

accidents

June 20, 2021

1 Open-Data des accidents de la circulation 2019

1.1 Chargement d'un fichier CSV

Nous allons charger le fichier `caracteristiques` de 2019 dans un tableau de dictionnaire de donnée

```
[81]: import csv

def load_csv_file(filename):
    with open(filename, newline='') as csvfile:
        # On utilise la bibliothèque csv de Python pour transformer chaque
        ↪enregistrement du fichier en un objet
        reader = csv.DictReader(csvfile, delimiter=';')
        # On construit le tableau en compréhension contenant chaque enregistrement
        ↪sous la forme d'un dictionnaire
        return [record for record in reader]

accidents_2019 = load_csv_file('./data/2019-caracteristiques.csv')
print(accidents_2019[0])
```

```
{'Num_Acc': '201900000001', 'jour': '30', 'mois': '11', 'an': '2019', 'hrmn':
'01:30', 'lum': '4', 'dep': '93', 'com': '93053', 'agg': '1', 'int': '1', 'atm':
'1', 'col': '2', 'adr': 'AUTOROUTE A3', 'lat': '48,8962100', 'long':
'2,4701200'}
```

1.2 Réponses aux questions

Nous allons principalement utiliser des listes en compréhension pour filtrer les données

```
[82]: # Liste des accidents du Rhône
accidents_2019_rhones = [accident for accident in accidents_2019 if
    ↪accident['dep'] == '69']
print(f'1. {len(accidents_2019_rhones)} accidents dans le département du
    ↪Rhône')

# Liste des numéros d'accident impliquant au moins un vélo
vehicules_2019 = load_csv_file('./data/2019-vehicules.csv')
velos_2019 = [vehicule for vehicule in vehicules_2019 if vehicule['catv'] ==
    ↪'1']
```

```

numeros_accidents_avec_velos = [numero['Num_Acc'] for numero in velos_2019]

accidents_avec_velos_2019 = [accident for accident in accidents_2019 if
    ↳accident['Num_Acc'] in numeros_accidents_avec_velos]
print(f'2. {len(accidents_avec_velos_2019)} accidents impliquant au moins un
    ↳vélo')

# Fonction retournant la date de l'accident
def date_accident(accident):
    return f'{accident["an"]}-{accident["mois"]:0>2}-{accident["jour"]
    ↳:0>2}-{accident["hrmn"]}'

# Les accidents doivent être ordonnés par date, pour cela il faut construire la
    ↳date à partir des propriétés an, mois, jour et heures/minutes
accidents_2019_sorted = sorted(accidents_2019, key=date_accident)
premier_accident = accidents_2019_sorted[0]
print(f'3. Le premier accident de l\'année a eu lieu au coordonnées
    ↳({premier_accident["lat"]}, {premier_accident["long"]})')

```

1. 2523 accidents dans le département du Rhône
2. 4834 accidents impliquant au moins un vélo
3. Le premier accident de l'année a eu lieu au coordonnées (-20,9611120, 55,6577380)

```

[83]: # Constuction du jeu de données indiquant les accidents de vélos de 2019

usagers_2019 = load_csv_file('./data/2019-usagers.csv')
numeros_accidents_mortels_2019 = [usager['Num_Acc'] for usager in usagers_2019
    ↳if usager['grav'] == '2']
accidents_velos_2019 = [{
    'Num_Acc': accident['Num_Acc'],
    'dep': accident['dep'],
    'lat': accident['lat'].replace(',', ' '),
    'long': accident['long'].replace(',', ' '),
    'date': f'{accident["an"]}-{accident["mois"]:0>2}-{accident["jour"]
    ↳:0>2}',
    'hrmn': accident['hrmn'],
    'atm': accident['atm'],
    'mortel': accident['Num_Acc'] in numeros_accidents_mortels_2019,
} for accident in accidents_avec_velos_2019]

print(accidents_velos_2019[0:2])

```

```

[{'Num_Acc': '201900000037', 'dep': '67', 'lat': '48.5708480', 'long':
'7.7587680', 'date': '2019-11-29', 'hrmn': '13:40', 'atm': '2', 'mortel':
False}, {'Num_Acc': '201900000038', 'dep': '66', 'lat': '42.6969500', 'long':
'2.8993500', 'date': '2019-11-29', 'hrmn': '13:50', 'atm': '1', 'mortel':
False}]

```

```
[84]: import folium

accidents_velos_2019_rhones = [accident for accident in accidents_velos_2019 if
    ↪accident['dep'] == '69']

map = folium.Map(location=[45.7, 4.8], zoom_start=11)

COND_ATM = {
    '-1': 'Non renseigné',
    '1': 'Normale',
    '2': 'Pluie légère',
    '3': 'Pluie forte',
    '4': 'Neige - grêle',
    '5': 'Brouillard - fumée',
    '6': 'Vent fort - tempête',
    '7': 'Temps éblouissant',
    '8': 'Temps couvert',
    '9': 'Autre',
}

for accident in accidents_velos_2019_rhones:
    coordonnees = [accident['lat'], accident['long']]

    cond_atm = COND_ATM[accident['atm']]

    popup=f'<b>N°{accident["Num_Acc"]}</b><br>{accident["date"]}<br>
    ↪{accident["hrmn"]}<br>{cond_atm}'

    # Si l'accident est mortel, il est affiché en rouge
    if accident['mortel']:
        color = 'red'
    else:
        color = 'blue'

    # L'icône est dépendant des conditions atmosphériques
    if accident['atm'] != '1':
        icon = folium.Icon(icon="cloud", color=color)
    else:
        icon = folium.Icon(color=color)

    # Construction et ajout du marqueur
    folium.Marker(
        coordonnees,
        popup=popup,
        icon=icon,
    ).add_to(map)
```

```
map
```

```
[84]: <folium.folium.Map at 0x10f7a6820>
```

