**Open Visual Studio**

Install required modules

* + *pip install powerbiclient*
  + *pip install wheel*
  + *pip install pandas*

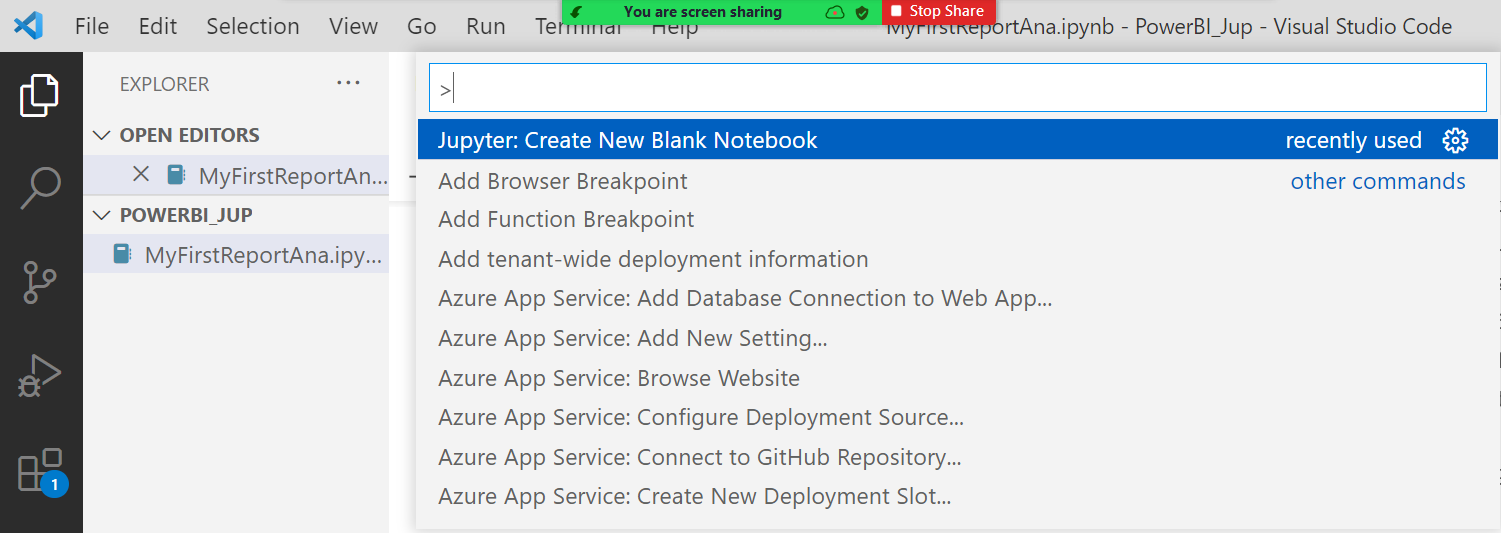
Open your project folder in cmd

Type code .

It open visual studio

Click view and select Command Palette

Then click Jupyter: create new blank notebook

****

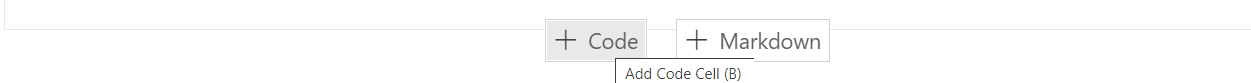
**Connect Power BI**

from powerbiclient import Report, models

from io import StringIO

import pandas as pd

Click code to add new step



**Connect Power BI and get access token**

# Import the DeviceCodeLoginAuthentication class to authenticate against Power BI

from powerbiclient.authentication import DeviceCodeLoginAuthentication

# Initiate device authentication

device\_auth = DeviceCodeLoginAuthentication()

access\_token = device\_auth.get\_access\_token()

**Report**

group\_id="70e996e2-470b-4e95-9aa6-b07fa8acf3d5"

report\_id="596c85a1-1618-486b-a8e2-6b093e16f639"

report = Report(group\_id=group\_id, report\_id=report\_id, access\_token = access\_token)

**Report Status**

def loaded\_callback(event\_details):

    print('The report successfully loaded!!')

def error\_callback(event\_details):

    print('error', event\_details)

report.on('loaded', loaded\_callback)

report.on('error', error\_callback)

**Open report**

report

**Get Report first Page**

import tracemalloc

***// The tracemalloc module is a debug tool to trace memory blocks allocated by Python***

tracemalloc.start()

active\_page = report.get\_pages()[0]['name']

**Get Visual from active page**

visuals = report.visuals\_on\_page(active\_page)

