Packages

In [1]:
```
#Basic packages
import pandas as pd
import numpy as np

#plotly packages
import plotly.express as px
import plotly.graph_objects as go

#json
import json
```

Loading and Manipulating Data

In [2]:
```
#Loading employee data
emp_data = pd.read_csv('employeedata.csv')

#Calculating median income and job statisfaction by department and gender
dept_gen_incm = pd.DataFrame(emp_data.groupby(['Department',
                                              'Gender'])[['MonthlyIncome',
                                                         'JobSatisfaction']].median()).reset_index()

#Calculating median income and job statisfaction by department and business travel
dept_tvl_incm = pd.DataFrame(emp_data.groupby(['Department',
                                              'BusinessTravel'])[['MonthlyIncome',
                                                                 'JobSatisfaction']].median()).reset_index()

#Calculating number of employees by business travel
bt_employee = pd.DataFrame(emp_data.groupby(['BusinessTravel'])['EmployeeCount'].count()).reset_index()
```

In [3]:
```
#Loading GDP data
gdp_data = px.data.gapminder()
```

In [4]:
```
#Loading Tip data
tip_data = px.data.tips()
```

In [5]:
```
#Loading covid data
covid = (pd.read_csv('covid_vaccine.csv')).dropna()

#Converting the dataframe into melt data for visualization purpose
covid_data = pd.melt(covid,
             id_vars = 'Updated On',
             value_vars = ['Male Vaccinated', 'Female Vaccinated'],
             var_name = 'Gender vaccination',
             value_name = 'Quantity')
```

In [6]:
```
#Loading vaccination data
vaccination_data = pd.read_csv('vaccinations.csv')

#Calculating total number of vaccination by department and Country
vaccination_data = pd.DataFrame(vaccination_data.groupby(['location',
                                                         'iso_code'])['daily_vaccinations'].sum()).reset_index()
```

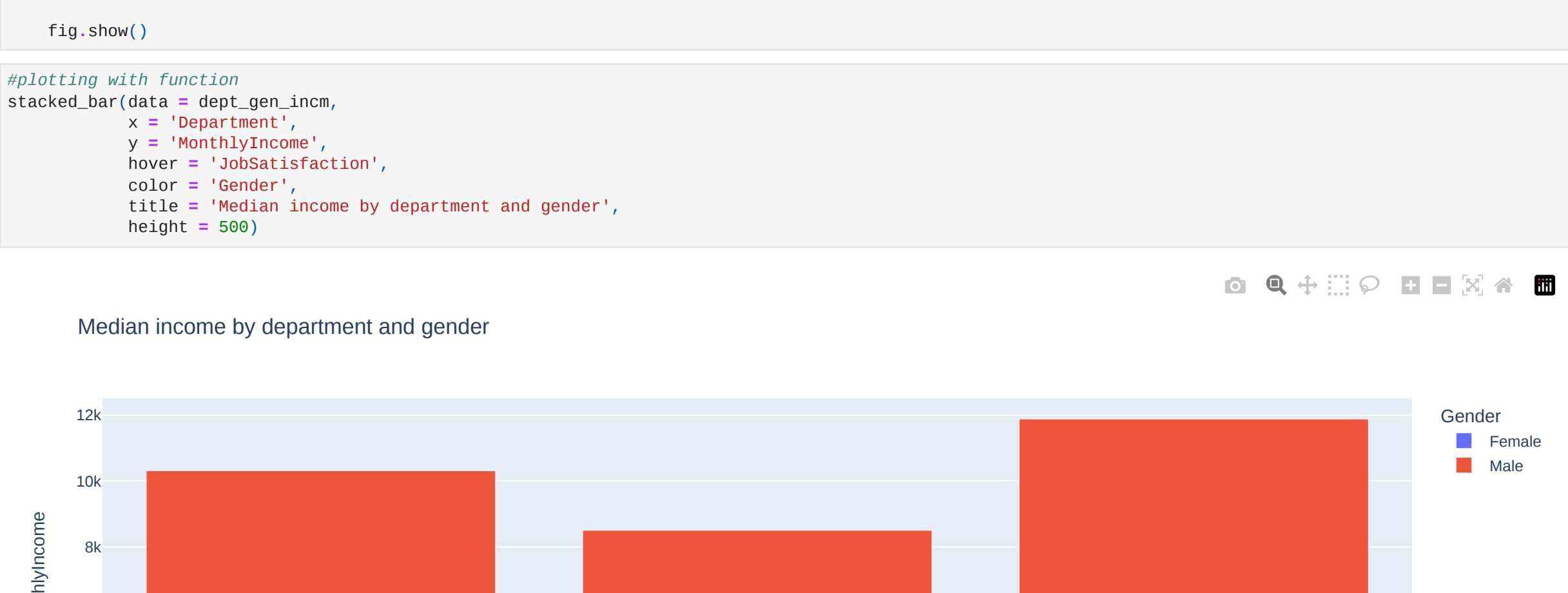
Plotting with Plotly

In [7]:
```
#Function
def stacked_bar(data, x, y, hover, color, title, height):

    fig = px.bar(data, x=x, y=y,
             hover_data=[hover], color=color,
             labels={y:'Median ' + y},
             title = title,
             height=height)

    fig.show()
```

In [8]:
```
#plotting with function
stacked_bar(data = dept_gen_incm,
            x = 'Department',
            y = 'MonthlyIncome',
            hover = 'JobSatisfaction',
            color = 'Gender',
            title = 'Median income by department and gender',
            height = 500)
```

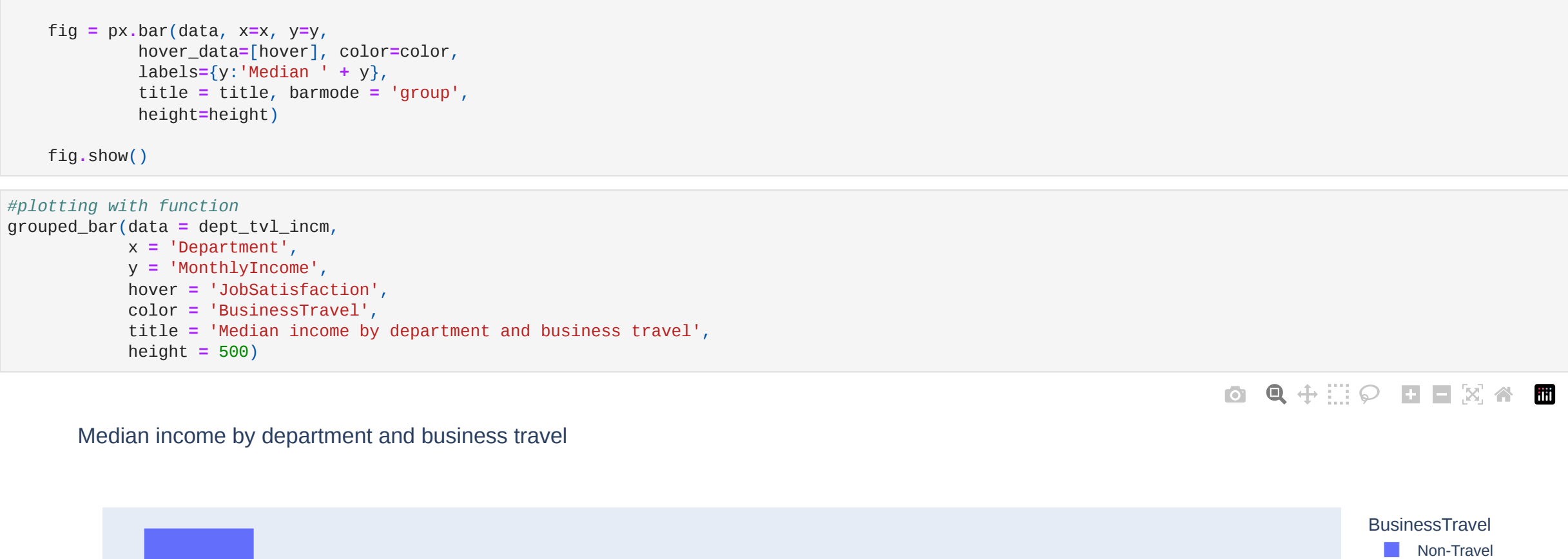


In [9]:
```
#Function
def grouped_bar(data, x, y, hover, color, title, height):

    fig = px.bar(data, x=x, y=y,
             hover_data=[hover], color=color,
             labels={y:'Median ' + y},
             title = title, barmode = 'group',
             height=height)

    fig.show()
```

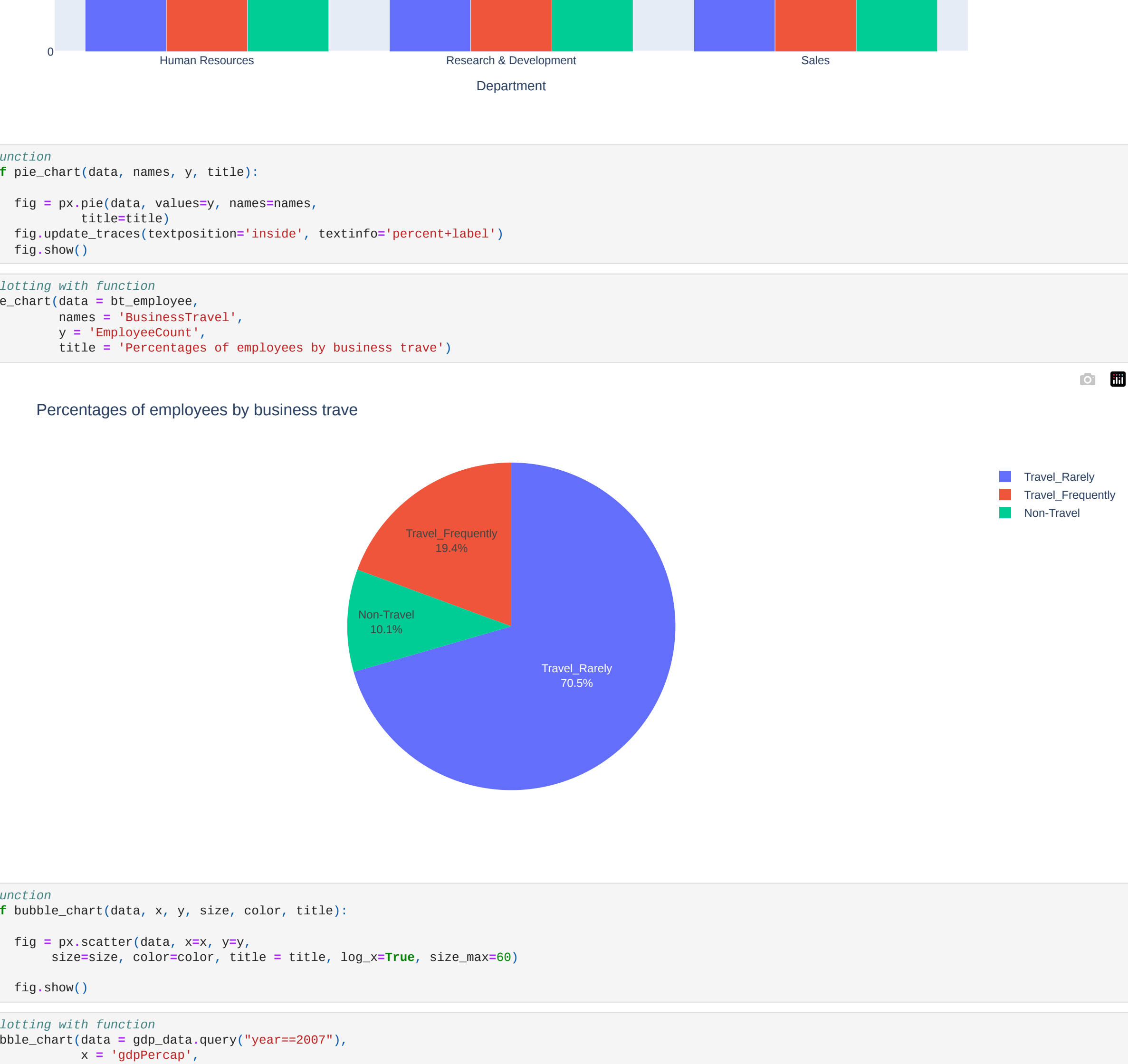
In [10]:
```
#plotting with function
grouped_bar(data = dept_tvl_incm,
            x = 'Department',
            y = 'MonthlyIncome',
            hover = 'JobSatisfaction',
            color = 'BusinessTravel',
            title = 'Median income by department and business travel',
            height = 500)
```



In [11]:
```
#Function
def pie_chart(data, names, y, title):

    fig = px.pie(data, values=y, names=names,
             title=title)
    fig.update_traces(textposition='inside', textinfo='percent+label')
    fig.show()
```

In [12]:
```
#plotting with function
pie_chart(data = bt_employee,
          names = 'BusinessTravel',
          y = 'EmployeeCount',
          title = 'Percentages of employees by business trave')
```

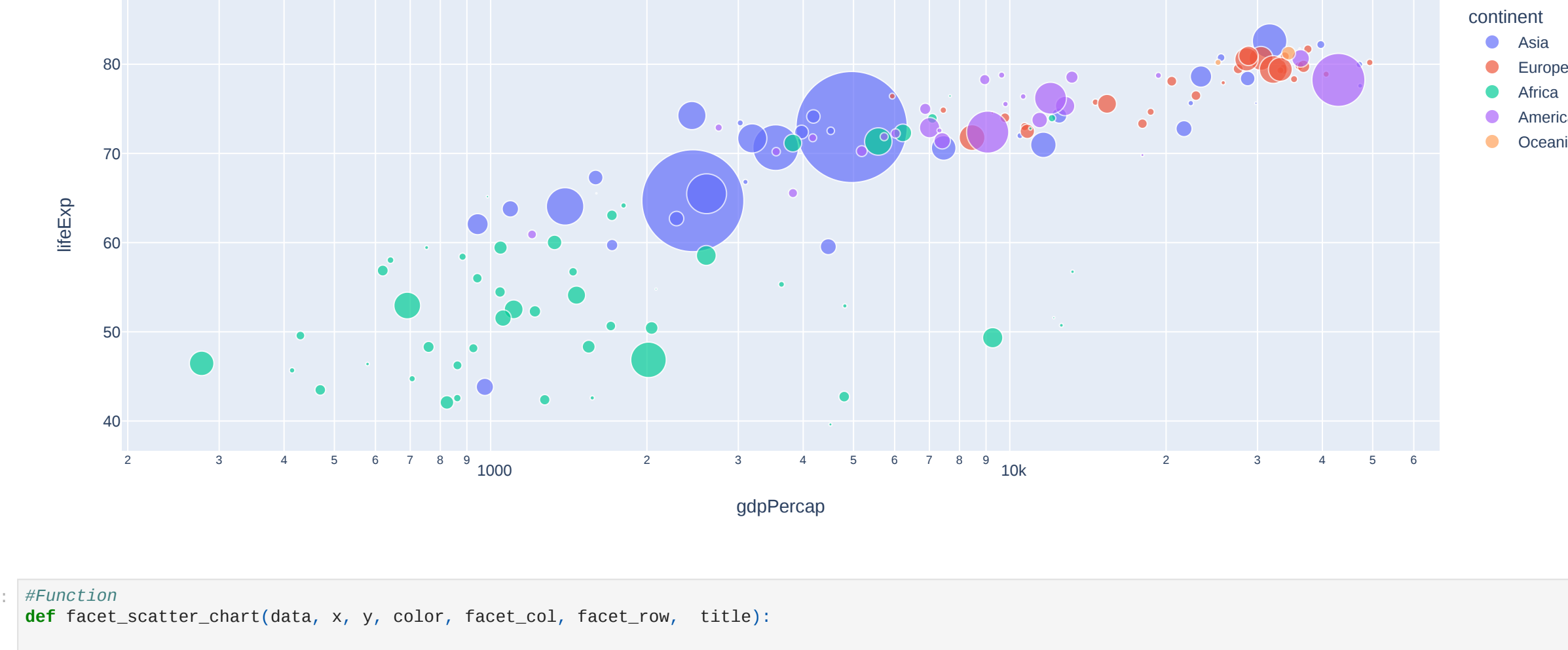


In [13]:
```
#Function
def bubble_chart(data, x, y, size, color, title):

    fig = px.scatter(data, x=x, y=y,
         size=size, color=color, title = title, log_x=True, size_max=60)

    fig.show()
```

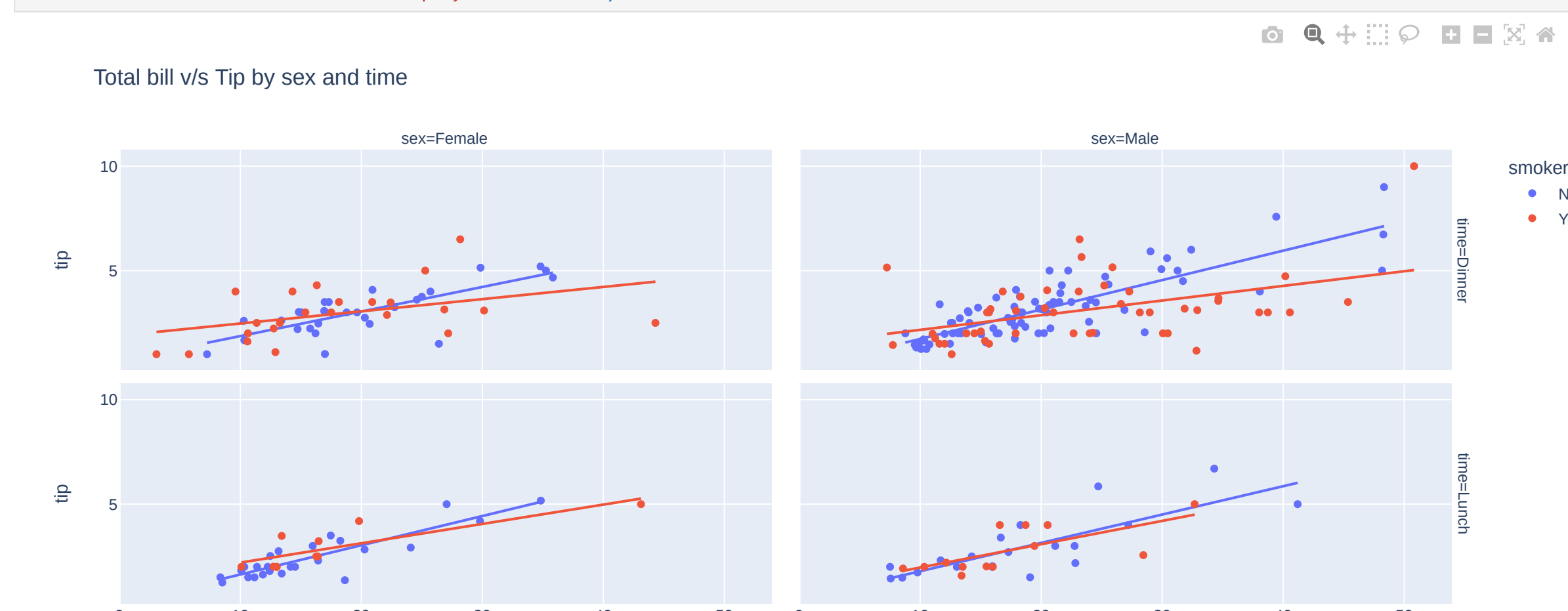
In [14]:
```
#plotting with function
bubble_chart(data = gdp_data.query("year==2007"),
             x = 'gdpPercap',
             y = 'lifeExp',
             size = 'pop',
             color = 'continent',
             title = 'GDP v/s Life Expectancy by Continent')
```



In [15]:
```
#Function
def facet_scatter_chart(data, x, y, color, facet_col, facet_row,  title):

    fig = px.scatter(data,
                     x=x,
                     y=y,
                     color=color,
                     facet_col=facet_col,
                     facet_row=facet_row, trendline='ols',
                     title = title)
    fig.show()
```

In [16]:
```
#plotting with function
facet_scatter_chart(data = tip_data,
             x = 'total_bill',
             y = 'tip',
             color = 'smoker',
             facet_col='sex',
             facet_row='time',
             title = 'Total bill v/s Tip by sex and time')
```



In [17]:
```
#Function
def line_chart(data, x, y, color, title):

    fig = px.line(data, x=x, y=y, color=color, title = title)

    fig.show()
```

In [18]:
```
#plotting with function
line_chart(data = covid_data,
          x = 'Updated On',
          y = 'Quantity',
          color = 'Gender vaccination',
          title = 'Number of vaccination trend by gender')
```

In [19]:
```
#Function
def box_plot(data, x, y, color, title):

    fig = px.box(data, x=x, y=y, color=color, title = title)
    fig.update_traces(quartilemethod="exclusive")

    fig.show()
```

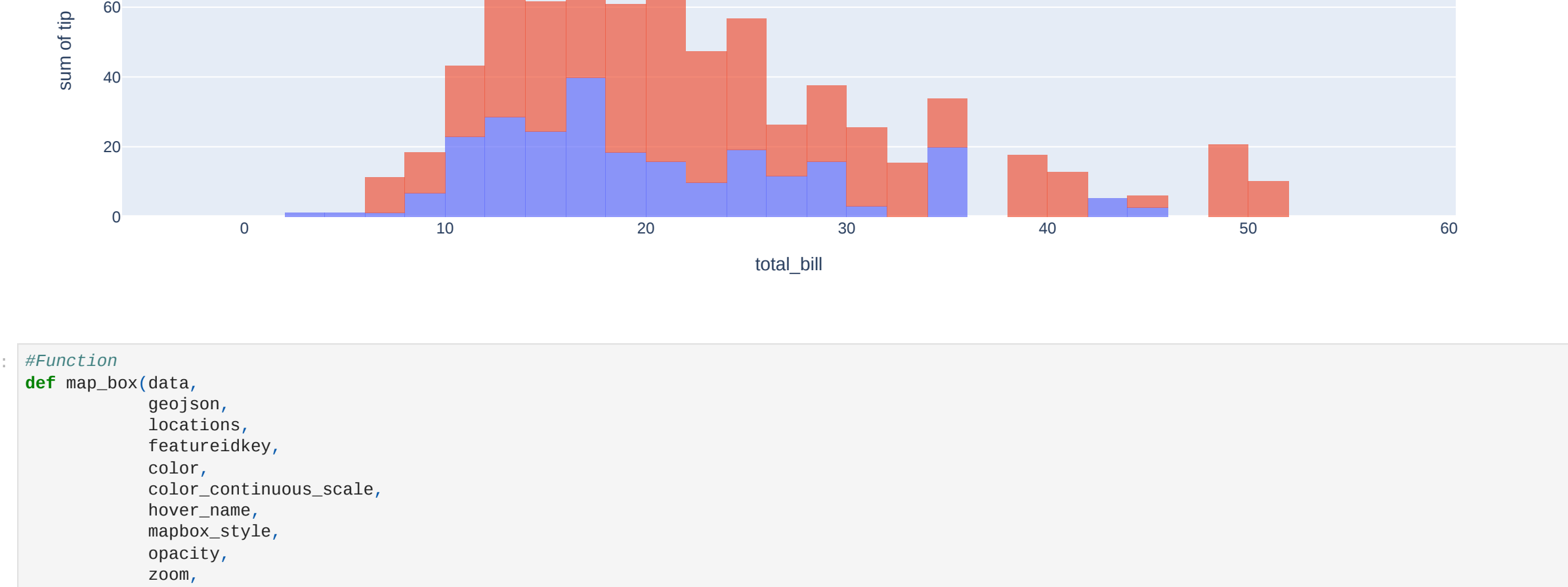
In [20]:
```
#plotting with function
box_plot(data = tip_data,
         x = 'day',
         y = 'total_bill',
         color = 'smoker',
         title = 'Summary statistics of total bill by day and smoker')
```

In [21]:
```
#Function
def histogram(data, x, y, color, title):

    fig = px.histogram(data, x=x, y=y, color=color, title = title,
                marginal='violin', opacity = 0.7) # or box, rug

    fig.show()
```

In [22]:
```
#plotting with function
histogram(data = tip_data,
         x = 'total_bill',
         y = 'tip',
         color = 'sex',
         title = 'Total bill distribution with total tip by genders')
```

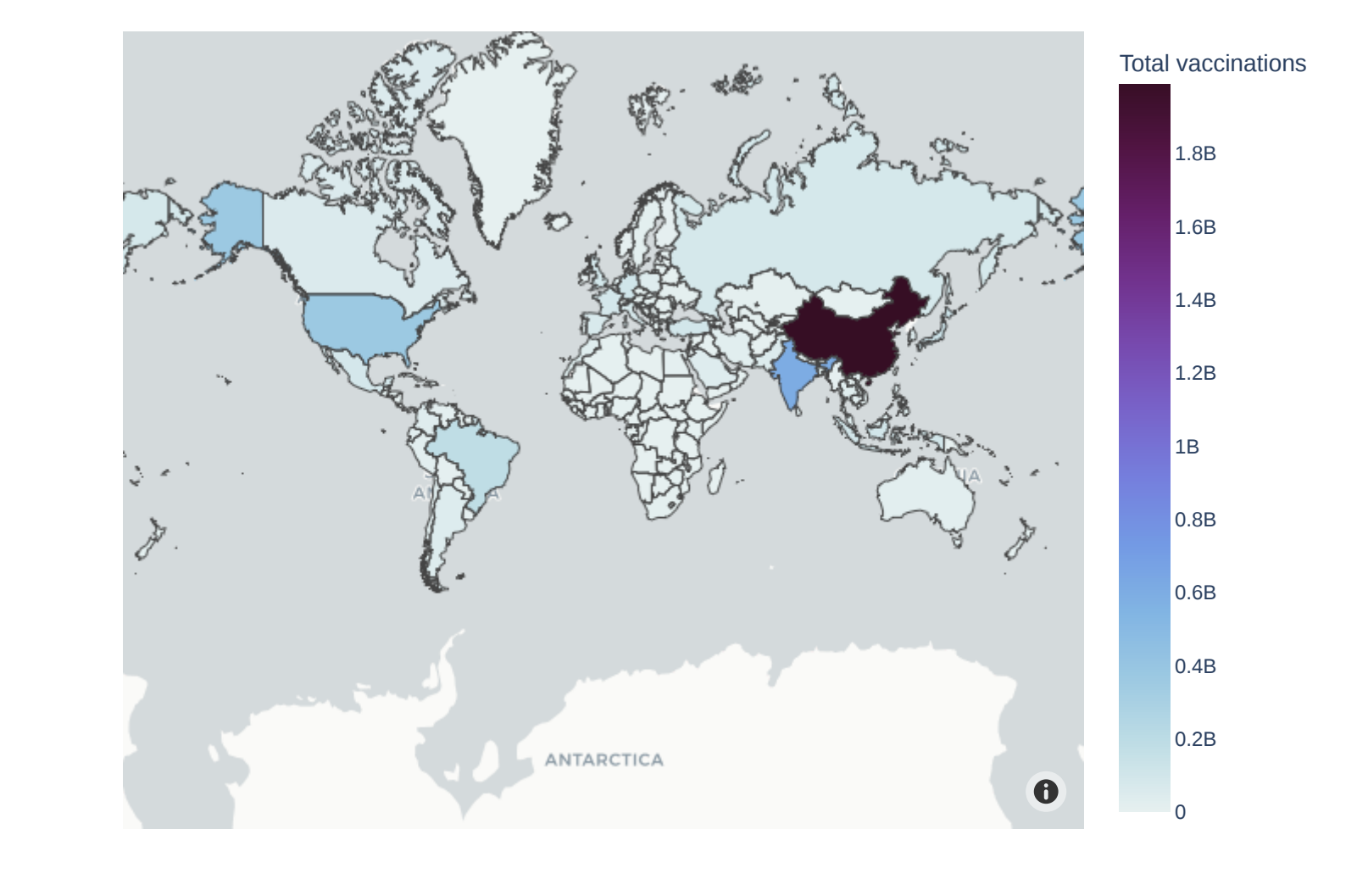


In [23]:
```
#Function
def map_box(data,
            geojson,
            locations,
            featureidkey,
            color,
            color_continuous_scale,
            hover_name,
            mapbox_style,
            opacity,
            zoom,
            height,
            width,
            title):

    fig = px.choropleth_mapbox(data, geojson=geojson,
                           locations=locations,
                           featureidkey=featureidkey,
                           color=color,
                           color_continuous_scale=color_continuous_scale,
                           hover_name = hover_name,
                           range_color=(np.min(data[color]), np.max(data[color])),
                           mapbox_style=mapbox_style,
                           opacity = opacity,
                           zoom = zoom,
                            title = title,
                            labels = {color: 'Total vaccinations'})

    fig.update_layout(height = height, width = width)
    fig.show()
```

In [24]:
```
#plotting with function
map_box(data = vaccination_data,
        geojson = 'https://raw.githubusercontent.com/datasets/geo-countries/master/data/countries.geojson',
        locations = 'iso_code',
        featureidkey = 'properties.ISO_A3',
        color = 'daily_vaccinations',
        color_continuous_scale = 'dense',
        hover_name = 'location',
        mapbox_style = 'carto-positron',
        opacity = 1,
        zoom = 0,
        height = 650,
        width = 800,
        title = 'Total vaccination by countries')
```



Thank you!