



EuroPython





EuroPython



**Python and Data Storytelling
to create and deliver better presentations**

Sebastián Flores, EuroPyCon 2025



EuroPython

Part 1 - Block 3

Data Storytelling Basics



EuroPython

Agenda

-
- No
- Nope
- No chance

Never give away the ending!
Always raise the stakes!

 **Important**

Do whatever works for your presentation. Be as unconventional as you want/can.

You are in charge of the **narrative** and the **instruments** to tell it.



EuroPython

Why Data Storytelling?

¿Storytelling?



Stories are the first human technology
It's part of how our brains are wired!

What's the main plot of:

- Cinderella, The beauty and the beast, Lillo & Stich, ...
- Jurassic Park, Star Wars, Back to the future, ...

Stories are easy to remember.

A presentation **with stories** will be easier to remember.

The hackable brain

The main point of Data Storytelling is that brains have some **very predictable patterns**.

Storytellers know and exploit this.

Example #1

Peak-End Rule

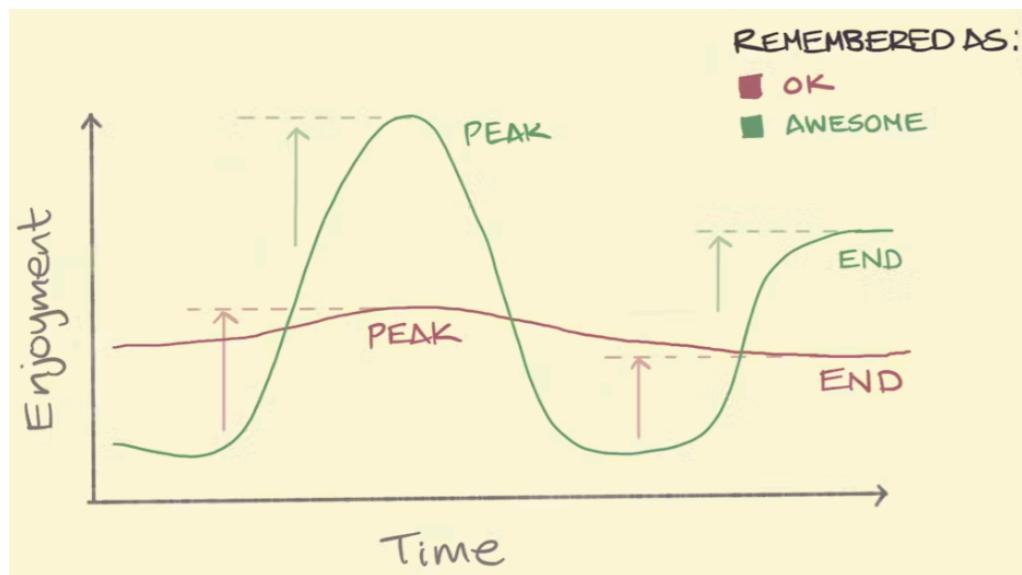
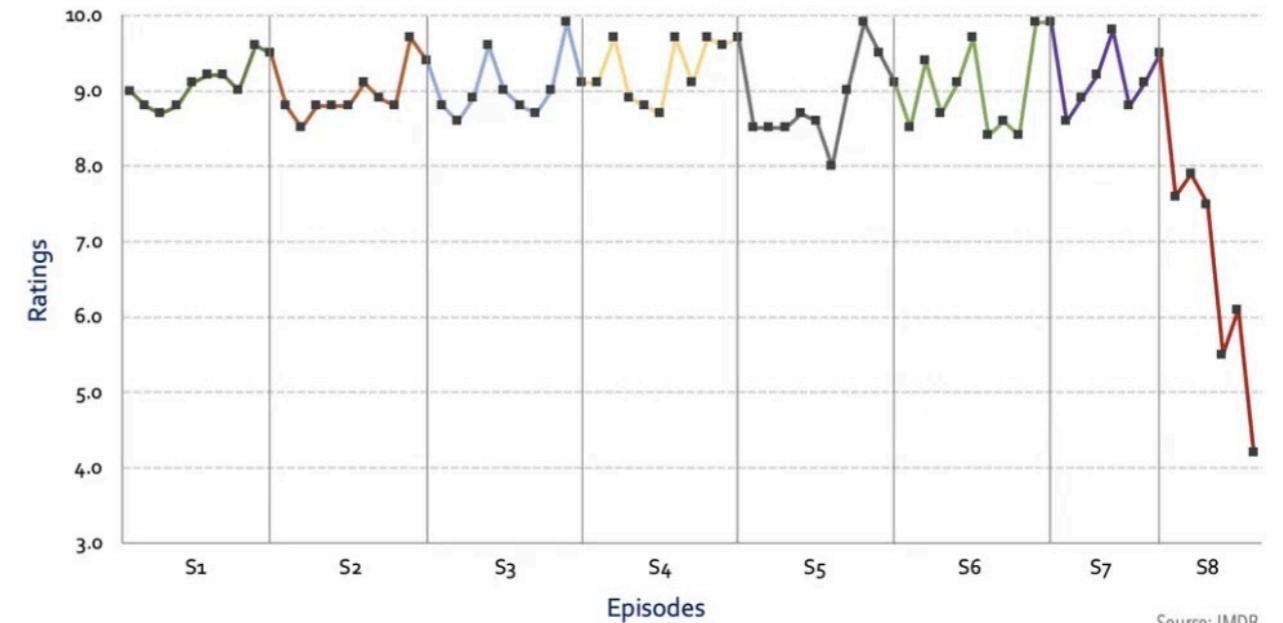


