# CSCD01 Team 6 Deliverable 3

## Bug 1: Using Minute Locator to set x-axis ticks exceeds Locator.MAXTICKS

**The problem**: When plotting data which has to do with time, using the minute locator module to set the number of ticks over 1000 results in a runtime error. The variable MAXTICKS in function locator sets an artificial limit of 1000 ticks, when what is wanted is that your system should determine the maximum number of ticks, and if there are too many ticks it should run out of memory, just like it does for large images.

```
greens14@cms-vm27:~/team06-Project/d3_bug_tests$ python bug_test_3-broken.py
Traceback (most recent call last):
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/backends/backend_qt5agg.py", line 199, in __draw_idle_agg
    FigureCanvasAgg.draw(self)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/backends/backend_agg.py", line 467, in draw
    self.figure.draw(self.renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/figure.py", line 1248, in draw
    renderer, self, artists, self.suppressComposite)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/image.py", line 139, in _draw_list_compositing_images
    a.draw(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axes/_base.py", line 2386, in draw
    mimage._draw_list_compositing_images(renderer, self, artists)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/image.py", line 139, in _draw_list_compositing_images
    a.draw(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 1113, in draw
    ticks_to_draw = self._update_ticks(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 954, in _update_ticks
    tick_tups = list(self.iter_ticks())
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 897, in iter_ticks
    majorLocs = self.major.locator()
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/dates.py", line 802, in __call__
    return self.tick_values(dmin, dmax)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/dates.py", line 824, in tick_values
    return self.raise_if_exceeds(date2num(dates))
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/ticker.py", line 1339, in raise_if_exceeds
    raise RuntimeError(msg)
RuntimeError: Locator attempting to generate 1000 ticks from 736374.96875 to 736375.6625: exceeds Locator.MAXTICKS
```

*Figure 1: Error code when setting x-axis ticks over 1000*

**How to fix this:**

First we removed the limiting variable MAXTICKS in the file ticker.py, which was set to 1000, which caused the runtime error.

```
1386  class Locator(TickHelper):
1387      """
1388      Determine the tick locations;
1389
1390      Note, you should not use the same locator between different
1391      :class:`~matplotlib.axis.Axis` because the locator stores references to
1392      the Axis data and view limits
1393      """
1394
1395      # Some automatic tick locators can generate so many ticks they
1396      # kill the machine when you try and render them.
1397      # This parameter is set to cause locators to raise an error if too
1398      # many ticks are generated.
1399      MAXTICKS = 1000
1400
```

*Figure 2: Removed the MAXTICKS variable from class Locator in ticker.py (https://github.com/matplotlib/matplotlib/blob/master/lib/matplotlib/ticker.py)*

Then in the function raise_if_exceeds, we removed the code which checked if MAXTICKS had been exceeded.

```
1432      def raise_if_exceeds(self, locs):
1433          """raise a RuntimeError if Locator attempts to create more than
1434             MAXTICKS locs"""
1435          if len(locs) >= self.MAXTICKS:
1436              msg = ('Locator attempting to generate %d ticks from %s to %s: ' +
1437                     'exceeds Locator.MAXTICKS') % (len(locs), locs[0], locs[-1])
1438              raise RuntimeError(msg)
1439
1440          return locs
```

*Figure 3: Removed the if statement (https://github.com/matplotlib/matplotlib/blob/master/lib/matplotlib/ticker.py)*

We replaced the code in raise_if_exceeds with code that just return the locs argument so that any function which calls raise_if_exceeds still works.

```
1426        def raise_if_exceeds(self, locs):
1427            """if something trys to call this function just return the locs"""
1428
1429            return locs
```

*Figure 4: raise_if_exceeds now just returns the locs argument to ensure no other functions crash (https://github.com/CSCD01-Winter2017/team06-Project/blob/master/matplotlib/lib/matplotlib/ticker.py)*

Finally we removed any unnecessary calls to the function raise_if_exceeds from the code.

