More

~~Advanced~~ Probability

Distributions

→ Study

DS/ML

Camera Settings

Prev:

✓ Random variable → discrete & continous

PMF ; PDF ; CDF

✓ Bernouli r.v (p)

→ Binomial r.V (n, p)

→ Overbooking of airline tickets ←

✓ → PMF & CDF

→ Normal dist

→ Poisson dist

→ Expo dist

→ log-normal

→ Geometric

+CLT

2 / 4

Ops:

@ topic → chat window [flow]

off-topic (Audio) → @ end of the session

$\boxed{\text{11:30 PM}}$ $\boxed{\text{Q\&A}}$ ✓

stop

```
[1] #PMF and CDF of Binomial r.v
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from scipy import stats
```

```
[2] showsup_probability = 0.9
```

```
# create 1000 points from a X ~ Binomial(n=110,p=0.9)
```

$\boxed{X} \sim \text{Binomial}(n=110, p=0.9)$

1000 flights

$\boxed{n=110;\ p=0.9}$

$\boxed{\text{disb}}$ → Concise way to represent a phenemenon

Model

```
[1] #PMF and CDF of Binomial r.v
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from scipy import stats
```

```
[2] showsup_probability = 0.9
```

```
# create 1000 points from a X ~ Binomial(n=110,p=0.9)
showsup_distribution =  stats.binom(n = 110, p= showsup_probability)
```

from scipy import stats

[1]

[2]
```
showsup_probability = 0.9
```

[3]
```
# create 1000 points from a X ~ Binomial(n=110,p=0.9)
showsup_distribution =  stats.binom(n = 110, p= showsup_probability)

showsup_data = showsup_distribution.rvs(1000)
```

X ~

[4]
```
showsup_data[:100]
```

```
array([100, 101, 101, 101, 103, 103,  91, 104,  95,  98, 101,  96,  95,
        99,  98, 101, 102,  95,  95, 102,  97,  96,  98, 101, 100, 101,
        98, 103, 102,  98,  94, 102, 102, 100,  94, 101, 102, 103, 100,
        97,  98, 105,  99,  96, 103,  93,  96,  98,  99, 101,  95, 103,
        98,  96,  94,  98, 101,  96, 102, 101,  98,  96,  93, 102,  98,
       101, 100, 102, 101, 106,  97, 103, 102, 103,  99, 101, 102, 100,
       101,  93,  98, 102,  96, 100, 100, 102,  90, 100, 103,  98,  98,
       101, 100, 102,  98, 102,  99,  93,  98, 103])
```

```python
# create 1000 points from a X ~ Binomial(n=110,p=0.9)
showsup_distribution =  stats.binom(n = 110, p= showsup_probability)

showsup_data = showsup_distribution.rvs(1000)
```

[5]
```python
type(showsup_distribution)
```

```
scipy.stats._distn_infrastructure.rv_frozen
```

1 penalty

1000 flights

110 tickets
p = 0.9

[4]
```python
showsup_data[:100]
```

```
array([100, 101, 101, 101, 103, 103,  91, 104,  95,  98, 101,  96,  95,
        99,  98, 101, 102,  95,  95, 102,  97,  96,  98, 101, 100, 101,
        98, 103, 102,  98,  94, 102, 102, 100,  94, 101, 102, 103, 100,
        97,  98, 105,  99,  96, 103,  93,  96,  98,  99, 101,  95, 103,
        98,  96,  94,  98, 101,  96, 102, 101,  98,  96,  93, 102,  98,
       101, 100, 102, 101, 106,  97, 103, 102, 103,  99, 101, 102, 100,
       101,  93,  98, 102,  96, 100, 100, 102,  90, 100, 103,  98,  98,
       101, 100, 102,  98, 102,  99,  93,  98, 103])
```

```python
# create 1000 points from a X ~ Binomial(n=110,p=0.9)
showsup_distribution =  stats.binom(n = 110, p= showsup_probability)

showsup_data = showsup_distribution.rvs(1000)
```

X : discrete r.V

PMF or PDF for X

[5]
```python
type(showsup_distribution)
```

```
scipy.stats._distn_infrastructure.rv_frozen
```

[4]
```python
showsup_data[:100]
```

```
array([100, 101, 101, 101, 103, 103,  91, 104,  95,  98, 101,  96,  95,
        99,  98, 101, 102,  95,  95, 102,  97,  96,  98, 101, 100, 101,
        98, 103, 102,  98,  94, 102, 102, 100,  94, 101, 102, 103, 100,
        97,  98, 105,  99,  96, 103,  93,  96,  98,  99, 101,  95, 103,
        98,  96,  94,  98, 101,  96, 102, 101,  98,  96,  93, 102,  98,
       101, 100, 102, 101, 106,  97, 103, 102, 103,  99, 101, 102, 100,
       101,  93,  98, 102,  96, 100, 100, 102,  90, 100, 103,  98,  98,
       101, 100, 102,  98, 102,  99,  93,  98, 103])
```

```
[5] type(showsup_distribution)
```

```
scipy.stats._distn_infrastructure.rv_frozen
```

```
[4] showsup_data[:100]
```

```
array([100, 101, 101, 101, 103, 103,  91, 104,  95,  98, 101,  96,  95,
        99,  98, 101, 102,  95,  95, 102,  97,  96,  98, 101, 100, 101,
        98, 103, 102,  98,  94, 102, 102, 100,  94, 101, 102, 103, 100,
        97,  98, 105,  99,  96, 103,  93,  96,  98,  99, 101,  95, 103,
        98,  96,  94,  98, 101,  96, 102, 101,  98,  96,  93, 102,  98,
       101, 100, 102, 101, 106,  97, 103, 102, 103,  99, 101, 102, 100,
       101,  93,  98, 102,  96, 100, 100, 102,  90, 100, 103,  98,  98,
       101, 100, 102,  98, 102,  99,  93,  98, 103])
```

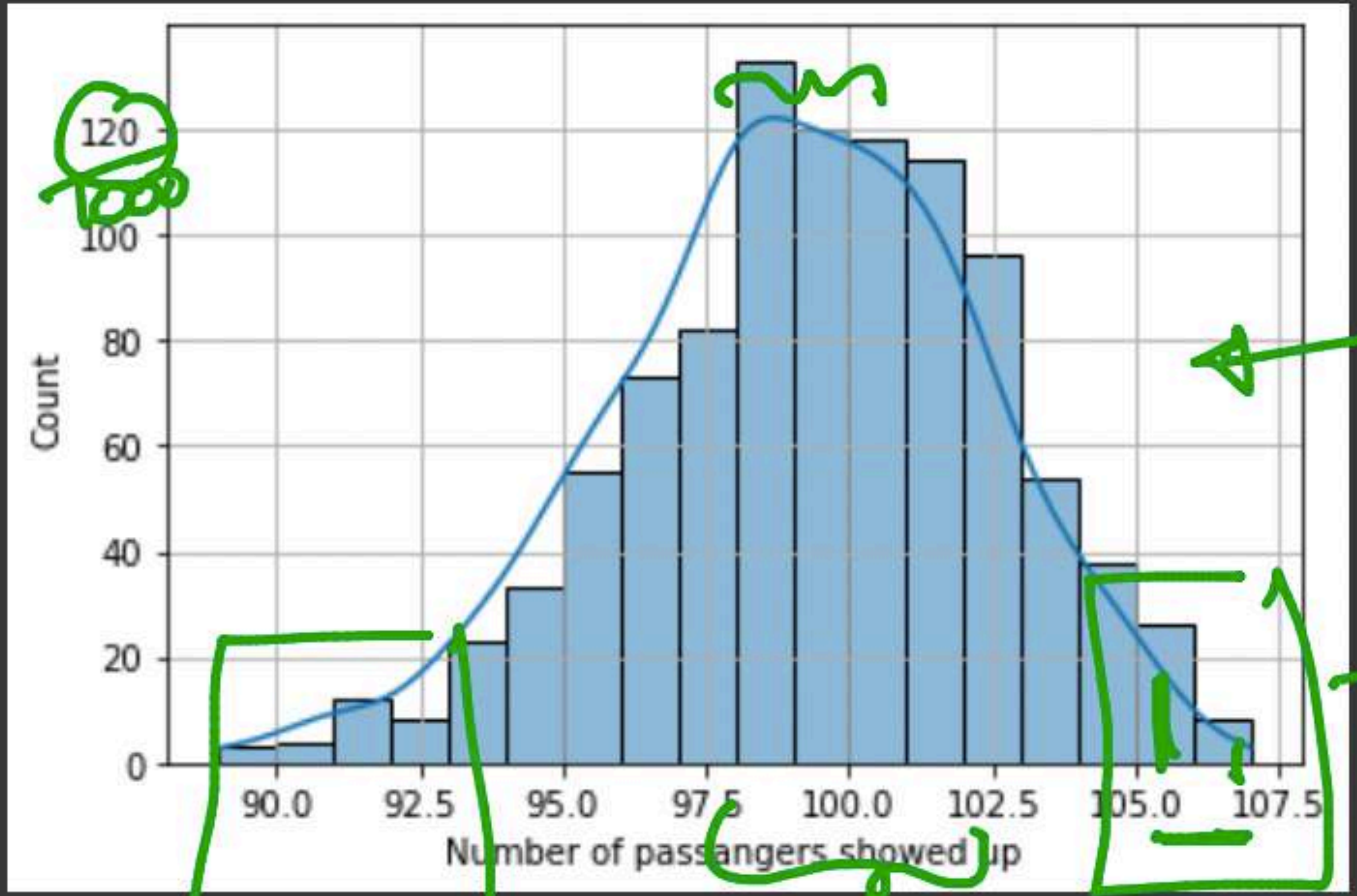
1000

```
#PMF of data
sns.histplot(showsup_data, bins=18, kde=True)
plt.xlabel("Number of passangers showed up")
plt.grid()
plt.show()
```

```python
#PMF of data
sns.histplot(showsup_data, bins=18, kde=True)
plt.xlabel("Number of passangers showed up")
plt.grid()
plt.show()
```

$X \sim Bin(n=110, p=0.9)$

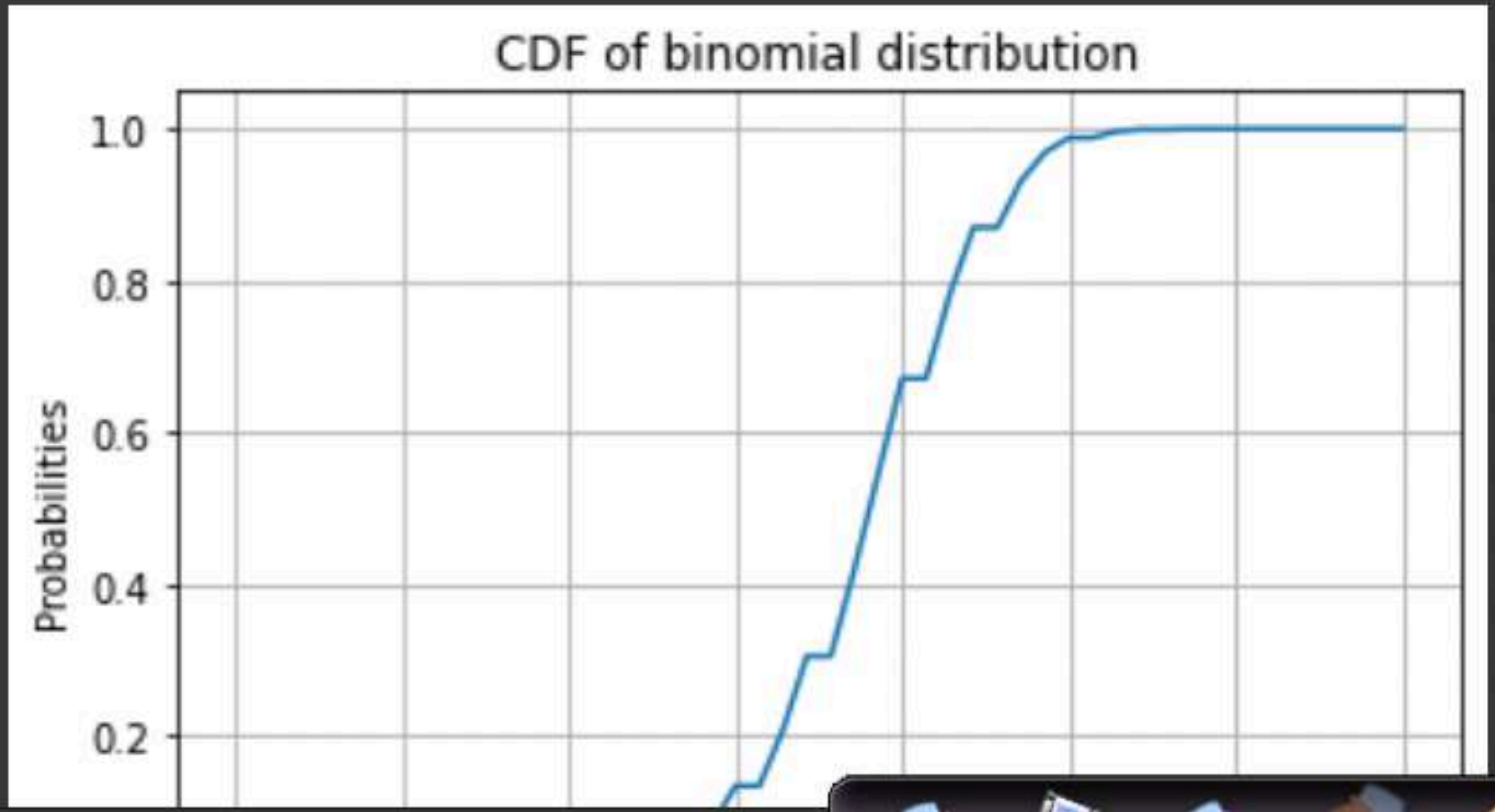
$x_1, x_2, \ldots \; x_{1000}$



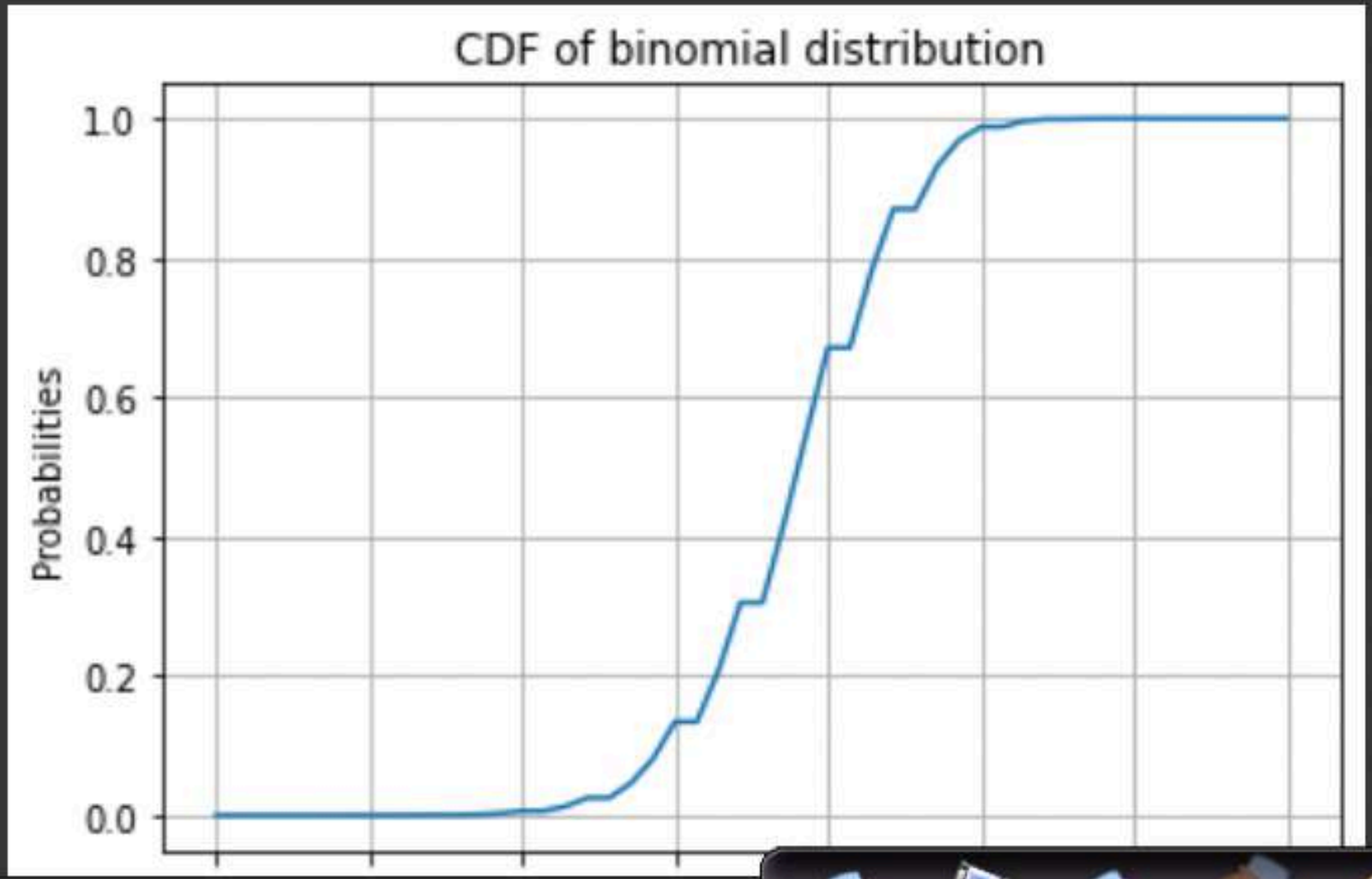
90.0 92.5 95.0 97.5 100.0 102.5 105.0 107.5
Number of passangers showed up

```
#CDF:
x = np.linspace(80, 115)

cdf = showsup_distribution.cdf(x)

plt.plot(x, cdf)
plt.xlabel("X-values")
plt.ylabel("Probabilities")
plt.title("CDF of binomial distribution")
plt.grid()
plt.show()
```

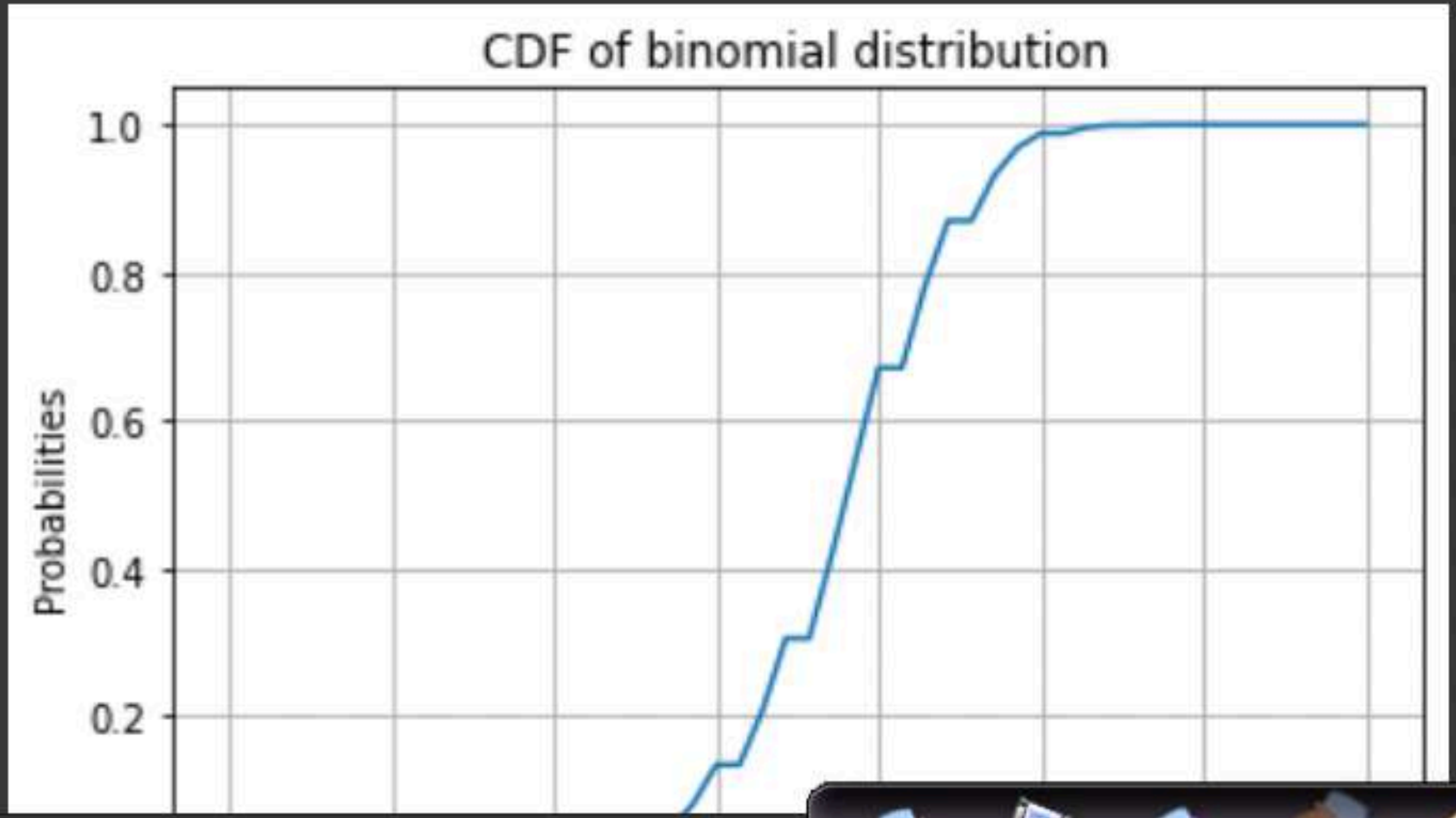
stats module in scipy



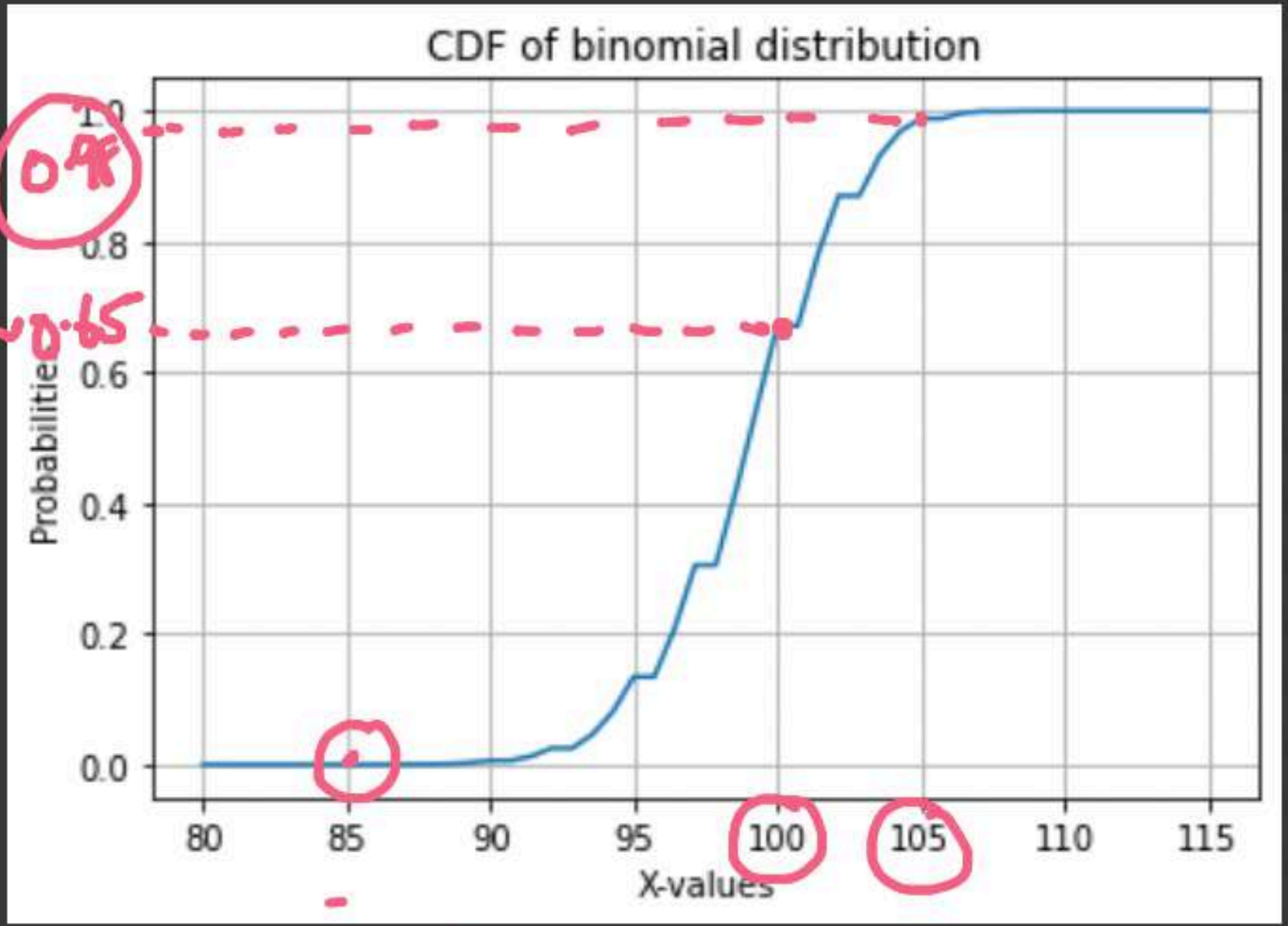
```
#CDF:
x = np.linspace(80, 115)

cdf = showsup_distribution.cdf(x)

plt.plot(x, cdf)
plt.xlabel("X-values")
plt.ylabel("Probabilities")
plt.title("CDF of binomial distribution")
plt.grid()
plt.show()
```




```python
#CDF:
x = np.linspace(80, 115)

cdf = showsup_distribution.cdf(x)

plt.plot(x, cdf)
plt.xlabel("X-values")
plt.ylabel("Probabilities")
plt.title("CDF of binomial distribution")
plt.grid()
plt.show()
```



```python
plt.show()
```



