Name: Shahnwaz Khan Roll: Ac-1272 Practical 2

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

Q2. Perform the following preprocessing tasks on the dirty_iris datasetii.

In [ ]:
```
data=pd.read_csv('dirty_irisdata.csv')
display(data)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	6.4	3.2	4.5	1.5	versicolor
1	6.3	3.3	6.0	2.5	virginica
2	6.2	NaN	5.4	2.3	virginica
3	5.0	3.4	1.6	0.4	setosa
4	5.7	2.6	3.5	1.0	versicolor
...	...	...	...	...	...
145	6.7	3.1	5.6	2.4	virginica
146	5.6	3.0	4.5	1.5	versicolor
147	5.2	3.5	1.5	0.2	setosa
148	6.4	3.1	NaN	1.8	virginica
149	5.8	2.6	4.0	NaN	versicolor

150 rows × 5 columns

i) Calculate the number and percentage of observations that are complete

In [ ]:
```
complete=(data.isna().sum(axis=1)==0).values.sum()
percentage=complete*(complete/len(data))
print("Number of observations that are complete: ",complete)
print("Percentage is: ",percentage)
```

```
Number of observations that are complete:  96
Percentage is:  61.44
```

ii) Replace all the special values in data with NA

In [ ]:
```
# Infinite value is the onyl special value in our data
data.replace(np.inf,np.nan)
```

Out[ ]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	6.4	3.2	4.5	1.5	versicolor
1	6.3	3.3	6.0	2.5	virginica
2	6.2	NaN	5.4	2.3	virginica
3	5.0	3.4	1.6	0.4	setosa

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4	5.7	2.6	3.5	1.0	versicolor
...	...	...	...	...	...
145	6.7	3.1	5.6	2.4	virginica
146	5.6	3.0	4.5	1.5	versicolor
147	5.2	3.5	1.5	0.2	setosa
148	6.4	3.1	NaN	1.8	virginica
149	5.8	2.6	4.0	NaN	versicolor

iii) Define these rules in a separate text file and read them. (Use editfile function in R (package editrules). Use similar function in Python). Print the resulting constraint object.

In [ ]:
```
rules=[]
# - Species should be one of the following values: setosa, versicolor or virginic
rules.append(ruleset.checkSpecies(data))
#All measured numerical properties of an iris should be positive.
rules.append(ruleset.check_num(data))
# - The petal length of an iris is at least 2 times its petal width
rules.append(ruleset.checkPetalLength(data))
# - The sepal length of an iris cannot exceed 30 cm.
rules.append(ruleset.checkSepalLength(data))
# - The sepals of an iris are longer than its petals.
rules.append(ruleset.checkSepals(data))
```

In [ ]:
```
print(rules)
```

```
[(0, 'ruleset 1'), (1, 'ruleset 2'), (4, 'ruleset 3'), (2, 'ruleset 4'), (3, 'rule
set 5')]
```

iv)Determine how often each rule is broken (violatedEdits). Also summarize and plot the result.

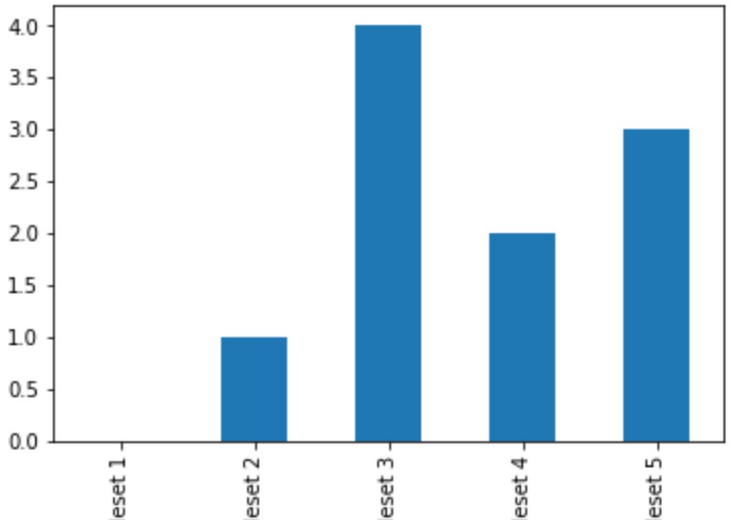
In [ ]:
```
print("Violations count in each ruleset.")
for violations in rules:
    print(violations[1]," : ",violations[0])
```

```
Violations count in each ruleset.
ruleset 1  :  0
ruleset 2  :  1
ruleset 3  :  4
ruleset 4  :  2
ruleset 5  :  3
```

In [ ]:
```
df=pd.Series()
for i in rules:
    df[i[1]]=i[0]
```

In [ ]:
```
df.plot.bar()
```

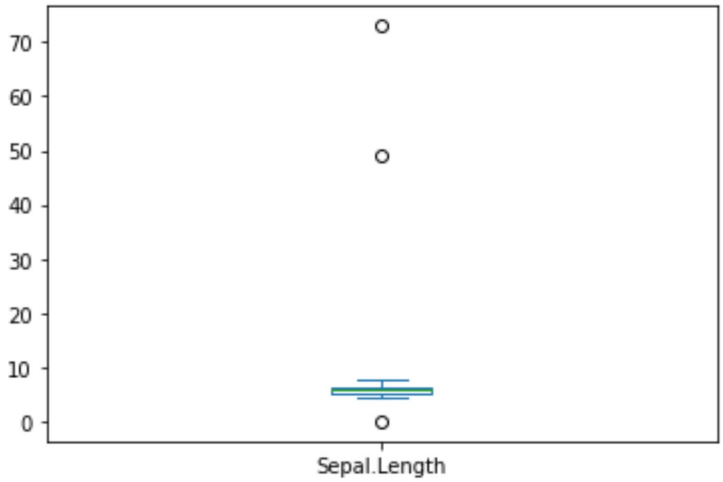
Out[ ]:
```
<AxesSubplot:>
```



v) Find outliers in sepal length using boxplot and boxplot.stats

In [ ]:
```
data['Sepal.Length'].plot.box()
```

Out[ ]: <AxesSubplot:>



In [ ]:
```

```

Ruletset

```
# - Species should be one of the following values:
setosa, versicolor or virginica.
import numpy as np
def checkSpecies(data):
  arr1=np.array(data['Species'].value_counts().index)
  count=0
  for i in arr1:
    if i not in ['setosa','versicolor','virginica']:
       count+=1
  return count,'ruleset 1'

# - All measured numerical properties of an iris
should be positive.
def check_num(data):
    count=0
    for i in data.columns:
        if data[i].dtypes == "float64":
            if data[data[i]<0].shape[0]>0:
                count=count+1
    return count, "ruleset 2"
# - The petal length of an iris is at least 2 times
its petal width.
def checkPetalLength(data):
    count=0
    for i in range(len(data)):
     if(data.loc[i]['Petal.Length']<
2*(data.loc[i]['Petal.Width'])):
      count+=1
    return count,'ruleset 3'

# - The sepal length of an iris cannot exceed 30 cm.
def checkSepalLength(data):
 count=(data['Sepal.Length']>30).sum()
```

```
 return count,'ruleset 4'

# - The sepals of an iris are longer than its petals.
def checkSepals(data):
    count=((data['Sepal.Length']+data['Sepal.Width'])<
(data['Petal.Length']+data['Petal.Width'])).sum()
    return count,'ruleset 5'
```