[1]:
```
#IMPORT REQUIRED LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[3] :
```
#import dataset and load in dataframe
df=pd.read_csv('chronickidneydisease.csv')
df.head()
```

[3]:
```
   id   age    bp     sg   al   su     rbc        pc         pcc          ba  
0   0  48.0  80.0  1.020  1.0  0.0     NaN    normal  notpresent  notpresent  
1   1   7.0  50.0  1.020  4.0  0.0     NaN    normal  notpresent  notpresent  
2   2  62.0  80.0  1.010  2.0  3.0  normal    normal  notpresent  notpresent  
3   3  48.0  70.0  1.005  4.0  0.0  normal  abnormal     present  notpresent  
4   4  51.0  80.0  1.010  2.0  0.0  normal    normal  notpresent  notpresent  

   … pcv    wc   rc  htn   dm  cad appet   pe  ane classification  
0  …  44  7800  5.2  yes  yes   no  good   no   no            ckd  
1  …  38  6000  NaN   no   no   no  good   no   no            ckd  
2  …  31  7500  NaN   no  yes   no  poor   no  yes            ckd  
3  …  32  6700  3.9  yes   no   no  poor  yes  yes            ckd  
4  …  35  7300  4.6   no   no   no  good   no   no            ckd  

[5 rows x 26 columns]
```

[4]:
```
#checking the description and gathering the information about the dataset
df.describe().T
```

[4]:
```
     count        mean         std     min    25%     50%     75%      max
id   400.0  199.500000  115.614301   0.000  99.75  199.50  299.25  399.000
age  391.0   51.483376   17.169714   2.000  42.00   55.00   64.50   90.000
bp   388.0   76.469072   13.683637  50.000  70.00   80.00   80.00  180.000
sg   353.0    1.017408    0.005717   1.005   1.01    1.02    1.02    1.025
al   354.0    1.016949    1.352679   0.000   0.00    0.00    2.00    5.000
```

```
su     351.0    0.450142    1.099191   0.000    0.00    0.00    0.00   5.000
bgr    356.0  148.036517   79.281714  22.000   99.00  121.00  163.00 490.000
bu     381.0   57.425722   50.503006   1.500   27.00   42.00   66.00 391.000
sc     383.0    3.072454    5.741126   0.400    0.90    1.30    2.80  76.000
sod    313.0  137.528754   10.408752   4.500  135.00  138.00  142.00 163.000
pot    312.0    4.627244    3.193904   2.500    3.80    4.40    4.90  47.000
hemo   348.0   12.526437    2.912587   3.100   10.30   12.65   15.00  17.800
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 26 columns):
 #   Column          Non-Null Count  Dtype
-----------------------------------  -------
 0   id              400 non-null    int64
 1   age             391 non-null    float64
 2   bp              388 non-null    float64
 3   sg              353 non-null    float64
 4   al              354 non-null    float64
 5   su              351 non-null    float64
 6   rbc             248 non-null    object
 7   pc              335 non-null    object
 8   pcc             396 non-null    object
 9   ba              396 non-null    object
 10  bgr             356 non-null    float64
 11  bu              381 non-null    float64
 12  sc              383 non-null    float64
 13  sod             313 non-null    float64
 14  pot             312 non-null    float64
 15  hemo            348 non-null    float64
 16  pcv             330 non-null    object
 17  wc              295 non-null    object
 18  rc              270 non-null    object
 19  htn             398 non-null    object
 20  dm              398 non-null    object
 21  cad             398 non-null    object
 22  appet           399 non-null    object
 23  pe              399 non-null    object
 24  ane             399 non-null    object
 25  classification  400 non-null    object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

```
[6]: #counting for the null values
     df.isna().sum()
```

[6]:

```
id                 0
age                9
bp                12
sg                47
al                46
su                49
rbc              152
pc                65
pcc                4
ba                 4
bgr               44
bu                19
sc                17
sod               87
pot               88
hemo              52
pcv               70
wc               105
rc               130
htn                2
dm                 2
cad                2
appet              1
pe                 1
ane                1
classification     0
dtype: int64
```

[11]:

```
#replacing the null values with median and mode
oc=[]#object data type columns
ic=[]#int type columns

for i in df.columns:
    if(df[i].dtype=='object'):
        oc.append(i)
    else:
        ic.append(i)
print("ic\t",ic,"\noc\t",oc)
```

```
ic       ['id', 'age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot',
'hemo']
oc       ['rbc', 'pc', 'pcc', 'ba', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
'appet', 'pe', 'ane', 'classification']
```

[40]:

```
 #replacing the null with median
for i in ic:
    if(df[i].isna().any()==True):
        df[i]=df[i].fillna(df[i].median())
    #checking
    print("Attribute "+i+"\t",df[i].isna().sum())
```

```
Attribute: id    0
Attribute: age   0
Attribute: bp    0
Attribute: sg    0
Attribute: al    0
Attribute: su    0
Attribute: bgr   0
Attribute: bu    0
Attribute: sc    0
Attribute: sod   0
Attribute: pot   0
Attribute: hemo  0
```