CDF: $P(X \leq 105)$

X = #passengs who board a flight

given $n=110$, $p=0.9$

65% prob that ≤ 100 passengers $\Rightarrow$ 35% prob of >100 passengers

Problem: DS @ Google - HR

population

T-shirts

100k employees

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

1000 employees

Sample

Data

(let) height

Manufacturer

domain knowledge

>180cm: XL

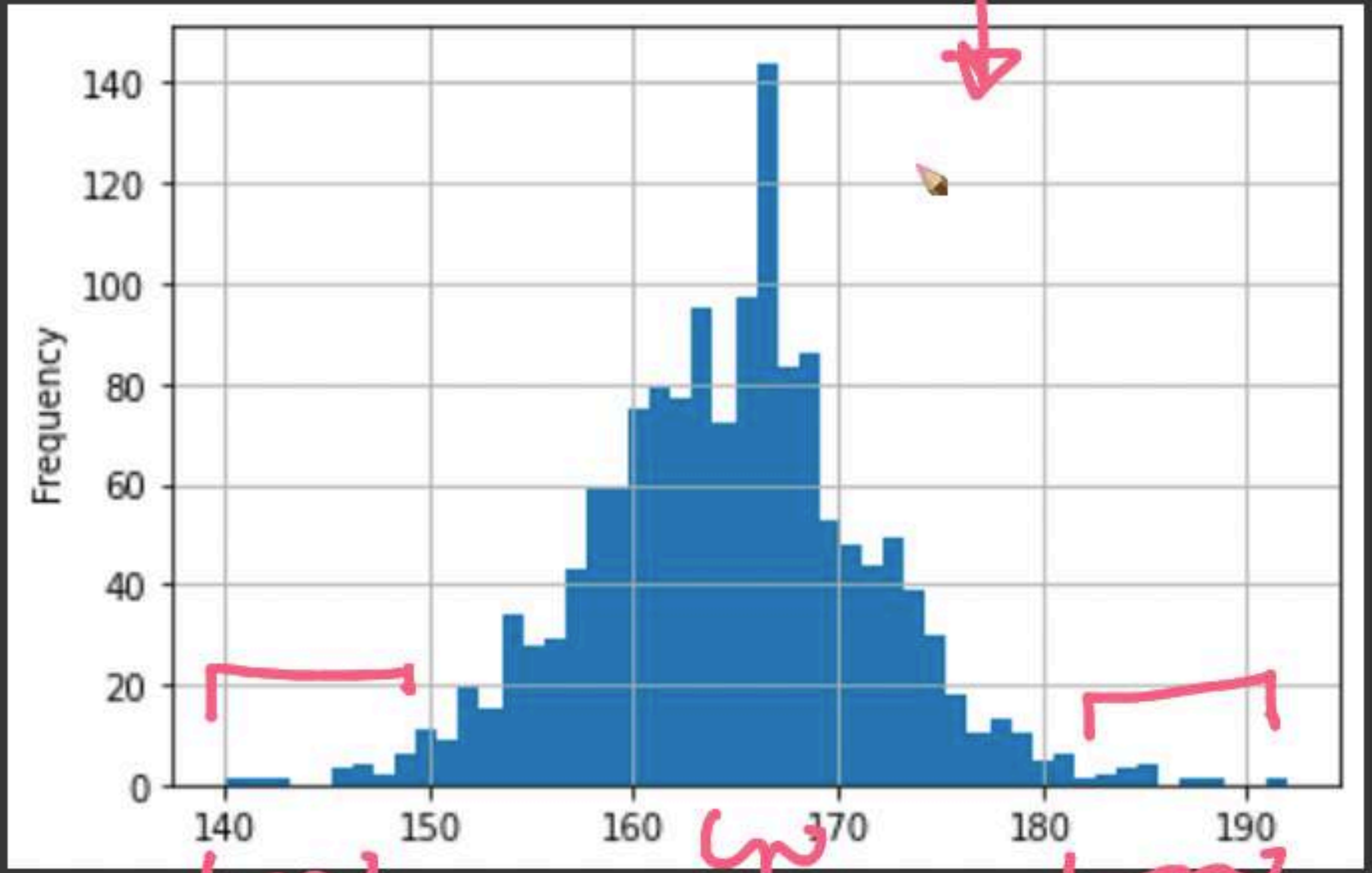
170-180: L

⋮

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

Heights : H : discrete or continous → 182.5 cm

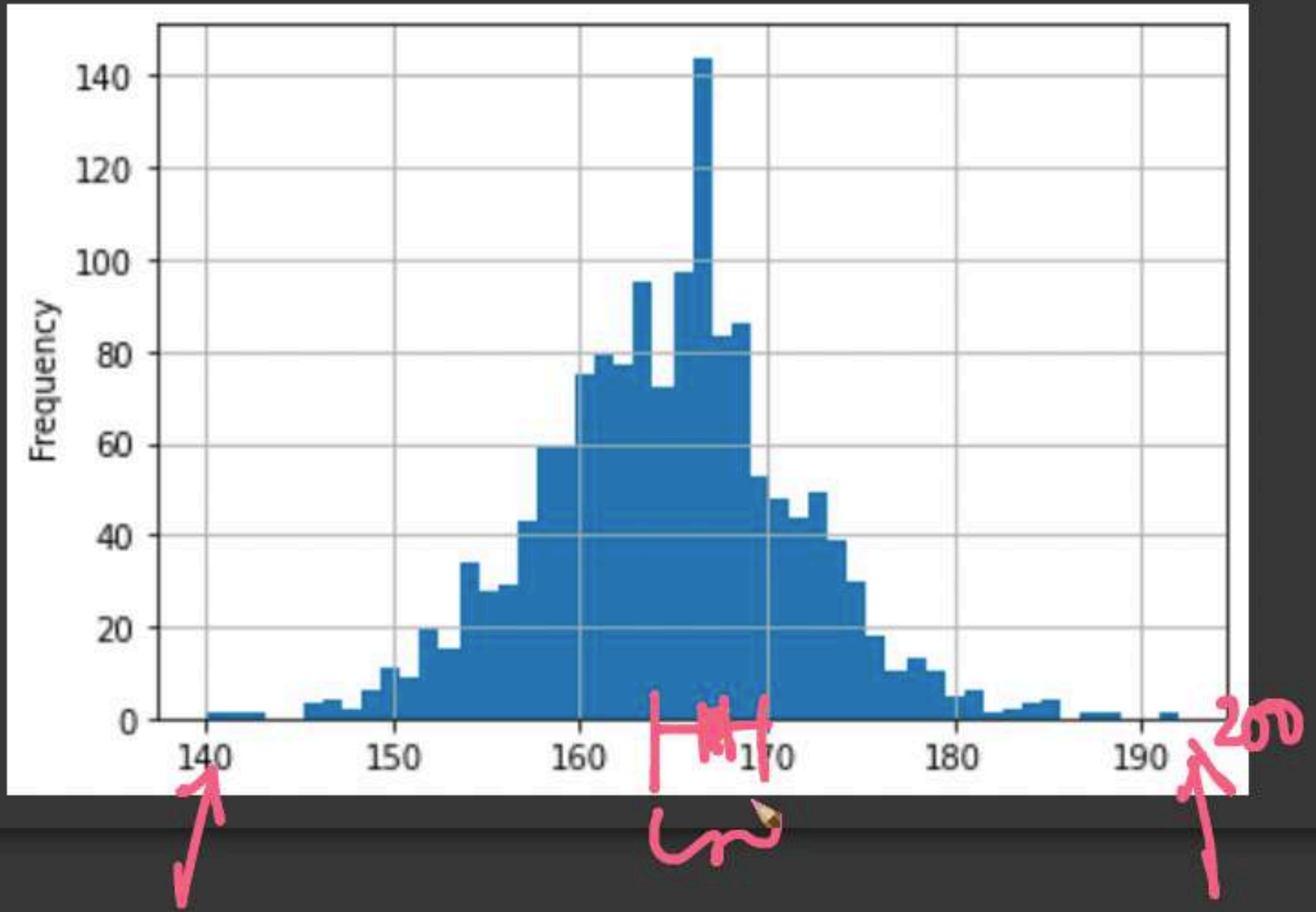
```
employees['Height'].plot(kind='hist', bins=50) # change with the bins value to show
plt.grid()
plt.show()
```



H : Continous

pdf

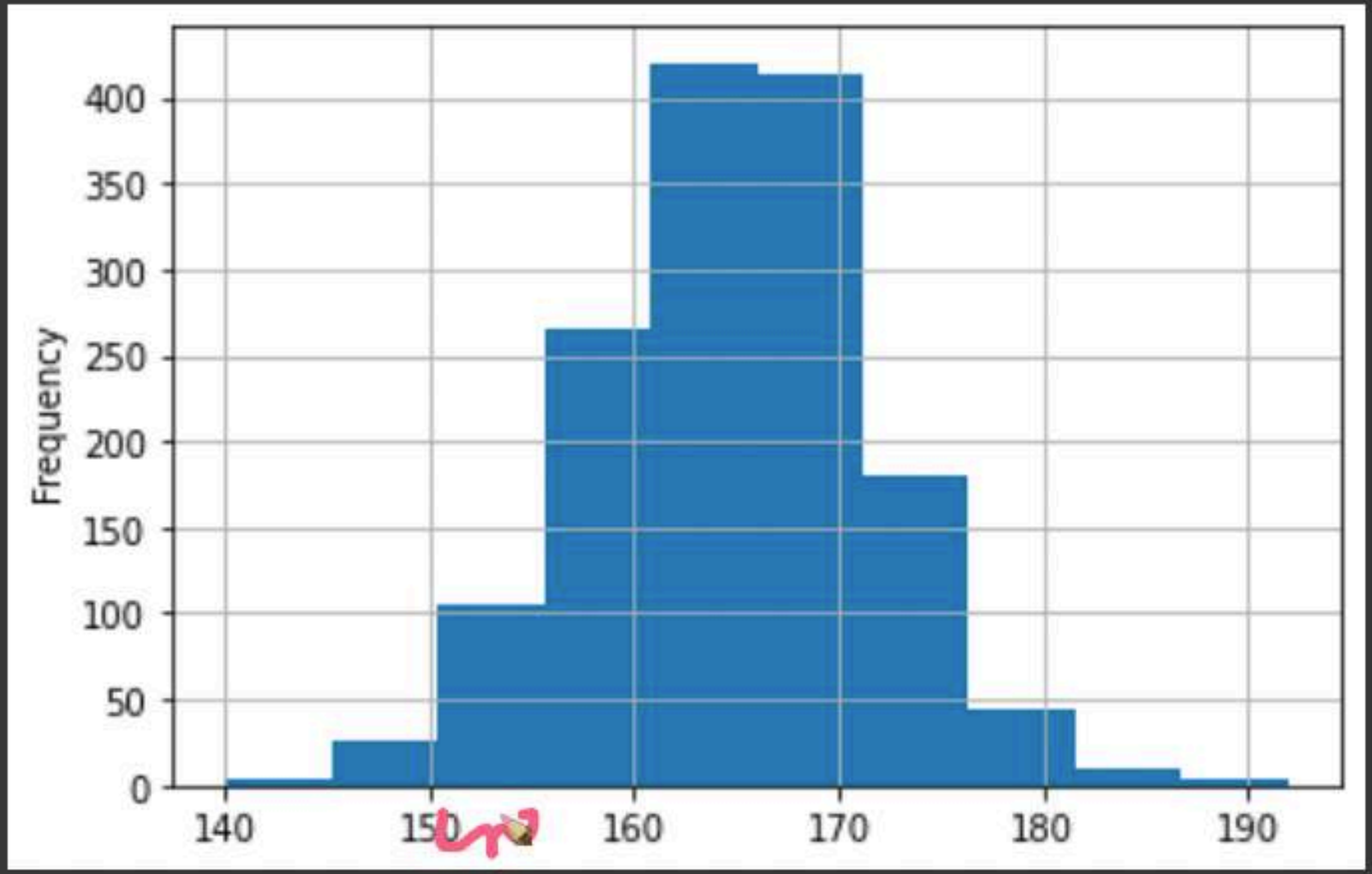
```
employees['Height'].plot(kind='hist', bins=) # change with the bins value to show
plt.grid()
plt.show()
```



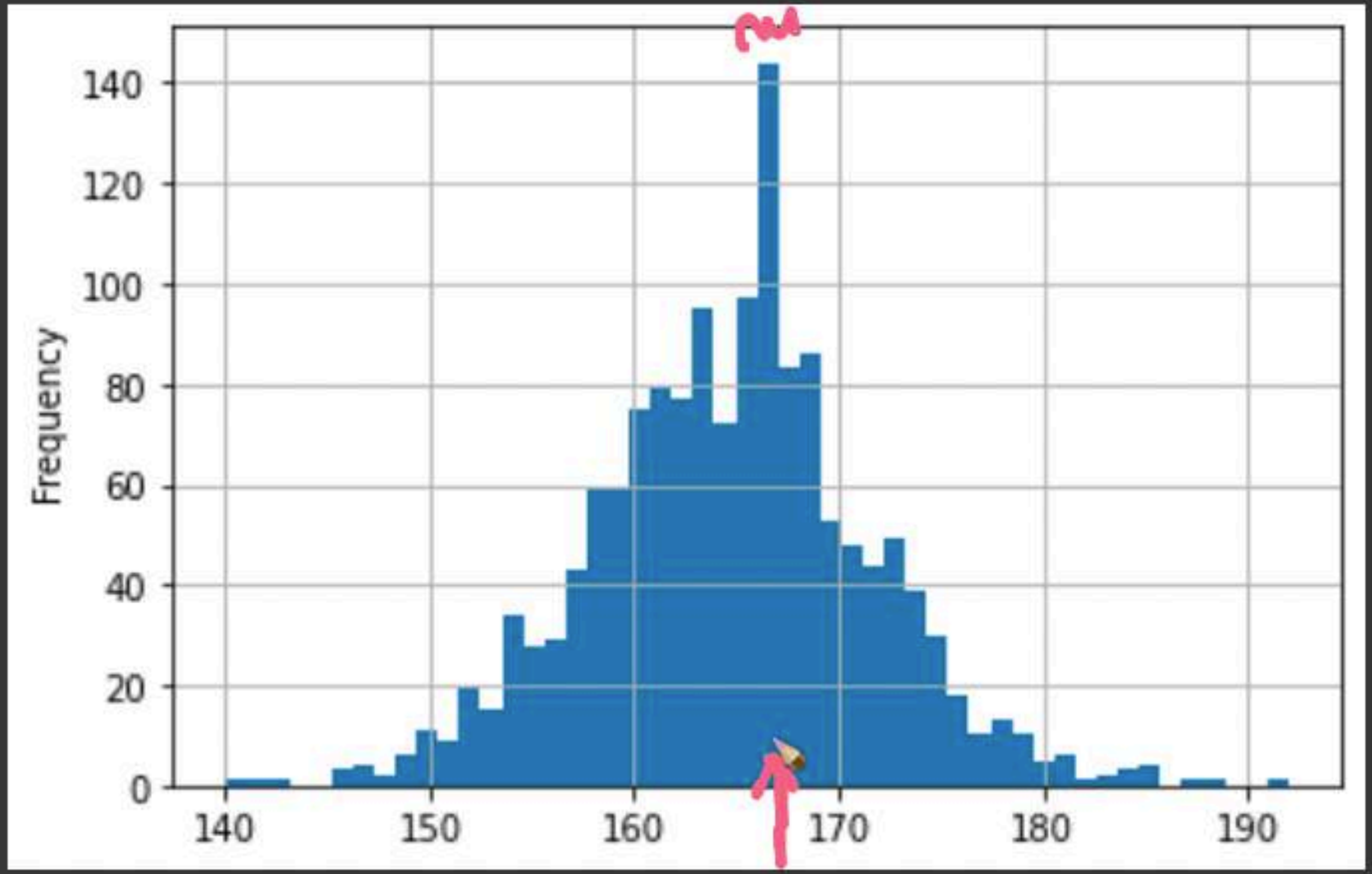
Histogram

50 → 10

```
employees['Height'].plot(kind='hist', bins=10) # change with the bins value to show
plt.grid()
plt.show()
```



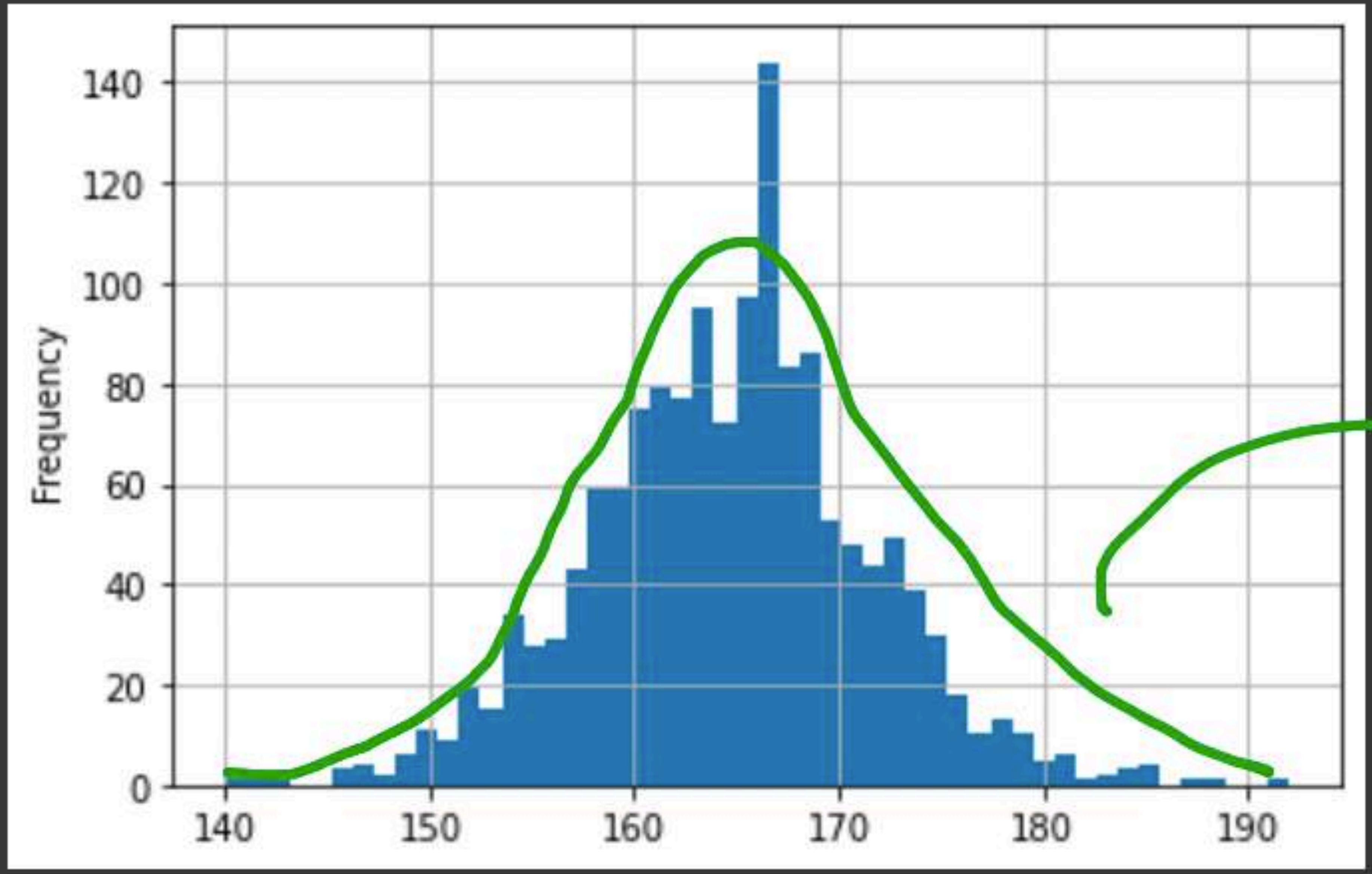
```
employees['Height'].plot(kind='hist', bins=50) # change with the bins value to show
plt.grid()
plt.show()
```



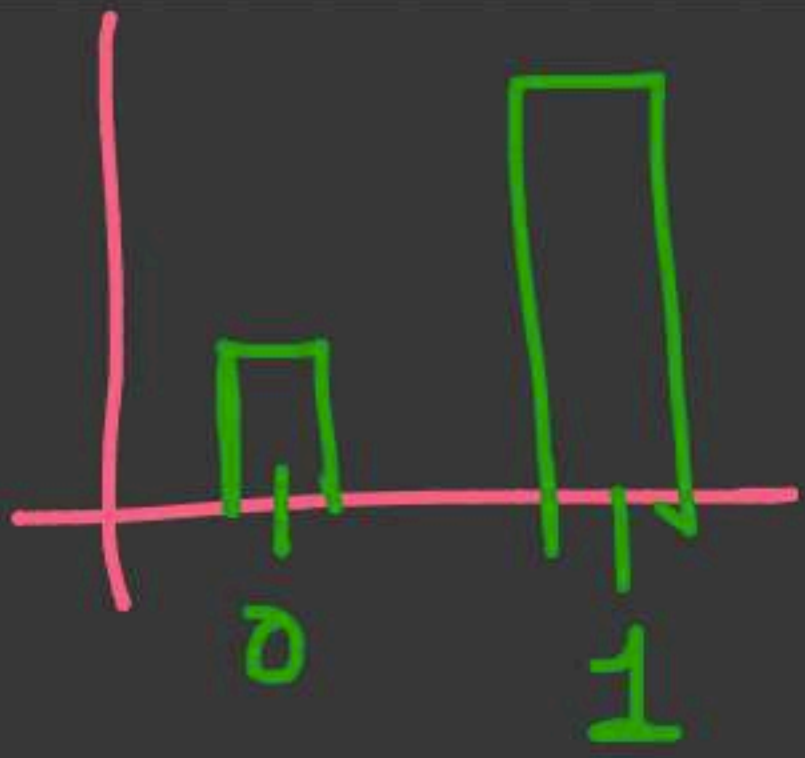
```
employees['Height'].plot(kind='hist', bins=50) # change with the bins value to show
plt.grid()
plt.show()
```

