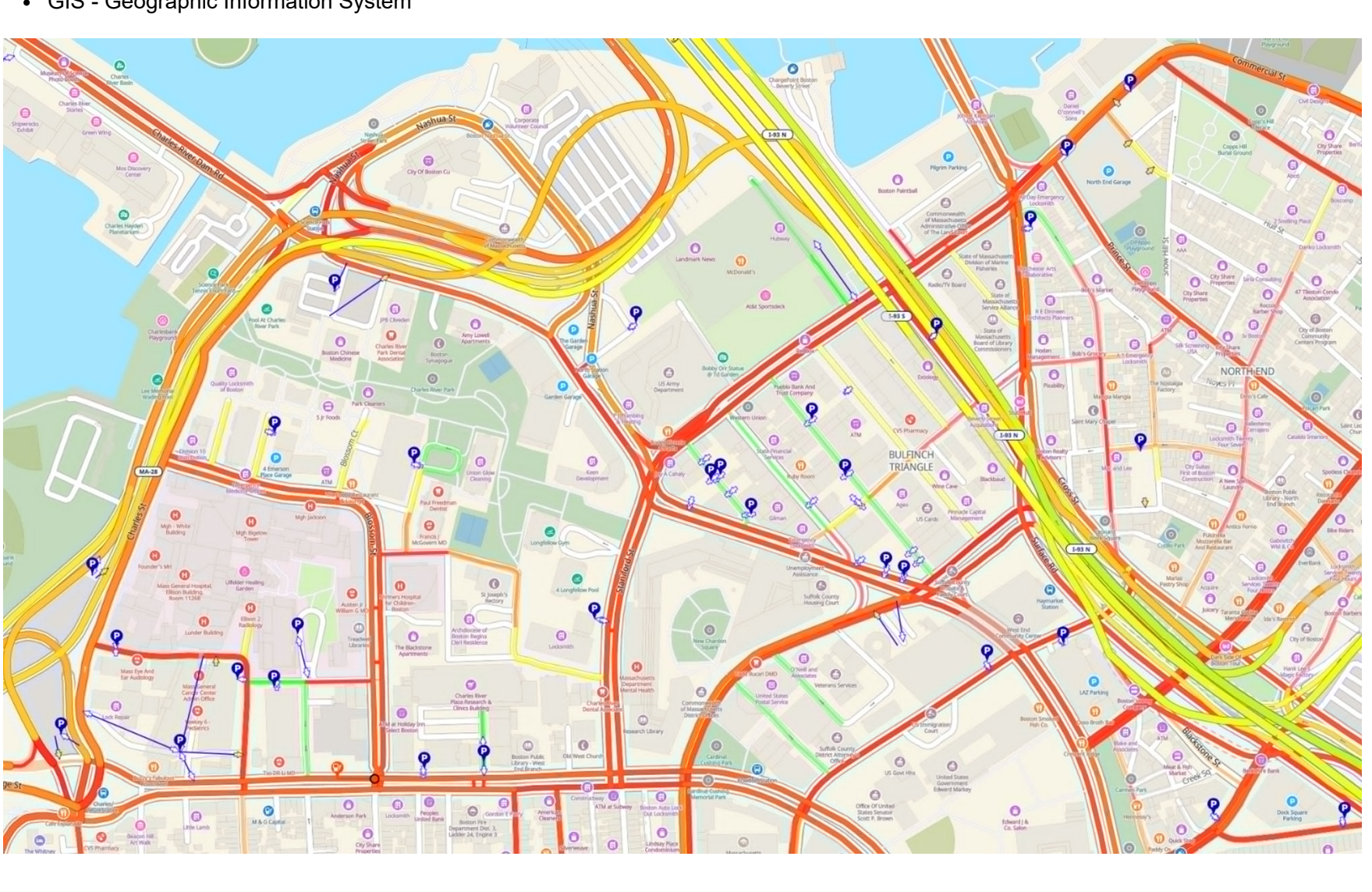
Today's agenda

- Location analytics
  - Real-time geographic DA
  - Historical geographic DA
- Important fields where LA is used
- Python pacakges for Geospatial visualization
- Earthquake data preparation
- Earthquake data visualization

In [ ]:

Location Analytics - Geospatial Visualization

- The ability to gain insights from the location or geographic component of business data.
- The important component is the location data.
- GIS - Geographic Information System



Credits - Image from Internet

In [ ]:

Real-time Geographic Data Analytics

Getting insights from the data that comes into the system and relating that to a particluar location is called Real-time Geographic data analytics.

