```
In [697]:   1  # import packages
            2  from kats.detectors.cusum_detection import CUSUMDetector
            3
            4  # synthesize data with simulation
            5  np.random.seed(10)
            6  df_increase_decrease = pd.DataFrame(
            7      {
            8          'time': pd.date_range('2019-01-01', periods=60),
            9          'increase':np.concatenate([np.random.normal(1,0.2,30), np.random.normal(2,0.2,30)]),
           10          'decrease':np.concatenate([np.random.normal(1,0.3,50), np.random.normal(0.5,0.3,10)]),
           11      }
           12  )
```
executed in 147ms, finished 16:38:23 2021-08-12

```
In [698]:   1  tsd = TimeSeriesData(df_increase_decrease.loc[:,['time','increase']])
            2  detector = CUSUMDetector(tsd)
            3  change_points = detector.detector(change_directions=["increase"])
            4
            5  plt.xticks(rotation=45)
            6  detector.plot(change_points)
            7  plt.show()
```
executed in 197ms, finished 16:38:39 2021-08-12

Out[698]: (array([0. , 0.2, 0.4, 0.6, 0.8, 1.]), <a list of 6 Text xticklabel objects>)

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-698-e5aa86acf883> in <module>
      4
      5 plt.xticks(rotation=45)
----> 6 detector.plot(change_points)
      7 plt.show()

~/anaconda3/lib/python3.7/site-packages/kats/detectors/cusum_detection.py in plot(self, change_points)
    570                 )
    571
--> 572             plt.show()
    573
    574

~/anaconda3/lib/python3.7/site-packages/matplotlib/pyplot.py in show(*args, **kw)
    252     """
    253     Display a figure.
--> 254
    255     When running in ipython with its pylab mode, display all
    256     figures and return to the ipython prompt.

~/anaconda3/lib/python3.7/site-packages/ipykernel/pylab/backend_inline.py in show(close, block)
     37             display(
     38                 figure_manager.canvas.figure,
---> 39                 metadata=_fetch_figure_metadata(figure_manager.canvas.figure)
     40             )
     41         finally:

~/anaconda3/lib/python3.7/site-packages/ipykernel/pylab/backend_inline.py in _fetch_figure_metadata(fig)
    175         # the background is transparent
    176         ticksLight = _is_light([label.get_color()
--> 177                                 for axes in fig.axes
    178                                 for axis in (axes.xaxis, axes.yaxis)
    179                                 for label in axis.get_ticklabels()])

~/anaconda3/lib/python3.7/site-packages/ipykernel/pylab/backend_inline.py in <listcomp>(.0)
    177                                 for axes in fig.axes
    178                                 for axis in (axes.xaxis, axes.yaxis)
--> 179                                 for label in axis.get_ticklabels()])
    180         if ticksLight.size and (ticksLight == ticksLight[0]).all():
    181             # there are one or more tick labels, all with the same lightness

~/anaconda3/lib/python3.7/site-packages/matplotlib/axis.py in get_ticklabels(self, minor, which)
   1284
   1285         if which is not None:
-> 1286             if which == 'minor':
   1287                 return self.get_minorticklabels()
   1288             elif which == 'major':

~/anaconda3/lib/python3.7/site-packages/matplotlib/axis.py in get_majorticklabels(self)
   1238     def get_label(self):
   1239         'Return the axis label as a Text instance'
-> 1240         return self.label
   1241
   1242     def get_offset_text(self):

~/anaconda3/lib/python3.7/site-packages/matplotlib/axis.py in get_major_ticks(self, numticks)
   1389         'Get the locator of the major ticker'
   1390         return self.major.locator
-> 1391
   1392     def get_minor_locator(self):
   1393         'Get the locator of the minor ticker'

~/anaconda3/lib/python3.7/site-packages/matplotlib/dates.py in __call__(self)
   1227             vmin -= 2 * unit * interval
   1228             vmax += 2 * unit * interval
-> 1229         return vmin, vmax
   1230
   1231

~/anaconda3/lib/python3.7/site-packages/matplotlib/dates.py in refresh(self)
   1247
   1248     def tick_values(self, vmin, vmax):
-> 1249         delta = relativedelta(vmax, vmin)
   1250
   1251         # We need to cap at the endpoints of valid datetime

~/anaconda3/lib/python3.7/site-packages/matplotlib/dates.py in viewlim_to_dt(self)
    998
    999         # Or more simply, perhaps just a format string for each
-> 1000         # possibility...
   1001
   1002     def __init__(self, locator, tz=None, defaultfmt='%Y-%m-%d'):

ValueError: view limit minimum -7.756991999922634e+16 is less than 1 and is an invalid Matplotlib date value. This often happens if you pass a non-datetime value to an axis that has datetime units
```