

Programación básica con Python

Taller para el análisis y visualización de
datos básico con Python



frani.be



frani.be

~ Estructura

1. Presentación personal
2. ¿Qué lograremos hoy?
3. Conceptos iniciales
4. Introducción a Python
5. Análisis de datos con Pandas y visualización con Altair
6. ¡Manos al código!



frani.be

Presentación personal

~ Francisca B. Medina Concha

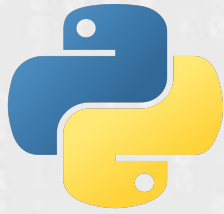
- Team Leader de equipo DataViz en LATAM Airlines





frani.be

¿Qué lograremos hoy?



python™



frani.be

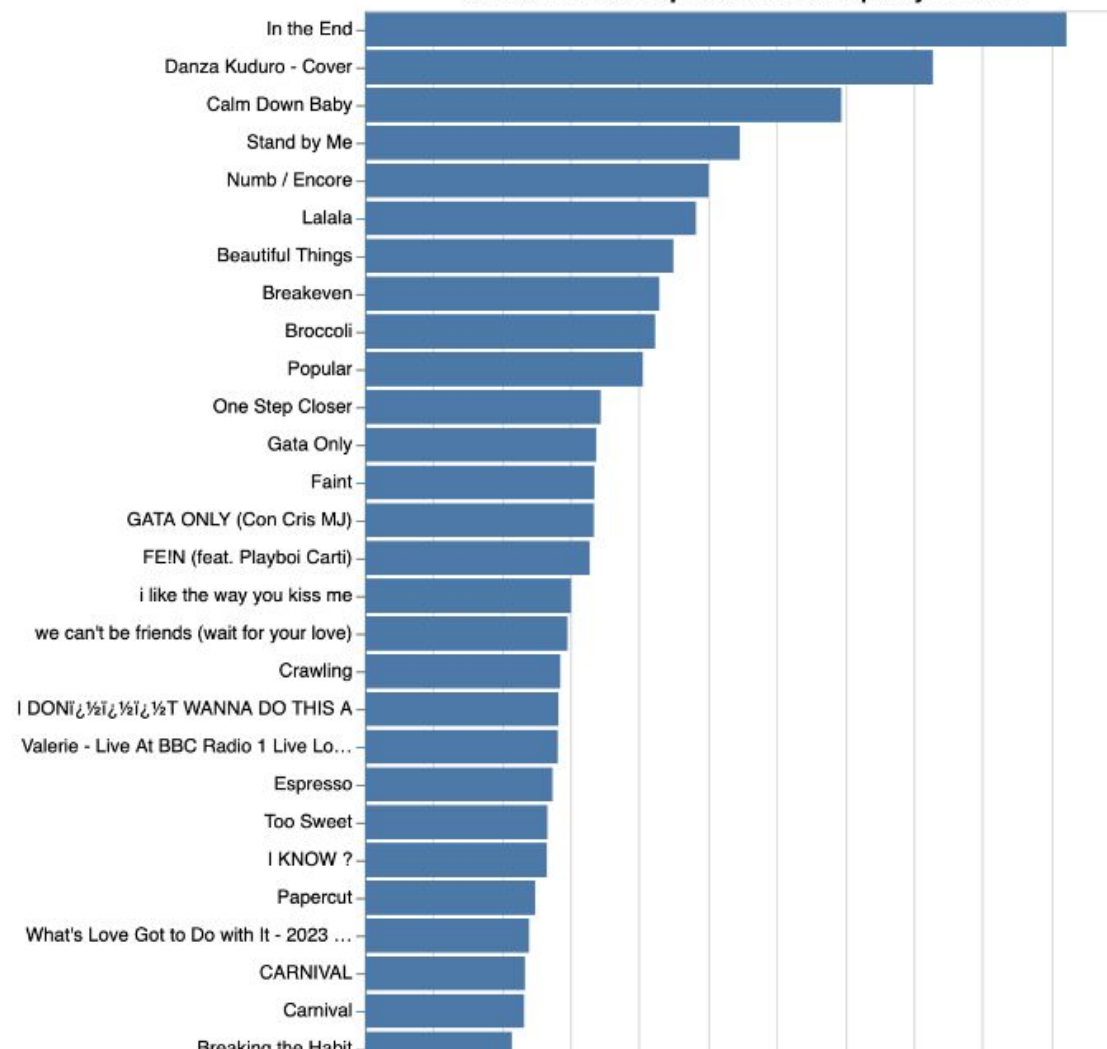
kaggle

Google
colab

 pandas

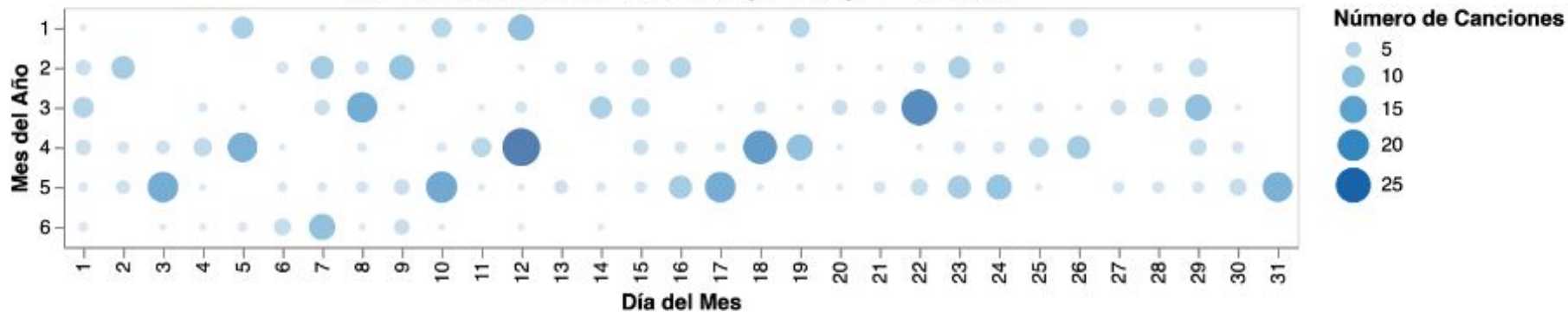


Canciones más reproducidas en Spotify en 2024

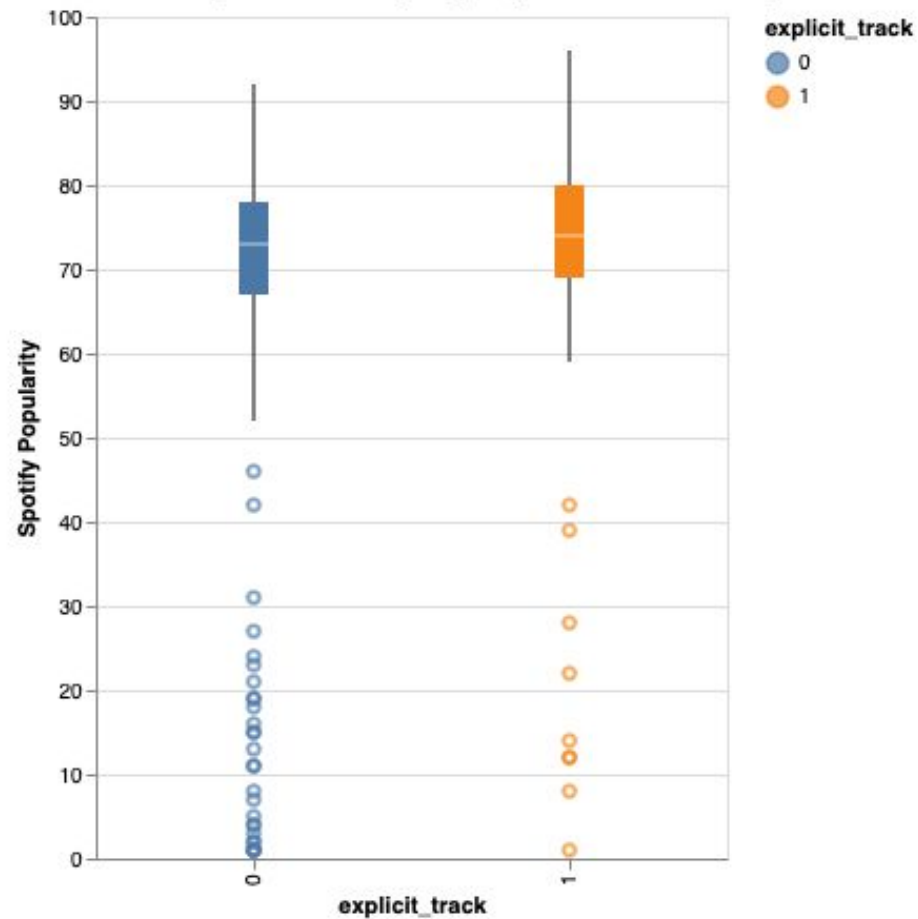


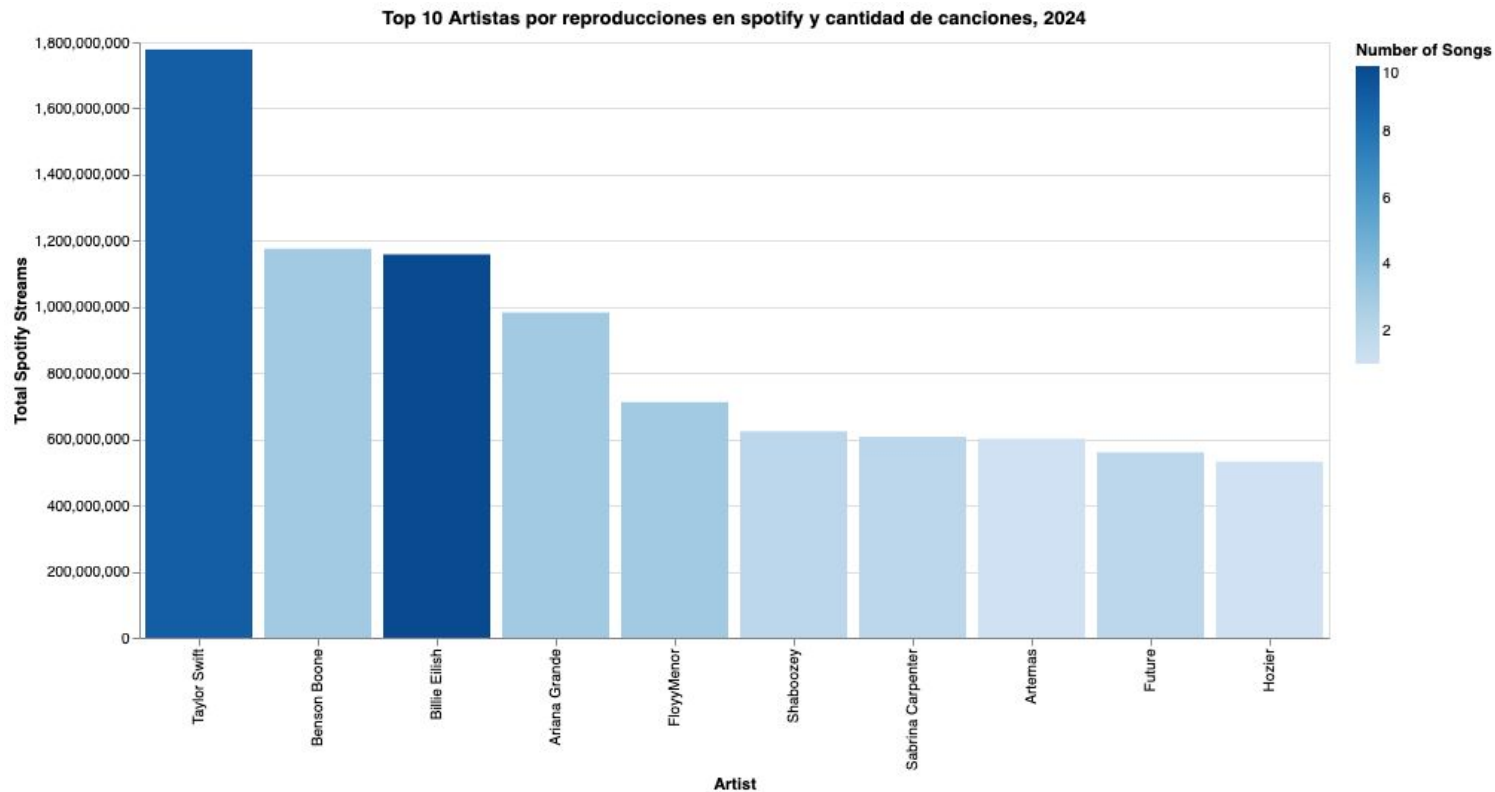
frani.be

Número de Canciones Lanzadas por Día y Mes en 2024



Distribución de Popularidad en Spotify según el Contenido Explícito

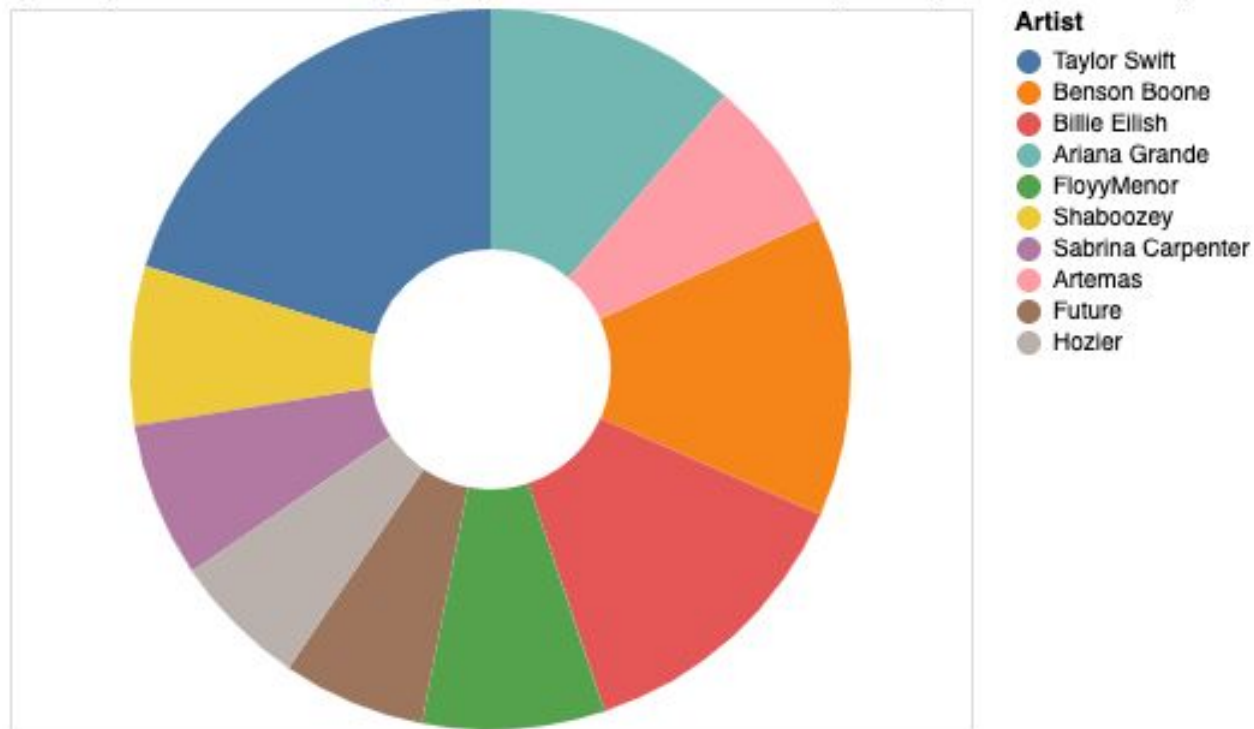






frani.be

Top 10 Artistas por reproducciones en spotify y cantidad de canciones, 2024 (Gráfico de Dona)





frani.be

Conceptos iniciales

~ ¿Qué es la programación?



frani.be



~ ¿Qué es la programación?



frani.be

Indicar al dispositivo computacional lo que tiene que hacer,
mediante **algoritmos**



frani.be

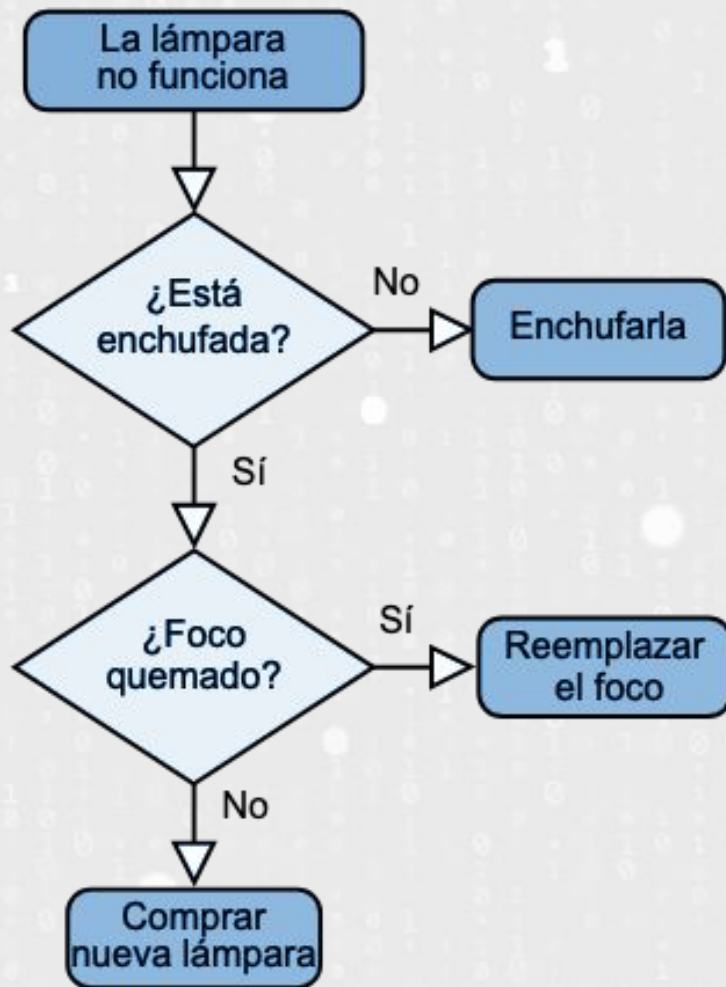
~ Concepto de algoritmo

Conjunto ordenado y finito de **operaciones** que permite hallar la solución de un problema.

Se describe en tres partes básicas: **entrada** (escritura), **proceso** (ejecución) y **salida** (resultado).



frani.be



~Lenguaje de programación



frani.be

Es una **herramienta traductora** de un **lenguaje fuente** basado en algoritmos, a un **lenguaje máquina** comprensible por el computador, para la realización de un determinado proceso.



Swift



Y muchos, MUCHOS más...

~ Librería



frani.be

Una librería es una **colección preescrita de código** que los desarrolladores pueden utilizar para realizar tareas comunes o repetitivas. En lugar de escribir código desde cero para realizar una función específica, un desarrollador puede usar una función predefinida desde una librería.



frani.be

~ Análisis de datos

El análisis de datos se refiere al **proceso de inspeccionar, limpiar, transformar y modelar datos** con el objetivo de descubrir información útil, llegar a conclusiones y apoyar la toma de decisiones. Utiliza técnicas cuantitativas y cualitativas para entender patrones, relaciones y estructuras dentro de los datos.



frani.be

~ ¿Qué son los datos?

En la vida cotidiana usamos la palabra "**datos**" de manera general como sinónimo de "**información**", pero debemos comenzar a entenderlo como un **concepto más específico** para este taller.



frani.be

~ ¿Qué son los datos?

- ➔ Se refiere a hechos, eventos, transacciones, entre otros, que han sido registrados.
- ➔ Es la unidad más pequeña de información ***fáctica**.
- ➔ Puede ser empleado para el cálculo, el razonamiento, la discusión, la inferencia, la deducción, y su presentación o visualización.

Fáctico: Alude a aquello vinculado a los hechos. El término suele usarse para calificar a lo que depende de los hechos o se fundamenta en ellos



frani.be

~Diferencia entre datos e información

Datos

Entrada sin procesar de la cual se produce la información.

Información

Datos procesados y comunicados para ser interpretados por el receptor.

Un dato por sí mismo no constituye información, es el procesamiento de los datos lo que nos proporciona información.

~Diferencia entre datos e información



frani.be



~ Herramientas para el análisis de datos



frani.be

- **Excel:** Una herramienta básica pero poderosa para la manipulación y visualización de datos.
- **Python (especialmente con bibliotecas como Pandas):** Lenguaje de programación versátil para análisis de datos y aprendizaje automático.
- **R:** Lenguaje de programación diseñado específicamente para análisis estadístico.
- **Tableau:** Herramienta de visualización de datos para transformar datos crudos en gráficos interactivos.
- **SQL:** Lenguaje de consulta estructurado para gestionar y recuperar datos de bases de datos relacionales.
- **Power BI:** Herramienta de Microsoft para visualización y análisis de datos.



frani.be

~ Visualización de datos

La visualización de datos es el acto de **representar información en forma gráfica**. Permite a los usuarios ver y comprender tendencias, patrones y correlaciones que podrían no ser evidentes en los datos brutos. La visualización eficiente de datos puede ayudar a simplificar complejas relaciones de datos y facilitar su interpretación.



frani.be

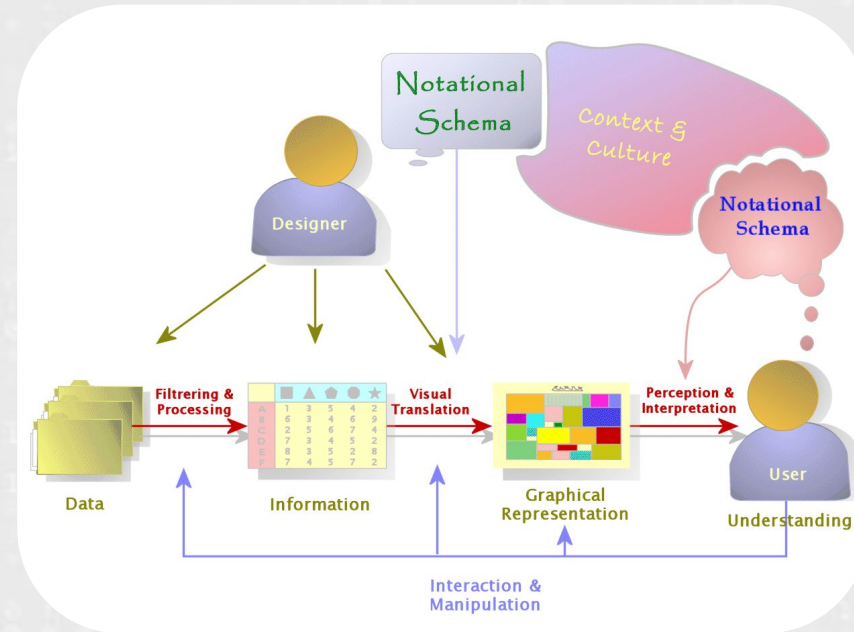
~ Importancia

- Comprensión rápida
- Descubrimiento de patrones
- Facilita la toma de decisiones
- Comunicación efectiva

~ Diagrama InfoVis

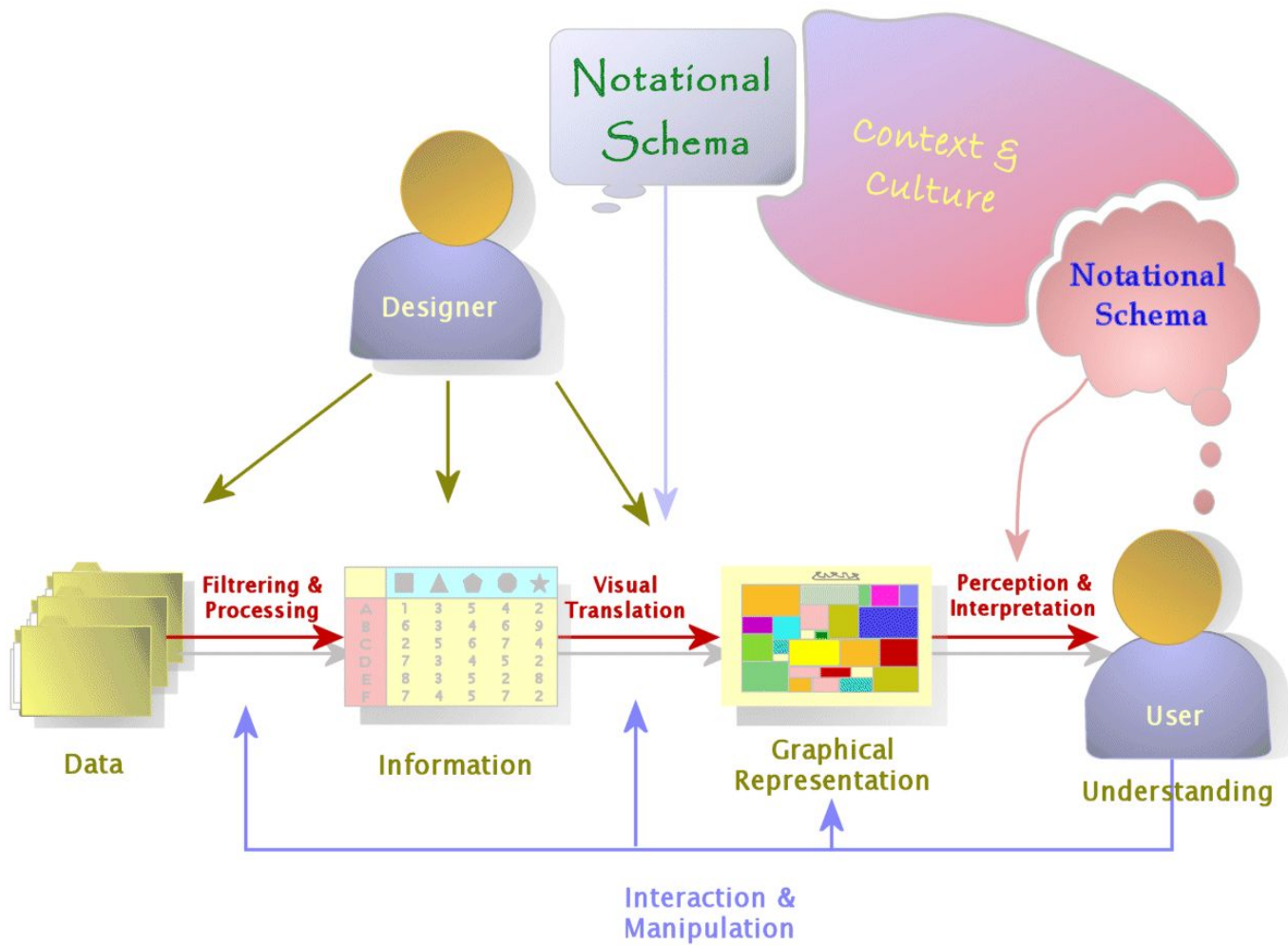


frani.be



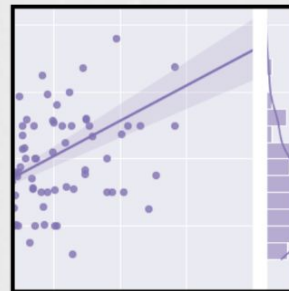
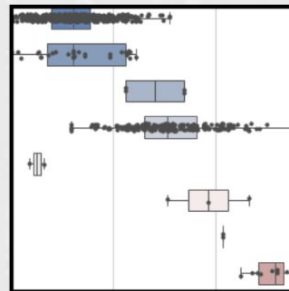
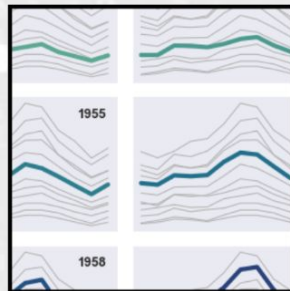
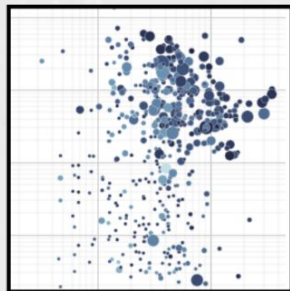
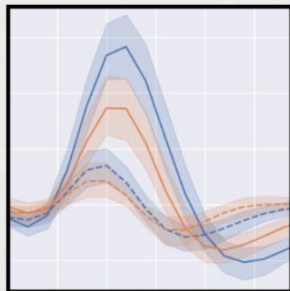
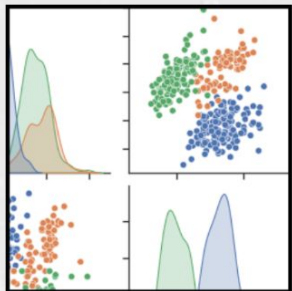


frani.be





frani.be



~ Una efectiva visualización de datos es...



frani.be

- ➔ Simple
- ➔ Con uso adecuado de colores
- ➔ Consistente



frani.be

~ Cuarteto Anscombe

	I		II		III		IV	
	x	y	x	y	x	y	x	y
	10,0	8,04	10,00	9,14	10,00	7,46	8,00	6,58
	8,0	6,95	8,00	8,14	8,00	6,77	8,00	5,76
	13,0	7,58	13,00	8,74	13,00	12,74	8,00	7,71
	9,0	8,81	9,00	8,77	9,00	7,11	8,00	8,84
	11,0	8,33	11,00	9,26	11,00	7,81	8,00	8,47
	14,0	9,96	14,00	8,1	14,00	8,84	8,00	7,04
	6,0	7,24	6,00	6,13	6,00	6,08	8,00	5,25
	4,0	4,26	4,00	3,1	4,00	5,39	19,00	12,50
	12,0	10,84	12,00	9,13	12,00	8,15	8,00	5,56
	7,0	4,82	7,00	7,26	7,00	6,42	8,00	7,91
	5,0	5,68	5,00	4,74	5,00	5,73	8,00	6,89
media	9,0	7,50	9,0	7,50	9,0	7,50	9,0	7,50
std	3,3	2,03	3,3	2,03	3,3	2,03	3,3	2,03
varianza	11,0	4,13	11,0	4,13	11,0	4,12	11,0	4,12

https://es.wikipedia.org/wiki/Cuarteto_de_Anscombe



frani.be

~ Cuarteto Anscombe

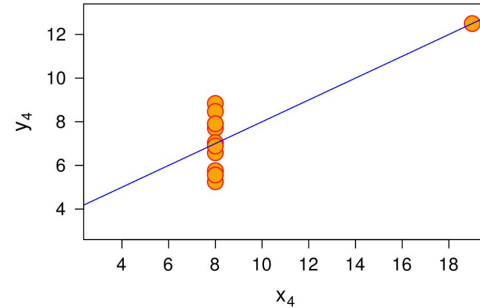
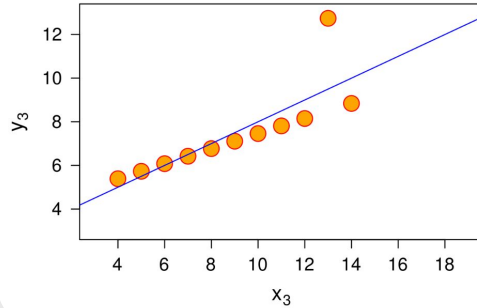
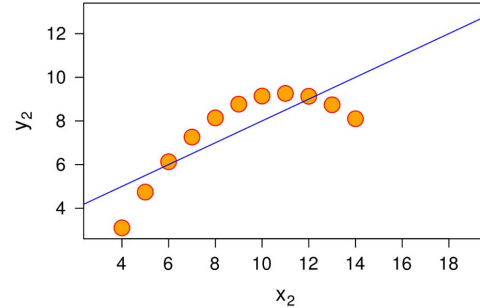
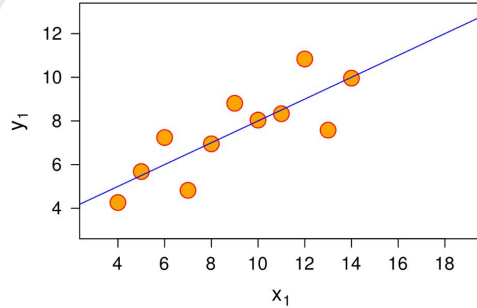
	I		II		III		IV	
	x	y	x	y	x	y	x	y
media	9,0	7,50	9,0	7,50	9,0	7,50	9,0	7,50
std	3,3	2,03	3,3	2,03	3,3	2,03	3,3	2,03
varianza	11,0	4,13	11,0	4,13	11,0	4,12	11,0	4,12

https://es.wikipedia.org/wiki/Cuarteto_de_Anscombe



frani.be

~ Cuarteto Anscombe



https://es.wikipedia.org/wiki/Cuarteto_de_Anscombe



frani.be

~Tabla de datos original

	A	B
1	nombreColor	valorHexadec
2	rojo	#f00
3	verde	#0f0
4	azul	#00f
5	cyan	#0ff
6	magenta	#f0f
7	amarillo	#ff0
8	negro	#000



~ JSON

```
json
{
  "arrayColores": [
    {
      "nombreColor": "rojo",
      "valorHexadec": "#f00"
    },
    {
      "nombreColor": "verde",
      "valorHexadec": "#0f0"
    },
    {
      "nombreColor": "azul",
      "valorHexadec": "#00f"
    },
    {
      "nombreColor": "cyan",
      "valorHexadec": "#0ff"
    },
    {
      "nombreColor": "magenta",
      "valorHexadec": "#f0f"
    },
    {
      "nombreColor": "amarillo",
      "valorHexadec": "#ff0"
    },
    {
      "nombreColor": "negro",
      "valorHexadec": "#000"
    }
  ]
}
```



frani.be

~ CSV



frani.be

CSV

```
nombreColor,valorHexadec  
rojo,#f00  
verde,#0f0  
azul,#00f  
cyan,#0ff  
magenta,#f0f  
amarillo,#ff0  
negro,#000
```

~ XML



frani.be

xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<arrayColores>
  <nombreColor>rojo</nombreColor>
  <valorHexadec>#f00</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>verde</nombreColor>
  <valorHexadec>#0f0</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>azul</nombreColor>
  <valorHexadec>#00f</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>cyan</nombreColor>
  <valorHexadec>#0ff</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>magenta</nombreColor>
  <valorHexadec>#f0f</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>amarillo</nombreColor>
  <valorHexadec>#ff0</valorHexadec>
</arrayColores>
<arrayColores>
  <nombreColor>negro</nombreColor>
  <valorHexadec>#000</valorHexadec>
</arrayColores>
```

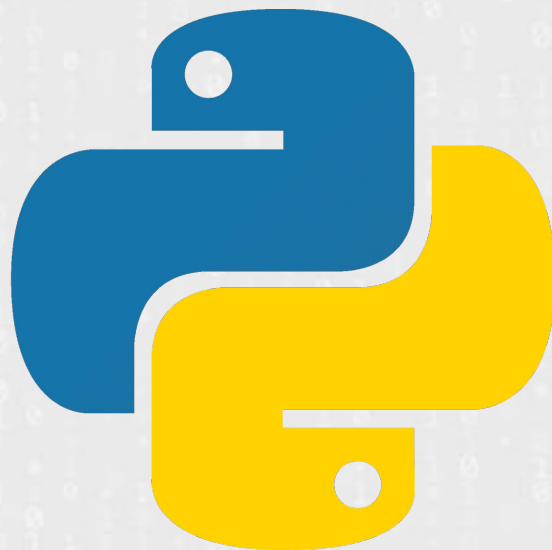


frani.be

Introducción a Python



frani.be



pythonTM

~ Python



frani.be

Inicialmente diseñado como una forma de escribir scripts que "automatizan las cosas aburridas"

Líder en ciencia de datos, machine learning y gestión de infraestructura

~ Características de Python



frani.be

- Lenguaje interpretado
- Lenguaje de fácil lectura y escritura
- Amplio repertorio de librerías y frameworks



frani.be

~ Sintaxis de Python

Python es conocido por su **legibilidad** y **simplicidad**, lo que hace que aprender y escribir código en este lenguaje sea más sencillo en comparación con muchos otros.

Python utiliza la **indentación** (espacios o tabulaciones) para definir bloques de código. La cantidad de espacio debe ser consistente dentro del mismo bloque.



frani.be

~ Elementos básicos

- Comentarios
- Print
- Variables
- Tipos de datos
- Operaciones matemáticas
- Condicionales
- Bucles
- Funciones
- Listas, tuplas, diccionarios
- Importación de módulos



frani.be

~ Comentarios

Los comentarios de una sola línea se inician con #.

Para comentarios multilínea, se pueden usar tres comillas dobles o simples.

```
# Esto es un comentario
```

```
"""
```

```
Esto es un comentario  
multilínea
```

```
"""
```



frani.be

~ Print

Un print es una función que nos permite mostrar un mensaje en la pantalla. En este caso, el mensaje es 'Hola mundo'.

```
print('Hola mundo')
```



frani.be

~ Variables

Una variable es un espacio en la memoria de la computadora donde podemos guardar un valor. En este caso, el valor es 'Hola mundo', y el nombre de la variable es mensaje.

```
mensaje = 'Hola mundo'  
print(mensaje)
```



frani.be

~Tipos de datos

- **int**: números enteros
- **float**: números decimales
- **str**: texto
- **bool**: booleano (True o False)
- **list**: lista
- **tuple**: tupla
- **dict**: diccionario



frani.be

~ Operaciones matemáticas

Los operadores son símbolos que nos permiten hacer operaciones matemáticas.

```
print(1 + 2)
```

```
print(1 - 2)
```

```
print(1 * 2)
```

```
print(1 / 2)
```

```
print(1 // 2)
```

```
print(1 % 2)
```

```
print(1 ** 2)
```



frani.be

~ Condicionales

Los condicionales son instrucciones que nos permiten tomar decisiones.

```
a = 1
b = 2
if a == b:
    print('a es igual que b')
else:
    print('a es diferente de
b')
```



frani.be

~ Condicionales

Los condicionales son instrucciones que nos permiten tomar decisiones.

```
a = 1
b = 2
if a > b:
    print('a es mayor que b')
elif a < b:
    print('a es menor que b')
else:
    print('a es igual a b')
```

~ Bucles



frani.be

Es una estructura de control que permite repetir un conjunto específico de instrucciones varias veces, según se cumpla una condición o hasta que una condición determinada se cumpla.



frani.be

~ Bucle for

Este bucle se utiliza para iterar sobre una secuencia o para ejecutar un bloque de código un número determinado de veces.

```
for i in range(5):  
    print(i)
```



frani.be

~ Bucle while

Permite ejecutar un conjunto de instrucciones mientras una condición sea verdadera.

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1
```



frani.be

~ Funciones

Una función es un bloque de código organizado y reutilizable que se utiliza para realizar una tarea determinada.

- ➔ Permiten la reutilización de código
- ➔ Facilitan la modularidad
- ➔ Hacen que el código sea más legible y mantenible



frani.be

~ Funciones

Una función se define utilizando la palabra clave `def`. Se invoca la función por su nombre seguido de paréntesis, y pasando los argumentos necesarios.

```
def suma(a, b):  
    return a + b  
  
total = suma(5, 3)
```




~Listas, tuplas, diccionarios

```
lista = [1, 2, 3, 4]

tupla = (1, 2, 3, 4)

diccionario = {
    "clave1": "valor1",
    "clave2": "valor2"
}
```

- **Listas:** Colecciones ordenadas y mutables.
- **Tuplas:** Colecciones ordenadas e inmutables.
- **Diccionarios:** Colecciones no ordenadas, mutables e indexadas. Compuestos por pares clave-valor.



frani.be

~ Importación de módulos

Un módulo es un archivo que contiene definiciones, como funciones, clases y variables, así como código ejecutable. La importación de módulos permite reutilizar código escrito por otros o por ti mismo en diferentes programas y scripts.

La capacidad de importar módulos es esencial para mantener el código organizado, modular y reutilizable.



frani.be

~ Importación de módulos

```
import math  
from datetime import date
```



frani.be

Análisis de datos con Pandas y visualización con Altair

~ Librería Pandas



Pandas es una **librería** que proporciona **estructuras** de datos y herramientas de **análisis de datos** de alto rendimiento y fáciles de usar.

pandas.pydata.org/

- Lectura y escritura de datos en múltiples formatos (CSV, Excel, SQL, etc)
- Filtrado, selección y manipulación de datos.
- Gestión de datos faltantes.
- Fusionar y unir datos.
- Transformación de datos.
- Funcionalidades de análisis.

~ Librería Altair



Altair es una **librería de visualización de datos**. Está diseñada para crear visualizaciones atractivas y es útil para visualizar datos estadísticos.

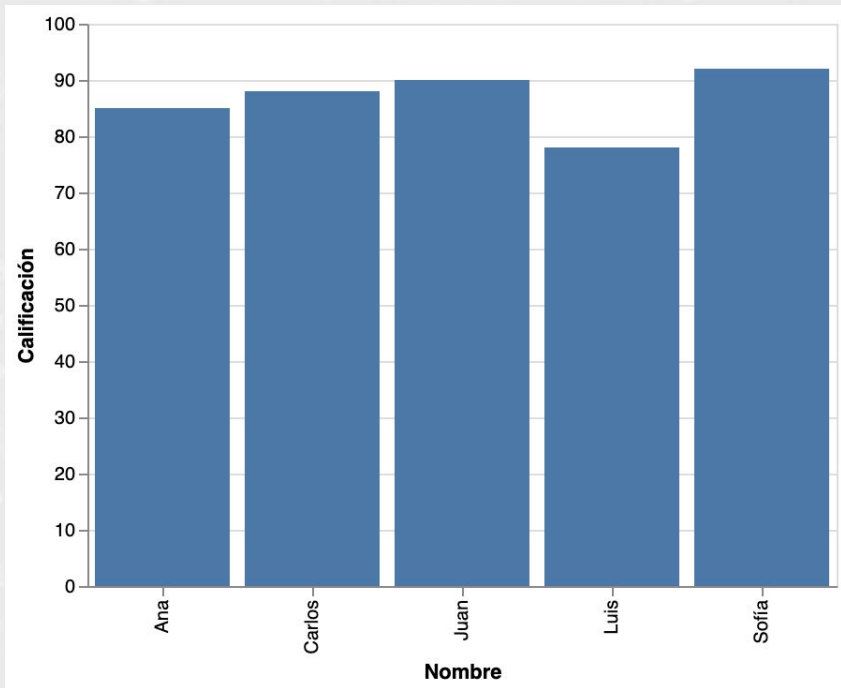
altair-viz.github.io/

- Está integrado con Pandas, lo que facilita la visualización directa de DataFrames.
- Incluye una amplia variedad de estilos y paletas de colores.
- De código simple y legible.
- Las visualizaciones se pueden exportar a HTML.



frani.be

~ Ejemplo gráfico de barras



```
import pandas as pd
import altair as alt

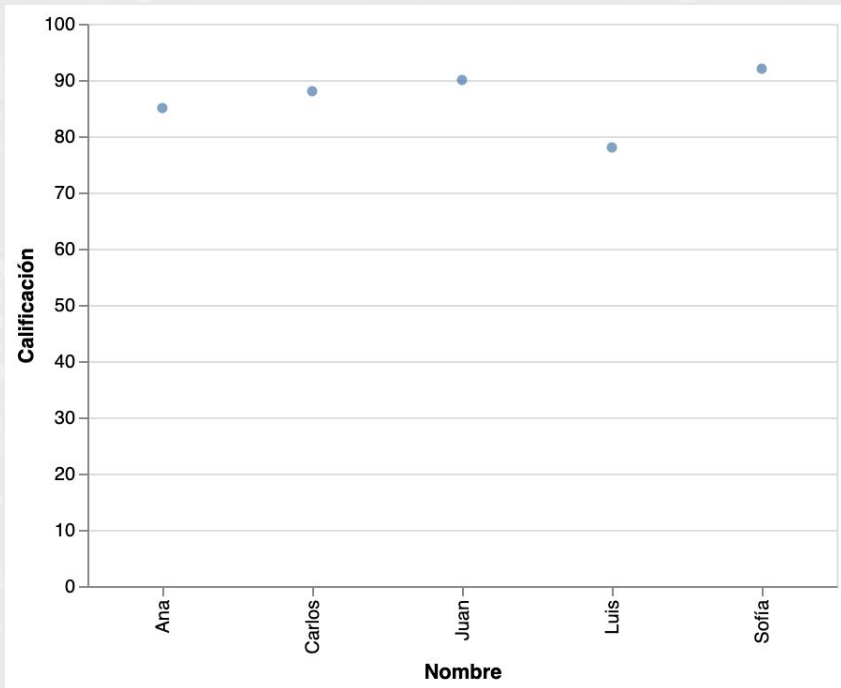
data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_bar().encode(
    x='Nombre:O',
    y='Calificación:Q'
).properties(width=400)
```



frani.be

~ Ejemplo gráfico de puntos



```
import pandas as pd
import altair as alt

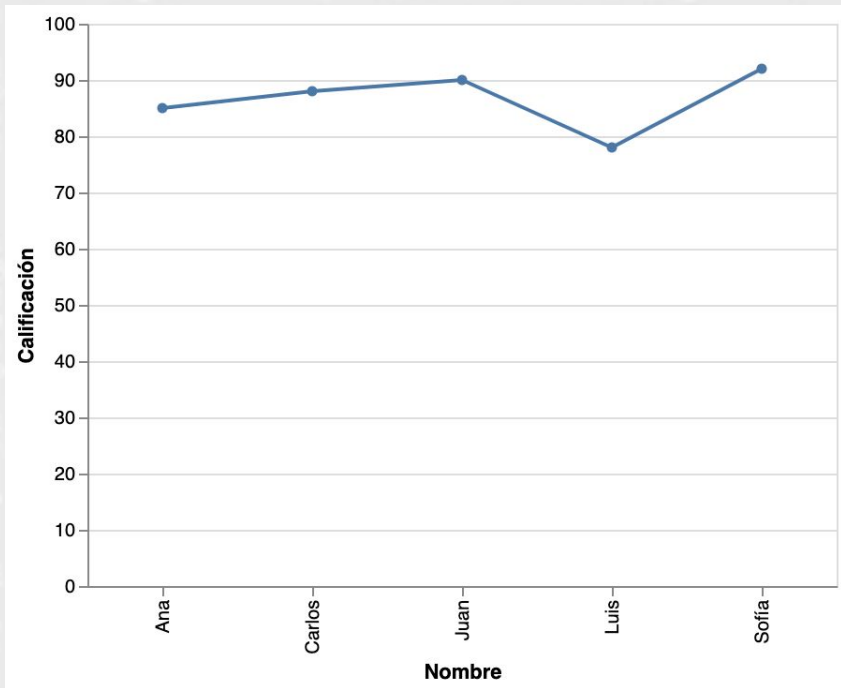
data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_circle().encode(
    x='Nombre:O',
    y='Calificación:Q'
).properties(width=400)
```




frani.be

~ Ejemplo gráfico de línea



```
import pandas as pd
import altair as alt

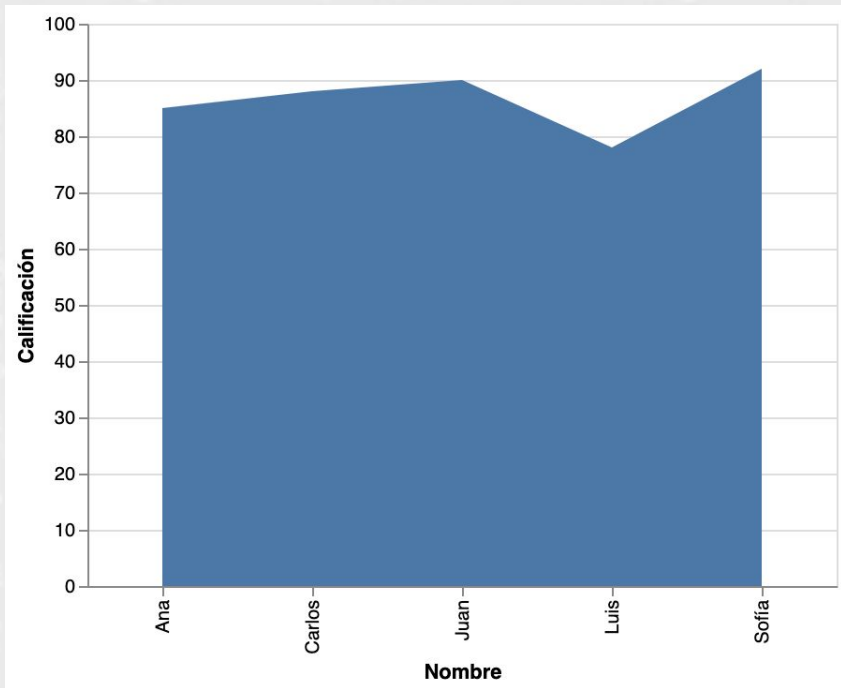
data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_line(point=True).encode(
    x='Nombre:O',
    y='Calificación:Q'
).properties(width=400)
```



frani.be

~ Ejemplo gráfico de área



```
import pandas as pd
import altair as alt

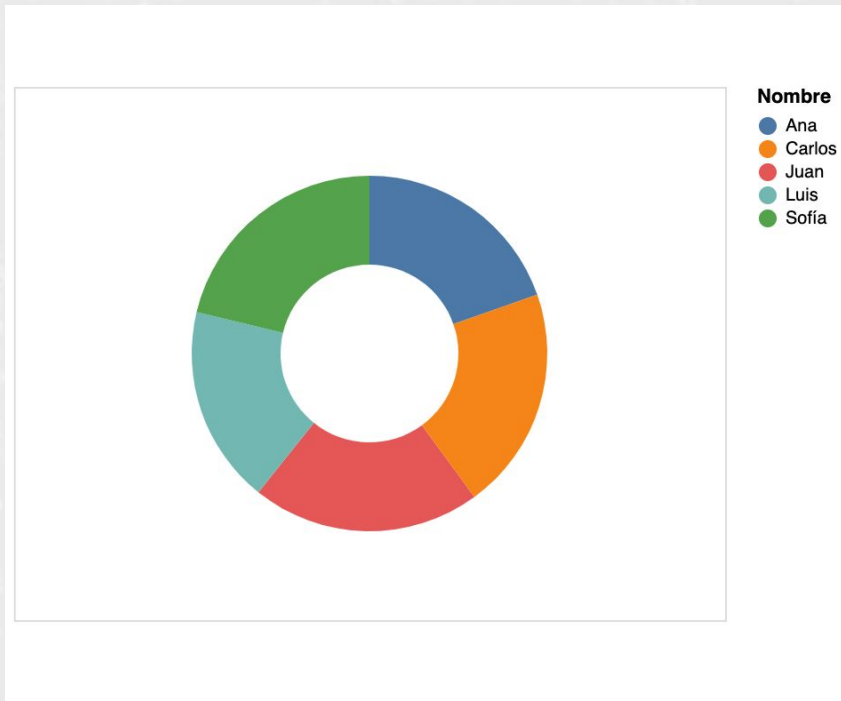
data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_area().encode(
    x='Nombre:O',
    y='Calificación:Q'
).properties(width=400)
```



frani.be

~ Ejemplo gráfico de dona



```
import pandas as pd
import altair as alt

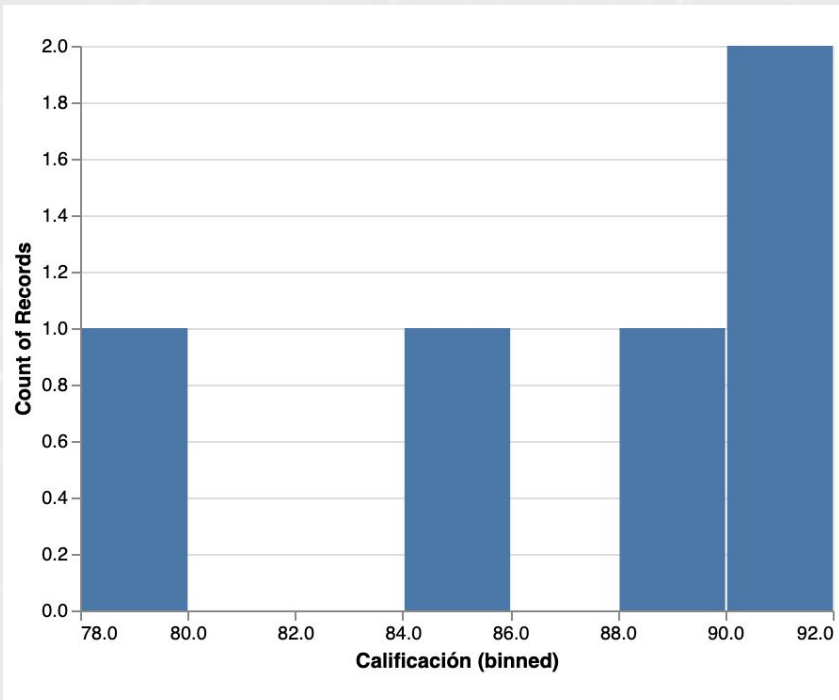
data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_arc(innerRadius=50,
                       outerRadius=100).encode(
    theta='Calificación:Q',
    color='Nombre:N'
).properties(width=400)
```



frani.be

~ Ejemplo histograma



```
import pandas as pd
import altair as alt

data = {
    'Nombre': ['Ana', 'Juan', 'Luis',
               'Sofía', 'Carlos'],
    'Calificación': [85, 90, 78, 92, 88]
}
df = pd.DataFrame(data)

alt.Chart(df).mark_bar().encode(
    x=alt.X("Calificación:Q", bin=True),
    y='count()'
)
```



frani.be

¡Manos al código!



frani.be

<https://www.kaggle.com/datasets/nelgiriyeewithana/most-streamed-spotify-songs-2024>



frani.be

~ Pasos iniciales

1. Abrir [Google Colab](#)
2. Click "New Notebook"
3. Activar Drive
4. Actualizar
5. Subir CSV a la carpeta Colab Notebooks
6. ¡Escribir código!



github.com/frani-be/pycon-taller_de_visualizacion_de_datos