Image by sketplanation



Game of Thrones rating, by Kelvin Neo



Example #1: What did we learn?

Endings are the most important part of the story (and the presentation).

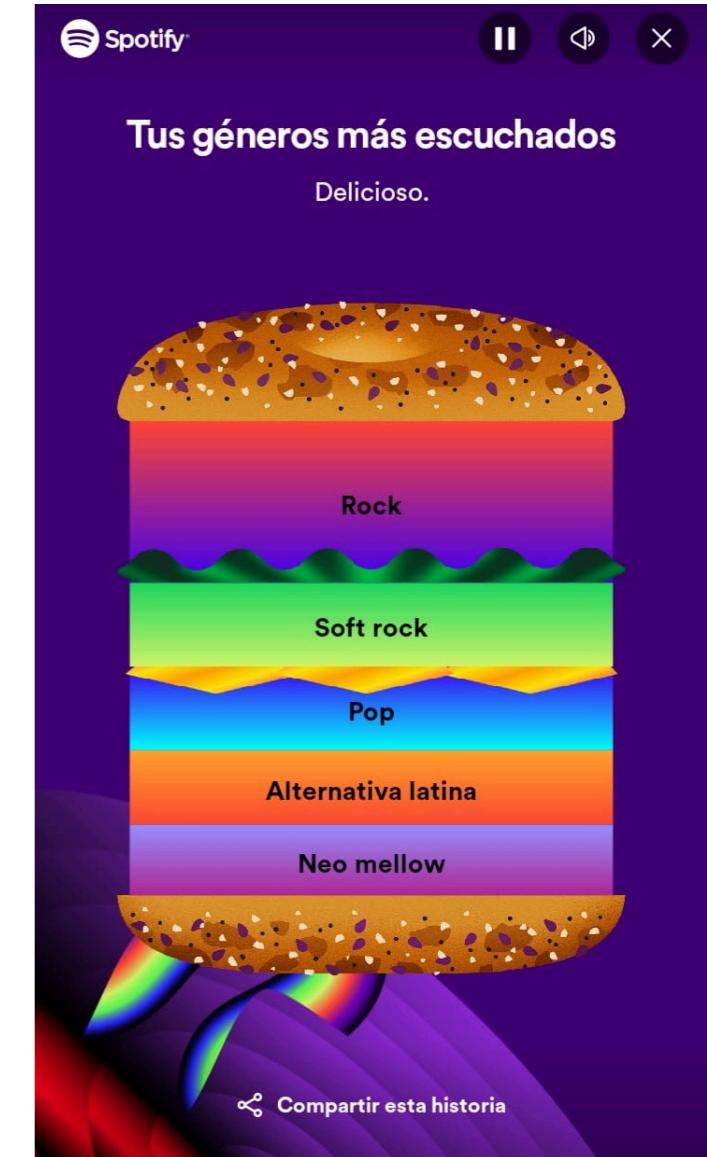
The ending is what gets remembered. It should tie everything together.

