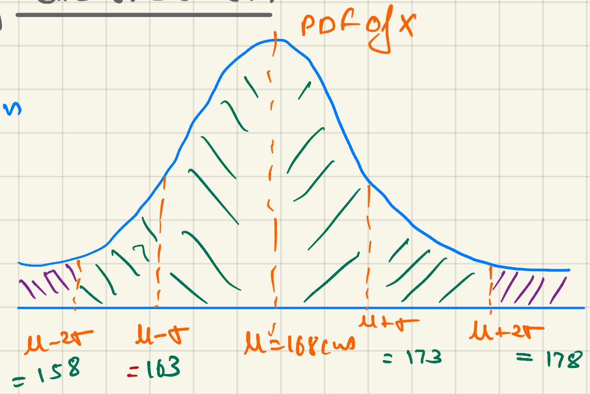
Computing C.I given the underlying distribution

$\Rightarrow$ heights $\leftarrow X \sim N(\mu, \sigma)$ $\longrightarrow$ 'x' follow Gaussian

Let $\begin{pmatrix} \mu = 168 \text{ cm} \\ \sigma = 5\text{cm} \end{pmatrix}$



$\rightarrow (\mu - 2\sigma, \ \mu + 2\sigma)$

95% of my observations

$[158, 178]$ with 95% prob.

$\rightarrow$ If you know the underlying disb^n (like in above case gaussian) we can compute C.I.

C.I for mean (μ) of a r.v

$X \sim F$ with pop-mean of 'μ' & std-dev. of 'σ'.

$\{x_1, x_2, \ldots\ldots x_{10}\} \rightarrow$ Sample of size $= n = 10$

$\{180, 162, 158, 172, 168, 150, 171, 183, 165, 176\}$

Q) what is the 95% C.I of μ?

$\rightarrow$ case ① $\sigma = 5\text{cm}$ {we know pop. std dev} $\leftarrow$ <u>we assume that</u>

CLT:- $\bar{x}$ = sample mean $= \frac{1}{10}\sum_{i=1}^{10} x_i$

$\bar{x} = 168.5\text{cm}$

$\bar{x} \sim N(\mu, \frac{\sigma}{\sqrt{n}}) \longrightarrow$ CLT

(N → Gaussian; μ → pop-mean; $\frac{\sigma}{\sqrt{n}}$ → $\frac{\text{pop-std-dev}}{\sqrt{n}}$)

$\sqrt{n} = \sqrt{10}$

$\mu \in \left[\bar{x} - \frac{2\sigma}{\sqrt{n}}, \ \bar{x} + \frac{2\sigma}{\sqrt{n}}\right]$ with 95% confidence

$\mu \in [165.34, 171.66]$ " " "

Case ② if we do not know σ (pop-std dev)

sample → n

Here, we use 't-disbⁿ' (students' t-disbⁿ)

$$\boxed{\bar{x} \sim t(n-1)}$$

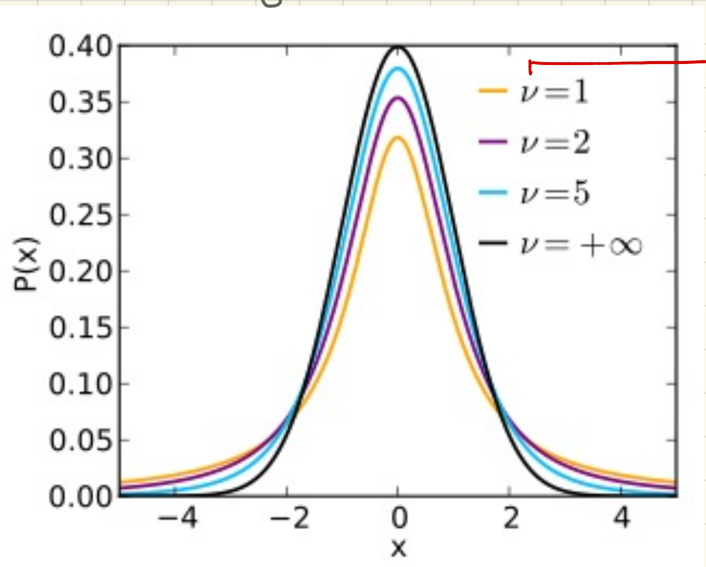
→ Your sample mean ($\bar{x}$) follow 't-disbⁿ' with (n−1) degrees of freedom.

