Satej Soman
CAPP30254: Machine Learning for Public Policy
Spring 2019

HW 1
DIAGNOSTIC ASSIGNMENT

Notes

- Representative code snippets are interspersed with analysis and explanations below; all code is available on GitHub: `https://github.com/satejsoman/capp30254/tree/master/hw1/code`.
- Sources for data and techniques are cited at the end of this report.

1 Data Acquisition & Analysis

1.1 Chicago Open Data Portal

Chicago crime data is available, filtered by year, from the Chicago Data Portal (`https://data.cityofchicago.org/browse?category=Public%20Safety`). We can download this data and load it into a Pandas `DataFrame`:

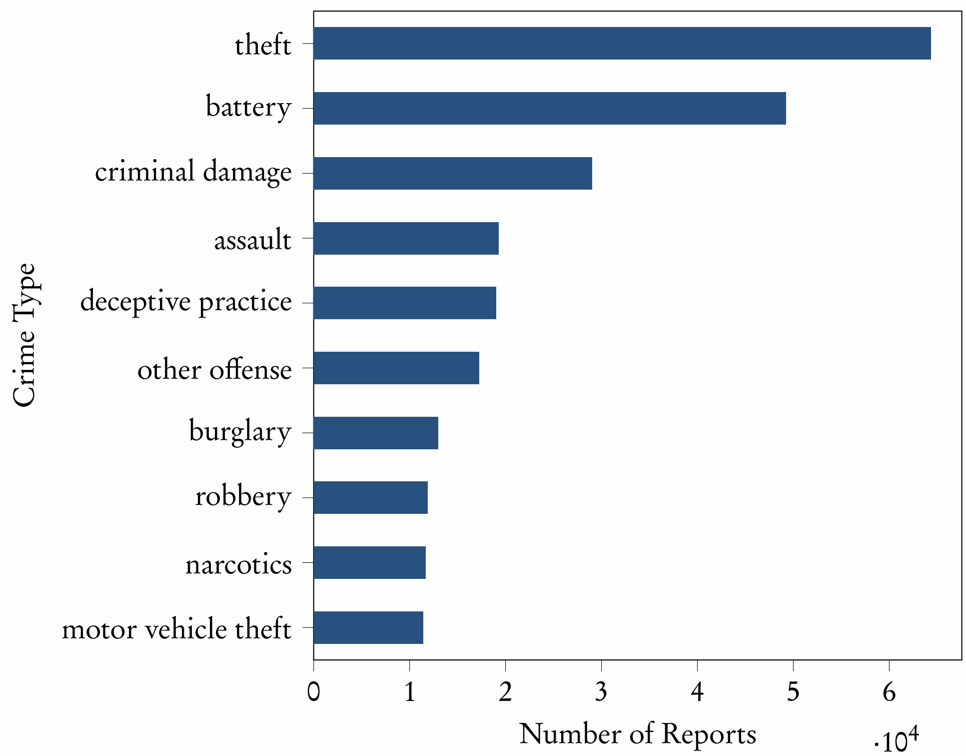
```
from pathlib import Path

import pandas as pd
import requests

# download crime data if we don't have it locally
base_url = "https://data.cityofchicago.org/api/views/{}/rows.csv?accessType=DOWNLOAD"
crime_resources = {
    2017: (Path("./crime_data_2017.csv"), "3i3m-jwuy"),
    2018: (Path("./crime_data_2018.csv"), "d62x-nvdr"),
}

for (year, (path, identifier)) in crime_resources.items():
    if not path.exists():
        url = base_url.format(identifier)
        print("{} data not found locally, downloading from {}".format(year, url))
        response = requests.get(url)
        with path.open("wb") as f:
            f.write(response.content)

crime_stats = pd.concat([
    pd.read_csv(crime_resources[2017][0]),
    pd.read_csv(crime_resources[2018][0])
])
```

1.2 Summary Statistics for Crime Report Data, 2017-2018

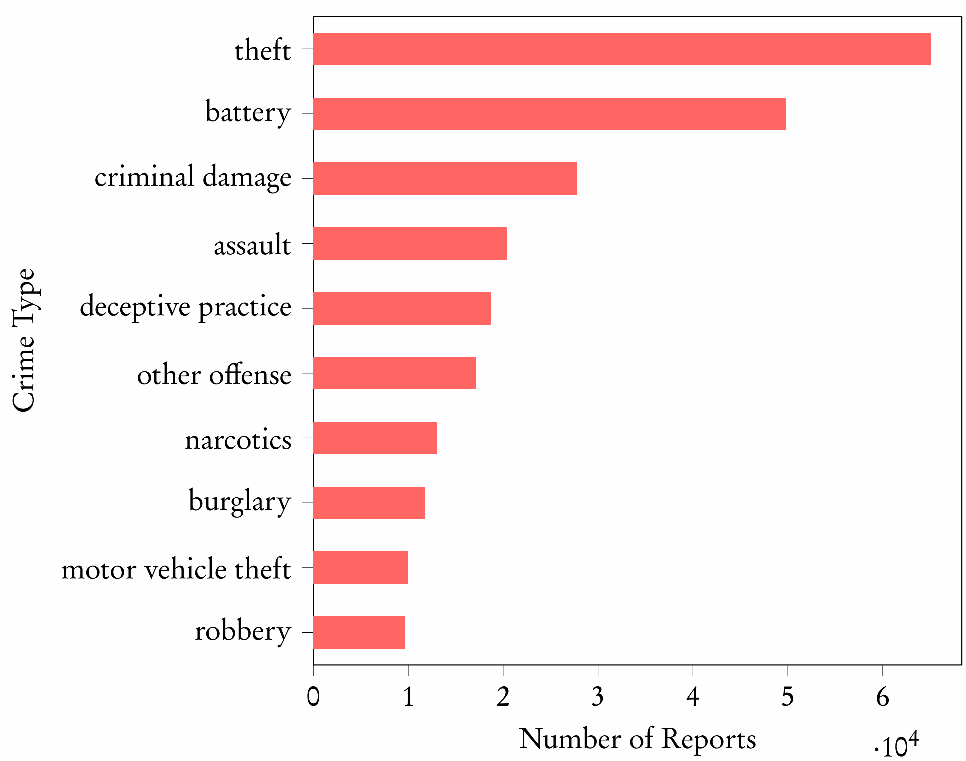
year	2017	2018	AVG
number of reported crimes	268094	266246	267170

year	2017	2018	OVERALL
crimes involving an arrest	19.53%	19.75%	19.64%
crimes considered domestic	15.90%	16.39%	16.14%

TOP CRIME TYPES IN CHICAGO (2017)



TOP CRIME TYPES IN CHICAGO (2018)



2 Data Augmentation & APIs

2.1 Chicago Crime Reports, Augmented with ACS Demographic Information

To pull in data from the American Community Survey, we need to identify which census tract each crime report corresponds to. This correspondence can be found by performing a *spatial join*: with shapefiles representing the geometry of each Chicago-area census tracts as a polygon, each crime report's latitude/longitude pair can be assigned to a census tract based on which geometry contains the report's coordinates. Census tract shapefiles are available from the City of Chicago's Data Portal.

```
from shapely.geometry import Point
import geopandas as gpd

def assign_census_tracts(crime_stats):
    boundary_shp = "./Boundaries - Census Blocks -
        2000/geo_export_8e9f6d85-3c5b-429f-b625-25afcc3dea85.shp"
    census_tracts = gpd.read_file(boundary_shp).drop(columns=["perimeter", "shape_area",
        "shape_len"])
    # restrict geocoding to valid locations
    crime_stats = crime_stats[crime_stats["Location"].notna()]
    crime_stats["geometry"] = crime_stats.apply(lambda row: Point(row["Longitude"],
        row["Latitude"]), axis = 1)
    return gpd.tools.sjoin(gpd.GeoDataFrame(crime_stats), census_tracts, how="inner")
```

With the census tracts assigned, we can query the Census Bureau's API via DataMade's census Python package to find representative data about each census tract. We'll need the following ACS variables to pull in demographic data:

ACS VARIABLE NAME	LABEL
B01003_001E	total population
B02001_003E	race (Black or African American alone)
B03003_001E	Hispanic or Latino origin
B19013_001E	median household income in the past 12 months
B22002_001E	receipt of food stamps/SNAP in the past 12 months by children under 18

```
from census import Census

census_client = Census(census_api_key)

illinois = "17"
cook_county = "031"
acs_vars = {
    "NAME" : "tract_name",
    "B01003_001E": "total_pop",
    "B02001_003E": "black_pop",
    "B03003_003E": "hispanic_pop",
    "B19013_001E": "median_income",
    "B22002_001E": "child_snap"
}

tract_numbers = set(crime_stats["census_tra"].to_list())
```

```
response = census_client.acs5.state_county_tract(list(acs_vars.keys()), illinois, cook_county,
    Census.ALL)
demography = pd.DataFrame([elem for elem in response if elem["tract"] in
    tract_numbers]).rename(columns=acs_vars)
# normalize by population
demography[["black_pct", "hispanic_pct", "child_snap_pct"]] = demography[["black_pop",
    "hispanic_pop", "child_snap"]].div(demography.total_pop, axis=0)
demography["tract"] = pd.to_numeric(demography["tract"])
demography.set_index("tract")
crime_stats["census_tra"] = pd.to_numeric(crime_stats["census_tra"])
crime_stats = crime_stats.merge(demography, left_on=["census_tra"], right_on=["tract"])

demographic_vars = ["black_pct", "hispanic_pct", "child_snap_pct", "median_income"]

# battery
crime_stats[crime_stats["Primary Type"] == "BATTERY"][demographic_vars].describe()
# homicide
crime_stats[crime_stats["Primary Type"] == "HOMICIDE"][demographic_vars].describe()

# homicide over time
crime_stats[(crime_stats["Primary Type"] == "HOMICIDE") & (crime_stats["Year"] ==
    2017)][demographic_vars].describe()
crime_stats[(crime_stats["Primary Type"] == "HOMICIDE") & (crime_stats["Year"] ==
    2018)][demographic_vars].describe()

# deceptive practice vs. sex offense
crime_stats[crime_stats["Primary Type"] == "DECEPTIVE PRACTICE"][demographic_vars].describe()
crime_stats[crime_stats["Primary Type"] == "SEX OFFENSE"][demographic_vars].describe()
```

2.1.1 What types of blocks have reports of "Battery"?

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.590492	0.197004	0.373272	43079.587196

The typical block with incidents of battery is generally roughly 60% Black and 20% Hispanic. On average, 37% of children receive food stamps, and the median income is about $43,000.

2.1.2 What types of blocks get "Homicide"?

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.732555	0.172105	0.352350	35031.494636

The typical block with incidents of battery is generally roughly 73% Black and 17% Hispanic. On average, 35% of children receive food stamps, and the median income is about $35,000.

2.1.3 Does that change over time in the data you collected?

2017 Homicide characteristics:

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.728826	0.185749	0.353375	34954.426374

2018 Homicide characteristics:

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.736974	0.155940	0.351134	35122.81250

Comparing the 2017 to 2018 statistics, the characterization of the typical block for homicide stays effectively the same.

2.1.4 What is the difference in blocks that get "Deceptive Practice" vs "Sex Offense"?

Deceptive Practice:

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.359146	0.182752	0.425968	65201.237182

Sex Offense:

% BLACK	% HISPANIC	% CHILDREN ON SNAP	MEDIAN INCOME
0.413284	0.255555	0.385575	52807.952566

Comparing the block characteristics between the two crime types, blocks with deceptive practice reports tend to have a higher median income (though also a higher percentage of children receiving food assistance). They also tend to have fewer Black or Hispanic residents than blocks with sex offense reports.

3 Analysis & Communication

3.1 Changes in Crime, 2017-2018

3.2 Analysis of Jacob Ringer's Claims

Jacob Ringer, a candidate for alderman for the 43rd Ward, claims:

> Let's break down the Chicago Police Departments report for the month leading up to July 26, 2018, compared to the same week in 2017:
>
> - Robberies – up 21 percent over the same time-frame in 2017
> - Aggravated batteries – up 136 percent
> - Burglaries – an increase of 50 percent
> - Motor vehicle theft – up 41 percent.
>
> All told, crime rose 16 percent in the same 28-day time period in just one year.

To evaluate Ringer's claims, we can filter down the relevant ward, and isolate the time periods he analyzes.

```
import datetime

one_month = datetime.timedelta(days = 28) # "same 28-day time period in just one year"
target17 = datetime.datetime(year=2017, month=7, day=26)
target18 = datetime.datetime(year=2018, month=7, day=26)
```

```
crime_stats = crime_stats[crime_stats["Ward"] == 43]
crime_stats["Date"] = pd.to_datetime(ward_crime_stats["Date"])
crime_stats = crime_stats[
    ((target17 - one_month <= crime_stats["Date"]) & (crime_stats["Date"] <= target17)) |
    ((target18 - one_month <= crime_stats["Date"]) & (crime_stats["Date"] <= target18))
]

# overall changes
crime_stats.groupby("Year").count()["ID"]
crime_stats.groupby("Year").count()["ID"].pct_change()

crime_stats = crime_stats[crime_stats["Primary Type"].isin(["ROBBERY", "BATTERY", "BURGLARY",
    "MOTOR VEHICLE THEFT"])]
crime_agg_2017 = crime_stats[crime_stats["Year"] == 2017]["Primary Type"].value_counts()
crime_agg_2018 = crime_stats[crime_stats["Year"] == 2018]["Primary Type"].value_counts()
crime_agg_2017.name, crime_agg_2018.name = "2017", "2018"
pd.DataFrame([crime_agg_2017, crime_agg_2018])
```

Calculated changes in crime for the 43rd ward:

year	2017	2018	ACTUAL % CHANGE	CLAIMED % CHANGE
all crimes	340	378	+11.18 %	+16 %
battery	38	33	-13.16 %	+136 %
robbery	17	8	-52.94 %	+21 %
burglary	16	13	-18.75 %	+50 %
motor vehicle theft	5	10	+100.00 %	+41 %

While some of Ringer's claims are directionally correct, the magnitudes of his crime statistics should be rejected overall.

3.3 Key Findings

1. The majority of crime across the time period analyzed is dominated by incidents of: theft, battery, criminal damage, and assault.
2. The proportion of motor vehicle theft and narcotics crimes are growing from 2017 to 2018, while robbery incidents are proportionally decreasing.
3.
4.
5.

3.4 Caveats & Limitations

Some caveats apply to this analysis:

- Demographic information comes from the American Community Survey, in which responses are voluntary. These data may therefore be incomplete or flawed due to non-response.
- Statistics solely about crime are from a two-year window; a more comprehensive analysis would take into account several years of crime statistics.

4 Probability Exercise

4.1 Probabilities of Crime Type for a Call from a Given Address

We can aggregate the crime types for the given block to see that battery is the most probably report type for the block at 2111 S Michigan Ave. The overall probabilities can also be calculated:

```
def analyze_crime_for_block(crime_stats, block_address):
    return 100 * crime_stats[crime_stats["Block"].str.contains("021XX S MICHIGAN")]["Primary
        Type"].value_counts(normalize=True)

analyze_crime_for_block(crime_stats, "021XX S MICHIGAN")
```

PRIMARY TYPE	PROBABILITY
battery	26.667%
other offense	21.667%
criminal damage	10.000%
theft	10.000%
assault	10.000%
deceptive practice	10.000%
robbery	3.333%
motor vehicle theft	3.333%
burglary	1.667%
public peace violation	1.667%
criminal trespass	1.667%

4.2 Theft in Garfield Park vs. Uptown

The City of Chicago's Data Portal indicated that Garfield Park corresponds to community areas 26 and 27 and Uptown to community area 3. With this mapping, we can aggregate community areas over reports of theft and find the probabilities:

```
def theft_probabilities(crime_stats, areas):
    return 100 * crime_stats[crime_stats["Primary Type"] == "THEFT"]["Community
        Area"].value_counts(normalize=True)[[float(a) for a in areas]]

theft_probabilities(crime_stats, [26, 27, 3])
```

COMMUNITY AREA	PROBABILITY OF ORIGIN, GIVEN THEFT CALL
26.0	0.937%
27.0	0.990%
3.0	1.510%

The total probability of the call being from Garfield Park is 1.927%, which is 0.42 percentage points more likely than Uptown.

4.3 Calculation under Simulated Frequencies

We can use Bayes' Theorem to calculate the conditional probabilities:

$$P(A\,|\,B) = \frac{P(B|A)P(A)}{P(A)}$$

From the problem statement, we know:

$$P(\text{GARFIELD PARK}) = \frac{600}{1000} = 0.6$$

$$P(\text{UPTOWN}) = \frac{400}{1000} = 0.4$$

$$P(\text{BATTERY}) = \frac{100 + 160}{1000} = 0.26$$

$$P(\text{BATTERY}\,|\,\text{GARFIELD PARK}) = \frac{100}{600} = 0.16667$$

$$P(\text{BATTERY}\,|\,\text{UPTOWN}) = \frac{160}{400} = 0.4$$

Therefore,

$$\begin{aligned} P(\text{GARFIELD PARK}\,|\,\text{BATTERY}) &= \frac{P(\text{BATTERY}\,|\,\text{GARFIELD PARK})P(\text{GARFIELD PARK})}{P(\text{BATTERY})} \\ &= \frac{0.16667 \cdot 0.6}{0.26} = 0.3846 \end{aligned}$$

$$\begin{aligned} P(\text{UPTOWN}\,|\,\text{BATTERY}) &= \frac{P(\text{BATTERY}\,|\,\text{UPTOWN})P(\text{UPTOWN})}{P(\text{BATTERY})} \\ &= \frac{0.4 \cdot 0.4}{0.26} = 0.6154 \end{aligned}$$

From these calculations, a call about battery is 23% more likely to come from Uptown than from Garfield Park.

Sources