```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES
# TO THE CORRECT LOCATION (/kaggle/input) IN YOUR NOTEBOOK,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.

import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil

CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = 'smartphone-decimeter-2022:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-competitions-data%2Fkaggle-v2

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass

for data_source_mapping in DATA_SOURCE_MAPPING.split(','):
    directory, download_url_encoded = data_source_mapping.split(':')
    download_url = unquote(download_url_encoded)
    filename = urlparse(download_url).path
    destination_path = os.path.join(KAGGLE_INPUT_PATH, directory)
    try:
        with urlopen(download_url) as fileres, NamedTemporaryFile() as tfile:
            total_length = fileres.headers['content-length']
            print(f'Downloading {directory}, {total_length} bytes compressed')
            dl = 0
            data = fileres.read(CHUNK_SIZE)
            while len(data) > 0:
                dl += len(data)
                tfile.write(data)
                done = int(50 * dl / int(total_length))
                sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {dl} bytes downloaded")
                sys.stdout.flush()
                data = fileres.read(CHUNK_SIZE)
            if filename.endswith('.zip'):
              with ZipFile(tfile) as zfile:
                zfile.extractall(destination_path)
            else:
              with tarfile.open(tfile.name) as tarfile:
                tarfile.extractall(destination_path)
            print(f'\nDownloaded and uncompressed: {directory}')
    except HTTPError as e:
        print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
        continue
    except OSError as e:
        print(f'Failed to load {download_url} to path {destination_path}')
        continue

print('Data source import complete.')
```

```
Downloading smartphone-decimeter-2022, 5270329079 bytes compressed
[==================================================] 5270329079 bytes downloaded
Downloaded and uncompressed: smartphone-decimeter-2022
Data source import complete.
```

```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in /usr/local/lib/python3.10/dist-packages (4.4.0)
Requirement already satisfied: numpy>=1.17.0 in /usr/local/lib/python3.10/dist-packages (from lightgbm) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from lightgbm) (1.13.1)
```

Import

```
pip install shap
```

```
Collecting shap
  Downloading shap-0.46.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from shap) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from shap) (1.13.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from shap) (1.3.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from shap) (2.1.4)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-packages (from shap) (4.66.5)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packages (from shap) (24.1)
Collecting slicer==0.0.8 (from shap)
  Downloading slicer-0.0.8-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from shap) (0.60.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->shap)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2024.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (1.4.2
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas-
Downloading shap-0.46.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 540.1/540.1 kB 41.0 MB/s eta 0:00:00
Downloading slicer-0.0.8-py3-none-any.whl (15 kB)
Installing collected packages: slicer, shap
Successfully installed shap-0.46.0 slicer-0.0.8
```

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import os
!pip install folium
!pip install simdkalman
import pickle
import sys
import warnings
from glob import glob
import requests
import folium
from shapely.geometry import Point, shape
import shapely.wkt
from geopandas import GeoDataFrame
import simdkalman
import shap
import xgboost
from scipy.stats import spearmanr
from sklearn.ensemble import (
    ExtraTreesRegressor,
    GradientBoostingRegressor,
    RandomForestRegressor,
)
from sklearn.metrics import accuracy_score, mean_squared_error
from tqdm.notebook import tqdm
pd.options.mode.use_inf_as_na = True
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.10/dist-packages (0.17.0)
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from folium) (0.7.2)
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from folium) (3.1.4)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from folium) (1.26.4)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from folium) (2.32.3)
Requirement already satisfied: xyzservices in /usr/local/lib/python3.10/dist-packages (from folium) (2024.6.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2>=2.9->folium) (2.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->foliu
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (20
Requirement already satisfied: simdkalman in /usr/local/lib/python3.10/dist-packages (1.0.4)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from simdkalman) (1.26.4)
<ipython-input-8-d96e97a965a9>:29: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future ver
  pd.options.mode.use_inf_as_na = True
```

```
for dirname, _,filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname,filename))
```

```
/kaggle/input/smartphone-decimeter-2022/test/2022-03-17-US-SJC-1/GooglePixel5/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-03-17-US-SJC-1/GooglePixel5/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-09-20-US-MTV-2/GooglePixel4/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-09-20-US-MTV-2/GooglePixel4/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-09-20-US-MTV-2/GooglePixel4/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-09-20-US-MTV-2/GooglePixel4/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-11-30-US-MTV-1/GooglePixel5/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-11-30-US-MTV-1/GooglePixel5/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-11-30-US-MTV-1/GooglePixel5/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-11-30-US-MTV-1/GooglePixel5/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-08-24-US-SVL-2/GooglePixel5/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-08-24-US-SVL-2/GooglePixel5/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-08-24-US-SVL-2/GooglePixel5/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-08-24-US-SVL-2/GooglePixel5/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-06-22-US-MTV-1/XiaomiMi8/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-06-22-US-MTV-1/XiaomiMi8/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-06-22-US-MTV-1/XiaomiMi8/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-06-22-US-MTV-1/XiaomiMi8/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-01-26-US-MTV-1/XiaomiMi8/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-01-26-US-MTV-1/XiaomiMi8/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-01-26-US-MTV-1/XiaomiMi8/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-01-26-US-MTV-1/XiaomiMi8/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-04-25-US-OAK-2/GooglePixel4/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-25-US-OAK-2/GooglePixel4/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-25-US-OAK-2/GooglePixel4/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-04-25-US-OAK-2/GooglePixel4/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-1/SamsungGalaxyS20Ultra/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-1/SamsungGalaxyS20Ultra/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-1/SamsungGalaxyS20Ultra/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-1/SamsungGalaxyS20Ultra/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-3/XiaomiMi8/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-3/XiaomiMi8/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-3/XiaomiMi8/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-04-01-US-LAX-3/XiaomiMi8/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-08-12-US-MTV-1/GooglePixel4/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-08-12-US-MTV-1/GooglePixel4/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-08-12-US-MTV-1/GooglePixel4/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-08-12-US-MTV-1/GooglePixel4/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-01-11-US-MTV-1/GooglePixel6Pro/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-01-11-US-MTV-1/GooglePixel6Pro/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-01-11-US-MTV-1/GooglePixel6Pro/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-01-11-US-MTV-1/GooglePixel6Pro/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-11-05-US-MTV-1/XiaomiMi8/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-11-05-US-MTV-1/XiaomiMi8/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-11-05-US-MTV-1/XiaomiMi8/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-11-05-US-MTV-1/XiaomiMi8/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2021-09-28-US-MTV-1/GooglePixel5/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-09-28-US-MTV-1/GooglePixel5/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2021-09-28-US-MTV-1/GooglePixel5/supplemental/gnss_rinex.21o
/kaggle/input/smartphone-decimeter-2022/test/2021-09-28-US-MTV-1/GooglePixel5/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-03-31-US-LAX-3/SamsungGalaxyS20Ultra/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-03-31-US-LAX-3/SamsungGalaxyS20Ultra/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-03-31-US-LAX-3/SamsungGalaxyS20Ultra/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-03-31-US-LAX-3/SamsungGalaxyS20Ultra/supplemental/gnss_log.txt
/kaggle/input/smartphone-decimeter-2022/test/2022-03-14-US-MTV-1/GooglePixel5/device_gnss.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-03-14-US-MTV-1/GooglePixel5/device_imu.csv
/kaggle/input/smartphone-decimeter-2022/test/2022-03-14-US-MTV-1/GooglePixel5/supplemental/gnss_rinex.22o
/kaggle/input/smartphone-decimeter-2022/test/2022-03-14-US-MTV-1/GooglePixel5/supplemental/gnss_log.txt
```

