Bilenkin530Week9_Exercises_9.2

February 8, 2025

0.0.1 Chapter 11 page 142 (Exercise 11.1 Regression)

What variables could you use to make the best prediction?

0.0.2 Variables

Gestational Age

Health Conditions

Mother's Age

Past Births (if any)

Nutrition/Diet

Work Stress (if any)

Exercise Activity

Multiple or Single Pregnancy

Pregnancy Condition

Prenatal Care

Family History

0.0.3 Chapter 11 page 142 (Exercise 11.3 Regression)

A 35 years old, black, and a college graduate woman whose annual household income exceeds $75,000. How many children would you predict she has born?

[38]:
```
import requests

urls = [
    "https://github.com/AllenDowney/ThinkStats2/raw/master/code/nsfg.py",
    "https://github.com/AllenDowney/ThinkStats2/raw/master/code/first.py",
    "https://github.com/AllenDowney/ThinkStats2/raw/master/code/2002FemPreg.
  ↪dct",
    "https://github.com/AllenDowney/ThinkStats2/raw/master/code/2002FemPreg.dat.
  ↪gz"
]
```

```
for url in urls:
    filename = url.split("/")[-1]
    response = requests.get(url)
    with open(filename, 'wb') as file:
        file.write(response.content)
    print(f"Downloaded {filename}")
```

```
Downloaded nsfg.py
Downloaded first.py
Downloaded 2002FemPreg.dct
Downloaded 2002FemPreg.dat.gz
```

It appears that the 'numbabes' column name doesn't exist in the dataset. Thus, I used column name 'nbrnaliv' which represents number of babies born alive.

[39]:
```
# Importing necessary libraries
import thinkstats2
import thinkplot
import nsfg
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import warnings

# Ignoring warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# Loading the NSFG data file using nsfg.py
pregnancy = nsfg.ReadFemPreg()

# Cleaning and preparing the data
pregnancy['birthwgt_lb'] = pregnancy['birthwgt_lb'].replace([97, 98, 99], np.
 ↪nan)
pregnancy['birthwgt_oz'] = pregnancy['birthwgt_oz'].replace([97, 98, 99], np.
 ↪nan)
pregnancy['agepreg'] = pregnancy['agepreg'].replace([97, 98, 99], np.nan)

# Extracting relevant columns and dropping NaNs at the same time
data_clean = pregnancy[['agepreg', 'nbrnaliv']].dropna()

# Define independent and dependent variables
X = data_clean[['agepreg']]
y = data_clean['nbrnaliv']

# Adding a constant to the model
X = sm.add_constant(X)
```

```
# Fitting the Poisson Regression model
poisson_model = sm.GLM(y, X, family=sm.families.Poisson()).fit()
print(poisson_model.summary())

# Manually adding new variables of the new woman's information because it's not␣
 ↪in the original dataset.
new_data = pd.DataFrame({
    'const': [1],
    'agepreg': [35],
    'race_black': [1],
    'education_college': [1],
    'income_75k_plus': [1]
})

# Predicting the number of children
prediction = poisson_model.predict(new_data[['const', 'agepreg']])
print(f'Forcasted number of children: {prediction[0]}')
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:               nbrnaliv   No. Observations:                 9144
Model:                            GLM   Df Residuals:                     9142
Model Family:                 Poisson   Df Model:                            1
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -9306.8
Date:                Sat, 08 Feb 2025   Deviance:                       211.74
Time:                        23:21:45   Pearson chi2:                     321.
No. Iterations:                     4   Pseudo R-squ. (CS):          6.694e-05
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0141      0.048     -0.297      0.767      -0.107       0.079
agepreg        0.0015      0.002      0.783      0.434      -0.002       0.005
==============================================================================
Forcasted number of children: 1.03744178658645
```

We can state that on average any black woman who is 35 years old with a college graduate whose annual household income exceeds $75,000 will give a birth to 1.04 child or one.

0.0.4 Chapter 11 page 143 (Exercise 11.4 Regression)

A woman who is 25 years old, white, and a high school graduate whose annual household income is about $45,000. What is the probability that she is married, cohabitating, etc?

[40]:
```
# Importing necessary libraries
import nsfg
import pandas as pd
```

```python
import numpy as np
import statsmodels.api as sm
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# Load the NSFG data file using nsfg.py
pregnancy = nsfg.ReadFemPreg()

# Clean and prepare the data
pregnancy['agepreg'] = pregnancy['agepreg'].replace([97, 98, 99], np.nan)
pregnancy['rmarital'] = pregnancy['rmarital'].replace([97, 98, 99], np.nan)
pregnancy['educat'] = pregnancy['educat'].replace([7, 8], np.nan)

# Extract relevant columns and drop NaNs
data_clean = pregnancy[['agepreg', 'race', 'educat', 'rmarital']].dropna()

# Create dummy variables
data_clean = pd.get_dummies(data_clean, columns=['race', 'educat'],␣
 ↪drop_first=True)

# Ensure all data is numeric
data_clean = data_clean.apply(pd.to_numeric)

# Define independent and dependent variables based on available dummy variables
# Assuming 'race_2' corresponds to 'white' and 'educat_10' corresponds to 'high␣
 ↪school graduate'
X = data_clean[['agepreg', 'race_2', 'educat_10']]
y = data_clean['rmarital']

# Convert dependent and independent variables to numeric arrays explicitly
X = np.asarray(X, dtype=np.float64)
y = np.asarray(y, dtype=np.int64)

# Add a constant to the model
X = sm.add_constant(X)

# Fit the multinomial logistic regression model
mnlogit_model = sm.MNLogit(y, X).fit()
print(mnlogit_model.summary())

# Add new woman's information
new_data = pd.DataFrame({
    'const': [1],
    'agepreg': [25],
    'race_2': [1],             # Assuming white race coded as 1
    'educat_10': [1]           # Assuming high school education coded as 1
})
```

```python
# Ensure new data is numeric and convert to array
new_data = np.asarray(new_data, dtype=np.float64)

# Predict the probabilities for marital status
prediction = mnlogit_model.predict(new_data)

# Define labels for marital status categories
marital_status_labels = ["Married", "Cohabitating", "Widowed", "Divorced",␣
 ↪"Separated", "Never Married"]

# Printing each probability with corresponding label
print('The forcasted probabilities for 25 years old, white woman, and high␣
 ↪school graduate with $45,000 income as follows:')
for label, prob in zip(marital_status_labels, prediction[0]):
    print(f'{label}: {prob:.2%}')
```

```
Optimization terminated successfully.
         Current function value: 1.264024
         Iterations 8
                          MNLogit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                13241
Model:                        MNLogit   Df Residuals:                    13221
Method:                           MLE   Df Model:                           15
Date:                Sat, 08 Feb 2025   Pseudo R-squ.:                 0.06246
Time:                        23:21:47   Log-Likelihood:                -16737.
converged:                       True   LL-Null:                       -17852.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
       y=2       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.8884      0.142      6.245      0.000       0.610       1.167
x1            -0.0877      0.006    -15.401      0.000      -0.099      -0.077
x2            -0.6702      0.062    -10.767      0.000      -0.792      -0.548
x3             1.1274      0.100     11.282      0.000       0.932       1.323
------------------------------------------------------------------------------
       y=3       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.9532      0.384     -7.684      0.000      -3.707      -2.200
x1            -0.0195      0.014     -1.350      0.177      -0.048       0.009
x2            -0.6325      0.171     -3.688      0.000      -0.969      -0.296
x3             0.4524      0.334      1.353      0.176      -0.203       1.108
------------------------------------------------------------------------------
       y=4       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3298      0.139     -2.375      0.018      -0.602      -0.058
```

```
x1            -0.0421      0.005     -8.019      0.000      -0.052      -0.032
x2            -0.3495      0.064     -5.460      0.000      -0.475      -0.224
x3             0.3353      0.127      2.646      0.008       0.087       0.584
------------------------------------------------------------------------------
       y=5       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1427      0.168     -0.848      0.397      -0.473       0.187
x1            -0.0542      0.007     -8.212      0.000      -0.067      -0.041
x2            -1.0208      0.074    -13.842      0.000      -1.165      -0.876
x3             0.8313      0.129      6.448      0.000       0.579       1.084
------------------------------------------------------------------------------
       y=6       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.5513      0.122     20.926      0.000       2.312       2.790
x1            -0.1151      0.005    -23.040      0.000      -0.125      -0.105
x2            -1.6240      0.052    -31.123      0.000      -1.726      -1.522
x3             0.7228      0.097      7.488      0.000       0.534       0.912
==============================================================================
The forcasted probabilities for 25 years old, white woman, and high school
graduate with $45,000 income as follows:
Married: 45.85%
Cohabitating: 19.65%
Widowed: 1.23%
Divorced: 11.35%
Separated: 8.48%
Never Married: 13.43%
```