Richa Patel Final Project - Research Report

Rhode Island Police and local Weather data

In [92]:
```
import pandas as pd
import numpy as np
from numpy import cov
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="white", color_codes=True)


figsize = (16,8)

# read file for police and weather Data

police = pd.read_csv('police.csv')
print(police)

#we can see that we have 91741 rows and 15 columns

weather = pd.read_csv('weather.csv')
print(weather)

# [4017 rows x 27 columns]

print(police.describe())
police.info()

print(weather.describe())
weather.info()

police['district'].value_counts()
```

```
police['district'].value_counts()
```

```
      state   stop_date stop_time  county_name driver_gender driver_race  \
0        RI  2005-01-04     12:55          NaN             M       White
1        RI  2005-01-23     23:15          NaN             M       White
2        RI  2005-02-17     04:15          NaN             M       White
3        RI  2005-02-20     17:15          NaN             M       White
4        RI  2005-02-24     01:20          NaN             F       White
...     ...         ...       ...          ...           ...         ...
91736    RI  2015-12-31     21:21          NaN             F       Black
91737    RI  2015-12-31     21:59          NaN             F       White
91738    RI  2015-12-31     22:04          NaN             M       White
91739    RI  2015-12-31     22:09          NaN             F    Hispanic
91740    RI  2015-12-31     22:47          NaN             M       White

                      violation_raw           violation  search_conducted  \
0      Equipment/Inspection Violation           Equipment             False
1                           Speeding            Speeding             False
2                           Speeding            Speeding             False
3                  Call for Service               Other             False
4                           Speeding            Speeding             False
...                              ...                 ...               ...
91736         Other Traffic Violation    Moving violation             False
91737                       Speeding            Speeding             False
91738         Other Traffic Violation    Moving violation             False
91739  Equipment/Inspection Violation           Equipment             False
91740          Registration Violation  Registration/plates             False

      search_type   stop_outcome is_arrested stop_duration  \
0             NaN       Citation       False      0-15 Min
1             NaN       Citation       False      0-15 Min
2             NaN       Citation       False      0-15 Min
3             NaN  Arrest Driver        True     16-30 Min
4             NaN       Citation       False      0-15 Min
...           ...            ...         ...           ...
91736         NaN       Citation       False      0-15 Min
91737         NaN       Citation       False      0-15 Min
```

```
91738         NaN       Citation       False      0-15 Min
91739         NaN        Warning       False      0-15 Min
91740         NaN       Citation       False      0-15 Min

       drugs_related_stop district
0                   False  Zone X4
1                   False  Zone K3
2                   False  Zone X4
3                   False  Zone X1
4                   False  Zone X3
...                   ...      ...
91736               False  Zone K2
91737               False  Zone K3
91738               False  Zone X3
91739               False  Zone K3
91740               False  Zone X4

[91741 rows x 15 columns]
          STATION        DATE  TAVG  TMIN  TMAX   AWND  WSF2  WT01  WT02  \
0     USW00014765  2005-01-01  44.0    35    53   8.95  25.1   1.0   NaN
1     USW00014765  2005-01-02  36.0    28    44   9.40  14.1   NaN   NaN
2     USW00014765  2005-01-03  49.0    44    53   6.93  17.0   1.0   NaN
3     USW00014765  2005-01-04  42.0    39    45   6.93  16.1   1.0   NaN
4     USW00014765  2005-01-05  36.0    28    43   7.83  17.0   1.0   NaN
...           ...         ...   ...   ...   ...    ...   ...   ...   ...
4012  USW00014765  2015-12-27  51.0    44    61   9.17  28.0   1.0   NaN
4013  USW00014765  2015-12-28  40.0    30    44  12.30  23.0   NaN   NaN
4014  USW00014765  2015-12-29  33.0    28    40  12.53  18.1   1.0   NaN
4015  USW00014765  2015-12-30  30.0    27    35   6.93  15.0   1.0   NaN
4016  USW00014765  2015-12-31  39.0    35    50   8.05  18.1   1.0   NaN

      WT03  ...  WT11  WT13  WT14  WT15  WT16  WT17  WT18  WT19  WT21  WT22
0      NaN  ...   NaN   1.0   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
1      NaN  ...   NaN   NaN   NaN   NaN   1.0   NaN   1.0   NaN   NaN   NaN
2      NaN  ...   NaN   1.0   NaN   NaN   1.0   NaN   NaN   NaN   NaN   NaN
3      NaN  ...   NaN   1.0   1.0   NaN   1.0   NaN   NaN   NaN   NaN   NaN
4      NaN  ...   NaN   1.0   NaN   NaN   1.0   NaN   1.0   NaN   NaN   NaN
```

```
...    ...  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
4012   NaN  ...   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
4013   NaN  ...   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
4014   NaN  ...   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
4015   NaN  ...   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
4016   NaN  ...   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN

[4017 rows x 27 columns]
       county_name
count          0.0
mean           NaN
std            NaN
min            NaN
25%            NaN
50%            NaN
75%            NaN
max            NaN
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91741 entries, 0 to 91740
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   state               91741 non-null  object
 1   stop_date           91741 non-null  object
 2   stop_time           91741 non-null  object
 3   county_name         0 non-null      float64
 4   driver_gender       86536 non-null  object
 5   driver_race         86539 non-null  object
 6   violation_raw       86539 non-null  object
 7   violation           86539 non-null  object
 8   search_conducted    91741 non-null  bool
 9   search_type         3307 non-null   object
 10  stop_outcome        86539 non-null  object
 11  is_arrested         86539 non-null  object
 12  stop_duration       86539 non-null  object
 13  drugs_related_stop  91741 non-null  bool
 14  district            91741 non-null  object
```

```
dtypes: bool(2), float64(1), object(12)
memory usage: 9.3+ MB
              TAVG         TMIN         TMAX         AWND         WSF2  \
count  1217.000000  4017.000000  4017.000000  4017.000000  4017.000000
mean     52.493016    43.484441    61.268608     8.593707    19.274782
std      17.830714    17.020298    18.199517     3.364601     5.623866
min       6.000000    -5.000000    15.000000     0.220000     4.900000
25%      39.000000    30.000000    47.000000     6.260000    15.000000
50%      54.000000    44.000000    62.000000     8.050000    17.900000
75%      68.000000    58.000000    77.000000    10.290000    21.900000
max      86.000000    77.000000   102.000000    26.840000    48.100000

         WT01   WT02   WT03   WT04   WT05  ...  WT11    WT13   WT14  WT15  \
count  1767.0  221.0  224.0  117.0  360.0  ...   1.0  1175.0  575.0   6.0
mean      1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0
std       0.0    0.0    0.0    0.0    0.0  ...   NaN     0.0    0.0   0.0
min       1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0
25%       1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0
50%       1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0
75%       1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0
max       1.0    1.0    1.0    1.0    1.0  ...   1.0     1.0    1.0   1.0

         WT16  WT17   WT18  WT19  WT21  WT22
count  1326.0  12.0  345.0   4.0  18.0  32.0
mean      1.0   1.0    1.0   1.0   1.0   1.0
std       0.0   0.0    0.0   0.0   0.0   0.0
min       1.0   1.0    1.0   1.0   1.0   1.0
25%       1.0   1.0    1.0   1.0   1.0   1.0
50%       1.0   1.0    1.0   1.0   1.0   1.0
75%       1.0   1.0    1.0   1.0   1.0   1.0
max       1.0   1.0    1.0   1.0   1.0   1.0

[8 rows x 25 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4017 entries, 0 to 4016
Data columns (total 27 columns):
 #   Column   Non-Null Count  Dtype
```

```
---  ------   --------------  -----
 0   STATION  4017 non-null   object
 1   DATE     4017 non-null   object
 2   TAVG     1217 non-null   float64
 3   TMIN     4017 non-null   int64
 4   TMAX     4017 non-null   int64
 5   AWND     4017 non-null   float64
 6   WSF2     4017 non-null   float64
 7   WT01     1767 non-null   float64
 8   WT02     221 non-null    float64
 9   WT03     224 non-null    float64
 10  WT04     117 non-null    float64
 11  WT05     360 non-null    float64
 12  WT06     25 non-null     float64
 13  WT07     79 non-null     float64
 14  WT08     404 non-null    float64
 15  WT09     69 non-null     float64
 16  WT10     2 non-null      float64
 17  WT11     1 non-null      float64
 18  WT13     1175 non-null   float64
 19  WT14     575 non-null    float64
 20  WT15     6 non-null      float64
 21  WT16     1326 non-null   float64
 22  WT17     12 non-null     float64
 23  WT18     345 non-null    float64
 24  WT19     4 non-null      float64
 25  WT21     18 non-null     float64
 26  WT22     32 non-null     float64
dtypes: float64(23), int64(2), object(2)
memory usage: 847.5+ KB
```

Out[92]:
```
Zone X4    24279
Zone K3    20405
Zone K2    18397
Zone X3    17013
Zone K1     8678
Zone X1     2969
```

```
Name: district, dtype: int64
```

List of all county name

In [93]:
```
police['county_name'].value_counts()
```

Out[93]:
```
Series([], Name: county_name, dtype: int64)
```

As above There is no any county name display here

In [94]:
```
police['violation_raw'].value_counts()
```

Out[94]:
```
Speeding                            48424
Other Traffic Violation             16224
Equipment/Inspection Violation      10922
Registration Violation               3703
Seatbelt Violation                   2856
Special Detail/Directed Patrol       2467
Call for Service                     1392
Motorist Assist/Courtesy              205
Violation of City/Town Ordinance      181
APB                                    91
Suspicious Person                      56
Warrant                                18
Name: violation_raw, dtype: int64
```

Top most reason for violation is Speeding and least reson is Seat Belt.

In [95]:
```
police['violation'].value_counts()
```

Out[95]:
```
Speeding               48424
Moving violation       16224
Equipment              10922
Other                   4410
Registration/plates     3703
Seat belt               2856
Name: violation, dtype: int64
```

In [96]:
```
police['stop_outcome'].value_counts()
```

Out[96]:
```
Citation            77092
Warning              5137
Arrest Driver        2735
No Action             625
N/D                   607
Arrest Passenger      343
Name: stop_outcome, dtype: int64
```

We can see citation has high value.

In [97]:
```
police['driver_gender'].value_counts()
```

Out[97]:
```
M    62762
F    23774
Name: driver_gender, dtype: int64
```

Male is stopped than Female.

In [98]:
```
police['driver_race'].value_counts()
```

Out[98]:
```
White       61872
Black       12285
Hispanic     9727
Asian        2390
Other         265
Name: driver_race, dtype: int64
```

There's no bias here for pulling over non-white drivers vs white drivers but other race is less stopped by.

In [99]:
```
police[["stop_date", "stop_time"]].head()
```

Out[99]:

	stop_date	stop_time
0	2005-01-04	12:55
1	2005-01-23	23:15
2	2005-02-17	04:15
3	2005-02-20	17:15
4	2005-02-24	01:20

In [100]:
```
police["is_arrested"] = police.is_arrested.astype("bool")
dt_index = police.stop_date.str.cat(police.stop_time, sep = " ")
police["stop_datetime"] = pd.to_datetime(dt_index)

police_cleaned = (police.
  drop(["county_name", "state"], axis = "columns").
  dropna(subset = ["driver_gender", "driver_race"]).
  set_index("stop_datetime")
)

police_cleaned.head()
```

