analysis10-2021

August 6, 2025

1 Colorado Spills

1.1 OLS, NBR, Interrupted Time Series Analysis

1.2 Just top 3 Counties

1.2.1 Author: David P. Adams, PhD

Date: 2025-08-06

---

2 Setup

[30]:
```
import os
import pandas as pd
import geopandas as gpd
from sqlalchemy import create_engine
from dotenv import load_dotenv

# Load environment variables (e.g., DB_USER, DB_PASSWORD)
load_dotenv()

# Database credentials from .env or shell
db_name = 'colorado_spills'
user = os.getenv('DB_USER')
password = os.getenv('DB_PASSWORD')
host = os.getenv('DB_HOST')
port = os.getenv('DB_PORT')

# SQLAlchemy engine for PostgreSQL/PostGIS
engine = create_engine(f'postgresql+psycopg2://{user}:{password}@{host}:{port}/
 ↪{db_name}')

# --- Load data with geometry preserved ---
try:
    spills = gpd.read_postgis(
        sql='SELECT * FROM spills_with_ruca',
        con=engine,
```

```
        geom_col='geometry',
        crs='EPSG:4326'
    )
except Exception as e:
    print("GeoPandas failed to load geometry. Falling back to plain Pandas.")
    spills = pd.read_sql('SELECT * FROM spills_with_ruca', engine)
    # Optional: spills.drop(columns='geometry', inplace=True, errors='ignore')
```

[31]:
```
# use longitude and latitude to create a GeoDataFrame from spills data
spills['geometry'] = gpd.points_from_xy(spills['Longitude'], spills['Latitude'])
```

[32]:
```
spills_gdf = gpd.GeoDataFrame(spills, crs='EPSG:4326')
```

[38]:
```
# Ensure date columns are in datetime format
spills_gdf['Date of Discovery'] = pd.to_datetime(spills_gdf['Date of␣
 ↪Discovery'])
spills_gdf['Initial Report Date'] = pd.to_datetime(spills_gdf['Initial Report␣
 ↪Date'])

# create a report year column
spills_gdf['Report Year'] = spills_gdf['Date of Discovery'].dt.year
```

[39]:
```
from datetime import datetime

# 48 months before and after January 1, 2021
start_date = datetime(2017, 7, 1)
end_date = datetime(2025, 7, 1)  # exclusive end date

spills_gdf = spills_gdf[
    (spills_gdf['Date of Discovery'] >= start_date) &
    (spills_gdf['Date of Discovery'] < end_date)
]
```

[40]:
```
# Separate Historical vs. Recent Spills
historical_spills = spills_gdf[spills_gdf['Spill Type'] == 'Historical']
recent_spills = spills_gdf[spills_gdf['Spill Type'] == 'Recent']
```

[41]:
```
# only use the top 3 counties by number of spills
top_counties = spills_gdf['county'].value_counts().nlargest(3).index.tolist()
# filter spills_gdf to only include top counties
spills_gdf = spills_gdf[spills_gdf['county'].isin(top_counties)]
# filter historical spills to only include top counties
historical_spills = historical_spills[historical_spills['county'].
 ↪isin(top_counties)]
# filter recent spills to only include top counties
recent_spills = recent_spills[recent_spills['county'].isin(top_counties)]
```

[50]:
```
import statsmodels.formula.api as smf

# Create 'Period' column
spills_gdf['Period'] = spills_gdf['Report Year'].apply(lambda x: 'Before 2021' if x < 2021 else '2021 and After')
```

[51]:
```
spills_gdf['Initial Report Date'] = pd.to_datetime(spills_gdf['Initial Report Date'])
spills_gdf['Date of Discovery'] = pd.to_datetime(spills_gdf['Date of Discovery'])
```

[52]:
```
spills_gdf['report_delay'] = (spills_gdf['Initial Report Date'] - spills_gdf['Date of Discovery']).dt.days
```

[53]:
```
spills_gdf = spills_gdf[spills_gdf['report_delay'] >= 0]
```

[54]:
```
spills_gdf = spills_gdf.rename(columns={
    'Report Delay (Days)': 'report_delay',
    'Spill Type': 'spill_type'
})
```

