

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as plt

health1 = pd.read_csv('C:\\Users\\HP\\Documents\\healthcare-dataset-
stroke-data1.csv')
```

```
health1
```

	sex	age	hyper_tension	heart_condition	maritital_status	\
0	Male	67.0	0	1	Yes	
1	Female	61.0	0	0	Yes	
2	Male	80.0	0	1	Yes	
3	Female	49.0	0	0	Yes	
4	Female	79.0	1	0	Yes	
...	
5105	Female	80.0	1	0	Yes	
5106	Female	81.0	0	0	Yes	
5107	Female	35.0	0	0	Yes	
5108	Male	51.0	0	0	Yes	
5109	Female	44.0	0	0	Yes	

	working_type	residential_area	glucose_level	body_mass_index	\
0	Private	Urban	228.69	36.600000	
1	Self-employed	Rural	202.21	28.893237	
2	Private	Rural	105.92	32.500000	
3	Private	Urban	171.23	34.400000	
4	Self-employed	Rural	174.12	24.000000	
...	
5105	Private	Urban	83.75	28.893237	
5106	Self-employed	Urban	125.20	40.000000	
5107	Self-employed	Rural	82.99	30.600000	
5108	Private	Rural	166.29	25.600000	
5109	Govt_job	Urban	85.28	26.200000	

	smoking_condition	stroke
0	formerly smoked	1
1	never smoked	1
2	never smoked	1
3	smokes	1
4	never smoked	1
...
5105	never smoked	0
5106	never smoked	0
5107	never smoked	0
5108	formerly smoked	0
5109	Unknown	0

```
[5110 rows x 11 columns]
```

```
health1.describe()
```

	age	hyper_tension	heart_condition	glucose_level \
count	5110.000000	5110.000000	5110.000000	5110.000000
mean	43.226614	0.097456	0.054012	106.147677
std	22.612647	0.296607	0.226063	45.283560
min	0.080000	0.000000	0.000000	55.120000
25%	25.000000	0.000000	0.000000	77.245000
50%	45.000000	0.000000	0.000000	91.885000
75%	61.000000	0.000000	0.000000	114.090000
max	82.000000	1.000000	1.000000	271.740000

	body_mass_index	stroke
count	5110.000000	5110.000000
mean	28.893237	0.048728
std	7.698018	0.215320
min	10.300000	0.000000
25%	23.800000	0.000000
50%	28.400000	0.000000
75%	32.800000	0.000000
max	97.600000	1.000000

```
health1.drop('id', inplace=True, axis=1)
```

```
health1
```

	sex	age	hyper_tension	heart_condition	marital_status \
0	Male	67.0	0	1	Yes
1	Female	61.0	0	0	Yes
2	Male	80.0	0	1	Yes
3	Female	49.0	0	0	Yes
4	Female	79.0	1	0	Yes
...
5105	Female	80.0	1	0	Yes
5106	Female	81.0	0	0	Yes
5107	Female	35.0	0	0	Yes
5108	Male	51.0	0	0	Yes
5109	Female	44.0	0	0	Yes

	working_type	residential_area	glucose_level	body_mass_index \
0	Private	Urban	228.69	36.6
1	Self-employed	Rural	202.21	NaN
2	Private	Rural	105.92	32.5
3	Private	Urban	171.23	34.4
4	Self-employed	Rural	174.12	24.0
...
5105	Private	Urban	83.75	NaN
5106	Self-employed	Urban	125.20	40.0
5107	Self-employed	Rural	82.99	30.6
5108	Private	Rural	166.29	25.6

5109	Govt_job	Urban	85.28	26.2
------	----------	-------	-------	------

	smoking_condition	stroke
0	formerly smoked	1
1	never smoked	1
2	never smoked	1
3	smokes	1
4	never smoked	1
...
5105	never smoked	0
5106	never smoked	0
5107	never smoked	0
5108	formerly smoked	0
5109	Unknown	0

[5110 rows x 11 columns]

```
health1.isnull().sum()
```

```
sex          0
age          0
hyper_tension  0
heart_condition  0
marital_status  0
working_type  0
residential_area  0
glucose_level  0
body_mass_index  201
smoking_condition  0
stroke       0
dtype: int64
```

```
health1['body_mass_index'].mean()
```

28.893236911794673

```
mean1 = health1["body_mass_index"].mean()
```

```
health1['body_mass_index'].replace(np.nan, mean1 ,inplace=True)
```

```
health1.isnull().sum()
```

```
sex          0
age          0
hyper_tension  0
heart_condition  0
marital_status  0
working_type  0
residential_area  0
glucose_level  0
body_mass_index  0
smoking_condition  0
```

```
stroke          0
dtype: int64

cat1_features
=health1[["sex","marital_status","working_type","smoking_condition",
"residential_area"]]
numeric1_features = health1[["age","hyper_tension",
"heart_condition","glucose_level", "body_mass_index", "stroke"]]
cat1_features.head()
```

	sex	marital_status	working_type	smoking_condition	residential_area
0	Male	Yes	Private	formerly smoked	Urban
1	Female	Yes	Self-employed	never smoked	Rural
2	Male	Yes	Private	never smoked	Rural
3	Female	Yes	Private	smokes	Urban
4	Female	Yes	Self-employed	never smoked	Rural

```
numeric1_features.head()
```

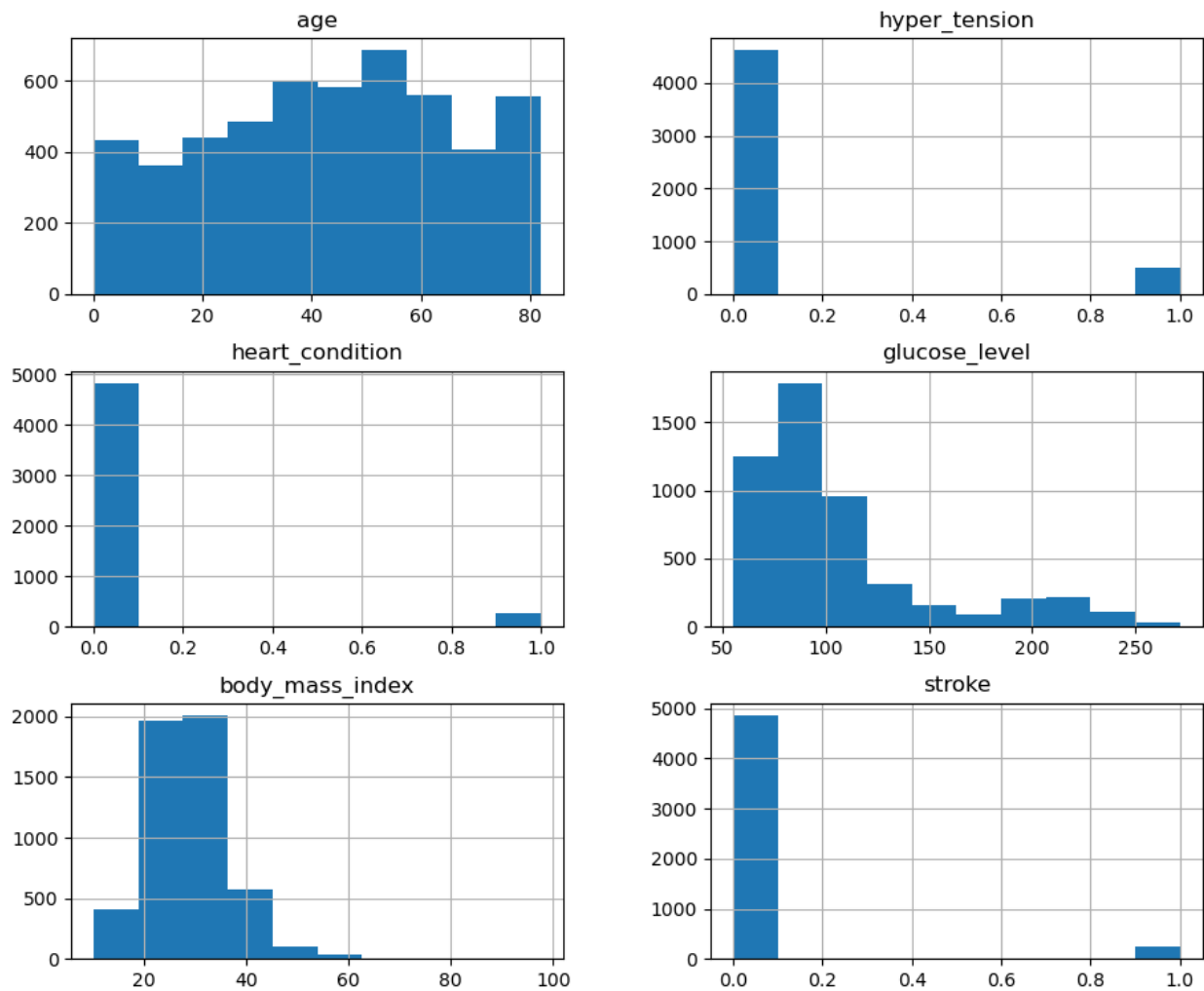
	age	hyper_tension	heart_condition	glucose_level	body_mass_index \
0	67.0	0	1	228.69	36.600000
1	61.0	0	0	202.21	28.893237
2	80.0	0	1	105.92	32.500000
3	49.0	0	0	171.23	34.400000
4	79.0	1	0	174.12	24.000000

	stroke
0	1
1	1
2	1
3	1
4	1

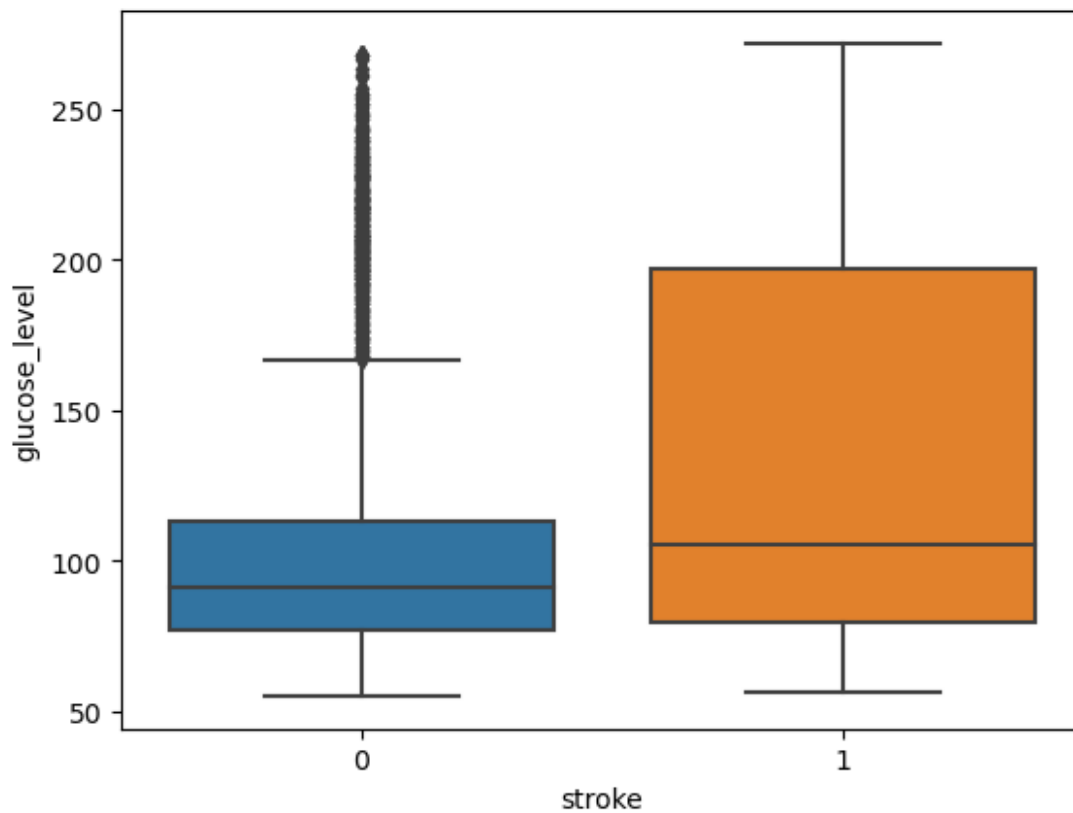
```
numeric1_features.hist(figsize=[11,9])
```

