

Tech Challenge

Data Science

Sommaire

1. Analyse descriptive exploratoire (EDA)
2. Trouver la boutique correspondante
3. Explication de l'impact de la Météo
4. Prévion des ventes

Pour cette analyse j'utiliserai les bibliothèques suivantes

1. Pandas (https://pandas.pydata.org/docs/user_guide/index.html)
2. Datetime (<https://docs.python.org/fr/3/library/datetime.html>)
3. Matplotlib (<https://matplotlib.org/stable/index.html>)
4. Pandas_profiling
(https://pandas-profiling.ydata.ai/docs/master/pages/getting_started/quickstart.html)

1- Analyse descriptive exploratoire (EDA)

Overview

Overview

Alerts **3**

Reproduction

Dataset statistics

Number of variables	4
Number of observations	520
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	16.4 KiB
Average record size in memory	32.2 B

Variable types

DateTime	1
Categorical	2
Numeric	1

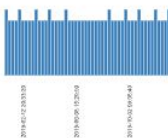
Variables

DATE

Date

Distinct	260
Distinct (%)	50.0%
Missing	0
Missing (%)	0.0%
Memory size	4.2 KiB

Minimum	2019-01-02 00:00:00
Maximum	2019-12-29 00:00:00



Toggle details

ITEM

Categorical

UNIFORM

Distinct	2
Distinct (%)	0.4%
Missing	0
Missing (%)	0.0%
Memory size	4.2 KiB

A	260
B	260

Toggle details

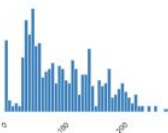
SALES

Real number (R₆₄)

ZEROS

Distinct	185
Distinct (%)	35.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	98.20769231

Minimum	0
Maximum	282
Zeros	23
Zeros (%)	4.4%
Negative	0
Negative (%)	0.0%
Memory size	4.2 KiB



Toggle details

DAY

Categorical

UNIFORM

Distinct	5
Distinct (%)	1.0%
Missing	0
Missing (%)	0.0%
Memory size	4.2 KiB

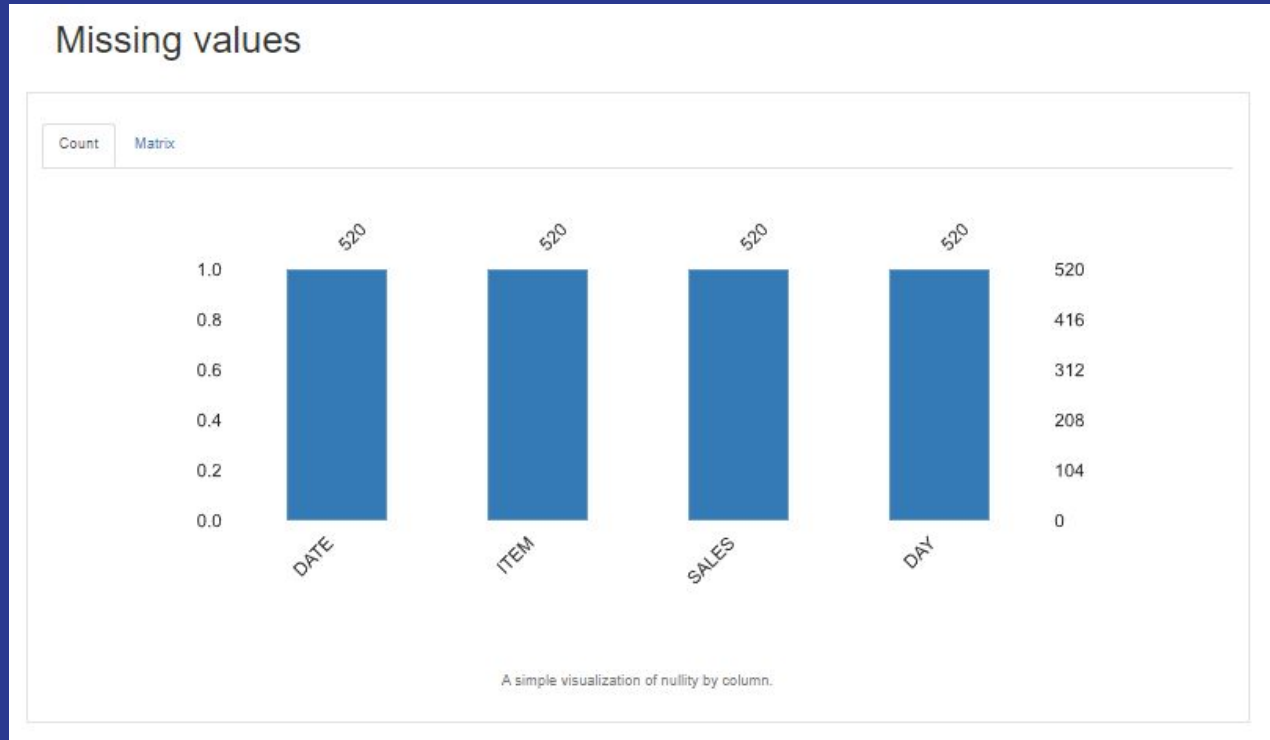
Wednesday	104
Thursday	104
Friday	104
Saturday	104
Sunday	104

Toggle details

Grâce à la bibliothèque “datetime”, je peux retourner le jour de la semaine correspondant.

je constate que le magasin est ouvert du Mercredi au Dimanche

Aucune valeur n'est manquante. Le data frame (tableau des données) est complet.



