Mani Sarkar
github: neomatrix369
kaggle: @neomatrix369 | twitter: @theNeomatrix369
blogs: https://medium.com/@neomatrix369

Studying the limitations of statistical measurements

—

my musings and perspectives on stats, while on my ML/Data Science journey

26th March 2022

hashtag:
#kaggledays
#delhiNCR

About me

Mani Sarkar

Senior Software, Data, ML Engineer

Java / JVM

Cloud / Infra / DevOps

Polyglot developer

Code quality, testing, performance, DevOps, deep affinity for AI/ML/DL, NN...

LJC, Devoxx, developer communities

Strengthening teams and helping them *accelerate*

JCP member, F/OSS projects: @adoptopenjdk @graalvm @truffleruby

Java Champion, Oracle Groundbreaker Ambassador, Software Crafter, Blogger, Speaker

More about me

@theNeomatrix369

Agenda

About the talk

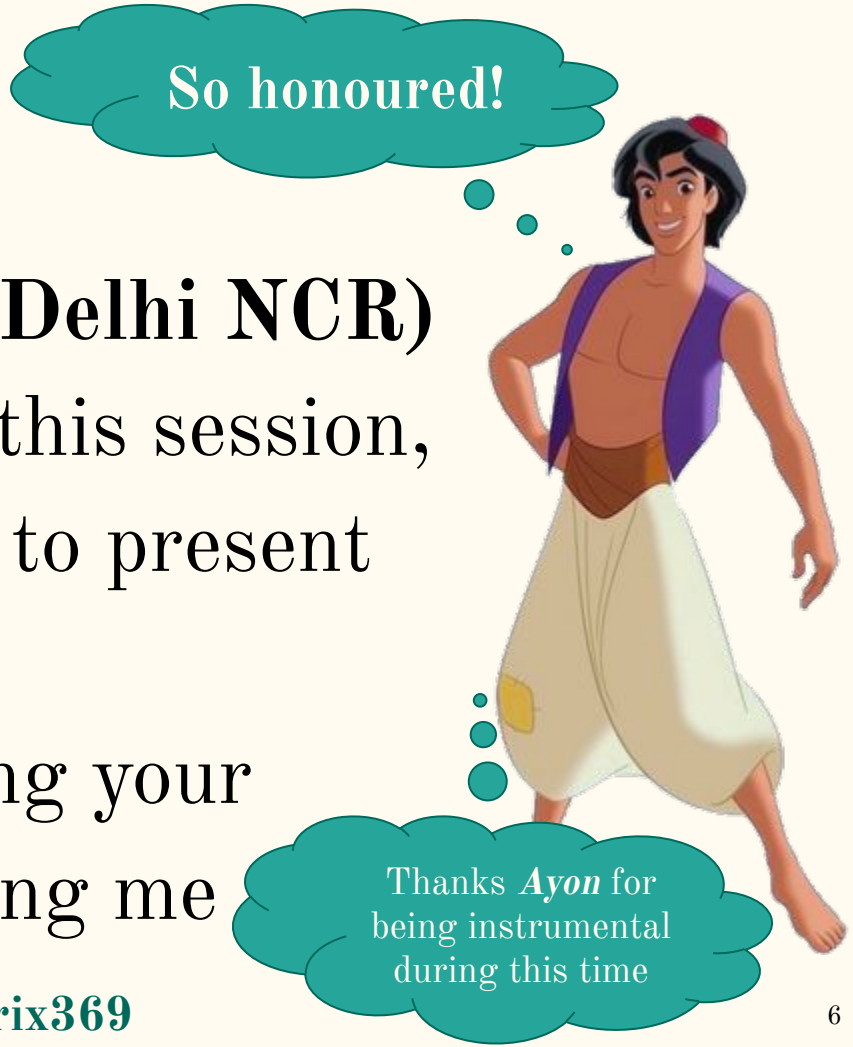
- *Introduction*
- *Explanations via notebook*
- *Additional questions to ask*
- *Ideas/insights*
- *Final questions*
- *Summary, Closing, Resources, Thanks*
- *Q&A*
- ***Appendix** section: few additional resources*

Presentation slides: *live*

https://bit.ly/studying-stats-limits

Thank You!

- **Kaggle Days MeetUp (Delhi NCR)** & **team**, for organising this session, and giving me a chance to present at this forum
- And to "**you**", for sparing your valuable time and trusting me



Disclaimer

- ***YMMV***
- Might be untested, and/or have **inaccuracies**
- Sharing our **learnings** over the past years
- Gathered ideas from **different experiences**
- We are making **inquiries (questions)** and **not claiming** anything yet
- We will be **playing with ideas** and go away **thinking about** them further
- **Sharing ideas** and **experiences**

Citation

The respective authors and creators are, and remain the true owners of the images and other artifacts used in this presentation.

Thank you for your creations!

Introduction

How did this start?

Tweet

We won't cover the topics mentioned here. But do take a glance at them.

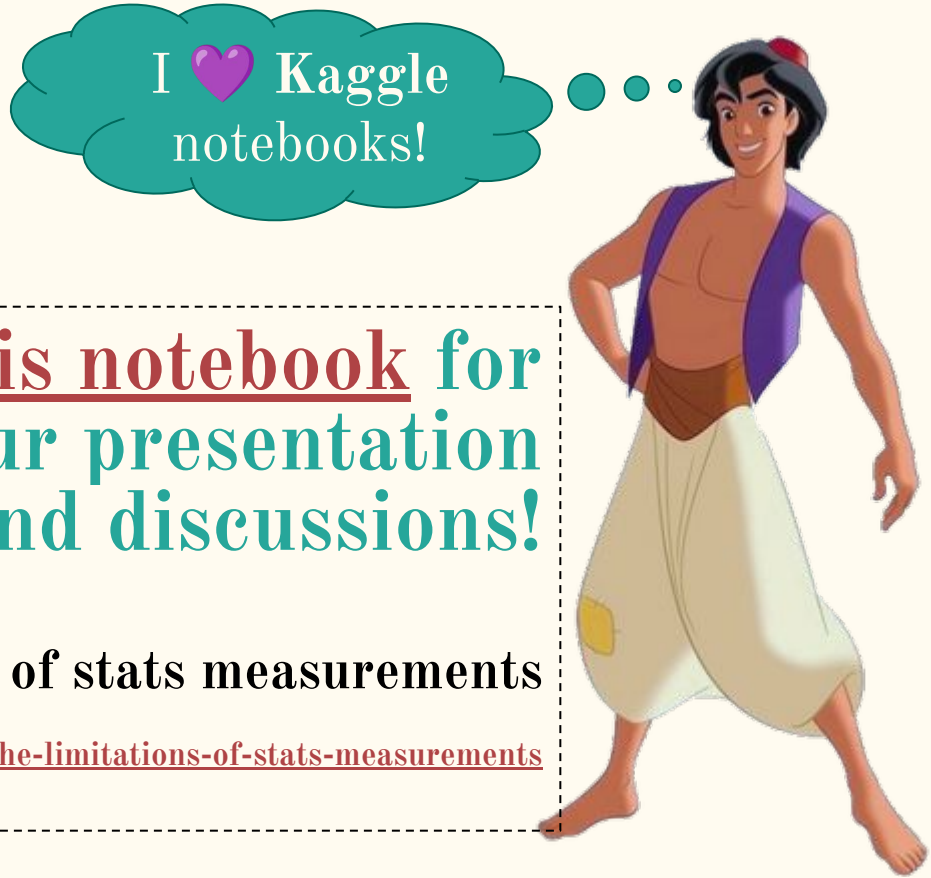
Around 2020/21 I came across these articles

- The trinity of errors in financial models: An introductory analysis using TensorFlow Probability
- The trinity of errors in applying confidence intervals: An exploration using Statsmodels

Simple discussions

Start from a simpler perspective

Creation of the notebook



We will be using this notebook for different parts of our presentation and discussions!

Studying the limitations of stats measurements

https://www.kaggle.com/code/neomatrix369/studying-the-limitations-of-stats-measurements

Let’s delve into the subject

First thing that came to my mind: *Correlation coefficient*

A **correlation coefficient** is a numerical measure of some type of correlation, meaning a statistical relationship between two variables.[a] The variables may be two columns of a given data set of observations, often called a sample, or two components of a multivariate random variable with a known distribution.[*citation needed*]

Wikipedia

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

How to calculate Correlation Coefficient?

