Bicycles use in Paris

State of the art and future

Damian Borsiak

Data Analytics Bootcamp
December 2022

Plan of Presentation

1. Introduction
2. Data sources
3. Data collection
4. Data cleaning
5. Exploratory data analysis
6. Database comparison
7. Entity-relationship model of MySQL database
8. Forecast
9. Conclusions
10. Q&A

1. Introduction

City of Paris has been deploying permanent bicycle counters for the last several years to assess the development of cycling.

Goals:

- Provide insight into bicycles traffic in Paris
- Visualize map of counters / data collection points
- Decide which Paris District has the highest bicycle traffic
- See the influence of Covid-19 pandemy on the bicycle use
- Forecast future traffic

2. Data sources

PARIS | Data

Main data source: https://opendata.paris.fr/

Website was created by an initiative of the City of Paris. We can find there datasets published by the City's services and its partners under the ODbL licence .

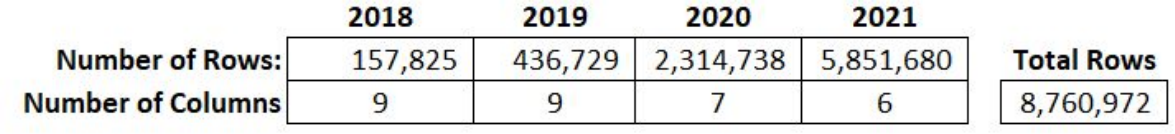
Producer: Department of Roads and Transport - City of Paris
Territory: Paris
Time zone: Europe/Paris
Language: French

Last processing:
April 25, 2022 12:28 (metadata)
April 30, 2020 12:23 (data created and posted

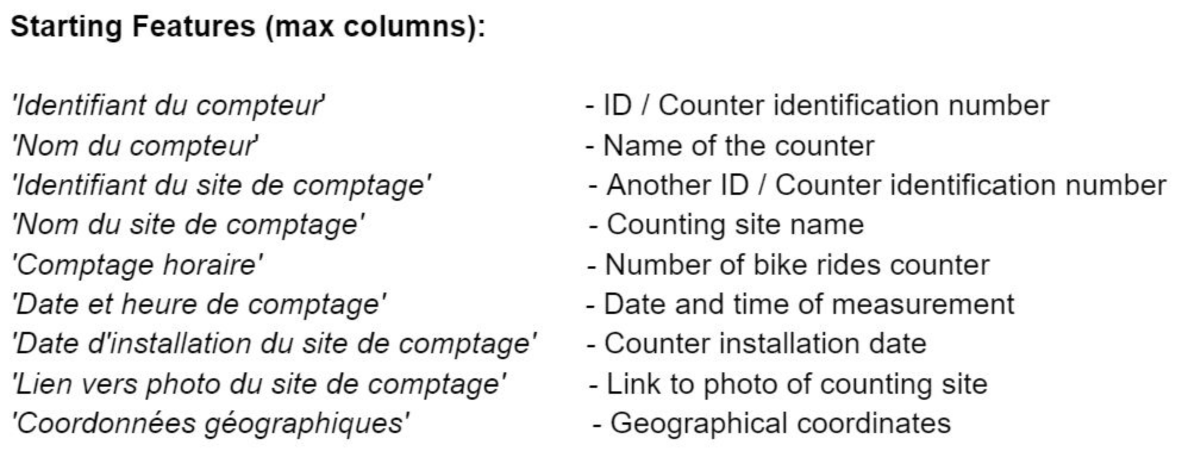
3. Data collection

Data were downloaded separately for each year: 2018, 2019, 2020, 2021
Format: compressed (zip) .csv files

https://opendata.paris.fr/explore/dataset/comptage-velo-historique-donnees-compteurs/information/



	2018	2019	2020	2021		Total Rows
Number of Rows:	157,825	436,729	2,314,738	5,851,680		8,760,972
Number of Columns	9	9	7	6		

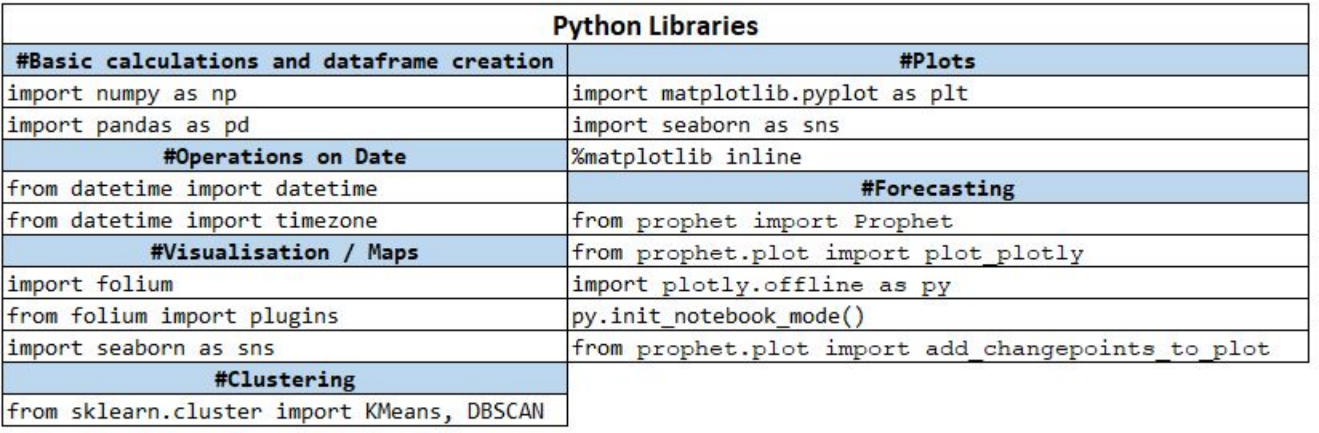


Starting Features (max columns):

'Identifiant du compteur'	- ID / Counter identification number
'Nom du compteur'	- Name of the counter
'Identifiant du site de comptage'	- Another ID / Counter identification number
'Nom du site de comptage'	- Counting site name
'Comptage horaire'	- Number of bike rides counter
'Date et heure de comptage'	- Date and time of measurement
'Date d'installation du site de comptage'	- Counter installation date
'Lien vers photo du site de comptage'	- Link to photo of counting site
'Coordonnées géographiques'	- Geographical coordinates

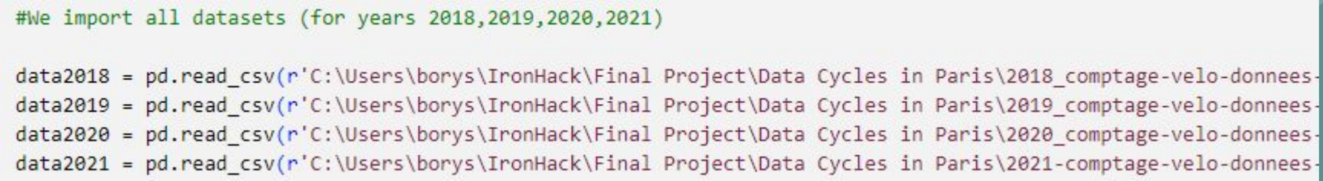
4. Data cleaning

To achieve my goals, I was using below Python Libraries:



Python Libraries	
#Basic calculations and dataframe creation	#Plots
import numpy as np	import matplotlib.pyplot as plt
import pandas as pd	import seaborn as sns
#Operations on Date	%matplotlib inline
from datetime import datetime	#Forecasting
from datetime import timezone	from prophet import Prophet
#Visualisation / Maps	from prophet.plot import plot_plotly
import folium	import plotly.offline as py
from folium import plugins	py.init_notebook_mode()
import seaborn as sns	from prophet.plot import add_changepoints_to_plot
#Clustering	
from sklearn.cluster import KMeans, DBSCAN	

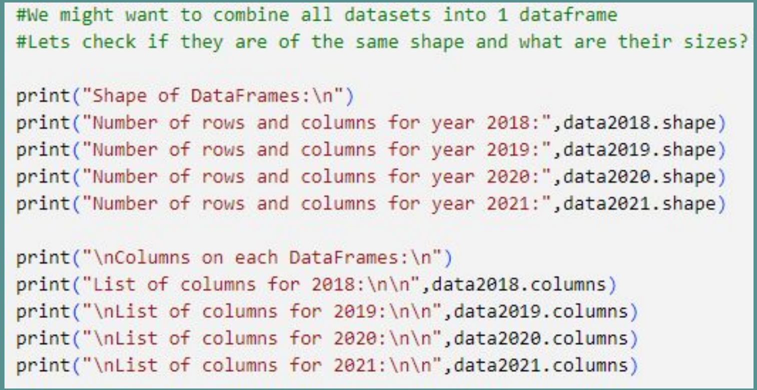
Import of 4 files with data into Python



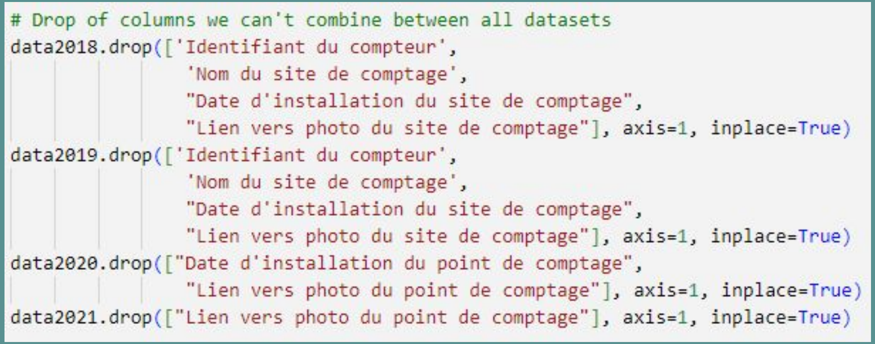
```
#We import all datasets (for years 2018,2019,2020,2021)

data2018 = pd.read_csv(r'C:\Users\borys\IronHack\Final Project\Data Cycles in Paris\2018_comptage-velo-donnees-
data2019 = pd.read_csv(r'C:\Users\borys\IronHack\Final Project\Data Cycles in Paris\2019_comptage-velo-donnees-
data2020 = pd.read_csv(r'C:\Users\borys\IronHack\Final Project\Data Cycles in Paris\2020_comptage-velo-donnees-
data2021 = pd.read_csv(r'C:\Users\borys\IronHack\Final Project\Data Cycles in Paris\2021-comptage-velo-donnees-
```

