In [1]:

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

Reading the Data

In [2]:

```
df = pd.read_csv("healthcare-dataset-stroke-data.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_typ
0	9046	Male	67.0	0	1	Yes	Private	Urba
1	51676	Female	61.0	0	0	Yes	Self-employed	Rura
2	31112	Male	80.0	0	1	Yes	Private	Rura
3	60182	Female	49.0	0	0	Yes	Private	Urba
4	1665	Female	79.0	1	0	Yes	Self-employed	Rura

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

In [5]:

```
df.duplicated().sum()
```

Out[5]:

```
0
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
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 [7]:

```
# dealing with nan values with BMI column
df["bmi"]= df["bmi"].fillna(value=df["bmi"].mean())
df.isna().sum()
```

Out[7]:

```
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                  0
smoking_status       0
stroke               0
dtype: int64
```

In [8]:

```
df['age'] = df['age'].astype(int)
df.dtypes
```

Out[8]:

```
id                     int64
gender                object
age                    int32
hypertension           int64
heart_disease          int64
ever_married          object
work_type             object
Residence_type        object
avg_glucose_level    float64
bmi                  float64
smoking_status        object
stroke                 int64
dtype: object
```

In [9]:

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

Out[9]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_
0	Male	67	0	1	Yes	Private	Urban	
1	Female	61	0	0	Yes	Self-employed	Rural	
2	Male	80	0	1	Yes	Private	Rural	
3	Female	49	0	0	Yes	Private	Urban	
4	Female	79	1	0	Yes	Self-employed	Rural	

In [10]:

```
df["ever_married"].value_counts()
```

Out[10]:

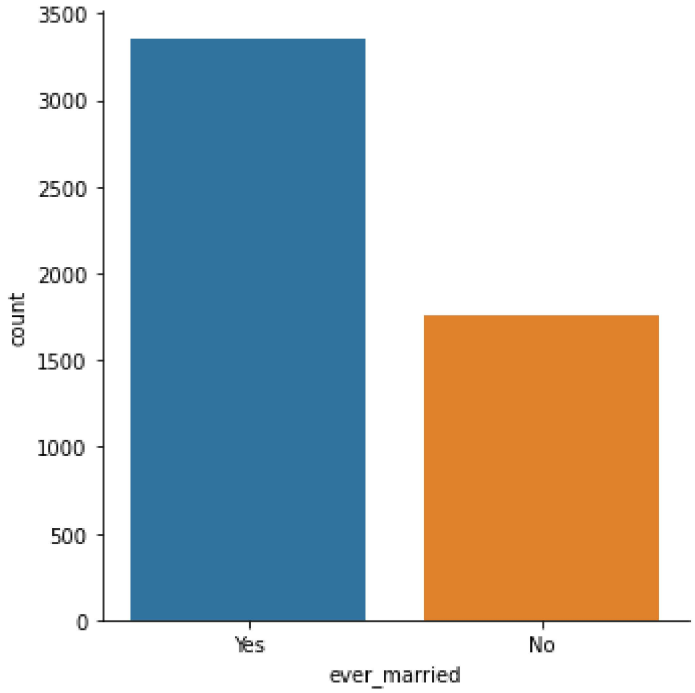
```
Yes    3353
No     1757
Name: ever_married, dtype: int64
```

In [11]:

```
sns.catplot(x="ever_married",
           data=df ,
           kind="count")
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x29c17f3dac0>
```



In [12]:

```
df["gender"].value_counts()
```

Out[12]:

```
Female    2994
Male      2115
Other        1
Name: gender, dtype: int64
```

In [13]:

```
df = df[df["gender"] != "Other"]
df["gender"].value_counts()
```

Out[13]:

