AirportAnalysis

May 23, 2021

[42]:
```
import pandas as pd
import json
```

[43]:
```
df = pd.read_csv("data/spain_airports.csv")
df = df.drop(columns=["city", "Unnamed: 0"],axis=1)
```

[44]:
```
df.head()
```

[44]:

	icao	iata	name	type	position
0	DAAB	QLD	Blida Airport	medium_airport	{'longitude': 2.8141698837280273, 'latitude': …
1	DAAK	NaN	Boufarik Airport	medium_airport	{'longitude': 2.87611, 'latitude': 36.545799, …
2	DAAX	NaN	Chéraga Airport	small_airport	{'longitude': 2.9284, 'latitude': 36.7782, 'al…
3	DAOI	CFK	Ech Cheliff Airport	medium_airport	{'longitude': 1.33176994324, 'latitude': 36.21…
4	LE83	NaN	Aeródromo forestal de Mojados	small_airport	{'longitude': -4.713068, 'latitude': 41.465728…

[45]:
```
df["position"] = df.apply(lambda x: json.loads(x.get('position').replace("'",␣
↪"\"").replace("True", "\"True\"")),axis=1)
```

[46]:
```
for i,v in df["position"].items():
    df.loc[i,"latitude"] = v["latitude"]
    df.loc[i,"longitude"] = v["longitude"]
    df.loc[i,"altitude"] = v["altitude"]
```

[47]:
```
df.to_csv("data/spain_airports_processed.csv")
```

[48]:
```
df.head()
```

[48]:

	icao	iata	name	type
0	DAAB	QLD	Blida Airport	medium_airport
1	DAAK	NaN	Boufarik Airport	medium_airport

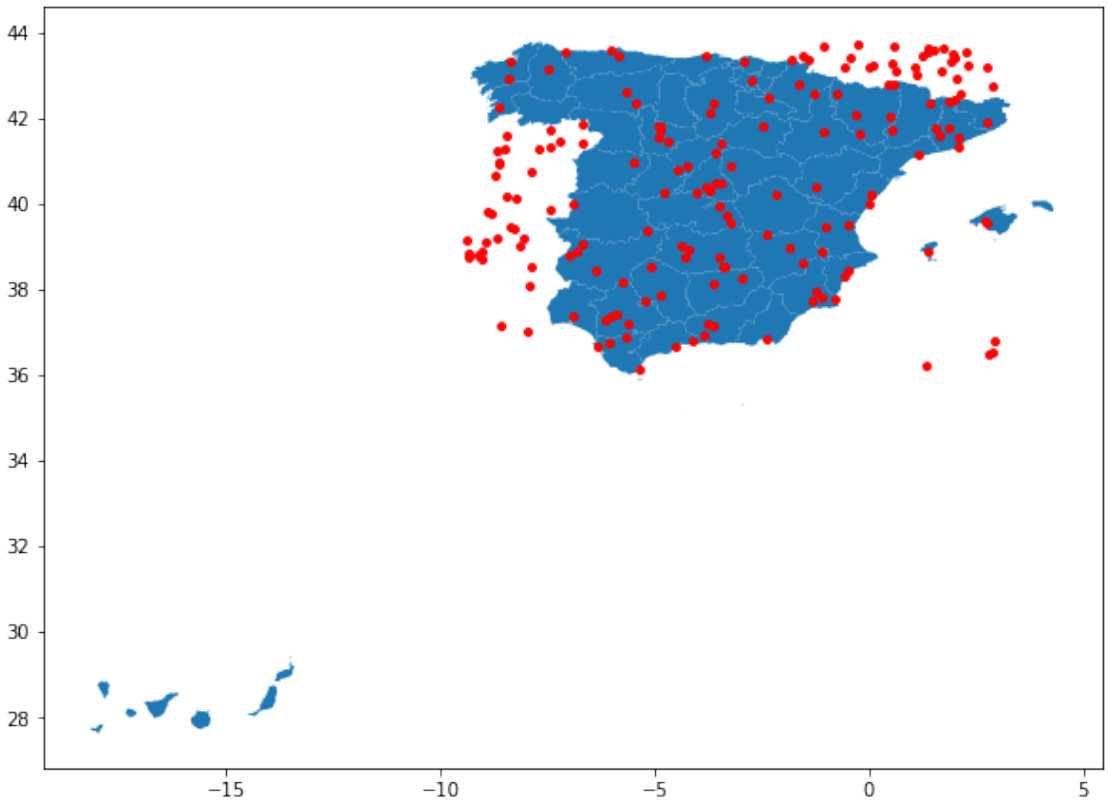
```
2  DAAX  NaN                Chéraga Airport   small_airport
3  DAOI  CFK            Ech Cheliff Airport  medium_airport
4  LE83  NaN  Aeródromo forestal de Mojados   small_airport

                                              position   latitude  longitude  \
0  {'longitude': 2.8141698837280273, 'latitude': …  36.503601   2.814170
1  {'longitude': 2.87611, 'latitude': 36.545799, …  36.545799   2.876110
2  {'longitude': 2.9284, 'latitude': 36.7782, 'al…  36.778200   2.928400
3  {'longitude': 1.33176994324, 'latitude': 36.21…  36.212700   1.331770
4  {'longitude': -4.713068, 'latitude': 41.465728…  41.465728  -4.713068

   altitude
0  163.0680
1  102.1080
2  120.7008
3  141.1224
4  709.8792
```

