Data Analytics

Charging Ahead: France's Electric Vehicle Revolution

Hilda Monterrubio

November 17th, 2023

Table of content

Introduction

Electric vehicles (EVs) are becoming increasingly important as more people look for ways to help the environment and cut down on pollution. They're leading the change by providing a cleaner way to travel that doesn't rely on more polluting energy like gasoline. **EVs are here to stay**; they're a key element in moving us towards a future where we can breathe easier and the planet is healthier. Additionally, with technological advancements, EVs are becoming more accessible and efficient, offering longer ranges and shorter charging times.

In France, the EV market is really taking off. The French government is encouraging people to buy electric cars by offering incentives such as cash-back and tax breaks, and many people are starting to switch from traditional cars to EVs. This growth is also seen in businesses and government agencies adding more EVs to their fleets, showing that the trend in France is leaning towards electric.

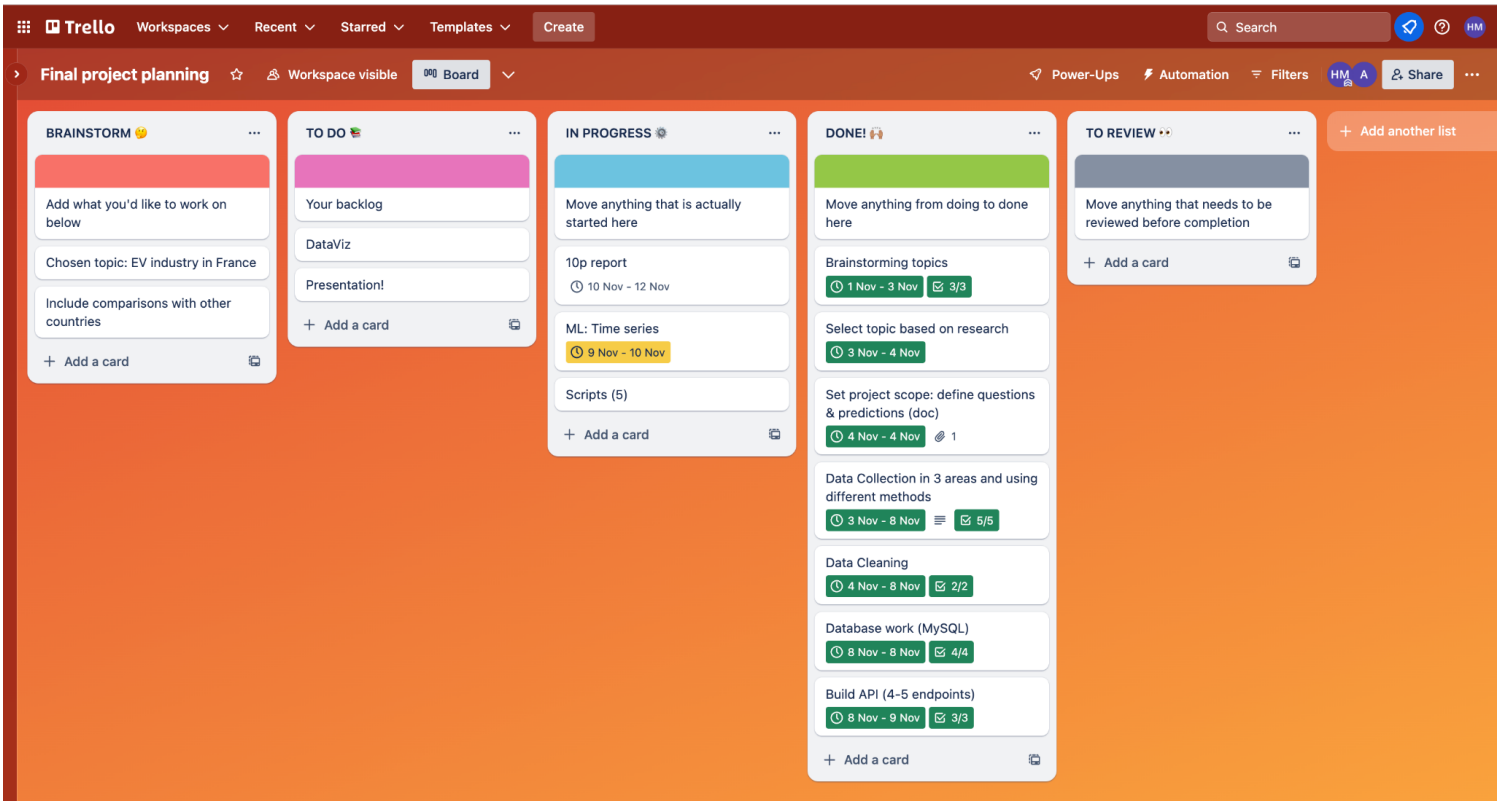
The main goal of this project is to look into two big questions about EVs in France. First, we want to understand how French consumers are reacting to the push for more EVs and what new buying patterns are showing. We'll look at how many people are buying EVs, if companies and governments are using more of them, and how government initiatives are helping this along. Second, we'll see if France's infrastructure is getting ready for more EVs by enabling the necessary amount of charging points. This will give us a clearer picture of how ready France is for more electric cars on the roads.

Plan (high-level):

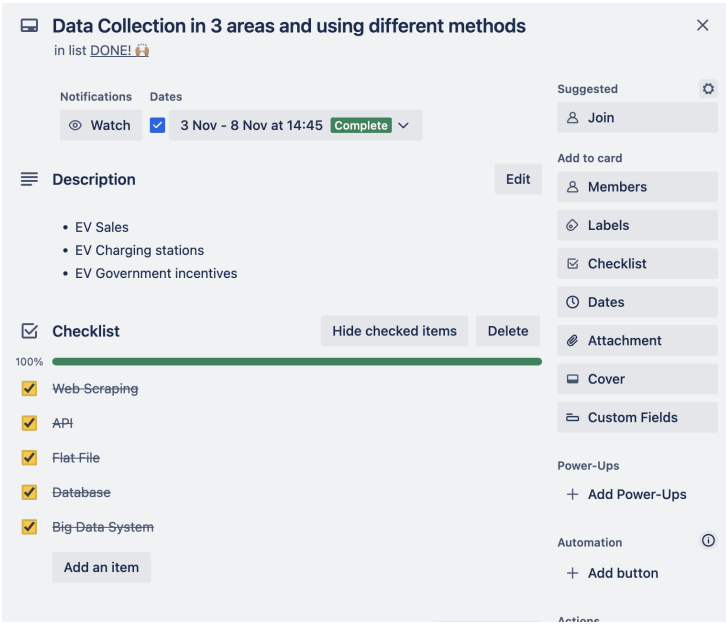
- Brainstorming and research
- Topic selection
- Scope definition
- End-to-end project planning using Trello
- Data collection using these following sources (Web Scraping, API, Flat File, Database & Big Data System)
- Exploratory data analysis in Python (data wrangling, data cleaning & data visualization)
- Selection and creation of a database using MySQL
- Added data to database and designed Entity Relationship Diagram
- Explored data with MySQL
- Exposed data via API
- Data visualization of main insights using Tableau
- Process data for time series analysis
- Train and test models

Project Management

Kanban board to manage daily project tasks (Trello)



The majority of the 'cards' in the Kanban board had a checklist item to track specific sub-tasks to be completed (see image). Once all of the tasks in the checklist have been completed, the card was updated and moved to 'Done'.



Data and data sources

Before diving into the details of data collection methods and sources, it's important to note that the topic has been segmented into three distinct categories: EV sales in France (which encompasses registration and fleet data), EV charging points in France, and governmental initiatives related to electric vehicles.

After defining those categories, I moved on to data collection, which was carried out through five distinct methods.

1. Web Scraping

French websites Capital auto and Automobile propre were scrapped in compliance with their respective Terms of Service. The scraping was intended to gather the 'Top 10 EV selling vehicles' in France for years 2019, 2020, 2021 and 2022.

As a result, five DataFrames were generated. Following data cleaning and manipulation in Python, these were combined into a single DataFrame and then exported as CSV data into MySQL. The table '***sales_top_selling_models***' was formed with the columns illustrated in the image below.

sales_top_selling_models	
model_id	unique identifier for each vehicle model
modele	vehicle model name
ventes	units sold
annee	year to which the sales data corresponds

2. API

I accessed France's charging points data, categorized by type and department, through ENEDIS's public API, which oversees the majority of the national electricity distribution network. The most demanding and time-consuming part of this task involved finding the exact datasets I needed, given that ENEDIS provides access to thousands of datasets.

The API documentation was straightforward and didn't need a lot of parameter tweaking. Two DataFrames were generated out of the information pulled from the API, and after data cleaning and wrangling in Python were separately exported as CSV data into MySQL. The Tables '***charging_points_per_type***' and '***charging_points_per_department***' were formed with the columns illustrated below.

charging_points_per_type	
id	unique identifier for each record
annee	year associated with the data
points_de_charge_par_type	charging points accessibility type (public, private or corporate)
valeur	number of charging points for the corresponding type

charging_points_per_departement	
id	unique identifier for each record
departement	name of the French department
code_departement	admin code assigned to the department
nombre_pc_au_public_100_000	number of public charging points per 100,000 habitants
evolution_sur_12_mois	evolution or change over the past 12 months
code_region	admin code for the larger regional area

3. Flat File

Two sources of data were used in the form of flat files in CSV format.

1. The International Energy Agency (IEA) has an official website at "https://www.iea.org/". This site offers extensive resources on various energy-related subjects, including electric vehicles (EVs). By utilizing the Global EV Data Explorer tool available on their site, I gathered data on global EV sales and the number of EV charging points, including specific figures for France.

