Shepherd
CS 654
Homework 2

1. Imported my data as a dataframe and then converted the columns of interest to lists. I chose to use bmi and blood pressure for the quantitative attributes, region for the categorical, and smoker for the Boolean attribute.

```
In [28]: df = pd.read_csv("cs654_homework2_dataset_shepherd.csv")
```

```
In [29]: df.head()
```

Out[29]:

	index	gender	bmi	bloodpressure	smoker	region
0	0	male	23.2	91	No	southeast
1	1	male	30.1	87	No	southeast
2	2	male	33.3	82	No	southeast
3	3	male	33.7	80	No	northwest
4	4	male	34.1	100	No	northwest

```
In [30]: #convert columns to parallel lists
         #2 quantitative
         bp_list = df['bloodpressure'].tolist()
         bmi_list = df['bmi'].tolist()
         #1 binary
         sm_list = df['smoker'].tolist()
         # 1 categorical
         reg_list = df['region'].tolist()
```

2. Next, I created a function to normalize the quantitative attributes and another to compute the distances between each entry in the list.

```
: #normalize quantitative data
  def norm_data(data_list, mini, maxi):
      norm_list = []
      for x in data_list:
          norm_list.append((x-mini)/(maxi - mini))
      return norm_list
```

```
: bmi_list = norm_data(bmi_list, min(bmi_list), max(bmi_list))
  bp_list = norm_data(bp_list, min(bp_list), max(bp_list))
```

```
35]: import math
     def comp_dist_matrix(data_list):
         dist_matrix = []
         for i in range(len(data_list)):
             cur_row = []
             for j in range(len(data_list)):
                 cur_row.append(math.fabs(data_list[i]-data_list[j]))
             dist_matrix.append(cur_row)
         return dist_matrix
```

```
36]: bmi_dif = comp_dist_matrix(bmi_list)
     bp_dif = comp_dist_matrix(bp_list)
```

bmi_dif and bp_df are both lists of lists. This is the reason we need nested loops to create them and then traverse them.

3. I created another function that constructs the difference matrix for categorical attributes. Since Boolean attributes are a type of categorical, I used this function to create the distance matrices for both region and smoker.

```
In [53]: # calculate difference matrix for categorical data
         #also works for Boolean attribute
         def comp_dif_cat(data_list):
             dist_matrix = []
             for i in range(len(data_list)):
                 cur_row = []
                 for j in range(len(data_list)):
                     if data_list[i] == data_list[j]:
                         cur_row.append(0)
                     else:
                         cur_row.append(1)
                 dist_matrix.append(cur_row)
             return dist_matrix
```

```
In [55]: sm_dif = comp_dif_cat(sm_list)
         reg_dif = comp_dif_cat(reg_list)
         print(sm_dif[2][2])
         print(reg_dif[2][2])
```

```
0
0
```

4. I then wrote a function that combined all four distance matrices into one. I weighted each of the four attributes equally.

```
In [39]: def comp_final_dist(list1, list2, list3, list4):
             dist_matrix = []
             for i in range(len(list1)):
                 cur_row = []
                 for j in range(len(list1)):
                     d = round((list1[i][j] + list2[i][j] + list3[i][j] + list4[i][j
                     cur_row.append(d)
                 dist_matrix.append(cur_row)
             return dist_matrix

In [40]: final_matrix = comp_final_dist(bp_dif, bmi_dif, sm_dif, reg_dif)
```

5. Lastly, I wrote a function that asked the user to enter the indexes of the patients that they wanted to find the differences between. The function prints out the findings and also returns the difference.

```
In [41]: def calc_dist():
             i = int(input("Enter the index of the first patient: 0-99. "))
             j = int(input("Enter the index of the second patient: 0-99. "))
             print("The distance between the two patients is ", final_matrix[i][j])
             return(final_matrix[i][j])
```

Results:

Trial 1:

```
             return(final_matrix[i][j])

In [60]: calc_dist()

         Enter the index of the first patient: 0-99. 4
         Enter the index of the second patient: 0-99. 90
         The distance between the two patients is  0.31

Out[60]: 0.31
```

Trial 2:

```
In [61]: calc_dist()

         Enter the index of the first patient: 0-99. 67
         click to scroll output; double click to hide
                                    the second patient: 0-99. 33
         The distance between the two patients is  0.38

Out[61]: 0.38
```

```
In [59]: final_matrix[0][0]
```

Trial 3:

```
    print("The distance between the two patients is ", final_matrix[i][j])
    return(final_matrix[i][j])
```

In [62]:
```
calc_dist()
```

```
Enter the index of the first patient: 0-99. 42
Enter the index of the second patient: 0-99. 42
The distance between the two patients is  0.0
```

Out[62]:
```
0.0
```