Completude des données

```
df['ITEM'].unique()
```

✓ 0.6s

```
array(['A', 'B'], dtype=object)
```

```
df['ITEM'].describe() ⓘ
```

✓ 0.4s

count 520

unique 2

top A

freq 260

Name: ITEM, dtype: object

```
print('Nombre de valeurs manquantes : ', df['ITEM'].isnull().sum())
```

✓ 0.4s

Nombre de valeurs manquantes : 0

```
df.info()
```

✓ 0.4s

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

DatetimeIndex: 520 entries, 2019-01-02 to 2019-12-29

Data columns (total 3 columns):

#	Column	Non-Null Count	Dtype
0	ITEM	520 non-null	object
1	SALES	520 non-null	int64
2	DAY	520 non-null	object

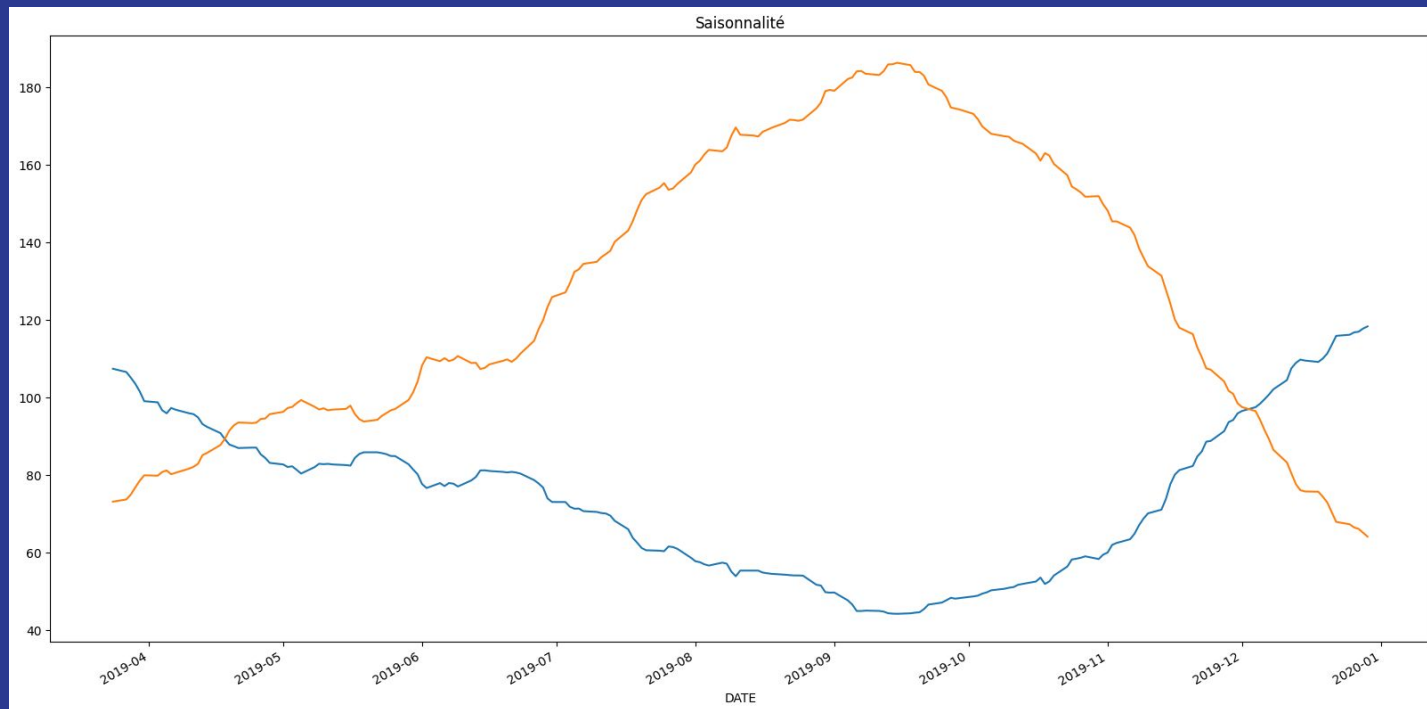
dtypes: int64(1), object(2)

memory usage: 16.2+ KB

Données brutes sans mise en forme avec tableau de bord

Pour repérer la saisonnalité suivant les Items, j'ai lissé les moyennes des ventes, afin d'avoir un graphique qui représente les tendances de ventes.

On peut voir que les 2 items sont opposés, quand "item A" se vend bien, "item B" est en baisse et inversement.



