Jeremy Diamond PA2

For PA2 we were asked to build a python program to analyze data about the marvel comics universe. I used python 3 installed via anaconda my methodology was to read in the list of comics and characters as 1D arrays than generate 2D arrays that contained the data on collaboration. I then used the 2D arrays to generate the data to be analyzed. For testing purposes the a function called outputTest was written. It writes the content of a 2D array to a csv.

**Reading**

Each read block is essentially the same. They specify a descriptive variable name for each entry in a row and use an if statement to insert the data into its proper location. When reading in nodes.csv a 0 is placed at the beginning of each list so that later 2D arrays have no overlapping names. Hero-network.csv contains a number of special case typos that had t be dealt with upon reading the file. Some were simply edited in the data.

**2D Array processing**

The 2 2D arrays are an array of all character crossovers called collabArray and an array of all character appearances in a book called inBookArray. In each a 1 was inserted for entries in edges.csv and hero-network.csv respectively. Outputting these arrays created the largest spreadsheets I’ve ever seen. Excel had trouble loading them but I managed to color each 1 in red and screenshot dense areas in them (These images are horizontally squished because I use an 21/9 monitor)

**Solving for the desired values**

To solve for the assigned values I simply summed the relevant data from the 2D arrays. For the number of characters per book a made a 1D array and summoned the coulmbs of inBookArray. For the number of appearances for a character I summed the rows and for the number of collaborations I could have done either dimension in collabArray ( I did the coulmbs for the record). I then sorted these 1D arrays (which was super slow I want to look into the Array.sort() function) to find min and max. I then found the mean for these things. Here are my sample results

reading nodes.csv

reading edges.csv

reading hero-network.csv

The number of comics is: 12652

The number of characters is: 6440

The max, mean, and min number of characters per book are :1, 7.596553632123943, and 111

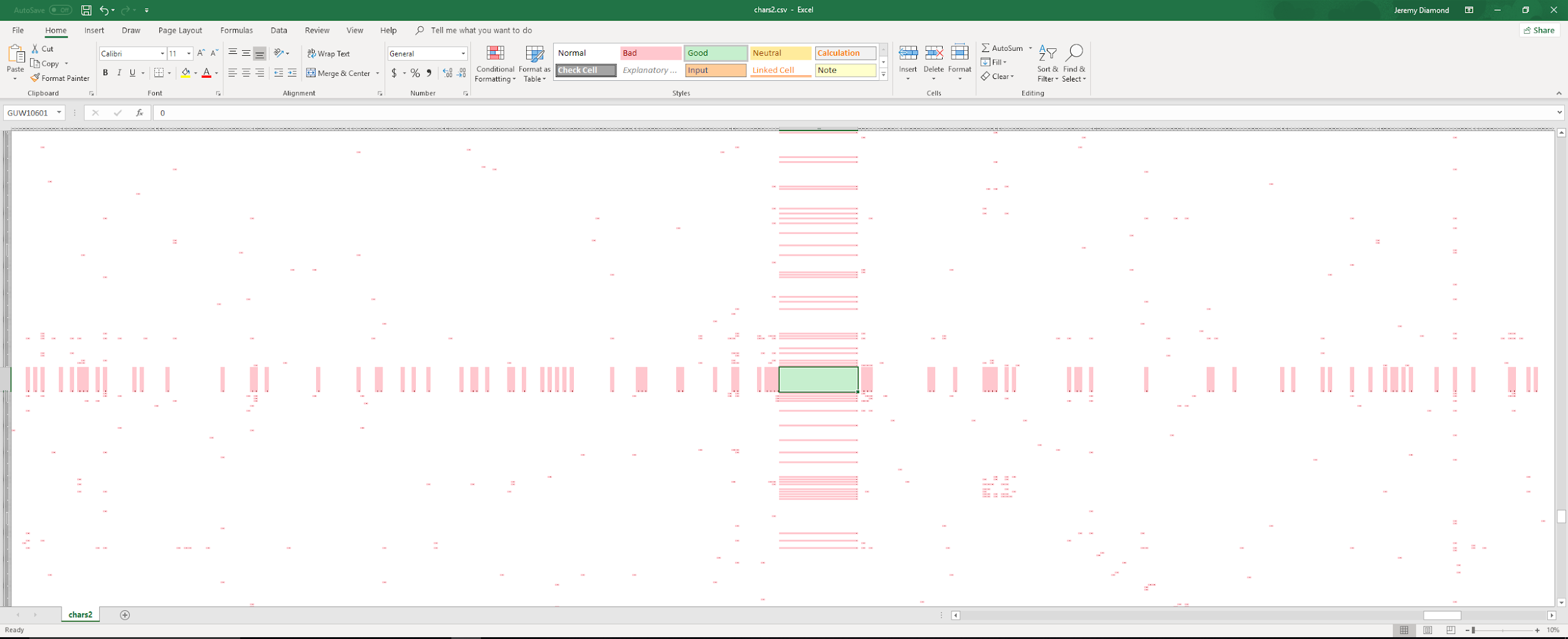
The max, mean, and min number of book appearances per character are :1, 14.925298959465756, and 1577

