hw_2

November 16, 2022

1 Homework 2

[2]:
```python
import ipytest

ipytest.autoconfig()
```

1.1 Coding: Using keywords to categorize 311 requests

Problem Statement: When you read through the `descriptor` and `resolution_description` columns in the 311 data, you will see that complaints related to graffiti are actually scattered throughout multiple `complaint_type` categories. We want to identify all complaints related to graffiti and see which community districts have the most instances of graffiti.

To help make this assignment easier, there's a smaller subset of the 311 data for you to use:

https://storage.googleapis.com/python-public-policy/data/cleaned_311_data_hw2.csv.zip

This smaller dataset only contains ~65,000 records from relevant complaint type categories, and has columns renamed to be lowercase and underscored.

1.1.1 Hints

- You can adapt the `recode_borocd_counts()` example from Lecture 2 for this problem.
- You may run into issues with empty values; how to deal with them.
- Ways to do case-insensitive string comparison in pure Python, which translates over to pandas.

1.1.2 Step 0

Load the data.

[3]:
```python
import pandas as pd
```

[4]:
```python
# your code here
df = pd.read_csv(
    "https://storage.googleapis.com/python-public-policy/data/
↪cleaned_311_data_hw2.csv.zip"
)
```

1.1.3 Step 1

Create a `flag_graffiti` function that checks each row in the 311 dataframe to see if the word "graffiti" is present in the `complaint_type`, `descriptor`, and/or `resolution_description`. Any of the columns may contain the word, so you should check all of them. If the word "graffiti" is found, the function should return the boolean value `True`. If "graffiti" is not found, the function should return the boolean value `False`.

Hints

- Make sure to look for "graffiti" *in* those strings. The strings may contain more than just that word.
- Capitalization may be inconsistent.

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64577 entries, 0 to 64576
Data columns (total 43 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Unnamed: 0.1                    64577 non-null  int64
 1   Unnamed: 0                      64577 non-null  int64
 2   unique_key                      64577 non-null  int64
 3   created_date                    64577 non-null  object
 4   closed_date                     49914 non-null  object
 5   agency                          64577 non-null  object
 6   agency_name                     64577 non-null  object
 7   complaint_type                  64577 non-null  object
 8   descriptor                      64577 non-null  object
 9   location_type                   59389 non-null  object
 10  incident_zip                    57322 non-null  float64
 11  incident_address                41973 non-null  object
 12  street_name                     41925 non-null  object
 13  cross_street_1                  17406 non-null  object
 14  cross_street_2                  17409 non-null  object
 15  intersection_street_1           13966 non-null  object
 16  intersection_street_2           13946 non-null  object
 17  address_type                    40799 non-null  object
 18  city                            55627 non-null  object
 19  landmark                        4338 non-null   object
 20  facility_type                   1787 non-null   object
 21  status                          64577 non-null  object
 22  due_date                        49816 non-null  object
 23  resolution_description          59613 non-null  object
 24  resolution_action_updated_date  59542 non-null  object
 25  community_board                 64577 non-null  object
 26  bbl                             35586 non-null  float64
 27  borough                         64577 non-null  object
 28  x_coordinate_(state_plane)      49671 non-null  float64
 29  y_coordinate_(state_plane)      49671 non-null  float64
 30  open_data_channel_type          64577 non-null  object
 31  park_facility_name              64577 non-null  object
 32  park_borough                    64577 non-null  object
 33  vehicle_type                    0 non-null      float64
 34  taxi_company_borough            0 non-null      float64
 35  taxi_pick_up_location           0 non-null      float64
 36  bridge_highway_name             3482 non-null   object
 37  bridge_highway_direction        3864 non-null   object
 38  road_ramp                       3864 non-null   object
 39  bridge_highway_segment          3864 non-null   object
 40  latitude                        49671 non-null  float64
 41  longitude                       49671 non-null  float64
 42  location                        49671 non-null  object
dtypes: float64(9), int64(3), object(31)
memory usage: 21.2+ MB
```

[4]:
```
df.head(20)
```

[4]:
```
    Unnamed: 0.1  Unnamed: 0  unique_key            created_date  \
0            238         238    39887282  08/01/2018 01:42:01 AM
1            353         353    39889407  08/01/2018 03:20:07 AM
2            403         403    39893067  08/01/2018 04:19:15 AM
3            655         655    39890578  08/01/2018 07:32:33 AM
4            658         658    39896325  08/01/2018 07:33:12 AM
5            661         661    39895587  08/01/2018 07:34:44 AM
6            713         713    39890857  08/01/2018 07:53:38 AM
7            789         789    39894929  08/01/2018 08:07:49 AM
8            860         860    39897352  08/01/2018 08:21:08 AM
9            876         876    39891938  08/01/2018 08:24:41 AM
10           964         964    39897349  08/01/2018 08:40:34 AM
11          1001        1001    39894633  08/01/2018 08:46:27 AM
12          1007        1007    39893314  08/01/2018 08:47:45 AM
13          1013        1013    39897422  08/01/2018 08:49:15 AM
14          1050        1050    39888412  08/01/2018 08:56:53 AM
15          1113        1113    39890371  08/01/2018 09:05:14 AM
16          1125        1125    39890363  08/01/2018 09:06:03 AM
17          1196        1196    39898317  08/01/2018 09:15:07 AM
18          1247        1247    39890214  08/01/2018 09:23:58 AM
19          1253        1253    39891954  08/01/2018 09:24:17 AM