**Tests:**

We created a file called bug_test_3-broken.py which demonstrates what the error was and how it has been fixed.

```
1  #should not set artificial MAXTICKS limit
2
3  import matplotlib as mpl
4  import matplotlib.pyplot as plt
5  import datetime
6
7  num_list=[]
8  date_list=[]
9
10 #Create List of Datetime Objects to Plot on X Axis
11 start = datetime.datetime(2017 , 2, 15)
12 end = datetime.datetime(2017, 2, 15, 15, 10, 0)
13
14 step=datetime.timedelta(minutes=1)
15
16 while start < end:
17     date_list.append(start)
18     start+=step
19
20 #Create List of Numbers to Plot on Y Axis
21 num=0
22 lastnum=len(date_list)
23
24 while num < lastnum:
25     num_list.append(num)
26     num+=1
27
28 locator = mpl.dates.MinuteLocator(interval=1)
29
30 fig, (ax1,ax2) = plt.subplots(2)
31
32 ax1.plot_date(date_list,num_list,'b.')
33 ax2.plot_date(date_list,num_list,'r.')
34
35 plt.gca().xaxis.set_major_locator(locator)
36
37 plt.show()
```

*Figure 5: Code to test the fix of this bug (should output two line graphs, which combined contain over 1000 ticks)*

Running this before would have resulted in:

```
greens14@cmsvm27:~/team06-Project/d3_bug_tests$ python bug_test_3-broken.py
Traceback (most recent call last):
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/backends/backend_qt5agg.py", line 199, in __draw_idle_agg
    FigureCanvasAgg.draw(self)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/backends/backend_agg.py", line 467, in draw
    self.figure.draw(self.renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/figure.py", line 1248, in draw
    renderer, self, artists, self.suppressComposite)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/image.py", line 139, in _draw_list_compositing_images
    a.draw(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axes/_base.py", line 2386, in draw
    mimage._draw_list_compositing_images(renderer, self, artists)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/image.py", line 139, in _draw_list_compositing_images
    a.draw(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/artist.py", line 68, in draw_wrapper
    return draw(artist, renderer, *args, **kwargs)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 1113, in draw
    ticks_to_draw = self._update_ticks(renderer)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 954, in _update_ticks
    tick_tups = list(self.iter_ticks())
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/axis.py", line 897, in iter_ticks
    majorLocs = self.major.locator()
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/dates.py", line 802, in __call__
    return self.tick_values(dmin, dmax)
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/dates.py", line 824, in tick_values
    return self.raise_if_exceeds(date2num(dates))
  File "/home/greens14/team06-Project/matplotlib/lib/matplotlib/ticker.py", line 1339, in raise_if_exceeds
    raise RuntimeError(msg)
RuntimeError: Locator attempting to generate 1000 ticks from 736374.96875 to 736375.6625: exceeds Locator.MAXTICKS
```

*Figure 6: Error code from running code before fixing bug*

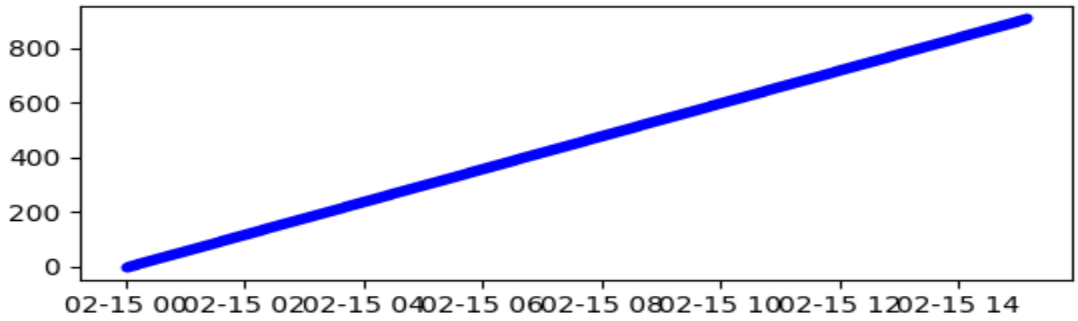


*Figure 7: Output before bug fix, notice how there were too many ticks so the second graph was not drawn*

Now running this results in:

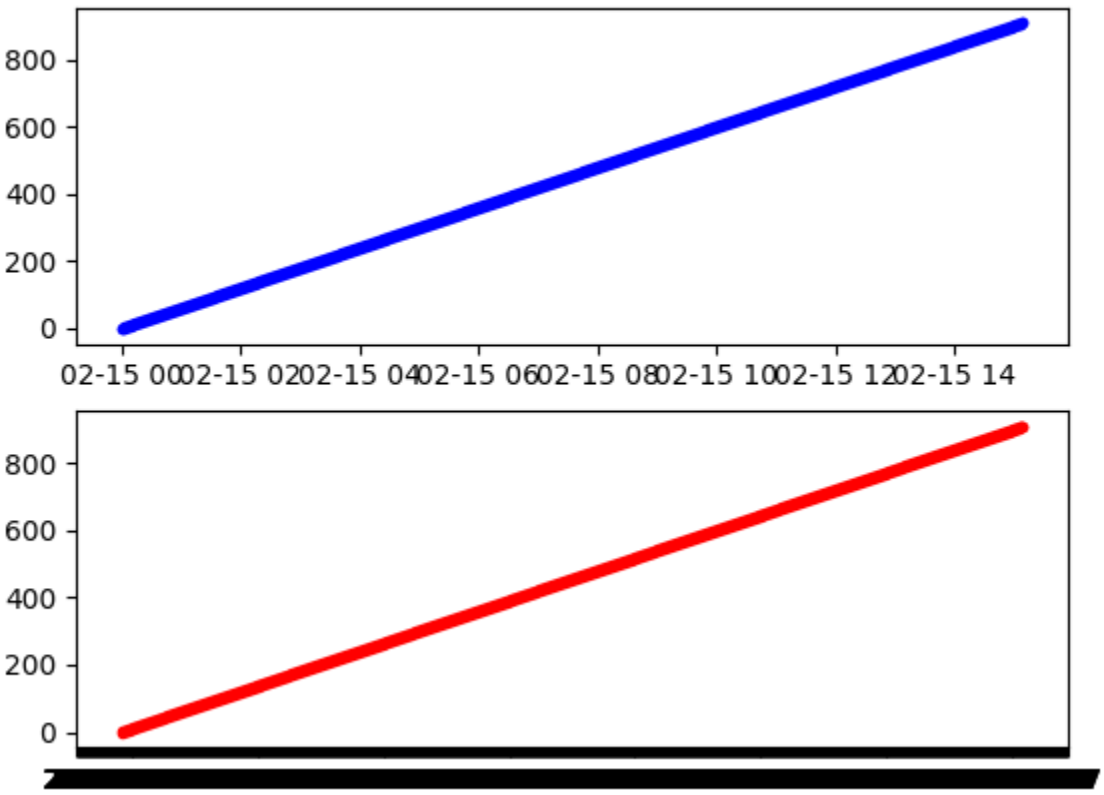


*Figure 8: Output after bug fix*

**Bug 2: figure.Figure.autofmt_xdate applied to major xtick labels only**

The problem: The function autofmt_xdate in figure.py would only rotate major ticks on the x-axis, when the user wanted the ability to choose which labels they wanted to rotated, either the major, minor or both.

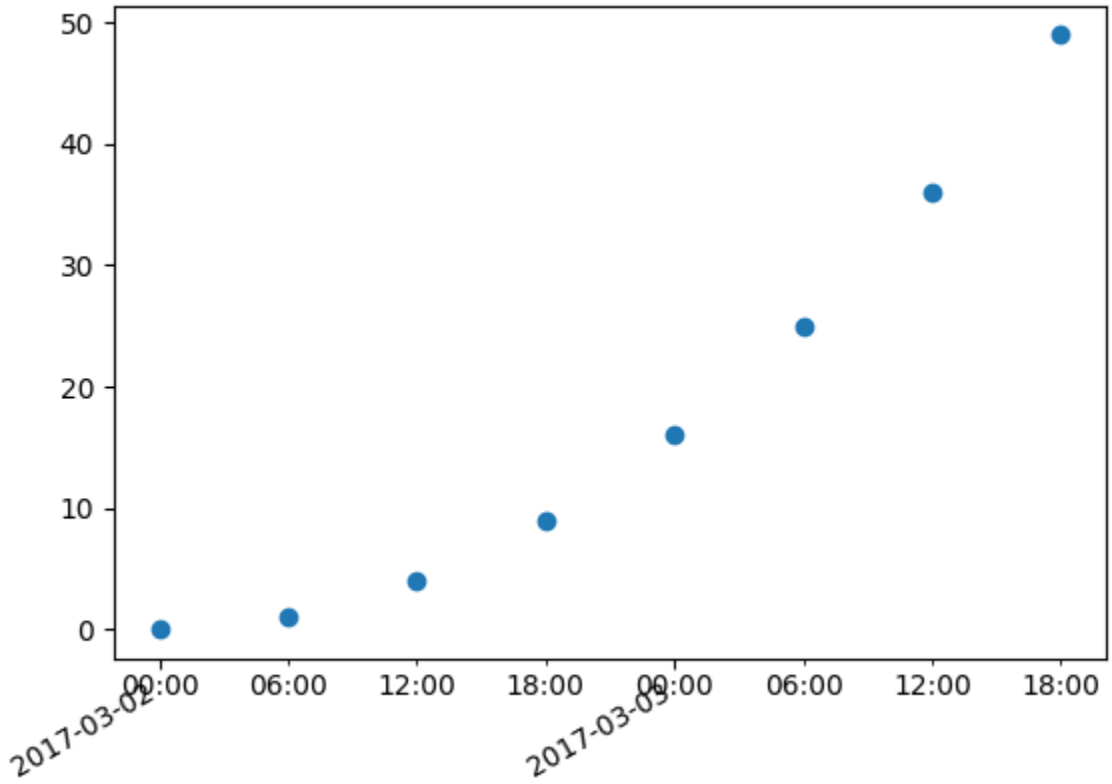


*Figure 9: Only the major ticks on the x-axis are rotated resulting in overlapping*

**How to fix this:**

First we added an argument to autofmt_xdate which would allow to user to specify which ticks they wanted to be rotated

```
442     def autofmt_xdate(self, bottom=0.2, rotation=30, ha='right', which='major'):
443         #which is passed in initially as major but the user can change it to
444         #both if wanted.
445         """
446         Date ticklabels often overlap, so it is useful to rotate them
447         and right align them.  Also, a common use case is a number of
448         subplots with shared xaxes where the x-axis is date data.  The
449         ticklabels are often long, and it helps to rotate them on the
450         bottom subplot and turn them off on other subplots, as well as
451         turn off xlabels.
452
453         *bottom*
454             The bottom of the subplots for :meth:`subplots_adjust`
455
456         *rotation*
457             The rotation of the xtick labels
458
459         *ha*
460             The horizontal alignment of the xticklabels
461
462         *which*
463             The labels to which autofmt_xdate will be applied
464         """
```

*Figure 10: Added new argument which, initially set to major but user can set to both if he wants both ticks to be rotated (https://github.com/CSCD01-Winter2017/team06-Project/blob/master/matplotlib/lib/matplotlib/figure.py)*

Then we passed the which argument to every call to the function get_xticklabels so that the correct ticks would be rotated

```
465         allsubplots = all(hasattr(ax, 'is_last_row') for ax in self.axes)
466         if len(self.axes) == 1:
467             #added the argument which to all calls of get_xticklabels
468             for label in self.axes[0].get_xticklabels(which=which):
469                 label.set_ha(ha)
470                 label.set_rotation(rotation)
471         else:
472             if allsubplots:
473                 for ax in self.get_axes():
474                     if ax.is_last_row():
475                         for label in ax.get_xticklabels(which=which):
476                             label.set_ha(ha)
477                             label.set_rotation(rotation)
478                     else:
479                         for label in ax.get_xticklabels(which=which):
480                             label.set_visible(False)
481                         ax.set_xlabel('')
482
```

*Figure 11: Added argument to each call of get_xticklabels*

**Tests:**

We created files called bug_test_4.py and bug_test_4-fixed.py which demonstrates the output before and after the bug is fixed.

```
#minor ticks are not rotated when formatted
#want them to be formatted along with the major ticks


import datetime
import matplotlib.pyplot as plt
from matplotlib.dates import DayLocator, HourLocator, DateFormatter, drange
from numpy import arange

date1 = datetime.datetime(2017, 3, 2)
date2 = datetime.datetime(2017, 3, 4)
delta = datetime.timedelta(hours=6)
dates = drange(date1, date2, delta)

y = arange(len(dates)*1.0)

fig, ax = plt.subplots()
ax.plot_date(dates, y*y)

ax.xaxis.set_major_locator(DayLocator())
ax.xaxis.set_minor_locator(HourLocator(arange(0, 25, 6)))
ax.xaxis.set_major_formatter(DateFormatter('%Y-%m-%d'))
ax.xaxis.set_minor_formatter(DateFormatter('%H:%M'))

ax.fmt_xdata = DateFormatter('%Y-%m-%d %H:%M:%S')
fig.autofmt_xdate()

plt.show()
```

*Figure 12: Code to demonstrate before bug fix (https://github.com/CSCD01-Winter2017/team06-Project/blob/master/d3_bug_tests/bug_test_4.py)*

Before the bug was fixed the output would look like this:

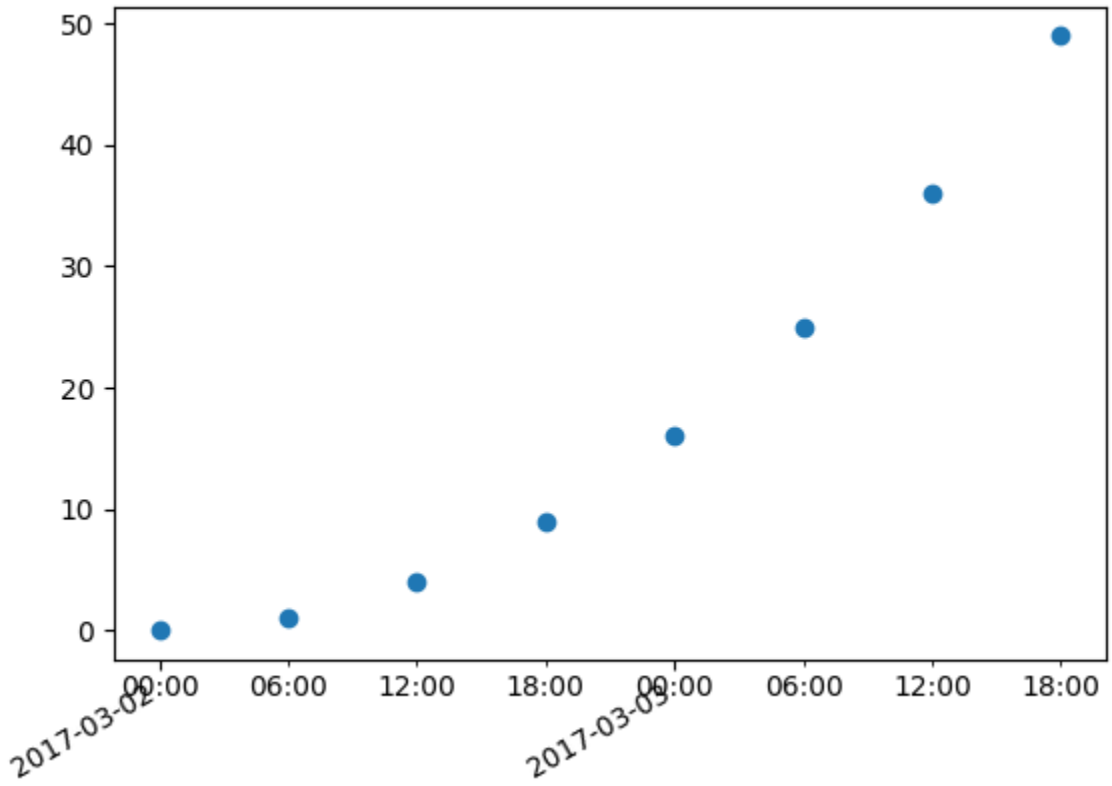


*Figure 13: Only major labels are rotated*

```
1  #adding the which argument to fig.atuofmt_xdate() makes all the labels on the xais rotate
2
3  import datetime
4  import matplotlib.pyplot as plt
5  from matplotlib.dates import DayLocator, HourLocator, DateFormatter, drange
6  from numpy import arange
7
8  date1 = datetime.datetime(2017, 3, 2)
9  date2 = datetime.datetime(2017, 3, 4)
10 delta = datetime.timedelta(hours=6)
11 dates = drange(date1, date2, delta)
12
13 y = arange(len(dates)*1.0)
14
15 fig, ax = plt.subplots()
16 ax.plot_date(dates, y*y)
17
18 ax.xaxis.set_major_locator(DayLocator())
19 ax.xaxis.set_minor_locator(HourLocator(arange(0, 25, 6)))
20 ax.xaxis.set_major_formatter(DateFormatter('%Y-%m-%d'))
21 ax.xaxis.set_minor_formatter(DateFormatter('%H:%M'))
22 ax.xaxis.set_tick_params(which='minor', pad=15)
23 ax.fmt_xdata = DateFormatter('%Y-%m-%d %H:%M:%S')
24 fig.autofmt_xdate(which='both')
25
26 plt.show()
```

