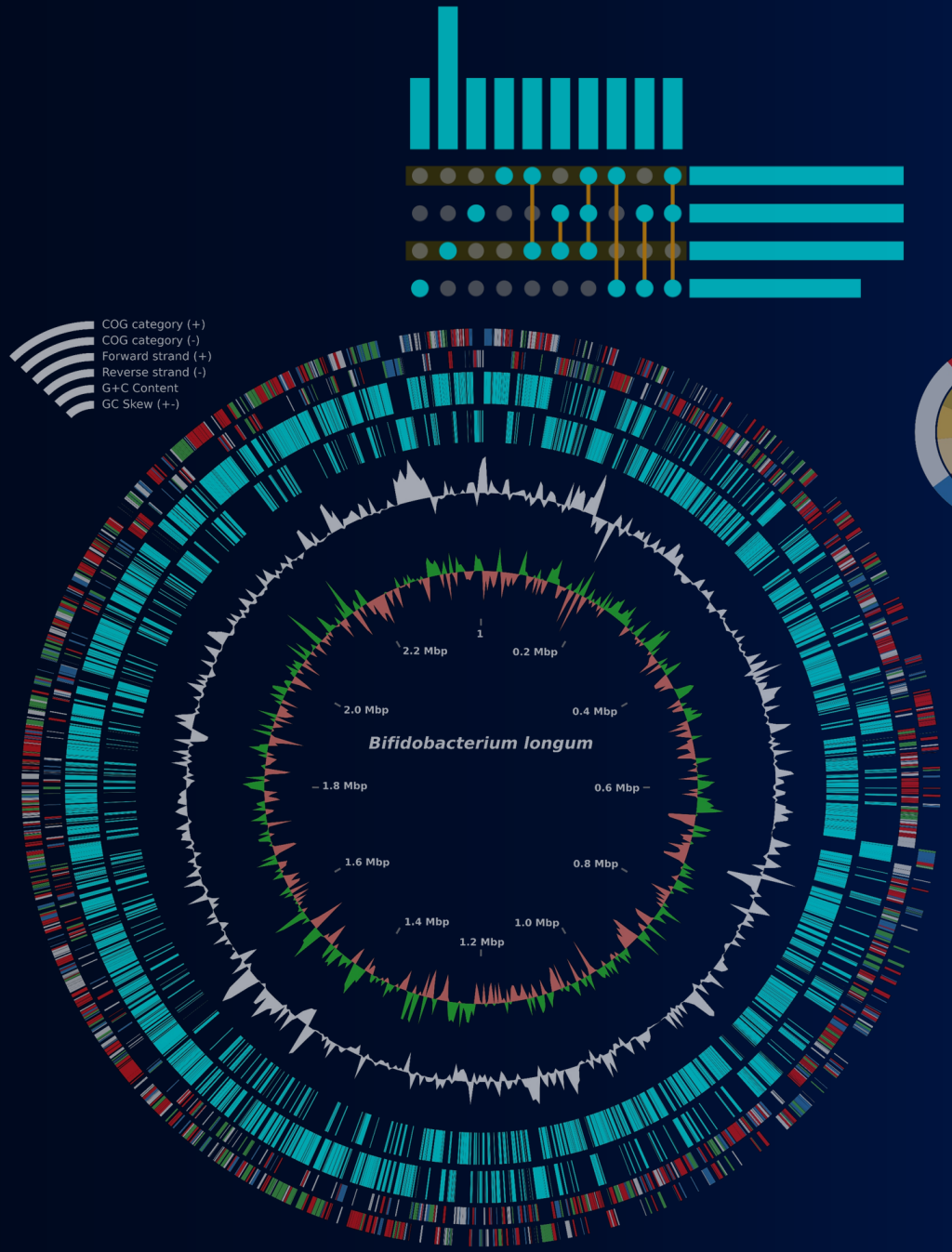
Visualización de Datos con Python para Publicaciones Científicas



GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

OBJETIVO GENERAL

Aprender a crear visualizaciones básicas y avanzadas con el lenguaje de programación Python para publicaciones científicas.

OBJETIVOS ESPECÍFICOS

- Implementar técnicas de visualización de datos utilizando bibliotecas de Python, como Matplotlib, Seaborn, y Plotly.
- Implementar el uso de Widgets para controlar visualizaciones.
- Implementar código de programación para crear gráficos de novo.

DURACIÓN Y LUGAR DEL CURSO

Del 6 - 8 de noviembre.

15 horas, 3 sesiones de 5 horas.

Modalidad presencial de las 14:00 - 19:00 horas.

Aula Multimedia de la Coordinación de Educación Continua.

Descanso: de 16:00 – 16:30 horas.

PORCENTAJE MÍNIMO DE ASISTENCIA PARA OBTENER CONSTANCIA

80% de asistencia.

Programa del curso

https://eduardo1011.my.canva.site/curso3-2024

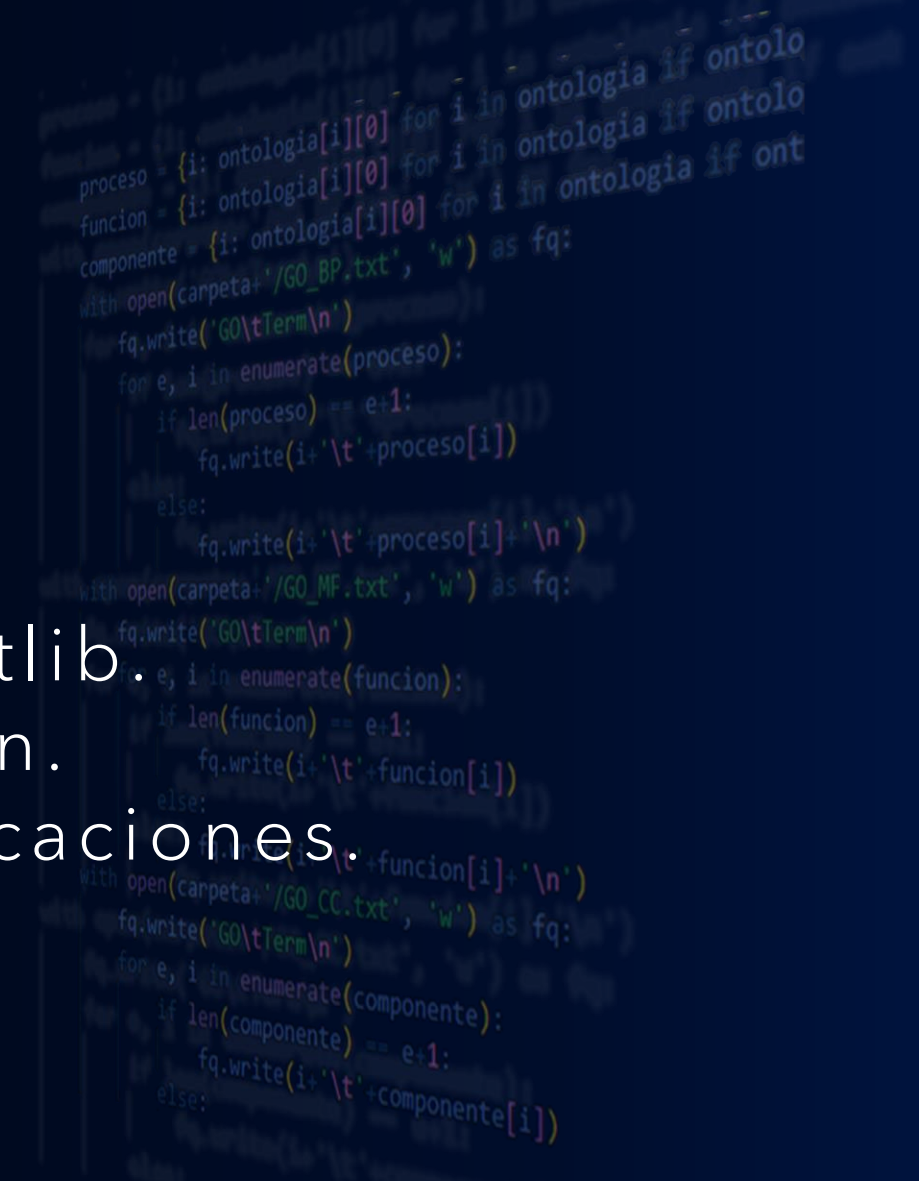
Sesión 1

Python.
Jupyter Notebook.
Variables.
Operadores booleanos.
Listas y tuplas (list, tuple).
Diccionarios (dict).
DataFrame (Pandas).
Condicionales if, elif y else.
Bucle for.

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Sesión 2

Integración de estructuras.
Visualizaciones básicas con Matplotlib.
Visualizaciones básicas con Seaborn.
Gráficos personalizados para publicaciones.



GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Sesión 3

Mapas Genómicos de bacterias.
Datos Metagenómicos.
Introducción a la visualización con Plotly.

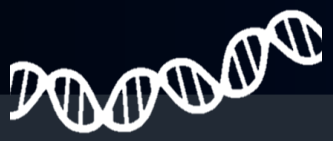
GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Introducción

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

CASCADA DE LAS ÓMICAS

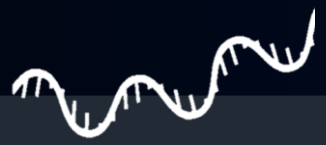
TGTGG...



GENÓMICA

GENES

UGUGG...



TRANSCRIPTÓMICA

RNAm

DMYEE...

PROTEÓMICA

PROTEÍNAS

OTRAS ÓMICAS

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Otras Tecnologías Ómicas

METAPROTEÓMICA

METABOLÓMICA

NUTRIGENÓMICA

SECRETÓMICA

METAGENÓMICA

UBIQUITINÓMICA

EPIGENÓMICA

ACETILÓMICA

METATRANSCRIPTÓMICA

METILÓMICA

INTERACTÓMICA

FOSFOPROTEÓMICA

LIPIDÓMICA

FARMACOGENÓMICA

FENÓMICA

SUBPROTEÓMICA

IONÓMICA

GLICÓMICA

FLUXÓMICA

ADN, ARN, PROTEÍNAS, etc.

- Caracteres
- Composición
- Señales
- Tamaños
- Modificaciones
- Funciones
- Interacciones

Análisis
De Datos

Qué 

Cuánto 

Cómo 

Para 

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Input

Cadenas de caracteres, archivos.
Formatos de inputs
(fasta, fastq, txt, ab1, json, entre otros).

.fastq

```
@P5_S371.1
GTGTCAGCAGCCGCGGTAATACGTAGTTGGCGAGCGTTGTCCGGATTTACTGGG
+
<A@A@CECCF@:@7@+,C,;,E,,<,,6+++++8@+BCE,+,8@C,<C,,,9+@
@P5_S371.2
GTGCCAGCAGCCGCGGTAATACGTAGGTGGCAAGCGTTTTCCGGAATTATTGGG
+
CCCCCCD<@CFC@C@F7@C,CCF9,C,8C,9,9;BC+8CC7+,+:C,EE,,,6+
@P5_S371.3
GTGCCAGCAGCCGCGGTAATACGTAGGGGGCAAGCGTTATCCGGATTTACTGGG
+
<CBCCD@@<FEGD7FEF7,C,C<F9@;++6:,,C:BF,CCC::@@FE,69<,,E
```

.fasta

```
>Sequence1
ATGGTATACCGCCAGATCAACCAGGACGGCGCCGGCCC
CCTGACGGCCATGGTCGACGGCACCTCGGGCGGCACCG
ACAACAGCGCTTTCCAAAAGGCCGAGGTCACCCAGGAC
GTCCCCGGTCTCGGCATCGGCGGCCTCTCACTCGCCAC
CAACACTGACTTCCCTCTGAAGGTTCAGATGCCCGCCG
GCATGACCTGCGAGGGCACCGTCGGCGGCGTCAAGAAC
```

> 10,000 líneas

.txt

```
30956  72  4D2ZZ.faa.final_tree.fa -  148 9.8e-29 97.0   1.8 1  1  5.9e-32 1.1e-28 96.8   1.8 1  60  1  60  1  68
30957  260 4D302.faa.final_tree.fa -  266 6.4e-136  448.9  1.6 1  1  7.6e-139  7e-136 448.7  1.6 9  266 3  260
30958  86  4D302.faa.final_tree.fa -  266 0.83   5.9 6.2 1  1  0.001  0.94   5.7 6.2 46 83  37 73  21 84  0.72
30959  260 4D303.faa.final_tree.fa -  269 7e-77  255.3  6.9 1  1  2.7e-78 5e-75  249.3  6.9 26 269 3  260 1  260
30960  206 4D304.faa.final_tree.fa -  215 8.6e-127  417.9  0.0 1  1  5.1e-130  9.4e-127  417.7  0.0 6  211 3
30961  159 4D306.faa.final_tree.fa -  198 3.1e-05 20.4   5.0 1  1  2e-08  3.7e-05 20.1   4.1 91 150 68 127 54 158
30962  138 4D307.faa.final_tree.fa -  141 3.2e-90 296.2  6.1 1  1  1.9e-93 3.5e-90 296.0  6.1 4  141 1  138 1  138
30963  322 4D309.faa.final_tree.fa -  425 1.6e-75 251.7  105.0  1  2  7.3e-46 1.4e-42 143.3  13.1   1  94  1  90
30964  322 4D309.faa.final_tree.fa -  425 1.6e-75 251.7  105.0  2  2  8.1e-37 1.5e-33 113.5  85.8   158 399 101 315
```

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Input

Cadenas de caracteres, archivos.
Formatos de inputs
(fasta, fastq, txt, ab1, json, entre otros).

Conocimiento informático

Visualización de Datos para Publicaciones Científicas

Reglas para crear visualizaciones personalizadas

Conocer los datos.
Definir la mejor opción de gráfico (barras, etc.)
Remover detalles innecesarios (sombras, bordes).
Definir colores adecuados.
Dimensiones adecuadas.
Variables categóricas asociadas a los datos.

Simplicidad > Orden > Calidad > Elegancia

Contents lis

Environmenta

journal homepage:

Short communication

Ten guidelines for effective data visualization in scientific publications

Christa Kelleher*, Thorsten Wagener

Department of Civil and Environmental Engineering, The Pennsylvania State University, 212 Sackett Building, University Park, PA 16802, USA

EDITORIAL

Annals of Internal Medicine

Bringing Data to Life: Interactive Visualizations of Complex Data

When reflecting on his favorite part of data analysis, the esteemed statistician John Tukey (1915–2000), often called the father of exploratory data analysis, said, "Taking boring flat data and bringing it to

(2.81%) who were healthy at baseline and at 2-year follow-up but then experienced an ADL impairment 4 years after baseline. All individual participant trajectories that, when pooled, add up to the 22% of partici-

Patterns

Perspective

Principles of Effective Data Visualization

Stephen R. Midway[1,*]
[1]Department of Oceanography and Coastal Sciences, Louisiana State University, Baton Rouge, LA 70803, USA
*Correspondence: smidway@lsu.edu
https://doi.org/10.1016/j.patter.2020.100141

Psychological Science in the Public Interest
Volume 22, Issue 3, December 2021, Pages 110-161

https://doi.org/10.1177/15291006211051956

SAGE journals

The Science of Visual Data Communication: What Works

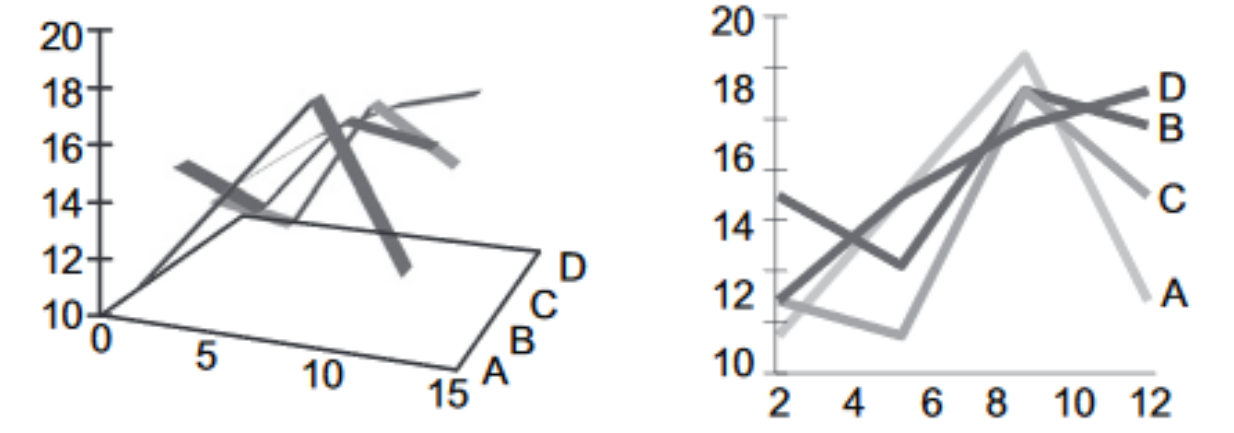
Steven L. Franconeri[1], Lace M. Padilla[2], Priti Shah[3], Jeffrey M. Zacks[4], and Jessica Hullman[5]

Abstract

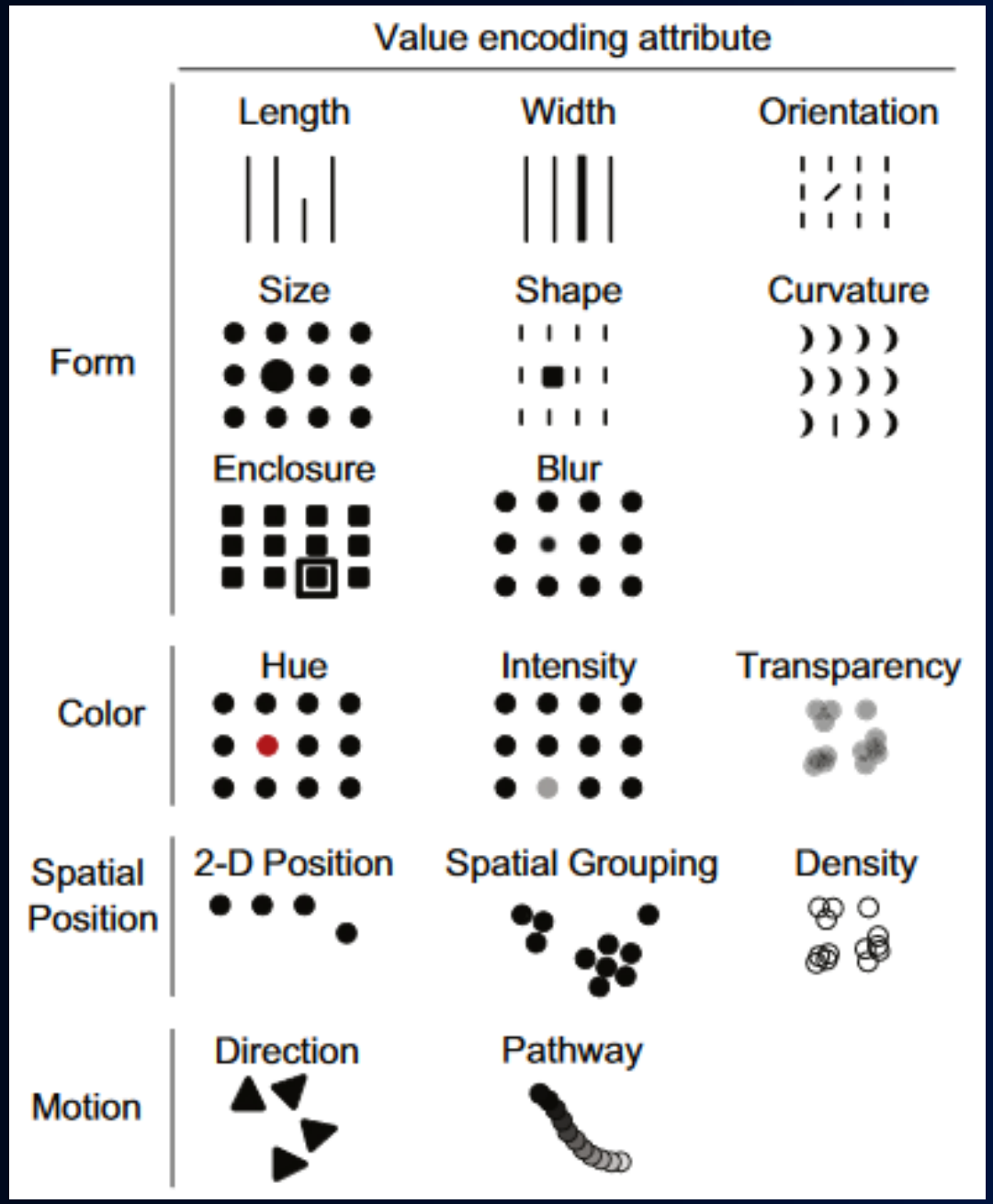
Effectively designed data visualizations allow viewers to use their powerful visual systems to understand patterns in data across science, education, health, and public policy. But ineffectively

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

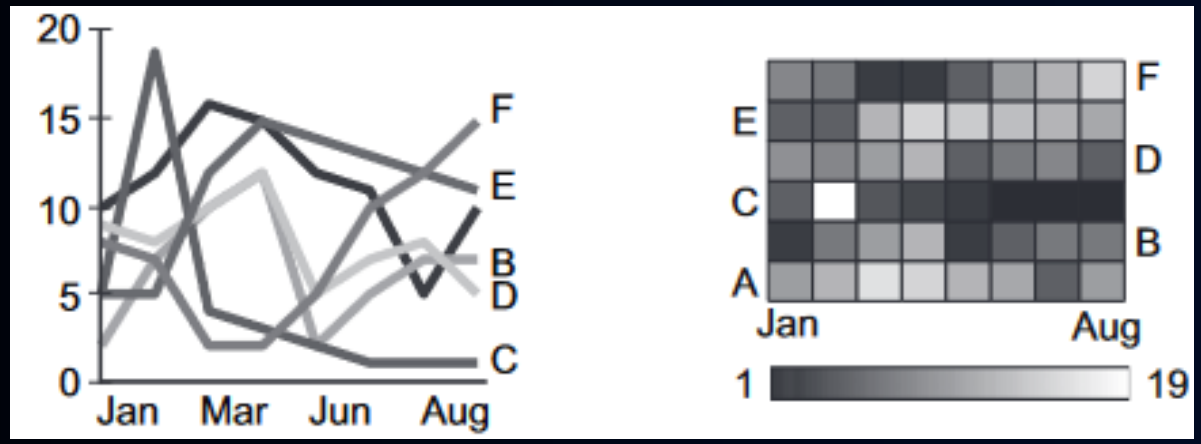
1) Crea el gráfico más simple que transmita la información que quieres transmitir.



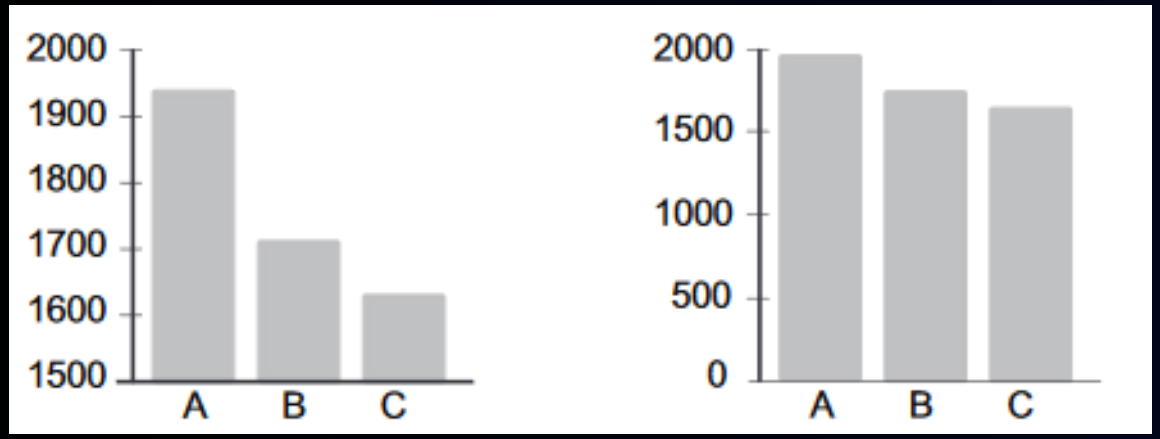
2) Considerar el tipo de objeto de codificación y atributo utilizado para crear un gráfico (puntos, líneas, etc.)

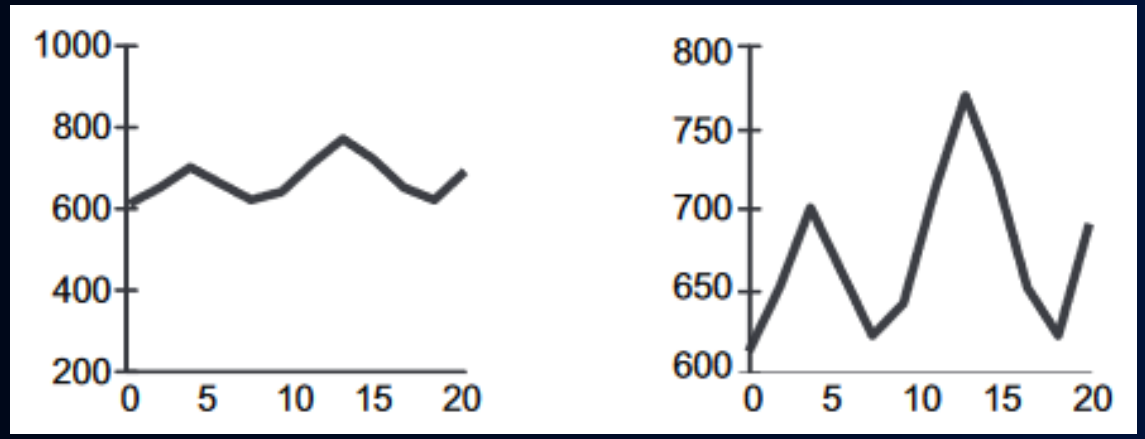


3) Centrarse en visualizar patrones o en visualizar detalles, dependiendo del propósito de la trama.

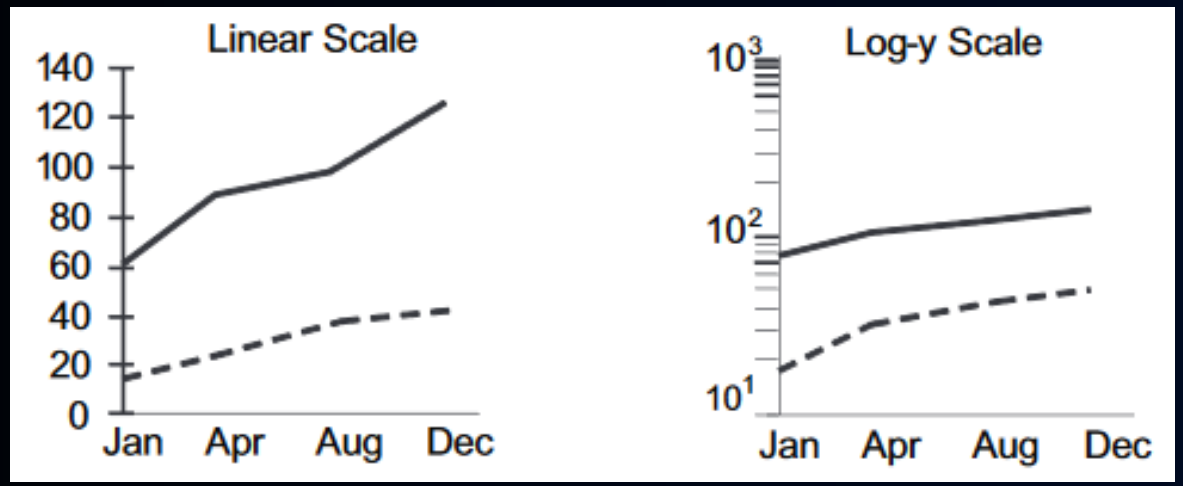


4) Seleccione rangos de ejes significativos.

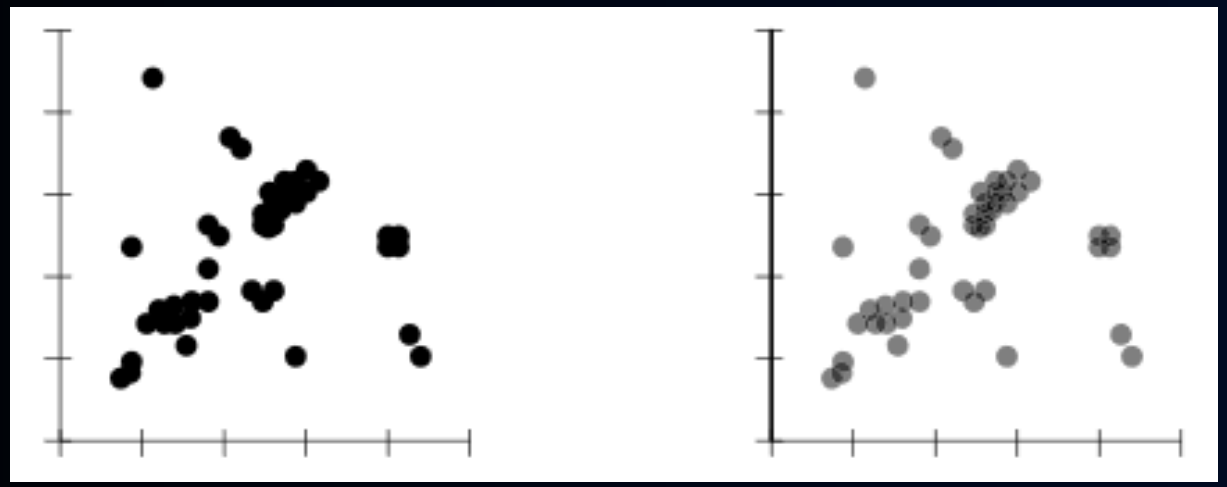




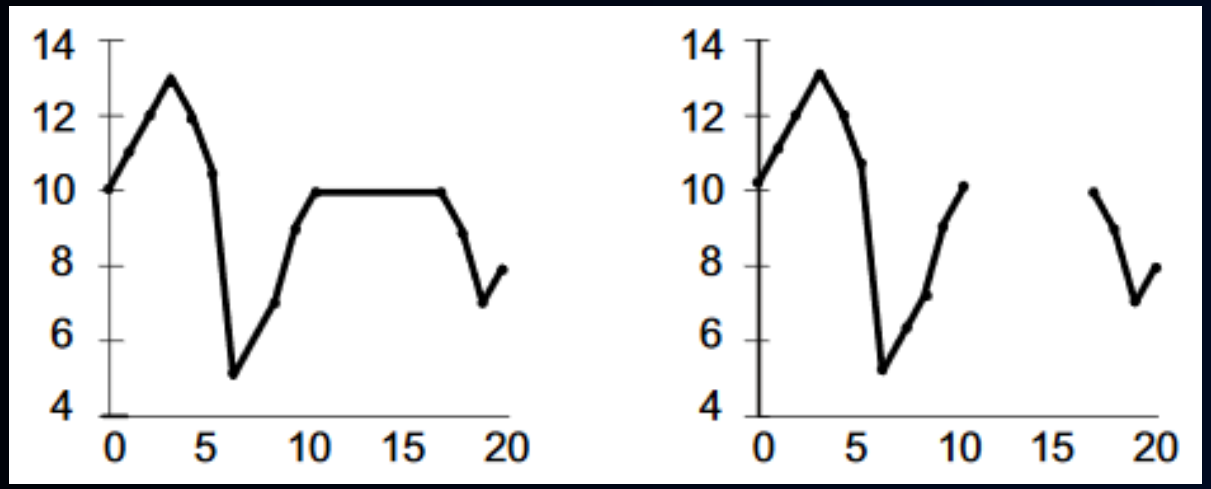
5) Elegir cuidadosamente la transformación de datos (normalización).



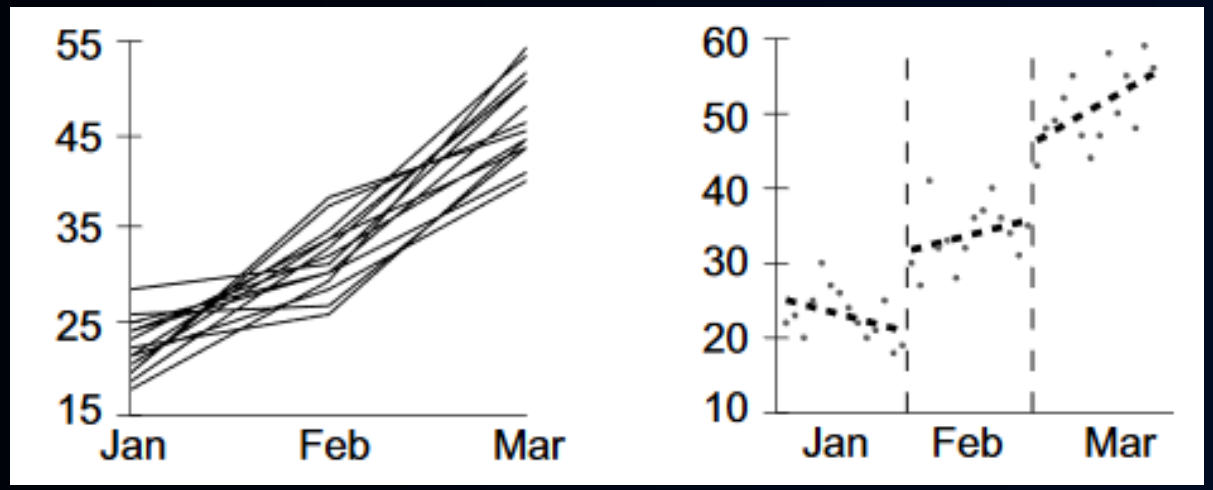
6) Graficar puntos superpuestos de manera que las diferencias de densidad se hagan evidentes en los diagramas de dispersión.



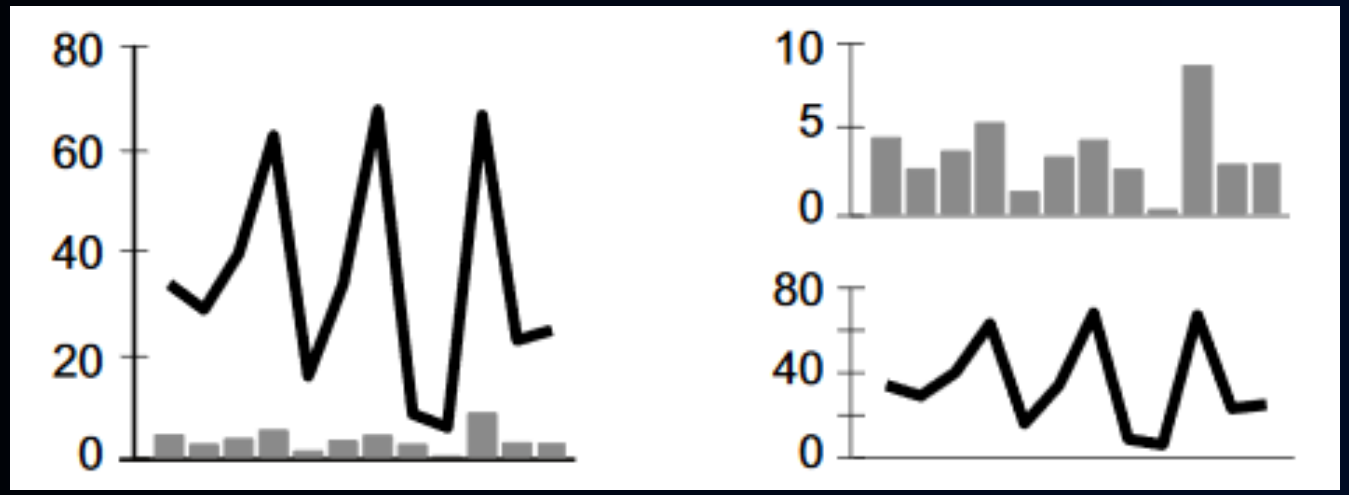
7) Utilice líneas al conectar datos secuenciales en gráficos de series temporales.



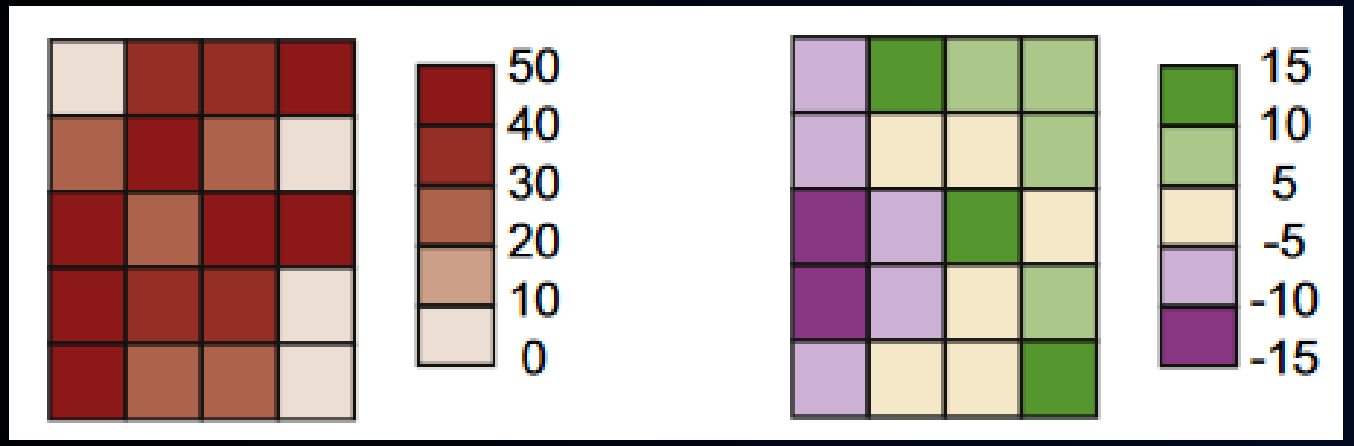
8) Agregar conjuntos de datos más grandes de maneras significativas.

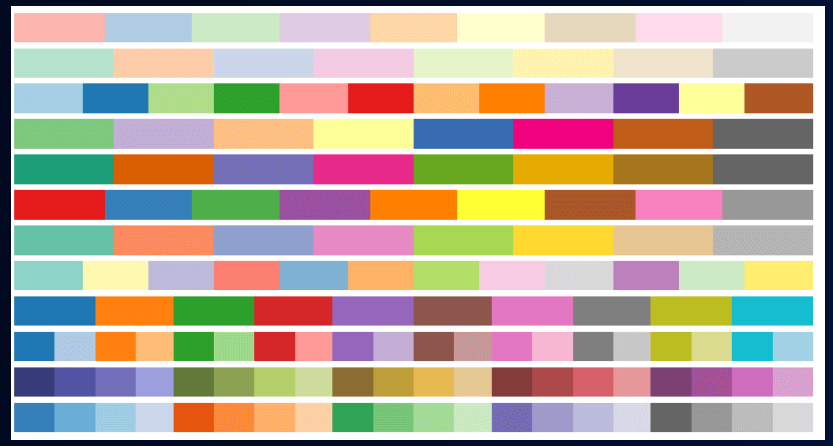


9) Mantener los rangos de los ejes lo más similares posible para poder comparar variables.



10) Seleccione un esquema de color apropiado según el tipo de datos.





¿Qué necesito para crear visualizaciones personalizadas?

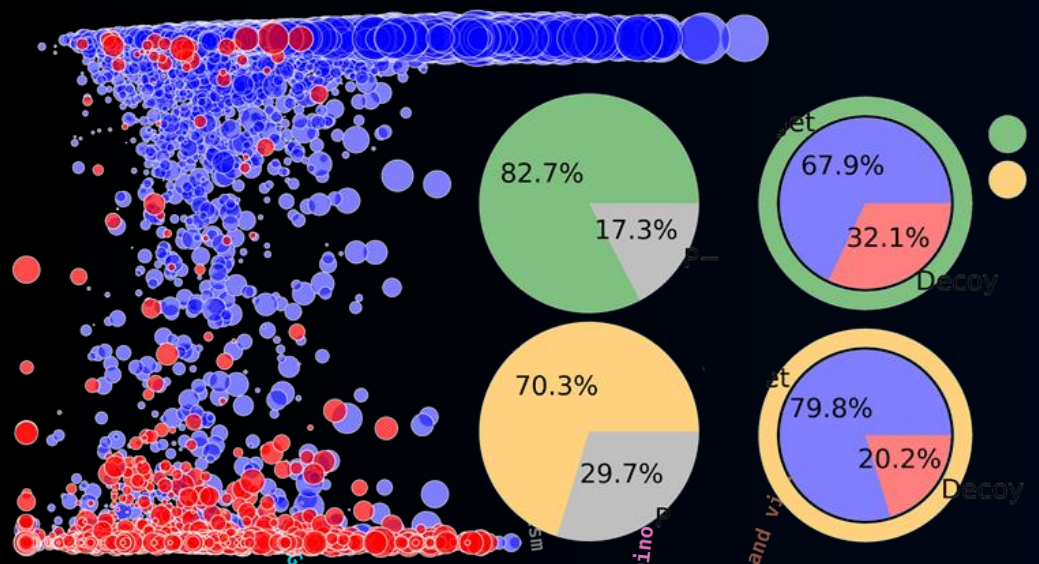
GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

> programar

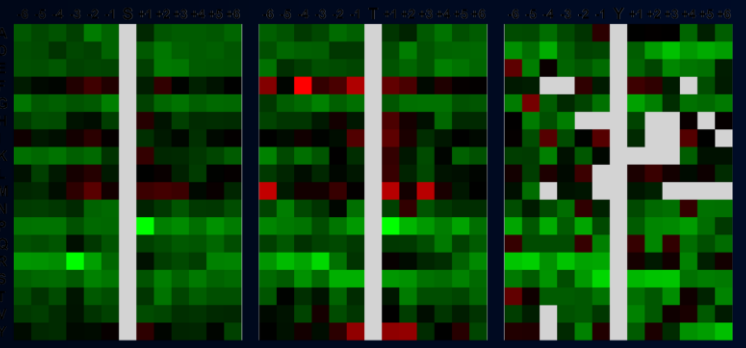
GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Análisis de datos y visualización

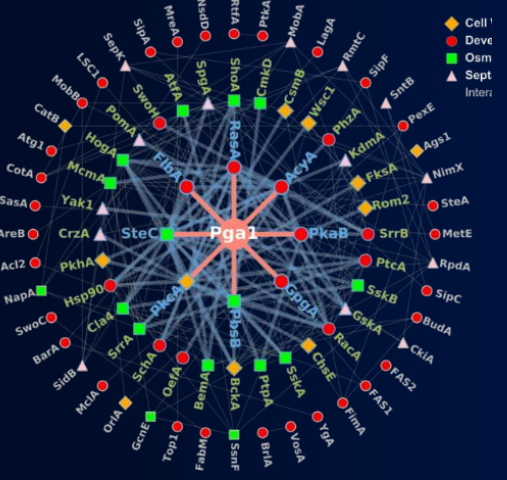
Distribuciones



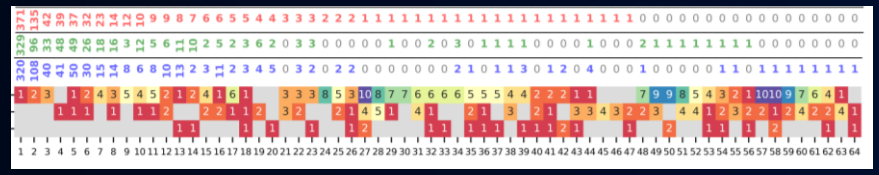
Patrones



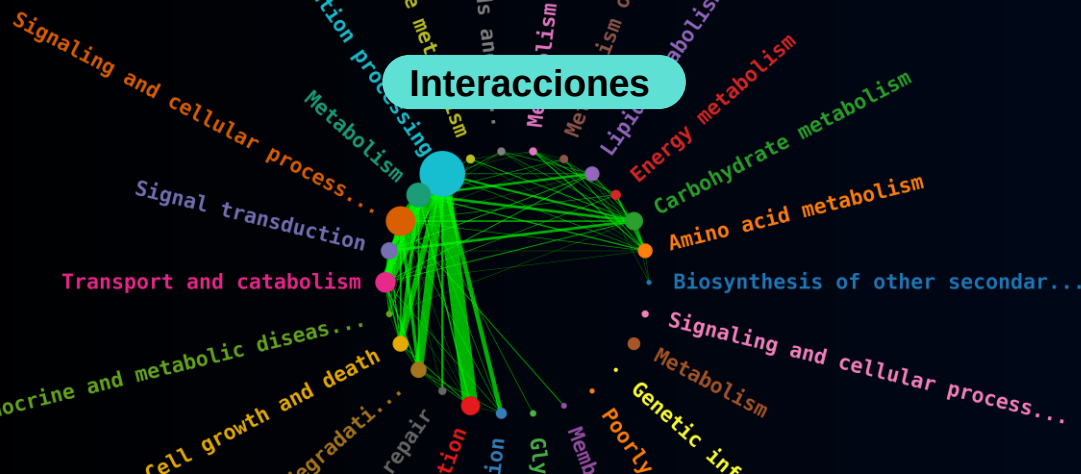
Interacciones



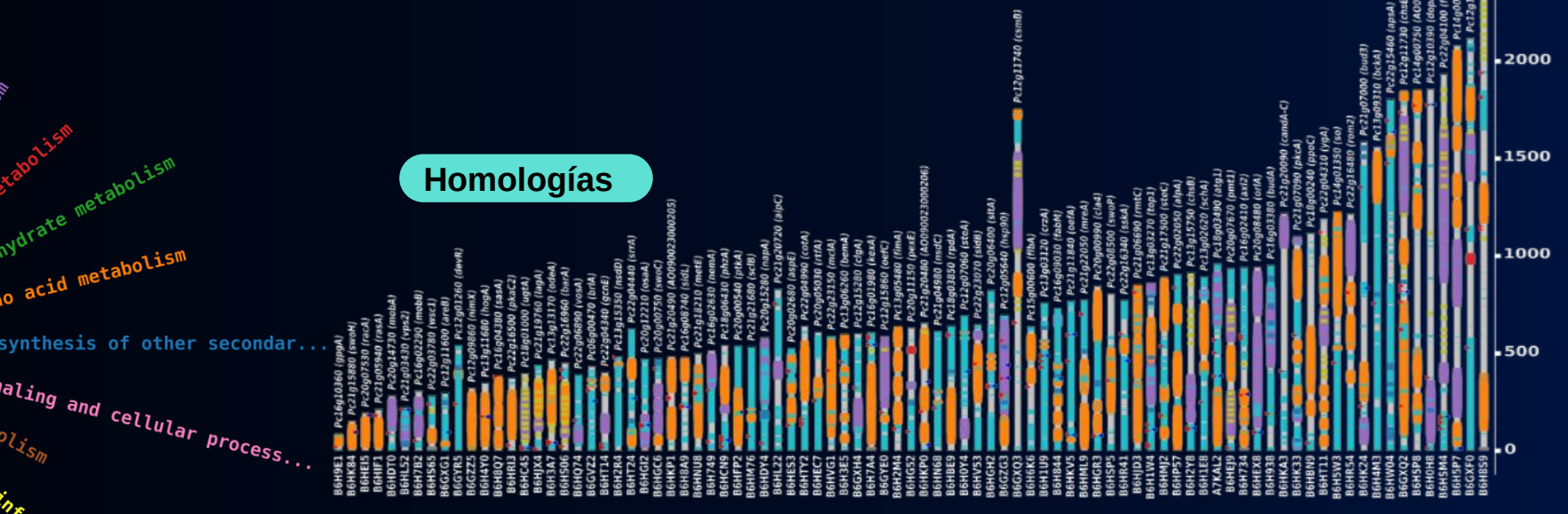
Patrones



Interacciones



Homologías



Análisis de datos y visualización

Order Article Reprints

Open Access Article

NeVOmics: An Enrichment Tool for Gene Ontology and Functional Network Analysis and Visualization of Data from OMICs Technologies

by Eduardo Zúñiga-León, Ulises Carrasco-Navarro and Francisco Fierro *

Departamento de Biotecnología, Universidad Autónoma Metropolitana-Unidad Iztapalapa, Ciudad de Mexico 09340, Mexico

* Author to whom correspondence should be addressed.

Genes **2018**, *9*(12), 569; https://doi.org/10.3390/genes9120569

ORIGINAL RESEARCH article

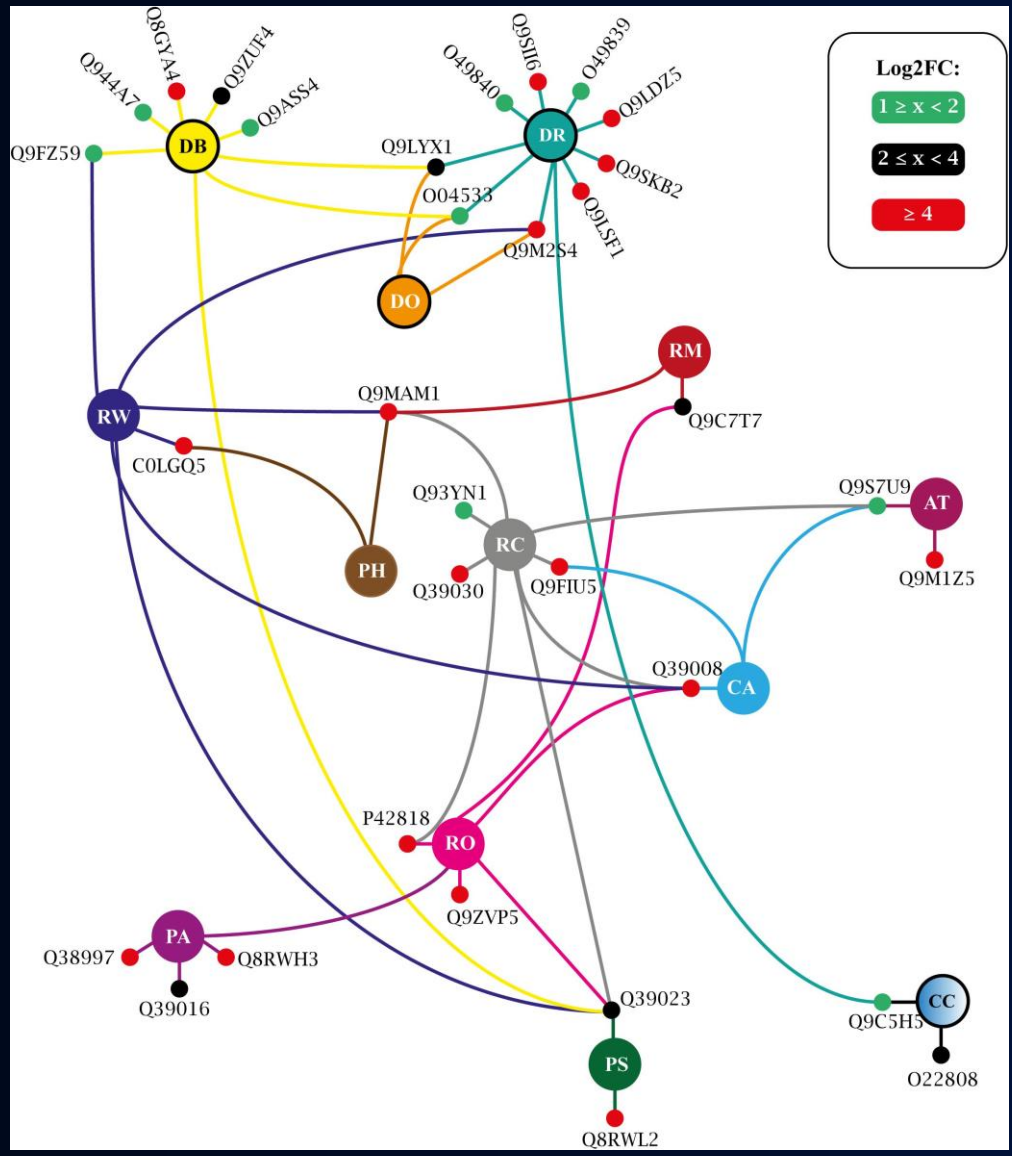
Front. Plant Sci., 13 June 2021
Sec. Plant Systems and Synthetic Biology
Volume 12 - 2021 | https://doi.org/10.3389/fpls.2021.667013

This article is part of the Research Topic
A Systems View of Plant Cellular Communication

View all 8 articles >

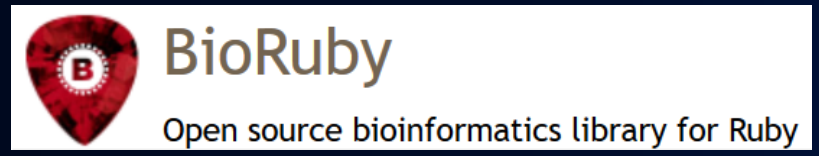
The Cowpea Kinome: Genomic and Transcriptomic Analysis Under Biotic and Abiotic Stresses

José Ribamar Costa Ferreira-Neto[1] Artemisa Nazaré da Costa Borges[2] Manassés Daniel da Silva[1]
David Anderson de Lima Morais[3] João Pacífico Bezerra-Neto[2] Guillaume Bourque[4] Ederson Akio Kido[1]
Ana Maria Benko-Iseppon[2*]



GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

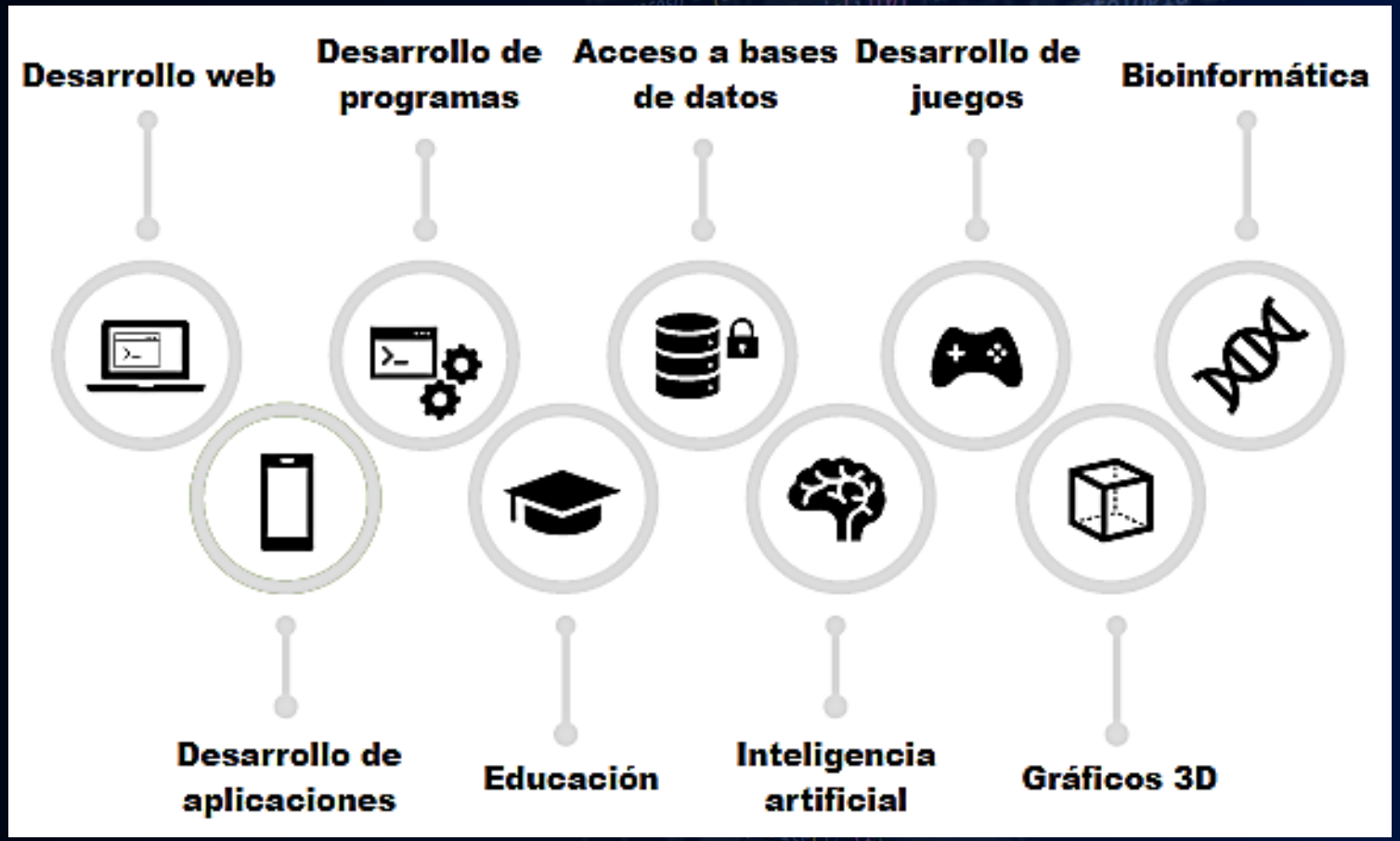
Lenguajes



GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Python

- Propuesta general.
- Lenguaje de alto nivel.
- Tipado dinámico.
- Sintaxis sencilla.
- Altamente interactivo.
- Gran cantidad de bibliotecas.
- Código abierto.
- Aplicaciones científicas.
- Tiene una gran comunidad.



Esencia de la programación

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Estructura de datos (objetos)

Números (*int* o *float*): `1, 2, 3, 4, 5  ó 0.1, 3.5, 11.2`

Cadenas de caracteres (*str*): `"ACTGACAGTCTCA"`

Listas (*list*): `["ACG", "CCT", "ATT", "GCT"]`

Tuplas (*tuple*): `("ACG", "CCT", "ATT", "GCT")`

Diccionarios (*dict*): `{"A":123, "C":"GGG", "G":"--", "T":5}`

Arreglos (*array*)

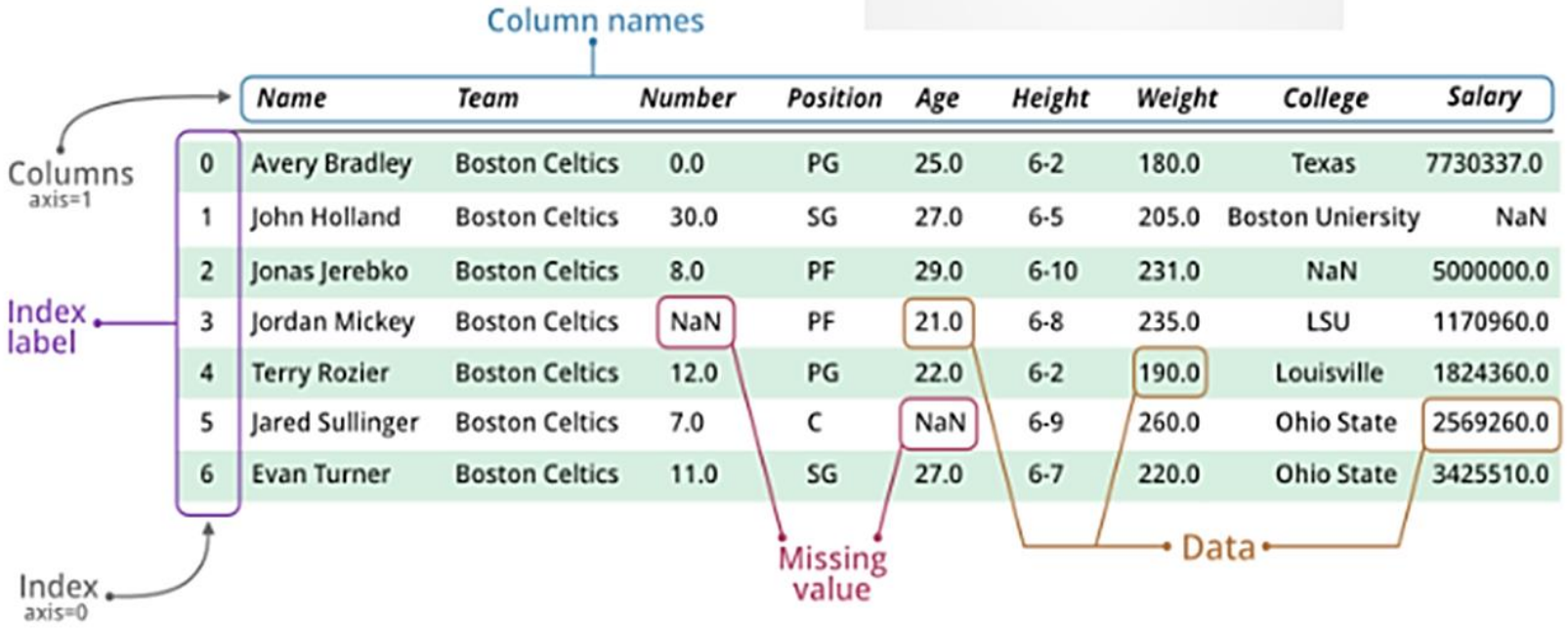
Conjuntos (*set*)

```
set([1, 1, 2, 3])
```

⬇

```
{1, 2, 3}
```

DataFrames



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Jupyter Notebook

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

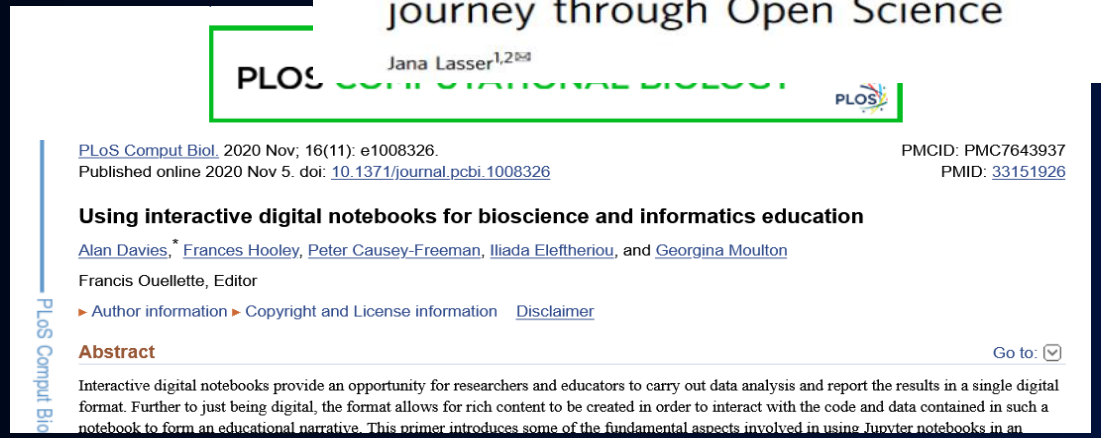
Jupyter Notebook

Educación y ciencia.

Aplicado en programas de enseñanza.

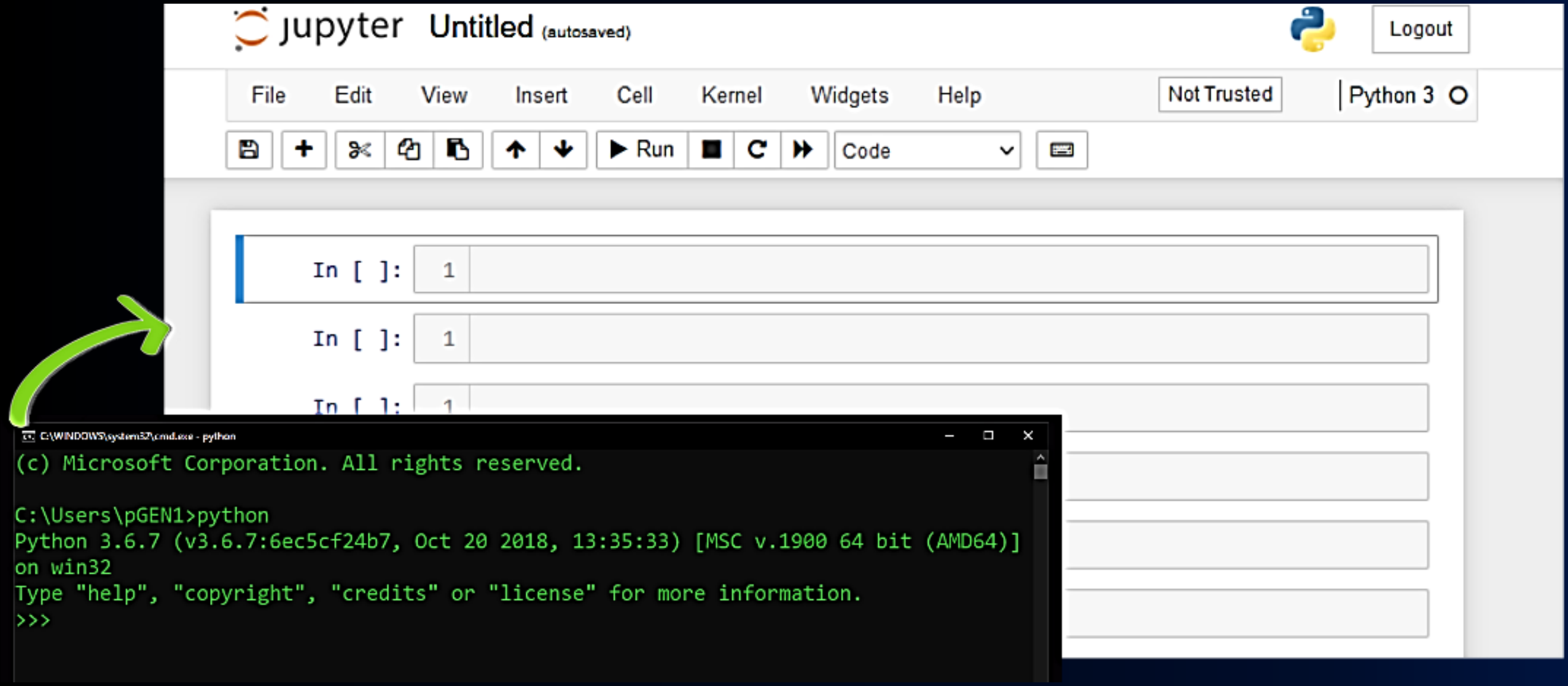
Permite el análisis de grandes volúmenes de datos, apoya la reproducibilidad y permite visualizaciones interactivas.

Los documentos ejecutables ofrecen una manera transparente y reproducible de comunicar los resultados de una investigación.



GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC

Entorno de programación



Entorno de trabajo

Documentos

nov2024

- notebooks
- archivos

Sesión 1

Python.
Jupyter Notebook.
Variables.
Operadores booleanos.
Listas y tuplas (list, tuple).
Diccionarios (dict).
DataFrame (Pandas).
Condicionales if, elif y else.
Bucle for.

GAACACATAGCCGATCACCAACACCGACAAGGTGTACGGAATGAAGAAGATCGCACGGAACCCGTTTTTGAACGCGATTTTCGAATTCAGCAGCACCGCAAGGAACATGCCAAGAAAATTCGTGC