Closed-Source Development Deliverable 2

Justin Abrokwah
Yi Shiun Chiu
Spencer McCoubrey
Parth Thakkar

Issue Shortlist

Issue 16482

This issue is a bug related to the *hlines* and *vlines* methods exposed by the Axis and PyPlot interfaces. Essentially, unlike the rest of the API, these methods do not default to the configured rcParams settings for the colour of lines created through this interface. Estimated work will include determining the project's design on how to default to rcParams, properly and consistently implementing the fix, designing intuitive tests for determining acceptance, and making any required changes to documentation in order to support the change to the API. Estimated at 4 hours of work due to investigation time, implementation and subsequent tests due to this being an API breaking change.

Issue 16583

This issue is a bug related to the configuration of an rc parameter configured in the .matplotlibrc file, the *xtick.alignment* field. Another parameter, *horizontalalignment*, consists of the possible values for *xtick.alignment*, but the User gets a ValueError when selecting some of those values. To resolve this issue, we must update the set of possible alignments for the xticks; so we must find where the values are being defined and determine if anything else calls it. After implementing the update, we must test all possible options for the alignment, to validate that the configurations are successfully set. Then, changes to documentation must be made to reflect our changes. Since guidance has been provided by a matplotlib developer, we estimate that this task will only take approximately 2 hours of work: 30 minutes for adding the change appropriately and 1-2 hours for updating/adding relevant tests.

Issue 16580

Blitting is an operation performed in computer graphics, used as an animation tool in Matplotlib. Images are converted into a bitmap allowing animated Artists to be rendered on top of it. So the image that is blitted, becomes a background for the other Artists. This issue resides in the TkAgg backend, where a segmentation fault arises when the user tries to blit a figure that is closed. Thankfully, the person who listed the issue provided code to reproduce the issue, and proposed an idea to solve the problem which we can look into. Also, a Matplotlib developer also used his insight into the project to suggest an even more detailed potential solution in the comments. To resolve this issue, we must implement a fix for the TkAgg backend. Following up with test cases and documentation changes, this shouldn't be a strenuous bug fix. We estimate that this would take about 3 hours to complete fully, as a potential solution has been given in the issue, but the majority of the work will be in recreating the issue and designing an appropriate test suite.

Issue 16666

This issue occurs when the user tries to set the x-axis value to show date time formatting. When setting the x-axis value using a string object such as '2001-14-2020', with the previous x value being set using the datetime object, the user is presented with a conversion error. The issue resides in the *dates.DateConverter.convert* method which does not perform the check for a string to be defined as the x-axis. The original developer who opened the issue has suggested a potential solution which can be a useful starting point, along with some useful comments made by other developers. Estimated work required for this issue will be around 3-4 hours which includes finding the appropriate changes that fixes the issue, writing test cases to ensure the workability of the fix and appropriately documenting as this is a breaking API change.

Issue 16723

This is related to the *twinx* method that is defined in the Pyplot and Axes class. When this function is called, the user is returned a new Axes which has a y-axis on the opposite side of your existing one. In some cases however, the *xlim* (x-axis view limit) property fails to scale properly resulting in an undesired graph. A possible culprit may be the *axhline* function that is known to hard code the *xlim* (as pointed out by one developer). Work to resolve this issue is estimated around five hours: two hours evaluating the conflict between the *twinx* and *axhline* function, two hours implementing a fix for the issue and one hour for writing tests/documentation.

Implemented Issues

Issue 16482

(link to pull request for issue(not official pull request into matplotlib)

This issue is one of the best to implement and test due to the reproducibility of the task and the clear scope of effect of this task. The submitted issue has a clear set of requirements in order to fix, as well as clear acceptance criteria which will make designing tests simpler. When investigating, reproducing the issue was possible in several environments, including Windows, MacOSX and Jupyter Notebooks; this implies the issue is well-founded and consistent. As well, the scope of this task means that we will not have to worry too much about possible side effects of a change, such as unintended effects on plotting or within other exposed portions of the API. Work to implement the fix will include 1-2 hours for proper investigation into the correct way to approach the problem, and 2-3 hours to design adequate tests that properly satisfy the acceptance criteria. The risk involved here is that as this is a public facing API that is most likely used often, our changes will need to be backwards compatible in order to not produce unintended errors for current users or appropriately updated documentation to communicate the change going forward. Relying on the existing test suite for regression and possibly adding new tests to retroactively test existing functionality will be high priority.

Technical Commentary

Our changes will address the issue linked where in the configured '*lines.color*' parameter of *rcParams* was not being used as the default parameter for the *vlines*/*hlines* methods of both the PyPlot and Axes API. The changes fix the reproducible issue provided, while being as backwards-compatible with existing users as possible by defaulting to the previously used value. The overall design of the project is not changed due to the minimal scope of the issue, the only design changed is that the API changes were made to be more consistently inline with the rest of the project's exposed API.

Files Modified

lib/matplotlib/axes/_axes.py: this file contains the source code for the *vlines* and *hlines* methods for the Axes class and as such was changed to address the bug.
lib/matplotlib/pyplot.py: this file contains the source code for the *vlines* and *hlines* methods for the PyPlot interface and as such was changed to address the bug.
lib/matplotlib/tests/test_axes.py: this file contains the test suite for the Axes API. Additions to the respective test functions for *vlines* and *hlines* were made in line with the bug fixes.
lib/matplotlib/tests/test_pyplot.py: this file contains the test suite for the PyPlot interface. 2 new test functions for *vlines* and *hlines* were added in line with the bug fixes.

doc/api/next_api_changes/behaviour.rst: this file was edited in line with the contributing guidelines, wherein if a public facing API is changed it must be documented here.

Files Added

lib/matplotlib/tests/baseline_images/test_axes/hlines_rc_params.png: file added as a part of the matplotlib image comparator test suite.
lib/matplotlib/tests/baseline_images/test_axes/vlines_rc_params.png: file added as a part of the matplotlib image comparator test suite.
lib/matplotlib/tests/baseline_images/test_pyplot/hlines_rc_params.png: file added as a part of the matplotlib image comparator test suite.
lib/matplotlib/tests/baseline_images/test_pyplot/vlines_rc_params.png: file added as a part of the matplotlib image comparator test suite.

Test Suite

Test Name	Acceptance Criteria	Test Designed
test_axes.py::test_vlines	Configuring *lines.color* within a script and rendering a figure through the Axes interface, vertical lines rendered should default to configured value. When unconfigured, lines default back to '*k'*.	Test is configured with rcparams set to *'blue'*, and a figure is created through the Axes interface using *vlines* with no *colors* value set and *hlines*. Resulting plot is matched against **vlines_rc_params.png**.
test_axes.py::test_hlines	Configuring *lines.color* within a script and rendering a figure through the Axes interface, horizontal lines rendered should default to configured value. When unconfigured, lines default back to '*k'*.	Test is configured with rcparams set to *'blue'*, and a figure is created through the Axes interface using *vlines* and *hlines* with no *colors* value set. Resulting plot is matched against **hlines_rc_params.png**.
test_pyplot.py::test_vlines	Configuring *lines.color* within a script and rendering a figure through the PyPlot interface, vertical lines rendered should default to configured value. When unconfigured, lines default back to '*k'*.	Test is configured with rcparams set to *'blue'*, and a figure is created through the PyPlot interface using *vlines* with no *colors* value set and *hlines*. Resulting plot is matched against **vlines_rc_params.png**.

test_pyplot.py::test_hlines	Configuring *lines.color* within a script and rendering a figure through the PyPlot interface, horizontal lines rendered should default to configured value. When unconfigured, lines default back to '*k'*.	Test is configured with rcparams set to *'blue'*, and a figure is created through the PyPlot interface using *vlines* and *hlines* with no *colors* value set. Resulting plot is matched against **hlines_rc_params.png**.

Issue 16580

(link to pull request for issue(not official pull request into matplotlib)

Technical Commentary

The changes we implemented address the issue where the TkAgg backend was propagating a segmentation fault to the user. Our alterations mend this problem by performing a check on the referenced figure before calling the *blit*() function on it. Due to this issue being very minor in the big picture of the project, our changes don't affect the overall Matplotlib experience. Instead, it provides a more effective message to the user upon processing invalid actions.

Files Modified

lib/matplotlib/backends/_backend_tk.py: this file contains the source code for the *blit* function for the TkAgg backend implementation.
lib/matplotlib/tests/test_backend_tk.py: this file contains the test suite for the TkAgg backend implementation. Created a new test case to address fixes made.

Files Added

N/A

Test Suite

Test Name	Acceptance Criteria	Test Designed
test_backend_tk::test_blit_error_on_closed_figure	Within a TkAgg environment, closing a figure before calling the *blit*() function on it should propagate a RuntimeError to the caller with appropriate context.	Test is configured to force the TkAgg backend implementation to create a figure, then close it. Subsequently, the test calls the *blit*() function on this figure, expecting a RuntimeError including the information regarding the closed figure.