



Interactive Visualization with Plotly - Plotly Express - 3

One should look for what is and not what he thinks should be. (Albert Einstein)

Warm up

- Check out this article on various visualizations that can be made with Plottly express:
Link
 - What did you find the most interesting and relevant? How do you plan to use interactive visualizations in your work?

Module completion checklist

Objective	Complete
Build heatmaps and geographic maps using plotly	
Add animations to maps	

Introduction to Heatmaps

- Exploratory data analysis comes in many forms, but if we are trying to create a model to predict a certain variable then we want to know if our target variable is correlated to our predictors
- This is where a **correlation plot** comes in handy, each square represents the correlation value between two variables

Creating heatmaps

- Here's how we can create a heatmap using plotly

```
# Select only numeric columns from
iris_dataset
numeric_iris =
iris_dataset.select_dtypes(include='numb

# Compute the correlation matrix on
numeric columns
cormat = numeric_iris.corr()

# Create a heatmap of iris_dataset
correlation
fig = px.imshow(cormat,
                zmin = -1,
                zmax = 1)
fig.show()
```



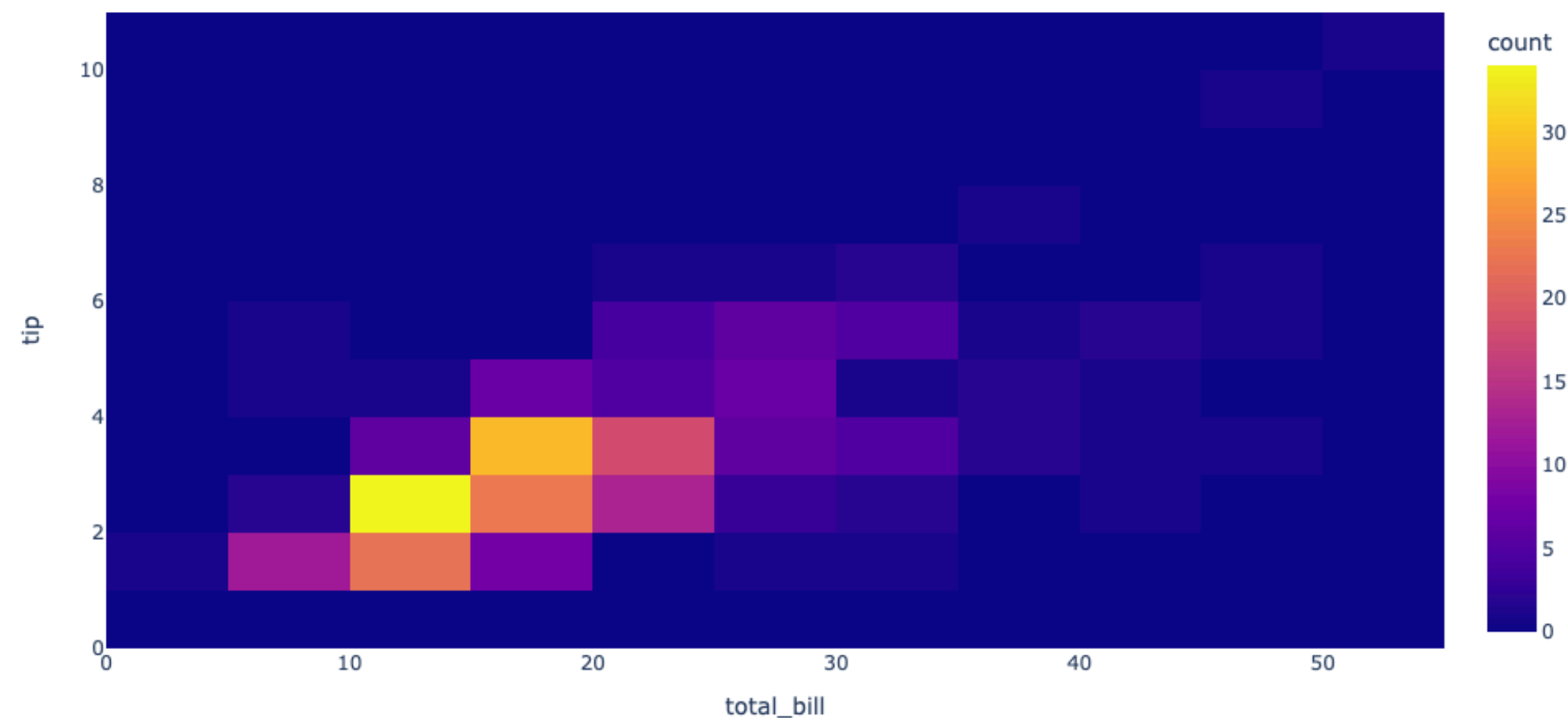
Density heatmaps

- A density heatmap is a two dimensional histogram
- This allows us to see if two variables have approximately the same distribution shape
- This can be very helpful for regression models, if two variables have the same distribution shape, we do not need to do any data transformation for one to predict the other

Density heatmaps (cont'd)

- We can also make a density heatmap between predictors to see if another predictor is going to add to the variance explained

```
# Create a density heatmap of tips dataset using plotly function density_heatmap
fig = px.density_heatmap(tips_dataset,
                        x = "total_bill",
                        y = "tip")
fig.show()
```

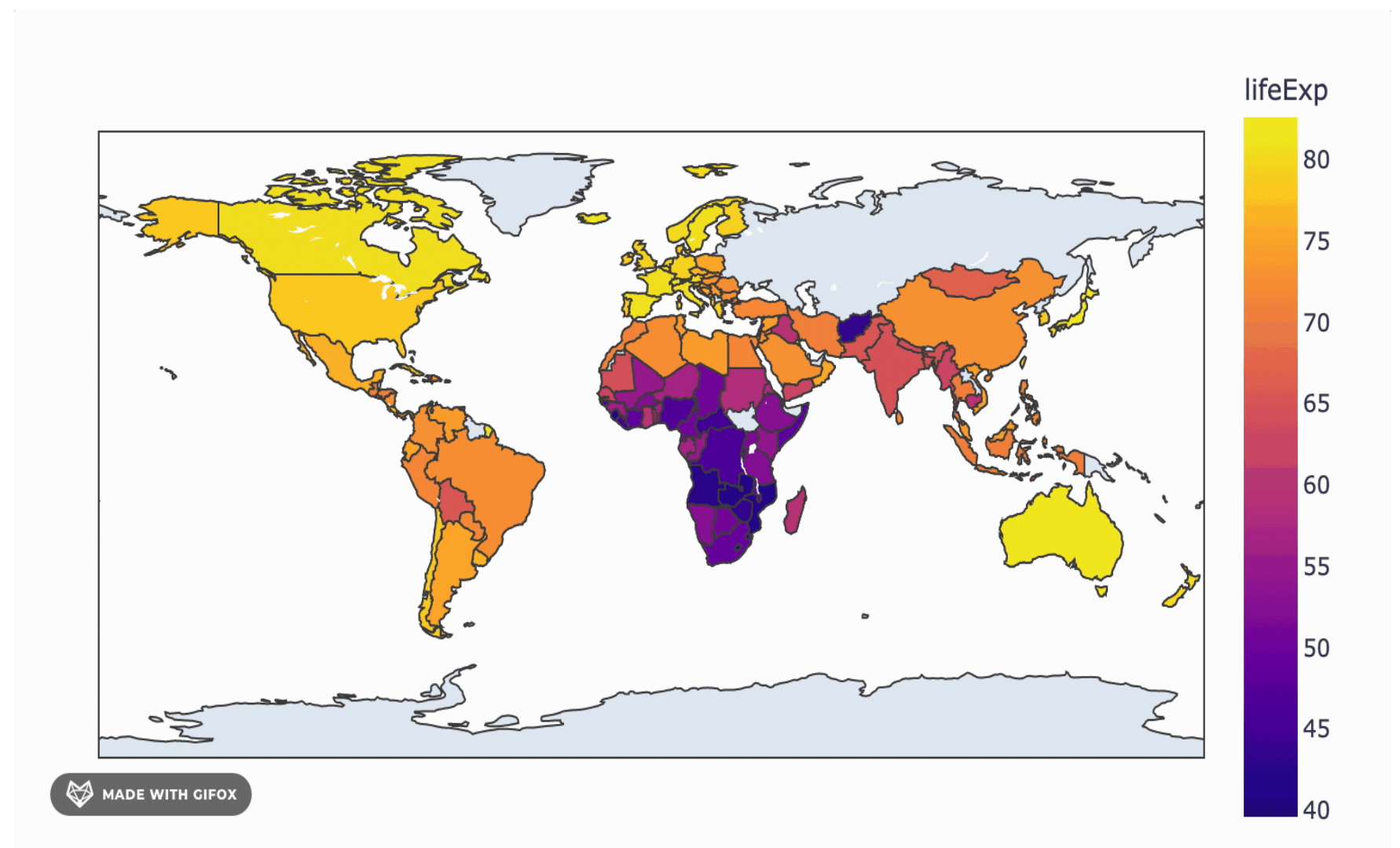


World map

- Data is being gathered at a high speed all over the world and comparisons between areas become necessary
- The best way to represent this data is on a map
- We'll be using the gapminder dataset to demo the maps
- This dataset includes life expectancy and total population data at the level of continent, country and year
- We'll take a subset of this dataset as mentioned here

```
# Take a subset from gapminder dataset
gapminder_subset =
px.data.gapminder().query("year==2007")
```

```
# Create a map using plotly function choropleth
fig = px.choropleth(gapminder_subset,
                    locations="iso_alpha",
                    color = "lifeExp", #a variable of
gapminder
                    hover_name = "country", # add
hover information
                    color_continuous_scale =
px.colors.sequential.Plasma)
fig.show()
```

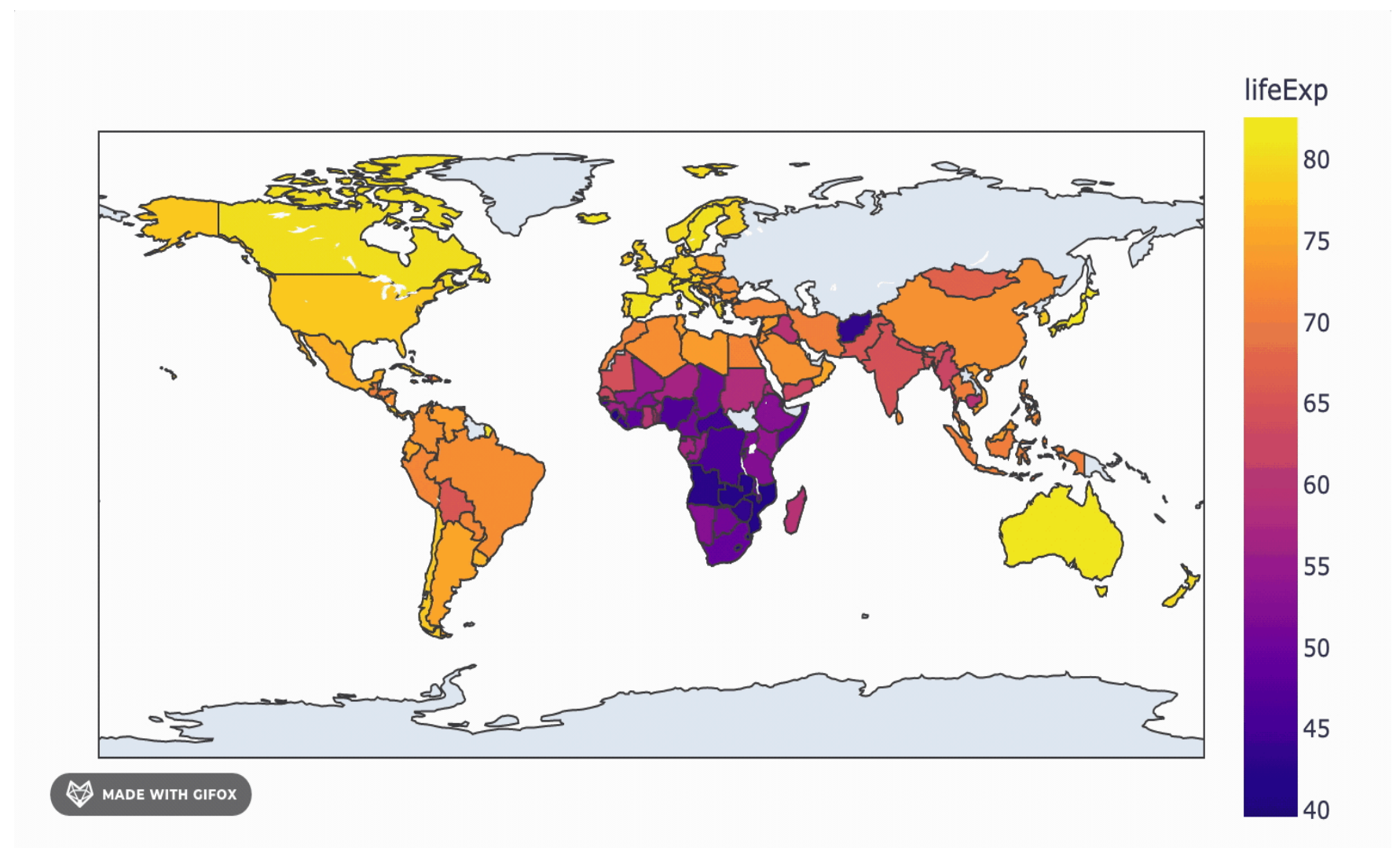


World map (cont'd)

- This dataset includes life expectancy and total population data at the level of continent, country and year
- We'll take a subset of this dataset as mentioned here

```
# Take a subset from gapminder dataset
gapminder_subset =
px.data.gapminder().query("year==2007")
```

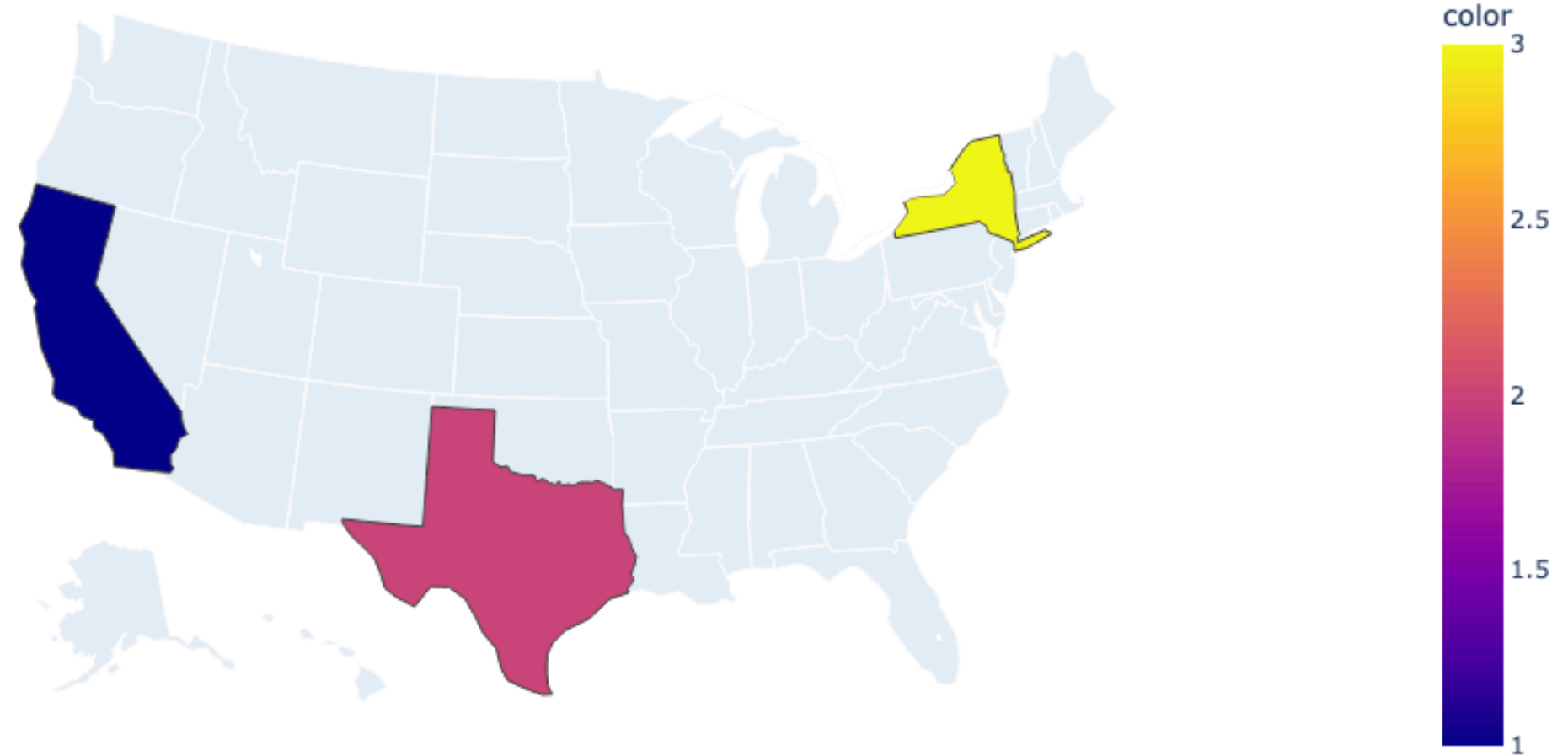
```
# Create a map using plotly function choropleth
fig = px.choropleth(gapminder_subset,
                    locations="iso_alpha",
                    color = "lifeExp", #a variable of
gapminder
                    hover_name = "country", # add
hover information
                    color_continuous_scale =
px.colors.sequential.Plasma)
fig.show()
```



The US map

- `plotly express` also has the built-in capabilities to put data on a US map by using an extra parameter called `scope`
- Here we have simply made-up data, but we can easily download data with U.S. states information to plot on this map

```
fig = px.choropleth(locations=["CA", "TX", "NY"],  
                    locationmode="USA-states",  
                    color=[1,2,3],  
                    scope="usa")  
fig.show()
```



Downloading a custom GeoJSON to make a map

- Not all data fits nicely on a world map or a U.S. map; perhaps we want to show things at a more granular level, like county or district or even neighborhood level
- If this is the case, we might need to get a 'geoJSON' file to add to our map
- A 'geoJSON' contains all the border lines for the areas we want our map to be divided into
- You can generally find a 'geoJSON' already available for what you are looking to map, but if not, there are plenty of generators that are free online

Downloading a custom GeoJSON to make a map (cont'd)

- We will create a map of Montreal Election data by district
- We need to be sure our `locations` parameter is set to the location column in our dataframe and the `featureidkey` parameter is set to the feature key in the JSON which holds the matching granularity

Downloading a custom GeoJSON to make a map (cont'd)

```
election_dataset = px.data.election()
geojson = px.data.election_geojson()

fig = px.choropleth(election_dataset,
                    geojson=geojson, color="Bergeron",
                    locations="district",
                    featureidkey="properties.district",
                    projection="mercator")

fig.update_geos(fitbounds="locations",
                visible=False)

fig.update_layout(margin=
{"r":0,"t":0,"l":0,"b":0})

fig.show()
```



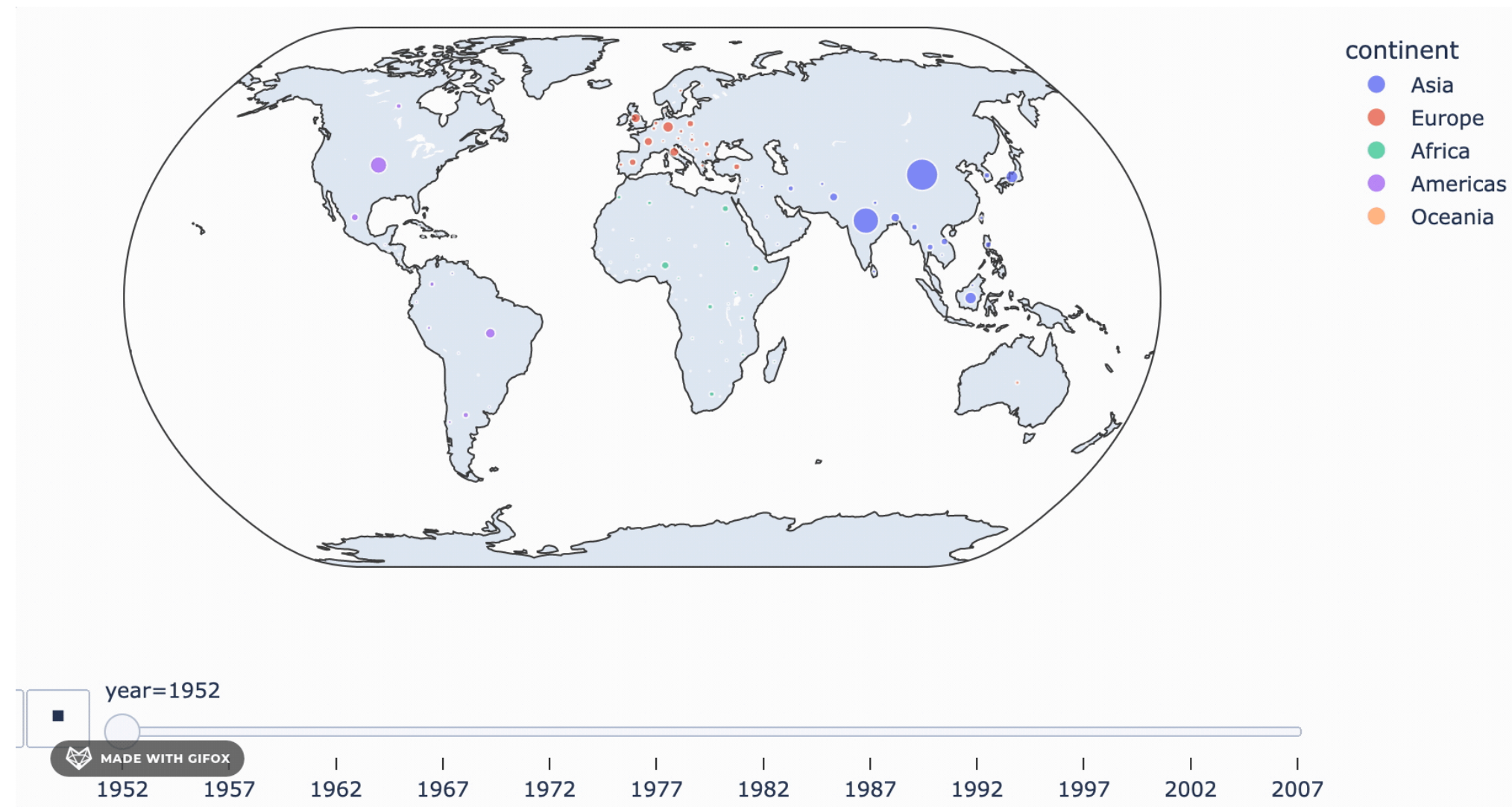
Module completion checklist

Objective	Complete
Build heatmaps and geographic maps using plotly	✓
Add animations to maps	

Animation of a scatter map

- Instead of color shades, we may want to add dots in locations based on data, this is called a **scatter map**
- We can also add animation to all of these maps by adding in a time variable in the argument `animation_frame`

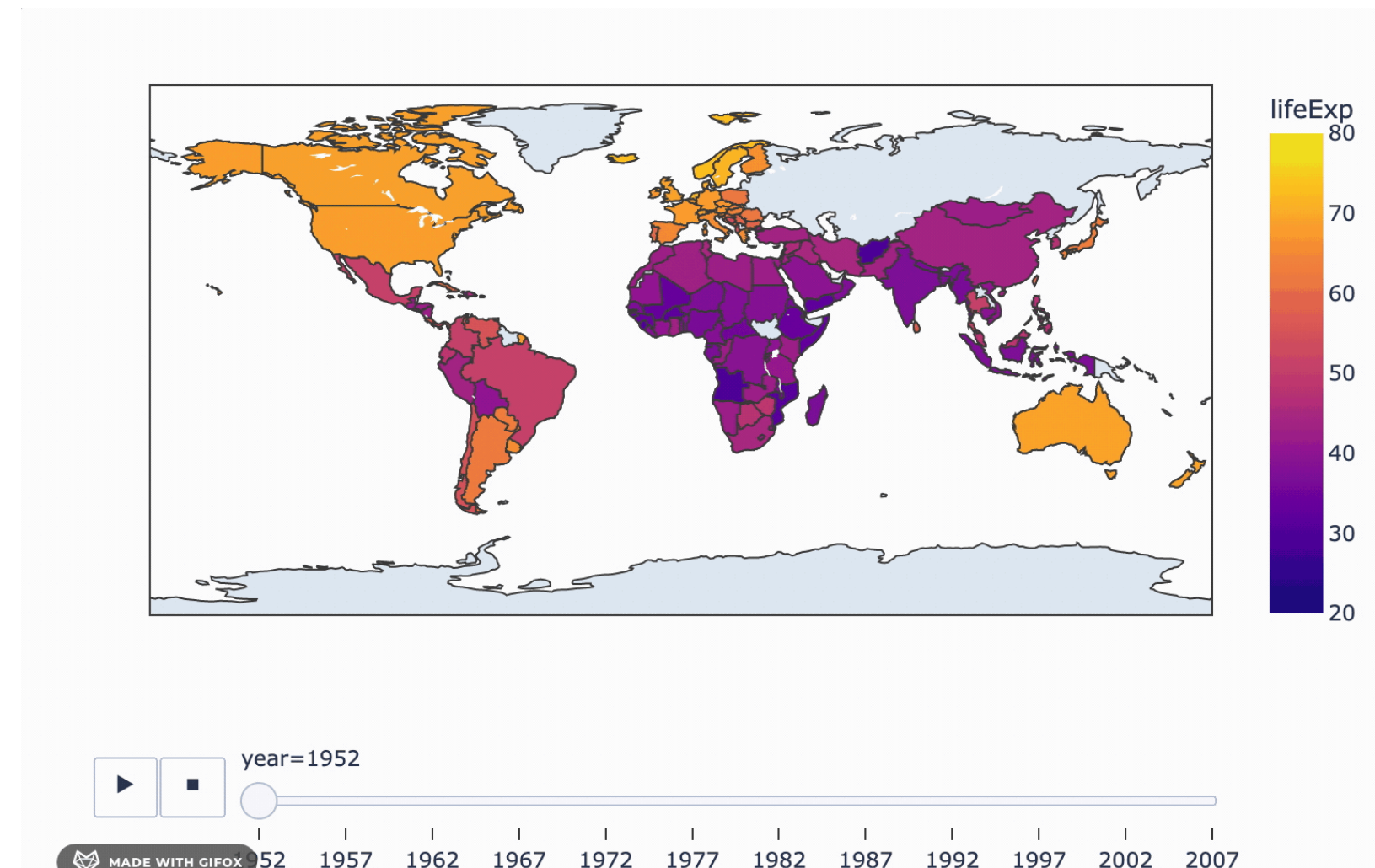
```
gapminder_dataset = px.data.gapminder()
fig = px.scatter_geo(gapminder_dataset,
                    locations="iso_alpha",
                    color="continent",
                    hover_name="country",
                    size="pop",
                    animation_frame="year",
                    projection="natural earth")
fig.show()
```



Animation of a heat map

- We can also add this animation to our choropleth maps using that same argument
- We can add to the animation effects by using the argument `animation_group`

```
fig = px.choropleth(gapminder_dataset,  
                    locations="iso_alpha",  
                    color="lifeExp",  
                    hover_name="country",  
                    animation_frame="year",  
                    range_color=[20,80])  
fig.show()
```



Knowledge check



Exercise



You are now ready to try tasks 9-14 in the Exercise for this topic

Module completion checklist

Objective	Complete
Build heatmaps and geographic maps using plotly	✓
Add animations to maps	✓

Plotly Express: Topic summary

In this part of the course, we have covered:

- Introduction to plotly express
- Organize and visualize data with plotly express

Congratulations on completing this module!

