In [11]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
```

In [162…
```
df = pd.read_csv('human_trafficking.csv')
ad = pd.read_csv('healthcare-dataset-stroke-data.csv')
ad
```

```
C:\Users\coold\AppData\Local\Temp\ipykernel_21312\1092470314.py:1: DtypeWarning: Columns (54) have mixed types.
Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('human_trafficking.csv')
```

Out[162]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked
...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown

5110 rows × 12 columns

In [137…
```
## Now we will clean the data by finding the where null values are
nulls=ad.isnull().sum()

print(ffd)
```

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

In [139…
```
## now we wil display the null values
nana_df = ad[ad.isna().any(axis=1)]
nana_df
```

Out[139]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked
8	27419	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown
13	8213	Male	78.0	0	1	Yes	Private	Urban	219.84	NaN	Unknown
19	25226	Male	57.0	0	1	No	Govt_job	Urban	217.08	NaN	Unknown
27	61843	Male	58.0	0	0	Yes	Private	Rural	189.84	NaN	Unknown
...	...	...	...	...	...	...	...	...	...	...	...
5039	42007	Male	41.0	0	0	No	Private	Rural	70.15	NaN	formerly smoked
5048	28788	Male	40.0	0	0	Yes	Private	Urban	191.15	NaN	smokes
5093	32235	Female	45.0	1	0	Yes	Govt_job	Rural	95.02	NaN	smokes
5099	7293	Male	40.0	0	0	Yes	Private	Rural	83.94	NaN	smokes
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked

201 rows × 12 columns

In [163…
```
## here is another way

ad[ad['bmi'].isnull()]
```

Out[163]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked
8	27419	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown
13	8213	Male	78.0	0	1	Yes	Private	Urban	219.84	NaN	Unknown
19	25226	Male	57.0	0	1	No	Govt_job	Urban	217.08	NaN	Unknown
27	61843	Male	58.0	0	0	Yes	Private	Rural	189.84	NaN	Unknown
...	...	...	...	...	...	...	...	...	...	...	...
5039	42007	Male	41.0	0	0	No	Private	Rural	70.15	NaN	formerly smoked
5048	28788	Male	40.0	0	0	Yes	Private	Urban	191.15	NaN	smokes
5093	32235	Female	45.0	1	0	Yes	Govt_job	Rural	95.02	NaN	smokes
5099	7293	Male	40.0	0	0	Yes	Private	Rural	83.94	NaN	smokes
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked

201 rows × 12 columns

In [164…
```
## Here we will replace the null values with the average valuw of bmi
test2 = ad.fillna(np.mean(ad['bmi']))
test2
```

Out[164]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_s
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly sm
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never sm
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never sm
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	sm
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never sm
...	...	...	...	...	...	...	...	...	...	...	
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	28.893237	never sm
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	never sm
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.600000	never sm
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	formerly sm
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.200000	Unk

5110 rows × 12 columns

In [169…
```
## now here we are dropping all values that have null values a different way of dealing with nulls
test3=ad.dropna()
test3
```

Out[169]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked
5	56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked
...	...	...	...	...	...	...	...	...	...	...	...
5104	14180	Female	13.0	0	0	No	children	Rural	103.08	18.6	Unknown
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown

4909 rows × 12 columns

In [193…
```
##Now lastly we want to drop any duplicates if we have any
finaldf = test3.drop_duplicates()
finaldf
```

Out[193]:

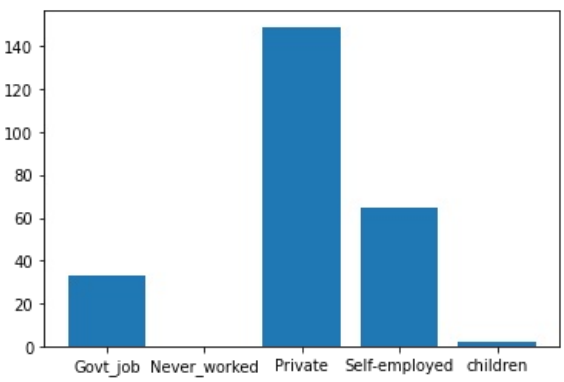
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked
5	56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked
...	...	...	...	...	...	...	...	...	...	...	...
5104	14180	Female	13.0	0	0	No	children	Rural	103.08	18.6	Unknown
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown

4909 rows × 12 columns

In [ ]:

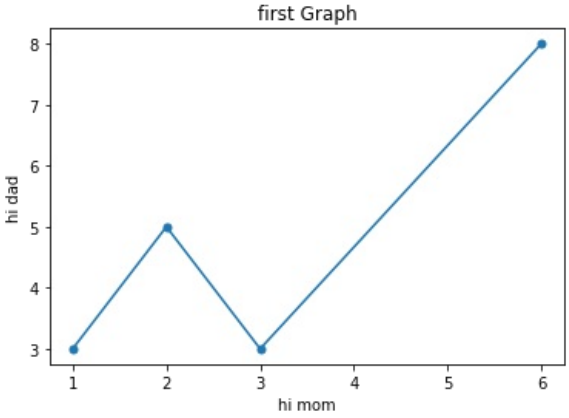
In [176…
```
## how to create a bar chart first create a key like this
## now use this forumula listed below to create the chart the keys is the x and the ad.groupby is the y
keys = [work_type for work_type, df in test3.groupby(['work_type'])]

