First Work

This ia a sub header

Writing a text here

This is a markdown

Congrats we run the app successfully

This is a information

Be cautious

Oops you run into an error,you need to rerun your app again

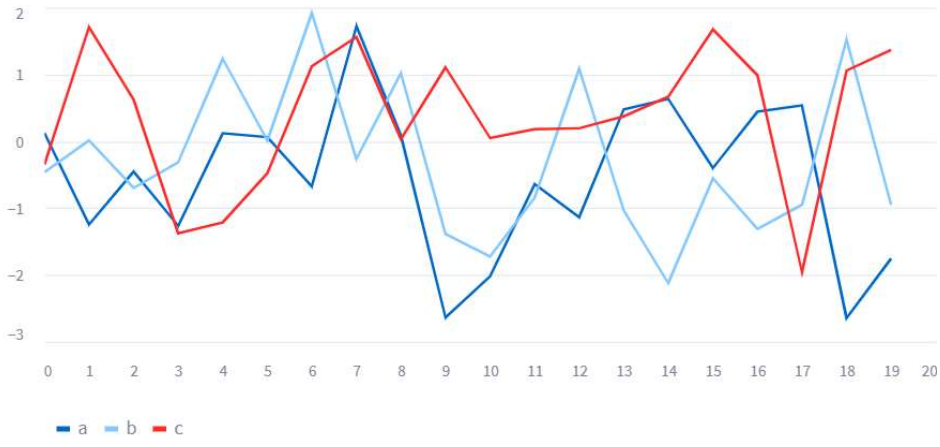
```
range class builtins.range(...)

range(stop) -> range object
range(start, stop[, step]) -> range object

Return an object that produces a sequence of integers from start (inclusive)
to stop (exclusive) by step.  range(i, j) produces i, i+1, i+2, ..., j-1.
start defaults to 0, and stop is omitted!  range(4) produces 0, 1, 2, 3.
These are exactly the valid indices for a list of 4 elements.
When step is given, it specifies the increment (or decrement).
```

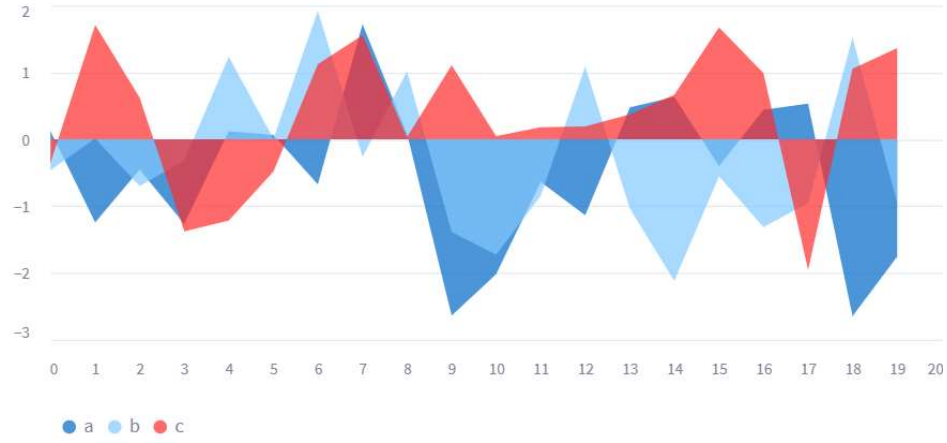
start member_descriptor	<member 'start' of 'range' objects>
step member_descriptor	<member 'step' of 'range' objects>
stop member_descriptor	<member 'stop' of 'range' objects>
count method_descriptor	rangeobject.count(value) -> integer -- return number of occurrences of value
index method_descriptor	rangeobject.index(value) -> integer -- return index of value.

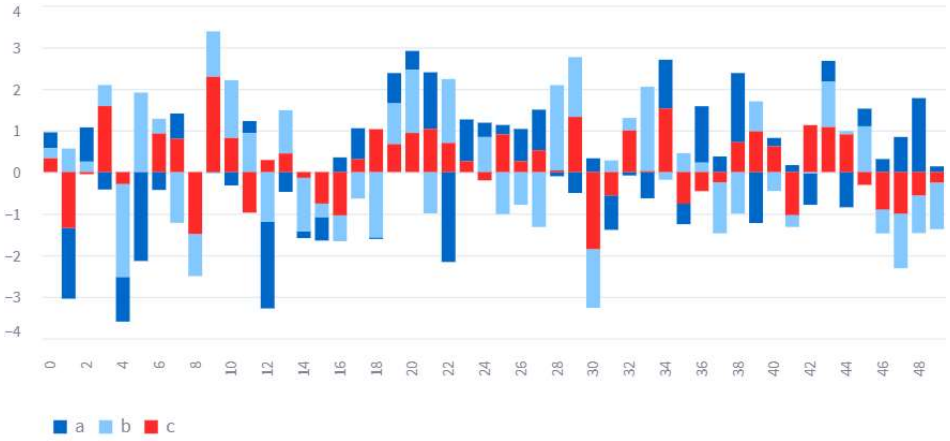
	0	1	2	3	4	5	6	7	8	9	10	11
0	0.2166	0.4271	0.7566	0.6529	0.6406	0.512	0.3481	0.2878	0.5378	0.2173	0.9656	0.874
1	0.3881	0.2362	0.4922	0.6622	0.8154	0.3969	0.0667	0.2177	0.225	0.0512	0.8541	0.571
2	0.8102	0.8048	0.8029	0.0892	0.6537	0.7824	0.6473	0.4996	0.7923	0.1344	0.8688	0.236
3	0.0147	0.9042	0.0314	0.9753	0.504	0.7374	0.2101	0.9114	0.9946	0.4984	0.4387	0.490
4	0.353	0.0262	0.3801	0.649	0.2033	0.856	0.5086	0.0908	0.9055	0.8247	0.5388	0.277
5	0.3659	0.9067	0.9215	0.983	0.3566	0.6473	0.3849	0.4678	0.4408	0.9592	0.6637	0.426
6	0.5371	0.4966	0.7181	0.5517	0.7713	0.0907	0.5252	0.9166	0.4996	0.2123	0.6584	0.771
7	0.385	0.7652	0.4029	0.1083	0.4448	0.6349	0.4576	0.1355	0.4522	0.0739	0.2509	0.394
8	0.6082	0.1496	0.4812	0.9261	0.1612	0.8158	0.5566	0.3333	0.6988	0.2273	0.4242	0.069
9	0.1793	0.9509	0.3516	0.9086	0.4599	0.2944	0.317	0.5747	0.4908	0.6738	0.1709	0.295

---

	col 0	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col
0	0.564711	0.341355	0.481923	0.369372	0.318828	0.579594	0.290297	0.936051	0.349597	0.2
1	0.251974	0.531893	0.317537	0.289216	0.907203	0.772595	0.465167	0.062447	0.118552	0.8
2	0.050429	0.471812	0.679004	0.980073	0.968771	0.407646	0.465571	0.098688	0.867267	0.8
3	0.135271	0.370004	0.393398	0.959118	0.733045	0.553063	0.979221	0.542538	0.770704	0.0
4	0.204476	0.302318	0.977885	0.235364	0.863231	0.245242	0.887111	0.764629	0.700380	0.0
5	0.120345	0.371052	0.619513	0.323072	0.383401	0.987874	0.815381	0.713226	0.221548	0.2
6	0.240806	0.584774	0.866349	0.216947	0.783150	0.759424	0.182824	0.258793	0.183262	0.6
7	0.661960	0.443724	0.774168	0.162125	0.933233	0.254523	0.751409	0.772256	0.719089	0.9
8	0.649835	0.379163	0.930344	0.161239	0.483956	0.014007	0.801772	0.070967	0.347383	0.8
9	0.052334	0.383997	0.662147	0.399169	0.514264	0.214962	0.467478	0.450794	0.601843	0.4







PyplotGlobalUseWarning: You are calling `st.pyplot()` without any arguments. After December 1st, 2020, we will remove the ability to do this as it requires the use of Matplotlib's global figure object, which is not thread-safe.

To future-proof this code, you should pass in a figure as shown below:

```
>>> fig, ax = plt.subplots()
>>> ax.scatter([1, 2, 3], [1, 2, 3])
>>>    ... other plotting actions ...
>>> st.pyplot(fig)
```

You can disable this warning by disabling the config option: `deprecation.showPyplotGlobalUse`

```
st.set_option('deprecation.showPyplotGlobalUse', False)
```

or in your `.streamlit/config.toml`

```
[deprecation]
showPyplotGlobalUse = false
```

ValueError: Image size of 128000x96000 pixels is too large. It must be less than 2^16 in each direction.

Traceback:

```
File "C:\Program Files\Python311\Lib\site-packages\streamlit\runtime\scriptrun
    exec(code, module.__dict__)
File "C:\Users\sinha\introduction.py", line 62, in <module>
    st.pyplot()
File "C:\Program Files\Python311\Lib\site-packages\streamlit\runtime\metrics_u
    result = non_optional_func(*args, **kwargs)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "C:\Program Files\Python311\Lib\site-packages\streamlit\elements\pyplot.p
    marshall(

File "C:\Program Files\Python311\Lib\site-packages\streamlit\elements\pyplot.p
    fig.savefig(image, **kwargs)

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\pyplot.py", line
    fig.canvas.draw_idle()  # Need this if 'transparent=True', to reset colors
    ^^^^^^^^^^^^^^^^^^^^^^

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\backend_bases.py
    self.draw(*args, **kwargs)

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\backends\backend
    self.renderer = self.get_renderer()
                    ^^^^^^^^^^^^^^^^^^^

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\_api\deprecation
    return func(*inner_args, **inner_kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\backends\backend
    self.renderer = RendererAgg(w, h, self.figure.dpi)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Program Files\Python311\Lib\site-packages\matplotlib\backends\backend
    self._renderer = _RendererAgg(int(width), int(height), dpi)
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```