   From this source, two DataFrames were generated. Following data cleaning and manipulation in Python, these were separately exported into MySQL as CSV data. The tables '***sales_ev_world***" and '***charging_points_world***' were formed with the columns illustrated below.

charging_points_world	
id	unique identifier for each record
region	country name
category	classification of data (historical)

parameter	variable being measured (charging points)
mode	type of EV charging points
powertrain	charging speed (fast/slow)
year	year data was recorded
unit	unit of measurement
value	number of charging points

sales_ev_world	
country_id	unique identifier for each record
region	country name
category	classification of data (historical)
parameter	variable being measured (ev sales)
mode	transportation type (vehicles)
powertrain	type of electric vehicle
year	year data was recorded
unit	unit of measurement
value	number of ev vehicles sold

2. The website "https://www.senat.fr/", which is the official site of the French Senate. This website provides a vast amount of information on many public-related topics so I had to be very specific on the type of data I needed to collect. My research led me to find a couple of files that contained the amount of money/budget the French government has allocated to EV-related initiatives over the past six years.

Although the information was spread across different files, only one single DataFrame was generated from it, and after data cleaning and wrangling in Python it was exported as CSV data into MySQL. The table '***gov_ev_initiatives_budget***' was formed with the columns illustrated below.

gov_ev_initiatives_budget	
id	unique identifier for each record

gov_initiative	name of government initiative
annee	year data was recorded
valeur	budget associated with the initiative (millions)

4. Database

I consulted the "Insee" database, which is the official site for the National Institute of Statistics and Economic Studies in France. From this source I was able to retrieve all the information related to vehicle registrations in France, more specifically, number of EV registrations per month from 2011-2021, and number of registrations by energy type for the same period.

Based on this information, two DataFrames were generated. Following data cleaning and manipulation in Python, these were separately exported into MySQL as CSV data. The tables '***immatriculations_voitures_ev'*** and '***immatriculations_voitures_ev_par_type***' were formed with the columns illustrated below.

immatriculations_voitures_ev	
id	unique identifier for each record
mois	month of the year for the data
annee	year of the year for the data
valeur	number of EV vehicles registered

immatriculations_voitures_ev_par_type	
id	unique identifier for each record
type_energie	type of energy or fuel used
annee	year data was recorded
valeur	value associated with each type of energy (thousands)

Next, I consulted the Données et études statistiques database from the Service des données et études statistiques (SDES), which is the statistical service for the French ministries responsible for transportation, energy, environment, climate, and sustainable development

Using this resource, I obtained comprehensive data on the total number of electric vehicles (EVs) currently on the roads in France, also known as the 'PARC en circulation' from 2011-2021, and per type of vehicle (private, corporate, public transport).

Based on this information, a single DataFrame was generated, and after data cleaning and wrangling in Python it was exported as CSV data into MySQL. Table '***parc_voitures_ev_par_type'*** was formed with the columns illustrated below.

parc_voitures_ev_par_type	
id	unique identifier for each record
type_energie	type of energy or fuel used (diesel, petrol, electric, etc)
type_voiture	category of vehicle ownership (public, private, corporate)
annee	year data was recorded
valeur	actual number of vehicles considered as part of the fleet in circulation*

Definition of fleet in circulation (PARC) per French law
A vehicle is considered to be in the in-use fleet if it meets the following conditions:
> the vehicle was registered in the vehicle registration system (SIV) before January 1 of the year
> the Agence Nationale des Titres Sécurisées (ANTS)has not been notified of any vehicle withdrawal.
> the vehicle is up to date with its roadworthiness test

5. Big Data System

As a final method for collecting data, I had to use a Big Data System in this project. While there are many big data systems on the market, such as Snowflake, Amazon Redshift, Oracle database, there is no doubt that Google BigQuery stands out due to its scalability, speed, ease of use, and cost effectiveness.. For these reasons, BigQuery was the chosen system.

Ironhack granted me access to BigQuery through a shared account. All I needed to do was log in using my Gmail address and begin exploring. As this account was set up for testing purposes, I depended on 'public datasets' to evaluate BigQuery's features. Unfortunately, I couldn't find any publicly available dataset that was relevant to the subject of my project.

To achieve the desired outcome, I employed a method that involved utilizing BigQuery's features to obtain the necessary data. This process included: accessing the database from Data.gouv.fr, locating a dataset with the required information (charging points in each French region), exporting this dataset as a CSV file, then conducting data cleaning and manipulation in Python.

Following this, I established a connection between Python and BigQuery and transferred the DataFrame to the appropriate location within BigQuery.

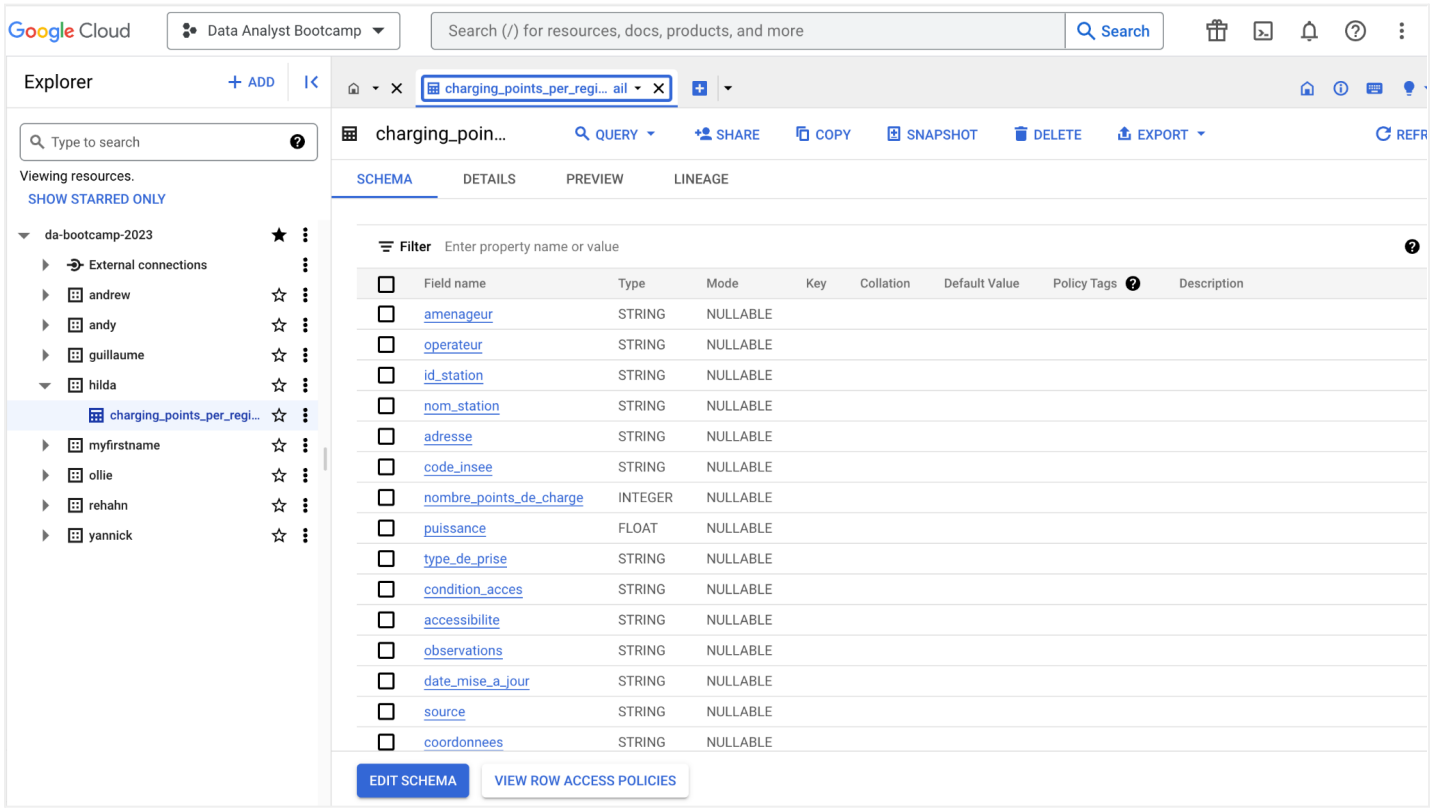
After completing the steps above, I was able to visualize my new table in BigQuery, named '***charging_points_per_region_detail***', and began running queries on it.

Line of code that was run to connect Python & BigQuery from my Jupyter notebook.

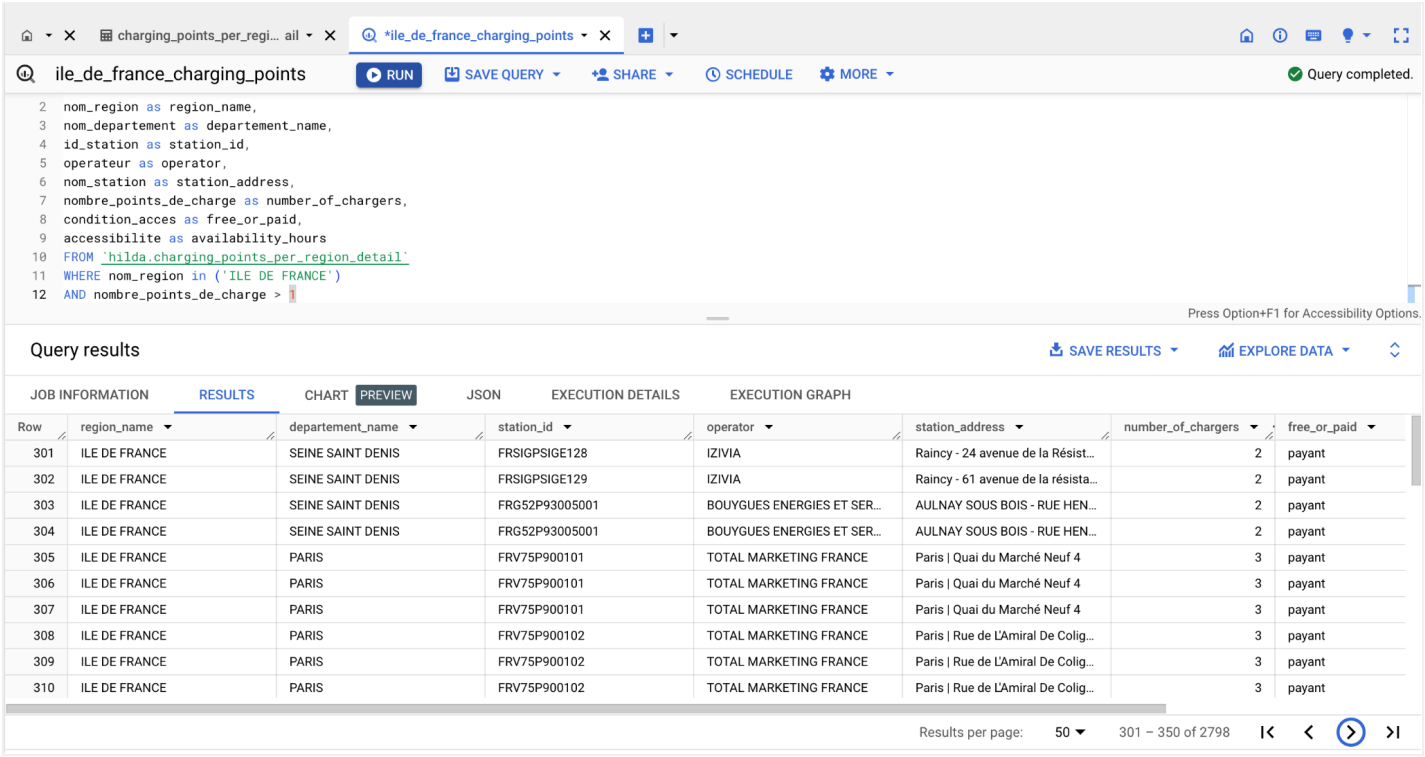
```
1 from google.cloud import bigquery
2
3 df.to_gbq("hilda.charging_points_per_region_detail", project_id='da-bootcamp-2023', if_exists='replace')
```

```
100%|██████████| 1/1 [00:00<00:00, 7244.05it/s]
```

View of newly created table in BigQuery



Example of a query to get the charging points in the 'Ile de France' region that have more than 1 charger available.



Data cleaning/wrangling

Data cleaning for this project was thorough and involved the application of a variety of techniques, such as:

- Handling duplicates
- Handling null values
- Handle date formats
- Filtering values
- Renaming columns
- Creating columns
- Dropping columns when necessary
- Dropping rows when necessary

- Verifying that the datatypes in the DataFrame were accurate, if not, handle conversion
- String formatting (strip, replace methods)
- Concatenation for joining multiple DataFrames
- Melting/unpivoting - to transform a wide format into a long format
- ASCII transliterations of unicode text
- Indexing

Datasets shape before and after cleaning

Dataset	***Before***		***After***	
	Columns	*Rows*	*Columns*	*Rows*
sales_top_selling_models	-scraping	-scraping	4	40
sales_ev_world	8	2,776	9	834
parc_voitures_ev_par_type	12	26	5	286
immatriculations _voitures_ev_par_type	12	6	4	66
immatriculations_voitures_ev	12	13	4	132
charging_points_world	8	604	9	604
charging_points_per_type	5	35	4	21
charging_points_per_departement	5	94	6	94
Charging_points_per_region_detail (BigQuery)	22	18,142	22	18,142
gov_ev_initiatives_budget	7	3	4	12

Exploratory data analysis

Exploratory Data Analysis (EDA) was conducted on the 9 DataFrames that were generated from the specified sources, following a similar set of steps for each.

a) Finding out the total number and rows using *.shape*

```
1 ev_sales_world.shape
```

```
(443, 9)
```

b) Having a look at the columns and their corresponding data types, along with finding whether they contain null values or not.

```
1 ev_sales_world.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 443 entries, 0 to 442
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   country_id  443 non-null    int64 
 1   region      443 non-null    object
 2   category    443 non-null    object
 3   parameter   443 non-null    object
 4   mode        443 non-null    object
 5   powertrain  443 non-null    object
 6   year        443 non-null    int64 
 7   unit        443 non-null    object
 8   value       443 non-null    int64 
dtypes: int64(3), object(6)
memory usage: 34.6+ KB
```

c) Using the describe() function in pandas to get various summary statistics.

```
1 ev_sales_world.describe()
```

	country_id	year	value
count	443.000000	443.00000	4.430000e+02
mean	416.925508	2016.30474	1.020588e+05
std	237.895888	3.61115	5.147943e+05
min	1.000000	2010.00000	3.000000e+00
25%	215.500000	2013.00000	3.750000e+02
50%	418.000000	2016.00000	3.000000e+03
75%	618.500000	2019.00000	2.350000e+04
max	834.000000	2022.00000	7.300000e+06

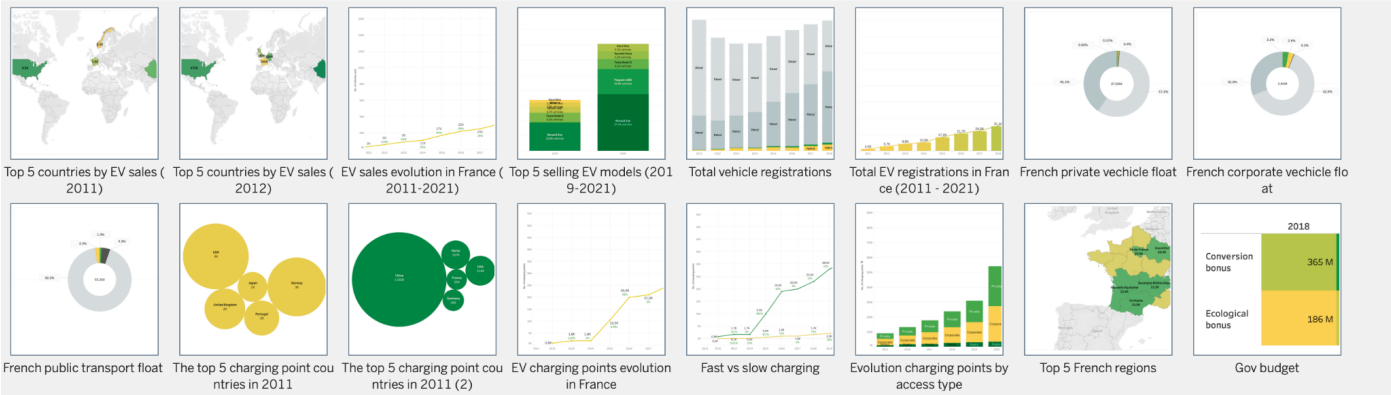
d) Finding the unique values of a certain column

```
1 ev_sales_world.region.unique()
```

```
array(['Australia', 'Austria', 'Belgium', 'Brazil', 'Canada', 'Chile',
       'China', 'Denmark', 'EU27', 'Europe', 'Finland', 'France',
       'Germany', 'Greece', 'Iceland', 'India', 'Israel', 'Italy',
       'Japan', 'Korea', 'Mexico', 'Netherlands', 'New Zealand', 'Norway',
       'Other Europe', 'Poland', 'Portugal', 'Rest of the world',
       'South Africa', 'Spain', 'Sweden', 'Switzerland', 'Turkiye',
       'United Kingdom', 'USA', 'World'], dtype=object)
```

1. Visualizing and analyzing historical data using Tableau. **Important note:** To ensure consistency between visualizations and overall analysis, I chose **2011 through 2021** as the range of available years for the vast majority of the charts.

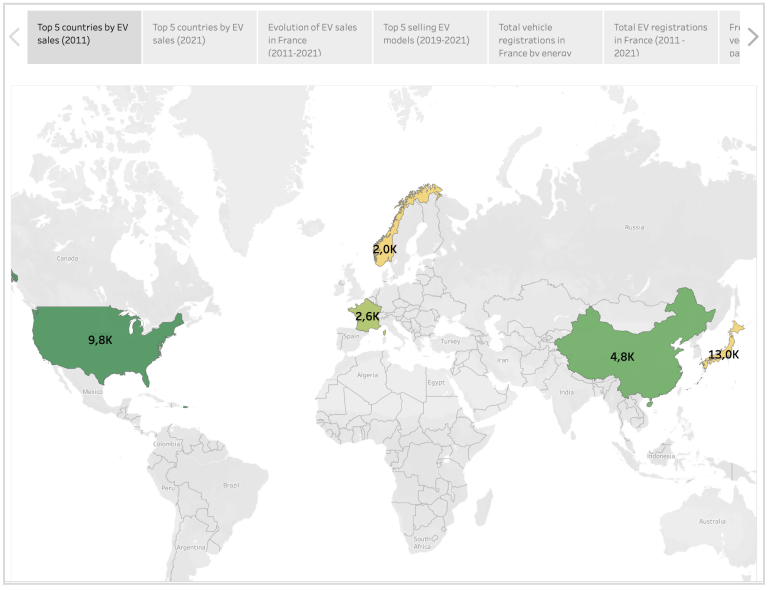
A total of sixteen visualizations were created from the ***'EV_in_France'*** database, which was directly integrated with Tableau using the available connector. Full story available here.