Sample mean

degrees of freedom

→ One of the biggest application of "students' t-disbⁿ" is to calculating C.I of mean of a r.v when 'σ' is unknown.

Pdf of "t-disbⁿ"



degrees of freedom

as "ν ↑" our "peakdness" also incⁿ.

→ estimate C.I of μ of a r.v

→ case 1 : σ is known CLT ; $N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$

→ case 2 : σ is unknown t-disbⁿ (n−1)

Q) How to estimate C.I for 'σ' of a r.v ?

→

Statistics

median

90th percentile

How to find C.I for these properties?

Single-Sample Confidence Interval Calculator

This simple confidence interval calculator uses a *t* statistic and sample mean (*M*) to generate an interval estimate of a population mean (μ).

The formula for estimation is:

$$\mu = M \pm t(s_M)$$

where:

M = sample mean
t = t statistic determined by confidence level
s_M = standard error = $\sqrt{(s^2/n)}$

As you can see, to perform this calculation you need to know your sample mean, the number of items in your sample, and your sample's standard deviation. (If you need to calculate mean and standard deviation from a set of raw scores, you can do so using our descriptive statistics tools.)

The Calculation

Please enter your data into the fields below, select a confidence level (the calculator defaults to 95%), and then hit Calculate. Your result will appear at the bottom of the page.

Sample Mean (*M*):	4.4
Sample Size (*n*):	30
Standard Deviation (*s*):	0.29
Confidence Level:	95%

Result

M = 4.4, 95% CI [4.292, 4.508].

You can be 95% confident that the population mean (μ) falls between 4.292 and 4.508.

Calculate

Calculation

$M = 4.4$
$t = 2.05$
$s_M = \sqrt{(0.29^2/30)} = 0.05$

$\mu = M \pm t(s_M)$
$\mu = 4.4 \pm 2.05*0.05$
$\mu = 4.4 \pm 0.108$

The central limit theorem (CLT) is a theoretical framework in probability and statistics that describes the distribution of sample means from a population. Bootstrapping is a resampling approach that involves creating new datasets from observed data to provide an empirical estimate of a statistic's distribution. Bootstrapping is non-parametric and makes minimal assumptions about the underlying distribution, while the CLT assumes a finite mean and variance in the population.

The CLT states that when independent random variables are added, their normalized sum tends toward a normal distribution, even if the original variables are not normally distributed. The CLT applies to almost all types of probability distributions, but the population must have a finite variance. The CLT is important in statistics for its normality assumption and the precision of its estimates.

Bootstrapping is a hands-on approach to learning the CLT and is often used instead of statistical inference. Bootstrapping uses the original sample as the population from which to resample.

C.I using empirical bootstrap :

bootstrap → computational method

→ C.I for median, var, std-dev, 90^{th} percentile.

→ Computers → programming & simulation

programming → 60 - 70 yrs

Q) $X \sim F$ task : estimate 95% C.I for median of 'X'.

→ S : sample of size 'n' : $\{x_1, x_2, \ldots\ldots x_n\}$ $[n=10]$

using only sample → C.I of median of 'x'

$U(1,n)$ $n=10$

→ generating discrete uniform r.v. → $x_1^{(1)}$, $x_2^{(1)}$

$S_1 = \underbrace{x_1^{(1)}, x_2^{(1)}, x_3^{(1)} \ldots\ldots x_m^{(1)}}$ such that $m \leq n$

→ random sample of size 'm' generated from S.

→ Sampling with repetition. (some value may repeat during generating this 'S_1'.)

$S = \{x_1, x_2, \ldots\ldots\ldots x_n\}$

↓ using sampling with replacement (single element can occur multiple times) (using $U(1,n)$)

bootstrap sample

$S_1 : x_1^{(1)}, x_2^{(1)}, \ldots\ldots\ldots x_m^{(1)} \longrightarrow m_1$

$S_2 : x_1^{(2)}, x_2^{(2)}, \ldots\ldots\ldots x_m^{(2)} \longrightarrow m_2$ = median of sample 2

$\vdots$

$S_K : x_1^{(K)}, x_2^{(K)}, \ldots\ldots\ldots x_m^{(K)} \longrightarrow m_K$

→ assume $K = 1000$

→ $m_1, m_2 \ldots\ldots\ldots m_{1000}$ ← 1000 medians gen. using bootstrap - sample.

↓ sort

$m_1' \leq m_2' \leq \ldots\ldots\ldots \leq m_{1000}'$ (increasing order)

C.I (95%)

$$\underset{(25\ \text{values})}{\underline{\underline{25}}} \quad m'_{25} \quad \underset{950\ (\text{value})}{\underline{\qquad\qquad}} \quad m'_{975}\ (25)$$

$$\frac{950}{1000} = 95\%$$

→ 95% C.I of median x is [m'_{25} , m'_{975}]

→ If you want C.I for variance or std. dev. you can compute it from above method.

→ This is a "$\underbrace{\text{non-parametric}}$" technique.

↳ not make any assumptions about the disb^n of data.

```python
import numpy
from pandas import read_csv
from sklearn.utils import resample
from sklearn.metrics import accuracy_score
from matplotlib import pyplot

# load dataset
x = numpy.array([180,162,158,172,168,150,171,183,165,176])
# configure bootstrap
n_iterations = 1000
n_size = int(len(x))

# run bootstrap
medians = list()
for i in range(n_iterations):
    # prepare train and test sets
    s = resample(x, n_samples=n_size);
    m = numpy.median(s);
    #print(m)
    medians.append(m)

# plot scores
pyplot.hist(medians)
pyplot.show()

# confidence intervals
alpha = 0.95
p = ((1.0-alpha)/2.0) * 100
lower =  numpy.percentile(medians, p)

p = (alpha+((1.0-alpha)/2.0)) * 100
upper =  numpy.percentile(medians, p)
print('%.1f confidence interval %.1f and %.1f' % (alpha*100, lower, upper))
```

→ sample (S)
↳ n = 10

n_iterations → K

n_size → m

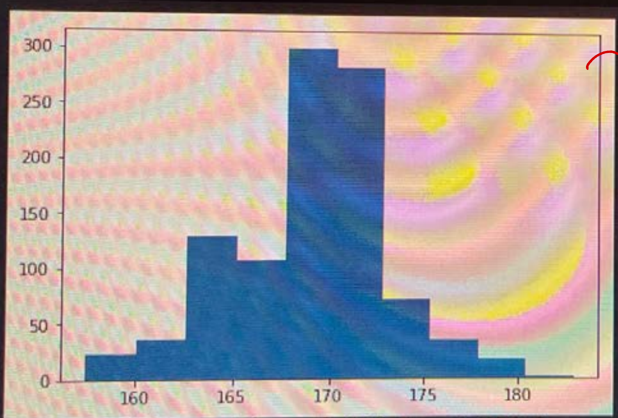
s → s_i

n_samples=n_size → m = n

→ 1000 bootstrap sample and finding medians ($m_1, m_2, \ldots\ldots m_{1000}$)

→ 25%

→ 975%



95.0 confidence interval 161.5 and 176.0

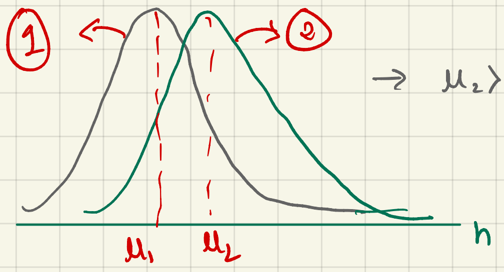
→ Not gaussian

Hypothesis testing

Q) Is there a diff. in heights of students in cl 1 & cl 2? (I want to be very sure about my ans. like cl1 student typically taller than cl2.)

	cl1
1	160
2	152
⋮	⋮
50	148

	cl2
1	162
⋮	56
⋮	⋮
50	182



→ $\mu_2 > \mu_1$ (how much are you sure about that, the gap is very small).

① choosing a test-statistics (x)
↳ $(\mu_2 - \mu_1)$

μ_2 = mean height of cl2 students

μ_1 = " " " cl1 "

② Null hypothesis :- (H_0) (PROOF OF CONTRADICTION)

H_0 : no-difference in μ_1 & μ_2

Alternative hypothesis :- (H_1) (Inverse of Null hypothesis)

H_1 : diff. in μ_1 & μ_2

③ p-value :- (what is the prob. of observing a value of 'x' $(\mu_1 - \mu_2)$ if my null hypothesis (H_0) is true)

eg :- if my p-value is 0.9 [It means a prob of observing diff.ⁿ of $(\mu_1 - \mu_2)$ is 0.9 if H_0 is true.]