4. Data cleaning

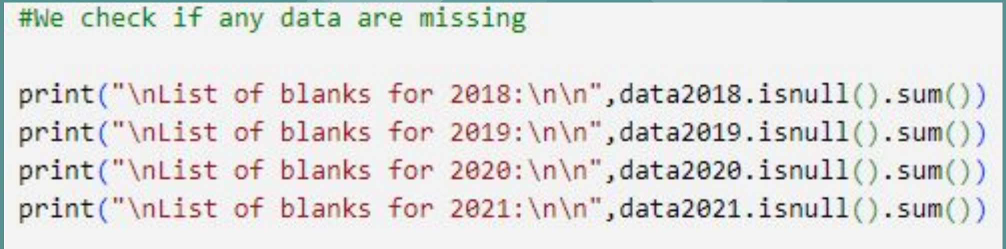
Identify data Shapes - number of rows and columns



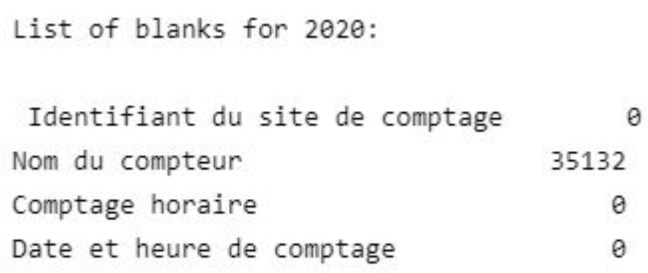
```
#We might want to combine all datasets into 1 dataframe
#Lets check if they are of the same shape and what are their sizes?

print("Shape of DataFrames:\n")
print("Number of rows and columns for year 2018:",data2018.shape)
print("Number of rows and columns for year 2019:",data2019.shape)
print("Number of rows and columns for year 2020:",data2020.shape)
print("Number of rows and columns for year 2021:",data2021.shape)

print("\nColumns on each DataFrames:\n")
print("List of columns for 2018:\n\n",data2018.columns)
print("\nList of columns for 2019:\n\n",data2019.columns)
print("\nList of columns for 2020:\n\n",data2020.columns)
print("\nList of columns for 2021:\n\n",data2021.columns)
```



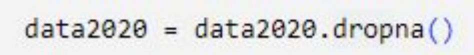
```
# Drop of columns we can't combine between all datasets
data2018.drop(['Identifiant du compteur',
               'Nom du site de comptage',
               "Date d'installation du site de comptage",
               "Lien vers photo du site de comptage"], axis=1, inplace=True)
data2019.drop(['Identifiant du compteur',
               'Nom du site de comptage',
               "Date d'installation du site de comptage",
               "Lien vers photo du site de comptage"], axis=1, inplace=True)
data2020.drop(["Date d'installation du point de comptage",
               "Lien vers photo du point de comptage"], axis=1, inplace=True)
data2021.drop(["Lien vers photo du point de comptage"], axis=1, inplace=True)
```



```
#We check if any data are missing

print("\nList of blanks for 2018:\n\n",data2018.isnull().sum())
print("\nList of blanks for 2019:\n\n",data2019.isnull().sum())
print("\nList of blanks for 2020:\n\n",data2020.isnull().sum())
print("\nList of blanks for 2021:\n\n",data2021.isnull().sum())
```

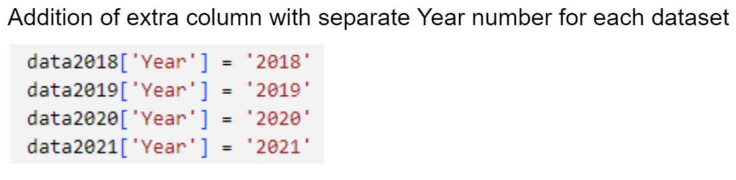


```
List of blanks for 2020:


 Identifiant du site de comptage        0
Nom du compteur                     35132
Comptage horaire                        0
Date et heure de comptage               0
```



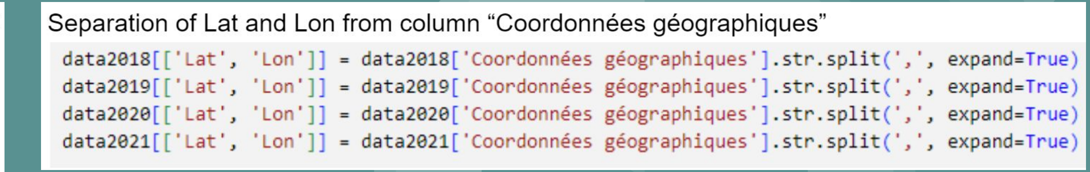
```
data2020 = data2020.dropna()
```

4. Data cleaning



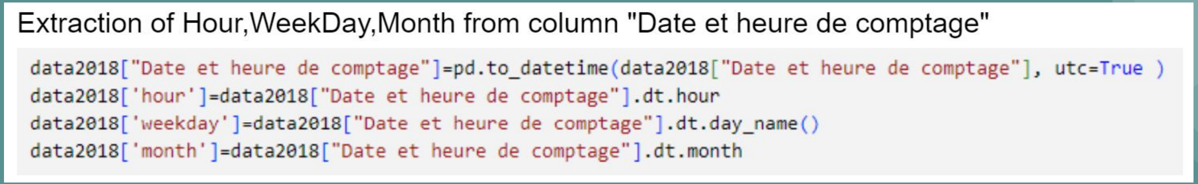
Addition of extra column with separate Year number for each dataset

```
data2018['Year'] = '2018'
data2019['Year'] = '2019'
data2020['Year'] = '2020'
data2021['Year'] = '2021'
```



Separation of Lat and Lon from column “Coordonnées géographiques”

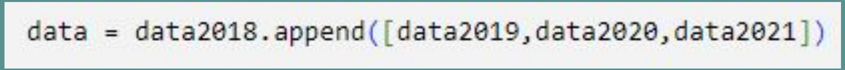
```
data2018[['Lat', 'Lon']] = data2018['Coordonnées géographiques'].str.split(',', expand=True)
data2019[['Lat', 'Lon']] = data2019['Coordonnées géographiques'].str.split(',', expand=True)
data2020[['Lat', 'Lon']] = data2020['Coordonnées géographiques'].str.split(',', expand=True)
data2021[['Lat', 'Lon']] = data2021['Coordonnées géographiques'].str.split(',', expand=True)
```



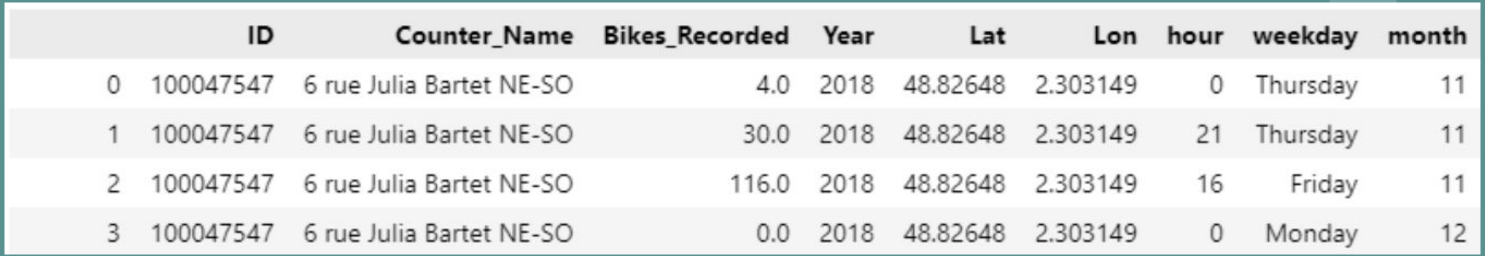
Extraction of Hour,WeekDay,Month from column "Date et heure de comptage"

```
data2018["Date et heure de comptage"]=pd.to_datetime(data2018["Date et heure de comptage"], utc=True )
data2018['hour']=data2018["Date et heure de comptage"].dt.hour
data2018['weekday']=data2018["Date et heure de comptage"].dt.day_name()
data2018['month']=data2018["Date et heure de comptage"].dt.month
```

Final Result



```
data = data2018.append([data2019,data2020,data2021])
```

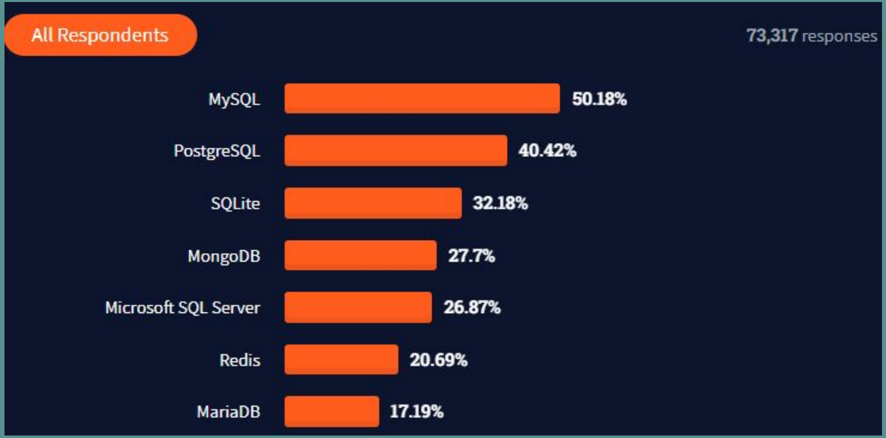


	ID	Counter_Name	Bikes_Recorded	Year	Lat	Lon	hour	weekday	month
0	100047547	6 rue Julia Bartet NE-SO	4.0	2018	48.82648	2.303149	0	Thursday	11
1	100047547	6 rue Julia Bartet NE-SO	30.0	2018	48.82648	2.303149	21	Thursday	11
2	100047547	6 rue Julia Bartet NE-SO	116.0	2018	48.82648	2.303149	16	Friday	11
3	100047547	6 rue Julia Bartet NE-SO	0.0	2018	48.82648	2.303149	0	Monday	12

5. Exploratory data analysis

Presentation in:

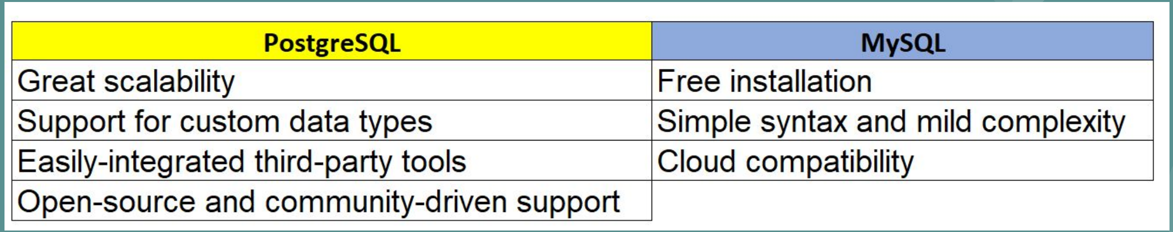
6. Database comparison



Most popular database systems. Source: 2021 Developer Survey by StackOverflow

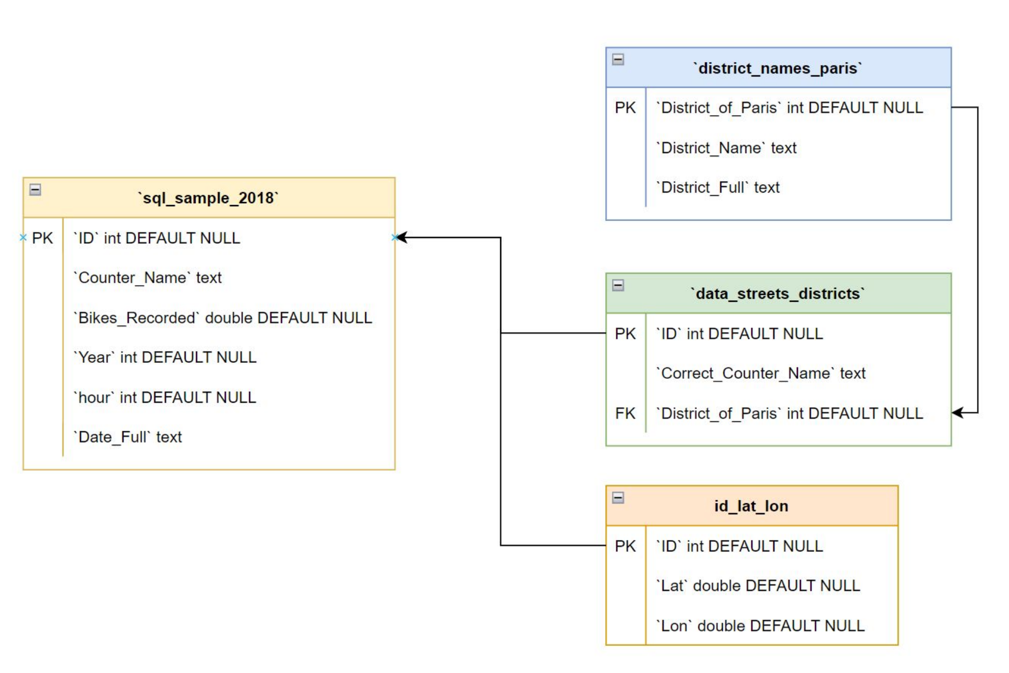
I have selected MySQL Database as its one of the most popular relational database systems in the industry and offer perfect capabilities for my project scale and complexity. As it is widely used and I am already familiar with it, it was the best choice.

I was also considering PostgreSQL as a potential Database, but decided that it could be less tailored to my needs, simply by offering other - not needed in this project functionalities.



PostgreSQL	MySQL
Great scalability	Free installation
Support for custom data types	Simple syntax and mild complexity
Easily-integrated third-party tools	Cloud compatibility
Open-source and community-driven support	