plt.bar(keys,ad.groupby(['work_type']).sum()['stroke'])
```

Out[176]:
```
<BarContainer object of 5 artists>
```



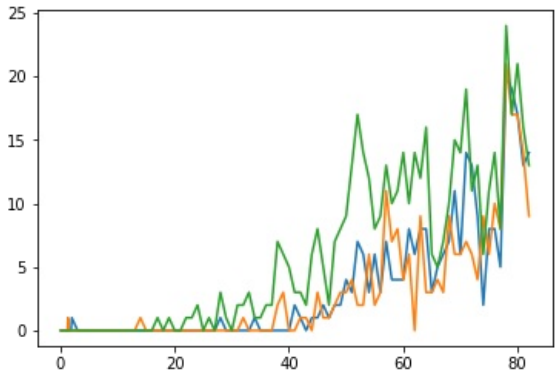
In [194…
```
x=[1,2,3,6]
y=[3.,5,3,8]
plt.plot(x,y,marker= '.',markersize=10)
```

```
plt.title('first Graph')
plt.xlabel('hi mom')
plt.ylabel('hi dad')
plt.show()
```



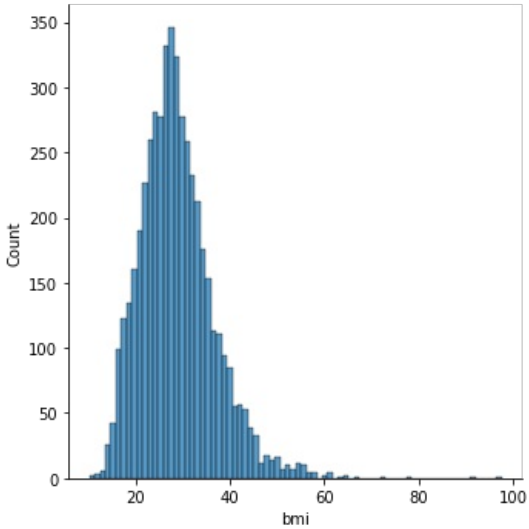
In [104…
```
## now we want to see the correlation of heart diease stroke and hypertenison as you increase in age
keys = [age for age, df in ad.groupby(['age'])]

