Topics:

- ✓ ① Hypothesis testing → Toy example + code (solve...)
- ② Terminology & Math
- ③ 1-sided vs 2-sided tests
- ④ Framework of Hyp. Testing ***
- ⑤ Errors & Power

after 11:30 ↴

- → log-normal distribution ✓
- → why use n-1 instead of n for variance estimation ✓
- → why does expo-dist have mode @ 0 [next class ... more math 11:30+]

Ops!

Questions → tab

→ "END ...."

Chat → Y/N & interactivity

Hyp. Testing

- Tricky : mind-bending
- fun & lots of applications
- time to digest

Toy:

Coin → Fair : $P(H) = 0.5 = P(T)$

→ biased towards heads

$P(H) > 0.5$

	Y	NO
① $\frac{6}{10}$	1	21
② $\frac{65}{100}$ ✓	11	23
③ $\frac{641}{1000}$ ✓	33	8

intuition ✓

Numbers & Math

Hyp. testing

- Toy example: coin
  - → M_1 vs M_2 → ....
  - → distb$_1$ vs distb$_2$
  - → gender → $ ecommerce

Hyp. testing

↳ Terminology ↗ [Eqns + English]

Toy ex:

Expt: Toss coin 100 times

[Test-statistic (T): # heads in 100 tosses

obs. # of heads = 65 (let) ← T_{obs}

default Null-hyp (H0) : Coin is fair

Alt-hyp : Coin is biased towards heads

Ha or H1

$P($ observing $\underline{65}$ or more heads $\mid H_0) = ?$ ← Tobs

H_0: Coin is fair ($P(H) = 0.5$)

$P(T \geq 65) = 1 - \overline{P(T \leq 64)}$

$P(T \leq 64)$ → CDF(T) 64

100 tosses → H, T

T = # heads in 100 tosses → binomial distribution

✓ [$T \sim \text{binomial}(n = 100;\ p = 0.5)$

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

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=65, n=100, p=0.5)
print(1-prob) # P(T >= 65 | H0 )
```

```
0.0008949651957433913
```

```
# Plot PDF of Binomial(100, 0.5)
X = stats.binom.rvs(n=100, p=0.5, size=100000)
plt.grid()
sns.distplot(X)
```

```
/usr/local/lib/py... distplot` is a deprec...
```

+ Code + Text

RAM Disk

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

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=64, n=100, p=0.5)
print(1-prob) # P(T >= 65 | H0 )
```

```
0.0017588208614850442
```

0.001758

↓

0.17%

```
# Plot PDF of Binomial(100, 0.5)
X = stats.binom.rvs(n=100, p=0.5, size=100000)
plt.grid()
sns.distplot(X)
```

```
/usr/local/lib/py                                    listplot` is a deprec:
```

65 H / 100 tosses → observed data

$$P(\text{observing } 65 \text{ or more heads in } 100 \text{ tosses} \mid H_0) = 0.17\%$$

$$P(T \geq 65 \mid H_0) = 0.17\%$$

H_0: H_0 is true

coin is fair

65H

100 Tosses

19

Y

0.17%

H_1: H_a is true

coin is biased towards heads

$p(T \geq 65 \mid H_a)$

(?)

35

N

V. low ⟶ 5% (rule of thumb)

Lets:

Tobs = 52 heads

$$P(\overbrace{T \geq 52}^{\sim} \mid \overbrace{Ho}^{\sqcap}) = \overbrace{38.2\%}^{\sqcap}$$

H_0: H_0 is true

$P(T \geq 52 \mid H_0)$

$= 38.21\%$

H_a: H_a is true

$P(T \geq 52 \mid H_a)$

Not possible to calc

$$p(T \geq T_{obs} \mid H_a)$$

H_a: coin is biased towards heads

∞ many Bin dists

$\begin{cases} p(H) = 0.51 \text{ or} \\ p(H) = 0.511 \end{cases}$ $p(H) = 0.52$ or ⋮

why is H_0: coin is fair

$P(H) = 0.5$

$P(T \geq T_{obs} \mid H_0)$

Binomial dist

$T_{obs} = 5$

5H / 100 tosses

H_0: coin is fair

H_a: coin is biased towards H

Variations

- H_0: coin is fair
  H_a: coin is unfair (later)
- H_0: coin is fair
  H_a: coin is biased towards tails

discrete

"END!"

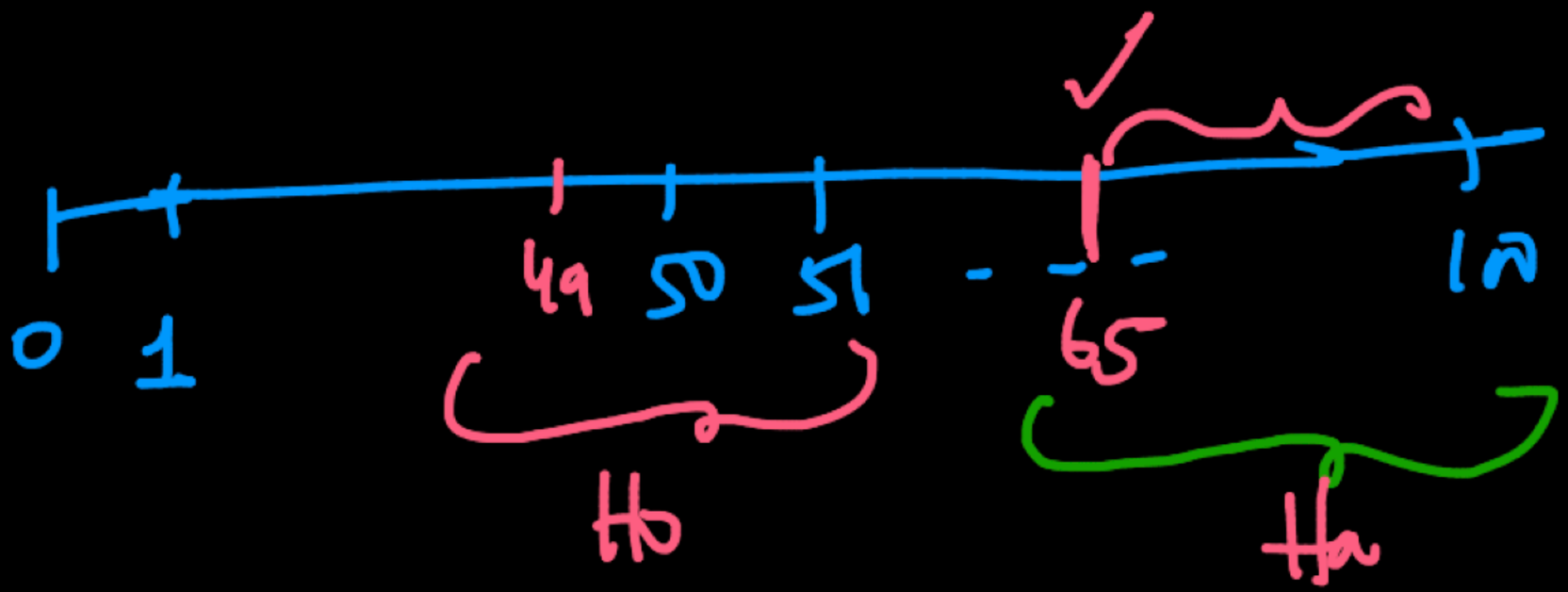
$$P(\underline{T} \geq \overline{65} = T_{obs} \mid H_0)$$

$$\{ P(T = 65 \mid H_0)$$

H_0: coin is fair

H_a: coin is biased two H

p(observing runs 65 or More extreme num heads | H_0)



$$\{ p(\text{observing } T \text{ as extreme as } T_{obs} \text{ or more} \mid H_0)$$

