Upgrade Open in app Published in CodeX

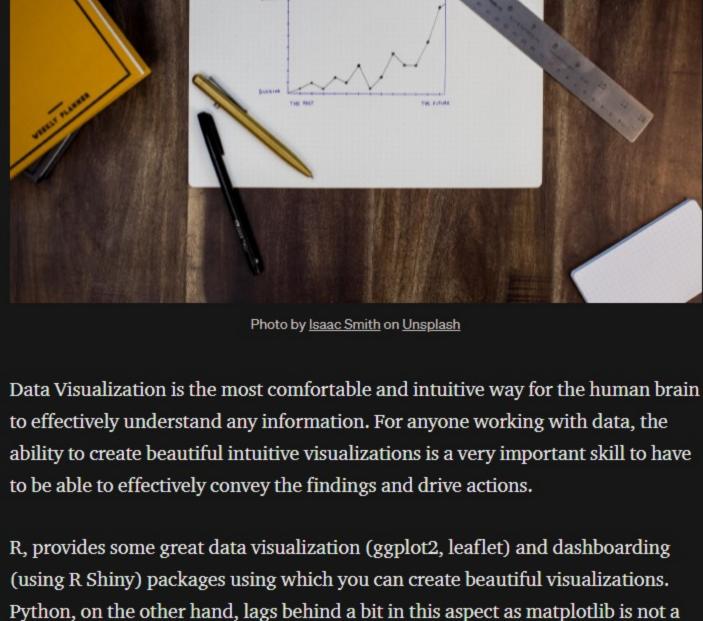
G III Ø

Plots It's time to up your visualization game!

Don't use Matplotlib or Seaborn for Your Python

Anmol Tomar Follow

Feb 3 · 5 min read · • Listen



the capability of making these interactive. With a static plot, we cannot zoom into the interesting sections of the plots, hover over the plots to see the specific information, and more.

great visualization package.

So, here comes the plotly package! iiii plotly

Plotly is a Python library for creating interactive, publication-quality visualizations. Plotly not only makes the plots interactive, a functionality missing in matplotlib or seaborn, but also provides a variety of more charts such as: · Statistical Charts such as tree plots, error bands, parallel categories diagrams,

Scientific Charts such as contour plots, log plots, etc.

Maps such as bubble map, density map, etc.

Financial Charts such as funnel charts, candlestick charts, etc

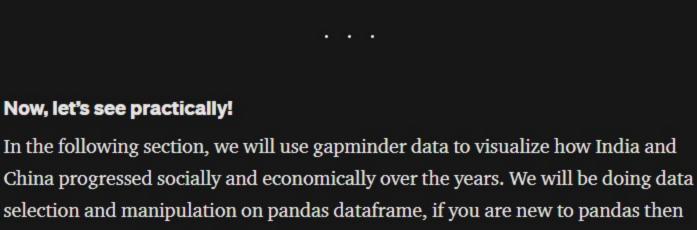
Image credits: plotly

 Bioinformatics and more. That was all about the theoretical explanation of why you should use plotly over

matplotlib or seaborn.

4

etc.



modification. Let's start by installing the plotly package and importing the required packages. # install plotly

and the express module will be used to read the gapminder data. Now, let's read

the gapminder data using the plotly express and look at the top 5 rows.

Using the graph_objects module, we will be creating the different types of graphs

reading the data and looking at the 1st five rows using head()

df = px.data.gapminder()

plotly_plots_data.py hosted with 💙 by GitHub

country continent year lifeExp

df.head()

697

698

699

700

701

702

8

10

12

2

4

6

12 13

14

fig

Life Expectancy (years) 55

8 9

10

11

12

fig

1.28

India

India

India

India

India

India

bars to the country name.

Afghanistan Asia 1952 28.801 8425333 779.445314 AFG 30.332 Afghanistan Asia 1957 9240934 820.853030 AFG 4 31.997 10267083 853.100710 Afghanistan Asia 1962 **AFG** Afghanistan 1967 34.020 AFG Asia 11537966 836.197138 Afghanistan 1972 36.088 13079460 739.981106 **AFG** 4 Asia From the above table, we can see that the data contains information around life

expectancy, population, GDP per capita of different countries across different

years. Now, we will filter the dataframe for India and China.

