

```

In [18]:
# import statements
from time import sleep
from kafka import KafkaConsumer
import datetime as dt
import matplotlib
import matplotlib.pyplot as plt

# this line is needed for the inline display of graphs in Jupyter Notebook
%matplotlib notebook

topic = 'Week9-Topic'

def connect_kafka_consumer():
    _consumer = None
    try:
        _consumer = KafkaConsumer(topic,
                                   consumer_timeout_ms=10000, # stop iteration if no
message after 10 sec
                                   auto_offset_reset='earliest', # comment this if
you don't want to consume earliest available message
                                   bootstrap_servers=['localhost:9092'],
                                   api_version=(0, 10))
    except Exception as ex:
        print('Exception while connecting Kafka')
        print(str(ex))
    finally:
        return _consumer

def init_plots():
    try:
        width = 9.5
        height = 6
        fig = plt.figure(figsize=(width,height)) # create new figure
        ax = fig.add_subplot(111) # adding the subplot axes to the given grid
position
        fig.suptitle('Real-time uniform stream data visualization') # giving figure a
title
        ax.set_xlabel('Time')
        ax.set_ylabel('Value')
        ax.set_ylim(0,110)
        ax.set_yticks([0,20,40,60,80,100])
        fig.show() # displaying the figure
        fig.canvas.draw() # drawing on the canvas
        return fig, ax
    except Exception as ex:
        print(str(ex))

def consume_messages(consumer, fig, ax):
    try:
        # container for x and y values
        x, y = [], []
        # print('Waiting for messages')
        for message in consumer:
            data = str(message.value.decode('utf-8')).split(', ')

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

x.append(data[0])
y.append(int(data[1]))
# print(y)
# we start plotting only when we have 10 data points
if len(y) > 10:
    ax.clear()
    ax.plot(x, y)
    ax.set_xlabel('Time')
    ax.set_ylabel('Value')
    ax.set_ylim(0,110)
    ax.set_yticks([0,20,40,60,80,100])
    fig.canvas.draw()
    x.pop(0) # removing the item in the first position
    y.pop(0)
plt.close('all')
except Exception as ex:
    print(str(ex))

if __name__ == '__main__':

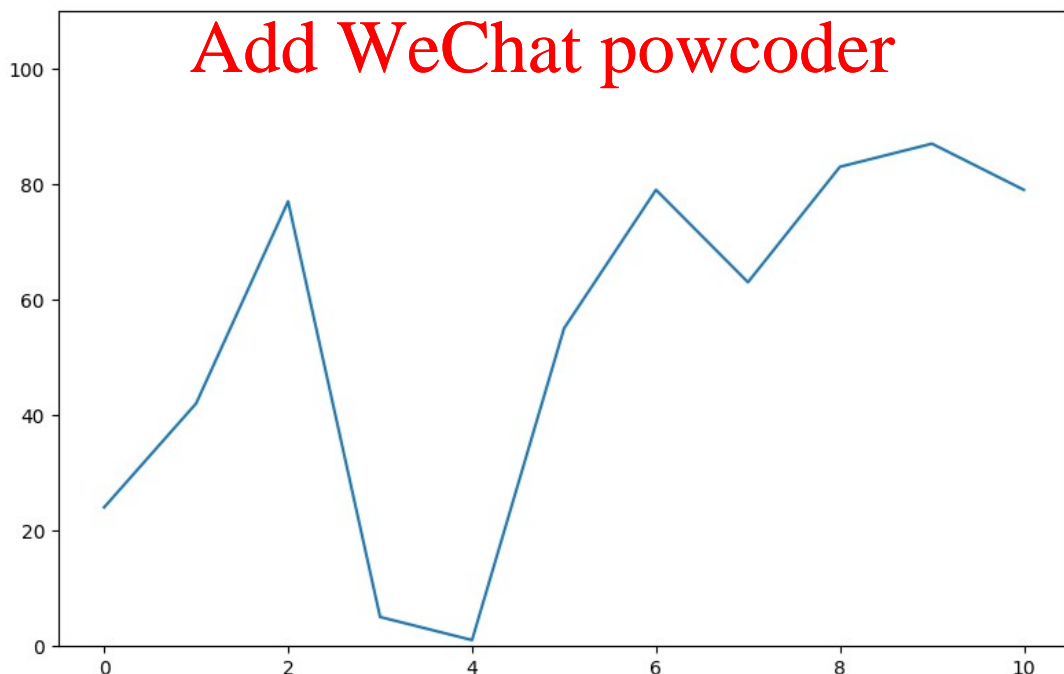
    consumer = connect_kafka_consumer()
    fig, ax = init_plots()
    consume_messages(consumer, fig, ax)

```

## Assignment Project Exam Help

<https://powcoder.com>

Real-time uniform stream data visualization



```

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-18-349a051dd430> in <module>
    71     consumer = connect_kafka_consumer()

