downloading_data

December 23, 2021

[ ]:
```
#getting row headers
#plotting headers and their positions/ index
#extracting and reading data/ high temp
#plotting data in temp chart
#plotting dates
```

[4]:
```
#getting row_header froms a csv file
import csv
file = 'data/sitka_weather_07-2018_simple.csv'
with open(filename) as f:
    reader = csv.reader(f)
    header_rows = next(reader)
    print(header_rows)
```

```
['STATION', 'NAME', 'DATE', 'PRCP', 'TAVG', 'TMAX', 'TMIN']
```

[2]:
```
#plotting headers and their positions/ index
import csv
file = 'data/sitka_weather_07-2018_simple.csv'
with open(file) as f:
    reader = csv.reader(f)
    header_rows = next(reader)
for index , headers in enumerate(header_rows):
    print(index, headers)
```

```
0 STATION
1 NAME
2 DATE
3 PRCP
4 TAVG
5 TMAX
6 TMIN
```

[17]:
```
#extracting and reading data/ high temp
import csv
file = 'data/sitka_weather_07-2018_simple.csv'
with open(file) as f:
    reader = csv.reader(f)
    header_row = next(reader)
```

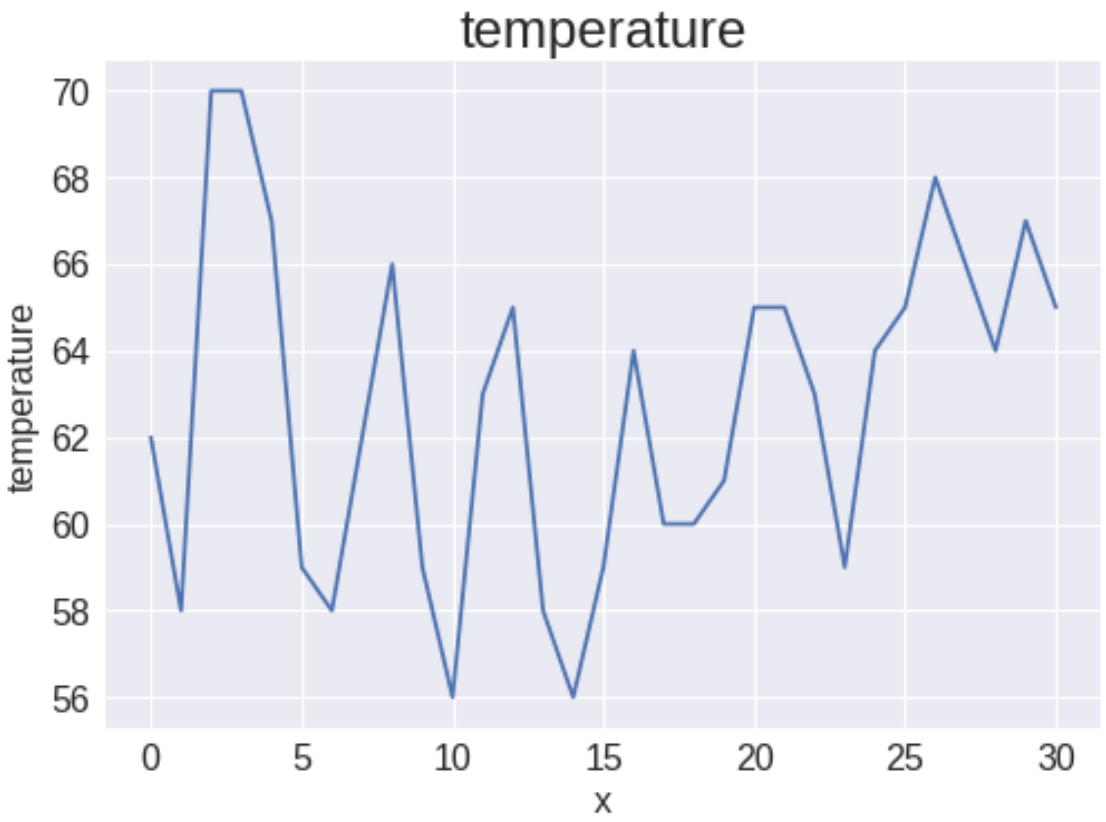
```
    highs = []
    for row in reader:
        high = int(row[5])
        highs.append(high)
print(highs)
```

```
[62, 58, 70, 70, 67, 59, 58, 62, 66, 59, 56, 63, 65, 58, 56, 59, 64, 60, 60, 61,
65, 65, 63, 59, 64, 65, 68, 66, 64, 67, 65]
```

