

```
In [1]: import pandas as pd
import json
import plotly.express as px
import os
```

```
In [2]: file=open(r"G:\guvi work\pulse-master\pulse-master\data\aggregated\transaction\country\india\2018\1.json", "r")
datas1=json.load(file)
datas1
```

```
Out[2]: {'success': True,
'code': 'SUCCESS',
'data': {'from': 1514745000000,
'to': 1522175400000,
'transactionData': [{'name': 'Recharge & bill payments',
'paymentInstruments': [{'type': 'TOTAL',
'count': 72550406,
'amount': 14472713558.652578}]},
{'name': 'Peer-to-peer payments',
'paymentInstruments': [{'type': 'TOTAL',
'count': 46982705,
'amount': 147245883542.77402}]},
{'name': 'Merchant payments',
'paymentInstruments': [{'type': 'TOTAL',
'count': 5368669,
'amount': 4656678915.140091}]},
{'name': 'Financial Services',
'paymentInstruments': [{'type': 'TOTAL',
'count': 3762820,
'amount': 815853105.1000277}]},
{'name': 'Others',
'paymentInstruments': [{'type': 'TOTAL',
'count': 5761576,
'amount': 4643217301.269438}]}],
'responseTimestamp': 1630346628866}
```

```
In [4]: #program(1)(1)
import pandas as pd
import os
import json
dic_yearbased={"year":[], "quater":[], "Transaction_type":[], "Transaction_count":[], "Transaction_amount":[]}
dic_state_yearbased={"year":[], "quater":[], "Transaction_type":[], "Transaction_count":[], "Transaction_amount":[]}

data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\aggregated\transaction\country\india")
common_data=os.listdir(data_aggre_trans_path)
print(common_data)
for i in common_data:
    if i.isnumeric()==True:
        #if i.isnumeric()==True:
        year=data_aggre_trans_path+'/'+i+'/'
        #print("year:", year)
        json_path=os.listdir(year)
        # print("json_path:", json_path)
        for onejson in json_path:
            files=year+onejson
            data=open(files, "r")
            final_data=json.load(data)
            #print("final_data", final_data )
            for datas in final_data["data"]["transactionData"]:
                name=datas["name"]
                count=datas["paymentInstruments"][0]["count"]
                amount=datas["paymentInstruments"][0]["amount"]
                dic_yearbased["Transaction_type"].append(name)
                dic_yearbased["Transaction_count"].append(count)
                dic_yearbased["Transaction_amount"].append(amount)
                dic_yearbased["quater"].append(int(onejson.strip(".json")))
                dic_yearbased["year"].append(i)
                print(dic_yearbased)

##print(Agg_Trans_only_year)

if i.isnumeric()==False:
    state=data_aggre_trans_path+"/"+i
    all_state=os.listdir(state)
    #print(all_state)
    for j in all_state:
        onebyone_state=state+"/"+j
        onebyone_state_year=os.listdir(onebyone_state)
        # print("onebyone_state_year:", onebyone_state_year)
        for y in onebyone_state_year:
            onebyone_state_year_json_path= onebyone_state+"/"+y
```

```

# print("onebyone_state_year_json_path:", onebyone_state_year_json_path)
onebyone_state_year_json_pos=os.listdir(onebyone_state_year_json_path)
#print("onebyone_state_year_json_pos:", onebyone_state_year_json_pos)
for onejson in onebyone_state_year_json_pos:
# print("onejson:", onejson )
onebyone_state_year_json_pos_weneed=onebyone_state_year_json_path+"/"+onejson
files_open=open(onebyone_state_year_json_pos_weneed,"r")
files_final=json.load(files_open)
#print(files_final)
for data in files_final["data"]["transactionData"]:
    name=data["name"]
    count=data['paymentInstruments'][0]["count"]
    amount=data['paymentInstruments'][0]["amount"]

# print(name)
# print(count)
# print(amount)

    dic_state_yearbased["Transaction_amount"].append(amount)
    dic_state_yearbased["Transaction_type"].append(name)
    dic_state_yearbased["Transaction_count"].append(count)
    dic_state_yearbased["year"].append(y)
    dic_state_yearbased["quater"].append(int(onejson.strip(".json")))
    dic_state_yearbased["state"].append(j)

```

In [4]: Agg_Trans_only_year

Out[4]:

	year	quater	Transaction_type	Transaction_count	Transaction_amount
0	2018	1	Recharge & bill payments	72550406	1.447271e+10
1	2018	1	Peer-to-peer payments	46982705	1.472459e+11
2	2018	1	Merchant payments	5368669	4.656679e+09
3	2018	1	Financial Services	3762820	8.158531e+08
4	2018	1	Others	5761576	4.643217e+09
...
95	2022	4	Merchant payments	6565690145	3.673601e+12
96	2022	4	Peer-to-peer payments	4459859921	1.457887e+13
97	2022	4	Recharge & bill payments	991049281	6.497768e+11
98	2022	4	Financial Services	4623387	5.360528e+09
99	2022	4	Others	10013085	7.361493e+09