H_0 ✓

H_a

p-value

$p(T \geq T_{obs} \mid H_0)$ < 5%

rule-of-thumb

significance level (α)

reject H_0

⇓

accept H_a

$\Rightarrow$ p-val $< \alpha \longrightarrow$ reject $H_0 \Rightarrow$ accept H_a

$p(\overline{T \geq T_{obs}} \mid H_0) \overset{\sim}{\geq} 5\%$

p-val $\geq 5\%$

Cannot reject H_0

$\Rightarrow$ accept $H_0 \Rightarrow$ reject H_a

$$p(T \geq T_{obs} = 52 \mid H_0) = 38.21\%$$

p-val $> 5\%$

$\Downarrow$

cannot reject $H_0 \Rightarrow$ accept H_0

p-val ≠ prob that H_0 is true

↳ $P(T \geq T_{obs} \mid H_0)$

$\sqrt{65}$

$\sim\sim$

$P(\bar{T} \geq T_{obs} \mid H_0) < 5\% = \alpha$

$\uparrow$ $=$

heads in 100 tosses

$\{ \frac{6\,H}{10\ \text{Tosses}}$

Some :- ✓ H_0 & H_a → coin is biased towards H need not be exhaustive

H_0 → Coin is fair

H_0 & H_a → mutually exclusive

accept H_a

H_0: coin is fair

H_a: coin is biased H

{ 5H / 100T

$p(T \geq 5 \mid H_0) \approx 1.0$

p-val $> 5\%$

$\Downarrow$ accept H_0

In this case

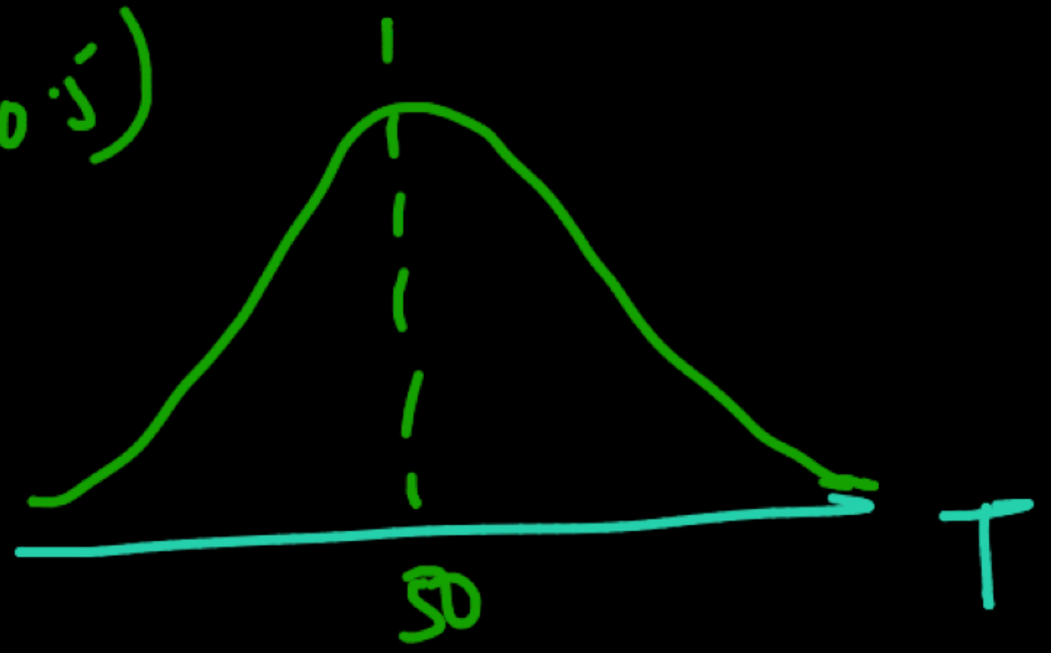
$$P(T \geq T_{obs} \mid H_0)$$

P(observing T as extreme or more than T_{obs} | H_0)

$P(-\leq T \leq - \mid H_0) \rightarrow$ (later)

$$p(\widetilde{T \geq T_{obs} \mid H_0})$$

$T \sim Bin(n = 100; p = 0.5)$



Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:16

Hypothesis Testing.ipynb | Log-normal distribution - | Bias of an estimator - Wik | New Tab | scipy.stats.binom — SciP

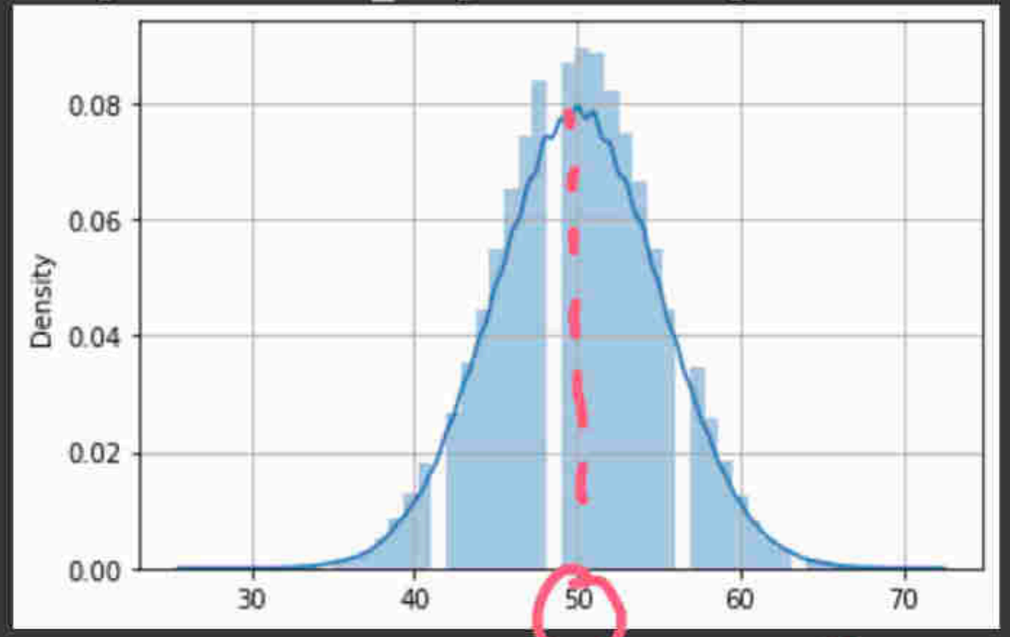
colab.research.google.com/drive/1uNuXZ3pBTxJxZTdJEx-AHzq3ix8teh5n#scrollTo=wBXEHYCIKb9u

