Lab 7 demo

In [1]:

```
# Install package for obtaining USGS streamflow data
!pip install -U dataretrieval
```

```
Requirement already satisfied: pandas in /Library/Frameworks/Python.f
ramework/Versions/3.9/lib/python3.9/site-packages (from dataretrieval
) (1.4.1)
Requirement already satisfied: requests in /Library/Frameworks/Python
.framework/Versions/3.9/lib/python3.9/site-packages (from dataretriev
al) (2.25.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Library/Fra
meworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (fr
om pandas->dataretrieval) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Library/Frameworks/Py
thon.framework/Versions/3.9/lib/python3.9/site-packages (from pandas-
>dataretrieval) (2021.3)
Requirement already satisfied: numpy>=1.18.5 in /Library/Frameworks/P
ython.framework/Versions/3.9/lib/python3.9/site-packages (from pandas
->dataretrieval) (1.22.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Library/Fram
eworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (fro
m requests->dataretrieval) (1.26.2)
Requirement already satisfied: chardet<4,>=3.0.2 in /Library/Framewor
ks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from re
quests->dataretrieval) (3.0.4)
```

In [2]:

```
# Import the functions for downloading data from NWIS
import dataretrieval.nwis as nwis

# Specify the USGS site code
site = '03339000'

# Get instantaneous values (iv)
df = nwis.get_record(sites=site, service='dv', start='2020-10-01', enc
df
```

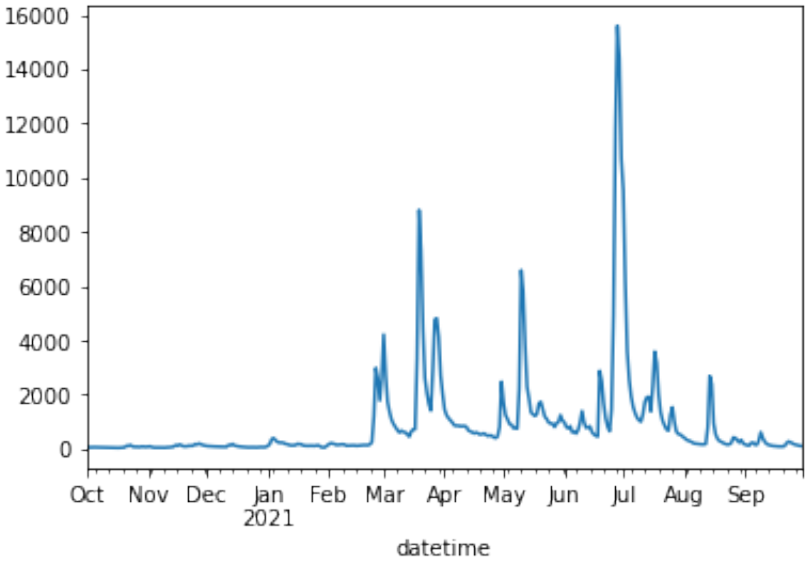
Out[2]:

	00010_Mean	00010_Mean_cd	site_no	00060_Mean	00060_Mean_cd	00065_M
datetime						
2020-10-01 00:00:00+00:00	14.9	A	03339000	75.7	A	
2020-10-02 00:00:00+00:00	14.5	A	03339000	66.0	A	
2020-10-03 00:00:00+00:00	14.2	A	03339000	60.2	A	
2020-10-04 00:00:00+00:00	14.4	A	03339000	68.8	A	
2020-10-05 00:00:00+00:00	13.4	A	03339000	66.8	A	
...	...	...	...	...	...	
2021-09-26 00:00:00+00:00	18.7	A	03339000	174.0	A	
2021-09-27 00:00:00+00:00	20.2	A	03339000	155.0	A	
2021-09-28 00:00:00+00:00	21.6	A	03339000	132.0	A	
2021-09-29 00:00:00+00:00	22.3	A	03339000	117.0	A	
2021-09-30 00:00:00+00:00	22.6	A	03339000	111.0	A	

365 rows × 41 columns

```
In [3]: # Simple plot
        df['00060_Mean'].plot()
```

```
Out[3]: <AxesSubplot:xlabel='datetime'>
```



Question 1

In [4]:

```
# Specify the USGS site code (USGS 14211720 WILLAMETTE RIVER AT PORTLA
site = '14211720'

# Get instantaneous values (iv)
df_portland = nwis.get_record(sites=site, service='dv', start='2019-01
df_portland
```

Out[4]:

	00010_Maximum	00010_Maximum_cd	site_no	00010_Minimum	00010_Minimu
datetime					
2019-01-02 00:00:00+00:00	7.1	A	14211720	6.4	
2019-01-03 00:00:00+00:00	6.4	A	14211720	6.0	
2019-01-04 00:00:00+00:00	6.0	A	14211720	5.9	
2019-01-05 00:00:00+00:00	6.3	A	14211720	5.9	
2019-01-06 00:00:00+00:00	6.6	A	14211720	6.3	
...	...	...	...	...	
2021-07-24 00:00:00+00:00	24.1	A	14211720	23.5	
2021-07-25 00:00:00+00:00	24.1	A	14211720	23.4	
2021-07-26 00:00:00+00:00	24.1	A	14211720	23.5	
2021-07-27 00:00:00+00:00	24.4	A	14211720	23.6	
2021-07-28 00:00:00+00:00	24.5	A	14211720	23.8	

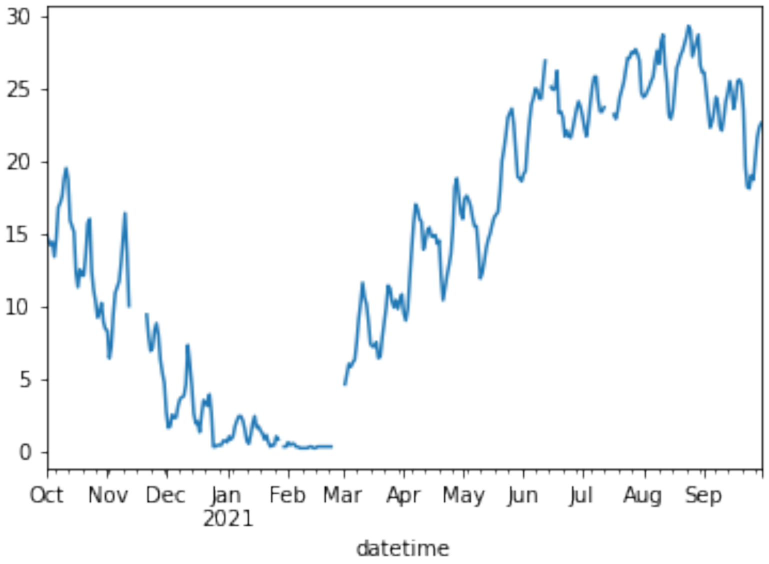
939 rows × 47 columns

In [5]:

```
# Simple plot (Temperature, water, degrees Celsius)
# https://waterdata.usgs.gov/nwis/inventory?agency_code=USGS&site_no=1
df['00010_Mean'].plot()
```

Out[5]:

```
<AxesSubplot:xlabel='datetime'>
```

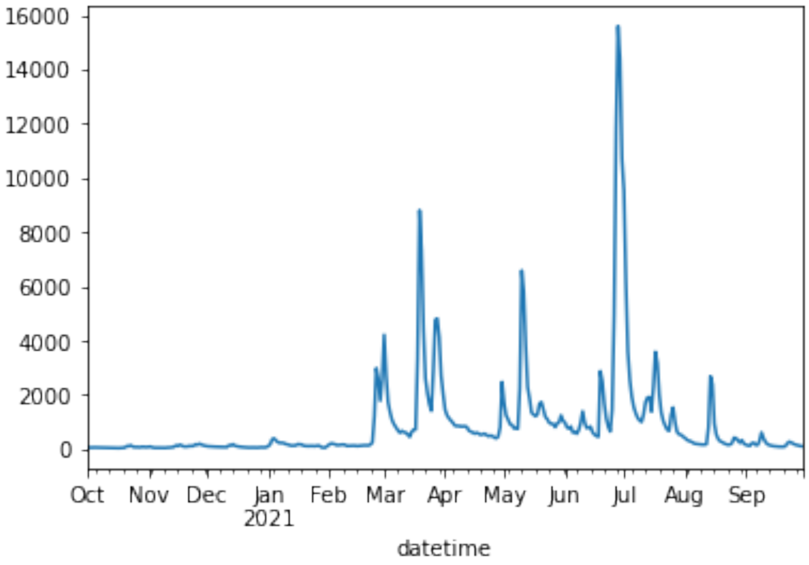


In [6]:

```
# Simple plot ()
# https://waterdata.usgs.gov/nwis/inventory?agency_code=USGS&site_no=1
df['00060_Mean'].plot()
```

Out[6]:

```
<AxesSubplot:xlabel='datetime'>
```



Question 2: HTML tables

In [7]:

```
# Specify the USGS site code (USGS 14211720 WILLAMETTE RIVER AT PORTLA
site = '14211720'

# Get instantaneous values (iv)
# Date Oct 31, 2020 and Sep 30, 2021
df_portland = nwis.get_record(sites=site, service='dv', start='2021-09
df_portland
```

Out[7]:

	00010_Maximum	00010_Maximum_cd	site_no	00010_Minimum	00010_Minimu
datetime					
2021-09-30 00:00:00+00:00	17.2	P	14211720	17.1	
2021-10-01 00:00:00+00:00	17.2	P	14211720	16.8	
2021-10-02 00:00:00+00:00	16.8	P	14211720	16.5	
2021-10-03 00:00:00+00:00	16.5	P	14211720	16.0	
2021-10-04 00:00:00+00:00	16.0	P	14211720	15.7	
2021-10-05 00:00:00+00:00	15.9	P	14211720	15.5	
2021-10-06 00:00:00+00:00	15.5	P	14211720	15.3	
2021-10-07 00:00:00+00:00	15.4	P	14211720	15.1	
2021-10-08 00:00:00+00:00	15.1	P	14211720	15.0	
2021-10-09 00:00:00+00:00	15.0	P	14211720	14.8	
2021-10-10 00:00:00+00:00	14.8	P	14211720	14.4	
2021-10-11 00:00:00+00:00	14.4	P	14211720	14.0	
2021-10-12 00:00:00+00:00	14.0	P	14211720	13.6	
2021-10-13 00:00:00+00:00	13.6	P	14211720	13.3	
2021-10-14 00:00:00+00:00	13.3	P	14211720	13.2	
2021-10-15	13.5	P	14211720	13.1	

00:00:00+00:00				
2021-10-16 00:00:00+00:00	13.4	P	14211720	13.1
2021-10-17 00:00:00+00:00	13.3	P	14211720	13.0
2021-10-18 00:00:00+00:00	13.1	P	14211720	12.8
2021-10-19 00:00:00+00:00	12.9	P, [4]	14211720	12.8
2021-10-20 00:00:00+00:00	13.1	P	14211720	12.9
2021-10-21 00:00:00+00:00	13.2	P	14211720	13.0
2021-10-22 00:00:00+00:00	13.4	P	14211720	13.2
2021-10-23 00:00:00+00:00	13.3	P	14211720	13.2
2021-10-24 00:00:00+00:00	13.2	P	14211720	12.8
2021-10-25 00:00:00+00:00	12.8	P	14211720	12.4
2021-10-26 00:00:00+00:00	12.4	P	14211720	12.3
2021-10-27 00:00:00+00:00	12.4	P	14211720	12.1
2021-10-28 00:00:00+00:00	12.1	P	14211720	11.9
2021-10-29 00:00:00+00:00	12.1	P	14211720	11.9
2021-10-30 00:00:00+00:00	12.0	P	14211720	11.7
2021-10-31 00:00:00+00:00	12.0	P	14211720	11.7

32 rows × 47 columns

In [8]:
```
# Calculate Mean for Portland
df_portland['00060_Mean'].mean()
```

Out[8]:
```
13064.193548387097
```

In [9]:
```
# Specify the USGS site code (USGS 14211720 WILLAMETTE RIVER AT PORTLA
site = '14163900'

# Get instantaneous values (iv)
# Date Oct 31, 2020 and Sep 30, 2021
df_Walterville = nwis.get_record(sites=site, service='dv', start='2021
df_Walterville
```

Out[9]:

	00010_Maximum	00010_Maximum_cd	site_no	00010_exo wq monitor_Maximum	000 monitor_Ma
datetime					
2021-09-30 00:00:00+00:00	11.8	P	14163900	11.8	
2021-10-01 00:00:00+00:00	13.0	P	14163900	13.1	
2021-10-02 00:00:00+00:00	12.9	P	14163900	12.9	
2021-10-03 00:00:00+00:00	12.6	P	14163900	12.7	
2021-10-04 00:00:00+00:00	12.0	P	14163900	12.0	
2021-10-05 00:00:00+00:00	11.2	P	14163900	11.2	
2021-10-06 00:00:00+00:00	11.8	P	14163900	11.9	
2021-10-07 00:00:00+00:00	11.1	P	14163900	11.1	
2021-10-08 00:00:00+00:00	10.7	P	14163900	10.7	
2021-10-09 00:00:00+00:00	11.2	P	14163900	11.2	
2021-10-10 00:00:00+00:00	11.3	P	14163900	11.3	
2021-10-11 00:00:00+00:00	10.9	P	14163900	10.9	
2021-10-12 00:00:00+00:00	9.7	P	14163900	9.8	
2021-10-13 00:00:00+00:00	10.2	P	14163900	10.3	
2021-10-14 00:00:00+00:00	10.8	P	14163900	10.8	
2021-10-15	11.0	P	14163900	11.0	

00:00:00+00:00				
2021-10-16 00:00:00+00:00	11.0	P	14163900	11.1
2021-10-17 00:00:00+00:00	10.4	P	14163900	10.4
2021-10-18 00:00:00+00:00	10.5	P	14163900	10.5
2021-10-19 00:00:00+00:00	10.6	P	14163900	10.7
2021-10-20 00:00:00+00:00	10.8	P	14163900	10.8
2021-10-21 00:00:00+00:00	10.7	P	14163900	10.6
2021-10-22 00:00:00+00:00	10.7	P	14163900	10.7
2021-10-23 00:00:00+00:00	11.1	P	14163900	11.2
2021-10-24 00:00:00+00:00	11.3	P	14163900	11.3
2021-10-25 00:00:00+00:00	10.7	P	14163900	10.7
2021-10-26 00:00:00+00:00	10.6	P	14163900	10.6
2021-10-27 00:00:00+00:00	10.8	P	14163900	10.8
2021-10-28 00:00:00+00:00	11.3	P	14163900	11.3
2021-10-29 00:00:00+00:00	11.4	P	14163900	11.4
2021-10-30 00:00:00+00:00	10.9	P	14163900	10.9
2021-10-31 00:00:00+00:00	10.2	P	14163900	10.2

In [10]:
```
# Calculate Mean for Walterville
df_Walterville['00060_Mean'].mean()
```

Out[10]:
```
1566.25
```

In [11]:
```
# Specify the USGS site code (USGS 14211720 WILLAMETTE RIVER AT PORTLA
site = '12422500'

# Get instantaneous values (iv)
# Date Oct 31, 2020 and Sep 30, 2021
df_Spokane = nwis.get_record(sites=site, service='dv', start='2021-09-
df_Spokane
```

Out[11]:

	00060_Mean	00060_Mean_cd	site_no	00065_Mean	00065_Mean_cd
datetime					
2021-09-30 00:00:00+00:00	1170.0	P	12422500	17.67	P
2021-10-01 00:00:00+00:00	1190.0	P	12422500	17.69	P
2021-10-02 00:00:00+00:00	1190.0	P	12422500	17.68	P
2021-10-03 00:00:00+00:00	1180.0	P	12422500	17.68	P
2021-10-04 00:00:00+00:00	1260.0	P	12422500	17.75	P
2021-10-05 00:00:00+00:00	1450.0	P	12422500	17.92	P
2021-10-06 00:00:00+00:00	1460.0	P	12422500	17.94	P
2021-10-07 00:00:00+00:00	1480.0	P	12422500	17.95	P
2021-10-08 00:00:00+00:00	1500.0	P	12422500	17.97	P
2021-10-09 00:00:00+00:00	1510.0	P	12422500	17.98	P
2021-10-10 00:00:00+00:00	1520.0	P	12422500	17.99	P
2021-10-11 00:00:00+00:00	1560.0	P	12422500	18.01	P
2021-10-12 00:00:00+00:00	1560.0	P	12422500	18.02	P
2021-10-13 00:00:00+00:00	1580.0	P	12422500	18.03	P
2021-10-14 00:00:00+00:00	1600.0	P	12422500	18.05	P
2021-10-15 00:00:00+00:00	1590.0	P	12422500	18.04	P