✓ can we conclude $\boxed{H} \sim \text{Binomial}(n,p)$

continous



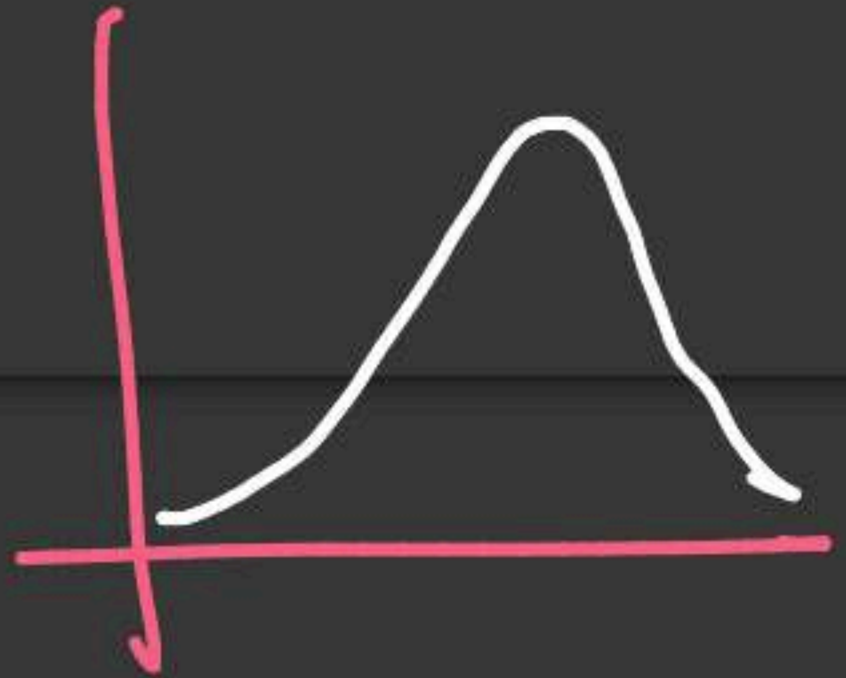
Bernouli



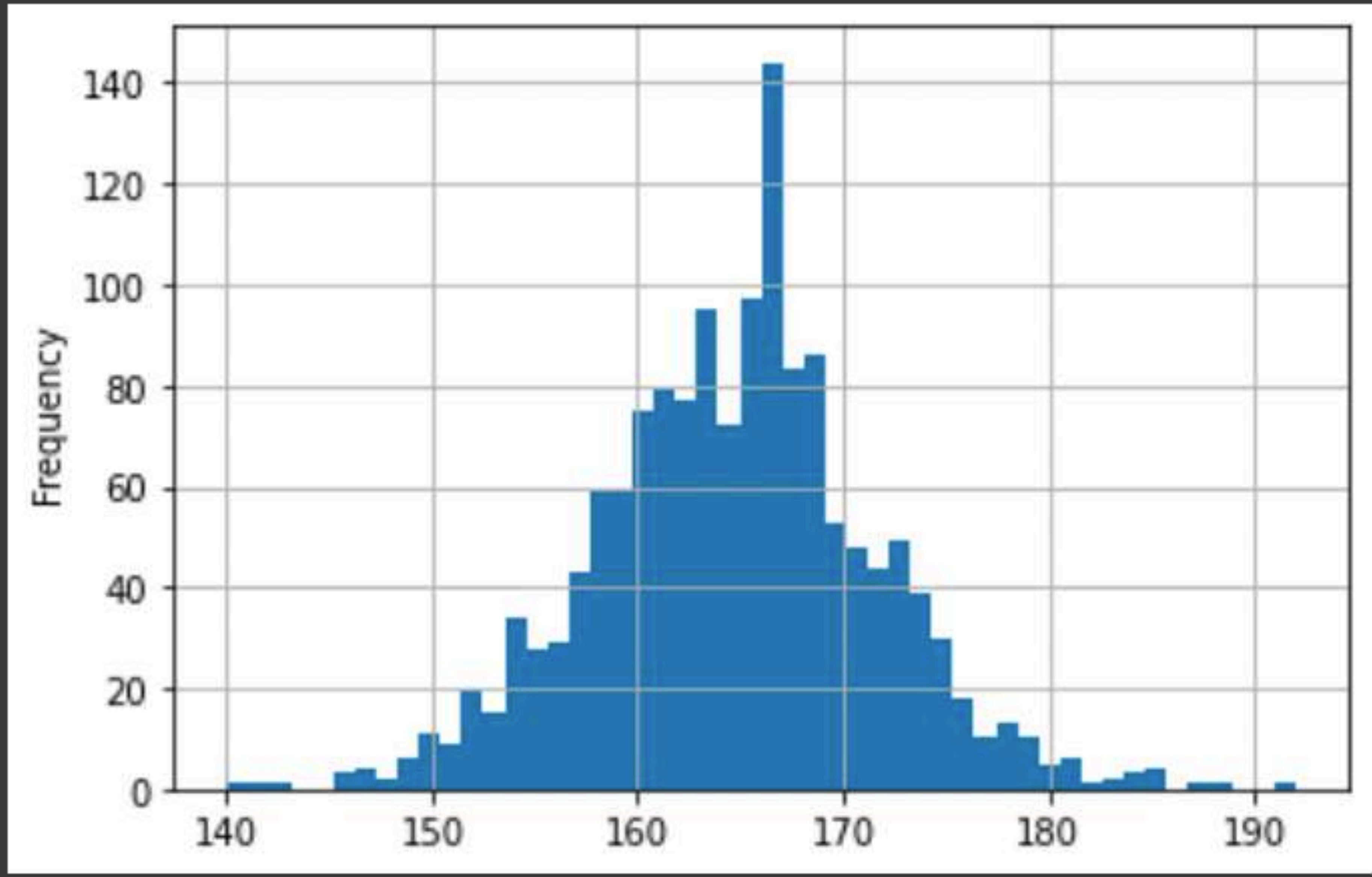
Binomial

discrete r.v

$n = 110$
$p = 0.9$



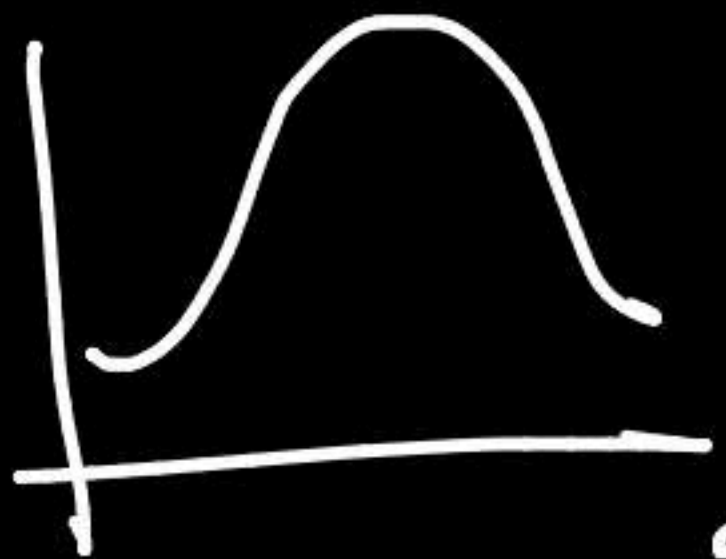
```python
employees['Height'].plot(kind='hist', bins=50) # change with the bins value to show
plt.grid()
plt.show()
```



H: Continous

Binomial $(n, p) \longrightarrow$ discrete

$\boxed{\text{Binomial r.V}}$



n indep bernoulli trails with 'p' of success

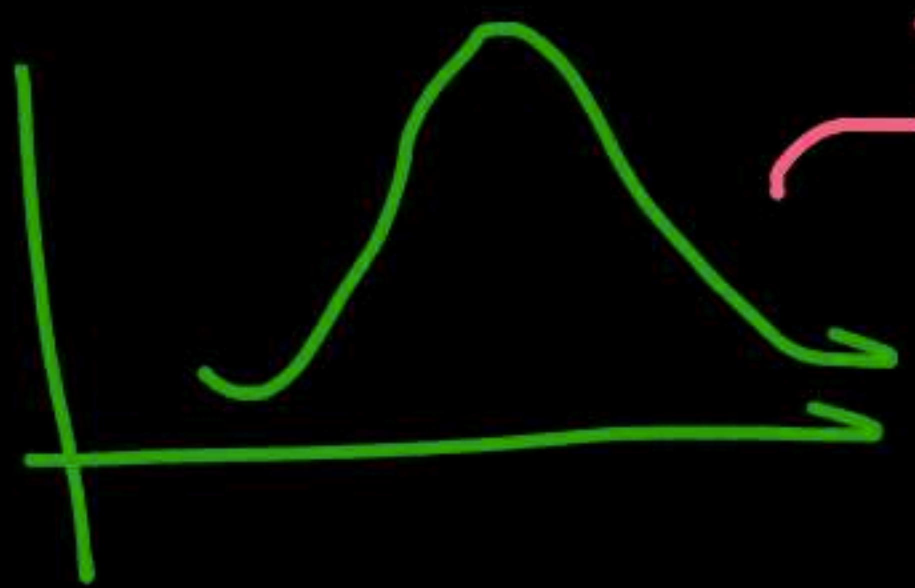
~~X~~ Heights

discrete

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

↓

PMF/PDF



H $\nsim$ Bin(n,p) → conclusions – XX

→ Binomial

1. Check for the conditions ~~for~~ a distb

2. PMF/PDF, CDF → Statistical tests → KS-test → Hyp. testing

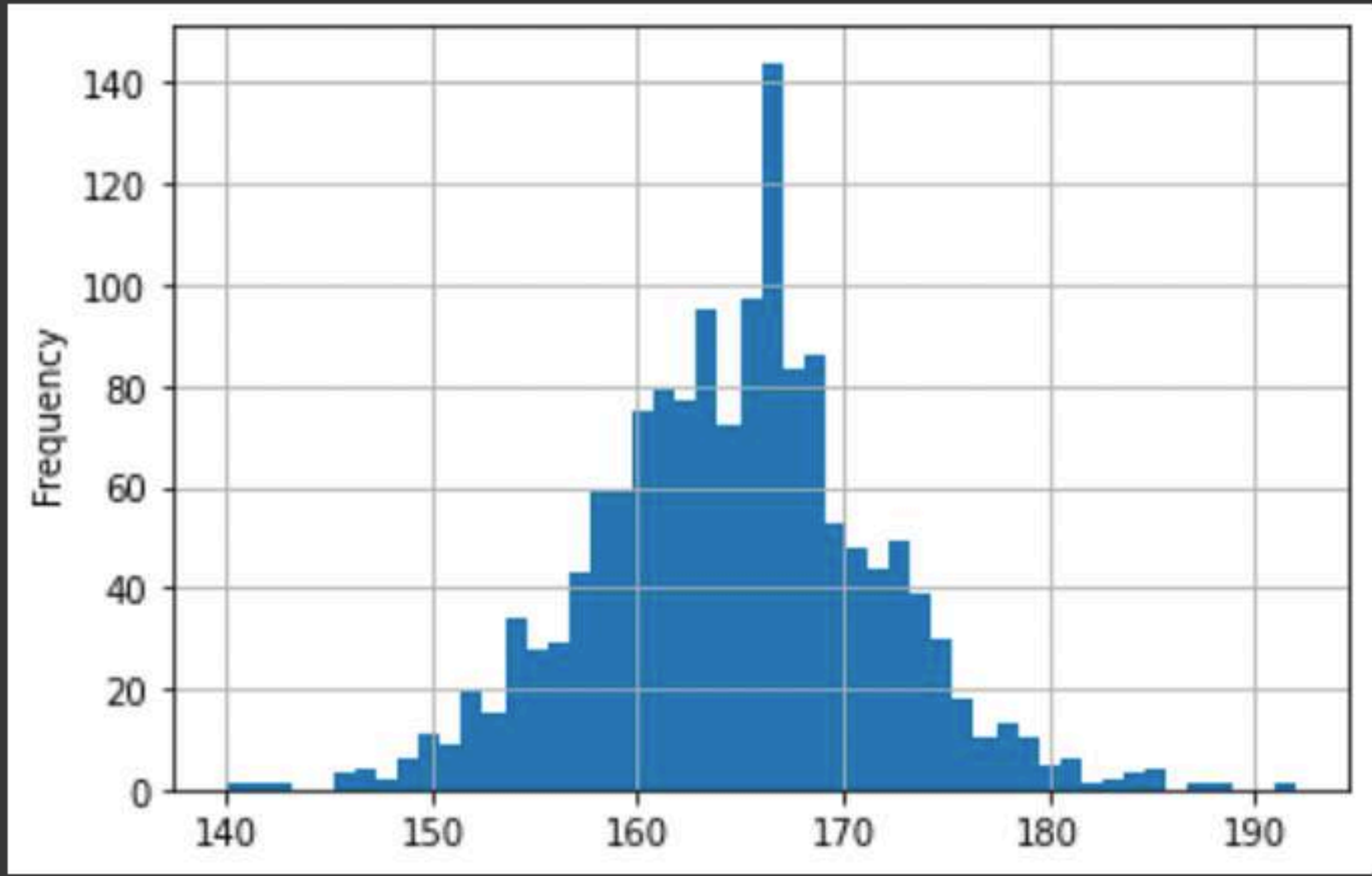
   Plots → QQ-plot

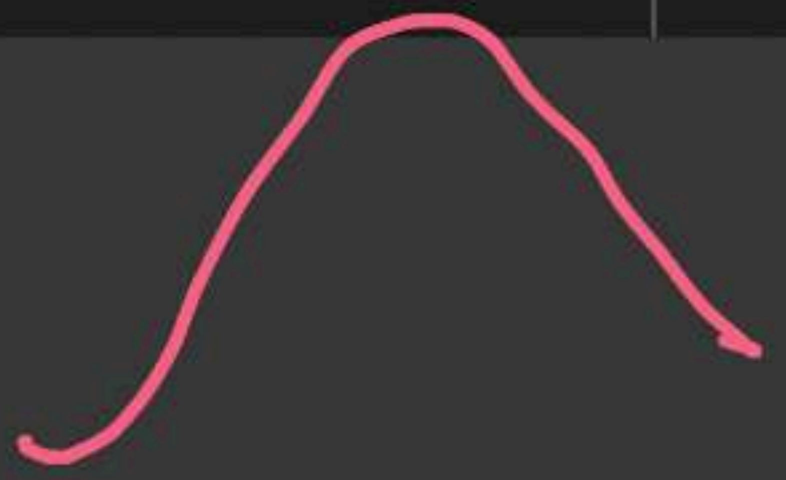
Disb $\rightarrow$ concise model of the disb of data from a natural phenomenon

$x_1, x_2 \dots x_{100}$

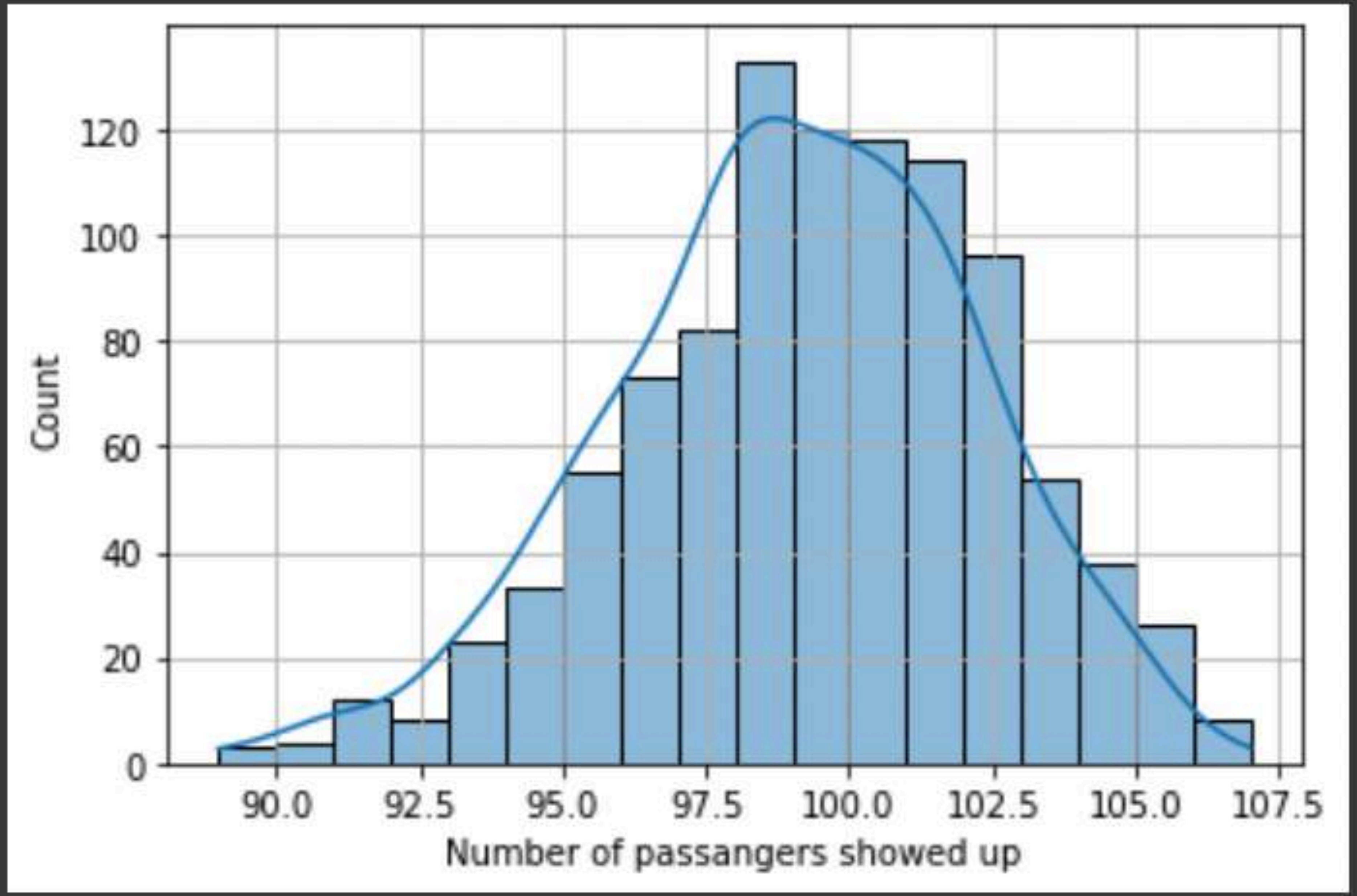
4	27	Research & Development	2	1	7	Male	157	Married	3468

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





```python
plt.xlabel("Number of passangers showed up")
plt.grid()
plt.show()
```



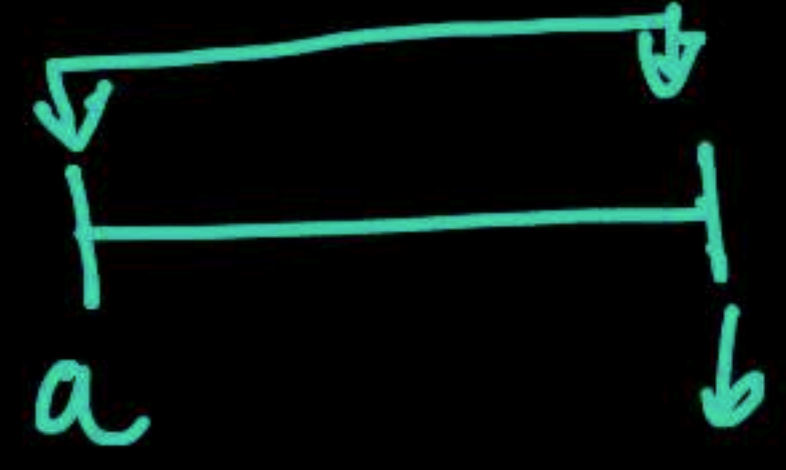
many disb whose shape of PMF/PDF looks similar

```python
#CDF:
x = np.linspace(80, 115)

cdf = showsup_distribution.cdf(x)

plt.plot(x, cdf)
plt.xlabel("X-values")
```

Normal or Gaussian

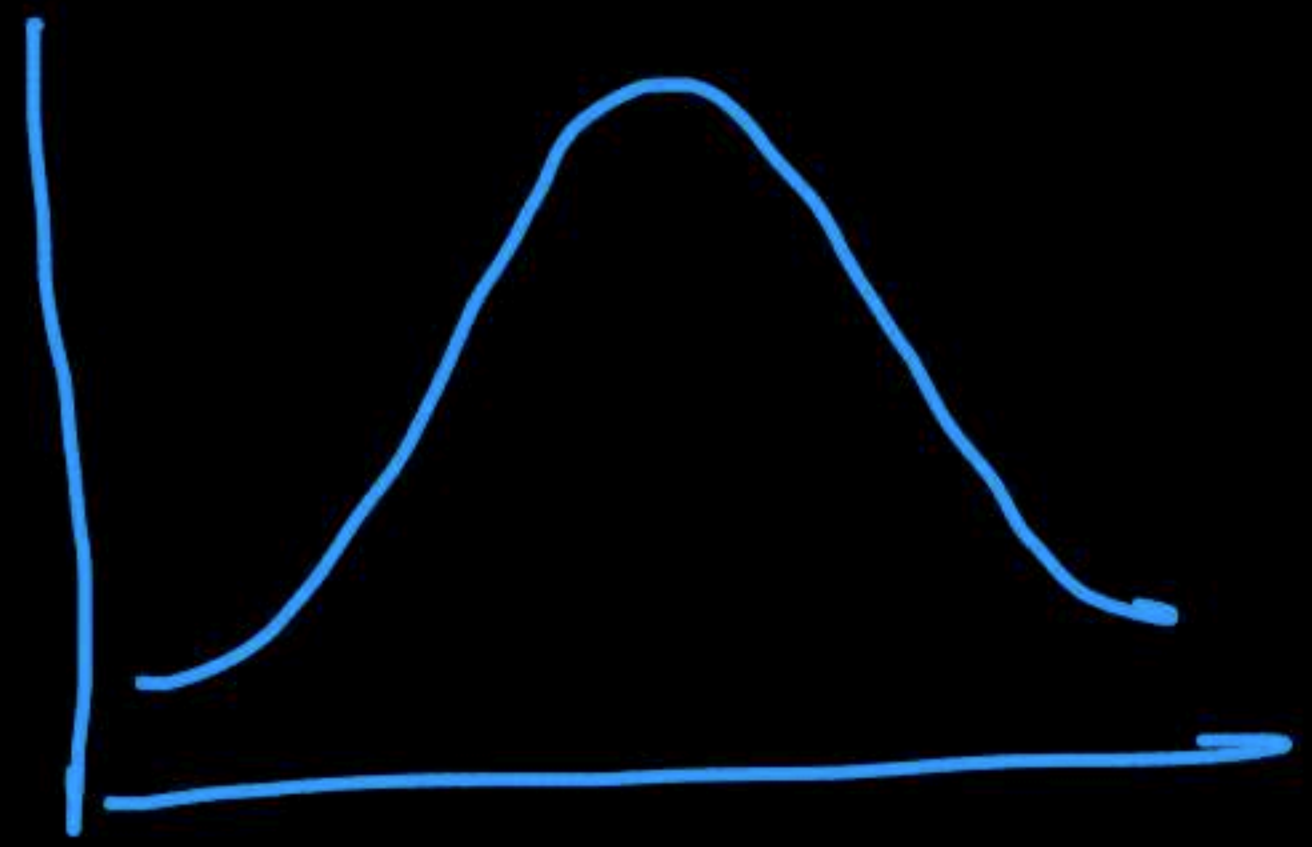
- Heights ✓
- Birth weights ✓
- Shoe sizes ✓
- measurement - error
- sizes of leaves/petals

Telescopes



① → 2.31 cm
② → 2.32 cm
③ → 2.321 cm
⋮

Plot the pdf of measurements

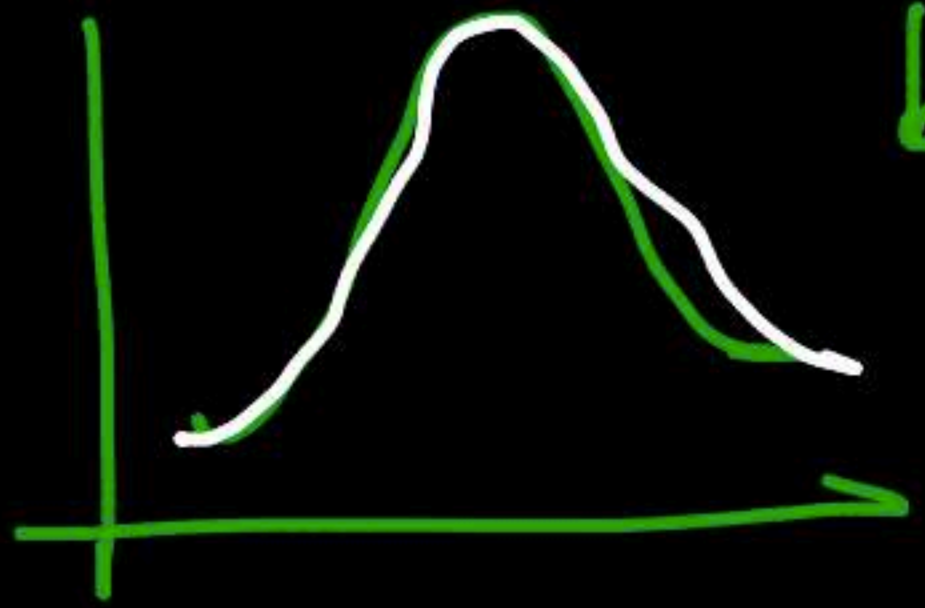


bell-shaped

Binomial(n, p)

blood-pressure of 30yr olds

$\underbrace{x_1, x_2, - - - x_{1000}}$



bell shaped

Normal

statistical tests

KS-test

AD-test

Continous r.v

PDF: bell-curve

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

$x_1, x_2, \ldots, x_{10}$

$X \sim \text{Normal}(\ )$

(μ, σ)