+ Code + Text

RAM
Disk

```
[3]  # Plot PDF of Binomial(100, 0.5)
     X = stats.binom.rvs(n=100, p=0.5, size=100000)
     plt.grid()
     sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```



n·p=50

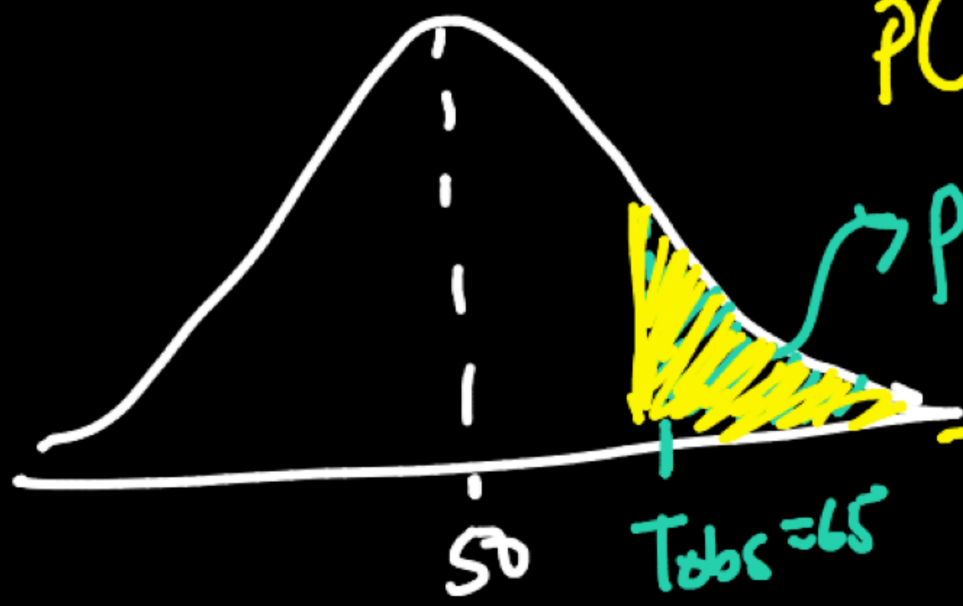
Double-click (or enter) to ed

37 / 37

$$P(\tilde{T} \geq T_{obs} = 65 \mid H_0)$$

given H_0: $T \sim$ Bin $(n=100, p=0.5)$ → Discrete r.v

$P(T \geq 65 \mid H_0)$



→ p-value 0.17%

T under H_0

1000 → A → Top 10 → $d_1, d_2, \ldots d_{100}$ → m

$m > m'$

μ vs μ' ✓

1000 → B → no Top 10 → $d_1', d_2' - d_{100}'$ → m'

Z-test
T-test

$$\begin{cases} H_0: \mu = \mu' \\ H_a: \mu \neq \mu' \end{cases}$$

z test

T test

(later)

{ ✓ 65 Heads / 100 Tosses } → { $\underline{P(H)} = 0.5$ ✓ or $P(H) > 0.5$

H_0
Coin is fair

rejecting H_0

H_a
Coin is biased H

A|B

C.I ✓

Hyp. testing ✓

reject H_0 → accept H_a

$P(T \geq T_{obs} \mid H_0)$ is v. low

$P(T \geq T_{obs} \mid H_0) < 5\%$

65

0.17%

reject H_0 ⇒ accept H_a

Coin is fair

$P(H) = 0.5$

Coin is Biased

H

$P(H) > 0.5$

Guilty

not Guilty

Terminology:

Critical-value

$T_{obs} = 65$

$p\text{-val} < 5\% \Rightarrow$ reject H_0

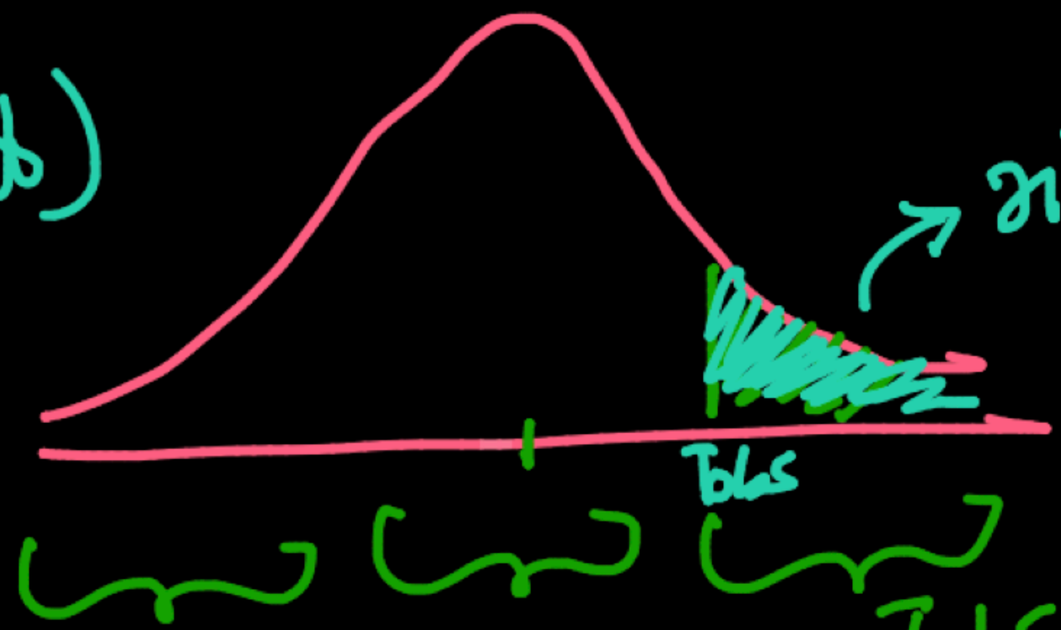
Var 1:

H_0: coin is fair

H_a: coin is biased

Tobs

$p(T \geq T_{obs} \mid H_0)$



Var: 2

H_0: coin is fair

H_a: coin is biased

Expt: 100 tosses

T: #heads

$T \sim T$

Tobs

$P(T \le T_{obs} \mid H_0)$

left-tailed test

$$p(\text{observing } T \text{ as extreme or more than } T_{obs} \mid H_0)$$

Var 3:

H0: Coin is Fair

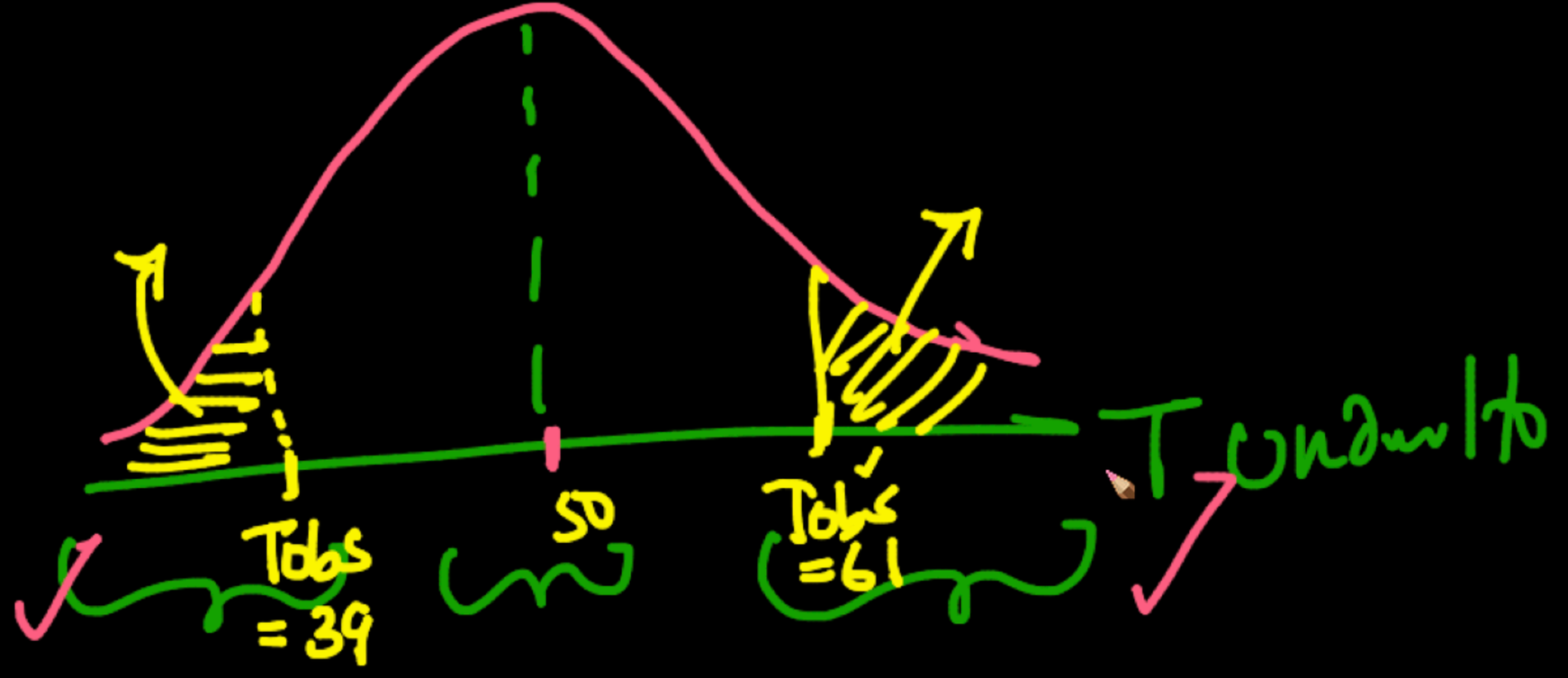
Ha: Coin is biased

H

T

Tobs

2-sided test



$$P(T \leq 39 \text{ or } T \geq 61 \mid H_0)$$

Fair or unfair coin?

when Ho is true

$T \sim Bin(n=100; p=0.5)$

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

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=4, n=100, p=0.5)
print(1-prob) # P(T >= 65 | H0 )
```

```
1.0
```