[21]:
```
#plotting data in temp chart

import matplotlib.pyplot as plt
import csv

file = 'data/sitka_weather_07-2018_simple.csv'
with open(file) as f:
    reader = csv.reader(f)
    header_row = next(reader)

    highs = []
    for HT_row1 in reader:
        high = int(HT_row1[5])
        highs.append(high)
print(highs)

plt.style.use('seaborn')
fig, ax = plt.subplots()
plt.plot(highs)
ax.set_title('temperature', fontsize = 24)
ax.set_xlabel('x',fontsize = 16 )
ax.set_ylabel('temperature', fontsize = 16)
ax.tick_params(axis = 'both', which = 'major', labelsize = 16)
plt.show
```

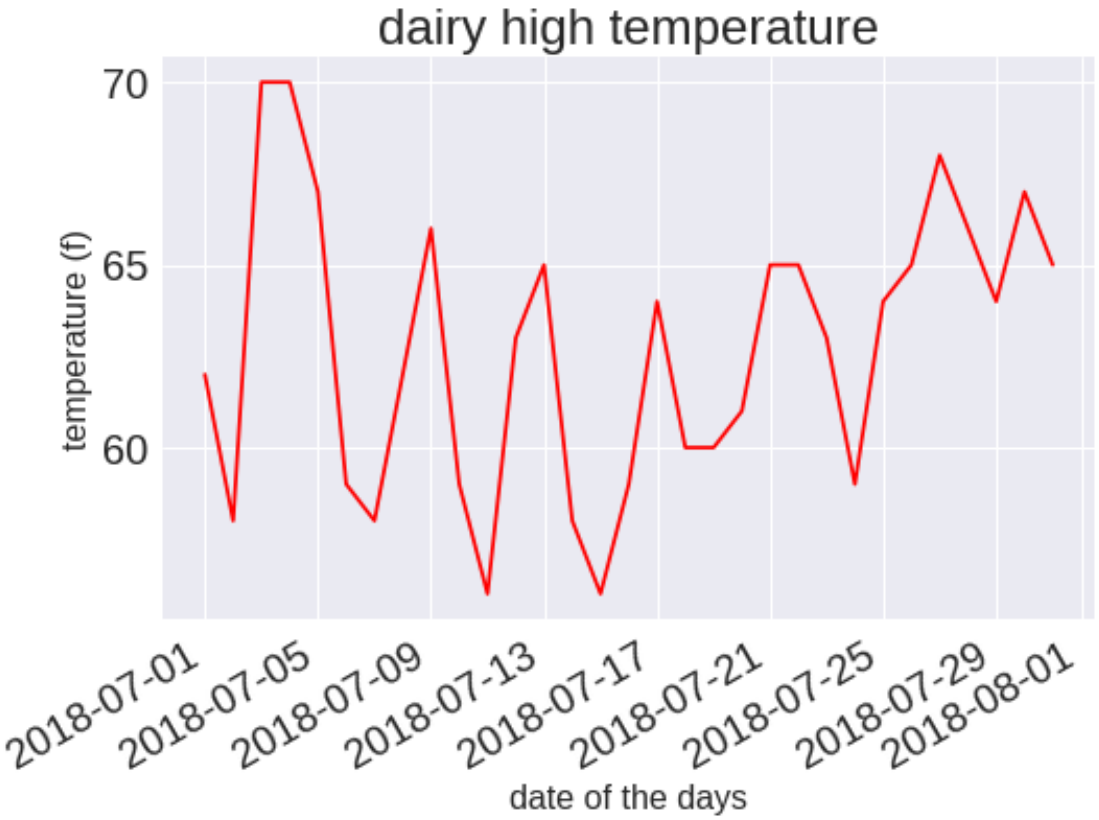
```
[62, 58, 70, 70, 67, 59, 58, 62, 66, 59, 56, 63, 65, 58, 56, 59, 64, 60, 60, 61,
65, 65, 63, 59, 64, 65, 68, 66, 64, 67, 65]
```

[21]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



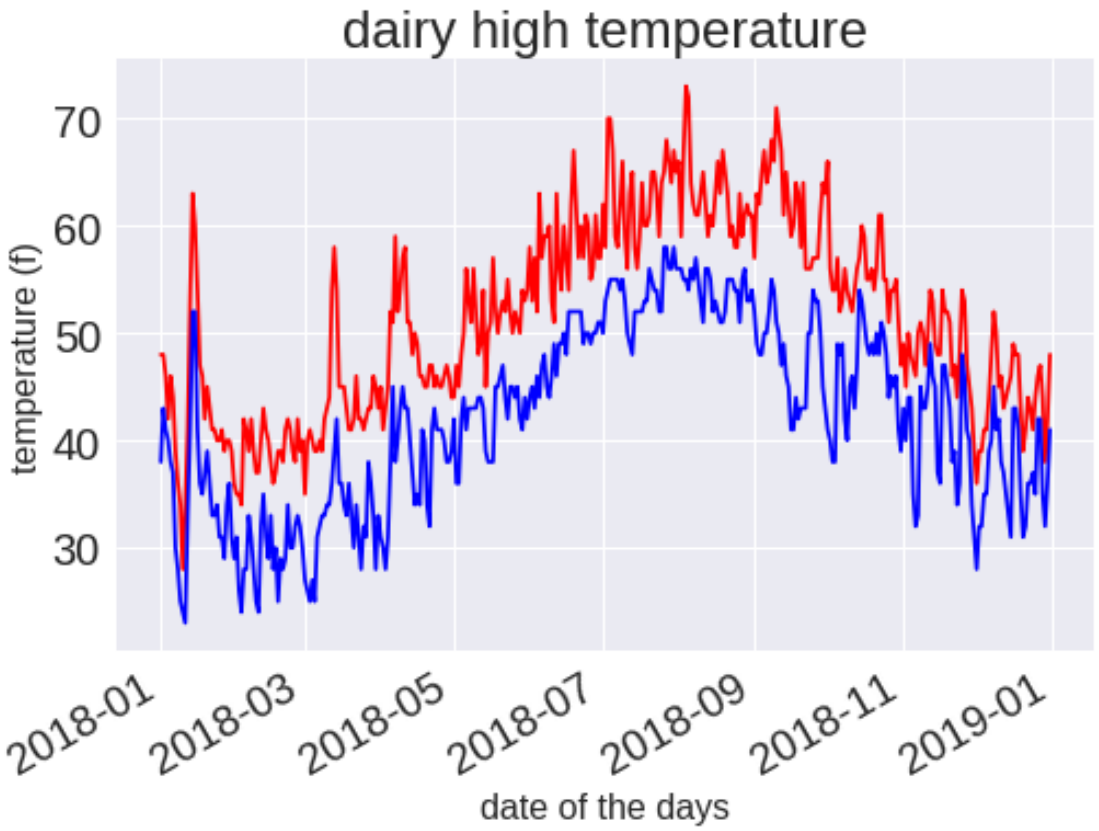
[28]:
```
#plotting dates
import matplotlib.pyplot as plt
import csv
from datetime import datetime
file = 'data/sitka_weather_07-2018_simple.csv'
with open(file) as f:
    reader = csv.reader(f)
    row_header = next(reader)

    highs = []
    dates = []
    for row in reader:
        high = int(row[5])
        highs.append(high)
        date = datetime.strptime(row[2], '%Y-%m-%d')
        dates.append(date)

plt.style.use('seaborn')
fig, ax = plt.subplots()
ax.plot(dates, highs, c = 'red')
```

```
ax.set_title('dairy high temperature', fontsize = 24)
ax.set_xlabel('date of the days', fontsize = 16)
fig.autofmt_xdate()
ax.set_ylabel('temperature (f)', fontsize = 16)
ax.tick_params(axis = 'both', which = 'major', labelsize = 20)
plt.show()
```