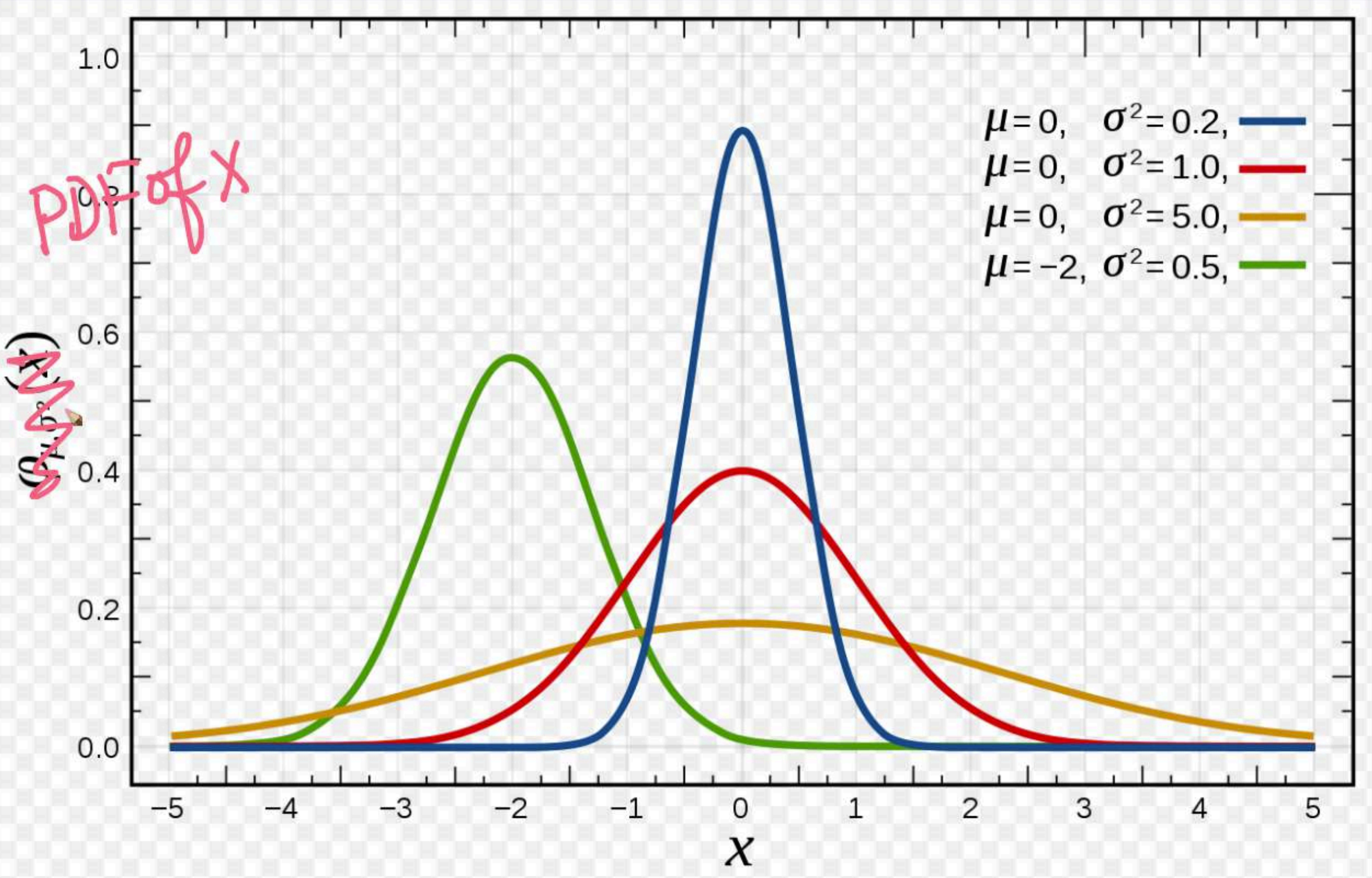
Fshirts

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

Sample mean

Sample std r.v

150

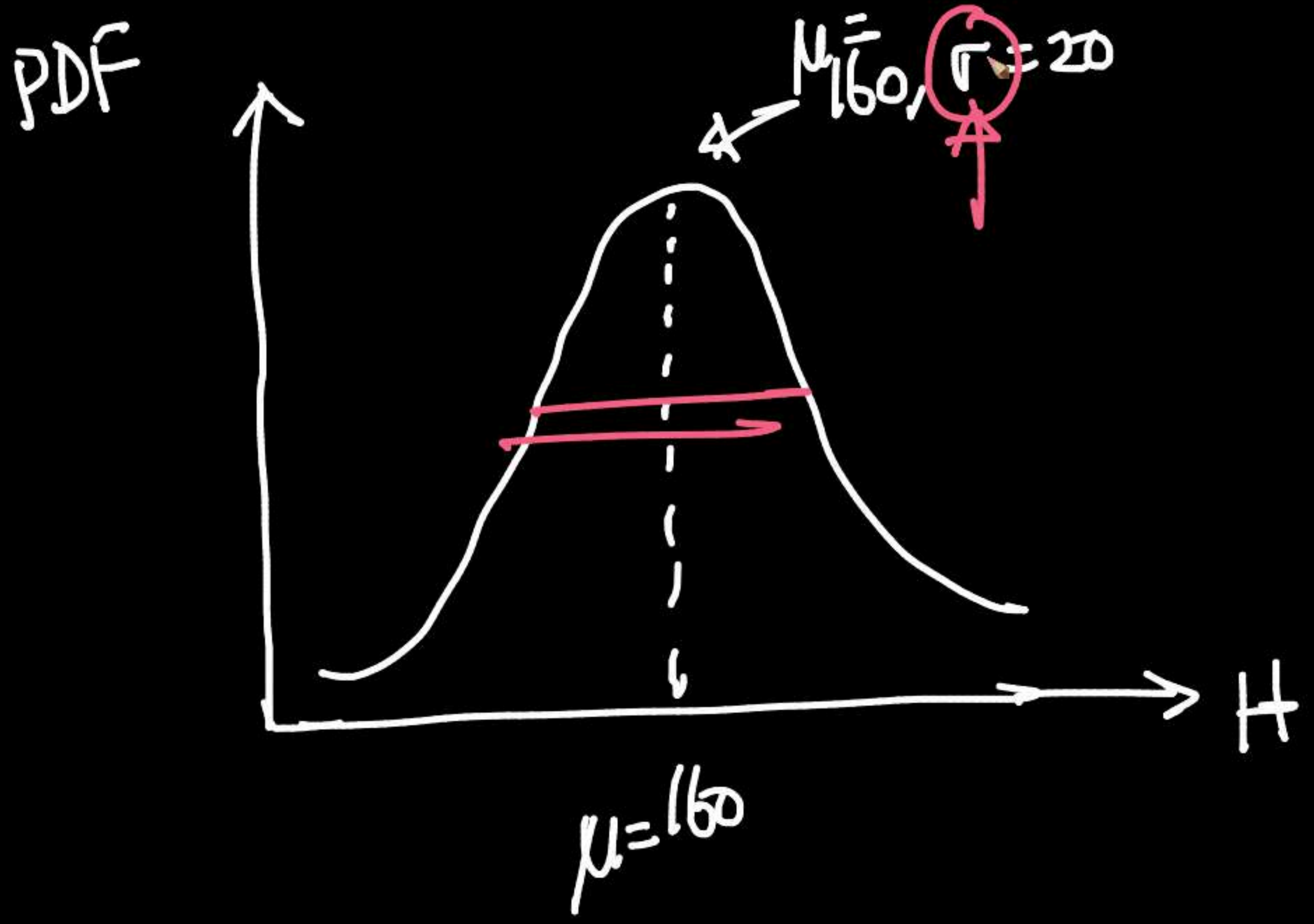


A selection of Normal Distribution Probability Density Functions (PDFs). Both the mean, μ, and variance, σ^2, are varied. The key is given on the graph.

More details

Inductiveload - self-made, Mathematica

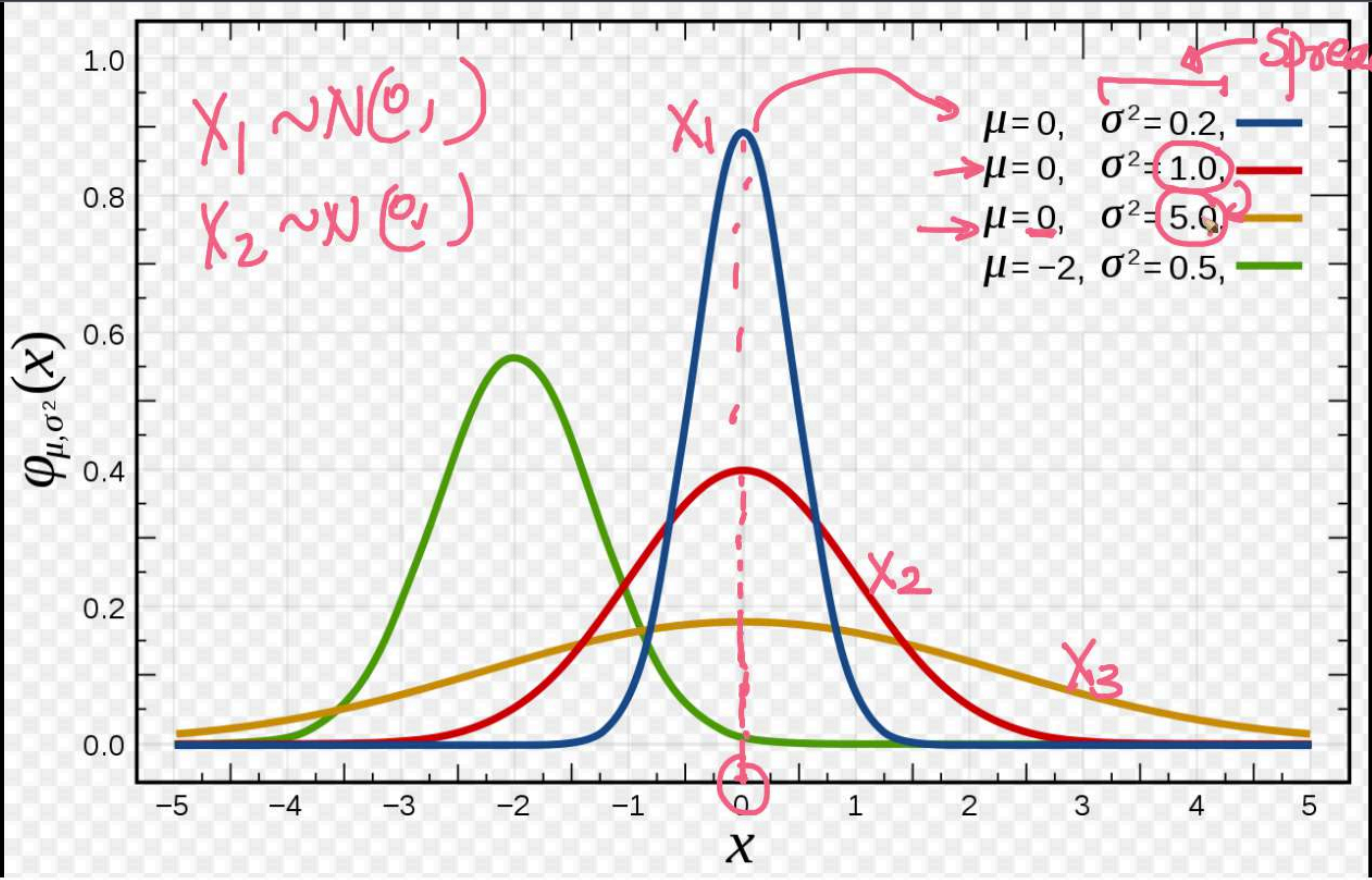
Public Domain view terms



(let)

$$H \sim N(\mu = 160, \sigma = 20)$$

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



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

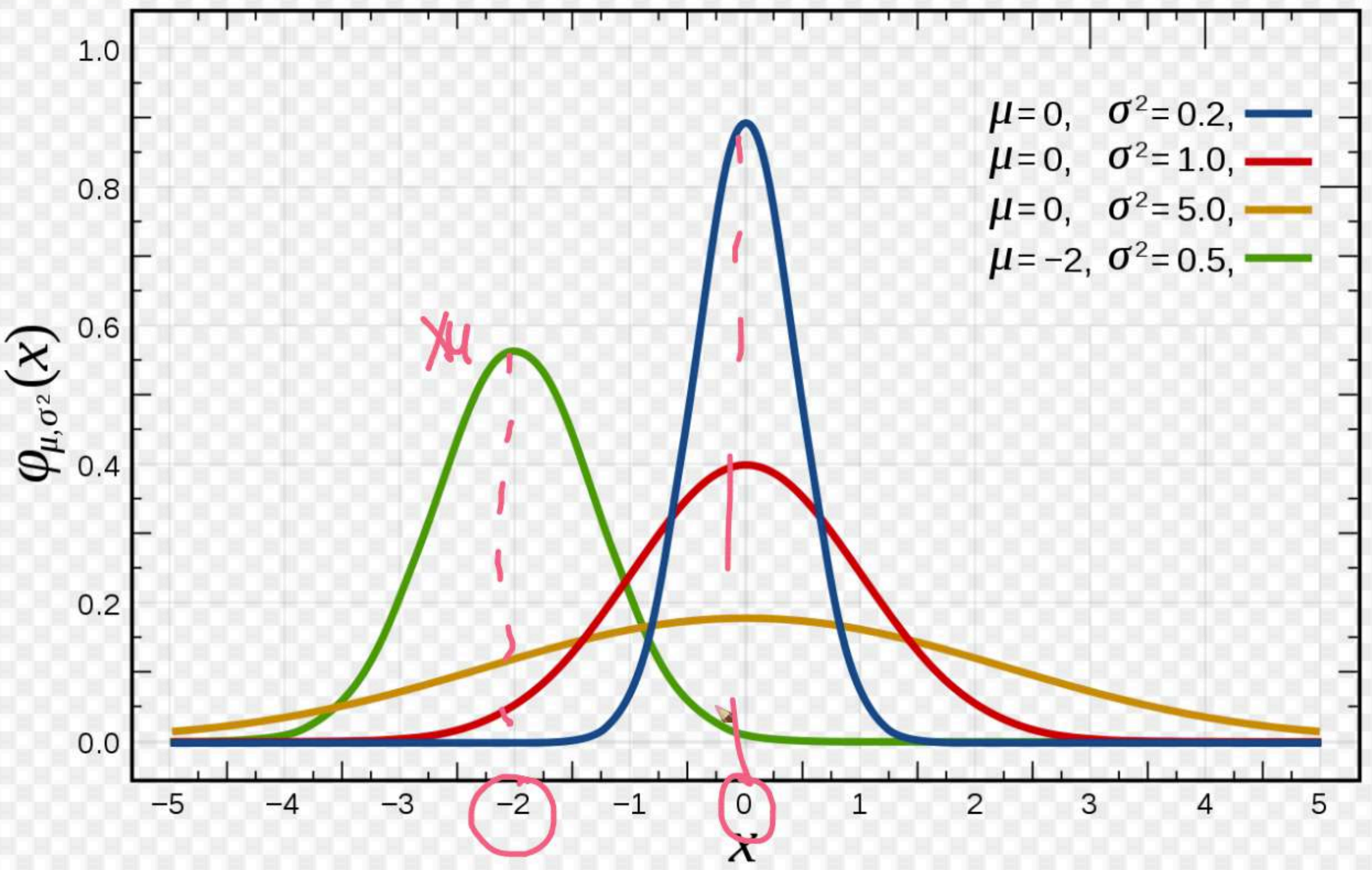
Inductiveload - self-made, Mathematica

Public Domain view terms

Chrome File Edit View History Bookmarks Profiles Tab Window Help 22:07

MoreDistributions.ipynb - Cola | Normal Distribution PDF - Norr | Poisson distribution - Wikiped | Geometric distribution - Wikip | Log-normal distribution - Wiki | scipy.stats.rv_continuous.rvs

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



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

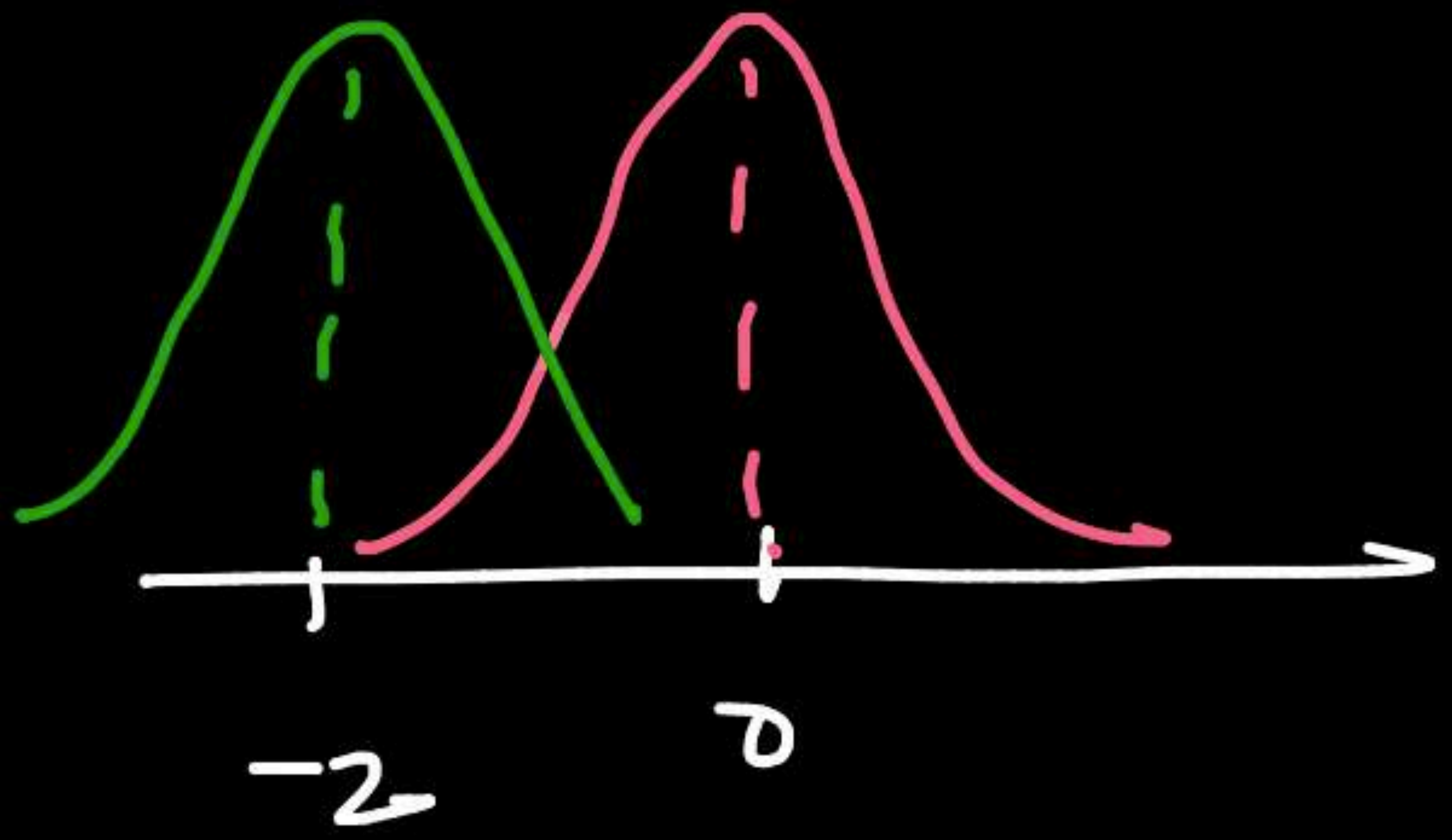
Inductiveload - self-made, Mathematica

Public Domain view terms

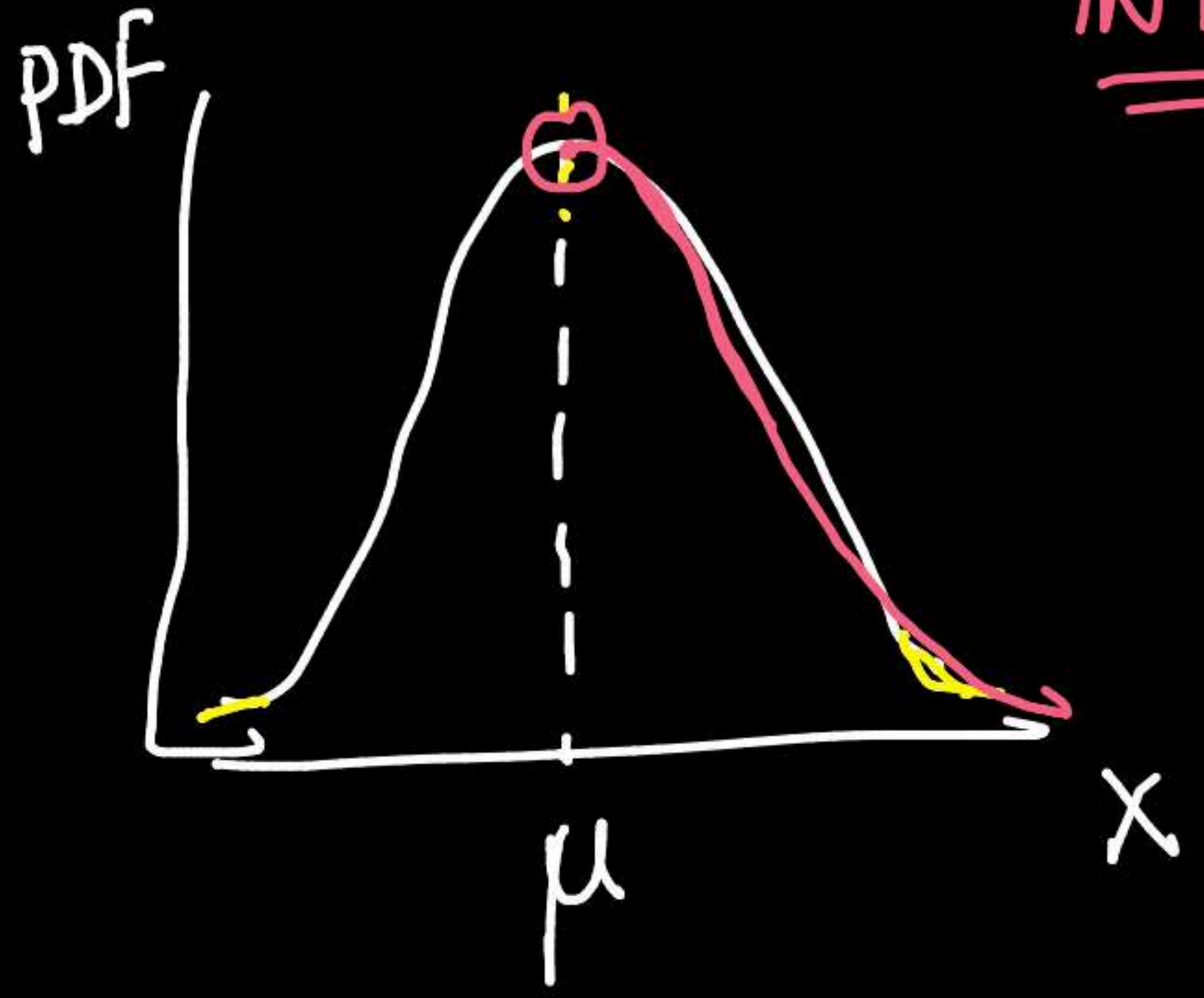
35 / 35

Normally

$$\begin{cases} X_1 \sim \mathcal{N}(0, 2) \\ X_2 \sim \mathcal{N}(-2, 2) \end{cases}$$



Measurement
- errors



INTUITION

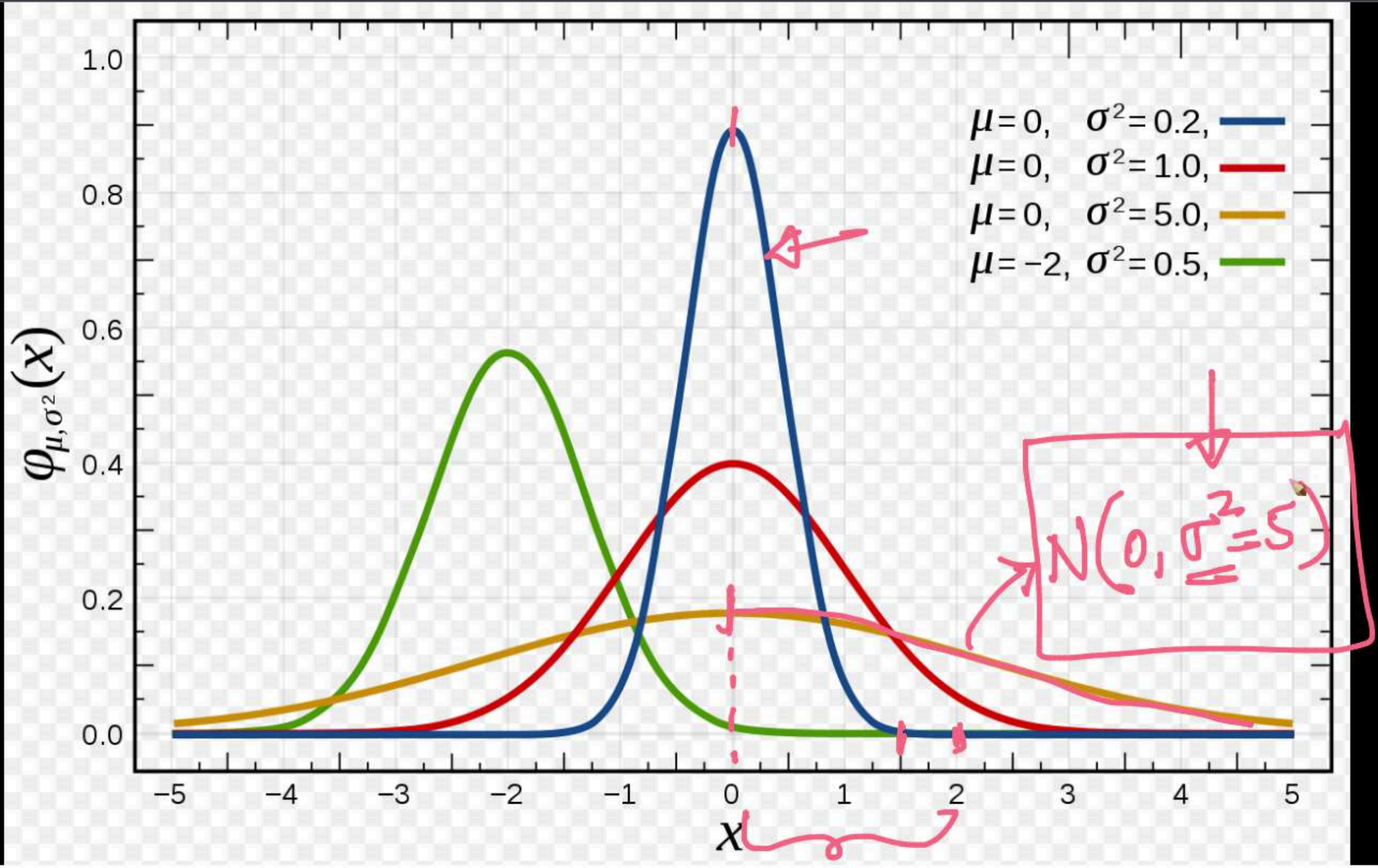
- peak @ μ
- σ determines the thickness/spread of the curve
- Symmetric across μ
- Curves fall @ an exponential rate

model

$H : \text{Normal}(\tilde{\mu}, \tilde{\sigma})$

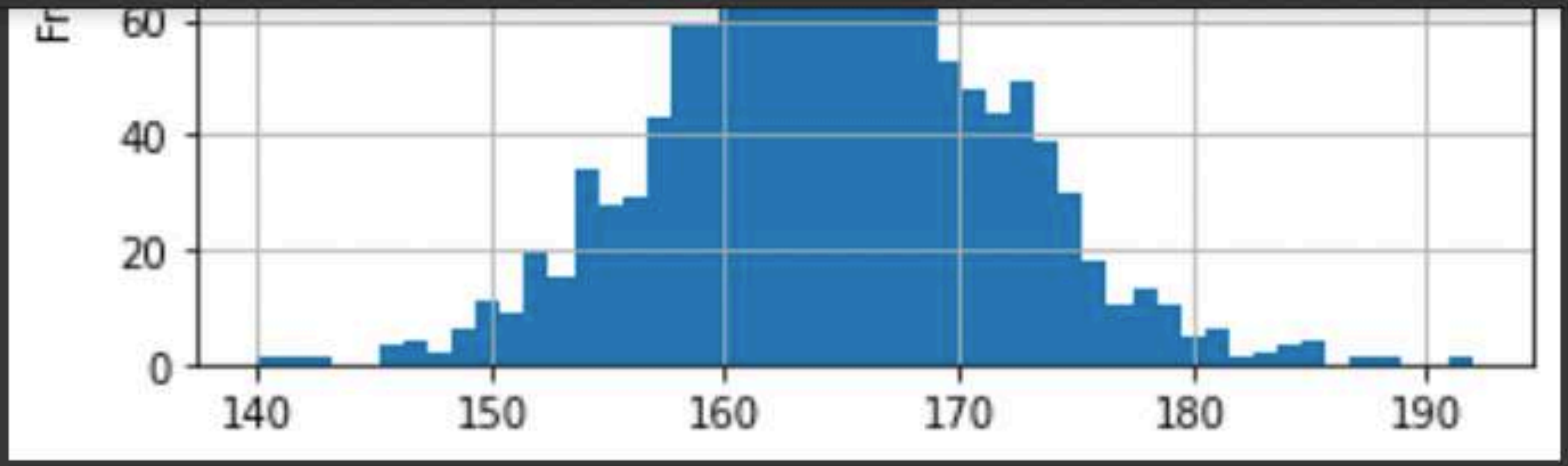
estimate from data

$\boxed{\text{T-shirts}}$



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




```
[22] # 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

1stn

② params

① μ σ

$H \sim N(164.67, 6.89)$

PDF & CDF

Model

[22]

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

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

164.6734693877551

$H \sim N(164.67, 6.89)$

[23]

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

6.887961959078209

μ σ

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

164.6734693877551

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

6.887961959078209

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

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

data 1470 → model

N(μ, σ)

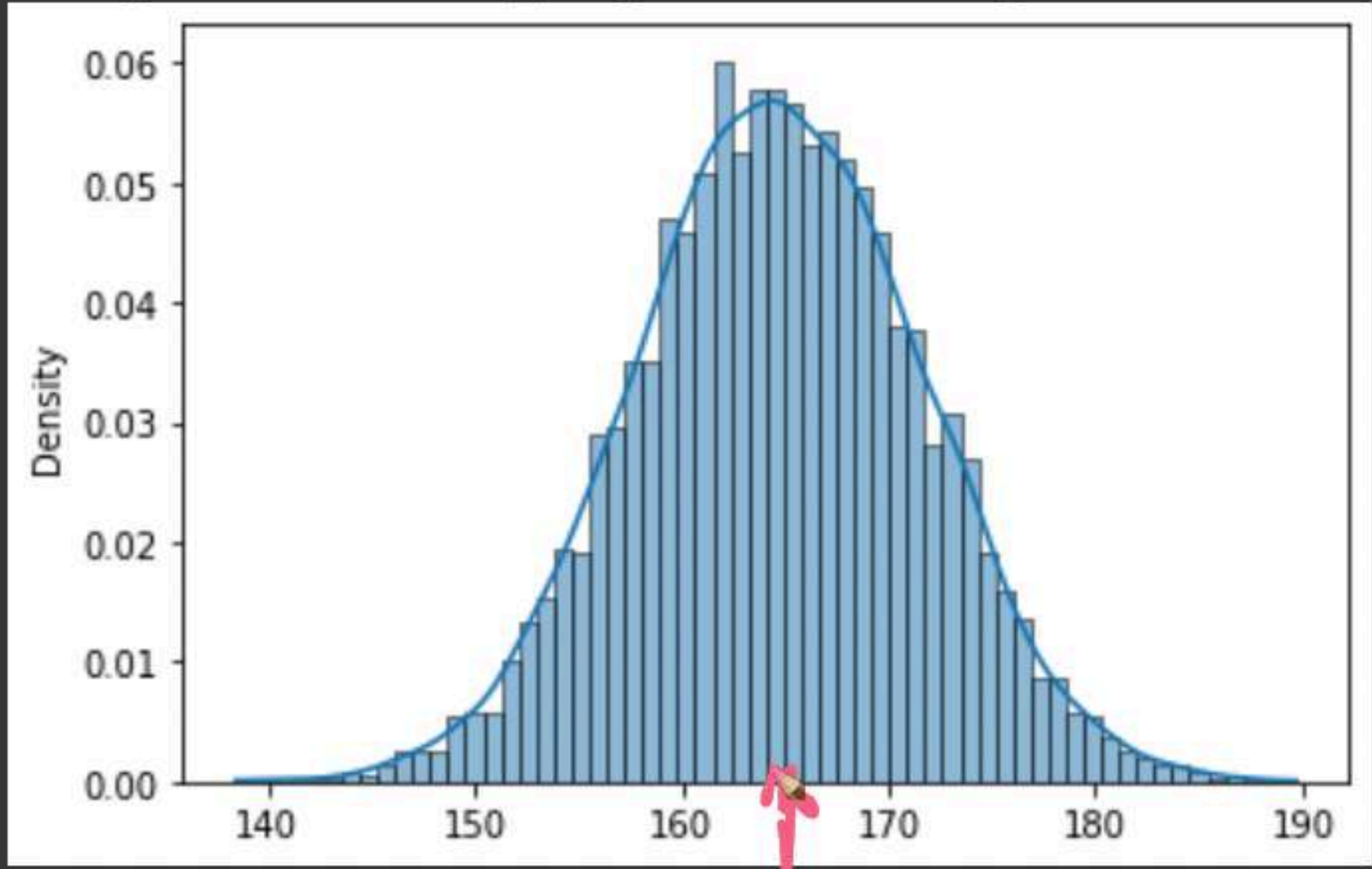
10k points

6.887961959078209

```
[24] 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 0x7f6c28fc2650>



$\boxed{Q}$

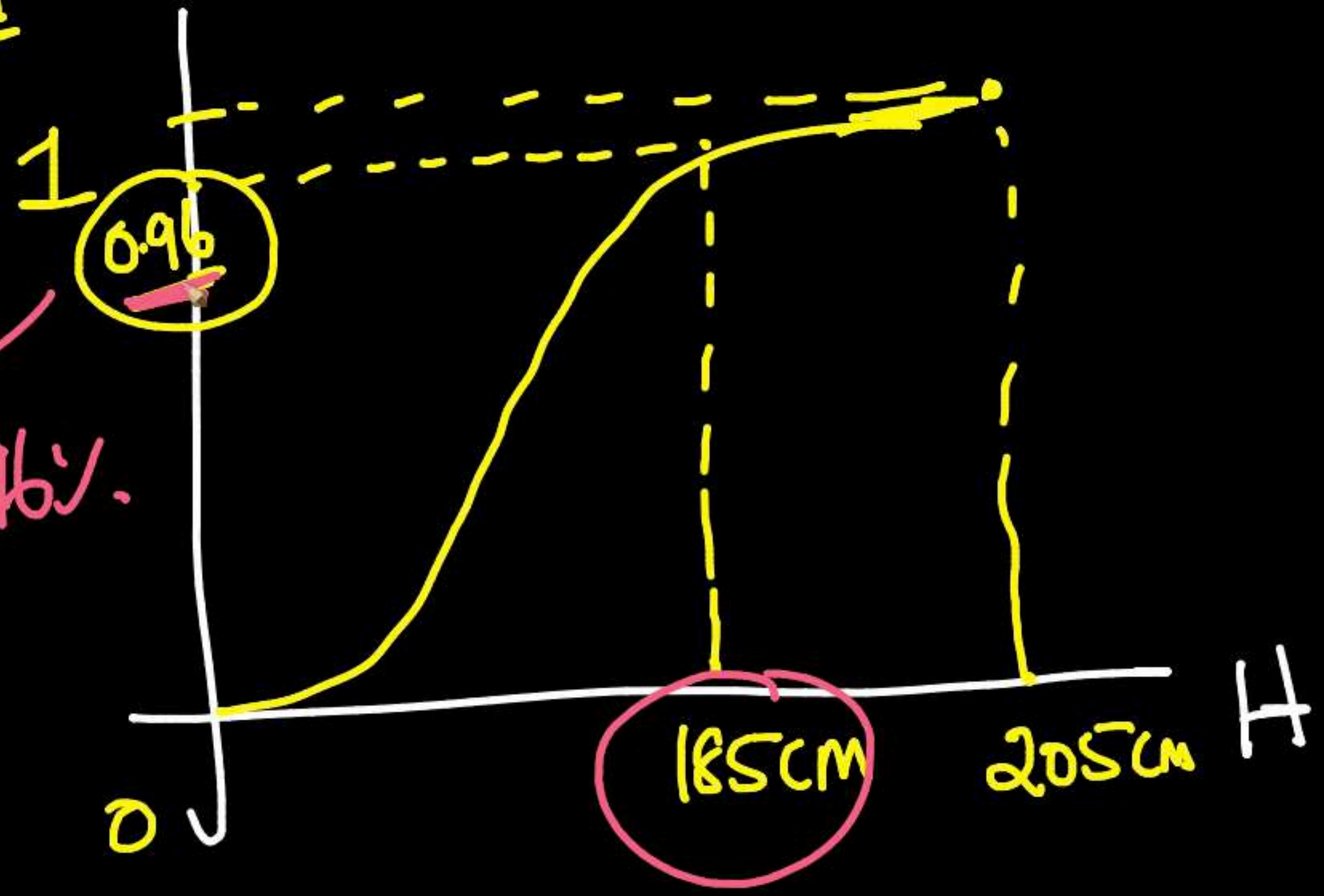
$$H \sim N(\mu = 164.67,\ \sigma = 6.89) \rightarrow \boxed{10K}$$

$\downarrow$

PDF, CDF (scipy)

T-shirt sizes: XL: $\boxed{> 185\ \text{cm}}$

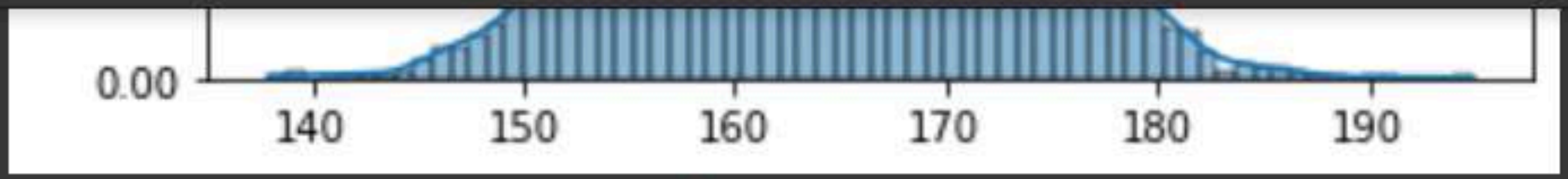
CDF



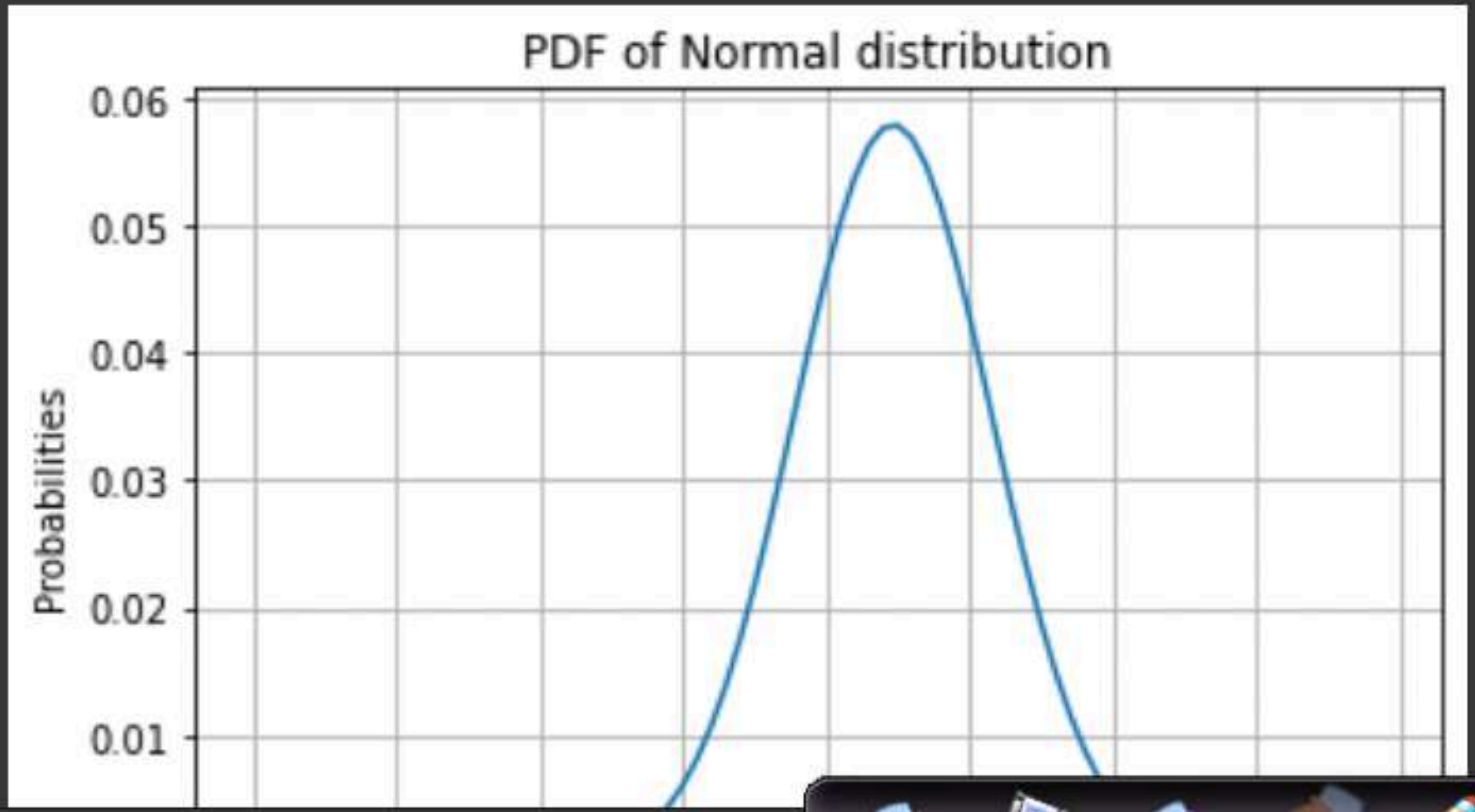
$P(H \le 185) = 96\%$

$P(H > 185)$

$= \underline{4\%}$




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

# Plotting pdf using SCIPY
pdf = height_dist.pdf(x) #scipy plotting
plt.plot(x, pdf)
plt.xlabel("Heights")
plt.ylabel("Probabilities")
plt.title("PDF of Normal distribution")
plt.grid()
plt.show()
```



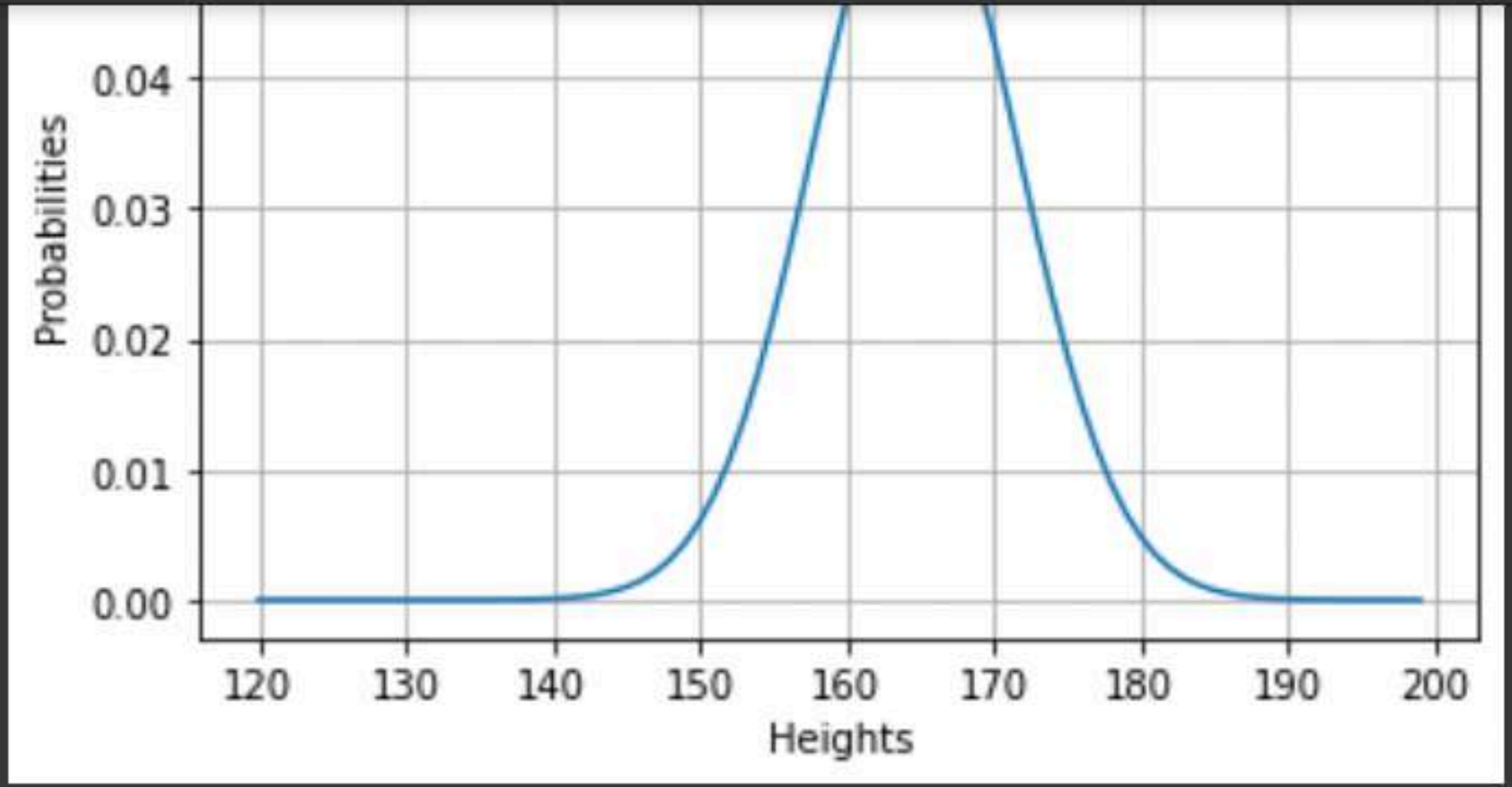
MoreDistributions.ipynb - Colab | Normal Distribution PDF - Nor | Poisson distribution - Wikiped | Geometric distribution - Wikip | Log-normal distribution - Wiki | scipy.stats.rv_continuous.rvs

colab.research.google.com/drive/1TetdffpLSis3xG6bRy1M8Zoow8DUrML4#scrollTo=93s4k1Mq6LkE

+ Code + Text

RAM
Disk

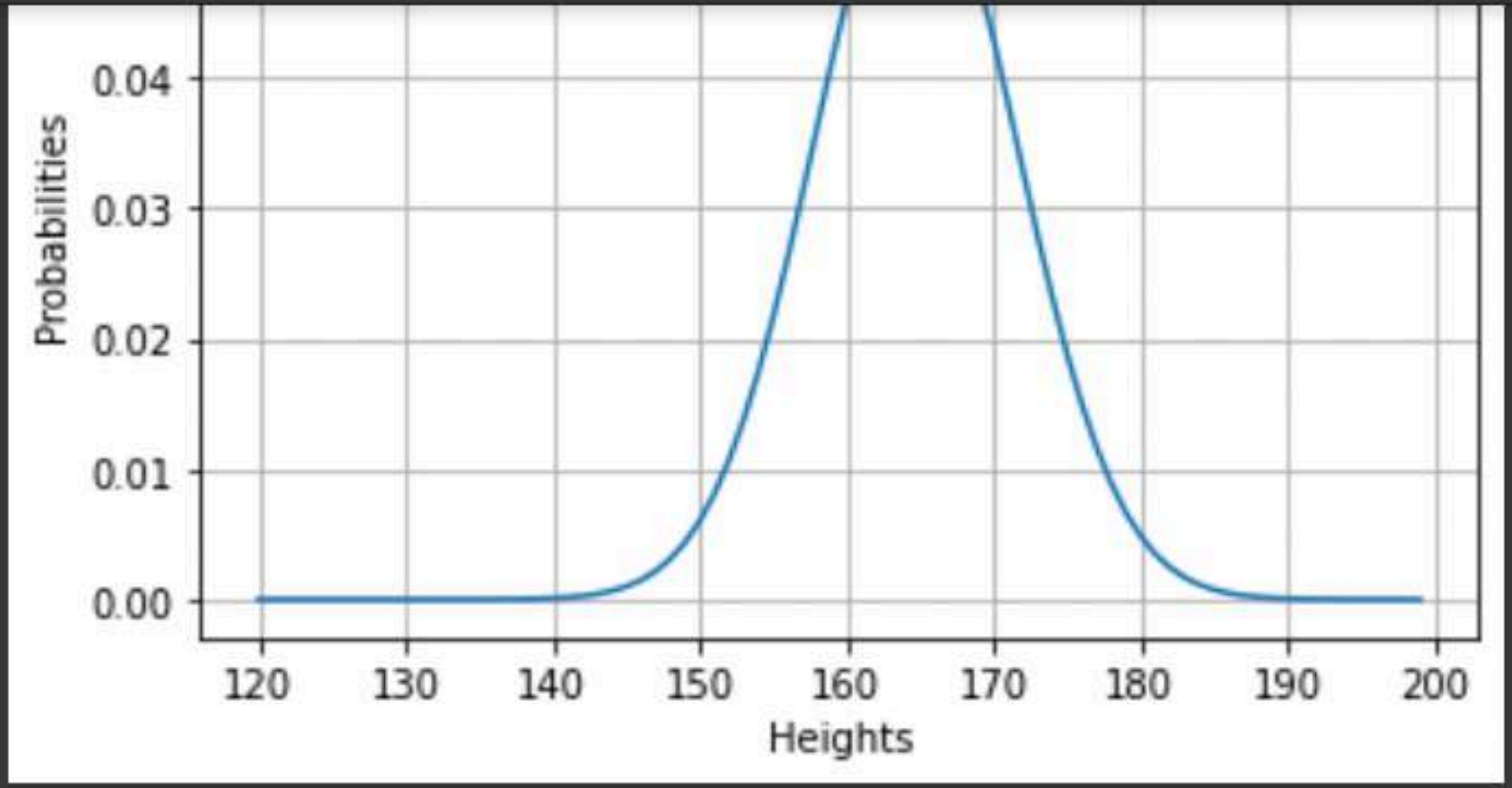
[32]



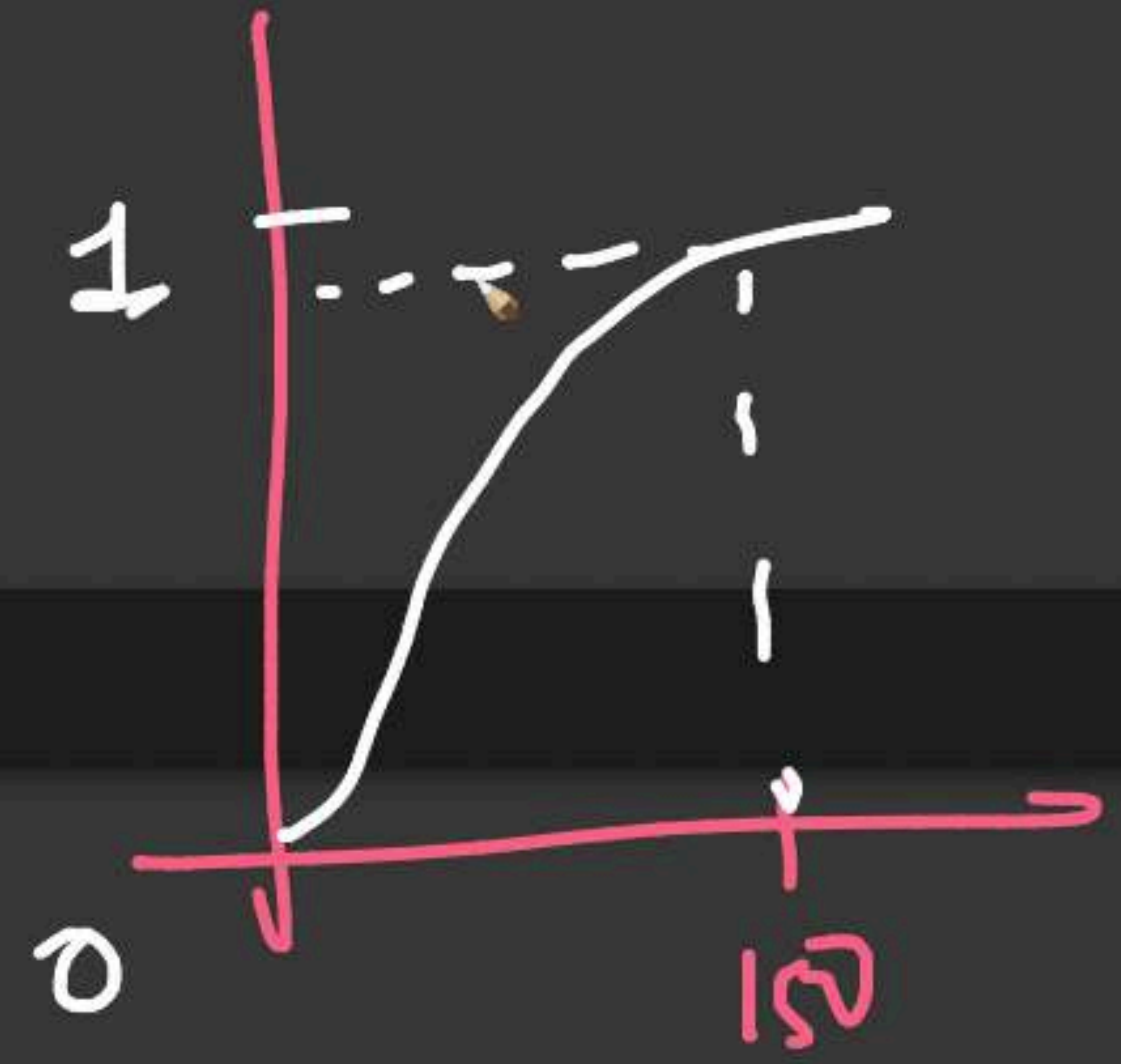

