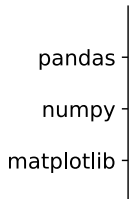
LÊ ĐỨC TÍN - 19522348

```
import pandas as pd

questions = pd.read_csv('/content/data/stackoverflow.zip', parse_dates=True, index_col='creation_date').loc[:,'pandas':'bokeh'].resample('1M'

questions.tail()
```

	pandas	matplotlib	numpy	seaborn	geopandas	geoviews	altair	yellowbrick	vega	holoviews
2021-05-31	200734.0	57853.0	89812.0	6855.0	1456.0	57.0	716.0	46.0	532.0	513.0
2021-06-30	205065.0	58602.0	91026.0	7021.0	1522.0	57.0	760.0	48.0	557.0	521.0
2021-07-31	209235.0	59428.0	92254.0	7174.0	1579.0	62.0	781.0	50.0	572.0	528.0
2021-										

```
from matplotlib.animation import FuncAnimation


import matplotlib.pyplot as plt
from matplotlib import ticker

def bar_plot(data):
  fig, ax = plt.subplots(figsize=(8,6))
  sort_order = data.last('1M').squeeze().sort_values().index
  bars = [
      bar.set_label(label) for label, bar in
      zip(sort_order, ax.barh(sort_order, [0]*data.shape[1]))
  ]

  ax.set_xlabel('total_questions', fontweight='bold')
  ax.set_xlim(0, 250_000)
  ax.xaxis.set_major_formatter(ticker.EngFormatter())
  ax.xaxis.set_tick_params(labelsize=12)
  ax.yaxis.set_tick_params(labelsize=12)

  for spine in ['top', 'right']:
    ax.spines[spine].set_visible(False)

  fig.tight_layout()

  return fig, ax


%config InlineBackend.figure_formats = ['svg']
%matplotlib inline
bar_plot(questions)
```

```
(<Figure size 800x600 with 1 Axes>, <Axes: xlabel='total_questions'>)
```




```
def generate_plot_text(ax):
  annotations = [
      ax.annotate('', xy=(0, bar.get_y() + bar.get_height()/2), ha='left', va='center') for bar in ax.patches  ]

  time_text = ax.text(0.9, 0.1, '', transform=ax.transAxes, fontsize=15, ha='center', va='center')
  return annotations, time_text
```



```
def update(frame, *, ax, df, annotations, time_text):
  data = df.loc[frame, :]

  #update bars
  for rect, text in zip(ax.patches, annotations):
    col = rect.get_label()
    if data[col]:
      rect.set_width(data[col])
      text.set_x(data[col])
      text.set_text(f'{data[col]:,.0f}')

  #update time
  time_text.set_text(frame.strftime('%b\n%Y'))


from functools import partial

def bar_plot_init(questions):
  fig, ax = bar_plot(questions)
  annotations, time_text = generate_plot_text(ax)

  bar_plot_update = partial(update, ax=ax, df=questions, annotations=annotations, time_text=time_text)
  return fig, bar_plot_update


fig, update_func = bar_plot_init(questions)

ani = FuncAnimation(fig, update_func, frames=questions.index, repeat=False)

ani.save('../media/stackoverflow_questions.mp4', writer='ffmpeg', fps=10, bitrate=100, dpi=300)

plt.close()


from IPython import display

display.Video('../media/stackoverflow_questions.mp4', width=600, height=400, embed=True, html_attributes='controls muted autoplay')
```

```
%ls
```

```
data/  data.zip  drive/  __MACOSX/  sample_data/
```

```
cp -av '../media' '/content/drive/MyDrive'
```

```
'../media/stackoverflow_questions.mp4' -> '/content/drive/MyDrive/media/stackoverflow_questions.mp4'
'../media/subway_entries_subplots.mp4' -> '/content/drive/MyDrive/media/subway_entries_subplots.mp4'
```

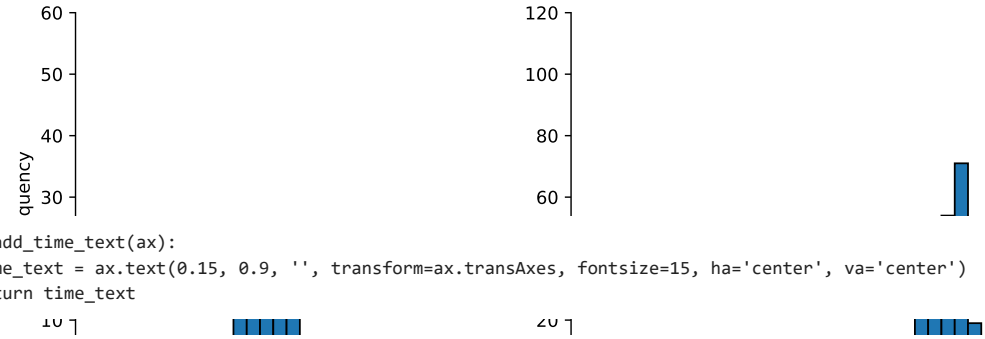
```
subway = pd.read_csv('/content/data/NYC_subway_daily.csv', parse_dates=['Datetime'], index_col=['Borough', 'Datetime'])
subway_daily = subway.unstack(0)
subway_daily.head()
```

