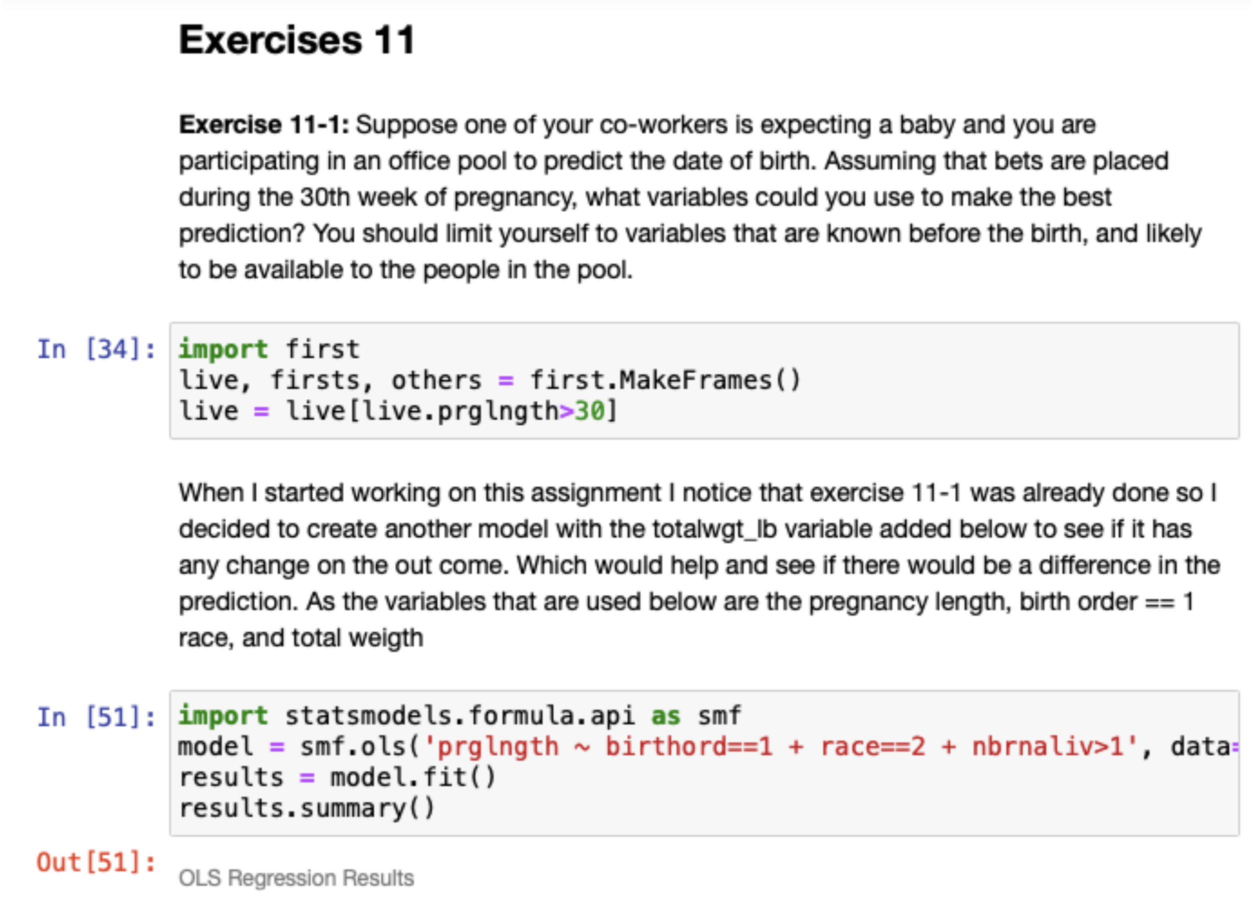
Theodore Koby-Hercsky
DSC530-T302

8.2 Exercises



Exercises 11

Exercise 11-1: Suppose one of your co-workers is expecting a baby and you are participating in an office pool to predict the date of birth. Assuming that bets are placed during the 30th week of pregnancy, what variables could you use to make the best prediction? You should limit yourself to variables that are known before the birth, and likely to be available to the people in the pool.

In [34]:
```
import first
live, firsts, others = first.MakeFrames()
live = live[live.prglngth>30]
```

When I started working on this assignment I notice that exercise 11-1 was already done so I decided to create another model with the totalwgt_lb variable added below to see if it has any change on the out come. Which would help and see if there would be a difference in the prediction. As the variables that are used below are the pregnancy length, birth order == 1 race, and total weigth

In [51]:
```
import statsmodels.formula.api as smf
model = smf.ols('prglngth ~ birthord==1 + race==2 + nbrnaliv>1', data=
results = model.fit()
results.summary()
```

Out[51]:

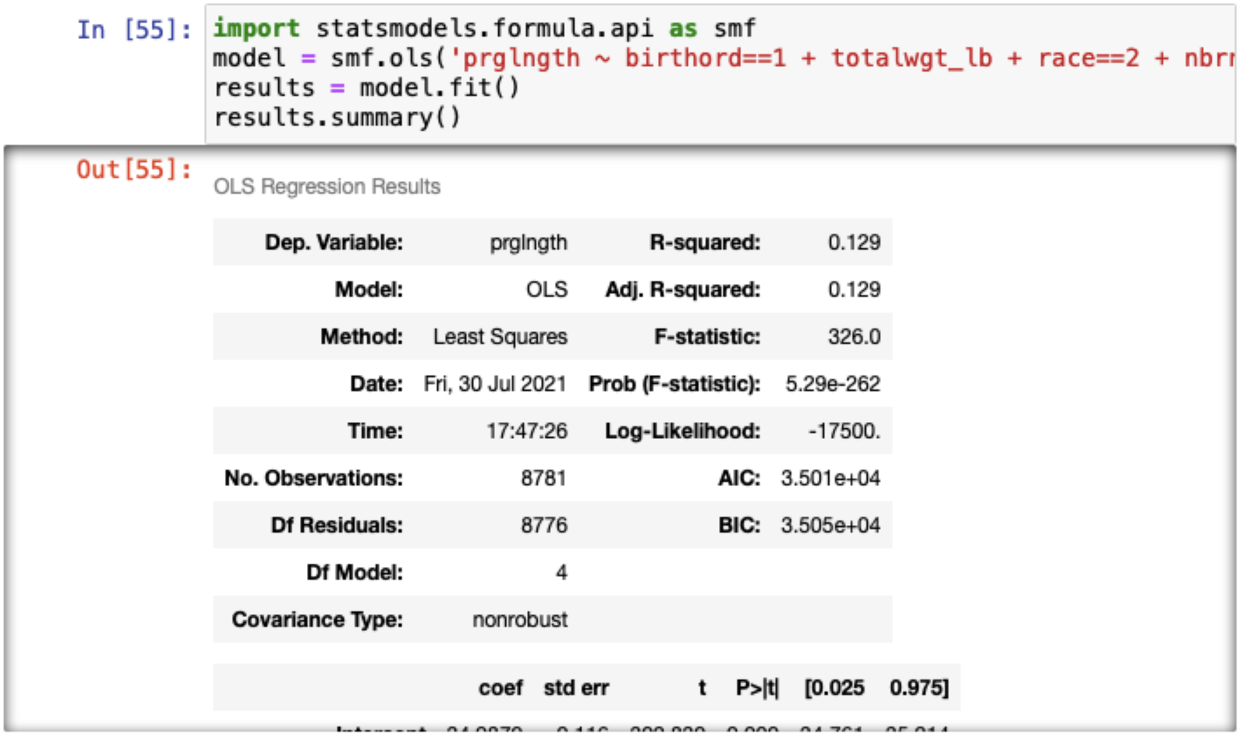
OLS Regression Results

Dep. Variable:	prglngth	**R-squared:**	0.011
Model:	OLS	**Adj. R-squared:**	0.011
Method:	Least Squares	**F-statistic:**	34.28
Date:	Fri, 30 Jul 2021	**Prob (F-statistic):**	5.09e-22
Time:	17:16:17	**Log-Likelihood:**	-18247.
No. Observations:	8884	**AIC:**	3.650e+04
Df Residuals:	8880	**BIC:**	3.653e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	38.7617	0.039	1006.410	0.000	38.686	38.837
birthord == 1[T.True]	0.1015	0.040	2.528	0.011	0.023	0.180
race == 2[T.True]	0.1390	0.042	3.311	0.001	0.057	0.221
nbrnaliv > 1[T.True]	-1.4944	0.164	-9.086	0.000	-1.817	-1.172

Omnibus:	1587.470	**Durbin-Watson:**	1.619
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	6160.751
Skew:	-0.852	**Prob(JB):**	0.00
Kurtosis:	6.707	**Cond. No.**	10.9

