

# Más allá de matplotlib y reportes PDF

creación de dashboards dinámicos con Dash de Plotly y Python



Dash



# Si se quedan

1. Principios de una buena visualización (Tamara Munzner)
2. Introducción a Dash con una app sencilla
3. Construcción de una app de tamaño medio con Dash
4. Extras y futuro

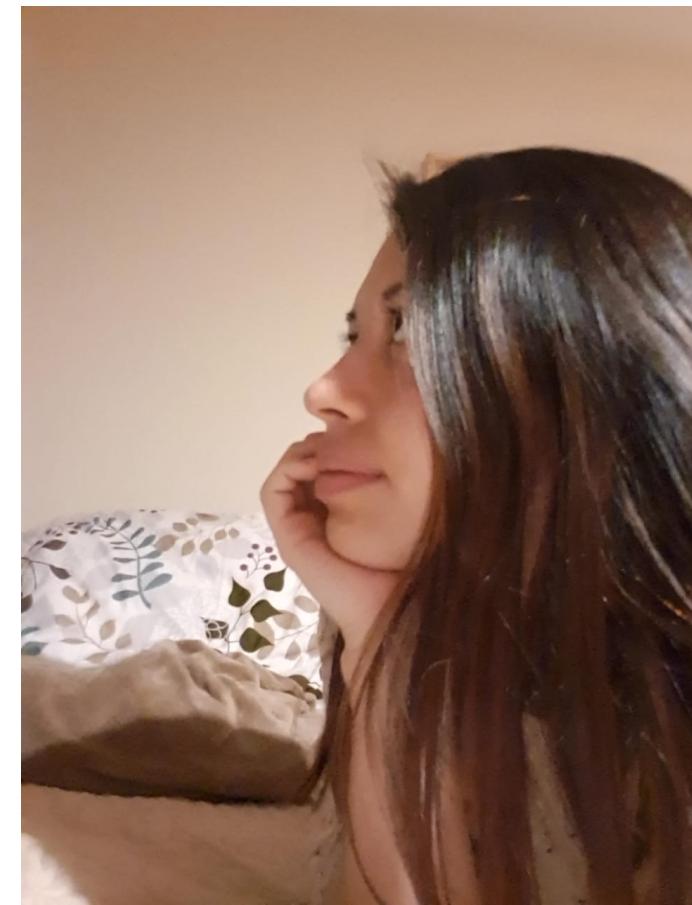
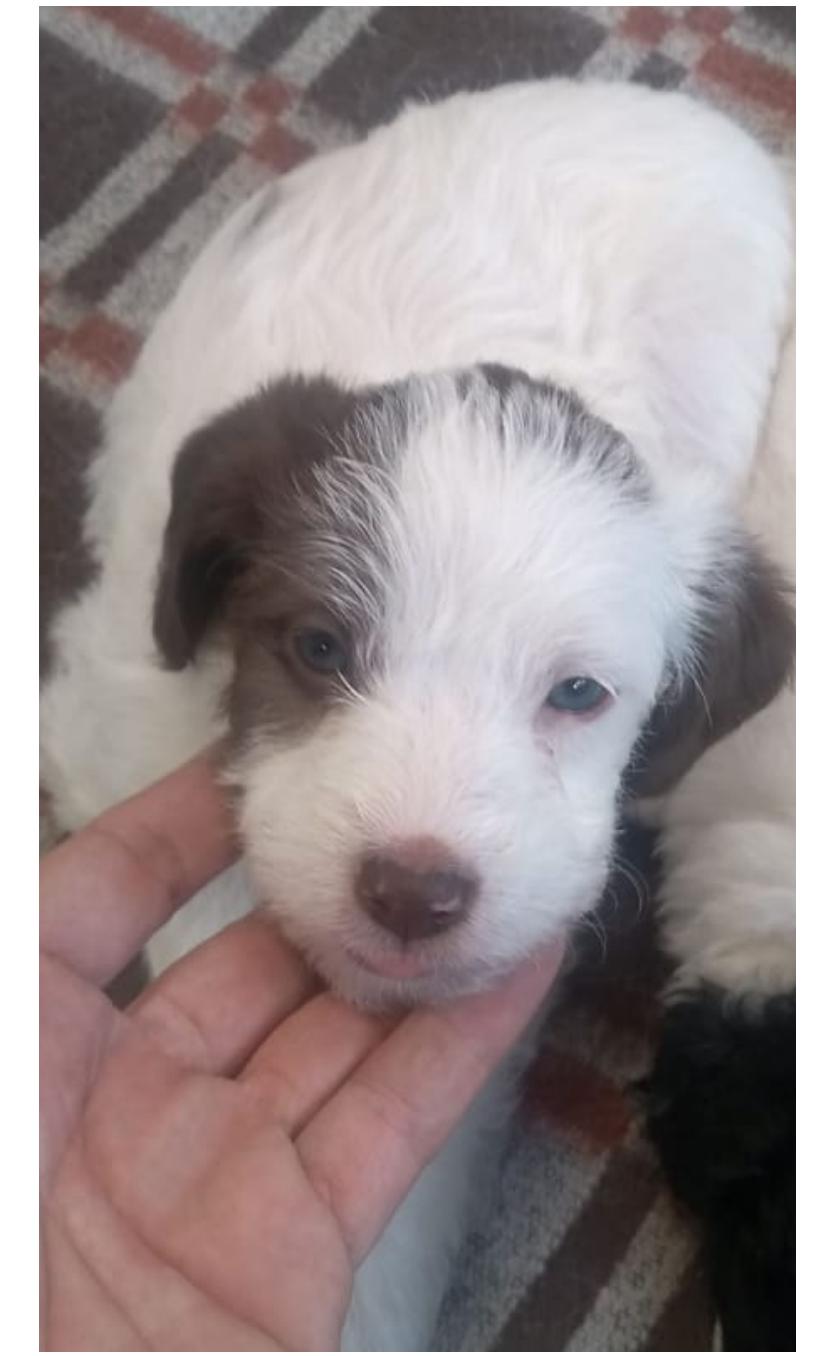
# Un poco sobre mí

# aré coffee



quantil

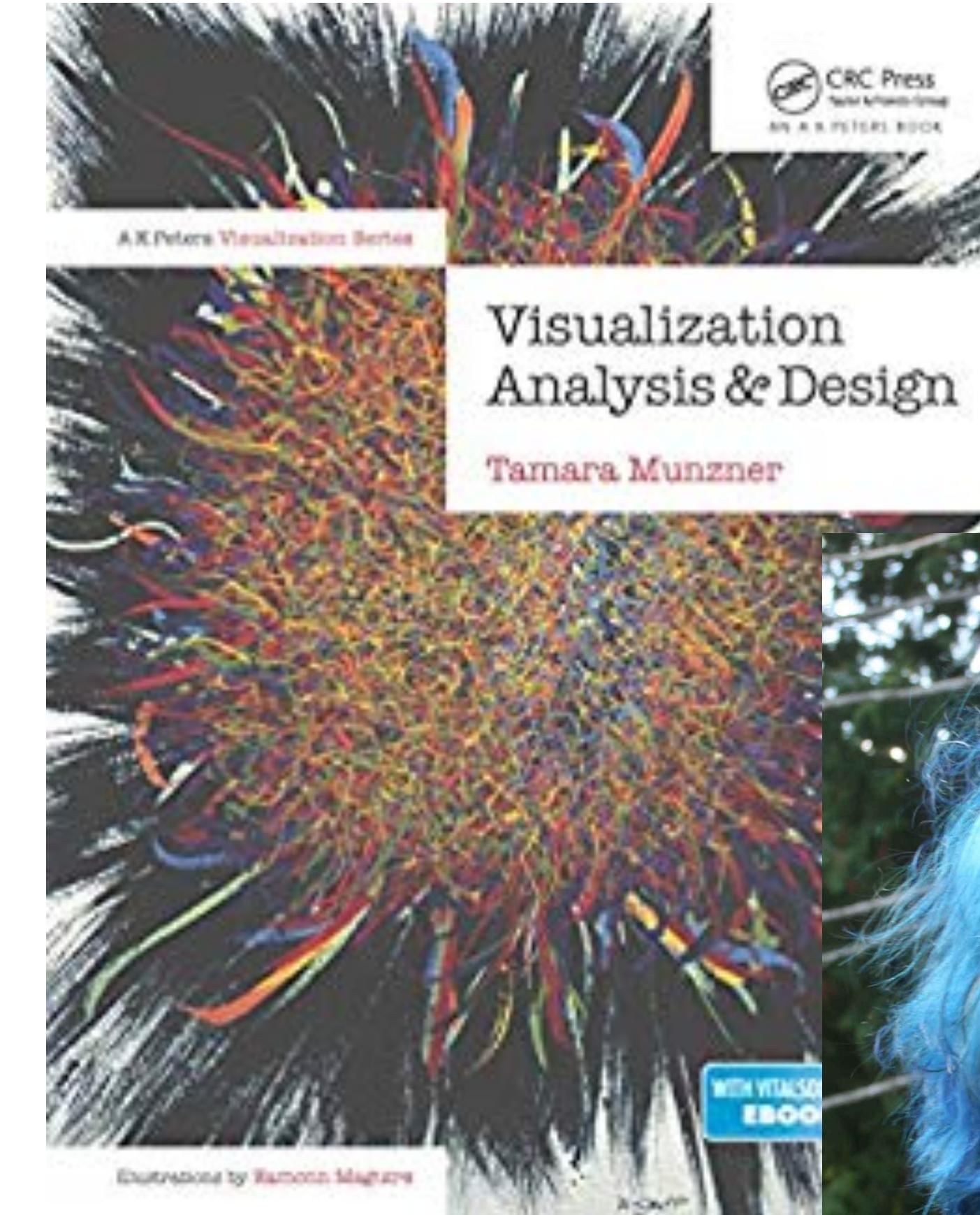
matemáticas aplicadas



# **Framework para una buena visualización**

# Ella es Tamara Munzner, ella se lo inventó

- University of British Columbia  
(actual)
- University of Minnesota
- Stanford University



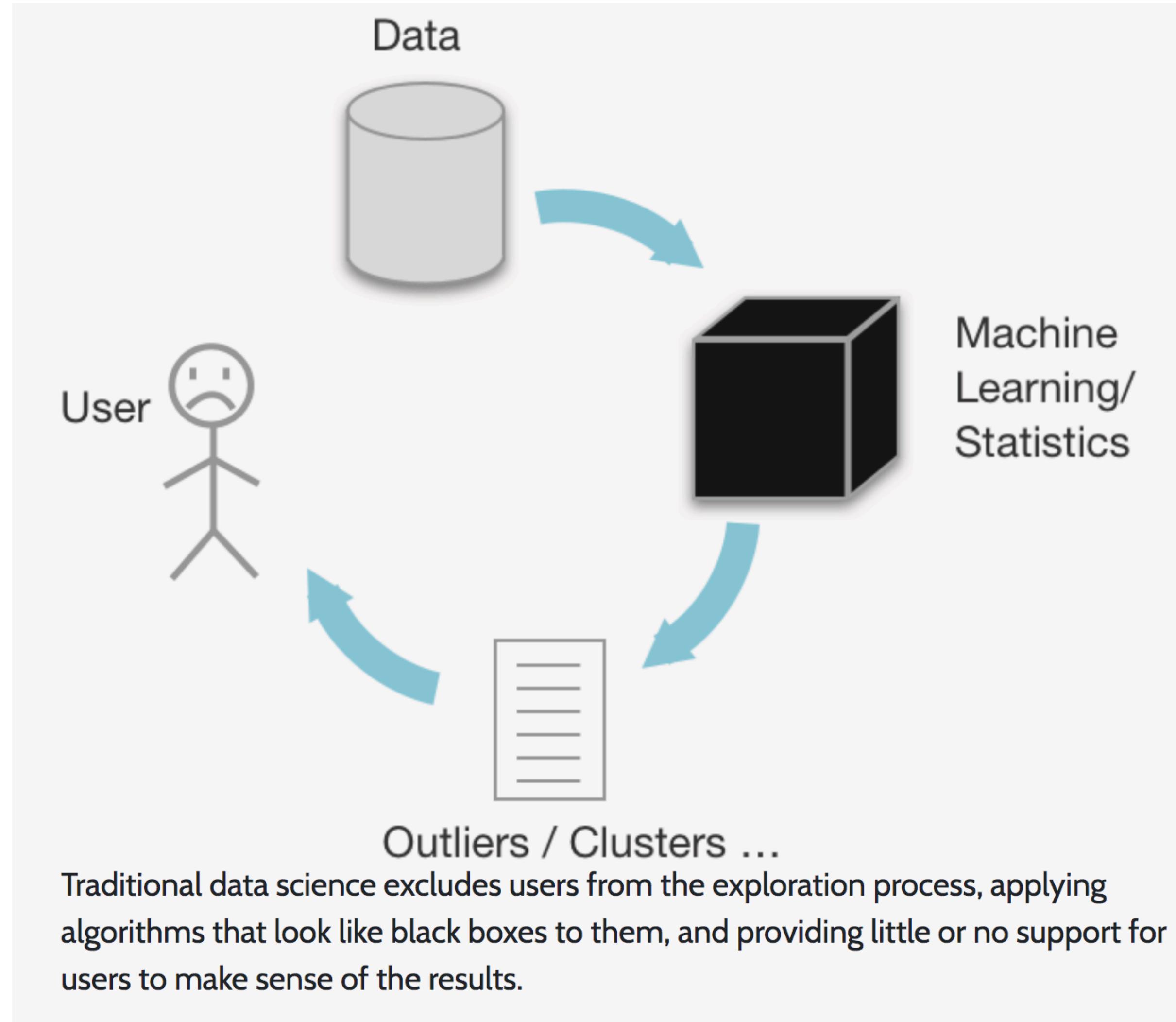
**Él es John Guerra,  
él me lo enseñó**

- Northeastern University at Silicon Valley
- UC Berkeley
- Universidad de los Andes (Yahoo Labs, Parc)
- University of Maryland

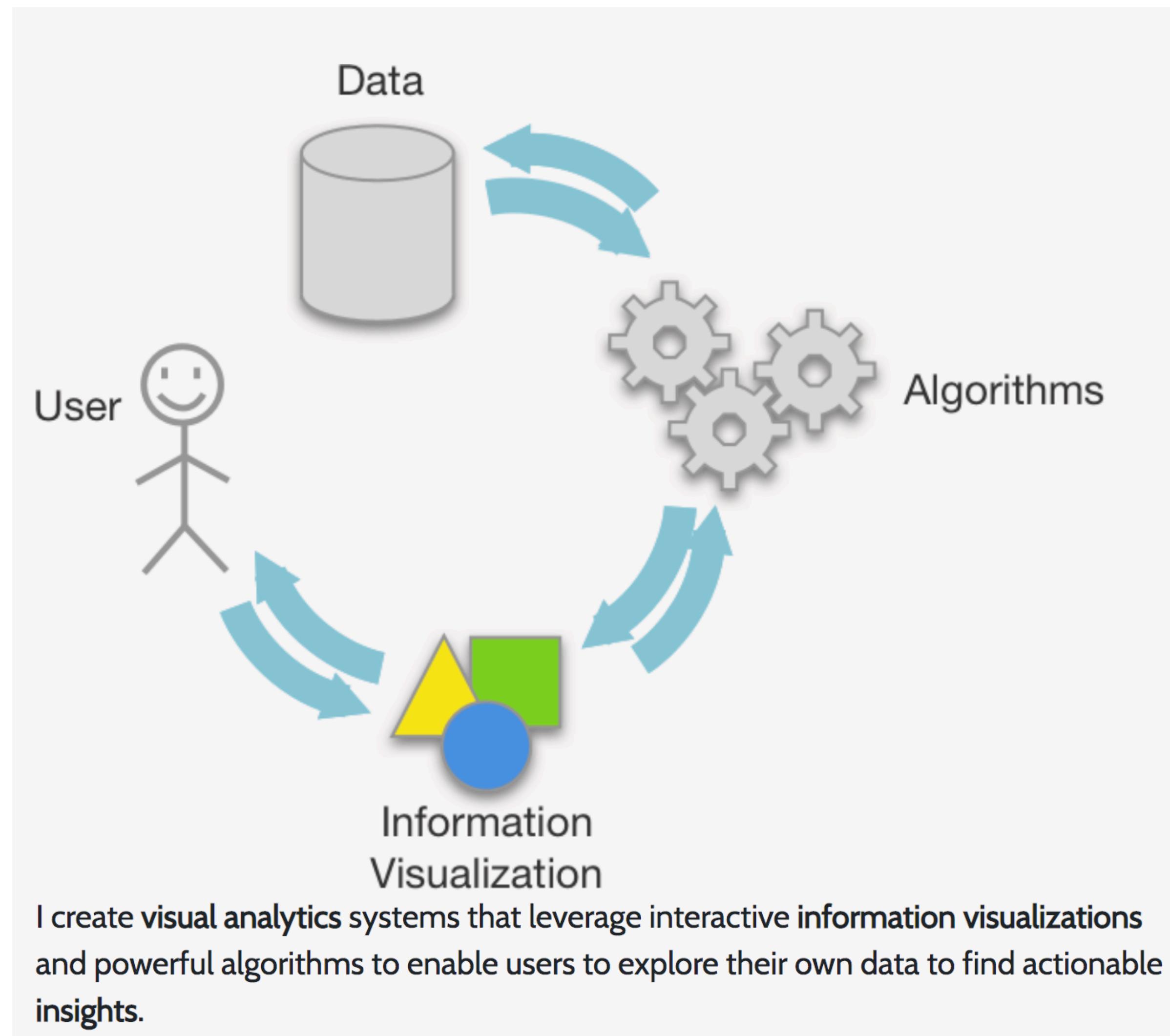


**Usamos visualización basada en computación  
cuando hay una necesidad de aumentar las  
capacidades humanas y no una de reemplazar  
dichas capacidades con algoritmos**

# Normalmente



# Para Visualización



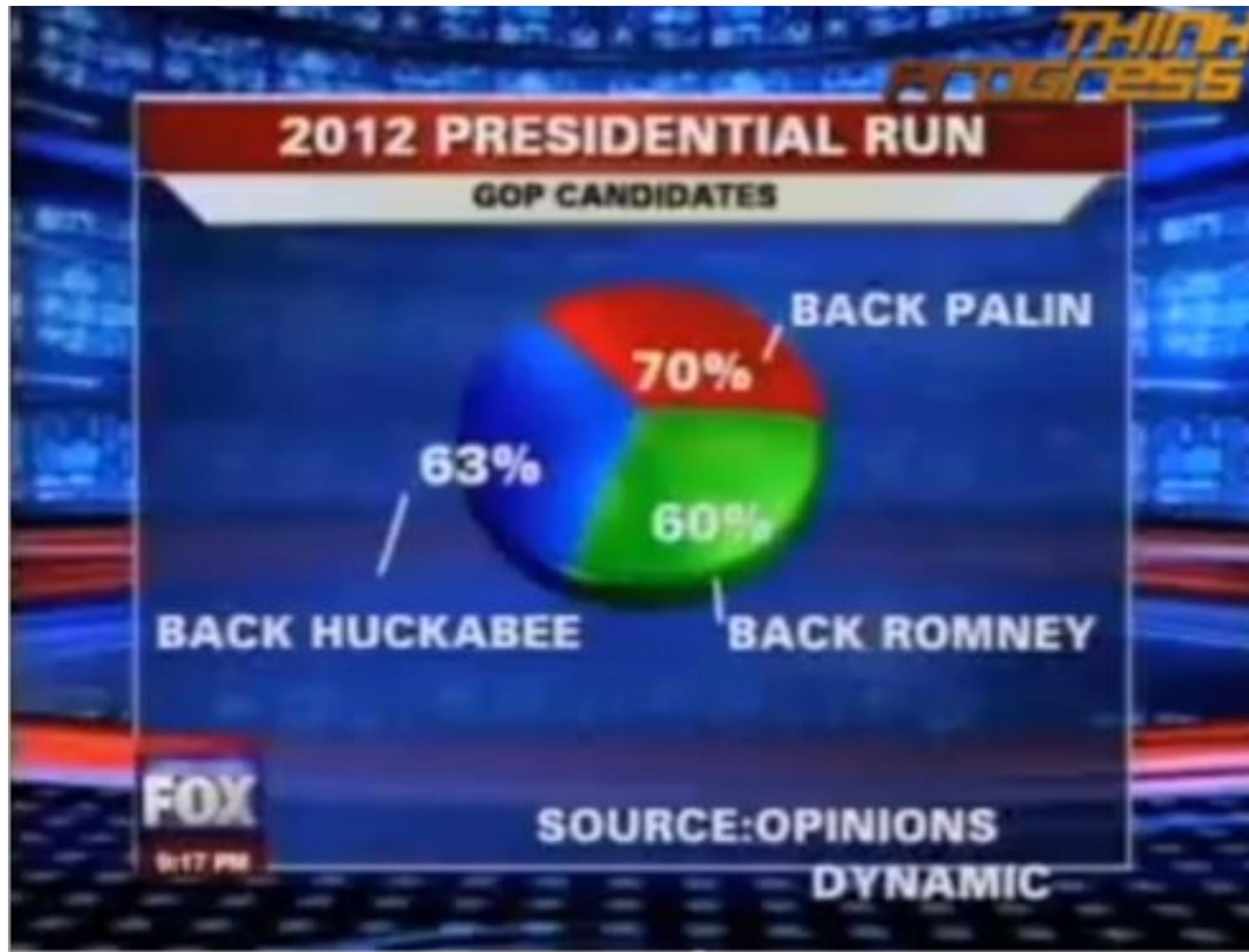
# Casos

- 1. Análisis exploratorio** (caso más común)
- 2. Presentación de resultados conocidos** (Periodismo):  
Datasketch, Vox, New York Times, etc.
- 3. Primera milla en Data Science:** Ayudar al constructor del modelo entender lo suficiente del problema para poder construir un modelo
- 4. Última milla en Data Science:** Asegurarse que el modelo tenga sentido para quien lo hizo y para los stakeholders con el suficiente know how. Todo con el fin de refinar el algoritmo o modelo

**Cognición => Percepción**

**Para qué necesitamos un  
framework?**

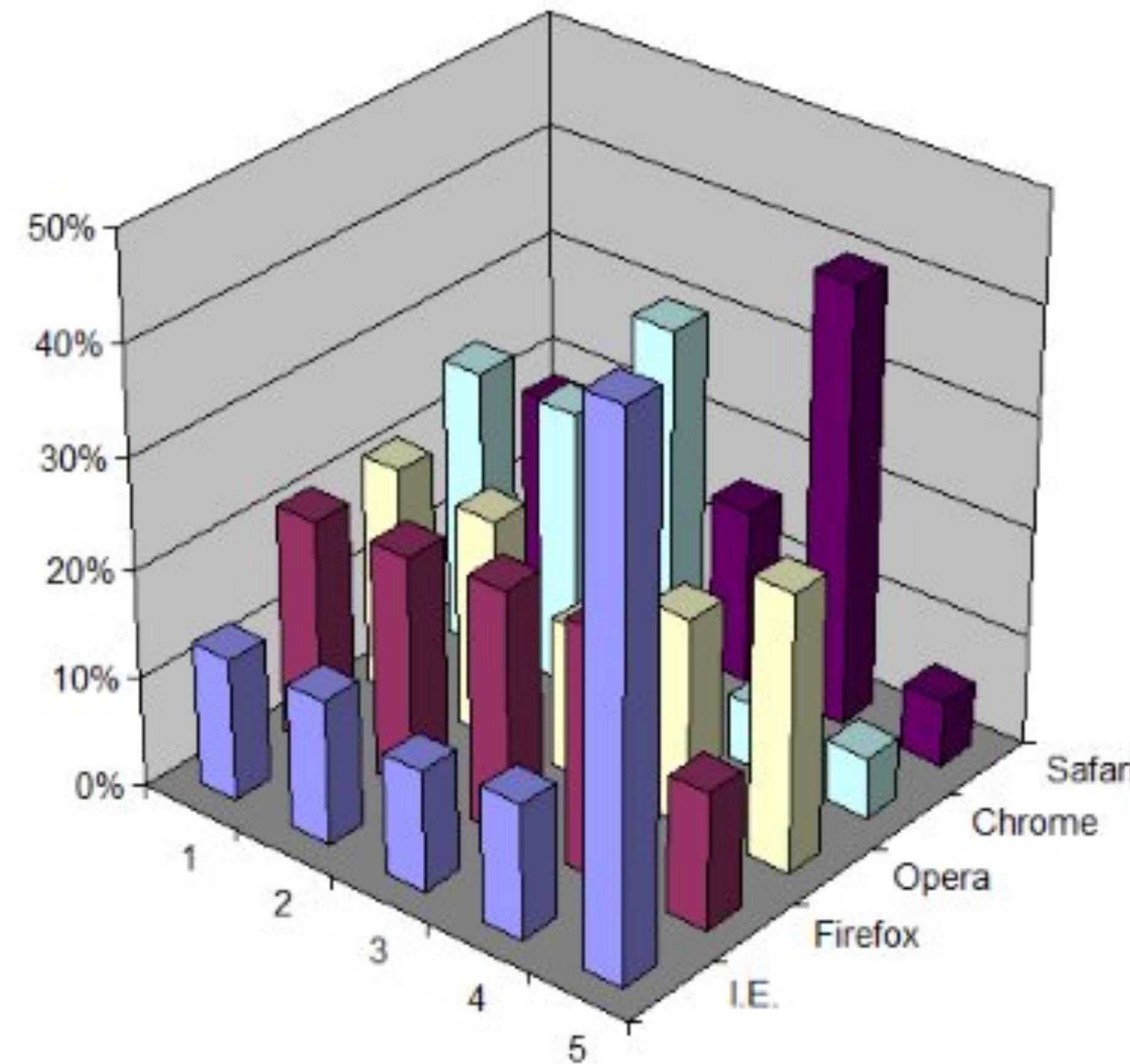
# Para que estas cosas no pasen



- Todo suma 193%
- El contraste con lo de atrás es horrible
- Digamos que tiene sentido.... los pedazos de Pie se ven iguales!

Tarea: Identificar % de votantes de cada candidato y poder compararlos entre sí

# Para que estas cosas no pasen



- Lo de atrás no se ve
- Los colores son horribles
- Qué es 1, 2, 3, 4, 5?
- Los porcentajes son qué?

Tarea: Identificar el % de mercado que cada navegador tuvo de 2011 a 2015 y compararlos entre sí a través del tiempo y a su vez entre navegadores

# Para que estas cosas no pasen



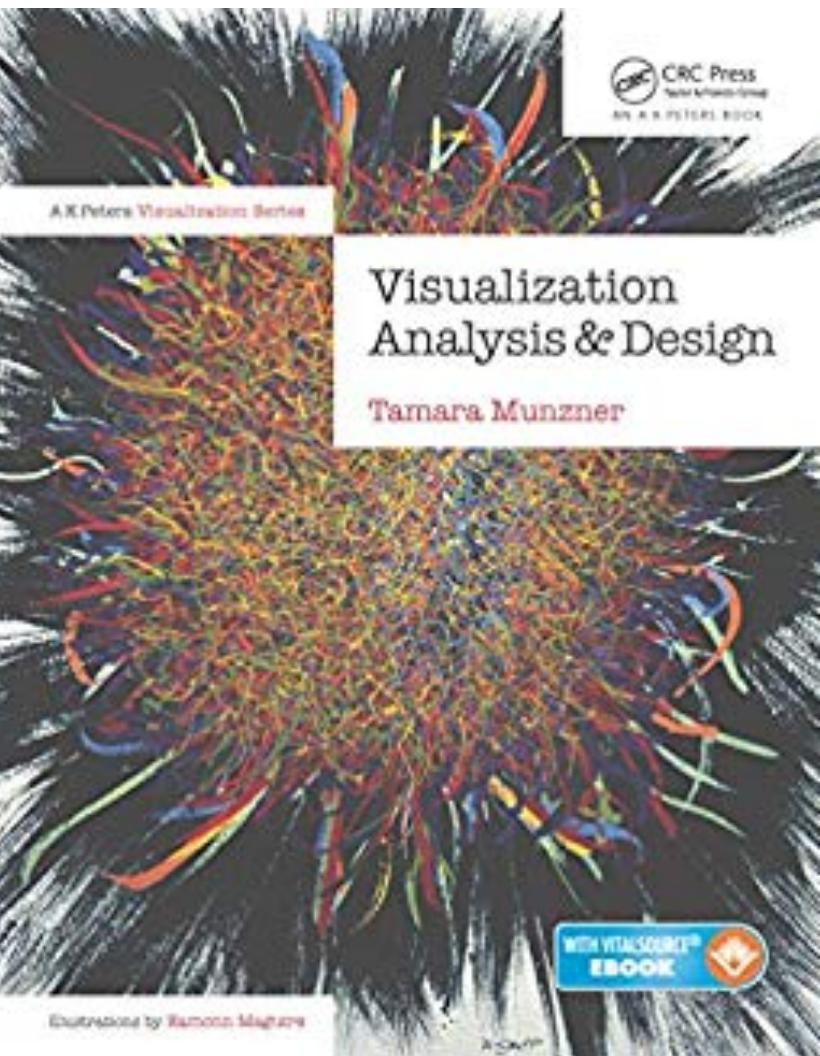
Usando  
mala  
visualización  
para  
ensuciar un  
proyecto  
político

**Tarea: Identificar el número que hace falta para alcanzar la meta de enrollment de marzo 31**

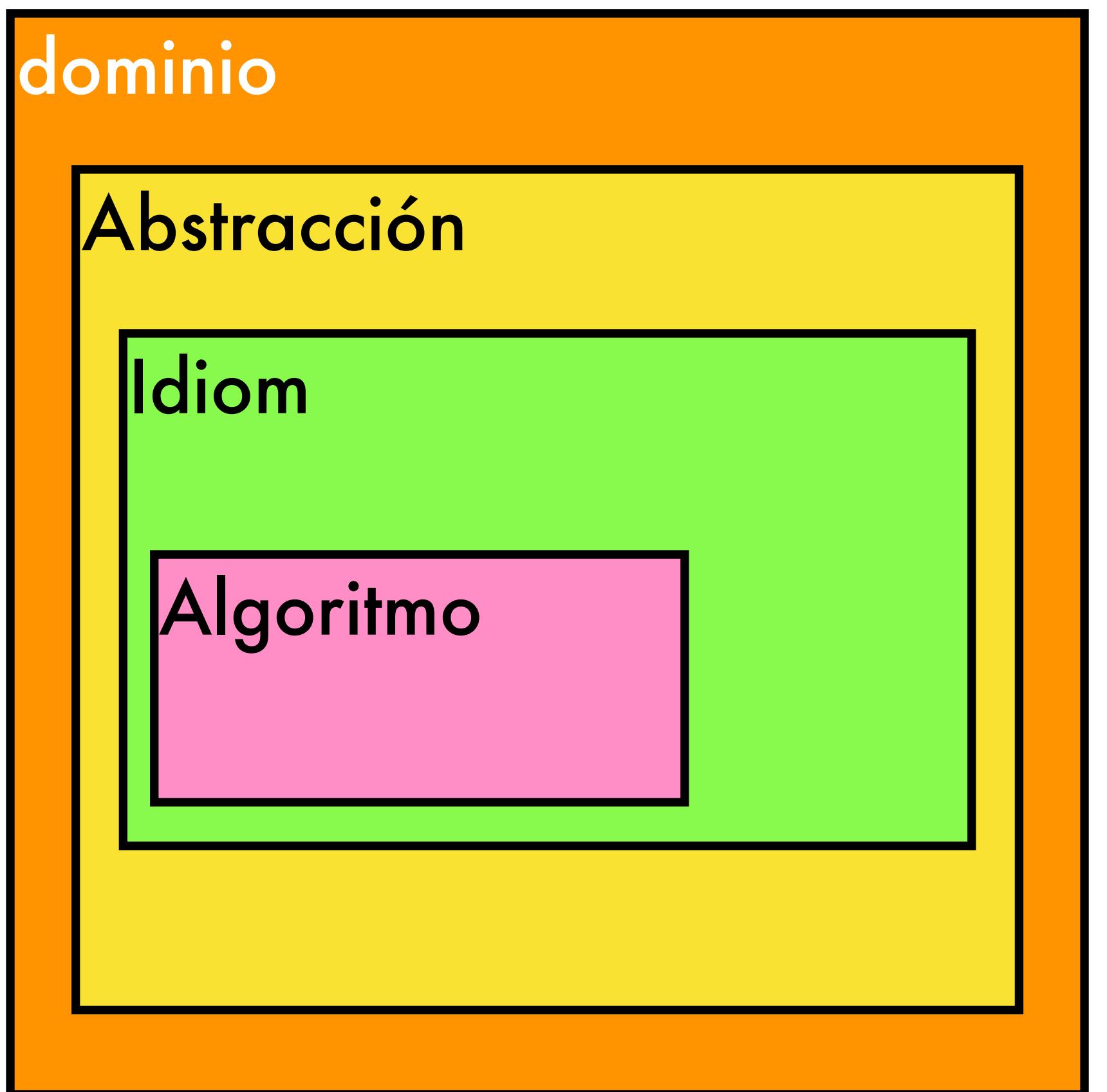
**Tarea: Identificar el % o proporción de personas que hacen falta respecto a la meta**

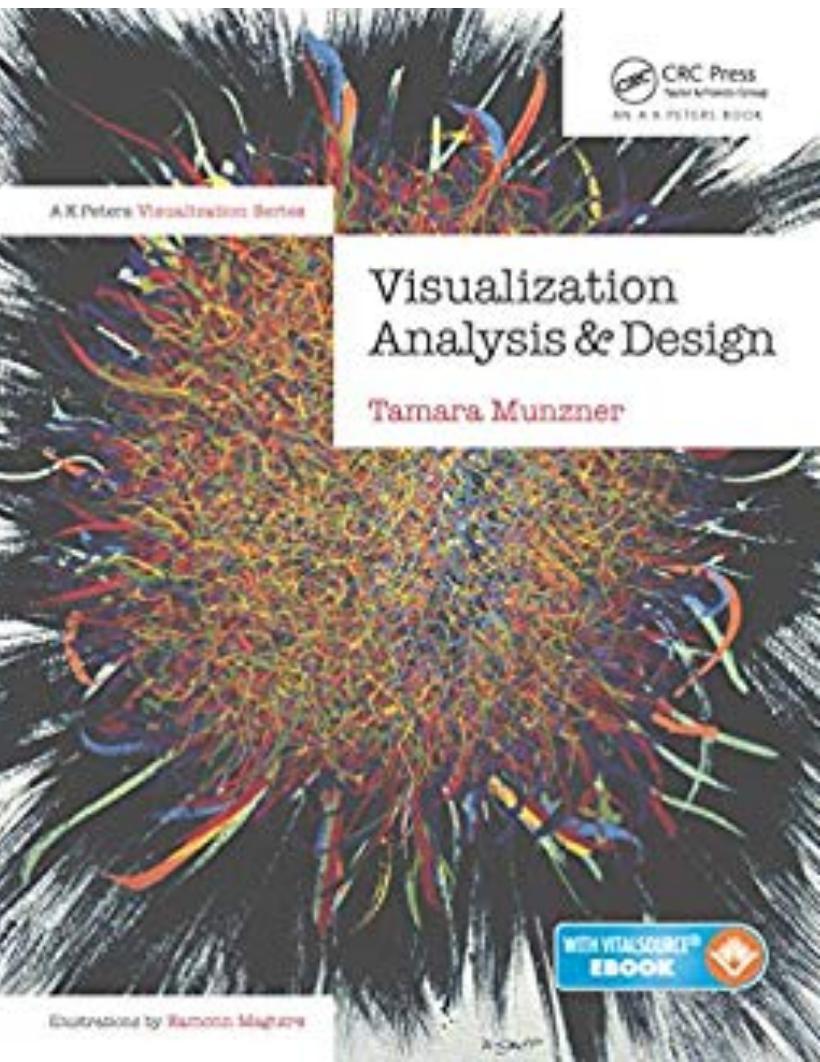
**El problema es: estos son casos  
específicos.....**

¿cómo pensamos de manera  
general?

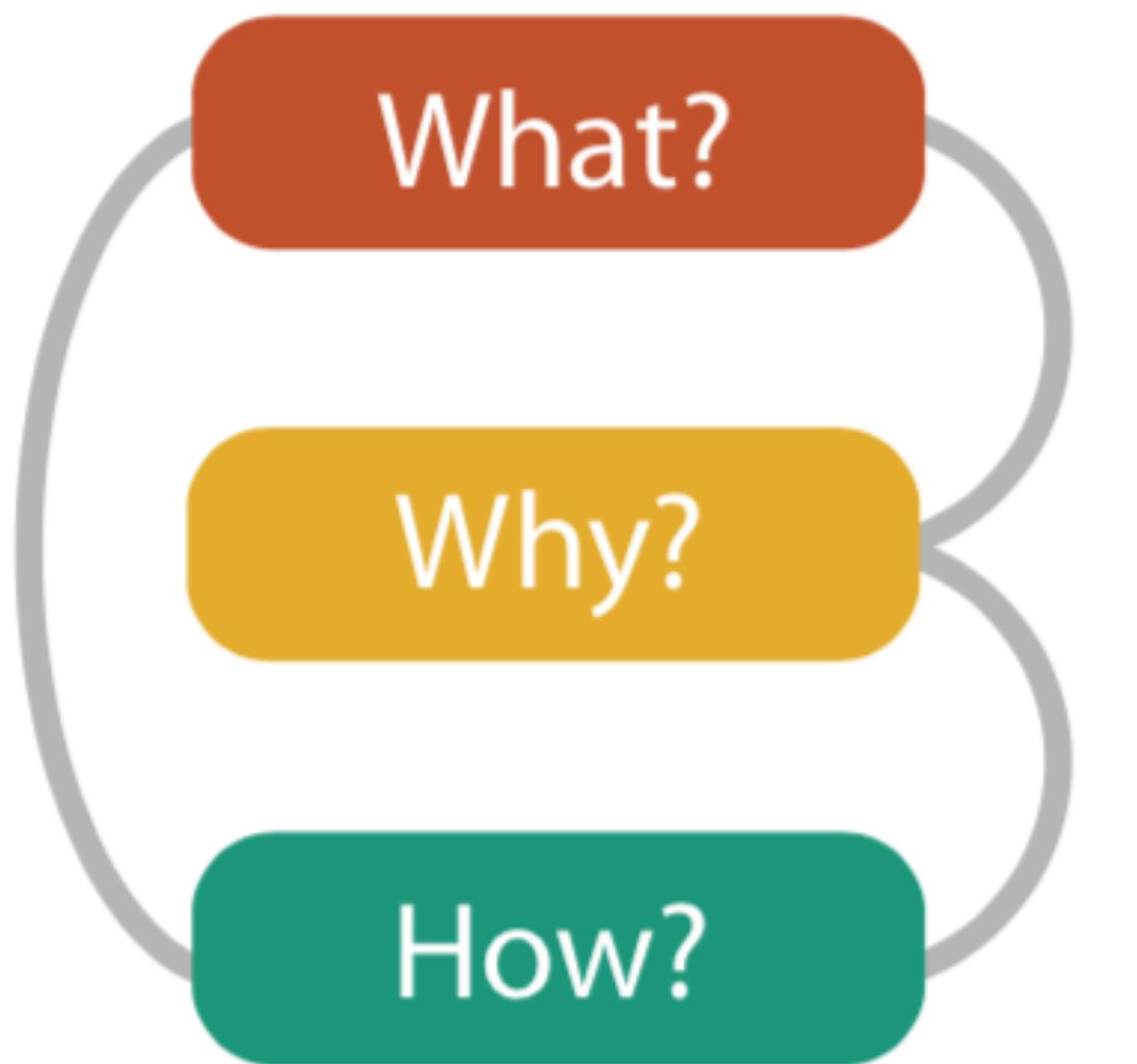


# El framework

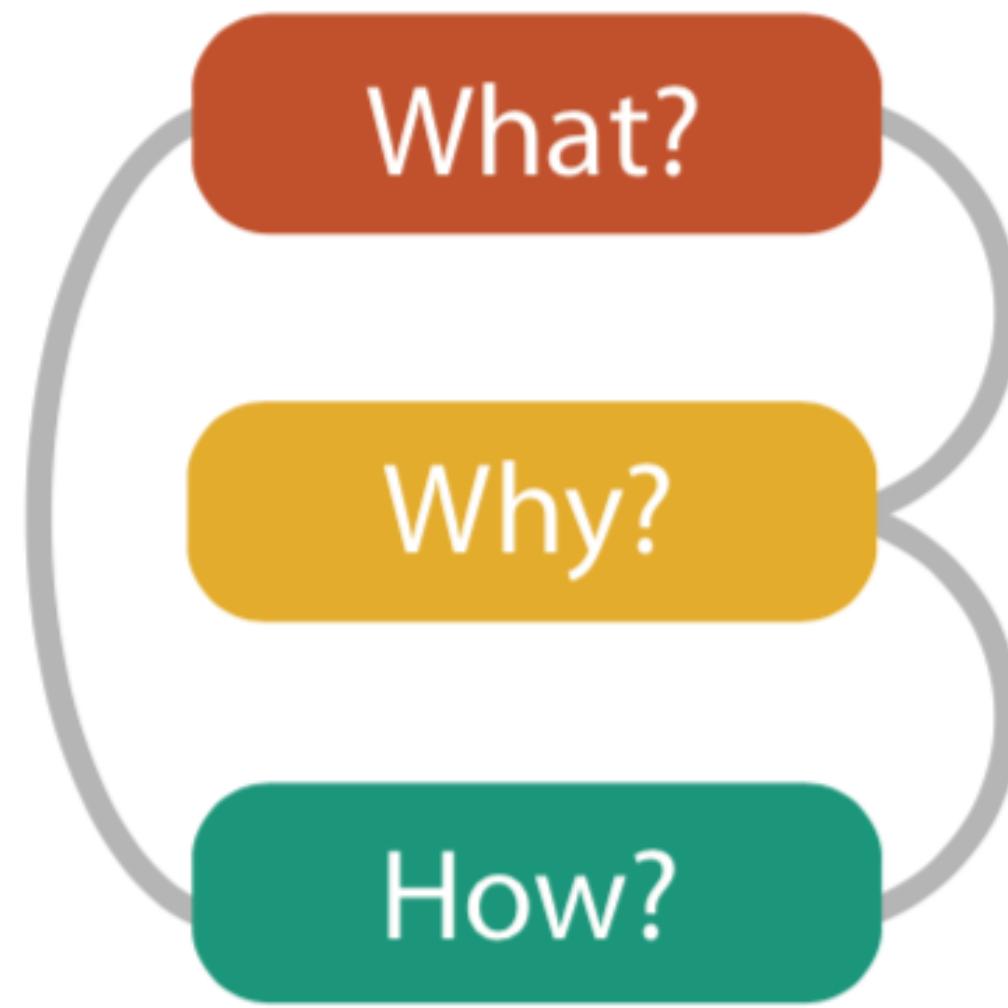




# EI framework



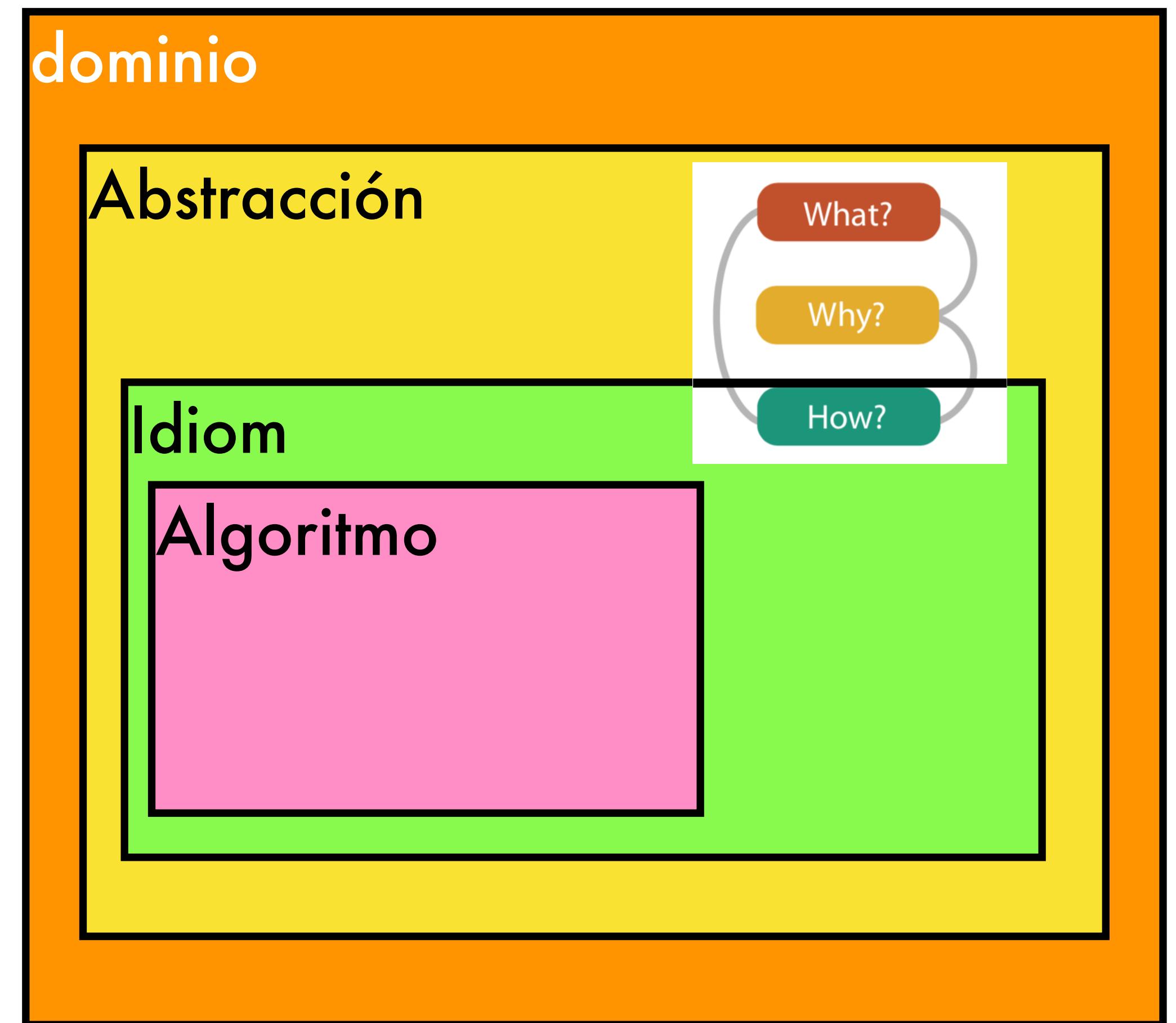
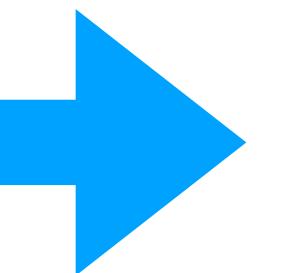
# El framework



1. **Quiénes** son los usuarios objetivo?
2. **Qué** datos queremos mostrar?
3. **Por qué** el usuario quiere verlo?
4. **Cómo** mostramos los datos?

# El framework

1. Quiénes son los usuarios objetivo?
2. Qué datos queremos mostrar?
3. Por qué el usuario quiere verlo?
4. Cómo mostramos los datos?



# Por qué es difícil?



## Domain situation

You misunderstood their needs



## Data/task abstraction

You're showing them the wrong thing



## Visual encoding/interaction idiom

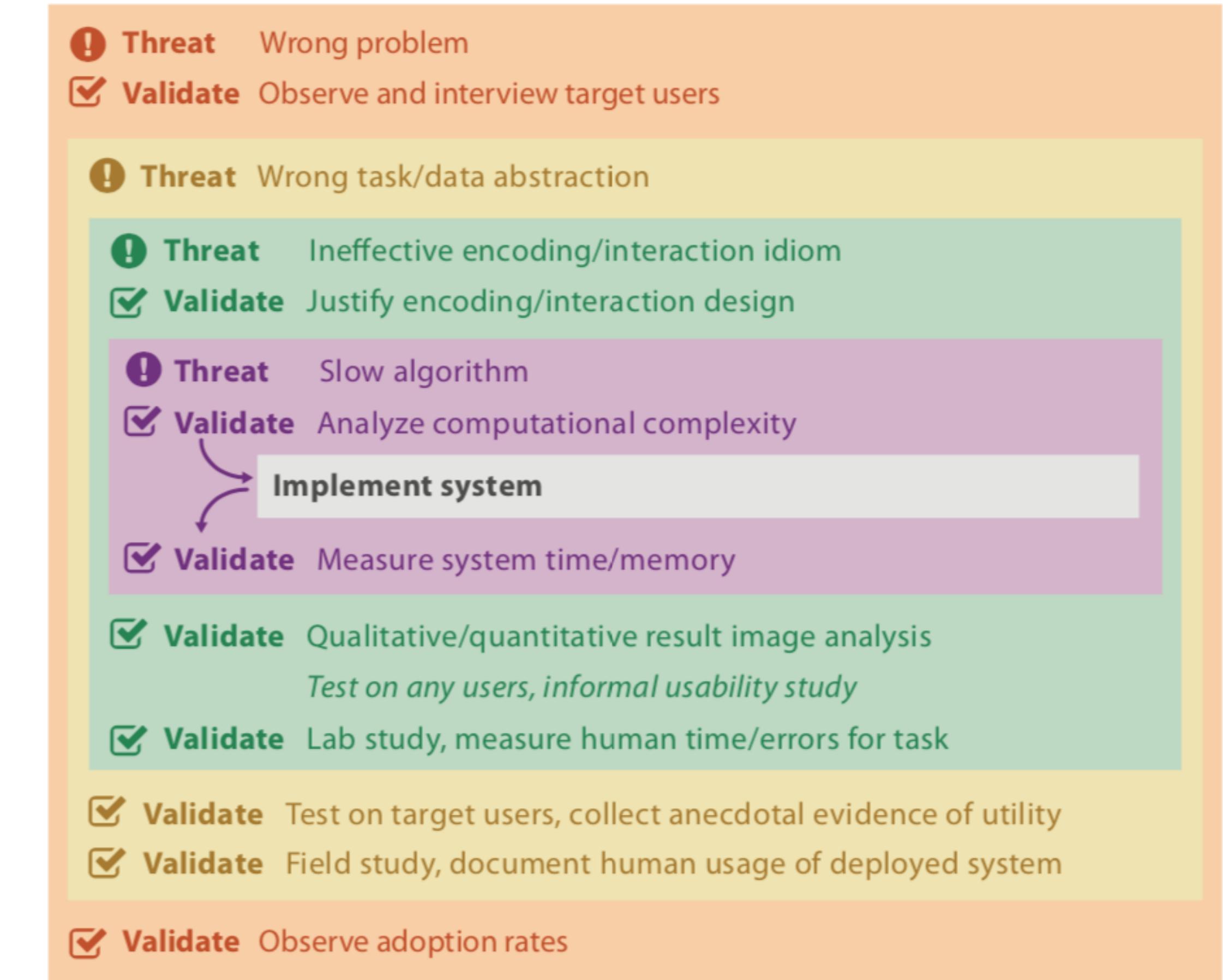
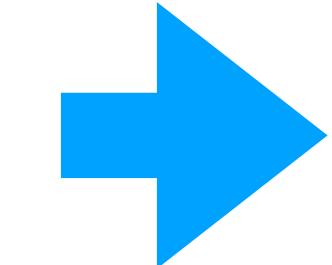
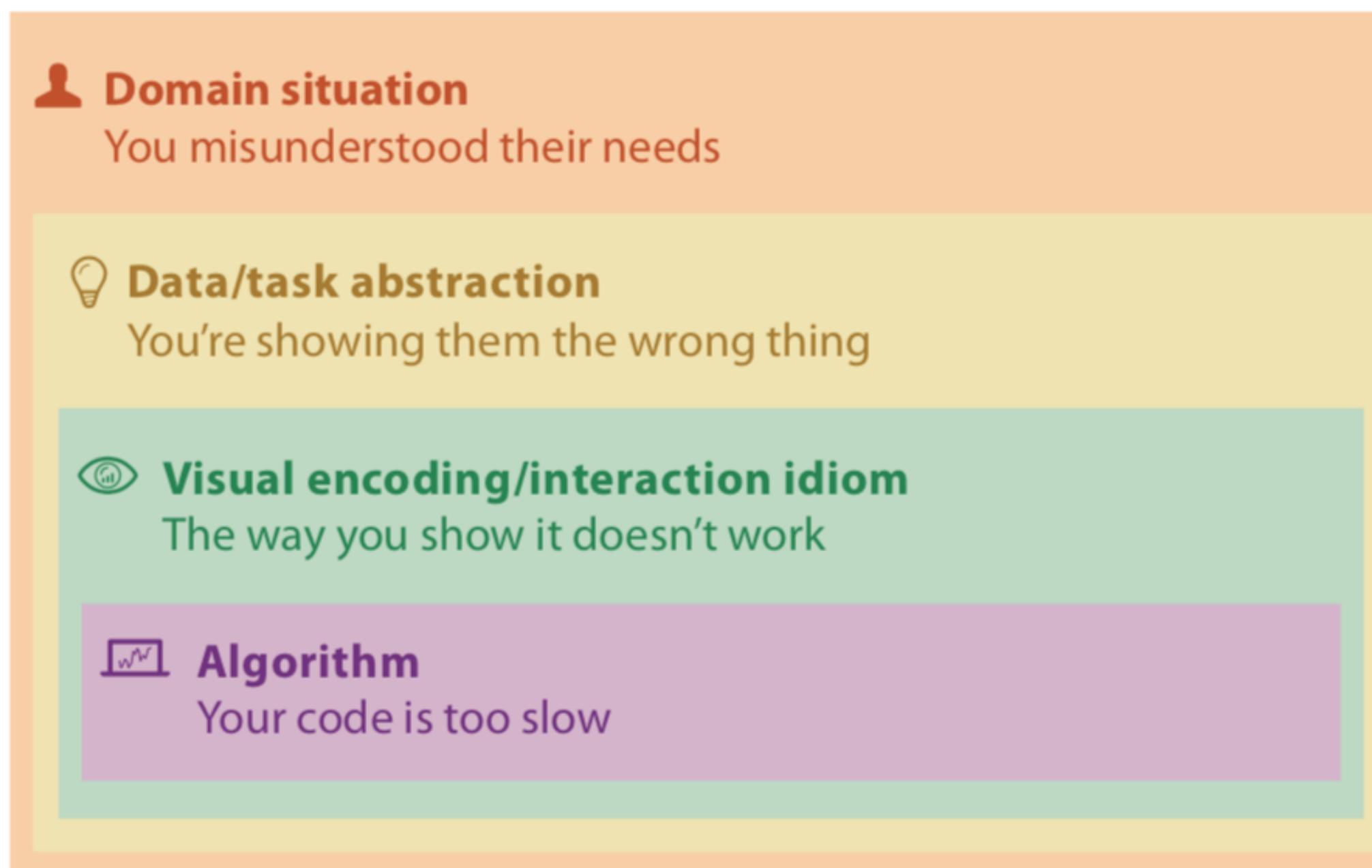
The way you show it doesn't work



## Algorithm

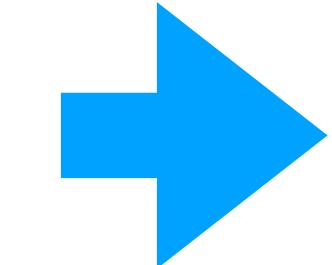
Your code is too slow

# Amenazas y validaciones



# Amenazas y validaciones

- Domain situation**  
You misunderstood their needs
- Data/task abstraction**  
You're showing them the wrong thing
- Visual encoding/interaction idiom**  
The way you show it doesn't work
- Algorithm**  
Your code is too slow



- Threat** Wrong problem  
**Validate** Observe and interview target users
- Threat** Wrong task/data abstraction  
**Threat** Ineffective encoding/interaction idiom  
**Validate** Justify encoding/interaction design
- Threat** Slow algorithm  
**Validate** Analyze computational complexity
  - Implement system**
  - Validate** Measure system time/memory**Validate** Qualitative/quantitative result image analysis  
*Test on any users, informal usability study*  
**Validate** Lab study, measure human time/errors for task
- Validate** Test on target users, collect anecdotal evidence of utility  
**Validate** Field study, document human usage of deployed system
- Validate** Observe adoption rates

# What / Qué?



Figure 2.1. What can be visualized: data, datasets, and attributes.

# Why / por qué?

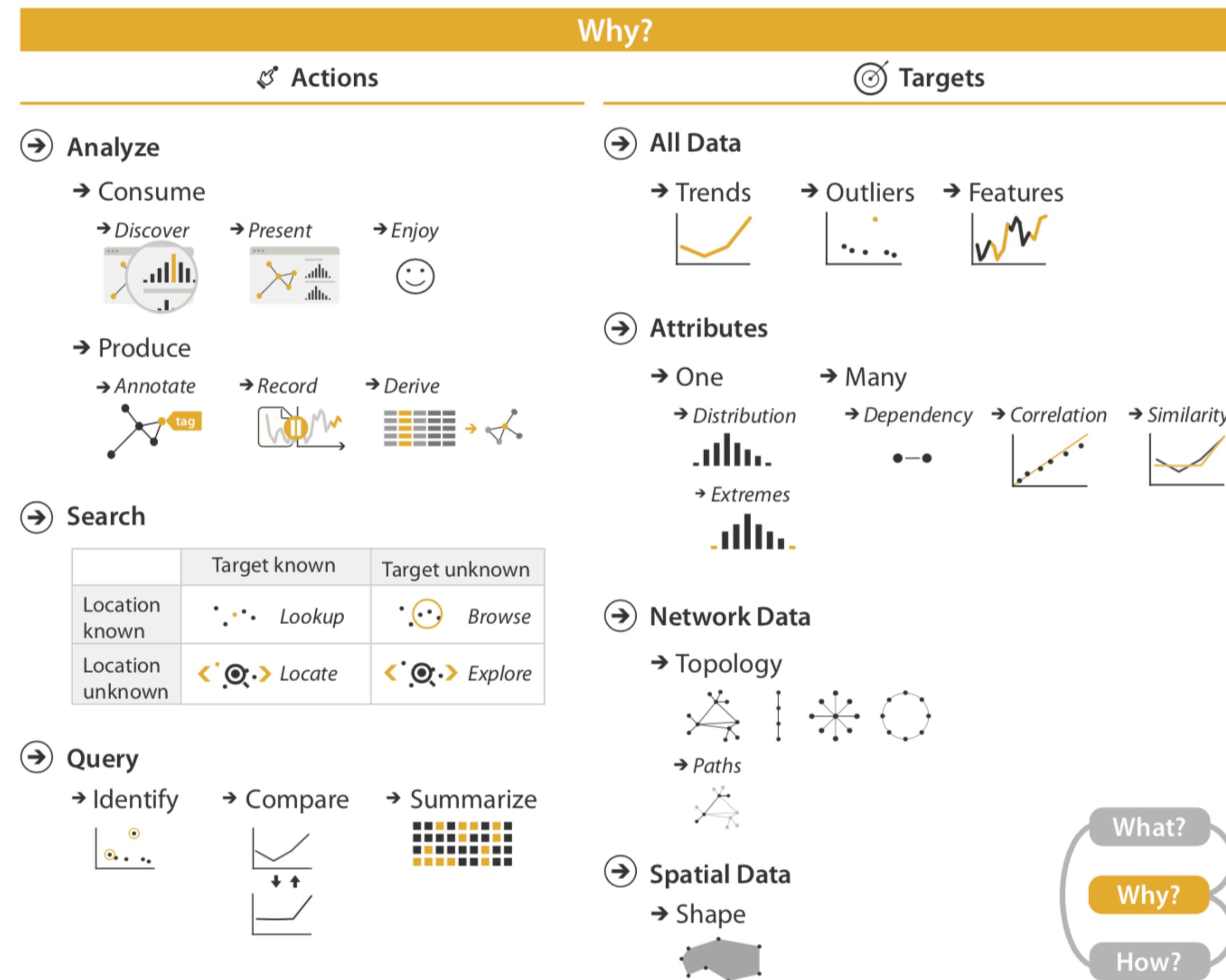
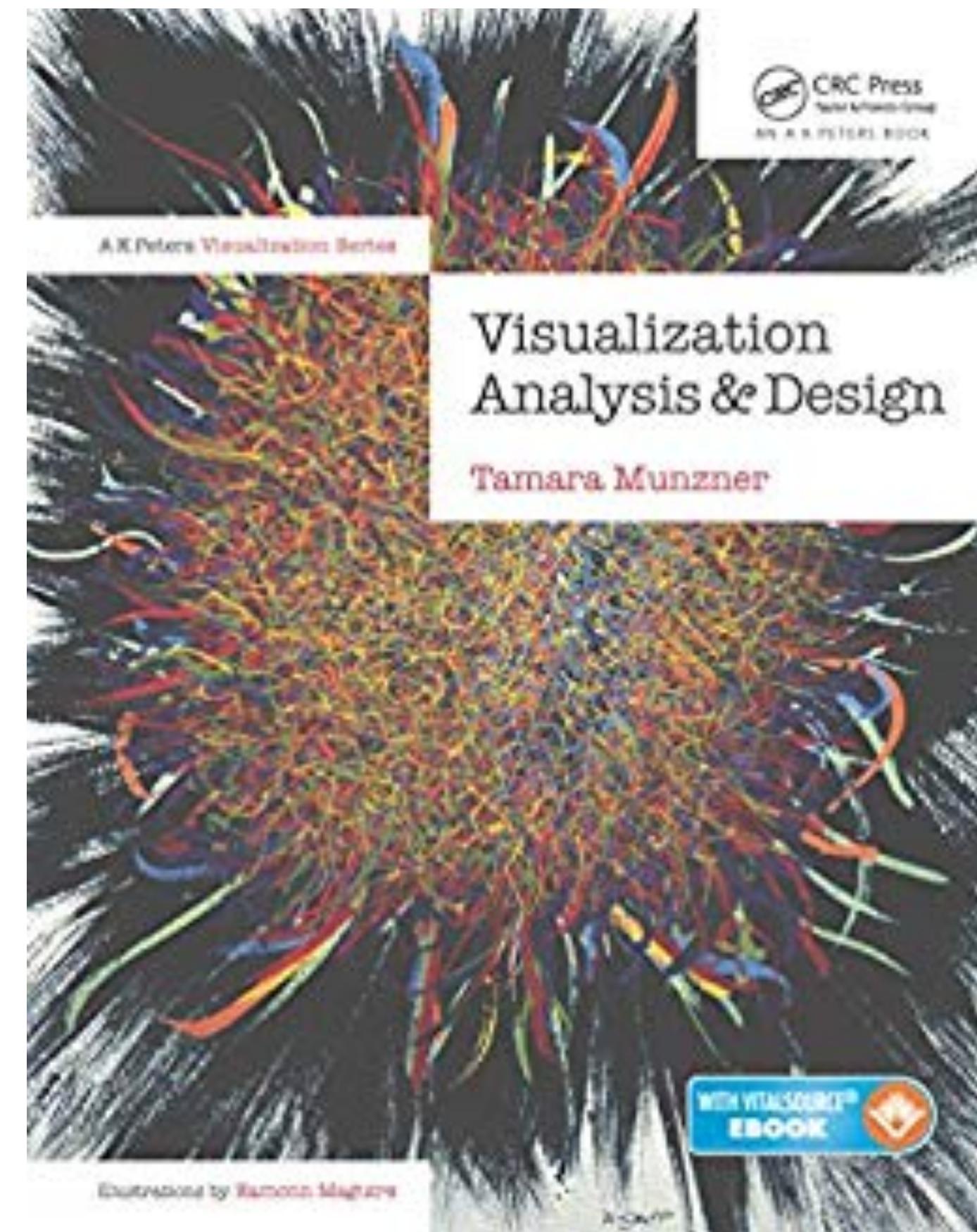


Figure 3.1. Why people are using vis in terms of actions and targets.

# How? Tienen 2 opciones



# How? Tienen 2 opciones



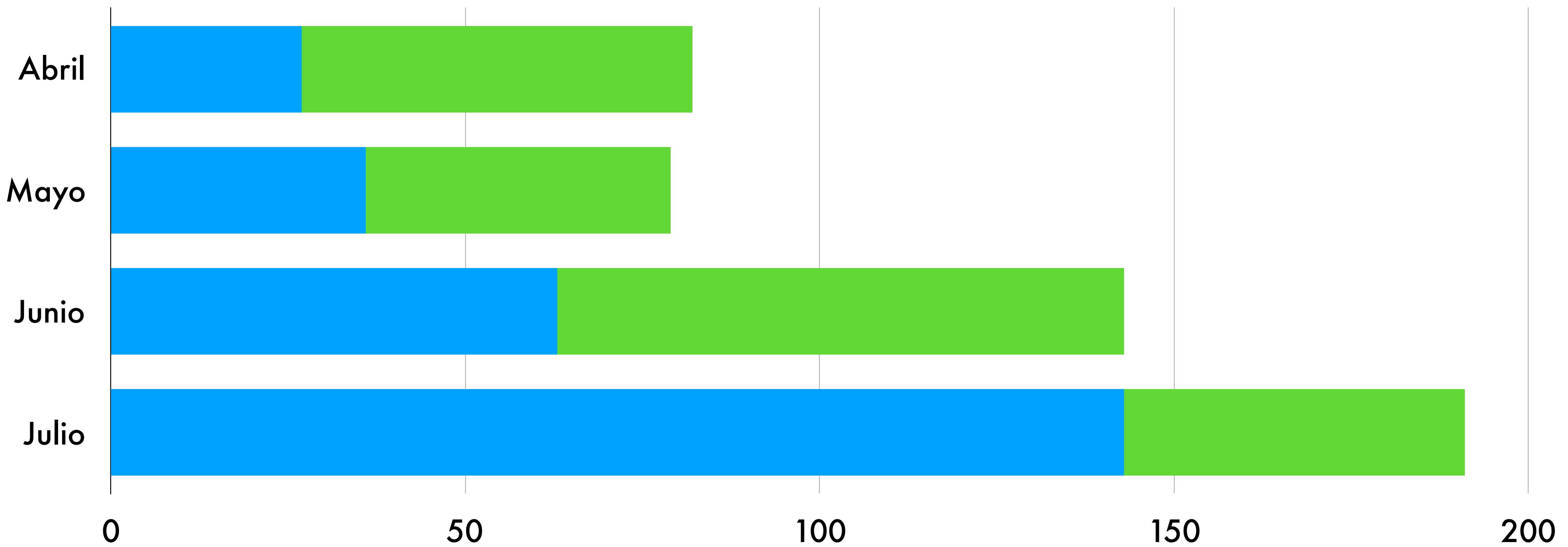
**hagamos un ejercicio!**

# Ejercicio

- **Usuario:** departamento de ventas de Apple
- **What:** Datos de ventas históricos para iPhone y iPad
- **Why?** Para saber cuál se vendió más en un año específico y a su vez ver una tendencia en las ventas de cada uno.
- **How?**

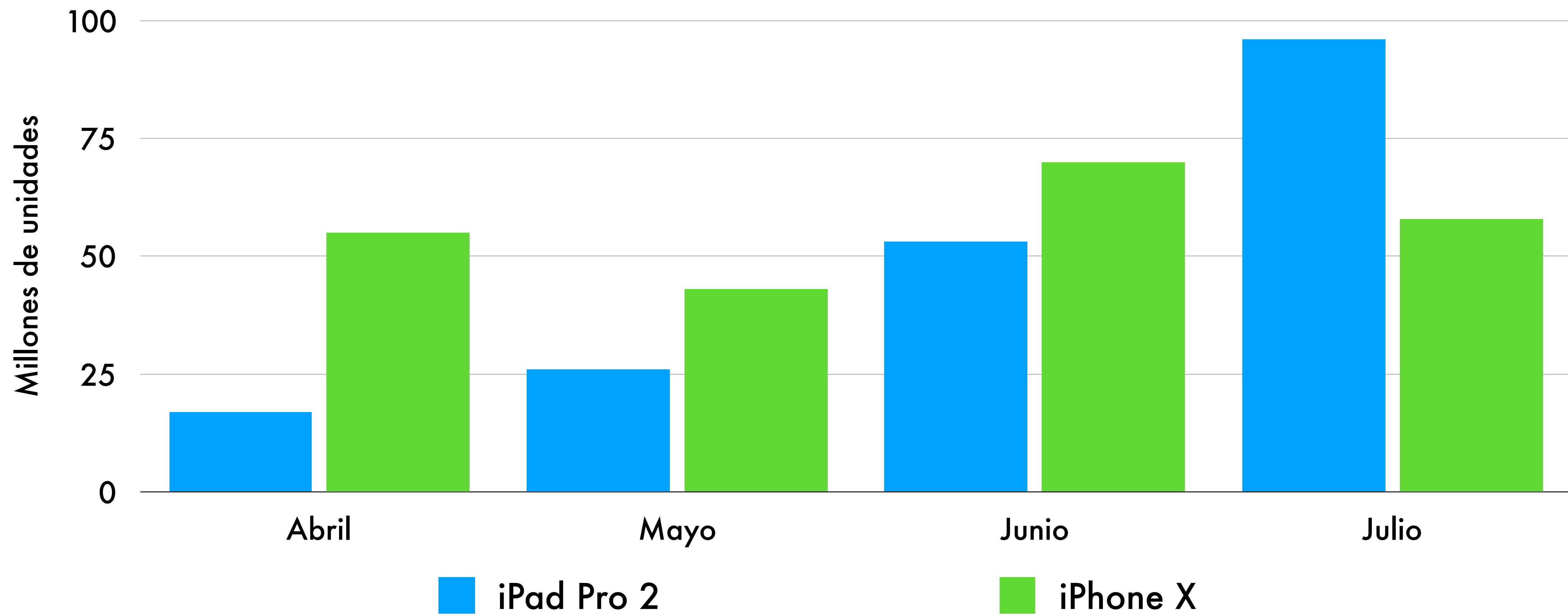
# ¿Cuál es mejor idiom?

Ventas de dispositivos Apple 2018

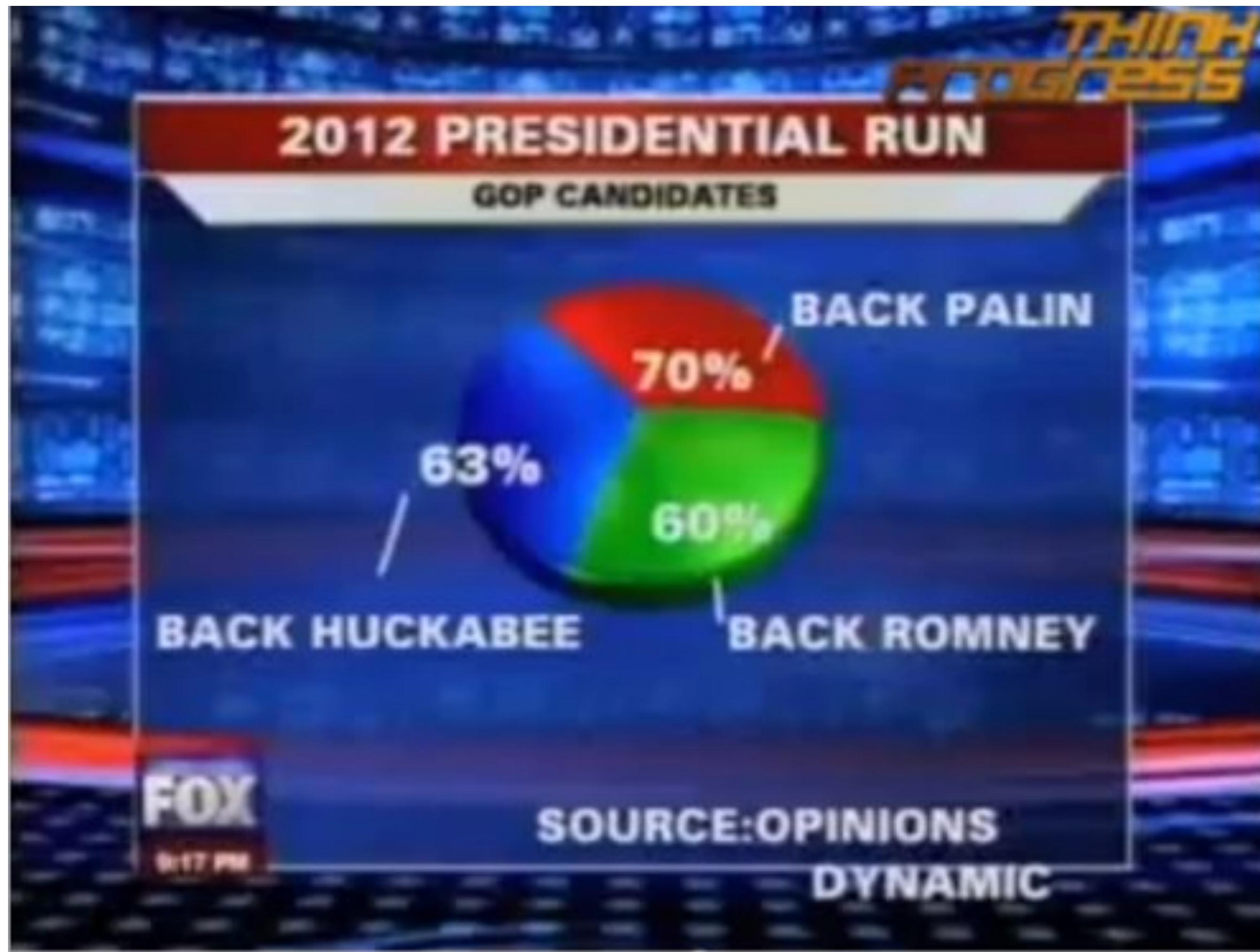


# Cuál es un mejor idiom?

Ventas de dispositivos Apple 2018



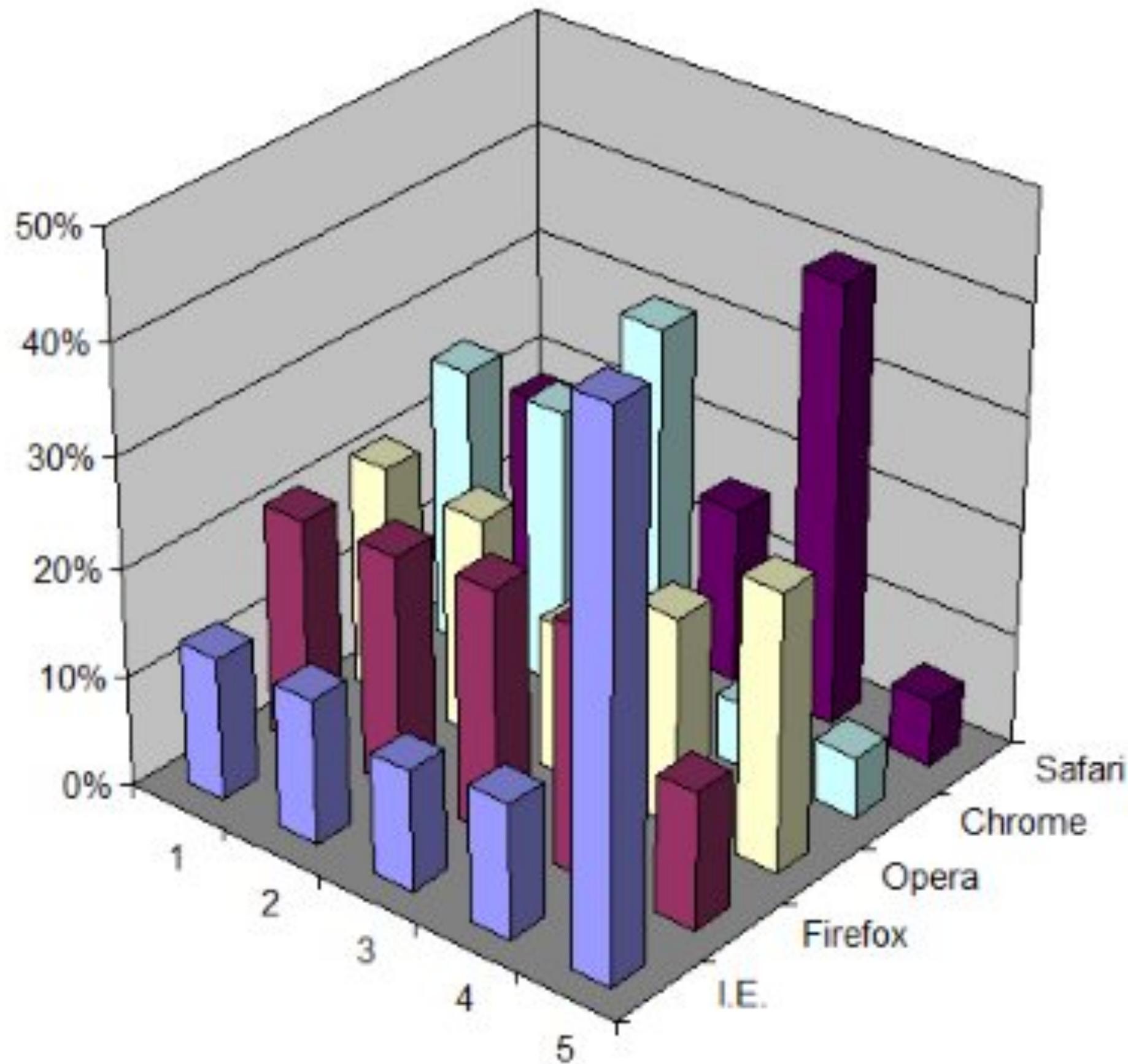
# Ejercicio 2



- Todo suma 193%
- El contraste con lo de atrás es horrible
- Digamos que tiene sentido.... los pedazos de Pie se ven iguales!

Tarea: Identificar % de votantes de cada candidato y poder compararlos entre sí

# Ejercicio 3



- Lo de atrás no se ve
- Los colores son horribles
- Qué es 1, 2, 3, 4, 5?
- Los porcentajes son qué?

**Tarea:** Identificar el % de mercado que cada navegador tuvo de 2011 a 2015 y compararlos entre sí a través del tiempo y a su vez entre navegadores

# Ejercicio 4

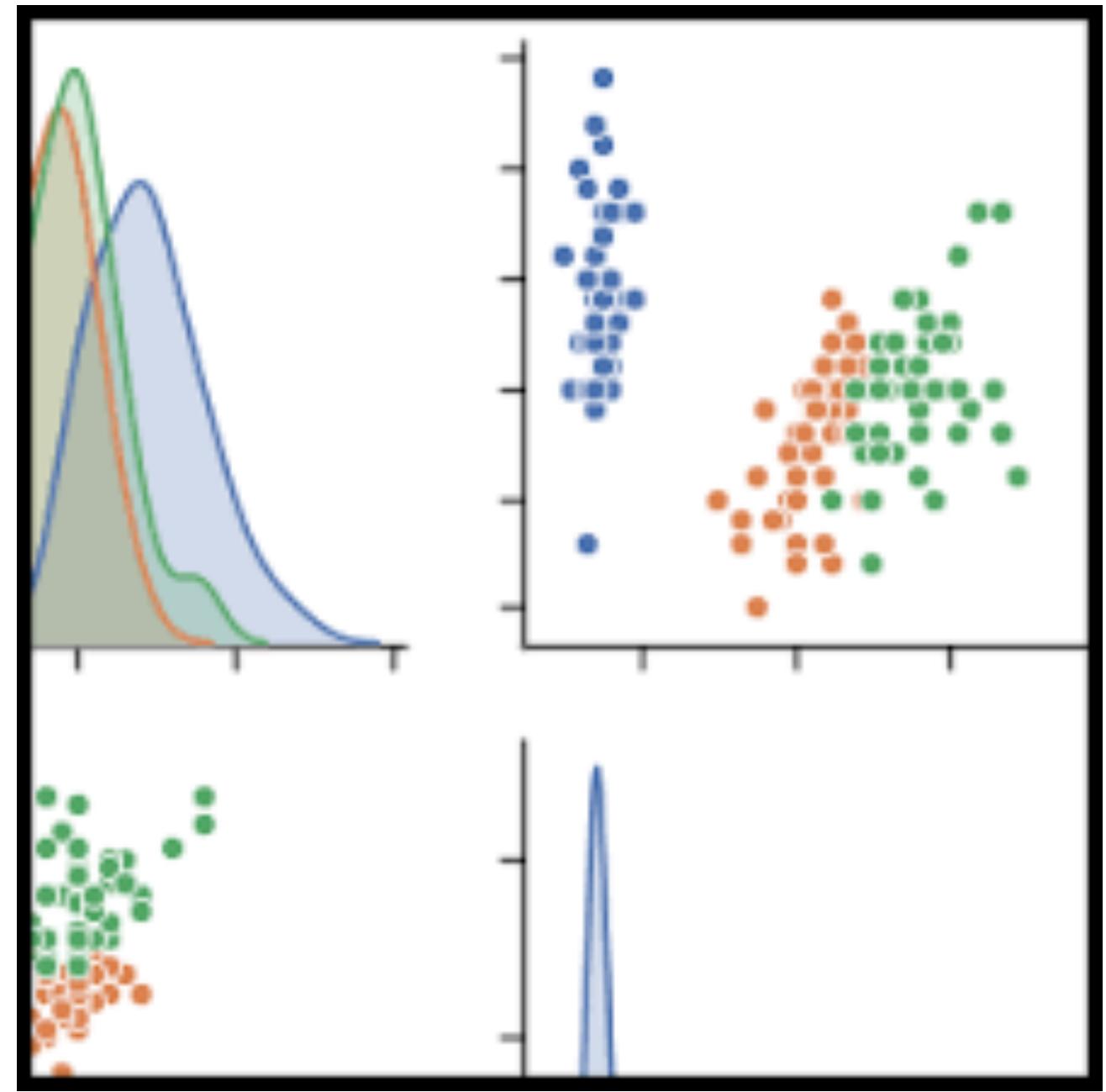


Usando  
mala  
visualización  
para  
ensuciar un  
proyecto  
político

**Tarea: Identificar el número que hace falta para alcanzar la meta de enrollment de marzo 31**

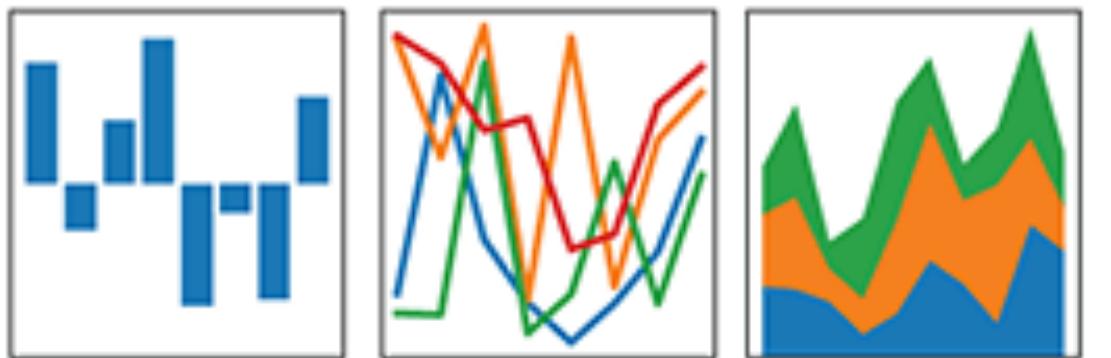
**Tarea: Identificar el % o proporción de personas que hacen falta respecto a la meta**

**Con qué hacemos  
visualizaciones?**



# pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



# bokeh

# matplotlib



**Paso 1 (10 min)**

**Que todos tengamos el ambiente  
necesario para el desarrollo**

# Paso 1 (10 min)

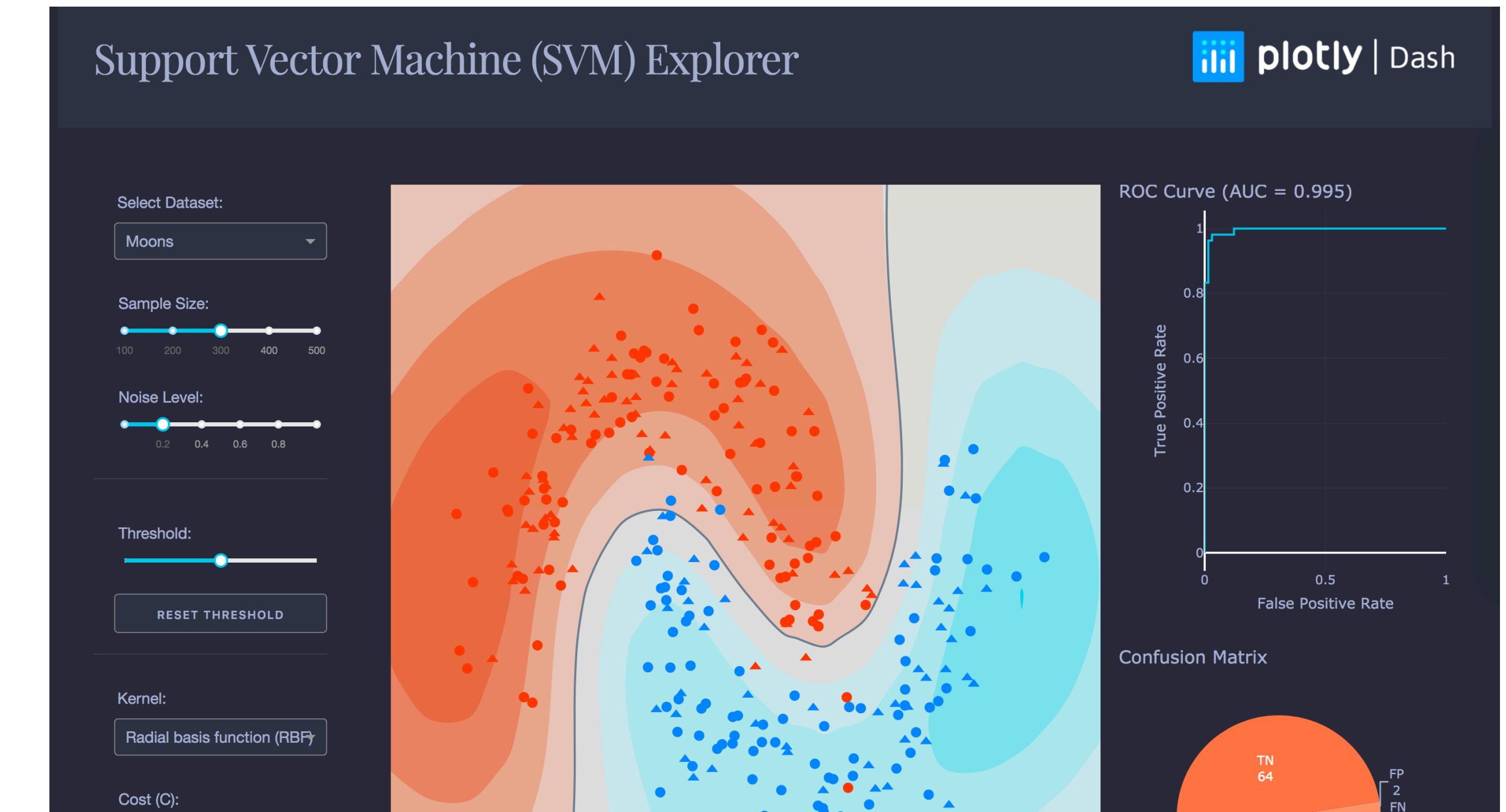
- Crear un ambiente de **Conda**, un pipenv, un virtualenv, poetry, lo que quieran pero que sepan manejar y entender. Si no conocen ninguna de estas palabras lo mejor es que trabajen con alguien que sí y luego en casa las conozcan ;)
- Descarga el siguiente repo en tu compu: <https://github.com/nonameable/basic-example> (si no sabes Git, descarga el repo siguiendo estas instrucciones)
- Instala todas las dependencias siguiendo las indicaciones del README.md que se encuentra en el repo.
- Correr el ejemplo básico y jugar con él. Si tienen problemas, levanten la mano y alguien les ayudará.

# Introducción a Dash

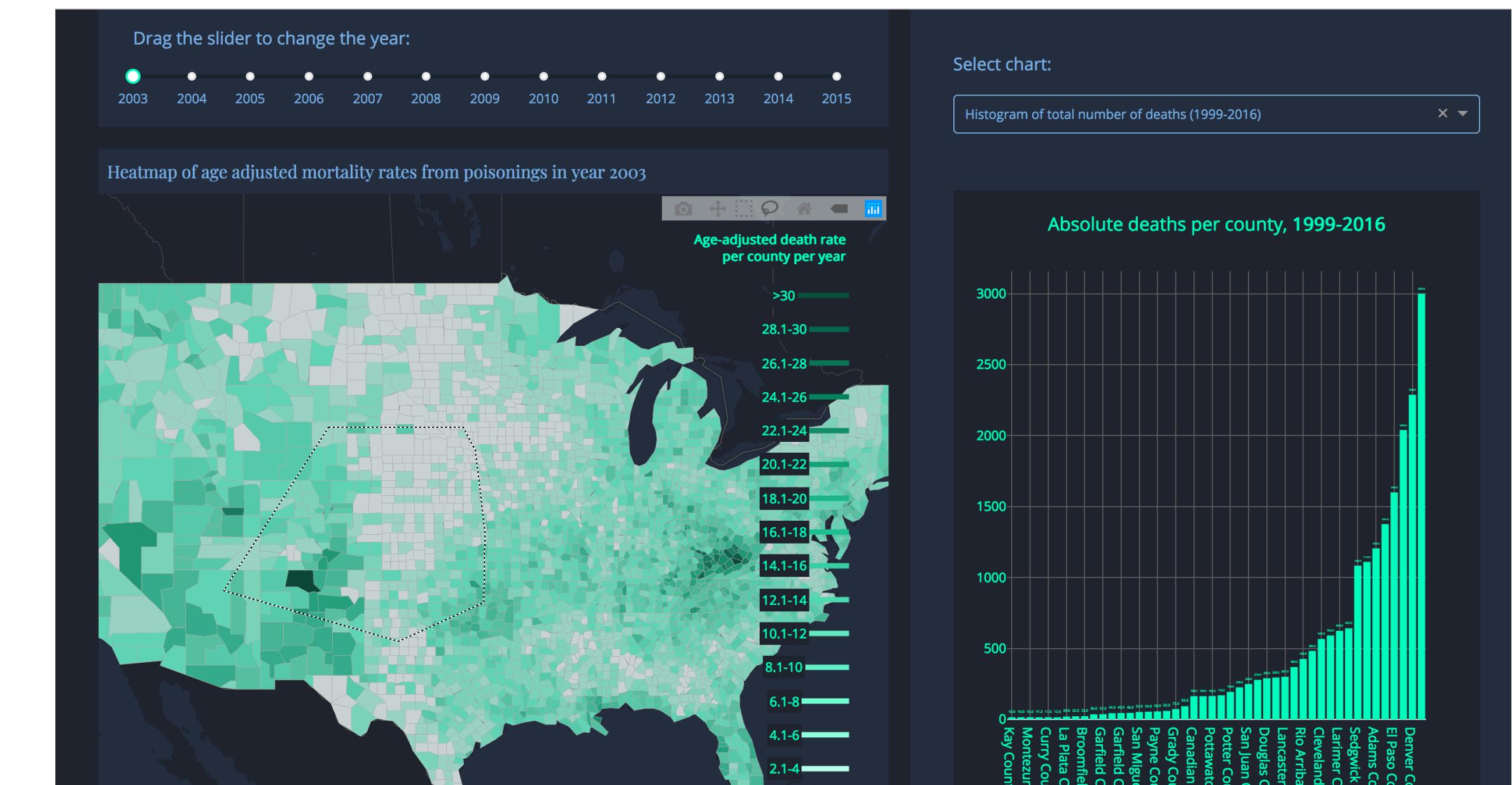
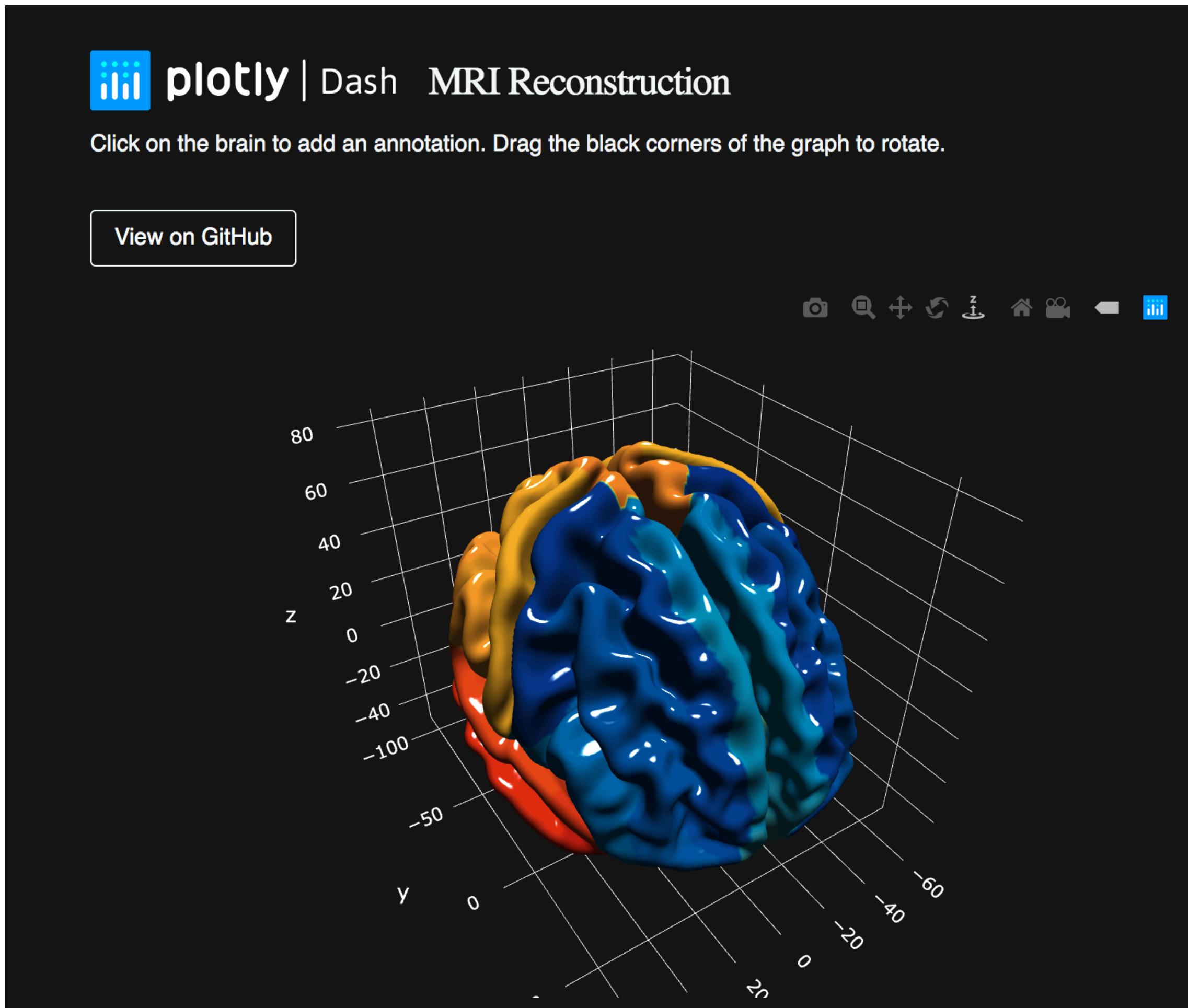
by Plotly (10 min)

# ¿Qué es Dash?

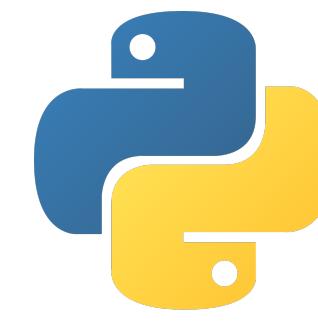
**Dash es un framework (código listo para usar) para Python (y recientemente R) que permite construir aplicaciones web analíticas hermosas sin requerir Javascript (aunque si saben HTML y CSS, es mucho mejor)**



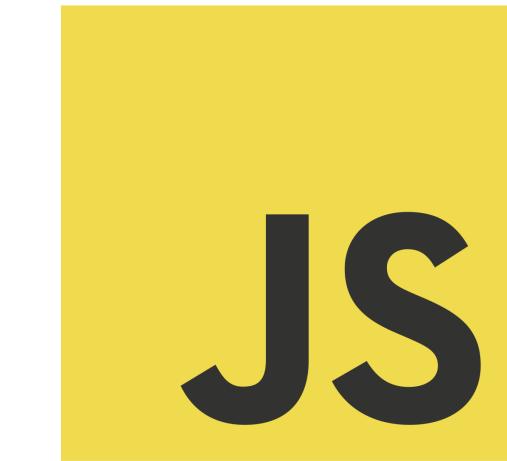
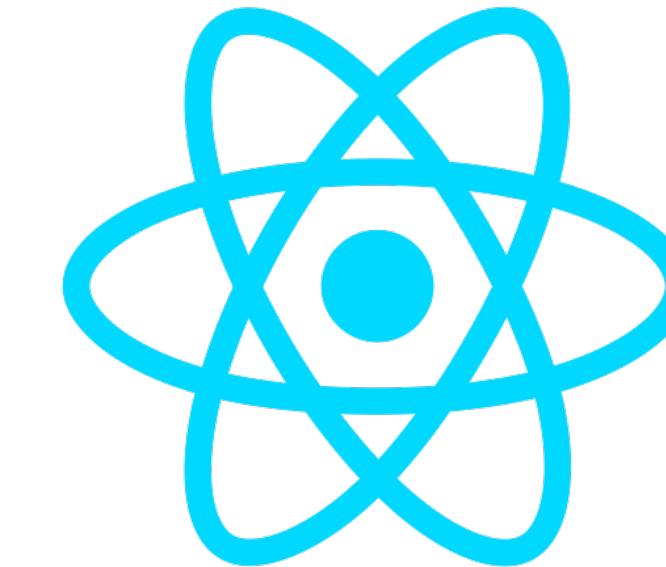
# Ejemplos



# ¿Cómo funciona?

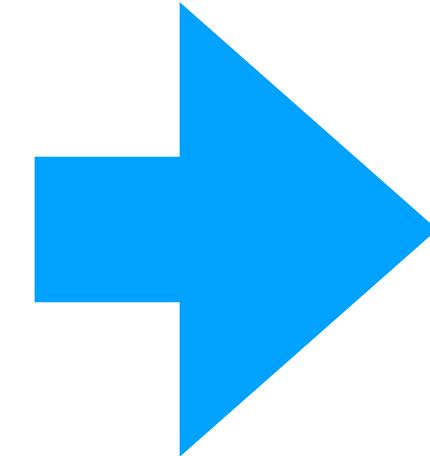


Flask



```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '%s [%label=%s]' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s';' % ast[1]
        else:
            print ''
    else:
        print ']';
else:
    print '';
    children = []
    for n, child in enumerate(ast[1:]):
        children.append(dotwrite(child))
    print '%s -> {' % nodename,
    for name in children:
        print '%s' % name,
```



```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Example</title>
5          <link rel="stylesheet" href="sty
6      </head>
7      <body>
8          <h1>
9              <a href="/">Header</a>
10         </h1>
11         <nav>
12             <a href="one/">One</a>
13             <a href="two/">Two</a>
14             <a href="three/">Three</a>
15         </nav>
```

# Más allá de matplotlib y reportes PDF

creación de dashboards dinámicos con Dash de Plotly y Python



Dash



# ¿Recuerdan el paso 1?

- Crear un ambiente de **Conda**, un pipenv, un virtualenv, poetry, lo que quieran pero que sepan manejar y entender. Si no conocen ninguna de estas palabras lo mejor es que trabajen con alguien que sí y luego en casa las conozcan ;)
- Descarga el siguiente repo en tu compu: <https://github.com/nonameable/basic-example> (si no sabes Git, descarga el repo siguiendo estas instrucciones)
- Instala todas las dependencias siguiendo las indicaciones del README.md que se encuentra en el repo.
- Correr el ejemplo básico y jugar con él. Si tienen problemas, levanten la mano y alguien les ayudará.

# Ahora sí, vamos a darle (25 min)

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos

Si ves esto y no estuviste en la presentación, mi recomendación es leer la [Starting guide de Dash](#)

# Código!

# Ahora sí, vamos a darle

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos

Si ves esto y no estuviste en la presentación, mi recomendación es leer la [Starting guide de Dash](#)

# Código!

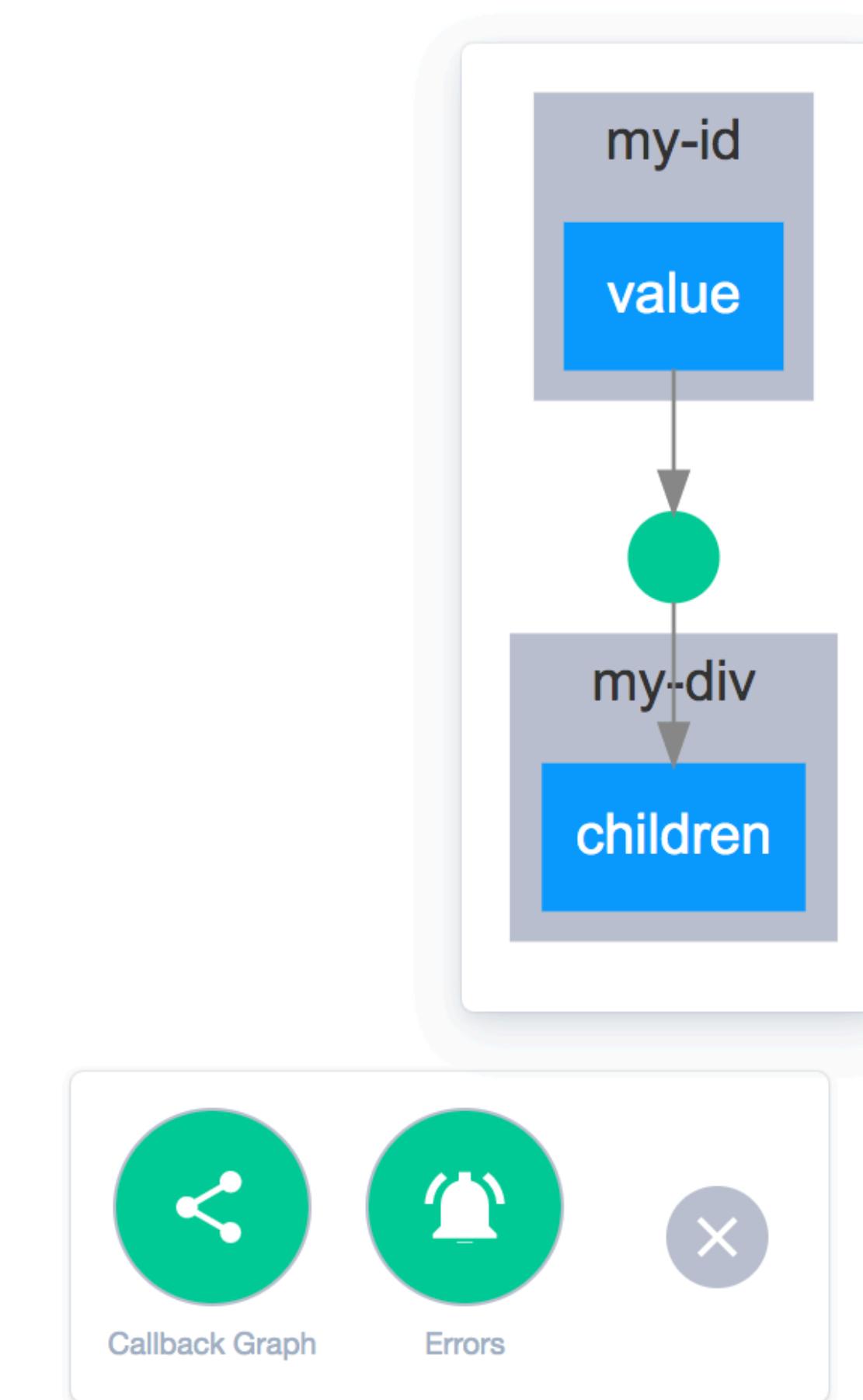
# Ahora sí, vamos a darle

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos

Si ves esto y no estuviste en la presentación, mi recomendación es leer la [Starting guide de Dash](#)

# Código!

# Aprovechen el callback graph!



# Ahora sí, vamos a darle

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos

Si ves esto y no estuviste en la presentación, mi recomendación es leer la [Starting guide de Dash](#)

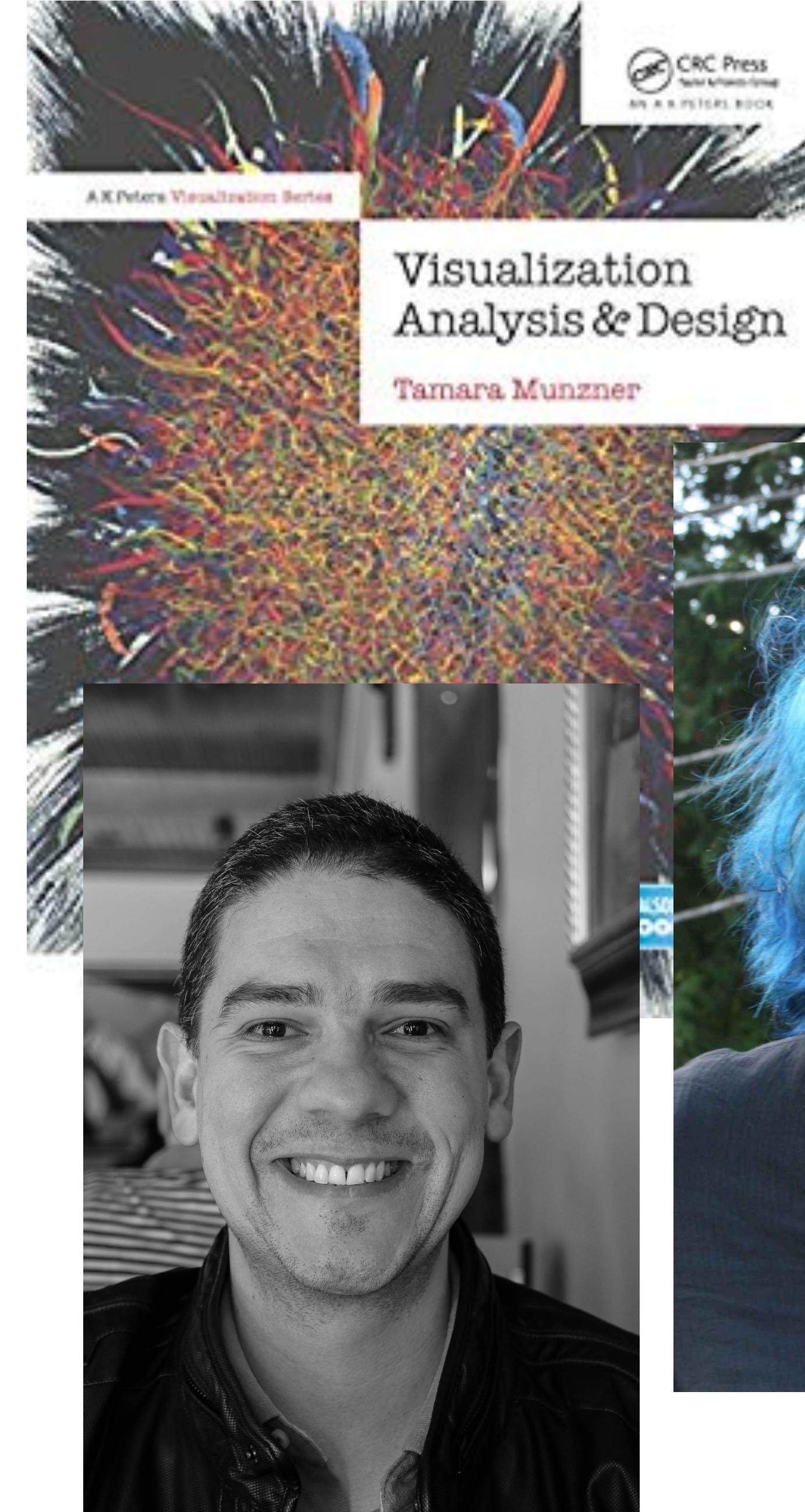
# Estructura de una app (sencilla)

1. Imports
2. variables y funciones a usar
3. Declaración de app
4. Layout
5. Callbacks
6. Que la app corra carajo!

# **Conexión Plotly - Framework para visualización efectiva**

# Es aquí donde implementamos el Framework

1. What, why, how
2. El How nos dice cuál idioma escoger.
3. Buscamos ese idiom en Plotly o lo hacemos nosotros mismos!



# ¿Cómo escogemos?

1. Graphing Libraries: <https://plot.ly/python/>
2. Copiando de otros cracks como la [Dash Gallery](#) y [Awesome Dash](#)
3. Usando [Plotly Dash Studio](#)

Vamos por nuestra primera app  
grande (55 min)

# Démole un poco más

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos
4. Más y más callbacks
5. Gráficos y tablas(aka cómo usar Plotly desde Python)
6. Integración con Bootstrap

# Démole un poco más

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos
4. Más y más callbacks
5. Gráficos y tablas (aka cómo usar Plotly desde Python)
6. Integración con Bootstrap

# Démole un poco más

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos
4. Más y más callbacks
5. Gráficos (aka cómo usar Plotly desde Python)
6. Integración con Bootstrap

# Démole un poco más

1. Importar Dash
2. Layouts (aka cómo trabajar con HTML y CSS)
3. Callbacks básicos
4. Más y más callbacks
5. Gráficos (aka cómo usar Plotly desde Python)
6. Integración con Bootstrap

# **Extras y Futuro (20 min)**

**aka “La ñapa si queda tiempo”**

# Despliegue en “producción”

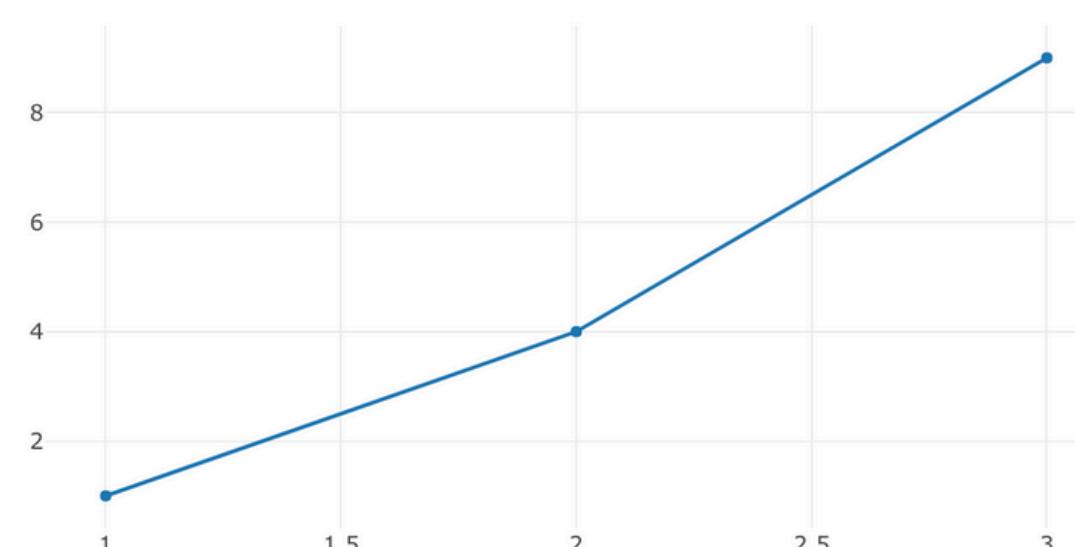


# Mejorar apariencia de apps

## Standard row and column-based layouts

Demo

Menu ▾ Link

| Heading  | Graph  |   |   |   |   |   |   |   |   |
|--|--|---|---|---|---|---|---|---|---|
| <p>Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui. Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.</p> <p><a href="#">View details</a></p> |  <table border="1"><thead><tr><th>x</th><th>y</th></tr></thead><tbody><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>4</td></tr><tr><td>3</td><td>6</td></tr></tbody></table> | x | y | 1 | 2 | 2 | 4 | 3 | 6 |
| x  | y  |   |   |   |   |   |   |   |   |
| 1  | 2  |   |   |   |   |   |   |   |   |
| 2  | 4  |   |   |   |   |   |   |   |   |
| 3  | 6  |   |   |   |   |   |   |   |   |

## Cards

Header

This card has a header

And some text in the body



Card with image

This card has an image

Here's another card

With some text, and a button

[Click here](#)

Here's yet another card

With some text, and some links

[External](#) [Internal](#)

# El futuro de Dash

1. Mejoras en el API de tablas
2. Integraciones más directas con Flask y sus componentes
3. Lo que la comunidad vaya pidiendo
4. Mejoras en el manejo de grandes volúmenes de datos
5. Dash Database para sesiones
6. Ustedes!



**“Para atrás ni para tomar impulso”**

*– Bernardo Fandiño Lizarraga*