[49]:
```
import pandas as pd
from shapely.geometry import Point
import geopandas as gpd
from geopandas import GeoDataFrame
from matplotlib import pyplot as plt
import matplotlib.patches as mpatches

geometry = [Point(xy) for xy in zip(df['longitude'], df['latitude'])]
gdf = GeoDataFrame(df, geometry=geometry)
world = gpd.read_file('./map/gadm36_ESP_2.shp')
gdf.plot(ax=world.plot(figsize=(10, 10)), marker='o', color='red',␣
 ↪markersize=15);
```

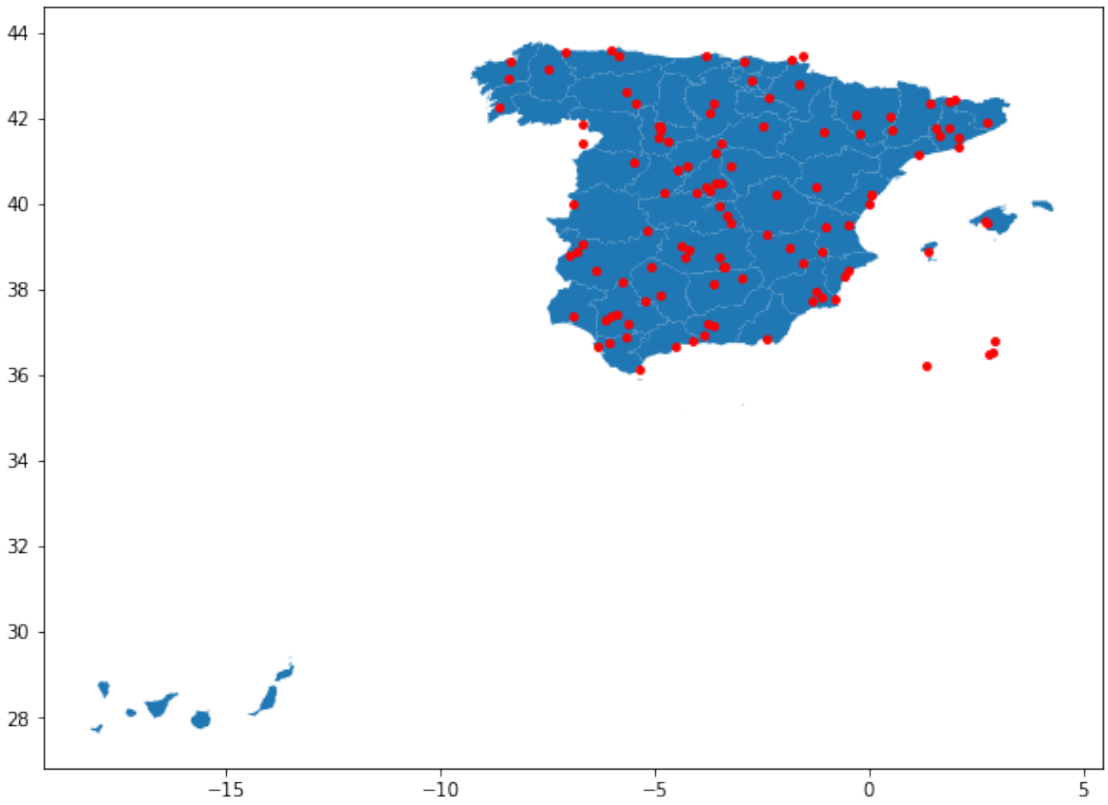


0.0.1 Procesado de outliers (Aeropuertos fuera de España)

[50]:
```
df = df.loc[~((df["longitude"] < -7) & (df["latitude"] < 42)) &␣
↪~((df["longitude"] > -1.5) & (df["latitude"] >42.5))]
```

[51]:
```
df_large = df.loc[df["type"] == "large_airport"]
df_large  = df_large.drop(columns=["position", "altitude", "geometry"], axis=1)
df_large.to_csv("test.csv")
```