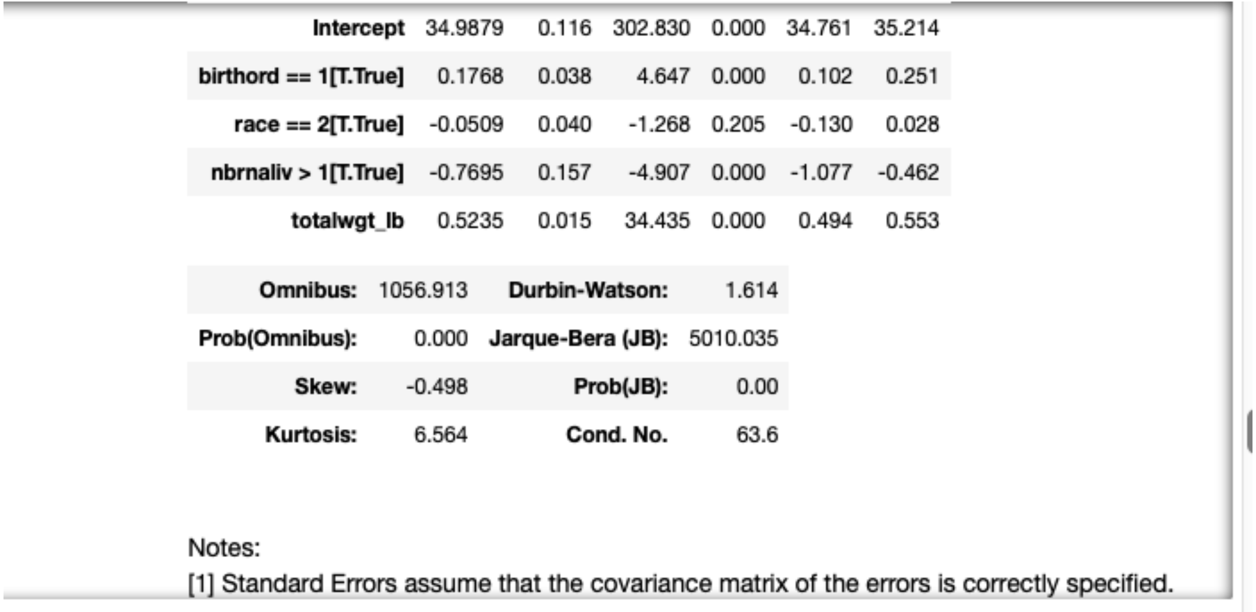
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



```
In [55]: import statsmodels.formula.api as smf
         model = smf.ols('prglngth ~ birthord==1 + totalwgt_lb + race==2 + nbr
         results = model.fit()
         results.summary()
```

```
Out[55]:
OLS Regression Results

       Dep. Variable:        prglngth          R-squared:      0.129
              Model:             OLS      Adj. R-squared:      0.129
             Method:   Least Squares         F-statistic:      326.0
               Date: Fri, 30 Jul 2021  Prob (F-statistic):  5.29e-262
               Time:         17:47:26     Log-Likelihood:     -17500.
   No. Observations:            8781                 AIC:  3.501e+04
       Df Residuals:            8776                 BIC:  3.505e+04
           Df Model:               4
    Covariance Type:       nonrobust
```



```
                         coef  std err        t   P>|t|  [0.025  0.975]
            Intercept 34.9879    0.116  302.830   0.000  34.761  35.214
  birthord == 1[T.True] 0.1768   0.038    4.647   0.000   0.102   0.251
     race == 2[T.True] -0.0509   0.040   -1.268   0.205  -0.130   0.028
  nbrnaliv > 1[T.True] -0.7695   0.157   -4.907   0.000  -1.077  -0.462
          totalwgt_lb  0.5235    0.015   34.435   0.000   0.494   0.553

        Omnibus: 1056.913     Durbin-Watson:    1.614
  Prob(Omnibus):    0.000  Jarque-Bera (JB): 5010.035
           Skew:   -0.498          Prob(JB):     0.00
       Kurtosis:    6.564          Cond. No.     63.6

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

As seen from the two models above from the one that does not include the weight variable we see an intercept of 38.7617 and from the model with the weight variable we get an intercept of 34.9879. So I did some research and found out that a premature baby is delivered before 37 weeks and the average pregnancy last 40 weeks. So from this data and the data that was collected from the models I would state that the baby will be born at the 39th week if we went without the weight variable and at the 35th week if we went with the weight variable. But for me overall I would assume it would be around 36.87 being that is waht the average of the two intercpts would be which would round it up to 37 weeks making it a normal pregnancy.

Exercise 11-3: If the quantity you want to predict is a count, you can use Poisson regression, which is implemented in StatsModels with a function called `poisson`. It works the same way as `ols` and `logit`. As an exercise, let's use it to predict how many children a woman has born; in the NSFG dataset, this variable is called `numbabes`.

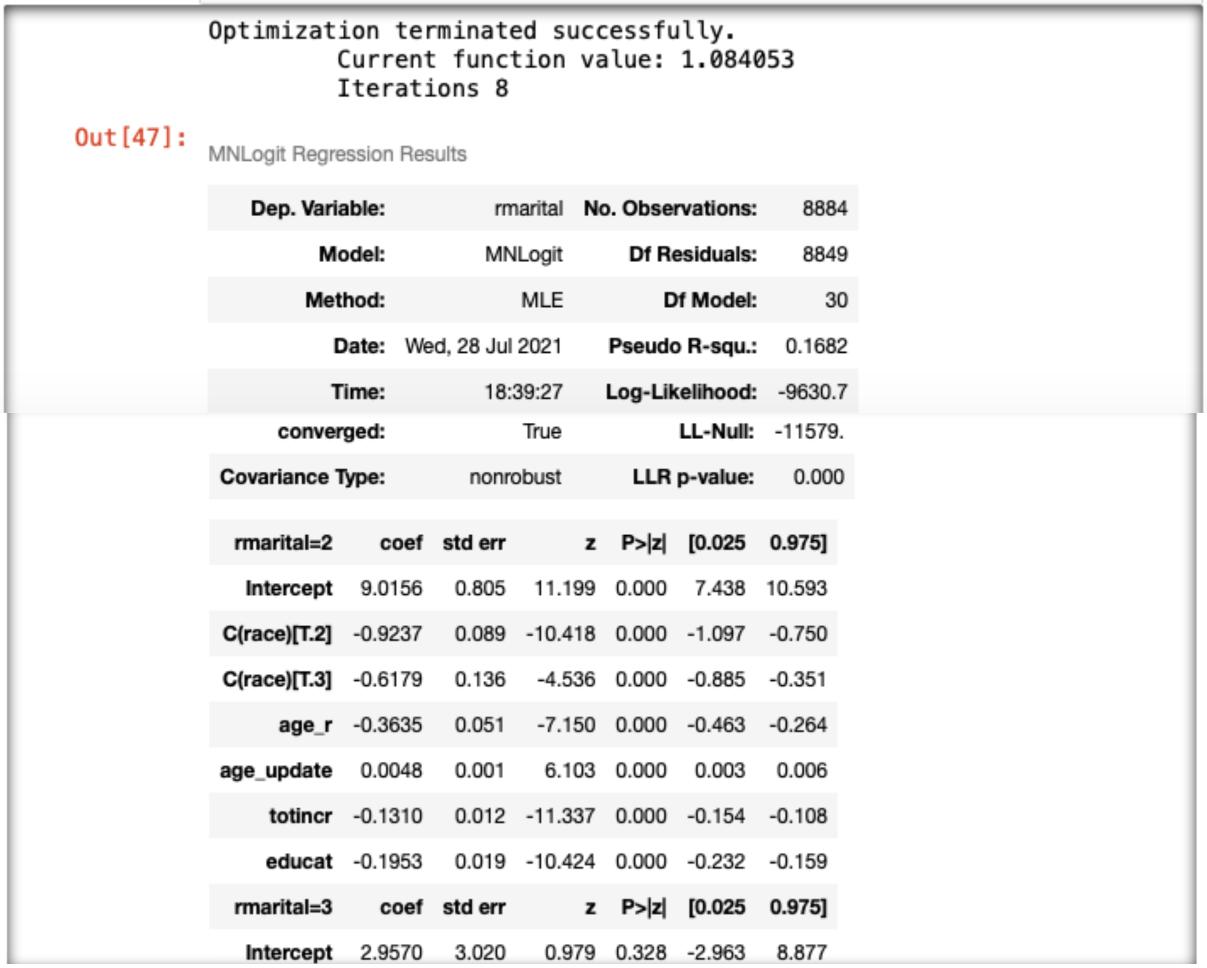
Suppose you meet a woman who is 35 years old, black, and a college graduate whose annual household income exceeds $75,000. How many children would you predict she has born?

In [38]:

```
# First I will be using a nonlinear of the age to create an updated ag
# and joining it by multiplying the age_r by 2.
join.numbabes.replace([95], np.nan, inplace=True)
join['age_update'] = join.age_r**2
```

In [47]:

```
# Next update the stats to help predict the amount of children a womar
# Then use poisson with the formula and set the data equal to join.
stats='numbabes ~ age_r + age_update + C(race) + totincr + educat'
mod = smf.poisson(stats, data=join)
fit_mod = model.fit()
fit_mod.summary()
```



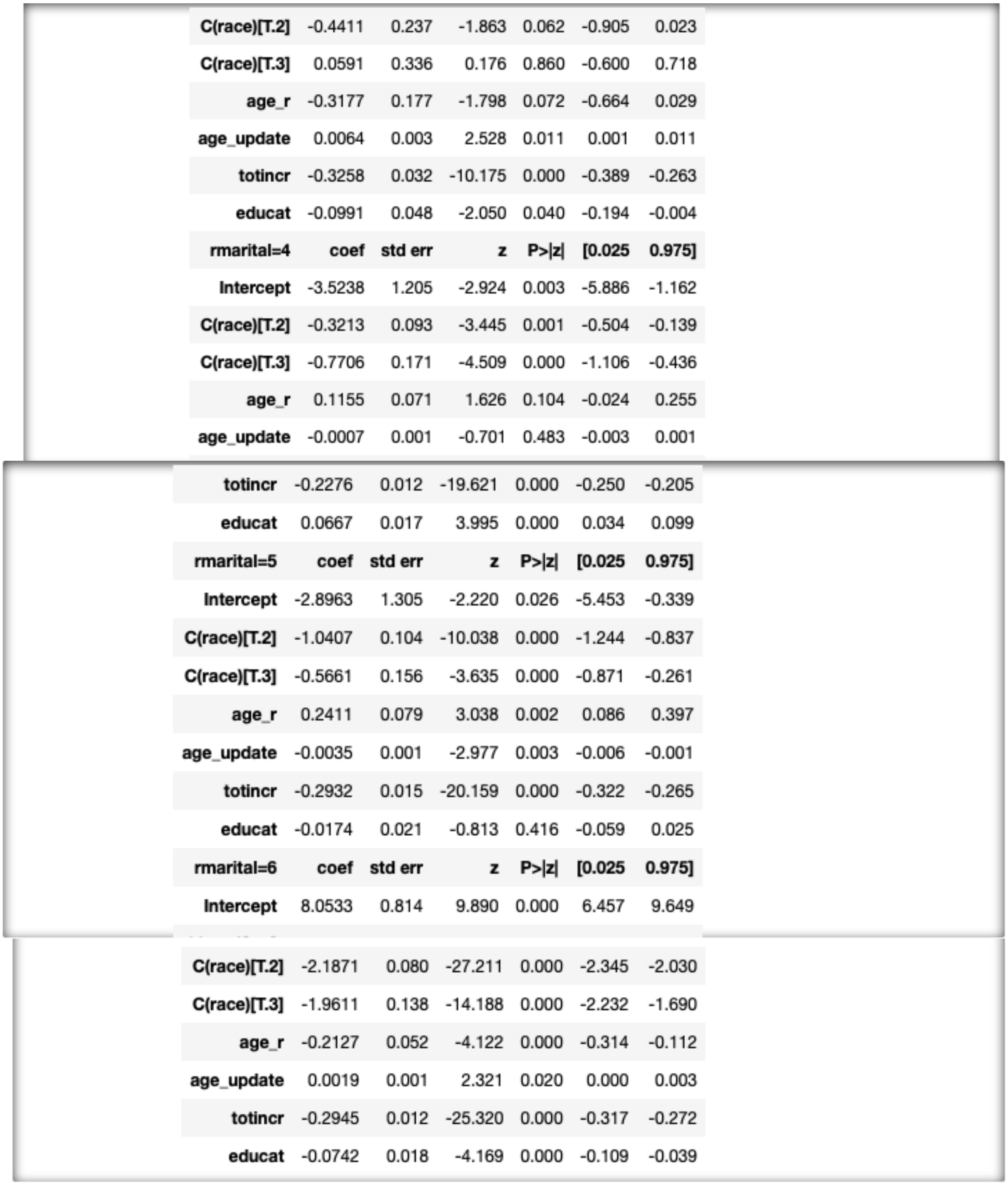
```
Optimization terminated successfully.
         Current function value: 1.084053
         Iterations 8
```

Out[47]:

MNLogit Regression Results

Dep. Variable:	rmarital	**No. Observations:**	8884
Model:	MNLogit	**Df Residuals:**	8849
Method:	MLE	**Df Model:**	30
Date:	Wed, 28 Jul 2021	**Pseudo R-squ.:**	0.1682
Time:	18:39:27	**Log-Likelihood:**	-9630.7
converged:	True	**LL-Null:**	-11579.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

rmarital=2	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	9.0156	0.805	11.199	0.000	7.438	10.593
C(race)[T.2]	-0.9237	0.089	-10.418	0.000	-1.097	-0.750
C(race)[T.3]	-0.6179	0.136	-4.536	0.000	-0.885	-0.351
age_r	-0.3635	0.051	-7.150	0.000	-0.463	-0.264
age_update	0.0048	0.001	6.103	0.000	0.003	0.006
totincr	-0.1310	0.012	-11.337	0.000	-0.154	-0.108
educat	-0.1953	0.019	-10.424	0.000	-0.232	-0.159
rmarital=3	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.9570	3.020	0.979	0.328	-2.963	8.877



C(race)[T.2]	-0.4411	0.237	-1.863	0.062	-0.905	0.023
C(race)[T.3]	0.0591	0.336	0.176	0.860	-0.600	0.718
age_r	-0.3177	0.177	-1.798	0.072	-0.664	0.029
age_update	0.0064	0.003	2.528	0.011	0.001	0.011
totincr	-0.3258	0.032	-10.175	0.000	-0.389	-0.263
educat	-0.0991	0.048	-2.050	0.040	-0.194	-0.004
rmarital=4	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-3.5238	1.205	-2.924	0.003	-5.886	-1.162
C(race)[T.2]	-0.3213	0.093	-3.445	0.001	-0.504	-0.139
C(race)[T.3]	-0.7706	0.171	-4.509	0.000	-1.106	-0.436
age_r	0.1155	0.071	1.626	0.104	-0.024	0.255
age_update	-0.0007	0.001	-0.701	0.483	-0.003	0.001
totincr	-0.2276	0.012	-19.621	0.000	-0.250	-0.205
educat	0.0667	0.017	3.995	0.000	0.034	0.099
rmarital=5	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-2.8963	1.305	-2.220	0.026	-5.453	-0.339
C(race)[T.2]	-1.0407	0.104	-10.038	0.000	-1.244	-0.837
C(race)[T.3]	-0.5661	0.156	-3.635	0.000	-0.871	-0.261
age_r	0.2411	0.079	3.038	0.002	0.086	0.397
age_update	-0.0035	0.001	-2.977	0.003	-0.006	-0.001
totincr	-0.2932	0.015	-20.159	0.000	-0.322	-0.265
educat	-0.0174	0.021	-0.813	0.416	-0.059	0.025
rmarital=6	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	8.0533	0.814	9.890	0.000	6.457	9.649
C(race)[T.2]	-2.1871	0.080	-27.211	0.000	-2.345	-2.030
C(race)[T.3]	-1.9611	0.138	-14.188	0.000	-2.232	-1.690
age_r	-0.2127	0.052	-4.122	0.000	-0.314	-0.112
age_update	0.0019	0.001	2.321	0.020	0.000	0.003
totincr	-0.2945	0.012	-25.320	0.000	-0.317	-0.272
educat	-0.0742	0.018	-4.169	0.000	-0.109	-0.039

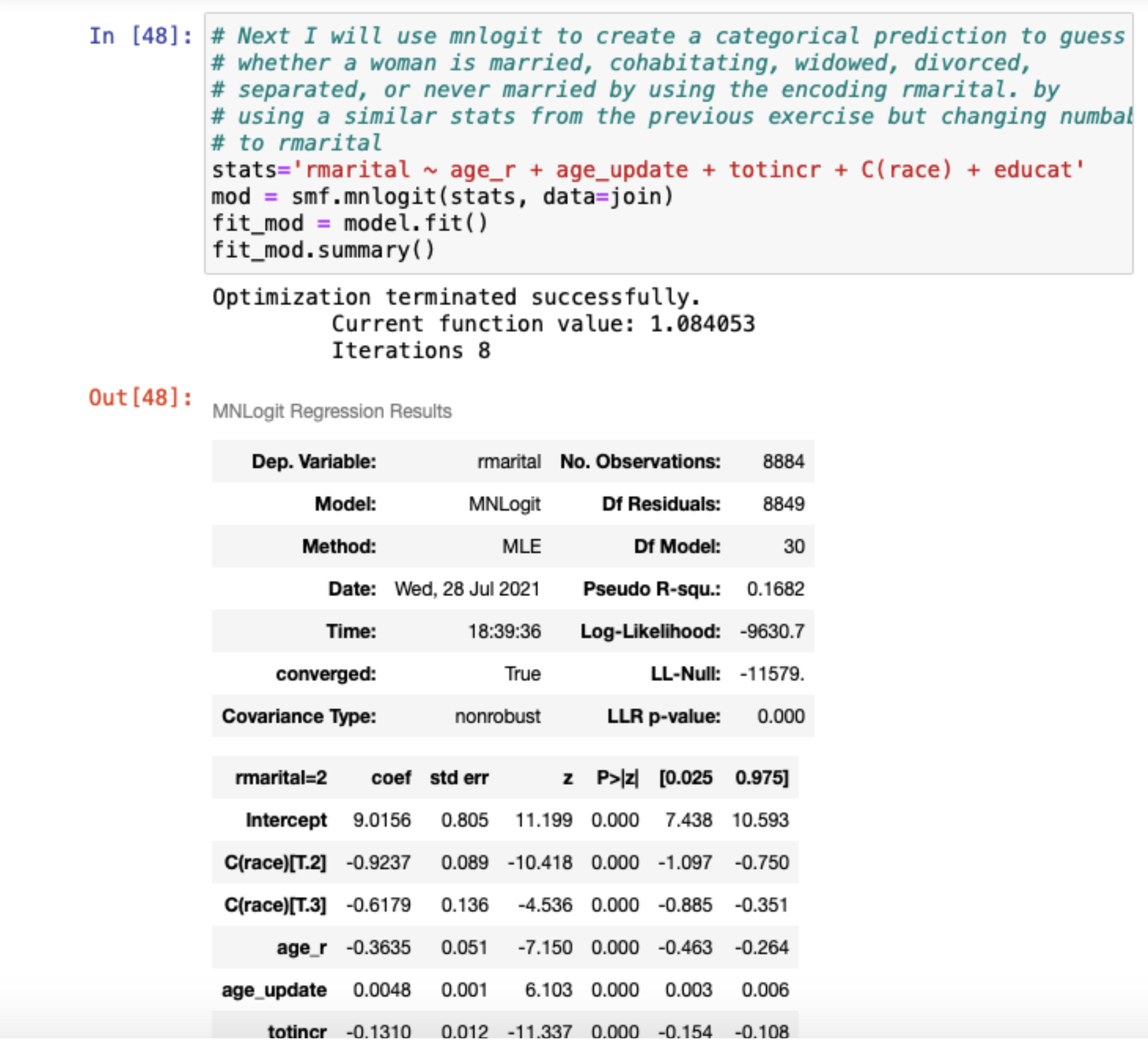
Now we can predict the number of children for a woman who is 35 years old, black, and a college graduate whose annual household income exceeds $75,000

