

Agency Anomaly Prediction V3

41

- 41 Outlier
- 41 Normal

11

- 11 Outlier
- 11 Normal

17

- 17 Outlier
- 17 Normal

```
In [105...]: import sklearn  
sklearn.__version__
```

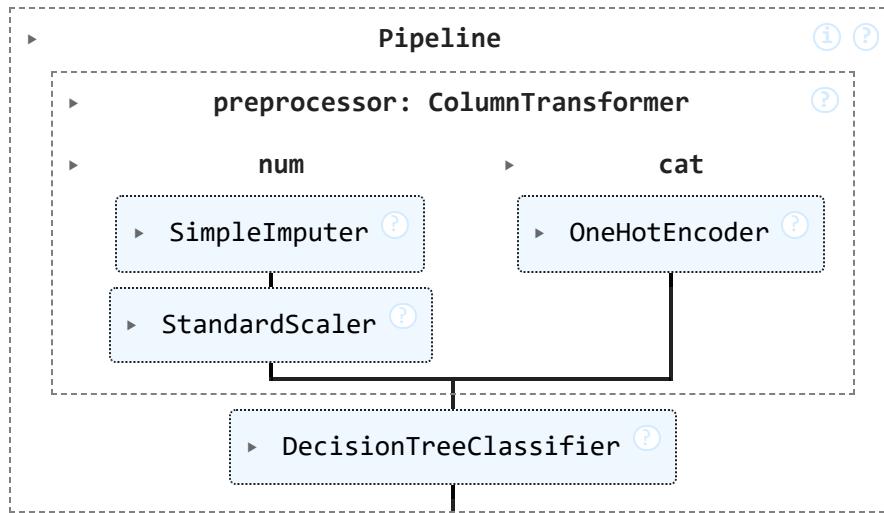
```
Out[105]: '1.4.2'
```

Prediction

```
In [106...]: import pickle  
import pandas as pd  
import numpy as np  
  
def load_model_file(filename):  
    obj = pickle.load(open(filename, 'rb'))  
    # print(obj)  
  
    model = obj['model']  
  
    return model
```

```
In [107...]: pipeline = load_model_file('agency_outlier_v3.pkl')  
pipeline
```

Out[107]:



```
In [108...]  
olddf = pd.read_csv('./data/grouped.csv')  
olddf.drop('date', axis=1, inplace=True)  
olddf['status'] = 'old'  
# olddf.head()
```

```
In [109...]  
# tps_df = pd.read_csv('tps-unLoad.csv')  
# tps_df.query("TPMI_TRANS_CD==41")
```

```
In [110...]  
# tps_df_dec = pd.read_csv('tp-unLoad-december.csv')  
# tps_df_dec.query("TPMI_TRANS_CD==41")
```

```
In [111...]  
# df = pd.DataFrame({  
#     'year': tps_df.TPMI_BLG_YR,  
#     'month': tps_df.TPMI_BLG_MM,  
#     'tpmi_agncy_cd': tps_df.TPMI_AGNCY_CD,  
#     'tpmi_trans_cd': tps_df.TPMI_TRANS_CD.astype(str),  
#     'tpmi_trans_count': tps_df.TPMI_TRANS_COUNT,  
#     'status': 'new'  
# })  
# del tps_df  
# # df.info()
```

```
In [113...]  
def get_and_convert(file, status):  
    tps_df = pd.read_csv(file)  
    print(len(tps_df))  
  
    df = pd.DataFrame({  
        'year': tps_df.TPMI_BLG_YR,  
        'month': tps_df.TPMI_BLG_MM,  
        'tpmi_agncy_cd': tps_df.TPMI_AGNCY_CD,  
        'tpmi_trans_cd': tps_df.TPMI_TRANS_CD.astype(str),  
        'tpmi_trans_count': tps_df.TPMI_TRANS_COUNT,  
        'status': status  
    })  
    return df
```

```
In [114...]  
# df.head()  
# df.tpmi_trans_cd.unique()
```

```
In [115...]  
# rec = df.query('tpmi_agncy_cd == "050" and tpmi_trans_cd=="41"')  
# rec
```

```
In [116...]  
combined = pd.concat([  
    olddf,  
    get_and_convert('./data/tps-unload-november.csv', 'old'),  
    get_and_convert('./data/tp-unload-december.csv', 'old'),  
    get_and_convert('./data/tps-data-2025-jan.csv', 'new')  
]).sort_values(by=['year', 'month']).reset_index(drop=True)
```

```
1169  
1165  
1194
```

```
In [117...]  
def extract_features(df):  
    df['count_change_pct'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).tpmi_trans  
    # df['count_change_pct_pct'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).count  
  
    df['std'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).tpmi_trans_count.trans  
    df['mean'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).tpmi_trans_count.tran  
    df['max'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).tpmi_trans_count.trans  
    df['min'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).tpmi_trans_count.trans  
  
    N = 4  
  
    df['last_n_pos'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).count_change_pct  
        group.shift(1).rolling(window=N).apply(lambda x: all(x > 0)))  
    df['last_n_neg'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd']).count_change_pct  
        group.shift(1).rolling(window=N).apply(lambda x: all(x < 0)))  
  
    # df['std'] = df.groupby(['tpmi_agncy_cd', 'tpmi_trans_cd'])  
  
    df.replace([np.inf, -np.inf], [99999, -99999], inplace=True)  
  
extract_features(combined)
```

```
In [118...]  
# combined.head()
```

```
Out[118]:  
tpmi_agncy_cd  tpmi_trans_cd  tpmi_trans_count  tpmi_trans_amt_due  year  month  status  count_c  
0             010          11         3483      2260887.5  2020       6   old  
1             010          14          52      123279.4  2020       6   old  
2             010          15          29          0.0  2020       6   old  
3             010          16         1225     266798.0  2020       6   old  
4             010          17          371     196233.8  2020       6   old
```

```
In [145...]  
# combined['outlier'] = 0
```

```
# new_df = combined.query('status=="new"').copy()  
# new_df['outlier'] = pipeline.predict(new_df)  
  
# combined['outlier'] = [  
#     new_df
```

```

# ]

combined['outlier'] = 1

# Filter the rows where status == "new"
new_status_rows = combined.query('status=="new"')

# Predict on the filtered rows
predictions = pipeline.predict(new_status_rows)

# Assign predictions back to the original DataFrame
combined.loc[combined['status'] == "new", 'outlier'] = predictions

```

```

/opt/conda/lib/python3.10/site-packages/sklearn/base.py:493: UserWarning: X does not
have valid feature names, but StandardScaler was fitted with feature names
    warnings.warn(
/opt/conda/lib/python3.10/site-packages/sklearn/preprocessing/_encoders.py:241: UserW
arning: Found unknown categories in columns [0, 1] during transform. These unknown ca
tegories will be encoded as all zeros
    warnings.warn(
/opt/conda/lib/python3.10/site-packages/sklearn/base.py:493: UserWarning: X does not
have valid feature names, but DecisionTreeClassifier was fitted with feature names
    warnings.warn(

```

In [146...]

```
combined.head()
```

Out[146]:

| | tpmi_agncy_cd | tpmi_trans_cd | tpmi_trans_count | tpmi_trans_amt_due | year | month | status | count_c |
|----------|---------------|---------------|------------------|--------------------|------|-------|--------|---------|
| 0 | 010 | 11 | 3483 | 2260887.5 | 2020 | 6 | old | |
| 1 | 010 | 14 | 52 | 123279.4 | 2020 | 6 | old | |
| 2 | 010 | 15 | 29 | 0.0 | 2020 | 6 | old | |
| 3 | 010 | 16 | 1225 | 266798.0 | 2020 | 6 | old | |
| 4 | 010 | 17 | 371 | 196233.8 | 2020 | 6 | old | |

In [159...]

```

import matplotlib.pyplot as plt

marker_dict = {
    1: 'o',
    -1: 'x'
}

def show_outliers(df, code, agencies=None):
    agencies = agencies if agencies is not None else df.tpmi_agncy_cd.unique()
    outlier_col = 'outlier'

    plt.figure(figsize=(12, 2))
    for agency in sorted(agencies):

        for category in df.outlier.unique():

```

```

# print(category)
df_subset = df[
    (df[outlier_col] == category)
    & (df['tpmi_agency_cd'] == agency)
    & (df['tpmi_trans_cd'] == code)
].copy()

if len(df_subset) == 0:
    continue

marker = 'x' if category == -1 else 'o'
plt.scatter(df_subset['date'], df_subset['tpmi_trans_count'], marker=marker)
# plt.plot(df_subset['date'], df_subset['tpmi_trans_amt_due'], marker=marker)
plt.title('Transaction Count Trend')
plt.xlabel('Date')
plt.ylabel('Transaction Count')
plt.legend()
plt.grid(True)
plt.show()

```

In [160...]

```

combined['date'] = (
    pd.to_datetime(
        combined[['year', 'month']]
        .assign(day=1)
    )
)

# combined

```

41

41 - Predicted as Outlier

In [161...]

```
# combined.query("tpmi_trans_cd=='41' and tpmi_agency_cd == '110'")[-5:]
```

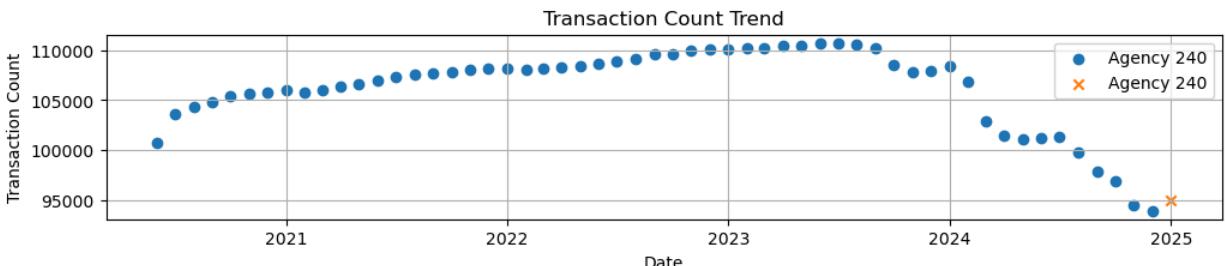
In [173...]

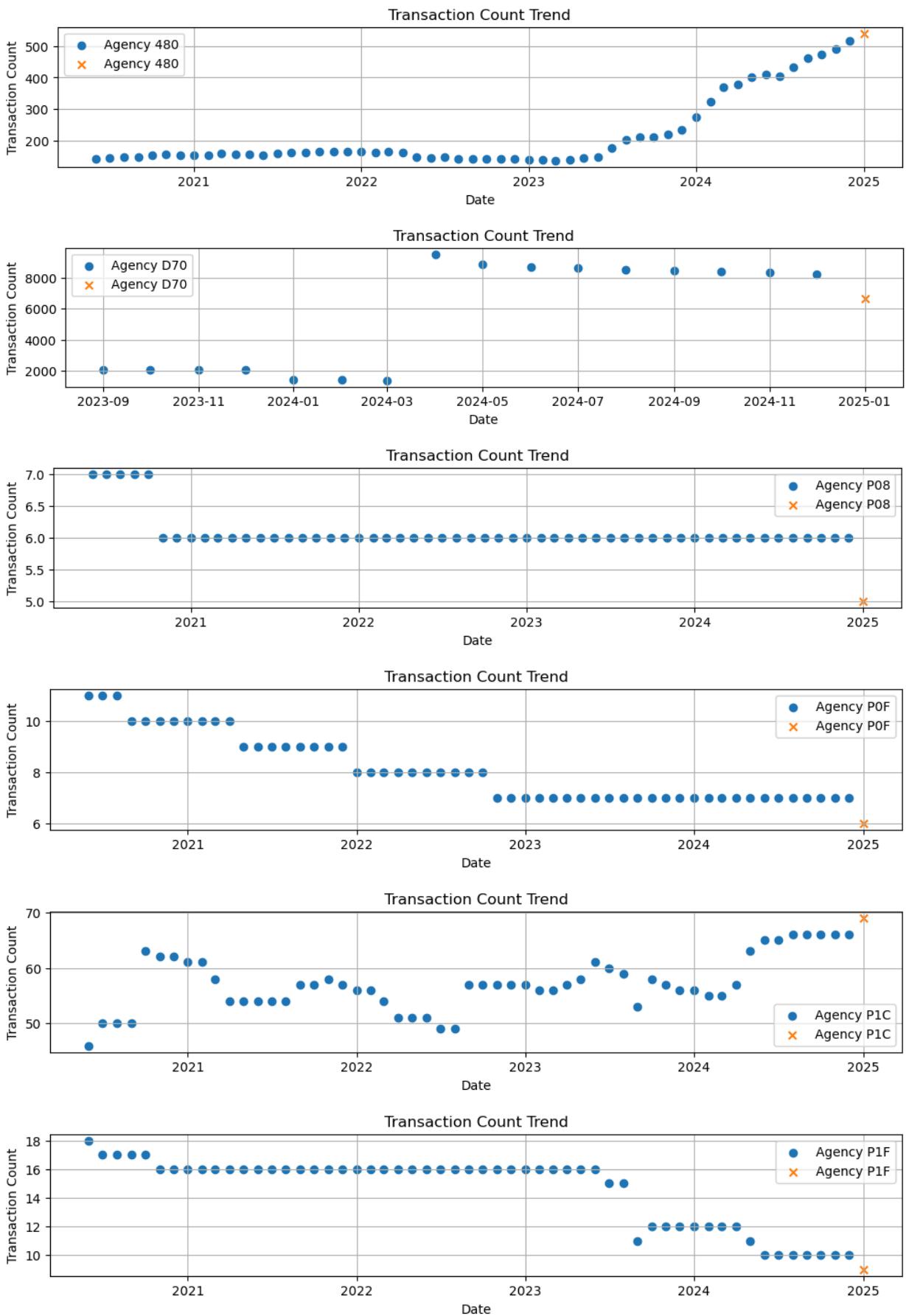
```
outlier_df_41 = combined.query("status=='new' and tpmi_trans_cd=='41' and outlier==-1")
print(' '.join(sorted(outlier_df_41.tpmi_agency_cd.unique())))
```

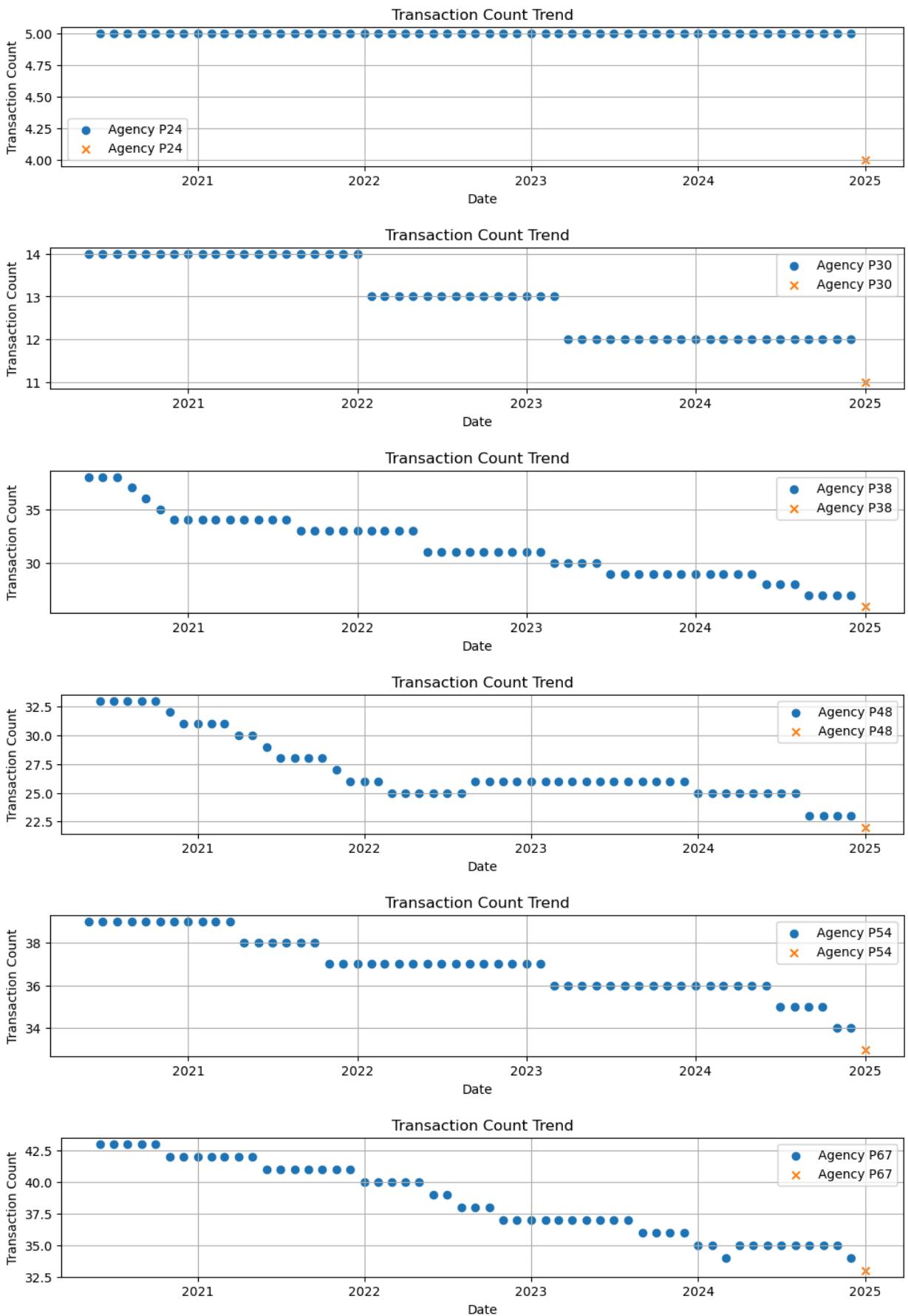
240 480 D70 P08 P0F P1C P1F P24 P30 P38 P48 P54 P67 P77 P83 S15 S27 S32 S34 S41 X05 X51

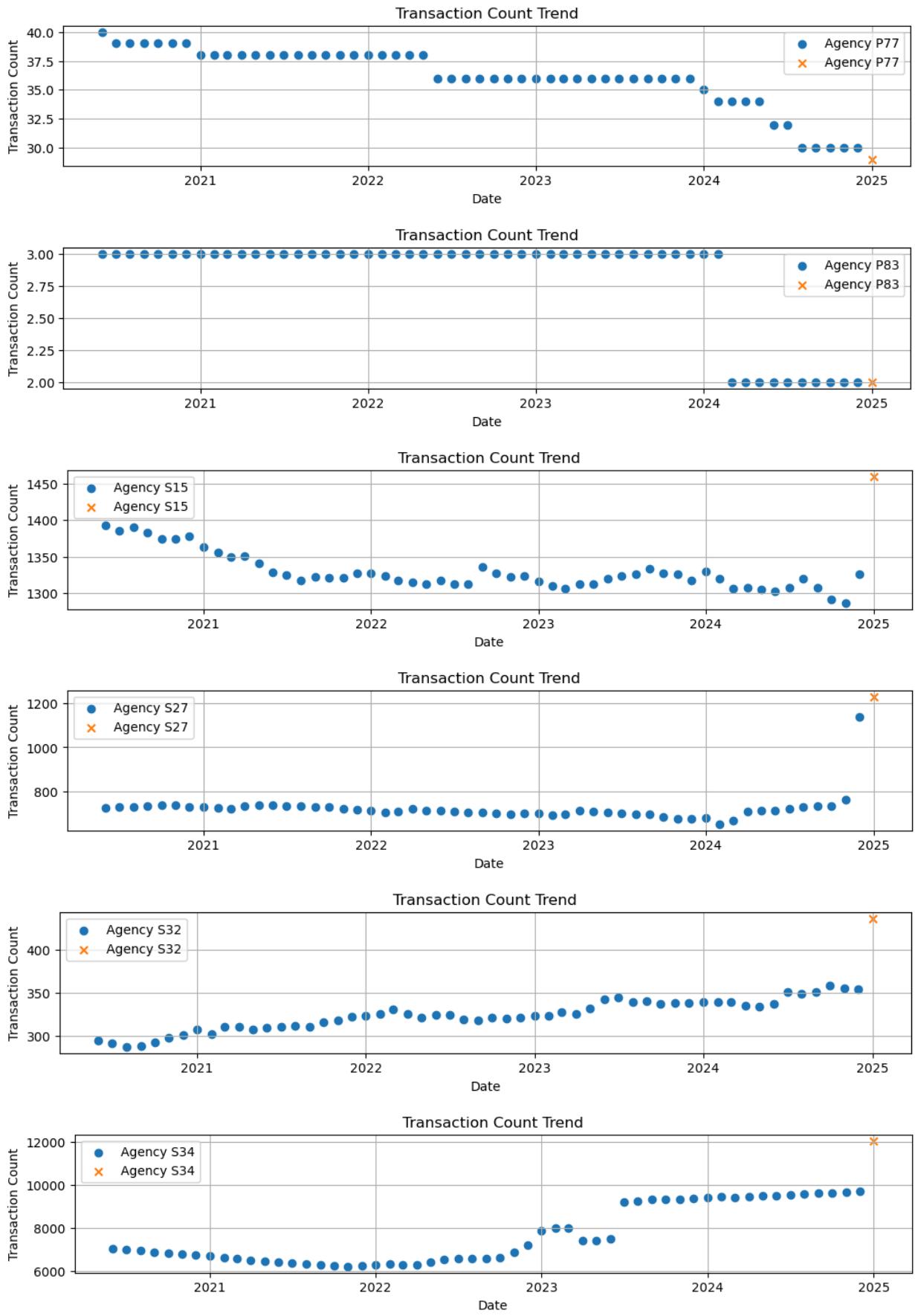
In [174...]

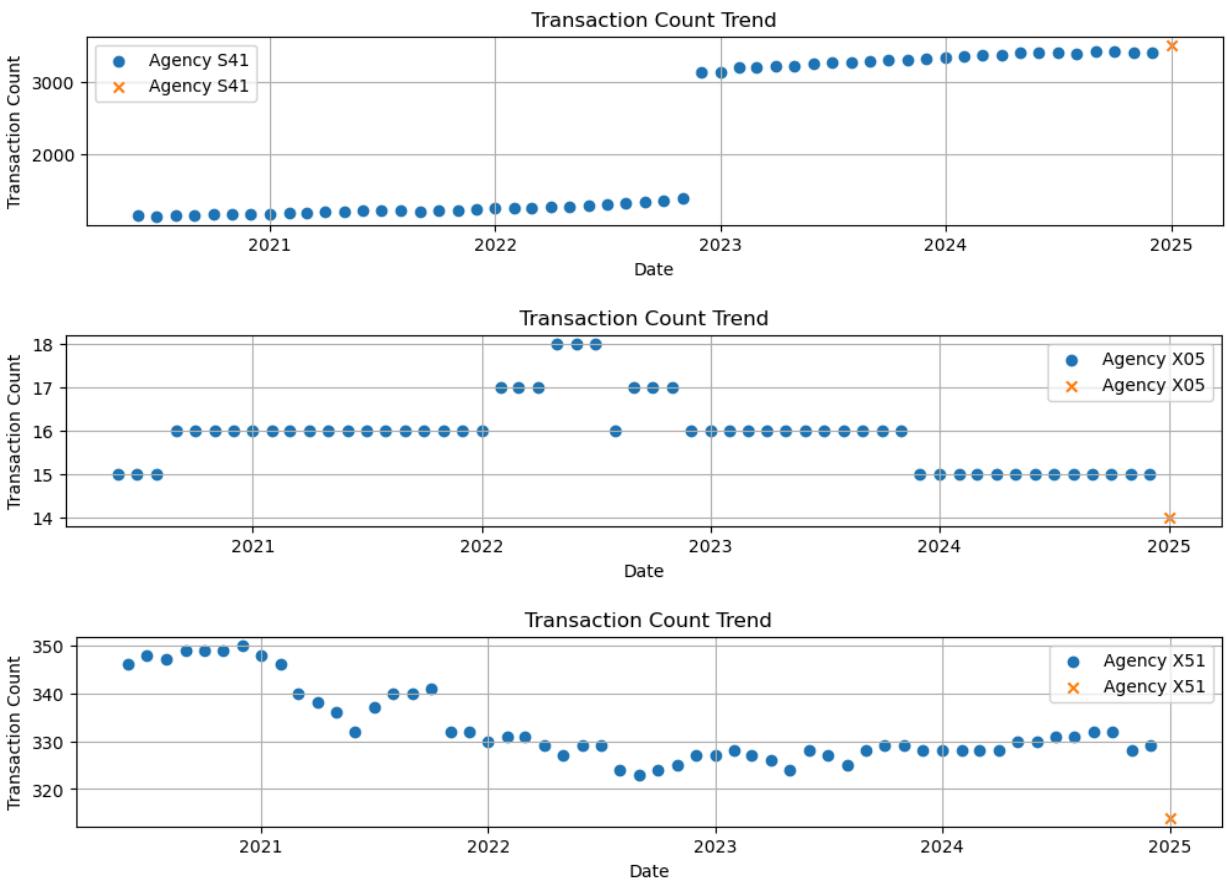
```
for agency in sorted(outlier_df_41.tpmi_agency_cd.unique()):
    show_outliers(combined, '41', [agency])
```







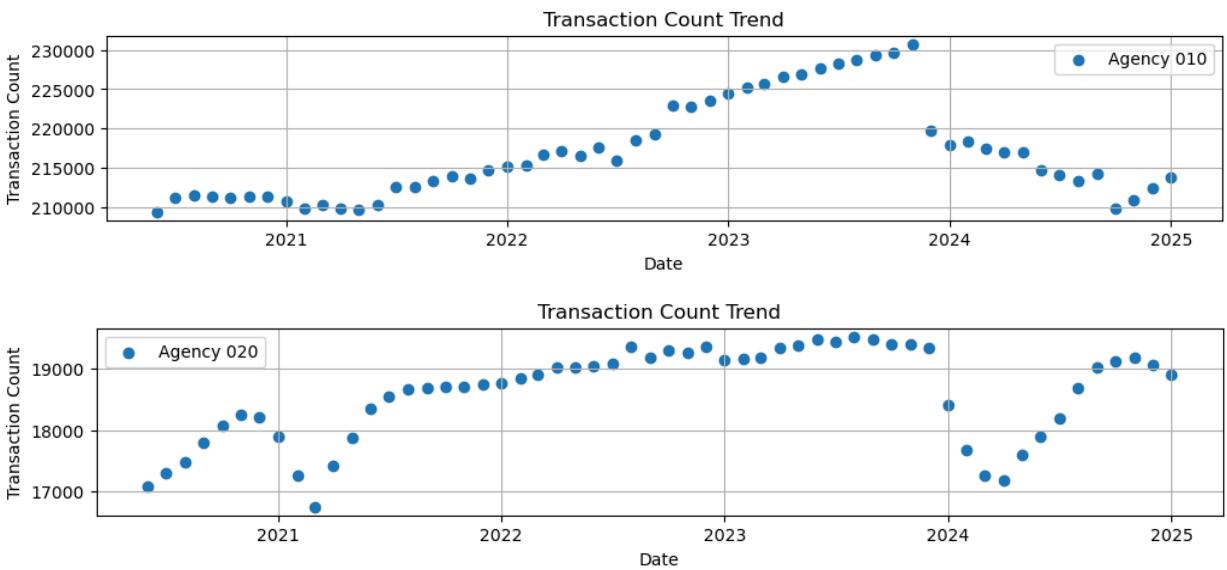


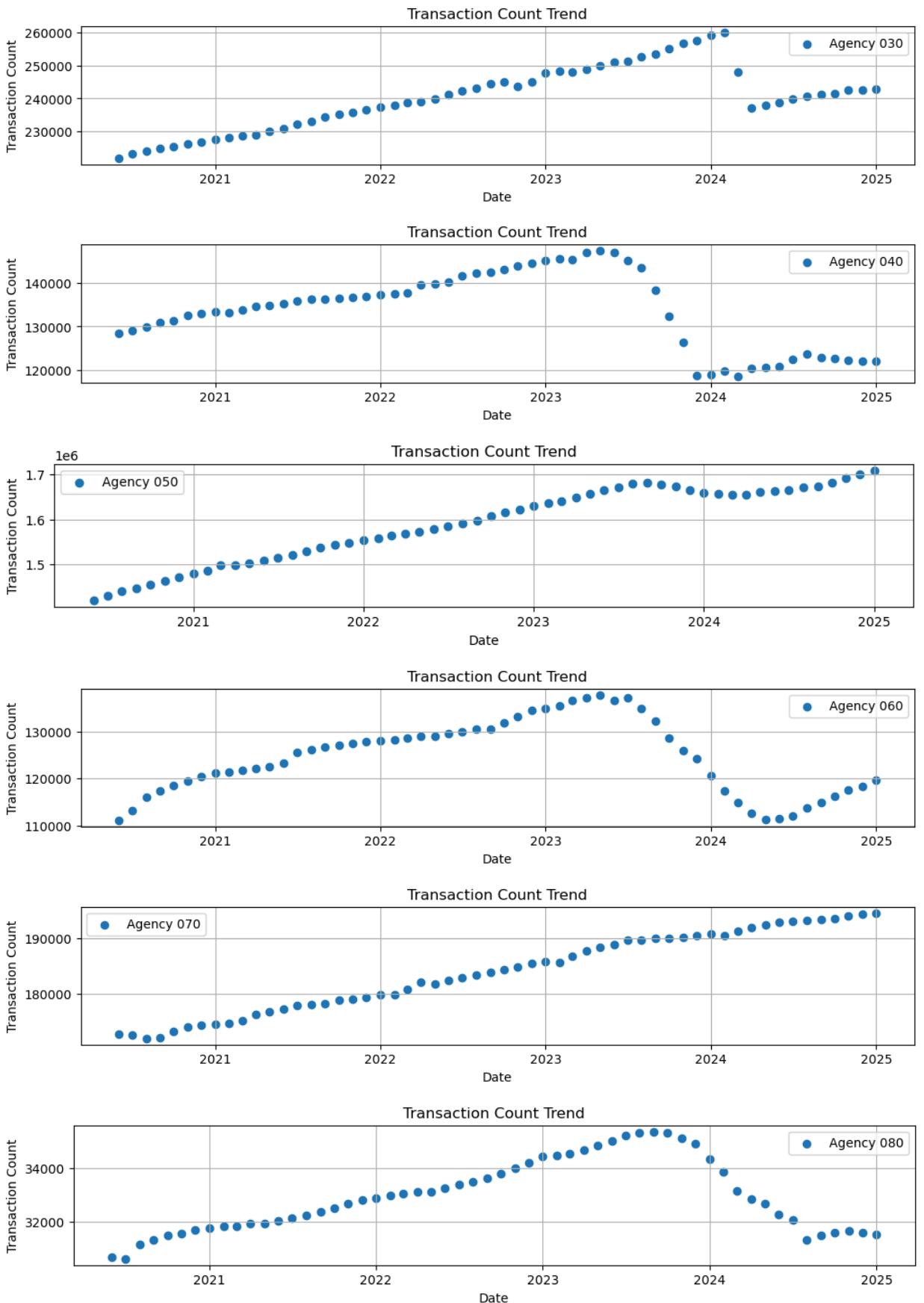


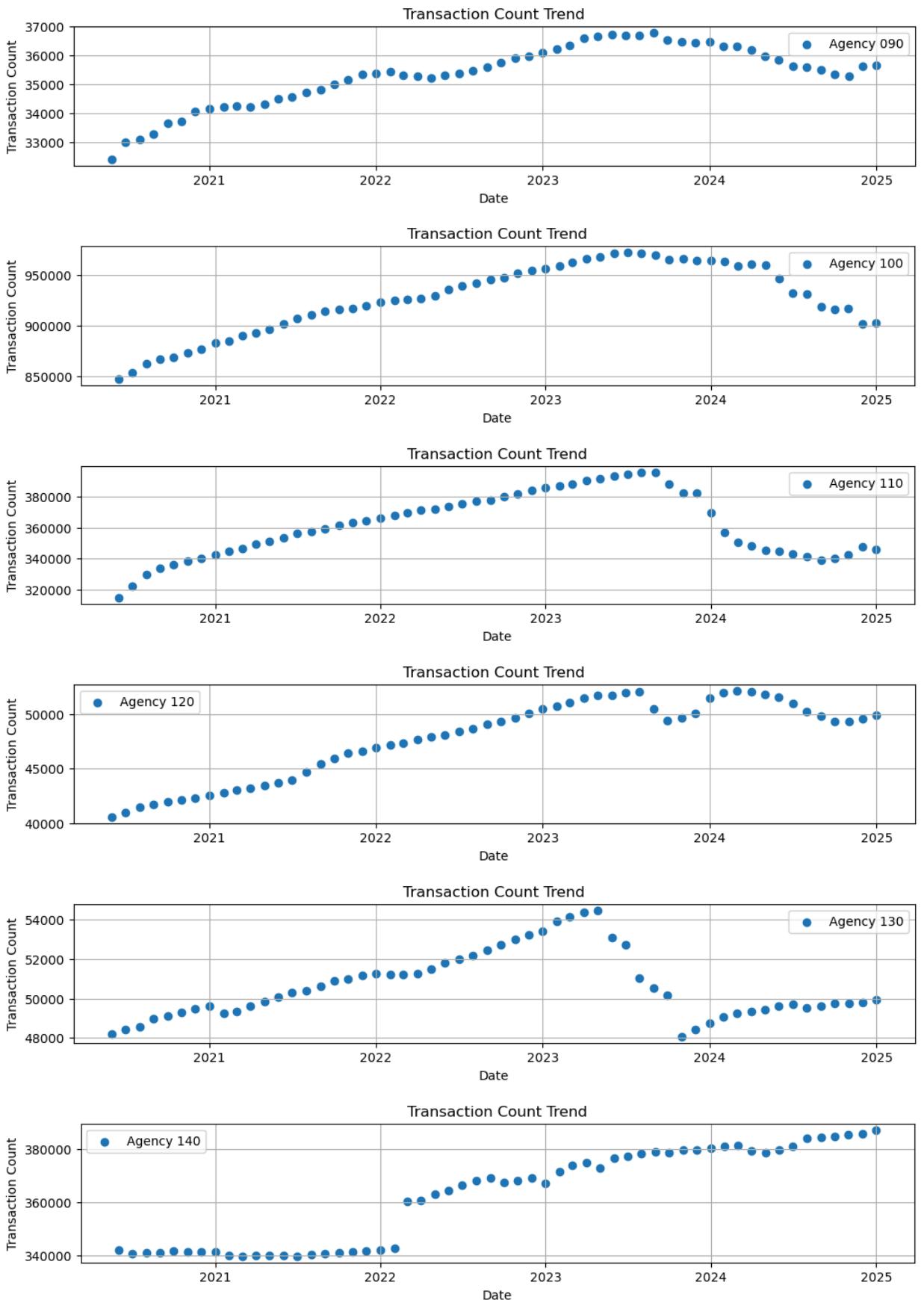
41 - Predicted as Normal

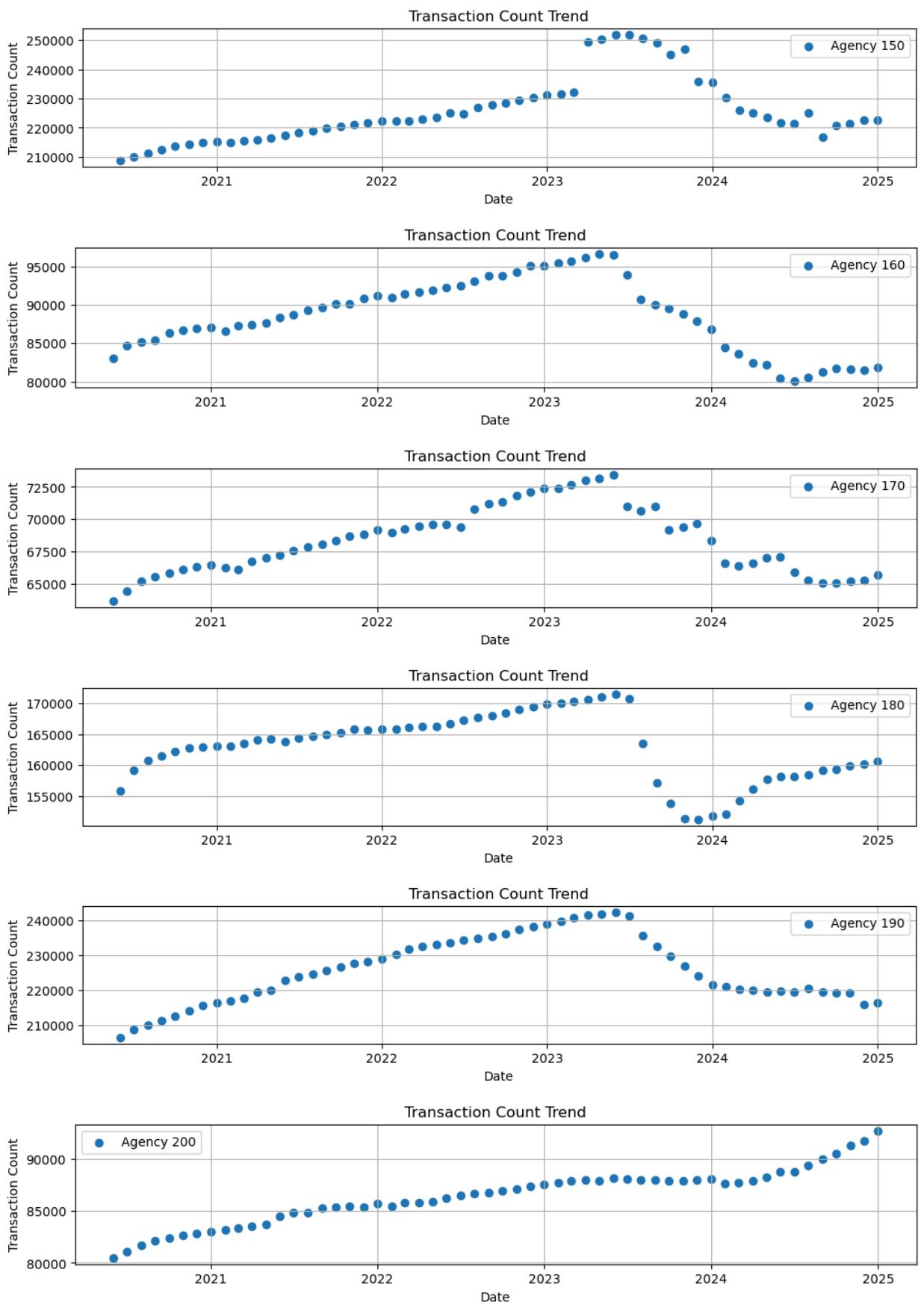
In [175...]

