Python pour la Data Science

Meta Brains

Section 1 : Remise à niveau rapide sur Python

Section 2: Data Science avec Python

Section 3: Structure des données Panda

Section 4: Nettoyage des données

Meta Brains

Section 5 : Visualisation des données sur Python

Section 6 : Analyse exploratoire des données

Meta Brains

ANALYSE EXPLORATOIRE DES DONNEES

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 **Qu'est-ce que l'analyse exploratoire des données**

2 Mise en place Python

3 Qu'est-ce que Jupyter

4 Installation d'Anaconda pour Windows OS

5 Installation d'Anaconda pour Mac OS

6 Installation d'Anaconda pour Ubuntu OS

7 Comment executer Python sur Jupyter

8 Gérer les répertoires dans Jupyter

9 Mise à l'echelle des caractéristiques

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Analyse des données exploratoires

- L'analyse exploratoire des données, ou EDA (de l'anglais Exploratory Data Analysis), est un moyen d'analyser des ensembles de données, souvent à l'aide de méthodes de visualisation des données, pour résumer les principales caractéristiques des données.
- L'EDA aide à mieux comprendre les différentes caractéristiques des données, la relation entre elles et à déterminer les techniques statistiques appropriées pour l'ensemble de données.
- Les bibliothèques matplotlib et seaborn sont assez bonnes pour l'EDA.

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 **Analyse unidimensionnelle**

3 Qu'est-ce que Jupyter

4 Installation d'Anaconda pour Windows OS

5 Installation d'Anaconda pour Mac OS

6 Installation d'Anaconda pour Ubuntu OS

7 Comment executer Python sur Jupyter

8 Gérer les répertoires dans Jupyter

9 Mise à l'echelle des caractéristiques

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Analyse univariée (1/7)

- Un ensemble de données peut contenir une ou plusieurs caractéristiques/variables/colonnes.
- L'analyse univariée fournit des statistiques récapitulatives sur une seule variable.
- L'analyse univariée décrit uniquement les données et aide à identifier les tendances dans les données.

Données catégoriques et données continues

- Il existe deux types de données :
  - Catégorique/discret, par exemple, spam ou pas de spam, homme ou femme
  - Continues, par exemple l'âge de la population
- L'EDA est effectuée différemment sur les deux types de données.

Meta Brains

Analyse univariée (2/7)

- Considérez la base de données suivante contenant 5 caractéristiques à propos des caractéristiques des plantes d'iris.
- La dernière colonne est catégorique, tandis que le reste est continu.

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

Meta Brains

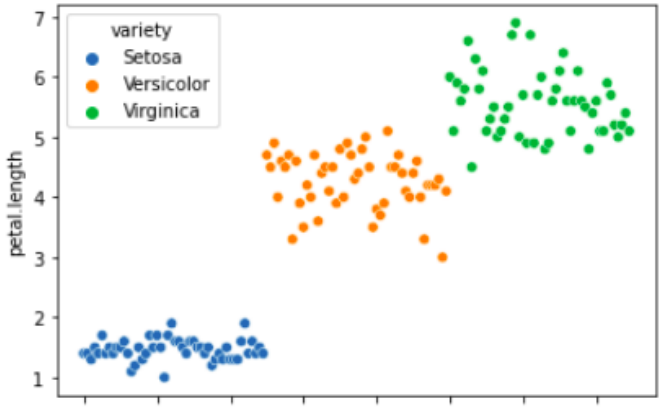
Analyse unidimensionnelle – données continues (3/7)

Nuage de points

- Nous pouvons utiliser un certain nombre de graphiques pour effectuer l'EDA sur des données continues.
- Dans la figure donnée, nous utilisons la fonction . scatterplot() de la bibliothèque seaborn pour tracer la colonne « petal.length » de la base de données.
- Le paramètre « hue » assigne une couleur différente aux points de données en fonction de la catégorie à laquelle ils appartiennent dans la colonne spécifiée dans le paramètre « hue ».

```
x_axis = df.index
y_axis = df['petal.length']
sns.scatterplot(x=x_axis, y=y_axis, hue=df['variety'])
```

```
<AxesSubplot:ylabel='petal.length'>
```



Meta Brains

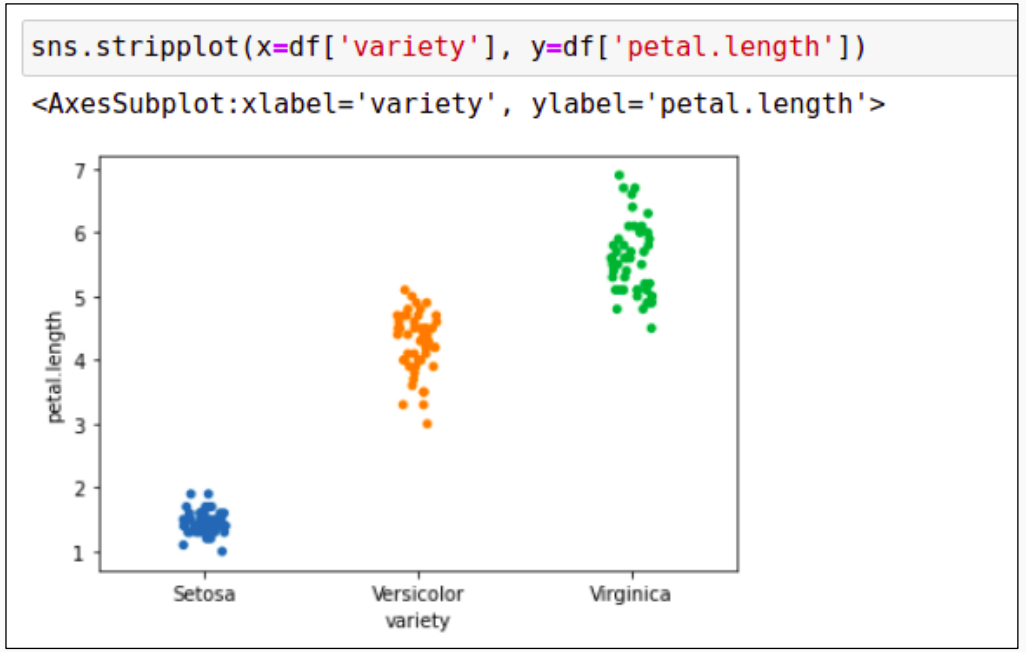
Analyse univariée – données continues(4/7)

Diagramme à bande

- Les diagrammes à bandes sont également un bon moyen d'analyser la distribution des variables pour chaque catégorie.
- Dans la figure donnée, nous avons tracé une colonne « longueur des pétales » pour chaque catégorie dans la colonne « variété ».
- Sur l'axe des Y, nous avons la distribution de chaque catégorie, et sur l'axe des X, nous avons les catégories.

```
sns.stripplot(x=df['variety'], y=df['petal.length'])
```

```
<AxesSubplot:xlabel='variety', ylabel='petal.length'>
```



Meta Brains

Analyse univariée – données continues(5/7)

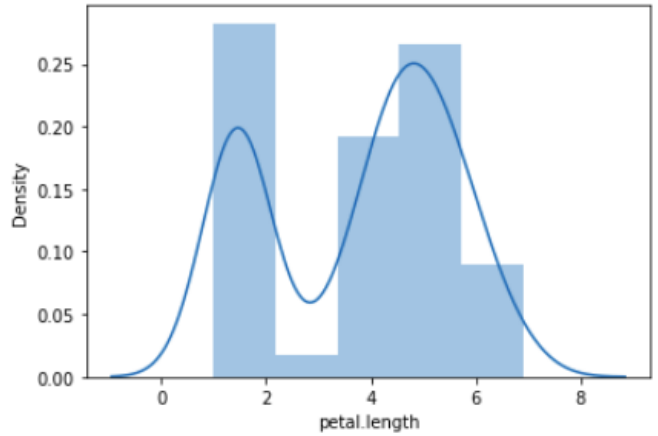
Distribution

- Pour trouver la distribution d'une variable/fonctionnalité/colonne, utilisez la fonction . distplot() de la bibliothèque seaborn.
- Dans cette figure, nous traçons la distribution de la colonne « longueur des pétales ».

```
sns.distplot(df['petal.length'])
```

```
/home/waqar/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWar
ning: `distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='petal.length', ylabel='Density'>
```



Meta Brains

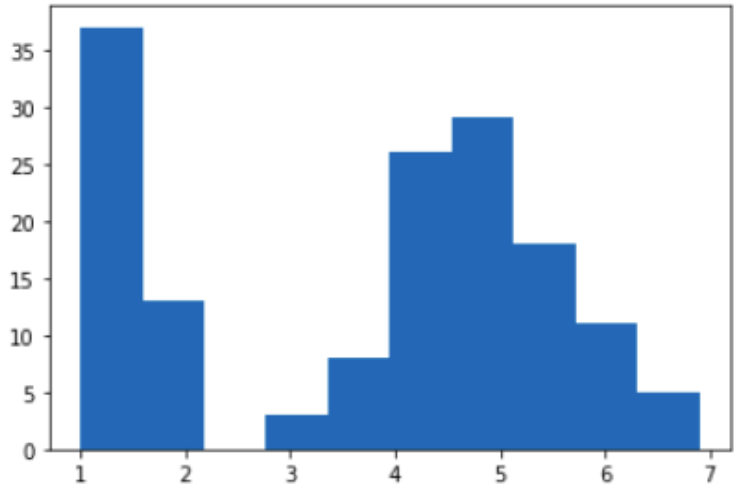
Analyse univariée – données continues(6/7)

Histogrammes

- Pour analyser la fréquence des valeurs, nous pouvons tracer un histogramme en utilisant la méthode . hist() de la bibliothèque matplotlib.
- Dans cette figure, nous traçons la fréquence des valeurs dans la colonne « petal.length ».

```
plt.hist(df['petal.length'])
```

```
(array([37., 13.,  0.,  3.,  8., 26., 29., 18., 11.,  5.]),
 array([1.  , 1.59, 2.18, 2.77, 3.36, 3.95, 4.54, 5.13, 5.72, 6.31, 6.9 ]),
 <BarContainer object of 10 artists>)
```

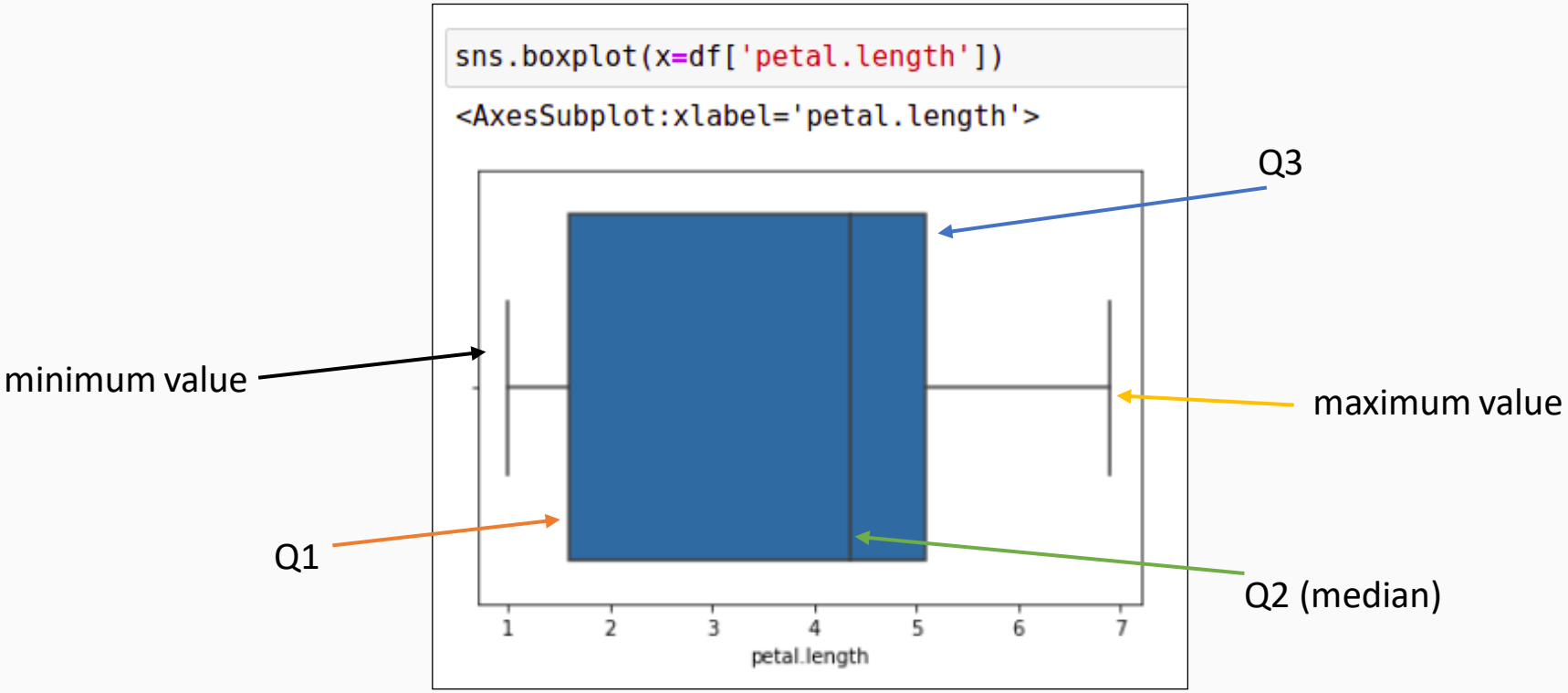


Meta Brains

Analyse univariée – données continues(7/7)

Graphique en boîte (“box plot”)

- Un graphique en boîte fournit d’excellents renseignements sur les données à l’aide d’un résumé à cinq chiffres : valeur minimale, premier quartile, deuxième quartile, troisième quartile et valeur maximale.
- Nous pouvons utiliser la fonction . boxplot() de matplotlib ou seaborn.



Meta Brains

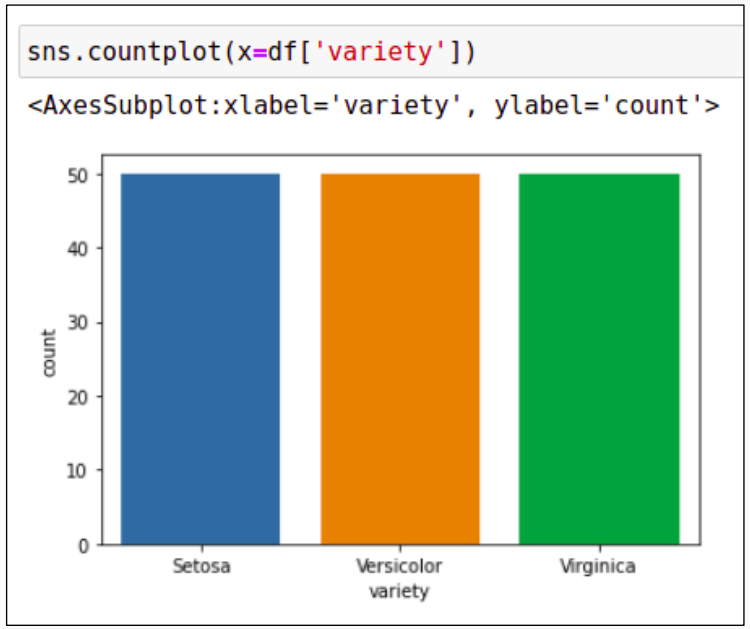
Analyse univariée – données catégoriques(1/2)

Dénombrement

- Nous pouvons utiliser des graphes de comptage pour effectuer l'EDA sur des données catégoriques.
- La fonction . countplot() de la bibliothèque seaborn trace le nombre total de chaque valeur sous forme de diagramme à barres.
- Dans la figure donnée, nous voyons que nous avons des instances égales de chaque catégorie dans la base de données.

```
sns.countplot(x=df['variety'])
```

```
<AxesSubplot:xlabel='variety', ylabel='count'>
```



Meta Brains

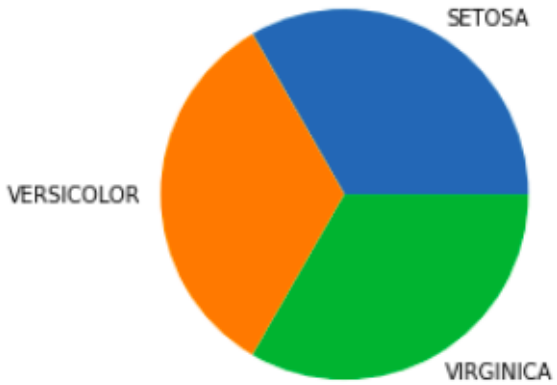
Analyse univariée – données catégoriques(2/2)

Diagramme circulaire

- Nous pouvons également visualiser la proportion de chaque catégorie à l'aide d'un diagramme circulaire comme le montre la figure.

```
_labels = ['SETOSA','VERSICOLOR','VIRGINICA']
plt.pie(df['variety'].value_counts(), labels = _labels)
```

```
([<matplotlib.patches.Wedge at 0x7f18c505e5b0>,
  <matplotlib.patches.Wedge at 0x7f18c505ea90>,
  <matplotlib.patches.Wedge at 0x7f18c505ef10>],
 [Text(0.5499999702695115, 0.9526279613277875, 'SETOSA'),
  Text(-1.0999999999999954, -1.0298943258065002e-07, 'VERSICOLOR'),
  Text(0.5500001486524352, -0.9526278583383436, 'VIRGINICA')])
```



Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 Installation de Python

3 **Analyse bivariée – continue et continue**

4 Installation d'Anaconda pour Windows OS

5 Installation d'Anaconda pour Mac OS

6 Installation d'Anaconda pour Ubuntu OS

7 Comment executer Python sur Jupyter

8 Gérer les répertoires dans Jupyter

9 Mise à l'echelle des caractéristiques

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Analyse bivariée – Continue et continue (1/4)

- L’analyse bivariée est utilisée pour étudier la relation entre exactement deux variables/caractéristiques/colonnes de l’ensemble de données.
- Considérez la base de données suivante.

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Meta Brains

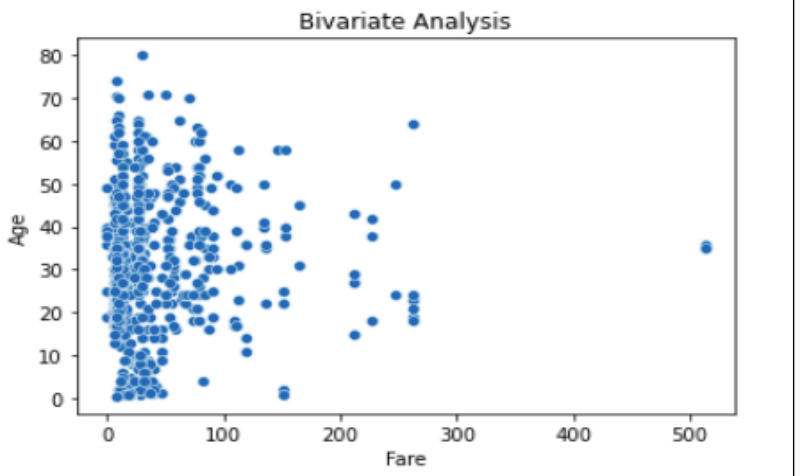
Analyse bivariée – Continue et continue (2/4)

Nuage de points

- Si les deux variables en question sont toutes deux continues, nous pouvons tracer un nuage de points entre elles pour avoir une idée de leur distribution.
- Dans la figure donnée, nous comparons la colonne « Âge » de la base de données à la colonne « Tarif », qui sont toutes deux continues.
- Il est évident que les deux caractéristiques sont indépendantes l'une de l'autre.

```
sns.scatterplot(x=df['Fare'], y=df['Age'])
plt.title('Bivariate Analysis')
```

```
Text(0.5, 1.0, 'Bivariate Analysis')
```



Meta Brains

Analyse bivariée – Continue et continue (3/4)

Corrélation

- Nous pouvons également trouver la corrélation entre deux caractéristiques continues.
- Si la corrélation est élevée, les deux caractéristiques sont significativement liées.
- Utilisez la fonction . corr() pour calculer la corrélation entre deux caractéristiques.

```
df[['Fare', 'Age']].corr()
```

	Fare	Age
Fare	1.000000	0.096067
Age	0.096067	1.000000

Meta Brains

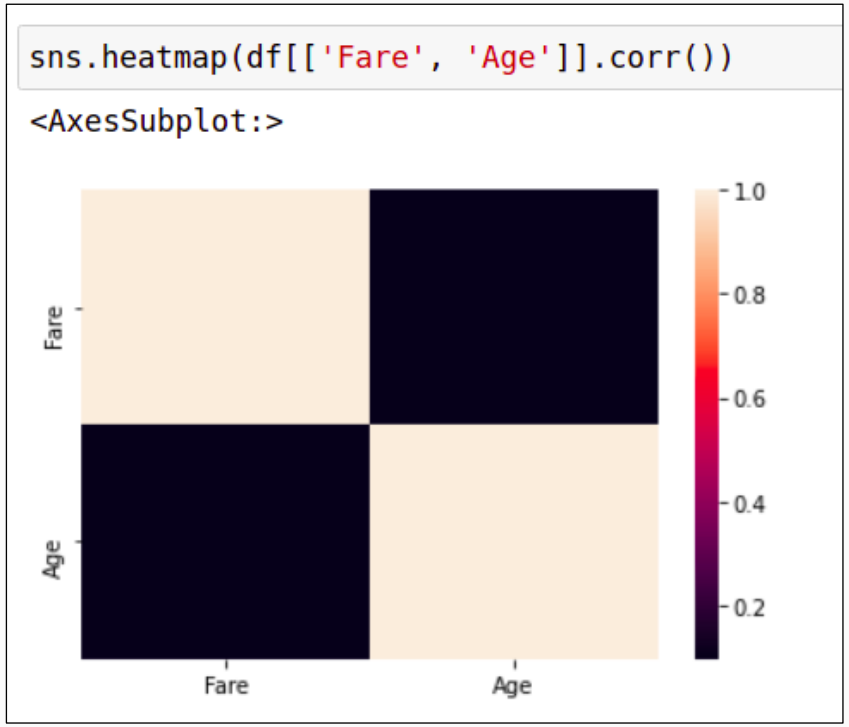
Analyse bivariée – Continue et continue (4/4)

Heatmp

- On peut aussi utiliser une heatmap pour visualiser graphiquement la corrélation entre quelque chose.
- Utilisez la méthode . heatmap() de la bibliothèque seaborn pour créer une heatmap.
- Fournir la base de données de corrélation que nous avons créée dans la diapositive précédente comme argument à la . heatmap(),

```
sns.heatmap(df[['Fare', 'Age']].corr())
```

```
<AxesSubplot:>
```



Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 Installation de Python

3 Analyse bivariée –
continue et continue

4 **Analyse bivariée –
Catégorique et catégorique**

5 Installation d'Anaconda
pour Mac OS

6 Installation d'Anaconda
pour Ubuntu OS

7 Comment executer Python sur
Jupyter

8 Gérer les répertoires
dans Jupyter

9 Mise à l'echelle des
caractéristiques

10 Travailler avec différents
types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Analyse bivariée – Catégorique et catégorique (1/3)

- Considérez la base de données suivante de la diapositive précédente.
- Nous verrons comment nous pouvons utiliser le graphique à barres pour identifier toute relation entre deux variables catégorielles, à savoir « Pclass » et « Survived ».

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Meta Brains

Analyse bivariée – Catégorique et catégorique(2/3)

- Nous choisissons d'abord les deux colonnes qui nous intéressent, à savoir « Pclass » et « Survived ».
- Nous regroupons ensuite cette nouvelle base de données en fonction de la colonne « Pclass » et appliquons la fonction . sum() qui permet de trouver le nombre total de survivants de chaque catégorie dans la Pclass.
- La sortie de cette partie est montrée ci-dessous.

```
survived_ratio = df[['Pclass', 'Survived']].groupby('Pclass').sum()
survived_ratio
```

Pclass	Survived
1	136
2	87
3	119

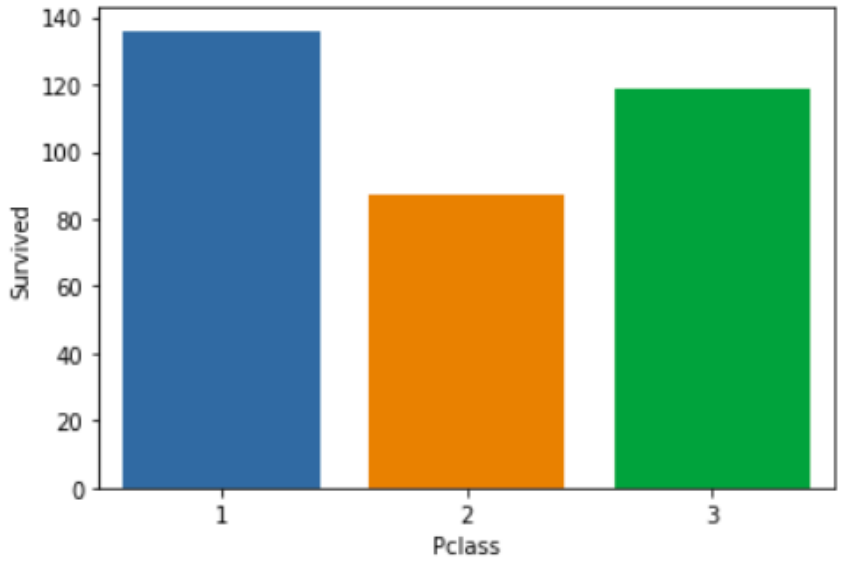
Meta Brains

Analyse bivariée – Catégorique et catégorique(3/3)

- Nous traçons ensuite un graphique à barres, avec « Pclass » sur l'axe X.
- Comme on peut le constater, plus de passagers de la « Pclass » 1 ont pu survivre que ceux de la « Pclass » 2 ou 3.

```
sns.barplot(x=survived_ratio.index, y=survived_ratio['Survived'])
```

```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 Installation de Python

3 Analyse bivariée –
continue et continue

4 Installation d'Anaconda
pour Windows OS

5 **Analyse bivariée – continue
et catégorique**

6 Installation d'Anaconda
pour Ubuntu OS

7 Comment executer Python sur
Jupyter

8 Gérer les répertoires
dans Jupyter

9 Mise à l'echelle des
caractéristiques

10 Travailler avec différents
types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Analyse bivariée – continue et catégorique (1/3)

- Parfois, nous voulons découvrir la relation entre une variable continue et une variable catégorielle.
- Considérez la base de données suivante de la diapositive précédente.

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Meta Brains

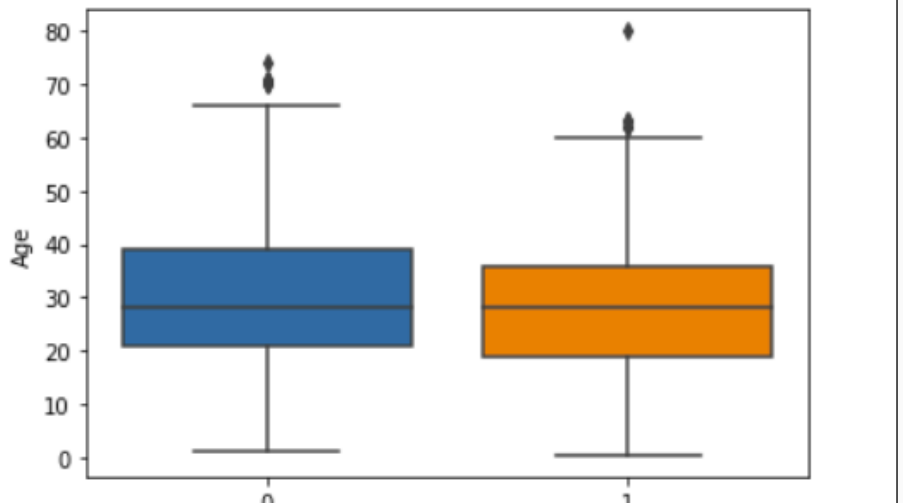
Analyse bivariée – continue et catégorique (2/3)

Box Plot (“boîte à moustache”)

- Nous pouvons utiliser un graphique en boîte pour effectuer EDA sur des données continues et catégoriques.
- Dans la figure donnée, nous traçons 'Age' (trait continu) contre 'Survive' (trait catégorique).
- L’intrigue suggère que les jeunes avaient une plus grande chance de survie.

```
sns.boxplot(x=df['Survived'], y=df['Age'])
```

```
<AxesSubplot:xlabel='Survived', ylabel='Age'>
```



Meta Brains

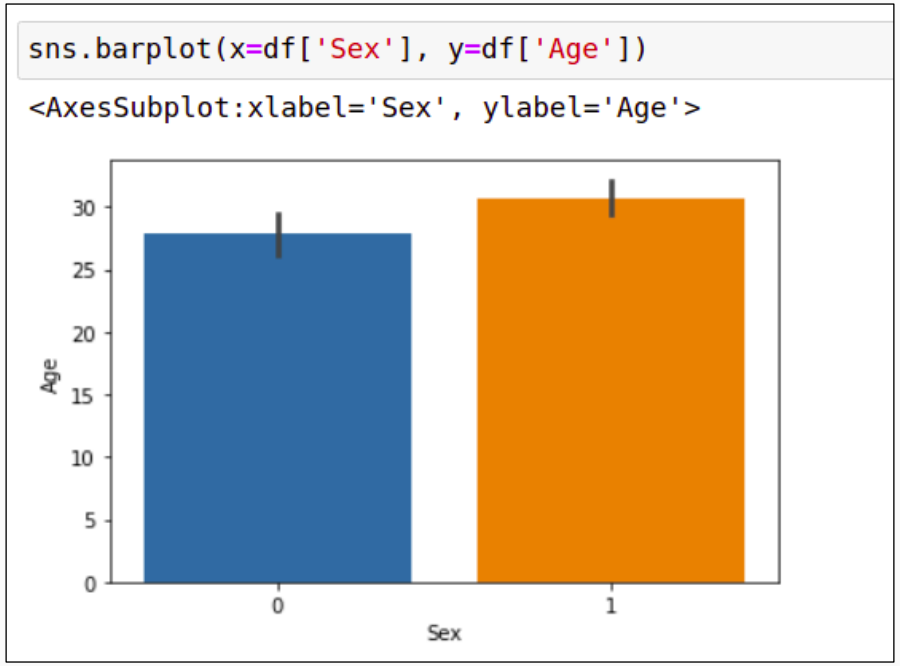
Analyse bivariée – continue et catégorique(3/3)

Graphiques à barres (Bar Plot)

- Les graphiques à barres peuvent également être utilisés pour l'analyse bivariée d'une caractéristique continue et d'une caractéristique catégorielle.
- Dans la figure donnée, nous comparons 'Âge' (trait continu) à 'Sexe' (trait catégorique). 1 dans la colonne 'Sexe' représente les passagers masculins et 0 représente les passagers féminins.
- La parcelle de bar suggère que les passagers âgés étaient principalement des hommes.

```
sns.barplot(x=df['Sex'], y=df['Age'])
```

```
<AxesSubplot:xlabel='Sex', ylabel='Age'>
```



Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 Installation de Python

3 Analyse bivariée –
continue et continue

4 Installation d'Anaconda
pour Windows OS

5 Installation d'Anaconda
pour Mac OS

6 **Détecter les valeurs
aberrantes**

7 Comment executer Python sur
Jupyter

8 Gérer les répertoires
dans Jupyter

9 Mise à l'echelle des
caractéristiques

10 Travailler avec différents
types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Détection des valeurs aberrantes

- Les valeurs aberrantes ou les anomalies dans les données sont les observations qui ne correspondent pas au modèle standard des données.
- Dans le chapitre 4, nous avons discuté des principales techniques pour détecter les valeurs aberrantes ou les anomalies, par exemple :
  - Détection médiane des anomalies
  - Détection d'anomalie basée sur la moyenne
  - Détection d'anomalie basée sur Z-score
  - Détection d'anomalie basée sur IQR
- Dans ce chapitre, nous apprendrons différentes façons de traiter ces valeurs aberrantes/anomalies dans les données.

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

Analyse exploratoire des
données

1 Introduction à Python

2 Installation de Python

3 Analyse bivariée –
continue et continue

4 Installation d'Anaconda
pour Windows OS

5 Installation d'Anaconda
pour Mac OS

6 Installation d'Anaconda
pour Ubuntu OS

7 **Traitement des valeurs
aberrantes**

8 Gérer les répertoires
dans Jupyter

9 Mise à l'echelle des
caractéristiques

10 Travailler avec différents
types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Traitement des valeurs aberrantes (1/3)

Correction des valeurs aberrantes

- Une façon naïve de traiter les valeurs aberrantes est de les retirer des données. Toutefois, cette approche n'est pas très bonne.
- Les valeurs aberrantes peuvent être supprimées des données de plusieurs façons.
- Considérez la série donnée. On peut dire que la valeur 150 est une valeur aberrante dans les données.

```
0      1
1      2
2      3
3      6
4      7
5      8
6    150
dtype: int64
```

Meta Brains

Traitement des valeurs aberrantes(2/3)

Correction des valeurs aberrantes

- Nous supprimons les lignes de la série pour lesquelles la valeur absolue du score Z est supérieure à 1,5, ce qui signifie que les valeurs se situent en dehors de 1,5 écart-type des données.

```
x = pd.Series([1, 2, 3, 6, 7, 8, 150])
mean = x.mean()
std = x.std()
z_scores = abs((x-mean)/std)
z_scores
```

```
0    0.441104
1    0.422941
2    0.404778
3    0.350288
4    0.332125
5    0.313962
6    2.265198
dtype: float64
```

```
outliers_removed = x[z_scores <= 1.5]
outliers_removed
```

```
0    1
1    2
2    3
3    6
4    7
5    8
dtype: int64
```

Meta Brains

Traitement des valeurs aberrantes(3/3)

Imputation moyenne/médiane

- Nous pouvons aussi remplacer les valeurs aberrantes par la moyenne ou la médiane.
- Dans la figure donnée, nous détectons les valeurs aberrantes dans les données à l'aide du Z-score et les remplaçons par la valeur médiane de la série.

```
x = pd.Series([1, 2, 3, 6, 7, 8, 150])
mean = x.mean()
std = x.std()
median = np.median(x)
z_scores = abs((x-mean)/std)
median
```

```
6.0
```

```
x[z_scores > 1.5] = median
x
```

```
0    1
1    2
2    3
3    6
4    7
5    8
6    6
dtype: int64
```

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas Dataframes et séries

Visualisation de données avec Python

Nettoyage des données

Analyse exploratoire des données

1 Introduction à Python

2 Installation de Python

3 Analyse bivariée – continue et continue

4 Installation d'Anaconda pour Windows OS

5 Installation d'Anaconda pour Mac OS

6 Installation d'Anaconda pour Ubuntu OS

7 Comment executer Python sur Jupyter

8 **Transformation des variables catégoriques**

9 Mise à l'echelle des caractéristiques

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Transformation des variables catégoriques(1/3)

- Nous avons discuté de la transformation des variables numériques au chapitre 4 en utilisant la normalisation.
- Dans ce chapitre, nous discuterons de la transformation des variables catégoriques.
- Il y a plusieurs façons de transformer les variables catégoriques pour les rendre plus significatives pour les machines, nous n'en discuterons que quelques-unes.
- Considérons le dataframe suivant. Nous allons transformer la variable 'Sex'.

	Name	Sex
0	Braund, Mr. Owen Harris	male
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	Heikkinen, Miss. Laina	female
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	Allen, Mr. William Henry	male

Meta Brains

Transformation des variables catégoriques(2/3)

Codage des étiquettes

- Dans l'encodage des étiquettes, nous remplaçons les données catégoriques par des chiffres.
- Nous remplaçons male par 1 et female par 0.

```
df['Sex'].replace({'male':1, 'female':0}, inplace=True)
df
```

	Name	Sex
0	Braund, Mr. Owen Harris	1
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0
2	Heikkinen, Miss. Laina	0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0
4	Allen, Mr. William Henry	1

Meta Brains

Transformation des variables catégoriques(3/3)

Code en fréquence

- Dans cette méthode d'encodage, nous remplaçons chaque valeur dans la variable catégorique par sa fréquence.
- Dans la figure donnée, nous avons remplacé les étiquettes masculines et féminines dans la colonne 'Sexe' par leurs fréquences.

```
freq = df['Sex'].value_counts()/len(df['Sex'])
freq
```

```
female    0.6
male      0.4
Name: Sex, dtype: float64
```

```
df['Sex'].replace({'male':freq['male'], 'female':freq['female']}, inplace=True)
df
```

	Name	Sex
0	Braund, Mr. Owen Harris	0.4
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0.6
2	Heikkinen, Miss. Laina	0.6
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0.6
4	Allen, Mr. William Henry	0.4

Meta Brains

Ressources

- https://www.kaggle.com/residentmario/univariate-plotting-with-pandas
- https://purnasaigudikandula.medium.com/exploratory-data-analysis-beginner-univariate-bivariate-and-multivariate-habberman-dataset-2365264b751

Meta Brains