The max, mean, and min number of collaborators per character are :0, 34.80524926230781, and 1428

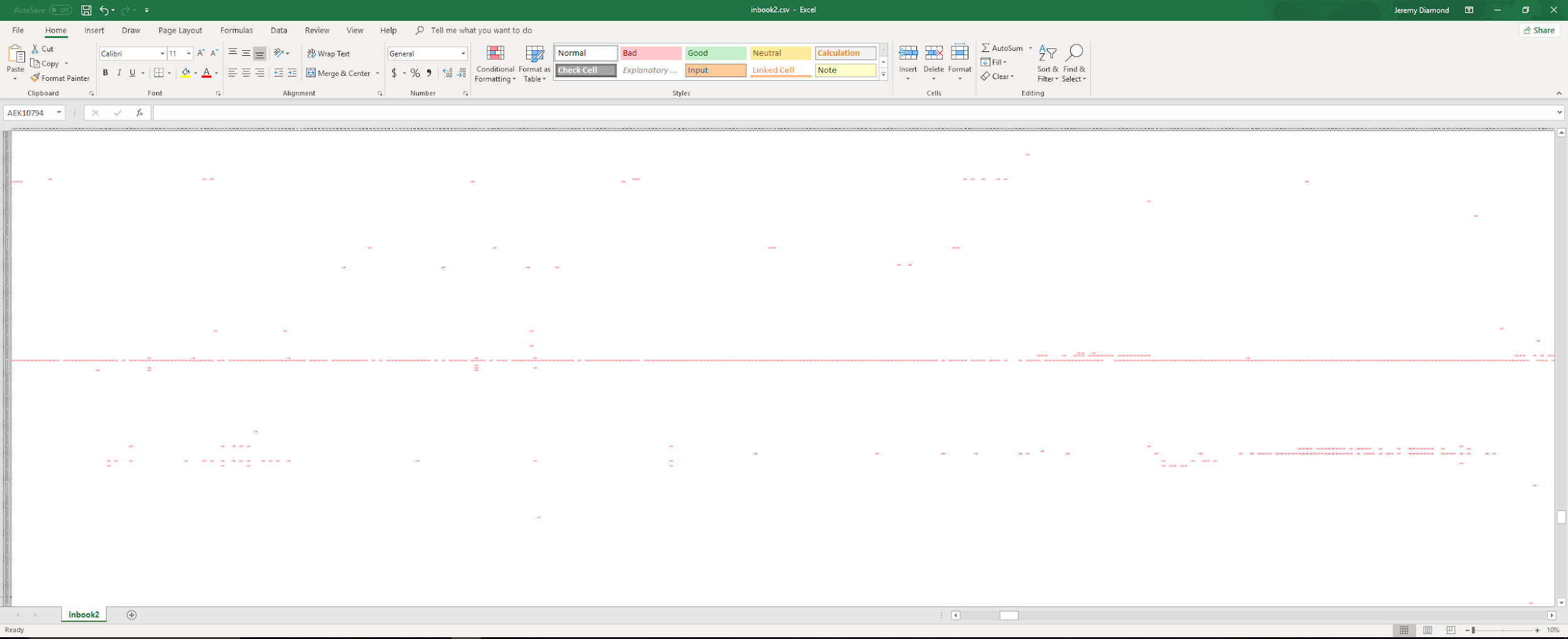
**Conclusions**

This project pushed my python skills in a welcome way and pushed my understanding of data processing.

