

Clase 6-7

Pandas, numpy y matplotlib

¿Qué es pandas?

pandas es un paquete de manipulación de datos en Python para datos tabulares. Es decir, datos en forma de filas y columnas, también conocidos como DataFrames. Intuitivamente, se puede pensar en un Marco de datos como en una hoja de Excel.

¿Para qué se utilizan los pandas?

pandas se utiliza en todo el flujo de trabajo del análisis de datos.

Con los pandas, se puede:

- Importa conjuntos de datos desde bases de datos, hojas de cálculo, archivos de valores separados por comas (CSV) etc.
- Limpia conjuntos de datos, por ejemplo, tratando los valores que faltan.
- Ordena conjuntos de datos remodelando su estructura en un formato adecuado para el análisis.
- Agrega datos calculando estadísticas de resumen, como la media de las columnas, la correlación entre ellas, etc.
- Visualiza conjuntos de datos y descubre perspectivas

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Instalación de librerías

paso 1: abrir una terminal (CMD)

Paso 2: ya en la terminal

```
C:\Users\jose luis>pip install pandas
```

Paso 3:

```
C:\Users\jose luis>pip install numpy
```

Paso 4:

```
C:\Users\jose luis>pip install matplotlib
```

Paso 5:

```
C:\Users\jose luis>pip install seaborn
```

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Verificamos la instalación de las librerías

```
In [2]: import pandas as pd # cara la libreria pandas y asigna el acrónimo pd
In [3]: print(pd.__version__)
2.2.2

In [4]: import numpy as np # carga la libreria numpy y asigna el acrónimo np
In [5]: print(np.__version__)
1.26.4

In [6]: import matplotlib as plt
In [7]: print(plt.__version__)
3.9.2
```

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Excel files

Installable with `pip install "pandas[excel]"`.

Dependency	Minimum Version	pip extra	Notes
xlrd	2.0.1	excel	Reading Excel
xlsxwriter	3.0.5	excel	Writing Excel
openpyxl	3.1.0	excel	Reading / writing for xlsx files
pyxlsb	1.0.10	excel	Reading for xlsb files
python-calamine	0.1.7	excel	Reading for xls/xlsx/xlsb/ods files

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

SQL databases

Traditional drivers are installable with `pip install "pandas[postgresql, mysql, sql-other]"`

Dependency	Version	Minimum pip extra	Notes
SQLAlchemy	2.0.0	postgresql, mysql, sql-other	SQL support for databases other than sqlite
psycopg2	2.9.6	postgresql	PostgreSQL engine for sqlalchemy
pymysql	1.0.2	mysql	MySQL engine for sqlalchemy
adbc-driver-postgresql	0.8.0	postgresql	ADBC Driver for PostgreSQL
adbc-driver-sqlite	0.8.0	sql-other	ADBC Driver for SQLite

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Access data in the cloud

Installable with `pip install "pandas[fss, aws, gcp]"`

Dependency	Version	Minimum pip extra	Notes
fsspec	2022.11.0	fss, gcp, aws	Handling files aside from simple local and HTTP (required dependency of s3fs, gcsfs).
gcsfs	2022.11.0	gcp	Google Cloud Storage access
pandas-gbq	0.19.0	gcp	Google Big Query access
s3fs	2022.11.0	aws	Amazon S3 access

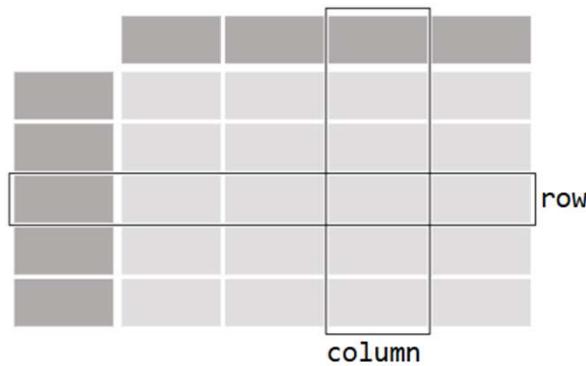
Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Representación de la tabla de datos de pandas

DataFrame



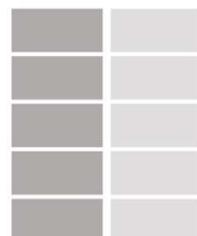
Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Each column in a `DataFrame` is a `Series`

