Topics:

Prev: Problems; mean, median

Retail data →
- mode ; weighted-mean
- std ; variance; MAD
- percentiles; Quantiles; IQR & outliers
- histogram; density-plots
- box plot; bar plot; scatterplot

} plots

{ Airlines booking } →
- Probability distributions →

mean: average

median: middle-value

Mode:

↓

{ most frequently occuring value

discrete r.v

— code —

```
[ ] #mode
    df["Education"].value_counts()
```

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```

Education: discrete r.v (Categorical)

↳ ordinal

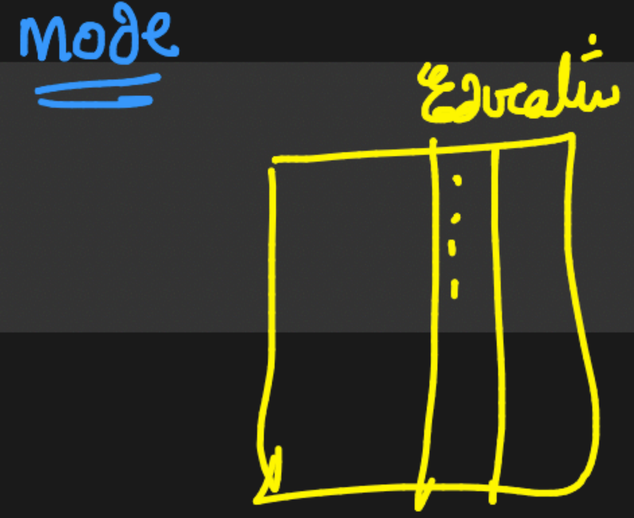
```
[ ] #variance
    print("Gold:",df['MntGoldProds'].std())
    print("Fruits:",df['MntFruits'].std())
    print("Sweets:",df['MntSweetProducts'].std())
    print("Wine:",df['MntWines'].std())
    print("Meat:",df['MntMeatProducts'].std())
    print("Fish:",df['MntFishProducts'].std())
```

```
Gold: 52.167438914997064
Fruits: 39.77343376457871
Sweets: 41.2804984878548
Wine: 336.5973926053717
Meat: 225.71537251175445
Fish: 54.62897940287769
```

```
[10] from scipy import stats
     print(stats.median_absolute_deviation(df['MntGoldProds']))
```

```
[ ] #mode
    df["Education"].value_counts()
```

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```




```
[ ] #variance
    print("Gold:",df['MntGoldProds'].std())
    print("Fruits:",df['MntFruits'].std())
    print("Sweets:",df['MntSweetProducts'].std())
    print("Wine:",df['MntWines'].std())
    print("Meat:",df['MntMeatProducts'].std())
    print("Fish:",df['MntFishProducts'].std())
```

```
Gold: 52.167438914997064
Fruits: 39.77343376457871
Sweets: 41.2804984878548
Wine: 336.5973926053717
Meat: 225.71537251175445
Fish: 54.62897940287769
```

```
[10] from scipy import stats
     print(stats.median_absolute_deviation(df['MntGoldProds']))
```

```
[ ] #mode
    df["Education"].value_counts()
```

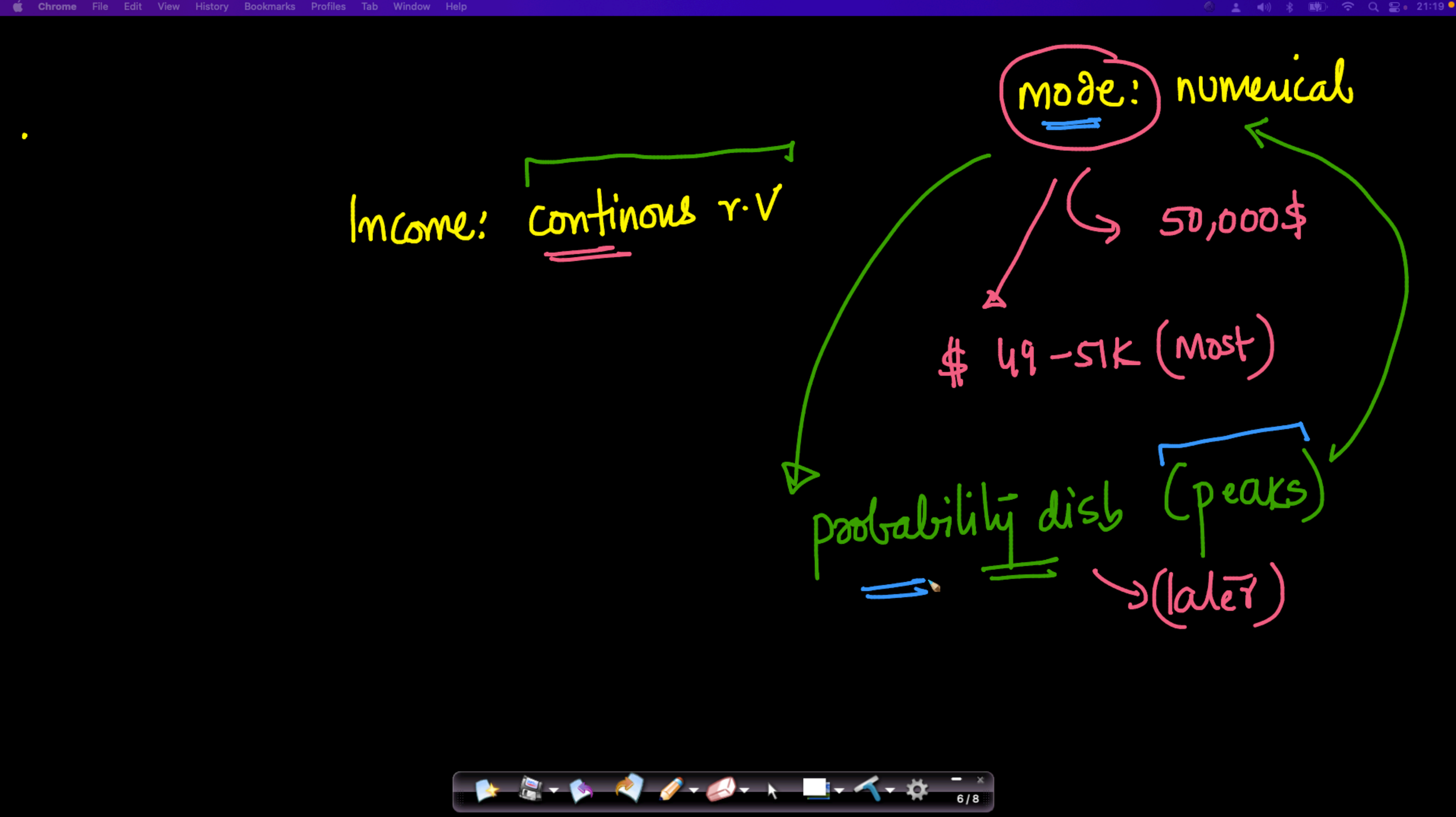
```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```

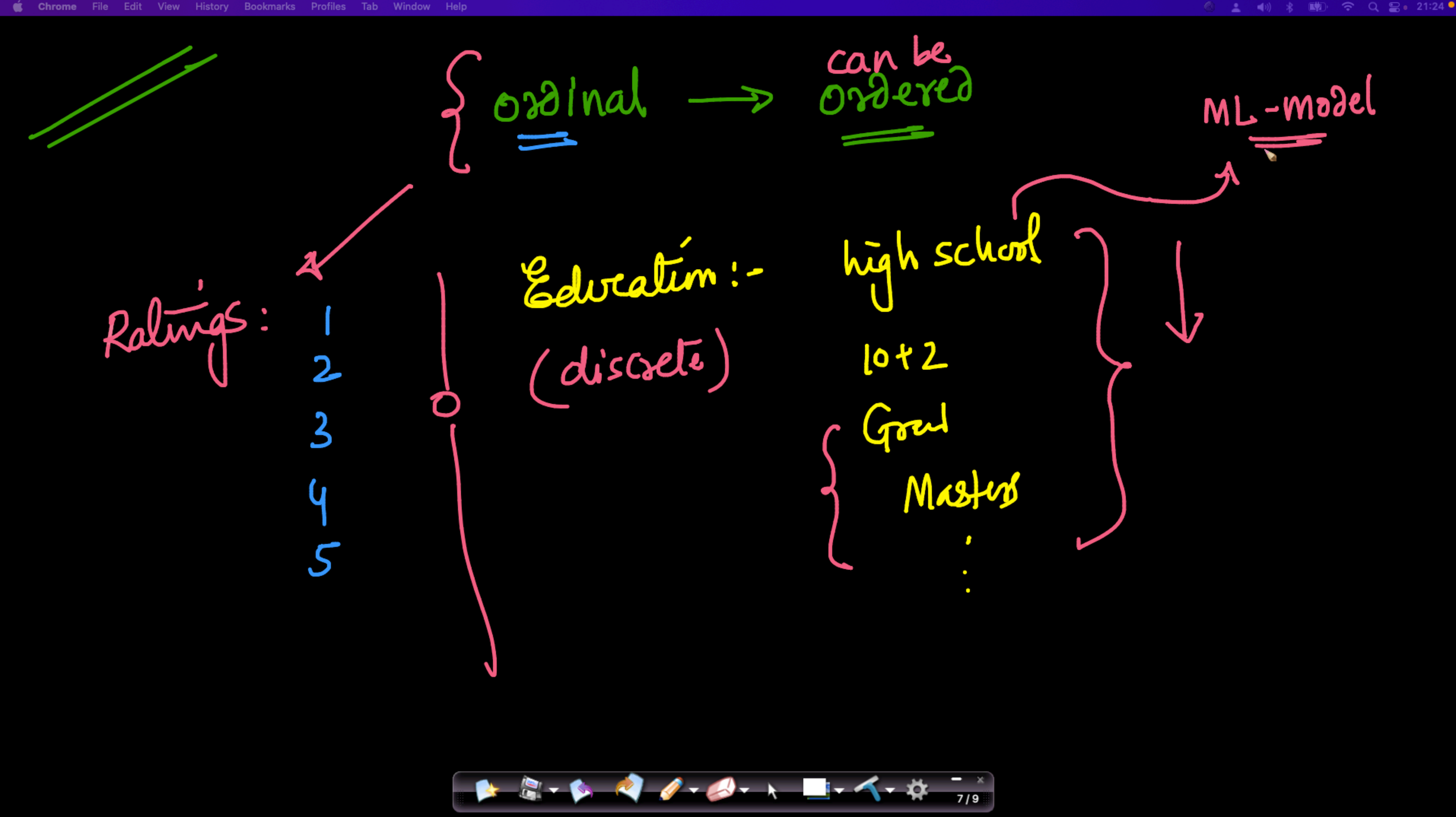
mode: most frequently occuring value

```
[ ] #variance
    print("Gold:",df['MntGoldProds'].std())
    print("Fruits:",df['MntFruits'].std())
    print("Sweets:",df['MntSweetProducts'].std())
    print("Wine:",df['MntWines'].std())
    print("Meat:",df['MntMeatProducts'].std())
    print("Fish:",df['MntFishProducts'].std())
```

```
Gold: 52.167438914997064
Fruits: 39.77343376457871
Sweets: 41.2804984878548
Wine: 336.5973926053717
Meat: 225.71537251175445
Fish: 54.62897940287769
```

```
[10] from scipy import stats
     print(stats.median_absolute_deviation(df['MntGoldProds']))
```


Income: continous r.v
mode: numerical
50,000$
$ 49-51K (Most)
probability dist
(peaks)
(later)


ordinal → can be ordered
ML-model
Ratings: 1 2 3 4 5
Education :- (discrete)
high school
10+2
Grad
Masters

Countries :- not ordinal

↳ categorial/discrete

mean → average

median → middle value

120

mode → continous & discrete

popular shoe-sizes

6, 7, 8, 8.5, ...

mean ✓

average money spent on wines by Cust-1 with more importance/weightage to more recent month purchases

{ weighted-mean

$$\bar{x}_w = \frac{w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_{12} x_{12}}{w_1 + w_2 + \cdots + w_{12}}$$

10.100 8.50 150.6

10 8 ... 1

problem:

1 2 3 4 5+ 6 ...

10 8 6 3 1 1 ...

weights

$x_1 = 100$; $x_2 = 50$;

$x_3 = 150$; $x_4 = 200$

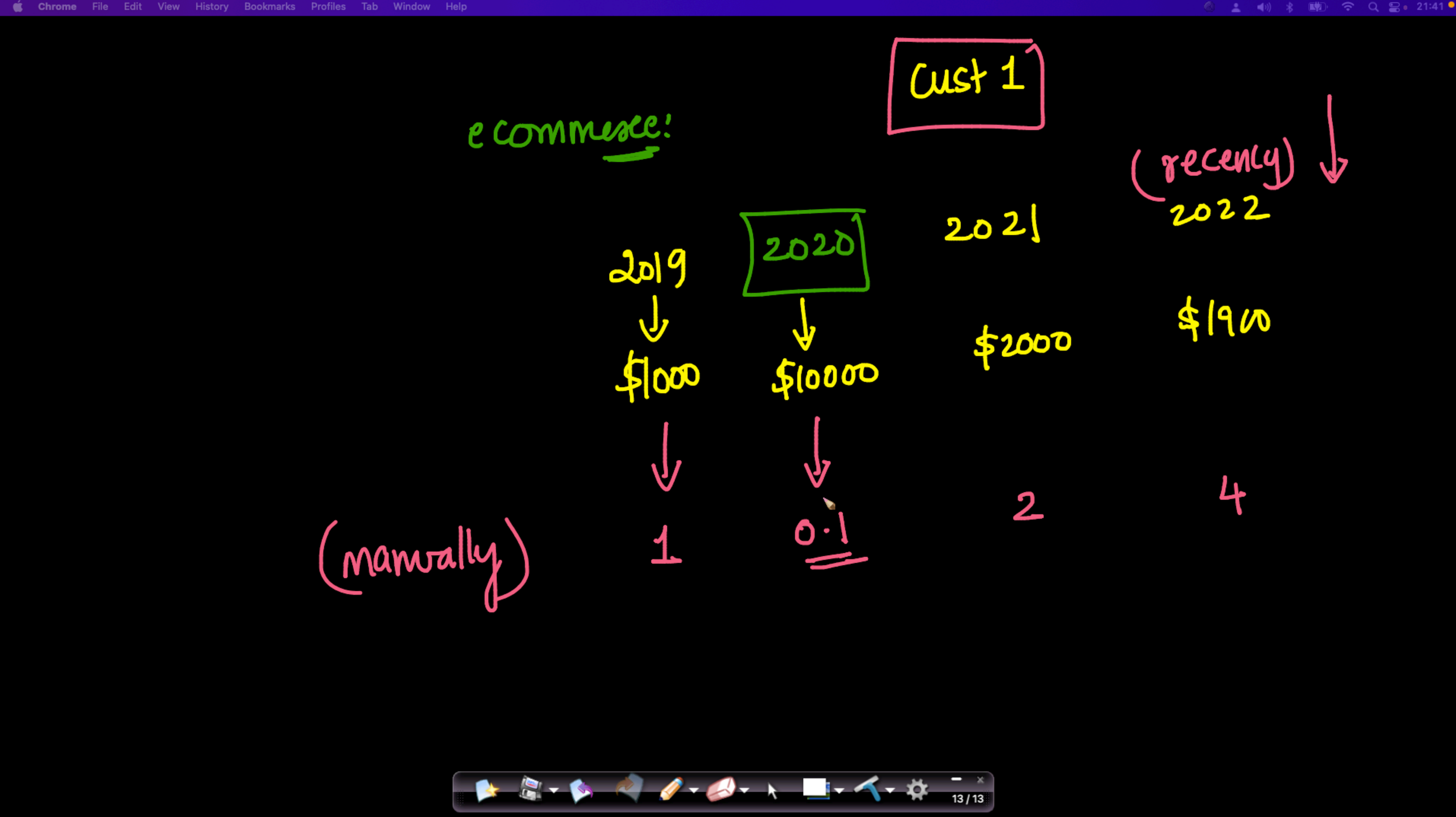
$x_5 = 100$; $x_6 = 200$

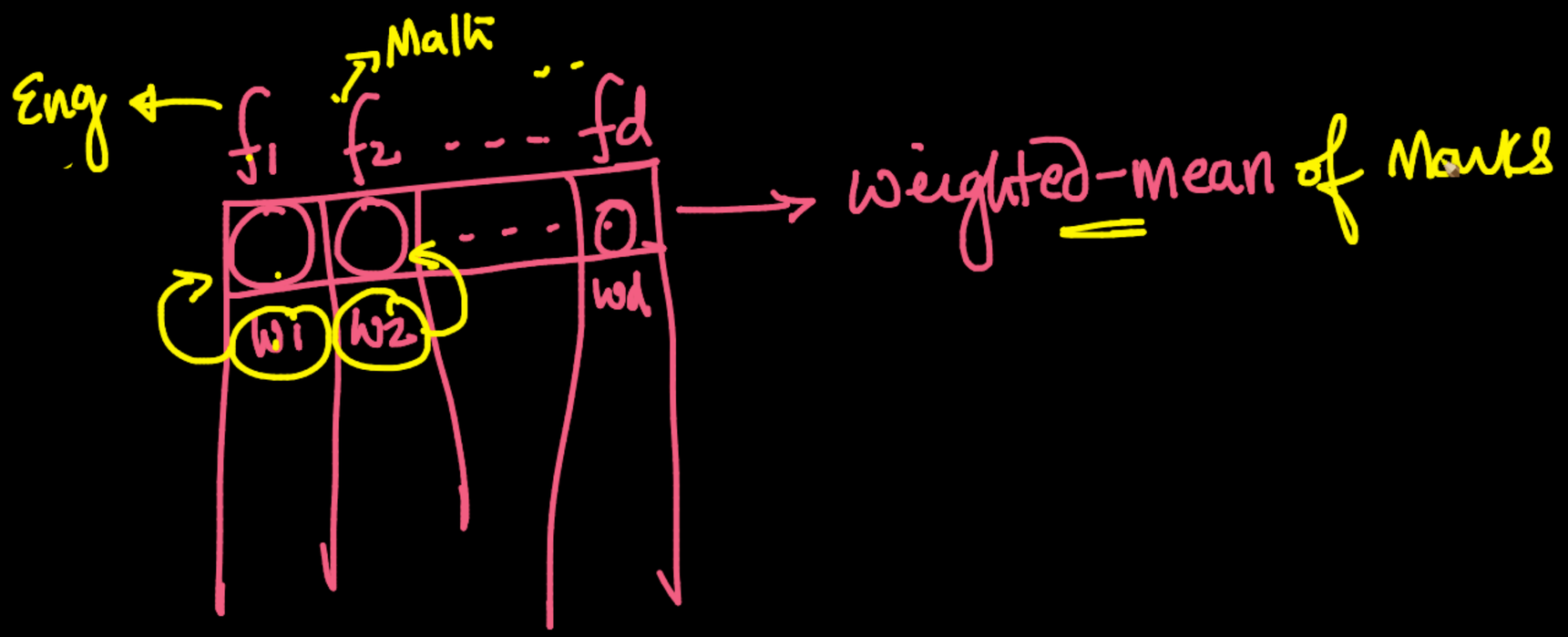
$$\bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

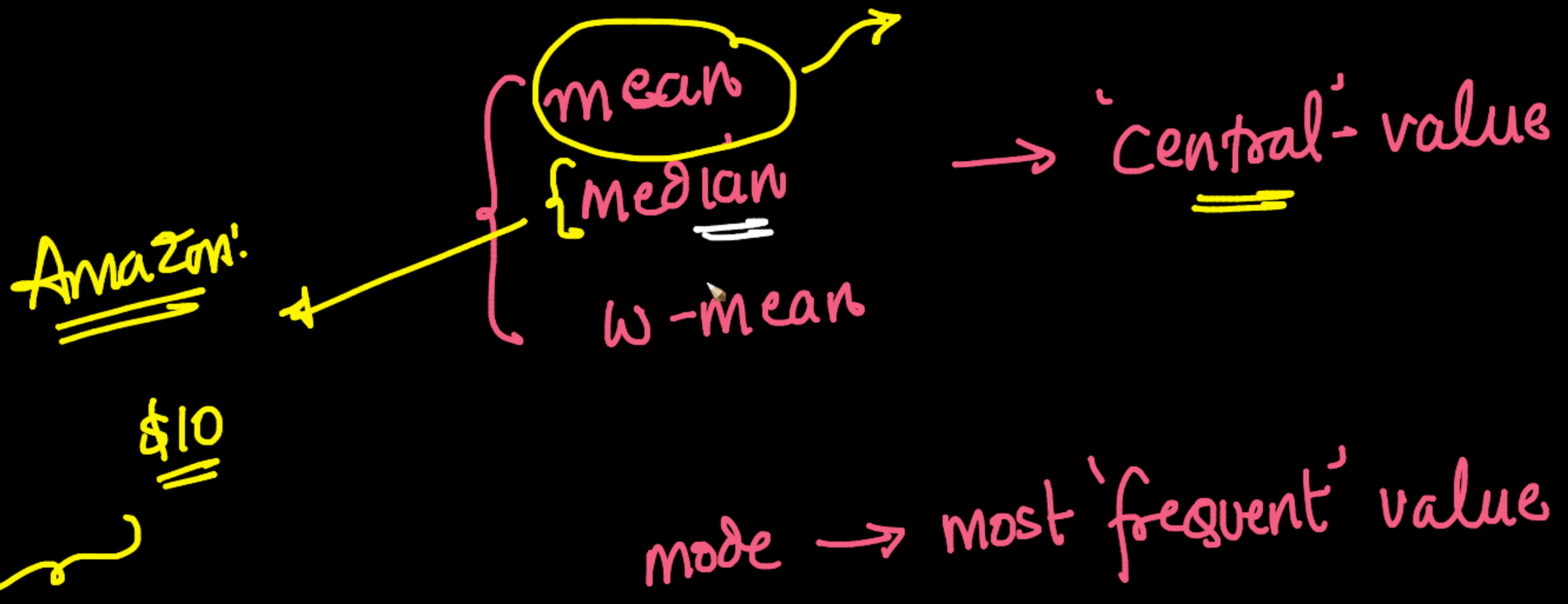
$n =$ 1000 Customers

Q) weighted mean with higher weightage to people living in France than Germany than Spain ...

$$\bar{x}_w = \frac{\sum_{i=1}^{1000} w_i x_i}{\sum_{i=1}^{1000} w_i}$$


Chrome File Edit View History Bookmarks Profiles Tab Window Help
21:41
ecommerce!
Cust 1
(recency)
2019
2020
2021
2022
$1000
$10000
$2000
$1900
(manually)
1
0.1
2
4
13 / 13


Eng
Math
f1
f2
fd
w1
w2
wd
weighted-mean of Marks


mean
median
w-mean