2- Trouver la boutique correspondante

```

meteo_bdx= pd.read_csv("https://raw.githubusercontent.com/Djez-Himself/meteo/master/meteo_bordeaux.csv", sep=',',usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', 'WINDSPEED_MAX_KMH', 'OPINION','PRECIP_TOTAL_DAY_MM'])
meteo_bdx['DATE'] = pd.to_datetime(meteo_bdx['DATE'], format="%Y-%m-%d") #transformation formate date
meteo_bdx = meteo_bdx.set_index(meteo_bdx['DATE']) #Modif index en Date
del meteo_bdx['DATE']#suppression de la colonne DATE

meteo_lyon= pd.read_csv("https://raw.githubusercontent.com/Djez-Himself/meteo/master/meteo_lyon.csv", sep=',',usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', 'WINDSPEED_MAX_KMH', 'OPINION','PRECIP_TOTAL_DAY_MM'])
meteo_lyon['DATE'] = pd.to_datetime(meteo_lyon['DATE'], format="%Y-%m-%d")
meteo_lyon = meteo_lyon.set_index(meteo_lyon['DATE'])
del meteo_lyon['DATE']

meteo_lille= pd.read_csv("https://raw.githubusercontent.com/Djez-Himself/meteo/master/meteo_lille.csv", sep=',',usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', 'WINDSPEED_MAX_KMH', 'OPINION','PRECIP_TOTAL_DAY_MM'])
meteo_lille['DATE'] = pd.to_datetime(meteo_lille['DATE'], format="%Y-%m-%d")
meteo_lille = meteo_lille.set_index(meteo_lille['DATE'])
del meteo_lille['DATE']

meteo_marseille=pd.read_csv("https://raw.githubusercontent.com/Djez-Himself/meteo/master/meteo_marseille.csv", sep=',',usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', 'WINDSPEED_MAX_KMH', 'OPINION','PRECIP_TOTAL_DAY_MM'])
meteo_marseille['DATE'] = pd.to_datetime(meteo_marseille['DATE'], format="%Y-%m-%d")
meteo_marseille = meteo_marseille.set_index(meteo_marseille['DATE'])
del meteo_marseille['DATE']

```

1. usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', etc] permet de choisir les colonnes à afficher pour trier les infos pertinentes.
2. Transformation de la donnée 'DATE' en format date
3. Remplacement de l'index par la 'DATE'
4. Suppression de la colonne 'DATE' pour éviter la répétition avec l'index.