(high)
(accept H_0)

if p-value is 0.05 ↳ 5% chance (prob.) that $(\mu_1 - \mu_2)$ if H_0 is true.

reject H_0 accept H_1

Hypothesis testing with coin toss example

example 1: Task → Given a coin, determine if the coin is biased towards heads or not.

biased towards heads :- $P(H) > 0.5$

not - " " " :- $P(H) = 0.5$

→ expt. (design a expt.) :- flip a coin 5 times and count # heads $= \underbrace{X}_{r.v}$ → Test-statistic

(flip coin)

→ perform expt.:→ f, f, f, f, f

$\downarrow$ H, H, H, H, H

$X = 5$ ← observation by experiment

$$p(\underbrace{X=5}_{\text{obs.}} \mid \underbrace{\text{coin is not biased towards heads}}_{\text{Null-hypothesis } (H_0)}) = p(\text{obs} \mid H_0)$$

H_0 : coin is not biased towards heads

$$p(X=5 / H_0) = f, f, f, f, f = \frac{1}{2^5} = \frac{1}{32} = 0.03 = 3\%$$

$p(X=5/H_0)$ → five heads in five tosses; H_0 → coin is not biased toward heads, $P(H) = \frac{1}{2} = 0.5$

$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}$

H H H H H

H H H H T

⋮

T T T T T

→ 32

$p(X=5 \mid H_0) = 3\%$ → There is a 3% chance of getting 5 heads in 5 flips if the coin is not biased towards heads.

hyp-testing

$$p(\text{obs by expt} \mid \underbrace{\text{assumption}}_{H_0}) = 3\%$$

conditional prob.

↳ small

P-value → $p(\text{obs} \mid \underset{H_0}{\text{assumption}})$

$< 5\%$ → rule of thumb

$p(\text{obs}|H_0) < 5\%$ then H_0 may be incorrect

⇓

assumption or Ho is not true

⇓

reject Ho ⇒ reject the idea coin is not biased towards heads.

Null-hyp :- H_0 : Coin is not biased towards heads

Alternative hyp- H_1 : Coin is biased towards heads

→ expt :- → (5) times
→ (3) times → my output may be diff. } → sample size
→ (10) times → " " " " "

So, expt. depended on How many flips I have done.

expt 1 :- flip the coin three times & count the no. of heads. = X

↑ test-statistic

perform expt :- f , f , f
↓ H ↓ H ↓ H

$X = 3$ ⟵ observation

$p(\text{obs} | \text{assumption } H_0) = \frac{1}{2^3} = \frac{1}{8} = 12.5\% > 5\%$

accept H_0

↳ coin is not biased towards heads.

⇒ So, here by changing my 5 flips to 3 flips my output is changed.

Careful sample-size

hyp-testing :- (1) design of the expt :

(2) Ho : Null-hyp

$P(\text{obs}|H_0)$ → easy, & feasible to solve

(3) design X

p-value = 3% $\longrightarrow$ reject H_0

$\hookrightarrow$ $p(\text{obs} \mid H_0)$ $\rightarrow$ prob. of 'obs' given the H_0 is true

$p(H_0 \text{ is true})$ $\longrightarrow$ X you can't say that.

Resampling and Permutation test :- [for calculating p-value]

	cl1		cl2
1	160	1	162
2	152		56
⋮	⋮	⋮	⋮
50	148	50	182
	μ_1		μ_2

$\longrightarrow$ $\Delta = \mu_1 - \mu_2$

Jumble $\rightarrow$ by Jumbling I'm assuming H_0

(100 students)

randomly sample

① x [1 … 50] y [1 … 50] $\mu_1 - \mu_2 \longrightarrow \delta_1$

② x [ ] y [ ] $\mu_1 - \mu_2 \longrightarrow \delta_2$

⋮

(10K) x [ ] y [ ] $\mu_1 - \mu_2 \longrightarrow \delta_{10K}$

Sorting my 'δ'

$\delta_1' \; \delta_2' \; \delta_3' \; ---- \; \Delta \; ---- \; \delta_{10K}'$

sorted (incⁿ) if 5% pt.s here

$\hookrightarrow$ p-value = 0.05

$\hookrightarrow$ p - value = x

$\boxed{\text{p-value} = 5\%}$

K-S Test for Similarity of two distributions

$X_1 : [x_1', x_2', \ldots\ldots x_n']$ n-term

X_2 : m-terms

Q) Is the X_1 and X_2 have same distribution or not?