Out[100]:

stop_datetime	stop_date	stop_time	driver_gender	driver_race	violation_raw	violation	search_conducted	search_type	st
2005-01-04 12:55:00	2005-01-04	12:55	M	White	Equipment/Inspection Violation	Equipment	False	NaN	
2005-01-23 23:15:00	2005-01-23	23:15	M	White	Speeding	Speeding	False	NaN	
2005-02-17 04:15:00	2005-02-17	04:15	M	White	Speeding	Speeding	False	NaN	
2005-02-20 17:15:00	2005-02-20	17:15	M	White	Call for Service	Other	False	NaN	
2005-02-24 01:20:00	2005-02-24	01:20	F	White	Speeding	Speeding	False	NaN	

In [101]:
```
police_cleaned.search_conducted.head()
```

Out[101]:
```
stop_datetime
2005-01-04 12:55:00    False
2005-01-23 23:15:00    False
2005-02-17 04:15:00    False
2005-02-20 17:15:00    False
2005-02-24 01:20:00    False
Name: search_conducted, dtype: bool
```

We can break the average search rate by gender

In [102]:
```
police_cleaned.groupby("driver_gender").search_conducted.mean()
```

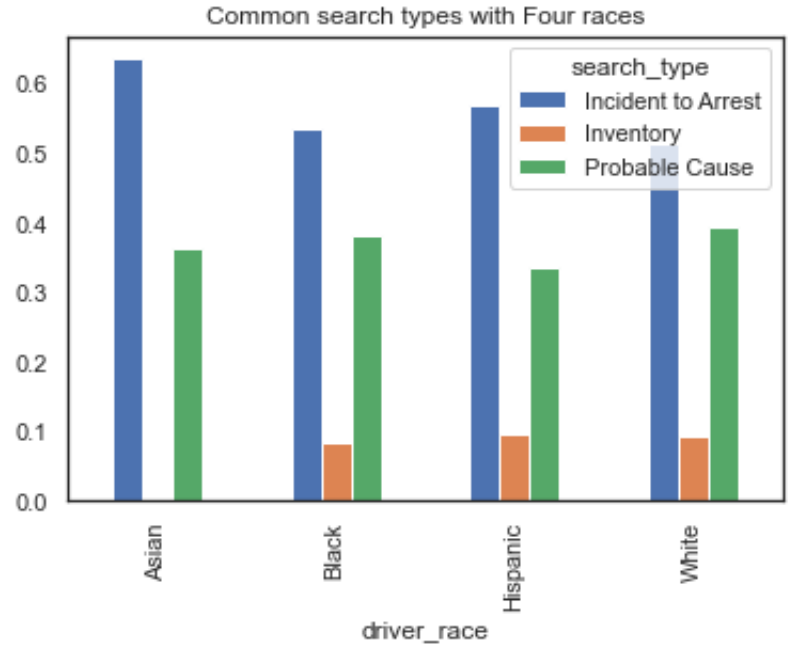
Out[102]:
```
driver_gender
F    0.019181
M    0.045426
Name: search_conducted, dtype: float64
```

In [103]:
```
police_cleaned.groupby(["violation", "driver_gender"]).search_conducted.mean()
```

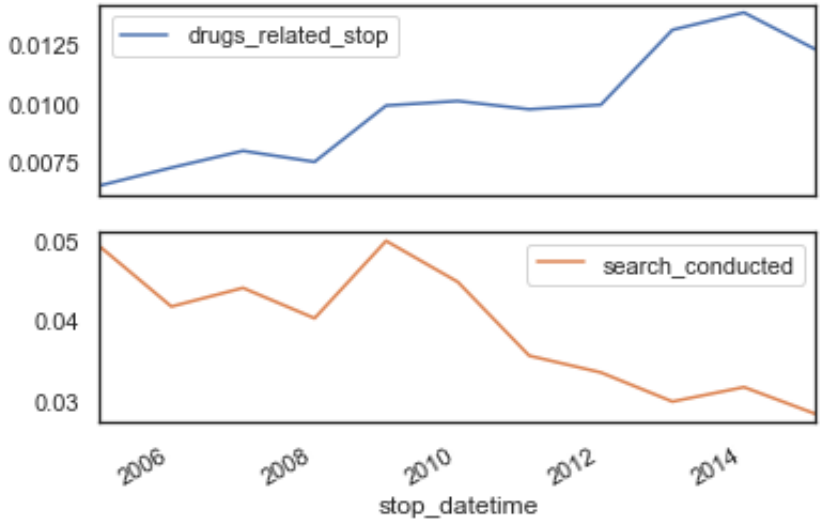
Out[103]:
```
violation            driver_gender
Equipment            F                0.039984
                     M                0.071496
Moving violation     F                0.039257
                     M                0.061524
Other                F                0.041018
                     M                0.046191
Registration/plates  F                0.054924
                     M                0.108802
Seat belt            F                0.017301
                     M                0.035119
Speeding             F                0.008309
                     M                0.027885
Name: search_conducted, dtype: float64
```

In [104]:
```
condition = police_cleaned.driver_race.isin(["White", "Black", "Hispanic", "Asian"]) & police_cle

search_type_by_race = (police_cleaned[condition].
  groupby("driver_race").
  search_type.
  value_counts(normalize = True).
  unstack()
)
```

In [105]:
```
import matplotlib.pyplot as plt
search_type_by_race.plot(kind = "bar")
plt.title("Common search types with Four races")
plt.show()
```

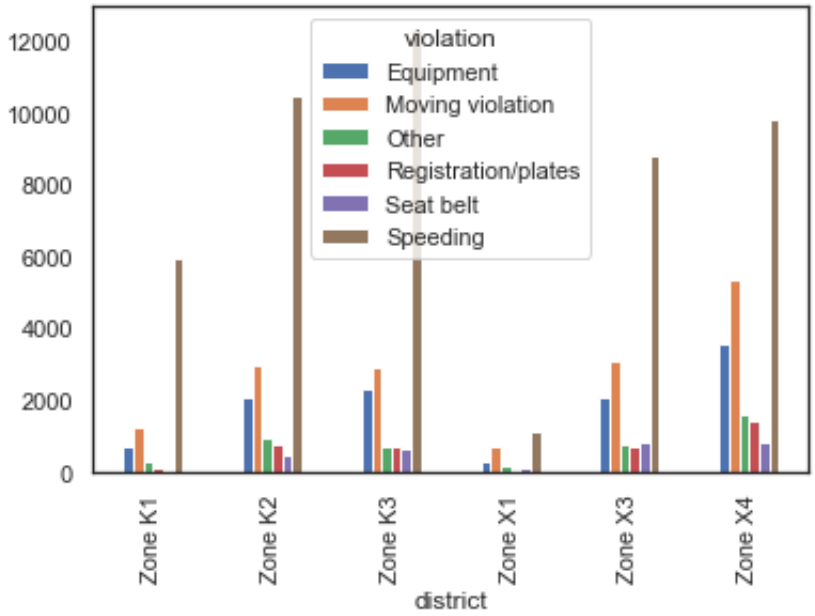


In [106]:
```
drug_rate = police_cleaned.drugs_related_stop.resample("A").mean()
search_rate = police_cleaned.search_conducted.resample("A").mean()
annual = pd.concat([drug_rate, search_rate], axis="columns")
annual.plot(subplots = True)
plt.show()
```



Distributation all the violation

In [107]:
```
all_zone = pd.crosstab(police_cleaned.district, police_cleaned.violation)
all_zone.plot(kind = 'bar')
plt.show()
```



Let's create function

In [108]:
```
def outcome_stats(df):
    total = len(df)
    warnings = len(df[df['stop_outcome'] == 'Warning'])
    citations = len(df[df['stop_outcome'] == 'Citation'])
    arrests = len(df[df['stop_outcome'] == 'Arrest Driver'])
    citations_per_warning = citations / warnings
    arrest_rate = arrests / total

    return(pd.Series(data = {
        'total': total,
        'warnings': warnings,
        'citations': citations,
        'arrests': arrests,
        'citations per warning': citations_per_warning,
        'arrest rate': arrest_rate
    }))
```

In [109]:
```
outcome_stats(police)
```

Out[109]:
```
total                   91741.000000
warnings                 5137.000000
citations               77092.000000
arrests                  2735.000000
citations per warning      15.007203
arrest rate                 0.029812
dtype: float64
```

```
Now we can use our function with Gender

# Comparing stop outcomes by gender
```

In [110]:
```
police.groupby('driver_gender').apply(outcome_stats)
```

Out[110]:

driver_gender	total	warnings	citations	arrests	citations per warning	arrest rate
F	23774.0	1485.0	21251.0	556.0	14.310438	0.023387
M	62762.0	3651.0	55840.0	2179.0	15.294440	0.034718

```
# Comparing stop outcomes by Race
```

In [111]:
```
police.groupby('driver_race').apply(compute_outcome_stats)
```

Out[111]:

driver_race	total	warnings	citations	arrests	citations per warning	arrest rate
Asian	2390.0	108.0	2206.0	42.0	20.425926	0.017573
Black	12285.0	800.0	10531.0	667.0	13.163750	0.054294
Hispanic	9727.0	656.0	8288.0	538.0	12.634146	0.055310
Other	265.0	14.0	244.0	2.0	17.428571	0.007547
White	61872.0	3559.0	55823.0	1486.0	15.685024	0.024017

In [112]:
```
police.groupby(['violation']).apply(compute_outcome_stats)
```

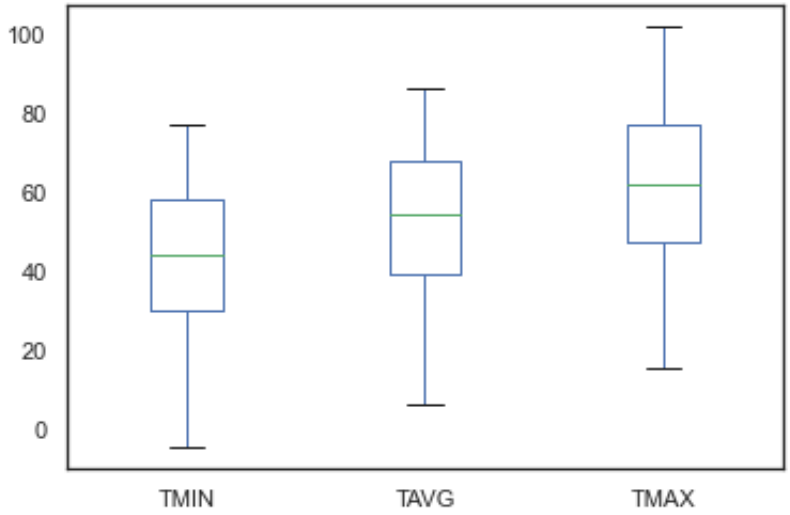
Out[112]:

violation	total	warnings	citations	arrests	citations per warning	arrest rate
Equipment	10922.0	1484.0	8220.0	553.0	5.539084	0.050632
Moving violation	16224.0	1248.0	13923.0	850.0	11.156250	0.052392
Other	4410.0	50.0	3536.0	341.0	70.720000	0.077324
Registration/plates	3703.0	186.0	3138.0	316.0	16.870968	0.085336
Seat belt	2856.0	356.0	2415.0	64.0	6.783708	0.022409
Speeding	48424.0	1813.0	45860.0	611.0	25.295091	0.012618

#Analyzing the effect of weather on policing

In [113]:
```
# Describe the temperature columns
print(weather[['TMIN', 'TAVG', 'TMAX']].describe())

# Create a box plot of the temperature columns
weather[['TMIN', 'TAVG', 'TMAX']].plot(kind = 'box')

# Display the plot
plt.show()
```

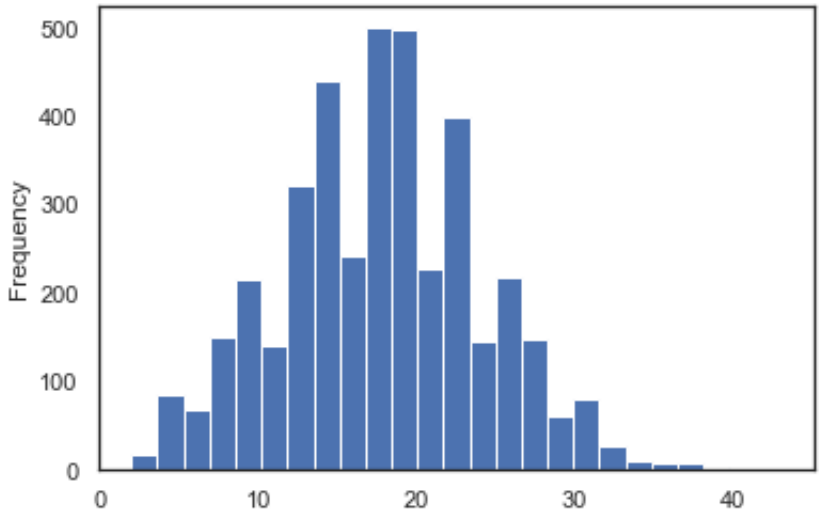
```
              TMIN         TAVG         TMAX
count  4017.000000  1217.000000  4017.000000
mean     43.484441    52.493016    61.268608
std      17.020298    17.830714    18.199517
min      -5.000000     6.000000    15.000000
25%      30.000000    39.000000    47.000000
50%      44.000000    54.000000    62.000000
75%      58.000000    68.000000    77.000000
max      77.000000    86.000000   102.000000
```

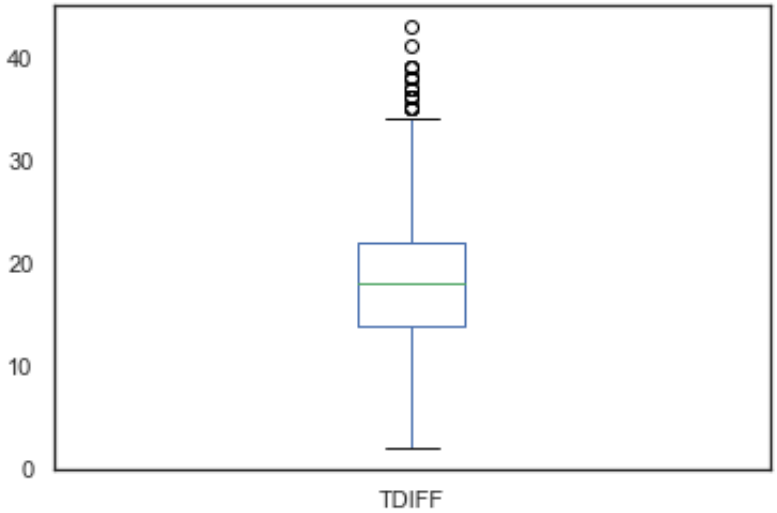


In [114]:
```
# Create a 'TDIFF' column that represents temperature difference
weather['TDIFF'] = weather.TMAX - weather.TMIN
```

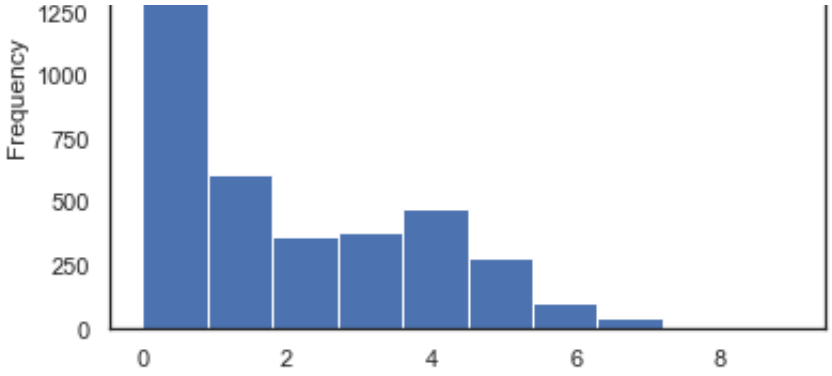
```
# Describe the 'TDIFF' column
print(weather.TDIFF.describe())

# Create a histogram with 20 bins to visualize 'TDIFF'
weather.TDIFF.plot(kind = 'hist', bins = 25)

# Create a box plot of the temperature columns
weather[['TDIFF']].plot(kind = 'box')

# Display the plot
plt.show()
```

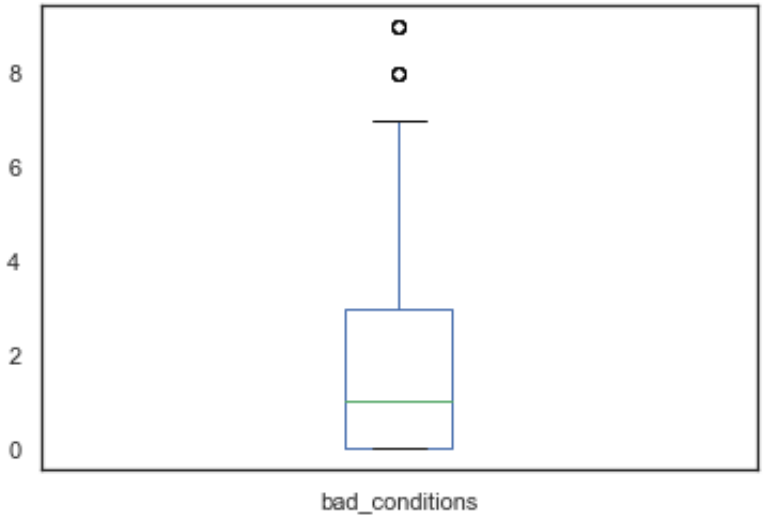
```
count    4017.000000
mean       17.784167
std         6.350720
min         2.000000
25%        14.000000
50%        18.000000
75%        22.000000
max        43.000000
Name: TDIFF, dtype: float64
```





In [115]:
```
# Copy 'WT01' through 'WT22' to a new DataFrame
WTTest = weather.loc[:, 'WT01' : 'WT22']

weather['bad_conditions'] = WTTest.sum(axis = 'columns')

# Replace the missing values in 'bad_conditions' with 0
weather['bad_conditions'] = weather.bad_conditions.fillna(0).astype('int')

# Create a histogram to visualize 'bad_conditions'
weather.bad_conditions.plot(kind = 'hist')

# Create a box plot of the bad conditions columns
weather[['bad_conditions']].plot(kind = 'box')

plt.show()
```


Frequency
1250
1000
750
500
250
0
0
2
4
6
8


8
6
4
2
0
bad_conditions

In [116]:
```
print(weather.bad_conditions.value_counts().sort_index())

# Create a dictionary that maps integers to strings
mappings = {0:'good', 1:'bad', 2:'bad', 3:'bad', 4:'bad', 5:'worse', 6:'worse', 7:'worse', 8:'wo

# Convert the bad_conditions integers to string
weather['rating'] = weather.bad_conditions.map(mappings)

# Count the unique values in rating
print(weather.rating.value_counts())
```

```
0    1749
1     613
2     367
3     380
4     476
5     282
6     101
7      41
8       4
9       4
Name: bad_conditions, dtype: int64
bad      1836
good     1749
worse     428
Name: rating, dtype: int64
```