Comparaison *meteo_bdx* avec vente item_A et Item B



















```
##Item A
merge_bdx_a = pd.DataFrame.join(meteo_bdx, df_a) # join 2 df grace a la fonction pandas .join(self='', other='')
del merge_bdx_a['ITEM'] #suppresion de la colonne item.
merge_bdx_a=merge_bdx_a.dropna(how = 'any') #Suppression des Valeurs 'NaN' ('DAY' non present sur le dataframe meteo, je peux donc supprimer les lignes pour faire correspondre les 2 df)

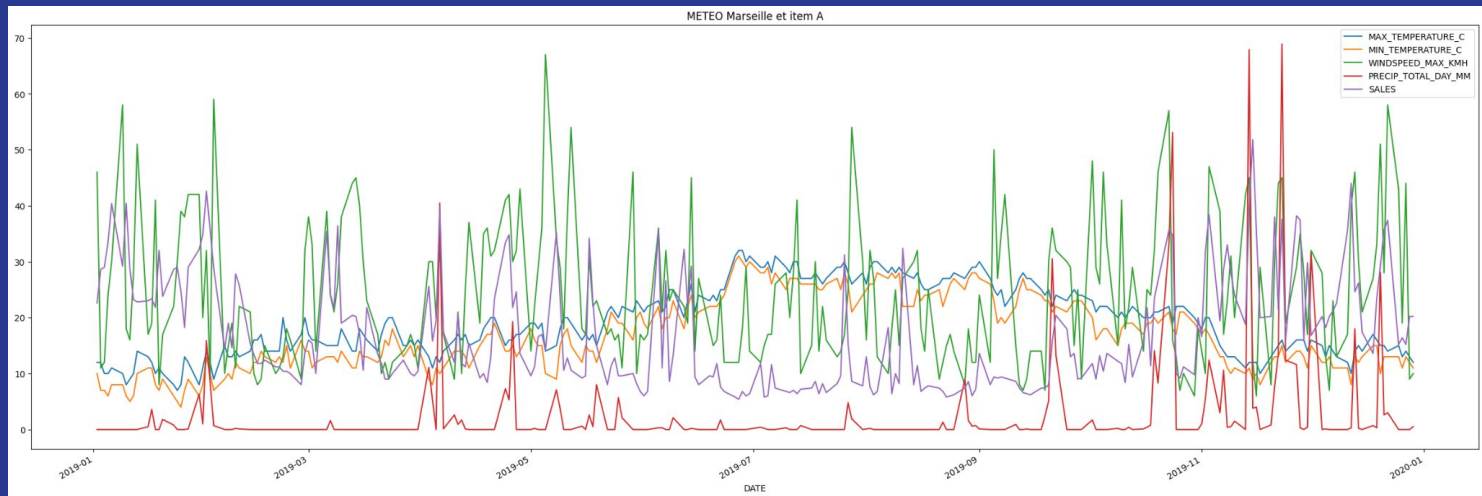
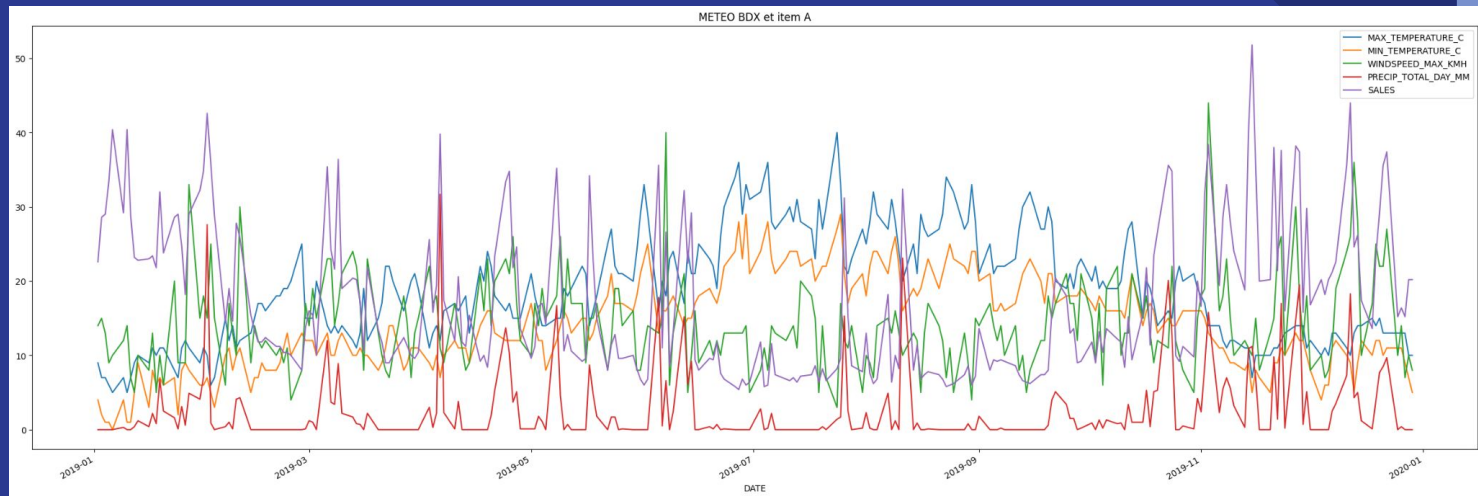
##Item B