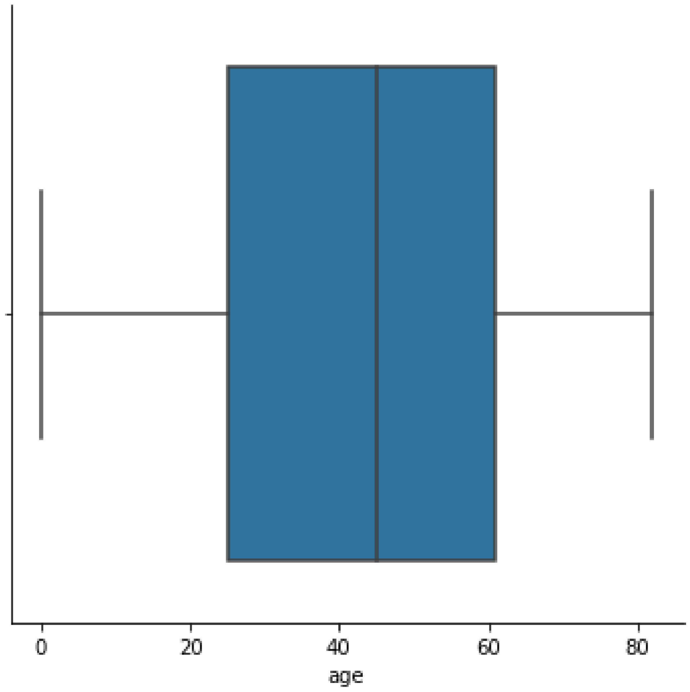
```
Female    2994
Male      2115
Name: gender, dtype: int64
```

In [14]:

```
# know the distribtion of the age and check if there any outliers or not
sns.catplot(x="age",
           data = df,
           kind="box")
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x29c18732af0>
```

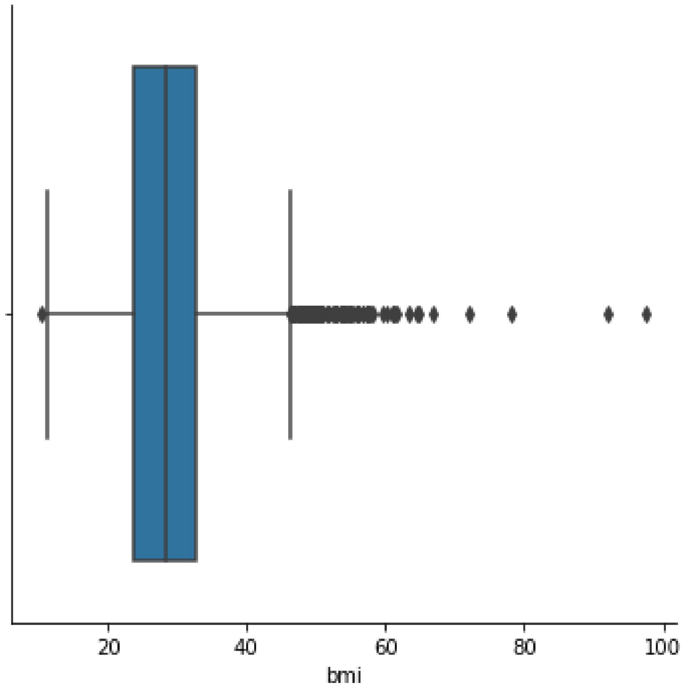


In [15]:

```
# checking for outliers
sns.catplot(x="bmi",
            data = df,
            kind="box")
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x29c18784a90>
```



In [16]:

```
df_bmi = df[df["bmi"] > 80]
df_bmi.head()
```

Out[16]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	av
2128	Male	17	1	0	No	Private	Rural	
4209	Male	38	1	0	Yes	Private	Rural	

In [17]:

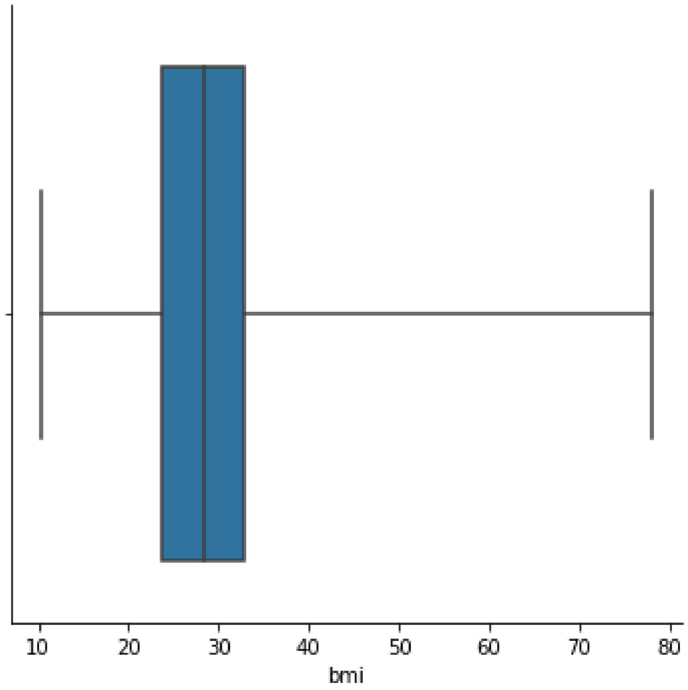
```
# removing out liers
df = df[df["bmi"] < 80]
```

In [18]:

```
# changing whis to include all vaules
sns.catplot(x="bmi",
           data = df,
           kind="box",
           whis=[0,100])
```

Out[18]:

```
<seaborn.axisgrid.FacetGrid at 0x29c18732d60>
```



In [19]:

```
df["smoking_status"].value_counts()
```

Out[19]:

```
never smoked       1891
Unknown            1543
formerly smoked     884
smokes              789
Name: smoking_status, dtype: int64
```

In [20]:

```
df["Residence_type"].value_counts()
```

Out[20]:

```
Urban    2596
Rural    2511
Name: Residence_type, dtype: int64
```

In [21]:

```
df["work_type"].value_counts()
```

Out[21]:

```
Private          2922
Self-employed     819
children          687
Govt_job          657
Never_worked       22
Name: work_type, dtype: int64
```

In [22]:

```
df["avg_glucose_level"].mean()
```

Out[22]:

```
106.1587487761894
```

In [23]:

```
df["avg_glucose_level"].median()
```

Out[23]:

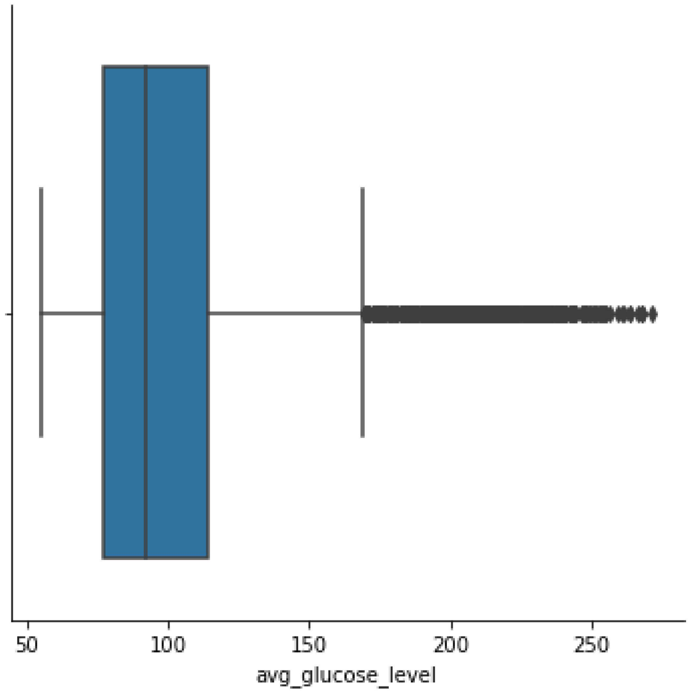
```
91.89
```

In [24]:

```
# i searched for glucose levels and i found it some times it over 250 so iwon't remove an
sns.catplot(x="avg_glucose_level",
           data = df,
           kind="box")
```