$X_1 \rightarrow n$
$X_2 \rightarrow m$

H_0:- X_1 & X_2 have same disb^n.

Two-sample Kolmogorov–Smirnov test [edit]

The Kolmogorov–Smirnov test may also be used to test whether two underlying one-dimensional probability distributions differ. In this case, the Kolmogorov–Smirnov statistic is

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|,$$

$F_{2,m}(x) \rightarrow$ CDF of X_2 (m-term)

$F_{1,n} \rightarrow$ CDF of X_1 (n-term)

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of the first and the second sample respectively, and sup is the supremum function. → max value.

For large samples, the null hypothesis is rejected at level α if

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{n \cdot m}}.$$

$\alpha \rightarrow$ p-value (significance level)

Where n and m are the sizes of first and second sample respectively. The value of $c(\alpha)$ is given in the table below for the most common levels of α

α	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001
$c(\alpha)$	1.073	1.138	1.224	1.358	1.48	1.628	1.731	1.949

and in general[19] by

$$c(\alpha) = \sqrt{-\ln\left(\tfrac{\alpha}{2}\right) \cdot \tfrac{1}{2}},$$

so that the condition reads

$$D_{n,m} > \sqrt{-\ln\left(\tfrac{\alpha}{2}\right) \cdot \frac{1 + \frac{m}{n}}{2m}}.$$

Here, again, the larger the sample sizes, the more sensitive the minimal bound: For a given ratio of sample sizes (e.g. $m = n$), the minimal bound scales in the size of either of the samples according to its inverse square root.

Note that the two-sample test checks whether the two data samples come from the same distribution. This does not specify what that common distribution

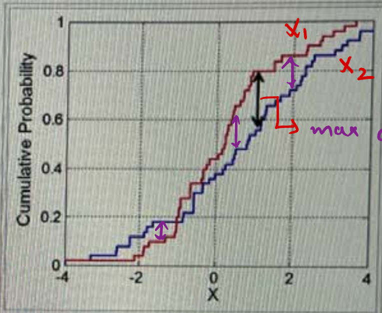
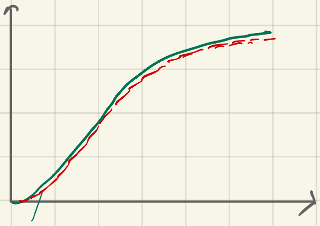


Illustration of the two-sample Kolmogorov–Smirnov statistic. Red and blue lines each correspond to an empirical distribution function, and the black arrow is the two-sample KS statistic.

if n → very large
m → very large

→ if assume they come from same disb^n, then their CDF will look like



→ The K-S test is much older than Permutation test (computational form)

→ if $\alpha = 0.05$
$n = 1000$
$m = 5000$

$D_{n,m} > 0.047$ then reject H_0 at 0.05 significance value.
↳ very small gap

→ if $n = 50$
$m = 30$
$\alpha = 0.05$

$D_{n,m} > 0.31$, reject H_0 at 0.05 sig. value.

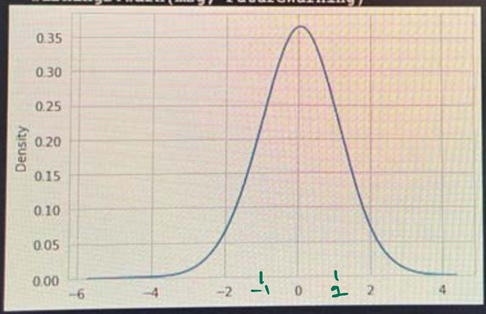
Code Snippet K-S Test

```
import numpy as np
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt

#generate a gaussian r.v X
x = stats.norm.rvs(size=1000);
print(len(x),type(x))
sns.set_style('whitegrid')
sns.kdeplot(x, bw=0.5)
plt.show()
```

→ N(0,1)

```
1000 <class 'numpy.ndarray'>
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:1657: Fut
  warnings.warn(msg, FutureWarning)
```




```
[ ] stats.kstest(x, 'norm')

KstestResult(statistic=0.0323825487968220l, pvalue=0.23991874284076287)
```