- Getting route navigation in Google Maps
- Courier and postal services
- Military serives
  - Getting the exact location of the enemy movements on the map to get informed

Credits - Image from Internet

In [ ]:

Historic Geographic Data Analytics

Getting insights from the historic data so as to predict the furture occurrence or disaster (purely by chance) completely based on the geographic location.

- Predict and prevent (taking necessary steps) the occurrence based on the past data
- Disaster prevention efforts
  - Floods
  - Volcanoes
  - Earthquakes

- Chain retailers benefit from the historical spending habits based on location to increase the sales

Credits - Image from Internet

In [ ]:

Important fields/topics

Location analytics (Geospatial science) can be used in every field/study where data is generated with respect to the location.

Credits - Image from Internet

- Demographic study - statistical study of population of all the areas based on location
  - Analysis based on education details
  - Analysis based on ethnicity
  - Analysis based on religion
  - Analysis based on births and deaths
  - ...
- Measurement of **Natural Occurrences** that happen on timely basis
- Increase or decrease of pollution in a particular city or country
- Geographic crime analysis
  - Analyse the crime data
  - Identify the crime types and hotspots
  - Help law enforcement plan for crime prevention
- ETC

In [ ]:

Python packages for Geospatial Visualization

- GeoViews → explore & visualize geographical, meteorological, and oceanographic datasets
- Folium → widely used geospatial data visualization library built on Leaflet.js framework
- Plotly → Interactive map visualization - effectively uses Mapbox api
- KeplerGL → built exclusively for jupyter notebook to visualize big geospatial data
- GeoPandas → workhorse for working geo-data
- geonamescache → used to retrieve location datasets in the form of python dictionaries

In [ ]:

`import` packages

In [1]:
```
import pandas as pd
import plotly.graph_objects as go
```

If you do not have the above packages, you can install by typing these commands on `Command Prompt` (CMD) -

- `pip install pandas --user`
- `pip install plotly --user`

Dataset description

Earthquake data (from Yesterday) - The data is obtained from `USGS` datasources. The data is updated every 1 minute. In this example we don't deal with streaming data.

- time
- latitude
- longitude
- mag (magnitude)
- place

In [2]:
```
eqdf = pd.read_csv('all_day.csv')
eqdf.shape
```

Out[2]: (253, 22)

In [3]:
```
eqdf.head()
```

Out[3]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	ty
0	2021-03-17T12:38:06.050Z	19.506332	-155.661331	6.04	2.42	ml	19.0	93.0	NaN	0.20	...	2021-03-17T12:43:36.970Z	22 km ENE of Honaunau-Napoopoo, Hawaii	earthqua
1	2021-03-17T12:32:16.628Z	54.570100	163.056500	10.00	4.90	mb	NaN	106.0	3.03800	0.79	...	2021-03-17T12:50:40.040Z	188 km S of Ust'-Kamchatsk Staryy, Russia	earthqua
2	2021-03-17T11:53:05.980Z	37.445999	-121.785164	3.34	1.47	md	19.0	63.0	0.01286	0.05	...	2021-03-17T12:08:06.410Z	10km NNE of Alum Rock, CA	earthqua
3	2021-03-17T11:36:27.640Z	47.772999	-120.086334	-0.57	2.50	md	12.0	54.0	NaN	0.07	...	2021-03-17T11:39:38.650Z	8 km WSW of Chelan Falls, Washington	earthqua
4	2021-03-17T11:23:20.740Z	35.984167	-117.438000	0.50	0.94	ml	12.0	212.0	0.08448	0.13	...	2021-03-17T11:27:06.820Z	24km N of Searles Valley, CA	earthqua

5 rows × 22 columns

In [4]:
```
print(eqdf.columns)
```

```
Index(['time', 'latitude', 'longitude', 'depth', 'mag', 'magType', 'nst',
       'gap', 'dmin', 'rms', 'net', 'id', 'updated', 'place', 'type',
       'horizontalError', 'depthError', 'magError', 'magNst', 'status',
       'locationSource', 'magSource'],
      dtype='object')
```

In [5]:
```
# take subset data (time, latitude, longitude, mag, place)
eqdf = eqdf[['time', 'latitude', 'longitude', 'mag', 'place']]
```

In [6]:
```
# head()
eqdf.head()
```

Out[6]:

	time	latitude	longitude	mag	place
0	2021-03-17T12:38:06.050Z	19.506332	-155.661331	2.42	22 km ENE of Honaunau-Napoopoo, Hawaii
1	2021-03-17T12:32:16.628Z	54.570100	163.056500	4.90	188 km S of Ust'-Kamchatsk Staryy, Russia
2	2021-03-17T11:53:05.980Z	37.445999	-121.785164	1.47	10km NNE of Alum Rock, CA
3	2021-03-17T11:36:27.640Z	47.772999	-120.086334	2.50	8 km WSW of Chelan Falls, Washington
4	2021-03-17T11:23:20.740Z	35.984167	-117.438000	0.94	24km N of Searles Valley, CA

In [7]:
```
# shape
eqdf.shape
```

Out[7]: (253, 5)

In [8]:
```
# take "place" separately and display head
eqdf['place'].head()
```

Out[8]:
```
0       22 km ENE of Honaunau-Napoopoo, Hawaii
1    188 km S of Ust'-Kamchatsk Staryy, Russia
2                   10km NNE of Alum Rock, CA
3         8 km WSW of Chelan Falls, Washington
4                24km N of Searles Valley, CA
Name: place, dtype: object
```

In [ ]:

Data Preperation

In [9]:
```
# example
dp = '3 km ENE of Pāhala, Hawaii'
sa, a = dp.split(', ')
print(sa)
print(a)
```

```
3 km ENE of Pāhala
Hawaii
```

Separate `place` by → ,

In [10]:
```
# area_list → split
area_list = eqdf['place'].str.split(', ').to_list()
```

Print first `5` from `area_list`

In [11]:
```
area_list[:5]
```

Out[11]:
```
[['22 km ENE of Honaunau-Napoopoo', 'Hawaii'],
 ['188 km S of Ust'-Kamchatsk Staryy', 'Russia'],
 ['10km NNE of Alum Rock', 'CA'],
 ['8 km WSW of Chelan Falls', 'Washington'],
 ['24km N of Searles Valley', 'CA']]
```

Make two columns `area` and `sub_area` from area_list

In [12]:
```
area = []
sub_area = []

for al in area_list:
    if len(al) == 2:
        area.append(al[1])
        sub_area.append(al[0])
    else:
        area.append(al[0])
        sub_area.append(al[0])
```

Print first `5` from `area`

In [13]:
```
area[:5]
```

Out[13]: ['Hawaii', 'Russia', 'CA', 'Washington', 'CA']

Print first `5` from `sub_area`

In [14]:
```
sub_area[:5]
```

Out[14]:
```
['22 km ENE of Honaunau-Napoopoo',
 '188 km S of Ust'-Kamchatsk Staryy',
 '10km NNE of Alum Rock',
 '8 km WSW of Chelan Falls',
 '24km N of Searles Valley']
```

Add `area` and `sub_area` as columns in `eqdf`

In [15]:
```
eqdf['sub_area'] = sub_area
eqdf['area'] = area
```

In [16]:
```
# head()
eqdf.head()
```

Out[16]:

	time	latitude	longitude	mag	place	sub_area	area
0	2021-03-17T12:38:06.050Z	19.506332	-155.661331	2.42	22 km ENE of Honaunau-Napoopoo, Hawaii	22 km ENE of Honaunau-Napoopoo	Hawaii
1	2021-03-17T12:32:16.628Z	54.570100	163.056500	4.90	188 km S of Ust'-Kamchatsk Staryy, Russia	188 km S of Ust'-Kamchatsk Staryy	Russia
2	2021-03-17T11:53:05.980Z	37.445999	-121.785164	1.47	10km NNE of Alum Rock, CA	10km NNE of Alum Rock	CA
3	2021-03-17T11:36:27.640Z	47.772999	-120.086334	2.50	8 km WSW of Chelan Falls, Washington	8 km WSW of Chelan Falls	Washington
4	2021-03-17T11:23:20.740Z	35.984167	-117.438000	0.94	24km N of Searles Valley, CA	24km N of Searles Valley	CA

In [17]:
```
# shape
eqdf.shape
```

Out[17]: (253, 7)

Remove the column `place` from `eqdf`

In [18]:
```
# prep_df
prep_df = eqdf.drop(columns=['place'], axis=1)
```

In [19]:
```
# head()
prep_df.head()
```

Out[19]:

	time	latitude	longitude	mag	sub_area	area
0	2021-03-17T12:38:06.050Z	19.506332	-155.661331	2.42	22 km ENE of Honaunau-Napoopoo	Hawaii
1	2021-03-17T12:32:16.628Z	54.570100	163.056500	4.90	188 km S of Ust'-Kamchatsk Staryy	Russia
2	2021-03-17T11:53:05.980Z	37.445999	-121.785164	1.47	10km NNE of Alum Rock	CA
3	2021-03-17T11:36:27.640Z	47.772999	-120.086334	2.50	8 km WSW of Chelan Falls	Washington
4	2021-03-17T11:23:20.740Z	35.984167	-117.438000	0.94	24km N of Searles Valley	CA

Consider the data where magnitude is greater than or equal to `2`

In [20]:
```
# shape
prep_df.shape
```

Out[20]: (253, 6)

In [21]:
```
# prep_df
prep_df = prep_df[prep_df['mag'] >= 2]
```

In [22]:
```
# shape
prep_df.shape
```

Out[22]: (77, 6)

In [ ]:

Data Visualization

Check the frequency of occurrence by `area`

In [23]:
```
# occ_freq → value_counts().to_frame()
occ_freq = prep_df['area'].value_counts().to_frame()
```

In [24]:
```
occ_freq
```

Out[24]:

	area
Alaska	14
Hawaii	13
Russia	12
Puerto Rico	6
CA	5
Idaho	4
New Mexico	3
Fiji region	2
China	2
Papua New Guinea	2
Carlsberg Ridge	1
Greenland Sea	1
western Texas	1
West Chile Rise	1
Philippines	1
Argentina	1
Dominican Republic	1
Puerto Rico region	1
Japan	1
Japan region	1
Washington	1
Kermadec Islands region	1
Kansas	1
Banda Sea	1

default plot will not be so interactive

In [25]:
```
# occ_freq.plot(kind='pie', subplots=True)
```

`Pie` chart with `Plotly`

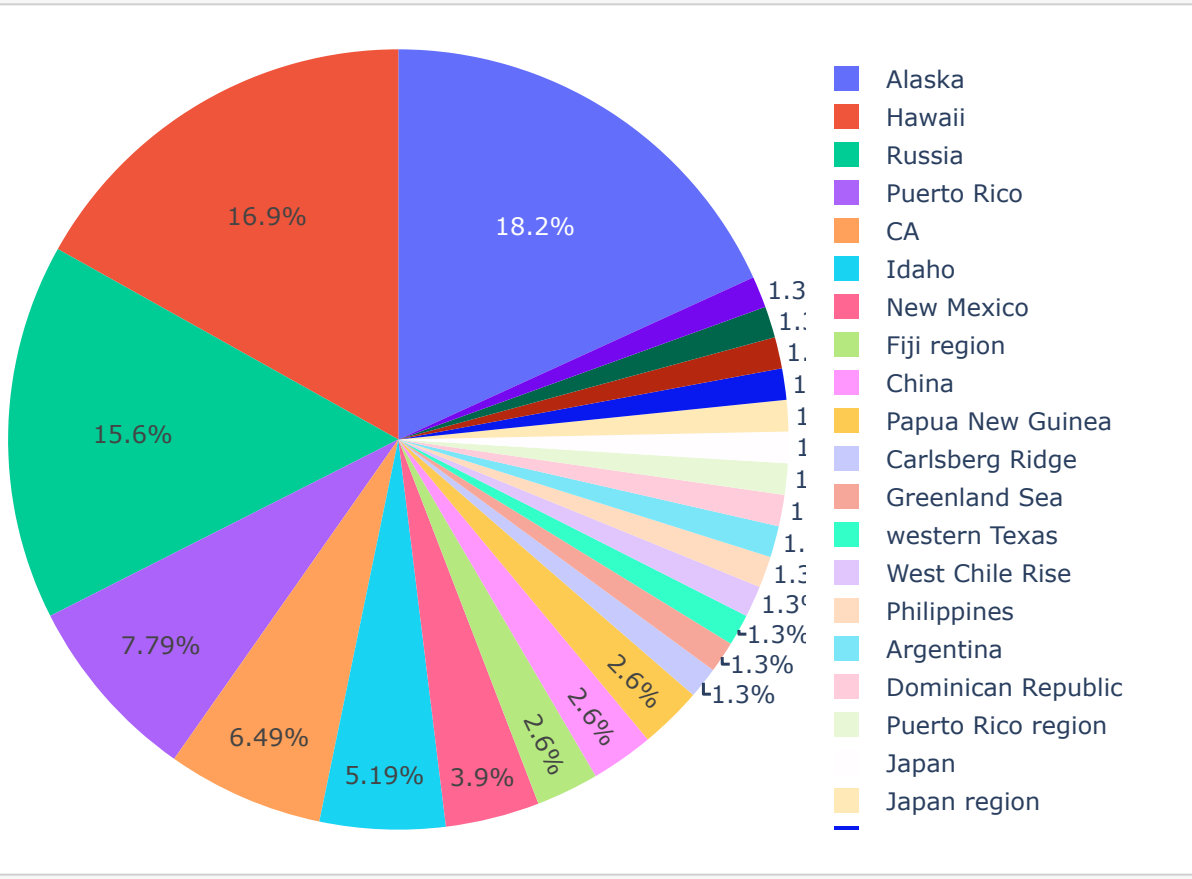
- labels → index
- values → area
- lrtb → 0

refer → https://plotly.com/python/pie-charts/

In [26]:
```
# index
trace = go.Pie(
    labels=occ_freq.index,
    values=occ_freq.area
)

layout = go.Layout(
    height=400,
    width=600,
    margin=dict(l=0, r=0, b=0, t=0)
)

fig = go.Figure(data=[trace], layout=layout)
fig.show()
```



In [ ]:

`Bar` chart of magnitude based on area with `Plotly`

- x → sub_area
- y → mag
- lrtb → 0

refer → https://plotly.com/python/bar-charts/

In [27]:
```
region = 'Hawaii'
# rdf from prep_df

rdf = prep_df[prep_df['area'] == region]
```

