importing useful libraries

In [2]:
```
from sklearn.cluster import KMeans
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import numpy as np
```

importing dataset

In [10]:
```
d=pd.read_csv("Iris.csv")
d
```

Out[10]:

	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]:
```
import pandas_profiling
d.profile_report()
```

Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Overview Warnings 4 Reproduction

Dataset statistics

Number of variables	6
Number of observations	150
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	7.2 KiB
Average record size in memory	48.9 B

Variable types

NUM	5
CAT	1

Variables

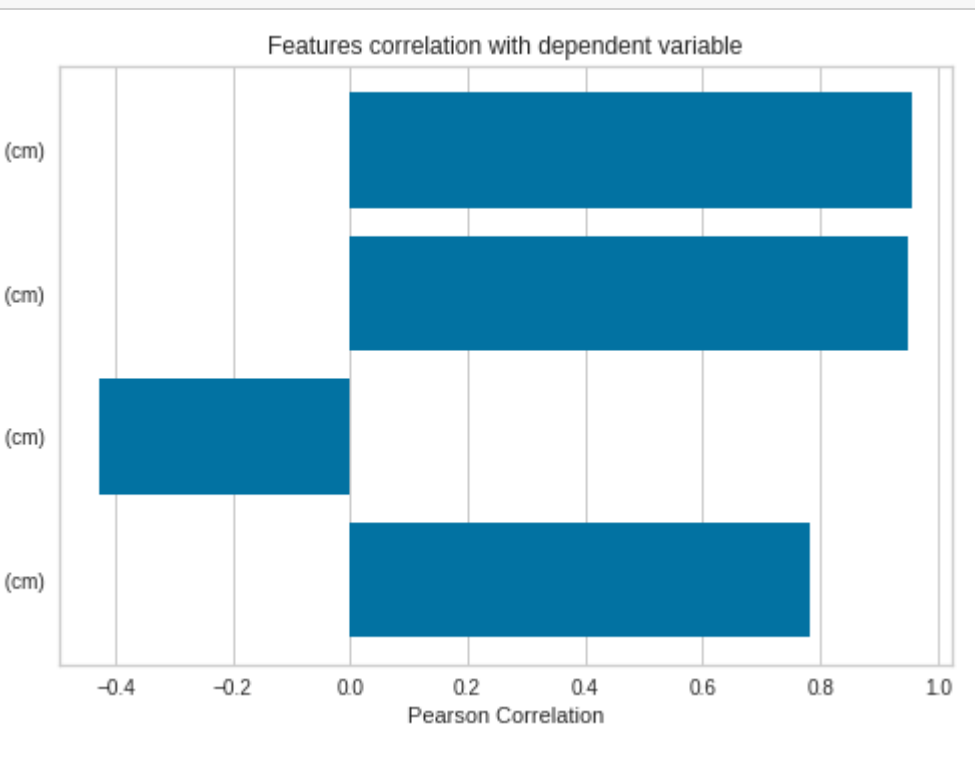
Id
Real number ($\mathbb{R}_{\geq 0}$)

Distinct	150	Mean	75.5
Distinct (%)	100.0%	Minimum	1

Out[11]:

Data Visualization

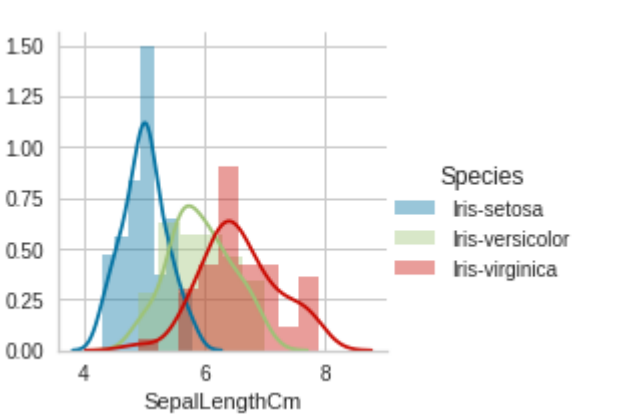
In [12]:
```
import numpy as np
from sklearn import datasets
import matplotlib.pyplot as plt
from yellowbrick.target.feature_correlation import feature_correlation

#Load the iris dataset
data = datasets.load_iris()
X, y = data['data'], data['target']

features = np.array(data['feature_names'])
visualizer = feature_correlation(X, y, labels=features)
plt.tight_layout()
```

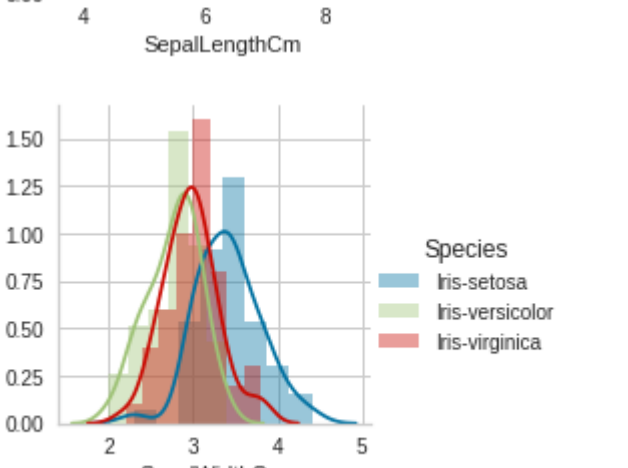


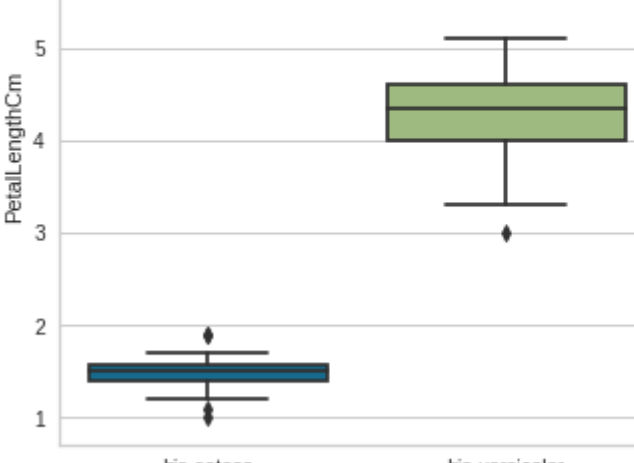

```
<Figure size 576x396 with 0 Axes>
```

In [21]:
```
iris_setosa=d.loc[d["Species"]=="Iris-setosa"]
iris_virginica=d.loc[d["Species"]=="Iris-virginica"]
iris_versicolor=d.loc[d["Species"]=="Iris-versicolor"]
sns.FacetGrid(d,hue="Species",size=3).map(sns.distplot,"PetalLengthCm").add_legend()
sns.FacetGrid(d,hue="Species",size=3).map(sns.distplot,"PetalWidthCm").add_legend()
sns.FacetGrid(d,hue="Species",size=3).map(sns.distplot,"SepalLengthCm").add_legend()
sns.FacetGrid(d,hue="Species",size=3).map(sns.distplot,"SepalWidthCm").add_legend()
plt.show()
```

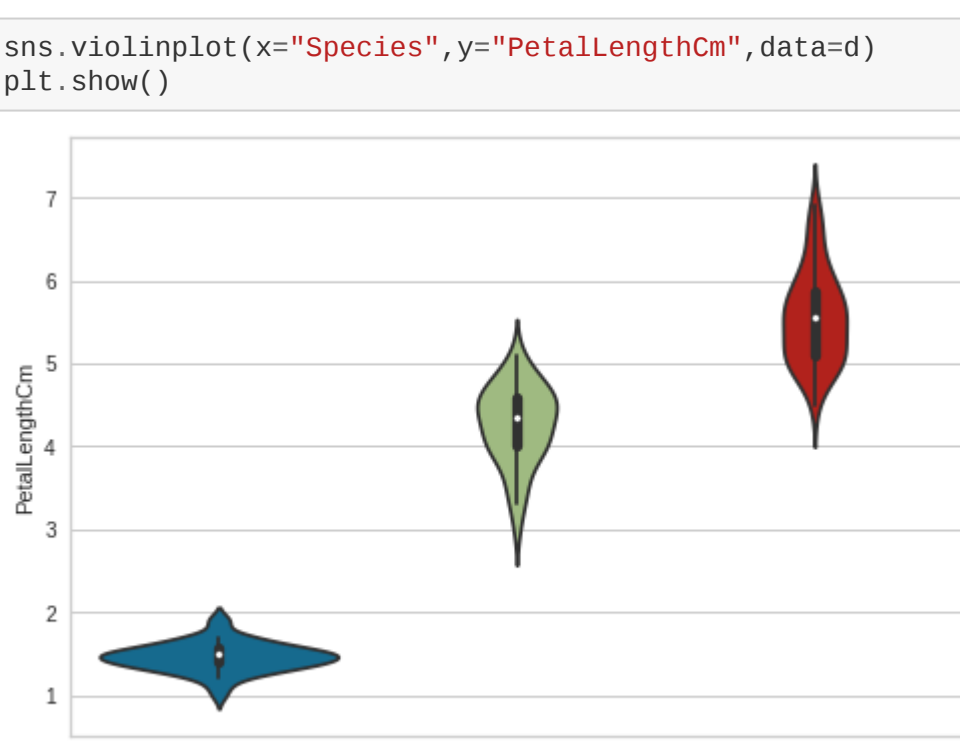
```
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:316: UserWarning: The
`size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:316: UserWarning: The
`size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:316: UserWarning: The
`size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:316: UserWarning: The
`size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/distributions.py:2551: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



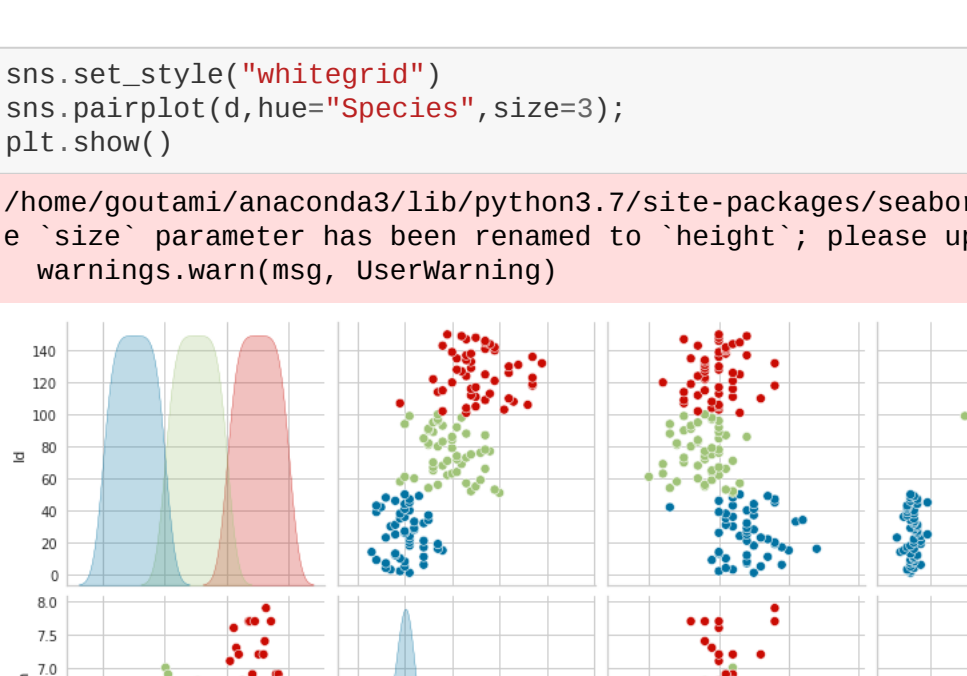




In [22]:
```
sns.boxplot(x="Species",y="PetalLengthCm",data=d)
plt.show()
```

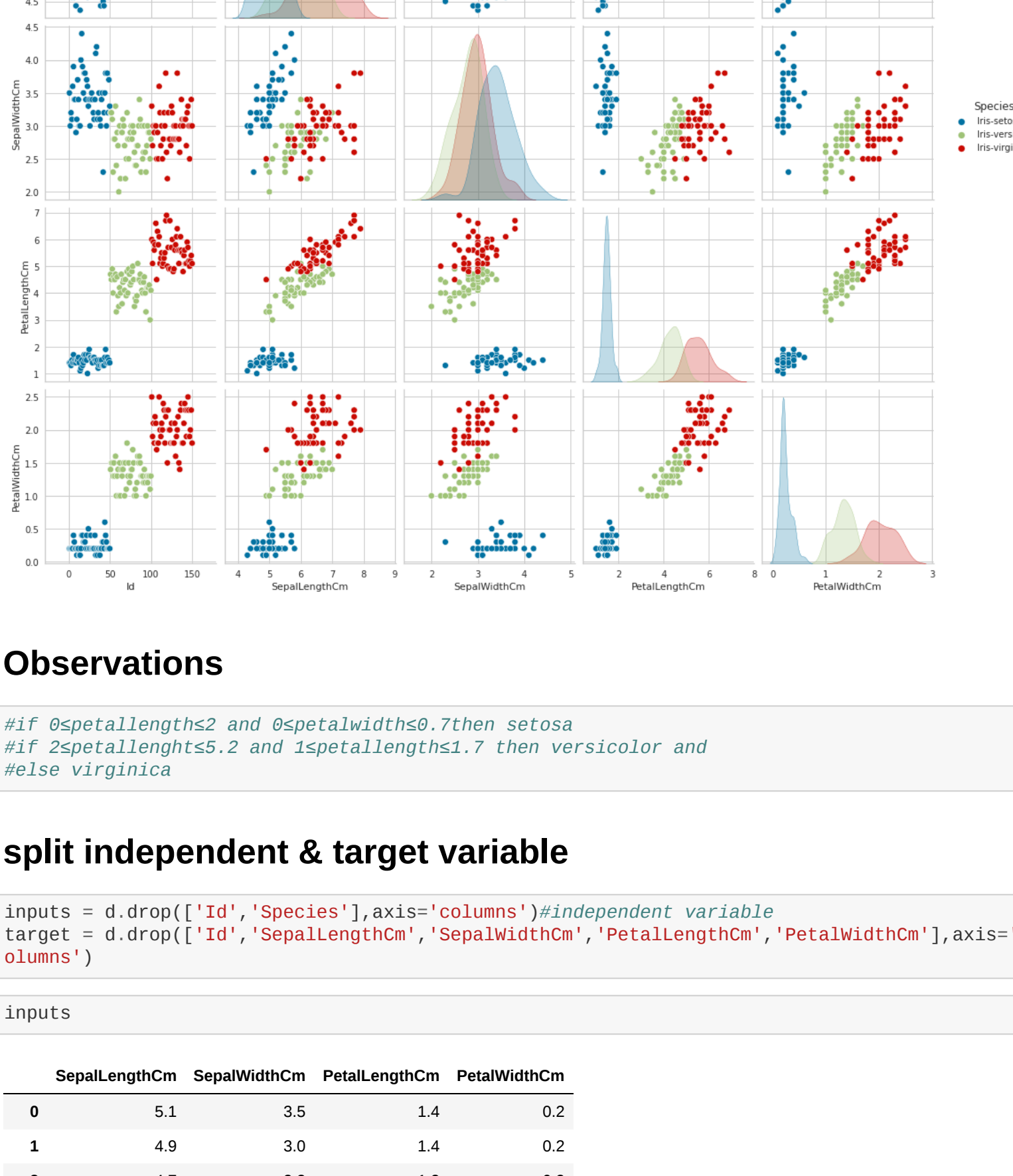


In [23]:
```
sns.violinplot(x="Species",y="PetalLengthCm",data=d)
plt.show()
```



In [29]:
```
sns.set_style("whitegrid")
sns.pairplot(d,hue="Species",size=3);
plt.show()
```

```
/home/goutami/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:1912: UserWarning: Th
e `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



Observations

In [31]:
```
#if 0≤petallength≤2 and 0≤petalwidth≤0.7then setosa
#if 2≤petallenght≤5.2 and 1≤petallength≤1.7 then versicolor and
#else virginica
```

split independent & target variable

In [32]:
```
inputs = d.drop(['Id','Species'],axis='columns')#independent variable
target = d.drop(['Id','SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],axis='c
olumns')
```

In [33]:
```
inputs
```

Out[33]:

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

150 rows × 4 columns

In [34]:
```
target
```

Out[34]:

	Species
0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
...	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica

150 rows × 1 columns

In [35]:
```
#encode categorical feature
from sklearn.preprocessing import LabelEncoder
```

In [36]:
```
le_Species = LabelEncoder()
```

In [37]:
```
target['Species'] =le_Species.fit_transform(target['Species'])
target.head(70)
```

Out[37]:

	Species
0	0
1	0
2	0
3	0
4	0
...	...
65	1
66	1
67	1
68	1
69	1

70 rows × 1 columns

Decision Tree model implementation

In [39]:
```
from sklearn import tree
```

In [40]:
```
model = tree.DecisionTreeClassifier()
```

In [41]:
```
model.fit(inputs,target)
```

Out[41]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

value prediction

In [42]:
```
model.predict([[5.1,3.5,1.4,0.2]])#iris setosa
```

Out[42]:
```
array([0])
```

In [43]:
```
model.predict([[6.7,3.0,5.2,2.3]])#iris virginica
```

Out[43]:
```
array([2])
```

In [44]:
```
model.predict([[6.4,2.9,4.3,1.3]])#iris versicolor
```

Out[44]:
```
array([1])
```

In [34]:
```
model.predict([[6.6,3.9,5.3,1.3]])
```

Out[34]:
```
array([2])
```

model accuracy

In [45]:
```
score=model.score(inputs,target)
score
```

Out[45]:
```
1.0
```

Thank you

Submitted By Goutami Dey

The Sparks Foundation