In [119]:

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

In [120]:

```
stroke_data = pd.read_csv('C:/Users/nebar/Downloads/stroke data.csv')
```

In [121]:

```
stroke_data.head()
```

Out[121]:

	id	gender	age	married	hypertension	heart_disease	occupation	residence	metric_1	metric_2	metric_3	metri
0	1	Male	3.0	No	0	0	A	Rural	95.12	18.0	1	9
1	2	Male	58.0	Yes	1	0	B	Urban	87.96	39.2	1	9
2	3	Female	8.0	No	0	0	B	Urban	110.89	17.6	0	9
3	4	Female	70.0	Yes	0	0	B	Rural	69.04	35.9	0	9
4	5	Male	14.0	No	0	0	C	Rural	161.28	19.1	1	9

In [122]:

```
#Since id has no statistical value other than identifying each patient we have to drop it
stroke_data.drop("id",axis = 1, inplace = True)
stroke_data.describe()
```

Out[122]:

	age	hypertension	heart_disease	metric_1	metric_2	metric_3	metric_4	metr
count	43400.000000	43400.000000	43400.000000	43400.000000	41938.000000	43400.000000	43400.000000	43400.00
mean	42.261212	0.093571	0.047512	104.482750	28.605038	0.289931	97.526855	104.48
std	23.438911	0.291235	0.212733	43.111751	7.770020	0.453735	1.466703	43.11
min	-10.000000	0.000000	0.000000	55.000000	10.100000	0.000000	87.420000	55.00
25%	24.000000	0.000000	0.000000	77.540000	23.200000	0.000000	96.590000	77.54
50%	44.000000	0.000000	0.000000	91.580000	27.700000	0.000000	97.610000	91.58
75%	60.000000	0.000000	0.000000	112.070000	32.900000	1.000000	98.700000	112.07
max	1000.000000	1.000000	1.000000	291.050000	97.600000	1.000000	100.000000	291.05

In [123]:

```
stroke_data.isnull().sum()
```

Out[123]:

```
gender                0
age                   0
married               0
hypertension          0
heart_disease         0
occupation            0
residence             0
metric_1              0
metric_2           1462
metric_3              0
metric_4              0
metric_5              0
smoking_status    13292
stroke                0
dtype: int64
```

```
In [124]: #To get missing numerical values and count
          num_vars= stroke_data.columns[stroke_data.dtypes != 'object']
          stroke_data[num_vars].isnull().sum()
```

```
Out[124]: age                0
          hypertension       0
          heart_disease      0
          metric_1           0
          metric_2        1462
          metric_3           0
          metric_4           0
          metric_5           0
          stroke             0
          dtype: int64
```

```
In [125]: #To get missing categorical values and count
          cat_vars= stroke_data.columns[stroke_data.dtypes == 'object']
          stroke_data[cat_vars].isnull().sum()
```

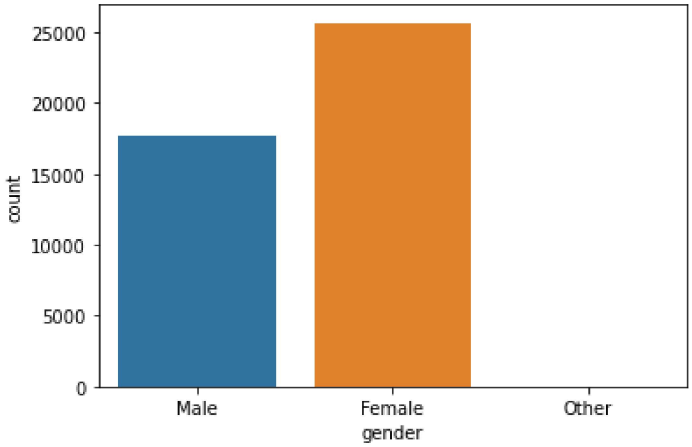
```
Out[125]: gender                0
          married               0
          occupation            0
          residence             0
          smoking_status    13292
          dtype: int64
```

```
In [126]: gender_count = stroke_data["gender"].value_counts()
          gender_count
```

```
Out[126]: Female    25665
          Male      17724
          Other        11
          Name: gender, dtype: int64
```

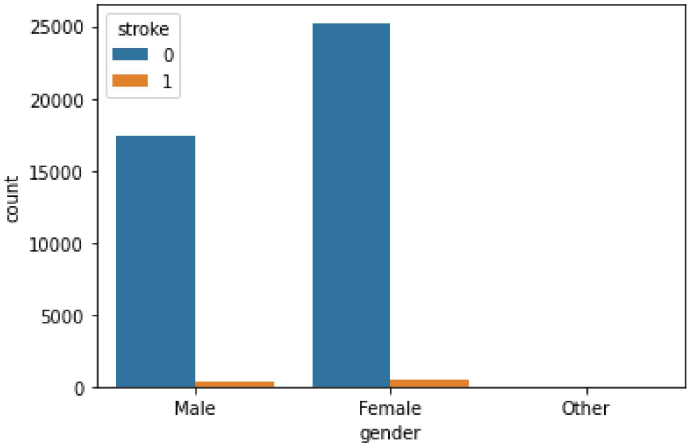
```
In [127]: sns.countplot(data = stroke_data, x='gender')
```

```
Out[127]: <AxesSubplot:xlabel='gender', ylabel='count'>
```



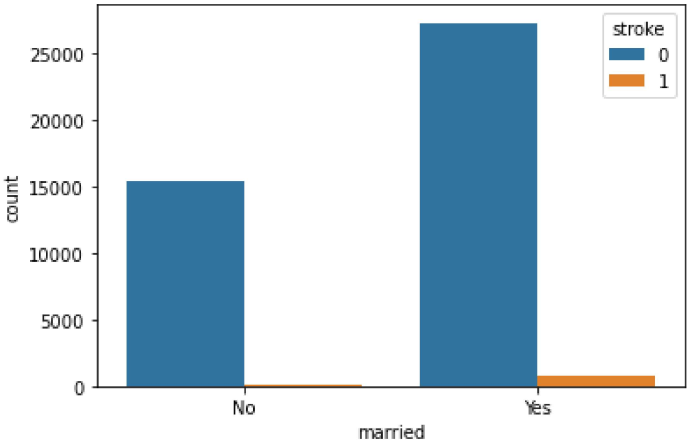
```
In [128]: sns.countplot(x='gender',hue = 'stroke',data = stroke_data)
```

```
Out[128]: <AxesSubplot:xlabel='gender', ylabel='count'>
```



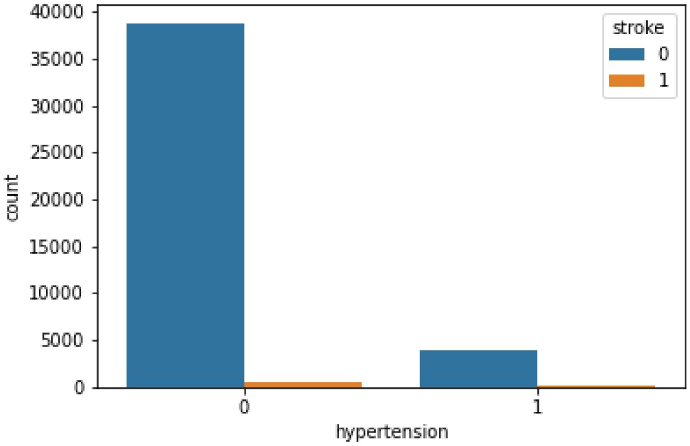
```
In [129]: sns.countplot(x='married',hue = 'stroke',data = stroke_data)
```

```
Out[129]: <AxesSubplot:xlabel='married', ylabel='count'>
```



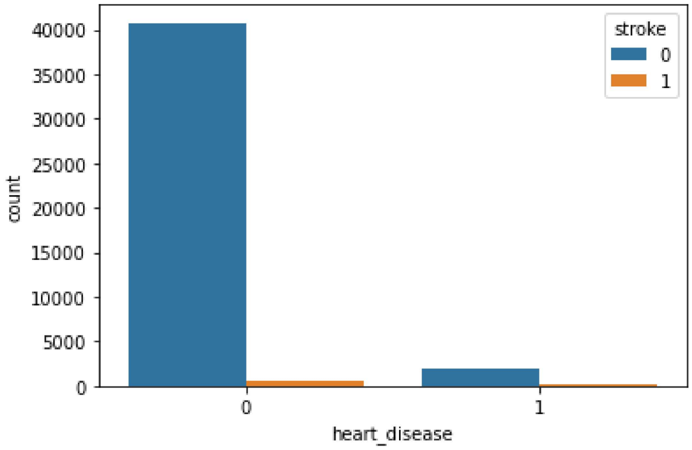
```
In [130]: sns.countplot(x='hypertension',hue = 'stroke',data= stroke_data)
```

```
Out[130]: <AxesSubplot:xlabel='hypertension', ylabel='count'>
```



