Code – Austin Hudgins

https://github.com/AustinHudgins/DataVisualizationCSE578

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.mosaicplot import mosaic
from scipy import stats



csvfile = r"Project\adult.csv"

#data frame with all the data.
adultsStats = pd.read_csv(csvfile,header=0)

#remove fnlwgt because it is not being used for this project.
adultsStats = adultsStats.drop(['fnlwgt'],axis=1)

#Look for '?' and replace with nulls
adultsStats =   adultsStats.replace('?','NULL')

#Transform Sex data to binary form
adultsStats['sex'] = adultsStats['sex'].replace('Female',0)
adultsStats['sex'] = adultsStats['sex'].replace('Male',1)

dictOfUniqueOccupation = {}
#get a dictonary of all the occupations in the data frame
for val in adultsStats['occupation'].unique():
    dictOfUniqueOccupation[val] = 0
```

Code – Austin Hudgins

https://github.com/AustinHudgins/DataVisualizationCSE578

```
#Remove NULL JOB

del dictOfUniqueOccupation['NULL']


#colors to use in graphs

arrayOfColors =
['yellow','crimson','deepskyblue','green','aquamarine','violet','brown','lightsalmon','steelblue','cyan','olive','mediumseagreen','grey','orange']

color1 = "cornflowerblue"

color2 = "orange"


#create 2 dict one for tracking ppl abve 50k one for below.

dictOfUniqueOccupationUnder50k = dict(dictOfUniqueOccupation)

dictOfUniqueOccupationOver50k = dict(dictOfUniqueOccupation)


for row in adultsStats.iterrows():
    occupation = row[1][5]
    income = row[1][13]
    sex = row[1][8]
    # Ignore Jobs with NULL not in the dict
    if(occupation == 'NULL'):
        continue
    if(income == '<=50K'):
        dictOfUniqueOccupationUnder50k[occupation] = dictOfUniqueOccupationUnder50k[occupation] +
1
    elif(income == '>50K'):
        dictOfUniqueOccupationOver50k[occupation] = dictOfUniqueOccupationOver50k[occupation] + 1
    else:
        print("Something is wrong check for this income:" + str(income))
```

Code – Austin Hudgins

https://github.com/AustinHudgins/DataVisualizationCSE578

```
#-----------------------------------------------------------------------------------------------\
def bar_graph_of_occupations_by_income_groups():


    # What Jobs should we target when looking for ppl to recuirt to our school?


    #make the bar graph
    plt.bar(x=dictOfUniqueOccupation.keys(),height=dictOfUniqueOccupationUnder50k.values(), color =
color1, label = "occupation under 50k", labels = "Yes")
    plt.bar(x=dictOfUniqueOccupation.keys(),height=dictOfUniqueOccupationOver50k.values(), color =
color2, label = "occupation over 50k", labels = "Yes")


    plt.title('Total people per occupuation based on income groups')
    plt.xticks(rotation =45, ha = "right")
    plt.ylabel('Number of people')
    plt.xlabel('Occupation')
    plt.legend(('Income at or less then $50k','Income over $50k'),loc = 'upper right', )
    plt.show()


def donut_graphs_of_occupantions_by_income_groups():



    # I condensed 'Armed-Forces' and Priv-house-serv' into Other-service clean up outliers that dont
afeect or help


    dictOfUniqueOccupationUnder50k['Other-service'] = dictOfUniqueOccupationUnder50k['Other-
service'] + dictOfUniqueOccupationUnder50k['Armed-Forces'] + dictOfUniqueOccupationUnder50k['Priv-
house-serv']
    copyOfdictOfUniqueOccupation = dict(dictOfUniqueOccupation)
    del copyOfdictOfUniqueOccupation['Armed-Forces']
```

```
    del copyOfdictOfUniqueOccupation['Priv-house-serv']
    del dictOfUniqueOccupationUnder50k['Armed-Forces']
    del dictOfUniqueOccupationUnder50k['Priv-house-serv']
    arrayOfPplPerOccupationUnder50k = dictOfUniqueOccupationUnder50k.values()


    plt.pie(arrayOfPplPerOccupationUnder50k,shadow=False,autopct='%1.2f%%',textprops={'fontsize':13},
colors=arrayOfColors, labels=copyOfdictOfUniqueOccupation.keys(), labeldistance =1.01 )
    plt.title("Job market for income at or under 50k", loc='center', fontsize = 20)
    plt.axis("equal")
    plt.gcf().gca().add_artist(plt.Circle( (0,0), 0.7, color='white'))
    plt.show()
    # ----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------