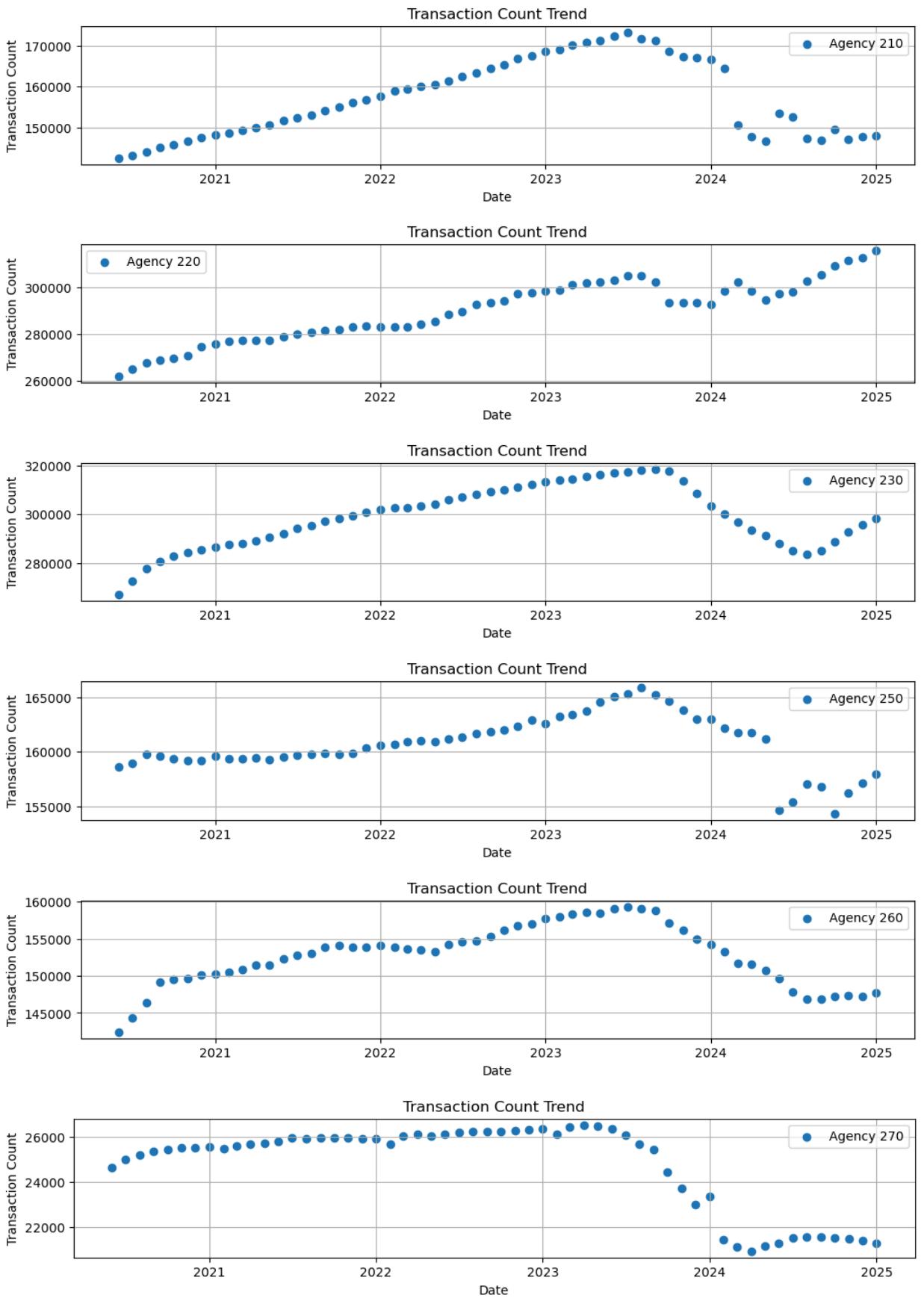
```
for agency in sorted(combined.query("status=='new' and tpmi_trans_cd=='41' and outlier
show_outliers(combined, '41', [agency])
```

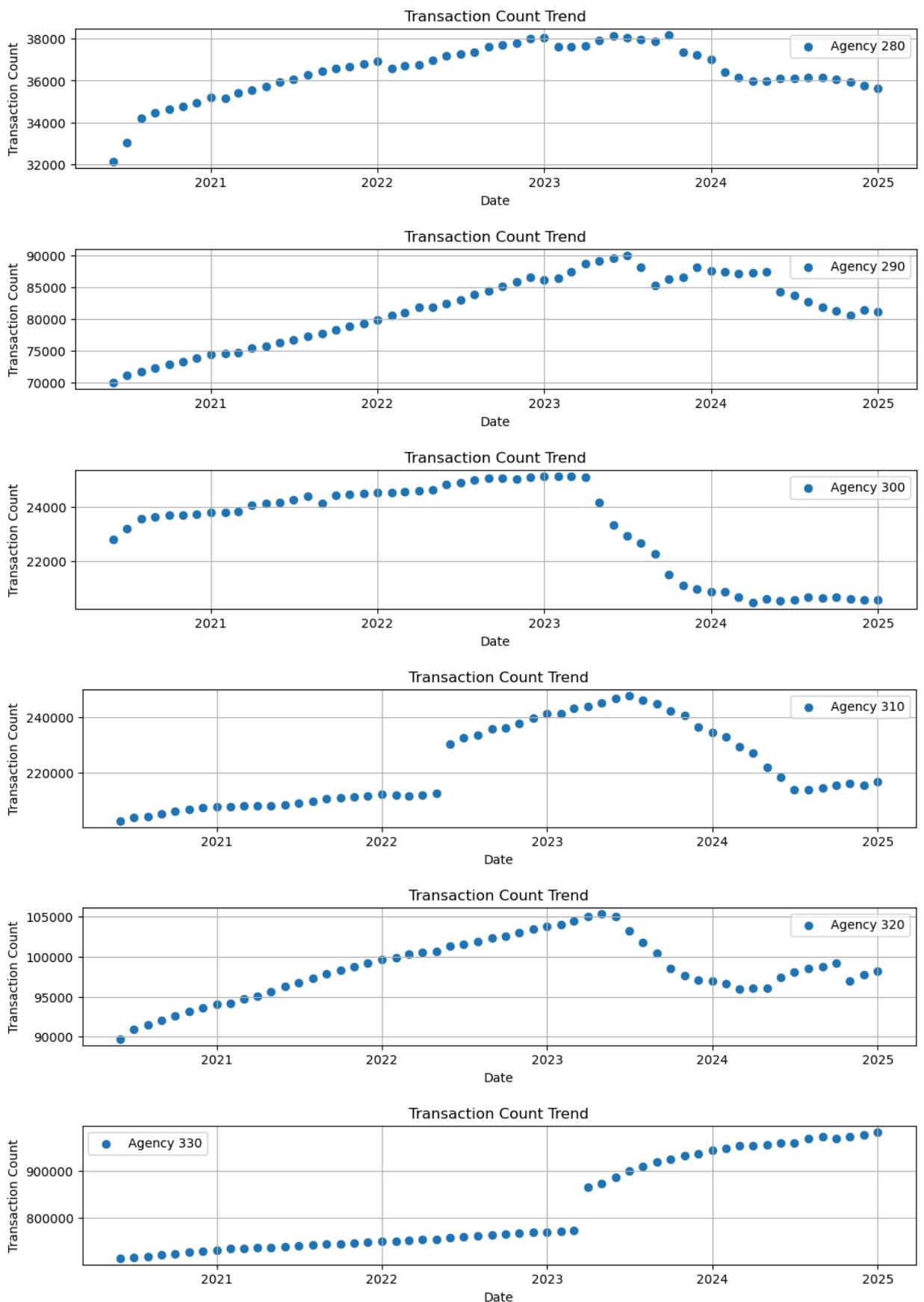


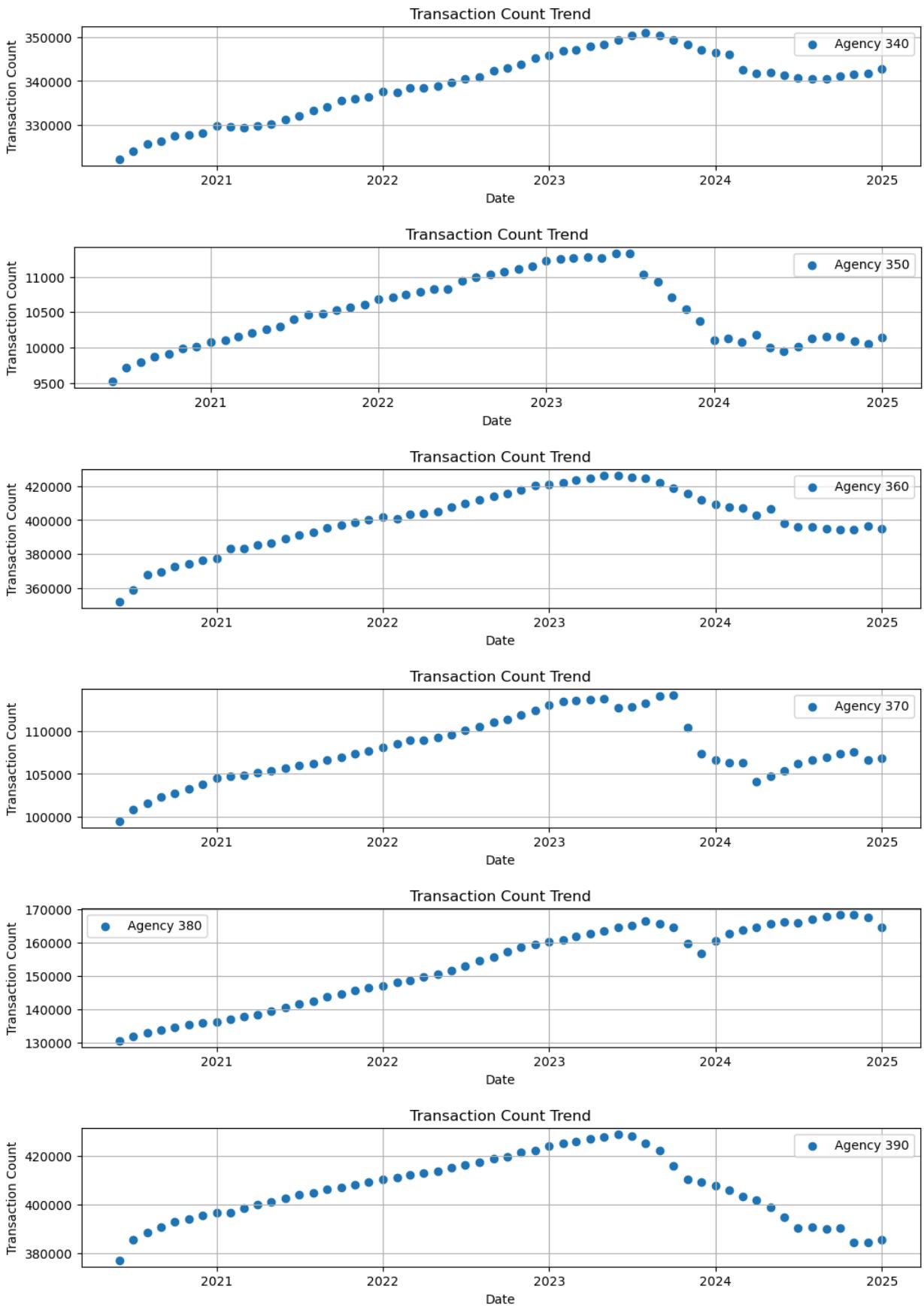


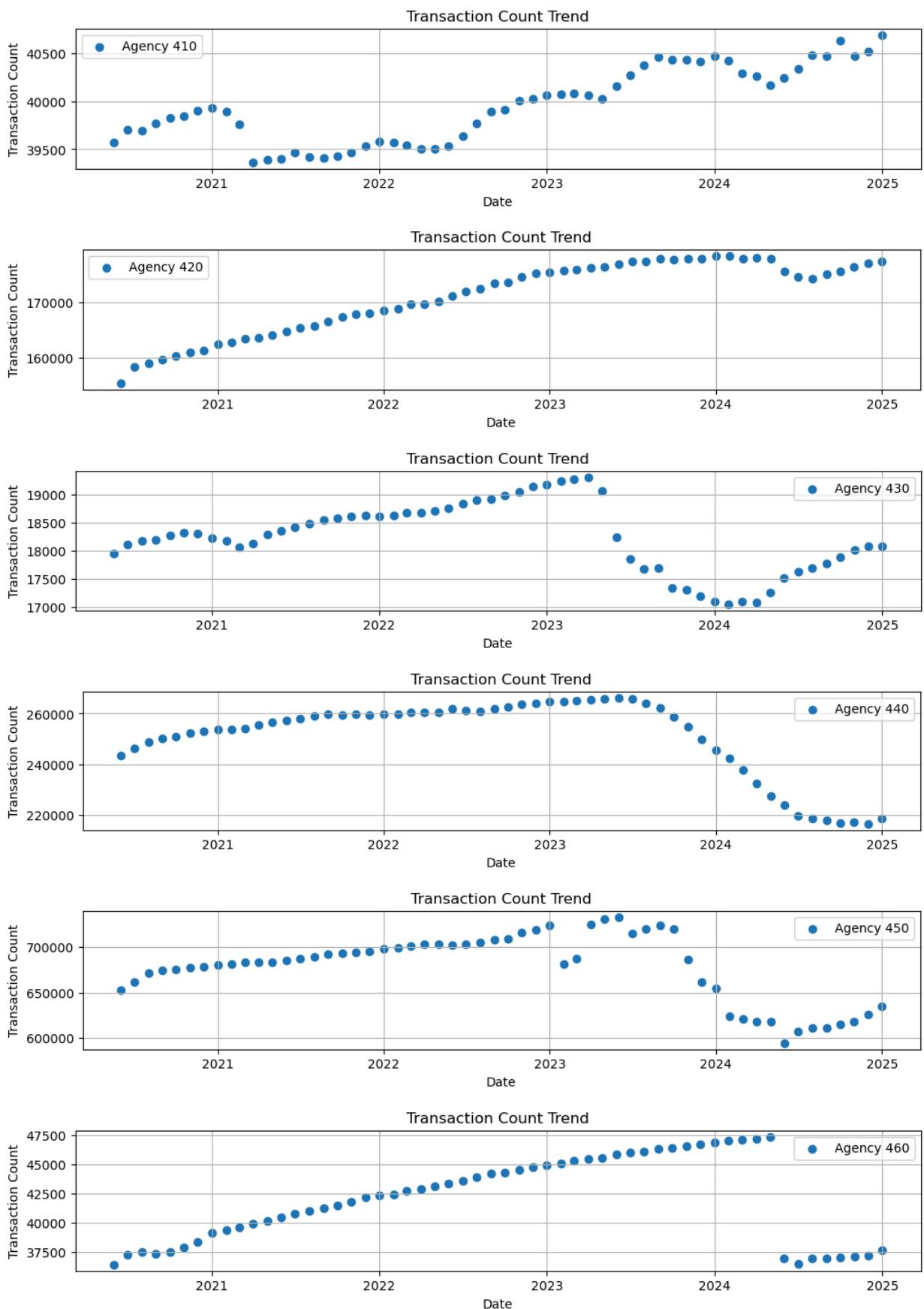


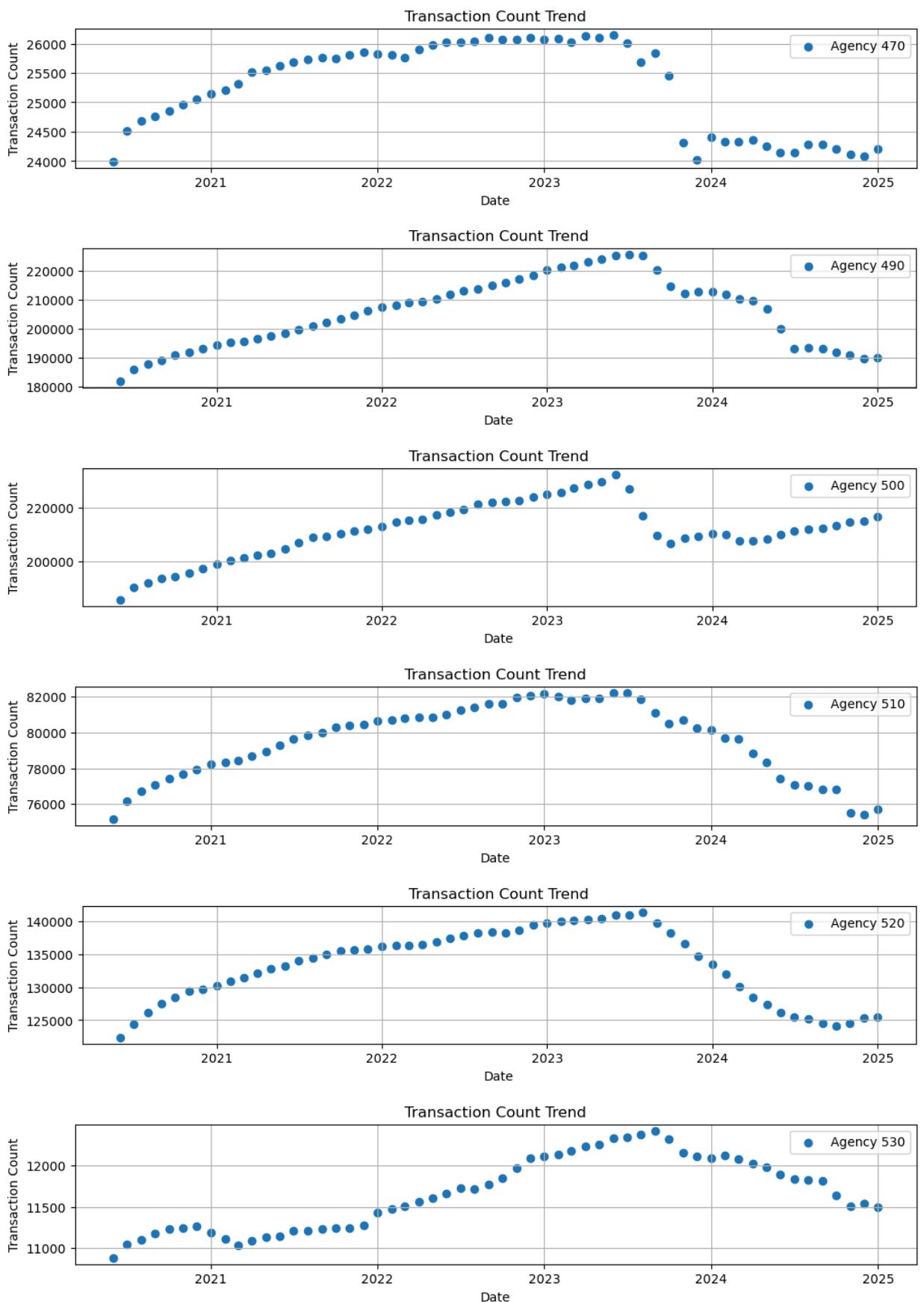


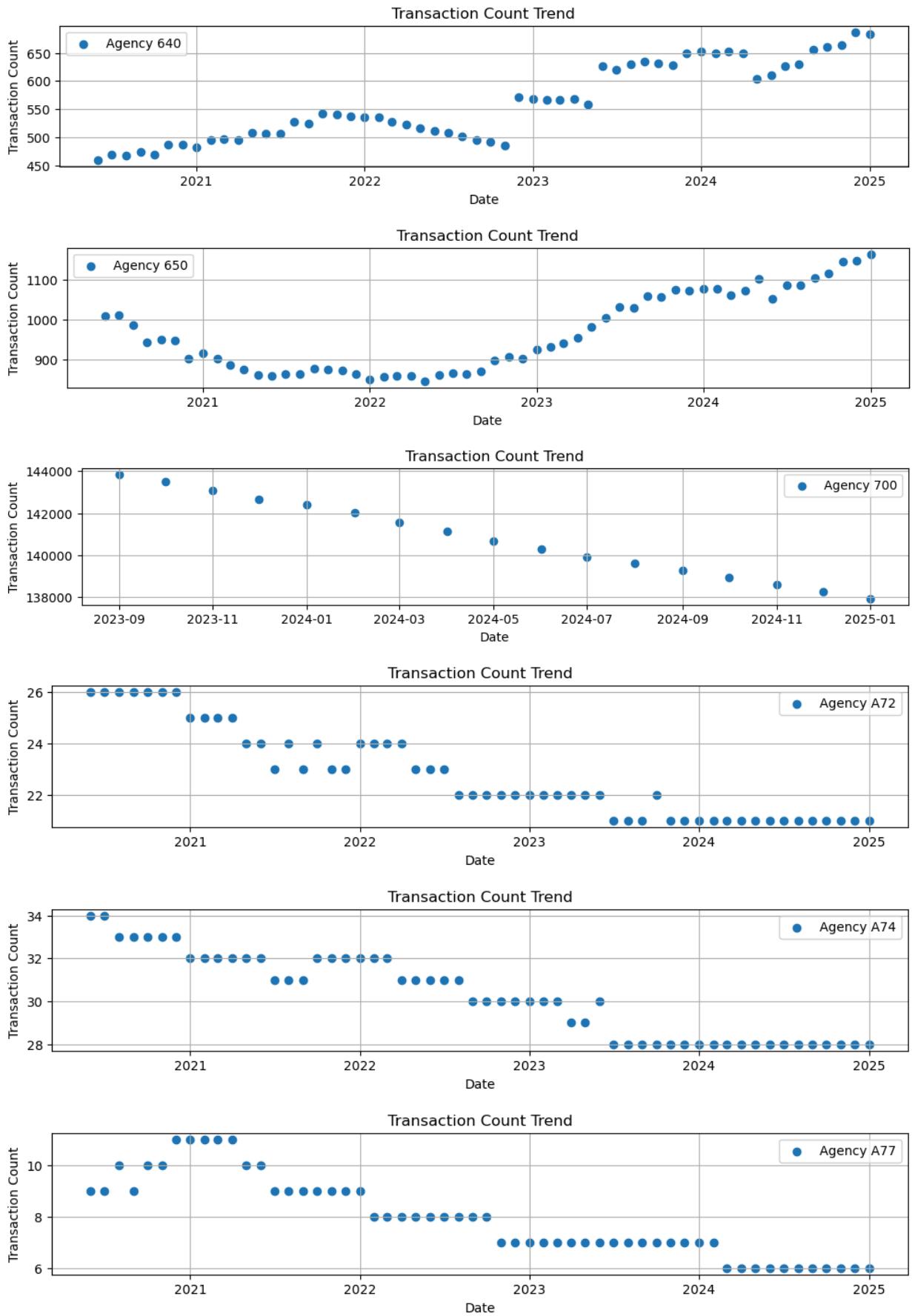


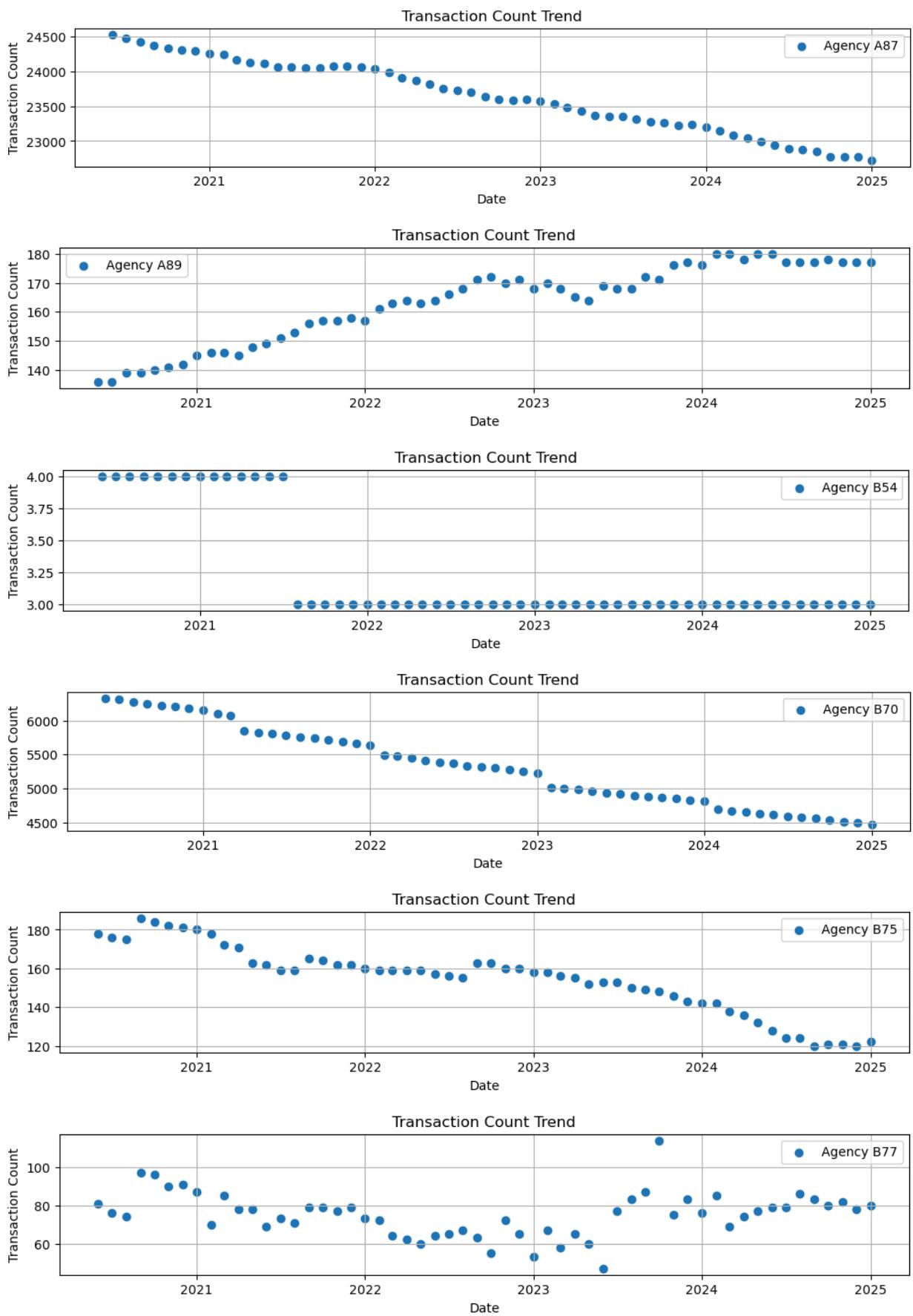


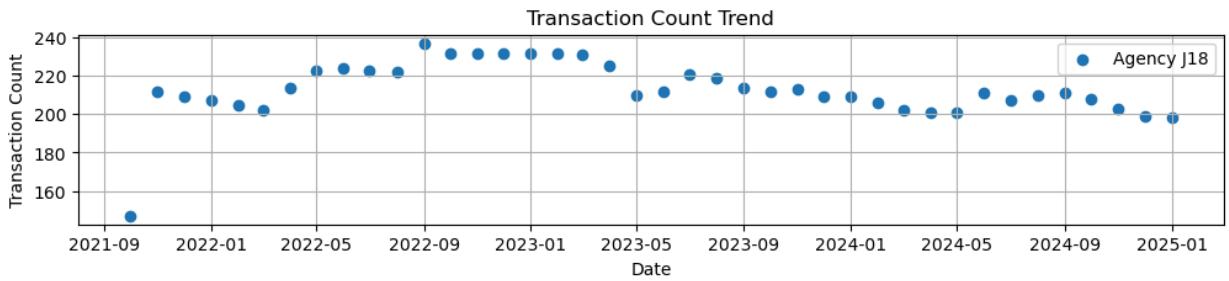
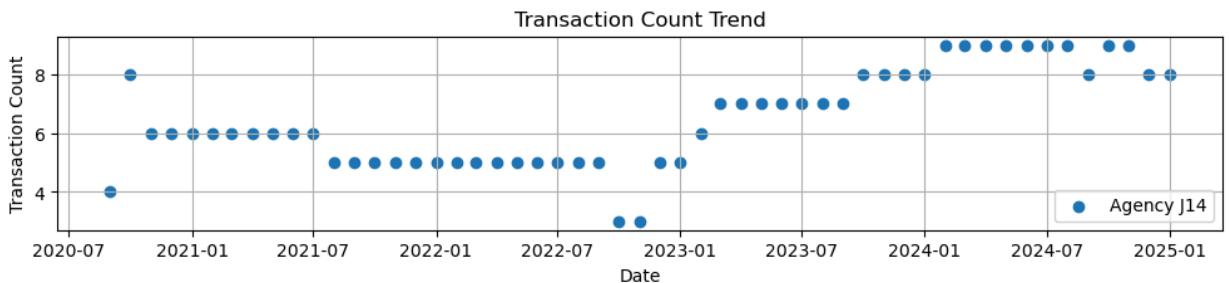
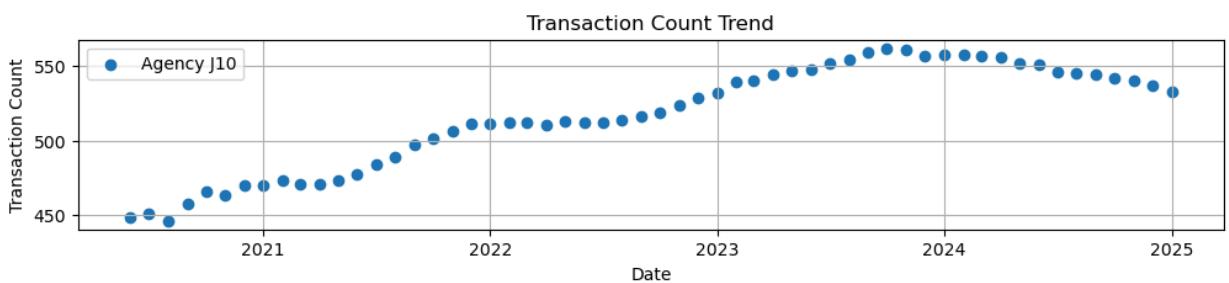
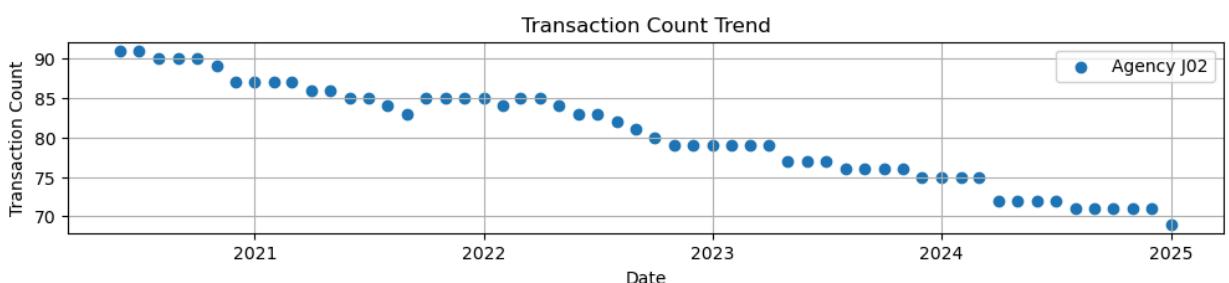
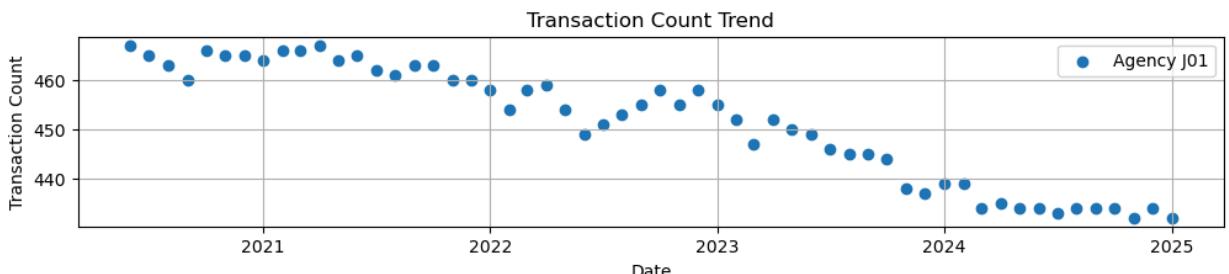
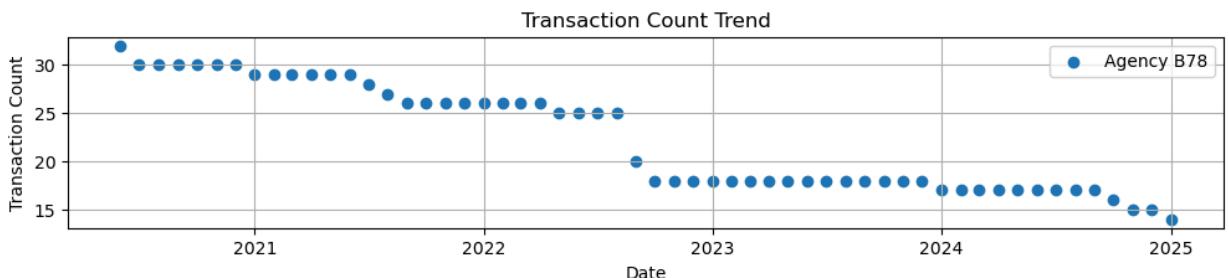