```
array([[<AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'hyper_tension'}>],
       [<AxesSubplot:title={'center':'heart_condition'}>,
        <AxesSubplot:title={'center':'glucose_level'}>],
```

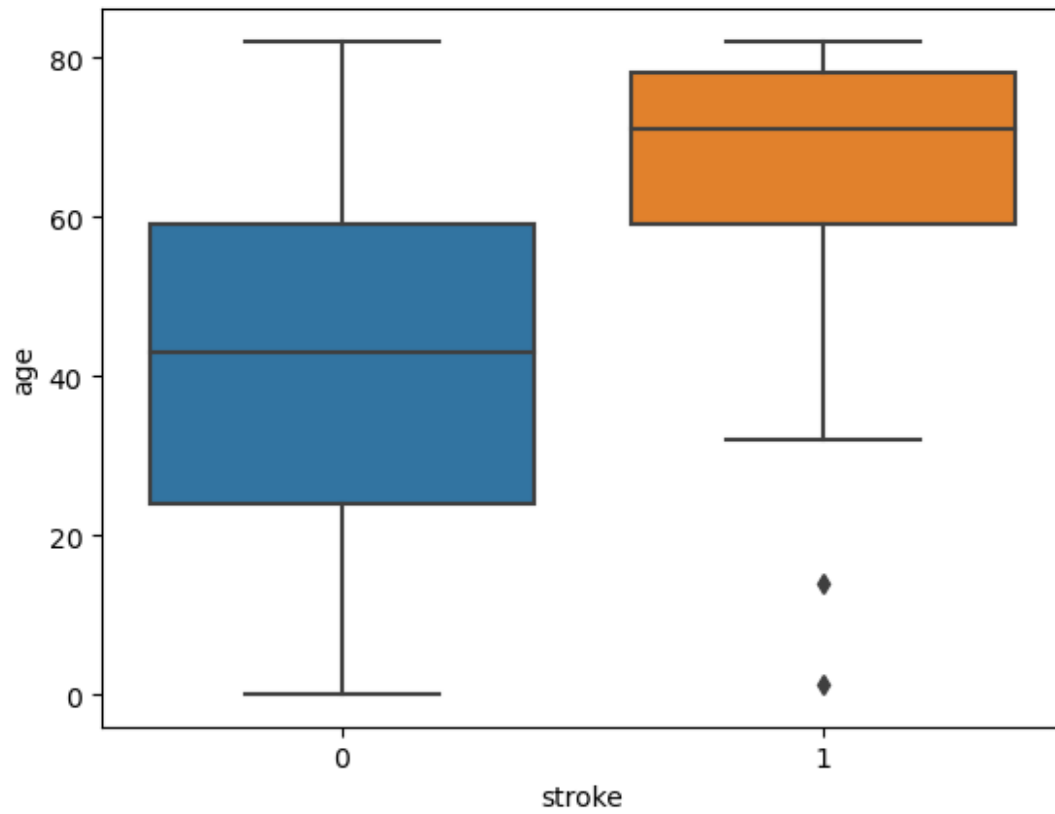
```
[<AxesSubplot:title={'center':'body_mass_index'}>,
 <AxesSubplot:title={'center':'stroke'}>]], dtype=object)