    dictOfUniqueOccupationOver50k['Other-service'] = dictOfUniqueOccupationOver50k['Other-service']
+ dictOfUniqueOccupationOver50k['Armed-Forces'] + dictOfUniqueOccupationOver50k['Priv-house-
serv']
    del dictOfUniqueOccupationOver50k['Armed-Forces']
    del dictOfUniqueOccupationOver50k['Priv-house-serv']
    arrayOfPplPerOccupationOver50k = dictOfUniqueOccupationOver50k.values()\


    plt.pie(arrayOfPplPerOccupationOver50k,shadow=False,autopct='%1.2f%%',textprops={'fontsize':13},
colors=arrayOfColors, labels=copyOfdictOfUniqueOccupation.keys(), labeldistance =1.01 )
    plt.title("Job market for income over 50k", loc='center', fontsize = 20)
    plt.axis("equal")
    plt.gcf().gca().add_artist(plt.Circle( (0,0), 0.7, color='white'))
    plt.show()


def Line_plot_age_vs_Captial_loss_by_income():
    # User Story 3 - People that didnt finish college, over all ages
```

Code – Austin Hudgins

https://github.com/AustinHudgins/DataVisualizationCSE578

```
df = adultsStats[['age','capital-loss','income']]


ageOfPeopleUnder50k = []
capitalLossOfPeopleUnder50k = []
agePeopleOver50k = []
capitalLossOfPeopleOver50k = []
for row in df.iterrows():

    age = row[1][0]
    capitalLoss = row[1][1]
    income = row[1][2]


    if(age == 'NULL' or capitalLoss == 'NULL' or income == 'NULL'):
        continue
    elif(income == '<=50K'):
        ageOfPeopleUnder50k.append(age)
        capitalLossOfPeopleUnder50k.append(capitalLoss)
    elif(income == '>50K'):
        agePeopleOver50k.append(age)
        capitalLossOfPeopleOver50k.append(capitalLoss)
    else:
        print("Something is wrong check for this income:" + str(income))
x1 = ageOfPeopleUnder50k
y1 = capitalLossOfPeopleUnder50k
x2 = agePeopleOver50k
y2 = capitalLossOfPeopleOver50k
```

```
    plt.scatter(x1,y1)
    plt.scatter(x2,y2)
    plt.show()

def Line_plot_age_vs_Captial_avg_gain_by_income():
    # User Story 3 - People that didnt finish college, over all ages

    df = adultsStats[['age','capital-gain','income']]
    # ageOfPeopleUnder50k = []
    # capitalGainOfPeopleUnder50k = []
    # agePeopleOver50k = []
    # capitalGainOfPeopleOver50k = []
    dictAgeAndGainUnder50k = {}
    dictAgeAndGainOver50k = {}

    nonzeroCaptialGain = []
    for val in df['capital-gain']:
        if val != 0:
            nonzeroCaptialGain.append(val)

    std = np.std(nonzeroCaptialGain)

    for row in df.iterrows():

        age = row[1][0]
        capitalGain = row[1][1]
        income = row[1][2]
```

```
        if(age == 'NULL' or capitalGain == 'NULL' or income == 'NULL'):
            continue
        elif(income == '<=50K' and capitalGain < 2 * std):
            if(age in dictAgeAndGainUnder50k):
                dictAgeAndGainUnder50k[age][0] = dictAgeAndGainUnder50k[age][0] + capitalGain
                dictAgeAndGainUnder50k[age][1] = dictAgeAndGainUnder50k[age][1] + 1
            else:
                dictAgeAndGainUnder50k[age] = [capitalGain,1]
        elif(income == '>50K'and capitalGain < 2 * std):
            if(age in dictAgeAndGainOver50k):
                dictAgeAndGainOver50k[age][0] = dictAgeAndGainOver50k[age][0] + capitalGain
                dictAgeAndGainOver50k[age][1] = dictAgeAndGainOver50k[age][1] + 1
            else:
                dictAgeAndGainOver50k[age] = [capitalGain,1]
        else:
            continue
            print("Something is wrong check for this income:" + str(capitalGain) )

    #https://www.freecodecamp.org/news/python-sort-dictionary-by-key/
    dictAgeAndGainUnder50k = dict(sorted(dictAgeAndGainUnder50k.items()))
    dictAgeAndGainOver50k = dict(sorted(dictAgeAndGainOver50k.items()))

    for key in dictAgeAndGainUnder50k:
        dictAgeAndGainUnder50k[key] =
round((dictAgeAndGainUnder50k[key][0]/dictAgeAndGainUnder50k[key][1]),2)
    for key in dictAgeAndGainOver50k:
```