```
height_dist.cdf(x = 150)    # P(X<=150)
```

$N(\mu, \sigma)$

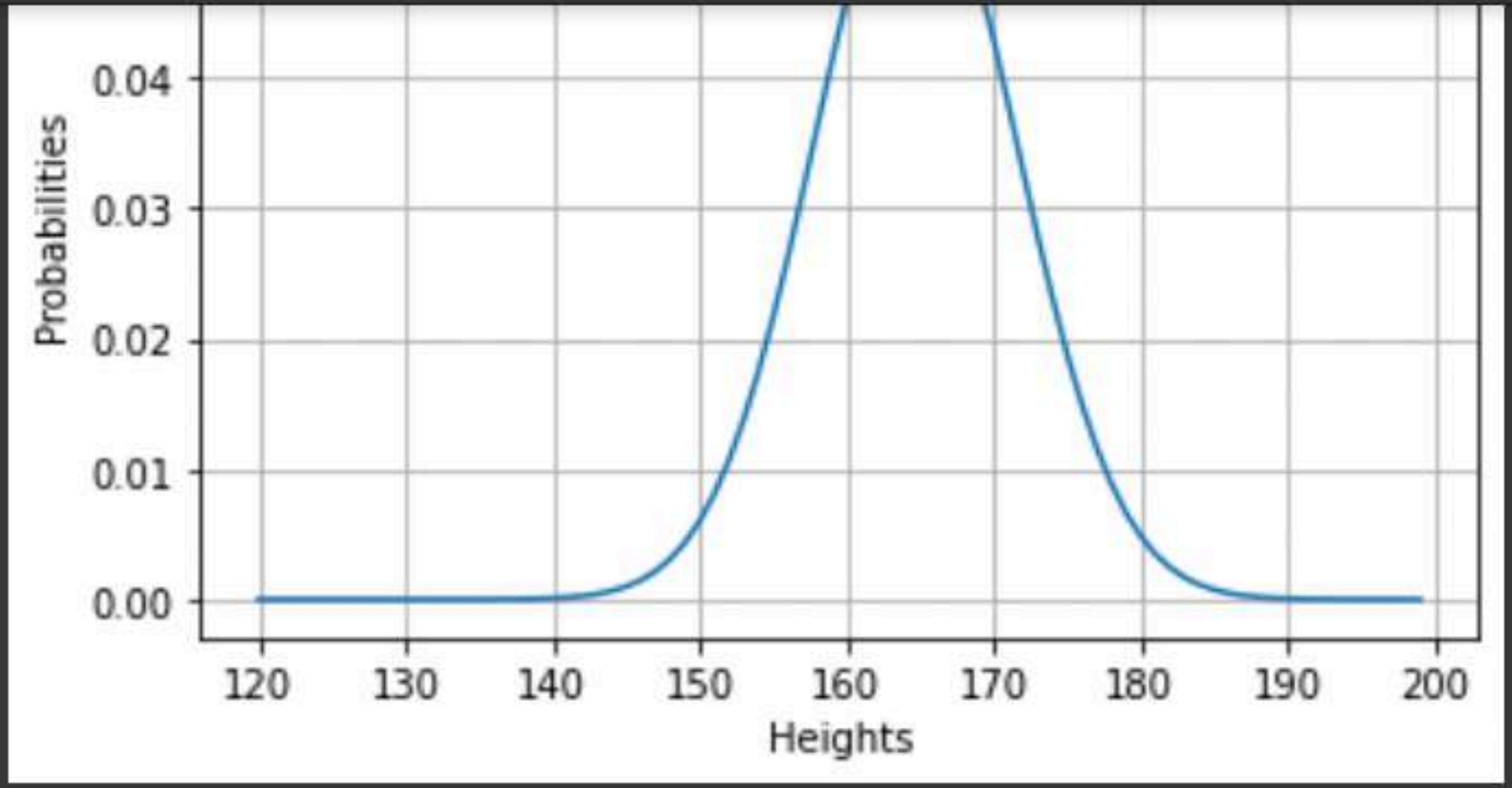
48 / 48




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



[32]



[33]
```
height_dist.cdf(x = 150)   # P(X<=150)
```

0.016573163101179463

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

1.66% X 10,000 = 166

CM

150 - 155 : XS

155 - 160 : S

160 - 170 : M

170 - 180 : L

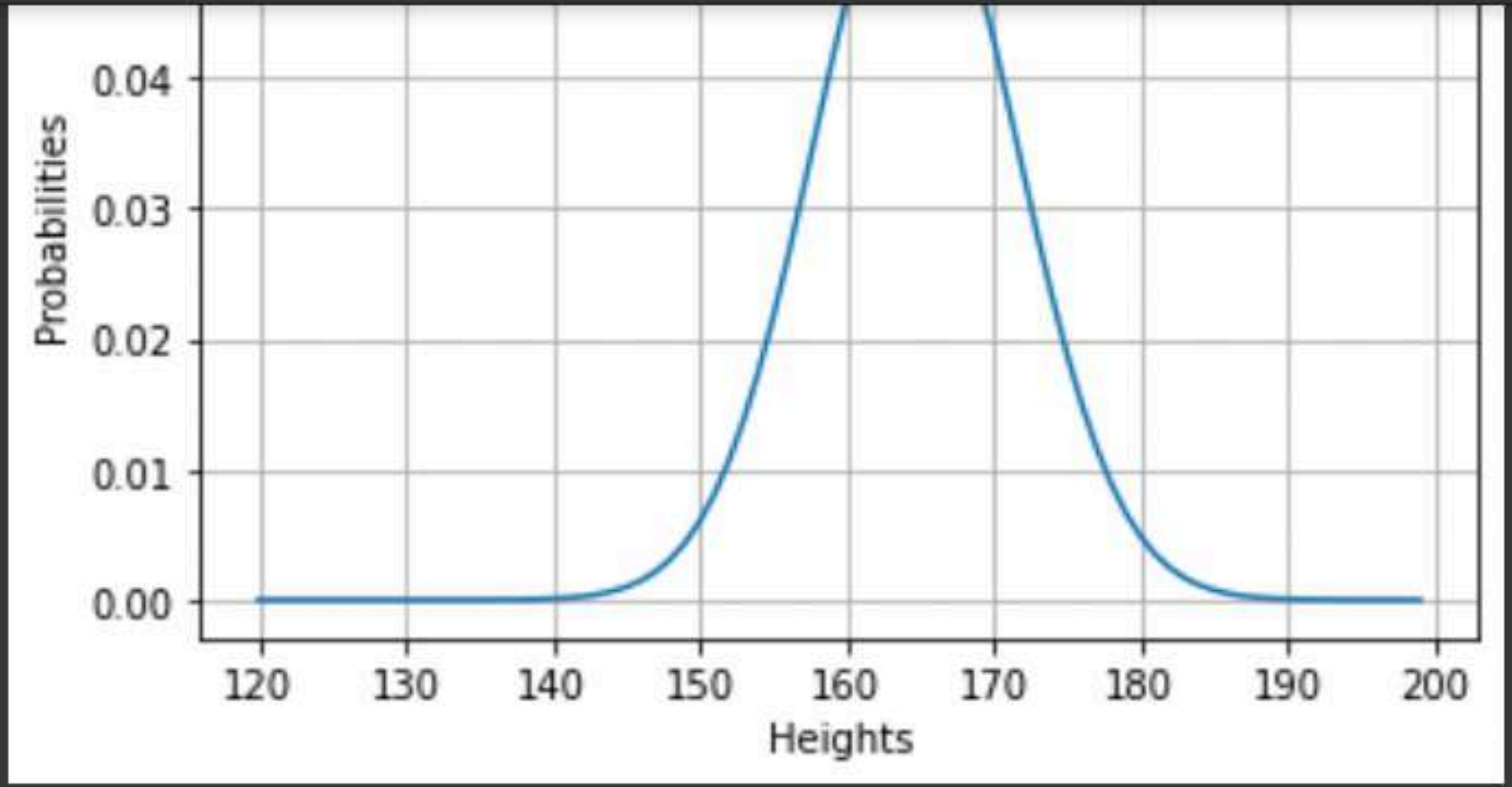
180+ : XL

120

200+ : XXL

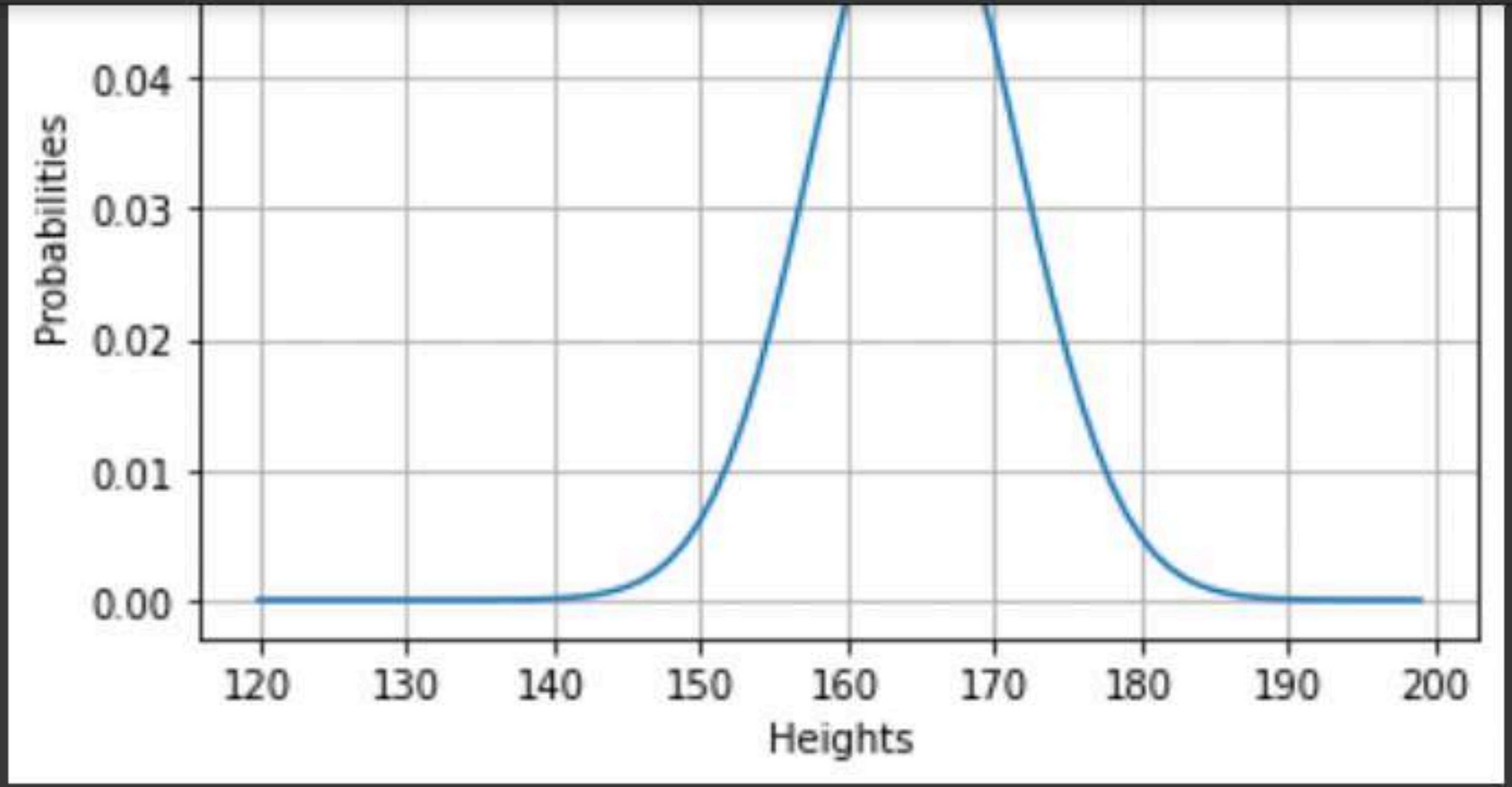
XXS : 166

[32]




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

```
0.016573163101179463
```



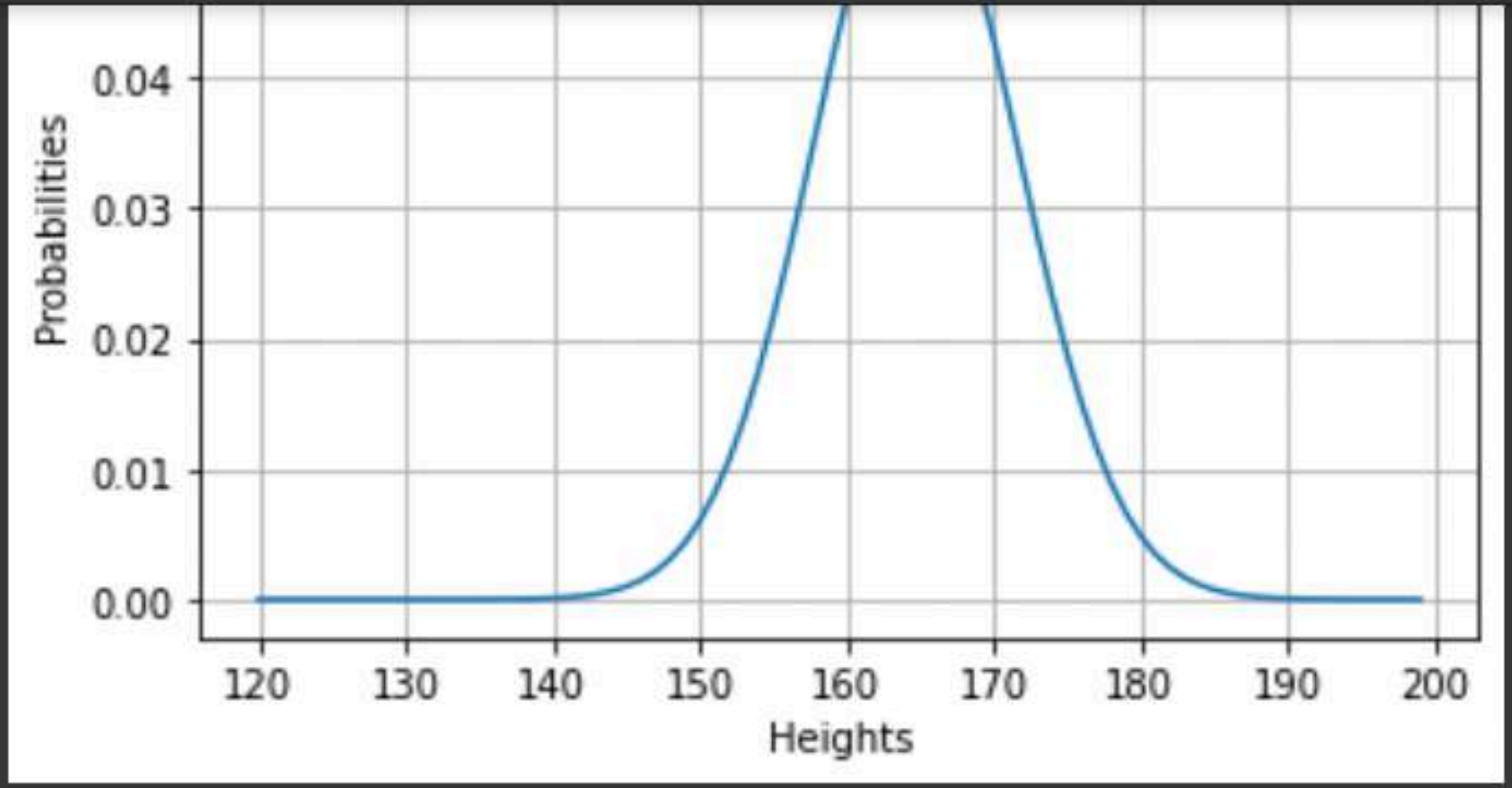
$P(H > 200)$
$= 1 - P(H \leq 200)$

[33]
```python
height_dist.cdf(x = 150)    # P(X<=150)
```

0.016573163101179463

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

0.9999998541520864



Data → heights

↓ estimate

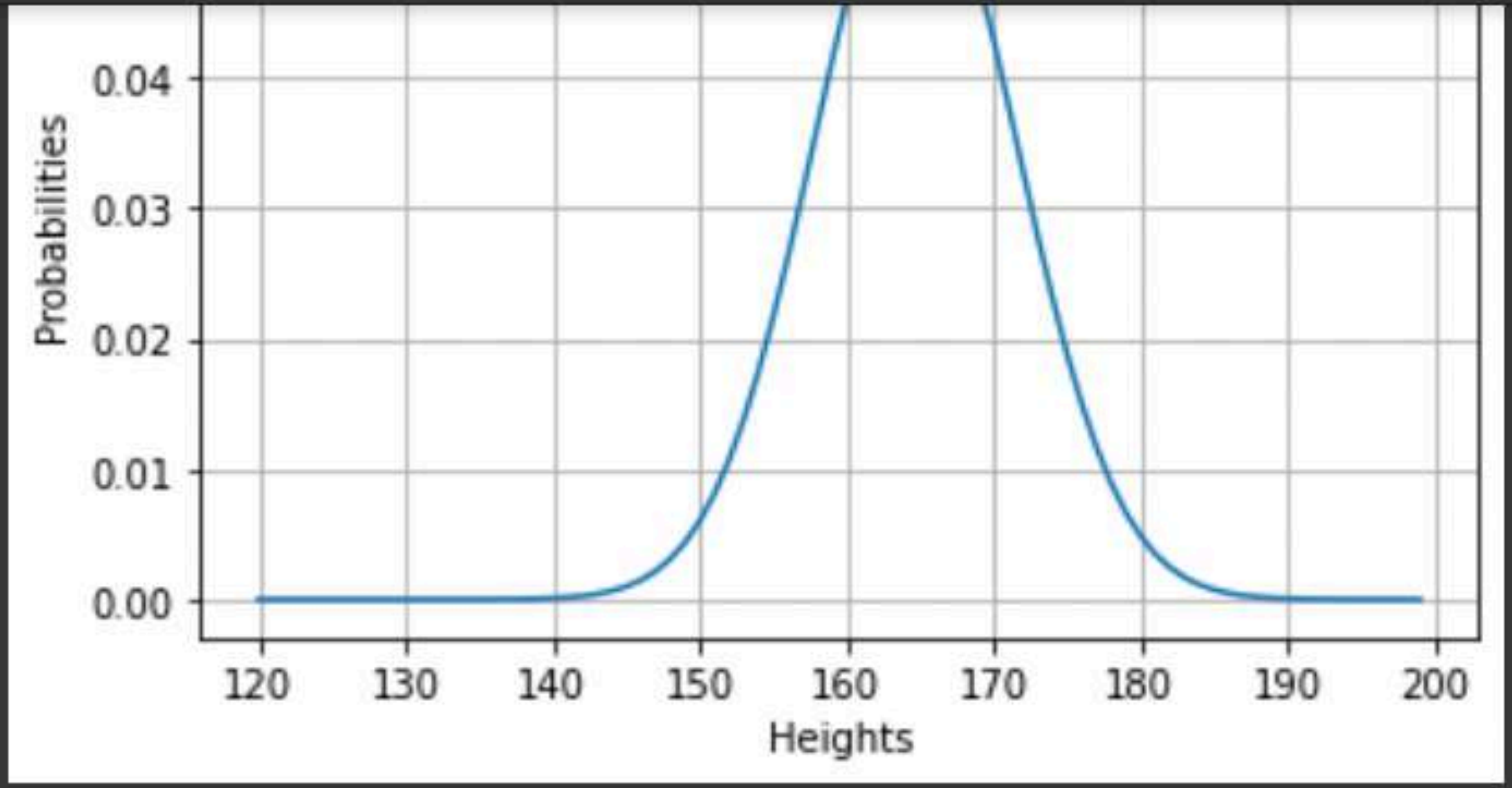
$N(\mu, \sigma)$

↓

CDF → 150-160

[33]
```
height_dist.cdf(x = 150)    # P(X<=150)
```
```
0.016573163101179463
```

[34]
```
height_dist.cdf(x = 200)    # P(X<=200)
```
```
0.9999998541520864
```




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

0.016573163101179463

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

0.9999998541520864

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

0.016573163101179463

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

0.9999998541520864

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