$W_0 \rightarrow H_0$ is true

$\downarrow$

accepted H_0

p-val $> 5\%$

$$p(T \geq T_{obs} \mid H_a) \quad X$$

$\Downarrow$

∞-Many Cases

$$\begin{cases} p = 0.51 \\ p = 0.501 \\ \vdots \end{cases}$$

+ Code + Text

RAM Disk

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=651, n=1000, p=0.5)
print(1-prob) # P(T >= 651 | H0 )
```

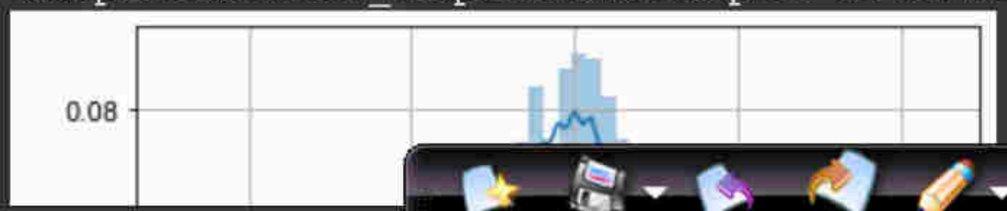
651H / 1000 Trails

$P(T \geq 651 \mid H_0) \approx 0.0 < 5\%$

```
1.1102230246251565e-16
```

```
# Plot PDF of Binomial(100, 0.5)
X = stats.binom.rvs(n=100, p=0.5, size=100000)
plt.grid()
sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```



```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=651, n=1000, p=0.5)
print(1-prob) # P(T >= 651 | H0 )
```

$p(T \geq T_{obs} \mid H_0)$

```
1.1102230246251565e-16
```

```
[3] # Plot PDF of Binomial(100, 0.5)
    X = stats.binom.rvs(n=100, p=0.5, size=100000)
    plt.grid()
    sns.distplot(X)
```

Disb of T under Ho

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```

+ Code + Text

RAM
Disk

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=651, n=1000, p=0.5)
print(1-prob) # P(T >= 651 | H0 )
```

```
1.1102230246251565e-16
```

```
[3] # Plot PDF of Binomial(100, 0.5)
    X = stats.binom.rvs(n=100, p=0.5, size=100000)
    plt.grid()
    sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```

61 / 61

+ Code + Text

RAM
Disk

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=651, n=1000, p=0.5)
print(1-prob) # P(T >= 651 | H0 )
```

```
1.1102230246251565e-16
```

```
[3] # Plot PDF of Binomial(100, 0.5)
    X = stats.binom.rvs(n=100, p=0.5, size=100000)
    plt.grid()
    sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```

