

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
insurance_data = pd.read_csv("/Users/avanthpakanati/Desktop/ECON:HLTH Research Seminar /Home
expansion_data = pd.read_csv("/Users/avanthpakanati/Desktop/ECON:HLTH Research Seminar /Home
medicaid_data = pd.read_csv("/Users/avanthpakanati/Desktop/ECON:HLTH Research Seminar /Home

```

```

#Question 1

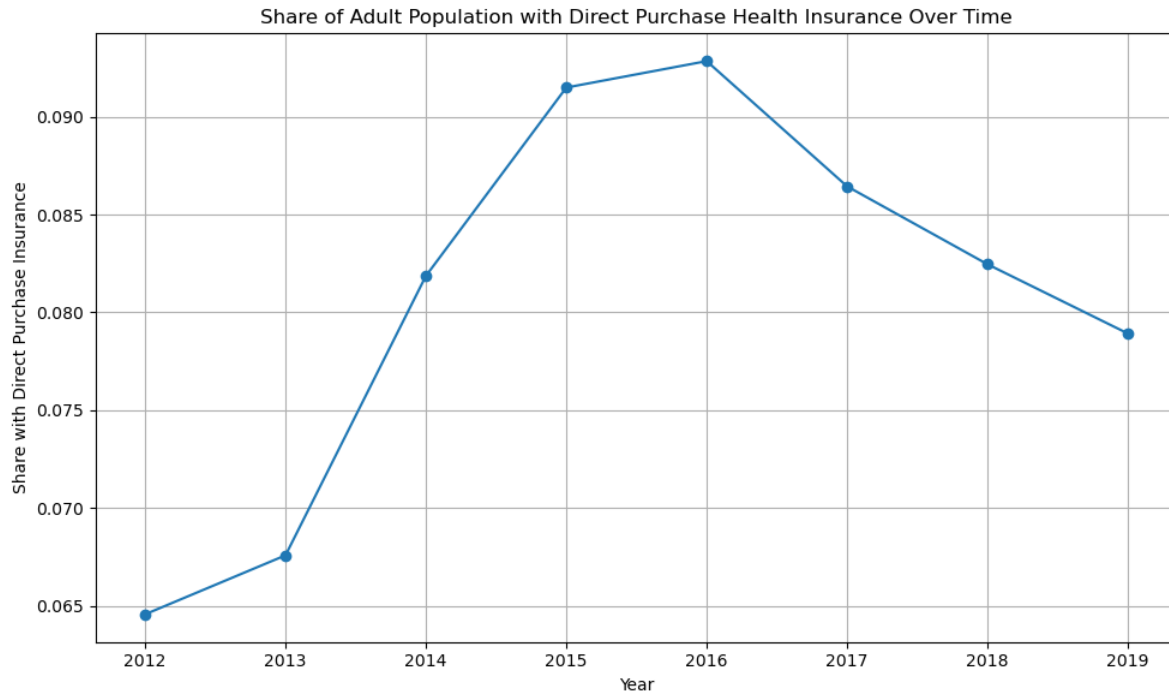
# Calculate the share of adults with direct purchase insurance
insurance_data['direct_share'] = insurance_data['ins_direct'] / insurance_data['adult_pop']

# Group by year and sum across all states
national_share = insurance_data.groupby('year')[['ins_direct', 'adult_pop']].sum()

# Calculate national share of direct purchase insurance
national_share['direct_share'] = national_share['ins_direct'] / national_share['adult_pop']

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(national_share.index, national_share['direct_share'], marker='o')
plt.title('Share of Adult Population with Direct Purchase Health Insurance Over Time')
plt.xlabel('Year')
plt.ylabel('Share with Direct Purchase Insurance')
plt.grid(True)
plt.tight_layout()
plt.show()

```



#Question 2 Originally went up due to ACA, but several policy changes caused it to decline

#Question 3

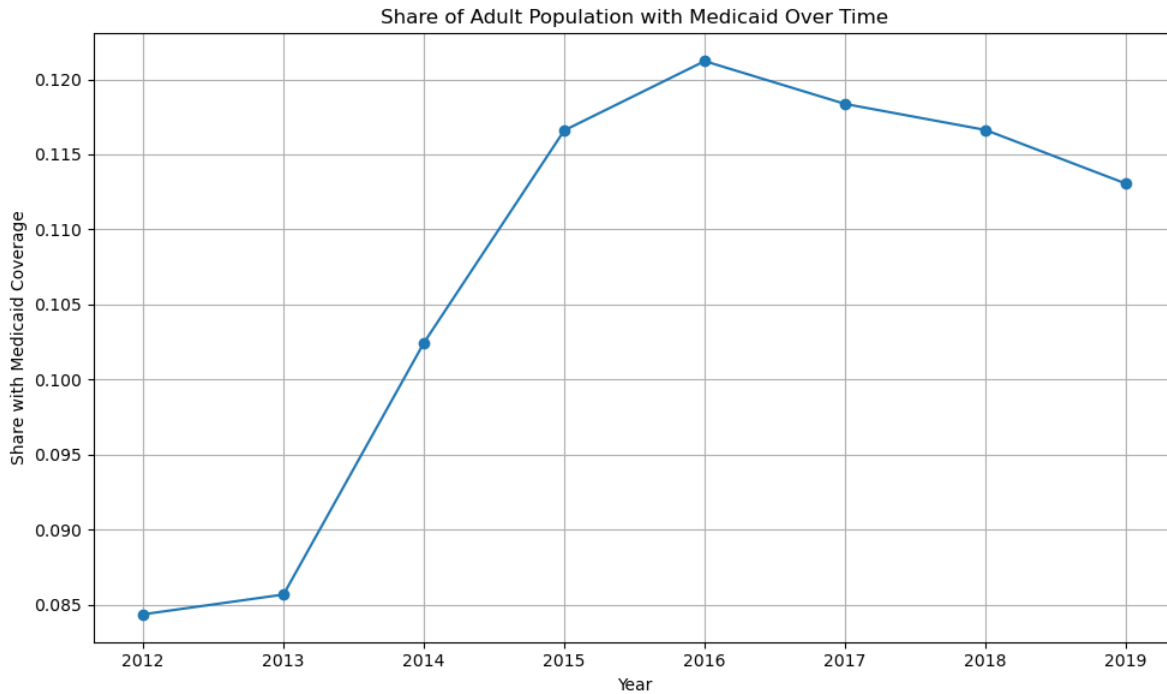
```
insurance_data['medicaid_share'] = insurance_data['ins_medicaid'] / insurance_data['adult_pop']
medicaid_national_share = insurance_data.groupby('year')[['ins_medicaid', 'adult_pop']].sum()

medicaid_national_share['medicaid_share'] = medicaid_national_share['ins_medicaid'] / medicaid_national_share['adult_pop']

medicaid_national_share = insurance_data.groupby('year')[['ins_medicaid', 'adult_pop']].sum()

medicaid_national_share['medicaid_share'] = medicaid_national_share['ins_medicaid'] / medicaid_national_share['adult_pop']

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(medicaid_national_share.index, medicaid_national_share['medicaid_share'], marker='o')
plt.title('Share of Adult Population with Medicaid Over Time')
plt.xlabel('Year')
plt.ylabel('Share with Medicaid Coverage')
plt.grid(True)
plt.tight_layout()
plt.show()
```



#Question 4

```
expansion_data['date_adopted'] = pd.to_datetime(expansion_data['date_adopted'], errors='coerce')
expansion_data['expand_year'] = expansion_data['date_adopted'].dt.year
expansion_data_clean = expansion_data[expansion_data['expand_year'] <= 2014]
```

```
expansion_data_clean['expanded_2014'] = expansion_data_clean['expand_year'] <= 2014
```

```
merged_data = pd.merge(insurance_data, expansion_data_clean[['State', 'expanded_2014']], on='State')
```

```
merged_data['uninsured_share'] = merged_data['uninsured'] / merged_data['adult_pop']
```

Group by year and expansion status, then aggregate

```
grouped = merged_data.groupby(['year', 'expanded_2014'])[['uninsured', 'adult_pop']].sum()
```