Out[24]:

```
<seaborn.axisgrid.FacetGrid at 0x29c1885f4f0>
```

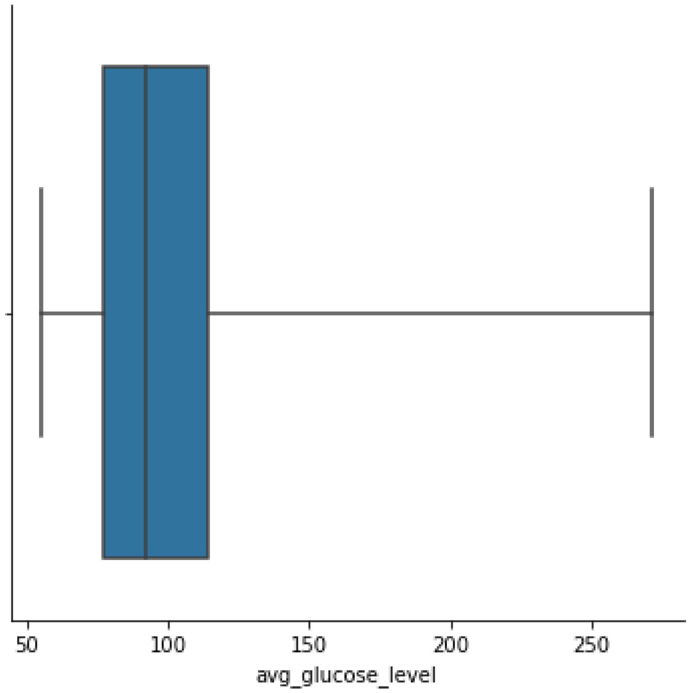


In [25]:

```
# we will change the whis to include all the data
sns.catplot(x="avg_glucose_level",
           data = df,
           kind="box",
           whis = [0,100])
```

Out[25]:

```
<seaborn.axisgrid.FacetGrid at 0x29c18906220>
```



making analysis on the work_type data set

In [26]:

```
df_jop = df[df["work_type"] == "children"]
df_jop.head()
```

Out[26]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg
162	Female	1	0	0	No	children	Urban	
245	Female	14	0	0	No	children	Rural	
249	Male	3	0	0	No	children	Rural	
282	Female	3	0	0	No	children	Urban	
290	Male	13	0	0	No	children	Urban	

In [27]:

```
df_jop["age"].mean()
```

Out[27]:

```
6.756914119359534
```

In [28]:

```
df_jop["age"].median()
```

Out[28]:

```
6.0
```

In [29]:

```
df_jop["smoking_status"].value_counts()
```

Out[29]:

```
Unknown            618
never smoked        54
formerly smoked     13
smokes               2
Name: smoking_status, dtype: int64
```

In [30]:

```
df_jop["ever_married"].value_counts()
```

Out[30]:

```
No    687
Name: ever_married, dtype: int64
```

In [31]:

```
df_jop["gender"].value_counts()
```

Out[31]:

```
Male      361
Female    326
Name: gender, dtype: int64
```

In [32]:

```
df_jop["bmi"].mean()
```

Out[32]:

```
20.244238414248493
```

In [33]:

```
df_jop["bmi"].median()
```

Out[33]:

```
19.0
```

answering questions

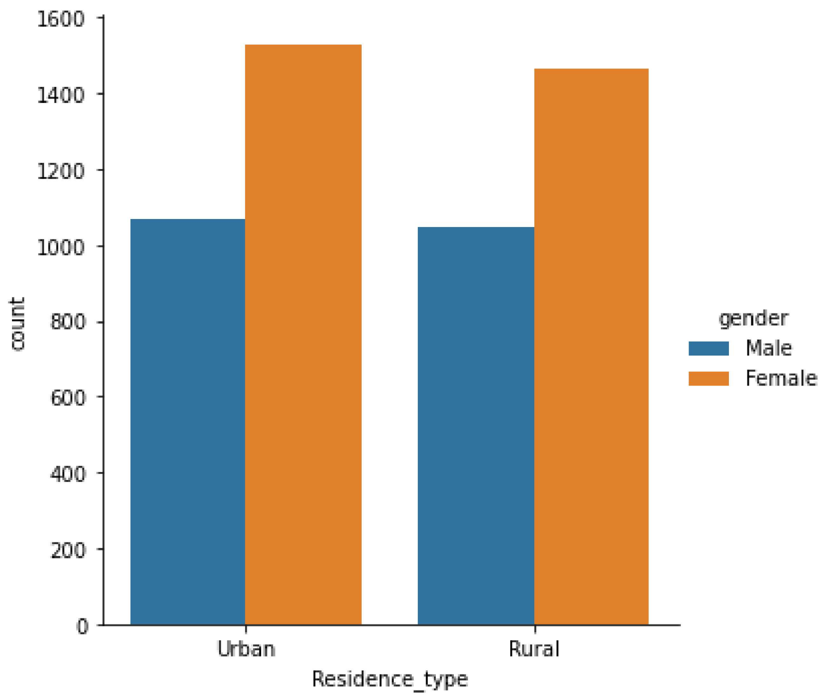
Q1: what is the number of male and female in each Residence_type

