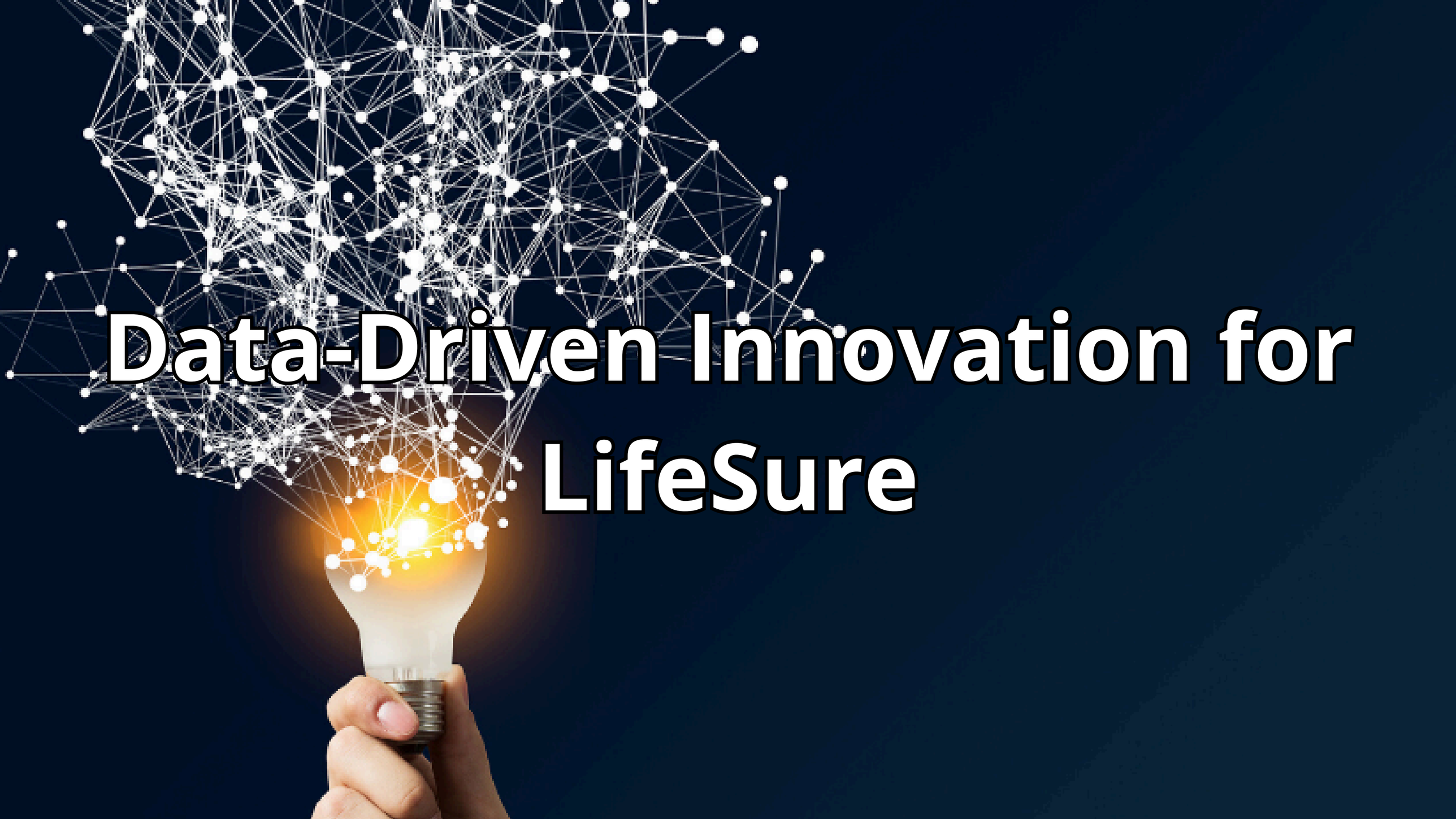
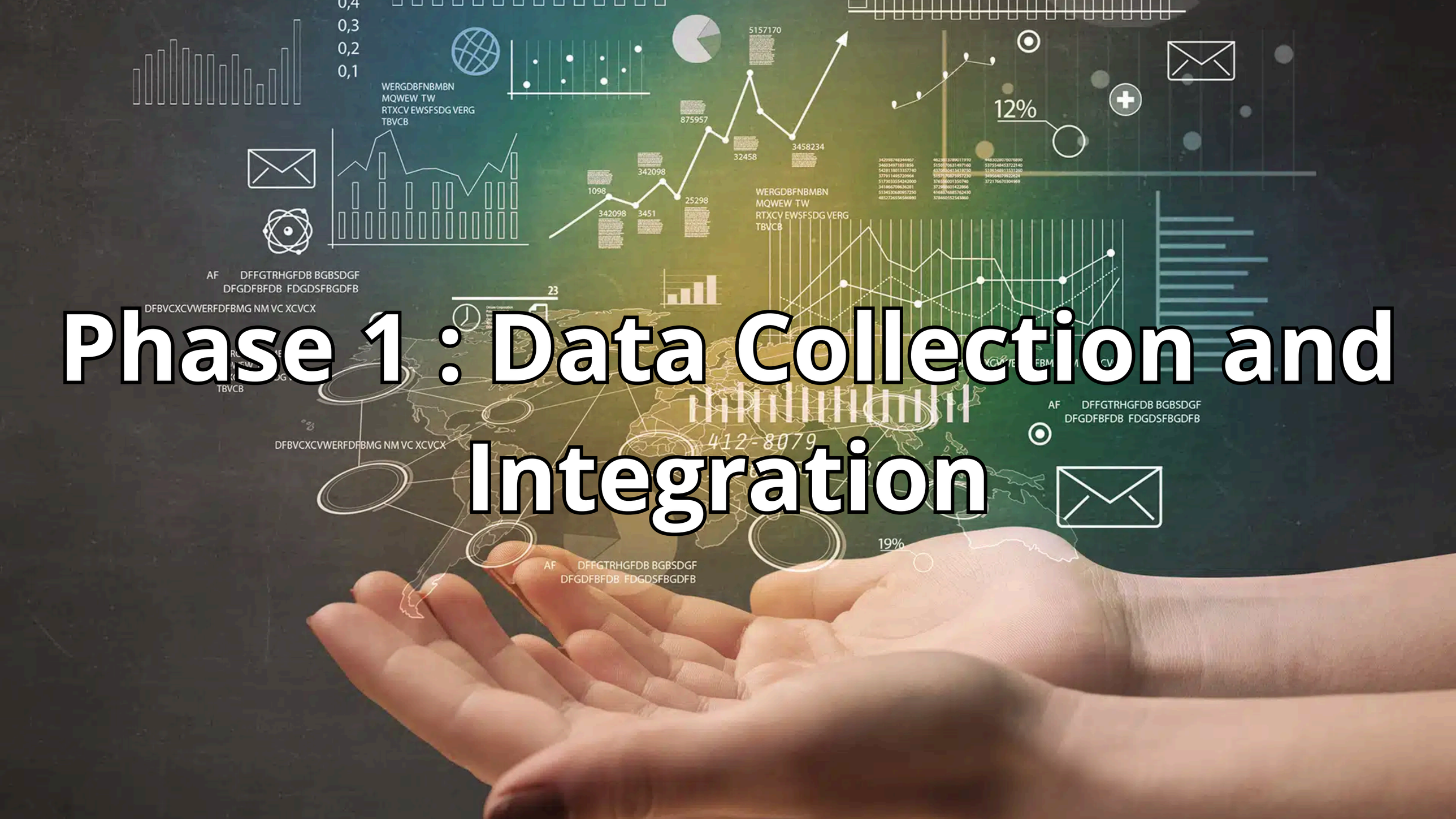

Data-Driven Innovation for
LifeSure

Summary

1. **Phase 1 : Data Collection and Integration**
    a. **Customer personnality**
    b. **Travel insurance prediction**
    c. **Insurance and medical cost**
    d. **Global environment indicators**
    e. **Twitter airline sentiment**
    f. **Vehicle insurance data**
2. **Phase 2 : Data Analysis and Specifications**
    a. **EDA**
    b. **Key insights**
    c. **Specifications document**
3. **Phase 3 : Vizualizations with Power BI**
    a. **Dashboard**


Phase 1 : Data Collection and Integration

Customer personality

Data pre-processing

- Droping useless columns
- Creating new features for advenced analysis
- Encoding categorical variables for visualisations

Customer personality

Travel insurance prediction

```
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           1987 non-null   int64
 1   Age                  1987 non-null   int64
 2   Employment Type      1987 non-null   object
 3   GraduateOrNot        1987 non-null   object
 4   AnnualIncome         1987 non-null   int64
 5   FamilyMembers        1987 non-null   int64
 6   ChronicDiseases      1987 non-null   int64
 7   FrequentFlyer        1987 non-null   object
 8   EverTravelledAbroad  1987 non-null   object
 9   TravelInsurance      1987 non-null   int64
dtypes: int64(6), object(4)
```

Data pre-processing

- Droping useless columns
- Creating new features for advenced analysis
- Encoding categorical variables for visualisations

```python
# Catégoriser l'âge en groupes
bins = [18, 30, 40, 50, 60, 100]
labels = ['18-30', '31-40', '41-50', '51-60', '60+']
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)
```

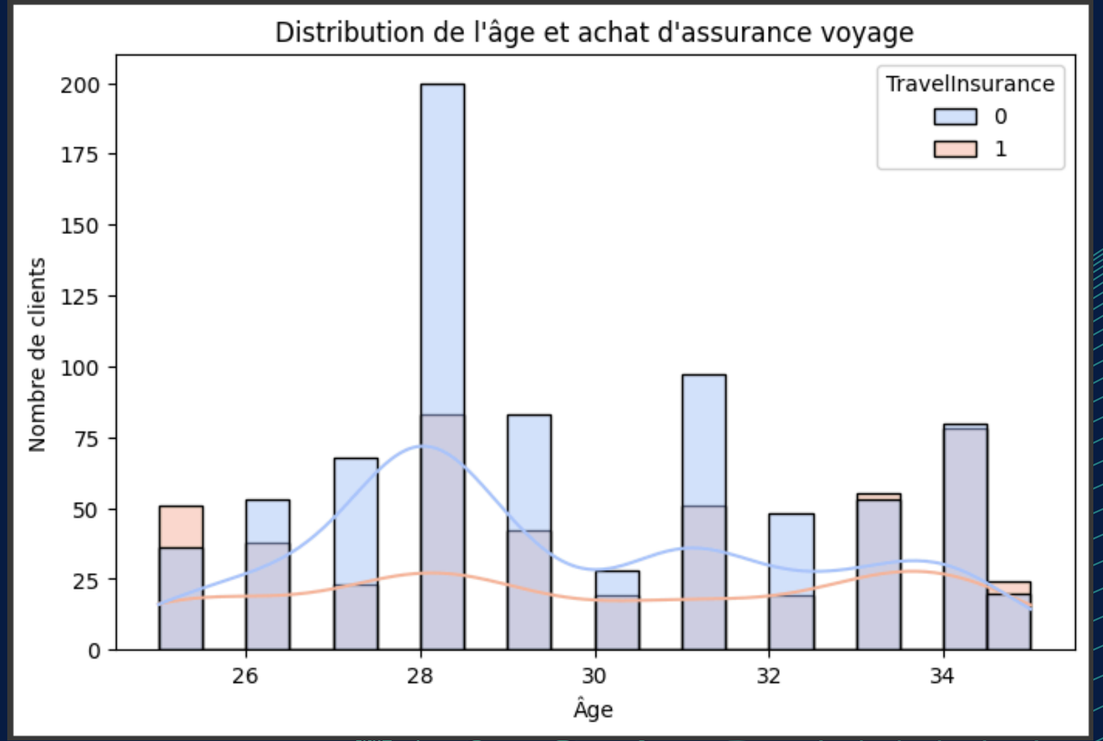
```python
# Encoder les variables catégoriques pour les visualisations
df['GraduateOrNot'] = df['GraduateOrNot'].map({'Yes': 1, 'No': 0})
df['FrequentFlyer'] = df['FrequentFlyer'].map({'Yes': 1, 'No': 0})
df['EverTravelledAbroad'] = df['EverTravelledAbroad'].map({'Yes': 1, 'No': 0})
```

Travel insurance prediction

Data analysis

- Creation of different graphs to visualise the data
- Use of new functions for interesting results

```python
# 1. Répartition de l'âge avec l'achat d'une assurance voyage
plt.figure(figsize=(8, 5))
sns.histplot(df, x="Age", hue="TravelInsurance", kde=True, bins=20, palette="coolwarm")
plt.title("Distribution de l'âge et achat d'assurance voyage")
plt.xlabel("Âge")
plt.ylabel("Nombre de clients")
plt.show()
```



Insurance and medical cost

Data analysis

- Used df.drop_duplicates() to remove any duplicate rows
- Used df.isnull().sum() to verify if any column had missing data.

3. Data Cleaning

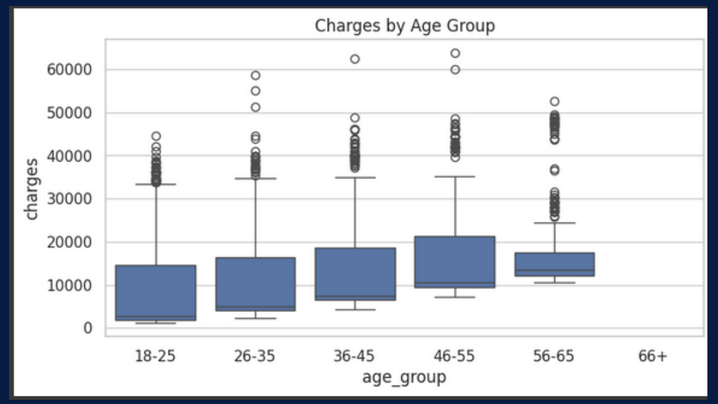
```
[ ] # Duplicates removal
    df = df.drop_duplicates()

    # Potential missing values
    print("Missing values:\n", df.isnull().sum())
```

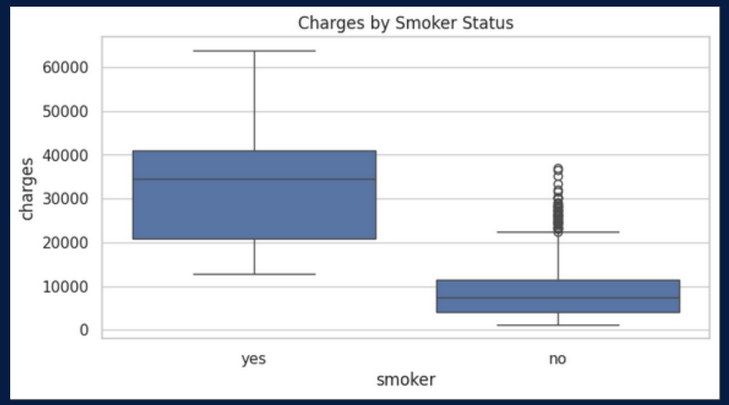
```
Missing values:
 age         0
sex          0
bmi          0
children     0
smoker       0
region       0
charges      0
dtype: int64
```

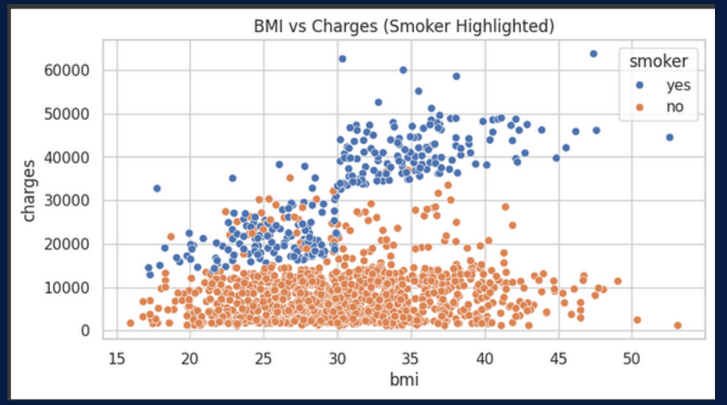
Insurance and medical cost

Data analysis

- Smokers Pay a lot more for Insurance
- Age Strongly Affects Insurance Charges
- High BMI and Smoking Together Lead to Very High Charges







