pp290_ctc

December 7, 2022

[ ]:
```
import pandas as pd
import numpy as np
import os
from data_utils import build_pulse, filter_pulse, basic_processing
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
from statsmodels.iolib.smpickle import load_pickle
```

[ ]:
```
# Periods: https://www.census.gov/programs-surveys/household-pulse-survey/
 ↪datasets.html
week_start = 18
week_end = 45
```

[ ]:
```
file_path = f"processed_data/{week_start}_{week_end}_pulse.csv"
if os.path.exists(file_path):
    df = pd.read_csv(file_path)
else:
    df = build_pulse(week_start, week_end)
    df = filter_pulse(df, verbose=True)
    df = basic_processing(df)
    df.to_csv(file_path, index=False)
```

```
Opening local file for pulse week18
Opening local file for pulse week19
Opening local file for pulse week20
Opening local file for pulse week21
Opening local file for pulse week22
Opening local file for pulse week23
Opening local file for pulse week24
Opening local file for pulse week25
Opening local file for pulse week26
Opening local file for pulse week27
Opening local file for pulse week28
Opening local file for pulse week29
Opening local file for pulse week30
Opening local file for pulse week31
```

```
Opening local file for pulse week32
Opening local file for pulse week33
Opening local file for pulse week34
Opening local file for pulse week35
Opening local file for pulse week36
Opening local file for pulse week37
Opening local file for pulse week38
Opening local file for pulse week39
Opening local file for pulse week40
Opening local file for pulse week41
Opening local file for pulse week42
Opening local file for pulse week43
Opening local file for pulse week44
Opening local file for pulse week45
Shape before: (1961696, 423)
Shape after: (1492226, 423)
```

[ ]:
```
df.pivot_table(columns="Eligible", index="Week", values='Depressed_or_Anxious')
```

[ ]:
```
Eligible         0         1
Week
18        0.359125  0.434565
19        0.358977  0.457224
20        0.335233  0.440929
21        0.341051  0.444727
22        0.347673  0.416660
23        0.329956  0.408707
24        0.303432  0.390223
25        0.288603  0.391735
26        0.275443  0.368456
27        0.260051  0.342203
28        0.233116  0.325137
29        0.225814  0.309905
30        0.219393  0.313104
31        0.209030  0.306195
32        0.215463  0.316479
33        0.205963  0.303399
34        0.215549  0.329154
35        0.226197  0.331866
36        0.240367  0.358593
37        0.241455  0.351434
38        0.233655  0.355143
39        0.227371  0.341084
40        0.233776  0.339217
41        0.247849  0.346669
42        0.243051  0.346814
43        0.233548  0.331908
```

```
44          0.220890  0.317901
45          0.212115  0.327428
```