In [34]:

```
sns.catplot(x="Residence_type",
           data=df ,
           kind="count",
           hue="gender")
```

Out[34]:

```
<seaborn.axisgrid.FacetGrid at 0x29c18784ac0>
```



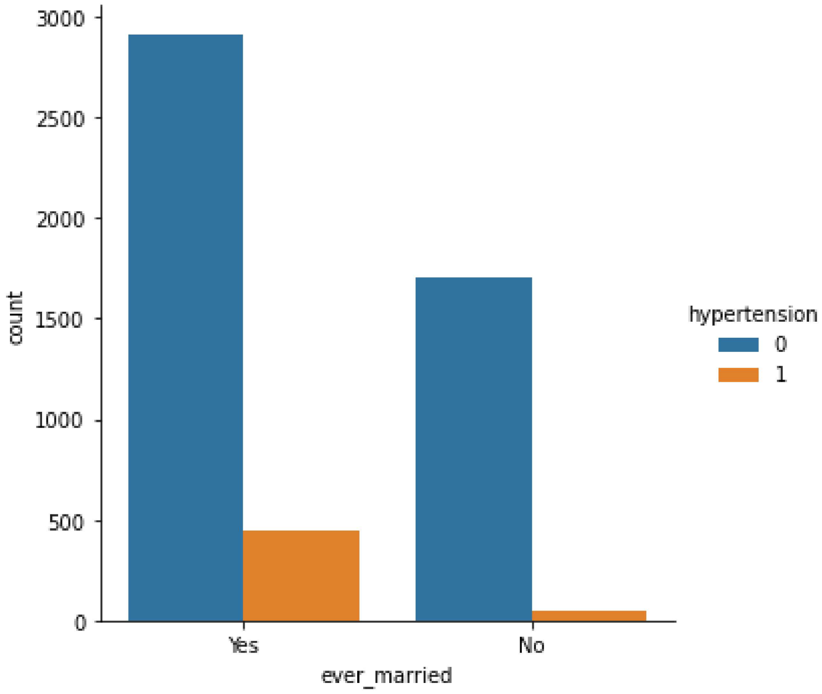
Q2 : is there corr between marriage and hypertension ?

In [45]:

```
# checking for corr marriage and hypertension Fjo
sns.catplot(x="ever_married",
           data=df ,
           kind="count",
           hue="hypertension")
```

Out[45]:

```
<seaborn.axisgrid.FacetGrid at 0x29c1a24be20>
```



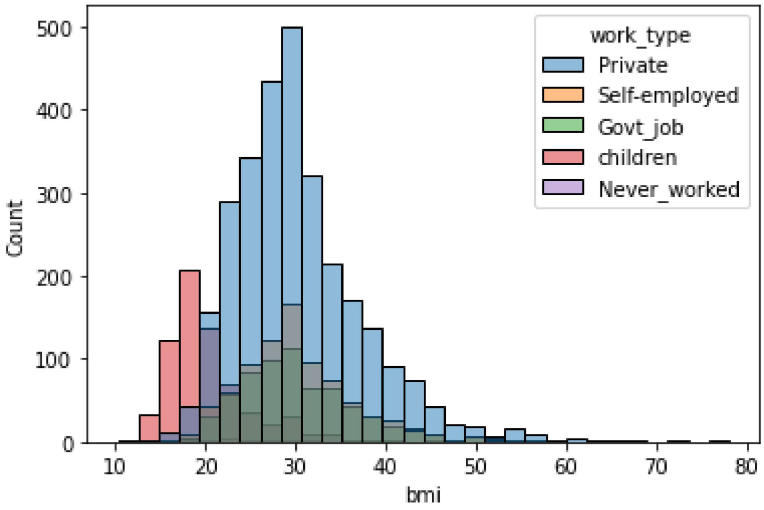
Q3 : what is the bmi of each work type

In [44]:

```
sns.histplot(x="bmi",
            data=df,
            hue= "work_type",
            bins=30)
```

Out[44]:

```
<AxesSubplot:xlabel='bmi', ylabel='Count'>
```



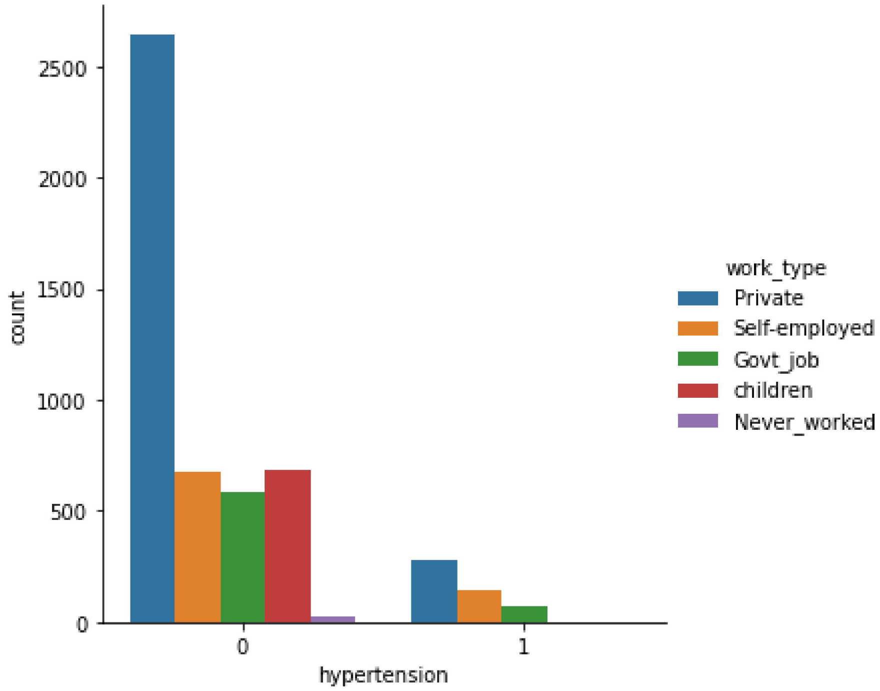
Q4 : which is the work type that has the highest number of hypertension

In [48]:

```
sns.catplot(x="hypertension",
           data=df ,
           kind="count",
           hue="work_type")
```

