

```
In [2]: import pandas as pd
tidyiris= pd.read_csv("./datasets/tidyiris.csv")
```

```
In [7]: display(tidyiris)
```

	measurement	species	value
0	SepalLengthCm	Iris-setosa	5.1
1	SepalLengthCm	Iris-setosa	4.9
2	SepalLengthCm	Iris-setosa	4.7
3	SepalLengthCm	Iris-setosa	4.6
4	SepalLengthCm	Iris-setosa	5.0
...
595	PetalWidthCm	Iris-virginica	2.3
596	PetalWidthCm	Iris-virginica	1.9
597	PetalWidthCm	Iris-virginica	2.0
598	PetalWidthCm	Iris-virginica	2.3
599	PetalWidthCm	Iris-virginica	1.8

600 rows x 3 columns

1. Reading the Dataset:

```
In [9]: import pandas as pd
iris=pd.read_csv("./datasets/iris.csv")
display(iris)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
In [11]: iris.rename(columns={'SepalLengthCm': 'sepal_length',
                              'SepalWidthCm': 'sepal_width',
                              'PetalLengthCm': 'petal_length',
                              'PetalWidthCm': 'petal_width',
                              'Species': 'species'}, inplace=True)
display(iris)
```

	Id	sepal_length	sepal_width	petal_length	petal_width	species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

2. Data Summarization:

```
In [12]: # data summarization
summary=iris.describe()
display(summary)
```

	Id	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
In [13]: #median:
iris.median(axis=0, numeric_only=True)
```

```
Out[13]: Id          75.50
sepal_length    5.80
sepal_width      3.00
petal_length     4.35
petal_width      1.30
dtype: float64
```

```
In [14]: #make a df for summary:
sums= [
    [5.843333,3.054000,3.758667,1.198667],
    [5.80,3.00,4.35,1.30],
    [0.828066,0.433594,1.764420,0.763161]
]
sums = pd.DataFrame(sums, columns=['sepal_length', 'sepal_width', 'petal_length', 'petal_width'],
                    index = ['mean','median','standard deviation'])

display(sums) #yay it worked
```

	sepal_length	sepal_width	petal_length	petal_width
mean	5.843333	3.054000	3.758667	1.198667
median	5.800000	3.000000	4.350000	1.300000
standard deviation	0.828066	0.433594	1.764420	0.763161

```
In [15]: #turn df into a csv file

iris_summary=sums.to_csv("iris_summary.csv")
iris_summary=pd.read_csv("iris_summary.csv")
display(iris_summary) #yay it worked again
```

	Unnamed: 0	sepal_length	sepal_width	petal_length	petal_width
0	mean	5.843333	3.054000	3.758667	1.198667
1	median	5.800000	3.000000	4.350000	1.300000
2	standard deviation	0.828066	0.433594	1.764420	0.763161

3. Data Visualization:

pairplot:

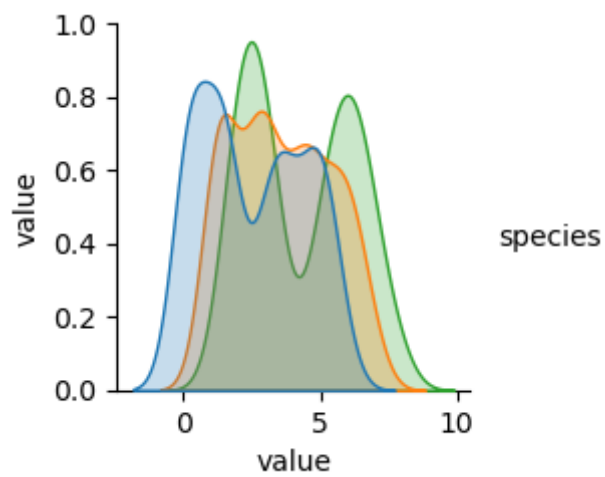
```
In [10]: import seaborn as sns
display(tidyiris)
```

	measurement	species	value
0	SepalLengthCm	Iris-setosa	5.1
1	SepalLengthCm	Iris-setosa	4.9
2	SepalLengthCm	Iris-setosa	4.7
3	SepalLengthCm	Iris-setosa	4.6
4	SepalLengthCm	Iris-setosa	5.0
...
595	PetalWidthCm	Iris-virginica	2.3
596	PetalWidthCm	Iris-virginica	1.9
597	PetalWidthCm	Iris-virginica	2.0
598	PetalWidthCm	Iris-virginica	2.3
599	PetalWidthCm	Iris-virginica	1.8

600 rows x 3 columns

```
In [67]: sns.pairplot(data=tidyiris, hue="species")
```

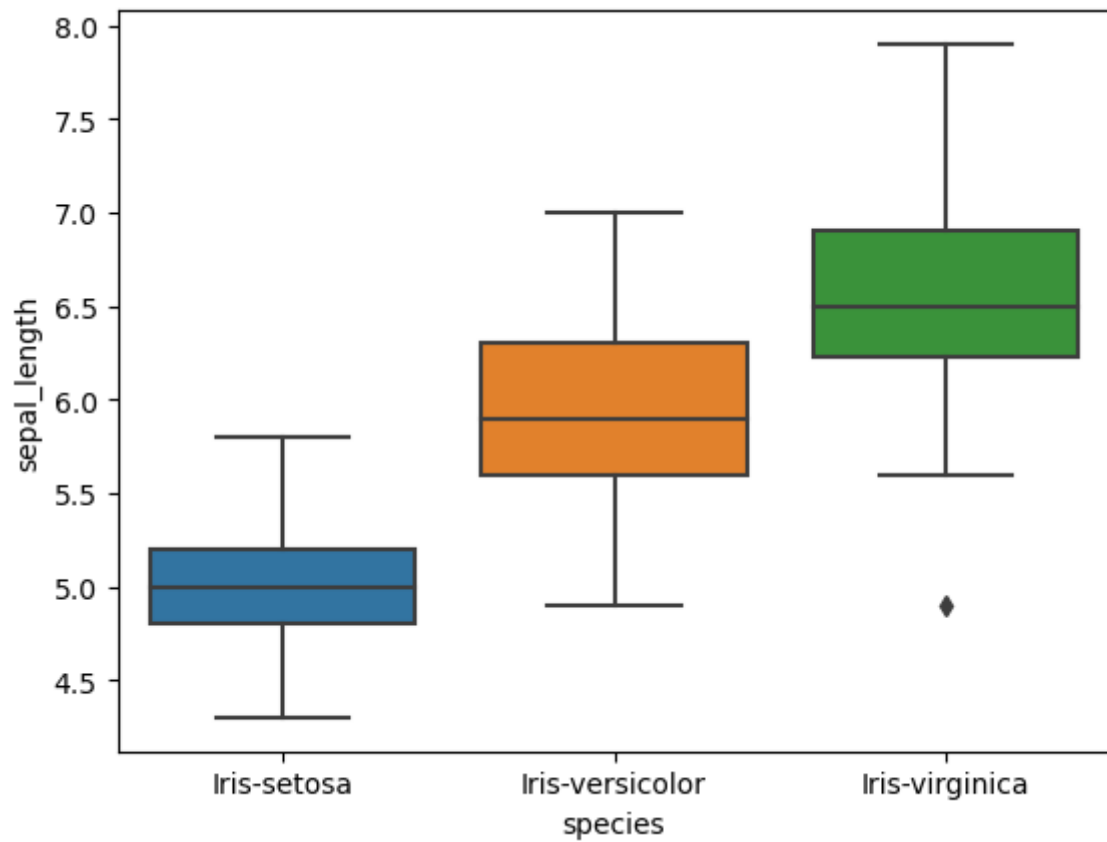
```
Out[67]: <seaborn.axisgrid.PairGrid at 0x16b2113d0>
```



boxplot:

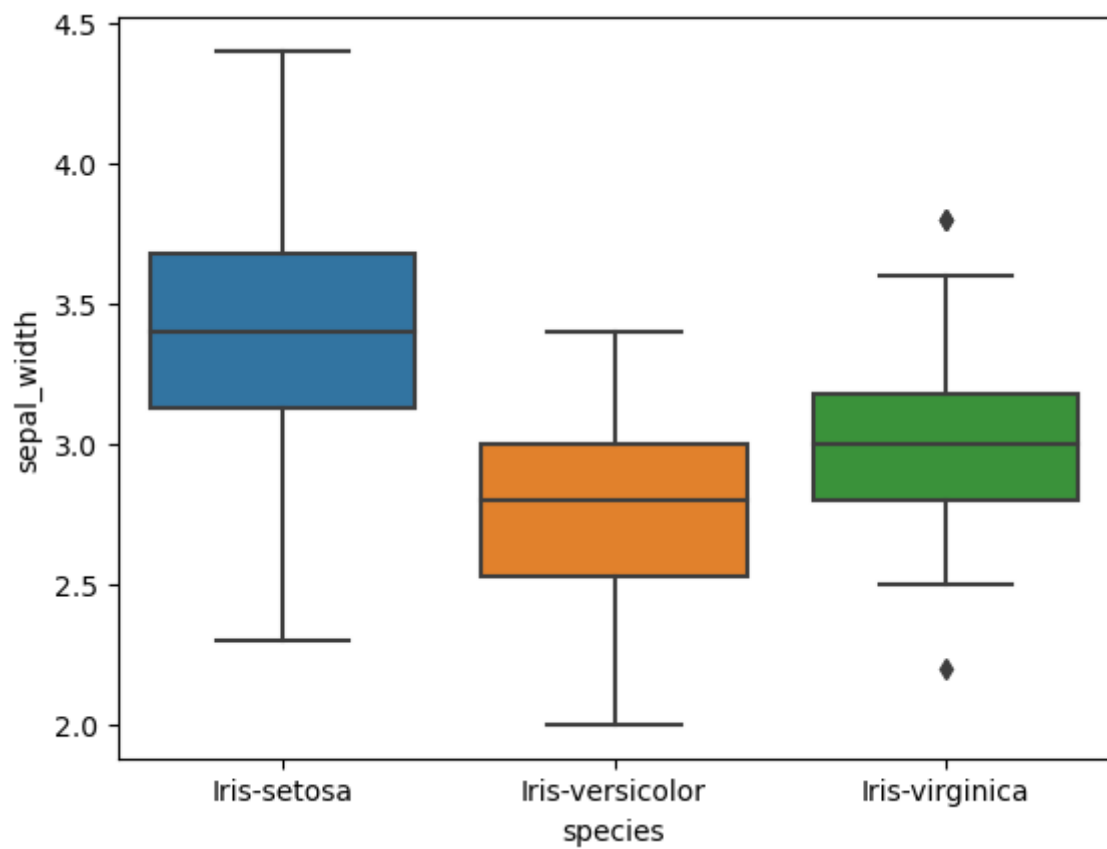
```
In [69]: sns.boxplot(data=iris, y='sepal_length', x='species')
```

```
Out[69]: <Axes: xlabel='species', ylabel='sepal_length'>
```



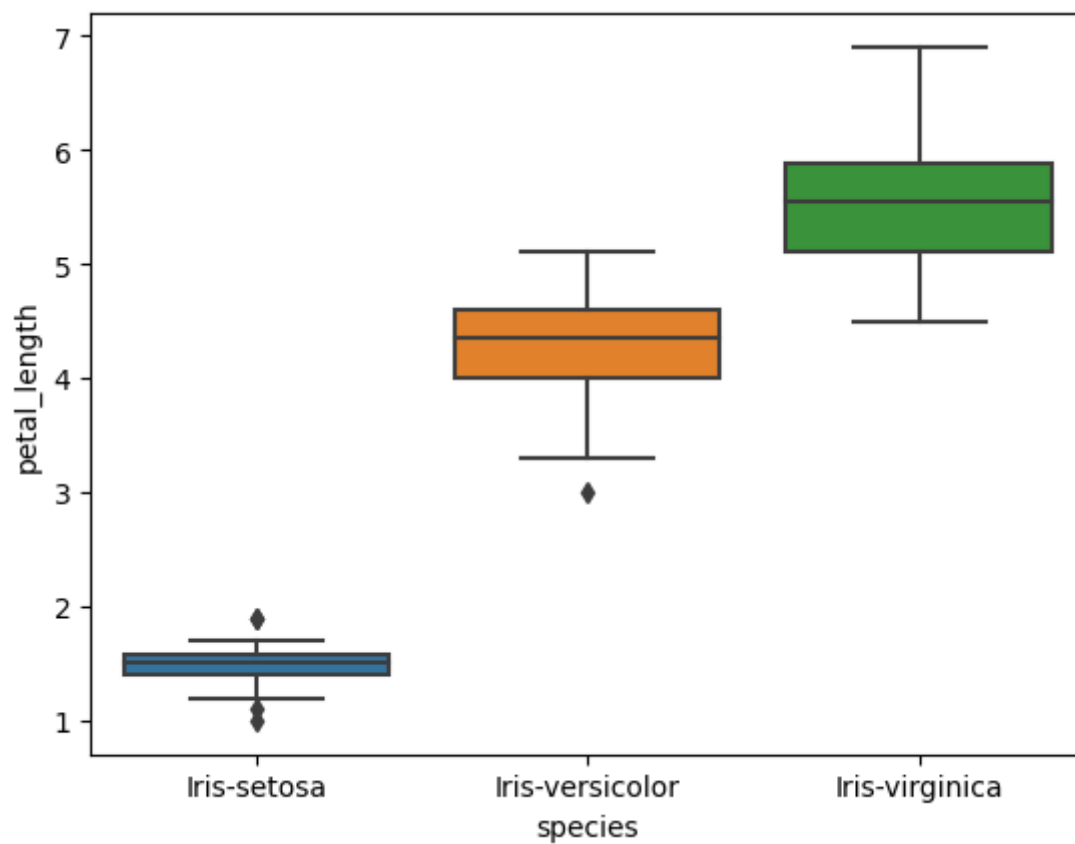
```
In [35]: sns.boxplot(data=iris, y='sepal_width', x='species')
```

```
Out[35]: <Axes: xlabel='species', ylabel='sepal_width'>
```



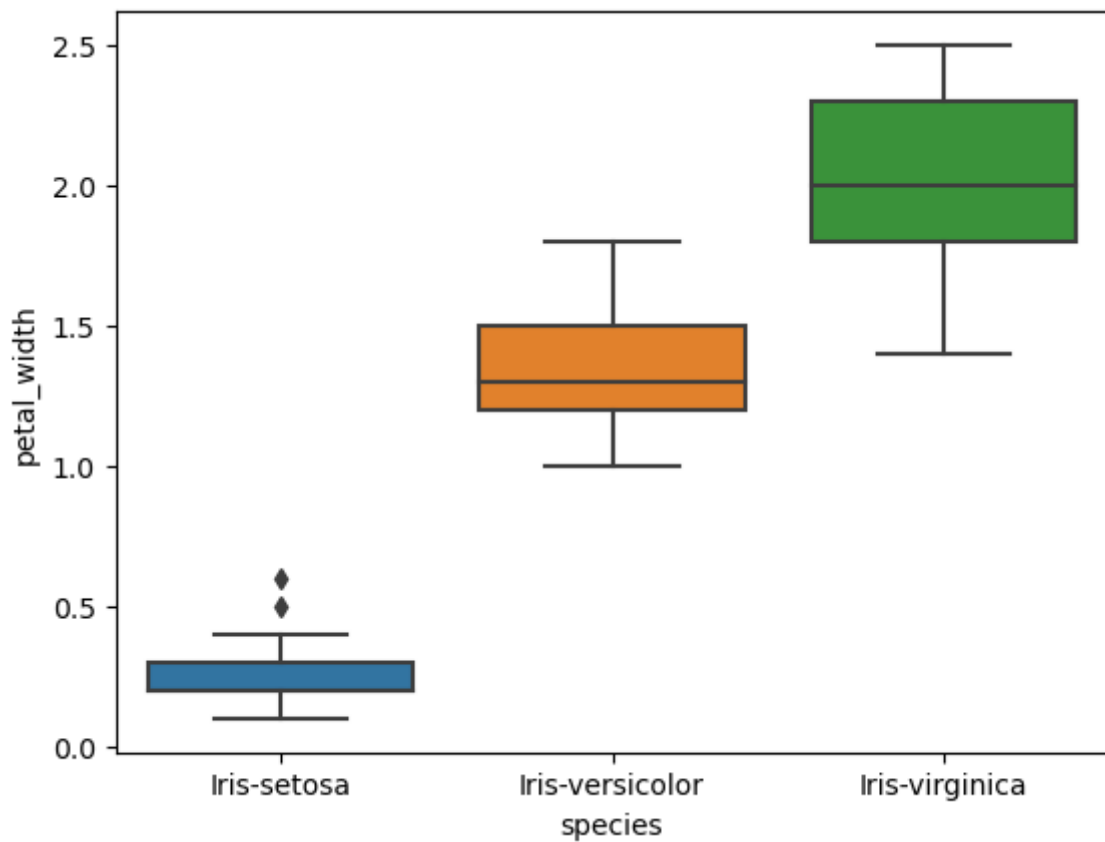
```
In [36]: sns.boxplot(data=iris, y='petal_length', x='species')
```

```
Out[36]: <Axes: xlabel='species', ylabel='petal_length'>
```



```
In [37]: sns.boxplot(data=iris, y='petal_width', x='species')
```

```
Out[37]: <Axes: xlabel='species', ylabel='petal_width'>
```



4. Plotting Function:

```
In [20]: import seaborn as sns
```

```
In [65]: def plotiris(iris, plot_type=2):  
    '''  
    1 = Strip plot  
    2 = Violin plot  
    3 = Box plot  
    default = Violin plot  
    '''  
    if plot_type == 1:  
        my_plot = sns.stripplot(data=iris)  
        return(my_plot)  
    elif plot_type == 2:  
        my_plot = sns.violinplot(data=iris)  
        return(my_plot)  
    elif plot_type == 3:  
        my_plot = sns.boxplot(data=iris)  
        return(my_plot)  
    else:  
        print("Invalid input. Try again")
```

```
In [66]: help(plotiris)
```

Help on function plotiris in module __main__:

```
plotiris(iris, plot_type=2)
  1 = Strip plot
  2 = Violin plot
  3 = Box plot
  default = Violin plot
```

In [60]: `plotiris(tidyiris,1)`

Out[60]: `<Axes: >`

