In [ ]:

```
NAME : ARYAN SIRDESAI
ROLL No. : TACO20175
Lab Assignment 3 : Data Wrangling II

Problem Statement : Descriptive Statistics - Measures of Central Tendency and vari
open source dataset (e.g., data.csv)
1. Provide summary statistics (mean, median, minimum, maximum, standard deviation)
income etc.) with numeric variables grouped by one of the qualitative (categorical
if your categorical variable is age groups and quantitative variable is income, the
statistics of income grouped by the age groups. Create a list that contains a numer
to the categorical variable.
2. Write a Python program to display some basic statistical details like percentil
etc. of the species of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris-versicolor’ of i
```

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
df=pd.read_csv("apmc.csv")
```

loading dataset

Dataset name : Agricultural Produce & Livestock Market Committee (APMC)

In [3]:

```
df.head(10)
```

Out[3]:

	APMC	Commodity	Year	Month	arrivals_in_qtl	min_price	max_price	modal_price
0	Ahmednagar	Bajri	2015	April	79	1406	1538	1463
1	Ahmednagar	Bajri	2016	April	106	1788	1925	1875
2	Ahmednagar	Wheat(Husked)	2015	April	1253	1572	1890	1731
3	Ahmednagar	Wheat(Husked)	2016	April	387	1750	2220	1999
4	Ahmednagar	Sorgum(Jawar)	2015	April	3825	1600	2200	1900
5	Ahmednagar	Sorgum(Jawar)	2016	April	2093	1695	2454	2119
6	Ahmednagar	Maize	2015	April	75	1345	1401	1373
7	Ahmednagar	Maize	2016	April	155	1367	1392	1375
8	Ahmednagar	Gram	2015	April	1794	3533	3762	3647
9	Ahmednagar	Gram	2016	April	630	4790	5553	5216

In [4]:

```
df.tail(10)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[4]:

	APMC	Commodity	Year	Month	arrivals_in_qtl	min_price	max_price	modal
62419	Shevgaon-Bodhegaon	SOYBEAN	2016	November	2	2650	2650	
62420	Shrigonda	BAJRI	2016	November	308	1083	1483	
62421	Shrigonda	WHEAT(HUSKED)	2016	November	231	1558	2100	
62422	Shrigonda	SORGUM(JAWAR)	2016	November	70	1767	2117	
62423	Shrigonda	MAIZE	2016	November	1872	1108	1333	
62424	Shrigonda	GRAM	2016	November	586	5700	6367	
62425	Shrigonda	GREEN GRAM	2016	November	2	5000	5000	
62426	Shrigonda	BLACK GRAM	2016	November	46	4700	6933	
62427	Shrigonda	SOYBEAN	2016	November	166	2583	2708	
62428	Shrigonda	SUNFLOWER	2016	November	74	2933	3200	

Description

In [5]:

```
df.describe()
```

Out[5]:

	Year	arrivals_in_qtl	min_price	max_price	modal_price
count	62429.000000	6.242900e+04	6.242900e+04	6.242900e+04	62429.000000
mean	2015.337503	6.043088e+03	2.945228e+03	3.688814e+03	3296.003989
std	0.690451	3.470331e+04	1.318396e+04	7.662962e+03	3607.792534
min	2014.000000	1.000000e+00	0.000000e+00	0.000000e+00	0.000000
25%	2015.000000	3.800000e+01	1.250000e+03	1.600000e+03	1450.000000
50%	2015.000000	2.110000e+02	1.976000e+03	2.797000e+03	2425.000000
75%	2016.000000	1.364000e+03	3.900000e+03	4.647000e+03	4257.000000
max	2016.000000	1.450254e+06	3.153038e+06	1.600090e+06	142344.000000

In [6]:

```
df.columns
```

Out[6]:

```
Index(['APMC', 'Commodity', 'Year', 'Month', 'arrivals_in_qtl', 'min_price',
       'max_price', 'modal_price', 'date', 'district_name', 'state_name'],
      dtype='object')
```

In [7]:

```
df.isnull().sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[7]: APMC              0
        Commodity         0
        Year              0
        Month             0
        arrivals_in_qtl   0
        min_price         0
        max_price         0
        modal_price       0
        date              0
        district_name     0
        state_name        0
        dtype: int64
```

```
In [8]: df.shape
```

```
Out[8]: (62429, 11)
```

```
In [9]: df.dtypes
```

```
Out[9]: APMC               object
        Commodity          object
        Year                int64
        Month              object
        arrivals_in_qtl     int64
        min_price           int64
        max_price           int64
        modal_price         int64
        date               object
        district_name      object
        state_name         object
        dtype: object
```

1. Provide summary statistics

```
In [10]: df_min_price = df.groupby(['Commodity'], as_index=False).agg(mean=('min_price', 'm
```

Statistical data for min_price attribute over Commodity

```
In [11]: df_min_price.head(20)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[11]:

	Commodity	mean	minimum	maximum	median	standard_deviation
0	AMBAT CHUKA	1014.500000	815	1214	1014.5	282.135606
1	AMLA	1100.000000	1100	1100	1100.0	NaN
2	APPLE	4960.000000	2853	7800	4514.5	1835.697748
3	ARVI	1804.500000	1150	2459	1804.5	925.602777
4	AWALA	1450.000000	800	1800	1750.0	563.471383
5	Amba Koy	2791.666667	1000	4125	3250.0	1612.128510
6	Ambat Chuka	314.843750	0	1031	141.5	385.813268
7	Amla	1078.086957	600	1800	1100.0	321.881671
8	Apple	5311.246637	300	21846	4519.0	3288.269315
9	Arvi	1952.692308	870	2933	1922.5	640.489903
10	Aster	7066.666667	6200	8000	7000.0	901.849951
11	Awala	1151.133333	550	3100	1000.0	648.467091
12	BAJRI	1315.479167	900	2126	1302.5	194.338986
13	BANANA	1234.800000	221	6778	617.0	1784.814244
14	BATBATI	3756.500000	3013	4500	3756.5	1051.467784
15	BEET ROOT	1023.666667	233	3000	864.5	797.098526
16	BETELNUTS	22285.000000	22285	22285	22285.0	NaN
17	BHAGAR/VARI	1413.000000	1413	1413	1413.0	NaN
18	BITTER GOURD	1180.052632	103	2533	1077.5	546.471134
19	BLACK GRAM	4970.423729	2225	7600	5110.0	1119.713443

In [12]:

```
df_min_price
```

Out[12]:

	Commodity	mean	minimum	maximum	median	standard_deviation
0	AMBAT CHUKA	1014.500000	815	1214	1014.5	282.135606
1	AMLA	1100.000000	1100	1100	1100.0	NaN
2	APPLE	4960.000000	2853	7800	4514.5	1835.697748
3	ARVI	1804.500000	1150	2459	1804.5	925.602777
4	AWALA	1450.000000	800	1800	1750.0	563.471383
...	...	...	...	...	...	...
347	Water Melon	581.156463	8	8278	464.0	865.400000
348	Wheat(Husked)	1555.045616	0	5761	1513.0	250.978731
349	Wheat(Unhusked)	2964.236025	1500	4057	2999.0	531.065621
350	Wood Apple	1325.000000	500	2000	1400.0	788.986692
351	Zendu	2122.727273	300	5000	2000.0	1271.728340

352 rows × 6 columns

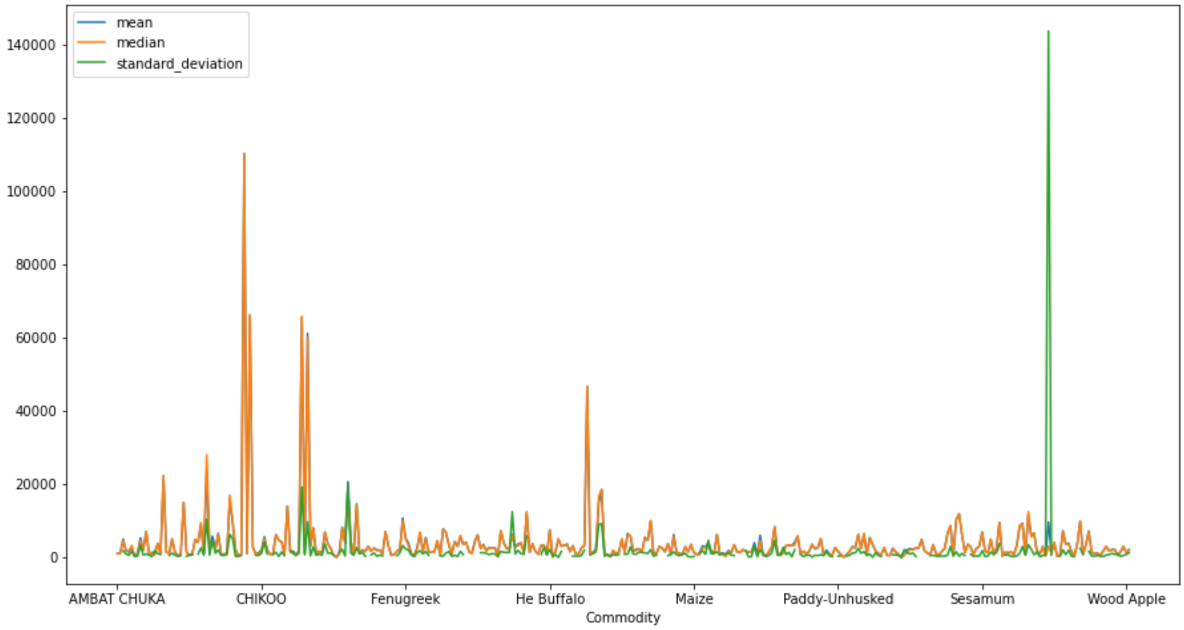
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [13]:

```
# plt.plot(df_min_price['mean'])
# plt.axhline(df_min_price['median'])
df_min_price.plot('Commodity',['mean','median','standard_deviation'],figsize=(15,
```

Out[13]:

```
<AxesSubplot:xlabel='Commodity'>
```



In [14]:

```
apmc= df.groupby(['APMC'], as_index=False).agg(mean=('min_price', 'mean'),minimum=
```

In [15]:

```
apmc
```

Out[15]:

	APMC	mean	minimum	maximum	median	standard_deviation
0	Aamgaon	1510.740741	1348	1651	1505.0	87.818691
1	Aarni	4254.607143	863	10775	3750.0	2250.922756
2	Achalpur	3843.046154	1275	12250	3418.0	2293.906218
3	Aheri	3374.550239	615	8000	3300.0	1846.452547
4	Ahmednagar	2640.573190	2	15000	1650.0	2458.195905
...	...	...	...	...	...	...
344	Washim-Ansing	4810.446809	1233	10000	4863.0	2747.413327
345	Yawal	2396.544643	226	7700	1371.5	1888.065438
346	Yeola	2659.588542	125	8500	1584.5	2133.817721
347	Yeotmal	4274.560000	1297	10720	3979.0	2089.495522
348	Zarijamini	4122.250000	1166	9055	3878.5	1729.358080

349 rows × 6 columns

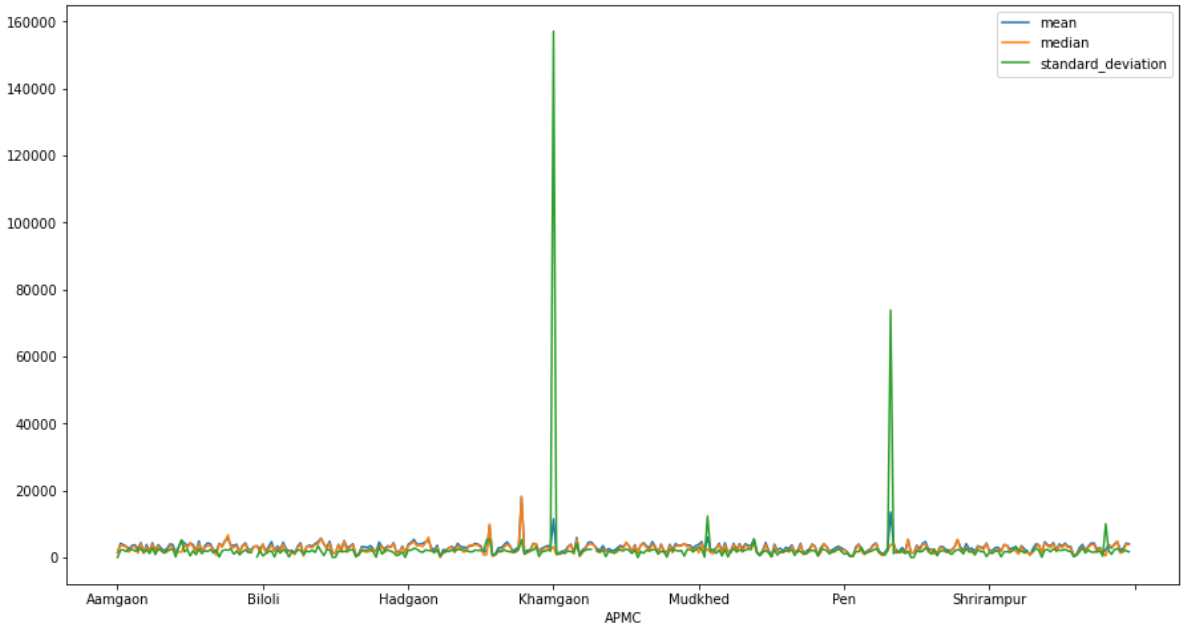
In [16]:

```
apmc.plot('APMC',['mean','median','standard_deviation'],figsize=(15, 8))
```

Out[16]:

```
<AxesSubplot:xlabel='APMC'>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



2. Iris dataset

In [17]:
```
iris = pd.read_csv('iris.csv')
```

In [18]:
```
iris
```

Out[18]:

	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 [19]:
```
mean_df = iris.groupby(['Species'], as_index=False).agg('mean')
```

In [20]:
```
mean_df
```

Out[20]:

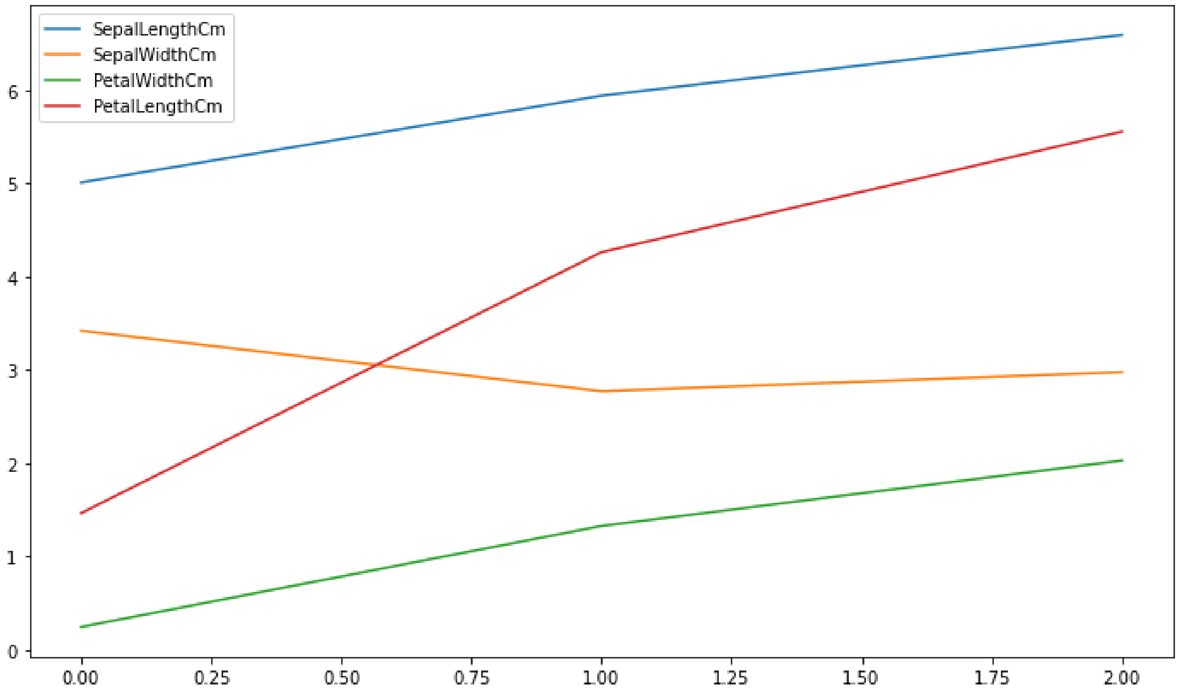
	Species	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	Iris-setosa	25.5	5.006	3.418	1.464	0.244
1	Iris-versicolor	75.5	5.936	2.770	4.260	1.326
2	Iris-virginica	125.5	6.588	2.974	5.552	2.026

In [21]:

```
mean_df[['SepalLengthCm','SepalWidthCm','PetalWidthCm','PetalLengthCm']].plot(figs
```

Out[21]:

```
<AxesSubplot:>
```



In [22]:

```
std_df = iris.groupby(['Species'], as_index=False).agg('std')
```

In [23]:

```
std_df
```

Out[23]:

	Species	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	Iris-setosa	14.57738	0.352490	0.381024	0.173511	0.107210
1	Iris-versicolor	14.57738	0.516171	0.313798	0.469911	0.197753
2	Iris-virginica	14.57738	0.635880	0.322497	0.551895	0.274650

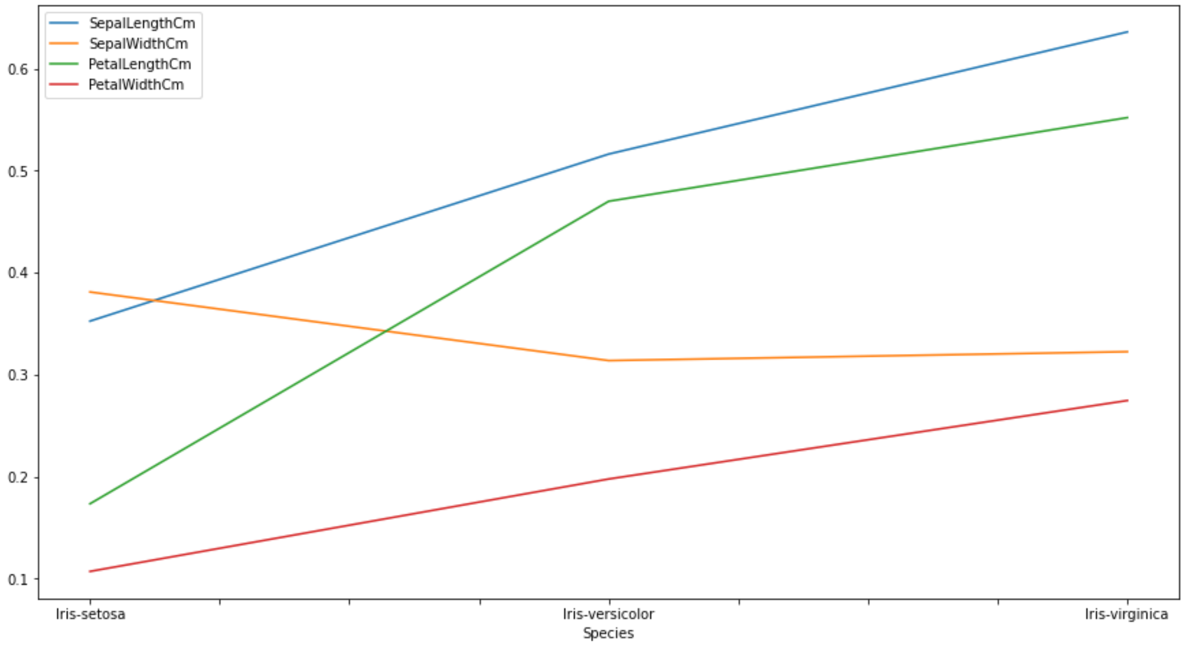
In [24]:

```
std_df.plot('Species',['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthC
```

Out[24]:

```
<AxesSubplot:xlabel='Species'>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [25]:
```
med_iris = iris.groupby(['Species'], as_index=False).agg('median')
```

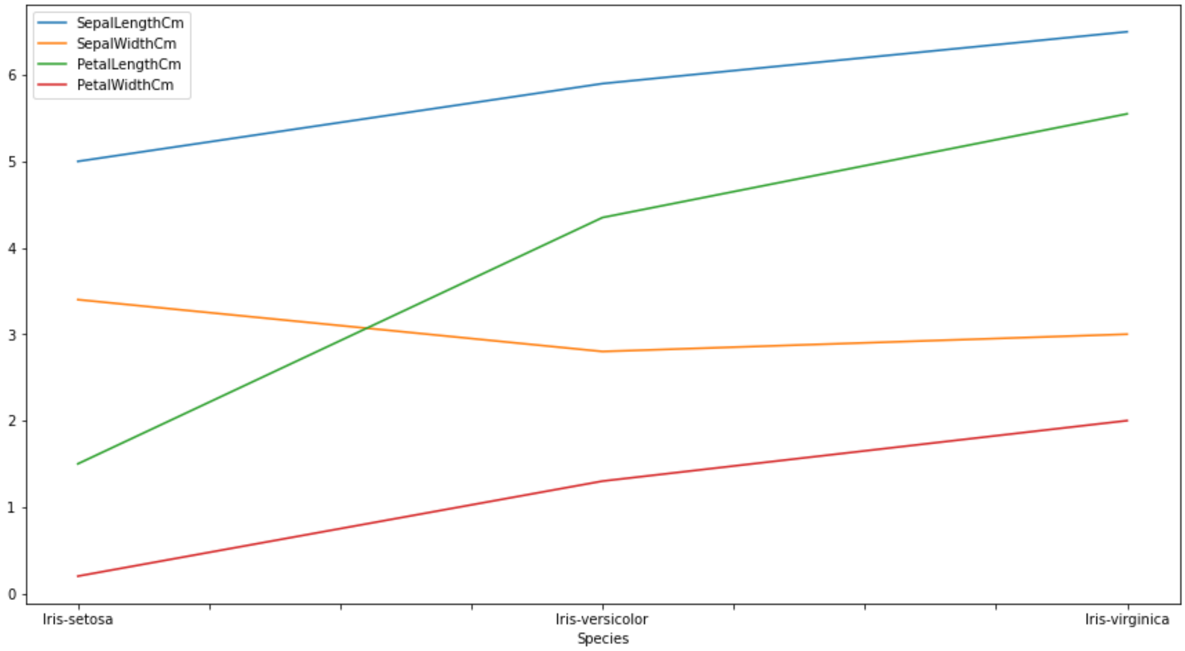
In [26]:
```
med_iris
```

Out[26]:

	Species	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	Iris-setosa	25.5	5.0	3.4	1.50	0.2
1	Iris-versicolor	75.5	5.9	2.8	4.35	1.3
2	Iris-virginica	125.5	6.5	3.0	5.55	2.0

In [27]:
```
med_iris.plot('Species',['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidtl
```

Out[27]:
```
<AxesSubplot:xlabel='Species'>
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js