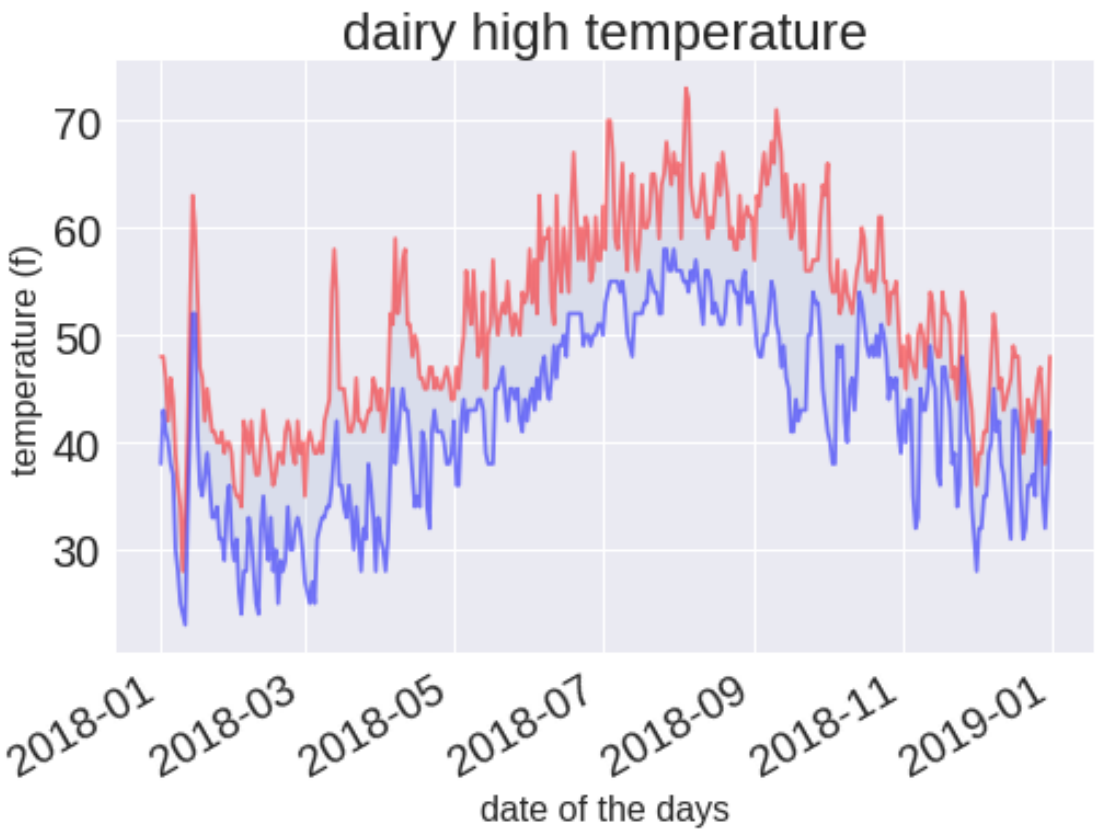


[30]:
```
#plotting a larger ttimestamp / second data series
import matplotlib.pyplot as plt
import csv
from datetime import datetime
file = 'data/sitka_weather_2018_simple.csv' #c
with open(file) as f:
    reader = csv.reader(f)
    row_header = next(reader)

    highs = []
    dates = []
    lows = [] #c
```

```
    for row in reader:
        high = int(row[5])
        highs.append(high)
        date = datetime.strptime(row[2], '%Y-%m-%d')
        dates.append(date)
        low = int(row[6])#c
        lows.append(low)#c

plt.style.use('seaborn')
fig, ax = plt.subplots()
ax.plot(dates, highs, c = 'red')
ax.plot(dates, lows, c = 'blue')#c

ax.set_title('dairy high temperature', fontsize = 24)
ax.set_xlabel('date of the days', fontsize = 16)
fig.autofmt_xdate()
ax.set_ylabel('temperature (f)', fontsize = 16)
ax.tick_params(axis = 'both', which = 'major', labelsize = 20)
plt.show()
```



[1]:
```
#filling in between two series
import matplotlib.pyplot as plt
import csv
from datetime import datetime
file = 'data/sitka_weather_2018_simple.csv'
with open(file) as f:
    reader = csv.reader(f)
    row_header = next(reader)

    highs, dates, lows= [], [], [] #C
    for row in reader:
        high = int(row[5])
        highs.append(high)
        date = datetime.strptime(row[2], '%Y-%m-%d')
        dates.append(date)
        low = int(row[6])
        lows.append(low)

plt.style.use('seaborn')
fig, ax = plt.subplots()
ax.plot(dates, highs, c = 'red', alpha = 0.5)#c
ax.plot(dates, lows, c = 'blue', alpha = 0.5)#c
ax.fill_between( dates, highs, lows, alpha = 0.1)#c