```
round((below_160 - below_150)*10000)
```

2322

XS

Q

150-160 → 2322

why round?

Q

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

0.016573163101179463

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

0.9999998541520864

round ?

ceil ?

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

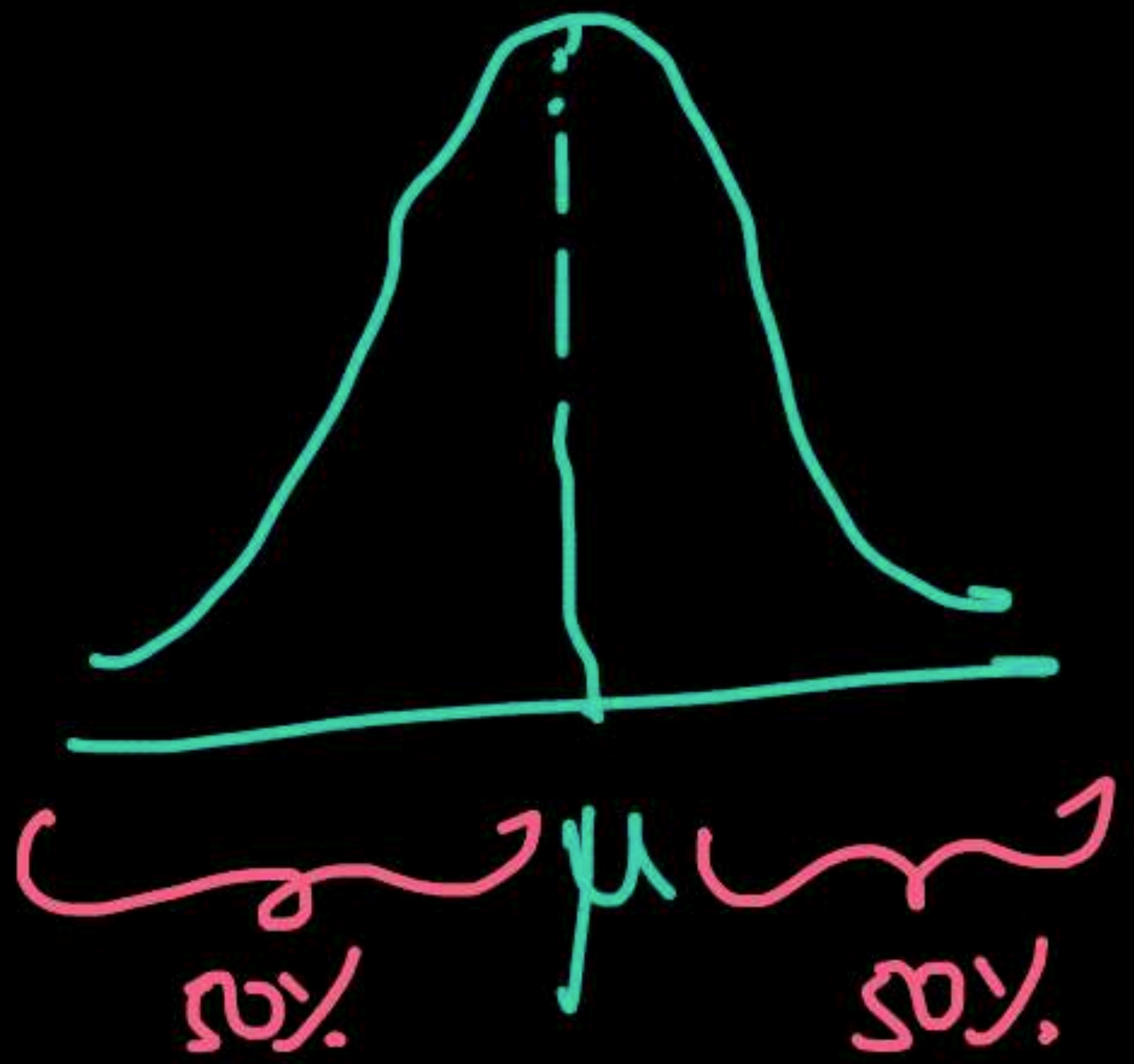
```
((below_160 - below_150)*10000)
```

2321.5478780907047

Q

X → $\boxed{\text{Mean} = \text{Median}}$ $\not\Rightarrow$ X ~ Normal

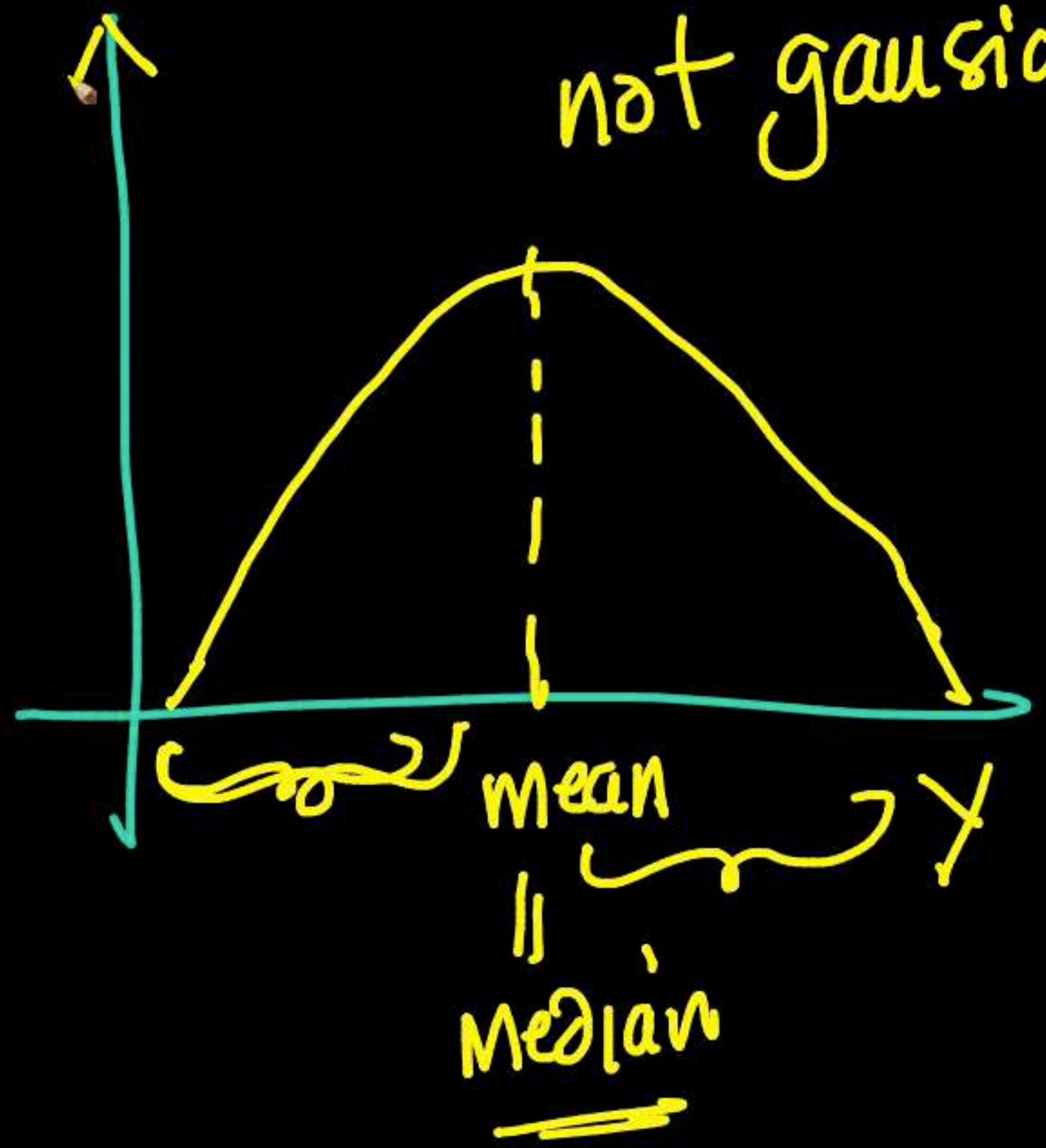
Can we conclude X ~ Normal?

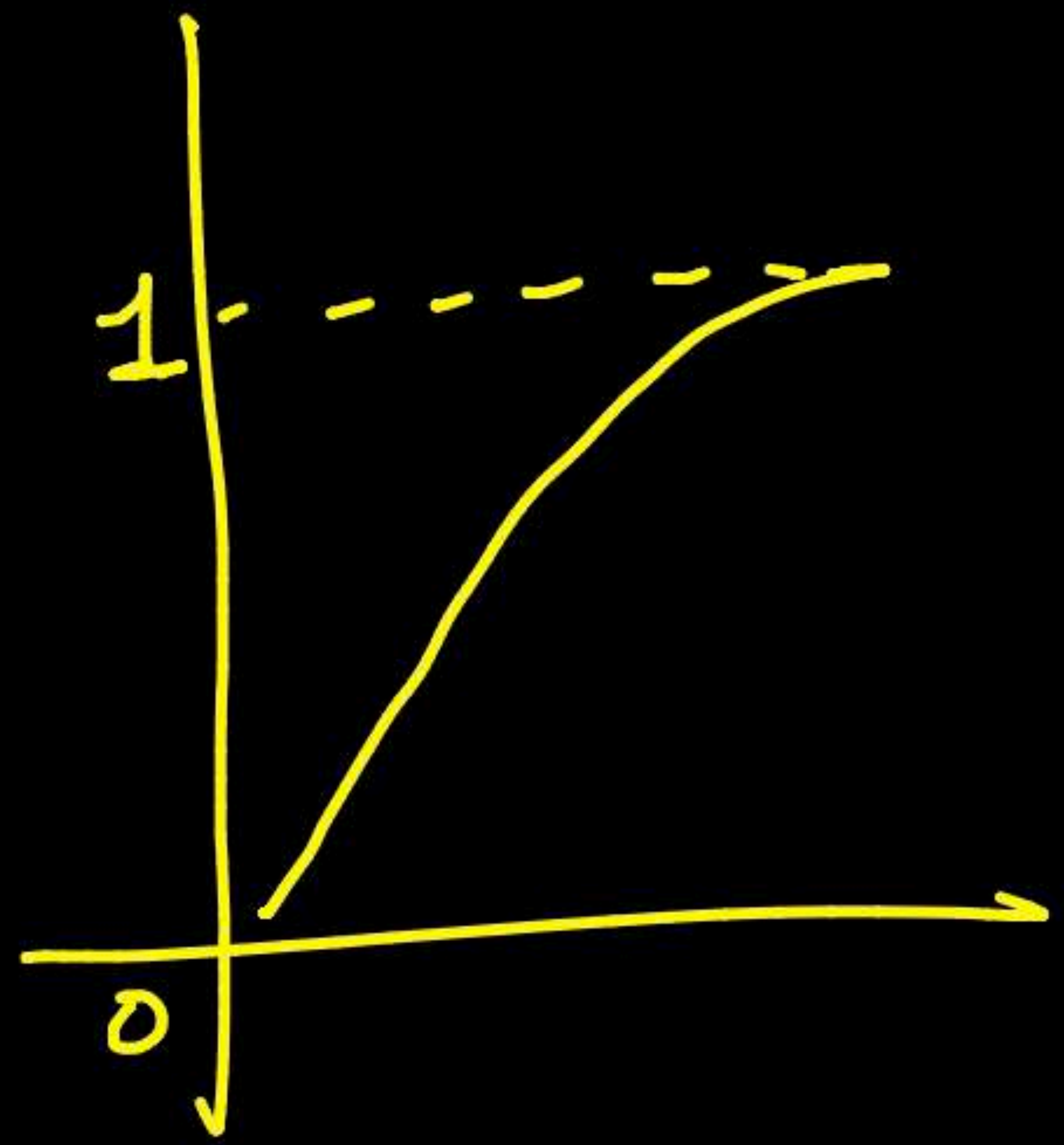


for Gaussian distb r.v

μ = Median

not gausian





statistical test (KS, AD)

If $X \sim$ Normal then $\mu_x = \text{Median}(x)$

If X is s.t. $\mu_x = \text{median}(x)$ then it may/may not be Normally distb

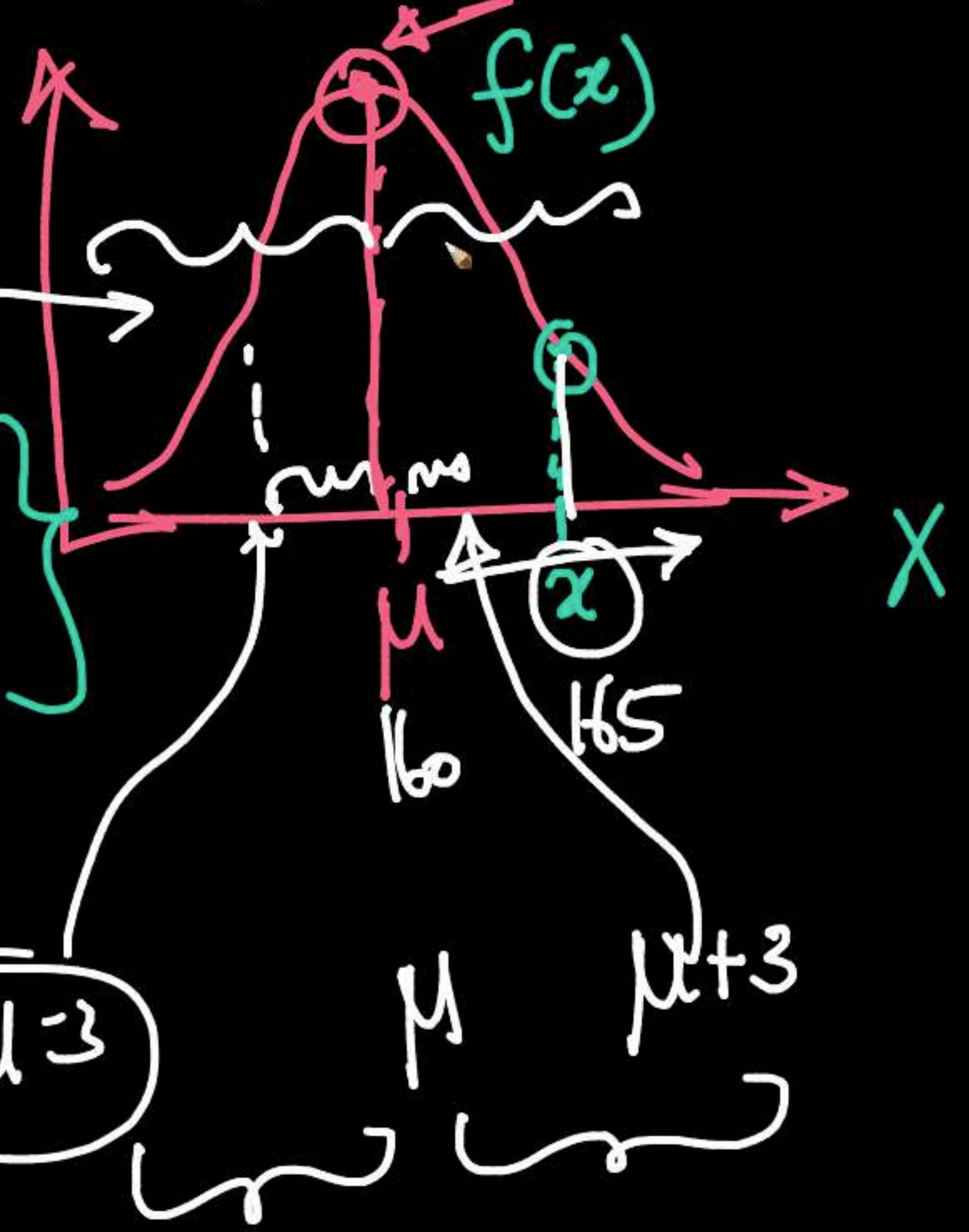
Math

$$e^x = \exp(x) = \exp\{x\}$$

$$X \sim N(\overset{160}{\mu}, \overset{10}{\sigma})$$

PDF

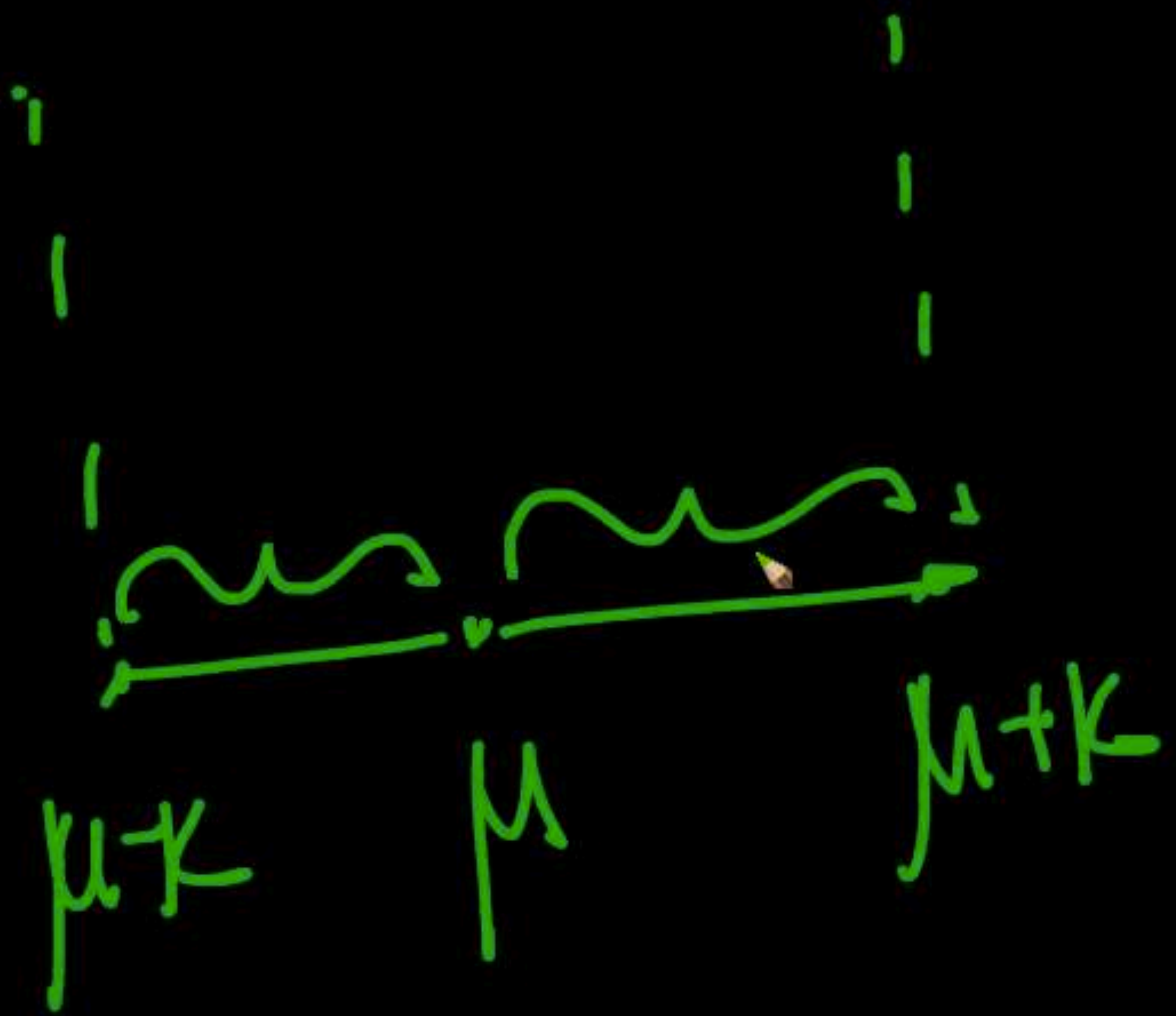
$$f(x) = \boxed{\frac{1}{\sqrt{2\pi\sigma^2}}}\, \exp\left\{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2\right\}$$



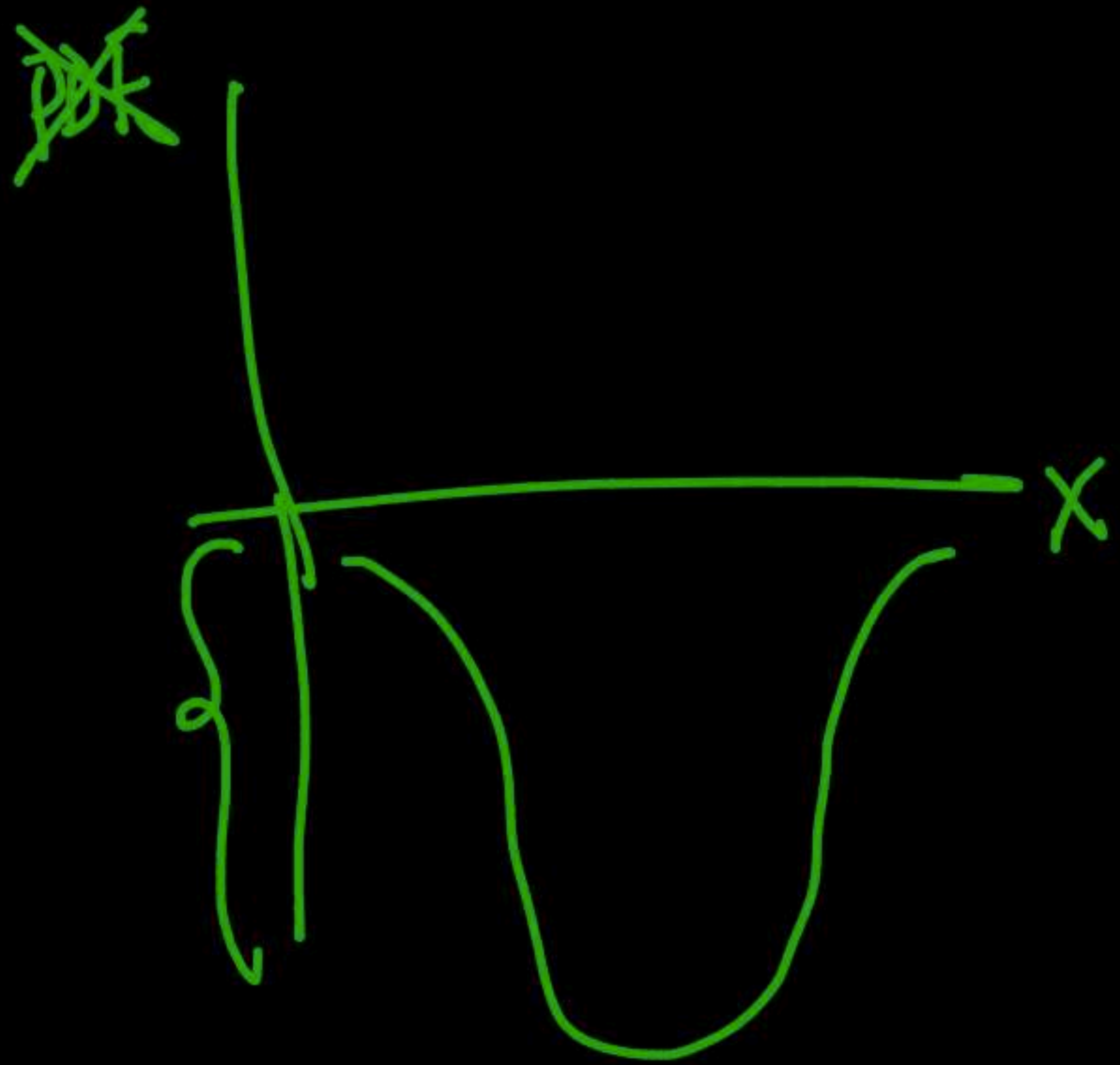
11th & 12th {Maxima & Minima (Calc)

$\mu+k$ $\mu-k$

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

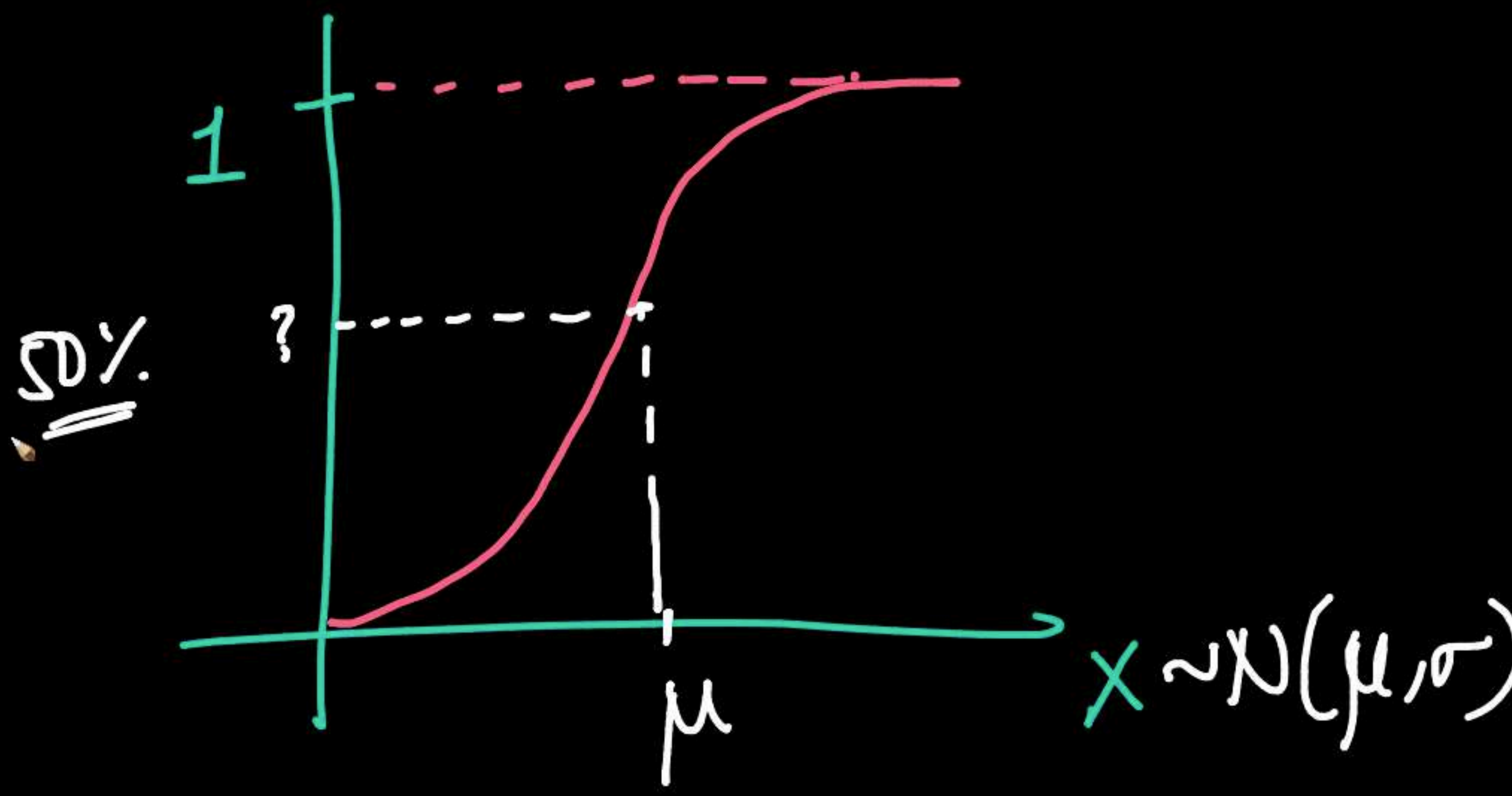


$-f(x)$



Mirror image

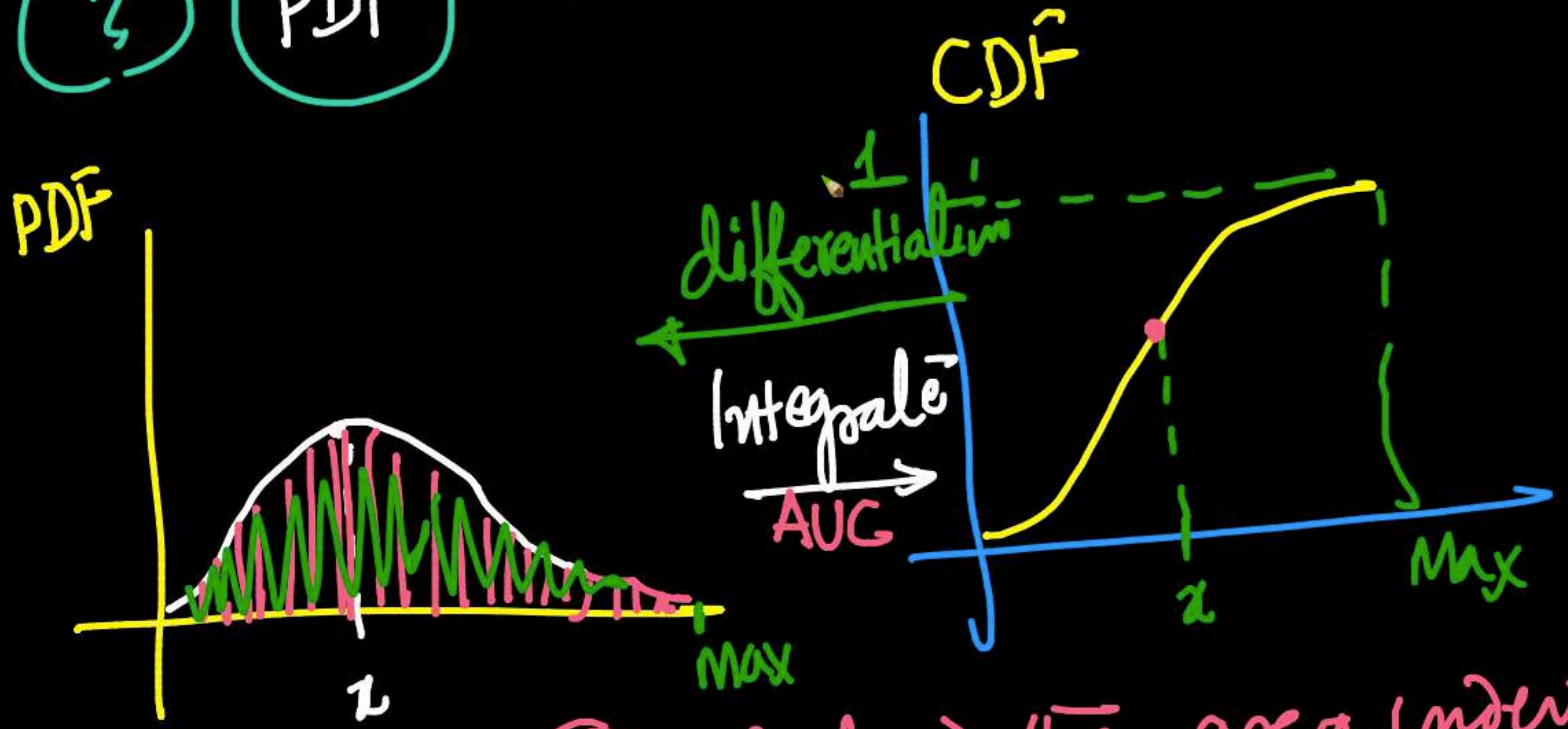
CDF of Normal



PMF → CDF

cumulative addition

(?) PDF → CDF



Q What is the area under the whole PDF-curve $f(x)$

$$\begin{cases} f(x) = \text{PDF} \longrightarrow \text{area under curve} = 1 \\ F(x) = \text{CDF} \end{cases}$$

$$f(x) = \frac{dF(x)}{dx}$$

$$\int_{-\infty}^{x} f(x) = F(x)$$

ignore this if it too overwhelming

revisit it in calc

Camera Settings 23:03

Normal: → property: $\boxed{\text{68-95-99 rule}}$

68.27%

$$X \sim N(\overset{160,\ 10}{\mu, \sigma})$$

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

150 170

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

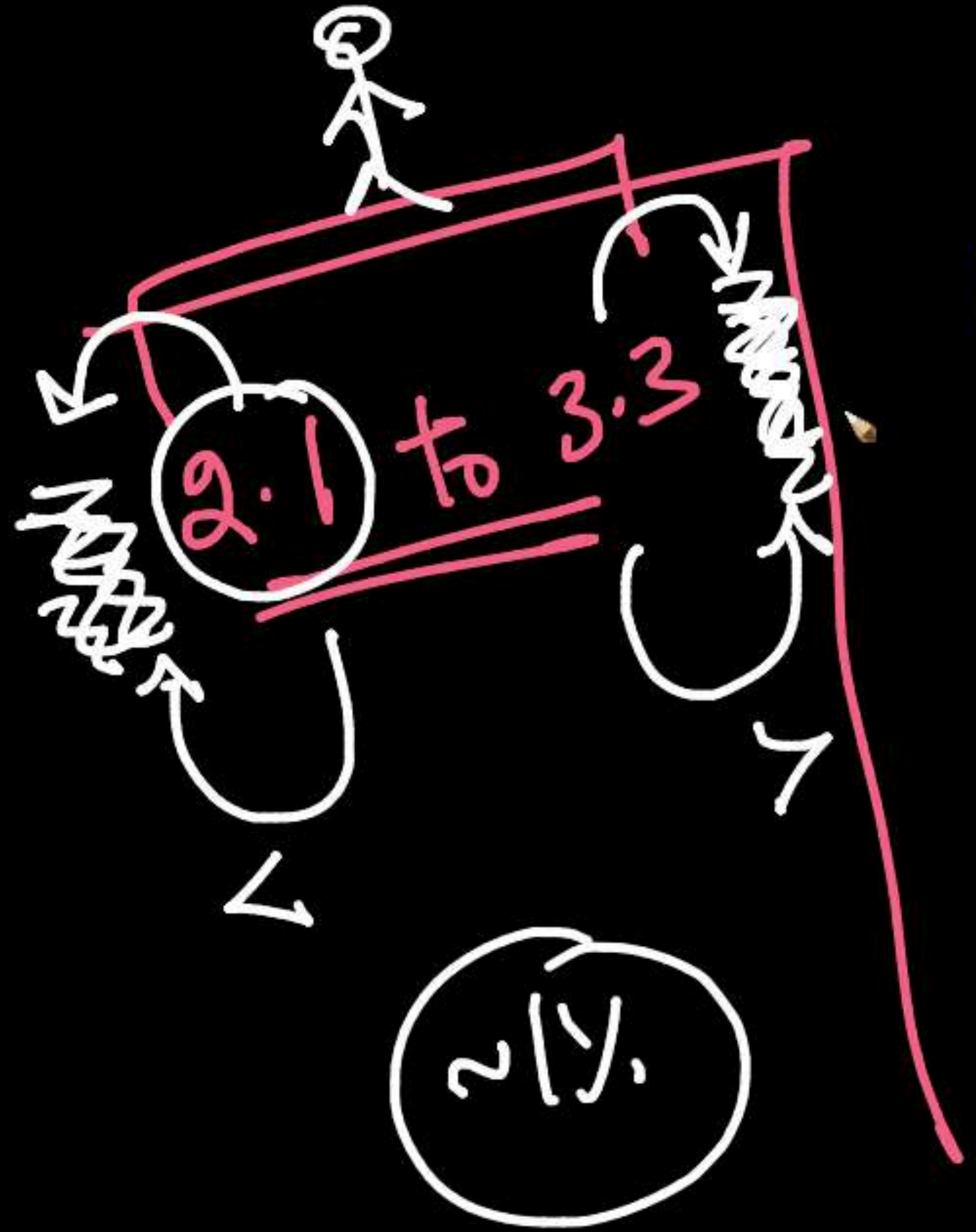
140 180

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

Q

birth weights: $\mu = 2.7$ kgs, $\sigma = 0.2$ kgs ②

Normal ①



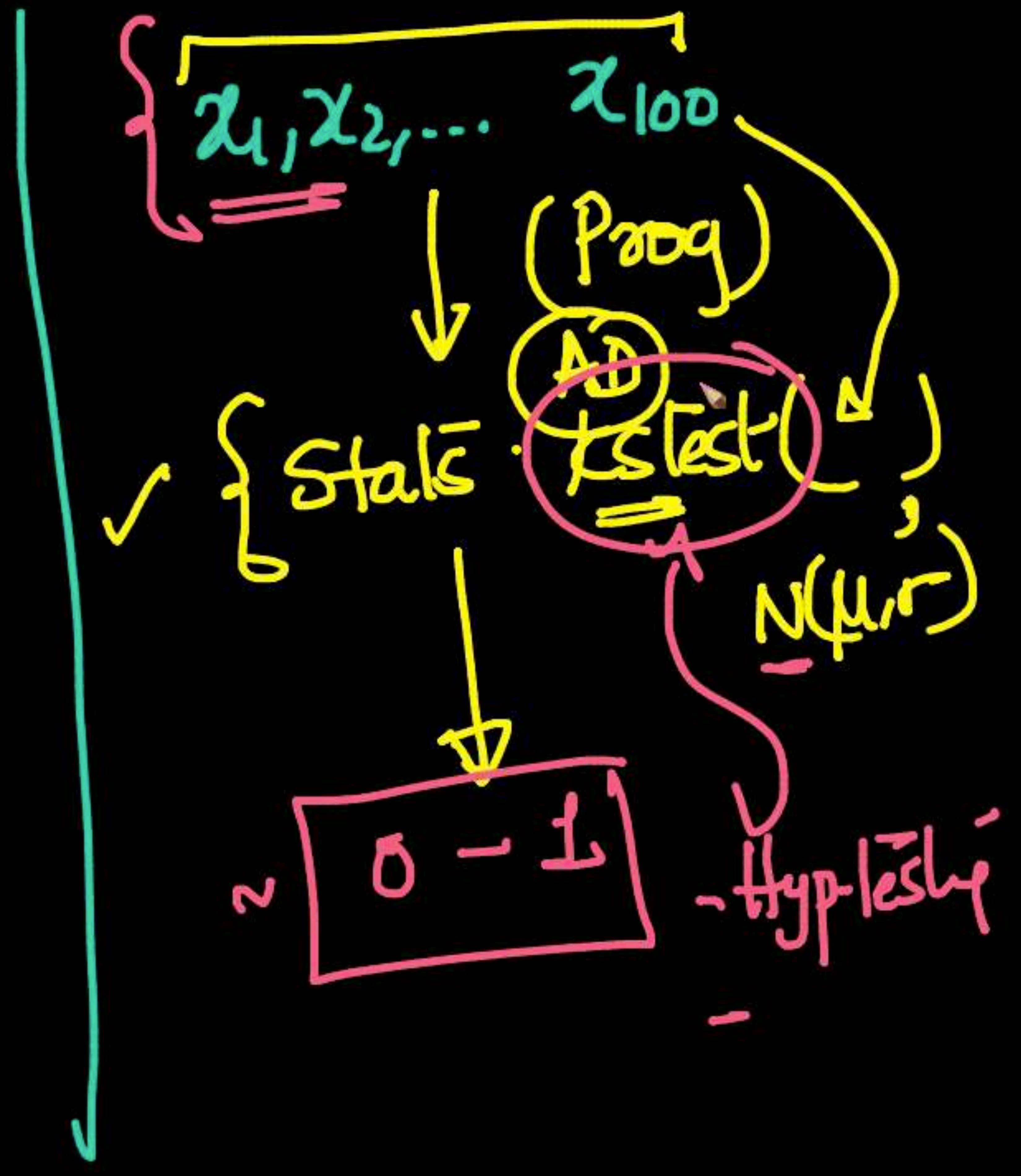
What %age of newborns will have a wght (2.5 to 2.9 kgs)

$(\mu - \sigma \leftrightarrow \mu + \sigma)$

~68%

Camera Settings 23:10

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



68 / 68

Camera Settings 23:18

Standard normal Variate

Transformation

for each

$X \sim N(\mu, \sigma)$ with $\mu = 100$, $\sigma = 5$

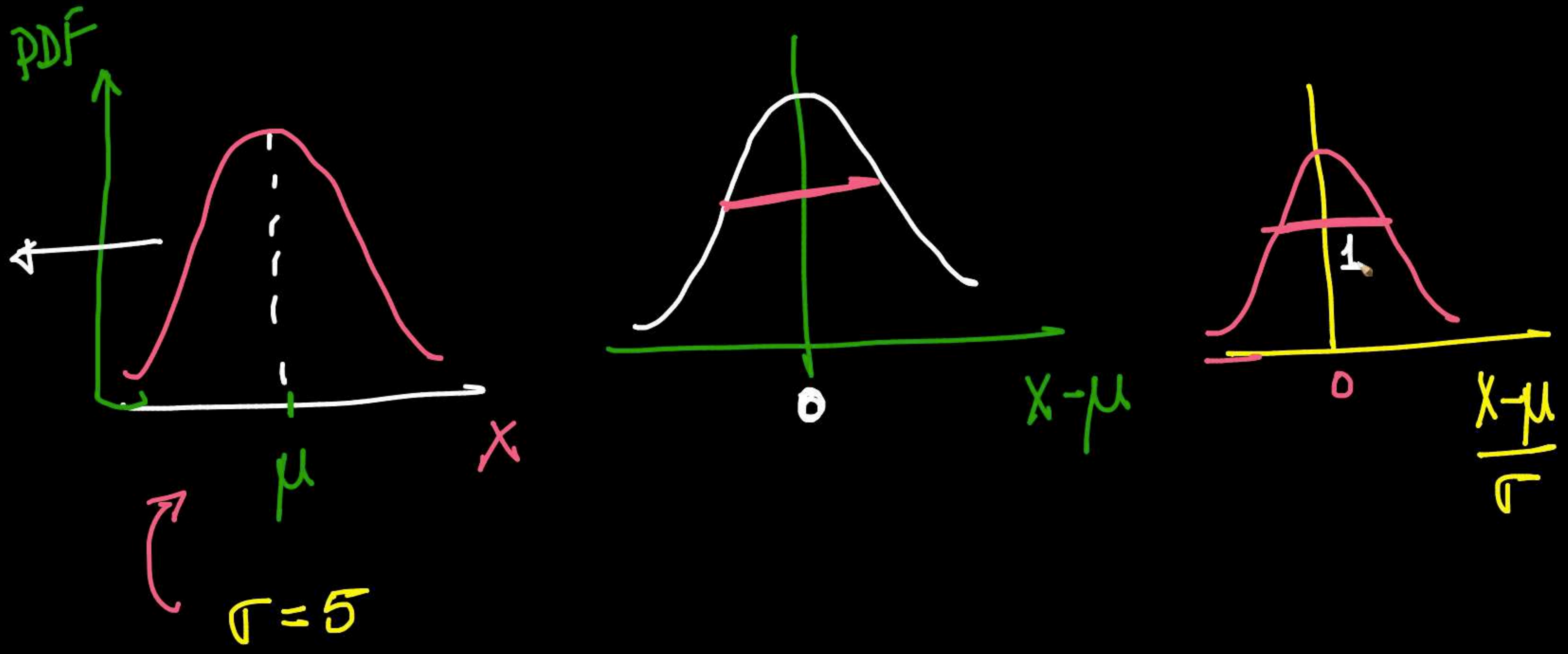
$x_1, x_2, \ldots x_i, \ldots x_n$

$$z_i = \frac{x_i - \mu}{\sigma} \quad \forall i$$

$Z \sim N(0, 1)$

69 / 72

Camera Settings 23:16



$$\frac{1}{n}\sum_{i=1}^{n}\frac{x_i-\mu}{\sigma} = \frac{\frac{1}{n}\sum x_i - \mu}{\sigma} = \frac{0}{\sigma} = 0$$

Std-dev of $\frac{x_i-\mu}{\sigma}$'s $\rightarrow$ 1 cm

$h_1, h_2, \dots h_{10}$

Standardization → $N(0,1)$

$H \sim N(\mu = 160, \sigma = 10)$

$\frac{h_i - 160}{10} = Z_i$

$B \sim N(\mu = 2.7, \sigma = 0.2)$

$\frac{b_i - 2.7}{0.2} = Y_i$

kgs

$b_1, b_2, b_3 \dots b_{50}$

$N(0,1)$

Def: $\boxed{SNV = N(0,1)}$

Math-proofs

ML & DL → preprocessing

↓

(Standardization)

Questions

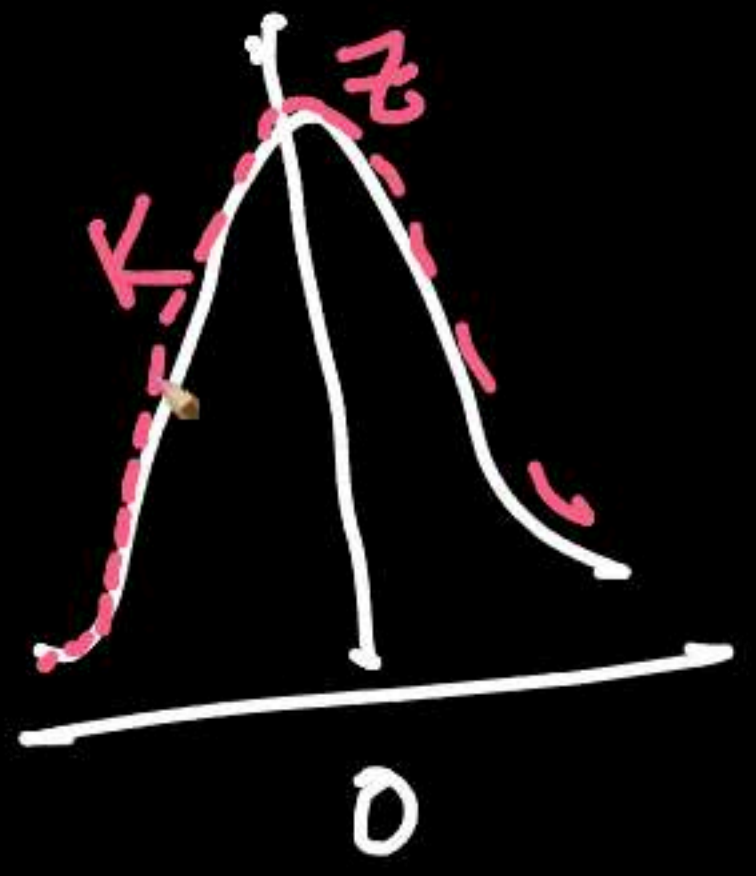
- ✓ Poisson disb → # visitors to your website in 1hr
- ✓ Expo disb → disb of Δt
- ✓ Geometric →
- ✓ Log-Normal → Sales prices @ Amazon, del. time of Zomato

CLT; Hyp-testing

QQ-plot

$$X \sim N(\mu_1, \sigma_1) \xrightarrow{\text{standardize}} Z \sim N(0,1)$$

$$Y \sim N(\mu_2, \sigma_2) \longrightarrow K \sim N(0,1)$$



Camera Settings 23:32

Baby:

$H \sim N(\mu_1, \sigma_1) \longrightarrow H' \sim N(0,1)$ μ

$W \sim N(\mu_2, \sigma_2) \longrightarrow W' \sim N(0,1)$ σ

no-scale/units

30 cm, 1.6 kgs

h' w'

baby: $(0, -1.6)$ → height ~ avg baby

weight is 1.6 std-dev lower than avg baby

$h', w' = (-1.2, -1.8) \longrightarrow$

76 / 76

newborn

(Q1) $h', w' = (\overset{ht}{-1.2}, -1.8) \rightarrow \mu = 0, \sigma = 1$

95% of babies $(-2, 2)$

(Q2) $h', w' = (-2.1, -3)$

premature

(Q3) $h', w' = (2, 3) \rightarrow \ldots$

$x_1, x_2 \dots x_n$

$X \sim N(\mu, \sigma) \xrightarrow[\text{standardize}]{\frac{x_i-\mu}{\sigma}} X' \sim N(0,1)$ ✓

(Q)

$Y \sim Bin(n, p)$

$y_1, y_2 \dots y_m$

$\hookrightarrow \mu_Y - \sigma_Y$

$\xrightarrow{\text{standarize}} \dfrac{y_i - \mu_Y}{\sigma_Y}$

$Y' \not\sim N(0,1)$

{Yes/No}

$W \sim N(\mu, \sigma) \xrightarrow{\text{std}} W' \sim N(0,1)$

BP $\not\sim$ Normally $\xrightarrow{\text{std}}$

Some Disb

$\boxed{\text{Normal}}$ (log; box-cox)

$BP' \rightarrow$ mean $= 0$, std $= 1$

Disb $\neq$ Normal