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

%matplotlib inline
sns.set(rc={'figure.figsize': [20, 20]}, font_scale=1.4)


from google.colab import files
import pandas as pd

# Upload the file from your local machine to Colab
uploaded = files.upload()

# Assuming the uploaded file is named 'hypothyroid.csv'
file_path = 'hypothyroid.csv'

# Read the CSV file into a Pandas DataFrame
df = pd.read_csv(file_path)

# Display the DataFrame
df
```

Choose Files hypothyroid.csv

- **hypothyroid.csv**(text/csv) - 279880 bytes, last modified: 10/9/2023 - 100% done

```
Saving hypothyroid.csv to hypothyroid (1).csv
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment	
0	41	F	f	f	f	f	f	f	f	
1	23	F	f	f	f	f	f	f	f	
2	46	M	f	f	f	f	f	f	f	
3	70	F	t	f	f	f	f	f	f	
4	70	F	f	f	f	f	f	f	f	
...	...	...	...	...	...	...	...	...	...	
3767	30	F	f	f	f	f	f	f	f	
3768	68	F	f	f	f	f	f	f	f	
3769	74	F	f	f	f	f	f	f	f	
3770	72	M	f	f	f	f	f	f	f	
3771	64	F	f	f	f	f	f	f	f	

3772 rows × 30 columns

```
df.head()
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment	hy
0	41	F	f	f	f	f	f	f	f	
1	23	F	f	f	f	f	f	f	f	
2	46	M	f	f	f	f	f	f	f	
3	70	F	t	f	f	f	f	f	f	
4	70	F	f	f	f	f	f	f	f	

5 rows × 30 columns

```
df.describe().T
```

	count	unique	top	freq
age	3772	94	59	95
sex	3772	3	F	2480
on thyroxine	3772	2	f	3308
query on thyroxine	3772	2	f	3722
on antithyroid medication	3772	2	f	3729
sick	3772	2	f	3625
pregnant	3772	2	f	3719
thyroid surgery	3772	2	f	3719
I131 treatment	3772	2	f	3713
query hypothyroid	3772	2	f	3538
query hyperthyroid	3772	2	f	3535
lithium	3772	2	f	3754
goitre	3772	2	f	3738
tumor	3772	2	f	3676
hypopituitary	3772	2	f	3771
psych	3772	2	f	3588
TSH measured	3772	2	t	3403
TSH	3772	288	?	369
T3 measured	3772	2	t	3003
T3	3772	70	?	769
TT4 measured	3772	2	t	3541
TT4	3772	242	?	231
T4U measured	3772	2	t	3385
T4U	3772	147	?	387
FTI measured	3772	2	t	3387
FTI	3772	235	?	385

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3772 entries, 0 to 3771
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   age                        3772 non-null   object
 1   sex                        3772 non-null   object
 2   on thyroxine               3772 non-null   object
 3   query on thyroxine         3772 non-null   object
 4   on antithyroid medication  3772 non-null   object
 5   sick                       3772 non-null   object
 6   pregnant                   3772 non-null   object
 7   thyroid surgery            3772 non-null   object
 8   I131 treatment             3772 non-null   object
 9   query hypothyroid          3772 non-null   object
 10  query hyperthyroid         3772 non-null   object
 11  lithium                    3772 non-null   object
 12  goitre                     3772 non-null   object
 13  tumor                      3772 non-null   object
 14  hypopituitary              3772 non-null   object
 15  psych                      3772 non-null   object
 16  TSH measured               3772 non-null   object
 17  TSH                        3772 non-null   object
 18  T3 measured                3772 non-null   object
 19  T3                         3772 non-null   object
 20  TT4 measured               3772 non-null   object
 21  TT4                        3772 non-null   object
 22  T4U measured               3772 non-null   object
 23  T4U                        3772 non-null   object
 24  FTI measured               3772 non-null   object
 25  FTI                        3772 non-null   object
 26  TBG measured               3772 non-null   object
 27  TBG                        3772 non-null   object
 28  referral source            3772 non-null   object
 29  binaryClass                3772 non-null   object
dtypes: object(30)
memory usage: 884.2+ KB
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41	F	f	f	f	f	f	f	f
1	23	F	f	f	f	f	f	f	f
2	46	M	f	f	f	f	f	f	f
3	70	F	t	f	f	f	f	f	f
4	70	F	f	f	f	f	f	f	f
...	...	...	...	...	...	...	...	...	...
3767	30	F	f	f	f	f	f	f	f
3768	68	F	f	f	f	f	f	f	f
3769	74	F	f	f	f	f	f	f	f
3770	72	M	f	f	f	f	f	f	f
3771	64	F	f	f	f	f	f	f	f

3772 rows × 30 columns

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

```
P    3481
N     291
Name: binaryClass, dtype: int64
```

```
df["binaryClass"]=df["binaryClass"].map({"P":0,"N":1})
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41	F	f	f	f	f	f	f	f
1	23	F	f	f	f	f	f	f	f
2	46	M	f	f	f	f	f	f	f
3	70	F	t	f	f	f	f	f	f
4	70	F	f	f	f	f	f	f	f
...	...	...	...	...	...	...	...	...	...
3767	30	F	f	f	f	f	f	f	f
3768	68	F	f	f	f	f	f	f	f
3769	74	F	f	f	f	f	f	f	f
3770	72	M	f	f	f	f	f	f	f
3771	64	F	f	f	f	f	f	f	f

3772 rows × 30 columns

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

```
f    3719
t      53
Name: pregnant, dtype: int64
```

```
df=df.replace({"t":1,"f":0})
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41	F	0	0	0	0	0	0	0
1	23	F	0	0	0	0	0	0	0
2	46	M	0	0	0	0	0	0	0
3	70	F	1	0	0	0	0	0	0
4	70	F	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30	F	0	0	0	0	0	0	0
3768	68	F	0	0	0	0	0	0	0

```
#df['target'].isnull().sum()
```

```
df['sex'].isnull().sum()
```

```
0
```

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

```
?    3772
Name: TBG, dtype: int64
```

```
del df["TBG"]
```

```
df=df.replace({"?":np.NAN})
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41	F	0	0	0	0	0	0	0
1	23	F	0	0	0	0	0	0	0
2	46	M	0	0	0	0	0	0	0
3	70	F	1	0	0	0	0	0	0
4	70	F	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30	F	0	0	0	0	0	0	0
3768	68	F	0	0	0	0	0	0	0
3769	74	F	0	0	0	0	0	0	0
3770	72	M	0	0	0	0	0	0	0
3771	64	F	0	0	0	0	0	0	0

3772 rows × 29 columns

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

```
age                          1
sex                        150
on thyroxine                 0
query on thyroxine           0
on antithyroid medication    0
sick                         0
pregnant                     0
thyroid surgery              0
I131 treatment               0
query hypothyroid            0
query hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH measured                 0
TSH                        369
```

```
T3 measured                  0
T3                         769
TT4 measured                 0
TT4                        231
T4U measured                 0
T4U                        387
FTI measured                 0
FTI                        385
TBG measured                 0
referral source              0
binaryClass                  0
dtype: int64
```

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

```
F    2480
M    1142
Name: sex, dtype: int64
```

```
df=df.replace({"F":1,"M":0})
```

```
df["referral source"].value_counts()
```

```
other    2201
SVI      1034
SVHC      386
STMW      112
SVHD       39
Name: referral source, dtype: int64
```

```
del df["referral source"]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3772 entries, 0 to 3771
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3771 non-null   object
 1   sex                        3622 non-null   float64
 2   on thyroxine               3772 non-null   int64
 3   query on thyroxine         3772 non-null   int64
 4   on antithyroid medication  3772 non-null   int64
 5   sick                       3772 non-null   int64
 6   pregnant                   3772 non-null   int64
 7   thyroid surgery            3772 non-null   int64
 8   I131 treatment             3772 non-null   int64
 9   query hypothyroid          3772 non-null   int64
 10  query hyperthyroid         3772 non-null   int64
 11  lithium                    3772 non-null   int64
 12  goitre                     3772 non-null   int64
 13  tumor                      3772 non-null   int64
 14  hypopituitary              3772 non-null   int64
 15  psych                      3772 non-null   int64
 16  TSH measured               3772 non-null   int64
 17  TSH                        3403 non-null   object
 18  T3 measured                3772 non-null   int64
 19  T3                         3003 non-null   object
 20  TT4 measured               3772 non-null   int64
 21  TT4                        3541 non-null   object
 22  T4U measured               3772 non-null   int64
 23  T4U                        3385 non-null   object
 24  FTI measured               3772 non-null   int64
 25  FTI                        3387 non-null   object
 26  TBG measured               3772 non-null   int64
 27  binaryClass                3772 non-null   int64
dtypes: float64(1), int64(21), object(6)
memory usage: 825.2+ KB
```

```
df["T3 measured"].value_counts()
```

```
1    3003
0     769
Name: T3 measured, dtype: int64
```

```
df["TT4 measured"].value_counts()
```

```
1    3541
0     231
Name: TT4 measured, dtype: int64
```

```
df["FTI measured"].value_counts()
```

```
1    3387
0     385
Name: FTI measured, dtype: int64
```

```
df["TBG measured"].value_counts()
```

```
0    3772
Name: TBG measured, dtype: int64
```

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

```
0    3481
1     291
Name: binaryClass, dtype: int64
```

```
df.dtypes
```

```
age                          object
sex                         float64
on thyroxine                  int64
query on thyroxine            int64
on antithyroid medication     int64
sick                          int64
pregnant                      int64
thyroid surgery               int64
I131 treatment                int64
query hypothyroid             int64
query hyperthyroid            int64
lithium                       int64
goitre                        int64
tumor                         int64
hypopituitary                 int64
psych                         int64
TSH measured                  int64
TSH                          object
T3 measured                   int64
T3                           object
TT4 measured                  int64
TT4                          object
T4U measured                  int64
T4U                          object
FTI measured                  int64
FTI                          object
TBG measured                  int64
binaryClass                   int64
dtype: object
```

```
cols = df.columns[df.dtypes.eq('object')]
df[cols] = df[cols].apply(pd.to_numeric, errors='coerce')
df.dtypes
```

```
age                         float64
sex                         float64
on thyroxine                  int64
query on thyroxine            int64
on antithyroid medication     int64
sick                          int64
pregnant                      int64
thyroid surgery               int64
I131 treatment                int64
query hypothyroid             int64
query hyperthyroid            int64
lithium                       int64
goitre                        int64
tumor                         int64
hypopituitary                 int64
psych                         int64
TSH measured                  int64
TSH                         float64
T3 measured                   int64
T3                          float64
TT4 measured                  int64
TT4                         float64
T4U measured                  int64
T4U                         float64
FTI measured                  int64
FTI                         float64
TBG measured                  int64
binaryClass                   int64
dtype: object
```

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

```
age                          1
sex                        150
on thyroxine                 0
query on thyroxine           0
on antithyroid medication    0
sick                         0
pregnant                     0
thyroid surgery              0
I131 treatment               0
query hypothyroid            0
query hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH measured                 0
TSH                        369
T3 measured                  0
T3                         769
TT4 measured                 0
TT4                        231
T4U measured                 0
T4U                        387
FTI measured                 0
FTI                        385
TBG measured                 0
binaryClass                  0
dtype: int64
```

```
df['T4U measured'].mean()
```

```
0.8974019088016967
```

```
df['T4U measured'].fillna(df['T4U measured'].mean(), inplace=True)
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41.0	1.0	0	0	0	0	0	0	0
1	23.0	1.0	0	0	0	0	0	0	0
2	46.0	0.0	0	0	0	0	0	0	0
3	70.0	1.0	1	0	0	0	0	0	0
4	70.0	1.0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30.0	1.0	0	0	0	0	0	0	0
3768	68.0	1.0	0	0	0	0	0	0	0
3769	74.0	1.0	0	0	0	0	0	0	0
3770	72.0	0.0	0	0	0	0	0	0	0
3771	64.0	1.0	0	0	0	0	0	0	0

3772 rows × 28 columns

```
df['sex'].fillna(df['sex'].mean(), inplace=True)
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41.0	1.0	0	0	0	0	0	0	0
1	23.0	1.0	0	0	0	0	0	0	0
2	46.0	0.0	0	0	0	0	0	0	0
3	70.0	1.0	1	0	0	0	0	0	0
4	70.0	1.0	0	0	0	0	0	0	0

```
df['age'].fillna(df['age'].mean(), inplace=True)
```

3767	30.0	1.0	0	0	0	0	0	0	0

```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='mean')
```

3770	72.0	0.0	0	0	0	0	0	0	0

```
df['TSH'] = imputer.fit_transform(df[['TSH']])

df['T3'] = imputer.fit_transform(df[['T3']])

df['TT4'] = imputer.fit_transform(df[['TT4']])

df['T4U'] = imputer.fit_transform(df[['T4U']])

df['FTI'] = imputer.fit_transform(df[['FTI']])

df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41.0	1.0	0	0	0	0	0	0	0
1	23.0	1.0	0	0	0	0	0	0	0
2	46.0	0.0	0	0	0	0	0	0	0
3	70.0	1.0	1	0	0	0	0	0	0
4	70.0	1.0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30.0	1.0	0	0	0	0	0	0	0
3768	68.0	1.0	0	0	0	0	0	0	0
3769	74.0	1.0	0	0	0	0	0	0	0
3770	72.0	0.0	0	0	0	0	0	0	0
3771	64.0	1.0	0	0	0	0	0	0	0

3772 rows × 28 columns

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

```
age                          0
sex                          0
on thyroxine                 0
query on thyroxine           0
on antithyroid medication    0
sick                         0
pregnant                     0
thyroid surgery              0
I131 treatment               0
query hypothyroid            0
query hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH measured                 0
TSH                          0
T3 measured                  0
T3                           0
```

```
TT4 measured                0
TT4                         0
T4U measured                0
T4U                         0
FTI measured                0
FTI                         0
TBG measured                0
binaryClass                 0
dtype: int64
```

```
df
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41.0	1.0	0	0	0	0	0	0	0
1	23.0	1.0	0	0	0	0	0	0	0
2	46.0	0.0	0	0	0	0	0	0	0
3	70.0	1.0	1	0	0	0	0	0	0
4	70.0	1.0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30.0	1.0	0	0	0	0	0	0	0
3768	68.0	1.0	0	0	0	0	0	0	0
3769	74.0	1.0	0	0	0	0	0	0	0
3770	72.0	0.0	0	0	0	0	0	0	0
3771	64.0	1.0	0	0	0	0	0	0	0

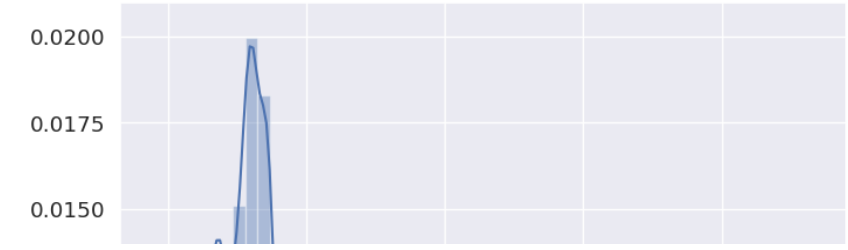
3772 rows × 28 columns

```
df.columns
```

```
Index(['age', 'sex', 'on thyroxine', 'query on thyroxine',
       'on antithyroid medication', 'sick', 'pregnant', 'thyroid surgery',
       'I131 treatment', 'query hypothyroid', 'query hyperthyroid', 'lithium',
       'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH measured', 'TSH',
       'T3 measured', 'T3', 'TT4 measured', 'TT4', 'T4U measured', 'T4U',
       'FTI measured', 'FTI', 'TBG measured', 'binaryClass'],
      dtype='object')
```

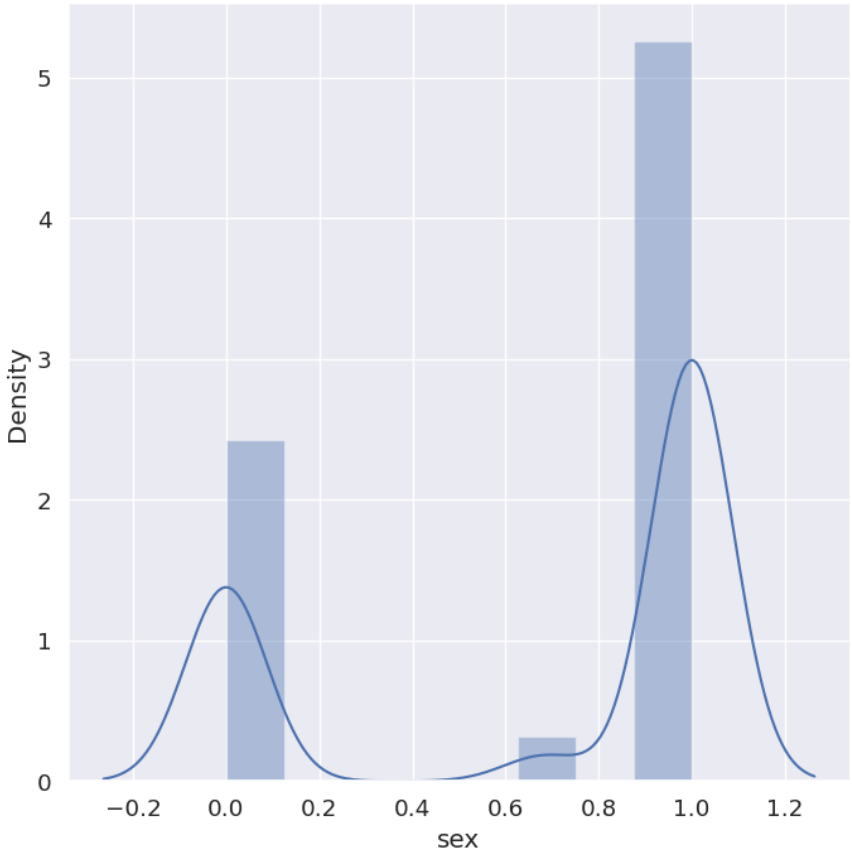
```
import seaborn as sns

%matplotlib inline
sns.set(rc={'figure.figsize': [8, 8]}, font_scale=1.2)


sns.distplot(df['age'])
```

```
<ipython-input-58-7452d86f8334>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['age'])
<Axes: xlabel='age', ylabel='Density'>
```



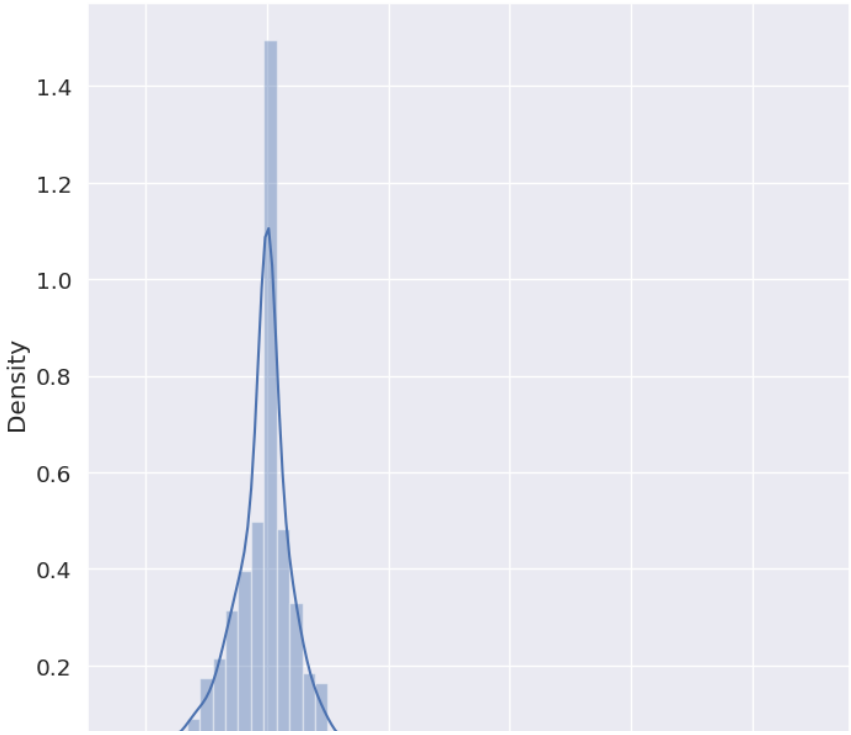

```
sns.distplot(df['sex'])
```

```
<ipython-input-59-6434e6ccc7f4>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['sex'])
<Axes: xlabel='sex', ylabel='Density'>
```




```
sns.distplot(df['T3'])
```

```
<ipython-input-60-9a4ccb1c10c1>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['T3'])
<Axes: xlabel='T3', ylabel='Density'>
```

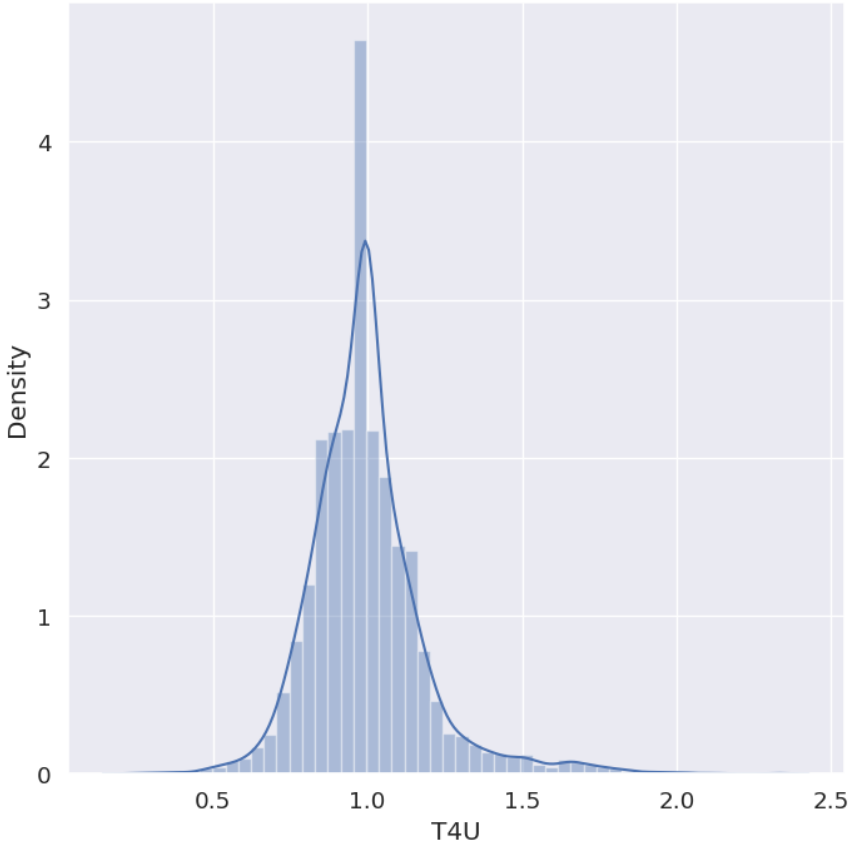



```
sns.distplot(df['TT4'])
```

```
<ipython-input-61-c3341520245c>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

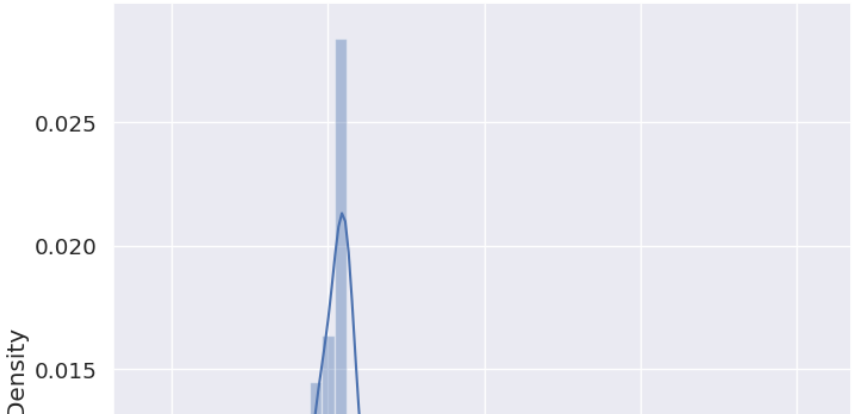
```
sns.distplot(df['T4U'])
```

```
<ipython-input-62-87b814d510ec>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['T4U'])
<Axes: xlabel='T4U', ylabel='Density'>
```



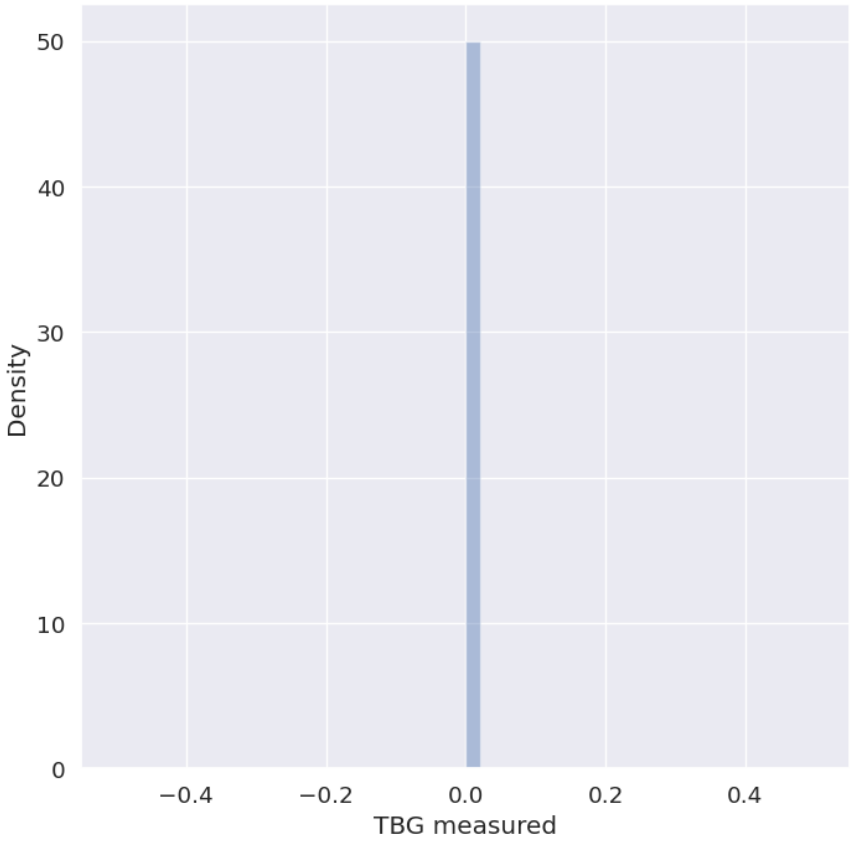

```
sns.distplot(df['FTI'])
```

```
<ipython-input-63-52884c87dd85>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['FTI'])
<Axes: xlabel='FTI', ylabel='Density'>
```



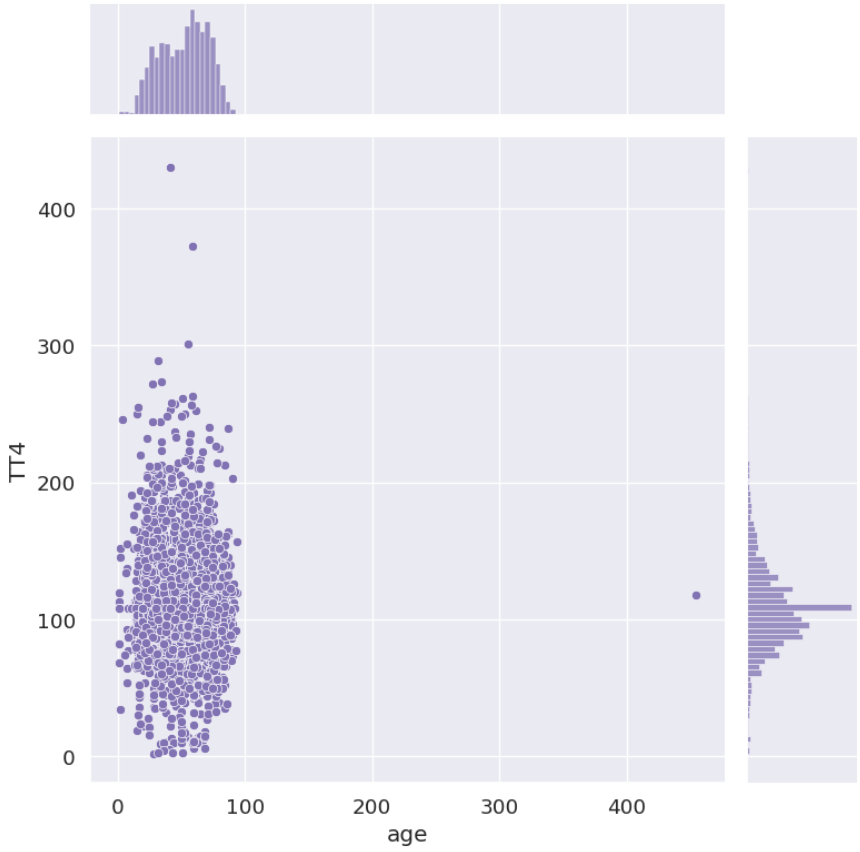

```
sns.distplot(df['TBG measured'])
```

```
<ipython-input-64-c1847644a7e5>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['TBG measured'])
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511: UserWarning: D
  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<Axes: xlabel='TBG measured', ylabel='Density'>
```



```
sns.jointplot(x='age', y='TT4', data=df, kind='scatter', height=8, color='m')
```

```
<seaborn.axisgrid.JointGrid at 0x7edf956e3070>
```

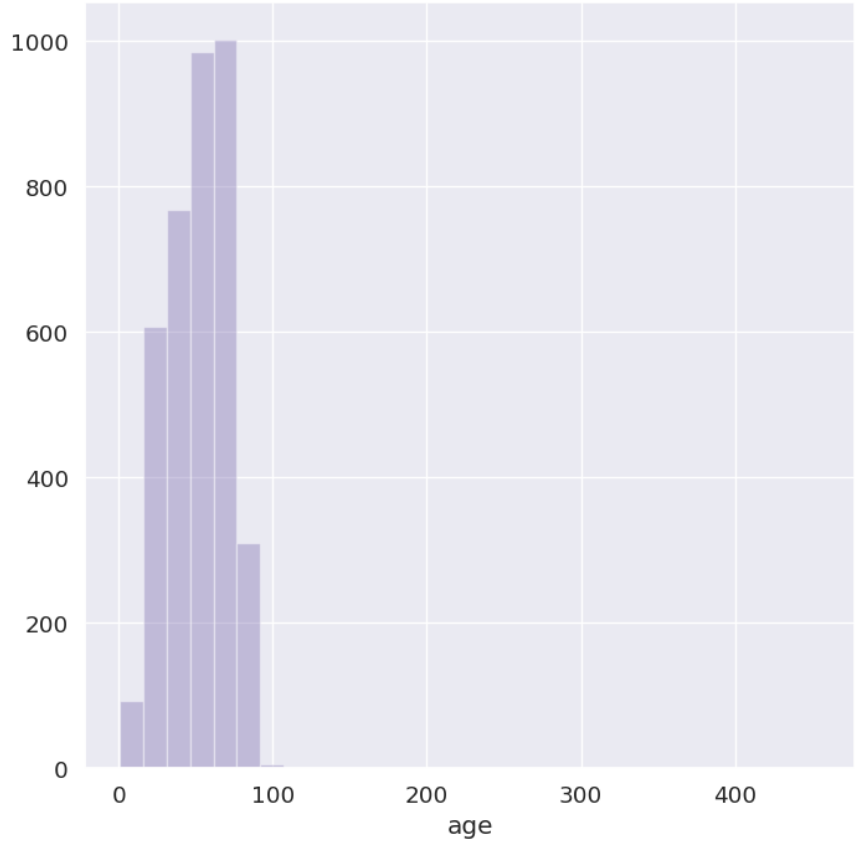



```
sns.jointplot(x='age', y='TT4', data=df, kind='reg', height=8, color='m')
```

```
<seaborn.axisgrid.JointGrid at 0x7edf956467d0>
```

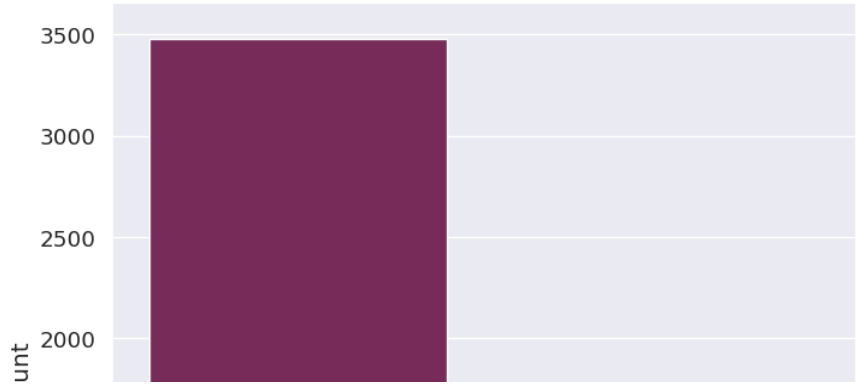
```
sns.distplot(df['age'], kde=False, bins=30, color='m')
```

```
<ipython-input-67-40b41b1df846>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['age'], kde=False, bins=30, color='m')
<Axes: xlabel='age'>
```



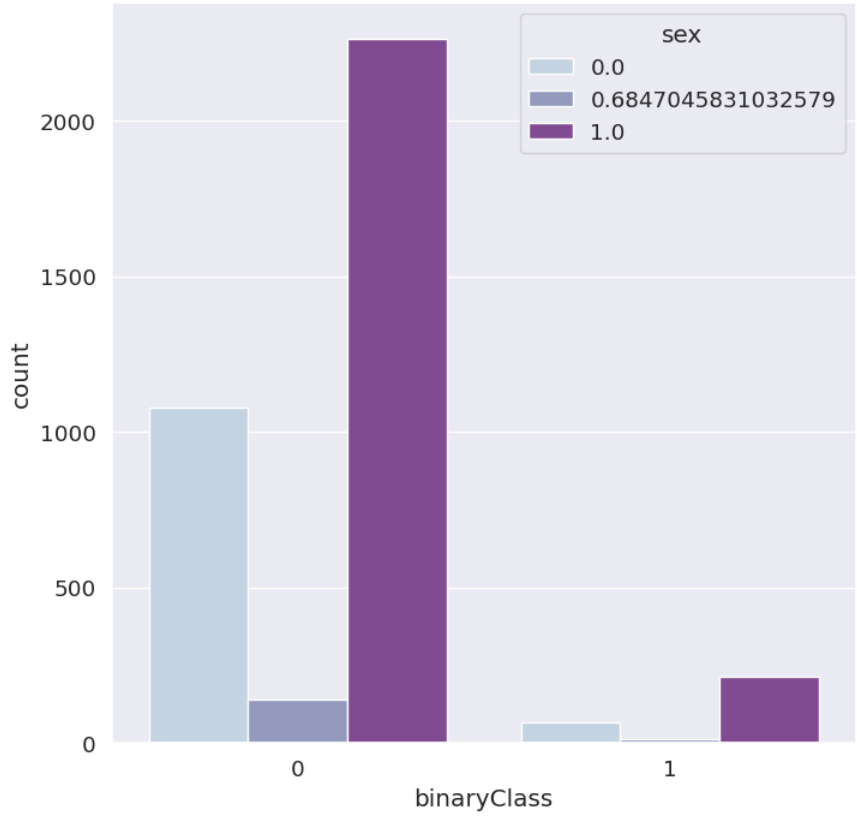

```
sns.countplot(x='binaryClass', data=df, palette='rocket')
```

```
<Axes: xlabel='binaryClass', ylabel='count'>
```




```
sns.countplot(x='binaryClass', data=df, hue='sex', palette='BuPu')
```

```
<Axes: xlabel='binaryClass', ylabel='count'>
```



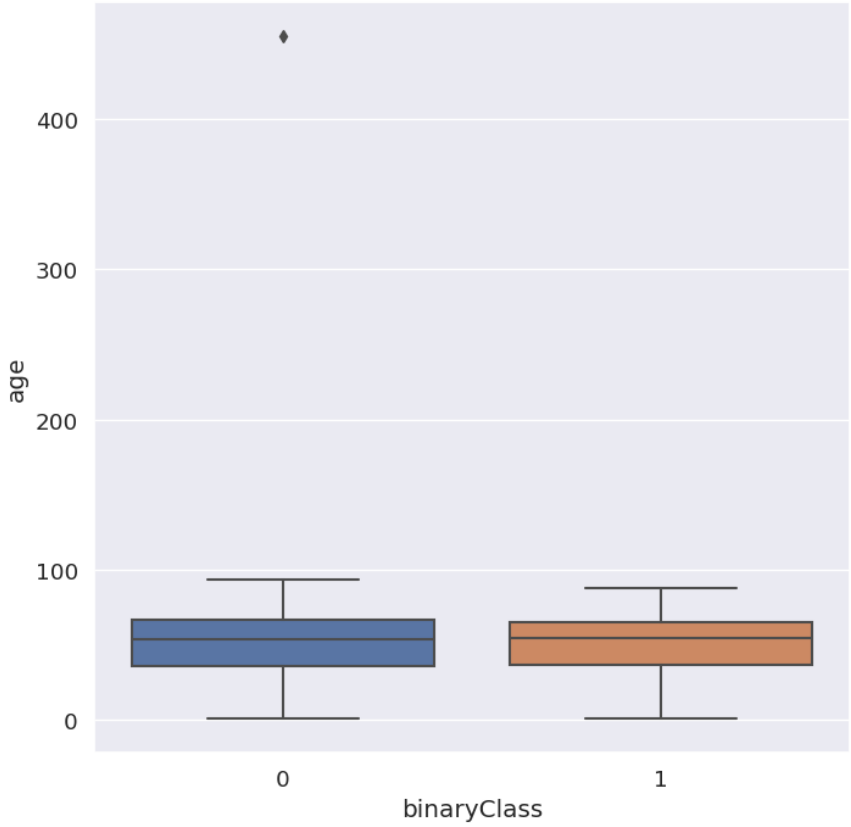

```
sns.stripplot(x="binaryClass", y="age", data=df, palette="viridis")
```

```
<ipython-input-70-70359f1e7388>:1: FutureWarning: Passing `palette` without assigning
  sns.stripplot(x="binaryClass", y="age", data=df, palette="viridis")
<Axes: xlabel='binaryClass', ylabel='age'>
```



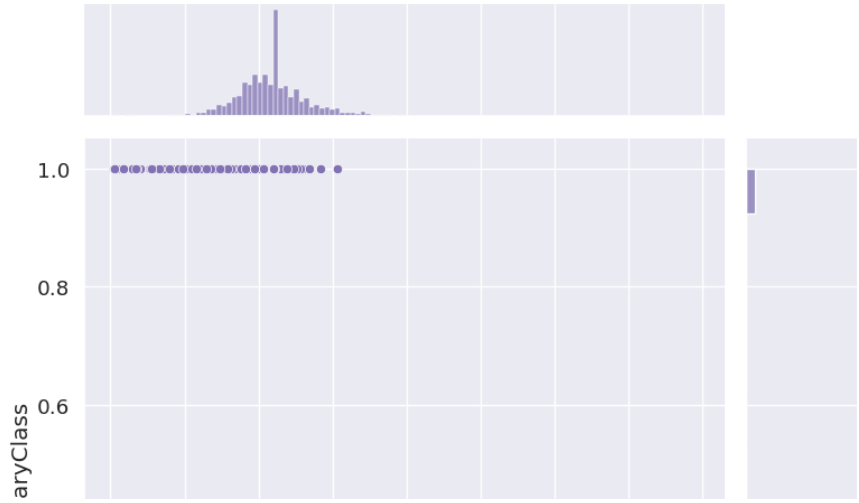
```
sns.boxplot(x='binaryClass', y='age', data=df)
```

```
<Axes: xlabel='binaryClass', ylabel='age'>
```




```
sns.jointplot(x='FTI', y='binaryClass', data=df, kind='scatter', height=8, color='m')
```

```
<seaborn.axisgrid.JointGrid at 0x7edf937557e0>
```

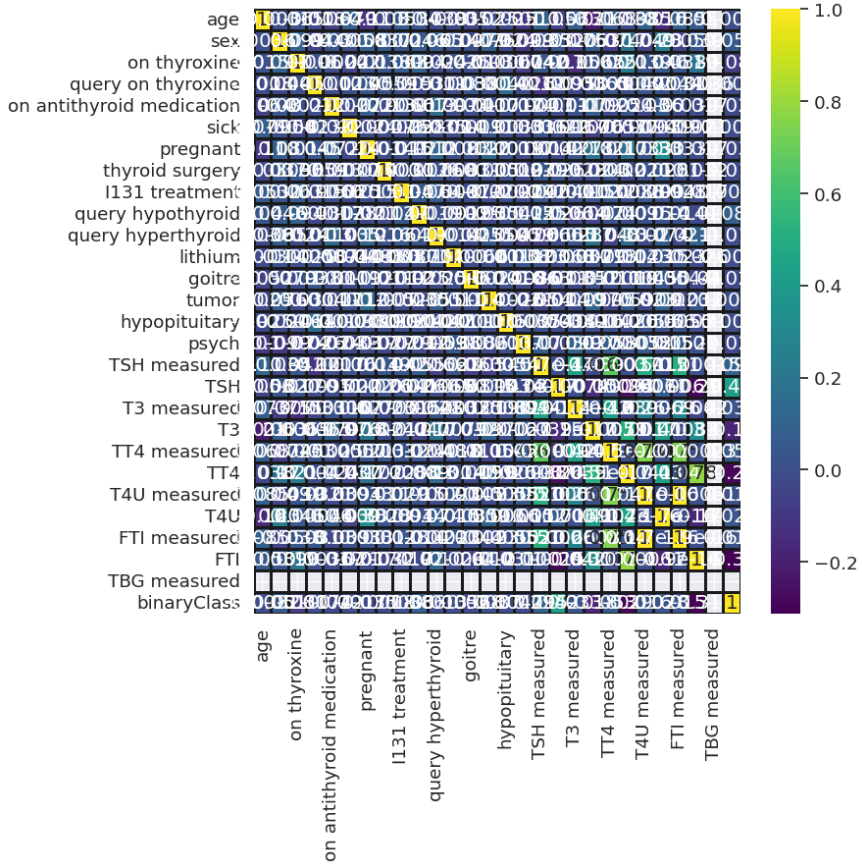



```
df_corr = df.corr()
df_corr
```

treatment							
query hypothyroid	0.039562	0.046353	0.094412	-0.029808	-0.017264	0.027718	-0.0
query hyperthyroid	-0.038054	0.065274	-0.023796	-0.010905	0.126566	-0.035206	0.
lithium	-0.030126	0.014155	-0.002509	-0.008026	-0.007436	-0.013944	-0.0
goitre	-0.051830	-0.007886	-0.010098	0.038000	-0.010241	-0.019205	0.0
tumor	-0.025037	0.076303	-0.029773	-0.004011	-0.017353	0.010949	0.
hypopituitary	-0.024927	-0.024489	-0.006099	0.140500	-0.001749	-0.003279	-0.0
psych	-0.100116	-0.098832	-0.073571	-0.026247	-0.024318	-0.032883	-0.0
TSH measured	0.105131	-0.035249	0.041818	-0.117891	0.001736	0.015588	0.0

```
sns.heatmap(df_corr, cmap='viridis', linecolor='k', linewidths=2, annot=True)
```

```
<Axes: >
```




```
#df.dropna(axis = 0, inplace=True)
```

```
x = df.drop('binaryClass', axis=1)
y = df['binaryClass']
```

```
x
```

	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment
0	41.0	1.0	0	0	0	0	0	0	0
1	23.0	1.0	0	0	0	0	0	0	0
2	46.0	0.0	0	0	0	0	0	0	0
3	70.0	1.0	1	0	0	0	0	0	0
4	70.0	1.0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
3767	30.0	1.0	0	0	0	0	0	0	0
3768	68.0	1.0	0	0	0	0	0	0	0
3769	74.0	1.0	0	0	0	0	0	0	0
3770	72.0	0.0	0	0	0	0	0	0	0
3771	64.0	1.0	0	0	0	0	0	0	0

3772 rows × 27 columns

```
y
```

```
0       0
1       0
2       0
3       0
4       0
       ..
3767    0
3768    0
3769    0
3770    0
3771    0
Name: binaryClass, Length: 3772, dtype: int64
```

```
x = sm.add_constant(x)
results = sm.OLS(y,x).fit()
results.summary()
```

Model:	OLS	**Adj. R-squared:**	0.247
Method:	Least Squares	**F-statistic:**	48.57
Date:	Tue, 14 Nov 2023	**Prob (F-statistic):**	1.41e-213
Time:	17:08:45	**Log-Likelihood:**	179.17
No. Observations:	3772	**AIC:**	-304.3
Df Residuals:	3745	**BIC:**	-136.0
Df Model:	26		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0334	0.058	0.578	0.563	-0.080	0.147
age	-4.813e-05	0.000	-0.242	0.809	-0.000	0.000
sex	0.0343	0.009	3.943	0.000	0.017	0.051
on thyroxine	-0.0530	0.012	-4.339	0.000	-0.077	-0.029
query on thyroxine	0.0377	0.034	1.109	0.268	-0.029	0.104
on antithyroid medication	-0.0469	0.036	-1.299	0.194	-0.118	0.024

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
```

```
x.shape
```

```
(3772, 28)
```

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
sc.fit(x_train)
x_train = sc.transform(x_train)
x_test = sc.transform(x_test)
```

TT4	-0.0016	0.000	-3.641	0.000	-0.002	-0.001

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlyStopping
```

TBG measured	0	0	nan	nan	0	0

```
x.shape
```

```
(3772, 28)
```

Kurtosis:	11.221	**Cond. No.**	1.03e+16

```
x.shape[1]
```

```
28
```

[2] The smallest eigenvalue is 1e-24. This might indicate that there are

```
model = Sequential()
model.add(Dense(256, input_shape=[x.shape[1]], activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(63, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(1, activation='sigmoid'))


model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 256)               7424

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 63)                8127

 dropout_2 (Dropout)         (None, 63)                0

 dense_3 (Dense)             (None, 1)                 64

=================================================================
Total params: 48511 (189.50 KB)
```

```
Trainable params: 48511 (189.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy'])
```

```
lrd = ReduceLROnPlateau(monitor = 'val_loss',
                        patience = 20,
                        verbose = 1,
                        factor = 0.75,
                        min_lr = 1e-10)

mcp = ModelCheckpoint('model.h5')

es = EarlyStopping(verbose=1, patience=20)
```

```
%%time
history = model.fit(x=x_train, y=y_train, epochs=100, callbacks=[lrd, mcp, es], batch_size=64, validation_split=0.1)
```

```
Epoch 27/100
40/40 [==============================] - 0s 11ms/step - loss: 0.0307 - accuracy: 0.9882 - val_loss: 0.0259 - val_accuracy: 0.989
Epoch 28/100
40/40 [==============================] - 0s 12ms/step - loss: 0.0309 - accuracy: 0.9878 - val_loss: 0.0330 - val_accuracy: 0.985
Epoch 29/100
40/40 [==============================] - 1s 15ms/step - loss: 0.0269 - accuracy: 0.9890 - val_loss: 0.0197 - val_accuracy: 0.992
Epoch 30/100
40/40 [==============================] - 0s 11ms/step - loss: 0.0311 - accuracy: 0.9859 - val_loss: 0.0339 - val_accuracy: 0.985
Epoch 31/100
40/40 [==============================] - 0s 11ms/step - loss: 0.0225 - accuracy: 0.9906 - val_loss: 0.0249 - val_accuracy: 0.989
Epoch 32/100
40/40 [==============================] - 0s 12ms/step - loss: 0.0234 - accuracy: 0.9902 - val_loss: 0.0115 - val_accuracy: 0.996
Epoch 33/100
40/40 [==============================] - 0s 11ms/step - loss: 0.0223 - accuracy: 0.9914 - val_loss: 0.0259 - val_accuracy: 0.992
Epoch 34/100
40/40 [==============================] - 0s 11ms/step - loss: 0.0273 - accuracy: 0.9870 - val_loss: 0.0233 - val_accuracy: 0.989
Epoch 35/100
40/40 [==============================] - 0s 12ms/step - loss: 0.0193 - accuracy: 0.9918 - val_loss: 0.0234 - val_accuracy: 0.992
Epoch 36/100
40/40 [==============================] - 0s 9ms/step - loss: 0.0219 - accuracy: 0.9918 - val_loss: 0.0229 - val_accuracy: 0.9929
Epoch 37/100
40/40 [==============================] - 0s 7ms/step - loss: 0.0211 - accuracy: 0.9910 - val_loss: 0.0218 - val_accuracy: 0.9929
Epoch 38/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0203 - accuracy: 0.9921 - val_loss: 0.0158 - val_accuracy: 0.9929
Epoch 39/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0326 - accuracy: 0.9898 - val_loss: 0.0302 - val_accuracy: 0.9859
Epoch 40/100
40/40 [==============================] - 0s 7ms/step - loss: 0.0165 - accuracy: 0.9945 - val_loss: 0.0141 - val_accuracy: 0.9894
Epoch 41/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0187 - accuracy: 0.9914 - val_loss: 0.0217 - val_accuracy: 0.9894
Epoch 42/100
40/40 [==============================] - 0s 7ms/step - loss: 0.0186 - accuracy: 0.9910 - val_loss: 0.0214 - val_accuracy: 0.9929
Epoch 43/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0126 - accuracy: 0.9945 - val_loss: 0.0287 - val_accuracy: 0.9859
Epoch 44/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0187 - accuracy: 0.9941 - val_loss: 0.0279 - val_accuracy: 0.9894
Epoch 45/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0168 - accuracy: 0.9933 - val_loss: 0.0416 - val_accuracy: 0.9859
Epoch 46/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0187 - accuracy: 0.9921 - val_loss: 0.0256 - val_accuracy: 0.9894
Epoch 47/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0142 - accuracy: 0.9945 - val_loss: 0.0303 - val_accuracy: 0.9929
Epoch 48/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0153 - accuracy: 0.9925 - val_loss: 0.0313 - val_accuracy: 0.9894
Epoch 49/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0159 - accuracy: 0.9941 - val_loss: 0.0219 - val_accuracy: 0.9929
Epoch 50/100
40/40 [==============================] - 0s 7ms/step - loss: 0.0183 - accuracy: 0.9929 - val_loss: 0.0325 - val_accuracy: 0.9859
Epoch 51/100
40/40 [==============================] - 0s 8ms/step - loss: 0.0132 - accuracy: 0.9945 - val_loss: 0.0288 - val_accuracy: 0.9929
Epoch 52/100
38/40 [===========================>..] - ETA: 0s - loss: 0.0128 - accuracy: 0.9951
Epoch 52: ReduceLROnPlateau reducing learning rate to 0.0007500000356230885.
40/40 [==============================] - 0s 8ms/step - loss: 0.0131 - accuracy: 0.9949 - val_loss: 0.0299 - val_accuracy: 0.9929
Epoch 52: early stopping
CPU times: user 20.1 s, sys: 903 ms, total: 21 s
Wall time: 20.8 s
```

```
model.evaluate(x_test, y_test)
```

```
30/30 [==============================] - 0s 3ms/step - loss: 0.0433 - accuracy: 0.9883
[0.04327028617262844, 0.9883350729942322]
```

```
y_pred = model.predict(x_test)
y_pred
```

```
            [5.70701371e-29],
            [6.51854797e-25],
            [5.16321706e-22],
            [5.95221024e-07],
            [2.47019272e-09],
            [2.71771417e-28],
            [1.21100732e-14],
            [1.20504509e-07],
            [1.00146172e-12],
            [6.68289388e-08],
            [5.61507563e-10],
            [9.99994040e-01],
            [5.37740981e-08],
            [2.53306462e-05],
            [3.03323790e-11],
            [4.54140967e-03],
            [1.31353912e-13],
            [5.43730110e-08],
            [7.32016875e-18],
            [1.11064730e-06],
            [9.66949350e-12],
            [3.62131300e-06],
            [2.76663247e-03],
            [1.90013061e-10],
            [3.17923509e-12],
            [2.54073602e-05],
            [3.39370079e-13],
            [5.76961895e-07],
            [7.41808760e-23],
            [4.76731188e-08],
            [4.69366372e-33],
            [2.90006092e-11],
            [1.36708722e-09],
            [4.03634112e-06],
            [2.63469178e-13],
            [1.16102347e-18],
            [2.36494463e-10],
            [4.84253046e-13],
            [5.29808233e-07],
            [6.20703626e-15],
            [1.77718834e-20],
            [9.95912850e-01],
            [7.44292805e-09],
            [3.17338600e-10],
            [6.02055561e-26],
            [1.78825008e-08],
            [2.50177127e-05],
            [9.92330246e-11],
            [2.53617114e-13],
            [1.51465950e-03],
            [1.65819823e-11],
            [4.89526315e-07],
            [1.53883288e-08],
            [7.24283982e-14],
            [1.81170049e-06],
            [2.43623690e-20],
            [1.88468424e-19],
            [6.98789707e-11]], dtype=float32)
```

```
y_pred[1]
```

```
array([0.12389065], dtype=float32)
```

```
y_test
```

```
270     0
2071    1
3660    0
3207    0
1675    0
       ..
1791    0
2132    0
162     0
1885    0
219     0
Name: binaryClass, Length: 943, dtype: int64
```

```
%%time
model.evaluate(x_test, y_test)
```

```
30/30 [==============================] - 0s 3ms/step - loss: 0.0433 - accuracy: 0.9883
CPU times: user 161 ms, sys: 9.4 ms, total: 170 ms
```

```
Wall time: 172 ms
[0.0432702861726284, 0.9883350729942322]
```

```
model.predict(x_test)[1]
```

```
30/30 [==============================] - 0s 2ms/step
array([0.12389065], dtype=float32)
```

```
# model.predict_classes(x_test)[1]
```

```
predictions = (model.predict(x_test) > 0.5).astype("int32")
```

```
30/30 [==============================] - 0s 2ms/step
```

```
y_test.iloc[1]
```

```
1
```

```
model.predict(x_test)[70]
```

```
30/30 [==============================] - 0s 2ms/step
array([2.3590855e-27], dtype=float32)
```

```
# model.predict_classes(x_test)[70]
```

```
predictions = (model.predict(x_test) > 0.5).astype("int32")
```

```
30/30 [==============================] - 0s 2ms/step
```

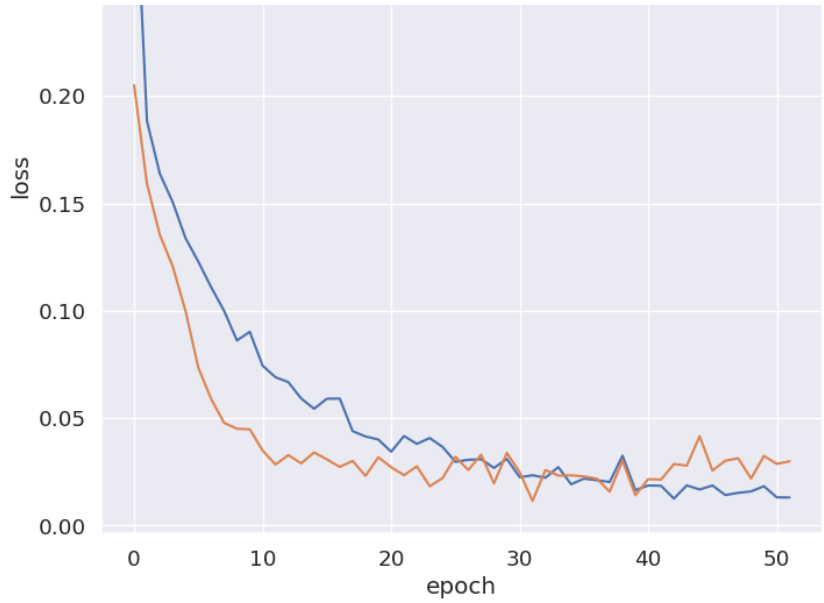
```
y_test.iloc[70]
```

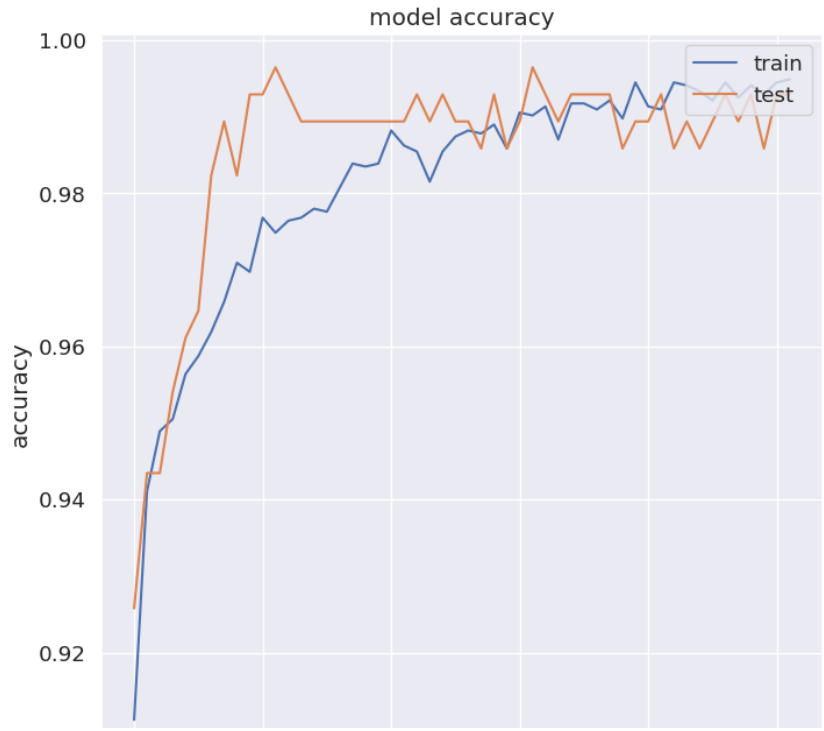
```
0
```

```
model.predict(sc.transform([[42.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.00,132,1.00,1.00,109.0,1.0,0.88,.100,110.00,0.00,0,1]]))
```

```
1/1 [==============================] - 0s 23ms/step
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler wa
  warnings.warn(
array([[1.]], dtype=float32)
```

```
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')
plt.show()

# # summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```






```
model.save('model.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file vi
  saving_api.save_model(
```

```
import joblib
```

```
joblib.dump(sc, 'scaler.pkl')
```

```
['scaler.pkl']
```

```
sc = joblib.load('scaler.pkl')
```

```
%%time
from tensorflow.keras.models import load_model
model = load_model('model.h5')
model.predict(sc.transform([[42.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.00,132,1.00,1.00,109.0,1.0,0.88,.100,110.00,0.00,0,1]]))
```