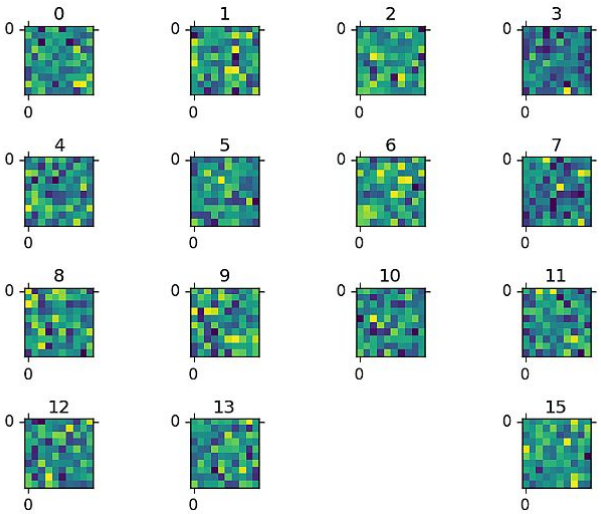
Deliverable 3 - Issue List (step 1)

Promising bugs to examine

ISSUE 1: missing imshow() subplots when using tight_layout() #4976
Link: https://github.com/matplotlib/matplotlib/issues/4976
Where the bug is in the code / where one would start investigating:

- Bug appears when making grids of subplots with **imshow** and **tight_layout** the second to last plot is missing in the output image.
- appears that **plt.tight_layout()** might be removing an axes from the fig.axes list.
- should try looking into **interactions** between **plt.tight_layout** and **plt.subplot()**.



ISSUE 2: 'bottom cannot be >= top' when using `tight_layout` #5456
Link: https://github.com/matplotlib/matplotlib/issues/5456
Where the bug is in the code / where one would start investigating:

- The bug happens when using tight_layout after setting a y-label to the axis to be `ax.set_ylabel('a'*50000),` which gives the user a ValueError
- One should start investigating tight_layout's code
  - look into the code that validates / checks the pad (specifically `tight_layout` that the pad can not be negative)
  - apparently there needs to be a check that the pad cannot be > .5 with the check above

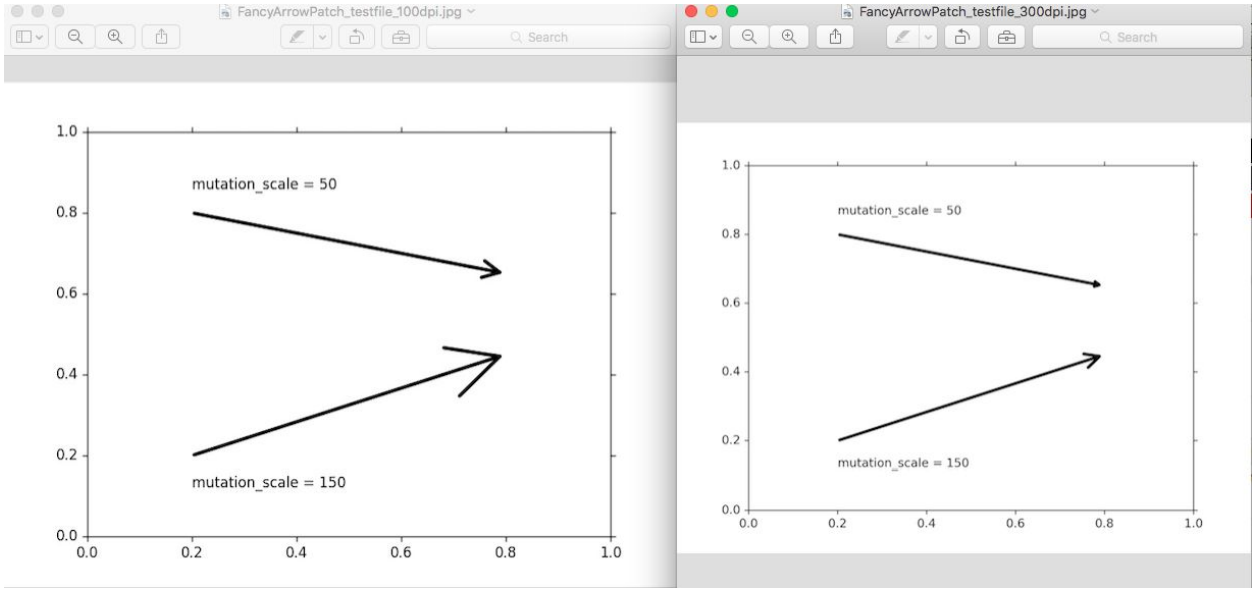
```
C:\Users\Crimson\Anaconda2\lib\site-packages\matplotlib-0+unknown-py2.7-win-amd64.egg\matplotlib\tight_layout.py:222: UserWarning: tight_layout : falling back to Agg renderer
  warnings.warn("tight_layout : falling back to Agg renderer")
Traceback (most recent call last):
  File "tightlayout.py", line 7, in <module>
    plt.tight_layout()
  File "C:\Users\Crimson\Anaconda2\lib\site-packages\matplotlib-0+unknown-py2.7-win-amd64.egg\matplotlib\pyplot.py", line 1284, in tight_layout
    fig.tight_layout(pad=pad, h_pad=h_pad, w_pad=w_pad, rect=rect)
  File "C:\Users\Crimson\Anaconda2\lib\site-packages\matplotlib-0+unknown-py2.7-win-amd64.egg\matplotlib\figure.py", line 1889, in tight_layout
    self.subplots_adjust(**kwargs)
  File "C:\Users\Crimson\Anaconda2\lib\site-packages\matplotlib-0+unknown-py2.7-win-amd64.egg\matplotlib\figure.py", line 1745, in subplots_adjust
    self.subplotpars.update(*args, **kwargs)
  File "C:\Users\Crimson\Anaconda2\lib\site-packages\matplotlib-0+unknown-py2.7-win-amd64.egg\matplotlib\figure.py", line 231, in update
    raise ValueError('bottom cannot be >= top')
ValueError: bottom cannot be >= top
```

ISSUE 3: head size of FancyArrowPatch changes between interactive figure and picture export #6035

https://github.com/matplotlib/matplotlib/issues/6035

Where the bug is in the code / where one would start investigating:

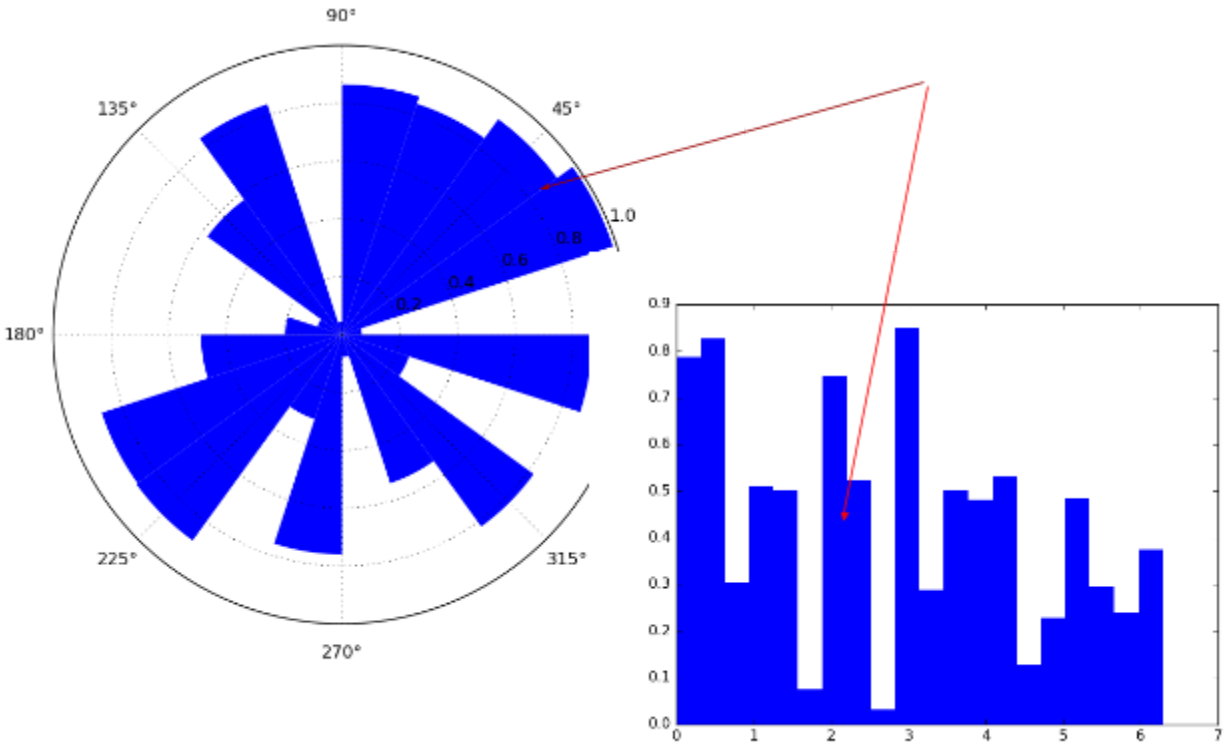
- the arrows are different in the output files generated from the example code
- Possible API issue and something is violating this **rule** “changing the output dpi or the file format should not change the relative dimensions of plot elements”
- Possible reasoning: dpi not being recalculated to reflect actual renderer dpi



ISSUE 4: bars plotted on polar axes still have visible boundaries when linewidth set to 0
https://github.com/matplotlib/matplotlib/issues/3873

Where the bug is in the code / where one would start investigating:

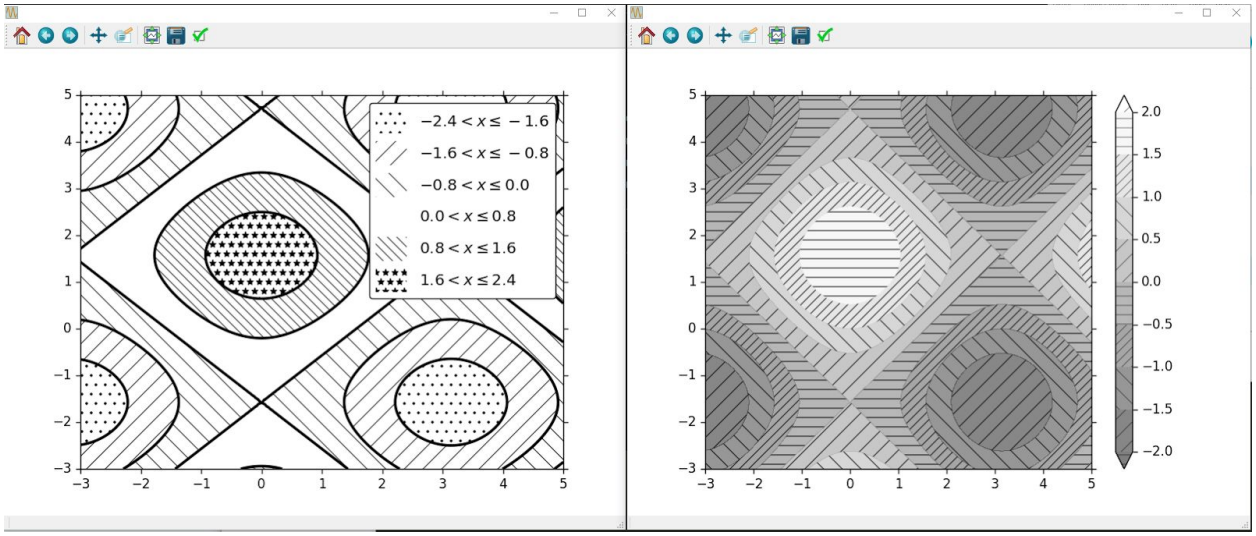
- The bug (visible line boundaries) happens when have this line of code
- ax = plt.subplot(111, polar=True)
- ax.bar(...,linewidth=0,...)
- Expected result : the polar system should not see the line width when it set to 0
- Actual result : Be able to see the width for polar system even it set to 0 while cartesian system does not show the width.

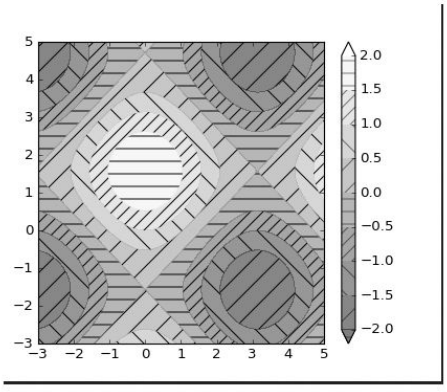


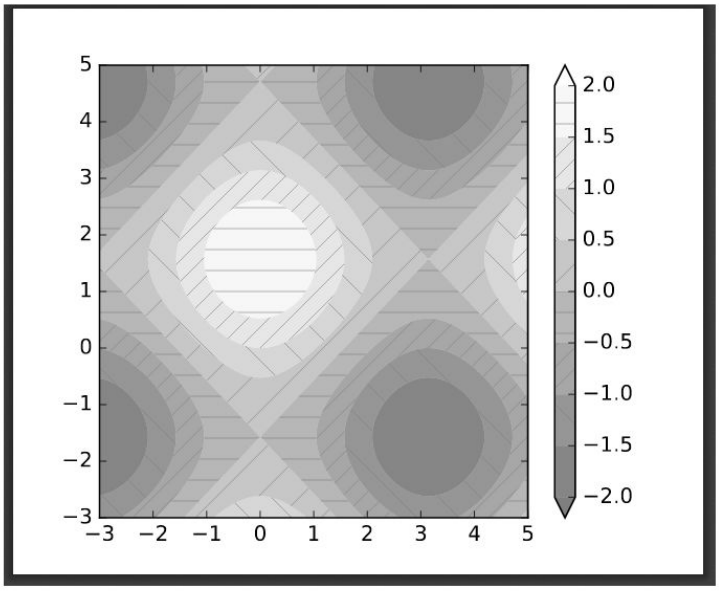
ISSUE 5: contourf hatching and saving to pdf
https://github.com/matplotlib/matplotlib/issues/3023

Where the bug is in the code / where one would start investigating:

- Left graph(png format): Color = darker, Right graph(pdf): color = gray
- contourf do not appear when saving to pdf
- should try looking at **plt.contourf(...)** and investigating what is happening when color is set to none.







ISSUE 6: Specifying histtype='stepfilled' and normed=True when using plt.hist causes ymax to be set incorrectly #4414

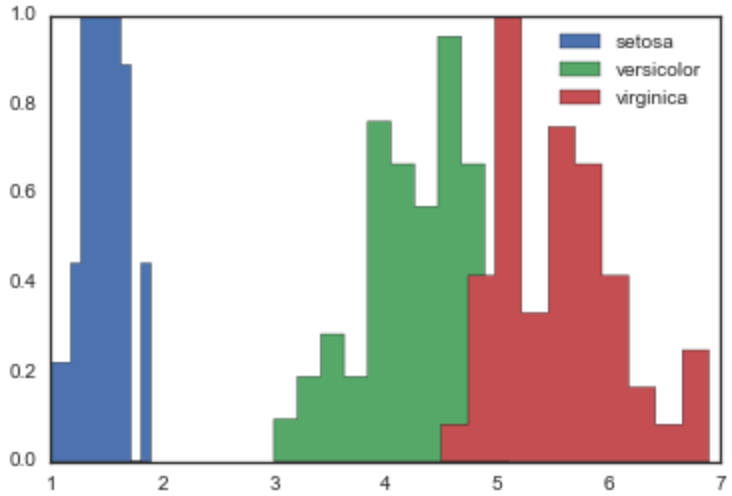
Link: https://github.com/matplotlib/matplotlib/issues/4414

Where the bug is in the code / where one would start investigating:

- Bug appears when creating a hist() instance with values histtype='stepfilled' and normed=True passed in. This causes the Figure's y-axis values to be incorrect.
- appears that somewhere hist(), the y-limits are set from calculations when this happens. The y-limits shouldn't be manually calculated but autoscaled when normed = True.

Look into the code that is used to create hist() instances. Specifically, look into how the different parameters interact with each other and how the y-limit is chosen.

Picture of the bug: The y-axis max value is 1.0 and is cutting off bars that go beyond it.



Selection of the bugs to implement and test:

- **Issue 1:** missing imshow() subplots when using tight_layout() #4976
- **Issue 2:** 'bottom cannot be >= top' when using `tight_layout` #5456

Work unit: Each day of work is usually 2-3 hours of teamwork between the allocated members

Issue 1

Estimated Time:

- 3 days to resolve bug
- 2 days to test

Risk:

- Potential to break or add bugs to pcolor() because using pcolor() to generate subplots with tight_layout() works currently
- Can cause tight_layout() to remove even more axes in fig.axes list (worsening the bug)
- Can cause the numbers on the axes to overlap each other, thus not readable

Allocated Members: Kai, Kevin, Richard

Issue 2

Estimated Time:

- 3 days to resolve bug
- 2 days to test

Risk:

- Breaking code in other places that use tight_layout()
- Potential to break other areas of code which uses padding of greater than 0.5
- Cause negative paddings to work in tight_layout()'s code (should be positive)
- Breaking the padding system all together

Allocated Members: Chun, Geoffrey

Reasoning for our choices: We choose issue 1 and 2 to fix first because they are relatively simpler to tinker with which gives our group a good boost to start off the project. In our opinion, issue 3 through 5 are more intricate and should be handled with more care and experience with matplotlib, while issue 6 is rather simplistic. In the event that our team completes issue 1 and 2 early, we will tackle issue 3, 4, 5 and 6. We have decided to allocate 3 and 2 team members to issue 1 and 2 respectively. This is because the team agreed that issue 1 is potentially more difficult than issue 2 and thus require more work. We will attempt to fix issue 6 (bug#4414) next after we have finish issue 1 and 2.

Planning and Software Development Process

We will use agile software development process for our bug fixes. Within the sprint, we will have taskboard that shows amount of work that is been done and work that's in progress.
We will conduct verification and validation for our bug fix at the end of sprint to make sure our implementation fully worked. We will use burndown chart to estimate time that we used for fixing the bugs and keep track of our progression. Also conduct quick code reviews of each other code to ensure we are following proper coding standards and style guides.