              closed_date agency                         agency_name  \
0  08/14/2018 02:42:12 PM    DPR  Department of Parks and Recreation
1  08/23/2018 11:15:00 AM    DPR  Department of Parks and Recreation
2  08/02/2018 08:19:55 AM    DPR  Department of Parks and Recreation
3  08/16/2018 10:28:38 AM    DPR  Department of Parks and Recreation
```

```
4   09/04/2018 05:38:02 PM    DPR  Department of Parks and Recreation
5   08/10/2018 01:45:07 PM    DPR  Department of Parks and Recreation
6   08/08/2018 04:21:55 PM    DPR  Department of Parks and Recreation
7   08/07/2018 12:20:24 PM    DOT        Department of Transportation
8                      NaN   DSNY            Department of Sanitation
9   08/30/2018 12:00:00 AM   DSNY            Department of Sanitation
10  08/23/2018 12:00:00 AM   DSNY            Department of Sanitation
11  08/23/2018 09:42:02 AM    DPR  Department of Parks and Recreation
12  08/20/2018 12:00:00 AM   DSNY            Department of Sanitation
13  08/28/2018 12:00:00 AM   DSNY            Department of Sanitation
14  08/09/2018 08:29:37 AM    DOT        Department of Transportation
15  08/20/2018 12:00:00 AM   DSNY            Department of Sanitation
16  10/03/2018 12:00:00 AM   DSNY            Department of Sanitation
17  08/01/2018 11:28:16 AM    DPR  Department of Parks and Recreation
18  08/07/2018 08:25:22 AM    DOT        Department of Transportation
19  08/07/2018 08:29:29 AM    DOT        Department of Transportation

             complaint_type                   descriptor    location_type  \
0   Maintenance or Facility          Structure - Outdoors             Park
1   Maintenance or Facility             Garbage or Litter             Park
2   Maintenance or Facility            Hours of Operation             Park
3   Maintenance or Facility             Garbage or Litter             Park
4   Maintenance or Facility             Garbage or Litter             Park
5   Maintenance or Facility            Unsecured Facility             Park
6   Maintenance or Facility                  Grass/Weeds  Street/Curbside
7      Broken Parking Meter  Coin or Card Did Not Register           Street
8                  Graffiti                     Graffiti              NaN
9                  Graffiti                     Graffiti        Mixed Use
10                 Graffiti                     Graffiti      Residential
11  Maintenance or Facility          Structure - Outdoors             Park
12                 Graffiti                     Graffiti        Mixed Use
13                 Graffiti                     Graffiti        Mixed Use
14     Broken Parking Meter  Coin or Card Did Not Register           Street
15                 Graffiti                     Graffiti      Residential
16                 Graffiti                     Graffiti      Residential
17  Maintenance or Facility          Structure - Outdoors             Park
18     Broken Parking Meter                   No Receipt           Street
19     Broken Parking Meter                 Out of Order           Street

    … vehicle_type taxi_company_borough taxi_pick_up_location  \
0   …          NaN                  NaN                   NaN
1   …          NaN                  NaN                   NaN
2   …          NaN                  NaN                   NaN
3   …          NaN                  NaN                   NaN
4   …          NaN                  NaN                   NaN
5   …          NaN                  NaN                   NaN
6   …          NaN                  NaN                   NaN
```

```
7   …           NaN                   NaN                    NaN
8   …           NaN                   NaN                    NaN
9   …           NaN                   NaN                    NaN
10  …           NaN                   NaN                    NaN
11  …           NaN                   NaN                    NaN
12  …           NaN                   NaN                    NaN
13  …           NaN                   NaN                    NaN
14  …           NaN                   NaN                    NaN
15  …           NaN                   NaN                    NaN
16  …           NaN                   NaN                    NaN
17  …           NaN                   NaN                    NaN
18  …           NaN                   NaN                    NaN
19  …           NaN                   NaN                    NaN

   bridge_highway_name bridge_highway_direction road_ramp  \
0                  NaN                      NaN       NaN
1                  NaN                      NaN       NaN
2                  NaN                      NaN       NaN
3                  NaN                      NaN       NaN
4                  NaN                      NaN       NaN
5                  NaN                      NaN       NaN
6                  NaN                      NaN       NaN
7                  NaN                      NaN       NaN
8                  NaN                      NaN       NaN
9                  NaN                      NaN       NaN
10                 NaN                      NaN       NaN
11                 NaN                      NaN       NaN
12                 NaN                      NaN       NaN
13                 NaN                      NaN       NaN
14                 NaN                      NaN       NaN
15                 NaN                      NaN       NaN
16                 NaN                      NaN       NaN
17                 NaN                      NaN       NaN
18                 NaN                      NaN       NaN
19                 NaN                      NaN       NaN

   bridge_highway_segment   latitude  longitude  \
0                     NaN        NaN        NaN
1                     NaN  40.683778 -73.769744
2                     NaN        NaN        NaN
3                     NaN        NaN        NaN
4                     NaN        NaN        NaN
5                     NaN        NaN        NaN
6                     NaN  40.711091 -73.747639
7                     NaN  40.852601 -73.827816
8                     NaN        NaN        NaN
9                     NaN  40.700620 -73.917173
```

```
10                     NaN  40.873534 -73.876124
11                     NaN        NaN        NaN
12                     NaN  40.642204 -74.013354
13                     NaN  40.702813 -73.921863
14                     NaN  40.858032 -73.883510
15                     NaN  40.720532 -73.942730
16                     NaN  40.720444 -73.942716
17                     NaN        NaN        NaN
18                     NaN  40.747529 -73.941220
19                     NaN  40.840808 -73.843322

                                    location
0                                        NaN
1    (40.68377771085733, -73.76974413109498)
2                                        NaN
3                                        NaN
4                                        NaN
5                                        NaN
6   (40.711091308732435, -73.74763946439865)
7   (40.852601460220185, -73.82781602326443)
8                                        NaN
9    (40.70061981717157, -73.91717344071574)
10   (40.87353414980636, -73.87612406980227)
11                                       NaN
12   (40.64220432954161, -74.01335393443593)
13   (40.70281342017164, -73.92186290937498)
14   (40.85803154781598, -73.88350951977466)
15  (40.720532158642264, -73.94272993232262)
16   (40.72044431919613, -73.94271557756703)
17                                       NaN
18   (40.74752880238678, -73.94121980592674)
19   (40.84080758451802, -73.84332208936512)

[20 rows x 43 columns]
```

```
[11]: def flag_graffiti(row):
          if "graffiti" in row['complaint_type'].lower():
              return True
          elif "graffiti" in row['descriptor'].lower():
              return True
          elif "graffiti" in str(row['resolution_description']).lower():
              return True
          else:
              return False
```

```
[12]: type(flag_graffiti)
```

[12]: function

Test by passing in a fake row.

```
[13]: %%ipytest --tb=short
#When %%XXX should be the first thing in the cell,
#Otherwise, IPython tries to interpret it as a 'line magic' hence the error you␣
 ↪see.

def test_complaint_type():
    test_row = pd.Series({
        "complaint_type": "graffiti",
        "descriptor": "",
        "resolution_description": ""
    })
    assert flag_graffiti(test_row) == True


def test_descriptor():
    test_row = pd.Series({
        "complaint_type": "",
        "descriptor": "graffiti",
        "resolution_description": ""
    })
    assert flag_graffiti(test_row) == True


def test_description():
    test_row = pd.Series({
        "complaint_type": "",
        "descriptor": "",
        "resolution_description": "graffiti"
    })
    assert flag_graffiti(test_row) == True


def test_none():
    test_row = pd.Series({
        "complaint_type": "",
        "descriptor": "",
        "resolution_description": ""
    })
    assert flag_graffiti(test_row) == False


def test_mixed_cases():
    test_row = pd.Series({
        "complaint_type": "GrafFiti",
        "descriptor": "",
        "resolution_description": ""
    })
    assert flag_graffiti(test_row) == True


def test_substring():
    test_row = pd.Series({
        "complaint_type": "",
        "descriptor": "there's graffiti on the wall",
        "resolution_description": ""
    })
    assert flag_graffiti(test_row) == True
```

```
......
[100%]
6 passed in 0.02s
```

1.1.4 Step 2

Apply the function created in Step 1 to the 311 dataframe and create a new column called `graffiti_flag` that captures the output from the function.

Tip: There are two checks you can use to confirm that the function worked as expected.

- Make sure there are records tagged with `graffiti_flag` `True`.
- Make sure that more than one `complaint_type` has `graffiti_flag` `True` (and `False`).

[14]:
```
type(flag_graffiti)
```

[14]:
```
function
```

[16]:
```
# your code here
df['graffiti_flag'] = df.apply(flag_graffiti,axis=1)
```

[17]:
```
%%ipytest --tb=short

def test_graffiti_flag():
    assert 'graffiti_flag' in df.columns, "column missing"
    assert df.dtypes['graffiti_flag'] == 'bool', "column should be booleans"
```

```
.
[100%]
1 passed in 0.01s
```

1.1.5 Step 3

Create another dataframe `df_graffiti` that only contains records where `graffiti_flag` is `True`.

[22]:
```
# your code here
df_graffiti=df[df['graffiti_flag']==True]
```

[24]:
```
print(df_graffiti)
```

```
       Unnamed: 0.1  Unnamed: 0  unique_key            created_date  \
8               860         860    39897352  08/01/2018 08:21:08 AM
9               876         876    39891938  08/01/2018 08:24:41 AM
10              964         964    39897349  08/01/2018 08:40:34 AM
12             1007        1007    39893314  08/01/2018 08:47:45 AM
13             1013        1013    39897422  08/01/2018 08:49:15 AM
…                 …           …         …                       …
64510       2854717     2854906    43625225  08/23/2019 12:06:05 PM
64511       2854787     2854976    43625582  08/23/2019 12:16:45 PM
64528       2855845     2856034    43626398  08/23/2019 03:28:10 PM
64566       2857045     2857234    43625297  08/23/2019 07:48:53 PM
64574       2857951     2858140    43619750  08/23/2019 10:52:13 PM

                  closed_date agency                       agency_name  \
8                         NaN   DSNY            Department of Sanitation
9      08/30/2018 12:00:00 AM   DSNY            Department of Sanitation
10     08/23/2018 12:00:00 AM   DSNY            Department of Sanitation
12     08/20/2018 12:00:00 AM   DSNY            Department of Sanitation
13     08/28/2018 12:00:00 AM   DSNY            Department of Sanitation
…                         …    …                                   …
64510  08/23/2019 12:44:10 PM   NYPD      New York City Police Department
64511                     NaN   NYPD      New York City Police Department
64528  08/23/2019 07:28:02 PM   NYPD      New York City Police Department
64566  08/23/2019 09:35:52 PM   NYPD      New York City Police Department
64574                     NaN    DPR  Department of Parks and Recreation

                complaint_type                   descriptor  \
8                     Graffiti                     Graffiti
9                     Graffiti                     Graffiti
10                    Graffiti                     Graffiti
12                    Graffiti                     Graffiti
13                    Graffiti                     Graffiti
…                          …                           …
64510                 Graffiti  Police Report Not Requested
64511                 Graffiti      Police Report Requested
64528                 Graffiti  Police Report Not Requested
64566                 Graffiti  Police Report Not Requested
64574  Maintenance or Facility         Graffiti or Vandalism
```

```
                    location_type  …  taxi_company_borough  \
8                             NaN  …                   NaN
9                       Mixed Use  …                   NaN
10                    Residential  …                   NaN
12                      Mixed Use  …                   NaN
13                      Mixed Use  …                   NaN
…                               …  …                     …
64510             Street/Sidewalk  …                   NaN
64511            Store/Commercial  …                   NaN
64528  Residential Building/House  …                   NaN
64566            Store/Commercial  …                   NaN
64574                        Park  …                   NaN

      taxi_pick_up_location bridge_highway_name bridge_highway_direction  \
8                       NaN                 NaN                      NaN
9                       NaN                 NaN                      NaN
10                      NaN                 NaN                      NaN
12                      NaN                 NaN                      NaN
13                      NaN                 NaN                      NaN
…                         …                   …                        …
64510                   NaN                 NaN                      NaN
64511                   NaN                 NaN                      NaN
64528                   NaN                 NaN                      NaN
64566                   NaN                 NaN                      NaN
64574                   NaN                 NaN                      NaN

      road_ramp bridge_highway_segment   latitude  longitude  \
8           NaN                    NaN        NaN        NaN
9           NaN                    NaN  40.700620 -73.917173
10          NaN                    NaN  40.873534 -73.876124
12          NaN                    NaN  40.642204 -74.013354
13          NaN                    NaN  40.702813 -73.921863
…             …                      …          …          …
64510       NaN                    NaN  40.530913 -74.193853
64511       NaN                    NaN  40.691514 -73.942629
64528       NaN                    NaN  40.883622 -73.909778
64566       NaN                    NaN  40.724822 -73.947586
64574       NaN                    NaN  40.792303 -73.807228

                                      location graffiti_flag
8                                          NaN          True
9       (40.70061981717157, -73.91717344071574)          True
10      (40.87353414980636, -73.87612406980227)          True
12      (40.64220432954161, -74.01335393443593)          True
13      (40.70281342017164, -73.92186290937498)          True
…                                            …             …
64510   (40.53091270292588, -74.19385298584154)          True
64511  (40.691514455259444, -73.94262864668438)          True
```

```
64528    (40.8836219178934, -73.90977771805365)          True
64566    (40.72482178470172, -73.9475859471725)          True
64574   (40.79230300094072, -73.80722774980956)          True

[26380 rows x 44 columns]
```

[25]:
```
%%ipytest --tb=short

def test_all_have_graffiti():
    assert df_graffiti['graffiti_flag'].all(), "not all have graffiti_flag set␣
 ↪to True"
```

```
.
[100%]
1 passed in 0.01s
```

1.1.6 Step 4

Group your dataframe `df_graffiti` to get the count of requests per `community_board`. Identify which Community District has the highest count.

[33]:
```
# your code here
group=df_graffiti.groupby('community_board').size().reset_index(name='count')
```

[34]:
```
print(group)
```

```
              community_board  count
0               0 Unspecified    540
1                    01 BRONX    311
2                 01 BROOKLYN   1797
3                01 MANHATTAN    146
4                   01 QUEENS    572
..                        …      …
66          Unspecified BRONX    615
67       Unspecified BROOKLYN    591
68      Unspecified MANHATTAN    405
69         Unspecified QUEENS    254
70  Unspecified STATEN ISLAND      8

[71 rows x 2 columns]
```

1.1.7 Bonus 1

0.2 points

Create a `graffiti_flag2` column using only built-in pandas operations, i.e. without using a custom function (`def`). Another way to think about it: Instead of operating on a single row at a time, how can you operate across entire columns? See working with text data for clues.

```
[31]: graffiti_flag2
```

```
[31]: 0        False
      1         True
      2        False
      3         True
      4         True
               …
      64572    False
      64573    False
      64574     True
      64575    False
      64576    False
      Name: community_board, Length: 64577, dtype: bool
```

1.1.8 Bonus 2

0.2 points

Clean another column of the dataset. Include explanation and code for how you got there.

```
[12]: # your code here
```

Now turn in the assignment.

1.2 Tutorials

- Pythonic Data Cleaning With Pandas and NumPy
- "You're Not Mapping Rats, You're Mapping Gentrification"—article about bias in 311 data
- Read about the Spatial Data Equity Tool
- Intro to Plotly Express. You don't have to work through every one of these examples; just review to get familiar with what types of charts are possible.

1.2.1 Optional

- Python Tools for Record Linkage
- Reshaping and pivot tables
- How to reshape the layout of tables