In [1]:
```
# -*- coding: utf-8 -*-
"""
Created on Sun Aug 28 10:04:13 2022

@author: Brandon Botzer - btb5103
"""

"""

Question 1:
Upload Assignment4_data.csv  Download Assignment4_data.csvinto Python.

Please perform the following steps:

1) Explore the datasets. (10 points)
2) Find and handle missing values are in the data. (It is your choice how you har
3) Explore the variable column and Convert the "variable" column to dummy variabl
4) Convert the "one" column into 3 bins. (20 points)

"""
```

Out[1]:
```
'\n\nQuestion 1:\nUpload Assignment4_data.csv  Download Assignment4_data.csvint
o Python.\n\nPlease perform the following steps:\n\n1) Explore the datasets. (1
0 points)\n2) Find and handle missing values are in the data. (It is your choic
e how you handle the missing data.) ( 20 points)\n3) Explore the variable colum
n and Convert the "variable" column to dummy variables and join the dummies to
the data. (20 points)\n4) Convert the "one" column into 3 bins. (20 points)\n
\n'
```

In [2]:
```
print("""
      Question 1:
      Upload Assignment4_data.csv  Download Assignment4_data.csvinto Python.

      Please perform the following steps:

      1) Explore the datasets. (10 points)
      2) Find and handle missing values are in the data. (It is your choice how y
      3) Explore the variable column and Convert the "variable" column to dummy v
      4) Convert the "one" column into 3 bins. (20 points)
      """)
```

```
      Question 1:
      Upload Assignment4_data.csv  Download Assignment4_data.csvinto Python.

      Please perform the following steps:

      1) Explore the datasets. (10 points)
      2) Find and handle missing values are in the data. (It is your choice how
you handle the missing data.) ( 20 points)
      3) Explore the variable column and Convert the "variable" column to dummy
variables and join the dummies to the data. (20 points)
      4) Convert the "one" column into 3 bins. (20 points)
```

In [3]:

```
#imports (may not need all of these but better safe than sorry later)
import os
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import csv
from numpy import nan as NA


#Prevent pandas from displying all of the DF
pd.options.display.max_rows = 10

#Read in a CSV file
#Set the path for the CSV file

readPath = "J:\DSDegree\PennState\DAN_862\Week 4\Homework"

#Change the directory
os.chdir(readPath)

#Read the CSV file in
data4 = pd.read_csv("Assignment4_data.csv")

print("The data frame to be used:\n")
print(data4)


#1) Explore the datasets. (10 points)
print("\n#1) Explore the datasets. (10 points)\n")


#print the first 5 rows
print("The data frame format:\n" + str(data4.head()))

#get the header info for later use
f = open('Assignment4_data.csv')
#Headers describing the data
h = list(csv.reader(f))[0]

#Check for duplicated data
dup = data4.duplicated()
#There are no duplicates in our data and without given more information as
#to what the data represents, we would not be dropping duplicates

print("\nStatisitcal data on the data frame:\n" + str(data4.describe()))
```

```
The data frame to be used:

      one   two  three  four  five variable
0   -92.0 -76.0  -33.0   3.0 -13.0       B2
1   -21.0  76.0   38.0  -6.0  80.0       B1
2    -2.0 -47.0  -34.0 -86.0 -66.0       A1
3   -76.0  43.0    7.0 -40.0 -42.0       A1
4    44.0  37.0   -7.0 -14.0  30.0       A1
..    ...   ...    ...   ...   ...      ...
195  63.0   3.0  -30.0 -24.0 -59.0       A1
196  97.0 -48.0  -61.0 -25.0 -21.0       B1
```

```
197 -93.0 -75.0  -18.0 -67.0 -58.0       B1
198  54.0 -66.0  -80.0  92.0  62.0       A1
199  82.0  53.0  -77.0  79.0  97.0       B2

[200 rows x 6 columns]

#1) Explore the datasets. (10 points)

The data frame format:
    one   two  three  four  five variable
0 -92.0 -76.0  -33.0   3.0 -13.0       B2
1 -21.0  76.0   38.0  -6.0  80.0       B1
2  -2.0 -47.0  -34.0 -86.0 -66.0       A1
3 -76.0  43.0    7.0 -40.0 -42.0       A1
4  44.0  37.0   -7.0 -14.0  30.0       A1

Statisitcal data on the data frame:
              one         two       three        four        five
count  195.000000  197.000000  199.000000  194.000000  196.000000
mean    -2.656410    2.208122    2.095477   -2.829897   -2.612245
std     67.489135   53.116759  101.864120   87.098996   84.158719
min   -363.000000 -100.000000 -100.000000 -576.000000 -821.000000
25%    -54.500000  -44.000000  -60.000000  -52.500000  -42.250000
50%      0.000000   -1.000000   -9.000000   -8.000000   -1.000000
75%     52.000000   45.000000   45.000000   44.000000   54.250000
max     97.000000   97.000000  832.000000  728.000000   99.000000
```

