02_eda

March 10, 2024

1 Exploratory Data Analysis

2 Import Modules

[231]:
```
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use("bmh")

import seaborn as sns
```

3 Import Cleaned Datasets

[232]:
```
train_features = pd.read_csv('../data/clean/train_features.csv')
train_labels = pd.read_csv('../data/clean/train_labels.csv')
test_features = pd.read_csv('../data/clean/test_features.csv')
```

4 Data Types

[233]:
```
train_features
```

[233]:

	city	year	weekofyear	week_start_date	ndvi_ne	ndvi_nw	\
0	sj	1990	18	1990-04-30 00:00:00	0.122600	0.103725	
1	sj	1990	19	1990-05-07 00:00:00	0.169900	0.142175	
2	sj	1990	20	1990-05-14 00:00:00	0.032250	0.172967	
3	sj	1990	21	1990-05-21 00:00:00	0.128633	0.245067	
4	sj	1990	22	1990-05-28 00:00:00	0.196200	0.262200	
…	…	…	…	…	…	…	
1451	iq	2010	21	2010-05-28 00:00:00	0.342750	0.318900	
1452	iq	2010	22	2010-06-04 00:00:00	0.160157	0.160371	
1453	iq	2010	23	2010-06-11 00:00:00	0.247057	0.146057	
1454	iq	2010	24	2010-06-18 00:00:00	0.333914	0.245771	
1455	iq	2010	25	2010-06-25 00:00:00	0.298186	0.232971	

	ndvi_se	ndvi_sw	precipitation_amt_mm	reanalysis_air_temp_k	…	\

```
0     0.198483  0.177617                12.42             297.572857  …
1     0.162357  0.155486                22.82             298.211429  …
2     0.157200  0.170843                34.54             298.781429  …
3     0.227557  0.235886                15.36             298.987143  …
4     0.251200  0.247340                 7.52             299.518571  …
…        …         …                     …                   …   …
1451  0.256343  0.292514                55.30             299.334286  …
1452  0.136043  0.225657                86.47             298.330000  …
1453  0.250357  0.233714                58.94             296.598571  …
1454  0.278886  0.325486                59.67             296.345714  …
1455  0.274214  0.315757                63.22             298.097143  …

      reanalysis_precip_amt_kg_per_m2  reanalysis_relative_humidity_percent  \
0                               32.00                             73.365714
1                               17.94                             77.368571
2                               26.10                             82.052857
3                               13.90                             80.337143
4                               12.20                             80.460000
…                                 …                                 …
1451                            45.00                             88.765714
1452                           207.10                             91.600000
1453                            50.60                             94.280000
1454                            62.33                             94.660000
1455                            36.90                             89.082857

      reanalysis_sat_precip_amt_mm  reanalysis_specific_humidity_g_per_kg  \
0                            12.42                              14.012857
1                            22.82                              15.372857
2                            34.54                              16.848571
3                            15.36                              16.672857
4                             7.52                              17.210000
…                                …                                 …
1451                         55.30                              18.485714
1452                         86.47                              18.070000
1453                         58.94                              17.008571
1454                         59.67                              16.815714
1455                         63.22                              17.355714

      reanalysis_tdtr_k  station_avg_temp_c  station_diur_temp_rng_c  \
0              2.628571           25.442857                 6.900000
1              2.371429           26.714286                 6.371429
2              2.300000           26.714286                 6.485714
3              2.428571           27.471429                 6.771429
4              3.014286           28.942857                 9.371429
…                 …                  …                      …
1451           9.800000           28.633333                11.933333
1452           7.471429           27.433333                10.500000
```

```
1453           7.500000           24.400000                  6.900000
1454           7.871429           25.433333                  8.733333
1455          11.014286           27.475000                  9.900000

      station_max_temp_c  station_min_temp_c  station_precip_mm
0                   29.4                20.0               16.0
1                   31.7                22.2                8.6
2                   32.2                22.8               41.4
3                   33.3                23.3                4.0
4                   35.0                23.9                5.8
…                    …                   …                  …
1451                35.4                22.4               27.0
1452                34.7                21.7               36.6
1453                32.2                19.2                7.4
1454                31.2                21.0               16.0
1455                33.7                22.2               20.4

[1456 rows x 24 columns]
```

[234]:
```
train_features.dtypes
```

[234]:
```
city                                    object
year                                     int64
weekofyear                               int64
week_start_date                         object
ndvi_ne                                float64
ndvi_nw                                float64
ndvi_se                                float64
ndvi_sw                                float64
precipitation_amt_mm                   float64
reanalysis_air_temp_k                  float64
reanalysis_avg_temp_k                  float64
reanalysis_dew_point_temp_k            float64
reanalysis_max_air_temp_k              float64
reanalysis_min_air_temp_k              float64
reanalysis_precip_amt_kg_per_m2        float64
reanalysis_relative_humidity_percent   float64
reanalysis_sat_precip_amt_mm           float64
reanalysis_specific_humidity_g_per_kg  float64
reanalysis_tdtr_k                      float64
station_avg_temp_c                     float64
station_diur_temp_rng_c                float64
station_max_temp_c                     float64
station_min_temp_c                     float64
station_precip_mm                      float64
dtype: object
```

[235]:
```
train_labels
```

[235]:

	city	year	weekofyear	total_cases
0	sj	1990	18	4
1	sj	1990	19	5
2	sj	1990	20	4
3	sj	1990	21	3
4	sj	1990	22	6
…	…	…	…	…
1451	iq	2010	21	5
1452	iq	2010	22	8
1453	iq	2010	23	1
1454	iq	2010	24	1
1455	iq	2010	25	4

[1456 rows x 4 columns]

[236]:
```
train_labels.dtypes
```

[236]:
```
city           object
year            int64
weekofyear      int64
total_cases     int64
dtype: object
```

5 Summary of Data Types

6 Features

6.1 Categorical Data

6.1.1 train_features

```
city           object
```

6.2 Timestamp (datetime64[ns])

6.2.1 train_features

```
week_start_date     object
```

6.3 Numeric Data

6.3.1 train_features

```
year                                  int64
weekofyear                            int64

ndvi_ne                             float64
ndvi_nw                             float64
```

```
ndvi_se                                 float64
ndvi_sw                                 float64
precipitation_amt_mm                    float64
reanalysis_air_temp_k                   float64
reanalysis_avg_temp_k                   float64
reanalysis_dew_point_temp_k             float64
reanalysis_max_air_temp_k               float64
reanalysis_min_air_temp_k               float64
reanalysis_precip_amt_kg_per_m2         float64
reanalysis_relative_humidity_percent    float64
reanalysis_sat_precip_amt_mm            float64
reanalysis_specific_humidity_g_per_kg   float64
reanalysis_tdtr_k                       float64
station_avg_temp_c                      float64
station_diur_temp_rng_c                 float64
station_max_temp_c                      float64
station_min_temp_c                      float64
station_precip_mm                       float64
```

7 Labels

7.1 Categorical Data

7.1.1 train_labels

```
city           object
```

7.2 Numeric Data

7.2.1 train_labels

```
year            int64
weekofyear      int64
total_cases     int64
```

[237]:
```
numeric_features = [
    'year',
    'weekofyear',
    'ndvi_ne',
    'ndvi_nw',
    'ndvi_se',
    'ndvi_sw',
    'precipitation_amt_mm',
    'reanalysis_air_temp_k',
    'reanalysis_avg_temp_k',
    'reanalysis_dew_point_temp_k',
    'reanalysis_max_air_temp_k',
    'reanalysis_min_air_temp_k',
    'reanalysis_precip_amt_kg_per_m2',
```

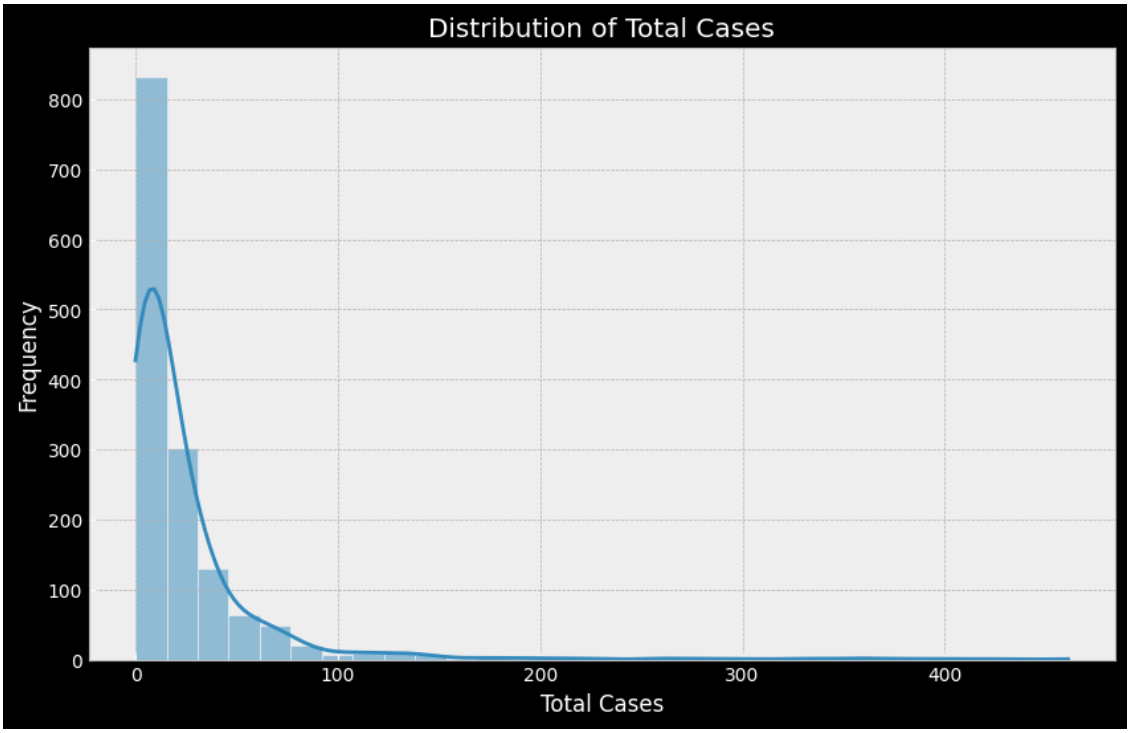
```
    'reanalysis_relative_humidity_percent',
    'reanalysis_sat_precip_amt_mm',
    'reanalysis_specific_humidity_g_per_kg',
    'reanalysis_tdtr_k',
    'station_avg_temp_c',
    'station_diur_temp_rng_c',
    'station_max_temp_c',
    'station_min_temp_c',
    'station_precip_mm'
]
```

8 Data Visualisation

9 Visualise Distribution of Target Variable

To check for class imbalance.

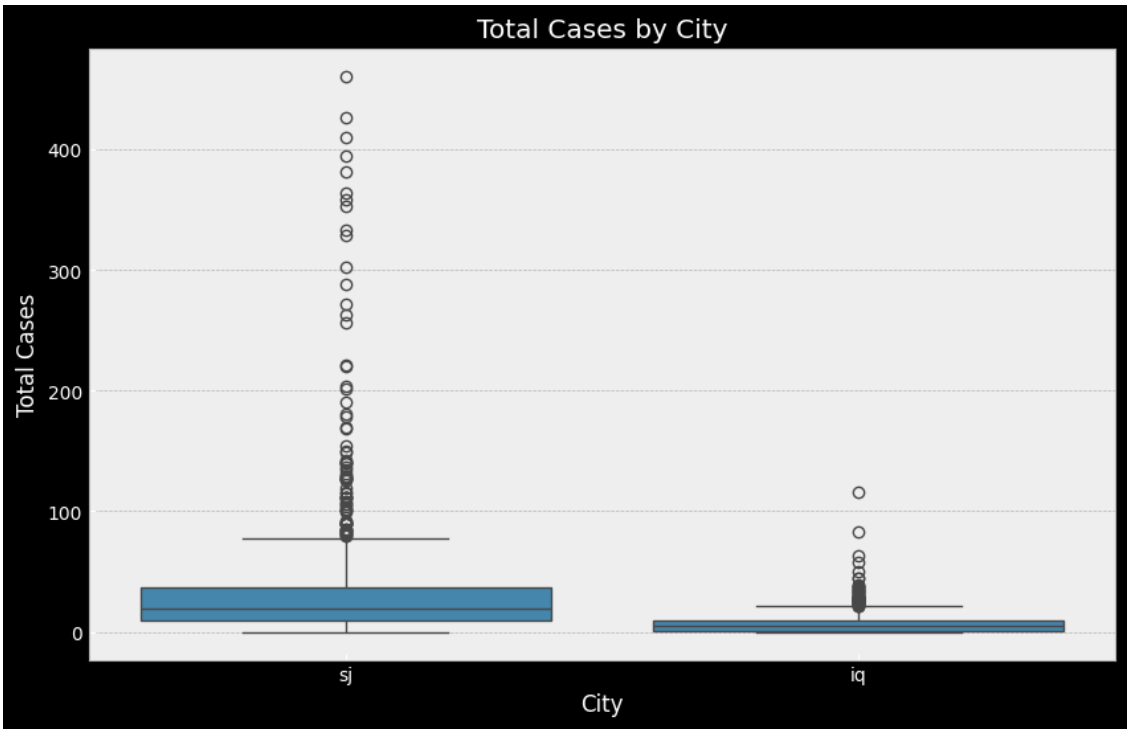
[238]:
```
plt.figure(figsize=(10, 6))
sns.histplot(train_labels['total_cases'], kde=True, bins=30)
plt.title('Distribution of Total Cases')
plt.xlabel('Total Cases')
plt.ylabel('Frequency')
plt.show()
```



10 Visualise Categorical Data

11 Categorical Features vs Target

[239]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='city', y='total_cases', data=pd.concat([train_features['city'],␣
 ↪train_labels['total_cases']], axis=1))
plt.title('Total Cases by City')
plt.xlabel('City')
plt.ylabel('Total Cases')
plt.show()
```

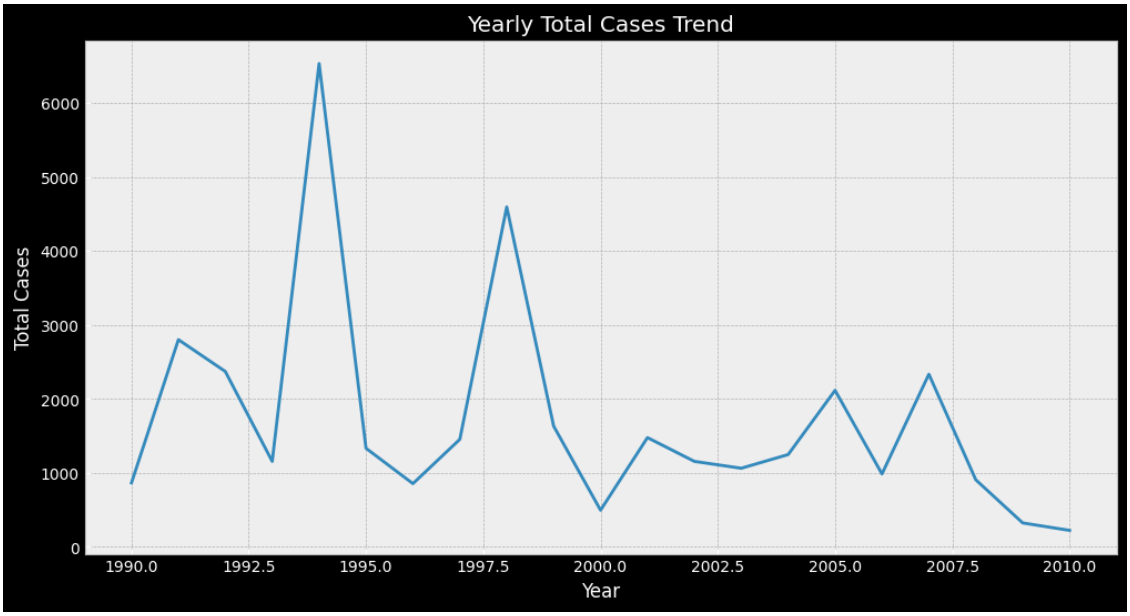


12 Time vs Target

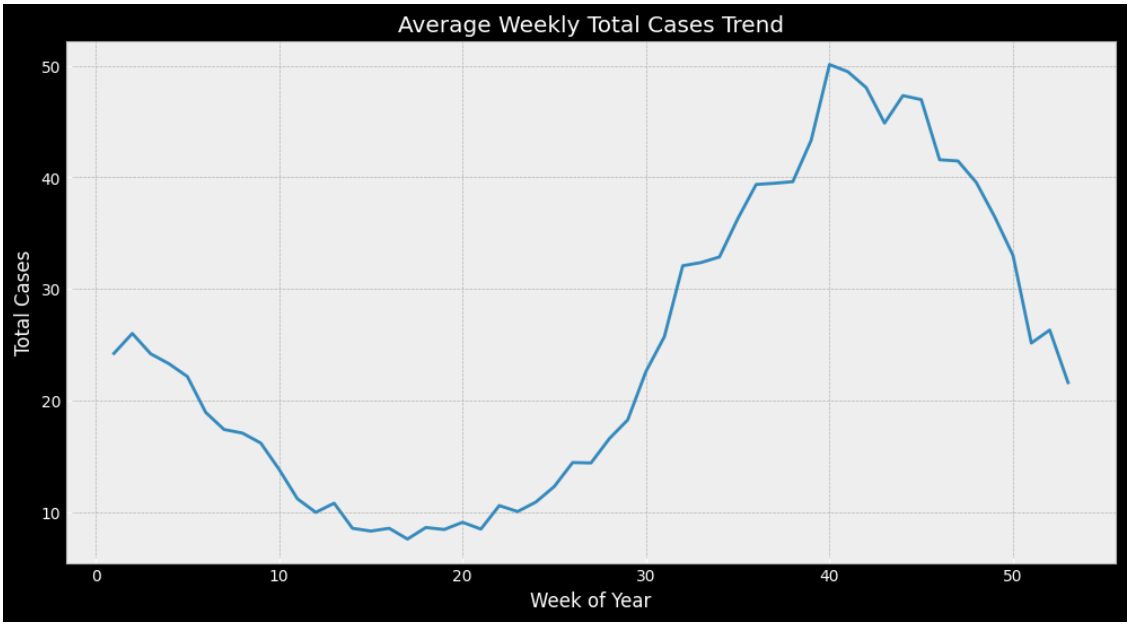
[240]:
```
# Yearly trend
plt.figure(figsize=(12, 6))
sns.lineplot(x='year', y='total_cases', estimator=sum, ci=None, data=pd.
 ↪concat([train_features[['year']], train_labels['total_cases']], axis=1))
plt.title('Yearly Total Cases Trend')
plt.xlabel('Year')
plt.ylabel('Total Cases')
plt.show()
```

```
# Weekly trend within a year (average across all years)
plt.figure(figsize=(12, 6))
sns.lineplot(x='weekofyear', y='total_cases', estimator=np.mean, ci=None,␣
 ↪data=pd.concat([train_features[['weekofyear']],␣
 ↪train_labels['total_cases']], axis=1))
plt.title('Average Weekly Total Cases Trend')
plt.xlabel('Week of Year')
plt.ylabel('Total Cases')
plt.show()
```

```
/tmp/ipykernel_111387/867156139.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(x='year', y='total_cases', estimator=sum, ci=None,
data=pd.concat([train_features[['year']], train_labels['total_cases']], axis=1))
```

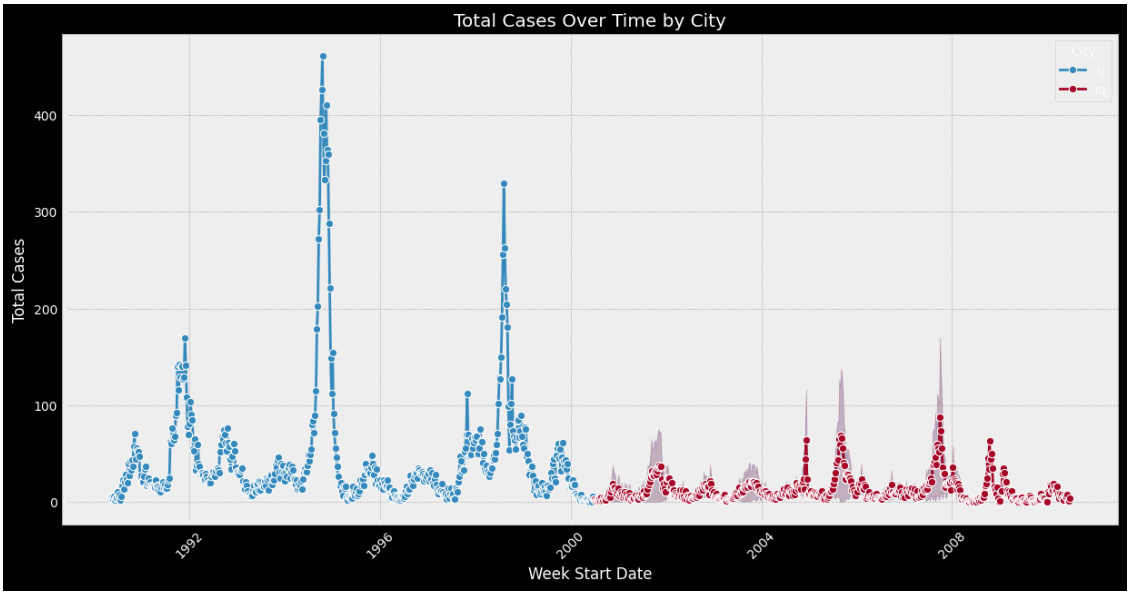



```
/tmp/ipykernel_111387/867156139.py:11: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(x='weekofyear', y='total_cases', estimator=np.mean, ci=None,
data=pd.concat([train_features[['weekofyear']], train_labels['total_cases']],
axis=1))
```



13 Timestamp (`week_start_date`) vs Target

[241]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Convert 'week_start_date' to datetime if it's not already
train_features['week_start_date'] = pd.
 ↪to_datetime(train_features['week_start_date'])

# Merge 'total_cases' into train_features for plotting
train_data_merged = train_features.merge(train_labels[['year', 'weekofyear',␣
 ↪'total_cases']], on=['year', 'weekofyear'], how='left')

# Plotting
plt.figure(figsize=(15, 7))
sns.lineplot(x='week_start_date', y='total_cases', data=train_data_merged,␣
 ↪hue='city', marker='o')
plt.title('Total Cases Over Time by City')
plt.xlabel('Week Start Date')
plt.ylabel('Total Cases')
plt.xticks(rotation=45)
plt.legend(title='City')
plt.savefig('../images/scatter-cases-time.png')
plt.show()
```

