The Nobel Prize has been among the most prestigious international awards since 1901. Each year, awards are bestowed in chemistry, literature, physics, physiology or medicine, economics, and peace. In addition to the honor, prestige, and substantial prize money, the recipient also gets a gold medal with an image of Alfred Nobel (1833 - 1896), who established the prize.

The Nobel Foundation has made a dataset available of all prize winners from the outset of the awards from 1901 to 2023. The dataset used in this project is from the Nobel Prize API and is available in the `nobel.csv` file in the `data` folder.

In this project, you'll get a chance to explore and answer several questions related to this prizewinning data. And we encourage you then to explore further questions that you're interested in!

```
# Loading in required libraries
import pandas as pd
import seaborn as sns
import numpy as np
```

```
# Load the Nobel Prize data
nobel_df = pd.read_csv('data/nobel.csv')

# Display the first few rows of the dataframe
nobel_df.head()
```

	year	c...	prize	motivation	pri...	lau...	laurea.
0	1901	Chemistry	The Nobel Prize in Chemistry 1901	"in recognition of the extraordinary services ...	1/1	160	Individ
1	1901	Literature	The Nobel Prize in Literature 1901	"in special recognition of his poetic composit...	1/1	569	Individ
2	1901	Medicine	The Nobel Prize in Physiology or Medicine 19...	"for his work on serum therapy, especially its...	1/1	293	Individ
3	1901	Peace	The Nobel Peace Prize 1901	null	1/2	462	Individ
4	1901	Peace	The Nobel Peace Prize 1901	null	1/2	463	Individ

Rows: 5

```
# Create a new column 'decade' in the dataframe
nobel_df['decade'] = (nobel_df['year'] // 10) * 10

# Filter for US-born winners
us_winners = nobel_df[nobel_df['birth_country'] == 'United States of America']

# Count the number of winners per decade
us_winners_per_decade = us_winners.groupby('decade').size()

# Identify the decade with the highest ratio of US-born winners
highest_ratio_decade = us_winners_per_decade.idxmax()

highest_ratio_decade
```

```
2000
```

```
# Group by decade and category, then count the number of female laureates
female_winners = nobel_df[nobel_df['sex'] == 'Female'].groupby(['decade', 'category']).size()

# Group by decade and category, then count the total number of laureates
total_winners = nobel_df.groupby(['decade', 'category']).size()

# Calculate the proportion of female laureates
female_proportion = female_winners / total_winners

# Identify the decade and category with the highest proportion of female laureates
highest_female_proportion = female_proportion.idxmax()

highest_female_proportion
```

```
(2020, 'Literature')
```

```
# Find the first woman to win a Nobel Prize
first_female_winner = nobel_df[nobel_df['sex'] == 'Female'].nsmallest(1, 'year')

# Determine repeat winners (those who have won more than once)
repeat_winners = nobel_df['full_name'].value_counts()
repeat_winners = repeat_winners[repeat_winners > 1]

(first_female_winner, repeat_winners)
```

```
(    year category  ... death_country decade
 19  1903  Physics  ...        France   1900

 [1 rows x 19 columns],
 Comité international de la Croix Rouge (International Committee of the Red Cross)    3
 Linus Carl Pauling                                                                   2
 John Bardeen                                                                         2
 Frederick Sanger                                                                     2
 Marie Curie, née Sklodowska                                                          2
 Office of the United Nations High Commissioner for Refugees (UNHCR)                  2
 Name: full_name, dtype: int64)
```

```python
import pandas as pd


# Extract the top values from 'sex'
top_sex_value = nobel_df['sex'].value_counts().idxmax()

# Extract the top values from 'birth_country'
top_birth_country_value = nobel_df['birth_country'].value_counts().idxmax()

(top_sex_value, top_birth_country_value)
```

```
('Male', 'United States of America')
```

```python
import matplotlib.pyplot as plt

# Create a flag for winners whose birth country is "United States of America"
nobel_df['usa_born_winner'] = nobel_df['birth_country'] == "United States of America"

# Create a decade column
nobel_df['decade'] = (nobel_df['year'] // 10) * 10

# Group by decade and calculate the ratio of USA born winners
usa_winners_ratio = nobel_df.groupby('decade')['usa_born_winner'].mean()

# Plotting
plt.figure(figsize=(10, 6))
usa_winners_ratio.plot(kind='line', marker='o')
plt.title('Ratio of US-born Nobel Prize Winners by Decade')
plt.xlabel('Decade')
plt.ylabel('Ratio of US-born Winners')
plt.grid(True)
plt.show()
```

```python
# Filtering for female winners
female_winners = nobel_df[nobel_df['sex'] == 'Female']

# Group by two columns: 'decade' and 'category'
grouped_female_winners = female_winners.groupby(['decade', 'category']).size()

# Finding the decade and category with the highest female winners
max_female_winners = grouped_female_winners.idxmax()
max_female_count = grouped_female_winners.max()

# Creating a dictionary
max_female_dict = {'year': max_female_winners[0], 'category': max_female_winners[1], 'count': max_female_count}

# Creating a DataFrame for the proportion of female winners by decade and category
female_winners_proportion = female_winners.groupby(['decade', 'category']).size() / nobel_df.groupby(['decade', 'category']).size()
female_winners_proportion = female_winners_proportion.unstack('category').fillna(0)

# Creating a relational line plot with multiple categories
plt.figure(figsize=(14, 8))
for category in female_winners_proportion.columns:
    plt.plot(female_winners_proportion.index, female_winners_proportion[category], marker='o', label=category)

plt.title('Proportion of Female Nobel Prize Winners by Decade and Category')
plt.xlabel('Decade')
plt.ylabel('Proportion of Female Winners')
plt.legend(title='Category')
plt.grid(True)
plt.show()
```

```
# Finding the first woman to win a Nobel Prize and her name
first_female_winner = female_winners[female_winners['year'] == female_winners['year'].min()]

# Displaying the earliest year, corresponding category, and her name
first_female_winner[['year', 'category', 'full_name']]
```

```
# Counting the number of times each winner has won
winner_counts = nobel_df['full_name'].value_counts()

# Selecting those with counts of two or more
repeats = winner_counts[winner_counts >= 2].index.tolist()

(repeats,winner_counts)
```

```
# Analyzing the most commonly awarded gender and birth country
top_gender = nobel_df['sex'].value_counts().idxmax()
top_country = nobel_df['birth_country'].value_counts().idxmax()

# Finding the decade with the highest ratio of US-born winners to total winners
nobel_df['usa_born_winner'] = nobel_df['birth_country'] == 'United States of America'
nobel_df['decade'] = (nobel_df['year'] // 10) * 10
usa_winners_by_decade = nobel_df.groupby('decade')['usa_born_winner'].sum()
total_winners_by_decade = nobel_df.groupby('decade').size()
ratio_usa_winners = usa_winners_by_decade / total_winners_by_decade
max_decade_usa = ratio_usa_winners.idxmax()

# Identifying the decade and category with the highest proportion of female laureates
female_winners['decade'] = (female_winners['year'] // 10) * 10
female_winners_by_decade_category = female_winners.groupby(['decade', 'category']).size()
total_winners_by_decade_category = nobel_df.groupby(['decade', 'category']).size()
proportion_female_winners = female_winners_by_decade_category / total_winners_by_decade_category
max_female_decade_category = proportion_female_winners.idxmax()
max_female_dict = {max_female_decade_category[0]: max_female_decade_category[1]}

# Finding the first woman to receive a Nobel Prize and her category
first_woman = female_winners[female_winners['year'] == female_winners['year'].min()].iloc[0]
first_woman_name = first_woman['full_name']
first_woman_category = first_woman['category']

# Identifying individuals or organizations that have won more than one Nobel Prize
winner_counts = nobel_df['full_name'].value_counts()
repeat_winners = winner_counts[winner_counts > 1]
repeat_list = repeat_winners.index.tolist()

# Storing the answers in the specified variables
top_gender, top_country, max_decade_usa, max_female_dict, first_woman_name, first_woman_category, repeat_list
```