**Display Output**

visuals

[{'name': 'f3feab200dd10010895c', 'type': 'slicer', 'layout': {'x': 640.5194805194806, 'y': 43.24675324675325, 'z': 0, 'width': 597.2727272727273, 'height': 60.77922077922078, 'displayState': {'mode': 0}}}, {'name': 'c6012d9c7ba00c7e6bd6', 'title': 'Revenue by Country', 'type': 'columnChart', 'layout': {'x': 257.910447761194, 'y': 119.40298507462687, 'z': 1000, 'width': 294.92537313432837, 'height': 263.88059701492534, 'displayState': {'mode': 0}}}, {'name': 'bc86088011d51323506e', 'title': 'Revenue and % Growth by Quarter', 'type': 'lineClusteredColumnComboChart', 'layout': {'x': 257.910447761194, 'y': 402.3880597014925, 'z': 2000, 'width': 980.2985074626865, 'height': 259.1044776119403, 'displayState': {'mode': 0}}}, {'name': 'f9b5967bcea0167c07b2', 'type': 'card', 'layout': {'x': 41.7910447761194, 'y': 380.8955223880597, 'z': 3000, 'width': 192.23880597014926, 'height': 77.61194029850746, 'displayState': {'mode': 0}}}, {'name': 'f63b02ead35aea90d68a', 'type': 'pivotTable', 'layout': {'x': 552.8358208955224, 'y': 119.40298507462687, 'z': 4000, 'width': 685.3731343283582, 'height': 263.88059701492534, 'displayState': {'mode': 0}}}, {'name': '1f912b730a0115b2b710', 'title': 'PY Sales and Revenue', 'type': 'gauge', 'layout': {'x': 35.82089552238806, 'y': 113.43283582089552, 'z': 5000, 'width': 198.2089552238806, 'height': 202.98507462686567, 'displayState': {'mode': 0}}}, {'name': '0c98467138a540029177', 'type': 'textbox', 'layout': {'x': 247.7922077922078, 'y': 43.24675324675325, 'z': 6000, 'width': 417.2727272727273, 'height': 60.77922077922078, 'displayState': {'mode': 0}}}, {'name': 'd23bf137b61bcc201c12', 'title': 'Year', 'type': 'PBI\_CV\_16948668\_E17D\_454B\_8664\_2F2C470EA8C1', 'layout': {'x': 34.62686567164179, 'y': 540.8955223880597, 'z': 7000, 'width': 201.7910447761194, 'height': 57.3134328358209, 'displayState': {'mode': 0}}}, {'name': 'a96ea9f44063d174cc0e', 'type': 'image', 'layout': {'x': 34.285714285714285, 'y': 43.42857142857143, 'z': 8000, 'width': 200, 'height': 60.57142857142857, 'displayState': {'mode': 0}}}]

**Get particular visual**

visual = next(filter(lambda visuals: visuals['type'] == 'pivotTable',visuals))

**To display name of the chart**

visual\_name = visual['name']