[46]:
```
#replacing the null with mode
for i in oc:
    if(df[i].isna().any()==True):
        df[i]=df[i].fillna(df[i].mode()[0])
    #checking
    print("Attribute: "+i+"\t\t\t",df[i].isna().sum())
```

```
Attribute: rbc                  0
Attribute: pc                   0
Attribute: pcc                  0
Attribute: ba                   0
Attribute: pcv                  0
Attribute: wc                   0
Attribute: rc                   0
Attribute: htn                  0
Attribute: dm                   0
Attribute: cad                  0
Attribute: appet                        0
Attribute: pe                   0
Attribute: ane                  0
Attribute: classification                       0
```

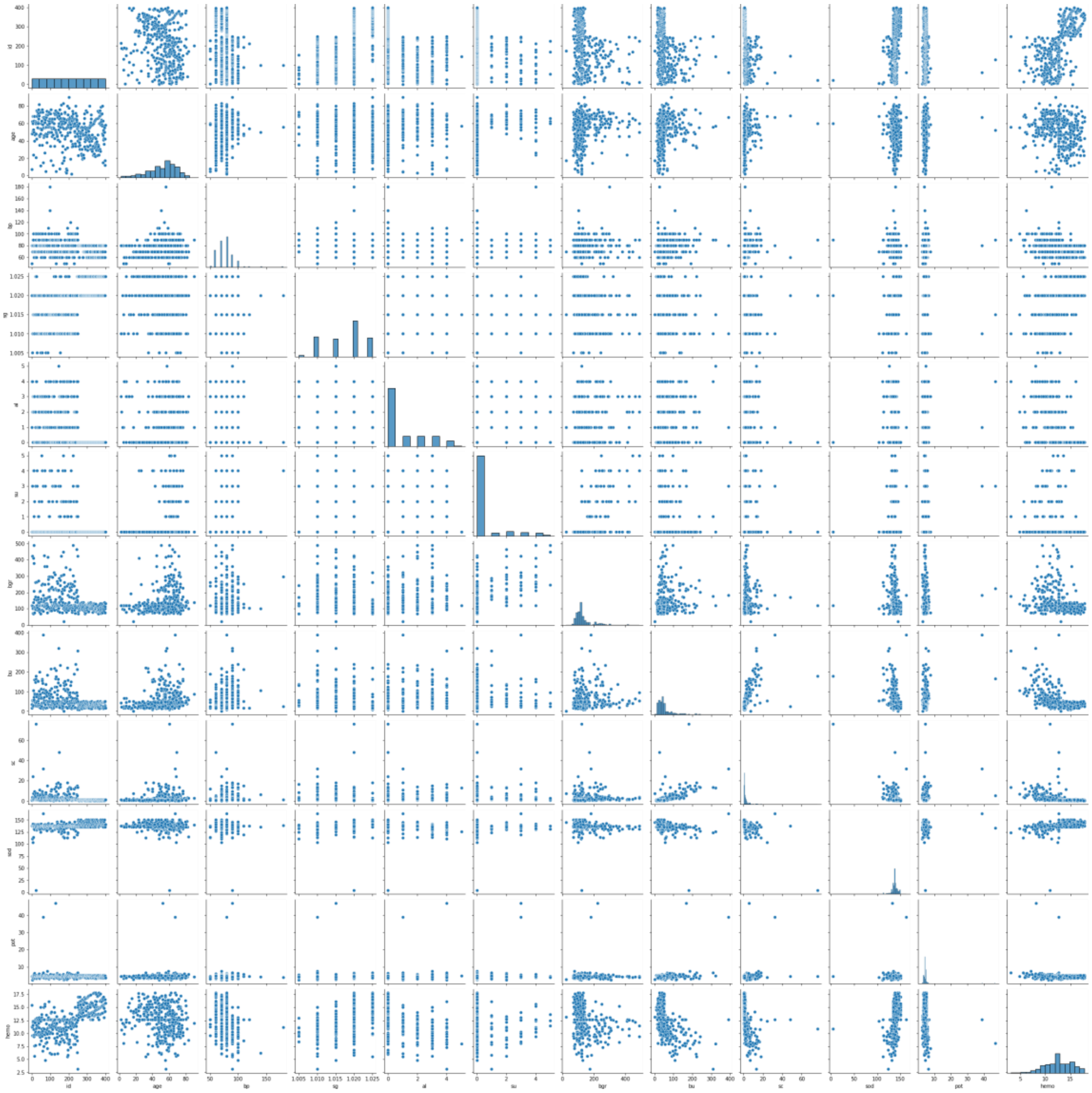
[47]:
```
df.isna().sum()
```

[47]:
```
id                0
age               0
bp                0
sg                0
al                0
su                0
```

```
rbc               0
pc                0
pcc               0
ba                0
bgr               0
bu                0
sc                0
sod               0
pot               0
hemo              0
pcv               0
wc                0
rc                0
htn               0
dm                0
cad               0
appet             0
pe                0
ane               0
classification    0
dtype: int64
```

[50]:
```
#visualizing the datasets
sns.pairplot(df)
```

[50]:
```
<seaborn.axisgrid.PairGrid at 0x7fbb94b144c0>
```



[1]:
```
%matplotlib inline
```

[2]:
```
#IMPORT REQUIRED LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
warnings.filterwarnings('ignore')
```

[3]:
```
#import dataset and load in dataframe
df=pd.read_csv('chronickidneydisease.csv')
df.head()
```

[3]:
```
   id   age    bp     sg   al   su     rbc        pc         pcc          ba
0   0  48.0  80.0  1.020  1.0  0.0     NaN    normal  notpresent  notpresent
1   1   7.0  50.0  1.020  4.0  0.0     NaN    normal  notpresent  notpresent
2   2  62.0  80.0  1.010  2.0  3.0  normal    normal  notpresent  notpresent
3   3  48.0  70.0  1.005  4.0  0.0  normal  abnormal     present  notpresent
4   4  51.0  80.0  1.010  2.0  0.0  normal    normal  notpresent  notpresent

  …  pcv    wc   rc  htn   dm  cad appet   pe  ane classification
0 …   44  7800  5.2  yes  yes   no  good   no   no            ckd
1 …   38  6000  NaN   no   no   no  good   no   no            ckd
2 …   31  7500  NaN   no  yes   no  poor   no  yes            ckd
3 …   32  6700  3.9  yes   no   no  poor  yes  yes            ckd
4 …   35  7300  4.6   no   no   no  good   no   no            ckd

[5 rows x 26 columns]
```

[4]:
```
#dataset adjustment
df['classification']=df['classification'].replace(['ckd\t'],['notckd'])
```

[5]:
```
df['classification'].value_counts()
```

```
[5]: ckd        248
     notckd     152
     Name: classification, dtype: int64
```

```
[6]: #checking the description and gathering the information about the dataset
     df.describe().T
```

```
[6]:       count        mean         std     min     25%     50%     75%      max
     id    400.0  199.500000  115.614301   0.000   99.75  199.50  299.25  399.000
     age   391.0   51.483376   17.169714   2.000   42.00   55.00   64.50   90.000
     bp    388.0   76.469072   13.683637  50.000   70.00   80.00   80.00  180.000
     sg    353.0    1.017408    0.005717   1.005    1.01    1.02    1.02    1.025
     al    354.0    1.016949    1.352679   0.000    0.00    0.00    2.00    5.000
     su    351.0    0.450142    1.099191   0.000    0.00    0.00    0.00    5.000
     bgr   356.0  148.036517   79.281714  22.000   99.00  121.00  163.00  490.000
     bu    381.0   57.425722   50.503006   1.500   27.00   42.00   66.00  391.000
     sc    383.0    3.072454    5.741126   0.400    0.90    1.30    2.80   76.000
     sod   313.0  137.528754   10.408752   4.500  135.00  138.00  142.00  163.000
     pot   312.0    4.627244    3.193904   2.500    3.80    4.40    4.90   47.000
     hemo  348.0   12.526437    2.912587   3.100   10.30   12.65   15.00   17.800
```

```
[7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
------------          --------------  ---------
 0   id               400 non-null    int64
 1   age              391 non-null    float64
 2   bp               388 non-null    float64
 3   sg               353 non-null    float64
 4   al               354 non-null    float64
 5   su               351 non-null    float64
 6   rbc              248 non-null    object
 7   pc               335 non-null    object
 8   pcc              396 non-null    object
 9   ba               396 non-null    object
 10  bgr              356 non-null    float64
 11  bu               381 non-null    float64
 12  sc               383 non-null    float64
 13  sod              313 non-null    float64
 14  pot              312 non-null    float64
 15  hemo             348 non-null    float64
 16  pcv              330 non-null    object
 17  wc               295 non-null    object
 18  rc               270 non-null    object
 19  htn             398 non-null    object
 20  dm              398 non-null    object
 21  cad             398 non-null    object
 22  appet           399 non-null    object
 23  pe              399 non-null    object
 24  ane             399 non-null    object
 25  classification  400 non-null    object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

[8]:
```
#counting for the null values
df.isna().sum()
```

[8]:
```
 id                  0
 age                 9
 bp                 12
 sg                 47
 al                 46
 su                 49
 rbc               152
 pc                 65
 pcc                 4
 ba                  4
 bgr                44
 bu                 19
 sc                 17
 sod                87
 pot                88
 hemo               52
 pcv                70
 wc                105
 rc                130
 htn                 2
 dm                  2
 cad                 2
 appet               1
 pe                  1
 ane                 1
classification      0
dtype: int64
```

[9]:
```
#replacing the null values with median and mode
oc=[]#object data type columns
ic=[]#int type columns

for i in df.columns:
    if(df[i].dtype=='object'):
        oc.append(i)
    else:
        ic.append(i)
print(″ic\t″,ic,″\noc\t″,oc)
```

```
ic       [’id’, ’age’, ’bp’, ’sg’, ’al’, ’su’, ’bgr’, ’bu’, ’sc’, ’sod’, ’pot’,
’hemo’]
oc       [’rbc’, ’pc’, ’pcc’, ’ba’, ’pcv’, ’wc’, ’rc’, ’htn’, ’dm’, ’cad’,
’appet’, ’pe’, ’ane’, ’classification’]
```

[10]:

```
#replacing the null with median
for i in ic:
    if(df[i].isna().any()==True):
        df[i]=df[i].fillna(df[i].median())
    #checking
    print(″Attribute ″+i+″\t″,df[i].isna().sum())
```

```
Attribute id     0
Attribute age    0
Attribute bp     0
Attribute sg     0
Attribute al     0
Attribute su     0
Attribute bgr    0
Attribute bu     0
Attribute sc     0
Attribute sod    0
Attribute pot    0
Attribute hemo   0
```

[11]:

```
#replacing the null with mode
for i in oc:
    if(df[i].isna().any()==True):
        df[i]=df[i].fillna(df[i].mode()[0])
    #checking
    print(″Attribute:  ″+i+″\t\t\t″,df[i].isna().sum())
```

```
Attribute: rbc                  0
Attribute: pc                   0
Attribute: pcc                  0
Attribute: ba                   0
Attribute: pcv                  0
Attribute: wc                   0
Attribute: rc                   0
Attribute: htn                  0
Attribute: dm                   0
Attribute: cad                  0
Attribute: appet                        0
```

```
Attribute: pe                    0
Attribute: ane                   0
Attribute: classification                      0
```

[12]:
```
df.isna().sum().sum()
```

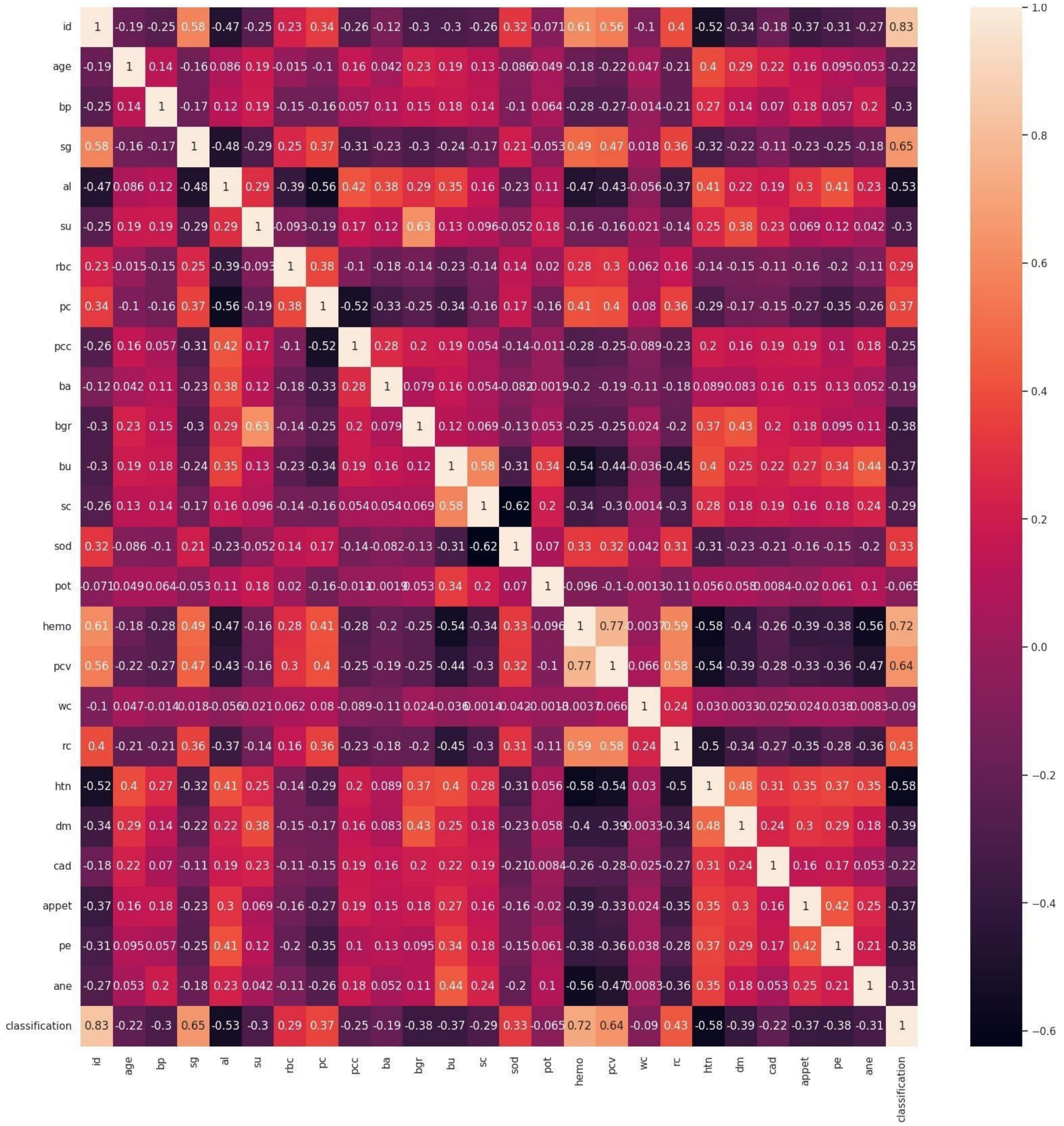
[12]:
```
0
```

[13]:
```
#encoding labels
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()#label encoder object
for i in oc:
    df[i]=le.fit_transform(df[i])#label encoding all the object dtypes

df.head(3)
```

[13]:
```
    id  age    bp    sg   al   su  rbc  pc  pcc  ba  …  pcv  wc  rc  htn  \
0    0  48.0  80.0  1.02  1.0  0.0   1   1    0   0  …   32  72  34   1
1    1   7.0  50.0  1.02  4.0  0.0   1   1    0   0  …   26  56  34   0
2    2  62.0  80.0  1.01  2.0  3.0   1   1    0   0  …   19  70  34   0

   dm  cad  appet  pe  ane  classification
0  4    1      0  0    0                0
1  3    1      0  0    0                0
2  4    1      1  0    1                0

[3 rows x 26 columns]
```

[14]:
```
df.corr()
fig=plt.figure(figsize=(20,20))

sns.heatmap(data=df.corr(),annot=True)
```

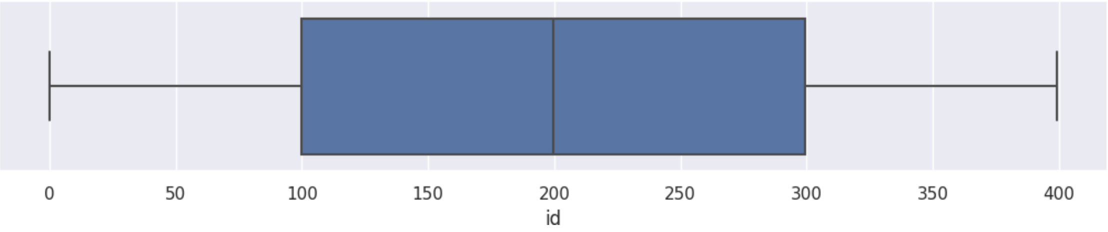
[14]:
```
<AxesSubplot: >
```



	id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
id	1	-0.19	-0.25	0.58	-0.47	-0.25	0.23	0.34	-0.26	-0.12	-0.3	-0.3	-0.26	0.32	-0.071	0.61	0.56	-0.1	0.4	-0.52	-0.34	-0.18	-0.37	-0.31	-0.27	0.83
age	-0.19	1	0.14	-0.16	0.086	0.19	-0.015	-0.1	0.16	0.042	0.23	0.19	0.13	-0.086	0.049	-0.18	-0.22	0.047	-0.21	0.4	0.29	0.22	0.16	0.095	0.053	-0.22
bp	-0.25	0.14	1	-0.17	0.12	0.19	-0.15	-0.16	0.057	0.11	0.15	0.18	0.14	-0.1	0.064	-0.28	-0.27	-0.014	-0.21	0.27	0.14	0.07	0.18	0.057	0.2	-0.3
sg	0.58	-0.16	-0.17	1	-0.48	-0.29	0.25	0.37	-0.31	-0.23	-0.3	-0.24	-0.17	0.21	-0.053	0.49	0.47	0.018	0.36	-0.32	-0.22	-0.11	-0.23	-0.25	-0.18	0.65
al	-0.47	0.086	0.12	-0.48	1	0.29	-0.39	-0.56	0.42	0.38	0.29	0.35	0.16	-0.23	0.11	-0.47	-0.43	-0.056	-0.37	0.41	0.22	0.19	0.3	0.41	0.23	-0.53
su	-0.25	0.19	0.19	-0.29	0.29	1	-0.093	-0.19	0.17	0.12	0.63	0.13	0.096	-0.052	0.18	-0.16	-0.16	0.021	-0.14	0.25	0.38	0.23	0.069	0.12	0.042	-0.3
rbc	0.23	-0.015	-0.15	0.25	-0.39	-0.093	1	0.38	-0.1	-0.18	-0.14	-0.23	-0.14	0.14	0.02	0.28	0.3	0.062	0.16	-0.14	-0.15	-0.11	-0.16	-0.2	-0.11	0.29
pc	0.34	-0.1	-0.16	0.37	-0.56	-0.19	0.38	1	-0.52	-0.33	-0.25	-0.34	-0.16	0.17	-0.16	0.41	0.4	0.08	0.36	-0.29	-0.17	-0.15	-0.27	-0.35	-0.26	0.37
pcc	-0.26	0.16	0.057	-0.31	0.42	0.17	-0.1	-0.52	1	0.28	0.2	0.19	0.054	-0.14	-0.011	-0.28	-0.25	-0.089	-0.23	0.2	0.16	0.19	0.19	0.1	0.18	-0.25
ba	-0.12	0.042	0.11	-0.23	0.38	0.12	-0.18	-0.33	0.28	1	0.079	0.16	0.054	-0.082	0.0019	-0.2	-0.19	-0.11	-0.18	0.089	0.083	0.16	0.15	0.13	0.052	-0.19
bgr	-0.3	0.23	0.15	-0.3	0.29	0.63	-0.14	-0.25	0.2	0.079	1	0.12	0.069	-0.13	0.053	-0.25	-0.25	0.024	-0.2	0.37	0.43	0.2	0.18	0.095	0.11	-0.38
bu	-0.3	0.19	0.18	-0.24	0.35	0.13	-0.23	-0.34	0.19	0.16	0.12	1	0.58	-0.31	0.34	-0.54	-0.44	-0.036	-0.45	0.4	0.25	0.22	0.27	0.34	0.44	-0.37
sc	-0.26	0.13	0.14	-0.17	0.16	0.096	-0.14	-0.16	0.054	0.054	0.069	0.58	1	-0.62	0.2	-0.34	-0.3	0.0014	-0.3	0.28	0.18	0.19	0.16	0.18	0.24	-0.29
sod	0.32	-0.086	-0.1	0.21	-0.23	-0.052	0.14	0.17	-0.14	-0.082	-0.13	-0.31	-0.62	1	0.07	0.33	0.32	0.042	0.31	-0.31	-0.23	-0.21	-0.16	-0.15	-0.2	0.33
pot	-0.071	0.049	0.064	-0.053	0.11	0.18	0.02	-0.16	-0.011	0.0019	0.053	0.34	0.2	0.07	1	-0.096	-0.1	-0.0013	-0.11	0.056	0.058	0.0084	-0.02	0.061	0.1	-0.065
hemo	0.61	-0.18	-0.28	0.49	-0.47	-0.16	0.28	0.41	-0.28	-0.2	-0.25	-0.54	-0.34	0.33	-0.096	1	0.77	-0.0037	0.59	-0.58	-0.4	-0.26	-0.39	-0.38	-0.56	0.72
pcv	0.56	-0.22	-0.27	0.47	-0.43	-0.16	0.3	0.4	-0.25	-0.19	-0.25	-0.44	-0.3	0.32	-0.1	0.77	1	0.066	0.58	-0.54	-0.39	-0.28	-0.33	-0.36	-0.47	0.64
wc	-0.1	0.047	-0.014	0.018	-0.056	0.021	0.062	0.08	-0.089	-0.11	0.024	-0.036	0.0014	0.042	-0.0013	-0.0037	0.066	1	0.24	0.03	0.0033	-0.025	0.024	0.038	0.0083	-0.09
rc	0.4	-0.21	-0.21	0.36	-0.37	-0.14	0.16	0.36	-0.23	-0.18	-0.2	-0.45	-0.3	0.31	-0.11	0.59	0.58	0.24	1	-0.5	-0.34	-0.27	-0.35	-0.28	-0.36	0.43
htn	-0.52	0.4	0.27	-0.32	0.41	0.25	-0.14	-0.29	0.2	0.089	0.37	0.4	0.28	-0.31	0.056	-0.58	-0.54	0.03	-0.5	1	0.48	0.31	0.35	0.37	0.35	-0.58
dm	-0.34	0.29	0.14	-0.22	0.22	0.38	-0.15	-0.17	0.16	0.083	0.43	0.25	0.18	-0.23	0.058	-0.4	-0.39	0.0033	-0.34	0.48	1	0.24	0.3	0.29	0.18	-0.39
cad	-0.18	0.22	0.07	-0.11	0.19	0.23	-0.11	-0.15	0.19	0.16	0.2	0.22	0.19	-0.21	0.0084	-0.26	-0.28	-0.025	-0.27	0.31	0.24	1	0.16	0.17	0.053	-0.22
appet	-0.37	0.16	0.18	-0.23	0.3	0.069	-0.16	-0.27	0.19	0.15	0.18	0.27	0.16	-0.16	-0.02	-0.39	-0.33	0.024	-0.35	0.35	0.3	0.16	1	0.42	0.25	-0.37
pe	-0.31	0.095	0.057	-0.25	0.41	0.12	-0.2	-0.35	0.1	0.13	0.095	0.34	0.18	-0.15	0.061	-0.38	-0.36	0.038	-0.28	0.37	0.29	0.17	0.42	1	0.21	-0.38
ane	-0.27	0.053	0.2	-0.18	0.23	0.042	-0.11	-0.26	0.18	0.052	0.11	0.44	0.24	-0.2	0.1	-0.56	-0.47	0.0083	-0.36	0.35	0.18	0.053	0.25	0.21	1	-0.31
classification	0.83	-0.22	-0.3	0.65	-0.53	-0.3	0.29	0.37	-0.25	-0.19	-0.38	-0.37	-0.29	0.33	-0.065	0.72	0.64	-0.09	0.43	-0.58	-0.39	-0.22	-0.37	-0.38	-0.31	1

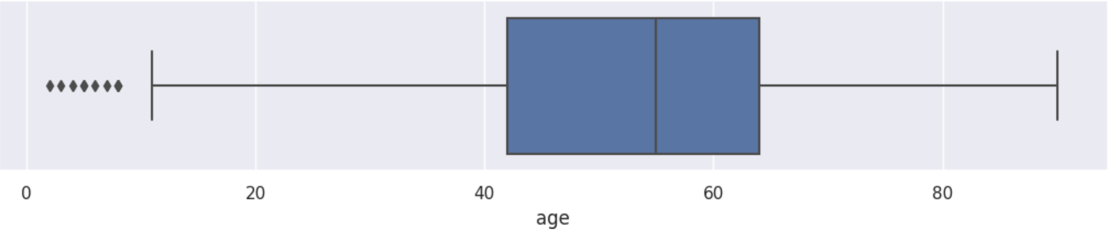
[16]:
```
#seeing outliers
sns.boxplot(df['id'])
```

[16] : <AxesSubplot: xlabel='id'>



[17] :
```
sns.boxplot(df['age'])
```

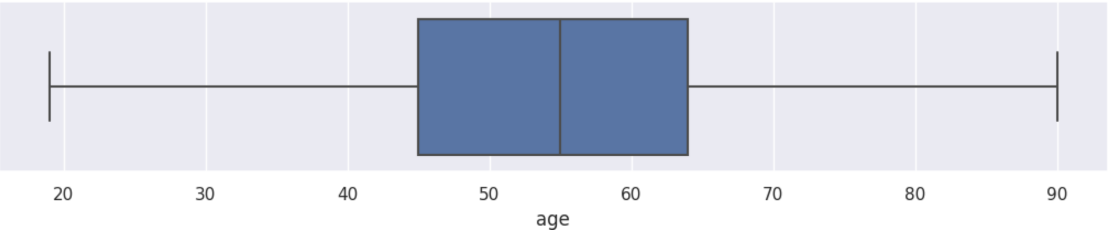
[17] :
```
<AxesSubplot: xlabel='age'>
```



[18] :
```
#replacing the outliers
median=df[’age’].median()
print(median)
df[’age’]=df[’age’].mask(df[’age’]<19,median)
sns.boxplot(df[’age’])
```

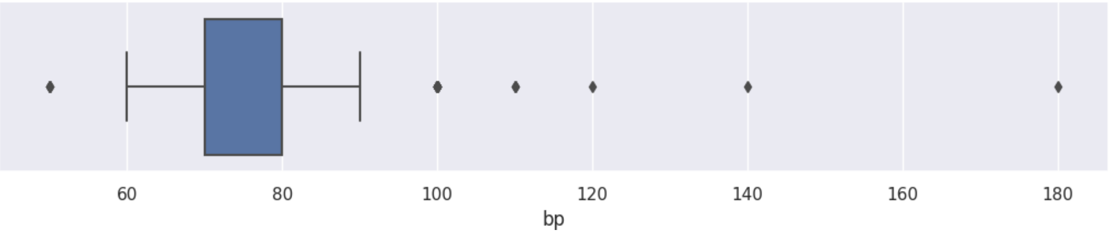
```
55.0
```

[18] :
```
<AxesSubplot: xlabel='age'>
```



[19] :
```
sns.boxplot(df['bp'])
```

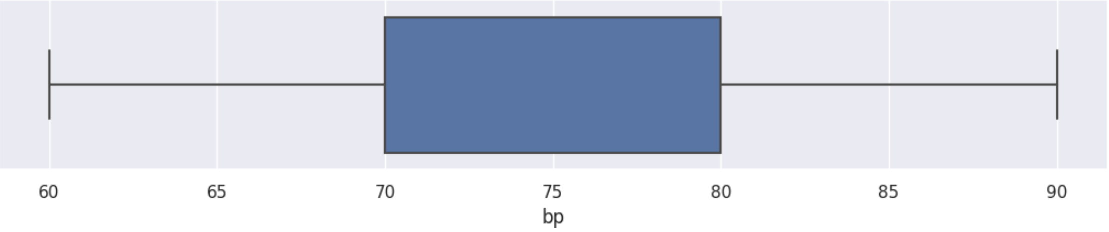
[19] :
```
<AxesSubplot: xlabel='bp'>
```



[20] :
```
#replacing outliers
median=df['bp'].median()
print(median)
df[’bp’]=df[’bp’].mask(df[’bp’]<60,median)
df[’bp’]=df[’bp’].mask(df[’bp’]>90,median)
sns.boxplot(df[’bp’])
```

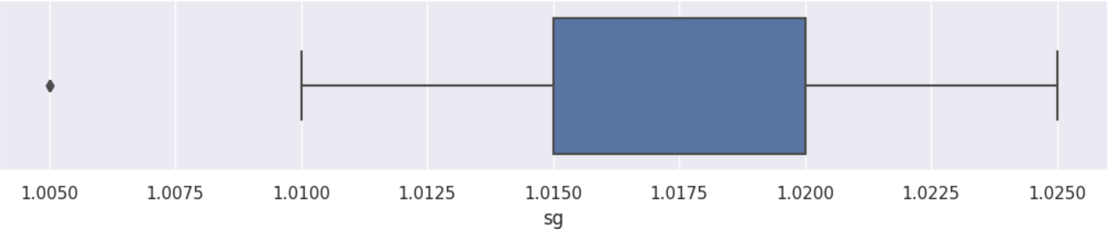
```
80.0
```

[20] : <AxesSubplot: xlabel='bp'>



[21] :
```
sns.boxplot(df['sg'])
```

[21] : <AxesSubplot: xlabel='sg'>

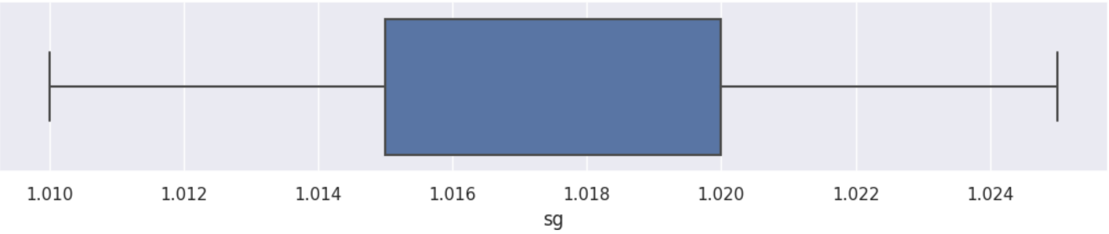


[22] :
```
#replacing outliers
median=df['sg'].median()
print(median)
```

```
df[’sg’]=df[’sg’].mask(df[’sg’]<1.0100,median)
sns.boxplot(df[’sg’])
```

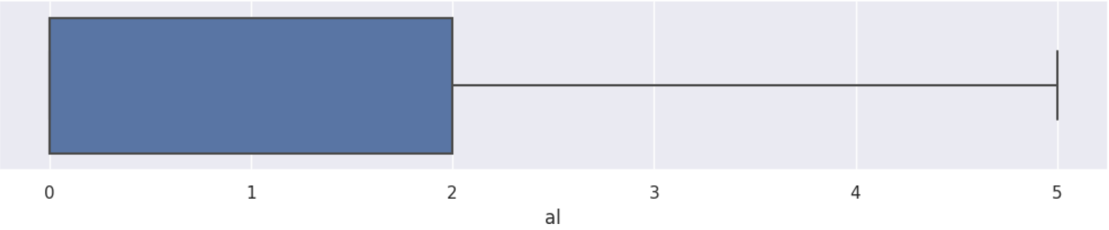
```
1.02
```

[22] : <AxesSubplot: xlabel='sg'>



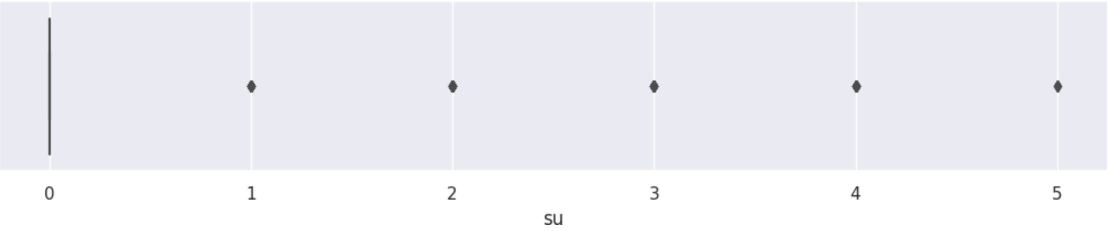
[23] :
```
sns.boxplot(df['al'])
```

[23] : <AxesSubplot: xlabel='al'>



[24] :
```
sns.boxplot(df['su'])
```

[24] : <AxesSubplot: xlabel='su'>



[25] :
```
#replacing outliers
 median=df['su'].median()
 print(median)
```

```
df['su']=df['su'].mask(df['su']>0,median)
sns.boxplot(df['su'])
```

```
0.0
```

[25] : <AxesSubplot: xlabel='su'>

−0.04
−0.02
0.00
0.02
0.04
su

[26] :
```
idv=df.iloc[:,:-1]#independent variables
dv=df.iloc[:,-1]#dependent variables
idv
```

[26]:
```
      id   age    bp     sg   al   su   rbc  pc  pcc   ba  ...  hemo  pcv  wc  \
0      0  48.0  80.0  1.020  1.0  0.0     1   1    0    0  ...  15.4   32  72
1      1  55.0  80.0  1.020  4.0  0.0     1   1    0    0  ...  11.3   26  56
2      2  62.0  80.0  1.010  2.0  0.0     1   1    0    0  ...   9.6   19  70
3      3  48.0  70.0  1.020  4.0  0.0     1   0    1    0  ...  11.2   20  62
4      4  51.0  80.0  1.010  2.0  0.0     1   1    0    0  ...  11.6   23  68
..   ...   ...   ...    ... ... ...  ...  ..  ...  ..   ...  ...  ...  ..
395  395  55.0  80.0  1.020  0.0  0.0     1   1    0    0  ...  15.7   35  62
396  396  42.0  70.0  1.025  0.0  0.0     1   1    0    0  ...  16.5   42  72
397  397  55.0  80.0  1.020  0.0  0.0     1   1    0    0  ...  15.8   37  61
398  398  55.0  60.0  1.025  0.0  0.0     1   1    0    0  ...  14.2   39  67
399  399  58.0  80.0  1.025  0.0  0.0     1   1    0    0  ...  15.8   41  63

     rc  htn  dm  cad  appet  pe  ane
0    34    1   4    1      0   0    0
1    34    0   3    1      0   0    0
2    34    0   4    1      1   0    1
3    19    1   3    1      1   1    1
4    27    0   3    1      0   0    0
..   ..  ...  ..  ...    ...  ..  ...
395  30    0   3    1      0   0    0
396  44    0   3    1      0   0    0
397  36    0   3    1      0   0    0
398  41    0   3    1      0   0    0
399  43    0   3    1      0   0    0
```

```
[400 rows x 25 columns]
```

[27] :

```
#splitting datasets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(idv,dv,test_size=0.
 ↪2,shuffle=True)
```

[28] :

```
x_train.shape
```

[28]:

```
(320, 25)
```

[29] :

```
#creating models
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

[30] :

```
model.fit(x_train,y_train)
```

[30]:

```
LogisticRegression()
```

[31] :

```
#acccuracy
pred=model.predict(x_test)
pred
```

[31]:

```
array([0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
     0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0])
```

[32] :

```
#for checking.......
from sklearn.svm import SVC
svmmodel=SVC()
```

[33] :

```
svmmodel.fit(x_train,y_train)
```

[33]:

```
SVC()
```

[34] :

```
#acccuracy
svc_pred=model.predict(x_test)
svc_pred
```

[34]:

```
array([0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
     0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0])
```

[35] :

```
from sklearn.metrics import accuracy_score,confusion_matrix
 accuracy_score(y_test,pred)
```

```
[35]:  0.9875
```

```
[36] : confusion_matrix(y_test,pred)
```

```
               [ 0,  33]])
```

```
[36]: array([[
```

```
[37] : y_train.value_counts()
```

```
[37]: 0        2
```

```
  1     119
  Name: classification, dtype: int64
```

```
[38] : #svm accuracy & confusion matrix
        accuracy_score(y_test,svc_pred)
```

```
[38]:  0.9875
```

```
[39] : confusion_matrix(y_test,svc_pred)
```

```
[39]:  array([[46,   1],
               [ 0,  33]])
```

```
[40] : #creating model
        import pickle
```

```
[41] : pickle.dump(model,open('ckd model.pkl','wb'))
        print("model saved successfully")
```

```
model saved successfully
```

```
[ ]:
```