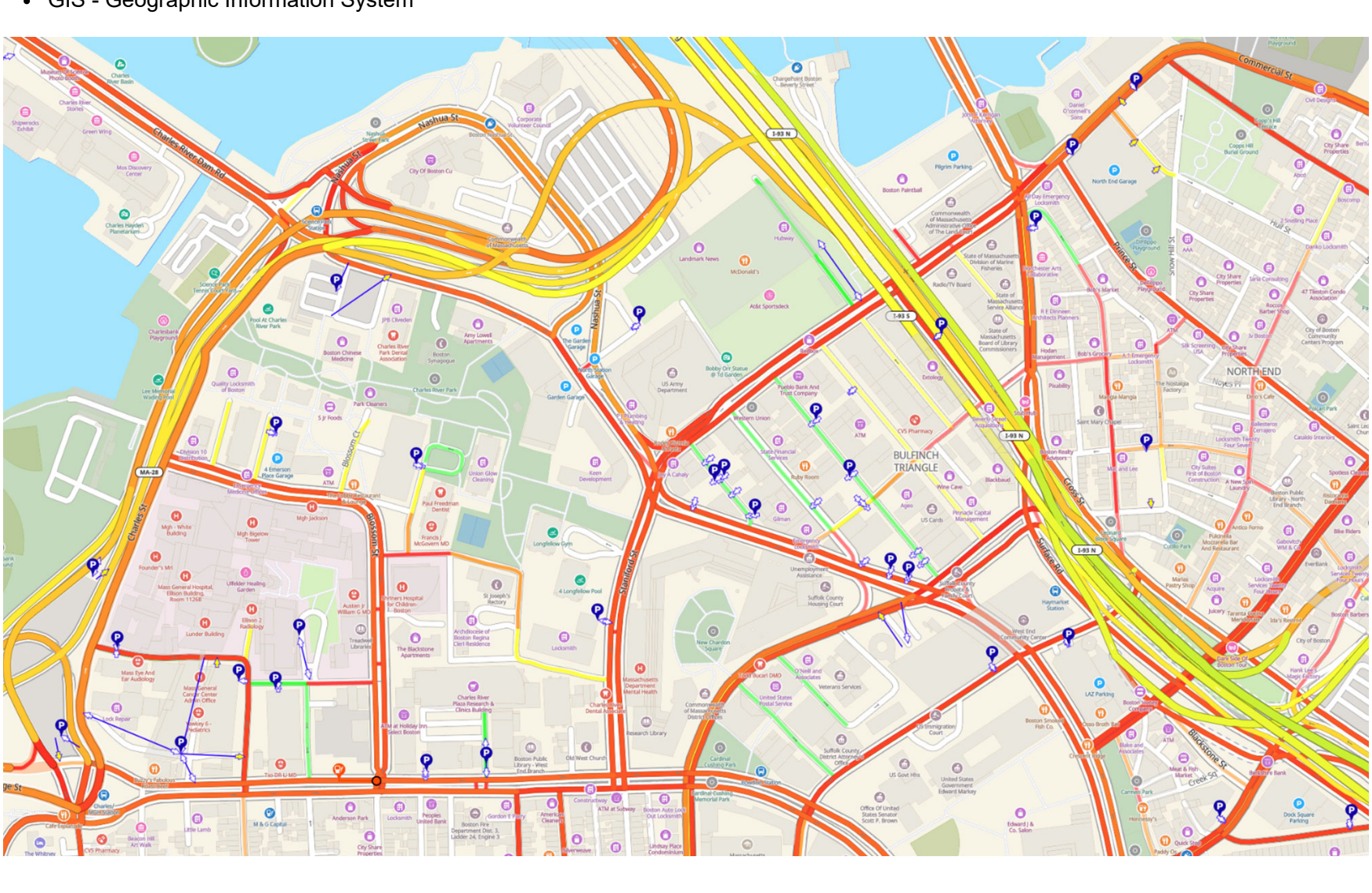
Today's agenda

- 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]: (227, 22)