2021-10-16 00:00:00+00:00	1600.0	P	12422500	18.05	P
2021-10-17 00:00:00+00:00	1600.0	P	12422500	18.05	P
2021-10-18 00:00:00+00:00	1600.0	P	12422500	18.05	P
2021-10-19 00:00:00+00:00	1620.0	P	12422500	18.07	P
2021-10-20 00:00:00+00:00	1690.0	P	12422500	18.12	P
2021-10-21 00:00:00+00:00	2020.0	P	12422500	18.36	P
2021-10-22 00:00:00+00:00	2080.0	P	12422500	18.40	P
2021-10-23 00:00:00+00:00	2120.0	P	12422500	18.42	P
2021-10-24 00:00:00+00:00	2130.0	P	12422500	18.44	P
2021-10-25 00:00:00+00:00	2170.0	P	12422500	18.46	P
2021-10-26 00:00:00+00:00	2200.0	P	12422500	18.48	P
2021-10-27 00:00:00+00:00	2180.0	P	12422500	18.47	P
2021-10-28 00:00:00+00:00	2210.0	P	12422500	18.48	P
2021-10-29 00:00:00+00:00	2250.0	P	12422500	18.51	P
2021-10-30 00:00:00+00:00	2210.0	P	12422500	18.49	P
2021-10-31 00:00:00+00:00	2250.0	P	12422500	18.51	P

In [12]:

```
# Calculate Mean for Spokane
df_Spokane['00060_Mean'].mean()
```

Out[12]:

```
1710.3125
```

In [13]:

```
# Website for mean value
#https://waterdata.usgs.gov/nwis/dv?cb_00060=on&format=gif_stats&site_
```

HTML Table

Site Name	Site Number	Mean Daily Discharge
Willamette River, Portland, Oregon	14211720	13064
McKenzie River near Walterville, Oregon	14163900	1566
Spokane River at Spokane, Washington	12422500	1710

```
In [14]: # Import packages
         import numpy as np
         import pandas as pd
         import folium
```

```
In [15]: # Read HTML table data
         mountains = pd.read_html('https://en.wikipedia.org/wiki/List_of_mounta
         mountains
```

```
 18   1962.9 m      941 m   17.4 km
 19   2633.3 m      927 m   45.5 km
 20     1131 m      925 m   97.9 km
 21   2722.9 m      920 m   25.9 km
 22   2553.3 m      908 m  12.49 km
 23     2482 m      884 m   23.5 km
 24   1094.8 m      868 m   45.4 km
 25     3075 m      837 m      7 km
 26     2764 m      818 m  11.02 km
 27   1556.9 m      811 m   31.1 km
 28   2384.4 m      792 m  19.01 km
 29     2377 m      785 m  16.33 km

                                                  Location
 0   45°22′25″N 121°41′45″W / 45.3735°N 121.6959°W
 1   45°14′42″N 117°17′34″W / 45.2450°N 117.2929°W
 2   44°40′27″N 121°47′59″W / 44.6743°N 121.7996°W
 3   44°06′13″N 121°46′09″W / 44.1035°N 121.7693°W
 4   44°49′00″N 118°06′14″W / 44.8168°N 118.1039°W
 5   42°26′40″N 122°18′56″W / 42.4445°N 122.3156°W
```

```
In [16]: # Print number of tables on webpage
         len(mountains)
```

```
Out[16]: 13
```

```
In [17]: # We would like the table that contains the highest summits of Oregon
         mountain_stats = mountains[1]
```

In [18]:
```
# Some wrangling
mountain_stats['Location'] = mountain_stats['Location'].str.replace(mo
```

In [19]:
```
mountain_stats
```

Out[19]:

	Rank	Mountain peak	Mountain range	Elevation	Prominence	Isolation	Location
0	1	Mount Hood[6][7][8][9][a]	Cascade Range	3428.8 m	2349 m	92.2 km	45°22′25″N 121°41′45″W / 45.3735°N 121.6959°W
1	2	Mount Jefferson[10][11][12][13][b]	Cascade Range	3201 m	1767 m	77.5 km	44°40′27″N 121°47′59″W / 44.6743°N 121.7996°W
2	3	South Sister[14][15][16][17]	Cascade Range	3158.5 m	1705 m	63.4 km	44°06′13″N 121°46′09″W / 44.1035°N 121.7693°W
3	4	North Sister[18][19][20][21][c]	Cascade Range	3075 m	837 m	7 km	44°10′00″N 121°46′20″W / 44.1666°N 121.7723°W
4	5	Middle Sister[22][23][24][25][d]	Cascade Range	3064 m	382 m	1.8 km	44°08′54″N 121°47′02″W / 44.1483°N 121.7840°W
5	6	Sacajawea Peak[26][27][28][e][f]	Wallowa Mountains	3000 m	1949 m	202 km	45°14′42″N 117°17′34″W / 45.2450°N 117.2929°W
6	7	Steens Mountain[29][30][31][g]	Steens Mountain	2968 m	1336 m	201 km	42°38′11″N 118°34′36″W / 42.6364°N 118.5767°W
7	8	Aneroid Mountain[32][33][34][35]	Wallowa Mountains	2958.7 m	647 m	9.48 km	45°12′11″N 117°10′30″W / 45.2030°N 117.1750°W
8	9	Twin Peaks[36][37][38][h]	Wallowa Mountains	2950 m	610 m	7.79 km	45°18′17″N 117°20′43″W / 45.3046°N 117.3452°W
9	10	Red Mountain[39][40][41][42]	Wallowa Mountains	2913.8 m	610 m	11.84 km	45°03′52″N 117°14′46″W / 45.0644°N 117.2460°W
							42°26′40″N

10	11	Mount McLoughlin[43][44][45][46][i][j]	Cascade Range	2895 m	1364 m	111.8 km	122°18′56″W / 42.4445°N 122.3156°W
11	12	Elkhorn Peak[47][48][49][k]	Wallowa Mountains	2816 m	567 m	5.32 km	45°13′20″N 117°23′48″W / 45.2223°N 117.3968°W
12	13	Mount Thielsen[50][51][52][53]	Cascade Range	2799.4 m	1025 m	81.1 km	43°09′10″N 122°03′59″W / 43.1528°N 122.0665°W
13	14	Broken Top[54][55][56][l]	Cascade Range	2798 m	669 m	5.52 km	44°04′59″N 121°41′58″W / 44.0830°N 121.6994°W
14	15	Rock Creek Butte[57][58][59][m]	Elkhorn Mountains	2777 m	1364 m	69.9 km	44°49′00″N 118°06′14″W / 44.8168°N 118.1039°W
15	16	Mount Bachelor[60][61][62][63]	Cascade Range	2764 m	818 m	11.02 km	43°58′46″N 121°41′19″W / 43.9794°N 121.6885°W
16	17	Strawberry Mountain[64][65][66][67][n]	Strawberry Range	2756.1 m	1253 m	74.2 km	44°18′44″N 118°43′00″W / 44.3123°N 118.7166°W
17	18	Mount Scott[68][69][70][71]	Cascade Range	2722.9 m	920 m	25.9 km	42°55′22″N 122°00′58″W / 42.9229°N 122.0162°W
18	19	Diamond Peak[72][73][74][75]	Cascade Range	2666.4 m	952 m	41.4 km	43°31′15″N 122°08′59″W / 43.5207°N 122.1496°W
19	20	Pueblo Mountain[76][77][78][79][o]	Pueblo Mountains	2633.3 m	927 m	45.5 km	42°05′58″N 118°39′02″W / 42.0995°N 118.6506°W
20	21	Crane Mountain[80][81][82][83]	Warner Mountains	2575.8 m	718 m	71.4 km	42°03′46″N 120°14′27″W / 42.0628°N 120.2408°W
21	22	Drake Peak[84][85][86][87][p]	Warner Mountains	2564 m	779 m	28.1 km	42°18′00″N 120°07′26″W / 42.3001°N 120.1238°W
22	23	Mount Bailey[88][89][90][91][q]	Cascade Range	2553.3 m	908 m	12.49 km	43°09′18″N 122°13′12″W / 43.1551°N