In [4]:
```
#2) Find and handle missing values are in the data. (It is your choice how you ha

print("\n\n#2) Find and handle missing values are in the data. (It is your choice

#The missing data values are read in a NaNs.
#I will fill them with the mean of each column
print("The missing data values are read in as NaNs.\nI will fill them with the me

#show the old data frame
print("Origional data frame: \n" + str(data4[6:9]))

#Fill the nan values with the means of the columns
#Only do this for the numeric columns (using the header) and ignore the categoric
data4 = data4.fillna(data4[h[0:5]].mean())

#Show the updated data frame
print("\nThe missing data has been filled with the column means: \n" + str(data4|
```

#2) Find and handle missing values are in the data. (It is your choice how you handle the missing data.) (20 points)

The missing data values are read in as NaNs.
I will fill them with the mean of each column.

Origional data frame:

	one	two	three	four	five	variable
6	41.0	0.0	-96.0	-9.0	87.0	B2
7	NaN	35.0	-51.0	75.0	93.0	A2
8	-39.0	-86.0	83.0	99.0	-20.0	B2

The missing data has been filled with the column means:

	one	two	three	four	five	variable
6	41.00000	0.0	-96.0	-9.0	87.0	B2
7	-2.65641	35.0	-51.0	75.0	93.0	A2
8	-39.00000	-86.0	83.0	99.0	-20.0	B2

In [5]:
```
#3) Explore the variable column and convert the "variable" column to dummy variab

print("\n\n\n#3) Explore the variable column and convert the variable column to d

#Get the variable dummy matrix
varDummies = pd.get_dummies(data4["variable"])
print("The dummy matrix:\n" + str(varDummies))

#Join the dummy matrix with the data table.
#Use the header values to indicate which values from data4 you'd like to keep
data4Dummies = data4[h[0:5]].join(varDummies)
print("\n\nThe new data frame with the dummy variables joined:\n" + str(data4Dumm
```

```
#3) Explore the variable column and convert the variable column to dummy variab
les and join the dummies to the data. (20 points)

The dummy matrix:
     A1  A2  B1  B2
0     0   0   0   1
1     0   0   1   0
2     1   0   0   0
3     1   0   0   0
4     1   0   0   0
..   ..  ..  ..  ..
195   1   0   0   0
196   0   0   1   0
197   0   0   1   0
198   1   0   0   0
199   0   0   0   1

[200 rows x 4 columns]


The new data frame with the dummy variables joined:
      one   two  three  four  five  A1  A2  B1  B2
0   -92.0 -76.0  -33.0   3.0 -13.0   0   0   0   1
1   -21.0  76.0   38.0  -6.0  80.0   0   0   1   0
2    -2.0 -47.0  -34.0 -86.0 -66.0   1   0   0   0
3   -76.0  43.0    7.0 -40.0 -42.0   1   0   0   0
4    44.0  37.0   -7.0 -14.0  30.0   1   0   0   0
..    ...   ...    ...   ...   ...  ..  ..  ..  ..
195  63.0   3.0  -30.0 -24.0 -59.0   1   0   0   0
196  97.0 -48.0  -61.0 -25.0 -21.0   0   0   1   0
197 -93.0 -75.0  -18.0 -67.0 -58.0   0   0   1   0
198  54.0 -66.0  -80.0  92.0  62.0   1   0   0   0
199  82.0  53.0  -77.0  79.0  97.0   0   0   0   1

[200 rows x 9 columns]
```

In [6]:
```
#4) Convert the "one" column into 3 bins. (20 points) (pg 203)

print("\n\n\n#4) Convert the 'one' column into 3 bins. (20 points)\n")


#Get the one column data as a list so it will be a Categorical object
oneData = list(data4["one"])
#set the bins
bins = [-400, 0, 53, 100]
#Cut (bin) the data into the Categroical object (my own bin sizes)
onesVals = pd.cut(oneData, bins)
#Get the count in each bin
binCount = pd.value_counts(onesVals)
print("The bin counts are:\n" + str(binCount))

#I can also do this with automated evenly spaced bins
onesValsSpaced = pd.cut(oneData, 3)
#Get the count in each bin
binCountSpaced = pd.value_counts(onesValsSpaced)
print("\nWhen evenly spaced bins, the bin counts are now:\n" + str(binCountSpaced

#I can also do this with even "quantiles" (Although they're are 3 not 4... so is
onesValsQuants = pd.qcut(oneData, 3)
binCountQuants = pd.value_counts(onesValsQuants)
print("\nWhen more evenly distributed bin distributions, the bin counts are now:\
```

```
#4) Convert the 'one' column into 3 bins. (20 points)

The bin counts are:
(-400, 0]    104
(0, 53]       49
(53, 100]     47
dtype: int64

When evenly spaced bins, the bin counts are now:
(-56.333, 97.0]        152
(-209.667, -56.333]     46
(-363.46, -209.667]      2
dtype: int64

When more evenly distributed bin distributions, the bin counts are now:
(-363.001, -32.667]    67
(29.667, 97.0]         67
(-32.667, 29.667]      66
dtype: int64
```

