Entrega Semana 7 Kmeans

- A01793808 Jorge Estivent Cruz Mahecha

Este notebook se basa en información de target

Ahora imagina que somos parte del equipo de data science de la empresa Target, una de las tiendas con mayor presencia en Estados Unidos. El departamento de logistica acude a nosotros para saber donde le conviene poner sus almacenes, para que se optimice el gasto de gasolina, los tiempos de entrega de los productos y se disminuyan costos. Para ello, nos pasan los datos de latitud y longitud de cada una de las tiendas.

https://www.kaggle.com/datasets/saejinmahlauheinert/target-store-locations?select=target-locations.csv

Si quieres saber un poco más de graficas geográficas consulta el siguiente notebook https://colab.research.google.com/github/QuantEcon/quantecon-notebooks-datascience/blob/master/applications/maps.ipynb#scrollTo=uo2oPtSCeAOz

In [2]:
```
! pip install qeds fiona geopandas xgboost gensim folium pyLDAvis descartes
```

```
Requirement already satisfied: qeds in c:\python310\lib\site-packages (0.7.0)
Requirement already satisfied: fiona in c:\python310\lib\site-packages (1.8.22)
Requirement already satisfied: geopandas in c:\python310\lib\site-packages (0.12.1)
Requirement already satisfied: xgboost in c:\python310\lib\site-packages (1.7.1)
Requirement already satisfied: gensim in c:\python310\lib\site-packages (4.2.0)
Requirement already satisfied: folium in c:\python310\lib\site-packages (0.13.0)
Requirement already satisfied: pyLDAvis in c:\python310\lib\site-packages (3.3.1)
Requirement already satisfied: descartes in c:\python310\lib\site-packages (1.1.0)
Requirement already satisfied: pandas in c:\python310\lib\site-packages (from qeds)
(1.5.0)
Requirement already satisfied: requests in c:\python310\lib\site-packages (from qeds)
(2.28.1)
Requirement already satisfied: quandl in c:\python310\lib\site-packages (from qeds)
(3.7.0)
Requirement already satisfied: scipy in c:\python310\lib\site-packages (from qeds)
(1.9.2)
Requirement already satisfied: numpy in c:\python310\lib\site-packages (from qeds)
(1.23.3)
Requirement already satisfied: quantecon in c:\python310\lib\site-packages (from qed
s) (0.5.3)
Requirement already satisfied: matplotlib in c:\python310\lib\site-packages (from qed
s) (3.6.1)
Requirement already satisfied: pyarrow in c:\python310\lib\site-packages (from qeds)
(10.0.0)
Requirement already satisfied: openpyxl in c:\python310\lib\site-packages (from qeds)
(3.0.10)
Requirement already satisfied: plotly in c:\python310\lib\site-packages (from qeds)
(5.11.0)
Requirement already satisfied: pandas_datareader in c:\python310\lib\site-packages (f
rom qeds) (0.10.0)
Requirement already satisfied: scikit-learn in c:\python310\lib\site-packages (from q
eds) (1.1.2)
Requirement already satisfied: seaborn in c:\python310\lib\site-packages (from qeds)
(0.12.0)
Requirement already satisfied: statsmodels in c:\python310\lib\site-packages (from qe
ds) (0.13.5)
Requirement already satisfied: cligj>=0.5 in c:\python310\lib\site-packages (from fio
na) (0.7.2)
Requirement already satisfied: attrs>=17 in c:\python310\lib\site-packages (from fion
a) (22.1.0)
Requirement already satisfied: click>=4.0 in c:\python310\lib\site-packages (from fio
na) (8.1.3)
Requirement already satisfied: certifi in c:\python310\lib\site-packages (from fiona)
(2022.9.24)
Requirement already satisfied: six>=1.7 in c:\python310\lib\site-packages (from fion
a) (1.16.0)
Requirement already satisfied: munch in c:\python310\lib\site-packages (from fiona)
(2.5.0)
Requirement already satisfied: setuptools in c:\python310\lib\site-packages (from fio
na) (63.2.0)
Requirement already satisfied: click-plugins>=1.0 in c:\python310\lib\site-packages
(from fiona) (1.1.1)
Requirement already satisfied: packaging in c:\python310\lib\site-packages (from geop
andas) (21.3)
Requirement already satisfied: shapely>=1.7 in c:\python310\lib\site-packages (from g
eopandas) (1.8.5.post1)
Requirement already satisfied: pyproj>=2.6.1.post1 in c:\python310\lib\site-packages
(from geopandas) (3.4.0)
Requirement already satisfied: Cython==0.29.28 in c:\python310\lib\site-packages (fro
m gensim) (0.29.28)
```

```
Requirement already satisfied: smart-open>=1.8.1 in c:\python310\lib\site-packages (f
rom gensim) (6.2.0)
Requirement already satisfied: branca>=0.3.0 in c:\python310\lib\site-packages (from
folium) (0.6.0)
Requirement already satisfied: jinja2>=2.9 in c:\python310\lib\site-packages (from fo
lium) (3.1.2)
Requirement already satisfied: sklearn in c:\python310\lib\site-packages (from pyLDAv
is) (0.0)
Requirement already satisfied: funcy in c:\python310\lib\site-packages (from pyLDAvi
s) (1.17)
Requirement already satisfied: joblib in c:\python310\lib\site-packages (from pyLDAvi
s) (1.2.0)
Requirement already satisfied: numexpr in c:\python310\lib\site-packages (from pyLDAv
is) (2.8.4)
Requirement already satisfied: future in c:\python310\lib\site-packages (from pyLDAvi
s) (0.18.2)
Requirement already satisfied: colorama in c:\python310\lib\site-packages (from click
>=4.0->fiona) (0.4.5)
Requirement already satisfied: MarkupSafe>=2.0 in c:\python310\lib\site-packages (fro
m jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: pytz>=2020.1 in c:\python310\lib\site-packages (from p
andas->qeds) (2022.4)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\python310\lib\site-packag
es (from pandas->qeds) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\python310\lib\site-packages (from
matplotlib->qeds) (9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\python310\lib\site-packages (fr
om matplotlib->qeds) (3.0.9)
Requirement already satisfied: cycler>=0.10 in c:\python310\lib\site-packages (from m
atplotlib->qeds) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\python310\lib\site-packages (f
rom matplotlib->qeds) (4.37.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\python310\lib\site-packages (f
rom matplotlib->qeds) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\python310\lib\site-packages (fr
om matplotlib->qeds) (1.0.5)
Requirement already satisfied: et-xmlfile in c:\python310\lib\site-packages (from ope
npyxl->qeds) (1.1.0)
Requirement already satisfied: lxml in c:\python310\lib\site-packages (from pandas_da
tareader->qeds) (4.9.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\python310\lib\site-package
s (from requests->qeds) (1.26.12)
Requirement already satisfied: idna<4,>=2.5 in c:\python310\lib\site-packages (from r
equests->qeds) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\python310\lib\site-pack
ages (from requests->qeds) (2.1.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\python310\lib\site-packages (fro
m plotly->qeds) (8.1.0)
Requirement already satisfied: inflection>=0.3.1 in c:\python310\lib\site-packages (f
rom quandl->qeds) (0.5.1)
Requirement already satisfied: more-itertools in c:\python310\lib\site-packages (from
quandl->qeds) (9.0.0)
Requirement already satisfied: numba in c:\python310\lib\site-packages (from quanteco
n->qeds) (0.56.4)
Requirement already satisfied: sympy in c:\python310\lib\site-packages (from quanteco
n->qeds) (1.11.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\python310\lib\site-packages
(from scikit-learn->qeds) (3.1.0)
Requirement already satisfied: patsy>=0.5.2 in c:\python310\lib\site-packages (from s
tatsmodels->qeds) (0.5.3)
```

```
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in c:\python310\lib\site-pa
ckages (from numba->quantecon->qeds) (0.39.1)
Requirement already satisfied: mpmath>=0.19 in c:\python310\lib\site-packages (from s
ympy->quantecon->qeds) (1.2.1)
```

In [3]:
```
import pandas as pd
import numpy as np
#from tqdm import tqdm
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import geopandas
```

Importa la base de datos

In [4]:
```
url="https://raw.githubusercontent.com/marypazrf/bdd/main/target-locations.csv"
df=pd.read_csv(url)
```

Exploremos los datos.

In [5]:
```
df.head()
```

Out[5]:

	name	latitude	longitude	address	phone	website
0	Alabaster	33.224225	-86.804174	250 S Colonial Dr, Alabaster, AL 35007-4657	205-564-2608	https://www.target.com/sl/alabaster/2276
1	Bessemer	33.334550	-86.989778	4889 Promenade Pkwy, Bessemer, AL 35022-7305	205-565-3760	https://www.target.com/sl/bessemer/2375
2	Daphne	30.602875	-87.895932	1698 US Highway 98, Daphne, AL 36526-4252	251-621-3540	https://www.target.com/sl/daphne/1274
3	Decatur	34.560148	-86.971559	1235 Point Mallard Pkwy SE, Decatur, AL 35601-...	256-898-3036	https://www.target.com/sl/decatur/2084
4	Dothan	31.266061	-85.446422	4601 Montgomery Hwy, Dothan, AL 36303-1522	334-340-1112	https://www.target.com/sl/dothan/1468

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1839 entries, 0 to 1838
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   name       1839 non-null   object
 1   latitude   1839 non-null   float64
 2   longitude  1839 non-null   float64
 3   address    1839 non-null   object
 4   phone      1839 non-null   object
 5   website    1839 non-null   object
dtypes: float64(2), object(4)
memory usage: 86.3+ KB
```

Definición de Latitud y Longitud

Latitud Es la distancia en grados, minutos y segundos que hay con respecto al paralelo principal, que es el ecuador (0º). La latitud puede ser norte y sur.

Longitud: Es la distancia en grados, minutos y segundos que hay con respecto al meridiano principal, que es el meridiano de Greenwich (0º).La longitud puede ser este y oeste.

In [7]:
```
latlong=df[["latitude","longitude"]]
```

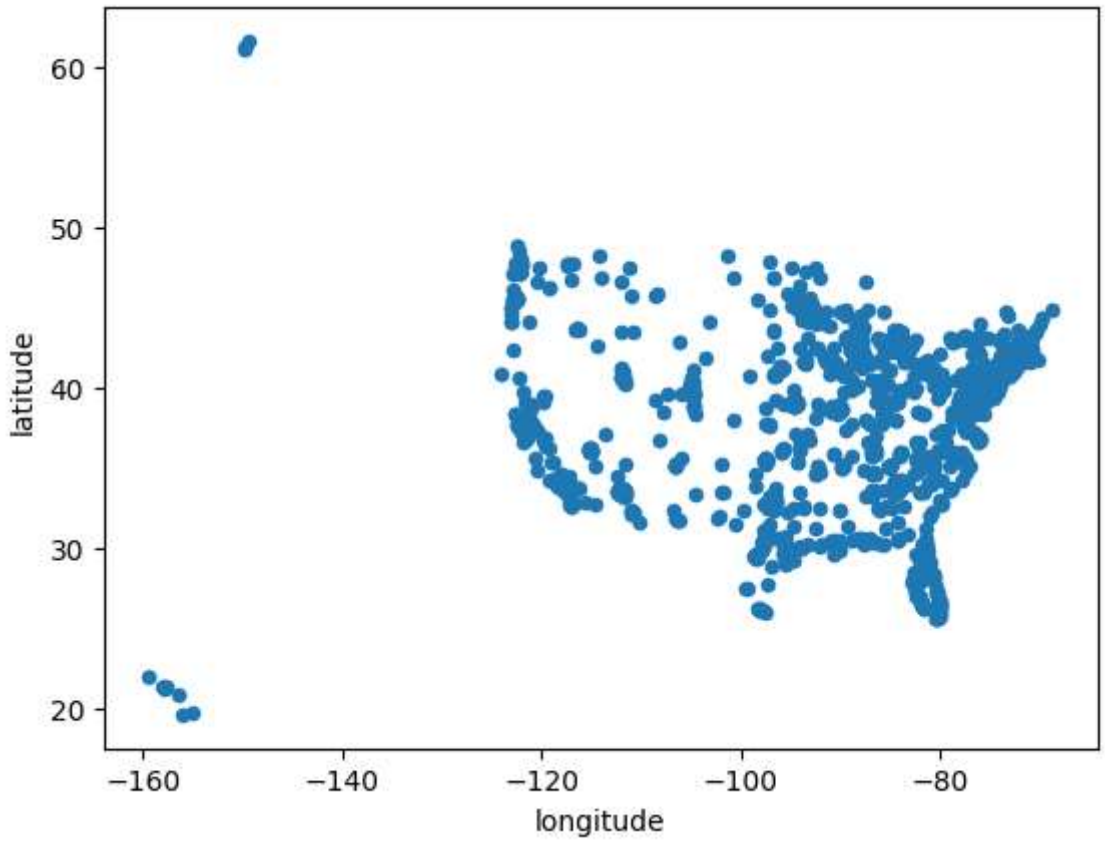
In [ ]:
```
```

¡Visualizemos los datos!, para empezar a notar algún patron.

A simple vista pudieramos pensar que tenemos algunos datos atípicos u outliers, pero .... no es así, simplemente esta grafica no nos está dando toda la información.

In [8]:
```
#extrae los datos interesantes
latlong.plot.scatter( "longitude","latitude")
```

Out[8]:
```
<AxesSubplot: xlabel='longitude', ylabel='latitude'>
```



In [9]:
```
latlong.describe()
```

Out[9]:

	latitude	longitude
count	1839.000000	1839.000000
mean	37.791238	-91.986881
std	5.272299	16.108046
min	19.647855	-159.376962
25%	33.882605	-98.268828
50%	38.955432	-87.746346
75%	41.658341	-80.084833
max	61.577919	-68.742331

Para entender un poco más, nos auxiliaremos de una librería para graficar datos geográficos. Esto nos ayudara a tener un mejor entendimiento de ellos.

In [10]:
```
import geopandas as gpd
import matplotlib.pyplot as plt
import pandas as pd

from shapely.geometry import Point

%matplotlib inline
# activate plot theme
```

```
import qeds
qeds.themes.mpl_style();
```

In [11]:
```
df["Coordinates"] = list(zip(df.longitude, df.latitude))
df["Coordinates"] = df["Coordinates"].apply(Point)
df.head()
```

Out[11]:

	name	latitude	longitude	address	phone	website	
0	Alabaster	33.224225	-86.804174	250 S Colonial Dr, Alabaster, AL 35007-4657	205-564-2608	https://www.target.com/sl/alabaster/2276	(-86.
1	Bessemer	33.334550	-86.989778	4889 Promenade Pkwy, Bessemer, AL 35022-7305	205-565-3760	https://www.target.com/sl/bessemer/2375	(-86.
2	Daphne	30.602875	-87.895932	1698 US Highway 98, Daphne, AL 36526-4252	251-621-3540	https://www.target.com/sl/daphne/1274	(-87.
3	Decatur	34.560148	-86.971559	1235 Point Mallard Pkwy SE, Decatur, AL 35601-...	256-898-3036	https://www.target.com/sl/decatur/2084	PO
4	Dothan	31.266061	-85.446422	4601 Montgomery Hwy, Dothan, AL 36303-1522	334-340-1112	https://www.target.com/sl/dothan/1468	PO

In [12]:
```
gdf = gpd.GeoDataFrame(df, geometry="Coordinates")
gdf.head()
```

Out[12]:

	name	latitude	longitude	address	phone	website	Coor
0	Alabaster	33.224225	-86.804174	250 S Colonial Dr, Alabaster, AL 35007-4657	205-564-2608	https://www.target.com/sl/alabaster/2276	(-8 3
1	Bessemer	33.334550	-86.989778	4889 Promenade Pkwy, Bessemer, AL 35022-7305	205-565-3760	https://www.target.com/sl/bessemer/2375	(-8 3
2	Daphne	30.602875	-87.895932	1698 US Highway 98, Daphne, AL 36526-4252	251-621-3540	https://www.target.com/sl/daphne/1274	(-8 3(
3	Decatur	34.560148	-86.971559	1235 Point Mallard Pkwy SE, Decatur, AL 35601-...	256-898-3036	https://www.target.com/sl/decatur/2084	(-8 3
4	Dothan	31.266061	-85.446422	4601 Montgomery Hwy, Dothan, AL 36303-1522	334-340-1112	https://www.target.com/sl/dothan/1468	(-8 3

In [13]:
```
#mapa

world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))
world = world.set_index("iso_a3")

world.head()
```

Out[13]:

iso_a3	pop_est	continent	name	gdp_md_est	geometry
FJI	889953.0	Oceania	Fiji	5496	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...
TZA	58005463.0	Africa	Tanzania	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...
ESH	603253.0	Africa	W. Sahara	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...
CAN	37589262.0	North America	Canada	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...
USA	328239523.0	North America	United States of America	21433226	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...

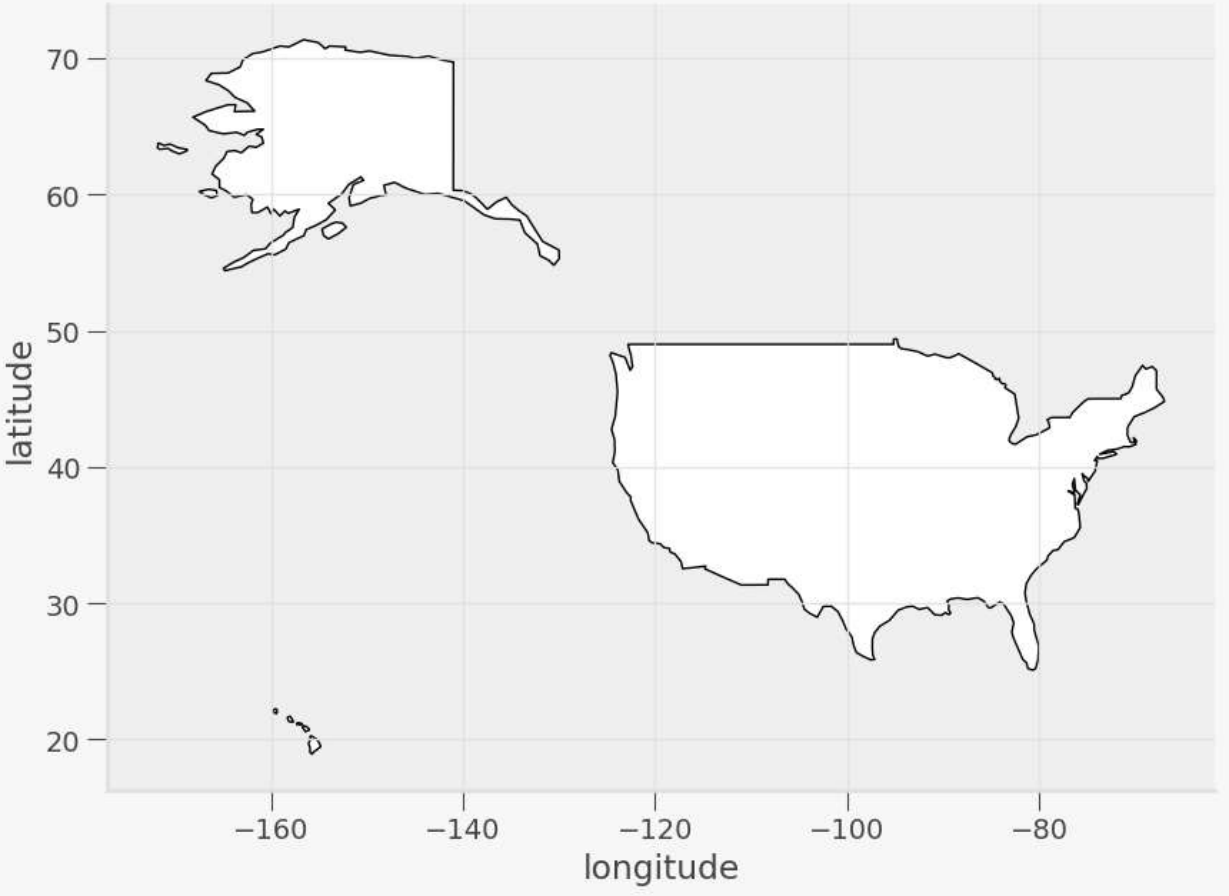
In [14]:
```
#graficar el mapa
world.name.unique()
```

Out[14]:
```
array(['Fiji', 'Tanzania', 'W. Sahara', 'Canada',
       'United States of America', 'Kazakhstan', 'Uzbekistan',
       'Papua New Guinea', 'Indonesia', 'Argentina', 'Chile',
       'Dem. Rep. Congo', 'Somalia', 'Kenya', 'Sudan', 'Chad', 'Haiti',
       'Dominican Rep.', 'Russia', 'Bahamas', 'Falkland Is.', 'Norway',
       'Greenland', 'Fr. S. Antarctic Lands', 'Timor-Leste',
       'South Africa', 'Lesotho', 'Mexico', 'Uruguay', 'Brazil',
       'Bolivia', 'Peru', 'Colombia', 'Panama', 'Costa Rica', 'Nicaragua',
       'Honduras', 'El Salvador', 'Guatemala', 'Belize', 'Venezuela',
       'Guyana', 'Suriname', 'France', 'Ecuador', 'Puerto Rico',
       'Jamaica', 'Cuba', 'Zimbabwe', 'Botswana', 'Namibia', 'Senegal',
       'Mali', 'Mauritania', 'Benin', 'Niger', 'Nigeria', 'Cameroon',
       'Togo', 'Ghana', "Côte d'Ivoire", 'Guinea', 'Guinea-Bissau',
       'Liberia', 'Sierra Leone', 'Burkina Faso', 'Central African Rep.',
       'Congo', 'Gabon', 'Eq. Guinea', 'Zambia', 'Malawi', 'Mozambique',
       'eSwatini', 'Angola', 'Burundi', 'Israel', 'Lebanon', 'Madagascar',
       'Palestine', 'Gambia', 'Tunisia', 'Algeria', 'Jordan',
       'United Arab Emirates', 'Qatar', 'Kuwait', 'Iraq', 'Oman',
       'Vanuatu', 'Cambodia', 'Thailand', 'Laos', 'Myanmar', 'Vietnam',
       'North Korea', 'South Korea', 'Mongolia', 'India', 'Bangladesh',
       'Bhutan', 'Nepal', 'Pakistan', 'Afghanistan', 'Tajikistan',
       'Kyrgyzstan', 'Turkmenistan', 'Iran', 'Syria', 'Armenia', 'Sweden',
       'Belarus', 'Ukraine', 'Poland', 'Austria', 'Hungary', 'Moldova',
       'Romania', 'Lithuania', 'Latvia', 'Estonia', 'Germany', 'Bulgaria',
       'Greece', 'Turkey', 'Albania', 'Croatia', 'Switzerland',
       'Luxembourg', 'Belgium', 'Netherlands', 'Portugal', 'Spain',
       'Ireland', 'New Caledonia', 'Solomon Is.', 'New Zealand',
       'Australia', 'Sri Lanka', 'China', 'Taiwan', 'Italy', 'Denmark',
       'United Kingdom', 'Iceland', 'Azerbaijan', 'Georgia',
       'Philippines', 'Malaysia', 'Brunei', 'Slovenia', 'Finland',
       'Slovakia', 'Czechia', 'Eritrea', 'Japan', 'Paraguay', 'Yemen',
       'Saudi Arabia', 'Antarctica', 'N. Cyprus', 'Cyprus', 'Morocco',
       'Egypt', 'Libya', 'Ethiopia', 'Djibouti', 'Somaliland', 'Uganda',
       'Rwanda', 'Bosnia and Herz.', 'North Macedonia', 'Serbia',
       'Montenegro', 'Kosovo', 'Trinidad and Tobago', 'S. Sudan'],
      dtype=object)
```

In [15]:
```
fig, gax = plt.subplots(figsize=(10,10))

# By only plotting rows in which the continent is 'South America' we only plot SA.
world.query("name == 'United States of America'").plot(ax=gax, edgecolor='black',color

# By the way, if you haven't read the book 'longitude' by Dava Sobel, you should...
gax.set_xlabel('longitude')
gax.set_ylabel('latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```