```
grouped['uninsured_share'] = grouped['uninsured'] / grouped['adult_pop']
```

```
grouped = grouped.reset_index()
```

Plotting

```
plt.figure(figsize=(10, 6))
```

```
for label, df in grouped.groupby('expanded_2014'):
```

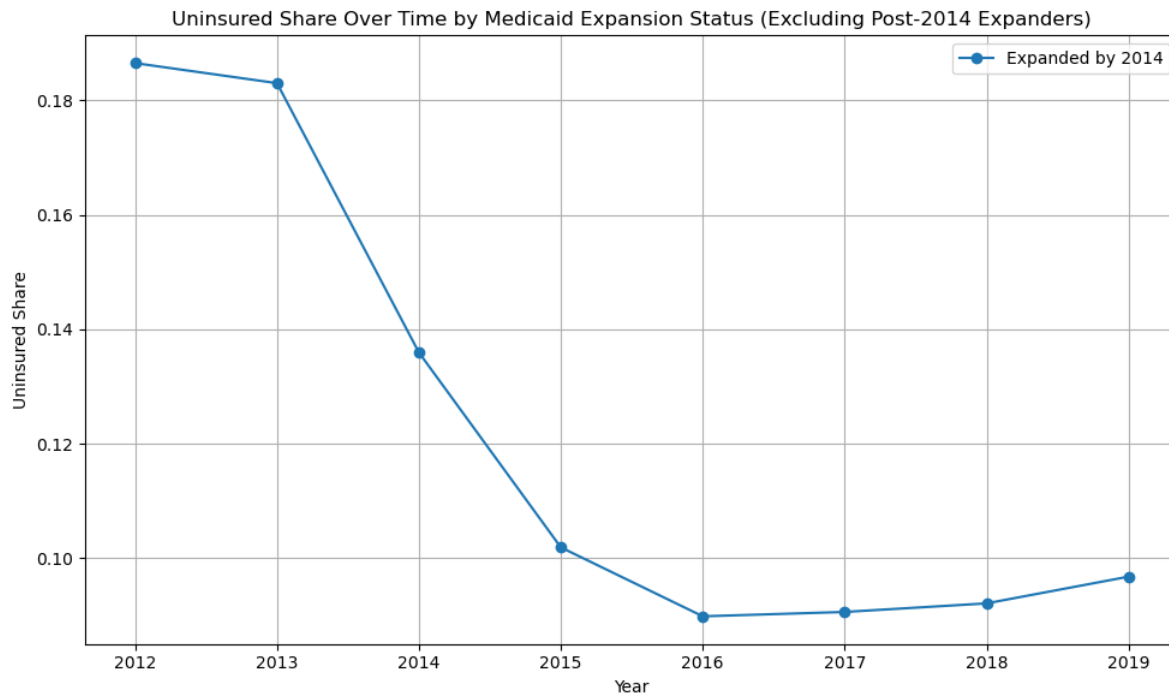
```
    label_str = 'Expanded by 2014' if label else 'Not Expanded by 2014'
```

```
    plt.plot(df['year'], df['uninsured_share'], marker='o', label=label_str)
```

```
plt.title('Uninsured Share Over Time by Medicaid Expansion Status (Excluding Post-2014 Expansions)')
plt.xlabel('Year')
plt.ylabel('Uninsured Share')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

/var/folders/2q/wzjp_2kd355b8clhzqwmtyb40000gn/T/ipykernel_44777/1013151488.py:6: 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/10min.html#boolean-indexing
expansion_data_clean['expanded_2014'] = expansion_data_clean['expand_year'] <= 2014



#Question 5

```
#filter for years 2012 and 2015
filtered_medicaid = medicaid_data[medicaid_data['year'].isin([2012, 2015])]
```

```

expansion = filtered_medicaid.loc[filtered_medicaid['expand_year'] <= 2014, 'State'].unique()
non_expansion = filtered_medicaid.loc[filtered_medicaid['expand_ever'] == False, 'State'].unique()

final_data = filtered_medicaid[filtered_medicaid['State'].isin(list(expansion) + list(non_expansion))]

# Calculate uninsured share
final_data['uninsured_share'] = final_data['uninsured'] / final_data['adult_pop']

grouped = final_data.groupby(['year', 'expand_ever'])['uninsured_share'].mean().reset_index()

# Pivot to create the 2x2 DiD table
dd_table = grouped.pivot(index='expand_ever', columns='year', values='uninsured_share')
dd_table.index = ['Non-Expansion States', 'Expansion States']

# Display the table
print(dd_table)