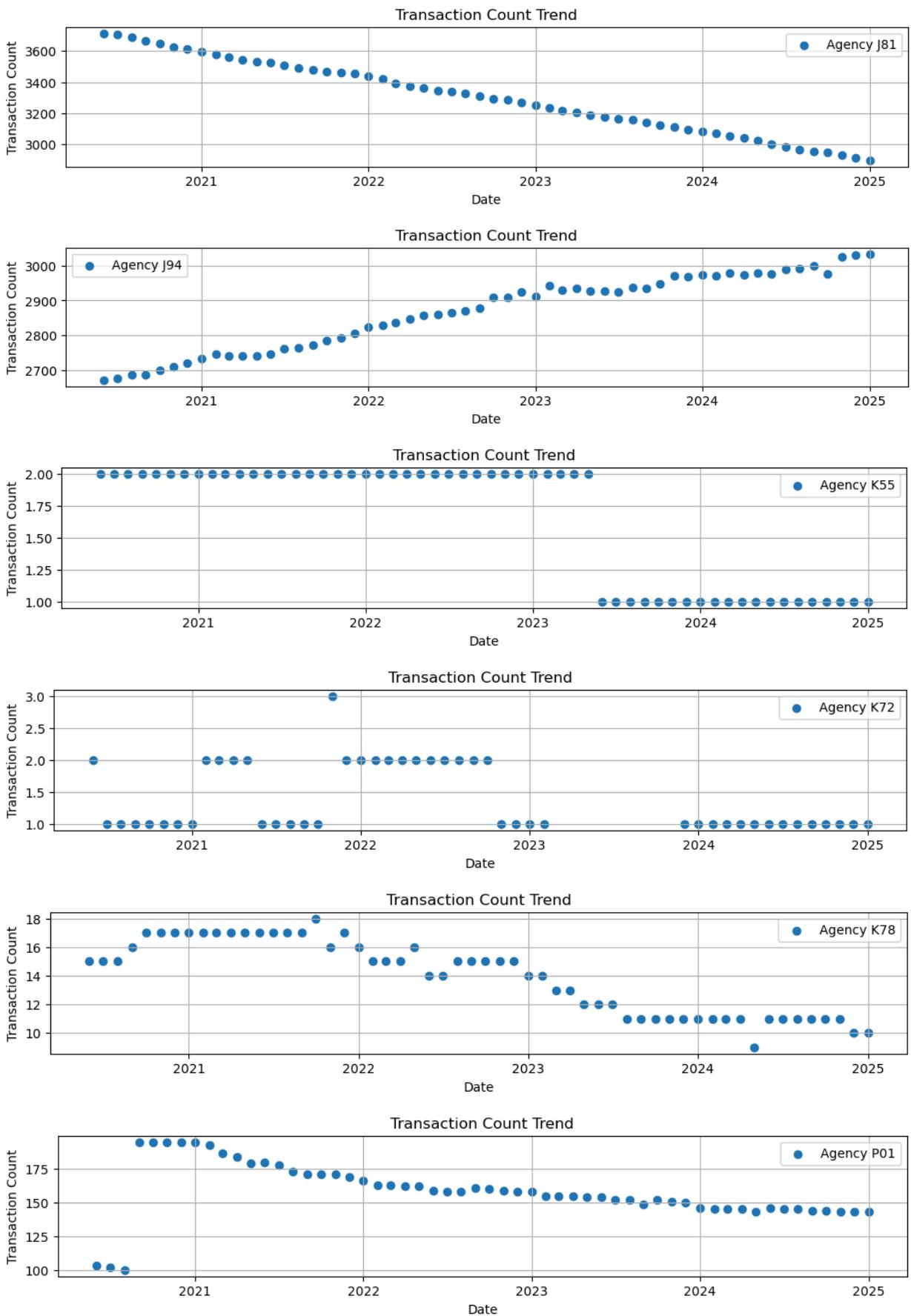


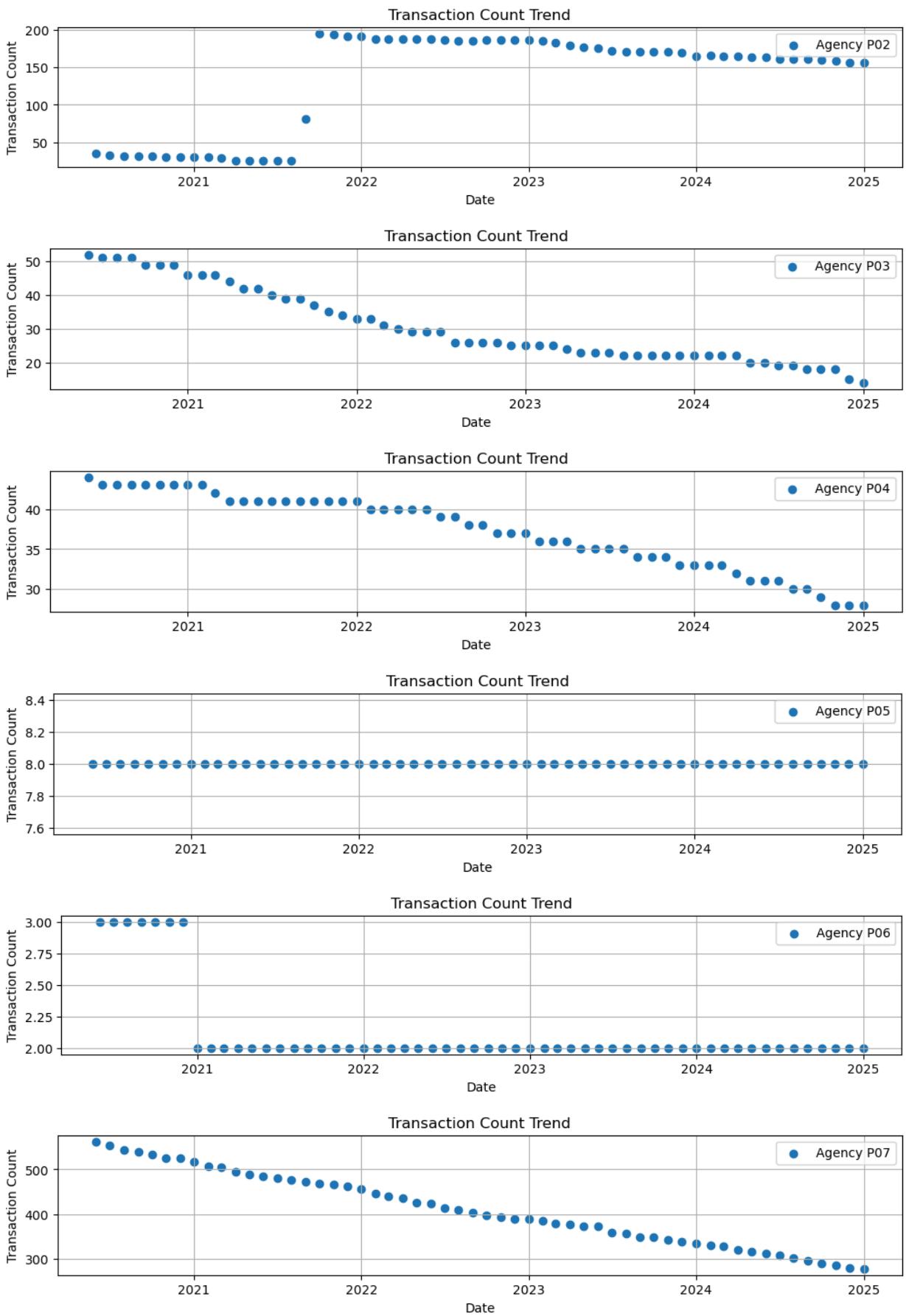


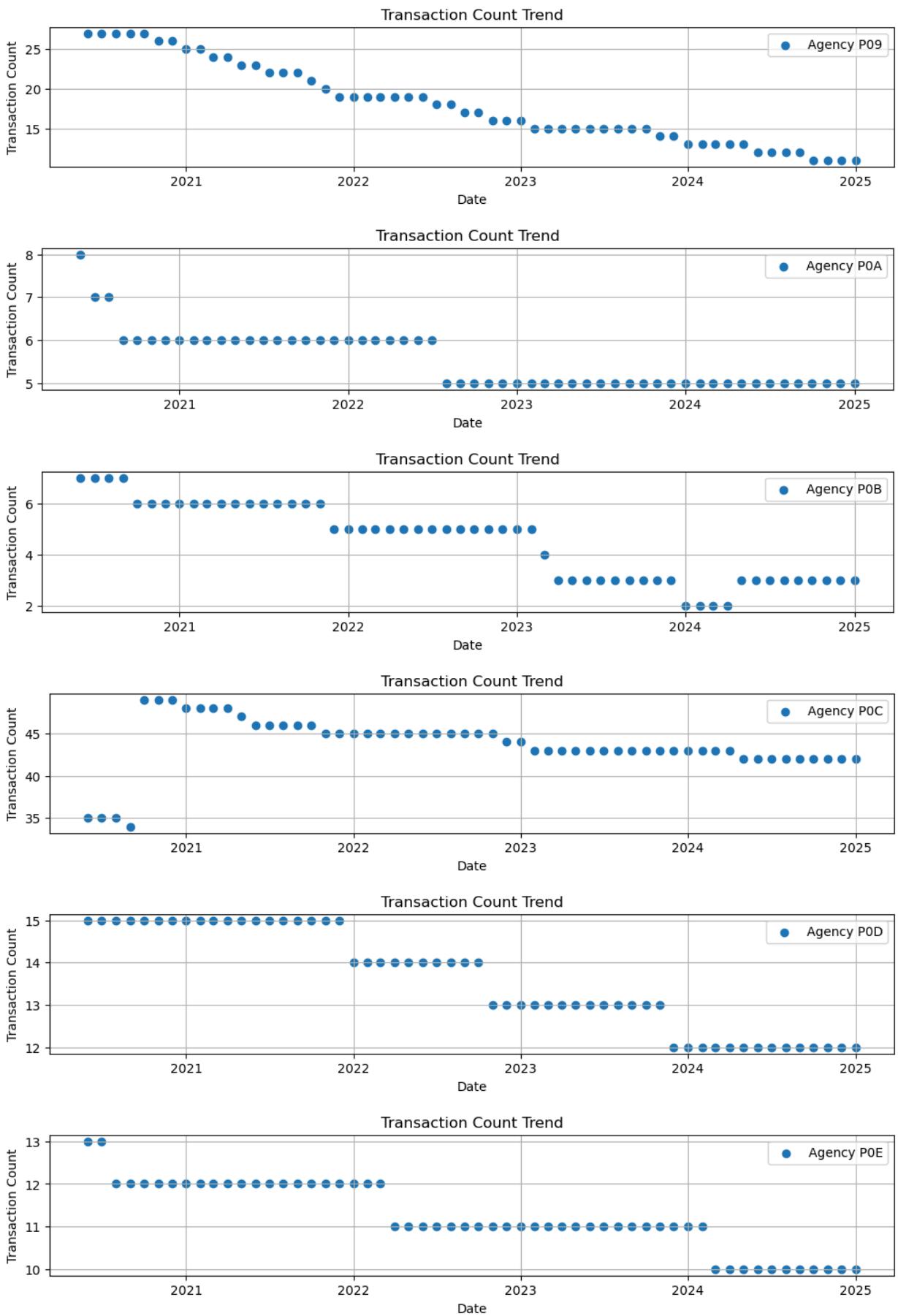


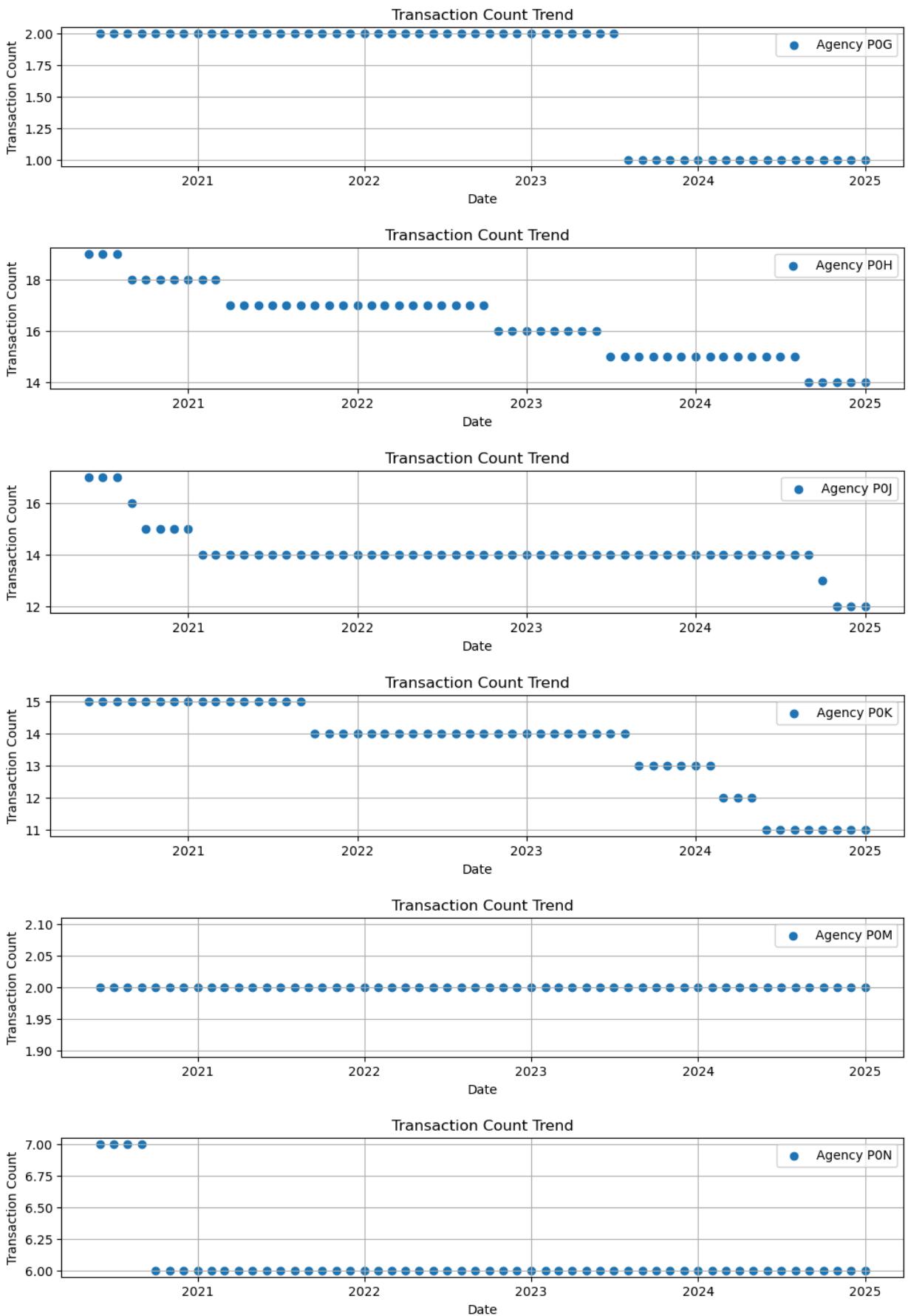


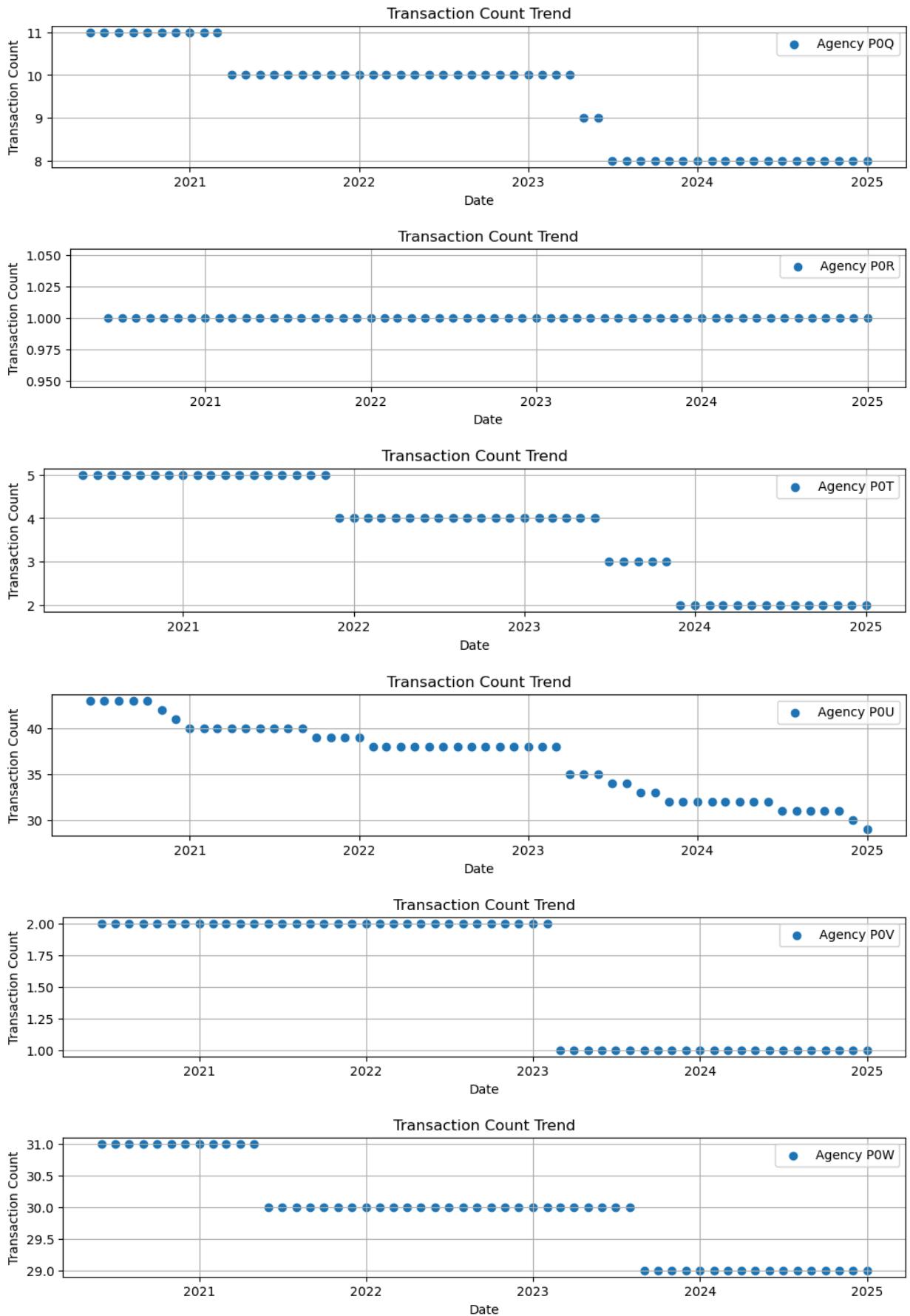


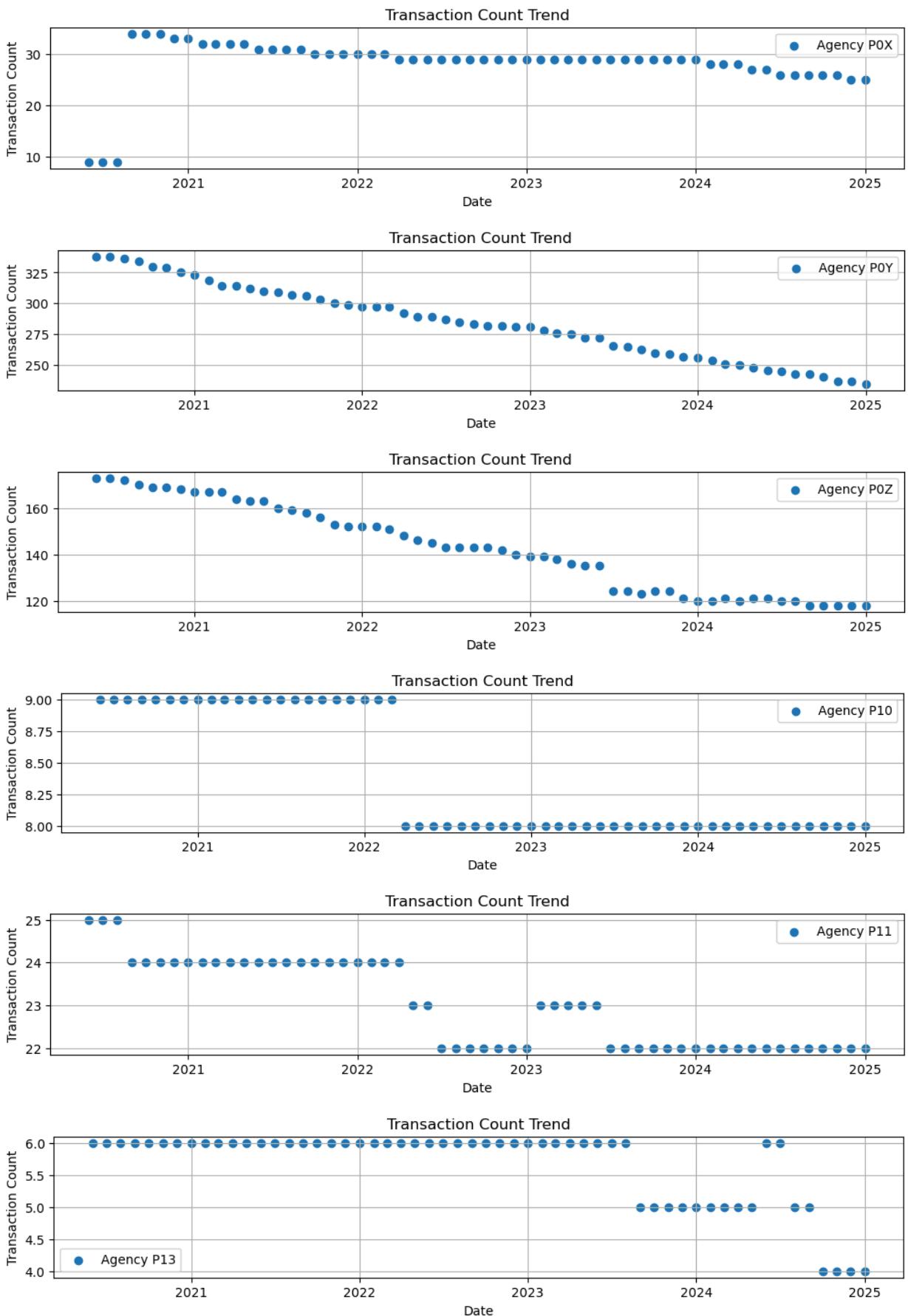


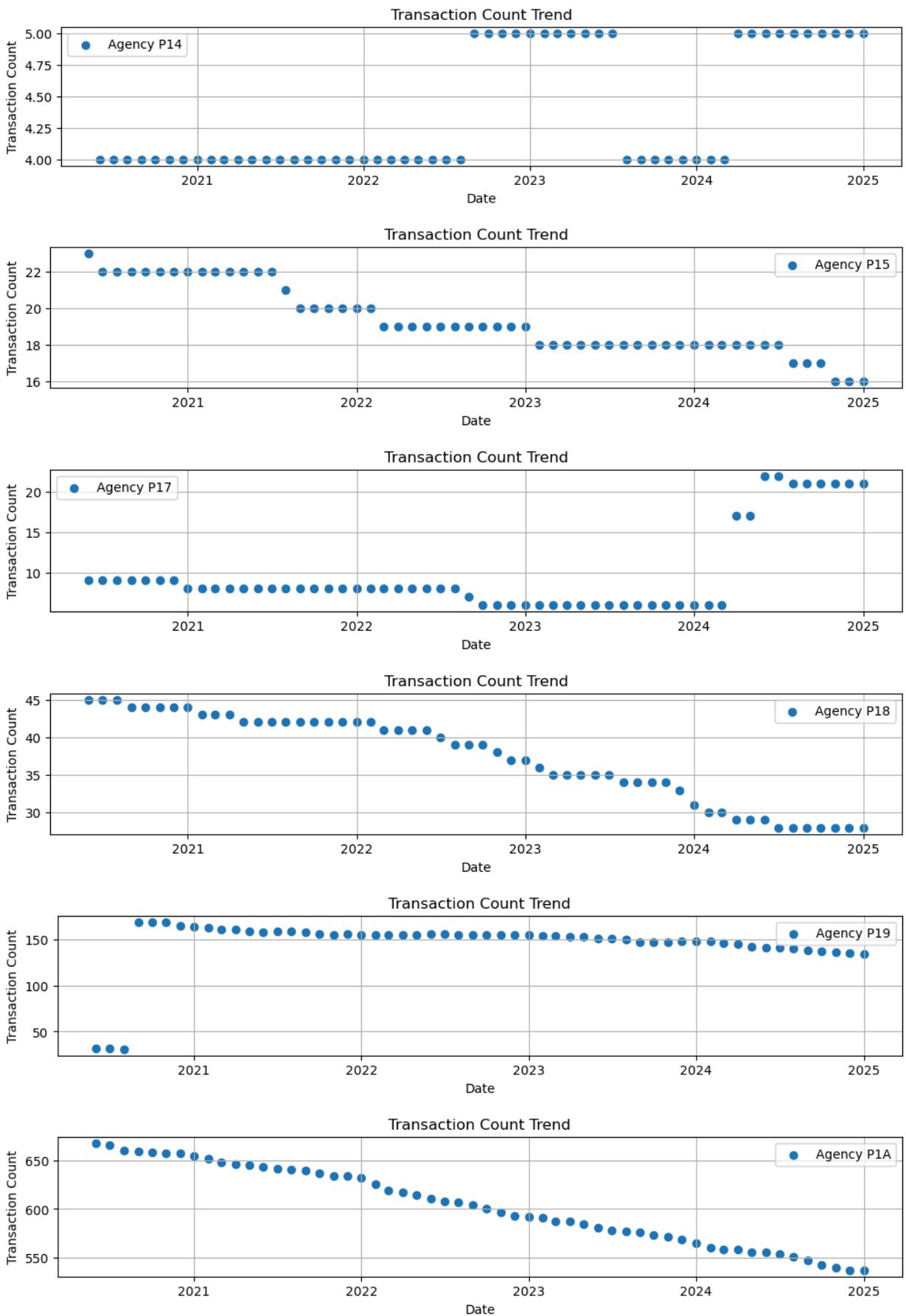


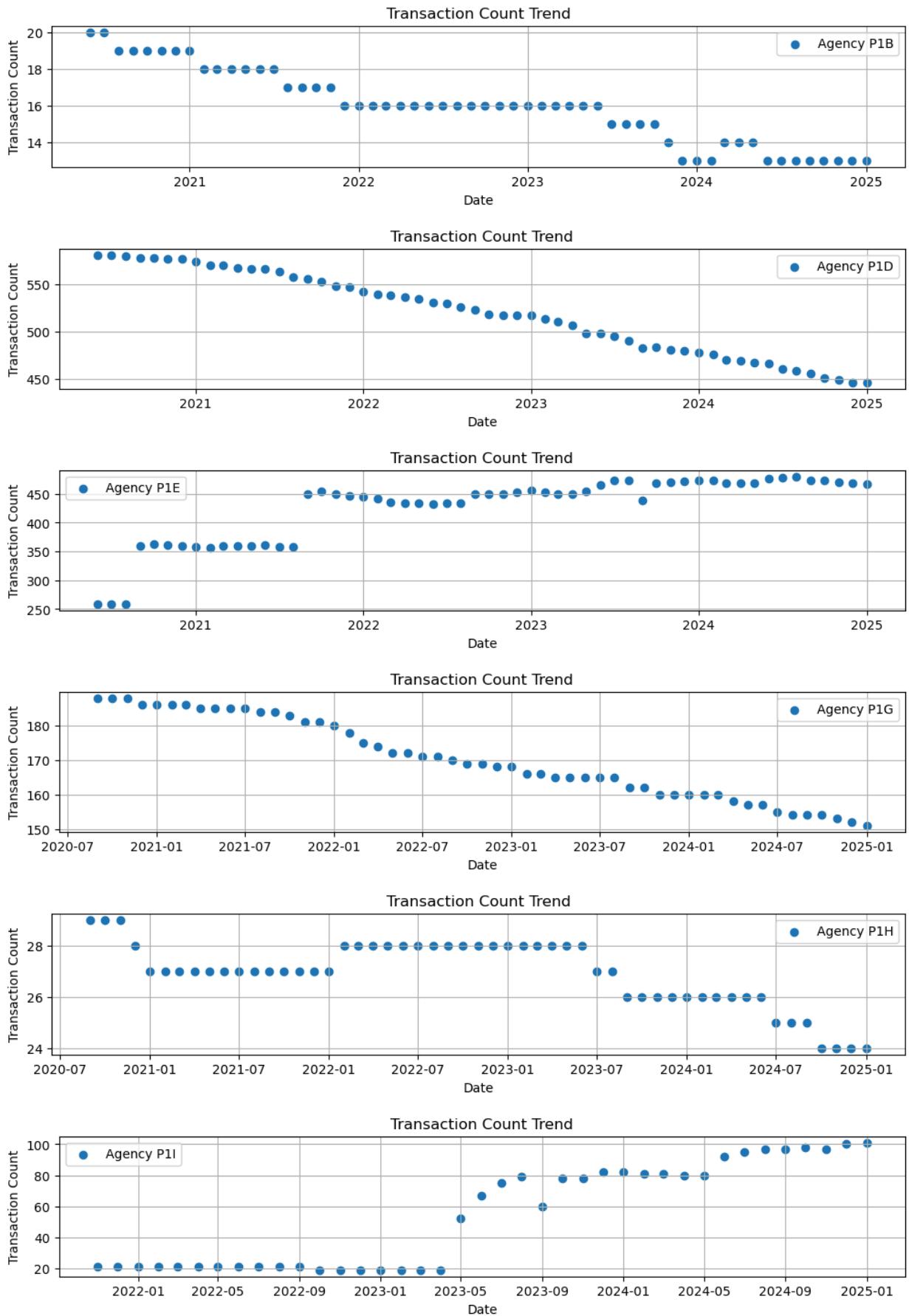


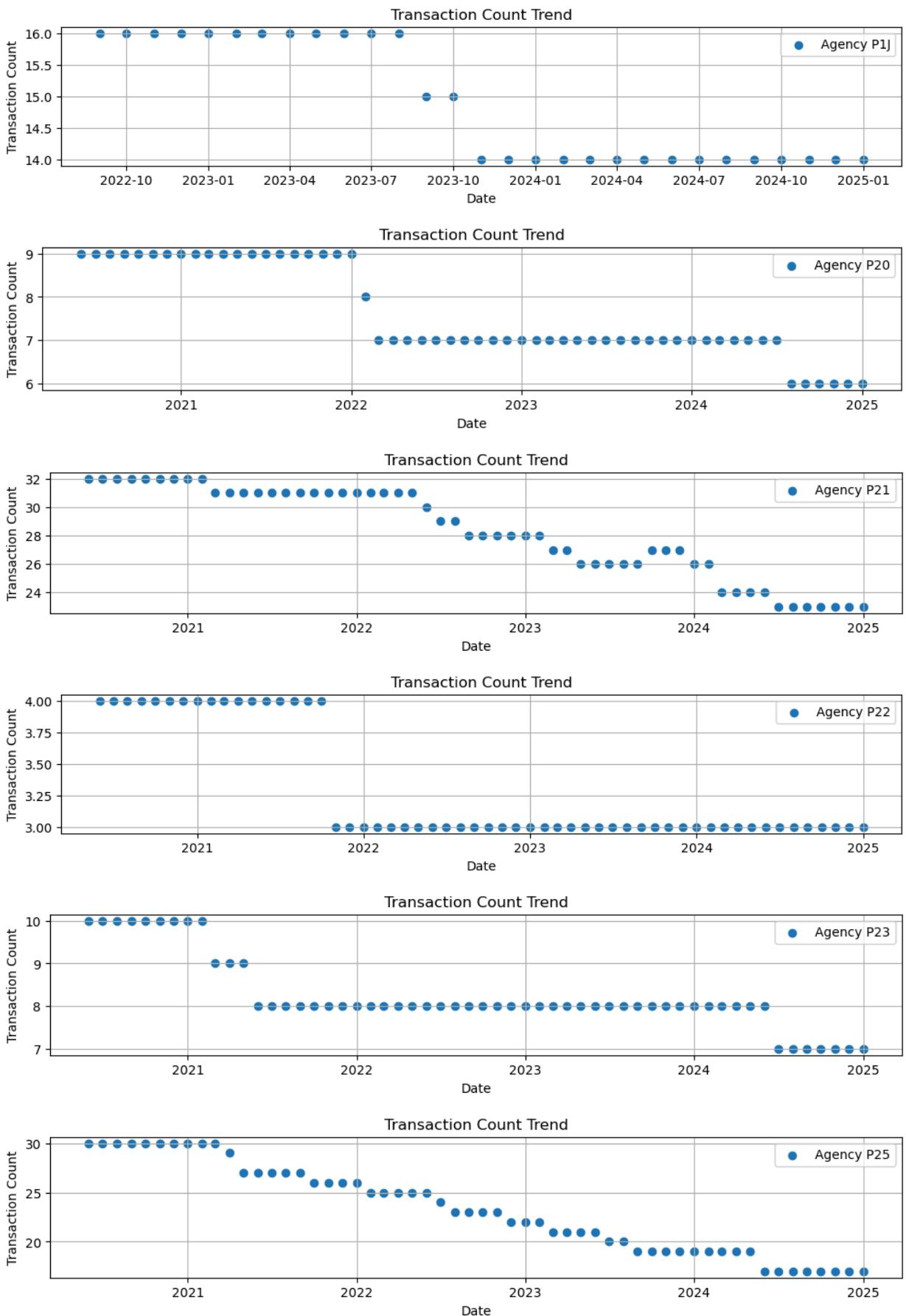


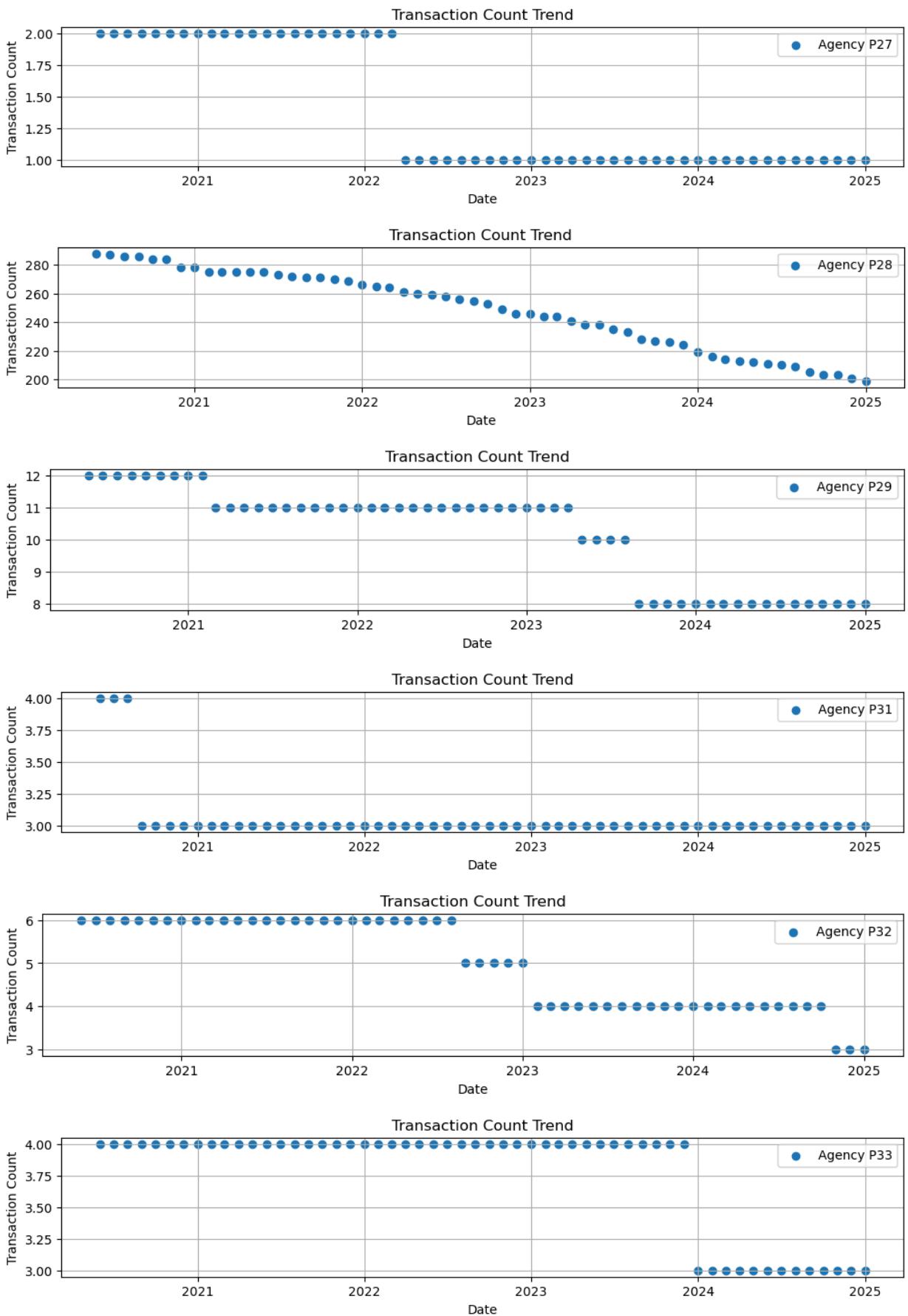


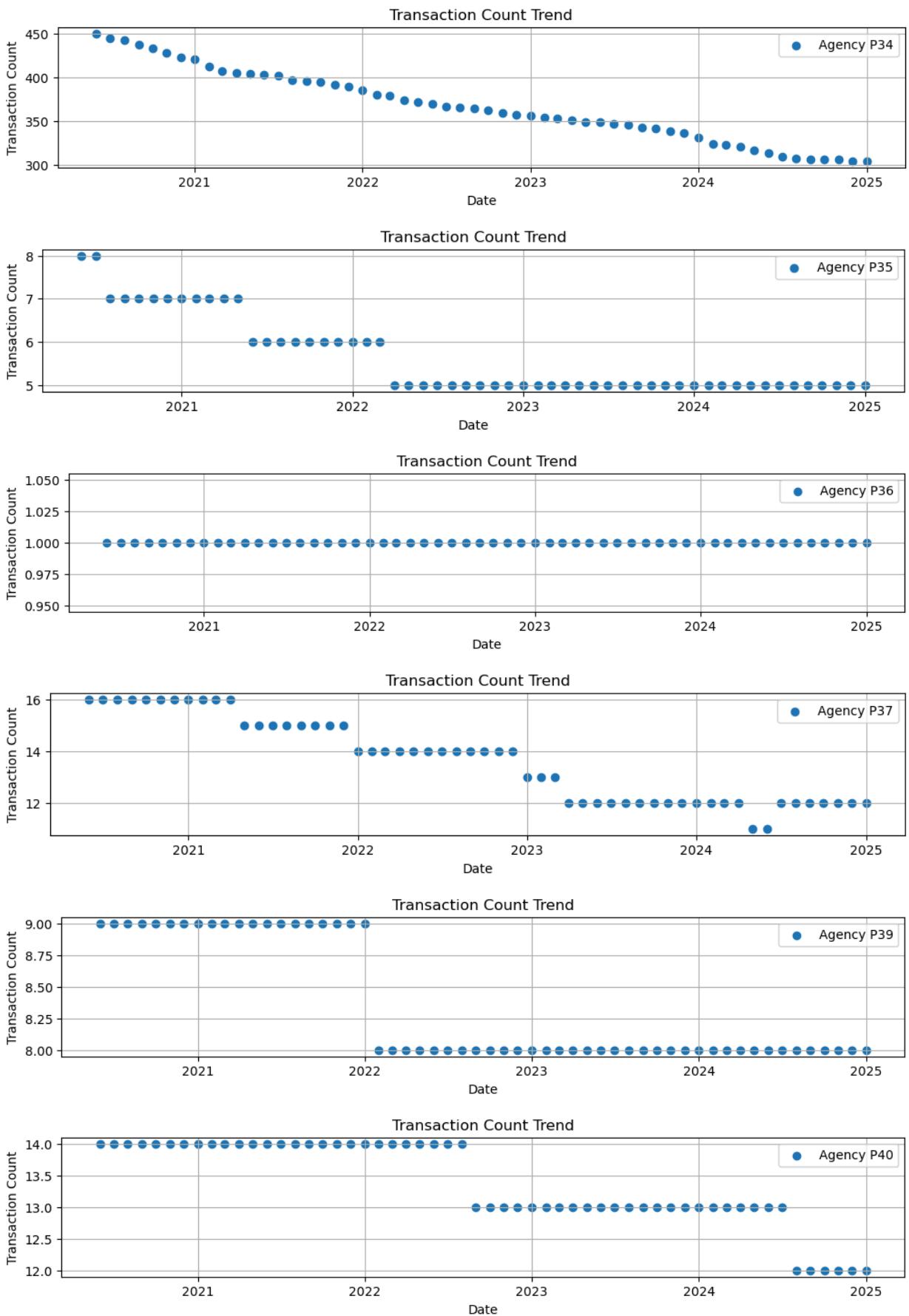


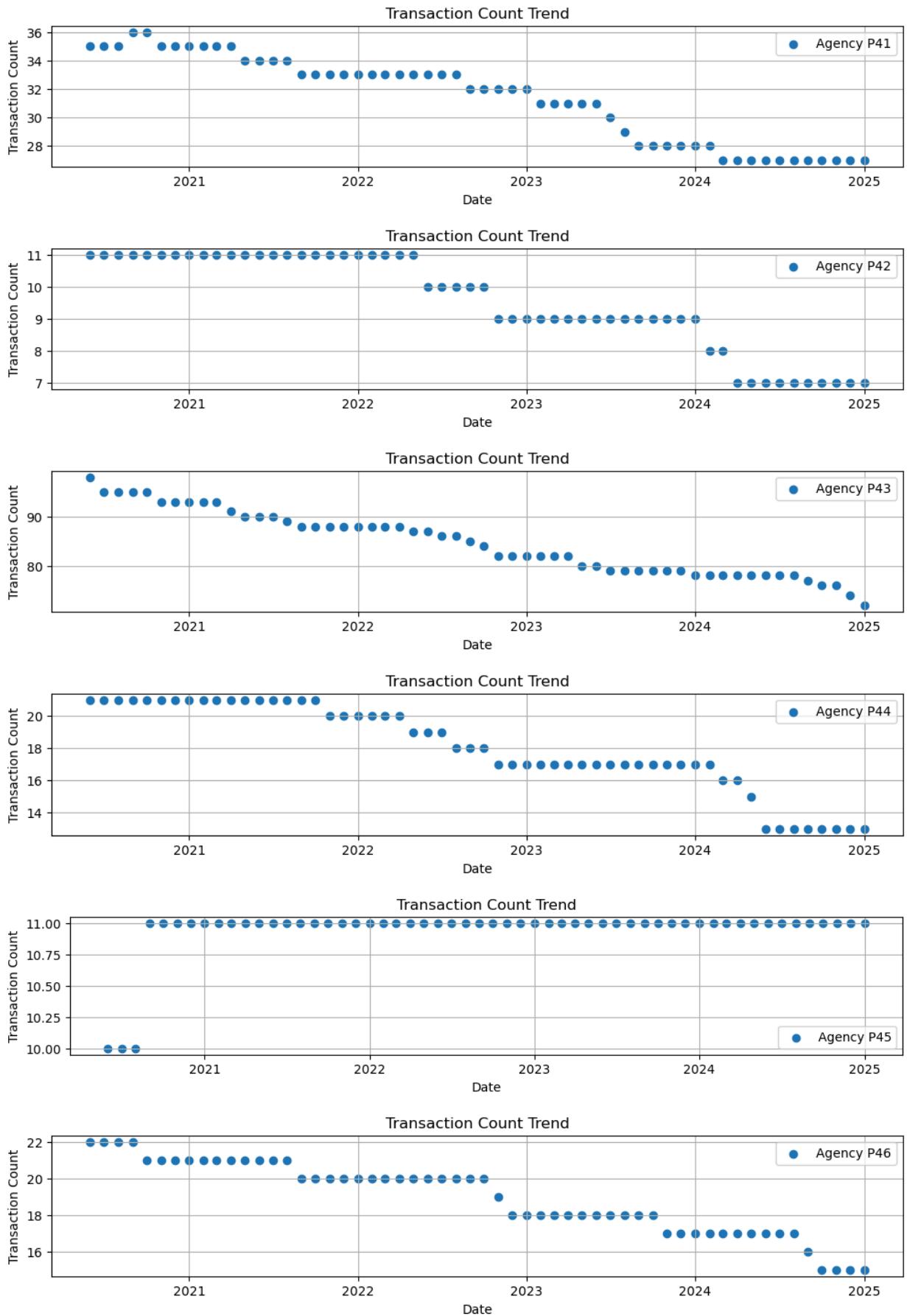


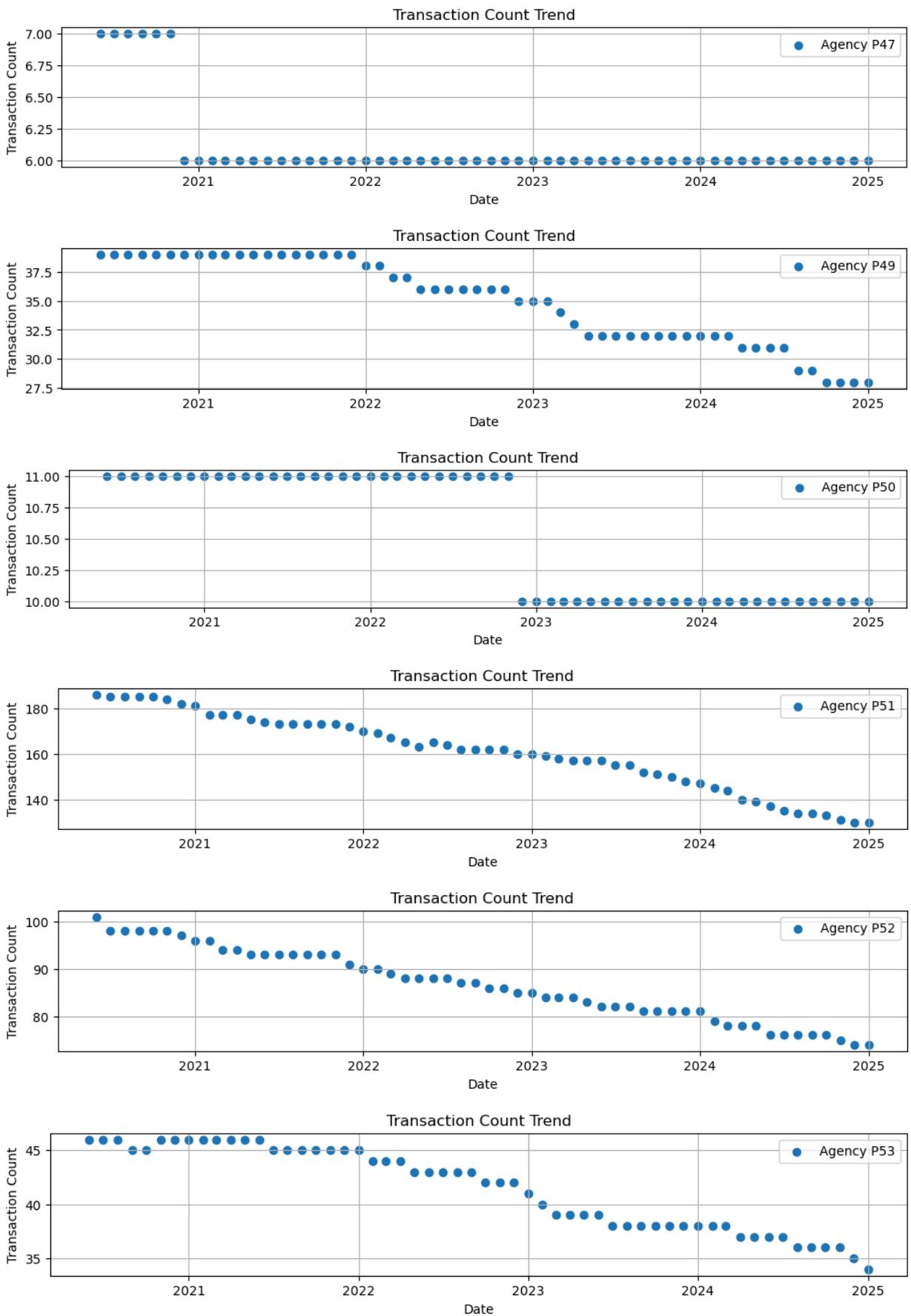


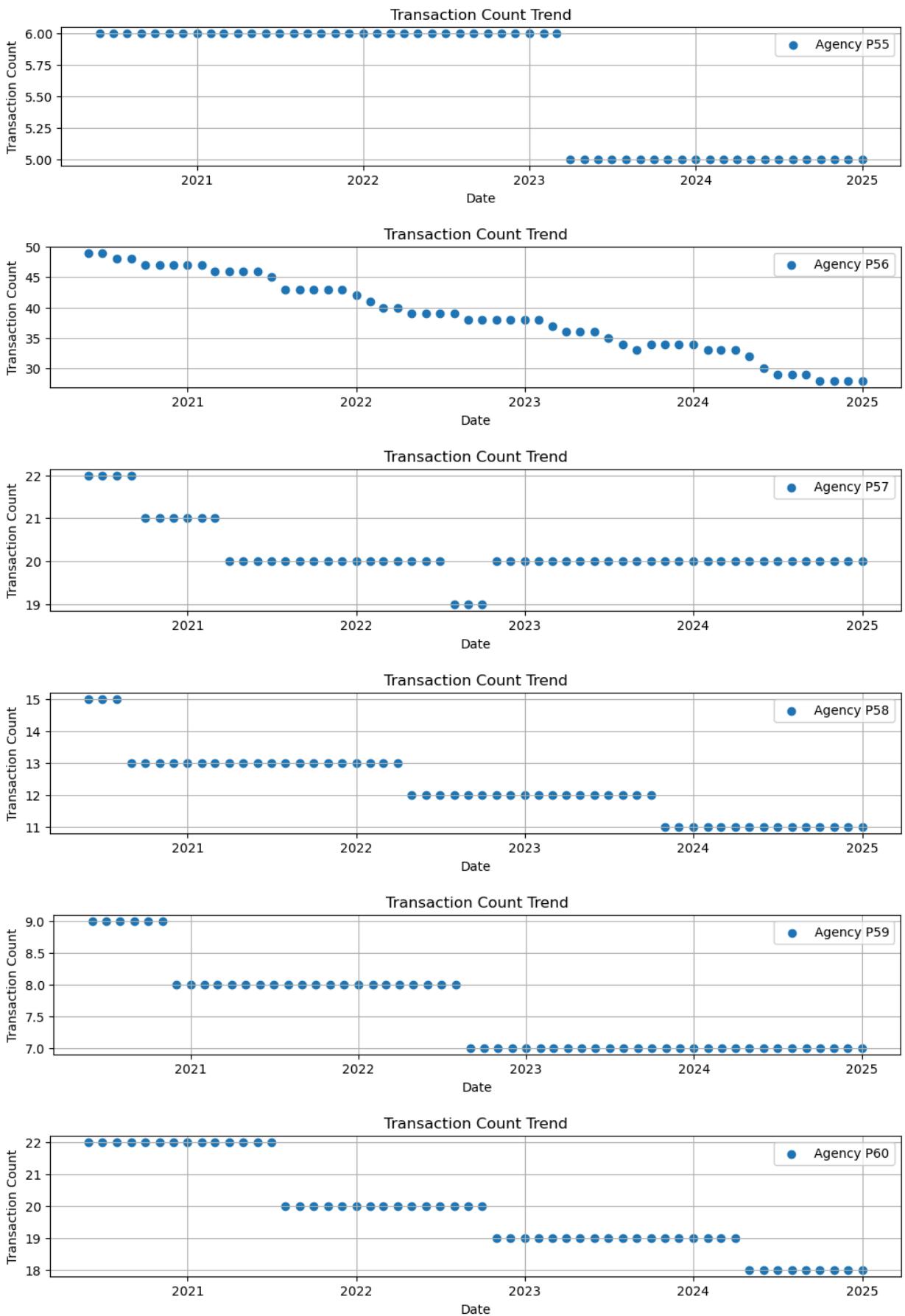


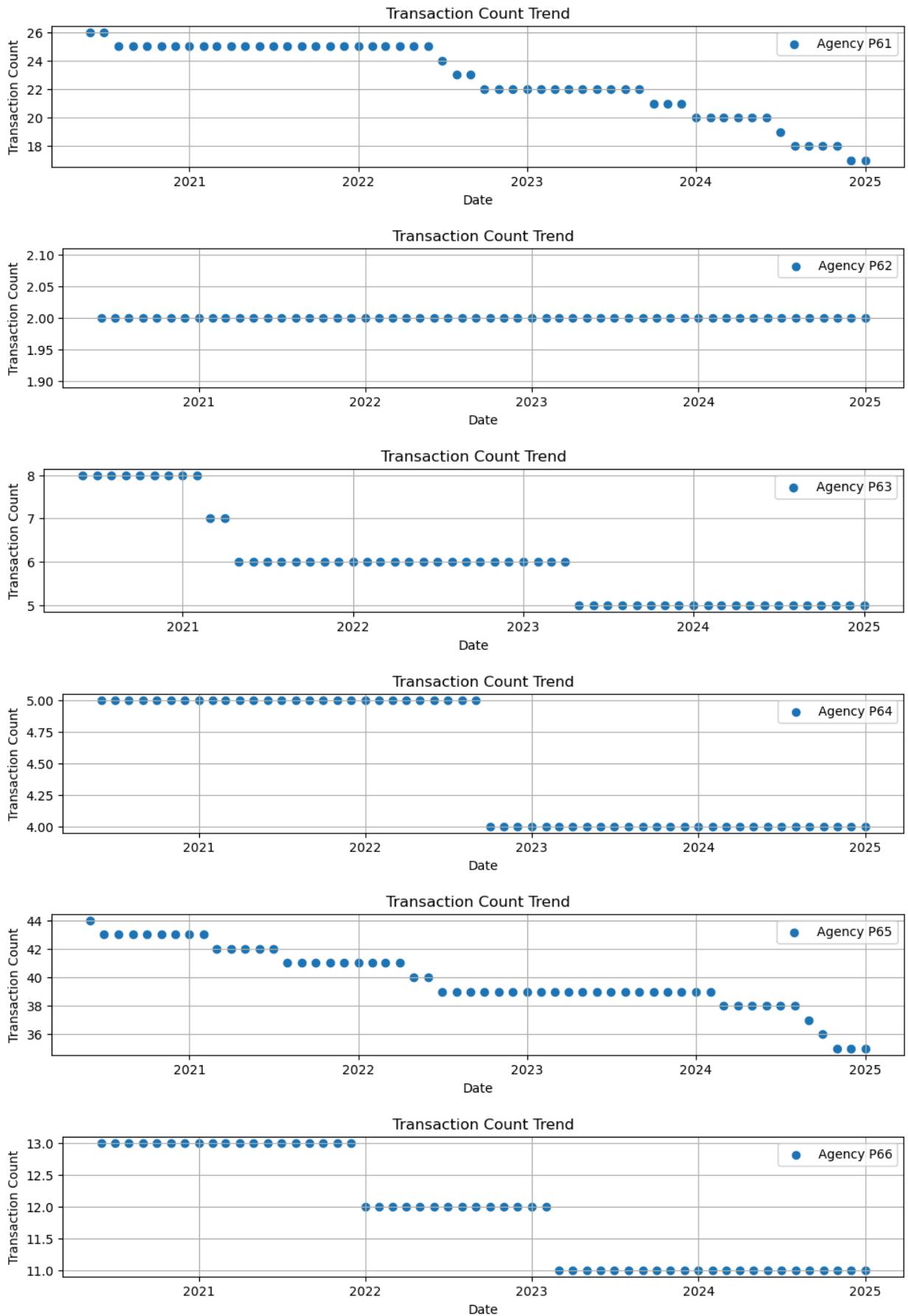


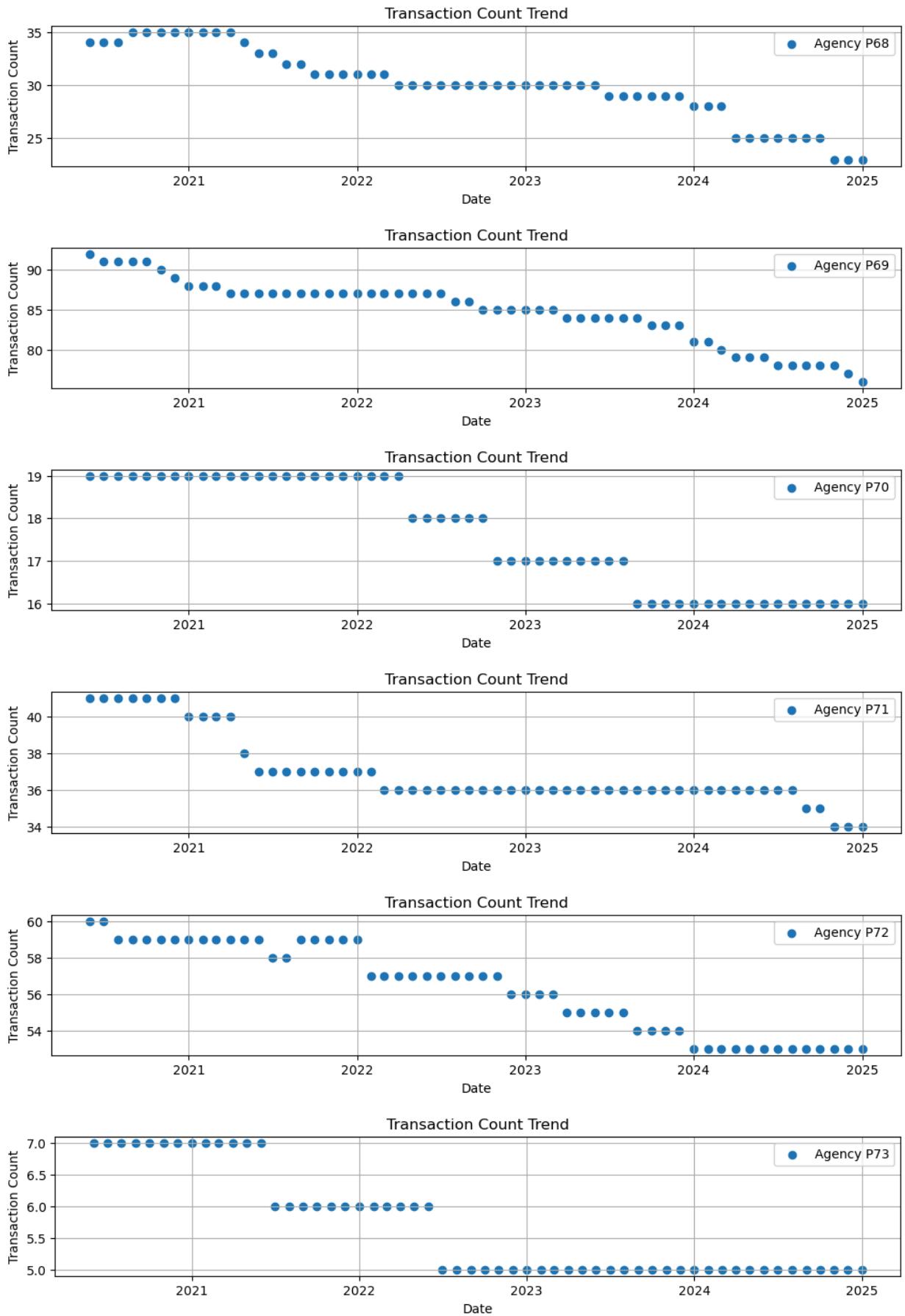


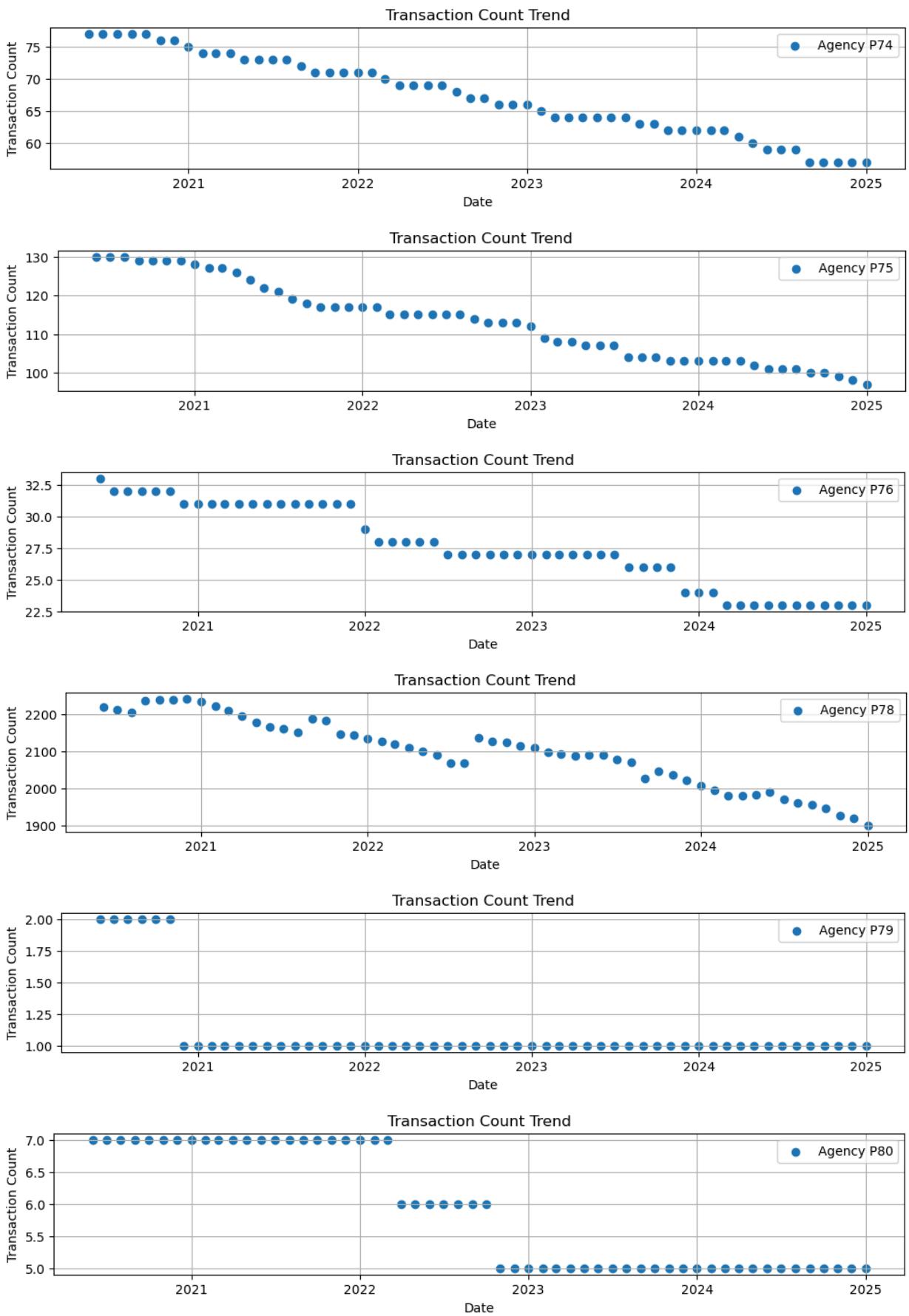


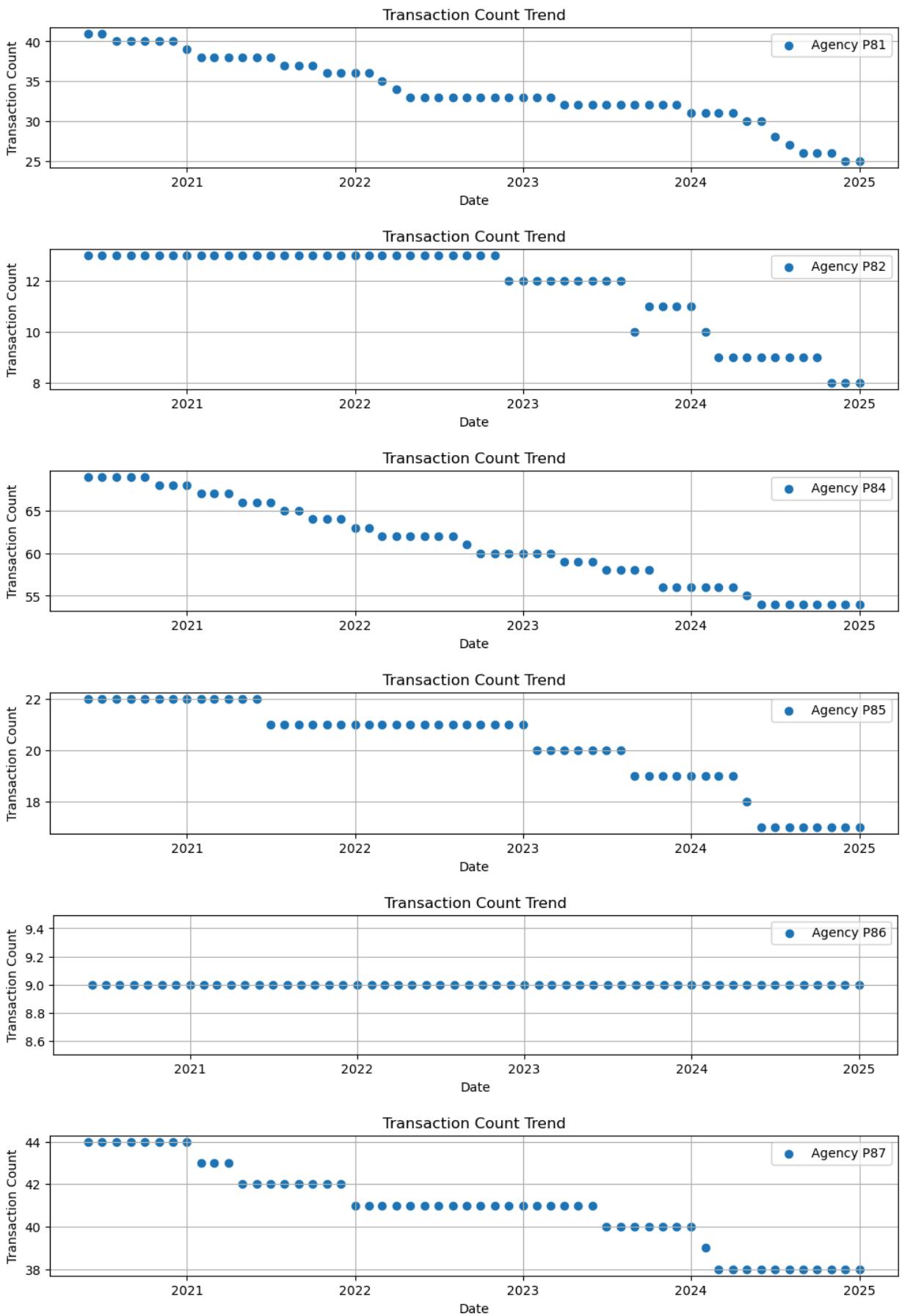


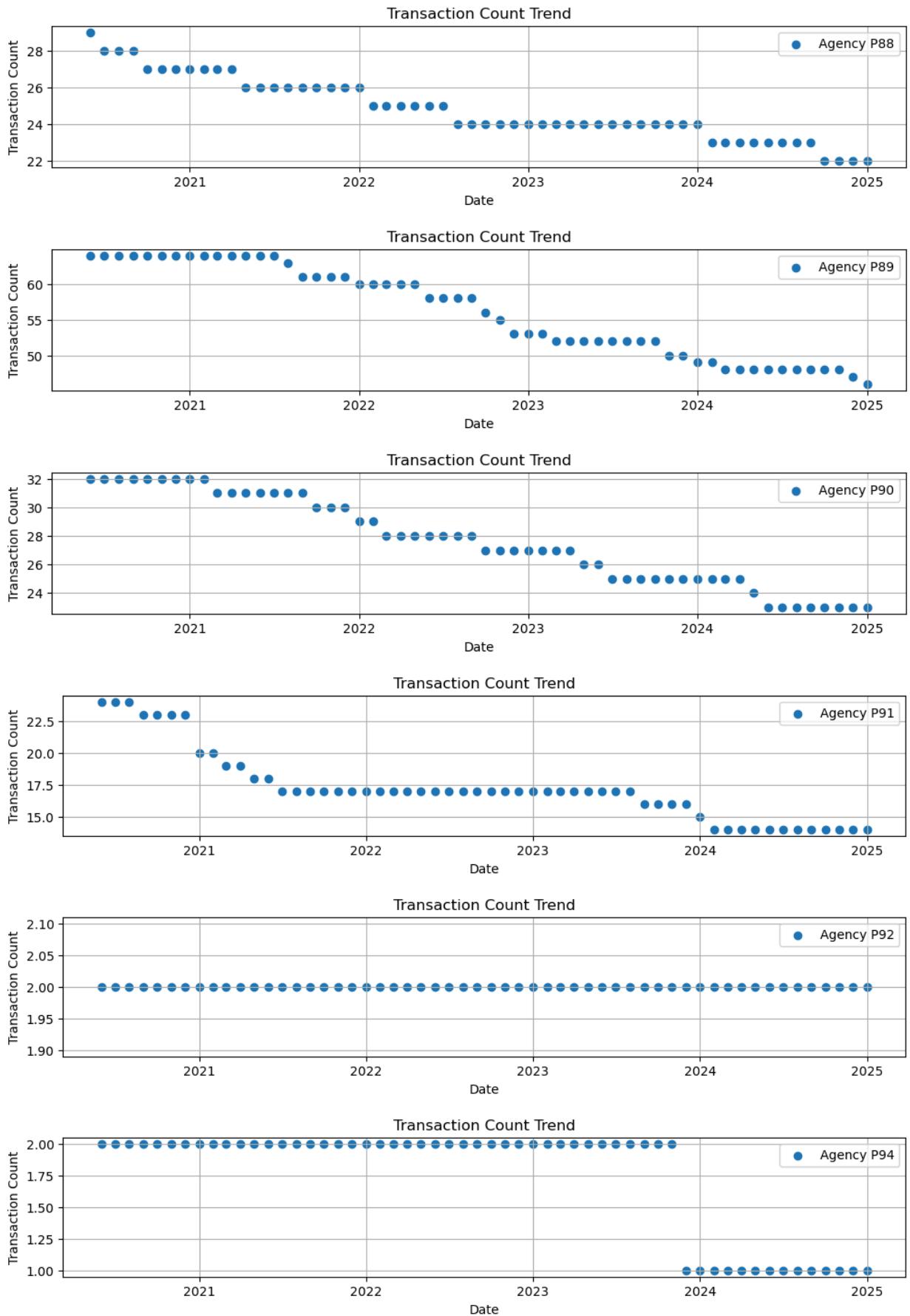


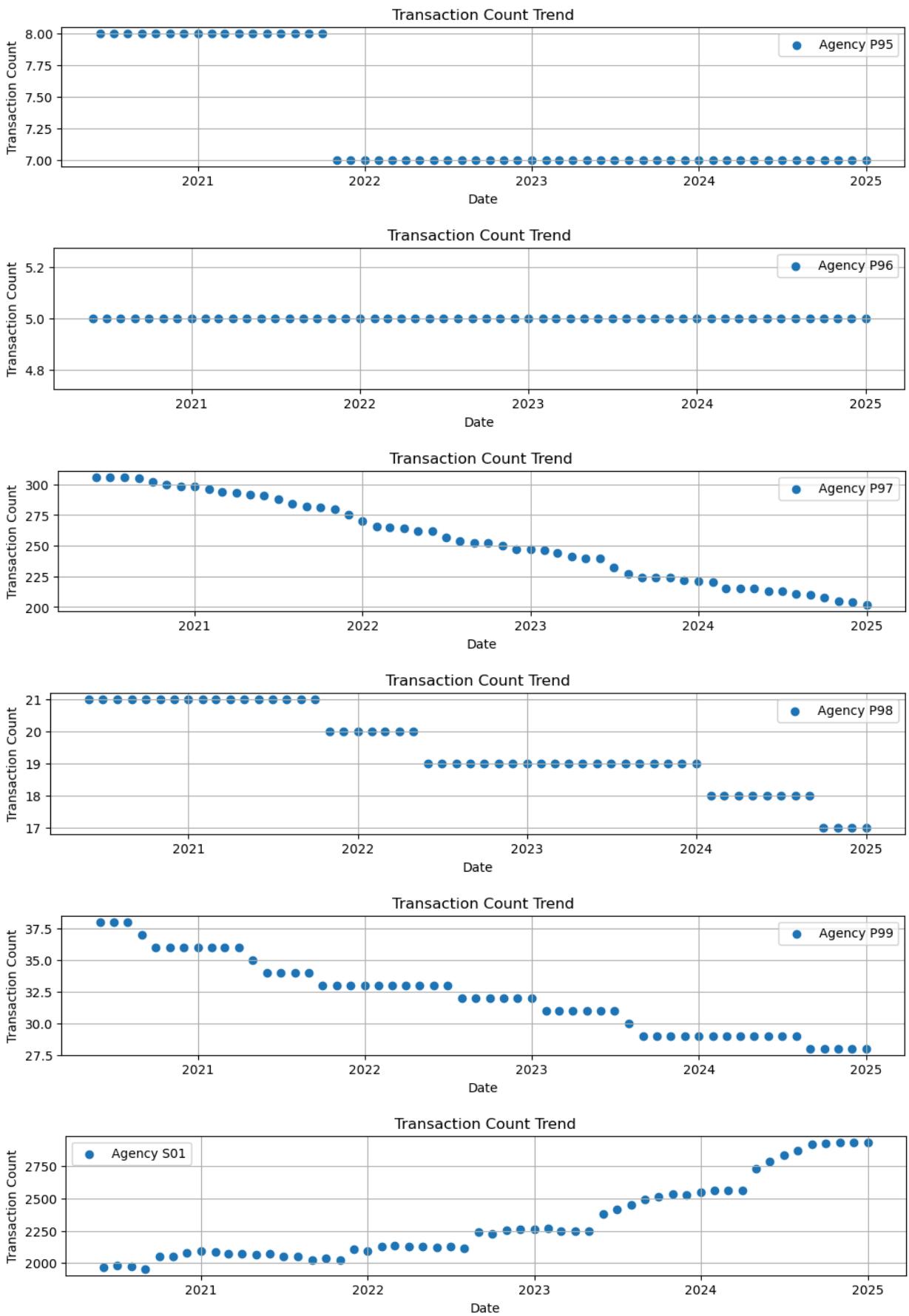


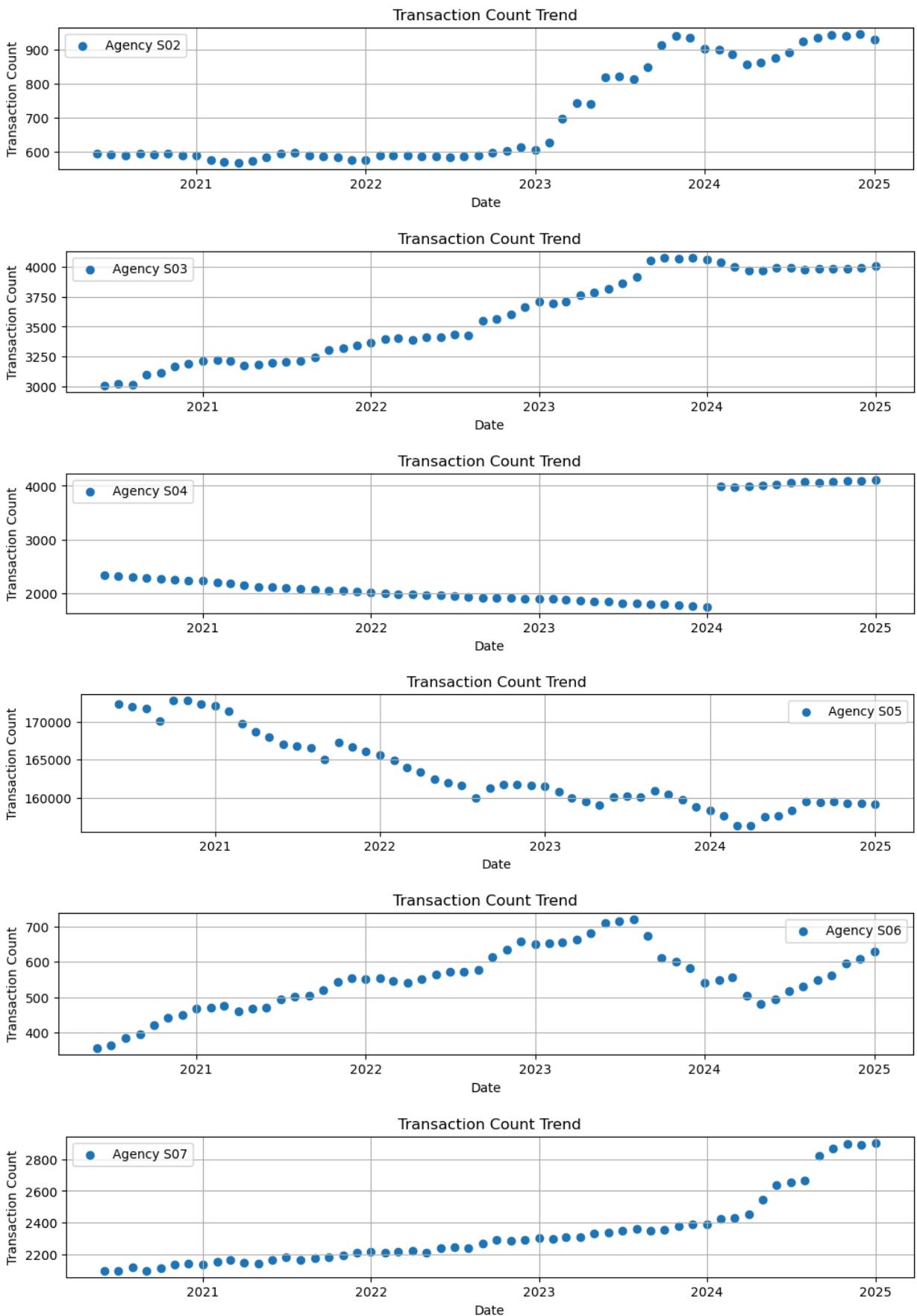


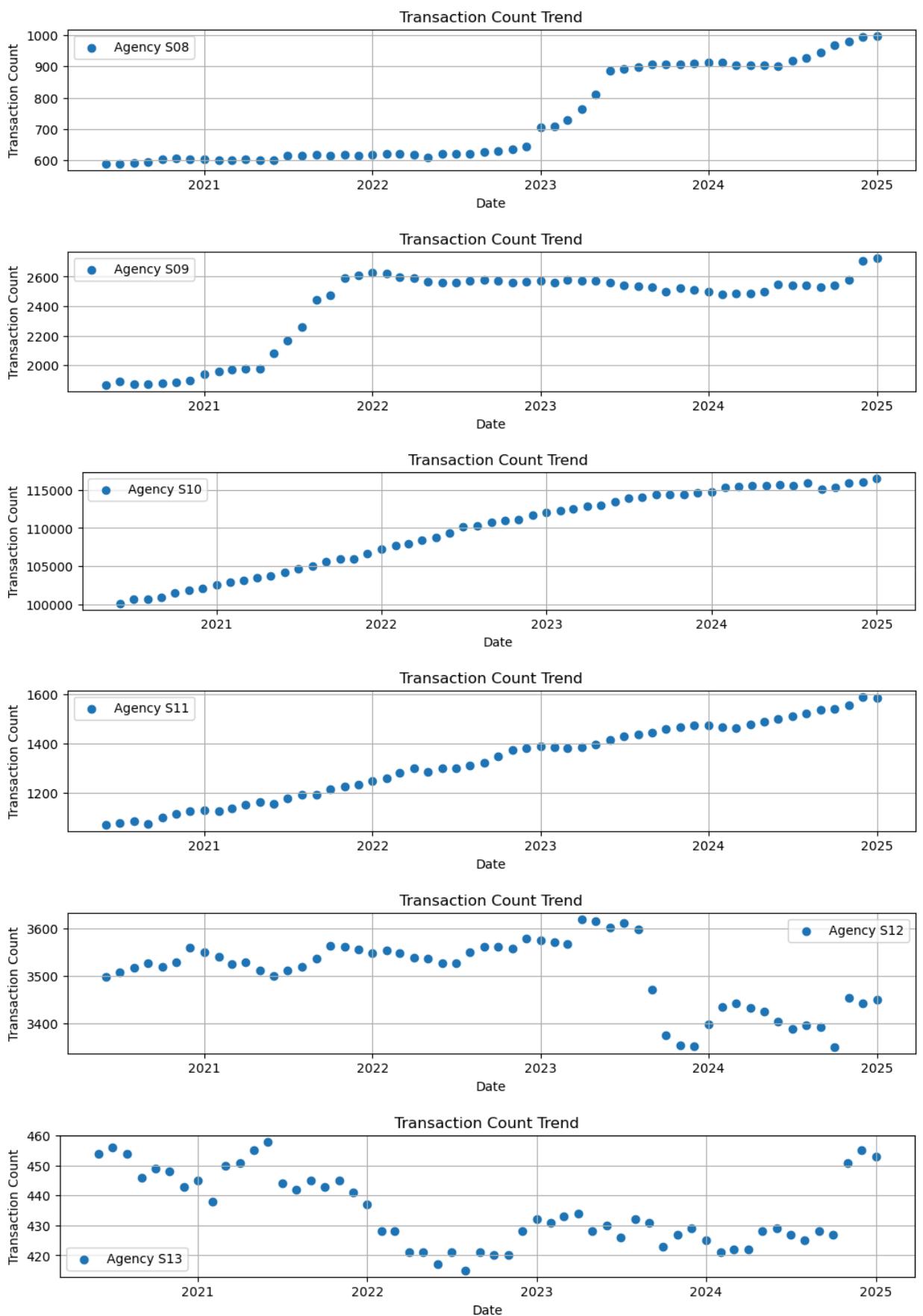


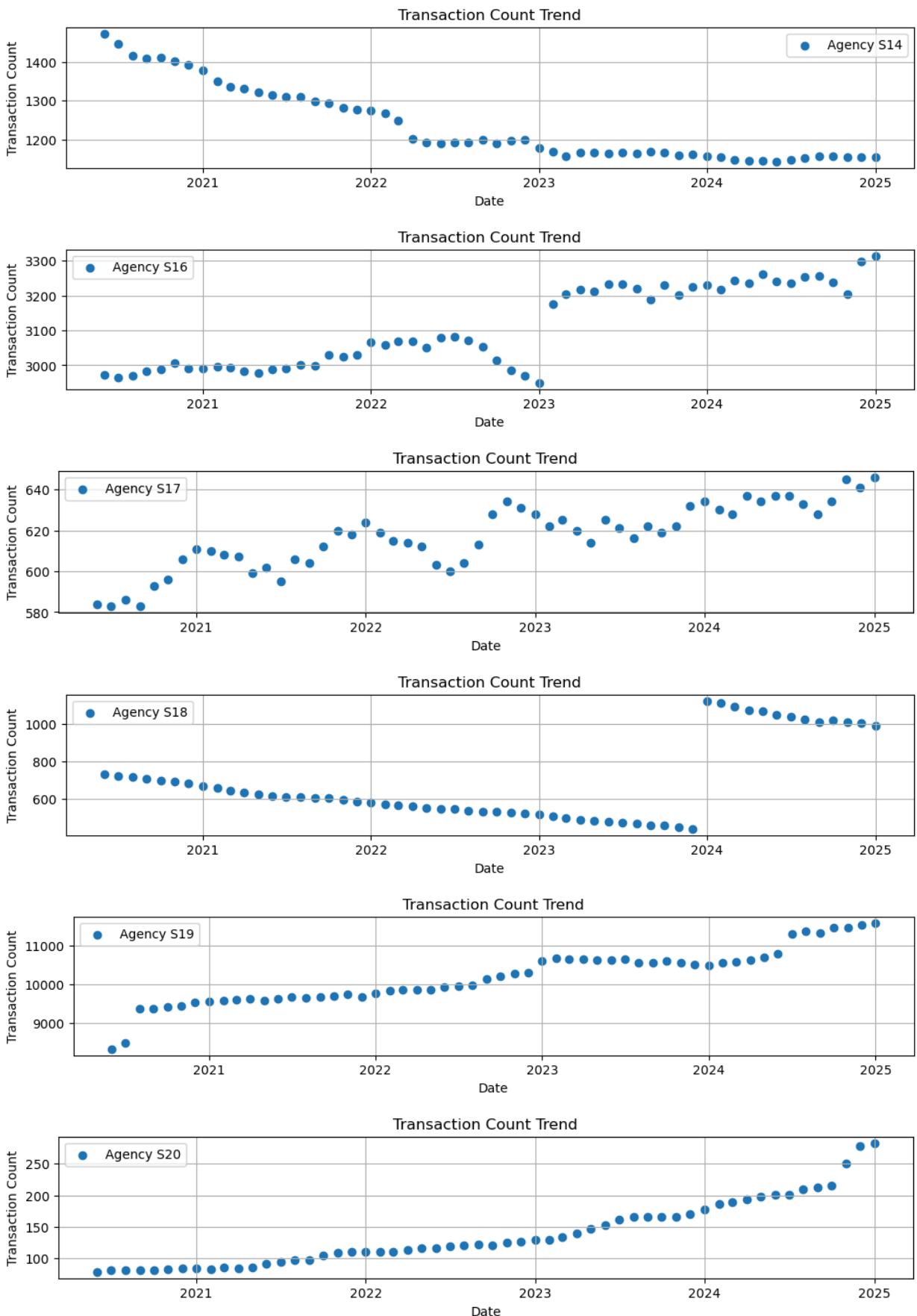


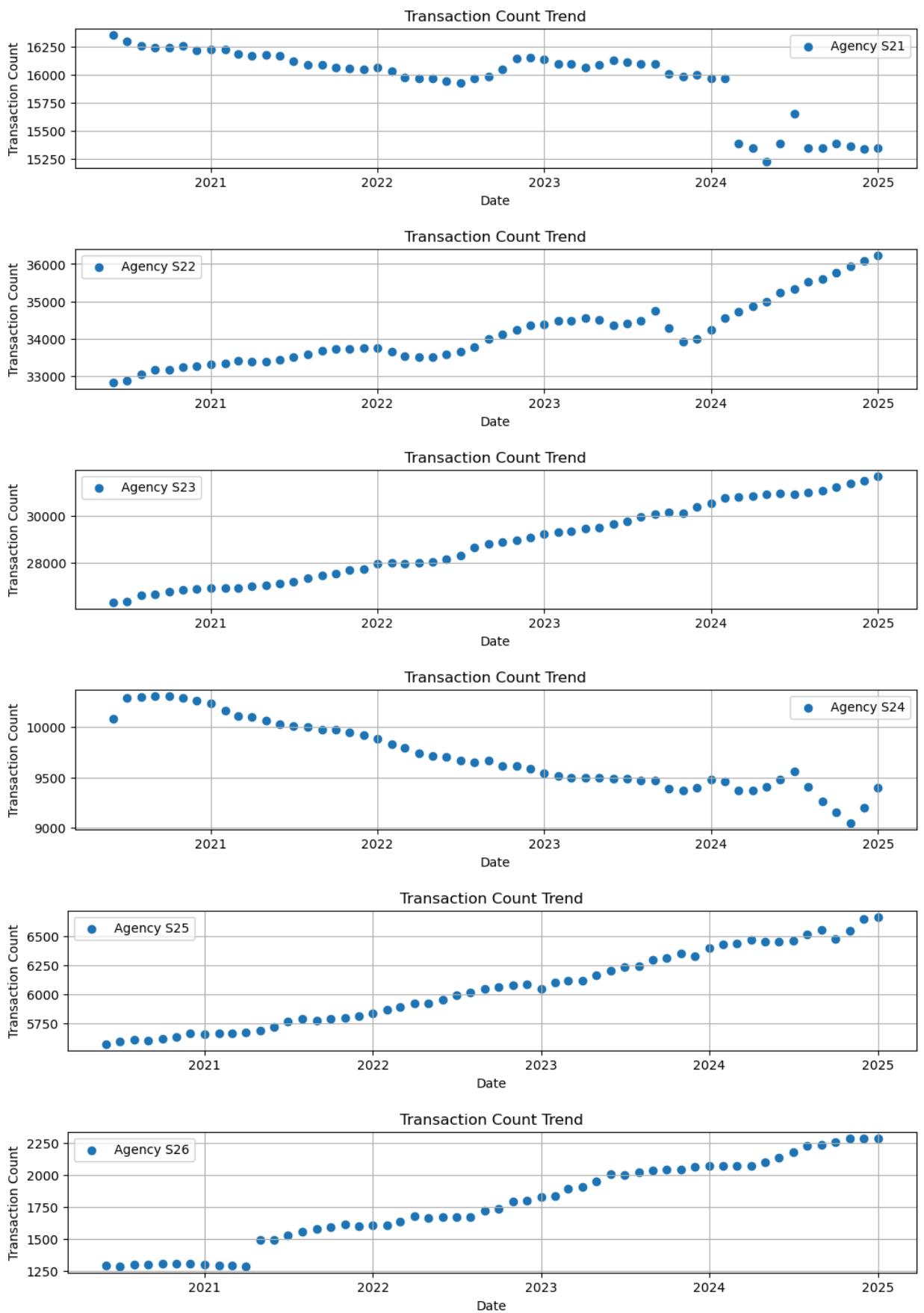


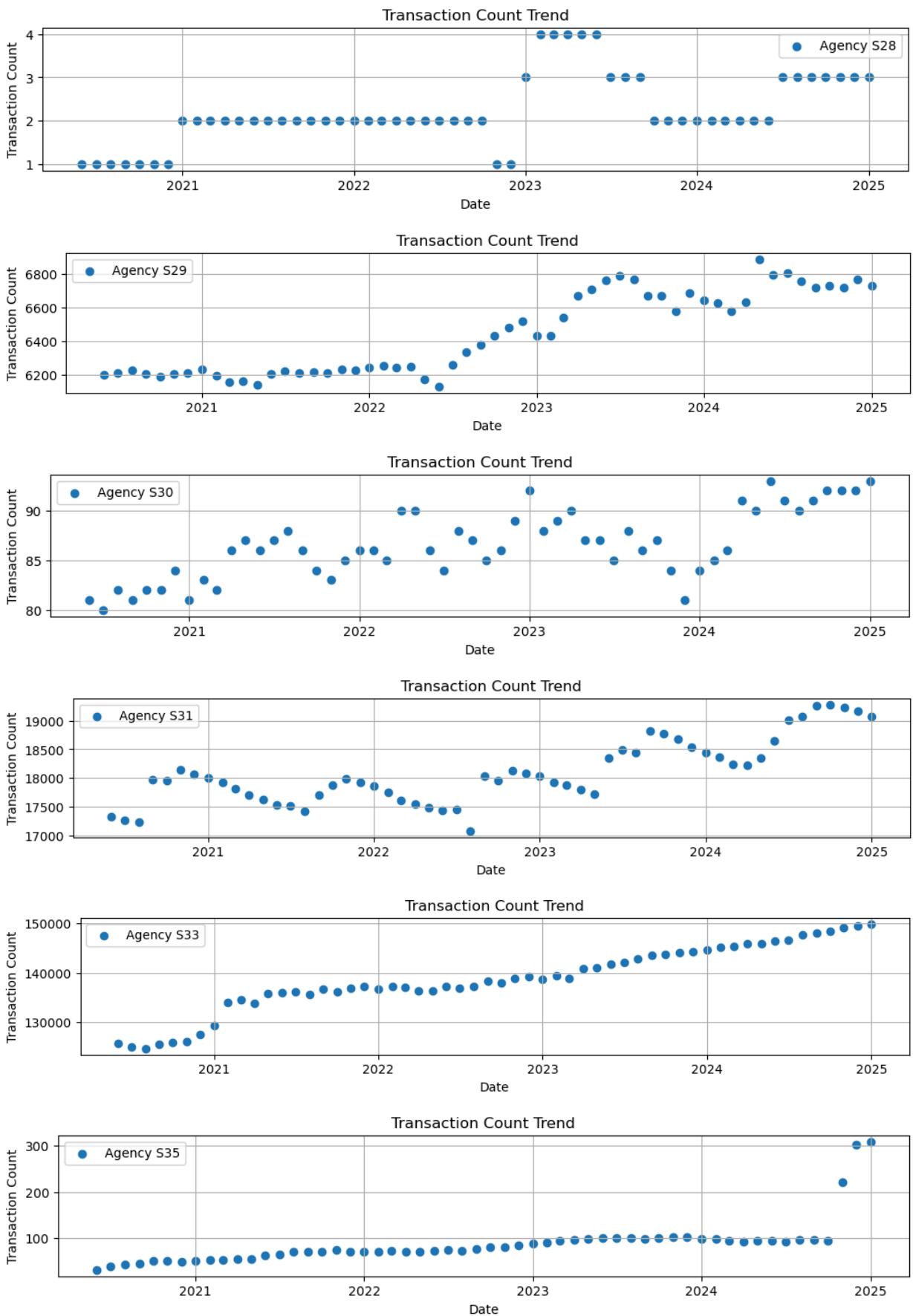


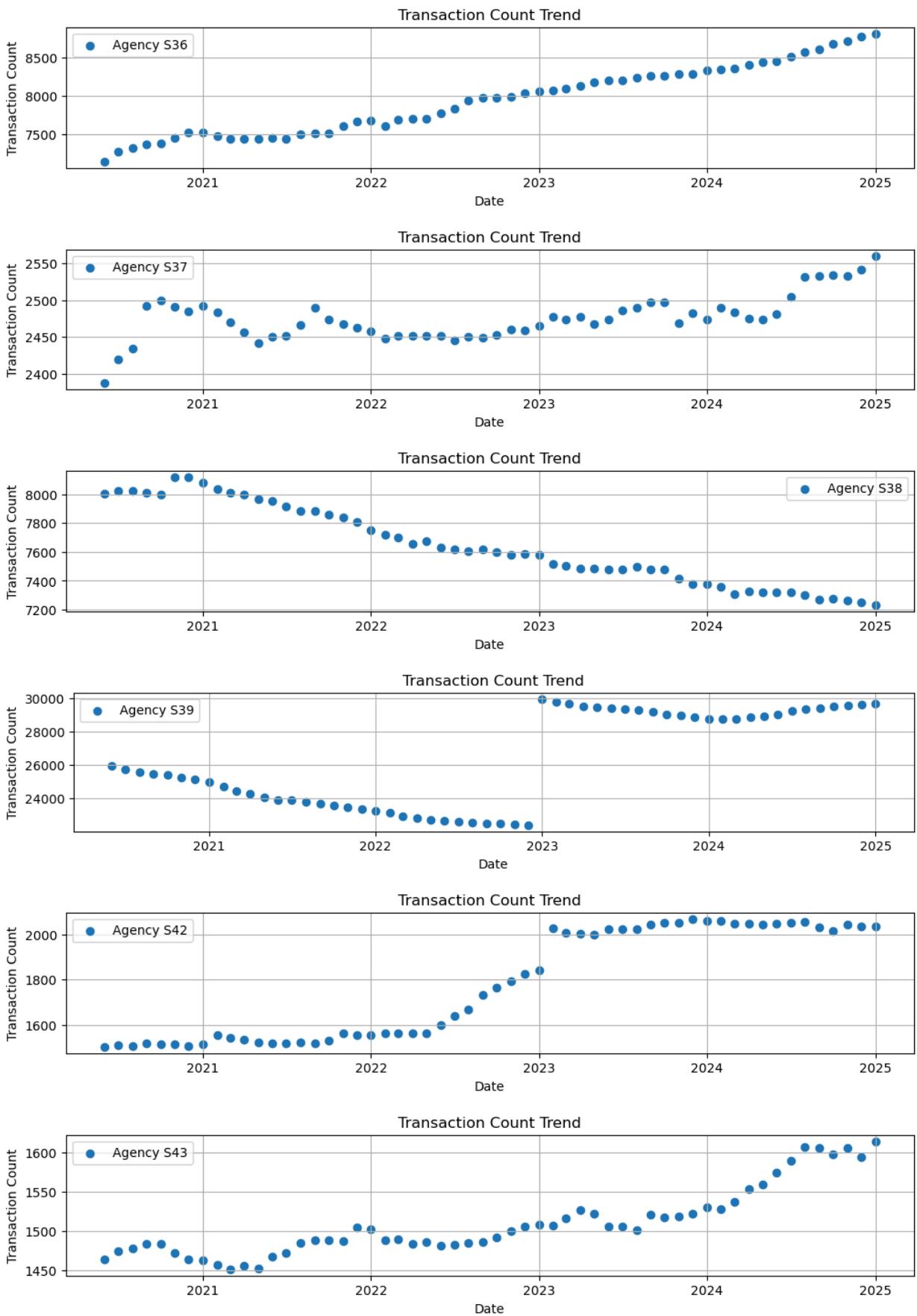


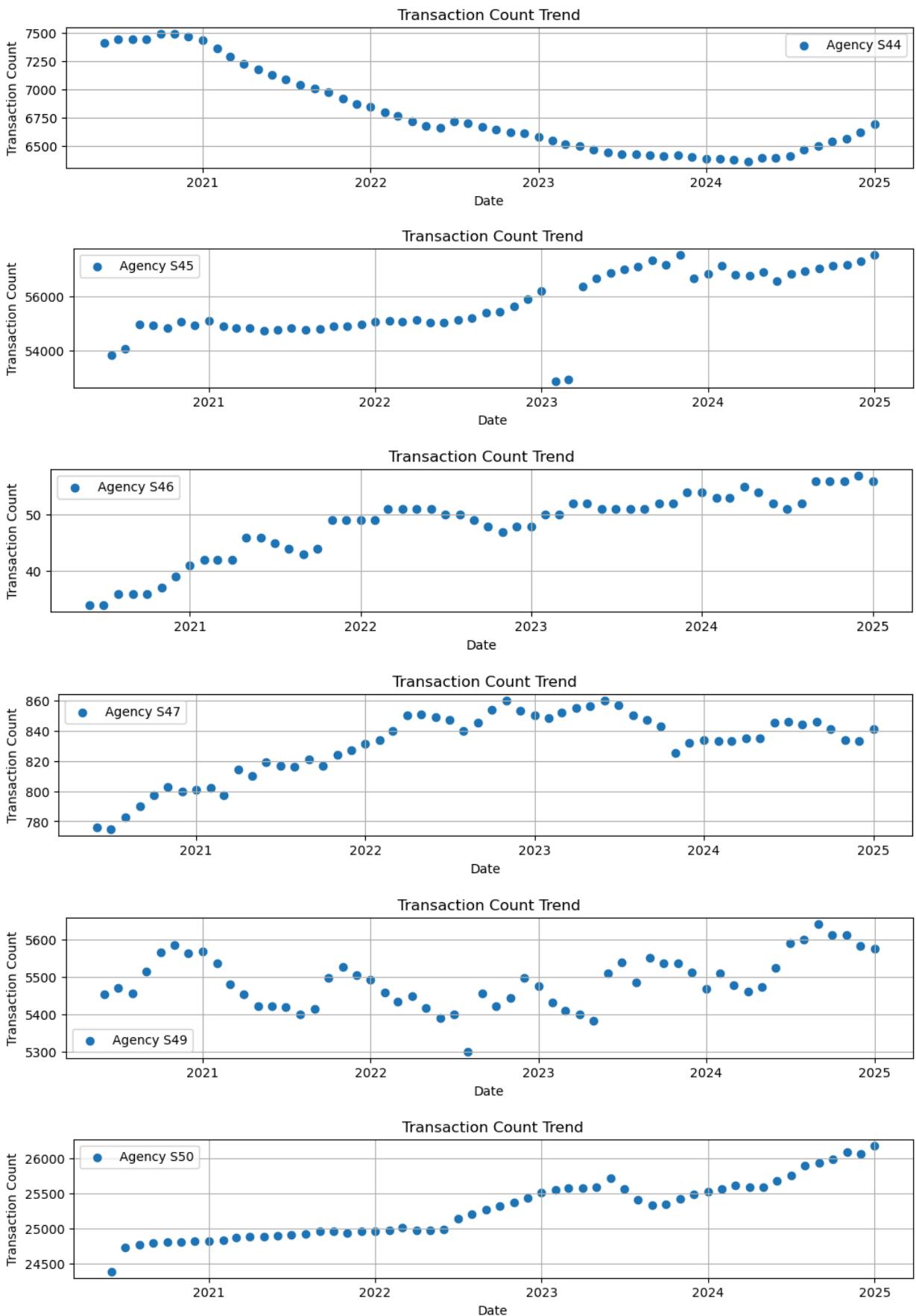


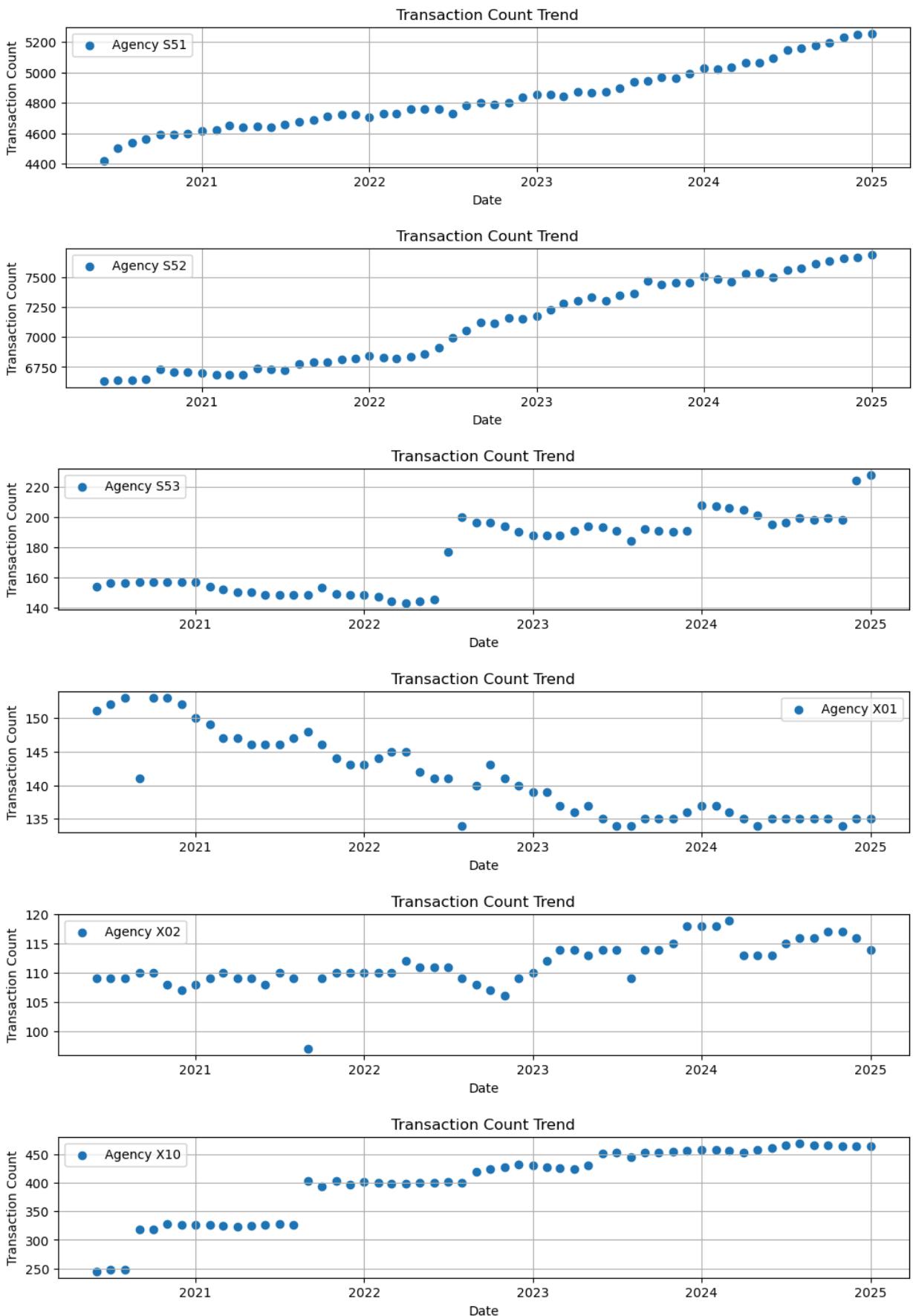


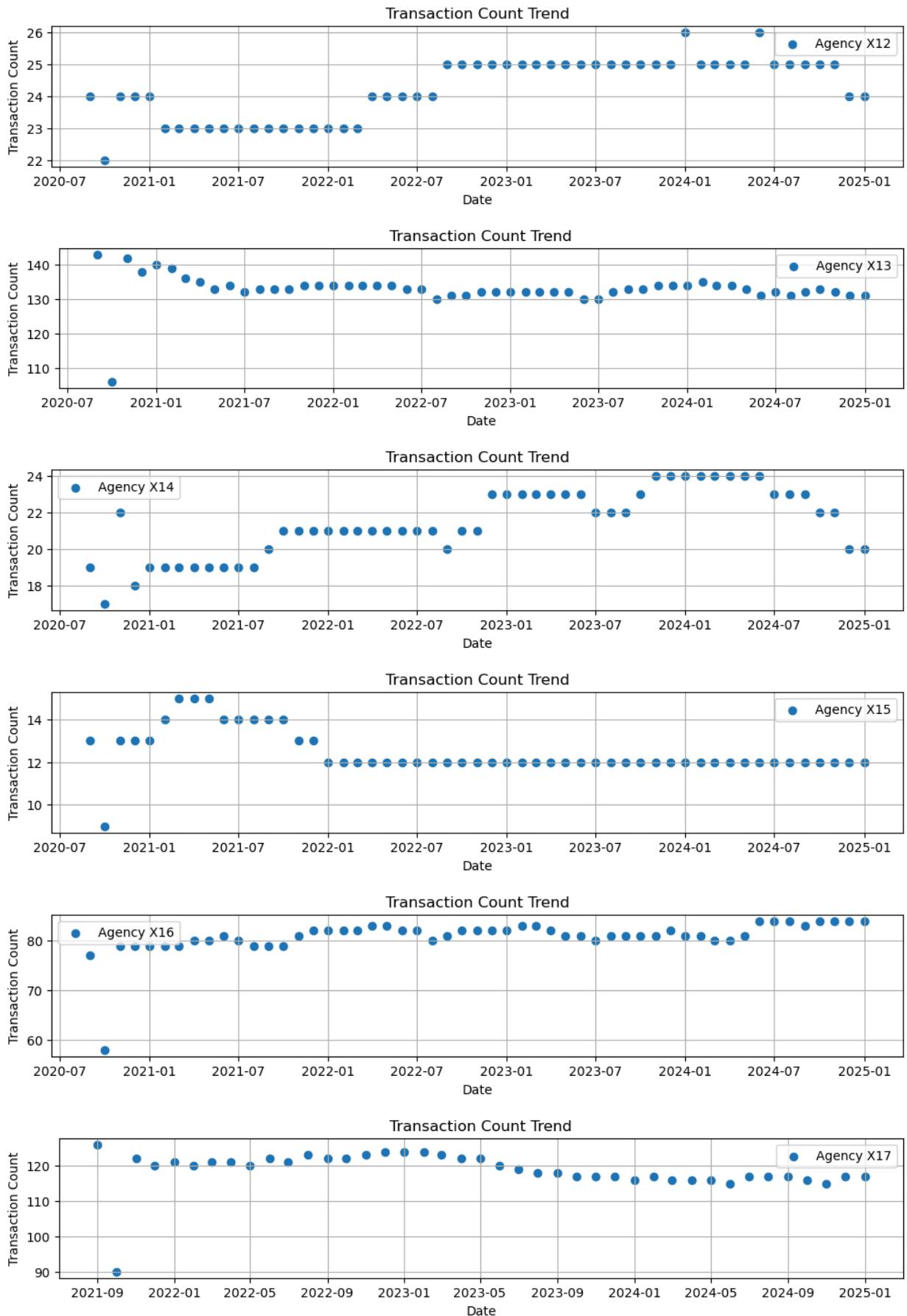


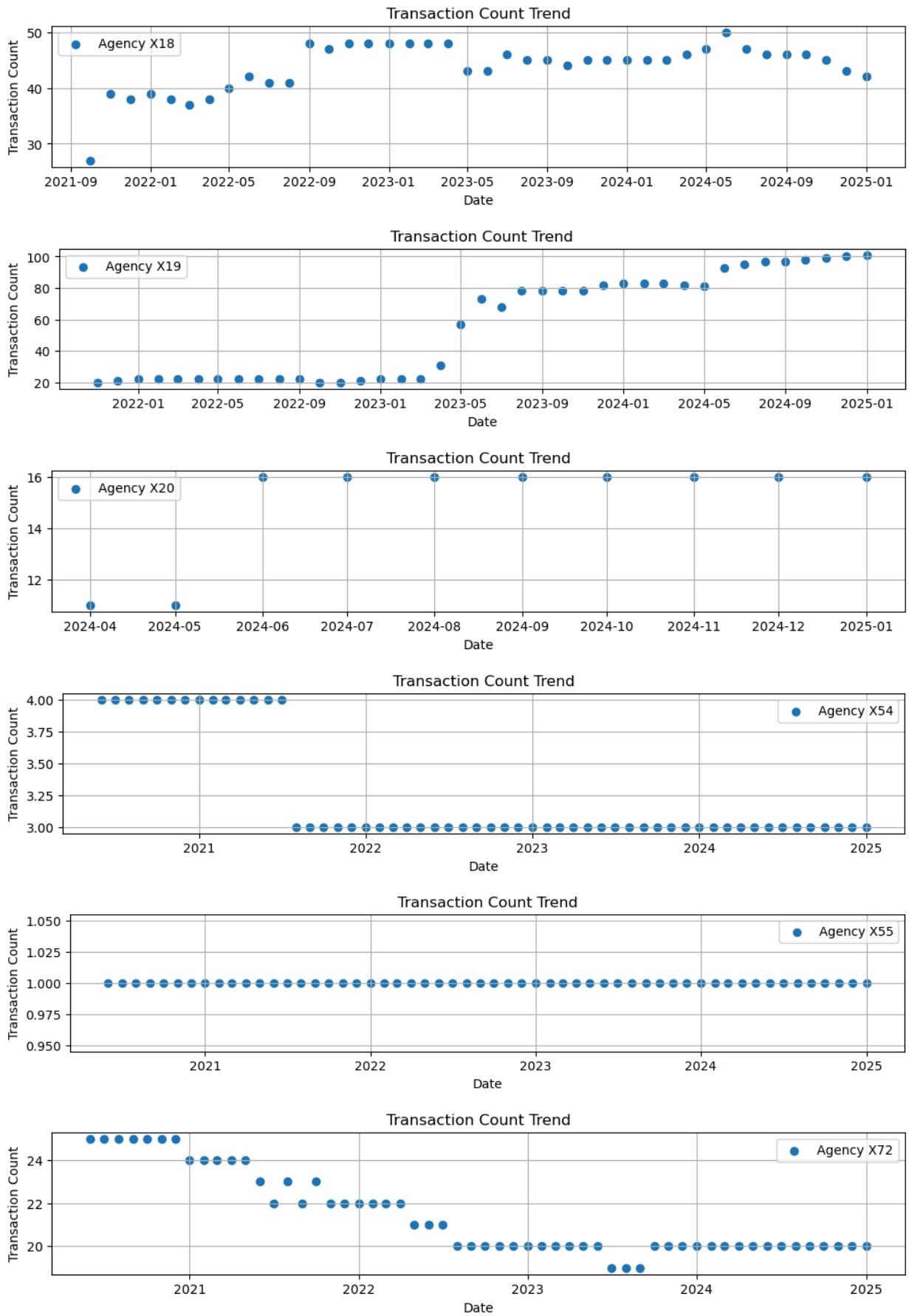


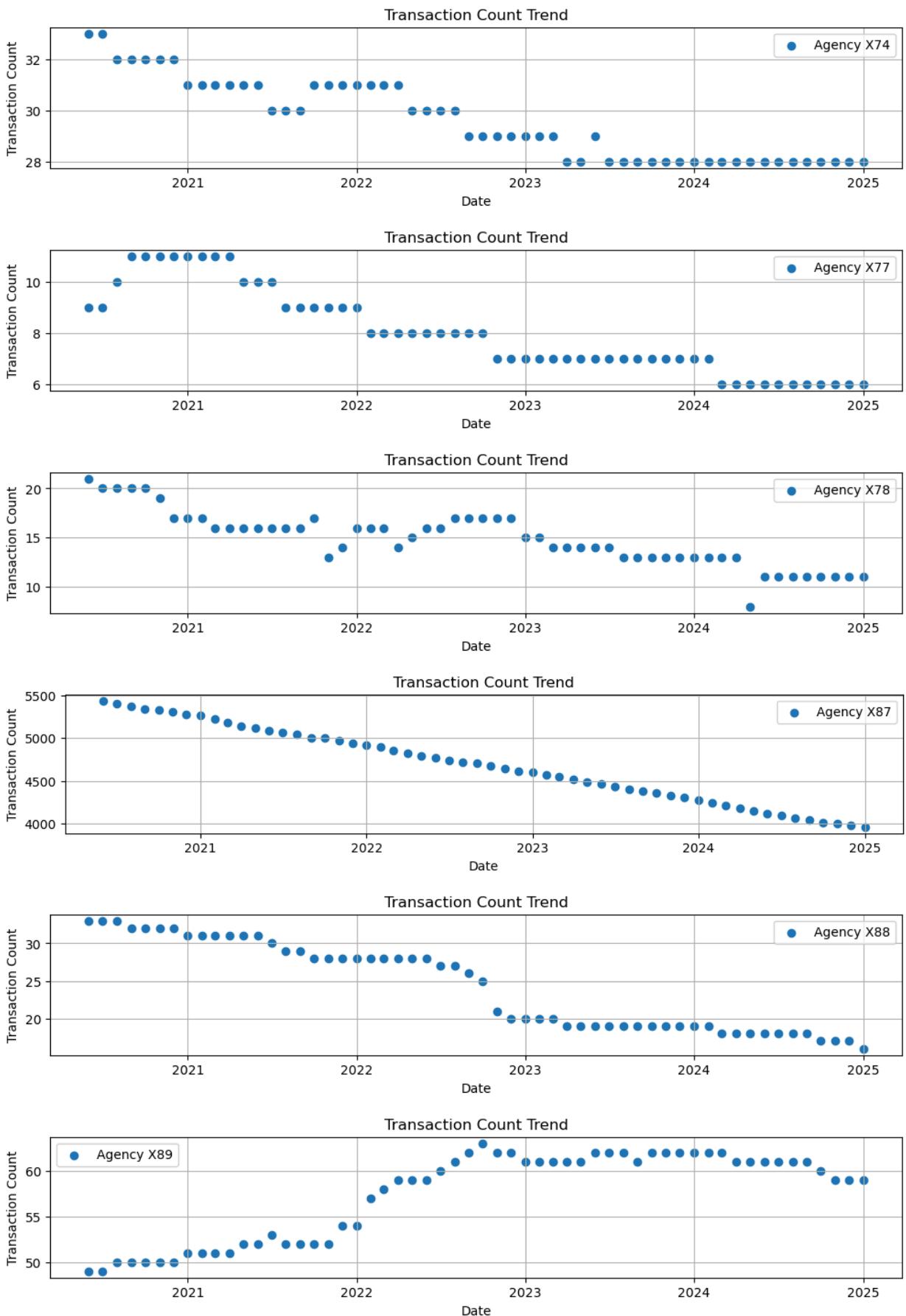


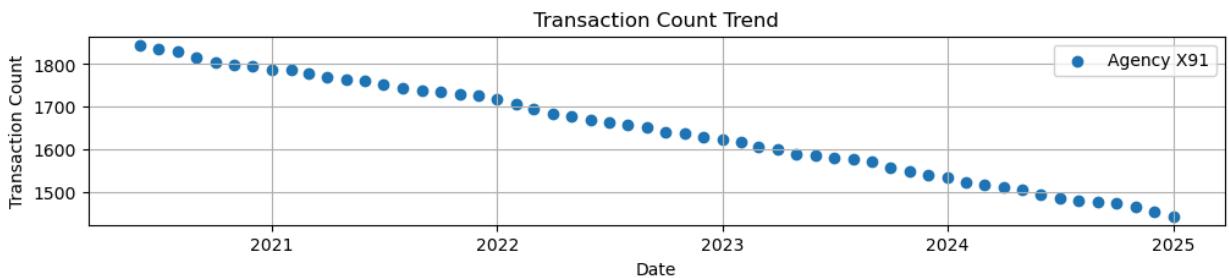












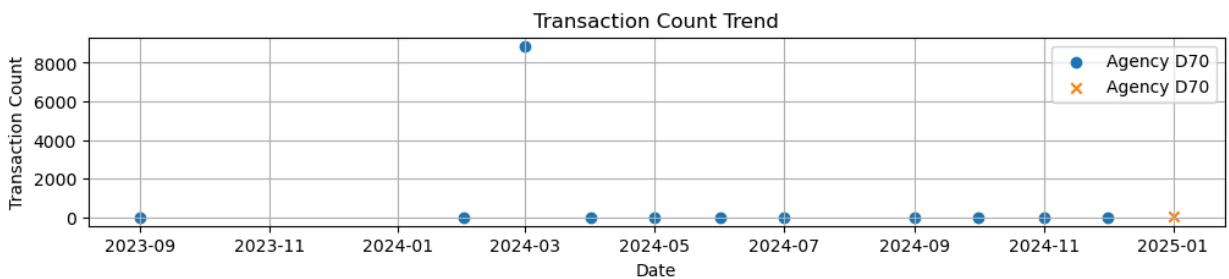
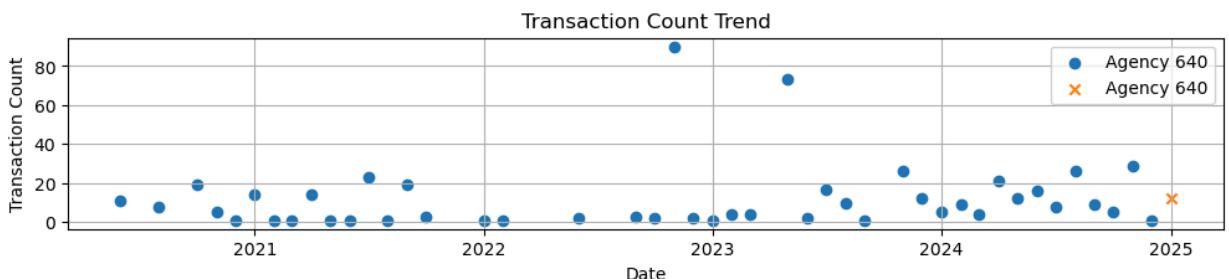
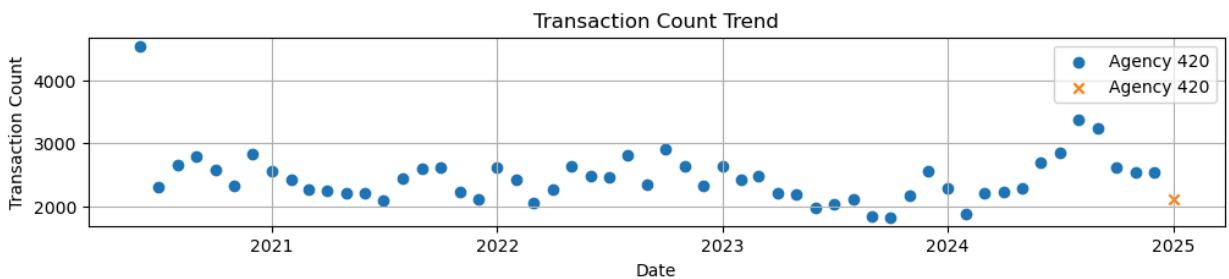
11

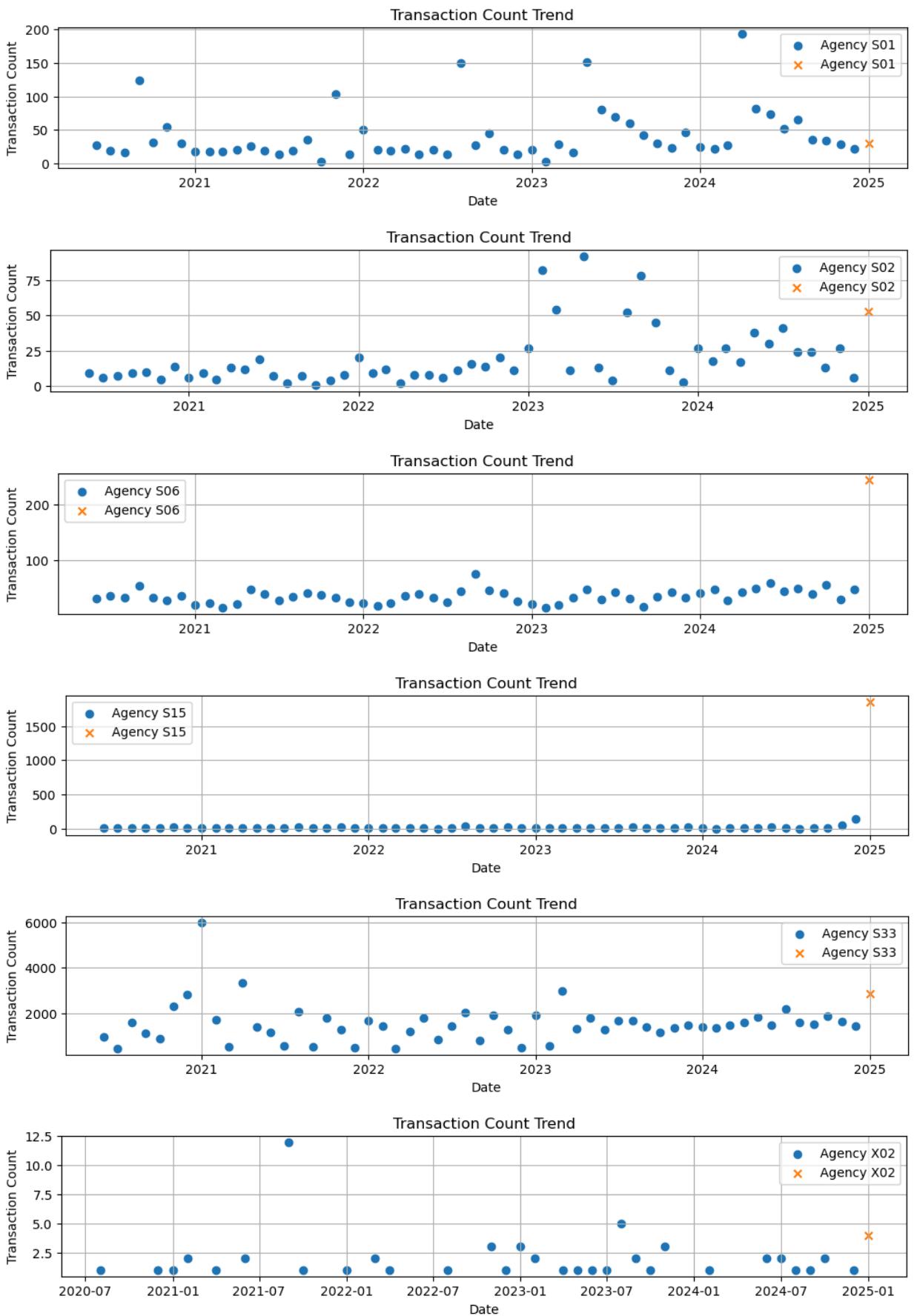
11 - Predicted as Outlier

```
In [176...]: outlier_df_11 = combined.query("status=='new' and status=='new' and tpmi_trans_cd=='11'").join(sorted(outlier_df_11.tpmi_agency_cd.unique()))
```

420 640 D70 S01 S02 S06 S15 S33 X02

```
In [177...]: for agency in sorted(outlier_df_11.tpmi_agency_cd.unique()): show_outliers(combined, '11', [agency])
```

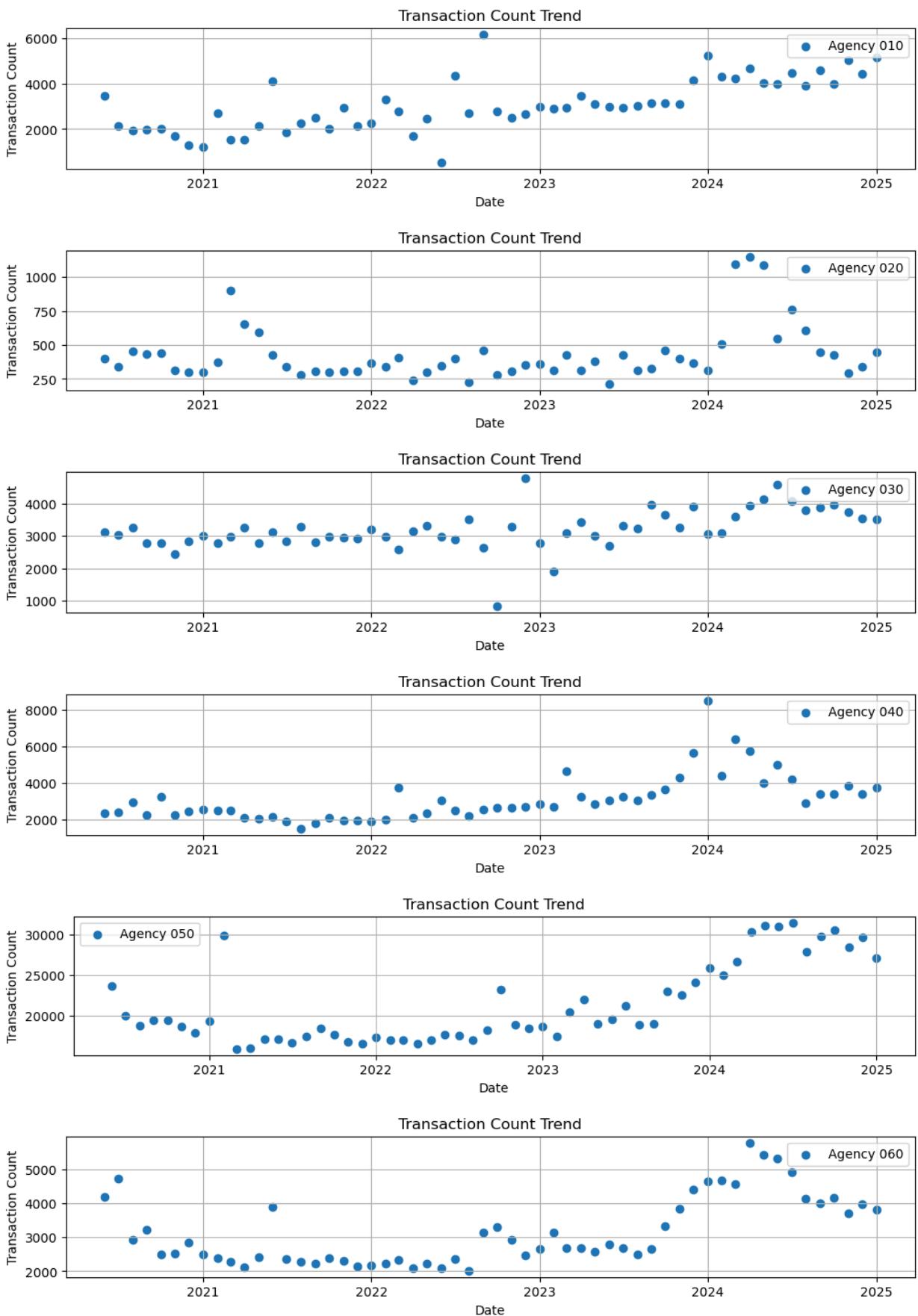


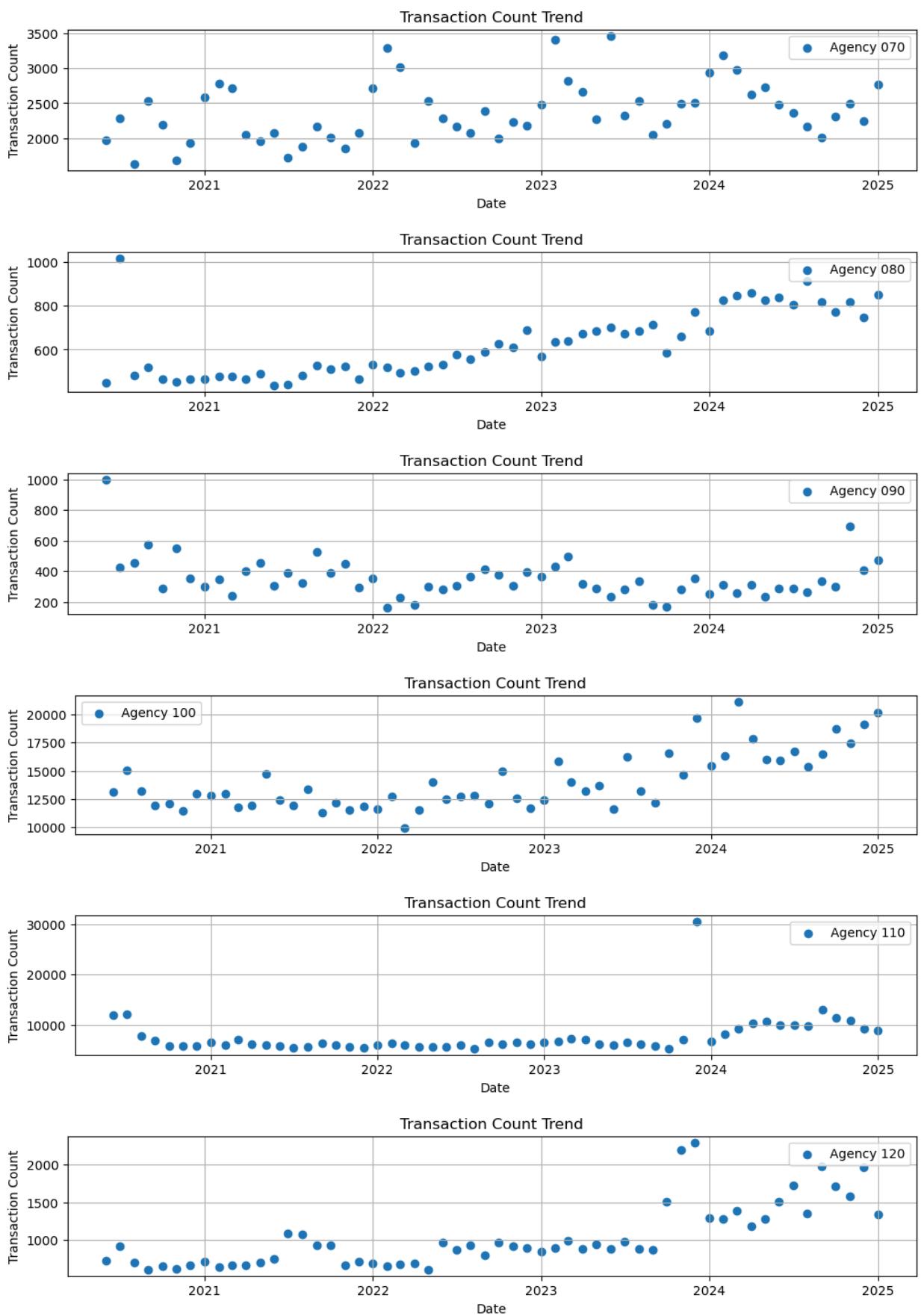


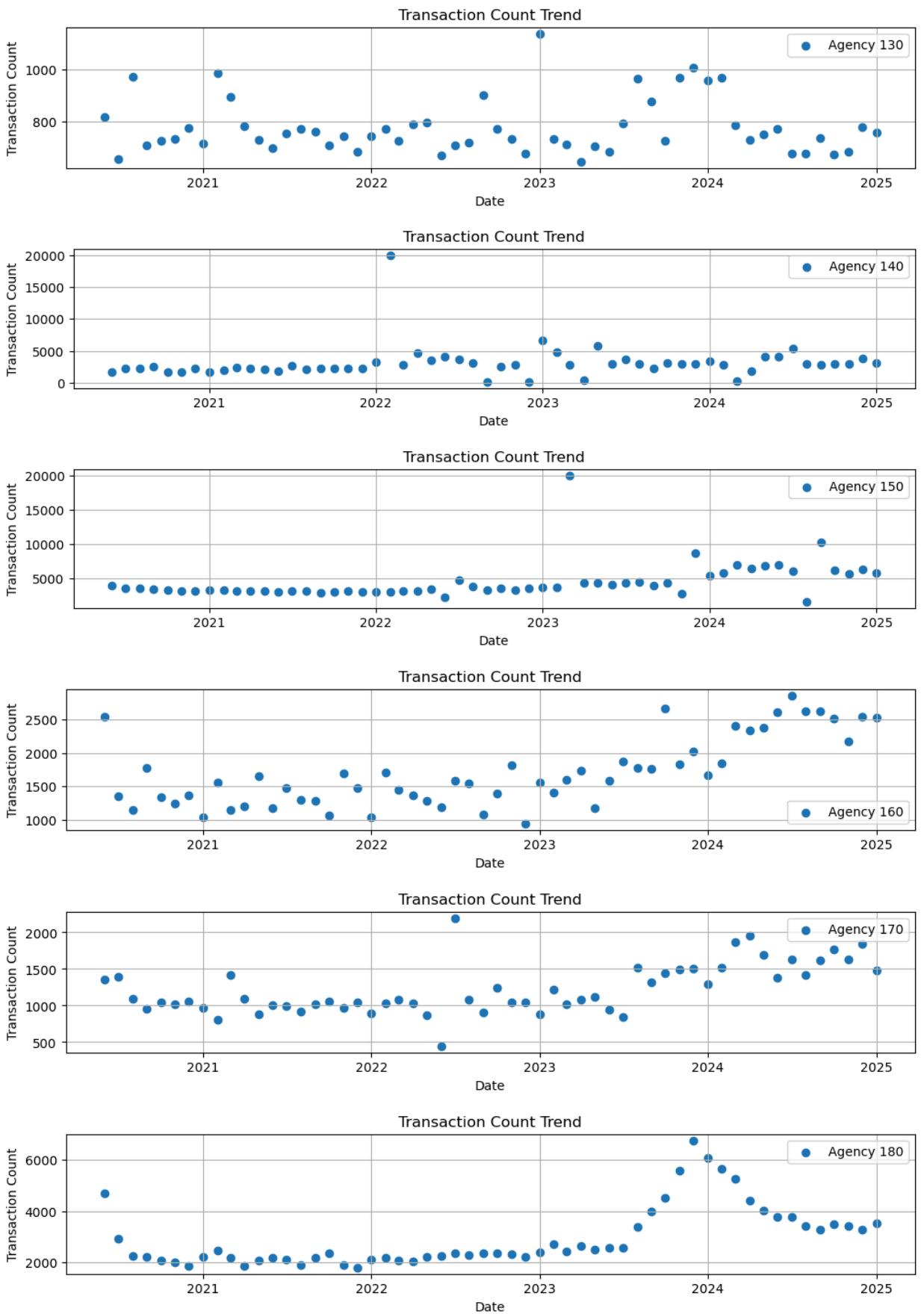
11 - Predicted as Normal

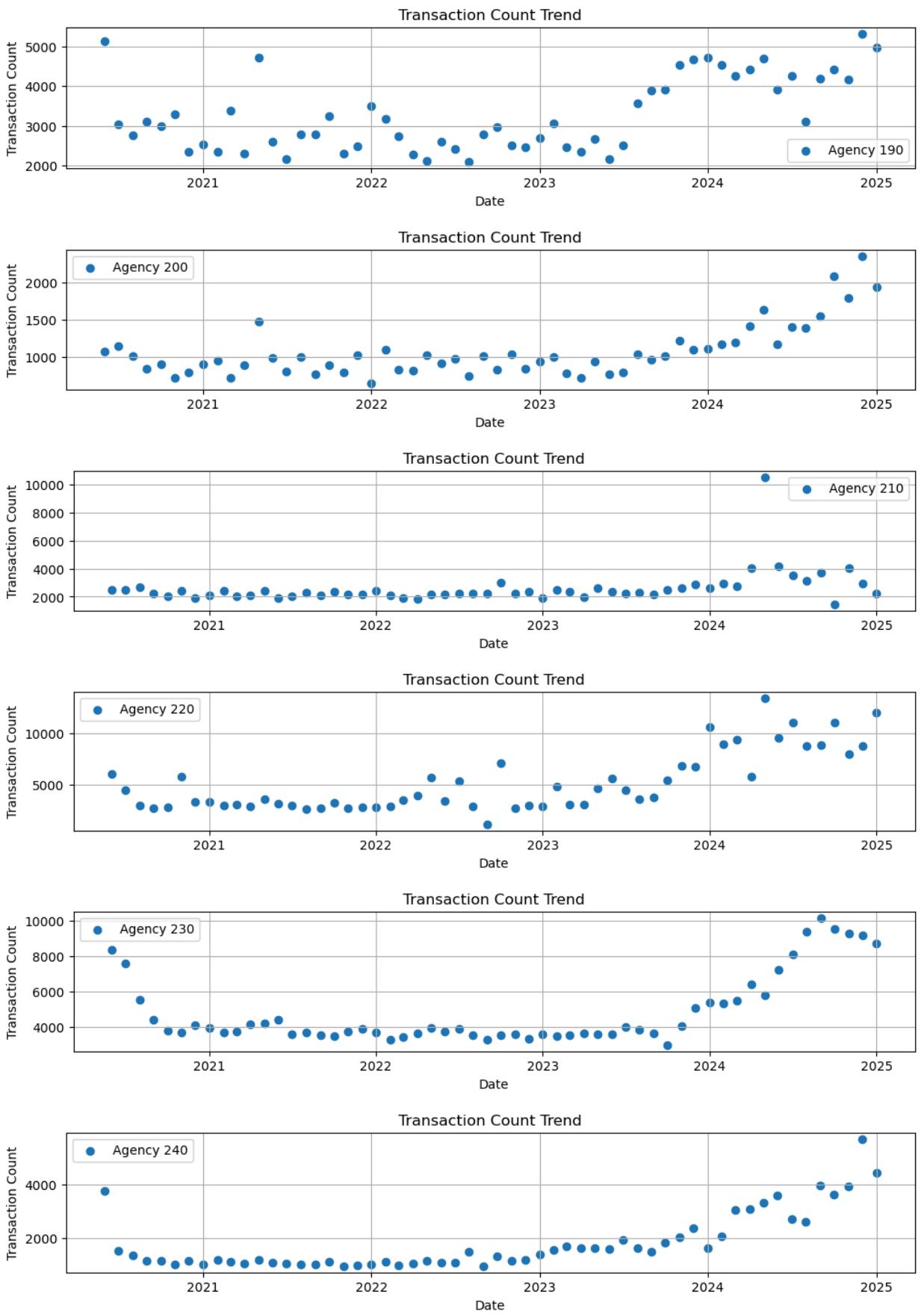
In [178]:

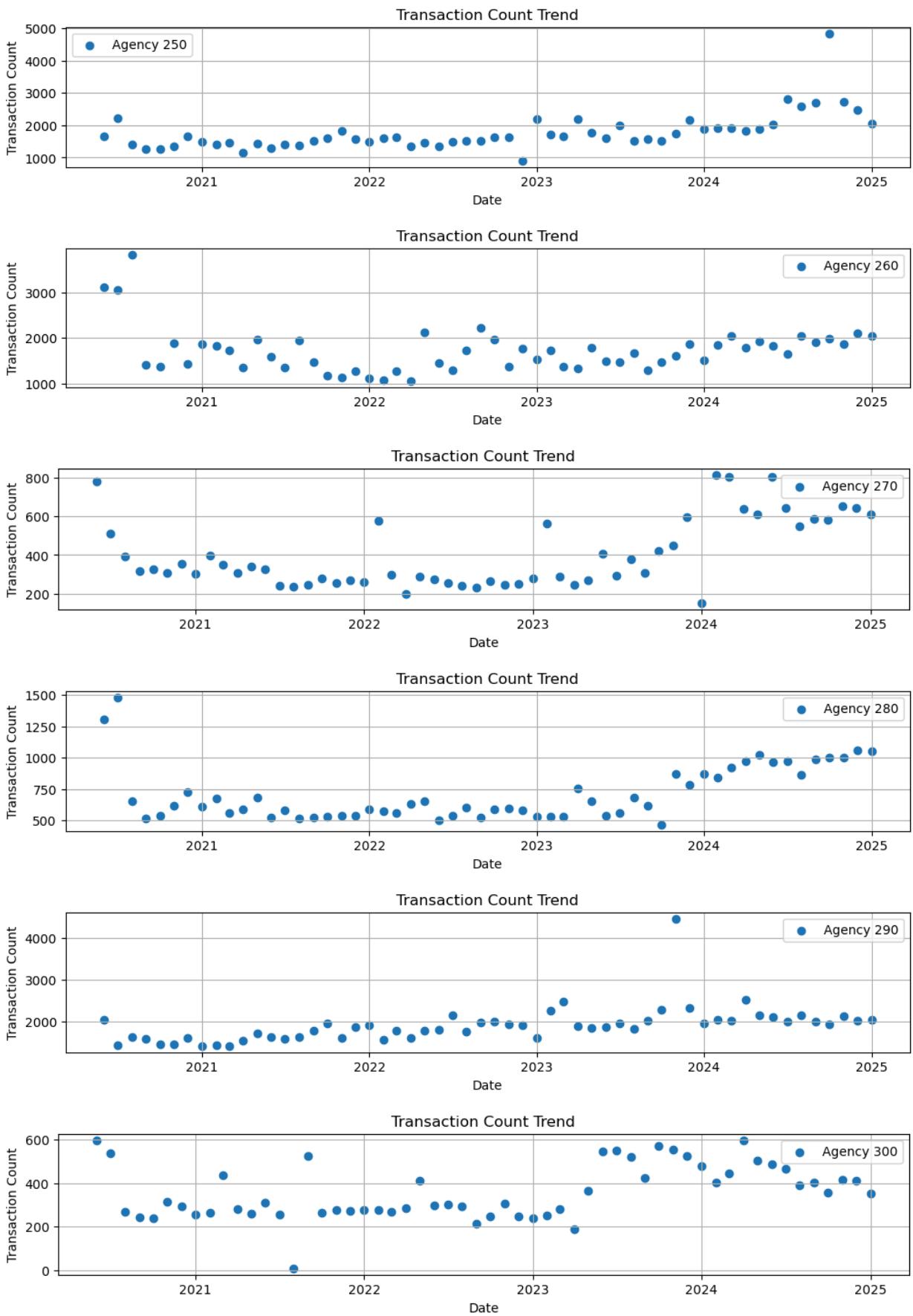
```
for agency in sorted(combined.query("status=='new' and tpmi_trans_cd=='11' and outlier  
show_outliers(combined, '11', [agency])
```

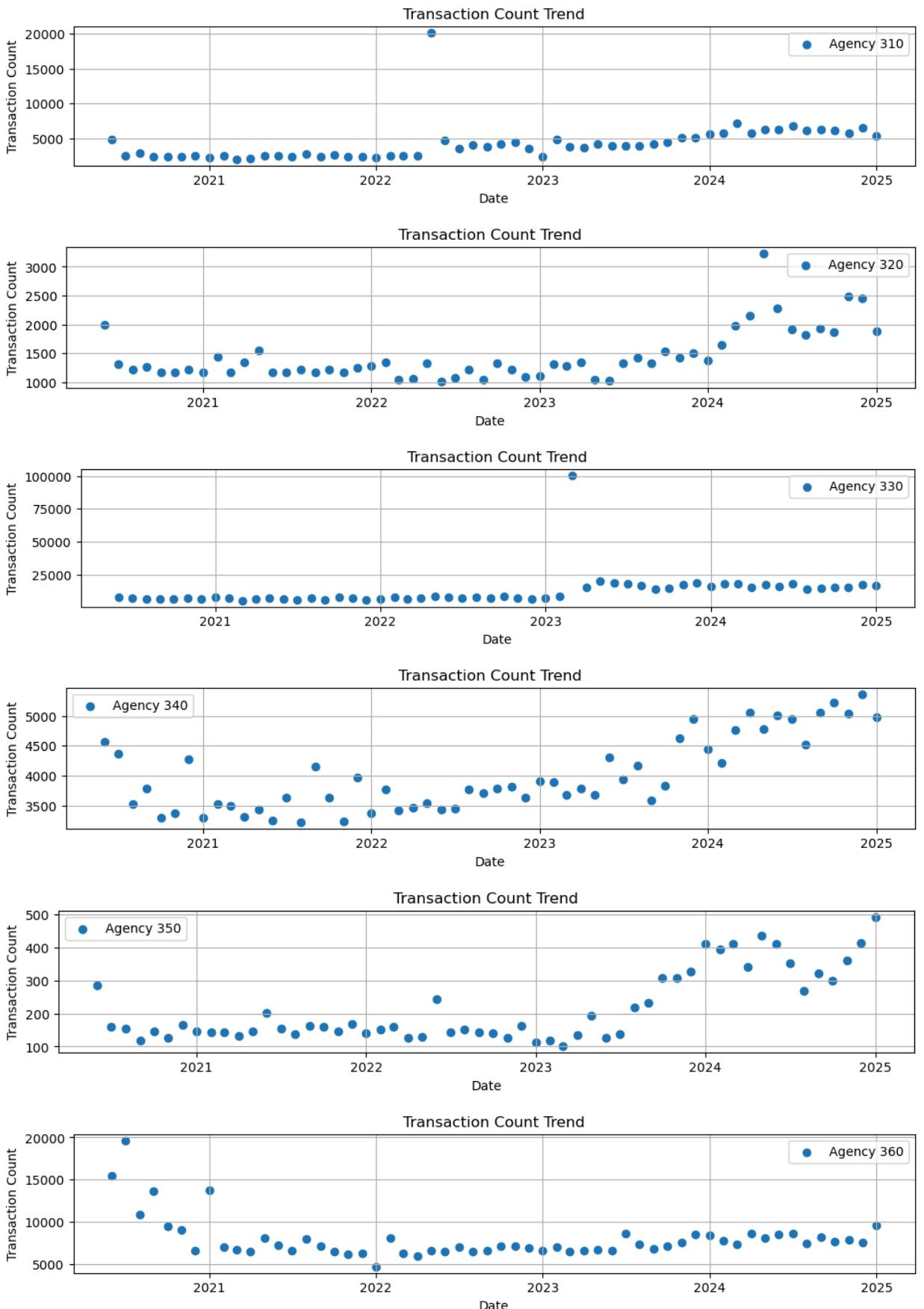


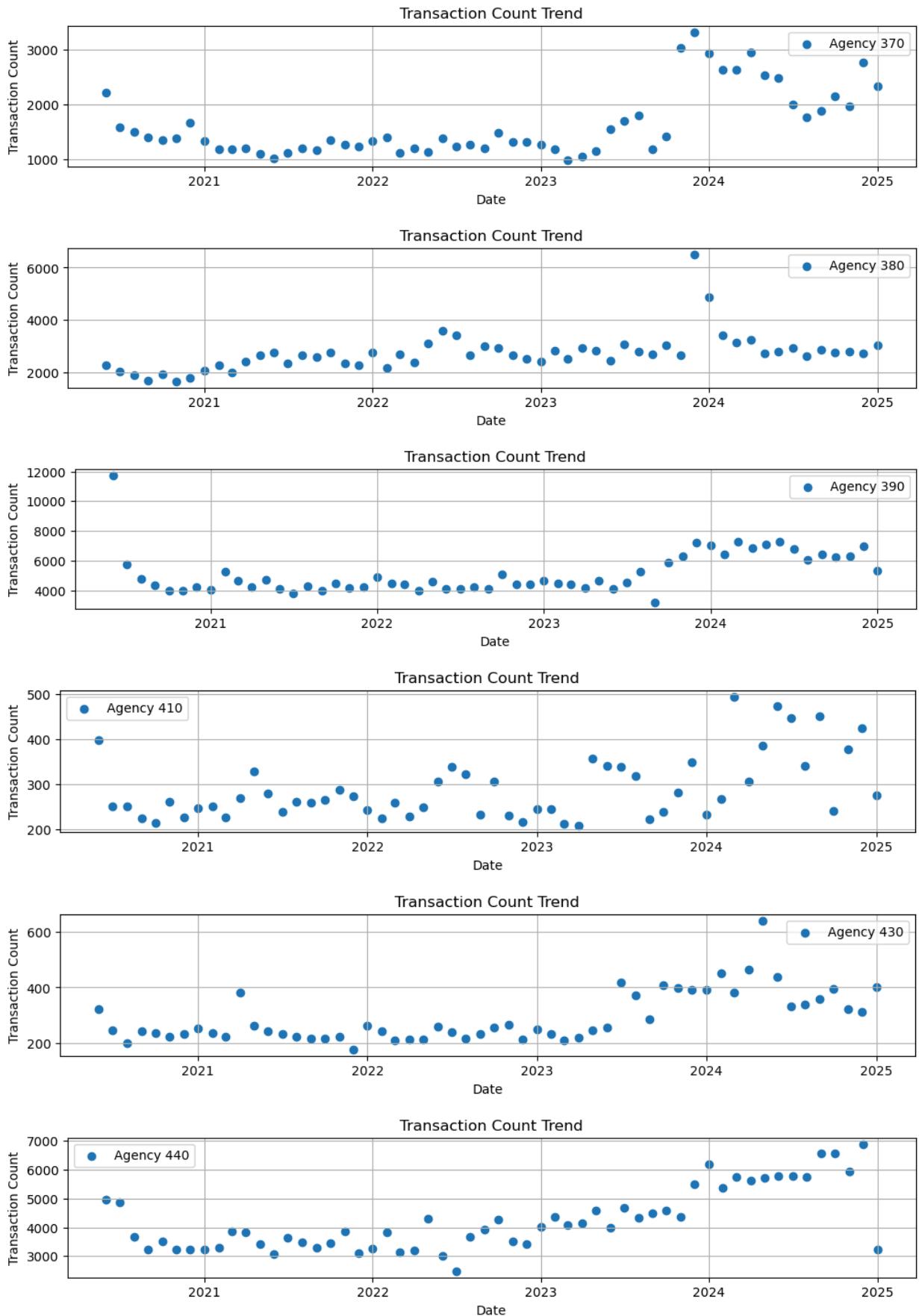


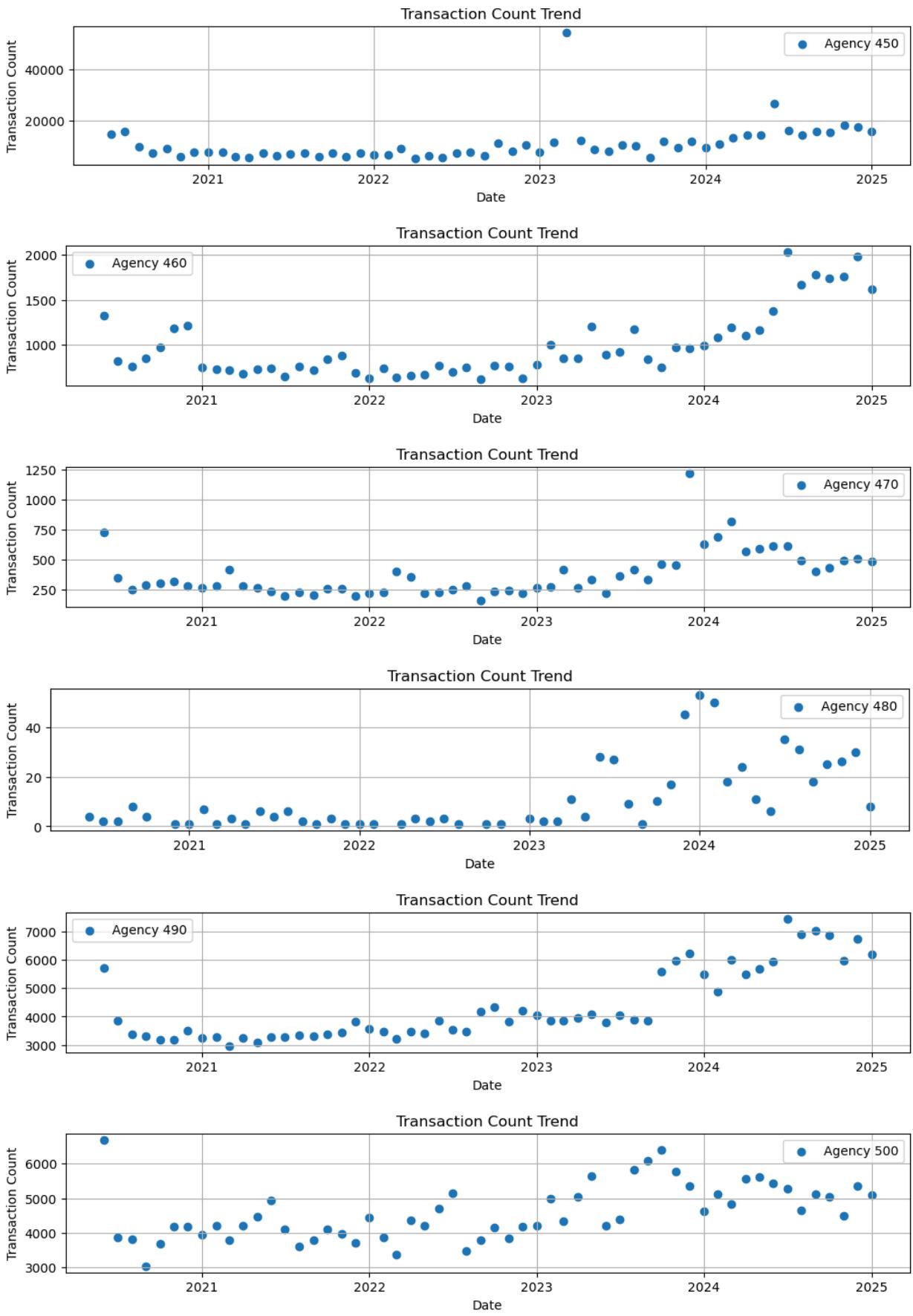


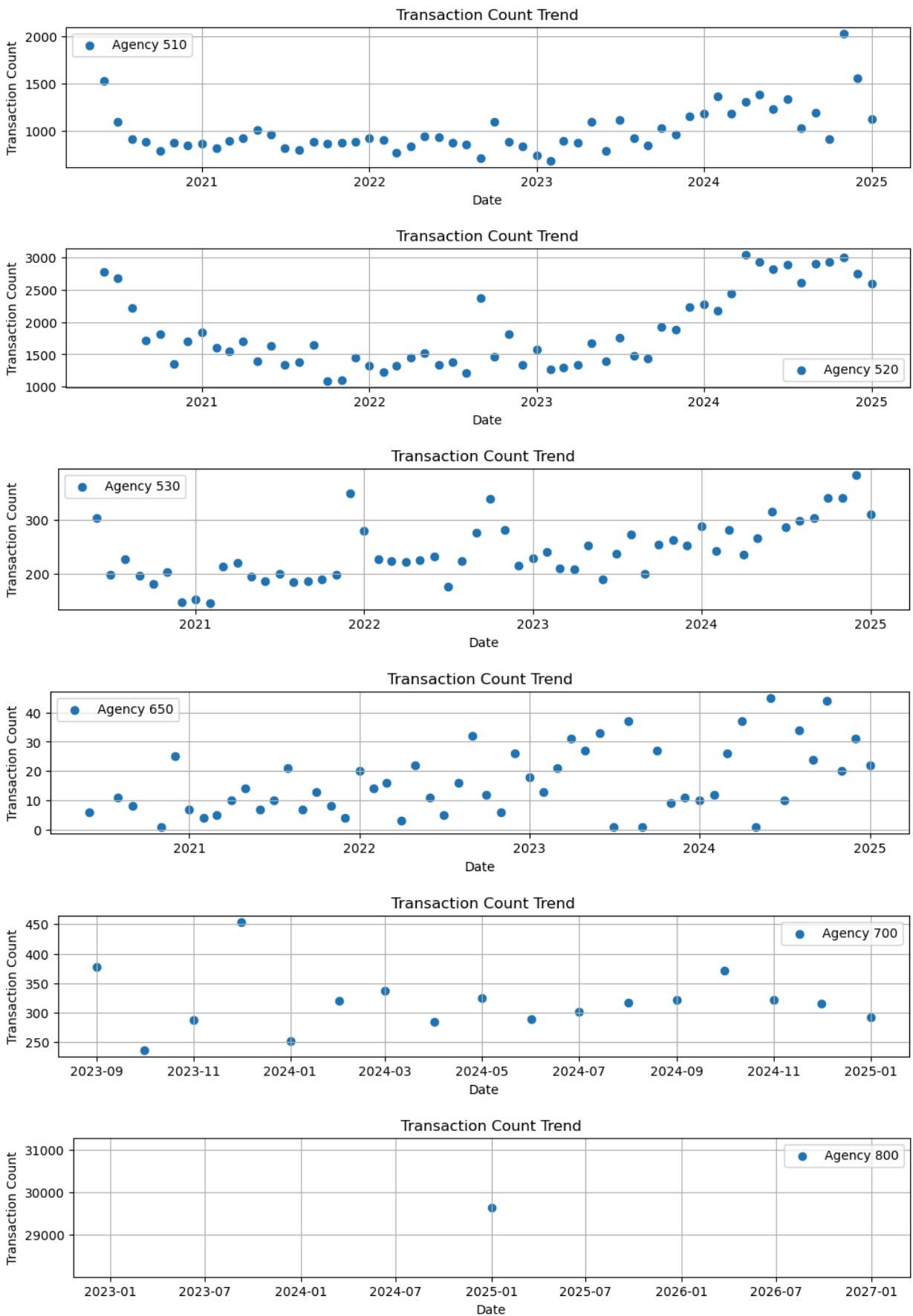


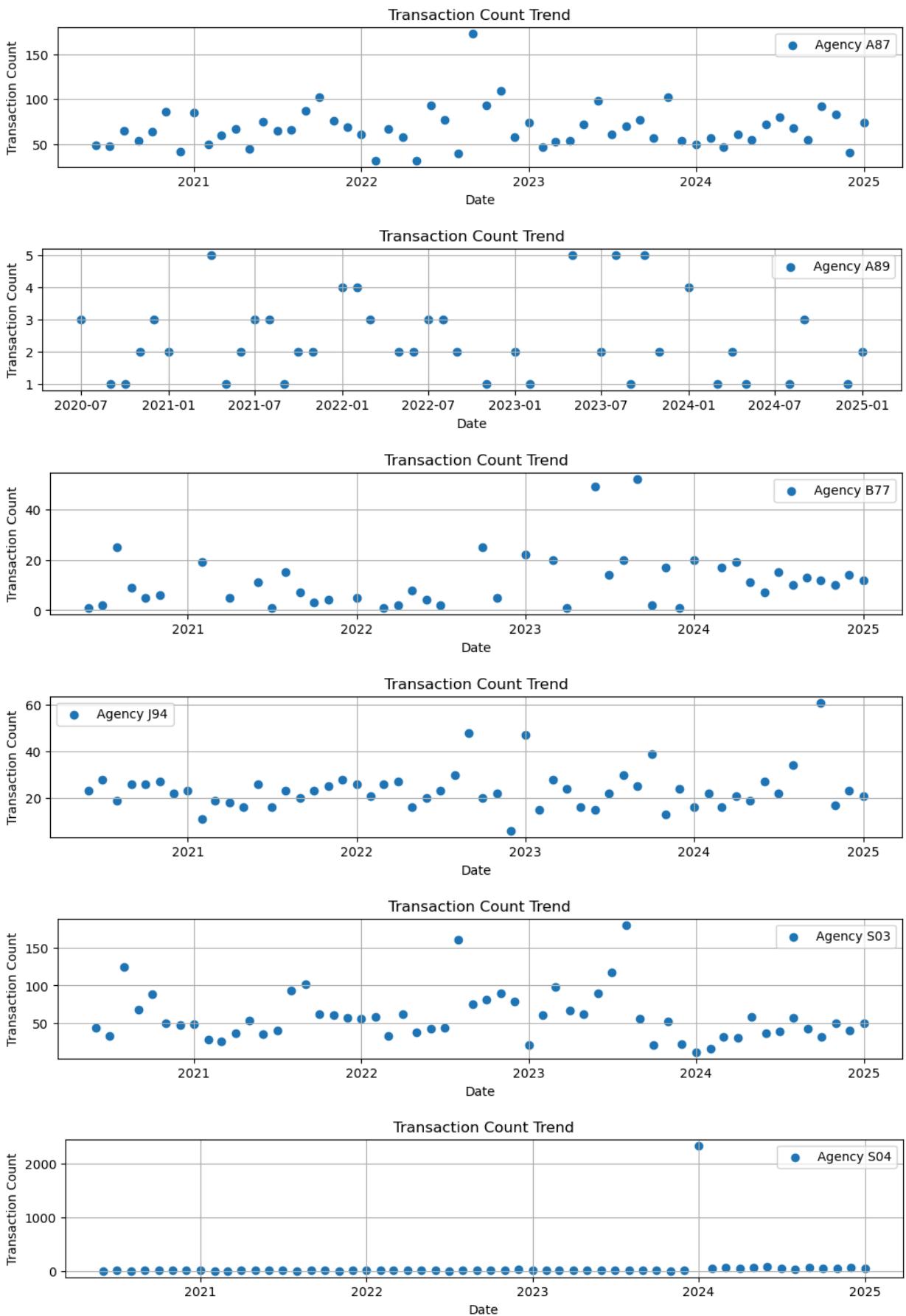


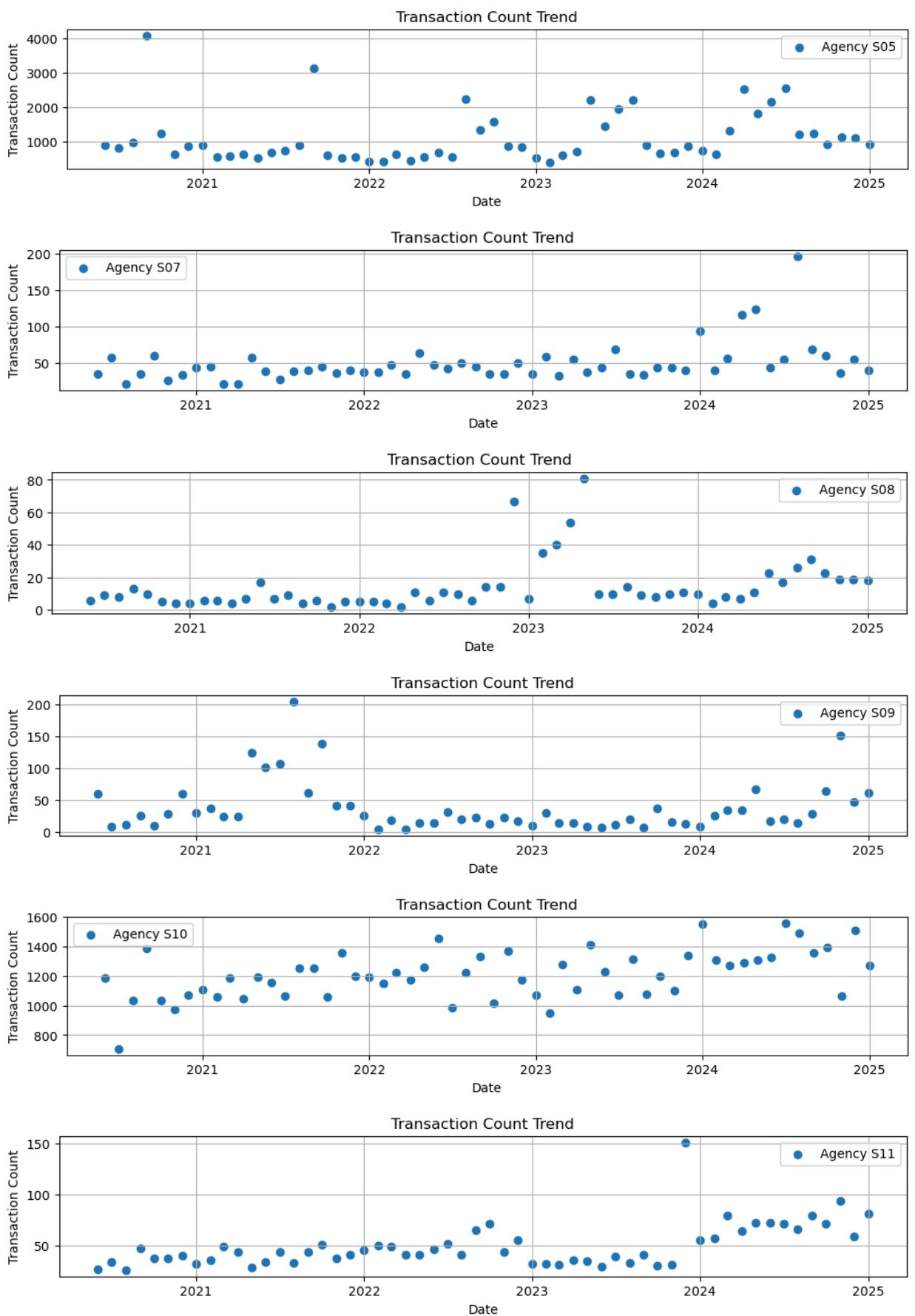


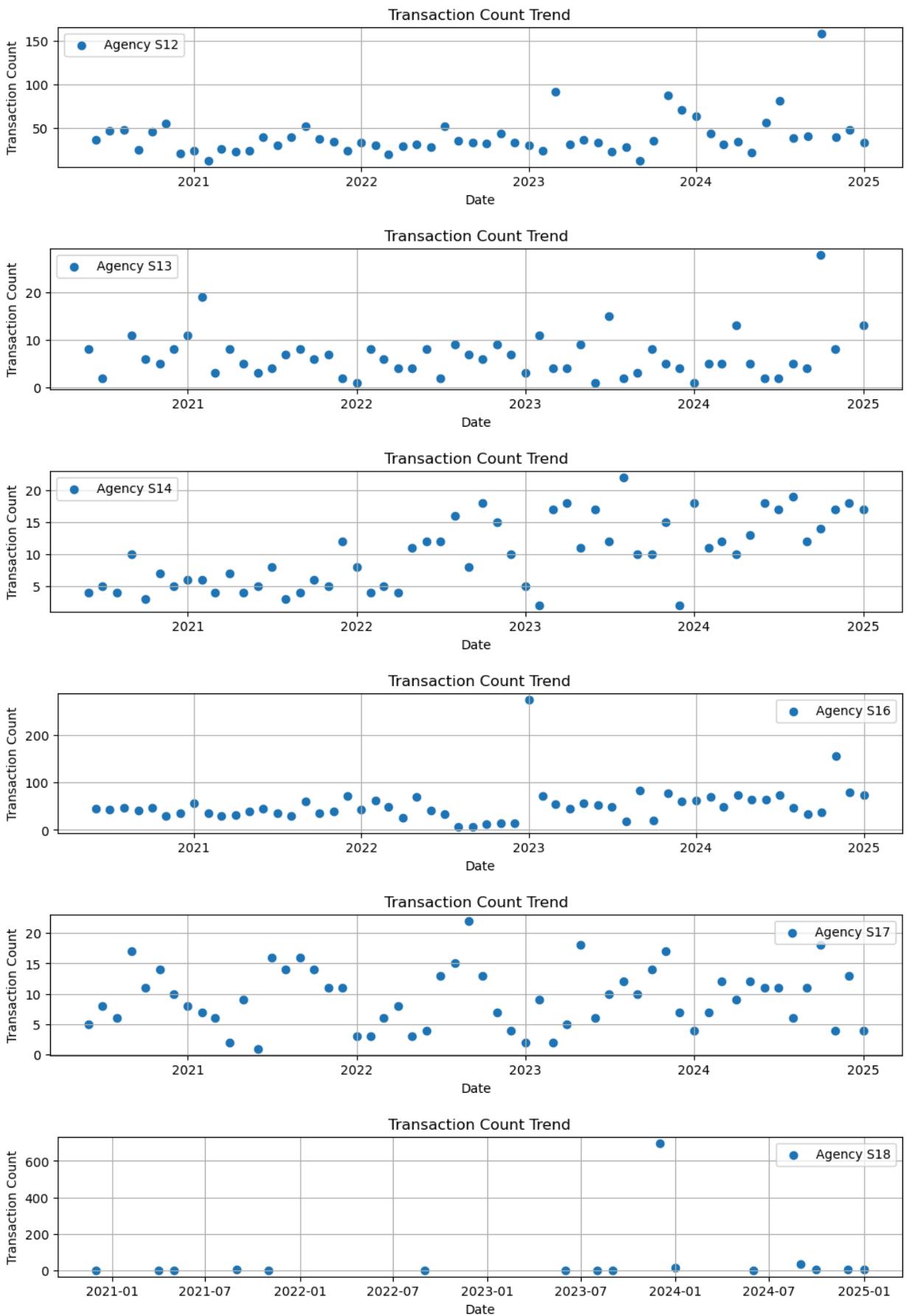


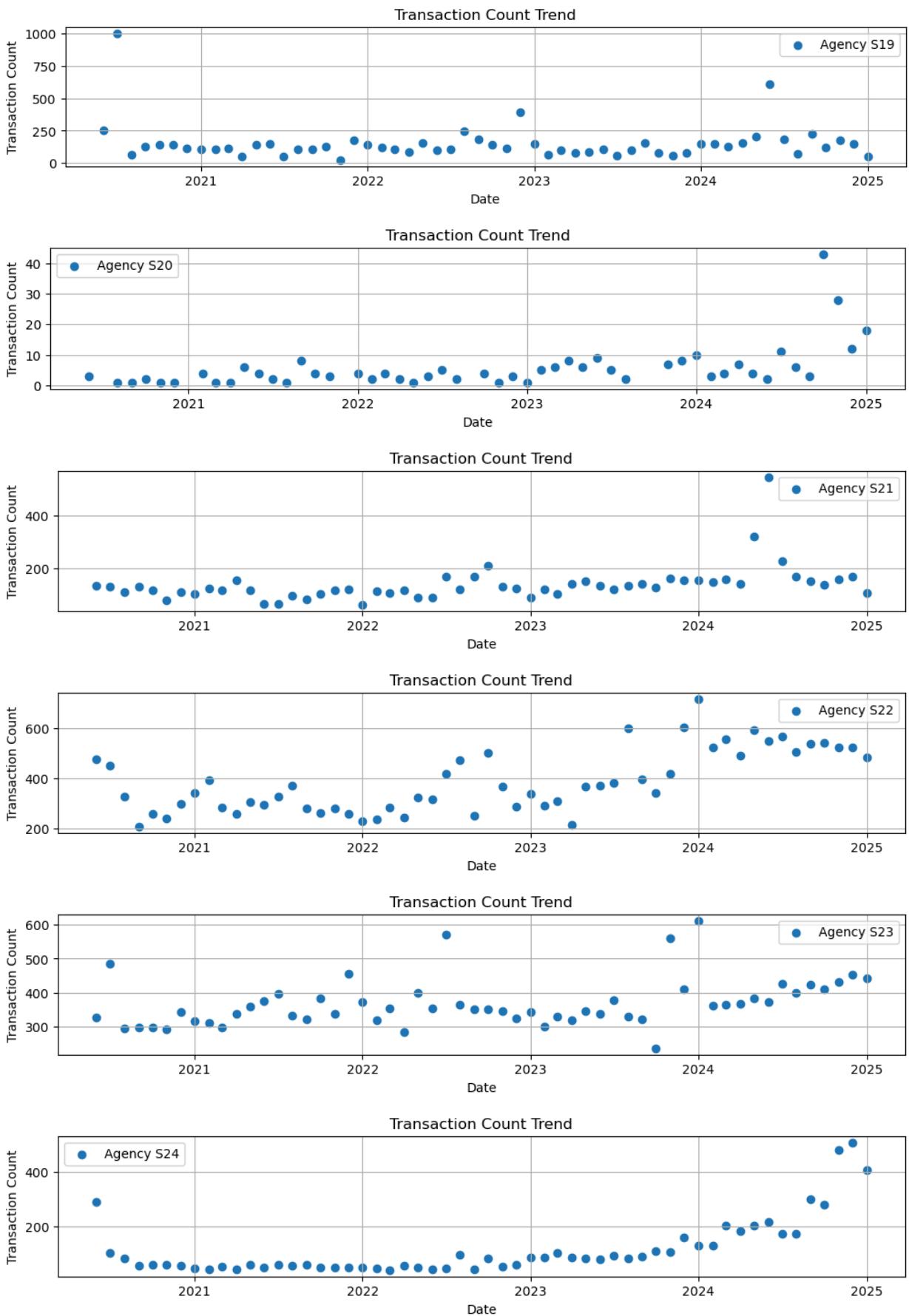


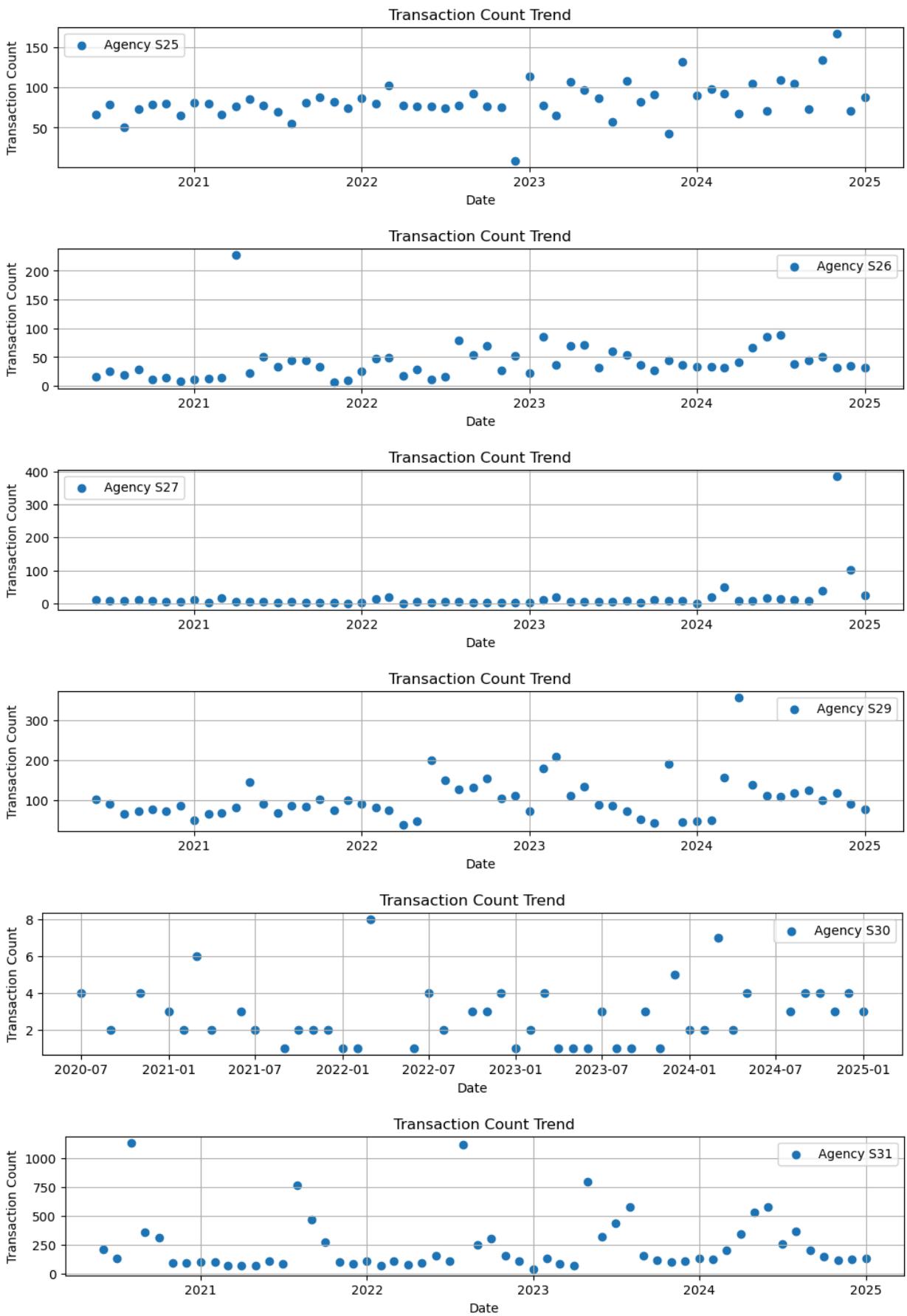


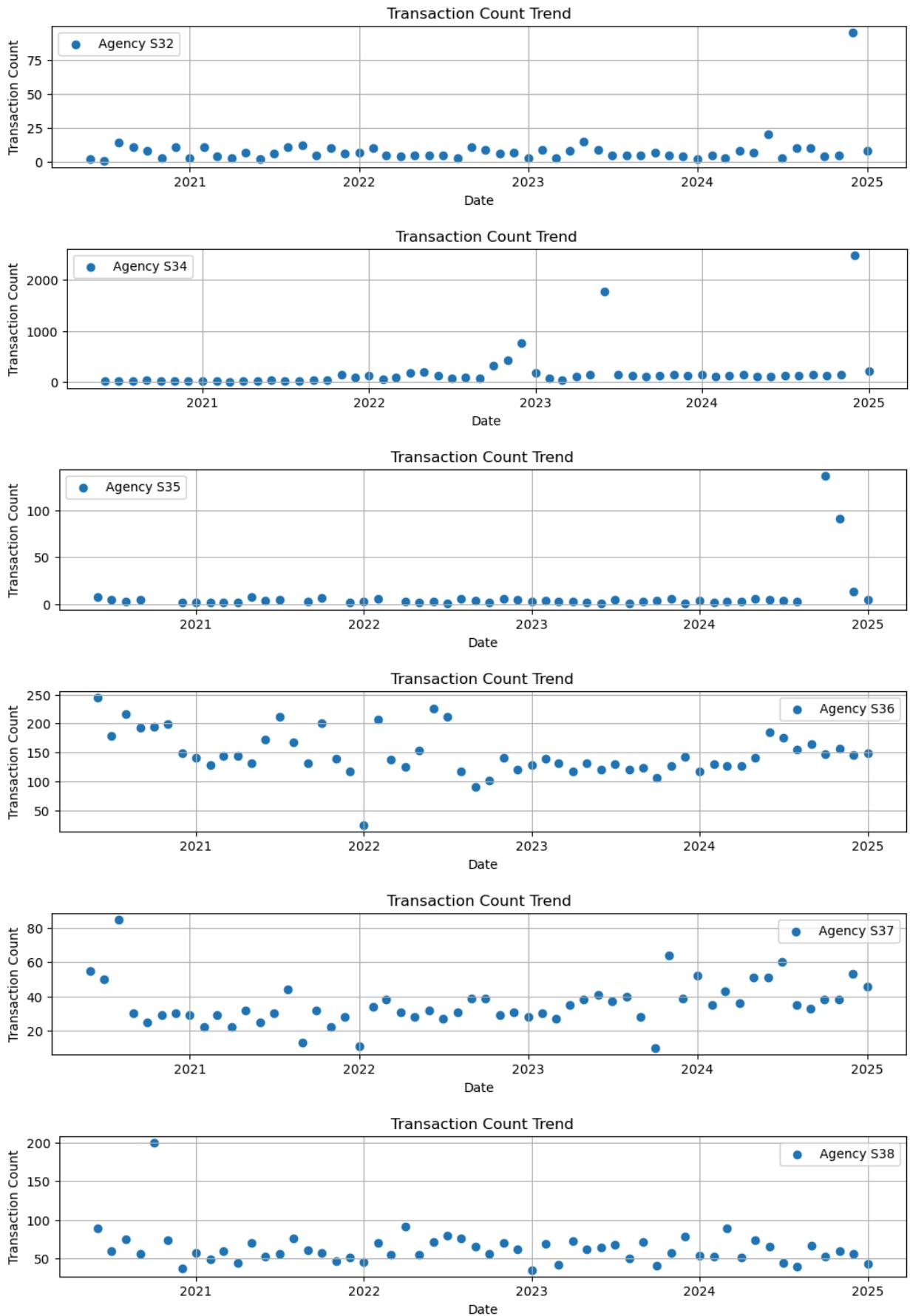


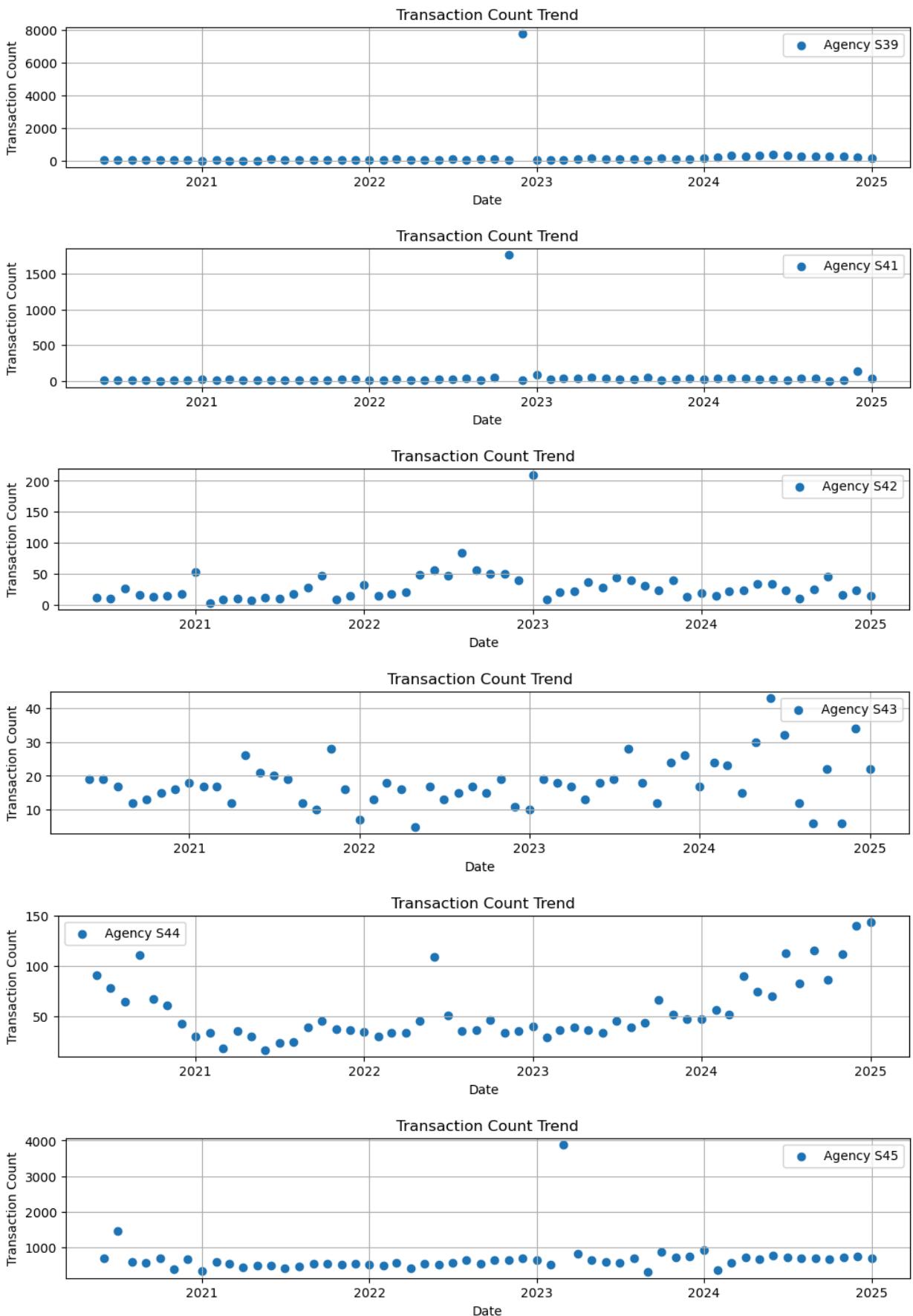


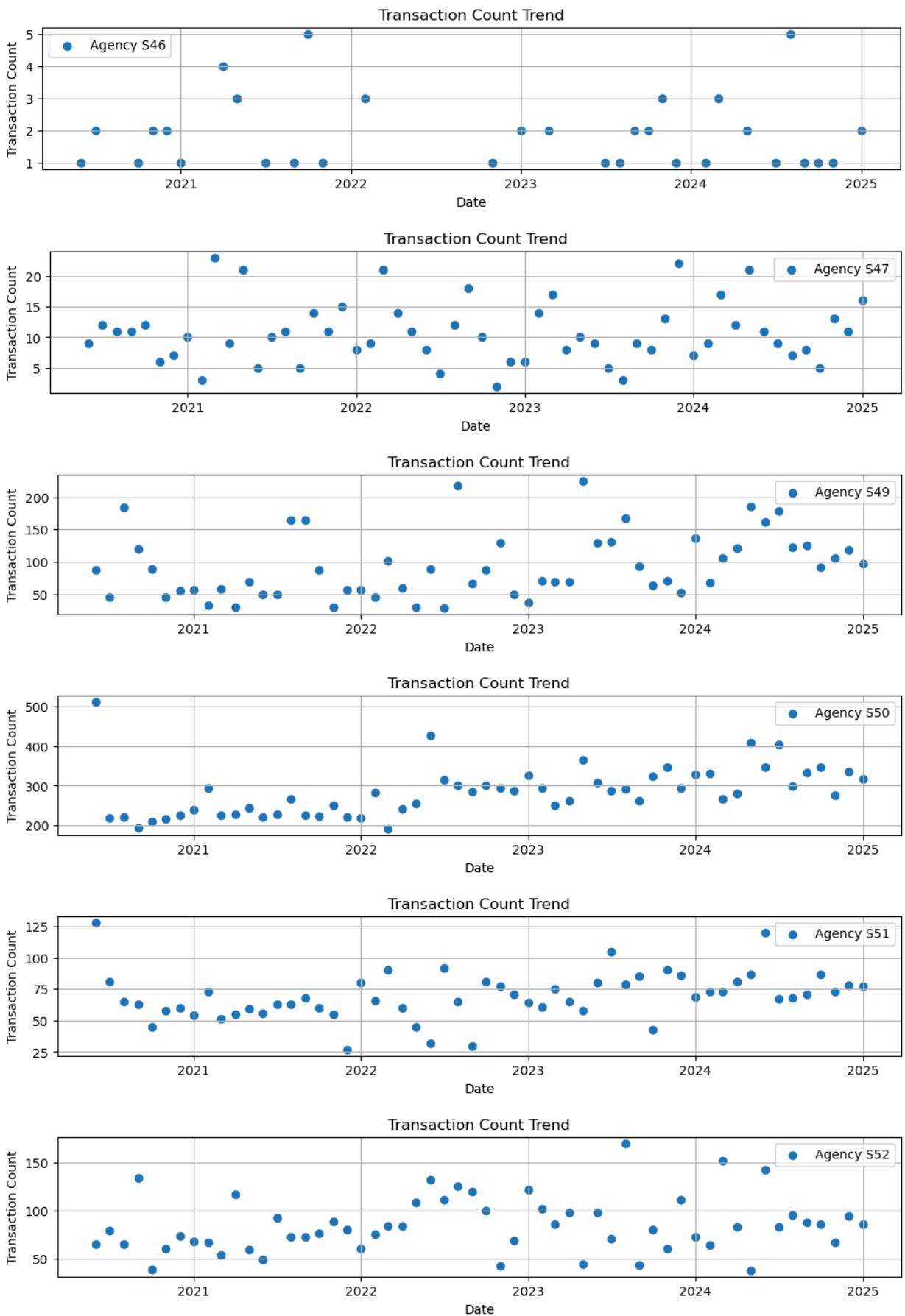


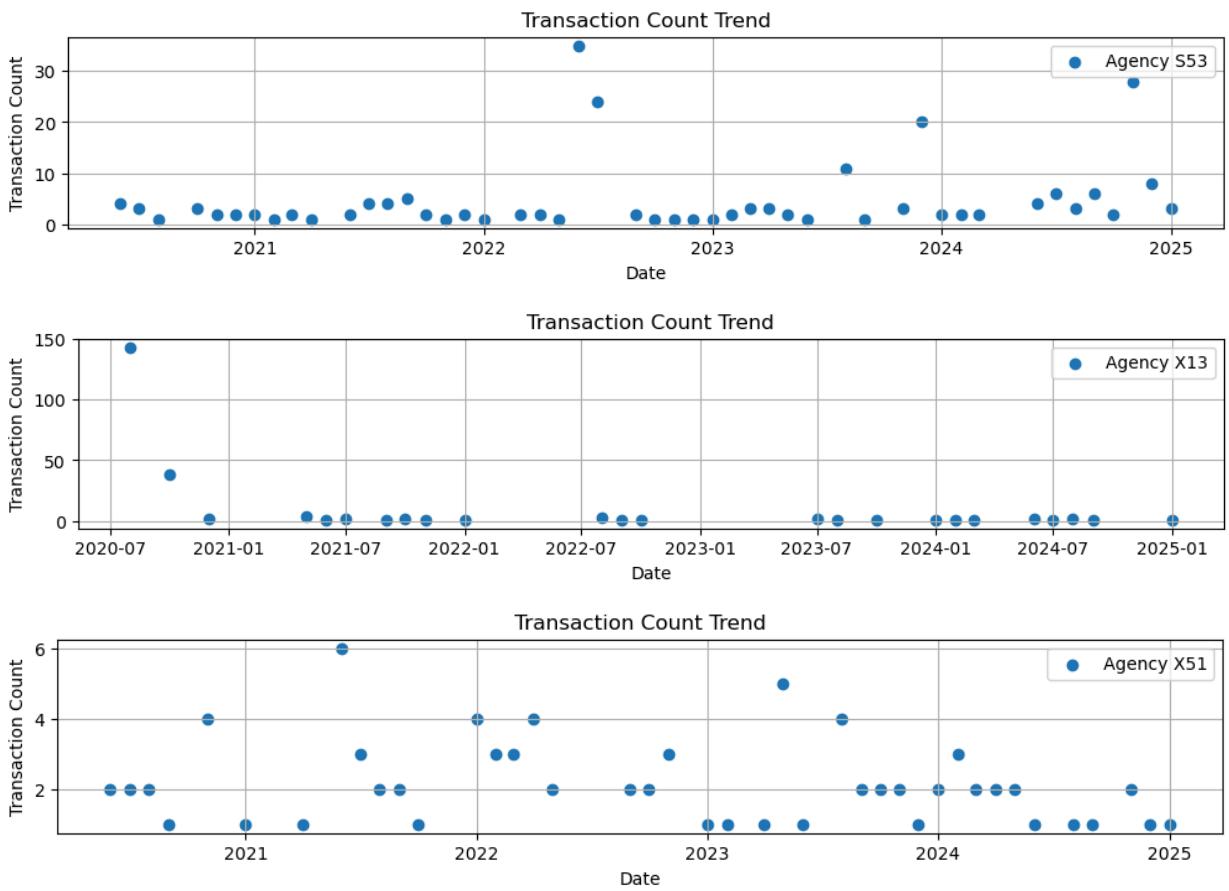










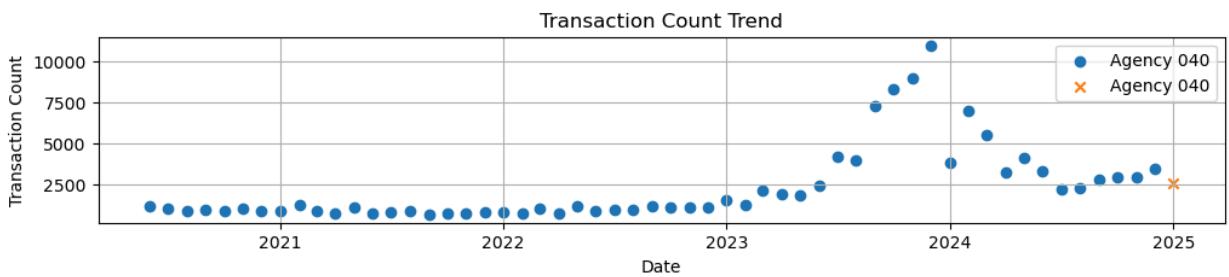


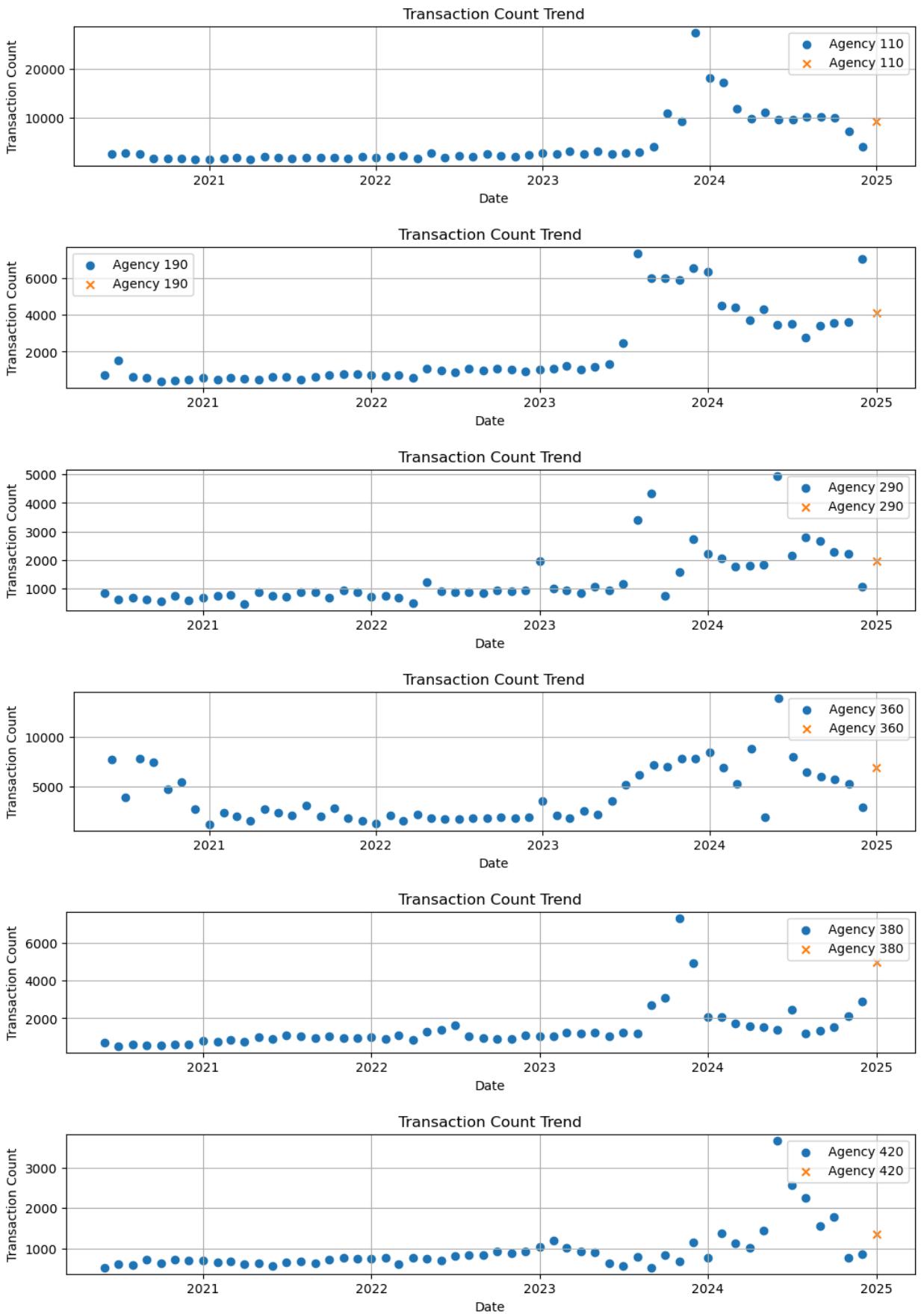
17

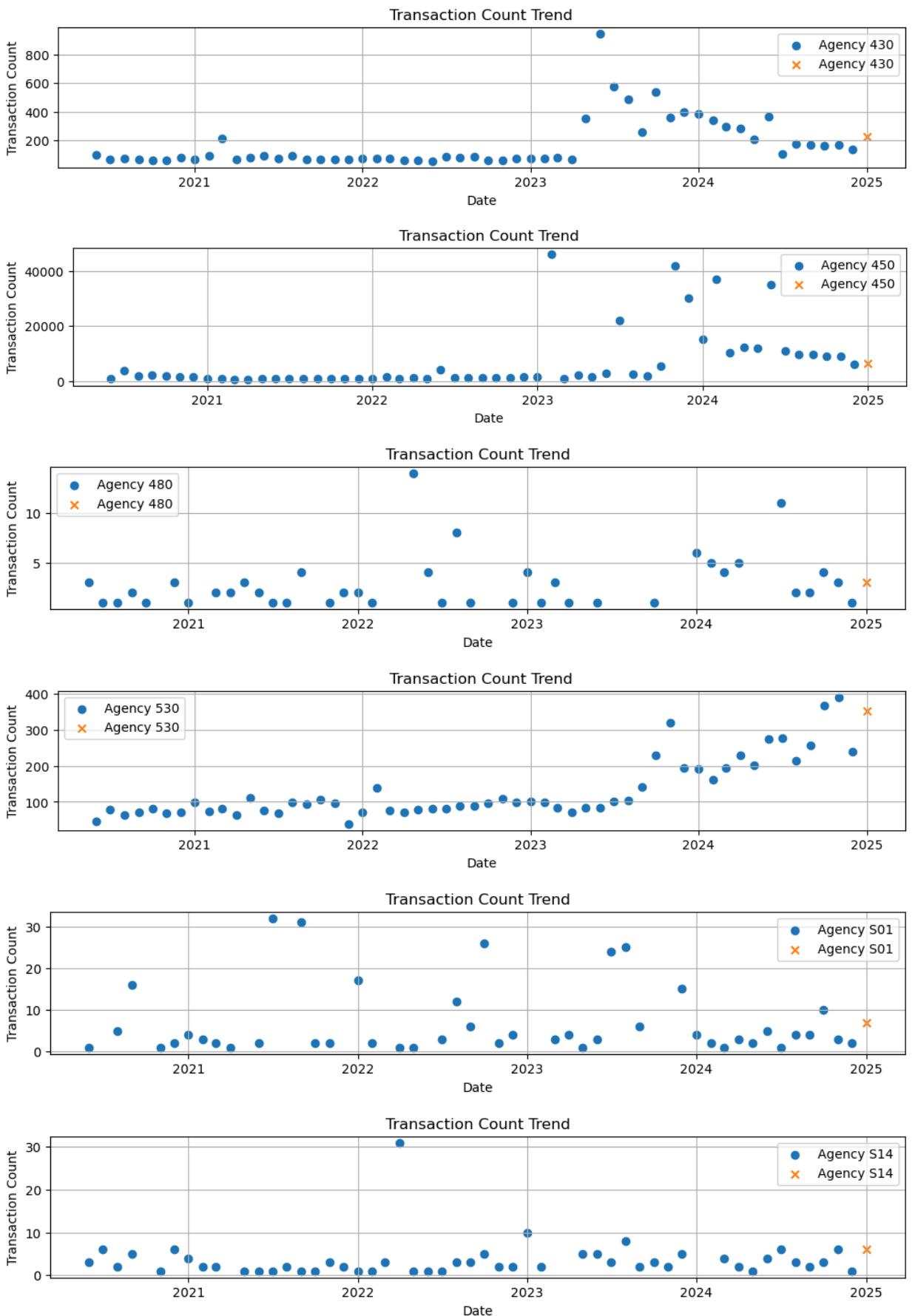
17 - Predicted as Outlier

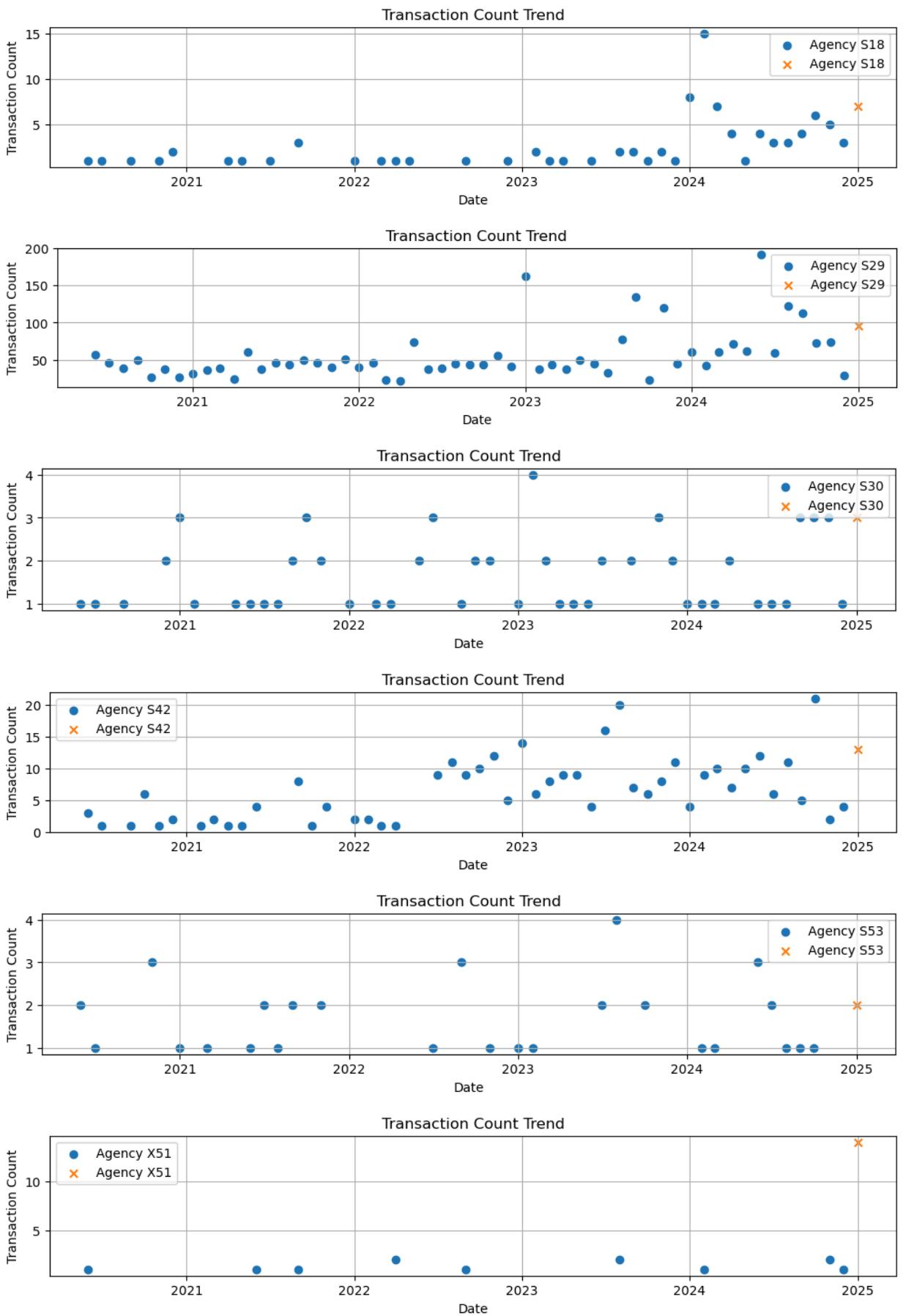
```
In [179]: outlier_df_17 = combined.query("status=='new' and tpmi_trans_cd=='17' and outlier== -1")
print(' '.join(sorted(outlier_df_17.tpmi_agency_cd.unique())))
040 110 190 290 360 380 420 430 450 480 530 S01 S14 S18 S29 S30 S42 S53 X51
```

```
In [180]: for agency in sorted(outlier_df_17.tpmi_agency_cd.unique()):
    show_outliers(combined, '17', [agency])
```





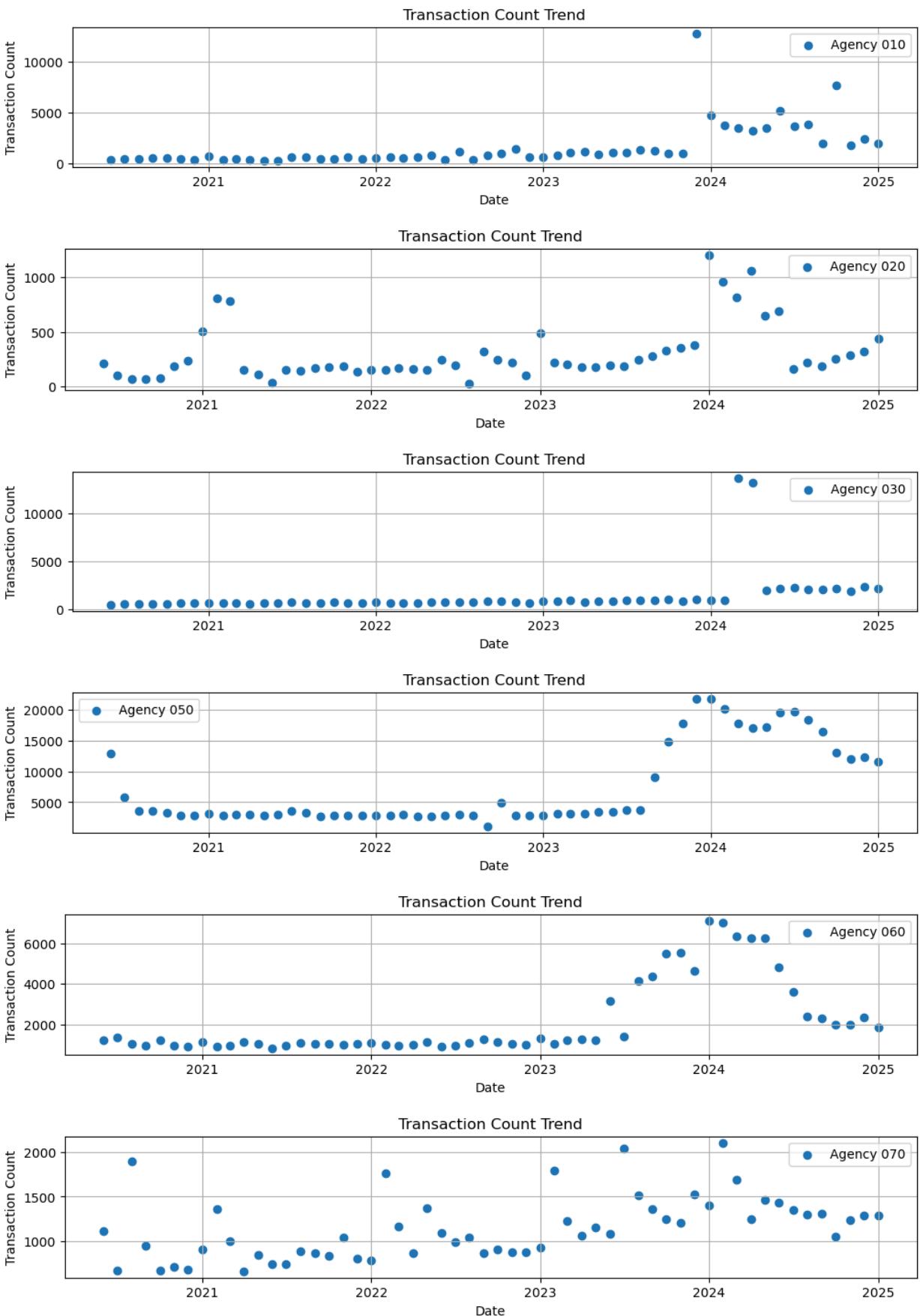


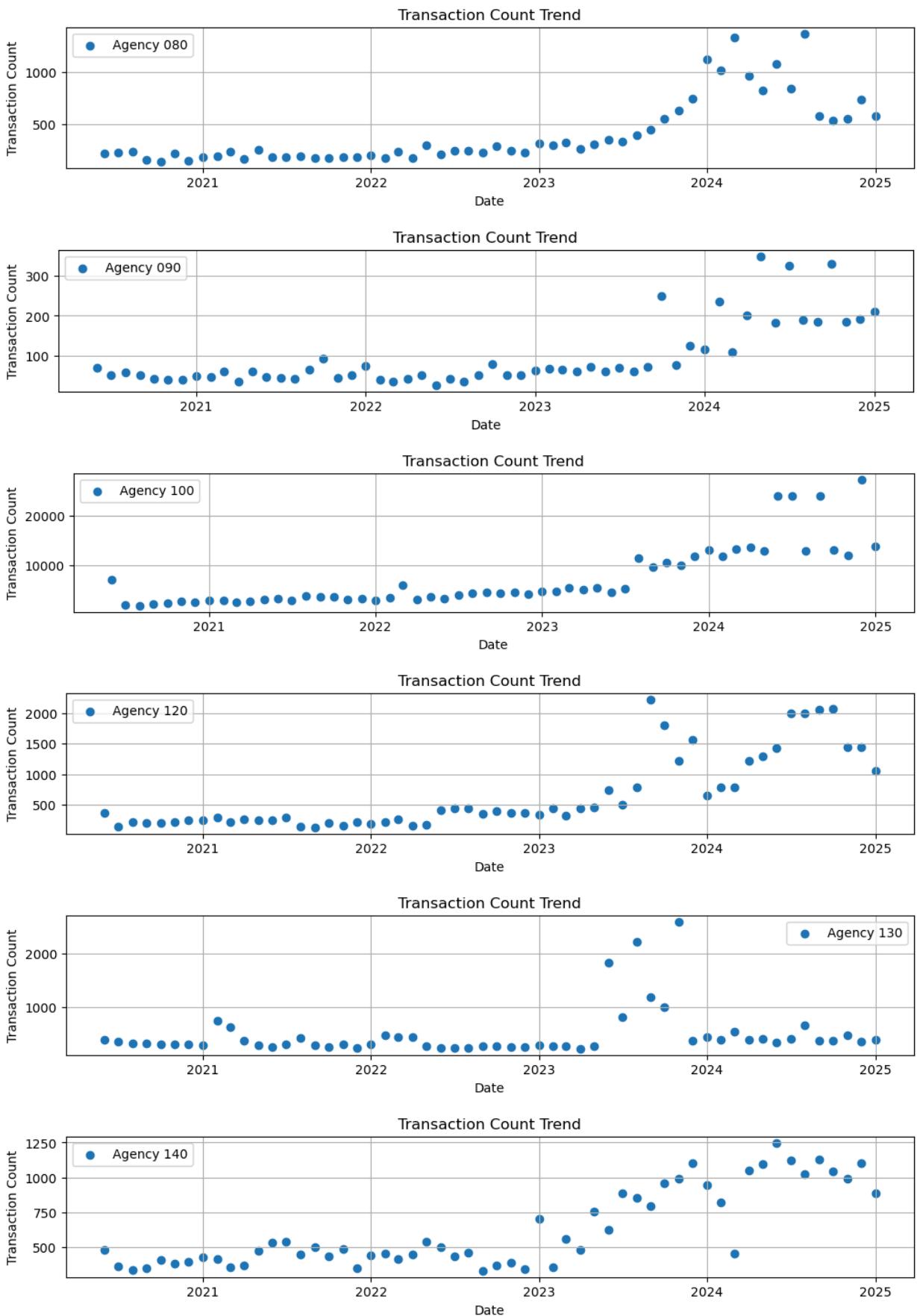


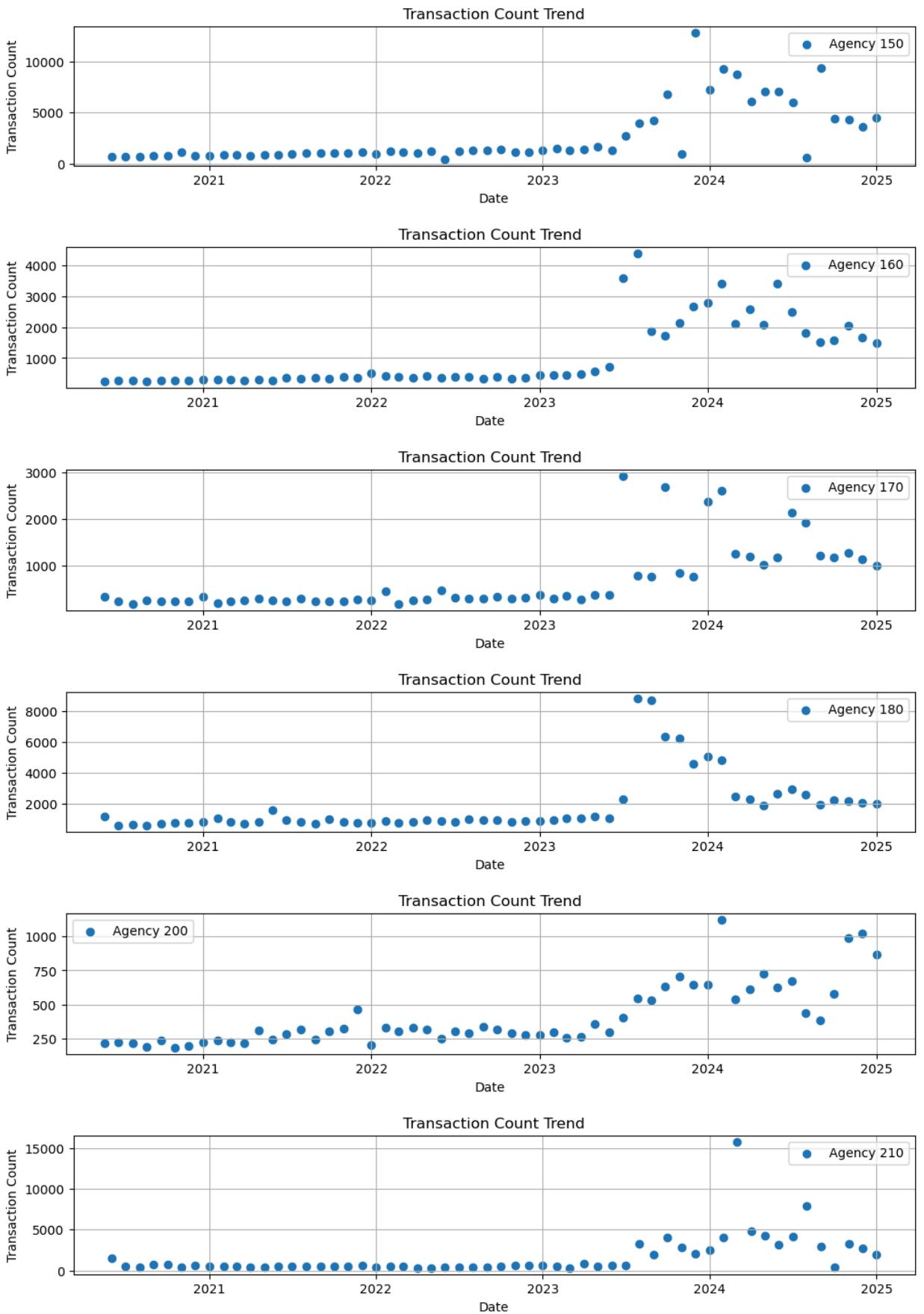
17 - Predicted as Normal

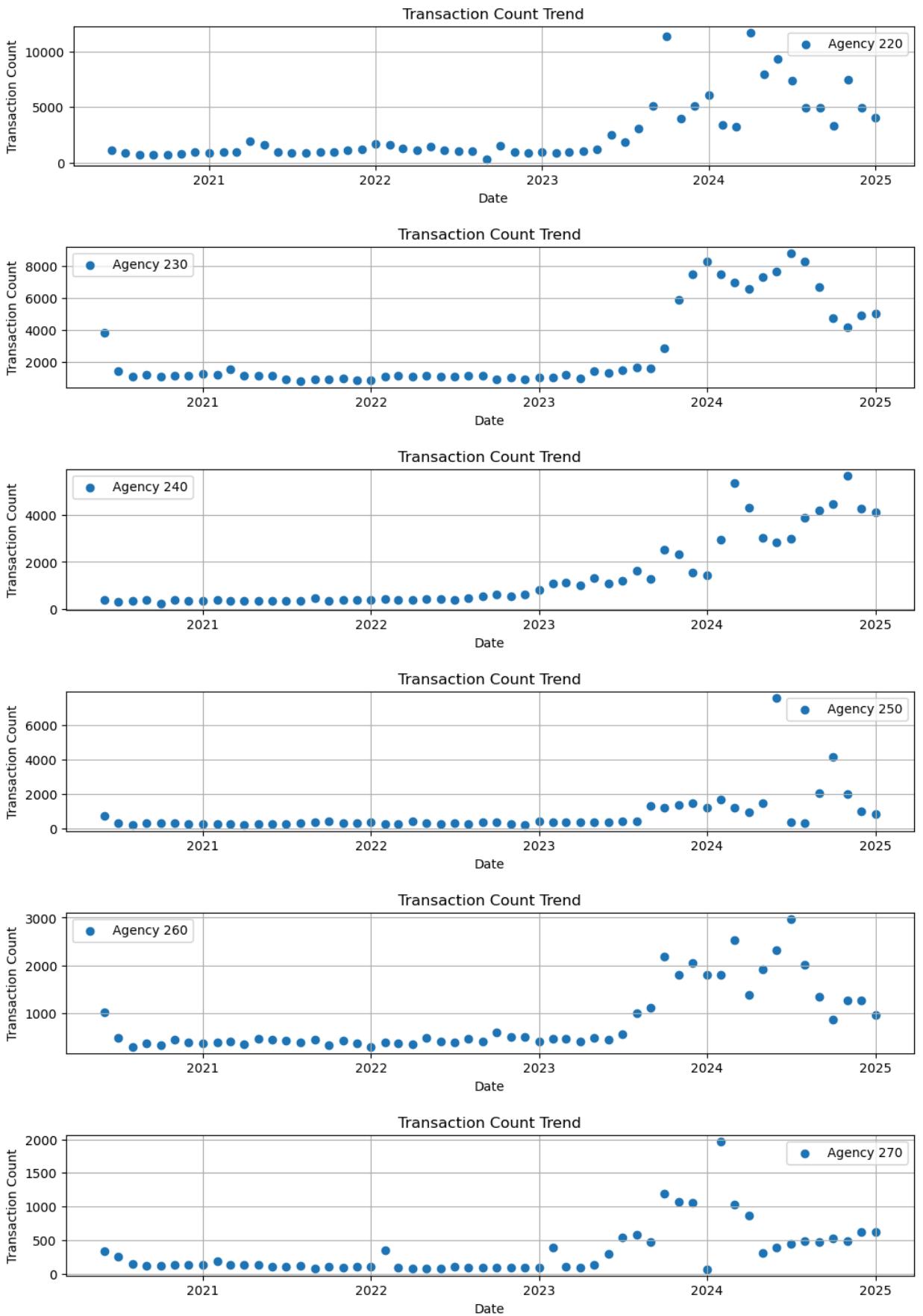
In [181...]

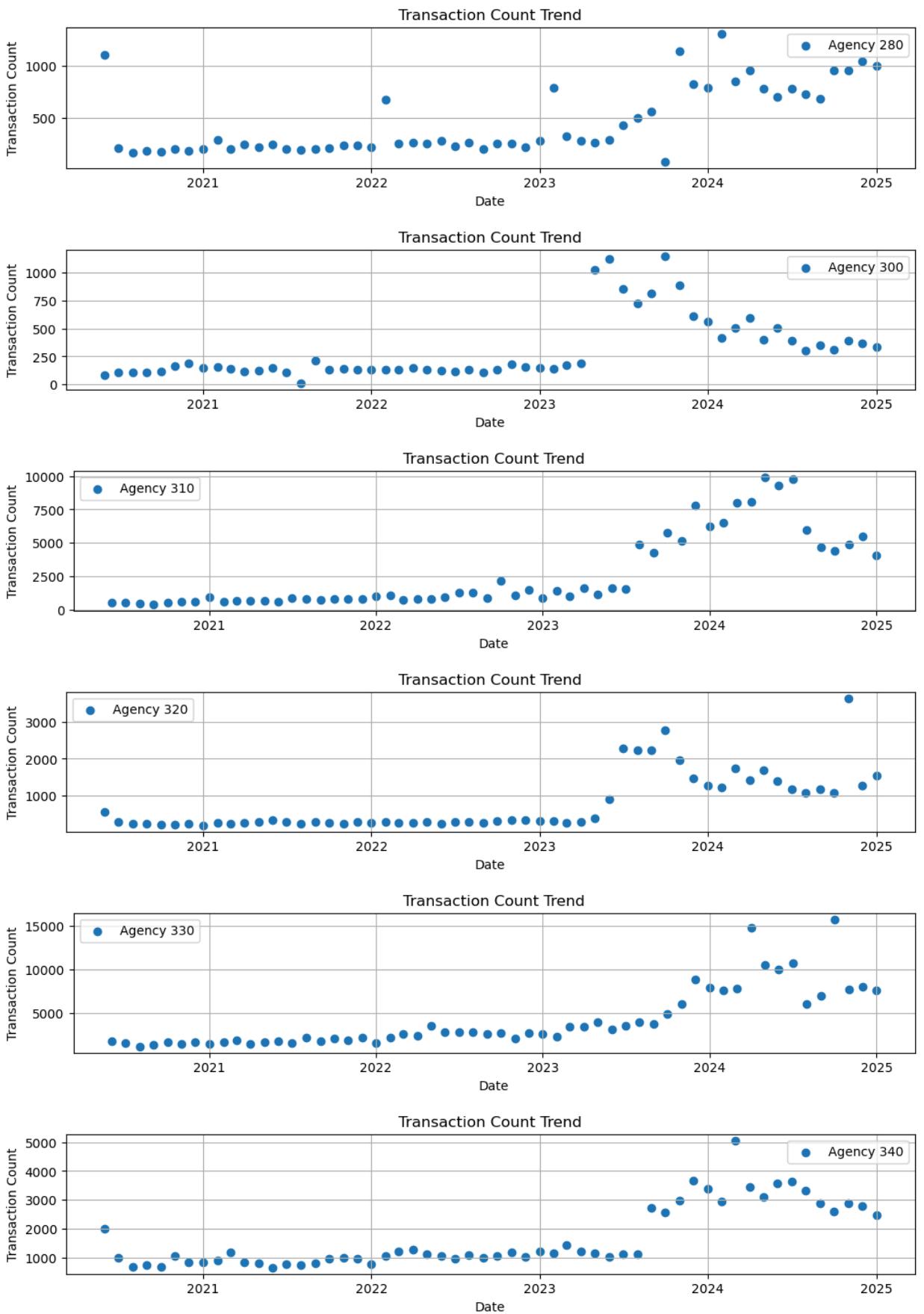
```
for agency in sorted(combined.query("status=='new' and tpmi_trans_cd=='17' and outlier  
show_outliers(combined, '17', [agency])
```

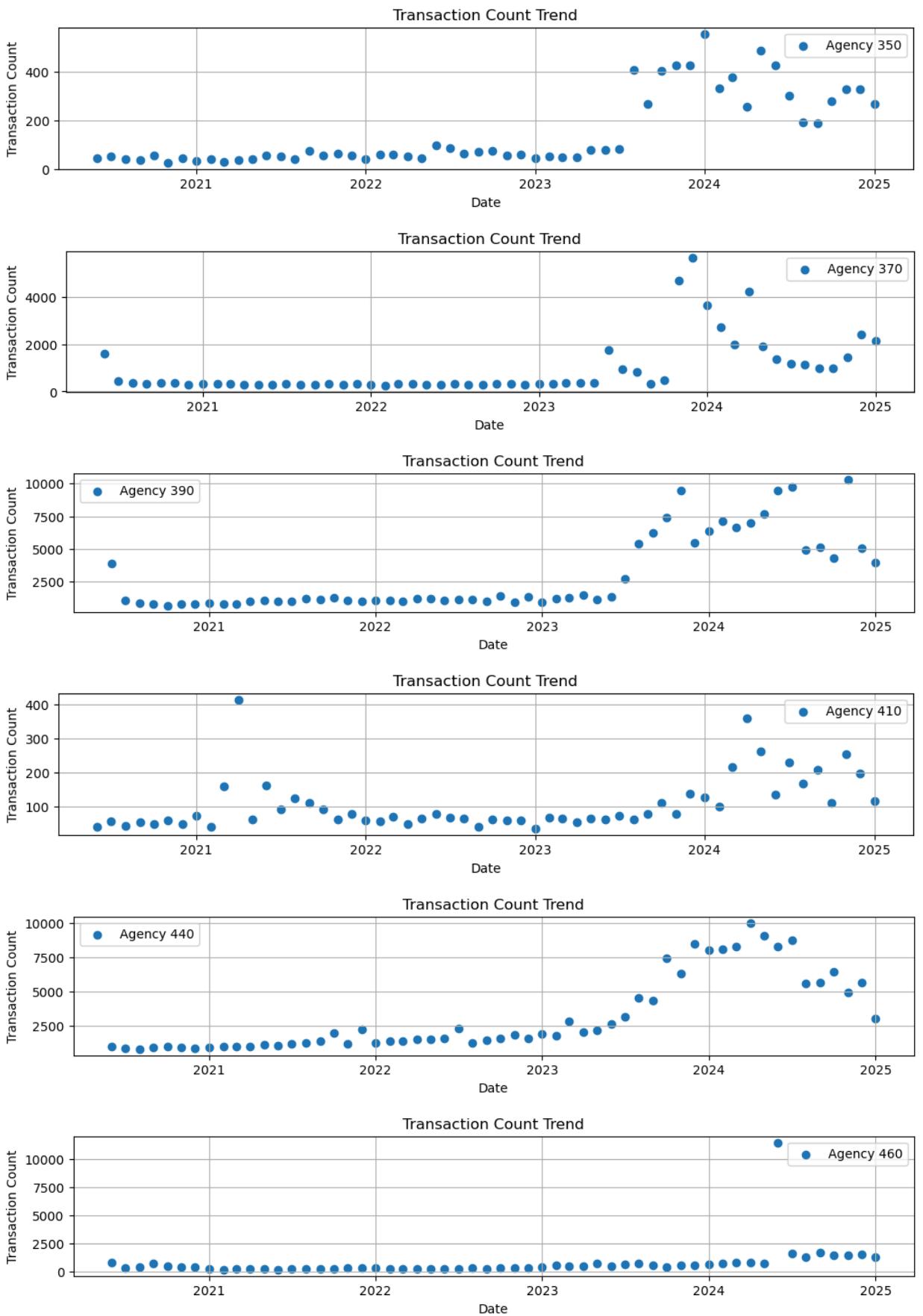


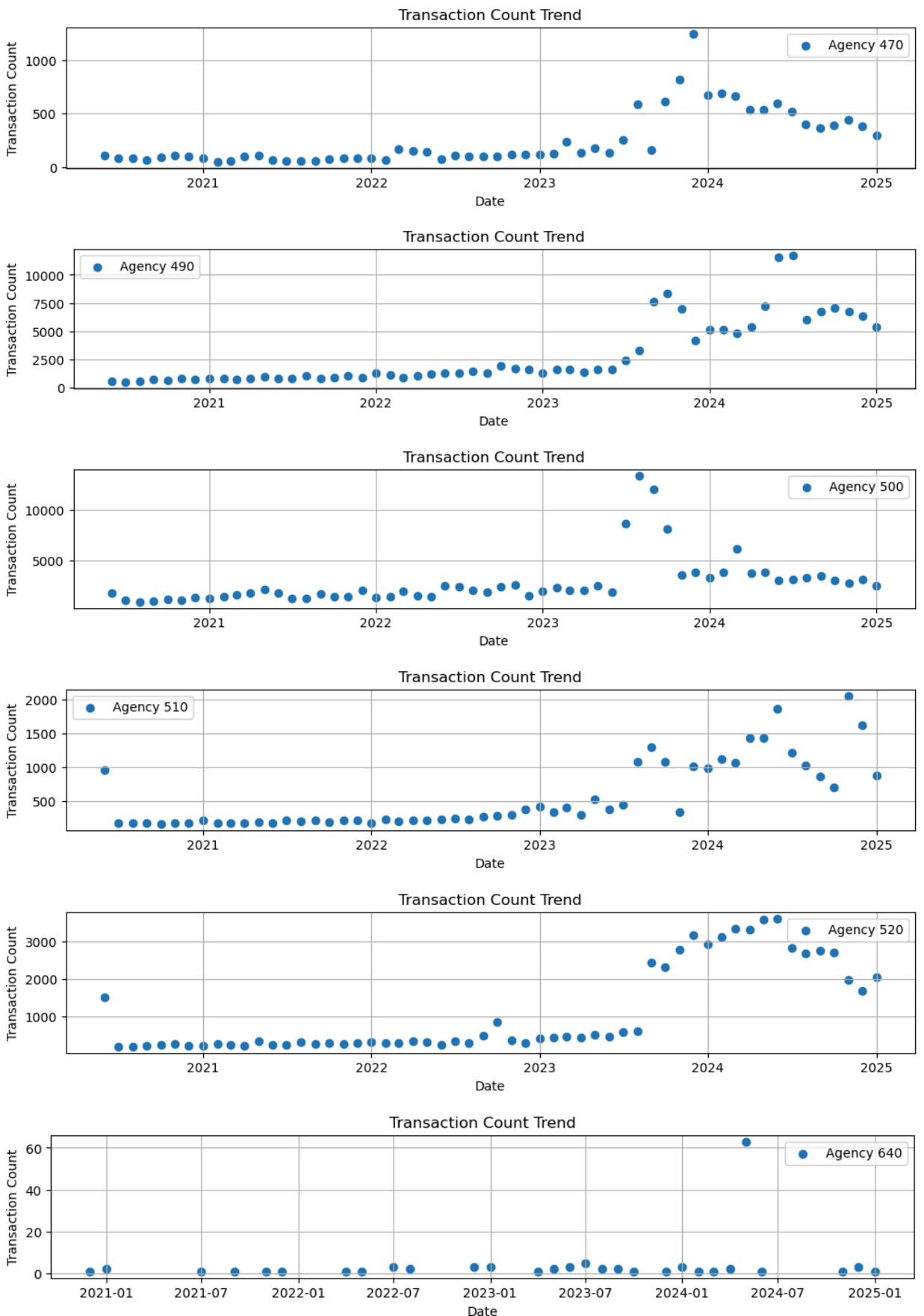


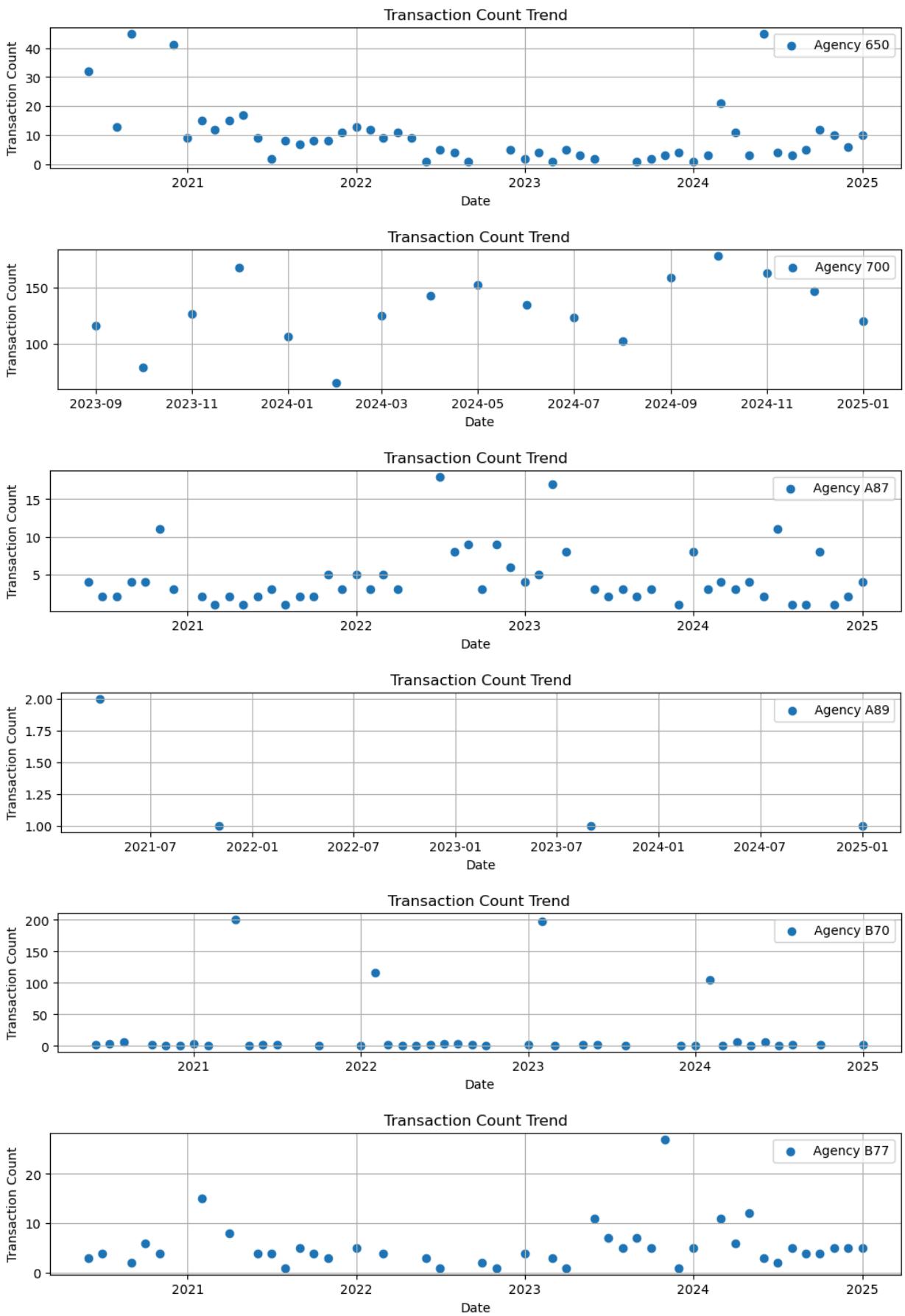


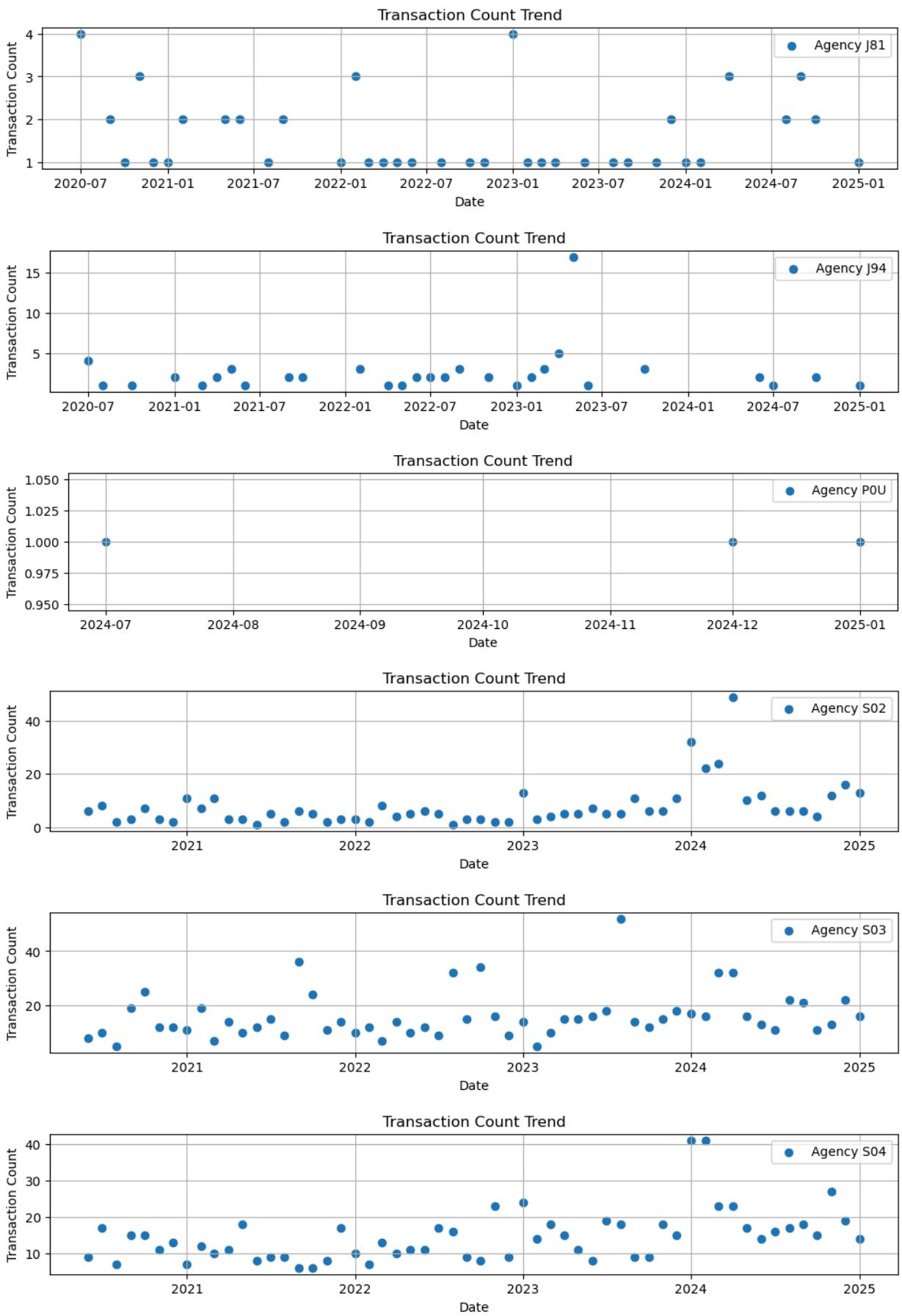


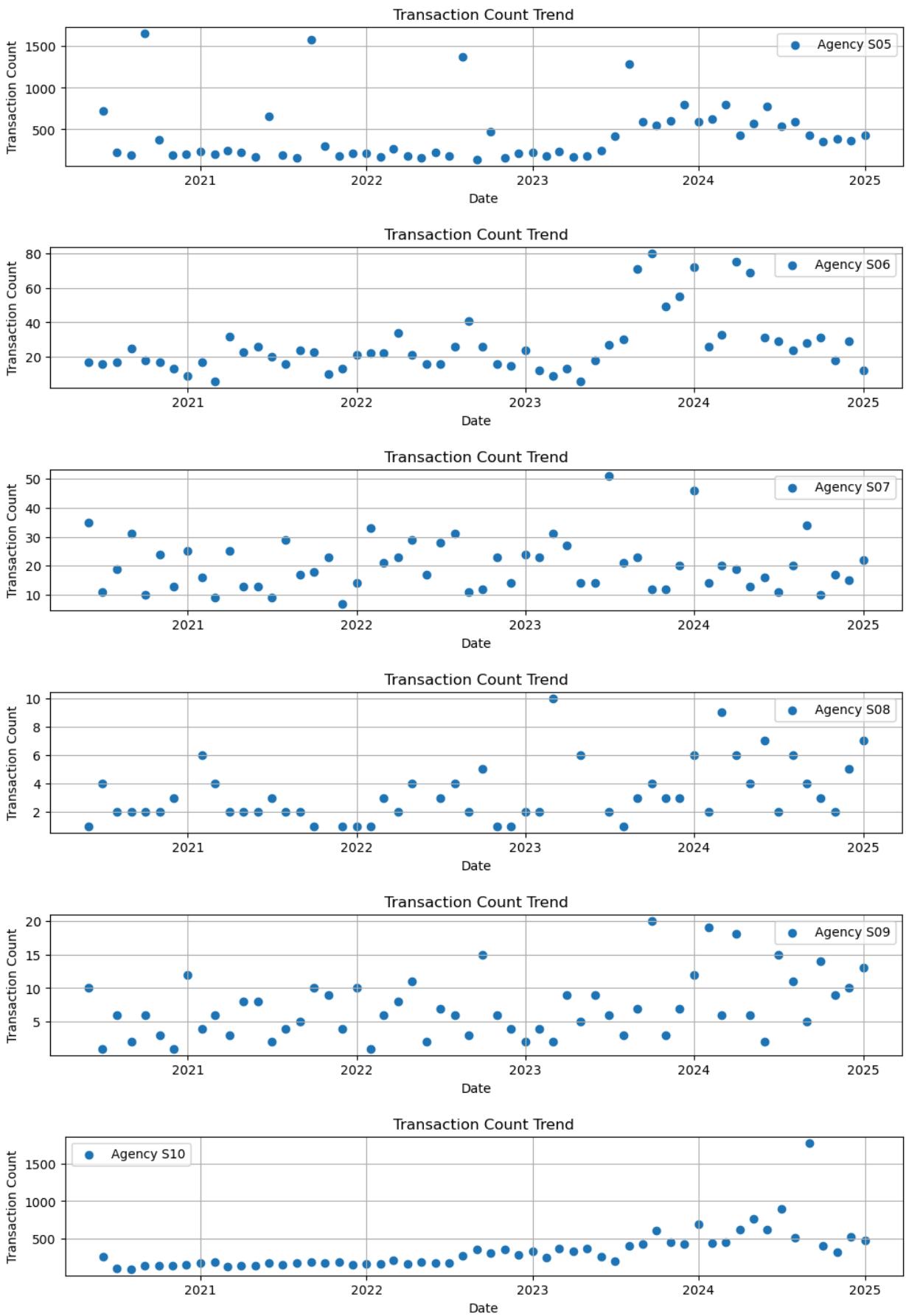


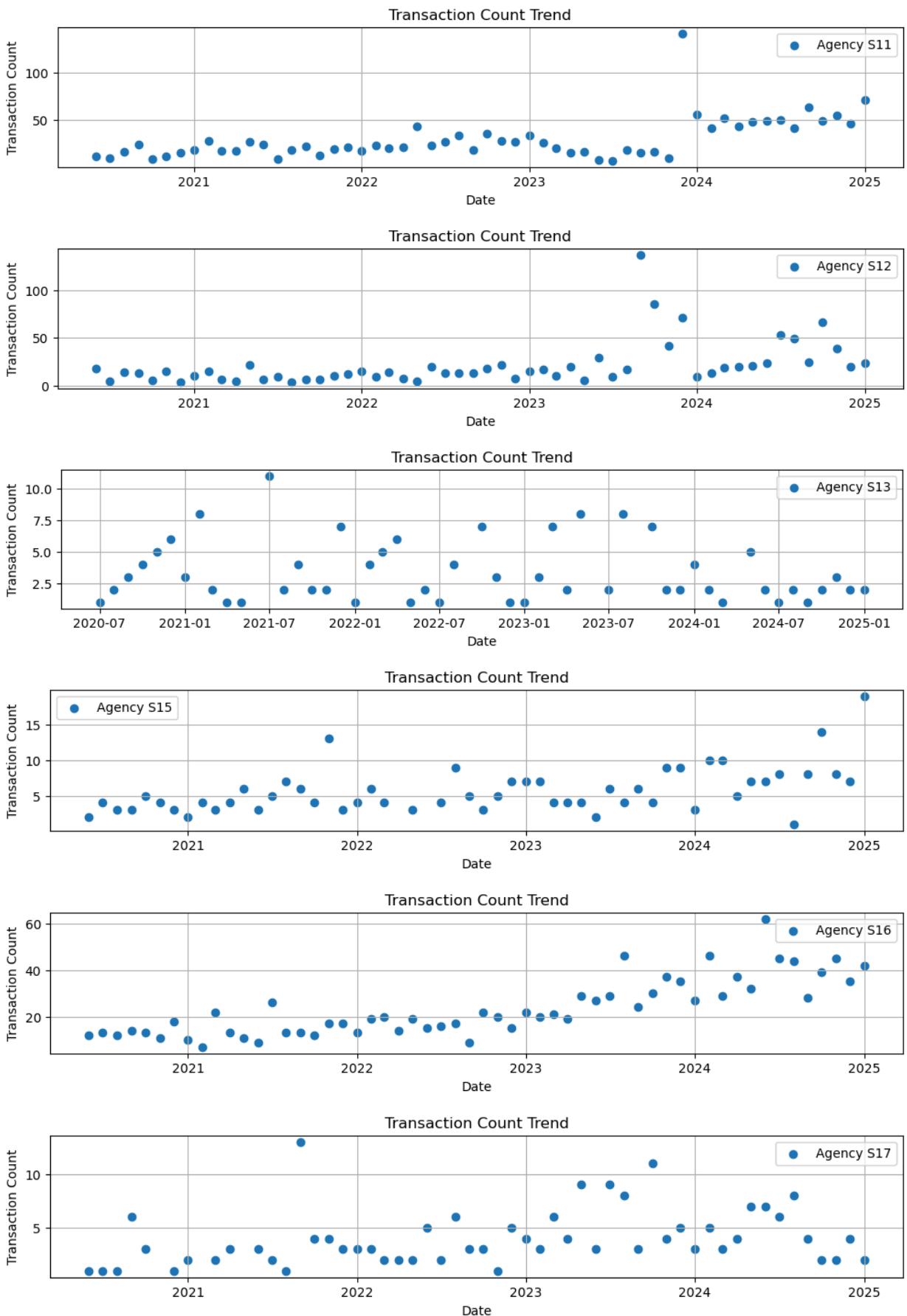


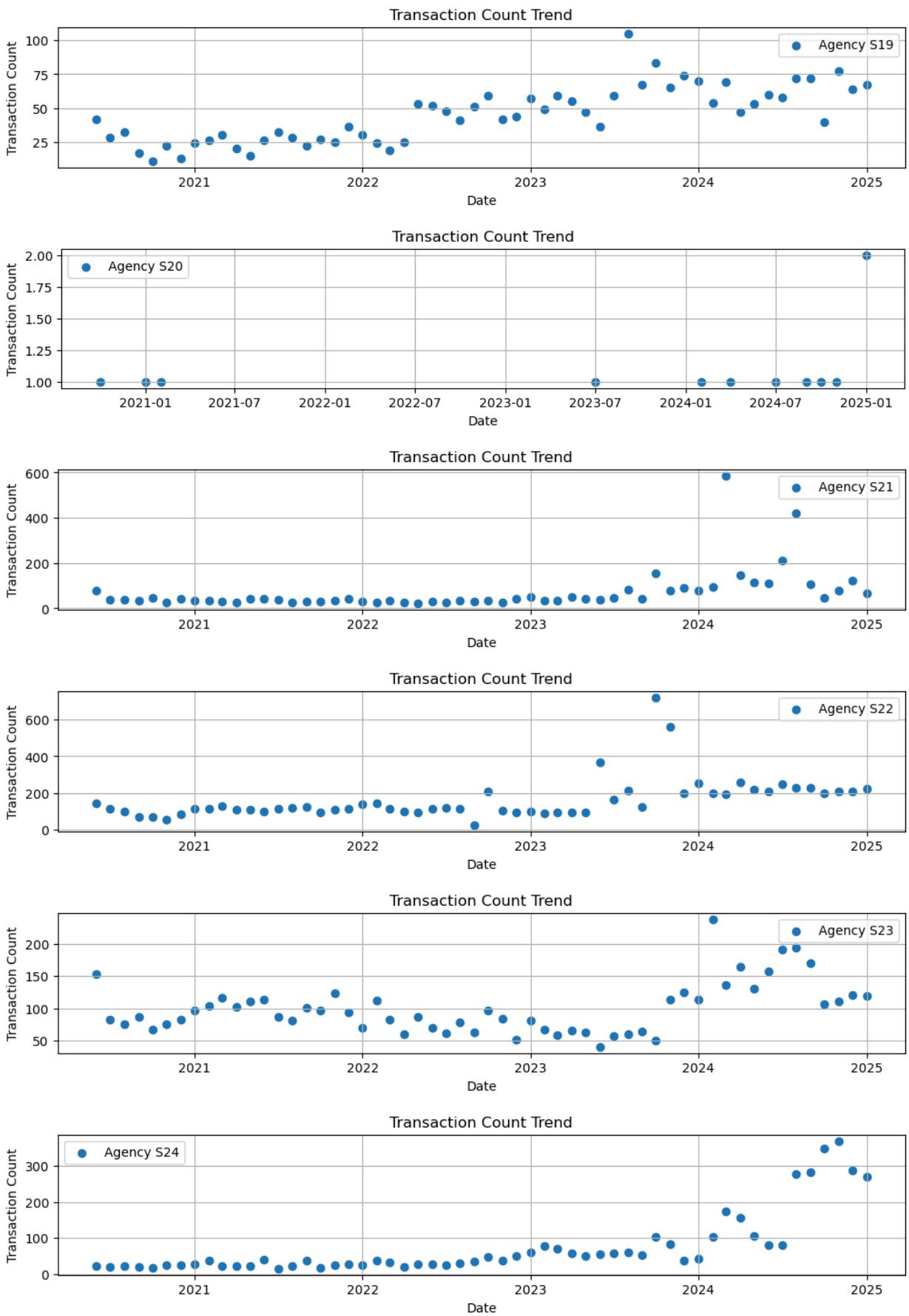


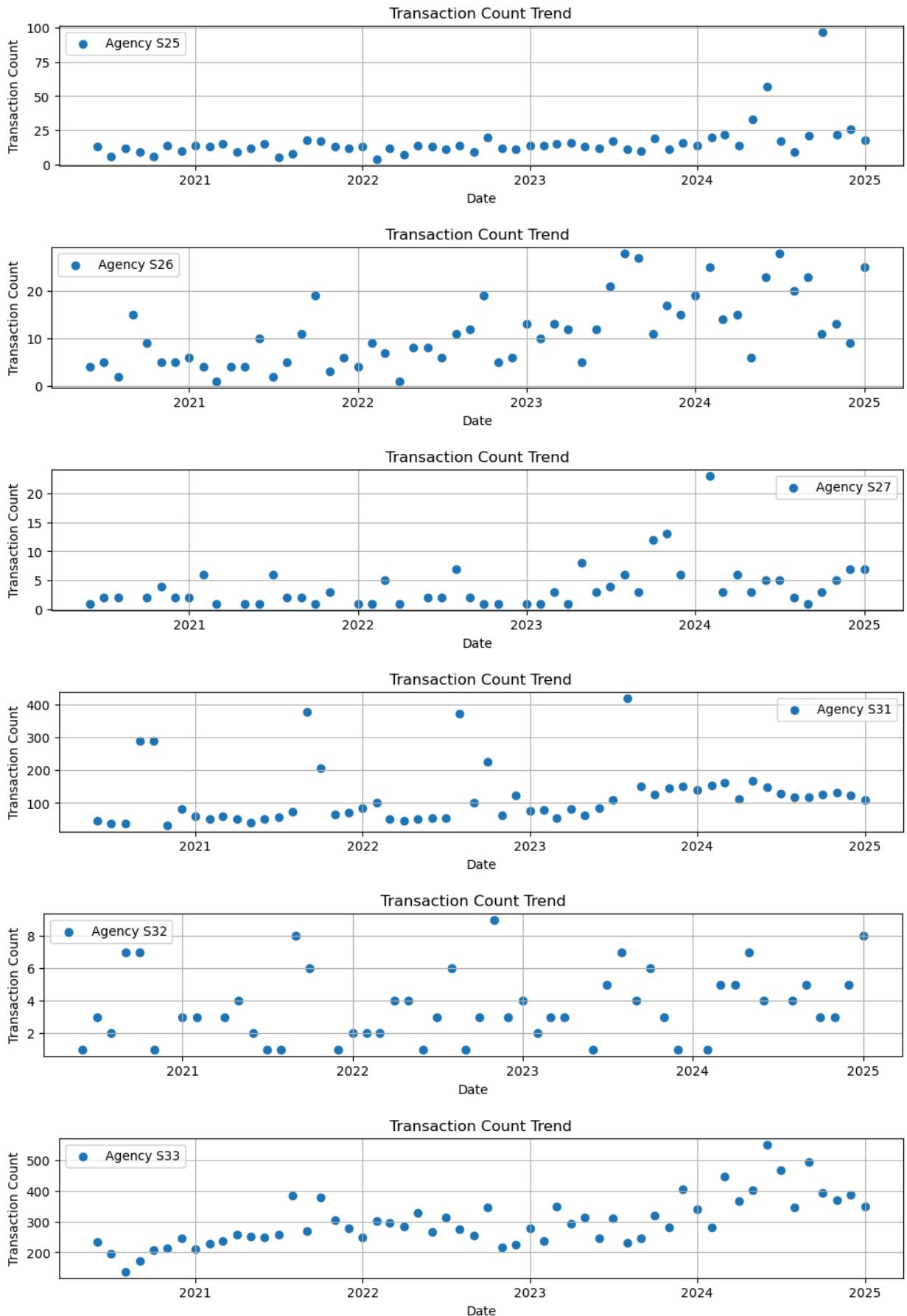


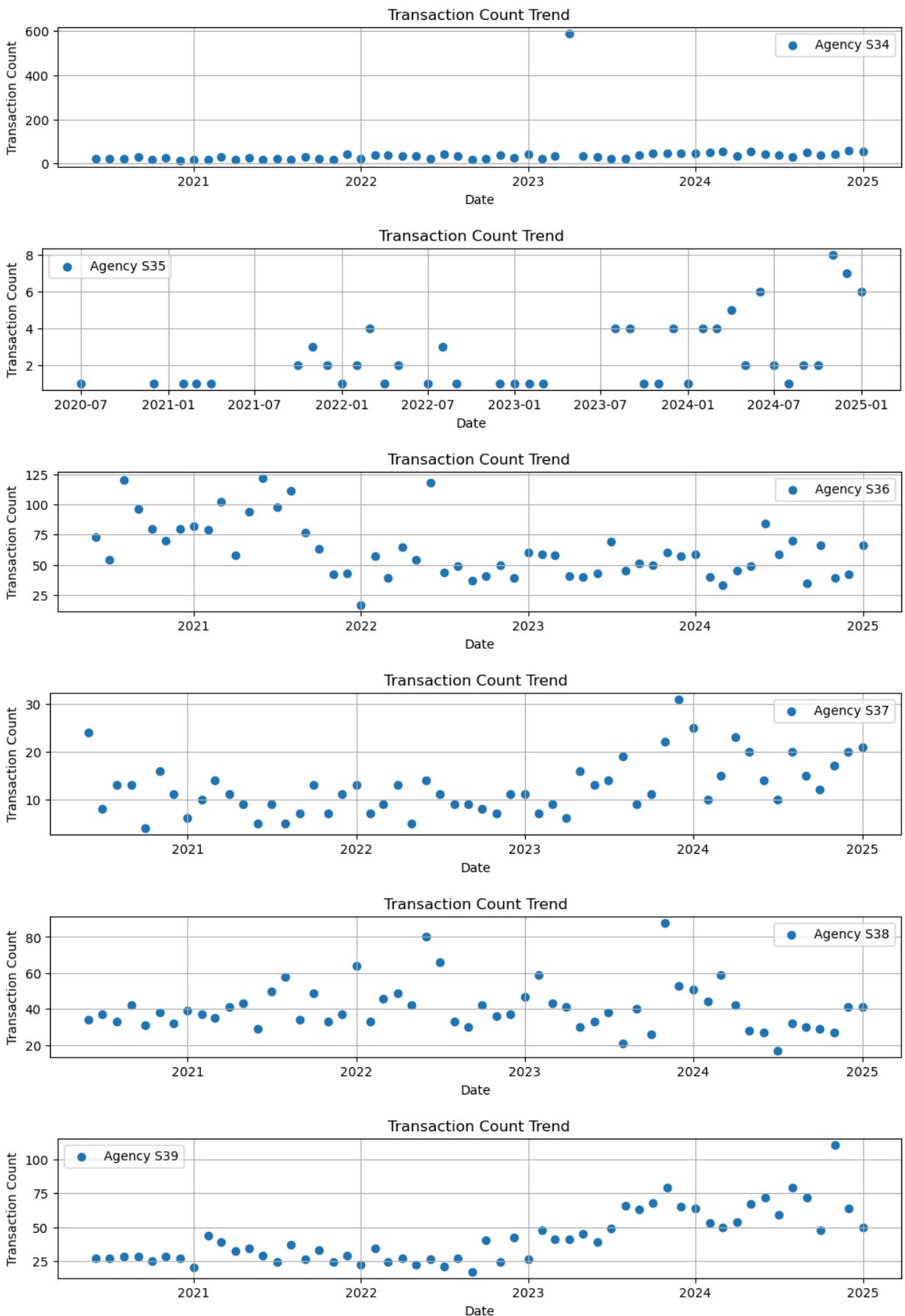


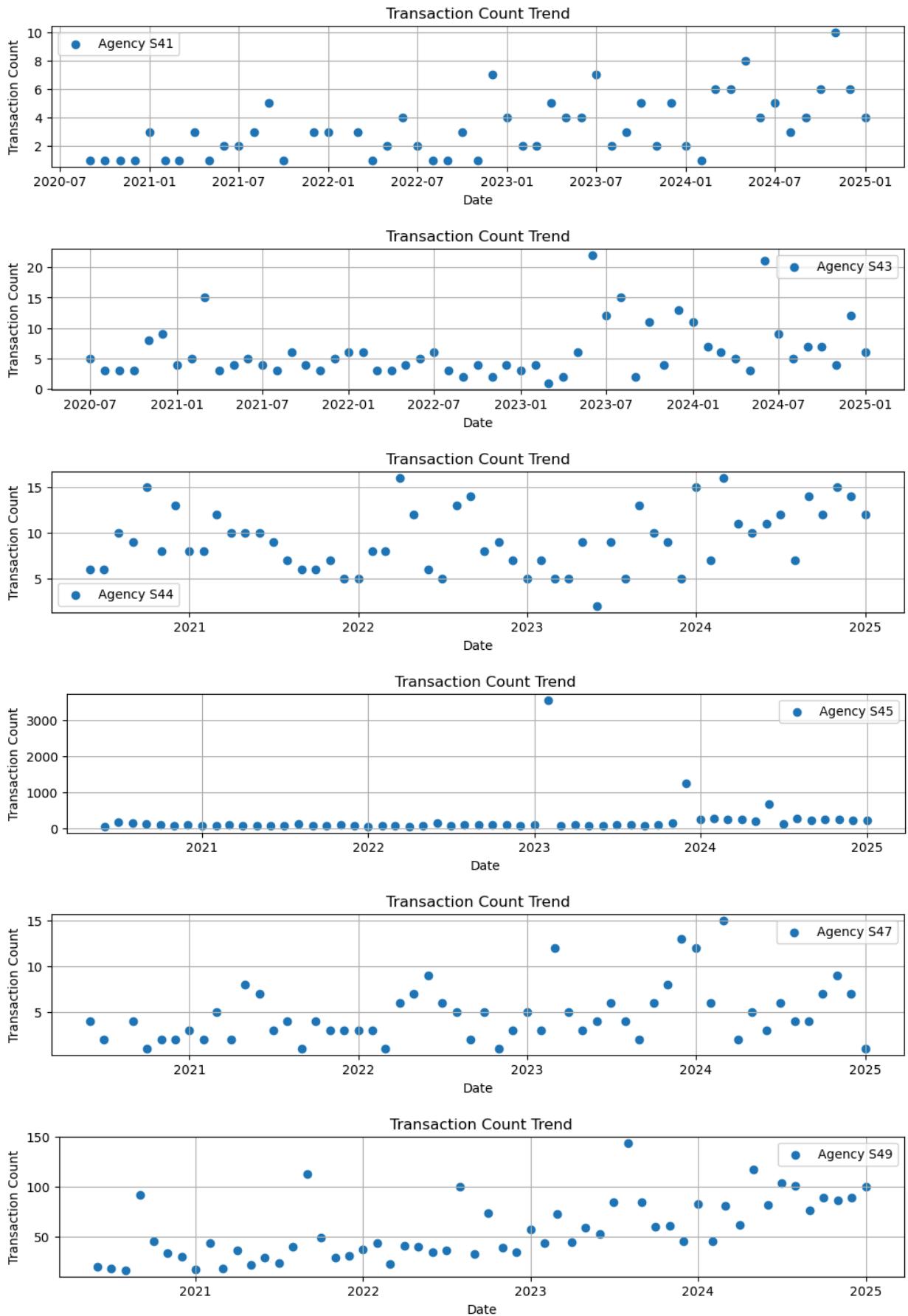


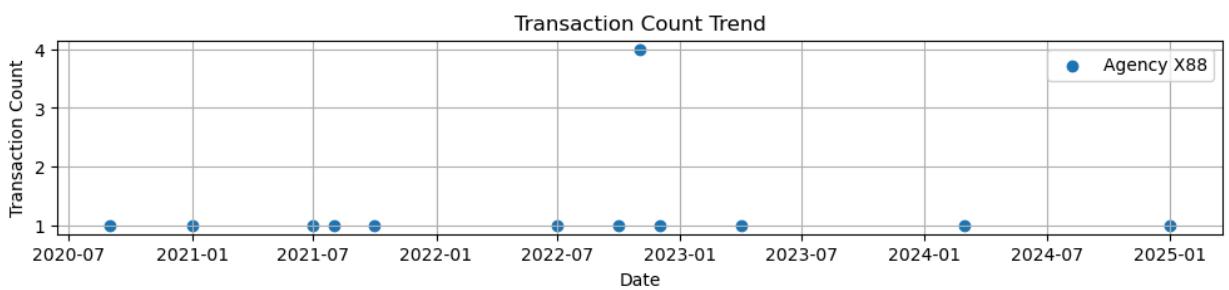
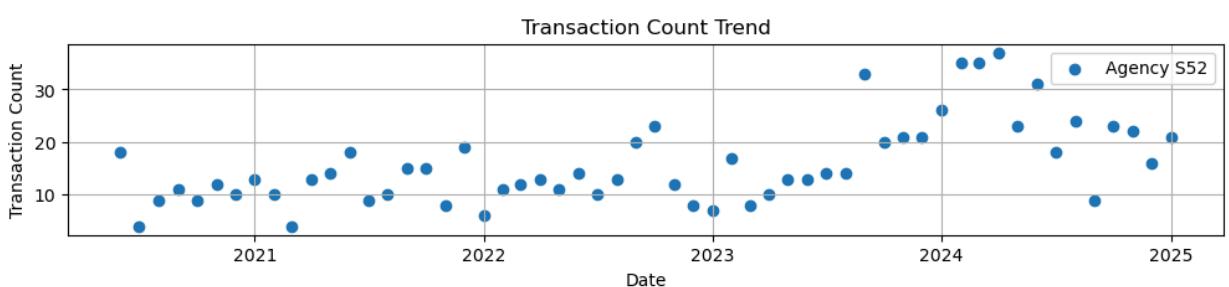
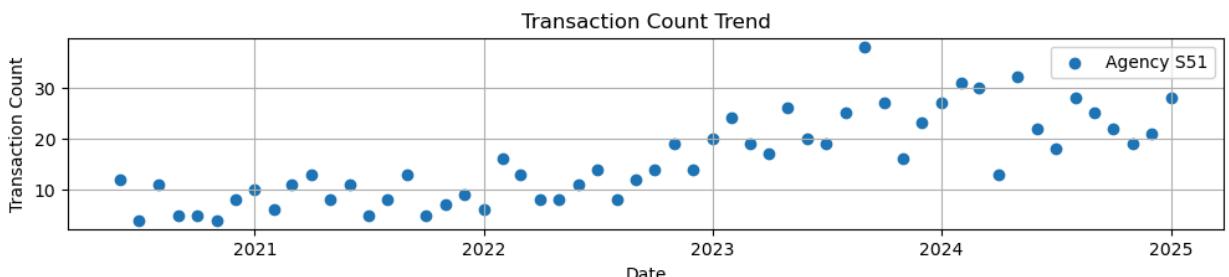
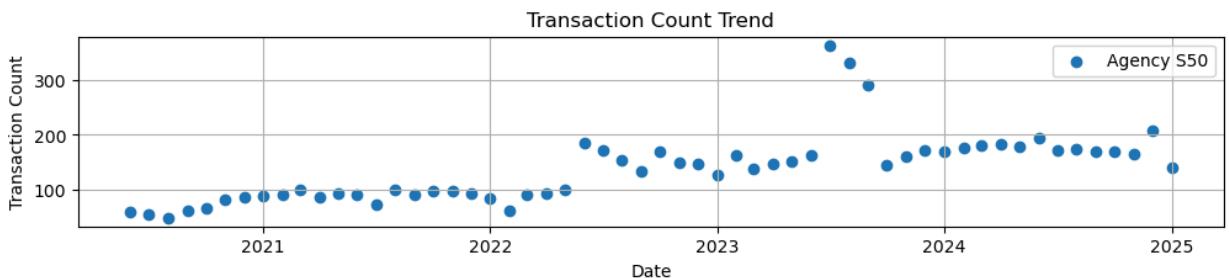












In []:

In []: