function

value-counts

a = [2, 0, 1, 0, 0, 0, 0, 0, 1]

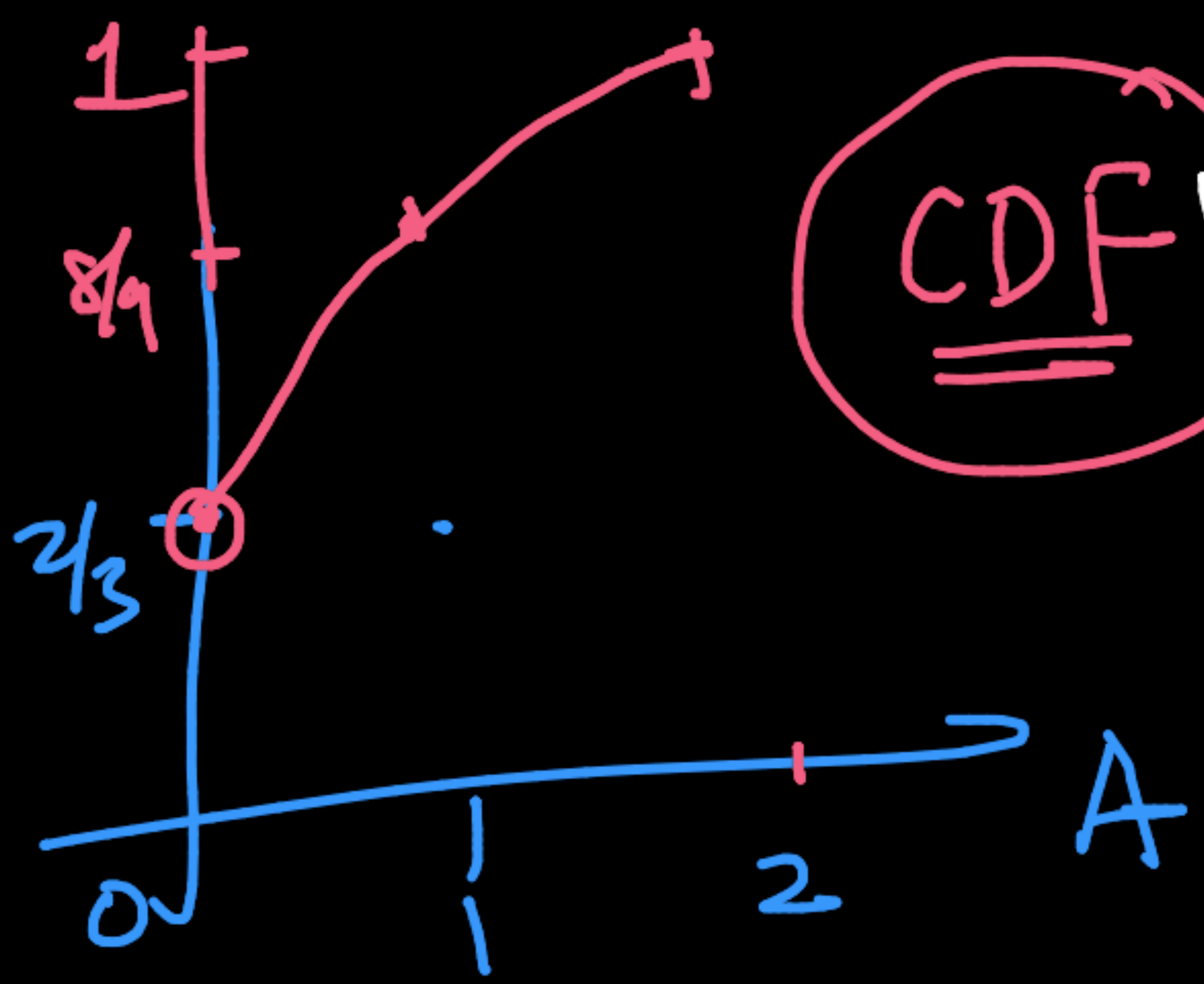
ndarray → scipy →

PMF

~~PDF~~ & CDF

$\frac{6}{9}$

0, 1, 2



CDF

coding + plot

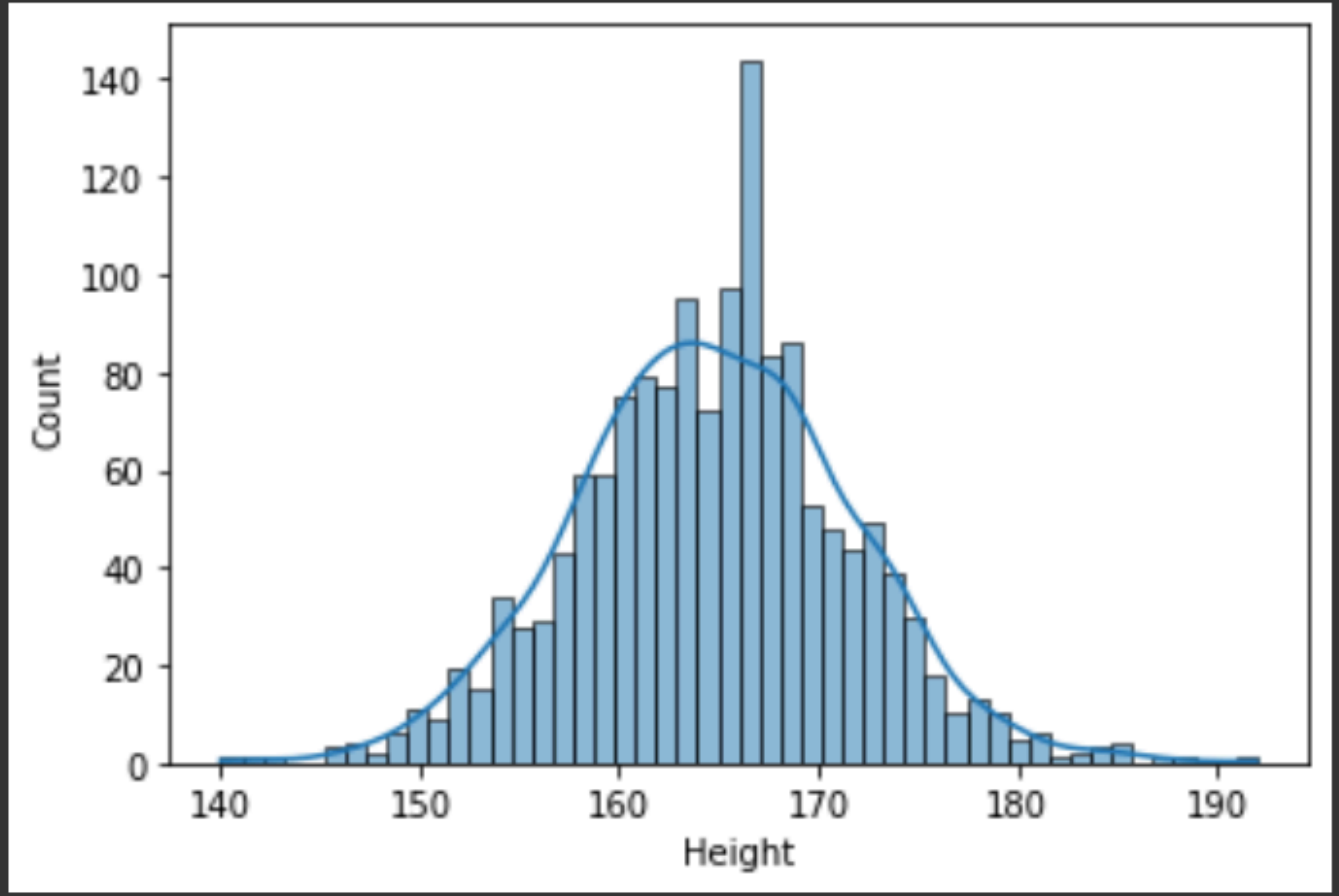
$a = [\, -\ -\ -\ \,]$ n-values

real

PDF & CDF : just using Matplotlib's line-plot

```
sns.histplot(employees['Height'], bins=50, kde=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8d560a2d0>




```
# lets get mean and std-dev from the data since we dont know population mean and std-dev

# ASSUMPTION: sample mean and ... on means and std-dev

employees['Height'].mean()
```

0s completed at 22:11