```



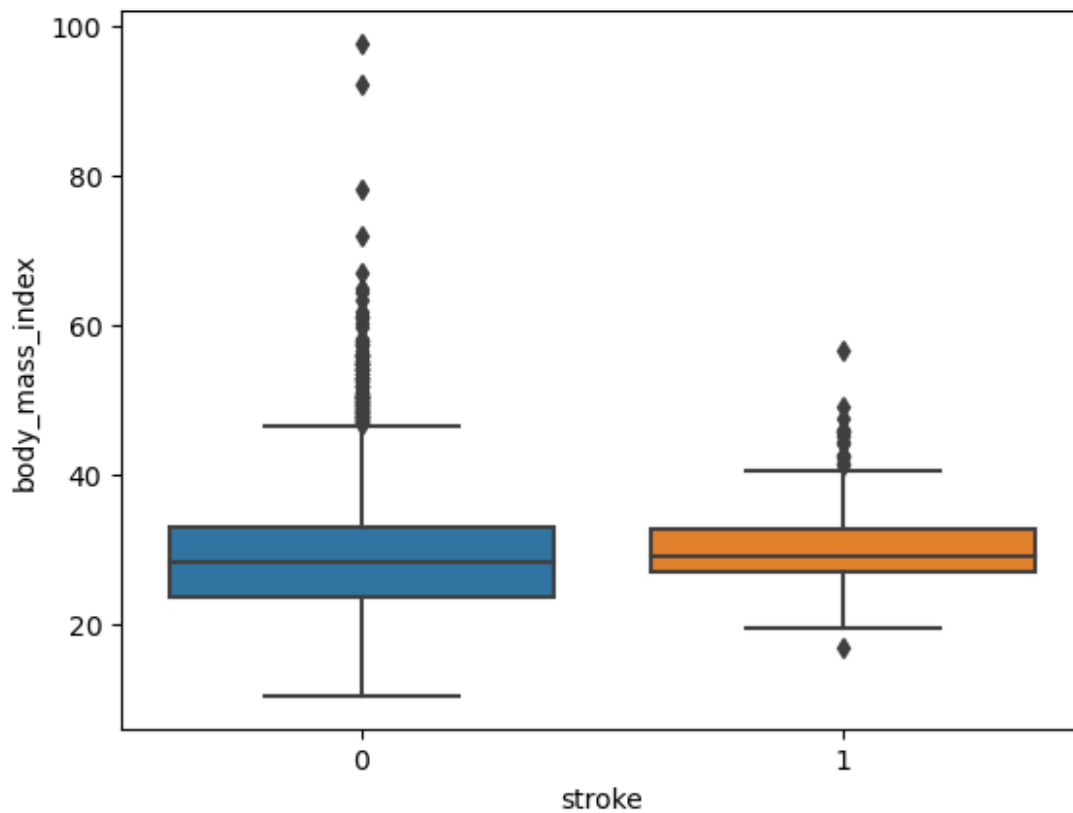
```
sns.boxplot(x='stroke',y='glucose_level',data=health1)
<AxesSubplot:xlabel='stroke', ylabel='glucose_level'>
```



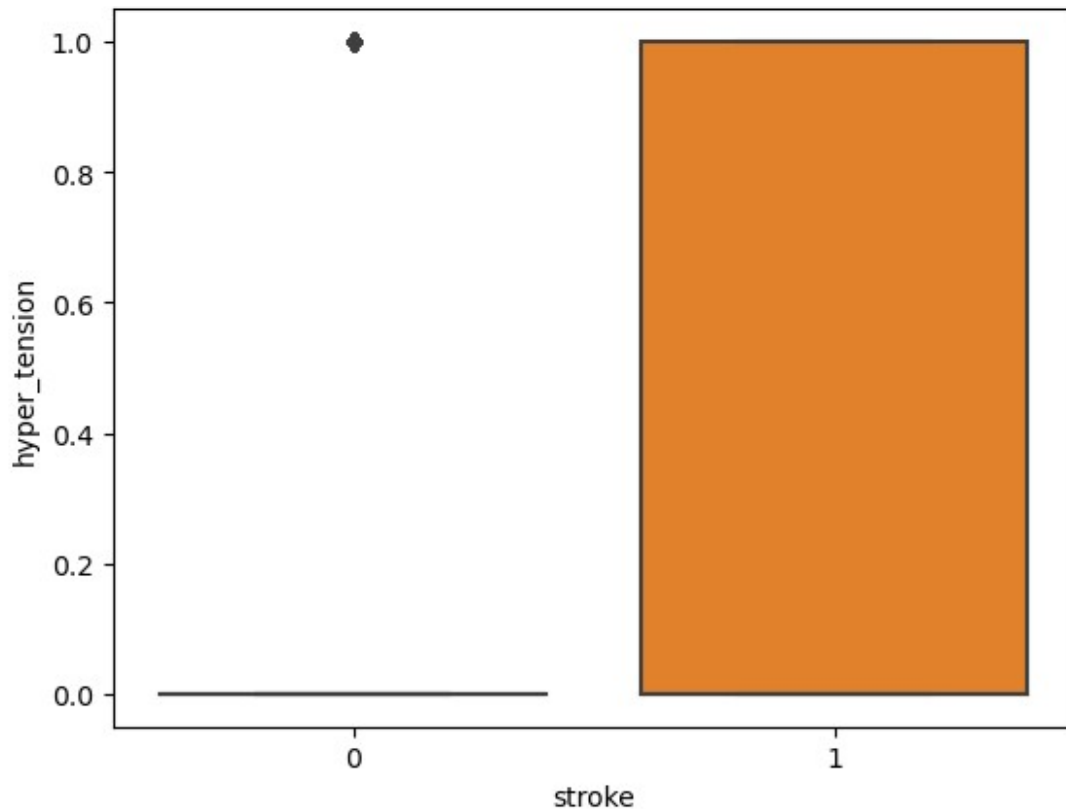
```
sns.boxplot(x='stroke',y='age',data=health1)  
<AxesSubplot:xlabel='stroke', ylabel='age'>
```



```
sns.boxplot(x='stroke',y='body_mass_index',data=health1)
<AxesSubplot:xlabel='stroke', ylabel='body_mass_index'>
```



```
sns.boxplot(x='stroke',y='hyper_tension',data=health1)  
<AxesSubplot:xlabel='stroke', ylabel='hyper_tension'>
```

```
sns.FacetGrid(health1,hue="stroke",height =  
3).map(sns.distplot,"age").add_legend()
```

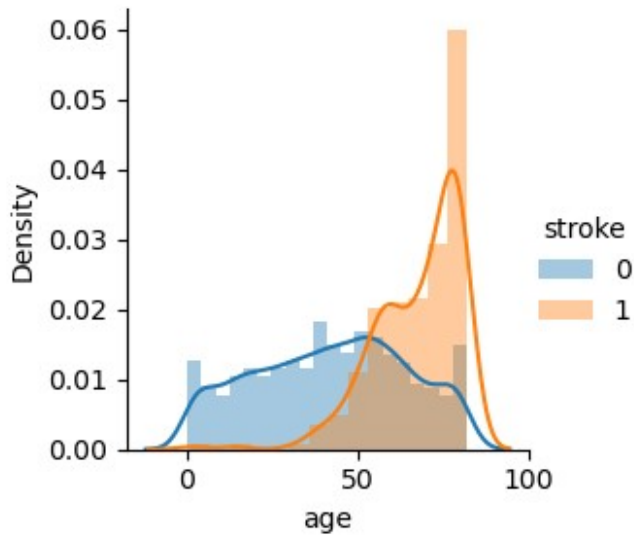
```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed  
in a future version. Please adapt your code to use either `displot` (a  
figure-level function with similar flexibility) or `histplot` (an  
axes-level function for histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed  
in a future version. Please adapt your code to use either `displot` (a  
figure-level function with similar flexibility) or `histplot` (an  
axes-level function for histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
<seaborn.axisgrid.FacetGrid at 0x2523e71e790>
```



```
sns.FacetGrid(health1,hue="stroke",height =  
3).map(sns.distplot,"body_mass_index").add_legend()
```

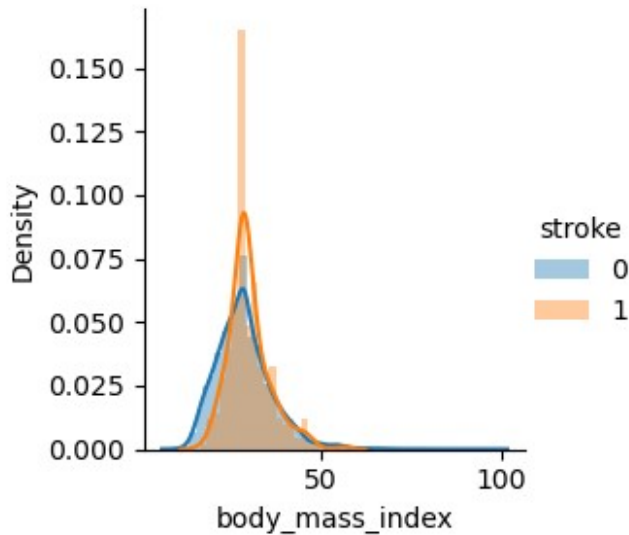
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
<seaborn.axisgrid.FacetGrid at 0x2523e92cca0>
```



```
count1 = cat1_features["sex"].value_counts()
percent100 =
cat1_features["sex"].value_counts(normalize=True).mul(100).round(1).as
type(str) + '%'
stroke1_conditions=pd.DataFrame({'counts': count1, 'Percent':
percent100})
print(stroke1_conditions)
count1 = cat1_features["marital_status"].value_counts()
percent100 =
cat1_features["marital_status"].value_counts(normalize=True).mul(100
).round(1).astype(str) + '%'
stroke1_conditions=pd.DataFrame({'counts': count1, 'Percent':
percent100})
print(stroke1_conditions)
count1 = cat1_features["working_type"].value_counts()
percent100 =
cat1_features["working_type"].value_counts(normalize=True).mul(100).ro
und(1).astype(str) + '%'
stroke1_conditions=pd.DataFrame({'counts': count1, 'Percent':
percent100})
print(stroke1_conditions)
count1 = cat1_features["residential_area"].value_counts()
percent100 =
cat1_features["residential_area"].value_counts(normalize=True).mul(100)
.round(1).astype(str) + '%'
stroke1_conditions=pd.DataFrame({'counts': count1, 'Percent':
percent100})
print(stroke1_conditions)
```

	counts	Percent
Female	2994	58.6%
Male	2115	41.4%

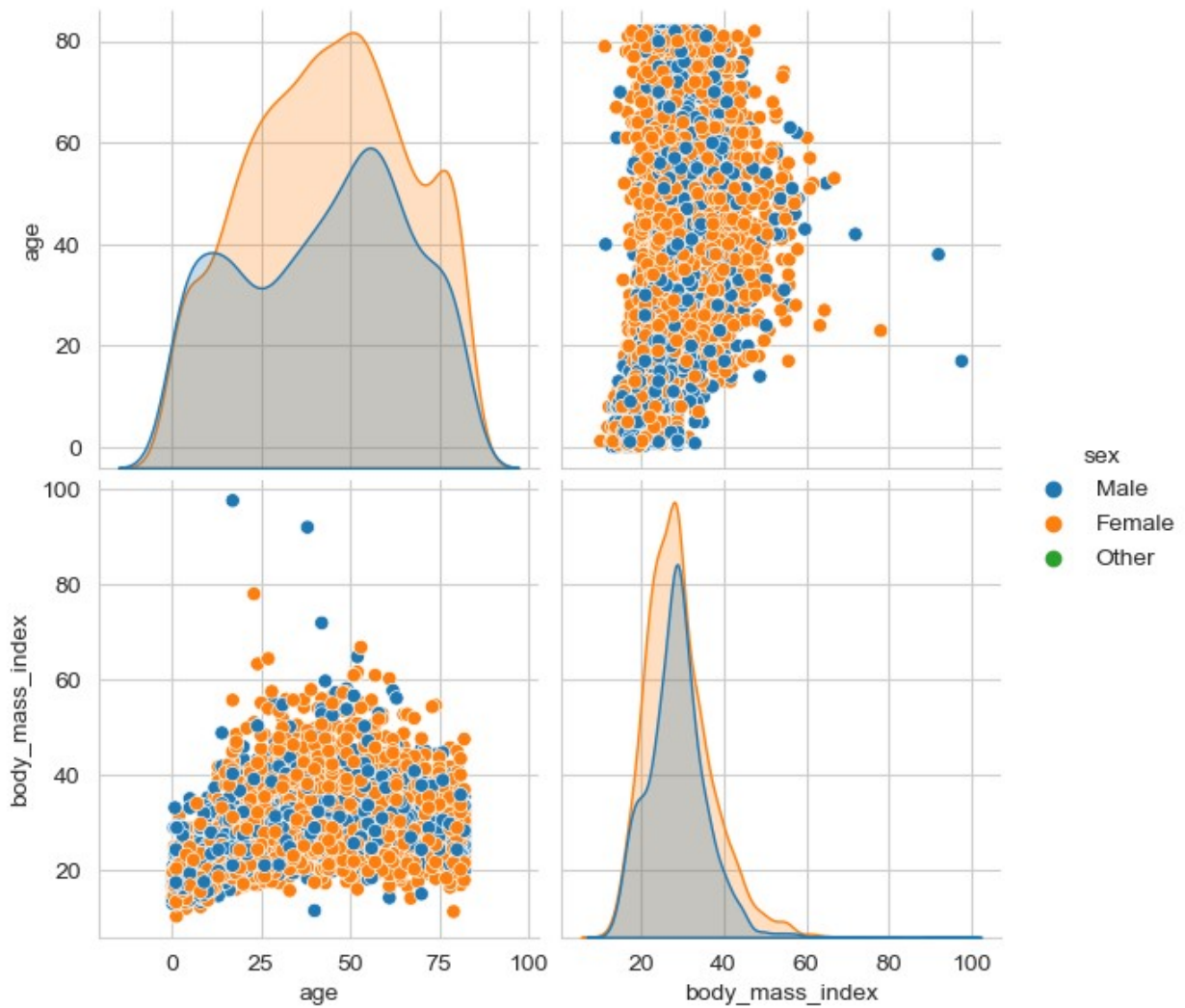
Other	1	0.0%
	counts	Percent
Yes	3353	65.6%
No	1757	34.4%
	counts	Percent
Private	2925	57.2%
Self-employed	819	16.0%
children	687	13.4%
Govt_job	657	12.9%
Never_worked	22	0.4%
	counts	Percent
Urban	2596	50.8%
Rural	2514	49.2%

```

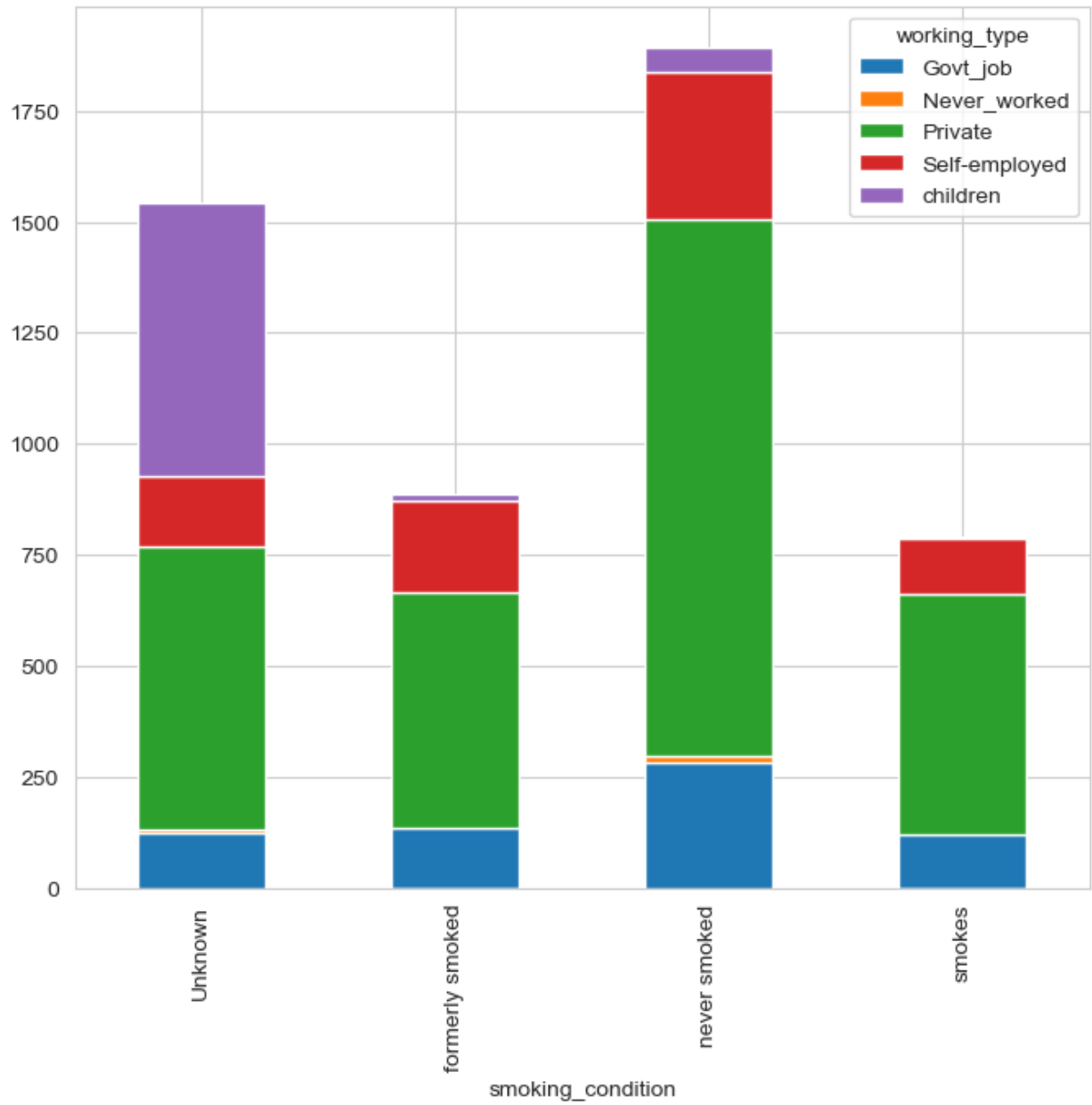
numeric1_features = health1[["age", "sex", "body_mass_index"]]
sns.set_style("whitegrid")
sns.pairplot(numeric1_features, hue = "sex", height = 3)

<seaborn.axisgrid.PairGrid at 0x2523d8e9850>

```



```
smoke1_vs_work1_type1 =
pd.crosstab(index=health1["smoking_condition"],columns=health1["workin
g_type"])
smoke1_vs_work1_type1.plot(kind="bar", figsize=(8,7),stacked=True)
<AxesSubplot:xlabel='smoking_condition'>
```



```
sns.heatmap(health1.corr(), annot = True, square=True)  
<AxesSubplot:>
```