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

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	
0	2021-02-15T13:29:38.240Z	19.211166	-155.441666	33.830002	1.93	md	43.0	129.0	NaN	0.10	...	2021-02-15T13:32:53.430Z	3 km ENE of Pāhala, Hawaii	earth
1	2021-02-15T13:24:55.830Z	33.471333	-116.411833	13.630000	0.45	ml	15.0	132.0	0.13150	0.18	...	2021-02-15T13:28:38.107Z	23km SSW of La Quinta, CA	earth
2	2021-02-15T13:17:58.660Z	35.918333	-117.677167	3.130000	1.19	ml	17.0	69.0	0.03665	0.18	...	2021-02-15T13:21:35.641Z	21km E of Little Lake, CA	earth
3	2021-02-15T12:53:58.610Z	18.204500	-67.123500	18.000000	2.96	md	17.0	123.0	0.02480	0.19	...	2021-02-15T13:33:36.412Z	1 km ENE of Mayagüez, Puerto Rico	earth
4	2021-02-15T12:31:34.610Z	33.258499	-117.155998	19.060000	1.48	ml	10.0	127.0	0.46510	0.28	...	2021-02-15T12:35:11.453Z	10km NE of Vista, CA	earth

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	2021-02-15T13:29:38.240Z	19.211166	-155.441666	1.93	3 km ENE of Pāhala, Hawaii
1	2021-02-15T13:24:55.830Z	33.471333	-116.411833	0.45	23km SSW of La Quinta, CA
2	2021-02-15T13:17:58.660Z	35.918333	-117.677167	1.19	21km E of Little Lake, CA
3	2021-02-15T12:53:58.610Z	18.204500	-67.123500	2.96	1 km ENE of Mayagüez, Puerto Rico
4	2021-02-15T12:31:34.610Z	33.258499	-117.155998	1.48	10km NE of Vista, CA

In [7]:
```
# shape
eqdf.shape
```
Out[7]: (227, 5)