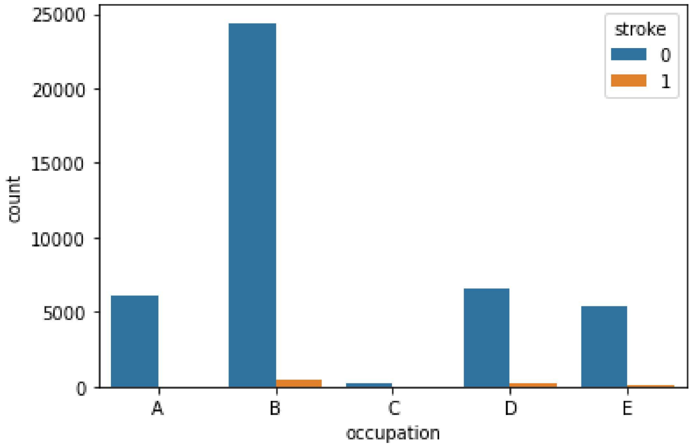
```
In [131]: sns.countplot(x='heart_disease',hue ='stroke',data= stroke_data)

Out[131]: <AxesSubplot:xlabel='heart_disease', ylabel='count'>
```



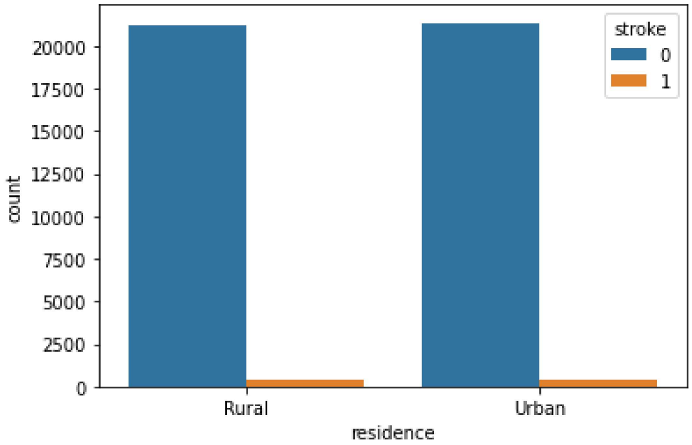
```
In [38]: sns.countplot(x='occupation',hue ='stroke',data= stroke_data)

Out[38]: <AxesSubplot:xlabel='occupation', ylabel='count'>
```



```
In [132]: sns.countplot(x='residence',hue ='stroke',data= stroke_data)

Out[132]: <AxesSubplot:xlabel='residence', ylabel='count'>
```

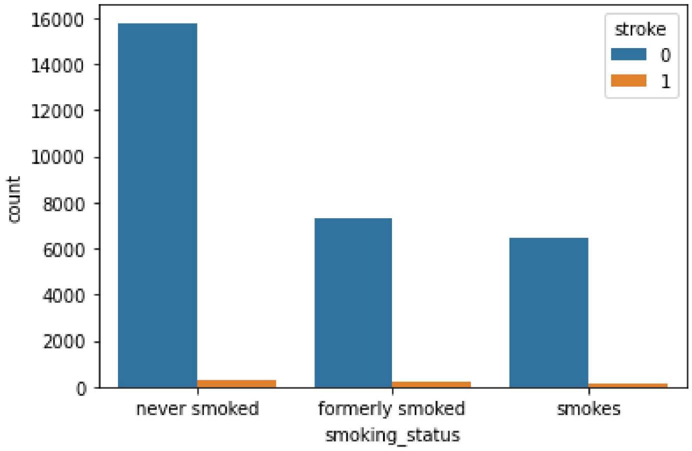


```
In [133]: smoke_count = stroke_data["smoking_status"].value_counts()
          smoke_count
```

```
Out[133]: never smoked       16053
          formerly smoked     7493
          smokes              6562
          Name: smoking_status, dtype: int64
```

```
In [134]: sns.countplot(data= stroke_data,x='smoking_status',hue ='stroke')
```

```
Out[134]: <AxesSubplot:xlabel='smoking_status', ylabel='count'>
```