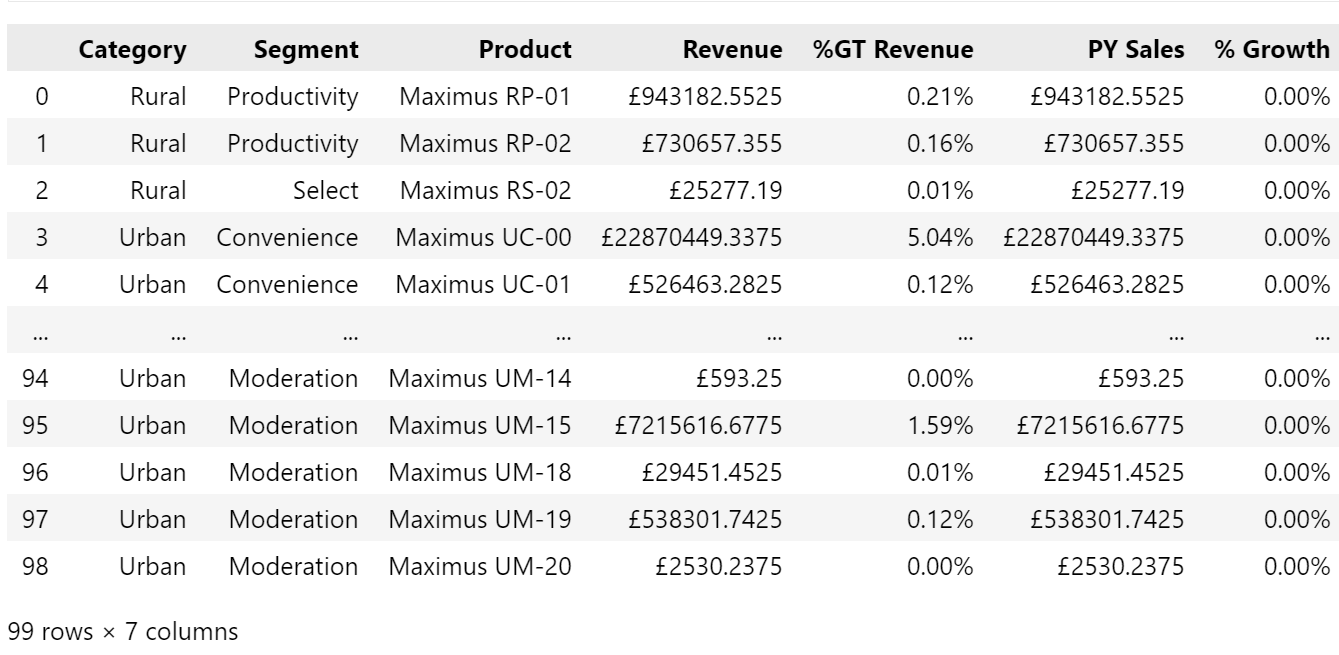
**Get the data**

data = report.export\_visual\_data(active\_page, visual\_name, rows=100)

\_data = StringIO(data)

df = pd.read\_csv(\_data,sep=",")

df



**Get one more Visual info**

**Get particular visual**

visual = next(filter(lambda visuals: visuals['type'] == 'lineClusteredColumnComboChart', visuals))

**To display name of the chart**

visual\_name = visual['name']

**Get the data**

summ\_data = report.export\_visual\_data(active\_page, visual\_name, rows=10)

data = StringIO(summ\_data)

df = pd.read\_csv(data, sep=",")

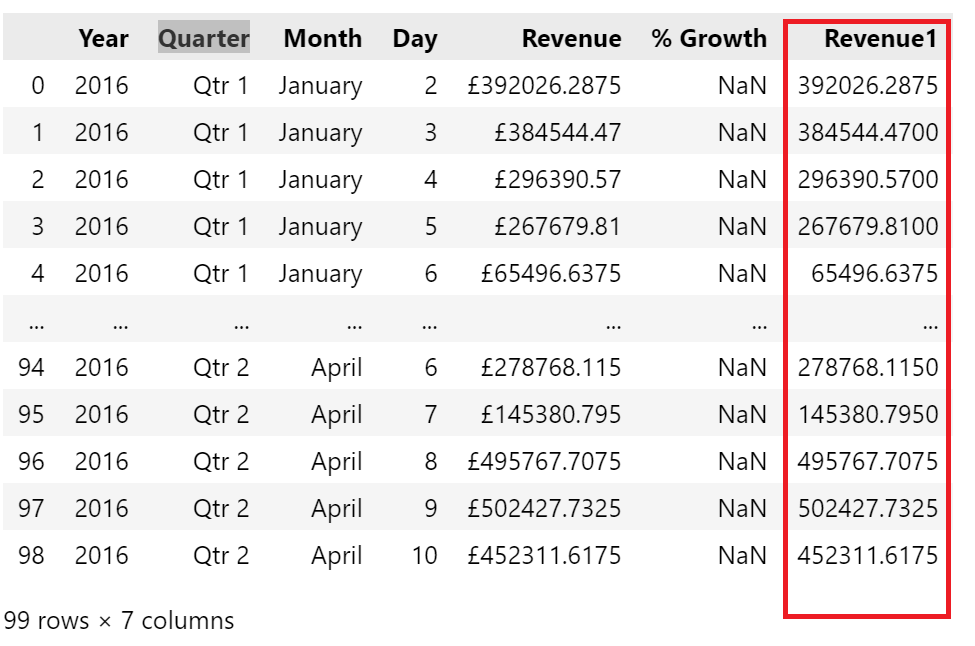
df

|  | **Year** | **Quarter** | **Month** | **Day** | **Revenue** | **% Growth** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 2013 | Qtr 1 | January | 15 | £1240958.2025 | NaN |
| 1 | 2013 | Qtr 1 | February | 15 | £1375329.365 | NaN |
| 2 | 2013 | Qtr 1 | March | 15 | £1848573.7275 | NaN |
| 3 | 2013 | Qtr 2 | April | 15 | £2151424.8375 | NaN |
| 4 | 2013 | Qtr 2 | May | 15 | £1930901.3325 | NaN |
| 5 | 2013 | Qtr 2 | June | 15 | £1931843.36 | NaN |
| 6 | 2013 | Qtr 3 | July | 15 | £1559326.555 | NaN |
| 7 | 2013 | Qtr 3 | August | 15 | £1629125.1925 | NaN |
| 8 | 2013 | Qtr 3 | September | 15 | £1773450.91 | NaN |