```
        dictAgeAndGainOver50k[key] =
round((dictAgeAndGainOver50k[key][0]/dictAgeAndGainOver50k[key][1]),2)


    x1 = dictAgeAndGainUnder50k.keys()
    y1 = dictAgeAndGainUnder50k.values()
    x2 = dictAgeAndGainOver50k.keys()
    y2 = dictAgeAndGainOver50k.values()
    plt.plot(x1,y1,color1)
    plt.plot(x2,y2,color2)
    plt.legend(["Income =< 50k", "Income > 50k"])
    plt.title("Average Capital Gain by Age based on Income")
    plt.ylabel("Average Captial Gain ($)")
    plt.xlabel("Age")
    plt.show()


def mosaic_plot_location_by_gender_by_income():
    df = adultsStats[['native-counntry','sex','income']]


    dict = {('Inside Of USA', 'Female', 'Income > 50k') : 0,
            ('Inside Of USA', 'Male', 'Income > 50k') : 0,
            ('Outside Of USA', 'Female', 'Income > 50k') : 0,
            ('Outside Of USA', 'Male', 'Income > 50k') : 0,
            ('Inside Of USA', 'Female', 'Income <= 50k') : 0,
            ('Inside Of USA', 'Male', 'Income <= 50k') : 0,
            ('Outside Of USA', 'Female', 'Income <= 50k') : 0,
            ('Outside Of USA', 'Male', 'Income <= 50k') : 0  }
```

```
    for row in df.iterrows():
        country = row[1][0]
        sex = row[1][1]
        income = row[1][2]
        if(country == 'NULL' or sex == 'NULL' or income == 'NULL'):
            continue
        elif(country=='United-States' or country=='Puerto-Rico' or country=='Outlying-US(Guam-USVI-etc)'):
            if(sex == 0):
                if(income == '<=50K'):
                    dict[('Inside Of USA', 'Female', 'Income <= 50k')] = dict[('Inside Of USA', 'Female', 'Income <=
50k')] + 1
                if(income == '>50K'):
                    dict[('Inside Of USA', 'Female', 'Income > 50k')] = dict[('Inside Of USA', 'Female', 'Income >
50k')] + 1


            elif(sex == 1):
                if(income == '<=50K'):
                    dict[('Inside Of USA', 'Male', 'Income <= 50k')] = dict[('Inside Of USA', 'Male', 'Income <= 50k')]
+ 1
                if(income == '>50K'):
                    dict[('Inside Of USA', 'Male', 'Income > 50k')] = dict[('Inside Of USA', 'Male', 'Income > 50k')] +
1
            else:
                print("THERE MIGHT BE A PROBLEM WITH THIS: " + row)
        else:
            if(sex == 0):
                if(income == '<=50K'):
                    dict[('Outside Of USA', 'Female', 'Income <= 50k')] = dict[('Outside Of USA', 'Female', 'Income
<= 50k')] + 1
```

```
            if(income == '>50K'):
                dict[('Outside Of USA', 'Female', 'Income > 50k')] = dict[('Outside Of USA', 'Female', 'Income > 50k')] + 1


        elif(sex == 1):
            if(income == '<=50K'):
                dict[('Outside Of USA', 'Male', 'Income <= 50k')] = dict[('Outside Of USA', 'Male', 'Income <= 50k')] + 1
            if(income == '>50K'):
                dict[('Outside Of USA', 'Male', 'Income > 50k')] = dict[('Outside Of USA', 'Male', 'Income > 50k')] + 1
        else:
            print("THERE MIGHT BE A PROBLEM WITH THIS: " + row)


    #https://www.statsmodels.org/devel/generated/statsmodels.graphics.mosaicplot.mosaic.html
    dictOfColor = {('Inside Of USA', 'Female', 'Income <= 50k') : 'goldenrod',
            ('Inside Of USA', 'Female', 'Income > 50k') : 'darkgoldenrod',
            ('Inside Of USA', 'Male', 'Income <= 50k') : 'orange',
            ('Inside Of USA', 'Male', 'Income > 50k') : 'darkorange',
            ('Outside Of USA', 'Female', 'Income <= 50k') : 'cornflowerblue',
            ('Outside Of USA', 'Female', 'Income > 50k') : 'royalblue',
            ('Outside Of USA', 'Male', 'Income <= 50k') : 'skyblue',
           ('Outside Of USA', 'Male', 'Income > 50k') : 'deepskyblue'  }
    label=lambda k:{k:str(round(dict[k]/sum(dict.values())*100,1))+'%'}[k]
    #https://stackoverflow.com/questions/61704718/choosing-another-color-palette-for-a-mosaic-plot
    mosaic(dict, gap=0.003,labelizer=label,  properties = lambda key: {'color': dictOfColor[key]})
    plt.title("Birthright US Citizens and Gender compared by Income Groups")
    plt.show()
```

