socioeconomic_models

August 1, 2025

[1]:

```
# analytics
import pandas as pd
import numpy as np
import scipy.stats as stats
import statsmodels.formula.api as smf
#spatial
import osmnx as ox
import geopandas as gpd
import contextily as cx
# plotting
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap
#settings
import warnings

# set dataframe outputs to three digits
pd.set_option("display.precision", 3)
#suppress warnings
warnings.filterwarnings('ignore')
```

[2]:

```
# import data
path = '/Users/philip/Documents/ESE/ESE_thesis/flood_experience/data/export/
  ↪clean_n.csv'
df_n = pd.read_csv(path)
df_n.columns
```

[2]:

```
Index(['id', 'state', 'zipcode', 'geographic_division', 'census_region',
       'county', 'experience', 'supplies', 'insured', 'involved',
       'learned_routes', 'made_plan', 'made_safer', 'planned_neighbors',
       'practiced_drills', 'documents', 'rainy_day', 'alerts',
       'family_communication', 'none', 'dont_know', 'age', 'sex', 'education',
       'race', 'homeownership', 'income', 'rentmortgage', 'rurality',
       'hazard_weight', 'geometry', 'zip_count'],
      dtype='object')
```

[3]:
```
path = '/Users/philip/Documents/ESE/ESE_thesis/flood_experience/data/export/
 ↪clean_k.csv'
df_k = pd.read_csv(path)
df_k.columns
```

[3]:
```
Index(['id', 'state', 'zipcode', 'geographic_division', 'census_region',
       'county', 'awareness', 'perception', 'experience', 'floodzone',
       'supplies', 'insured', 'involved', 'learned_routes', 'made_plan',
       'made_safer', 'planned_neighbors', 'practiced_drills', 'documents',
       'rainy_day', 'alerts', 'family_communication', 'none', 'dont_know',
       'age', 'sex', 'education', 'race', 'homeownership', 'income',
       'rentmortgage', 'rurality', 'hazard_weight', 'geometry', 'zip_count'],
      dtype='object')
```

We will go through these outcome variables one by one structural adaptation: - made_safer

non-structural adaptation: - insurance - learned_routes - supplies - involved - made_plan - practiced_drills - alerts - family_communication

[4]:
```
def r_square(model):
    # McKelvay-Zavoina
    xb = model.predict(linear=True) #fitted latent value
    var_xb = np.var(xb,ddof=1) # variance of xb
    r2_mz = var_xb / (var_xb + 1) # McKelvay-Zavoina R_2
    # McFadden
    r2_mf = model.prsquared
    return r2_mz
```

[5]:
```
def probit(functions, data):
    results_list = []
    for func in functions:
        model = smf.probit(formula=func, data=data).fit(disp=0)
        df_model = pd.DataFrame({
            'effect': model.params,
            'p': model.pvalues,
            'pseudoR_2': r_square(model),
            'LLPr': model.llr_pvalue,
            'BIC': model.bic
        })
        df_marginal = model.get_margeff().summary_frame()
        df_model = pd.concat([df_model, df_marginal], axis =1)

        df_model.index = pd.MultiIndex.from_product([[func], df_model.index],␣
 ↪names=['function', 'beta'])
        results_list.append(df_model)
    results = pd.concat(results_list)
    return results
```

[6]:

```
#duplicate but with logit
def logit(functions, data):
    results_list = []
    for func in functions:
        model = smf.logit(formula=func, data=data).fit(disp=0)
        marg_effects = model.get_margeff().summary_frame()

        df_model = pd.DataFrame({
            'effect': model.params,
            'p': model.pvalues,
            'marginal_effect': marg_effects['dy/dx'],
            'pseudoR_2': model.prsquared,
            'LLPr': model.llr_pvalue,
            'BIC': model.bic
        })
        df_model.index = pd.MultiIndex.from_product([[func], df_model.index],␣
 ↪names=['function', 'beta'])
        results_list.append(df_model)
    results = pd.concat(results_list)
    return results
```

[7]:

```
functions = [
    'made_safer ~ age + income + sex + education + homeownership +␣
 ↪rentmortgage',
    'documents ~ age + income + sex + education + homeownership + rentmortgage',
    'insured ~ age + income + sex + education + homeownership + rentmortgage',
    'learned_routes ~ age + income + sex + education + homeownership +␣
 ↪rentmortgage',
    'supplies ~ age + income + sex + education + homeownership + rentmortgage',
    'involved ~ age + income + sex + education + homeownership + rentmortgage',
    'made_plan ~ age + income + sex + education + homeownership + rentmortgage',
    'practiced_drills ~ age + income + sex + education + homeownership +␣
 ↪rentmortgage',
    'alerts ~ age + income + sex + education + homeownership + rentmortgage',
    'family_communication ~ age + income + sex + education + homeownership +␣
 ↪rentmortgage'
]
```

[8]:

```
results = probit(functions=functions, data=df_n)
results = results.round(3) # set to three decimal places
results.to_excel('results/probit_socioeconomic.xlsx')
```