In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
dataset=pd.read_csv("kidney_disease.csv")
dataset
```

Out[2]:

	id	age	F	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classificatio
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44.0	7800.0	5.2	yes	yes	no	good	no	no	ck
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38.0	6000.0	NaN	no	no	no	good	no	no	ck
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31.0	7500.0	NaN	no	yes	no	poor	no	yes	ck
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32.0	6700.0	3.9	yes	no	no	poor	yes	yes	ck
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300.0	4.6	no	no	no	good	no	no	ck
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47.0	6700.0	4.9	no	no	no	good	no	no	notck
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54.0	7800.0	6.2	no	no	no	good	no	no	notck
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49.0	6600.0	5.4	no	no	no	good	no	no	notck
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51.0	7200.0	5.9	no	no	no	good	no	no	notck
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53.0	6800.0	6.1	no	no	no	good	no	no	notck

400 rows × 26 columns

In [3]:
```
dataset.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   F               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             329 non-null    float64
 17  wc              294 non-null    float64
 18  rc              269 non-null    float64
 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(14), int64(1), object(11)
memory usage: 81.4+ KB
```

In [4]:
```
dataset.isnull().sum().T
#print(a,end='')
```

Out[4]:
```
id                  0
age                 9
F                  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                71
wc                106
rc                131
htn                 2
dm                  2
cad                 2
appet               1
pe                  1
ane                 1
classification      0
dtype: int64
```

Deleting Certain Rows with Missing Values

In [5]:
```
dataset.dropna(subset=['age','pcc', 'ba','htn','dm','cad','appet','pe','ane'], inplace=True)
dataset
```

Out[5]:

	id	age	F	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classificatio
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44.0	7800.0	5.2	yes	yes	no	good	no	no	ck
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38.0	6000.0	NaN	no	no	no	good	no	no	ck
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31.0	7500.0	NaN	no	yes	no	poor	no	yes	ck
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32.0	6700.0	3.9	yes	no	no	poor	yes	yes	ck
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300.0	4.6	no	no	no	good	no	no	ck
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47.0	6700.0	4.9	no	no	no	good	no	no	notck
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54.0	7800.0	6.2	no	no	no	good	no	no	notck
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49.0	6600.0	5.4	no	no	no	good	no	no	notck
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51.0	7200.0	5.9	no	no	no	good	no	no	notck
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53.0	6800.0	6.1	no	no	no	good	no	no	notck

384 rows × 26 columns

Filling Missing Numerical Values with Median Values

In [6]:
```
dataset.fillna(dataset.median(), inplace=True)
dataset
```

```
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_3973/3113556935.py:1: FutureWarning: The defaul
t value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False.
In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value
of numeric_only to silence this warning.
  dataset.fillna(dataset.median(), inplace=True)
```

Out[6]:

	id	age	F	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44.0	7800.0	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38.0	6000.0	4.7	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31.0	7500.0	4.7	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32.0	6700.0	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300.0	4.6	no	no	no	good	no	no	ckd
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47.0	6700.0	4.9	no	no	no	good	no	no	notckd
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54.0	7800.0	6.2	no	no	no	good	no	no	notckd
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49.0	6600.0	5.4	no	no	no	good	no	no	notckd
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51.0	7200.0	5.9	no	no	no	good	no	no	notckd
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53.0	6800.0	6.1	no	no	no	good	no	no	notckd

384 rows × 26 columns

In [7]:
```
dataset.isna().sum()
```

Out[7]:
```
id                  0
age                 0
F                   0
sg                  0
al                  0
su                  0
rbc               146
pc                 60
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
```

Finding Mode to Replace Missing Categorical Values

In [8]:
```
categorical_columns = ['rbc','pc']
modes = dataset[categorical_columns].mode()
modes
```

Out[8]:

	rbc	pc
0	normal	normal

Replacing the Missing Categorical Values With Calculated Mode

In [9]:
```
dataset[dataset.select_dtypes(include=['object']).columns] = dataset.select_dtypes(include=['object']).
dataset
```

Out[9]:

	id	age	F	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	normal	normal	notpresent	notpresent	...	44.0	7800.0	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	normal	normal	notpresent	notpresent	...	38.0	6000.0	4.7	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31.0	7500.0	4.7	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32.0	6700.0	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300.0	4.6	no	no	no	good	no	no	ckd
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47.0	6700.0	4.9	no	no	no	good	no	no	notckd
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54.0	7800.0	6.2	no	no	no	good	no	no	notckd
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49.0	6600.0	5.4	no	no	no	good	no	no	notckd
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51.0	7200.0	5.9	no	no	no	good	no	no	notckd
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53.0	6800.0	6.1	no	no	no	good	no	no	notckd

384 rows × 26 columns

In [10]:
```
dataset.isnull().sum()
```

Out[10]:
```
id                0
age               0
F                 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
```

Defining Quan & Qual for Outlier Removal

In [11]:
```
def quanQual(dataset):
    quan=[]
    qual=[]
    for columnName in dataset.columns:
        #print(columnName)
        if(dataset[columnName].dtype=='O'):
            #print("qual")
            qual.append(columnName)
        else:
            #print("quan")
            quan.append(columnName)
    return quan,qual
```

In [12]:
```
quan,qual=quanQual(dataset)
quan
```
Out[12]:
```
['id',
 'age',
 'F',
 'sg',
 'al',
 'su',
 'bgr',
 'bu',
 'sc',
 'sod',
 'pot',
 'hemo',
 'pcv',
 'wc',
 'rc']
```

In [13]:
```
dataset[qual]
```
Out[13]:

	rbc	pc	pcc	ba	htn	dm	cad	appet	pe	ane	classification
0	normal	normal	notpresent	notpresent	yes	yes	no	good	no	no	ckd
1	normal	normal	notpresent	notpresent	no	no	no	good	no	no	ckd
2	normal	normal	notpresent	notpresent	no	yes	no	poor	no	yes	ckd
3	normal	abnormal	present	notpresent	yes	no	no	poor	yes	yes	ckd
4	normal	normal	notpresent	notpresent	no	no	no	good	no	no	ckd
...	...	...	...	...	...	...	...	...	...	...	...
395	normal	normal	notpresent	notpresent	no	no	no	good	no	no	notckd
396	normal	normal	notpresent	notpresent	no	no	no	good	no	no	notckd
397	normal	normal	notpresent	notpresent	no	no	no	good	no	no	notckd
398	normal	normal	notpresent	notpresent	no	no	no	good	no	no	notckd
399	normal	normal	notpresent	notpresent	no	no	no	good	no	no	notckd

384 rows × 11 columns

Outlier Removal

In [14]:
```
descriptive=pd.DataFrame(index=["Mean","Median","Mode","Q1:25%","Q2:50%",
                               "Q3:75%","99%","Q4:100%","IQR","1.5rule","Lesser","Greater","Min","Max"],
for columnName in quan:
    descriptive[columnName]["Mean"]=dataset[columnName].mean()
    descriptive[columnName]["Median"]=dataset[columnName].median()
    descriptive[columnName]["Mode"]=dataset[columnName].mode()[0]
    descriptive[columnName]["Q1:25%"]=dataset.describe()[columnName]["25%"]
    descriptive[columnName]["Q2:50%"]=dataset.describe()[columnName]["50%"]
    descriptive[columnName]["Q3:75%"]=dataset.describe()[columnName]["75%"]
    descriptive[columnName]["99%"]=np.percentile(dataset[columnName],99)
    descriptive[columnName]["Q4:100%"]=dataset.describe()[columnName]["max"]
    descriptive[columnName]["IQR"]=descriptive[columnName]["Q3:75%"]-descriptive[columnName]["Q1:25%"]
    descriptive[columnName]["1.5rule"]=1.5*descriptive[columnName]["IQR"]
    descriptive[columnName]["Lesser"]=descriptive[columnName]["Q1:25%"]-descriptive[columnName]["1.5rule
    descriptive[columnName]["Greater"]=descriptive[columnName]["Q3:75%"]+descriptive[columnName]["1.5rul
    descriptive[columnName]["Min"]=dataset[columnName].min()
    descriptive[columnName]["Max"]=dataset[columnName].max()
```

In [15]:
```
lesser=[]
greater=[]

for columnName in quan:
    if(descriptive[columnName]["Lesser"]>descriptive[columnName]["Min"]):
        lesser.append(columnName)
    if(descriptive[columnName]["Greater"]<descriptive[columnName]["Q4:100%"]):
        greater.append(columnName)

```

In [16]:
```
lesser
```
Out[16]:
```
['age', 'F', 'sg', 'bgr', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc']
```

In [17]:
```
greater
```
Out[17]:
```
['F', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'wc', 'rc']
```

In [18]:
```
for column in lesser:
    dataset[column][dataset[column]<descriptive[column]["Lesser"]]=descriptive[column]["Lesser"]
for column in greater:
    dataset[column][dataset[column]>descriptive[column]["Greater"]]=descriptive[column]["Greater"]

```

```
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_3973/3400726572.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  dataset[column][dataset[column]<descriptive[column]["Lesser"]]=descriptive[column]["Lesser"]
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_3973/3400726572.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  dataset[column][dataset[column]<descriptive[column]["Lesser"]]=descriptive[column]["Lesser"]
/var/folders/07/ykgp85052b11h5kz22ghn8l40000gn/T/ipykernel_3973/3400726572.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
```

In [19]:
```
descriptive=pd.DataFrame(index=["Mean","Median","Mode","Q1:25%","Q2:50%",
                               "Q3:75%","99%","Q4:100%","IQR","1.5rule","Lesser","Greater","Min","Max"]
for columnName in quan:
    descriptive[columnName]["Mean"]=dataset[columnName].mean()
    descriptive[columnName]["Median"]=dataset[columnName].median()
    descriptive[columnName]["Mode"]=dataset[columnName].mode()[0]
    descriptive[columnName]["Q1:25%"]=dataset.describe()[columnName]["25%"]
    descriptive[columnName]["Q2:50%"]=dataset.describe()[columnName]["50%"]
    descriptive[columnName]["Q3:75%"]=dataset.describe()[columnName]["75%"]
    descriptive[columnName]["99%"]=np.percentile(dataset[columnName],99)
    descriptive[columnName]["Q4:100%"]=dataset.describe()[columnName]["max"]
    descriptive[columnName]["IQR"]=descriptive[columnName]["Q3:75%"]-descriptive[columnName]["Q1:25%"]
    descriptive[columnName]["1.5rule"]=1.5*descriptive[columnName]["IQR"]
    descriptive[columnName]["Lesser"]=descriptive[columnName]["Q1:25%"]-descriptive[columnName]["1.5rul
    descriptive[columnName]["Greater"]=descriptive[columnName]["Q3:75%"]+descriptive[columnName]["1.5ru
    descriptive[columnName]["Min"]=dataset[columnName].min()
    descriptive[columnName]["Max"]=dataset[columnName].max()
```

In [20]:
```
    descriptive
```

Out[20]:

	id	age	F	sg	al	su	bgr	bu	sc	sod	pot	hemo	pcv
Mean	198.979167	51.574219	76.002604	1.017728	0.90625	0.0	134.552083	49.758984	2.061328	138.205729	4.378385	12.544434	39.114583
Median	198.5	55.0	80.0	1.02	0.0	0.0	122.0	42.0	1.3	138.0	4.4	12.6	40.0
Mode	0	60.0	80.0	1.02	0.0	0.0	224.25	110.125	5.4	138.0	4.4	12.6	40.0
Q1:25%	97.75	42.0	70.0	1.015	0.0	0.0	101.75	27.0	0.9	135.0	4.0	10.9	34.0
Q2:50%	198.5	55.0	80.0	1.02	0.0	0.0	122.0	42.0	1.3	138.0	4.4	12.6	40.0
Q3:75%	301.25	65.0	80.0	1.02	2.0	0.0	150.75	60.25	2.7	141.0	4.8	14.525	44.0
99%	395.17	80.17	95.0	1.025	4.0	0.0	224.25	110.125	5.4	150.0	6.0	17.617	53.17
Q4:100%	399.0	90.0	95.0	1.025	5.0	0.0	224.25	110.125	5.4	150.0	6.0	17.8	54.0
IQR	203.5	23.0	10.0	0.005	2.0	0.0	49.0	33.25	1.8	6.0	0.8	3.625	10.0
1.5rule	305.25	34.5	15.0	0.0075	3.0	0.0	73.5	49.875	2.7	9.0	1.2	5.4375	15.0
Lesser	-207.5	7.5	55.0	1.0075	-3.0	0.0	28.25	-22.875	-1.8	126.0	2.8	5.4625	19.0
Greater	606.5	99.5	95.0	1.0275	5.0	0.0	224.25	110.125	5.4	150.0	6.0	19.9625	59.0
Min	0	7.5	55.0	1.0075	0.0	0.0	28.25	1.5	0.4	126.0	2.8	5.4625	19.0
Max	399	90.0	95.0	1.025	5.0	0.0	224.25	110.125	5.4	150.0	6.0	17.8	54.0

In [21]:
```
lesser=[]
greater=[]

for columnName in quan:
    if(descriptive[columnName]["Lesser"]>descriptive[columnName]["Min"]):
        lesser.append(columnName)
    if(descriptive[columnName]["Greater"]<descriptive[columnName]["Q4:100%"]):
        greater.append(columnName)
```

In [22]:
```
lesser
```

Out[22]:
```
[]
```

In [23]:
```
greater
```
Out[23]:
```
[]
```

In [24]:
```
dataset
```
Out[24]:

	id	age	F	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classifica
0	0	48.0	80.0	1.0200	1.0	0.0	normal	normal	notpresent	notpresent	...	44.0	7800.0	5.2000	yes	yes	no	good	no	no	
1	1	7.5	55.0	1.0200	4.0	0.0	normal	normal	notpresent	notpresent	...	38.0	6000.0	4.7000	no	no	no	good	no	no	
2	2	62.0	80.0	1.0100	2.0	0.0	normal	normal	notpresent	notpresent	...	31.0	7500.0	4.7000	no	yes	no	poor	no	yes	
3	3	48.0	70.0	1.0075	4.0	0.0	normal	abnormal	present	notpresent	...	32.0	6700.0	3.9000	yes	no	no	poor	yes	yes	
4	4	51.0	80.0	1.0100	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300.0	4.6000	no	no	no	good	no	no	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
395	395	55.0	80.0	1.0200	0.0	0.0	normal	normal	notpresent	notpresent	...	47.0	6700.0	4.9000	no	no	no	good	no	no	nc
396	396	42.0	70.0	1.0250	0.0	0.0	normal	normal	notpresent	notpresent	...	54.0	7800.0	5.8125	no	no	no	good	no	no	nc
397	397	12.0	80.0	1.0200	0.0	0.0	normal	normal	notpresent	notpresent	...	49.0	6600.0	5.4000	no	no	no	good	no	no	nc
398	398	17.0	60.0	1.0250	0.0	0.0	normal	normal	notpresent	notpresent	...	51.0	7200.0	5.8125	no	no	no	good	no	no	nc
399	399	58.0	80.0	1.0250	0.0	0.0	normal	normal	notpresent	notpresent	...	53.0	6800.0	5.8125	no	no	no	good	no	no	nc

384 rows × 26 columns

Saving the Preprocessed dataset as a New File

In [25]:
```
dataset.to_csv("PreProcessed_kidney_disease.csv",index=False)
```