Out[48]:

```
<seaborn.axisgrid.FacetGrid at 0x29c1d44eca0>
```



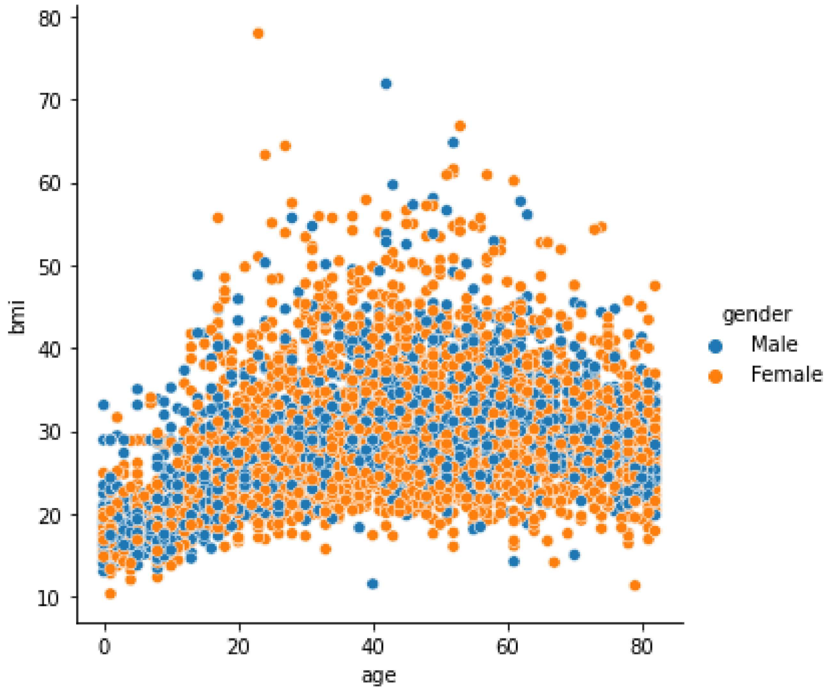
Q5 : is their corr between age and bmi?

In [60]:

```
sns.relplot(x= "age",
           y="bmi",
           data=df,
           kind="scatter",
            hue="gender")
```

Out[60]:

```
<seaborn.axisgrid.FacetGrid at 0x29c1e9ddcd0>
```



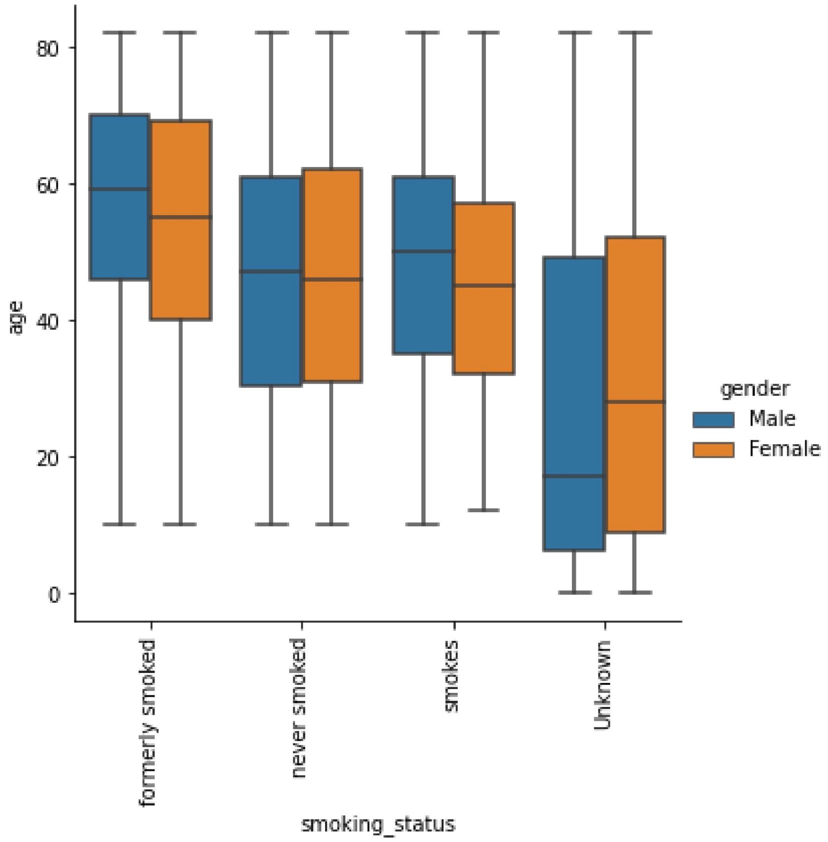
Q6 :knowing what is the mean of smoking status in each gender

In [59]:

```
sns.catplot(x="smoking_status",
       y="age",
       data=df,
        kind="box",
        hue= "gender")
plt.xticks(rotation=90)
plt.show
```

Out[59]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [ ]: