**a. Visualize Gujarat state’s population:**

*#importing libraries*

import folium

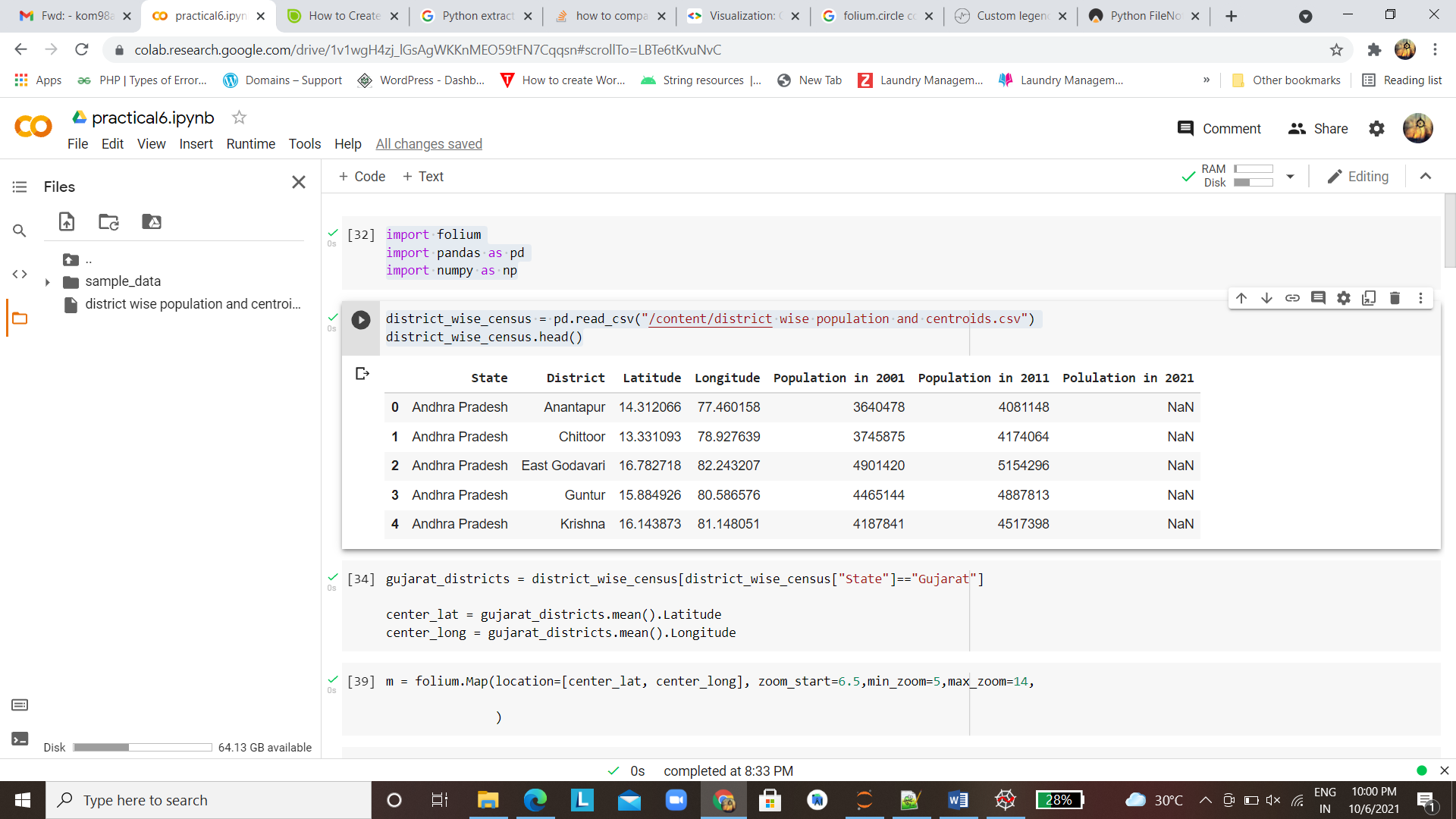
import pandas as pd

import numpy as np

*#reading the csv files which contains the population data of India*

district\_wise\_census = pd.read\_csv("/content/district wise population and centroids.csv")

district\_wise\_census.head()



*#select the state as Gujarat and their Latitude and Longitude*

gujarat\_districts = district\_wise\_census[district\_wise\_census["State"]=="Gujarat"]

center\_lat = gujarat\_districts.mean().Latitude

center\_long = gujarat\_districts.mean().Longitude

*#plotting the map*

m = folium.Map(location=[center\_lat, center\_long], zoom\_start=6.5)

m

for district in gujarat\_districts["District"].unique():

  district\_census = gujarat\_districts[gujarat\_districts["District"]==district]

*#adding the circle according to population of city in Gujarat state*

  folium.Circle(

    location=[district\_census.Latitude.values[0], district\_census.Longitude.values[0]],

    radius = float(district\_census["Population in 2011"].values[0]/100),

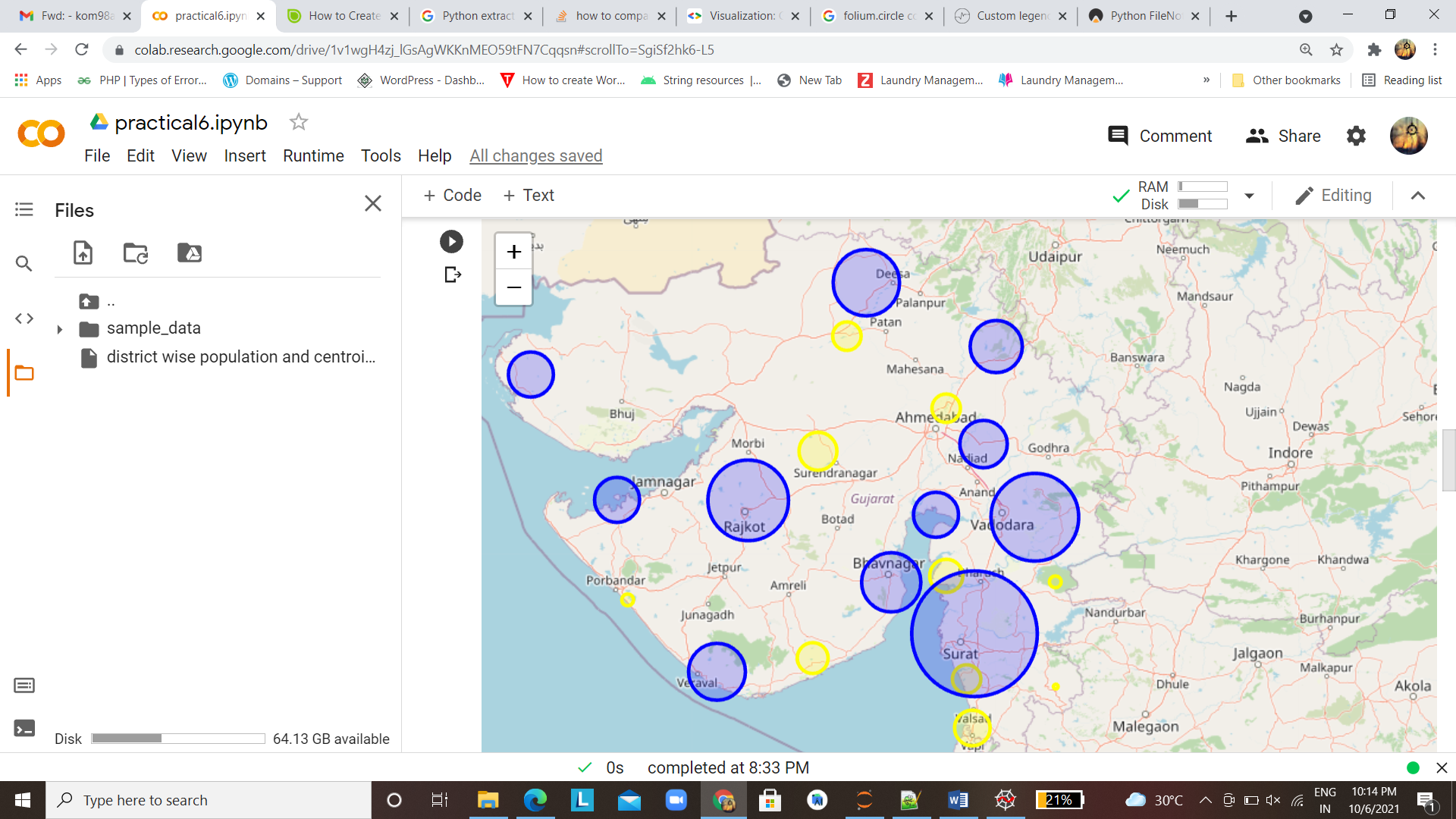
    color="yellow" if(district\_census["Population in 2011"].values<2000000) else"blue",