ax.set_title('dairy high temperature', fontsize = 24)
ax.set_xlabel('date of the days', fontsize = 16)
fig.autofmt_xdate()
ax.set_ylabel('temperature (f)', fontsize = 16)
ax.tick_params(axis = 'both', which = 'major', labelsize = 20)
plt.show()
```



[5]:
```
#examining json data
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)
```

[6]:
```
#creating a list out of json data
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)
```

```
eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))
```

```
158
```

[10]:
```
#extracting magnitude
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))

mags = []
for all_mags in all_mag_list:
    mag = all_mags['properties']['mag']
    mags.append(mag)
print(mags)
```

```
158
[0.96, 1.2, 4.3, 3.6, 2.1, 4, 1.06, 2.3, 4.9, 1.8, 3.29, 1.3, 4.5, 3.08, 2.2,
1.3, 1.6, 1.29, 2.1, 2, 1.8, 3.08, 1.4, 2.33, 1.4, 3.11, 2.9, 2.3, 1.5, 1.89,
1.05, 1.16, 1.4, 1.3, 1.68, 1.29, 1.4, 1.3, 1.87, 1, 1.8, 1.7, 1.34, 1.9, 1.7,
2.6, 1.19, 2.46, 1.4, 1.6, 2.4, 1.64, 2.9, 1.09, 1.58, 1.3, 1.57, 1.8, 2.24,
4.7, 1.09, 1.5, 1.85, 4.2, 2.4, 2.2, 2.23, 1.16, 1.4, 1.7, 1.8, 1.12, 4.7, 3.9,
1.84, 2.71, 1.8, 2.19, 1.13, 2.1, 5.2, 2.3, 3.1, 1.74, 2.5, 1.5, 4.6, 2.74,
1.61, 2.01, 4.8, 2.96, 2, 1.87, 2.2, 3.76, 2.21, 1.51, 1.51, 2.2, 2.87, 3.5,
1.1, 1.05, 2.77, 2.7, 2.6, 2.4, 1.03, 5.3, 1.6, 5.2, 4.3, 0.99, 1.4, 1.16, 1.7,
4.7, 2.5, 1.77, 1.6, 2.08, 4.2, 1.6, 1.8, 1.99, 2.38, 0.98, 3.23, 4.3, 3.4, 1.6,
2.2, 2.3, 2.4, 3, 2.1, 2.2, 4.8, 2.85, 1.6, 1.91, 2.6, 1.7, 4.2, 3.28, 2.2,
1.16, 2.5, 4.7, 3.32, 2.5, 2.75, 3.2, 1.6, 1.5, 1.3, 2.96]
```