---

3 OLS Regression

[55]:
```
import statsmodels.formula.api as smf

model = smf.ols("report_delay ~ C(spill_type) * C(Period)", data=spills_gdf).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           report_delay   R-squared:                       0.005
Model:                            OLS   Adj. R-squared:                  0.005
Method:                 Least Squares   F-statistic:                     19.14
Date:                Wed, 06 Aug 2025   Prob (F-statistic):           2.26e-12
Time:                        09:11:58   Log-Likelihood:                -53847.
No. Observations:               11196   AIC:                         1.077e+05
Df Residuals:                   11192   BIC:                         1.077e+05
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                                                    coef    std err
t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
```

```
-----------------------------------
Intercept                                          6.7689      0.478
14.166      0.000       5.832       7.706
C(spill_type)[T.Recent]                           -4.9332      0.691
-7.139      0.000      -6.288      -3.579
C(Period)[T.Before 2021]                          -3.4361      0.986
-3.484      0.000      -5.369      -1.503
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]     4.3645      1.249
3.496      0.000       1.917       6.812
==============================================================================
Omnibus:                    24335.353   Durbin-Watson:                   1.954
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        152505098.951
Skew:                          19.556   Prob(JB):                         0.00
Kurtosis:                     573.424   Cond. No.                         7.15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

---

4 Negative Binomial Regression

[56]:
```python
import statsmodels.formula.api as smf
import statsmodels.api as sm

# Make sure spill_type and Period are clean, and report_delay is numeric
spills_gdf = spills_gdf.rename(columns={
    'Report Delay (Days)': 'report_delay',
    'Spill Type': 'spill_type'
})

# Fit the Negative Binomial model with interaction
nb_model = smf.glm(
    formula="report_delay ~ C(spill_type) * C(Period)",
    data=spills_gdf,
    family=sm.families.NegativeBinomial()
).fit()

# View the results
print(nb_model.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:           report_delay   No. Observations:                11196
Model:                            GLM   Df Residuals:                    11192
Model Family:        NegativeBinomial   Df Model:                            3
```

```
Link Function:                        Log   Scale:                          1.0000
Method:                              IRLS   Log-Likelihood:                -26493.
Date:                  Wed, 06 Aug 2025   Deviance:                       29512.
Time:                          09:12:11   Pearson chi2:                 4.90e+05
No. Iterations:                       9   Pseudo R-squ. (CS):             0.2067
Covariance Type:              nonrobust
==================================================================================
                                                    coef    std err
z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept                                          1.9123      0.017
110.902      0.000       1.879       1.946
C(spill_type)[T.Recent]                           -1.3049      0.027
-48.162      0.000      -1.358      -1.252
C(Period)[T.Before 2021]                          -0.7085      0.037
-18.968      0.000      -0.782      -0.635
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]   1.1178      0.049
23.044      0.000       1.023       1.213
==================================================================================
```

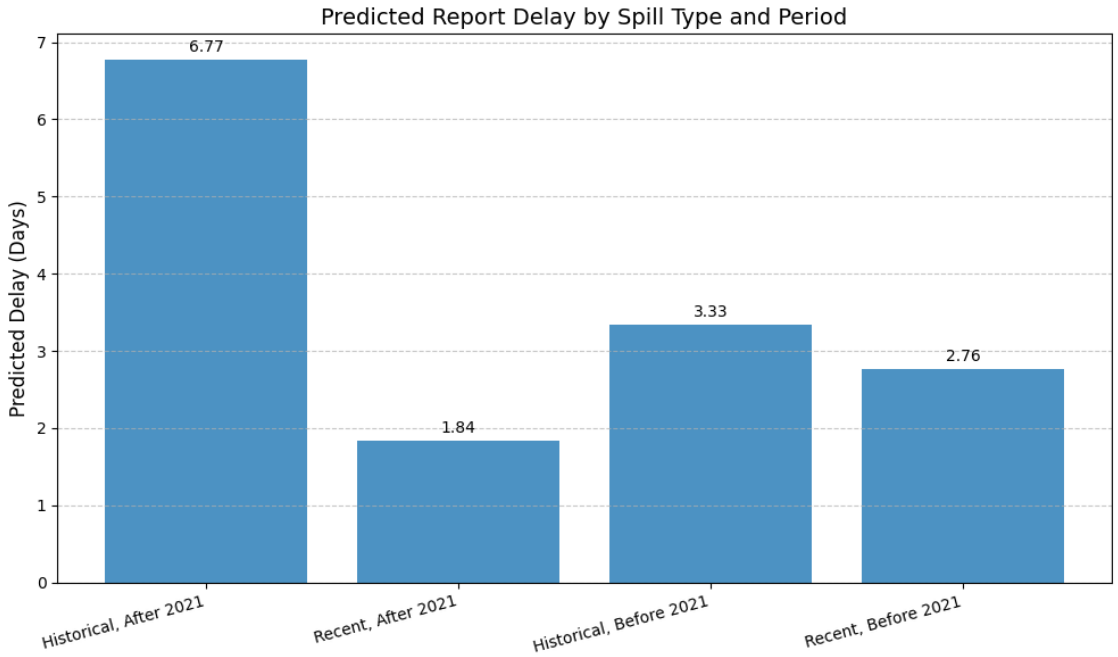
```
/home/dadams/miniconda3/lib/python3.12/site-
packages/statsmodels/genmod/families/family.py:1367: ValueWarning: Negative
binomial dispersion parameter alpha not set. Using default value alpha=1.0.
  warnings.warn("Negative binomial dispersion parameter alpha not "
```

[57]:
```
import numpy as np
np.exp(nb_model.params)
```

[57]:
```
Intercept                                          6.768912
C(spill_type)[T.Recent]                            0.271201
C(Period)[T.Before 2021]                           0.492364
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]   3.058085
dtype: float64
```

[58]:
```
import matplotlib.pyplot as plt
import pandas as pd

# Predicted delays by group
group_labels = [
    "Historical, After 2021",     # Reference categories (Intercept only)
    "Recent, After 2021",         # Recent effect only
    "Historical, Before 2021",    # Period effect only
    "Recent, Before 2021"         # Recent + Period + Interaction
]
```

```python
# From your latest exponentiated results
intercept = 6.768912
recent_effect = 0.271201
period_effect = 0.492364
interaction_effect = 3.058085

predicted_delays = [
    intercept,                                                    # 6.77
    intercept * recent_effect,                                    # 6.77 * 0.271 = 1.84
    intercept * period_effect,                                    # 6.77 * 0.492 = 3.33
    intercept * recent_effect * period_effect * interaction_effect # 6.77 * 0.271 * 0.492 * 3.058 = 2.75
]

# Create DataFrame for plotting
df_plot = pd.DataFrame({
    "Group": group_labels,
    "Predicted Delay (Days)": predicted_delays
})

# Plot
plt.figure(figsize=(10, 6))
bars = plt.bar(df_plot["Group"], df_plot["Predicted Delay (Days)"], alpha=0.8)
plt.title("Predicted Report Delay by Spill Type and Period", fontsize=14)
plt.ylabel("Predicted Delay (Days)", fontsize=12)
plt.xticks(rotation=15, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Annotate bars
for bar in bars:
    height = bar.get_height()
    plt.annotate(f"{height:.2f}",
                 xy=(bar.get_x() + bar.get_width() / 2, height),
                 xytext=(0, 3),  # 3 points vertical offset
                 textcoords="offset points",
                 ha='center', va='bottom')
plt.show()

print("Predicted delays:")
for group, delay in zip(group_labels, predicted_delays):
    print(f"{group}: {delay:.2f} days")
```




```
Predicted delays:
Historical, After 2021: 6.77 days
Recent, After 2021: 1.84 days
Historical, Before 2021: 3.33 days
Recent, Before 2021: 2.76 days
```

---

5 Interrupted Time Series

[59]:
```
import pandas as pd
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt

# 1. Convert to monthly time series
spills_gdf['report_month'] = spills_gdf['Initial Report Date'].dt.to_period('M')
monthly = spills_gdf.groupby('report_month')['report_delay'].mean().
 ↪reset_index()
monthly['report_month'] = monthly['report_month'].dt.to_timestamp()

# 2. Create ITS variables
monthly['time'] = (monthly['report_month'] - monthly['report_month'].min()).dt.
 ↪days // 30
monthly['post_2021'] = (monthly['report_month'] >= pd.Timestamp('2021-01-01')).
 ↪astype(int)
```

```
monthly['time_post'] = monthly['time'] * monthly['post_2021']

# 3. Run the ITS model
its_model = smf.ols("report_delay ~ time + post_2021 + time_post",␣
 ↪data=monthly).fit()
print(its_model.summary())

# 4. Predict and plot
monthly['predicted'] = its_model.predict(monthly)

plt.figure(figsize=(12, 6))
plt.plot(monthly['report_month'], monthly['report_delay'], marker='o',␣
 ↪label='Observed', alpha=0.6)
plt.plot(monthly['report_month'], monthly['predicted'], linestyle='--',␣
 ↪color='red', label='Predicted (ITS)')
plt.axvline(x=pd.Timestamp('2021-01-01'), color='black', linestyle='--',␣
 ↪label='Policy Change (2021)')
plt.title('Interrupted Time Series: Monthly Report Delay')
plt.xlabel('Month')
plt.ylabel('Mean Report Delay (Days)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

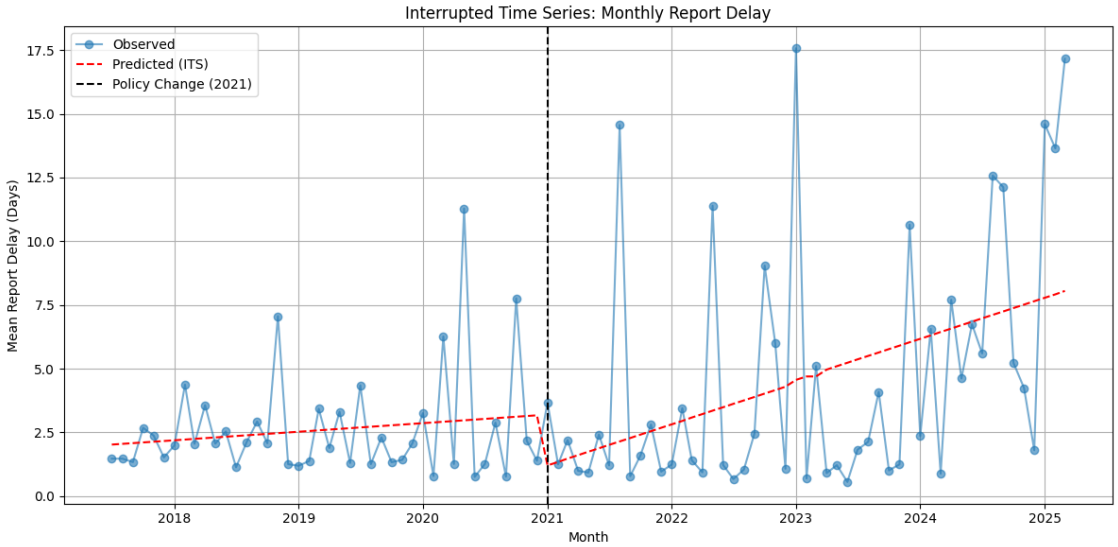
OLS Regression Results

Dep. Variable:	report_delay	R-squared:	0.216
Model:	OLS	Adj. R-squared:	0.189
Method:	Least Squares	F-statistic:	8.150
Date:	Wed, 06 Aug 2025	Prob (F-statistic):	7.46e-05
Time:	09:14:03	Log-Likelihood:	-248.17
No. Observations:	93	AIC:	504.3
Df Residuals:	89	BIC:	514.5
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.0214	1.081	1.870	0.065	-0.127	4.170
time	0.0280	0.045	0.617	0.539	-0.062	0.118
post_2021	-6.4546	2.524	-2.558	0.012	-11.469	-1.440
time_post	0.1062	0.056	1.893	0.062	-0.005	0.218

