

Numpy →

1) array()

2) arange()

3) where.replace(array, delimiter, to_replace)

a = [1, 2, 3, 4, 5]

b = [100, 200, 300, 400, 500]

4) np.column_stack([a, b])

↳ [[a₀, b₀], [a₁, b₁] ...]

5) np.where(a > 2, True, False)

↳ Value if condition is true. ↳ Value if condition is false.
↳ List of all the index where the condition is satisfied.

6) np.set_printoptions(suppress = True, precision = 3)

↳ pretty print.

7) np.argsort(aarr) / np.argsort(aarr)

↳ returns sorted array.

↳ returns indices according to the sorted array.

8) Matrix multiplication ↳

np.dot(a, b)

a @ b

np.matmul(a, b)

reshape but ↳

a = [[1, 2, 3], [4, 5, 6]]

a.reshape(6, 1) ↳ vector
a.reshape(6, -1) / a.reshape(6, 1) ↳ 6x1 matrix.

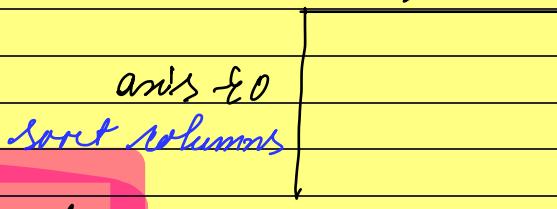
q) `np.vectorize` (python fn) \rightarrow Vectorsizes a normal python fn.
 - now the fn can take np.array as input & operate on the elements of the array individually.
 $\text{Or without a for loop. (classic np style)}$

10) `np.tile(a, n)`

11) `np.split(a, n)`
 $\text{Or number/list of indices.}$

3/6/25 Lecture - 2 - notes

1) Sorting of arrays & 1 \rightarrow sort rows



4) Broadcasting \rightarrow Expanding smaller array like two arrays to perform operations like them smoothly.

a) (2×3) op (1×3) op ✓

b) op ✓
 5×4 1×3
 \rightarrow One dimension of the broadcasted vector should be 1.

c) op 1×5 $\rightarrow 4 \times 5$
 4×1 1×5

Splitting ->

np.split (array, axis = 0/1) → split → splits array into equal sections.
(np.hsplit → splits columns)

$$\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & 8 \end{array} \quad \text{vsplit}$$

hsplit →

polymorphism \rightarrow np.split(ave, [list of indices])

Splits the array according to the indices provided.

3

Stacking \Rightarrow np.vstack((...)), np.vstack((...))

Input -> List of arrays.
Output ->

If (# of rows or col ==) array (rows, col, & colz) array (row, col, col)

else error

else error.

Indening

$$A = \begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix}$$

$\alpha [\cdot, 0] \rightarrow (3,)$
[1, 4, 2] (More unknown)

$\partial [: , [\circ]] \rightarrow (z,)$
 $[[1]]$ (columns)
 $[[4]]$,
 $[[?]]$

D-3 Pandas

- 1) df.info() →
 - o is default of Pandas will get confused.
- 2) df.drop(columns , axis=1, inplace = True) → False is default.
- 3) unique() / nunique()
- 4) df[col].value_counts()
- 5) Indexing & Implicit & Explicit.
- 6) df.duplicated()
- 7) df.drop_duplicates(keep = 'first')

X

- 1) column.unique() → Returns a list of unique values in a column.
- 2) column.nunique() → Returns an integer, # of unique values in the column.
- 3) column.value_counts() → Returns a dictionary of unique values & their corresponding counts.

Indexing →

- 1) Implicit index & Default index of integers.
- 2) Explicit index & Custom index assigned by programmer, usually values of a different column.

looks for indexing →

- loc[] & Poor implicit.
- iloc[] & Explicit.

Handling Duplicates →

- 1) df.duplicated(keep = first/last/False) & Returns true for duplicate rows.

24 df.drop_duplicated(keep, subset = [cats])

Drops rows based on duplication in a specific column
→ Drops duplicate rows & returns a new df.

In pandas ∇ 100% = max.
→ that is a number.

D - 4

Amaran

1) pd.merge(df₁, df₂, on = , how =)
↳ join df₁ & df₂. ↳ The when to join on. ↳ inner/outer/ left/right.

2) df.groupby([col/cols...]) groups dataframe based on common values in columns/columns.

Ex ↳ grouped_df = df.groupby(col)
↳ gdf!

gdf.apply(fu) → applies the fu to the entire group.

gdf.agg(['col']): [aggregate fns] → applies a list of aggregate fns to the columns provided in the list/happy as keys.

Advanced groupby ↳

↳ groupby returns an iterable of objects containing group by key dfpr & value.

Ex ↳ for group frame in df.groupby('category'): ↳ do some operation.

2) passing a fu to groupby ↳

df.set_index(col)
for group frame in df.groupby(custom_fu)
↳ operates on index if col is not provided.

groupby is a two step process

↳ Split the data to groups.

2) apply something to each group.

Split the data frame ↗

1) Splitting by default is done on index.
for multi-index → groupby(levels = (0,1))

Index of each column

Processing on the split dataframe ↗

Three data processing categories on a

groupby object ↗

1) Aggregation ↗ gdf.agg(Dictionary) ↗ Returns a single df.

2) Transformation ↗ gdf[cols].transform([fn/list of fn])
↗ Returns a dataframe, same size as the original dataframe.

3) Filtering

Scales ↗ Different types of data ↗

1) Nominal ↗ (categorical) & Data is split into categories with no particular order
e.g. the categories. ex. Nationality.

2) Ordinal ↗ Data is split into categories but there is order over the categories.

ex. Academic grades.

3) Interval → units in the data are equally spaced but there is no true zero. ✓

ex) Temperatures, compass. 0th unit has a non-null value.

4) Ratio Scale → - units are equally spaced.
- There is a true zero.

ex) Height, weight, length, time.

Scale operations

1) Nominal —

2) Ordinal L/Y (greater than / less than)

3) Interval +/-

4) Ratio +/-/x/%

→ Pd. CategoricalDTypes (List of categories I,
ordered = True)

→ pd.melt (id_vars = [], value_vars = [],
value_name = , value_name)
name of new column variable. name of value column.

- Converts a table from wide format to long.

↳
Suppose we have many columns
to a single column.

- Creates two columns

↳ To hold column names.

↳ To hold corresponding values.

D-5

Continuous & variable Boundary values to split & col to categories.

↳ pd.cut(val, bins=[], labels=[]) ↴

2) isna() / isnull → Returns a boolean mask.
what to name each category.

True → value missing.
False & not missing value.

Visualization ↴

↳ plt.axvline(val, color, linestyle) → Prints a vertical line on figure at position = val.

Univariate data ↴

Data Type

Continuous → histogram

Discrete → bar graph.

Q: Plot average ratings by month-year.

Bivariate Data ↴

x

y

Plot-type

Continuous vs Continuous

Scatter

Day - 6

17 Plt. grid (Visible, color, linestyle, linewidth)

- To plot grid lines in the figure.

Sachin Tendulkar's Career Analysis &

↳ df. interp).

24 Check if all the columns are in the right format.

Probability Theory &

Intersection & Inner join.

union & concat() + drop_duplicates()

Statistics

Measures of Central Tendency

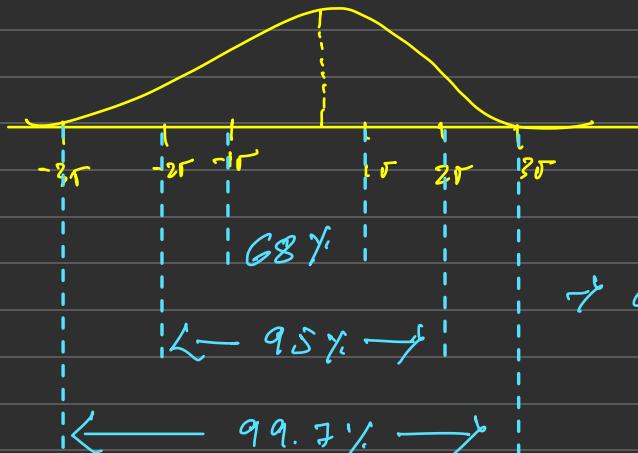
- 1) Mean (μ)
- 2) Median (M)
- 3) Q_1, Q_2 .

Measures of spread

- 1) Range
- 2) Variance
- 3) Standard Deviation.

Properties of normal distribution

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

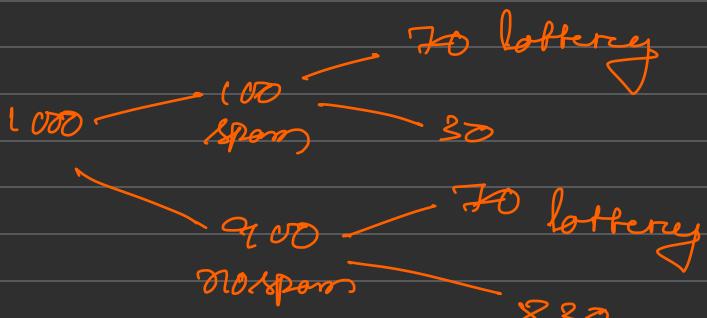


→ 68-95-99.7 rule.

Q2:

$$P(\text{lottery}) = 0.14$$

$$P(\text{lottery} | \text{spam}) = 0.7$$



$$\begin{aligned}
 P(\text{spam} | \text{lottery}) &= \frac{P(\text{lottery} | \text{spam}) P(\text{spam})}{P(\text{lottery})} \\
 &= 0.7 \times 0.1 / 0.14 = 0.5
 \end{aligned}$$

Bayes Theorem ↗

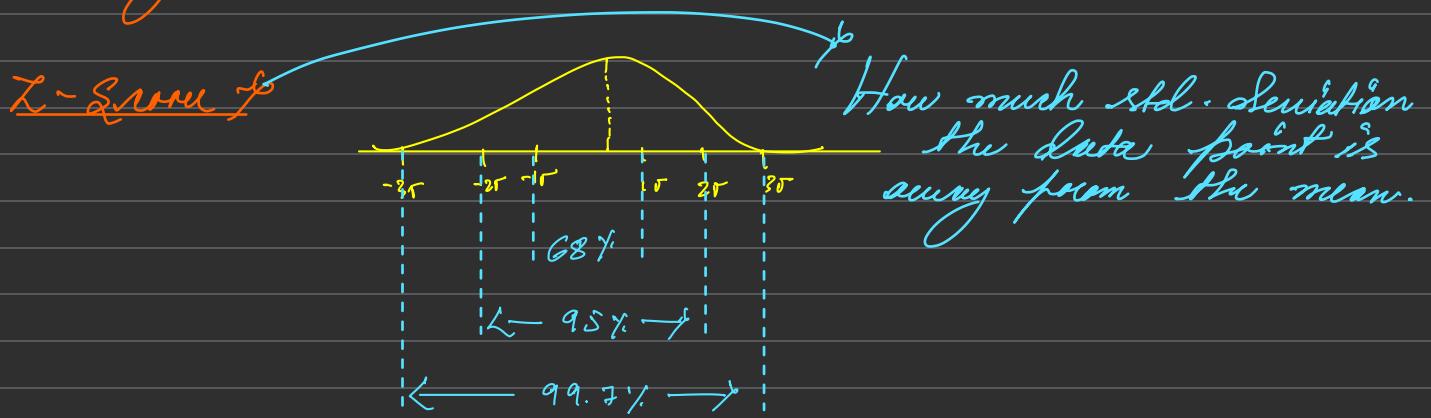
$$P(a|b) = \frac{P(b|a) P(a)}{P(b)}$$

Joint probability.
Prior
Class probability.

Posterior

Population & Sample ↗

Sampling Distribution ↗ (Distribution of sample means).



$$\lambda = P(x \leq X) \rightarrow \frac{|x - \mu|}{\sqrt{\sigma^2}} \rightarrow \lambda \text{ more } \rightarrow \text{Consult Z-table for } P(x \leq X).$$

$\sqrt{\sigma^2}$ → Standard Deviation.

Q. ↗ what should be the size of sample?

Central Limit Theorem ↗

1) If true any distribution, the sampling distribution of sample means has a normal distribution.

2) Sample size & fit of the sample dist of sample means.

Standard Deviation ↗ It is the measure of how much away the mean of the sampling dist of sample means is from the population mean.

$$S.E = \frac{\sigma}{\sqrt{\text{sample size}}}$$

or can be considered standard deviation of sample dist.

$S.E \propto \frac{1}{\sqrt{\text{sample size}}}$ measure of how well the sampling dist of sample means fits the population distribution.

Confidence Interval ↗ Converting from % of area to data-point values through Z-score.

↗ An estimated range that could include the true population statistic (mean) with a certain probability.

- Relation with hypothesis testing ↗

— Using confidence interval we can get a range within which the true population parameter must be present with certain confidence.

In SPSS medical_data ↗

import scipystats as stats ↗ Confidence interval.

stats.norm.ppf((1 + 0.1) / 2)

↳ for to return the Z-critical value.

lower_limit = mean - (std_error * Z-critical)

if sample mean between lower & upper
fail to reject H_0 .

else
reject H_0

Probability & Binomial distribution &

from scipy.stats import binom
binom(n, k, p)
& success probability.

Poisson -> Rate of occurrence of an event is given.

-> To find the $p(\# \text{ of occurrence})$ of an event within an interval.

$$P(x=k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$p(\# \text{ of occurrences of an event})$ given the rate of occurrence of event.

Q-Q plot -> To check if the data is normally distributed or not.

Hypothesis Testing \rightarrow

null \rightarrow no change.
alternative \rightarrow claimed change.

$P(\text{alternative} / \text{null})$ \rightarrow conditional probability.

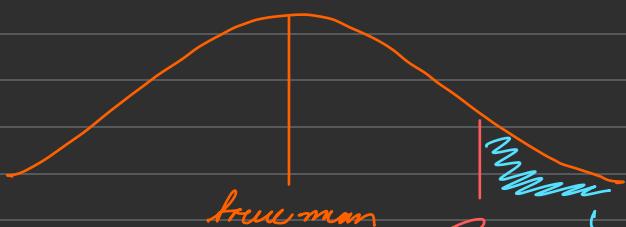
\hookrightarrow low \rightarrow reject null hypothesis.

	true h_0	
	+	-
same h_0	+ \rightarrow Type - I	
		- \rightarrow Type - II

Expt Model

burger < 200g \rightarrow alternative

burger $\geq 200g$ \rightarrow null



$$\text{P-value} = 1 - P(z \text{ zone})$$

$P(\text{alternative} / \text{null})$

P-value
 $P(\text{alternative} / \text{null})$

high

Conclusion

occurrence of alternate hypothesis is common.
- keep h_0 .
- alternate had less contribution.

low

occurrence of alternate is rare w.r.t null.
- reject h_0 .

- null had a lot of contribution to this variety.

14.14

$$Z = \frac{10/14.14 - 7.07}{2.23}$$

$$P(\text{alt(null)}) = 0.012 \quad \alpha = 0.01$$

- keep to.

$$\begin{array}{l} Q \neq \\ \mu = 65 \\ \sigma = 2.5 \end{array} \quad \left| \begin{array}{l} n = 20 \\ \bar{x} = 69.5 \\ 0.05 \end{array} \right.$$

$$Z = \frac{69.5 - 65}{2.5/\sqrt{20}} = -0.89$$

$$P(Z) = 0.186 \quad 0.05$$

Two sample test

$$Z = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

μ_1 → mean {sample 1}
 σ_1 → std.dev {sample 1}
 n_1 → sample-size {sample 1}

μ_2 → mean {sample 2}
 σ_2 → std.dev {sample 2}
 n_2 → sample-size {sample 2}

Proportion Test

$$Z = P - \bar{P}$$

Z score for proportions. $\sqrt{P(1-P)/n}$ } → standard error.

Two sample →

$$Z = \frac{\hat{P}_1 - \hat{P}_2}{\sqrt{\hat{P}(1-\hat{P}) \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

Combined proportion.

Paired T Test & Dependent

thistest(df[group1], df[group2])

import thistest from teststats

χ^2 - test → Check if the actual values differ significantly from expected values.

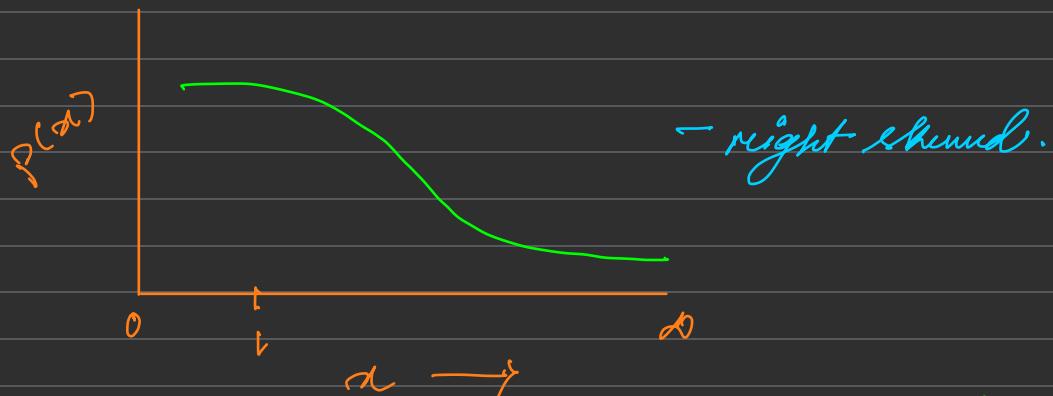
Coin toss

	Actual	Expected	
Heads	25	25	— no std. deviation.
Tails	28	22	— uses frequency.

H_0 : Coin is fair.

H_1 : Coin is not fair.

$$\chi^2 = \sum \frac{(actual_i - expected_i)^2}{expected_i}$$



→ right-skewed.

→ formula implementation.

from scipy.stats import chi2square, chi2
→ pdf & rpdf.

chi_stat, p_value = chi2square(actual, expected)

P-val = 1 - chi2.pdf(chi_stat, df-1)

→ Compute Expected values using marginal values.

χ^2 - two category & If one category affects the other.

Ch-T

	M	W		M	W
offline	527	72	509 66%	489	115
online	206	102	308 34%	(66% & 33%)	(66% & 18%)
	733	174	907	269	59

Actual.

Expected

from scipy.stats import chi2_contingency

chi2stat, pval, dof, emp = chi2_contingency(actual-values)
 \uparrow expected frequency.
 \uparrow degree of freedom.

Co-correlation &

χ^2 , & \rightarrow one/two numerical categories.
 \rightarrow n/a known.

$\gamma^2 \rightarrow$ one/two categorical categories. \rightarrow non-numerical

Correlation \rightarrow Two numeric categories.

- how two values are related to each other.

Given x_i, y_i :

$$\text{cov}(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n}$$

$$\text{correlation} \& \text{ corr}(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x, y)}}$$

Hypothesis Testing Revision

1) Z-test → To test if data from samples differ significantly from a prior belief/claim. + single-group

1) one-sample → To test if sample measurements differ significantly from a known population measurement.

$$Z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \quad \{ \rightarrow \text{standard-error}$$

2) Two-Sample → To test if the mean of two groups from a test statistic differ significantly or not.

$$Z = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Null & Alternate Hypothesis →

null (H_0) → A natural belief about data.
alternate (H_{alt}) → A claim on the data.



Z-score → # of std. deviations away from the mean.

Z-score

P-val ($P(\text{alternate}/\text{null})$)

reject H_0

high

low

Yes, as the observed data is too far away from expected data.

low

high

No, observed data is close to expected data.