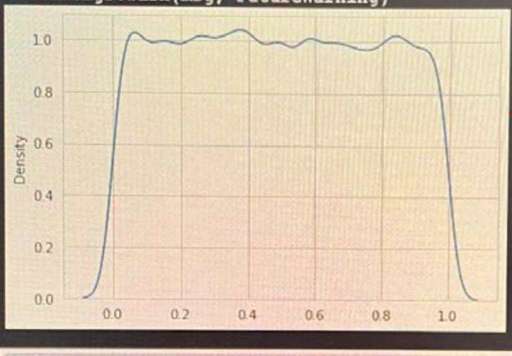
→ high, hence r.v (x) normal distributed.

→ D-value

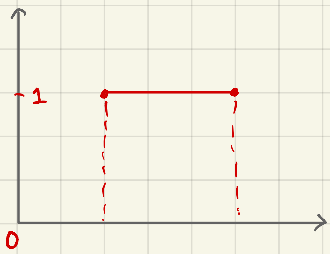
```
# Y - Continous Uniform Distribution(0,1)
y = np.random.uniform(0,1,10000);
sns.kdeplot(np.array(y), bw=0.1)
plt.show()
```

→ U(0,1)

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:
  warnings.warn(msg, FutureWarning)
```



→ Looks Like




```
[ ] !pip3 install scipy==1.6.3
import scipy
np.__version__
scipy.__version__

Collecting scipy==1.6.3
  Downloading https://files.pythonhosted.org/packages/7d/e8/43ffc
     |████████████████████████████████| 27.4MB 110kB/s
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in /usr/loca
ERROR: albumentations 0.1.12 has requirement imgaug<0.2.7,>=0.2.5
Installing collected packages: scipy
  Found existing installation: scipy 1.4.1
    Uninstalling scipy-1.4.1:
      Successfully uninstalled scipy-1.4.1
Successfully installed scipy-1.6.3
'1.4.1'
```

```
[ ] stats.kstest(y, x)

KstestResult(statistic=0.476, pvalue=2.6645352591003757e-14)
```

→ very low, so, 'y' is not normally distributed.

Hypothesis Testing :- difference of means:

eg ① :- (coin-toss) ⟶ simple to understand $p(\text{Obs}|H_0)$: very easy to compute

Task :- C_1 (city) ⟶ 1 million → (μ_1) C_2 ⟶ 2 million → (μ_2)

determine if the population mean of heights of people in these two cities is same or not.

μ_1 & μ_2 are same or different.

population mean ⟶ sample mean

expt :-

	C_1	C_2
	h_1	h_1'
randomly	h_2	h_2'
	⋮	⋮
	h_{50}	h_{50}'

↳ sample heights of 50 people

$\mu_1^s = \dfrac{h_1 + h_2 + \dots + h_{50}}{50}$; → sample mean → assume → 162 cm, observed height

$\mu_2^s = \dfrac{h_1' + h_2' + \dots + h_{50}'}{50}$; → 167 cm → observed value.

test-statistic :-

$$\mu_2^s - \mu_1^s = x^s = 167 - 162 = 5\text{cm}$$

Null-Hyp. (H_0) :-

There is no difference in population mean.

Compute :-

$p(x = 5\text{cm} \mid H_0)$ → The prob. of observing a diff. of 5cm in sample mean heights of sample size '50' b/w C_1 & C_2 if there is no population diff. in mean-heights.

↳ diff. in sample means with sample size 50.

→ we will see how to comput in next Lecture.

Let's assume

Case 1: $p(x=5 \mid H_0) = 0.2 = 20\%$ There is 20% chance of observing a diff. of 5cm in sample mean heights of C_1 & C_2 with sample of 50 if there is no population mean diff."

$$\underbrace{p(\overset{x=5}{obs} \mid \overset{H_0}{assumption})}_{\text{p-value}} = 20\% \rightarrow \text{significant} \rightarrow$$ Our assumption must be true.
⇒ accept H_0

Case 2:

$p(x=5 / H_0) = 0.03 = 3\%$

p-value

$p(obs/assum.) = 3\% \rightarrow$ small
↳ $< 5\%$

⇒ assumption must be incorrect
⇒ reject H_0 → accept H_1
↳ population mean of x_i, x_i' is not same.

Resampling and permutation test for another example:-

How to compute p-value.

we have, $x = \mu_1 - \mu_2$ ← diff. in sample means with sample size of 50.
$x = 5cm$

H_0 : no diff. in population mean

$$C_1 = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \qquad C_2 = \begin{bmatrix} h_1' \\ h_2' \\ \vdots \\ h_n' \end{bmatrix}$$

$p(x=5 \mid H_0) = ?$

step 1 :- $S = \{h_1, h_2, h_3, \ldots\ldots h_{50}, h_1', h_2', \ldots\ldots h'_{50}\}$
↳ $C_1 \cup C_2$
└→ union

step 2 :- S (100 pts) $\xrightarrow{50}$ S_1 (randomly select) → $\{h_1, h_1', h_2', h_{50} \ldots\}$ → μ_1
(Resampling) └→ S_2 (left) → μ_2
(simulate the H_0)

mean of sample

assume I get, ① $\mu_2 - \mu_1 \rightarrow 3cm \rightarrow \delta_1$

Repeat step 2 ② $\mu_2 - \mu_1 \rightarrow -2cm \rightarrow \delta_2$

,, ③ $\mu_2 - \mu_1 \rightarrow +1cm \rightarrow \delta_3$

⋮

⋮

(k) $\mu_2 - \mu_1 \rightarrow 6\text{ cm} \rightarrow \delta_K$

Let, assume $K = 1000$

Step (3) Sort δ_i's

80% of sim diff ≤ 5cm (obs. diff.)

$\underbrace{\delta_1' \leq \delta_2' \leq \delta_3' \leq}_{\text{simulated diff.}} \delta_{800}' < 5\text{cm} < \delta_{802}' \leq \delta_K'$ (incⁿ order)

20% of sim. diff. $>$ 5cm (obs. diff.)

Case 1: Obs. diff. $= x = 5$cm

p-value $\rightarrow$ $p(\text{obs. diff} \geq 5\text{cm} \mid H_0) = 20\% \rightarrow$ significant

$\rightarrow > 5\%$

assumption must be true.

$\rightarrow$ The prob. of observing a diff. of greater than equal to 5cm in sample mean of heights with a sample size of 50, assuming that there is no population diff. in means of heights is 20%.

Case 2:- $\underbrace{\delta_1' \leq \delta_2' \leq \dots \leq \delta_{970}'}_{97\%} \leq 5\text{cm} \leq \underbrace{\delta_{972}' \dots \leq \delta_{1000}'}_{3\%}$

obs. diff.

$p(\text{obs. diff.} \geq 5\text{cm} \mid H_0) = 3\% \rightarrow < 5\%$

assumption must be incorrect

$p(\text{obs. diff} \geq 5\text{cm} \mid H_0) \rightarrow$ Task:- H_0 is true or false.

$\rightarrow$ no. diffⁿ b/w μ_1 & μ_2.

But during expt. (or finding obs. diff.) we get the value of $\mu_1 - \mu_2$ (obs. diff.) $= 5$cm.

So, here my $\rightarrow$ this objective is changed into diff. b/w μ_1 & μ_2 is 5cm.

So, if $p(\text{obs. diff.} \geq 5\text{cm} \mid H_0) = 0.03$

$\rightarrow$ it means H_0 is not True.

How to use hypothesis - testing ?

→ K-S Test → used hyp-testing if two r.v have same disbⁿ or not.

Application

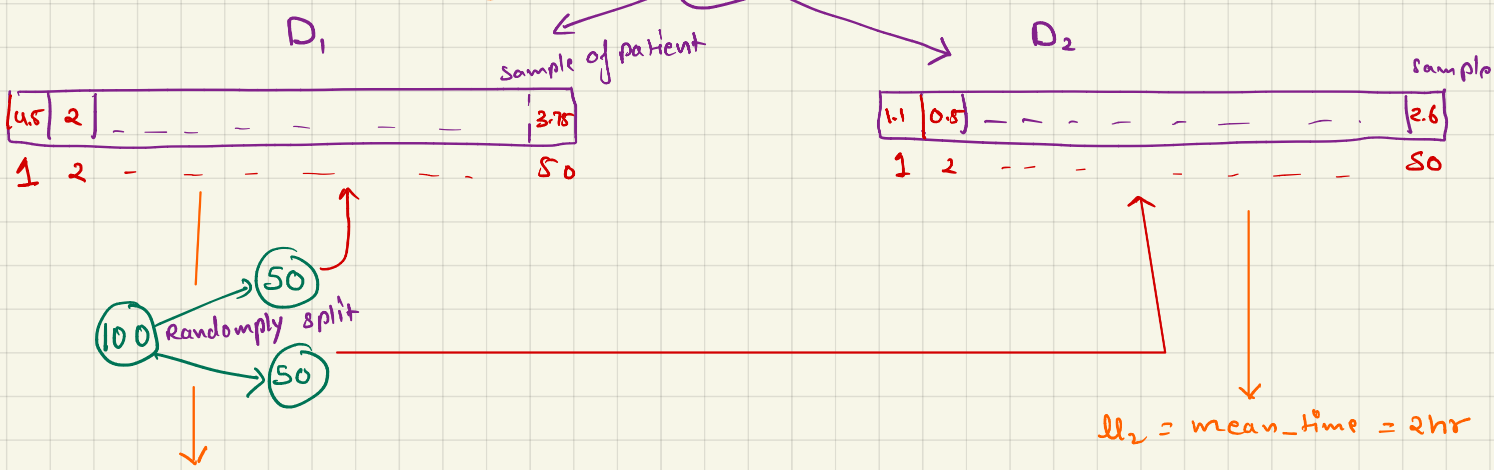
designing drugs/ medicine

D_1 (drug 1 made by C_1) (in market) → reduces your fever in 4 hrs

D_2 (C_2) → claim: we can reduce fever faster than D_1.

expt. :- To determine if the claim is true or Not.

100 (Randomly split)



μ_1 = mean-time = 4.0 hrs

μ_2 = mean_time = 2hr

→ By looking at μ_1 & μ_2 I can confidently say 'D_2' is doing well. But we are using only sample size of 50, 50. So, just using this sample size can I conduct hypothesis test?

Hyp-testing :- (1) H_0 :- The D_1 and D_2 take the same time to reduce fever.

(2) Test-statistic :- $X = \mu_2 - \mu_1 = 4 - 2 = 2$ hrs (observed value)

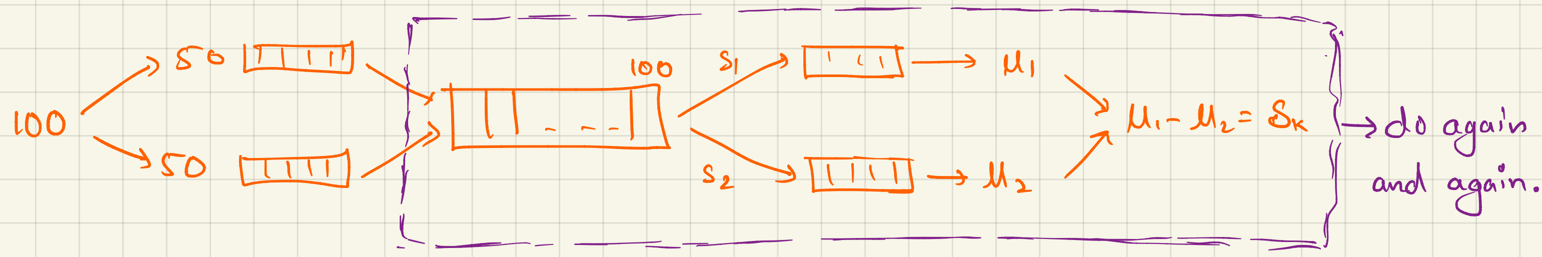
$P(x \geq 2 \mid H_0)$ = v. small → assume LET → 1%

If there is no difference in D_1 & D_2, the prob. of obs $X \geq 2$ is v. small (1%).

→ It implies, that H_0 & obs do not agree with each other. ∴ H_0 must be incorrect.

How to compute

→ resampling & permutation test



→ ③ Significance value (α)

If I want to v.v sure when I reject (H_0) / accept (H_0). we take $\alpha \rightarrow$ 1% or 0.1%

Generally, we take $\alpha \rightarrow$ 5%

If p-value < α → reject H_0
else
→ accept H_0

for eg. Hyp-testing in ecommerce → α = 3% or 5%

S_1 : Customer who use a Visa credit card → μ_1

S_2 : " " " " Master card → μ_2

→ Visa card claim customer who uses their card spend more on ecommerce.

$\boxed{\mu_1 - \mu_2 \text{ is large}}$

→ Testing.