Series



Mgr. Jose Luis Vera

FUNCIÓN/MÉTODOS**Clase 6-7****Pandas, numpy y matplotlib**

Librería math

In [1]: import math

Programa que resuelve un problema específico,
y puede ser llamado(reutilizado)

```

1 def suma(a,b):
2     s= a+b
3     print("La sumas es: ",s)
4
5 if __name__ == '__main__':
6     x1= int(input("ingrese primer numero: "))
7     x2= int(input("ingrese segundo numero: "))
8     suma(x1,x2)
9     n1=10
10    n2=30
11    suma(n1,n2)
12

```

```

1 import math
2 def suma(a,b):
3     s= a+b
4     print("La sumas es: ",s)
5
6 def raiz_cuadrada(x):
7     z=math.sqrt(x)
8     return z
9
10 if __name__ == '__main__':
11     x1= int(input("ingrese primer numero: "))
12     x2= int(input("ingrese segundo numero: "))
13     suma(x1,x2)
14     n1=10
15     n2=30
16     suma(n1,n2)
17     raiz=raiz_cuadrada(9)
18     print("La raiz es: ",raiz)
19

```

Mgr. Jose Luis Vera

Clase 6-7**Pandas, numpy y matplotlib**

Exercises 1 – 22 refer to the functions below. Find the indicated value of the function.

$$\begin{aligned} f(x) &= \sqrt{x+3} - x + 1 \\ g(t) &= t^2 - 1 \\ h(x) &= x^2 + \frac{1}{x} + 2 \end{aligned}$$

- 1. $f(0)$
- 2. $f(1)$
- 3. $f(\sqrt{2})$
- 4. $f(\sqrt{2} - 1)$

```

1 import math
2 def f(x):
3     return math.sqrt(x + 3) - x + 1
4
5 for x in range(0,100):
6     print("f({:.2f}) = {:.5f}".format(x,f(x)))

```

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

1.1. Operaciones básicas

Python permite realizar operaciones de cálculo básicas con números reales. Dado dos números reales a y b , la siguiente tabla muestra como hacerlo

$a+b$	suma
$a-b$	resta
$a*b$	producto
a/b	división
$a//b$	proporciona la parte entera del cociente
$a \% b$	proporciona el resto de la división entera
$a^{**}b$	a^b

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Funciones y constantes predefinidas en Python

Python tiene predeterminadas una serie de funciones que pueden usarse sin tener que importarlas desde ningún módulo o paquete. Enumeramos las más interesantes para nosotros.

- `abs(x)`. Proporciona el valor absoluto de x .
- `divmod(a,b)`. Para dos números enteros a y b retorna el cociente y el resto de la división entera de a entre b .
- `float(x)`. Proporciona la expresión decimal de x .
- `len(s)`. Retorna el tamaño (el número de elementos) de una lista u objeto s .
- `min(a,b,...)` y `max(a,b,...)`. Retornan el máximo y el mínimo de una lista $a,b,...$
- `pow(x,y)`. Se trata de x^y . Equivalente a $x^{**}y$.
- `round(a,n)`. Proporciona a con n cifras decimales.
- `sum(lista)`. Suma los elementos de un lista.

Respecto a las constantes, tenemos las siguientes:

- `True` y `False`. Son constantes de verdadero y falso.
- `None`. Se usa para indicar que ausencia de valor.

Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Librería math

Teoría de números

Las funciones para la teoría de números son las siguientes

<code>ceil(x)</code>	menor entero mayor o igual que x
<code>floor(x)</code>	mayor entero menor o igual que x
<code>gcd(n,m,...)</code>	máximo común divisor de un lista de enteros n,m,...
<code>lcm(n,m,...)</code>	mínimo común múltiplo de un lista de enteros n,m,...
<code>prod(lista)</code>	multiplica los elementos de una lista
<code>fsum(lista)</code>	suma los elementos de una lista
<code>remainder(x, y)</code>	resto de la división entera de x entre y

Para combinatoria tenemos

<code>comb(n,k)</code>	combinaciones de n elementos de k en k
<code>factorial(n)</code>	factorial de n
<code>perm(n, k)</code>	variaciones de n elementos de k en k
<code>perm(n)</code>	permutaciones de n elementos.

gr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Librería math

Funciones logarítmicas y exponenciales

Las funciones logarítmicas y exponenciales son las siguientes

<code>exp(x)</code>	función exponencial e^x
<code>log(x)</code>	logaritmo neperiano de x
<code>log(x,b)</code>	logaritmo en base b de x
<code>sqrt(x)</code>	\sqrt{x}