merge_bdx_b = pd.DataFrame.join(meteo_bdx, df_b)
del merge_bdx_b['ITEM']
merge_bdx_b=merge_bdx_b.dropna(how = 'any')

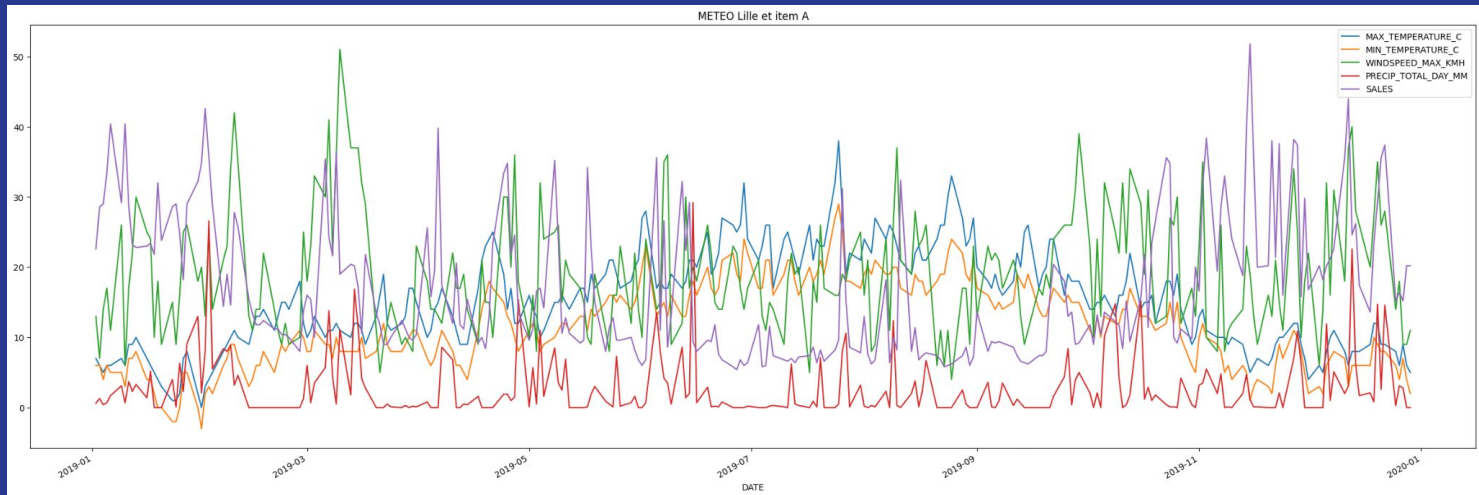
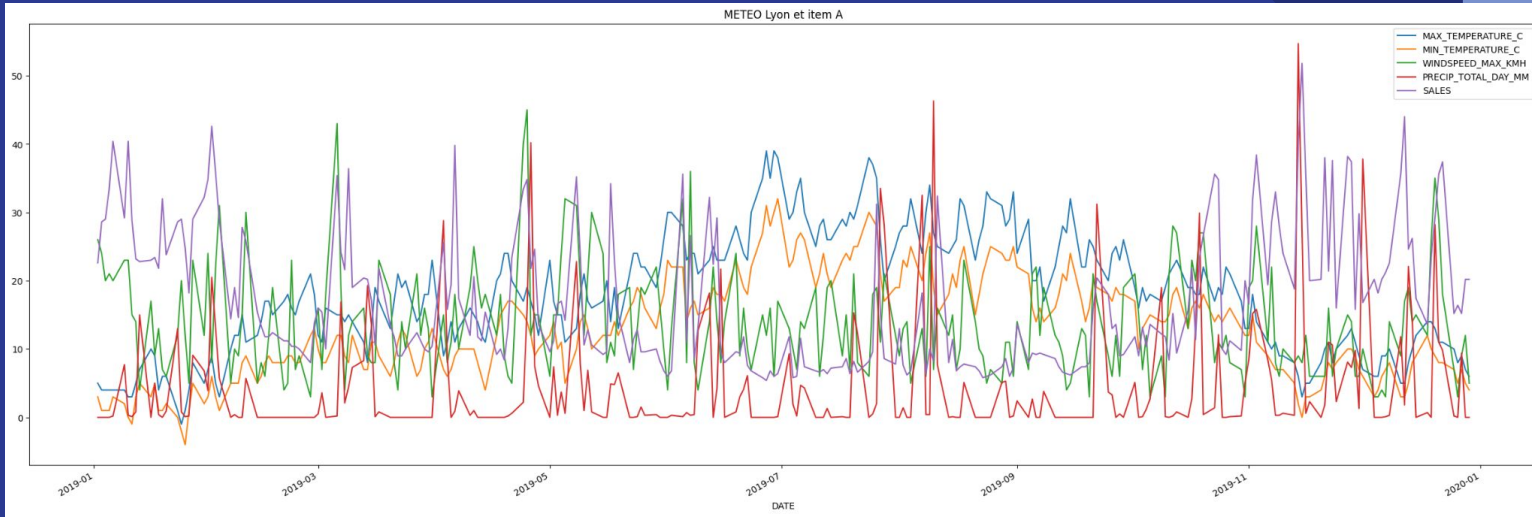
#Rapport
profile_a_bdx = merge_bdx_a.profile_report(title="Rapport Meteo bordeaux et Sales Item A",) #création d'un rapport
profile_a_bdx.to_file("rapports/bordeaux_a.html") #sauvegarde

profile_b_bdx = merge_bdx_b.profile_report(title="Rapport Meteo bordeaux et Sales Item B")
profile_b_bdx.to_file("rapports/bordeaux_b.html")
```

