

```
In [1]: import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.express as px
import pandas as pd
import json
import ipywidgets as widgets
from IPython.display import display
```

```
In [2]: init_notebook_mode(connected=True)
```

```
In [3]: data_in = dict(type = "choropleth",
                     locations=['india'],
                     locationmode = 'country names',
                     colorscale='Portland',
                     text=['India'],
                     z = [1.0],
                     colorbar = {'title':'Colorbar Title Goes Here'})
```

```
In [4]: # Reading the geojson file:
india_states = json.load(open("states_india.geojson", 'r'))
```

```
In [5]: lit_data = pd.read_csv('GOI.csv')
lit_data.head()
```

Out[5]:

	Category	Country/ States/ Union Territories Name	Literacy Rate (Persons) - Total - 2001	Literacy Rate (Persons) - Total - 2011	Literacy Rate (Persons) - Rural - 2001	Literacy Rate (Persons) - Rural - 2011	Literacy Rate (Persons) - Urban - 2001	Literacy Rate (Persons) - Urban - 2011
0	State	Andhra Pradesh	60.5	67.0	54.5	60.4	76.1	80.1
1	State	Arunachal Pradesh	54.3	65.4	47.8	59.9	78.3	82.9
2	State	Assam	63.3	72.2	59.7	69.3	85.3	88.5
3	State	Bihar	47.0	61.8	43.9	59.8	71.9	76.9
4	State	Chhattisgarh	64.7	70.3	60.5	66.0	80.6	84.0

Each state and UT in the geojson file has a unique "state_code" and "st_nm":

```
In [6]: print(india_states['features'][0]['properties'])
print(india_states['features'][1]['properties'])
```

```
{'cartodb_id': 1, 'state_code': 0, 'st_nm': 'Telangana'}
{'cartodb_id': 2, 'state_code': 35, 'st_nm': 'Andaman & Nicobar Island'}
```

- Create an empty dictionary "state_id_map".

```
In [7]: state_id_map = {}
```

- Iterate through the geojson data and store "st_nm" as keys and respective "state_code" as the values of the created dictionary.

```
In [8]: for feature in india_states['features']:
    feature['id']=feature['properties'][ 'state_code']
    state_id_map[feature['properties'][ 'st_nm']] = feature['id']

state_id_map
```

```
Out[8]: {'Telangana': 0,
 'Andaman & Nicobar Island': 35,
 'Andhra Pradesh': 28,
 'Arunanchal Pradesh': 12,
 'Assam': 18,
 'Bihar': 10,
 'Chhattisgarh': 22,
 'Daman & Diu': 25,
 'Goa': 30,
 'Gujarat': 24,
 'Haryana': 6,
 'Himachal Pradesh': 2,
 'Jammu & Kashmir': 1,
 'Jharkhand': 20,
 'Karnataka': 29,
 'Kerala': 32,
 'Lakshadweep': 31,
 'Madhya Pradesh': 23,
 'Maharashtra': 27,
 'Manipur': 14,
 'Chandigarh': 4,
 'Puducherry': 34,
 'Punjab': 3,
 'Rajasthan': 8,
 'Sikkim': 11,
 'Tamil Nadu': 33,
 'Tripura': 16,
 'Uttar Pradesh': 9,
 'Uttarakhand': 5,
 'West Bengal': 19,
 'Odisha': 21,
 'Dadara & Nagar Haveli': 26,
 'Meghalaya': 17,
 'Mizoram': 15,
 'Nagaland': 13,
 'NCT of Delhi': 7}
```

- Now iterate through the literacy dataset and apply a function which takes *Country/ States/ Union Territories Name* as argument and pass it through the newly created dictionary for filling the "state_code" as "id".

```
In [9]: lit_data['id'] = lit_data['Country/ States/ Union Territories Name'].apply(lambda x: state_id[x])
lit_data.head()
```

Out[9]:

Category	Country/ States/ Union Territories Name	Literacy Rate (Persons) - Total - 2001	Literacy Rate (Persons) - Total - 2011	Literacy Rate (Persons) - Rural - 2001	Literacy Rate (Persons) - Rural - 2011	Literacy Rate (Persons) - Urban - 2001	Literacy Rate (Persons) - Urban - 2011	id
0	State Andhra Pradesh	60.5	67.0	54.5	60.4	76.1	80.1	28
1	State Arunanchal Pradesh	54.3	65.4	47.8	59.9	78.3	82.9	12
2	State Assam	63.3	72.2	59.7	69.3	85.3	88.5	18
3	State Bihar	47.0	61.8	43.9	59.8	71.9	76.9	10
4	State Chhattisgarh	64.7	70.3	60.5	66.0	80.6	84.0	22

```
In [10]: lit_data['growth'] = (lit_data['Literacy Rate (Persons) - Total - 2011'] - lit_data['Literacy Rate (Persons) - Total - 2001']) / lit_data['Literacy Rate (Persons) - Total - 2001']
lit_data.head()
```

Out[10]:

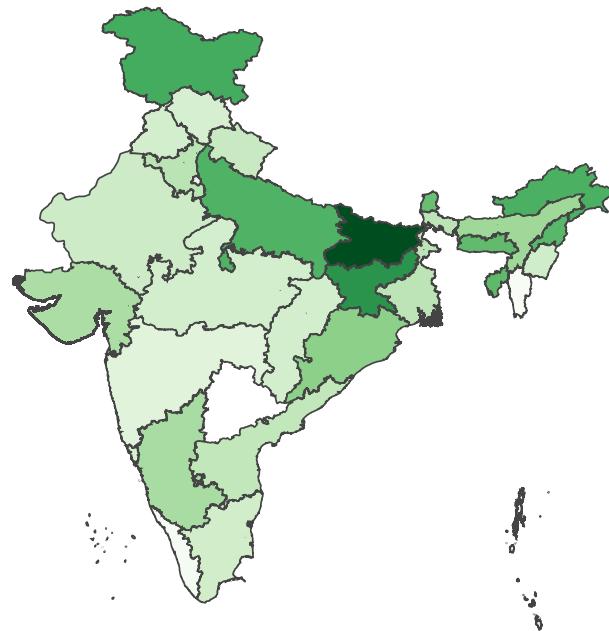
Category	Country/ States/ Union Territories Name	Literacy Rate (Persons) - Total - 2001	Literacy Rate (Persons) - Total - 2011	Literacy Rate (Persons) - Rural - 2001	Literacy Rate (Persons) - Rural - 2011	Literacy Rate (Persons) - Urban - 2001	Literacy Rate (Persons) - Urban - 2011	id	growth
0	State Andhra Pradesh	60.5	67.0	54.5	60.4	76.1	80.1	28	10.743802
1	State Arunanchal Pradesh	54.3	65.4	47.8	59.9	78.3	82.9	12	20.441989
2	State Assam	63.3	72.2	59.7	69.3	85.3	88.5	18	14.060032
3	State Bihar	47.0	61.8	43.9	59.8	71.9	76.9	10	31.489362
4	State Chhattisgarh	64.7	70.3	60.5	66.0	80.6	84.0	22	8.655332

Plotting the Data:

```
In [11]: fig = px.choropleth(lit_data, locations='id',
                           geojson=india_states,
                           color='growth',
                           color_continuous_scale='greens',
                           hover_name="Country/ States/ Union Territories Name",
                           width=1000, height=800)
```

fitbounds can be set to locations to automatically set the center and latitude and longitude range according to the data being plotted

```
In [12]: fig.update_geos(fitbounds='locations', visible=False)
fig.show()
```

**Note:**

Telangana didn't exist in our Dataframe(as it is pretty old) and that's the reason for blank area because the geojson is updated to the latest version.

```
In [13]: fig=px.histogram(lit_data,x="growth")
fig
```



```
In [14]: fig=px.histogram(lit_data,x="Literacy Rate (Persons) - Total - 2011", color = "Category")
fig
```



```
In [15]: # Create a dropdown widget for state and union territory
state_dropdown = widgets.Dropdown(options=lit_data['Category'].unique(), description='State')

# Define a function to update the histogram based on the selected state or union territory
def update_histogram(state):
    filtered_data = lit_data[lit_data['Category'] == state]
    fig = px.histogram(filtered_data, x="Literacy Rate (Persons) - Total - 2011", color="Category")
    fig.show()

# Register the update_histogram function as an event handler for the state dropdown
widgets.interact(update_histogram, state=state_dropdown)

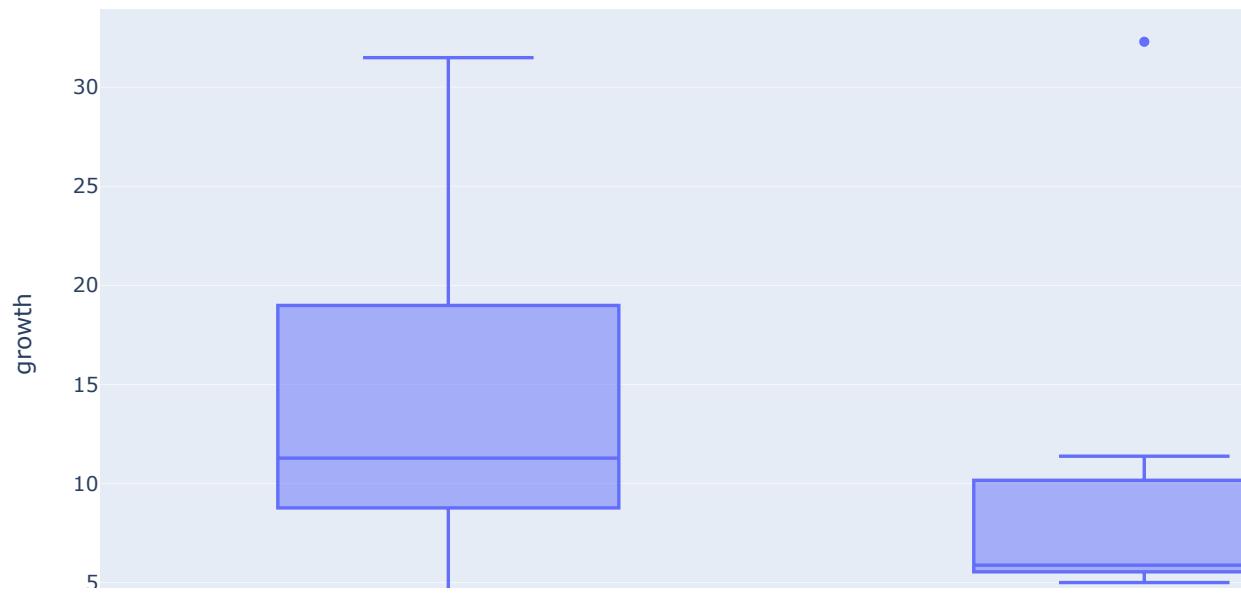
# Display the initial histogram
#initial_state = lit_data['Category'].unique()[0]
#update_histogram(initial_state)
```

State/UT: State

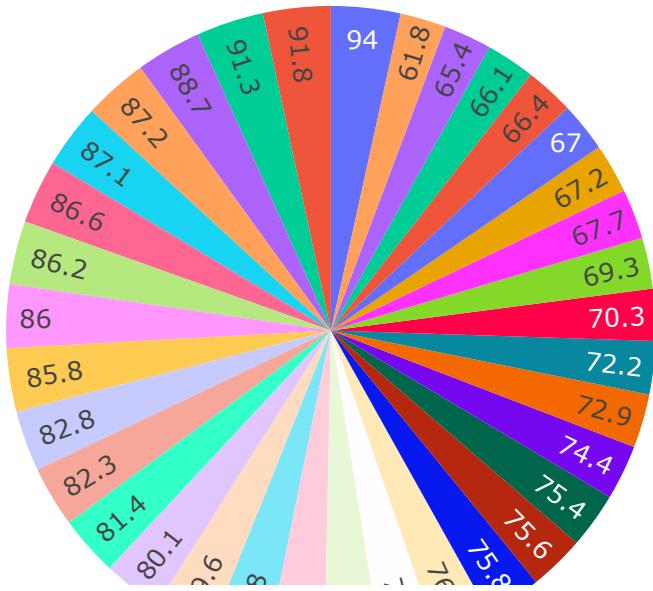


Out[15]: <function __main__.update_histogram(state)>

```
In [16]: fig= px.box(lit_data,x="Category" , y="growth")
fig
```



```
In [17]: fig = px.pie(lit_data, values='Literacy Rate (Persons) - Total - 2011', names='Country/ Stat  
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=15)  
fig.show()
```



In [18]:

```

# Set the center of the map to Asia
center_lat = 29.5
center_lon = 89.5

# Set the scope of the map to Asia
scope = 'asia'

# Create a scatter plot
fig = go.Figure()

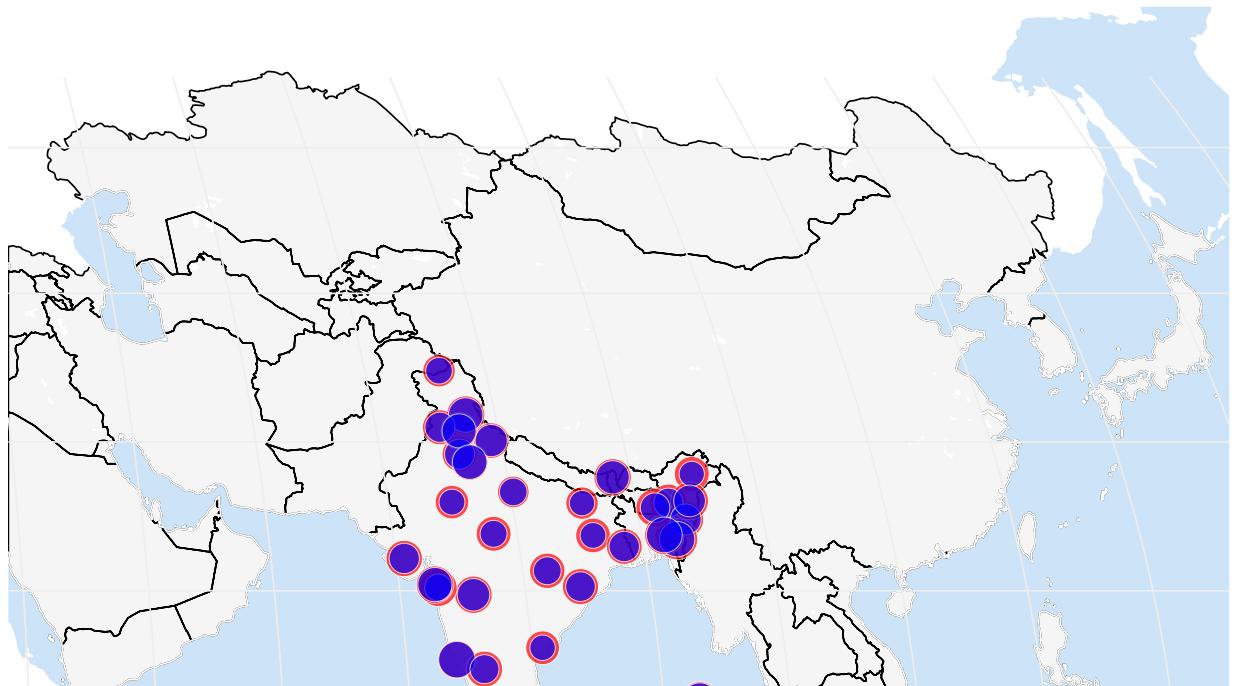
# Add scatter traces for 2001 and 2011 literacy rates
fig.add_trace(go.Scattergeo(
    locationmode='geojson-id', geojson= india_states ,
    marker=dict(
        size= lit_data['Literacy Rate (Persons) - Urban - 2011'],
        sizemode='diameter',
        sizeref=4, # Adjust the size of the markers
        color='red',
        opacity=0.7,
        line=dict(width=0.5, color='white')
    ),
    name='2011 Urabn Literacy Rate',
    locations= lit_data[ 'id']
))
fig.add_trace(go.Scattergeo(
    locationmode='geojson-id', geojson= india_states ,
    marker=dict(
        size= lit_data['Literacy Rate (Persons) - Rural - 2011'],
        sizemode='diameter',
        sizeref=4, # Adjust the size of the markers
        color='blue',
        opacity=0.7,
        line=dict(width=0.5, color='white')
    ),
    name='2011 Rural Literacy Rate',
    locations= lit_data[ 'id']
))

# Set the center and scope of the map
fig.update_geos(center=dict(lat=center_lat, lon=center_lon), projection_type='natural earth')

# Set the zoom level of the map
fig.update_layout(geo=dict(
    resolution=50,
    showcountries=True, countrycolor="black",
    showocean=True, oceancolor="#CDE3F8",
    showland=True, landcolor="#f5f5f5",
    showcoastlines=True, coastlinecolor="white",
    showframe=False,
    projection_type='natural earth',
    center=dict(lat=center_lat, lon=center_lon),
    scope=scope,
    lonaxis=dict(showgrid=True, gridwidth=1, range=[30, 120], dtick=10),
    lataxis=dict(showgrid=True, gridwidth=1, range=[0, 60], dtick=10)),
    margin=dict(l=0, r=0, t=0, b=0)
))

# Show the plot
fig.show()

```



In []: