Open Data Project: Major Wars

Name: Saipranav Venkatakrishnan

Date: 3/10/2020

Source for Data Set: Data.World

URL for Data Set: https://data.world/fivethirtyeight/college-majors/workspace/file?filename=all-ages.csv (https://data.world/fivethirtyeight/college-majors/workspace/file?filename=all-ages.csv)

Description of Data Set: ENTER DESCRIPTION HERE

- This data set gives information on a student ID number, their major, their industry of work, how many are employed/unemployed, the unemployment rate, and how many are employed full time

File Format for Data Set: .csv

Last Updated: 3/10/2020

Part 1: Import and Trim Data (6 points)

- Import file as a pandas dataframe
- Display size of data
- Create a numerical pandas series
- Convert series data type to floating point numbers
- Convert series to a numpy array, remove 'NaN' entries
- Compute three common statistics

Import file as a pandas data frame

- Store in an appropriately-named variable
- Show the first 10 entries:
    - This is a good way for you to see if your modification was successful and to see what the data looks like in your data set.

In [34]:

```
# I first imported panda as pd, and then used panda to import the csv file, which

import pandas as pd

Major_DF = pd.read_csv("College_Majors.csv")

Major_DF[:10]
```

Out[34]:

	Major_code	Major	Major_category	Total	Employed	Employed_full_time_year_roun
0	1100	GENERAL AGRICULTURE	Agriculture & Natural Resources	128148	90245	7407
1	1101	AGRICULTURE PRODUCTION AND MANAGEMENT	Agriculture & Natural Resources	95326	76865	6424
2	1102	AGRICULTURAL ECONOMICS	Agriculture & Natural Resources	33955	26321	2281
3	1103	ANIMAL SCIENCES	Agriculture & Natural Resources	103549	81177	6493
4	1104	FOOD SCIENCE	Agriculture & Natural Resources	24280	17281	1272
5	1105	PLANT SCIENCE AND AGRONOMY	Agriculture & Natural Resources	79409	63043	5107
6	1106	SOIL SCIENCE	Agriculture & Natural Resources	6586	4926	404
7	1199	MISCELLANEOUS AGRICULTURE	Agriculture & Natural Resources	8549	6392	507
8	1301	ENVIRONMENTAL SCIENCE	Biology & Life Science	106106	87602	6523
9	1302	FORESTRY	Agriculture & Natural Resources	69447	48228	3961

Display size of data

In [35]:

```
# Used to read how many entrys there are in the excel
len(Major_DF)
```

Out[35]:

```
173
```

Create a numerical pandas series

- REMINDER: A panda series is similar to a standard list, but works specifically with pandas list of functions
- Locate a numerical column of interest in your pandas data frame
- Store all values from that column in an appropriately-named variable
- Show the first ten entries
  - This is a good way for you to see if your modification was successful and to see what the data looks like

In [36]:

```
# Stored the Employed entries in Major_DF as Employed
Employed = Major_DF["Employed"]
Employed[:10]
```

Out[36]:

```
0    90245
1    76865
2    26321
3    81177
4    17281
5    63043
6     4926
7     6392
8    87602
9    48228
Name: Employed, dtype: int64
```

Convert series data type to floating point numbers

- Modify your data appropriately so that your array holds only decimals
  - Replace all special characters ('$', '%', '#', etc.)
  - Convert all entries to floating point numbers
- Show the first 10 entries
  - This is a good way for you to see if your modification was successful and to see what the data looks like

In [37]:

```
# Read the first 10 entries of Employed
Employed[:10]
```

Out[37]:

```
0    90245
1    76865
2    26321
3    81177
4    17281
5    63043
6     4926
7     6392
8    87602
9    48228
Name: Employed, dtype: int64
```

Convert series to a numpy array, remove 'NaN' entries

- Convert your pandas series to a numpy array with an appr

- Remove all 'NaN' entries
- Show the first 30 entries
    - This is a good way for you to see if your modification was successful and to see what the data looks like

In [38]:
```
# Imported numpy, turn Employed into an array called major_array, and read the f
import numpy as np

major_array = np.array(Employed)

major_array[:30]
```

Out[38]:
```
array([ 90245,  76865,  26321,  81177,  17281,  63043,   4926,   6392,
        87602,  48228,  65937, 216770,  75798, 790696, 314438, 170474,
       147433,  49609, 218248,  22828, 656372,  66393,  32366,  44071,
        33388, 843693,   3113,   1492, 819393,  47203], dtype=int64)
```

In [39]:
```
# Created an array called biggest_employment_rate, and stored the
# numbers that are above 2000000 in major_array and added them to biggest_employ

biggest_employment_rate = []
for employed in major_array:
    if employed > 2000000:
        biggest_employment_rate.append(employed)

biggest_employment_rate
```

Out[39]:
```
[2354398]
```

Compute three common statistics

- Using your numpy array, calculate various common statistics
    - Mean
    - Median
    - Mode
    - Standard deviation
    - Sums
    - Etc.
- Indicate which statistic is being calculated in the markdown cell above the code cell

Mean

In [40]:

```
# I added all the numbers in major_array and then divided them by the length of
Major_Mean = 0

for employed in major_array:
    Major_Mean += employed

Major_Mean = Major_Mean/len(major_array)

Major_Mean
```

Out[40]:

```
166161.98265895955
```

Sum

In [41]:

```
# I added all the numbers in major_array. Stored value in Major_Sum
Major_Sum = 0

for employed in major_array:
    Major_Sum += employed

Major_Sum
```

Out[41]:

```
28746023
```

Standard Deviation

In [42]:

```
# Used numpy to do standard deviation of major_array using np.std(). Stored it a

Major_STD = np.std(major_array)

Major_STD
```

Out[42]:

```
306434.89158024645
```

Part 2: Create a Histogram (6 points)

- WARNING: DO NOT COPY/PASTE ENTIRE GRAPH CODE ALL AT ONCE
  - Use code line-by-line to ensure all values are changed to reflect the ones in this notebook!
- REMINDER: Histograms display frequencies of data values grouped together in chosen ranges
  - Think - what list is storing the required data?
    - Should be a numpy array of numbers
  - Think - how will you divide up the data into bins
    - What is the minimum value?
    - What is the maximum value?
    - What should the size of each bin be?
      - What numbers should each bin start/stop at?
    - Which alignment will you use?
- Create frequency bins

- Set graph details
- Create and display graph

In [43]:
```
#Stored the entry called Total in Major_DF as total_people. Printed first 3 entr
total_people = Major_DF["Total"]

total_people[:3]
```

Out[43]:
```
0    128148
1     95326
2     33955
Name: Total, dtype: int64
```

In [44]:
```
#Turned total_people into an array and stored it as people_array useing numpy

people_array = np.array(total_people)

people_array
```

Out[44]:
```
array([ 128148,   95326,   33955,  103549,   24280,   79409,    6586,
          8549,  106106,   69447,   83188,  294692,  103740,  987676,
        418104,  211213,  186829,   62141,  253782,   29317,  783292,
         77805,   39362,   51771,   42325, 1438867,    4037,    2396,
       1446701,   68808,  281661,  157079,   56477,  224262,  149689,
        127022,   88067,  181445,  231861,  225553,  503080,   65734,
         32748,   19587,   18347,  188046,  358593,  154160,  671647,
         20582,   13016,    6264,  138366,   21430,  581529,   12818,
         10746,   16094,    9826,   19631,   57006,   37382,   47098,
         94697,   82142,   29348,   64196,   75791,  236342,   57793,
        402038,    9330,   67037, 1098647,   59211,  601221,   46188,
         16193,  839454,   75322,   14135,   28197,   45368,    6362,
         68885,    5015,   43984,   55395,   13676,   29389,  432806,
         19112,   24806,    4315,   45199,   56580,   64534,    7184,
          6898,   61871,  350409,  205763,  232865,    8856,    4700,
         14051,  308062,  107902,    8267,   10741,  122620,    7208,
        427953,   12166, 1484075,   14041,    7638,   17633,   17969,
         10871,   34102,  757141,   54636,   14782,   81786,  319163,
        127363,  757616,  143087,   75085,  115423,   77371,  748956,
        674558,   15882,   92346,   15726,  126639,  571961,  174817,
        276262,   55141,  504657,  133508,   90852,   81008,    8511,
        104516,   74977,  108510,   64316,  164990,   32514, 1769892,
        180084,  252138,   56741,   77647, 2148712, 1779219,    9763,
       3123510,   57200,   75547, 1114624,  816548,  187274,   86064,
        200854,  156673,  102753,  712509,   17746], dtype=int64)
```

Create frequency bins and alignment

- Create and store a list of 'edge numbers' for your bins in an appropriately-named variable
  - Your data should fit between the lowest and highest values
  - Use a loop to help create bins quickly

In [45]:

```
# To earn points, you must comment your code
# Stored All the numbers in intervals of 10000 between 20010000 and 0 in the arr

bins = []

for i in range(2001):
    binz = i * 10000
    bins.append(binz)

bins
```

Out[45]:

```
[0,
 10000,
 20000,
 30000,
 40000,
 50000,
 60000,
 70000,
 80000,
 90000,
 100000,
 110000,
 120000,
 130000,
 140000,
 150000,
 160000,
 170000,
 180000,
```

Set graph details

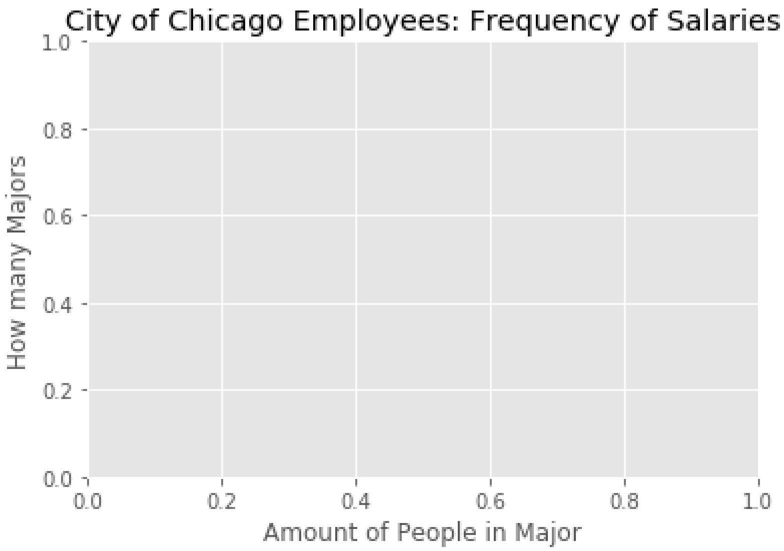
- Import matplotlib appropriately
- Your graph should be styled uniquely compared to any previously created, and must inlude:
    - Title
    - X-axis label
    - Y-axis label
    - Visually appealing style
    - Graph color not previously used

In [46]:

```
#Created a title, x, and y lable for my graph below


plot.title("City of Chicago Employees: Frequency of Salaries")
plot.xlabel("Amount of People in Major")
plot.ylabel("How many Majors")
```

Out[46]:

```
Text(0, 0.5, 'How many Majors')
```

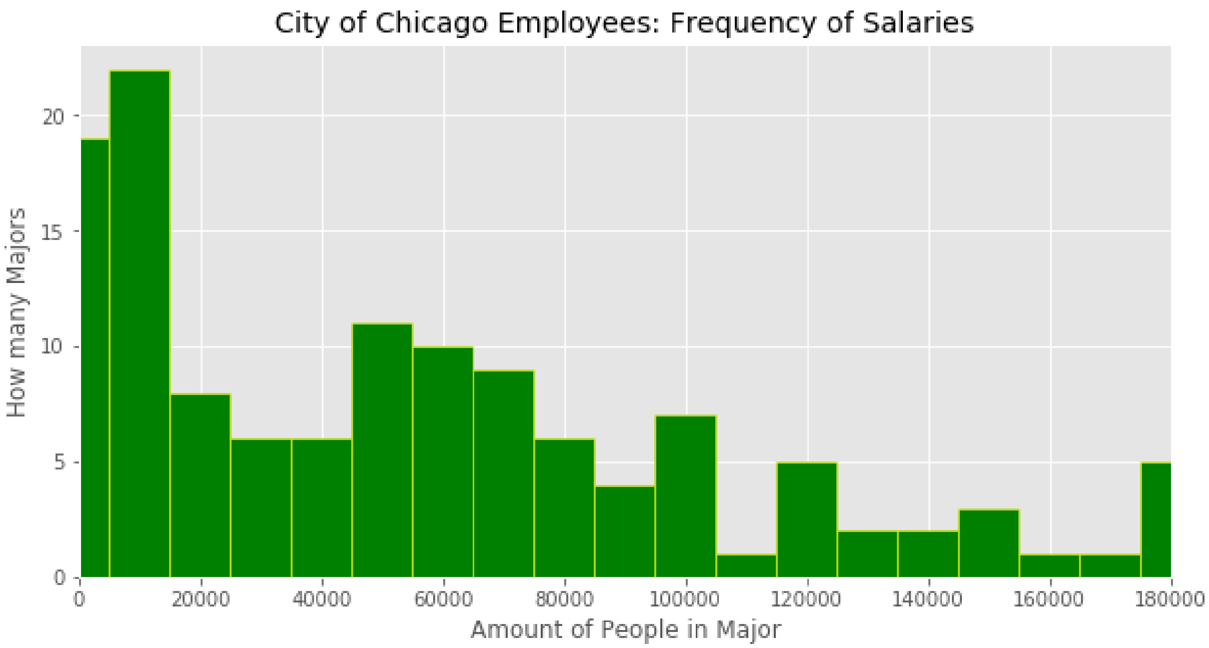


Create and display graph

- HISTOGRAM SYNTAX: plot.hist(data_array, bins_array, extras...)

In [47]:

```
# Copyed the graph from a previous project and changed it accordingly.
#Also afdded a random color picker to make the graph more astheticly pleaseing

import matplotlib.pyplot as plot
%matplotlib inline

# Graph style
plot.style.use("ggplot")
plot.figure(figsize=(10,5))
plot.xlim([0,180000])


colors = ["lightblue", "red", "brown", "orange", "blue", "green", "yellow", "pin
one = np.random.choice(colors)
two = np.random.choice(colors)
while one == two:
    one = np.random.choice(colors)
    two = np.random.choice(colors)


# Create title and lables
plot.title("City of Chicago Employees: Frequency of Salaries")
plot.xlabel("Amount of People in Major")
plot.ylabel("How many Majors")
# Create histogram
plot.hist(people_array, bins, align="left", color=one, edgecolor=two)
plot.show()
```



Part 3: Modifying Data Using Counts (6 points)

- Create a pandas series of value counts
- Create two parallel lists

Create a pandas series of value counts

- Locate a non-numerical column of interest from your original pandas dataframe
  - You will be counting how many times each entry appears in the list
  - Your chosen column should:
    - Have mulptiple repeated entries
    - Have the ability to be grouped into categories
- Create and store a pandas series of value counts in an appropriately-named variable
- Show the series of counts
  - This is a good way for you to see if your modification was successful and to see what the data looks like

To earn points, you must comment your code

In [48]:

```
# Organized the Majors by amount, and stored them in major frequencies
major_frequency = Major_DF["Major_category"].value_counts()

major_frequency
```

Out[48]:

```
Engineering                            29
Education                              16
Humanities & Liberal Arts              15
Biology & Life Science                 14
Business                               13
Health                                 12
Computers & Mathematics                11
Physical Sciences                      10
Agriculture & Natural Resources        10
Social Science                          9
Psychology & Social Work                9
Arts                                    8
Industrial Arts & Consumer Services     7
Law & Public Policy                     5
Communications & Journalism             4
Interdisciplinary                       1
Name: Major_category, dtype: int64
```

Create two parallel lists

- REMINDERS:
  - Use index.tolist() to extract only the category names
  - Use tolist() to extract only the statistic of interest
- Create a list of the category names from the value count panda series
  - Store in an appropriately-named variable
  - Ex: [POLICE, FIRE, OEMC, etc...]
- Create a list of the frequencies for each entry from the value count panda series
  - Store in an appropriately-named variable
  - Ex: [13061, 4837, 2120, etc...]
- Show each list individually
  - This is a good way for you to see if your modifications were successful and to see what the data looks like

In [49]:

```
#I took major frequencies, and split them into two arrays, same length and corro

major_names = major_frequency.index.tolist()


major_counts = major_frequency.tolist()

print(major_names)
print(major_counts)
```

```
['Engineering', 'Education', 'Humanities & Liberal Arts', 'Biology & Life Scien
ce', 'Business', 'Health', 'Computers & Mathematics', 'Physical Sciences', 'Agr
iculture & Natural Resources', 'Social Science', 'Psychology & Social Work', 'A
rts', 'Industrial Arts & Consumer Services', 'Law & Public Policy', 'Communicat
ions & Journalism', 'Interdisciplinary']
[29, 16, 15, 14, 13, 12, 11, 10, 10, 9, 9, 8, 7, 5, 4, 1]
```

Part 4: Creating a Pie Chart (6 points)

- WARNING: DO NOT COPY/PASTE ENTIRE GRAPH CODE ALL AT ONCE
  - Use code line-by-line to ensure all values are changed to reflect the ones in this notebook!
- REMINDER: Pie charts display the ratio of values in comparison to the whole
  - Think - Which lists are storing the required data?
    - One list of counts
    - One list of labels
- NOTE: If you have 9 or more categories you should create an "Other" category to earn full credit!
- Create an "Other" category (if necessary)
- Set graph details
- Create and display graph

Create an "Other" category (if necessary)

- If you have 9 or more categories, lump the remainging categories as one "Other" category

- All labels should be clearly visible

In [50]:

```
# Took the saparted arrays, and made them shorter.
#For their respective arrays, the names were shortened to others, and the others

short_names = major_names[:10]


short_names.append("Others")

short_counts = major_counts[:10]
remaining = sum(major_counts[10:])
short_counts.append(remaining)

print(short_names)
print(short_counts)
```

```
['Engineering', 'Education', 'Humanities & Liberal Arts', 'Biology & Life Scien
ce', 'Business', 'Health', 'Computers & Mathematics', 'Physical Sciences', 'Agr
iculture & Natural Resources', 'Social Science', 'Others']
[29, 16, 15, 14, 13, 12, 11, 10, 10, 9, 34]
```

Set graph details

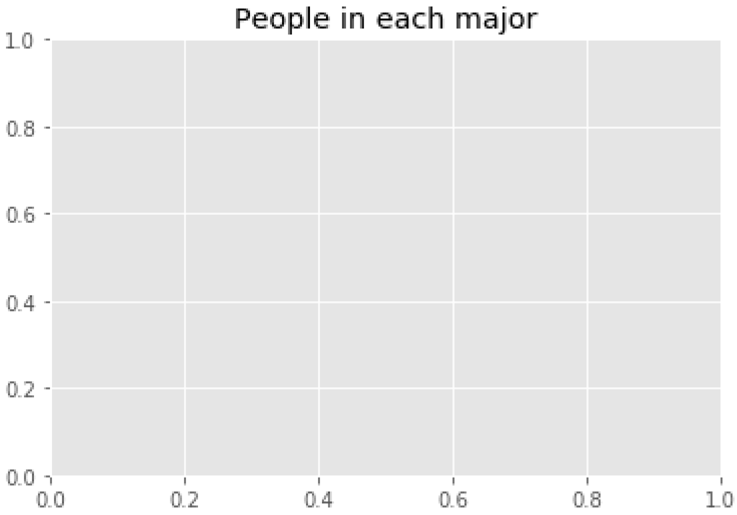
- Import matplotlib appropriately
- Your graph should be styled uniquely compared to any previously created, and must inlude:
  - Title
  - Appropriate labels
  - Visually appealing style
  - Graph color not previously used

In [51]:

```
# Created a title for the pie chart
plot.title("People in each major")
```

Out[51]:

```
Text(0.5, 1.0, 'People in each major')
```



Create and display graph

- PIE CHART SYNTAX: plot.pie(counts_list, labels_list, extras...)

In [52]:

```
# To earn points, you must comment your code
```

In [53]:

```
# Copyed the pie chart from the previous project, and changed things accodringly

import matplotlib.pyplot as plot
%matplotlib inline

# Graph details
plot.figure(figsize=(10,10))
colors_array = ["lightblue", "red", "brown", "orange", "blue", "green", "yellow"

plot.title("People in each major")

# Create pie chart
    # plot.pie(data array, labels array, color array, percentages, shadow)
plot.pie(short_counts, labels=short_names, colors=colors_array, autopct="%1.1f%%
#plot.legend(title="Legend), loc="lower_left")
plot.show()
```

Part 5: Be a Data Scienctist! (12 points)

- Create an additional question
- Create an appropriate pandas series
- Split into individual lists
- Create list of sorted tuples
- Split into individual sorted lists

Create an additional question

- How else can your categories be compared?
- What additional number can your categories be compared by?
  - Ex: Which category has largest/smallest...
    - ...average?
    - ...maximum?
    - ...minimum?
    - ...standard deviation?
- State your question relative to the data you are working with

Question: Which major has the most people employed?

Create an appropriate pandas series

- REMINDER: Use .groupby() to organize your series based on another column's data

In [54]:
```
# Stored the Emplored data set in terms of the Mojor data set. Stored as the var
dept_major_avg = Major_DF["Employed"].groupby(Major_DF["Major"]).mean()
dept_major_avg
```

Out[54]:
```
Major
ACCOUNTING                                 1335825
ACTUARIAL SCIENCE                             7846
ADVERTISING AND PUBLIC RELATIONS            147433
AEROSPACE ENGINEERING                        44944
AGRICULTURAL ECONOMICS                       26321
                                            ...
TRANSPORTATION SCIENCES AND TECHNOLOGIES     98814
TREATMENT THERAPY PROFESSIONS               199174
UNITED STATES HISTORY                        11887
VISUAL AND PERFORMING ARTS                   41098
ZOOLOGY                                      35714
Name: Employed, Length: 173, dtype: int64
```

Split into individual lists

- Create two parallel lists
  - One list of categories
  - One list of statistics of interest
- Show each list individually
  - This is a good way for you to see if your modifications were successful and to see what the data looks like

In [55]:

```
# Split the above dictionary into two separate arrays
dept_major_average = dept_major_avg.index.tolist()

# Array of salaries sorted by department
major_avg = dept_major_avg.tolist()

# Print the arrays
print(dept_major_average)
print(major_avg)
```

```
['ACCOUNTING', 'ACTUARIAL SCIENCE', 'ADVERTISING AND PUBLIC RELATIONS', 'AEROSP
ACE ENGINEERING', 'AGRICULTURAL ECONOMICS', 'AGRICULTURE PRODUCTION AND MANAGEM
ENT', 'ANIMAL SCIENCES', 'ANTHROPOLOGY AND ARCHEOLOGY', 'APPLIED MATHEMATICS',
'ARCHITECTURAL ENGINEERING', 'ARCHITECTURE', 'AREA ETHNIC AND CIVILIZATION STUD
IES', 'ART AND MUSIC EDUCATION', 'ART HISTORY AND CRITICISM', 'ASTRONOMY AND AS
TROPHYSICS', 'ATMOSPHERIC SCIENCES AND METEOROLOGY', 'BIOCHEMICAL SCIENCES', 'B
IOLOGICAL ENGINEERING', 'BIOLOGY', 'BIOMEDICAL ENGINEERING', 'BOTANY', 'BUSINES
S ECONOMICS', 'BUSINESS MANAGEMENT AND ADMINISTRATION', 'CHEMICAL ENGINEERING',
'CHEMISTRY', 'CIVIL ENGINEERING', 'CLINICAL PSYCHOLOGY', 'COGNITIVE SCIENCE AND
BIOPSYCHOLOGY', 'COMMERCIAL ART AND GRAPHIC DESIGN', 'COMMUNICATION DISORDERS S
CIENCES AND SERVICES', 'COMMUNICATION TECHNOLOGIES', 'COMMUNICATIONS', 'COMMUNI
TY AND PUBLIC HEALTH', 'COMPOSITION AND RHETORIC', 'COMPUTER ADMINISTRATION MAN
AGEMENT AND SECURITY', 'COMPUTER AND INFORMATION SYSTEMS', 'COMPUTER ENGINEERIN
G', 'COMPUTER NETWORKING AND TELECOMMUNICATIONS', 'COMPUTER PROGRAMMING AND DAT
A PROCESSING', 'COMPUTER SCIENCE', 'CONSTRUCTION SERVICES', 'COSMETOLOGY SERVIC
ES AND CULINARY ARTS', 'COUNSELING PSYCHOLOGY', 'COURT REPORTING', 'CRIMINAL JU
STICE AND FIRE PROTECTION', 'CRIMINOLOGY', 'DRAMA AND THEATER ARTS', 'EARLY CHI
LDHOOD EDUCATION', 'ECOLOGY', 'ECONOMICS', 'EDUCATIONAL ADMINISTRATION AND SUPE
RVISION', 'EDUCATIONAL PSYCHOLOGY', 'ELECTRICAL ENGINEERING', 'ELECTRICAL ENGIN
EERING TECHNOLOGY', 'ELECTRICAL, MECHANICAL, AND PRECISION TECHNOLOGIES AND PRO
DUCTION', 'ELEMENTARY EDUCATION', 'ENGINEERING AND INDUSTRIAL MANAGEMENT', 'ENG
INEERING MECHANICS PHYSICS AND SCIENCE', 'ENGINEERING TECHNOLOGIES', 'ENGLISH L
ANGUAGE AND LITERATURE', 'ENVIRONMENTAL ENGINEERING', 'ENVIRONMENTAL SCIENCE',
'FAMILY AND CONSUMER SCIENCES', 'FILM VIDEO AND PHOTOGRAPHIC ARTS', 'FINANCE',
'FINE ARTS', 'FOOD SCIENCE', 'FORESTRY', 'FRENCH GERMAN LATIN AND OTHER COMMON
FOREIGN LANGUAGE STUDIES', 'GENERAL AGRICULTURE', 'GENERAL BUSINESS', 'GENERAL
EDUCATION', 'GENERAL ENGINEERING', 'GENERAL MEDICAL AND HEALTH SERVICES', 'GENE
RAL SOCIAL SCIENCES', 'GENETICS', 'GEOGRAPHY', 'GEOLOGICAL AND GEOPHYSICAL ENGI
NEERING', 'GEOLOGY AND EARTH SCIENCE', 'GEOSCIENCES', 'HEALTH AND MEDICAL ADMIN
ISTRATIVE SERVICES', 'HEALTH AND MEDICAL PREPARATORY PROGRAMS', 'HISTORY', 'HOS
PITALITY MANAGEMENT', 'HUMAN RESOURCES AND PERSONNEL MANAGEMENT', 'HUMAN SERVIC
ES AND COMMUNITY ORGANIZATION', 'HUMANITIES', 'INDUSTRIAL AND MANUFACTURING ENG
INEERING', 'INDUSTRIAL AND ORGANIZATIONAL PSYCHOLOGY', 'INDUSTRIAL PRODUCTION T
ECHNOLOGIES', 'INFORMATION SCIENCES', 'INTERCULTURAL AND INTERNATIONAL STUDIE
S', 'INTERDISCIPLINARY SOCIAL SCIENCES', 'INTERNATIONAL BUSINESS', 'INTERNATION
AL RELATIONS', 'JOURNALISM', 'LANGUAGE AND DRAMA EDUCATION', 'LIBERAL ARTS', 'L
IBRARY SCIENCE', 'LINGUISTICS AND COMPARATIVE LANGUAGE AND LITERATURE', 'MANAGE
MENT INFORMATION SYSTEMS AND STATISTICS', 'MARKETING AND MARKETING RESEARCH',
'MASS MEDIA', 'MATERIALS ENGINEERING AND MATERIALS SCIENCE', 'MATERIALS SCIENC
E', 'MATHEMATICS', 'MATHEMATICS AND COMPUTER SCIENCE', 'MATHEMATICS TEACHER EDU
CATION', 'MECHANICAL ENGINEERING', 'MECHANICAL ENGINEERING RELATED TECHNOLOGIE
S', 'MEDICAL ASSISTING SERVICES', 'MEDICAL TECHNOLOGIES TECHNICIANS', 'METALLUR
GICAL ENGINEERING', 'MICROBIOLOGY', 'MILITARY TECHNOLOGIES', 'MINING AND MINERA
L ENGINEERING', 'MISCELLANEOUS AGRICULTURE', 'MISCELLANEOUS BIOLOGY', 'MISCELLA
NEOUS BUSINESS & MEDICAL ADMINISTRATION', 'MISCELLANEOUS EDUCATION', 'MISCELLAN
EOUS ENGINEERING', 'MISCELLANEOUS ENGINEERING TECHNOLOGIES', 'MISCELLANEOUS FIN
E ARTS', 'MISCELLANEOUS HEALTH MEDICAL PROFESSIONS', 'MISCELLANEOUS PSYCHOLOG
Y', 'MISCELLANEOUS SOCIAL SCIENCES', 'MOLECULAR BIOLOGY', 'MULTI-DISCIPLINARY O
R GENERAL SCIENCE', 'MULTI/INTERDISCIPLINARY STUDIES', 'MUSIC', 'NATURAL RESOUR
CES MANAGEMENT', 'NAVAL ARCHITECTURE AND MARINE ENGINEERING', 'NEUROSCIENCE',
'NUCLEAR ENGINEERING', 'NUCLEAR, INDUSTRIAL RADIOLOGY, AND BIOLOGICAL TECHNOLOG
IES', 'NURSING', 'NUTRITION SCIENCES', 'OCEANOGRAPHY', 'OPERATIONS LOGISTICS AN
D E-COMMERCE', 'OTHER FOREIGN LANGUAGES', 'PETROLEUM ENGINEERING', 'PHARMACOLOG
Y', 'PHARMACY PHARMACEUTICAL SCIENCES AND ADMINISTRATION', 'PHILOSOPHY AND RELI
GIOUS STUDIES', 'PHYSICAL AND HEALTH EDUCATION TEACHING', 'PHYSICAL FITNESS PAR
KS RECREATION AND LEISURE', 'PHYSICAL SCIENCES', 'PHYSICS', 'PHYSIOLOGY', 'PLAN
T SCIENCE AND AGRONOMY', 'POLITICAL SCIENCE AND GOVERNMENT', 'PRE-LAW AND LEGAL
STUDIES', 'PSYCHOLOGY', 'PUBLIC ADMINISTRATION', 'PUBLIC POLICY', 'SCHOOL STUDE
NT COUNSELING', 'SCIENCE AND COMPUTER TEACHER EDUCATION', 'SECONDARY TEACHER ED
UCATION', 'SOCIAL PSYCHOLOGY', 'SOCIAL SCIENCE OR HISTORY TEACHER EDUCATION',
'SOCIAL WORK', 'SOCIOLOGY', 'SOIL SCIENCE', 'SPECIAL NEEDS EDUCATION', 'STATIST
ICS AND DECISION SCIENCE', 'STUDIO ARTS', 'TEACHER EDUCATION: MULTIPLE LEVELS',
'THEOLOGY AND RELIGIOUS VOCATIONS', 'TRANSPORTATION SCIENCES AND TECHNOLOGIES',
'TREATMENT THERAPY PROFESSIONS', 'UNITED STATES HISTORY', 'VISUAL AND PERFORMIN
G ARTS', 'ZOOLOGY']
[1335825, 7846, 147433, 44944, 26321, 76865, 81177, 102399, 15136, 13713, 21677
0, 75798, 155159, 61295, 3400, 11252, 52594, 24270, 583079, 12876, 9284, 57983,
2354398, 131697, 198075, 262831, 5128, 5527, 379980, 49393, 49609, 790696, 4254
3, 44913, 32366, 218248, 128742, 44071, 22828, 656372, 79055, 33388, 13071, 727
0, 613369, 59534, 135071, 113460, 36708, 535446, 3113, 8751, 489965, 73737, 126
07, 819393, 27275, 14909, 30102, 708882, 9849, 87602, 241585, 107651, 670681, 3
86961, 17281, 48228, 153654, 90245, 1580978, 843693, 359172, 78198, 80165, 474
7, 83671, 4120, 75698, 6129, 85360, 19009, 478416, 163393, 142879, 61402, 2997
1, 101273, 11878, 65401, 66393, 43114, 43312, 66453, 56564, 314438, 111347, 404
932, 7091, 45657, 134478, 890125, 170474, 14687, 5866, 280902, 5874, 47203, 422
207, 24190, 51279, 121479, 6939, 45422, 1650, 7416, 6392, 22298, 77471, 126054,
43906, 53097, 6431, 52610, 23921, 12307, 20221, 308461, 35706, 192704, 65937, 1
0690, 8987, 7320, 9560, 1325711, 43878, 7882, 47341, 34696, 14002, 3481, 12405
8, 138734, 193542, 286683, 5872, 80797, 31394, 63043, 541630, 49259, 1055854, 3
7879, 11147, 1492, 36224, 129486, 6897, 78785, 225081, 459174, 4926, 108272, 18
808, 58799, 58885, 164827, 98814, 199174, 11887, 41098, 35714]
```

Create list of sorted tuples

- Loop through parallel lists to create tuples in the form (category, statistic)
- Sort appropriately
    - REMINDER: Use list_name.sort(key=lambda x: x[1], reverse=True) to sort
- Show the list
    - This is a good way for you to see if your modifications were successful and to see what the data looks like

In [56]:

```python
# Combined above arrays and tuples in one array. The tuples are sorted by how big
sorted_major_list = []

for i in range(len(dept_major_average)):
    sorted_major_list.append((dept_major_average[i], major_avg[i]))

# Sort the list in reverse
sorted_major_list.sort(key = lambda x: x[1], reverse=True)

# Print
sorted_major_list
```

Out[56]:

```
[('BUSINESS MANAGEMENT AND ADMINISTRATION', 2354398),
 ('GENERAL BUSINESS', 1580978),
 ('ACCOUNTING', 1335825),
 ('NURSING', 1325711),
 ('PSYCHOLOGY', 1055854),
 ('MARKETING AND MARKETING RESEARCH', 890125),
 ('GENERAL EDUCATION', 843693),
 ('ELEMENTARY EDUCATION', 819393),
 ('COMMUNICATIONS', 790696),
 ('ENGLISH LANGUAGE AND LITERATURE', 708882),
 ('FINANCE', 670681),
 ('COMPUTER SCIENCE', 656372),
 ('CRIMINAL JUSTICE AND FIRE PROTECTION', 613369),
 ('BIOLOGY', 583079),
 ('POLITICAL SCIENCE AND GOVERNMENT', 541630),
 ('ECONOMICS', 535446),
 ('ELECTRICAL ENGINEERING', 489965),
 ('HISTORY', 478416),
 ('SOCIOLOGY', 459174),
 ('MECHANICAL ENGINEERING', 4[illegible])
```

Split into individual sorted lists

- Option #1
  - List iterator
  - list_name = ['specific stat' for 'entry' in 'name of array']
  - Read as: A list of every 'specific stat' for every 'entry' in the array 'name of array'
- Option #2
  - Loops
  - Create an empty list
  - Run a loop that:
    - Stores each appropriate category and statistic
    - Converts them to a tuple
    - Adds the tuple to the list
- Show the list
  - This is a good way for you to see if your modifications were successful and to see what the data looks like

In [57]:

```
# Converted above  array into two separate array that is organized by how big the
major = []
major_number = []

for tuple in sorted_major_list:
    major.append(tuple[0])
    major_number.append(tuple[1])

print(major)
print(major_number)
```

```
['BUSINESS MANAGEMENT AND ADMINISTRATION', 'GENERAL BUSINESS', 'ACCOUNTING', 'N
URSING', 'PSYCHOLOGY', 'MARKETING AND MARKETING RESEARCH', 'GENERAL EDUCATION',
'ELEMENTARY EDUCATION', 'COMMUNICATIONS', 'ENGLISH LANGUAGE AND LITERATURE', 'F
INANCE', 'COMPUTER SCIENCE', 'CRIMINAL JUSTICE AND FIRE PROTECTION', 'BIOLOGY',
'POLITICAL SCIENCE AND GOVERNMENT', 'ECONOMICS', 'ELECTRICAL ENGINEERING', 'HIS
TORY', 'SOCIOLOGY', 'MECHANICAL ENGINEERING', 'LIBERAL ARTS', 'FINE ARTS', 'COM
MERCIAL ART AND GRAPHIC DESIGN', 'GENERAL ENGINEERING', 'JOURNALISM', 'MULTI-DI
SCIPLINARY OR GENERAL SCIENCE', 'PHYSICAL FITNESS PARKS RECREATION AND LEISUR
E', 'MATHEMATICS', 'CIVIL ENGINEERING', 'FAMILY AND CONSUMER SCIENCES', 'SOCIAL
WORK', 'COMPUTER AND INFORMATION SYSTEMS', 'ARCHITECTURE', 'TREATMENT THERAPY P
ROFESSIONS', 'CHEMISTRY', 'PHYSICAL AND HEALTH EDUCATION TEACHING', 'MUSIC', 'M
ASS MEDIA', 'THEOLOGY AND RELIGIOUS VOCATIONS', 'HOSPITALITY MANAGEMENT', 'ART
AND MUSIC EDUCATION', 'FRENCH GERMAN LATIN AND OTHER COMMON FOREIGN LANGUAGE ST
UDIES', 'ADVERTISING AND PUBLIC RELATIONS', 'HUMAN RESOURCES AND PERSONNEL MANA
GEMENT', 'PHILOSOPHY AND RELIGIOUS STUDIES', 'DRAMA AND THEATER ARTS', 'MANAGEM
ENT INFORMATION SYSTEMS AND STATISTICS', 'CHEMICAL ENGINEERING', 'SECONDARY TEA
CHER EDUCATION', 'COMPUTER ENGINEERING', 'MISCELLANEOUS EDUCATION', 'PHARMACY P
HARMACEUTICAL SCIENCES AND ADMINISTRATION', 'MEDICAL TECHNOLOGIES TECHNICIANS',
'EARLY CHILDHOOD EDUCATION', 'LANGUAGE AND DRAMA EDUCATION', 'SPECIAL NEEDS EDU
CATION', 'FILM VIDEO AND PHOTOGRAPHIC ARTS', 'ANTHROPOLOGY AND ARCHEOLOGY', 'IN
DUSTRIAL AND MANUFACTURING ENGINEERING', 'TRANSPORTATION SCIENCES AND TECHNOLOG
IES', 'GENERAL AGRICULTURE', 'ENVIRONMENTAL SCIENCE', 'HEALTH AND MEDICAL ADMIN
ISTRATIVE SERVICES', 'GEOGRAPHY', 'ANIMAL SCIENCES', 'PHYSICS', 'GENERAL SOCIAL
SCIENCES', 'CONSTRUCTION SERVICES', 'SOCIAL SCIENCE OR HISTORY TEACHER EDUCATIO
N', 'GENERAL MEDICAL AND HEALTH SERVICES', 'MISCELLANEOUS BUSINESS & MEDICAL AD
MINISTRATION', 'AGRICULTURE PRODUCTION AND MANAGEMENT', 'AREA ETHNIC AND CIVILI
ZATION STUDIES', 'GEOLOGY AND EARTH SCIENCE', 'ELECTRICAL ENGINEERING TECHNOLOG
Y', 'INTERNATIONAL BUSINESS', 'INFORMATION SCIENCES', 'NATURAL RESOURCES MANAGE
MENT', 'INDUSTRIAL PRODUCTION TECHNOLOGIES', 'PLANT SCIENCE AND AGRONOMY', 'HUM
AN SERVICES AND COMMUNITY ORGANIZATION', 'ART HISTORY AND CRITICISM', 'CRIMINOL
OGY', 'TEACHER EDUCATION: MULTIPLE LEVELS', 'STUDIO ARTS', 'BUSINESS ECONOMIC
S', 'INTERNATIONAL RELATIONS', 'MISCELLANEOUS ENGINEERING TECHNOLOGIES', 'MISCE
LLANEOUS HEALTH MEDICAL PROFESSIONS', 'BIOCHEMICAL SCIENCES', 'MEDICAL ASSISTIN
G SERVICES', 'COMMUNICATION TECHNOLOGIES', 'COMMUNICATION DISORDERS SCIENCES AN
D SERVICES', 'PRE-LAW AND LEGAL STUDIES', 'FORESTRY', 'OPERATIONS LOGISTICS AND
E-COMMERCE', 'MATHEMATICS TEACHER EDUCATION', 'LINGUISTICS AND COMPARATIVE LANG
UAGE AND LITERATURE', 'MICROBIOLOGY', 'AEROSPACE ENGINEERING', 'COMPOSITION AND
RHETORIC', 'COMPUTER NETWORKING AND TELECOMMUNICATIONS', 'MISCELLANEOUS ENGINEE
RING', 'NUTRITION SCIENCES', 'INTERDISCIPLINARY SOCIAL SCIENCES', 'INTERCULTURA
L AND INTERNATIONAL STUDIES', 'COMMUNITY AND PUBLIC HEALTH', 'VISUAL AND PERFOR
MING ARTS', 'PUBLIC ADMINISTRATION', 'ECOLOGY', 'SCIENCE AND COMPUTER TEACHER E
DUCATION', 'ZOOLOGY', 'MULTI/INTERDISCIPLINARY STUDIES', 'OTHER FOREIGN LANGUAG
ES', 'COSMETOLOGY SERVICES AND CULINARY ARTS', 'COMPUTER ADMINISTRATION MANAGEM
ENT AND SECURITY', 'PHYSIOLOGY', 'ENGINEERING TECHNOLOGIES', 'HUMANITIES', 'ENG
INEERING AND INDUSTRIAL MANAGEMENT', 'AGRICULTURAL ECONOMICS', 'BIOLOGICAL ENGI
NEERING', 'MECHANICAL ENGINEERING RELATED TECHNOLOGIES', 'MISCELLANEOUS PSYCHOL
OGY', 'COMPUTER PROGRAMMING AND DATA PROCESSING', 'MISCELLANEOUS BIOLOGY', 'MOL
ECULAR BIOLOGY', 'HEALTH AND MEDICAL PREPARATORY PROGRAMS', 'STATISTICS AND DEC
ISION SCIENCE', 'FOOD SCIENCE', 'APPLIED MATHEMATICS', 'ENGINEERING MECHANICS P
HYSICS AND SCIENCE', 'MATERIALS ENGINEERING AND MATERIALS SCIENCE', 'PETROLEUM 
ENGINEERING', 'ARCHITECTURAL ENGINEERING', 'COUNSELING PSYCHOLOGY', 'BIOMEDICAL
ENGINEERING', 'ELECTRICAL, MECHANICAL, AND PRECISION TECHNOLOGIES AND PRODUCTIO
N', 'MISCELLANEOUS SOCIAL SCIENCES', 'UNITED STATES HISTORY', 'INDUSTRIAL AND O
RGANIZATIONAL PSYCHOLOGY', 'ATMOSPHERIC SCIENCES AND METEOROLOGY', 'PUBLIC POLI
CY', 'NAVAL ARCHITECTURE AND MARINE ENGINEERING', 'ENVIRONMENTAL ENGINEERING', 
'NUCLEAR, INDUSTRIAL RADIOLOGY, AND BIOLOGICAL TECHNOLOGIES', 'BOTANY', 'NEUROS
CIENCE', 'EDUCATIONAL PSYCHOLOGY', 'OCEANOGRAPHY', 'ACTUARIAL SCIENCE', 'MINING
AND MINERAL ENGINEERING', 'NUCLEAR ENGINEERING', 'COURT REPORTING', 'LIBRARY SC
IENCE', 'METALLURGICAL ENGINEERING', 'SOCIAL PSYCHOLOGY', 'MISCELLANEOUS FINE A
RTS', 'MISCELLANEOUS AGRICULTURE', 'GEOSCIENCES', 'MATHEMATICS AND COMPUTER SCI
ENCE', 'PHYSICAL SCIENCES', 'MATERIALS SCIENCE', 'COGNITIVE SCIENCE AND BIOPSYC
HOLOGY', 'CLINICAL PSYCHOLOGY', 'SOIL SCIENCE', 'GENETICS', 'GEOLOGICAL AND GEO
PHYSICAL ENGINEERING', 'PHARMACOLOGY', 'ASTRONOMY AND ASTROPHYSICS', 'EDUCATION
AL ADMINISTRATION AND SUPERVISION', 'MILITARY TECHNOLOGIES', 'SCHOOL STUDENT CO
UNSELING']
[2354398, 1580978, 1335825, 1325711, 1055854, 890125, 843693, 819393, 790696, 7
08882, 670681, 656372, 613369, 583079, 541630, 535446, 489965, 478416, 459174, 
422207, 404932, 386961, 379980, 359172, 314438, 308461, 286683, 280902, 262831,
241585, 225081, 218248, 216770, 199174, 198075, 193542, 192704, 170474, 164827,
163393, 155159, 153654, 147433, 142879, 138734, 135071, 134478, 131697, 129486,
128742, 126054, 124058, 121479, 113460, 111347, 108272, 107651, 102399, 101273,
98814, 90245, 87602, 85360, 83671, 81177, 80797, 80165, 79055, 78785, 78198, 77
471, 76865, 75798, 75698, 73737, 66453, 66393, 65937, 65401, 63043, 61402, 6129
5, 59534, 58885, 58799, 57983, 56564, 53097, 52610, 52594, 51279, 49609, 49393,
49259, 48228, 47341, 47203, 45657, 45422, 44944, 44913, 44071, 43906, 43878, 43
312, 43114, 42543, 41098, 37879, 36708, 36224, 35714, 35706, 34696, 33388, 3236
6, 31394, 30102, 29971, 27275, 26321, 24270, 24190, 23921, 22828, 22298, 20221,
19009, 18808, 17281, 15136, 14909, 14687, 14002, 13713, 13071, 12876, 12607, 12
307, 11887, 11878, 11252, 11147, 10690, 9849, 9560, 9284, 8987, 8751, 7882, 784
6, 7416, 7320, 7270, 7091, 6939, 6897, 6431, 6392, 6129, 5874, 5872, 5866, 552
7, 5128, 4926, 4747, 4120, 3481, 3400, 3113, 1650, 1492]
```

In [58]:

```
# I shortend both arrays and tacked on the remainders at the end. Stored them in

short_major = major[:45]
short_major_number = major_number[:45]

short_major.append("Others")
short_major_number.append(sum(major_number[30:]))

print(short_major)
print(short_major_number)
```

```
['BUSINESS MANAGEMENT AND ADMINISTRATION', 'GENERAL BUSINESS', 'ACCOUNTING', 'N
URSING', 'PSYCHOLOGY', 'MARKETING AND MARKETING RESEARCH', 'GENERAL EDUCATION',
'ELEMENTARY EDUCATION', 'COMMUNICATIONS', 'ENGLISH LANGUAGE AND LITERATURE', 'F
INANCE', 'COMPUTER SCIENCE', 'CRIMINAL JUSTICE AND FIRE PROTECTION', 'BIOLOGY',
'POLITICAL SCIENCE AND GOVERNMENT', 'ECONOMICS', 'ELECTRICAL ENGINEERING', 'HIS
TORY', 'SOCIOLOGY', 'MECHANICAL ENGINEERING', 'LIBERAL ARTS', 'FINE ARTS', 'COM
MERCIAL ART AND GRAPHIC DESIGN', 'GENERAL ENGINEERING', 'JOURNALISM', 'MULTI-DI
SCIPLINARY OR GENERAL SCIENCE', 'PHYSICAL FITNESS PARKS RECREATION AND LEISUR
E', 'MATHEMATICS', 'CIVIL ENGINEERING', 'FAMILY AND CONSUMER SCIENCES', 'SOCIAL
WORK', 'COMPUTER AND INFORMATION SYSTEMS', 'ARCHITECTURE', 'TREATMENT THERAPY P
ROFESSIONS', 'CHEMISTRY', 'PHYSICAL AND HEALTH EDUCATION TEACHING', 'MUSIC', 'M
ASS MEDIA', 'THEOLOGY AND RELIGIOUS VOCATIONS', 'HOSPITALITY MANAGEMENT', 'ART
AND MUSIC EDUCATION', 'FRENCH GERMAN LATIN AND OTHER COMMON FOREIGN LANGUAGE ST
UDIES', 'ADVERTISING AND PUBLIC RELATIONS', 'HUMAN RESOURCES AND PERSONNEL MANA
GEMENT', 'PHILOSOPHY AND RELIGIOUS STUDIES', 'Others']
[2354398, 1580978, 1335825, 1325711, 1055854, 890125, 843693, 819393, 790696, 7
08882, 670681, 656372, 613369, 583079, 541630, 535446, 489965, 478416, 459174,
422207, 404932, 386961, 379980, 359172, 314438, 308461, 286683, 280902, 262831,
241585, 225081, 218248, 216770, 199174, 198075, 193542, 192704, 170474, 164827,
163393, 155159, 153654, 147433, 142879, 138734, 8364184]
```

Part 6: Create a Bar Chart to Visualize Data (6 points)

- WARNING: DO NOT COPY/PASTE ENTIRE GRAPH CODE ALL AT ONCE
  - Use code line-by-line to ensure all values are changed to reflect the ones in this notebook!
- REMINDER: Bar charts displays one item's value compared to others'
  - Think - which lists are holding the required data?
    - One list of 'bar positions'
    - One list of statistics of interest
    - One list of labels
- Create a 'bar position list'
- Set graph details
- Create and display graph

Create a 'bar position list'

- REMINDER: Use np.arange(start, length+1)

In [59]:

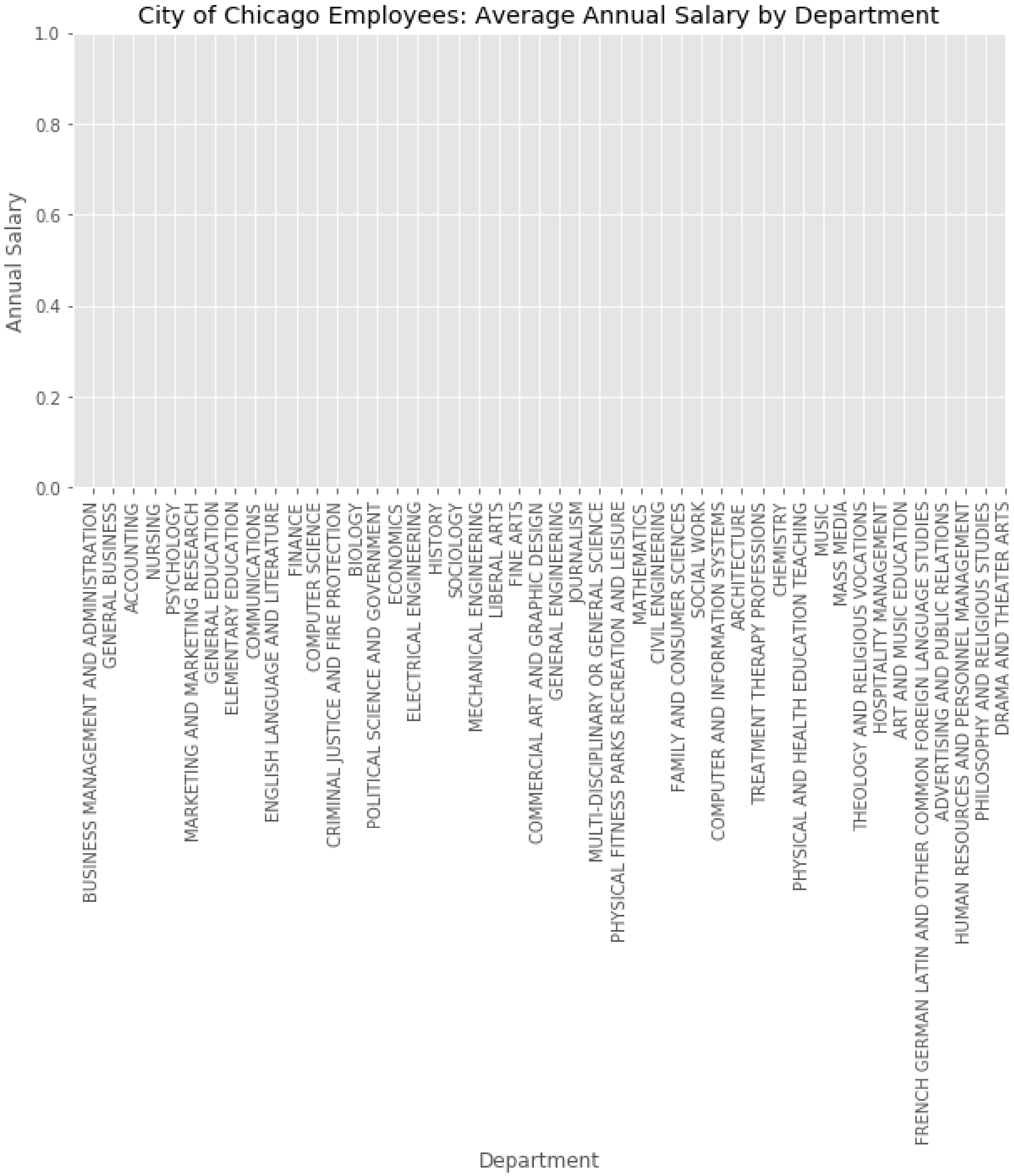
```
# Created bar_pos for plot.xticks
bar_pos = np.arange(1, length+1)
```

Set graph details

- Import matplotlib appropriately
- Your graph should be styled uniquely compared to any previously created, and must inlude:
  - Title
  - X-axis label
  - Y-axis label
  - Bar labels (xticks)
  - Visually appealing style
  - Graph color not previously used

```
In [60]: # Created the styling of the graph,size, and title and other stuff for the desig
         plot.style.use("ggplot")  #fivethirty eight, bmh; grayscale, dark_background, gg
         plot.figure(figsize=(10,5))
         plot.title('City of Chicago Employees: Average Annual Salary by Department')
         plot.ylabel('Annual Salary')
         plot.xlabel('Department')
         plot.xticks(bar_pos, major, rotation=90)
```

```
Out[60]: ([<matplotlib.axis.XTick at 0x1707faeb7c8>,
           <matplotlib.axis.XTick at 0x1707fa4ea48>,
           <matplotlib.axis.XTick at 0x1707e3f1c08>,
           <matplotlib.axis.XTick at 0x1707fb3a148>,
           <matplotlib.axis.XTick at 0x1707d3fd548>,
           <matplotlib.axis.XTick at 0x1707d3fd888>,
           <matplotlib.axis.XTick at 0x1707d3fd9c8>,
           <matplotlib.axis.XTick at 0x1707fa7f448>,
           <matplotlib.axis.XTick at 0x1707d434bc8>,
           <matplotlib.axis.XTick at 0x1707d41b848>,
           <matplotlib.axis.XTick at 0x1707d41b248>,
           <matplotlib.axis.XTick at 0x1707fd52f48>,
           <matplotlib.axis.XTick at 0x1707fa8cc08>,
           <matplotlib.axis.XTick at 0x1707fa8c488>,
           <matplotlib.axis.XTick at 0x1707fa768c8>,
           <matplotlib.axis.XTick at 0x1707fa8c608>,
           <matplotlib.axis.XTick at 0x1707d434988>,
           <matplotlib.axis.XTick at 0x1707d3d3588>,
           <matplotlib.axis.XTick at 0x1707d3d82c8>,
           <matplotlib.axis.XTick at 0x1707faa5a08>,
           <matplotlib.axis.XTick at 0x1707faa5b08>,
           <matplotlib.axis.XTick at 0x1707faab5c8>,
           <matplotlib.axis.XTick at 0x1707fb0f988>,
           <matplotlib.axis.XTick at 0x1707fa6c648>,
           <matplotlib.axis.XTick at 0x1707d404f48>,
           <matplotlib.axis.XTick at 0x1707d404948>,
           <matplotlib.axis.XTick at 0x1707faa5688>,
           <matplotlib.axis.XTick at 0x1707d40fd88>,
           <matplotlib.axis.XTick at 0x1707d40ffc8>,
           <matplotlib.axis.XTick at 0x1707d6c1508>,
           <matplotlib.axis.XTick at 0x1707e84ddc8>,
           <matplotlib.axis.XTick at 0x1707e84dd08>,
           <matplotlib.axis.XTick at 0x1707fa857c8>,
           <matplotlib.axis.XTick at 0x1707fd54cc8>,
           <matplotlib.axis.XTick at 0x1707fd54c48>,
           <matplotlib.axis.XTick at 0x1707ea4f8c8>,
           <matplotlib.axis.XTick at 0x1707faa5ec8>,
           <matplotlib.axis.XTick at 0x1707e83c2c8>,
           <matplotlib.axis.XTick at 0x1707fb0c108>,
           <matplotlib.axis.XTick at 0x1707fa8f708>,
           <matplotlib.axis.XTick at 0x1707e837208>,
           <matplotlib.axis.XTick at 0x1707fa53e08>,
           <matplotlib.axis.XTick at 0x1707e8453c8>,
           <matplotlib.axis.XTick at 0x1707e8489c8>,
           <matplotlib.axis.XTick at 0x1707e848448>,
           <matplotlib.axis.XTick at 0x1707e848648>],
          <a list of 46 Text xticklabel objects>)
```