**Convert to Numeric**

df["Revenue1"] = pd.to\_numeric(df["Revenue"].replace({r'\£':''}, regex = True))

df

****

**Install matplotlib for get the visual chart**

python -m pip install -U matplotlib

import matplotlib.pyplot as plt

**Group by Country**

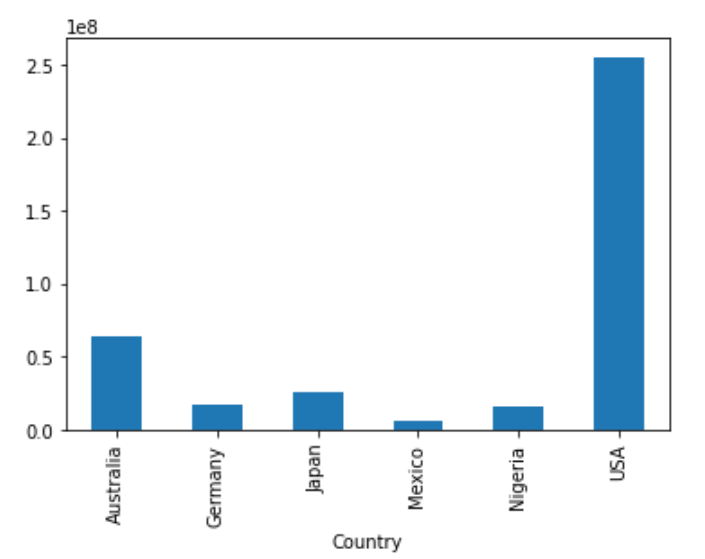
df1 = df.groupby('Country')["Revenue1"].sum()

df1



df1.plot(x='Country', y='Revenue1', kind = 'bar')

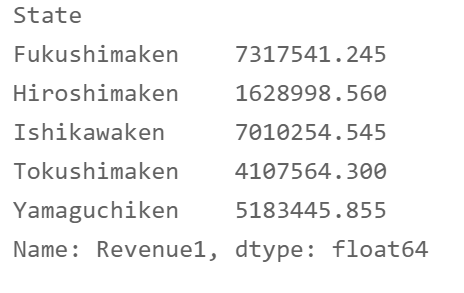
plt.show



**Group by State and filter by country - Japan**

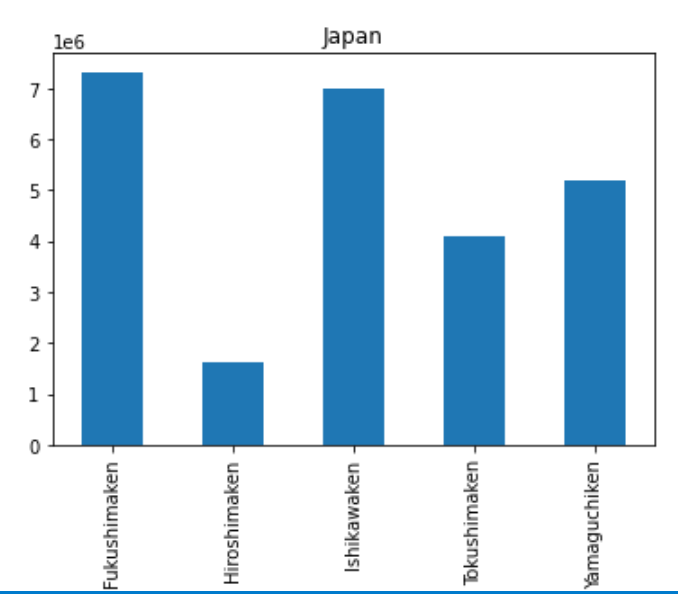
df2 = df[df.Country == "Japan"].groupby('State')["Revenue1"].sum()

df2



df2.plot(x='State', y='Revenue1', kind = 'bar', title="Japan")

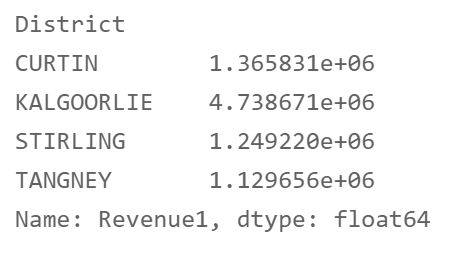
plt.show



**Group by District and filter by state**

df3 = df[df.State == "Western Australia"].groupby('District')["Revenue1"].sum()

df3



df3.plot(x='District', y='Revenue1', kind = 'bar', title='Western Australia')

plt.show