In [7]:
```
"""

Use the following speech by the Rev. Dr. Martin Luther King, Jr:

s = “I am happy to join with you today in what will go down in history as the gre

1) Find out how many unique words in s. (10 points)
2) Which word appears the most? (10 points)
3) How many words start with ‘t’. (10 points).


"""


print("""
      \n\nQuestion 2:\n\nUse the following speech by the Rev. Dr. Martin Luther K

      s = “I am happy to join with you today in what will go down in history as t

      1) Find out how many unique words in s. (10 points)
      2) Which word appears the most? (10 points)
      3) How many words start with ‘t’. (10 points).
      """)
```

Question 2:

Use the following speech by the Rev. Dr. Martin Luther King, Jr:

s = “I am happy to join with you today in what will go down in history as the greatest demonstration for freedom in the history of our nation. Five score years ago, a great American, in whose symbolic shadow we stand today, signed th e Emancipation Proclamation. This momentous decree came as a great beacon light of hope to millions of Negro slaves who had been seared in the flames of wither ing injustice. It came as a joyous daybreak to end the long night of their capt ivity. But one hundred years later, the Negro still is not free. One hundred ye ars later, the life of the Negro is still sadly crippled by the manacles of seg regation and the chains of discrimination. One hundred years later, the Negro l ives on a lonely island of poverty in the midst of a vast ocean of material pro sperity. One hundred years later, the Negro is still languishing in the corners of American society and finds himself an exile in his own land. So we have come here today to dramatize a shameful condition."

1) Find out how many unique words in s. (10 points)
2) Which word appears the most? (10 points)
3) How many words start with ‘t’. (10 points).

In [8]:
```
s = "I am happy to join with you today in what will go down in history as the gre
```

In [9]:
```
#1) Find out how many unique words in s. (10 points)
print("\n\n1) Find out how many unique words are in s. (10 points)")

#Clean out commas and period punctuation while not adding extra spaces
s = s.replace(',', '')
s = s.replace('.', '')

#I am going to include words that are capitalized as the same word
#as their uncapitalized counterparts.  ie.  It == it

#Convert all words to lower case
s = s.lower()

#Split by spaces and strip the whitespace
words = [i.strip() for i in s.split(' ')]

#Find the unique words.  Numpy does this and sorts alphabetically
uniqueWords = np.unique(words)

#The number of unique words
uniqueCount = len(uniqueWords)

print("\nThe number of unique words in the Rev. Dr.'s speech is: " + str(uniqueCo
```

```
1) Find out how many unique words are in s. (10 points)

The number of unique words in the Rev. Dr.'s speech is: 107
```

In [10]:
```
#2) Which word appears the most? (10 points)
print("\n\n2) Which word appears the most? (10 points)\n")

#Create an 'empty' array
wordCount = np.empty(uniqueCount)

#fill the 'empty' array with the number of counts each word appears
for i in range(0, len(uniqueWords)):
    wordCount[i] = s.count(' '+ uniqueWords[i] + ' ')

#Join the two arrays into a dataframe
wordData = pd.DataFrame({'Word': uniqueWords,
                         'Count': wordCount})

#Find the location (index) of the maximum count (most used word)
loc = wordData['Count'].idxmax()

#Most common word statistics
commWord = wordData.iloc[loc]

#Print out the information
print("The most common word is '" + str(commWord[0]) + "'.")
print("'" + str(commWord[0]) + "' is used " + str(commWord[1]) + " times within t
```

```
2) Which word appears the most? (10 points)

The most common word is 'the'.
'the' is used 14.0 times within the speach.
```

In [11]:
```
#3) How many words start with ‘t’. (10 points).
print("\n\n#3) How many words start with ‘t’. (10 points).\n")

#Create a new column for starting with a 't'
wordData.insert(2,'t start', np.zeros(uniqueCount))

#Assign True/False values to the 't start' column based on if the word starts wit
wordData['t start'] = wordData['Word'].str.startswith('t')

#Sum the 't start' column to get the total number of words that begin with 't'
tCount = wordData['t start'].sum()

#Print the information
print("The number of words that begin with the letter 't' is: " + str(tCount))
```

```
#3) How many words start with ‘t’. (10 points).

The number of words that begin with the letter 't' is: 5
```