61 / 61

+ Code + Text

RAM
Disk

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards heads
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=651, n=1000, p=0.5)
print(1-prob) # P(T >= 651 | H0 )
```

```
1.1102230246251565e-16
```

```
[3] # Plot PDF of Binomial(100, 0.5)
    X = stats.binom.rvs(n=100, p=0.5, size=100000)
    plt.grid()
    sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```

61 / 61

Chrome File Edit View History Bookmarks Profiles Tab Window Help 11:08

Hypothesis Testing | Log-normal distribu | Bias of an estimator | New Tab | scipy.stats.binom | What is an A/B Test

colab.research.google.com/drive/1uNuXZ3pBTxJxZTdJEx-AHzq3ix8teh5n#scrollTo=wBXEHYCIKb9u

+ Code + Text

RAM Disk

[5]

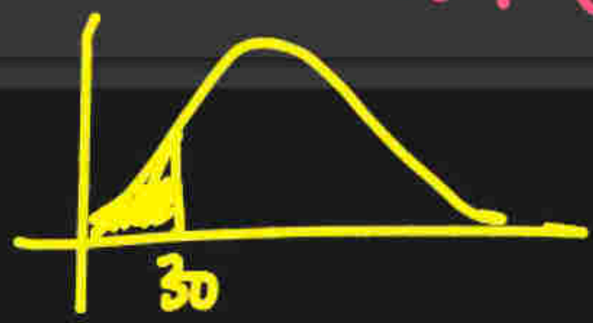
```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is biased towards tails
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=30, n=100, p=0.5)
print(prob) # P(T >= 651 | H0 )
```

$P(T \leq 30 \mid H_0)$

$= CDF(T, 30)$

```
3.925069822796833e-05
```

$= 3.9 \times 10^{-5} < 5\%$

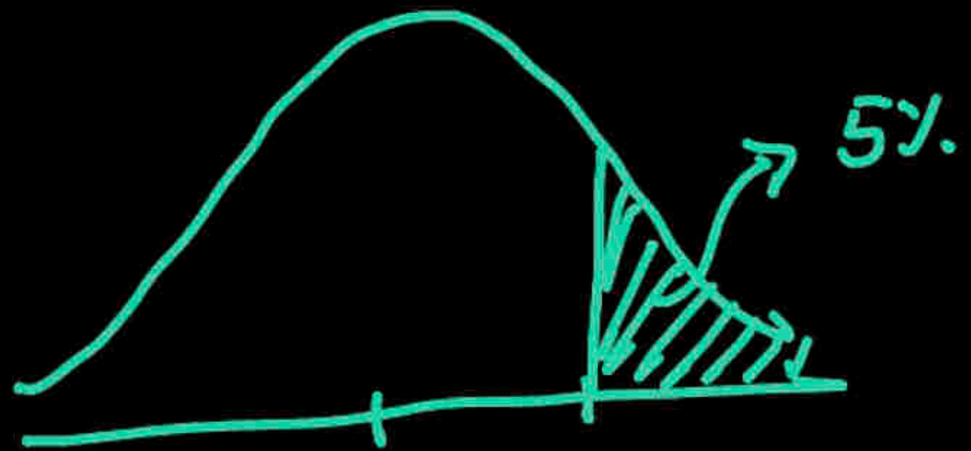



```
# Plot PDF of Binomial(100, 0.5)
X = stats.binom.rvs(n=100, p=0.5, size=100000)
plt.grid()
sns.distplot(X)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprec
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fd257cb9110>
```

62 / 62

Critical-Val: T_{cr}



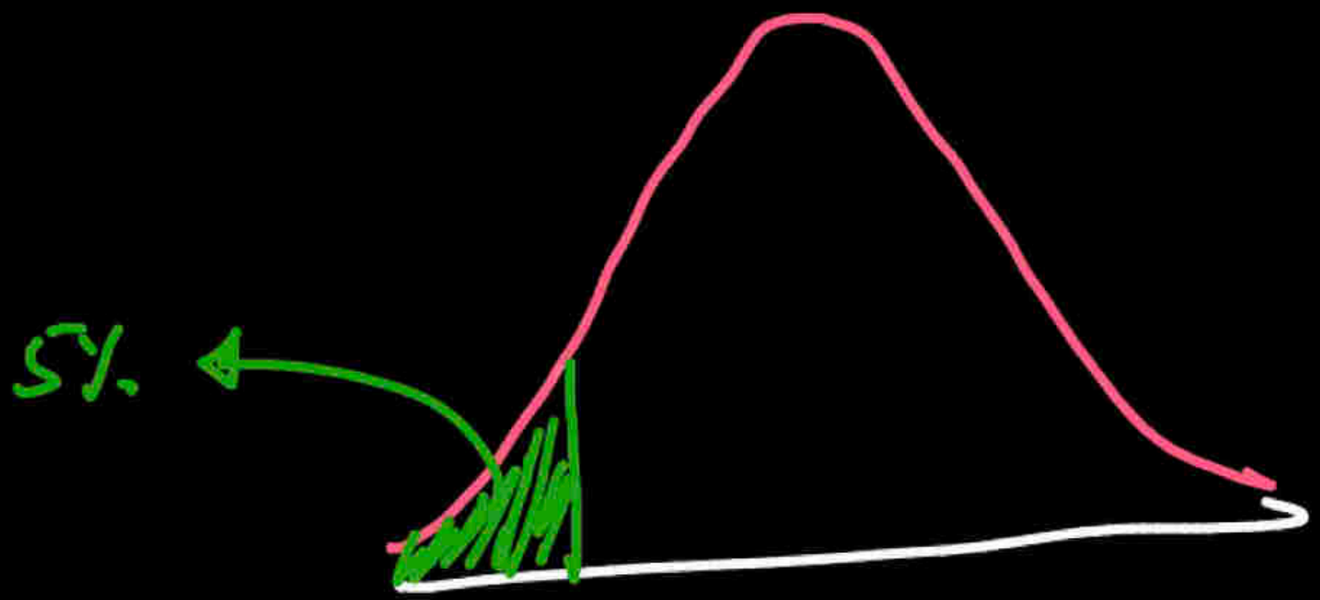
accept H_0 ← → reject H_0

T_{cr}

H_0: coin is fair

H_a: coin is biased

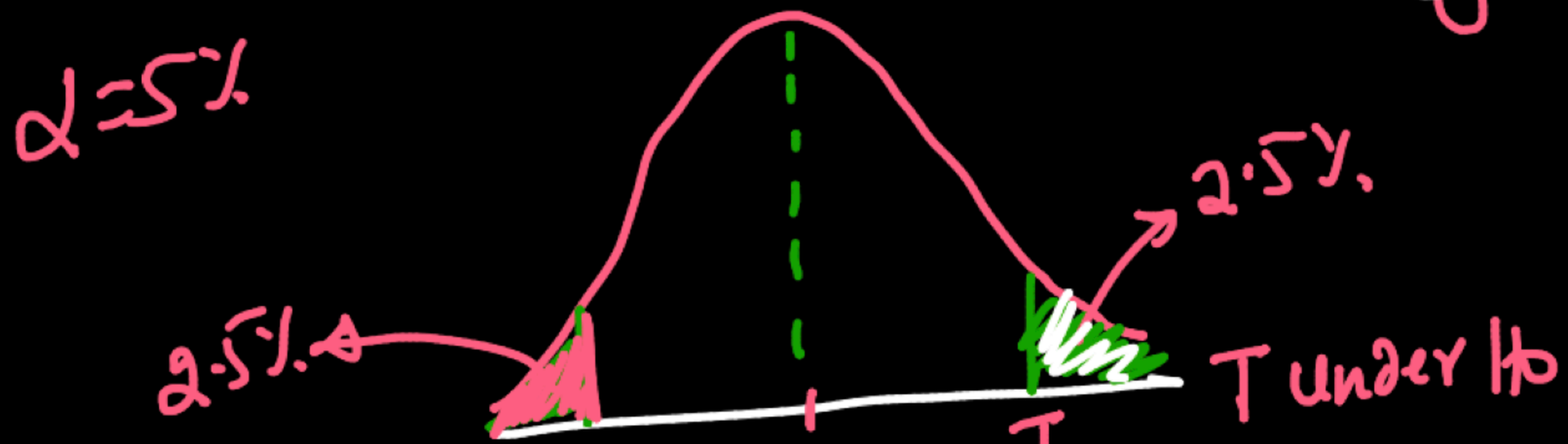
T = #heads



reject H_0 ← → accept H_0

T_{cv} for 2-tailed

H_0: coin is fair
H_a: coin is unfair
Symm. Curve



reject Ho
accept Ho
reject Ho

Framework of Hyp. Testing

① Define H_0, H_a carefully based on what you want

② Define Expt & "sensible" Test Statistic

✓ [dist of T under H_0]

(3) One-sided (R or L) or 2-sided test

↓

T, H_0, H_a

(4) $P(T \text{ is as extreme as } T_{obs} \mid H_0) = p\text{-val}$

(5) fix α = 5% → default

some-cases 1% or 0.1% (later)

(6) Compare p-val vs α

p-val $< \alpha$ or p-val $\geq \alpha$ ⇓ reject H_0 ⇓ accept H_a

cannot reject H_0 ⇓ accept H_0

Same-framework

- z-list
- t-list
- z-poop list
- x1-list
- AwsOA

→

what if T under H0 doesn't follow any dist

↓

[permutation - resampling test]

more advanced

bootstrapping like

$p(T \geq T_{obs} \mid H_0)$ and

$p(T \geq T_{obs} \mid H_a)$

CONFUSION MATRIX → Hyp. testing

actual

Errors

	Ho ✓	Ha ✓
Ho	✓	β (error) X
accept Ha ✓	α (error)	1-β ✓ = power of stat. test

$P(T \geq T_{obs} \mid H_0) < \alpha = 5\%$

$\{ 4.999999\%$

reject H_0 incorrectly

reject H_0

α = prob of rejecting H_0 incorrectly

Medicine!

M_1 (placebo)

M_2 (Remdesivir)

$\alpha = 5\%$ (let)

z-Test

prob of concl. Ha is true when Ho is actually true

Ho: both Medicines are same

Ha: both Medicines are different

α = prob of incorrectly rej Ho

$\alpha = 1\%$ or 0.1%

RESULTS

A total of 1062 patients underwent randomization (with 541 assigned to remdesivir and 521 to placebo). Those who received remdesivir had a median recovery time of 10 days (95% confidence interval [CI], 9 to 11), as compared with 15 days (95% CI, 13 to 18) among those who received placebo (rate ratio for recovery, 1.29; 95% CI, 1.12 to 1.49; P<0.001, by a log-rank test). In an analysis that used a proportional-odds model with an eight-category ordinal scale, the patients who received remdesivir were found to be more likely than those who received placebo to have clinical improvement at day 15 (odds ratio, 1.5; 95% CI, 1.2 to 1.9, after adjustment for actual disease severity). The Kaplan–Meier estimates of mortality were 6.7% with remdesivir and 11.9% with placebo by day 15 and 11.4% with remdesivir and 15.2% with placebo by day 29 (hazard ratio, 0.73; 95% CI, 0.52 to 1.03). Serious adverse events were reported in 131 of the 532 patients who received remdesivir (24.6%) and in 163 of the 516 patients who received placebo (31.6%).

CONCLUSIONS

$\alpha \rightarrow$ prob of incorr rej Ho

high cost $\Rightarrow$ lower α

Tails

$P(T \geq 65 \mid H_0) = 0.17\% = 0.0017$

very low

very low

coin is fair

very

$T_{obs} = 65$ kya:

Ho is true ✓

↓

p(obs 65 or More)

↓

v. low

↓

X

Ha is true ✓

p(obs 65 or More)