Global environment indicators

Data pre-processing

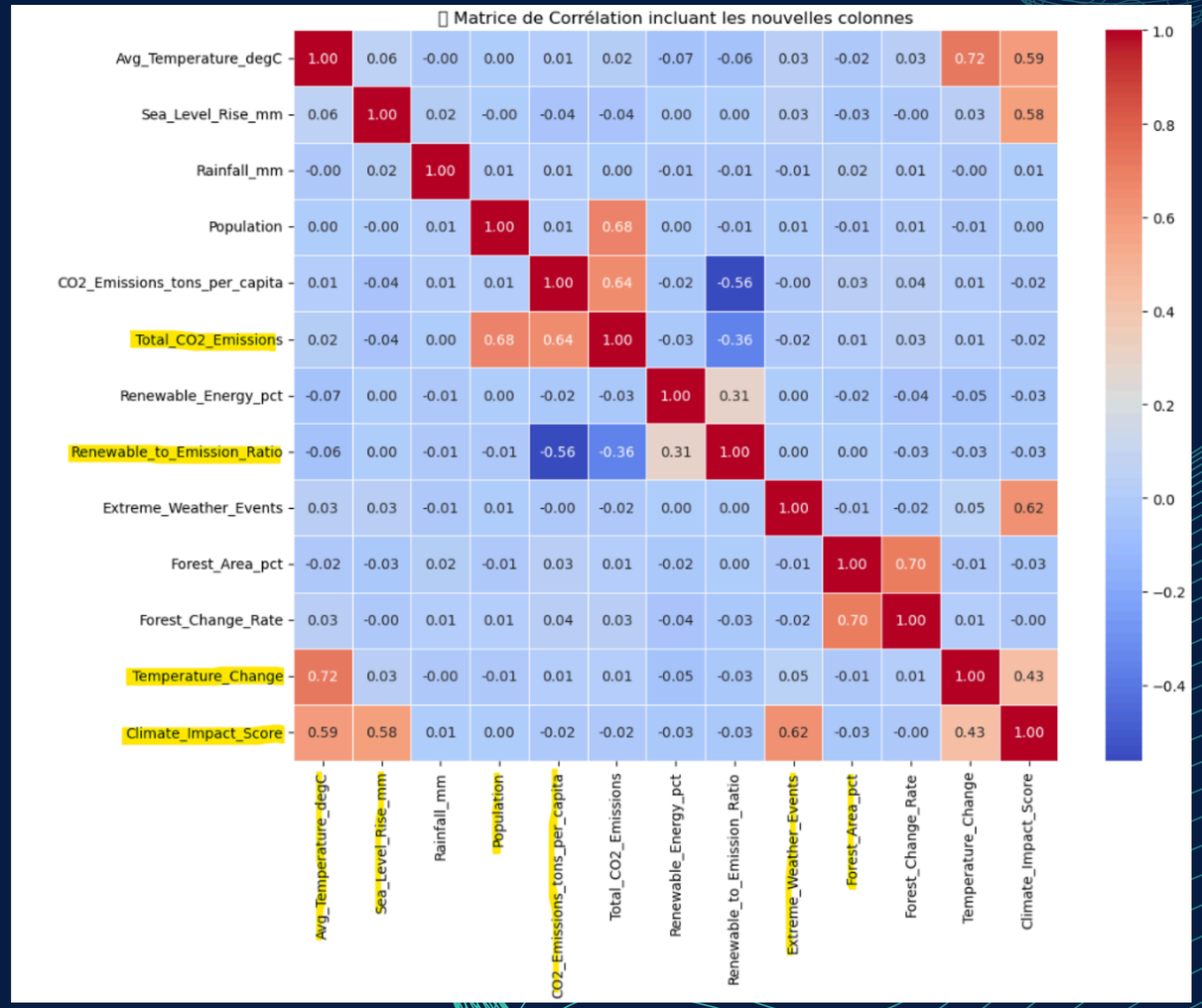
- droping rows with missing values + duplicates
- converting data types
- feature engineerings to extract deeper insights and create meaningful indicators

```python
# Décennie (utile pour visualiser par grande période)
df["Decade"] = (df["Year"] // 10) * 10

# Émissions totales de CO2 (pas juste par habitant)
df["Total_CO2_Emissions"] = df["CO2_Emissions_tons_per_capita"] * df["Population"]

# Ratio renouvelable/émissions - indicateur de transition verte
df["Renewable_to_Emission_Ratio"] = df["Renewable_Energy_pct"] / (df["CO2_Emissions_tons_per_capita"] + 1e-5)

# Catégorie de risque météo (basé sur le nombre d'événements extrêmes)
def classify_risk(x):
    if x >= 20:
        return "Élevé"
    elif x >= 10:
        return "Modéré"
    else:
        return "Faible"

df["Weather_Risk_Level"] = df["Extreme_Weather_Events"].apply(classify_risk)

# Tendance de déforestation ou pas (en comparant par pays + années)
df.sort_values(by=["Country", "Year"], inplace=True)
df["Forest_Change_Rate"] = df.groupby("Country")["Forest_Area_pct"].diff()

# Variation annuelle de température par pays
df["Temperature_Change"] = df.groupby("Country")["Avg_Temperature_degC"].diff()

# Score d'impact climatique composite (température + météo + niveau mer)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[["Norm_Temp", "Norm_Sea", "Norm_Events"]] = scaler.fit_transform(
    df[["Avg_Temperature_degC", "Sea_Level_Rise_mm", "Extreme_Weather_Events"]]
)
df["Climate_Impact_Score"] = df[["Norm_Temp", "Norm_Sea", "Norm_Events"]].mean(axis=1)
```

Global environment indicators

Data visualization

- created graphs using matplotlib & seaborn thanks to cleaned data
- first overview of analytics



Twitter airline sentiment

Data pre-processing

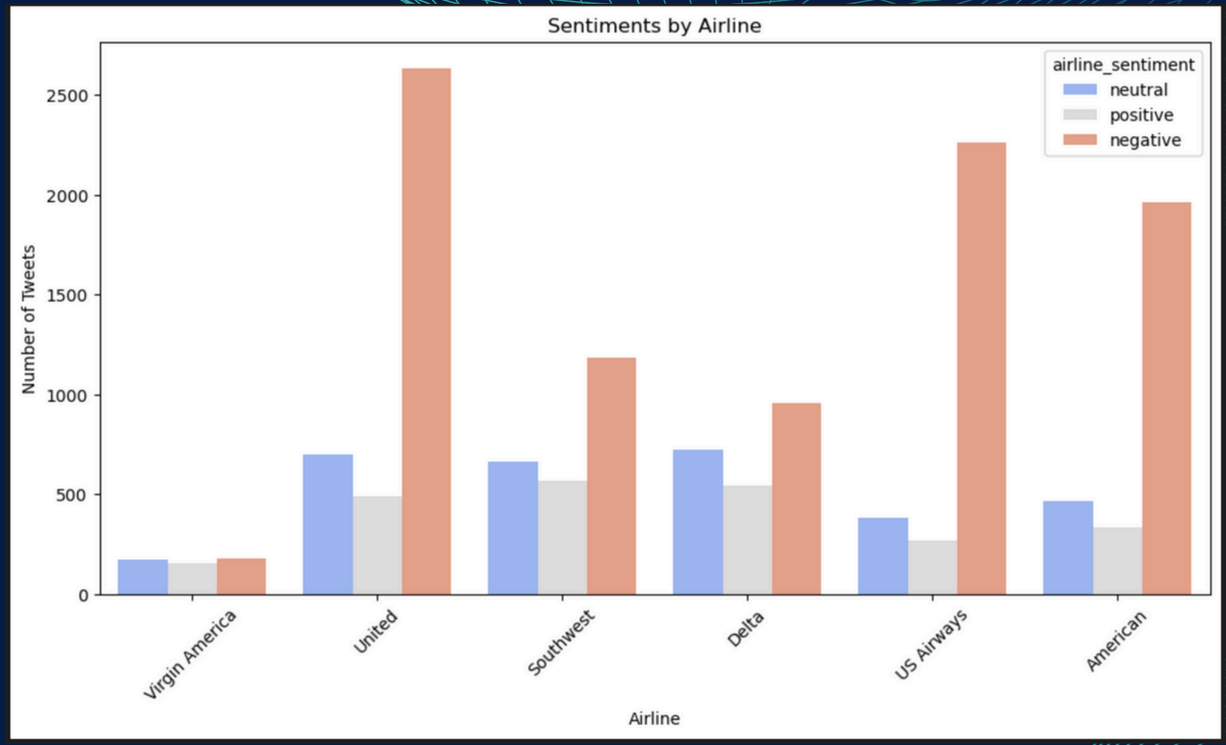
Cleaning

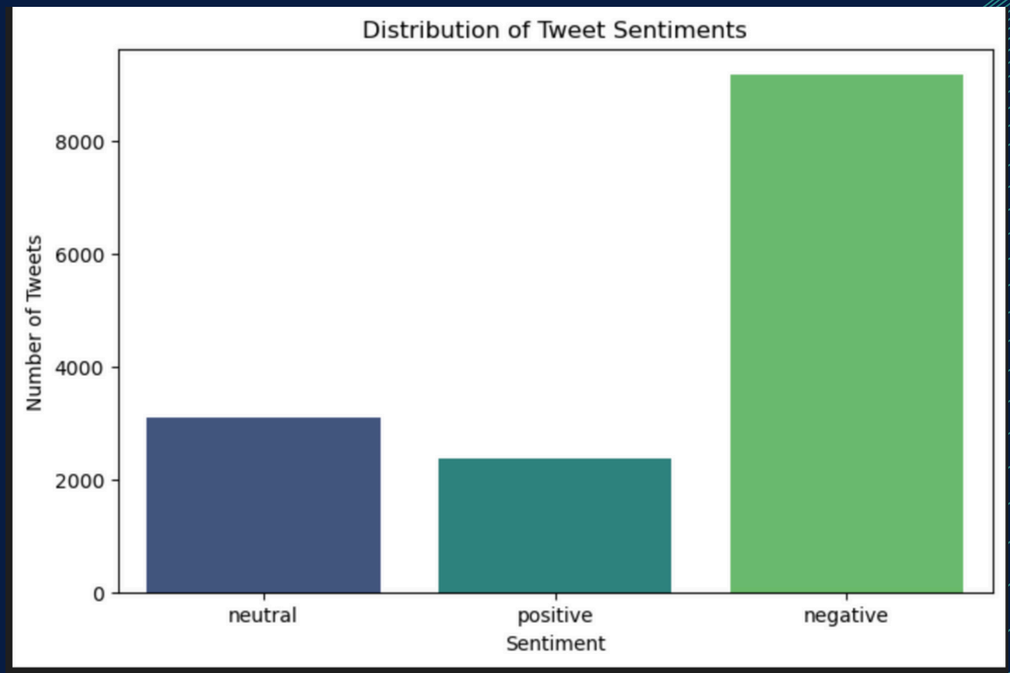
```
# Convert the date column to datetime format
df['tweet_created'] = pd.to_datetime(df['tweet_created'])

# Check for missing values
df.fillna({'negativereason': 'Unknown', 'tweet_location': 'Unknown', 'user_timezone': 'Unknown'}, inplace=True)
```

Twitter airline sentiment

Data visualization

```
# Visualization of sentiment distribution
plt.figure(figsize=(8,5))
sns.countplot(x='airline_sentiment', data=df, palette='viridis')
plt.title('Distribution of Tweet Sentiments')
plt.xlabel('Sentiment')
plt.ylabel('Number of Tweets')
plt.show()
```





Vehicle insurance data

Data pre-processing

- data conversion to allow better readability
- filling missing data (average, 0 or median)

```python
# Convertir les dates au format standard YYYY-MM-DD
df['INSR_BEGIN'] = pd.to_datetime(df['INSR_BEGIN'], format='%d-%b-%y', errors='coerce')
df['INSR_END'] = pd.to_datetime(df['INSR_END'], format='%d-%b-%y', errors='coerce')

# Convertir SEX en catégories lisibles
df['SEX'] = df['SEX'].map({0: 'Homme', 1: 'Femme'})

# Convertir EFFECTIVE_YR en entier
df['EFFECTIVE_YR'] = pd.to_numeric(df['EFFECTIVE_YR'], errors='coerce')

# Vérifier les modifications apportées
df.head()
```

```python
# Vérifier les valeurs manquantes par colonne
missing_values = df.isnull().sum()
missing_values[missing_values > 0]
```

	0
SEX	10649
EFFECTIVE_YR	276
PREMIUM	2
PROD_YEAR	32
SEATS_NUM	43
CARRYING_CAPACITY	47202
CLAIM_PAID	144737

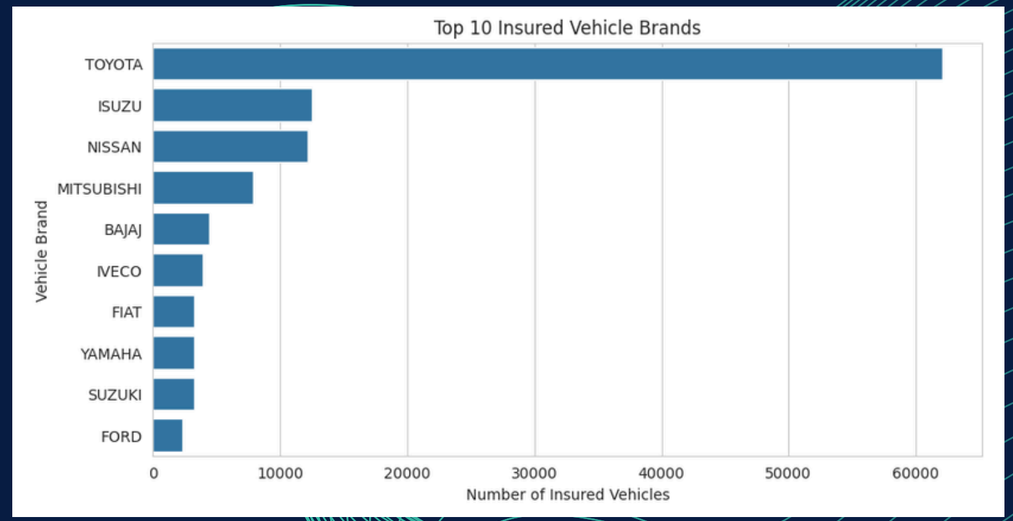
dtype: int64

Vehicle insurance data

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Configurer le style
sns.set_style("whitegrid")

# Répartition des assurés par sexe (Camembert)
plt.figure(figsize=(6, 6))
df['SEX'].value_counts().plot.pie(autopct='%1.1f%%', colors=['lightblue', 'pink'], labels=["Male", "Female"])
plt.title("Distribution of Policyholders by Gender")
plt.ylabel('')
plt.show()
```

Data visualization

- created graphs using matplotlib & seaborn thanks to cleaned data
- allowed for a first overview of analytics

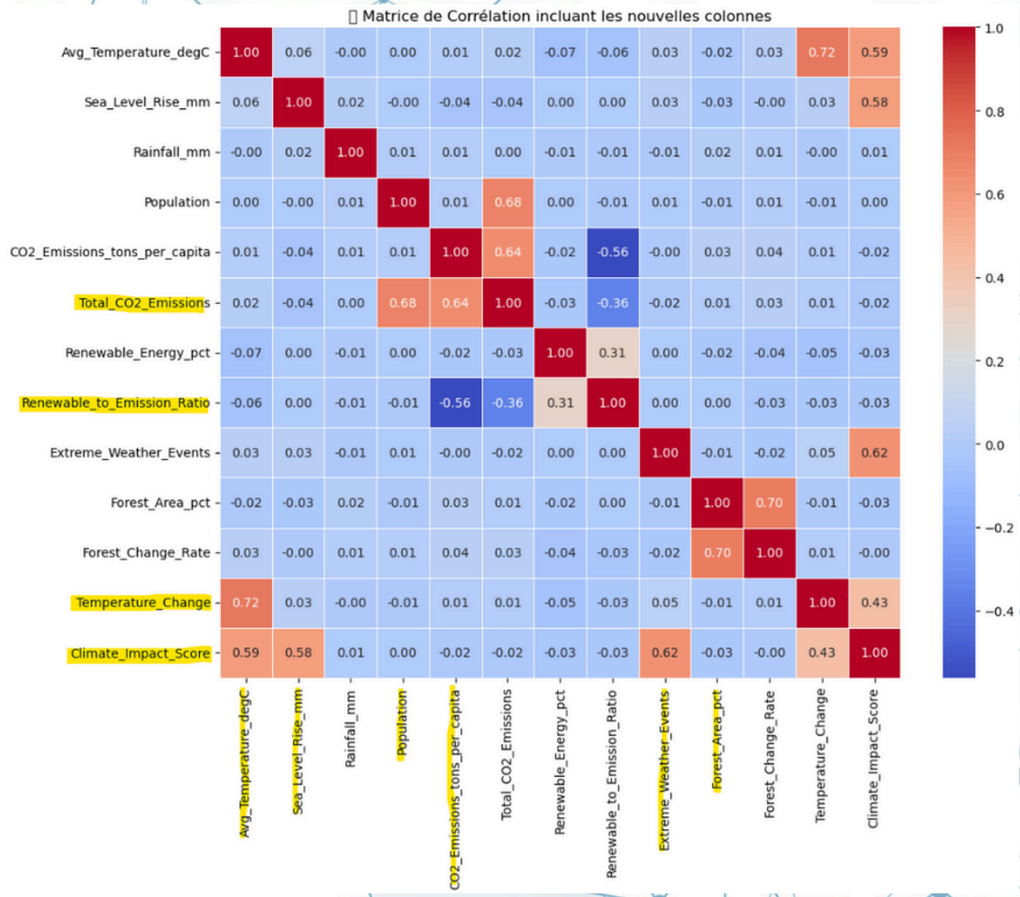


Phase 2 : Data Analysis and Specifications

Exploratory Data Analysis

Environmental analysis

Environmental Correlations:
Positive: CO2 emissions increase with population & industrialization.
Negative: Higher renewable energy adoption reduces emissions.

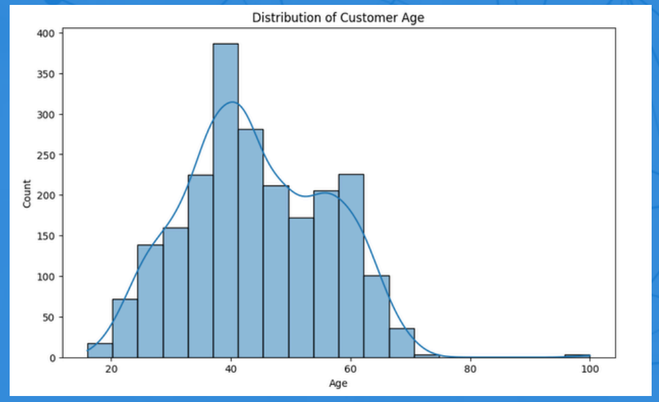


Exploratory Data Analysis

Customer behavior & insurance preferences

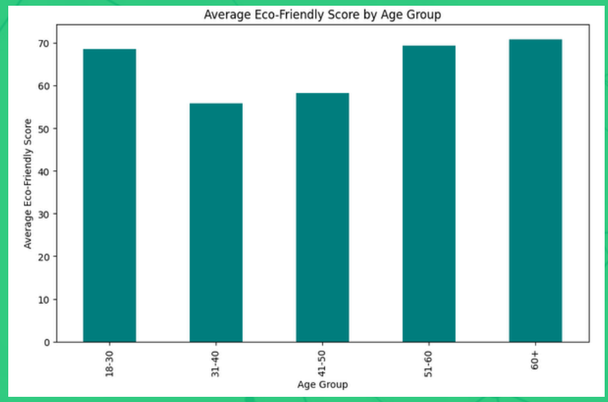
Age & insurance:

35-45 years old: highest likelihood of purchasing insurance.
28-34: Most interested in travel insurance.
18-20: More likely to have medical insurance.



Eco-friendly Behavior:

People aged 30-50 are less interested in eco-friendly scores.
Customers with 0 children are more eco-conscious.



Spending Patterns:

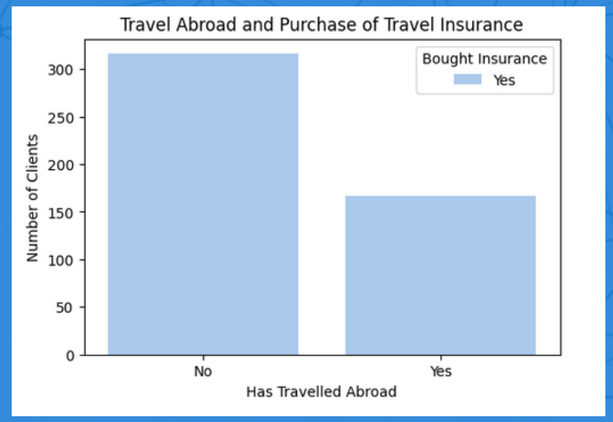
Widowed customers tend to spend more.
Smokers & those with higher BMI pay higher health premiums.

Exploratory Data Analysis

Travel & car insurance

Travel Insurance Trends:

People who have traveled abroad are more likely to buy travel insurance.

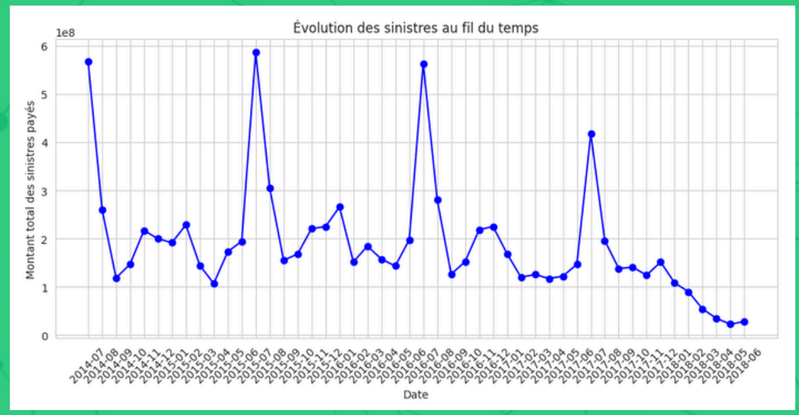


Car Insurance Trends:

Men are more likely to have car insurance.

Toyota owners are more likely to be insured.

Accidents peak during July & August (summer vacation).

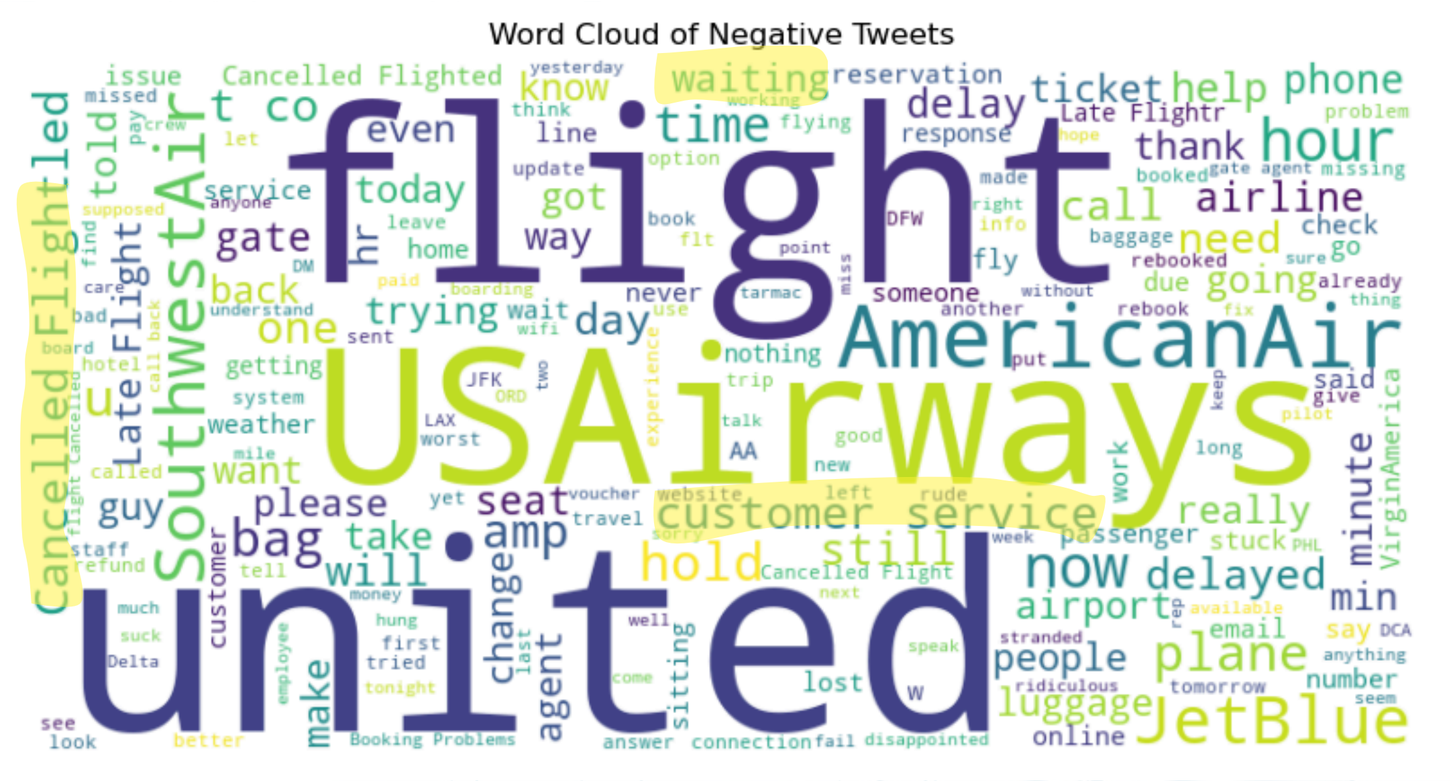


Exploratory Data Analysis

public sentiment

Public sentiment on airlines:

- United Airlines is the most disliked airline.
- Complaints mostly about canceled flights, delays, and customer service.



Key insights

Customer-Centric Policies:
Create specialized policies for key segments (e.g., eco-conscious customers, young travelers).

Data-Driven Decisions:
Implement dynamic pricing models based on risk factors.

Sustainability-Focused Insurance:
Offer discounts for sustainable behavior (e.g., electric cars, green homes).

Travel & Medical Insurance Enhancements:
More tailored packages for frequent travelers & young adults.

SPECIFICATIONS DOCUMENT

Project Goals :

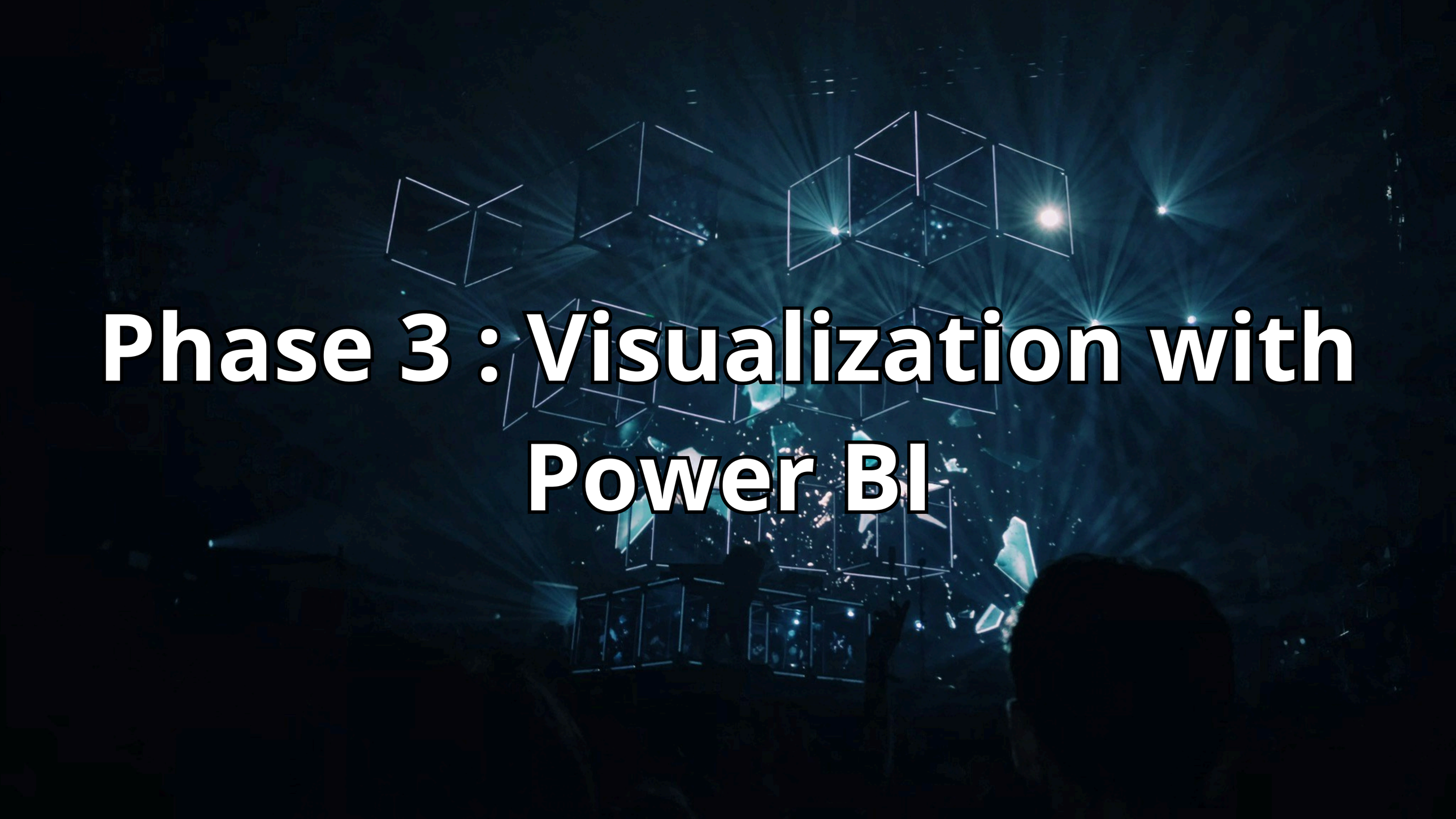
- Customer Behavior Analysis
- Sustainability Insights
- Policy Innovation
- Imporved Customer Satisfaction
- Data-Driven Decision Making

Functional Requirements :

- Data Sources
- Key Visualizations
- Interactivity Features

Dashboard Design Recommendations :

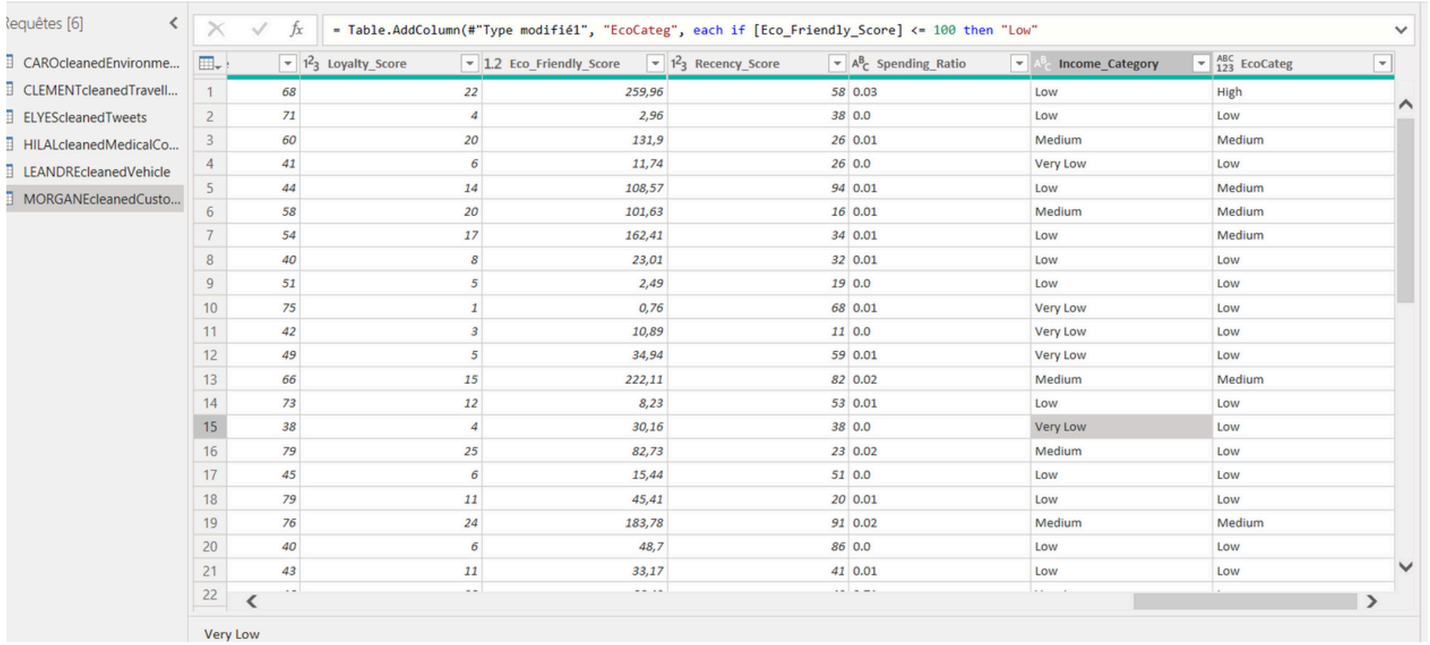
- Layout and Navigation
- Color Scheme and Visual Hierarchy
- Critical Metric Positioning

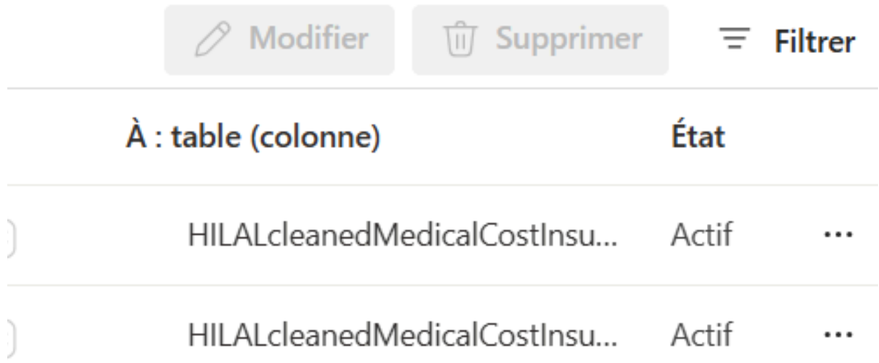

Phase 3 : Visualization with
Power BI

Power BI datasets transformation

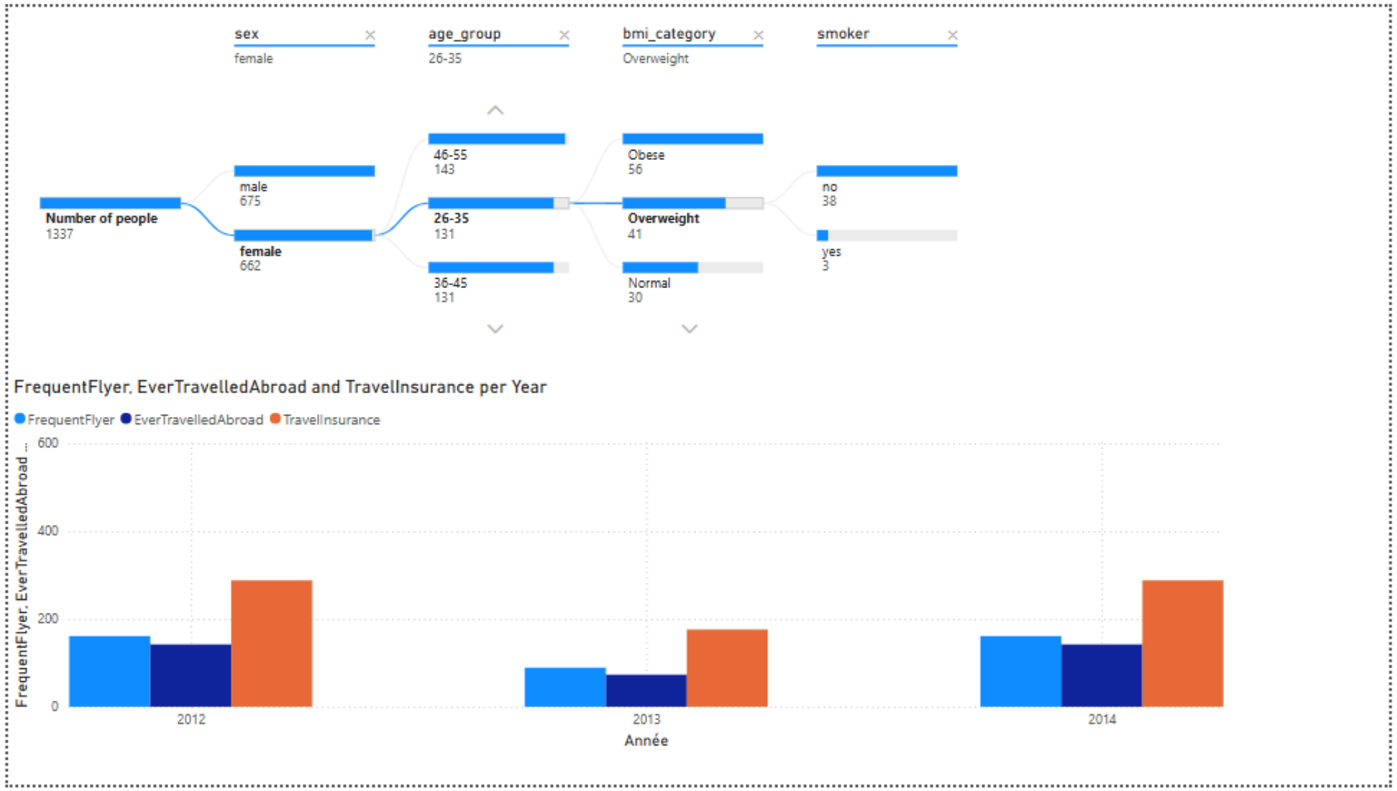
Aim : make final adjustments to the datasets that weren't covered during data cleaning in Python, ensuring it's fully prepared for analysis and visualization

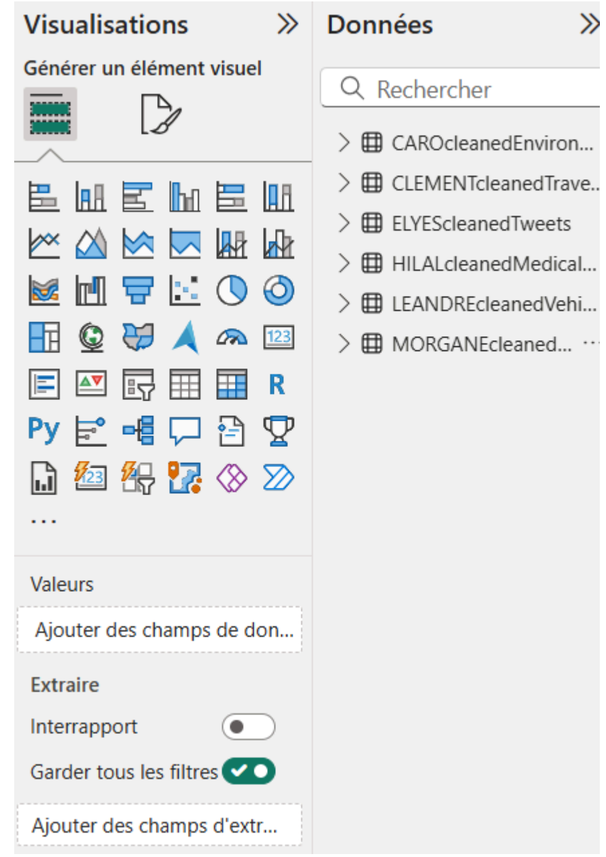
E.g : add a final column for better analysis or link several datasets based on the same attribute

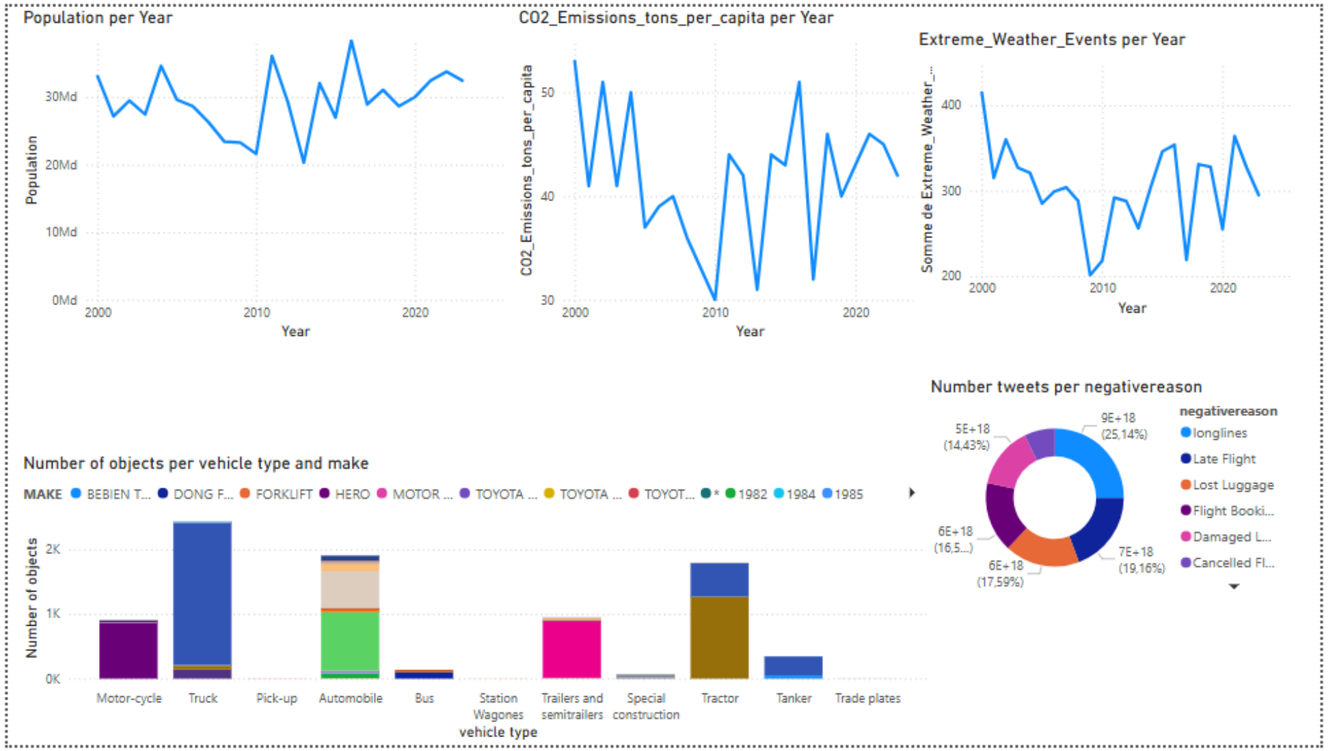


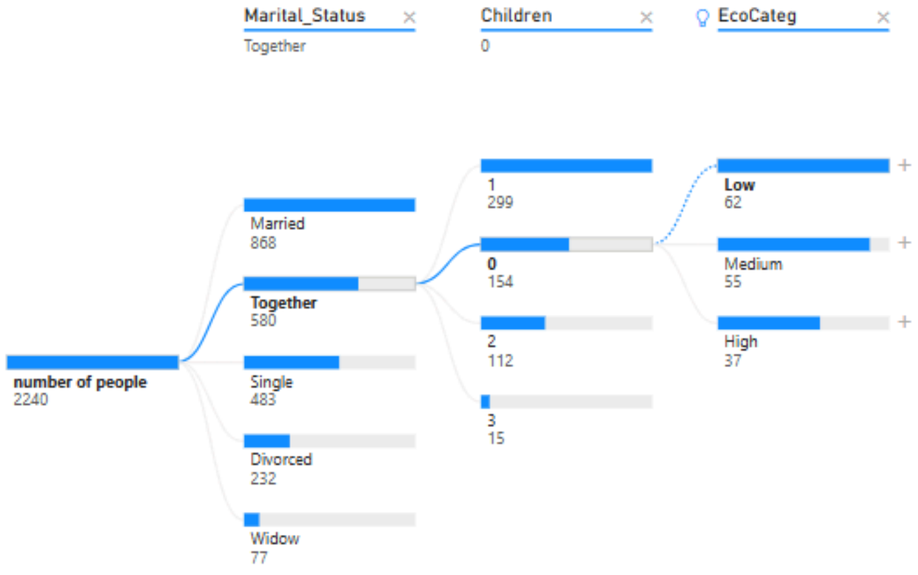


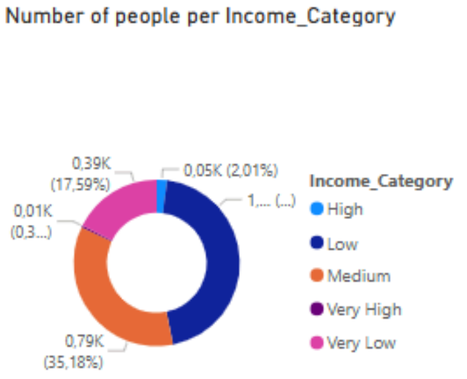
Power BI Visualization and Dashboard






Population per Year
Population
30Md
20Md
10Md
0Md
2000
2010
2020
Year
CO2_Emissions_tons_per_capita per Year
CO2_Emissions_tons_per_capita
50
40
30
2000
2010
2020
Year
Extreme_Weather_Events per Year
Somme de Extreme_Weather_...
400
300
200
2000
2010
2020
Year
Number of objects per vehicle type and make
MAKE
BEBIEN T...
DONG F...
FORKLIFT
HERO
MOTOR ...
TOYOTA ...
TOYOTA ...
TOYOT...
1982
1984
1985
Number of objects
2K
1K
0K
Motor-cycle
Truck
Pick-up
Automobile
Bus
Station Wagones
Trailers and semitrailers
Special construction
Tractor
Tanker
Trade plates
vehicle type
Number tweets per negativereason
9E+18 (25,14%)
5E+18 (14,43%)
6E+18 (16,5...)
6E+18 (17,59%)
7E+18 (19,16%)
negativereason
longlines
Late Flight
Lost Luggage
Flight Booki...
Damaged L...
Cancelled Fl...


Marital_Status
Together
Children
0
EcoCateg
number of people
2240
Married
868
Together
580
Single
483
Divorced
232
Widow
77
1
299
0
154
2
112
3
15
Low
62
Medium
55
High
37


Number of people per Income_Category
0,39K (17,59%)
0,05K (2,01%)
1,... (...)
0,01K (0,3...)
0,79K (35,18%)
Income_Category
High
Low
Medium
Very High
Very Low