21) Logit and Probit II

Vitor Kamada

October 2018

Import Libraries

```
import numpy as np

import pandas as pd from scipy

import stats import matplotlib.pyplot as plt

import statsmodels.api as sm

from statsmodels.formula.api import logit, probit

print(sm.datasets.fair.SOURCE)
```

Fair, Ray. 1978. "A Theory of Extramarital Affairs," ‘Journal of Political Economy‘, February, 45-61.

print(sm.datasets.fair.NOTE)

```
rate_marriage   : How rate marriage, 1 = very poor, 2 = poor, 3 = fair,
                  4 = good, 5 = very good
age             : Age
yrs_married     : No. years married. Interval approximations. See
                  original paper for detailed explanation.
children        : No. children
religious       : How relgious, 1 = not, 2 = mildly, 3 = fairly,
                  4 = strongly
educ            : Level of education, 9 = grade school, 12 = high
                  school, 14 = some college, 16 = college graduate,
                  17 = some graduate school, 20 = advanced degree
occupation      : 1 = student, 2 = farming, agriculture; semi-skilled,
                  or unskilled worker; 3 = white-colloar; 4 = teacher
                  counselor social worker, nurse; artist, writers;
                  technician, skilled worker, 5 = managerial,
                  administrative, business, 6 = professional with
                  advanced degree
occupation_husb : Husband's occupation. Same as occupation.
affairs         : measure of time spent in extramarital affairs
```

dta = sm.datasets.fair.load_pandas().data

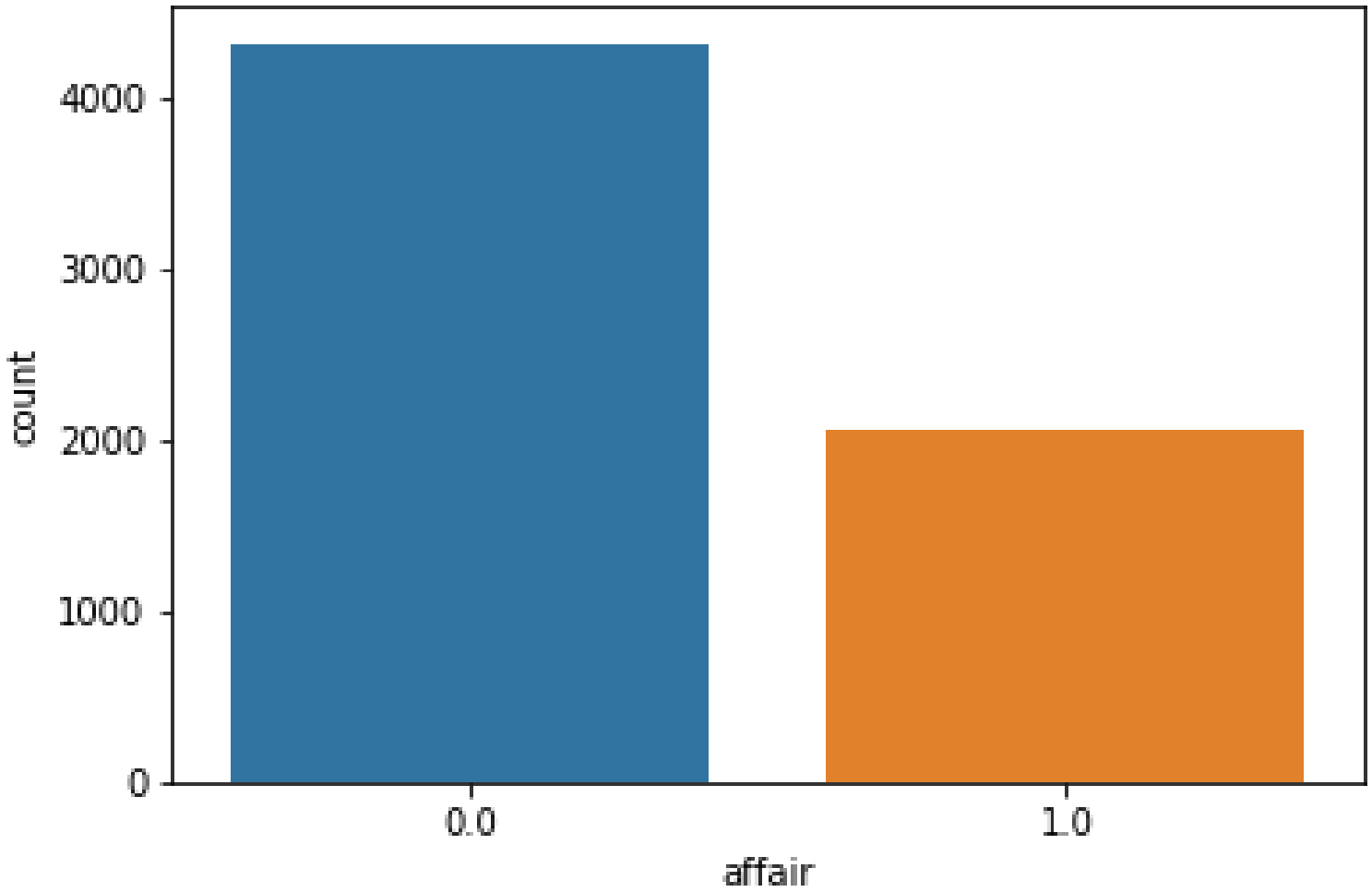
dta['affair'] = (dta['affairs'] > 0).astype(float)

	rate_marriage	age	yrs_married	children	religious	\
count	6366.000000	6366.000000	6366.000000	6366.000000	6366.000000	
mean	4.109645	29.082862	9.009425	1.396874	2.426170	
std	0.961430	6.847882	7.280120	1.433471	0.878369	
min	1.000000	17.500000	0.500000	0.000000	1.000000	
25%	4.000000	22.000000	2.500000	0.000000	2.000000	
50%	4.000000	27.000000	6.000000	1.000000	2.000000	
75%	5.000000	32.000000	16.500000	2.000000	3.000000	
max	5.000000	42.000000	23.000000	5.500000	4.000000	

	educ	occupation	occupation_husb	affairs	affair
count	6366.000000	6366.000000	6366.000000	6366.000000	6366.000000
mean	14.209865	3.424128	3.850141	0.705374	0.322495
std	2.178003	0.942399	1.346435	2.203374	0.467468
min	9.000000	1.000000	1.000000	0.000000	0.000000
25%	12.000000	3.000000	3.000000	0.000000	0.000000
50%	14.000000	3.000000	4.000000	0.000000	0.000000
75%	16.000000	4.000000	5.000000	0.484848	1.000000
max	20.000000	6.000000	6.000000	57.599991	1.000000

import seaborn as sns

sns.countplot(x='affair', data=dta)



dta.groupby('rate_marriage').mean()

rate_marriage	age	yrs_married	children
1.0	33.823232	13.914141	2.308081
2.0	30.471264	10.727011	1.735632
3.0	30.008056	10.239174	1.638469
4.0	28.856601	8.816905	1.369536
5.0	28.574702	8.311662	1.252794

rate_marriage	religious	educ	affair
1.0	2.343434	13.848485	0.747475
2.0	2.330460	13.864943	0.635057
3.0	2.308157	14.001007	0.550856
4.0	2.400981	14.144514	0.322926
5.0	2.506334	14.399776	0.181446

Pseudo R-squared

$$\text{Pseudo } R^2 = 1 - \frac{\mathcal{L}_{ur}}{\mathcal{L}_0}$$

$\mathcal{L}_0$ the model with only an intercept

$$|\mathcal{L}_{ur}| \leq |\mathcal{L}_0|$$

Logit Model

```
Logit = logit("affair ~ C(occupation) + educ + C(occupation_husb)"
              "+ rate_marriage + age + yrs_married + children"
               " + religious", dta).fit()

stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)
print(Logit.summary())
print(np.exp(Logit.params))
```

Logit Regression Results

Dep. Variable:	affair	No. Observations:	6366
Model:	Logit	Df Residuals:	6349
Method:	MLE	Df Model:	16
Date:	Sat, 29 Sep 2018	Pseudo R-squ.:	0.1365
Time:	15:42:28	Log-Likelihood:	-3456.2
converged:	True	LL-Null:	-4002.5
		LLR p-value:	1.534e-222

Odds Ratio and Raw Coefficients

		coef	std err	z	P>|z|
19.506642	Intercept	2.9708	0.572	5.192	0.000
1.477333	C(occupation)[T.2.0]	0.3902	0.448	0.872	0.383
2.019155	C(occupation)[T.3.0]	0.7027	0.441	1.592	0.111
1.602231	C(occupation)[T.4.0]	0.4714	0.443	1.065	0.287
2.869671	C(occupation)[T.5.0]	1.0542	0.447	2.360	0.018
3.028342	C(occupation)[T.6.0]	1.1080	0.494	2.242	0.025
1.185835	C(occupation_husb)[T.2.0]	0.1704	0.186	0.916	0.360
1.328662	C(occupation_husb)[T.3.0]	0.2842	0.202	1.406	0.160
1.153546	C(occupation_husb)[T.4.0]	0.1428	0.181	0.789	0.430
1.188068	C(occupation_husb)[T.5.0]	0.1723	0.183	0.944	0.345
1.200530	C(occupation_husb)[T.6.0]	0.1828	0.204	0.897	0.369
0.998276	educ	-0.0017	0.017	-0.099	0.921
0.491532	rate_marriage	-0.7102	0.031	-22.560	0.000
0.940561	age	-0.0613	0.010	-5.936	0.000
1.114021	yrs_married	0.1080	0.011	9.836	0.000
1.015768	children	0.0156	0.032	0.488	0.625
0.687024	religious	-0.3754	0.035	-10.766	0.000

Probit Model

```
Probit = probit("affair ~ C(occupation) + educ + C(occupation_husb)"
                "+ rate_marriage + age + yrs_married + children"
                " + religious", dta).fit()
```

Probit Regression Results

Dep. Variable:	affair	No. Observations:	6366
Model:	Probit	Df Residuals:	6349
Method:	MLE	Df Model:	16
Date:	Sat, 29 Sep 2018	Pseudo R-squ.:	0.1370
Time:	15:55:06	Log-Likelihood:	-3454.0
converged:	True	LL-Null:	-4002.5
		LLR p-value:	1.815e-223

print(Probit.summary())

	coef	std err	z	P>|z|
Intercept	1.7922	0.325	5.514	0.000
C(occupation)[T.2.0]	0.2107	0.250	0.843	0.399
C(occupation)[T.3.0]	0.3992	0.246	1.622	0.105
C(occupation)[T.4.0]	0.2631	0.247	1.066	0.286
C(occupation)[T.5.0]	0.6091	0.250	2.441	0.015
C(occupation)[T.6.0]	0.6480	0.279	2.321	0.020
C(occupation_husb)[T.2.0]	0.0944	0.107	0.882	0.378
C(occupation_husb)[T.3.0]	0.1663	0.117	1.420	0.156
C(occupation_husb)[T.4.0]	0.0766	0.104	0.738	0.461
C(occupation_husb)[T.5.0]	0.0916	0.105	0.872	0.383
C(occupation_husb)[T.6.0]	0.1030	0.117	0.877	0.380
educ	-0.0018	0.010	-0.180	0.857
rate_marriage	-0.4251	0.018	-23.157	0.000
age	-0.0359	0.006	-5.924	0.000
yrs_married	0.0643	0.006	9.941	0.000
children	0.0086	0.019	0.453	0.651
religious	-0.2236	0.021	-10.891	0.000

Average Partial Effect (APE) or Average Marginal Effect (AME)

$$n^{-1} \sum_{i=1}^{n} [g(\hat{\beta}_0 + x_i\hat{\beta})\hat{\beta}_j]$$

AME is the average of the nonlinear function rather than the nonlinear function of the average

$$g[E(x\beta)] \neq E[g(x\beta)]$$

AME = Probit.get_margeff(at='overall')

	dy/dx	std err	z	P>|z|
C(occupation)[T.2.0]	0.0646	0.077	0.843	0.399
C(occupation)[T.3.0]	0.1223	0.075	1.623	0.105
C(occupation)[T.4.0]	0.0806	0.076	1.066	0.286
C(occupation)[T.5.0]	0.1866	0.076	2.444	0.015
C(occupation)[T.6.0]	0.1986	0.085	2.323	0.020
C(occupation_husb)[T.2.0]	0.0289	0.033	0.882	0.378
C(occupation_husb)[T.3.0]	0.0510	0.036	1.420	0.156
C(occupation_husb)[T.4.0]	0.0235	0.032	0.738	0.461
C(occupation_husb)[T.5.0]	0.0281	0.032	0.873	0.383
C(occupation_husb)[T.6.0]	0.0315	0.036	0.878	0.380
educ	-0.0006	0.003	-0.180	0.857
rate_marriage	-0.1302	0.005	-26.444	0.000
age	-0.0110	0.002	-5.966	0.000
yrs_married	0.0197	0.002	10.155	0.000
children	0.0026	0.006	0.453	0.651
religious	-0.0685	0.006	-11.171	0.000

Partial Effect at the Average (PEA)

$$\Delta \hat{P}(y = 1|x) \approx [g(\hat{\beta}_0 + x\hat{\beta})\hat{\beta}_j]\Delta x_j$$

$$g(\hat{\beta}_0 + \bar{x}\hat{\beta}) = g(\hat{\beta}_0 + \hat{\beta}_1\bar{x}_1 + \hat{\beta}_2\bar{x}_2 + ... + \hat{\beta}_k\bar{x}_k)$$

Partial Effect of x_j for the "average" person in the sample.

PEA = Probit.get_margeff(at='mean')

	dy/dx	std err	z	P>|z|
C(occupation)[T.2.0]	0.0732	0.087	0.843	0.399
C(occupation)[T.3.0]	0.1388	0.086	1.622	0.105
C(occupation)[T.4.0]	0.0914	0.086	1.066	0.286
C(occupation)[T.5.0]	0.2117	0.087	2.442	0.015
C(occupation)[T.6.0]	0.2252	0.097	2.321	0.020
C(occupation_husb)[T.2.0]	0.0328	0.037	0.882	0.378
C(occupation_husb)[T.3.0]	0.0578	0.041	1.420	0.156
C(occupation_husb)[T.4.0]	0.0266	0.036	0.738	0.461
C(occupation_husb)[T.5.0]	0.0318	0.036	0.872	0.383
C(occupation_husb)[T.6.0]	0.0358	0.041	0.877	0.380
educ	-0.0006	0.004	-0.180	0.857
rate_marriage	-0.1477	0.006	-23.086	0.000
age	-0.0125	0.002	-5.928	0.000
yrs_married	0.0223	0.002	9.954	0.000
children	0.0030	0.007	0.453	0.651
religious	-0.0777	0.007	-10.931	0.000

Probit.pred_table()

Classified/True	y_0	y_1
$\hat{y}_0$	3890	423
$\hat{y}_1$	1322	731

$$\% \text{ correctly predicted} =$$

$$\frac{3890+731}{3890+731+423+1322} = 72.6\%$$

respondent11 = dta.iloc[[10]]

print(respondent11)

```
rate_marriage   age  yrs_married  children  religious  educ  occupation
          2.0  27.0          6.0       2.0        1.0  16.0         3.0

occupation_husb   affairs  affair
            5.0  3.266665     1.0
```

Probit.predict()[0:11]

```
array([0.33452118, 0.73748271, 0.38220025, 0.44394865, 0.31242976,
       0.38356882, 0.53224527, 0.40988183, 0.49357343, 0.56782298,
       0.73011362])
```