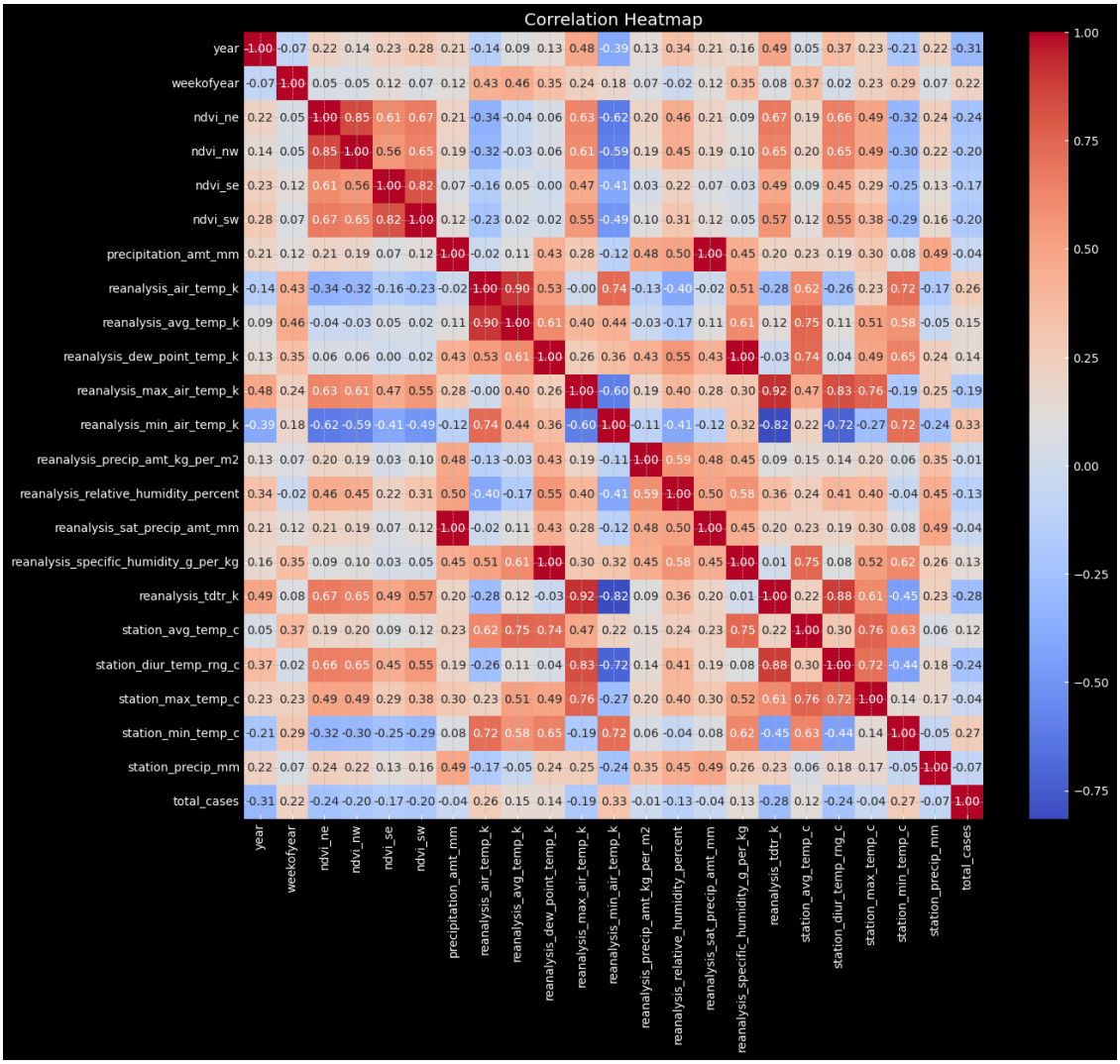


14 Numeric Features vs Target

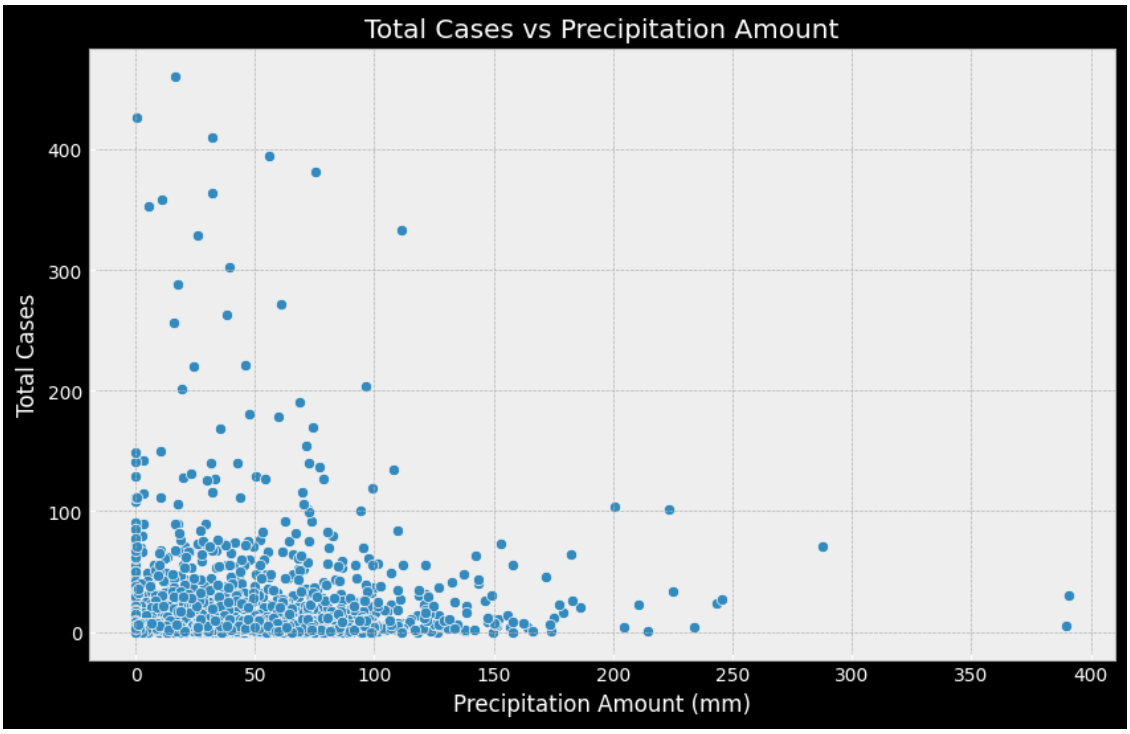
To interpret this heatmap, follow these steps:

- Locate the variable of interest (e.g., `total_cases`) on the x-axis or y-axis of the heatmap.
- Scan across the row or down the column to see how total_cases correlates with other variables.
- Variables with a strong blue color in the `total_cases` row/column have a positive correlation, meaning as one increases, the other tends to increase as well.
- Variables with a strong red color in the `total_cases` row/column have a negative correlation, meaning as one increases, the other tends to decrease.
- Lighter colors closer to white suggest a weak or no linear correlation.

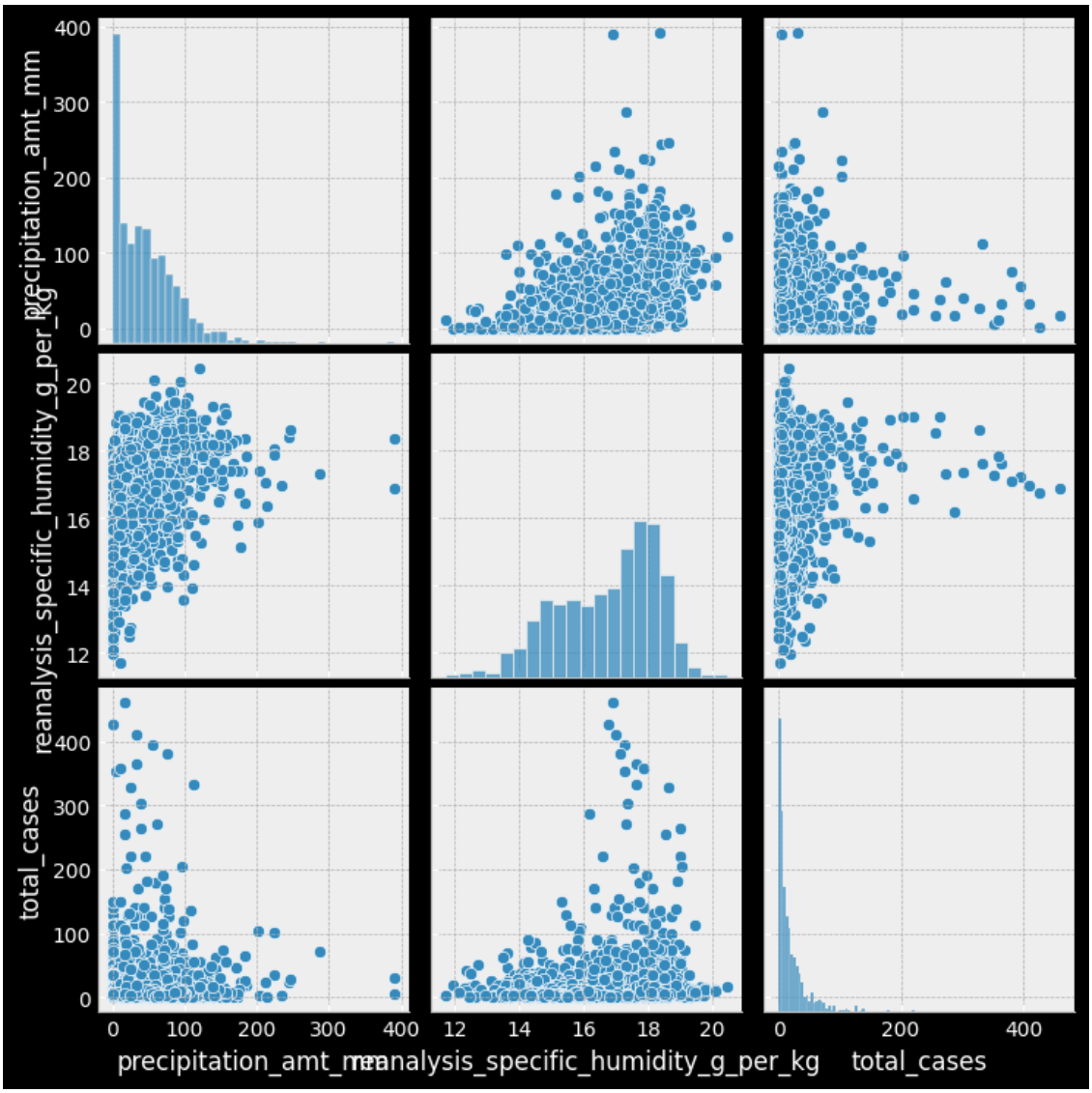
[242]:
```
# Correlation matrix
correlation_data = pd.concat([train_features[numeric_features],␣
 ↪train_labels['total_cases']], axis=1)
corr_matrix = correlation_data.corr()

plt.figure(figsize=(14, 12))
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[243]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='precipitation_amt_mm', y='total_cases', data=pd.
 ↪concat([train_features['precipitation_amt_mm'],␣
 ↪train_labels['total_cases']], axis=1))
plt.title('Total Cases vs Precipitation Amount')
plt.xlabel('Precipitation Amount (mm)')
plt.ylabel('Total Cases')
plt.show()
```



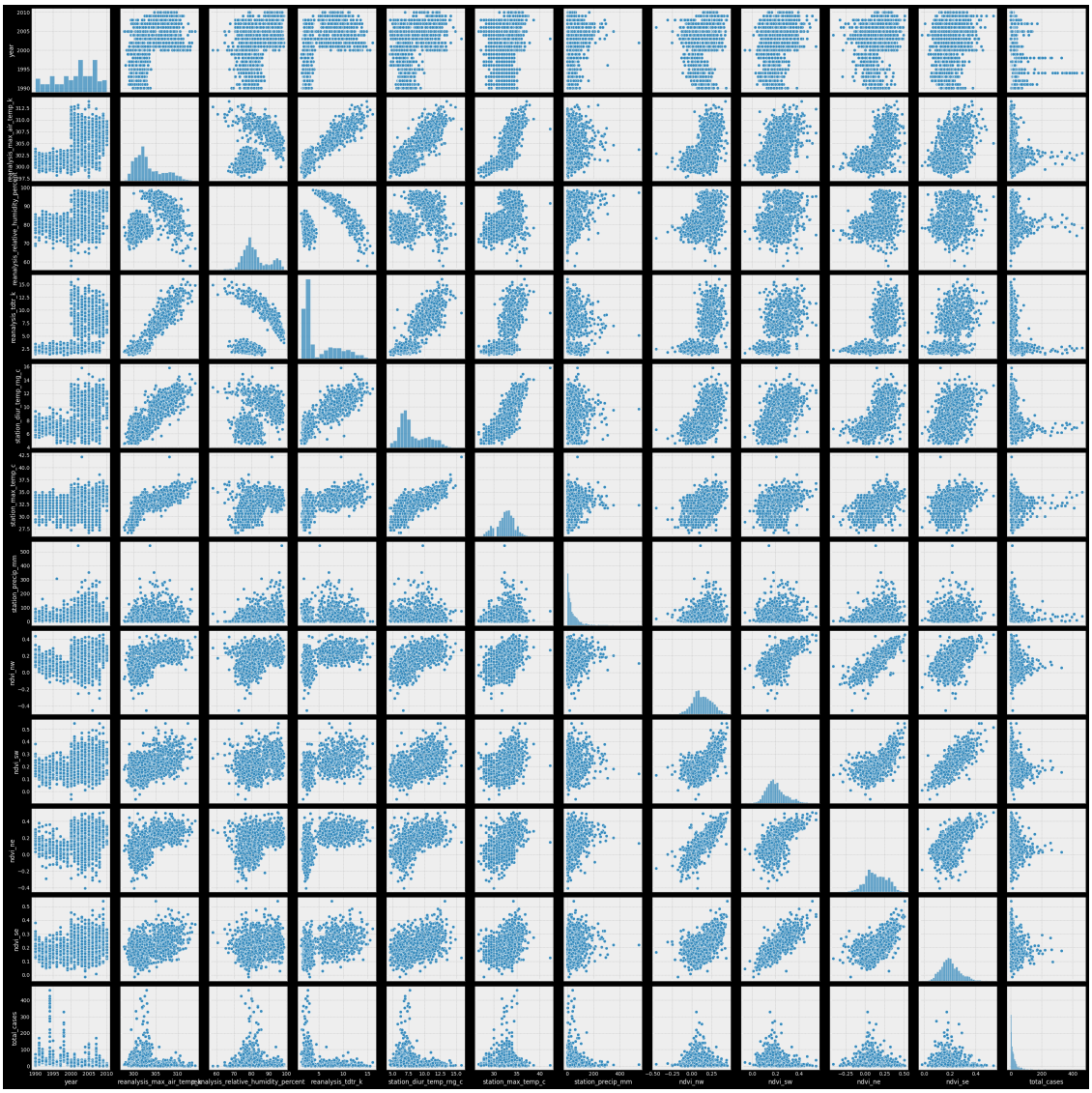
[244]:
```
# Select a subset of highly correlated features for pairwise plots
high_corr_features = ['precipitation_amt_mm',␣
 ↪'reanalysis_specific_humidity_g_per_kg', 'total_cases']  # Example features
sns.pairplot(pd.concat([train_features[high_corr_features[:-1]],␣
 ↪train_labels[high_corr_features[-1:]]], axis=1))
plt.show()
```



[245]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# List of features with moderate blue correlations with 'total_cases'
features_with_moderate_correlation = [
    'year',
    'reanalysis_max_air_temp_k',
    'reanalysis_relative_humidity_percent',
    'reanalysis_tdtr_k',
    'station_diur_temp_rng_c',
    'station_max_temp_c',
    'station_precip_mm',
    'ndvi_nw',
```

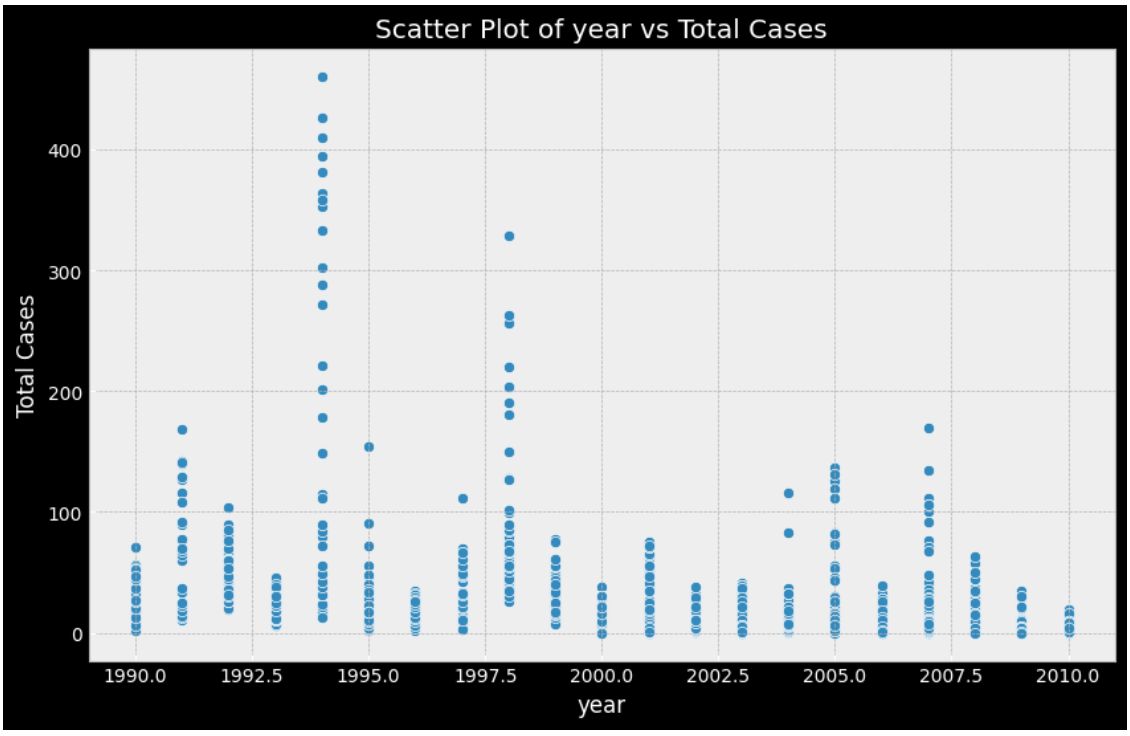
```
    'ndvi_sw',
    'ndvi_ne',
    'ndvi_se',
]

# Assuming 'train_features' is your DataFrame containing the feature variables
# and 'train_labels' is your DataFrame containing the 'total_cases' variable
# Combining the features with 'total_cases' into a single DataFrame for pairplot
plot_data = train_features[features_with_moderate_correlation]
plot_data['total_cases'] = train_labels['total_cases']  # Make sure indices␣
 ↪match!

# Creating a pairplot
sns.pairplot(plot_data)
plt.show()
```

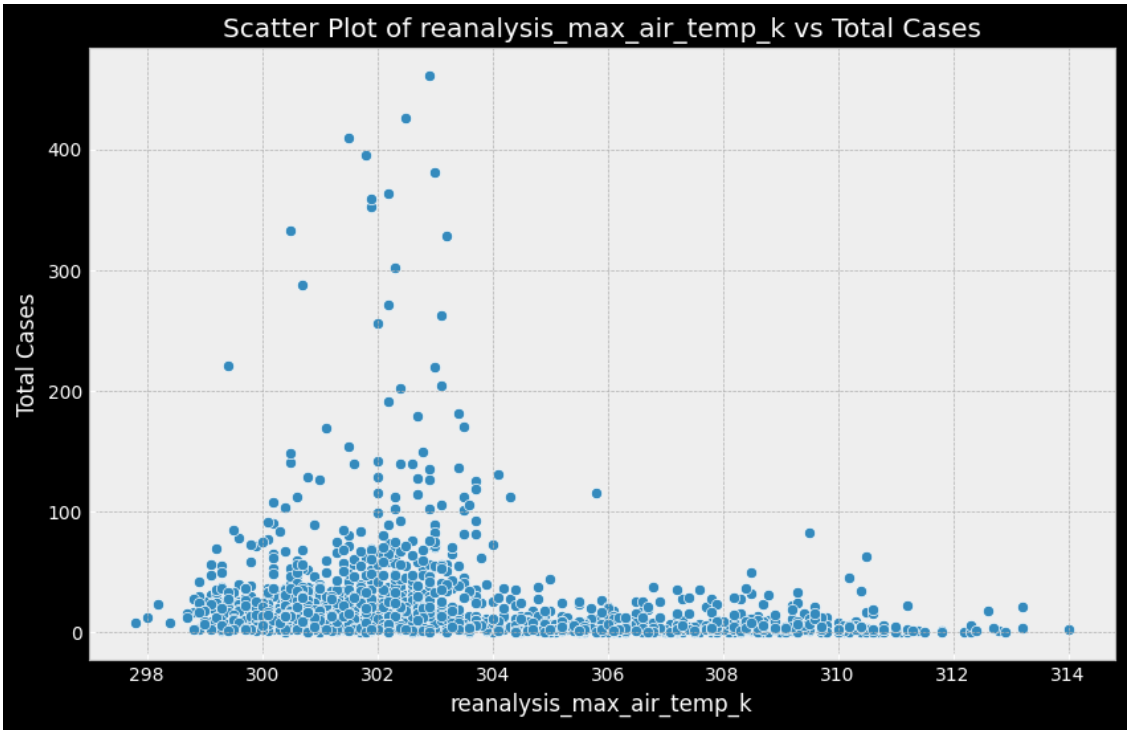
```
/tmp/ipykernel_111387/3222927614.py:23: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  plot_data['total_cases'] = train_labels['total_cases']  # Make sure indices
match!
```

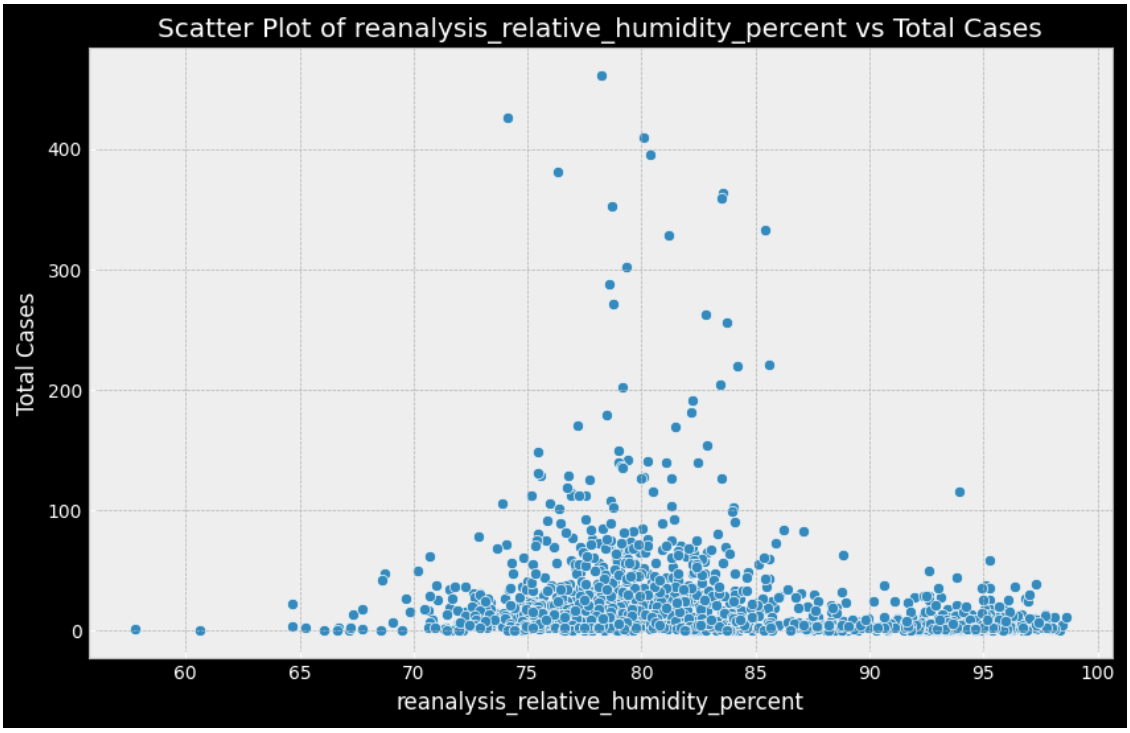


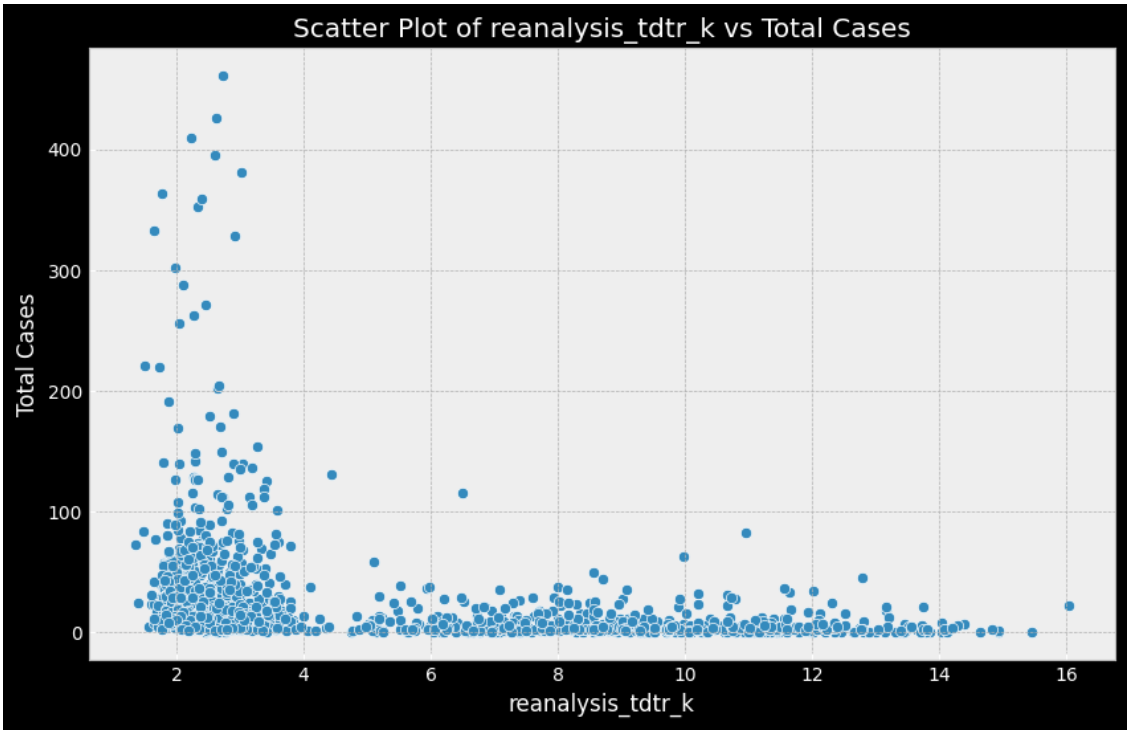
[246]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# List of features with moderate blue correlations with 'total_cases'
features_with_moderate_correlation = [
    'year',
    'reanalysis_max_air_temp_k',
    'reanalysis_relative_humidity_percent',
    'reanalysis_tdtr_k',
    'station_diur_temp_rng_c',
    'station_max_temp_c',
    'station_precip_mm',
    'ndvi_nw',
```

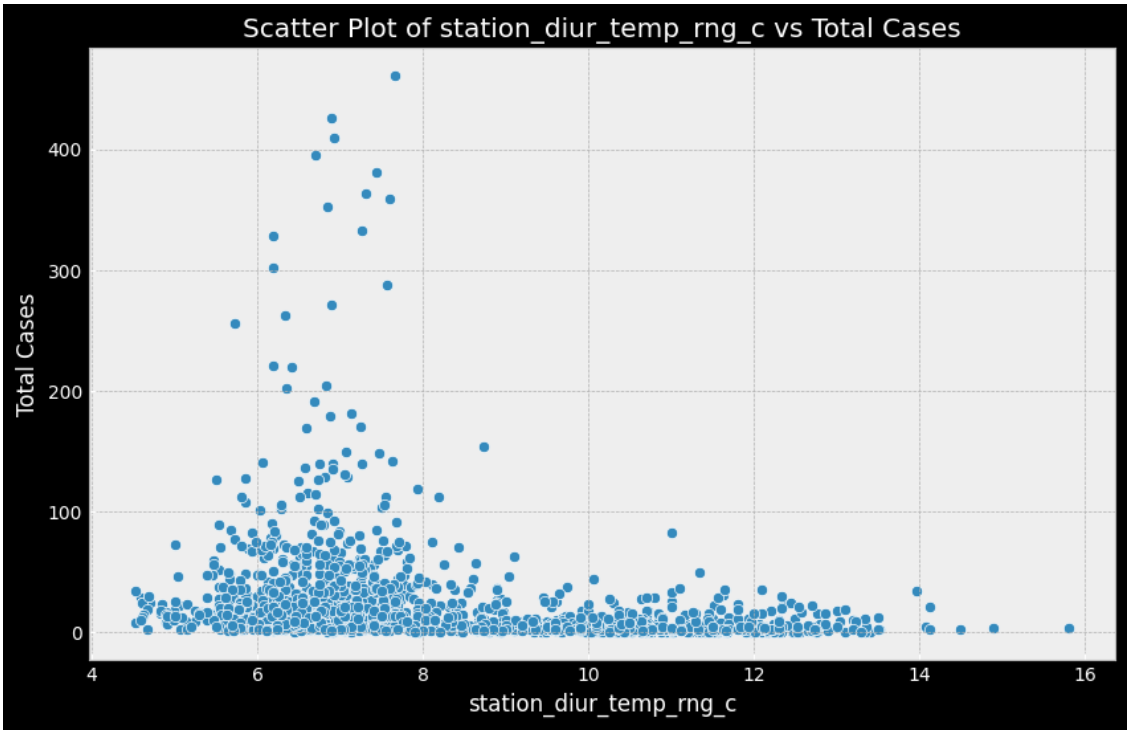
```
    'ndvi_sw',
    'ndvi_ne',
    'ndvi_se',
]

# Assuming you have a DataFrame 'train_features' containing your feature␣
 ↪variables
# and a DataFrame 'train_labels' containing the 'total_cases' variable

# Loop over the features and create scatterplots
for feature in features_with_moderate_correlation:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=train_features[feature], y=train_labels['total_cases'])
    plt.title(f'Scatter Plot of {feature} vs Total Cases')
    plt.xlabel(feature)
    plt.ylabel('Total Cases')
    plt.show()
```

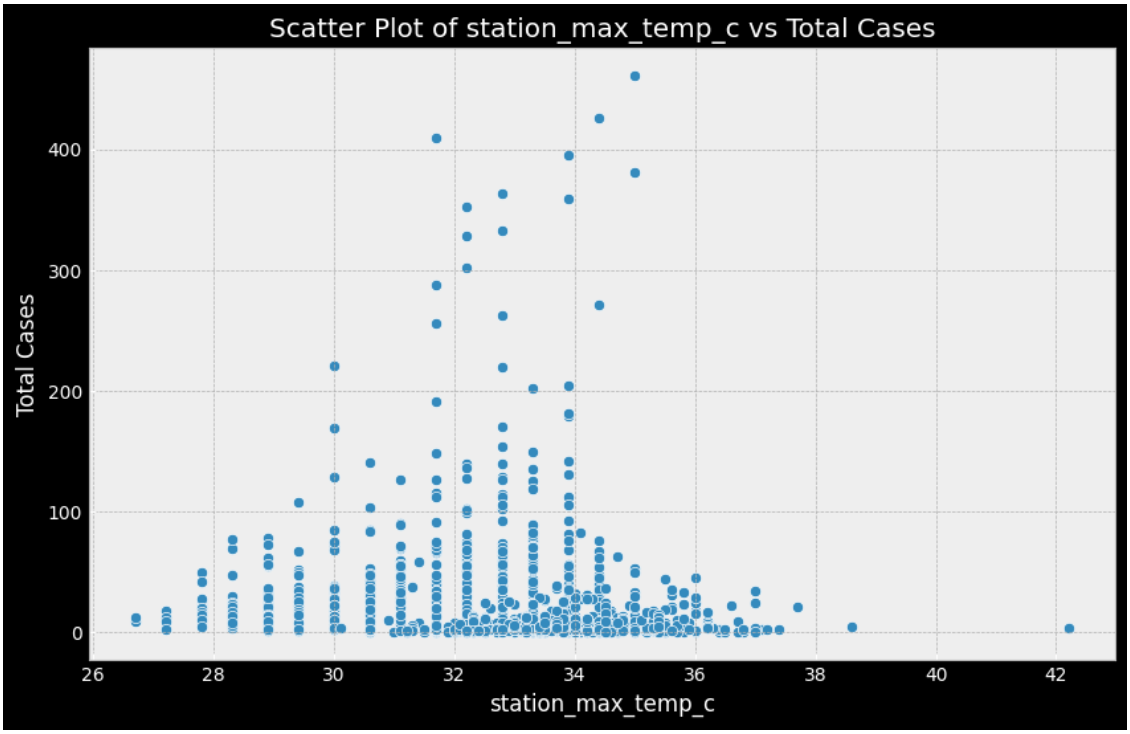


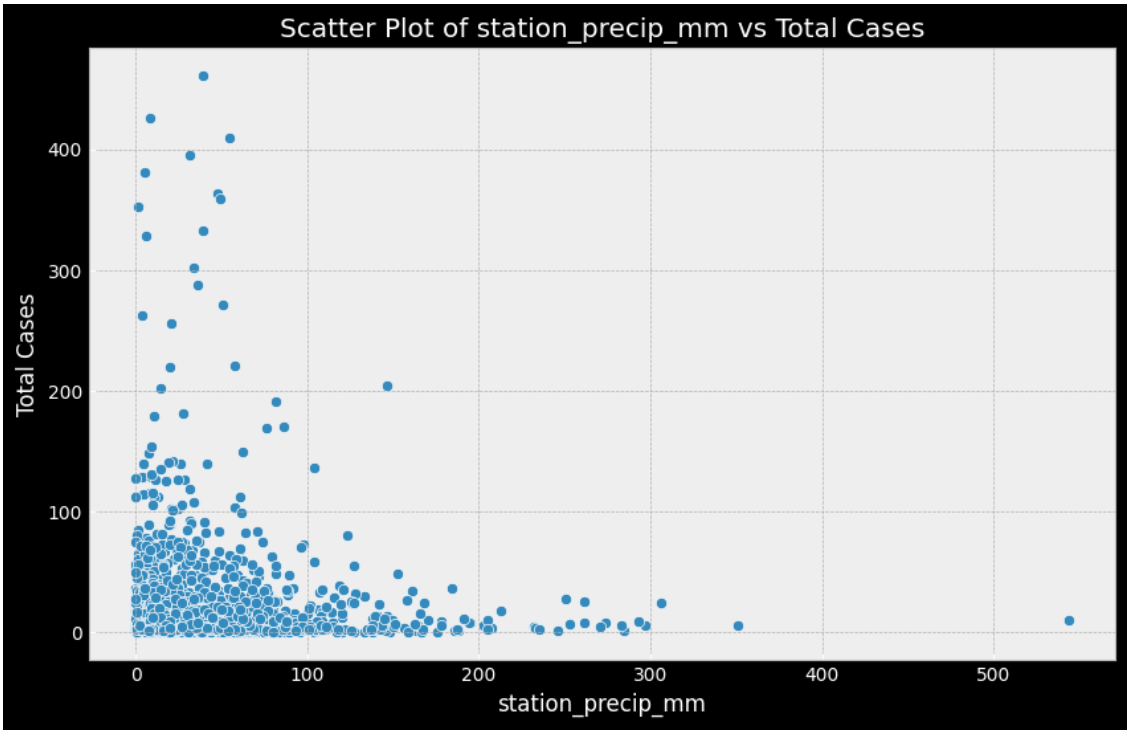

Scatter Plot of reanalysis_max_air_temp_k vs Total Cases
Total Cases
0
100
200
300
400
298
300
302
304
306
308
310
312
314
reanalysis_max_air_temp_k

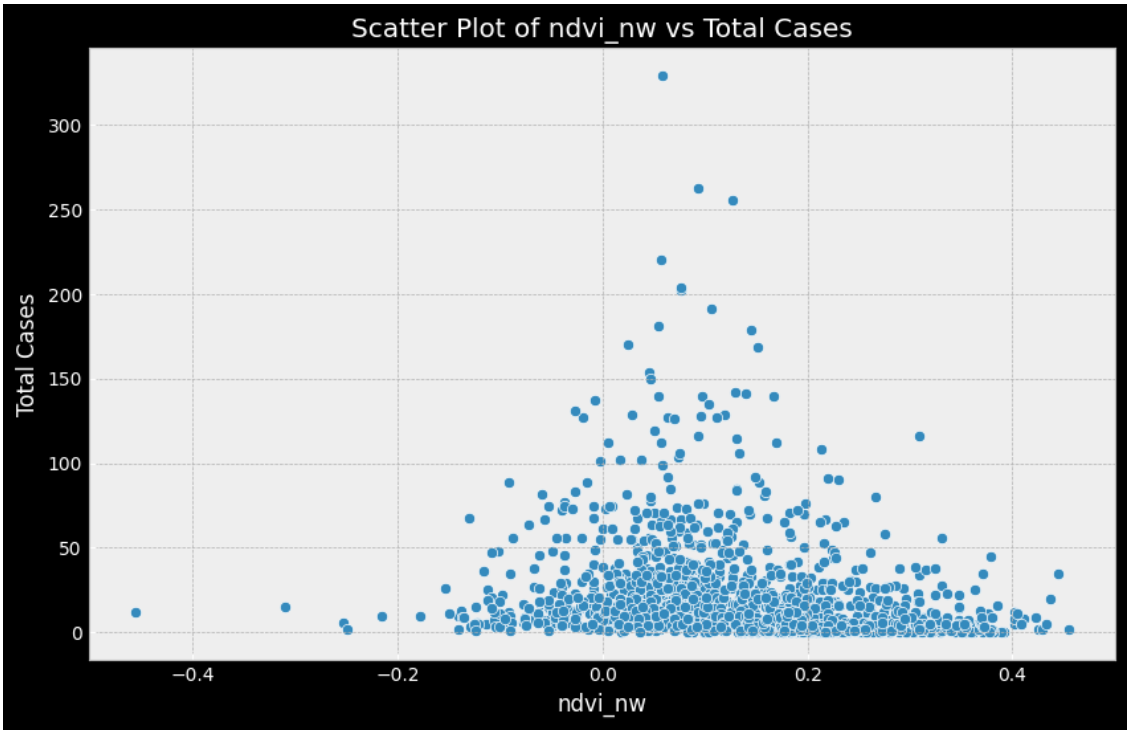

Scatter Plot of reanalysis_relative_humidity_percent vs Total Cases
Total Cases
0
100
200
300
400
60
65
70
75
80
85
90
95
100
reanalysis_relative_humidity_percent

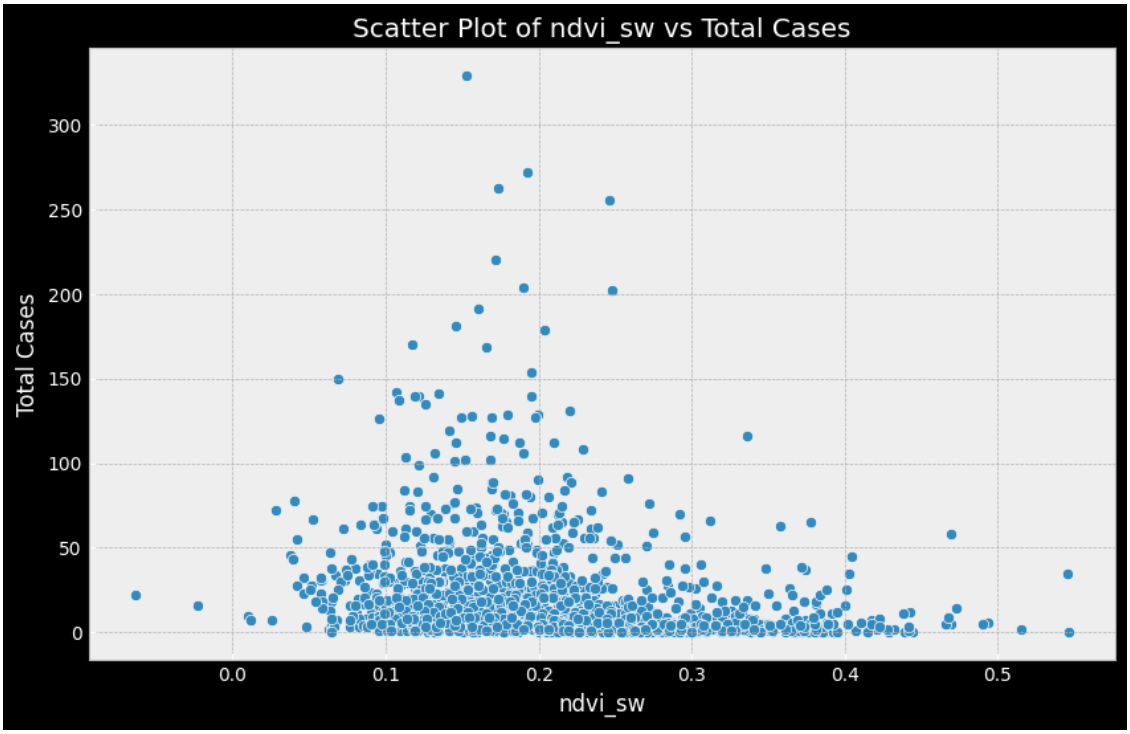

Scatter Plot of reanalysis_tdtr_k vs Total Cases
Total Cases
0
100
200
300
400
2
4
6
8
10
12
14
16
reanalysis_tdtr_k

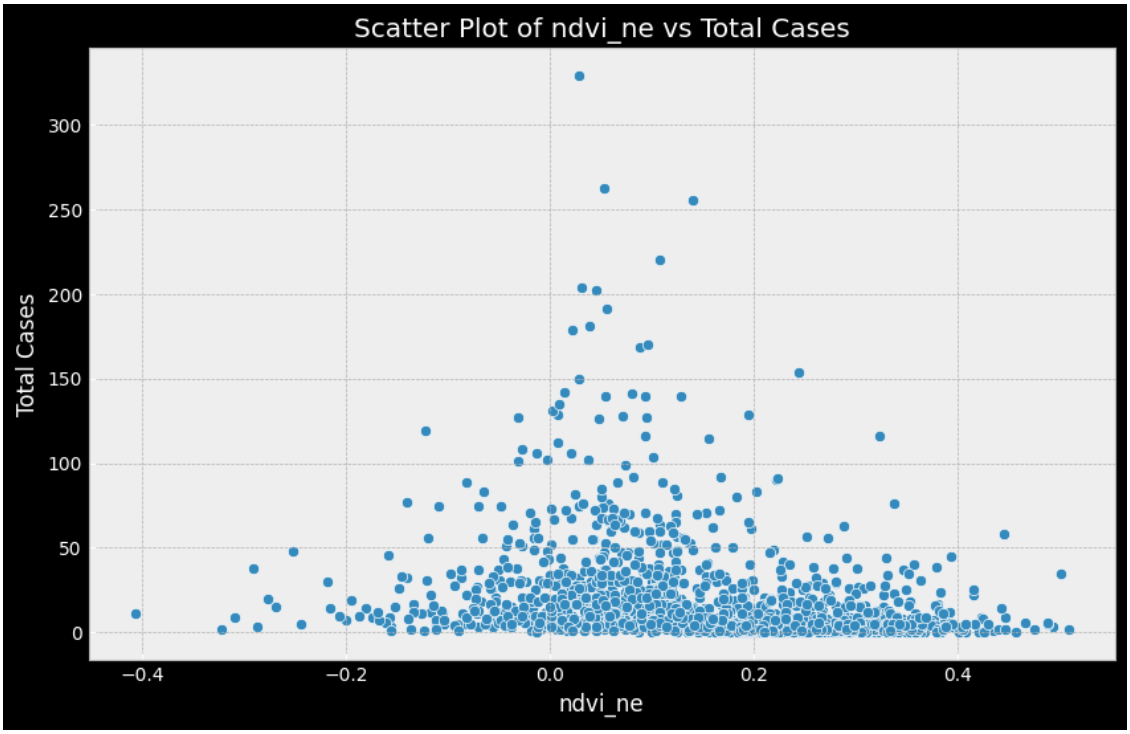

Scatter Plot of station_diur_temp_rng_c vs Total Cases
Total Cases
0
100
200
300
400
4
6
8
10
12
14
16
station_diur_temp_rng_c

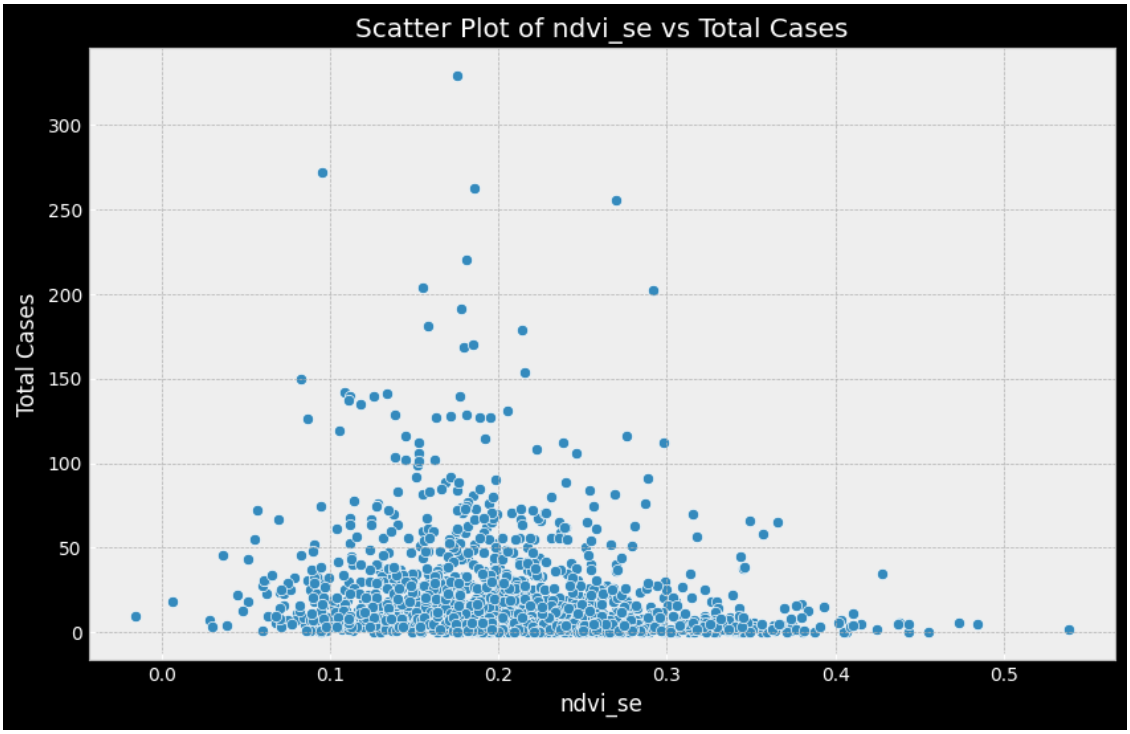

Scatter Plot of station_max_temp_c vs Total Cases
Total Cases
0
100
200
300
400
26
28
30
32
34
36
38
40
42
station_max_temp_c


Scatter Plot of station_precip_mm vs Total Cases
Total Cases
0
100
200
300
400
0
100
200
300
400
500
station_precip_mm

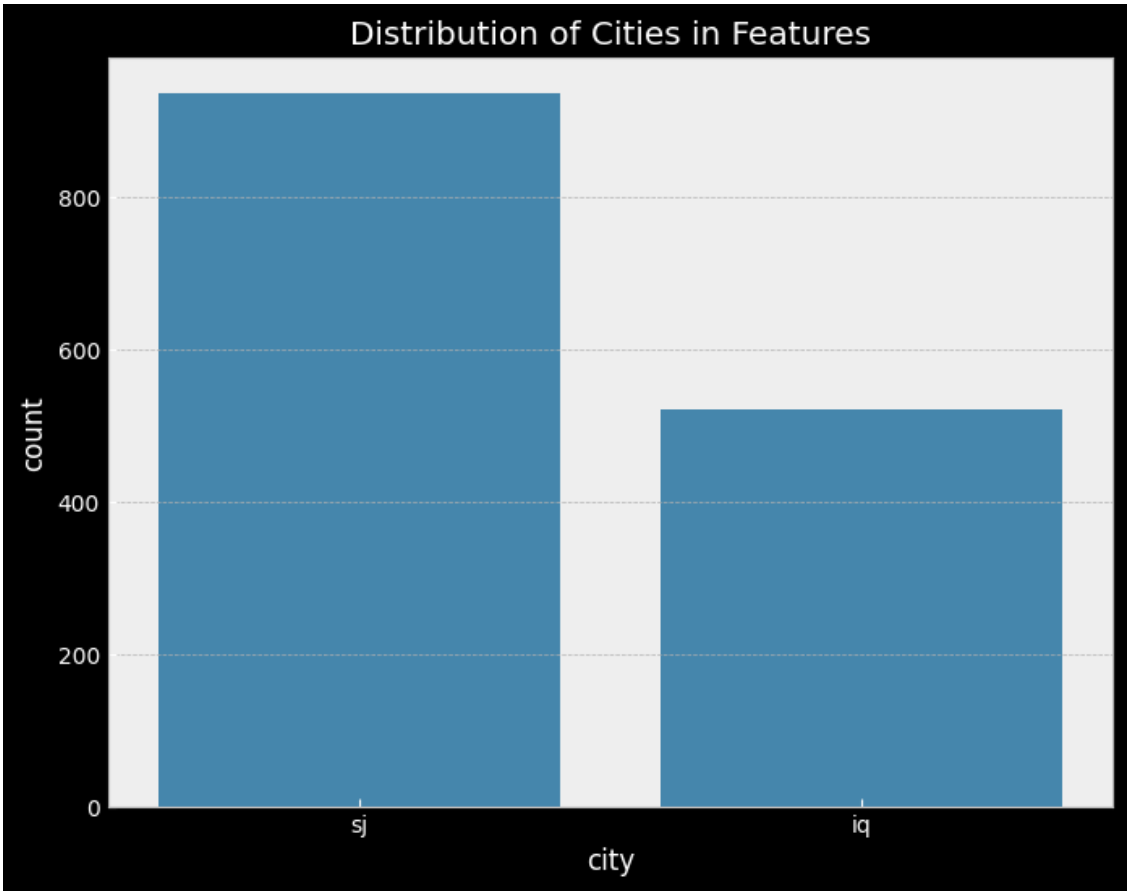

Scatter Plot of ndvi_nw vs Total Cases
Total Cases
ndvi_nw
300
250
200
150
100
50
0
−0.4
−0.2
0.0
0.2
0.4

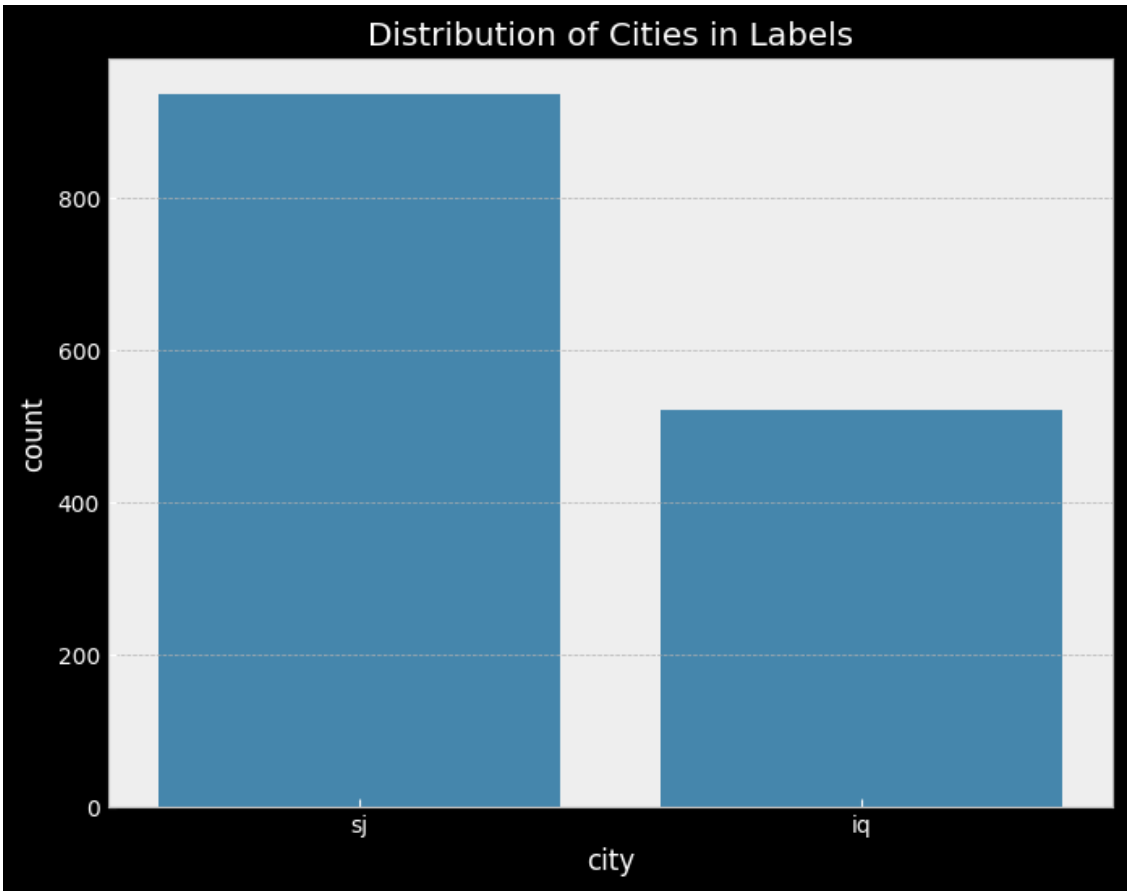

Scatter Plot of ndvi_sw vs Total Cases
Total Cases
ndvi_sw
300
250
200
150
100
50
0
0.0
0.1
0.2
0.3
0.4
0.5


Scatter Plot of ndvi_ne vs Total Cases
Total Cases
ndvi_ne
300
250
200
150
100
50
0
−0.4
−0.2
0.0
0.2
0.4


Scatter Plot of ndvi_se vs Total Cases
Total Cases
ndvi_se
300
250
200
150
100
50
0
0.0
0.1
0.2
0.3
0.4
0.5

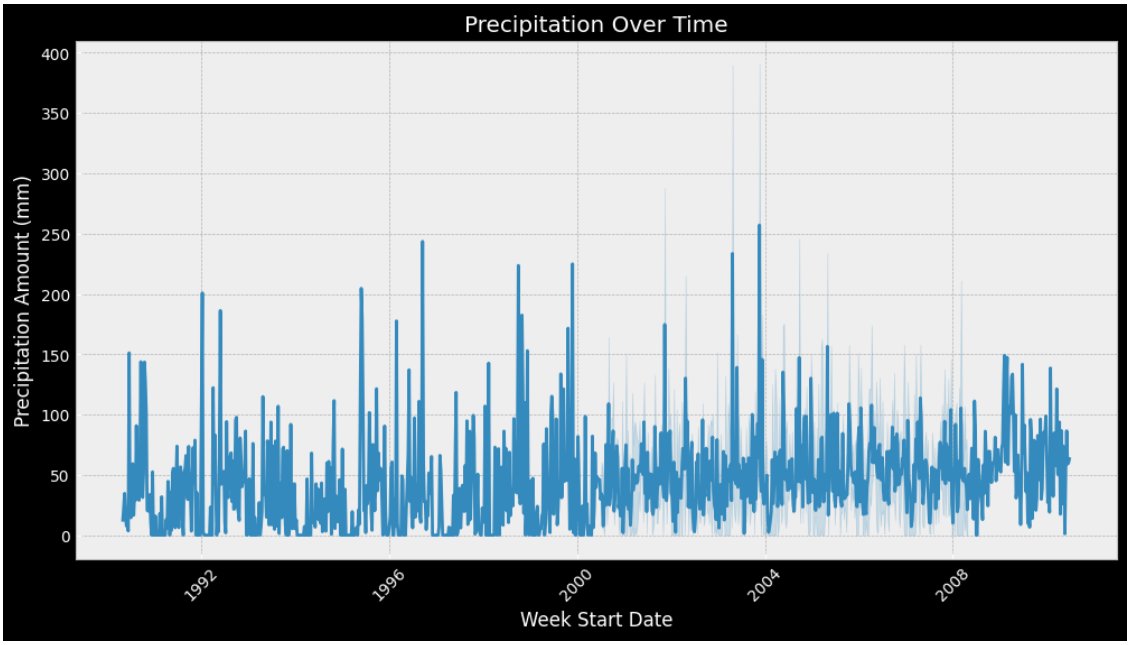
[247]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Categorical data: 'city' column in train_features
plt.figure(figsize=(8, 6))
sns.countplot(x='city', data=train_features)
plt.title('Distribution of Cities in Features')
plt.show()

# Categorical data: 'city' column in train_labels
plt.figure(figsize=(8, 6))
sns.countplot(x='city', data=train_labels)
plt.title('Distribution of Cities in Labels')
plt.show()
```





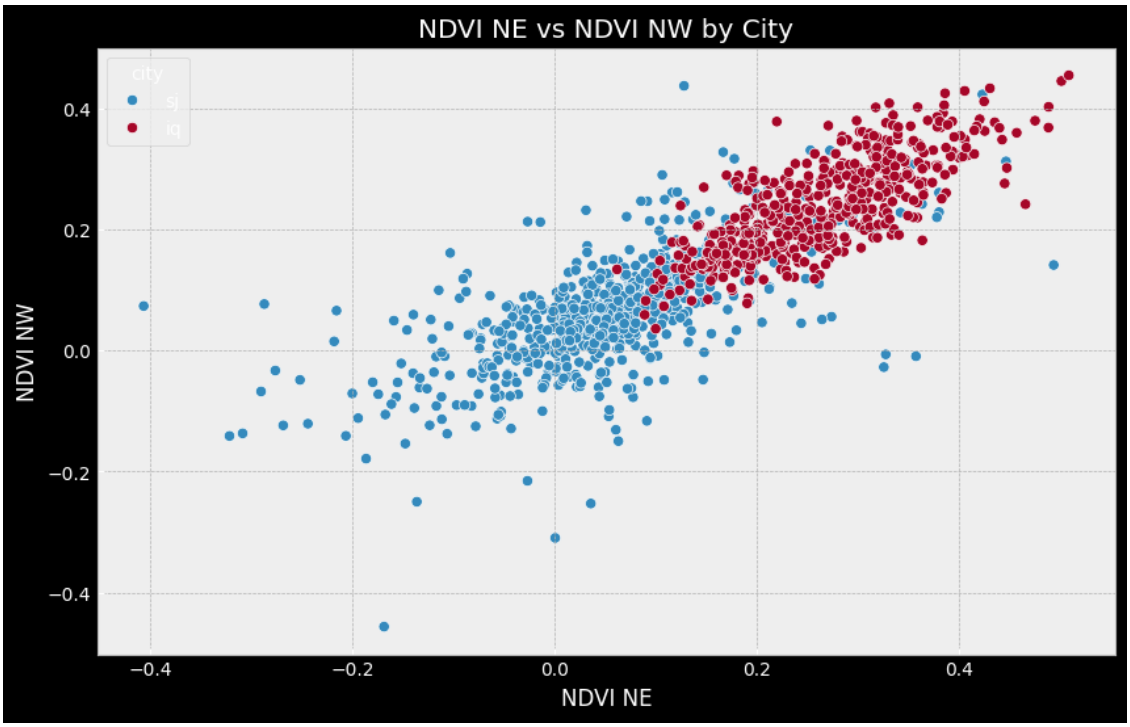
15 Visualise Timestamp Data

[248]:
```
import pandas as pd

# Convert 'week_start_date' to datetime
train_features['week_start_date'] = pd.
 ↪to_datetime(train_features['week_start_date'])

# Plotting
plt.figure(figsize=(12, 6))
sns.lineplot(x='week_start_date', y='precipitation_amt_mm',␣
 ↪data=train_features)  # Example variable
plt.title('Precipitation Over Time')
plt.xlabel('Week Start Date')
plt.ylabel('Precipitation Amount (mm)')
plt.xticks(rotation=45)
plt.show()
```

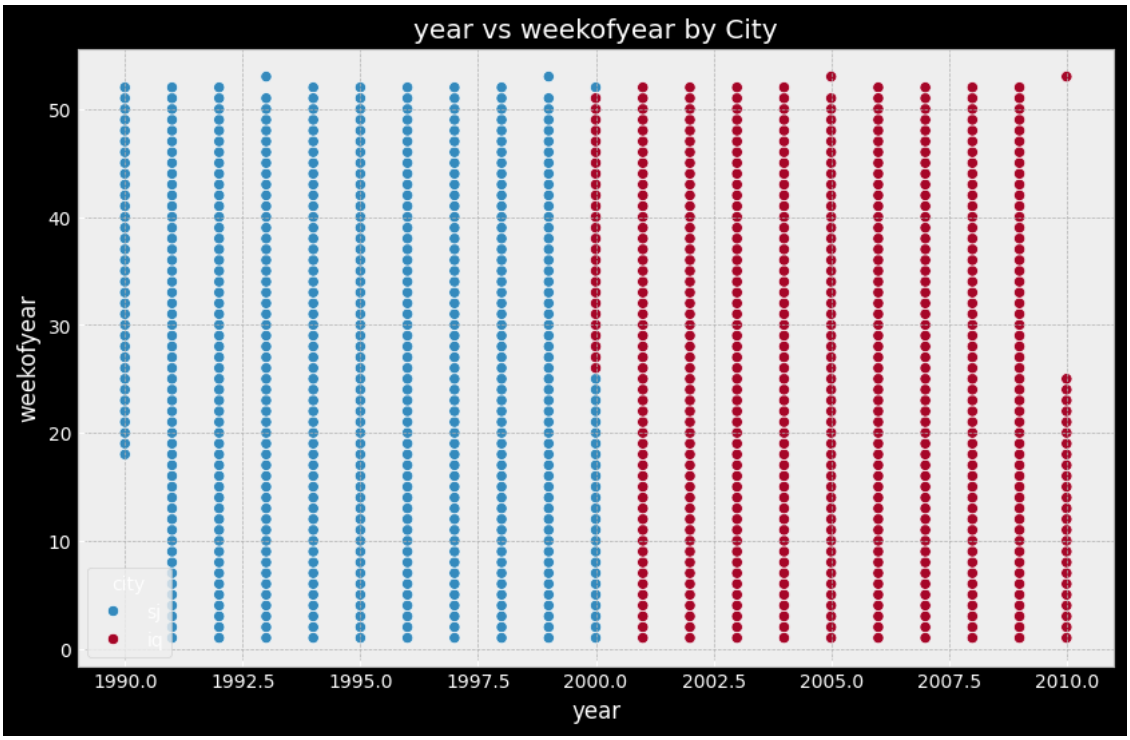


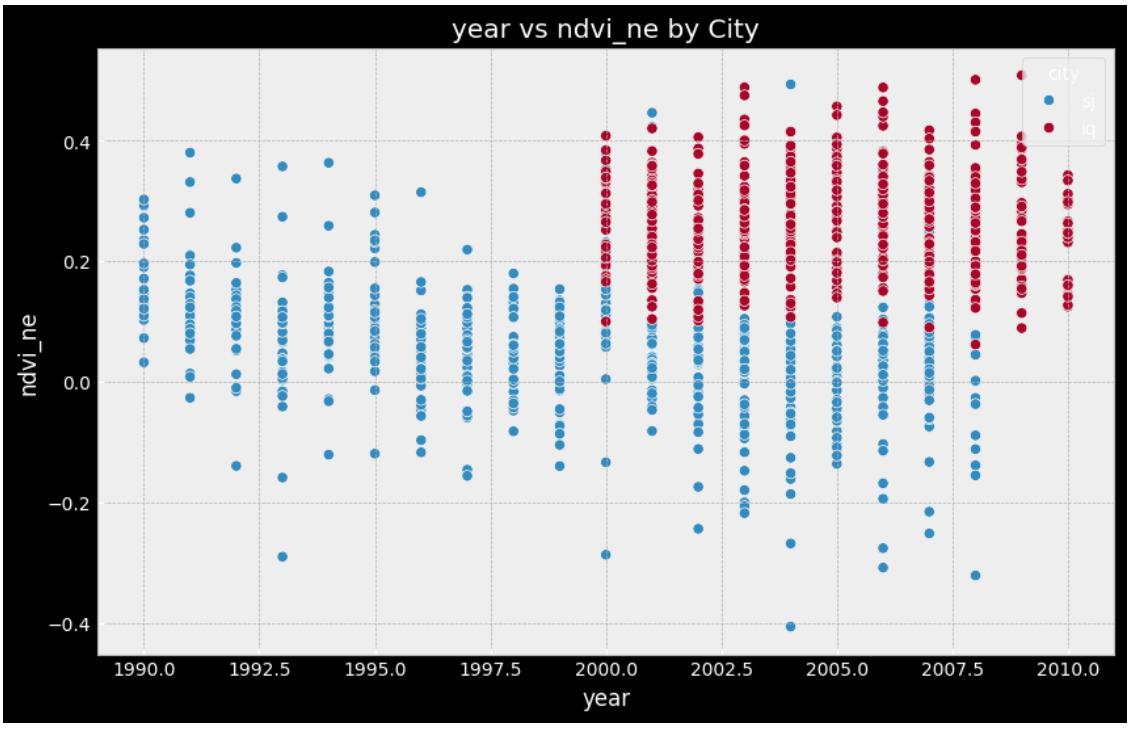
16 Visualise Numeric Data

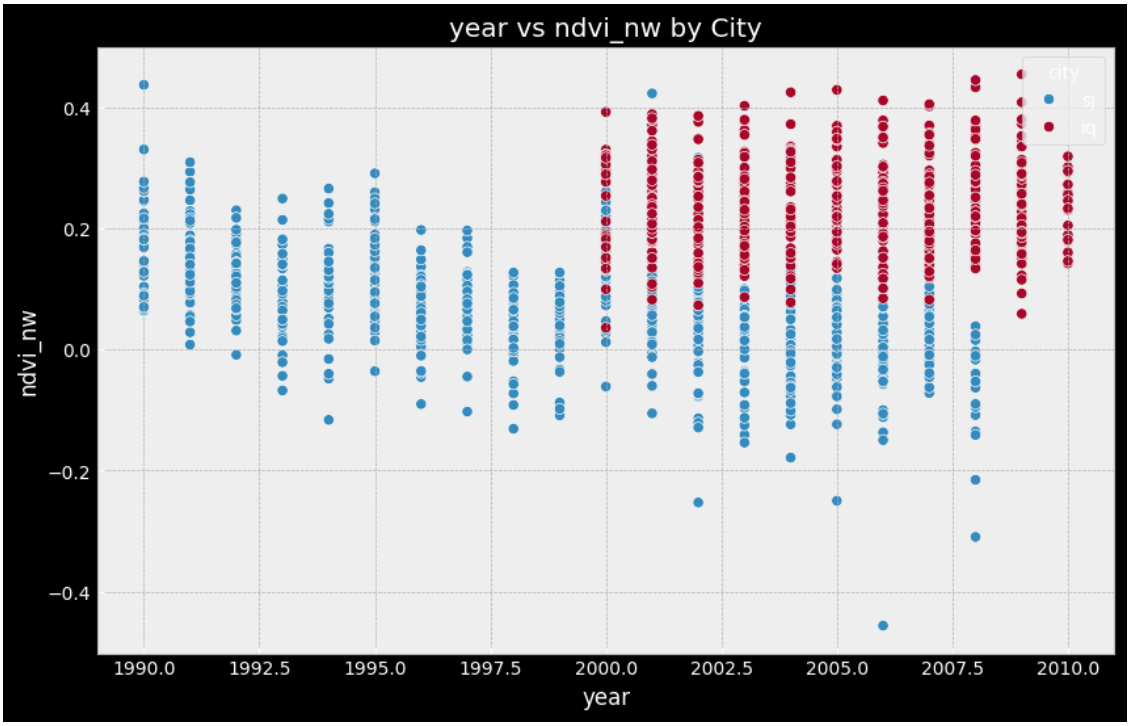
[249]:
```
# Numeric data: plotting 'ndvi_ne' vs 'ndvi_nw' as an example
plt.figure(figsize=(10, 6))
sns.scatterplot(x='ndvi_ne', y='ndvi_nw', hue='city', data=train_features)  #␣
↪'city' as hue to add color distinction
plt.title('NDVI NE vs NDVI NW by City')
plt.xlabel('NDVI NE')
plt.ylabel('NDVI NW')
plt.show()
```

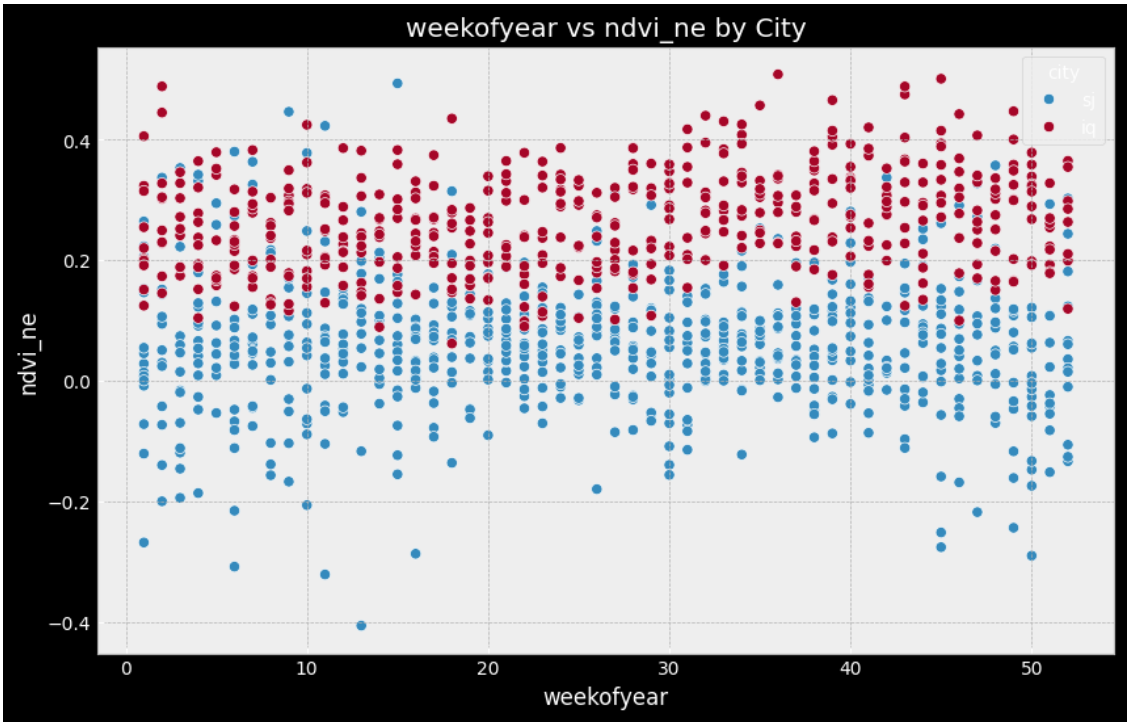


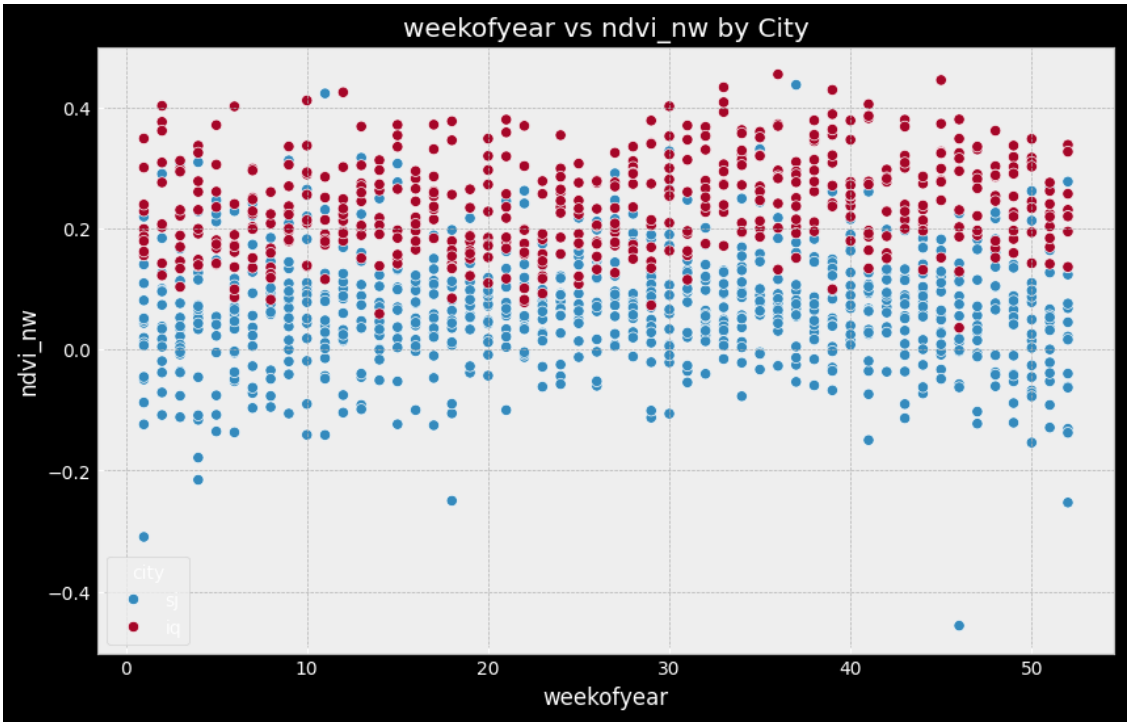
[250]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Subset of numeric features for demonstration
numeric_features_subset = numeric_features[:4]  # Adjust this as needed

# Loop through each pair of features
for i in range(len(numeric_features_subset)):
    for j in range(i+1, len(numeric_features_subset)):
        plt.figure(figsize=(10, 6))
        sns.scatterplot(x=numeric_features_subset[i],␣
↪y=numeric_features_subset[j], hue='city', data=train_features)
        plt.title(f'{numeric_features_subset[i]} vs␣
↪{numeric_features_subset[j]} by City')
        plt.xlabel(numeric_features_subset[i])
        plt.ylabel(numeric_features_subset[j])
        plt.show()
```

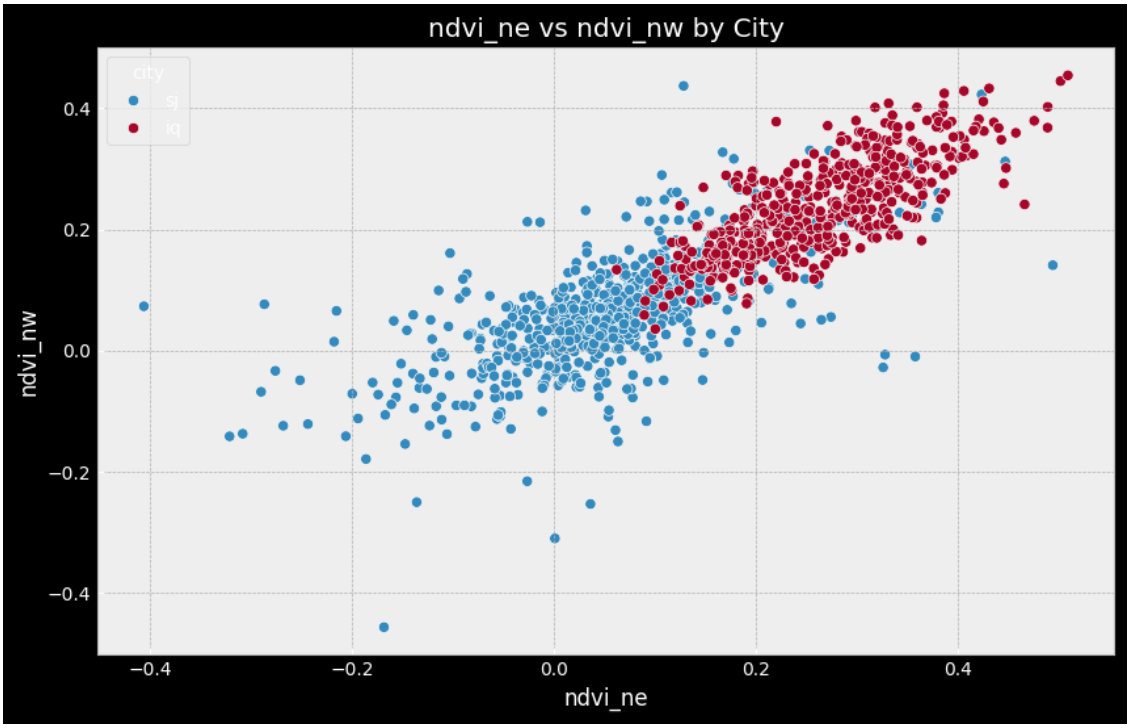

year vs weekofyear by City
weekofyear
year
city
sj
iq
0
10
20
30
40
50
1990.0
1992.5
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0


year vs ndvi_ne by City
ndvi_ne
year
city
sj
iq
−0.4
−0.2
0.0
0.2
0.4
1990.0
1992.5
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0


year vs ndvi_nw by City
ndvi_nw
year
city
sj
iq
0.4
0.2
0.0
−0.2
−0.4
1990.0
1992.5
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0


weekofyear vs ndvi_ne by City
ndvi_ne
weekofyear
city
sj
iq
0.4
0.2
0.0
−0.2
−0.4
0
10
20
30
40
50


weekofyear vs ndvi_nw by City
ndvi_nw
weekofyear
city
sj
iq
0.4
0.2
0.0
−0.2
−0.4
0
10
20
30
40
50


ndvi_ne vs ndvi_nw by City
ndvi_nw
ndvi_ne
city
sj
iq
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4