```

```

72     fig, ax = init_plots()
--> 73     consume_messages(consumer, fig, ax)
74
75

<ipython-input-18-349a051dd430> in consume_messages(consumer, fig, ax)
54         # we start plotting only when we have 10 data points
55         if len(y) > 10:
--> 56             ax.clear()
57             ax.plot(x, y)
58             ax.set_xlabel('Time')

~/local/lib/python3.8/site-packages/matplotlib/axes/_base.py in clear(self)
1180         """Get the facecolor of the Axes."""
1181         return self.patch.get_facecolor()
-> 1182
1183     def set_facecolor(self, color):
1184         """

~/local/lib/python3.8/site-packages/matplotlib/axes/_base.py in cla(self)
1055         spine.cla()
1056
-> 1057         self.ignore_existing_data_limits = True
1058         self.callbacks = cbbook.CallbackRegistry()
1059

~/local/lib/python3.8/site-packages/matplotlib/axis.py in cla(self)
786         mps.rcParams[axes._rcp[which]] in ('both', 'minor'))
787
--> 788         self.reset_ticks()
789
790         self.converter = None

~/local/lib/python3.8/site-packages/matplotlib/axis.py in reset_ticks(self)
809         pass
810         try:
--> 811             self.set_clip_path(self.axes.patch)
812         except AttributeError:
813             pass

~/local/lib/python3.8/site-packages/matplotlib/axis.py in set_clip_path(self,
clippath, transform)
899     def set_clip_path(self, clippath, transform=None):
900         martist.Artist.set_clip_path(self, clippath, transform)
--> 901     for child in self.majorTicks + self.minorTicks:
902         child.set_clip_path(clippath, transform)
903     self.stale = True

~/local/lib/python3.8/site-packages/matplotlib/axis.py in __get__(self, instance,
cls)
621         else:
622             instance.minorTicks = []
--> 623             tick = instance._get_tick(major=False)
624             instance.minorTicks.append(tick)
625             return instance.minorTicks

```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
~/.local/lib/python3.8/site-packages/matplotlib/axis.py in _get_tick(self, major)
2301         else:
2302             tick_kw = self._minor_tick_kw
-> 2303         return YTick(self.axes, 0, major=major, **tick_kw)
2304
2305     def set_label_position(self, position):
```

```
~/.local/lib/python3.8/site-packages/matplotlib/axis.py in __init__(self, *args,
**kwargs)
488
489     def __init__(self, *args, **kwargs):
--> 490         super().__init__(*args, **kwargs)
491         # x in axes coords, y in data coords
492         self.tick1line.set(
```

```
~/.local/lib/python3.8/site-packages/matplotlib/cbook/deprecation.py in
wrapper(*inner_args, **inner_kwargs)
409             else deprecation_addendum,
410             **kwargs)
-> 411         return func(*inner_args, **inner_kwargs)
412
413     return wrapper
```

```
~/.local/lib/python3.8/site-packages/matplotlib/axis.py in __init__(self, axes, loc,
label, size, width, color, tickdir, pad, labelsizes, labelcolor, zorder, gridOn,
tick1On, tick2On, label1On, label2On, major, labelrotation, grid_color,
grid_linestyle, grid_linewidth, grid_alpha, **kw
152         markeredgewidth=width,
153         )
-> 154         self.tick2line = mlines.Line2D(
155             [], [],
156             color=color, linestyle="none", zorder=zorder, visible=tick2On,
```

```
~/.local/lib/python3.8/site-packages/matplotlib/lines.py in __init__(self, xdata,
ydata, linewidth, linestyle, color, marker, markersize, markeredgewidth,
markeredgewidth, markerfacecolor, markerfacecoloralt, fillstyle, antialiased,
dash_capstyle, solid_capstyle, dash_joinstyle, solid_joinstyle, pickradius,
drawstyle, markevery, **kwargs)
361
362         self.set_linewidth(linewidth)
-> 363         self.set_linestyle(linestyle)
364         self.set_drawstyle(drawstyle)
365
```

```
~/.local/lib/python3.8/site-packages/matplotlib/lines.py in set_linestyle(self, ls)
1139
1140         # get the unscaled dashes
-> 1141         self._us_dashOffset, self._us_dashSeq = _get_dash_pattern(ls)
1142         # compute the linewidth scaled dashes
1143         self._dashOffset, self._dashSeq = _scale_dashes(
```

```
~/.local/lib/python3.8/site-packages/matplotlib/lines.py in _get_dash_pattern(style)
35         # go from short hand -> full strings
36         if isinstance(style, str):
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
---> 37         style = ls_mapper.get(style, style)
      38         # un-dashed styles
      39         if style in ['solid', 'None']:
```

KeyboardInterrupt:

In []:

In []:

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**