```
import os
os.listdir('/kaggle/input/')
```

```
['smartphone-decimeter-2022']
```

```
data=pd.read_csv('../input/smartphone-decimeter-2022/sample_submission.csv')
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
data.head()
```

	tripId	UnixTimeMillis	LatitudeDegrees	LongitudeDegrees
0	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650832999	37.904611	-86.481078
1	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650833999	37.904611	-86.481078
2	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650834999	37.904611	-86.481078
3	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650835999	37.904611	-86.481078
4	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650836999	37.904611	-86.481078

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66097 entries, 0 to 66096
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   tripId            66097 non-null  object 
 1   UnixTimeMillis    66097 non-null  int64  
 2   LatitudeDegrees   66097 non-null  float64
 3   LongitudeDegrees  66097 non-null  float64
dtypes: float64(2), int64(1), object(1)
memory usage: 2.0+ MB
```

```
data['tripId'].value_counts()
```

tripId	count
2022-02-24-US-LAX-5/SamsungGalaxyS20Ultra	4514
2022-02-24-US-LAX-1/SamsungGalaxyS20Ultra	3581
2021-08-24-US-SVL-2/GooglePixel5	3314
2022-02-23-US-LAX-3/XiaomiMi8	2880
2022-02-23-US-LAX-1/GooglePixel5	2845
2021-09-28-US-MTV-1/GooglePixel5	2485
2022-02-24-US-LAX-3/XiaomiMi8	2464
2022-02-23-US-LAX-5/XiaomiMi8	2420
2022-03-22-US-MTV-1/SamsungGalaxyS20Ultra	2109
2022-04-25-US-OAK-1/GooglePixel5	1912
2022-03-31-US-LAX-3/SamsungGalaxyS20Ultra	1820
2021-09-07-US-MTV-1/SamsungGalaxyS20Ultra	1802
2021-09-20-US-MTV-1/XiaomiMi8	1795
2021-09-20-US-MTV-2/GooglePixel4	1742
2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1724
2022-01-26-US-MTV-1/XiaomiMi8	1698
2021-08-17-US-MTV-1/GooglePixel5	1673
2022-04-01-US-LAX-1/SamsungGalaxyS20Ultra	1670
2022-02-08-US-SJC-1/XiaomiMi8	1665
2022-03-14-US-MTV-1/GooglePixel5	1635
2022-04-25-US-OAK-2/GooglePixel4	1584
2021-11-30-US-MTV-1/GooglePixel5	1521
2022-04-01-US-LAX-3/XiaomiMi8	1462
2021-11-05-US-MTV-1/XiaomiMi8	1442
2022-04-22-US-OAK-1/GooglePixel5	1432
2021-06-22-US-MTV-1/XiaomiMi8	1398
2022-04-22-US-OAK-2/XiaomiMi8	1394
2022-02-15-US-SJC-1/GooglePixel5	1392
2022-03-31-US-LAX-1/GooglePixel5	1301
2021-09-14-US-MTV-1/GooglePixel5	1270
2021-08-12-US-MTV-1/GooglePixel4	1265
2022-03-17-US-SJC-1/GooglePixel5	1172
2022-01-18-US-SJC-2/GooglePixel5	942
2022-01-11-US-MTV-1/GooglePixel6Pro	942
2022-01-04-US-MTV-1/SamsungGalaxyS20Ultra	922
2022-02-01-US-SJC-1/XiaomiMi8	910

dtype: int64

```
X=data[['UnixTimeMillis','LatitudeDegrees','LongitudeDegrees']]
y=data['tripId']


data=pd.DataFrame(data)
print(data)
```

```
                                       tripId  UnixTimeMillis  \
0      2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra   1619650832999
1      2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra   1619650833999
2      2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra   1619650834999
3      2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra   1619650835999
4      2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra   1619650836999
...                                          ...             ...
66092           2022-04-25-US-OAK-2/GooglePixel4   1650927742650
66093           2022-04-25-US-OAK-2/GooglePixel4   1650927743642
66094           2022-04-25-US-OAK-2/GooglePixel4   1650927744651
```

```
66095           2022-04-25-US-OAK-2/GooglePixel4   1650927745640
66096           2022-04-25-US-OAK-2/GooglePixel4   1650927746632

       LatitudeDegrees  LongitudeDegrees
0            37.904611        -86.481078
1            37.904611        -86.481078
2            37.904611        -86.481078
3            37.904611        -86.481078
4            37.904611        -86.481078
...                ...               ...
66092        37.904611        -86.481078
66093        37.904611        -86.481078
66094        37.904611        -86.481078
66095        37.904611        -86.481078
66096        37.904611        -86.481078

[66097 rows x 4 columns]
```

```
 X=data.iloc[:,[2,3]].values
print(X)
```

```
[[ 37.90461132 -86.48107806]
 [ 37.90461132 -86.48107806]
 [ 37.90461132 -86.48107806]
 ...
 [ 37.90461132 -86.48107806]
 [ 37.90461132 -86.48107806]
 [ 37.90461132 -86.48107806]]
```

```
y=data.iloc[:,0:4].values
print(y)
```

```
[['2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra' 1619650832999
  37.9046113156345 -86.48107806249548]
 ['2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra' 1619650833999
  37.9046113156345 -86.48107806249548]
 ['2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra' 1619650834999
  37.9046113156345 -86.48107806249548]
 ...
 ['2022-04-25-US-OAK-2/GooglePixel4' 1650927744651 37.9046113156345
  -86.48107806249548]
 ['2022-04-25-US-OAK-2/GooglePixel4' 1650927745640 37.9046113156345
  -86.48107806249548]
 ['2022-04-25-US-OAK-2/GooglePixel4' 1650927746632 37.9046113156345
  -86.48107806249548]]
```

```
cname_ = glob('../input/smartphone-decimeter-2022/train/*')
tmp = []
for i in cname_:
    tmp.extend(glob(f'{i}/*'))

cname=[]

for r in tmp:
    cname.append([r.split('/')[4],r.split('/')[5]])

cname = pd.DataFrame(sorted(cname))
cname
```

	0	1
0	2020-05-15-US-MTV-1	GooglePixel4XL
1	2020-05-21-US-MTV-1	GooglePixel4
2	2020-05-21-US-MTV-1	GooglePixel4XL
3	2020-05-21-US-MTV-2	GooglePixel4
4	2020-05-21-US-MTV-2	GooglePixel4XL
...	...	...
165	2021-12-15-US-MTV-1	XiaomiMi8
166	2021-12-28-US-MTV-1	GooglePixel5
167	2021-12-28-US-MTV-1	GooglePixel6Pro
168	2021-12-28-US-MTV-1	SamsungGalaxyS20Ultra
169	2021-12-28-US-MTV-1	XiaomiMi8

170 rows × 2 columns

Next steps: Generate code with cname | View recommended plots | New interactive sheet

List of mobile phones used in train

```
cname[1].value_counts()
```

	count
1	
GooglePixel4	43
GooglePixel4XL	39
XiaomiMi8	29
GooglePixel5	28
SamsungGalaxyS20Ultra	23
GooglePixel6Pro	8

dtype: int64

List of mobile phones used in test

```
cname_ = glob('../input/smartphone-decimeter-2022/test/*')
tmp = []
for i in cname_:
    tmp.extend(glob(f'{i}/*'))

cname=[]

for r in tmp:
    cname.append([r.split('/')[4],r.split('/')[5]])

cname = pd.DataFrame(sorted(cname))
cname[1].value_counts()
```

	count
1	
GooglePixel5	13
XiaomiMi8	11
SamsungGalaxyS20Ultra	8
GooglePixel4	3
GooglePixel6Pro	1

dtype: int64

Since the train data is from 2020 and the test is from 2021-2022, it seems that the mobile phone used is a little different.

Read the metadata file

```
import json
raw = open('../input/smartphone-decimeter-2022/metadata/raw_state_bit_map.json', 'r')
json.load(raw)
```

```
{'0': 'Code Lock',
 '1': 'Bit Sync',
 '2': 'Subframe Sync',
 '3': 'Time Of Week Decoded State',
 '4': 'Millisecond Ambiguity',
 '5': 'Symbol Sync',
 '6': 'GLONASS String Sync',
 '7': 'GLONASS Time Of Day Decoded',
 '8': 'BEIDOU D2 Bit Sync',
 '9': 'BEIDOU D2 Subframe Sync',
 '10': 'Galileo E1BC Code Lock',
 '11': 'Galileo E1C 2^nd^ Code Lock',
 '12': 'Galileo E1B Page Sync',
 '13': 'SBAS Sync',
 '14': 'Time Of Week Known',
 '15': 'GLONASS Time Of Day Known'}
```

```
import json
bit = open('../input/smartphone-decimeter-2022/metadata/accumulated_delta_range_state_bit_map.json', 'r')
json.load(bit)
```

```
{'0': 'VALID',
 '1': 'RESET',
 '2': 'CYCLE_SLIP',
 '3': 'HALF_CYCLE_RESOLVED',
 '4': 'HALF_CYCLE_REPORTED'}
```

```
mapping = pd.read_csv('../input/smartphone-decimeter-2022/metadata/constellation_type_mapping.csv')
mapping
```

	constellationType	constellationName
0	0	UNKNOWN
1	1	GPS
2	2	SBAS
3	3	GLONASS
4	4	QZSS
5	5	BEIDOU
6	6	GALILEO
7	7	IRNSS

Next steps: Generate code with mapping | View recommended plots | New interactive sheet

Read the sample file(2020-05-15-US-MTV-1)

```
ground = pd.read_csv('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/ground_truth.csv')
ground
```

	MessageType	Provider	LatitudeDegrees	LongitudeDegrees	AltitudeMeters	SpeedMps	AccuracyMeters	BearingDegrees	l
0	Fix	GT	37.416619	-122.082065	NaN	0.002044	0.1	92.968750	
1	Fix	GT	37.416619	-122.082065	NaN	0.002198	0.1	92.969666	
2	Fix	GT	37.416619	-122.082065	NaN	0.001414	0.1	92.969850	
3	Fix	GT	37.416619	-122.082065	NaN	0.001414	0.1	92.969850	
4	Fix	GT	37.416619	-122.082065	NaN	0.001414	0.1	92.969910	
...	...	...	...	...	...	...	...	...	
3357	Fix	GT	37.631664	-122.424975	NaN	0.797332	0.1	53.082825	
3358	Fix	GT	37.631667	-122.424971	NaN	0.056422	0.1	53.170044	
3359	Fix	GT	37.631667	-122.424972	NaN	0.006325	0.1	53.168518	
3360	Fix	GT	37.631667	-122.424972	NaN	0.005113	0.1	53.166443	
3361	Fix	GT	37.631667	-122.424972	NaN	0.002995	0.1	53.167360	

3362 rows × 9 columns

Next steps: Generate code with ground | View recommended plots | New interactive sheet

MessageType - "Fix", the prefix of sentence.

Provider - "GT", short for ground truth.

[Latitude/Longitude]Degrees - The WGS84 latitude, longitude (in decimal degrees) estimated by the reference GNSS receiver (NovAtel SPAN). When extracting from the NMEA file, linear interpolation has been applied to align the location to the expected non-integer timestamps.

AltitudeMeters - The height above the WGS84 ellipsoid (in meters) estimated by the reference GNSS receiver.

SpeedMps* - The speed over ground in meters per second.

AccuracyMeters - The estimated horizontal accuracy radius in meters of this location at the 68th percentile confidence level. This means that there is a 68% chance that the true location of the device is within a distance of this uncertainty of the reported location.

BearingDegrees - Bearing is measured in degrees clockwise from north. It ranges from 0 to 359.999 degrees.

UnixTimeMillis - An integer number of milliseconds since the GPS epoch (1970/1/1 midnight UTC). Converted from GnssClock.

```
imu= pd.read_csv('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/device_imu.csv')
imu
```

	MessageType	utcTimeMillis	MeasurementX	MeasurementY	MeasurementZ	BiasX	BiasY	BiasZ
0	UncalAccel	1589573679447	0.237913	9.250001	1.784782	0.00000	0.000000	0.00000
1	UncalGyro	1589573679450	-0.022198	0.004974	0.014298	0.00000	0.000000	0.00000
2	UncalMag	1589573679450	2.812906	-37.320263	-1.189655	74.85504	-10.046193	-44.48028
3	UncalAccel	1589573679457	0.134411	9.673122	1.313673	0.00000	0.000000	0.00000
4	UncalMag	1589573679460	2.601327	-37.849490	-0.733490	74.85504	-10.046193	-44.48028
...	...	...	...	...	...	...	...	...
734851	UncalAccel	1589577041402	0.393715	9.786283	0.759665	0.00000	0.000000	0.00000
734852	UncalMag	1589577041407	6.118653	-41.951572	-16.720966	74.85504	-10.046193	-44.48028
734853	UncalAccel	1589577041412	0.113705	9.996571	1.240902	0.00000	0.000000	0.00000
734854	UncalMag	1589577041417	6.266749	-41.931866	-16.268131	74.85504	-10.046193	-44.48028
734855	UncalGyro	1589577041422	0.005176	-0.001123	-0.036102	0.00000	0.000000	0.00000

734856 rows × 8 columns

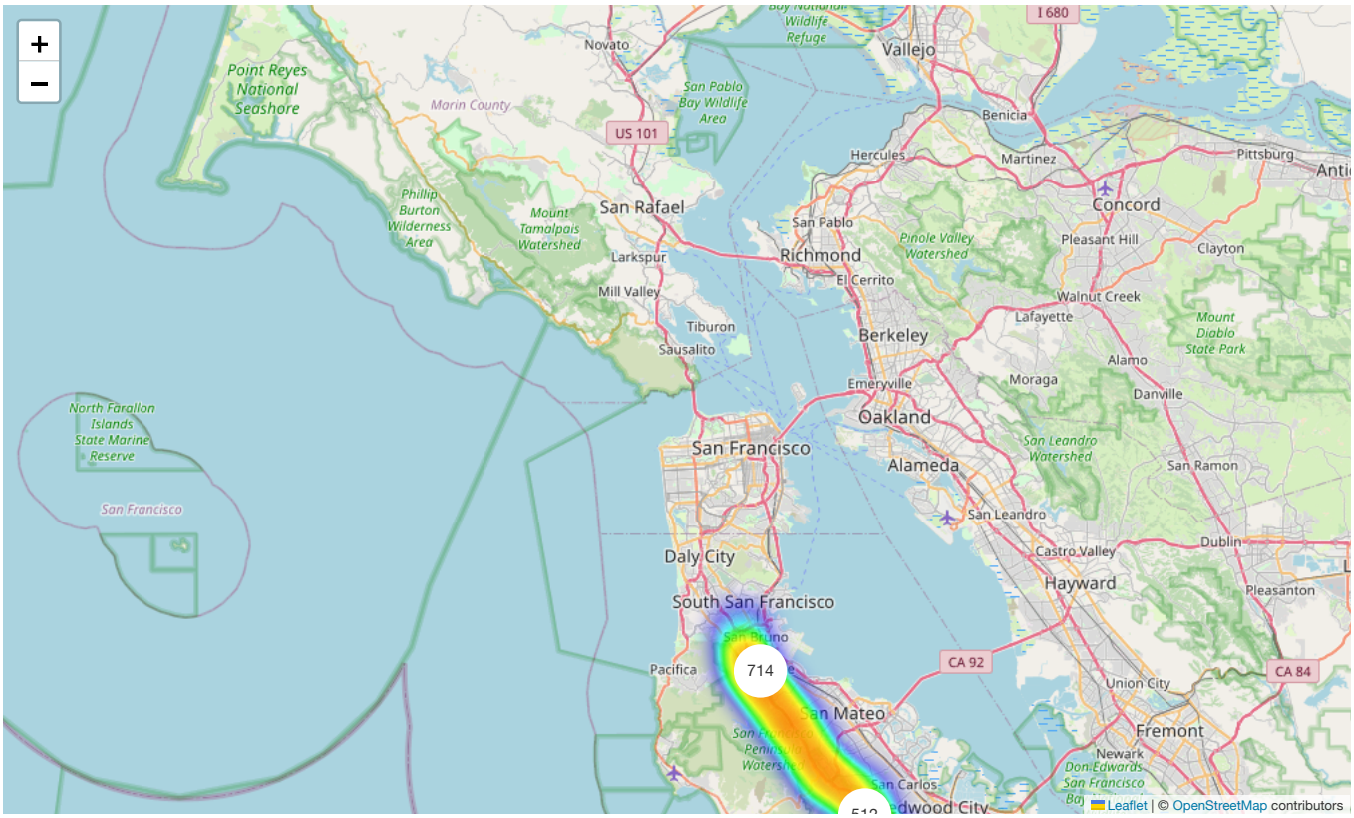
```
gnss = pd.read_csv('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/device_gnss.csv')
gnss
```

	MessageType	utcTimeMillis	TimeNanos	LeapSecond	FullBiasNanos	BiasNanos	BiasUncertaintyNanos	DriftNa
0	Raw	1589573679445	9825908000000	NaN	-1273599071537525599	0.414394	29.760322	
1	Raw	1589573679445	9825908000000	NaN	-1273599071537525599	0.414394	29.760322	
2	Raw	1589573679445	9825908000000	NaN	-1273599071537525599	0.414394	29.760322	
3	Raw	1589573679445	9825908000000	NaN	-1273599071537525599	0.414394	29.760322	
4	Raw	1589573679445	9825908000000	NaN	-1273599071537525599	0.414394	29.760322	
...	...	...	...	...	...	...	...	
90148	Raw	1589577040445	13186908000000	NaN	-1273599071537517829	0.863317	18.450928	
90149	Raw	1589577040445	13186908000000	NaN	-1273599071537517829	0.863317	18.450928	
90150	Raw	1589577040445	13186908000000	NaN	-1273599071537517829	0.863317	18.450928	
90151	Raw	1589577040445	13186908000000	NaN	-1273599071537517829	0.863317	18.450928	
90152	Raw	1589577040445	13186908000000	NaN	-1273599071537517829	0.863317	18.450928	

90153 rows × 47 columns

mapping the Ground True data on the map.

```
from folium import plugins
df_locs = list(ground[['LatitudeDegrees','LongitudeDegrees']].values)
fol_map = folium.Map([ground['LatitudeDegrees'].median(), ground['LongitudeDegrees'].median()],zoom_start=11)
heat_map = plugins.HeatMap(df_locs)
fol_map.add_child(heat_map)
markers = plugins.MarkerCluster(locations = df_locs)
fol_map.add_child(markers)
```



Read the supplemental

gnss_log.txt

```
f = open('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/gnss_log.txt', 'r')
log = f.read()
f.close()
log[:500]
```

```
'# \n# Header Description:\n# \n# Version: v3.0.0.16 Platform: null Manufacturer: Google Model: Pixel 4 XL\n# \n# Raw,u
tcTimeMillis,TimeNanos,LeapSecond,TimeUncertaintyNanos,FullBiasNanos,BiasNanos,BiasUncertaintyNanos,DriftNanosPerSecon
d,DriftUncertaintyNanosPerSecond,HardwareClockDiscontinuityCount,Svid,TimeOffsetNanos,State,ReceivedSvTimeNanos,Receive
dSvTimeUncertaintyNanos,Cn0DbHz,PseudorangeRateMetersPerSecond,PseudorangeRateUncertaintyMetersPerSecond,AccumulatedDel
```

Txt to Pandas

```
path ='../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/gnss_log.txt'
gnss_section_names = {'Raw','UncalAccel', 'UncalGyro', 'UncalMag', 'Fix', 'Status', 'OrientationDeg'}
with open(path) as f_open:
    datalines = f_open.readlines()

datas = {k: [] for k in gnss_section_names}
gnss_map = {k: [] for k in gnss_section_names}
for dataline in datalines:
  if dataline !='' and dataline[0] !='':
    is_header = dataline.startswith('#')
    dataline = dataline.strip('#').strip().split(',')
    # skip over notes, version numbers, etc
    if is_header and dataline[0] in gnss_section_names:
        gnss_map[dataline[0]] = dataline[1:]
    elif not is_header:
        if dataline !='' and dataline[0] !='':
            datas[dataline[0]].append(dataline[1:])

results = dict()
for k, v in datas.items():
    results[k] = pd.DataFrame(v, columns=gnss_map[k])
for k, df in results.items():
    for col in df.columns:
        if col == 'CodeType':
            continue
        results[k][col] = pd.to_numeric(results[k][col])
```

```
results['Raw']
```

	utcTimeMillis	TimeNanos	LeapSecond	TimeUncertaintyNanos	FullBiasNanos	BiasNanos	BiasUncertaintyNanos
0	1589573679445	9825908000000	NaN	NaN	-1273599071537525599	0.414394	29.760322
1	1589573679445	9825908000000	NaN	NaN	-1273599071537525599	0.414394	29.760322
2	1589573679445	9825908000000	NaN	NaN	-1273599071537525599	0.414394	29.760322
3	1589573679445	9825908000000	NaN	NaN	-1273599071537525599	0.414394	29.760322
4	1589573679445	9825908000000	NaN	NaN	-1273599071537525599	0.414394	29.760322
...	...	...	...	...	...	...	...
90148	1589577040445	13186908000000	NaN	NaN	-1273599071537517829	0.863317	18.450928
90149	1589577040445	13186908000000	NaN	NaN	-1273599071537517829	0.863317	18.450928
90150	1589577040445	13186908000000	NaN	NaN	-1273599071537517829	0.863317	18.450928
90151	1589577040445	13186908000000	NaN	NaN	-1273599071537517829	0.863317	18.450928
90152	1589577040445	13186908000000	NaN	NaN	-1273599071537517829	0.863317	18.450928

90153 rows × 36 columns

```
results['UncalAccel']
```

	utcTimeMillis	elapsedRealtimeNanos	UncalAccelXMps2	UncalAccelYMps2	UncalAccelZMps2	BiasXMps2	BiasYMps2	Bias
0	1589573679447	NaN	0.237913	9.250001	1.784782	0.0	0.0	
1	1589573679457	NaN	0.134411	9.673122	1.313673	0.0	0.0	
2	1589573679467	NaN	-0.085421	10.311209	0.171189	0.0	0.0	
3	1589573679477	NaN	-0.091420	9.795237	0.594629	0.0	0.0	
4	1589573679486	NaN	-0.106993	9.221649	1.844052	0.0	0.0	
...	...	...	...	...	...	...	...	...
226347	1589577041373	NaN	0.389894	9.521341	1.182911	0.0	0.0	
226348	1589577041383	NaN	0.550553	9.661240	1.435501	0.0	0.0	
226349	1589577041393	NaN	0.528499	9.866081	0.680037	0.0	0.0	
226350	1589577041402	NaN	0.393715	9.786283	0.759665	0.0	0.0	
226351	1589577041412	NaN	0.113705	9.996571	1.240902	0.0	0.0	