[12]:
```
#extracting location data
import json

file = 'eqdata/eq_data_1_day_m1.json'
```

```
with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))

mags, lons, lats= [], [], []
for values in all_mag_list:
    mag = values['properties']['mag']
    lon = values['geometry']['coordinates'][0]
    lat = values['geometry']['coordinates'][1]
    mags.append(mag)
    lons.append(lon)
    lats.append(lat)
print(mags)
print(lons)
print(lats)
```

```
158
[0.96, 1.2, 4.3, 3.6, 2.1, 4, 1.06, 2.3, 4.9, 1.8, 3.29, 1.3, 4.5, 3.08, 2.2,
1.3, 1.6, 1.29, 2.1, 2, 1.8, 3.08, 1.4, 2.33, 1.4, 3.11, 2.9, 2.3, 1.5, 1.89,
1.05, 1.16, 1.4, 1.3, 1.68, 1.29, 1.4, 1.3, 1.87, 1, 1.8, 1.7, 1.34, 1.9, 1.7,
2.6, 1.19, 2.46, 1.4, 1.6, 2.4, 1.64, 2.9, 1.09, 1.58, 1.3, 1.57, 1.8, 2.24,
4.7, 1.09, 1.5, 1.85, 4.2, 2.4, 2.2, 2.23, 1.16, 1.4, 1.7, 1.8, 1.12, 4.7, 3.9,
1.84, 2.71, 1.8, 2.19, 1.13, 2.1, 5.2, 2.3, 3.1, 1.74, 2.5, 1.5, 4.6, 2.74,
1.61, 2.01, 4.8, 2.96, 2, 1.87, 2.2, 3.76, 2.21, 1.51, 1.51, 2.2, 2.87, 3.5,
1.1, 1.05, 2.77, 2.7, 2.6, 2.4, 1.03, 5.3, 1.6, 5.2, 4.3, 0.99, 1.4, 1.16, 1.7,
4.7, 2.5, 1.77, 1.6, 2.08, 4.2, 1.6, 1.8, 1.99, 2.38, 0.98, 3.23, 4.3, 3.4, 1.6,
2.2, 2.3, 2.4, 3, 2.1, 2.2, 4.8, 2.85, 1.6, 1.91, 2.6, 1.7, 4.2, 3.28, 2.2,
1.16, 2.5, 4.7, 3.32, 2.5, 2.75, 3.2, 1.6, 1.5, 1.3, 2.96]
[-116.7941667, -148.9865, -74.2343, -161.6801, -118.5316667, -144.1283,
-116.7433333, -153.7845, 59.3991, -116.2045, -112.4931667, -148.7061, 161.1651,
-111.0103333, -154.4451, -150.0831, -149.9686, -116.9223333, -143.4818,
-150.0881, -155.2371674, -121.2385025, -148.6093, -66.8275, -152.6213, -68.0565,
-151.6356, -150.5944, -150.0873, -118.5568333, -116.5841667, -116.3608333,
-149.5674, -149.2188, -115.54, -122.8098297, -146.8137, -151.7243, -67.0001,
-122.7545013, -149.517, -152.7856, -118.2513333, -153.0445, -151.5818,
-103.5309, -117.671, -66.5226, -150.0767, -151.1402, -151.2276, -121.675499,
-104.39, -116.7905, -122.7925034, -149.9197, -122.758667, -151.4699, -66.9725,
155.5658, -116.7101667, -149.9036, -121.2993317, 164.5151, -154.4521,
-121.3024979, -155.418335, -117.4768333, -148.1971, -150.096, -150.0625,
-117.8083333, 140.1743, -176.7088, -155.119339, -121.1493301, -150.6634,
-66.9566, -120.5666656, -150.585, 125.0184, -145.8011, -149.8121, -66.9698,
```

```
