

This notebook will permit you to manipulate Python libraries three parts.

- Data Analysis and Manipulation (pandas)
- Data Preparation (Matplotlib)
- Data Visualization (NumPy)

▼ Part I - Pandas

Pandas is a fast, powerful, flexible and easy-to-use open source data analysis built on top of the Python programming language.

▼ Description of the data

Titanic Data Set

The data set that we will be using is taken from Kaggle. The "[titanic-train.csv](#)" <https://www.kaggle.com/c/titanic> file contains data for **897** of the real Titanic passengers. Each row represents one person

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

2. Whether they Survived

3. Passenger Class

4. Name

5. Sex

6. Age

7. Siblings Aboard

8. Parents Aboard

9. Ticket

10. Fare paid in £s

11. Cabin

12. Port of Embarkation

On **April 15, 1912**, the largest passenger liner ever made collided with an iceberg during her maiden voyage and killed **1502** out of **2224** passengers and crew. This sensational tragedy shocked the international shipping industry and led to new regulations for ships. One of the reasons that the shipwreck resulted in such loss of life was that the majority of passengers were steerage class. Although there was some element of luck involved in surviving the accident, it appears that men were more likely to survive than others.

```
# Importing libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline

# Connecting Google Drive Data Files to Google Colab Notebook.
# Load the Google Drive helper and mount the drive.
from google.colab import drive

# This will prompt for authorization. Click the link and provide the required information.

drive.mount('/content/drive')
```

☞ Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=9473

Enter your authorization code:
.....
Mounted at /content/drive

```
# List the content of YOUR google drive
!ls "/content/drive/My Drive"
```

☞

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

```
'01. Piste 1.mp3'  
'02. Piste 2.mp3'  
'03-04 Fichiers Repertoires Unix.pdf'  
_20160629_122354.JPG  
2016-08-14-18-31-04.amr  
2016-08-15-18-11-45.amr  
2016-08-15-18-33-32.amr  
2016-08-16-21-18-23.3gpp  
2016-08-17-19-03-29.3gpp  
2016-08-17-19-23-12.3gpp  
2016-08-29-18-03-05.amr  
2016-08-31-18-27-29.amr  
2016-09-01-18-42-54.amr  
2016-09-01-19-07-28.amr  
2016-09-04-18-01-50.amr  
2016-09-04-18-14-56.amr  
2016-09-06-12-14-15.amr  
2016-09-15-20-08-30.amr  
2016-09-26-15-50-48.amr  
2016-11-06-17-21-17.amr  
2016-11-06-17-30-05.amr  
2016-11-06-17-48-32.amr  
2016-11-07-17-57-07.amr  
2016-11-08-17-46-51.amr  
2016-11-08-17-58-39.amr  
2016-11-12-14-43-24.amr  
2016-11-12-15-33-06.amr  
2016-11-19-18-25-17.amr  
|^2i`.gslides  
aadaboul-masdjide.amr  
'adaboul masdjide-p11.amr'  
'adaboul-masdiide -o.amr'
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

```
afile_bassi.amr  
afile_bassi.mp3  
'AISSATOU FAYE (1).gslides'  
'AISSATOU FAYE (2).gslides'  
'AISSATOU FAYE.gslides'  
aj.mp3  
'ala ya r.amr'  
'ala yarasoul (2).amr'  
'ala yarasoul.amr'  
'ala yarasoulou.amr'  
'ala yarasouloul.3gpp'  
'ala ya rasouloul lahi koulli-zamanoune baka.amr'  
'ala ya rassoulala koulli.3gpp'  
Android  
Annale_S1S2_ndg.pdf  
'Attestation bac.pdf'  
Attestation_cheikh.pdf  
'ATTESTATION DE PASSAGE.jpeg'  
'attestation de passage.pdf 1.pdf'  
Attestation.jpg  
'Attestation passage 12.pdf'  
attestation.pdf  
'Attestation Reussite.pdf'  
b10.pdf  
b11.pdf  
b12.pdf
```

```
b13.pdf  
b14.pdf  
b15.pdf  
b1.pdf  
b2.pdf  
b3.pdf  
B5.pdf  
b6.pdf  
b7.pdf  
b8.pdf  
b9.pdf  
'baba Aidara.mp4'  
'BAC SENEGALAIS DE SCIENCES PHYSIQUES TOME 2 (2007-2012).pdf'  
'BAC TS1.pdf'  
'B$.pdf'  
'CamScanner-v5.14.0 (1).20191010_build_51400.apk'  
CamScanner-v5.14.0.20191010_build_51400.apk  
'Carte sans titre.gmap'  
"CERTIFICAT D'INSCRIPTION 15-16.jpeg"  
"CERTIFICAT D'INSCRIPTION.jpeg"  
'Chapitre 1 - COURS UML Dr M SALL.pdf'  
'Chapitre 2 - COURS UML Dr M SALL.pdf'  
'Chapitre 3 UML - 2018 Dr M SALL .pdf'  
cni1.pdf  
cni2.pdf  
colab  
'Colab Notebooks'  
'Copie de !`i` .gslides'  
'Copie de Team 8 - Product Backlog - Semences.gsheet'  
'Copie de Tutoriel - Cours Complet PhP & MySQL [Chapitre 4-27] - Les variables en Ph  
'~$CSS.ppt'  
cv.pdf  
...
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

```
Enfance.pdf  
'Enregistrer - Eamac.ne2.htm'  
'Enregistrer - Eamac.ne.htm'  
'Enregistrer - Eamac.ne.htm.gdoc'  
'fa_nadjina.amr'  
Formation.pdf  
'Formulaire sans titre.gform'  
'~$HTML.ppt'  
inscription_ch1.pdf  
Installer_serveur_LAMP_Ubuntu.pdf  
'interface graphique de base.pdf'  
'INTRODUCTION A GNU.pdf'  
'inux (2).pdf'  
JavaSwing.pdf  
kh.amr  
khantara.3gpp  
khantara.amr  
13ids-unix-exam1112.pdf  
'La gestion de projet.odt'  
'Langage C EPT.pdf'  
"Le manuel de l'utilisateur avancé de Ubuntu.pdf"  
Les_45_commandes_Linux_les_plus_utiles.pdf  
les_bases_de_l_administration_linux.pdf  
'Les commandes fondamentales de Linux.doc'  
'Les commandes fondamentales de Linux.pdf'  
'Les commandes.pdf'  
linux1.pdf
```

```
Linux_-_chap2_-_Système_de_fichiers.pps
'Linux embarqué (2e Ed).pdf'
linux-gestion-des-utilisateurs-646-k8qjjk.pdf
linux-initiation-et-utilisation.pdf
linux.pdf
'Linux sécuriser un réseau Edition 3.pdf'
linux_vi_f77.pdf
Liste-principale-et-Liste-dattente-DIC-Informatique.pdf
'LIVRE RESUME MATHS TS.pdf'
lpi101-v0.2-fr-v0.1-1.pdf
'~$madou makhtar mbaye est un étudiant de l.docx'
'Magix Video Deluxe 17 PREMIUM HD (2011) [App][French]'
'Mamadou makhtar mbaye IA Note.gdoc'
'Mamadou makhtar Mbaye_IA__Note.gdoc'
'MAMDOU MAKHTAR MBAYE-scanner'
mandoupe.amr
'manual_fr code block.pdf'
master.pdf
'Maximum clique.pdf'
mbaye1.pdf
mbaye2.pdf
MCD_Perso_groupe4.mcd
'memo_commandes_unix commandes.pdf'
'memo_commandes_unix.pdf'
memoir2
'mention genie logiciel.pdf'
'Menu restaurant.pdf'
"Michel Thomas - Harrap's Apprendre Anglais CD7 - Raccourci.lnk"
'!Mon cours d'\''algorithmique Chap1-Chap4.pdf'
'!Mon cours d'\''algorithmique Chap5.pdf'
'!Mon cours d'\''algorithmique Chap6.pdf'
'!Mon cours d'\''algorithmique Chap7.pdf'
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

```
'PHP Login Form with Sessions and MySQL.mp4'
'PHP Login Sessions.mp4'
'PolyExoc_MM langageC.pdf'
'Préparation à la certification LPIC-1 (examens LPI 101 et LPI 102) (Ed 2).pdf'
Présentation1.pptx
'prise en main de wamp.pdf'
processus.pdf
'programmation es interfaces graphiques .pdf'
'Programmation graphique avancé.pdf'
'programmation linéaire.mp4'
'Projet algo.pdf'
'Projet bataille naval.pdf'
'Projet de programmation C.pdf'
'Projet de Services réseaux L2 AMRT_D2AW 2017.pdf'
"Projet d'un système de réservation.pdf"
'Raccourci vers Wahab Diop-PHYSIQUE WPS-lsll.lnk'
Rapport_mini_projet_linux_embarque.pdf
'releve de note L1.pdf'
'releve de note L2.pdf'
'RELEVE DE NOTES.jpeg'
relevee_cheikh.pdf
releveel3.jpg
'Relevee L#.pdf'
'RELEVE L1.pdf'
'Releve L3.pdf'
R_L3.pdf
```