Amazon:
$10
→ 'central' value
mode → most 'frequent' value

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

$\bar{x}$: mean

→ $\text{Variance} = \dfrac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$

Unbiased?

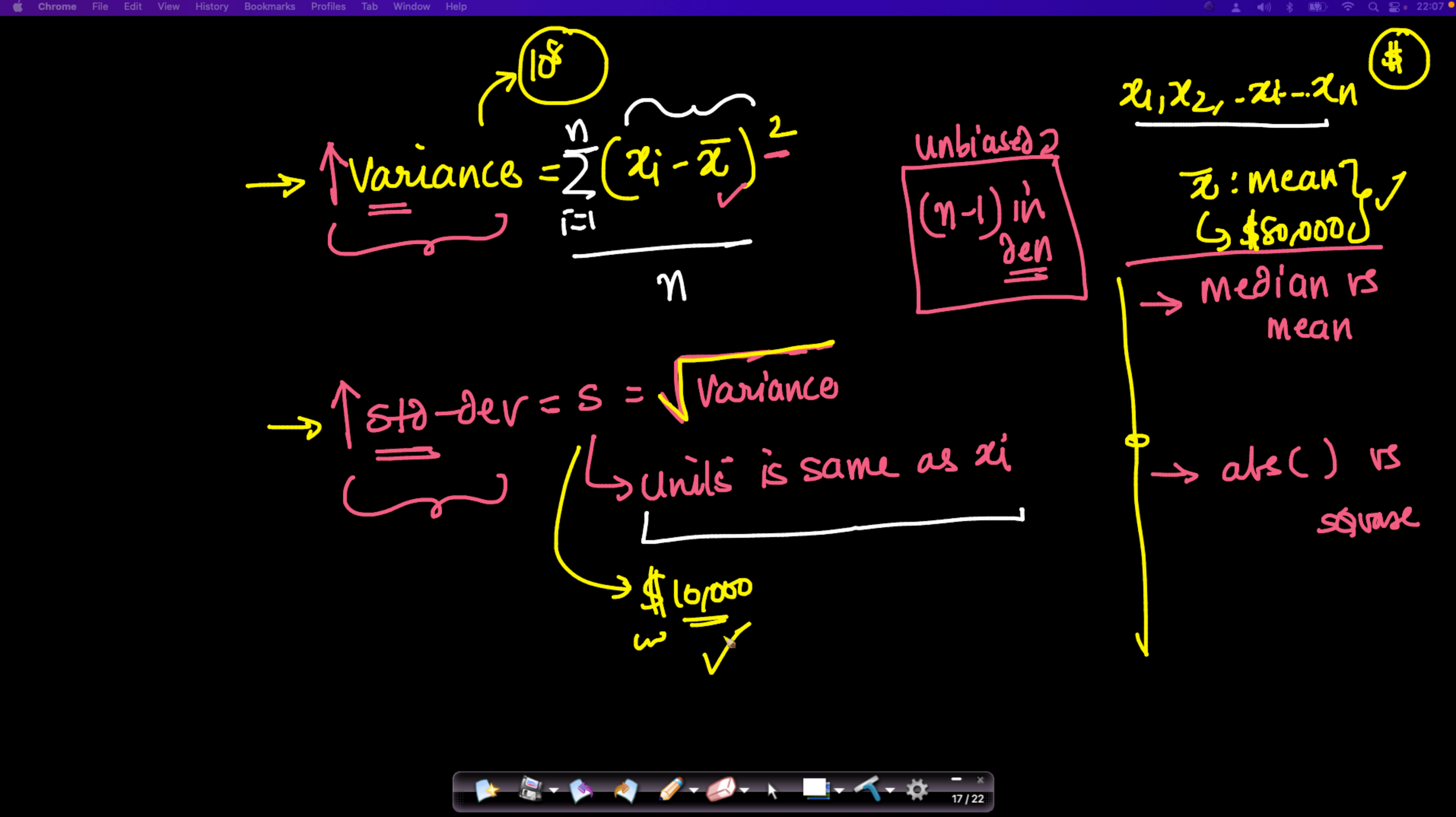
$(n-1)$ in den

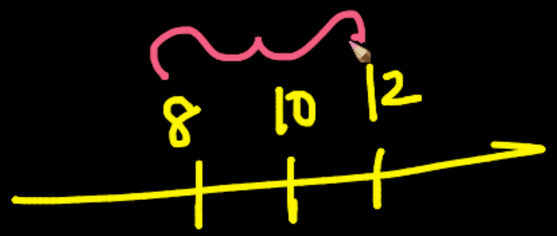
→ $\text{std-dev} = s = \sqrt{\text{Variance}}$

↳ units is same as x_i

→ median vs mean

→ abs( ) vs square

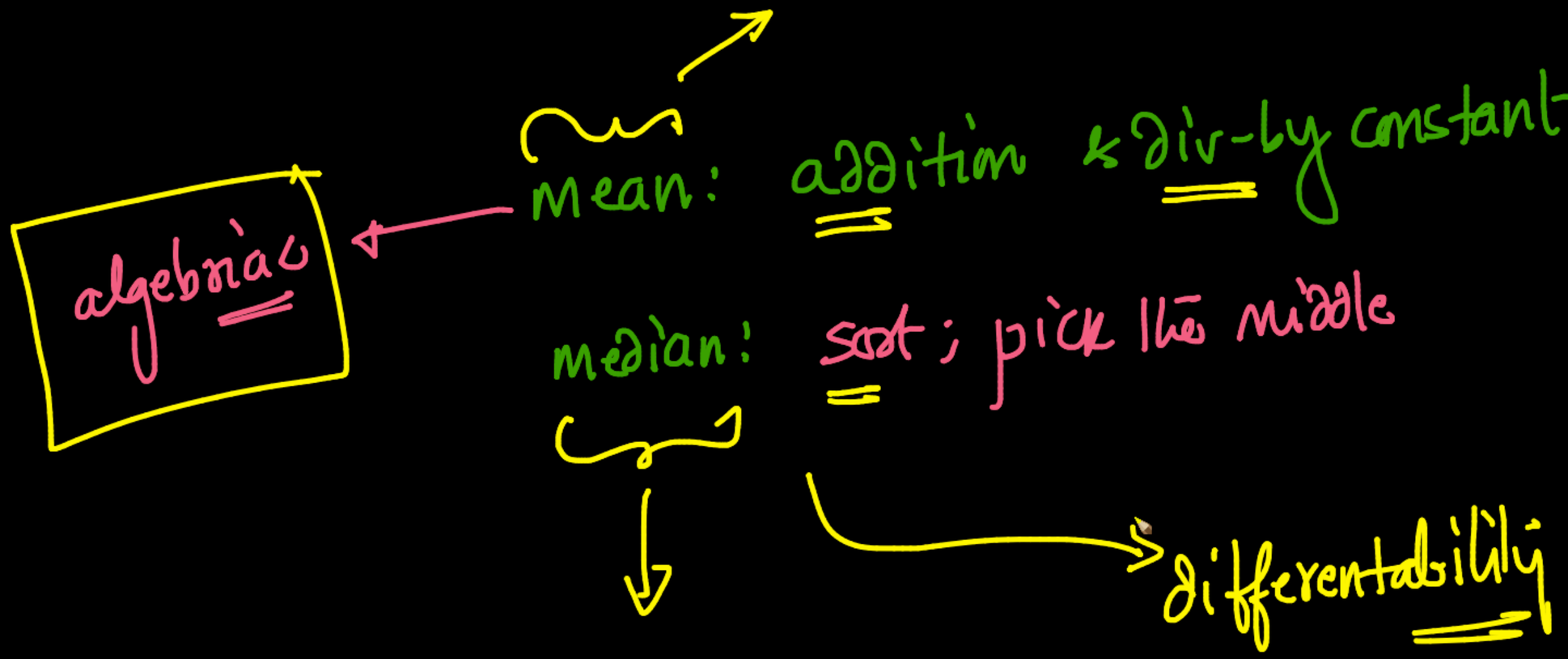

Chrome File Edit View History Bookmarks Profiles Tab Window Help
18$
Variance = $\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$
Unbiased?
(n-1) in den
$x_1, x_2, \ldots x_i \ldots x_n$
$
$\bar{x}$: mean
↳ $80,000
median vs mean
abs( ) vs square
std-dev = s = $\sqrt{\text{Variance}}$
↳ units is same as x_i
$10,000
17 / 22



$$\sum_{i=1}^{n} (x_i - \bar{x}) \rightarrow \frac{-2+0+2}{3}$$

$$\bar{n}$$

$$= \frac{0}{3} = 0$$


algebraic
mean: addition & div-by constant
median: sort; pick the middle
differentiability

Median abs deviation around the median (MAD) = median (abs $|x_i - x_M|$)

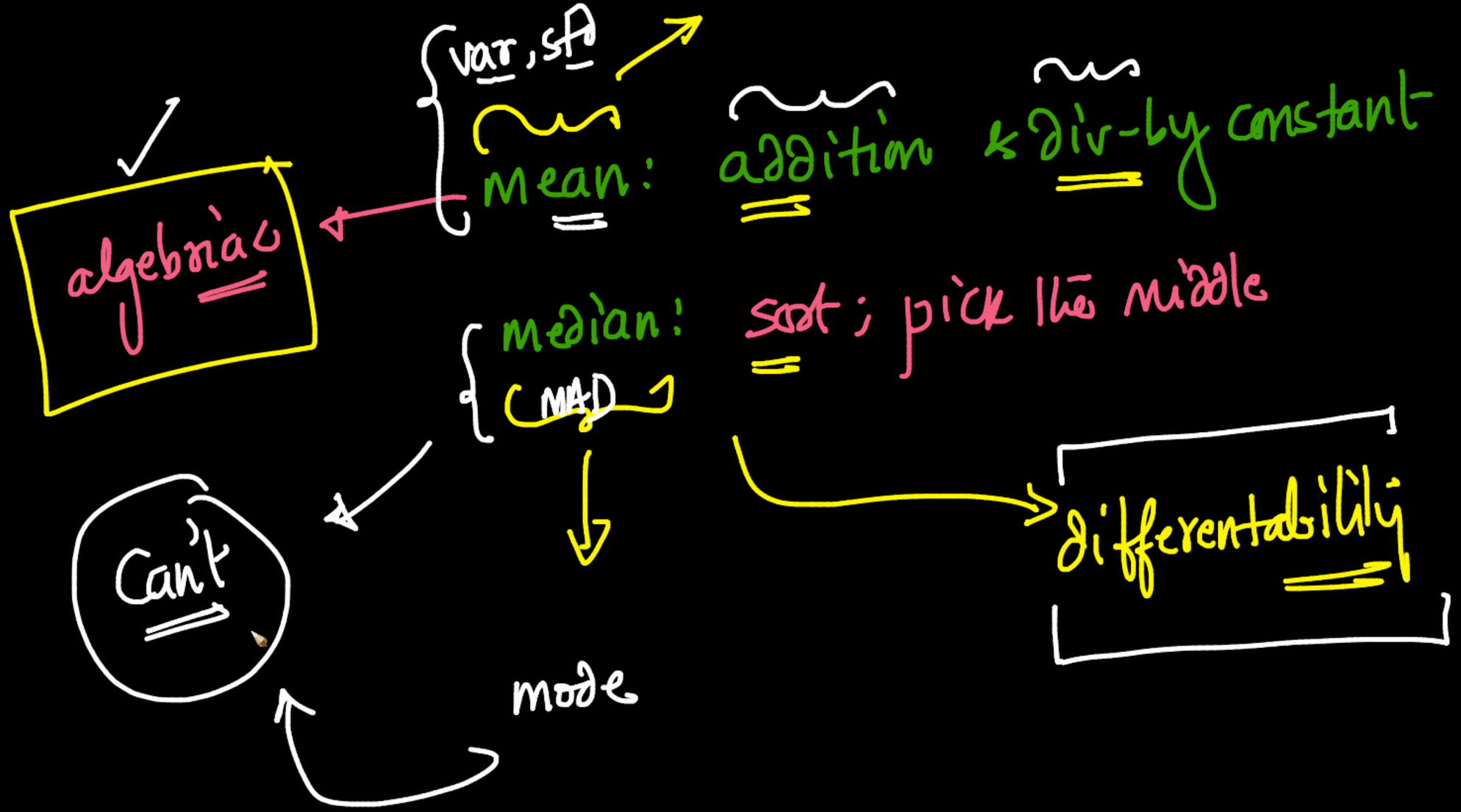
$x_1, x_2 \; - \; - \; - \; x_n$: x_M = median (x_i's)

very very outlier proof

Median abs-dev around the mean

↓

very seldom used


var, sd
mean:
addition & div-by constant
algebraic
median:
sort; pick the middle
MAD
differentiability
Can't
mode

MAD vs std

MAD → robust estimator

std → impacted by outliers

Q

When does weighted mean behave like the mean?

$$\bar{x}_w = \frac{1\cdot w_1 x_1 + 1\cdot w_2 x_2 + \cdots + 1\cdot w_n x_n}{w_1 + w_2 + \cdots + w_n} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \bar{x}$$

$w_1 + w_2 + \cdots + w_n = n$

Percentiles:

$x_1, x_2, \ldots x_n$

one-def

k^{th} percentile $= P_k$

"atleast" k percentage of points $\leq P_k$

e.g: 3, 2, 1, 5, 6 → sort → 1, 2, 3, 5, 6 $\quad n = 5$

$P_{50} = 3$ $\quad 3/5 = 60\%$. A points ≤ 3

$n = 5$

$2, 3, 1, 6, 5$ ⟶ find P_{10} ?

sort ↓

1 2 3 4 5

1, 2, 3, 5, 6

$p = 10$

≈ 20% of values are ≤ 1

$P_{10} = 1$ ✓

$\left\lceil \frac{n \times p}{100} \right\rceil = \left\lceil \frac{50}{100} \right\rceil = \lceil 0.5 \rceil = 1$

Q

n-values ; p^{th} - percentile value

- sort them

[index: 1 2 3 - too-- n]

[index of the p^{th} percentile = in the sorted array. $\lceil \frac{n \times p}{100} \rceil$ ✓

$\frac{20}{100} \times n$

[zsum(0.1) = 0

$\lceil 0.1 \rceil = 1$

median = P_{50}

Quantiles:

Q_0: P_0 → min

$Q_1 = P_{25}$

$Q_2 = P_{50}$ → median

$Q_3 = P_{75}$

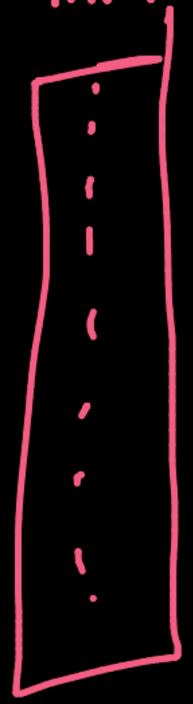
$Q_4 = P_{100}$ → max

$x_1, x_2 \dots x_n$ → sort

$\lceil \frac{25 \times n}{100} \rceil = Q_1$

$\lceil \frac{75 \times n}{100} \rceil = Q_3$

Mnt Wires



Uni-variate

Variable

$P_0 - P_{50}$

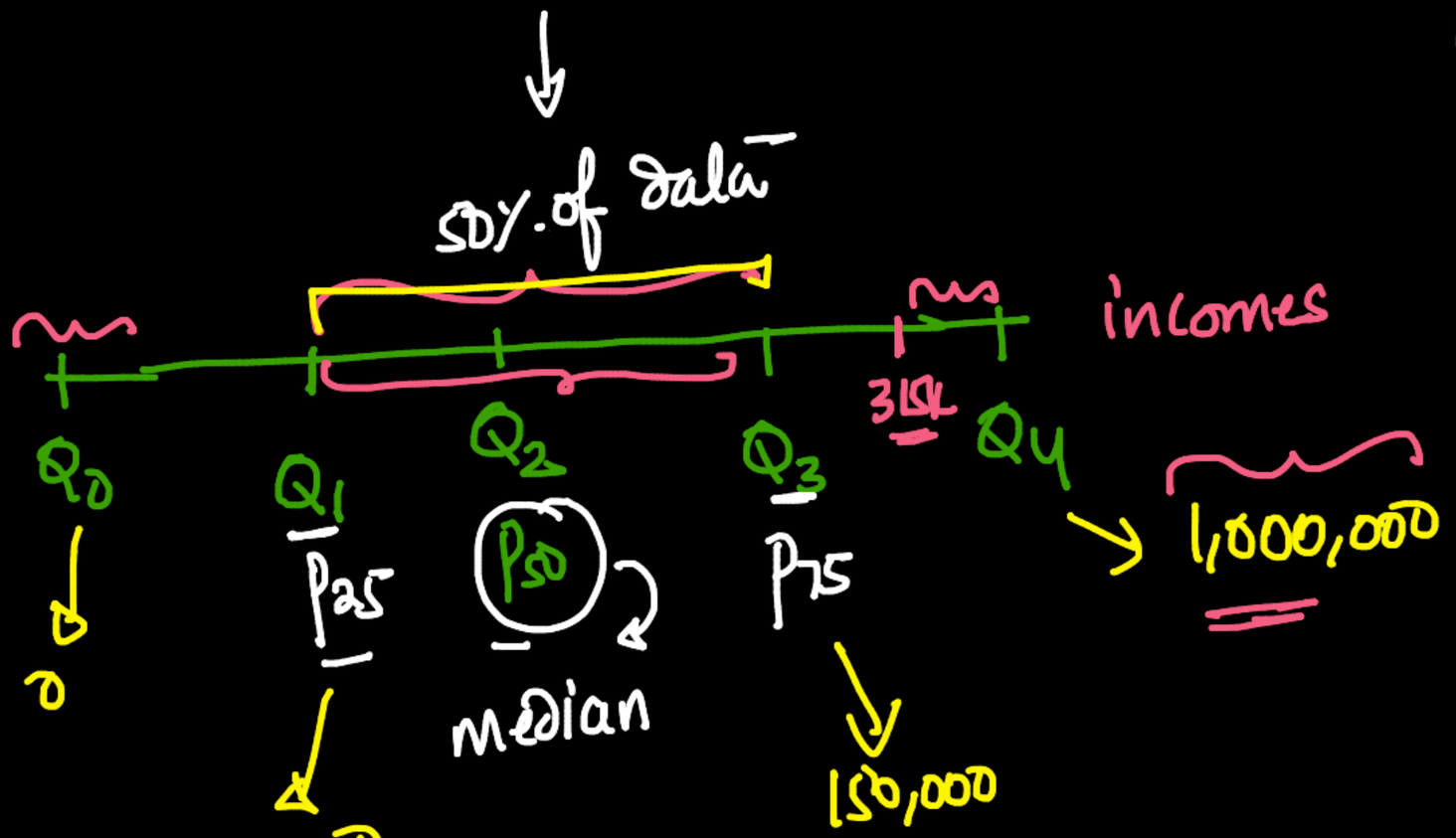
$P_{50} - P_{100}$

IQR & outliers

↳ $Q_3 - Q_1$

150,000 − 40,000

= 110,050 $



I Rule-of-thumb:

detect Outliers

- subjective
- domain specific

$Q_1 - 1.5 \text{IQR} = 40{,}000 - 1.5 \times 110{,}000 = \text{-ve}$

(or)

$Q_3 + 1.5 \text{IQR} = 150{,}000 + 1.5 \times 110{,}000$

$= 150\text{K} + 165\text{K} = 315{,}000$

Probability2.ipynb - Colaborat | ProbabilityDisb_1.ipynb - Cola | Percentile - Wikipedia

colab.research.google.com/drive/1l5T7TVIAASw9TdI4JxqJxuDqRgFBGOW3#scrollTo=GHq8XaUuftQE

+ Code + Text

RAM Disk

```
#outliers using IQR range
r = 1.5*stats.iqr(df['MntWines'])
lb = np.percentile(df['MntWines'], 25)-r
ub = np.percentile(df['MntWines'], 75)+r
print(lb)
print(ub)
```

```
-447.0
1225.0
```

lb

$Q_1 - 1.5IQR$

ub

$Q_3 + 1.5IQR$

Pin=Max

1493

1225

[14]
```
df['MntWines'].max()
```

```
1493
```

[15]
```
sum(df['MntWines'] > ub)
```

```
35
```

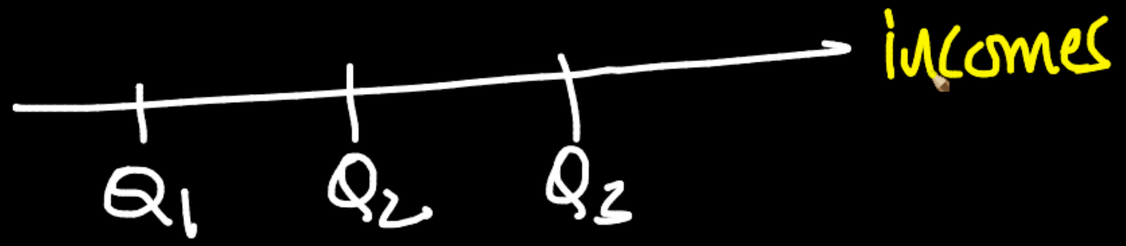
```
df['MntWines'].plot.hist()
#chooses nbins=10
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb570fc8d0>
```

33 / 33

$Q_3 - Q_2 \leq Q_2 - Q_1$

not always



Recap:

Uni-variate

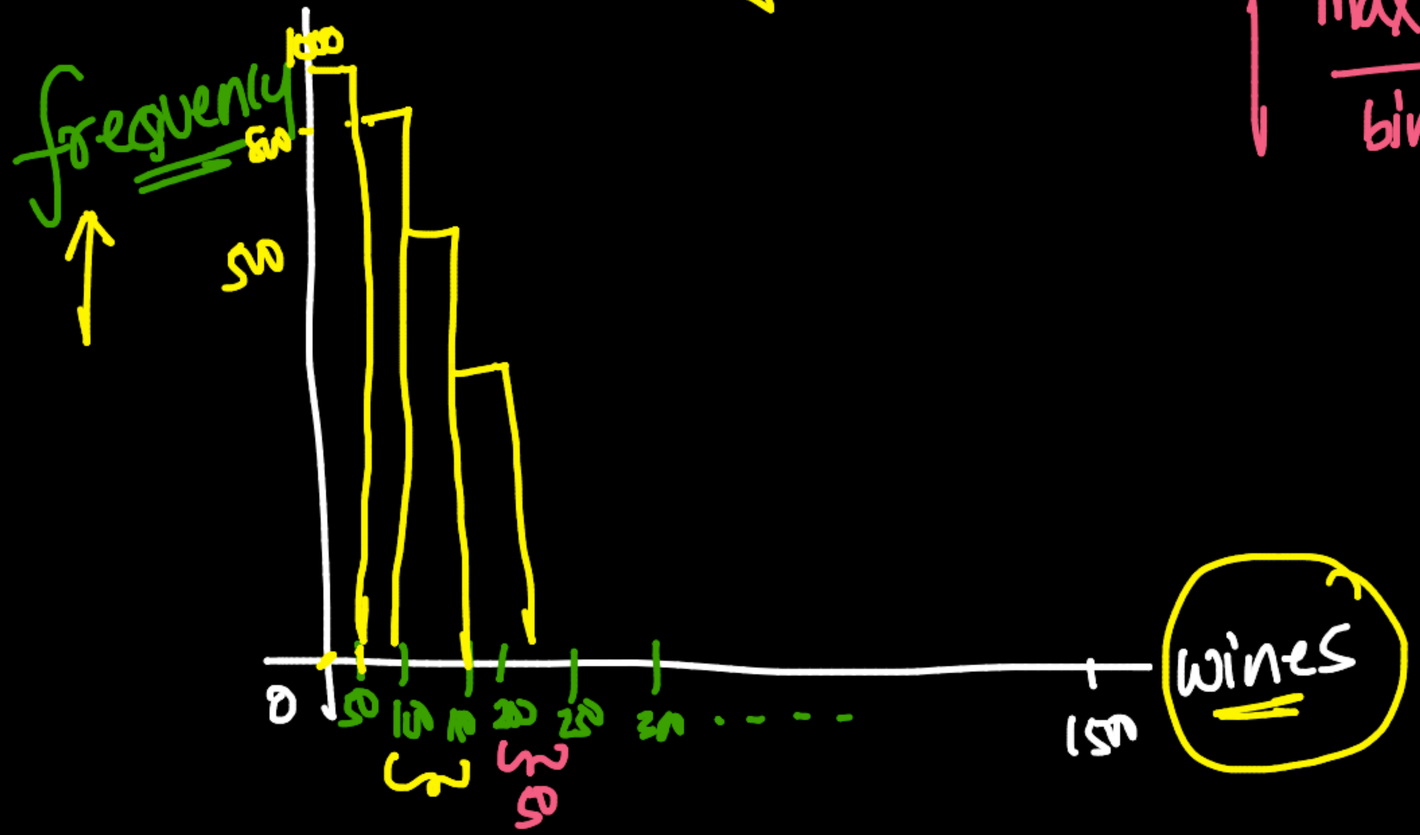
- mean
- median
- mode
- percentiles
- quantiles
- IQR

stdev
MAD

Visualize

Visualize data dist

histogram



$$\left[\frac{\text{Max} - \text{Min}}{\text{binsize}}\right] = \text{bins}$$

bins = 30 bins ✓

bin-size = 50

Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:03

Probability2.ipynb - Colaborat | ProbabilityDisb_1.ipynb - Cola | Percentile - Wikipedia

colab.research.google.com/drive/1l5T7TVIAASw9TdI4JxqJxuDqRgFBGOW3#scrollTo=_IfNrEJzl2EX

Update

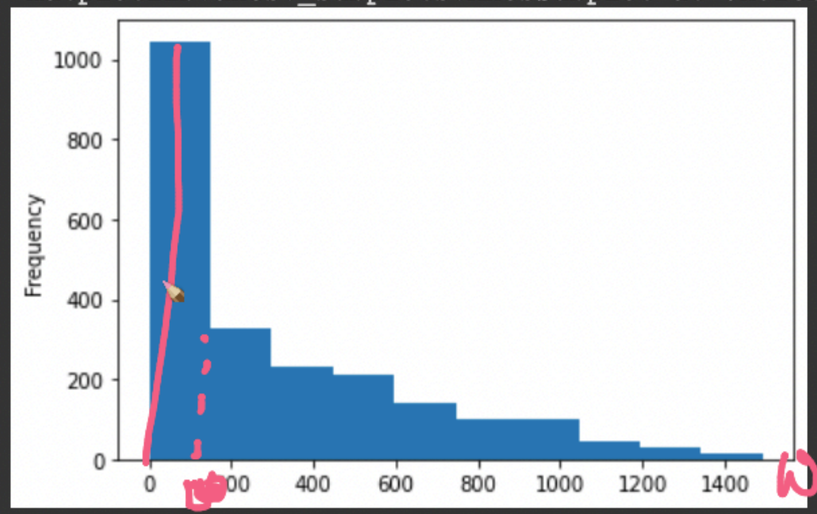
+ Code + Text

RAM Disk

pandas

```
[16] df['MntWines'].plot.hist()
     #chooses nbins=10
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb570fc8d0>
```

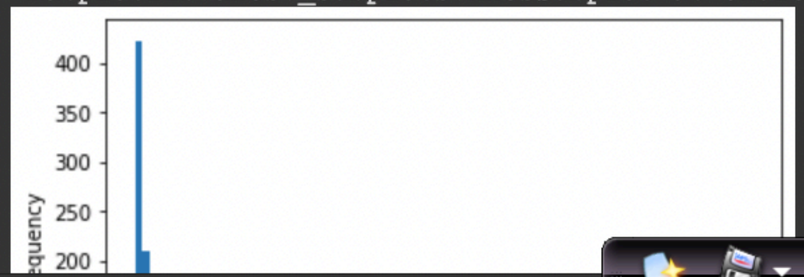


wines

```
df['MntWines'].plot.hist(bins=100)
```

1s

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```



37 / 37

Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:04

Probability2.ipynb - Colaborat | ProbabilityDisb_1.ipynb - Cola | Percentile - Wikipedia

colab.research.google.com/drive/1l5T7TVIAASw9TdI4JxqJxuDqRgFBGOW3#scrollTo=_IfNrEJzl2EX

Update

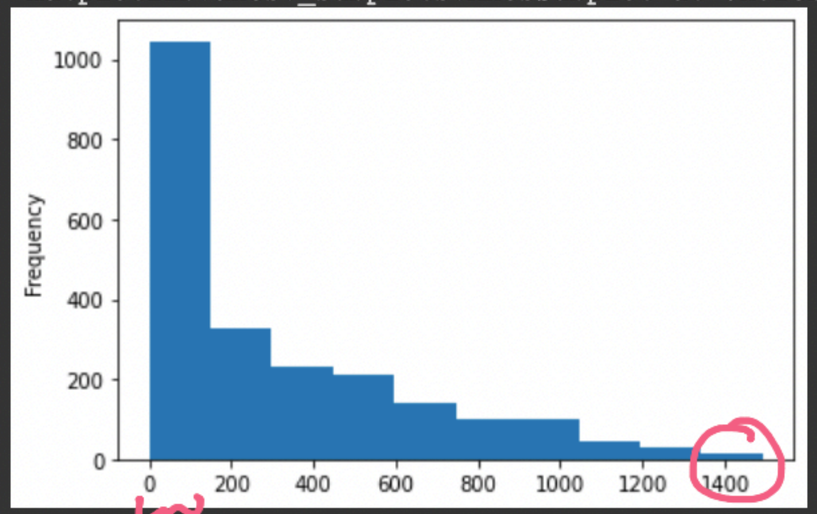
+ Code + Text

RAM Disk

```
[16] df['MntWines'].plot.hist()
     #chooses nbins=10
```

2

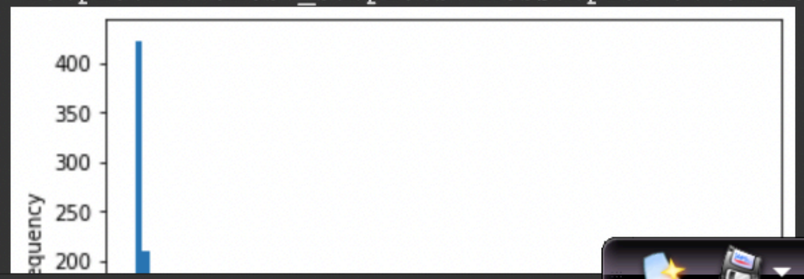
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb570fc8d0>
```




```
df['MntWines'].plot.hist(bins=100)
```

1s

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```



38 / 38

Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:06

Probability2.ipynb - Colaborat | ProbabilityDisb_1.ipynb - Cola | Percentile - Wikipedia

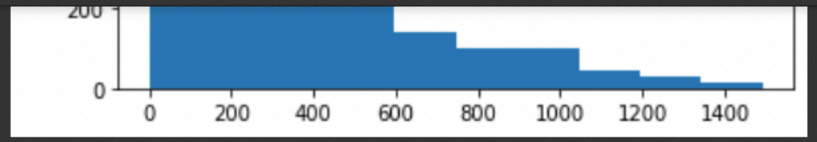
colab.research.google.com/drive/1l5T7TVIAASw9TdI4JxqJxuDqRgFBGOW3#scrollTo=_IfNrEJzl2EX

Update

+ Code + Text

RAM Disk

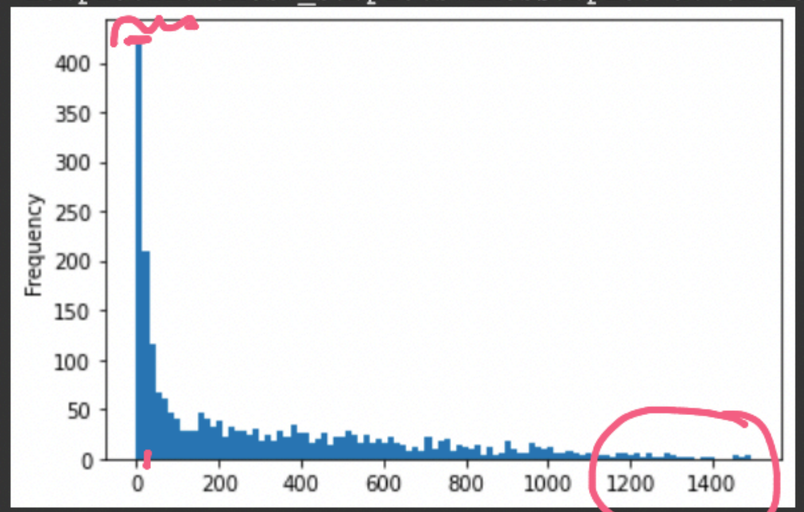
[16]




```
df['MntWines'].plot.hist(bins=100)
```

1s

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```



bins ↑ finer-graph

1450

[ ]

```
[22] import seaborn as sns
     sns.kdeplot(data=df['MntWines'])
```

0s

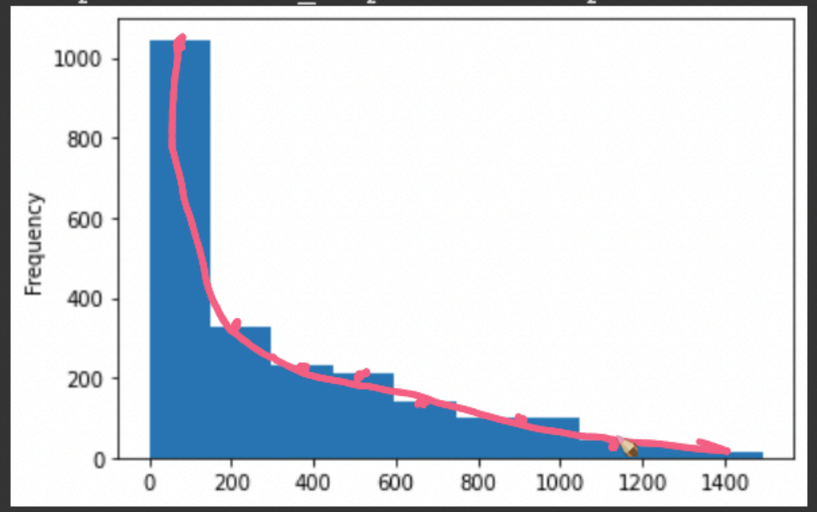
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb568a5150>
```

39 / 40

Smoothed histogram

[16]

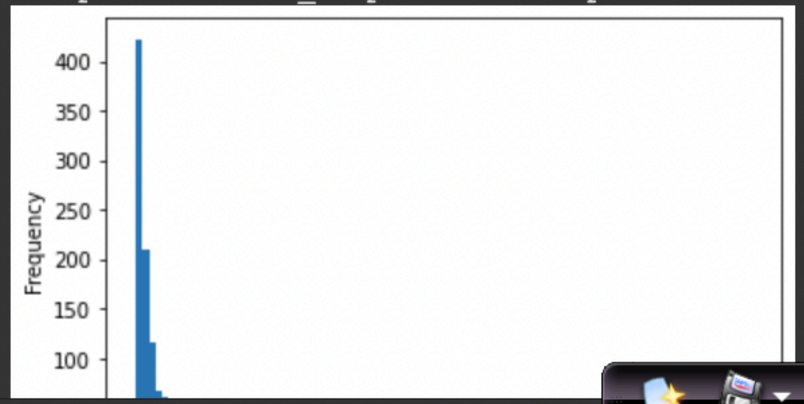
```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb570fc8d0>
```



1s

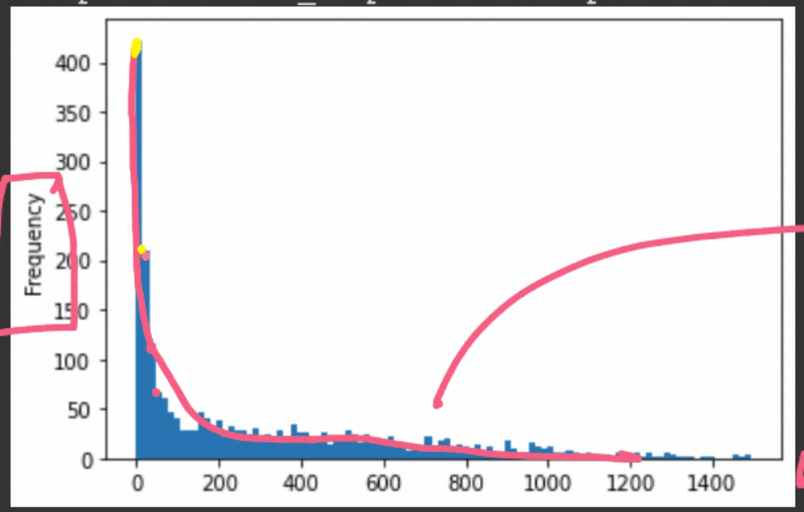
```
df['MntWines'].plot.hist(bins=100)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```



```
df['MntWines'].plot.hist(bins=100)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```



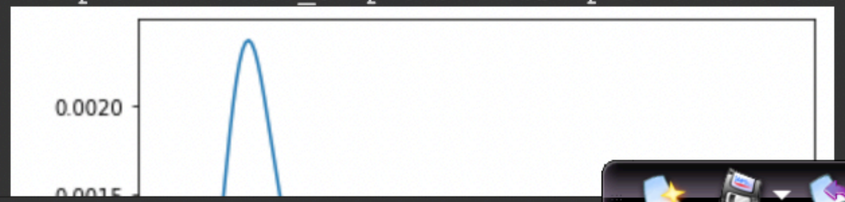
Smoothed histogram

wines

```
[ ]
```

```
[22] import seaborn as sns
     sns.kdeplot(data=df['MntWines'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb568a5150>
```



Kernel density plots

KDE

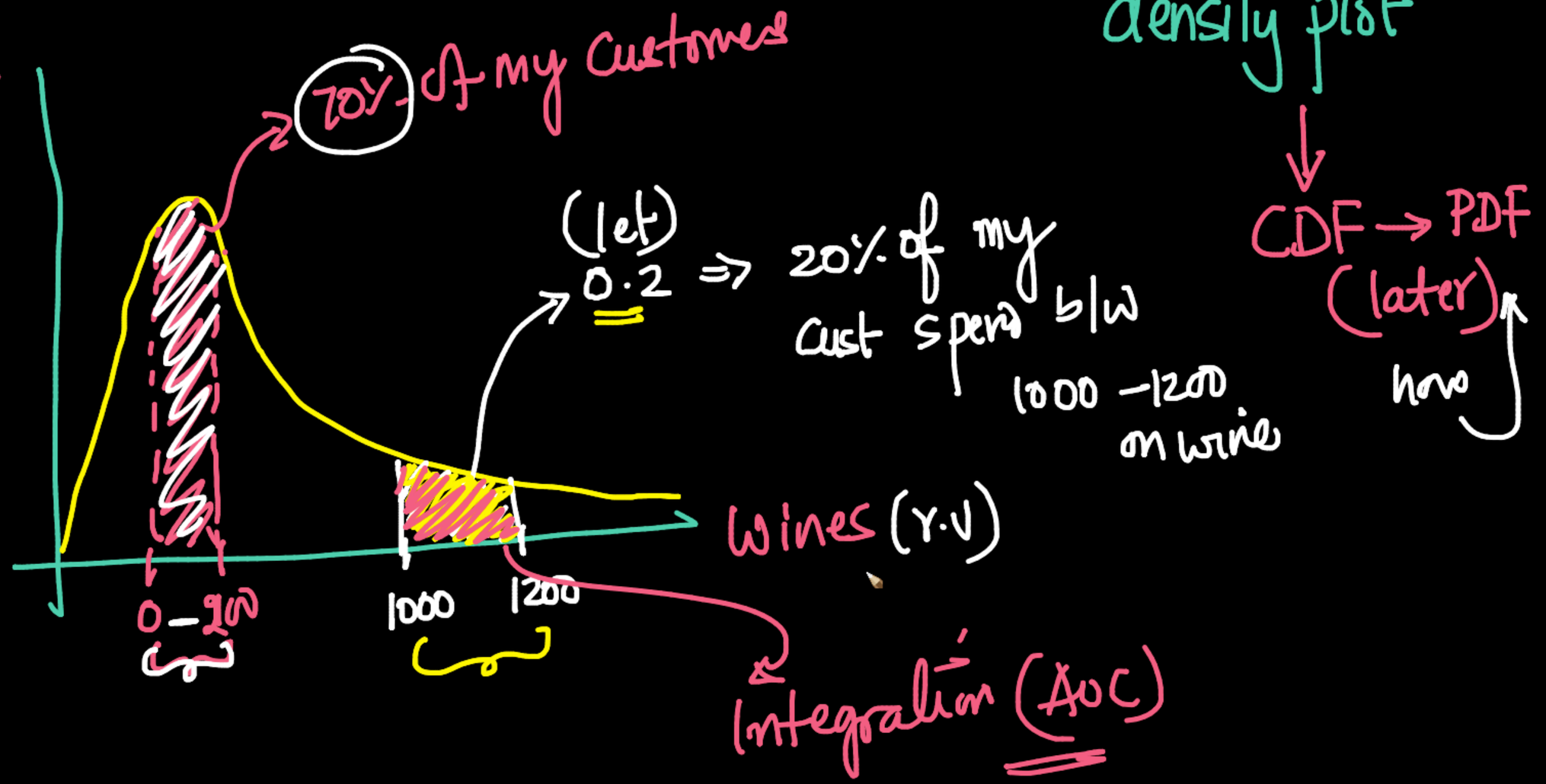
Later

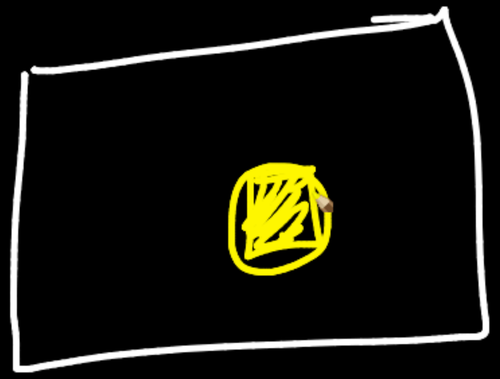
alternative form of smoothing

- Kernel
- Gaussian Kernel

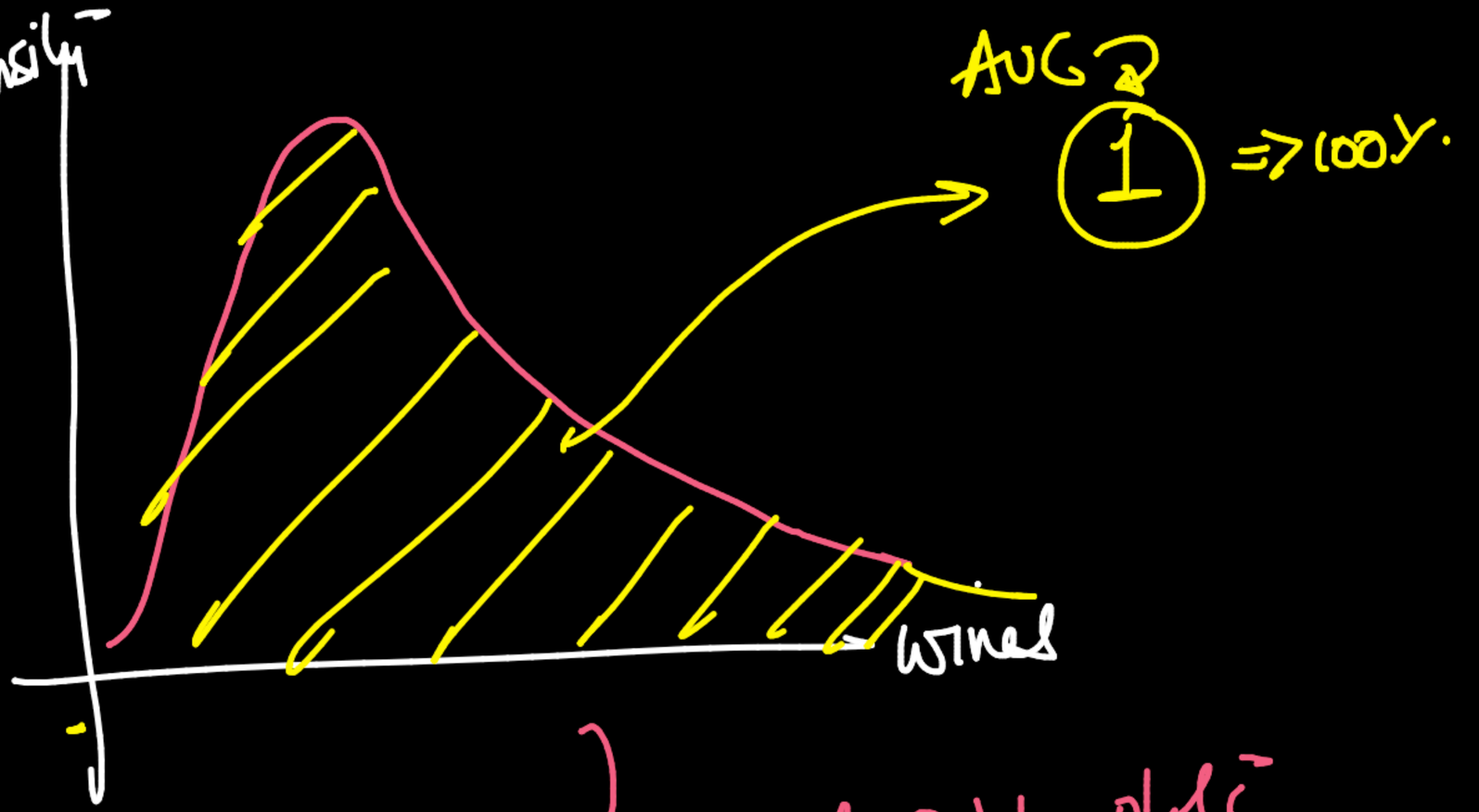
density plot

CDF → PDF (later) how

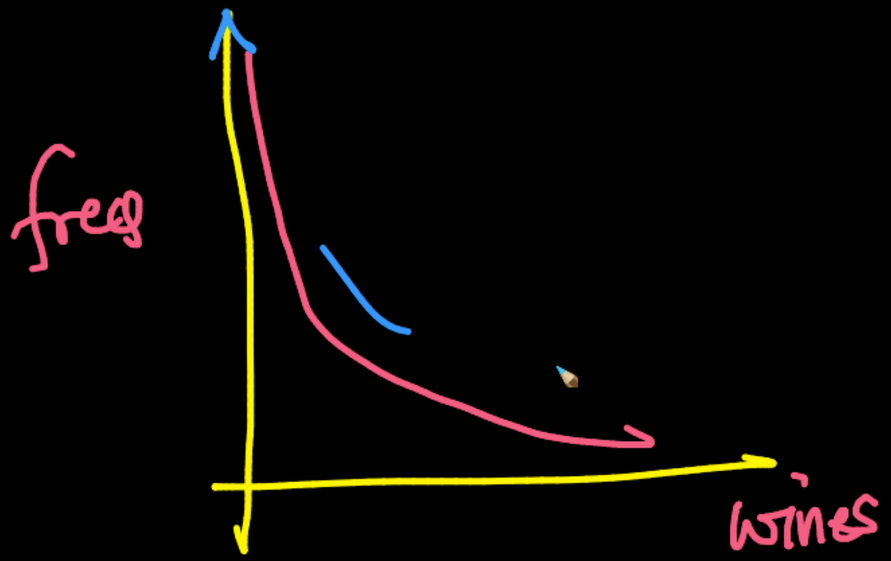


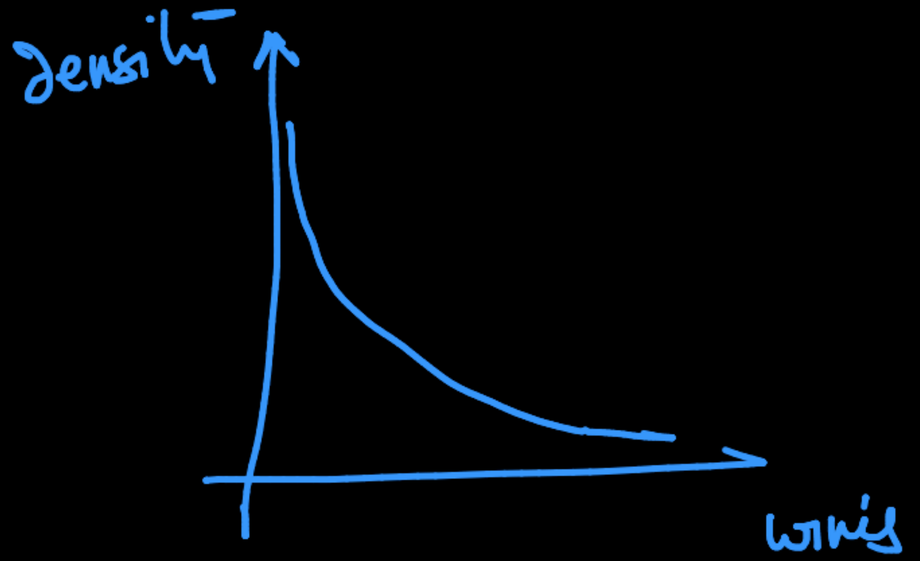


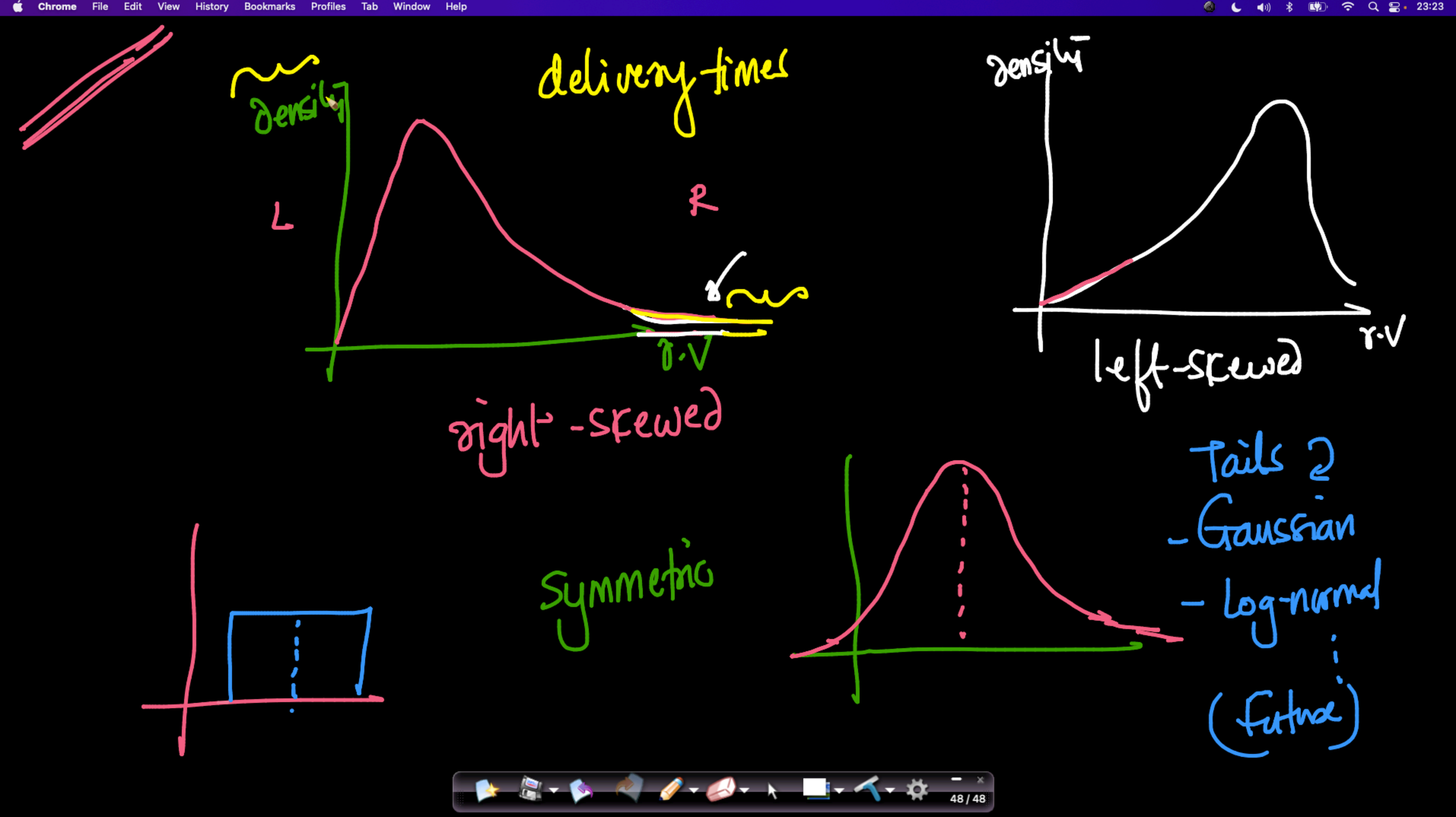
Q

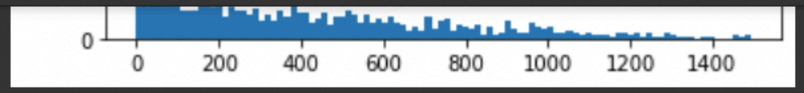


↳ proof. dist. plots (later)





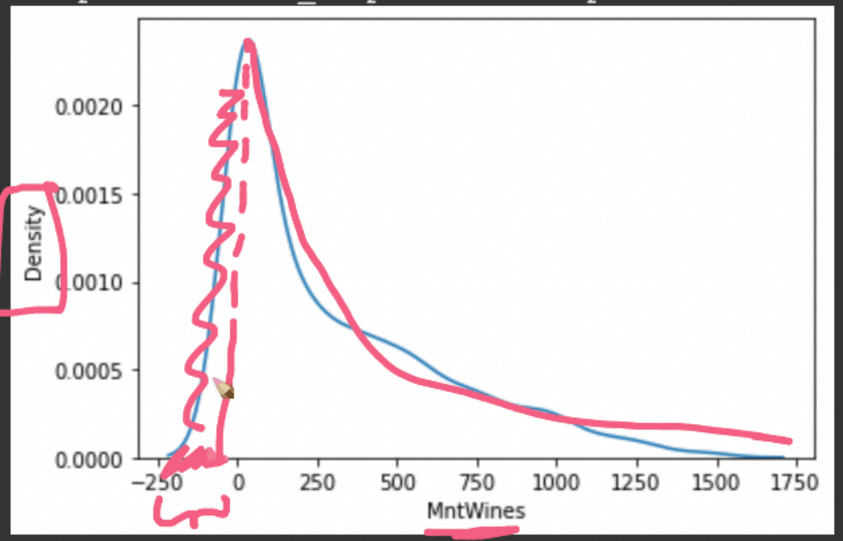

delivery times
density
L
R
r.v
right-skewed
density
r.v
left-skewed
Symmetric
Tails 2
- Gaussian
- log-normal
(future)



[ ]

```
[22] import seaborn as sns
     sns.kdeplot(data=df['MntWines'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb568a5150>
```

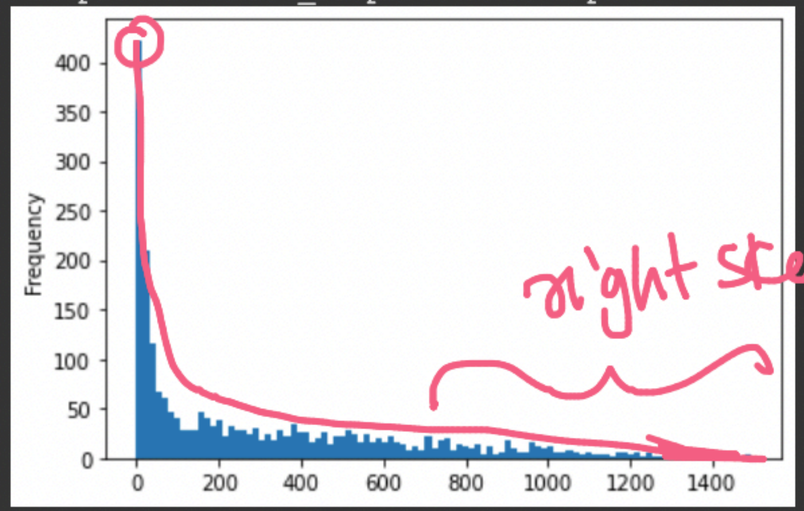



```
[24] df[' Income ']
```

```
0       $84,835.00
1       $57,091.00
2       $67,267.00
3       $32,474.00
```

1s

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb56ffbf50>
```

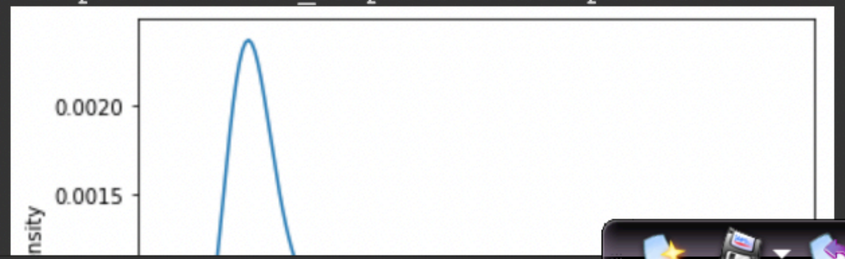


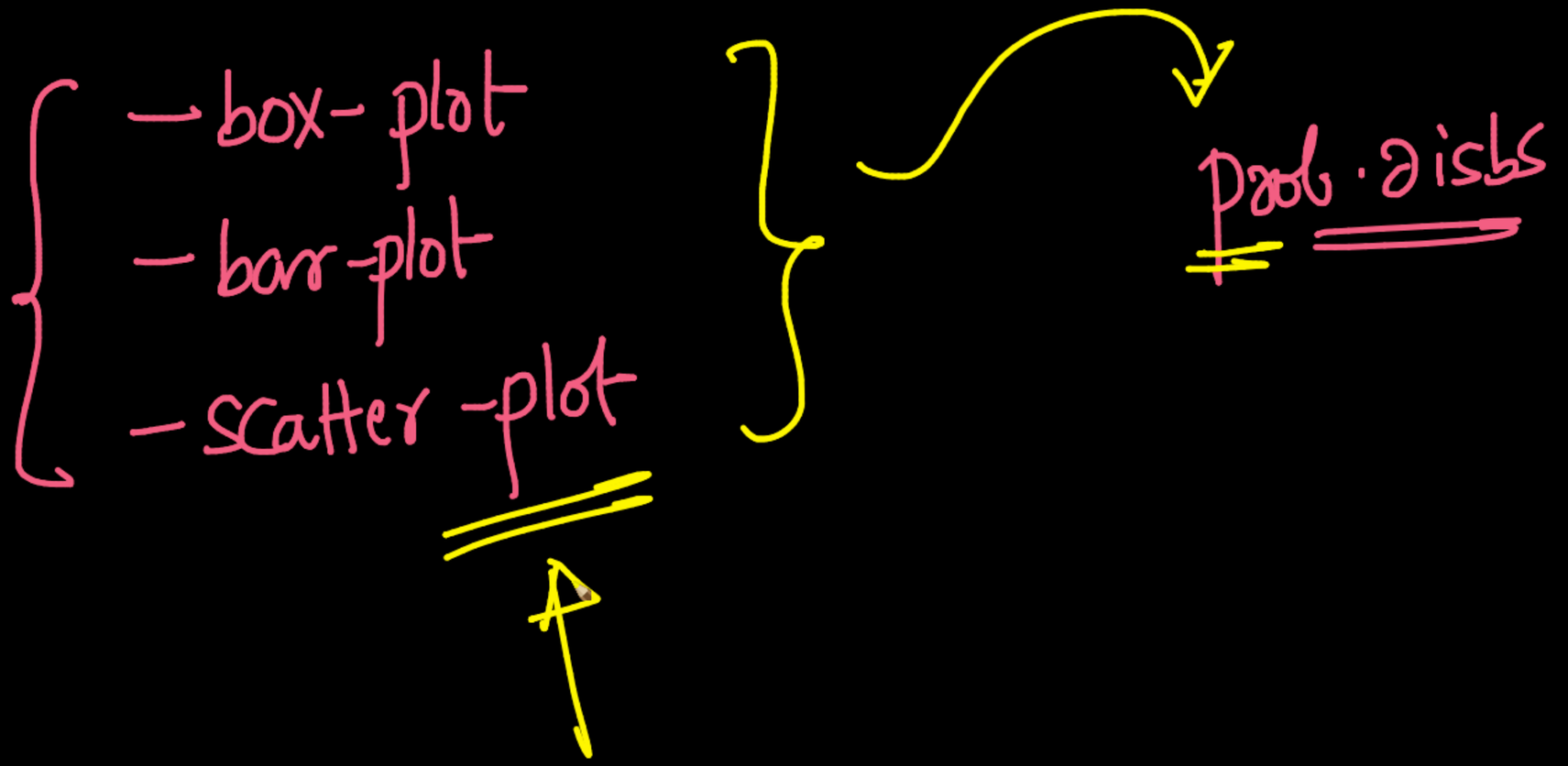

```
[ ]
```

0s

```
import seaborn as sns
sns.kdeplot(data=df['MntWines'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb568a5150>
```





Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:44

Probability2.ipynb -

InterviewBit Software Services | Live | DSML Advanced : Pr

scaler.com/meetings/i/dsml-advanced-probability-distributions-1-3/live

Update

DSML Advanced : Probability Distributions 1 | Lecture

GEOMRTT

$$\bar{x}_w = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}$$

You are sharing your screen now

Stop Sharing

w_i's are equal to

Srikanth Varma Chekuri (You) (Screen)

02:44:06

Srikanth Varma Chekuri (You)

People 27

Chat

Questions 3

Chat

Notify me about Nothing

New a message

Avijit Swain

Hi. Did you get the chance to look at the question that I asked in the last class. Independent events are not mutually exclusive and vice versa.

Already Answered | Answer Now

11:43 pm 0

Harpreet Singh To: Everyone 11:44 pm

yes sir

Yes No

To: Everyone Enable/Disable Chat

Type message

Doubt Session Ongoing

52 / 52

than seven—and
noothed

est when

Scott's normal reference rule [edit]

Bin width h is given by

$$h = \frac{3.49\hat{\sigma}}{\sqrt[3]{n}},$$

mutually excl $\not\Rightarrow$ independent

$P(A) \neq 0$
$P(B) \neq 0$ } ✓

$$\text{Var} = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2$$

Chrome File Edit View History Bookmarks Profiles Tab Window Help 23:48

Probability2.ipynb -

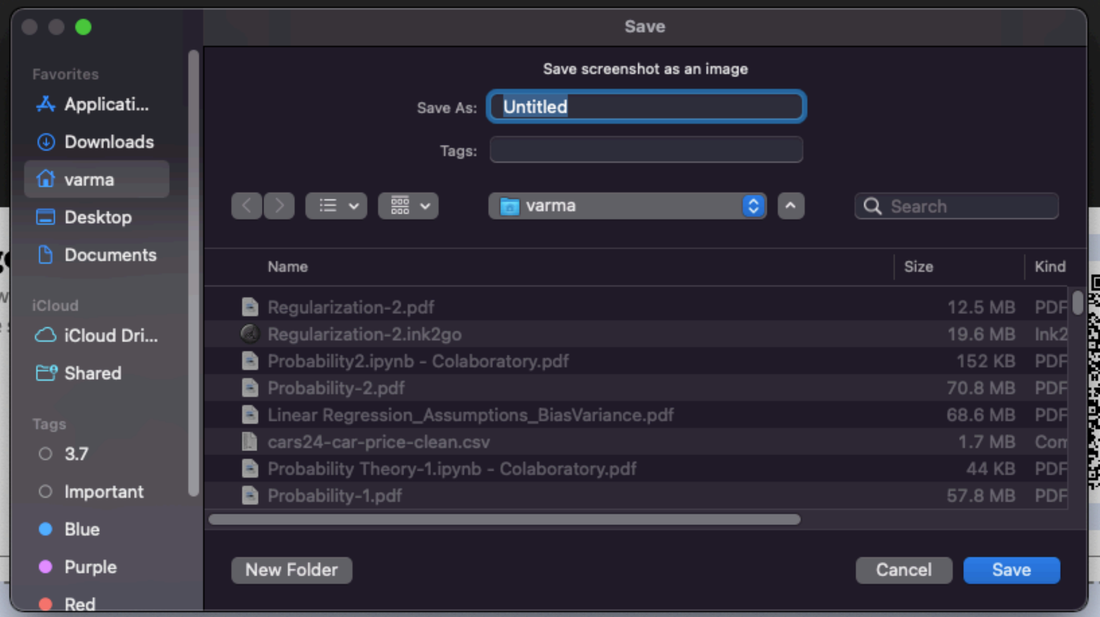
InterviewBit Software Services | Live | DSML Advanced : Probat

en.wikipedia.org

scaler.com/meetings/i/dsml-advanced-probability-distributions-1-3/live

Update

Update



We g

Notes w

simple

1

2

Drag and drop files or click here to upload

Files Uploaded from your computer appear here

than seven—and

noothed

est when

Scott's normal reference rule [edit]

Bin width h is given by

$$h = \frac{3.49\hat{\sigma}}{\sqrt[3]{n}},$$

55 / 55