Asia 1957 40.249

Asia 1962 43.605

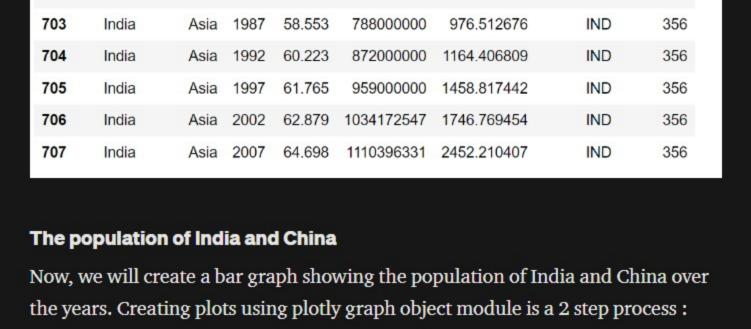
Asia 1967 47.193

Asia 1972 50.651

Asia 1977 54.208

Asia 1982 56.596

pop



1. Setting up the figure function within which we will set the data argument.

The data argument is set to a list having the bar functions (go.Bar) for India

and China. Within the bar function, we set the x-axis to year column, y-axis to

population column, the marker color to indianred/blue, and the name of the

plotly_plots_bar.py hosted with \ by GitHub view raw Population of India and China over the years

- Image by Author From the above chart, we can see the population of both the countries has grown at a Life Wh ve etc na. are
- Image by Author There has been a consistent gap between the life expectancy of the 2 countries, except around 1960 when the life expectancy in China dropped suddenly due to 'The Great Chinese Famine'. Life Expectancy vs GDP per capita over the years When we want to see the interaction between 2 continuous (numeric) features, the scatter plot is the go-to plot. In the following code, we are creating the scatter plot between Life expectancy and GDP per capita of the 2 countries. fig = go.Figure(data = [go.Scatter(y = df_india['lifeExp'], x = df_india['gdpPercap'],\ mode = 'markers', name = 'India') 2 3 go.Scatter(y = df_china['lifeExp'], x = df_china['gdpPercap'],\ 4 5 mode = 'markers', name = 'China') 6]) 7

fig.update_layout(title='Life Expectency vs GDP per Capita',

xaxis_title='gdpPercap',

plotly_plots_scatter.py hosted with \ by GitHub

Life Expectency vs GDP per Capita

65

Life Expectancy (years)

per capita.

yaxis_title='Life Expectancy (years)',

Years

2 fig = px.scatter(df, x = 'gdpPercap', y = 'lifeExp', size = 'pop', 4 color = 'continent', hover_name='country', log_x= True, size_max=50, title = 'World Development in 2007', 5 6 animation_frame="year", animation_group="country", range_y = [25,90]) fig.update_layout(xaxis = dict(showgrid=False), yaxis = dict(showgrid=False)) 8 9 10 fig plotly_plots_bubble.py hosted with 💜 by GitHub view raw Let's understand some important arguments of the function: 1. size: This is set to a numeric column that will capture the size of the bubble. color: This is set in a categorical column that will capture the color of the bubble. In our example, one color is assigned to each continent by default.

gdpPercap

Image by Author

For both countries, life expectancy has increased with the increase in GDP per

Using a bubble plot, we can show 3 dimensions (x-axis, y-axis, and size of

bubble) over a 2D plot. Using the following code, we can create a bubble chart

World Development over time: An Animation

df = px.data.gapminder()

using the scatter function of the plotly express module.

capita. Compared to India, China has made significant improvements in the GDP

5 Ways You Can Stand Out as an Early-Career Software Engineer

Share your ideas with millions of readers. Write on Medium up, he pays you 60% of the bet. If it lands tails, you pay him 40%. This sounds like a great deal because his offer has a mean return of 10%. (0.6 ... ₩ ... Python 5 min read

₩ ...

₩ ...

Why The Metaverse is a Threat to Human Society —

Seaborn is a good alternative for creating static plots in python but doesn't have

view raw

view raw

gdpPercap iso alpha iso num

you can learn python for data analysis. You can open the jupyter notebook and use the codes directly without any ! pip install plotly # import plotly packages import plotly.graph_objects as go import plotly.express as px install_plotly.py hosted with 💙 by GitHub

reading the datasets for India and China df_india = df[df['country'] == 'India'] df_china = df[df['country'] == 'China'] 3 df_india plotly_plots_datasets.py hosted with 💖 by GitHub view raw The below table shows the data for India. The data ranges from the year 1952 to 2007. country continent year lifeExp gdpPercap iso_alpha iso_num pop Asia 1952 37.373 372000000 546.565749 696 India IND 356

409000000

454000000

506000000

567000000

634000000

708000000

590.061996

658.347151

700.770611

724.032527

813.337323

855.723538

IND

IND

IND

IND

IND

IND

356

356

356

356

356

356

update_layout function. ## Using figure function to define the data argument and setting it to the bars for India and Chin 2 fig = go.Figure(data = [go.Bar(x = df_india['year'], y = df_india['pop'], marker_color = 'indianre go.Bar(x = df_china['year'], y = df_china['pop'], marker_color = 'blue',name = 'Ch 3 4]) 5 ## Setting the titles, xaxis and yaxis 6

2. Setting the text for the title, x-axis, and y-axis of the charts using the

fig.update_layout(title='Population of India and China over the years',

xaxis_title='Years',

)

yaxis_title='Population',

a similar rate.	
e Expectancy over the years	
	asurements over years/months/weeks
Y	the trend. Using the following code, we anged over the years for India and China.
# step 1 : Setting the figure function	
fig = go.Figure(data = [go.Scatter(x = df_ind	dia['year'], y = df_india['lifeExp'],\
line = di	ict(color = 'firebrick', width = 4),
text = df	f_india['country'], name = 'India'),
<pre>go.Scatter(x = df_chi</pre>	ina['year'], y = df_china['lifeExp'],\
line = dic	t(color = 'blue', width = 4), text = df_china['countr
# step 2 : Setting the update_layout function	1
<pre>fig.update_layout(title='Life Expectency over</pre>	the years',
xaxis_title='Years',	

yaxis_title='Life Expectancy (years)',

view raw

view raw

China

)

plotly_plots_line.py hosted with W by GitHub

Life Expectency over the years

Sign up for CrunchX By CodeX
A weekly newsletter on what's going on around the tech and programming space <u>Take a</u> <u>look.</u>
Emails will be sent to jeabinay@gmail.com. Not you?
More from CodeX
more from codex
Everything connected with Tech & Code. Follow to join our 900K+ monthly readers
Matt Stine · Feb 3 ★

I will tell you the five things I wish I had known when I began my career. I've had a very successful career. But that success was determined by... Software Engineering 2 min read 🚺 Carl Westerby · Feb 3 🖈 Heads I Win, Tails You Lose Suppose your friend offers you to bet on a coin flip. If the coin lands heads

A quick story of how crazy we've gone. First, let's remind everyone what is a metaverse. The definition is really simple: a digital world where users ca...

Read more from CodeX

🧱 Denys Opria - Ukrainian · Feb 3 🖈 30 Days of Work with Blue-Filter Glasses Is there any sense? — I am a software engineer. I spend 8 hours per day in front of a screen working on my primary job. Then I spend 2-3 hours more working on my pet project. Then I give myself a rest watching YouTube or...

This acts as a size standardization parameter. year column. they describe the same object in each frame. We want to see how each country progressed over the years, so setting it to the country column. World Development in 2007 Image by Author

3. log x : Set the X-axis(GDP per capita) to log-scaled. 4. size_max: Set the maximum size of the bubble, when the size argument is set. 5. animation_frame: Values of the dataframe column that are used to assign marks to the animation frames. In our example, the argument is set to the 6. animation_group: Rows with matching 'animation_group' will be treated as if Europe Africa Americas Oceania Countries across the world have developed quite a lot over these years. We can see the direct correlation between life expectancy and GDP per capita (both increase with time) for all the countries. A lot more findings can be derived from

this chart, share what you find over the comments. What's more! You can also create interactive dashboards using plotly. Dash is the plotly learn HTML, CSS and Javascript. Learn more about Dash here: https://bit.ly/311k37f use cases. Q 19 1,402

framework for creating beautiful python dashboards quickly without having to You can follow me if you love to read about Data Science tools, technologies, and

Health 3 min read

Reactis 7 min read

Pierre-Louis Lemaire · Feb 3

Not an Opportunity

Metaverse 4 min read

Part)

Imran Faroog · Feb 3 ★

Master The UseEffect Hook With Simple App (Final eact **Hoo** Now, there is one last important thing you need to know about useEffect or about the built-in features of UseEffect. We learned about effect function...