```

Scan.pdf
SE3.pdf
'Semence Distribution.gslides'
'Serveur DHCP sous Linux.pdf'
serveur_dns_simple.pdf

# In this case, data is stored in the directory described below
# Change the directory based on YOUR own Google Drive organization
df = pd.read_csv('/content/drive/My Drive/titanic-train.csv')
#pd.read_csv() -> Read the data from specified location and stores in dataframe 'df'

type(df) # Type is used to check the type of a variable. In this case, df is a DataFrame.

```

↳ pandas.core.frame.DataFrame

+ 123 rows × 12 columns

```
df.head() # Displaying the header which by default is the first 5 lines of the data.
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/545422
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	P/31432
2	3	1	3	Futrelle, Mrs. Jacques Heath (Br... Allen, Mr. William Henry	female	26.0	0	0	S/4337
3	4	1	1	Futrelle, Mrs. Jacques Heath (Br... Allen, Mr. William Henry	female	35.0	1	0	

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

```
df.info()
```

```
# Gives information about the Data Frame. This is a summary of the data containing the total number of rows and columns.
# We can also detect if there are missing data.
# object are string data
# Example in 'Age' the total number of rows data is 714 which means that there are missing values.
```

↳ <class 'pandas.core.frame.DataFrame'>

```

RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64 
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare          891 non-null    float64 
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

```

```
df.describe() # Statistical summary of numeric data - minimum, maximum, std, mean etc.
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	F
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329

▼ Indexing

Pandas allows to index data in various ways.

```
df.iloc[3] # To retrieve the records based on the index. The index starts with 0 in Python
```

PassengerId	4
Survived	1
Pclass	1

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

Age	33
SibSp	1
Parch	0
Ticket	113803
Fare	53.1
Cabin	C123
Embarked	S
Name:	3, dtype: object

```
df.loc[0:4,'Ticket'] # To retreive the data for one coloumn and several rows.  
# Here we get the Ticket colum of the first 5 rows.
```

0	A/5 21171
1	PC 17599
2	STON/O2. 3101282
3	113803
4	373450
	Name: Ticket, dtype: object

```
df['Ticket'].head() # You can get first five rows of the colum Ticket with the help of head
```

⇨

```
0          A/5 21171
1          PC 17599
2    STON/O2. 3101282
```

```
type(df[['Embarked', 'Ticket']]) # It is useful to check the types of expressions before u
```

```
↳ pandas.core.frame.DataFrame
```

```
df[['Embarked', 'Ticket']].head() # You can get multiple column data by providing the lis
#Here you get data of the 'Embarked' and 'Tickets' column
```

	Embarked	Ticket
0	S	A/5 21171
1	C	PC 17599
2	S	STON/O2. 3101282
3	S	113803
4	S	373450

▼ Selections

Pandas allows to perform different selections.

```
df[df['Age'] > 65] # We can get the data of passengers whose age is above 65. To get the c
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0

```
df['Age'] > 65 # This statement displays if the condition is True or False.
```

```
↳ 0    False
1    False
2    False
3    False
4    False
...
886   False
887   False
888   False
889   False
890   False
Name: Age, Length: 891, dtype: bool
```

```
df.query('Age > 65') # Query can also be used to select the data based on the condition p
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0

```
df[(df['Age'] == 10) & (df['SibSp'] == 5)] # Conditions can be combined with Boolean oper
# In this example we have combined two conditi
# Condition : Selection of passengers with 'Age
# The condition is False so nothing is display
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E

```
df[(df.Age == 10) | (df.SibSp == 5)] # Here we are using OR.
```

```
↳
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parcl
59	60	0	3	Goodwin, Master. William Frederick	male	11.0	5	;
71	72	0	3	Goodwin, Miss. Lillian Amy	female	16.0	5	;
386	387	0	3	Goodwin, Master. Sidney Leonard	male	1.0	5	;
419	420	0	3	Van Impe, Miss. Catharina	female	10.0	0	;

```
df.query('Age == 10 | (SibSp == 5)') # Similarly, we can use query.
```

↪	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parcl
59	60	0	3	Goodwin, Master. William Frederick	male	11.0	5	;
71	72	0	3	Goodwin, Miss. Lillian Amy	female	16.0	5	;
386	387	0	3	Goodwin, Master. Sidney Leonard	male	1.0	5	;
419	420	0	3	Van Impe, Miss. Catharina	female	10.0	0	;
480	481	0	3	Goodwin, Master. Harold Victor	male	9.0	5	;
683	684	0	3	Goodwin, Mr. Charles Edward	male	14.0	5	;

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

▼ Unique Values