7. Entity-relationship model of MySQL database

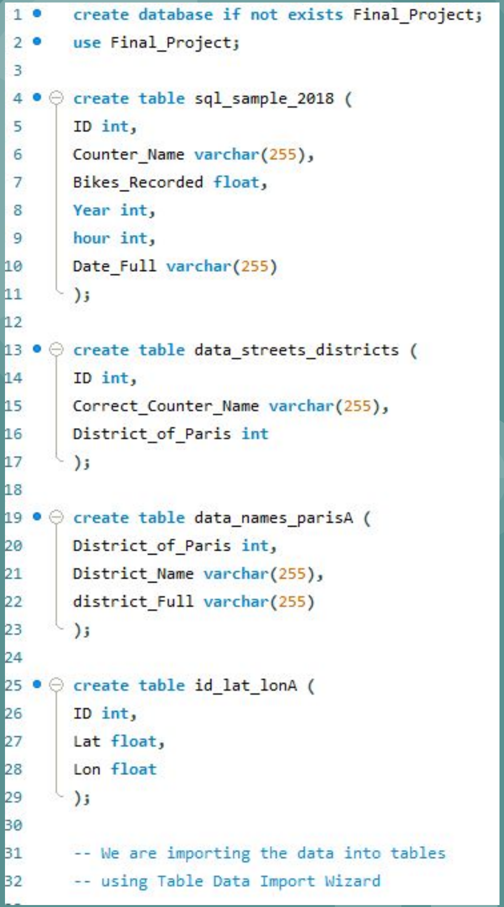


	`sql_sample_2018`
PK	`ID` int DEFAULT NULL
	`Counter_Name` text
	`Bikes_Recorded` double DEFAULT NULL
	`Year` int DEFAULT NULL
	`hour` int DEFAULT NULL
	`Date_Full` text

	`district_names_paris`
PK	`District_of_Paris` int DEFAULT NULL
	`District_Name` text
	`District_Full` text

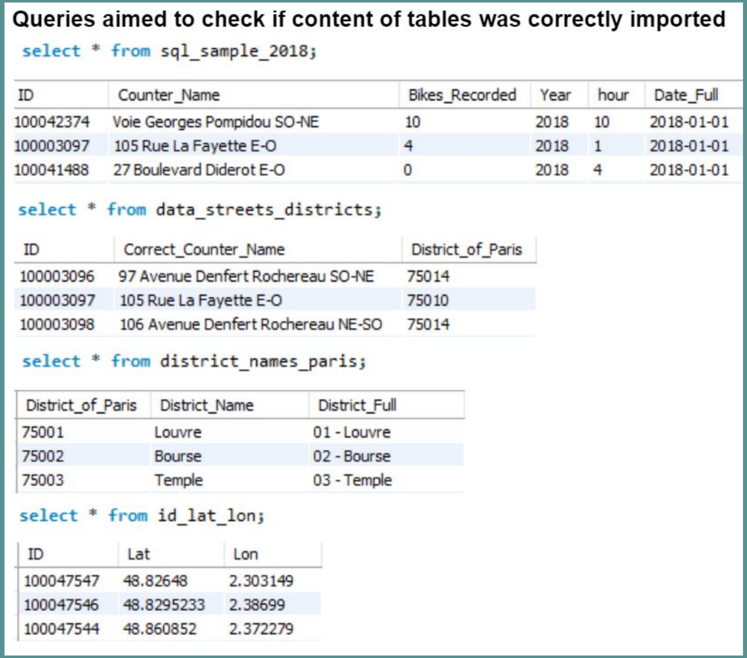
	`data_streets_districts`
PK	`ID` int DEFAULT NULL
	`Correct_Counter_Name` text
FK	`District_of_Paris` int DEFAULT NULL

	id_lat_lon
PK	`ID` int DEFAULT NULL
	`Lat` double DEFAULT NULL
	`Lon` double DEFAULT NULL



```sql
create database if not exists Final_Project;
use Final_Project;

create table sql_sample_2018 (
ID int,
Counter_Name varchar(255),
Bikes_Recorded float,
Year int,
hour int,
Date_Full varchar(255)
);

create table data_streets_districts (
ID int,
Correct_Counter_Name varchar(255),
District_of_Paris int
);

create table data_names_parisA (
District_of_Paris int,
District_Name varchar(255),
district_Full varchar(255)
);

create table id_lat_lonA (
ID int,
Lat float,
Lon float
);

-- We are importing the data into tables
-- using Table Data Import Wizard
```

SQL Queries



Queries aimed to check if content of tables was correctly imported

```sql
select * from sql_sample_2018;
```

ID	Counter_Name	Bikes_Recorded	Year	hour	Date_Full
100042374	Voie Georges Pompidou SO-NE	10	2018	10	2018-01-01
100003097	105 Rue La Fayette E-O	4	2018	1	2018-01-01
100041488	27 Boulevard Diderot E-O	0	2018	4	2018-01-01

```sql
select * from data_streets_districts;
```

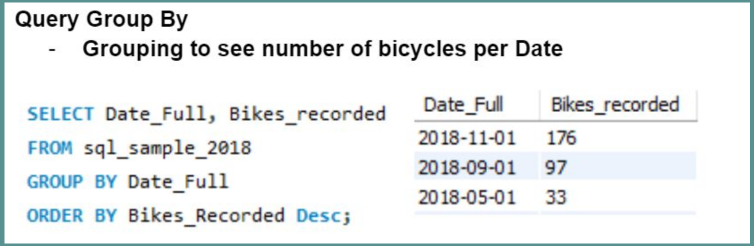
ID	Correct_Counter_Name	District_of_Paris
100003096	97 Avenue Denfert Rochereau SO-NE	75014
100003097	105 Rue La Fayette E-O	75010
100003098	106 Avenue Denfert Rochereau NE-SO	75014

```sql
select * from district_names_paris;
```

District_of_Paris	District_Name	District_Full
75001	Louvre	01 - Louvre
75002	Bourse	02 - Bourse
75003	Temple	03 - Temple

```sql
select * from id_lat_lon;
```

ID	Lat	Lon
100047547	48.82648	2.303149
100047546	48.8295233	2.38699
100047544	48.860852	2.372279

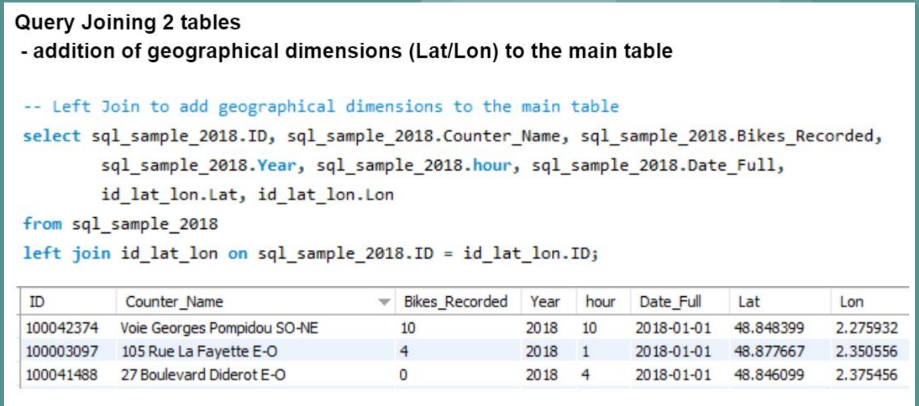


Query Group By

- Grouping to see number of bicycles per Date

```sql
SELECT Date_Full, Bikes_recorded
FROM sql_sample_2018
GROUP BY Date_Full
ORDER BY Bikes_Recorded Desc;
```

Date_Full	Bikes_recorded
2018-11-01	176
2018-09-01	97
2018-05-01	33

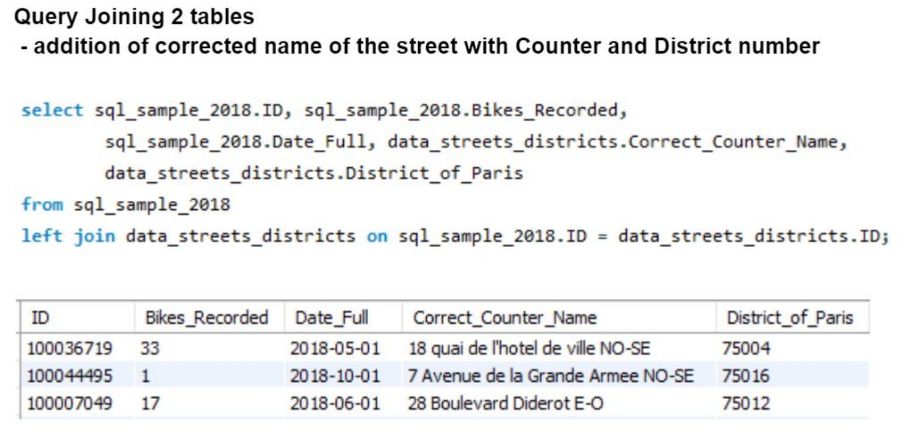


Query Joining 2 tables

- addition of geographical dimensions (Lat/Lon) to the main table

```sql
-- Left Join to add geographical dimensions to the main table
select sql_sample_2018.ID, sql_sample_2018.Counter_Name, sql_sample_2018.Bikes_Recorded,
       sql_sample_2018.Year, sql_sample_2018.hour, sql_sample_2018.Date_Full,
       id_lat_lon.Lat, id_lat_lon.Lon
from sql_sample_2018
left join id_lat_lon on sql_sample_2018.ID = id_lat_lon.ID;
```

ID	Counter_Name	Bikes_Recorded	Year	hour	Date_Full	Lat	Lon
100042374	Voie Georges Pompidou SO-NE	10	2018	10	2018-01-01	48.848399	2.275932
100003097	105 Rue La Fayette E-O	4	2018	1	2018-01-01	48.877667	2.350556
100041488	27 Boulevard Diderot E-O	0	2018	4	2018-01-01	48.846099	2.375456



Query Joining 2 tables

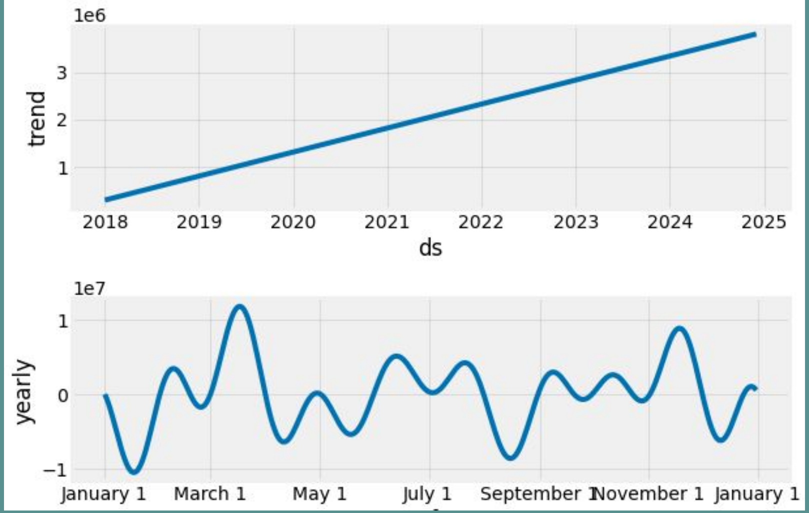
- addition of corrected name of the street with Counter and District number

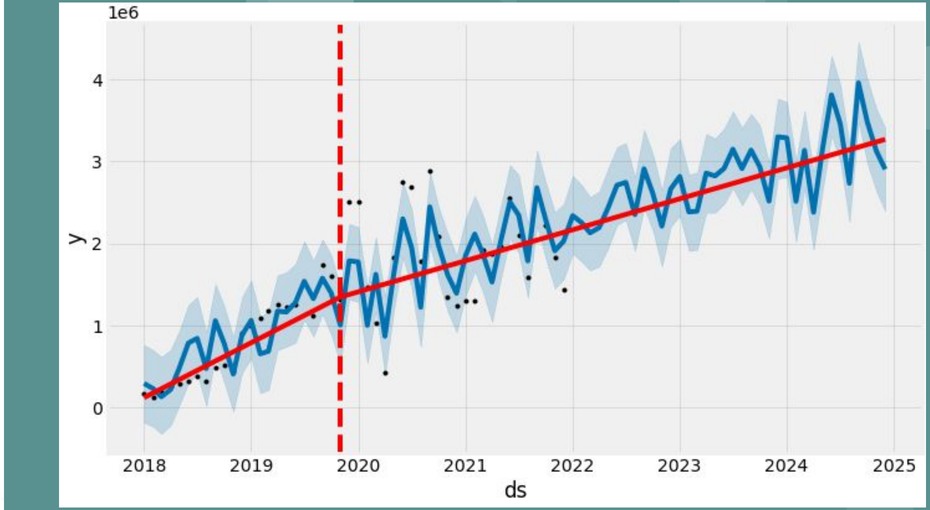
```sql
select sql_sample_2018.ID, sql_sample_2018.Bikes_Recorded,
       sql_sample_2018.Date_Full, data_streets_districts.Correct_Counter_Name,
       data_streets_districts.District_of_Paris
from sql_sample_2018
left join data_streets_districts on sql_sample_2018.ID = data_streets_districts.ID;
```

ID	Bikes_Recorded	Date_Full	Correct_Counter_Name	District_of_Paris
100036719	33	2018-05-01	18 quai de l'hotel de ville NO-SE	75004
100044495	1	2018-10-01	7 Avenue de la Grande Armee NO-SE	75016
100007049	17	2018-06-01	28 Boulevard Diderot E-O	75012

8. Forecasting - Time Series Forecasting with Facebook Prophet

The Prophet library is an open-source library designed for making forecasts for univariate time series datasets. It is easy to use and designed to automatically find a good set of hyperparameters for the model in an effort to make skillful forecasts for data with trends and seasonal structure by default.





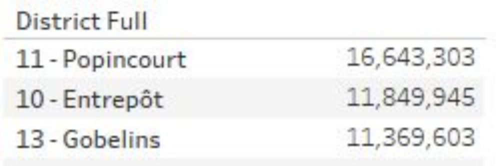
We can expect a **strong increase in the number of bicycle rides in Paris** - despite temporarily lower numbers during the Covid-19 pandemy. Trend is clearly positive and together with earlier information about plans to make Paris completely cyclable by 2026 - **we can be optimistic about their future.**

9. Conclusions

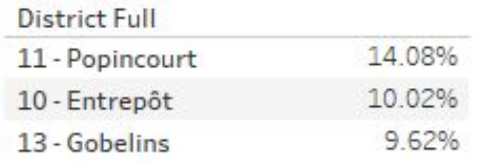
Based on detailed analysis of received data, we can conclude that Bicycles have a bright future in the city of Paris.

Number of bicycle rides is expected to increase significantly in coming years, despite Covid-19 Pandemic

Top 3 Districts with highest number of travelers are:



District Full	
11 - Popincourt	16,643,303
10 - Entrepôt	11,849,945
13 - Gobelins	11,369,603



District Full	
11 - Popincourt	14.08%
10 - Entrepôt	10.02%
13 - Gobelins	9.62%

While working on the data, I had to consider an uneven number of the counting machines that were first installed in small numbers in 2018 - factor that could negatively affect my analysis and the forecast.

QUESTIONS ?

Let’s stay in touch!

Damian Borsiak
Email: damian@borsiak.com
LinkedIn: link