Work on your final slides - don't leave it to change. Skip the middle slides if needed, never the last ones.

Example #2



How to make millions of people share statistics on social media?





Example #2: What did we learn?

Statistics get ignored.

Emotions are remembered.

Stories get shared.

Data Storytelling:
Using storytelling **tricks** to create
presentations that will be **remembered**
and that **create impact**



Emotions create actions



EuroPython

5 Data Storytelling Principles

Principle #1

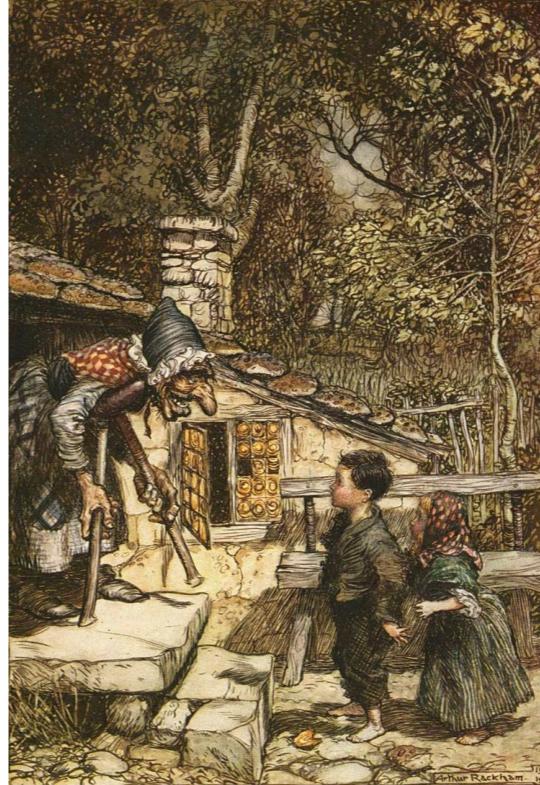
Always raise the stakes!

Never give away the ending!

This is called “**creating a narrative debt**”

Hansel and Gretel

- **Main Problem:** Food scarcity at home
 - **Problem 1:** Father wants to abandon H&G at forest
 - *Solution:* Use breadcrumbs to mark the road
 - *Problem:* Birds eat breadcrumbs and they get lost
 - **Problem 2:** H&G get lost and hungry
 - *Solution:* Find a sugar house. Eat it!
 - *Problem:* They get captured by house owner: a witch!
 - **Problem 3:** Witch has cannibalistic intentions
 - *Solution:* Use chicken finger to trick the witch
 - *Problem:* She is suspicious
 - **Problem 4:** She wants to put them in the oven.
 - *Solution:* Push witch in oven.
- **Solution:** Return home with witch's treasure.



Principle #2

Details are important, but not all details are important.

Keep only the essential.

This is called “**The Chekhov’s Gun**”

This manifests in:

- Less text
- Rounding numbers
- Simpler graphs
- Analogies and metaphors

Don't try to look smart by using **fancy words**.

Smart people are those who can explain complex ideas in simple ways,
not who knows more **jargon**.

Principle #3

People like better when they participate on the process.

Make people work for their meal (but never too much).

This is called “The IKEA effect”.

People don't like to be spoonfed nor ignored.

Leave enough breadcrumbs for people to follow and anticipate the story/solution, without revealing the ending.

THE IKEA EFFECT WE LOVE IT MORE IF WE MADE IT

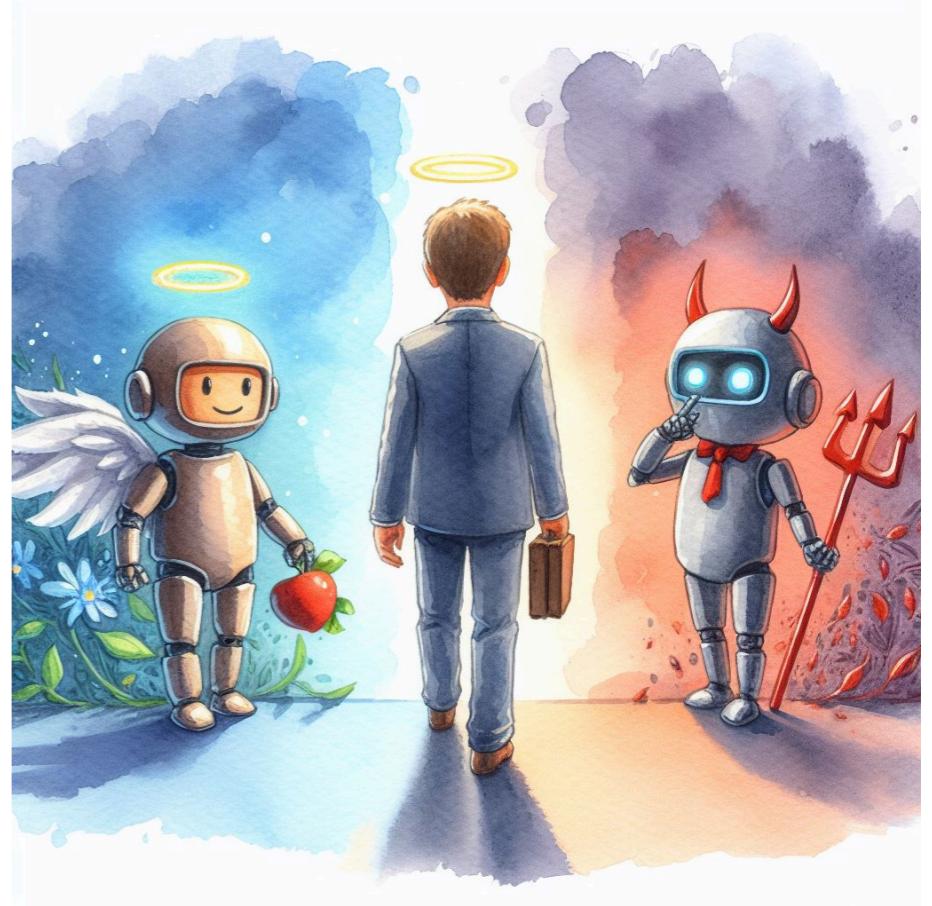


Sketchplanations

Principle #4

Your first draft is always going to be terrible.

- Use AI chatbots for:
 - Analogies and examples
 - Prompts to create better images!
- Don't search for images, create them!



Principle #5

Show, don't tell.

- Don't say "this is the best": show why it's be the best.
- Don't say the code is clean: show the code.

Show, don't tell.

Guido's eyes lit up as the terminal finally executed without errors. A wide grin spread across his face. He jumped up from his chair, fists raised in triumph. "Yes!" he exclaimed, as he savored the moment of triumph with a deep, contented sigh.

Guido was happy because he solved the bug on his code.

Show, don't tell for engineers

Tell:

```
print(*objects, sep=' ', end='\n', file=None, flush=False)
```

Print objects to the text stream file, separated by sep and followed by end. sep, end, file, and flush, if present, must be given as keyword arguments.

All non-keyword arguments are converted to strings like str() does and written to the stream, separated by sep and followed by end. Both sep and end must be strings; they can also be None, which means to use the default values. If no objects are given, print() will just write end.

The file argument must be an object with a write(string) method; if it is not present or None, sys.stdout will be used. Since printed arguments are converted to text strings, print() cannot be used with binary mode file objects. For these, use file.write(...) instead.

Output buffering is usually determined by file. However, if flush is true, the stream is forcibly flushed.

Show, don't tell for engineers

Show:

Documentation:

```
1 print(*objects, sep=' ', end='\n', file=None, flush=False)
```

Executable code:



A screenshot of a Jupyter Notebook cell. The cell contains the following code:

```
1 print("Hello", "lovely", "people", sep="\n") # Try: "", " ", "🇨🇿", ...
```

The cell has a green 'Run Code' button at the top left. At the top right, there are icons for refresh, copy, and paste. The code area has a light gray background with a dark gray border.

The framework

Brandon Sanderson's storytelling framework:

- **Promise:** What is the story about?
- **Progress:** Is the story advancing towards the promise?
- **Payoff:** Did you fulfill or break the promise?

Payoff != Ending.

Payoff is how satisfactory is the ending of the story.

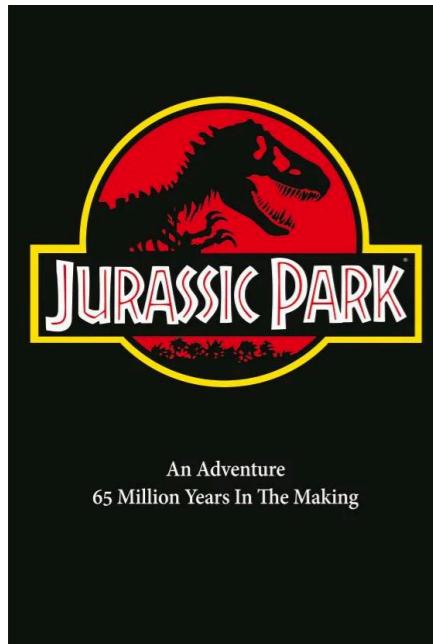
Was the effort worth it?

Example: Star Wars IV



- **Promise:** Rebels fight back against the Empire.
- **Progress:** Luke develops his skills as a Jedi.
- **Payoff:** The death star explodes!

Example: Jurassic Park I



- **Promise:** Accidents in a park with dinosaurs.
- **Progress:** Accidents get progressively worse.
- **Payoff:** Protagonists survive but dinasaurs overrun the park.

Example:

Let's apply it to any story or presentation!

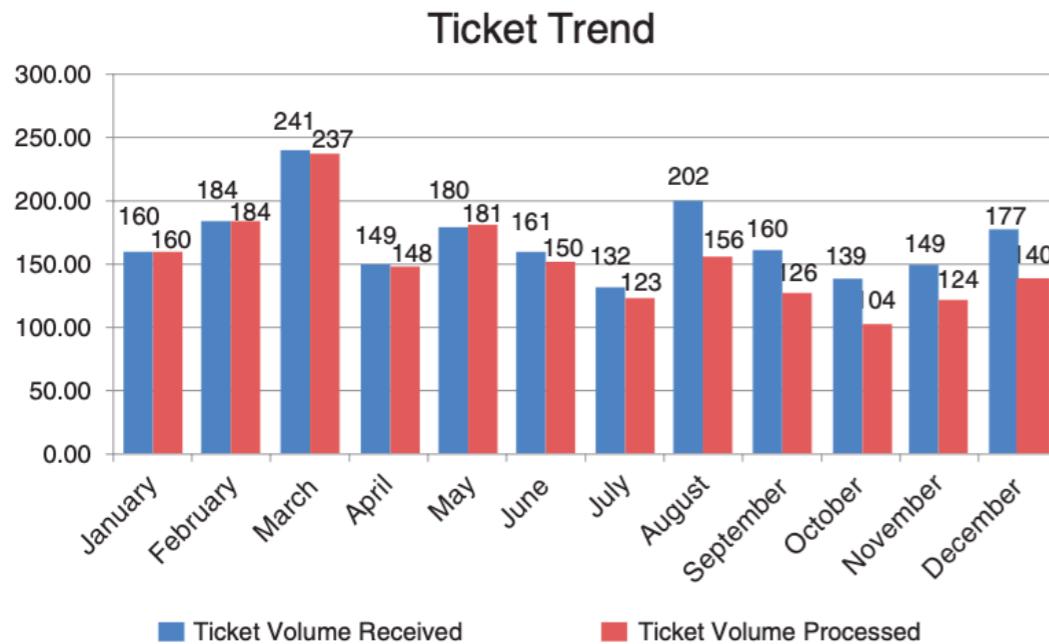
- **Promise:**
- **Progress:**
- **Payoff:**

Some ideas:

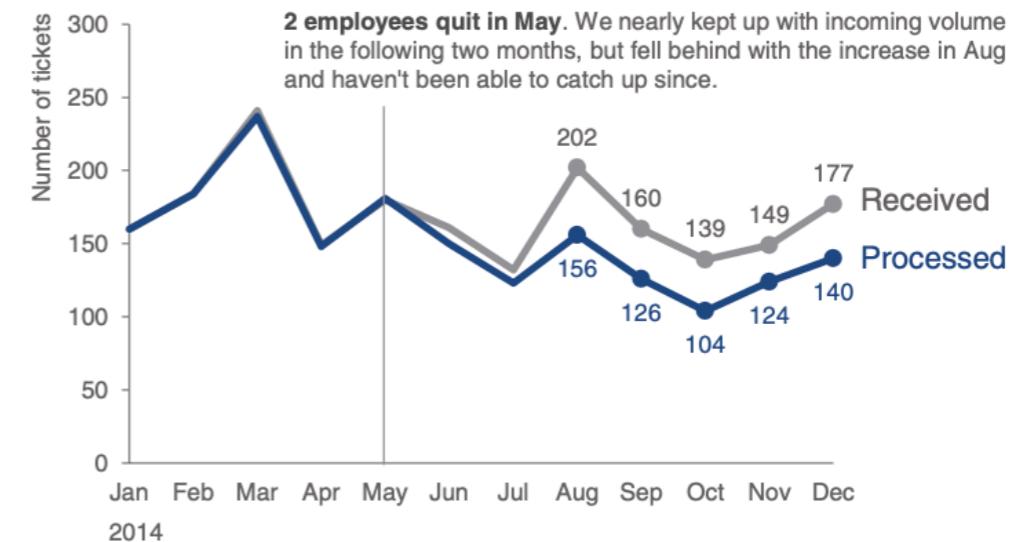
The Lord of the Rings, The Name of the Wind, Ring World, ...

How does this applies to data and slides?

Same data, two plots:



Ticket volume over time



Don't share numbers. Share a story.

(C) Storytelling with Data, por Cole Nussbaumer Knaflic.

Recommended reading - The short list

- Storyworthy, Matthew Dicks
- Storytelling with Data, Cole Nussbaumer Knaflic
- How charts lie, Alberto Cairo
- The Visual Display of Quantitative Information, Edward Tufte

Recommended reading for Storytelling

- Storyworthy - Matthew Dicks
- Made to Stick - Chip Heath and Dan Heath
- Bird by Bird – Anne LaMott
- The Science of Storytelling – Will Storr
- A Swim in a Pond in the Rain – George Saunders
- Steering the Craft – Ursula K. Le Guin

Recommended reading for Data Storytelling

- Storytelling with Data - Cole Nussbaumer Knaflic
- The Visual Display of Quantitative Information - Edward Tufte
- Beautiful Evidence - Edward Tufte
- How charts lie - Alberto Cairo
- The Truthful Art - Alberto Cairo
- The Functional Art - Alberto Cairo
- Matplotlib Journey (<https://www.matplotlib-journey.com/>) - Yann Holtz and Joseph Barbier