							122.2200°W
23	24	Gearhart Mountain[92][93][94][95]	Gearhart Mountain	2550.6 m	1049 m	65.7 km	42°29′46″N 120°52′38″W / 42.4960°N 120.8773°W
24	25	Aspen Butte[96][97][98][99]	Cascade Range	2503.83 m	947 m	23.7 km	42°18′56″N 122°05′15″W / 42.3155°N 122.0876°W
25	26	Yamsay Mountain[100][101][102][103]	Cascade Volcanic Arc	2499.3 m	970 m	53.1 km	42°55′50″N 121°21′39″W / 42.9306°N 121.3607°W
26	27	Vinegar Hill[104][105][106][107][r]	Greenhorn Mountains	2482 m	884 m	23.5 km	44°42′50″N 118°33′42″W / 44.7138°N 118.5617°W
27	28	Pelican Butte[108][109][110][111]	Cascade Range	2449.8 m	669 m	15.98 km	42°30′48″N 122°08′43″W / 42.5134°N 122.1453°W
28	29	Lookout Mountain[112][113][114][s]	Strawberry Range	2450 m	650 m	10.73 km	44°17′20″N 118°29′43″W / 44.2889°N 118.4954°W
29	30	Warner Peak[115][116][117][118][t]	Hart Mountain	2445.8 m	648 m	35.6 km	42°27′35″N 119°44′29″W / 42.4597°N 119.7414°W
30	31	Paulina Peak[119][120][121][122][u]	Paulina Mountains	2435 m	981 m	46.5 km	43°41′21″N 121°15′18″W / 43.6892°N 121.2549°W

In [20]:

```
mountain_stats.dtypes
```

Out[20]:

```
Rank                int64
Mountain peak      object
Mountain range     object
Elevation          object
Prominence         object
Isolation          object
Location           object
dtype: object
```

In [21]:
```
# Have a look at the location object
mountain_stats['Location'].iloc[0]
```

Out[21]:
```
'45°22′25″N 121°41′45″W\ufeff / \ufeff45.3735°N 121.6959°W'
```

In [22]:
```
# The latitude is string position 27 to 34
lat1 = mountain_stats['Location'].iloc[0][27:34]

# The longitude is string position 37 to 45
lon1 = mountain_stats['Location'].iloc[0][37:45]
```

In [23]:
```
# Convert to float and multiple by -1
float(mountain_stats['Location'].iloc[0][37:45]) * -1
```

Out[23]:
```
-121.6959
```

In [24]:

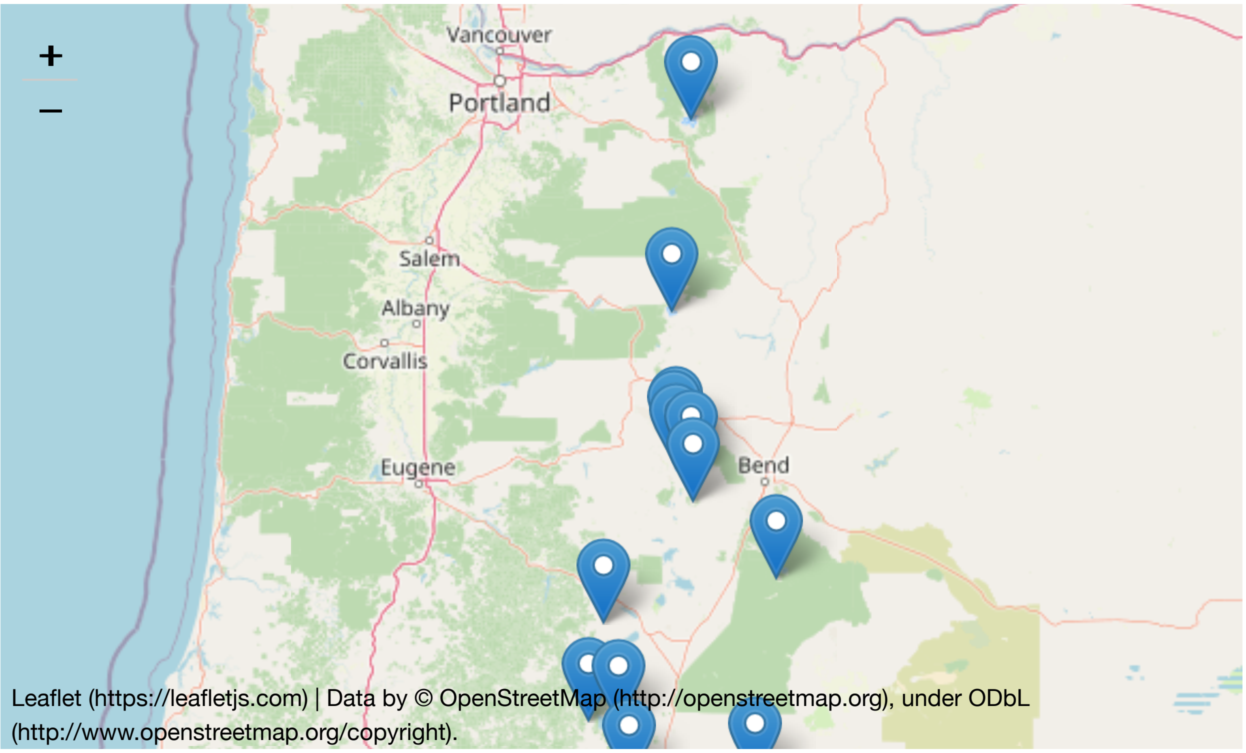
```
# To get these data from every row, we can write a quick for loop
coords = []
for i in range(len(mountain_stats)):
    lat = float(mountain_stats['Location'].iloc[i][27:34])
    lon = float(mountain_stats['Location'].iloc[i][37:45]) * -1
    coords.append((lat, lon))
coords
```

Out[24]:

```
[(45.3735, -121.6959),
 (44.6743, -121.7996),
 (44.1035, -121.7693),
 (44.1666, -121.7723),
 (44.1483, -121.784),
 (45.245, -117.2929),
 (42.6364, -118.5767),
 (45.203, -117.175),
 (45.3046, -117.3452),
 (45.0644, -117.246),
 (42.4445, -122.3156),
 (45.2223, -117.3968),
 (43.1528, -122.0665),
 (44.083, -121.6994),
 (44.8168, -118.1039),
 (43.9794, -121.6885),
 (44.3123, -118.7166),
 (42.9229, -122.0162),
 (43.5207, -122.1496),
 (42.0995, -118.6506),
 (42.0628, -120.2408),
 (42.3001, -120.1238),
 (43.1551, -122.22),
 (42.496, -120.8773),
 (42.3155, -122.0876),
 (42.9306, -121.3607),
 (44.7138, -118.5617),
 (42.5134, -122.1453),
 (44.2889, -118.4954),
 (42.4597, -119.7414),
 (43.6892, -121.2549)]
```

In [25]:
```
map = folium.Map(location=[44, -121], zoom_start=7)
for i in range(0, len(coords)):
    folium.Marker(coords[i]).add_to(map)
map
```

Out[25]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [26]:
```
# Get elevation value as a float
float(mountain_stats['Elevation'].iloc[0][:-2])
```

Out[26]:
```
3428.8
```

In [27]:

```
# To get these data from every row, we can write another quick for loo
elevation = []
for i in range(len(mountain_stats)):
    elev = float(mountain_stats['Elevation'].iloc[i][:-2])
    elevation.append(elev)
elevation
```

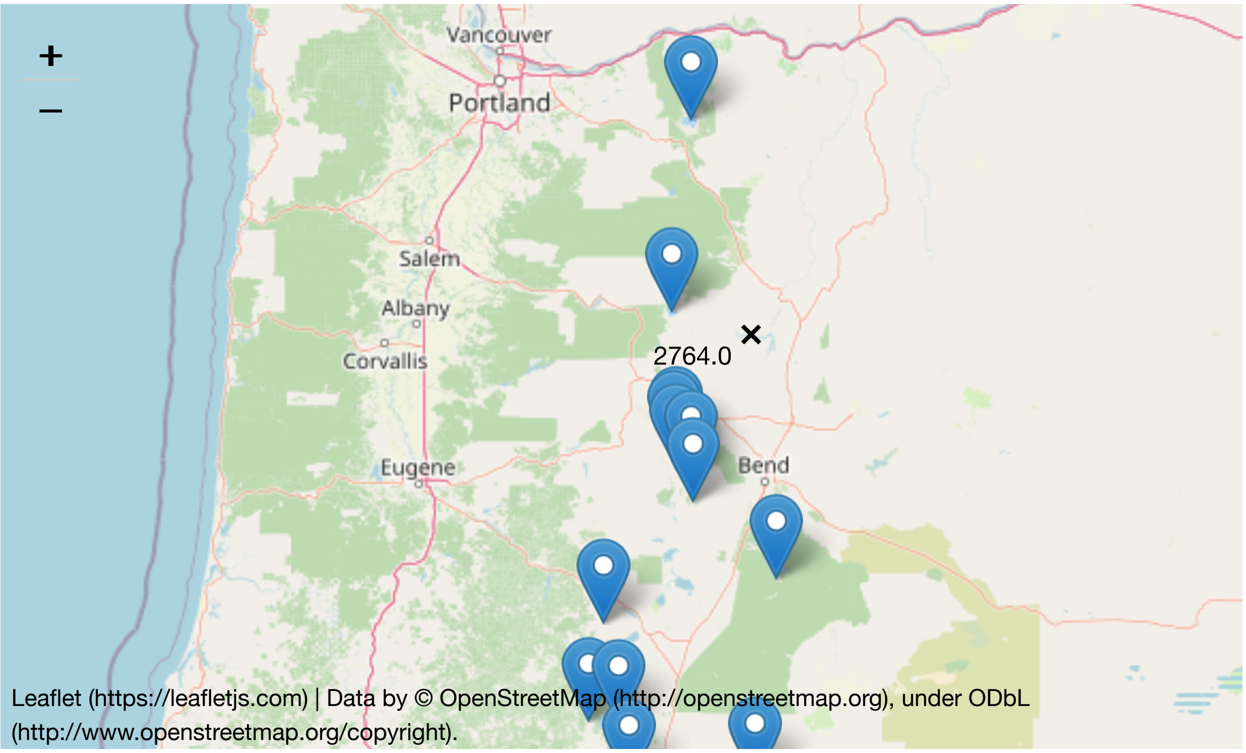
Out[27]:

```
[3428.8,
 3201.0,
 3158.5,
 3075.0,
 3064.0,
 3000.0,
 2968.0,
 2958.7,
 2950.0,
 2913.8,
 2895.0,
 2816.0,
 2799.4,
 2798.0,
 2777.0,
 2764.0,
 2756.1,
 2722.9,
 2666.4,
 2633.3,
 2575.8,
 2564.0,
 2553.3,
 2550.6,
 2503.83,
 2499.3,
 2482.0,
 2449.8,
 2450.0,
 2445.8,
 2435.0]
```

In [28]:

```
map = folium.Map(location=[44, -121], zoom_start=7)
for i in range(0, len(coords)):
    folium.Marker(coords[i], popup=elevation[i]).add_to(map)
map
```

Out[28]:


+
–
Vancouver
Portland
Salem
Albany
Corvallis
Eugene
Bend
2764.0
×
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

Question 3

In [29]:

```
mountain_stats.dtypes
```

Out[29]:

```
Rank                 int64
Mountain peak       object
Mountain range      object
Elevation           object
Prominence          object
Isolation           object
Location            object
dtype: object
```

In [30]:
```
mountain_stats.Isolation
```

Out[30]:
```
0      92.2 km
1      77.5 km
2      63.4 km
3         7 km
4       1.8 km
5       202 km
6       201 km
7      9.48 km
8      7.79 km
9     11.84 km
10    111.8 km
11     5.32 km
12     81.1 km
13     5.52 km
14     69.9 km
15    11.02 km
16     74.2 km
17     25.9 km
18     41.4 km
19     45.5 km
20     71.4 km
21     28.1 km
22    12.49 km
23     65.7 km
24     23.7 km
25     53.1 km
26     23.5 km
27    15.98 km
28    10.73 km
29     35.6 km
30     46.5 km
Name: Isolation, dtype: object
```

In [31]:
```
# Get elevation value as a float
float(mountain_stats['Isolation'].iloc[0][:-2])
```

Out[31]:
```
92.2
```

In [32]:

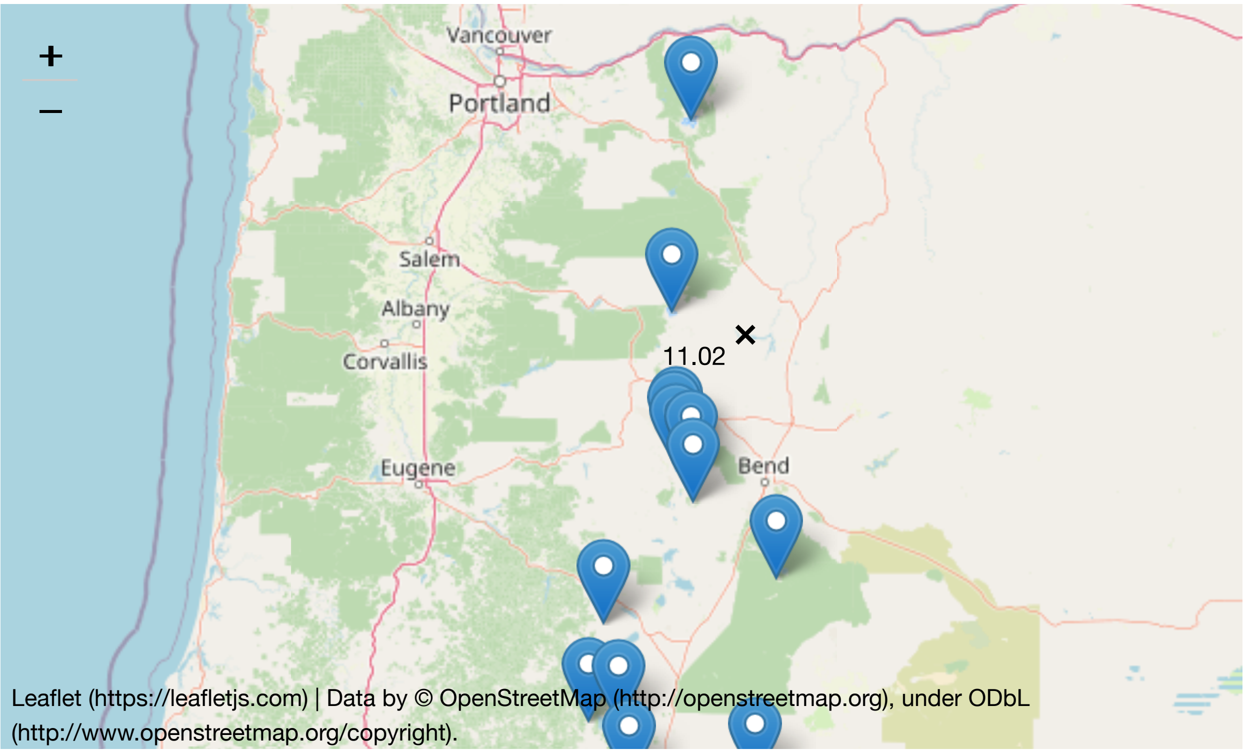
```
# Convert isolation into float
#To get these data from every row, we can write another quick for loop
isolation = []
for i in range(len(mountain_stats)):
    iso = float(mountain_stats['Isolation'].iloc[i][:-2])
    isolation.append(iso)
isolation
```

Out[32]:

```
[92.2,
 77.5,
 63.4,
 7.0,
 1.8,
 202.0,
 201.0,
 9.48,
 7.79,
 11.84,
 111.8,
 5.32,
 81.1,
 5.52,
 69.9,
 11.02,
 74.2,
 25.9,
 41.4,
 45.5,
 71.4,
 28.1,
 12.49,
 65.7,
 23.7,
 53.1,
 23.5,
 15.98,
 10.73,
 35.6,
 46.5]
```

In [33]:
```
# Displays the Isolation data
map = folium.Map(location=[44, -121], zoom_start=7)
for i in range(0, len(coords)):
    folium.Marker(coords[i], popup=isolation[i]).add_to(map)
map
```

Out[33]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

Extra Credit: Add a popup that includes the name of the mountain

In [34]:
```
# Have a look at the mountain peak
mountain_stats['Mountain peak'].iloc[0]
name = mountain_stats['Mountain peak'].iloc[0].rsplit('[')[0]
name
```

Out[34]:
```
'Mount Hood'
```

In [35]:

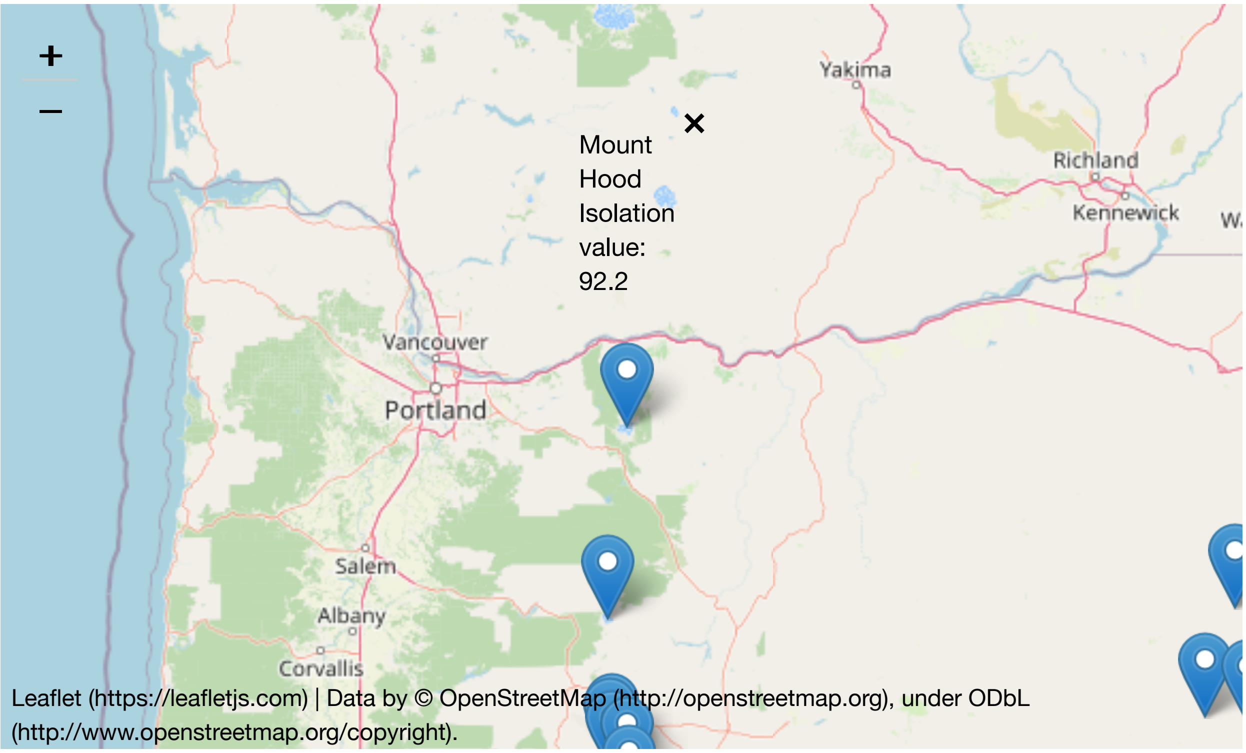
```
names =[]
for i in range(len(mountain_stats)):
    name = mountain_stats['Mountain peak'].iloc[i].rsplit('[')[0]
    names.append(name)
names
```

Out[35]:

```
['Mount Hood',
 'Mount Jefferson',
 'South Sister',
 'North Sister',
 'Middle Sister',
 'Sacajawea Peak',
 'Steens Mountain',
 'Aneroid Mountain',
 'Twin Peaks',
 'Red Mountain',
 'Mount McLoughlin',
 'Elkhorn Peak',
 'Mount Thielsen',
 'Broken Top',
 'Rock Creek Butte',
 'Mount Bachelor',
 'Strawberry Mountain',
 'Mount Scott',
 'Diamond Peak',
 'Pueblo Mountain',
 'Crane Mountain',
 'Drake Peak',
 'Mount Bailey',
 'Gearhart Mountain',
 'Aspen Butte',
 'Yamsay Mountain',
 'Vinegar Hill',
 'Pelican Butte',
 'Lookout Mountain',
 'Warner Peak',
 'Paulina Peak']
```

In [36]:
```
# Mountain name and/or isolation value
map = folium.Map(location=[44, -121], zoom_start=7)
for i in range(0, len(coords)):
    popups= names[i]  + (' Isolation value: ') + str(isolation[i])
    folium.Marker(coords[i], popup= popups).add_to(map)
map
```

Out[36]:



Mount Hood Isolation value: 92.2

Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [37]:

```
# Install webdriver_manager: https://github.com/SergeyPirogov/webdrive
!pip install -U webdriver_manager
```

```
Requirement already satisfied: webdriver_manager in /Library/Framewor
ks/Python.framework/Versions/3.9/lib/python3.9/site-packages (3.5.3)
Requirement already satisfied: requests in /Library/Frameworks/Python
.framework/Versions/3.9/lib/python3.9/site-packages (from webdriver_m
anager) (2.25.0)
Requirement already satisfied: configparser in /Library/Frameworks/Py
thon.framework/Versions/3.9/lib/python3.9/site-packages (from webdriv
er_manager) (5.2.0)
Requirement already satisfied: crayons in /Library/Frameworks/Python.
framework/Versions/3.9/lib/python3.9/site-packages (from webdriver_ma
nager) (0.4.0)
Requirement already satisfied: colorama in /Library/Frameworks/Python
.framework/Versions/3.9/lib/python3.9/site-packages (from crayons->we
bdriver_manager) (0.4.4)
Requirement already satisfied: idna<3,>=2.5 in /Library/Frameworks/Py
thon.framework/Versions/3.9/lib/python3.9/site-packages (from request
s->webdriver_manager) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Framewo
rks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from r
equests->webdriver_manager) (2020.11.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Library/Fram
eworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (fro
m requests->webdriver_manager) (1.26.2)
Requirement already satisfied: chardet<4,>=3.0.2 in /Library/Framewor
ks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from re
quests->webdriver_manager) (3.0.4)
```

In [38]:

```
# Import packages
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
```

In [39]:

```
# Install Chrome webdriver
driver = webdriver.Chrome(service=Service(ChromeDriverManager().instal

# Open a web browser at the following page
driver.get("https://en.wikipedia.org/wiki/Category:Ski_areas_and_resor
```

```
====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache
```

In [40]:

```
# Retrieve ski resort names
html_list = driver.find_element(By.ID, "mw-pages")
items = html_list.find_elements(By.TAG_NAME, "li")
```

In [41]:

```
ski_resort_names = []
for item in items:
    text = item.text
    print(text)
    ski_resort_names.append(text)
driver.close()
```

```
Anthony Lakes (ski area)
Mount Ashland Ski Area
Cooper Spur ski area
Ferguson Ridge Ski Area
Hoodoo (ski area)
Mount Ashland Ski Area Expansion
Mount Bachelor ski area
Mount Hood Meadows
Mount Hood Skibowl
Snow Bunny
Spout Springs Ski Area
Summit Pass (Oregon)
Timberline Lodge ski area
Warner Canyon
Willamette Pass Resort
```

In [42]:
```
# Define test URL
url = 'https://www.google.com/maps/place/Hoodoo+Ski+Area+Oregon/'

# Install Chrome webdriver
driver = webdriver.Chrome(service=Service(ChromeDriverManager().instal

# Open URL
driver.get(url)
```

```


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache
```

In [43]:
```
# Click search
element = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((
element.click()
```

In [44]:
```
# Retrieve the URL
link = driver.current_url

# We can find the first occurrence of a character by using the "find"
link.find('@')
```

Out[44]:
```
-1
```

In [45]:
```
print(driver.current_url)
```

```
https://www.google.com/maps/place/Hoodoo+Ski+Area+Oregon/
(https://www.google.com/maps/place/Hoodoo+Ski+Area+Oregon/)
```

In [46]:
```
split1 = link.rsplit('@', 1)
split1
```

Out[46]:
```
['https://www.google.com/maps/place/Hoodoo+Ski+Area+Oregon/']
```

Error at split2, have re-run Install Chrome webdriver

```
Deleted split2
```

In [47]:
```
lat, lon
```

Out[47]:
```
(43.6892, -121.2549)
```

In [48]:
```
driver.close()
```

In [49]:
```
ski_resort_coords = []
# Loop through every ski resort to find it's coordinates
for resort in ski_resort_names:

    # Define URL to search in Google Maps and add 'Oregon' in for good
    url = 'https://www.google.com/maps/place/' + resort + ' Oregon/'

    # Import web driver and search for ski resorts
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().in
    driver.get(url)

    # Click search
    element = WebDriverWait(driver, 20).until(EC.element_to_be_clickab
    element.click()

    # Make the web driver wait until the URL updates (i.e. contains th
    WebDriverWait(driver, 20).until(EC.url_contains("@"))

    # Retrieve the URL
    link = driver.current_url

    # Split string
    lat, lon = link.rsplit('@', 1)[1].rsplit(',', 1)[0].rsplit(',', 1)

    # Append to list
    ski_resort_coords.append((lat, lon))

    # Close driver
    driver.close()
```

```
====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache
```

```
====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache
```

In [50]:

```
ski_resort_coords
```

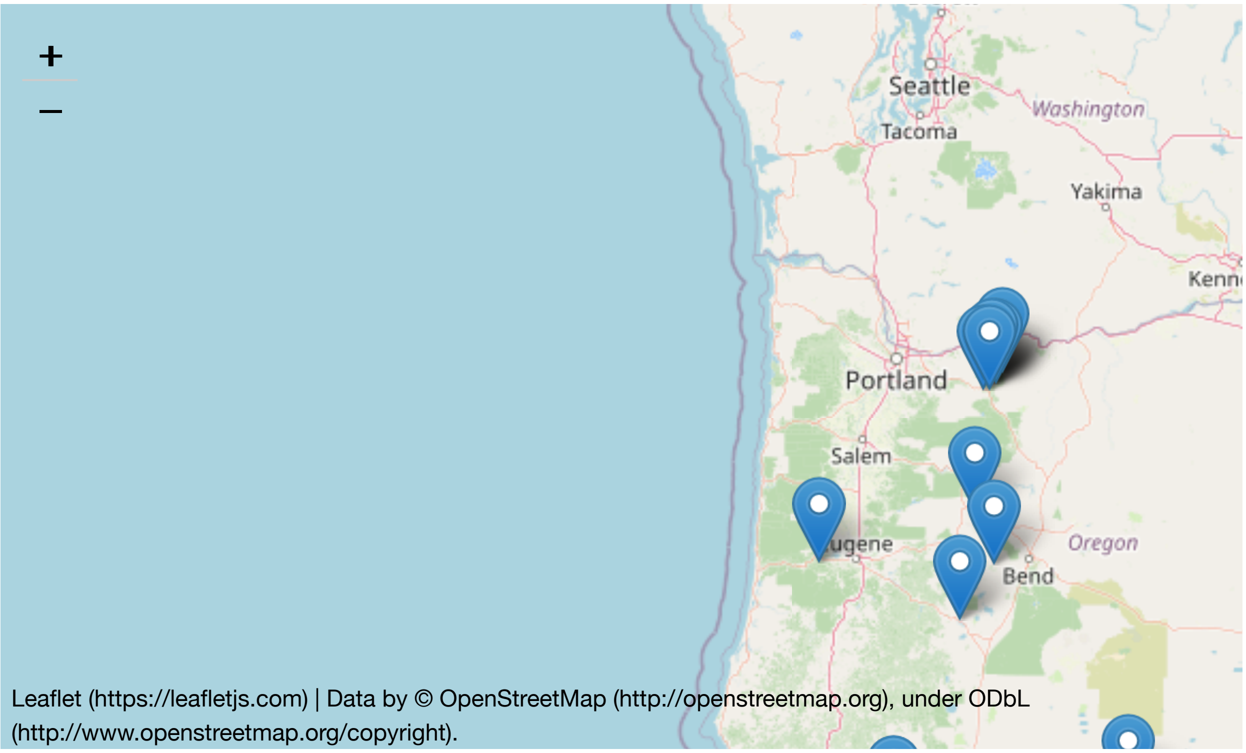
Out[50]:

```
[('44.9629235', '-118.2357129'),
 ('42.081685', '-122.7069427'),
 ('45.4188458', '-121.6064525'),
 ('45.2816851', '-117.1148305'),
 ('44.4086439', '-121.8736045'),
 ('42.081685', '-122.7069427'),
 ('44.0028937', '-121.6812601'),
 ('45.3317552', '-121.6673735'),
 ('45.2943342', '-121.7896261'),
 ('45.2871418', '-121.7312302'),
 ('45.7552425', '-118.0536097'),
 ('44.0265109', '-123.4892255'),
 ('45.3311281', '-121.7131951'),
 ('42.237374', '-120.2968271'),
 ('43.600054', '-122.0387287')]
```

In [51]:

```
map = folium.Map(location=[44, -121], zoom_start=7)
for i in range(0, len(ski_resort_coords)):
    folium.Marker(ski_resort_coords[i], popup=ski_resort_names[i]).add
map
```

Out[51]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

Question 4

Question 4 (10 points)

- Write a script to automatically derive the geographic coordinates for the following addresses:
  - 1844 SW Morrison St, Portland, OR 97205
  - 800 Occidental Ave S, Seattle, WA 98134
  - 1001 Stadium Dr, Inglewood, CA 90301
  - 2700 Martin Luther King Jr Blvd, Eugene, OR 97401

You can **either** find each one individually **or** make a list of the addresses and use a for loop.

- Plot the coordinates of these addresses on an interactive map using `folium`

In [52]:

```
location_4 = ['1844 SW Morrison St, Portland, OR 97205', '800 Occident
coordinates_4 = []
# Loop through every ski resort to find it's coordinates
for loc_4 in location_4:

    # Define URL to search in Google Maps and add 'Oregon' in for good
    url = 'https://www.google.com/maps/place/' + loc_4

    # Import web driver and search for ski resorts
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().ir
    driver.get(url)

    # Click search
    element = WebDriverWait(driver, 20).until(EC.element_to_be_clickab
    element.click()

    # Make the web driver wait until the URL updates (i.e. contains th
    WebDriverWait(driver, 20).until(EC.url_contains("@"))

    # Retrieve the URL
    link = driver.current_url

    # Split string
    lat, lon = link.rsplit('@', 1)[1].rsplit(',', 1)[0].rsplit(',', 1)

    # Append to list
    coordinates_4.append((lat, lon))

    # Close driver
    driver.close()
```

```
====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache


====== WebDriver manager ======
Current google-chrome version is 98.0.4758
Get LATEST chromedriver version for 98.0.4758 google-chrome
Driver [/Users/jack/.wdm/drivers/chromedriver/mac64/98.0.4758.102/chr
omedriver] found in cache
```

In [53]:

```
#1844 SW Morrison St, Portland, OR 97205: 45.52181664425629, -122.6907
#800 Occidental Ave S, Seattle, WA 98134: 47.59349096388847, -122.3322
#1001 Stadium Dr, Inglewood, CA 90301: 33.953165071858955, -118.338534
#2700 Martin Luther King Jr Blvd, Eugene, OR 97401: 44.059621427301096
```

In [54]:

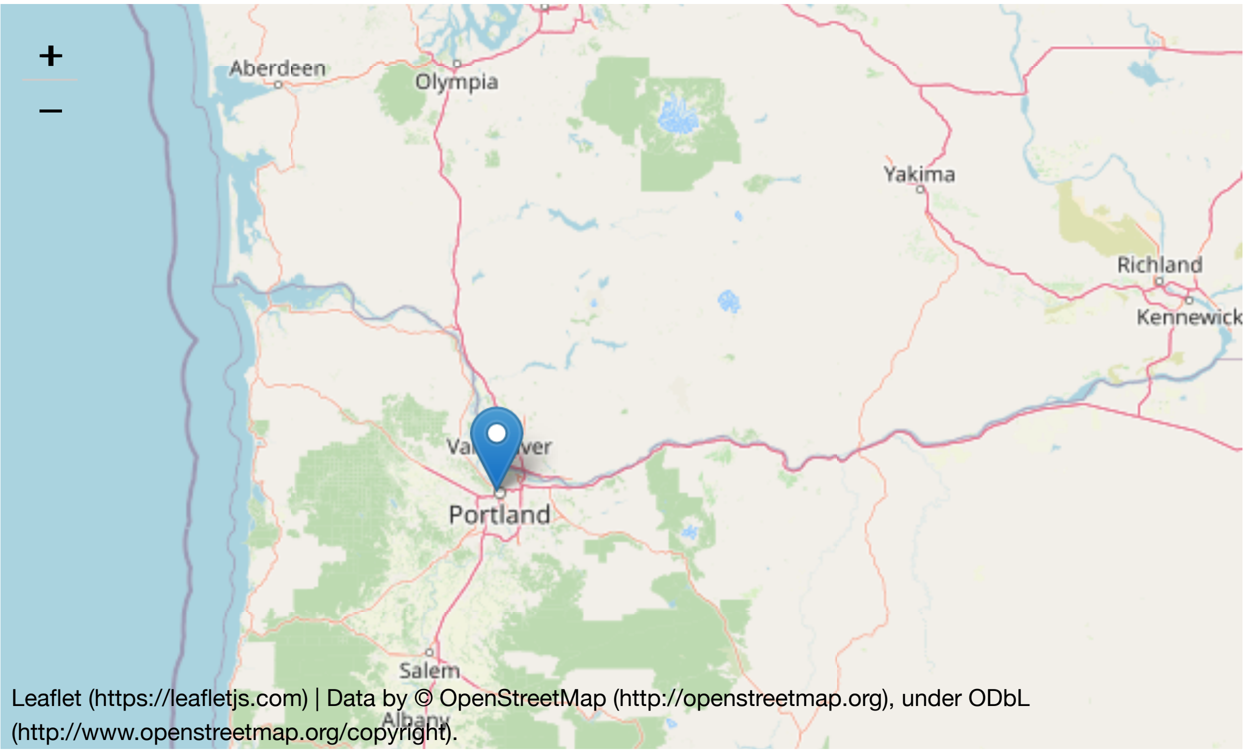
```
coordinates_4
```

Out[54]:

```
[('45.5216776', '-122.693017'),
 ('47.5933101', '-122.3344609'),
 ('33.9530049', '-118.3407129'),
 ('44.0594287', '-123.0710918')]
```

In [55]:
```
map = folium.Map(location=[45.5, -121], zoom_start=7)
for i in range(0, len(coordinates_4)):
    folium.Marker(coordinates_4[i], popup=location_4[i]).add_to(map)
map
```

Out[55]:


+
–
Aberdeen
Olympia
Yakima
Richland
Kennewick
Vancouver
Portland
Salem
Albany
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

Question 5

In [56]:
```
# Import package
import xarray as xr

# Define filepath
fp = '/Users/jack/Documents/GitHub/geospatial-data-science/labs/lab7'

# Read data
xds = xr.open_dataset(fp + '/era_monthly_snowfall_2020.nc', decode_coc
```

In [57]:

```
xds
```

Out[57]:

```
xarray.Dataset

▶ Dimensions:       (longitude: 49, latitude: 25, time: 12)

▼ Coordinates:

    longitude       (longitude)                      float32   -128.0 -127.8 ......
    latitude        (latitude)                       float32   47.0 46.75 46.5...
    time            (time)                    datetime64[ns]   2020-01-01 ... 2...

▼ Data variables:

    sf              (time, latitude, longitude)      float32   ...

▼ Attributes:

    Conventions :   CF-1.6
    history :       2022-01-30 21:04:05 GMT by grib_to_netcdf-2.23.0: /opt/ecmwf/m
                    ars-client/bin/grib_to_netcdf -S param -o /cache/data6/adaptor.ma
                    rs.internal-1643576645.547142-29574-12-6e006e1c-6452-4b43-8
                    b38-b506dd10f98b.nc /cache/tmp/6e006e1c-6452-4b43-8b38-b5
                    06dd10f98b-adaptor.mars.internal-1643576640.5525317-29574-17
                    -tmp.grib
```

In [58]:
```
xds.head()
```

Out[58]:
```
xarray.Dataset

▶ Dimensions:      (longitude: 5, latitude: 5, time: 5)

▼ Coordinates:

   longitude      (longitude)                     float32  -128.0 -127.8 -...
   latitude       (latitude)                      float32  47.0 46.75 46.5...
   time           (time)                   datetime64[ns]  2020-01-01 ... 2...

▼ Data variables:

   sf             (time, latitude, longitude)     float32  -4.657e-10 3.79...

▼ Attributes:

   Conventions :  CF-1.6
   history :      2022-01-30 21:04:05 GMT by grib_to_netcdf-2.23.0: /opt/ecmwf/m
                  ars-client/bin/grib_to_netcdf -S param -o /cache/data6/adaptor.ma
                  rs.internal-1643576645.547142-29574-12-6e006e1c-6452-4b43-8
                  b38-b506dd10f98b.nc /cache/tmp/6e006e1c-6452-4b43-8b38-b5
                  06dd10f98b-adaptor.mars.internal-1643576640.5525317-29574-17
                  -tmp.grib
```

In [59]:
```
ashland = xds.sel(latitude = 42.081685, longitude= -122.7069427, metho
hoodoo = xds.sel(latitude =44.4086439, longitude= -121.8736045, methoc
willammette = xds.sel(latitude =43.600054, longitude= -122.0387287, me
```

In [60]:
```
ashland['sf'].values.sum()
```

Out[60]:
```
0.009228621
```

In [61]:
```
hoodoo['sf'].values.sum()
```

Out[61]:
```
0.018596929
```

In [62]:
```
willammette['sf'].values.sum()
```

Out[62]:
```
0.019636936
```

The ski resort that received more snowfall is Willammette Pass.

Extra Credit (Did not finished)

In [63]:

```
# Import package
import xarray as xr

# Define filepath
fp = '/Users/jack/Documents/GitHub/geospatial-data-science/labs/lab7'

# Read data
xds_2 = xr.open_dataset(fp + '/era_monthly_snowfall_1979_2020.nc', dec
```

In [64]:

```
xds_2
```

Out[64]:

```
xarray.Dataset

▶ Dimensions:      (longitude: 49, latitude: 25, time: 504)

▼ Coordinates:

   longitude       (longitude)                    float32   -128.0 -127.8 ......
   latitude        (latitude)                     float32   47.0 46.75 46.5...
   time            (time)                  datetime64[ns]   1979-01-01 ... 2...

▼ Data variables:

   sf              (time, latitude, longitude)    float32   ...

▼ Attributes:

   Conventions :   CF-1.6
   history :       2022-01-30 21:07:38 GMT by grib_to_netcdf-2.23.0: /opt/ecmwf/m
                   ars-client/bin/grib_to_netcdf -S param -o /cache/data4/adaptor.ma
                   rs.internal-1643576857.706256-30892-9-b95be943-bb21-4f41-943
                   1-360954ab03da.nc /cache/tmp/b95be943-bb21-4f41-9431-36095
                   4ab03da-adaptor.mars.internal-1643576690.9933307-30892-10-tm
                   p.grib
```