Funciones trigonométricas

<code>sin(x)</code>	seno de x en radianes
<code>cos(x)</code>	coseno de x en radianes
<code>tan(x)</code>	tangente de x en radianes
<code>asin(x)</code>	arcoseno de x
<code>acos(x)</code>	arcocoseno de x
<code>atan(x)</code>	arcotangente de x
<code>degrees(x)</code>	convierte x de radianes a grados
<code>radians(x)</code>	convierte x de grados a radianes

Mgr. Jose Luis Vera

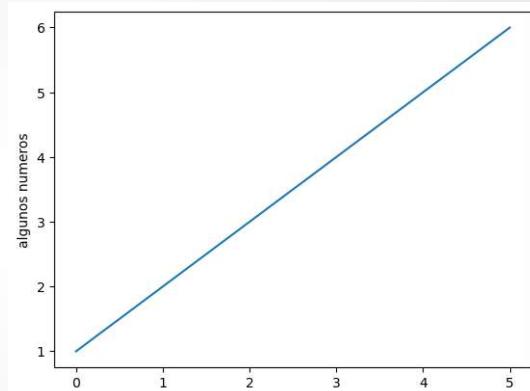
Clase 6-7

Pandas, numpy y matplotlib

```

1 import matplotlib.pyplot as plt
2
3 plt.plot([1, 2, 3, 4, 5, 6])
4 plt.ylabel('algunos numeros')
5 plt.show()
6

```



Mgr. Jose Luis Vera

Clase 6-7

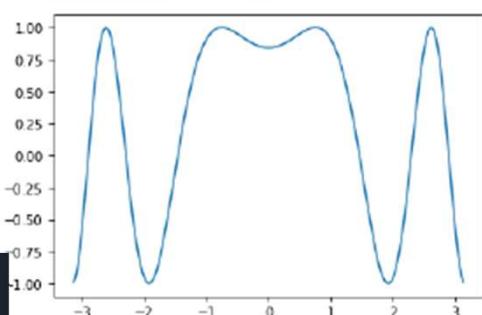
Pandas, numpy y matplotlib

Representación gráfica de la función $f(x)=\sin(x^2 + 1)$ en el intervalo $[-\pi, \pi]$

```

1 from numpy import linspace,sin,pi
2 import matplotlib.pyplot as plt
3 x=linspace(-pi,pi, 200)
4 y=sin(x**2+1)
5 plt.plot(x,y)
6 plt.show()
7

```



Mgr. Jose Luis Vera

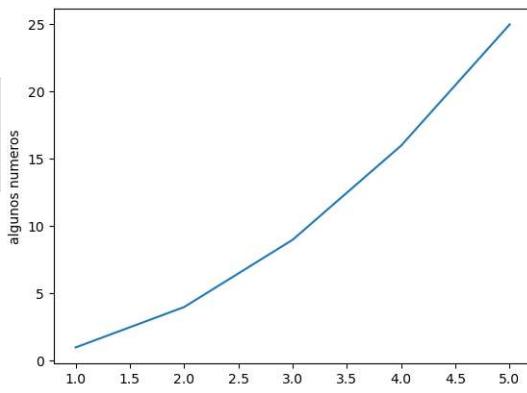
Clase 6-7

Pandas, numpy y matplotlib

```

1 import matplotlib.pyplot as plt
2 plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 25])
3 plt.ylabel('algunos numeros')
4 plt.show()

```



Mgr. Jose Luis Vera

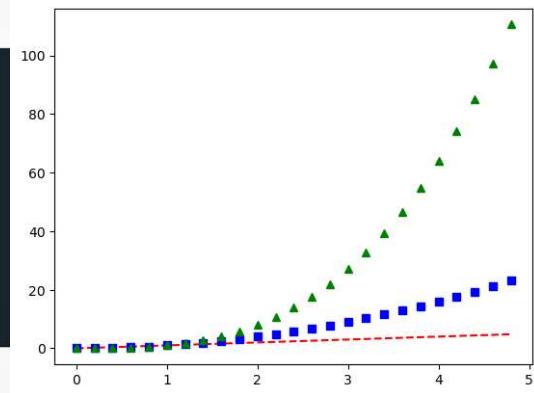
Clase 6-7

Pandas, numpy y matplotlib

```

1 # ejemplo con la libreria numpy
2 # colores, formas
3 # varias gráficas en uno
4 import numpy as np
5 import matplotlib.pyplot as plt
6 # tiempo muestreado uniformemente a intervalos de 200 ms
7 t = np.arange(0.0, 5.0, 0.2)
8
9 # guiones rojos, cuadrados azules y triángulos verdes
10 plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^') #
11 plt.show()
12

```



Mgr. Jose Luis Vera

Clase 6-7

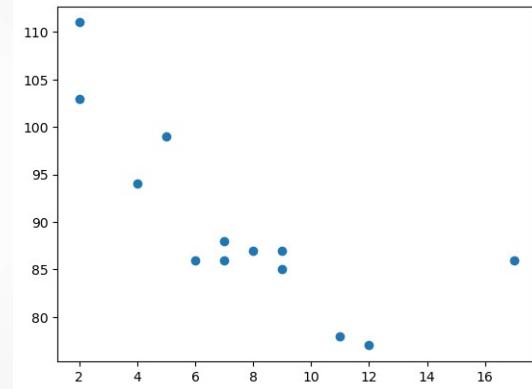
Pandas, numpy y matplotlib

Diagrama de dispersión (scatter)

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
5 y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
6
7 plt.scatter(x, y)
8 plt.show()
9

```



Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Diagrama de dispersión (scatter)

```

"""
ejemplos del uso de:
np.arange()
np.random.randint(0, 50, 25)
np.random.randn(25)
"""

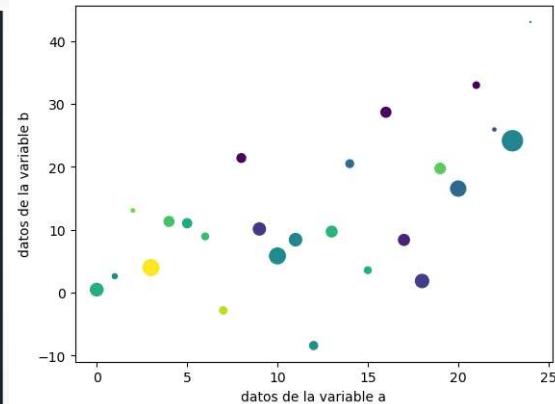
import numpy as np
import matplotlib.pyplot as plt

datos = {'a': np.arange(25),
          'c': np.random.randint(0, 50, 25),
          'd': np.random.randn(25)}

datos['b'] = datos['a'] + 10 * np.random.randn(25)
datos['d'] = np.abs(datos['d']) * 100

plt.scatter('a', 'b', c='c', s='d', data=datos)
plt.xlabel('datos de la variable a')
plt.ylabel('datos de la variable b')
plt.show()

```



Mgr. Jose Luis Vera

Clase 6-7

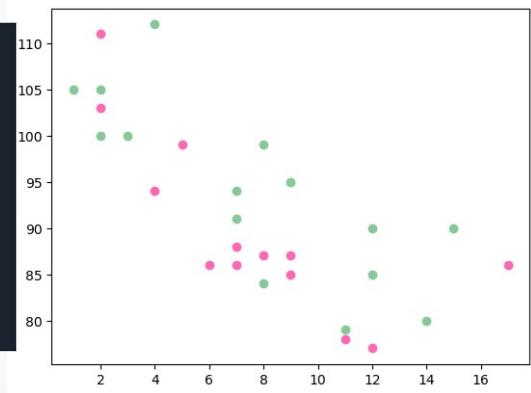
Pandas, numpy y matplotlib

Diagrama de dispersión (scatter)

```

1  """
2  ejemplo con uso de colores (hotpink) y RGB
3  """
4  import matplotlib.pyplot as plt
5  import numpy as np
6
7  x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
8  y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
9  plt.scatter(x, y, color = 'hotpink')
10
11 x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
12 y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
13 plt.scatter(x, y, color = '#88c999')
14
15 plt.show()

```



Mgr. Jose Luis Vera

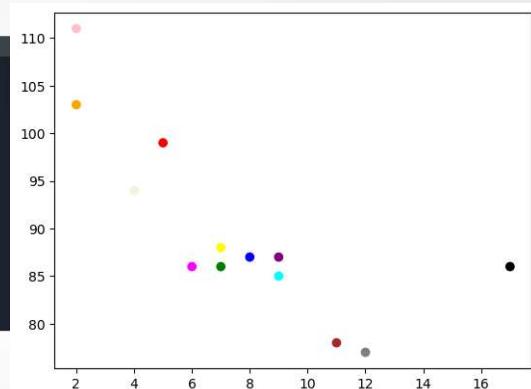
Clase 6-7

Pandas, numpy y matplotlib

```

1  """
2  # utilizacion de una lista definida de colores
3  """
4  import matplotlib.pyplot as plt
5  import numpy as np
6
7  x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
8  y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
9  mis_colores = np.array(['red','green','blue','yellow','pink',
10   'black','orange','purple','beige',
11   'brown','gray','cyan','magenta'])
12
13 plt.scatter(x, y, c=mis_colores)
14 plt.show()

```



Mgr. Jose Luis Vera

Clase 6-7

Pandas, numpy y matplotlib

Mapa de colores en python

```

1 """
2   mapa de colores que se puede utilizar
3 """
4
5 from matplotlib import colormaps
6 print(list(colormaps))
7

```

```
In [86]: runfile('C:/Users/jose_luis/Desktop/diploPython/ejemplosPY/Sin
titulo01.py', wdir='C:/Users/jose_luis/Desktop/diploPython/ejemplosPY')
['magma', 'inferno', 'plasma', 'viridis', 'cividis', 'twilight', 'twilight_shifted',
'turbo', 'Blues', 'BrBG', 'BuGn', 'BuPu', 'CMRmap', 'GnBu', 'Greens', 'Greys',
'OrRd', 'Oranges', 'PRGn', 'PiYG', 'PuBu', 'PuBuGn', 'PuOr', 'PuRd', 'Purples',
'RdBu', 'RdGy', 'RdPu', 'RdYlBu', 'RdYlGn', 'Reds', 'Spectral', 'Wistia', 'YlGn',
'YlGnBu', 'YlOrBr', 'YlOrRd', 'afmhot', 'autumn', 'binary', 'bone', 'brg', 'bwr',
'cool', 'coolwarm', 'copper', 'cubehelix', 'flag', 'gist_earth', 'gist_gray',
'gist_heat', 'gist_ncar', 'gist_rainbow', 'gist_stern', 'gist_yarg', 'gnuplot',
'gnuplot2', 'gray', 'hot', 'hsv', 'jet', 'nipy_spectral', 'ocean', 'pink', 'prism',
'rainbow', 'seismic', 'spring', 'summer', 'topographic', 'winter', 'Accent', 'Dark2']

```

Mgr. Jose Luis Vera

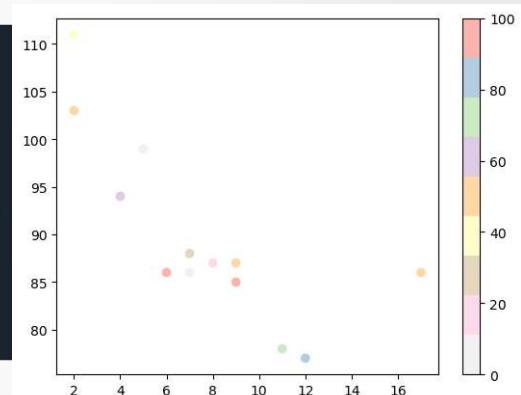
Clase 6-7

Pandas, numpy y matplotlib

```

1 """
2   utilización del mapa de colores viridis, Pastel1_r
3   despliegue del mapa de colores
4 """
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
9 y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
10 colores = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])
11
12 plt.scatter(x, y, c=colores, cmap='Pastel1_r')
13 #plt.scatter(x, y, c=colores, cmap='viridis')
14
15 plt.colorbar()
16
17 plt.show()

```



Mgr. Jose Luis Vera

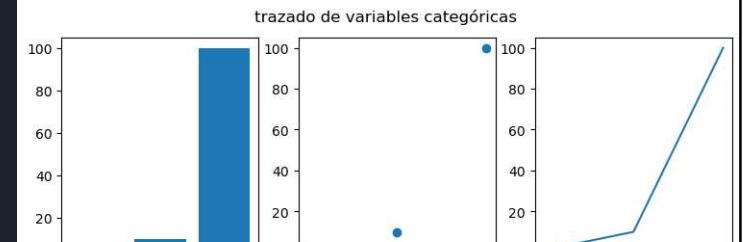
Clase 6-7

Pandas, numpy y matplotlib

```

1  """
2   graficas con variables categóricas
3
4 """
5 import matplotlib.pyplot as plt
6
7 categorias = ['grupo_a', 'grupo_b', 'grupo_c']
8 valores = [1, 10, 100]
9
10 plt.figure(figsize=(9, 3))
11
12 plt.subplot(131)
13 plt.bar(categorias, valores)
14 plt.subplot(132)
15 plt.scatter(categorias, valores)
16 plt.subplot(133)
17 plt.plot(categorias, valores)
18 plt.suptitle('trazado de variables categóricas')
19 plt.show()
20

```



Mgr. Jose Luis Vera

Tipos de datos/Métodos

FIN

Mgr. Jose Luis Vera