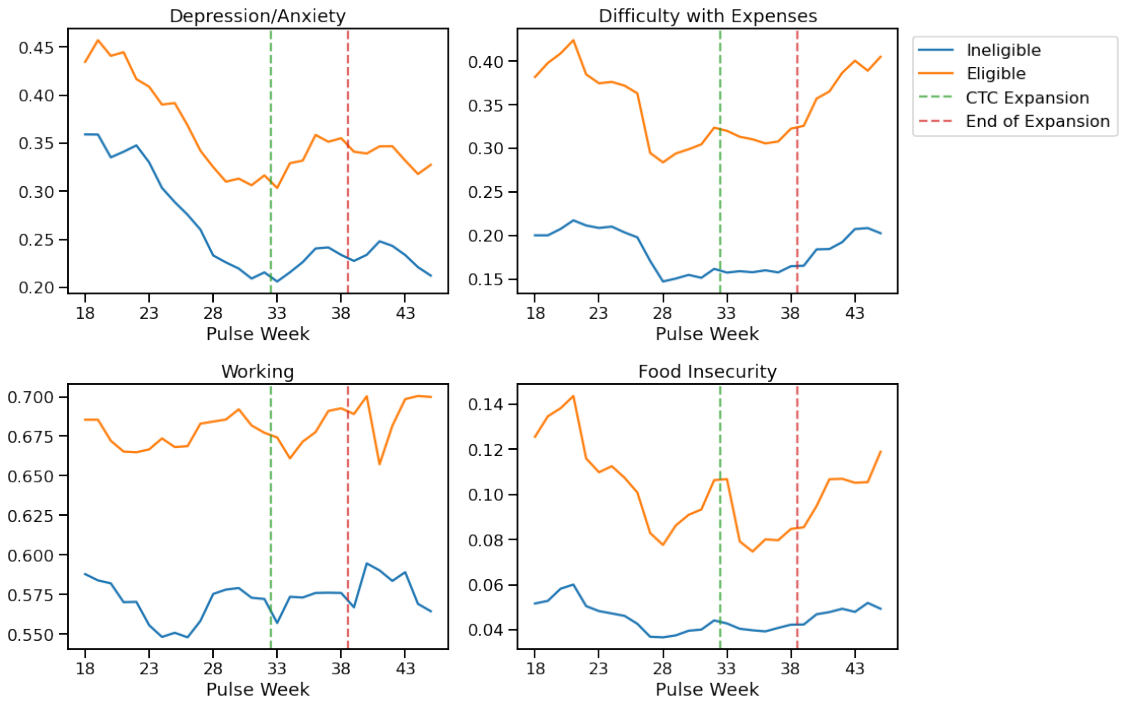
[ ]:
```
df[(df["Eligible"] == 1) & (df['Post'] == 1)]["Received_CTC"].
 ↪value_counts(normalize=True)
```

[ ]:
```
1    0.661219
0    0.338781
Name: Received_CTC, dtype: float64
```

1) Less than $25,000
2) $25,000 - $34,999
3) $35,000 - $49,999
4) $50,000 - $74,999
5) $75,000 - $99,999
6) $100,000 - $149,999
7) $150,000 - $199,999
8) $200,000 and above

[ ]:
```
df_m = df.query("Income < 7")
```

[ ]:
```
sns.set_context("talk")
fig, ax = plt.subplots(2, 2, figsize=(16, 10))
dependent_vars = ['Depressed_or_Anxious', 'Difficulty_with_Expenses',␣
 ↪'Worked_in_last_week', 'Food_Insecurity']
titles = ['Depression/Anxiety', 'Difficulty with Expenses', 'Working', 'Food␣
 ↪Insecurity']
counter = 0
for i in range(2):
    for j in range(2):
        df_m.pivot_table(columns="Eligible", index="Week",␣
 ↪values=dependent_vars[counter]).plot(ax=ax[i, j], legend=False)
        ax[i, j].axvline(14.5, c='tab:green', linestyle="--", alpha=.7)
        ax[i, j].axvline(20.5, c='tab:red', linestyle="--", alpha=.7)
        ax[i, j].set_title(titles[counter])
        ax[i, j].set_xlabel("Pulse Week")
        counter += 1
ax[0, 1].legend(["Ineligible", "Eligible", "CTC Expansion", "End of␣
 ↪Expansion"], loc="upper right", bbox_to_anchor=(1.6, 1))
fig.tight_layout();
fig.savefig("figures/trendlines.png", dpi=300, facecolor=fig.get_facecolor())
```



[ ]:
```
df_m.columns
```

[ ]:
```
Index(['Received_CTC', 'Eligible', 'Post', 'Week', 'Treat', 'Number_of_kids',
       'Depressed_or_Anxious', 'Depressed', 'Anxious',
       'Difficulty_with_Expenses', 'Food_Insecurity', 'Rent_Confidence',
       'Vaccinated', 'Enrolled_in_SNAP', 'Worked_in_last_week', 'Income',
       'Education', 'Age', 'Received_EBT_card', 'Received_Free_Food',
       'Household_Weight'],
      dtype='object')
```

[ ]:
```
dependent = 'Food_Insecurity'
control_list = ["Enrolled_in_SNAP", "Received_Free_Food", "Received_EBT_card"]
controls = " + ".join(control_list)
print("Controls:", controls)
```

```
Controls: Enrolled_in_SNAP + Received_Free_Food + Received_EBT_card
```

[ ]:
```
dids = []
for var in dependent_vars:
    formula = f"{var} ~  Eligible + Post + {controls} + Treat"
    basic_did = (
        sm.WLS.from_formula(
            formula,
            data=df_m,
            weights=df_m['Household_Weight']
```

```
        ).fit(cov_type="HC1"))
    basic_did.save(f"models/{var}_did_model.p")
    dids.append(basic_did)
```

[ ]:
```
summary_col([*dids], stars=True)
```

[ ]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""

==========================================================================
====================
                   Depressed_or_Anxious Difficulty_with_Expenses
Worked_in_last_week Food_Insecurity
--------------------------------------------------------------------------
--------------------
Intercept          0.2830***            0.2154***                0.5820***
0.0595***
                   (0.0009)             (0.0009)                 (0.0010)
(0.0006)
Eligible           0.0509***            0.1277***                0.1203***
0.0513***
                   (0.0020)             (0.0020)                 (0.0020)
(0.0016)
Post               -0.0342***           -0.0319***               0.0020
-0.0079***
                   (0.0020)             (0.0019)                 (0.0021)
(0.0014)
Enrolled_in_SNAP   0.1577***            0.3036***                -0.2767***
0.1356***
                   (0.0031)             (0.0031)                 (0.0029)
(0.0028)
Received_Free_Food 0.1055***            0.2070***                -0.0880***
0.1291***
                   (0.0037)             (0.0037)                 (0.0036)
(0.0034)
Received_EBT_card  -0.0378***           0.0304***                0.0225***
-0.0176***
                   (0.0061)             (0.0061)                 (0.0062)
(0.0053)
Treat              0.0195***            -0.0181***               -0.0069
-0.0287***
                   (0.0043)             (0.0043)                 (0.0045)
(0.0032)
R-squared          0.0232               0.1019                   0.0392
0.0522
R-squared Adj.     0.0232               0.1019                   0.0392
0.0522
```

```
==========================================================================
====================
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
"""
```

[ ]:
```
fe_dids = []
for var in dependent_vars:
    formula = f"{var} ~  Eligible + C(Week) + {controls} + Treat"
    model_path = f"models/{var}_fe_models.p"
    if os.path.exists(model_path):
        print("Loading", var, "locally")
        fixed_effects_did = load_pickle(model_path)
    fixed_effects_did = (
        sm.WLS.from_formula(
            formula,
            data=df_m,
            weights=df_m['Household_Weight']
            ).fit(cov_type="HC1"))
    fixed_effects_did.save(model_path)
    fe_dids.append(fixed_effects_did)
```

0.1 Fixed Effects Results:

[ ]:
```
summary_col([*fe_dids], stars=True, regressor_order=["Treat"],␣
 ↪drop_omitted=True)
```

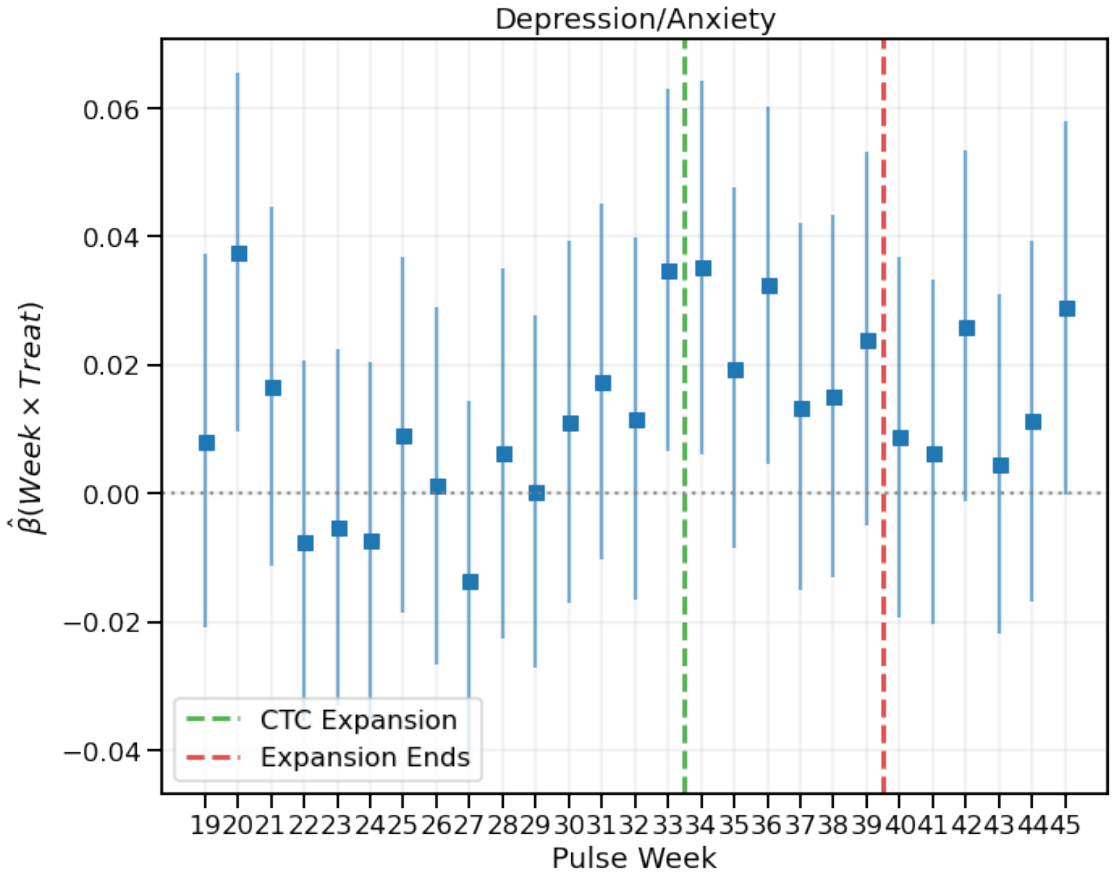
[ ]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""

==========================================================================
================
               Depressed_or_Anxious Difficulty_with_Expenses Worked_in_last_week
Food_Insecurity
--------------------------------------------------------------------------
----------------
Treat          0.0170***            -0.0183***               -0.0063
-0.0290***
               (0.0043)             (0.0043)                 (0.0045)
(0.0032)
R-squared      0.0309               0.1059                   0.0397
0.0534
R-squared Adj. 0.0309               0.1058                   0.0397
0.0534
==========================================================================
================
Standard errors in parentheses.
```

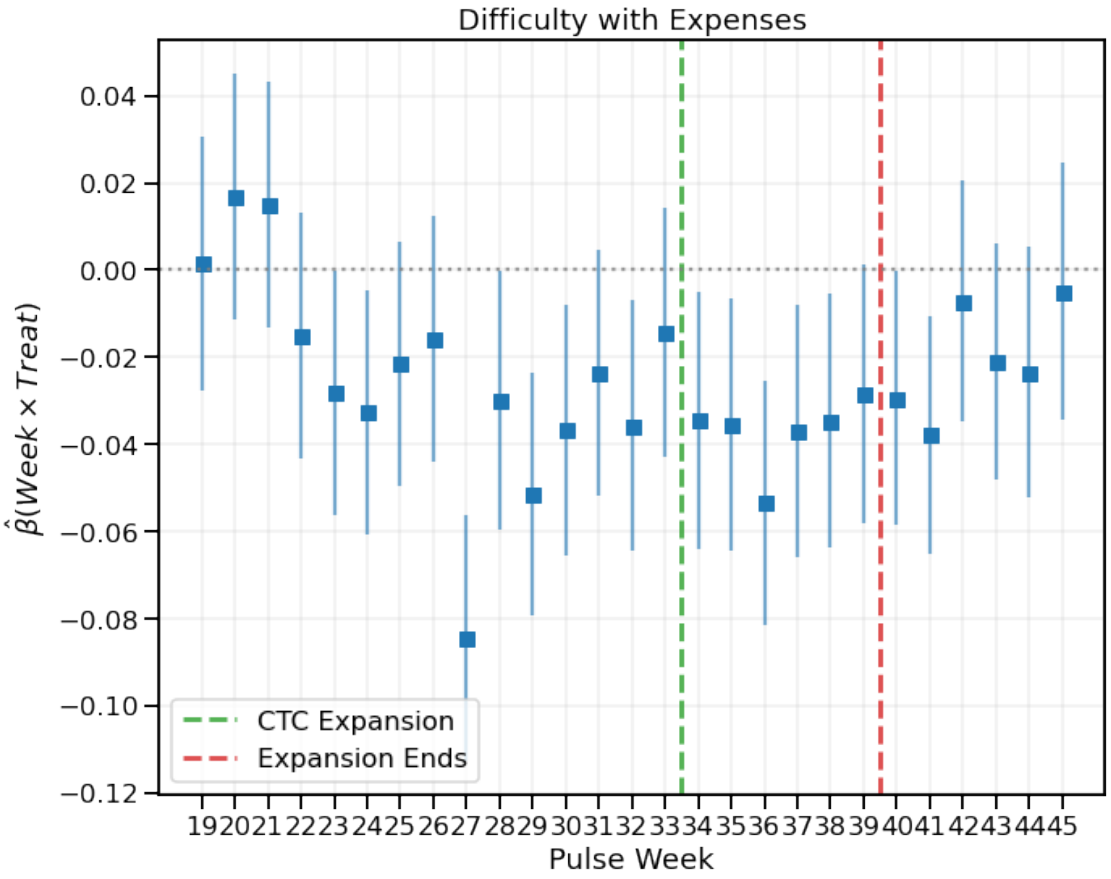
```
* p<.1, ** p<.05, ***p<.01
"""
```

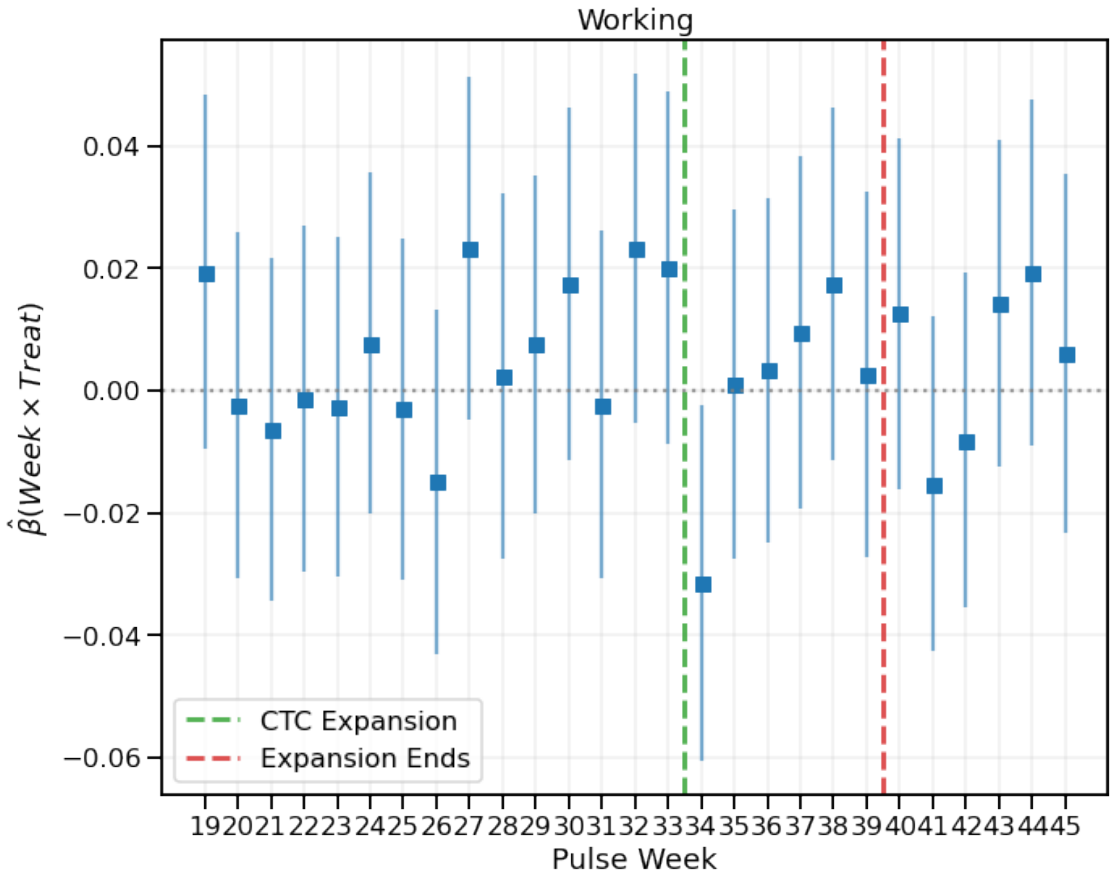
[ ]:
```
event_studies = []
for var in dependent_vars:
    formula = f"{var} ~ C(Week)*Eligible"
    event_study = (
        sm.WLS.from_formula(
            formula,
            data=df_m,
            weights=df_m['Household_Weight']
            ).fit(cov_type='HC1'))
    event_study.save(f"models/{var}_event_study.p")
    event_studies.append(event_study)
```

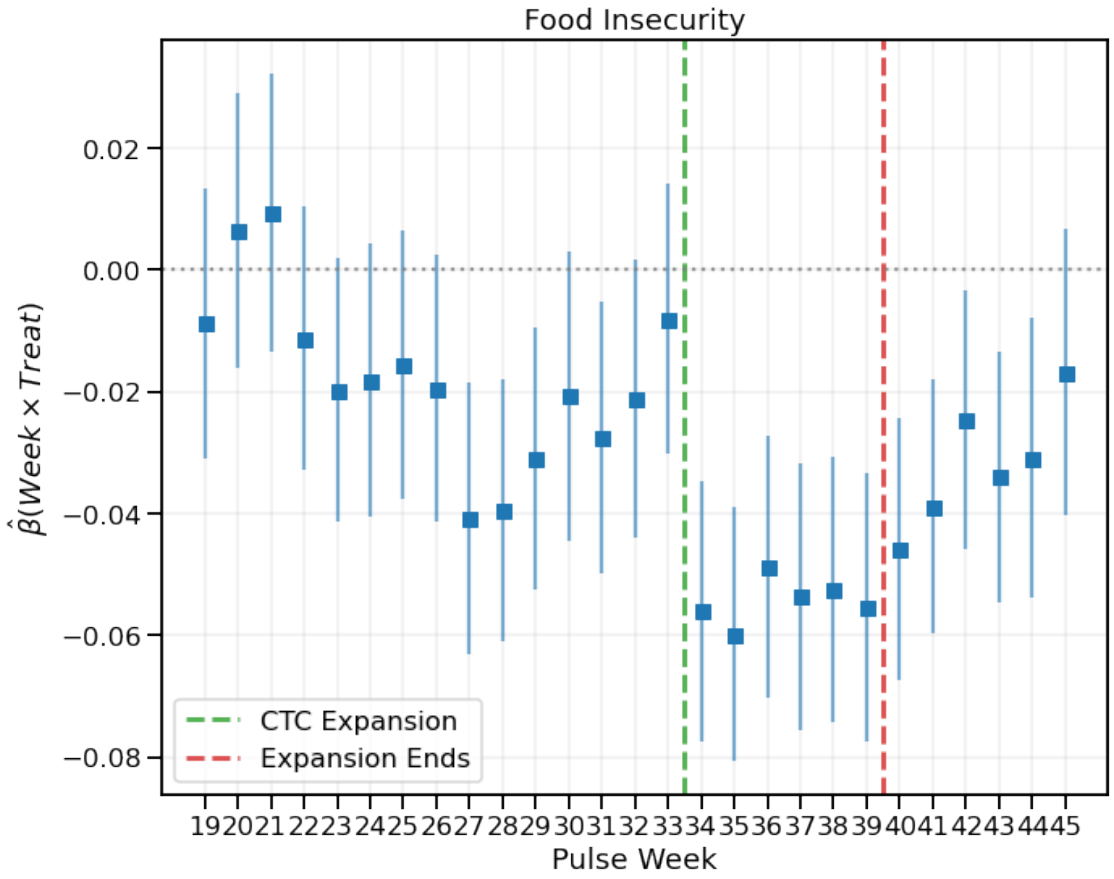
[ ]:
```
counter=0
for event_study in event_studies:
    estimates = pd.DataFrame({"coef": event_study.params, "std_err":event_study.
 ↪HC1_se, "low": event_study.conf_int()[0], "high": event_study.conf_int()[1]})
    estimates = estimates.reset_index().rename({"index": "name"}, axis=1)
    did_estimates = estimates[estimates['name'].str.startswith('C(Week)[T.') &␣
 ↪estimates['name'].str.endswith('Eligible')]
    did_estimates = did_estimates.reset_index(drop=True)
    did_estimates['Week'] = range(df_m['Week'].astype(int).min()+1,␣
 ↪df_m['Week'].astype(int).max()+1)

    fig, ax = plt.subplots(1, 1, figsize=(10, 8))
    ax.axvline(14.5, c="tab:green", alpha=.8, linestyle="--", linewidth=3,␣
 ↪label="CTC Expansion")
    ax.axvline(20.5, c="tab:red", alpha=.8, linestyle="--", linewidth=3,␣
 ↪label="Expansion Ends")
    ax.errorbar(did_estimates.index, did_estimates['coef'],␣
 ↪yerr=did_estimates['high']-did_estimates['coef'], alpha=.6, c="tab:blue",␣
 ↪fmt='none')
    ax.plot(did_estimates['coef'], "s")
    # ax.plot(did_estimates['low'], "s", c="tab:blue")
    # ax.plot(did_estimates['high'], "s", c="tab:blue")
    ax.axhline(0, c="grey", alpha=.8, linestyle=":")
    ax.grid(alpha=.2)
    ax.legend(loc="lower left")

    ax.set_xlabel("Pulse Week")
    ax.set_ylabel(r"$\hat{\beta} (Week \times Treat)$")
    ax.set_title(titles[counter])
    counter+=1

    ax.set_xticks(did_estimates.index, did_estimates['Week'])
```

```
fig.tight_layout()
```




Difficulty with Expenses
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
$\hat{\beta}$(Week × Treat)
CTC Expansion
Expansion Ends
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
Pulse Week


Working
0.04
0.02
0.00
−0.02
−0.04
−0.06
$\hat{\beta}$(Week × Treat)
CTC Expansion
Expansion Ends
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
Pulse Week



[ ]:
```
# estimates = pd.DataFrame({"coef": event_study.params, "std_err":event_study.
 ↪HC1_se, "low": event_study.conf_int()[0], "high": event_study.conf_int()[1]})
# estimates = estimates.reset_index().rename({"index": "name"}, axis=1)
# did_estimates = estimates[estimates['name'].str.startswith('C(WEEK)[T.') &␣
 ↪estimates['name'].str.endswith('CTC_Eligible')]
# did_estimates = did_estimates.reset_index(drop=True)
# did_estimates['week'] = range(df['WEEK'].min()+1, df['WEEK'].max()+1)
```

[ ]:
```
# fig, ax = plt.subplots(1, 1, figsize=(10, 8))
# ax.axvline(14.5, c="tab:green", alpha=.8, linestyle="--", linewidth=3,␣
 ↪label="CTC Expansion")
# ax.axvline(20.5, c="tab:red", alpha=.8, linestyle="--", linewidth=3,␣
 ↪label="Expansion Ends")
# ax.errorbar(did_estimates.index, did_estimates['coef'],␣
 ↪yerr=did_estimates['high']-did_estimates['coef'], alpha=.6, c="tab:blue",␣
 ↪fmt='none')
# ax.plot(did_estimates['coef'], "s")
# # ax.plot(did_estimates['low'], "s", c="tab:blue")
# # ax.plot(did_estimates['high'], "s", c="tab:blue")
```

```
# ax.axhline(0, c="grey", alpha=.8, linestyle=":")
# ax.grid(alpha=.2)
# ax.legend(loc="lower left")

# ax.set_xlabel("Pulse Week")
# ax.set_ylabel(r"$\hat{\beta} (Week \times Treat)$")
# ax.set_title(f"{dependent}")

# ax.set_xticks(did_estimates.index, did_estimates['week'])

# fig.tight_layout()
```