Cesar Vazquez

DS 6339

Data Visualization Project

1. How is use of government federal aid/assistance associated with food insecurity as measured by the USDA index or categories?

1.1 install pywaffle to be able to make waffle graphs

```
pip install pywaffle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pywaffle in /usr/local/lib/python3.10/dist-packages (1.1.0)
Requirement already satisfied: fontawesomefree in /usr/local/lib/python3.10/dist-packages (from pywaffle) (6.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from pywaffle) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (1.4.4)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->pywaffle) (1.
```

1.2 Install necessary libraries

```
from google.colab import files
import pandas as pd
import matplotlib.pyplot as plt
from pywaffle import Waffle
from statsmodels.graphics.mosaicplot import mosaic
import numpy as np
```

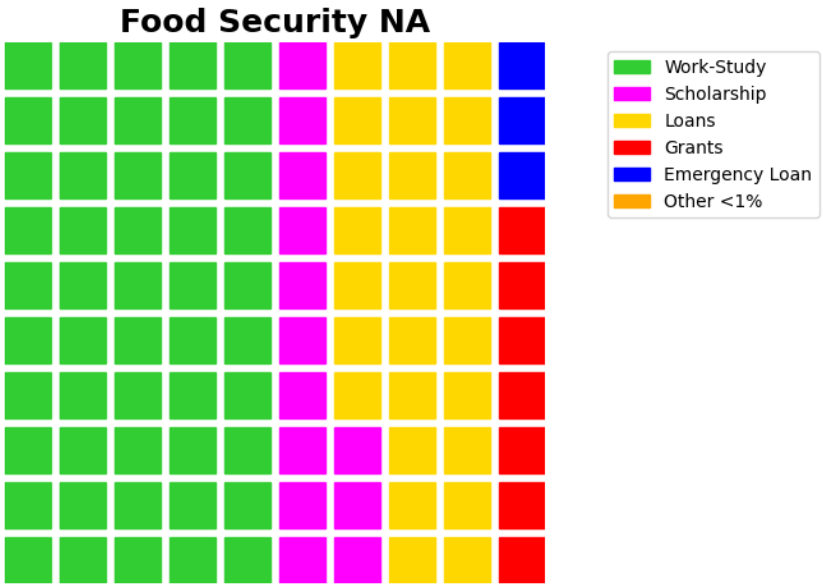
1.3 Upload necessary files

```
df = pd.read_csv('master.csv')
```

1.4 Graph food security 'Not Available' based on the the different types of government funding using a waffle chart where each square represents 1%.

```
data ={'fund': ['Work-Study', 'Scholarship',
                'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
       'stock': [50, 13, 27, 7, 3, 0]
     }

df = pd.DataFrame(data)
default_colors = ['limegreen', 'magenta', 'gold', 'red', 'blue', 'orange']

fig = plt.figure(
    FigureClass = Waffle,
    rows = 10,
    values = df.stock,
    labels = list(df.fund),
    colors=default_colors,
```

```
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
)

plt.title("Food Security NA", fontsize=18, fontweight='bold')
plt.show()
```

