

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Lire les csv exportés de notre base de données (AirTable)
BASE_DIR = 'Resources/'

land = pd.read_csv(BASE_DIR + 'type de sol-Grid view.csv')
plant = pd.read_csv(BASE_DIR+ 'BasePlants-All projects.csv')
rain = pd.read_csv(BASE_DIR+ 'Précipitation2022-Grid view.csv')

#####
# DataFrames #
# #####
# plant
# land
# rain

plant

      Plantes  besoin en eau (m³/ha)  Durée ( en jours ) \
0          Le blé          5500          270
1  La betterave sucrière          6500           60
2        Le tournesol          1000          130
3          Le soja          6370          130
4        Le maïs          4000          180
5        Le sorgho          4750          130
6      Les Vignes          2150          335

      Racine (m)    Kc  ETRef (mm)  besoin en eau annuel (m³/ha) \
0          1.50  0.90          4.0          NaN
1          0.35  0.80          4.0          NaN
2          2.00  0.60          4.0        2800.0
3          1.00  0.70          4.0          NaN
4          0.45  0.80          4.0        8100.0
5          1.00  0.85          4.0       13330.0
6          3.00  0.55          4.0          NaN

      Précision
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
5          NaN
6  Le sol le plus adapté à la vigne sera pierreux...

land

      Type de sol  RU (Réserve Utile d'eau) \
0          Argiles          1.85

```

1	Limons	1.80
2	Sables	0.70
3	Argiles - sableuses	1.65
4	Argiles - limoneux	1.95
5	Limons - sableux	1.25
6	Argiles - sablo - limoneux	1.70
7	NaN	NaN
8	NaN	NaN

Notes

0	Retiens de façon efficace l'eau / une terre du...
1	Intermédiaire entre argiles et sables
2	Séchent facilement donc les apports doivent être...
3	NaN
4	NaN
5	NaN
6	NaN
7	Plus le sol est sableux, plus l'infiltration est...
8	plus le sol est argileux, plus il retient l'eau

rain

	Mois	Précipitations
0	1	39.6
1	2	48.9
2	3	34.4
3	4	62.4
4	5	19.1
5	6	99.8
6	7	3.0
7	8	26.3
8	9	39.4
9	10	6.4
10	11	NaN
11	12	NaN

#####

# Formulaire #

#####

# Tous les variables qui commencent par 'enter\_' sont les données de formulaire par utilisateur

enter\_plant = "Le maïs"

enter\_land = "Argiles"

enter\_nb\_day=150

enter\_surface = 15689 # m2

# Simulation - recherche la donnée

plant\_searched = plant.loc[plant['Plantes'] == enter\_plant]

plant\_searched

	Plantes	besoin en eau (m³/ha)	Durée ( en jours )	Racine (m)	Kc
4	Le maïs	4000	180	0.45	0.8

	ETRef (mm)	besoin en eau annuel (m³/ha)	Précision
4	4.0	8100.0	NaN

*# Simulation - recherche la donnée*

```
land_searched = land.loc[land['Type de sol'] == enter_land]
```

land\_searched

	Type de sol	RU (Réserve Utile d'eau)	\
0	Argiles	1.85	

Notes

0 Retiens de façon efficace l'eau / une terre du...

```
rain_from_2010 = rain.loc[rain['Année']>=2010]
```

*# Précipitations à Mérignac (2010 - 2022)*

rain\_from\_2010

	Année	Précipitations
10	2010	762.8
11	2011	579.7
12	2012	814.3
13	2013	1027.5
14	2014	945.2
15	2015	559.4
16	2016	1040.0
17	2017	762.3
18	2018	763.1
19	2019	971.8
20	2020	1149.6
21	2021	884.0
22	2022	379.3

*# Getteur plant, land objet par son nom*

```
def get_plant(enter_plant):
    return plant.loc[plant['Plantes'] == enter_plant]
```

```
def get_land(enter_land):
    return land.loc[land['Type de sol'] == enter_land]
```

*# Getteur les attributs 'Plantes', 'Durée en jours', 'ru - réserve utile d'eau', 'pr - profondeur racine', 'besoin en eau' par Getteur plant, land objet*

```
def get_duration(enter_plant):
    return get_plant(enter_plant)["Durée ( en jours )"]
```

```

def get_name(enter_plant):
    return get_plant(enter_plant)["Plantes"]

def get_pr(enter_plant):
    return float(get_plant(enter_plant)["Racine (m)"])

def get_ru(enter_land):
    return float(get_land(enter_land)["RU (Réserve Utile d'eau)"])

def get_need_water(enter_plant):
    return float(get_plant(enter_plant)["besoin en eau"])

```

```
get_pr("Le maïs")
```

```
0.45
```

```
get_ru("Argiles")
```

```
1.85
```

```
# 1. Stockage d'eau (m -> mm)
```

```
def stock_water(ru, pr):
    return pr*(2/3)*ru*1000
```

```
# (/10000 : m2 -> h)
```

```
def volume_stock(stock_water, enter_surface):
    return stock_water*(enter_surface/10000)*10
```

```
corn_pr = get_pr("Le maïs")
argiles_ru = get_ru("Argiles")
```

```
stock = stock_water(corn_pr, argiles_ru)
current_volume = volume_stock(stock, enter_surface)
```

```
#####
# Result #
#####
```

```
print("stockage d'eau : " +str(stock) + " mm")
print("volume de stockage : " +str(current_volume) + " m3")
```

```
stockage d'eau : 555.0 mm
volume de stockage : 8707.395 m3
```

```
# 2. User saisit le nombre de jour lors de remplir le formulaire
```

```
def need_of_water(enter_nb_day, kc, etref):
    return float(enter_nb_day*kc*etref)
```

```
def get_kc(enter_plant):
    return float(get_plant(enter_plant)["Kc"])
```

```

def get_etref(enter_plant):
    return float(get_plant(enter_plant)["ETRef (mm)"])

kc = get_kc("Le maïs")
etref = get_etref("Le maïs")

water = need_of_water(enter_nb_day,kc,etref)
surface = enter_surface/10000

#####
# Result #
#####

print(str(water) + " mm")
# 1 mm = 10 m3/ha

480.0 mm

# 3. Volume d'eau nécessaire

def volume_need_for_plant(water,enter_surface):

    return water*(enter_surface/10000)*10

needed_plant_volume = volume_need_for_plant(water, enter_surface)

#####
# Result #
#####

print(str(needed_plant_volume) + " m3")

7530.72 m3

# -- Comparaison -- (1) volume_stock , (3) volume_need_for_plant
# (4) Checker pour l'irrigation si besoin

def check_irrigation_needed(current_volume,needed_plant_volume):
    if(current_volume < needed_plant_volume):
        return True
    else:
        return False

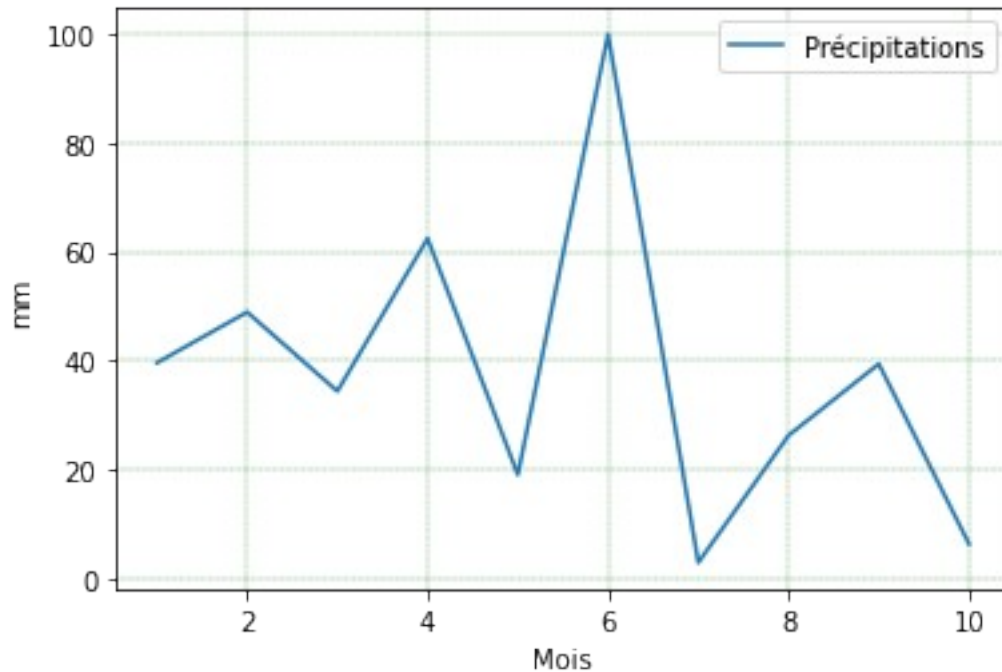
if check_irrigation_needed(current_volume,needed_plant_volume):
    print("Il faut arroser")
else:
    print("Pas besoin d'irriguer")

Pas besoin d'irriguer

```

```
# (5) Précipitations mensuelles 2022
```

```
rain_graph = rain.plot(x="Mois", y=["Précipitations"])  
rain_graph.set_ylabel("mm")  
rain_graph.grid(color = 'green', linestyle = '--', linewidth = 0.3)
```



```
# En moyen, "{get_name(enter_plant)}" consomme  
"{get_need_water(enter_plant)}" (m³/ha)  
# et dure "{get_duration(enter_plant)}" jour(s).
```