TP9

STATISTIQUES AVEC PANDAS

I Ouvrir un tableur

[In]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[In]:
```
France = pd.read_excel('Deces-France.xls')
```

Affichons un aperçu du tableau

[In]:
```
France
```

[Out]:

	Country	Year	Week	Sex	0-14	15-64	65-74	75-84	85+	Total
0	FRATNP	2000	1	m	42.000000	1612.000000	1540.000000	1796.000000	1757.000000	6747.0
1	FRATNP	2000	1	f	34.000000	678.000000	757.000000	1641.000000	3773.000000	6883.0
2	FRATNP	2000	1	b	76.000000	2290.000000	2297.000000	3437.000000	5530.000000	13630.0
3	FRATNP	2000	2	m	56.000000	1529.000000	1431.000000	1843.000000	1672.000000	6531.0
4	FRATNP	2000	2	f	53.007978	657.098901	781.117567	1670.251392	3482.524161	6644.0
...										
3400	FRATNP	2021	38	f	24.123664	532.730916	643.297710	1019.224809	3047.622901	5267.0
3401	FRATNP	2021	38	b	60.366497	1615.988926	1760.785062	2355.175905	4847.683609	10640.0
3402	FRATNP	2021	39	m	35.238863	1039.043033	1079.316019	1356.192796	1801.209289	5311.0
3403	FRATNP	2021	39	f	30.132565	521.293372	605.664553	1043.591162	3027.318348	5228.0
3404	FRATNP	2021	39	b	65.371427	1560.336405	1684.980572	2399.783959	4828.527637	10539.0

[3405 rows x 10 columns]

On visualise les premières données et les dernières. Il y a 3405 lignes au total.

Le début nous donne les données concernant les décès en 2000, semaine 1, pour les hommes (m) puis les femmes (f) puis en troisième ligne pour les deux (b - both).

On passe ensuite à la semaine 2, etc.

Les données sont rangées par tranche d'âge et pas sexe donc.

Affichons le type des données de chaque colonne (entier (int), flottant (float) ou autre (object))

[In]:
```
France.info()
```

[Out]:
```
RangeIndex: 3405 entries, 0 to 3404
Data columns (total 10 columns):
Country    3405 non-null object
Year       3405 non-null int64
Week       3405 non-null int64
Sex        3405 non-null object
0-14       3405 non-null float64
15-64      3405 non-null float64
65-74      3405 non-null float64
75-84      3405 non-null float64
85+        3405 non-null float64
Total      3405 non-null float64
dtypes: float64(6), int64(2), object(2)
memory usage: 266.1+ KB
```

On lit ici les noms des colonnes, avec le nombre de données (non vides) et le type de données. Par exemple, la colonne 'Year' a 3405 données et type entier (int64).

II Statistiques de base

Nous pouvons maintenant calculer automatiquement les statistiques élémentaires.

- Nombre de données
- Moyenne (mean), Écart-type (std)
- minimum, maximum
- quartiles (25% c'est-à-dire Q_1, 50% c'est-à-dire la médiane, et 75% c'est-à-dire Q_3)

[In]:
```
France.describe()
```

[Out]:

	Year	Week	0-14	15-64	65-74
count	3405.000000	3405.000000	3405.000000	3405.000000	3405.000000
mean	2010.385022	26.370044	52.678138	1285.872088	1056.378048
std	6.280232	15.001486	21.783598	546.002165	449.234369
min	2000.000000	1.000000	15.054108	505.103410	374.082488
25%	2005.000000	13.000000	36.000000	643.000000	609.000000
50%	2010.000000	26.000000	45.008893	1333.000000	1014.000000
75%	2016.000000	39.000000	70.000000	1839.000000	1418.710404
max	2021.000000	53.000000	129.000000	2716.084914	2670.183055

	75-84	85+	Total
count	3405.000000	3405.000000	3405.000000
mean	1903.404354	2826.046080	7124.378708
std	727.923273	1359.097477	2686.827141
min	892.254072	777.000000	3986.000000
25%	1350.000000	1737.000000	5059.000000
50%	1561.255316	2668.000000	5623.000000
75%	2601.253927	3621.000000	9785.000000
max	5341.293816	9420.383836	18805.000000

Remarquons que les données fournissent les décès par semaine des hommes (m), femmes (f) et des deux réunis (b). Cela n'a pas de sens de calculer des statistiques sur toutes ces données en même temps.

Nous allons séparer les données en fonction du sexe à l'aide d'un filtre.

[In]:
```
# Données mixtes
France_b = France[ France['Sex'] == 'b']
France_b
```

[Out]:

	Country	Year	Week	Sex	0-14	15-64	65-74	75-84	85+	Total
2	FRATNP	2000	1	b	76.000000	2290.000000	2297.000000	3437.000000	5530.000000	13630.0
5	FRATNP	2000	2	b	109.007978	2186.098901	2212.117567	3513.251392	5154.524161	13175.0
8	FRATNP	2000	3	b	99.000000	2182.000000	2284.000000	3317.000000	5310.000000	13192.0
11	FRATNP	2000	4	b	113.000000	2128.000000	2300.000000	3359.000000	5293.000000	13193.0
14	FRATNP	2000	5	b	103.000000	2209.000000	2199.000000	3246.000000	5128.000000	12885.0
...										
3392	FRATNP	2021	35	b	60.435390	1656.922656	1880.381648	2600.324451	5182.935855	11381.0
3395	FRATNP	2021	36	b	61.460832	1745.361779	1890.364772	2609.482566	5294.330050	11601.0
3398	FRATNP	2021	37	b	76.551773	1522.407936	1782.238477	2445.863484	4933.938329	10761.0
3401	FRATNP	2021	38	b	60.366497	1615.988926	1760.785062	2355.175905	4847.683609	10640.0
3404	FRATNP	2021	39	b	65.371427	1560.336405	1684.980572	2399.783959	4828.527637	10539.0

[1135 rows x 10 columns]

Ce tableau contient uniquement les lignes où la colonne 'Sex' vaut 'b', c'est-à-dire les données mixtes.
On pense à mettre des guillemets aux noms de colonnes et aux données textuelles.

Calculer alors les statistiques de base pour les données mixtes

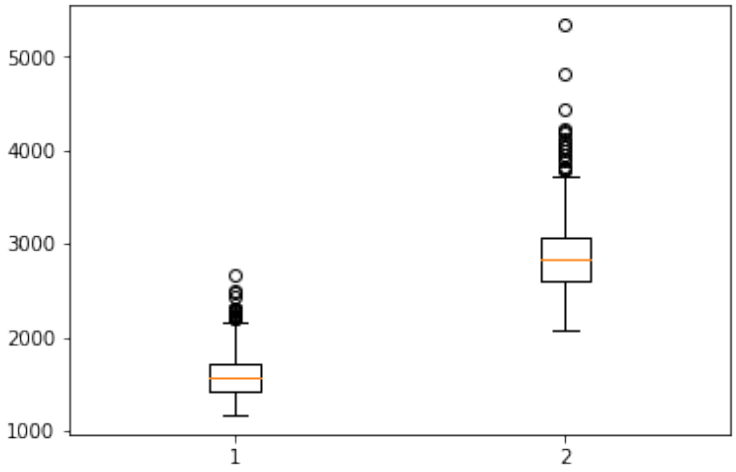
[In]:
```
France_b.describe()
```

[Out]:

	Year	Week	0-14	15-64	65-74	75-84	85+	Total
count	1135.000000	1135.000000	1135.000000	1135.000000	1135.000000	1135.000000	1135.000000	1135.000000
mean	2010.385022	26.370044	79.017206	1928.808133	1584.567072	2855.106531	4239.069119	10686.568062
std	6.282078	15.005895	13.801490	137.131103	218.901900	364.810238	1063.415493	1312.187947
min	2000.000000	1.000000	41.283813	1522.407936	1168.247629	2084.212949	2474.000000	8458.000000
25%	2005.000000	13.000000	69.006108	1839.000000	1410.500000	2600.833210	3422.000000	9785.500000
50%	2010.000000	26.000000	78.000000	1945.403261	1564.000000	2826.000000	4053.000000	10370.000000
75%	2016.000000	39.000000	88.000000	2019.631549	1725.111848	3059.156982	4826.165430	11228.500000
max	2021.000000	53.000000	129.000000	2716.084914	2670.183055	5341.293816	9420.383836	18805.000000

Comparons par exemple les tranches 65-74 et 75-84 en traçant les boîtes à moustaches correspondantes.

[In]:
```
plt.boxplot([France_b['65-74'], France_b['75-84']])
```



On remarque que les données de la colonne '75-84' sont plus élevées (plus de décès dans cette tranche d'âge que dans celle des '65-74', cela semble normal) et également un peu plus dispersées puisque la bôite à moustaches est plus étirée.
Remarque : les moustaches ne vont pas jusqu'au max, c'est une autre convention, les valeurs extrêmes étant représentées par des points.

III Données mixtes, par année

Commençons par regarder de plus près les données de l'année 2020, avec un filtre.

[In]:
```
annee2020 = France_b[France_b['Year']==2020]
annee2020
```

[Out]:

	Country	Year	Week	Sex	0-14	15-64	65-74	75-84	85+	Total
3131	FRATNP	2020	1	b	58.000000	1941.000000	1931.000000	2739.000000	6068.000000	12737.0
3134	FRATNP	2020	2	b	68.008402	1846.186347	2050.218455	2852.395499	6383.191298	13200.0
3137	FRATNP	2020	3	b	59.000000	1798.000000	1901.000000	2738.000000	6102.000000	12598.0
...										
3281	FRATNP	2020	51	b	73.000000	1754.000000	2094.000000	3191.000000	6994.000000	14106.0
3284	FRATNP	2020	52	b	58.004123	1772.170225	2075.202179	3089.252540	6806.370932	13801.0
3287	FRATNP	2020	53	b	70.000000	1775.000000	2073.000000	3213.000000	7234.000000	14365.0

Calculer le nombre total de décès en 2020 avec sum

[In]:
```
annee2020['Total'].sum()
```

[Out]:
```
664335.0
```

Afficher tous les totaux de décès par tranche d'âge

[In]:
```
annee2020.sum()
```

[Out]:
```
Country    FRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRAT...
Year                                                  107060
Week                                                    1431
Sex        bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb...
0-14                                                3372.14
15-64                                               94178.2
65-74                                                103624
75-84                                                148117
85+                                                  315044
Total                                                664335
dtype: object
```

Cette commande réalise la somme sur chacune des colonnes.

Pour les données textuelles, cela revient à faire une concaténation. Ici, seules les 6 dernières colonnes sont pertinentes.

On va maintenant tracer un diagramme en barres des décès par années.

[In]:
```
annees = France['Year'].unique()
annees
```

[Out]:
```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021])
```

Remarque : C'est la liste des années étudiées. On aurait aussi pu écrire :
`annees = range(2000, 2022)`.

[In]:
```
deces_par_annee = [France_b[France_b['Year']==k]['Total'].sum() for k in annees]
deces_par_annee
```

[Out]:
```
[525655.0, 528888.0, 531135.0, 550707.0, 519387.0, 528207.0, 516181.0, 520601.0,
530138.0, 550272.75, 540864.0, 533318.0, 557207.0, 558241.0, 546129.0, 592498.0,
578416.0, 591715.0, 595029.0, 597917.0, 664335.0, 472414.0]
```

Cette commande permet de créer la liste des nombres de décès par année. On reprend donc ce qu'on a fait pour l'année 2020 mais le fait pour toutes les années.

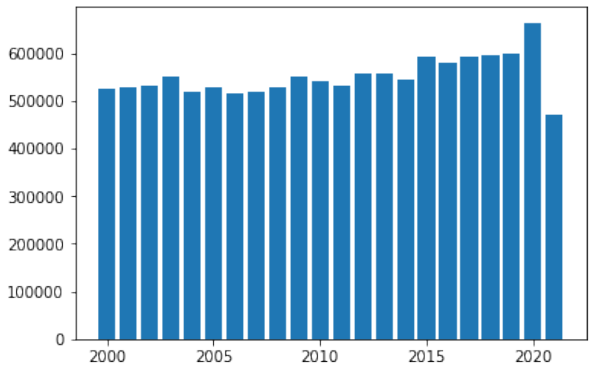
`France_b[France_b['Year']==k]` : tableau donnant l'année k

`France_b[France_b['Year']==k]['Total'].sum()` : décès pour l'année k, en une ligne.

Puis on fait varier k dans la liste des années.

On réalise enfin le diagramme en barres avec **plt.bar(liste des abscisses , liste des ordonnées)**.

[In]:
```
plt.bar(annees, deces_par_annee)
```



On peut remarquer une augmentation significative en 2020.

L'année 2021 n'est pas achevée, d'où un nombre de décès moindre.

IV Comparaison hommes-femmes, année 2020

Commençons par créer les bases de données hommes et femmes.

[In]:
```
France_f = France[ France['Sex'] == 'f'] France_m = France[ France['Sex'] == 'm']
```

Puis filtrons pour n'avoir que les données de 2020.

[In]:
```
f2020 = France_f[France_f['Year'] == 2020]
m2020 = France_m[France_m['Year'] == 2020]
```

Afficher le nombre de décès de l'année 2020 par tranche d'âge pour les femmes.

[In]:
```
f2020.sum()
```

[Out]:
```
Country    FRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRAT...
Year                                                  107060
Week                                                    1431
Sex        fffffffffffffffffffffffffffffffffffffffffffff...
0-14                                                1485.05
15-64                                                 31923
65-74                                               36518.1
75-84                                                 64498
85+                                                  197771
Total                                                332195
dtype: object
```

Afficher les autres statistiques pour les femmes

[In]:
```
f2020.describe()
```

[Out]:
```
          Year        Week        0-14        15-64        65-74         75-84
count     53.0   53.000000   53.000000    53.000000    53.000000     53.000000
mean    2020.0   27.000000   28.019763   602.320677   689.020948   1216.942521
std        0.0   15.443445    4.790009    45.908004    74.625533    205.950030
min     2020.0    1.000000   17.000000   529.000000   558.000000    946.000000
25%     2020.0   14.000000   25.000000   561.000000   636.000000   1060.000000
50%     2020.0   27.000000   28.000000   604.000000   680.000000   1175.188785
75%     2020.0   40.000000   31.000000   628.000000   740.000000   1360.000000
max     2020.0   53.000000   40.000000   712.000000   935.000000   1880.000000

                85+         Total
count     53.000000     53.000000
mean    3731.526280   6267.830189
std      731.688044   1036.695689
min     2802.000000   4933.000000
25%     3074.000000   5392.000000
50%     3573.000000   6013.000000
75%     4287.000000   7009.000000
max     5782.000000   9326.000000
```

Faire de même pour les hommes.

[In]:
```
m2020.sum()
m2020.describe()
```

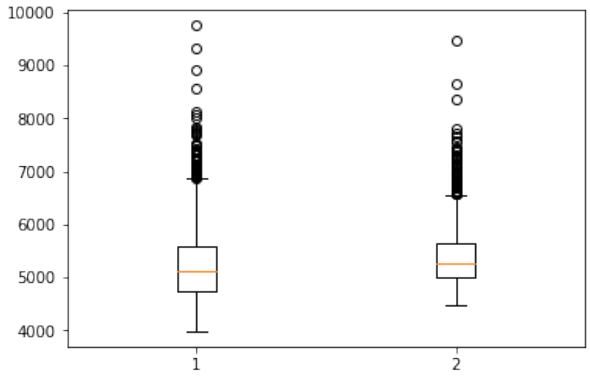
[Out]:
```
Country    FRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRATNPFRAT...
Year                                                  107060
Week                                                    1431
Sex        mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm...
0-14                                                  1887.1
15-64                                                62255.2
65-74                                                67105.6
75-84                                                83618.6
85+                                                   117273
Total                                                 332140

          Year        Week        0-14         15-64         65-74         75-84
count     53.0   53.000000   53.000000     53.000000     53.000000     53.000000
mean    2020.0   27.000000   35.605599   1174.627187   1266.143184   1577.709720
std        0.0   15.443445    7.156200     78.035727    157.122275    307.663287
min     2020.0    1.000000   22.000000   1027.000000   1061.000000   1224.000000
25%     2020.0   14.000000   31.000000   1142.000000   1154.217081   1352.000000
50%     2020.0   27.000000   36.000000   1167.000000   1221.000000   1509.000000
75%     2020.0   40.000000   40.000000   1206.212399   1354.000000   1806.000000
max     2020.0   53.000000   52.008505   1506.158894   1735.183055   2564.270521

                85+         Total
count     53.000000     53.000000
mean    2212.706763   6266.792453
std      490.182239   1002.253775
min     1572.000000   5087.000000
25%     1829.000000   5474.000000
50%     2123.000000   6059.000000
75%     2553.362951   6930.000000
max     3638.383836   9479.000000
```

Comparaison des totaux (par semaine) suivant le sexe.

[In]:
```
plt.boxplot([ France_f['Total'], France_m['Total']])
```



Ici, on peut remarquer que les données des femmes (nombre de décès par semaine) sont plus dispersées que celles des hommes.

Le nombre de décès médian (barre rouge) pour les hommes est légèrement supérieur à celui des femmes.

Comparons les répartitions par tranche d'âge.

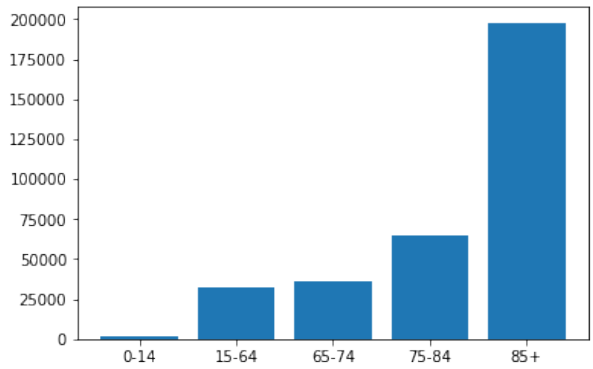
[In]:
```
ages = France.columns[4:-1]
ages
```

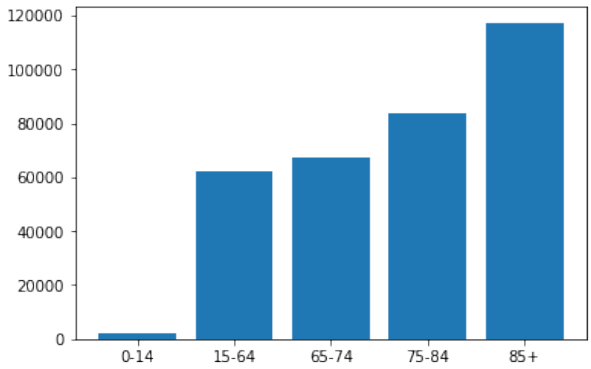
[Out]:
```
['0-14', '15-64', '65-74', '75-84', '85+']
```

[In]:
```
# Total par age
deces_f = f2020[ages].sum()
deces_m = m2020[ages].sum()

# Diagramme en barres pour les femmes
plt.bar(ages, deces_f)

# Et pour les hommes
plt.bar(ages, deces_m)
```

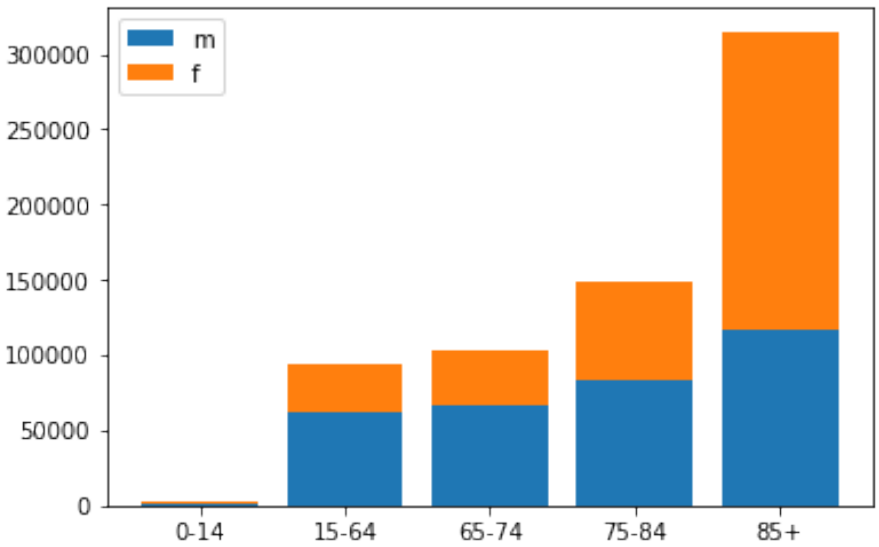




On voit une différence de répartition, mais les échelles ne sont pas les mêmes. Regroupons cela en un seul graphique.

[In]:
```
# Diagrammes empilés
plt.bar(ages,deces_m, label='m')
plt.bar(ages, deces_f, bottom = deces_m, label='f')

plt.legend()
plt.show()
```



Les femmes ont donc tendance à mourir plus agées que les hommes.