Résumé des méthodes de l'analyse univariée :

Table des matières

1. Introduction

Lors du module de Data Science Révision nous somme amener à lier les contenues théoriques avec les exercices pratiques à réaliser. Nous tenterons dans ce document d'appuyer les connaissances utilisées dans les révisions en voyant dans un cas concret comment elles sont utilisées.

Mais rappelons tout d'abord que la méthodologie d'un data analyste consiste à :

- Explorer les données
- Résumer les données
- Prendre une décision
- Communiquer les résultats.

Nous verrons donc tout d'abord la réalisation de l'import des données puis la manière dont un data analyste peu Résumer de manière pertinente leurs contenus

Dans le cadre de ce TP, les packages « Pandas » et « matplotlib » ont été utilisés. Le jeu de donnée FrenchCities.csv, composé des données météo de 32 villes, composera la base de notre étude.

2. Import des données

Afin d'apporter les données à étudiant dans python, on utilise le package Pandas qui simplifie la création et la manipulation de dataframe.

Suite à l'import des packages,

```
import pandas as pd
import matplotlib.pyplot as plt
```

Il suffit d'utiliser read_csv, une fonction de pandas, pour lire un csv et le récupérer sous forme de dataframe exemple :

```
df = pd.read_csv("Data\FrenchCities.csv", sep=";", header=0, index_col=0)
```

Il en sort notre dataframe sur Python.

	CLIMAT	NO2	DENSITY	JANr	FEBr	MARr	APRr	MAYr	JUNr	JULr	...	SEPdr	OCTdr	NOVdr	DECdr	DAYS_RAINFALL	TEMP	TEMP_RAN
Ajaccio	2	18.5	808	78	69	51	39	43	23	10	...	6	10	11	13	95	14.71	1
Angers	3	16.5	3490	65	50	60	45	50	55	35	...	12	13	15	16	154	11.28	1
Angoulme	4	17.0	1923	79	68	64	62	70	58	53	...	12	13	15	16	160	12.02	1
Besancon	1	27.3	1789	94	87	75	74	86	107	80	...	13	14	15	15	169	10.04	1
Biarritz	3	16.0	2172	128	105	98	102	100	91	69	...	14	15	16	17	177	13.58	1

5 rows × 34 columns

3. Exploration des données.

Il est commun de réaliser une exploration des données afin de vérifier la qualité et la pertinence de celles-ci. Pour ce faire on peut afficher l'index et les colonnes pour se familiariser avec les variables. On peut aussi, grâce à a fonction .info() afficher les types des variables composants le tableau.

```
# Affichages des info des variables du dataframe
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 32 entries, Ajaccio to Error
Data columns (total 34 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   CLIMAT        32 non-null     int64
 1   NO2           32 non-null     float64
 2   DENSITY       32 non-null     int64
 3   JANr          32 non-null     int64
 4   FEBr          32 non-null     int64
 5   MARr          32 non-null     int64
 6   APRr          32 non-null     int64
 7   MAYr          32 non-null     int64
 8   JUNr          32 non-null     int64
 9   JULr          32 non-null     int64
 10  AUGr          32 non-null     int64
 11  SEPr          32 non-null     int64
```

Nom de la variable

Type de la variable

Nombre de datas non null pour chaque variable

Cela permet de se rendre compte de la qualité des données (s'il y a beaucoup de valeurs manquantes) et des types de données (à première vue) données (quantitative, qualitative). L'étude univarié des données est différente suivant leurs types.

4. Variables quantitatives

Il existe plusieurs moyens plus ou moins pertinent suivant les cas, d'étudier les variables quantitatives d'un data frame.

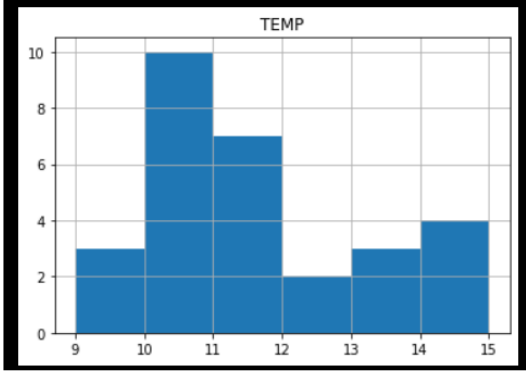
4.1. Graphiques

On peut ainsi réaliser des graphiques pour rendre l'analyse de données plus visuel. Pour les variables discrètes, les diagrammes bâton sont pertinent lorsque l'histogramme se prête plus à l'analyse de variables quantitatives continues.

```
# Histograme sur TEMP, fonctionne sur variable qualitative, nécéssite classes pour quantitative. (Ici grâce à bins)
df.hist(column='TEMP', bins=[9,10,11,12,13,14,15])
```

✓ 0.2s

```
array([[<AxesSubplot:title={'center':'TEMP'}>]], dtype=object)
```



4.2. Indicateur de position

Il est également nécessaire, lors de l'exploration des données de vérifier les indicateurs de position (**Moyenne**, **Mode**, **Médiane** et **Quartiles**)

La fonction .describe() permet de récupérer facilement ces informations :

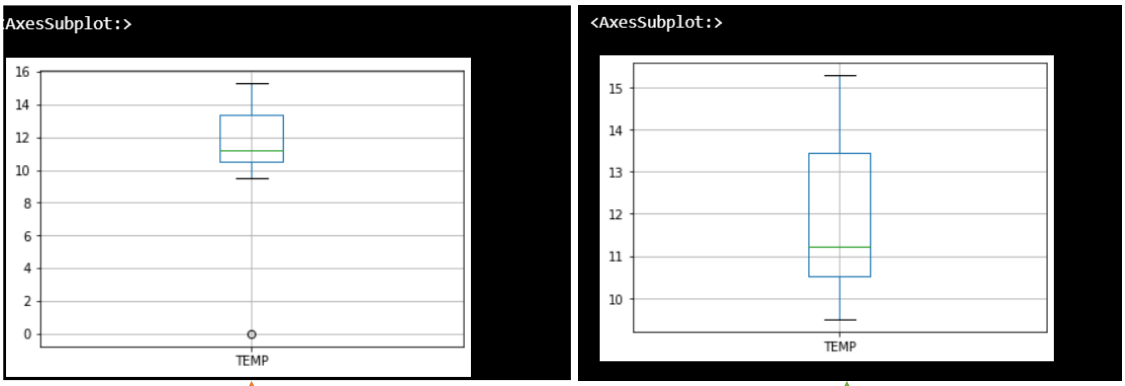
```
# Étude du df avec compte des données, moyenne, écrat type, min, 1Q, mediane, 3Q et max
summary = df.describe()
summary = summary.transpose()
summary.head()
```

✓ 0.1s

	count	mean	std	min	25%	50%	75%	max
CLIMAT	32.0	2.375000	1.099853	1.0	1.00	2.00	3.00	4.0
NO2	32.0	29.678125	10.918192	0.0	25.60	28.75	35.10	58.4
DENSITY	32.0	4364.593750	3769.979848	0.0	2149.25	3840.50	4734.50	21258.0
JANr	32.0	64.500000	26.370805	0.0	52.75	64.00	78.25	130.0
FEBr	32.0	59.500000	20.730218	0.0	48.75	56.50	70.50	105.0

Compte des données — Moyenne — Ecart type — Minimum — 1er quartile — Médiane — 3ieme quartile — Maximum

Le graph boxplot ou boite à moustaches est une manière très visuelle de réaliser cette étude.



Boxplot avec valeur aberrante

Boxplot sans valeur aberrante

Cette étude de position nous a permis éliminer une variable aberrante, rendant notre jeu de données plus propre.

4.3. Indicateur de dispersion

Une fois la compréhension de ce que représente chaque variable ainsi que ce que sont leurs valeurs « normales », il est possible de trouver les indicateurs de dispersions. (**Étendu**, **écart-type**, **variance**, **écart-median**)

4.4. Centrer et réduire

Enfin afin de rendre les variables plus pertinentes à étudier et à comprendre, il est fortement conseiller de centrer et réduire les variables. (Seulement les variables quantitatives !!)

```
temp = df.drop(['CLIMAT'], axis = 1) # on retire la variable qualitative
temp = temp.sub(temp.mean()) # Soustraire la moyenne de chaque colonne à chaque valeur : centrer les valeurs

temp = temp.div(temp.std()) # Divisé les valeurs de la dataframe par l'écart-type de chaque colonne
```

	NO2	DENSITY	JANr	FEBr	MARr	APRr	MAYr	JUNr	JULr	AUGr	...	SEPdr	OCTdr	NOVdr
Ajaccio	-1.259246	-0.987075	0.476033	0.422288	-0.367765	-0.964404	-1.105487	-1.848344	-1.933731	-1.959687	...	-1.791058	-0.972818	-0.856260
Angers	-1.466778	-0.271073	-0.065892	-0.636127	0.205455	-0.550134	-0.630456	-0.245584	-0.552495	-0.149248	...	0.330832	0.233476	0.454558
Angoulme	-1.414895	-0.689408	0.517720	0.366582	0.460220	0.623633	0.726775	-0.095325	0.441996	0.092144	...	0.330832	0.233476	0.454558
Besancon	-0.346109	-0.725182	1.143017	1.424997	1.160823	1.452174	1.812560	2.358901	1.933731	2.103744	...	0.684481	0.635574	0.454558
Biarritz	-1.518661	-0.622934	2.560359	2.427706	2.625720	3.385437	2.762622	1.557521	1.325987	2.385368	...	1.038129	1.037673	0.782263

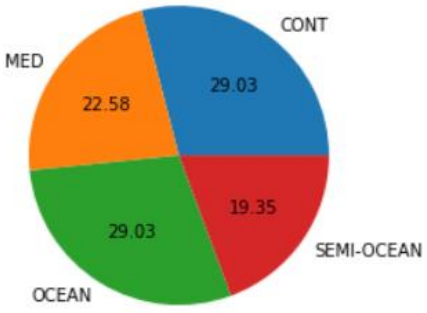
5. Variables qualitatives

Le traitement des variables qualitative est différents. La plupart du temps, dans ces cas, les informations numériques parlent moins que les représentations graphiques. C'est pour cela que pour étudier la variable 'CLIMAT' une étude graphique est choisie. Un diagramme de secteur nous permet de se rendre compte des proportions de villes parmi quatre types de climats différents :

```
# Diagrame de secteur sur les classes de climats créées (autopct= )
df.groupby('CLIMAT').size().plot(kind='pie', autopct='%.2f')
```

✓ 0.1s

```
<AxesSubplot:ylabel='None'>
```



s

6. Conclusion

En somme le data scientiste doit pour chaque jeu de données, mener une étude rigoureuse avant de pouvoir décider de la pertinence d'une étude dessus. Il doit aussi prévoir lesquelles sont appropriées ou non en fonction de chaque variables (quantitative ou qualitative).