X

✓

{ ~ 65 H / 100 Tosses → fair ~50H / 100 Toss | biased H >>50H / 100 Tosses

coin-toss

$P(T \geq 65 \mid H_0) = 0.17\% > \alpha$

$\alpha \downarrow$

$\alpha = 0.1\%$

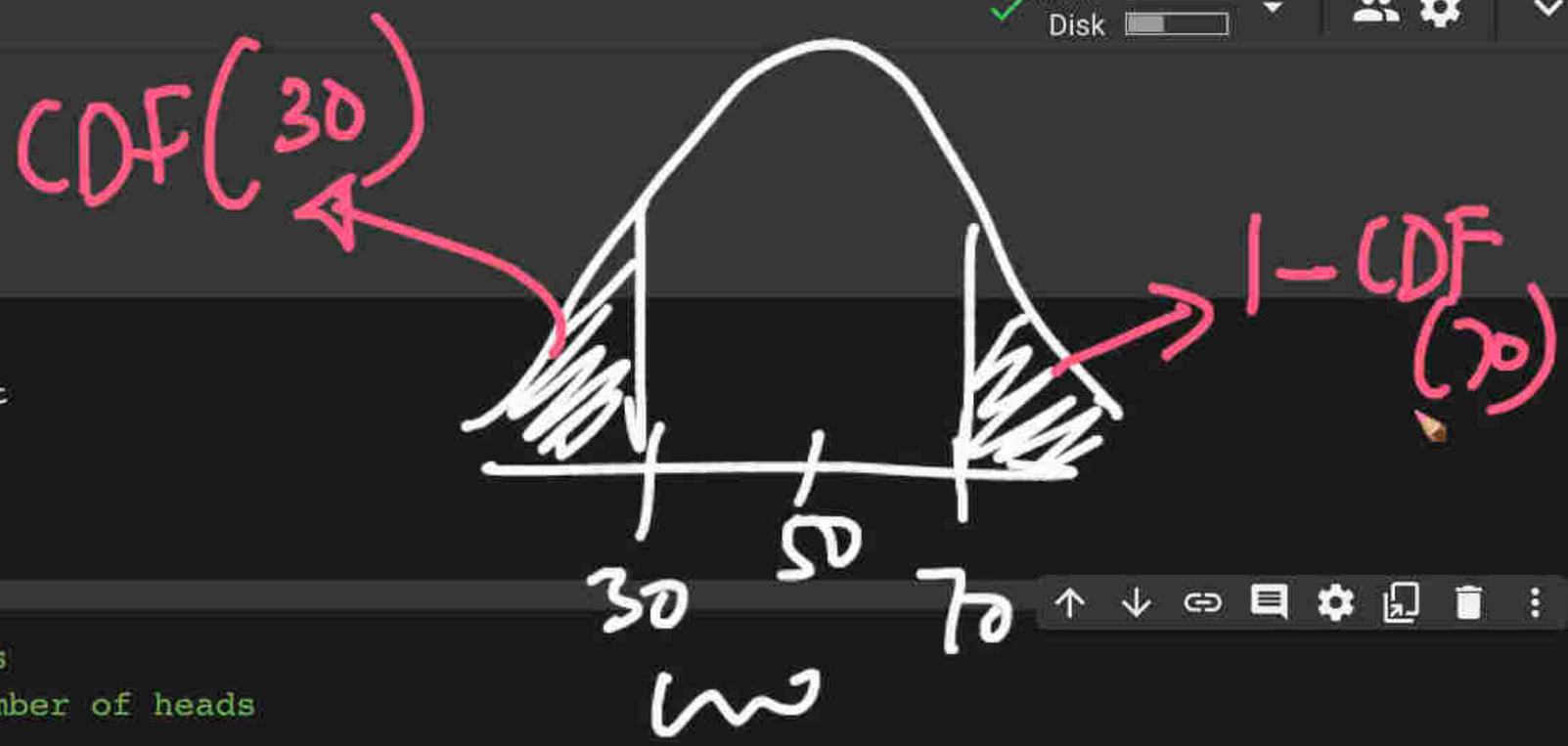
do not want incorrectly reject H_0

reject H_0

accept H_0

$$\begin{cases} \text{Z-test}: & \widetilde{\text{Test-st.}} \sim Z(0,1) \\ \text{T-test} & \to \text{〃} \sim \text{t-dist}(\nu) \\ \chi^2\text{-test} & \to \text{〃} \sim \chi^2\text{-dist} \\ \text{ANOVA} & \to \text{〃} \sim F\text{-dist} \end{cases}$$

Fair or unfair coin?




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

```
# Expt: Toss the coin 100 times
# Test statistic: Count the number of heads
# H0: Coin is fair
# Ha: Coin is unfair
# T ~ Binomial(n=100, p=0.5) under Null Hypothesis(=H0)

prob = stats.binom.cdf(k=30, n=100, p=0.5)
print(prob) # P( T<=30 or T>=70  | H0 )
```

```
3.9250698227968 33e-05
```

Mutually exclusive

Dice = 1, 2, ... 6

{1, 2}

{3, 4, 5}

$E_1 = \{1, 2, 3\}$

$E_2 = \{2, 4, 6\}$

$E_1 \cap E_2$

$E_1 \cap E_2 = \phi$

no common outcomes

↓

Mutually exclusive

mutually exhaustive

$E_1 \cup E_2 = \text{all}$ `U`

possible

$E_1 = \{2, 4, 6\}$

$E_2 = \{1, 3, 5\}$

→ mutually exd

prob ≈ 'likelihood'

0 - 1

PMF → disc. r.v

1

$P(X=x)$

0

X

PDF → Cont r.v

∞

ref. likelihood

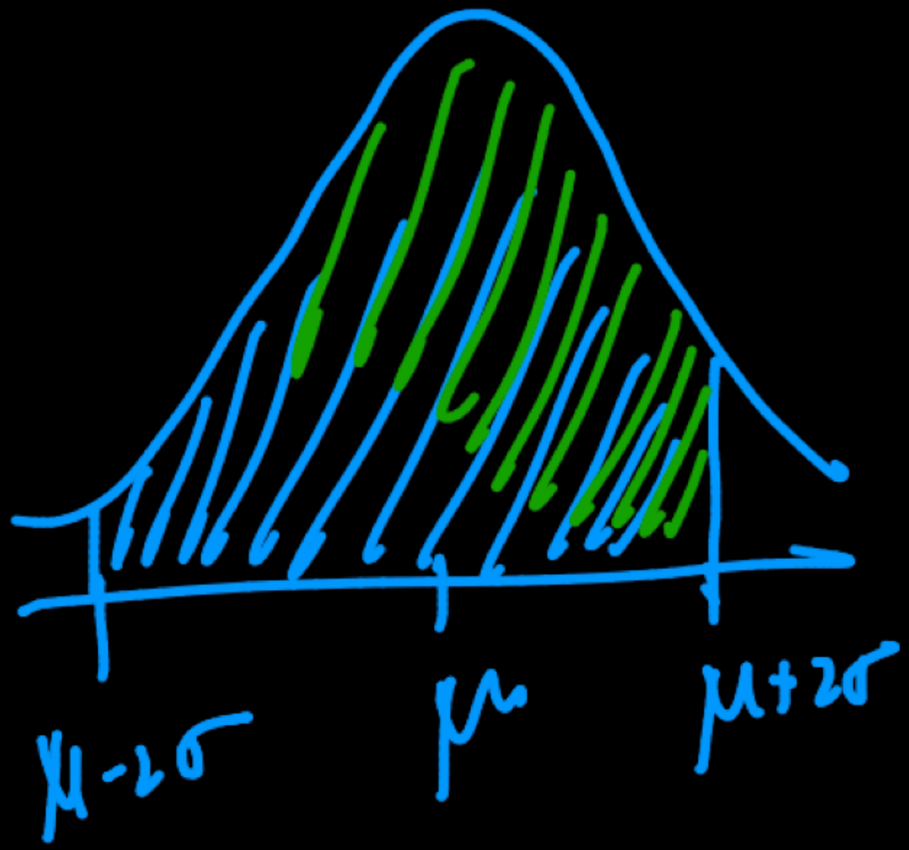
0 x_1 x_2 X

$P(X=x_1)=0$

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

$$\int_{\mu-1.96\sigma}^{\mu+1.96\sigma} p(x)\,dx = 0.95$$

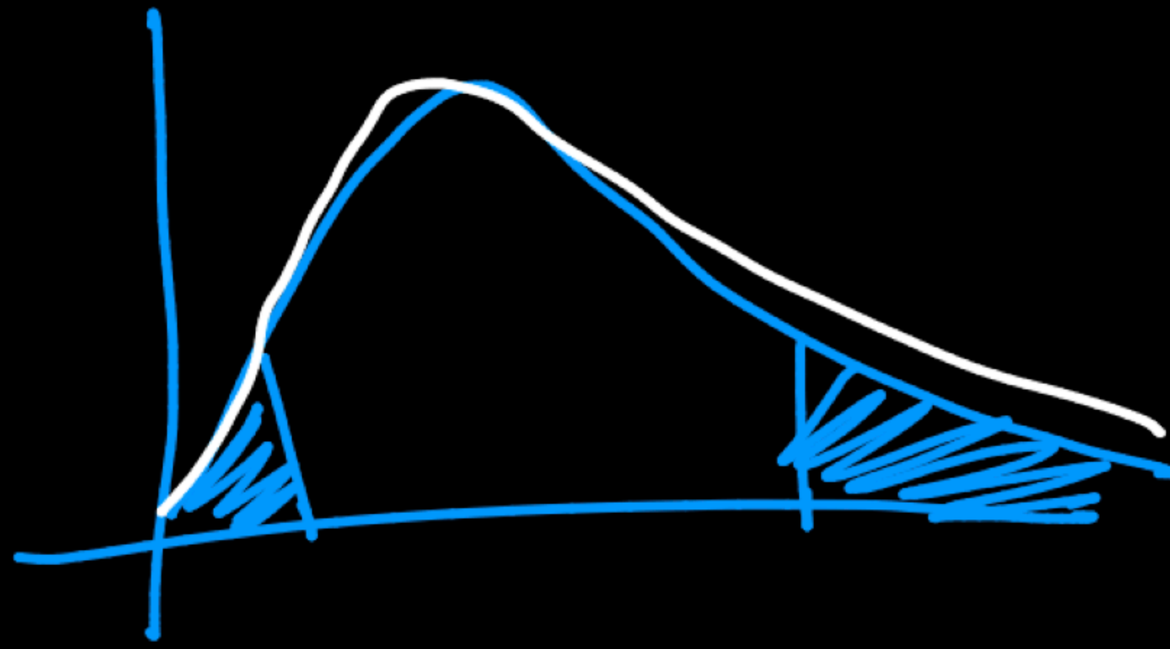
pdf



2-tailed

p-val vs α = 5%

[area under both tails



$$\alpha = 0.1\%$$

✓ [H_0: both med are same

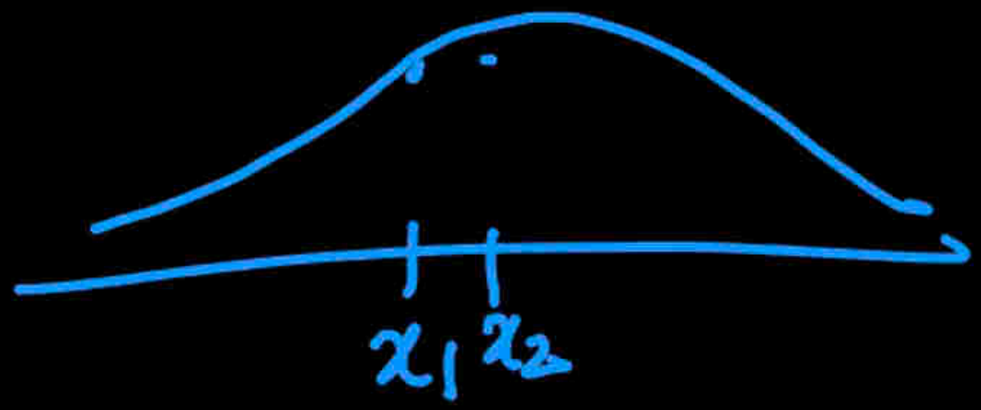
H_a: " " diff

$\alpha = 0.1\%$

PDF

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

PDF : $P(x)$ → eqn



Dataset: $x_1 \cdots x_n$

↓

CDF

CDF

1

y_2

y_1

0

x_1 x_2

↳ v. small

X

PDF

$\frac{y_2 - y_1}{x_2 - x_1}$

x_1

X

Simplified

Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

In other projects
Wikimedia Commons

Languages
العربية
Español
Français
हिन्दी
मराठी
Português
தமிழ்
اردو
中文
61 more

Edit links

natural and social sciences to represent real-valued random variables whose distributions are not known.[2][3] Their importance is partly due to the central limit theorem. It states that, under some conditions, the average of many samples (observations) of a random variable with finite mean and variance is itself a random variable—whose distribution converges to a normal distribution as the number of samples increases. Therefore, physical quantities that are expected to be the sum of many independent processes, such as measurement errors, often have distributions that are nearly normal.[4]

Moreover, Gaussian distributions have some unique properties that are valuable in analytic studies. For instance, any linear combination of a fixed collection of normal deviates is a normal deviate. Many results and methods, such as propagation of uncertainty and least squares parameter fitting, can be derived analytically in explicit form when the relevant variables are normally distributed.

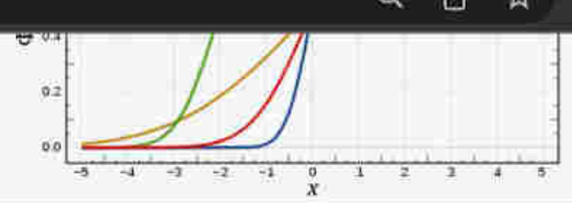
A normal distribution is sometimes informally called a **bell curve**.[5] However, many other distributions are bell-shaped (such as the Cauchy, Student's *t*, and logistic distributions).

The univariate probability distribution is generalized for vectors in the multivariate normal distribution and for matrices in the matrix normal distribution.

Contents [hide]

1 Definit

1.1 Standard normal distribution



Notation	$\mathcal{N}(\mu, \sigma^2)$
Parameters	$\mu \in \mathbb{R}$ = mean (location) $\sigma^2 \in \mathbb{R}_{>0}$ = variance (squared scale)
Support	$x \in \mathbb{R}$
PDF	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
CDF	$\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$
Quantile	$\mu + \sigma\sqrt{2}\,\operatorname{erf}^{-1}(2p-1)$
Mean	μ
Median	μ
Mode	μ
Variance	σ^2
MAD	$\sigma\sqrt{2/\pi}$
Skewness	0
Ex. kurtosis	0
Entropy	$\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}$
MGF	$\exp(\mu t + \sigma^2 t^2/2)$
Fisher	$\begin{pmatrix} 1/\sigma^2 & 0 \end{pmatrix}$