univariate_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 df2img
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap
#settings
import warnings

#set output to 3 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')
```

[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, determinant, data):
    # create a dataframe to capture results
    results = pd.DataFrame(columns=['effect', 'p', 'marginal_effect',␣
 ↪'pseudoR_2', 'LLRp','BIC'])
    # iterate over functions and compute models and capture results
    for var in functions:
        model = smf.probit(formula=var, data=data).fit(disp=0) # run the model
        #fill the dataframe columns
        results.at[var,'effect'] = model.params[determinant]
        results.at[var,'p'] = model.pvalues[determinant]
        temporary = model.get_margeff().summary_frame()
        results.at[var,'marginal_effect'] = temporary.at[determinant,'dy/dx']
        results.at[var,'pseudoR_2'] = r_square(model)
        results.at[var,'LLRp'] = model.llr_pvalue
        results.at[var,'BIC'] = model.bic
    return results
```

[6]:
```
df_k.perception
```

[6]:
```
0       1.0
1       1.0
2       1.0
3       1.0
4       1.0
```

```
       …
404    0.0
405    0.0
406    0.0
407    0.0
408    0.0
Name: perception, Length: 409, dtype: float64
```

[7]:

```
perception = [
    'made_safer ~ perception',
    'documents ~ perception',
    'insured ~ perception',
    'learned_routes ~ perception',
    'supplies ~ perception',
    'involved ~ perception',
    'made_plan ~ perception',
    'practiced_drills ~ perception',
    'alerts ~ perception',
    'family_communication ~ perception'
]

awareness = [
    'made_safer ~ awareness',
    'documents ~ awareness',
    'insured ~ awareness',
    'learned_routes ~ awareness',
    'supplies ~ awareness',
    'involved ~ awareness',
    'made_plan ~ awareness',
    'practiced_drills ~ awareness',
    'alerts ~ awareness',
    'family_communication ~ awareness'
]

experience = [
    'made_safer ~ experience',
    'documents ~ experience',
    'insured ~ experience',
    'learned_routes ~ experience',
    'supplies ~ experience',
    'involved ~ experience',
    'made_plan ~ experience',
    'practiced_drills ~ experience',
    'alerts ~ experience',
    'family_communication ~ experience'
]
```

```
floodzone = [
    'made_safer ~ floodzone',
    'documents ~ floodzone',
    'insured ~ floodzone',
    'learned_routes ~ floodzone',
    'supplies ~ floodzone',
    'involved ~ floodzone',
    'made_plan ~ floodzone',
    'practiced_drills ~ floodzone',
    'alerts ~ floodzone',
    'family_communication ~ floodzone'
]

insurance = [
    'made_safer ~ insured',
    'documents ~ insured',
    'learned_routes ~ insured',
    'supplies ~ insured',
    'involved ~ insured',
    'made_plan ~ insured',
    'practiced_drills ~ insured',
    'alerts ~ insured',
    'family_communication ~ insured'
]

age = [
    'made_safer ~ age',
    'documents ~ age',
    'insured ~ age',
    'learned_routes ~ age',
    'supplies ~ age',
    'involved ~ age',
    'made_plan ~ age',
    'practiced_drills ~ age',
    'alerts ~ age',
    'family_communication ~ age'
]

income = [
    'made_safer ~ income',
    'documents ~ income',
    'insured ~ income',
    'learned_routes ~ income',
    'supplies ~ income',
    'involved ~ income',
    'made_plan ~ income',
    'practiced_drills ~ income',
    'alerts ~ income',
    'family_communication ~ income'
]

sex = [
     'made_safer ~ sex',
    'documents ~ sex',
    'insured ~ sex',
    'learned_routes ~ sex',
    'supplies ~ sex',
    'involved ~ sex',
    'made_plan ~ sex',
    'practiced_drills ~ sex',
    'alerts ~ sex',
    'family_communication ~ sex'
]

education = [
    'made_safer ~ education',
    'documents ~ education',
    'insured ~ education',
    'learned_routes ~ education',
    'supplies ~ education',
    'involved ~ education',
    'made_plan ~ education',
    'practiced_drills ~ education',
    'alerts ~ education',
    'family_communication ~ education'
]

homeownership = [
    'made_safer ~ homeownership',
    'documents ~ homeownership',
    'insured ~ homeownership',
    'learned_routes ~ homeownership',
    'supplies ~ homeownership',
    'involved ~ homeownership',
    'made_plan ~ homeownership',
    'practiced_drills ~ homeownership',
    'alerts ~ homeownership',
    'family_communication ~ homeownership'
]

rentmortgage = [
    'made_safer ~ rentmortgage',
    'documents ~ rentmortgage',
    'insured ~ rentmortgage',
    'learned_routes ~ rentmortgage',
    'supplies ~ rentmortgage',
    'involved ~ rentmortgage',
    'made_plan ~ rentmortgage',
    'practiced_drills ~ rentmortgage',
    'alerts ~ rentmortgage',
    'family_communication ~ rentmortgage'
]
```

[ ]:

```
functions = [
    ('perception', perception, df_k),
    ('awareness', awareness, df_k),
    ('experience', experience, df_n),
    ('floodzone', floodzone, df_k),
    ('insured', insurance, df_n),
    ('age', age, df_n),
    ('income', income, df_n),
    ('sex', sex, df_n),
    ('education', education, df_n),
    ('homeownership', homeownership, df_n),
    ('rentmortgage', rentmortgage, df_n)
]



res_list = []
determinant_keys = [] # first level index

for determinant, formula_list, data in functions:
    res_df = probit(functions=formula_list, determinant=determinant, data=data)
    res_list.append(res_df)
    determinant_keys.append(determinant)
probit_df = pd.concat(res_list, keys = determinant_keys, names =␣
  ↪['Determinant','Function'])
probit_df.to_excel('results/probit_univariate.xlsx')
```

0.1 What is the effect of risk perception on awareness?

[9]:

```
functions = ['awareness ~ perception']
determinant = 'perception'
```

[10]:

```
probit(functions=functions, determinant='perception', data=df_k)
```

[10]:

```
                       effect    p marginal_effect pseudoR_2 LLRp      BIC
awareness ~ perception  0.675  0.0           0.254     0.102  0.0  549.382
```