

**DATA SCIENCE**

**LES ACCIDENTS A**

**LOS ANGELES**



# PLAN

**01**

Problématique

**02**

Objectif

**03**

Solution

**04**

Outils utilisés

**05**

Code

**06**

Résultat

# Problématique

La croissance continue du nombre de véhicules et de conducteurs à l'échelle mondiale suscite des préoccupations croissantes. Évaluer le taux d'accidents routiers et identifier leurs principales causes devient crucial.

Cette problématique souligne l'importance de comprendre les facteurs contribuant aux accidents, ouvrant ainsi la voie à des analyses approfondies et à la mise en place de mesures pour améliorer la sécurité routière.

# Objectif

L'objectif de cette phase consiste à recueillir des données à Los Angeles à partir du site Web dédié aux statistiques sur les collisions de la circulation.

À l'aide d'un script écrit en Python, nous utilisons la bibliothèque "BeautifulSoup" pour extraire ces données et les sauvegarder sous forme de fichiers CSV

# Solution

Notre approche repose sur un script Python automatisant la collecte de données à partir d'un site spécifié. En utilisant la bibliothèque "Beautiful Soup", le script analyse le code HTML du site pour extraire les informations liées aux collisions de la circulation.

Les données extraites sont ensuite organisées et stockées dans des fichiers CSV, offrant ainsi une méthode automatisée et efficace pour analyser les statistiques sur les accidents routiers au Canada.

# Outils de développement

Langage de programmation :



Les bibliothèques :



Navigateur Web :



Environnement de développement et IDE :



# Script Python

Avoir le lien url:

```
import os
import requests
from bs4 import BeautifulSoup
import csv

url = "https://tc.canada.ca/en/road-transportation/publications(canadian-motor-vehicle-traffic-collision-statistics-2018#wb5"

# Send a request to the URL
response = requests.get(url)
```

## Préciser la classe, et le path du fichier csv générés

```
# Send a request to the URL
response = requests.get(url)

if response.status_code == 200:
    # Parse the HTML content
    soup = BeautifulSoup(response.text, 'html.parser')

    # Find tables directly, based on the structure of the HTML
    tables = soup.find_all('table', {'class': 'table table-condensed table-bordered'})

    # Specify the path where you want to save the CSV files
    save_path = r'C:\DataScience'
    os.makedirs(save_path, exist_ok=True) # Create the directory if it doesn't exist
```

Parcourir les tables dans la page web:

```
for index, table in enumerate(tables):
    # Extract data from the table
    table_data = []
    for row in table.find_all('tr'):
        row_data = [cell.get_text(strip=True) for cell in row.find_all(['th', 'td'])]
```

Extraire les données des tables:

```
for index, table in enumerate(tables):
    # Extract data from the table
    table_data = []
    for row in table.find_all('tr'):
        row_data = [cell.get_text(strip=True) for cell in row.find_all(['th', 'td'])]
        table_data.append(row_data)
```

# Résultat

4 Fichiers CSV générés avec données nettoyés:

- Collisions et victimes 1991-2010
- Décès et blessures par groupe d'âge 2010
- Décès par classe d'utilisateur 2006-2010
- Taux de victimes 2010



# PLAN PHASE 2

**01**

Problématique

**02**

Objectif

**03**

Solution

**04**

Outils utilisés

**05**

Code

**06**

Résultat

# Problématique

Dans un environnement où les données sur les accidents de la route sont essentielles pour la sécurité publique et les politiques de transport, la problématique réside dans la garantie de la qualité et de la fiabilité de ces données.

Après la collecte des données dans la PHASE1 nous devons à présent passer au nettoyage de ces données.

Comment utiliser efficacement Python pour nettoyer et préparer ces données, assurant ainsi des analyses précises et des décisions informées pour améliorer la sécurité routière?

# Objectif

L'objectif de cette phase consiste aux nettoyages des données à Los Angeles à partir des fichiers CSV générés dans la phase 1 qui est dédié aux groupement de données sur les collisions de la circulation.

À l'aide d'un script écrit en Python, nous utilisons la bibliothèque "PANDA" pour le nettoyage de ces données.

# Solution

Pour nettoyer efficacement les données sur les accidents de la route, un script Python est une solution idéale.

En utilisant des bibliothèques telles que Pandas et NumPy, le script peut automatiser l'identification et le traitement des valeurs manquantes, des aberrations et des incohérences. En fournissant également des visualisations claires, il facilite une analyse approfondie et garantit la qualité des données.

# Outils de développement

Langage de programmation :



Les bibliothèques :



Navigateur Web :



Environnement de développement et IDE :



# Préciser le path du fichier csv générés avec les données avant nettoyage

```
"source": [
    "from bs4 import BeautifulSoup\n",
    "import pandas as pd\n",
    "import requests\n",
    "\n",
    "\n",
    "# Chemin vers le fichier CSV sur Google Drive\n",
    "chemin_fichier_csv = 'C:\\\\DataScience\\\\Traffic_Collision_Data_from_2010_to_Present_20240210.csv'\\n",
    "\n",
    "# Charger les données CSV dans un DataFrame\\n",
    "donnees = pd.read_csv(chemin_fichier_csv)\\n",
    "\n",
    "# Afficher des informations de base sur l'ensemble de données avant le nettoyage\\n",
    "print(\"Informations sur l'ensemble de données (Avant le nettoyage):\")\\n",
    "print(donnees.info())\\n",
    "\n",
    "# Statistiques sommaires avant le nettoyage\\n",
    "print(\"\\nStatistiques sommaires (Avant le nettoyage):\")\\n",
    "print(donnees.describe())\\n",
    "\n",
    "# Vérifier les valeurs manquantes avant le nettoyage\\n",
    "print(\"\\nValeurs manquantes (Avant le nettoyage):\")\\n",
    "print(donnees.isnull().sum())\\n",
    "\n",
    "# Nettoyage des données\\n",
    "# Supprimer les lignes avec des valeurs manquantes\\n",
    "donnees.dropna(inplace=True)\\n",
    "\n",
    "# Supprimer les lignes en double\\n",
    "donnees.drop_duplicates(inplace=True)\\n",
    "\n",
    "# Afficher des informations de base sur l'ensemble de données nettoyé\\n",
    "print(\"\\nInformations sur l'ensemble de données (Après le nettoyage):\")\\n",
    "print(donnees.info())\\n",
    "\n",
    "# Statistiques sommaires de l'ensemble de données nettoyé\\n",
    "print(\"\\nStatistiques sommaires (Après le nettoyage):\")\\n",
    "print(donnees.describe())\\n",
    "\n",
    "# Vérifier les valeurs manquantes dans l'ensemble de données nettoyé\\n",
```

## visualiser le nombre d'accidents par zone

```
"# Compter le nombre d'accidents par zone\n",
"accidents_par_zone = donnees[\"Area Name\"].value_counts()\n",
"\n",
"# Visualiser le nombre d'accidents par zone\n",
"plt.figure(figsize=(12, 8))\n",
"accidents_par_zone.plot(kind=\"bar\")\n",
"plt.title(\"Nombre d'accidents par zone à Los Angeles\")\n",
"plt.xlabel(\"Zone\")\n",
"plt.ylabel(\"Nombre d'accidents\")\n",
"plt.xticks(rotation=45)\n",
"plt.tight_layout()\n",
"plt.show()\n"
```

## nombre de victimes par sexe

```
source ->
"## Compter le nombre d'accidents par sexe des victimes\n",
"accidents_par_sexe = donnees[\"Victim Sex\"].value_counts()\n",
"\n",
"## Visualiser le nombre d'accidents par sexe des victimes\n",
"plt.figure(figsize=(8, 6))\n",
"accidents_par_sexe.plot(kind=\"bar\", color=\"skyblue\")\n",
"plt.title(\"Nombre d'accidents par sexe des victimes à Los Angeles\")\n",
"plt.xlabel(\"Sexe des victimes\")\n",
"plt.ylabel(\"Nombre d'accidents\")\n",
"plt.xticks(rotation=0)\n",
"plt.tight_layout()\n",
"plt.show()"
```

# Résultat

**4 Fichiers CSV générés:**

- Collisions et victimes 1991-2010
- Décès et blessures par groupe d'âge 2010
- Décès par classe d'utilisateur 2006-2010
- Taux de victimes 2010

**Merci pour votre attention**