collabArray around the intersection of Spiderman and Spiderman (the cell in green)



inBook array around the intersection of Spiderman and the issues of Amazing Spiderman (the comic)



Project2.py

import csv

def outputTest(name, testArray = []):

#testing output

with open(name + ".csv","w+") as my\_csv:

csvWriter = csv.writer(my\_csv,delimiter=',')

csvWriter.writerows(testArray)

comics = []

characters = []

#read in data from nodes.csv

with open('nodes.csv') as csvfile:

readCSV = csv.reader(csvfile, delimiter=',')

for row in readCSV:

name = row[0]

typeOf = row[1]

if name == "node":

print ("reading nodes.csv")

comics.append(0)

characters.append(0)

elif typeOf == 'comic':

comics.append(name)

else:

characters.append(name)

# build arrays for collabs and book apperanaces

collabArray = [[0] \* len(characters) for i in range(len(characters))]

for x in range(len(characters)):

collabArray[0][x] = characters[x]

collabArray[x][0] = characters[x]

inBookArray = [[0] \* len(comics) for i in range(len(characters))]

for x in range(len(characters)):

y = x+1

inBookArray[x][0] = characters[x]

for x in range(len(comics)):

inBookArray[0][x] = comics[x]

# outputTest("collab",collabArray)

# outputTest("inbook",inBookArray)

#populate the arrays

#populate inBookArray

with open('edges.csv') as csvfile:

readCSV = csv.reader(csvfile, delimiter=',')

for row in readCSV:

hero = row[0]

comic = row[1]

if hero == "hero":

print ("reading edges.csv")

else:

y = characters.index(hero)

x = comics.index(comic)

inBookArray[y][x] = 1

#outputTest("inbook2",inBookArray)

count = -1

#populate collaberate array

with open('hero-network.csv') as csvfile:

readCSV = csv.reader(csvfile, delimiter=',')

for row in readCSV:

count = count+1

hero1 = row[0]

hero2 = row[1]

if hero1.endswith(" "):

hero1 = hero1[:-1]

if hero2.endswith(" "):

hero2 = hero2[:-1]

if hero1 == "hero1":

print ("reading hero-network.csv")

else:

y = characters.index(hero1)

x = characters.index(hero2)

collabArray[y][x] = 1

#outputTest("chars2",collabArray)

# evaluate the max min and mean arrays and print results

print ("The number of comics is: " + str(len(comics)))

print ("The number of charaters is: " + str(len(characters)))

CharsPerBook = []

for x in range(1, len(comics)):

sum = 0

for i in range(1, len(characters)):

sum = sum + inBookArray[i][x]

CharsPerBook.append(sum)

CharsPerBook.sort()

minim = CharsPerBook[0]

maxim = CharsPerBook[len(CharsPerBook)-1]

mean = 0

for x in range(len(CharsPerBook)):

mean = mean + CharsPerBook[x]

mean = mean/len(CharsPerBook)

print ("The max, mean, and min number of charaters per book are :" + str(minim) + ", " + str(mean) + ", and " + str(maxim))

BooksPerChar = []

for y in range(1, len(characters)):

sum = 0

for i in range(1, len(comics)):

sum = sum + inBookArray[y][i]

BooksPerChar.append(sum)

BooksPerChar.sort()

minim = BooksPerChar[0]

maxim = BooksPerChar[len(BooksPerChar)-1]

mean = 0

for x in range(len(BooksPerChar)):

mean = mean + BooksPerChar[x]

mean = mean/len(BooksPerChar)

print ("The max, mean, and min number of book apperances per charater are :" + str(minim) + ", " + str(mean) + ", and " + str(maxim))

CollabsPerChar = []

for x in range(1, len(characters)):

sum = 0

for i in range(1, len(characters)):

sum = sum + collabArray[i][x]

CollabsPerChar.append(sum)

CollabsPerChar.sort()

minim = CollabsPerChar[0]

maxim = CollabsPerChar[len(CollabsPerChar)-1]

mean = 0

for x in range(len(CollabsPerChar)):

mean = mean + CollabsPerChar[x]

mean = mean/len(CollabsPerChar)

print ("The max, mean, and min number of collaborators per charater are :" + str(minim) + ", " + str(mean) + ", and " + str(maxim))