```
In [43]: # Next I will predict the number of children for a woman who is 35
         # years old, black, and a college graduate whose annual household
         # income exceeds $75,000 by setting up a table with the data needed
         # for the age and updated age then the education level, race of indivi
         # and the total increase
         headers = ['age_r', 'age_update', 'educat', 'totincr', 'race']
         updated_data = pd.DataFrame([[35, 35**2, 16, 14, 1]], columns=headers)
         results.predict(updated_data)
         # As we see that the number of children that fall into this statistic
         # of a woman who is 35 years old, black, and a college graduate whose
         # annual household income exceeds $75,000 would be 2.496802 children
         # so between two to three
```

```
Out[43]: 0    2.496802
         dtype: float64
```

Exercise 11-4: If the quantity you want to predict is categorical, you can use multinomial logistic regression, which is implemented in StatsModels with a function called `mnlogit`. As an exercise, let's use it to guess whether a woman is married, cohabitating, widowed, divorced, separated, or never married; in the NSFG dataset, marital status is encoded in a variable called `rmarital`.

Suppose you meet 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?



In [48]:

```
# Next I will use mnlogit to create a categorical prediction to guess
# whether a woman is married, cohabitating, widowed, divorced,
# separated, or never married by using the encoding rmarital. by
# using a similar stats from the previous exercise but changing numbal
# to rmarital
stats='rmarital ~ age_r + age_update + totincr + C(race) + educat'
mod = smf.mnlogit(stats, data=join)
fit_mod = model.fit()
fit_mod.summary()
```

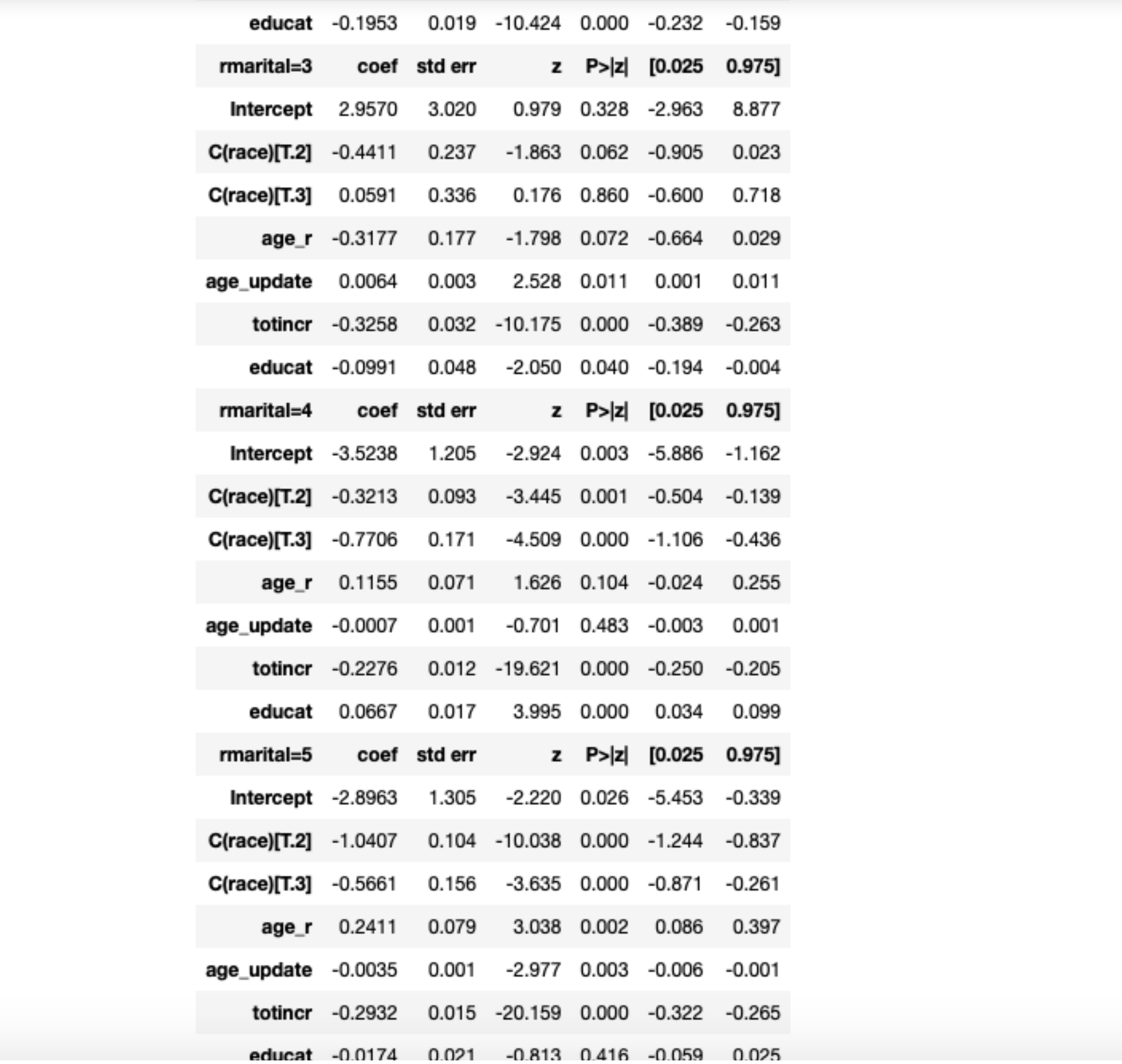
```
Optimization terminated successfully.
         Current function value: 1.084053
         Iterations 8
```

Out[48]:

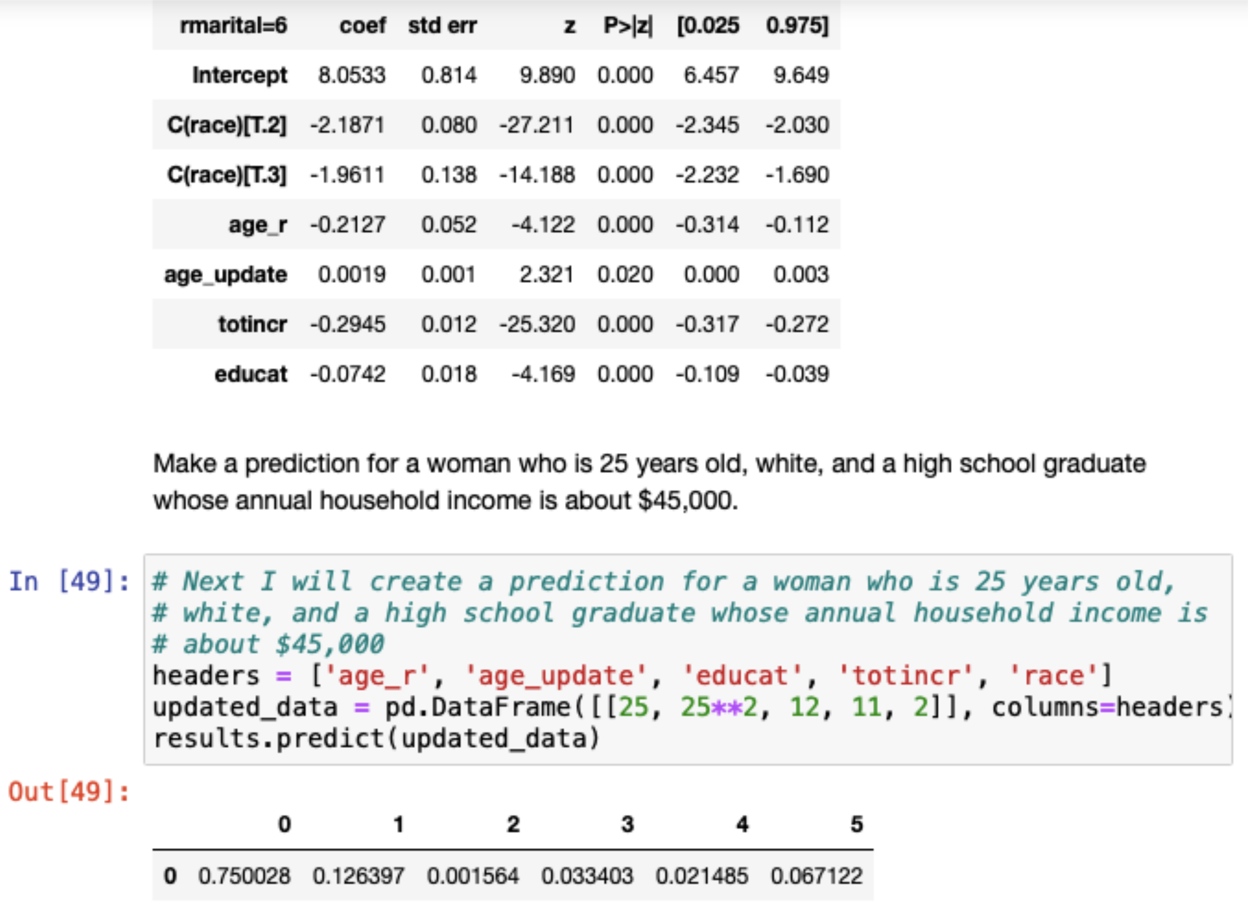
MNLogit Regression Results

Dep. Variable:	rmarital	**No. Observations:**	8884
Model:	MNLogit	**Df Residuals:**	8849
Method:	MLE	**Df Model:**	30
Date:	Wed, 28 Jul 2021	**Pseudo R-squ.:**	0.1682
Time:	18:39:36	**Log-Likelihood:**	-9630.7
converged:	True	**LL-Null:**	-11579.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

rmarital=2	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	9.0156	0.805	11.199	0.000	7.438	10.593
C(race)[T.2]	-0.9237	0.089	-10.418	0.000	-1.097	-0.750
