

denvercrimeandweatherFINAL

December 11, 2022

Crime and Weather

Graphs and Exploratory Analysis

Lauren Paredes

```
[1]: import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
```

Read in both crime and weather into data frames using pandas.

```
[2]: crimeDF= pd.read_csv("crime.csv", encoding='windows-1254')
weatherDF = pd.read_csv("weather_data_long.csv")
```

Clean up and trim down the dataframes

```
[3]: del weatherDF['Time']
del weatherDF["WinSpeed"]
del weatherDF["Pressure"]
del weatherDF["Humidity"]
del weatherDF["Wind"]
del weatherDF["DewPoint"]
del weatherDF["WindGust"]
del weatherDF["Precip."]
del weatherDF["Condition"]
```

```
[4]: del crimeDF["incident_id"]
del crimeDF["offense_id"]
del crimeDF["OFFENSE_CODE"]
del crimeDF["OFFENSE_CODE_EXTENSION"]
del crimeDF["LAST_OCCURRENCE_DATE"]
del crimeDF["REPORTED_DATE"]
del crimeDF["VICTIM_COUNT"]
del crimeDF["GEO_X"]
del crimeDF["GEO_Y"]
del crimeDF["GEO_LON"]
del crimeDF["GEO_LAT"]
del crimeDF["DISTRICT_ID"]
```

```
del crimeDF["PRECINCT_ID"]
del crimeDF["NEIGHBORHOOD_ID"]
del crimeDF["IS_CRIME"]
del crimeDF["IS_TRAFFIC"]
del crimeDF["INCIDENT_ADDRESS"]
```

```
[5]: # need the occurrence date to be in the format of datetime
crimeDF['Date'] = pd.to_datetime(crimeDF["FIRST_OCCURRENCE_DATE"]).dt.date
del crimeDF["FIRST_OCCURRENCE_DATE"]
# crimeDF.rename(columns={'OCCURRENCE_DATE': 'Date'}, inplace = True)
```

```
[6]: # Making sure both dataframes have a common col to merge on with the correct
      ↳ datatype
crimeDF['Date'] = pd.to_datetime(crimeDF['Date'])
weatherDF['Date'] = pd.to_datetime(weatherDF['Date'])
```

```
[7]: # Set the temperature for the average divided by group
weatherDF = weatherDF.groupby(['Date']).max()
```

Merging crime and weather on a common date

```
[8]: mergeOnDate = pd.merge(crimeDF, weatherDF, how='outer', on='Date')
mergeOnDate = mergeOnDate.dropna(subset=['OFFENSE_TYPE_ID'])
mergeOnDate = mergeOnDate.dropna(subset=['Temperature'])
mergeOnDate['Temperature'] = mergeOnDate['Temperature'].astype('int')
display(mergeOnDate)
```

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date \ |
|--------|-----------------------------|---------------------|------------|
| 0 | criminal-mischief-other | public-disorder | 2017-06-25 |
| 1 | criminal-mischief-other | public-disorder | 2017-06-25 |
| 2 | criminal-mischief-other | public-disorder | 2017-06-25 |
| 3 | criminal-mischief-other | public-disorder | 2017-06-25 |
| 4 | criminal-mischief-other | public-disorder | 2017-06-25 |
| ... | ... | ... | ... |
| 379047 | fraud-by-use-of-computer | white-collar-crime | 2019-01-14 |
| 379048 | fraud-by-use-of-computer | white-collar-crime | 2019-01-14 |
| 379049 | pawn-broker-viol | all-other-crimes | 2019-01-14 |
| 379050 | outside-steal-recovered-veh | all-other-crimes | 2019-01-14 |
| 379051 | outside-steal-recovered-veh | all-other-crimes | 2019-01-14 |

| | Temperature |
|--------|-------------|
| 0 | 73 |
| 1 | 73 |
| 2 | 73 |
| 3 | 73 |
| 4 | 73 |
| ... | ... |
| 379047 | 43 |

```
379048      43
379049      43
379050      43
379051      43
```

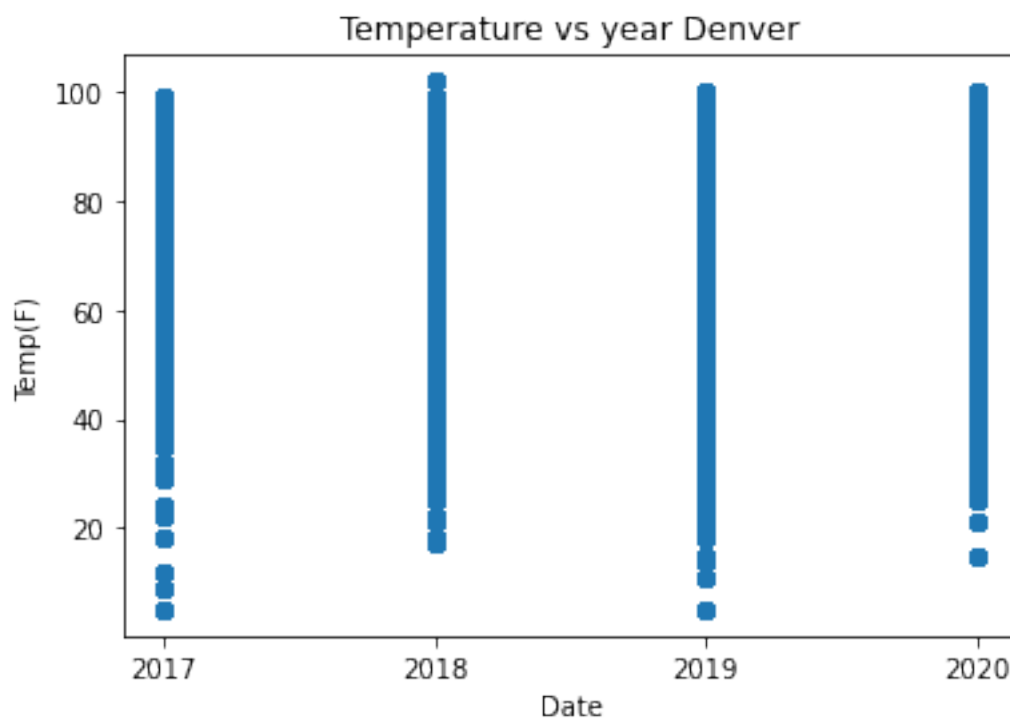
[228089 rows x 4 columns]

Graphs and Exploratory Analysis

```
[9]: graphJoin = mergeOnDate
graphJoin['Year'] = pd.DatetimeIndex(graphJoin['Date']).year
graphJoin['Month'] = pd.DatetimeIndex(graphJoin['Date']).month
```

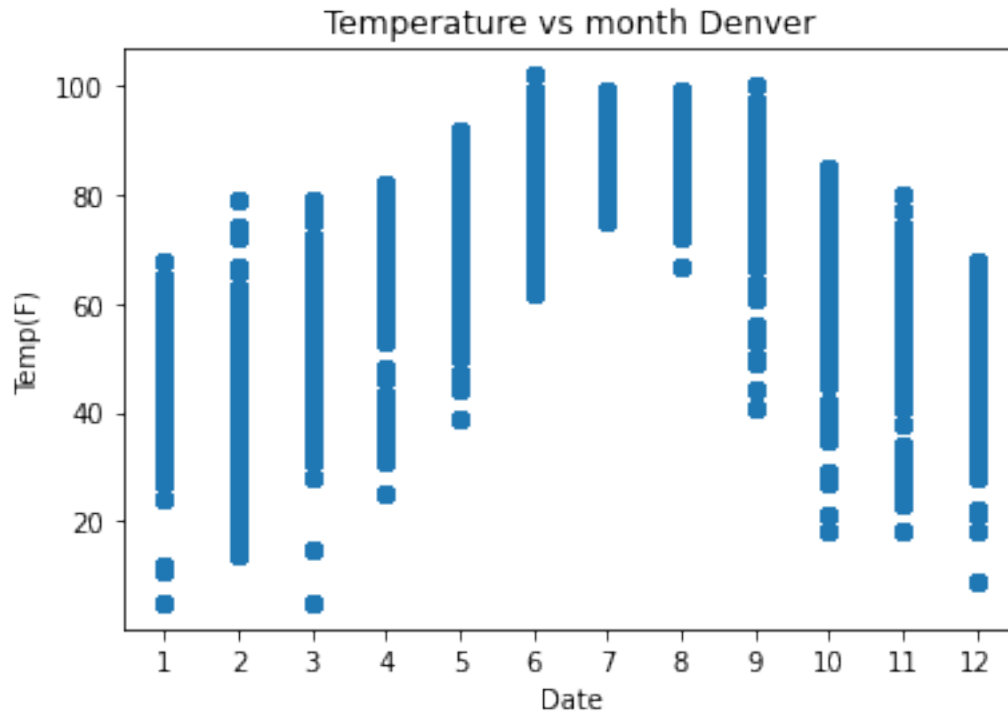
```
[10]: plt.scatter(graphJoin['Year'],graphJoin['Temperature'])
plt.title("Temperature vs year Denver")

plt.xticks(np.arange( min(graphJoin["Year"]) , max(graphJoin["Year"])+1 ,1.0))
plt.xlabel("Date")
plt.ylabel("Temp(F)")
plt.show()
```

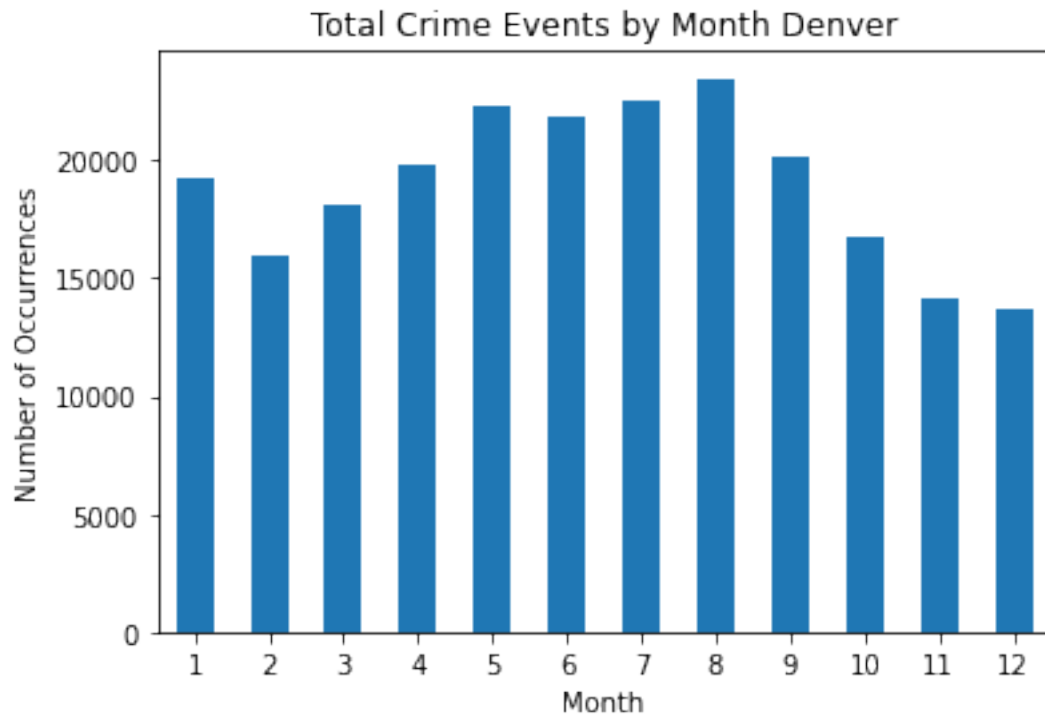


```
[11]: plt.scatter(graphJoin['Month'],graphJoin['Temperature'])
plt.title("Temperature vs month Denver")
```

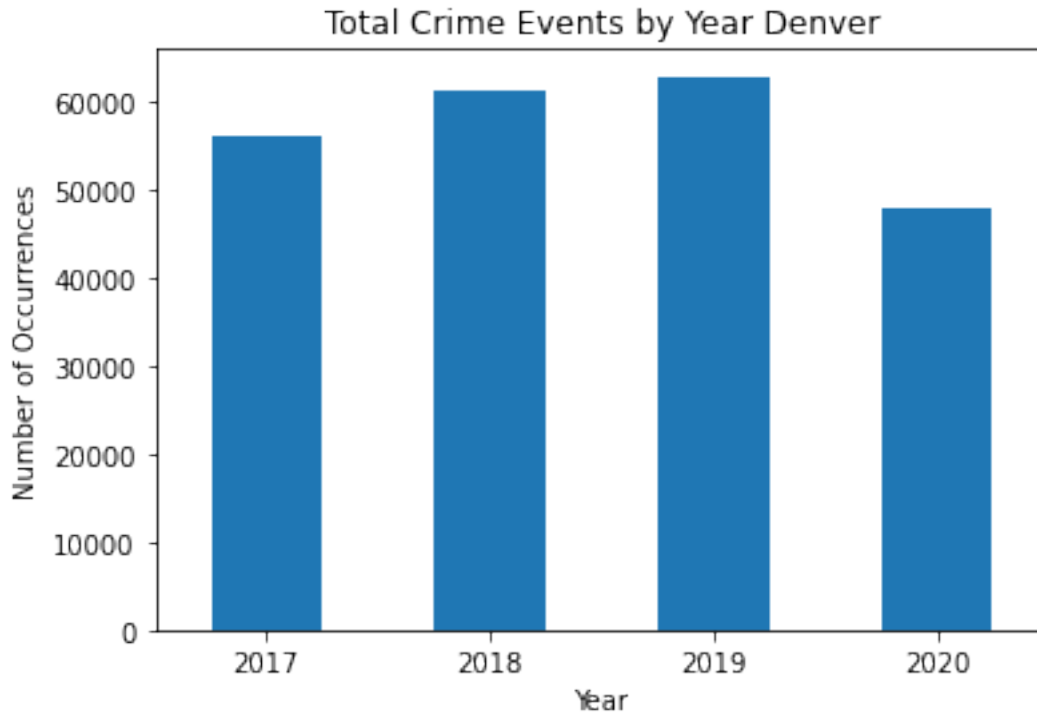
```
plt.xticks(np.arange( min(graphJoin["Month"]) , max(graphJoin["Month"]+1 ),1.0))
plt.xlabel("Date")
plt.ylabel("Temp(F)")
plt.show()
```



```
[12]: ax = graphJoin.groupby("Month").size().plot(kind = "bar", title= "Total Crime_
      ↳Events by Month Denver")
ax.set_ylabel("Number of Occurrences")
plt.xticks(rotation=0)
plt.show()
```



```
[13]: ax = graphJoin.groupby("Year").size().plot(kind = "bar", title= "Total Crime_Events by Year Denver")
ax.set_ylabel("Number of Occurrences")
plt.xticks(rotation=0)
plt.show()
```



Assigning crime types in terms of society, person, property. This categorization logic is explained in report.

```
[14]: # all-other-crimes = 37218/total size = 12.92
offenseType = (mergeOnDate["OFFENSE_TYPE_ID"].value_counts())
print(mergeOnDate.shape)
mergeOnDate= mergeOnDate[~mergeOnDate['OFFENSE_TYPE_ID'].
    ↪isin(offenseType[offenseType < 5000].index)]

t=mergeOnDate.groupby('OFFENSE_CATEGORY_ID').size()
g=mergeOnDate.groupby('OFFENSE_TYPE_ID')
# mergeOnDate.to_csv('CrimeTypeMapping')
display(mergeOnDate)
display(t)
print(mergeOnDate.shape)
```

(228089, 6)

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date | Temperature | \ |
|-----|-------------------------|---------------------|------------|-------------|---|
| 0 | criminal-mischief-other | public-disorder | 2017-06-25 | 73 | |
| 1 | criminal-mischief-other | public-disorder | 2017-06-25 | 73 | |
| 2 | criminal-mischief-other | public-disorder | 2017-06-25 | 73 | |
| 3 | criminal-mischief-other | public-disorder | 2017-06-25 | 73 | |
| 4 | criminal-mischief-other | public-disorder | 2017-06-25 | 73 | |
| ... | ... | ... | ... | ... | |

| | | | | |
|--------|------------------------|------------|------------|----|
| 379037 | theft-of-motor-vehicle | auto-theft | 2019-01-14 | 43 |
| 379038 | theft-of-motor-vehicle | auto-theft | 2019-01-14 | 43 |
| 379039 | theft-of-motor-vehicle | auto-theft | 2019-01-14 | 43 |
| 379040 | theft-of-motor-vehicle | auto-theft | 2019-01-14 | 43 |
| 379041 | theft-of-motor-vehicle | auto-theft | 2019-01-14 | 43 |

| | Year | Month |
|--------|------|-------|
| 0 | 2017 | 6 |
| 1 | 2017 | 6 |
| 2 | 2017 | 6 |
| 3 | 2017 | 6 |
| 4 | 2017 | 6 |
| ... | ... | ... |
| 379037 | 2019 | 1 |
| 379038 | 2019 | 1 |
| 379039 | 2019 | 1 |
| 379040 | 2019 | 1 |
| 379041 | 2019 | 1 |

[131376 rows x 6 columns]

| OFFENSE_CATEGORY_ID | |
|------------------------------|-------|
| all-other-crimes | 11557 |
| auto-theft | 20412 |
| burglary | 5803 |
| larceny | 28935 |
| other-crimes-against-persons | 15521 |
| public-disorder | 19475 |
| theft-from-motor-vehicle | 29673 |

dtype: int64

(131376, 6)

```
[15]: h= mergeOnDate.groupby(['OFFENSE_TYPE_ID']).size()
```

```
h.shape
h.head(30)
```

```
[15]: OFFENSE_TYPE_ID
assault-dv          6061
assault-simple      9460
burglary-residence-no-force  5803
criminal-mischief-mtr-veh  10005
criminal-mischief-other    9470
criminal-trespassing    11557
theft-bicycle         6202
theft-items-from-vehicle  20159
theft-of-motor-vehicle  20412
```

```

theft-other                13685
theft-parts-from-vehicle    9514
theft-shoplift              9048
dtype: int64

```

```

[16]: types ={'auto-theft': 'property',
              'robbery': 'property',
              'arson': 'property',
              'theft-from-motor-vehicle': 'property',
              'burglary': 'property',
              'larceny': 'property',
              'sexual-assault': 'person',
              'drug-alcohol': 'society',
              'other-crimes-against-persons': 'person',
              'aggravated-assault': 'person',
              'murder': 'person',
              'white-collar-crime': 'society',
              'public-disorder': 'society',
              'all-other-crimes': 'society'}

```

```

[17]: mergeOnDate["OFFENSE_CATEGORY_ID"] = mergeOnDate["OFFENSE_CATEGORY_ID"].
      ↪map(types)
      display(mergeOnDate)

```

```

/var/folders/pd/tbnq7vld3nz_s0qzjlhjvr2w0000gn/T/ipykernel_12530/3593886119.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

mergeOnDate["OFFENSE_CATEGORY_ID"] =
mergeOnDate["OFFENSE_CATEGORY_ID"].map(types)

```

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date | Temperature | \ |
|--------|-------------------------|---------------------|------------|-------------|---|
| 0 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 1 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 2 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 3 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 4 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| ... | ... | ... | ... | ... | |
| 379037 | theft-of-motor-vehicle | property | 2019-01-14 | 43 | |
| 379038 | theft-of-motor-vehicle | property | 2019-01-14 | 43 | |
| 379039 | theft-of-motor-vehicle | property | 2019-01-14 | 43 | |
| 379040 | theft-of-motor-vehicle | property | 2019-01-14 | 43 | |
| 379041 | theft-of-motor-vehicle | property | 2019-01-14 | 43 | |

Year Month

| | | |
|--------|------|-----|
| 0 | 2017 | 6 |
| 1 | 2017 | 6 |
| 2 | 2017 | 6 |
| 3 | 2017 | 6 |
| 4 | 2017 | 6 |
| ... | ... | ... |
| 379037 | 2019 | 1 |
| 379038 | 2019 | 1 |
| 379039 | 2019 | 1 |
| 379040 | 2019 | 1 |
| 379041 | 2019 | 1 |

[131376 rows x 6 columns]

Separate into hot, mild, cold dataframes

```
[18]: hotDays = mergeOnDate[mergeOnDate['Temperature'] > 80]
mildDays = mergeOnDate[(mergeOnDate['Temperature'] >= 50) &
↳ (mergeOnDate['Temperature'] <= 80)]
coldDays = mergeOnDate[mergeOnDate['Temperature'] < 50]

numofweathercrimes = len(mergeOnDate.index)
print("Total Entry Count: ", numofweathercrimes)
print("Hot days overview")
display(hotDays)
print("Mild days overview")
display(mildDays)
print("cold days overview")
display(coldDays)
```

Total Entry Count: 131376

Hot days overview

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date | Temperature | \ |
|--------|-------------------------|---------------------|------------|-------------|---|
| 347 | criminal-mischief-other | society | 2017-06-27 | 96 | |
| 348 | criminal-mischief-other | society | 2017-06-27 | 96 | |
| 349 | criminal-mischief-other | society | 2017-06-27 | 96 | |
| 350 | criminal-mischief-other | society | 2017-06-27 | 96 | |
| 351 | criminal-mischief-other | society | 2017-06-27 | 96 | |
| ... | ... | ... | ... | ... | |
| 377506 | theft-of-motor-vehicle | property | 2018-07-02 | 94 | |
| 377507 | theft-of-motor-vehicle | property | 2018-07-02 | 94 | |
| 377508 | theft-of-motor-vehicle | property | 2018-07-02 | 94 | |
| 377509 | theft-of-motor-vehicle | property | 2018-07-02 | 94 | |
| 377510 | theft-of-motor-vehicle | property | 2018-07-02 | 94 | |
| | | | | | |
| | Year | Month | | | |
| 347 | 2017 | 6 | | | |
| 348 | 2017 | 6 | | | |

| | | |
|--------|------|-----|
| 349 | 2017 | 6 |
| 350 | 2017 | 6 |
| 351 | 2017 | 6 |
| ... | ... | ... |
| 377506 | 2018 | 7 |
| 377507 | 2018 | 7 |
| 377508 | 2018 | 7 |
| 377509 | 2018 | 7 |
| 377510 | 2018 | 7 |

[43371 rows x 6 columns]

Mild days overview

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date | Temperature | \ |
|--------|-------------------------|---------------------|------------|-------------|---|
| 0 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 1 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 2 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 3 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| 4 | criminal-mischief-other | society | 2017-06-25 | 73 | |
| ... | ... | ... | ... | ... | |
| 377016 | theft-of-motor-vehicle | property | 2018-04-08 | 63 | |
| 377017 | theft-of-motor-vehicle | property | 2018-04-08 | 63 | |
| 377018 | theft-of-motor-vehicle | property | 2018-04-08 | 63 | |
| 377019 | theft-of-motor-vehicle | property | 2018-04-08 | 63 | |
| 377020 | theft-of-motor-vehicle | property | 2018-04-08 | 63 | |

| | Year | Month |
|--------|------|-------|
| 0 | 2017 | 6 |
| 1 | 2017 | 6 |
| 2 | 2017 | 6 |
| 3 | 2017 | 6 |
| 4 | 2017 | 6 |
| ... | ... | ... |
| 377016 | 2018 | 4 |
| 377017 | 2018 | 4 |
| 377018 | 2018 | 4 |
| 377019 | 2018 | 4 |
| 377020 | 2018 | 4 |

[60827 rows x 6 columns]

cold days overview

| | OFFENSE_TYPE_ID | OFFENSE_CATEGORY_ID | Date | Temperature | \ |
|------|-------------------------|---------------------|------------|-------------|---|
| 2034 | criminal-mischief-other | society | 2017-04-01 | 40 | |
| 2035 | criminal-mischief-other | society | 2017-04-01 | 40 | |
| 2036 | criminal-mischief-other | society | 2017-04-01 | 40 | |
| 2037 | criminal-mischief-other | society | 2017-04-01 | 40 | |
| 2038 | criminal-mischief-other | society | 2017-04-01 | 40 | |

| | | | | |
|--------|------------------------|----------|------------|-----|
| ... | ... | ... | ... | ... |
| 379037 | theft-of-motor-vehicle | property | 2019-01-14 | 43 |
| 379038 | theft-of-motor-vehicle | property | 2019-01-14 | 43 |
| 379039 | theft-of-motor-vehicle | property | 2019-01-14 | 43 |
| 379040 | theft-of-motor-vehicle | property | 2019-01-14 | 43 |
| 379041 | theft-of-motor-vehicle | property | 2019-01-14 | 43 |

| | | |
|------|------|-------|
| | Year | Month |
| 2034 | 2017 | 4 |
| 2035 | 2017 | 4 |
| 2036 | 2017 | 4 |
| 2037 | 2017 | 4 |
| 2038 | 2017 | 4 |

| | | |
|--------|------|-----|
| ... | ... | ... |
| 379037 | 2019 | 1 |
| 379038 | 2019 | 1 |
| 379039 | 2019 | 1 |
| 379040 | 2019 | 1 |
| 379041 | 2019 | 1 |

[27178 rows x 6 columns]

General distributions of crime types on different temperature categories

```
[19]: # overall merged data counts
print("All Data grouped by categoryID counts")
crimetypes = mergeOnDate.groupby(["OFFENSE_CATEGORY_ID"]).size()
print(crimetypes.head())
print(mergeOnDate.shape)
```

```
All Data grouped by categoryID counts
OFFENSE_CATEGORY_ID
person      15521
property    84823
society     31032
dtype: int64
(131376, 6)
```

```
[20]: # category types on hot days from the hotDays dataframe
print("Hot Data grouped by categoryID counts")
hotTypes=hotDays.groupby(["OFFENSE_CATEGORY_ID"]).size()
print(hotTypes.head())
print(hotDays.shape)
```

```
Hot Data grouped by categoryID counts
OFFENSE_CATEGORY_ID
person      4976
property    28375
society     10020
```

```
dtype: int64
(43371, 6)
```

```
[21]: # category types on mild days from the mildDays dataframe
print("Mild Data grouped by categoryID counts")
mildTypes = mildDays.groupby(["OFFENSE_CATEGORY_ID"]).size()
print(mildTypes.head())
print(mildDays.shape)
```

```
Mild Data grouped by categoryID counts
OFFENSE_CATEGORY_ID
person          7277
property        39075
society         14475
dtype: int64
(60827, 6)
```

```
[22]: # category types on cold days from the coldDays dataframe
print("Cold Data grouped by categoryID counts")
coldTypes = coldDays.groupby(["OFFENSE_CATEGORY_ID"]).size()
print(coldTypes.head())
print(coldDays.shape)
```

```
Cold Data grouped by categoryID counts
OFFENSE_CATEGORY_ID
person          3268
property        17373
society         6537
dtype: int64
(27178, 6)
```

```
[23]: pdisHot=(hotTypes/numofweathercrimes)*100
pdisCold=(coldTypes/numofweathercrimes)*100
pdisMild = (mildTypes/numofweathercrimes)*100
print("General Disributions for type and temperature")
print("Hot general Distribution:")
display(pdisHot)
print("Mild general Distribution:")
display(pdisMild)
print("Cold general Distribution:")
display(pdisCold)
```

```
General Disributions for type and temperature
Hot general Distribution:

OFFENSE_CATEGORY_ID
person          3.787602
property        21.598313
```

```
society      7.626964
dtype: float64
```

Mild general Distribution:

```
OFFENSE_CATEGORY_ID
person      5.539063
property    29.742875
society     11.017994
dtype: float64
```

Cold general Distribution:

```
OFFENSE_CATEGORY_ID
person      2.487517
property    13.223877
society      4.975795
dtype: float64
```

Bayesian Classifications

$P(\text{Category}|\text{Temp}) = \frac{P(C \text{ and } T)}{P(T)}$

P(Category) is represented by a series with Person, Property, and Society crimes P(Temperature) is represented by a series with Hot, Mild, and Cold days

```
[24]: # def calculateClassPropbs(dataset):
#      numDataPoint = dataset.size
#      classProbs= {}
#      for dataPoint in
```

```
[25]: priorCategories = (crimetypes / (numofweathercrimes))
print("Prior Probabilities of Categories")
display(priorCategories)
print("Sum of percentages of Categories: ",priorCategories.sum())
```

Prior Probabilities of Categories

```
OFFENSE_CATEGORY_ID
person      0.118142
property     0.645651
society      0.236208
dtype: float64
```

Sum of percentages of Categories: 1.0

```
[26]: pHot = ((hotDays.shape[0])/numofweathercrimes)
pMild= ((mildDays.shape[0])/numofweathercrimes)
pCold=((coldDays.shape[0])/numofweathercrimes)
d = {'Hot':pHot, 'Mild':pMild,'Cold':pCold}
priorTemperatures= pd.Series(data=d, index=['Hot','Mild','Cold'])
print("Probabilities of Temperatures")
```

```
display(priorTemperatures)
print("Sum of percentages of Temperatures",priorTemperatures.sum())
```

Probabilities of Temperatures

```
Hot      0.330129
Mild     0.462999
Cold     0.206872
dtype: float64
```

Sum of percentages of Temperatures 1.0

$P(T|C) = P(T \text{ and } C) * P(C)$

```
[27]: pHotTypes=(hotTypes/hotDays.shape[0])*100
pColdTypes=(coldTypes/coldDays.shape[0])*100
pMildTypes = (mildTypes/mildDays.shape[0])*100
print("Given a hot day probability of a type of crime: ")
print("Total sum: ", pHotTypes.sum())
display(pHotTypes)
print("Given a mild day probability of a type of crime: ")
print("Total sum: ", pColdTypes.sum())
display(pColdTypes)
print("Given a cold day probability of a type of crime: ")
print("Total sum: ", pMildTypes.sum())
display(pMildTypes)
# display(hotTypes.sum())
```

Given a hot day probability of a type of crime:
Total sum: 100.0

```
OFFENSE_CATEGORY_ID
person      11.473104
property    65.423901
society     23.102995
dtype: float64
```

Given a mild day probability of a type of crime:
Total sum: 100.0

```
OFFENSE_CATEGORY_ID
person      12.024432
property    63.923026
society     24.052542
dtype: float64
```

Given a cold day probability of a type of crime:
Total sum: 100.00000000000001

```
OFFENSE_CATEGORY_ID
person      11.963437
property    64.239565
```

```
society      23.796998  
dtype: float64
```