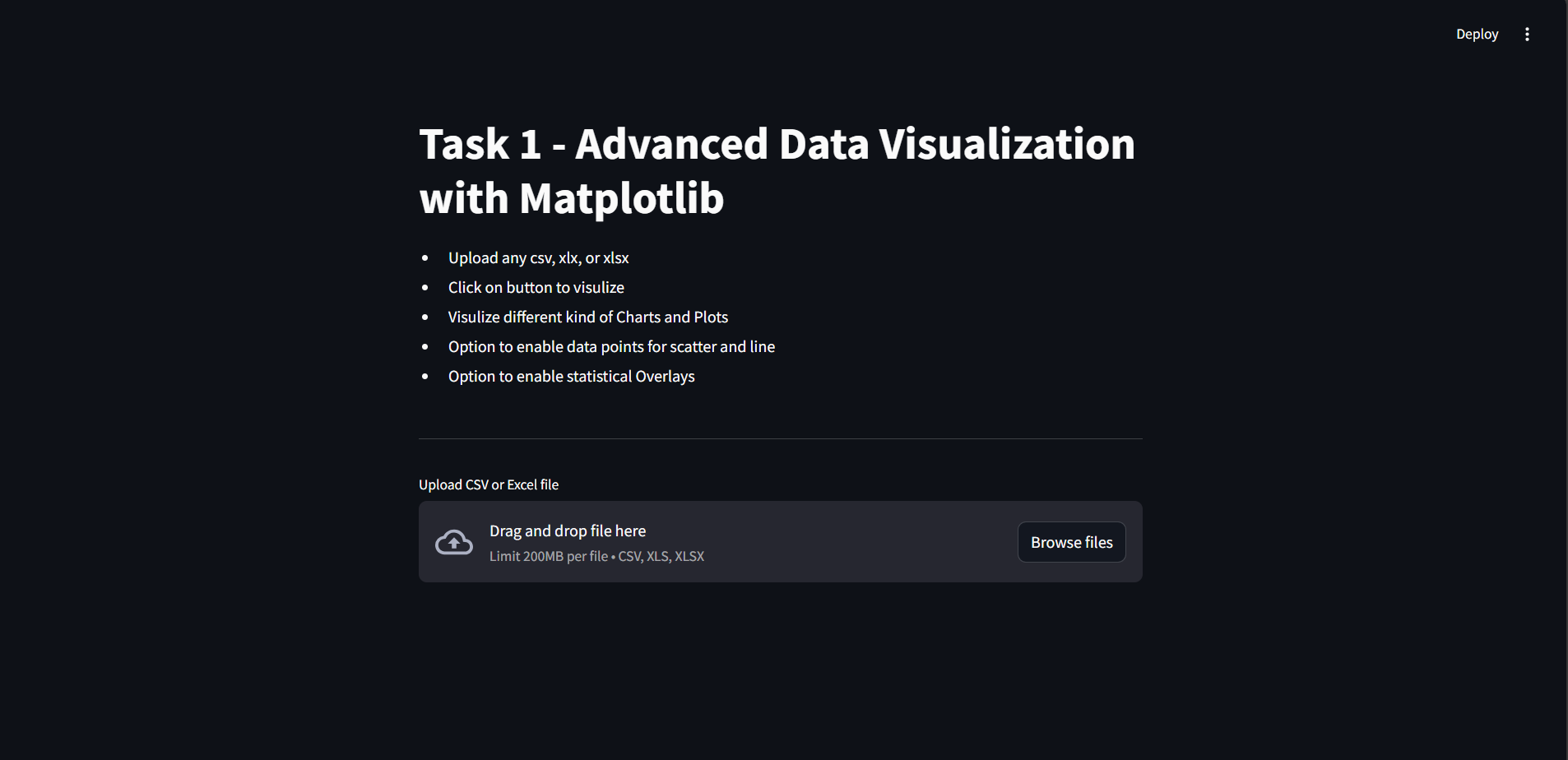
**Task 1 - Advanced Data Visualization with Matplotlib**

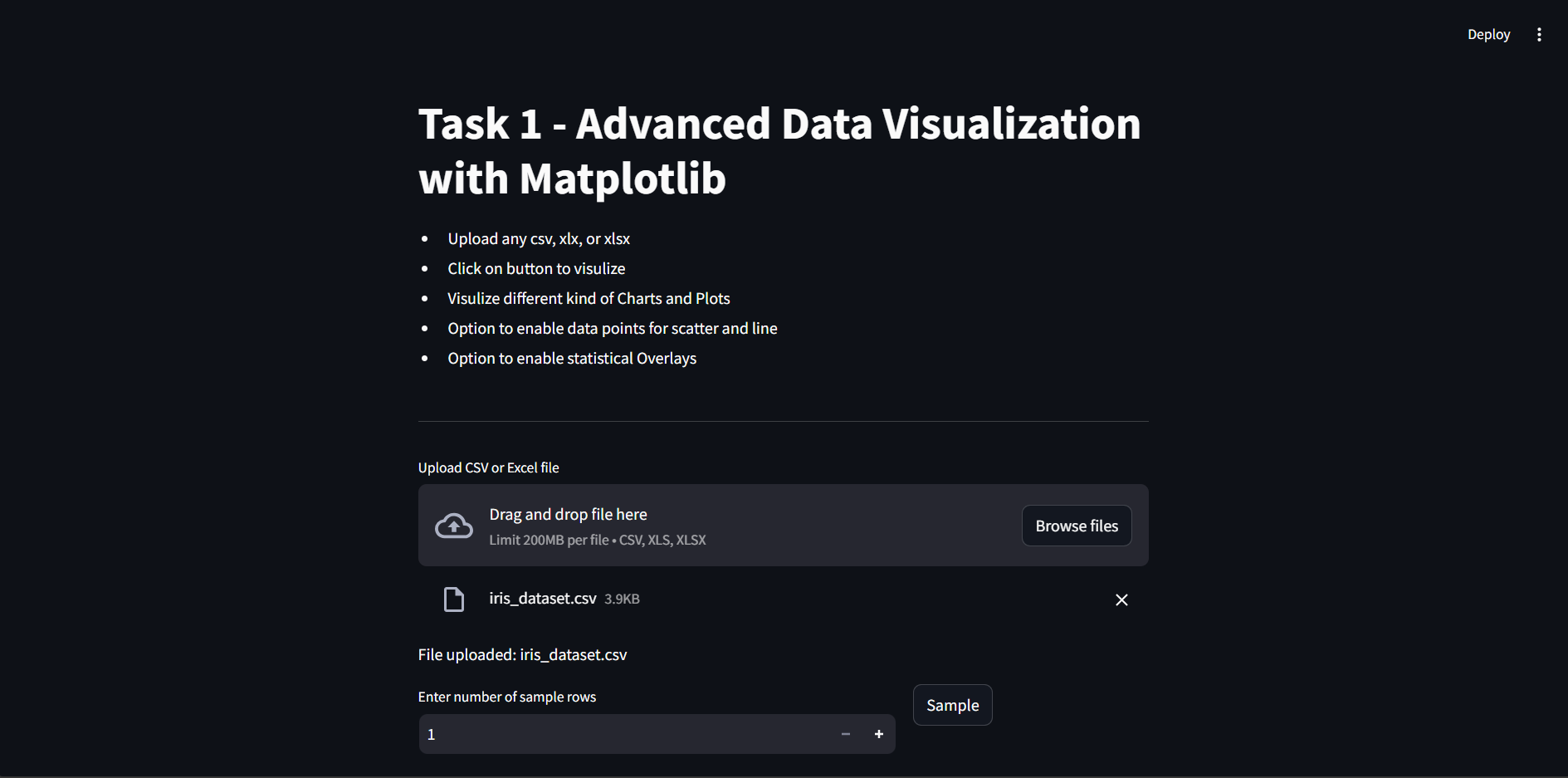
1. **Task Description**

Combine Matplotlib with pandas to create advanced data visualizations with labeled data points and statistical overlays.

1. **Task Output Screenshot**



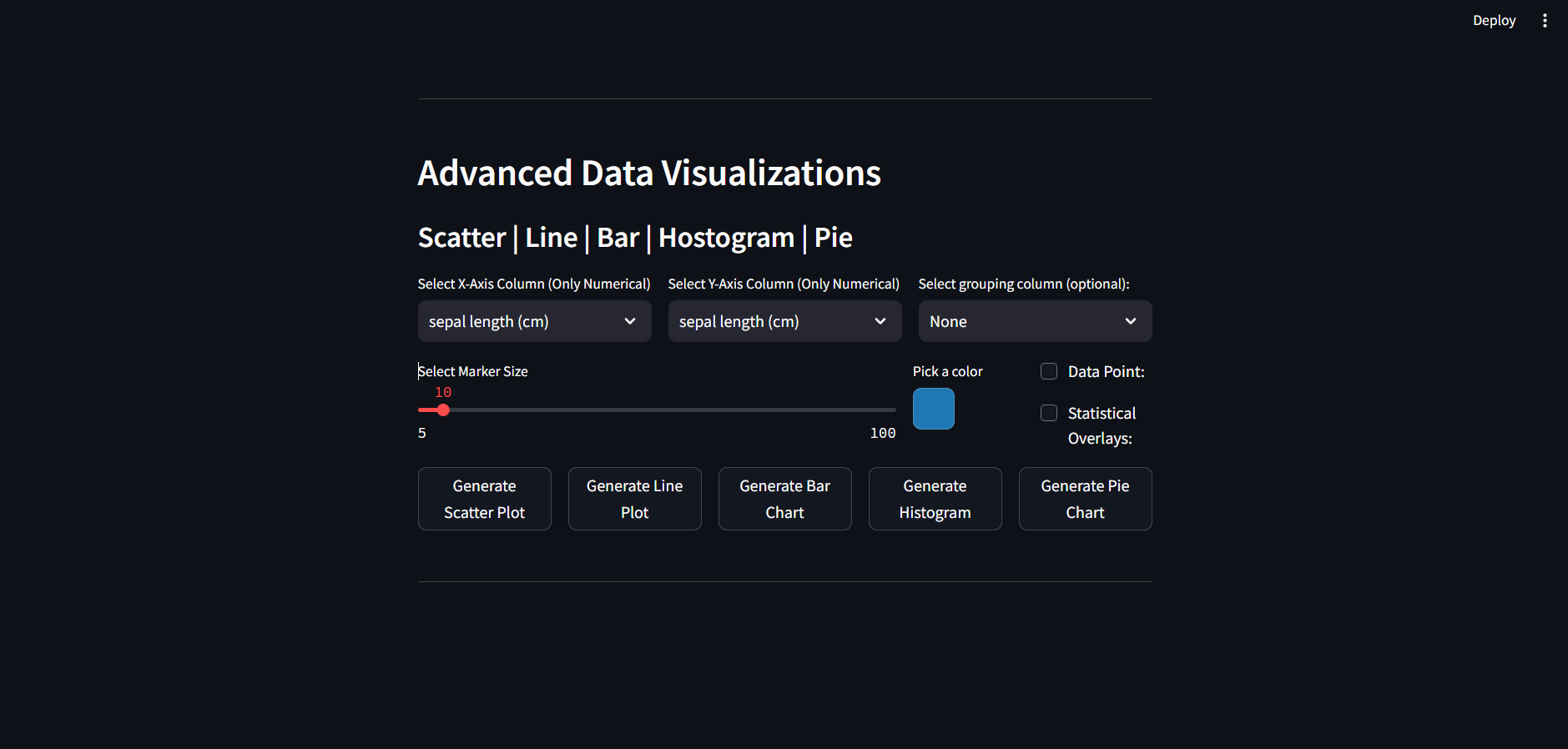
**After Selecting File:**

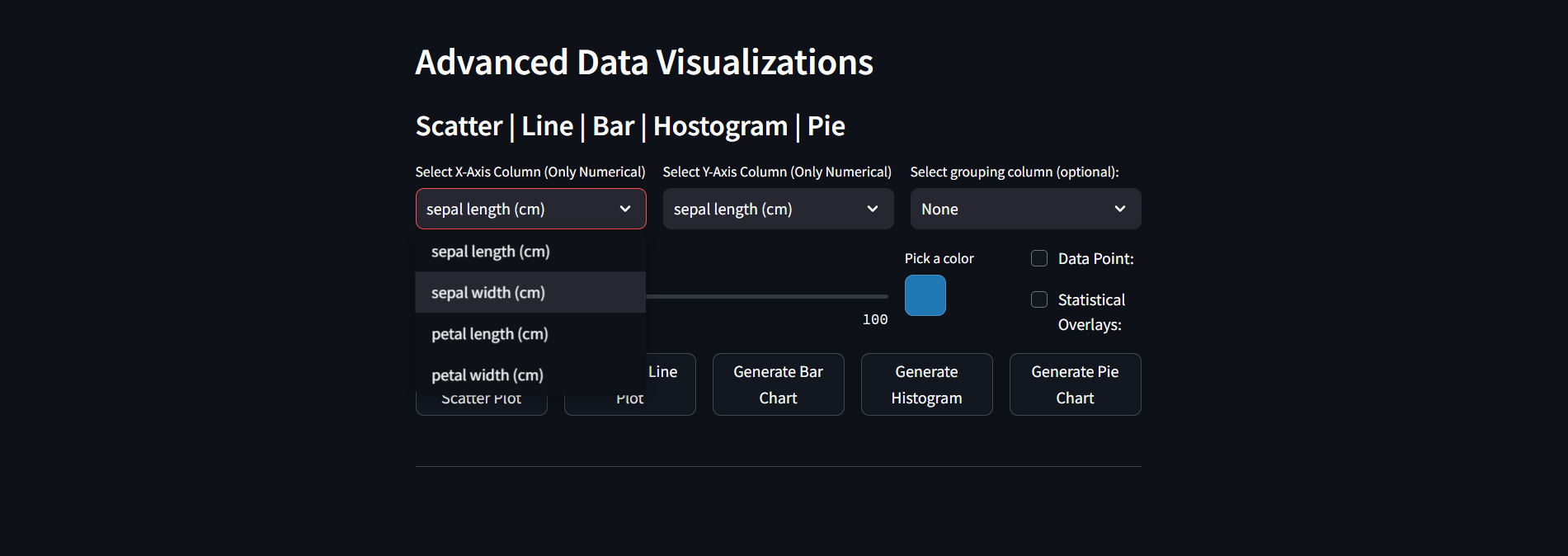




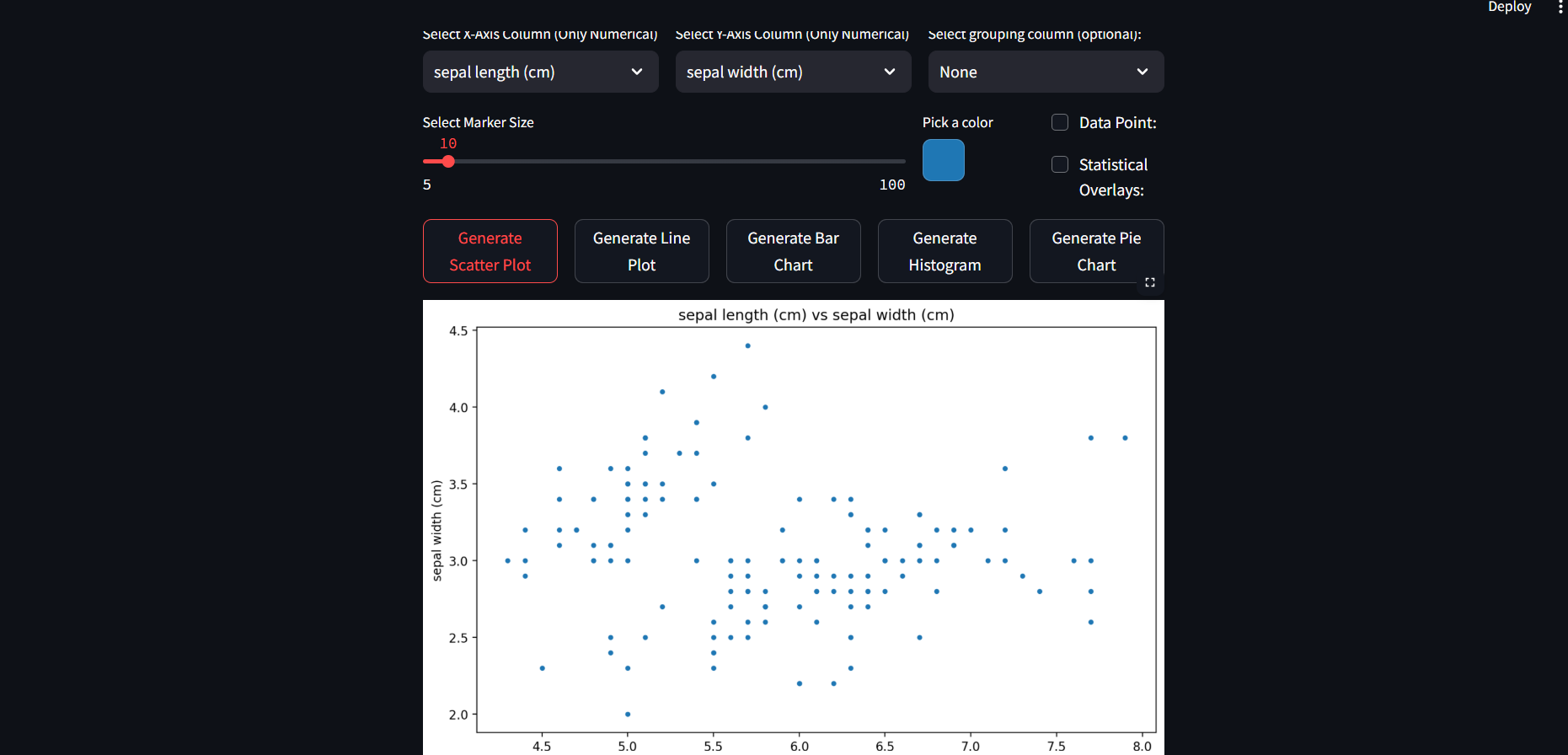
**Different types of chars with data points and statistical overlays**

* Option to select X-axis,
* Option to select Y-axis,
* Grouping
* Marker size
* Color

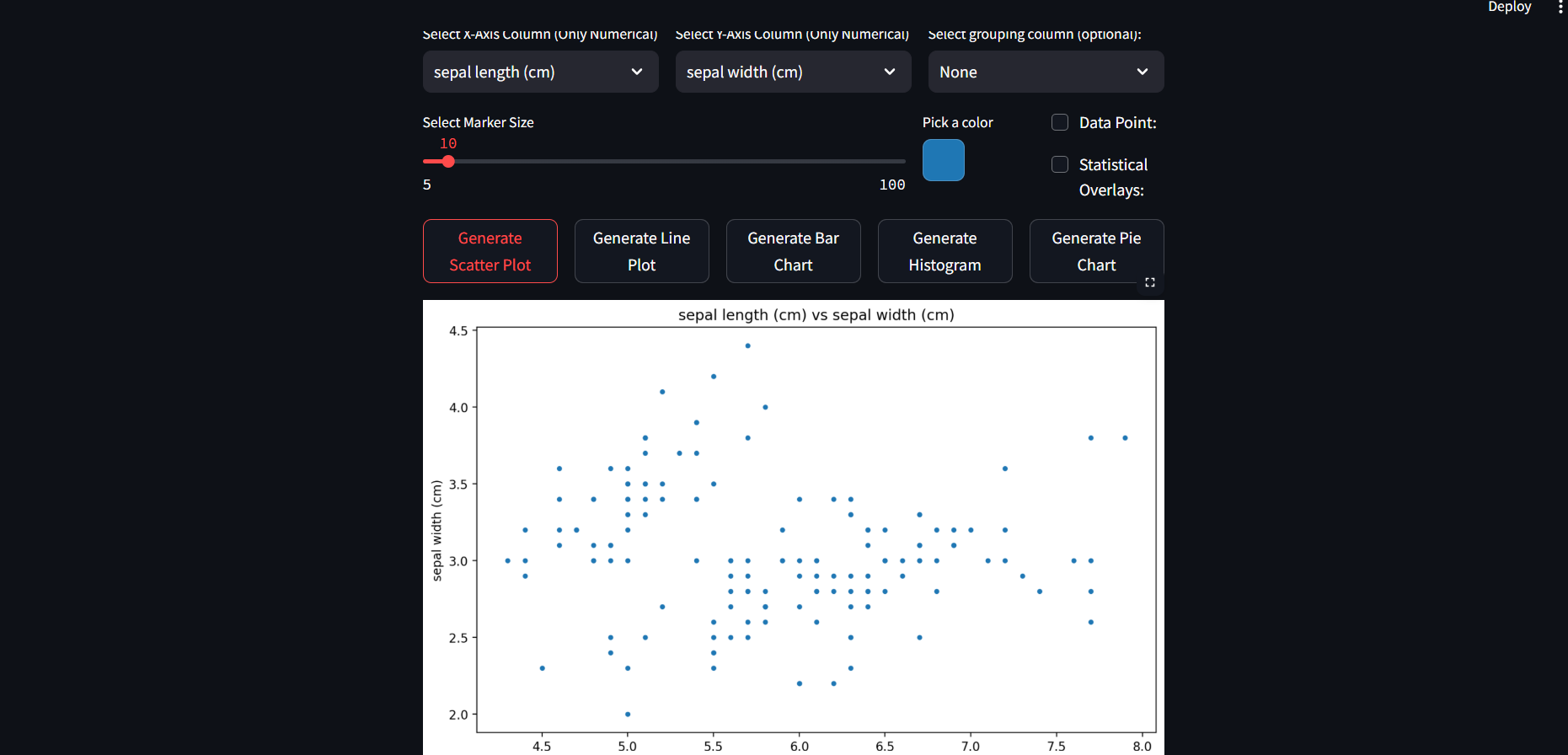
****



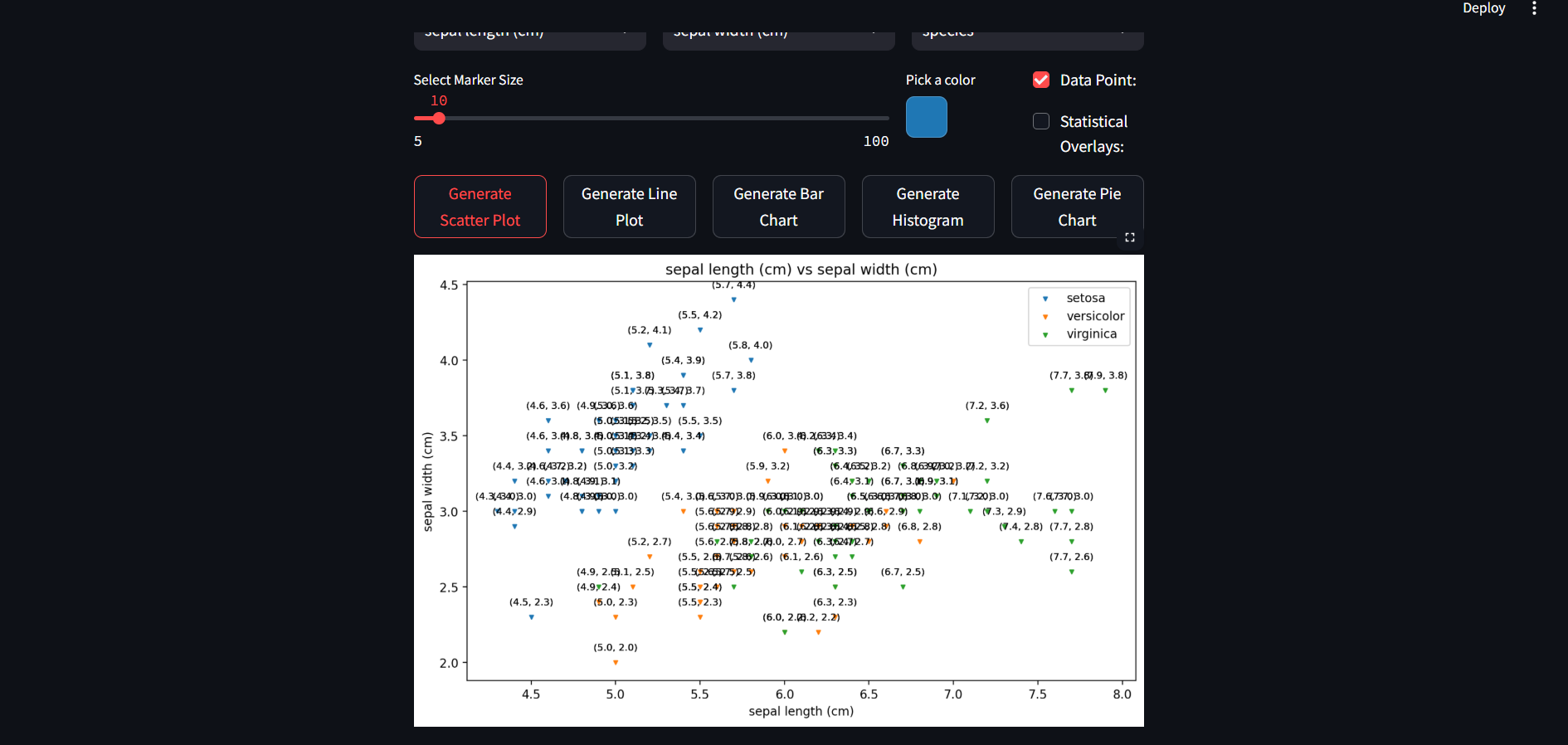
**Without Grouping**



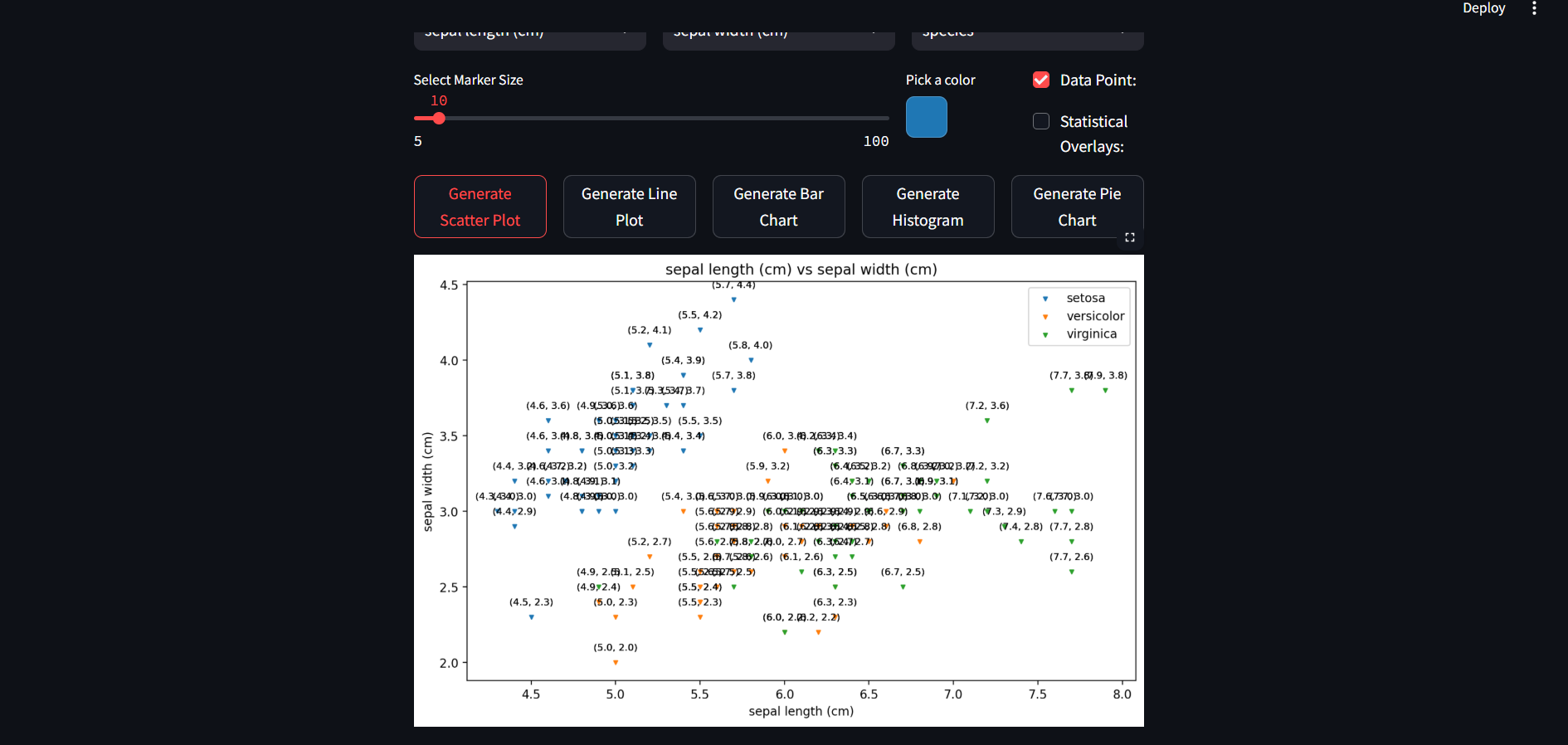
**With Grouping**



**Enable Data Points**

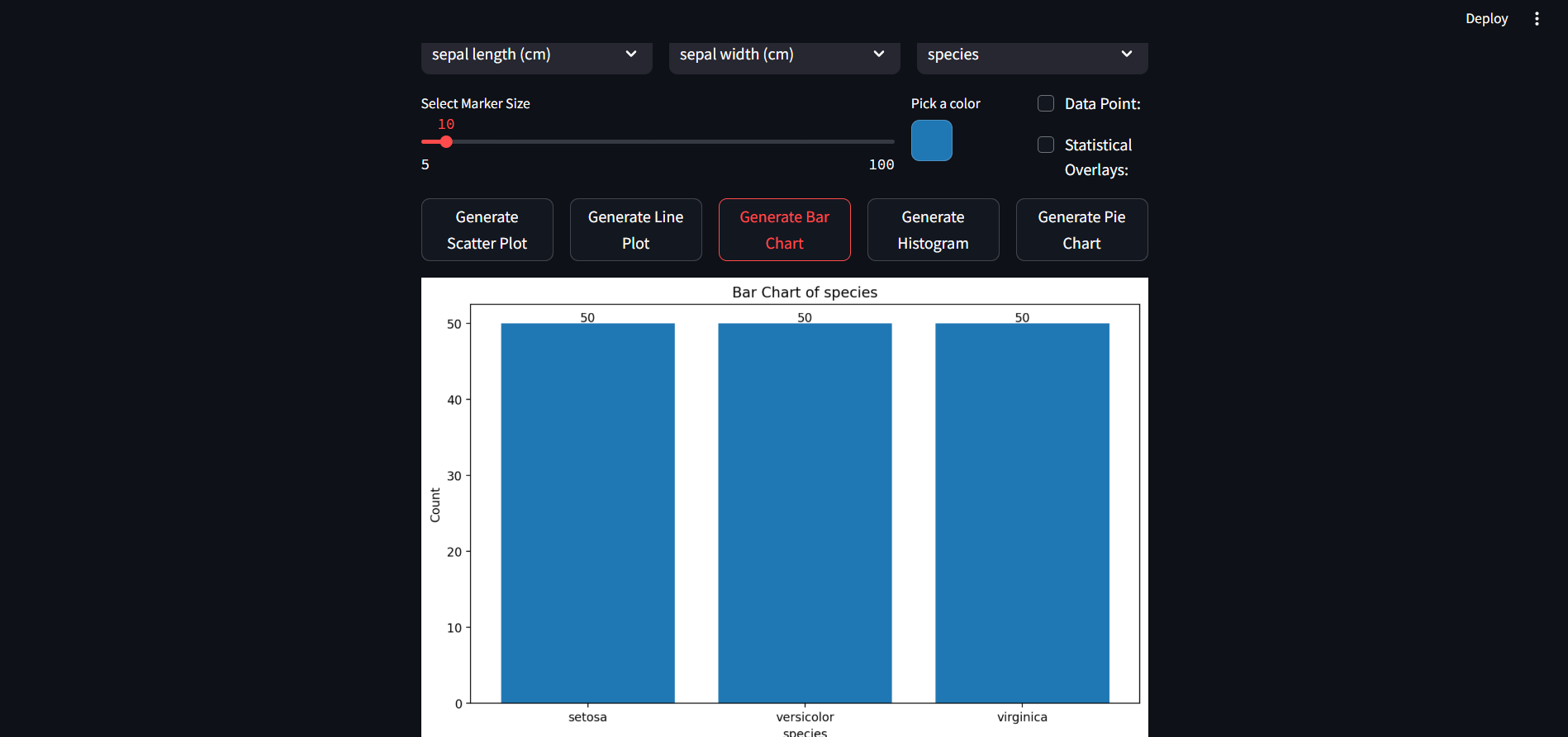


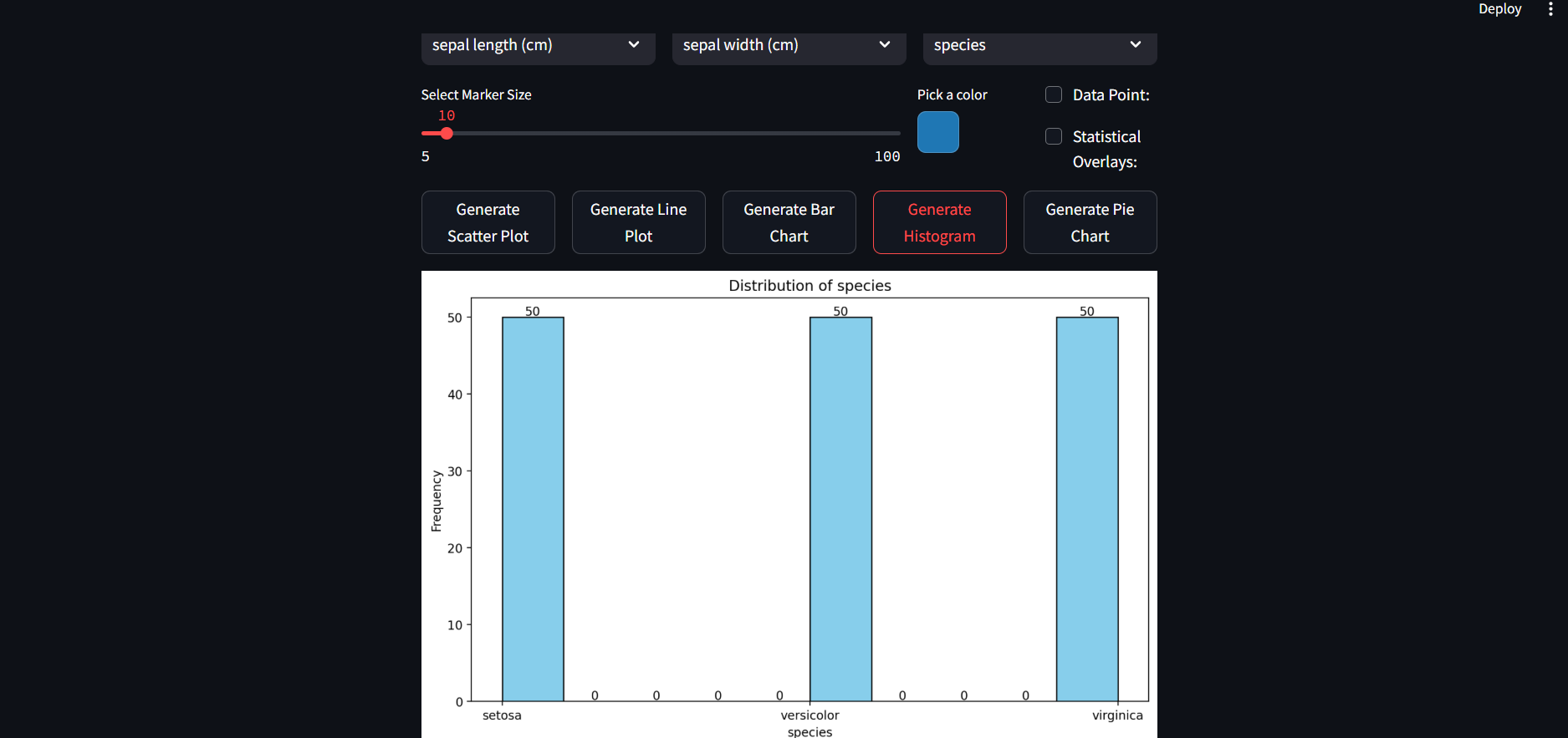
**Enable Statistical Overlays**

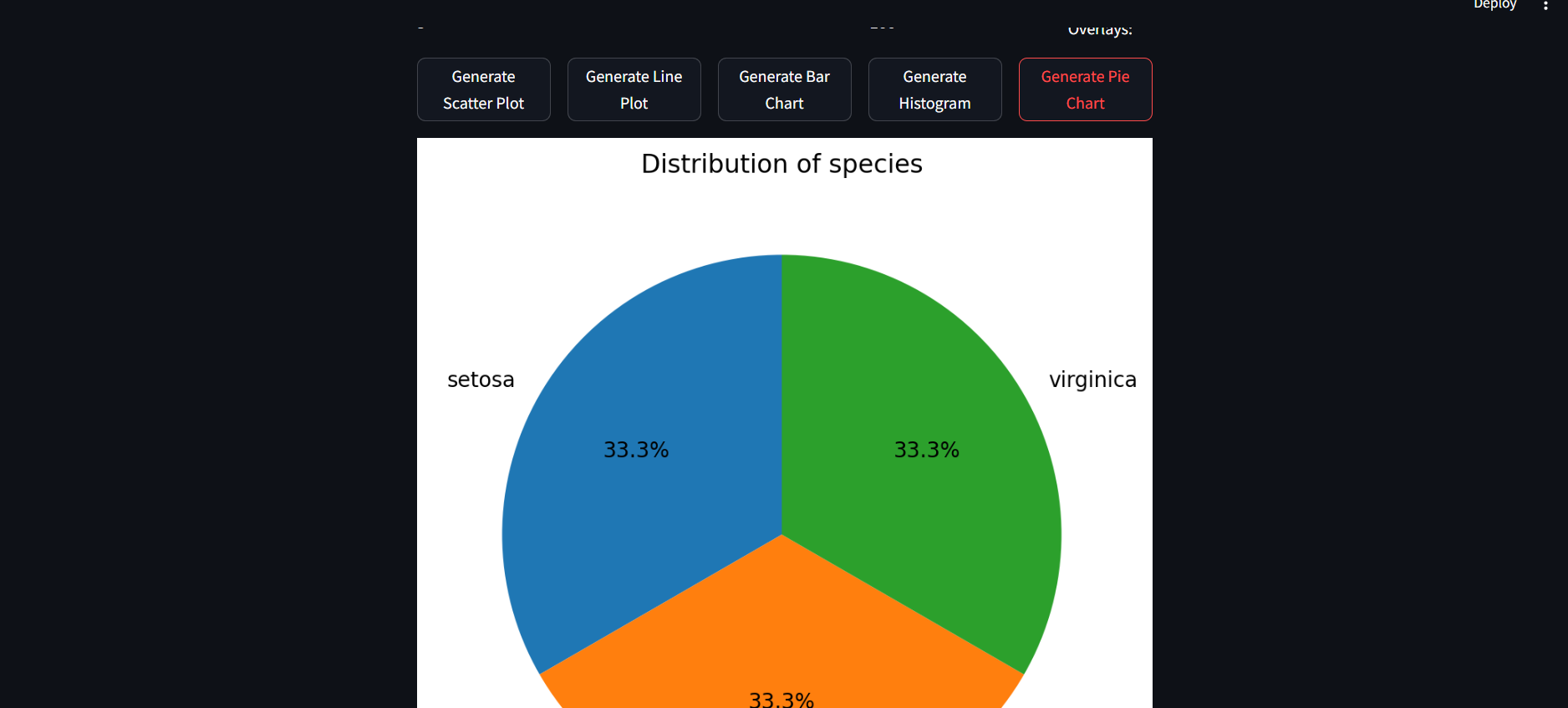
****

**Other Charts and Graphs**

****

****

****

****

1. **Widget/Algorithm Used In Task** 
   1. **Statistical Overlays**
      1. The **Statistical Overlay** option in the scatter and line plot involves adding **regression lines** or trend lines to the plot. It uses **linear regression** to fit a line through the data, helping to visualize trends. This can be enabled using the isStatisticalOverlays checkbox.
   2. **Algorithm Used for Statistical Overlay**
      * The algorithm used for creating the **regression line** is **Linear Regression** with the following steps:
      1. **numpy.polyfit()** is used to perform polynomial fitting (degree 1 for a linear regression)
      2. The function returns the slope and intercept of the regression line.
      3. **np.polyval()** evaluates the polynomial at each point to generate the y-values of the regression line.
   3. **Inputs From Users**
      1. File – csv, xls, xlsx
      2. Choose x-axis, y-aixs
      3. Grouping option
      4. Marker size,
      5. Choose color
      6. Data point flag
      7. Statistical Overlays flag
      8. Buttons to generate graphs and charts