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 Pandas

Section 4: Nettoyage des données

Meta Brains

Section 5 : Visualisation des données sur Python

Meta Brains

NETTOYAGE DES DONNÉES

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 **Introduction au nettoyage 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 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Introduction au nettoyage des données

- Les données extradites du monde reel contiennent des valeurs incorrectes, incompletes, non pertinentes ou manquantes qui doivent être nettoyées.
- Le nettoyage peut être effectué en modifiant, remplaçant ou supprimant ces valeurs.
- Le nettoyage des données est une partie fondamentale du cycle de vie des données.

https://www.educative.io/blog/what-is-data-cleaning

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage des données

2 **Qualité des données**

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 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Qualité des données

- Le monde d'aujourd'hui est entièrement axé sur la prise de décisions fondées sur les données. Par conséquent, la qualité de nos données influera en fin de compte sur la qualité de la décision que nous prendrons en fonction de ces données.
- Les données sont généralement considérées comme de grande qualité si elles « **conviennent aux utilisations prévues dans les opérations, la prise de décisions et la planification** ».
- Grâce à différentes techniques de nettoyage des données, nous pouvons améliorer la qualité de nos données extraites du monde réel.

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage des données

2 Mise en place Python

3 **Exemples d'anomalies**

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 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Exemples d'anomalies (1/2)

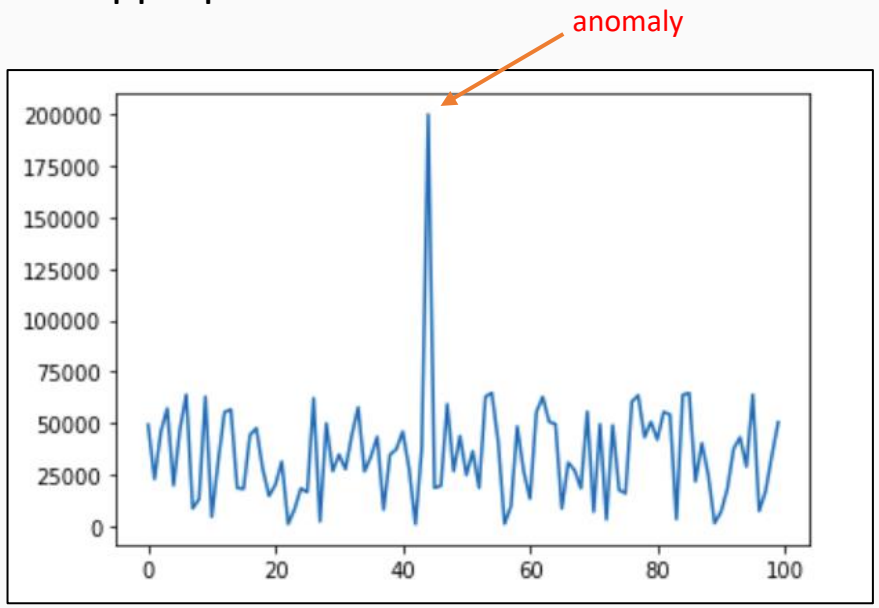
- Une anomalie dans les données, aussi appelée aberration, est l'observation qui diffère considérablement du modèle standard.
- Des anomalies dans les données pourraient survenir pour un certain nombre de raisons, comme l'erreur humaine.
- Considérez la série donnée contenant la largeur d'une table mesurée par 5 élèves différents.
  - La troisième valeur est significativement différente du reste – une anomalie !

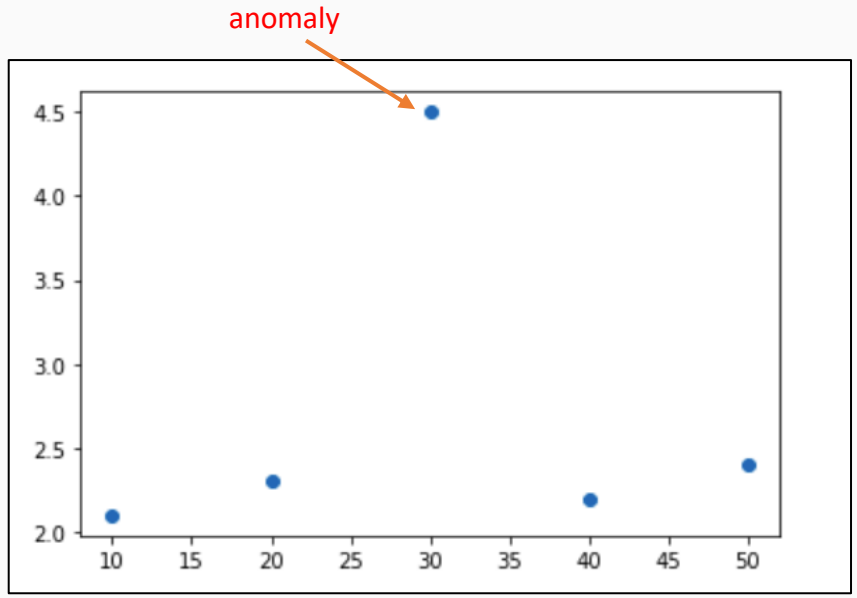
```
0    2.1
1    2.3
2    4.5
3    2.2
4    2.4
dtype: float64
```

Meta Brains

Exemples d'anomalies (2/2)

- Les données extradites du mode reel contiennent souvent des anomalies;
- Les données de température sur une période de 6 mois pourraient contenir des valeurs irrégulières en raison des changements énormes dans les conditions météorologiques.
- Il est important de détecter ces anomalies dans les données et de les traiter de façon appropriée.





Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage des données

2 Mise en place Python

3 Qu'est-ce que Jupyter

4 **Détection des anomalies grâce aux médianes**

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 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Détection des anomalies grâce aux médianes

- Une façon de détecter les anomalies est d'utiliser la valeur médiane.
- Nous fixons un seuil raisonnable et si pour une certaine valeur, |valeur – médiane| > seuil, alors la valeur est considérée comme une anomalie.

```
x = pd.Series([2.1, 2.3, 4.5, 2.2, 2.4])

median = np.median(x)
threshold = 2
outliers = []
for item in x:
    if abs(item - median) > threshold:
        outliers.append(item)

print(outliers)
```

```
[4.5]
```

Meta Brains

Quiz

Un certain ensemble de données a une valeur médiane 3, et le seuil de détection d'anomalie est 2. La troisième valeur de l'ensemble de données est 7.
Selon la détection d'anomalie basée sur la médiane, la troisième valeur est-elle une anomalie?

- Oui
- Non

Meta Brains

Quiz Time

Un certain ensemble de données a une valeur médiane 3, et le seuil de détection d'anomalie est 2. La troisième valeur de l'ensemble de données est 7.
Selon la détection d'anomalie basée sur la médiane, la troisième valeur est-elle une anomalie?

- Oui
- Non

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage des données

2 Mise en place Python

3 Qu'est-ce que Jupyter

4 Installation d'Anaconda pour Windows OS

5 **Détection des anomalies grâce à la moyenne**

6 Installation d'Anaconda pour Ubuntu OS

7 Comment executer Python sur Jupyter

8 Gérer les répertoires dans Jupyter

9 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Détection des anomalies grâce à la moyenne

- Une autre façon de détecter les anomalies est d'utiliser la moyenne et l'écart-type des données.
- Nous définissons la plage (moyenne – écart-type) <= valeur <= (moyenne + écart-type)
  - C'est-à-dire que toute valeur inférieure à (moyenne – écart-type) ou supérieure à (moyenne + écart-type) est considérée comme une anomalie.

```
x = pd.Series([2.1, 2.3, 4.5, 2.2, 2.5])

mean = np.mean(x)
std = np.std(x)
outliers = []
for item in x:
    if (item < mean - std) or (item > mean + std):
        outliers.append(item)

outliers
```

```
[4.5]
```

Meta Brains

Quiz

1. Un certain ensemble de données a une valeur moyenne de 20 et un écart-type de 2,5. Une certaine valeur x dans l'ensemble de données est de 16. Selon la détection d'anomalie basée sur la moyenne, x est une anomalie?

- Oui
- Non

Meta Brains

Quiz Time

1. Un certain ensemble de données a une valeur moyenne de 20 et un écart-type de 2,5. Une certaine valeur x dans l'ensemble de données est de 16. Selon la détection d'anomalie basée sur la moyenne, x est une anomalie?

- Oui
- Non

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage 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 **Détection des anomalies avec la méthode Z-score**

7 Comment executer Python sur Jupyter

8 Gérer les répertoires dans Jupyter

9 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Détection d'anomalies basée sur la méthode Z-score

- Une autre technique de détection d'anomalie est le Z-score.
- Z-score est une mesure statistique montrant le nombre d'écarts-types par rapport à la moyenne.
- Il est défini comme :
  - Z = (valeur – moyenne) / écart type
- Si le Z-score d'une valeur est supérieur à un seuil raisonnable, il est considéré comme une anomalie.

```
x = pd.Series([2.1, 2.3, 4.5, 2.2, 2.4])

mean = np.mean(x)
std = np.std(x)
outliers = []
for item in x:
    z_score = (item - mean) / std
    if z_score > 1.5:
        outliers.append(item)


print(outliers)
```

```
[4.5]
```

Meta Brains

Quiz

1. Un certain ensemble de données a une valeur moyenne de 12 et un écart-type de 2. Une certaine valeur x dans l'ensemble de données est 8. Calculer le score Z de la valeur x?

- 4
- 20
- 2
- 10

Meta Brains

Quiz Time

1. Un certain ensemble de données a une valeur moyenne de 12 et un écart-type de 2. Une certaine valeur x dans l'ensemble de données est 8. Calculer le score Z de la valeur x?

- 4
- 20
- 2
- 10

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage 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 **Détection des anomalies avec l'écart interquartile**

8 Gérer les répertoires dans Jupyter

9 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

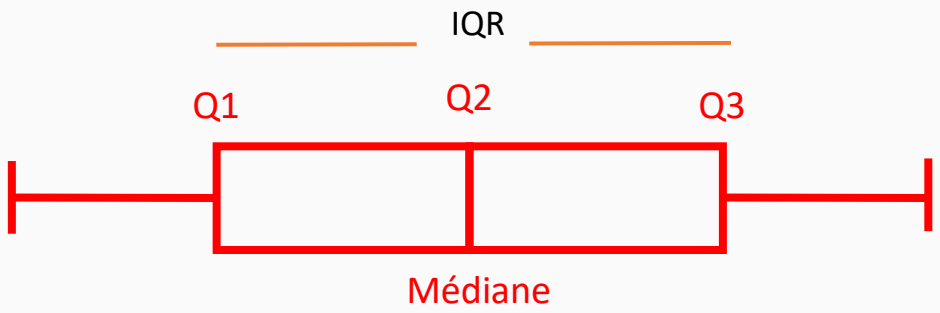
12 Opérateurs arithmétiques

Meta Brains

Ecart interquartile pour la detection d'anomalies (1/3)

IQR

- Nous pouvons également utiliser la plage interquartile (IQR) pour détecter les anomalies dans les données.
- Un quartile divise l'ensemble de données (trié du plus petit au plus grand) en 3 points et 4 intervalles.
- La plage interquartile (IQR) est la différence entre le 3e quartile et le 1er quartile (IQR = Q3 -Q1).



Meta Brains

Ecart interquartile pour la detection d'anomalies (2/3)

- Considérez la série de largeurs de la diapositive précédente comme une liste comme ci-dessous;
  - Largeurs = [2.3, 2.2, 4.5, 2.1, 2.5]
- Nous commençons par trier cette liste du plus petit au plus grand.
  - Largeurs = [2.1, 2.2, 2.3, 2.5, 4.5]
- Le premier quartile est à 25 p. 100, soit 2,2.
- Le deuxième quartile est de 50 %, soit 2,3 (médiane).
- Le troisième quartile est de 75 p. 100, soit 2,5.
- Utilisez la fonction numpy.percentile() pour obtenir les quartiles d'un jeu de données.
  - Trie automatiquement les données du plus petit au plus grand.

Meta Brains

Ecart interquartile pour la detection d'anomalies (3/3)

Anomalies

- Toute valeur inférieure à (Q1 1.5x IQR) ou supérieure à (Q3 + 1.5x IQR) est considérée comme une anomalie.

```
x = pd.Series([2.1, 2.3, 4.5, 2.2, 2.5])

Q1, Q3 = np.percentile(x, [25, 75])
IQR = Q3 - Q1
outliers = []
for item in x:
    if item < (Q1 - 1.5 * IQR) or item > (Q3 + 1.5 * IQR):
        outliers.append(item)

outliers
```

```
[4.5]
```

Meta Brains

Quiz

1. Tenez compte de l'ensemble de données x=[1,2,3,4,5]. Quel est le deuxième quartile (Q2) de l'ensemble de données?
   - 2
   - 3
   - 4

2. Calculer la plage interquartile pour l'ensemble de données x=[1,2,3,4,5].
   - 1
   - 2
   - 3

Meta Brains

Quiz Time

1. Tenez compte de l'ensemble de données x=[1,2,3,4,5]. Quel est le deuxième quartile (Q2) de l'ensemble de données?
   - 2
   - 3
   - 4

2. Calculer la plage interquartile pour l'ensemble de données x=[1,2,3,4,5].
   - 1
   - 2
   - 3

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage 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 **Gestion des valeurs manquantes**

9 Entrée/sortie

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Gestion des valeurs manquantes (1/6)

- Outre les valeurs aberrantes/anomalies, les données contiennent souvent des valeurs manquantes.
- Une valeur manquante signifie une perte d'information.
- Dans pandas, un NaN indique une valeur manquante.
- Considérez la base de données suivante appelée données avec une valeur manquante dans la colonne « Âge ».

	Name	Age
0	Edison	28
1	Edward	27
2	James	NaN
3	Neesham	36

Meta Brains

Gestion des valeurs manquantes(2/6)

Trouver des valeurs manquantes sur Pandas

- La fonction . isnull() nous dit si une cellule est vide ou non.
- Utilisez la fonction . sum() avec la fonction . isnull() pour trouver le nombre total de valeurs manquantes dans les données.

```
data.isnull()
```

	Name	Age
0	False	False
1	False	False
2	False	True
3	False	False

```
data.isnull().sum()
```

```
Name    0
Age     1
dtype: int64
```

Meta Brains

Gestion des valeurs manquantes (3/6)

Gestion des valeurs manquantes sur Pandas

- Il y a plusieurs façons de traiter ces valeurs manquantes.
- La méthode à utiliser dépend du type de données et de la tâche que les données sont censées accomplir.
- Les différentes méthodes utilisées sont :
- Suppression de lignes avec des valeurs manquantes.
- Remplacement des valeurs manquantes par la moyenne/médiane/mode.

Meta Brains

Gestion des valeurs manquantes (4/6)

Suppression de lignes avec des valeurs manquantes

- Une façon de traiter les valeurs manquantes est de supprimer les lignes contenant des valeurs manquantes.
- Utilisez le paramètre . dropna() avec inplace réglé sur True (vrai) pour supprimer les valeurs manquantes de l'ensemble de données.

```
data.dropna(inplace=True)
data
```

	Name	Age
0	Edison	28
1	Edward	27
3	Neesham	36

Meta Brains

Gestion des valeurs manquantes (5/6)

Remplacement des valeurs manquantes par la moyenne/médiane/mode

- Nous pouvons également remplacer les valeurs manquantes dans chaque colonne par l'une des mesures statistiques (moyenne/médiane/mode) de cette colonne.
- Utilisez la méthode . fillna() pour remplir les valeurs manquantes avec la moyenne, la médiane ou le mode.

```
data.fillna(data.mean(), inplace=True)
data
```

	Name	Age
0	Edison	28.000000
1	Edward	27.000000
2	James	30.333333
3	Neesham	36.000000

Meta Brains

Gestion des valeurs manquantes (6/6)

Remplacement des valeurs manquantes par la moyenne/médiane/mode

- Dans cet exemple, nous remplaçons la valeur manquante dans la colonne « Âge » par le mode de la colonne « Âge ».

```
data
```

	0	1
0	Edison	28.0
1	Edward	27.0
2	James	NaN
3	Neesham	36.0
4	Stuart	27.0

```
data['Age'].mode()
```

```
0    27.0
dtype: float64
```

```
data['Age'].fillna(data['Age'].mode()[0], inplace=True)
data
```

	Name	Age
0	Edison	28.0
1	Edward	27.0
2	James	27.0
3	Neesham	36.0
4	Stuart	27.0

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage 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'échelle des caractéristiques**

10 Travailler avec différents types de données

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Mise à l’échelle des caractéristiques(1/5)

- Parfois, nous pourrions vouloir normaliser la plage de chaque caractéristique (colonne) de l’ensemble de données.
- Par exemple, considérez la base de données ci-dessous où la plage de chaque caractéristique (colonne) est différente.
- Nous aimerions adapter toutes les fonctions à la même plage, par exemple, 0-1, 1-100, etc.

```
df
```

	Age	Salary
0	28.0	10000
1	27.0	15000
2	30.0	11000
3	36.0	11000
4	27.0	13000

Meta Brains

Mise à l'échelle des caractéristiques (2/5)

Normalisation

- Une méthode simple de mise à l'échelle des fonctionnalités est la normalisation, également appelée mise à l'échelle min-max.
- Pour chaque valeur d'une caractéristique (colonne), nous soustrayons la valeur minimale de la caractéristique particulière (colonne) et la divisons par la différence entre la valeur maximale et la valeur minimale de cette caractéristique (colonne).
- Valeur normalisée = (valeur initiale – minimum) / (maximum – minimum)
- Cette méthode met à l'échelle les caractéristiques sur la plage [0, 1]

Meta Brains

Mise à l'échelle des caractéristiques (3/5)

Normalisation

- Pour la mise à l'échelle min-max, utilisez la ligne de code suivante;
- normalized_df=(df-df.min())/(df.max()-df.min())
- Pandas utilisera automatiquement les valeurs min-max de la fonction (colonne) pour chaque fonction.

```
df = (df - df.min()) / (df.max() - df.min())
df
```

	Age	Salary
0	0.111111	0.0
1	0.000000	1.0
2	0.333333	0.2
3	1.000000	0.2
4	0.000000	0.6

Meta Brains

Mise à l'échelle des caractéristiques (4/5)

Standardisation

- En strandardisation, pour chaque valeur d'une caractéristique (colonne), nous soustrayons la moyenne de cette caractéristique (colonne) de la valeur et divisons le résultat par l'écart-type de cette caractéristique (colonne).
- Valeur standardisée= (valeur initiale – moyenne) / écart-type
- Par conséquent, l'écart-type de la caractéristique (colonne) devient 1.

Meta Brains

Mise à l'échelle des caractéristiques (5/5)

Standardisation

- Pour la standartdisation, utiliser la ligne de code suivante :
  standardized_df=(df-df.mean())/df.std()
- Encore une fois, Pandas utilisera automatiquement les valeurs de moyenne (colonne) et de std pour chaque fonctionnalité.

```
df = (df - df.mean()) / df.std()
df
```

	Age	Salary
0	-0.423109	-1.0
1	-0.687552	1.5
2	0.105777	-0.5
3	1.692435	-0.5
4	-0.687552	0.5

```
df.std()
```

```
Age       1.0
Salary    1.0
dtype: float64
```

Meta Brains

Quiz

1. L’échelle min-max est aussi connue sous le nom :
- Normalisation
- Standardisation

Meta Brains

Quiz

1. L’échelle min-max est aussi connue sous le nom :

- Normalisation
- Standardisation

Meta Brains

Remise à niveau Python

Data Science avec Python

Python Pandas
Dataframes et séries

Visualisation de
données avec Python

Nettoyage des données

1 Introduction au nettoyage 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 Entrée/sortie

10 **Expression rationnelle**

11 Variables

12 Opérateurs arithmétiques

Meta Brains

Expressions rationnelles (1/8)

- Une expression rationnelle (regular expression en anglais), ou RegEx est une expression contenant une séquence de caractères pour faire correspondre les motifs dans les chaînes.
- Presque tous les principaux langages de programmation ont une implémentation pour RegEx.
- Le module 're' de Python est utilisé pour le motif correspondant en utilisant RegEx en Python.
- Les fonctions suivantes sont disponibles dans le module 're';
  - findall()
  - search()
  - sous()

Meta Brains

Expressions rationnelles (2/8)

re.findall()

- re.findall() est utilisé pour correspondre à toutes les occurrences d'un motif dans une chaîne.
- Une liste avec toutes les correspondances est renvoyée.
- Dans la figure suivante, nous trouvons combien de fois le mot 'Python' apparaît dans la chaîne donnée.

```
import re
```

```
txt = 'Python is my favorite programming language. I love Python.'
x = re.findall('Python', txt)
x
```

```
['Python', 'Python']
```

```
len(x)
```

```
2
```

Meta Brains

Expressions rationnelles (3/8)

re.findall()

- Dans la figure donnée, on vérifie si la chaîne x commence ou non par le mot 'Python'.
- Le caractère ^ retourne une correspondance seulement si la chaîne commence avec le modèle donné après le symbole ^.
- La chaîne y contient le mot Python mais ne commence pas par lui, donc nous obtenons une liste vide.

```
import re
```

```
x = 'Python is my favorite programming language.'
re.findall('^Python', x)
```

```
['Python']
```

```
y = 'I love Python.'
re.findall('^Python', y)
```

```
[]
```

Meta Brains

Expressions rationnelles (4/8)

re.findall()

- Pour faire correspondre les nombres dans une chaîne, nous utilisons la séquence \d.
- Un signe + à la fin de \d fait en sorte que le nombre tel que 50 est traité comme 50 et non comme 5 et 0.
- Vous pouvez trouver une liste de séquences sur https://www.w3schools.com/python/python_regex.asp

```
import re
```

```
txt = 'Python was released in 1991.'
re.findall('\d', txt)
```

```
['1', '9', '9', '1']
```

```
txt = 'Python was released in 1991.'
re.findall('\d+', txt)
```

```
['1991']
```

Meta Brains

Expressions rationnelles (5/8)

re.findall()

- Pour trouver des correspondances dans une série, nous convertissons d'abord la série en une chaîne en utilisant la méthode . to_string().

```
import pandas as pd
import re
```

```
txtList = ['Pakistan', 'Indonesia', 'Jordan', 'Pakistan']
txt = pd.Series(txtList)
txt
```

```
0     Pakistan
1    Indonesia
2       Jordan
3     Pakistan
dtype: object
```

```
re.findall('Pakistan', txt.to_string())
```

```
['Pakistan', 'Pakistan']
```

Meta Brains

Expressions rationnelles (6/8)

re.search()

- re.search() retourne l'objet correspondant en cas de correspondance dans la chaîne.
- Nous pouvons obtenir la position de la correspondance en utilisant la méthode . span() de l'objet de correspondance

```
import pandas as pd
import re
```

```
txt = 'Hello World'
match_object = re.search('World', txt)
match_object
```

```
<re.Match object; span=(6, 11), match='World'>
```

```
match_object.span()
```

```
(6, 11)
```

Meta Brains

Expressions rationnelles (7/8)

re.sub()

- Pour remplacer le texte d'une chaîne par un texte différent, utilisez la méthode re.sub().

```
import pandas as pd
import re
```

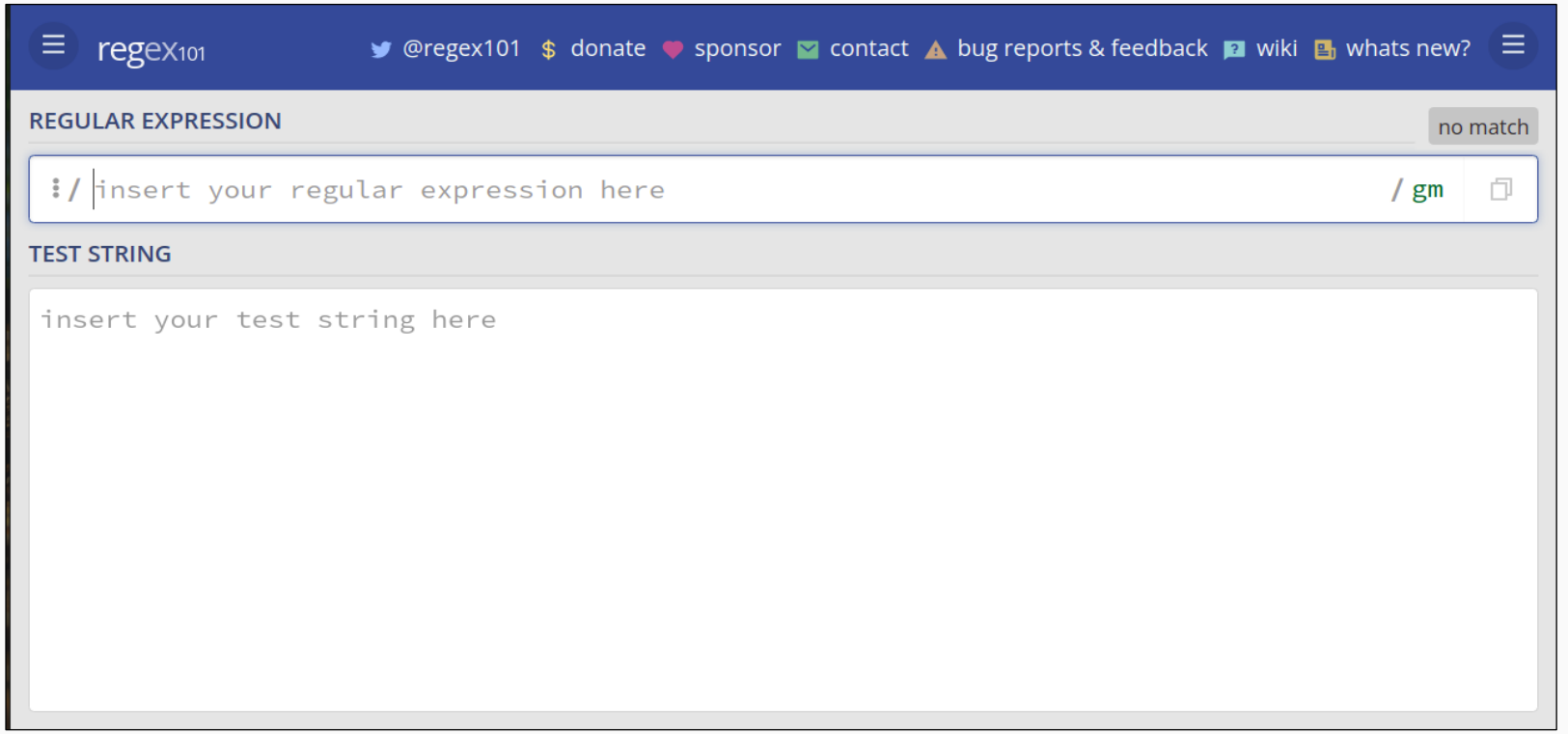
```
txt = 'C is my favorite programming language.'
re.sub(pattern='C', repl='Python', string=txt)
```

```
'Python is my favorite programming language.'
```

Meta Brains

Expressions rationnelles (8/8)

- https://regex101.com/ est un très bon site pour créer et tester des expressions régulières.
- Nous avons également ajouté des ressources pour en savoir plus sur l'expression régulière en Python dans la diapositive 'Ressources'.



Meta Brains

Ressources

- https://www.w3schools.com/python/pandas/pandas_cleaning.asp
- https://www.geeksforgeeks.org/interquartile-range-to-detect-outliers-in-data/
- https://www.kdnuggets.com/2021/04/data-science-101-normalization-standardization-regularization.html
- https://www.w3schools.com/python/python_regex.asp

Meta Brains