Omnibus:	35.237	Durbin-Watson:	2.065
Prob(Omnibus):	0.000	Jarque-Bera (JB):	67.022
Skew:	1.515	Prob(JB):	2.79e-15

```
Kurtosis:                       5.849   Cond. No.                         511.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```



5.1 ITS by Spill Type

[60]:
```
import pandas as pd
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt

# Group by month and spill type
spills_gdf['report_month'] = spills_gdf['Initial Report Date'].dt.to_period('M')
monthly_by_type = spills_gdf.groupby(['report_month',␣
 ↪'spill_type'])['report_delay'].mean().reset_index()
monthly_by_type['report_month'] = monthly_by_type['report_month'].dt.
 ↪to_timestamp()

# ITS variables
monthly_by_type['time'] = (monthly_by_type['report_month'] -␣
 ↪monthly_by_type['report_month'].min()).dt.days // 30
monthly_by_type['post_2021'] = (monthly_by_type['report_month'] >= pd.
 ↪Timestamp('2021-01-01')).astype(int)
monthly_by_type['time_post'] = monthly_by_type['time'] *␣
 ↪monthly_by_type['post_2021']
```

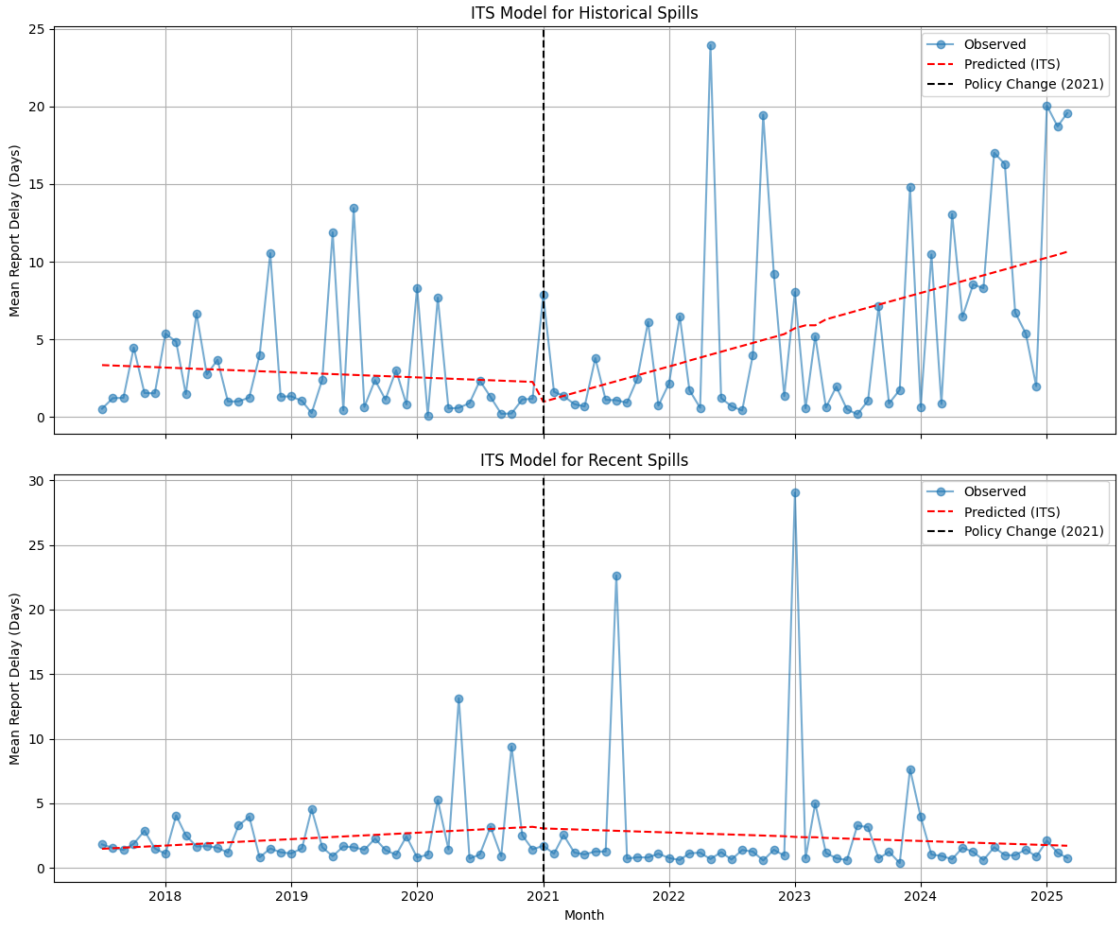
```python
# Run ITS models and print summaries
its_results_by_type = {}
for stype in monthly_by_type['spill_type'].unique():
    df = monthly_by_type[monthly_by_type['spill_type'] == stype].copy()
    model = smf.ols("report_delay ~ time + post_2021 + time_post", data=df).
 ↪fit()
    df['predicted'] = model.predict(df)
    its_results_by_type[stype] = (model, df)

    print(f"\n=== ITS Model Summary for {stype} Spills ===\n")
    print(model.summary())
```

```
=== ITS Model Summary for Historical Spills ===

                            OLS Regression Results
==============================================================================
Dep. Variable:           report_delay   R-squared:                       0.229
Model:                            OLS   Adj. R-squared:                  0.203
Method:                 Least Squares   F-statistic:                     8.814
Date:                Wed, 06 Aug 2025   Prob (F-statistic):           3.54e-05
Time:                        09:14:17   Log-Likelihood:                -277.64
No. Observations:                  93   AIC:                             563.3
Df Residuals:                      89   BIC:                             573.4
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      3.3388      1.484      2.249      0.027       0.389       6.288
time          -0.0264      0.062     -0.423      0.673      -0.150       0.097
post_2021    -10.3194      3.465     -2.978      0.004     -17.204      -3.435
time_post      0.2159      0.077      2.803      0.006       0.063       0.369
==============================================================================
Omnibus:                       32.115   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               59.323
Skew:                           1.383   Prob(JB):                     1.31e-13
Kurtosis:                       5.767   Cond. No.                         511.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.

=== ITS Model Summary for Recent Spills ===

                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:          report_delay   R-squared:                       0.013
Model:                         OLS   Adj. R-squared:                 -0.020
Method:              Least Squares   F-statistic:                     0.3852
Date:             Wed, 06 Aug 2025   Prob (F-statistic):              0.764
Time:                     09:14:17   Log-Likelihood:                -259.51
No. Observations:               93   AIC:                             527.0
Df Residuals:                   89   BIC:                             537.2
Df Model:                        3
Covariance Type:         nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.4880      1.221      1.218      0.226      -0.939       3.915
time           0.0414      0.051      0.808      0.421      -0.060       0.143
post_2021      2.6802      2.851      0.940      0.350      -2.985       8.345
time_post     -0.0677      0.063     -1.068      0.288      -0.194       0.058
==============================================================================
Omnibus:                      129.874   Durbin-Watson:                   2.198
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3176.252
Skew:                           4.916   Prob(JB):                         0.00
Kurtosis:                      29.889   Cond. No.                         511.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[61]:
```python
import pandas as pd
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt

# 1. Group by month and spill type
spills_gdf['report_month'] = spills_gdf['Initial Report Date'].dt.to_period('M')
monthly_by_type = spills_gdf.groupby(['report_month',␣
 ↪'spill_type'])['report_delay'].mean().reset_index()
monthly_by_type['report_month'] = monthly_by_type['report_month'].dt.
 ↪to_timestamp()

# 2. Create ITS variables
monthly_by_type['time'] = (monthly_by_type['report_month'] -␣
 ↪monthly_by_type['report_month'].min()).dt.days // 30
monthly_by_type['post_2021'] = (monthly_by_type['report_month'] >= pd.
 ↪Timestamp('2021-01-01')).astype(int)
monthly_by_type['time_post'] = monthly_by_type['time'] *␣
 ↪monthly_by_type['post_2021']

# 3. Run ITS models separately by spill type
```

```
its_results_by_type = {}
for stype in monthly_by_type['spill_type'].unique():
    df = monthly_by_type[monthly_by_type['spill_type'] == stype].copy()
    model = smf.ols("report_delay ~ time + post_2021 + time_post", data=df).
 ↪fit()
    df['predicted'] = model.predict(df)
    its_results_by_type[stype] = (model, df)

# 4. Plot side-by-side
fig, axs = plt.subplots(2, 1, figsize=(12, 10), sharex=True)

for ax, (stype, (model, df)) in zip(axs, its_results_by_type.items()):
    ax.plot(df['report_month'], df['report_delay'], marker='o',␣
 ↪label='Observed', alpha=0.6)
    ax.plot(df['report_month'], df['predicted'], linestyle='--', color='red',␣
 ↪label='Predicted (ITS)')
    ax.axvline(x=pd.Timestamp('2021-01-01'), color='black', linestyle='--',␣
 ↪label='Policy Change (2021)')
    ax.set_title(f"ITS Model for {stype} Spills")
    ax.set_ylabel("Mean Report Delay (Days)")
    ax.legend()
    ax.grid(True)

axs[1].set_xlabel("Month")
plt.tight_layout()
plt.show()
```



---

6 Conclusion: Comparative Summary

[62]:
```python
def extract_regression_results(model, model_name):
    coefs = model.params
    ses = model.bse
    pvals = model.pvalues

    df = pd.DataFrame({
        'Coefficient': coefs,
        'Std. Error': ses,
        'P-Value': pvals
    })
    df['Model'] = model_name
    return df.reset_index().rename(columns={'index': 'Term'})
```

[63]:
```
# Collect regression summaries
results_frames = []

# Add your models - use your actual variable names
results_frames.append(extract_regression_results(model, 'OLS (Interaction)'))
results_frames.append(extract_regression_results(nb_model, 'Negative Binomial'))
results_frames.append(extract_regression_results(its_model, 'ITS (Monthly, All␣
 ↪Spills)'))


# ITS by spill type (e.g., 'Historical', 'Recent')
for spill_type, (model, df) in its_results_by_type.items():
    label = f"ITS (Monthly, {spill_type})"
    results_frames.append(extract_regression_results(model, label))

# Combine all
regression_summary_table = pd.concat(results_frames, ignore_index=True)

# Pivot just the coefficients for display
regression_coef_table = regression_summary_table.pivot_table(
    index='Term',
    columns='Model',
    values='Coefficient',
    aggfunc='first'
).round(3)

regression_coef_table
```

[63]:
```
Model                                              ITS (Monthly, All Spills)  \
Term
C(Period)[T.Before 2021]                                                 NaN
C(spill_type)[T.Recent]                                                  NaN
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]                         NaN
Intercept                                                              2.021
post_2021                                                             -6.455
time                                                                   0.028
time_post                                                              0.106

Model                                              ITS (Monthly, Historical)  \
Term
C(Period)[T.Before 2021]                                                 NaN
C(spill_type)[T.Recent]                                                  NaN
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]                         NaN
Intercept                                                              3.339
post_2021                                                            -10.319
time                                                                  -0.026
time_post                                                              0.216
```

Model	ITS (Monthly, Recent)
Term	
C(Period)[T.Before 2021]	NaN
C(spill_type)[T.Recent]	NaN
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]	NaN
Intercept	1.488
post_2021	2.680
time	0.041
time_post	-0.068

Model	Negative Binomial
Term	
C(Period)[T.Before 2021]	-0.709
C(spill_type)[T.Recent]	-1.305
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]	1.118
Intercept	1.912
post_2021	NaN
time	NaN
time_post	NaN

Model	OLS (Interaction)
Term	
C(Period)[T.Before 2021]	NaN
C(spill_type)[T.Recent]	NaN
C(spill_type)[T.Recent]:C(Period)[T.Before 2021]	NaN
Intercept	1.488
post_2021	2.680
time	0.041
time_post	-0.068