In [28]:
```
# head()
rdf.head()
```

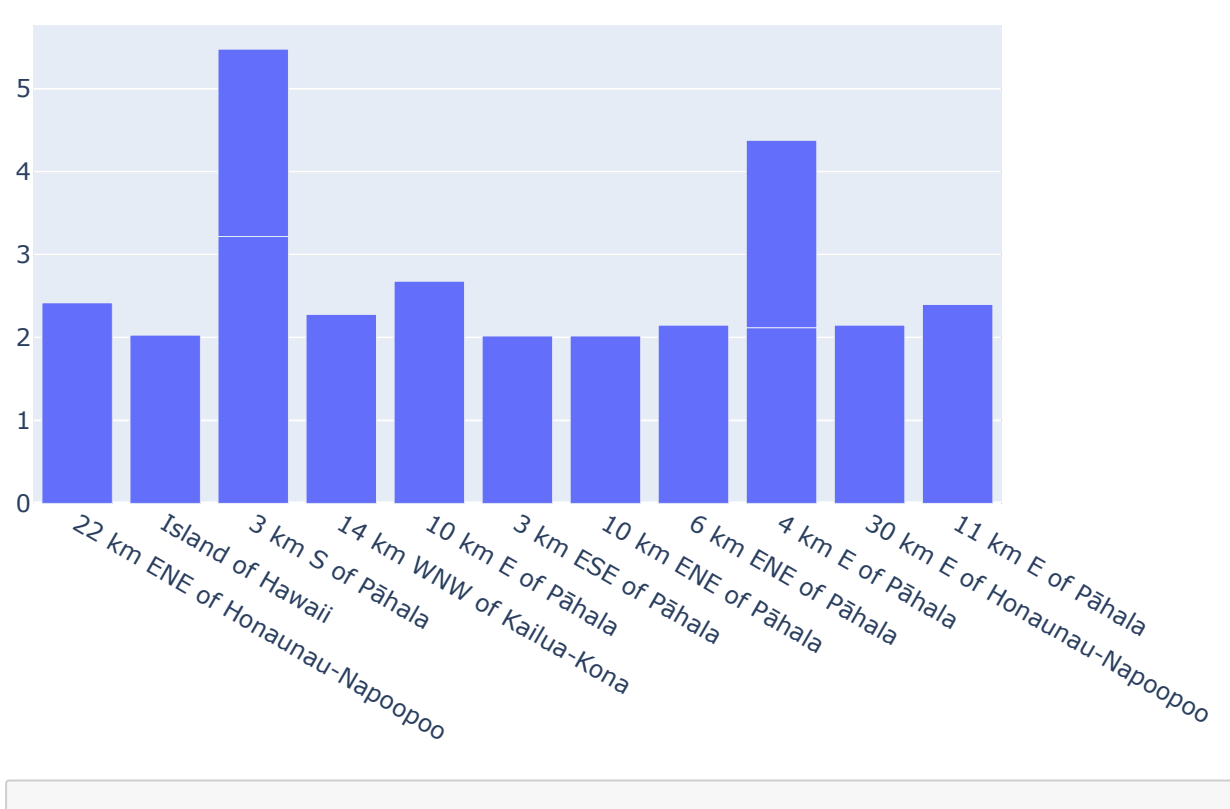
Out[28]:

	time	latitude	longitude	mag	sub_area	area
0	2021-03-17T12:38:06.050Z	19.506332	-155.661331	2.42	22 km ENE of Honaunau-Napoopoo	Hawaii
19	2021-03-17T10:34:23.580Z	19.227667	-155.379501	2.03	Island of Hawaii	Hawaii
21	2021-03-17T10:25:19.990Z	19.172000	-155.475167	3.22	3 km S of Pāhala	Hawaii
31	2021-03-17T09:44:28.220Z	19.668833	-156.138833	2.28	14 km WNW of Kailua-Kona	Hawaii
50	2021-03-17T08:14:51.670Z	19.218334	-155.379166	2.68	10 km E of Pāhala	Hawaii

In [29]:
```
# bar chart
trace = go.Bar(
    x=rdf['sub_area'],
    y=rdf['mag']
)

layout = go.Layout(
    height=400,
    width=600,
    title=region,
    margin=dict(l=0, r=0, b=10, t=40)
)

fig = go.Figure(data=[trace], layout=layout)
fig.show()
```



In [ ]:

Map Visualization

- lat → latitudes
- lon → longitudes
- mode → markers
  - size → 10
  - color → red
- text → sub_area

refer → https://plotly.com/python/scattermapbox/

In [30]:
```
region = 'Alaska'

# filter it
rdf = prep_df[prep_df['area'] == region]
```

In [31]:
```
# lats, lons, text (sub_area)
region_lats = rdf['latitude'].to_list()
region_lons = rdf['longitude'].to_list()
region_text = rdf['sub_area'].to_list()
```

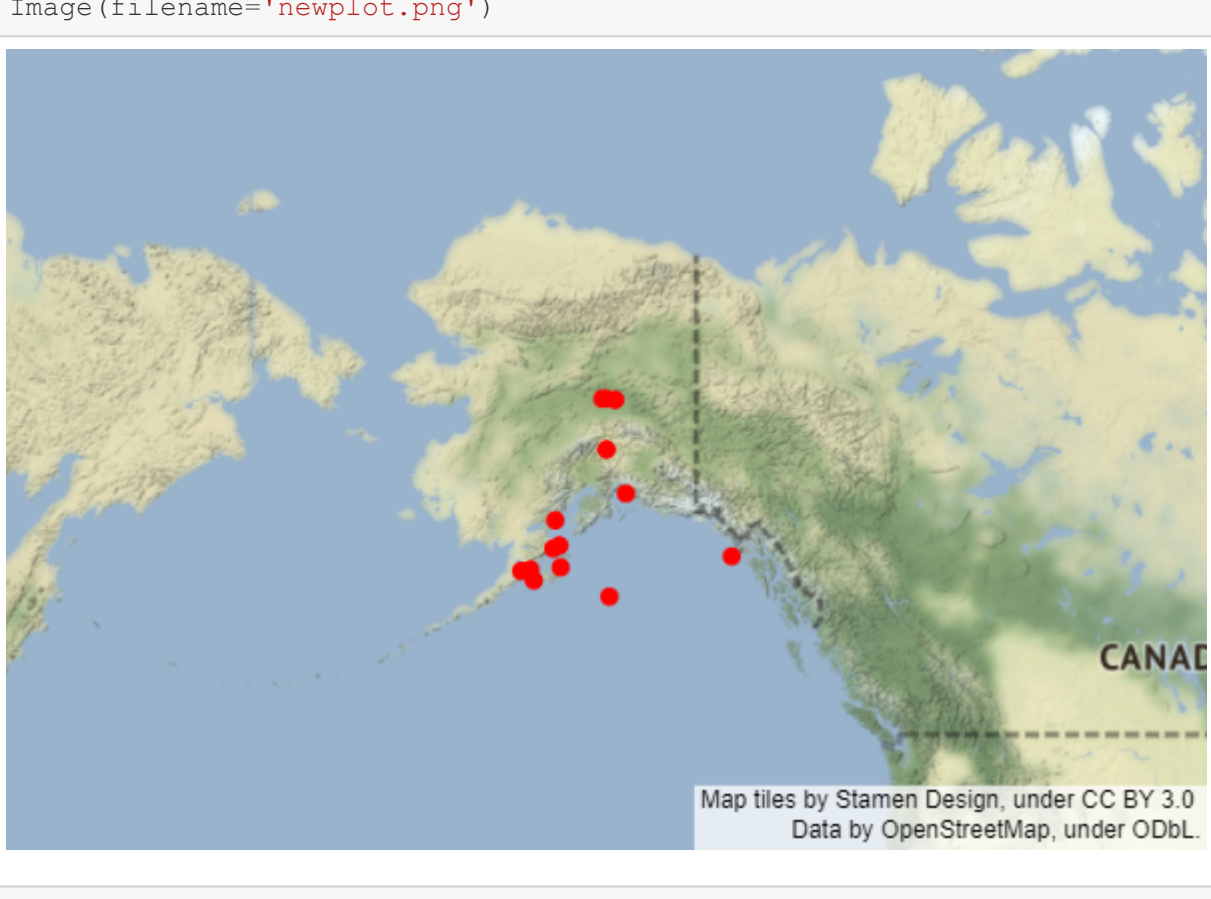
In [32]:
```
trace = go.Scattermapbox(
    lat=region_lats,
    lon=region_lons,
    mode='markers',
    marker=dict(
        size=10,
        color='red'
    ),
    text=region_text,
    hoverinfo='text'
)

layout = go.Layout(
    height=400,
    width=600,
    margin=dict(l=0, r=0, t=0, b=0),
    mapbox_style='stamen-terrain',
    mapbox=dict(
        center=dict(
            lat=region_lats[0],
            lon=region_lons[0]
        ),
        zoom=2
    )
)

# fig = go.Figure(data=[trace], layout=layout)
# fig.show()
```

In [33]:
```
from IPython.display import Image
Image(filename='newplot.png')
```

Out[33]:



In [ ]:

What did we learn?

- Location Analytics
- Why location intelligence is important for the business or the govt agency
- Python libraries that support geospatial visualization
- Geospatial data processing pipeline
  - Data preparation
  - Data cleaning
- Earthquake data visualization