C(race)[T.3]	-0.6179	0.136	-4.536	0.000	-0.885	-0.351
age_r	-0.3635	0.051	-7.150	0.000	-0.463	-0.264
age_update	0.0048	0.001	6.103	0.000	0.003	0.006
totincr	-0.1310	0.012	-11.337	0.000	-0.154	-0.108



educat	-0.1953	0.019	-10.424	0.000	-0.232	-0.159
rmarital=3	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.9570	3.020	0.979	0.328	-2.963	8.877
C(race)[T.2]	-0.4411	0.237	-1.863	0.062	-0.905	0.023
C(race)[T.3]	0.0591	0.336	0.176	0.860	-0.600	0.718
age_r	-0.3177	0.177	-1.798	0.072	-0.664	0.029
age_update	0.0064	0.003	2.528	0.011	0.001	0.011
totincr	-0.3258	0.032	-10.175	0.000	-0.389	-0.263
educat	-0.0991	0.048	-2.050	0.040	-0.194	-0.004
rmarital=4	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-3.5238	1.205	-2.924	0.003	-5.886	-1.162
C(race)[T.2]	-0.3213	0.093	-3.445	0.001	-0.504	-0.139
C(race)[T.3]	-0.7706	0.171	-4.509	0.000	-1.106	-0.436
age_r	0.1155	0.071	1.626	0.104	-0.024	0.255
age_update	-0.0007	0.001	-0.701	0.483	-0.003	0.001
totincr	-0.2276	0.012	-19.621	0.000	-0.250	-0.205
educat	0.0667	0.017	3.995	0.000	0.034	0.099
rmarital=5	**coef**	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-2.8963	1.305	-2.220	0.026	-5.453	-0.339
C(race)[T.2]	-1.0407	0.104	-10.038	0.000	-1.244	-0.837
C(race)[T.3]	-0.5661	0.156	-3.635	0.000	-0.871	-0.261
age_r	0.2411	0.079	3.038	0.002	0.086	0.397
age_update	-0.0035	0.001	-2.977	0.003	-0.006	-0.001
totincr	-0.2932	0.015	-20.159	0.000	-0.322	-0.265
educat	-0.0174	0.021	-0.813	0.416	-0.059	0.025



rmarital=6	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	8.0533	0.814	9.890	0.000	6.457	9.649
C(race)[T.2]	-2.1871	0.080	-27.211	0.000	-2.345	-2.030
C(race)[T.3]	-1.9611	0.138	-14.188	0.000	-2.232	-1.690
age_r	-0.2127	0.052	-4.122	0.000	-0.314	-0.112
age_update	0.0019	0.001	2.321	0.020	0.000	0.003
totincr	-0.2945	0.012	-25.320	0.000	-0.317	-0.272
educat	-0.0742	0.018	-4.169	0.000	-0.109	-0.039

Make a prediction for a woman who is 25 years old, white, and a high school graduate whose annual household income is about $45,000.

In [49]:

```
# Next I will create a prediction for a woman who is 25 years old,
# white, and a high school graduate whose annual household income is
# about $45,000
headers = ['age_r', 'age_update', 'educat', 'totincr', 'race']
updated_data = pd.DataFrame([[25, 25**2, 12, 11, 2]], columns=headers)
results.predict(updated_data)
```

Out[49]:

	0	1	2	3	4	5
0	0.750028	0.126397	0.001564	0.033403	0.021485	0.067122