



# Data Storytelling with Python and AI

Angelica Lo Duca


IIT-CNR, Italy

Author of Data Storytelling with Altair and AI (Manning)

# Outline

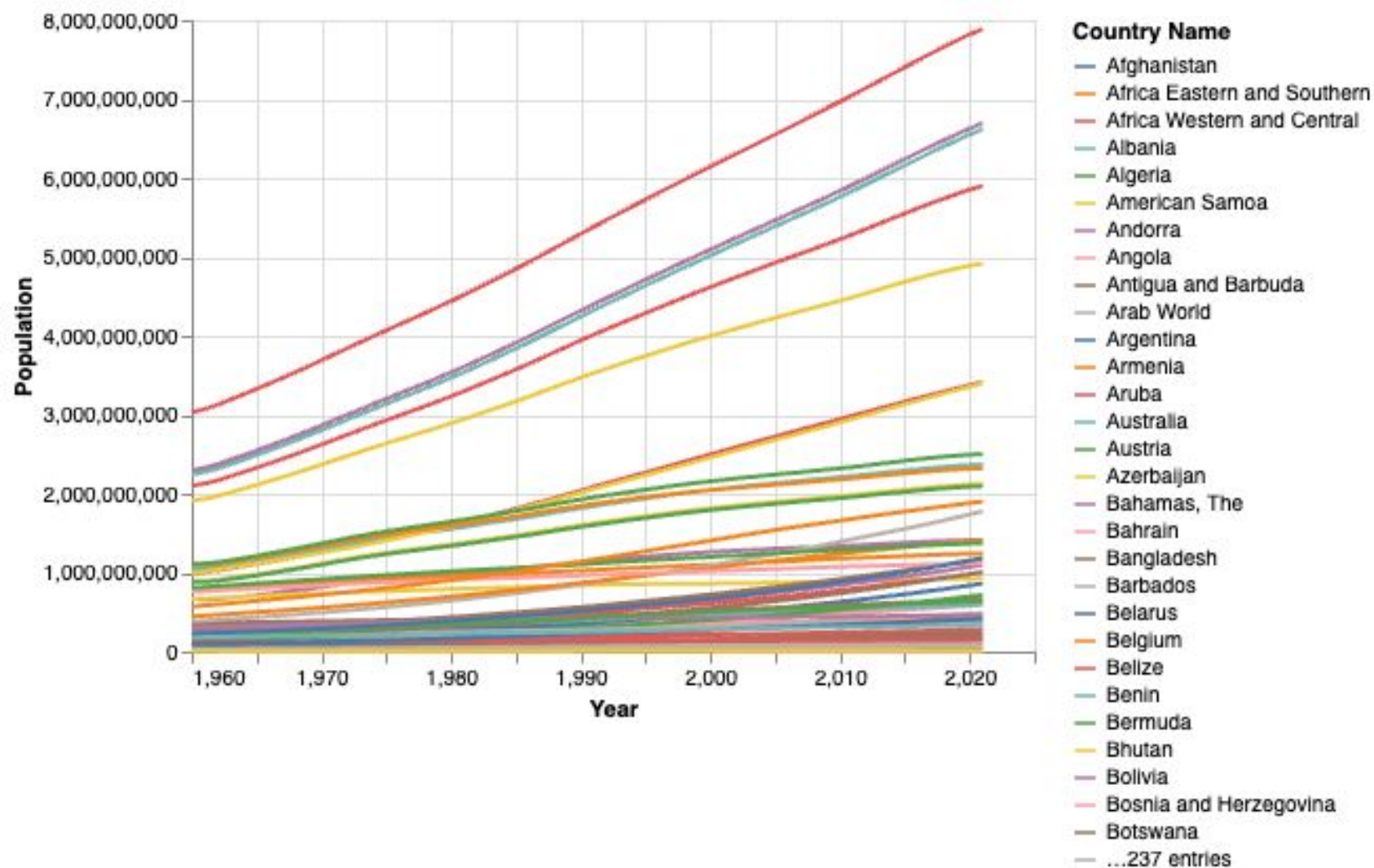
1. Introducing Python Altair
2. Introducing Data Storytelling and the S-DIKW Pyramid
3. Combining Data Storytelling and AI
4. A Practical Case Study

<https://github.com/alod83/hhai2025-tutorial>

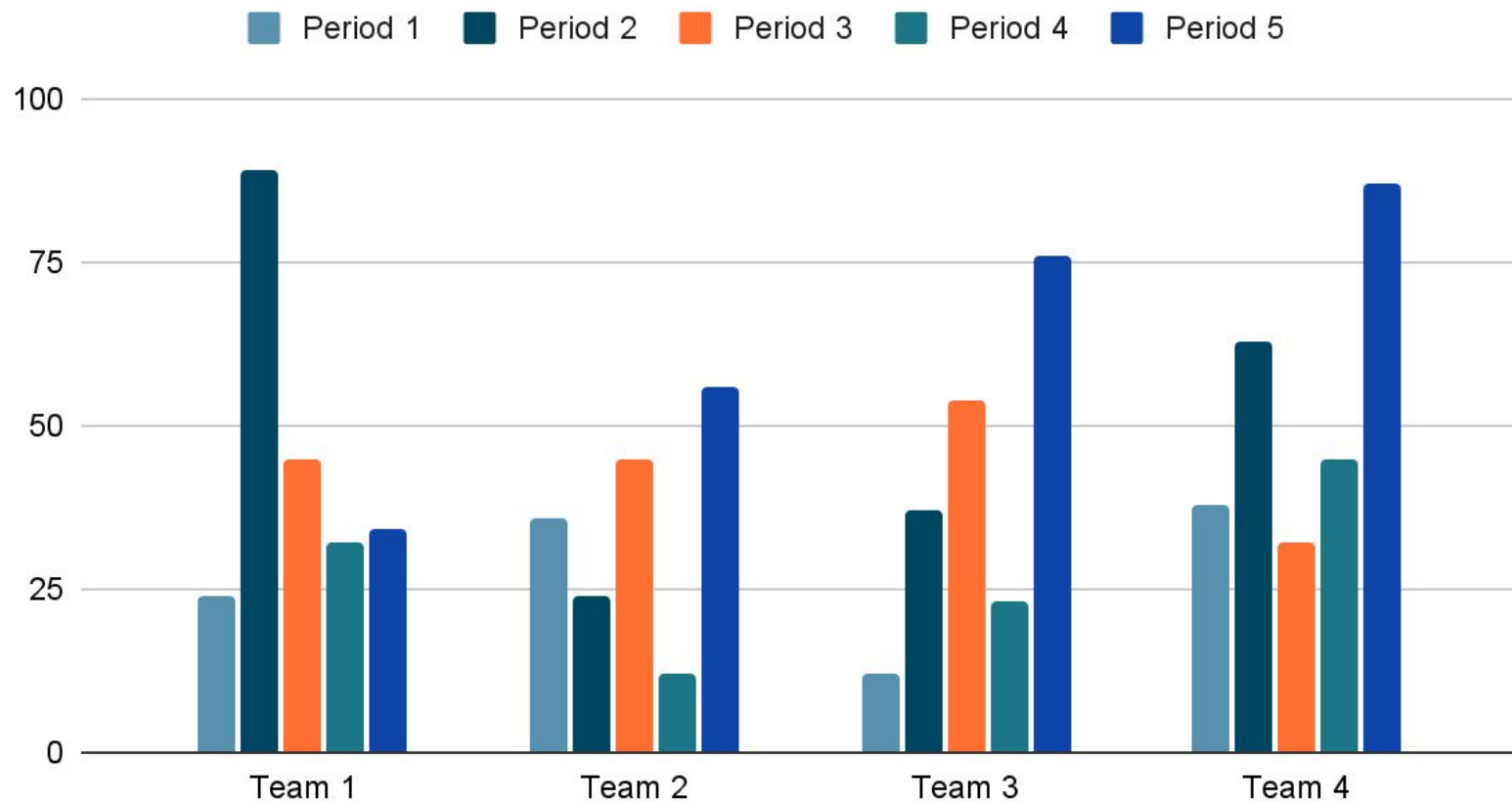
A person with dark hair is shown in profile, sitting at a desk and working on a laptop. The scene is dimly lit with a strong blue and purple color palette. In the background, a large monitor displays code or data. The walls and background are decorated with glowing circuit patterns and digital elements, creating a high-tech, futuristic atmosphere. The person's hands are on the laptop keyboard.

# Module 1

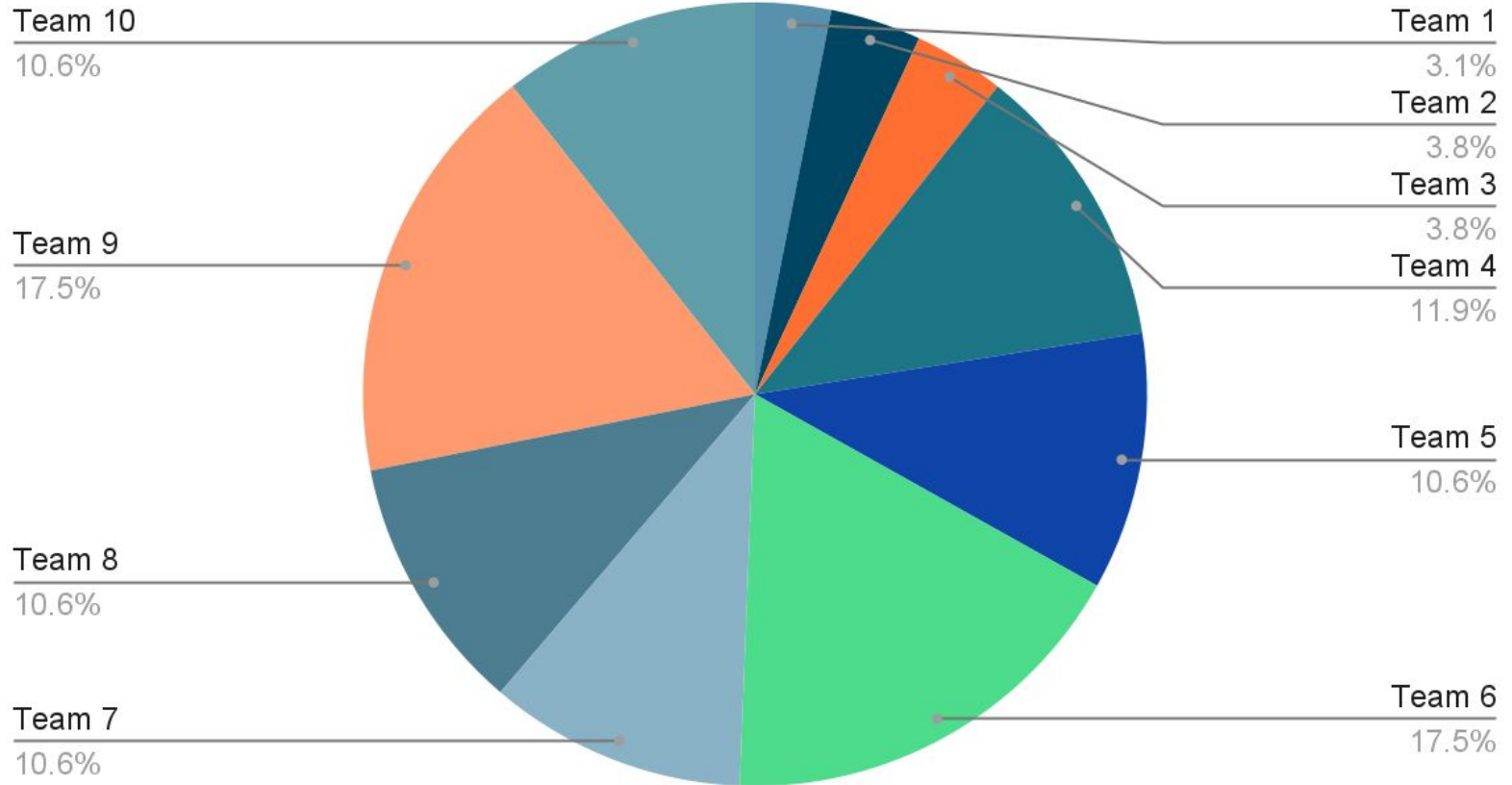
## Introducing Python **Altair**



## Points scored

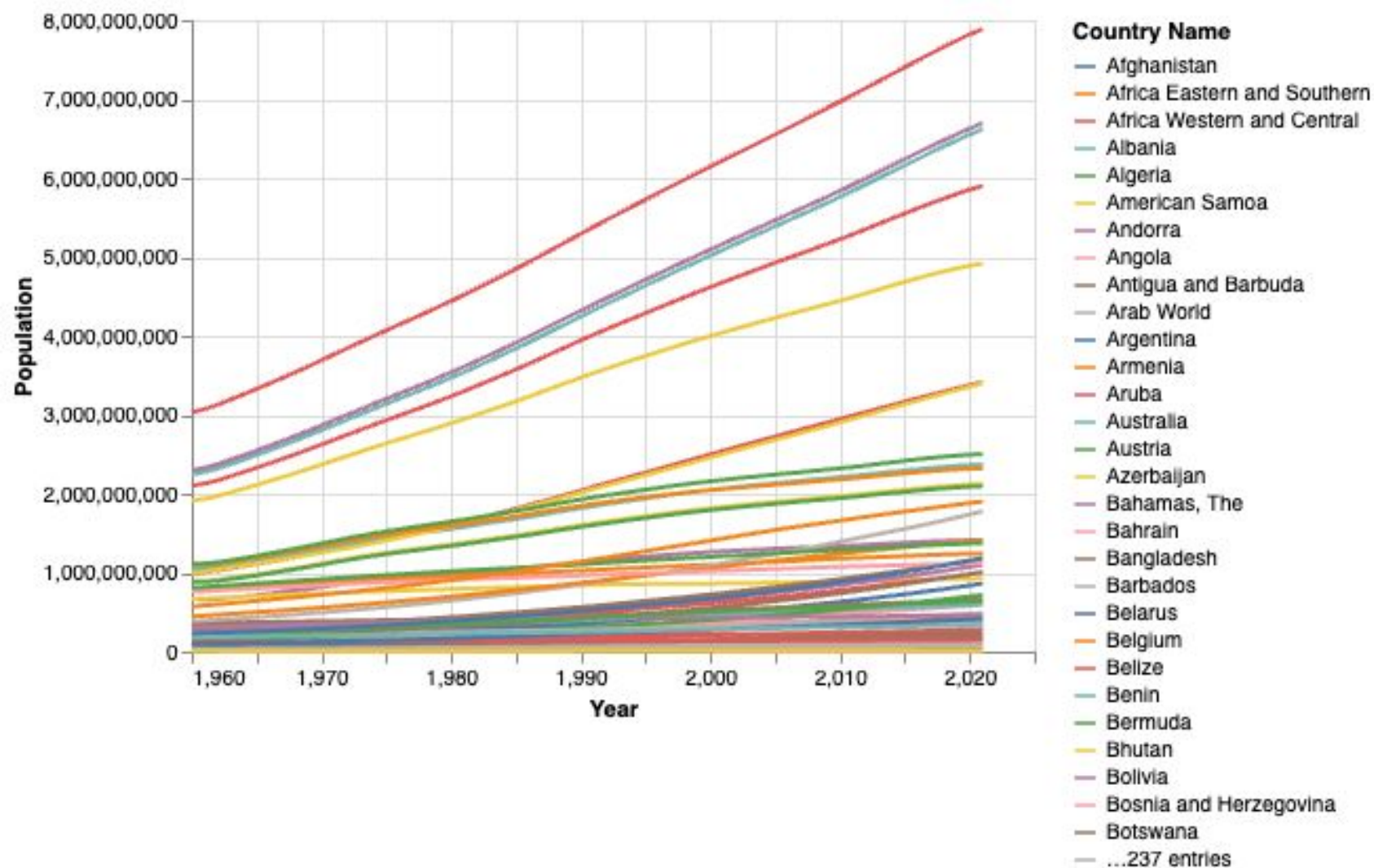


# Points scored

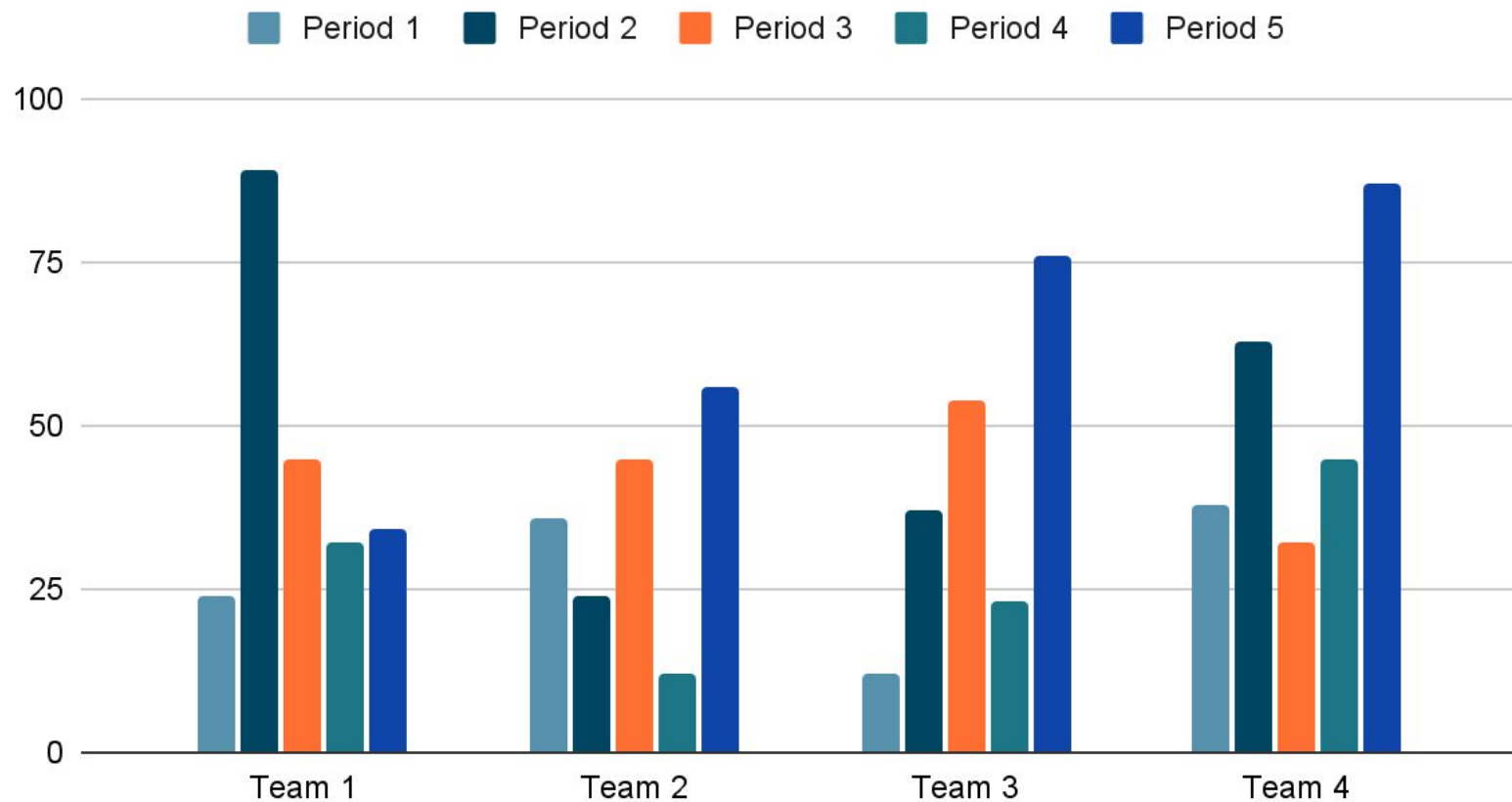


Too much information means no information.

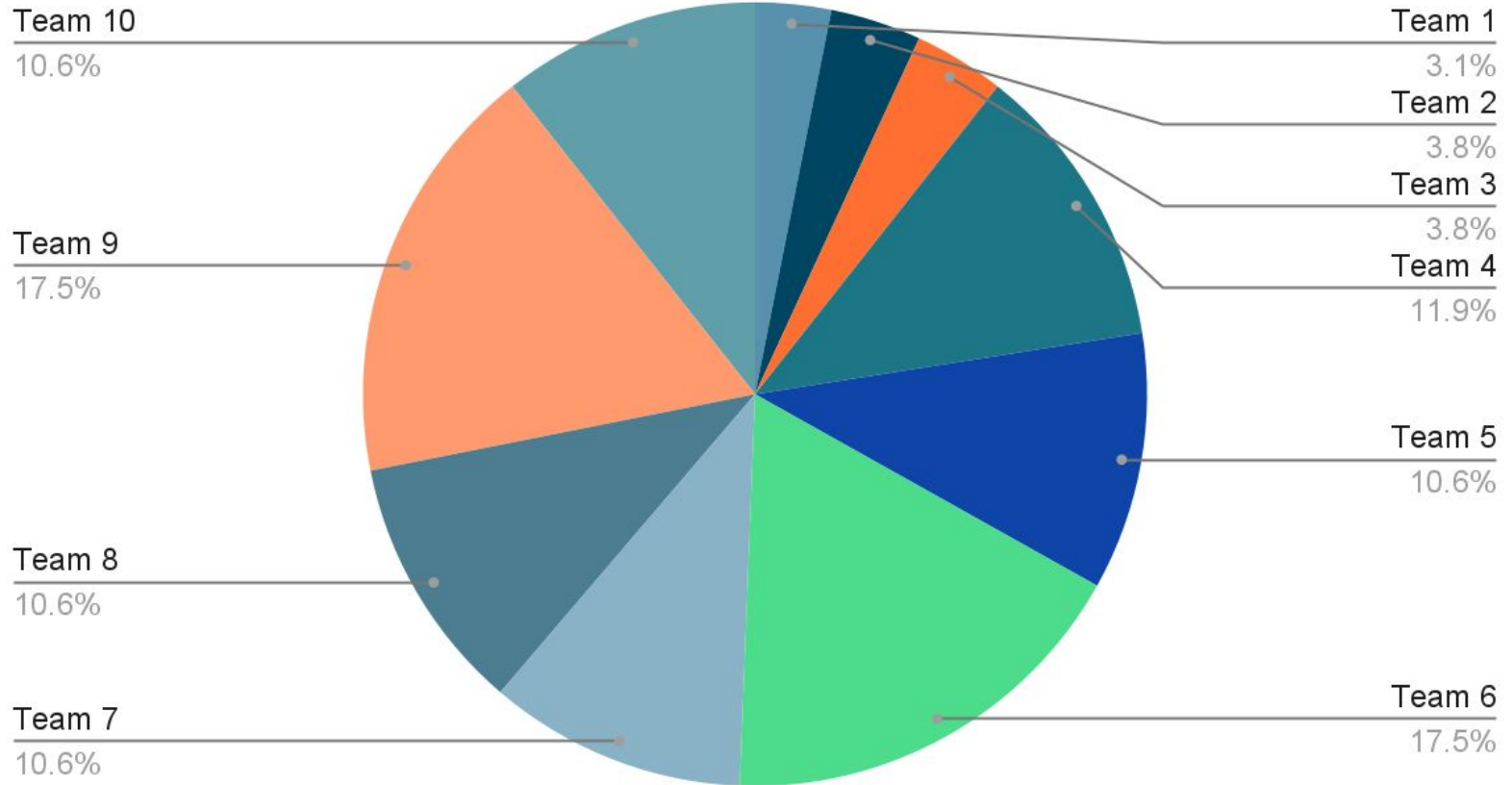




## Points scored



# Points scored



You must do the **dirty work**, not your audience.  
The audience should be able to read your chart **quickly**.

You must **spend time** to build your chart.  
Your audience should get the point of your chart **quickly**.

If I had more time, I would have written  
you a shorter letter. (Blaise Pascal)  
Brevity takes time. (Matthew Dicks)

Making the dirty work in your chart means **decluttering** your chart.

Declutter a chart  
using  
the Focus rule



# Focus Rule

**F**ilter Out Non-Essentials. Remove unnecessary data, elements, or decorations (like excessive gridlines, 3D effects, or drop shadows) that do not contribute to the message of the chart.

**O**rganize for Clarity. Arrange the chart elements (titles, legends, labels) in a logical and readable way. Use consistent scales, sorting, and grouping to enhance understanding.

**C**hoose the Right Chart Type. Select the chart type that best matches the data and message you want to convey. Avoid flashy or complex visuals if a simple bar or line chart will do.



# Focus Rule

**U**se Visual Hierarchy. Highlight the most important data using color, size, or position to draw the viewer's attention. De-emphasize less important details.

**S**implify Labels and Legends. Keep axis labels, titles, and legends concise and clear. Use plain language and avoid redundant or overly technical terms.



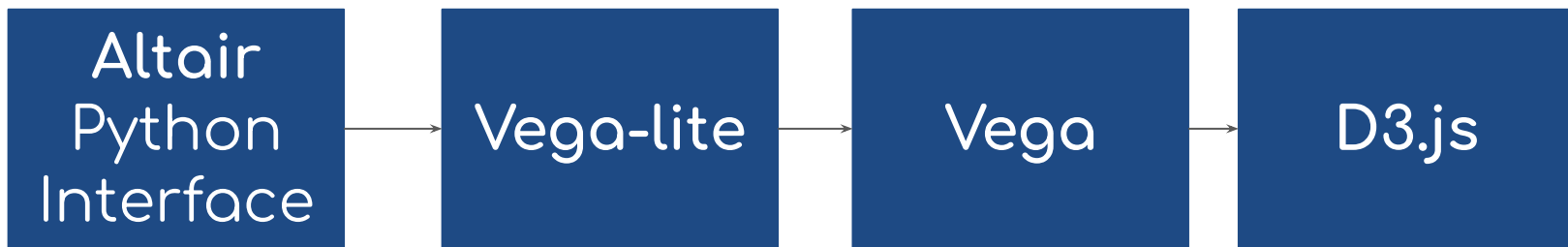
Python Altair  
makes applying  
the FOCUS rule  
easy.

# Altair

The Vega-Altair library (Altair, for short) is a declarative Python library for statistical visualization based on the Vega and Vega-Lite visualization grammars.

**Declarative libraries** specify what we want to see in a chart. We can specify the data and the type of visualization we want, and the library creates the visualization for us automatically.

**Imperative libraries** focus on building a visualization manually, for example specifying the desired axis, size, legend, and labels (e.g. Matplotlib)



# Altair Parameters

**Marks:** define the type of chart we want to build (e.g. bar chart, line chart, ...)

**Encodings:** mapping of visual properties (channels) to data columns in the DataFrame

Visual properties include axes, colors, size, ...

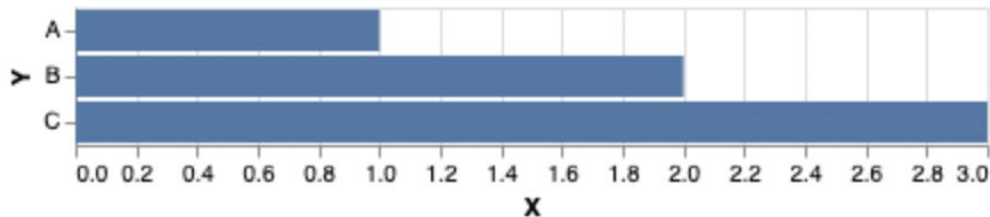
```
pip install altair
```

# A First Example

```
import pandas as pd
import altair as alt

df = pd.DataFrame({
    'X' : [3,2,4],
    'Y' : ['M','N','O']
})
```

```
chart = alt.Chart(df
).mark_bar(
).encode(
    x = 'X:Q',
    y = 'Y:N'
)
```



# Encoding Channels

- x, y
- color, size, opacity,
- Tooltip, text

# Data Types in Encodings

- :Q = Quantitative
- :N = Nominal
- :O = Ordinal
- :T = Temporal



# References

- <https://altair-viz.github.io/>
- <https://github.com/alod83/Data-Storytelling-with-Altair-and-AI/>
  - Many examples of charts under 06/
- Lo Duca, A. (2024). Data Storytelling with Altair and AI. Manning Publications.

# Let's practice!

Given an array of x and an array of y, plot an corresponding (x, y) points as a line chart.

```
def plot_line_chart(x, y):  
    chart = altair.Chart(data="x, y")  
    chart.mark_line().encode(x="x",  
                             chart
```

# Example:

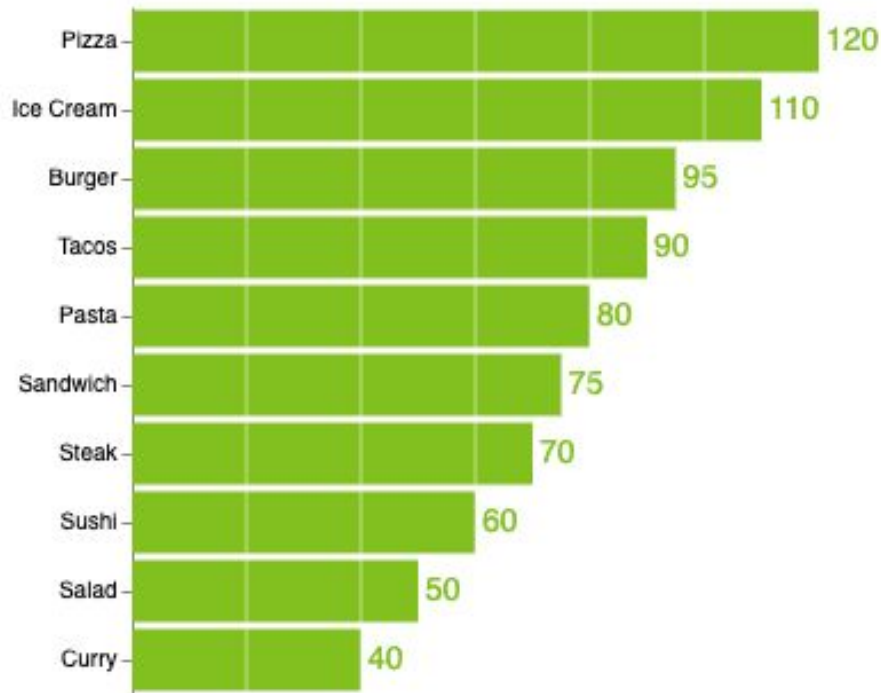
x = [1, 2, 3, 4, 5, 5]

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



# Exercise 1 - Bar Chart

- <https://github.com/alod83/Data-Storytelling-with-AI/>
  - 06
    - Bar-charts
      - bar-chart.py



## Exercise 2 - Donut Chart

- <https://github.com/alod83/Data-Storytelling-with-Altair-and-AI/>
  - 06
    - cooking-charts
      - donut-chart.py