2) T-test -> Similar to Z-test but uses sample statistics instead of population.

use when

1) Population std. deviation is unknown.

2) Sample size is comparatively small. ($n \leq 30$)

3) Samples are independent (from two samples).

a) one-sample -> To check if the sample mean from a measurement is similar to population mean.

- population std. dev (σ) is unknown.

t-score =

$$\frac{\bar{x} - \mu}{\text{ }}$$

$\left(\frac{s}{\sqrt{n}} \right)$ } t sample std. error.

} Sample std. deviation.

b) Two-Sample -> To check if the mean for a given statistic/measurement is similar b/w two groups.

- σ is unknown for both the groups.

- two groups are independent.

t-score =

$$\frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

} t sample std. error.

3) Proportions test -> To check if the ratios for a binary-measurement are similar for sample & population.

Requirements ->

1) Binary variable.

2) Independent observations.

3) Sample-size adequacy -> If one sample -> $n(P_0) \geq 10$ & $n(1-P_0) \geq 10$

2) Two-Sample -> $n_1 P_1 \geq 10$ & $n_1(1-P_1) \geq 10$ } Population proportion.
 $n_2 P_2 \geq 10$ & $n_2(1-P_2) \geq 10$

by one-sample χ^2 Test if the proportion in a binary measurement is similar to claimed proportion or not.

$$Z\text{-Score} = \frac{\hat{P} - P_0}{\sqrt{\frac{P_0(1-P_0)}{n}}} \rightarrow \text{Sample proportion.}$$

\rightarrow Sample size.

by Two-sample χ^2

$$R_1 = \frac{x_1}{n_1}$$

$$R_2 = \frac{x_2}{n_2}$$

$$P = \frac{x_1 + x_2}{n_1 + n_2}$$

$$\frac{R_1 - R_2}{\sqrt{P(1-P)(\frac{1}{n_1} + \frac{1}{n_2})}}$$

\rightarrow Combined proportion.

W/ χ^2 test \rightarrow (chi-square)

$$\chi^2 = \sum \frac{(expected_i - actual_i)^2}{expected_i}$$

χ^2
low
high

P-val

high χ^2
low

chi-square dist.



Used for χ^2

W/ goodness of fit \rightarrow Check if the freq. count of two categorical variables differ significantly or not.

on & Test if a var is pure by comparing the observed freqs to expected frequencies.

W/ Test for independence \rightarrow To test if two categorical variables are independent.

on & Test if political views were related to demographics or gender. \rightarrow Categorical variable.

5) Anova & Analysis of variance.

To check if the mean of multiple categorical variables differ with one another or not.

<u>Motile</u>	<u>Non-motile</u>
apple	X
lansium	Y
oppo	Z

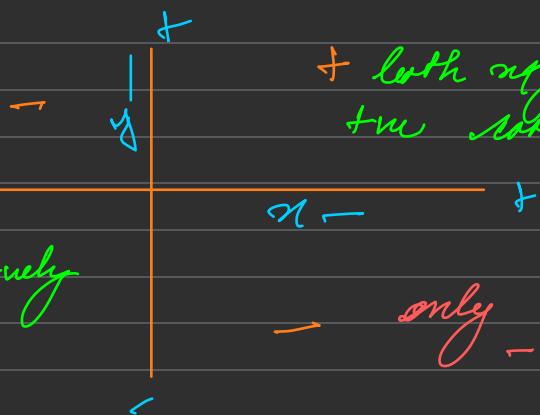
→ To check if the means of the three phone categories vary with each other or not.

6) Correlation &

logit &

If the least $x = -ve$,
only correlated.

both x & y are -vely
away from the mean.
the correlation.



+ both x & y are +vely away from mean.
+ve correlation.

→ only x is the y is -ve.
-vely correlated.

$$\text{cov}(x, y) = \frac{\text{var}(x, y)}{\sigma_{xy}}$$

σ_{xy} or joint std. dev.

$$\text{var}(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