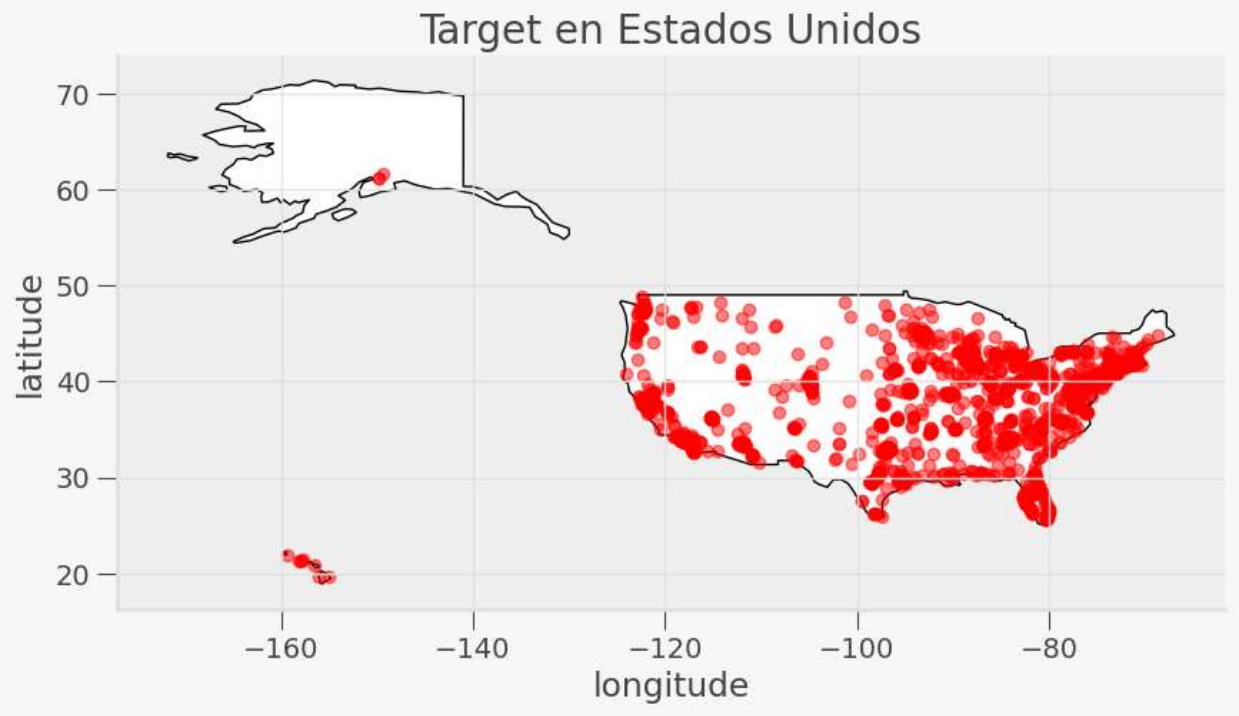


In [16]:

```
# Step 3: Plot the cities onto the map
# We mostly use the code from before --- we still want the country borders plotted ---
# add a command to plot the cities
fig, gax = plt.subplots(figsize=(10,10))

# By only plotting rows in which the continent is 'South America' we only plot, well,
# South America.
world.query("name == 'United States of America'").plot(ax = gax, edgecolor='black', co

# This plot the cities. It's the same syntax, but we are plotting from a different Geo
# I want the cities as pale red dots.
gdf.plot(ax=gax, color='red', alpha = 0.5)

gax.set_xlabel('longitude')
gax.set_ylabel('latitude')
gax.set_title('Target en Estados Unidos')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```



¿qué tal ahora?, tiene mayor sentido verdad, entonces los datos lejanos no eran atípicos, de aquí la importancia de ver los datos con el tipo de gráfica correcta.

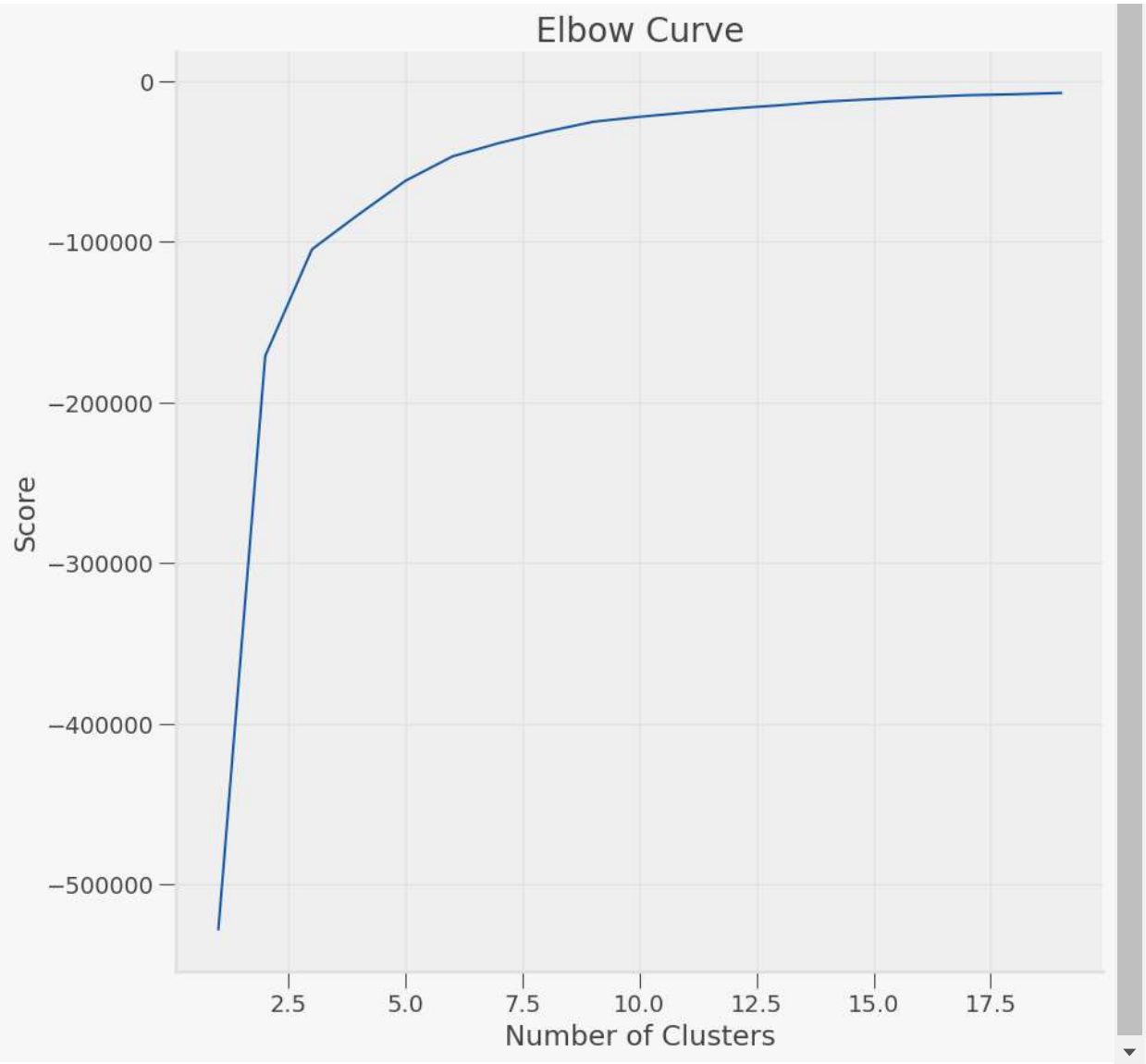
Ahora sí, implementa K means a los datos de latitud y longitud :) y encuentra donde colocar los almacenes.

Nota: si te llama la atención implementar alguna otra visualización con otra librería, lo puedes hacer, no hay restricciones.

In [19]:
```
#tu codigo aquí

from sklearn.cluster import KMeans

X = df[["longitude","latitude"]]
#se usa la grafica elbow para estqblecer cuantos clusteres se van a implementqt con  K
Nc = range(1, 20)
kmeans = [KMeans(n_clusters=i) for i in Nc]
kmeans
score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]
score
plt.subplots(figsize=(10,10))
plt.plot(Nc,score)
plt.xlabel('Number of Clusters')
plt.ylabel('Score')
```

```python
plt.title('Elbow Curve')
plt.show()
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```



In [20]:
```
#de acuerdo a la grafica se definen 3 clusteres como optimos
kmeans = KMeans(n_clusters=3).fit(X)
centroids = kmeans.cluster_centers_
labels = kmeans.predict(X)
# se obirnr los centros de los almacenes(Clusters)
C = kmeans.cluster_centers_

C_DF = pd.DataFrame(C)
C_DF["Coordinates"] = list(zip(C_DF[0], C_DF[1]))
C_DF["Coordinates"] = C_DF["Coordinates"].apply(Point)


gdf_C = gpd.GeoDataFrame(C_DF, geometry="Coordinates")
gdf_C
```

Out[20]:

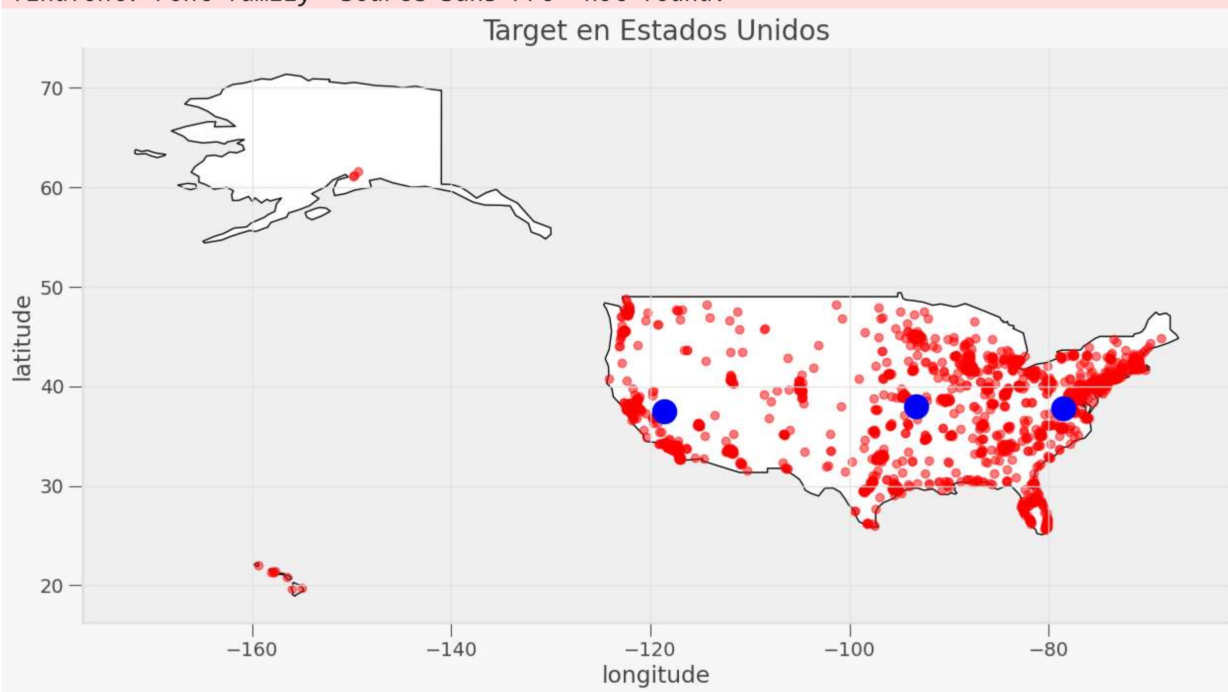
	0	1	Coordinates
0	-78.569908	37.789554	POINT (-78.56991 37.78955)
1	-93.327172	37.980063	POINT (-93.32717 37.98006)
2	-118.624473	37.487342	POINT (-118.62447 37.48734)

In [22]:

```
#se grafica la posicion de los almacenes requeridos:
fig, gax = plt.subplots(figsize=(15,10))

world.query("name == 'United States of America'").plot(ax = gax, edgecolor='black', co

gdf.plot(ax=gax, color='red', alpha = 0.5)
gdf_C.plot(ax=gax, color='blue', alpha = 1, markersize = 300)

gax.set_xlabel('longitude')
gax.set_ylabel('latitude')
gax.set_title('Target en Estados Unidos')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```

```
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
findfont: Font family 'Source Sans Pro' not found.
```



In [25]:
```
#tiendas atendidas por almacen
latlong['kmeans'] = kmeans.labels_
latlong.loc[:, 'kmeans'].value_counts()
```

```
C:\Users\joshi\AppData\Local\Temp\ipykernel_12368\772880896.py:2: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  latlong['kmeans'] = kmeans.labels_
```

Out[25]:
```
0    826
1    628
2    385
Name: kmeans, dtype: int64
```

In [28]:
```
from pandas.core.internals.concat import concat_arrays

Location1 = str(gdf_C[1][0]) + ", " + str(gdf_C[0][0])
print(Location1)
Location2 = str(gdf_C[1][1]) + ", " + str(gdf_C[0][1])
print(Location2)
Location3 = str(gdf_C[1][2]) + ", " + str(gdf_C[0][2])
print(Location3)
```

```
37.789554004474006, -78.56990807484885
37.98006260590112, -93.32717230430622
37.48734203064935, -118.62447331844156
```

- Encuentra el numero ideal de almacenes, justifica tu respuesta:

El grafico de codonos da el mejor resultado minimizando el error cuadrado de estas interacciones el cual es de 3 almacenes.

- Encuentra las latitudes y longitudes de los almacenes, ¿qué ciudad es? 1 almacen : 1076 James River Rd, Scottsville, VA 24590, EE. UU. 2 almacen: Municipio de Center, Misuri 65668, EE. UU. 3 almacen: Round Valley Joint Elementary, California 93514, EE. UU.

- ¿a cuantas tiendas va surtir?

1 almacen: 826 tiendas 2 alamacen 628 tiendas 3er almacen 385 tiendas

¿sabes a que distancia estará? distancia 1 y 2 almacenes : 981 millas distancia 2 y 3almacenes: 1695 millas

- ¿Cómo elegiste el número de almacenes?, justifica tu respuesta técnicamente. El grafico de codo nos da el mejor resultado minimizando el error cuadrado de estas interacciones.

Adicionalmente, en el notebook notaras que al inicio exploramos los datos y los graficamos de manera simple, después nos auxiliamos de una librería de datos geográficos.

- ¿qué librerías nos pueden ayudar a graficar este tipo de datos?

  seaborn es una gran herramienta para poder graficar diferentes tipos de figuras y geopandas hace su trabajo para el desarrollo directo en mapas.

- ¿Consideras importante que se grafique en un mapa?, ¿por qué? para el caso de estos almacenes claro para poder ver la distribucion correcta y entender de mejor manera el como se estan distribuyendo estos almacenes.

- Agrega las conclusiones la grafica de codo es fundamental para poder etablecer la K optima para el proceso de kmedias. se puede visualizar de manera correcta la distribucion al ponerlo en un mapa. al obtener la informacion deseada se puede presentar una estrategia para ubicar de manera correcta estos almacenes.

In [ ]:
```
```