Lab 1 (QMSS5018 Advanced analytic techniques)

Question: Option 1
Submitted by: Gideon Tay
My UNI: gt2528

Overview: In this lab, we will explore whether individuals with higher levels of education tend to support greater government spending on scientific research.

Q1. Run a multiple multinomial logistic regression. The outcome can be truly unordered or simply ordinal. Tell me how you think your independent variables will be related to your dependent variable. Interpret your results. Compare coefficients on your X variable of interest (not all of them) across different cuts of the multinomial outcomes, as we did in class (i.e., the Z test). For extra credit, generate some predicted probabilities. Tell me what you learned about your hypothesized relationship(s) from this exercise.

Import all necessary libraries for this lab

In [1]:

```python
# Libraries for data analysis
import pandas as pd # also used to load data but primarily used in analysis
import numpy as np
from scipy.stats import norm, chi2
import statsmodels.api as sm
from statsmodels.miscmodels.ordinal_model import OrderedModel  # Ordinal logistic regression

# Libraries to load in data
import requests
import zipfile
import io
from tqdm.notebook import tqdm
```

Load in and clean 2006 General Social Survey (GSS) data

We load in the GSS 2006 data directly from the website (codebook here). We loaded in the following variables: 'id', 'year', 'educ', 'natsci'. For 'educ' and 'natsci', we load in both the numeric value labels and the categorical names.

We preface the categorical name columns with a 'z'. Our final data frame thus contains 6 columns: 'id', 'year', 'educ', 'natsci', 'zeduc', 'znatsci'.

In [2]:

```python
# Step 1: Download the ZIP file with progress bar
url = 'https://gss.norc.org/content/dam/gss/get-the-data/documents/stata/2006_stata.zip'

# Make a streaming request to get the content in chunks
response = requests.get(url, stream=True)
total_size = int(response.headers.get('content-length', 0))  # Get the total file size
block_size = 1024  # 1 Kilobyte

# Progress bar for downloading
tqdm_bar = tqdm(total=total_size, unit='iB', unit_scale=True)
content = io.BytesIO()

# Download the file in chunks with progress bar
for data in response.iter_content(block_size):
    tqdm_bar.update(len(data))
    content.write(data)

tqdm_bar.close()

# Check if the download is successful
if total_size != 0 and tqdm_bar.n != total_size:
    print("Error in downloading the file.")
else:
    print("Download completed!")

# Step 2: Extract the ZIP file in memory and display progress
with zipfile.ZipFile(content) as z:
    # List all files in the zip
    file_list = z.namelist()

    # Filter for the .dta file (assuming there is only one)
    stata_files = [file for file in file_list if file.endswith('.dta')]

    # If there is a Stata file, proceed to extract and read it
    if stata_files:
        stata_file = stata_files[0]  # Take the first .dta file

        # Step 3: Define and load numeric values of selected columns
        with z.open(stata_file) as stata_file_stream:
            columns_to_load = ['id', 'year', 'educ', 'natsci']
            df_numeric = pd.read_stata(
                stata_file_stream,
                columns=columns_to_load,
                convert_categoricals=False)
            print("Data with numeric labels loaded successfully!")

        # Step 4: Load categorical values of selected columns and add 'z' prefix
        with z.open(stata_file) as stata_file_stream:
            df_categorical = pd.read_stata(stata_file_stream, columns=columns_to_load)
            df_categorical = df_categorical.rename(
                columns={col: f'z{col}' for col in df_categorical.columns})
            print("Categorical columns renamed with 'z' prefix.")

# Step 5: Concatenate both numeric and categorical dataframes
df = pd.concat([df_numeric, df_categorical], axis=1)

# Step 6: Display the first few rows of the final DataFrame
df.head()
```

```
100% ████████████████████ 1.67M/1.67M [00:00<00:00, 7.95MiB/s]
```

```
Download completed!
Data with numeric labels loaded successfully!
Categorical columns renamed with 'z' prefix.
```

Out[2]:

	id	year	educ	natsci	zid	zyear	zeduc	znatsci
0	1	2006	13.0	2.0	1	2006	1 year of college	about right
1	2	2006	14.0	3.0	2	2006	2 years of college	too much
2	3	2006	9.0	NaN	3	2006	9th grade	NaN
3	4	2006	12.0	1.0	4	2006	12th grade	too little
4	5	2006	14.0	2.0	5	2006	2 years of college	about right

Let's clean the data by dropping missing values in `educ` and `natsci`, our key variables of interest:

In [3]:

```
# Drop missing values in 'educ' and 'natsci' and create a clean copy
df_clean = df.dropna(subset=['educ', 'natsci']).copy()
print(f"Data frame shape before cleaning: {df.shape}")
print(f"Data frame shape after cleaning: {df_clean.shape}")
```

```
Data frame shape before cleaning: (4510, 8)
Data frame shape after cleaning: (2782, 8)
```

We observe that after cleaning, we have 2782 rows in the data frame, compared to 4510 rows before.

Tell me how you think your independent variables will be related to your dependent variable and why

Before explaining how the variables are related, let's first understand them individually: their meaning and descriptive statistics.

The **independent variable** `educ` is the respondent's education. Its values range from 0 to 20, with higher values associated with higher levels of education. 1 unit generally corresponds to 1 year of education (starting from 1st grade). However, there is censoring at very high levels of education, with 8 or more years of college education represented by a constant `educ` =20.

In [4]:

```
# Count summary of educ values and associated category names
educ_summary = (
    df_clean.groupby('educ')
    .agg(
        zeduc=('zeduc', 'first'), # as zeduc is unique to educ, take first occurence
        count=('educ', 'count')) # count how many times each educ value appears
    .reset_index()
    .sort_values(by='educ')
)
educ_summary
```

Out[4]:

	educ	zeduc	count
0	0.0	no formal schooling	11
1	1.0	1st grade	2
2	2.0	2nd grade	14
3	3.0	3rd grade	6
4	4.0	4th grade	5
5	5.0	5th grade	16
6	6.0	6th grade	46
7	7.0	7th grade	18
8	8.0	8th grade	47
9	9.0	9th grade	63
10	10.0	10th grade	93
11	11.0	11th grade	127
12	12.0	12th grade	696
13	13.0	1 year of college	264
14	14.0	2 years of college	399
15	15.0	3 years of college	149
16	16.0	4 years of college	446
17	17.0	5 years of college	111
18	18.0	6 years of college	150
19	19.0	7 years of college	48
20	20.0	8 or more years of college	71

The variable `natsci` is based on the following question:

> We are faced with many problems in this country, none of which can be solved easily or inexpensively. I'm going to name some of these problems, and for each one I'd like you to tell me whether you think we're spending too much money on it, too little money, or about the right amount. Are we spending too much, too little, or about the right amount on (ITEM)? Q. Supporting scientific research (NATSCI)

Lower scores reflect greater support for scientific research spending.

In [5]:

```
# Count summary of natsci values and associated category names
natsci_summary = (
    df_clean.groupby('natsci')
    .agg(
        znatsci=('znatsci', 'first'),
        count=('natsci', 'count'))
    .reset_index()
    .sort_values(by='natsci')
)
natsci_summary
```

Out[5]:

	natsci	znatsci	count
0	1.0	too little	1216
1	2.0	about right	1215
2	3.0	too much	351

I shall recode `natsci` to produce the **dependent variable** `rnatsci`. I will recode it so that higher scores reflect greater support for scientific research spending. This is an ordinal variable:

In [6]:

```python
# Reverse coding 'natsci' and converting to categorical
df_clean['rnatsci'] = (4 - df_clean['natsci']).astype('category')

# Ensure rnatsci is correctly ordered as 1-2-3
df_clean['rnatsci'] = df_clean['rnatsci'].cat.reorder_categories([1, 2, 3], ordered=True)

# Count summary of rnatsci values and associated category names
rnatsci_summary = (
    df_clean.groupby('rnatsci', observed=False)
    .agg(
        znatsci=('znatsci', 'first'),
        count=('rnatsci', 'count'))
    .reset_index()
    .sort_values(by='rnatsci')
)
rnatsci_summary
```

Out[6]:

	rnatsci	znatsci	count
0	1	too much	351
1	2	about right	1215
2	3	too little	1216

Let's view the correlation table of `educ` and `rnatsci`. They are positively correlated (round 0.12):

In [7]:

```python
# Compute correlation table
correlation_table = df_clean[['educ', 'rnatsci']].corr()
correlation_table
```

Out[7]:

	educ	rnatsci
educ	1.000000	0.122232
rnatsci	0.122232	1.000000

Expected relation between independent variable `educ` and and dependent variable `rnatsci`:

I expect a positive relationship: individuals with higher levels of education (higher `educ`) support greater scientific research spending (higher `rnatsci`).

Why: this is because people with more education tend to value knowledge and knowledge generation through research more, and would thus tend to have greater support for scientific research spending compared to those with less education.

Note: this is in line with the positive correlation (around 0.12 observed between the two variables)

Run an OLS regression

Before we run our multinomial logistic regression, let us run an OLS to examine the relationship between a respondent's education `educ` and the reported importance of spending more money on scientific research `rnatsci`.

In [8]:

```python
# 1. OLS Regression (Ignoring Ordinality)
X_ols = sm.add_constant(df_clean[['educ']])  # Add intercept
y_ols = df_clean['rnatsci'].astype(float)  # Convert categorical to numeric for OLS

# Fit the model and display summary
model_ols = sm.OLS(y_ols, X_ols).fit()
```

```
summary_ols = model_ols.summary()
print(summary_ols)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                rnatsci   R-squared:                       0.015
Model:                            OLS   Adj. R-squared:                  0.015
Method:                 Least Squares   F-statistic:                     42.16
Date:                Sat, 15 Feb 2025   Prob (F-statistic):           9.90e-11
Time:                        23:57:42   Log-Likelihood:                -2866.2
No. Observations:                2782   AIC:                             5736.
Df Residuals:                    2780   BIC:                             5748.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.9571      0.056     34.957      0.000       1.847       2.067
educ           0.0263      0.004      6.493      0.000       0.018       0.034
==============================================================================
Omnibus:                      355.583   Durbin-Watson:                   2.001
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              173.531
Skew:                          -0.449   Prob(JB):                     2.08e-38
Kurtosis:                       2.169   Cond. No.                         60.6
==============================================================================

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

Interpret your results.

- For every additional unit (approx. equal to a year) of education, the reported support for greater scientific research spending increases by 0.0263 points, on the scientific research concern scale, which ranges from 1 to 3.

- This result is highly statistically significant ($p < 0.001$).

- Education explains 1.5% of the variance in the support for scientific research spending (R-squared = 0.015).

Why OLS is not an appropriate model

- OLS treats `rnatsci` as a cardinal continuous variable
  - It assumes that the difference in attitudes between saying that scientific research support spending is 'too little' and 'about right' is identical to the difference between saying 'about right' and 'too much'.
  - In reality, `rnatsci` is just an ordinal variable and the gap between 'too little' and 'about right' may be different from the gap between 'about right' and 'too much'.
- OLS assumes `educ` has an identical effect on `rnatsci` across `rnatsci` categories
  - This assumption arises as a single slope is estimated in OLS.
  - It is possible that the magnitude, or even direction, of `educ` 's true effect may differ between moving someone's response from 'about right' to 'too little' and 'about right' to 'too much'.
  - For example, it is possible that higher education levels may increase the likelihood of someone saying that scientific research spending is 'too little' compared to it being 'about right', but the likelihood of someone saying that it is 'about right' rather than 'too much' may not be affected as much by education levels.

Run an ordinal logistic regression

The ordinal logistic regression model accounts for the ordinal nature of the dependent variable. It treats `rnatsci` as an ordinal rather than cardinal variable, which is an improvement from OLS regression.

Behind the scenes, an ordinal logistic regression is a weighted average of (k-1) simple binary logistic regressions, where k is the number of ordinal categories. In our case, k=3.

Let's run an ordinal logistic regression model:

In [9]:

```
# 2. Ordinal Logistic Regression
model_ord = OrderedModel(
    df_clean['rnatsci'],  # Dependent variable
    df_clean[['educ']],  # Independent variable (NO CONSTANT)
    distr='logit')  # Using logistic distribution

# Fit the model and display summary
result_ord = model_ord.fit(method='bfgs')
summary_ord = result_ord.summary()
print(summary_ord)
```

```
Optimization terminated successfully.
         Current function value: 0.978286
         Iterations: 13
         Function evaluations: 15
         Gradient evaluations: 15
                             OrderedModel Results
==============================================================================
Dep. Variable:                rnatsci   Log-Likelihood:                -2721.6
Model:                   OrderedModel   AIC:                             5449.
Method:            Maximum Likelihood   BIC:                             5467.
Date:                Sat, 15 Feb 2025
Time:                        23:57:42
No. Observations:                2782
Df Residuals:                    2779
Df Model:                           1
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
educ           0.0688      0.012      5.956      0.000       0.046       0.091
1/2           -1.0214      0.162     -6.289      0.000      -1.340      -0.703
2/3            0.7926      0.026     30.356      0.000       0.741       0.844
==============================================================================
```

Interpret your results.

- For every additional unit (approx. equal to a year) of education, the log-odds of reporting a higher level of support for scientific research spending increases by 0.0688.
- This result is highly statistically significant (p < 0.001).
- This indicates that more educated individuals are more likely to report higher levels of support for greater scientific research spending.

Possible issue with ordinal logistic regression's appropriateness as a model:

- Ordinary logistic regression assumes parallel slopes, meaning that the shape of each (k-1) binary logistic curve is equivalent.
- In other words, it assumes that the odds are proportional from one category-break to the next. This may or may not be true.

Run a multinomial logistic regression

The multinomial logistic model works by running a set of simultaneous logit regressions, each made in reference to one baseline category.

The model allows us to examine how education affects the choice between different levels of support for scientific research spending, treating each level as a distinct category. We set the reference category as 'about right' (`mlrrnatsci` = 2).

In [10]:

```
# 3. Multinomial Logistic Regression

# Set reference category for 'rnatsci' as 'about right' (2)
df_clean['mlrnatsci'] = df_clean['rnatsci'].cat.reorder_categories([2, 1, 3], ordered=True)

# Define independent and dependent variables
X_mnl = sm.add_constant(df_clean[['educ']])  # Add intercept
y_mnl = df_clean['mlrnatsci']

# Fit the model and display summary
model_mnl = sm.MNLogit(y_mnl, X_mnl).fit()
summary_mnl = model_mnl.summary()
print(summary_mnl)
```

```
Optimization terminated successfully.
         Current function value: 0.975608
         Iterations 6
                          MNLogit Regression Results
==============================================================================
Dep. Variable:              mlrnatsci   No. Observations:                 2782
Model:                        MNLogit   Df Residuals:                     2778
Method:                           MLE   Df Model:                            2
Date:                Sat, 15 Feb 2025   Pseudo R-squ.:                0.009265
Time:                        23:57:42   Log-Likelihood:                -2714.1
converged:                       True   LL-Null:                       -2739.5
Covariance Type:            nonrobust   LLR p-value:                 9.486e-12
===============================================================================
mlrnatsci=1       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
const           0.1088      0.235      0.462      0.644      -0.353       0.570
educ           -0.1042      0.018     -5.816      0.000      -0.139      -0.069
-------------------------------------------------------------------------------
mlrnatsci=3       coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
const          -0.3527      0.185     -1.910      0.056      -0.715       0.009
educ            0.0259      0.013      1.963      0.050    3.6e-05       0.052
===============================================================================
```

Let's get greater clarity on the p-values beyond the 3 decimal places. We wan to know if the p-value for `mlrnatsci` =3 over `mlrnatsci` =2 is more or less than 0.05. We do not know due to rounding:

In [11]:

```
z_scores = [-5.816, 1.963]
p_values = [2 * (1 - norm.cdf(abs(z))) for z in z_scores]
p_values
```

Out[11]:

```
[np.float64(6.027246701734157e-09), np.float64(0.04964617403475802)]
```

Interpret your results.

- For every additional unit (approx. equal to a year) of education, the log odds of choosing `mlrnatsci` =1 (spending is 'too much', indicating low support) over `mlrnatsci` =2 (spending is 'about right') decreases by 0.1042.
  - This result is highly statistically significant ($p < 0.001$).
  - This suggests that more educated individuals are less likely to choose `mlrnatsci` =1 over `mlrnatsci` =2. They are less likely to think that scientific research spending levels are 'too much' rather than 'about right'.
- For every additional unit (approx. equal to a year) of education, the log odds of choosing `mlrnatsci` =3 (spending is 'too little', indicating high support) over `mlrnatsci` =2 (spending is 'about right') increases by 0.0259.
  - This result is statistically significant ($0.01 < p < 0.05$).
  - This suggests that more educated individuals are more likely to choose `mlrnatsci` =3 over `mlrnatsci` =2. They are more likely to think that scientific research spending levels are 'too little' rather than 'about right'.
- Hence, the effect of education on choosing `mlrnatsci` =1 over `mlrnatsci` =2 is larger in magnitude (coefficient of -0.1042 vs 0.0259) and more statistically significant (p-value of 6×10^{-9}. vs 0.0496) than the effect of education on choosing `mlrnatsci` =3 over `mlrnatsci` =2.

Compare coefficients on your X variable of interest (`educ`) across different cuts of the multinomial outcomes

Since we used the middle value `mlrnatsci` =2 as the reference category, we do expect opposite signs across the two logit regressions' coefficients on `educ` .

However, under the parallel slopes assumption used in the ordinal logit regressions, they should have equal magnitude, but they appear quite different.

We conduct a Wald (Chi-squared) test to determine if the two logit coefficients on `educ` in the multinomial logistic regression model are equivalent in absolute terms:

In [12]:

```
def compare_educ_coefficients(model):
    """
    Extracts the coefficients and standard errors for 'educ' from a fitted MNLogit model
    and computes a Wald test statistic to compare the absolute values of two contrasts.
    """
    # Extract model parameters
    params = model.params
    b1 = abs(params.loc['educ', params.columns[0]])  # Absolute value of coefficient for first contrast
    b2 = abs(params.loc['educ', params.columns[1]])  # Absolute value of coefficient for second contrast

    # Extract standard errors
    bse = model.bse
    SE1 = bse.loc['educ', bse.columns[0]]  # Standard error for first contrast
    SE2 = bse.loc['educ', bse.columns[1]]  # Standard error for second contrast

    # Compute Wald test statistic using absolute values of coefficients first
    wald_statistic = (b1 - b2) / np.sqrt(SE1**2 + SE2**2)

    # Compute chi-square test statistic
    wald_chi2_statistic = wald_statistic ** 2

    # Compute p-value from chi-square distribution with 1 degree of freedom
    p_value = 1 - chi2.cdf(wald_chi2_statistic, df=1)

    return {
        "Wald Test Statistic": wald_statistic,
        "Chi-Square Statistic": wald_chi2_statistic,
        "P-Value": p_value
    }

# Compare coefficients on educ from the multinomial logit model
results = compare_educ_coefficients(model_mnl)
results
```

Out[12]:

```
{'Wald Test Statistic': np.float64(3.5167318515797863),
 'Chi-Square Statistic': np.float64(12.367402915915791),
 'P-Value': np.float64(0.0004368949083817242)}
```

Interpret your results.

- There is a highly statistically significant difference between the coefficients in absolute terms (p-value < 0.001)
- We reject the null hypothesis and conclude that the effect of education on the choice of `rnatsci` =1 compared to `rnatsci` =2 is significantly different from the effect of age on the choice of `rnatsci` =3 compared to `rnatsci` =2.
- Hence, the parallel slopes assumption is not a good assumption.

Why multinomial regression is an appropriate model

Compared to OLS: we have explained previously that OLS is not an appropriate model as it...

- Treats `rnatsci` as a cardinal continuous variable when it is in fact ordinal. The gap between `rnatsci` =1 and `rnatsci` =2 may be different from the gap between `rnatsci` =2 and `rnatsci` =3, since they are ordinal categories and not cardinal values.
- Assumes `educ` has an identical effect on `rnatsci` across `rnatsci` categories, when this is not true, as demonstrated in the multinomial logistic regression's results.

Compared to ordinal logistic regression:

- We have shown that the parallel slopes assumption used in ordinal logistic regression is a poor assumption for the relationship between education and support for scientific research spending.
- As such, although our dependent variable `rnatsci` is ordinal, we prefer multinomial logit regression over ordinal logit regression which is based on an invalid assumption.
- We are able to get a richer understanding of the effect of education on support for scientific research spending with a multinomial logistic regression, compared to an ordinal logistic regression.

For extra credit, generate some predicted probabilities. Tell me what you learned about your hypothesized relationship(s) from this exercise.

Using the fitted multinomial logit model, we predict probabilities for each `mlrnatsci` outcome (1, 2, 3) for each individual (row) in the original dataset:

In [13]:

```python
# Generate predicted probabilities for the dataset
predicted_probs = model_mnl.predict(X_mnl)

# Ensure correct column labels match rnatsci categories
category_labels = list(df_clean['mlrnatsci'].cat.categories)  # Get actual labels [2, 1, 3]
predicted_probs.columns = [f'P(mlrnatsci={cat})' for cat in category_labels]  # Rename columns properly

# Merge predicted probabilities back into the original dataset
df_clean = df_clean.reset_index(drop=True)
predicted_probs = predicted_probs.reset_index(drop=True)
df_clean = pd.concat([df_clean, predicted_probs], axis=1)

# Show the first few rows to verify
df_clean[['educ', 'mlrnatsci'] + list(predicted_probs.columns)].head()
```

Out[13]:

	educ	mlrnatsci	P(mlrnatsci=2)	P(mlrnatsci=1)	P(mlrnatsci=3)
0	13.0	2	0.440107	0.126614	0.433279
1	14.0	1	0.440614	0.114215	0.445171
2	12.0	3	0.438871	0.140125	0.421004
3	14.0	2	0.440614	0.114215	0.445171
4	16.0	3	0.439645	0.092524	0.467832

For easier viewing, let us summarize the predicted probabilities for each category of `mlrnatsci` (support for scientific research spending) at all possible values of `educ`, from 0 to 20:

In [14]:

```python
# Create data frame with all values of educ (0 to 20 in steps of 1)
new_data = pd.DataFrame({'educ': range(0, 21, 1)})
new_data['zeduc'] = new_data['educ'].map(df_clean.set_index('educ')['zeduc'].to_dict())  # Map zeduc labels
new_data = sm.add_constant(new_data)  # Add intercept

# Predict probabilities for the new dataset
predicted_probs_new = model_mnl.predict(new_data.drop(columns=['zeduc']))  # Drop 'zeduc' to match model input

# Rename columns correctly in new data
predicted_probs_new.columns = [f'P(mlrnatsci={cat})' for cat in category_labels]

# Merge predicted probabilities with new data and display it
new_data = pd.concat([new_data, predicted_probs_new], axis=1)
new_data
```

Out[14]:

	const	educ	zeduc	P(mlrnatsci=2)	P(mlrnatsci=1)	P(mlrnatsci=3)
0	1.0	0	no formal schooling	0.354889	0.395693	0.249418
1	1.0	1	1st grade	0.366852	0.368552	0.264597
2	1.0	2	2nd grade	0.378012	0.342181	0.279807
3	1.0	3	3rd grade	0.388307	0.316715	0.294977
4	1.0	4	4th grade	0.397691	0.292268	0.310041
5	1.0	5	5th grade	0.406130	0.268933	0.324936
6	1.0	6	6th grade	0.413608	0.246781	0.339611
7	1.0	7	7th grade	0.420121	0.225860	0.354019
8	1.0	8	8th grade	0.425678	0.206200	0.368122
9	1.0	9	9th grade	0.430298	0.187811	0.381892
10	1.0	10	10th grade	0.434011	0.170685	0.395304
11	1.0	11	11th grade	0.436855	0.154801	0.408345
12	1.0	12	12th grade	0.438871	0.140125	0.421004
13	1.0	13	1 year of college	0.440107	0.126614	0.433279
14	1.0	14	2 years of college	0.440614	0.114215	0.445171
15	1.0	15	3 years of college	0.440442	0.102872	0.456686
16	1.0	16	4 years of college	0.439645	0.092524	0.467832
17	1.0	17	5 years of college	0.438272	0.083107	0.478621
18	1.0	18	6 years of college	0.436375	0.074558	0.489066
19	1.0	19	7 years of college	0.434003	0.066815	0.499182
20	1.0	20	8 or more years of college	0.431200	0.059814	0.508986

Interpret your results.

- As education levels increase from 0 to 14 (two years in college), the predicted probability of choosing `mlrnatsci` =2 (the reference category) steadily increases from about 0.355 to 0.441. However, it slowly decreases to 0.431 with additional years of education. These changes in predicted values over education levels are relatively small compared to the changes seen in the probability of choosing `mlrnatsci` =3 or `mlrnatsci` =1.
- As education levels increase, the predicted probability of choosing `mlrnatsci` =1 ('too much' scientific research spending) steadily and monotonically decreases, from about 0.396 to 0.060.
- As education levels increase, the predicted probability of choosing `mlrnatsci` =3 ('too little' scientific research spending) steadily and monotonically increases, from about 0.249 to 0.509.
- This indicates that as people get more education, they are much more likely to fall into the spend-more-on-scientific-research (`mlrnatsci` =3) category and less likely to fall into the spend-less-on-scientific-research (`mlrnatsci` =1) category. The probability of being in the `mlrnatsci` =2 category changes to a smaller extent and the effect appears non-unidirectional.