    fill\_color="yellow"if(district\_census["Population in 2011"].values<2000000)else"blue",

    popup="Population 2011 : %s"%district\_census["Population in 2011"].values[0],

    tooltip = district\_census.District.values[0],

    ).add\_to(m)



**b. State wise literacy rate of India:**

*#importing libraries*

import json

import numpy as np

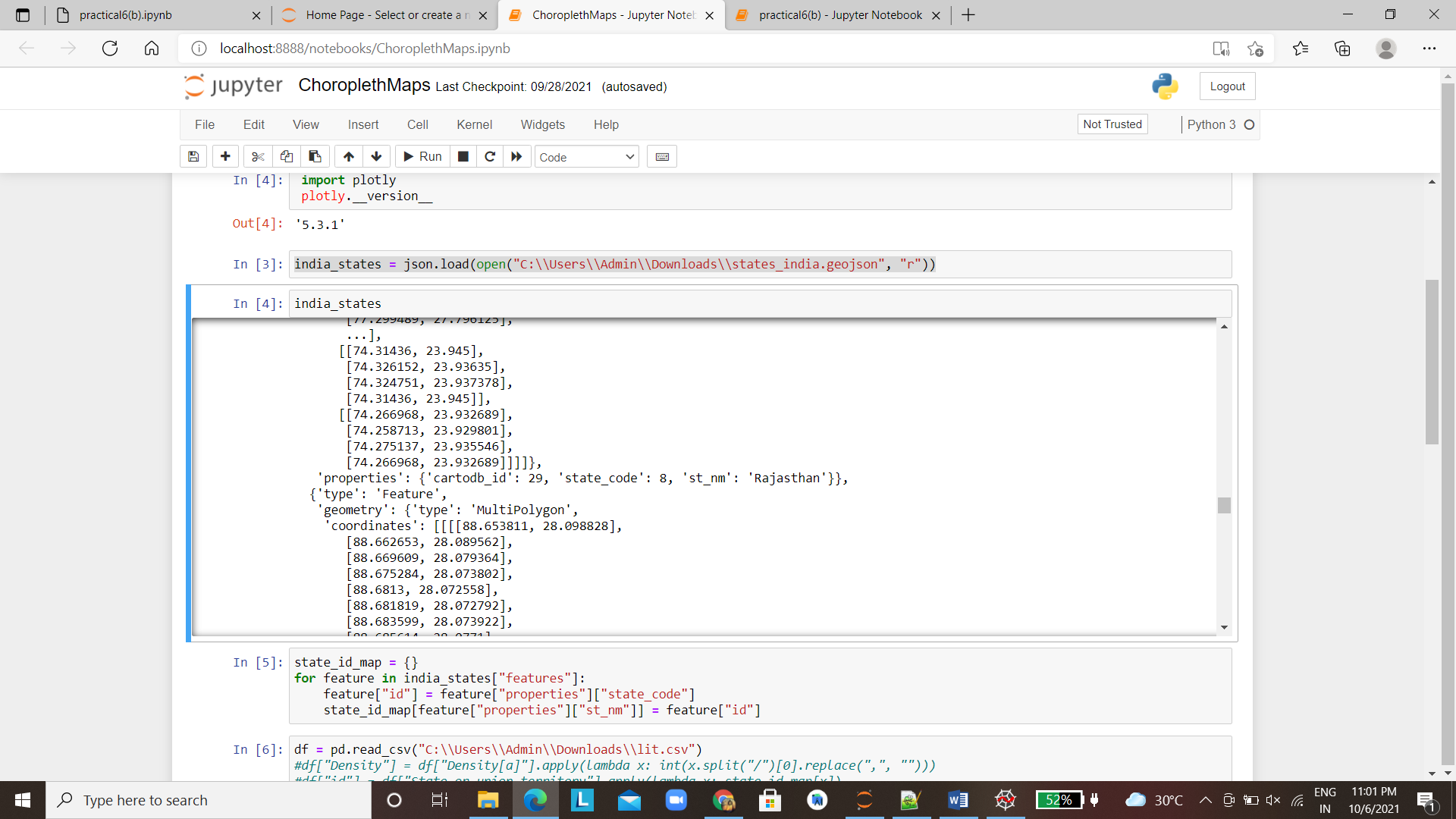
import pandas as pd

import plotly.express as px

*#reading the Geojson file*

india\_states = json.load(open("C:\\Users\\Admin\\Downloads\\states\_india.geojson", "r"))

india\_states



*#from the geojson file making the feature is of state*

state\_id\_map = {}

for feature in india\_states["features"]:

feature["id"] = feature["properties"]["state\_code"]

state\_id\_map[feature["properties"]["st\_nm"]] = feature["id"]

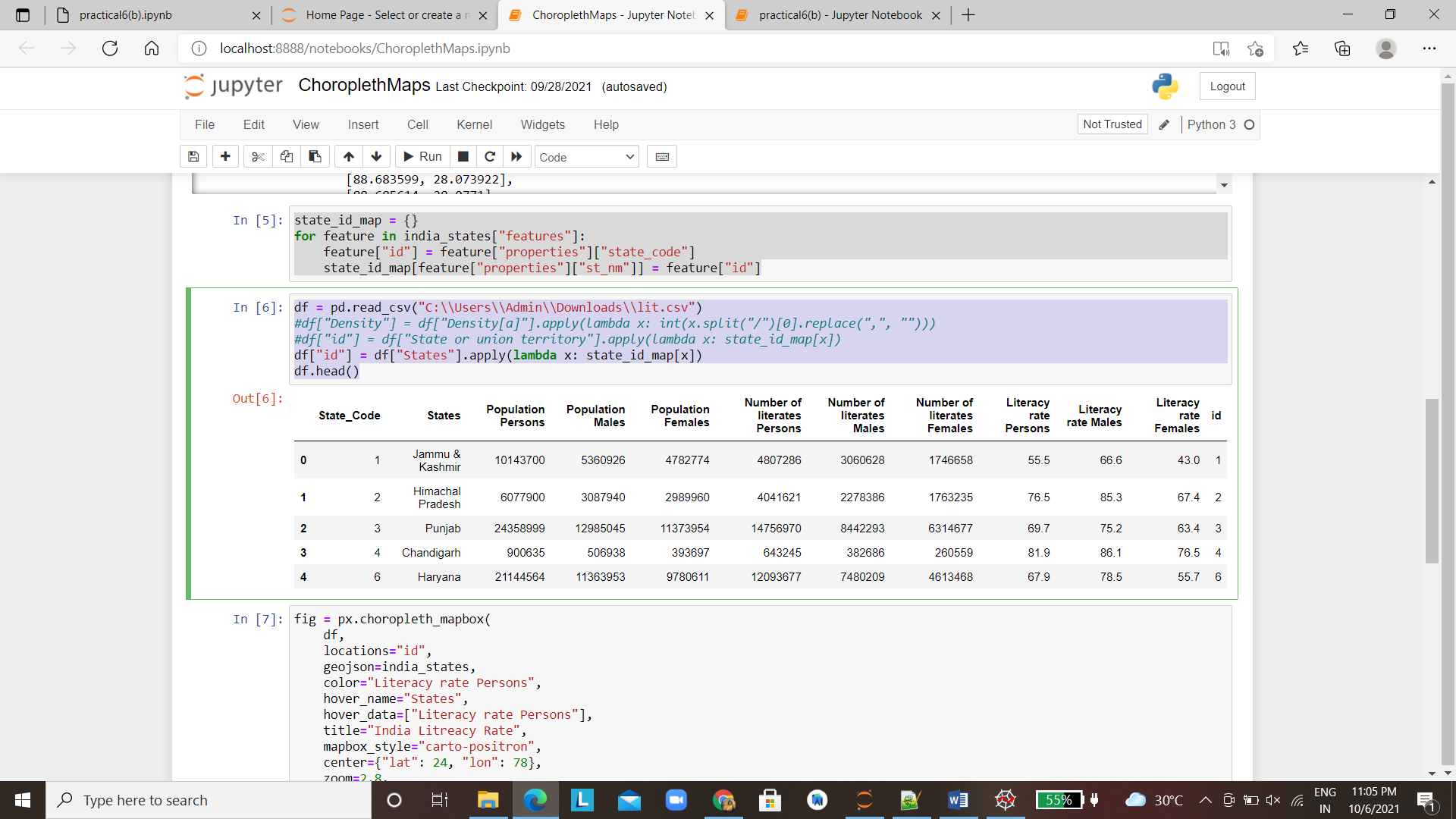
*#reading the csv file having literacy data of india*

df = pd.read\_csv("C:\\Users\\Admin\\Downloads\\lit.csv")

*#mapping the states of geojson file and csv file*

df["id"] = df["States"].apply(lambda x: state\_id\_map[x])

df.head()



*#plotting the map*

fig = px.choropleth\_mapbox(

df,

locations="id",

geojson=india\_states,

color="Literacy rate Persons",

hover\_name="States",

hover\_data=["Literacy rate Persons"],

title="India Litreacy Rate",

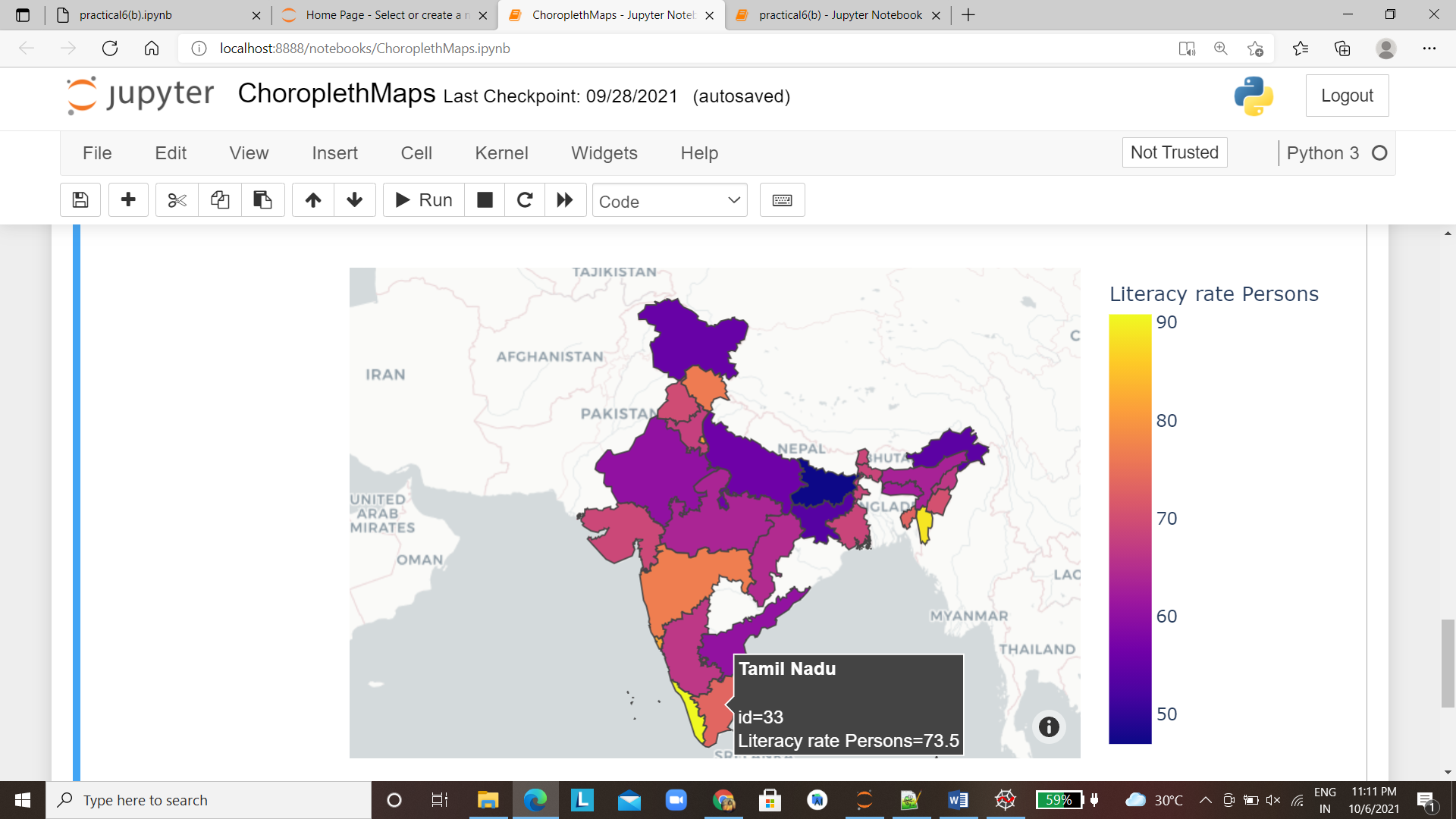
mapbox\_style="carto-positron",

center={"lat": 24, "lon": 78},

zoom=2.8,

)

fig.show()

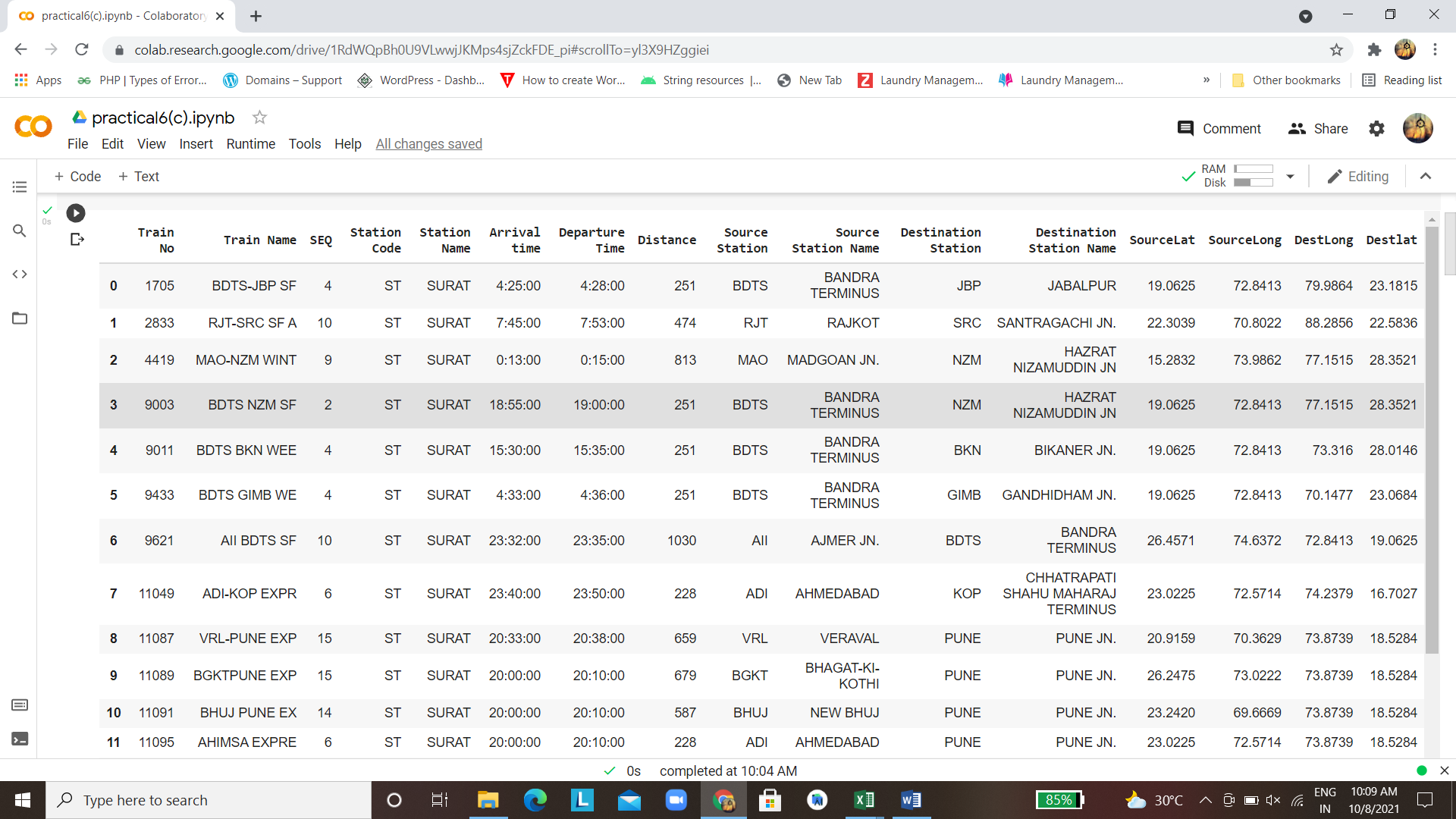


**c.** **Railway lines in Gujarat**

#reading the csv file

import pandas as pd

df = pd.read\_csv("/content/Indian\_railway1.csv",encoding="latin1",nrows=50)



#importing libraries

import pandas as pd

import geopandas as gpd

import matplotlib.pyplot as plt

import plotly.graph\_objects as go

fig=go.Figure()

source\_to\_dest = zip(df["SourceLat"], df["Destlat"],

                     df["SourceLong"], df["DestLong"],

                     df["Distance"])

*# Loop thorugh each station entry to add line between source and destination*

for slat,dlat, slon, dlon, distance in source\_to\_dest:

    fig.add\_trace(go.Scattergeo(

                      lat = [slat,dlat],

                      lon = [slon, dlon],

                      mode = 'lines',

                      line = dict(color='#111021', width=1,dash='dash')

                        ))

*# Logic to create labels of source and destination cities of stations*

cities =df["Source Station Name"].values.tolist()+df["Destination Station Name"].values.tolist()

scatter\_hover\_data = [city for city in cities]

*#Loop thorugh each station entry to plot source and destination as points.*

fig.add\_trace(

    go.Scattergeo(

                lon = df["SourceLong"].values.tolist()+df["DestLong"].values.tolist(),

                lat = df["SourceLat"].values.tolist()+df["Destlat"].values.tolist(),

                text = scatter\_hover\_data,

                textposition='top right',

                textfont=dict(color='#f0251a'),

                mode = 'markers+text',

                marker = dict(size = 10, color = 'green', opacity=0.1,))

    )

*# Update graph layout to improve graph styling.*

fig.update\_layout(

    height=500, width=800, margin={"t":0,"b":0,"l":0, "r":0, "pad":0},

    showlegend=False,

    ,

    geo = dict(projection\_type = 'natural earth',scope = 'asia'),

                )

fig.show()

