

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=fWntfXullWPA

+ Code + Text

RAM Disk

Task: Determine the eligibility for granting Home loan.

Objective of this notebook is:

- 1. To understand the patterns in the data.
- 2. How to Handle the categorical features.
- 3. How to deal with missing data.
- 4. Feature Engineering
- 5. Finding the most important features while taking the decision of granting a loan application.
- 6. Understanding the Normalization and standardisation of the data.

Plotting
Postressals:-
→ Save simple 'DS'

Load data and libraries

```
[1]: import numpy as np
import pandas as pd
from scipy import stats

import matplotlib.pyplot as plt
import seaborn as sns
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=fWntfXullWPA

+ Code + Text RAM Disk

Task: Determine the eligibility for granting Home loan.

Objective of this notebook is:

1. To understand the patterns in the data.
2. How to Handle the categorical features.
3. How to deal with missing data.
4. Feature Engineering
5. Finding the most important features while taking the decision of granting a loan application.
6. Understanding the Normalization and standardisation of the data.

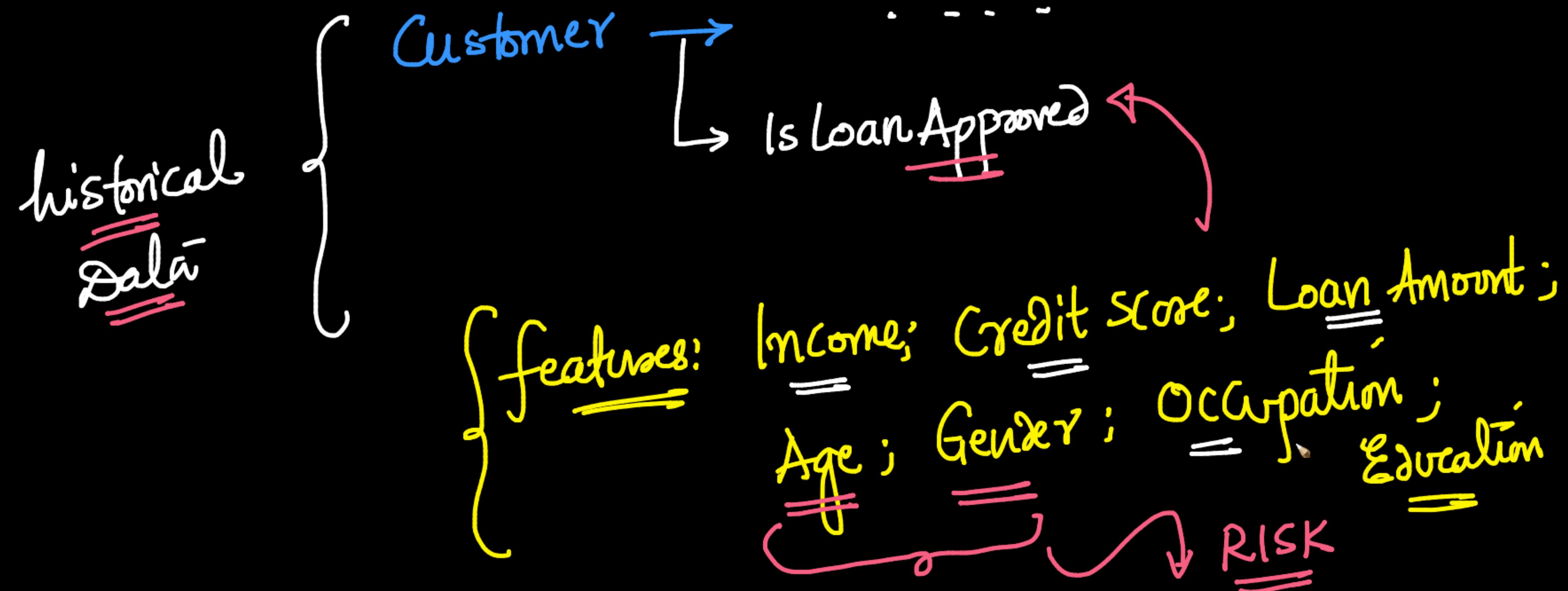
Load data and libraries

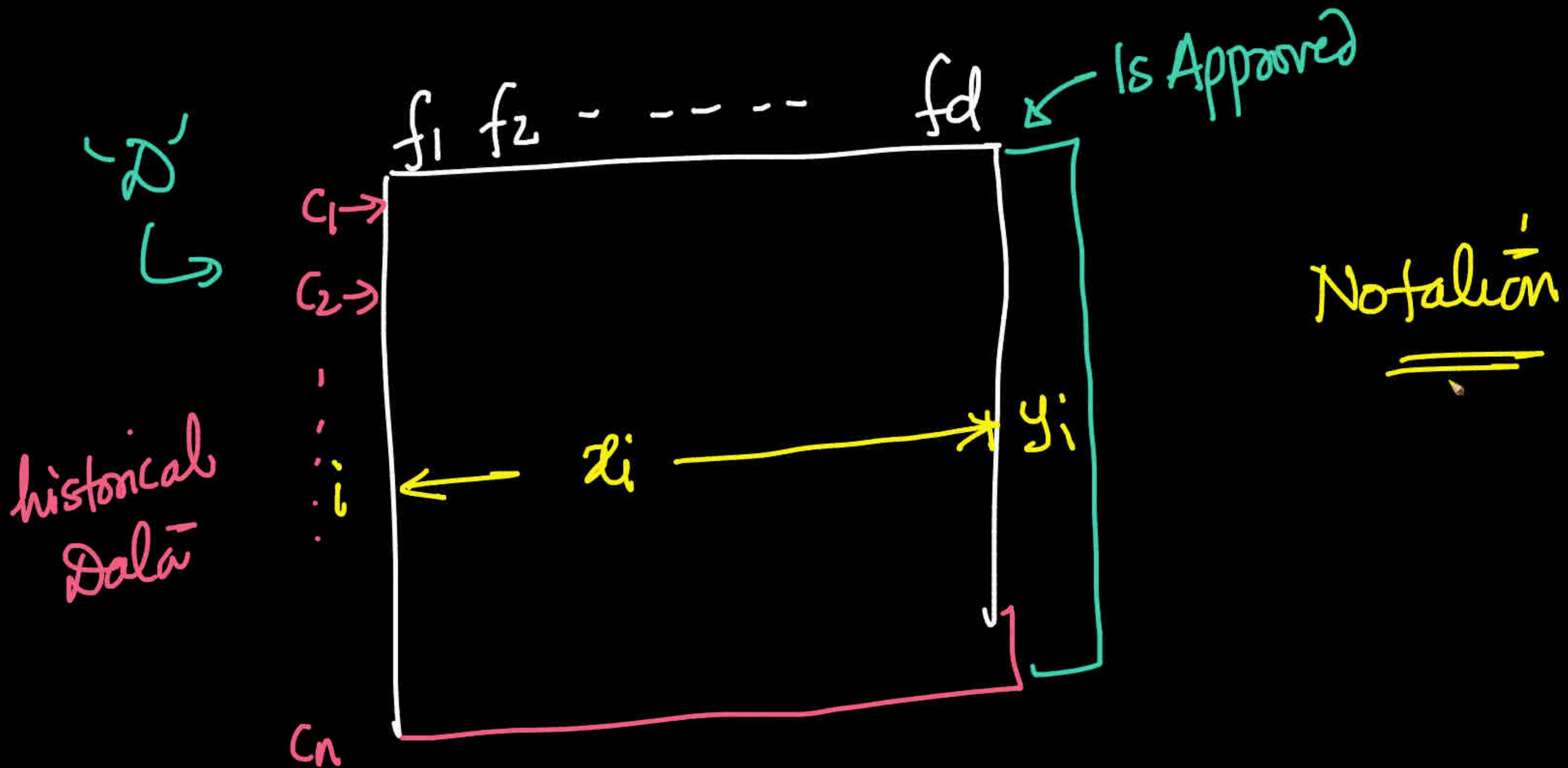
```
[1]: import numpy as np  
import pandas as pd  
from scipy import stats  
  
import matplotlib.pyplot as plt  
import seaborn as sns
```

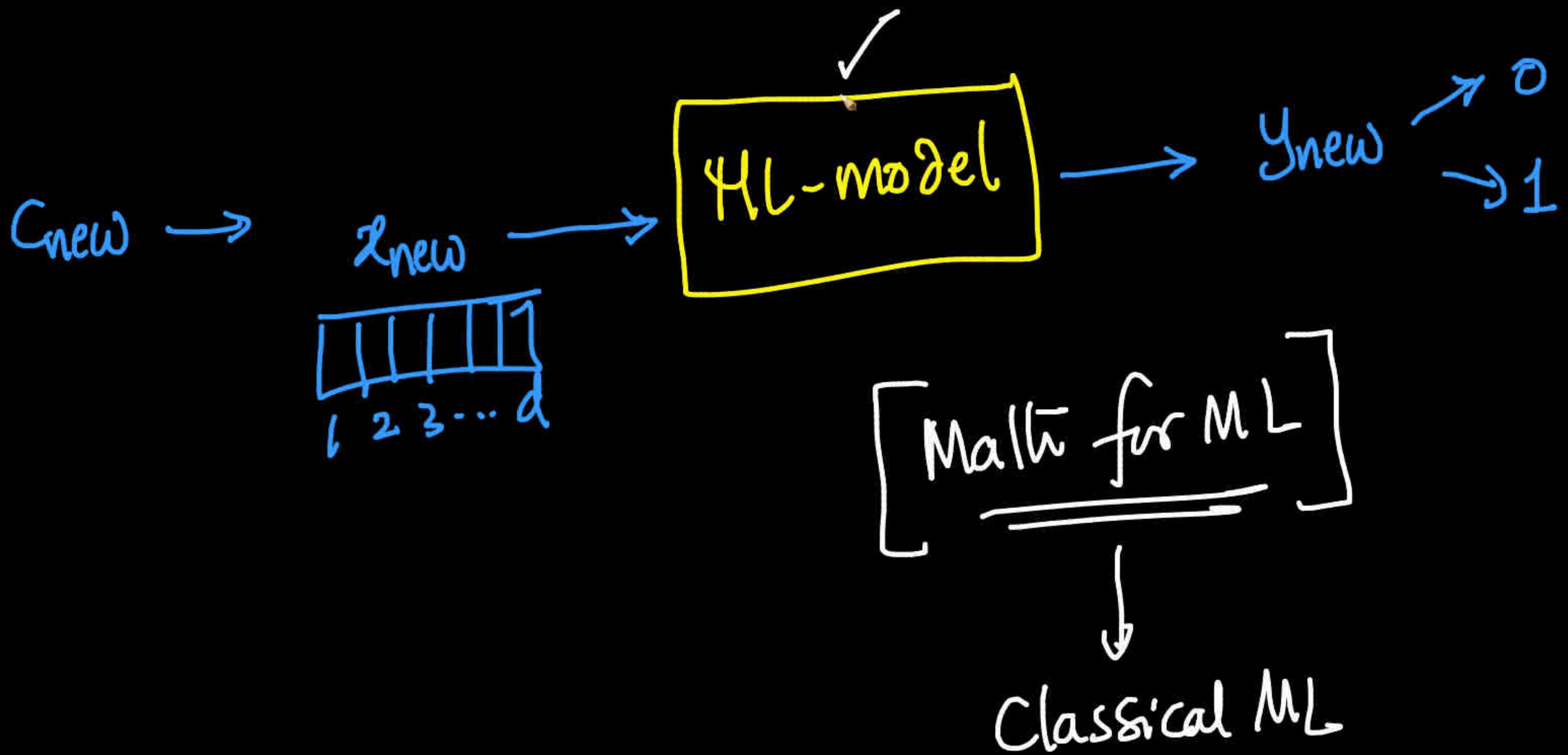
2 / 2

Home-loan!

Variables { features







EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Ej8pWNLIGL-i

+ Code + Text RAM Disk

LP001267, Female, Yes, 2, Graduate, No, 1378, 1881, 167, 360, 1, Urban, N
LP001273, Male, Yes, 0, Graduate, No, 6000, 2250, 265, 360, , Semiurban, N

{x} { } []

[] data = pd.read_csv('./train.csv')
data.shape

(614, 13)

[] data.columns

Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
 'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
 'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
 dtype='object')

[] data.head()

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loa
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Not	2	Not Graduate	Yes	2583	2358.0	120.0	

6/7

 EDA_FE.ipynb - Colaboratory X

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Ej8pWN

◎ 内 容 索 引

+ Code + Text

 RAM Disk

V

LP001267, Female, Yes, 2, Graduate, No, 1378, 1881, 167, 360, 1, Urban, N

LP001273, Male, Yes, 0, Graduate, No, 6000, 2250, 265, 360, , Semiurban, 1

A set of small, light-gray navigation icons located at the bottom right of the page. From left to right, they include: a downward arrow, a circular arrow, a speech bubble, a gear, a square with a diagonal line, a trash can, and three vertical dots.

```
▶ data = pd.read_csv('./train.csv')
    data.shape
```

(614, 13)

[8] data.column

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

▶ `data.head()`

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loa
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	2	Not Graduate	No	2583	2358.0	120.0	

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Ej8pWNLIGL-i

+ Code + Text RAM Disk

LP001267, Female, Yes, 2, Graduate, No, 1378, 1881, 167, 360, 1, Urban, N
LP001273, Male, Yes, 0, Graduate, No, 6000, 2250, 265, 360, , Semiurban, N

{x} [8] data.columns

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',  
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',  
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],  
      dtype='object')
```

[8] data.head()

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loa
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	2	Not Graduate	No	2583	2358.0	120.0	

RAM Disk

8/9

Chrome File Edit View History Bookmarks Profiles Tab Window Help

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Ej8pWNLIGL-i

+ Code + Text RAM Disk

LP001267, Female, Yes, 2, Graduate, No, 1378, 1881, 167, 360, 1, Urban, N
LP001273, Male, Yes, 0, Graduate, No, 6000, 2250, 265, 360, , Semiurban, N

{x} [8] [9]

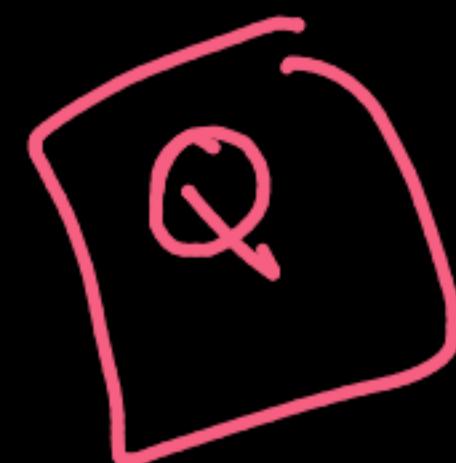
```
data = pd.read_csv('./train.csv')
data.shape
(614, 13)

[8] data.columns
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')

[9] data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loa
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	2	Not Graduate	No	2583	2358.0	120.0	

9/10

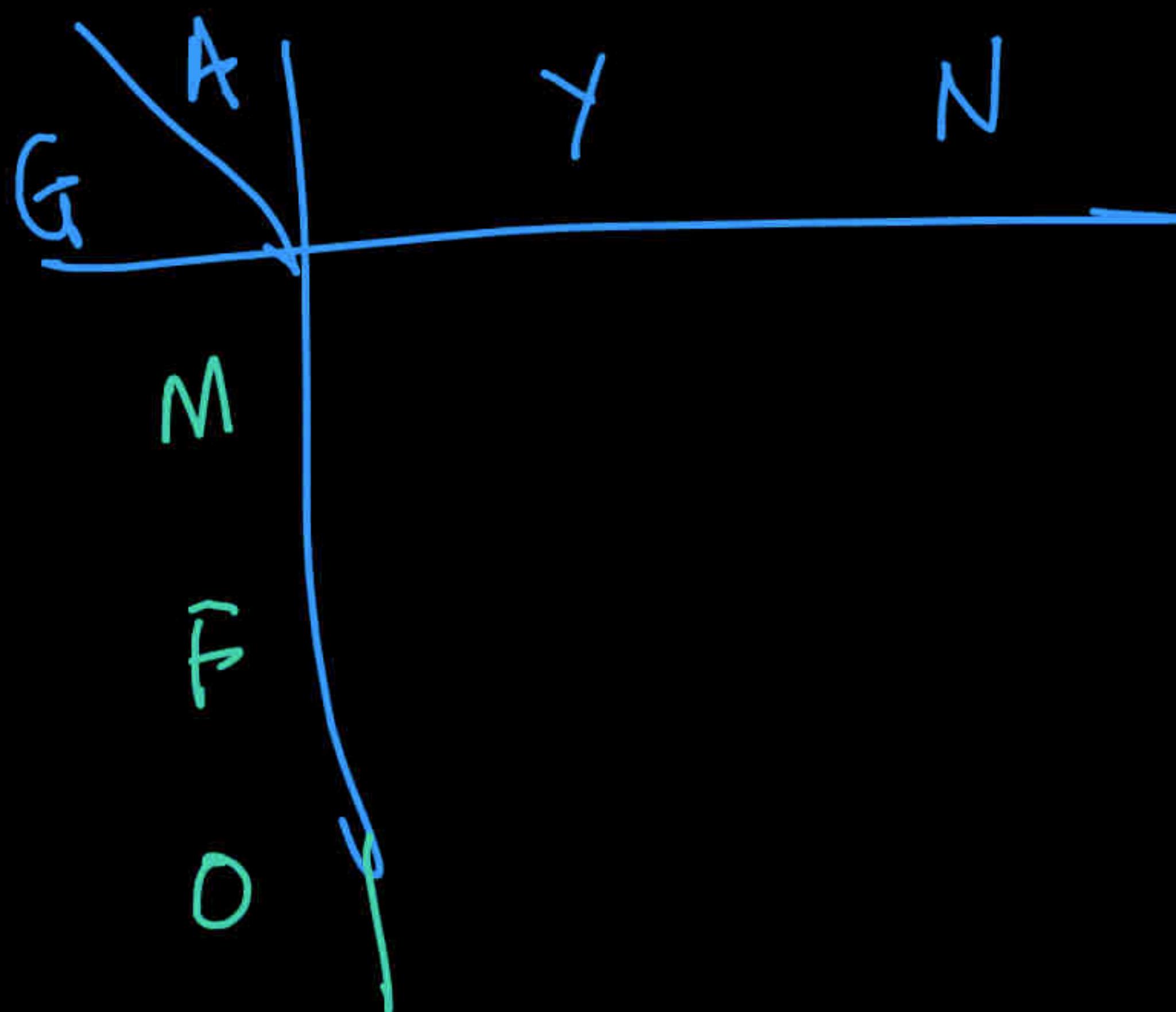


statis
Indep. columns → ML features

↳ Loan - Approved
(cat) ↗ Y
↗ N

Gender
(cat) ↗ M
↗ F
↗ O

→ correlation
→ chi-square test

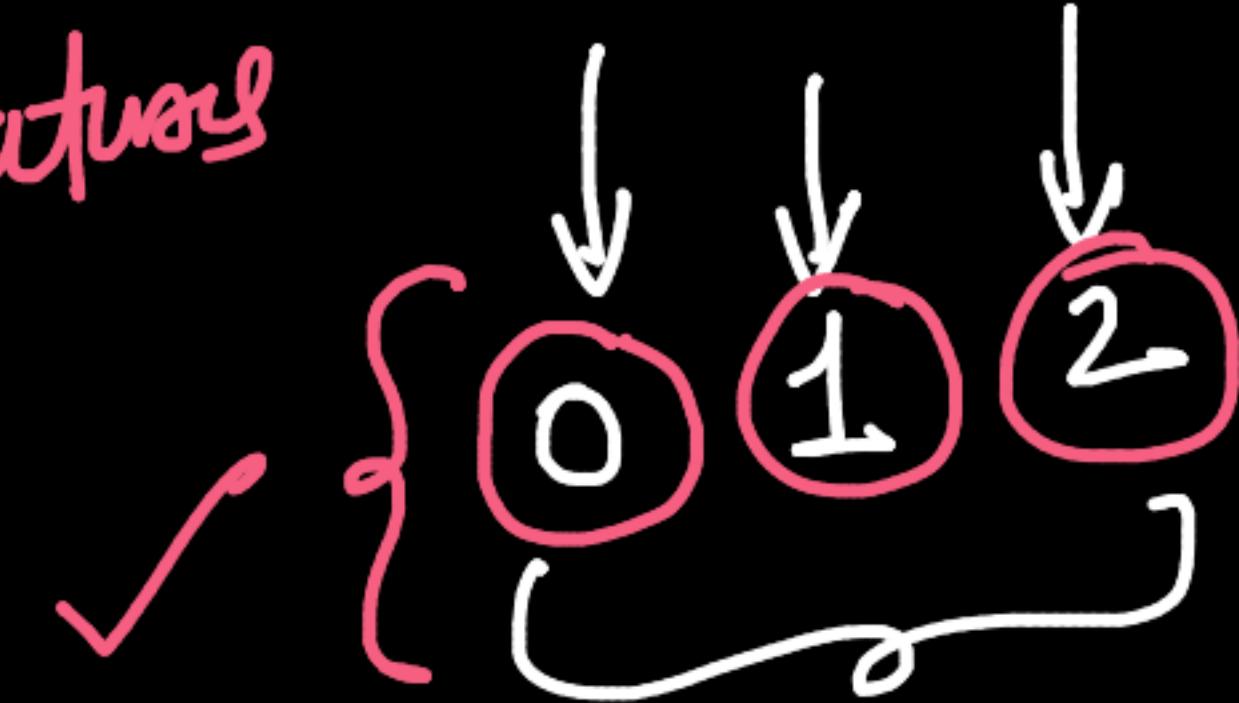


G indep of
Approval or
not?

Gender

non-ordinal
M, F, O

features



numerical

not-an ideal

Loan-status

Y N

↓ ↓

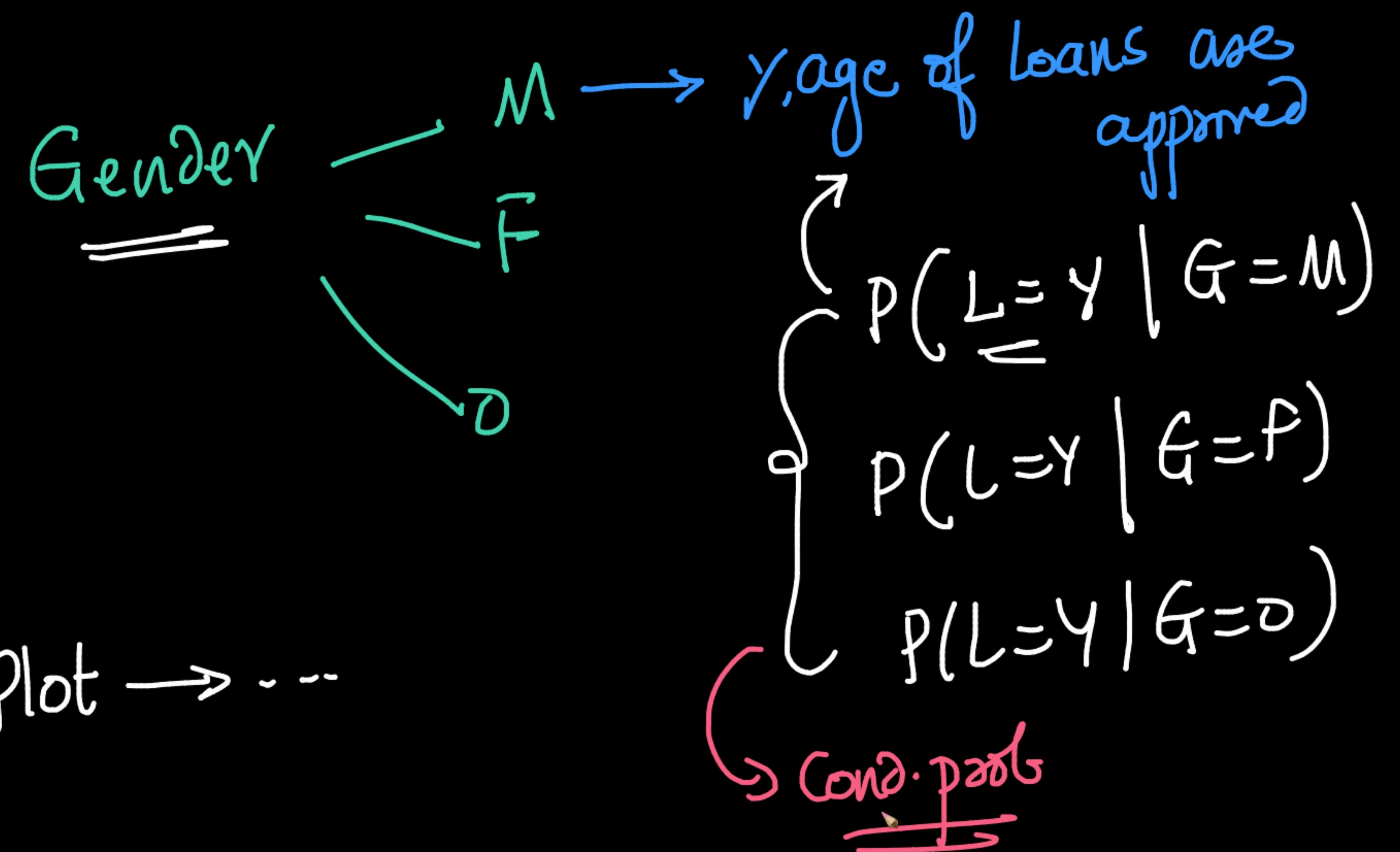
1 0

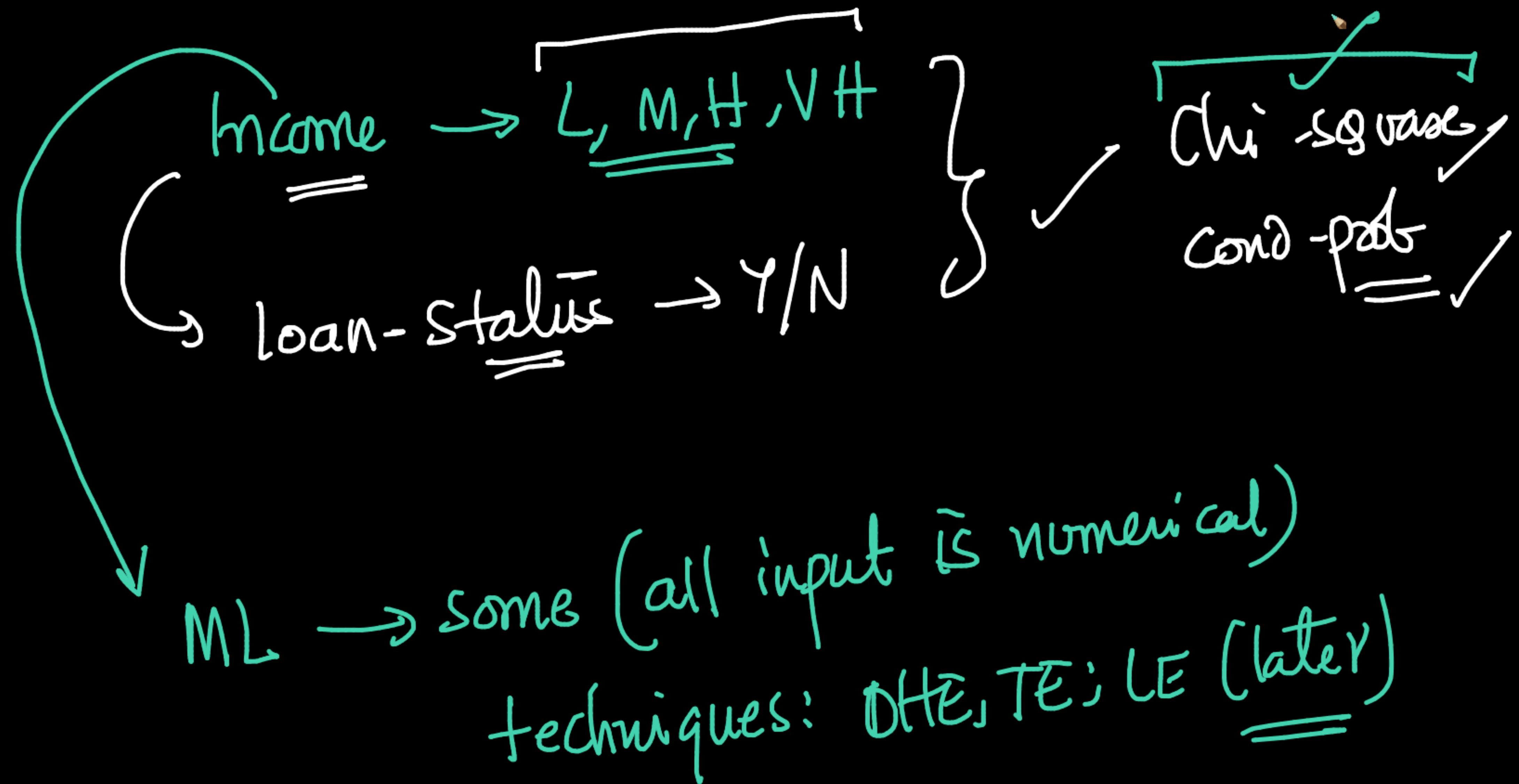
~

numerical

SRCC or PCC ?

High-slash





 EDA_FE.ipynb - Colaboratory X

```
[10] data.dtypes  
#object => typically categorical/IDs  
#Int64, Float64
```

ID	object
Gender	object
Married	object
Dependents	object
Education	object
Self_Employed	object
ApplicantIncome	int64
CoapplicantIncome	float64
LoanAmount	float64
Loan_Amount_Term	float64
Credit_History	float64
Property_Area	object
Loan_Status	object
dtype:	object

غایبی غایب
عدالتی ایجاد کننده

```
✓ [11] # drop loanID column  
data = data.drop('Loan_ID', axis = 1)
```

EDA_FE.ipynb - Colaboratory + colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=DEyPpdvDGfFI

+ Code + Text RAM Disk

[10] data.dtypes
{x} #object => typically categorical/IDs
#Int64, Float64

Loan_ID	object
Gender	object
Married	object
Dependents	object
Education	object
Self_Employed	object
ApplicantIncome	int64
CoapplicantIncome	float64
LoanAmount	float64
Loan_Amount_Term	float64
Credit_History	float64
Property_Area	object
Loan_Status	object
dtype:	object

[11] # drop loanID column
data = data.drop('Loan_ID', axis = 1)

Basic Data Exploration

16 / 16

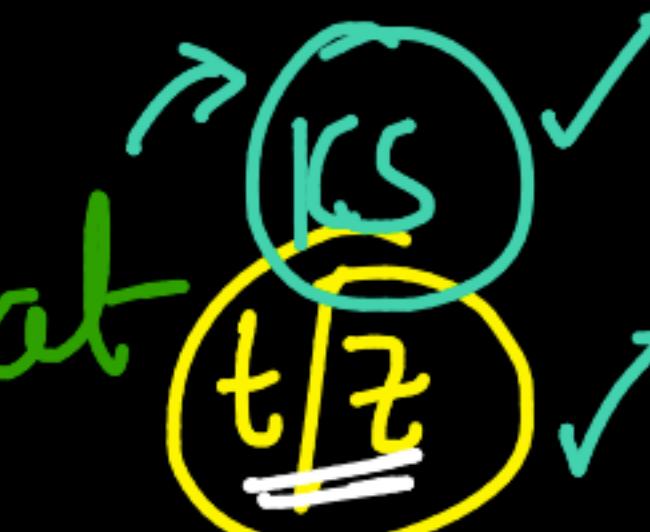
(Q)

f_j : cond. pop

f_j : cat & y : cat

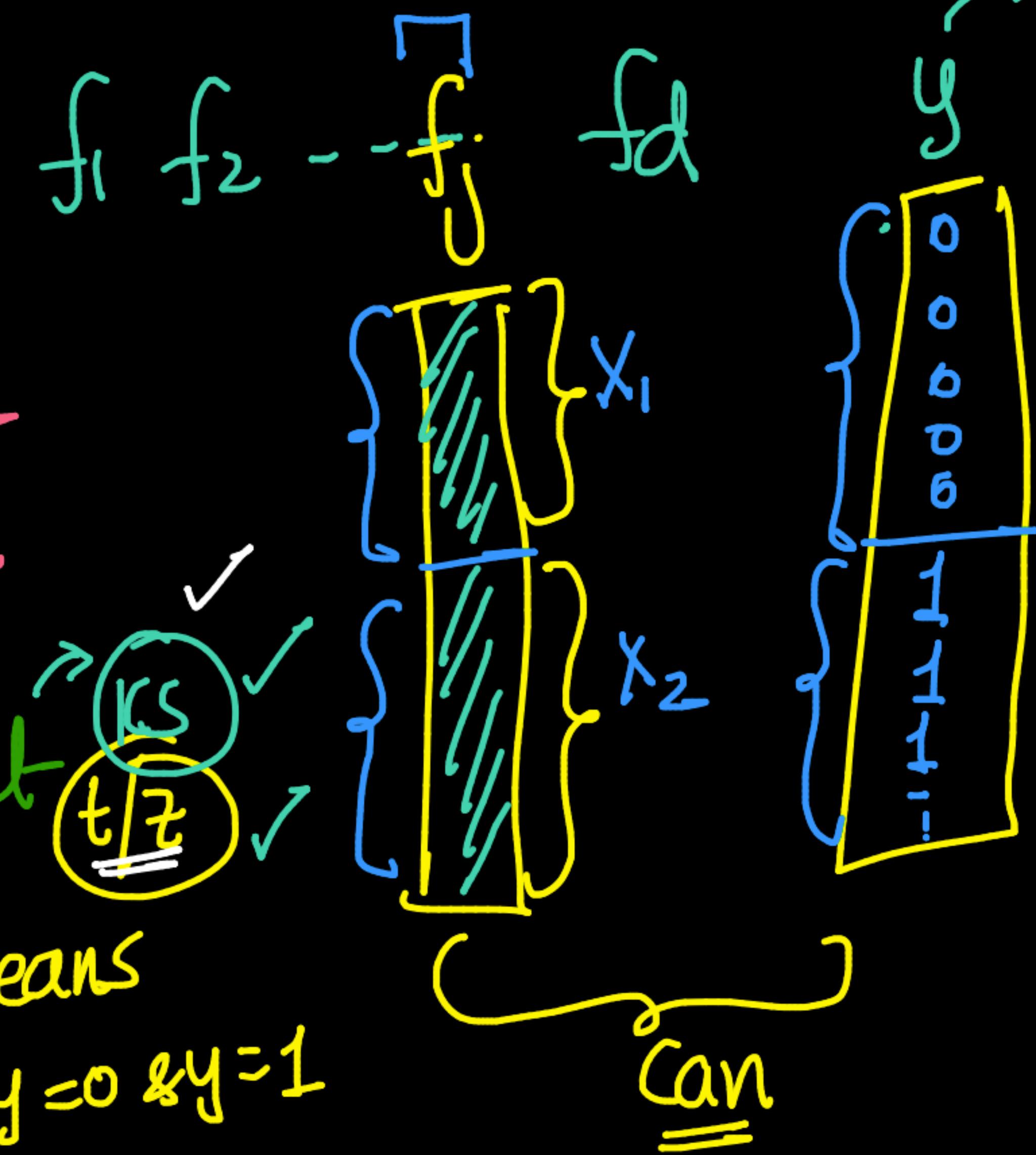
$\hookrightarrow \chi^2$ -test

f_j : num & y : cat



\hookrightarrow comp of means

w/\bar{h} $y=0$ & $y=1$

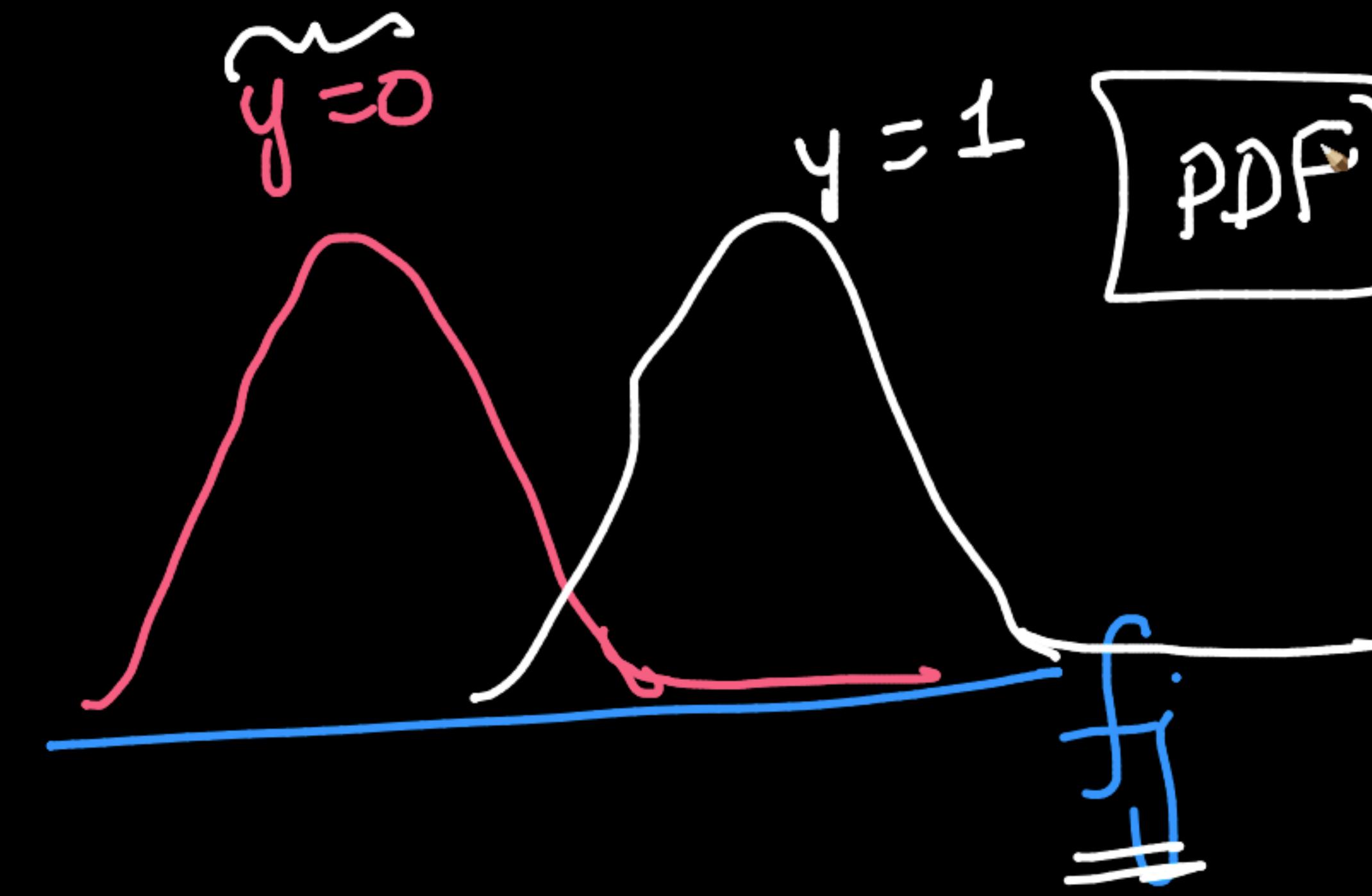


Is loan Approved

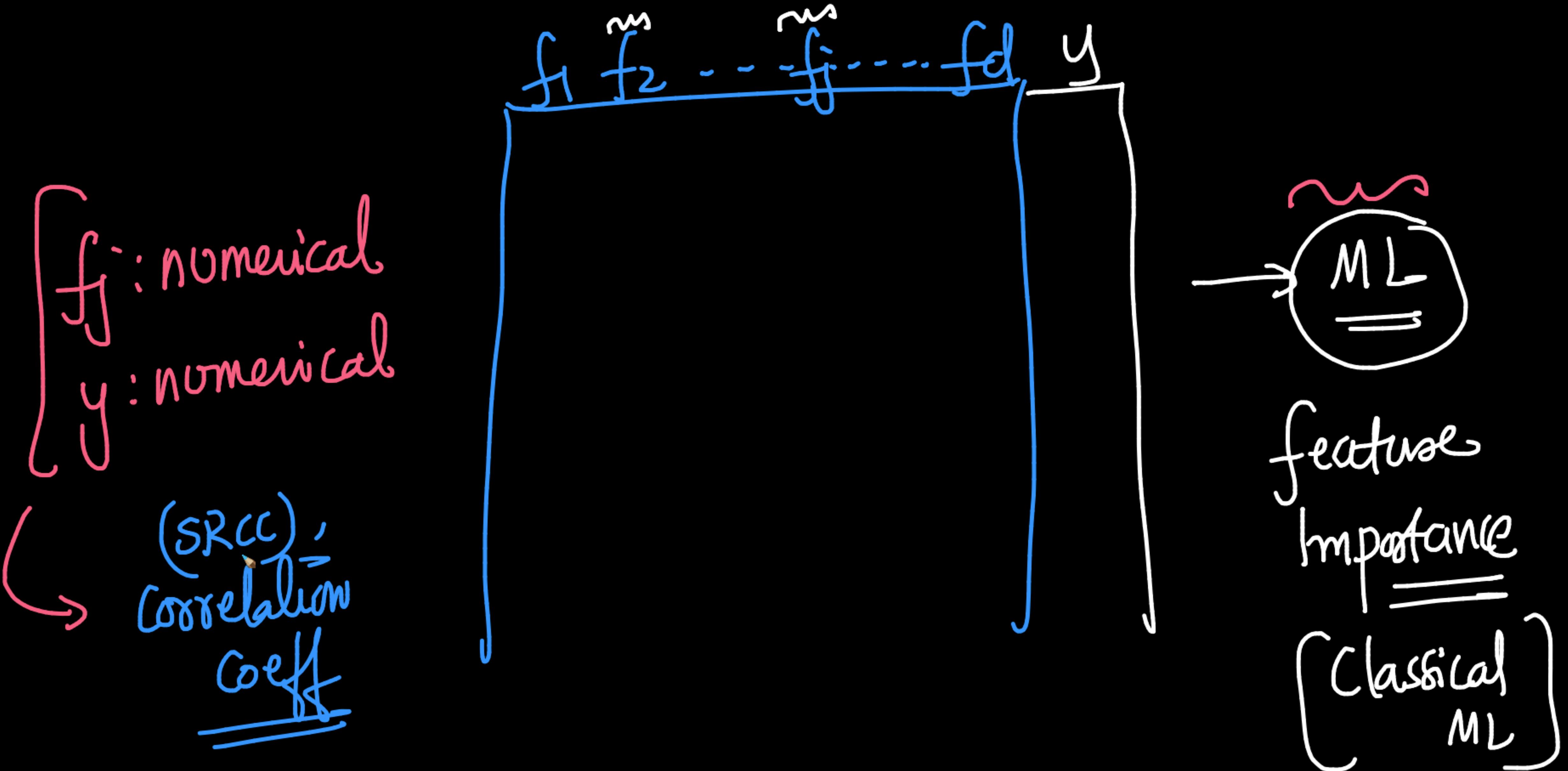
$P(f_j | y=0)$

$\neq P(f_j | y=1)$

KS-test:
↓
CDFs



ANOVA: → Pop. per gp are Gaussian
↳ var across gps is same



Q
fj

income
fj: numeric

loan-status
y: categorical
y
N

X1 Y
fj → n1

X2 N
fj → n2

Compose Means

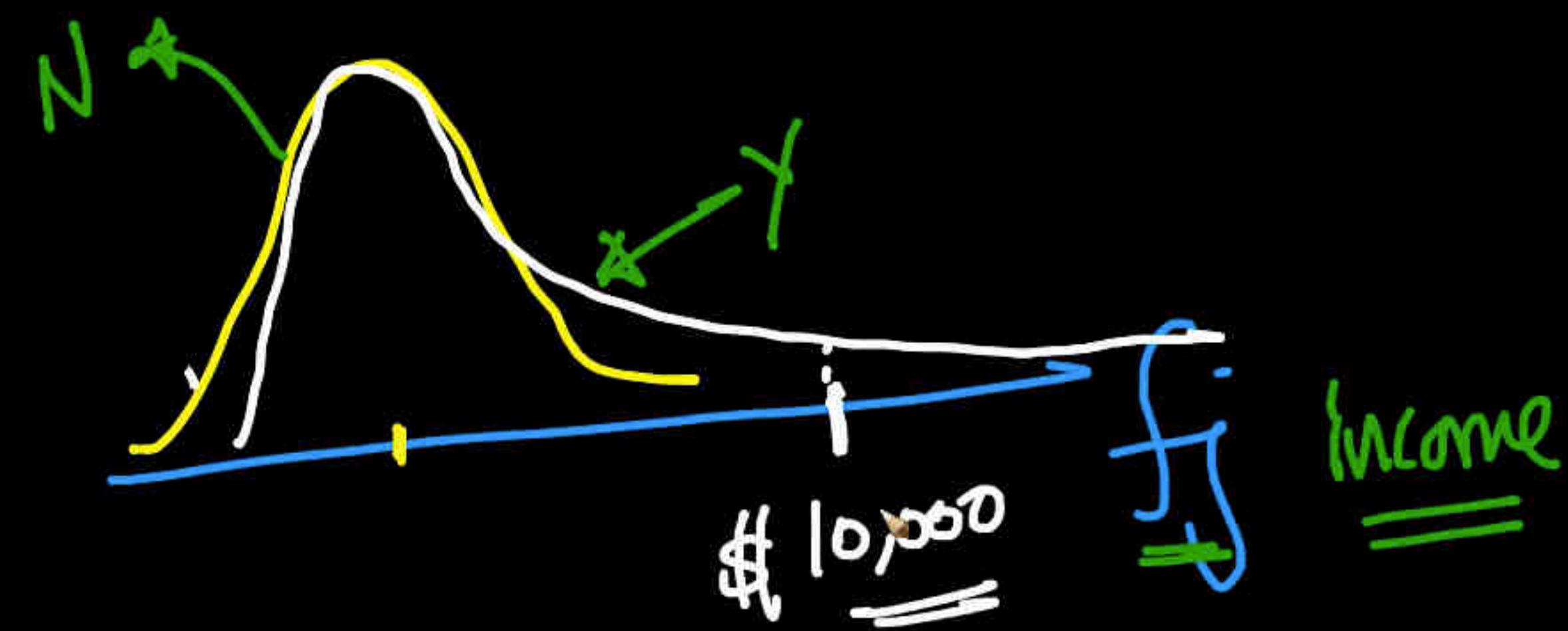
= { t-test →

n1 & n2 are small

z-test → n1 & n2 > 30

{ KS-test →

↳ compare dist: More powerful



n_1 & n_2 are < 30

σ_1 & σ_2 are known

\rightarrow Z-test

n_1 & $n_2 > 30$

σ_1 & σ_2 are not known

estimated

\rightarrow Z-test

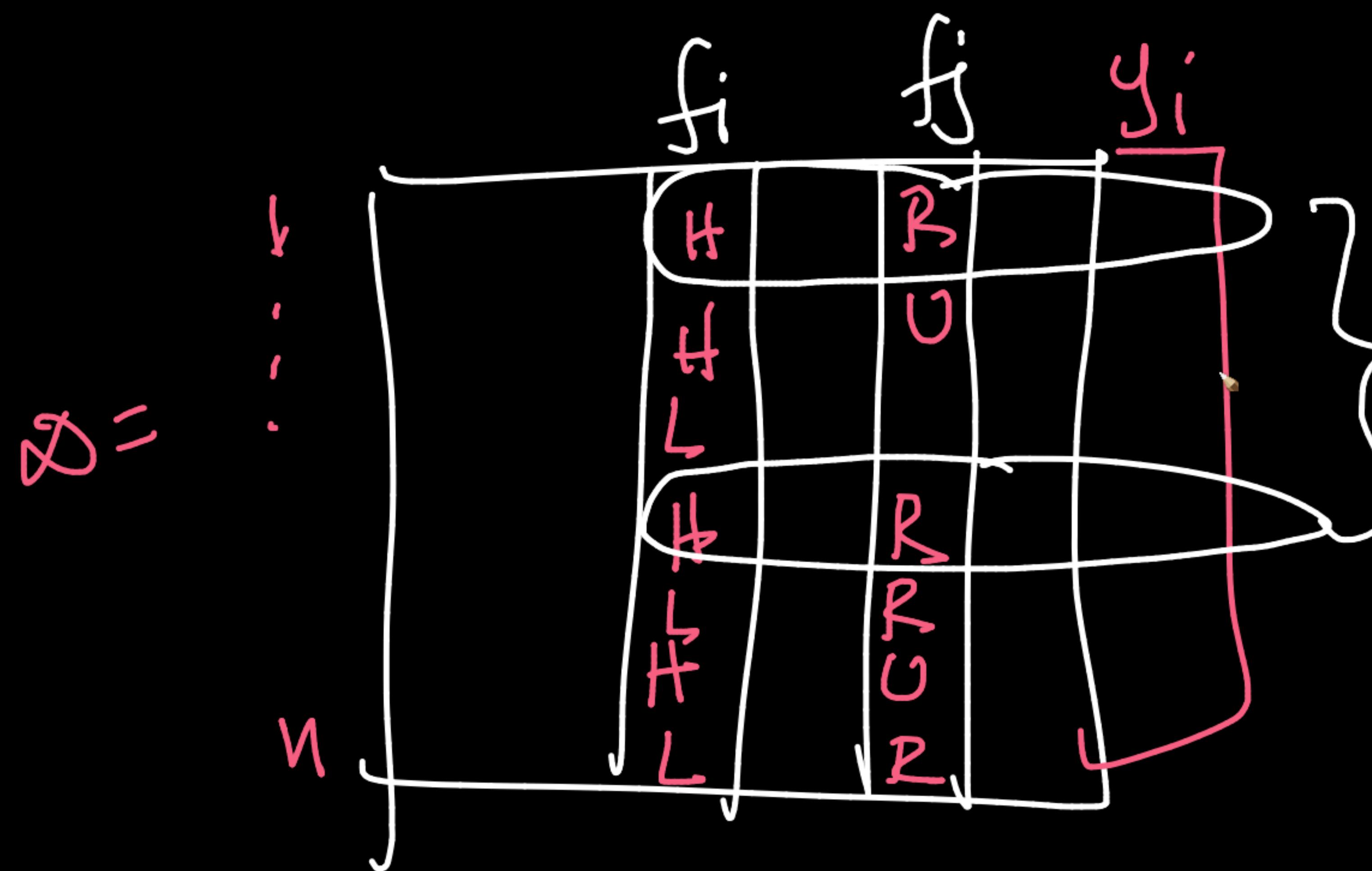
σ_1 & σ_2 are not known } \rightarrow t-test
 n_1 & n_2 are small }

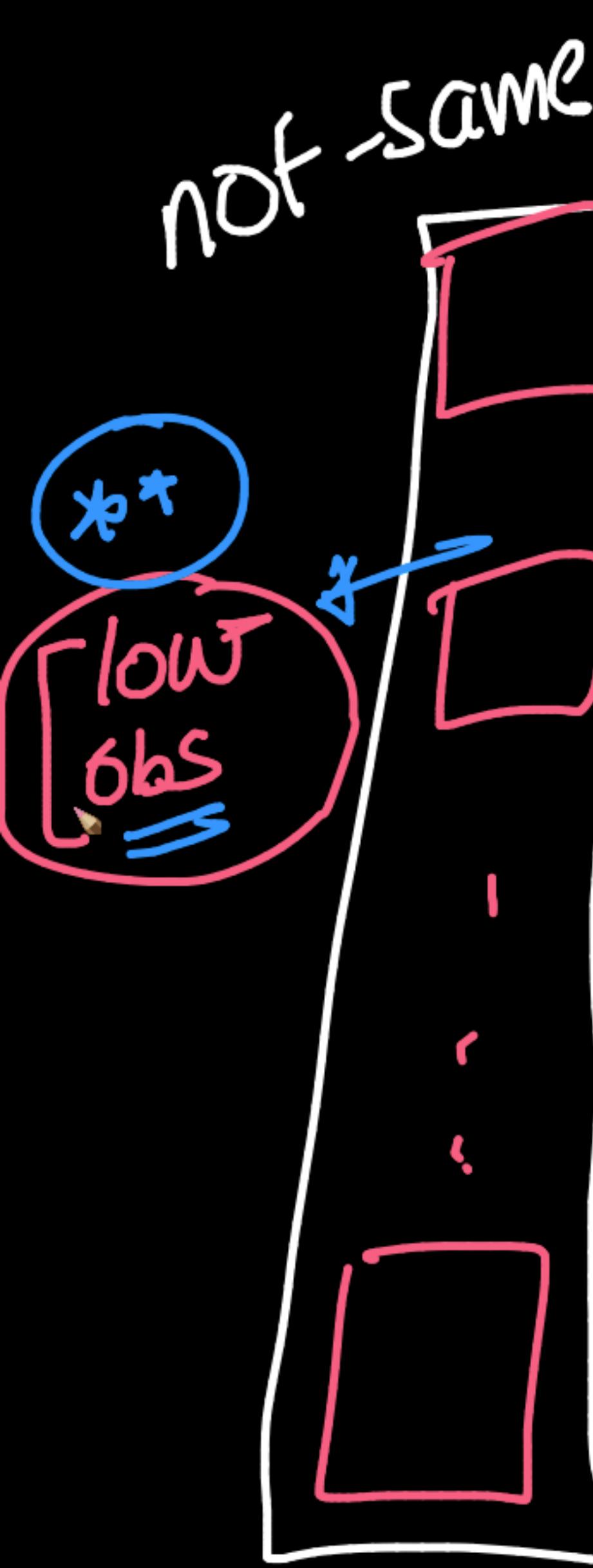
σ_1 & σ_2 are not known } \rightarrow t-test
 n_1 & n_2 are large } \rightarrow z-test

income (cat)
property
loc(U, SU, R) Feature Combinations

f_i f_j

$$P(y=1 \mid \underbrace{f_i = H}_{\text{INCOME}} \text{ and } \underbrace{f_j = R}_{\text{loc}})$$





$$\begin{aligned} \text{not same} &= P(Y=1) \left[\underbrace{\text{INC} = H}_{\text{INC} = UH} \text{ and } \underbrace{\text{LOC} = R}_{\text{LOC} = SU} \right] \\ &= P(Y=1) \end{aligned}$$

one-line \Rightarrow

$$= \frac{1}{5}$$

$f_1 f_2 \dots f_d$

1 feat $\rightarrow d$

2 feat $\rightarrow d_{c_2}$

3feat $\rightarrow d_{c_3}$

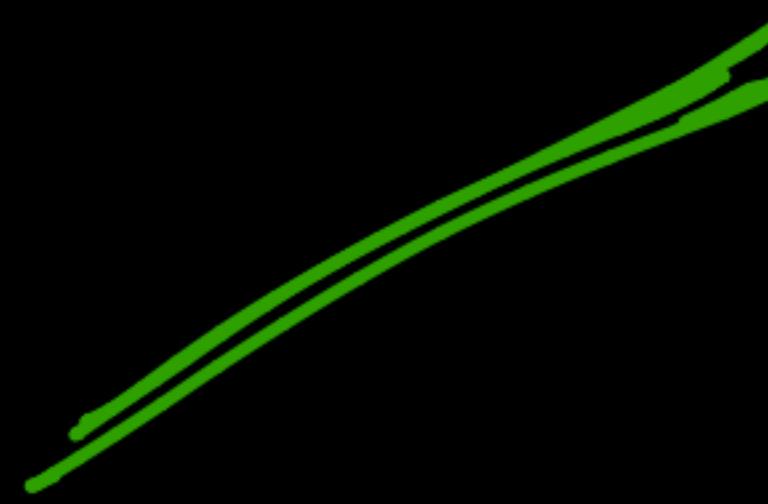
:

d-feat $\rightarrow d_d$

$d=10$

$2d=1024$

$2d$



ML-algo:

GBDT & RF



{ automatically
discover
good combinations of
features



$$n_1 = \underline{85}$$

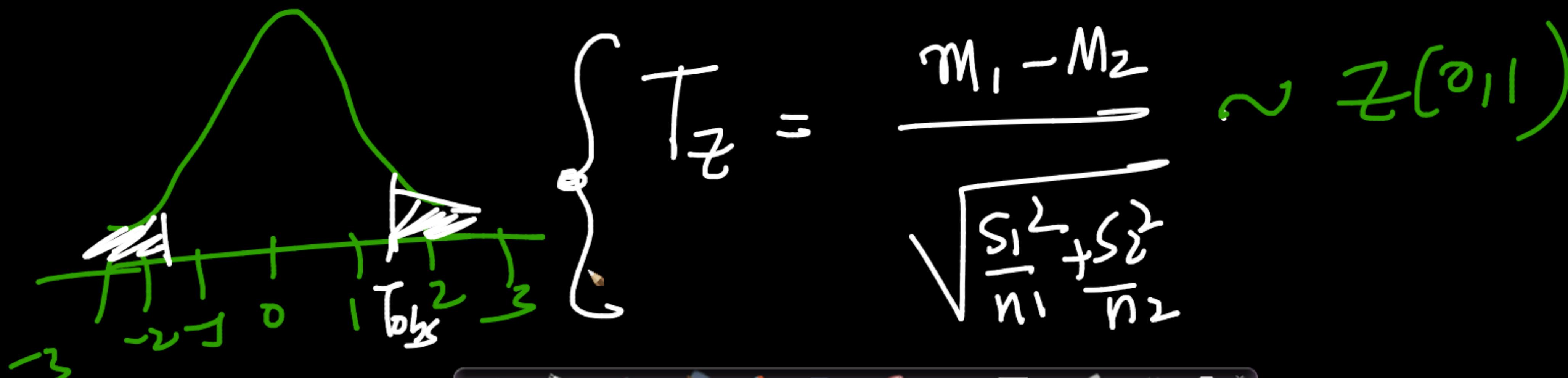
Task
 $H_0: \mu_1 = \mu_2$
 $H_a: \mu_1 \neq \mu_2$

Z-test

$$n_2 = 46$$

$$\begin{array}{ll} m_1 = \underline{\underline{83}} & s_1 = \underline{\underline{8}} \\ \text{old data} & \checkmark \\ \cancel{\text{new data}} & \times \end{array}$$

$$m_2 = \underline{\underline{88}} \quad s_2 = \underline{\underline{12}}$$



EDA_FE.ipynb - Colaboratory

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=f5n7flegGQEe



+ Code + Text

✓ RAM Disk

```
[12] data.describe()  
    # only numeric features
```

~~Missing-data~~

~~nan~~)

~~Nan~~

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.000000	564.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

```
# categorical features  
data.describe(include = ['object'])
```

```
Gender Married Dependents Education Self_Employed Property_Area Loan_Status
```

count

601

611

614

614

EDA_FE.ipynb - Colaboratory

[12]	25%	2877.500000	0.000000	100.000000	360.000000	1.000000
	50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
	75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
	max	81000.000000	41667.000000	700.000000	480.000000	1.000000

RAM Disk

A set of small, light-gray navigation icons located at the bottom right of the page. From left to right, they include: a downward arrow, a circular arrow, a speech bubble, a gear, a square with a diagonal line, a trash can, and three vertical dots.

```
✓ 0s # categorical features  
data.describe(include = ['object'])
```

```
[16] #missing values  
      data.isna().sum()
```

Gender 1
Married
Dependents 1

EDA_FE.ipynb - Colaboratory

- Code ≠ Text

✓ RAM Disk

25%	2877.500000	0.000000	100.000000	360.000000	1.000000
[12]					
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

A set of small, light-gray navigation icons located at the bottom of the page. From left to right, they include: a double arrow pointing up and down, a circular arrow, a magnifying glass, a gear, a square with a diagonal line, a trash can, and three vertical dots.

```
# categorical features  
data.describe(include = ['object'])
```

	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
count	601	611	599	614	582	614	614
unique	2	2	4	2	2	3	2
top	Male	Yes	0	Graduate	No	Semiurban	Y
freq	489	398	345	480	500	233	422

```
[16] #missing values  
      data.isna().sum()
```

Gender 1
Married
Dependents 1

EDA_FE.ipynb - Colaboratory

	25%	2877.500000	0.000000	100.000000	360.000000	1.000000
[12]	50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
	75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
	max	81000.000000	41667.000000	700.000000	480.000000	1.000000

RAM Disk

A small white gear icon located in the bottom right corner of the slide.

1

```
# categorical features  
data.describe(include = ['object'])
```

	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
count	601	611	599	614	582	614	614
unique	2	2	4	2	2	3	2
top	Male	Yes	0	Graduate	No	Semiurban	Y
freq	489	398	345	480	500	233	422

```
[16] #missing values  
      data.isna().sum()
```

Gender 1
Married
Dependents 1

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfHCG7

+ Code + Text ✓ RAM Disk

[12]

25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

{ } { }

0s

catgeorical features

data.describe(include = ['object'])

Gender Married Dependents Education Self_Employed Property_Area Loan_Status

	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
count	601	611	599	614	582	614	614
unique	2	2	4	2	2	3	2
top	Male	Yes	0	Graduate	No	Semiurban	Y
freq	489	398	345	480	500	233	422

[16] #missing values

data.isna().sum()

Gender	13
Married	3
Dependents	15

34 / 34

EDA_FE.ipynb - Colaboratory

	25%	2877.500000	0.000000	100.000000	360.000000	1.000000
[12]	50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
	75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
	max	81000.000000	41667.000000	700.000000	480.000000	1.000000

RAM Disk

```
# categorical features  
data.describe(include = [ 'object'])
```

	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
count	601	611	599	614	582	614	614
unique	2	2	4	2	2	3	2
top	Male	Yes	0	Graduate	No	Semiurban	Y
freq	489	398	345	480	500	233	422

```
[16] #missing values  
      data.isna().sum()
```

Gender 1
Married
Dependents 1

EDA_FE.ipynb - Colaboratory

colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfH

◎ 内 容 提 取

+ Code + Text

✓ RAM Disk

```
[12] data.describe()  
      # only numeric features
```

NaN

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.000000	564.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

```
# categorical features  
data.describe(include = ['object'])
```

Gender **Married** **Dependents** **Education** **Self Employed** **Property Area** **Loan Status**

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfHCG7

+ Code + Text

unique 2 2 4 2 2 3 2

RAM Disk

top Male Yes 0 Graduate No Semiurban Y

freq 489 398 345 480 500 233 422

{x}

[16] #missing values
data.isna().sum()

Gender 13
Married 5
Dependents 15
Education 0
Self_Employed 32
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount 22
Loan_Amount_Term 14
Credit_History 50
Property_Area 0
Loan_Status 0
dtype: int64

NaN

[17] # catgeorical and numerical columns
cat_cols = data.dtypes == 'object'
cat_cols = list(cat_cols[cat_cols].index)

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfHCG7

+ Code + Text

unique 2 2 4 2 2 3 2

RAM Disk

top Male Yes 0 Graduate No Semiurban Y

freq 489 398 345 480 500 233 422

{x}

[16] #missing values
data.isna().sum()

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	
	13	3	15	0	32	0	0	22	14	50	0	0	0

dtype: int64

[17] # catgeorical and numerical columns
cat_cols = data.dtypes == 'object'
cat_cols = list(cat_cols[cat_cols].index)

38 / 38

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfHCG7

+ Code + Text

RAM Disk

Self_Employed 32

[16] ApplicantIncome 0

CoapplicantIncome 0

LoanAmount 22

Loan_Amount_Term 14

Credit_History 50

Property_Area 0

Loan_Status 0

dtype: int64

[17] # catgeorical and numerical columns

✓ cat_cols = (data.dtypes == 'object')

cat_cols = list(cat_cols[cat_cols].index)

num_cols = data.dtypes != 'object'

num_cols = list(num_cols[num_cols].index)

cat_cols.remove('Loan_Status')

[18] cat_cols

['Gender',
 'Married',
 'Dependents',
 'Education',
 'Self_Employed',
 'Property_Area']

EDA_FE.ipynb - Colaboratory



colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=WsyIZ1nfHCG7



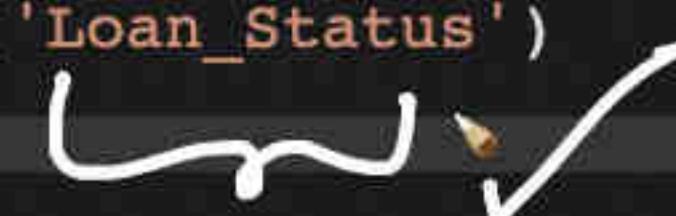
+ Code + Text

✓ RAM
Disk

```
Self_Employed      32
[16] ApplicantIncome      0
     CoapplicantIncome      0
     LoanAmount            22
     Loan_Amount_Term      14
     Credit_History        50
     Property_Area          0
     Loan_Status             0
dtype: int64
```

```
✓ [17] # catgeorical and numerical columns
✓ { cat_cols = data.dtypes == 'object'
✓ { cat_cols = list(cat_cols[cat_cols].index)

✓ { num_cols = data.dtypes != 'object'
num_cols = list(num_cols[num_cols].index)
cat_cols.remove('Loan_Status')
```



```
✓ [18] cat_cols
```

```
['Gender',
 'Married',
 'Dependents',
 'Education',
 'Self_Employed',
 'Property_Area']
```



EDA_FE.ipynb - Colaboratory

Code ≠ Text

freq

48

39

45

480

23

422

RAM Disk

```
[16] #missing values  
      data.isna().sum()
```

```
Gender          13  
Married         3  
Dependents     19  
Education        0  
Self_Employed   32  
ApplicantIncome  0  
CoapplicantIncome 0  
LoanAmount      21  
Loan_Amount_Term 14  
Credit_History   50  
Property_Area    0  
Loan_Status       0  
dtype: int64
```

```
✓ [17] # categorical and numerical columns  
cat_cols = data.dtypes == 'object'  
cat_cols = list(cat_cols[cat_cols].index)
```

```
num_cols = data.dtypes != 'object'  
num_cols = list(num_cols[num_cols].index)  
cat_cols.remove('Loan_Status')
```

→ Numerical
loan Amount

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=z7fWWrnv8wv3

+ Code + Text RAM Disk

[10] `Loan_ID` object
Gender object
Married object
Dependents object
Education object
Self_Employed object
ApplicantIncome int64
CoapplicantIncome float64
LoanAmount float64
Loan_Amount_Term float64
Credit_History float64
Property_Area object
Loan_Status object
dtype: object

data['Dependents'].value_counts()

Dependents	Count
0	345
1	102
2	101
3+	51

Name: Dependents, dtype: int64

[96] # drop loanID column
data = data.drop('Loan_ID', axis = 1)

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=z7fWWrnv8wv3

+ Code + Text RAM Disk

[10] `Loan_ID` object
Gender object
Married object
Dependents object
Education object
Self_Employed object
ApplicantIncome int64
CoapplicantIncome float64
LoanAmount float64
Loan_Amount_Term float64
Credit_History float64
Property_Area object
Loan_Status object
dtype: object

data['Dependents'].value_counts()

{
0 345
1 102
2 101
3+ 51
Name: Dependents, dtype: int64

[96] # drop loanID column
data = data.drop('Loan_ID', axis = 1)

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=z7fWWrnv8wv3

+ Code + Text RAM Disk

[10] `Loan_ID object
Gender object
Married object
Dependents object
Education object
Self_Employed object
ApplicantIncome int64
CoapplicantIncome float64
LoanAmount float64
Loan_Amount_Term float64
Credit_History float64
Property_Area object
Loan_Status object
dtype: object`

data['Dependents'].value_counts()

Dependents	Count
0	345
1	102
2	101
3	51

Name: Dependents, dtype: int64

[96] # drop loanID column
data = data.drop('Loan_ID', axis = 1)

44 / 45

EDA_FE.ipynb - Colaboratory

- Code ≠ Text

918B LEADERS SUMMIT

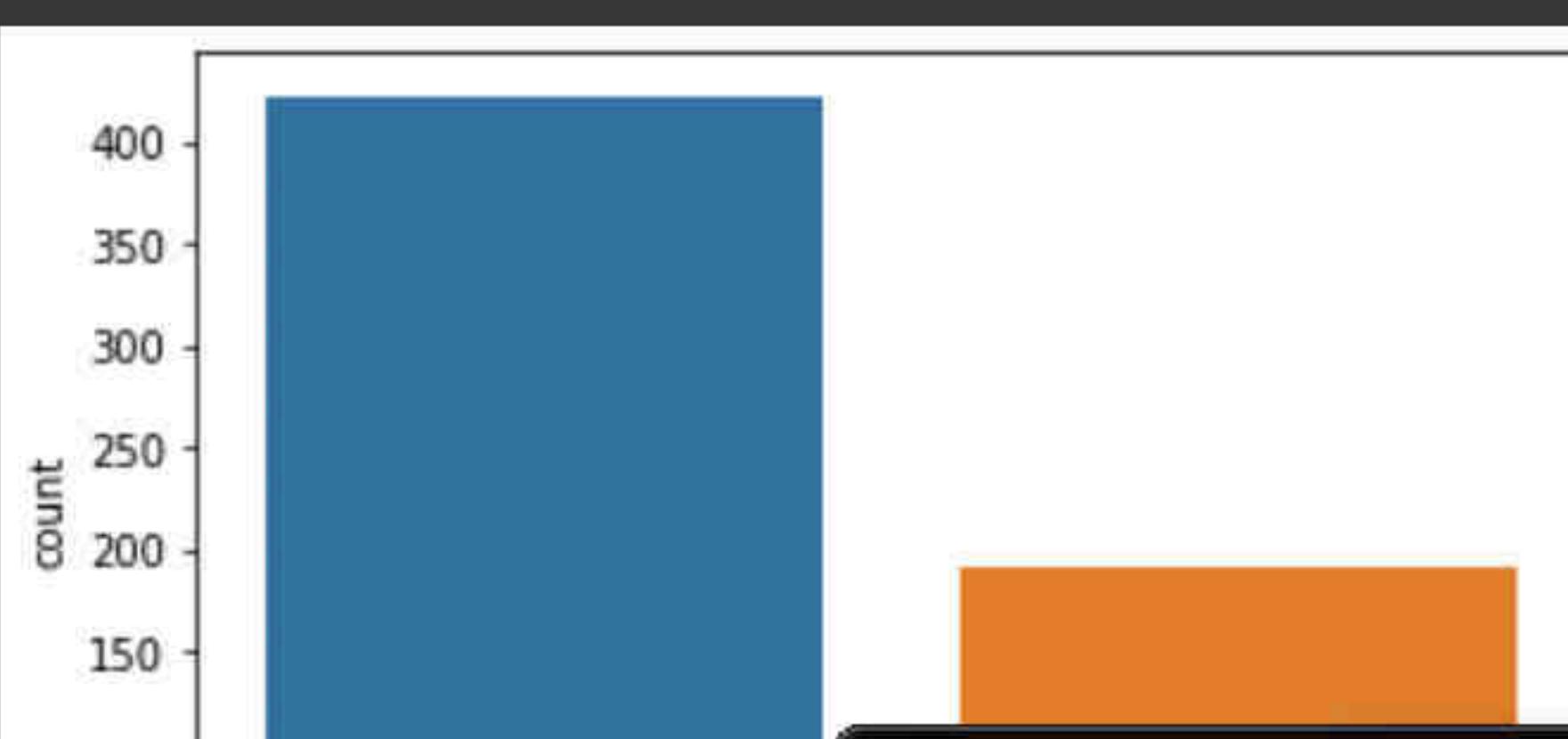
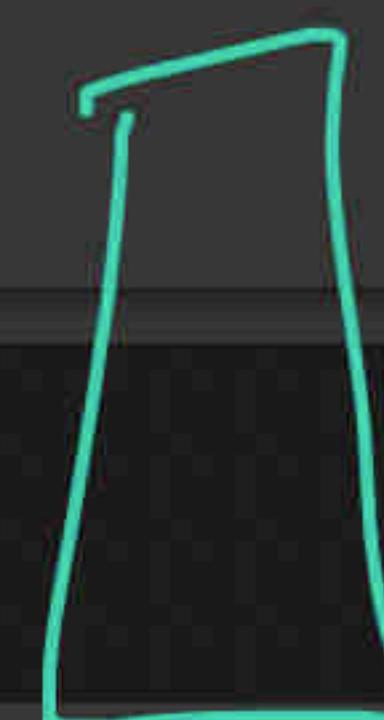
[96] data = data.drop('Loan ID', axis = 1)

{x} ▶ Basic Data Exploration

[1] ↳ 9 cells hidden

▼ Basic Data visualization: Univariate

```
[22] #Q: How many loans the company has approved in the past  
sns.countplot(data=data, x='Loan_Status')  
plt.show()
```



EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=GNmtX3um9oJR

+ Code + Text

[22] #Q: How many loans the company has approved in the past?

```
sns.countplot(data=data, x='Loan_Status')
plt.show()
```

Questions
[Sherlock-holmes]
+ tools(Pak-stats,
ML, DL--)
high-school

Count

400
350
300
250
200
150
100
50
0

Y N

Loan_Status

[23] target = 'Loan_Status'
data[target].value_counts()

Imbalanced data

Loan_Status	Count
Y	422
N	192

Y 422
N 192

46 / 47

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=YJaGsvMVJcpC

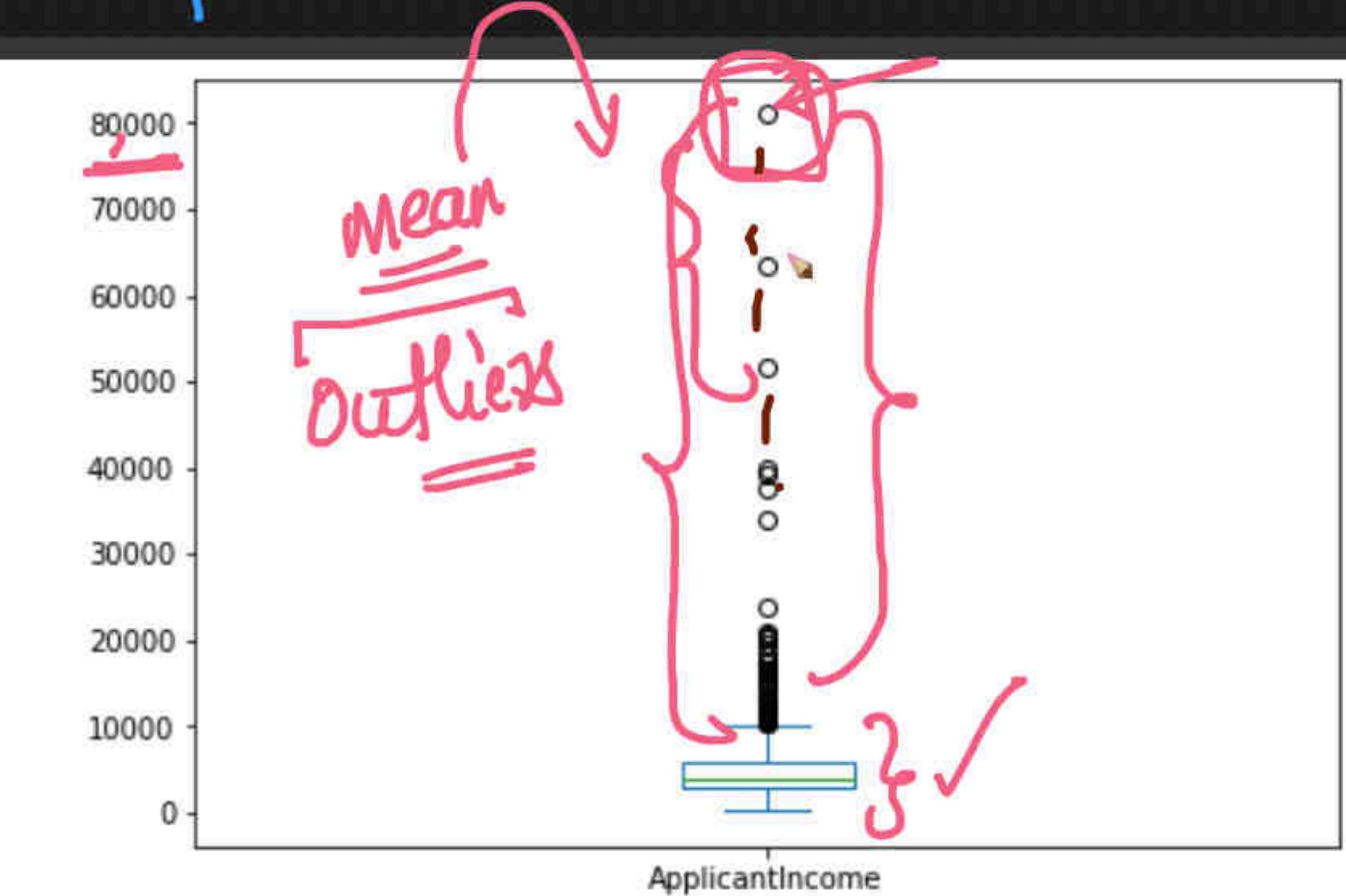
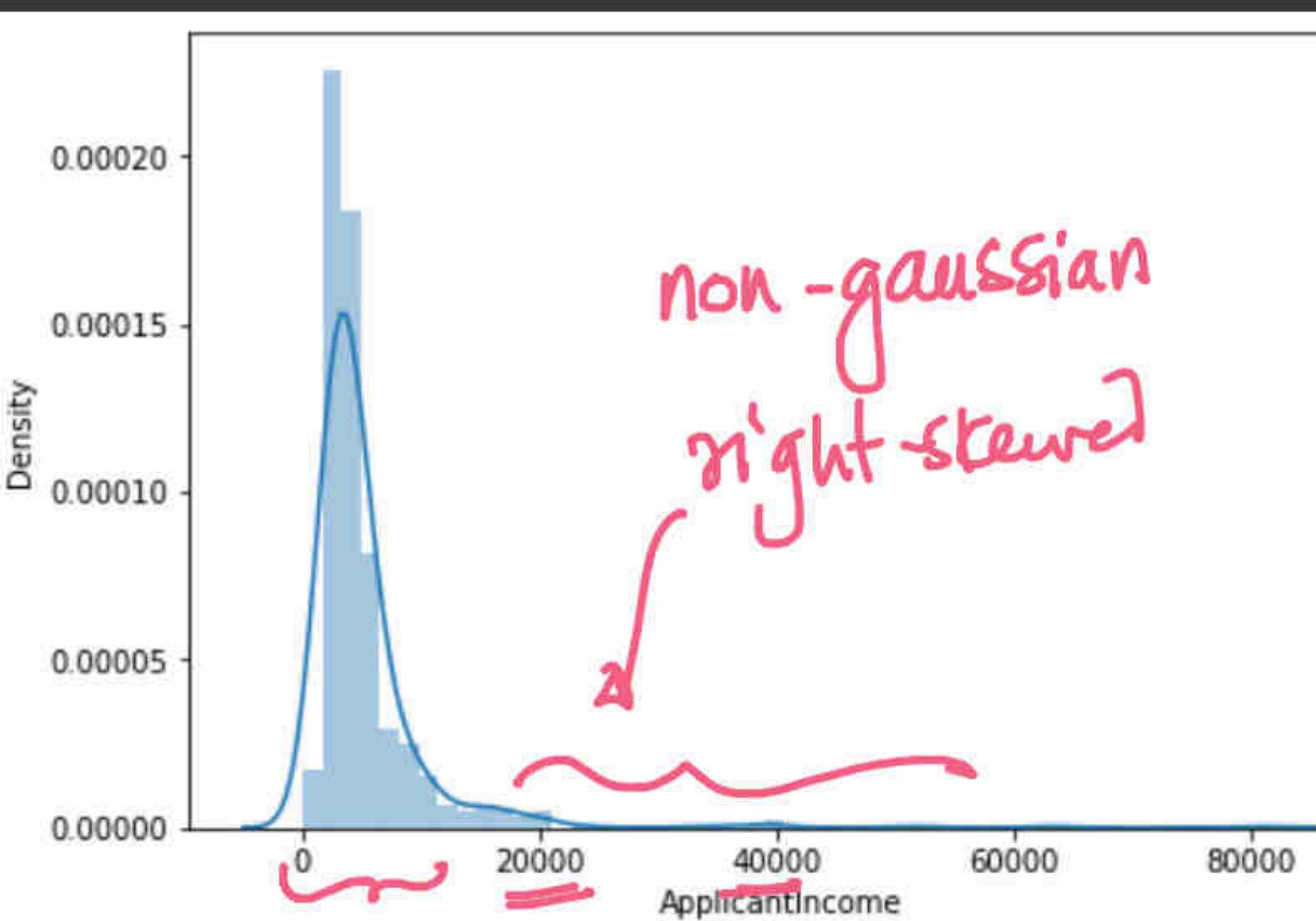
+ Code + Text

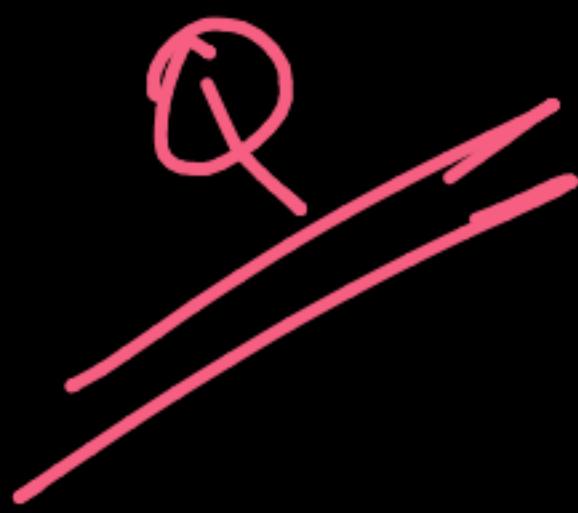
RAM Disk

```
#Income of the applicant
plt.subplot(121)
sns.distplot(data["ApplicantIncome"])
plt.subplot(122)
data["ApplicantIncome"].plot.box(figsize=(16,5))
plt.show()
```

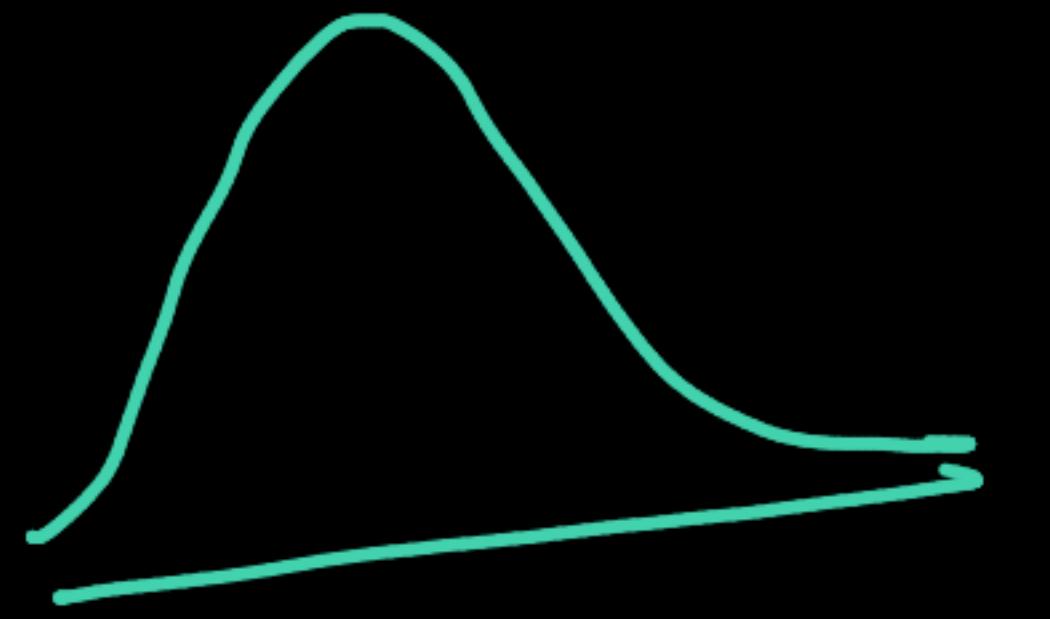
$y=1$ or 0

comp of means \leftarrow t-test / ~~t-test~~

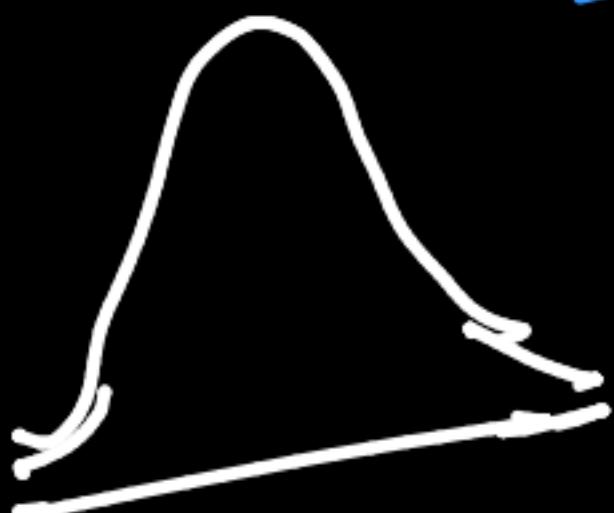




Income \rightarrow outliers \Rightarrow we want to avoid t-test



\downarrow log boxcox



Can I somehow find a sdn using t-test
 \hookrightarrow remove outliers (QR) ✓
 \rightarrow median X
 \rightarrow

∞ EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=mg6keQsT_PL5

+ Code + Text

RAM Disk

ApplicantIncome

1s

plt.subplot(121)
sns.distplot(np.log(data["ApplicantIncome"]))
plt.show()

box-cox ✓

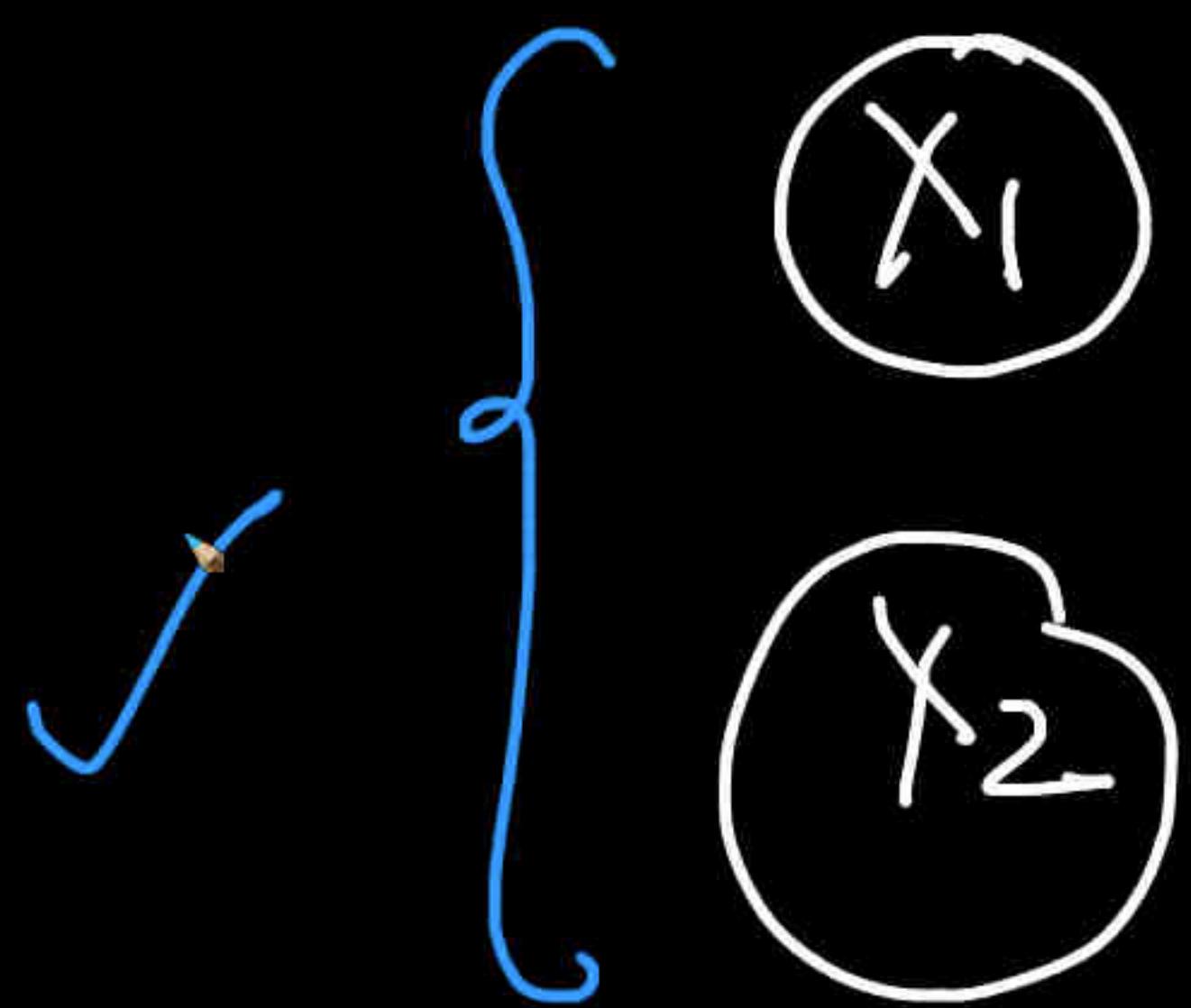
Density

0.8
0.6
0.4
0.2
0.0

6 8 10 12

ApplicantIncome

[] #Slice this data by Education



$$\xrightarrow{\log} \frac{x_1}{x_2} \xrightarrow{\text{t-test}} \frac{\log(\mu_1)}{\log(\mu_2)}$$

A flowchart illustrating a statistical analysis process. It starts with the ratio $\frac{x_1}{x_2}$, which is then converted to $\log\left(\frac{x_1}{x_2}\right)$. This result is then compared using a t-test, represented by the fraction $\frac{\log(\mu_1)}{\log(\mu_2)}$. The term "t-test" is underlined. The entire equation $\frac{\log(\mu_1)}{\log(\mu_2)}$ is also underlined in blue.

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=mg6keQsT_PL5

+ Code + Text

RAM Disk

[25] `data.boxplot(column='ApplicantIncome', by="Education", figsize=(8,5))
plt.suptitle("")
plt.show()`

In [25]

ApplicantIncome

Graduate

Not Graduate

[26] #co-applicant income

EDA_FE.ipynb - Colaboratory

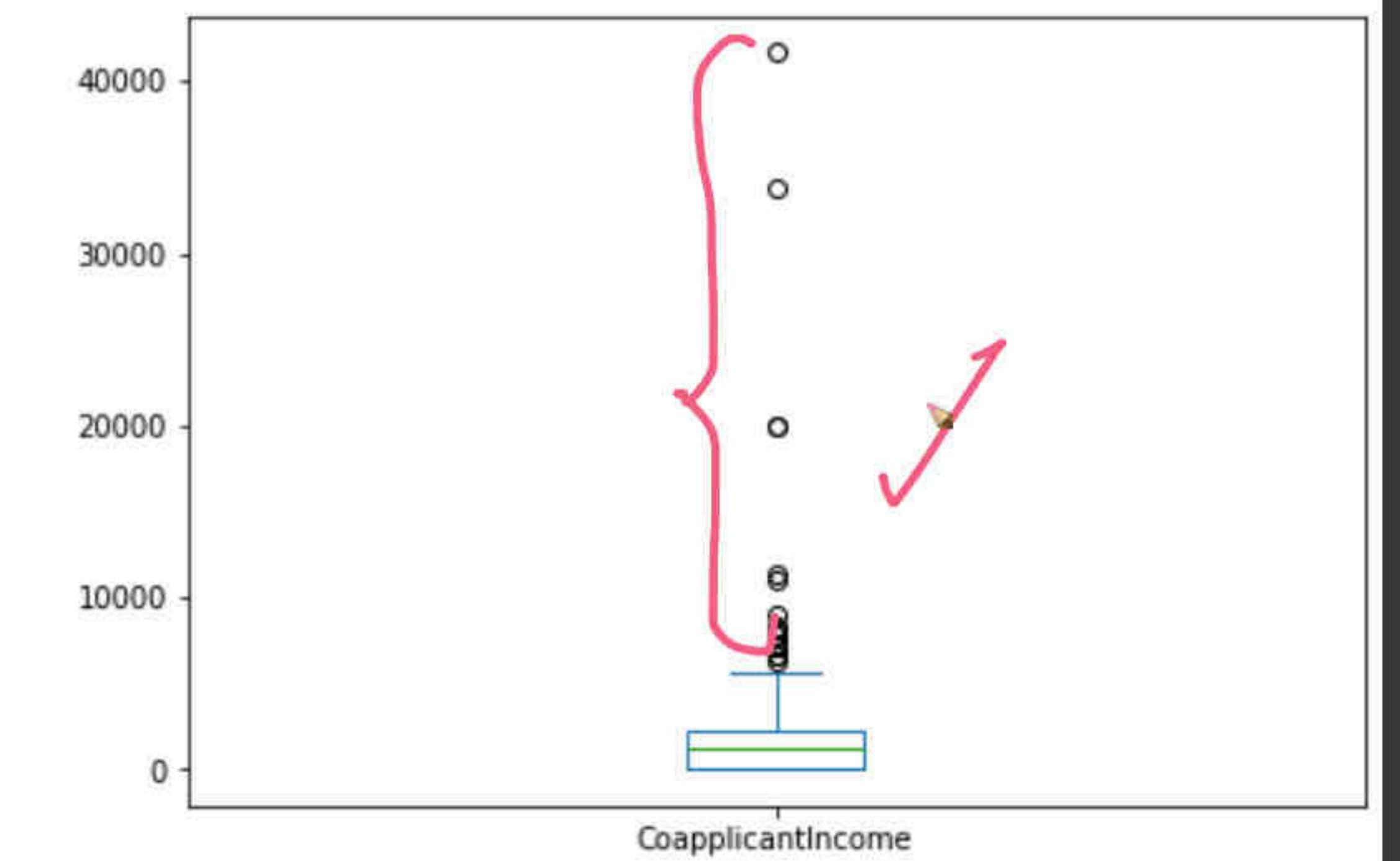
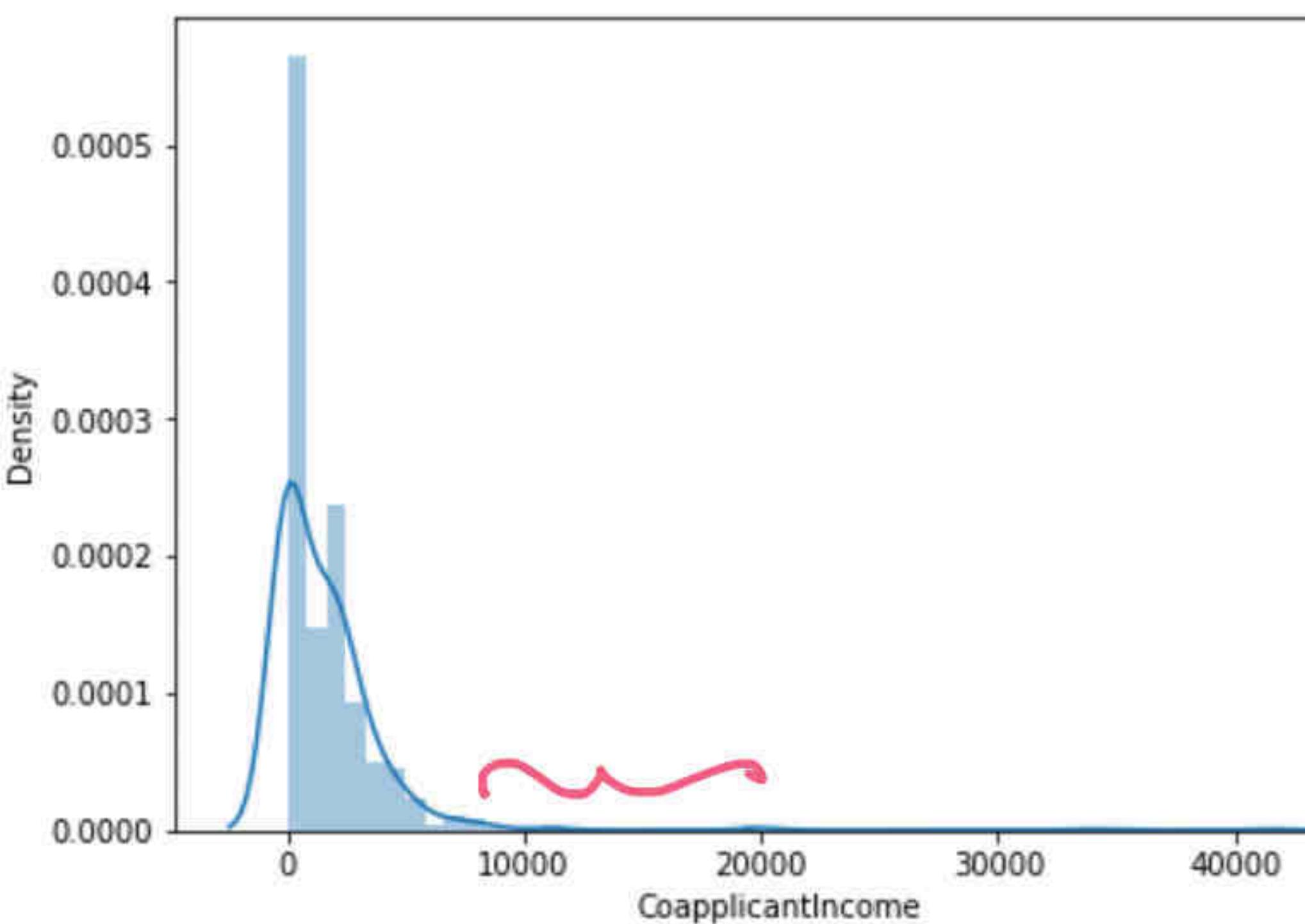
colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=juKACrkOJ4gk

+ Code + Text

RAM
Disk

```
#co-applicant income
plt.subplot(121)
sns.distplot(data["CoapplicantIncome"])

plt.subplot(122)
data[ "CoapplicantIncome" ].plot.box(figsize=(16,5))
plt.show()
```



EDA_FE.ipynb - Colaboratory

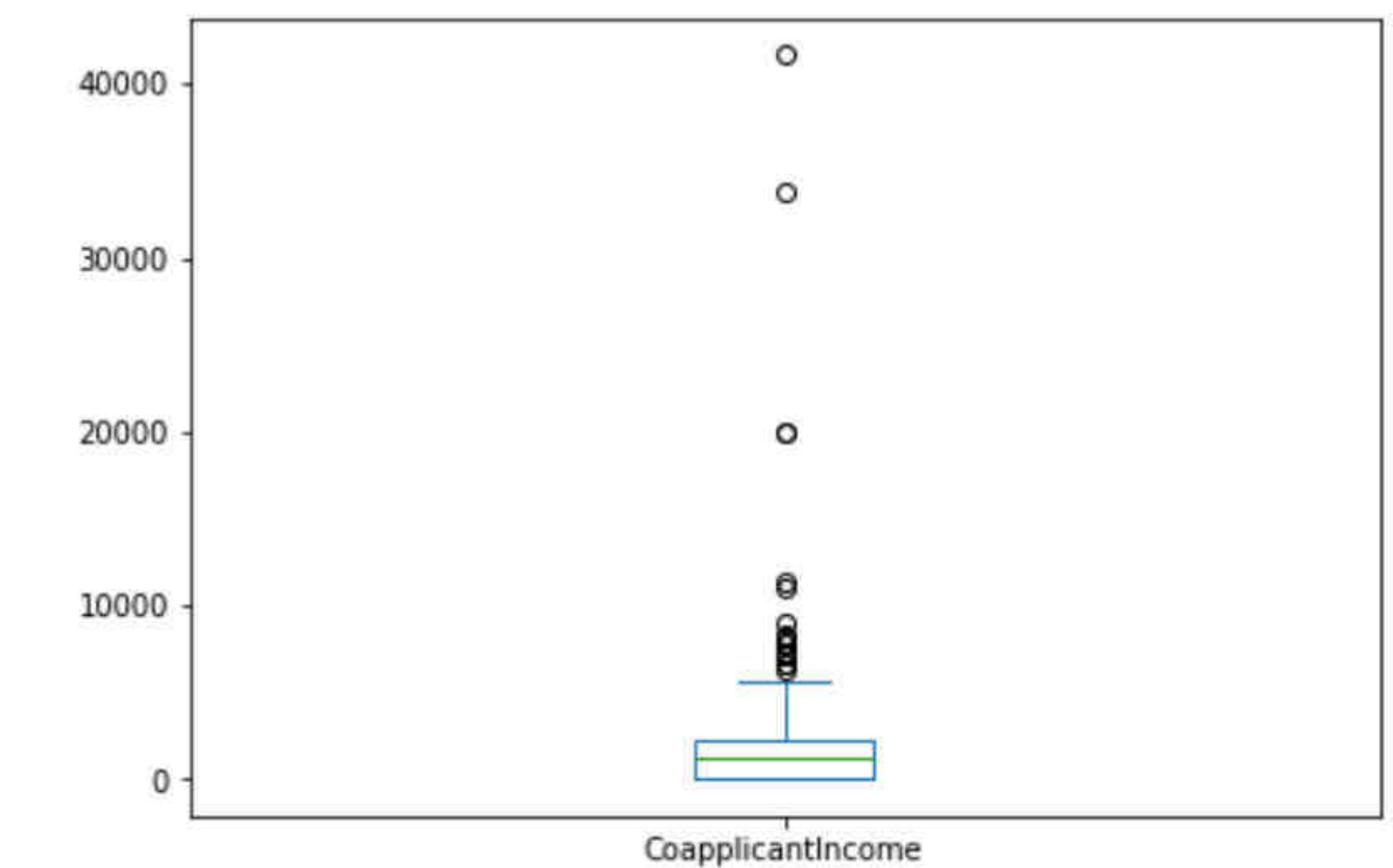
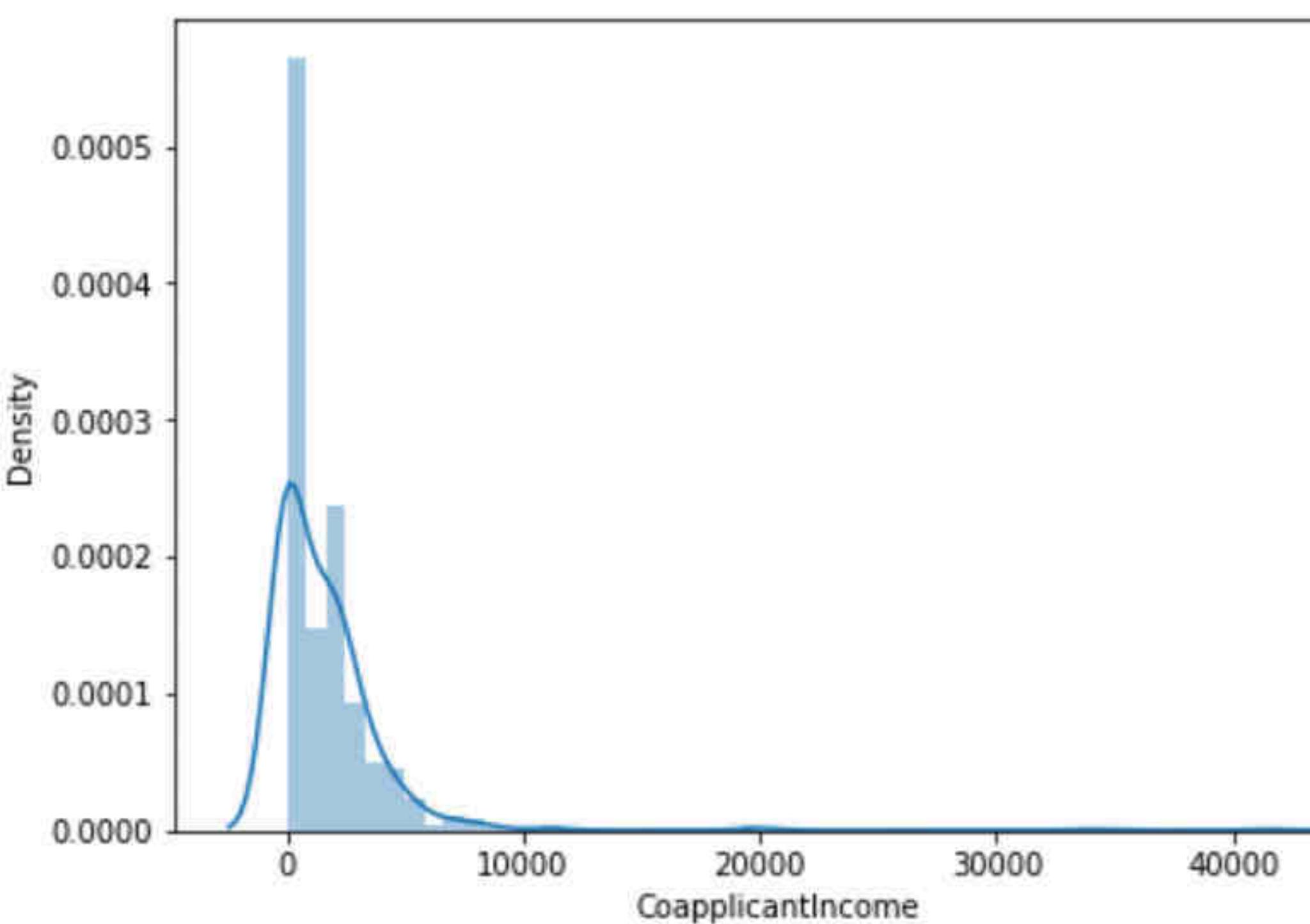
colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=juKACrkOJ4gk

+ Code + Text

RAM
Disk

```
#co-applicant income
plt.subplot(121)
sns.distplot(data["CoapplicantIncome"])

plt.subplot(122)
data[ "CoapplicantIncome" ].plot.box(figsize=(16,5))
plt.show()
```



EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=gwludxlnKDvf

+ Code + Text RAM Disk

#Relation between "Loan_Status" and "Income"

[27] data.groupby("Loan_Status").mean()["ApplicantIncome"]

Loan_Status
N 5446.078125
Y 5384.068720
Name: ApplicantIncome, dtype: float64

Mean-ing where Status = Y
Mean-ing x = N

M₁

M₂

```
data.groupby("Loan_Status").mean()["ApplicantIncome"].plot.bar()  
plt.ylabel("Mean Income of applicant")  
plt.show()
```



EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=gwludxlnKDvf

+ Code + Text RAM Disk

#Relation between "Loan_Status" and "Income"

[27] data.groupby("Loan_Status").mean()['ApplicantIncome']

Loan_Status
N 5446.078125
Y 5384.068720
Name: ApplicantIncome, dtype: float64

480↑ P(30)
N, 8N2

0s

data.groupby("Loan_Status").mean()['ApplicantIncome'].plot.bar()
plt.ylabel("Mean Income of applicant")
plt.show()

Mean Income of applicant

5000
4000
3000
2000
1000

55 / 55

EDA_FE.ipynb - Colaboratory

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=gwludxlnKDvf

+ Code + Text

✓ RAM Disk

#Relation between "Loan_Status" and "Income"

n₁ & n₂ are large

[27] data.groupby("Loan_Status").mean()['ApplicantIncome']

{

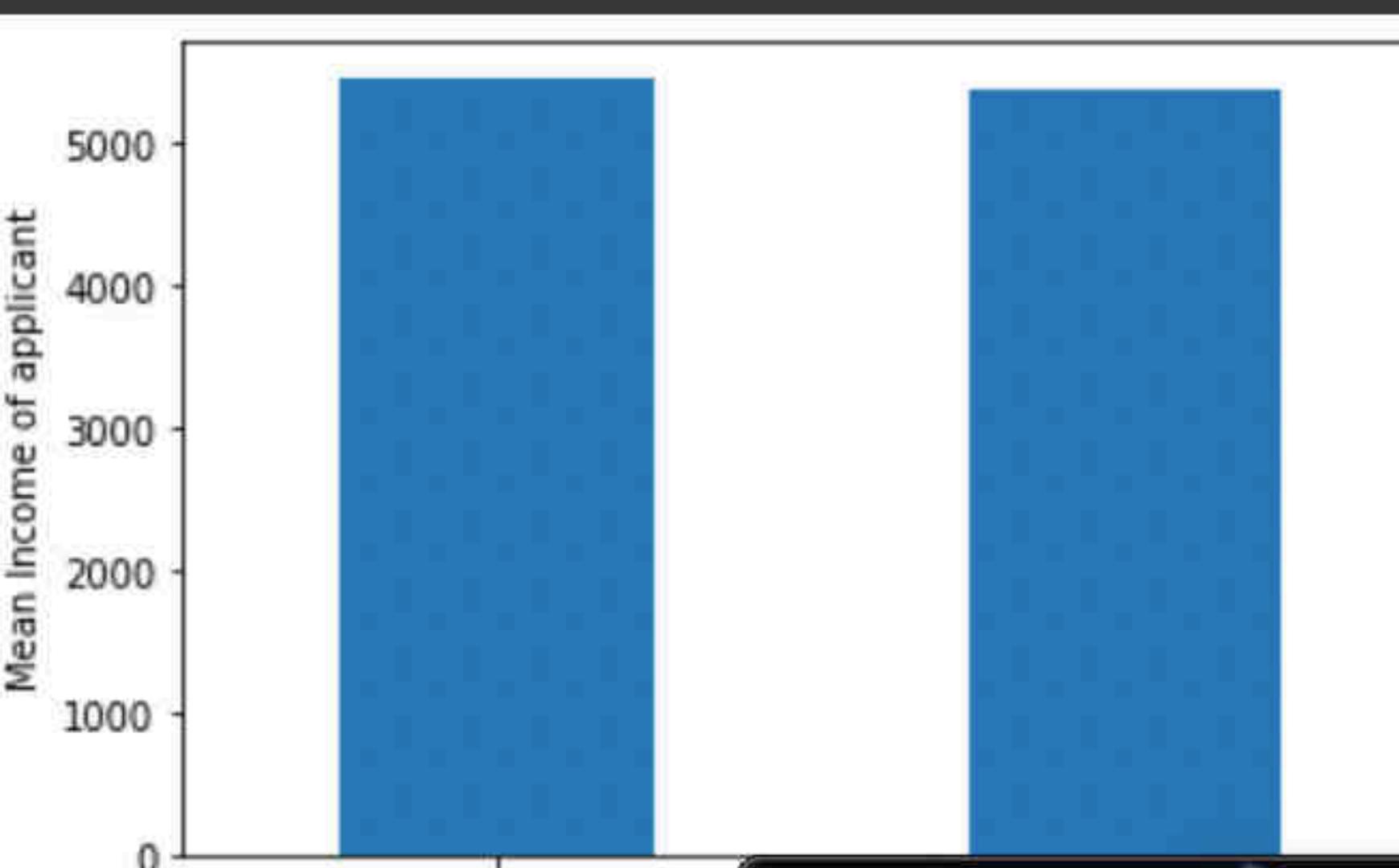
Loan_Status	
N	5446.078125
Y	5384.068720

Name: ApplicantIncome, dtype: float64}{ $M_1 - M_2$ is very small}[28] data.groupby("Loan_Status").mean()['ApplicantIncome'].plot.bar()
plt.ylabel("Mean Income of applicant")
plt.show()

Ques: error in this

analysis

outliers



EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=gwludxlnKDvf

+ Code + Text

#Relation between "Loan_Status" and "Income"

[27] data.groupby("Loan_Status").mean()['ApplicantIncome']

Loan_Status	ApplicantIncome
N	5446.078125
Y	5384.068720

Name: ApplicantIncome, dtype: float64

[28] data.groupby("Loan_Status").mean()['ApplicantIncome'].plot.bar()
plt.ylabel("Mean Income of applicant")
plt.show()

task! ↗ (OR remove
= outliers ↘ Mean, & Mean_2

Mean Income of applicant

5000
4000
3000
2000
1000
0

N Y

5446.078125
5384.068720

57 / 57

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=gwludxlnKDvf

+ Code + Text

RAM Disk

#Relation between "Loan_Status" and "Income"

[27] data.groupby("Loan_Status").mean()['ApplicantIncome']

Loan_Status
N 5446.078125
Y 5384.068720
Name: ApplicantIncome, dtype: float64

3 → 95% ↴

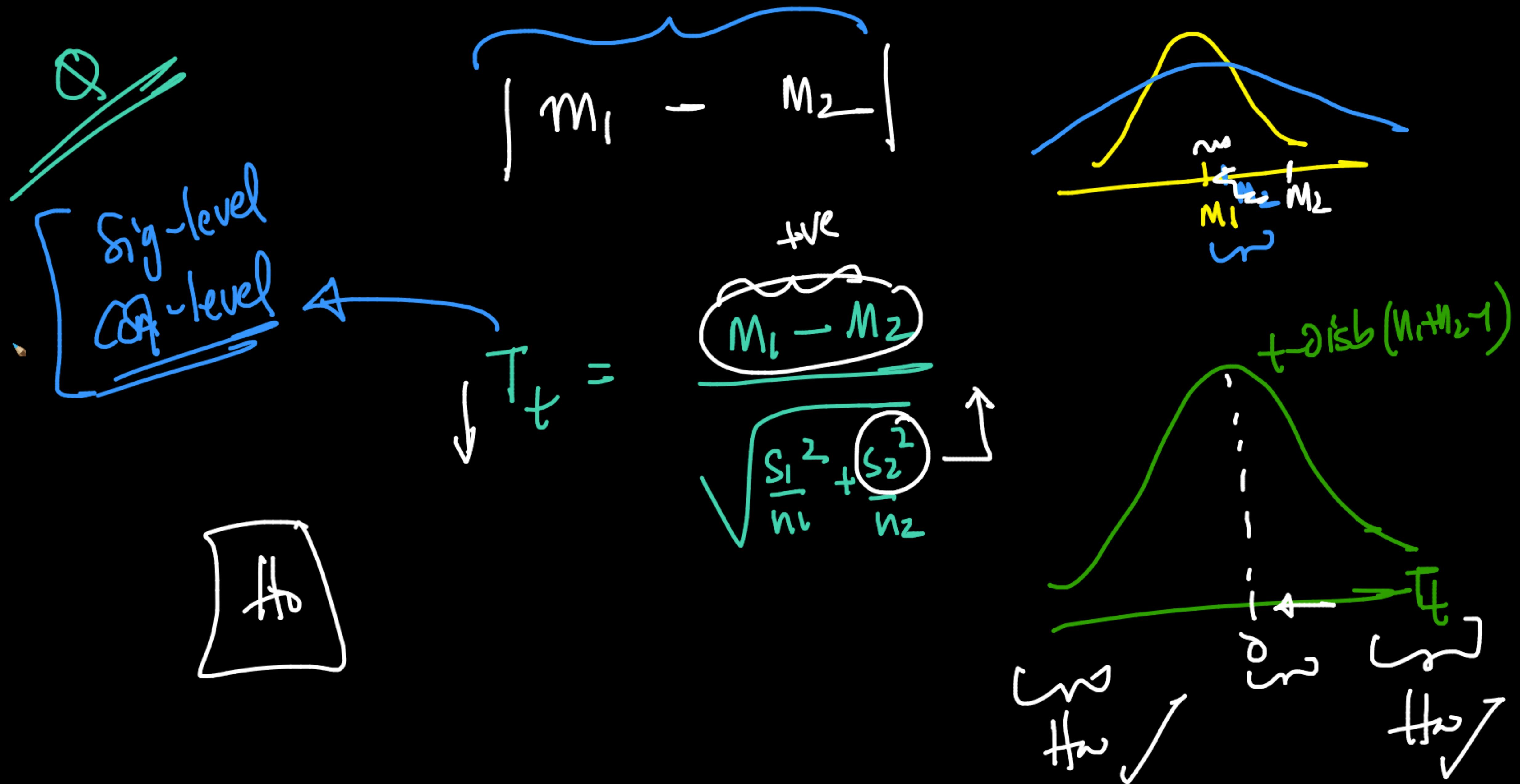
PDF:

CDF:

QQ-plot: →
 $\theta_1, \theta_2, \theta_3$ →

(log)
(KS)
(t-test)

58 / 58



95% - CI
= =
→ t-test

KS-test

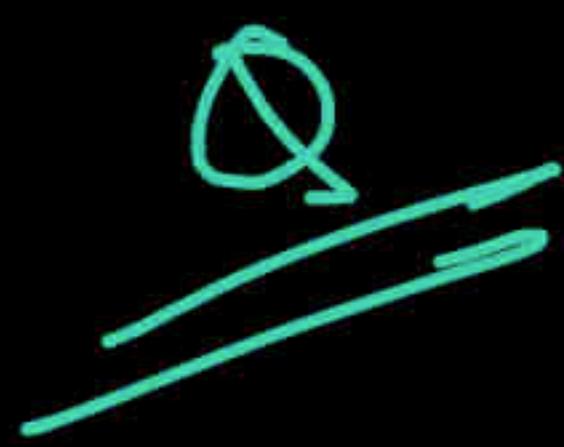
outlier analysis

inc
loan
=

Race/ethnicity $\xrightarrow{\text{Cat}}$ loan approval

χ^2 -test $\xrightarrow{\text{Cat}}$ ✓

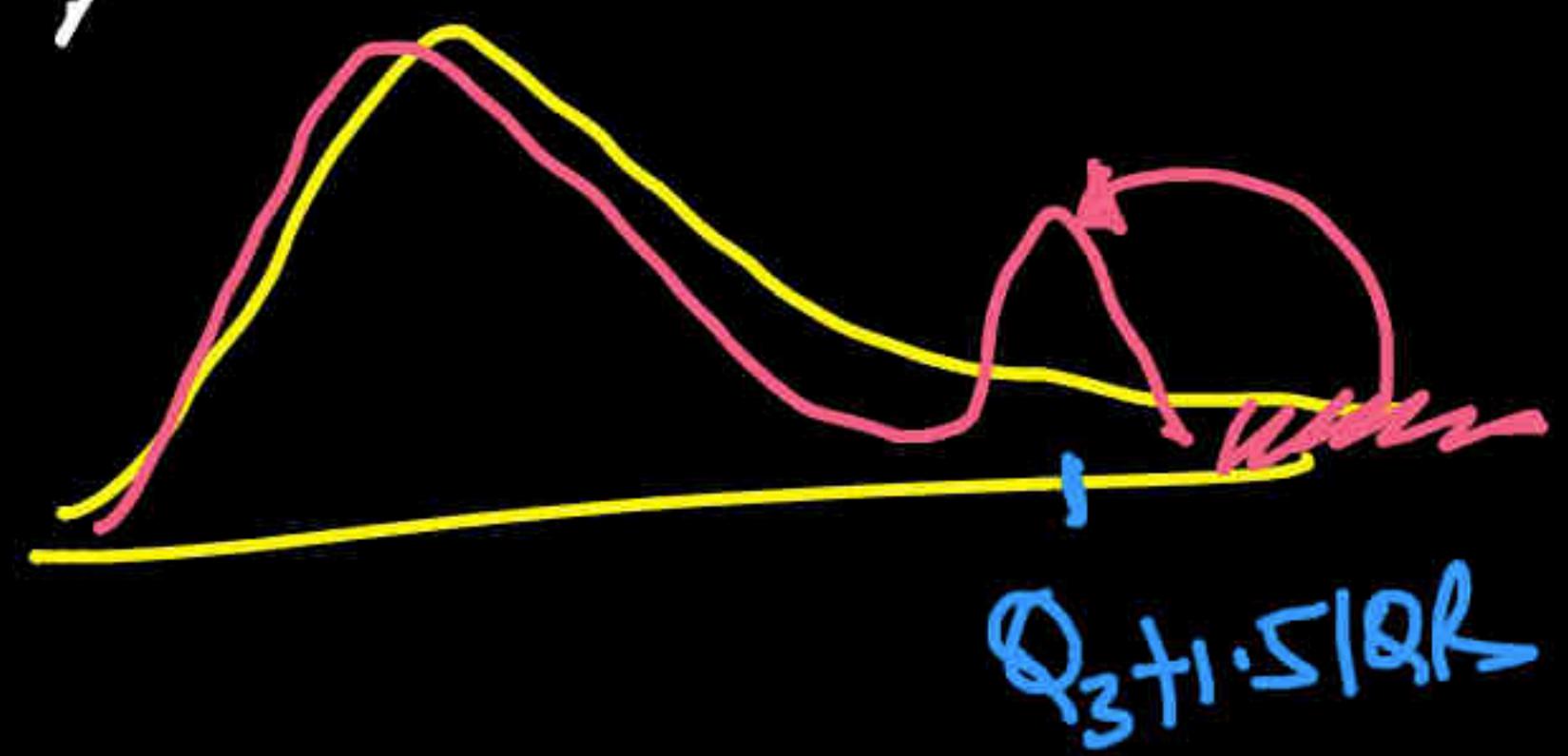
Harvard
(Asian-American)



$Q_3 + 1.5 IQR$

$\hookrightarrow \underline{Q_3 + 1.5 IQR}$

$bq / box - box$



Multi-modal

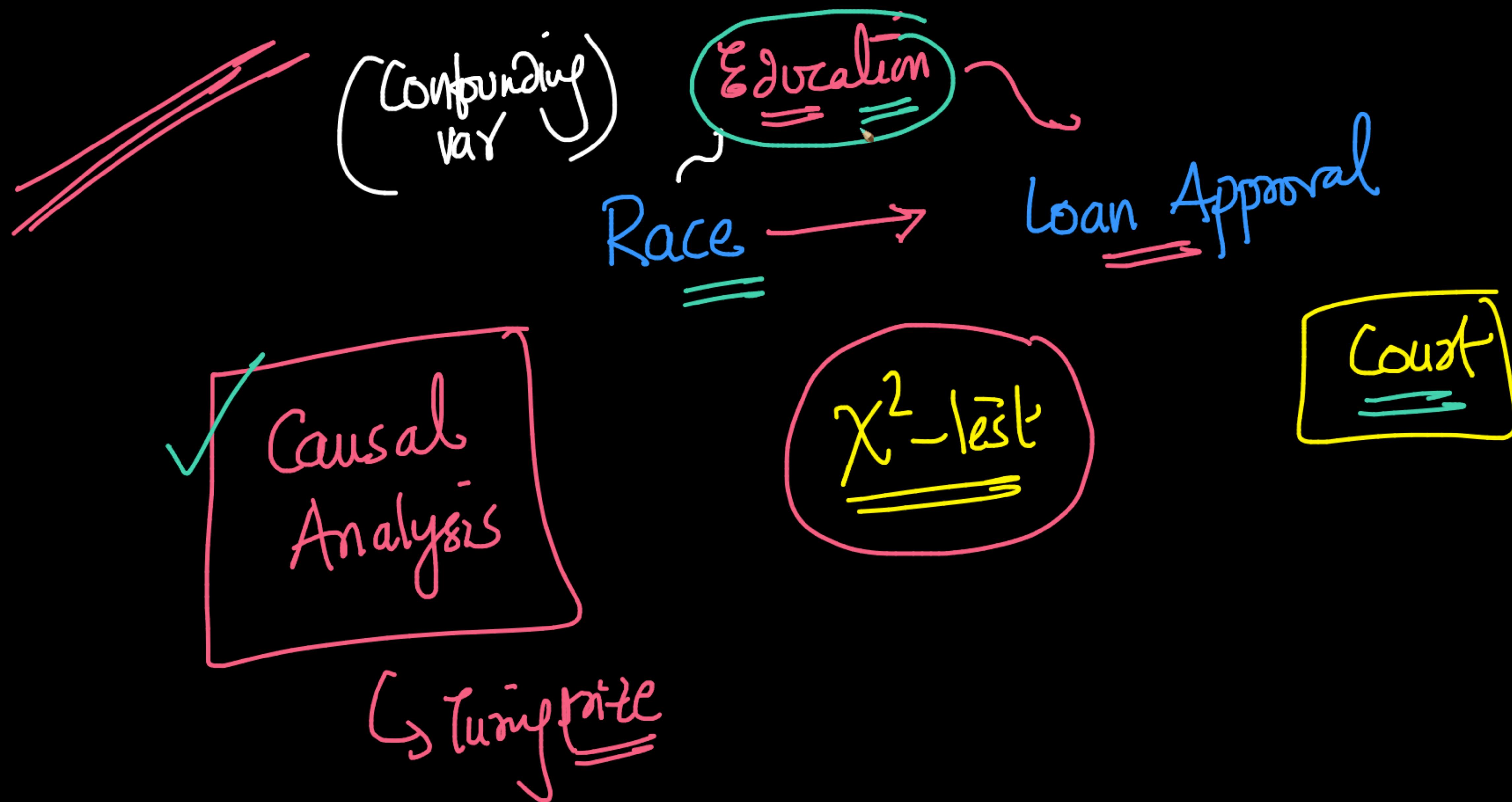
Q

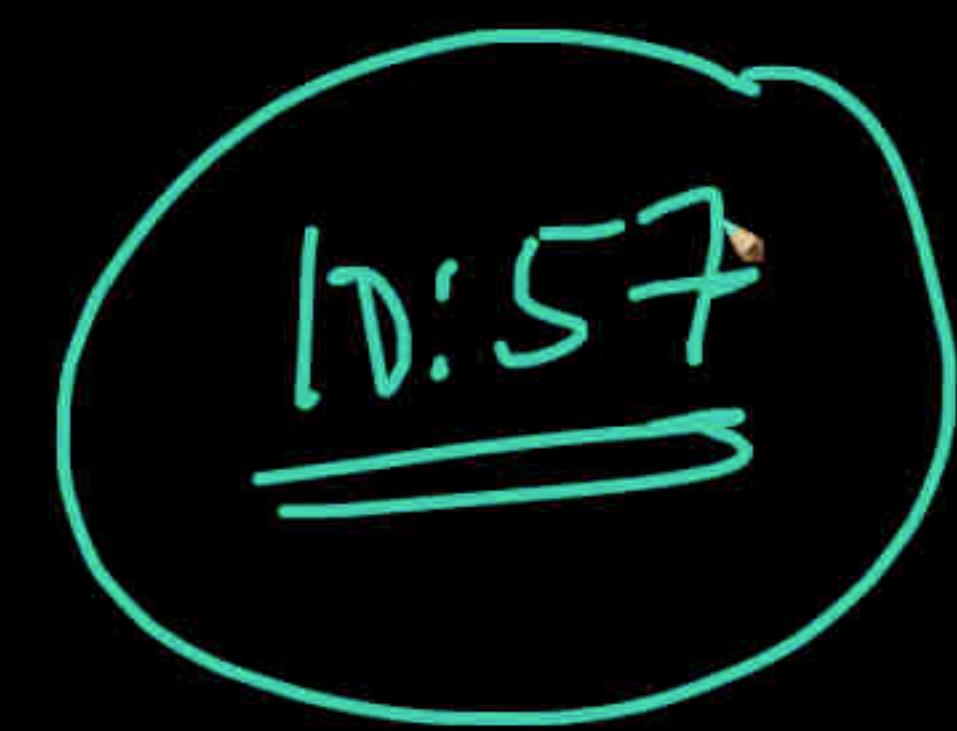
INC → loan approval

$$\left\{ \begin{array}{l} H_0: \mu_1 \neq \mu_2 \\ H_a: \mu_1 = \mu_2 \end{array} \right.$$

T_t under $H_0 \rightarrow$ DO NOT KNOW

↓
Cannot use hypothesis





feature Engineering

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=19FtHN8LKZS_

+ Code + Text RAM Disk

Simple Feature Engineering

binning / bucketing

```
# Feature binning: income
bins=[0,2500,4000,6000,81000]
group=['Low', 'Average', 'High', 'Very high']
data['Income_bin']= pd.cut(data['ApplicantIncome'],bins,labels=group)
```

[30] data.head()

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Tenure
0	Male	No	0	Graduate	No	5849	0.0	NaN	365.0
1	Male	Yes	1	Graduate	No	4583	1508.0	128.0	365.0
2	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	365.0
3	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	365.0
4	Male	No	0	Graduate	No	6000	0.0	141.0	365.0

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=19FtHN8LKZS_

+ Code + Text RAM Disk

Simple Feature Engineering

0 → 2499

2500 - 3999

```
# Feature binning: income
bins=[0,2500,4000,6000,81000]
group=['Low', 'Average', 'High', 'Very high']
data['Income_bin']= pd.cut(data['ApplicantIncome'],bins,labels=group)
```

[30] data.head()

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Tenure
0	Male	No	0	Graduate	No	5849	0.0	Nan	365
1	Male	Yes	1	Graduate	No	4583	1508.0	128.0	365
2	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	365
3	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	365
4	Male	No	0	Graduate	No	6000	0.0	141.0	365

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=19FtHN8LKZS_

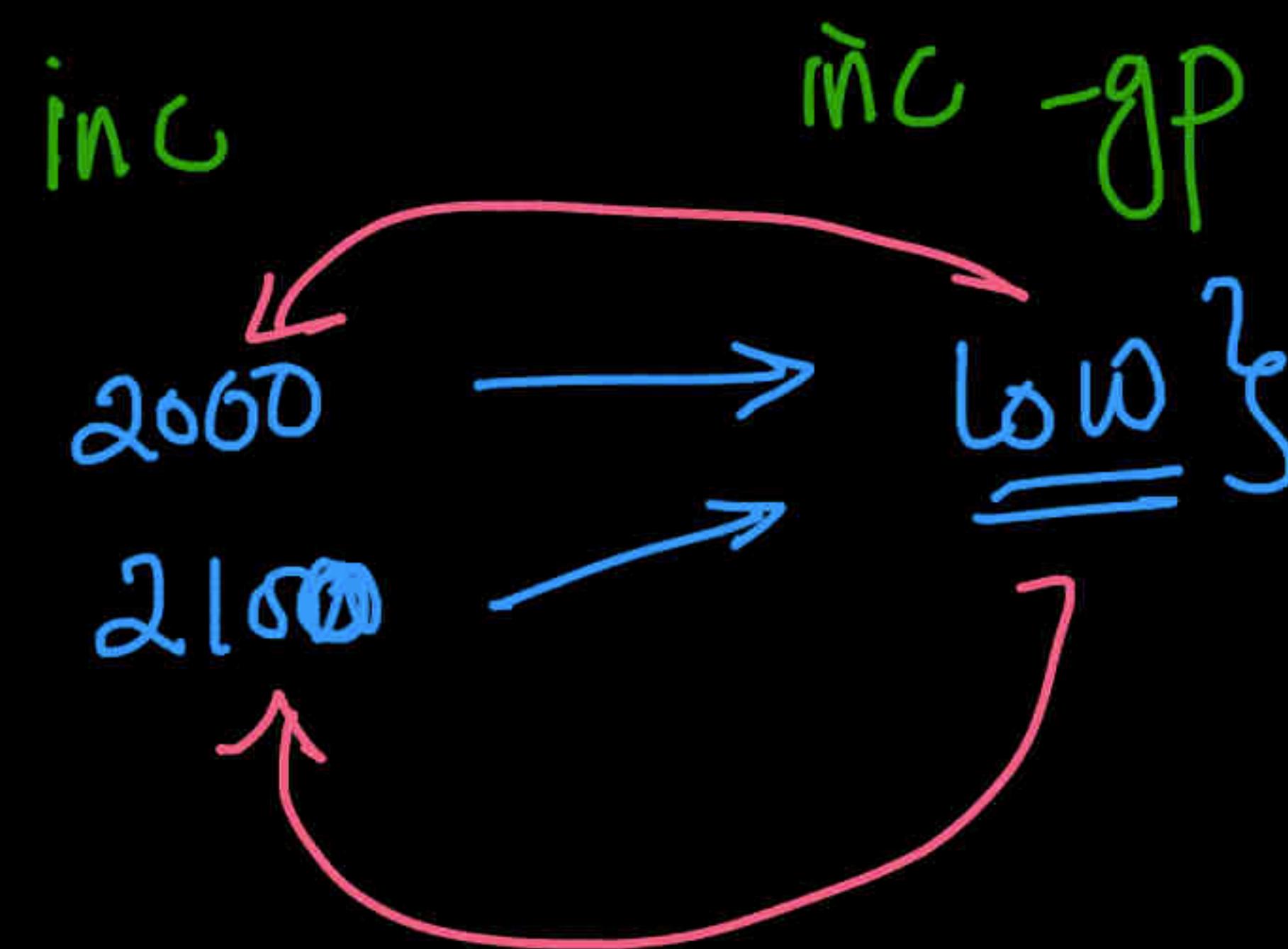
+ Code + Text RAM Disk

Simple Feature Engineering

Feature binning: income
bins=[0,2500,4000,6000,81000]
group=['Low', 'Average', 'High', 'Very high']
data['Income_bin']= pd.cut(data['ApplicantIncome'],bins,labels=group)

[30] data.head()

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Tenure
0	Male	No	0	Graduate	No	5849	0.0	Nan	365.0
1	Male	Yes	1	Graduate	No	4583	1508.0	128.0	365.0
2	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	365.0
3	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	365.0
4	Male	No	0	Graduate	No	6000	0.0	141.0	365.0



non-invertable
≡

EDA_FE.ipynb - Colaboratory

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=D8DZbk7hM4DF

+ Code + Text

✓ RAM Disk



```
[31] #observed  
{x} pd.crosstab(data["Income_bin"], data["Loan_Status"])
```

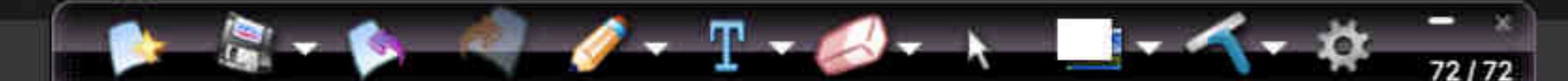
χ^2 -test

Income_bin	N	Y
Loan_Status		
✓ Low	34	74
✗ Average	67	159
✓ High	45	98
✗ Very high	46	91

```
[32] Income_bin = pd.crosstab(data["Income_bin"], data["Loan_Status"])

Income_bin.div(Income_bin.sum(axis=1), axis=0).plot(kind="bar", figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per ou
```



EDA_FE.ipynb - Colaboratory

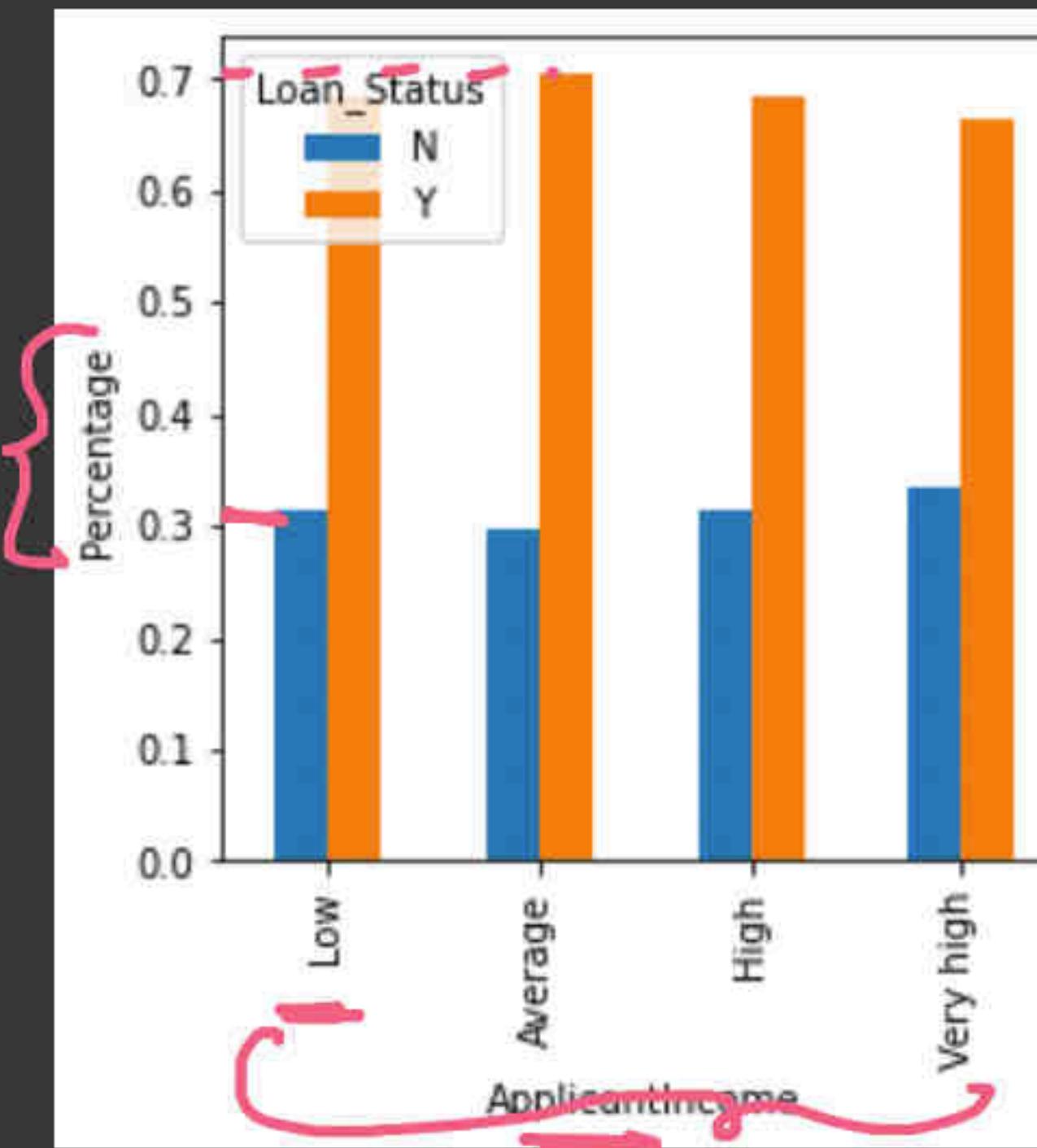
colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=19FtHN8LKZS_

+ Code + Text

✓ RAM Disk

```
[32] Income_bin.div(Income_bin.sum(axis=1),axis=0).plot(kind="bar",figsize=(4,4))
    plt.xlabel("ApplicantIncome")
    plt.ylabel("Percentage")
    plt.show()
```

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per ou



Insightful

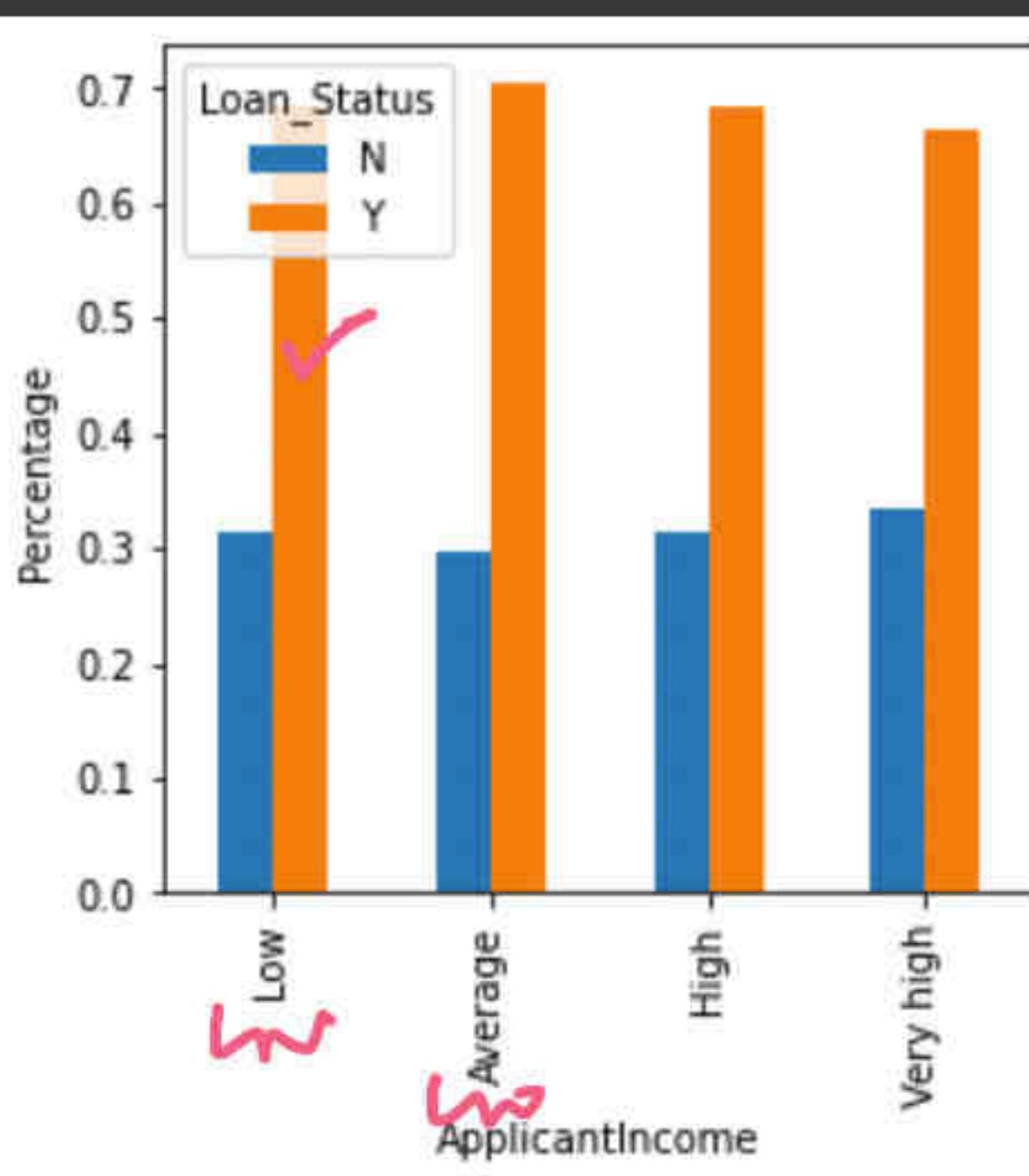
INC → INC-bIAS

Code # Text

RAM Disk

```
[32] Income_bin.div(Income_bin.sum(axis=1),axis=0).plot(kind="bar",figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()
```

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per our analysis.



{ INC-blins ↗ loan-status
INC ↗ loan-status

EDA_FE.ipynb - Colaboratory + colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=19FtHN8LKZS_ + Code + Text RAM Disk ✓

```
[32] Income_bin.div(Income_bin.sum(axis=1),axis=0).plot(kind="bar",figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()
```

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per ou

ApplicantIncome	N	Y
Low	~0.32	~0.68
Average	~0.30	~0.70
High	~0.32	~0.68
Very high	~0.34	~0.66

A pink curly brace is drawn around the 'Y' bars for all four income levels, highlighting that the percentage of approved loans remains relatively constant (around 0.65-0.70%) regardless of the income level.

EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=EcFsZRhLK1DI

+ Code + Text

RAM Disk

Q {x} □

INCOME_BIN = pd.crosstab(data['INCOME_BIN'], data['Loan_Status'])

Income_bin.div(Income_bin.sum(axis=1), axis=0).plot(kind="bar", figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per ou

0.3 - 0.33

0.28

0.26

0.24

0.22

0.20

0.18

0.16

0.14

0.12

0.10

0.08

0.06

0.04

0.02

0.00

Percentage

Low Average medium H1 H2 H3 H4 Very high

ApplicantIncome

Loan_Status
N
Y

ApplicantIncome	N (%)	Y (%)
Low	~0.30	~0.68
Average	~0.29	~0.70
medium	~0.31	~0.68
H1	~0.37	~0.58
H2	~0.30	~0.57
H3	~0.32	~0.68
H4	~0.32	~0.66
Very high	~0.33	~0.67

76 / 76

+ Code + Text

RAM Disk



51043 Income bin

[104]

Low	34	7
Average	67	15
medium	45	9
H1	20	3
h2	9	2
h3	13	2
h4	3	
Very high	1	

```
Income_bin = pd.crosstab(data["Income_bin"], data["Loan_Status"])
```

```
Income_bin.div(Income_bin.sum(axis=1),axis=0).plot(kind="bar",figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()
```

 EDA_FE.ipynb – Colaboratory

colab.research.google.com/drive/1_P-nKswaKnx5v77jOx3pCxvJve73MyN#scrollTo=Wt87woF

◎ 金 星 网 口 ◎

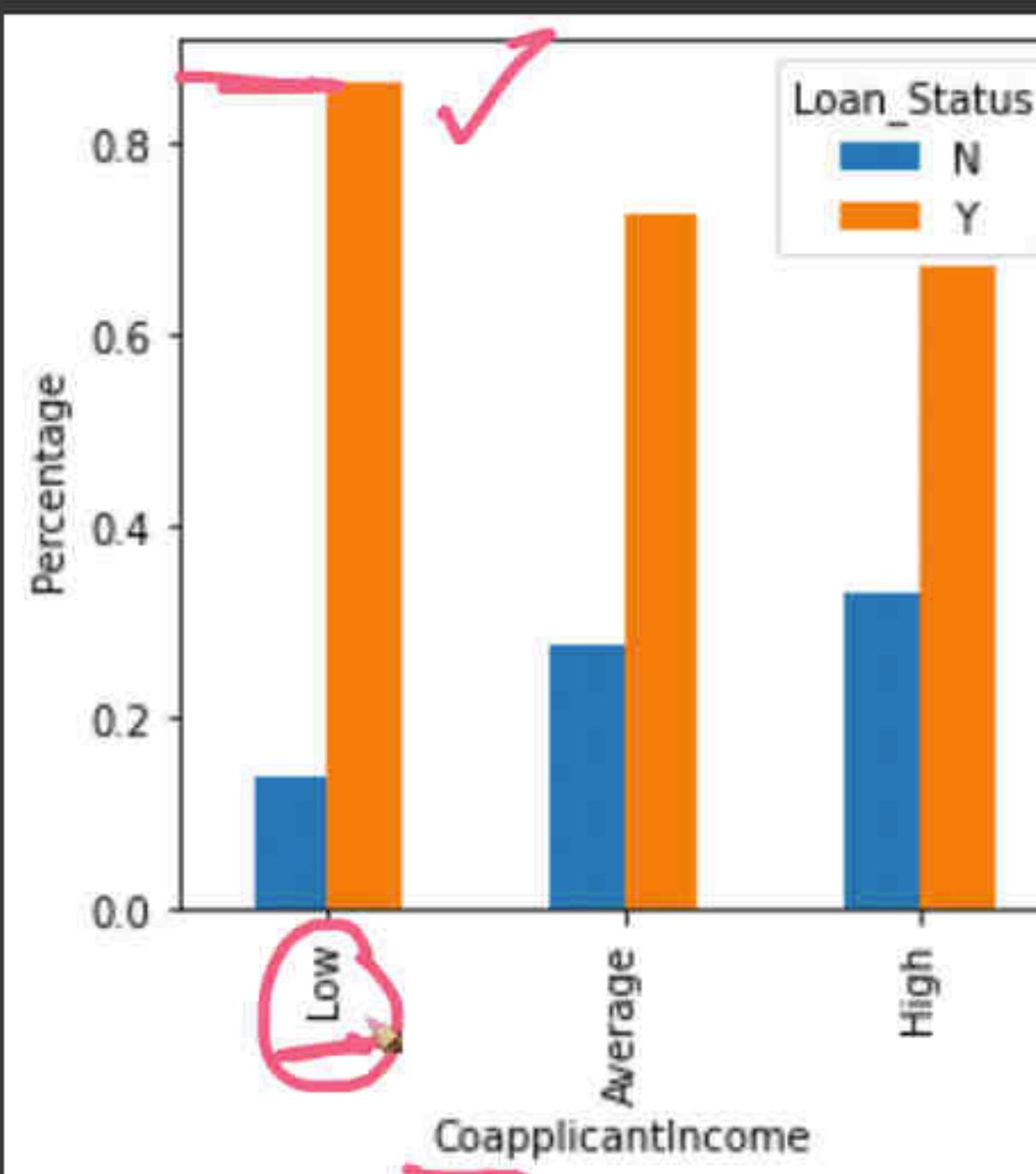
+ Code + Text

RAM Disk

V

```
CoapplicantIncome_Bin.div(Coap  
plt.xlabel("CoapplicantIncome"  
plt.ylabel("Percentage")  
plt.show()
```

What's the problem here? Why co-applicant having low income is getting maximum loan approved?



```
[36] data['CoapplicantIncome'].value_counts().head
```

 EDA_FE.ipynb – Colaboratory X

Code + Text

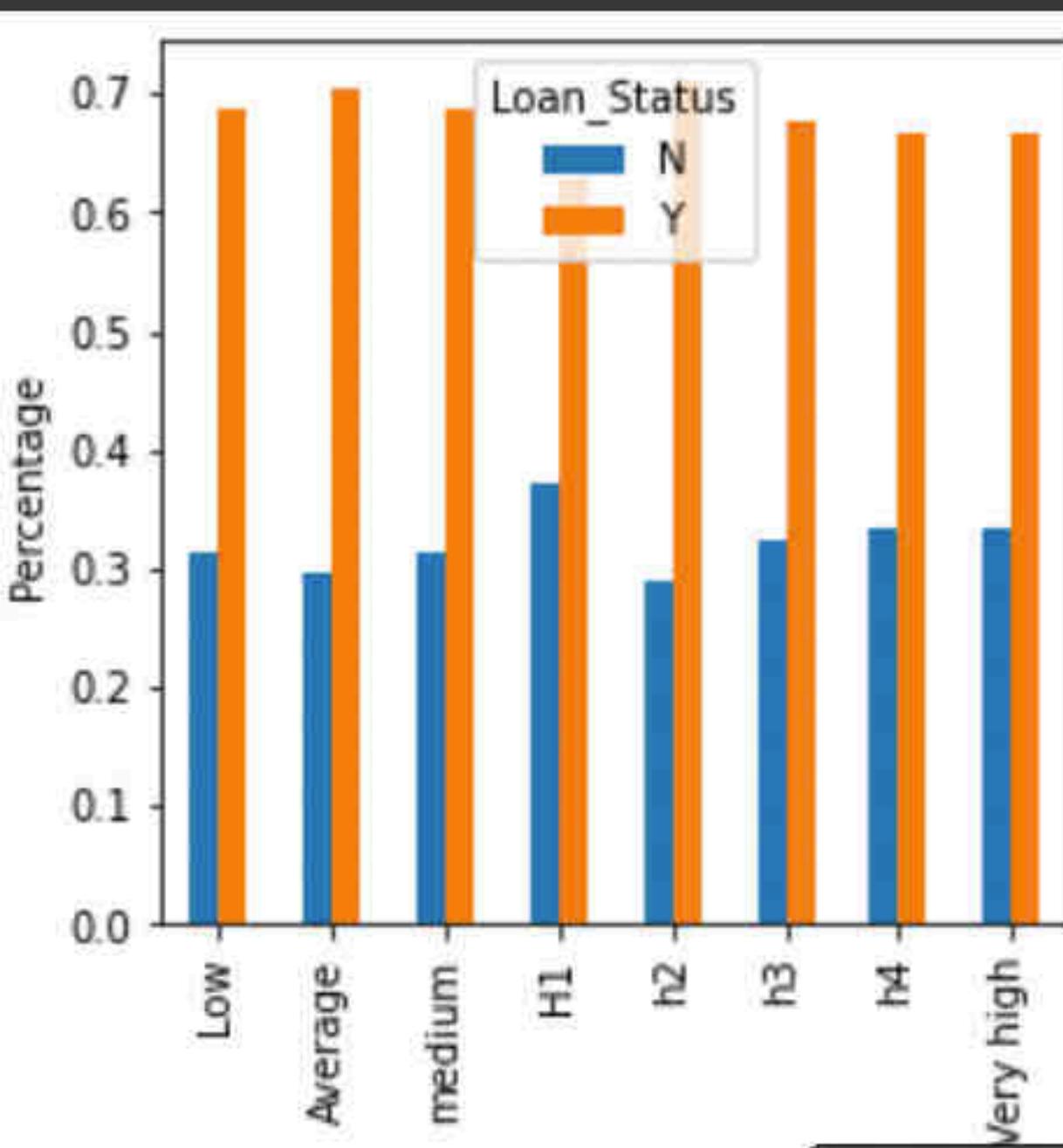
✓ RAM Disk

☆ 白手

1

```
Income_bin = pd.crosstab(data["Income_bin"], data["Loan_Status"])
Income_bin.div(Income_bin.sum(axis=1), axis=0).plot(kind="bar", figsize=(4,4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()
```

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per our analysis.



EDA_FE.ipynb - Colaboratory +

colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=EcFsZRhLK1DI

+ Code + Text

RAM Disk

Up Down Reload Settings Copy Paste

Income_bin = pd.crosstab(data["Income_bin"], data["Loan_Status"])

Income_bin.div(Income_bin.sum(axis=1), axis=0).plot(kind="bar", figsize=(4, 4))
plt.xlabel("ApplicantIncome")
plt.ylabel("Percentage")
plt.show()

#It can be inferred that Applicant income does not affect the chances of loan approval. Which seems wrong as per ou

Income Bin	N (%)	Y (%)
Low	~0.32	~0.68
Average	~0.30	~0.70
medium	~0.32	~0.68
H1	~0.38	~0.60
h2	~0.29	~0.68
h3	~0.33	~0.67
h4	~0.34	~0.66
Very high	~0.33	~0.67

80 / 80

Chrome File Edit View History Bookmarks Profiles Tab Window Help

EDA_FE.ipynb - Colaboratory x pandas.DataFrame.div – pand x +

colab.research.google.com/drive/1_P-nKswaKnx5v77lOx3pCxvJve73MvnN#scrollTo=EcFsZRhLK1DI

+ Code + Text RAM Disk

CoapplicantIncome_Bin = pd.crosstab(data["CoapplicantIncome_bin"],data["Loan_Status"]) CoapplicantIncome_Bin.div(CoapplicantIncome_Bin.sum(axis = 1),axis=0).plot(kind='bar',figsize=(4,4)) plt.xlabel("CoapplicantIncome") plt.ylabel("Percentage") plt.show()

What's the problem here? Why co-applicant having low income is getting maximum loan approved?

CoapplicantIncome	N	Y
Low	~0.12	~0.85
Average	~0.28	~0.72
High	~0.34	~0.68

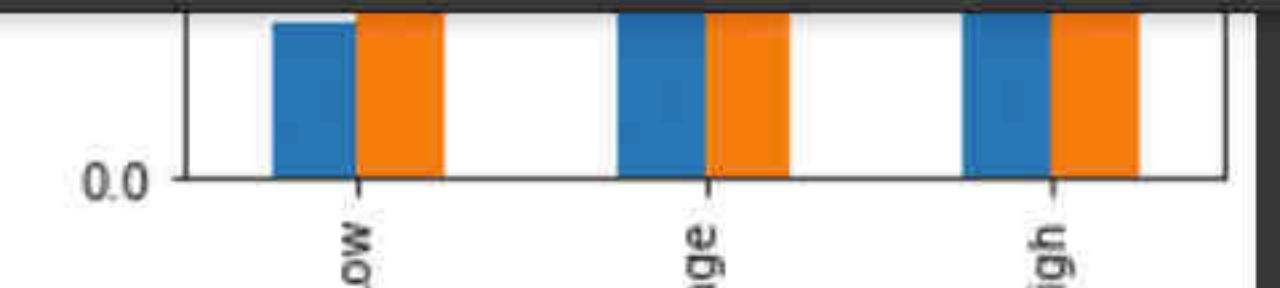
81 / 81

EDA_FE.ipynb - Colaboratory pandas.DataFrame.div – pandas

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=EcFsZRhLK1DI

+ Code + Text

RAM Disk

[108] 

{x}

[36] data['CoapplicantIncome'].value_counts().head()

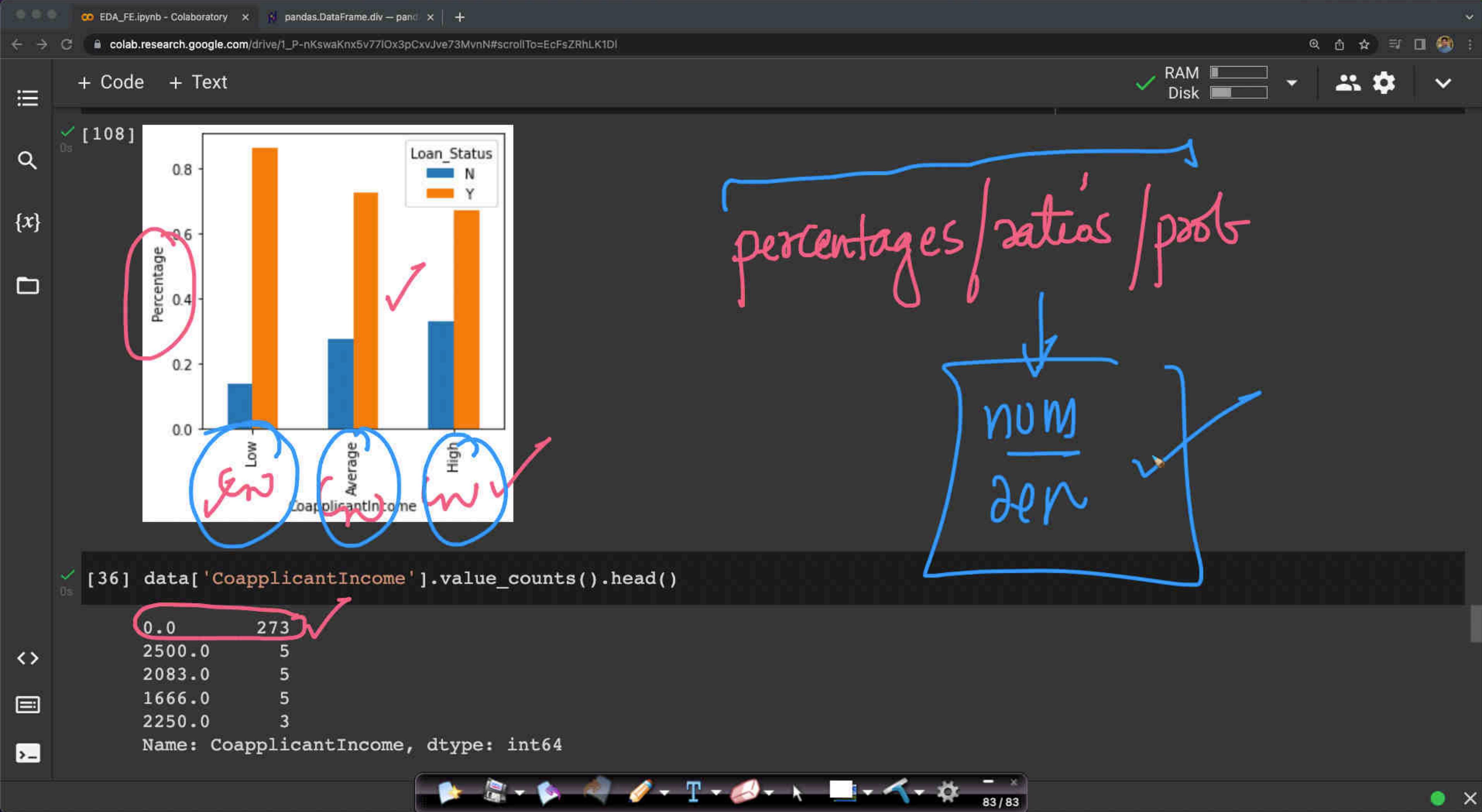
0.0 ← 273 ✓
2500.0 5
2083.0 5 }
1666.0 5
2250.0 3 }

Name: CoapplicantIncome, dtype: int64

[37] # New feature: total household income
data["TotalIncome"] = data["ApplicantIncome"] + data["CoapplicantIncome"]

[38] bins = [0, 2500, 4000, 6000, 81000]
group = ['Low', 'Average', 'High', 'Very High']
data["TotalIncome_bin"] = pd.cut(data["TotalIncome"], bins, labels=group)

[39] pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])



EDA_FE.ipynb - Colaboratory x pandas.DataFrame.div - pand x

colab.research.google.com/drive/1_P-nKswaKnx5v77jOx3pCxyJve73MvnN#scrollTo=qDO5prvhMK2

◎ 内 容 提 要

+ Code + Text

✓ RAM Disk

1

```
# New feature: total household income  
data["TotalIncome"] = data["ApplicantIncome"] + data["CoapplicantIncome"]
```

ta["To

ta["To

```
[38] bins = [0,2500,4000,6000,81000]
      group = ['Low','Average','High','Very High']
      data["TotalIncome_bin"] = pd.cut(data["TotalIncome"],bins,labels=group)
```

```
[39] pd.crosstab(data["TotalIncome bin"], data["Loan Status"])
```

Loan_Status	N	Y
TotalIncome_bin		
Low	14	10
Average	32	87
High	65	159
Very High	81	166

Ideas

1

90-95%

will fail

```
[41] TotalIncome = pd.crosstab(data["TotalIncome"], axis=0, columns=a["Loan_Status"])
    TotalIncome.div(TotalIncome.sum(axis = 1), axis=0).plot(kind='bar', figsize=(7,5))
```

EDA_FE.ipynb - Colaboratory pandas.DataFrame.div – pand

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=oDO5pryhMK21

+ Code + Text RAM Disk

New feature: total household income
data["TotalIncome"] = data["ApplicantIncome"] + data["CoapplicantIncome"]

{x} [38] bins = [0,2500,4000,6000,81000]
group = ['Low', 'Average', 'High', 'Very High']
data["TotalIncome_bin"] = pd.cut(data["TotalIncome"], bins, labels=group)

[39] pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])

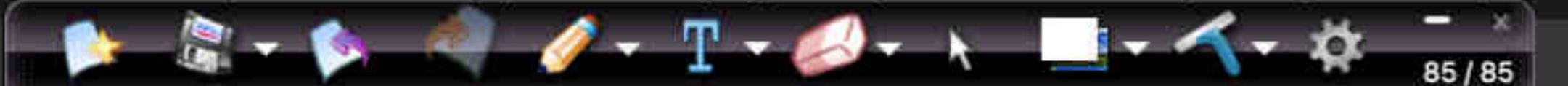
Loan_Status N Y

TotalIncome_bin

TotalIncome_bin	N	Y
Low	14	10
Average	32	87
High	65	159
Very High	81	166

[41] TotalIncome = pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])

TotalIncome.div(TotalIncome.sum(axis = 1), axis=0).plot(kind='bar', figsize=(7,5))



EDA_FE.ipynb - Colaboratory pandas.DataFrame.div – pandas

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=oDO5pryhMK21

+ Code + Text RAM Disk

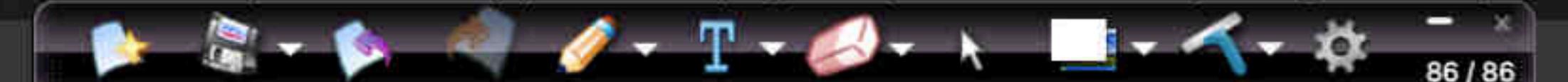
New feature: total household income
data["TotalIncome"] = data["ApplicantIncome"] + data["CoapplicantIncome"]

{x} [38] bins = [0, 2500, 4000, 6000, 81000]
group = ['Low', 'Average', 'High', 'Very High']
data["TotalIncome_bin"] = pd.cut(data["TotalIncome"], bins, labels=group)

[39] pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])

Loan_Status	N	Y
TotalIncome_bin		
Low	14	10
Average	32	87
High	65	159
Very High	81	166

[41] TotalIncome = pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])
TotalIncome.div(TotalIncome.sum(axis = 1), axis=0).plot(kind='bar', figsize=(7,5))



EDA_FE.ipynb - Colaboratory

pandas.DataFrame.div — pandas

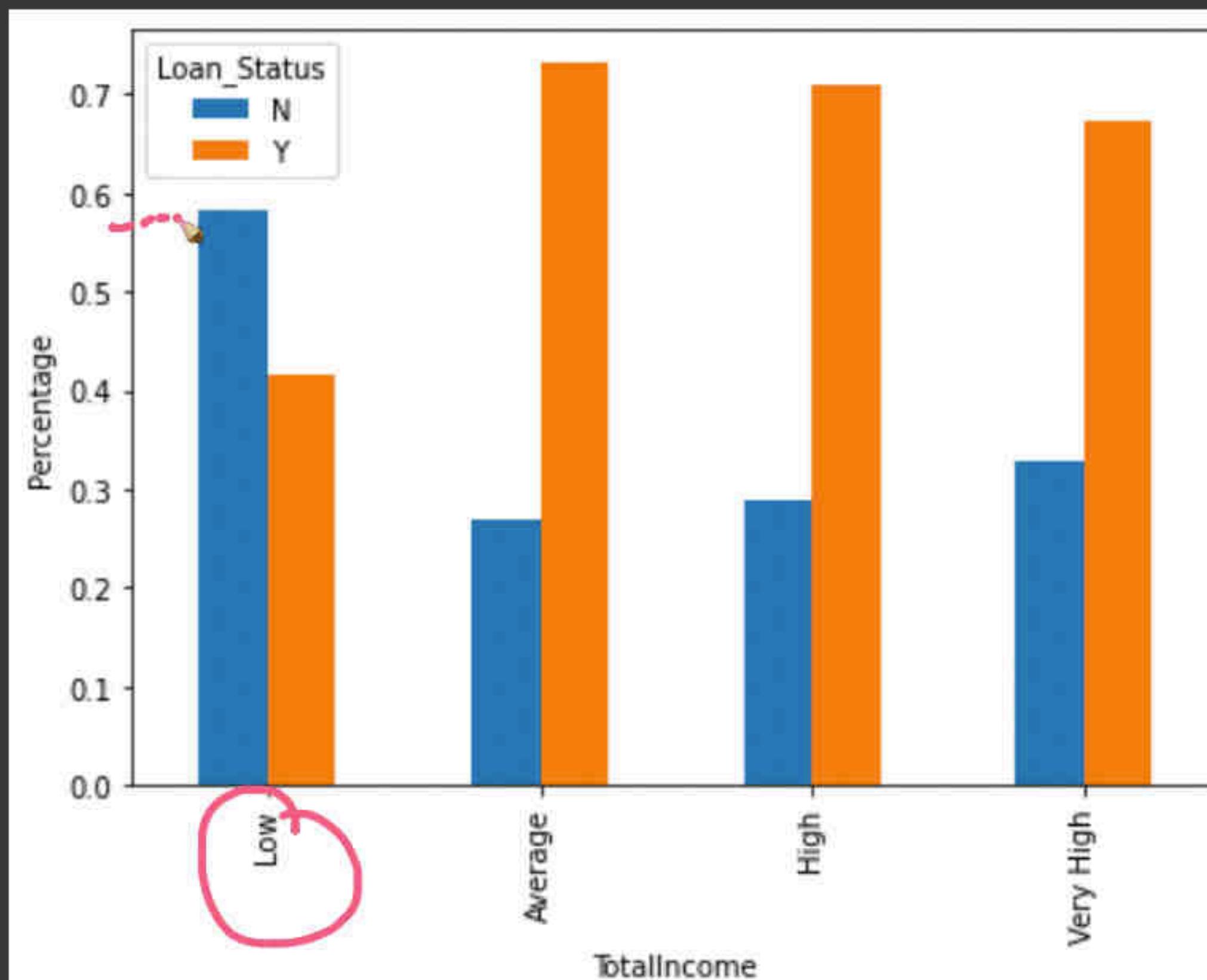
RAM
Disk

+ Code + Text

plt.show()

[41]

Observation: We can see that Proportion of loans getting approved for
applicants having low Total_Income is very less as compared to that of applicants
with Average, High and Very High Income.



EDA_FE.ipynb - Colaboratory pandas.DataFrame.div – pand +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Au_kXNNzMWID

+ Code + Text

RAM Disk

Up Down Reload Settings Copy Paste

```
plt.xlabel("TotalIncome")
plt.ylabel("Percentage")
plt.show()

# Observation: We can see that Proportion of loans getting approved for
# applicants having low Total_Income is very less as compared to that of applicants
# with Average, High and Very High Income.
```

Percentage

Loan_Status

N

Y

Low

Average

High

Very High

The chart displays the percentage of loans approved ('Y') and denied ('N') across four income levels: Low, Average, High, and Very High. The Y-axis represents the percentage from 0.0 to 0.7. The X-axis categories are Low, Average, High, and Very High. For each category, there are two bars: a blue bar for 'N' (Denied) and an orange bar for 'Y' (Approved). A red line with a circular arrow is drawn around the 'Low' category, highlighting it. A red circle also highlights the 'Low' label at the bottom of the X-axis.

TotalIncome	Loan_Status	Percentage
Low	N	~0.43
Low	Y	~0.58
Average	N	~0.31
Average	Y	~0.69
High	N	~0.29
High	Y	~0.72
Very High	N	~0.32
Very High	Y	~0.68

88 / 88

Chrome File Edit View History Bookmarks Profiles Tab Window Help

EDA_FE.ipynb - Colaboratory x pandas.DataFrame.div – pand x +

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Au_kXNNzMWID

+ Code + Text RAM Disk

[112] bins = [0,3000,5000,8000,81000]
group = ['Low', 'Average', 'High', 'Very High']
data["TotalIncome_bin"] = pd.cut(data["TotalIncome"], bins, labels=group)

[113] pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])

		N	Y	
TotalIncome_bin	Low	20	27	
Average		69	154	
High		61	151	
Very High		42	90	

20
47 ;
27
47 } ✓

TotalIncome = pd.crosstab(data["TotalIncome_bin"], data["Loan_Status"])
TotalIncome.div(TotalIncome.sum(axis = 1), axis=0).plot(kind='bar', figsize=(7,5))
plt.xlabel("TotalIncome")
plt.ylabel("Percentage")
plt.show()



88 FDA_EE.ipynb - Colaboratory x 89 pandas DataFrame div — pandas x

https://colab.research.google.com/drive/1-R-pKswaKpx5v7zIQx3pCxyJye73MvpN#scrollTo=AuI_kYNNzMW

◎ 金言

Code # Text

```
plt.xlabel('Total Income')  
plt.ylabel("Percentage")  
plt.show()
```

[2]

100

1

A grouped bar chart titled "Percentage vs TotalIncome". The y-axis is labeled "Percentage" and ranges from 0.0 to 0.7. The x-axis is labeled "TotalIncome" and has four categories: "Low", "Average", "High", and "Very High". For each category, there are two bars: a blue bar representing "Loan_Status N" and an orange bar representing "Loan_Status Y".

TotalIncome	Loan_Status N	Loan_Status Y
Low	0.42	0.58
Average	0.31	0.69
High	0.29	0.72
Very High	0.32	0.69

$$f_{\text{new}} = f_{\text{inc}} + \underline{f_{\text{coapp-inc}}}$$

EDA_FE.ipynb - Colaboratory pandas.DataFrame.div – pandas

colab.research.google.com/drive/1_P-nKswaKnx5v77IOx3pCxvJve73MvnN#scrollTo=Au_kXNNzMWID

+ Code + Text

RAM Disk

Up Down Refresh Stop Save : Handwriting

Code:

```
plt.xlabel('TotalIncome')
plt.ylabel("Percentage")
plt.show()
```

{x}

Observation: We can see that Proportion of loans getting approved for
applicants having low Total_Income is very less as compared to that of applicants
with Average, High and Very High Income.

Figure:

A bar chart comparing the percentage of loans approved ('Y') versus denied ('N') across four income levels: Low, Average, High, and Very High. The Y-axis represents the percentage from 0.0 to 0.7. The X-axis represents the total income levels. For each income level, there are two bars: a blue bar for 'N' (Denied) and an orange bar for 'Y' (Approved). The chart shows that the percentage of approved loans increases significantly as total income increases, with the highest percentage of approved loans occurring at the 'High' income level.

TotalIncome	N (%)	Y (%)
Low	~0.43	~0.58
Average	~0.31	~0.69
High	~0.29	~0.72
Very High	~0.32	~0.68

Task: Y or N

Q&A

$\tilde{d} = \# \text{features} : \underline{\underline{1000}}$

Stats: \Rightarrow
 $\Rightarrow 10^3 \text{ of}$
features

(1000) \rightarrow DA \rightarrow ML-models

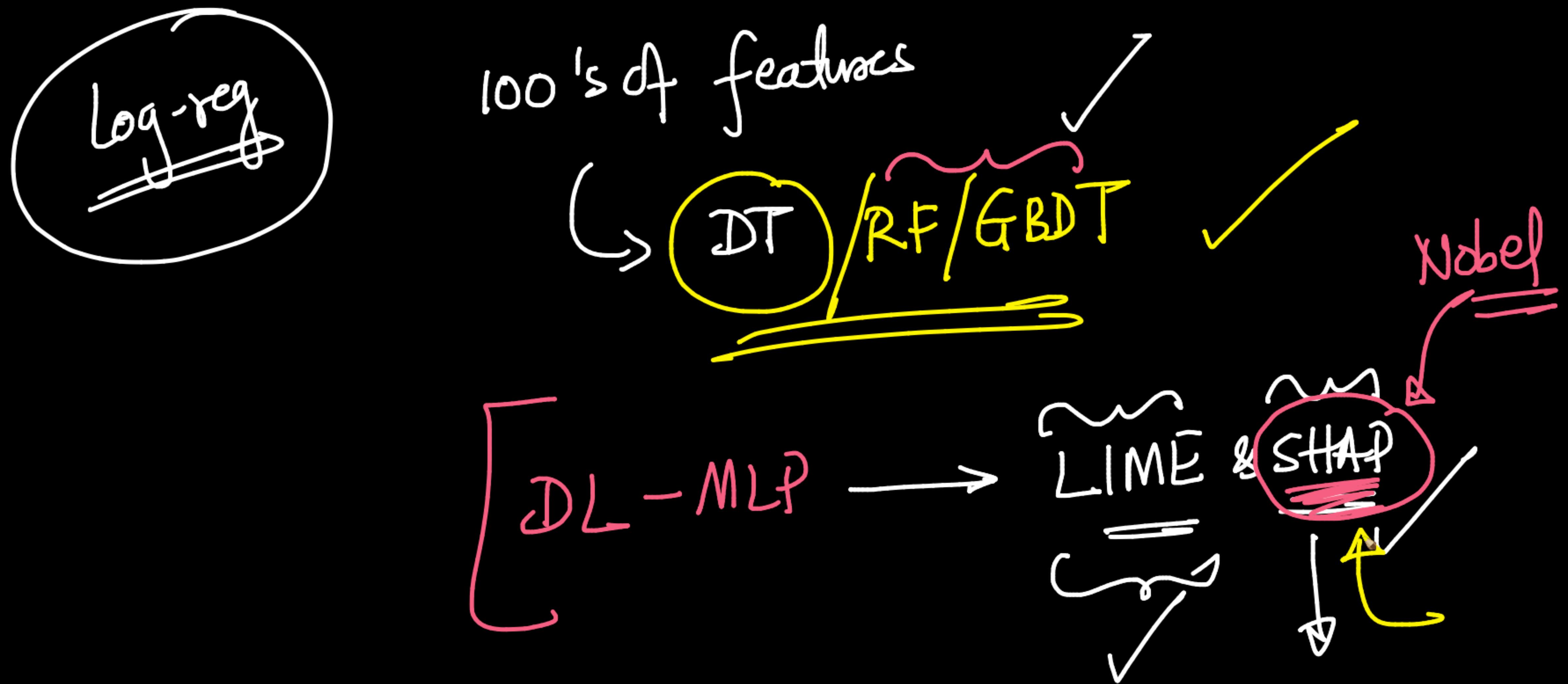
cat \rightarrow numerical

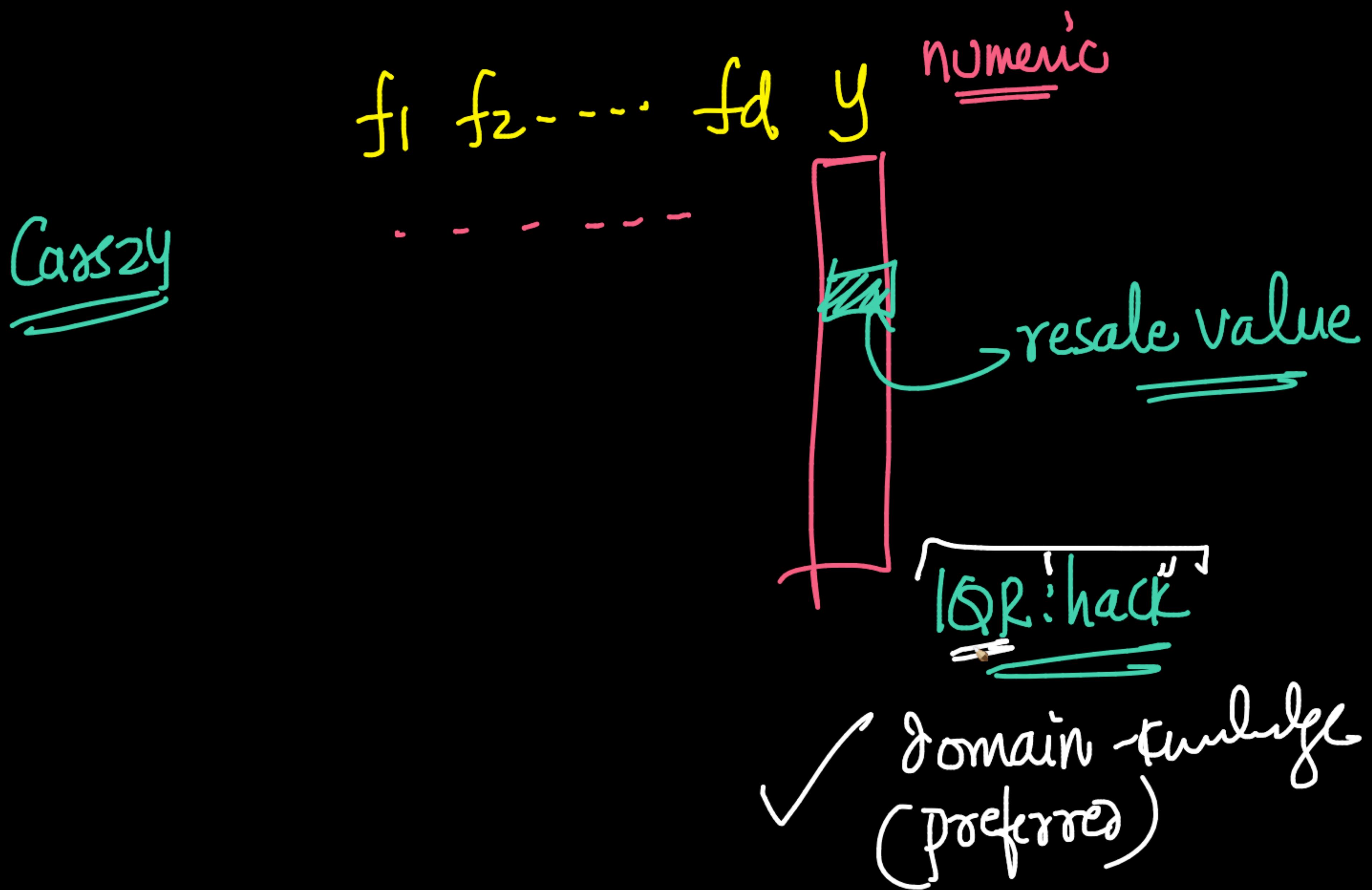
\rightarrow SRCC \rightarrow Univariate analysis

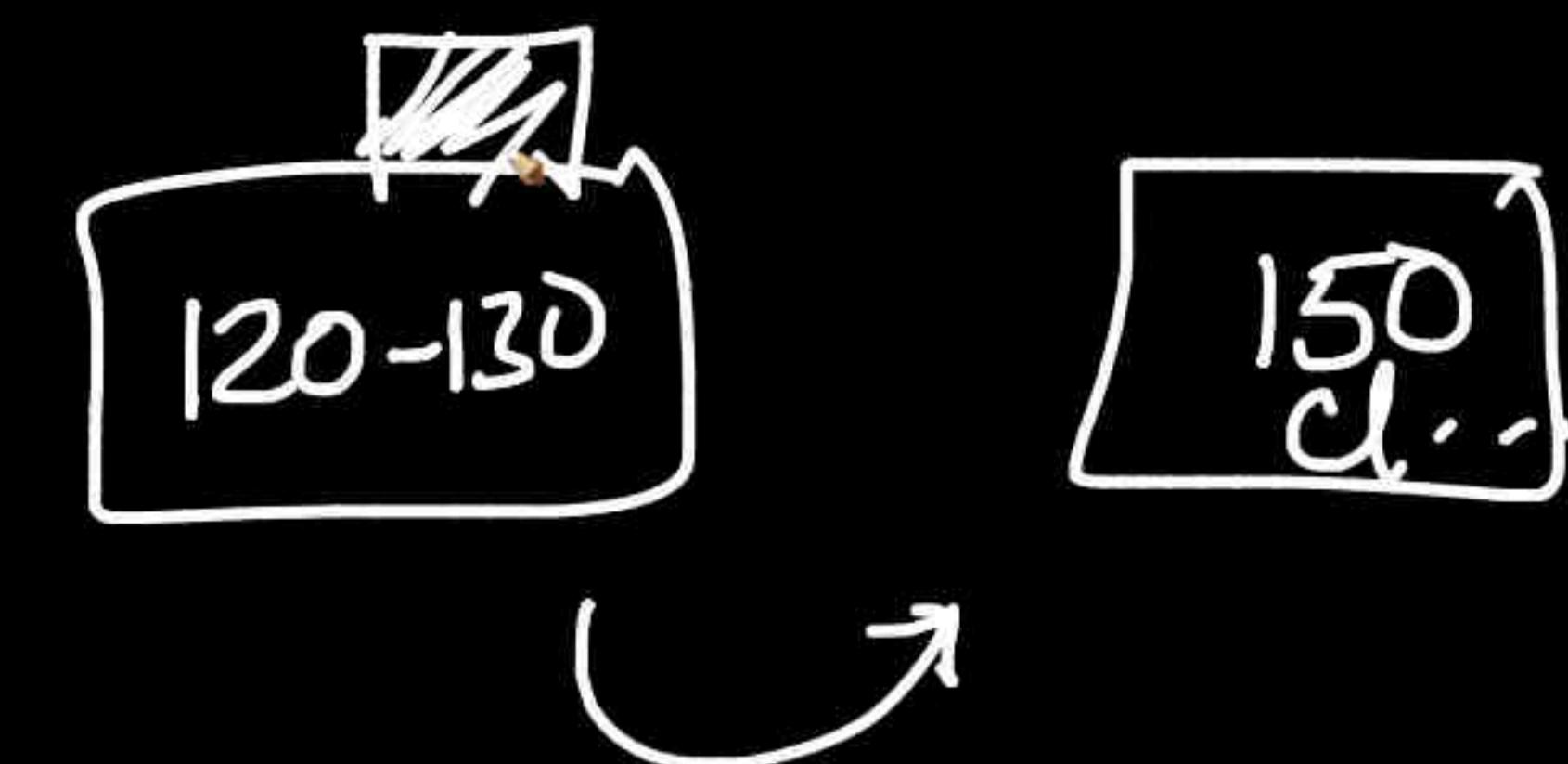
f_i, y

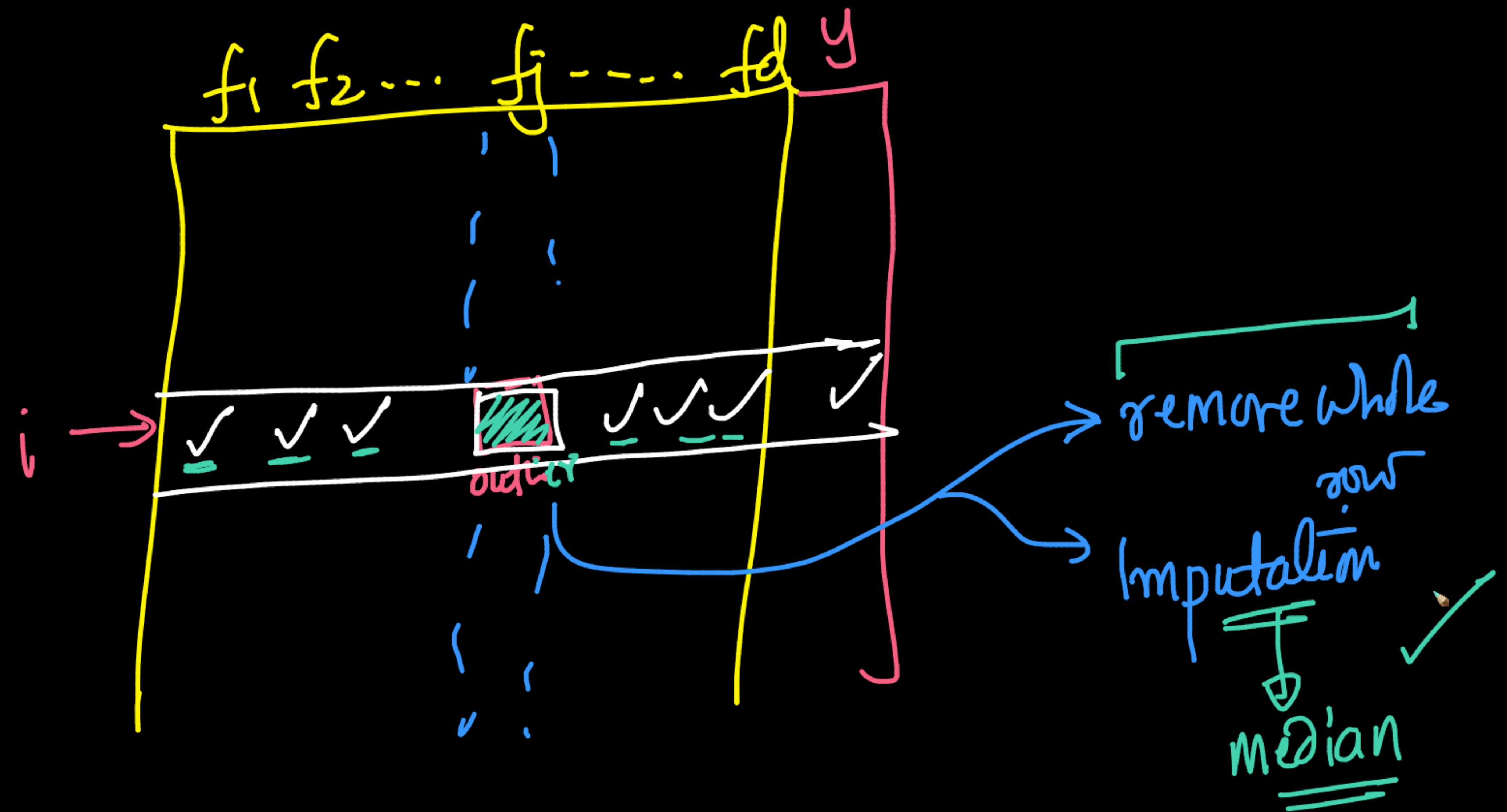
NOT

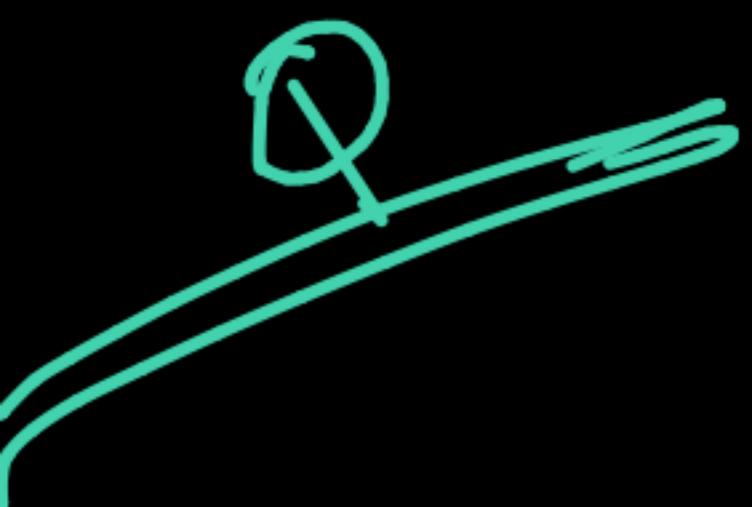
$f_i, f_j \text{ vs } y$











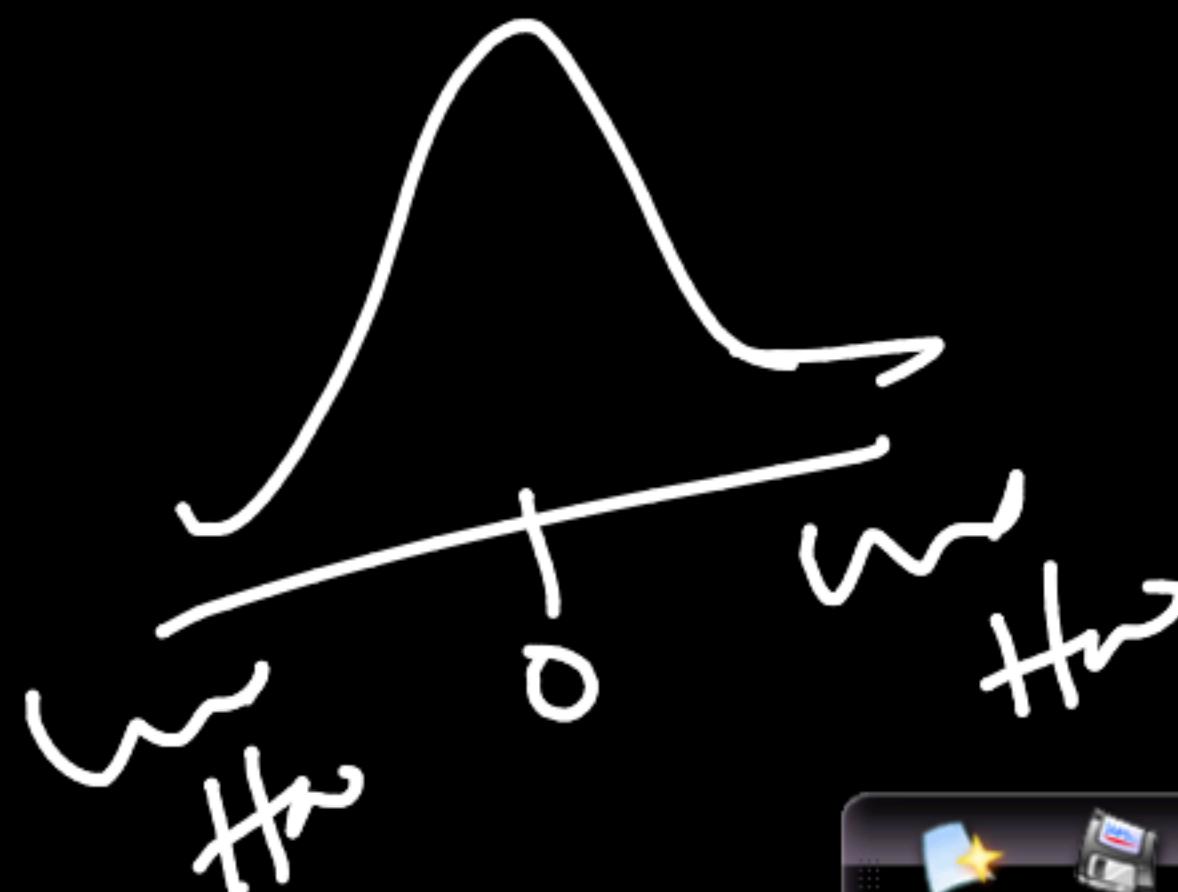
$$\text{mean-inc-} M = 2000 \text{ \$}$$

||

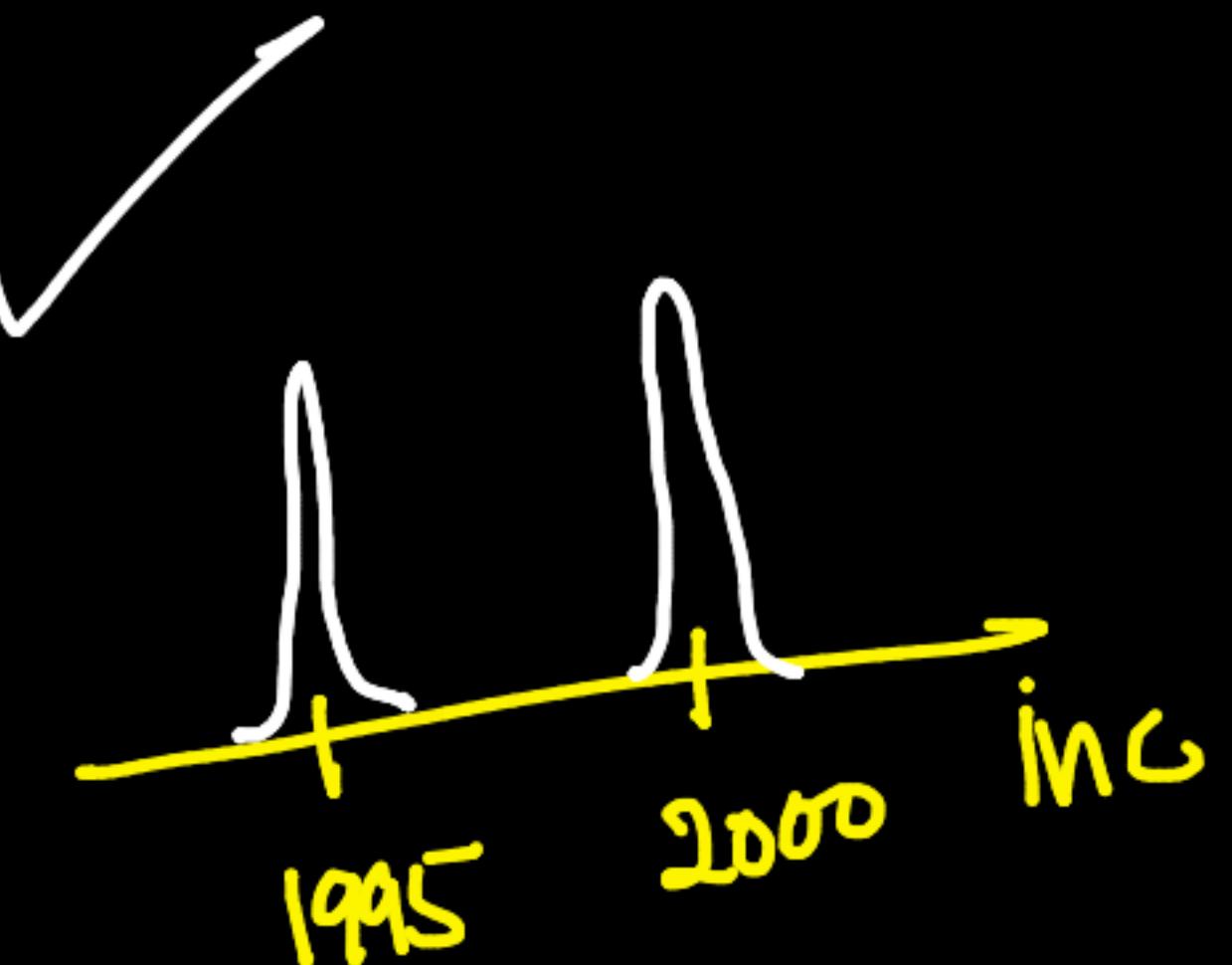
no outliers

$$\text{mean-inc-f} = 1995 \text{ \$}$$

$\hat{\sigma}_1$ & $\hat{\sigma}_2$
are V.V small

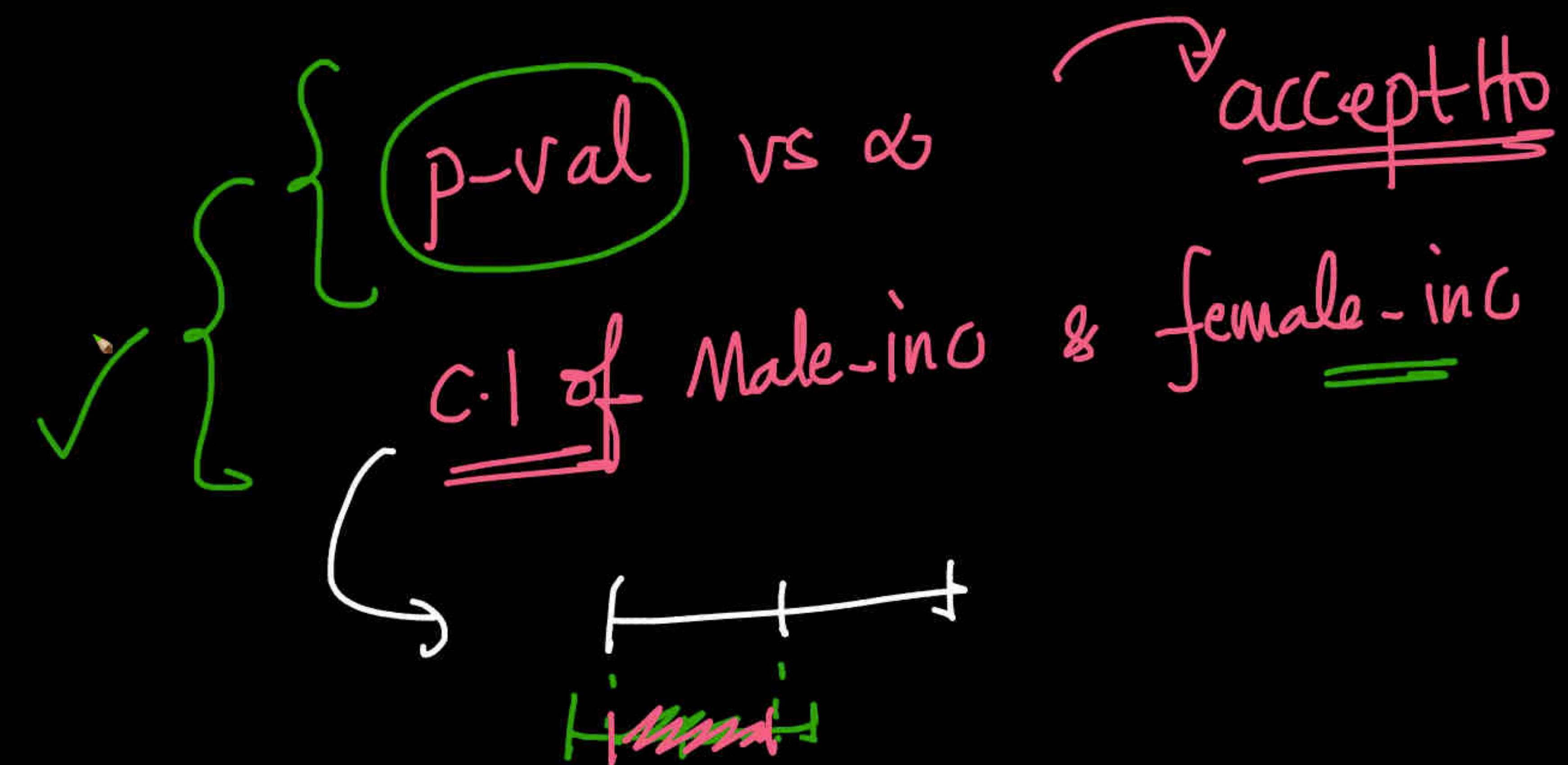


n_1 & n_2 are large ✓



$$\frac{M_1 - M_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Q



EDA_FE.ipynb - Colaboratory | pandas.DataFrame.div — pandas | pandas.DataFrame.describe — pandas | pandas.DataFrame.corr — pandas | +

pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

pandas

Getting started User Guide API reference Development Release notes 1.4.2

[\[source\]](#)

pandas.DataFrame.corr

`DataFrame.corr(method='pearson', min_periods=1)`

Compute pairwise correlation of columns, excluding NA/null values.

Parameters: `method : {'pearson', 'kendall', 'spearman'} or callable`

Method of correlation:

- `pearson` : standard correlation coefficient
- `kendall` : Kendall Tau correlation coefficient
- `spearman` : Spearman rank correlation
- ~~callable~~: callable with input two 1d ndarrays and returning a float. Note that the returned matrix from corr will have 1 along the diagonals and will be symmetric regardless of the callable's behavior.

min_periods : int, optional

Minimum number of observations required per pair of columns to have a valid result. Currently only available for Pearson and Spearman correlation.

Returns: `DataFrame`

Correlation matrix.

See also

`DataFrame.corrwith`

Compute pairwise correlation with another DataFrame or Series.

Education :- Cat 1,2,3,4

Income :- NUM

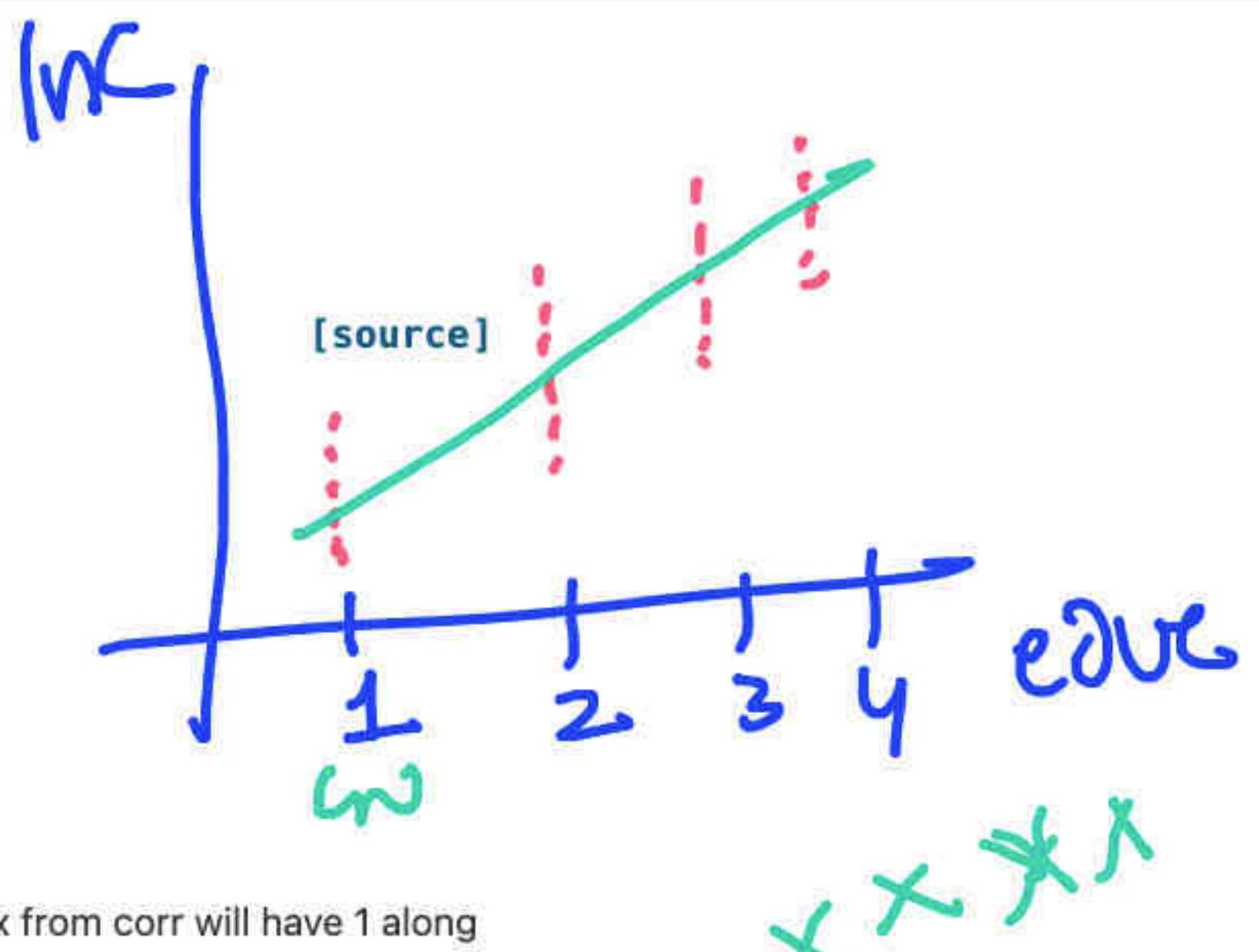
PDF/QF

1 2 3 4

HC

EDA_FE.ipynb - Colaboratory | pandas.DataFrame.div — pandas | pandas.DataFrame.describe — pandas | pandas.DataFrame.corr — pandas | +

pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html



pandas

Getting started User Guide API reference Development Release notes 1.4.2

[pandas.DataFrame.expanding](#)

[pandas.DataFrame.ewm](#)

[pandas.DataFrame.abs](#)

[pandas.DataFrame.all](#)

[pandas.DataFrame.any](#)

[pandas.DataFrame.clip](#)

[pandas.DataFrame.corr](#)

[pandas.DataFrame.corrwith](#)

[pandas.DataFrame.count](#)

[pandas.DataFrame.cov](#)

[pandas.DataFrame.cummax](#)

[pandas.DataFrame.cummin](#)

[pandas.DataFrame.cumprod](#)

[pandas.DataFrame.cumsum](#)

[pandas.DataFrame.describe](#)

[pandas.DataFrame.diff](#)

[pandas.DataFrame.eval](#)

[pandas.DataFrame.kurt](#)

[pandas.DataFrame.kurtosis](#)

[pandas.DataFrame.mad](#)

[pandas.DataFrame.max](#)

[pandas.DataFrame.mean](#)

[pandas.DataFrame.median](#)

[pandas.DataFrame.min](#)

pandas.DataFrame.corr

`DataFrame.corr(method='pearson', min_periods=1)`

Compute pairwise correlation of columns, excluding NA/null values.

Parameters: `method : {'pearson', 'kendall', 'spearman'} or callable`

Method of correlation:

- `pearson` : standard correlation coefficient
- `kendall` : Kendall Tau correlation coefficient
- `spearman` : Spearman rank correlation
- `callable`: callable with input two 1d ndarrays and returning a float. Note that the returned matrix from corr will have 1 along the diagonals and will be symmetric regardless of the callable's behavior.

min_periods : int, optional

Minimum number of observations required per pair of columns to have a valid result. Currently only available for Pearson and Spearman correlation.

Returns: `DataFrame`

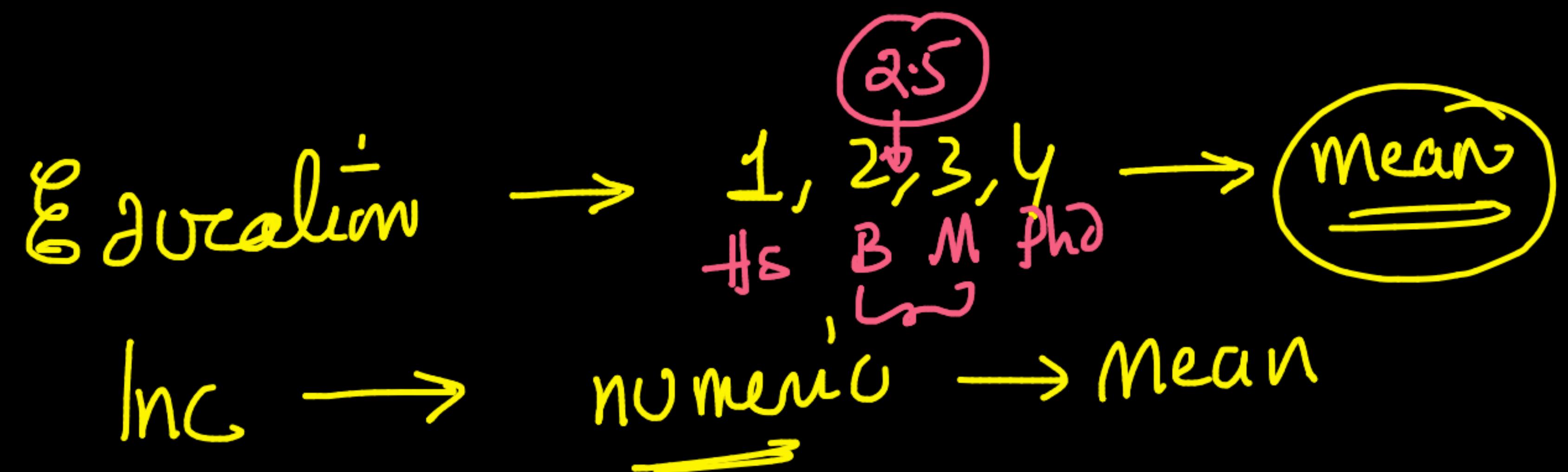
Correlation matrix.

See also

[DataFrame.corrwith](#)

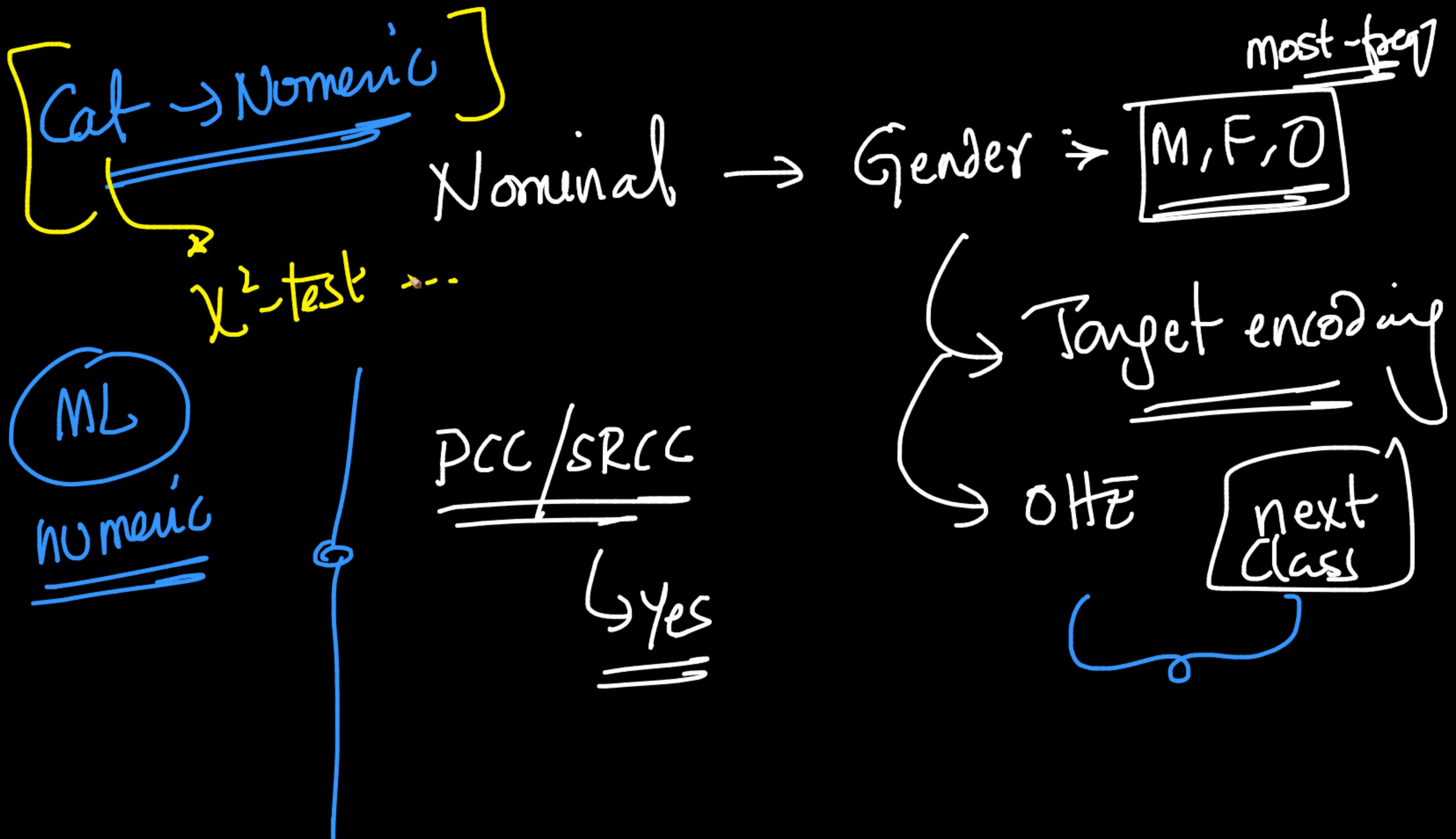
Compute pairwise correlation with another DataFrame or Series.

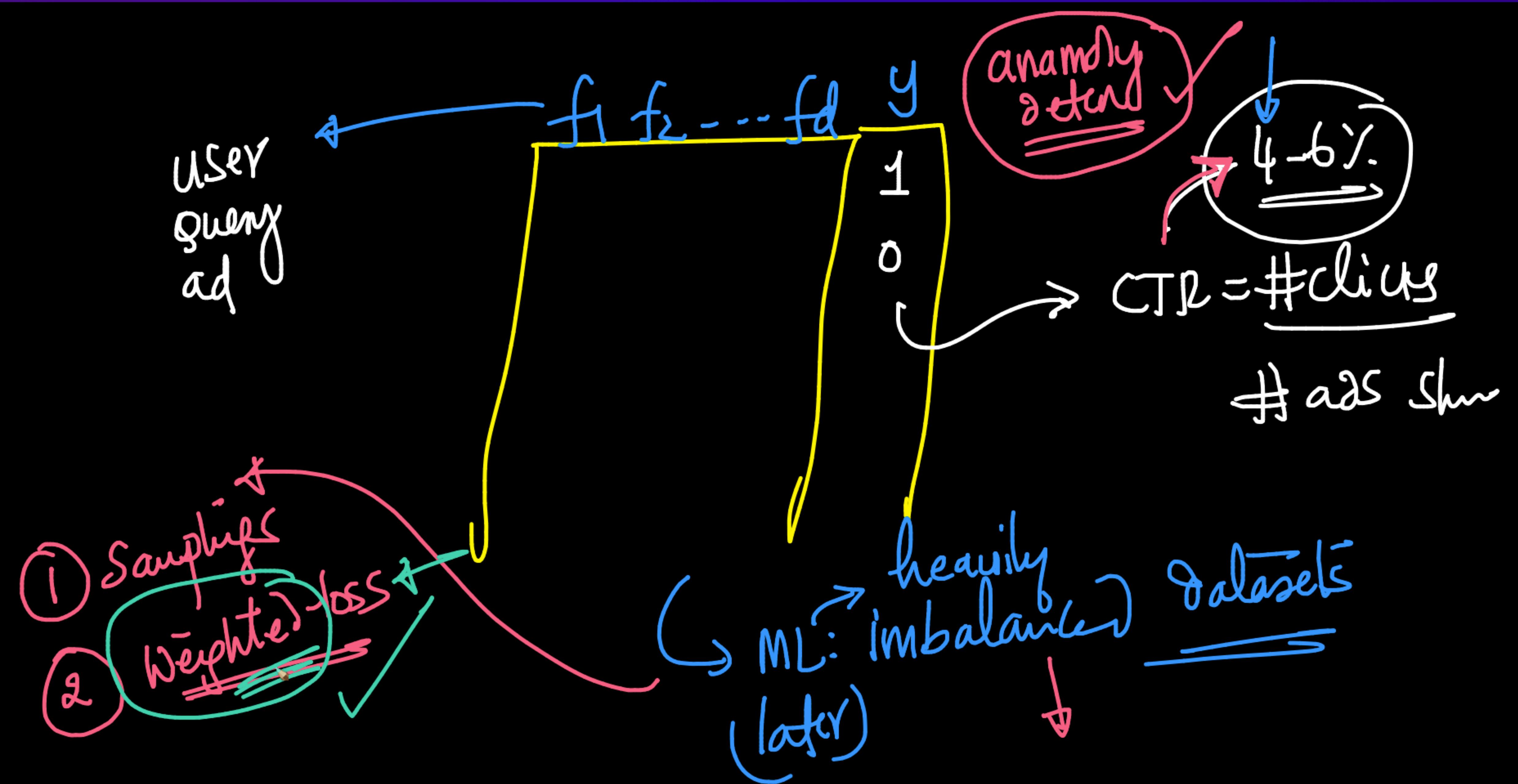
100 / 101

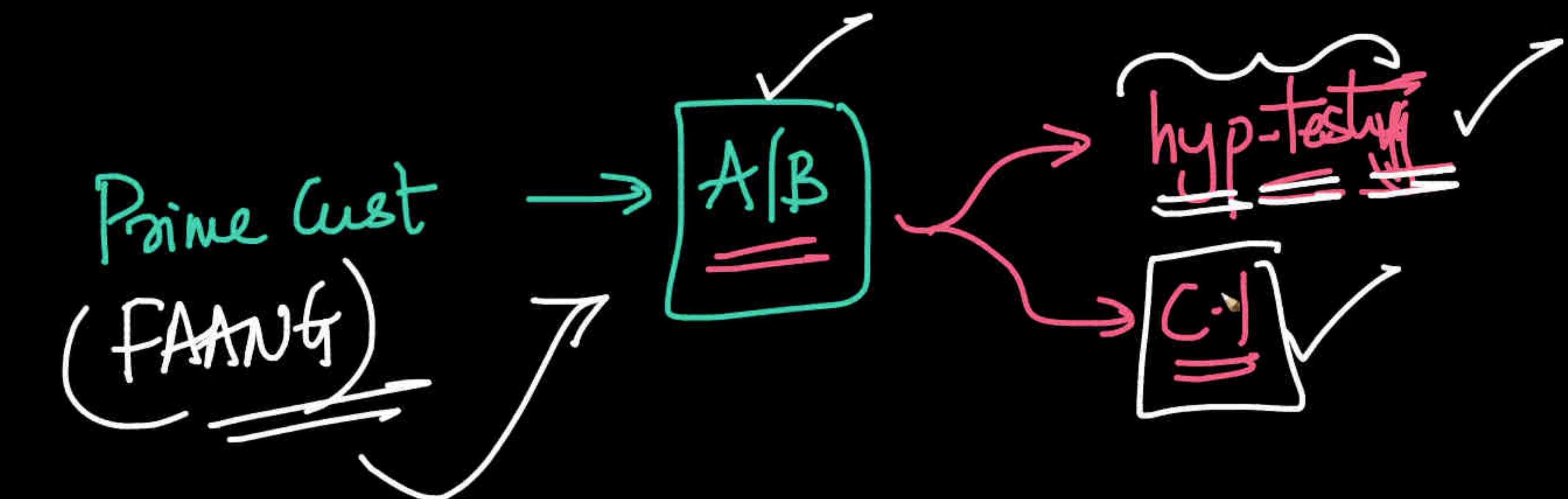


Task → Education corr with incomes

↓
Corr. Coeff 0-1



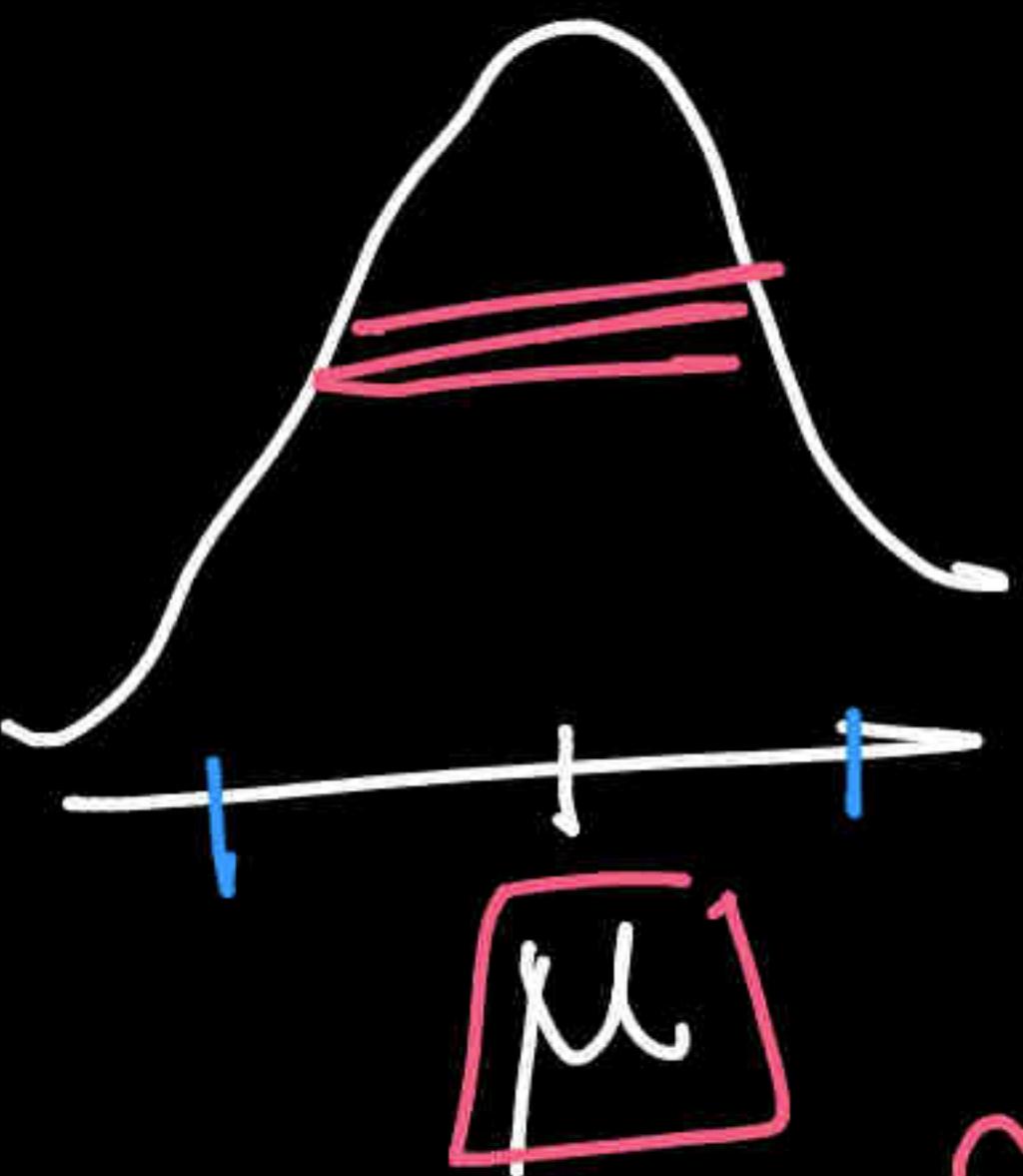




St. Dev. of the
distrib. of sample
means

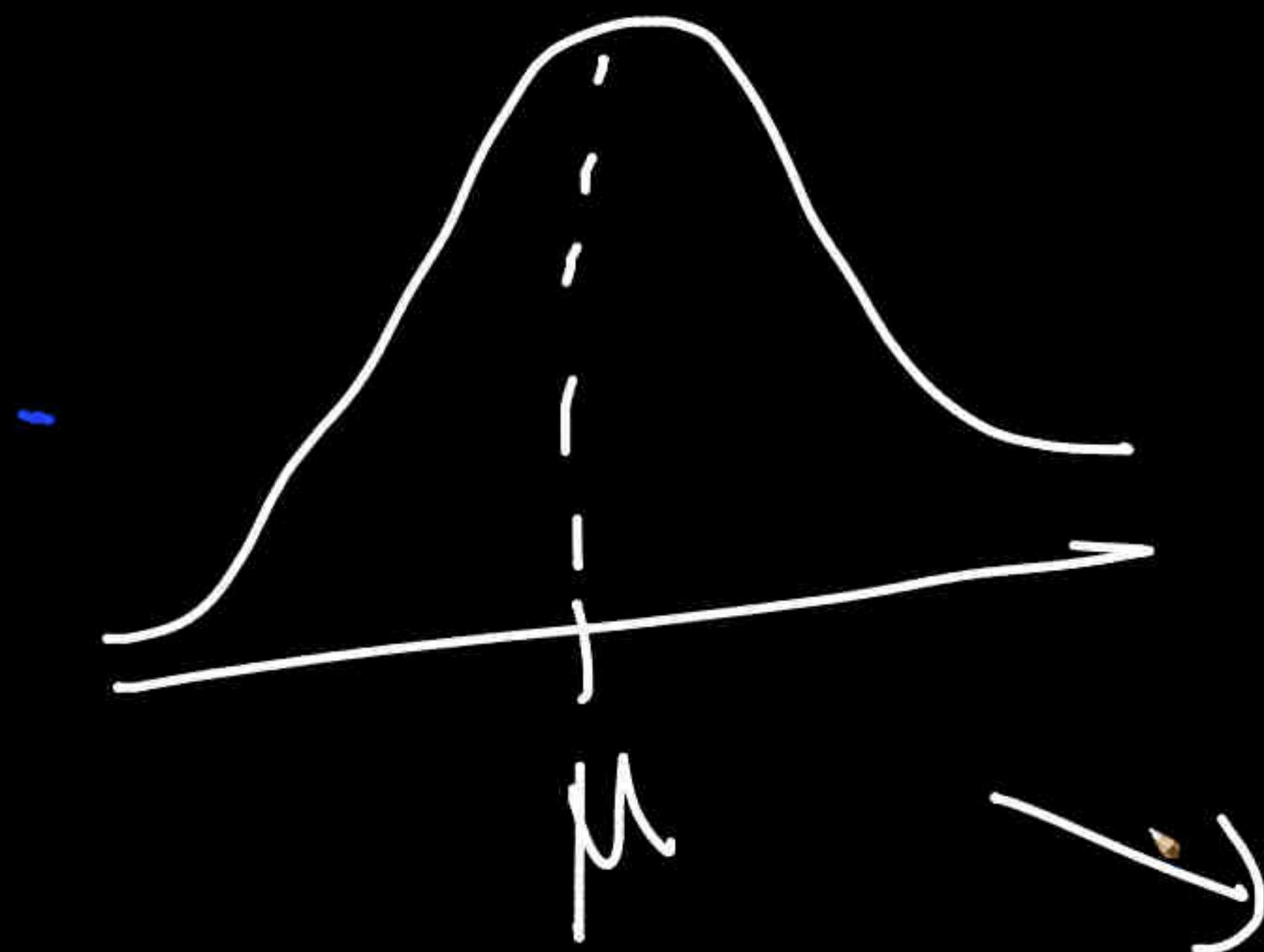
CLT:

$$\left. \begin{array}{l} S_1 - M_1 \\ S_2 - M_2 \\ \vdots \quad \vdots \\ S_n - M_n \end{array} \right\}$$



$$St. Dev = \frac{\sigma}{\sqrt{n}} \text{ as } n \rightarrow \infty$$

disb. of sample means (M_i)



$$\text{mean} = \mu$$

$$\text{std-dev} = \frac{\sigma}{\sqrt{n}}$$

$$M_1, M_2, \dots, M_n$$

$M_i \sim N(\text{mean}, \text{st})$

$M_i - \text{mean}(M_i) = M_i'$

$\text{st}(M_i)$

$M_i' \sim Z(0, 1)$

✓

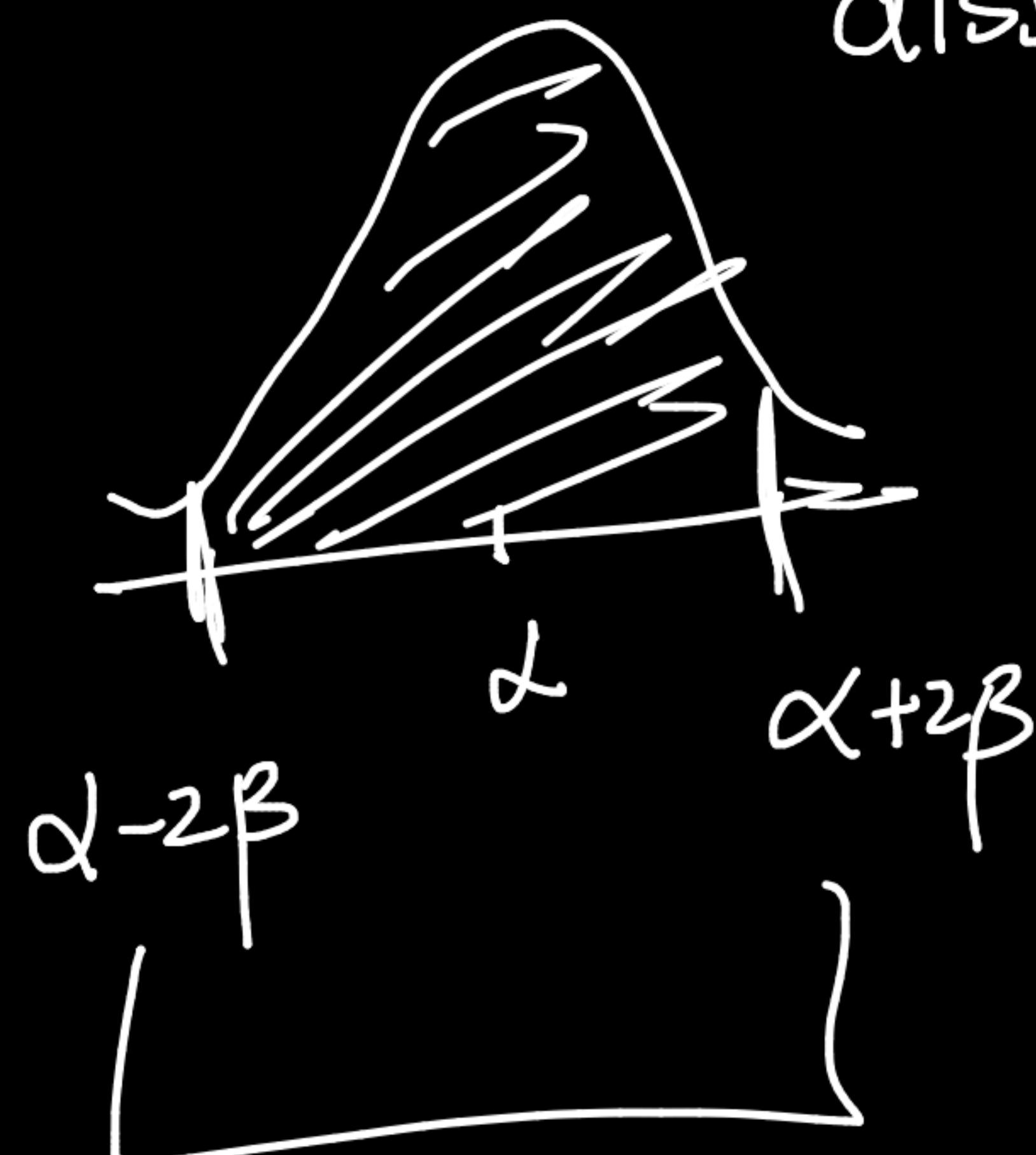
Gaussian dist (μ, σ)



Standardize



$$\underline{z(\theta_{11})} = N(\theta_{11})$$



dist of sample means
($M_i S$)

$$\text{mean} (M_i S) = d$$

$$\text{std} (M_i S) = \beta$$

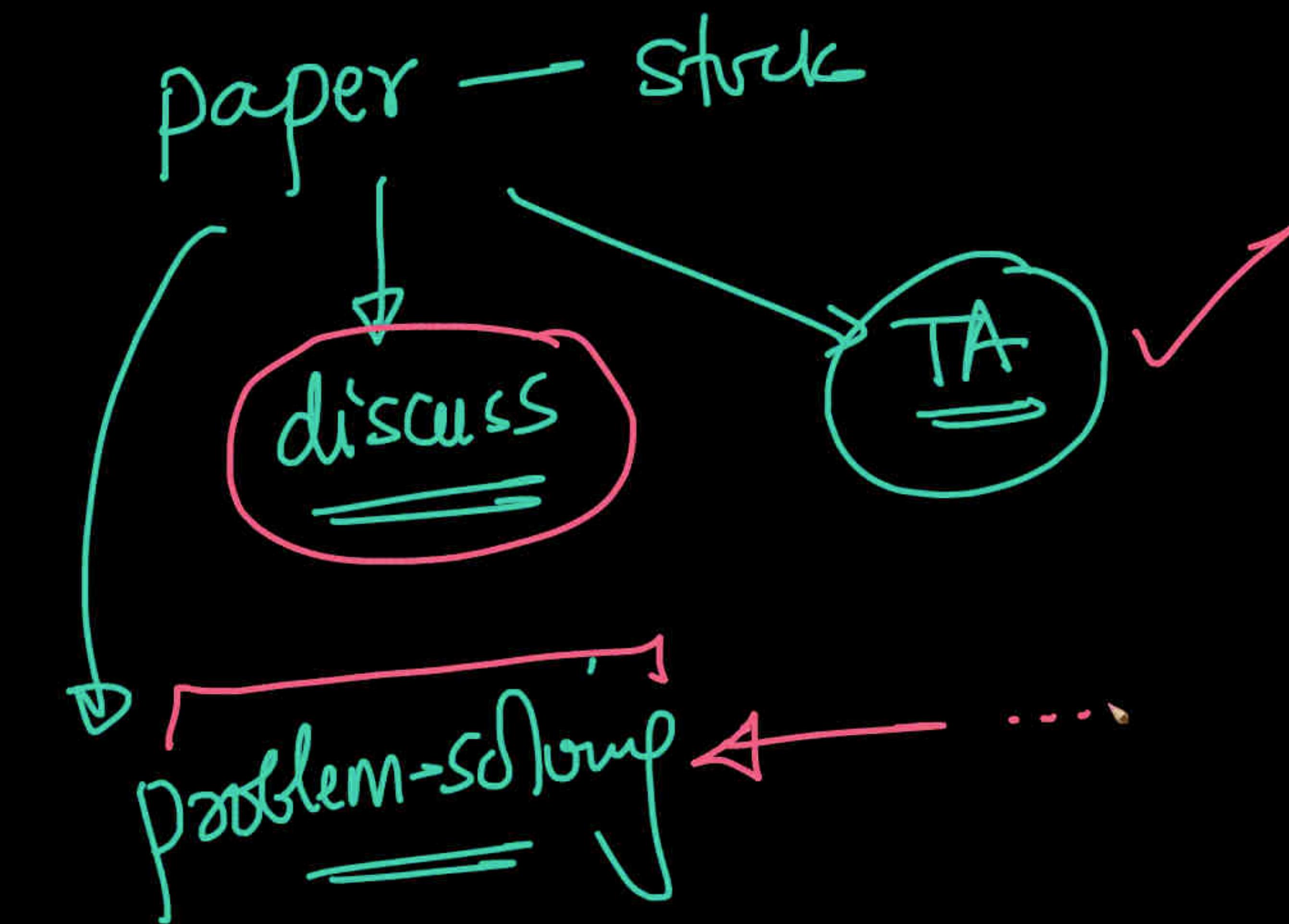
$$d \approx \mu \xrightarrow{\text{POP}}$$
$$\beta \approx \frac{\sigma}{\sqrt{n}}$$

2-way ANOVA

$d_1, d_2, d_3 \}$

$A, O, K \}$

2 classes





Search the docs ...

- [numpy.nanpercentile](#)
- [numpy.quantile](#)
- [numpy.nanquantile](#)
- [numpy.median](#)
- [numpy.average](#)
- [numpy.mean](#)
- [numpy.std](#)
- [numpy.var](#)
- [numpy.nanmedian](#)
- [numpy.nanmean](#)
- [numpy.nanstd](#)
- [numpy.nanvar](#)
- [numpy.corrcoef](#)
- [numpy.correlate](#)
- [numpy.cov](#)
- [numpy.histogram](#)
- [numpy.histogram2d](#)
- [numpy.histogramdd](#)
- [numpy.bincount](#)
- [numpy.histogram_bin_edges](#)
- [numpy.digitize](#)

numpy.std

`numpy.std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=<no value>, *, where=<no value>)` [\[source\]](#)

Compute the standard deviation along the specified axis.

Returns the standard deviation, a measure of the spread of a distribution, of the array elements. The standard deviation is computed for the flattened array by default, otherwise over the specified axis.

Parameters: `a : array_like`

Calculate the standard deviation of these values.

`axis : None or int or tuple of ints, optional`

Axis or axes along which the standard deviation is computed. The default is to compute the standard deviation of the flattened array.

New in version 1.7.0.

If this is a tuple of ints, a standard deviation is performed over multiple axes, instead of a single axis or all the axes as before.

`dtype : dtype, optional`

Type to use in computing the standard deviation. For arrays of integer type the default is float64, for arrays of float types it is the same as the array type.



112 / 112

as the expected output but the type (of the calculated values) will be cast if necessary.