https://github.com/matplotlib/matplotlib/issues/7869

# Issue #7869:

**Take text objects into account in legend auto positioning**

## Potential source of the bug:

The issue seems to arise from the legend.py file in the matplotlib folder. Inside the legend.py file, it seems that there are two methods that are involved in the optimal placement of the legend in the graph. Those methods are _find_best_position and _auto_legend_data.

<u>According to the documentation:</u>

- _find_best_position
    - This is a method that finds the best location to put the legend, taking in the width and height of the legend and its renderer. It makes a call to another function _auto_legend_data to help get the best location for the legend.
- _auto_legend_data
    - This method returns the display coordinates to be used in hit testing in order to assist in finding the best positioning of the legend.

It seems that the text objects are not included in the hit testing in the _auto_legend_data and the issue should be resolved if there is checking for text objects included in that method.

## Impact on other parts of the system:

Implementing a fix to this bug should in no way affect other parts of the system because the fix should only prevent the overlapping of textboxes with the legend. Thus, only the function that checks overlap needs to be changed, which should not disturb any other parts of the system.

## Work estimates:

### Software Design: 5 hours

At the end of this stage, the bug will be analyzed, and the impact of the bug will be understood. The team will have a lead as to approximately where the bug is located and get a general idea of how to approach the fix to the bug. This stage would take about a day. It would take a little longer than the previous phase as some more time would be required to analyze the source code, understand how it functions and then determine an approach to fix the bug.

### Implementing and Unit Testing: 30 hours

At the end of this stage, the bug will be fixed. Some unit tests will also be written to verify that the fixes function as required. This phase will take the longest as the development phase requires the fixing of the code, where further issues in the code need to be accounted for. Some fixes can result in further bugs and more time will be needed to account for those additional fixes. Also, some time will be needed to write the unit test cases.

### Integration and System Testing: 10 hours

Here, further tests will be done to verify that the fixes have not impacted other functionalities of matplotlib. The related existing unit tests will be run to ensure that other legend's functionalities are not affected. Manual testing will also be done to ensure that features that cannot be tested with unit testing are tested. This would be the visual elements such as how the legend looks and its positioning in the graph. Testing will take about a day or two as there are many areas that need to be tested from the unit tests, and the manual testing. If issues were to arise, time will be required to analyze and solve the issues.

### Operation and Maintenance: 1 hours

Once all the previous phases are done, a pull request will be made for the fix and will attempt to close the issue depending if the fix is accepted. It will take about a day or less as in this stage, the task to be done is to make a pull request.

## Justification for bug selection:

This bug has been selected to be worked on because out of all the other bugs, it appears that with this issue, there is a hint of where the bug is. With the additional clues, it seems that more progress can be made on this issue as the clues give us a head start on fixing the bug within a timely manner.

The risks with working on this issue is that there is a possibility the bug is in another location or requires more in-depth knowledge of the structures working together. If this were the case, then it would render our work/time estimates inaccurate and the team would need to reprioritize the development to meet the deadline, which could possibly break the Waterfall process.