ICE: Sources of Data

Name: Cabot

DATA 3300

In this in-class exercise we will examine different ways that we can source data, beginning with precompiled datasets, followed by web APIs, and then finally we will produce our own dataset from a class survey.

Whenever I am working on putting together a dataset, I do so with the intention of solving a particular problem or answering a particular question. This will guide what data I need to source!

So, let's say we're starting a Travel Agency business to help individuals plan optimal trips to National Parks, what sort of data might we want to make data-driven recommendations?

- customer level data - primary data collection - survey
- park level data - secondary data

Part A) Primary Data Collection

Surveys are one common way of collecting data, and it can be a helpful way for gathering information from specific individuals. Let's create a survey with five or so questions we think will help us plan a National Parks Holiday trip!

These questions should help us narrow down their preferences around:

- Region of the country to visit
- Preference around traffic
- Type of sites preferred
- Trail activity type preference
- Weather preferences

In [84]:
```
import pandas as pd
import requests
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
```

Part B) Secondary Data Collection

Let's begin with examining some sources of precompiled datasets. Websites I often use for sourcing datasets include:

- Government websites, like Transparent Utah
- Kaggle
- Data is Plural

Let's see what precompiled data is out there that might be useful related to finding info on national parks!

National Parks Dataset

National Park Trails

In [85]:
```
parks = pd.read_csv('df_2.csv') # load in data set
parks.head()
```

Out[85]:

	Unnamed: 0	Name	Image	Location	Date established as park[7][12]	Area (2021)[13]	Recreation visitors (2021)[11]	Description
0	0	Acadia	NaN	Maine.mw-parser-output .geo-default,.mw-parser...	26-Feb-19	49,071.40 acres (198.6 km2)	4069098	Covering most of Mount Desert Island and other...
1	1	American Samoa	NaN	American Samoa14°15′S 170°41′W / 14.25°S 170...	31-Oct-88	8,256.67 acres (33.4 km2)	8495	The southernmost national park is on three Sam...
2	2	Arches	NaN	Utah38°41′N 109°34′W / 38.68°N 109.57°W	12-Nov-71	76,678.98 acres (310.3 km2)	1806865	This site features more than 2,000 natural san...
3	3	Badlands	NaN	South Dakota43°45′N 102°30′W / 43.75°N 102.50°W	10-Nov-78	242,755.94 acres (982.4 km2)	1224226	The Badlands are a collection of buttes, pinna...
4	4	Big Bend	NaN	Texas29°15′N 103°15′W / 29.25°N 103.25°W	12-Jun-44	801,163.21 acres (3,242.2 km2)	581220	Named for the prominent bend in the Rio Grande...

1) Narrow Down the Dataset: Let's filter down this dataset some to parks of interest based on their Location -- Assume we want to stick to locations in the Intermountain West. How can we use the Location columns to narrow down our list?

In [86]:
```
# filtered_parks = parks[parks['Location'].str.startswith(('Utah', 'Colorado', 'Wyoming', 'Montana', 'Idaho', 'Arizor
filtered_parks = parks[parks['Location'].str.contains('Utah|Colorado|Wyoming|Montana|Idaho|Arizona', case=False)]
filtered_parks.head()
```

Out[86]:

	Unnamed: 0	Name	Image	Location	Date established as park[7][12]	Area (2021) [13]	Recreation visitors (2021)[11]	Description
2	2	Arches	NaN	Utah38°41′N 109°34′W / 38.68°N 109.57°W	12-Nov-71	76,678.98 acres (310.3 km2)	1806865	This site features more than 2,000 natural san...
6	6	Black Canyon of the Gunnison	NaN	Colorado38°34′N 107°43′W / 38.57°N 107.72°W	21-Oct-99	30,779.83 acres (124.6 km2)	308910	The park protects a quarter of the Gunnison Ri...
7	7	Bryce Canyon	NaN	Utah37°34′N 112°11′W / 37.57°N 112.18°W	25-Feb-28	35,835.08 acres (145.0 km2)	2104600	Bryce Canyon is a geological amphitheater on s...
8	8	Canyonlands	NaN	Utah38°12′N 109°56′W / 38.2°N 109.93°W	12-Sep-64	337,597.83 acres (1,366.2 km2)	911594	This landscape was eroded into a maze of canyo...
9	9	Capitol Reef	NaN	Utah38°12′N 111°10′W / 38.20°N 111.17°W	18-Dec-71	241,904.50 acres (979.0 km2)	1405353	The park's Waterpocket Fold is a 100-mile (160...

In [87]:
```
# view the parks within those locations
filtered_parks['Name'].value_counts()
```

Out[87]:
```
Name
Arches                          1
Black Canyon of the Gunnison    1
Bryce Canyon                    1
Canyonlands                     1
Capitol Reef                    1
Glacier                         1
Grand Canyon *                  1
Grand Teton                     1
Great Sand Dunes                1
Mesa Verde *                    1
Petrified Forest                1
Rocky Mountain                  1
Saguaro                         1
Yellowstone                     1
Zion                            1
Name: count, dtype: int64
```

2) If we know our customer is interested in specific geographic sites or types of sites, how can we further filter down parks?

see desc_filter variable below

In [88]:
```
mountain_parks = filtered_parks[filtered_parks['Description'].str.contains('mountain', case=False)] # add in descript
mountain_parks
```

Out[88]:

	Unnamed: 0	Name	Image	Location	Date established as park[7][12]	Area (2021) [13]	Recreation visitors (2021)[11]	Description
21	21	Glacier	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...
24	24	Grand Teton	NaN	Wyoming43°44′N 110°48′W / 43.73°N 110.80°W	26-Feb-29	310,044.36 acres (1,254.7 km2)	3885230	Grand Teton is the tallest mountain in the sce...
26	26	Great Sand Dunes	NaN	Colorado37°44′N 105°31′W / 37.73°N 105.51°W	24-Sep-04	107,345.73 acres (434.4 km2)	602613	The tallest sand dunes in North America, up to...
50	50	Rocky Mountain	NaN	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	26-Jan-15	265,807.24 acres (1,075.7 km2)	4434848	Bisected north to south by the Continental Div...
51	51	Saguaro	NaN	Arizona32°15′N 110°30′W / 32.25°N 110.50°W	14-Oct-94	92,867.42 acres (375.8 km2)	1079786	Split into the separate Rincon Mountain and Tu...
60	60	Yellowstone	NaN	Wyoming, Montana, Idaho	March 1, 1872	2,219,790.71 acres (8,983.2 km2)	4860242	Situated on the Yellowstone Caldera, the park ...

In [89]:
```
mountain_parks['Name'].value_counts() # check the parks count
```

Out[89]:
```
Name
Glacier             1
Grand Teton         1
Great Sand Dunes    1
Rocky Mountain      1
Saguaro             1
Yellowstone         1
Name: count, dtype: int64
```

3) Now let's Incorporate our Trail Data! What pieces of information tie together these two datasets?

In [90]:
```
pd.set_option('display.max_columns', None)
```

In [91]:
```
trails = pd.read_csv('NPS_-_Trails_-_Geographic_Coordinate_System.csv') # read in second dataset
trails.head()
```

Out[91]:

	OBJECTID	TRLNAME	TRLALTNAME	MAPLABEL	TRLSTATUS	TRLSURFACE	TRLTYPE	TRLCLASS	TRLUSE	PUBLICDI
0	1	Jumbo Mine Trail	NaN	Jumbo Mine Trail	Existing	Unknown	Standard Terra Trail	Class 2: Moderately Developed	Hiker/Pedestrian	Publi D
1	2	Jumbo Mine Trail	NaN	Jumbo Mine Trail	Existing	Unknown	Standard Terra Trail	Class 2: Moderately Developed	Hiker/Pedestrian	Publi D
2	3	Jumbo Mine Trail	NaN	Jumbo Mine Trail	Existing	Unknown	Standard Terra Trail	Class 2: Moderately Developed	Hiker/Pedestrian	Publi D
3	4	Jumbo Mine Trail	NaN	Jumbo Mine Trail	Existing	Unknown	Standard Terra Trail	Class 2: Moderately Developed	Hiker/Pedestrian	Publi D
4	5	Jumbo Mine Trail	NaN	Jumbo Mine Trail	Existing	Unknown	Standard Terra Trail	Class 2: Moderately Developed	Hiker/Pedestrian	Publi D

4) Join the datasets by mapping Name to Unitcode then joining on the same primary key!

In [92]:
```
parks_nps_codes = {
    "Glacier": "GLAC",
    "Grand Teton": "GRTE",
    "Great Sand Dunes": "GRSA",
    "Rocky Mountain": "ROMO",
    "Saguaro": "SAGU",
    "Yellowstone": "YELL"
}

# Rename the 'Name' column to 'UNITCODE' and replace values using parks_nps_codes
mountain_parks = mountain_parks.rename(columns={'Name': 'UNITCODE'}) # fill in rename argument
# we really should keep the name column as it provides a lot of information...
# it would be super simple to just create a new column

mountain_parks['UNITCODE'] = mountain_parks['UNITCODE'].replace(parks_nps_codes) # fill in replace argument
mountain_parks.head()
```

Out[92]:

	Unnamed: 0	UNITCODE	Image	Location	Date established as park[7][12]	Area (2021)[13]	Recreation visitors (2021)[11]	Description
21	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...
24	24	GRTE	NaN	Wyoming43°44′N 110°48′W / 43.73°N 110.80°W	26-Feb-29	310,044.36 acres (1,254.7 km2)	3885230	Grand Teton is the tallest mountain in the sce...
26	26	GRSA	NaN	Colorado37°44′N 105°31′W / 37.73°N 105.51°W	24-Sep-04	107,345.73 acres (434.4 km2)	602613	The tallest sand dunes in North America, up to...
50	50	ROMO	NaN	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	26-Jan-15	265,807.24 acres (1,075.7 km2)	4434848	Bisected north to south by the Continental Div...
51	51	SAGU	NaN	Arizona32°15′N 110°30′W / 32.25°N 110.50°W	14-Oct-94	92,867.42 acres (375.8 km2)	1079786	Split into the separate Rincon Mountain and Tu...

In [93]:
```
print(len(trails))
print(len(mountain_parks))
```

```
30527
6
```

In [94]:
```
merged_df = pd.merge(mountain_parks, trails, on='UNITCODE', how='inner')# create new dataframe by merging mountain_pa
merged_df.head()
```

Out[94]:

	Unnamed: 0	UNITCODE	Image	Location	Date established as park[7] [12]	Area (2021) [13]	Recreation visitors (2021)[11]	Description	OBJECTID	TRLNAME
0	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	359	Quartz Creek
1	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	365	Martha's Basin
2	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	369	Piegan Pass
3	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	399	North Shore Josephine Lake
4	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	406	Boulder Pass Campground Spur

In [95]:
```
# view all columns in merged_df
merged_df.columns
```

Out[95]:
```
Index(['Unnamed: 0', 'UNITCODE', 'Image', 'Location',
       'Date established as park[7][12]', 'Area (2021)[13]',
       'Recreation visitors (2021)[11]', 'Description', 'OBJECTID', 'TRLNAME',
       'TRLALTNAME', 'MAPLABEL', 'TRLSTATUS', 'TRLSURFACE', 'TRLTYPE',
       'TRLCLASS', 'TRLUSE', 'PUBLICDISPLAY', 'DATAACCESS', 'ACCESSNOTES',
       'ORIGINATOR', 'UNITNAME', 'UNITTYPE', 'GROUPCODE', 'GROUPNAME',
       'REGIONCODE', 'CREATEDATE', 'EDITDATE', 'LINETYPE', 'MAPMETHOD',
       'MAPSOURCE', 'SOURCEDATE', 'XYACCURACY', 'GEOMETRYID', 'FEATUREID',
       'FACLOCID', 'FACASSETID', 'IMLOCID', 'OBSERVABLE', 'ISEXTANT',
       'OPENTOPUBLIC', 'ALTLANGNAME', 'ALTLANG', 'SEASONAL', 'SEASDESC',
       'MAINTAINER', 'NOTES', 'StagingTable', 'ShapeSTLength'],
      dtype='object')
```

5) What has changed about the structure of the dataset? (Apart from adding new columns from the join)?

- we may be "missing" data if it wasn't in both tables
- we have established a one to many relationship i.e. trail level information

Part C) OpenWeatherMaps API - Let's Check the Forecast at our Parks of Interest!

To access the OpenWeatherMap API, first navigate here and sign up for an account!

Then we will navigate to the `My API Keys` page under your profile, to generate an API key. This is all we need (aprt from the request/endpoint) to start pulling data from this website!

1) Store API Credentials

Note: This isn't a great way to do this. But since this is a free account we're not too worried about security.

In [96]:
```
# API Key (replace 'your_api_key' with the actual API key)
api_key = '42e919afac4c1f87a02bbbff6fab2f54' #9c9681d1ee3a680fd15053968fa4c0a5'
```

2) Set up your request!

Here, the goal is to add the average temperature for each of our six parks into our dataset, using the Openweather maps API.

In [97]:
```
parks_info = {
    "GLAC": {"lat": 48.6966, "lon": -113.7183},
    "GRTE": {"lat": 43.7904, "lon": -110.6818},
    "GRSA": {"lat": 37.7926, "lon": -105.5943},
    "ROMO": {"lat": 40.3428, "lon": -105.6836},
    "SAGU": {"lat": 32.2967, "lon": -111.1666},
    "YELL": {"lat": 44.4280, "lon": -110.5885}
}

# provides location data for each national park in our dataset
```

In [98]:
```
# Add comments
avg_temp_summary = []

# iterate through each park
for unit_code, coords in parks_info.items():
    lat, lon = coords['lat'], coords['lon']

    # create url for each park, this is not the request
    url = f"http://api.openweathermap.org/data/2.5/forecast?lat={lat}&lon={lon}&appid={api_key}&units=imperial"

    # request the data and get the data if successful otherwise return a code in the 400s
    response = requests.get(url)

    if response.status_code == 200:
        forecast_data = response.json()
        forecast_list = forecast_data['list']

        # extract out date and time
        df_forecast = pd.DataFrame([{
            'datetime': item['dt_txt'],
            'temperature': item['main']['temp']
        } for item in forecast_list])  # FYI this will have performance issues if your data scales.

        # convert to a date in pandas
        df_forecast['datetime'] = pd.to_datetime(df_forecast['datetime'])

        # for the five days of weather data, create 5 day avg.
        df_forecast['date'] = df_forecast['datetime'].dt.date
```

```
        daily_avg_temps = df_forecast.groupby('date')['temperature'].mean().reset_index()

        #
        five_day_avg_temp = daily_avg_temps['temperature'].mean()

         #
        avg_temp_summary.append({'UNITCODE': unit_code, 'five_day_avg_temp': five_day_avg_temp})

    else:
        print(f"Failed to retrieve data for {unit_code}: Status code", response.status_code)


df_avg_temp_summary = pd.DataFrame(avg_temp_summary) # Create a DataFrame from the summary list


merged_df = merged_df.merge(df_avg_temp_summary, on='UNITCODE', how='left') # Merge the summary DataFrame with your e


merged_df.head()
```

Out[98]:

	Unnamed: 0	UNITCODE	Image	Location	Date established as park[7] [12]	Area (2021) [13]	Recreation visitors (2021)[11]	Description	OBJECTID	TRLNAME
0	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	359	Quartz Creek
1	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	365	Martha's Basin
2	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	369	Piegan Pass
3	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	399	North Shore Josephine Lake
4	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	406	Boulder Pass Campground Spur

3) Why are the values in five_day_avg_temp repeated?

- because its a one -> many relationship

In [99]:
```
merged_df['UNITCODE'].value_counts()

len(merged_df)
```

Out[99]: 3659

4) What is the difference between an inner (what we did the first time) and left merge (what we just did)?

- Inner Join: All common records between the dataframes even on the specified column
- Left Join: All records in the left table are returned even if there isn't a matching record in the other table

*** In Pandas NULLs are matched with eachother unlike in SQL ***

In [100…
```
merged_df.head()
```

Out[100…

	Unnamed: 0	UNITCODE	Image	Location	Date established as park[7][12]	Area (2021)[13]	Recreation visitors (2021)[11]	Description	OBJECTID	TRLNAME
0	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	359	Quartz Creek
1	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	365	Martha's Basin
2	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	369	Piegan Pass
3	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	399	North Shore Josephine Lake
4	21	GLAC	NaN	Montana48°48′N 114°00′W / 48.80°N 114.00°W	11-May-10	1,013,126.39 acres (4,100.0 km2)	3081656	The U.S. half of Waterton-Glacier Internationa...	406	Boulder Pass Campground Spur

Part D) Build a Park & Trail Recommender!

In [101…

```
def park_recommender(activity, state, n=5, popular=True):
    # filter down to parks that start with requested state
```

```
    state_parks = merged_df[merged_df['Location'].str.startswith(state, na=False)]

    # filter by trailer use
    activity_parks = state_parks[state_parks['TRLUSE'].str.contains(activity, case=False, na=False)]

    # sort by populatrity
    activity_parks = activity_parks.sort_values(
        'Recreation visitors (2021)[11]',
        ascending=not popular
    )

    # return top results
    top_parks = activity_parks.head(n)

    return top_parks[['UNITCODE', 'Location', 'TRLNAME', 'five_day_avg_temp', 'TRLSURFACE']]
```

In [102…

```
recommendations = park_recommender("hik", "Colorado", n=10)
recommendations
```

Out[102...

	UNITCODE	Location	TRLNAME	five_day_avg_temp	TRLSURFACE
1076	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	New Storm Pass	7.0805	Earth
1168	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	The Pool/Odessa Lake	7.0805	Earth
1166	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Moraine Park Museum Trails	7.0805	Earth
1165	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Fall River - Little Horseshoe Park Complex Trail	7.0805	Earth
1164	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Continental Divide National Scenic Trail - Bow...	7.0805	Earth
1163	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Upper Lawn Lake Trail	7.0805	Gravel
1162	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	North Boundary, Upper	7.0805	Earth
1161	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Old Entrance Livery Trails	7.0805	Earth
1160	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	Moraine Park Museum Trails	7.0805	Earth
1159	ROMO	Colorado40°24′N 105°35′W / 40.40°N 105.58°W	North Inlet-Flattop Junction	7.0805	Earth

1) Describe how the recommender works -- why have these particular trails been recommended?

- its in colorado and contains hike

2) Does this recommender suit the needs of our client (based on the data we wanted to collect from them)? Explain.

- Not really, could be more detailed...
- need more information on type of difficulty, trail conditions, current weather conditions (is there snow).

3) How can we improve our approach? Should we be asking different questions in our survey? Trying to source different data? Trying to change our recommender function?

- Change it to only show what parks contain that information and show how many trails provide the requirements at each park
- current ground conditions (is there snow)
- preference for weather temperature
- distance/travel time from them
- could actually use the weather data