```
def Education_and_avg_capital_loss_by_income_group():
    df = adultsStats[['capital-loss','education-num','income']]


    dictLossAndEducationUnder50k = {}
    dictLossAndEducationOver50k = {}


    # nonzeroCaptialloss = []
    # for val in df['capital-loss']:
    #     if val != 0:
    #         nonzeroCaptialloss.append(val)


    # std = np.std(nonzeroCaptialloss)


    # print(std)
    # print(df)
    for row in df.iterrows():


        loss = row[1][0]
        education = row[1][1]
        income = row[1][2]


        if(loss == 'NULL' or education == 'NULL' or income == 'NULL'):
            continue
        elif(income == '<=50K'):
            if(education in dictLossAndEducationUnder50k):
                dictLossAndEducationUnder50k[education][0] = dictLossAndEducationUnder50k[education][0]
+ loss
```

```
            dictLossAndEducationUnder50k[education][1] = dictLossAndEducationUnder50k[education][1]
+ 1
        else:
            dictLossAndEducationUnder50k[education] = [loss,1]
    elif(income == '>50K'):
        if(education in dictLossAndEducationOver50k):
            dictLossAndEducationOver50k[education][0] = dictLossAndEducationOver50k[education][0] +
loss
            dictLossAndEducationOver50k[education][1] = dictLossAndEducationOver50k[education][1] + 1
        else:
            dictLossAndEducationOver50k[education] = [loss,1]
    else:
        continue


  for key in dictLossAndEducationOver50k:
    dictLossAndEducationOver50k[key] =
round((dictLossAndEducationOver50k[key][0]/dictLossAndEducationOver50k[key][1]),2)
  for key in dictLossAndEducationUnder50k:
    dictLossAndEducationUnder50k[key] =
round((dictLossAndEducationUnder50k[key][0]/dictLossAndEducationUnder50k[key][1]),2)


  x1 = dictLossAndEducationUnder50k.keys()
  y1 = dictLossAndEducationUnder50k.values()
  x2 = dictLossAndEducationOver50k.keys()
  y2 = dictLossAndEducationOver50k.values()
  print(x1)
  print(y1)
  plt.scatter(x1,y1, color = color1)
  plt.scatter(x2,y2, color = color2)
```

```
    plt.xticks([1,2,3,4,5,6,7,8,9,10,11,12,13,14,14,15,16])
    plt.title("Average Capital Loss by Education Level based on Income")
    plt.ylabel("Average Captial Loss ($)")
    plt.xlabel("Education Level")
    plt.legend(["Income =< 50k", "Income > 50k"])
    plt.show()

def box_plot_workclass_by_Income_under_50k():
    df = adultsStats[['hours-per-week','income']]

    dictHoursWorkPerWeekByIncome = {'under50k' : [],
                                    'over50k' : []}

    for row in df.iterrows():
        hoursPerWeek= row[1][0]
        income = row[1][1]



        if(hoursPerWeek == 'NULL' or income == 'NULL'):
            continue
        if(income == '<=50K'):
            dictHoursWorkPerWeekByIncome['under50k'].append(hoursPerWeek)
        elif(income == '>50K'):
            dictHoursWorkPerWeekByIncome['over50k'].append(hoursPerWeek)
        else:
            print("Something is wrong check for this income:" + str(income))
```

```
    fig, ax = plt.subplots()
    ax.boxplot(dictHoursWorkPerWeekByIncome.values())
    ax.set_xticklabels(labels = dictHoursWorkPerWeekByIncome.keys(),  fontsize = 20)
    ax.set_yticklabels(["0","10","20","30","40","50","60","70","80","90","100"],fontsize = 20)
    plt.title("Hours worked per week by Income Group" , fontsize = 25)
    plt.ylabel("Hours worked per week" , fontsize = 20)
    plt.xlabel("Income ($)" , fontsize = 20)

    plt.show()



# bar_graph_of_occupations_by_income_groups()
#donut_graphs_of_occupantions_by_income_groups()
#Line_plot_age_vs_Captial_loss_by_income()
Line_plot_age_vs_Captial_avg_gain_by_income()
# mosaic_plot_location_by_gender_by_income()
# Education_and_avg_capital_loss_by_income_group()

#box_plot_workclass_by_Income_under_50k()
# print(uniqueOccupation)
#print(adultsStats[adultsStats['workclass'] == '?']) FINDS SPECFIC VAL in COL
#print(adultsStats['occupation'].unique())  FINDS UNQUIE VALS IN COL
```