100 rows × 5 columns

In [5]: Agg_Trans_state_year

Out[5]:

	year	quater	Transaction_type	Transaction_count	Transaction_amount	state
0	2018	1	Recharge & bill payments	4200	1.845307e+06	andaman-&-nicobar-islands
1	2018	1	Peer-to-peer payments	1871	1.213866e+07	andaman-&-nicobar-islands
2	2018	1	Merchant payments	298	4.525072e+05	andaman-&-nicobar-islands
3	2018	1	Financial Services	33	1.060142e+04	andaman-&-nicobar-islands
4	2018	1	Others	256	1.846899e+05	andaman-&-nicobar-islands
...
3589	2022	4	Peer-to-peer payments	184380244	6.202222e+11	west-bengal
3590	2022	4	Merchant payments	171667404	1.408077e+11	west-bengal
3591	2022	4	Recharge & bill payments	48921147	2.602663e+10	west-bengal
3592	2022	4	Financial Services	268388	2.611229e+08	west-bengal
3593	2022	4	Others	610414	4.579379e+08	west-bengal

3594 rows × 6 columns

In [7]:

```

#step 1
#program for user trans
# #program(1)(2)
import pandas as pd
import os
import json

dic_user_yearbased={"year":[],"quater":[],"brand_name":[],"brand_count":[],"Percentage":[]}

```

```

dic_user_yearbased1={"reg_users":[],"app_opens":[],"year":[],"quater":[]}
dic_user_year_state_brand={"year":[],"quater":[],"brand_name":[],"brand_count":[],"Percentage":[],"state":[]}
dic_user_year_state_reg_app={"reg_users":[],"app_opens":[],"year":[],"quater":[],"state":[]}

data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\aggregated\user\country\india")
common_data=os.listdir(data_aggre_trans_path)
print("common_data:", common_data)
for i in common_data:
    print("i:", i)
    if i.isnumeric()==True:
        #if i.isnumeric()==True:
            year=data_aggre_trans_path+'/'+i+'/'
            print("year:", year)
            json_path=os.listdir(year)
            print("json_path:", json_path)
            for onejson in json_path:
                files=year+onejson
                data=open(files,"r")
                final_data=json.load(data)
                print("final_data", final_data )
            #for datas in final_data["data"]:
                dic_user_yearbased1["year"].append(i)
                dic_user_yearbased1["quater"].append(int(onejson.strip(".json")))
                reg_users=final_data["data"]["aggregated"]["registeredUsers"]
                app_opens=final_data["data"]["aggregated"]["appOpens"]
                dic_user_yearbased1["reg_users"].append(reg_users)
                print("[reg_users:] ",dic_user_yearbased1["reg_users"])
                print("[app_opens:] ",dic_user_yearbased1["app_opens"])
                dic_user_yearbased1["app_opens"].append(app_opens)

            print("registeredUsers", reg_users)
            print("app_opens:", app_opens)
            try:
                for data in final_data["data"]["usersByDevice"]:
                    name=data["brand"]

                    count=data["count"]
                    percentage=data["percentage"]

                    dic_user_yearbased["brand_count"].append(count)
                    dic_user_yearbased["Percentage"].append(percentage)
                    dic_user_yearbased["brand_name"].append(name)
                    dic_user_yearbased["quater"].append(int(onejson.strip(".json")))
                    dic_user_yearbased["year"].append(i)
            except:
                pass

#data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\aggregated\user\country\india")
# dic_user_year_state_brand={"year":[],"quater":[],"brand_name":[],"brand_count":[],"Percentage":[],"state":[]}
# dic_user_year_state_reg_app={"reg_users":[],"app_opens":[],"year":[],"quater":[],"state":[]}
    if i.isnumeric()==False:
        state=data_aggre_trans_path+"/"+i
        all_state=os.listdir(state)
        #print(all_state)
        for j in all_state:
            onebyone_state=state+"/"+j
            onebyone_state_year=os.listdir(onebyone_state)
            # print("onebyone_state_year:",onebyone_state_year)
            for y in onebyone_state_year:
                onebyone_state_year_json_path= onebyone_state+"/"+y
                # print("onebyone_state_year_json_path:", onebyone_state_year_json_path)
                onebyone_state_year_json_pos=os.listdir(onebyone_state_year_json_path)
                #print("onebyone_state_year_json_pos:", onebyone_state_year_json_pos)
                for onejson in onebyone_state_year_json_pos:
                    # print("onejson:", onejson )
                    onebyone_state_year_json_pos_weneed=onebyone_state_year_json_path+"/"+onejson
                    files_open=open(onebyone_state_year_json_pos_weneed,"r")
                    files_final=json.load(files_open)
                    print(files_final)
                    reg_users=files_final["data"]["aggregated"]["registeredUsers"]
                    app_opens=files_final["data"]["aggregated"]["appOpens"]
                    dic_user_year_state_reg_app["reg_users"].append(reg_users)
                    dic_user_year_state_reg_app["app_opens"].append(app_opens)
                    dic_user_year_state_reg_app["year"].append(y)
                    dic_user_year_state_reg_app["quater"].append(int(onejson.strip(".json")))
                    dic_user_year_state_reg_app["state"].append(j)
                # print(dic_user_year_state_reg_app)
                try:
                    for data in files_final["data"]["usersByDevice"]:
                        name=data["brand"]
                        count=data["count"]
                        percentage=data["percentage"]
                        dic_user_year_state_brand["year"].append(y)