```
df['Embarked'].unique() # We can get the unique values from a column.
```

```
↪ array(['S', 'C', 'Q', nan], dtype=object)
```

▼ Sorting

```
df.sort_values('Age', ascending = False).head() # Sort the data in DataFrame either in asc
```

```
↪
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
630	631	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0

▼ Aggregations

Like in SQL, Pandas allows you to do aggregations and group by.

```
df['Survived'].value_counts() # You can aggregate the data to get the number of survivors.  
# '0's' and '1's' values in the Survived field determine if
```

```
0    549  
1    342  
Name: Survived, dtype: int64
```

```
df['Pclass'].value_counts() # Number of people per class
```

```
3    491  
1    216  
2    184  
Name: Pclass, dtype: int64
```

```
df.groupby(['Pclass', 'Survived'])['PassengerId'].count() # Number of survivors per class
```

```
Pclass  Survived  
1      0          80
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

```
3    0        372  
1    1        119  
Name: PassengerId, dtype: int64
```

```
df['Age'].min()
```

```
0.42
```

```
df['Age'].max()
```

```
80.0
```

```
df['Age'].mean()
```

```
29.69911764705882
```

```
df['Age'].median()
```

```
28.0
```

```
# Information about the survivors
```

```
mean_age_by_survived = df.groupby('Survived')['Age'].mean()
mean_age_by_survived
```

```
↳ Survived
 0    30.626179
 1    28.343690
Name: Age, dtype: float64
```

```
std_age_by_survived = df.groupby('Survived')['Age'].std()
std_age_by_survived
```

```
↳ Survived
 0    14.172110
 1    14.950952
Name: Age, dtype: float64
```

```
min_age_by_survived = df.groupby('Survived')['Age'].min()
min_age_by_survived
```

```
↳ Survived
 0    1.00
 1    0.42
Name: Age, dtype: float64
```

```
max_age_by_survived = df.groupby('Survived')['Age'].max()
max_age_by_survived
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff...](#)

```
Name: Age, dtype: float64
```

▼ Merging Two Data Frames.

```
df1 = mean_age_by_survived.round(0).reset_index()
df2 = std_age_by_survived.round(0).reset_index()
```

```
df1
```

	Survived	Age
0	0	31.0
1	1	28.0

```
df2
```

```
↳
```

Survived	Age
----------	-----

0	0	14.0
---	---	------

```
df3 = pd.merge(df1, df2, on='Survived')
```

df3

	Survived	Age_x	Age_y
--	----------	-------	-------

0	0	31.0	14.0
---	---	------	------

1	1	28.0	15.0
---	---	------	------

```
df3.columns = ['Survived', 'Age Mean', 'Age Standard Deviation']
```

df3

	Survived	Age Mean	Age Standard Deviation
--	----------	----------	------------------------

0	0	31.0	14.0
---	---	------	------

1	1	28.0	15.0
---	---	------	------

▼ Pivot Tables

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

```
df.pivot_table(index='Pclass',
               columns='Survived',
               values='PassengerId',
               aggfunc='count')
```

	Survived	0	1
--	----------	---	---

Pclass		
--------	--	--

1	80	136
---	----	-----

2	97	87
---	----	----

3	372	119
---	-----	-----

▼ Correlations

In Pandas you can also calculate correlations between features.

```
correlated_with_survived = df.corr()['Survived'].sort_values()
correlated_with_survived
```

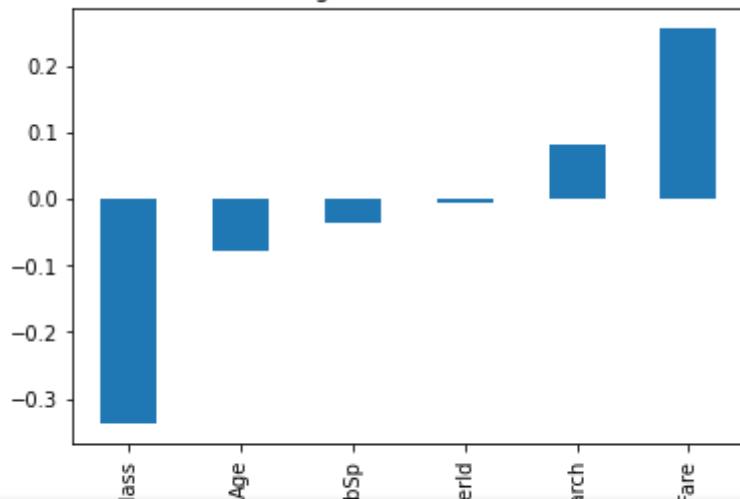
```
↳ Pclass      -0.338481
    Age       -0.077221
    SibSp     -0.035322
    PassengerId -0.005007
    Parch      0.081629
    Fare       0.257307
    Survived   1.000000
Name: Survived, dtype: float64
```

```
%matplotlib inline
```

```
correlated_with_survived.iloc[:-1].plot(kind='bar',
                                         title='Titanic Passengers: correlation with survival')
```

```
↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f316a1f3ef0>
```

