Welcome to INFO 2950 (Intro to Data Science)!

Pick up 1 whiteboard, 1 marker, and a few tissues (erasers) on your way in.

Feel free to draw a cat while you wait for class to start.

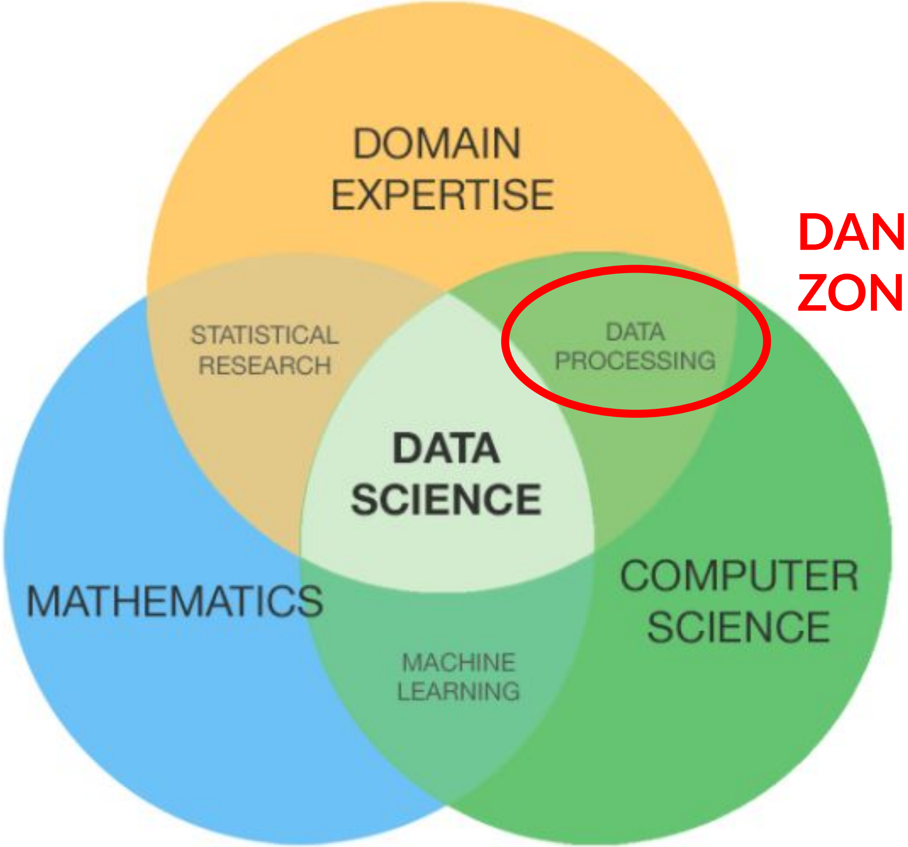
(Make sure to return these at the end of class!)

INFO 2950:
Intro to Data Science

Lecture 3
2023-08-28

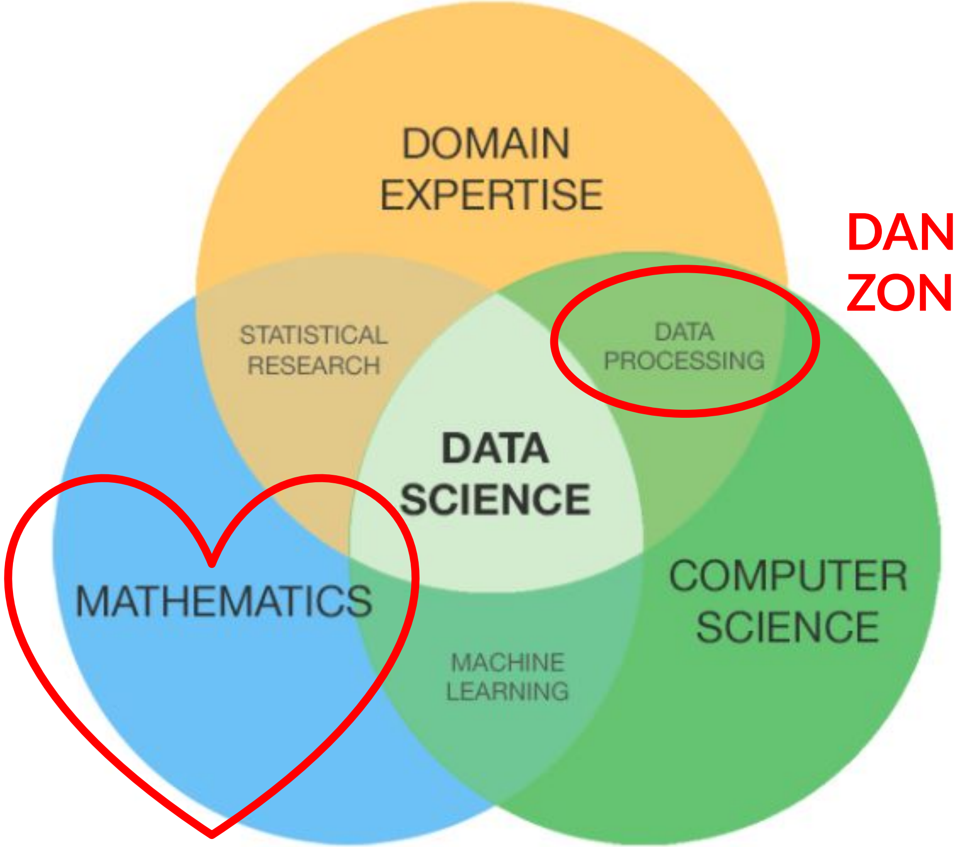
Agenda

1. Stats on single variables
2. Stats in code
3. Sorting
4. Outliers
5. SQL: inner joins
6. Admin



DANGER ZONE

Content source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram



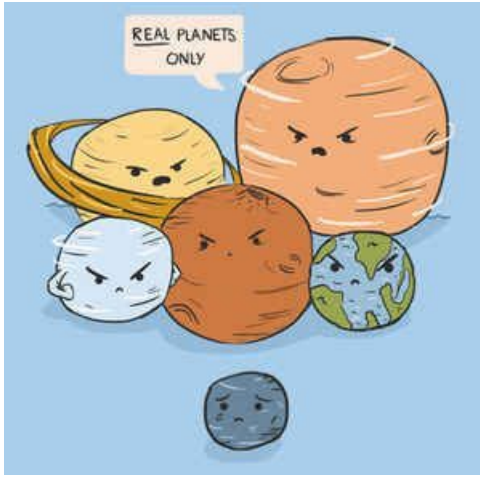
DANGER ZONE

Content source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

One-variable statistics

- **These never go out of style:**
  a. Mean
  b. Variance
  c. Median

- **The Plutos of stats:** mode, range



https://www.buzzfeed.com/lukelewis/sad-pluto

What is the difference between a **population** and a **sample**?

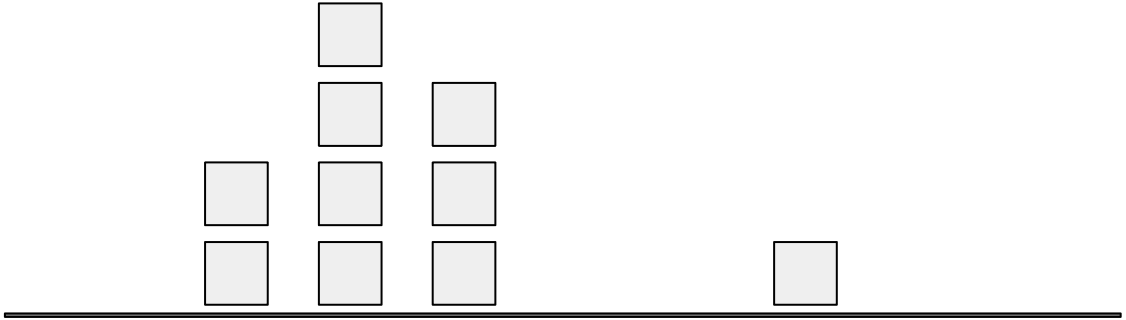
- The **population** defines what you *could* have observed

- A **sample** is the array of numbers that **you actually observed**

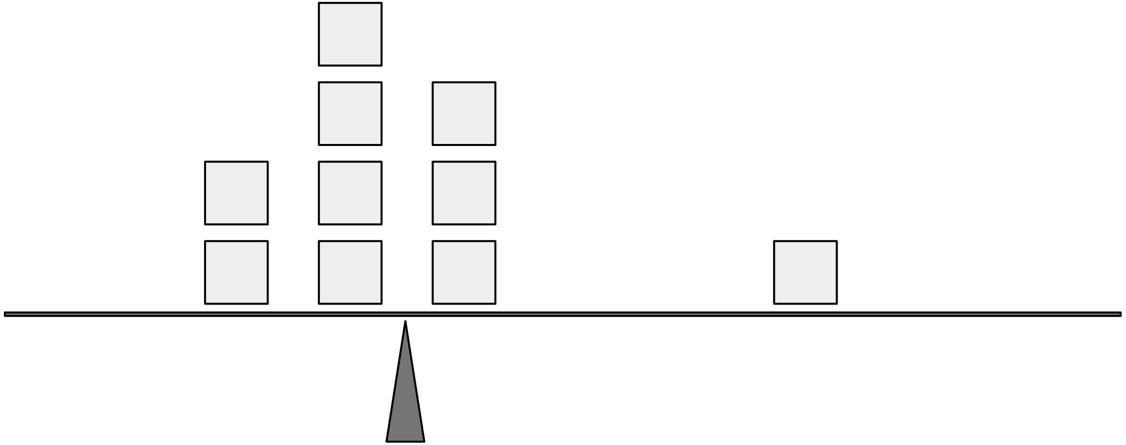
"True" values and noisy samples

- The **population** defines what you *could* have observed
  - Properties like mean **μ** and variance $\boldsymbol{\sigma}^2$ are not directly known
- A **sample** is the array of numbers that **you actually observed**
  - Sample mean $\bar{\mathbf{X}}$ and sample variance $\mathbf{s}^2$ are actual numbers that **you can calculate**
- Sample mean and variance are typically *not equal* to the population mean and variance, but they get closer with larger samples