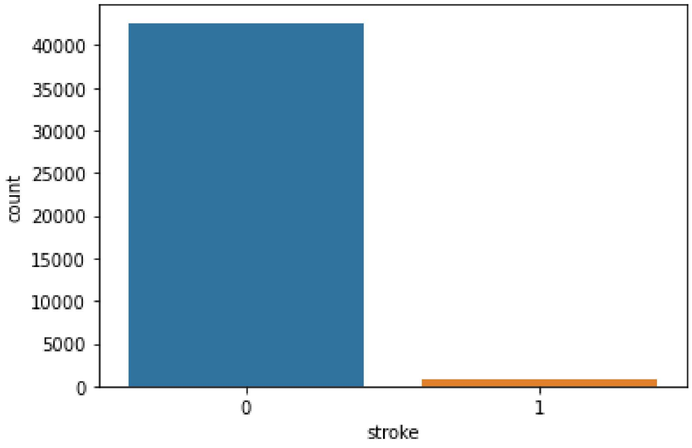


```
In [135]: #number of patients who had stroke
          stroke_count = stroke_data["stroke"].value_counts()
          stroke_count
```

```
Out[135]: 0    42617
          1      783
          Name: stroke, dtype: int64
```

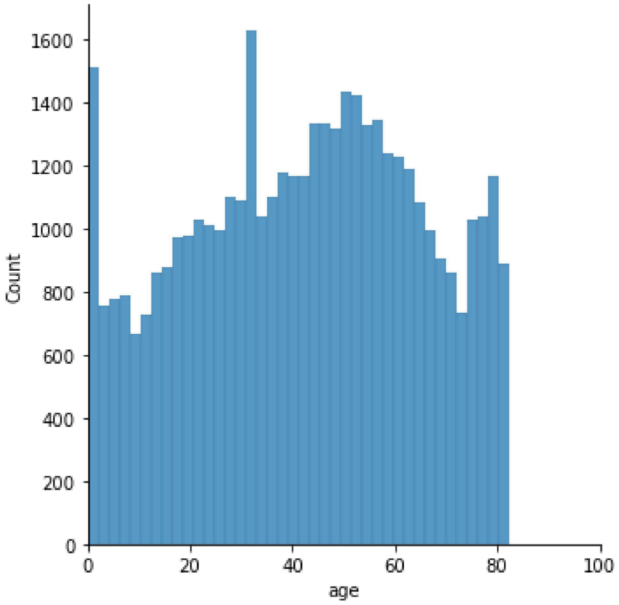
```
In [136]: sns.countplot(x= 'stroke',data=stroke_data)
```

```
Out[136]: <AxesSubplot:xlabel='stroke', ylabel='count'>
```



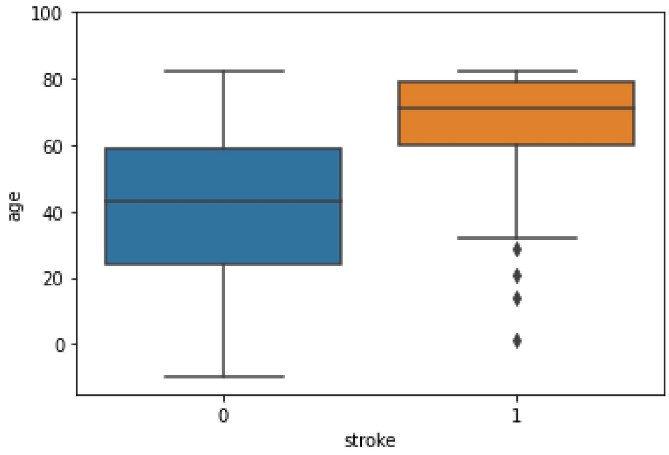
```
In [137]: sns.displot(stroke_data['age'])
          plt.xlim(0,100)

Out[137]: (0.0, 100.0)
```



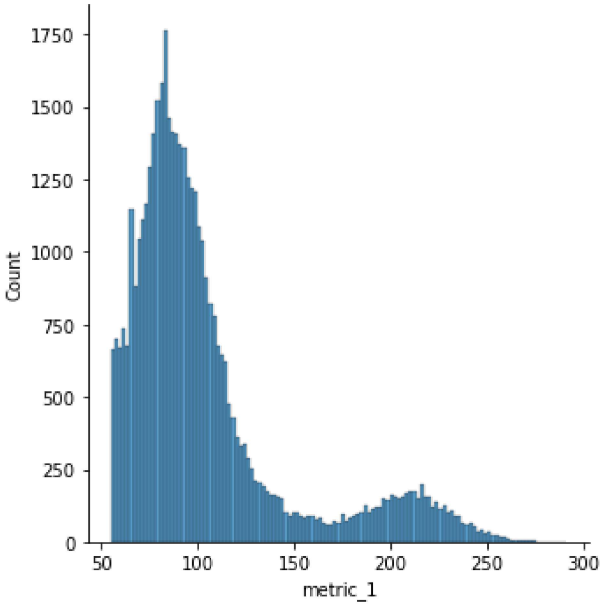
```
In [138]: #There are some outliers which shows cases of stroke in patients younger than 30
          #which might be other underlying diseases or error during data entry
          sns.boxplot(x='stroke',y='age',data=stroke_data)
          #Age range starts at -15 because during summarizing the data, the minimum age is -10 which migh
          plt.ylim(-15,100)

Out[138]: (-15.0, 100.0)
```



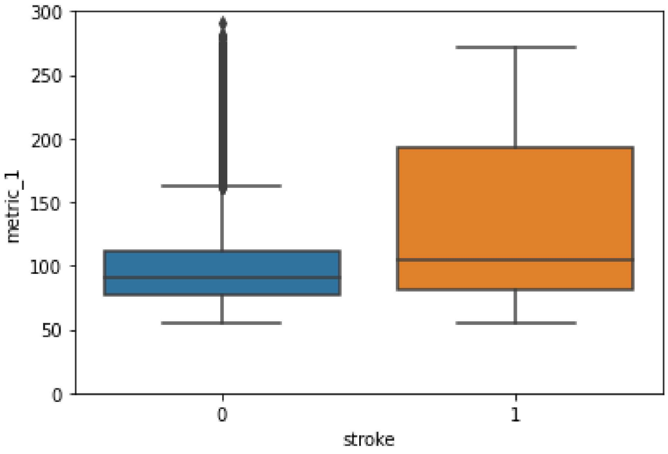
```
In [139]: sns.displot(stroke_data['metric_1'])
```

```
Out[139]: <seaborn.axisgrid.FacetGrid at 0x1b0e9d95f70>
```



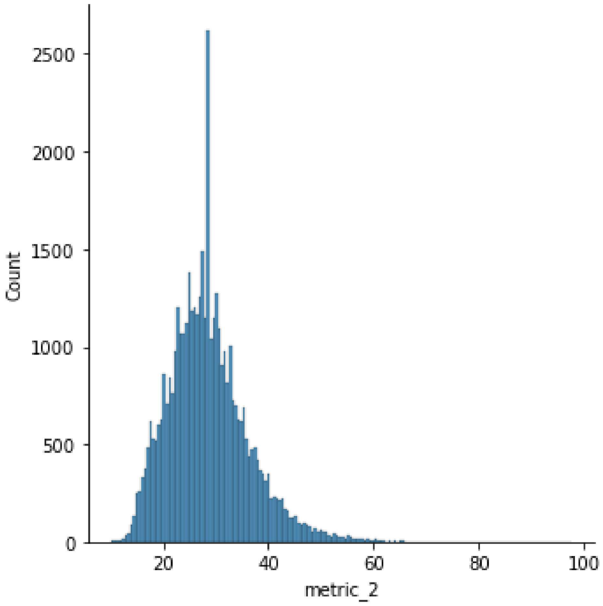
```
In [140]: sns.boxplot(data= stroke_data,x ='stroke',y='metric_1')
          plt.ylim(0,300)
```

```
Out[140]: (0.0, 300.0)
```



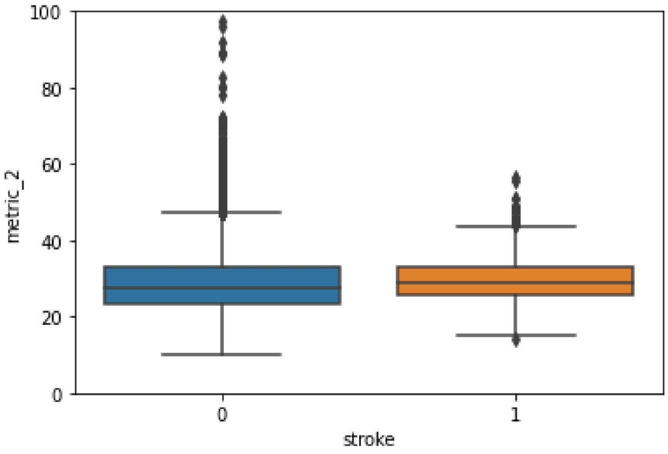
```
In [146]: stroke_data['metric_2'].fillna(stroke_data['metric_2'].mean(),inplace= True)
          sns.displot(stroke_data['metric_2'])

Out[146]: <seaborn.axisgrid.FacetGrid at 0x1b0e6136520>
```



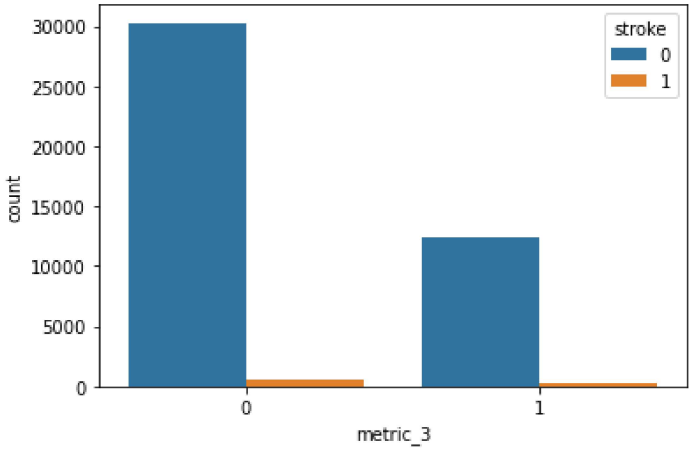
```
In [143]: sns.boxplot(x ='stroke',y='metric_2',data= stroke_data)
          plt.ylim(0,100)

Out[143]: (0.0, 100.0)
```



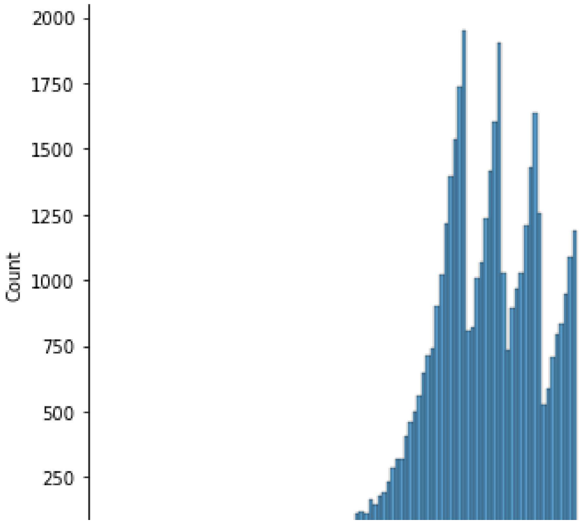
```
In [145]: sns.countplot(x ='metric_3',hue= 'stroke',data= stroke_data)

Out[145]: <AxesSubplot:xlabel='metric_3', ylabel='count'>
```



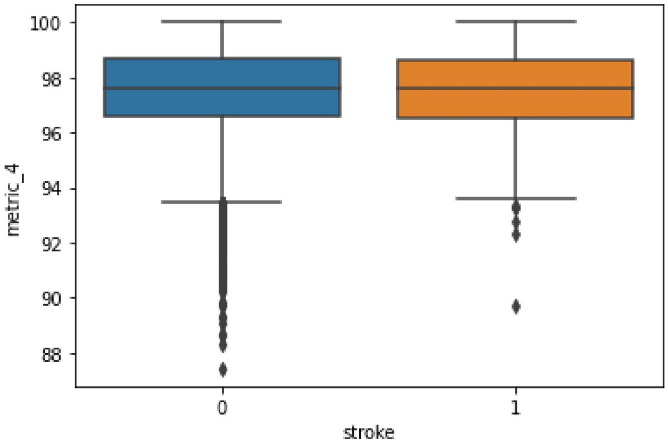
```
In [87]: sns.displot(stroke_data['metric_4'])

Out[87]: <seaborn.axisgrid.FacetGrid at 0x1b0e751ad90>
```



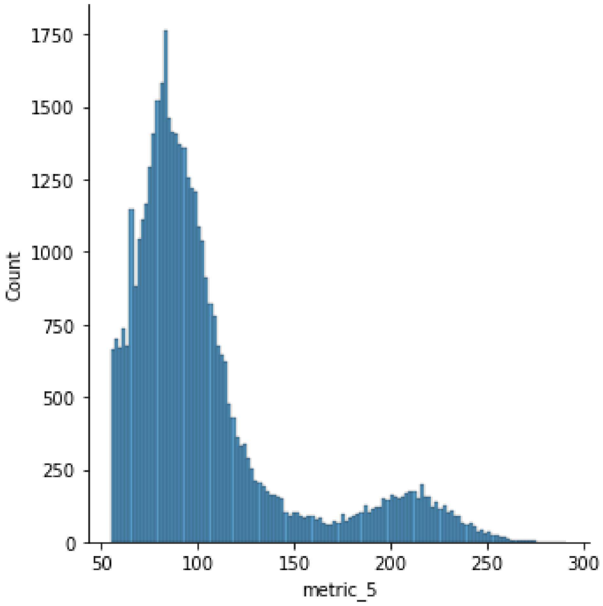
```
In [91]: sns.boxplot(x='stroke',y='metric_4',data= stroke_data)

Out[91]: <AxesSubplot:xlabel='stroke', ylabel='metric_4'>
```



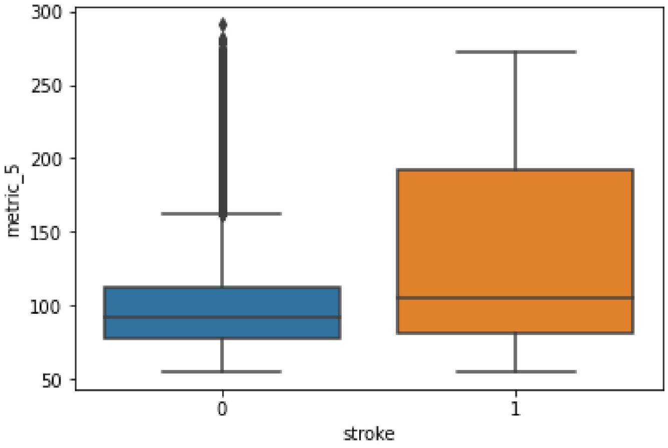
```
In [92]: sns.displot(stroke_data['metric_5'])
```

```
Out[92]: <seaborn.axisgrid.FacetGrid at 0x1b0e75640a0>
```



```
In [93]: sns.boxplot(x='stroke',y='metric_5',data= stroke_data)
```

```
Out[93]: <AxesSubplot:xlabel='stroke', ylabel='metric_5'>
```



In [148]:

```
stroke_data.corr()
```

Out[148]:

	age	hypertension	heart_disease	metric_1	metric_2	metric_3	metric_4	metric_5	stroke
age	1.000000	0.264053	0.244279	0.226538	0.337114	0.000002	-0.005108	0.226538	0.149678
hypertension	0.264053	1.000000	0.119777	0.160211	0.153779	-0.002164	0.012179	0.160211	0.075332
heart_disease	0.244279	0.119777	1.000000	0.146938	0.054133	-0.006168	0.001507	0.146938	0.113763
metric_1	0.226538	0.160211	0.146938	1.000000	0.184199	-0.008735	-0.005511	1.000000	0.078917
metric_2	0.337114	0.153779	0.054133	0.184199	1.000000	-0.003122	0.000975	0.184199	0.018407
metric_3	0.000002	-0.002164	-0.006168	-0.008735	-0.003122	1.000000	-0.007261	-0.008735	-0.003440
metric_4	-0.005108	0.012179	0.001507	-0.005511	0.000975	-0.007261	1.000000	-0.005511	-0.008088
metric_5	0.226538	0.160211	0.146938	1.000000	0.184199	-0.008735	-0.005511	1.000000	0.078917
stroke	0.149678	0.075332	0.113763	0.078917	0.018407	-0.003440	-0.008088	0.078917	1.000000

In [149]:

```
from sklearn.preprocessing import LabelEncoder
cols = stroke_data.select_dtypes(include=['object']).columns
x= LabelEncoder()
stroke_data[cols]= stroke_data[cols].apply(x.fit_transform)
stroke_data.head()
```

Out[149]:

	gender	age	married	hypertension	heart_disease	occupation	residence	metric_1	metric_2	metric_3	metric_4
0	1	3.0	0	0	0	0	0	95.12	18.0	1	99.35
1	1	58.0	1	1	0	1	1	87.96	39.2	1	99.70
2	0	8.0	0	0	0	1	1	110.89	17.6	0	96.35
3	0	70.0	1	0	0	1	0	69.04	35.9	0	95.52
4	1	14.0	0	0	0	2	0	161.28	19.1	1	95.10

In [150]:
```
#After changing the categorical data to numeric in order to see the whole variables correlatio
stroke_data.corr()
```

Out[150]:

	gender	age	married	hypertension	heart_disease	occupation	residence	metric_1	met
gender	1.000000	-0.028438	-0.031351	0.023709	0.082061	-0.036784	0.001508	0.035465	-0.02
age	-0.028438	1.000000	0.665224	0.264053	0.244279	0.433214	-0.000605	0.226538	0.33
married	-0.031351	0.665224	1.000000	0.176575	0.128833	0.367858	0.004422	0.153607	0.33
hypertension	0.023709	0.264053	0.176575	1.000000	0.119777	0.108407	-0.003124	0.160211	0.15
heart_disease	0.082061	0.244279	0.128833	0.119777	1.000000	0.079233	-0.002743	0.146938	0.05
occupation	-0.036784	0.433214	0.367858	0.108407	0.079233	1.000000	-0.003625	0.095049	0.24
residence	0.001508	-0.000605	0.004422	-0.003124	-0.002743	-0.003625	1.000000	0.000014	-0.00
metric_1	0.035465	0.226538	0.153607	0.160211	0.146938	0.095049	0.000014	1.000000	0.18
metric_2	-0.021570	0.337114	0.337517	0.153779	0.054133	0.245115	-0.003685	0.184199	1.00
metric_3	-0.008886	0.000002	0.005611	-0.002164	-0.006168	-0.001977	-0.007895	-0.008735	-0.00
metric_4	0.002863	-0.005108	-0.001890	0.012179	0.001507	0.002514	-0.005431	-0.005511	0.00
metric_5	0.035465	0.226538	0.153607	0.160211	0.146938	0.095049	0.000014	1.000000	0.18
smoking_status	0.042775	-0.365058	-0.303543	-0.118643	-0.066340	-0.242066	0.001532	-0.096956	-0.24
stroke	0.011198	0.149678	0.071920	0.075332	0.113763	0.045946	0.002247	0.078917	0.01

In [154]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
#We extract the predcitor variables and store them on z
z= stroke_data[['gender','age','married','hypertension','heart_disease','occupation','residenc
#train the model
lm.fit(z,stroke_data['stroke'])
yhat  = lm.predict(z)
```

In [155]:
```
lm.intercept_
```

Out[155]:
```
0.05422372147345281
```

In [156]:
```
lm.coef_
```

Out[156]:
```
array([ 1.32415282e-03,  8.92195207e-04, -1.00038464e-02,  1.47136886e-02,
        4.56630947e-02, -1.25847674e-03,  6.80060867e-04,  6.29409450e-05,
       -5.96843347e-04, -7.15825281e-04, -6.86772167e-04,  6.29409450e-05,
        7.43177934e-04])
```

In [157]:
```
lm.score(z,stroke_data['stroke'])
```

Out[157]:
```
0.0335719248062033
```