```

```

dic_user_year_state_brand["quater"].append(int(onejson.strip(".json")))
dic_user_year_state_brand["brand_name"].append(name)
dic_user_year_state_brand["brand_count"].append(count)
dic_user_year_state_brand["Percentage"].append(percentage)
dic_user_year_state_brand["state"].append(j)

except:
    pass

#dic_user_year_state_brand={"year":[],"quater":[],"brand_name":[],"brand_count":[],"Percentage":[],"state":[]}

#
#
# usersByDevice': [{'brand': 'Xiaomi', 'count': 1665, 'percentage': 0.2470326409495549},
#                   {'brand': 'Samsung', 'count': 1445, 'percentage': 0.21439169}
#
#   for data in files_final["data"]["transactionData"]:
#       name=data["name"]
#       count=data['paymentInstruments'][0]["count"]
#       amount=data['paymentInstruments'][0]["amount"]
#       # print(name)
#       # print(count)
#       # print(amount)
#
#       dic_state_yearbased["Transaction_amount"].append(amount)
#       dic_state_yearbased["Transaction_type"].append(name)
#       dic_state_yearbased["Transaction_count"].append(count)
#       dic_state_yearbased["year"].append(y)
#       dic_state_yearbased["quater"].append(int(onejson.strip(".json")))
#       dic_state_yearbased["state"].append(i)
#program first 2 line
#dic_yearbased={ "year":[],"quater":[],"Transaction_type":[],"Transaction_count":[],"Transaction_amount":[]}
#dic_state_yearbased={"year":[], "quater":[], "Transaction_type":[], "Transaction_amount":[], "state":[]}

#Agg_Trans_only_year=pd.DataFrame(dic_yearbased)
Agg_Trans_only_year=pd.DataFrame(dic_yearbased)
Agg_Trans_state_year_brand1=pd.DataFrame(dic_user_year_state_brand)
Agg_Trans_state_year_reguser_app_opens1=pd.DataFrame(dic_user_year_state_reg_app)
Agg_user_years1=pd.DataFrame(dic_user_yearbased)
Agg_reg_appopens1=pd.DataFrame(dic user yearbased1)
```

Out[9]:

6732 rows × 6 columns

Out[10]:

	reg_users	app_opens	year	quater	state
0	6740	0	2018	1	andaman-&-nicobar-islands
1	9405	0	2018	2	andaman-&-nicobar-islands
2	12149	0	2018	3	andaman-&-nicobar-islands
3	15222	0	2018	4	andaman-&-nicobar-islands
4	18596	0	2019	1	andaman-&-nicobar-islands
...
715	20644527	512335839	2021	4	west-bengal
716	21919787	236131065	2022	1	west-bengal
717	23124388	250276369	2022	2	west-bengal
718	24372048	256445748	2022	3	west-bengal
719	25536381	678066327	2022	4	west-bengal

720 rows × 5 columns

In [11]: Agg_user_years1

Out[11]:

	year	quater	brand_name	brand_count	Percentage
0	2018	1	Xiaomi	11926334	0.254413
1	2018	1	Samsung	9609401	0.204988
2	2018	1	Vivo	5894293	0.125737
3	2018	1	Oppo	4479351	0.095554
4	2018	1	Realme	2376866	0.050703
...
182	2022	1	OnePlus	7974546	0.021382
183	2022	1	Motorola	5610140	0.015043
184	2022	1	Huawei	4882972	0.013093
185	2022	1	Tecno	4550078	0.012200
186	2022	1	Others	25835848	0.069274

187 rows × 5 columns

In [12]: Agg_reg_appopens1

Out[12]:

	reg_users	app_opens	year	quater
0	46877867	0	2018	1
1	63648311	0	2018	2
2	80010952	0	2018	3
3	102262055	0	2018	4
4	123432699	0	2019	1
5	141808226	920413791	2019	2
6	159293934	3448366103	2019	3
7	178279085	4301071954	2019	4
8	197575164	4768944662	2020	1
9	218996326	4357634272	2020	2
10	241169683	5739326491	2020	3
11	264271293	7461623162	2020	4
12	284985430	8635508502	2021	1
13	305258086	9630620549	2021	2
14	328354923	12256377890	2021	3
15	351161395	15172529039	2021	4
16	372951831	15724195113	2022	1
17	393424643	17786667567	2022	2
18	414324881	19223454074	2022	3
19	433947940	21262699669	2022	4

In [8]: import pandas as pd
import os

```

import json
dic_yearbased_map_trans={"year":[],"quater":[],"state":[],"Transaction_count":[],"Transaction_amount":[]}
dic_year_state_map_trans={"state":[],"Transaction_amount":[],"Transaction_count":[],"year":[],"quater":[],"dist
#dic_state_yearbased={"year":[], "quater":[], "Transaction_type":[],"Transaction_count":[], "Transaction_amount

data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\map\transaction\hover\country\india")
common_data=os.listdir(data_aggre_trans_path)
print(common_data)
for i in common_data:
    if i.isnumeric()==True:
        #if i.isnumeric()==True:
            year=data_aggre_trans_path+'/'+i+'/'
            #print("year:", year)
            json_path=os.listdir(year)
            # print("json path:", json_path)
            for onejson in json_path:
                files=year+onejson
                data=open(files,"r")
                final_data=json.load(data)
                #print("final_data", final_data )
                for data in final_data["data"]["hoverDataList"]:
                    name=data["name"]
                    count=data["metric"][0]["count"]
                    amount=data["metric"][0]["amount"]
                    dic_yearbased_map_trans["state"].append(name)
                    dic_yearbased_map_trans["Transaction_count"].append(count)
                    dic_yearbased_map_trans["Transaction_amount"].append(amount)
                    dic_yearbased_map_trans["year"].append(i)
                    dic_yearbased_map_trans["quater"].append(int(onejson.strip(".json")))

                #print(dic_yearbased_map_trans)
                #dic_yearbased_map_trans={"year":[],"quater":[],"state":[],"Transaction_count":[],"Transaction_amou

                #TypeError: list indices must be integers or slices, not str count=data["metric"]["count"]

            #output
# ['2018', '2019', '2020', '2021', '2022', 'state']
# final_data {'success': True, 'code': 'SUCCESS', 'data': {'hoverDataList': [{'name': 'puducherry',
# 'metric': [{'type': 'TOTAL', 'count': 104212, 'amount': 165825971.36983618}]}], {'name': 'tamil nadu', 'metric'
#map_trans_year_based=pd.DataFrame(dic_yearbased_map_trans)
    if i.isnumeric()==False:
        state=data_aggre_trans_path+"/"+i
        all_state=os.listdir(state)
        #print(all_state)
        for j in all_state:
            onebyone_state=state+"/"+j
            onebyone_state_year=os.listdir(onebyone_state)
            # print("onebyone_state_year:",onebyone_state_year)
            for y in onebyone_state_year:
                onebyone_state_year_json_path= onebyone_state+"/"+y
                # print("onebyone_state_year_json_path:", onebyone_state_year_json_path)
                onebyone_state_year_json_pos=os.listdir(onebyone_state_year_json_path)
                #print("onebyone_state_year_json_pos:", onebyone_state_year_json_pos)
                for onejson in onebyone_state_year_json_pos:
                    # print("onejson:", onejson )
                    onebyone_state_year_json_pos_weneed=onebyone_state_year_json_path+"/"+onejson
                    files_open=open(onebyone_state_year_json_pos_weneed,"r")
                    files_final=json.load(files_open)
                    print(files_final)
                    for data in files_final["data"]["hoverDataList"]:
                        name=data["name"]
                        count=data["metric"][0]["count"]
                        amount=data["metric"][0]["amount"]
                        dic_year_state_map_trans["district"].append(name)
                        dic_year_state_map_trans["Transaction_count"].append(count)
                        dic_year_state_map_trans["Transaction_amount"].append(amount)
                        dic_year_state_map_trans["year"].append(y)
                        dic_year_state_map_trans["quater"].append(int(onejson.strip(".json")))
                        dic_year_state_map_trans["state"].append(j)
map_trans_year_based=pd.DataFrame(dic_yearbased_map_trans)
map_trans_year_state_based=pd.DataFrame(dic_year_state_map_trans)

```

In [14]: map_trans_year_based

Out[14]:

	year	quater	state	Transaction_count	Transaction_amount
0	2018	1	puducherry	104212	1.658260e+08
1	2018	1	tamil nadu	6726622	1.126156e+10
2	2018	1	uttar pradesh	12537805	1.393997e+10
3	2018	1	madhya pradesh	8025395	8.681603e+09
4	2018	1	andhra pradesh	9039585	1.199628e+10
...
715	2022	4	jammu & kashmir	27316158	4.431530e+10
716	2022	4	goa	17192674	2.492499e+10
717	2022	4	arunachal pradesh	4632709	9.684910e+09
718	2022	4	delhi	515203457	7.449150e+11
719	2022	4	uttarakhand	77983880	1.154960e+11

720 rows × 5 columns

In [15]: map_trans_year_state_based

Out[15]:

	state	Transaction_amount	Transaction_count	year	quater	district
0	andaman-&-nicobar-islands	9.316631e+05	442	2018	1	north and middle andaman district
1	andaman-&-nicobar-islands	1.256025e+07	5688	2018	1	south andaman district
2	andaman-&-nicobar-islands	1.139849e+06	528	2018	1	nicobars district
3	andaman-&-nicobar-islands	1.317863e+06	825	2018	2	north and middle andaman district
4	andaman-&-nicobar-islands	2.394824e+07	9395	2018	2	south andaman district
...
14631	west-bengal	2.804568e+10	12690126	2022	4	nadia district
14632	west-bengal	1.614650e+10	7617444	2022	4	birbhum district
14633	west-bengal	3.309949e+10	14484229	2022	4	purba medinipur district
14634	west-bengal	2.721861e+10	12492746	2022	4	maldah district
14635	west-bengal	1.801650e+10	8827502	2022	4	darjiling district

14636 rows × 6 columns

```
In [9]: import pandas as pd
import os
import json

dic_yearbased_map_user={"year":[],"quater":[],"state":[],"reg_user":[],"app_opens":[]}
dic_year_stata_map_user={"year":[],"quater":[],"district":[],"reg_user":[],"app_opens":[]}
data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\map\user\hover\country\india")
common_data=os.listdir(data_aggre_trans_path)
print("common_data:", common_data)
for i in common_data:
    print("i:", i)
    if i.isnumeric()==True:
        #if i isnumeric()==True:
            year=data_aggre_trans_path+'/'+i+'/'
            print("year:", year)
            json_path=os.listdir(year)
            print("json_path:", json_path)
            for onejson in json_path:
                files=year+onejson
                data=open(files,"r")
                final_data=json.load(data)
                #print("final_data", final_data)
                for data in final_data["data"]["hoverData"].items():
                    state=data[0]
                    reg_user=data[1]['registeredUsers']
                    app_opens=data[1]['appOpens']
                    #print("state:", state)
                    #print("reg_users:", reg_user)

                    dic_yearbased_map_user["state"].append(state)
                    dic_yearbased_map_user["app_opens"].append(app_opens)
                    dic_yearbased_map_user["year"].append(i)
                    dic_yearbased_map_user["quater"].append(int(onejson.strip(".json")))
                    dic_yearbased_map_user["reg_user"].append(reg_user)

#yearbased_map_user=pd.DataFrame(dic_yearbased_map_user)
if i.isnumeric()==False:
```

```

state=data_aggre_trans_path+"/"+i
all_state=os.listdir(state)
#print(all_state)
for j in all_state:
    print(j)
    onebyone_state=state+"/"+j
    onebyone_state_year=os.listdir(onebyone_state)
    print("onebyone_state_year:",onebyone_state_year)
    for y in onebyone_state_year:
        onebyone_state_year_json_path= onebyone_state+"/"+y
        print("onebyone_state_year_json_path:", onebyone_state_year_json_path)
        onebyone_state_year_json_pos=os.listdir(onebyone_state_year_json_path)
        print("onebyone_state_year_json_pos:", onebyone_state_year_json_pos)
        for onejson in onebyone_state_year_json_pos:
            print("onejson:", onejson )
            onebyone_state_year_json_pos_weneed=onebyone_state_year_json_path+"/"+onejson
            files_open=open(onebyone_state_year_json_pos_weneed,"r")
            files_final=json.load(files_open)
            print("final_files:",files_final)
            for data in files_final["data"]["hoverData"].items():
                # print("data:", data)
                district=data[0]
                reg_user=data[1]['registeredUsers']
                app_opens=data[1]['appOpens']

                dic_year_stata_map_user["district"].append(district)
                dic_year_stata_map_user["year"].append(y)
                dic_year_stata_map_user["reg_user"].append(reg_user)
                dic_year_stata_map_user["app_opens"].append(app_opens)
                dic_year_stata_map_user["quater"].append(int(onejson.strip(".json")))

        #print("state:", state)

yearbased_map_user=pd.DataFrame(dic_yearbased_map_user)
year_state_map_user=pd.DataFrame(dic_year_stata_map_user)

```

In [17]: yearbased_map_user

Out[17]:

	year	quater	state	reg_user	app_opens
0	2018	1	puducherry	49318	0
1	2018	1	tamil nadu	2104754	0
2	2018	1	uttar pradesh	4694250	0
3	2018	1	madhya pradesh	2553603	0
4	2018	1	andhra pradesh	3336450	0
...
715	2022	4	jammu & kashmir	1793434	60205865
716	2022	4	goa	855202	43005212
717	2022	4	arunachal pradesh	390307	21775038
718	2022	4	delhi	15764149	401864749
719	2022	4	uttarakhand	4272049	160685547

720 rows × 5 columns

In [18]: year_state_map_user

Out[18]:

	year	quater	district	reg_user	app_opens
0	2018	1	north and middle andaman district	632	0
1	2018	1	south andaman district	5846	0
2	2018	1	nicobars district	262	0
3	2018	2	north and middle andaman district	911	0
4	2018	2	south andaman district	8143	0
...
14635	2022	4	nadia district	1359420	33853990
14636	2022	4	birbhum district	855236	20950662
14637	2022	4	purba medinipur district	1346908	38278506
14638	2022	4	maldah district	954892	29023743
14639	2022	4	darjiling district	564562	15982631

14640 rows × 5 columns

In [10]:

```
import pandas as pd
import os
import json
dic_yearbased_top_user={"state":[],"year":[],"quater":[],"count":[],"amount":[]}
dic_year_stata_top_user_pincode={"pincode":[],"year":[],"quater":[],"amount":[],"count":[]}
dic_year_stata_top_user_district={"year":[],"quater":[],"amount":[],"count":[],"state":[]}
data_aggre_trans_path=(r"G:\guvi work\pulse-master\pulse-master\data\top\transaction\country\india")
common_data=os.listdir(data_aggre_trans_path)
#print("common_data:", common_data)
for i in common_data:
    #print("i:", i)
    if i.isnumeric()==True:
        #if i.isnumeric()==True:
            year=data_aggre_trans_path+'/'+i+'/'
            #print("year:", year)
            json_path=os.listdir(year)
            # print("json_path:", json_path)
            for onejson in json_path:
                files=year+onejson
                data=open(files,"r")
                final_data=json.loads(data.read())
                #print("final_data", final_data)
                for data in final_data["data"]["states"]:
                    state=data["entityName"]
                    count=data["metric"]["count"]
                    amount=data["metric"]["amount"]

                    dic_yearbased_top_user["state"].append(state)
                    dic_yearbased_top_user["count"].append(count)
                    dic_yearbased_top_user["amount"].append(amount)
                    dic_yearbased_top_user["year"].append(i)
                    dic_yearbased_top_user["quater"].append(int(onejson.strip(".json")))

            if i.isnumeric()==False:
                print("i:", i)
                statel=data_aggre_trans_path+"/"+i
                all_state=os.listdir(statel)
                print("all_state:", all_state)
                for j in all_state:
                    print("state:", j)
                    onebyone_state=statel+"/"+j
                    print("onebyone_state", onebyone_state)
                    onebyone_state_year=os.listdir(onebyone_state)
                    #print("onebyone_state_year:",onebyone_state_year)
                    for y in onebyone_state_year:
                        print(y)
                        onebyone_state_year_json_path= onebyone_state+"/"+y
                        print("onebyone_state_year_json_path:", onebyone_state_year_json_path)
                        onebyone_state_year_json_pos=os.listdir(onebyone_state_year_json_path)
                        # print("onebyone_state_year_json_pos:", onebyone_state_year_json_pos)
                        for onejson in onebyone_state_year_json_pos:
                            #print("onejson:", onejson )
                            onebyone_state_year_json_pos_weneed=onebyone_state_year_json_path+"/"+onejson
                            files_open=open(onebyone_state_year_json_pos_weneed,"r")
                            files_final=json.load(files_open)
                            print("final_files:",files_final)
                            for data in files_final["data"]["districts"]:
                                state=data["entityName"]
                                count=data["metric"]["count"]
                                amount=data["metric"]["amount"]
                                #dic_year_stata_top_user_district={"year":[],"quater":[],"amount":[],"count":[],"state":}
                                dic_year_stata_top_user_district["year"].append(y)
```

```

dic_year_stata_top_user_district["quater"].append(int(onejson.strip(".json")))
dic_year_stata_top_user_district["amount"].append(amount)
dic_year_stata_top_user_district["state"].append(state)
dic_year_stata_top_user_district["count"].append(count)
#
    print("state:", state)
    print("count:", count)
    print("amount:", amount)
    for datas in files_final["data"]["pincodes"]:
#dic_year_stata_top_user_pincode={"pincode":[],"year":[],"quater":[],"amount":[],"count":[]}
    pincode=datas["entityName"]
    count1=datas["metric"]["count"]
    amount1=datas["metric"]["amount"]
    dic_year_stata_top_user_pincode["pincode"].append(pincode)
    dic_year_stata_top_user_pincode["count"].append(count1)
    dic_year_stata_top_user_pincode["amount"].append(amount1)
    dic_year_stata_top_user_pincode["year"].append(y)
    dic_year_stata_top_user_pincode["quater"].append(int(onejson.strip(".json")))
#
#
#
    print("pincode:", pincode)
    print("count:", count1)
    print("amount:", amount1)

```

```

yearbased_top_user=pd.DataFrame(dic_yearbased_top_user)
year_state_top_user_district=pd.DataFrame(dic_year_stata_top_user_district)
year_state_top_user_pincode=pd.DataFrame(dic_year_stata_top_user_pincode)

```

In [20]: yearbased_top_user

Out[20]:

	state	year	quater	count	amount
0	maharashtra	2018	1	16387034	2.171161e+10
1	uttar pradesh	2018	1	12537805	1.393997e+10
2	karnataka	2018	1	12016899	1.921790e+10
3	west bengal	2018	1	11710225	1.062598e+10
4	andhra pradesh	2018	1	9039585	1.199628e+10
...
195	uttar pradesh	2022	4	836369071	1.376379e+12
196	andhra pradesh	2022	4	690017077	1.424725e+12
197	bihar	2022	4	517084649	9.360121e+11
198	delhi	2022	4	515203457	7.449150e+11
199	odisha	2022	4	445714743	6.522171e+11

200 rows × 5 columns

In [21]: year_state_top_user_district

Out[21]:

	year	quater	amount	count	state
0	2018	1	1.256025e+07	5688	south andaman
1	2018	1	1.139849e+06	528	nicobars
2	2018	1	9.316631e+05	442	north and middle andaman
3	2018	2	2.394824e+07	9395	south andaman
4	2018	2	3.072437e+06	1120	nicobars
...
5915	2022	4	3.373698e+10	15651650	south twenty four parganas
5916	2022	4	3.309949e+10	14484229	purba medinipur
5917	2022	4	2.755409e+10	13931352	hooghly
5918	2022	4	2.793786e+10	13350090	howrah
5919	2022	4	2.521681e+10	12768161	paschim medinipur

5920 rows × 5 columns

In [22]: year_state_top_user_pincode

Out[22]:

	pincode	year	quater	amount	count
0	744101	2018	1	2.769298e+06	1622
1	744103	2018	1	2.238042e+06	1223
2	744102	2018	1	3.519060e+06	969
3	744105	2018	1	1.298561e+06	685
4	744104	2018	1	1.039715e+06	340
...
7134	722101	2022	4	5.748321e+09	2900058
7135	700135	2022	4	3.527457e+09	2471048
7136	732101	2022	4	5.052109e+09	2407008
7137	700091	2022	4	2.176640e+09	2348447
7138	711101	2022	4	4.048329e+09	1765314

7139 rows × 5 columns

In [11]:

```
import os
import json
import pandas as pd

top_user_path=(r'G:\guvi work\pulse-master\pulse-master\data\top\user\country\india')
dic_top_user_year={"state":[],"year":[],"quater":[],"registeredUsers":[]}
top_user_state={"state":[],"reg_users":[],"year":[],"quater":[]}
top_user_pincode={"pincode":[],"reg_users":[],"year":[],"quater":[]}
#top_user_year=pd.DataFrame(dic_top_user_year)
common_path=os.listdir(top_user_path)
for i in common_path:
    if i.isnumeric()==True:
        yearlistpath=top_user_path+"/"+i
        # print(yearlistpath)
        json_path=os.listdir(yearlistpath)
        # print(json_path)
        for onejson in json_path:
            #print(onejson)
            json_open_path=yearlistpath+"/"+onejson
            final=open(json_open_path,"r")
            alloutput=json.load(final)
            #print(alloutput)
            for data in alloutput["data"]["states"]:
                state_name=data["name"]
                reg_users=data["registeredUsers"]

                dic_top_user_year["state"].append(state_name)
                dic_top_user_year["year"].append(i)
                dic_top_user_year["quater"].append(int(onejson.strip(".json")))
                dic_top_user_year["registeredUsers"].append(reg_users)

if i.isnumeric()==False:
    #print("i", i)
    path=top_user_path+"/"+i
    one_by_onestate=os.listdir(path)
    # print("one_by_onestate", one_by_onestate)
    for eachstate in one_by_onestate:
        year_path=path+"/"+eachstate
        yearlist=os.listdir(year_path)
        #print(year)
        for onebyoneyear in yearlist:
            yearinside=year_path+"/"+onebyoneyear
            json_path=os.listdir(yearinside)
            for onejson in json_path:
                final_path=yearinside+"/"+onejson
                file=open(final_path,"r")
                outputfiles=json.load(file)
                print(outputfiles)
                for data in outputfiles["data"]["districts"]:
                    state=data["name"]
                    reg_users=data["registeredUsers"]
                    print("state:", state)
                    print("reg_users:", reg_users)

                    top_user_state["state"].append(state)
                    top_user_state["reg_users"].append(reg_users)
                    top_user_state["year"].append(onebyoneyear)
                    top_user_state["quater"].append(int(onejson.strip(".json")))
                for datal in outputfiles["data"]["pincodes"]:
                    pincode=datal["name"]
                    reg_usersl=datal["registeredUsers"]
                    print(" pincode:", pincode)
                    print("reg usersl:", reg usersl)
```

```

top_user_pincode["pincode"].append(pincode)
top_user_pincode["reg_users"].append(reg_users1)
top_user_pincode["year"].append(onebyoneyear)
top_user_pincode["quater"].append(int(onejson.strip(".json")))

```

```

top_user_year_only=pd.DataFrame(dic_top_user_year)
top_user_stateonly=pd.DataFrame(top_user_state)
top_user_pincodeonly=pd.DataFrame(top_user_pincode)

```

In [24]: top_user_year_only

Out[24]:

	state	year	quater	registeredUsers
0	maharashtra	2018	1	6106994
1	uttar pradesh	2018	1	4694250
2	karnataka	2018	1	3717763
3	andhra pradesh	2018	1	3336450
4	telangana	2018	1	3315560
...
195	telangana	2022	4	25656381
196	west bengal	2022	4	25536381
197	tamil nadu	2022	4	25136112
198	madhya pradesh	2022	4	22913924
199	gujarat	2022	4	22412353

200 rows × 4 columns

In [25]: top_user_stateonly

Out[25]:

	state	reg_users	year	quater
0	south andaman	5846	2018	1
1	north and middle andaman	632	2018	1
2	nicobars	262	2018	1
3	south andaman	8143	2018	2
4	north and middle andaman	911	2018	2
...
5915	howrah	1422011	2022	4
5916	nadia	1359420	2022	4
5917	purba medinipur	1346908	2022	4
5918	paschim medinipur	1217113	2022	4
5919	purba bardhaman	1119310	2022	4

5920 rows × 4 columns

In [26]: top_user_pincodeonly

Out[26]:

	pincode	reg_users	year	quater
0	744103	1608	2018	1
1	744101	1108	2018	1
2	744105	1075	2018	1
3	744102	1006	2018	1
4	744104	272	2018	1
...
7135	700015	108457	2022	4
7136	742304	105471	2022	4
7137	721101	105279	2022	4
7138	700091	102363	2022	4
7139	700150	101966	2022	4

7140 rows × 4 columns

In [27]: import mysql.connector as sql

```
mydb = sql.connect(
    host="localhost",
    user="root",
    password="Risvana123",
    database="phonepay_project"

)

print(mydb)
mycursor = mydb.cursor(buffered=True)
```

<mysql.connector.connection_cext.CMySQLConnection object at 0x0000026E62E6BF90>

```
In [28]: mappings={ 'Andaman-&-Nicobar-islands': 'Andaman & Nicobar',
                    'Andhra-Pradesh': 'Andhra Pradesh',
                    'Arunachal-Pradesh': 'Arunachal Pradesh',
                    'Dadra-&-Nagar-haveli-&-daman-&-diu': 'Dadra and Nagar Haveli and Daman and Diu',
                    'Himachal-Pradesh': 'Himachal Pradesh',
                    'Jammu-&-Kashmir': 'Jammu & Kashmir',
                    'Madhya-Pradesh': 'Madhya Pradesh',
                    'Tamil-Nadu': 'Tamil Nadu',
                    'Uttar-Pradesh': 'Uttar Pradesh',
                    'West-Bengal': 'West Bengal'
                }
```

```
In [30]: Agg_Trans_state_year["alter_state"]=Agg_Trans_state_year["alter_state"].replace(mappings)
```

```
In [ ]: #Establishing DataBase Connection
```

```
In [31]: !pip install sqlalchemy
```

Requirement already satisfied: sqlalchemy in c:\users\sayd abuthahir\appdata\local\programs\python\python311\lib\site-packages (2.0.0)
Requirement already satisfied: typing-extensions>=4.2.0 in c:\users\sayd abuthahir\appdata\local\programs\python\python311\lib\site-packages (from sqlalchemy) (4.5.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\sayd abuthahir\appdata\local\programs\python\python311\lib\site-packages (from sqlalchemy) (2.0.2)
[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

```
In [32]: import sqlalchemy
import pymysql
# import the module
from sqlalchemy import create_engine
from pandas.io import sql
# create sqlalchemy engine
engine=create_engine("mysql+pymysql://root:Risvana123@localhost:3306/phonepay_project")
```

```
In [33]: Agg_user_years1.to_sql("agg_user_years",engine,if_exists="replace",index=False)
```

```
Out[33]: 187
```

```
In [35]: from sqlalchemy import create_engine, text
```

```
In [37]: a="yearbased_top_user"
sql=f'select * from {a}'
with engine.begin() as conn:
    df = pd.read_sql_query(con=engine.connect(),sql=text(sql))
df
```

Out[37]:

	index	year	quater	count	amount	alter_state
0	0	2018	1	16387034	2.171161e+10	Maharashtra
1	1	2018	1	12537805	1.393997e+10	Uttar Pradesh
2	2	2018	1	12016899	1.921790e+10	Karnataka
3	3	2018	1	11710225	1.062598e+10	West Bengal
4	4	2018	1	9039585	1.199628e+10	Andhra Pradesh
...
195	195	2022	4	836369071	1.376379e+12	Uttar Pradesh
196	196	2022	4	690017077	1.424725e+12	Andhra Pradesh
197	197	2022	4	517084649	9.360121e+11	Bihar
198	198	2022	4	515203457	7.449150e+11	Delhi
199	199	2022	4	445714743	6.522171e+11	Odisha

200 rows × 6 columns

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js