*Figure 14: Code to demonstrate after bug fix (https://github.com/CSCD01-Winter2017/team06-Project/blob/master/d3_bug_tests/bug_test_4-fixed.py) We also added padding(line 22) so that the labels did not overlap eachother*

After the bug fix the code outputted this:

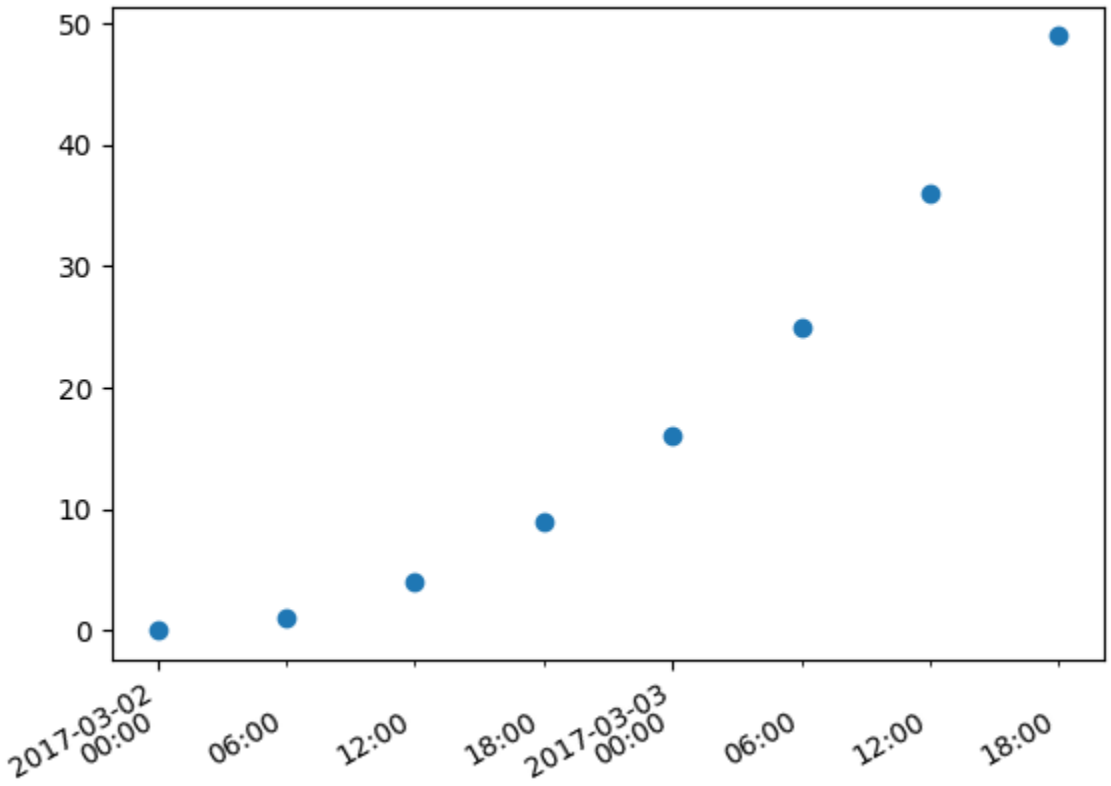


*Figure 15: Output after bug was fixed*

References

**Bug 1:**

The bug: https://github.com/matplotlib/matplotlib/issues/8089

The initial code: https://github.com/matplotlib/matplotlib/blob/master/lib/matplotlib/ticker.py

The code with the fixed bug: https://github.com/CSCD01-Winter2017/team06-Project/blob/master/matplotlib/lib/matplotlib/ticker.py

The test file: https://github.com/CSCD01-Winter2017/team06-Project/blob/master/d3_bug_tests/bug_test_3-broken.py

**Bug 2:**

The bug: https://github.com/matplotlib/matplotlib/issues/8128

The initial code: https://github.com/matplotlib/matplotlib/blob/master/lib/matplotlib/figure.py

The code with the fixed bug: https://github.com/CSCD01-Winter2017/team06-Project/blob/master/matplotlib/lib/matplotlib/figure.py

The test files: https://github.com/CSCD01-Winter2017/team06-Project/blob/master/d3_bug_tests/bug_test_4.py

https://github.com/CSCD01-Winter2017/team06-Project/blob/master/d3_bug_tests/bug_test_4-fixed.py