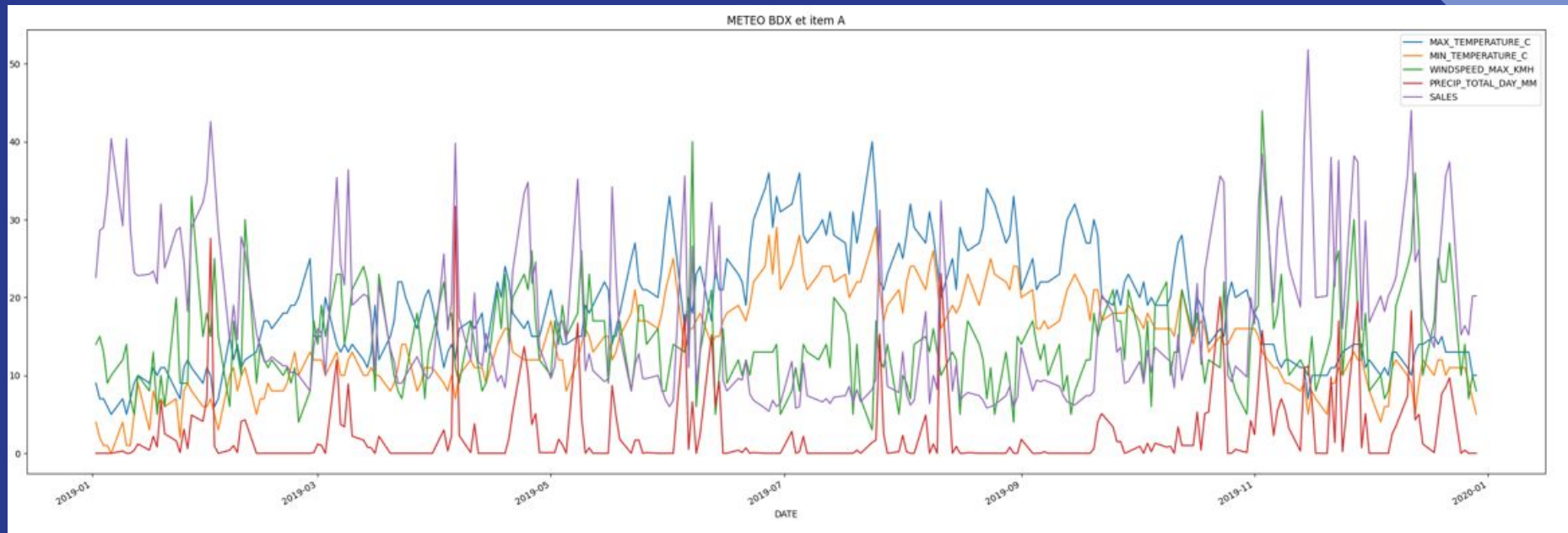
Génération des rapports

 bordeaux_a.html	24/08/2022 16:09	Brave HTML Docu...	1 879 Ko
 bordeaux_b.html	24/08/2022 16:09	Brave HTML Docu...	1 879 Ko
 lille_a.html	24/08/2022 16:10	Brave HTML Docu...	1 879 Ko
 lille_b.html	24/08/2022 16:10	Brave HTML Docu...	1 880 Ko
 lyon_a.html	24/08/2022 16:09	Brave HTML Docu...	1 878 Ko
 lyon_b.html	24/08/2022 16:09	Brave HTML Docu...	1 878 Ko
 marseille_a.html	24/08/2022 16:10	Brave HTML Docu...	1 895 Ko
 marseille_b.html	24/08/2022 16:10	Brave HTML Docu...	1 895 Ko
 meteo_bdx_a.png	24/08/2022 16:10	Fichier PNG	458 Ko
 meteo_bdx_b.png	24/08/2022 16:10	Fichier PNG	453 Ko
 meteo_lille_a.png	24/08/2022 16:10	Fichier PNG	468 Ko
 meteo_lille_b.png	24/08/2022 16:10	Fichier PNG	462 Ko
 meteo_lyon_a.png	24/08/2022 16:10	Fichier PNG	479 Ko
 meteo_lyon_b.png	24/08/2022 16:10	Fichier PNG	488 Ko
 meteo_marseille_a.png	24/08/2022 16:10	Fichier PNG	432 Ko
 meteo_marseille_b.png	24/08/2022 16:10	Fichier PNG	448 Ko





3- Explication de l'impact de la Météo



Corrélation température maximum et Item A pour Bordeaux

Ce graph nous permet de voir qu'il y a corrélation entre max température et ventes.

Plus les températures sont hautes, moins il y a de ventes.



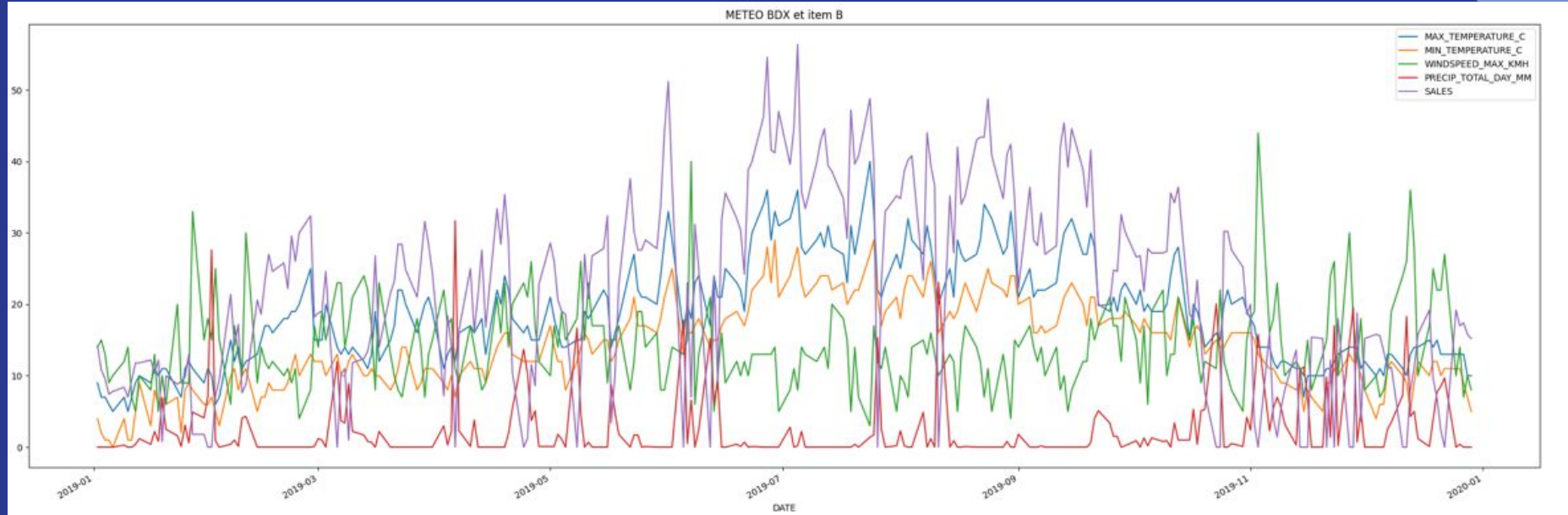
Corrélation précipitation Item A pour Bordeaux