-149.9367, -149.6652, -102.8196, -123.1094971, -121.5001678, -66.9858, 124.9248,
-66.5431, -155.3906708, -66.9795, -153.0101, -66.974, -122.7218323,
-118.0016667, -122.6969986, -157.222, -67.2206, -155.268, -150.0115,
-122.8075027, -66.5303, -149.7727, -148.5229, -151.9045, -122.8550034, -29.2389,
-111.916, 140.123, -67.31, -117.116, -150.1082, -122.8195038, -150.1297,
-29.2708, -144.8571, -120.3855, -147.2053, -66.0431, 102.7618, -148.5062,
-152.8891, -122.8151703, -67.9588, -122.8093338, -64.8163, -4.5477, -149.5711,
-149.9127, -103.516, -153.3, -150.0994, -65.1178, -148.7989, -149.8974,
151.4974, -66.0208, -152.22, -155.272995, -157.2918, -149.9289, 20.7165,
-67.4526, -152.3045, -122.7653351, -149.4308, 101.9982, -65.7698, -143.7027,
-67.1808, -64.6176, -149.982, -141.1985, -147.2898, -67.2263]
[33.4863333, 64.6673, -12.1025, 54.2232, 35.3098333, 69.5346, 33.5148333,
59.6106, -30.7399, 37.0572, 38.7381667, 64.8513, 53.566, 44.4668333, 58.9382,
61.3962, 61.4763, 34.1115, 62.9725, 61.2929, 19.3859997, 39.9780006, 63.5896,
18.0063, 60.2343, 19.0778, 60.5105, 61.8365, 61.4155, 35.3471667, 33.807,
33.4156667, 61.7445, 61.9231, 32.9328333, 38.8139992, 61.2207, 60.2853, 17.9666,
38.777832, 62.8545, 59.7132, 35.6781667, 61.3963, 63.066, 31.3115, 36.1148333,
19.1563, 61.4484, 63.3234, 63.388, 37.3089981, 31.6283, 33.4871667, 38.8308334,
61.4028, 38.7929993, 62.2867, 18.2445, -6.9349, 33.5293333, 61.373, 36.6803322,
54.7415, 58.9606, 36.6758347, 19.9256668, 34.1168333, 64.6191, 61.4036, 62.1657,
36.0328333, -2.4946, 51.2506, 19.3426666, 37.0415001, 62.0432, 18.24,
36.0043335, 62.9592, 25.293, 62.076, 61.2137, 18.2406, 61.401, 61.4829, 17.4382,
39.2228317, 37.041832, 18.272, 25.3087, 19.2095, 19.1879997, 18.2796, 59.7778,
18.2478, 38.7808342, 36.0621667, 38.7721672, 66.3024, 19.174, 58.532, 61.4857,
38.8311653, 19.2395, 63.1081, 59.7954, 61.4739, 38.8206673, 41.3292, 40.3815,
-2.4375, -24.105, 34.018, 61.3612, 38.8149986, 61.4301, 41.3247, 69.5205,
46.132333333333, 61.4891, 17.9191, 47.7498, 61.8179, 60.1279, 38.8143349,
18.0605, 38.8224983, 19.3835, 34.9073, 56.1963, 61.5014, 31.3972, 60.0044,
61.3753, 18.6975, 63.8818, 61.4737, 45.3002, 19.1953, 61.1183, 19.413166, 66.29,
61.493, 37.5408, 18.7816, 61.0343, 38.7921677, 56.1936, -4.4716, 19.1911,
69.4029, 18.2208, 19.1621, 61.4021, 60.4072, 64.9632, 19.0268]
```