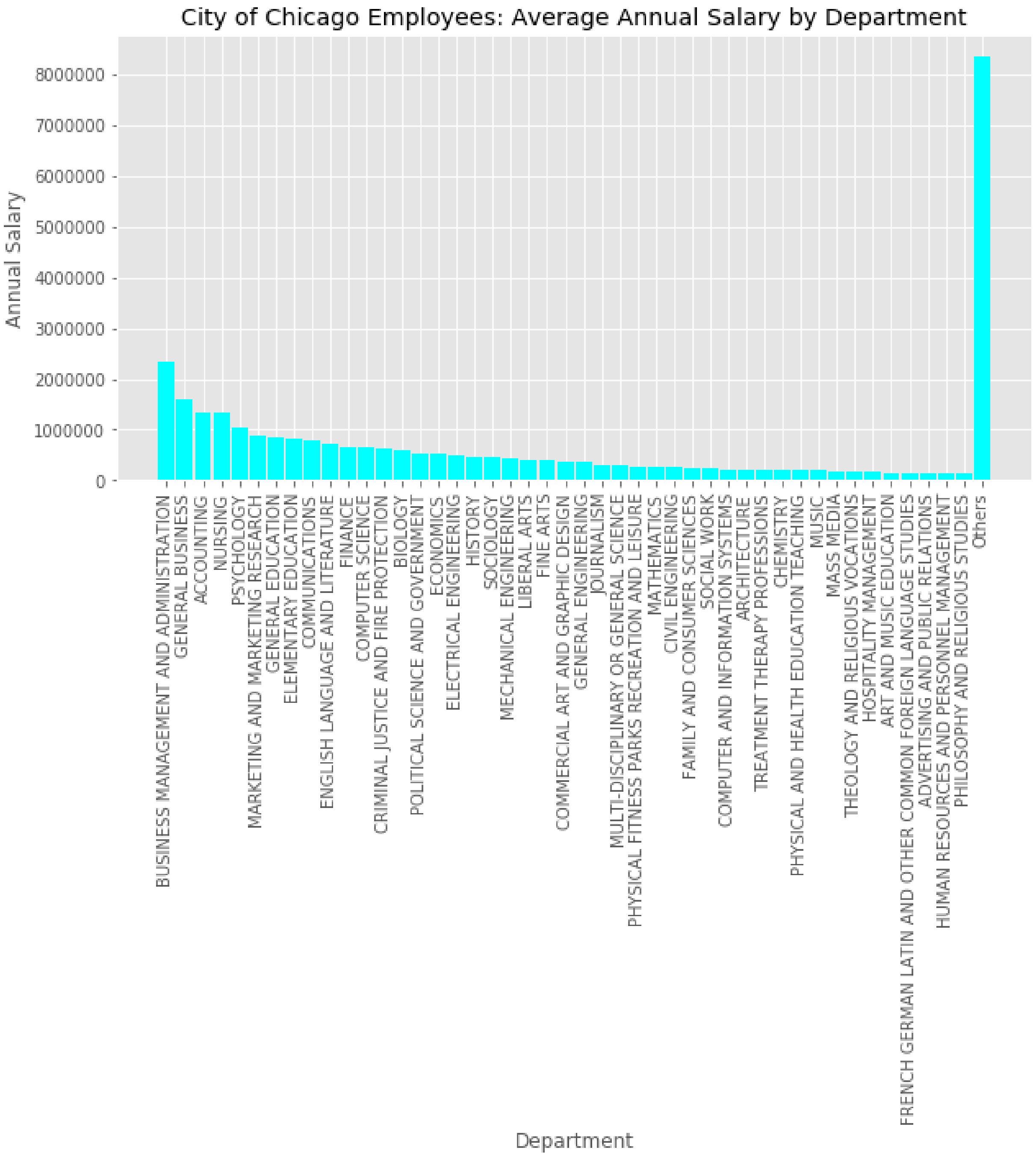

City of Chicago Employees: Average Annual Salary by Department
Annual Salary
1.0
0.8
0.6
0.4
0.2
0.0
BUSINESS MANAGEMENT AND ADMINISTRATION
GENERAL BUSINESS
ACCOUNTING
NURSING
PSYCHOLOGY
MARKETING AND MARKETING RESEARCH
GENERAL EDUCATION
ELEMENTARY EDUCATION
COMMUNICATIONS
ENGLISH LANGUAGE AND LITERATURE
FINANCE
COMPUTER SCIENCE
CRIMINAL JUSTICE AND FIRE PROTECTION
BIOLOGY
POLITICAL SCIENCE AND GOVERNMENT
ECONOMICS
ELECTRICAL ENGINEERING
HISTORY
SOCIOLOGY
MECHANICAL ENGINEERING
LIBERAL ARTS
FINE ARTS
COMMERCIAL ART AND GRAPHIC DESIGN
GENERAL ENGINEERING
JOURNALISM
MULTI-DISCIPLINARY OR GENERAL SCIENCE
PHYSICAL FITNESS PARKS RECREATION AND LEISURE
MATHEMATICS
CIVIL ENGINEERING
FAMILY AND CONSUMER SCIENCES
SOCIAL WORK
COMPUTER AND INFORMATION SYSTEMS
ARCHITECTURE
TREATMENT THERAPY PROFESSIONS
CHEMISTRY
PHYSICAL AND HEALTH EDUCATION TEACHING
MUSIC
MASS MEDIA
THEOLOGY AND RELIGIOUS VOCATIONS
HOSPITALITY MANAGEMENT
ART AND MUSIC EDUCATION
FRENCH GERMAN LATIN AND OTHER COMMON FOREIGN LANGUAGE STUDIES
ADVERTISING AND PUBLIC RELATIONS
HUMAN RESOURCES AND PERSONNEL MANAGEMENT
PHILOSOPHY AND RELIGIOUS STUDIES
DRAMA AND THEATER ARTS
Department

Create and display graph

- BAR GRAPH SYNTAX: plot.bar(bars_list, stat_list, etc.)

In [61]:

```
# copyed the graph from previous project, and changed info accordingly added a co
# Import matplotlib and numpy
import random

import matplotlib.pyplot as plot
%matplotlib inline
import numpy as np

length = len(short_major)
bar_pos = np.arange(1, length+1)

# Graph details
plot.style.use("ggplot")  #fivethirty eight, bmh; grayscale, dark_background, ggp
plot.figure(figsize=(10,5))
plot.title('City of Chicago Employees: Average Annual Salary by Department')
plot.ylabel('Annual Salary')
plot.xlabel('Department')
plot.xticks(bar_pos, short_major, rotation=90)


 # numpy array of numbers [1-36]
width = .9

colors = ["lightblue", "red", "brown", "orange", "blue", "green", "yellow", "pinl
one = np.random.choice(colors)

plot.bar(bar_pos, short_major_number, width, color=one)
plot.show()
```



Part 7: Answer (6 points)

Answer the original question at hand

- Your response must:
  - Clearly state a data-based answer
  - Contain a brief summary of how the data was organized and compared
  - Describe how your visualization helps verify the stated answer

Answer: According to the created simulation, Buissness Managment and Administration have the most people officially employed. I did this by first grabing on to the employed people with respect to majors, and then sorting them for most employed to least employed i tuples. I then split the tuples into two organized arrays, which I then shortended to accomidate the visual appeal of the graph. I finally imputed the arrays into a pre_pasted template of a bar graph, and troubleshooted for any errors.

Part 8: Reflection (2 points)

What do you think are some of the limitations of your analysis?

- Was there additional data that you would've liked to have used if it were available?
- Did time limit you from answering a different interesting question?
- Is there anything you would have liked to have dug into deeper?
- Provide a thoughtful, well-written response

ANSWER HERE

While I think that I did well on my simulation, if I had more time, I would have tryed to explore the relationship between the amount of people in each degree, and how many of those people actually are employed. Some limitations that I faced when doing this were the fact that comppared to other computer scientist and data anyliszers, I dont have much coding experience. Also, I really wanted to see which majors made the most poepl happy, but unfortuantly, machienes and code can analise and compute the immencsity of human emotions. Finally, i felt that a key limitation was time for me, as I really wanted to do some other comparisions for this project, but I dont have time to do any of them.