Ce graph nous permet de voir qu'il y a corrélation entre les précipitations et les ventes.

les ventes augmentent avec les précipitations.



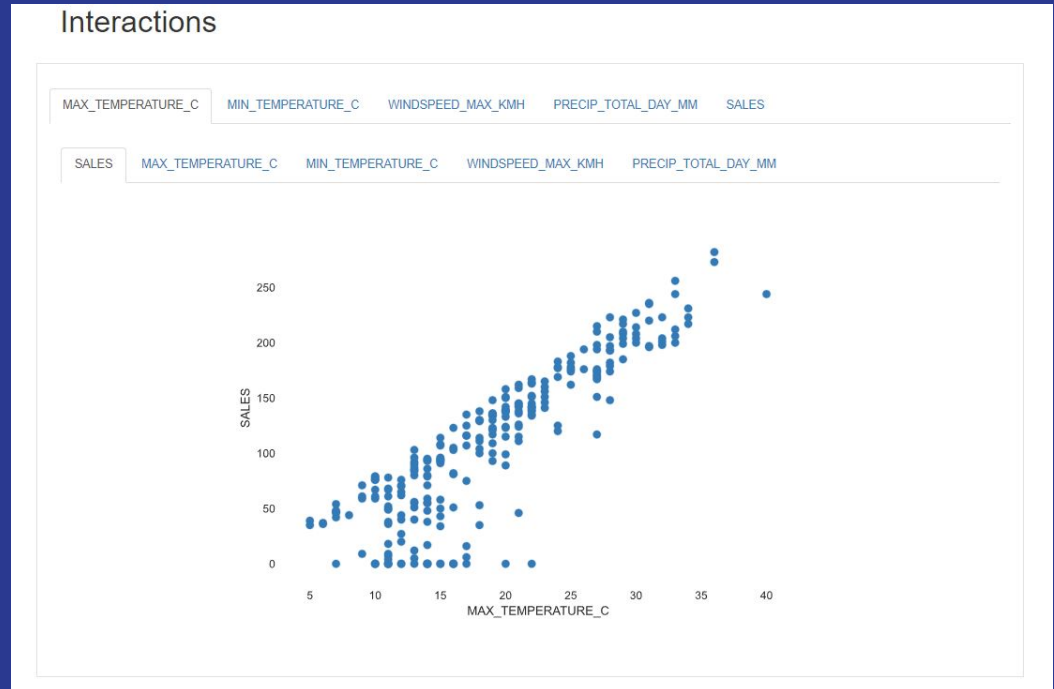
Graphique meteo Bordeaux + Ventes item B



Corrélation Max température Item B Bordeaux

Ce graph nous permet de voir qu'il y a corrélation entre max température et ventes.

Plus les températures sont hautes, plus il y a de ventes.



Corrélation précipitation

Item B Bordeaux

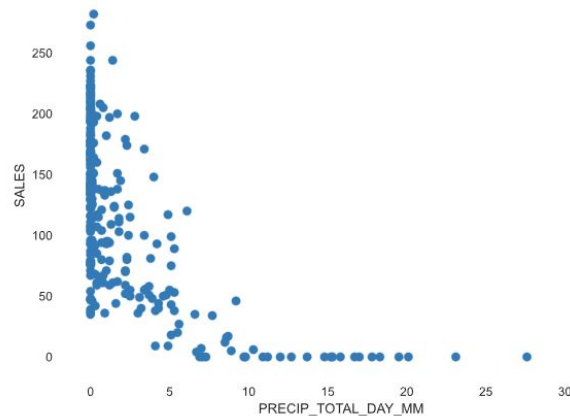
Ce graph nous permet de voir qu'il y a corrélation entre les précipitations et les ventes.

Les ventes baissent plus il y a de précipitations.

Interactions

MAX_TEMPERATURE_C MIN_TEMPERATURE_C WINDSPEED_MAX_KMH PRECIP_TOTAL_DAY_MM SALES

SALES MAX_TEMPERATURE_C MIN_TEMPERATURE_C WINDSPEED_MAX_KMH PRECIP_TOTAL_DAY_MM



4- Prévision des ventes

extraction de la dernière semaine de Juin 2019

Je constate que les ventes de A sont faibles et que les ventes de B sont élevées. Les températures sont élevées et sans pluie.

Dernière semaine de JUIN (Mercredi à Dimanche)

```
smooth_merge_bdx_a = pd.DataFrame(join(meteo_bdx, df_a) # join 2 df grace a la fonction pandas .join(self='', other='')
del smooth_merge_bdx_a['ITEM']
smooth_merge_bdx_a=smooth_merge_bdx_a.dropna(how = 'any')
df_a_week = smooth_merge_bdx_a.loc['2019-06-26':'2019-06-30']
```

```
smooth_merge_bdx_b = pd.DataFrame(join(meteo_bdx, df_b) # join 2 df grace a la fonction pandas .join(self='', other='')
del smooth_merge_bdx_b['ITEM']
smooth_merge_bdx_b.dropna(how = 'any')
df_b_week = smooth_merge_bdx_b.loc['2019-06-26':'2019-06-30']
```

```
temp_moy_2019 = (df_a_week['MIN_TEMPERATURE_C'] + df_a_week['MAX_TEMPERATURE_C'])/2
preci_moy_2019 = df_a_week['PRECIP_TOTAL_DAY_MM'].mean()
```

```
print(df_a_week.to_markdown(), '\n', df_b_week.to_markdown())
print('\n')
print('Moyenne des vente pour item A',df_a_week['SALES'].mean())
print('Moyenne des vente pour item B',df_b_week['SALES'].mean())
print('Pour une moyenne de',temp_moy_2019.mean(), 'degrés')
print('Précipitation moyenne de',preci_moy_2019,'mm')
```

DATE	MAX_TEMPERATURE_C	MIN_TEMPERATURE_C	WINDSPEED_MAX_KMH	PRECIP_TOTAL_DAY_MM	OPINION	SALES	DAY
2019-06-26 00:00:00	34	24	13	0	météo favorable	29	wednesday
2019-06-27 00:00:00	36	28	13	0	météo favorable	27	Thursday
2019-06-28 00:00:00	29	23	13	0	météo favorable	34	Friday
2019-06-29 00:00:00	33	29	14	0	météo favorable	30	Saturday
2019-06-30 00:00:00	31	21	5	0	météo correcte	32	Sunday

DATE	MAX_TEMPERATURE_C	MIN_TEMPERATURE_C	WINDSPEED_MAX_KMH	PRECIP_TOTAL_DAY_MM	OPINION	SALES	DAY
2019-06-26 00:00:00	34	24	13	0	météo favorable	231	wednesday
2019-06-27 00:00:00	36	28	13	0	météo favorable	273	Thursday
2019-06-28 00:00:00	29	23	13	0	météo favorable	208	Friday
2019-06-29 00:00:00	33	29	14	0	météo favorable	206	Saturday
2019-06-30 00:00:00	31	21	5	0	météo correcte	235	Sunday

Moyenne des vente pour item A 30.4
Moyenne des vente pour item B 230.6
Pour une moyenne de 28.8 degrés
Précipitation moyenne de 0.0 mm

Import Meteo last week 2021

```
[76] meteo_lastweek_2021 = pd.read_csv('https://raw.githubusercontent.com/murpi/wilddata/master/test/forecast.csv', usecols=['DATE', 'MAX_TEMPERATURE_C', 'MIN_TEMPERATURE_C', 'WINDSPEED_MAX_KMH', 'PRECIP_TOTAL_DAY_MM'])
meteo_lastweek_2021['TEMPS MOY'] = (meteo_lastweek_2021['MAX_TEMPERATURE_C'] + meteo_lastweek_2021['MIN_TEMPERATURE_C']) / 2

preci_moy_2021 = round(meteo_lastweek_2021['PRECIP_TOTAL_DAY_MM'].mean(),2)

print(meteo_lastweek_2021.to_markdown())
print('\n')
print('Température moyenne sur la semaine', meteo_lastweek_2021['TEMPS MOY'].mean())
print('Précipitation moyenne de',preci_moy_2021,'MM')
print('\n')
print('Différence de',round(meteo_lastweek_2021['TEMPS MOY'].mean()-temp_moy_2019.mean(),2),'degrés entre 2021 et 2019')
```

	DATE	MAX_TEMPERATURE_C	MIN_TEMPERATURE_C	WINDSPEED_MAX_KMH	PRECIP_TOTAL_DAY_MM	TEMPS MOY
0	2021-06-21	20	14	23	2	17
1	2021-06-22	25	17	10	0	21
2	2021-06-23	25	18	19	0	21.5
3	2021-06-24	32	23	12	0	27.5
4	2021-06-25	28	20	13	1	24
5	2021-06-26	20	16	19	0	18
6	2021-06-27	21	15	31	1	18

Température moyenne sur la semaine 21.0
Précipitation moyenne de 0.57 MM

Différence de -7.8 degrés entre 2021 et 2019

Je constate que la météo est moins bonne qu'en 2019, température plus basse, des précipitations et du vent. Nous devrions vendre plus de A et moins de B.