226352 rows × 8 columns

```
results['UncalGyro']
```

	utcTimeMillis	elapsedRealtimeNanos	UncalGyroXRadPerSec	UncalGyroYRadPerSec	UncalGyroZRadPerSec	DriftXRadPerSe
0	1589573679450	NaN	-0.022198	0.004974	0.014298	0
1	1589573679469	NaN	0.024007	-0.005940	-0.015149	0
2	1589573679489	NaN	-0.026377	0.020333	-0.026209	0
3	1589573679508	NaN	0.026483	-0.013112	0.001743	0
4	1589573679527	NaN	-0.027226	0.005423	0.021891	0
...	...	...	...	...	...	
173582	1589577041344	NaN	-0.004325	0.005394	-0.031962	0
173583	1589577041364	NaN	-0.034431	-0.017382	0.093108	0
173584	1589577041383	NaN	-0.003963	-0.010917	0.039844	0
173585	1589577041402	NaN	0.002304	-0.000161	-0.015200	0
173586	1589577041422	NaN	0.005176	-0.001123	-0.036102	0

173587 rows × 8 columns

```
results[ 'UncalMag']
```

	utcTimeMillis	elapsedRealtimeNanos	UncalMagXMicroT	UncalMagYMicroT	UncalMagZMicroT	BiasXMicroT	BiasYMicroT	I
0	1589573679450	NaN	2.812906	-37.320263	-1.189655	74.85504	-10.046193	
1	1589573679460	NaN	2.601327	-37.849490	-0.733490	74.85504	-10.046193	
2	1589573679470	NaN	2.339869	-37.497173	-1.048279	74.85504	-10.046193	
3	1589573679480	NaN	2.481942	-37.612180	-1.341384	74.85504	-10.046193	
4	1589573679490	NaN	3.130273	-37.157420	-1.035259	74.85504	-10.046193	
...	...	...	...	...	...	...	...	...
334912	1589577041377	NaN	6.067253	-42.330900	-15.664912	74.85504	-10.046193	
334913	1589577041387	NaN	6.083881	-42.204082	-15.967022	74.85504	-10.046193	
334914	1589577041397	NaN	5.801288	-42.114414	-16.875364	74.85504	-10.046193	
334915	1589577041407	NaN	6.118653	-41.951572	-16.720966	74.85504	-10.046193	
334916	1589577041417	NaN	6.266749	-41.931866	-16.268131	74.85504	-10.046193	

334917 rows × 8 columns

```
results['Fix']
```

	Provider	LatitudeDegrees	LongitudeDegrees	AltitudeMeters	SpeedMps	AccuracyMeters	BearingDegrees	UnixTimeMillis	S

```
results['Status']
```

	UnixTimeMillis	SignalCount	SignalIndex	ConstellationType	Svid	CarrierFrequencyHz	Cn0DbHz	AzimuthDegrees	Elevatio

```
results['OrientationDeg']
```

	utcTimeMillis	elapsedRealtimeNanos	yawDeg	rollDeg	pitchDeg

gnss_rinex.20o

```
f = open('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/gnss_rinex.20o', 'r')
rinex = f.read()
f.close()
rinex[:500]
```

```
'     3.03           OBSERVATION DATA    M                   RINEX VERSION / TYPE\nGNSSToRINEXV3       Google
20200515 201457 UTC PGM / RUN BY / DATE\nGoogle GnssLogger                                                MARKER NAME\nUnkno
wn                                                  MARKER NUMBER\nUnknown             Unknown
OBSERVER / AGENCY\nUnknown             GNSSToRINEXV3       v1.0.0.0            REC # / TYPE / VERS\nUnknown
```

```
rinex =pd.read_csv('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/gnss_rinex.20o')
rinex
```

	3.03 OBSERVATION DATA M RINEX VERSION / TYPE
0	GNSSToRINEXV3 Google 202005...
1	Google GnssLogger ...
2	Unknown ...
3	Unknown Unknown ...
4	Unknown GNSSToRINEXV3 v1.0.0...
...	...
68040	E02 23507128.61206 55703.29306 -922....
68041	E15 26871450.84104 -23420.39504 2742....
68042	E25 -0.00023 -2627....
68043	E30 23006428.44005 -336548.29305 1622....
68044	E36 24028866.72105 332714.16205 -1636....

68045 rows × 1 columns

Next steps: Generate code with rinex | View recommended plots | New interactive sheet

span_log.nmea

```
f = open('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/span_log.nmea', 'r')
span = f.read()
f.close()
span[:500]
```

```
'$GPGGA,201439.45,3724.9971160,N,12204.9239000,W,1,0,0.0,0.0000,M,0.0000,M,0.0000,0\n$GPRMC,201439.45,A,3724.9971160,N,
12204.9239000,W,0.0020,92.9688,150520,,,,\n$GPGGA,201440.45,3724.9971160,N,12204.9239000,W,1,0,0.0,0.0000,M,0.0000,M,0.
0000,0\n$GPRMC,201440.45,A,3724.9971160,N,12204.9239000,W,0.0022,92.9697,150520,,,,\n$GPGGA,201441.45,3724.9971160,N,12
204.9239000,W,1,0,0.0,0.0000,M,0.0000,M,0.0000,0\n$GPRMC,201441.45,A,3724.9971160,N,12204.9239000,W,0.0014,92.9698,1505
```

```
span = pd.read_csv('../input/smartphone-decimeter-2022/train/2020-05-15-US-MTV-1/GooglePixel4XL/supplemental/span_log.nmea')
span
```

	$GPGGA	201439.45	3724.9971160	N	12204.9239000	W	1	0	0.0	0.0000	M	0.0000.1
0	$GPRMC	201439.45	A	3724.9971160	N	12204.9239000	W	0.0020	92.9688	150520.0	NaN	NaN
1	$GPGGA	201440.45	3724.9971160	N	12204.9239000	W	1	0.0000	0.0000	0.0	M	0.0
2	$GPRMC	201440.45	A	3724.9971160	N	12204.9239000	W	0.0022	92.9697	150520.0	NaN	NaN
3	$GPGGA	201441.45	3724.9971160	N	12204.9239000	W	1	0.0000	0.0000	0.0	M	0.0
4	$GPRMC	201441.45	A	3724.9971160	N	12204.9239000	W	0.0014	92.9698	150520.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
6718	$GPRMC	211038.45	A	3737.8999900	N	12225.4982960	W	0.0063	53.1685	150520.0	NaN	NaN
6719	$GPGGA	211039.45	3737.8999900	N	12225.4983020	W	1	0.0000	0.0000	0.0	M	0.0
6720	$GPRMC	211039.45	A	3737.8999900	N	12225.4983020	W	0.0051	53.1664	150520.0	NaN	NaN
6721	$GPGGA	211040.45	3737.8999900	N	12225.4983020	W	1	0.0000	0.0000	0.0	M	0.0
6722	$GPRMC	211040.45	A	3737.8999900	N	12225.4983020	W	0.0030	53.1674	150520.0	NaN	NaN

6723 rows × 15 columns

Next steps: Generate code with span | View recommended plots | New interactive sheet

presentation

```
sub = pd.read_csv('../input/smartphone-decimeter-2022/sample_submission.csv')
sub
```

	tripId	UnixTimeMillis	LatitudeDegrees	LongitudeDegrees
0	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650832999	37.904611	-86.481078
1	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650833999	37.904611	-86.481078
2	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650834999	37.904611	-86.481078
3	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650835999	37.904611	-86.481078
4	2021-04-28-US-MTV-2/SamsungGalaxyS20Ultra	1619650836999	37.904611	-86.481078
...	...	...	...	...
66092	2022-04-25-US-OAK-2/GooglePixel4	1650927742650	37.904611	-86.481078
66093	2022-04-25-US-OAK-2/GooglePixel4	1650927743642	37.904611	-86.481078
66094	2022-04-25-US-OAK-2/GooglePixel4	1650927744651	37.904611	-86.481078
66095	2022-04-25-US-OAK-2/GooglePixel4	1650927745640	37.904611	-86.481078
66096	2022-04-25-US-OAK-2/GooglePixel4	1650927746632	37.904611	-86.481078

66097 rows × 4 columns

Next steps: Generate code with sub | View recommended plots | New interactive sheet

```
pd.read_csv('../input/smartphone-decimeter-2022/train/2020-08-06-US-MTV-2/GooglePixel4/ground_truth.csv')
```

	MessageType	Provider	LatitudeDegrees	LongitudeDegrees	AltitudeMeters	SpeedMps	AccuracyMeters	BearingDegrees	l
0	Fix	GT	37.525222	-122.353397	NaN	0.002236	0.1	228.166660	
1	Fix	GT	37.525222	-122.353397	NaN	0.001869	0.1	228.166610	
2	Fix	GT	37.525222	-122.353397	NaN	0.002195	0.1	228.166400	
3	Fix	GT	37.525222	-122.353397	NaN	0.003162	0.1	228.166530	
4	Fix	GT	37.525222	-122.353397	NaN	0.003162	0.1	228.166760	
...	...	...	...	...	...	...	...	...	
1697	Fix	GT	37.427243	-122.069876	NaN	4.882882	0.1	87.660034	
1698	Fix	GT	37.427279	-122.069876	NaN	3.104689	0.1	87.495026	
1699	Fix	GT	37.427299	-122.069876	NaN	1.314388	0.1	86.996640	
1700	Fix	GT	37.427306	-122.069876	NaN	0.159897	0.1	86.879730	
1701	Fix	GT	37.427306	-122.069876	NaN	0.016617	0.1	86.871580	

1702 rows × 9 columns

```
import requests
import shapely.wkt
import geopandas as gpd
import pandas as pd

# Download geojson file of US San Francisco Bay Area
r = requests.get("https://data.sfgov.org/api/views/wamw-vt4s/rows.json?accessType=DOWNLOAD")
r.raise_for_status()

# Get geojson from response
data = r.json()

# Get polygons that represent San Francisco Bay Area
shapes = []
for d in data["data"]:
    shapes.append(shapely.wkt.loads(d[8]))

# Convert list of polygons to a GeoDataFrame
gdf_bayarea = gpd.GeoDataFrame()

# I'll use only 6th and 7th objects
for shp in shapes[5:7]:
    tmp = gpd.GeoDataFrame(geometry=[shp])
    gdf_bayarea = pd.concat([gdf_bayarea, tmp])

# Final GeoDataFrame
gdf_bayarea = gpd.GeoDataFrame(gdf_bayarea, geometry="geometry")

# Display the GeoDataFrame
gdf_bayarea
```

	geometry
0	MULTIPOLYGON (((-122.50248 37.70813, -122.4977...
0	MULTIPOLYGON (((-122.02427 37.46094, -122.0203...

Next steps: Generate code with gdf_bayarea | View recommended plots | New interactive sheet

gdf_bayarea

	geometry
0	MULTIPOLYGON (((-122.50248 37.70813, -122.4977...
0	MULTIPOLYGON (((-122.02427 37.46094, -122.0203...

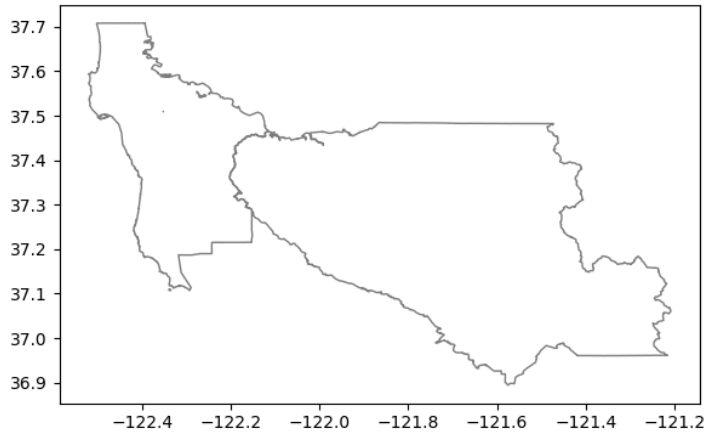
Next steps: Generate code with gdf_bayarea | View recommended plots | New interactive sheet

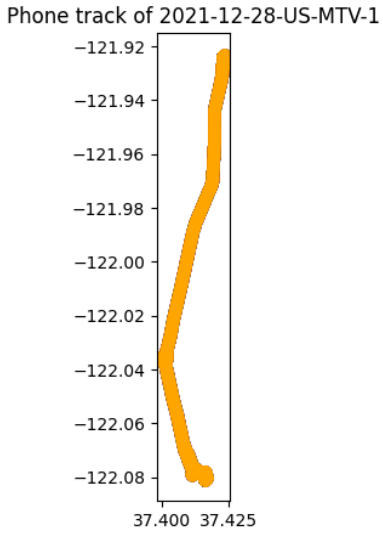
```
%%capture
collectionNames = [item.split("/")[-1] for item in glob("../input/smartphone-decimeter-2022/train/*")]

gdfs = []
for collectionName in collectionNames:
    gdfs_each_collectionName = []
    csv_paths = glob(f"../input/smartphone-decimeter-2022/train/{collectionName}/*/ground_truth.csv")
    for csv_path in csv_paths:
        df_gt = pd.read_csv(csv_path)
        df_gt["geometry"] = [Point(lngDeg, latDeg) for lngDeg, latDeg in zip(df_gt["LatitudeDegrees"], df_gt["LongitudeDegre
        gdfs_each_collectionName.append(GeoDataFrame(df_gt))
    gdfs.append(gdfs_each_collectionName)

colors = ['blue', 'green', 'purple', 'orange']


gdfs_each_collectionName
```

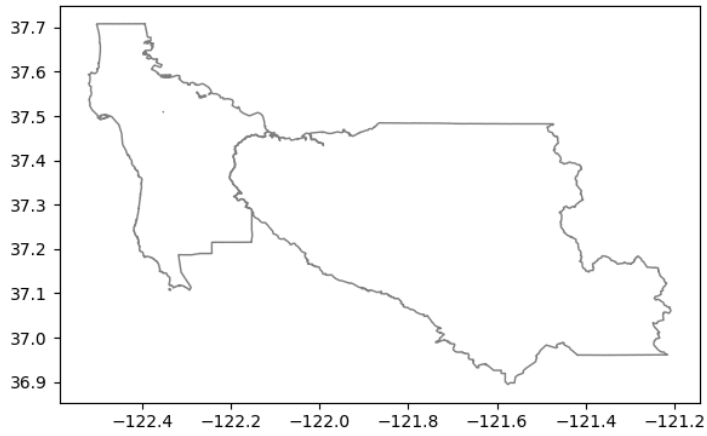
```
[     MessageType Provider  LatitudeDegrees  LongitudeDegrees  AltitudeMeters  \
0            Fix       GT        37.626536       -122.425670             NaN
1            Fix       GT        37.626536       -122.425670             NaN
2            Fix       GT        37.626536       -122.425670             NaN
3            Fix       GT        37.626536       -122.425670             NaN
4            Fix       GT        37.626536       -122.425670             NaN
...          ...      ...              ...               ...             ...
2430         Fix       GT        37.416686       -122.082116             NaN
2431         Fix       GT        37.416686       -122.082116             NaN
2432         Fix       GT        37.416686       -122.082116             NaN
2433         Fix       GT        37.416686       -122.082116             NaN
2434         Fix       GT        37.416686       -122.082116             NaN

      SpeedMps  AccuracyMeters  BearingDegrees  UnixTimeMillis  \
0     0.000610             0.1       83.618286   1599242401431
1     0.021130             0.1       83.618256   1599242402431
2     0.047988             0.1       83.619870   1599242403431
3     0.029310             0.1       83.621826   1599242404431
4     0.003092             0.1       83.622375   1599242405431
...        ...             ...             ...             ...
2430  0.003900             0.1      274.705080   1599244831431
2431  0.003980             0.1      274.704530   1599244832431
2432  0.003152             0.1      274.705400   1599244833431
2433  0.001769             0.1      274.705780   1599244834431
2434  0.001881             0.1      274.705540   1599244835431

                     geometry
0     POINT (37.627 -122.426)
1     POINT (37.627 -122.426)
2     POINT (37.627 -122.426)
3     POINT (37.627 -122.426)
4     POINT (37.627 -122.426)
...                       ...
2430  POINT (37.417 -122.082)
2431  POINT (37.417 -122.082)
2432  POINT (37.417 -122.082)
2433  POINT (37.417 -122.082)
2434  POINT (37.417 -122.082)

[2435 rows x 10 columns],
     MessageType Provider  LatitudeDegrees  LongitudeDegrees  AltitudeMeters  \
0            Fix       GT        37.626537       -122.425670             NaN
1            Fix       GT        37.626537       -122.425670             NaN
2            Fix       GT        37.626537       -122.425670             NaN
3            Fix       GT        37.626537       -122.425670             NaN
4            Fix       GT        37.626537       -122.425670             NaN
...          ...      ...              ...               ...             ...
```

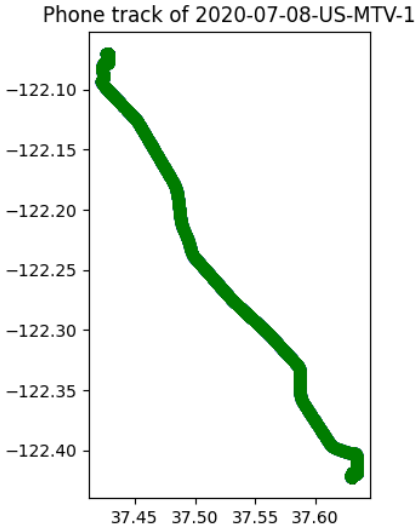
```
2431         Fix       GT        37.416685       -122.082116             NaN
2432         Fix       GT        37.416685       -122.082116             NaN
2433         Fix       GT        37.416685       -122.082116             NaN
2434         Fix       GT        37.416685       -122.082116             NaN
2435         Fix       GT        37.416685       -122.082116             NaN

      SpeedMps  AccuracyMeters  BearingDegrees  UnixTimeMillis  \
0     0.001414             0.1       83.618320   1599242399429
1     0.000806             0.1       83.618256   1599242400429
2     0.000608             0.1       83.618286   1599242401429
3     0.021046             0.1       83.618256   1599242402429
```

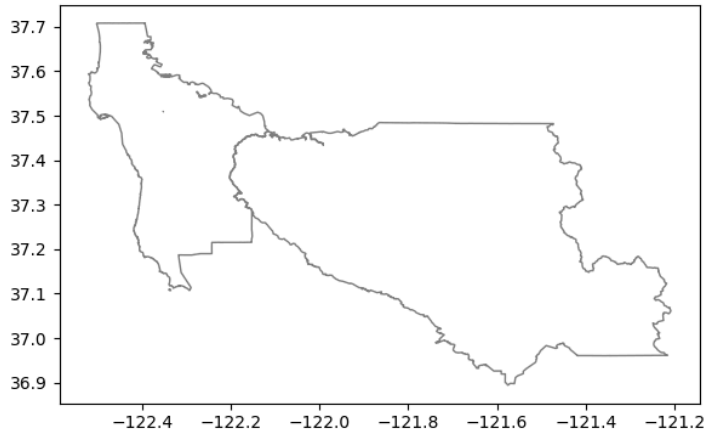
```
for collectionName, gdfs_each_collectionName in zip(collectionNames, gdfs):
    fig, axs = plt.subplots(1, 2, figsize=(15, 5))
    gdf_bayarea.plot(figsize=(10,10), color='none', edgecolor='gray', zorder=5, ax=axs[0])
    for i, gdf in enumerate(gdfs_each_collectionName):
        g2 = gdf.plot(color=colors[i], ax=axs[1])
        g2.set_title(f"Phone track of {collectionName}")
```

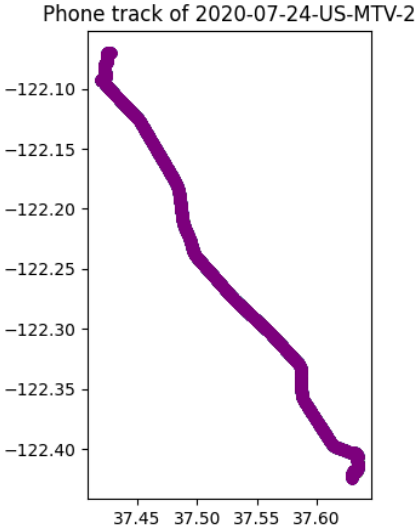

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

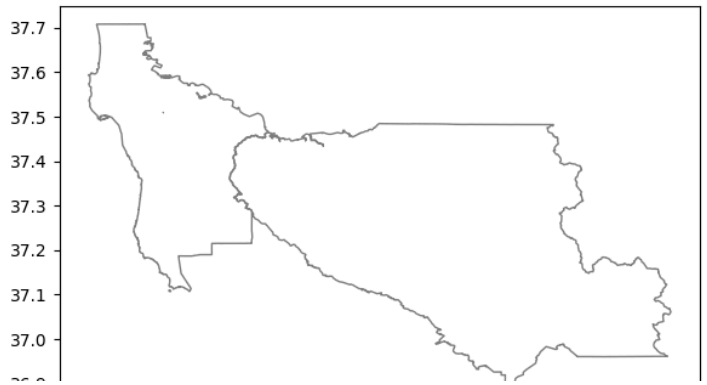

Phone track of 2021-12-28-US-MTV-1
−121.92
−121.94
−121.96
−121.98
−122.00
−122.02
−122.04
−122.06
−122.08
37.400
37.425

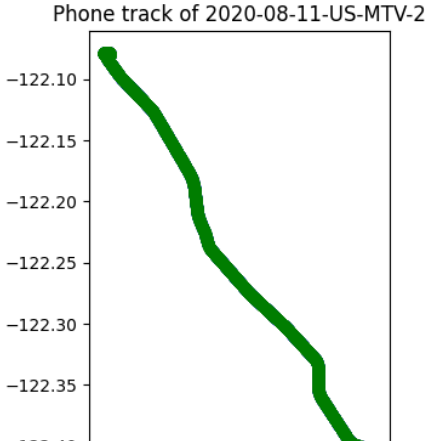

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

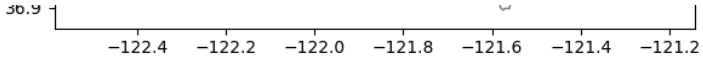

Phone track of 2020-07-08-US-MTV-1
−122.10
−122.15
−122.20
−122.25
−122.30
−122.35
−122.40
37.45
37.50
37.55
37.60

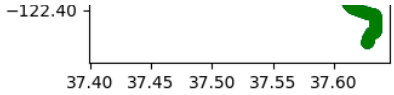

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

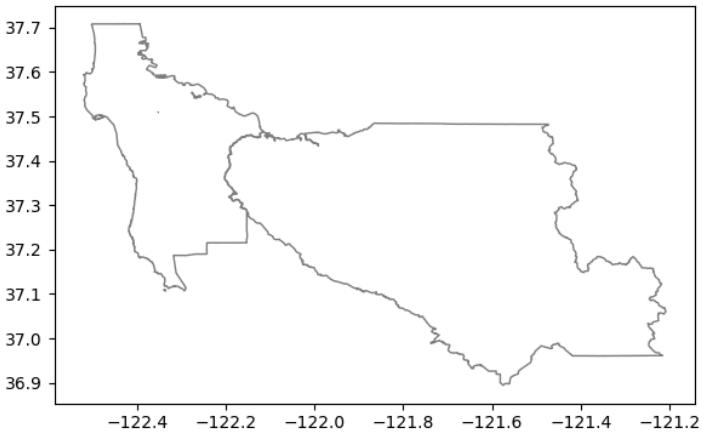

Phone track of 2020-07-24-US-MTV-2
−122.10
−122.15
−122.20
−122.25
−122.30
−122.35
−122.40
37.45
37.50
37.55
37.60

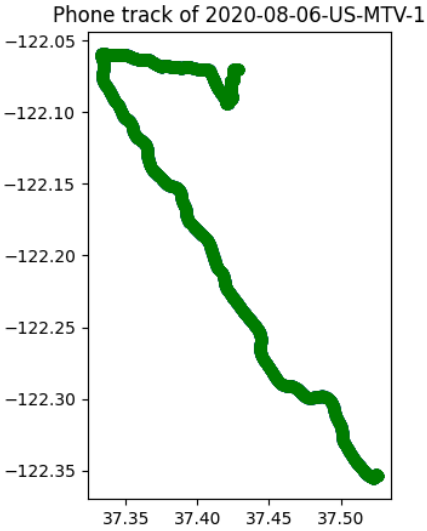

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0

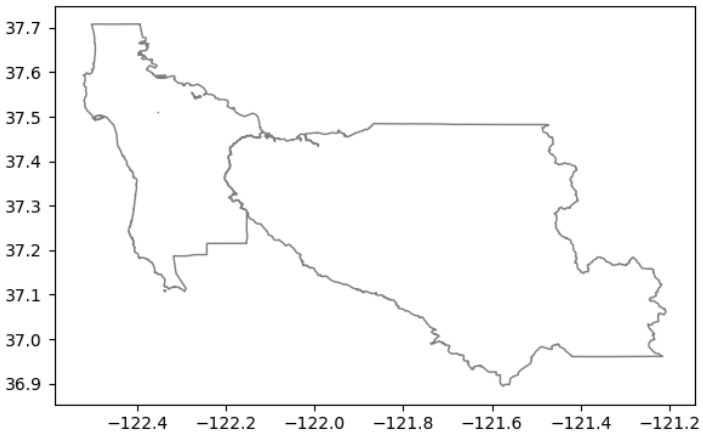

Phone track of 2020-08-11-US-MTV-2
−122.10
−122.15
−122.20
−122.25
−122.30
−122.35

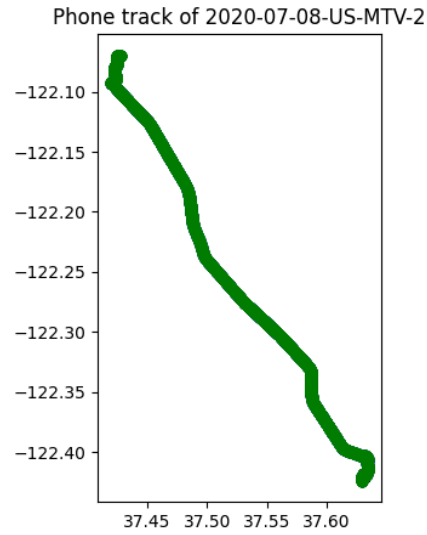

36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

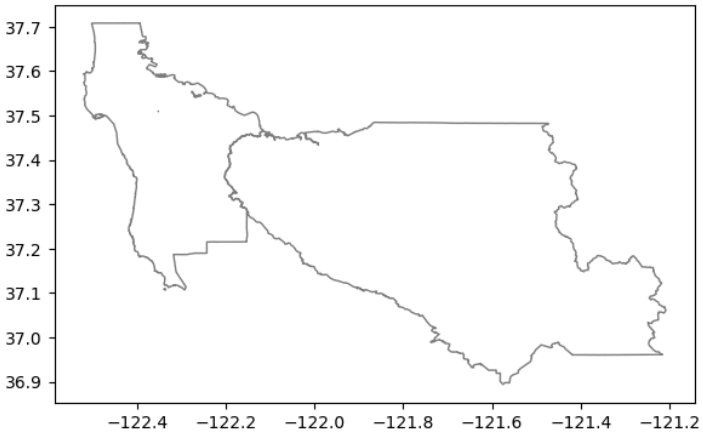

−122.40
37.40
37.45
37.50
37.55
37.60

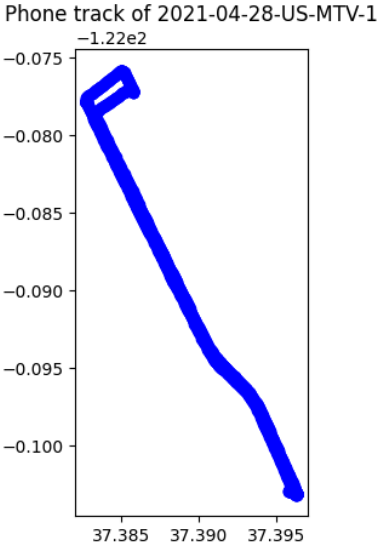

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

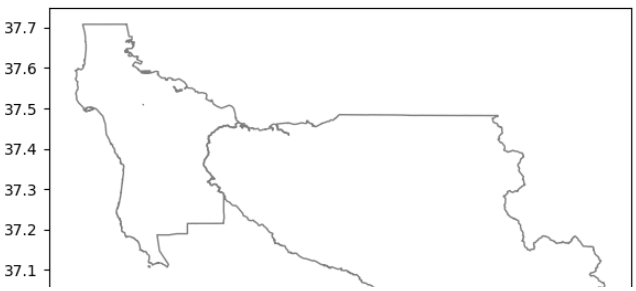

Phone track of 2020-08-06-US-MTV-1
−122.05
−122.10
−122.15
−122.20
−122.25
−122.30
−122.35
37.35
37.40
37.45
37.50

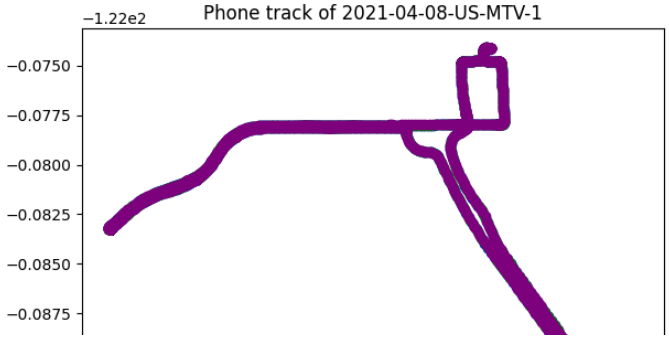

37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2


Phone track of 2020-07-08-US-MTV-2
−122.10
−122.15
−122.20
−122.25
−122.30
−122.35
−122.40
37.45
37.50
37.55
37.60


37.7
37.6
37.5
37.4
37.3
37.2
37.1
37.0
36.9
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2


Phone track of 2021-04-28-US-MTV-1
−1.22e2
−0.075
−0.080
−0.085
−0.090
−0.095
−0.100
37.385
37.390
37.395


37.7
37.6
37.5
37.4
37.3
37.2
37.1


Phone track of 2021-04-08-US-MTV-1
−1.22e2
−0.0750
−0.0775
−0.0800
−0.0825
−0.0850
−0.0875