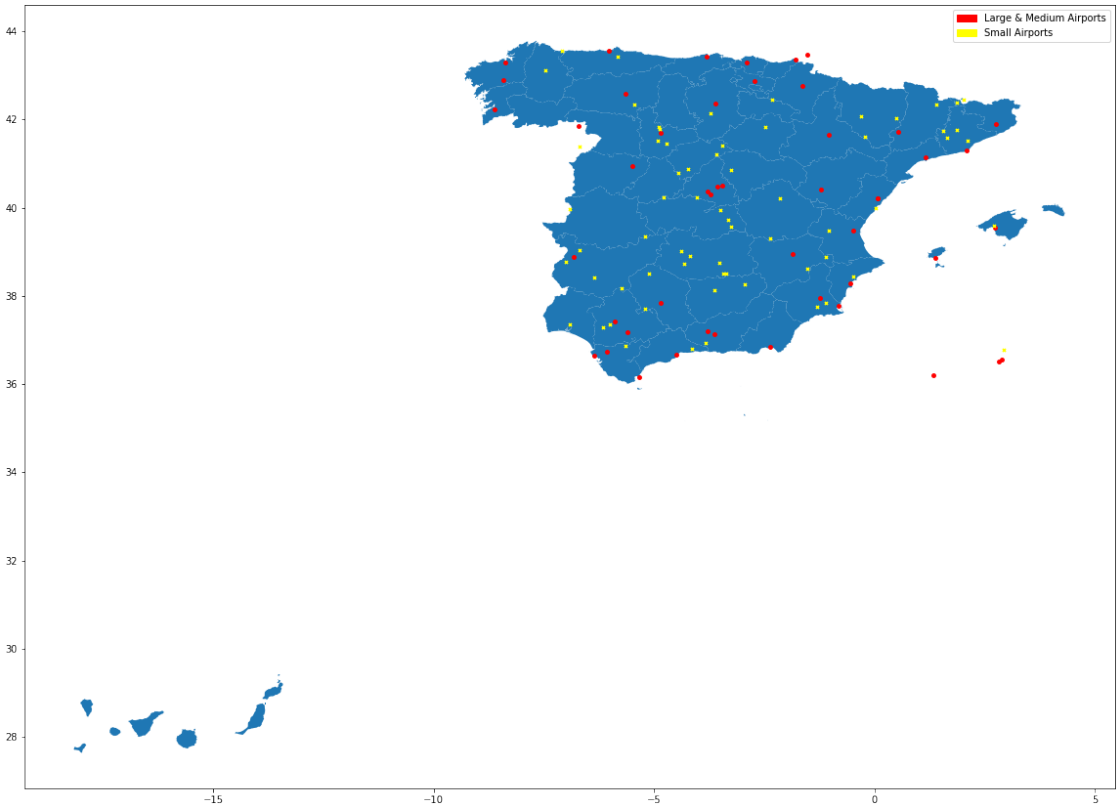
[52]:
```
geometry = [Point(xy) for xy in zip(df['longitude'], df['latitude'])]
gdf = GeoDataFrame(df, geometry=geometry)
world = gpd.read_file('./map/gadm36_ESP_2.shp')
gdf.plot(ax=world.plot(figsize=(10, 10)), marker='o', color='red',␣
↪markersize=15);
```



0.1 Geoposicionado de los Aeropuertos en España

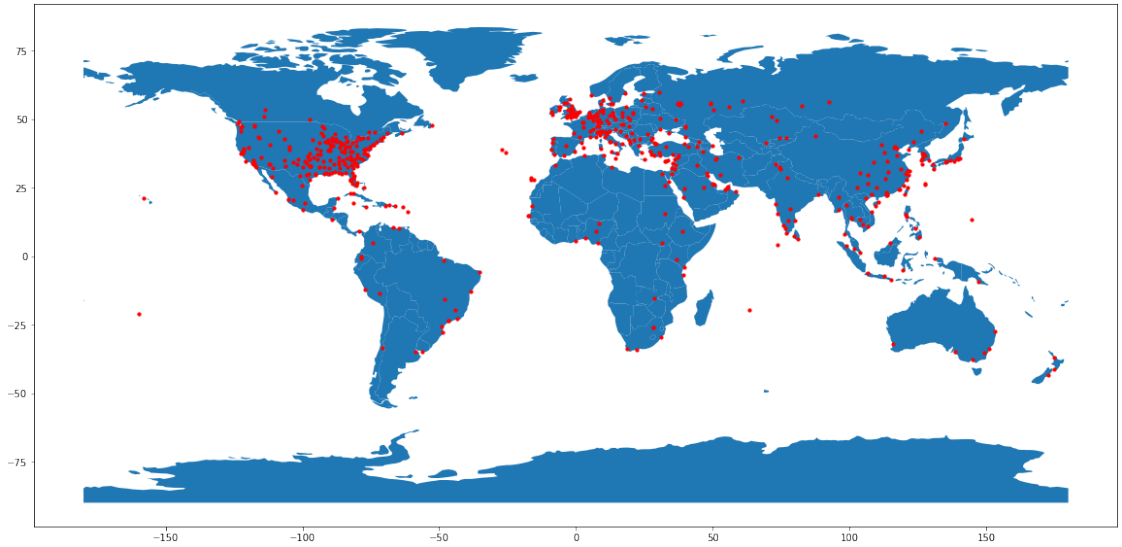
[53]:
```
df_large = df.loc[~(df["type"] == "small_airport")]
df_small = df.loc[df["type"] == "small_airport"]
geometry_l = [Point(xy) for xy in zip(df_large['longitude'],␣
 ↪df_large['latitude'])]
geometry_s = [Point(xy) for xy in zip(df_small['longitude'],␣
 ↪df_small['latitude'])]
gdf_l = GeoDataFrame(df_large, geometry=geometry_l)
gdf_s = GeoDataFrame(df_small, geometry=geometry_s)

base = world.plot(figsize=(20, 20))
#this is a simple map that goes with geopandas
world = gpd.read_file('./map/gadm36_ESP_2.shp')
#column='id', categorical=True
gdf_l.plot(ax=base, marker='o', color='red', markersize=15);
gdf_s.plot(ax=base, marker='x', color='yellow', markersize=10);
l = mpatches.Patch(color='red', label='Large & Medium Airports')
s = mpatches.Patch(color='yellow', label='Small Airports')
plt.legend(handles=[l,s])
plt.show()
```

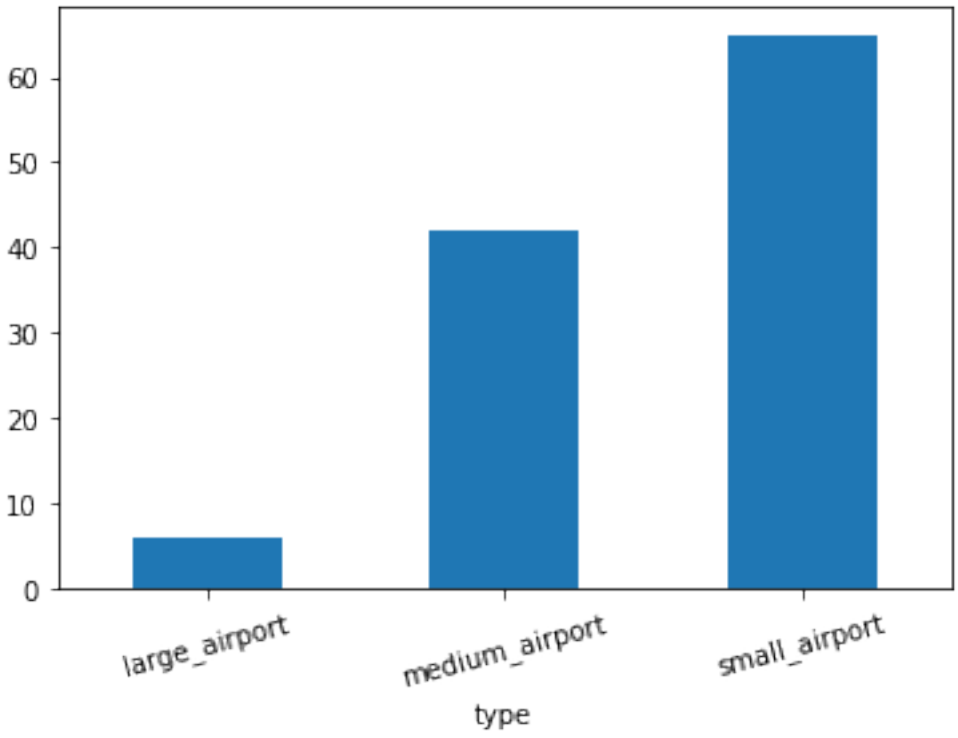


[54]:
```
df_total = pd.read_csv("data/airports_processed.csv")
df_total = df_total.loc[(df_total["type"] == "large_airport")]
geometry = [Point(xy) for xy in zip(df_total['longitude'],␣
 ↪df_total['latitude'])]
gdf = GeoDataFrame(df_total, geometry=geometry)

world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))
gdf.plot(ax=world.plot(figsize=(20, 10)), marker='o', color='red',␣
 ↪markersize=10);
```



[55]:
```
df_plot = df.groupby(['type']).size().plot.bar(rot=15)
```



[ ]:
```

```