```

year	2012	2015
Non-Expansion States	0.215653	0.158208
Expansion States	0.168052	0.094090

/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/2992481859.py:15: 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
final_data['uninsured_share'] = final_data['uninsured'] / final_data['adult_pop']

```

#Question 6
final_data['D'] = final_data['expand_ever'].astype(int)
final_data['Post'] = (final_data['year'] == 2015).astype(int)

# Run Difference-in-Differences regression
model = smf.ols('uninsured_share ~ D + Post + D:Post', data=final_data).fit()
model_summary = model.summary()

model_summary

```

```
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1486860411.py:2: 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/1d.html#boolean-indexing
final_data['D'] = final_data['expand Ever'].astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1486860411.py:3: 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/1d.html#boolean-indexing
final_data['Post'] = (final_data['year'] == 2015).astype(int)
```

Table 1: OLS Regression Results

Dep. Variable:	uninsured_share	R-squared:	0.487
Model:	OLS	Adj. R-squared:	0.468
Method:	Least Squares	F-statistic:	26.55
Date:	Wed, 23 Apr 2025	Prob (F-statistic):	3.55e-12
Time:	09:39:14	Log-Likelihood:	148.55
No. Observations:	88	AIC:	-289.1
Df Residuals:	84	BIC:	-279.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.2157	0.011	19.420	0.000	0.194	0.238
D	-0.0476	0.014	-3.358	0.001	-0.076	-0.019
Post	-0.0574	0.016	-3.658	0.000	-0.089	-0.026
D:Post	-0.0165	0.020	-0.824	0.412	-0.056	0.023

Omnibus:	0.440	Durbin-Watson:	1.970
Prob(Omnibus):	0.803	Jarque-Bera (JB):	0.265
Skew:	0.134	Prob(JB):	0.876
Kurtosis:	3.017	Cond. No.	7.80

#Question 7

```
final_data['D'] = final_data['expand Ever'].astype(int)
```

```

final_data['Post'] = (final_data['year'] == 2015).astype(int)

# Calculate uninsured share
final_data['uninsured_share'] = final_data['uninsured'] / final_df['adult_pop']

final_df = final_data.set_index(['State', 'year'])

# Run the DiD regression with state and year fixed effects
model = PanelOLS.from_formula('uninsured_share ~ D*Post + EntityEffects + TimeEffects', data=

```

/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1903189088.py:3: 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
 final_data['D'] = final_data['expand_ever'].astype(int)
 /var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1903189088.py:4: 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
 final_data['Post'] = (final_data['year'] == 2015).astype(int)

NameError: name 'final_df' is not defined

```

-----
NameError                                Traceback (most recent call last)
Input In [8], in <cell line: 7>()
      4 final_data['Post'] = (final_data['year'] == 2015).astype(int)
      6 # Calculate uninsured share
----> 7 final_data['uninsured_share'] = final_data['uninsured'] / final_df['adult_pop']
      9 final_df = final_data.set_index(['State', 'year'])
     11 # Run the DiD regression with state and year fixed effects
NameError: name 'final_df' is not defined

```

```

#Question 8
# Filter for the years 2012 and 2015 (including all states)
filtered_df = medicaid_data[medicaid_data['year'].isin([2012, 2015])]

# Create treatment indicator (D) for any state that ever expanded Medicaid
filtered_df['D'] = filtered_df['expand_ever'].astype(int)

# Create post-period indicator (Post) for 2015

```

```

filtered_df['Post'] = (filtered_df['year'] == 2015).astype(int)

# Calculate the uninsured share
filtered_df['uninsured_share'] = filtered_df['uninsured'] / filtered_df['adult_pop']

# Run Difference-in-Differences regression including all states
model_all_states = smf.ols('uninsured_share ~ D + Post + D:Post', data=filtered_df).fit()

# Show the regression summary
print(model_all_states.summary())

```

ValueError: cannot convert float NaN to integer

ValueError Traceback (most recent call last)

Input In [9], in <cell line: 6>()

```

3 filtered_df = medicaid_data[medicaid_data['year'].isin([2012, 2015])]
5 # Create treatment indicator (D) for any state that ever expanded Medicaid
----> 6 filtered_df['D'] = filtered_df['expand_ever'].astype(int)
8 # Create post-period indicator (Post) for 2015
9 filtered_df['Post'] = (filtered_df['year'] == 2015).astype(int)

```

File ~/anaconda/lib/python3.9/site-packages/pandas/core/generic.py:6643, in NDFrame.astype(self, dtype, copy, errors)

```

6637     results = [
6638         ser.astype(dtype, copy=copy, errors=errors) for _, ser in self.items()
6639     ]
6641 else:
6642     # else, only a single dtype is given
-> 6643     new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
6644     res = self._constructor_from_mgr(new_data, axes=new_data.axes)
6645     return res.__finalize__(self, method="astype")

```

File ~/anaconda/lib/python3.9/site-packages/pandas/core/internals/managers.py:430, in BaseBlockManager.astype(self, dtype, copy, errors, using_cow)

```

427     elif using_copy_on_write():
428         copy = False
--> 430     return self.apply(
431         "astype",
432         dtype=dtype,
433         copy=copy,
434         errors=errors,
435         using_cow=using_copy_on_write(),
436     )

```

File ~/anaconda/lib/python3.9/site-packages/pandas/core/internals/managers.py:363, in BaseBlockManager.apply(self, f, args, kwargs)

```

361         applied = b.apply(f, **kwargs)
362     else:

```



```

--> 363         applied = getattr(b, f)(**kwargs)
      364         result_blocks = extend_blocks(applied, result_blocks)
      366 out = type(self).from_blocks(result_blocks, self.axes)
File ~/anaconda/lib/python3.9/site-packages/pandas/core/internals/blocks.py:758, in Block.astype
      755         raise ValueError("Can not squeeze with more than one column.")
      756         values = values[0, :] # type: ignore[call-overload]
--> 758 new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
      760 new_values = maybe_coerce_values(new_values)
      762 refs = None
File ~/anaconda/lib/python3.9/site-packages/pandas/core/dtypes/astype.py:237, in astype_array
      234 dtype = dtype.numpy_dtype
      236 try:
--> 237     new_values = astype_array(values, dtype, copy=copy)
      238 except (ValueError, TypeError):
      239     # e.g. _astype_nansafe can fail on object-dtype of strings
      240     # trying to convert to float
      241     if errors == "ignore":
File ~/anaconda/lib/python3.9/site-packages/pandas/core/dtypes/astype.py:182, in astype_array
      179 values = values.astype(dtype, copy=copy)
      181 else:
--> 182     values = _astype_nansafe(values, dtype, copy=copy)
      184 # in pandas we don't store numpy str dtypes, so convert to object
      185 if isinstance(dtype, np.dtype) and issubclass(values.dtype.type, str):
File ~/anaconda/lib/python3.9/site-packages/pandas/core/dtypes/astype.py:133, in _astype_nansafe
      129 raise ValueError(msg)
      131 if copy or arr.dtype == object or dtype == object:
      132     # Explicit copy, or required since NumPy can't view from / to object.
--> 133     return arr.astype(dtype, copy=True)
      135 return arr.astype(dtype, copy=copy)
ValueError: cannot convert float NaN to integer

```

#Question 9

```

final_data.loc[final_data['expand_ever'] == False, 'event_time'] = -99

for yr in range(-2, 4): # event window from -2 to +3 years
    if yr != -1: # omit -1 as base year
        final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)

final_data['uninsured_share'] = final_data['uninsured'] / final_data['adult_pop']

# Run regression with event dummies, state fixed effects, and year fixed effects
event_dummies = ' + '.join([f'event_{yr}' for yr in range(-2, 4) if yr != -1])

```

```
formula = f'uninsured_share ~ {event_dummies} + C(State) + C(year)'
model = smf.ols(formula, data=final_data).fit()
```

/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:7: 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
final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:7: 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
final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:7: 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
final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:7: 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
final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:7: 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
final_data[f'event_{yr}'] = (final_data['event_time'] == yr).astype(int)
/var/folders/2q/wzjp_2kd355b8clhzqwmtytb40000gn/T/ipykernel_44777/1343767846.py:9: 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
final_data['uninsured_share'] = final_data['uninsured'] / final_data['adult_pop']

PatsyError: numbers besides '0' and '1' are only allowed with **
uninsured_share ~ event_-2 + event_0 + event_1 + event_2 + event_3 + C(State) + C(year)

```

-----
PatsyError                                Traceback (most recent call last)
Input In [12], in <cell line: 14>()
    12 event_dummies = ' + '.join([f'event_yr' for yr in range(-2, 4) if yr != -1])
    13 formula = f'uninsured_share ~ event_dummies + C(State) + C(year)'
--> 14 model = smf.ols(formula, data=final_data).fit()
File ~/anaconda/lib/python3.9/site-packages/statsmodels/base/model.py:200, in Model.from_formula
    197 if missing == 'none': # with patsy it's drop or raise. let's raise.
    198     missing = 'raise'
--> 200 tmp = handle_formula_data(data, None, formula, depth=eval_env,
    201                             missing=missing)
    202 ((endog, exog), missing_idx, design_info) = tmp
    203 max_endog = cls._formula_max_endog
File ~/anaconda/lib/python3.9/site-packages/statsmodels/formula/formulatools.py:63, in handle_formula_data
    61 else:
    62     if data_util._is_using_pandas(Y, None):
--> 63         result = dmatrices(formula, Y, depth, return_type='dataframe',
    64                             NA_action=na_action)
    65     else:
    66         result = dmatrices(formula, Y, depth, return_type='dataframe',
    67                             NA_action=na_action)
File ~/anaconda/lib/python3.9/site-packages/patsy/highlevel.py:309, in dmatrices(formula_like, data, eval_env, NA_action, return_type)
    299 """Construct two design matrices given a formula_like and data.
    300
    301 This function is identical to :func:`dmatrix`, except that it requires
    (...)
    306 See :func:`dmatrix` for details.
    307 """
    308 eval_env = EvalEnvironment.capture(eval_env, reference=1)
--> 309 (lhs, rhs) = _do_highlevel_design(formula_like, data, eval_env,
    310                                   NA_action, return_type)
    311 if lhs.shape[1] == 0:
    312     raise PatsyError("model is missing required outcome variables")
File ~/anaconda/lib/python3.9/site-packages/patsy/highlevel.py:164, in _do_highlevel_design(formula_like, data, eval_env, NA_action, return_type)
    162 def data_iter_maker():
    163     return iter([data])
--> 164 design_infos = _try_incr_builders(formula_like, data_iter_maker, eval_env,
    165                                   NA_action)
    166 if design_infos is not None:
    167     return build_design_matrices(design_infos, data,
    168                                 NA_action=NA_action,
    169                                 return_type=return_type)

```

```

File ~/anaconda/lib/python3.9/site-packages/patsy/highlevel.py:62, in _try_incr_builders(form
    55         raise PatsyError(
    56             "On Python 2, formula strings must be either 'str' objects, "
    57             "or else 'unicode' objects containing only ascii "
    58             "characters. You passed a unicode string with non-ascii "
    59             "characters. I'm afraid you'll have to either switch to "
    60             "ascii-only, or else upgrade to Python 3.")
    61 if isinstance(formula_like, str):
--> 62     formula_like = ModelDesc.from_formula(formula_like)
    63     # fallthrough
    64 if isinstance(formula_like, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:165, in ModelDesc.from_formula(cls
    163 else:
    164     tree = parse_formula(tree_or_string)
--> 165 value = Evaluator().eval(tree, require_evalexp=False)
    166 assert isinstance(value, cls)
    167 return value
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:400, in Evaluator.eval(self, tree,
    396 if key not in self._evaluators:
    397     raise PatsyError("I don't know how to evaluate this "
    398                       "'%s' operator" % (tree.type,),
    399                       tree.token)
--> 400 result = self._evaluators[key](self, tree)
    401 if require_evalexp and not isinstance(result, IntermediateExpr):
    402     if isinstance(result, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:221, in _eval_any_tilde(evaluator,
    220 def _eval_any_tilde(evaluator, tree):
--> 221     exprs = [evaluator.eval(arg) for arg in tree.args]
    222     if len(exprs) == 1:
    223         # Formula was like: "~ foo"
    224         # We pretend that instead it was like: "0 ~ foo"
    225         exprs.insert(0, IntermediateExpr(False, None, True, []))
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:221, in <listcomp>(.0)
    220 def _eval_any_tilde(evaluator, tree):
--> 221     exprs = [evaluator.eval(arg) for arg in tree.args]
    222     if len(exprs) == 1:
    223         # Formula was like: "~ foo"
    224         # We pretend that instead it was like: "0 ~ foo"
    225         exprs.insert(0, IntermediateExpr(False, None, True, []))
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:400, in Evaluator.eval(self, tree,
    396 if key not in self._evaluators:
    397     raise PatsyError("I don't know how to evaluate this "
    398                       "'%s' operator" % (tree.type,),

```

```

399             tree.token)
--> 400 result = self._evaluators[key](self, tree)
401 if require_evalexpr and not isinstance(result, IntermediateExpr):
402     if isinstance(result, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:233, in _eval_binary_plus(evaluator, tree):
232 def _eval_binary_plus(evaluator, tree):
--> 233     left_expr = evaluator.eval(tree.args[0])
234     if tree.args[1].type == "ZERO":
235         return IntermediateExpr(False, None, True, left_expr.terms)
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:400, in Evaluator.eval(self, tree, key):
396 if key not in self._evaluators:
397     raise PatsyError("I don't know how to evaluate this "
398                     "'%s' operator" % (tree.type,),
399                     tree.token)
--> 400 result = self._evaluators[key](self, tree)
401 if require_evalexpr and not isinstance(result, IntermediateExpr):
402     if isinstance(result, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:233, in _eval_binary_plus(evaluator, tree):
232 def _eval_binary_plus(evaluator, tree):
--> 233     left_expr = evaluator.eval(tree.args[0])
234     if tree.args[1].type == "ZERO":
235         return IntermediateExpr(False, None, True, left_expr.terms)
[... skipping similar frames: Evaluator.eval at line 400 (4 times), _eval_binary_plus at
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:233, in _eval_binary_plus(evaluator, tree):
232 def _eval_binary_plus(evaluator, tree):
--> 233     left_expr = evaluator.eval(tree.args[0])
234     if tree.args[1].type == "ZERO":
235         return IntermediateExpr(False, None, True, left_expr.terms)
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:400, in Evaluator.eval(self, tree, key):
396 if key not in self._evaluators:
397     raise PatsyError("I don't know how to evaluate this "
398                     "'%s' operator" % (tree.type,),
399                     tree.token)
--> 400 result = self._evaluators[key](self, tree)
401 if require_evalexpr and not isinstance(result, IntermediateExpr):
402     if isinstance(result, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:256, in _eval_binary_minus(evaluator, tree):
254     return IntermediateExpr(False, None, True, left_expr.terms)
255 else:
--> 256     right_expr = evaluator.eval(tree.args[1])
257     terms = [term for term in left_expr.terms
258             if term not in right_expr.terms]
259     if right_expr.intercept:

```

```

File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:400, in Evaluator.eval(self, tree,
    396 if key not in self._evaluators:
    397     raise PatsyError("I don't know how to evaluate this "
    398                     "'%s' operator" % (tree.type,),
    399                     tree.token)
--> 400 result = self._evaluators[key](self, tree)
    401 if require_evaexpr and not isinstance(result, IntermediateExpr):
    402     if isinstance(result, ModelDesc):
File ~/anaconda/lib/python3.9/site-packages/patsy/desc.py:354, in _eval_number(evaluator, tree)
    353 def _eval_number(evaluator, tree):
--> 354     raise PatsyError("numbers besides '0' and '1' are "
    355                     "only allowed with **", tree)
PatsyError: numbers besides '0' and '1' are only allowed with **
    uninsured_share ~ event_-2 + event_0 + event_1 + event_2 + event_3 + C(State) + C(year)
                        ^

```

```
#Question 10
```