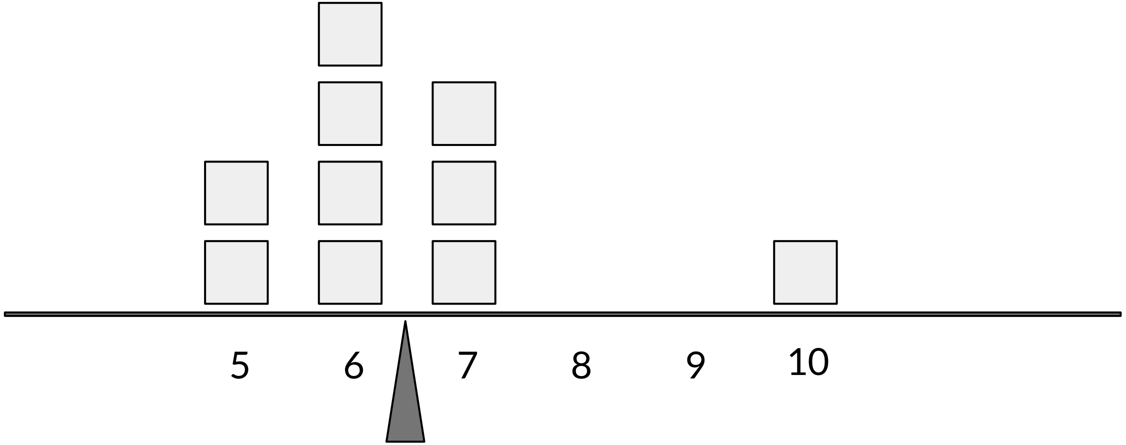
The mean is a balance point



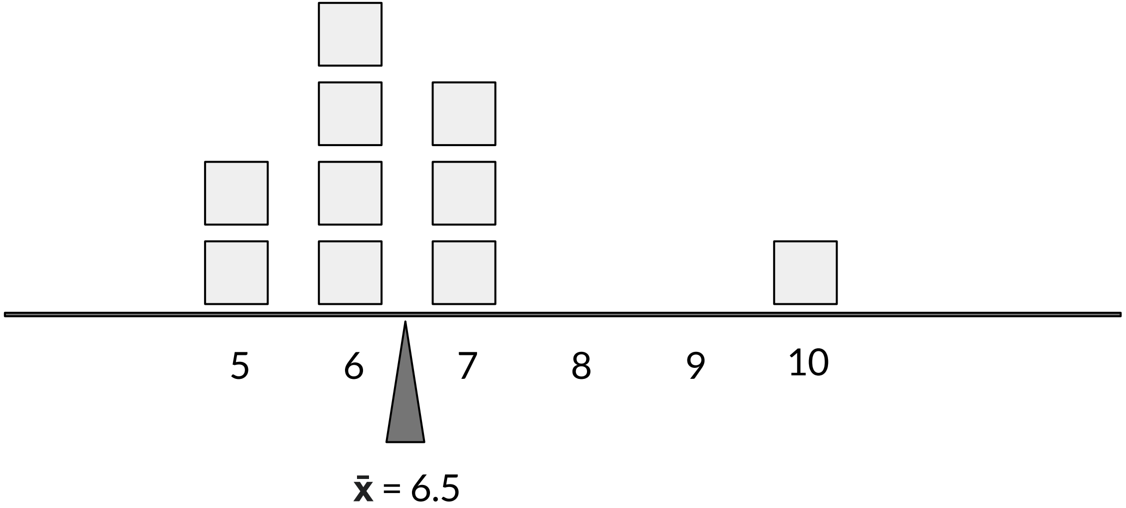
The mean is a balance point



The mean is a balance point

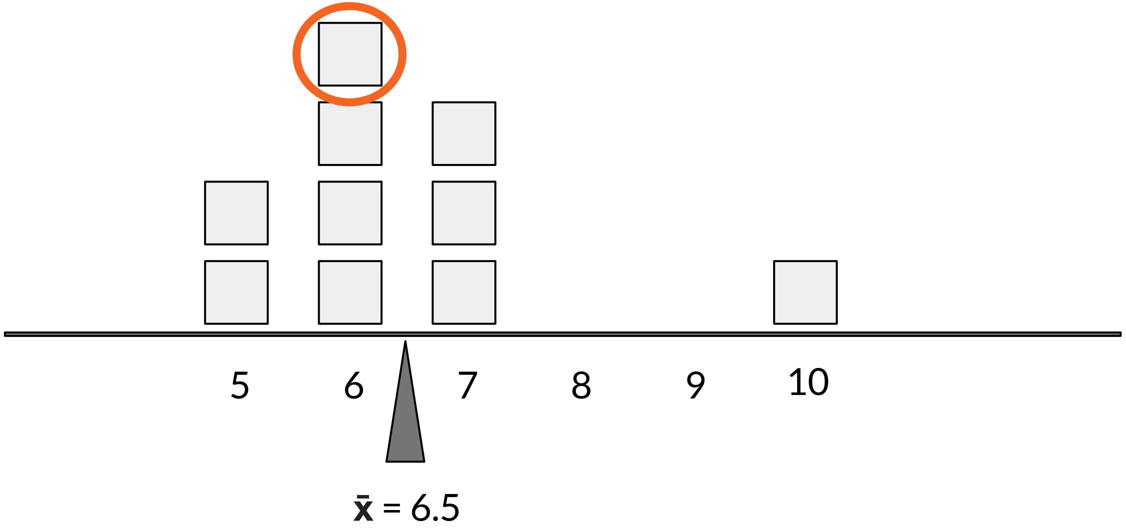


The mean is a balance point



The mean is a balance point

$X_3 = 6$



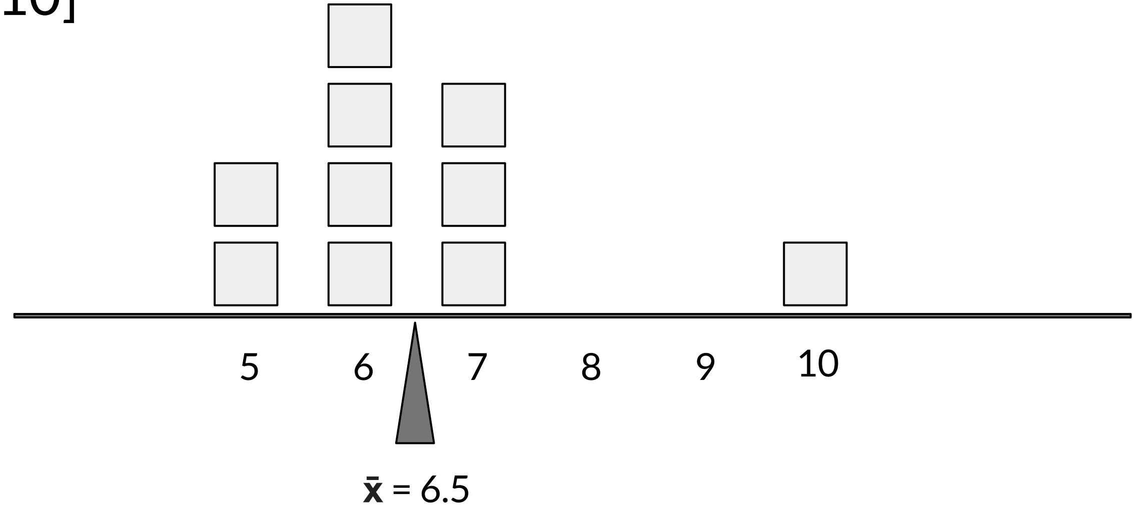
The mean is a balance point

$X = [5, 5, 6, 6, 6, 6, 7, 7, 7, 10]$

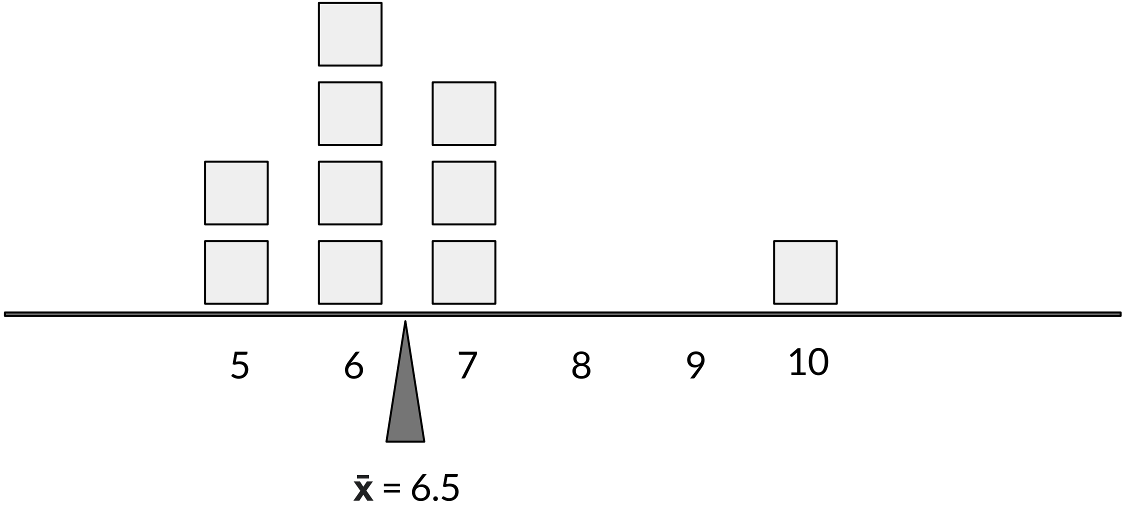
$X_3 = 6$

$\Sigma_i X_i = 65$

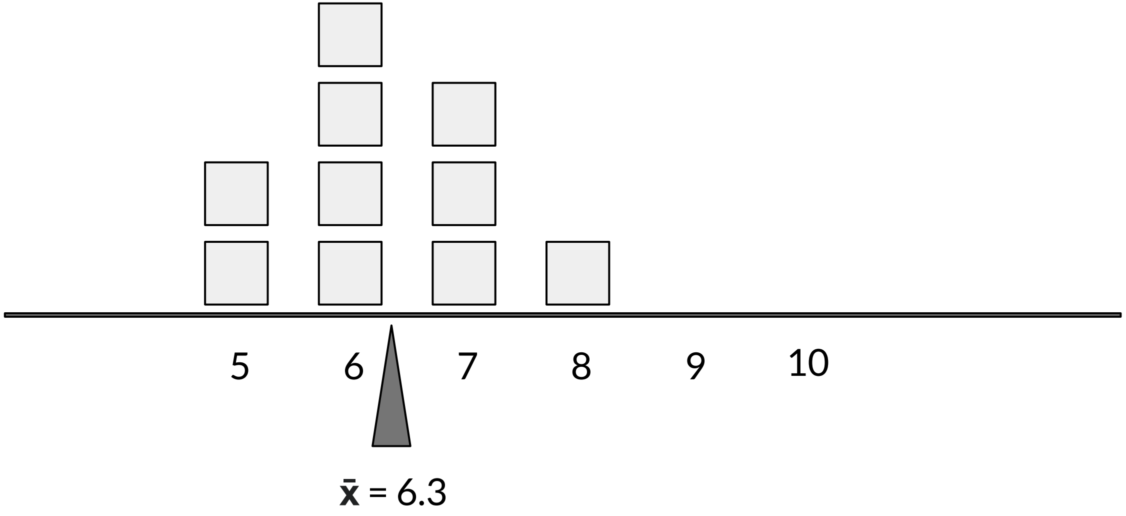
$\overline{X} = \Sigma_i X_i / N = 6.5$



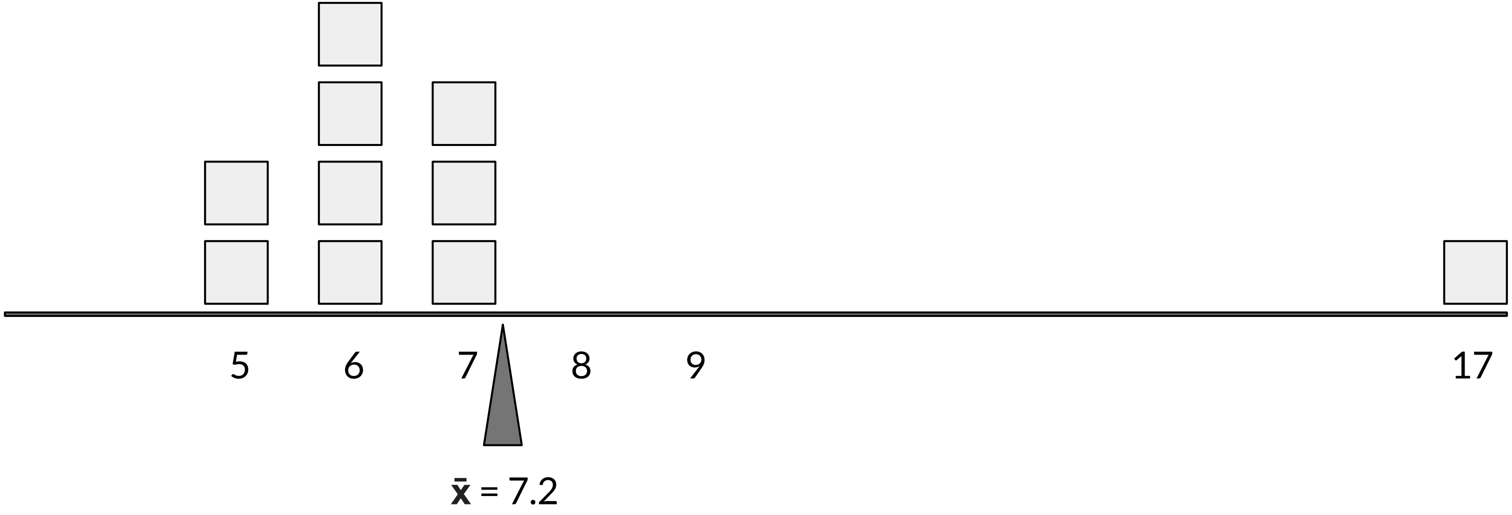
The mean is a balance point



The mean is a balance point



The mean is a balance point



Sample variance is the average squared distance to the sample mean

$$\frac{\Sigma_i (X_i - \bar{X})^2}{N}$$

—

Sample variance is the average squared distance to the sample mean

$$\frac{\Sigma_i (X_i - \bar{X})^2}{N}$$

Can the mean be negative? Can the variance? Why or why not?

—

Sample variance is the average squared distance to the sample mean

$$\frac{\Sigma_i (X_i - \bar{X})^2}{N}$$

Can the mean be negative? Yes. Can the variance? No. Why or why not? The numerator is squared, the denominator is a count.

But wait, isn't there something about N and N-1?

$$\frac{\Sigma_i (X_i - \bar{X})^2}{N - 1}$$

Which is larger, something divided by N or by (N-1)?

But wait, isn't there something about N and N-1?

$$\frac{\Sigma_i (X_i - \bar{X})^2}{N - 1}$$

Which is larger, something divided by N or by (N-1)?

If you divide by a smaller number, the result is larger

How do you do this in code?

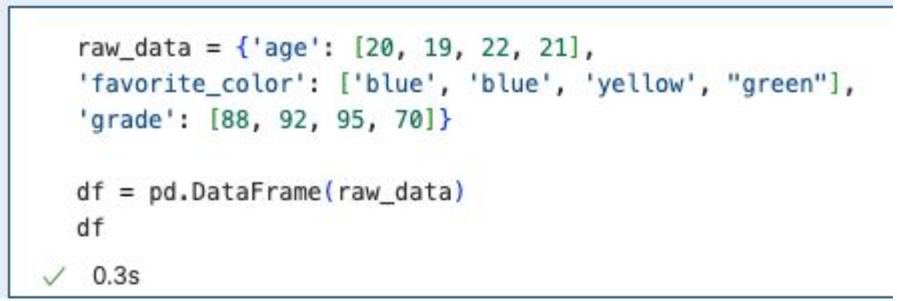
- In SQL, we talked about how making a new column out of columns is “manipulating data”

- What do we call generating a new value out of a column’s data?

How do you do this in code?

- In SQL, we talked about how making a new column out of columns is “manipulating data”
- What do we call generating a new value out of a column’s data? “summarizing/aggregating data”
- If summarizing array: numpy (next time)
- If summarizing df: pandas

Pandas stats in 1-D



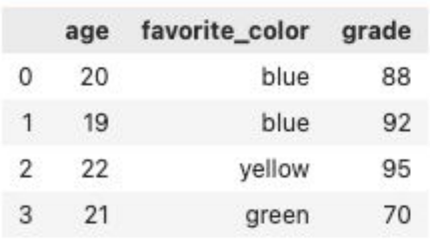
```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s

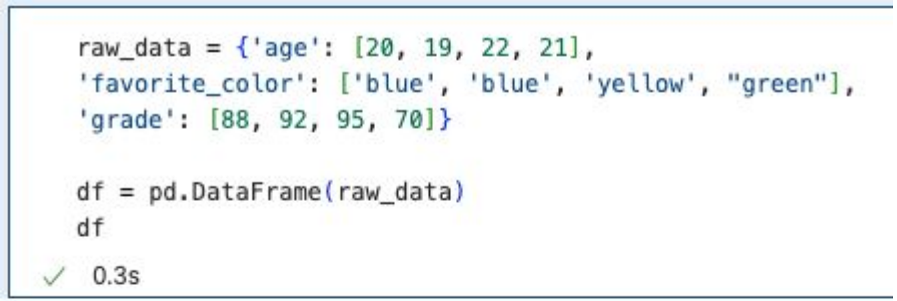
How many rows and columns?

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Pandas stats in 1-D



	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



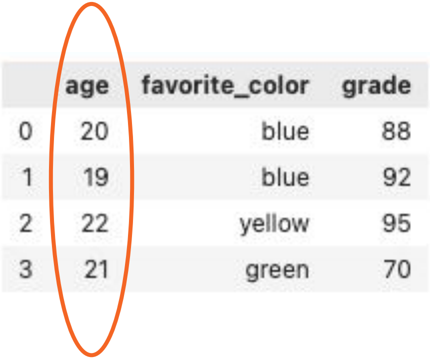
```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s

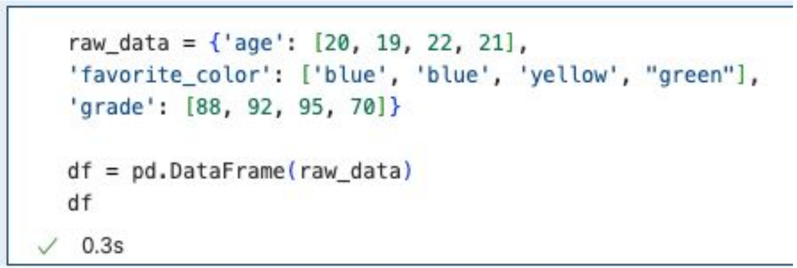
3 columns, 4 rows
Note that the index doesn’t usually get counted as a column

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Pandas stats in 1-D



	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s

```
>>> df['age'].mean()

>>> df['age'].var()

>>> df['age'].std()
```

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Table named `Fruits`

- **SELECT** 2*Q1 **AS** DoubledQ1 **FROM** Fruits;

Product	**Q1**	**Q2**
Apple	$100	$20
Banana	$50	$2
Cantaloupe	$600	$500

➡

DoubledQ1
$200
$100
$1200

SQL stats in 1-D

```
SELECT AVG(column_name)

FROM table_name

WHERE condition;
```

```
SELECT VARIANCE(column_name)

FROM table_name

WHERE condition;
```

- We know that the **mean** minimizes the sum of “squared distances”
    - $\sum(x-\mu)^2$

Medians explained

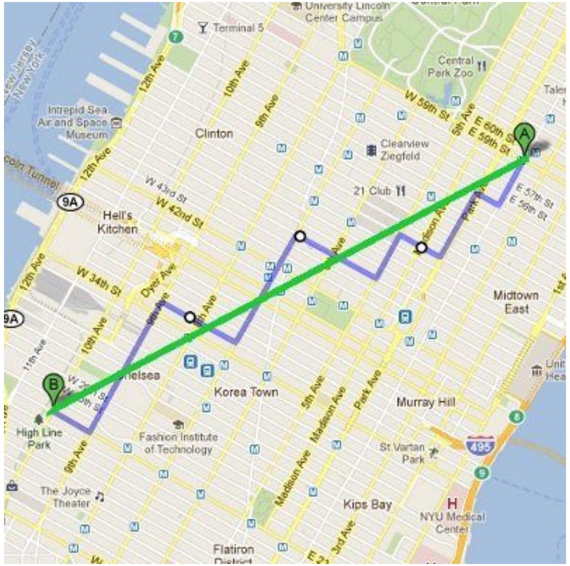
- We know that the **mean** minimizes the sum of “squared distances”
  - $\sum(x-\mu)^2$

- The **median** minimizes the sum of “absolute distances”
  - $\sum|x-m|$

- Same concept, just a different metric!

Absolute difference: why?

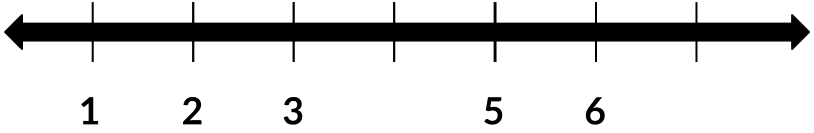
- Have you ever walked in Manhattan?



https://www.quora.com/What-is-the-difference-between-Manhattan-and-Euclidean-distance-measures

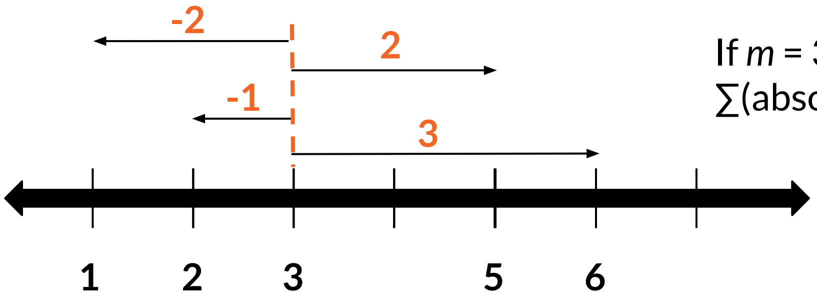
Median explained

- [1, 2, 3, 5, 6]: what is the median?



Median explained

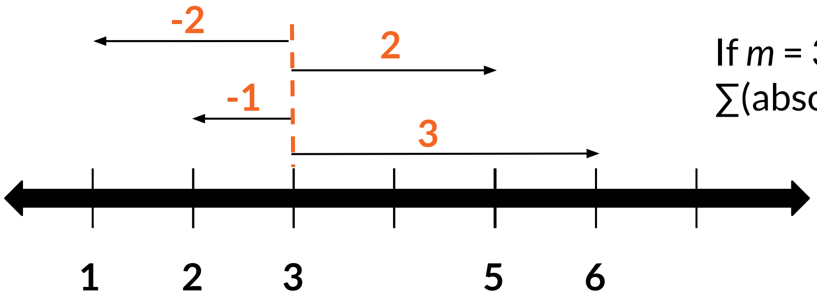
- [1, 2, 3, 5, 6]



If m = 3,
∑(absolute distance) = ?

Median explained

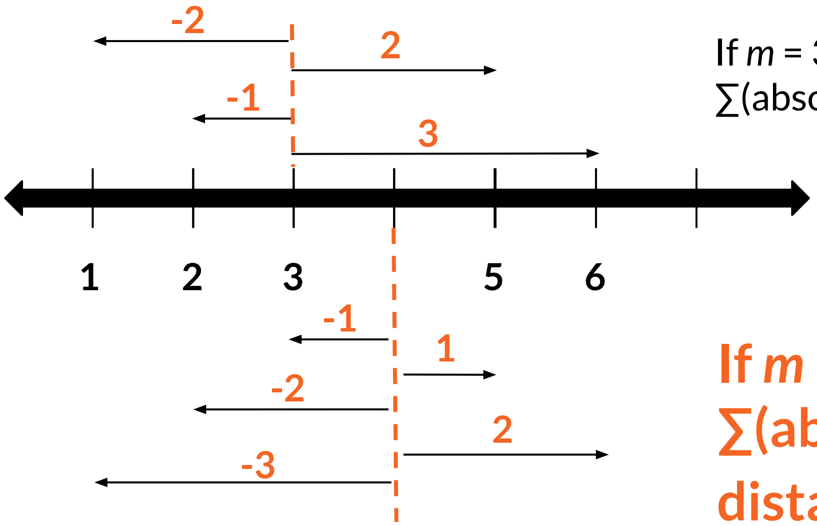
- [1, 2, 3, 5, 6]



If m = 3,
∑(absolute distance) = 8

Median explained

- [1, 2, 3, 5, 6]

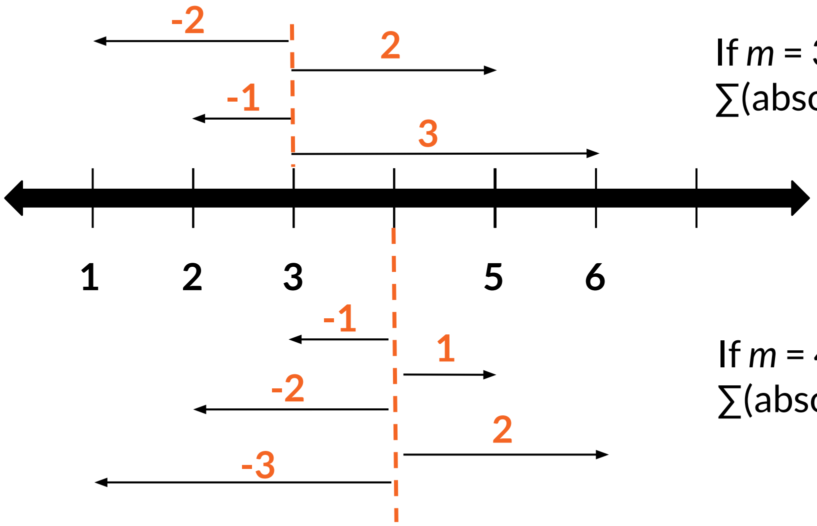


If *m* = 3,
∑(absolute distance) = 8

If *m* = 4,
∑(absolute distance) = ?

Median explained

- [1, 2, 3, 5, 6]

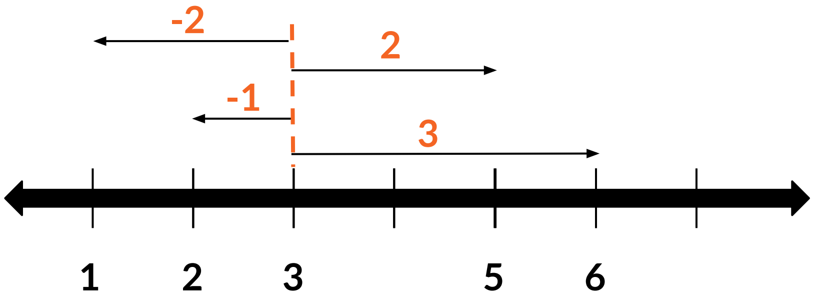


If *m* = 3,
∑(absolute distance) = 8

If *m* = 4,
∑(absolute distance) = 9

Median explained

- [1, 2, 3, 5, 6]: *m=3* minimizes ∑(absolute distance)



- Is this true generally?

Median (Math’s Version)

- We want to know when ∑|x-*m*| is minimized

- Key insight: derivative of abs() is sign()
  - **d/dm(∑|x-*m*|) = ∑sign(x-*m*)**

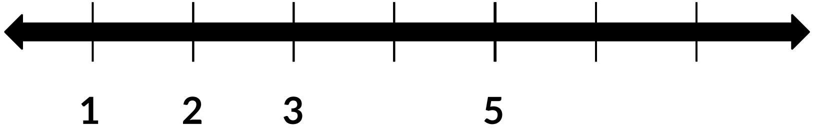
- Set the derivative = 0 to find where ∑|x-*m*| is minimized
- This only occurs when:

  # positive *(x–m) values* = the # negative *(x-m) values*

- This can only happen when *m* is the median!

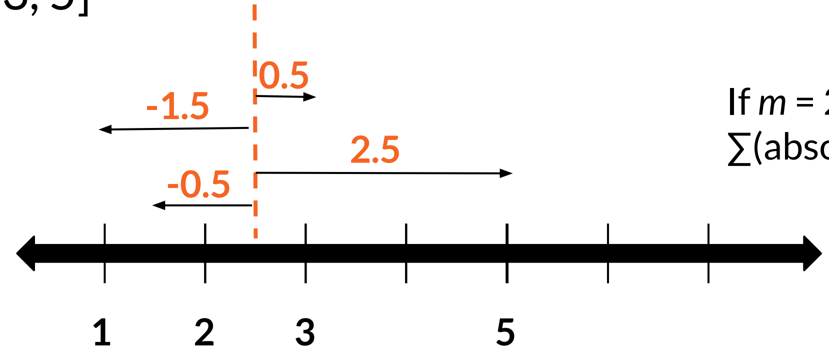
Median explained

- [1, 2, 3, 5]: what is the median?



Median in even set

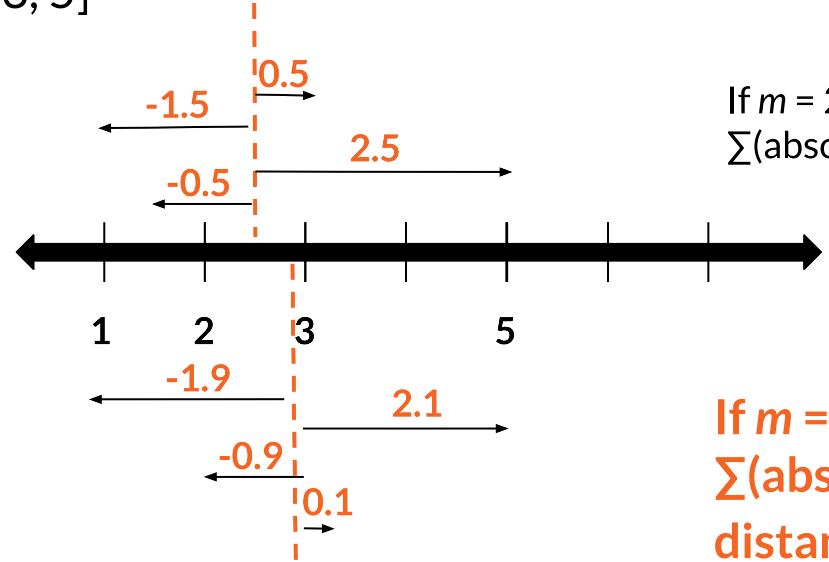
- [1, 2, 3, 5]



If *m* = 2.5,
∑(absolute distance) = 5

Median in even set

- [1, 2, 3, 5]

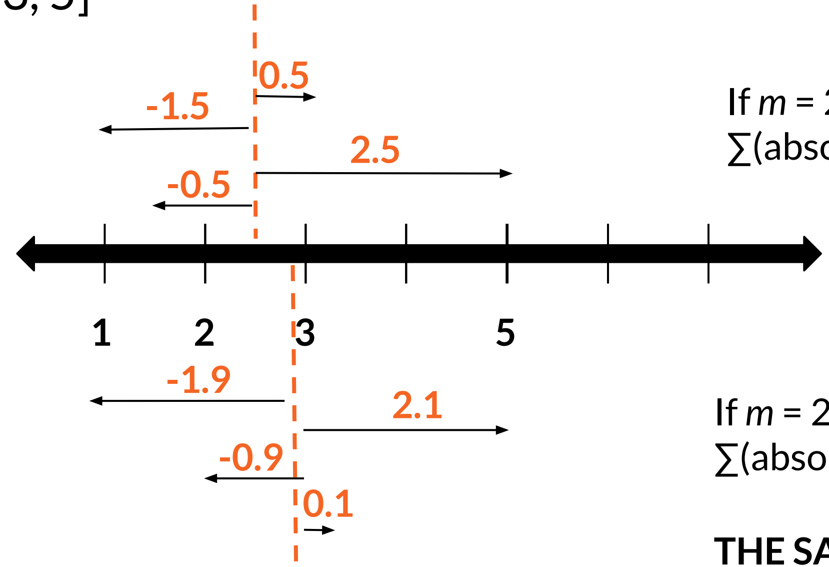


If *m* = 2.5,
∑(absolute distance) = 5

**If *m* = 2.9,
∑(absolute distance) = ?**

Median in even set

- [1, 2, 3, 5]



If *m* = 2.5,
∑(absolute distance) = 5

If *m* = 2.9,
∑(absolute distance) = 5

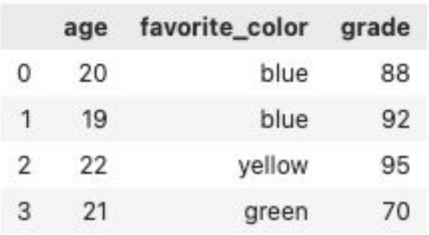
THE SAME!

Median takeaways

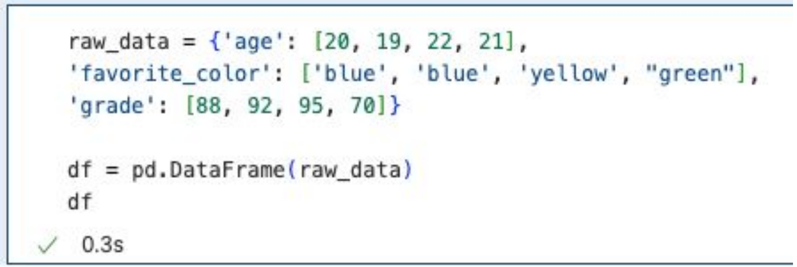
- What you learned in high school about the median needing to be the average of the middle two numbers… not necessarily!

- Because the median looks at absolute distance and not squared distance, outliers have less of an effect

Pandas stats in 1-D

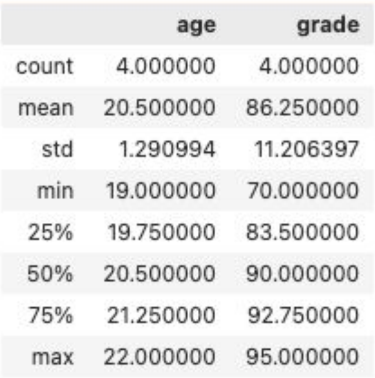


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s



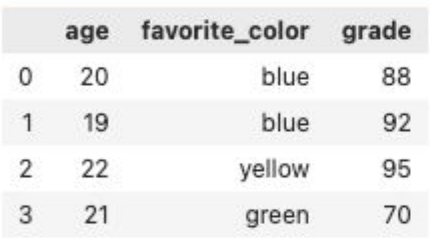
	age	grade
count	4.000000	4.000000
mean	20.500000	86.250000
std	1.290994	11.206397
min	19.000000	70.000000
25%	19.750000	83.500000
50%	20.500000	90.000000
75%	21.250000	92.750000
max	22.000000	95.000000

```
>>> df['age'].median()

>>> df.describe()
```

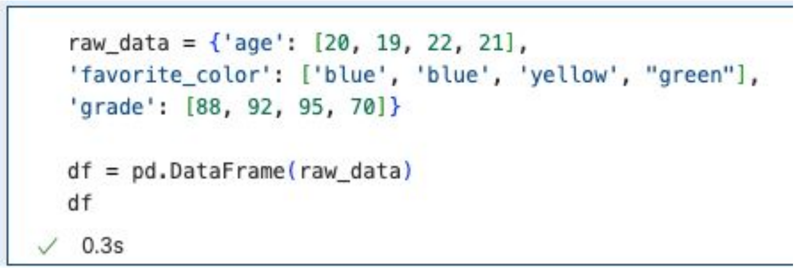
- Stats for all numeric columns
- Where is median?

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Pandas stats in 1-D

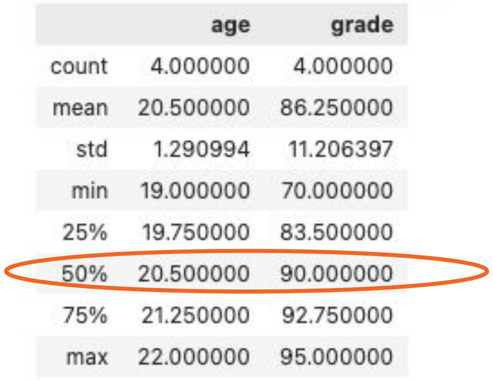


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s



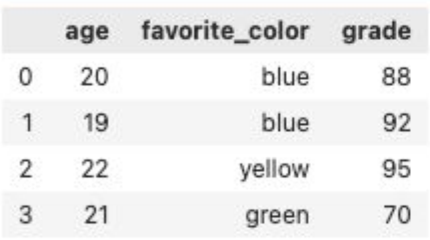
	age	grade
count	4.000000	4.000000
mean	20.500000	86.250000
std	1.290994	11.206397
min	19.000000	70.000000
25%	19.750000	83.500000
50%	20.500000	90.000000
75%	21.250000	92.750000
max	22.000000	95.000000

```
>>> df.describe()
```

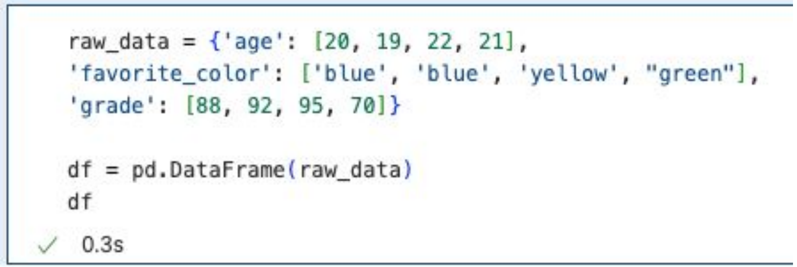
- Stats for all numeric columns
- Where is median? Where 50% of the data values are below it.

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Pandas stats in 1-D

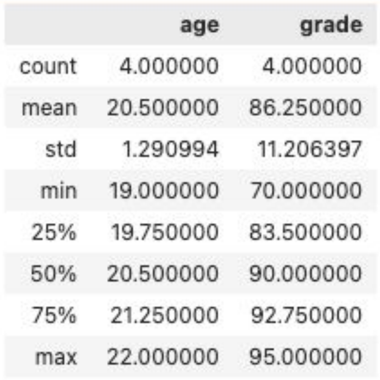


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s



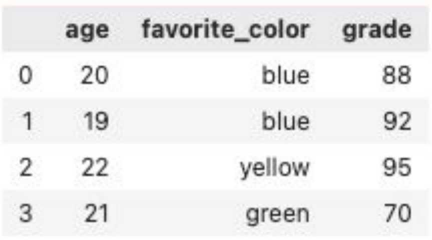
	age	grade
count	4.000000	4.000000
mean	20.500000	86.250000
std	1.290994	11.206397
min	19.000000	70.000000
25%	19.750000	83.500000
50%	20.500000	90.000000
75%	21.250000	92.750000
max	22.000000	95.000000

```
>>> df.describe()
```

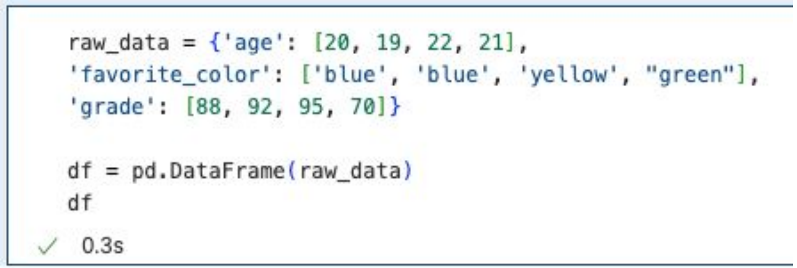
- **Stats for all numeric columns**
- **Where is median?**
- **Where is variance?**

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Pandas stats in 1-D

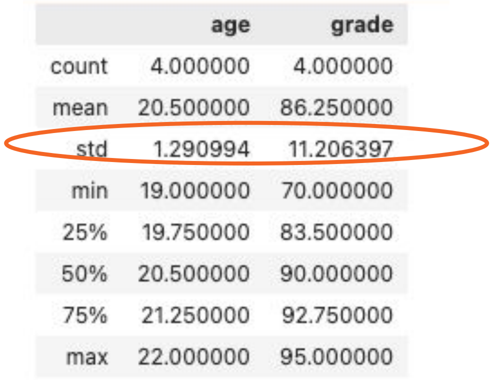


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70



```
raw_data = {'age': [20, 19, 22, 21],
'favorite_color': ['blue', 'blue', 'yellow', "green"],
'grade': [88, 92, 95, 70]}

df = pd.DataFrame(raw_data)
df
```

✓ 0.3s



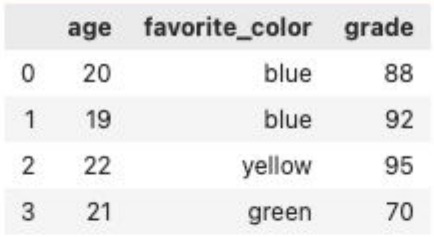
	age	grade
count	4.000000	4.000000
mean	20.500000	86.250000
std	1.290994	11.206397
min	19.000000	70.000000
25%	19.750000	83.500000
50%	20.500000	90.000000
75%	21.250000	92.750000
max	22.000000	95.000000

```
>>> df.describe()
```

- Stats for all numeric columns
- Where is median?
- Where is variance? Square the std

https://erikrood.com/Python_References/pandas_column_average_median_final.html

Which syntax issues can you find?

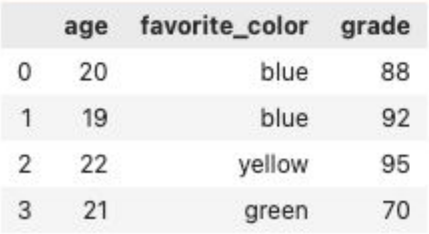


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

```
>>> df[grade].median(df)
```

```
>>> df['favorite_color'].describe()
```

Which syntax issues can you find?



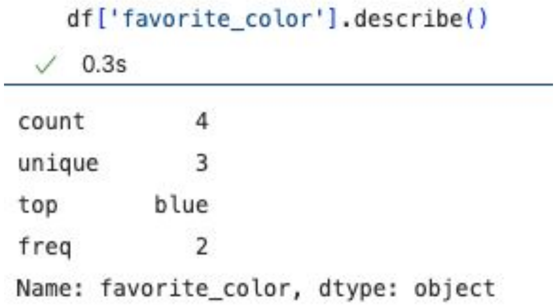
	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

~~>>> df[grade].median(df)~~

```
>>> df['grade'].median()
```

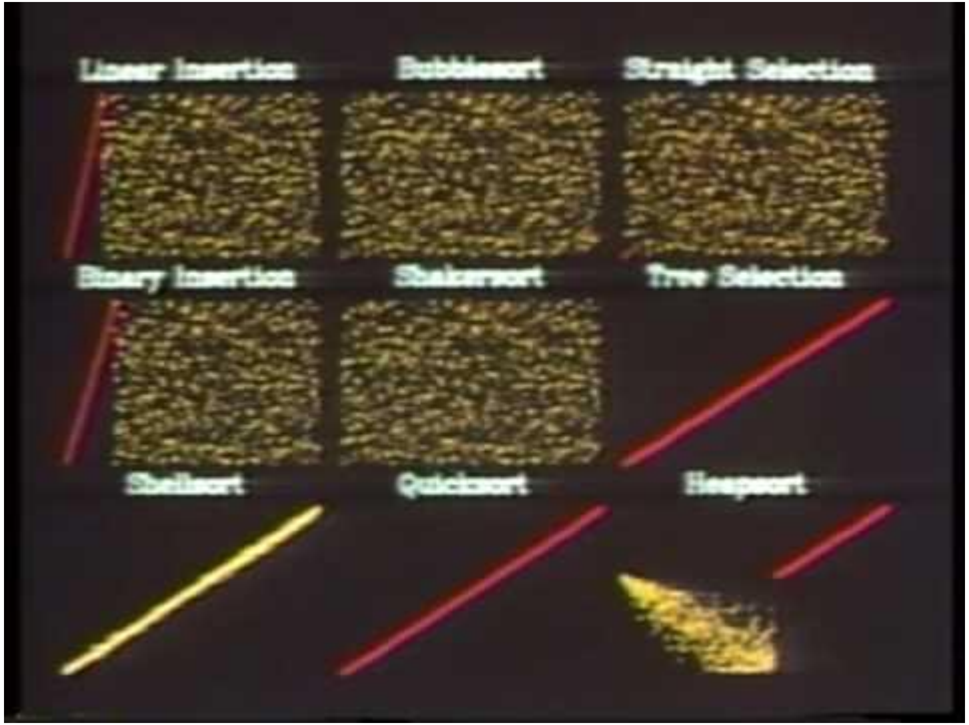
```
>>> df['favorite_color'].describe()
```

✅



```
df['favorite_color'].describe()
✓ 0.3s

count        4
unique       3
top       blue
freq         2
Name: favorite_color, dtype: object
```

Sorting out sorting



https://www.imdb.com/title/tt0210301/

Sorting out sorting (SQL)

```
SELECT * FROM season_df WHERE Position = ‘D’ ORDER BY Name LIMIT 5
```

Sorting out sorting (SQL)

```
SELECT * FROM season_df WHERE Position = ‘D’ ORDER BY Name LIMIT 5
```

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

https://www.w3schools.com/sql/sql_orderby.asp

Sorting out sorting (SQL)

```
SELECT * FROM season_df WHERE Position = ‘D’ ORDER BY Name LIMIT 5
```

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

Default is ascending sort

https://www.w3schools.com/sql/sql_orderby.asp

Sorting out sorting (SQL)

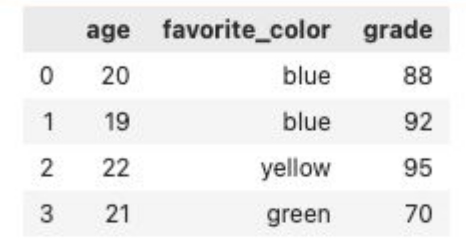
```
SELECT * FROM season_df WHERE Position = 'D' ORDER BY Name LIMIT 5
```

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

https://www.w3schools.com/sql/sql_orderby.asp

What would the new order of the age column?

Students

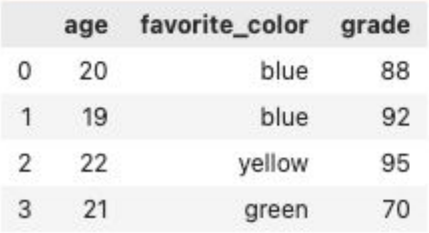


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

```
SELECT *
FROM Students
ORDER BY grade Desc;
```

What would the new order of the age column?

Students



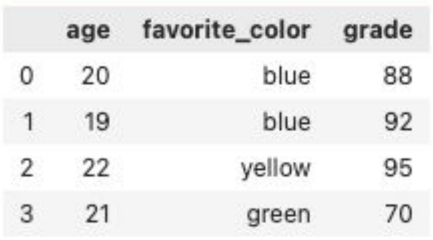
	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

```
SELECT *
FROM Students
ORDER BY grade Desc;
```

Age: 22, 19, 20, 21

What SQL command would produce this table?

Students

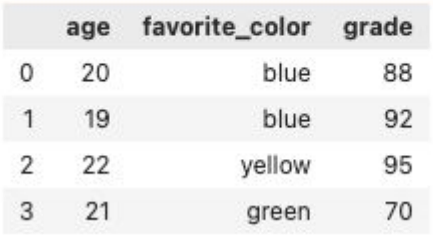


	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

	age	favorite_color	grade
3	21	green	70
0	20	blue	88
1	19	blue	92
2	22	yellow	95

What SQL command would produce this table?

Students



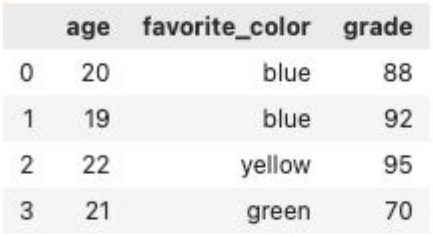
	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

	age	favorite_color	grade
3	21	green	70
0	20	blue	88
1	19	blue	92
2	22	yellow	95

```
SELECT *
FROM Students
ORDER BY grade;
```

What SQL command would produce this table?

Students



	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

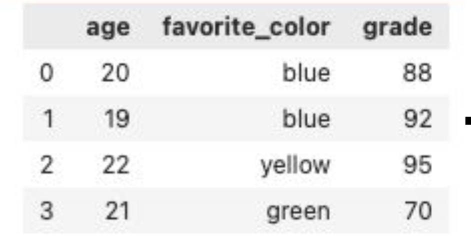
	age	favorite_color	grade
3	21	green	70
0	20	blue	88
1	19	blue	92
2	22	yellow	95

SELECT *
FROM Students
ORDER BY grade;

In Python: `duckdb.sql("SELECT * FROM Students ORDER BY grade").df()`

What about in pandas?

Students



	age	favorite_color	grade
0	20	blue	88
1	19	blue	92
2	22	yellow	95
3	21	green	70

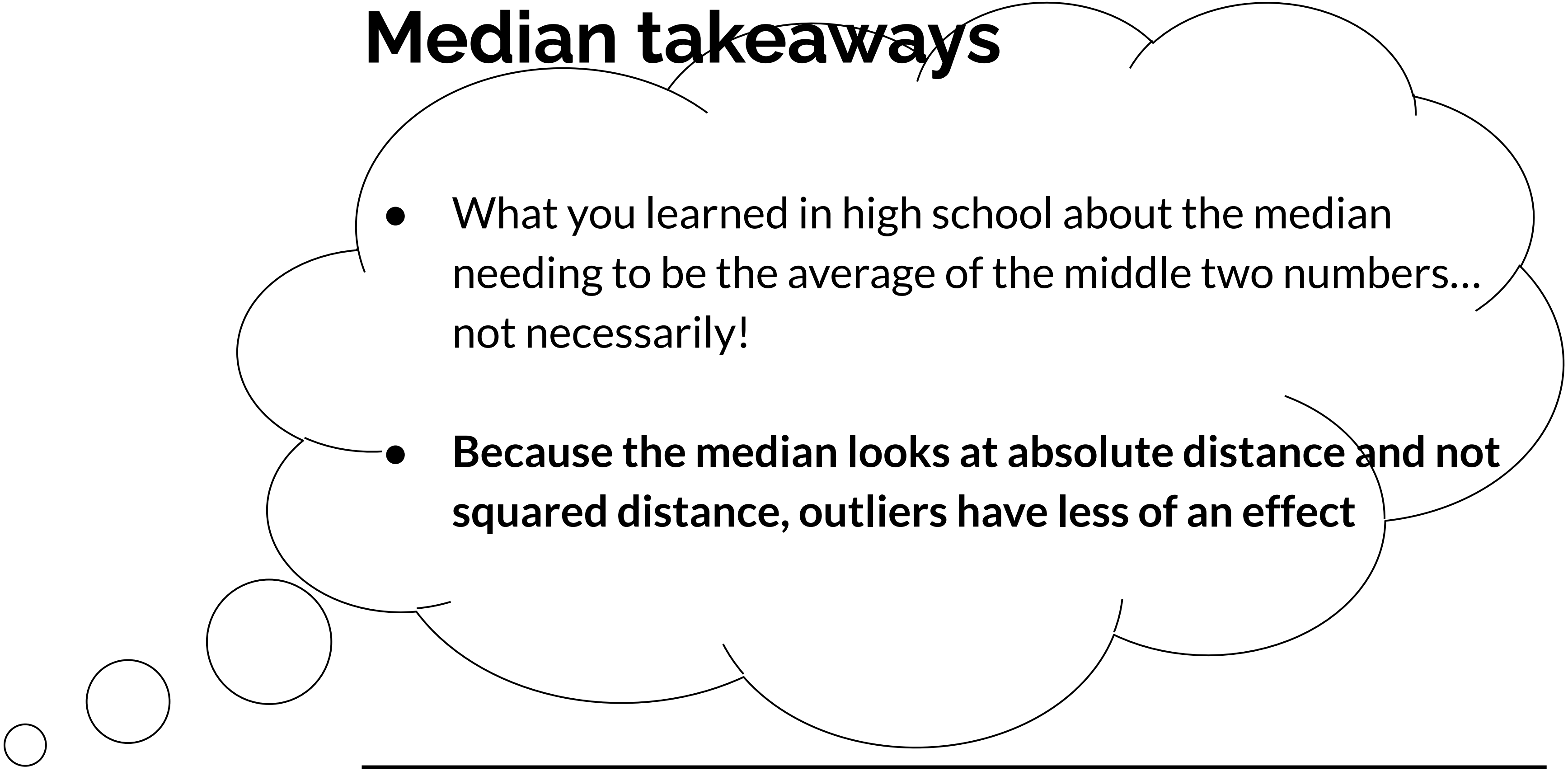
	age	favorite_color	grade
3	21	green	70
0	20	blue	88
1	19	blue	92
2	22	yellow	95

```
Students.sort_values(by = [‘grade’])
```

1 min break + Think, Pair, Share

- **When is the median a more useful statistic than mean?**

Median takeaways



- What you learned in high school about the median needing to be the average of the middle two numbers… not necessarily!

- **Because the median looks at absolute distance and not squared distance, outliers have less of an effect**

DF: age of living creatures we take care of

Prof. In Charge	Beings taken care of	Age
Prof. Mimno	Human child	16
Prof. Mimno	Human child	114
Prof. Mimno	Adult cat	1.5
Prof. Mimno	French lop rabbit	8
Prof. Mimno	Kitten	0.5
Roz	Adult Cat	7
Prof. Koenecke	Plant (dead)	0.002

Anything seem off?

🐈

:(

DF: age of living creatures we take care of

Prof. In Charge	Beings taken care of	Age
Prof. Mimno	Human child	16
Prof. Mimno	Human child	114
Prof. Mimno	Adult cat	1.5
Prof. Mimno	French lop rabbit	8
Prof. Mimno	Kitten	0.5
Roz	Adult Cat	7
Prof. Koenecke	Plant (dead)	0.002

Use domain expertise

Check units

Anything seem off?

Some basic stats in LaTeX...

a = [16, 114, 1.5, 8, 0.5, 7, 0.002]

● \bar{a}	$\bar{a}$	**Mean**
● \eta_{a}	η_{a}	**Median**
● \sigma_{a}^2	σ_{a}^2	**Variance**

Some basic stats in LaTeX...

a = [16, 114, 1.5, 8, 0.5, 7, 0.002]

- **\bar{a}** = 147.002/7 = 21.00 $\bar{a}$
- **\eta_{a}** = 7 η_a
- **\sigma_{a}^2** = 1713.40 σ_{a}^{2}

If outliers are removed, what happens to the stats?

a = [16, 114, 1.5, 8, 0.5, 7, 0.002]   a = [16, 1.5, 8, 0.5, 7]

- $\bar{a} = 147.002/7 = 21.00$ ⬆ = ⬇
- $\eta_{a} = 7$ ⬆ = ⬇
- $\sigma_{a}^2 = 1713.40$ ⬆ = ⬇

If outliers are removed, what happens to the stats?

a = [16, 114, 1.5, 8, 0.5, 7, 0.002]		a = [16, 1.5, 8, 0.5, 7]
● \bar{a} = 147.002/7 = 21.00	↓	● \bar{a} = 33/5 = 6.6
● \eta_{a} = 7	=	● \eta_{a} = 7
● \sigma_{a}^2 = 1713.40	↓	● \sigma_{a}^2 = 38.43

Think, pair, share: what happens if you remove outliers generally?

a = [16, 114, 1.5, 8, 0.5, 7, 0.002]

- \bar{a} = 147.002/7 = 21.00
- \eta_{a} = 7
- \sigma_{a}^2 = 1713.40

Any outlier removal

- Always ⬇ ? Y/N
- Always = ? Y/N
- Always ⬇ ? Y/N

Think, pair, share: what happens if you remove outliers generally?

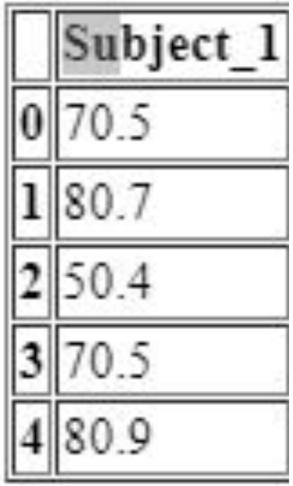
a = [16, 114, 1.5, 8, 0.5, 7, 0.002]	Any outlier removal
● \bar{a} = 147.002/7 = 21.00	● Not always ⬇
● \eta_{a} = 7	● Not always =
● \sigma_{a}^2 = 1713.40	● Always ⬇ !

Outlier takeaways

- You should remove outliers if they represent measurement or data errors
- Removing outliers will decrease your variance
- Do not remove data that are not outliers in an attempt to decrease variability

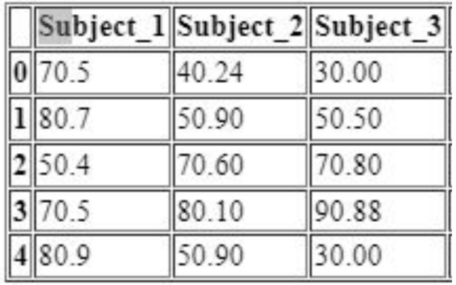
How to check for outliers?

- Draw a plot to visualize a single-variable dataset (e.g. this one)



	Subject_1
0	70.5
1	80.7
2	50.4
3	70.5
4	80.9

Histograms with pandas

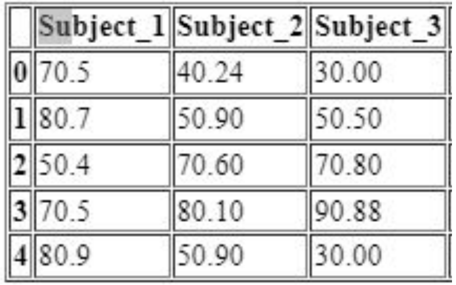


	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00

```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```

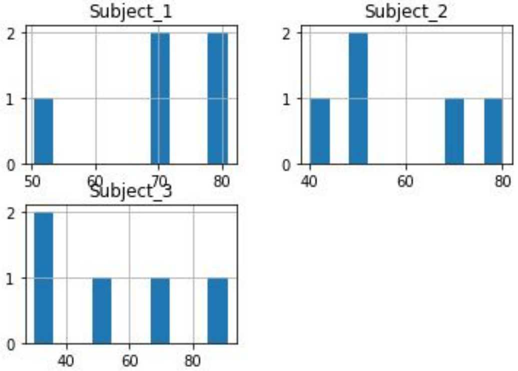
https://www.machinelearningplus.com/pandas/pandas-histogram/

Histograms with pandas



	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00

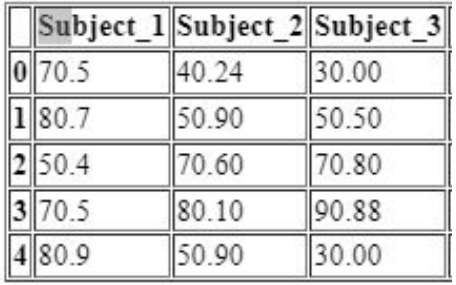
```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```



```
df.hist();
```

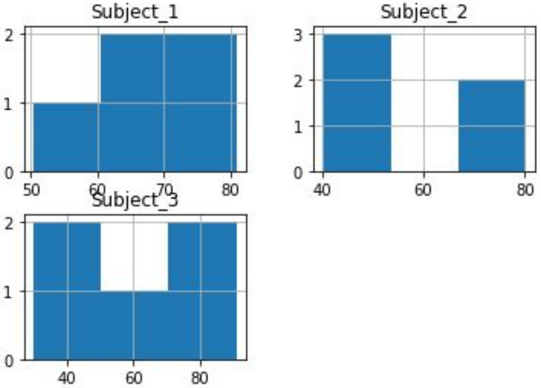
https://www.machinelearningplus.com/pandas/pandas-histogram/

Histograms with pandas



	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00

```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```

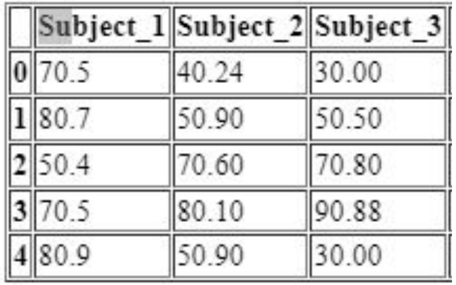


```
df.hist(bins=3);
```

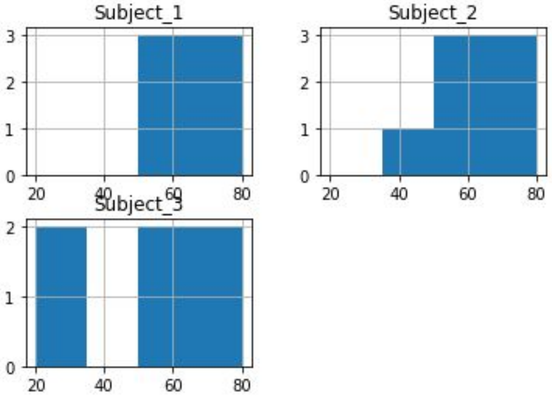
This makes 3 category bins for each facet

https://www.machinelearningplus.com/pandas/pandas-histogram/

Histograms with pandas



	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00



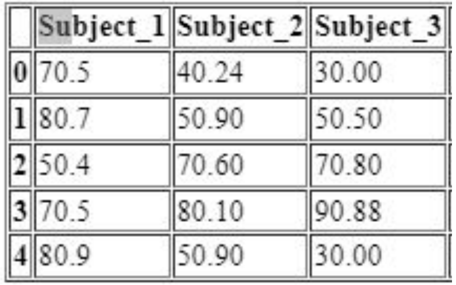
```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```

How many bins is this?

```
df.hist(bins=[20, 35, 50, 80]);
```

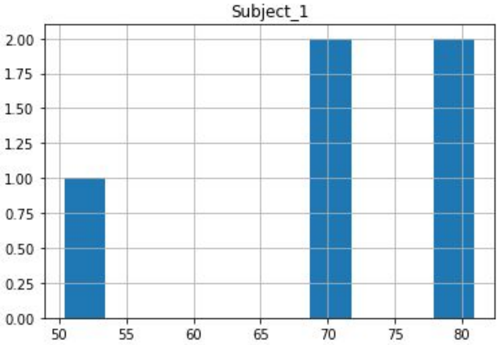
https://www.machinelearningplus.com/pandas/pandas-histogram/

Histograms with pandas



	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00

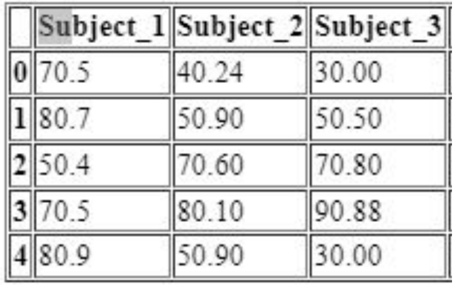
```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```



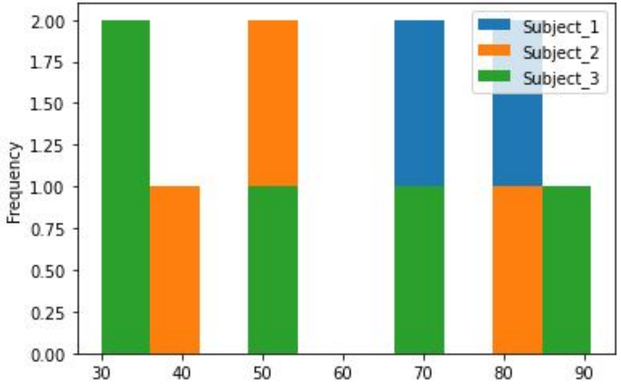
```
df.hist(column='Subject_1');
```

https://www.machinelearningplus.com/pandas/pandas-histogram/

Histograms with pandas



	Subject_1	Subject_2	Subject_3
0	70.5	40.24	30.00
1	80.7	50.90	50.50
2	50.4	70.60	70.80
3	70.5	80.10	90.88
4	80.9	50.90	30.00

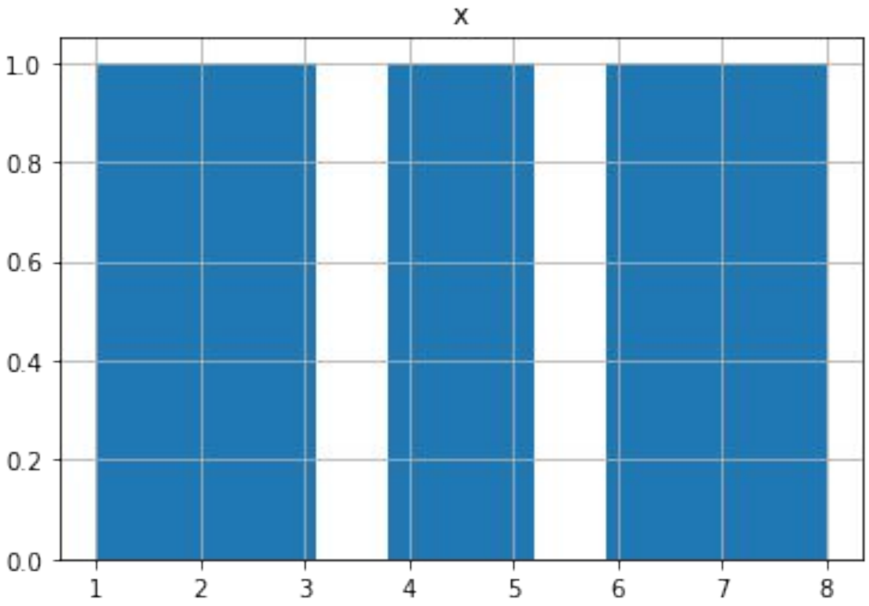


```
import pandas as pd
import numpy as np
df = pd.DataFrame({
    'Subject_1': [70.5, 80.7, 50.4, 70.5, 80.9],
    'Subject_2': [40.24, 50.9, 70.6, 80.1, 50.9],
    'Subject_3': [30, 50.5, 70.8, 90.88, 30]
})
```

```
df.plot(kind='hist');
```

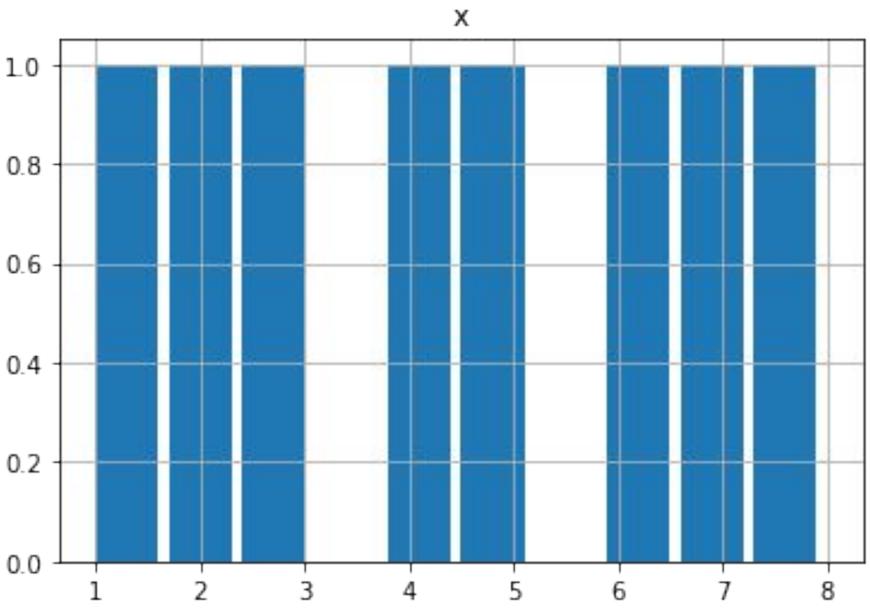
https://www.machinelearningplus.com/pandas/pandas-histogram/

Why does this histogram look so janky?



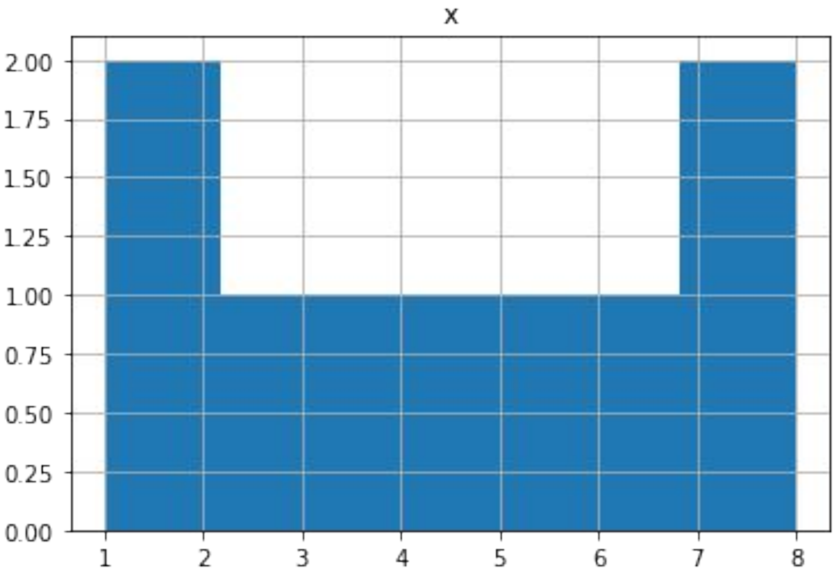
```
import pandas as pd
import numpy as np

uniform_df = pd.DataFrame({"x":
    np.array([1, 2, 3, 4, 5, 6, 7, 8])})

uniform_df.hist(column="x")
```

Why does this histogram look so janky?



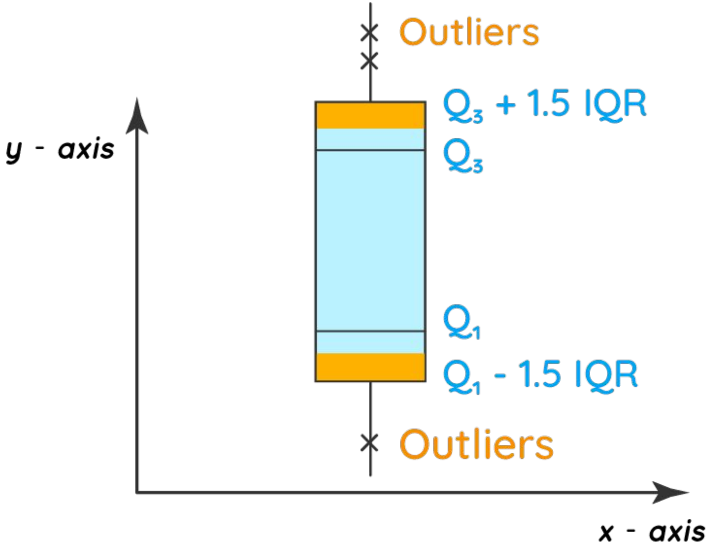
```
import pandas as pd
import numpy as np

uniform_df = pd.DataFrame({"x":
    np.array([1, 2, 3, 4, 5, 6, 7, 8])})

uniform_df.hist(column="x", width=0.6)
```

Why does this histogram look so janky?



```
import pandas as pd
import numpy as np

uniform_df = pd.DataFrame({"x":
    np.array([1, 2, 3, 4, 5, 6, 7, 8])})

uniform_df.hist(column="x", bins=6)
```

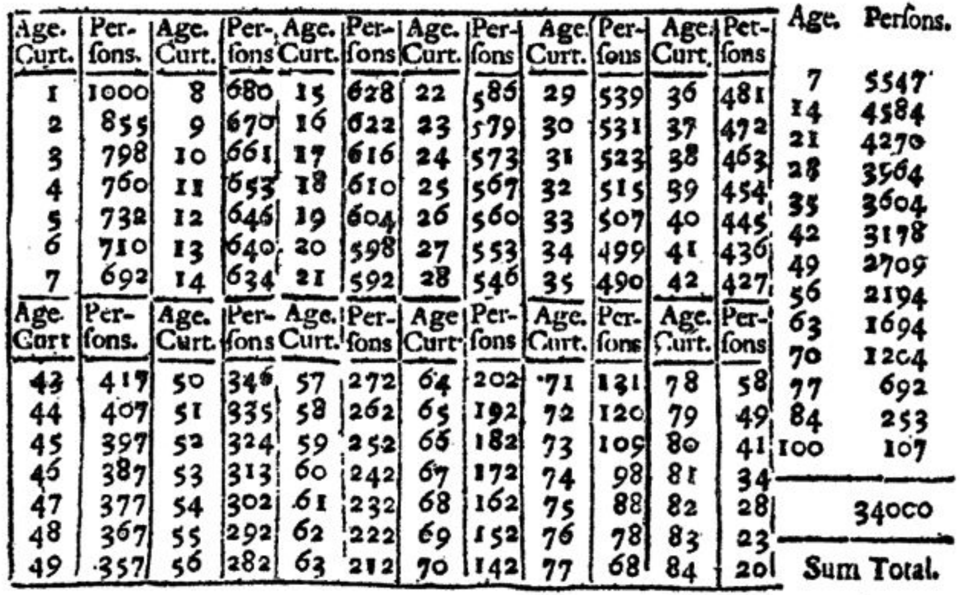
What about the outlier formula?



https://www.cuemath.com/outlier-formula/

- **When are neither mean nor median all that informative?**

Halley's Life Table (1693)



Age. Curt.	Per-ſons.	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons
1	1000	8	680	15	628	22	585	29	539	36	481
2	855	9	670	16	622	23	579	30	531	37	472
3	798	10	661	17	616	24	573	31	523	38	463
4	760	11	653	18	610	25	567	32	515	39	454
5	732	12	646	19	604	26	560	33	507	40	445
6	710	13	640	20	598	27	553	34	499	41	436
7	692	14	634	21	592	28	546	35	490	42	427
Age. Curt	Per-ſons.	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons	Age Curt	Per-ſons	Age. Curt.	Per-ſons	Age. Curt.	Per-ſons
43	417	50	346	57	272	64	202	71	131	78	58
44	407	51	335	58	262	65	192	72	120	79	49
45	397	52	324	59	252	66	182	73	109	80	41
46	387	53	313	60	242	67	172	74	98	81	34
47	377	54	302	61	232	68	162	75	88	82	28
48	367	55	292	62	222	69	152	76	78	83	23
49	357	56	282	63	212	70	142	77	68	84	20

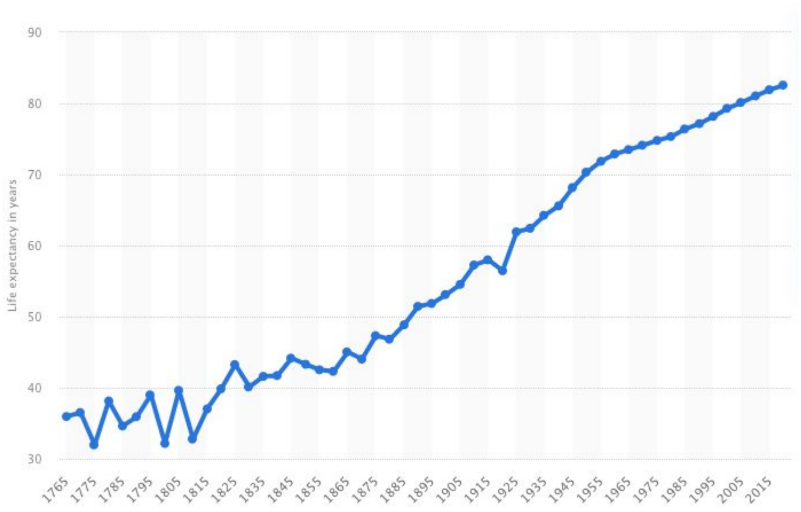
Age.	Perſons.
7	5547
14	4584
21	4270
28	3964
35	3604
42	3178
49	2709
56	2194
63	1694
70	1204
77	692
84	253
100	107
	34000
Sum Total.	

https://fairmlbook.org/classification.html

What was the average age of death in the 1800s?

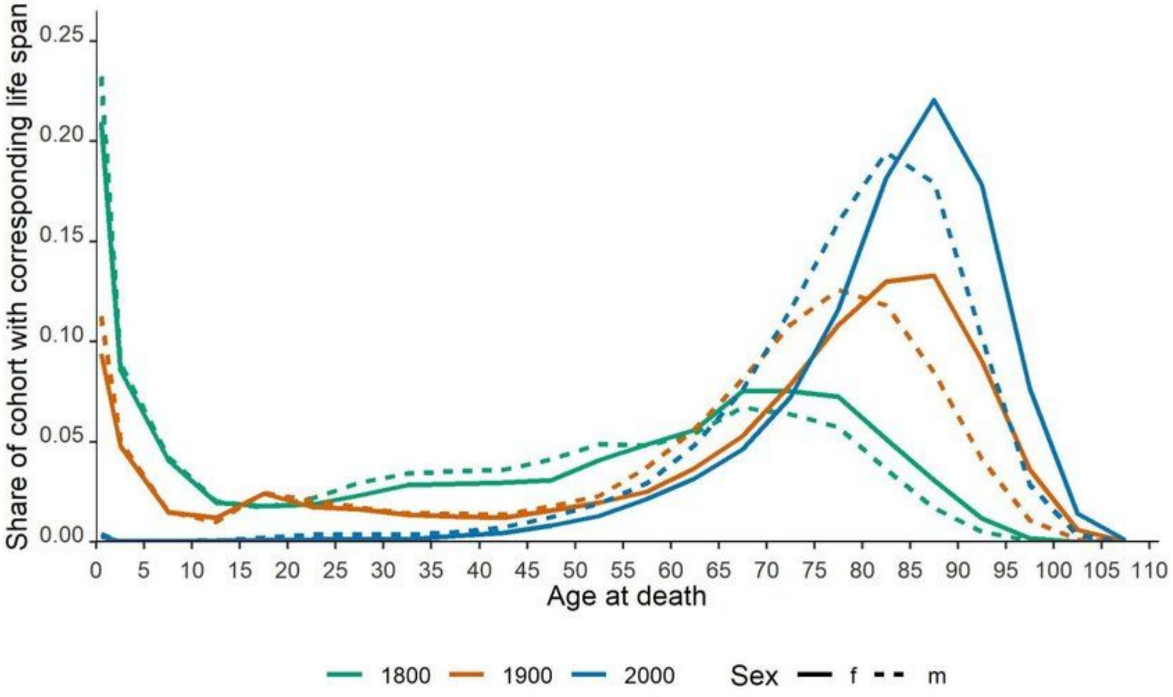
What was the average age of death in the 1800s?

- Life expectancy in Sweden, 1850**: 43.3 years**



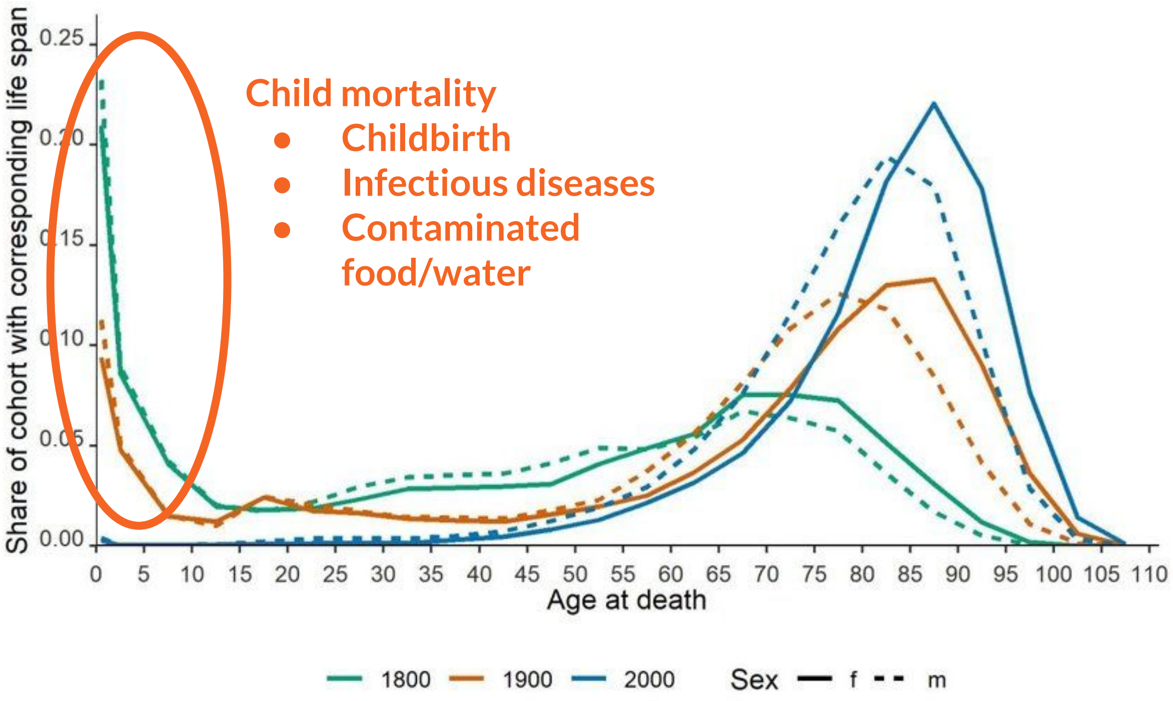
https://www.statista.com/statistics/1041305/life-expectancy-sweden-all-time/

Swedish ages at death



https://www.researchgate.net/publication/339721309_Exceptional_lives_extraordinary_families_Familial_clustering_of_longevity_in_the_19th_and_early_20th_centuries

Swedish ages at death



Child mortality

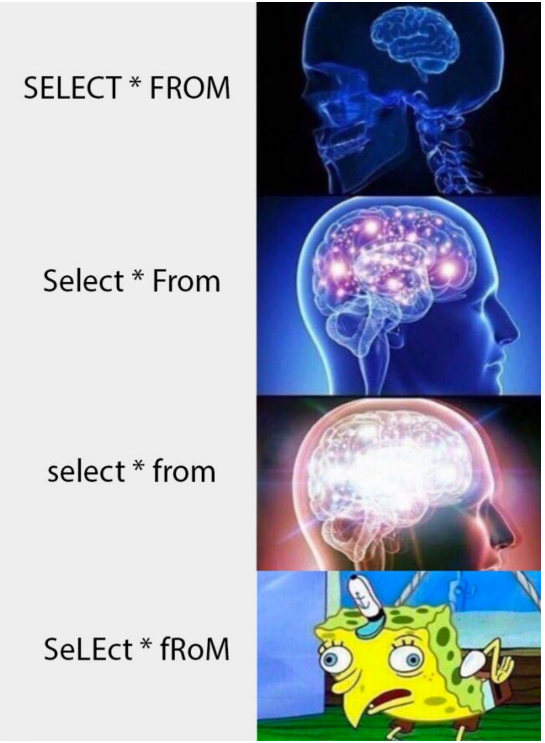
- Childbirth
- Infectious diseases
- Contaminated food/water

https://www.researchgate.net/publication/339721309_Exceptional_lives_extraordinary_families_Familial_clustering_of_longevity_in_the_19th_and_early_20th_centuries

When mean/median aren't meaningful

- E.g., on bimodal data
- Visualize & inspect the distribution
- Make sure to come up with a metric that *is* meaningful
  - Average adult age of death
    - Survival after age 50
  - QALY (quality-adjusted life year)

1 min break + attendance + back to SQL!



https://tinyurl.com/yk878c45

https://medium.com/google-cloud/sql-is-structured-query-language-but-the-structure-can-be-different-in-different-databases-3f38b90c319

SQL stats in 1-D

```
SELECT AVG(column_name)

FROM table_name

WHERE condition;
```

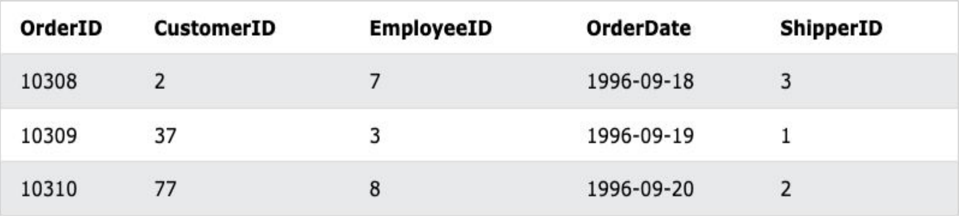
```
SELECT VARIANCE(column_name)

FROM table_name

WHERE condition;
```

Extending your data

- Make your data “bigger” by combining datasets!
- Different languages, different terms: merges, joins, …
- Types of SQL joins:
  - INNER JOIN
  - FULL JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - SELF JOIN

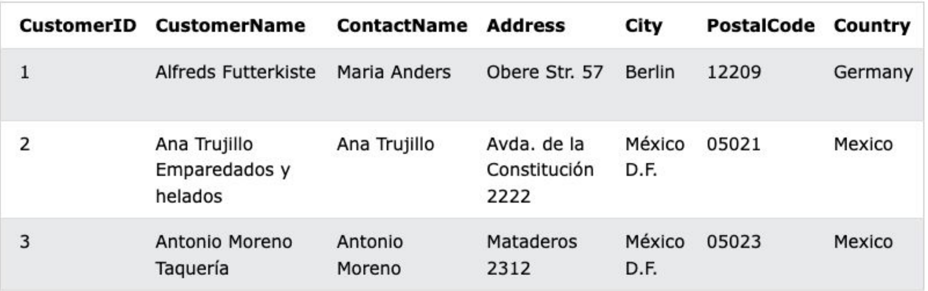
A tale of 2 tables

Table: ‘**Orders**’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

Table: ‘**Customers**’

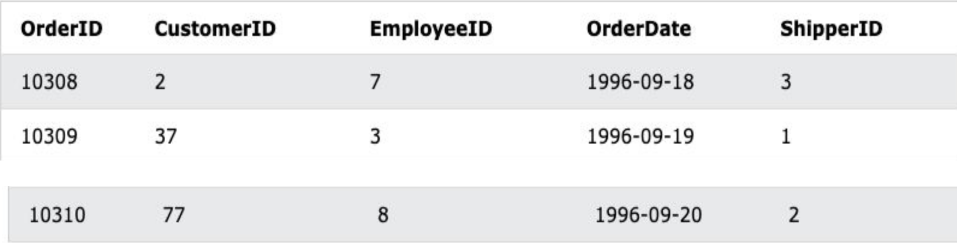


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

https://www.w3schools.com/sql/sql_join_inner.asp

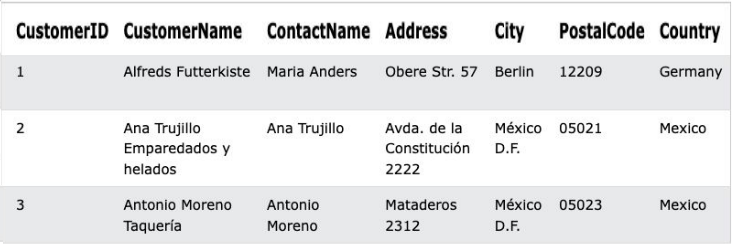
Can we do this? Should we do this?

Table: ‘**Orders**’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

Table: ‘**Customers**’

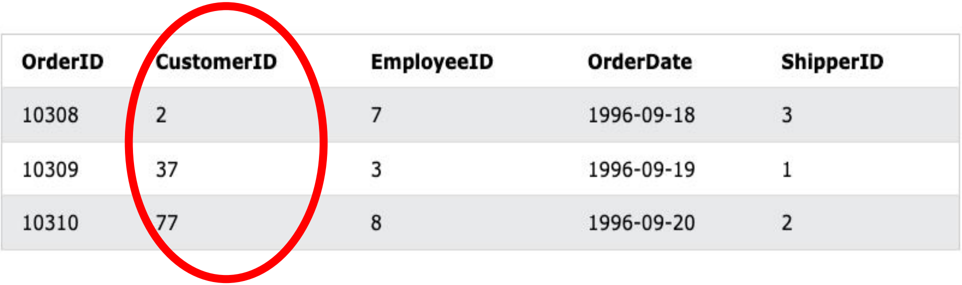


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

https://www.w3schools.com/sql/sql_join_inner.asp

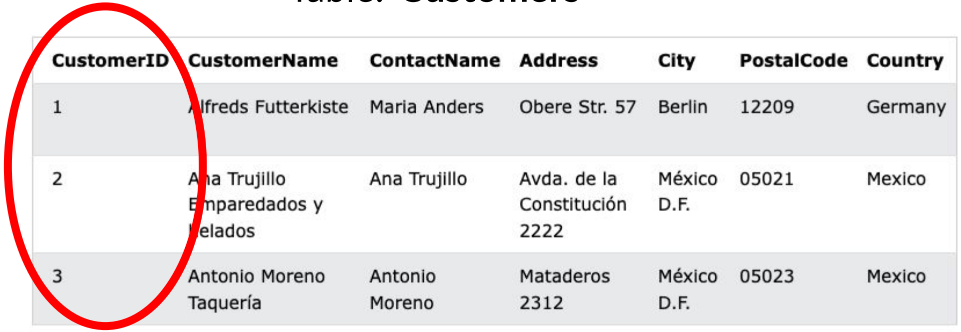
Notice: commonalities

Table: ‘**Orders**’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

Table: ‘**Customers**’

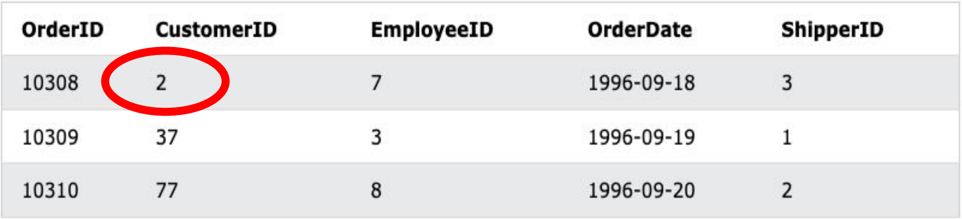


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

https://www.w3schools.com/sql/sql_join_inner.asp

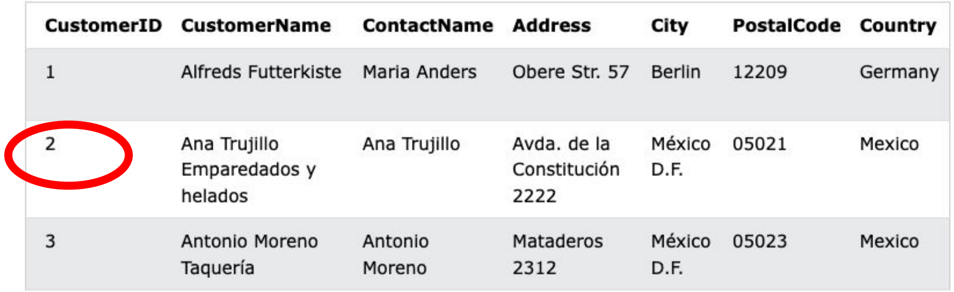
Notice: commonalities

Table: ‘**Orders**’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

Table: ‘**Customers**’

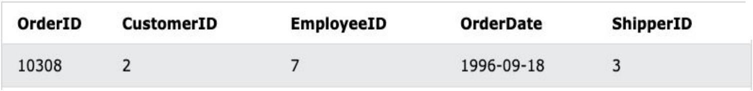


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

https://www.w3schools.com/sql/sql_join_inner.asp

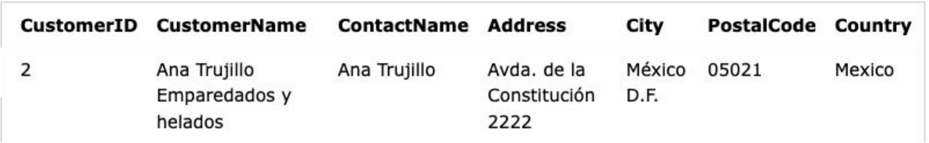
How do we get Ana’s info?

Table: ‘Orders’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3

Table: ‘Customers’

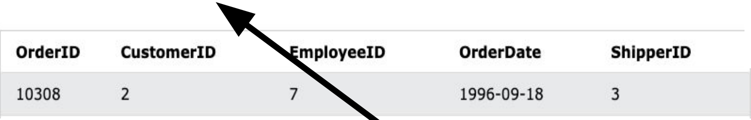


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

https://www.w3schools.com/sql/sql_join_inner.asp

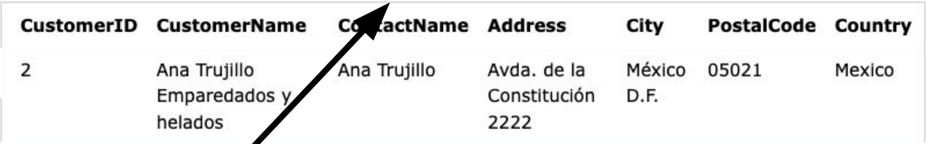
INNER JOIN

Table: ‘Orders’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3

Table: ‘Customers’



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

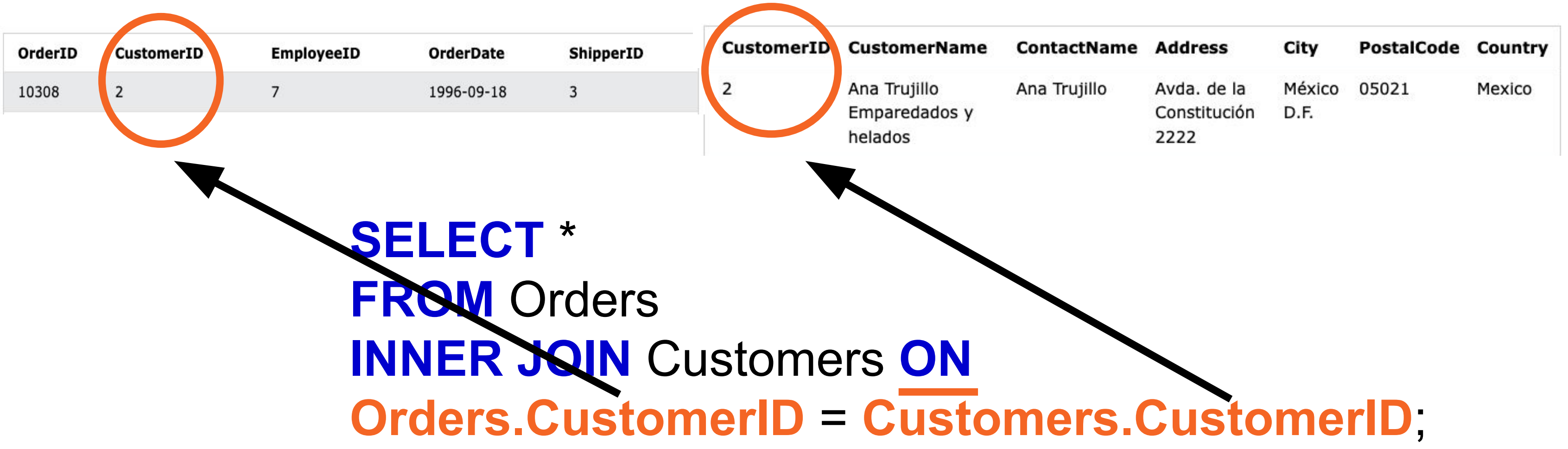
```
SELECT *
FROM Orders
INNER JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```

Note: if we did this, CustomerID would appear twice in the resulting joined table, so it would get relabeled as e.g. Orders.CustomerID and Customers.CustomerID

INNER JOIN

Table: ‘Orders’

Table: ‘Customers’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3

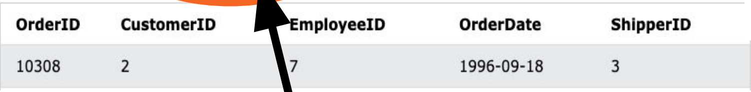
CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

```
SELECT *
FROM Orders
INNER JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```

Note: if we did this, CustomerID would appear twice in the resulting joined table, so it would get relabeled as e.g. Orders.CustomerID and Customers.CustomerID

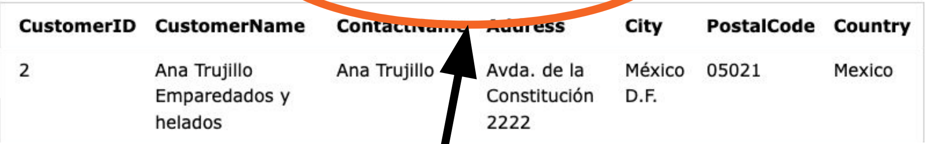
INNER JOIN

Table: ‘Orders’



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3

Table: ‘Customers’

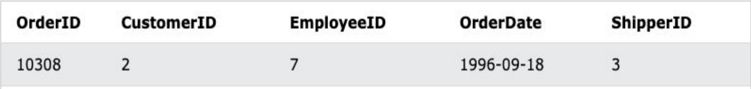


CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

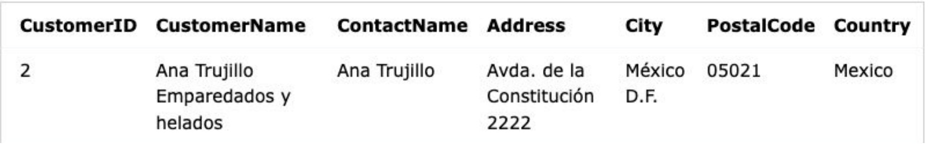
```
SELECT *
FROM Orders
INNER JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```

Note: if we did this, CustomerID would appear twice in the resulting joined table, so it would get relabeled as e.g. Orders.CustomerID and Customers.CustomerID

INNER JOIN



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

OrderID	CustomerName
10308	Ana Trujillo Emparedados y helados

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON
Orders.CustomerID = Customers.CustomerID;
```

https://www.w3schools.com/sql/sql_join_inner.asp

Extending your data

- INNER JOIN step can result in lots of rows
  (our first example just had one, Ana)
- You can also use WHERE to further filter after doing an INNER JOIN
- We used INNER JOIN because we wanted to know what customers showed up in BOTH tables

Admin

- Student hours posted on Canvas / Ed
- HW1 is posted & due Thursday 08/31

1. Cap your marker
2. Return marker & whiteboards
to each of their bins
3. Throw your tissues in the trash