Empirical Project

May 12, 2023

ECON 3140

By Matthew Wear

In this project, we look at the paper "Mortage Lending in Boston: Interpreting HMDA Data" by Alicia Munnell, Geofferey Tootell, Lynn Browne, and James McEneaney.

Another paper titled "Evidence on Discrimination in Mortgage Lending" by Helen Ladd was used as a supplement.

Questions:

1. Economic substance
2. Indicator variable for race
3. Estimates from the fraction of applicants rejected by race
4. OLS regression with payment-to-income ratio and race omitted
5. Predicted rejection rates
6. Probit regression with payment-to-income ratio and race omitted
7. OLS and Probit regressions with race included
8. Problems with race as a causal factor for denial
9. Regressions with additional variables
10. Economic interpretation of the effect of race
11. Interpreting the Probit estimated coefficient for race
12. Predicted effect of average sample and average predicted effect

1 Economic substance

The authors want to detect the effect of race on the probability of a loan being denied to a minority applicant. Their results consider three racial groups: whites, blacks, and Hispanics. This paper is a follow-up to previous studies that looked at the problem of racial discrimination in the Boston mortgage lending market using data from the Home Mortgage Disclosure Act (HMDA). One issue with prior studies was the lack of regressors that are highly coorelated with race. Including these variables would provide a less biased estimate for the effect of race. There are several important comments that will not be described here. The data was a mix of binary and continuous variables. The authors interviewed lenders to determine which variables were relevant or not in this study and provided counterarguments to challenges against their data collection of the independent and dependent variables (e.g., what defines a declined loan).

The authors conclude that race is a significant factor when comparing white and black applicants. The results collected for discrimination of Hispanic borrowers was not statistically significant, mostly due to the fact that not enough data was available for this group. Specifically, the authors ran an OLS and Logit regression and found that these models are not only much stronger by including more relevant variables which the authors confidently find to exhibit all factors that lenders consider, but also the difference in the predicted mean rejection rates between rejected and accepted applicants increased by twenty-seven percentage points from the original HMDA model. The estimated coefficient of race was 1.00 in the Logit model and 0.07 in the OLS model. These coefficients tell us that the probability of loan denial was 8.2 percentage points higher (Logit) and 7 percentage points (!) higher (OLS) for black than white applicants.

These results demonstrate racial discrimination in the market for mortgages. Concerns about economic inequality and discrimination that led to the observed data itself is not considered. Finally, the authors find the argument that the lenders were statistically discriminating to maximize profits based on race is weak, since data on race was not collected and no prior model using race was estimated by lenders.

2 Indicator variable for race

Transform data for the dependent variable into a indicator variable for *denied* since the HDMA data categorizes outcomes as five possible values. Also, create an indicator variable that takes the value 1 for black applicants and 0 for white applicants.

Only applications that are explicitly denied will be considered to be a loan denial. Similarly, we will create a variable for black applicants that takes the value 1 if the applicant is black, 0 if white, and na if other.

We will only keep observations for loans that are explicitly approved or denied as well as originating from a white or black applicant.

Definitions for each variable is found in the codebook.

[121]:
```
import numpy as np
import pandas as pd
import math
import statsmodels.api as sm
```

[122]:
```
data = pd.read_stata('hmda-project.dta')
data.head()
```

[122]:
```
   seq   s3   s4   s5     s6   s7      s9  s11  s13  s14  ...     PI     HI  \
0  2.0  1.0  1.0  1.0   88.0  1.0  1120.0  0.0  5.0  5.0  ...  0.221  0.221
1  3.0  1.0  1.0  1.0  118.0  1.0  1120.0  0.0  5.0  5.0  ...  0.265  0.265
2  7.0  1.0  1.0  1.0  185.0  1.0  1120.0  0.0  5.0  5.0  ...  0.372  0.248
```

```
3   9.0  1.0  1.0  1.0  185.0  1.0  1120.0  0.0  5.0  5.0  ...  0.320  0.250
4  10.0  1.0  1.0  1.0  330.0  1.0  1120.0  0.0  5.0  5.0  ...  0.360  0.350

         LV  MLV  HLV  MCS  CCS self  black_verify NoMI
0  0.800000  1.0  0.0  2.0  5.0  0.0           0.0  0.0
1  0.921875  1.0  0.0  2.0  2.0  0.0           0.0  0.0
2  0.943878  1.0  0.0  2.0  1.0  0.0           0.0  0.0
3  0.880952  1.0  0.0  2.0  1.0  0.0           0.0  0.0
4  0.600000  0.0  0.0  1.0  1.0  0.0           0.0  0.0

[5 rows x 73 columns]
```

[123]:

```python
# note that in the provided data, denied is already created

# make any value of 's7' (denied) that is not 3 equal to 0, otherwise 1
denied = np.where(data['s7'] == 3, 1, 0)

# make 'black' column that takes the value 1 if black and 0 if white
black = 0
black = np.where(data['s13'] == 3, 1, black)
black = np.where(data['s13'] == 5, 0, black)

# concatenate columns
new_data = pd.DataFrame({'black': black, 'denied': denied})

# remove rows where 'black' is still 2 (i.e., other races)
new_data = new_data[new_data['black'].isin([0, 1])]

# reset the data frame indices
new_data = new_data.reset_index(drop=True)

# print the result
print(new_data)
```

```
      black  denied
0         0       0
1         0       0
2         0       0
3         0       0
4         0       0
...     ...     ...
2375      0       0
2376      0       0
2377      0       0
2378      1       1
2379      0       1
```

[2380 rows x 2 columns]

3 Estimates from the fraction of applicants rejected by race

The fraction of black applicants in the sample that was rejected among black applicants is 28.3%. The fraction of white applicants in the sample that was rejected among white applicants is 9.3%.

[124]:
```
black_denied = new_data[new_data['black'] == 1]['denied'].mean()
white_denied = new_data[new_data['black'] == 0]['denied'].mean()
print(black_denied, 1 - black_denied)
print(white_denied, 1 - white_denied)
```

```
0.2831858407079646 0.7168141592920354
0.0926016658500735 0.9073983341499265
```

	Denied	Approved
Black	0.283	0.717
White	0.093	0.907

In the model,

$$denied = \beta_0 + \beta_1 black + u$$

We can calculate estimates without running the regression, since

$$\begin{aligned}\beta_0 &= \overline{denied_{black=0}} \\ &= 0.093\end{aligned}$$

$$\begin{aligned}\beta_1 &= \overline{denied_{black=1}} - \overline{denied_{black=0}} \\ &= 0.190\end{aligned}$$

We will run the regression to confirm these estimates.

[125]:
```
# regress 'denied' on 'PI'
X = new_data[['black']].copy()
X = sm.add_constant(X)
y = new_data[['denied']].copy()

model = sm.OLS(y, X)
results1 = model.fit()
results2 = model.fit(cov_type='HC3')
print("Parameters: ")
```

```
print(results1.params) # estimates
print("Conventional standard errors:")
print(results1.bse) # standard errors
print("Heteroskedastic-robust standard errors")
print(results2.bse)
```

```
Parameters:
const    0.092602
black    0.190584
dtype: float64
Conventional standard errors:
const    0.007037
black    0.018644
dtype: float64
Heteroskedastic-robust standard errors
const    0.006419
black    0.025368
dtype: float64
```

4 OLS regression with payment-to-income ratio and race omitted

We will now estimate

$$denied = \beta_0 + \beta_1 PI + u$$

[126]:

```
# add a column for the payment-to-income ratio
# as your PI ratio increases, denied should be more likely
new_data['PI'] = data['s46']/100

# regress 'denied' on 'PI'
X = new_data[['PI']].copy()
X = sm.add_constant(X)
y = new_data[['denied']].copy()

model = sm.OLS(y, X)
results1 = model.fit()
results2 = model.fit(cov_type='HC3')
print("Parameters: ")
print(results1.params) # estimates
print("Conventional standard errors:")
print(results1.bse) # standard errors
print("Heteroskedastic-robust standard errors")
print(results2.bse)
results1.summary()
```

```
Parameters:
const   -0.079910
```

```
PI        0.603535
dtype: float64
Conventional standard errors:
const     0.021158
PI        0.060840
dtype: float64
Heteroskedastic-robust standard errors
const     0.038458
PI        0.118287
dtype: float64
```

[126]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                 denied   R-squared:                       0.040
Model:                            OLS   Adj. R-squared:                  0.039
Method:                 Least Squares   F-statistic:                     98.41
Date:                Fri, 12 May 2023   Prob (F-statistic):           9.37e-23
Time:                        22:21:42   Log-Likelihood:                -651.42
No. Observations:                2380   AIC:                             1307.
Df Residuals:                    2378   BIC:                             1318.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0799      0.021     -3.777      0.000      -0.121      -0.038
PI             0.6035      0.061      9.920      0.000       0.484       0.723
==============================================================================
Omnibus:                     1018.085   Durbin-Watson:                   1.461
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3273.764
Skew:                           2.280   Prob(JB):                         0.00
Kurtosis:                       6.497   Cond. No.                         10.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

The economic interpretation of these estimates tells us that increasing the payment-to-income ratio by 0.1 (i.e. after dividing PI by 100) would increase the probability of denial by 60 percent.

