

# hurricanes

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 processed data
path = '/Users/philip/Documents/ESE/ESE_thesis/flood_experience/data/export/
↳clean_k.csv'
df = pd.read_csv(path)
df.columns
```

```
[2]: 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')
```

```
[3]: # add hurricane dummy
df['hurricane'] = np.nan
hurricane_state = 'Florida'
non_hurricane = ['Washington', 'Oregon', 'Pennsylvania']
df['hurricane'] = [1 if s in hurricane_state else 0 if s in non_hurricane else
↳ np.nan for s in df.state]
```

```
[4]: df = df.dropna()
df.shape
```

```
[4]: (270, 36)
```

```
[5]: def r_square(model):
    # McKelvey-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) # McKelvey-Zavoina R_2
    # McFadden
    r2_mf = model.prsquared
    return r2_mz
```

```
[6]: def probit(functions, data):
    results_list = []
    for func in functions:
        model = smf.probit(formula=func, data=data).fit(dis=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
```

```
[7]: #duplicate but with logit
def logit(functions, data):
    results_list = []
    for func in functions:
        model = smf.logit(formula=func, data=data).fit(dis=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

```

```

[8]: # functions
M1 = [
    'made_safer ~ hurricane + perception + age + homeownership',
    'documents ~ hurricane + perception + age + homeownership',
    'learned_routes ~ hurricane + perception + age + homeownership',
    'supplies ~ hurricane + perception + age + homeownership',
    'involved ~ hurricane + perception + age + homeownership',
    'made_plan ~ hurricane + perception + age + homeownership',
    'practiced_drills ~ hurricane + perception + age + homeownership',
    'alerts ~ hurricane + perception + age + homeownership',
    'family_communication ~ hurricane + perception + age + homeownership',
    'insured ~ hurricane + perception + age + homeownership'
]

M2 = [
    'made_safer ~ hurricane + awareness + age + homeownership',
    'documents ~ hurricane + awareness + age + homeownership',
    'learned_routes ~ hurricane + awareness + age + homeownership',
    'supplies ~ hurricane + awareness + age + homeownership',
    'involved ~ hurricane + awareness + age + homeownership',
    'made_plan ~ hurricane + awareness + age + homeownership',
    'practiced_drills ~ hurricane + awareness + age + homeownership',
    'alerts ~ hurricane + awareness + age + homeownership',
    'family_communication ~ hurricane + awareness + age + homeownership',
    'insured ~ hurricane + awareness + age + homeownership'
]

```

```

[9]: # probit regression
M1 = probit(functions = M1, data = df)
M2 = probit(functions = M2, data = df)

```

```
[10]: # store results
M1, M2 = M1.round(3), M2.round(3)
M1.to_excel('results/M1.xlsx')
M2.to_excel('results/M2.xlsx')
```

### 0.0.1 Sensitivity Check: Exclude dummy

```
[11]: # functions
D1 = [
    'made_safer ~ perception + age + homeownership',
    'documents ~ perception + age + homeownership',
    'learned_routes ~ perception + age + homeownership',
    'supplies ~ perception + age + homeownership',
    'involved ~ perception + age + homeownership',
    'made_plan ~ perception + age + homeownership',
    'practiced_drills ~ perception + age + homeownership',
    'alerts ~ perception + age + homeownership',
    'family_communication ~ perception + age + homeownership',
    'insured ~ perception + age + homeownership'
]

D2 = [
    'made_safer ~ awareness + age + homeownership',
    'documents ~ awareness + age + homeownership',
    'learned_routes ~ awareness + age + homeownership',
    'supplies ~ awareness + age + homeownership',
    'involved ~ awareness + age + homeownership',
    'made_plan ~ awareness + age + homeownership',
    'practiced_drills ~ awareness + age + homeownership',
    'alerts ~ awareness + age + homeownership',
    'family_communication ~ awareness + age + homeownership',
    'insured ~ awareness + age + homeownership'
]
```

```
[12]: # probit regression
D1 = probit(functions = D1, data = df)
D2 = probit(functions = D2, data = df)
```

```
[13]: # store results
D1, D2 = D1.round(3), D2.round(3)
D1.to_excel('results/D1.xlsx')
D2.to_excel('results/D2.xlsx')
```

## 0.0.2 Now including insurance

```
[14]: # functions
M1 = [
    'made_safer ~ hurricane + perception + age + homeownership + insured',
    'documents ~ hurricane + perception + age + homeownership + insured',
    'learned_routes ~ hurricane + perception + age + homeownership + insured',
    'supplies ~ hurricane + perception + age + homeownership + insured',
    'involved ~ hurricane + perception + age + homeownership + insured',
    'made_plan ~ hurricane + perception + age + homeownership + insured',
    'practiced_drills ~ hurricane + perception + age + homeownership + insured',
    'alerts ~ hurricane + perception + age + homeownership + insured',
    'family_communication ~ hurricane + perception + age + homeownership +  

    ↪insured',
]
M2 = [
    'made_safer ~ hurricane + awareness + age + homeownership + insured',
    'documents ~ hurricane + awareness + age + homeownership + insured',
    'learned_routes ~ hurricane + awareness + age + homeownership + insured',
    'supplies ~ hurricane + awareness + age + homeownership + insured',
    'involved ~ hurricane + awareness + age + homeownership + insured',
    'made_plan ~ hurricane + awareness + age + homeownership + insured',
    'practiced_drills ~ hurricane + awareness + age + homeownership + insured',
    'alerts ~ hurricane + awareness + age + homeownership + insured',
    'family_communication ~ hurricane + awareness + age + homeownership +  

    ↪insured',
]
```

```
[15]: # probit regression
M1 = probit(functions = M1, data = df)
M2 = probit(functions = M2, data = df)
```

```
[16]: # store results
M1, M2 = M1.round(3), M2.round(3)
M1.to_excel('results/M1_insured.xlsx')
M2.to_excel('results/M2_insured.xlsx')
```