- *In python: https://www.statology.org/correlation-in-python/*
- *Deep dive: https://www.wallstreetmojo.com/correlation-coefficient-formula/*

Argument / discussion

Pass two distributions to `corrcoef()`, and you get back a ***single value***

Kind of like a ***one-way data compression*** function

```
correlation_coef = np.corrcoef(distribution1, distribution2)
correlation_coef[0, 1]
0.24639418228457885
```

Issues with ***single*** values

Pairs of disparate distributions in theory can have ***near similar correlation coefficient***, but is that true about their correlation?

Terse / too compact, ***less insightful***

Sort of ***black box***?

Cannot easily trace back to distribution. ***No history*** of the steps?

Loss of information?

How to throw more light on the end-result(s)?

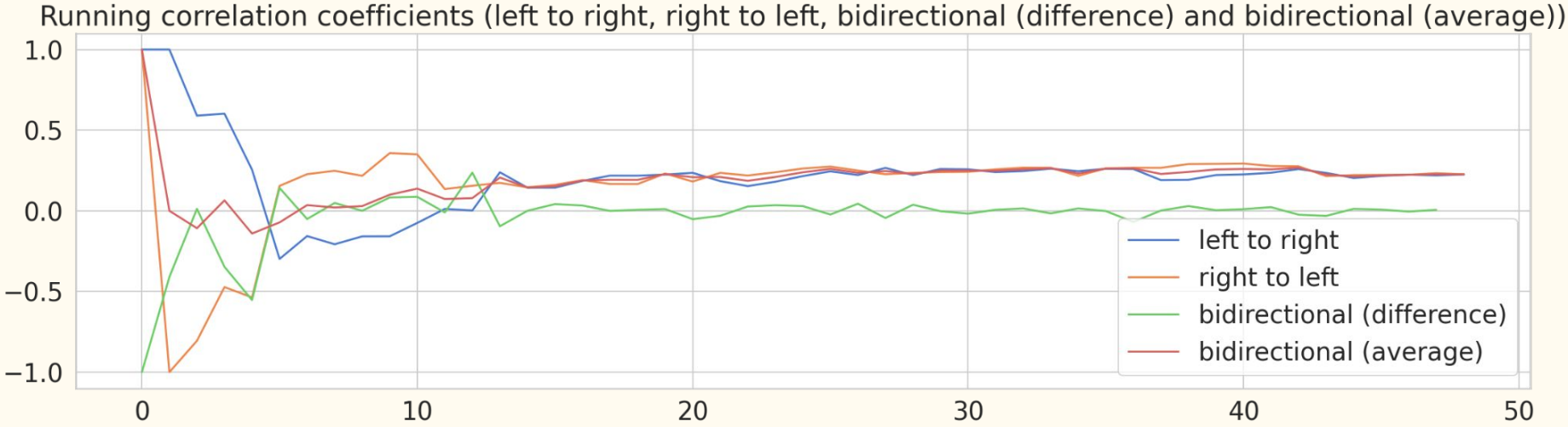
Coming up with two new ways to measure and plot correlation coefficients (experimental in nature):

- Running correlation coefficient
- Moving window coefficient

How to throw more light on the end-result(s)?

Similar to Simple moving average

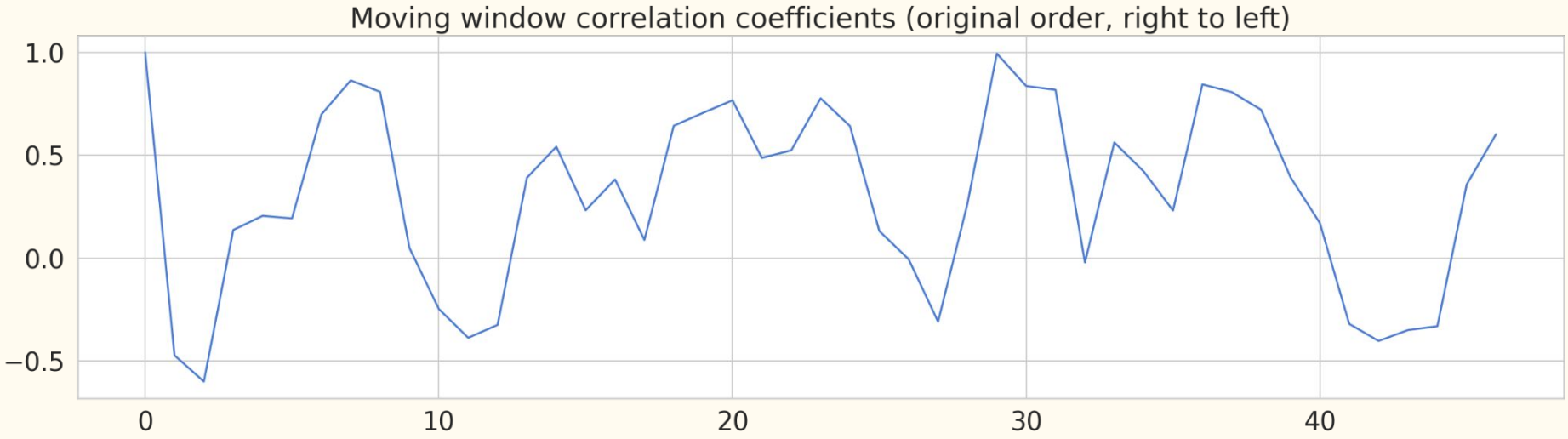
Running correlation coefficient



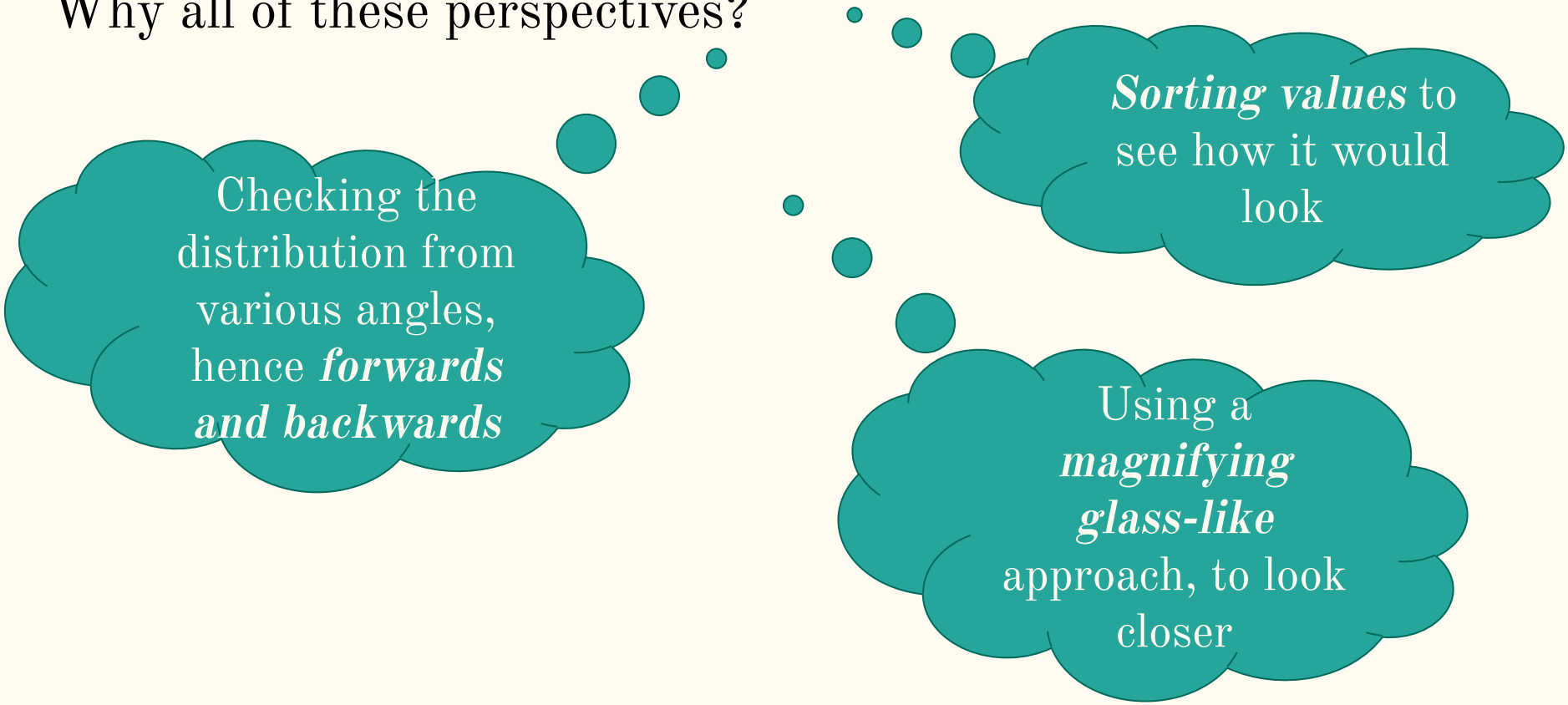
How to throw more light on the end-result(s)?

Simple moving average, applying windows or segments

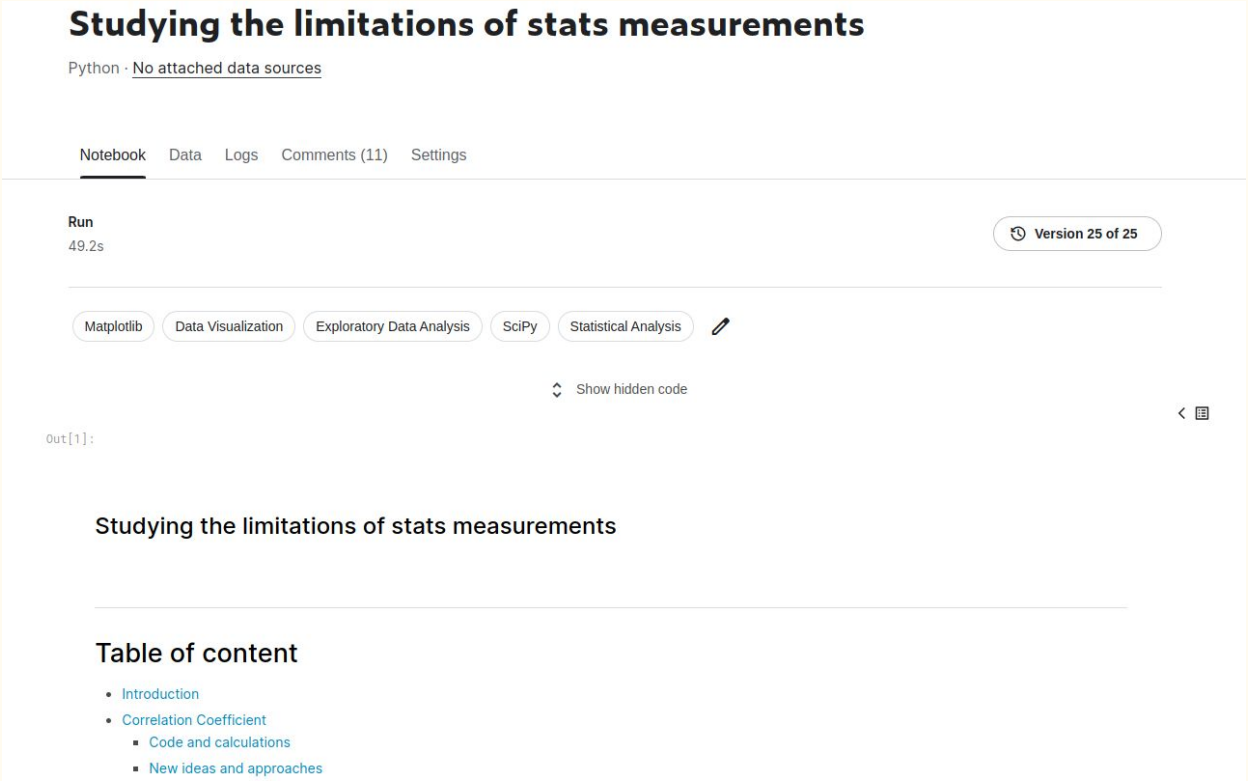
Moving window coefficient



Why all of these perspectives?



Checkout the rest of the notebook



Few commentaries from notebook

[snipped]...We can also say that some coefficients may contradict in it's polarity (or direction) due to the nature of the distribution across the two variables. But this is also something that the run-of-the-mill correlation coefficient function does not capture (or cannot capture).

Again, the above shows how the distribution contain a portion of the other polarities of correlations as well, not just the final projected correlation coefficient. Meaning positive, negative and no correlation can be part and parcel of the distributions. So it's not entirely only positively or negatively correlated, it's many a times a mix bag. And now if we compare the above with the singular value of the correlation coefficient derived out of the traditional toolbox, we will see that the reality is different.

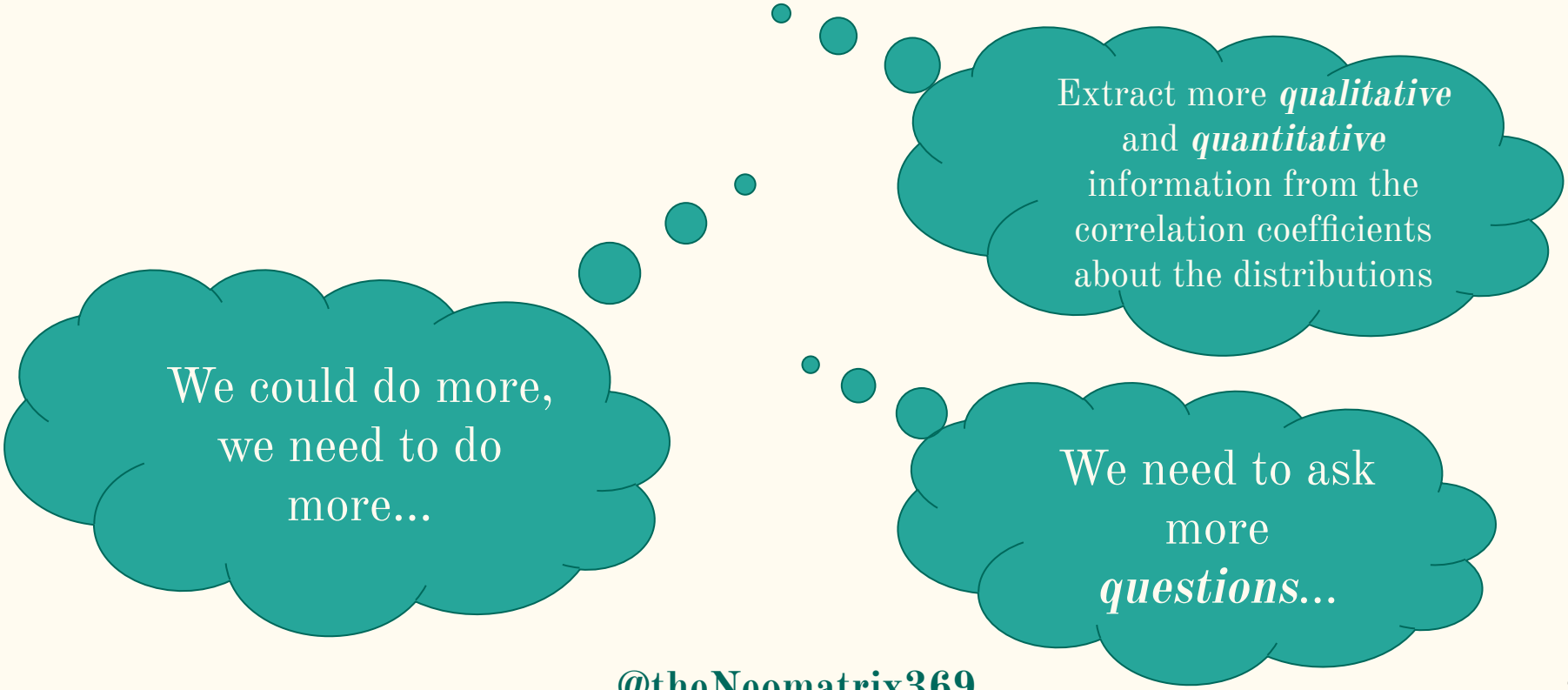
[snipped]…it's a mixed bag, and not one sided like the scalar "correlation coefficient" values try to portray. ***We can also say that some coefficients may contradict in it's polarity (or direction) due to the nature of the distribution across the two variables. Here contradictions are either natural or not taken note of, either of which has an impact on the final results and decisions made on such facts (unknown knowns). And these are also things that the run-of-the-mill "correlation coefficient" function(s) do not capture (or may not be able to capture).***

Few commentaries from notebook

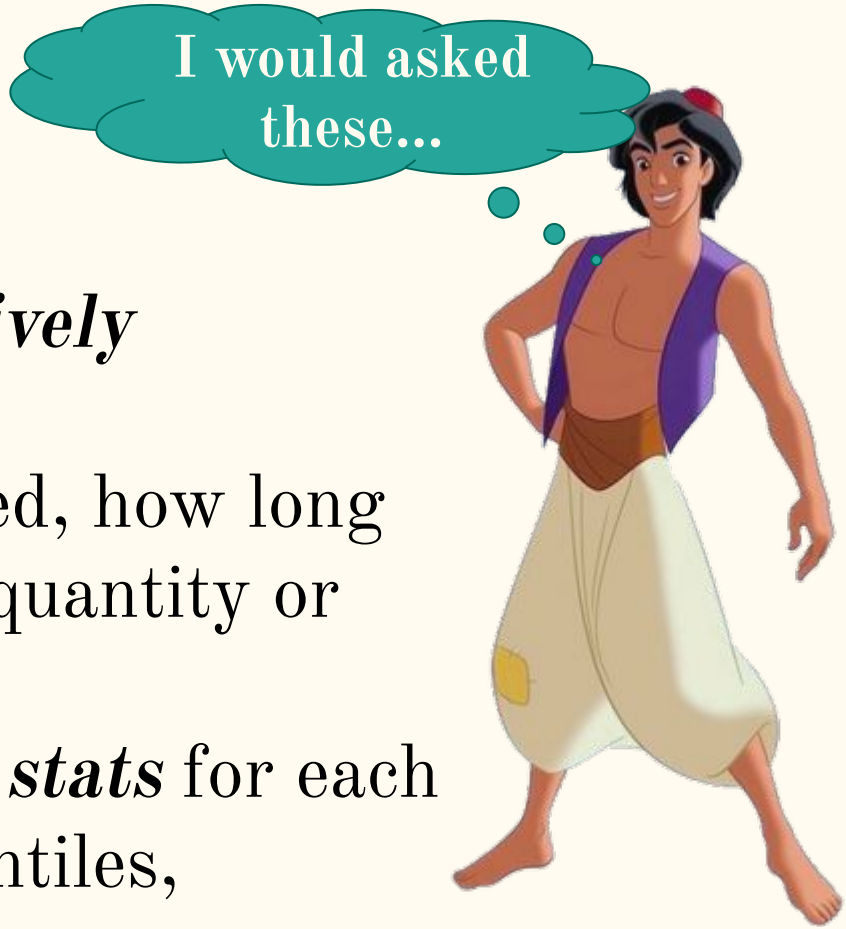
	Negatively correlated	Not correlated	Postively correlated
index			
count	13.0000	0.0000	15.0000
mean	-0.5299	NaN	0.6250
std	0.2693	NaN	0.3017
min	-0.9506	NaN	0.1666
5%	-0.9153	NaN	0.2107
10%	-0.8778	NaN	0.2451
20%	-0.8204	NaN	0.3132
25%	-0.8183	NaN	0.3353
30%	-0.6622	NaN	0.3655
40%	-0.5057	NaN	0.5860
50%	-0.4902	NaN	0.6899
60%	-0.4678	NaN	0.7204
70%	-0.3777	NaN	0.8756
75%	-0.3676	NaN	0.9196
80%	-0.3364	NaN	0.9375
90%	-0.2507	NaN	0.9700
95%	-0.1703	NaN	0.9843
max	-0.0740	NaN	1.0000
% out of the total	27.6596	0.0000	31.9149

*We can now see a breakdown of how much portion of the distribution makes up for positive, negative and no correlation types of correlations across the distribution (**% out of the total**). And within that we can also see the percentile values of the distribution values under each of the three bigger categories (these can be seen as the magnitude of change exhibited by one variable when the other changes while they are either positively or negatively correlated). We can see that the correlation coefficient relation between the distribution is only partially correct, ie. **~76%** of the times, while the **~24%** of the times it maybe the reverse (inversely correlated).* ***"Averaging gives some compactness, but it does not mean it maybe the right thing to do", because in many instances such a difference in value, can be a big difference, moreover it's an important aspect of the detail we maybe missing when we use older/traditional tools.*** *Note that with each run of this notebook you may see different values, not just **~76%** and **~24%**, as the distribution values are a result of some random function call via* `numpy` *. But the point remains that there are many a times significant differences between the two values (the correlation polarities), and even if there isn't, it is still worth knowing about them, which we fail to do so when using the old/traditional tools.*

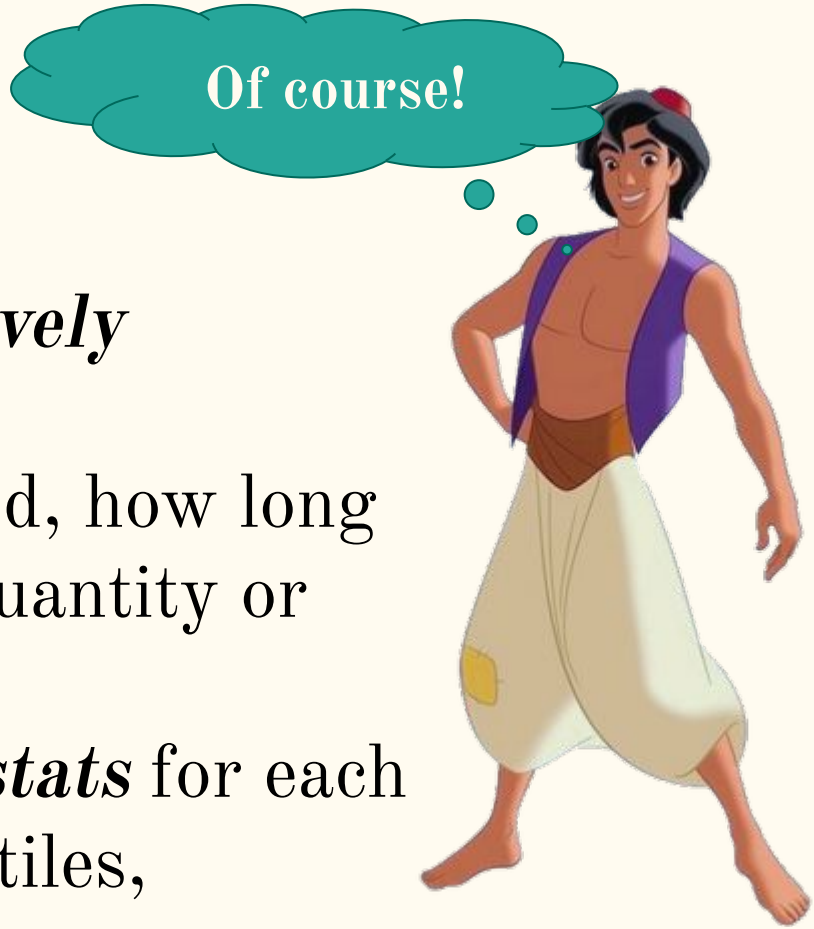
Are we complete? Have we resolved the limitation(s)?



Additional questions to ask



- How often are they ***positively*** correlated? (*PC1*)
- When ***positively*** correlated, how long do they stay like that (in quantity or time)? (*PC2*)
- What are the ***descriptive stats*** for each of these (min, max, percentiles, histograms)? (*PC3*)



- How often are they ***negatively*** correlated? (*NC1*)
- When ***negatively*** correlated, how long do they stay like that (in quantity or time)? (*NC2*)
- What are the ***descriptive stats*** for each of these (min, max, percentiles, histograms)? (*NC3*)

- How often are they ***not*** correlated? (*NoC1*)
- When ***not*** correlated, how long do they stay like that (in quantity or time)? (*NoC2*)
- (skipping the ***descriptive stats*** for this category)

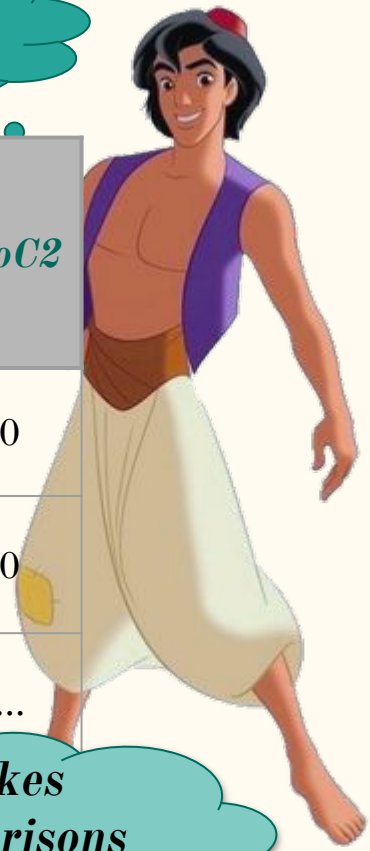
Gathering the quantities

Additional ***qualities*** derived from the *correlation coefficient* (features as quantities) between two distributions:

- *PC1*, *PC2*, *PC3*,
- *NC1*, *NC2*, *NC3*,
- *NoC1*, *NoC2*

Could this make it easier to make comparisons?

Comparisons made easy

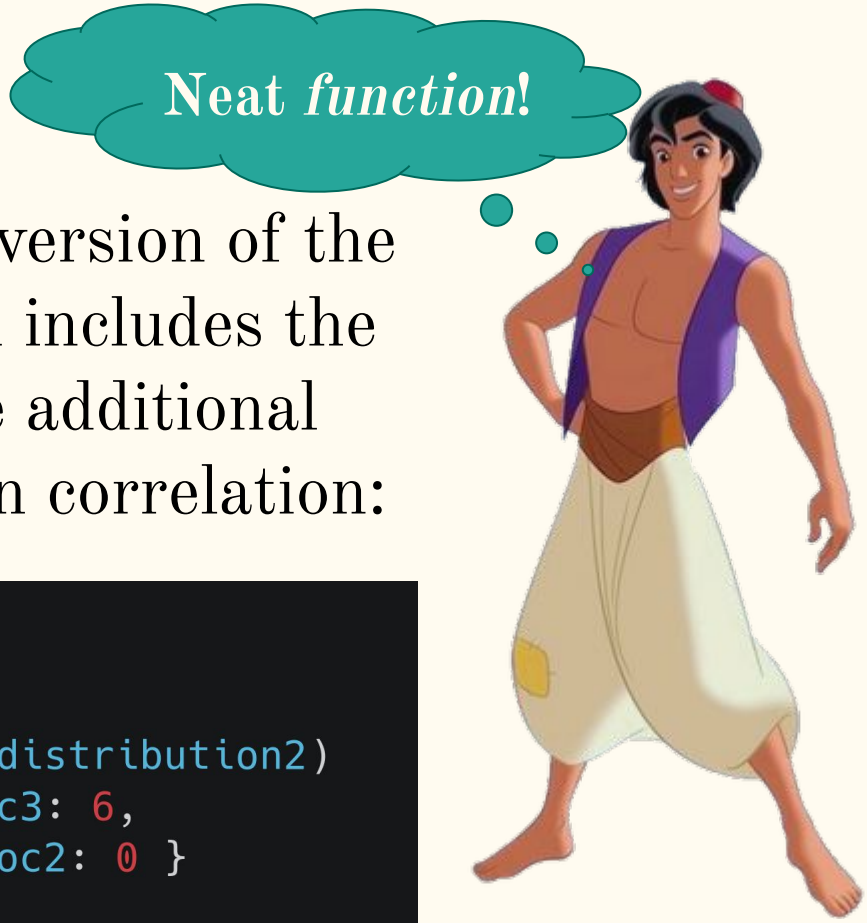


Comparing correlation coefficients between Distributions	*Corr. Coef.*	*PC1*	*PC2*	*PC3*	*NC1*	*NC2*	*NC3*	*NoC1*	*NoC2*
Normal v/s Random	0.24	1	5	6	8	5	6	0	0
Random v/s Pareto	0.26	3	2	7	9	2	7	0	0
Normal v/s Pareto	…	…	…	…	…	…	…	…	…

Do not take these values literally, it's an example.

Makes comparisons more objective

@theNeomatrix369



We could implement our own version of the `corrcoeff()` function, which includes the standard result along with the additional ***qualities*** about the distribution correlation:

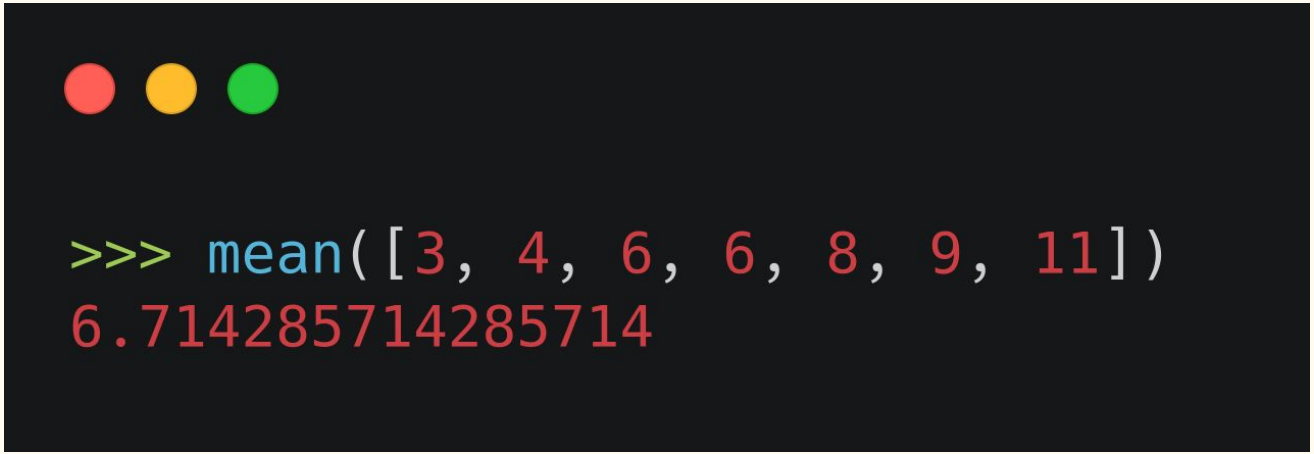
```
>>> better_corr_coef(distribution1, distribution2)
{ corr_coef: 0.24, pc1: 1, pc2: 5, pc3: 6,
  nc1: 8, nc2: 5, nc3: 6, noc1: 0, noc2: 0 }
```

Ideas/insights

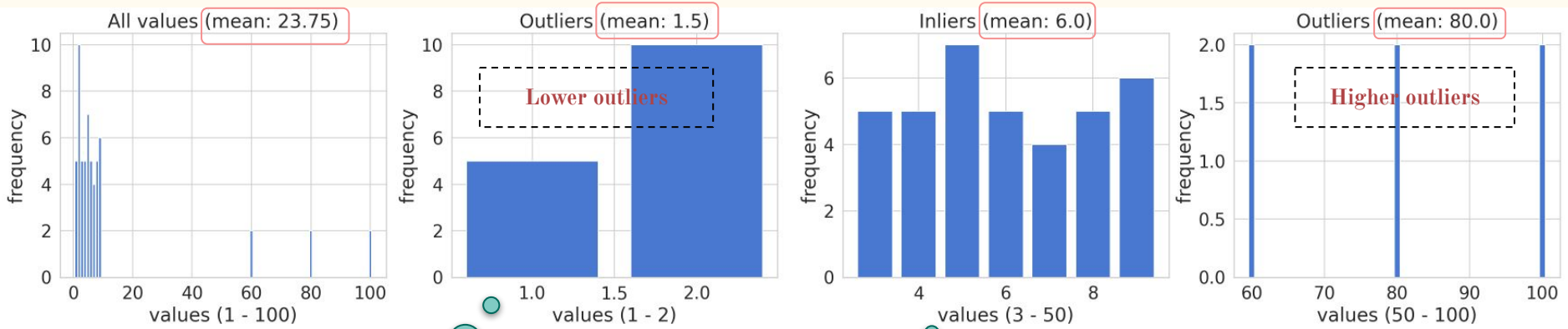
Ideas/insights: not yet mentioned in this Notebook

- Arithmetic **mean** or **average:** `mean()` returns a ***single value***
- ***Could outliers dampen*** the end result?
- What if we ***accounted for outliers*** in our end result? (Idea 1)
- Is there room for something like ***hierarchical "mean"***? How do we do that? (Idea 2)

Arithmetic **mean** or **average:** `mean(values)` returns a ***single value***



Arithmetic **mean** or **average:** ***Could outliers dampen*** the end result?

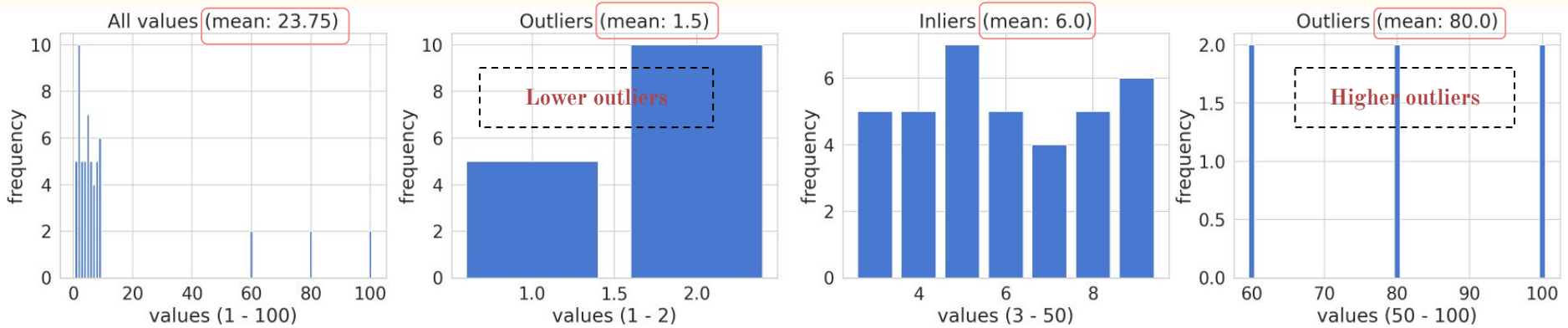


We can see how the <u>inlier mean</u> is quite different from the <u>all values mean</u> - why is this?

You can see the differences in the <u>mean value</u> above - is it outliers causing it?

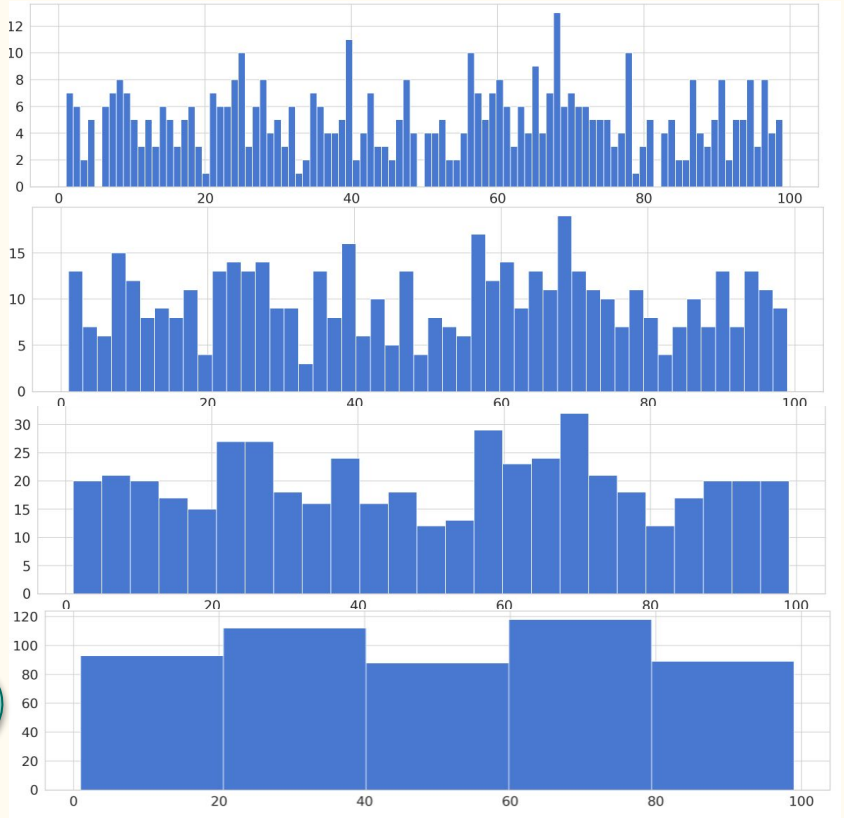
@theNeomatrix369

Arithmetic **mean** or **average:** What if we ***accounted for outliers*** in our end result?




```
>>> better_mean(values, inlier_by_value_limits = (3, 50))
{ all_values: 23.75, lower_outliers: 1.5,
  inliers: 6.0, higher_outliers: 80.0 }
```

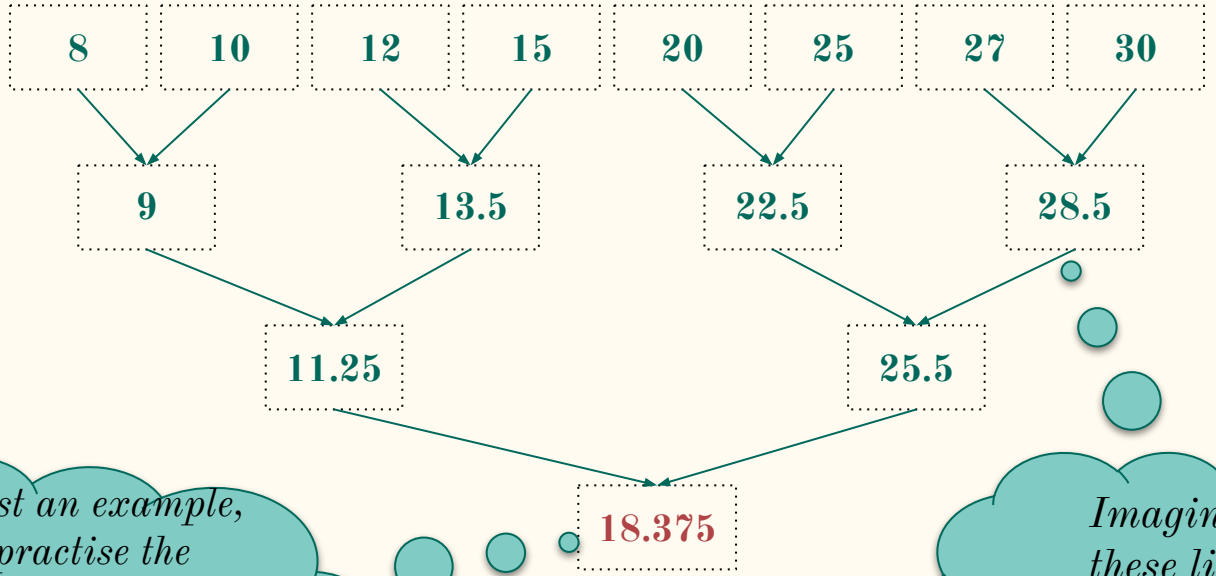
Arithmetic **mean** or **average:** Is there room for something like ***hierarchical "mean"***?



Grouping, merging and gathering values in stages

We can do similar to ***histogram binning*** *or using the* `pandas.cut()` *function.*

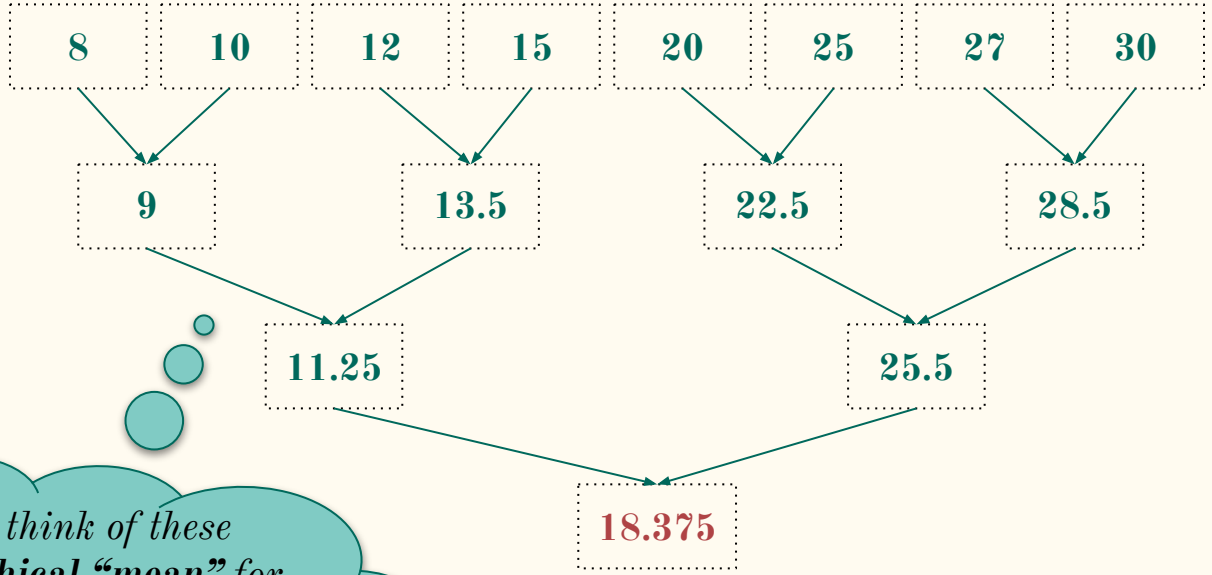
Arithmetic **mean** or **average:** Is there room for something like ***hierarchical "mean"***?



It's just an example, in practise the end-result can vary ***based on number of values, grouping and merging.***

Imagine returning these linked values with the `better_mean()` *function?*

Arithmetic **mean** or **average:** Is there room for something like ***hierarchical "mean"***?



Now think of these ***hierarchical "mean"*** *for each of these groups:* ***lower outliers***, ***inliers***, *and* ***higher outliers***.

I hope to put these new ideas in another notebook and share with everyone soon. And maybe that would make things clearer.

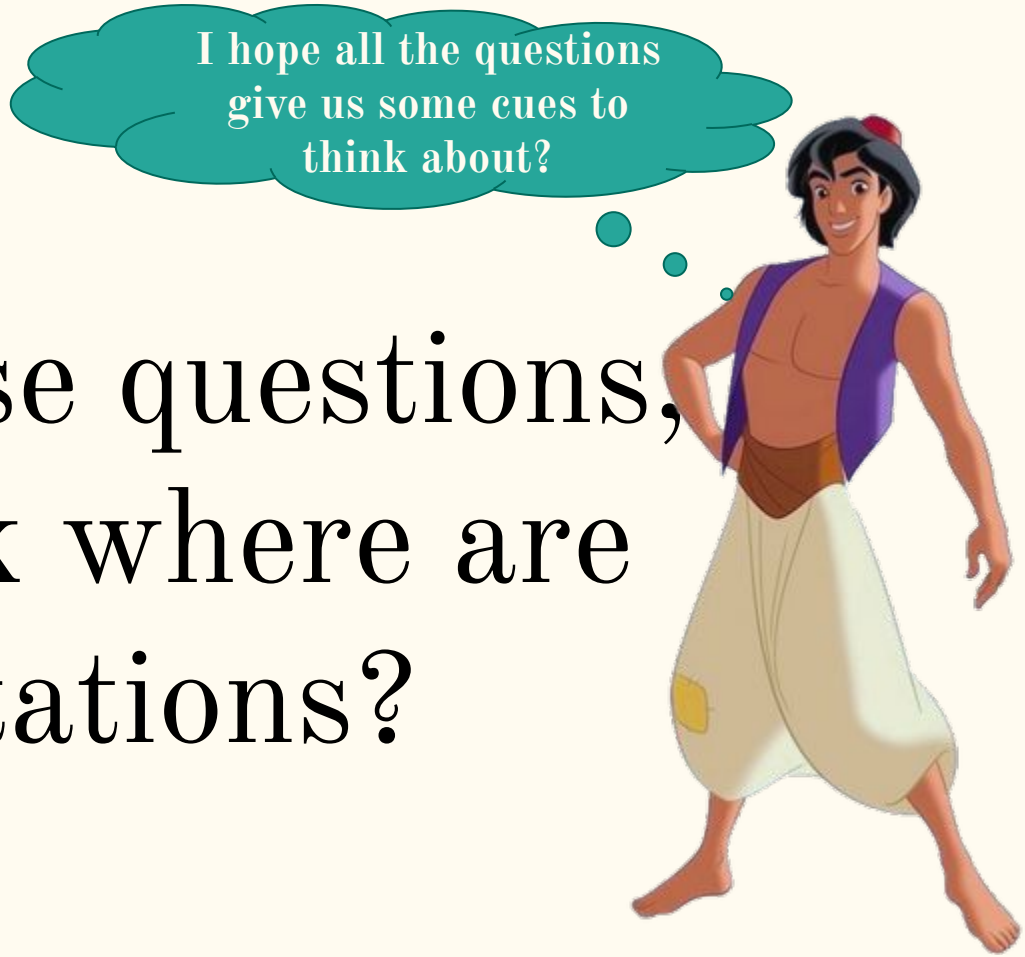
Finally let’s ask ourselves

What is missing in these ideas or observations, are the results or conclusions correct?

If they are ***correct***, then *"why"* - why are they correct?

If they are ***not correct***, then *"why not"* - why are they not correct?

How can we justify ***any of these positions***, based on the ideas and observations shown?



After all these questions, you may ask where are the limitations?

Summary

Summary

- What made me start with this kind of thinking? [slide]
- Into the subject: via notebook, examples [slide]
- Additional questions we should ask? [slide]
- New ideas and insights [slide]
- Final questions [slide]
- We have seen limitations and tried to overcome them
- We have tried to be curious and constructive throughout the process, haven't we? Hence please share ***constructive feedback***!

In Closing: we *cannot always win*!

We *cannot always* win

Really!

We may be in the minority

We could be wrong about these arguments (fully or partly)

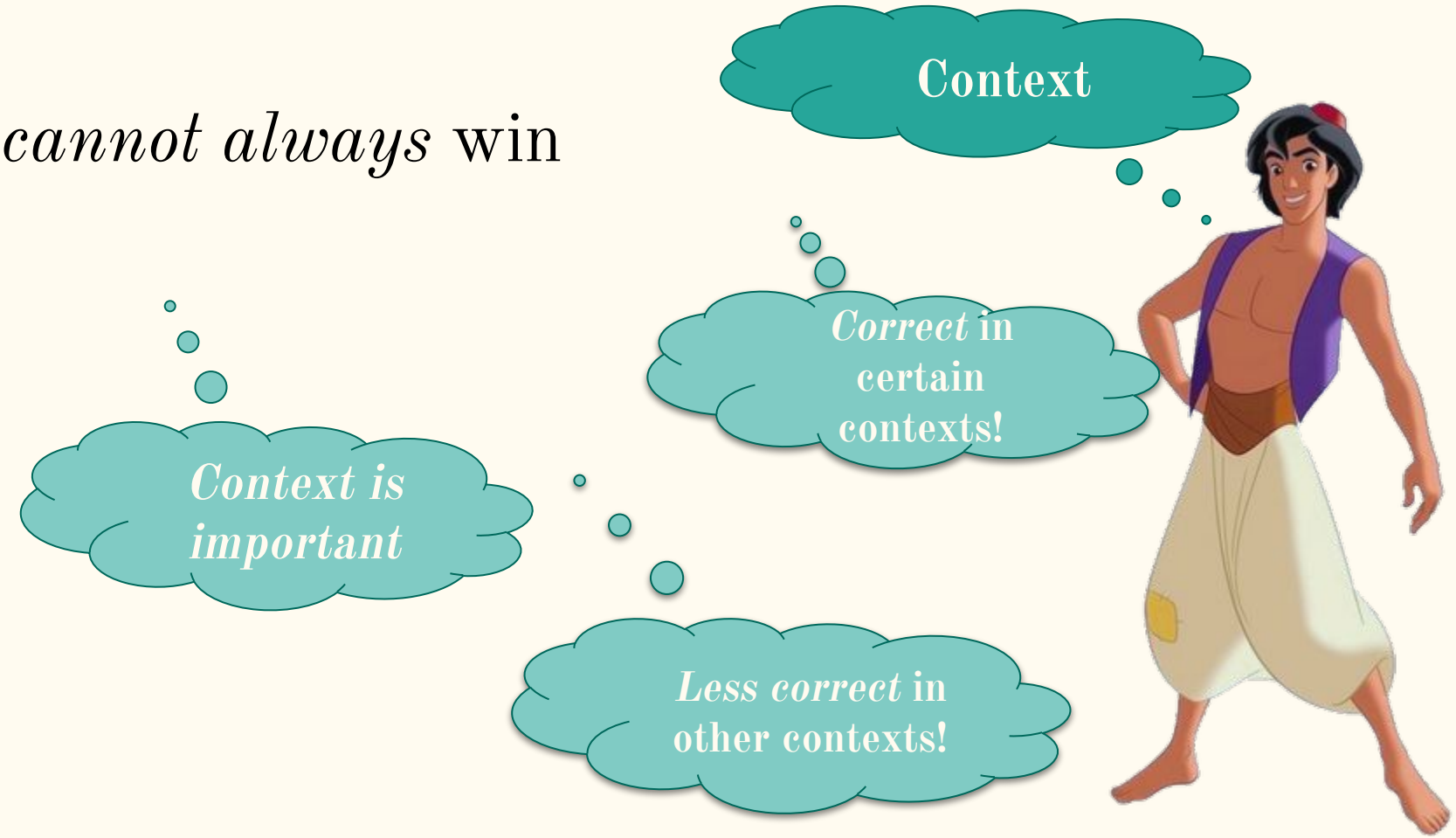
Argument(s) may need more ground

Are these conjectures?

Others may have other valid reasons!

@theNeomatrix369

We *cannot always* win



Resources

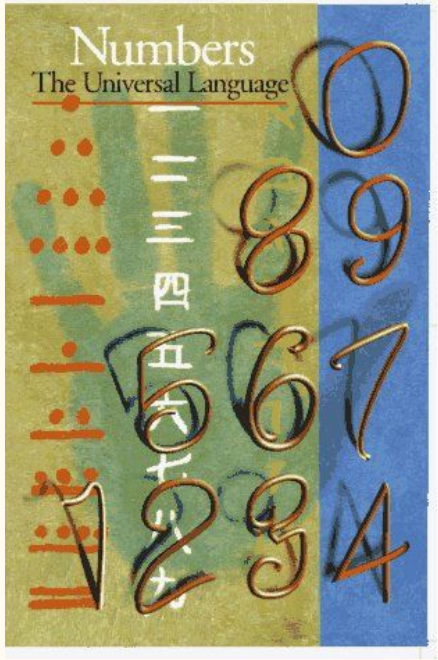
Resources to follow

- The "Studying the limitations of stats measurements" Notebook | The "Normalising a distribution" Notebook | Other Notebooks
- Awesome AI/ML/DL
- Maths, Stats, Probabilities on Awesome AI/ML/DL: [1][2][3]
- MadeWithML
- Virgilio | GitHub

People and resources to follow

- Ajit Jaokar, see post on ML & Stats thinking
- Vincent Granville (see new ebook)
- Thomas Nield (Essential Math for Data Science)
- Ian Ozsvald (newsletter: NotANumber)
- Abhishek Thakur (see AAAMLP book | YouTube)
- O'Reilly resources (and on Math)
- And many others...

Interesting book on Numbers



Numbers: The Universal Language (French: ***L'empire des nombres***, lit. 'The Empire of Numbers') is a 1996 illustrated monograph on numbers and their history. Written by the French historian of science Denis Guedj, and published in pocket format by Éditions Gallimard as the 300th volume in their "Découvertes" collection[1] (known as "Abrams Discoveries" in the United States, and "New Horizons" in the United Kingdom). The book was adapted into a documentary film of the same title in 2001.[2]. **Wikipedia**

Book available on Amazon

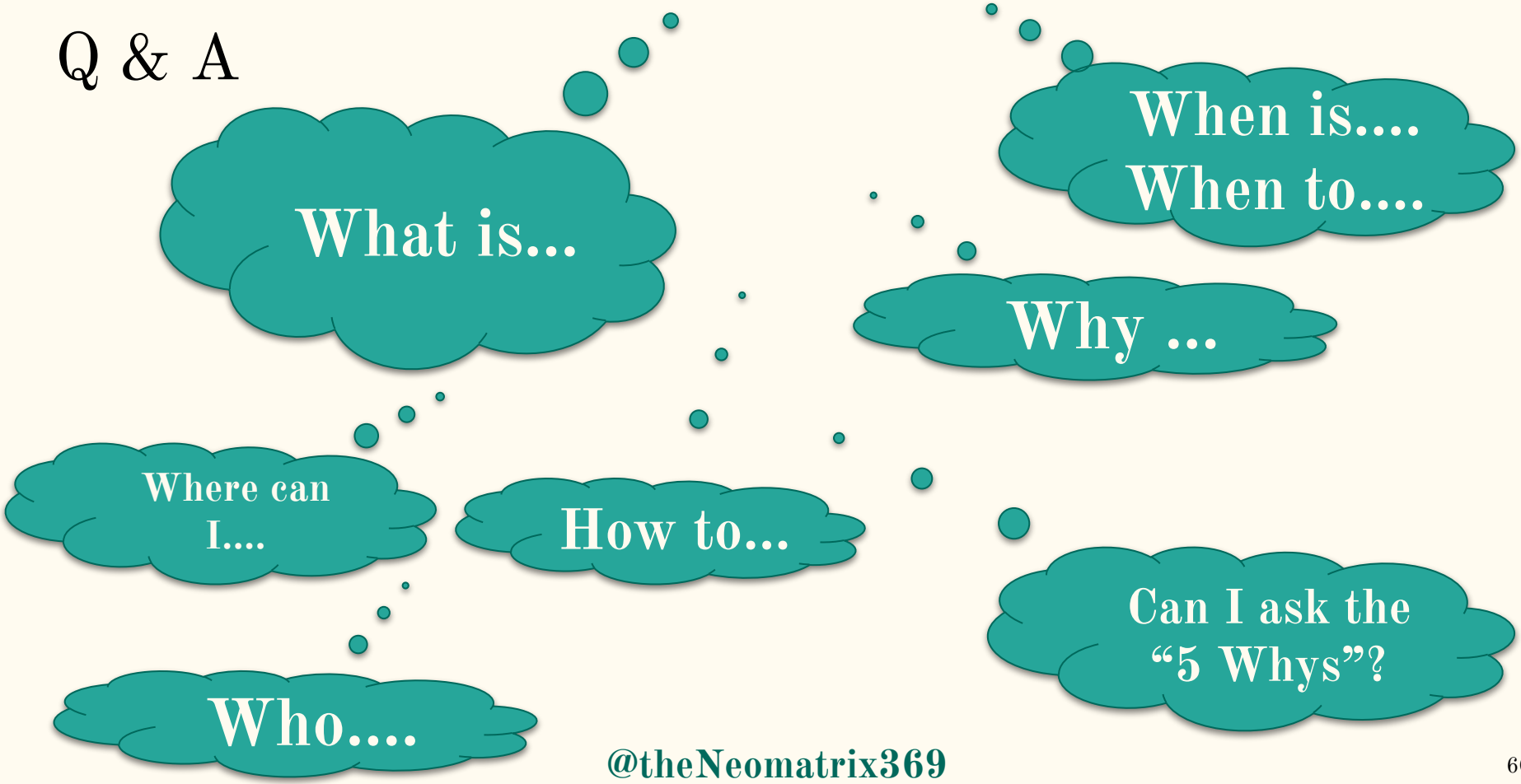
Thank You, again!

Thanks to organisers and audience

- **Kaggle Days MeetUp (Delhi NCR)** & **team**, for organising this session, and giving me a chance to present at this forum
- And to <u>you</u>, for sparing your valuable time and trusting in me

Q & A

Q & A


What is...
When is....
When to....
Why ...
Where can I....
How to...
Can I ask the "5 Whys"?
Who....

Contact and keep in touch

hashtag: #kaggledays #delhiNCR

- **twitter: @theNeomatrix369**
- **medium: https://medium.com/@neomatrix369**
- **github: https://github.com/neomatrix369/**
- **linkedin: https://www.linkedin.com/in/mani-sarkar/**
- **youtube: channel | playlists**
- **kaggle: https://kaggle.com/neomatrix369**
- **about me: https://neomatrix369.wordpress.com/about**

Appendix

Previous talks

- My most recent presentations can be found here
- My last in 2021 was Looking into Java ML/DL libraries: Tribuo and DeepNetts
- Tribuo: an introduction to a Java ML Library
- “nn” things every Java developer should know about AI/ML/DL
- Naturally, getting productive, my journey with Grakn and Graql
- Do we know our data as well as our tools?
- Java N.n: What to know? How to learn?
- Some of my other talks a can be found here and here (and others on Slideshare)

Non-mainstream sources

Explore & learn!

Wikipedia

Wolfram Mathematica

Wolfram Alpha

Find out what Mayans, Greeks, Egyptians, and other ancient civilisations knew about maths, stats and numbers!

These are some clues to start with...

Objectively research on YouTube: GaiaTV, History, National Geographic, and others

@theNeomatrix369

Other Resources

- No need to feel overwhelmed
- Kagglers, read this for inspiration
- Tackling Kaggle competitions interview by Ian Ozsvald [1][2]