In [8]:
```
# take "place" separately and display head
eqdf['place'].head()
```
Out[8]:
```
0           3 km ENE of Pāhala, Hawaii
1            23km SSW of La Quinta, CA
2            21km E of Little Lake, CA
3    1 km ENE of Mayagüez, Puerto Rico
4                 10km NE of Vista, 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]:
```
[['3 km ENE of Pāhala', 'Hawaii'],
 ['23km SSW of La Quinta', 'CA'],
 ['21km E of Little Lake', 'CA'],
 ['1 km ENE of Mayagüez', 'Puerto Rico'],
 ['10km NE of Vista', 'CA']]
```

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

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

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

Print first `5` from `area`

In [13]:
```
area[:5]
```
Out[13]: ['Hawaii', 'CA', 'CA', 'Puerto Rico', 'CA']

Print first `5` from `sub_area`

In [14]:
```
sub_area[:5]
```
Out[14]:
```
['3 km ENE of Pāhala',
 '23km SSW of La Quinta',
 '21km E of Little Lake',
 '1 km ENE of Mayagüez',
 '10km NE of Vista']
```

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	2021-02-15T13:29:38.240Z	19.211166	-155.441666	1.93	3 km ENE of Pāhala, Hawaii	3 km ENE of Pāhala	Hawaii
1	2021-02-15T13:24:55.830Z	33.471333	-116.411833	0.45	23km SSW of La Quinta, CA	23km SSW of La Quinta	CA
2	2021-02-15T13:17:58.660Z	35.918333	-117.677167	1.19	21km E of Little Lake, CA	21km E of Little Lake	CA
3	2021-02-15T12:53:58.610Z	18.204500	-67.123500	2.96	1 km ENE of Mayagüez, Puerto Rico	1 km ENE of Mayagüez	Puerto Rico
4	2021-02-15T12:31:34.610Z	33.258499	-117.155998	1.48	10km NE of Vista, CA	10km NE of Vista	CA

In [17]:
```
# shape
eqdf.shape
```
Out[17]: (227, 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	2021-02-15T13:29:38.240Z	19.211166	-155.441666	1.93	3 km ENE of Pāhala	Hawaii
1	2021-02-15T13:24:55.830Z	33.471333	-116.411833	0.45	23km SSW of La Quinta	CA
2	2021-02-15T13:17:58.660Z	35.918333	-117.677167	1.19	21km E of Little Lake	CA
3	2021-02-15T12:53:58.610Z	18.204500	-67.123500	2.96	1 km ENE of Mayagüez	Puerto Rico
4	2021-02-15T12:31:34.610Z	33.258499	-117.155998	1.48	10km NE of Vista	CA

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

In [20]:
```
prep_df.shape
```
Out[20]: (227, 6)

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

In [22]:
```
# shape
prep_df.shape
```
Out[22]: (80, 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	18
Hawaii	15
Puerto Rico	12
CA	7
southeast of the Loyalty Islands	5
Chile	3
Vanuatu	3
China	2
Japan	2
Nevada	2
Peru	1
U.S. Virgin Islands	1
Indonesia	1
Colombia	1
Texas	1
south of the Fiji Islands	1
off the coast of Ecuador	1
Timor Leste	1
Japan region	1
Owen Fracture Zone region	1
Fiji	1

default plot will not be so interactive

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

`Pie` chart with `Plotly`

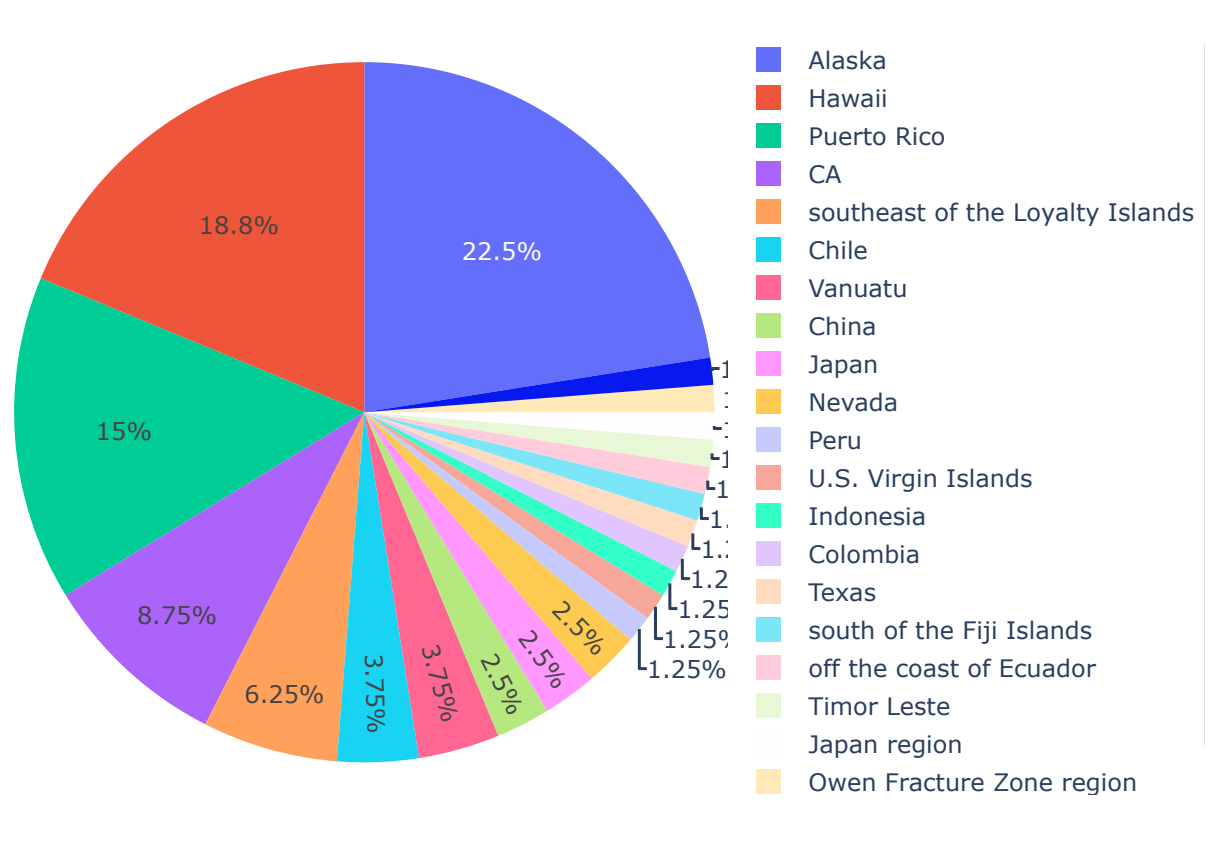
- labels → index
- values → area
- lrtb → 0

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 = 'Alaska'
# rdf from prep_df
rdf = prep_df[prep_df['area'] == region]
```

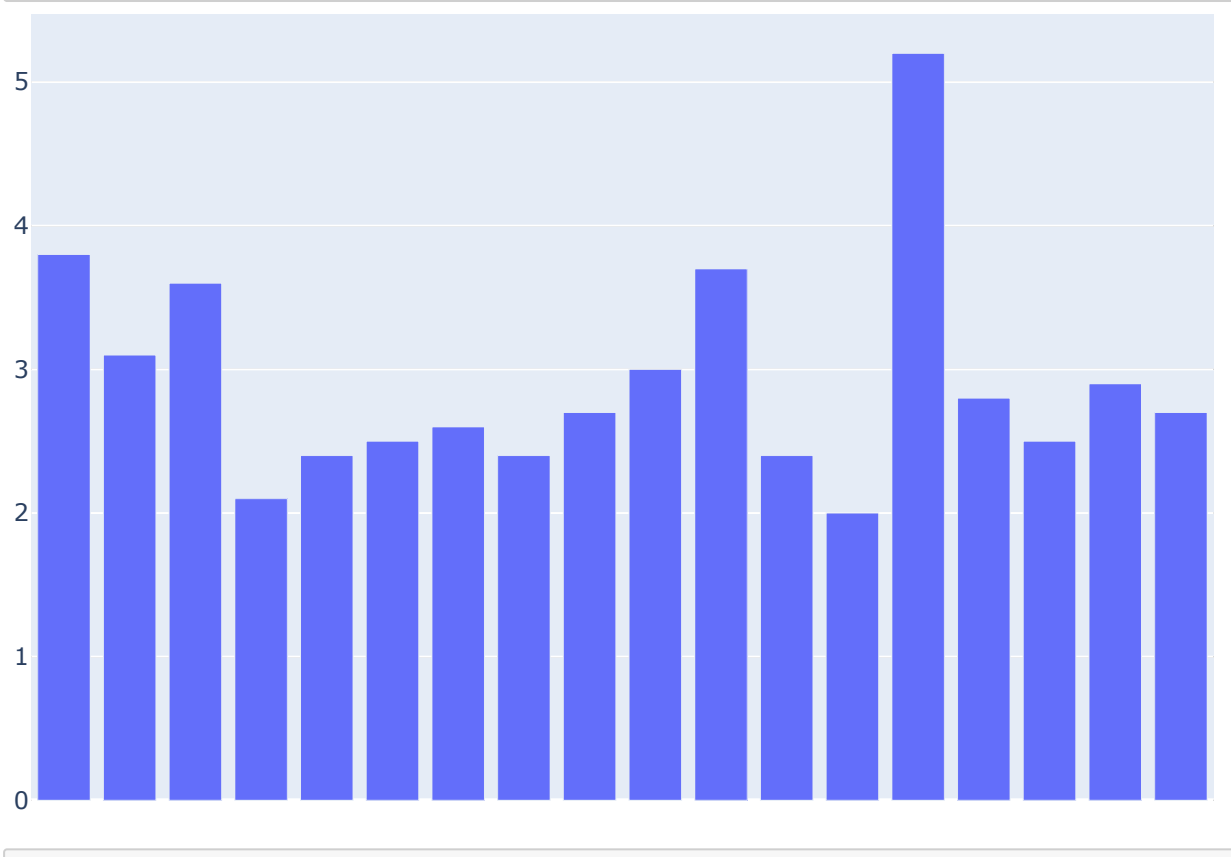
In [28]:
```
# head()
rdf.head()
```
Out[28]:

	time	latitude	longitude	mag	sub_area	area
28	2021-02-15T10:13:46.731Z	68.2043	-146.3224	3.8	33 km WNW of Arctic Village	Alaska
35	2021-02-15T09:39:15.653Z	64.0322	-148.9389	3.1	8 km E of Ferry	Alaska
52	2021-02-15T06:37:29.388Z	60.1758	-151.2184	3.6	11 km ESE of Clam Gulch	Alaska
57	2021-02-15T06:28:50.671Z	63.1601	-150.5821	2.1	71 km SE of Denali National Park	Alaska
95	2021-02-15T02:34:55.454Z	63.1296	-155.1604	2.4	29 km NE of McGrath	Alaska

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

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 [ ]:

Map Visualization

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

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

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

rdf = prep_df[prep_df['area'] == region]
rdf.head()
```
Out[30]:

	time	latitude	longitude	mag	sub_area	area
36	2021-02-15T09:30:49.080Z	36.082832	-119.903503	2.47	10km NNE of Kettleman City	CA
59	2021-02-15T06:14:16.920Z	36.157333	-118.178333	2.04	21km SW of Olancha	CA
90	2021-02-15T03:07:30.340Z	33.162167	-115.637833	2.52	12km WNW of Calipatria	CA
112	2021-02-15T00:50:43.700Z	34.164833	-117.468833	2.14	8km N of Fontana	CA
118	2021-02-15T00:03:18.210Z	38.229500	-122.191002	2.23	4km SW of Green Valley	CA

In [31]:
```
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'
    )
)

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
- Where it is used and how it is important for the business or the govt agency
- Python modules that support geospatial visualization
- Earthquake data - using pandas
  - Data preparation
  - Data filtering
- Earthquake data visualization