[17]:
```python
#building a world map
from plotly.graph_objs import Scattergeo, Layout
from plotly import offline
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
```

```
print(len(all_mag_list))

mags, lons, lats= [], [], []
for values in all_mag_list:
    mag = values['properties']['mag']
    lon = values['geometry']['coordinates'][0]
    lat = values['geometry']['coordinates'][1]
    mags.append(mag)
    lons.append(lon)
    lats.append(lat)

data = [Scattergeo(lon = lons, lat = lats)]
my_layout = Layout(title = 'my firt world map rep eq mags')

fig = {'data': data, 'layout':my_layout}
offline.plot(fig, filename= ' eqworlmap.html')
```

```
158
```

[17]:
```
' eqworlmap.html'
```

[4]:
```
#different way of specifying the chat data
from plotly.graph_objs import Scattergeo, Layout
from plotly import offline
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))

mags, lons, lats= [], [], []
for values in all_mag_list:
    mag = values['properties']['mag']
    lon = values['geometry']['coordinates'][0]
    lat = values['geometry']['coordinates'][1]
    mags.append(mag)
    lons.append(lon)
    lats.append(lat)
#change
data = [
```

```
    {
    'type':'scattergeo',
    'lon':lons,
    'lat':lats,
    }
]
#end of changes
my_layout = Layout(title = 'my firt world map rep eq mags')

fig = {'data': data, 'layout':my_layout}
offline.plot(fig, filename= ' eqworlmap.html')
```

158

[4]: ' eqworlmap.html'

[13]:
```
#customaizing the marker size
from plotly.graph_objs import Scattergeo, Layout
from plotly import offline
import json

file = 'eqdata/eq_data_1_day_m1.json'

with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))

mags, lons, lats = [], [], []
for values in all_mag_list:
    mag = values['properties']['mag']
    lon = values['geometry']['coordinates'][0]
    lat = values['geometry']['coordinates'][1]
    mags.append(mag)
    lons.append(lon)
    lats.append(lat)
#change
data = [
    {
    'type':'scattergeo',
    'lon':lons,
    'lat':lats,
    'marker':{
```

```
        'size':[5*mag for mag in mags],
        'color':mags,
        'colorscale':'viridis',
        'reversescale':True,
        'colorbar':{'title':'magnitude'},
    },
    }
]
#end of changes
my_layout = Layout(title = 'my firt world map rep eq mags')

fig = {'data': data, 'layout':my_layout}
offline.plot(fig, filename= ' eqworlmap.html')
```

```
158
```

[13]:
```
' eqworlmap.html'
```

[17]:
```
#other color scale
from plotly import colors
for key in colors.PLOTLY_SCALES.keys():
    print(key)
```

```
Greys
YlGnBu
Greens
YlOrRd
Bluered
RdBu
Reds
Blues
Picnic
Rainbow
Portland
Jet
Hot
Blackbody
Earth
Electric
Viridis
Cividis
```

[26]:
```
#adding hoover text
from plotly.graph_objs import Scattergeo, Layout
from plotly import offline
import json

file = 'eqdata/eq_data_30_day_m1.json'
```

```
with open(file) as f:
    loader = json.load(f)

eqd_readable = 'eqdata/eq_data_1_day_readerble.json'
with open(eqd_readable, 'w') as f:
    json.dump(loader, f, indent = 4)

all_mag_list = loader['features']
print(len(all_mag_list))

mags, lons, lats, hoover_text= [], [], [], []
for values in all_mag_list:
    mag = values['properties']['mag']
    lon = values['geometry']['coordinates'][0]
    lat = values['geometry']['coordinates'][1]
    title = values['properties']['title']
    mags.append(mag)
    lons.append(lon)
    lats.append(lat)
    hoover_text.append(title)
#change
data = [
    {
    'type':'scattergeo',
    'lon':lons,
    'lat':lats,
    'text':hoover_text,
    'marker':{
        'size':[5*mag for mag in mags],
        'color':mags,
        'colorscale':'viridis',
        'reversescale':True,
        'colorbar':{'title':'magnitude'},
    },
    }
]
#end of changes
my_layout = Layout(title = 'my firt world map rep earhquake magnitude')

fig = {'data': data, 'layout':my_layout}
offline.plot(fig, filename= ' eqworlmap.html')
```

```
6274
```

[26]:
```
' eqworlmap.html'
```

[ ]:
```

```