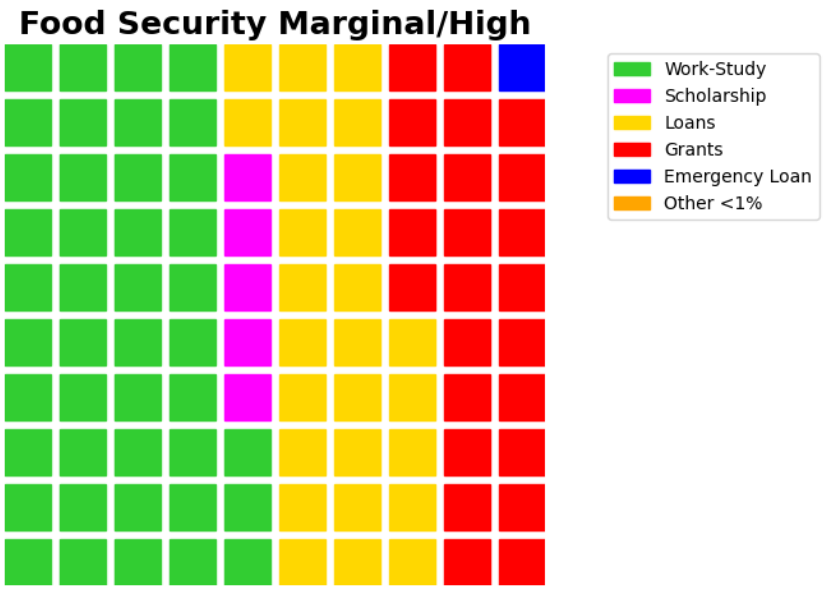


1.5 Graph food security 'Very Low' based on the the different types of government funding using a waffle chart where each square represents 1%.

```
data ={'fund': ['Work-Study', 'Scholarship',
                'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
       'stock': [39, 14, 35, 10, 2, 0]
     }

df = pd.DataFrame(data)
default_colors = ['limegreen', 'magenta', 'gold', 'red', 'blue', 'orange']

fig = plt.figure(
    FigureClass = Waffle,
    rows = 10,
    values = df.stock,
    labels = list(df.fund),
    colors=default_colors,
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
)

plt.title("Food Security Very Low", fontsize=18, fontweight='bold')
plt.show()
```

1.6 Graph food security 'Marginal/High' based on the the different types of government funding using a waffle chart where each square represents 1%.

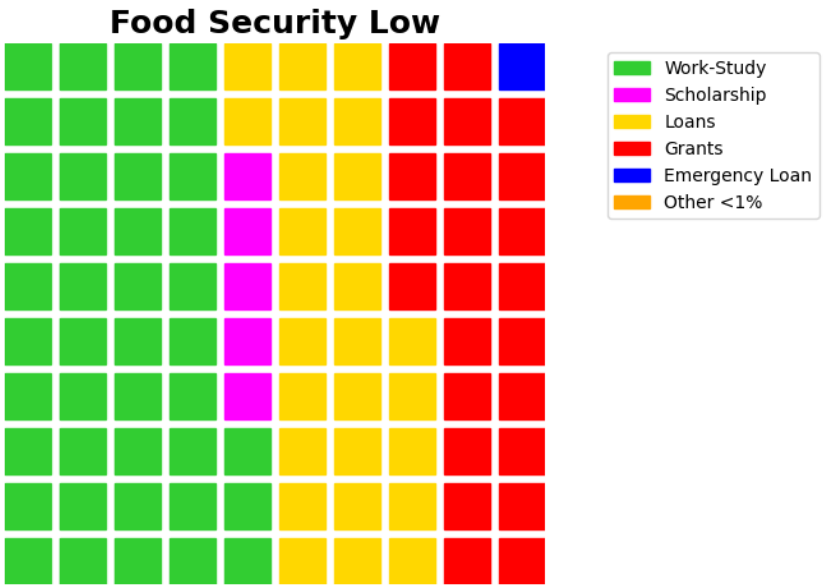
```
data ={'fund': ['Work-Study', 'Scholarship',
                'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
       'stock': [43, 5, 27, 24, 1, 0]
     }

df = pd.DataFrame(data)
default_colors = ['limegreen', 'magenta', 'gold', 'red', 'blue', 'orange']

fig = plt.figure(
    FigureClass = Waffle,
    rows = 10,
    values = df.stock,
    labels = list(df.fund),
    colors=default_colors,
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
)

plt.title("Food Security Marginal/High", fontsize=18, fontweight='bold')
plt.show()
```



1.7 Graph food security 'Low' based on the the different types of government funding using a waffle chart where each square represents 1%.

```
data ={'fund': ['Work-Study', 'Scholarship', 'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
       'stock': [43, 5, 27, 24, 1, 0]
     }

fig = plt.figure(
    FigureClass = Waffle,
    rows = 10,
    values = df.stock,
    labels = list(df.fund),
    colors=default_colors,
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
```

```
)

plt.title("Food Security Low", fontsize=18, fontweight='bold')
plt.show()
```

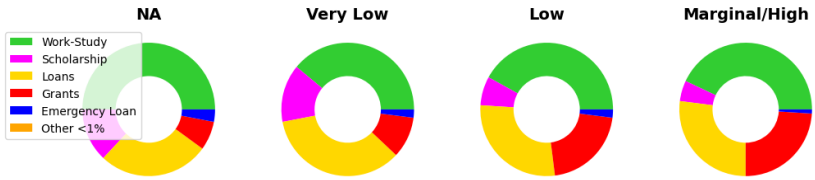


1.8 Another view of the charts, so the proportions are seen better.

```
data1 ={'fund': ['Work-Study', 'Scholarship', 'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
        'stock': [50, 13, 27, 7, 3, 0]
      }

data2 ={'fund': ['Work-Study', 'Scholarship', 'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
        'stock': [39, 14, 35, 10, 2, 0]
      }

data3 ={'fund': ['Work-Study', 'Scholarship', 'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
        'stock': [43, 5, 27, 24, 1, 0]
      }

data4 ={'fund': ['Work-Study', 'Scholarship', 'Loans', 'Grants', 'Emergency Loan', 'Other <1%'],
        'stock': [42, 7, 28, 21, 2, 0]
      }

df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
df3 = pd.DataFrame(data3)
df4 = pd.DataFrame(data4)

default_colors = ['limegreen', 'magenta', 'gold', 'red', 'blue', 'orange']

# Create a figure with 4 subplots
fig, axs = plt.subplots(1, 4, figsize=(12, 4), gridspec_kw={'width_ratios': [1, 1, 1, 1]})

# Plot the first waffle chart on the first subplot
axs[0].set_title("NA", fontsize=14, fontweight='bold')
axs[0].axis('off')
axs[0].pie(df1.stock, colors=default_colors, wedgeprops=dict(width=0.5))
axs[0].legend(df1.fund, loc='upper left', bbox_to_anchor=(-0.4, 1))

# Plot the second waffle chart on the second subplot
axs[1].set_title("Very Low", fontsize=14, fontweight='bold')
axs[1].axis('off')
axs[1].pie(df2.stock, colors=default_colors, wedgeprops=dict(width=0.5))

# Plot the third waffle chart on the third subplot
axs[3].set_title("Marginal/High", fontsize=14, fontweight='bold')
axs[3].axis('off')
axs[3].pie(df3.stock, colors=default_colors, wedgeprops=dict(width=0.5))
```

```
# Plot the fourth waffle chart on the fourth subplot
axs[2].set_title("Low", fontsize=14, fontweight='bold')
axs[2].axis('off')
axs[2].pie(df4.stock, colors=default_colors, wedgeprops=dict(width=0.5))
```

```
([<matplotlib.patches.Wedge at 0x7f1637c40f70>,
  <matplotlib.patches.Wedge at 0x7f16375fec50>,
  <matplotlib.patches.Wedge at 0x7f16375fc9a0>,
  <matplotlib.patches.Wedge at 0x7f16380b98d0>,
  <matplotlib.patches.Wedge at 0x7f16380b8d60>,
  <matplotlib.patches.Wedge at 0x7f16380bbc40>],
 [Text(0.27355891977302554, 1.065441465972024, ''),
  Text(-1.0563230292468058, 0.30689030268624007, ''),
  Text(-0.7530018780839292, -0.8018654323526335, ''),
  Text(0.7778173864806726, -0.7778175321297253, ''),
  Text(1.0978293932846428, -0.06906969842320938, ''),
  Text(1.0999999999999925, -1.287367909193746e-07, '')])
```



1.9 Conclusion

After using a waffle chart for all 4 different securities (Very Low, Low, Moderate/High, NA) we can see some trends worth discussing. First and foremost, the higher the food security the less loans. This does make intuitive sense and higher food security probably means more money for tuition. Second, the second pattern is that the higher the food security, the higher the amounts of grants. This isn't intuitively clear as to why it might be that way, however, that does not mean there is a clear explanation. Lastly, we see that emergency loans are rarely given out, but they are given at much higher rates for those with low or very low food security as opposed to those with moderate or high food security which makes intuitive sense.

2. Does food insecurity (as measured by USDA index or categories) have a relationship with the items pertaining to concentration on school and degree progress/completion?

2.1 Output the number of entries in each of Classification, College Choice, and Food Security

```
# count number of entries for each class
classes = ['Freshman', 'Sophomore', 'Junior', 'Senior', 'Graduate (Masters)', 'Doctoral', 'Professional (Certificate Program)']
for c in classes:
    count = df[df['Classification'] == c]['Classification'].count()
    print(f'{c}: {count}')
print("\n")
```

```
# create dictionary to map numbers to words
college_dict = {
    '1': 'Business',
    '2': 'Education',
    '3': 'Engineering',
    '4': 'Liberal Arts',
    '5': 'Health Sciences',
    '6': 'Nursing',
    '7': 'Science',
    '8': 'Pharmacy',
    '9': 'Other'
}
```

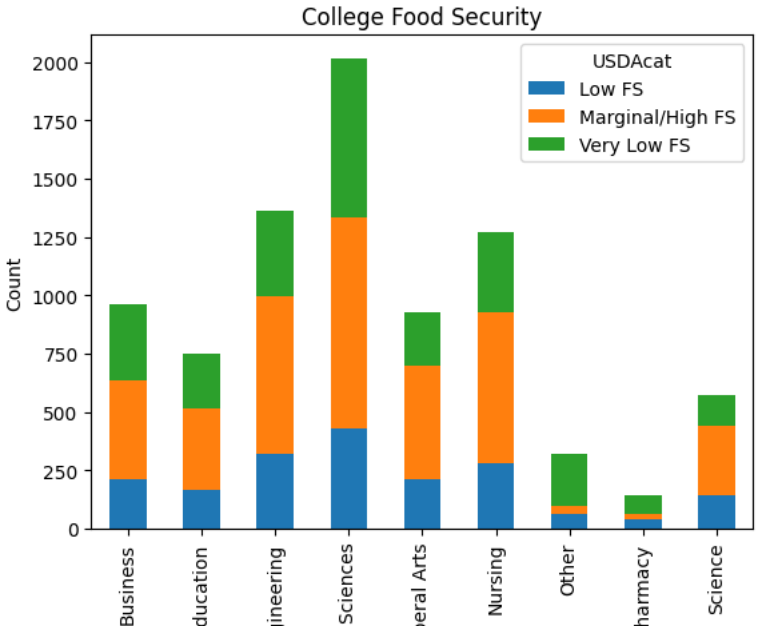
```
# count number of entries for each class and output word instead of number
colleges = ['1', '2', '3', '4', '5', '6', '7', '8', '9']
for n in colleges:
    count = df[df['College'] == n]['College'].count()
    college_word = college_dict[n]
    print(f'{college_word}: {count}')
print("\n")



# count number of entries for each class
foodsecurity = ['Very Low FS', 'Low FS', 'Marginal/High FS']
for c in foodsecurity:
    count = df[df['USDAcat'] == c]['USDAcat'].count()
    print(f'{c}: {count}')
```

```
 Freshman: 1355
 Sophomore: 1363
 Junior: 2311
 Senior: 2877
 Graduate (Masters): 1116
 Doctoral: 507
 Professional (Certificate Program): 26


 Business: 1320
 Education: 1022
 Engineering: 1877
 Liberal Arts: 1114
 Health Sciences: 2278
 Nursing: 1349
 Science: 622
 Pharmacy: 679
 Other: 876


 Very Low FS: 2804
 Low FS: 1920
 Marginal/High FS: 3904
```

2.2 Compare College Choice with Food Security using a Stacked Bar Plot

```
# create dictionary to map numbers to words
college_dict = {
    '1': 'Business',
    '2': 'Education',
    '3': 'Engineering',
    '4': 'Liberal Arts',
    '5': 'Health Sciences',
    '6': 'Nursing',
    '7': 'Science',
    '8': 'Pharmacy',
    '9': 'Other'
}

# create a new column with the college names
df['CollegeName'] = df['College'].map(college_dict)

# create a cross-tabulation table of CollegeName and USDAcat
ct = pd.crosstab(df['CollegeName'], df['USDAcat'])

# display the table
print(ct)



# plot the cross-tabulation table as a stacked bar chart
ct.plot(kind='bar', stacked=True)
plt.title('College Food Security')
plt.xlabel('College Name')
plt.ylabel('Count')
plt.show()
```

USDAcat	Low FS	Marginal/High FS	Very Low FS
CollegeName			
Business	213	425	321
Education	168	349	231
Engineering	320	675	370
Health Sciences	427	904	684
Liberal Arts	209	490	229
Nursing	283	643	345
Other	60	40	219
Pharmacy	38	25	80
Science	141	302	131



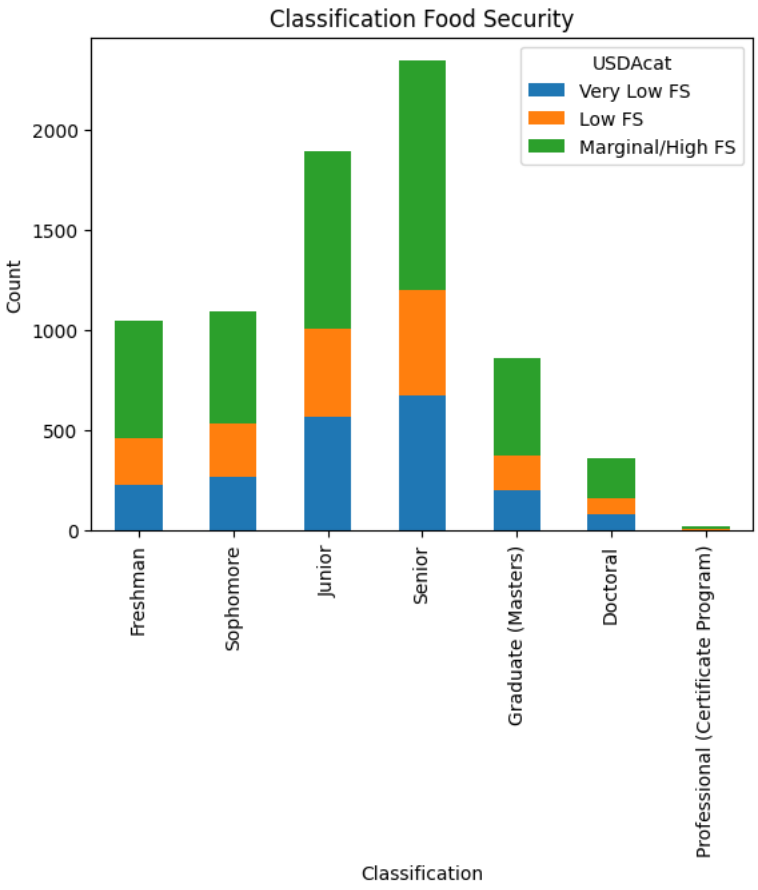
2.3 Observations between College Choice and Food Security

When comparing College Choice with Food Security there is no clear relationship. The first thing to not is that Marginal/High food security make up the majority in almost all bars. The only bars in which Marginal/High food security are not the majority are Other and Pharmacy, so it may be worth analyzing those two responses in themselves, especially because the section that makes up the majority for those is Very Low food security. I could make a conclusion about people choosing pharmacy because it is a safe and stable job market, or because they have been victims of high princes when it comes to medication, but it is not clear from this graph itself. The second thing to note is that very few students who have Very Low food security chose a science major, again I can make conclusions about why that is, but I would say that no reason is evident jsut from this data. Finally, the ratios would be musch more intersting to study to see if there are relationships between college choices and food security, but as it was mentioned before, there doesn't seem to be much by looking at the graph, so I doubt it would tell us anything we cannot already see.

2.4 Compare Classification with Food Security using a Stacked Bar Plot

```
# create a new column with the college names
df['Classes'] = df['Classification']

# define the order of classes for the x-axis
class_order = ['Freshman', 'Sophomore', 'Junior', 'Senior', 'Graduate (Masters)', 'Doctoral', 'Professional (Certificate Program)']

# create a cross-tabulation table of CollegeName and USDAcat with the specified order
ct = pd.crosstab(df['Classes'], df['USDAcat'], colnames=['USDAcat'], rownames=['Classes']).reindex(columns=foodsecurity, index=class_order)

# display the table
print(ct)

# plot the cross-tabulation table as a stacked bar chart with the specified order
ct.plot(kind='bar', stacked=True)
plt.title('Classification Food Security')
plt.xlabel('Classification')
plt.ylabel('Count')
plt.xticks(rotation=90) # rotate x-axis labels for better visibility
plt.show()
```

USDAcat	Very Low FS	Low FS	Marginal/High FS
Classes			
Freshman	225	235	591
Sophomore	268	264	562
Junior	567	441	889
Senior	678	522	1147
Graduate (Masters)	200	172	487
Doctoral	82	82	197
Professional (Certificate Program)	2	8	13



2.5 Observations between Classificationa and Food Security

An interesting observation of this bar plot is that Marginal/High make up more than half the students when it comes to freshmen or doctoral students. This makes intuitive sense because they are more likely the students who can afford to either start college or start a doctoral program.

3. Are there gender or ethnicity differences in the items pertaining to concentration on school and degree progress/completion?

3.1 Output the number of entries in each of Classification, College Choice, Ethnicity and Gender

```
# create dictionary to map numbers to words
ethnicity_dict = {
    '1': 'Hispanic',
    '2': 'American Indian/Alaska native',
    '3': 'Asian',
    '4': 'African American',
    '5': 'Native Hawaiian',
    '6': 'White/Caucasian',
    '7': 'Other',
    '8': 'Prefer not to say'
}
```

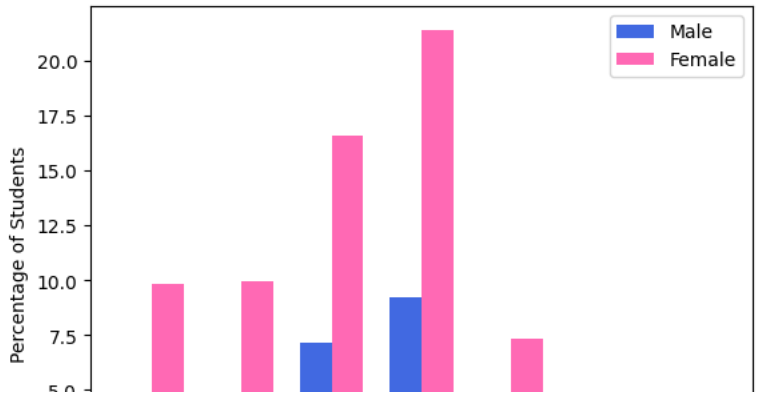
```
# count number of entries for each class and output word instead of number
ethnicities = ['1', '2', '3', '4', '5', '6', '7', '8']
for n in ethnicities:
    count = df[df['Ethnicity'] == n]['Ethnicity'].count()
    ethnicity_word = ethnicity_dict[n]
    print(f'{ethnicity_word}: {count}')
print("\n")


# count number of entries for each class
genders = ['Male', 'Female', 'Transgender', 'Other']
for c in genders:
    count = df[df['Gendercats'] == c]['Gendercats'].count()
    print(f'{c}: {count}')
print("\n")



# count number of entries for each class
classes = ['Freshman', 'Sophomore', 'Junior', 'Senior', 'Graduate (Masters)', 'Doctoral', 'Professional (Certificate Program)']
for c in classes:
    count = df[df['Classification'] == c]['Classification'].count()
    print(f'{c}: {count}')
print("\n")



# create dictionary to map numbers to words
college_dict = {
    '1': 'Business',
    '2': 'Education',
    '3': 'Engineering',
    '4': 'Liberal Arts',
    '5': 'Health Sciences',
    '6': 'Nursing',
    '7': 'Science',
    '8': 'Pharmacy',
    '9': 'Other'
}
# count number of entries for each class and output word instead of number
colleges = ['1', '2', '3', '4', '5', '6', '7', '8', '9']
for n in colleges:
    count = df[df['College'] == n]['College'].count()
    college_word = college_dict[n]
    print(f'{college_word}: {count}')
print("\n")
```

```
Hispanic: 8893
American Indian/Alaska native: 50
Asian: 319
African American: 241
Native Hawaiian: 20
White/Caucasian: 685
Other: 128
Prefer not to say: 93


Male: 2774
Female: 6142
Transgender: 12
Other: 161


Freshman: 1355
Sophomore: 1363
Junior: 2311
Senior: 2877
Graduate (Masters): 1116
Doctoral: 507
Professional (Certificate Program): 26


Business: 1320
Education: 1022
Engineering: 1877
Liberal Arts: 1114
Health Sciences: 2278
Nursing: 1349
Science: 622
```

```
Pharmacy: 679
Other: 876
```

3.2 Compare differences of concentration in gender when it comes to degree progress (classification).

```
# create dictionary to map numbers to words
classification_dict = {
    'Freshman': 0,
    'Sophomore': 1,
    'Junior': 2,
    'Senior': 3,
    'Graduate (Masters)': 4,
    'Doctoral': 5,
    'Professional (Certificate Program)': 6
}

# count number of entries for each classification and gender
class_counts_male = []
class_counts_female = []
for c in classification_dict.keys():
    count_male = df[(df['Classification'] == c) & (df['Gendercats'] == 'Male')]['Classification'].count()
    count_female = df[(df['Classification'] == c) & (df['Gendercats'] == 'Female')]['Classification'].count()
    class_counts_male.append(count_male)
    class_counts_female.append(count_female)

# convert counts to percentages
total_count = sum(class_counts_male) + sum(class_counts_female)
class_percentages_male = [count/total_count*100 for count in class_counts_male]
class_percentages_female = [count/total_count*100 for count in class_counts_female]

# set up x-axis
classifications = list(classification_dict.keys())
x_pos = np.arange(len(classifications))

# create bar chart
bar_width = 0.35
fig, ax = plt.subplots()
rects1 = ax.bar(x_pos - bar_width/2, class_percentages_male, bar_width, label='Male', color='royalblue')
rects2 = ax.bar(x_pos + bar_width/2, class_percentages_female, bar_width, label='Female', color='hotpink')

# rotate x-axis labels
plt.xticks(rotation=60)

# add labels and legend
ax.set_ylabel('Percentage of Students')
ax.set_xlabel('Classification')
ax.set_xticks(x_pos)
ax.set_xticklabels(classifications)
ax.legend()


plt.show()
```



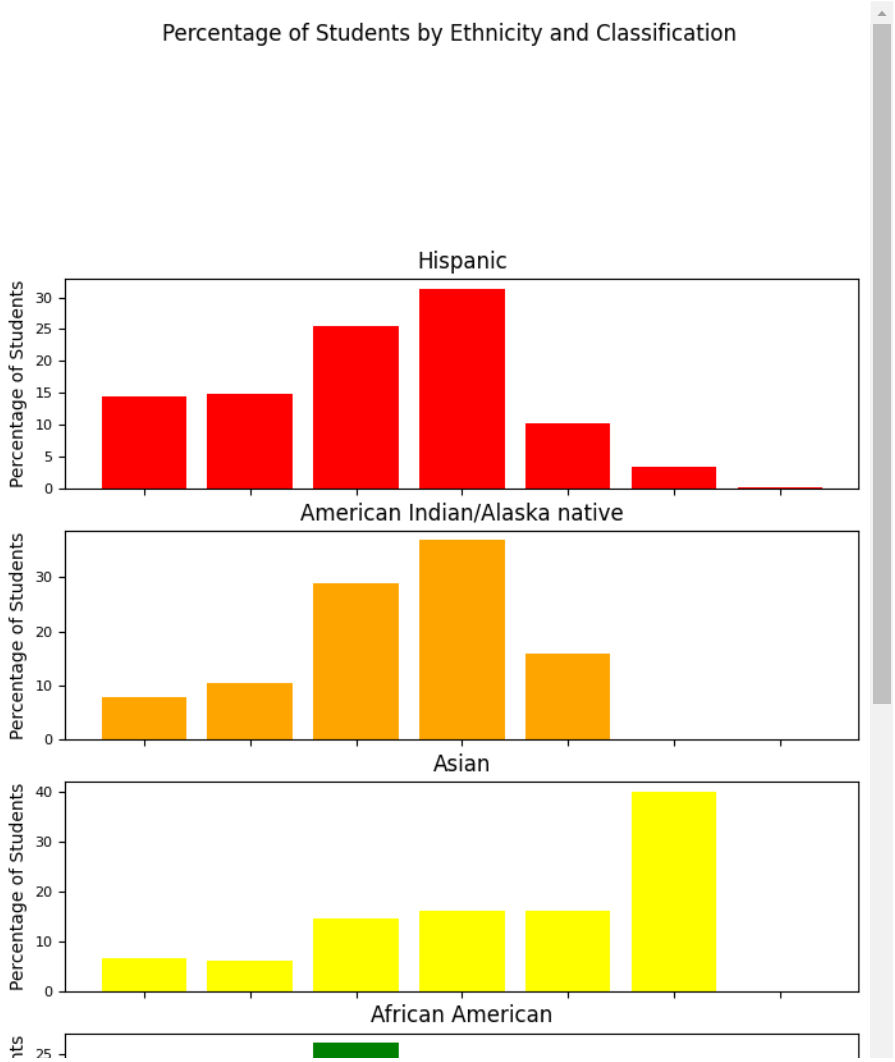
3.3 Observations

Here we see that altough there are more females than males, the concentration or distribution of them is extremely similar. There is enough evidence here to be able to say that both males and females advance at the same pace (or ratios) and don't necessarily complete degrees at higher rates than the other gender.

3.4 Observations between different ethnicities and their completion of degrees

```
# create dictionary to map numbers to words
ethnicity_dict = {
    '1': 'Hispanic',
    '2': 'American Indian/Alaska native',
    '3': 'Asian',
    '4': 'African American',
    '5': 'Native Hawaiian',
    '6': 'White/Caucasian',
    '7': 'Other',
    '8': 'Prefer not to say'
}

classification_dict = {
    'Freshman': 0,
    'Sophomore': 1,
    'Junior': 2,
    'Senior': 3,
    'Graduate (Masters)': 4,
    'Doctoral': 5,
    'Professional (Certificate Program)': 6
}

# set up x-axis
classifications = list(classification_dict.keys())
x_pos = np.arange(len(classifications))

# set up colors for each ethnicity
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'purple', 'brown', 'pink']

# create subplots for each ethnicity
fig, axs = plt.subplots(len(ethnicity_dict), figsize=(8, 20), sharex=True)

# count number of entries for each classification and ethnicity
for i, (ethnicity_num, ethnicity_word) in enumerate(ethnicity_dict.items()):
    ethnicity_counts = []
    for c in classifications:
        count = df[(df['Classification'] == c) & (df['Ethnicity'] == ethnicity_num)]['Classification'].count()
        ethnicity_counts.append(count)

    # convert counts to percentages
    total_count = sum(ethnicity_counts)
    ethnicity_percentages = [count/total_count*100 for count in ethnicity_counts]

    # plot bar chart for each ethnicity
    axs[i].bar(x_pos, ethnicity_percentages, color=colors[i])
    axs[i].set_ylabel('Percentage of Students')
    axs[i].set_title(ethnicity_word)
```

```
        axs[i].tick_params(axis='both', which='major', labelsize=8)

        # set the x-ticks to be classification names
        axs[i].set_xticks(x_pos)
        axs[i].set_xticklabels(classifications)

    # rotate x-axis labels
    plt.xticks(rotation=60)

    # add labels and legend
    fig.suptitle('Percentage of Students by Ethnicity and Classification')
    axs[-1].set_xlabel('Classification')

    plt.show()
```



3.5 Oservations

In order to avoid overwhelming information we avoided graphing all the bars next to each other. Instead, each ethnicity go its own bar chart which still enables us to see any differences. Looking at the different discrete histograms, we see that a much higher ratio of students who prefer not to say and african americans are doctoral students. On the other hand caucasians and hispanics seem to be more normally distributed. Finally, Asian, Native Hawaiian, and Other all tend to be more skewed. Conclusions are hard to make by only looking at the concentration without considering other variables, but a difference in completion is evident.