Titanic Passengers: correlation with survival



Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

▼ Part II - Visual Data Exploratory Analysis with Matplotlib

Matplotlib is a Python 2D plotting library which produces quality figures in a variety of hardcopy formats.

```
# Generating some random data for visualization!
data1 = np.random.normal(0, 0.1, 1000)
data2 = np.random.normal(1, 0.4, 1000) + np.linspace(0, 1, 1000)
data3 = 2 + np.random.random(1000) * np.linspace(1, 5, 1000)
data4 = np.random.normal(3, 0.2, 1000) + 0.3 * np.sin(np.linspace(0, 20, 1000))

data = np.vstack([data1, data2, data3, data4]).transpose() #with transpose() - Transposing
```



```
df = pd.DataFrame(data, columns=['data1', 'data2', 'data3', 'data4'])
df.head()
```

```
↳
```

```
      data1     data2     data3     data4
0 -0.158278  0.486991  2.905003  2.986280
1 -0.000464  1.095844  2.074239  3.111881
2 -0.007073  1.232826  2.746732  2.851716
```

```
df.info()
```

```
↳ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   data1    1000 non-null    float64
 1   data2    1000 non-null    float64
 2   data3    1000 non-null    float64
 3   data4    1000 non-null    float64
dtypes: float64(4)
memory usage: 31.4 KB
```

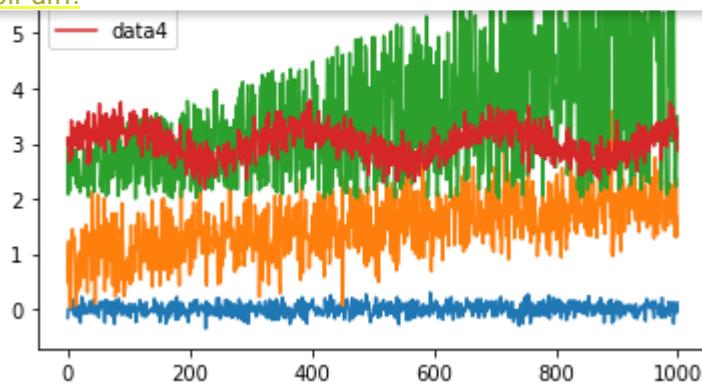
▼ Line Plot

```
df.plot(title='Line plot')
```

```
↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f316a148e48>
Line plot
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

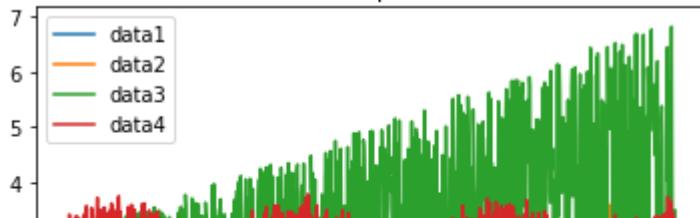


```
plt.plot(df)
plt.title('Line plot')
plt.legend(['data1', 'data2', 'data3', 'data4'])
```

```
↳
```

```
<matplotlib.legend.Legend at 0x7f3169b3f860>
```

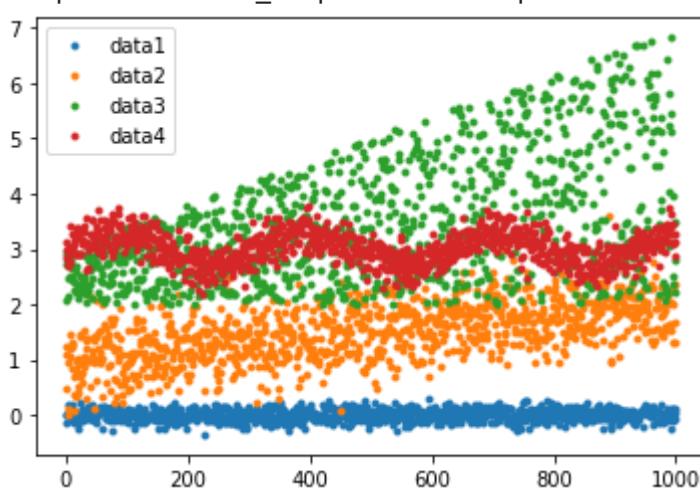
Line plot



▼ Scatter Plot

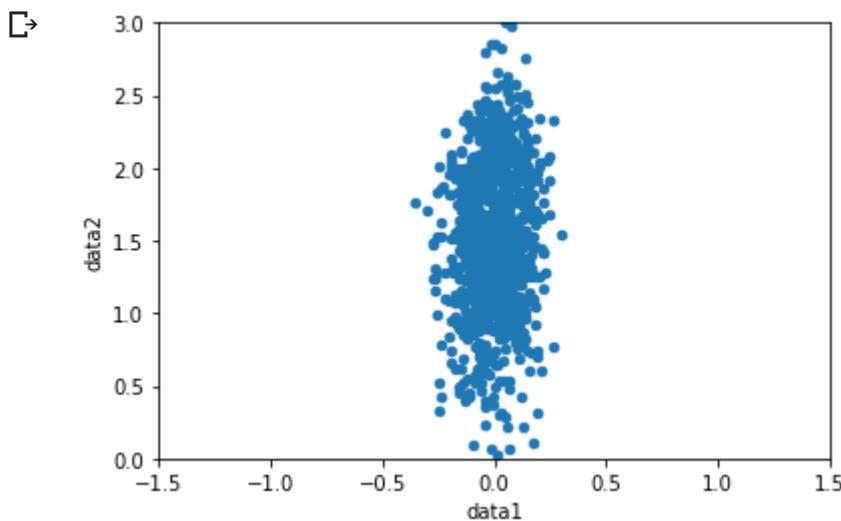
```
1 ↴ df.plot(style='.')
```

```
↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f3169aab128>
```



Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

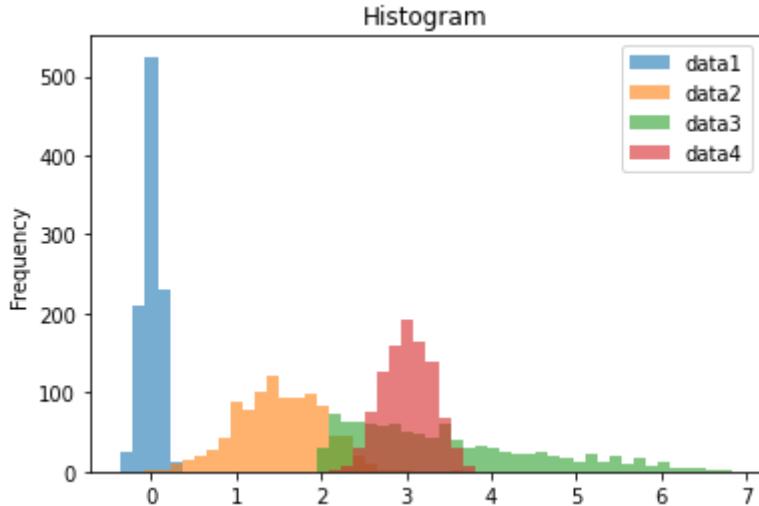


▼ Histograms

```
df.plot(kind='hist',
        bins=50,
        title='Histogram',
```

```
alpha=0.6)
```

↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f31659ad198>

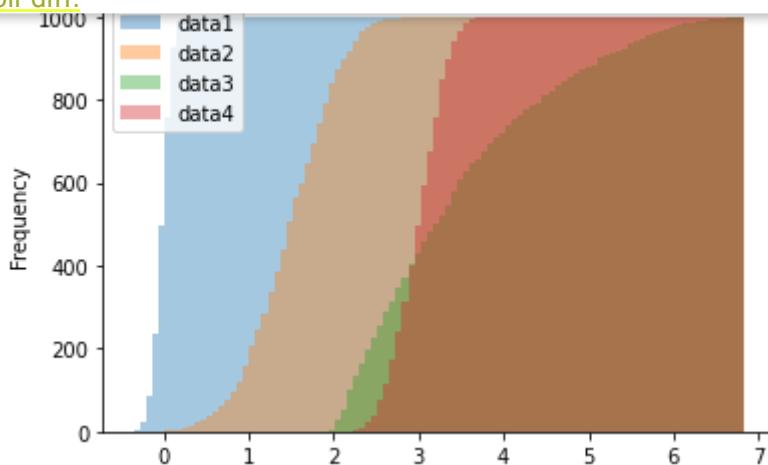


▼ Cumulative distribution

```
df.plot(kind='hist',
        bins=100,
        title='Cumulative distributions',
        #normed=True,
        cumulative=True,
        alpha=0.4)
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

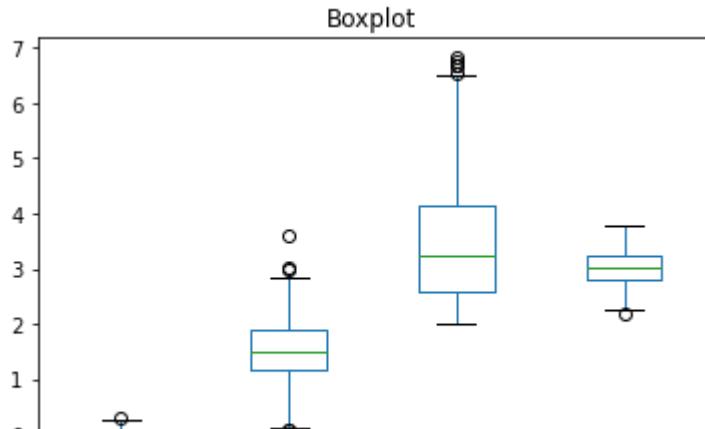


▼ Box Plot

```
df.plot(kind='box',
        title='Boxplot')
```

↳

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3165790f98>
```



▼ Subplots

```
fig, ax = plt.subplots(2, 2, figsize=(5, 5))
```

```
df.plot(ax=ax[0][0],  
        title='Line plot')
```

```
df.plot(ax=ax[0][1],  
        style='o',  
        title='Scatter plot')
```

```
df.plot(ax=ax[1][0],  
        kind='hist',  
        bins=50,
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)
~~df.plot(ax=ax[1][1],
 kind='box',
 title='Boxplot')~~

```
plt.tight_layout()
```



Line plot

Scatter plot

▼ Pie charts

```
data4 | data4 |
```

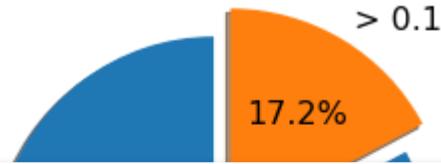
```
gt01 = df['data1'] > 0.1
piecounts = gt01.value_counts()
piecounts
```

```
False    828
True     172
Name: data1, dtype: int64
```

```
| | data3 | | |
```

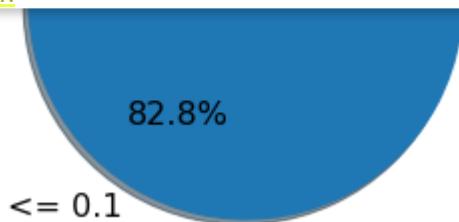
```
piecounts.plot(kind='pie',
                figsize=(5, 5),
                explode=[0, 0.15],
                labels=['<= 0.1', '> 0.1'],
                autopct='%1.1f%%',
                shadow=True,
                startangle=90,
                fontsize=16)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3164e70550>
```



Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)



▼ Hexbin plot

```
data = np.vstack([np.random.normal((0, 0), 2, size=(1000, 2)),
                 np.random.normal((9, 9), 3, size=(2000, 2))])
df = pd.DataFrame(data, columns=['x', 'y'])
```

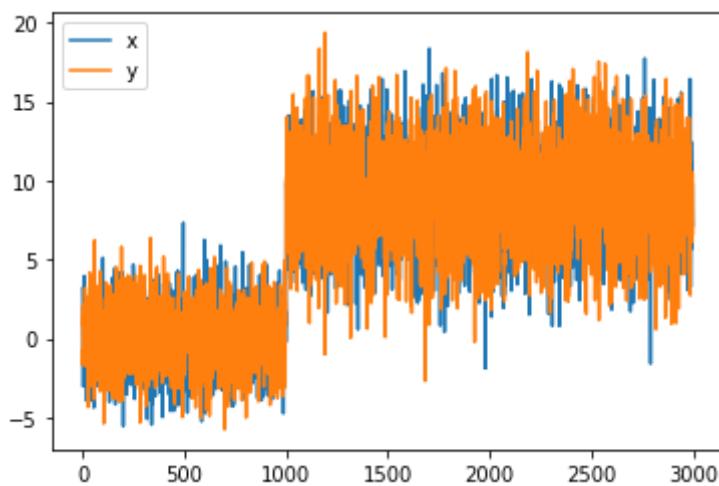
```
df.head()
```

```
df
```

	x	y
0	3.179026	-0.777404
1	1.943804	-1.696777
2	-0.937212	2.072441

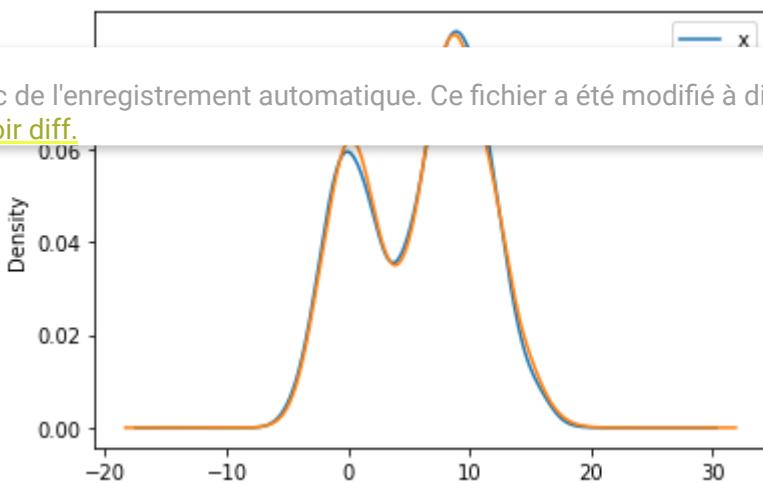
```
df.plot()
```

```
↪ <matplotlib.axes._subplots.AxesSubplot at 0x7f3165212cf8>
```



```
df.plot(kind='kde')
```

```
↪ <matplotlib.axes._subplots.AxesSubplot at 0x7f3164dca908>
```



```
df.plot(kind='hexbin', x='x', y='y', bins=100, cmap='rainbow')
```

```
↪
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f315b55dd68>
```



▼ Part III - Numpy

▼ Linear Algebra with Numpy

```
| [REDACTED] | [REDACTED] | [REDACTED]
```

```
a = np.array([1, 3, 2, 4])
```

```
a
```

```
↳ array([1, 3, 2, 4])
```

```
type(a)
```

```
↳ numpy.ndarray
```

```
A = np.array([[3, 1, 2],  
             [2, 3, 4]])
```

```
B = np.array([[0, 1],  
             [2, 3],
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

```
C = np.array([[0, 1],  
             [2, 3],  
             [4, 5],  
             [0, 1],  
             [2, 3],  
             [4, 5]])
```

```
print("A is a {} matrix".format(A.shape))  
print("B is a {} matrix".format(B.shape))  
print("C is a {} matrix".format(C.shape))  
print("A is a " + str(A.shape) + " matrix")
```

```
↳ A is a (2, 3) matrix  
B is a (3, 2) matrix  
C is a (6, 2) matrix  
A is a (2, 3) matrix
```

```
A[0]
```

```
↳ array([3, 1, 2])
```

```
C[2, 0]
```

```
↳ 4
```

```
B[:, 0]
```

```
↳ array([0, 2, 4])
```

▼ Elementwise operations

```
3 * A
```

```
↳ array([[ 9,  3,  6],  
         [ 6,  9, 12]])
```

```
A + A
```

```
↳ array([[6, 2, 4],  
         [4, 6, 8]])
```

```
A * A
```

```
↳ array([[ 9,  1,  4],  
         [ 4,  9, 16]])
```

```
A / A
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.
[Voir diff.](#)

```
A - A
```

```
↳ array([[0, 0, 0],  
         [0, 0, 0]])
```

▼ Dot product

```
A.shape
```

```
↳ (2, 3)
```

```
B.shape
```

```
↳ (3, 2)
```

```
A.dot(B) # Product of matrices
```

```
↳ array([[10, 16],  
         [22, 31]])
```

```
np.dot(A, B)
```

```
↳ array([[10, 16],  
         [22, 31]])
```

```
B.dot(A)
```

```
↳ array([[ 2,  3,  4],  
         [12, 11, 16],  
         [22, 19, 28]])
```

```
C.shape
```

```
↳ (6, 2)
```

```
A.shape
```

```
↳ (2, 3)
```

```
C.dot(A)
```

```
↳ array([[ 2,  3,  4],  
         [12, 11, 16],  
         [22, 19, 28],  
         [ 2,  3,  4],  
         [12, 11, 16],  
         [22, 19, 28]])
```

Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)

~~Unstructured data, images, sounds etc.~~

Here we will explore images and sounds.

▼ Images

```
from PIL import Image  
  
# Change the path to YOUR image  
img = Image.open('/content/drive/My Drive/WTC.jpg')  
img
```

```
↳
```



Échec de l'enregistrement automatique. Ce fichier a été modifié à distance ou dans un autre onglet.

[Voir diff.](#)



```
type(img)
```

↳ `PIL.JpegImagePlugin.JpegImageFile`



```
imgarray = np.asarray(img)
```

```
type(imgarray)
```

↳ `numpy.ndarray`