plt.plot(keys,ad.groupby(['age']).sum()['heart_disease'])
plt.plot(keys,ad.groupby(['age']).sum()['stroke'])
plt.plot(keys,ad.groupby(['age']).sum()['hypertension'])
```

Out[104]:
```
[<matplotlib.lines.Line2D at 0x17b4be60c10>]
```



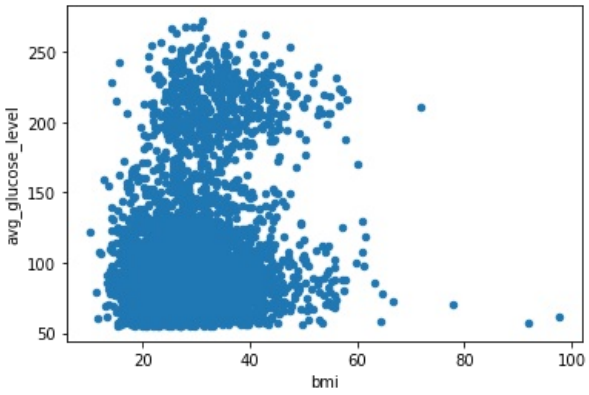
In [109…
```
## Here I want to create a histogram for bmi as well as find attributes for this variable
sns.displot(ad['bmi'])
ad['bmi'].describe()
```

Out[109]:
```
count    4909.000000
mean       28.893237
std         7.854067
min        10.300000
25%        23.500000
50%        28.100000
75%        33.100000
max        97.600000
Name: bmi, dtype: float64
```



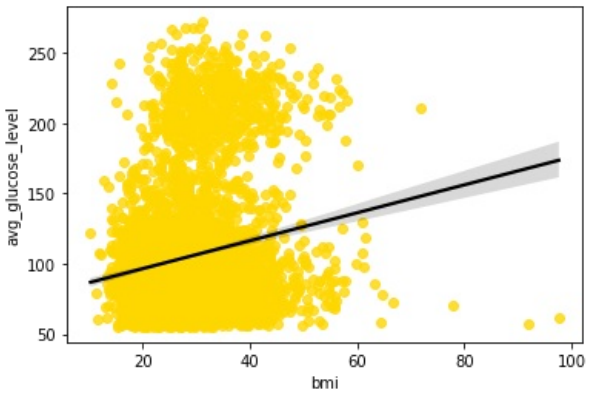
In [177…
```
## now I am creating some graphs