We prefer the heteroskedastic-robust standard errors over the usual standard errors, because the model is necessarily heteroskedastic since the outcome variable is a Bernoulli.

The coefficient given for PI is statistically significant.

Economically, these results make sense. For a fixed level of income, a higher payment of a mortgage would lead to a higher payment-to-income ratio. If the lender requires a higher payment, then it is more likely that the loan would be denied to the applicant.

As seen in the provided scatter plot, the estimated coefficient is consistent with the plot.

5 Predicted rejection rates

[127]:
```
x_new = np.array([[0.20], [0.10]])
x_new = sm.add_constant(x_new)
pred = results2.predict(x_new)

print(pred)
```

```
[ 0.04079734 -0.01955615]
```

Given the current linear prediction model (LPM),

$$\mathbb{P}(denied = 1 \mid PI = 0.20) = 0.041$$

$$\mathbb{P}(denied = 1 \mid PI = 0.10) = -0.020$$

This means an applicant with $PI = 0.20$ is about 60 percent more likely to be denied than an applicant with $PI = 0.10$.

The prediction for the second applicant does not make sense, because it violates the rules of probability. Therefore, we must rely solely on the coefficient of PI for interpreting the effect of PI, or we can use a non-linear model such as Logit or Probit to be able to model low PI values without predicting negative probabilities. We will use a Probit model.

6 Probit regression with payment-to-income ratio and race omitted

[128]:
```
# use Logit or Probit to correct negative probability problem with LPM
prob_model = sm.Probit(y, X) # endog, exog
results1 = prob_model.fit(disp=False)
results2 = prob_model.fit(cov_type='HC3', disp=False)

print("Parameters: ")
print(results1.params) # estimates
print("Conventional standard errors:")
print(results1.bse) # standard errors
print("Heteroskedastic-robust standard errors")
print(results2.bse)

results1.summary()
```

```
Parameters:
const   -2.194159
PI       2.967908
dtype: float64
Conventional standard errors:
const    0.128990
PI       0.359105
dtype: float64
Heteroskedastic-robust standard errors
const    0.164941
PI       0.465224
dtype: float64
```

[128]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                          Probit Regression Results
==============================================================================
Dep. Variable:                 denied   No. Observations:                 2380
Model:                         Probit   Df Residuals:                     2378
Method:                           MLE   Df Model:                            1
Date:                Fri, 12 May 2023   Pseudo R-squ.:                 0.04620
Time:                        22:21:44   Log-Likelihood:                -831.79
converged:                       True   LL-Null:                       -872.09
Covariance Type:            nonrobust   LLR p-value:                 2.783e-19
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.1942      0.129    -17.010      0.000      -2.447      -1.941
PI             2.9679      0.359      8.265      0.000       2.264       3.672
==============================================================================
"""
```

[129]:

```python
x_new = np.array([[0.20], [0.10]])
x_new = sm.add_constant(x_new)
pred = results2.predict(x_new)

print(pred)
```

```
[0.05473526 0.02888967]
```

Using the Probit model, we predict

$$\mathbb{P}(denied = 1 \mid PI = 0.20) = 0.055$$

$$\mathbb{P}(denied = 1 \mid PI = 0.10) = 0.029$$

7 OLS and Probit regressions with race included

We will now estimate the following equation with *black* to control for the effect of race when regressing *denied* on *PI*.

$$denied = \beta_0 + \beta_1 PI + \beta_2 black + u$$

[130]:

```python
# modify input matrix
X = new_data[['PI', 'black']].copy()
X = sm.add_constant(X)

model = sm.OLS(y, X)
ols1 = model.fit()
ols2 = model.fit(cov_type='HC3')
print(ols2.summary())

prob_model = sm.Probit(y, X) # endog, exog
probit1 = prob_model.fit(disp=False)
probit2 = prob_model.fit(cov_type='HC3', disp=False)
print(probit2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 denied   R-squared:                       0.076
Model:                            OLS   Adj. R-squared:                  0.075
Method:                 Least Squares   F-statistic:                     44.10
Date:                Fri, 12 May 2023   Prob (F-statistic):           1.57e-19
Time:                        22:21:45   Log-Likelihood:                -605.61
No. Observations:                2380   AIC:                             1217.
Df Residuals:                    2377   BIC:                             1235.
Df Model:                           2
Covariance Type:                  HC3
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0905      0.033     -2.708      0.007      -0.156      -0.025
PI             0.5592      0.104      5.394      0.000       0.356       0.762
black          0.1774      0.025      7.081      0.000       0.128       0.227
==============================================================================
Omnibus:                      969.841   Durbin-Watson:                   1.517
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3013.280
Skew:                           2.168   Prob(JB):                         0.00
Kurtosis:                       6.403   Cond. No.                         10.5
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
```

```
                           Probit Regression Results
==============================================================================
Dep. Variable:                 denied   No. Observations:                 2380
Model:                         Probit   Df Residuals:                     2377
Method:                           MLE   Df Model:                            2
Date:                Fri, 12 May 2023   Pseudo R-squ.:                 0.08594
Time:                        22:21:45   Log-Likelihood:                -797.14
converged:                       True   LL-Null:                       -872.09
Covariance Type:                  HC3   LLR p-value:                 2.818e-33
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -2.2587      0.159    -14.225      0.000      -2.570      -1.948
PI             2.7416      0.444      6.174      0.000       1.871       3.612
black          0.7082      0.083      8.515      0.000       0.545       0.871
==============================================================================
```

In both regressions, the coefficient on *black* is statistically significant, which tells us that the effect of *black* on the probability of denial is non-trivial.

The coefficient of *black* from the LPM is large. The probability of denial increases by 0.18 for an applicant who is black.

In the case of the Probit, the coefficient remains significant. We cannot directly interpret the coefficient, but we can estimate the difference in probability by some value of PI fixed.

[137]:
```
x_n = np.array([[1, 0.20, 1], [1, 0.20, 0]])
pred = probit1.predict(x_n)

print(pred)
print(pred[0]-pred[1])
```

```
[0.15811076 0.04359498]
0.1145157868100415
```

Here, we see that the difference in probability for an applicant who is black and one who is not black is 0.11. This is slightly lower than our estimated effect using the LPM, but it is still a large effect on the probability of denial.

8 Problems with race as a causal factor for denial

From the paper by Munnell, interpreting causality from these simple models is not accurate, because the estimators confound the effect of unobservable economic factors that may bias the effect of race. Munnel et al. add 38 variables to control for omitted variable bias.

9 Regressions with additional variables

[243]:
```
# modify input matrix
X = new_data[['black']]
X[['PI', 'HI', 'LV']] = data[['PI', 'HI', 'LV']].copy()
X['LVsq'] = data[['LV']].copy()**2
X[['CCS', 'MCS', 'NoMI', 'self']] = data[['CCS', 'MCS', 'NoMI', 'self']].copy()
X = sm.add_constant(X)

# Delete rows with missing values
missing = np.isnan(X).any(axis=1)
X = X[~missing]
y = y[~missing]


model = sm.OLS(y, X)
ols1 = model.fit()
ols2 = model.fit(cov_type='HC3')
print(ols2.summary())

prob_model = sm.Probit(y, X)
probit1 = prob_model.fit(disp=False)
probit2 = prob_model.fit(cov_type='HC3', disp=False)
print(probit2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 denied   R-squared:                       0.232
Model:                            OLS   Adj. R-squared:                  0.229
Method:                 Least Squares   F-statistic:                     69.63
Date:                Fri, 12 May 2023   Prob (F-statistic):          3.85e-114
Time:                        23:41:43   Log-Likelihood:                -386.05
No. Observations:                2379   AIC:                             792.1
Df Residuals:                    2369   BIC:                             849.8
Df Model:                           9
Covariance Type:                  HC3
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2393      0.036     -6.712      0.000      -0.309      -0.169
black          0.1058      0.023      4.525      0.000       0.060       0.152
PI             0.5168      0.118      4.367      0.000       0.285       0.749
HI            -0.0635      0.123     -0.515      0.607      -0.305       0.178
LV             0.0617      0.039      1.574      0.115      -0.015       0.139
LVsq          -0.0142      0.009     -1.666      0.096      -0.031       0.003
CCS            0.0393      0.005      8.165      0.000       0.030       0.049
MCS            0.0261      0.011      2.273      0.023       0.004       0.049
```

```
NoMI           0.7418      0.043     17.097      0.000       0.657       0.827
self           0.0640      0.021      3.014      0.003       0.022       0.106
==============================================================================
Omnibus:                      993.194   Durbin-Watson:                   1.587
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3558.408
Skew:                           2.127   Prob(JB):                         0.00
Kurtosis:                       7.218   Cond. No.                         72.1
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
                          Probit Regression Results
==============================================================================
Dep. Variable:                 denied   No. Observations:                 2379
Model:                         Probit   Df Residuals:                     2369
Method:                           MLE   Df Model:                            9
Date:                Fri, 12 May 2023   Pseudo R-squ.:                  0.2389
Time:                        23:41:43   Log-Likelihood:                -663.62
converged:                       True   LL-Null:                       -871.96
Covariance Type:                  HC3   LLR p-value:                 3.779e-84
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -3.7031      0.519     -7.134      0.000      -4.720      -2.686
black          0.4616      0.095      4.868      0.000       0.276       0.647
PI             2.6047      0.594      4.386      0.000       1.441       3.769
HI            -0.2191      0.654     -0.335      0.738      -1.502       1.063
LV             1.2813      1.221      1.049      0.294      -1.113       3.675
LVsq          -0.5154      0.769     -0.671      0.502      -2.022       0.991
CCS            0.1886      0.020      9.338      0.000       0.149       0.228
MCS            0.1714      0.071      2.428      0.015       0.033       0.310
NoMI           2.6026      0.292      8.922      0.000       2.031       3.174
self           0.3709      0.110      3.359      0.001       0.155       0.587
==============================================================================

/var/folders/g3/57cvvdrs6ps_tdhq9wvjbjwr0000gn/T/ipykernel_56368/2035317712.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/indexing.html#returning-a-view-versus-a-copy
  X[['PI', 'HI', 'LV']] = data[['PI', 'HI', 'LV']].copy()
/var/folders/g3/57cvvdrs6ps_tdhq9wvjbjwr0000gn/T/ipykernel_56368/2035317712.py:1
1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  y = y[~missing]
```

The following slope coefficients are positive: $black$, PI, LV, CCS, MCS, $NoMI$, and $self$.

Only the slope coefficients on HI and $LVsq$ are negative.

This tells us, for example, that the effect of the loan-to-value ratio is increasing but diminishing as LV increases.

The following coefficients are significant: $black$, PI, CCS, MCS, $NoMI$, and $self$.

The sign and significance of all slope coefficients are the same across both models. These coefficients are consistent with the OLS and Logit regressions estimated in the study.

10 Economic interpretation of the effect of race

The predicted effect of $race$ in the OLS specification tells us that the probability of denial increases by 0.11 if the applicant is black.

Economically, this is also significant. By controlling for the primary factors of lenders when deciding whether to approve a mortgage or not, the market can be shown to be influenced by racial discrimination. Since the lenders are not building models that incorporate race, the argument that this effect is a form of statistical discrimination is weak. Therefore, black applicants are more likely to be denied with all other significant factors held constant.

11 Interpreting the Probit estimated coefficient for race

[226]:
```
# We use the following to get a random data point
import random
print(f"n = {len(X)}")
x_r = X.loc[random.randint(0, len(X))]
```

```
n = 2379
```

[227]:
```
# Saved results from the random data point
x_r = np.array([[1, 0, 0.320000, 0.221000, 0.897196, 0.804961, 1, 2, 0, 0], #␣
 ↪original random point
               [1, 1, 0.320000, 0.221000, 0.897196, 0.804961, 1, 2, 0, 0]]) #␣
 ↪data point with black=1
pred = probit1.predict(x_n)

print(pred)
print(pred[1]-pred[0])
```

```
[0.04927886 0.11695973]
0.06768086517688143
```

Although we cannot easily generate a number to adequately compare the effects of the LPM and Probit model with these additional covariates, we can estimate the effect by either holding constant the sample averages of all features besides $race$, or get a random data point and consider the effect of changing the value for $race$.

If we do the second approach, we find that the estimated effect of $race$ is 0.068, which is comparable to the coefficient of 0.11 in the LPM model.

12 Predicted effect of average and average predicted effect

[239]:
```
print(f"Predicted effect from OLS: 0.1058")

sample_avg = X.mean().to_numpy()
pred = ols1.predict(sample_avg)
print(f"Predicted effect of sample average: {pred[0]}")

pred = np.mean(ols1.predict(X))
print(f"Average predicted effect: {pred}")
```

```
Predicted effect from OLS: 0.1058
Predicted effect of sample average: 0.11979822969898082
Average predicted effect: 0.11979823455233272
```

The estimated coefficient is slightly smaller than both the predicted effect of the sample average and the average predicted effect. The discrepancy may arise from the OLS model specification, where the coefficient underestimates the effect of race when holding other variables constant. Alternatively, the predicted effect of the sample average and the predicted average effect may be higher in the given sample compared to effect of *race* in the true model.