	Entries				Exits			
Borough	Bk	Bx	M	Q	Bk	Bx	M	Q
Datetime								
2017-02-04	617650.0	247539.0	1390496.0	408736.0	417449.0	148237.0	1225689.0	279699.0
2017-02-05	542667.0	199078.0	1232537.0	339716.0	405607.0	139856.0	1033610.0	268626.0
2017-02-06	1184916.0	472846.0	2774016.0	787206.0	761166.0	267991.0	2240027.0	537780.0
2017-02-07	1192638.0	470573.0	2892462.0	790557.0	763653.0	270007.0	2325024.0	544828.0
2017-02-08	1243658.0	497412.0	2998897.0	825679.0	788356.0	275695.0	2389534.0	559639.0

```
manhattan_entries = subway_daily['Entries']['M']
```

```
import numpy as np

count_per_bin, bin_ranges = np.histogram(manhattan_entries, bins=30)
```

```
def subway_histogram(data, bins, date_range):
  _, bin_ranges = np.histogram(data, bins=bins)

  weekday_mask = data.index.weekday < 5
  configs = [
      {'label': 'Weekend', 'mask' : ~weekday_mask, 'ymax':60},
      {'label': 'Weekday', 'mask' : weekday_mask, 'ymax': 120}
  ]

  fig, axes = plt.subplots(1,2, figsize=(8,4), sharex=True)
  for ax, config in zip(axes, configs):
    _, _, config['hist'] = ax.hist(
        data[config['mask']].loc[date_range], bin_ranges, ec='black'
    )

    ax.xaxis.set_major_formatter(ticker.EngFormatter())
    ax.set(xlim=(0, None), ylim=(0, config['ymax']),
           xlabel=f'{config["label"]} Entries')
    for spine in ['top', 'right']:
      ax.spines[spine].set_visible(False)

  axes[0].set_ylabel('Frequency')
  fig.suptitle("Histogram of Daily Subway Entries in Manhattan")
  fig.tight_layout()

  return fig, axes, bin_ranges, configs


_ = subway_histogram(manhattan_entries, bins=30, date_range='2017')
```




```
def add_time_text(ax):
  time_text = ax.text(0.15, 0.9, '', transform=ax.transAxes, fontsize=15, ha='center', va='center')
  return time_text
```

```
def update(frame, *, data, configs, time_text, bin_ranges):
  artists = []

  time = frame.strftime('%b\n%Y')
  if time != time_text.get_text():
    time_text.set_text(time)
    artists.append(time_text)

  for config in configs:
    time_frame_mask = \
      (data.index > frame - pd.Timedelta(days=365)) & (data.index <= frame)
    counts, _ = np.histogram(
        data[time_frame_mask & config['mask']], bin_ranges
    )

    for count, rect in zip(counts, config['hist'].patches):
      if count != rect.get_height():
        rect.set_height(count)
        artists.append(rect)

    return artists


def histogram_init(data, bins, initial_date_range):
  fig, axes, bin_ranges, configs = subway_histogram(data, bins, initial_date_range)

  update_func = partial(
      update, data=data, configs=configs,
      time_text=add_time_text(axes[0]),
      bin_ranges=bin_ranges
  )
  return fig, update_func


fig, update_func = histogram_init(
    manhattan_entries, bins=30, initial_date_range=slice('2017', '2019-07')
)

ani = FuncAnimation(
    fig, update_func, frames=manhattan_entries['2019-08': '2021'].index, repeat=False, blit=True
)

ani.save(
    '../media/subway_entries_subplots.mp4',
    writer='ffmpeg', fps=30, bitrate=500, dpi=300
)

plt.close()


from IPython import display

display.Video('../media/subway_entries_subplots', width=600, height=400, embed=True, html_attributes='controls muted autoplay')
```

```
!pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: geopandas in /usr/local/lib/python3.10/dist-packages (0.12.2)
Requirement already satisfied: pyproj>=2.6.1.post1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (3.5.0)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.5.3)
Requirement already satisfied: fiona>=1.8 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.9.3)
Requirement already satisfied: shapely>=1.7 in /usr/local/lib/python3.10/dist-packages (from geopandas) (2.0.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas) (23.1)
Requirement already satisfied: munch>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (2.5.0)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (1.1.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (2022.12.7)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (8.1.3)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (0.7.2)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8->geopandas) (23.1.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->geopandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->geopandas) (1.22.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->geopandas) (2.8.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from munch>=2.3.2->fiona>=1.8->geopandas) (1.16.0)
```

```
import geopandas as gpd
import pandas as pd
earthquakes = gpd.read_file('/content/data/earthquakes.geojson').assign(
    time=lambda x:pd.to_datetime(x.time, unit='ms'),
    month=lambda x: x.time.dt.month
)[['geometry', 'mag', 'time', 'month']]

earthquakes.shape
```

```
(188527, 4)
```

```
earthquakes.head()
```

	geometry	mag	time	month
0	POINT Z (-67.12750 19.21750 12.00000)	2.75	2020-01-01 00:01:56.590	1
1	POINT Z (-67.09010 19.07660 6.00000)	2.55	2020-01-01 00:03:38.210	1
2	POINT Z (-66.85410 17.87050 6.00000)	1.81	2020-01-01 00:05:09.440	1
3	POINT Z (-66.86360 17.89930 8.00000)	1.84	2020-01-01 00:05:36.930	1
4	POINT Z (-66.86850 17.90660 8.00000)	1.64	2020-01-01 00:09:20.060	1

```
!pip install geoviews
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: geoviews in /usr/local/lib/python3.10/dist-packages (1.9.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geoviews) (23.1)
Requirement already satisfied: shapely in /usr/local/lib/python3.10/dist-packages (from geoviews) (2.0.1)
Requirement already satisfied: panel in /usr/local/lib/python3.10/dist-packages (from geoviews) (0.14.4)
Requirement already satisfied: param in /usr/local/lib/python3.10/dist-packages (from geoviews) (1.13.0)
Requirement already satisfied: holoviews>=1.14.2 in /usr/local/lib/python3.10/dist-packages (from geoviews) (1.15.4)
Requirement already satisfied: cartopy>=0.18.0 in /usr/local/lib/python3.10/dist-packages (from geoviews) (0.21.1)
Requirement already satisfied: bokeh<2.5,>=2.4.0 in /usr/local/lib/python3.10/dist-packages (from geoviews) (2.4.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from geoviews) (1.22.4)
Requirement already satisfied: typing-extensions>=3.10.0 in /usr/local/lib/python3.10/dist-packages (from bokeh<2.5,>=2.4.0->geoviews) (
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from bokeh<2.5,>=2.4.0->geoviews) (8.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (from bokeh<2.5,>=2.4.0->geoviews) (6.0)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (from bokeh<2.5,>=2.4.0->geoviews) (6.2)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from bokeh<2.5,>=2.4.0->geoviews) (3.1.2)
```

```
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.10/dist-packages (from cartopy>=0.18.0->geoviews) (3.7.1)
Requirement already satisfied: pyshp>=2.1 in /usr/local/lib/python3.10/dist-packages (from cartopy>=0.18.0->geoviews) (2.3.1)
Requirement already satisfied: pyproj>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from cartopy>=0.18.0->geoviews) (3.5.0)
Requirement already satisfied: pandas>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from holoviews>=1.14.2->geoviews) (1.5.3)
Requirement already satisfied: colorcet in /usr/local/lib/python3.10/dist-packages (from holoviews>=1.14.2->geoviews) (3.0.1)
Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.10/dist-packages (from holoviews>=1.14.2->geoviews) (2.2.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (6.0.0)
Requirement already satisfied: setuptools>=42 in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (67.7.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (2.27.1)
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (3.4.3)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (4.65.0)
Requirement already satisfied: pyct>=0.4.4 in /usr/local/lib/python3.10/dist-packages (from panel->geoviews) (0.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->bokeh<2.5,>=2.4.0->geoviews
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->geoviews)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->geovi
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->geovi
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->geov
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->geov
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->cartopy>=0.18.0->g
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20.0->holoviews>=1.14.2->geoviews
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from pyproj>=3.0.0->cartopy>=0.18.0->geoviews) (2022.
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from bleach->panel->geoviews) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->panel->geoviews) (0.5.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->panel->geoviews) (3.4)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->panel->geoviews) (2.
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->panel->geoviews) (1.26.1
```

```
import geoviews as gv
import geoviews.feature as gf
import holoviews as hv

gv.extension('matplotlib')
```

```
import calendar

def plot_earthquakes(data, month_num):
  points = gv.Points(
      data.query(f'month == {month_num}'),
      kdims=['longitude', 'latitude'],
      vdims=["mag"]
  ).redim.range(mag=(-2, 10), latitude=(-90, 90))


  overlay = gf.land * gf.coastline * gf.borders * points

  return overlay.opts(
      gv.opts.Points(color='mag', cmap='fire_r', colorbar=True, alpha=0.75),
      gv.opts.Overlay(
          global_extent=False, title=f'{calendar.month_name[month_num]}', fontscale=2
      )
  )


# plot_earthquakes(earthquakes,·1).opts(
#·····fig_inches·=(6,3),·aspect=2,·fig_size=250,·fig_bounds=(0.07,·0.05,·0.87,·0.95)
#·)


# frames = {
#     month_num:plot_earthquakes(earthquakes, month_num)
#     for month_num in range(1, 13)
# }

# holomap = hv.HoloMap(frames)


# hv.output(
#     holomap.opts(
#         fig_inches=(6,3), aspect=2, fig_size=250,
#         fig_bounds=(0.07, 0.05, 0.87, 0.95)
#     ), holomap='gif', fps=5
# )


# hv.save(
#     holomap.opts(
```

```
#         fig_inches=(6,3), aspect=2, fig_size=250,
#         fig_bounds=(0.07, 0.05, 0.87, 0.95)
#     ), 'earthquakes.gif', fps=5
# )
```

✓ 0 giây hoàn thành lúc 09:55