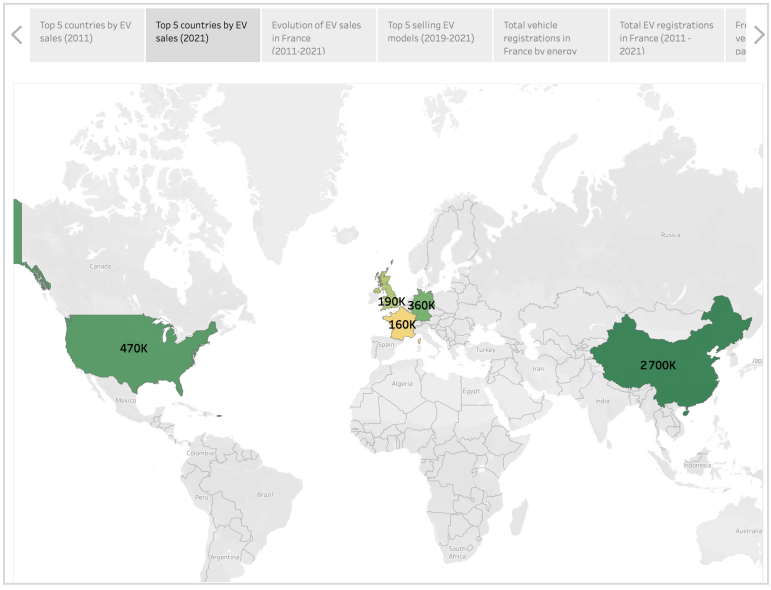


2. Visualizing and analyzing the results for the **Time Series** with the forecast for vehicle registrations in France for 2023 using plotly in Python (details can be found in the Time Series section of this document, page 23).

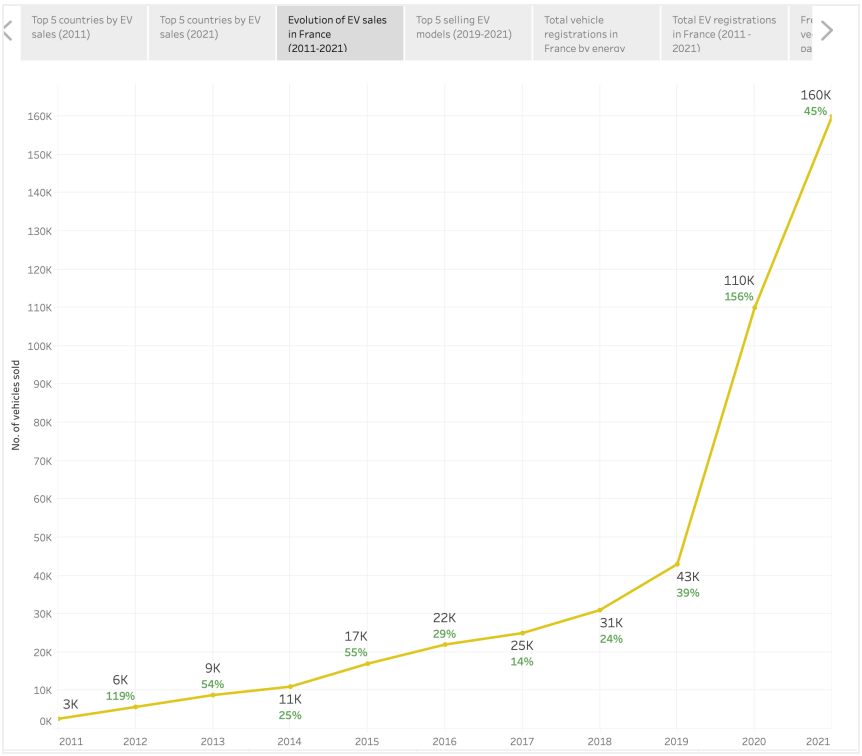
The following are examples of Tableau visualizations

1. Top 5 counties by EV sales. 2011 vs 2021 snapshot.





The maps above illustrate a global increase in electric vehicle (EV) sales over a decade, with particularly significant growth in certain countries. For instance, the United States saw an increase from 9.8K in 2011 to 470K in 2021. Similarly, China's EV sales skyrocketed from 4.8K to 2.7M during the same period, underscoring the massive expansion of the EV market there. France's growth over time really stands out as well.

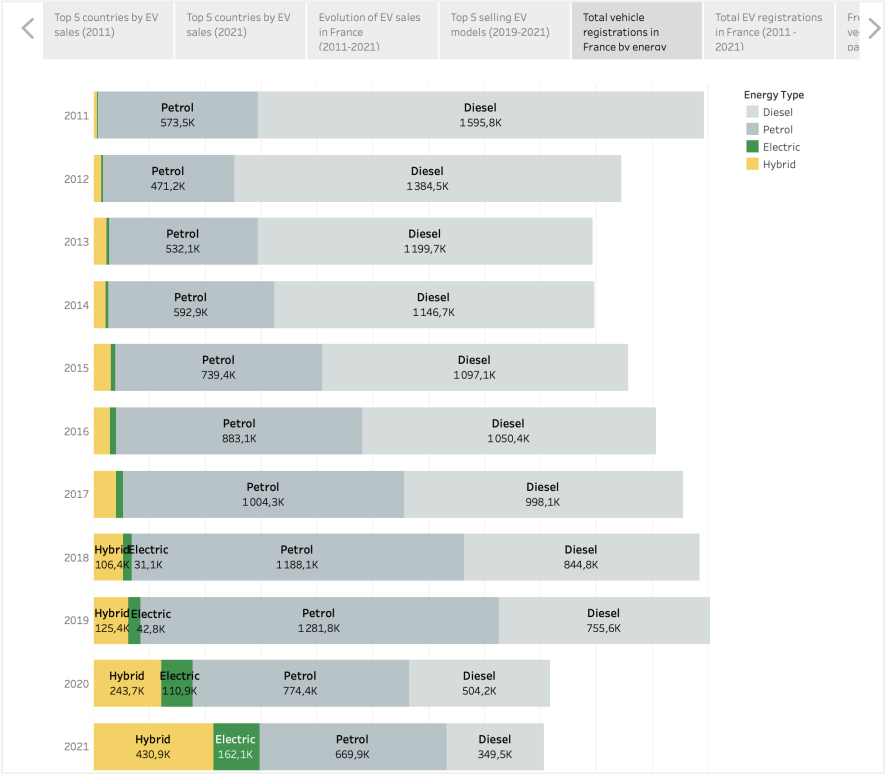


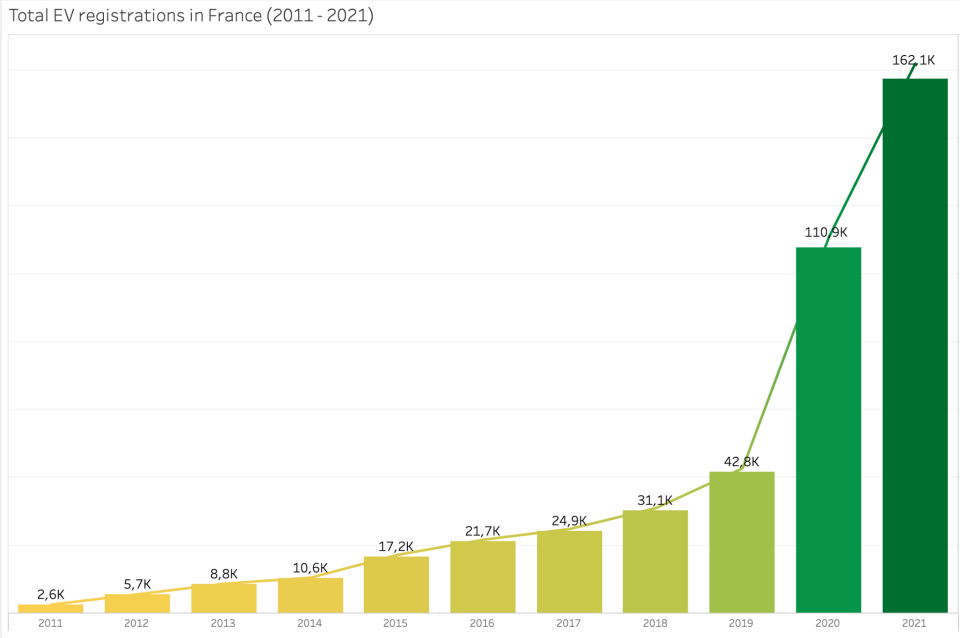
2. Evolution of EV sales in France (2011-2021)

There has been a significant increase in electric vehicle (EV) sales in France from 2011 to 2021, indicating a growing market and increased consumer adoption.Especially from 2019 to 2021, where the number of vehicles sold appears to have surged, with the percentage increase reaching 156% in the final year.

3.Total vehicle registrations in France by energy type (2011-2021)

This charts clearly shows a shift in vehicle energy types over the years in France. Diesel vehicles have seen a significant decline in numbers, while electric and hybrid vehicles have shown an increase, particularly noticeable from 2018 onwards. The data illustrates a growing trend towards more environmentally friendly vehicles, reflecting a change in consumer preferences.





4. Total EV registrations in France (2011 - 2021)

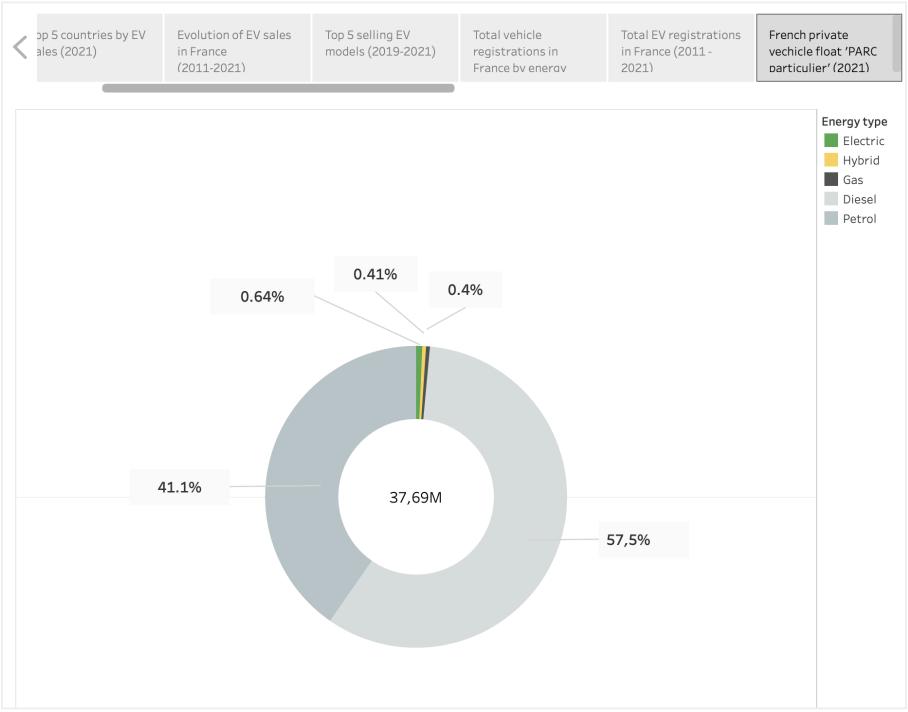
There has been an exponential growth in total electric vehicle (EV) registrations in France from 2011 to 2021, with a particularly sharp increase from 2019 to 2021. This surge suggests a rapidly accelerating shift towards EV adoption in the most recent years.

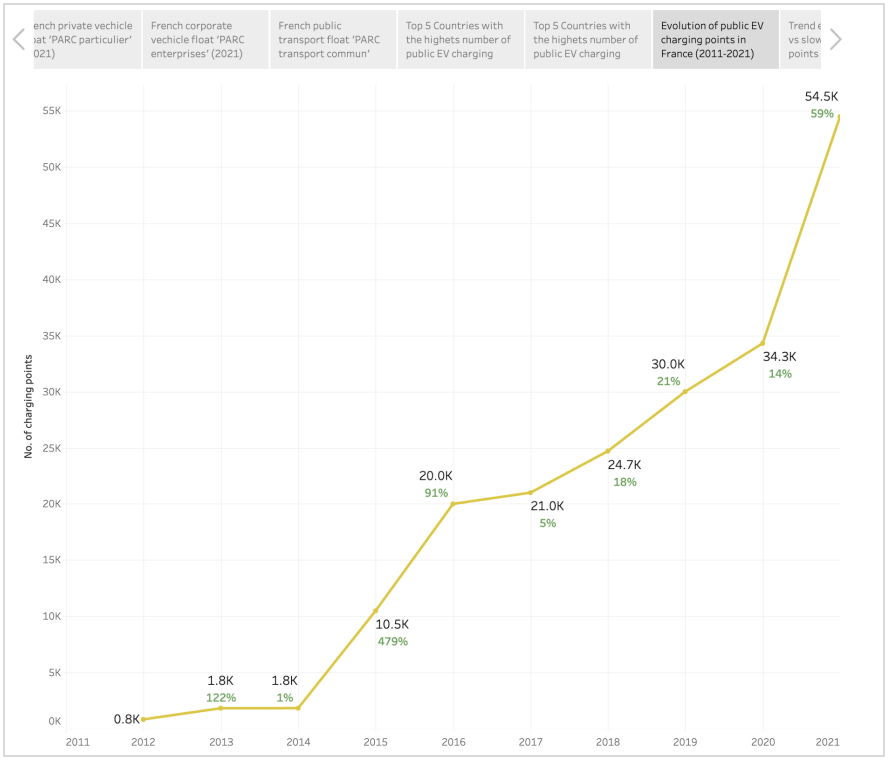
5. French private vehicle float 'PARC particulier (2021)

Of the 38.3 million passenger cars on the road in France in 2021, diesel (57.45%) and petrol (41.09%) engines still dominate.

One thing to keep in mind is that 'immatriculations/registrations' and 'PARC/float' are two different concepts in France. Registrations refers to the number of new electric vehicles that have been officially registered for use on public roads, while float refers to the actual number of electric vehicles that are operational on the roads.

Based on this chart, we can conclude that despite the growth of EVs, combustion engine vehicles, especially diesel, continue to be the type of vehicles that circulate the most on the roads. **Less than 1% of French cars run on all-electric power.**





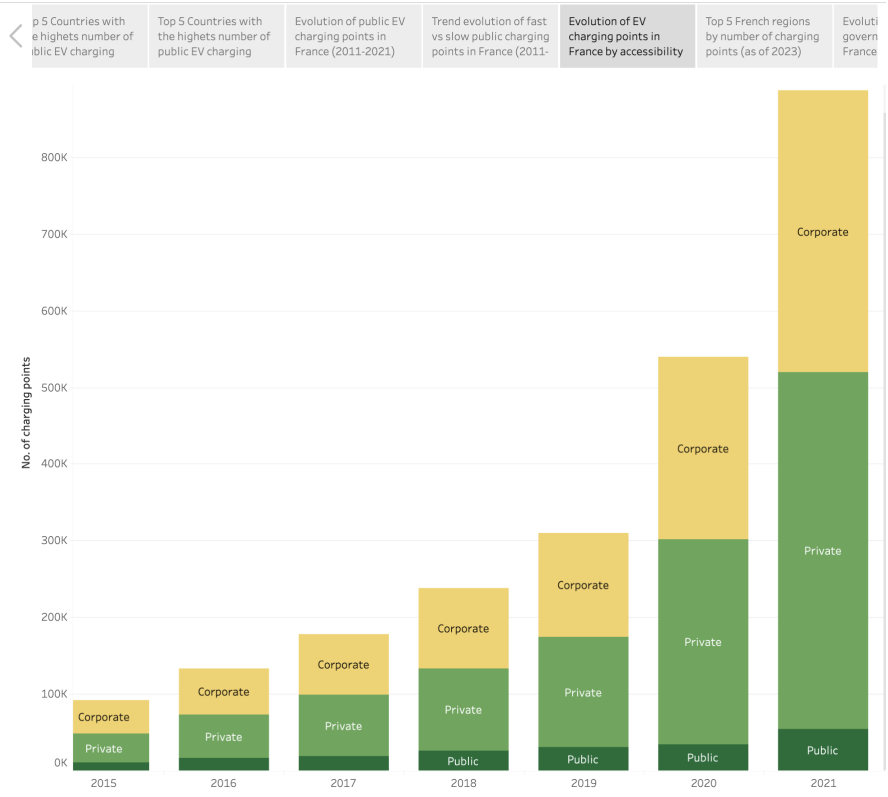
6. Evolution of public EV charging points in France (2011-2021)

There has been significant growth in the number of public electric vehicle (EV) charging points in France from 2011 to 2021. The data shows a dramatic increase, particularly from 2015 onwards, with the total number of charging points reaching approximately 54.5K by 2021. This indicates a strong development in EV infrastructure to support the rising number of electric vehicles.

7. Evolution of EV charging points in France by accessibility type (2015 - 2021)

From 2015 to 2021, France has seen a substantial rise in EV charging stations across private, corporate, and public sectors, with the most significant growth in public infrastructure. This trend suggests strong support from policies and investments to make EV charging more accessible.

Corporations are increasingly adopting EVs, and individuals are investing in home charging, indicating a broader commitment to sustainable transportation. France is actively fostering an EV-friendly ecosystem.



Database type selection

The world of databases is commonly divided into two main "buckets" or categories based on how they store and manage data:

1. **Relational Databases (SQL)**: These manage data by organizing it into tables, which are made up of rows and columns. Each table represents a different type of entity, and each row within a table represents a single instance of that entity. Columns represent the attributes of that entity, and each row-column intersection stores a single piece of data.

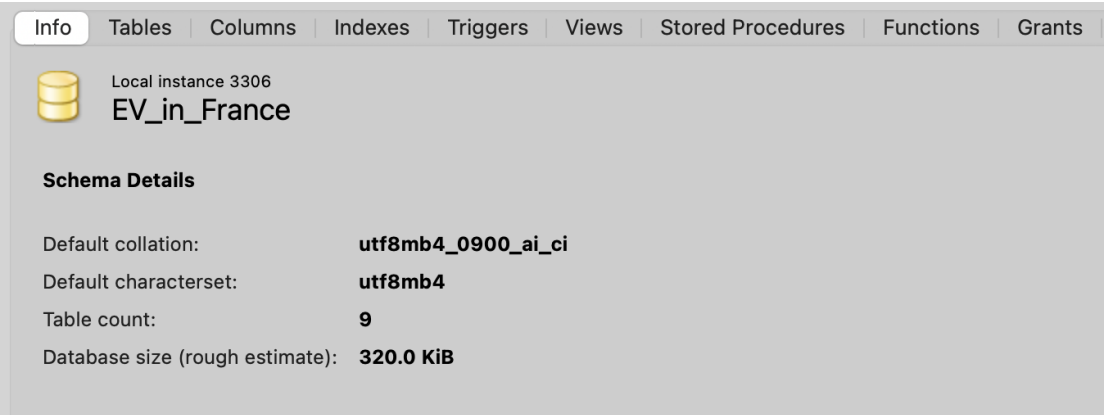
   The "relational" aspect comes from the ability to link tables through foreign keys, which are references from one table to another. Relational databases use Structured Query Language (SQL) to create, read, update, and delete data.

2. **Non-relational Databases (NoSQL)**: These store and manage data in ways that do not require the table-like schema of a relational database. They can handle unstructured, semi-structured, or structured data, and they do not strictly enforce a schema, allowing each entry to have its own unique structure. NoSQL databases use various querying languages that are typically not SQL-based.

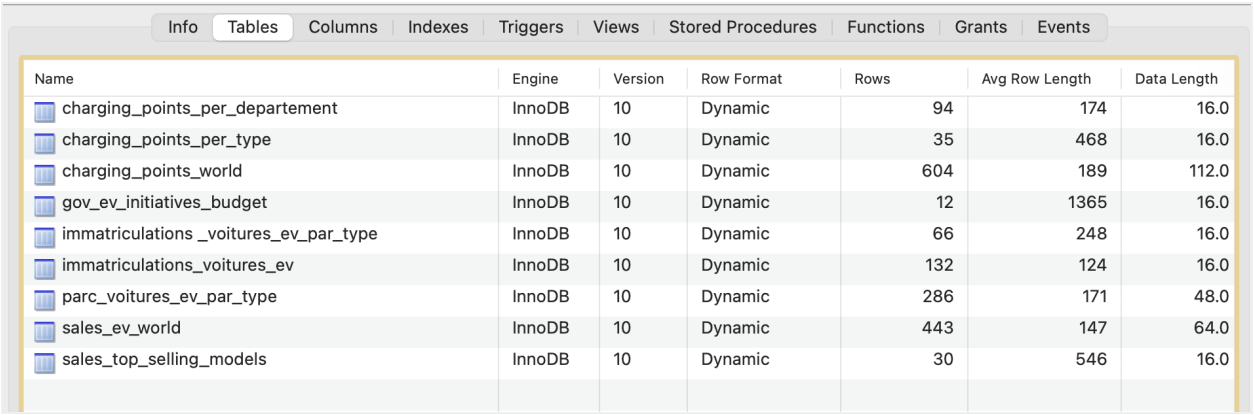
Since I have structured data organized into interrelated tables with a predefined schema, relationships, and foreign keys it seems appropriate to use a **Relational Database**. This type of database will help me to reduce data redundancy and improve data integrity, easily manipulate data, and of course, use SQL to perform complex queries that involve joining data from multiple tables.

Database creation

The '***EV_in_France***' database was created in MySQL Workbench to store 9 tables related to the EV industry in France.



Database tables:



Info | Tables | Columns | Indexes | Triggers | Views | Stored Procedures | Functions | Grants | Events

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length
charging_points_per_departement	InnoDB	10	Dynamic	94	174	16.0
charging_points_per_type	InnoDB	10	Dynamic	35	468	16.0
charging_points_world	InnoDB	10	Dynamic	604	189	112.0
gov_ev_initiatives_budget	InnoDB	10	Dynamic	12	1365	16.0
immatriculations _voitures_ev_par_type	InnoDB	10	Dynamic	66	248	16.0
immatriculations_voitures_ev	InnoDB	10	Dynamic	132	124	16.0
parc_voitures_ev_par_type	InnoDB	10	Dynamic	286	171	48.0
sales_ev_world	InnoDB	10	Dynamic	443	147	64.0
sales_top_selling_models	InnoDB	10	Dynamic	30	546	16.0

SQL Queries

Some examples of queries that were executed (all queries available in repository)

1. Total number of EVs sold and total number of charging points by country, regardless of year.

```sql
SELECT
c.region AS country,
SUM(s.value) AS total_ev_sales,
SUM(c.value) AS total_charging_points
FROM charging_points_world c
JOIN sales_ev_world s
ON c.region = s.region
GROUP BY c.region
ORDER BY total_ev_sales DESC;
```

2. Comparing the number of EV sales and charging points year over year for a specific country.

```sql
SELECT
s.region as country,
s.year,
s.value AS ev_sales,
c.value AS charging_points
FROM sales_ev_world s
JOIN charging_points_world c
ON s.region = c.region AND s.year = c.year
WHERE s.region = 'France'
ORDER BY s.year;
```

3. Total number of charging points per type (slow/fast charging) and per country.

```sql
SELECT
region AS country,
CASE
WHEN LOWER(TRIM(powertrain)) = 'publicly available fast' THEN 'Fast Charging'
WHEN LOWER(TRIM(powertrain)) = 'publicly available slow' THEN 'Slow Charging'
ELSE TRIM(powertrain)
END AS power_train_type,
SUM(value) AS total_charging_points
FROM charging_points_world
GROUP BY
country,
power_train_type
ORDER BY
country, power_train_type;
```

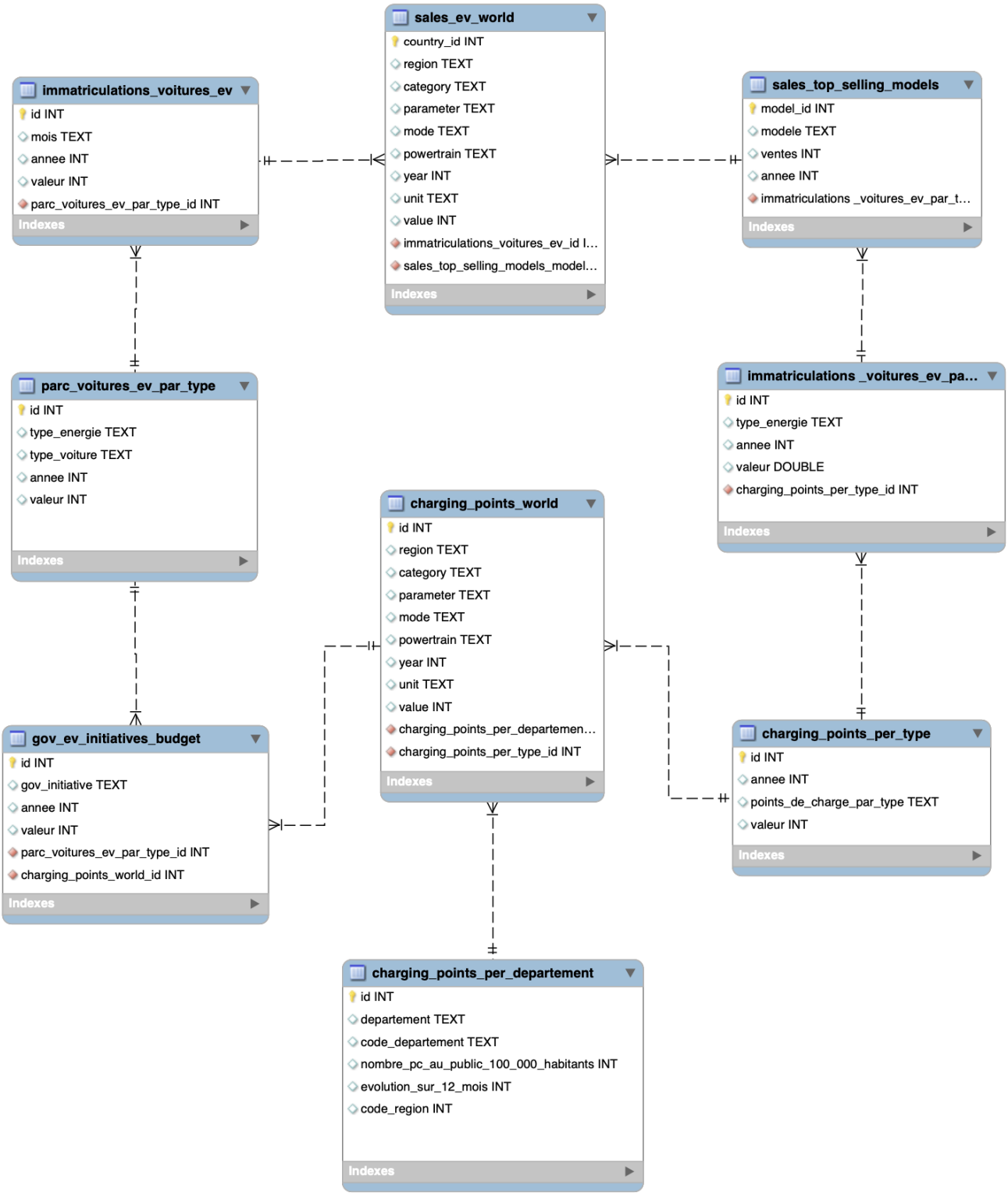
4. Top 5 vehicle models in France (based on units sold, years 2019-2022).

```sql
SELECT modele as vehicle_model,
SUM(ventes) AS units_sold,
'2019-2022' AS data_range
FROM sales_top_selling_models
GROUP BY modele
ORDER BY units_sold DESC
LIMIT 5;
```

5. Total charging points in France by type (public/private/corporate) for years 2015-2023.

```sql
SELECT
annee AS year,
SUM(accessible_au_public) AS total_public_charging_points,
SUM(particulier) AS total_private_charging_points,
SUM(societe) AS total_business_charging_points
FROM charging_points_per_type
GROUP BY annee
ORDER BY annee;
```

Entity Relationship Diagram (ERD)



Exposing Data via API

An API was developed to make a portion of the available data accessible for various applications. The documentation for this API was created directly in the Python file using HTML.

How it looks in code

```
@app.route("/")
def greetings():
    return """<html>
    <h1 style="color:olive;"> Hello, Welcome to the Charging Ahead EV API documentation!🛢️⚡ </h1>
    <p> To use it, please read the information below: </p>
    <h2 style="color:teal;"> EV Sales  </h2>
    <p> To find the information related to EV Sales Worldwive, try the <b>/evsales</b> endpoint. At the <b>bottom</b> of each page, you'll
find the links to access the next and last pages </p>
    <p> To find the information related to EV Sales in France or any other country, try the <b>/evsales/{country}</b> endpoint. </p>
    <h2 style="color:teal;"> Charging Points </h2>
    <p> To find the information related to Charging Points Worldwive, try the <b>/chargingpoints</b> endpoint. At the <b>bottom</b> of each
page, you'll find the links to access the next and last pages </p>
    <p> To find the information related to Charging Points in France or any other country, try the <b>/chargingpoints/{country}</b>
endpoint. </p>
    <p> To filter Charging Points in France or any other country by power train type (fast/slow) try the <b>/chargingpoints/{country}?
powertrain={fast_charging/slow_charging}</b> endpoint. </p>
    <p> </p>
    <h2 style="color:teal;"> Top Selling Vehicles </h2>
    <p> To find the information related to the top selling vehicles in France from (2019 - 2022), try the <b>/topvehiclesfrance</b>
endpoint. At the <b>top</b> of each page, you'll find the links to access the next and last pages </p>
</br>
Happy searching!😀

    </html>"""
```

How it looks when page is loaded

Hello, Welcome to the Charging Ahead EV API documentation!🛢️⚡

To use it, please read the information below:

EV Sales

To find the information related to EV Sales Worldwive, try the **/evsales** endpoint. At the **bottom** of each page, you'll find the links to access the next and last pages

To find the information related to EV Sales in France or any other country, try the **/evsales/{country}** endpoint.

Charging Points

To find the information related to Charging Points Worldwive, try the **/chargingpoints** endpoint. At the **bottom** of each page, you'll find the links to access the next and last pages

To find the information related to Charging Points in France or any other country, try the **/chargingpoints/{country}** endpoint.

To filter Charging Points in France or any other country by power train type (fast/slow) try the **/chargingpoints/{country}?powertrain={fast_charging/slow_charging}** endpoint.

Top Selling Vehicles

To find the information related to the top selling vehicles in France from (2019 - 2022), try the **/topvehiclesfrance** endpoint. At the **top** of each page, you'll find the links to access the next and last pages

Happy searching!😀

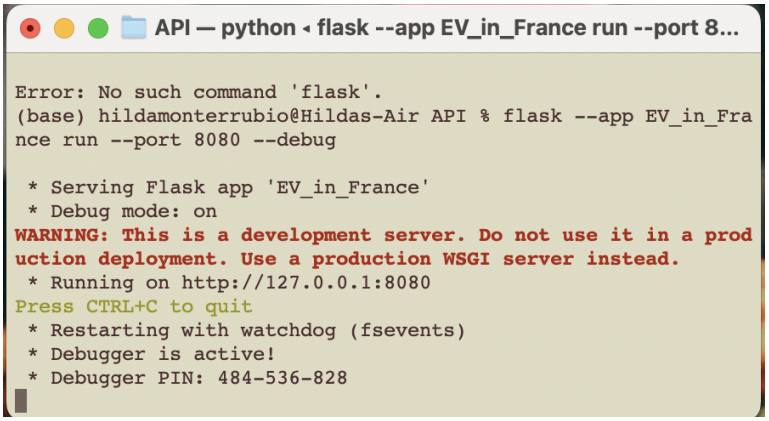
API endpoints:

As you can see from the API documentation, a total of six endpoints were created for different use cases:

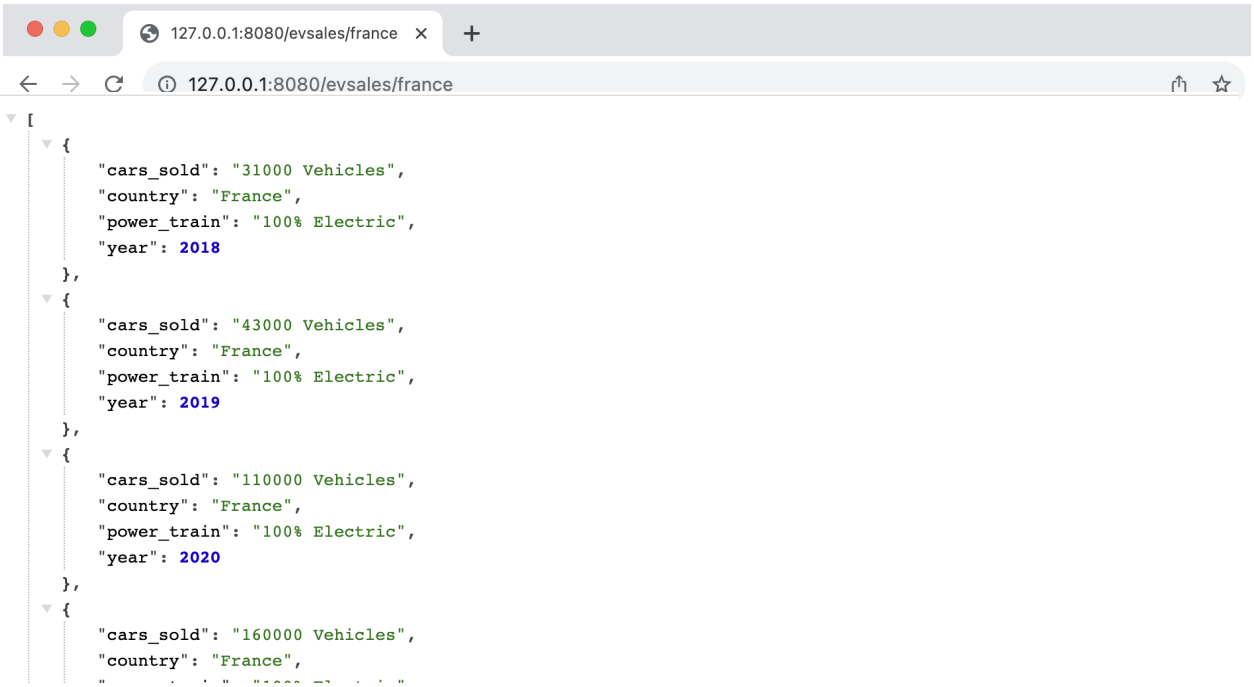
1. /evsales
2. /evsales/{country}
3. /chargingpoints
4. /chargingpoints{country}
5. /chargingpoints/{country}?powertrain={fast_charging/slow_charging}
6. /topvehiclesfrance

Let's use the /evsales/{country} endpoint and see how it works!

1. First, let's run Flask in our terminal.



2. Grab the http://127.0.0.1:8080/ port and add the selected endpoint. In this case, since we want to see the EV sales data for France, we will have to modify {country} for {France}. The full address to make this GET request will then be, http://127.0.0.1:8080/evsales/france. And, we have a successful request!



3. We can also track this API request via our terminal, since it was a successful request, it shows us a 200 HTTP code.

Time Series Analysis

Given the characteristics of the collected data, **ARIMA** (Autoregressive Integrated Moving Average) and **SARIMA** (Seasonal ARIMA) are the best models for predicting time series data. Their effectiveness lies in their capability to accurately capture and predict linear relationships and trends.

At a high level, these are the steps I followed for the implementation of these models using historical data collected on **French vehicle registrations** (full code available in repository).

1. Data Preparation and Decomposition

I started by converting a date column to a datetime format and setting it as an index, which is standard for time series analysis. I then proceeded to perform a seasonal decomposition of the data (vehicle registrations in France from 2011-2022), which is crucial for understanding underlying trends, seasonality, and residuals in the time series.

The plot itself showed me that there was a seasonal component in the time series, which meant I needed to choose p,d,q for ARIMA and P,D,Q for SARIMA

2. Model Selection and Fitting

I then used the '*auto_arima*' function from the *pmdarima* package. This function automatically selects the best ARIMA model based on the given data (grid search). It considers various combinations of the parameters (p, d, q) and seasonal parameters (P, D, Q, m) for the best fit.

From code

```
#Grid search
from pmdarima import auto_arima
stepwise_model = auto_arima(imm_france['Vehicle registrations'], start_p=1, start_q=1,
                            max_p=5, max_q=5, m=12,
                            start_P=0, start_Q=0, max_P=3, max_Q=3,
                            seasonal=True, d=1, D=1, trace=True,
                            error_action='ignore',
                            suppress_warnings=True,
                            stepwise=True)
```

3. Forecasting

Since I wanted to make a couple of "tests", I decided to make one-year and two-year predictions, where the model is fitted to the training data, and forecasts are made for the specified future periods.

```
#ONE YEAR Prediction
```

```
#Test & train split
split_date1 = '2022-01-01'
train1 = imm_france.loc[:split_date1] # Data up to and including Jan 2022 will be used for training
test1 = imm_france.loc[split_date1:] #Data from Jan 2022 onwards will be used for testing
```

```
#Fitting model
stepwise_model.fit(train1)
```

```
ARIMA
ARIMA(2,1,2)(1,1,0)[12]
```

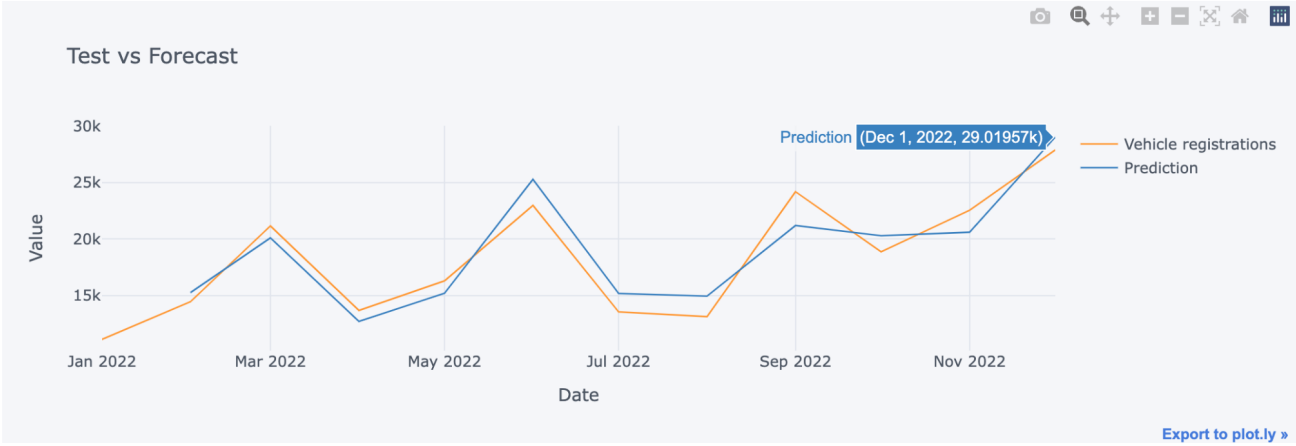
```
#Forecasting
future_forecast1 = stepwise_model.predict(n_periods=12)
future_forecast1 = pd.DataFrame(future_forecast1, index=test1.index, columns=['Prediction'])
```

4. Visualization and Analysis

I then proceeded to compare the '*future_forecast*' and '*test*' *DataFrames* together and did the exact same thing for comparing the '*future_forecast*' with the *entire dataset* to get a larger picture of the context of my prediction. For the data visualization, in both cases I used *iplot* from the *cufflinks* library, which is typically used for interactive plots.

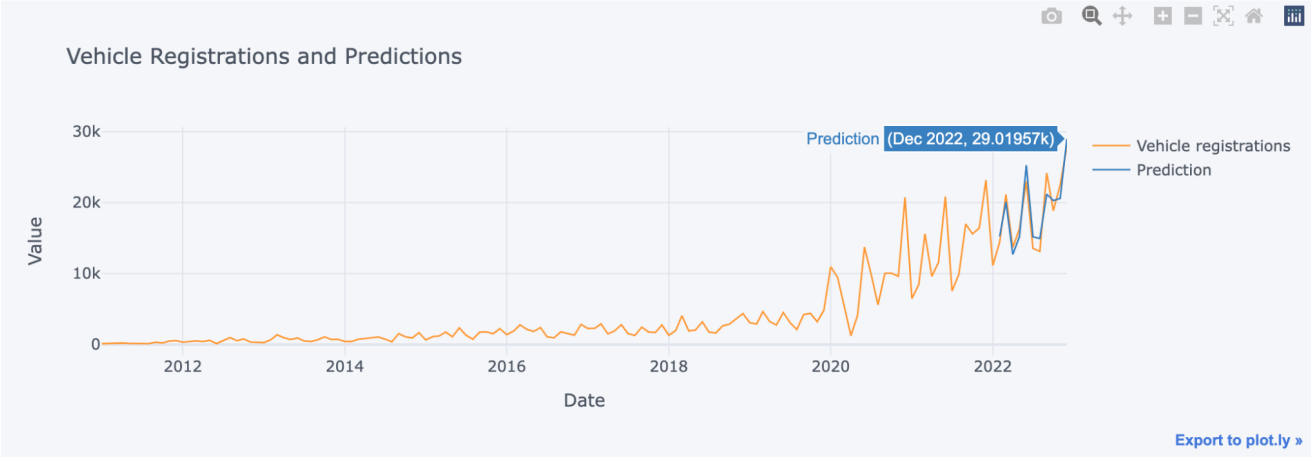
Overall, the closeness of the predicted to the actual values gave me an indicator of prediction accuracy.

Future_forecast vs test data comparison
The model captures the overall trend and seasonality in the test data quite well. The peaks and troughs of the predictions follow the actual vehicle registration data closely, which indicates a good fit



Future_forecast vs entire dataset

The model captures the overall trend and seasonality in the historical data quite well. The peaks and troughs of the predictions follow the actual vehicle registration data closely, which indicates a good fit.



5. Specific prediction

Finally, I wanted to forecast vehicle registrations for 2023 in order to apply the model in a real-world application scenario. The forecast for 2023 seems reasonable and well-aligned with the historical data's patterns.

```
#Forecasting vechicle registrations for 2023
```

```
stepwise_model.fit(imm_france)
```

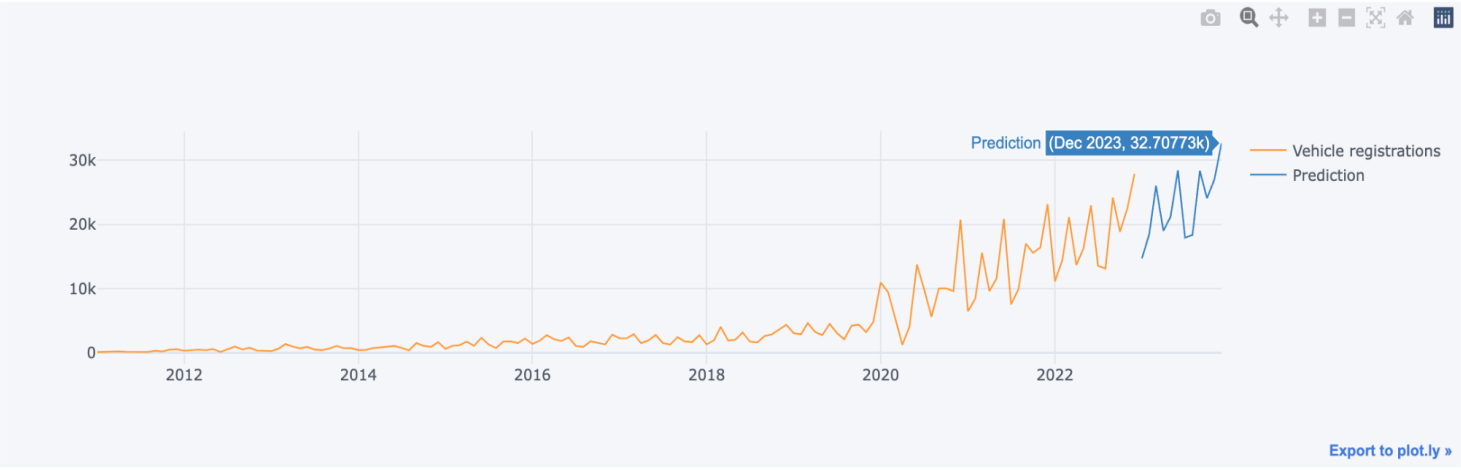
ARIMA
ARIMA(2,1,2)(1,1,0)[12]

```
future_forecast_1year = stepwise_model.predict(n_periods=12)
```

```
year_2023 = [pd.to_datetime('2023-01-01'),
             pd.to_datetime('2023-02-01'),
             pd.to_datetime('2023-03-01'),
             pd.to_datetime('2023-04-01'),
             pd.to_datetime('2023-05-01'),
             pd.to_datetime('2023-06-01'),
             pd.to_datetime('2023-07-01'),
             pd.to_datetime('2023-08-01'),
             pd.to_datetime('2023-09-01'),
             pd.to_datetime('2023-10-01'),
             pd.to_datetime('2023-11-01'),
             pd.to_datetime('2023-12-01')]
```

```
]: 1 future_forecast_1year = pd.DataFrame(future_forecast_1year, index=year_2023, columns=['Prediction'])
```

```
]: 1 pd.concat([imm_france,future_forecast_1year],axis=1).iplot()
```



Conclusion

In the document, I looked closely at two main questions about electric vehicles (EVs) in France. The first was about how French citizens are starting to buy more EVs. The graph showing EV sales from 2011 to 2021 makes it clear that more and more people are choosing electric cars. Sales shot up from just 3,000 EVs sold in 2011 to a huge 160,000 in 2021. This jump, especially from 2018 to 2021, shows that people in France are really embracing electric cars.

This suggests that a combination of factors, including environmental awareness and attractive government incentives, such as the ecological and conversion bonuses, are convincing more French consumers to go electric.

The second question was about whether France has enough charging stations for all these new electric cars. The chart tracking the number of charging points from 2011 to 2021 tells a good story. France went from having almost no public charging points to having more than 54,500 in ten years. That's a big deal because it means France is quickly getting ready for even more electric cars on the road.

So, by looking at these numbers, the future for electric cars in France is looking quite bright. With sales going up fast and more places to charge, it looks like France is on the right track to make electric cars a common choice for everyone.

GDPR

After a thorough assessment of the data collected for this project, I can confidently affirm that no personal data has been utilized in any part of the processes. The sources of the data are entirely public and at a country level, ensuring full transparency and accessibility. Therefore, this project adheres to the guidelines and principles of the General Data Protection Regulation (GDPR).

References

Scraping:

- https://photo.capital.fr/les-10-voitures-electriques-les-plus-vendues-en-france-en-2019-39395#-60j7f
- https://www.automobile-propre.com/voiture-electrique-le-top-10-des-ventes-en-france-en-2020
- https://www.capital.fr/auto/les-20-voitures-electriques-les-plus-vendues-en-france-en-2021-1424558
- https://www.automobile-propre.com/dossiers/chiffres-vente-immatriculations-france/

API:

- https://data.enedis.fr/api/explore/v2.1/console

Flat Files:

- https://www.iea.org/data-and-statistics/data-tools/global-ev-data-explorer
- https://www.data.gouv.fr/fr/datasets/fichier-consolide-des-bornes-de-recharge-pour-vehicules-electriques/#_

Databases:

- https://www.insee.fr/fr/statistiques/2015220#tableau-figure1
- https://www.data.gouv.fr/en/pages/odu/defi_vehicules-electriques/
- https://data.enedis.fr/explore/dataset/nombre-de-points-de-charge-accessibles-au-public-pour-100-000-habitants-par-reg/information/
- https://public.opendatasoft.com/explore/dataset/fichier-consolide-des-bornes-de-recharge-pour-vehicules-electriques-irve/information/

Tableau Story:

- https://public.tableau.com/views/EV_in_France_dataviz/EVstory?:language=en-GB&publish=yes&:display_count=n&:origin=viz_share_link

Github repository (in progress):

- https://github.com/hilmnr/Charging-Ahead-Frances-Electric-Vehicle-Revolution/tree/main