Task 1:
Search for bmi (quantitative variable):

In [1]:
```
import pandas as pd

df_c = pd.read_csv("/Users/burrisfaculty/Desktop/DSCode/CS621/cs621_homework_3_data_complete.csv")
```

In [2]:
```
#Check to make sure it loaded
df_c.head(3)
```

Out[2]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
2	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1

In [19]:
```
#Search feature of bmi-- numerical column
def bmi_search():
    bmi_list = list(df_c["bmi"])
    target = float(input("Enter the BMI to search for "))
    num_found = 0
    for bmi in bmi_list:
        if target == bmi:
            num_found += 1
    if num_found == 0:
        print("No records found")
    else:
        print("Yes, there are ", num_found, " records matching that search")
```

In [20]:
```
bmi_search()
```

```
Enter the BMI to search for 32.1
No records found
```

In [21]:
```
bmi_search()
```

```
Enter the BMI to search for 36.6
Yes, there are  1  records matching that search
```

In the code, the “bmi” column of the data frame is converted to a list. The user is asked to enter a bmi to search for and their response is cast to a float. A for-loop is then used to traverse the list. When the target matches a bmi-value in the list, the num_found variable is increased by 1. Once the entire list has been searched, the results are reported to the user. If num_found equals 0, the program reports “No records found.” Otherwise, the number of records found are reported.

In [25]:
```
def smoker_search():
    smoker_list = list(df_c["smoking_status"])
    target = input("Enter the smoking status to be searched. ")
    num_found = 0
    for smoke in smoker_list:
        if target == smoke:
            num_found += 1
    if num_found == 0:
        print("No records found")
    else:
        print("Yes, there are ", num_found, " records matching that search")
```

In [27]:
```
smoker_search()
smoker_search()
```

```
Enter the smoking status to be searched. lkdsajfl;askdj
No records found
Enter the smoking status to be searched. formerly smoked
Yes, there are  27  records matching that search
```

In the code, the “smoking status” column of the data frame is converted to a list. The user is asked to enter a smoking status in which to search. A for-loop is then used to traverse the list. When the target matches smoking status in the list, the num_found variable is increased by 1.

Once the entire list has been searched, the results are reported to the user. If num_found equals 0, the program reports “No records found.” Otherwise, the number of records found are reported.

Task 2:

```
In [15]: #Part 2
df_dup = pd.read_csv("/Users/burrisfaculty/Desktop/DSCode/CS621/cs621_homework_3_data_duplicate.csv")
```

```
In [16]: is_duplicated = df_dup.duplicated()
duplicates = []
non_duplicates = []
for i in range(len(is_duplicated)):
    if (is_duplicated[i]):
        duplicates.append(i)
    else:
        non_duplicates.append(i)
print("Duplicate records: ", len(duplicates))

print(duplicates)
print(non_duplicates)
```

```
Duplicate records:  3
[27, 90, 100]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 3
2, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 6
1, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 9
1, 92, 93, 94, 95, 96, 97, 98, 99, 101, 102, 103]
```

First, the program calls the **.duplicated()** function from the pandas module. It returns a list of Booleans that the program saves to the list **is_duplicated**. False is returned if the record is a duplicate of a previous record. A for-loop is used to traverse the list **is_duplicated**. If the current value is True, the index is saved to the list **non_duplicates**. If the current value is False, the index is saved to the list **duplicates.** The program then reports the length of the list **duplicates**, that is equal to the number of duplicate records. In my file, you can see the duplicate records are at indices 27, 90, and 100.

Task 3:

```
In [17]: #Part 3 -- Missing Data
df_miss = pd.read_csv("/Users/burrisfaculty/Desktop/DSCode/CS621/cs621_homework_3_data_missing.csv")
```

```
In [28]: num_missing = 0
for i in range(len(df_miss)):
    for j in range(len(df_miss.columns)):
        if (pd.isnull(df_miss.iloc[i, j])):
            num_missing += 1
            print(i,j)
if (num_missing > 0):
    print("Missing Values: ", num_missing)
else:
    print("No missing values.")
```

```
20 5
31 9
46 7
47 1
63 3
66 7
Missing Values:  6
```

```
In [ ]:
```

This code uses nested for loops to traverse the entire data frame. For each cell in the data frame, it calls the method **.isnull()** from the pandas module. If **.isnull()** returns True, then the value of the variable **num_missing** is increased by 1. I also had it print out the row and column

of the missing data. If **num_missing** is greater than 0, the program informs the user how many values are missing. Otherwise, it informs the user that the data is complete.

From the output, I can see that I have the following missing data:

Record Index	**Column Index**
20	5 (work_type)
31	9 (smoking_status)
46	7 (ave_glucose_level)
47	1 (age)
63	3 (heart_disease)
66	7 (ave_glucose_level)