test3.plot.scatter(x="bmi", y= 'avg_glucose_levelavg_glucose_level')
```

Out[177]: <AxesSubplot:xlabel='bmi', ylabel='avg_glucose_level'>

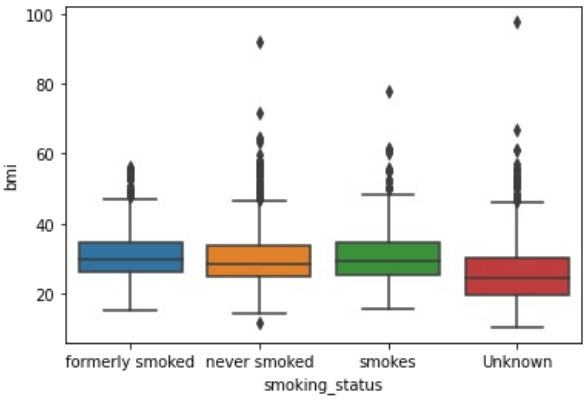


In [201…
```
##Here is another way to create this graph using the linear regression model
sns.regplot(x='bmi', y='avg_glucose_level', data=test3, scatter_kws={'color': 'gold'}, line_kws={'color':'black
```

Out[201]: <AxesSubplot:xlabel='bmi', ylabel='avg_glucose_level'>

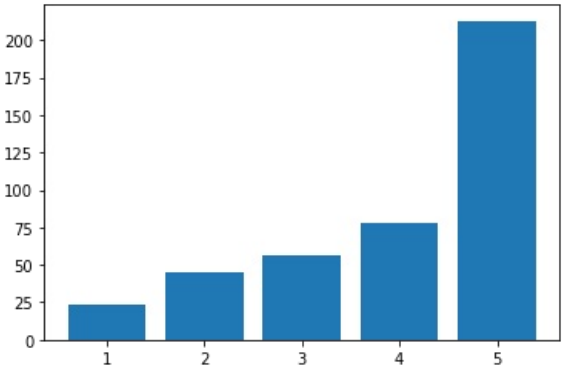


In [27]:
```
### I will create a boxplot on smoking status
fig = sns.boxplot(x='smoking_status', y='bmi', data=ad)
```



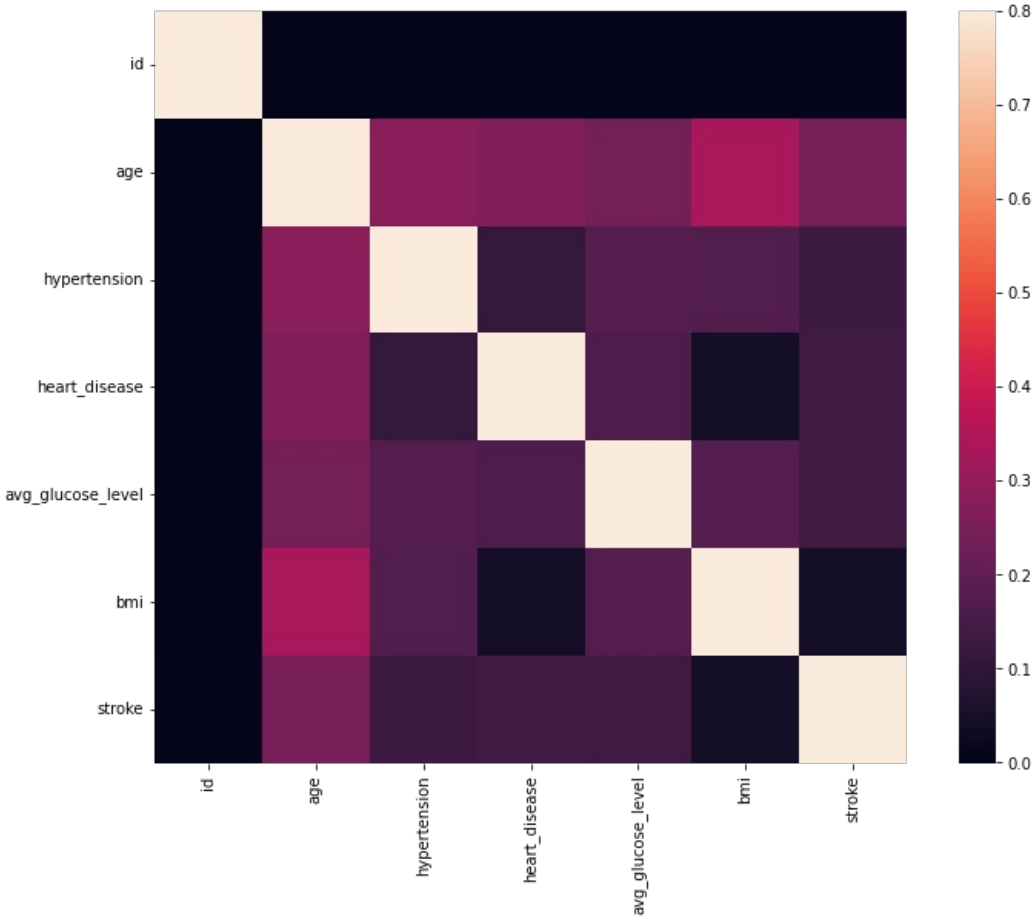
In [58]:
```
data = [23,45,56,78,213]
plt.bar([1,2,3,4,5,], data)
```

Out[58]: <BarContainer object of 5 artists>



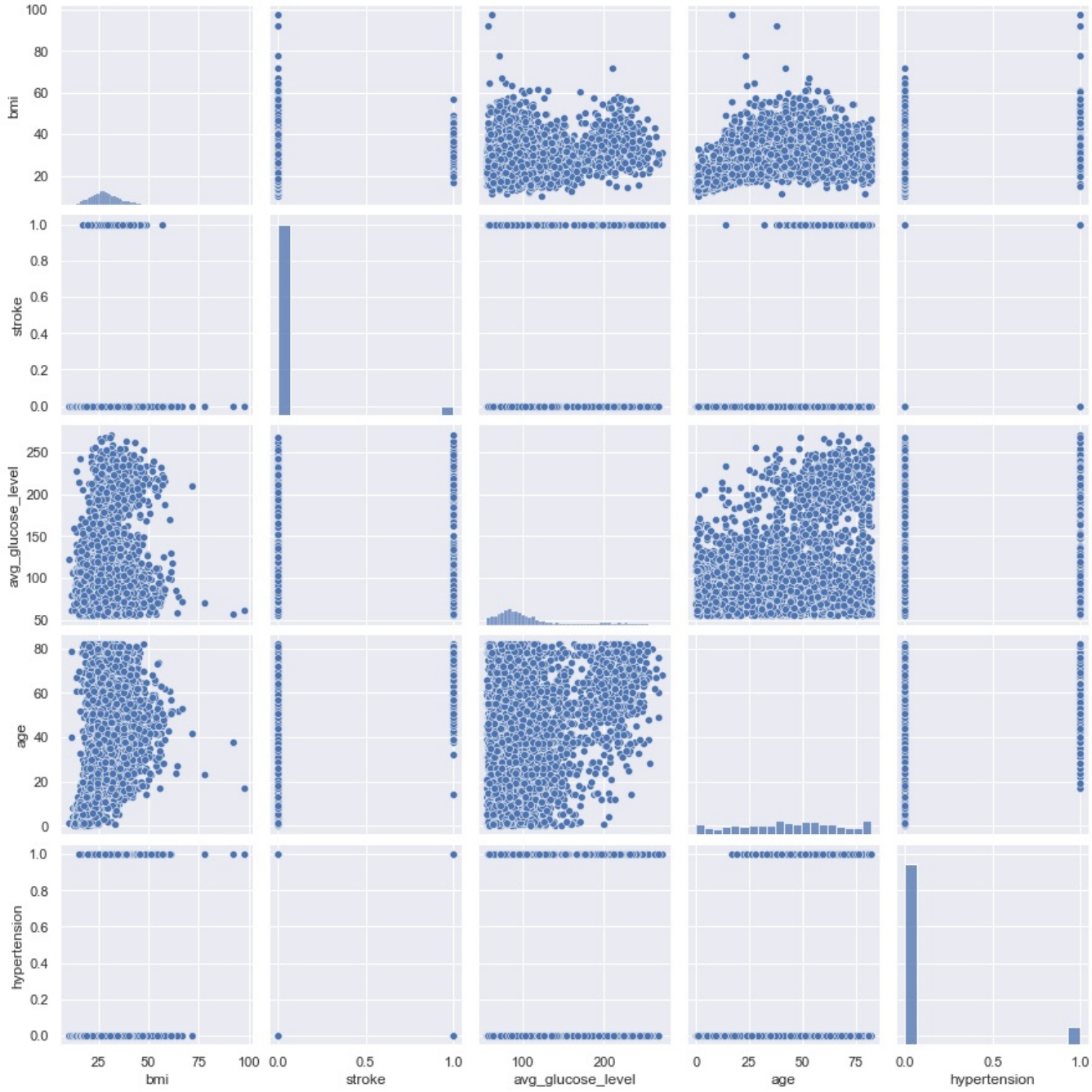
In [51]:
```
## Now I will run a correlation  matrix to see which variables effect each other the most
corr = ad.corr()
corrmat = ad.corr()
f, ax =plt.subplots(figsize=(12,9))
sns.heatmap(corrmat,vmax=.8, square=True)
```

Out[51]: <AxesSubplot:>



In [203…

```
sns.set()
cols = ['bmi','stroke', 'avg_glucose_level', 'age', 'hypertension']
sns.pairplot(test3[cols], height=2.5)
plt.show()
```



In [175…

```
## Lastly we will check to see which residence types have the highest glucose levels
ad.groupby("Residence_type").mean().sort_values("avg_glucose_level",ascending=False)
```

Out[175]:

Residence_type	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
Rural	36547.998011	42.900811	0.099841	0.053302	106.375235	28.894212	0.045346
Urban	36488.613636	43.542126	0.095146	0.054700	105.927307	28.892289	0.052003