FEDERAL UNIVERSITY OF TECHNOLOGY OWERRI

P.M.B 1526 OWERRI, IMO STATE

A REPORT ON SIX (6) MONTHS STUDENT INDUSTRIAL WORK EXPERIENCE SCHEME (SIWES)

COMPLETED AT:

ABUJA DATA SCHOOL/ MANGROOVE TECHNOLOGIES, ABUJA

PRESENTED BY:

NKOBIE CHINEDU EMMANUEL

20181092713

SUBMITTED TO:

THE DEPARTMENT OF PETROLEUM ENGINEERING

SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY

(SEET)

IN PARTIAL FUFILMENT FOR AWARD OF DEGREE OF BACHELOR OF ENGINEERING (B.ENG.) IN PETROLEUM ENGINEERING

SEPTEMBER, 2023

DECLARATION

I, the under-mentioned, solemnly declare that this internship report on ABUJA DATA SCHOOL/MANGROOVE TECHNOLOGIES is my original work. I further declare that I have strictly observed reporting ethics and duly discharged copyright obligation and properly referred all outsourcing of materials used in this report and nothing is confidential in this report in respect of the company of my internship. I take the responsibility for all legal and ethical requirements regarding this report.

Signature; ______________________________

NKOBIE CHINEDU EMMANUEL

20181092713

CERTIFICATION

ACKNOWLEDGEMENT

The six (6) month student industrial work experience scheme (SIWES) was a success and I would like to acknowledge **THE DEPARTMENT OF PETROLEUM ENGINEERING FUTO** for giving me such opportunity. I also acknowledge my **FAMILY** for providing necessary support and resources that ensured the 400 level SIWES is successful.

I acknowledge **MR. EMMANUEL OTORI** for providing such amazing platform as **ABUJA DATA SCHOOL/MANGROOVE TECHNOLOGIES** that allowed me gain useful industrial skills and work experience. And to my supervisors and tutors **MR. OLATOMIWA LAWALSON, MR. CALEB SOMANYA, MR. INNOCENT OJISUA** for their impact in the course of the Six (6) month SIWES. I also wish to acknowledge colleagues and every staff I worked with during the SIWES, you all made my experience a worthy one.

ABSTRACT

Industrial training is an important phase of a student life. A well planned, properly executed and evaluated industrial training helps a lot in developing a professional attitude and industrial practical experience. During a period of twenty-four (24) weeks training at ABUJA DATA SCHOOL/MANGROOVE TECHNOLOGIES, I worked and learned under the I.T training section of the company where I learnt to work with various Data Analytics tools such as; Microsoft Excel, Power BI (Business Intelligence), SPSS (Statistical Package for Social Science), Python and SQL (Structured Query Language) which are outlined in this report.

Included in this report is a brief summary about SIWES, its aims and objectives. Also included in this report is a brief detail about my place of internship and a summary of activities carried out for the period of internship. At the later part of this report, why oil and gas companies should act on data analytics is discussed. Also, limitations and challenges encountered during the period of internship and possible recommendations to mitigate them were stated.

I gained practical skills and experience by working on numerous Data Analysis projects, and with the guidance of my supervisors I was able to complete Data Analysis process. I could conclude that for this internship as a Data Analyst, the knowledge of tools and software proved to be beneficial not only for the tasks that were assigned to me, also it proves beneficial to me as an aspiring Petroleum Engineer that will eventually solve problems facing the energy sector by blending Domain knowledge with Analytics. I have become more skilled particularly in analyzing data, and exposed to an ideal working environment.

Keywords: Industrial Training, Abuja Data School, Data Analysis\Analytics, Microsoft Excel, Power BI, SPSS, Python, SQL, Oil and Gas companies

TABLE OF CONTENTS

CHAPTER ONE

1.0 ABOUT SIWES

Student Industrial work experience scheme (SIWES) is an essential criterion in a student's training program in tertiary institutions. This experience usually involves three to six or twelve months as the case may be of intensive training in an industry of the student's choice. It was established in 1973 by the Industrial Training Fund (ITF).

The SIWES program provide a vital technological industrial training component which is a needed enrichment of the formal engineering education. The petroleum Engineering Students undergo a total of 12-months industrial attachment program made up of:

a. A 3-month SIWES during the long vacation at the end of the second year.
b. A 3-month SIWES during the long vacation at the end of the third year.
c. A 4-month SIWES in relevant industry during the second semester of the fourth year.
d. A 2-month SIWES in relevant industry during the long vacation at the end of the fourth year.

1.1 AIMS AND OBJECTIVE OF SIWES

- Expose students to work methods and techniques in handling equipment that may not be available in the university.
- To provide the student with an opportunity to apply the theory in real life work situation, thereby bridging the gap between the university work and actual work experience.
- To provide an avenue for the student in Nigerian universities to acquire industrial training skills and experience in their course of study.
- Make the transition from the University to the world of work and thus enhance students contact for better job placement.

- To enlist and strengthen employer’s involvement in the entry process of preparing university graduates for employment in industries.

1.2 INTRODUCTION

The idea of using Data Analysis for business growth is not more than a decade old and businesses have actively started using their data for their betterment. Companies hire Data Analysis, Data Scientists, and Business Analysts for specific purpose of analyzing and deriving results from the data generated by various means and use them to develop strategies for growing their businesses.

Abuja Data School owned and managed by MANGROOVE TECHNOLOGIES was established to groom Data Analysts in order to ensure that professionals have the competitive advantage to work in challenging environments with competitive payment. They offer IT services and Consultancy with a company size of 10 to 15 employees.

They offer courses and training that are taken by competent instructors with real life examples and practical sessions to enhance the ability of the student to become proficient in Data Analysis, which take participants through the beginners, intermediate and advanced stages in Data Analysis. The Abuja Data School Training and Internship also offers an opportunity to work directly with experts, and that is a great way to practice and even learn some new skills in line with your career goal. A data analysis and Data analytics internship and training will help you strengthen your knowledge and skills both in areas in which you are quite familiar, and even more advanced areas such as data analytics, machine learning, artificial intelligence and other areas that deal with Big data. All it requires is for the intern to work closely and attentively with the experts.

1.3 SUMMARY OF ROLES AND ACTIVITIES

As an intern, my tasks were to:

1. Clean and query vetted public datasets using various Data Analysis tools such as Excel, SQL, Power BI, Python programming language and SPSS.

2. Perform Exploratory Data Analysis on the Datasets.
3. Analyze the cleaned and organized datasets and build simple visualizations.
4. Extract insights from the datasets and visualizations.
5. Create reports and in some cases dashboards about the insights.
6. And make possible recommendation(s) or solutions to problems or business tasks.

In summary, I completed basic Data Analysis process on open-sourced datasets and in some cases “dummy” datasets. These processes are summarized as follows; Ask, Prepare, Process, Analyze, Share and Act.

CHAPTER TWO

2.0 DETAILED INTERNSHIP ACTIVITIES

My goal and responsibility as a Data Analyst intern in Abuja Data School is to analyze open-sourced datasets (in some cases, create a dummy dataset) made available by the company and give reports, dashboards or both as the case may be. These reports and Dashboards contain results derived from the datasets, and recommendations or solutions to problems or business tasks which leads to making informed decisions.

The analysis process is carried out using the following tools:

1. Microsoft Excel
2. Power Business Intelligence (BI) Desktop
3. My SQL workbench 8.0
4. IBM Statistical Package for Social Science (SPSS) 25
5. Jupyter Notebook (anaconda 3)
6. Python Programming Language and its libraries
    a. NumPy
    b. Pandas
    c. Matplotlib
    d. Seaborn

All these tools together are used for the phases of Data Analysis which are: Data Cleaning, Exploratory Data Analysis, and Data Visualization.

2.1 METHODOLOGY

2.1.1 DATASET

Listed below are the datasets provided for Analysis:

1. **Attendance Record Sheet Dataset:** This Dataset is a dummy dataset that is created for the purpose of carrying out analysis. It contains names of employees in a company, the time they come in and go out, the estimated hours they spend in their work, and the target work hour. My task is to determine if each employee completed the stipulated work hour, estimated pay for the number of hours worked, if each employee is due for payment or not and plot a bar chart of names of employees and their respective estimated pay.

2. **Store sales Dataset:** This dataset contains sales record of a company across various sales location in the United States. My task is to create a pivot table containing sum of sale price and profit across each sales location for each month, and visualize the sum of sales price for each sales location inserting a slicer of months that serves as the filter.

3. **Company XYZ store sales Dataset**: This dataset belongs to a company whose identity is not mentioned for privacy reasons. It contains sales record of 3 months for its major branch supermarkets located at three cities across the country. My task is to perform an Exploratory Data Analysis (EDA) and visualizations to identify sales trends to enable them makes informed business decisions that will ensure the growth of the business and edge its competitors.

4. **Ds_salaries Dataset:** This Dataset contains the salaries of different Data Science roles in selected countries for a period of 3 years (i.e from 2020 to 2022). My task is to perform EDA on the dataset and visualize various components of the dataset using Pandas, matplotlib and Seaborn libraries in Python.

5. **Covid-19 Project Datasets:** These are datasets containing records of Covid-19 cases across the world. The datasets are public vetted datasets provided by John Hopkins College. My task is to clean the dataset, filter the dataset for Nigeria, perform EDA, Visualizations, and produce an executive summary in form of a report which also

contains recommendations to better handle any future pandemic. These are to be done using Python Programming Language and Microsoft Word.

6. **Open Source Datasets**; These are numerous datasets contained in a folder provided by the company. My task is carryout basic Data Analysis including; cleaning and transforming the datasets using power query editor, loading the transformed datasets into power BI desktop, modelling the datasets, analyzing the datasets, and creating dashboards for visualization. All these are to be done using Power BI desktop and Power query.
7. **Dummy datasets**
8. **Volve Production Dataset:** This dataset contains daily and monthly production data for seven (07) wellbores from the volve field in Norway. My task is to clean the dataset removing irrelevant columns and rows and create a dashboard using Power BI.

2.1.2 MICROSOFT EXCEL

Microsoft Excel is a spreadsheet editor developed by Microsoft for Windows, macOS, Android, iOS and iPadOS. It features calculation or computation capabilities, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications.

2.1.2.1 WORKING ON ATTENDANCE DATASET USING MICROSOFT EXCEL

Operations applied on this dataset

1. Loading of the dataset into Microsoft Excel
2. Applying *IF* function *[=IF(D3=F3,"COMPLETED","NOT COMPLETED")]* to determine if each employee completed the target work hour.
3. Applying Relative referencing formula *[=E3/240*] to determine the percentage of Estimated pay for each employee.
4. Applying Relative referencing formula [*=D3/F3*] to determine the percentage of Total hours spent on work.

5. Applying IF function *[=IF($I3>=0.59,"DUE","NOT DUE")]* to determine if employees are due for pay.
6. Applying Conditional Formatting to columns E, H, and I.
7. visualizing a Bar plot of Estimated pay and Names of Employees.

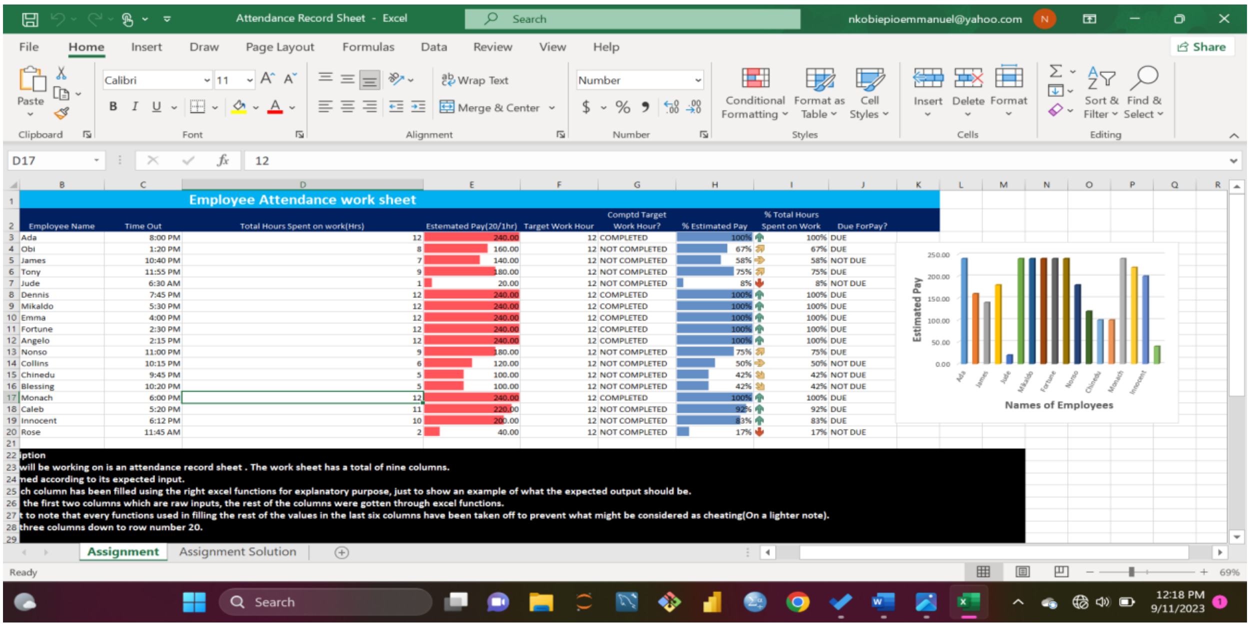
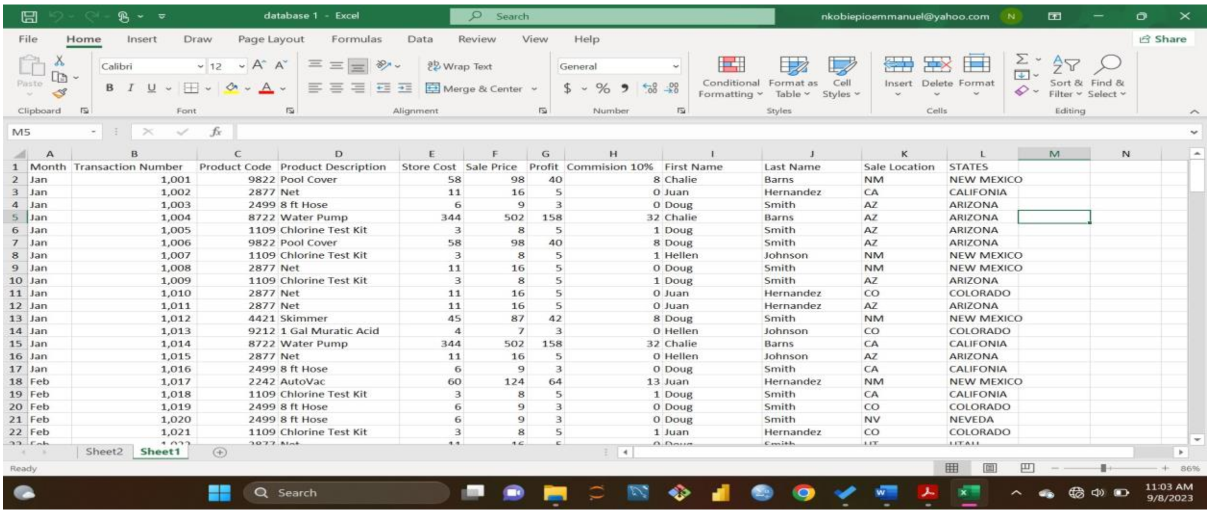


Fig 1. Image showing Final results of the Attendance Dataset spreadsheet.

2.1.2.2 WORKING ON STORE SALES DATASET USING MICROSOFT EXCEL

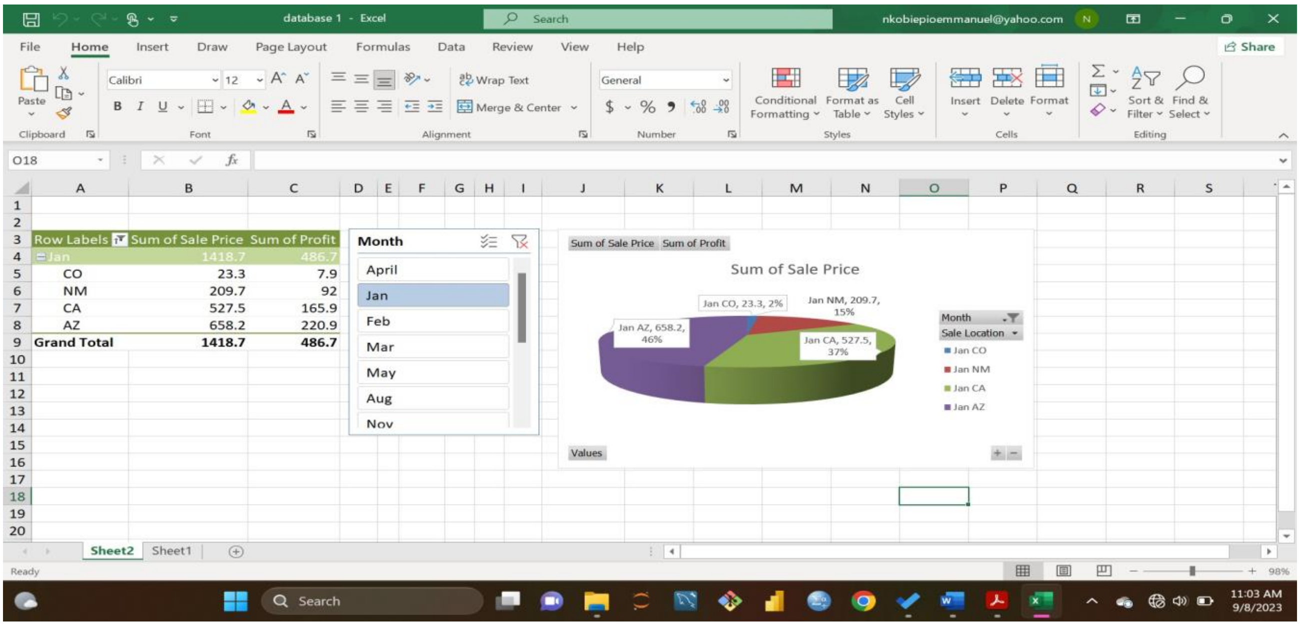
Operations applied on this dataset

1. Loading the dataset into Microsoft Excel
2. Creation of a Pivot Table having months and sales location as rows and sum of sales price and profit as columns.
3. Inserting a slicer containing months that serves as a filter for the Dataset.
4. Visualizing the Sum of sales price across each sales location for each month using a pivot bar chart.



Month	Transaction Number	Product Code	Product Description	Store Cost	Sale Price	Profit	Commision 10%	First Name	Last Name	Sale Location	STATES
Jan	1,001	9822	Pool Cover	58	98	40	8	Chalie	Barns	NM	NEW MEXICO
Jan	1,002	2877	Net	11	16	5	0	Juan	Hernandez	CA	CALIFONIA
Jan	1,003	2499	8 ft Hose	6	9	3	0	Doug	Smith	AZ	ARIZONA
Jan	1,004	8722	Water Pump	344	502	158	32	Chalie	Barns	AZ	ARIZONA
Jan	1,005	1109	Chlorine Test Kit	3	8	5	1	Doug	Smith	AZ	ARIZONA
Jan	1,006	9822	Pool Cover	58	98	40	8	Doug	Smith	AZ	ARIZONA
Jan	1,007	1109	Chlorine Test Kit	3	8	5	1	Hellen	Johnson	NM	NEW MEXICO
Jan	1,008	2877	Net	11	16	5	0	Doug	Smith	NM	NEW MEXICO
Jan	1,009	1109	Chlorine Test Kit	3	8	5	1	Doug	Smith	AZ	ARIZONA
Jan	1,010	2877	Net	11	16	5	0	Juan	Hernandez	CO	COLORADO
Jan	1,011	2877	Net	11	16	5	0	Juan	Hernandez	AZ	ARIZONA
Jan	1,012	4421	Skimmer	45	87	42	8	Doug	Smith	NM	NEW MEXICO
Jan	1,013	9212	1 Gal Muratic Acid	4	7	3	0	Hellen	Johnson	CO	COLORADO
Jan	1,014	8722	Water Pump	344	502	158	32	Chalie	Barns	CA	CALIFONIA
Jan	1,015	2877	Net	11	16	5	0	Hellen	Johnson	AZ	ARIZONA
Jan	1,016	2499	8 ft Hose	6	9	3	0	Doug	Smith	CA	CALIFONIA
Feb	1,017	2242	AutoVac	60	124	64	13	Juan	Hernandez	NM	NEW MEXICO
Feb	1,018	1109	Chlorine Test Kit	3	8	5	1	Doug	Smith	CA	CALIFONIA
Feb	1,019	2499	8 ft Hose	6	9	3	0	Doug	Smith	CO	COLORADO
Feb	1,020	2499	8 ft Hose	6	9	3	0	Doug	Smith	NV	NEVEDA
Feb	1,021	1109	Chlorine Test Kit	3	8	5	1	Juan	Hernandez	CO	COLORADO

Fig 2. Loading Store Sales dataset in Microsoft Excel



Row Labels	Sum of Sale Price	Sum of Profit
Jan	1418.7	486.7
CO	23.3	7.9
NM	209.7	92
CA	527.5	165.9
AZ	658.2	220.9
Grand Total	1418.7	486.7

Fig 3. Analyzing Store Sales dataset using a Pivot table, Slicer and a Pivot Chart

2.1.3 PYTHON PROGRAMMING LANGUAGE

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

2.1.3.1 WORKING ON COMPANY XYZ STORE SALES DATASETS USING PYTHON PROGRAMMING LANGUAGE

#Importing python libraries

```
import pandas as pd
import numpy as np
import glob
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib as mp
import calendar
```

#loading Datasets into Jupyter Notebook

```
combined_data = pd.concat(map(pd.read_csv,['Abuja_Branch.csv', 'Lagos_Branch.csv', 'Por
t_Harcourt_Branch.csv']))
combined_data.to_csv("combined_data.csv")

merged_data = pd.read_csv("combined_data.csv", index_col=['Invoice ID']).drop('Unnamed
: 0', axis=1)
```

#Data Exploration

 # Obtain the top 10 entries of the dataset

syntax: merged_data.head(10)

output:

```
            Branch   City Customer type  Gender          Product line  \
Invoice ID
692-92-5582      B  Abuja        Member  Female      Food and beverages
351-62-0822      B  Abuja        Member  Female     Fashion accessories
529-56-3974      B  Abuja        Member    Male  Electronic accessories
299-46-1805      B  Abuja        Member  Female       Sports and travel
319-50-3348      B  Abuja        Normal  Female      Home and lifestyle
371-85-5789      B  Abuja        Normal    Male       Health and beauty
273-16-6619      B  Abuja        Normal    Male      Home and lifestyle
649-29-6775      B  Abuja        Normal    Male     Fashion accessories
145-94-9061      B  Abuja        Normal  Female      Food and beverages
871-79-8483      B  Abuja        Normal    Male     Fashion accessories
```

Invoice ID	*Unit price*	*Quantity*	*Tax 5%*	*Total*	*Date*	*Time *
692-92-5582	*19742.4*	*3*	*2961.36*	*62188.56*	*2/20/2019*	*13:27*
351-62-0822	*5212.8*	*4*	*1042.56*	*21893.76*	*2/6/2019*	*18:07*
529-56-3974	*9183.6*	*4*	*1836.72*	*38571.12*	*3/9/2019*	*17:03*
299-46-1805	*33739.2*	*6*	*10121.76*	*212556.96*	*1/15/2019*	*16:19*
319-50-3348	*14508.0*	*2*	*1450.80*	*30466.80*	*3/11/2019*	*15:30*
371-85-5789	*31672.8*	*3*	*4750.92*	*99769.32*	*3/5/2019*	*10:40*
273-16-6619	*11952.0*	*2*	*1195.20*	*25099.20*	*3/15/2019*	*12:20*
649-29-6775	*12067.2*	*1*	*603.36*	*12670.56*	*2/8/2019*	*15:31*
145-94-9061	*31809.6*	*5*	*7952.40*	*167000.40*	*1/25/2019*	*19:48*
871-79-8483	*33886.8*	*5*	*8471.70*	*177905.70*	*2/25/2019*	*19:39*

Invoice ID	*Payment*	*cogs*	*gross margin percentage*	*gross income*	*Rating*
692-92-5582	Card	59227.2	4.761905	2961.36	5.9
351-62-0822	Epay	20851.2	4.761905	1042.56	4.5
529-56-3974	Cash	36734.4	4.761905	1836.72	6.8
299-46-1805	Cash	202435.2	4.761905	10121.76	4.5
319-50-3348	Epay	29016.0	4.761905	1450.80	4.4
371-85-5789	Epay	95018.4	4.761905	4750.92	5.1
273-16-6619	Card	23904.0	4.761905	1195.20	4.4
649-29-6775	Cash	12067.2	4.761905	603.36	6.7
145-94-9061	Cash	159048.0	4.761905	7952.40	9.6
871-79-8483	Card	169434.0	4.761905	8471.70	4.8

 # What is the size of the merged datasets

Syntax: merged_data.shape

Output: (1000, 16)

The dataset contains 1000 rows and 16 columns

 # To obtain the names of columns in the dataset

Syntax: merged_data.columns

Output: Index(['Branch', 'City', 'Customer type', 'Gender', 'Product line','Unit price', 'Quantit y', 'Tax 5%', 'Total', 'Date', 'Time', 'Payment','cogs', 'gross margin percentage', 'gross income' , 'Rating'], dtype='object')

 # To obtain Statistical information about the dataset.

Syntax: merged_data.describe()

Output:

	Unit price	Quantity	Tax 5%	Total	cogs \
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	20041.966800	5.510000	5536.572840	116268.029640	110731.456800
std	9538.066205	2.923431	4215.177173	88518.720636	84303.543463
min	3628.800000	1.000000	183.060000	3844.260000	3661.200000
25%	11835.000000	3.000000	2132.955000	44792.055000	42659.100000
50%	19882.800000	5.000000	4351.680000	91385.280000	87033.600000
75%	28056.600000	8.000000	8080.290000	169686.090000	161605.800000
max	35985.600000	10.000000	17874.000000	375354.000000	357480.000000

	gross margin percentage	gross income	Rating
count	1.000000e+03	1000.000000	1000.00000
mean	4.761905e+00	5536.572840	6.97270
std	6.131498e-14	4215.177173	1.71858
min	4.761905e+00	183.060000	4.00000
25%	4.761905e+00	2132.955000	5.50000
50%	4.761905e+00	4351.680000	7.00000
75%	4.761905e+00	8080.290000	8.50000
max	4.761905e+00	17874.000000	10.00000

 # To determine if the dataset contains null or empty values

Syntax: merged_data.isnull().sum()

Output:

Branch 0
City 0
Customer type 0
Gender 0
Product line 0
Unit price 0
Quantity 0
Tax 5% 0
Total 0
Date 0
Time 0
Payment 0
cogs 0
gross margin percentage 0
gross income 0
Rating 0
dtype: int64

There are no null or empty value(s) in the dataset

 # To obtain general information on the dataset

Syntax: merged_data.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, 692-92-5582 to 233-67-5758
Data columns (total 16 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Branch                   1000 non-null   object
 1   City                     1000 non-null   object
 2   Customer type            1000 non-null   object
 3   Gender                   1000 non-null   object
 4   Product line             1000 non-null   object
 5   Unit price               1000 non-null   float64
 6   Quantity                 1000 non-null   int64
 7   Tax 5%                   1000 non-null   float64
 8   Total                    1000 non-null   float64
 9   Date                     1000 non-null   object
 10  Time                     1000 non-null   object
 11  Payment                  1000 non-null   object
 12  cogs                     1000 non-null   float64
 13  gross margin percentage  1000 non-null   float64
 14  gross income             1000 non-null   float64
 15  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(8)
memory usage: 132.8+ KB
```

#Dealing with Data Frame Features

```
import datetime as dt

merged_data = pd.DataFrame(merged_data)

merge=merged_data.copy()
```

 # To convert Date in the dataset to standard python date

```
Syntax: merged_data['Date'] =pd.to_datetime(merged_data['Date'])

       merged_data

output:

            Branch     City Customer type  Gender  \
Invoice ID
692-92-5582      B    Abuja        Member  Female
351-62-0822      B    Abuja        Member  Female
529-56-3974      B    Abuja        Member    Male
```

299-46-1805	*B*	*Abuja*	*Member*	*Female*
319-50-3348	*B*	*Abuja*	*Normal*	*Female*
...	*...*	*...*	*...*	*...*
148-41-7930	*C*	*Port Harcourt*	*Normal*	*Male*
189-40-5216	*C*	*Port Harcourt*	*Normal*	*Male*
267-62-7380	*C*	*Port Harcourt*	*Member*	*Male*
652-49-6720	*C*	*Port Harcourt*	*Member*	*Female*
233-67-5758	*C*	*Port Harcourt*	*Normal*	*Male*

	Product line	*Unit price*	*Quantity*	*Tax 5% *
Invoice ID				
692-92-5582	*Food and beverages*	*19742.4*	*3*	*2961.36*
351-62-0822	*Fashion accessories*	*5212.8*	*4*	*1042.56*
529-56-3974	*Electronic accessories*	*9183.6*	*4*	*1836.72*
299-46-1805	*Sports and travel*	*33739.2*	*6*	*10121.76*
319-50-3348	*Home and lifestyle*	*14508.0*	*2*	*1450.80*
...	*...*	*...*	*...*	*...*
148-41-7930	*Health and beauty*	*35985.6*	*7*	*12594.96*
189-40-5216	*Electronic accessories*	*34693.2*	*7*	*12142.62*
267-62-7380	*Electronic accessories*	*29642.4*	*10*	*14821.20*
652-49-6720	*Electronic accessories*	*21942.0*	*1*	*1097.10*
233-67-5758	*Health and beauty*	*14526.0*	*1*	*726.30*

	Total	*Date*	*Time*	*Payment*	*cogs *
Invoice ID					
692-92-5582	*62188.56*	*2019-02-20*	*13:27*	*Card*	*59227.2*
351-62-0822	*21893.76*	*2019-02-06*	*18:07*	*Epay*	*20851.2*
529-56-3974	*38571.12*	*2019-03-09*	*17:03*	*Cash*	*36734.4*
299-46-1805	*212556.96*	*2019-01-15*	*16:19*	*Cash*	*202435.2*
319-50-3348	*30466.80*	*2019-03-11*	*15:30*	*Epay*	*29016.0*
...	*...*	*...*	*...*	*...*	*...*
148-41-7930	*264494.16*	*2019-01-23*	*10:33*	*Cash*	*251899.2*
189-40-5216	*254995.02*	*2019-01-09*	*11:40*	*Cash*	*242852.4*
267-62-7380	*311245.20*	*2019-03-29*	*19:12*	*Epay*	*296424.0*
652-49-6720	*23039.10*	*2019-02-18*	*11:40*	*Epay*	*21942.0*
233-67-5758	*15252.30*	*2019-01-29*	*13:46*	*Epay*	*14526.0*

	gross margin percentage	*gross income*	*Rating*
Invoice ID			
692-92-5582	*4.761905*	*2961.36*	*5.9*
351-62-0822	*4.761905*	*1042.56*	*4.5*
529-56-3974	*4.761905*	*1836.72*	*6.8*
299-46-1805	*4.761905*	*10121.76*	*4.5*
319-50-3348	*4.761905*	*1450.80*	*4.4*
...	*...*	*...*	*...*
148-41-7930	*4.761905*	*12594.96*	*6.1*
189-40-5216	*4.761905*	*12142.62*	*6.0*
267-62-7380	*4.761905*	*14821.20*	*4.3*
652-49-6720	*4.761905*	*1097.10*	*5.9*
233-67-5758	*4.761905*	*726.30*	*6.2*

[1000 rows x 16 columns]

 # To convert time in the dataset to standard python time

Syntax: merged_data['Time']=pd.to_datetime(merged_data['Time'])

```
merged_data
```

output:

```
            Branch          City Customer type  Gender  \
Invoice ID
692-92-5582      B         Abuja        Member  Female
351-62-0822      B         Abuja        Member  Female
529-56-3974      B         Abuja        Member    Male
299-46-1805      B         Abuja        Member  Female
319-50-3348      B         Abuja        Normal  Female
...            ...           ...           ...     ...
148-41-7930      C  Port Harcourt       Normal    Male
189-40-5216      C  Port Harcourt       Normal    Male
267-62-7380      C  Port Harcourt       Member    Male
652-49-6720      C  Port Harcourt       Member  Female
233-67-5758      C  Port Harcourt       Normal    Male

                       Product line  Unit price  Quantity    Tax 5%  \
Invoice ID
692-92-5582      Food and beverages     19742.4         3   2961.36
351-62-0822     Fashion accessories      5212.8         4   1042.56
529-56-3974  Electronic accessories      9183.6         4   1836.72
299-46-1805       Sports and travel     33739.2         6  10121.76
319-50-3348      Home and lifestyle     14508.0         2   1450.80
...                             ...         ...       ...       ...
148-41-7930       Health and beauty     35985.6         7  12594.96
189-40-5216  Electronic accessories     34693.2         7  12142.62
267-62-7380  Electronic accessories     29642.4        10  14821.20
652-49-6720  Electronic accessories     21942.0         1   1097.10
233-67-5758       Health and beauty     14526.0         1    726.30

                 Total        Date                Time Payment      cogs  \
Invoice ID
692-92-5582   62188.56 2019-02-20 2023-09-12 13:27:00    Card   59227.2
351-62-0822   21893.76 2019-02-06 2023-09-12 18:07:00    Epay   20851.2
529-56-3974   38571.12 2019-03-09 2023-09-12 17:03:00    Cash   36734.4
299-46-1805  212556.96 2019-01-15 2023-09-12 16:19:00    Cash  202435.2
319-50-3348   30466.80 2019-03-11 2023-09-12 15:30:00    Epay   29016.0
...                ...        ...                 ...     ...       ...
148-41-7930  264494.16 2019-01-23 2023-09-12 10:33:00    Cash  251899.2
189-40-5216  254995.02 2019-01-09 2023-09-12 11:40:00    Cash  242852.4
267-62-7380  311245.20 2019-03-29 2023-09-12 19:12:00    Epay  296424.0
```

```
652-49-6720   23039.10 2019-02-18 2023-09-12 11:40:00    Epay   21942.0
233-67-5758   15252.30 2019-01-29 2023-09-12 13:46:00    Epay   14526.0

             gross margin percentage  gross income  Rating
Invoice ID
692-92-5582                 4.761905       2961.36     5.9
351-62-0822                 4.761905       1042.56     4.5
529-56-3974                 4.761905       1836.72     6.8
299-46-1805                 4.761905      10121.76     4.5
319-50-3348                 4.761905       1450.80     4.4
...                              ...           ...     ...
148-41-7930                 4.761905      12594.96     6.1
189-40-5216                 4.761905      12142.62     6.0
267-62-7380                 4.761905      14821.20     4.3
652-49-6720                 4.761905       1097.10     5.9
233-67-5758                 4.761905        726.30     6.2

[1000 rows x 16 columns]
```

 # To simplify date and time into year, month, day, and hour

```
Syntax: merged_data['Year']= merged_data['Date'].dt.year
        merged_data['Month'] = merged_data['Date'].dt.month
        merged_data['Day'] = merged_data['Date'].dt.day
        merged_data['Hour']= merged_data['Time'].dt.hour

        merged_data
```

output:

```
            Branch           City Customer type  Gender  \
Invoice ID
692-92-5582      B          Abuja        Member  Female
351-62-0822      B          Abuja        Member  Female
529-56-3974      B          Abuja        Member    Male
299-46-1805      B          Abuja        Member  Female
319-50-3348      B          Abuja        Normal  Female
...            ...            ...           ...     ...
148-41-7930      C  Port Harcourt        Normal    Male
189-40-5216      C  Port Harcourt        Normal    Male
267-62-7380      C  Port Harcourt        Member    Male
652-49-6720      C  Port Harcourt        Member  Female
233-67-5758      C  Port Harcourt        Normal    Male

                       Product line  Unit price  Quantity    Tax 5%  \
Invoice ID
692-92-5582       Food and beverages     19742.4         3   2961.36
351-62-0822      Fashion accessories      5212.8         4   1042.56
529-56-3974   Electronic accessories      9183.6         4   1836.72
299-46-1805        Sports and travel     33739.2         6  10121.76
```

```
319-50-3348      Home and lifestyle     14508.0         2   1450.80
...                       ...         ...       ...       ...
148-41-7930       Health and beauty     35985.6         7  12594.96
189-40-5216  Electronic accessories     34693.2         7  12142.62
267-62-7380  Electronic accessories     29642.4        10  14821.20
652-49-6720  Electronic accessories     21942.0         1   1097.10
233-67-5758       Health and beauty     14526.0         1    726.30

                 Total       Date                Time Payment      cogs  \
Invoice ID
692-92-5582   62188.56 2019-02-20 2023-09-12 13:27:00    Card   59227.2
351-62-0822   21893.76 2019-02-06 2023-09-12 18:07:00    Epay   20851.2
529-56-3974   38571.12 2019-03-09 2023-09-12 17:03:00    Cash   36734.4
299-46-1805  212556.96 2019-01-15 2023-09-12 16:19:00    Cash  202435.2
319-50-3348   30466.80 2019-03-11 2023-09-12 15:30:00    Epay   29016.0
...                ...        ...                 ...     ...       ...
148-41-7930  264494.16 2019-01-23 2023-09-12 10:33:00    Cash  251899.2
189-40-5216  254995.02 2019-01-09 2023-09-12 11:40:00    Cash  242852.4
267-62-7380  311245.20 2019-03-29 2023-09-12 19:12:00    Epay  296424.0
652-49-6720   23039.10 2019-02-18 2023-09-12 11:40:00    Epay   21942.0
233-67-5758   15252.30 2019-01-29 2023-09-12 13:46:00    Epay   14526.0

             gross margin percentage  gross income  Rating  Year  Month  Day  \
Invoice ID
692-92-5582                 4.761905       2961.36     5.9  2019      2   20
351-62-0822                 4.761905       1042.56     4.5  2019      2    6
529-56-3974                 4.761905       1836.72     6.8  2019      3    9
299-46-1805                 4.761905      10121.76     4.5  2019      1   15
319-50-3348                 4.761905       1450.80     4.4  2019      3   11
...                              ...           ...     ...   ...    ...  ...
148-41-7930                 4.761905      12594.96     6.1  2019      1   23
189-40-5216                 4.761905      12142.62     6.0  2019      1    9
267-62-7380                 4.761905      14821.20     4.3  2019      3   29
652-49-6720                 4.761905       1097.10     5.9  2019      2   18
233-67-5758                 4.761905        726.30     6.2  2019      1   29

             Hour
Invoice ID
692-92-5582    13
351-62-0822    18
529-56-3974    17
299-46-1805    16
319-50-3348    15
...           ...
148-41-7930    10
189-40-5216    11
267-62-7380    19
652-49-6720    11
233-67-5758    13
```

[1000 rows x 20 columns]

 # To obtain the unique hours in the dataset

Syntax: merged_data['Hour'].nunique()

Output: 11

Syntax: merged_data['Hour'].unique()

Output: array([13, 18, 17, 16, 15, 10, 12, 19, 14, 11, 20], dtype=int64)

There are 11 unique hours in the dataset

 # To obtain Unique Values in Column

syntax: merged_data['City'].unique()

Output: array(['Abuja', 'Lagos', 'Port Harcourt'], dtype=object)

#To obtain counts of the unique cities in the dataset

Syntax: merged_data['City'].value_counts()

Output: Lagos 340
Abuja 332
Port Harcourt 328
Name: City, dtype: int64

 # To obtain Columns that contain categorical values in the dataset

*Syntax: categorical =[col **for** col **in** merged_data.columns **if** merged_data[col].dtype == 'object']*

categorical

Output: ['Branch', 'City', 'Customer type', 'Gender', 'Product line', 'Payment']

#To obtain unique values in the Customer Type column

Syntax: merged_data['Customer type'].unique()

Output: array(['Member', 'Normal'], dtype=object)

 # To obtain value count of the Customer Type column

Syntax: merged_data['Customer type'].value_counts()

Output: Member 501
Normal 499
Name: Customer type, dtype: int64

#To obtain the unique values in the Gender column and their counts

Syntax: merged_data['Gender'].unique()

Output: array(['Female', 'Male'], dtype=object)

Syntax: merged_data['Gender'].value_counts()

Output: Female 501
Male 499
Name: Gender, dtype: int64

 # To obtain unique values in the Product line column and their count

Syntax: merged_data['Product line'].unique()

Output: array(['Food and beverages', 'Fashion accessories',
'Electronic accessories', 'Sports and travel',
'Home and lifestyle', 'Health and beauty'], dtype=object)

Syntax: merged_data['Product line'].value_counts()

Output: Fashion accessories 178
Food and beverages 174
Electronic accessories 170
Sports and travel 166
Home and lifestyle 160
Health and beauty 152
Name: Product line, dtype: int64

 # TO obtain unique values in the Payment column and their count

Syntax: merged_data['Payment'].unique()

Output: array(['Card', 'Epay', 'Cash'], dtype=object)

columns=['Branch', 'City', 'Customer type', 'Gender', 'Product line', 'Payment']

#Aggregation with Groupby

#Grouping the Dataset by the type of Product line

Syntax: product=merged_data.groupby('Product line')

merged_data['Product line'].unique()

Output: array(['Food and beverages', 'Fashion accessories', 'Electronic accessories', 'Sports and travel', 'Home and lifestyle', 'Health and beauty'], dtype=object)

#To obtain datasets for a specific Group (e.g For Food and Beverages)

syntax: product.get_group('Food and beverages').head(10)

Output:

```
            Branch   City Customer type  Gender        Product line  \
Invoice ID
692-92-5582      B  Abuja        Member  Female  Food and beverages
145-94-9061      B  Abuja        Normal  Female  Food and beverages
727-46-3608      B  Abuja        Member  Female  Food and beverages
510-95-6347      B  Abuja        Member  Female  Food and beverages
847-38-7188      B  Abuja        Normal  Female  Food and beverages
548-46-9322      B  Abuja        Normal    Male  Food and beverages
316-55-4634      B  Abuja        Member    Male  Food and beverages
414-12-7047      B  Abuja        Normal    Male  Food and beverages
895-66-0685      B  Abuja        Member    Male  Food and beverages
790-29-1172      B  Abuja        Normal  Female  Food and beverages

             Unit price  Quantity   Tax 5%      Total       Date  \
Invoice ID
692-92-5582     19742.4         3  2961.36   62188.56 2019-02-20
145-94-9061     31809.6         5  7952.40  167000.40 2019-01-25
727-46-3608      7203.6         9  3241.62   68074.02 2019-02-06
510-95-6347     17467.2         3  2620.08   55021.68 2019-03-05
847-38-7188     34804.8         3  5220.72  109635.12 2019-01-26
548-46-9322     14364.0        10  7182.00  150822.00 2019-02-20
316-55-4634     28818.0         5  7204.50  151294.50 2019-01-26
414-12-7047      7124.4         8  2849.76   59844.96 2019-01-18
895-66-0685      6508.8         3   976.32   20502.72 2019-03-05
790-29-1172     20642.4         3  3096.36   65023.56 2019-03-10

                           Time Payment      cogs  gross margin percentage  \
Invoice ID
692-92-5582 2023-09-12 13:27:00    Card   59227.2                 4.761905
145-94-9061 2023-09-12 19:48:00    Cash  159048.0                 4.761905
727-46-3608 2023-09-12 15:47:00    Epay   64832.4                 4.761905
510-95-6347 2023-09-12 18:17:00    Epay   52401.6                 4.761905
847-38-7188 2023-09-12 19:56:00    Epay  104414.4                 4.761905
548-46-9322 2023-09-12 15:24:00    Card  143640.0                 4.761905
316-55-4634 2023-09-12 12:45:00    Card  144090.0                 4.761905
414-12-7047 2023-09-12 12:04:00    Epay   56995.2                 4.761905
```

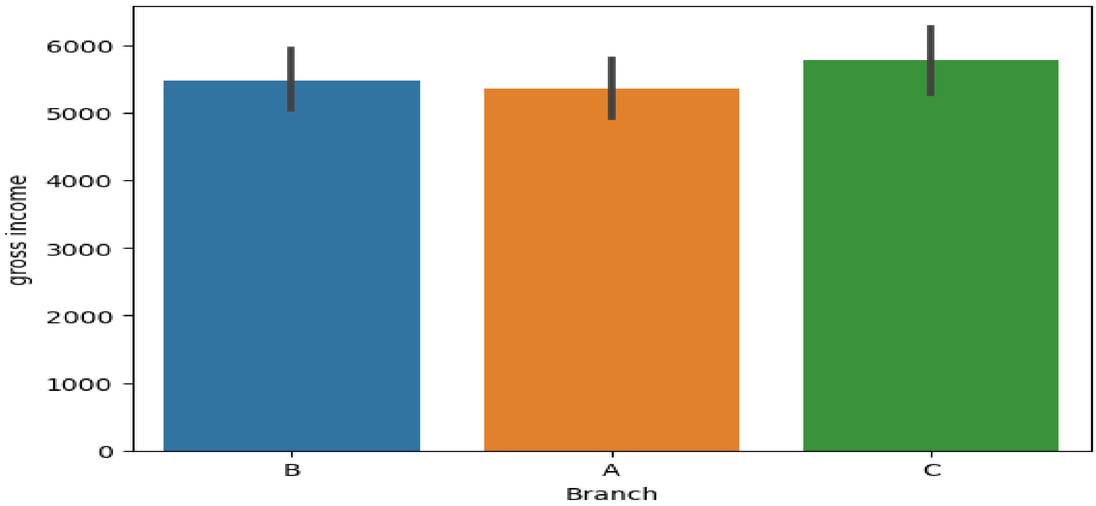
```
895-66-0685 2023-09-12 19:46:00    Epay   19526.4             4.761905
790-29-1172 2023-09-12 18:59:00    Card   61927.2             4.761905

            gross income  Rating  Year  Month  Day  Hour
Invoice ID
692-92-5582       2961.36     5.9  2019      2   20    13
145-94-9061       7952.40     9.6  2019      1   25    19
727-46-3608       3241.62     4.1  2019      2    6    15
510-95-6347       2620.08     4.0  2019      3    5    18
847-38-7188       5220.72     6.4  2019      1   26    19
548-46-9322       7182.00     5.9  2019      2   20    15
316-55-4634       7204.50     9.4  2019      1   26    12
414-12-7047       2849.76     8.7  2019      1   18    12
895-66-0685        976.32     8.0  2019      3    5    19
790-29-1172       3096.36     7.9  2019      3   10    18
```

 # Data Visualization

 # Visualize gross income for each branch using a bar plot

Syntax: sns.barplot(x='Branch',y='gross income',data=merged_data)

Output: <AxesSubplot:xlabel='Branch', ylabel='gross income'>

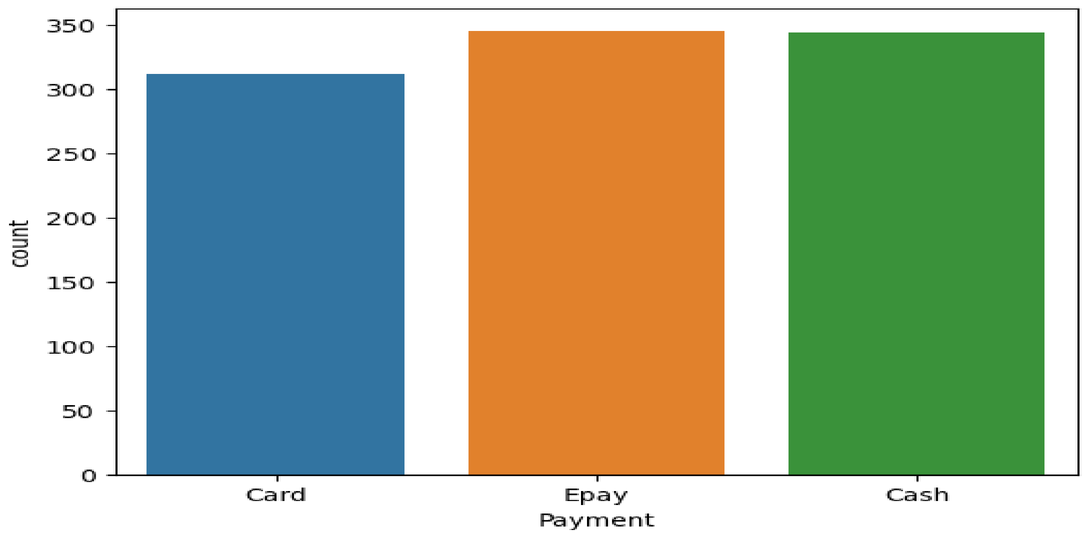


From the above bar plot, it is shown that Branch C has the most gross income and Branch A has the least.

 # Visualize the count of payment type using a countplot

Syntax: sns.countplot(x='Payment',data=merged_data)

Output: <AxesSubplot:xlabel='Payment', ylabel='count'>

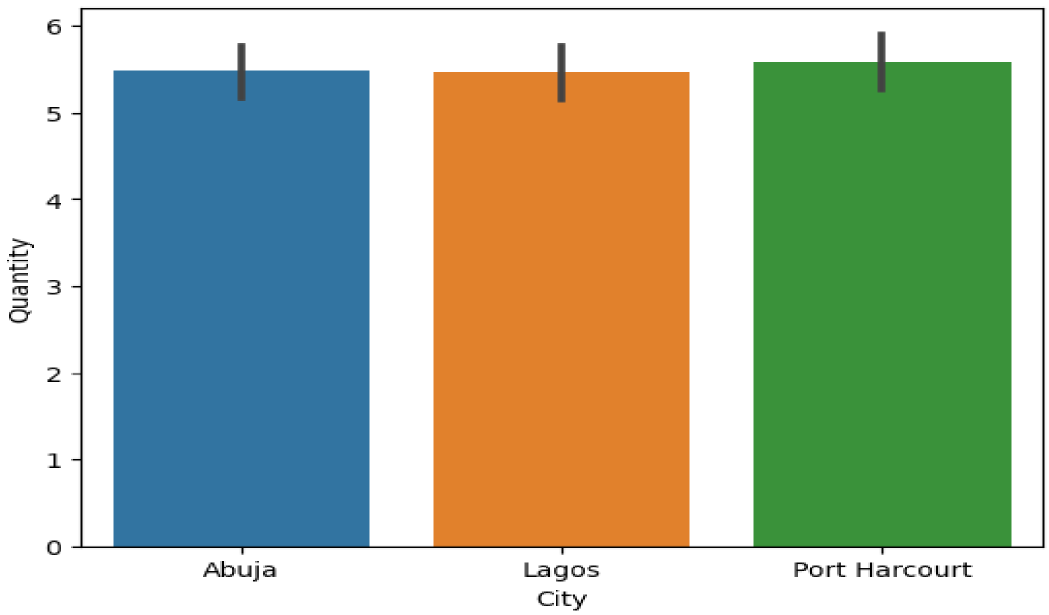


From the above plot, it is shown that Epay and cash were the joint most used payment type and card was the least payment method used across the three branches.

 # To Visualize the city with the Most quantity of goods sold

Syntax: sns.barplot(x='City',y='Quantity', data= merged_data)

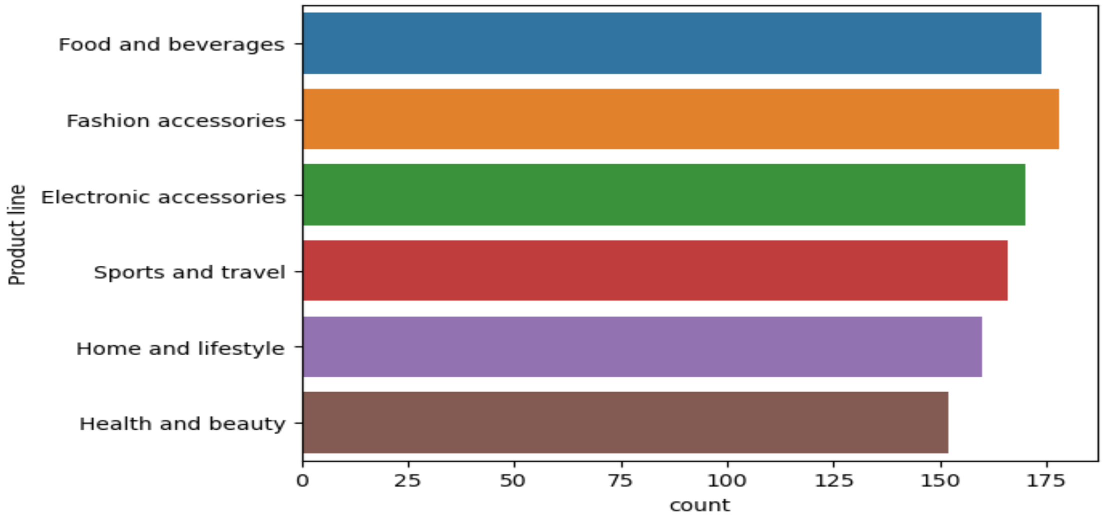
Output: <AxesSubplot:xlabel='City', ylabel='Quantity'>



From the above bar plot, it is shown that Port Harcourt sold the most quantity of goods.

 # To visualize the count of each product line sold

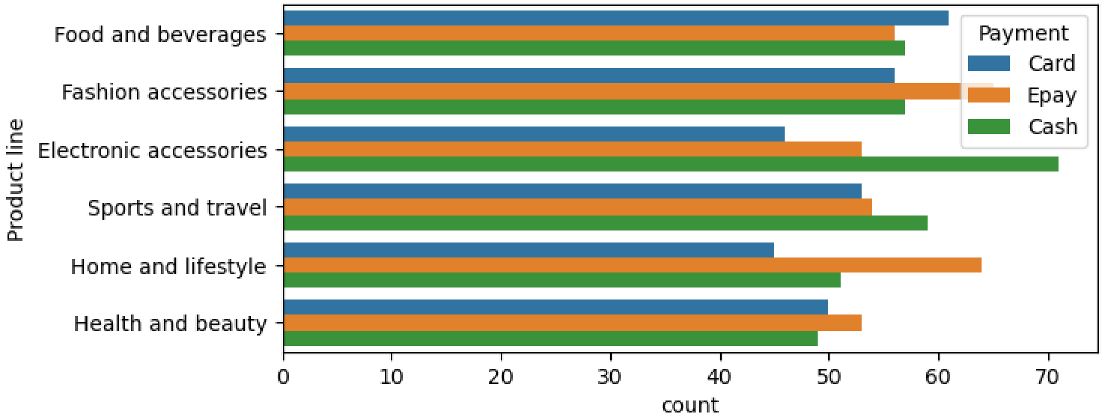
Syntax: sns.countplot(y='Product line', data= merged_data)
Output: mp.rcParams['figure.figsize']=(7,3)



 # To visualize the count of each product line sold and payment type

Syntax: sns.countplot(y='Product line', data= merged_data, hue='Payment')

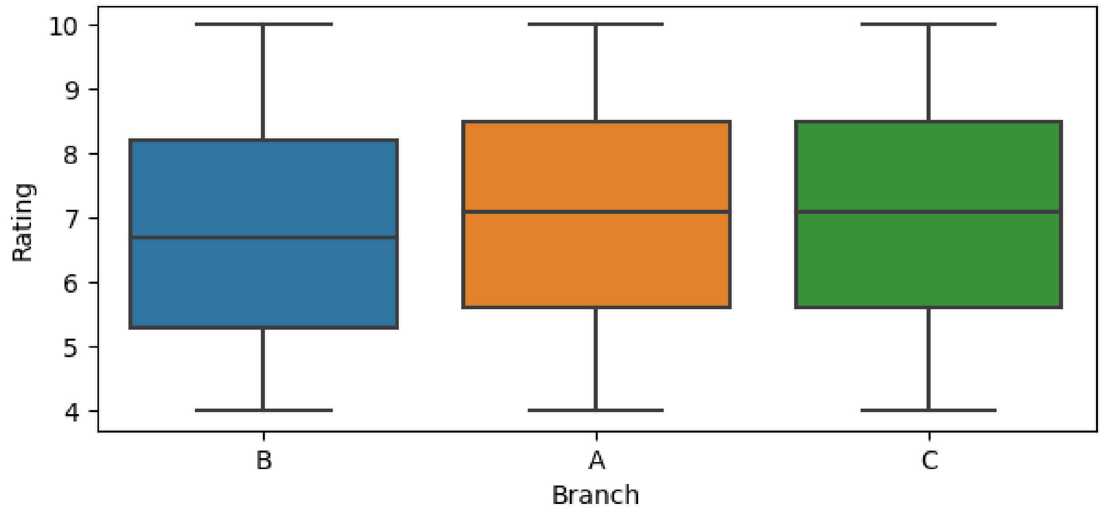
Output: <AxesSubplot:xlabel='count', ylabel='Product line'>



 # To visualize the rating for each branch using a boxplot

Syntax: sns.boxplot(x='Branch',y='Rating',data=merged_data)

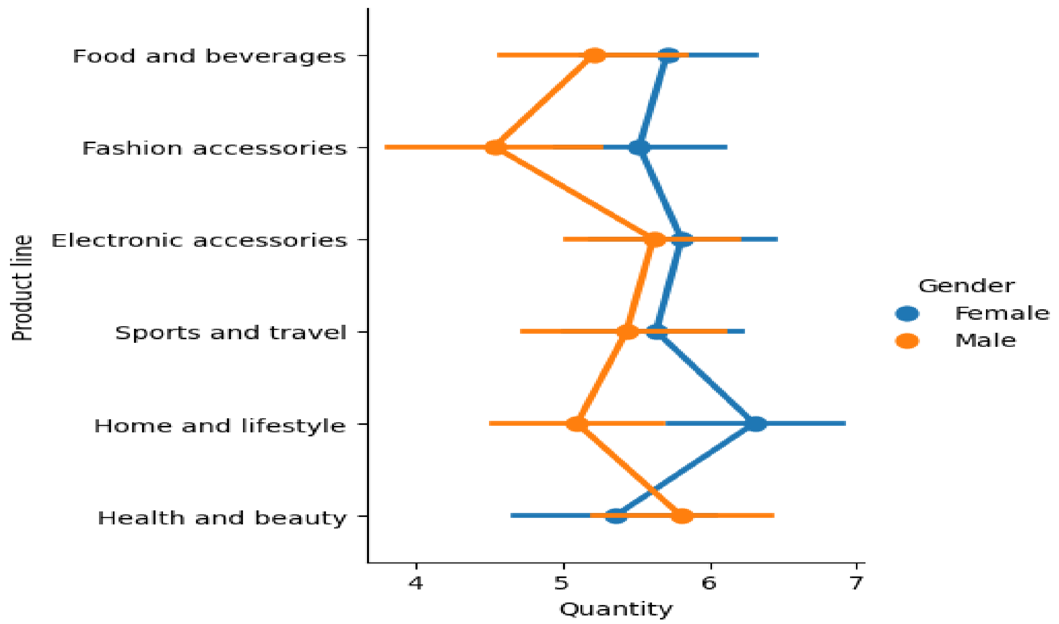
Output: <AxesSubplot:xlabel='Branch', ylabel='Rating'>



From the above, it is shown that Branch A and C has a joint highest rating and branch B has the least rating.

 # To visualize the Quantity of Product line for each gender using a catplot

Syntax: sns.catplot(y='Product line',x = 'Quantity', hue = 'Gender', data = merged_data, kind='point')
Output: mp.rcParams['figure.figsize']=(5,6)

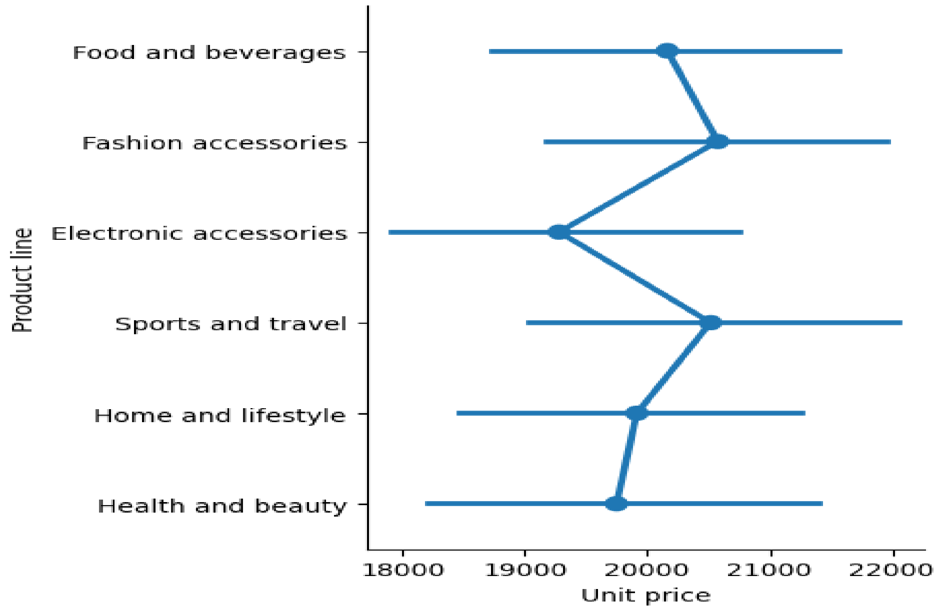


From the above, it is shown that the female gender bought more products than male. Females bought more Home and lifestyle product than any other product and bought health and beauty product the least the male gender bought more Health and beauty product than any other product and bought Fashion accessories the least.

 # To visualize the unit price for each product line using a catplot

Syntax: sns.catplot(y = 'Product line',x = 'Unit price',data = merged_data, kind='point')

Output: <seaborn.axisgrid.FacetGrid at 0x2379d8d8250>

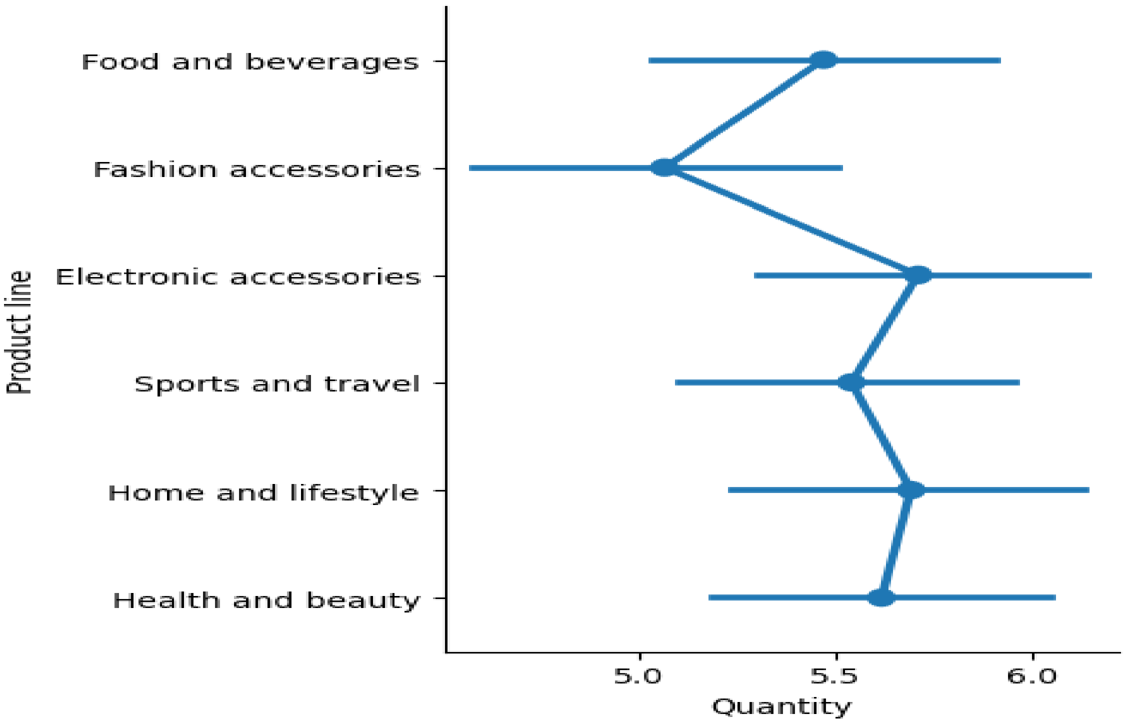


From the above chart, it is shown from the chart that Fashion accessories has the highest unit price, while, electronic accessories has the least unit price.

 # To visualize Quantity of each Product line sold using a catplot

Syntax: sns.catplot(y = 'Product line',x = 'Quantity',data = merged_data, kind='point')

Output:

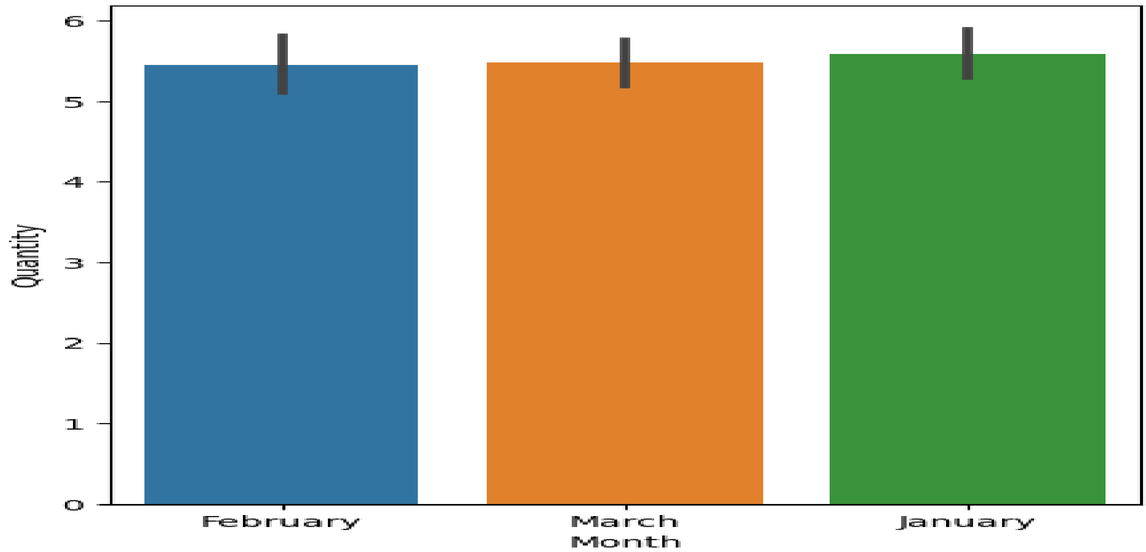


From the above chart, it is shown that Electronic accessories were sold more than any other p roduct Fashion accessories was the least product sold.

 # To visualize quantity of Goods sold for each of the 3 months

*Syntax: merged_data['Month']=merged_data['Month'].map(**lambda** x: calendar.month_nam e[x])*

Output: sns.barplot(x = 'Month',y = 'Quantity', data = merged_data)

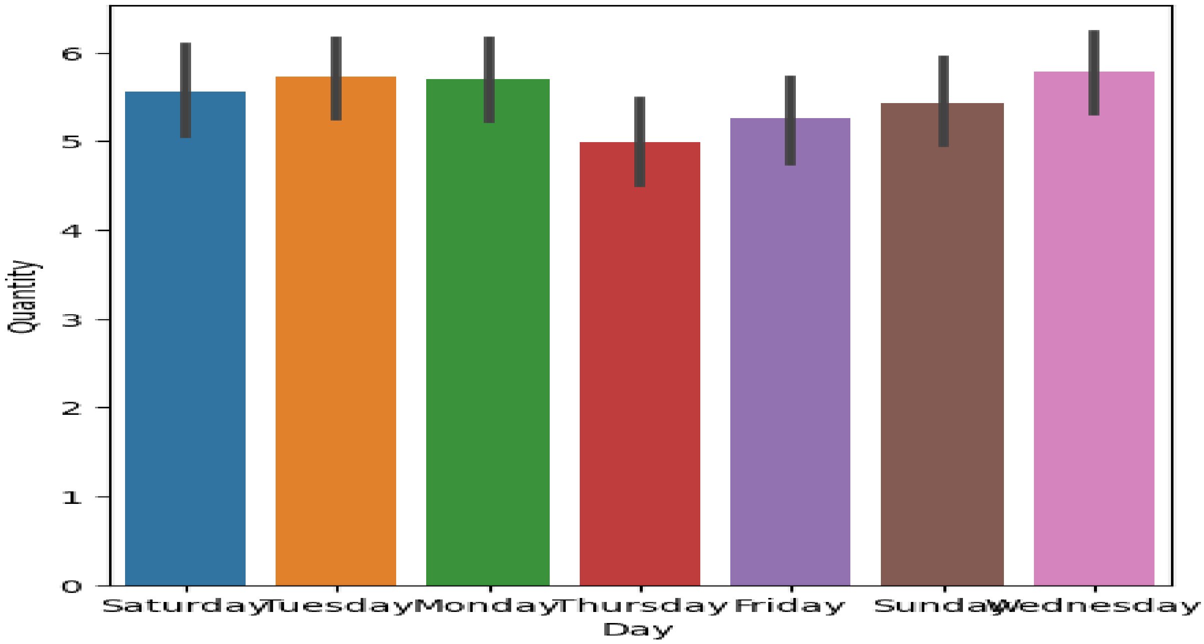


From the above bar plot, it is shown that more goods were sold in January than February and march

#To visualize quantity of goods sold for each day of the week.

Syntax: merged_data['Day']=merged_data['Day'].map(***lambda*** *x: calenda r.day_name[(x-1)%7])*

sns.barplot(x = 'Day', y = 'Quantity', data=merged_data)



From the bar plot, it is shown that Wednesday is the day that has the highest quantity of good s sold the least quantity of goods were sold on Thursday.

2.1.3.2 WORKING ON DS_SALARIES USING PYTHON PROGRAMMING LANGUAGE

 # Importing Python Libraries

import *pandas* ***as*** *pd*
import *numpy* ***as*** *np*
import *seaborn* ***as*** *sns*
import *matplotlib.pyplot* ***as*** *plt*
%matplotlib inline

#Loading ds_salaries dataset into jupyter notebook

Syntax: Salaries = pd.read_csv('ds_salaries.csv').drop('Unnamed: 0', axis = 1)

Salaries

Output:

```
    work_year experience_level employment_type                 job_title  \
0        2020               MI              FT            Data Scientist
1        2020               SE              FT  Machine Learning Scientist
2        2020               SE              FT          Big Data Engineer
3        2020               MI              FT       Product Data Analyst
4        2020               SE              FT   Machine Learning Engineer
..        ...              ...             ...                        ...
602      2022               SE              FT             Data Engineer
603      2022               SE              FT             Data Engineer
604      2022               SE              FT              Data Analyst
605      2022               SE              FT              Data Analyst
606      2022               MI              FT              AI Scientist

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0     70000             EUR          79833                 DE             0
1    260000             USD         260000                 JP             0
2     85000             GBP         109024                 GB            50
3     20000             USD          20000                 HN             0
4    150000             USD         150000                 US            50
..      ...             ...            ...                ...           ...
602  154000             USD         154000                 US           100
603  126000             USD         126000                 US           100
604  129000             USD         129000                 US             0
605  150000             USD         150000                 US           100
606  200000             USD         200000                 IN           100

    company_location company_size
0                 DE            L
1                 JP            S
2                 GB            M
3                 HN            S
4                 US            L
..               ...          ...
602               US            M
603               US            M
604               US            M
605               US            M
606               US            L

[607 rows x 11 columns]
```

 # DATA EXPLORATION

 # To obtain general information on the dataset

Syntax: Salaries.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   work_year           607 non-null    int64
 1   experience_level    607 non-null    object
 2   employment_type     607 non-null    object
 3   job_title           607 non-null    object
 4   salary              607 non-null    int64
 5   salary_currency     607 non-null    object
 6   salary_in_usd       607 non-null    int64
 7   employee_residence  607 non-null    object
 8   remote_ratio        607 non-null    int64
 9   company_location    607 non-null    object
 10  company_size        607 non-null    object
dtypes: int64(4), object(7)
memory usage: 52.3+ KB
```

In the Salaries dataset, there are a total of 11 columns, seven (7) are objects and four(4) are in tegers

 # To obtain statistical information about the dataset

Syntax: Salaries.describe()

Output:

```
         work_year        salary  salary_in_usd  remote_ratio
count   607.000000  6.070000e+02     607.000000     607.00000
mean   2021.405272  3.240001e+05  112297.869852      70.92257
std       0.692133  1.544357e+06   70957.259411      40.70913
min    2020.000000  4.000000e+03    2859.000000       0.00000
25%    2021.000000  7.000000e+04   62726.000000      50.00000
50%    2022.000000  1.150000e+05  101570.000000     100.00000
75%    2022.000000  1.650000e+05  150000.000000     100.00000
max    2022.000000  3.040000e+07  600000.000000     100.00000
```

In the dataset, the min, mean and max work year are 2020,2021 and 2022 respectively. The m in, mean and max salary in USD are 2859, 112297.87, and 600000 USD respectively

 # To obtain the number of rows and columns in the dataset

Syntax: Salaries.shape

Output: (607, 11)

In the dataset, there are 607 rows and 11 columns.

 # To obtain the size of the dataset

Syntax: Salaries.size

Output: 6677

In the dataset, there are a total of 6677 elements

 # To obtain the names of columns in the dataset

Syntax: Salaries.columns

Output:

Index(['work_year', 'experience_level', 'employment_type', 'job_title','salary', 'salary_currency', 'salary_in_usd', 'employee_residence','remote_ratio','company_location', 'company_size'], dtype='object')

 # To determine the number of Null or empty values in the dataset

Syntax: Salaries.isnull().sum()

Output:

```
work_year            0
experience_level     0
employment_type      0
job_title            0
salary               0
salary_currency      0
salary_in_usd        0
employee_residence   0
remote_ratio         0
company_location     0
company_size         0
dtype: int64
```

In the data frame, there are no null or empty cell.

 # To replace abbreviations in the dataset to their full meaning

Syntax:

Salaries['experience_level']=Salaries['experience_level'].replace({'MI':'Mid level','EX':'Excecutive','EN':'Entry level','SE':'Senior'})

```
Salaries['employment_type'] = Salaries['employment_type'].replace({'FT':'Full time','PT':'Part time','CT':'Contract','FL':'Freelance'})


Salaries['company_size'] =
Salaries['company_size'].replace({'M':'Medium','L':'Large','S':'Small'})


Salaries['remote_ratio'] =
Salaries['remote_ratio'].replace({0:'Onsite',50:'Hybrid',100:'Remote'})


Salaries['employee_residence'] = Salaries['employee_residence'].replace({'US':'United States','GB':'GreatBritain','CA':'Canada','DE':'Denmark','IN':'India','FR':'France','ES':'Spain','JP':'Japan','GR':'Greece','NL':'Netherlands','PT':'Portugal'})


Salaries['company_location'] = Salaries['company_location'].replace({'US':'United States','GB':'GreatBritain','CA':'Canada','DE':'Denmark','IN':'India','FR':'France','ES':'Spain','JP':'Japan','GR':'Greece','NL':'Netherlands','PT':'Portugal'})

Salaries['salary']= Salaries['salary'].astype(str) + ' '  + Salaries['salary_currency']

salaries_2 = Salaries.drop('salary_currency', axis =1)

salaries_2
```

Output:

```
    work_year experience_level employment_type                 job_title  \
0        2020        Mid level       Full time            Data Scientist
1        2020           Senior       Full time  Machine Learning Scientist
2        2020           Senior       Full time           Big Data Engineer
3        2020        Mid level       Full time        Product Data Analyst
4        2020           Senior       Full time   Machine Learning Engineer
..        ...              ...             ...                         ...
602      2022           Senior       Full time               Data Engineer
603      2022           Senior       Full time               Data Engineer
604      2022           Senior       Full time                Data Analyst
605      2022           Senior       Full time                Data Analyst
606      2022        Mid level       Full time                AI Scientist

         salary  salary_in_usd employee_residence remote_ratio  \
0     70000 EUR          79833            Denmark       Onsite
1    260000 USD         260000              Japan       Onsite
2     85000 GBP         109024      Great Britain       Hybrid
3     20000 USD          20000                 HN       Onsite
4    150000 USD         150000      United States       Hybrid
..          ...            ...                ...          ...
602  154000 USD         154000      United States       Remote
603  126000 USD         126000      United States       Remote
604  129000 USD         129000      United States       Onsite
```

605 150000 USD   150000   United States   Remote
606 200000 USD   200000   India   Remote

```
    company_location company_size
0          Denmark        Large
1            Japan        Small
2      Great Britain       Medium
3              HN        Small
4      United States       Large
..             ...          ...
602    United States       Medium
603    United States       Medium
604    United States       Medium
605    United States       Medium
606    United States        Large

[607 rows x 10 columns]
```

 # To obtain the count of Unique values in the Work year column

Syntax: salaries_2['work_year'].value_counts()

```
Output: 2022    318
        2021    217
        2020     72
        Name: work_year, dtype: int64
```

#To obtain the count of Unique values in the experience level column

Syntax: salaries_2['experience_level'].value_counts()

```
Output: Senior        280
        Mid level      213
        Entry level     88
        Excecutive      26
        Name: experience_level, dtype: int64
```

 # To obtain the count of Unique values in the Employment type column

Syntax: salaries_2['employment_type'].value_counts()

```
Output: Full time    588
        Part time     10
        Contract       5
        Freelance      4
        Name: employment_type, dtype: int64
```

 # To obtain the count of Unique values in the job title column

Syntax: salaries_2['job_title'].value_counts()

Output: Data Scientist 143
Data Engineer 132
Data Analyst 97
Machine Learning Engineer 41
Research Scientist 16
Data Science Manager 12
Data Architect 11
Big Data Engineer 8
Machine Learning Scientist 8
Principal Data Scientist 7
AI Scientist 7
Data Science Consultant 7
Director of Data Science 7
Data Analytics Manager 7
ML Engineer 6
Computer Vision Engineer 6
BI Data Analyst 6
Lead Data Engineer 6
Data Engineering Manager 5
Business Data Analyst 5
Head of Data 5
Applied Data Scientist 5
Applied Machine Learning Scientist 4
Head of Data Science 4
Analytics Engineer 4
Data Analytics Engineer 4
Machine Learning Developer 3
Machine Learning Infrastructure Engineer 3
Lead Data Scientist 3
Computer Vision Software Engineer 3
Lead Data Analyst 3
Data Science Engineer 3
Principal Data Engineer 3
Principal Data Analyst 2
ETL Developer 2
Product Data Analyst 2
Director of Data Engineering 2
Financial Data Analyst 2
Cloud Data Engineer 2
Lead Machine Learning Engineer 1
NLP Engineer 1
Head of Machine Learning 1
3D Computer Vision Researcher 1
Data Specialist 1
Staff Data Scientist 1

Big Data Architect 1
Finance Data Analyst 1
Marketing Data Analyst 1
Machine Learning Manager 1
Data Analytics Lead 1
Name: job_title, dtype: int64

 # To obtain the count of Unique values in the employee residence column

Syntax: salaries_2['employee_residence'].value_counts().head(10)

Output: United States 332
Great Britain 44
India 30
Canada 29
Denmark 25
France 18
Spain 15
Greece 13
Japan 7
Portugal 6
Name: employee_residence, dtype: int64

 # To obtain the count of Unique values in the remote ratio column

Syntax: salaries_2['remote_ratio'].value_counts()

Output: Remote 381
Onsite 127
Hybrid 99
Name: remote_ratio, dtype: int64

#To obtain the count of Unique values in the Company location column

Syntax: salaries_2['company_location'].value_counts().head(10)

Output: United States 355
Great Britain 47
Canada 30
Denmark 28
India 24
France 15
Spain 14
Greece 11
Japan 6
Netherlands 4
Name: company_location, dtype: int64

To obtain the count of Unique values in the company size column

Syntax: salaries_2['company_size'].value_counts()

Medium 326
Large 198
Small 83
Name: company_size, dtype: int64

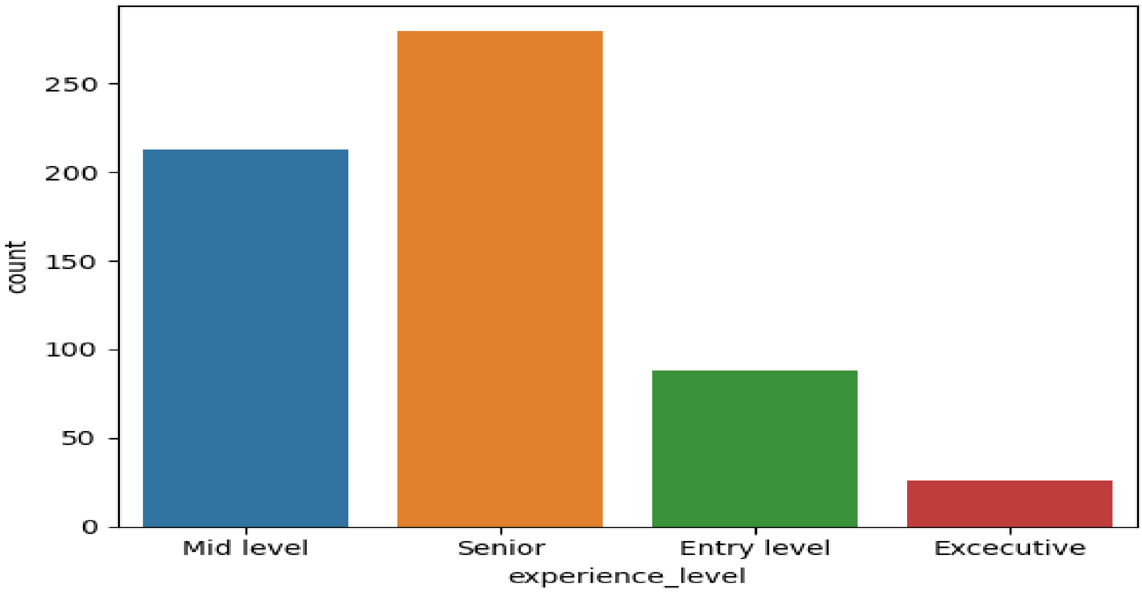
DATA VISUALIZATION

Univariate Visualization (Looks at one variable)

Get the count plot for the experience_level

Syntax: sns.countplot(data = salaries_2, x= 'experience_level')

Output: <AxesSubplot:xlabel='experience_level', ylabel='count'>

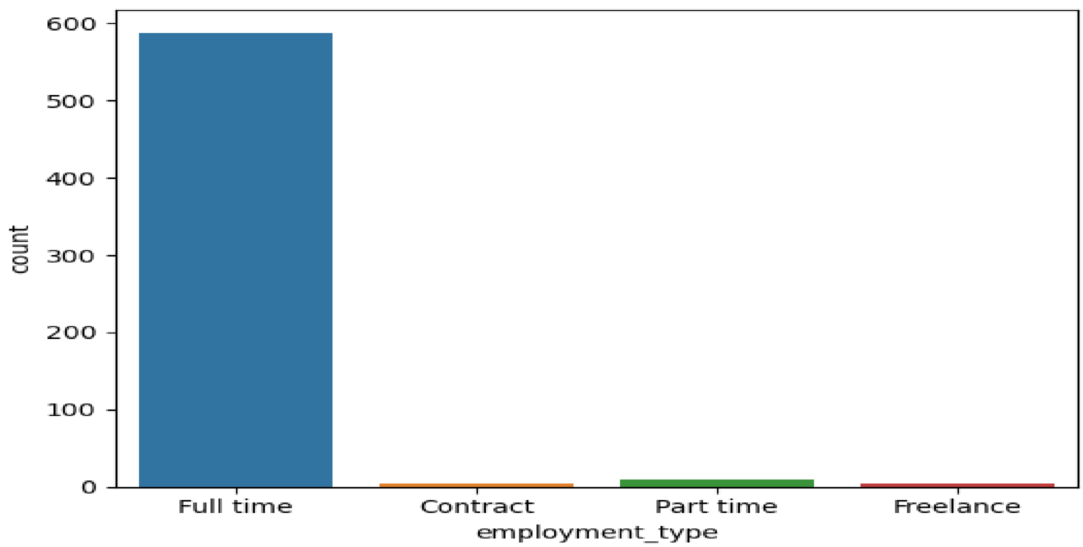


The Experience level represented the most in the dataset is Senior level, while Executive experience level is the least represented.

Get the countplot for employment_type

Syntax: sns.countplot(salaries_2['employment_type'])

Output: <AxesSubplot:xlabel='employment_type', ylabel='count'>

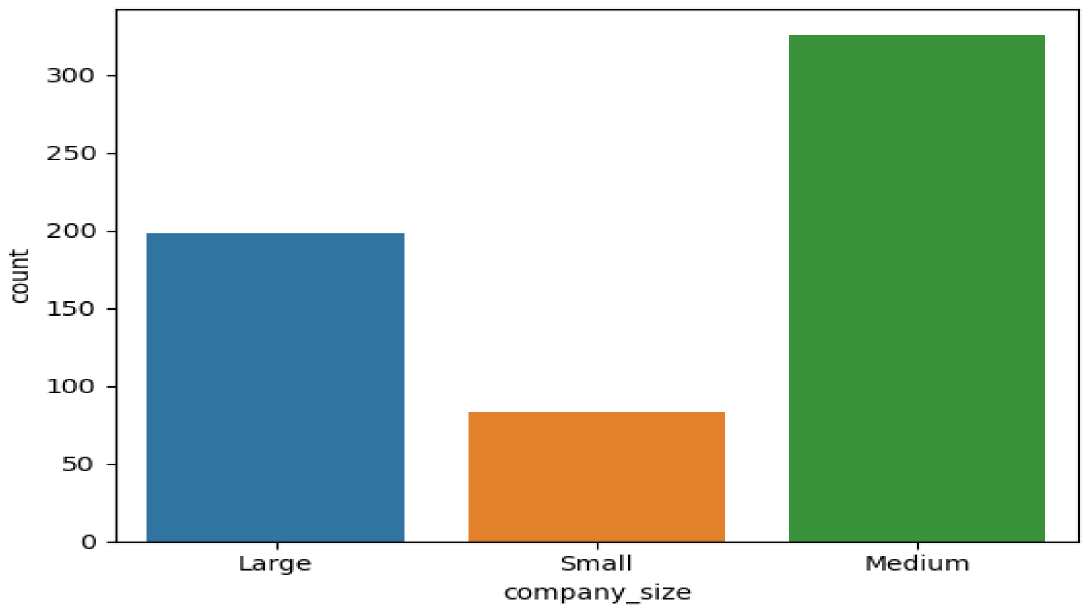


From the above plot, the employment type that is mostly represented is Full time employment.

 # Get the countplot for the company_size

Syntax: sns.countplot(salaries_2['company_size'])

Output: <AxesSubplot:xlabel='company_size', ylabel='count'>

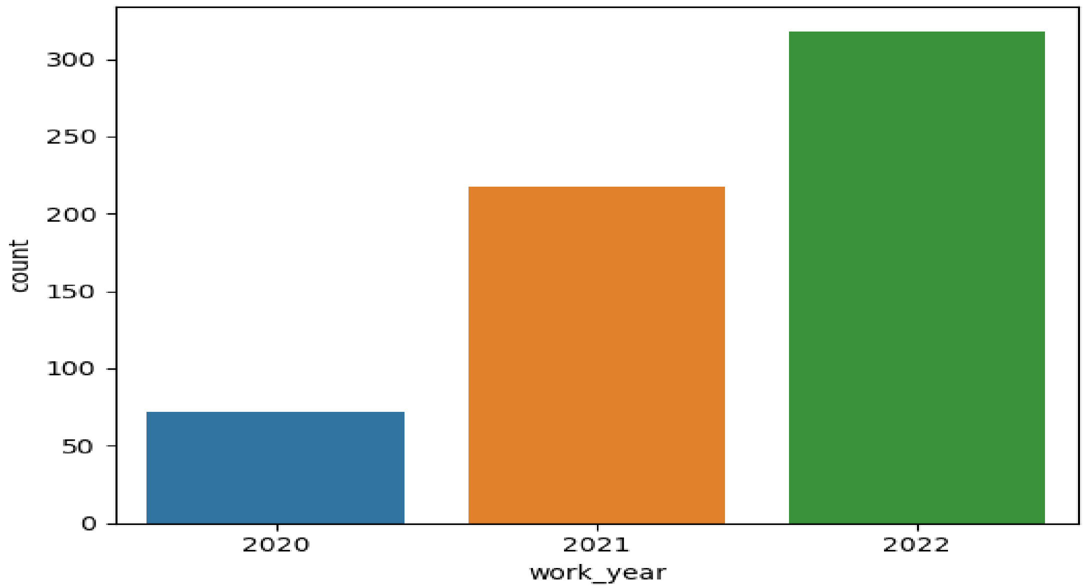


In the ds-salaries dataset, Medium company size is mostly represented followed by large Company. Small company size is the least represented.

 # Get the countplot for work_year

Syntax: sns.countplot(salaries_2['work_year'])

Output: <AxesSubplot:xlabel='work_year', ylabel='count'>

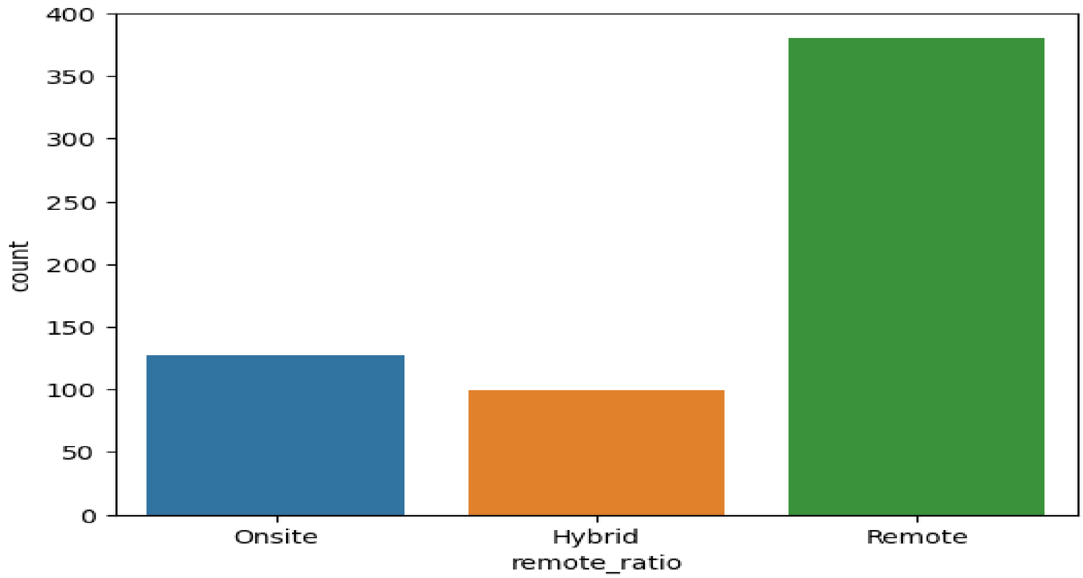


The above count plot shows that 2022 is the most represented work year in the dataset, followed by 2021. This could be because the demand for the captured job titles grew across the years.

 # Get the countplot for remote_ratio

Syntax: sns.countplot(salaries_2['remote_ratio'])

Output: <AxesSubplot:xlabel='remote_ratio', ylabel='count'>

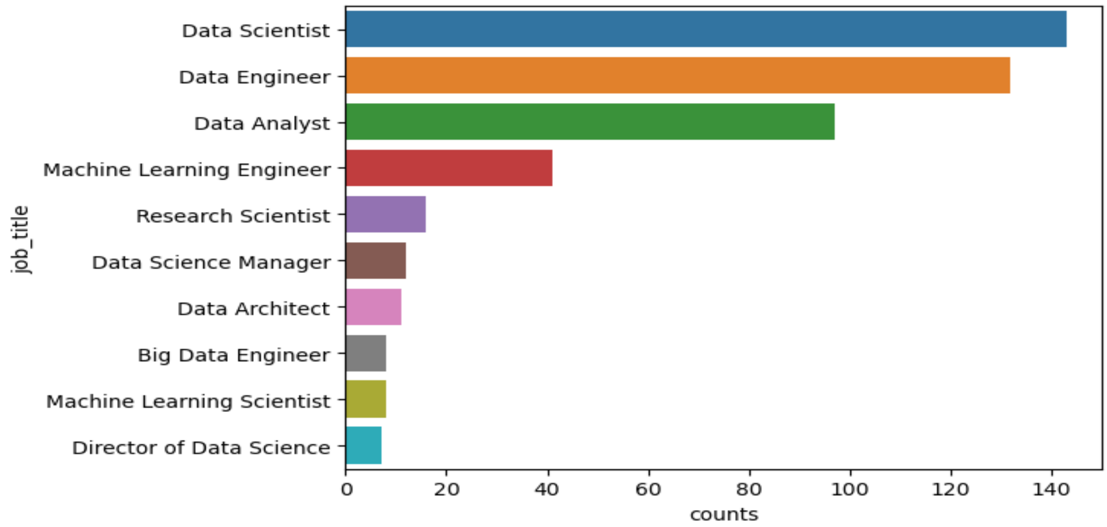


In the ds_salaries dataset, most jobs were remote.

 # Get the Total number of Each job_title in the data set

Syntax: sns.barplot(data=job_title_4,x='counts', y='job_title')

Output: <AxesSubplot:xlabel='counts', ylabel='job_title'>



In the ds_salaries dataset, Data scientist had the Highest number of jobs, followed by Data Engineer, and Data Analyst. Director of Data science is the least in the count of job title.

Bivariate Visualization (looks at two variables and their relationship)

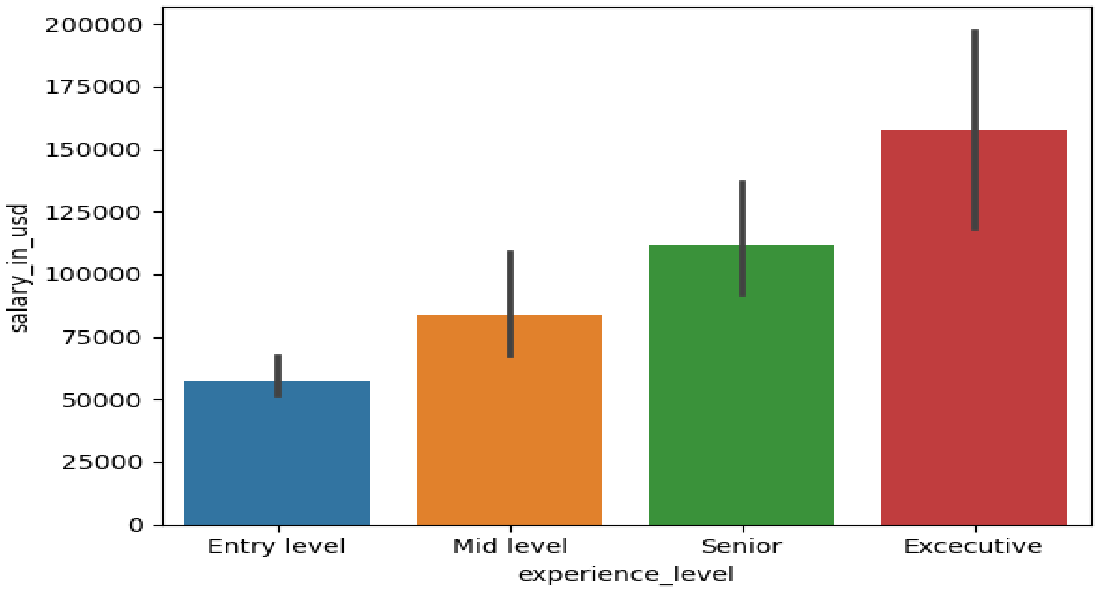
 # filtering the data set to show only company_location for canada

Syntax: data_canada = salaries_2[salaries_2['company_location'].isin(['Canada'])].sort_values('salary_in_usd').reset_index().drop('index', axis=1)

 # Obtaining a plot between experience_level and salary

Syntax: sns.barplot(data= data_canada, x='experience_level', y='salary_in_usd')

Output: <AxesSubplot:xlabel='experience_level', ylabel='salary_in_usd'>

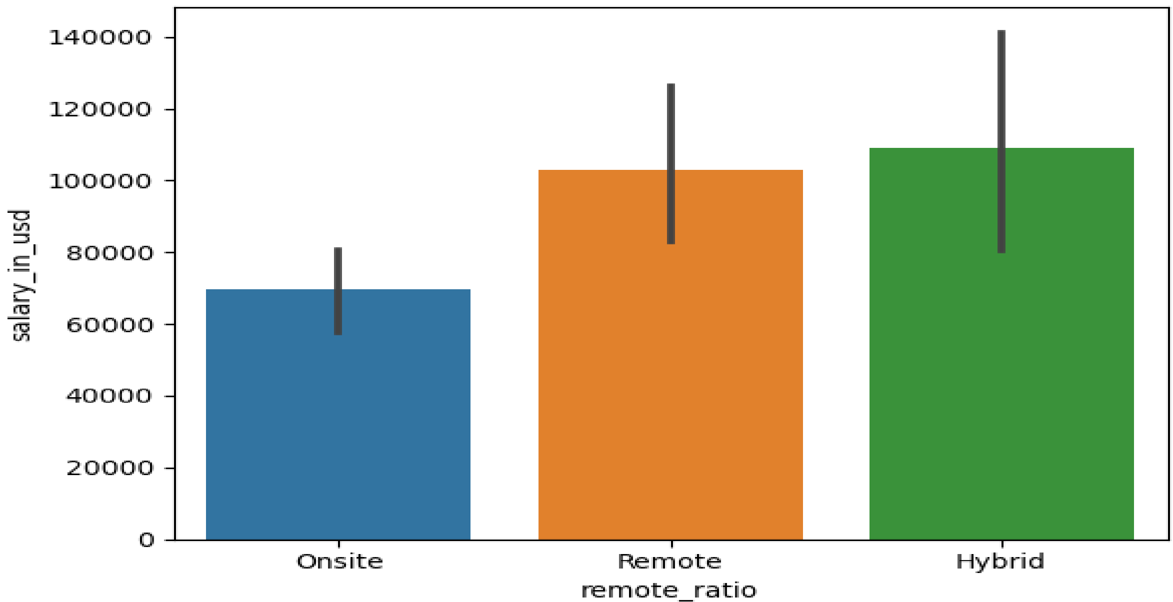


In the Dataset for Canada, Executive experience level receives the highest salary in Usd followed by Senior experience level. While Entry level receives the least salary.

 # Obtaining a plot between remote_ratio and salary_in_usd

Syntax: sns.barplot(data= data_canada, x='remote_ratio', y='salary_in_usd')

Output: <AxesSubplot:xlabel='remote_ratio', ylabel='salary_in_usd'>



From the above plot, in Canada, hybrid job type receives the highest salary followed by remote. Onsite job type receives the least.

2.1.3.3 Performing Exploratory Data Analysis (EDA) on the Covid-19 Datasets using Python Programming Language.

 # Install needed Python libraries

Syntax: pip install lxml html5lib beautifulsoup4

```
import requests
import numpy as np
import urllib.request
import pandas as pd
import csv
from bs4 import BeautifulSoup
import seaborn as sns
sns.set_style("darkgrid")
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore')
```

 # Read the time_series_covid19_confirmed_global file obtained from John Hopkins University

syntax: jh = pd.read_csv('time_series_covid19_confirmed_global.csv')

 # Obtain the dataset for Nigeria

syntax: jhnc= jh[jh['Country/Region']== 'Nigeria'].drop(['Province/State','Lat','Long','Country/Region'], axis=1)

 # Clean the dataset for Nigeria to obtain a better representation

Syntax: jhnc_1 =jhnc.T.reset_index().rename(columns ={'index':'Date',206:'Confirmed cases'})

 # Read the Dataset into a csv file for future use

Syntax: *jhnc_1.to_csv('JH_NG_confirmed.csv')*

 # Read the time_series_covid19_recovered_global file obtained from John Hopkins University

Syntax: *jhnr = pd.read_csv("time_series_covid19_recovered_global.csv")*

 # Obtain the Dataset for Nigeria and drop unnecessary columns

Syntax: *jhnr_1=jhnr[jhnr['Country/Region']=='Nigeria'].drop(['Province/State','Lat','Long',' Country/Region'], axis=1)*

 # Clean the Dataset for Nigeria to obtain a better representation

Syntax: jhnr_2 = jhnr_1.T.reset_index().rename(columns={}).rename(columns={'index':'Dat e',191: 'Recovered Cases'})

 # Read the Cleaned Dataset into a csv file for future referencing

Syntax: *jhnr_2.to_csv('JH_NG_Recovered.csv')*

 # Read the time_series_covid19_recovered_global file obtained from John Hopkins Uni versity

Syntax: *jhnd = pd.read_csv('time_series_covid19_deaths_global.csv')*

Loading Dataset into Jupyter notebook

 # Obtain the Dataset for Nigeria and drop unnecessary column(s)
Syntax: jhnd_1 = jhnd[jhnd['Country/Region']=='Nigeria'].drop(['Province/State','Lat','Lon g','Country/Region'], axis=1)

 # Clean the Dataset to obtain a better representation

Syntax: jhnd_2 = jhnd_1.T.reset_index().rename(columns= {'index':'Date',206:'Death Cases' })

 # Read the cleaned Dataset into a Csv file to obtain a better Representation

Syntax: jhnd_2.to_csv('JH_NG_death.csv')

 # Merge the Datasets to obtain a single Dataset for Analysis

Syntax: df_1 =(jhnc_1.merge(jhnr_2, how ='left', on = 'Date')).merge(jhnd_2, how = 'left', o n ='Date')

df_1.to_csv('Covid_data_Nigeria.csv')

 # Clean df_1 and convert each column to appropriate datatypes

Syntax: df_2 = df_1.drop(df_1.index[0:37], axis=0)

df_2['Date']= pd.to_datetime(df_2['Date'])

df_2

Output:

```
         Date  Confirmed cases  Recovered Cases  Death Cases
37   2020-02-28                1                0            0
38   2020-02-29                1                0            0
39   2020-03-01                1                0            0
40   2020-03-02                1                0            0
41   2020-03-03                1                0            0
...        ...              ...              ...          ...
1138 2023-03-05           266598                0         3155
1139 2023-03-06           266598                0         3155
1140 2023-03-07           266598                0         3155
1141 2023-03-08           266598                0         3155
1142 2023-03-09           266598                0         3155

[1106 rows x 4 columns]
```

#To Obtain the number of rows and columns in the dataset

Syntax: df_2.shape

Output: (1106, 4)

df_2 is a Dataset containing information on covid cases for Nigeria. it consists of 1106 rows a nd 4 columns.

#To Obtain general information on df_2

Syntax: df_2.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
Int64Index: 1106 entries, 37 to 1142
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Date             1106 non-null   datetime64[ns]
 1   Confirmed cases  1106 non-null   int64
 2   Recovered Cases  1106 non-null   int64
 3   Death Cases      1106 non-null   int64
dtypes: datetime64[ns](1), int64(3)
memory usage: 43.2 KB
```

df_2 contains four columns of which the values of three (3) are integers and one (1) consists of datetime values.

#To obtain statistical information on df_2

Syntax: df_2.describe()

Output:

```
      Confirmed cases  Recovered Cases  Death Cases
count      1106.000000      1106.000000  1106.000000
mean     170253.276673     35707.131103  2171.899638
std       93485.952252     56832.109704  1057.848600
min           1.000000         0.000000     0.000000
25%       67627.250000         0.000000  1173.750000
50%      193866.000000         0.000000  2491.500000
75%      256148.000000     58761.250000  3143.750000
max      266598.000000    165208.000000  3155.000000
```

In df_2 Dataset, the maximum, mean and minimum confirmed covid cases are 266598, 170253.27 and 1 respectively, while for recovered covid cases, the minimum, mean and maximum are 0,35707.13 and 165208 respectively. The average covid death cases is 2171.89

#To obtain names of column(s) contained in the Dataset df_2

Syntax: df_2.columns

Output: Index(['Date', 'Confirmed cases', 'Recovered Cases', 'Death Cases'], dtype='object')

 # What are the data types present in df_2?

Syntax: df_2.dtypes

```
Output: Date              datetime64[ns]
        Confirmed cases            int64
        Recovered Cases            int64
        Death Cases                int64
        dtype: object
```

 # Are there null values in the df_2 DataFrame

Syntax: df_2.isna().sum()

```
Output: Date              0
        Confirmed cases   0
        Recovered Cases   0
        Death Cases       0
        dtype: int64
```

In df_2 Dataset, there are no null values.

 # Read the covid_external file into a DataFrame

Syntax: e_data_1 = pd.read_csv('covid_external.csv')

 # What is the shape of the DataFrame ?

Syntax: e_data_1.shape

Output: (37, 12)

In the e_data_1 DataFrmae, there are a total number of 37 rows and 12 columns

 # obtain more information about the external data e_data_1

Syntax: e_data_1.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   states                  37 non-null     object
 1   region                  37 non-null     object
 2   Population              37 non-null     int64
 3   Overall CCVI Index      37 non-null     float64
 4   Age                     37 non-null     float64
 5   Epidemiological         37 non-null     float64
 6   Fragility               37 non-null     float64
 7   Health System           37 non-null     float64
 8   Population Density      37 non-null     float64
 9   Socio-Economic          37 non-null     float64
 10  Transport Availability  37 non-null     float64
 11  Acute IHR               37 non-null     float64
dtypes: float64(9), int64(1), object(2)
memory usage: 3.6+ KB
```

The e_data_1 contains twelve (12) columns of which the values of nine (9) are floats, the val ues of one (1) are integers and the values of two (2) are objects or strings

#To Obtain statistical information about e_data_1

Syntax: e_data_1.describe()

```
Output:    Population  Overall CCVI Index        Age  Epidemiological  \
count  3.700000e+01           37.000000  37.000000        37.000000
mean   5.843892e+06            0.502703   0.502703         0.500000
std    2.622344e+06            0.301373   0.301373         0.299073
min    2.606000e+06            0.000000   0.000000         0.000000
25%    4.272000e+06            0.300000   0.300000         0.300000
50%    5.185000e+06            0.500000   0.500000         0.500000
75%    6.376000e+06            0.800000   0.800000         0.700000
max    1.472600e+07            1.000000   1.000000         1.000000

       Fragility  Health System  Population Density  Socio-Economic  \
count  37.000000      37.000000                37.0       37.000000
```

```
mean    0.502703       0.502703               0.5        0.502703
std     0.301373       0.301373               0.3        0.301373
min     0.000000       0.000000               0.0        0.000000
25%     0.300000       0.300000               0.3        0.300000
50%     0.500000       0.500000               0.5        0.500000
75%     0.800000       0.800000               0.8        0.800000
max     1.000000       1.000000               1.0        1.000000

       Transport Availability  Acute IHR
count               37.000000  37.000000
mean                 0.502703   0.954054
std                  0.301373   0.100539
min                  0.000000   0.790000
25%                  0.300000   0.870000
50%                  0.500000   0.930000
75%                  0.800000   1.040000
max                  1.000000   1.140000
```

 # Obtain the column(s) present in the Dataset e_data_1

Syntax: e_data_1.columns

Output: Index(['states', 'region', 'Population', 'Overall CCVI Index', 'Age', 'Epidemiological', 'Fragility', 'Health System', 'Population Density', 'Socio-Economic', ' Transport Availability', 'Acute IHR'], dtype='object')

#Are there any null value(s) present in the e_data_1?

Syntax: e_data_1.isna().sum()

```
Output: states                  0
       region                  0
       Population              0
       Overall CCVI Index      0
       Age                     0
       Epidemiological         0
       Fragility               0
       Health System           0
       Population Density      0
       Socio-Economic          0
       Transport Availability  0
       Acute IHR               0
       dtype: int64
```

In e_data_1, there are no null or empty values

 # Obtain data types of column(s) present in e_data_1

Syntax: e_data_1.dtypes

```
Output: states                   object
        region                   object
        Population               int64
        Overall CCVI Index       float64
        Age                      float64
        Epidemiological          float64
        Fragility                float64
        Health System            float64
        Population Density       float64
        Socio-Economic           float64
        Transport Availability   float64
Acute IHR                        float64
dtype: object
```

 # Read the Budget data external data file into a DataFrame

Syntax: e_data_2 = pd.read_csv('Budget data.csv')

Output: e_data_2

	states	Initial_budget (Bn)	Revised_budget (Bn)
0	Abia	136.60	102.70
1	Adamawa	183.30	139.31
2	Akwa-Ibom	597.73	366.00
3	Anambra	137.10	112.80
4	Bauchi	167.20	128.00
5	Bayelsa	242.18	183.15
6	Benue	189.00	119.00
7	Borno	146.80	108.80
8	Cross River	1100.00	147.10
9	Delta	395.50	282.30
10	Ebonyi	178.40	131.80
11	Edo	179.20	128.80
12	Ekiti	124.50	91.10
13	Enugu	169.56	146.40
14	Gombe	130.83	107.40
15	Imo	197.60	108.30
16	Jigawa	152.92	124.00
17	Kaduna	259.25	223.60
18	Kano	200.00	138.00
19	Katsina	244.00	213.00
20	Kebbi	138.00	99.60
21	Kogi	176.00	102.00
22	Kwara	160.00	120.00

23	*Lagos*	*1680.00*	*920.50*
24	*Nasarawa*	*108.40*	*62.96*
25	*Niger*	*155.00*	*98.00*
26	*Ogun*	*449.90*	*280.00*
27	*Ondo*	*187.80*	*151.40*
28	*Osun*	*119.60*	*82.20*
29	*Oyo*	*213.00*	*174.00*
30	*Plateau*	*177.30*	*122.00*
31	*Rivers*	*530.80*	*300.40*
32	*Sokoto*	*202.40*	*153.00*
33	*Taraba*	*215.00*	*150.50*
34	*Yobe*	*108.00*	*86.00*
35	*Zamfara*	*188.50*	*127.30*
36	*FCT*	*278.78*	*199.00*

 # obtain general information about the e_data_2 DataFrame

Syntax: e_data_2.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   states               37 non-null     object
 1   Initial_budget (Bn)  37 non-null     float64
 2   Revised_budget (Bn)  37 non-null     float64
dtypes: float64(2), object(1)
memory usage: 1016.0+ bytes
```

In the e_data_2 Data Frame, there are a total of three (3) columns of which the values of two (2) columns are floats and the values of one (1) are objects

 # Obtain statistical information about e_data_2

Syntax: e_data_2.describe()

```
Output:    Initial_budget (Bn)  Revised_budget (Bn)
count           37.00000           37.000000
mean           276.22027          171.092432
std            299.37630          142.974439
min            108.00000           62.960000
25%            152.92000          108.300000
50%            183.30000          128.800000
75%            242.18000          174.000000
max           1680.00000          920.500000
```

In e_data_2, the mean or average initial and revised budgets are 276.22Bn and 171.09Bn resp ectively

 # Obtain the number of rows and columns present in e_data_2

Syntax: e_data_2.shape

Output: (37, 3)

e_data_2 contains 37 rows and 3 columns

 # Obtain the columns names present in e_data_2

Syntax: e_data_2.columns

Output: Index(['states', 'Initial_budget (Bn)', 'Revised_budget (Bn)'], dtype='object')

 # Are there null values present in e_data_2 ?

Syntax: e_data_2.isna().sum()

```
Output: states                0
        Initial_budget (Bn)   0
        Revised_budget (Bn)   0
        dtype: int64
```

There are no null values in the e_data_2 Data Frame

 # Obtain the data types present in e_data_2

Syntax: e_data_2.dtypes

```
Output: states                object
        Initial_budget (Bn)   float64
        Revised_budget (Bn)   float64
        dtype: object
```

 # Check for duplicate rows in e_data_2

Syntax: e_data_2.value_counts()

Output:

```
states     Initial_budget (Bn)  Revised_budget (Bn)
Abia       136.60               102.70               1
Kano       200.00               138.00               1
Kebbi      138.00               99.60                1
Kogi       176.00               102.00               1
Kwara      160.00               120.00               1
Lagos      1680.00              920.50               1
Nasarawa   108.40               62.96                1
Niger      155.00               98.00                1
```

```
Ogun        449.90          280.00          1
Ondo        187.80          151.40          1
Osun        119.60          82.20           1
Oyo         213.00          174.00          1
Plateau     177.30          122.00          1
Rivers      530.80          300.40          1
Sokoto      202.40          153.00          1
Taraba      215.00          150.50          1
Yobe        108.00          86.00           1
Katsina     244.00          213.00          1
Kaduna      259.25          223.60          1
Adamawa     183.30          139.31          1
Jigawa      152.92          124.00          1
Akwa-Ibom   597.73          366.00          1
Anambra     137.10          112.80          1
Bauchi      167.20          128.00          1
Bayelsa     242.18          183.15          1
Benue       189.00          119.00          1
Borno       146.80          108.80          1
Cross River  1100.00        147.10          1
Delta       395.50          282.30          1
Ebonyi      178.40          131.80          1
Edo         179.20          128.80          1
Ekiti       124.50          91.10           1
Enugu       169.56          146.40          1
FCT         278.78          199.00          1
Gombe       130.83          107.40          1
Imo         197.60          108.30          1
Zamfara     188.50          127.30          1
dtype: int64
```

There are no duplicated rows in the e_data_2 Data frame

#Read the covidnig external data into a Data Frame and clean the Data Frame by converting values of each column into their appropriate type

Syntax: e_data_3 = pd.read_csv('covidnig.csv')

e_data_3['No. of Cases (Lab Confirmed)']= e_data_3['No. of Cases (Lab Confirmed)'].str.replace(',','')

e_data_3['No. of Cases (Lab Confirmed)']= e_data_3['No. of Cases (Lab Confirmed)'].astype(int)

e_data_3['No. of Cases (on admission)']= e_data_3['No. of Cases (on admission)'].str.replace(',','')

e_data_3['No. of Cases (on admission)']= e_data_3['No. of Cases (on admission)'].astype(int

)

```
e_data_3['No. Discharged']= e_data_3['No. Discharged'].str.replace(',','')
e_data_3['No. Discharged']= e_data_3['No. Discharged'].astype(int)
```

 # Obtain general information about e_data_3

Syntax: e_data_3.info()

```
Output: <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 37 entries, 0 to 36
        Data columns (total 5 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   States Affected               37 non-null     object
 1   No. of Cases (Lab Confirmed)  37 non-null     int32
 2   No. of Cases (on admission)   37 non-null     int32
 3   No. Discharged                37 non-null     int32
 4   No. of Deaths                 37 non-null     int64
dtypes: int32(3), int64(1), object(1)
memory usage: 1.1+ KB
```

The e_data_3 Data Frame consist of five (5) columns, of which the values in three (3 column are integers and the values in one (1) column are objects and the values of one (1) column are objects.

 # What datatypes values are contained in e_data_3?

Syntax: e_data_3.dtypes

```
Output: States Affected                 object
        No. of Cases (Lab Confirmed)    int32
        No. of Cases (on admission)     int32
        No. Discharged                  int32
        No. of Deaths                   int64
        dtype: object
```

 # Obtain statistical information contained in e_data_3

Syntax: e_data_3.describe()

```
Output:    No. of Cases (Lab Confirmed)  No. of Cases (on admission)  \
count                  37.000000                    37.000000
```

mean 2119.837838 240.810811
std 4537.417740 595.255773
min 5.000000 0.000000
25% 381.000000 25.000000
50% 897.000000 57.000000
75% 1843.000000 183.000000
max 26708.000000 2840.000000

```
      No. Discharged  No. of Deaths
count      37.000000      37.000000
mean     1846.027027      33.000000
std      4009.464785      41.797794
min         3.000000       2.000000
25%       300.000000      11.000000
50%       775.000000      21.000000
75%      1737.000000      36.000000
max     24037.000000     236.000000
```

The minimum, mean and maximum number of covid death cases in DataFrame e_data_3 is 2, 33 and 236 respectively

 # how many row(s) and column(s) are contained in e_data_3

Syntax: e_data_3.shape

Output: (37, 5)

There are a total number of 37 rows and 5 columns in Data Frame e_data_3

 # Are there null values in e_data_3

Syntax: e_data_3.isna().sum()

```
Output: States Affected                0
        No. of Cases (Lab Confirmed)   0
        No. of Cases (on admission)    0
        No. Discharged                 0
        No. of Deaths                  0
        dtype: int64
```

There are no null or empty values in e_data_3

 # Obtain the column names contained in e_data_3

Syntax: e_data_3.columns

*Output: Index(['States Affected', 'No. of Cases (Lab Confirmed)',
'No. of Cases (on admission)', 'No. Discharged', 'No. of Deaths'], dtype='object')*

 # Read the external data set (RealGDP) into a DataFrame

Syntax: e_data_4 = pd.read_csv('RealGDP.csv')

Output: e_data_4

```
   Year        Q1          Q2          Q3          Q4
0  2014  15438679.50  16084622.31  17479127.58  18150356.45
1  2015  16050601.38  16463341.91  17976234.59  18533752.07
2  2016  15943714.54  16218542.41  17555441.69  18213537.29
3  2017  15797965.83  16334719.27  17760228.17  18598067.07
4  2018  16096654.19  16580508.07  18081342.10  19041437.59
5  2019  16434552.65  16931434.89  18494114.17  19530000.00
6  2020  16740000.00  15890000.00  17820000.00         0.00
```

 # Obtain general information about e_data_4

Syntax: e_data_4.info()

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Year    7 non-null      int64
 1   Q1      7 non-null      float64
 2   Q2      7 non-null      float64
 3   Q3      7 non-null      float64
 4   Q4      7 non-null      float64
dtypes: float64(4), int64(1)
memory usage: 408.0 bytes
```

In the e_data_4 DataFrame, there are a total of five (5) columns. Four (4) columns have float values and one (1) column has integer values.

 # Obtain Statistical information contained in e_data_4

Syntax: e_data_4.describe()

Output:

```
              Year            Q1            Q2            Q3            Q4
count     7.000000  7.000000e+00  7.000000e+00  7.000000e+00  7.000000e+00
mean   2017.000000  1.607174e+07  1.635760e+07  1.788093e+07  1.600959e+07
std       2.160247  4.225676e+05  3.423407e+05  3.442170e+05  7.075830e+06
min    2014.000000  1.543868e+07  1.589000e+07  1.747913e+07  0.000000e+00
25%    2015.500000  1.587084e+07  1.615158e+07  1.765783e+07  1.818195e+07
```

```
50%    2017.000000  1.605060e+07  1.633472e+07  1.782000e+07  1.853375e+07
75%    2018.500000  1.626560e+07  1.652192e+07  1.802879e+07  1.881975e+07
max    2020.000000  1.674000e+07  1.693143e+07  1.849411e+07  1.953000e+07
```

 # Are there null values in e_data_4 DataFrame?

Syntax: e_data_4.isna().sum()

```
Output: Year    0
        Q1      0
        Q2      0
        Q3      0
        Q4      0
        dtype: int64
```

There are no null values in dataset e_data_4

 # How many rows and columns are contained in the e_data_4 Data Frame?

Syntax: e_data_4.shape

Output: (7, 5)

In the e_data_4 Data Frame, there a total of seven (7) rows and five (5) columns.

 # Are there duplicated rows in e_data_4?

Syntax: e_data_4.value_counts()

Output:

```
Year  Q1           Q2           Q3           Q4
2014  15438679.50  16084622.31  17479127.58  18150356.45    1
2015  16050601.38  16463341.91  17976234.59  18533752.07    1
2016  15943714.54  16218542.41  17555441.69  18213537.29    1
2017  15797965.83  16334719.27  17760228.17  18598067.07    1
2018  16096654.19  16580508.07  18081342.10  19041437.59    1
2019  16434552.65  16931434.89  18494114.17  19530000.00    1
2020  16740000.00  15890000.00  17820000.00  0.00           1
dtype: int64
```

There is no duplicate row in e_data_4

 # What are the Top 10 states in terms of Confirmed Covid cases by Laboratory test?

Syntax:

```
plot_1 = pd.DataFrame(e_data_3.groupby(e_data_3['States Affected'])['No. of Cases (Lab Confirmed)'].sum())
```

plot_1 = plot_1.sort_values('No. of Cases (Lab Confirmed)' ,ascending= False).head(10).reset_index()

plot_1

	States Affected	No. of Cases (Lab Confirmed)
0	Lagos	26708
1	FCT	9627
2	Kaduna	4504
3	Plateau	4262
4	Oyo	3788
5	Rivers	3279
6	Edo	2768
7	Ogun	2382
8	Kano	2032
9	Delta	1843

plot_1 is a Data Frame that contains the top 10 relationship between States Affected by Covid in Nigeria and the number of covid cases confirmed by testing suspected victims of the virus.

Data Visualization

 # Generate a bar plot for plot_1

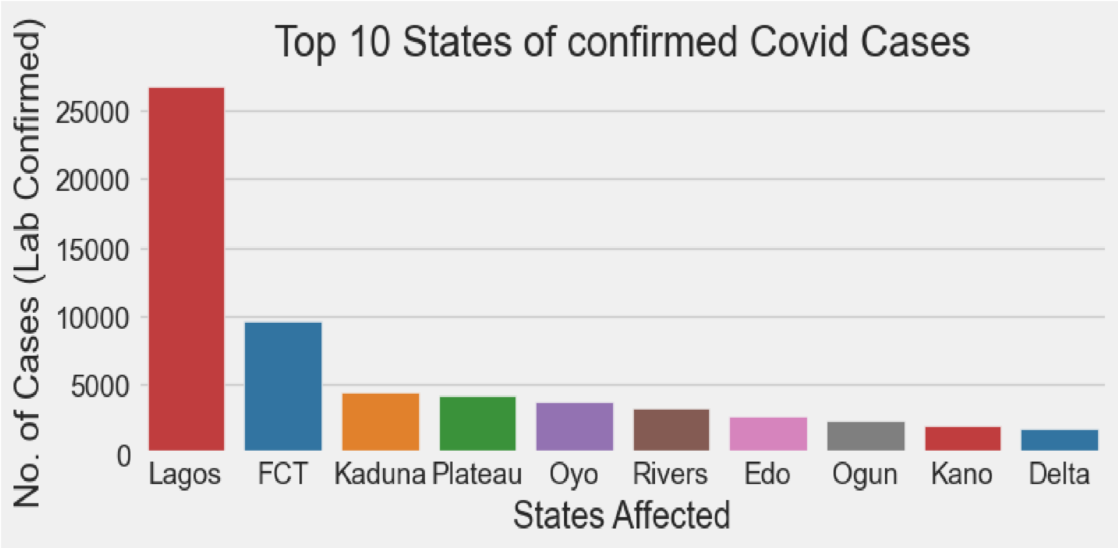
Syntax: fig, ax= plt.subplots(figsize=(8,3))

colors =['tab:red','tab:blue', 'tab:orange','tab:green','tab:purple','tab:brown','tab:pink','tab:gray']

sns.barplot(data=plot_1, x='States Affected', y='No. of Cases (Lab Confirmed)', ax=ax, palette=colors).set_title("Top 10 States of confirmed Covid Cases")

plt.show()

Output:



From the above bar plot, it is shown that Lagos state has the highest number of confirmed covid cases as confirmed in the laboratory. Lagos states tops the chart with more than 25,000 confirmed cases, Federal capital Teritory comes second with about 10,000 confirmed cases. Kaduna, Plateau, Oyo, Rivers, Edo, Ogun, Kano and Delta state all made it to the top 10 states with most confirmed covid cases.

 # What are the Top 10 states in terms of Continued Admitted Covid cases?

Synatax:
```
plot_20=pd.DataFrame(e_data_3.groupby(e_data_3['States Affected'])['No. of Cases (on admission)'].sum())
plot_20=plot_20.sort_values('No. of Cases (on admission)' ,ascending= False).head(10).reset_index()
plot_20
```

Output:

	States Affected	No. of Cases (on admission)
0	FCT	2840
1	Lagos	2435
2	Kaduna	579
3	Oyo	368
4	Plateau	280
5	Nasarawa	262
6	Rivers	232
7	Katsina	214
8	Kano	198
9	Gombe	183

Generate a bar plot for plot_20

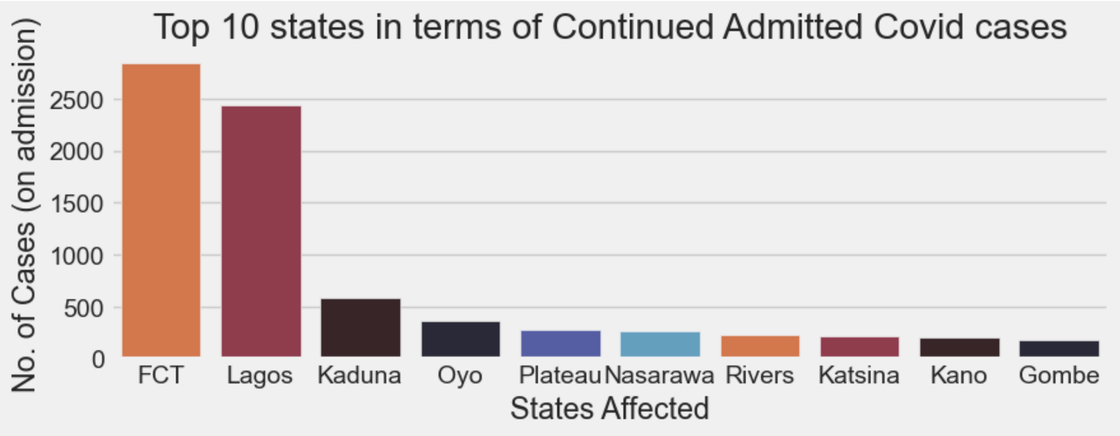
Syntax: fig, ax= plt.subplots(figsize=(9,3))

sns.barplot(data=plot_20, x='States Affected',y='No. of Cases (on admission)',

ax=ax, palette=sns.color_palette('icefire_r')).set_title("Top 10 states in terms of Continued Admitted Covid cases")

plt.show()

Output:



From the above, it is shown that the FCT tops the list of Continued admitted Covid cases with more than 2,500 cases still on admission. Lagos comes second with a little below 2,500, followed by Kaduna, Oyo, Plateau, Nasarawa, Rivers, Kaduna, Kano and Gombe all having the highest number of people not yet discharged.

What are the Top 10 states in terms of Discharged Covid cases?

Syntax: plot_2=pd.DataFrame(e_data_3.groupby(e_data_3['States Affected'])['No. Discharged'].sum())

plot_2=plot_2.sort_values('No. Discharged' ,ascending= False).head(10).reset_index()

plot_2

Output:

	States Affected	No. Discharged
0	Lagos	24037
1	FCT	6694
2	Plateau	3948
3	Kaduna	3877
4	Oyo	3374
5	Rivers	2987
6	Edo	2603

7	*Ogun*	*2175*
8	*Kano*	*1778*
9	*Delta*	*1737*

plot_2 is a Data Frame that contains the top 10 relationship between States Affected by Covid in Nigeria and the number of Discharged covid cases.

 # Generate a bar plot for plot_1

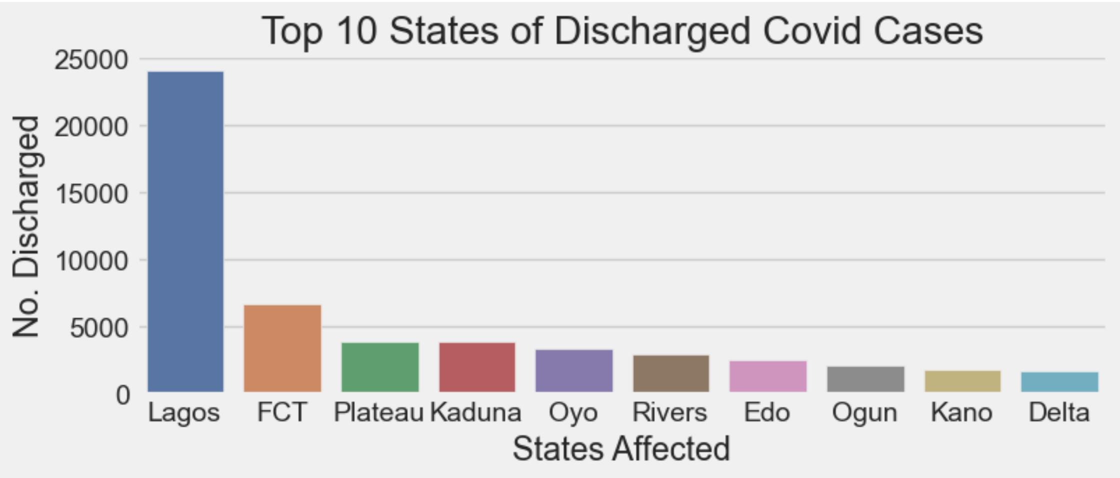
Syntax: fig, ax= plt.subplots(figsize=(8,3))

colors =['tab:cyan','tab:magenta', 'tab:yellow','tab:blue','tab:indigo']

sns.barplot(data=plot_2, x='States Affected', y='No. Discharged', ax=ax, palette=sns.color_palette('deep')).set_title("Top 10 States of Discharged Covid Cases")

plt.show()

Output:



From the above bar plot, it is shown that Lagos state tops the list of most Discharged Covid cases in Nigeria with about 25,0000 Discharged cases. The FCT comes second with more than 5,000 Discharged Covid Cases. Plateau, Kaduna, Oyo, Rivers, Edo, Ogun, Kano, and Delta all made it to the list.

 # Which states has the top 10 Death from Covid?

Syntax: plot_3 =pd.DataFrame(e_data_3.groupby(e_data_3['States Affected'])['No. of Deaths'].sum())

plot_3=plot_3.sort_values('No. of Deaths' ,ascending= False).head(10).reset_index()

plot_3

Output:

	States Affected	No. of Deaths
0	Lagos	236
1	Edo	113
2	FCT	93
3	Rivers	60
4	Kano	56
5	Delta	49
6	Kaduna	48
7	Oyo	46
8	Ondo	41
9	Borno	36

plot_3 is a Data Frame containing States in Nigeria and the total number of deaths from Covi d.

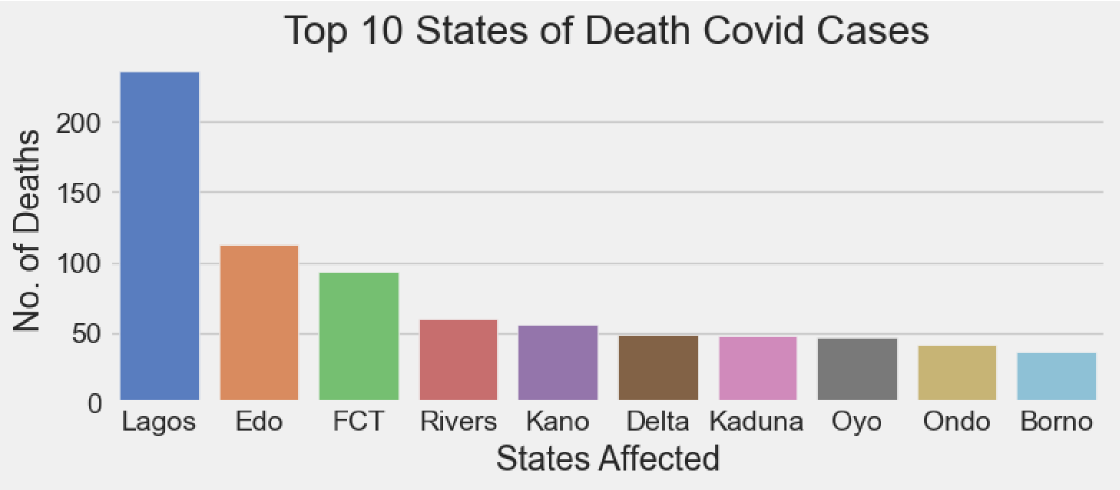
 # Generate a bar plot for plot_3

Syntax: fig, ax= plt.subplots(figsize=(8,3))

sns.barplot(data=plot_3, x='States Affected', y='No. of Deaths', ax=ax, palette=sns.color_palette('muted')).set_title("Top 10 States of Death Covid Cases")

plt.show()

Output:



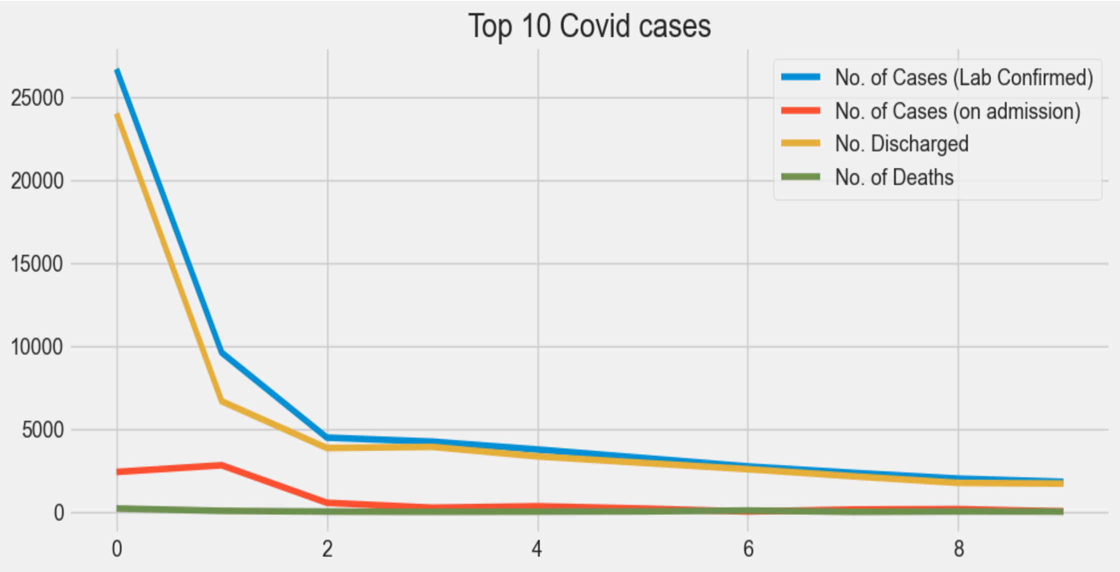
From the above plot, it is shown that Lagos state has the most death Cases from covid in Nigeria with more than 200 death cases recorded. Edo state comes second in the list with more than 100 covid death cases recorded. FCT, Rivers, Kano, Delta, Kaduna, Oyo, Ondo, and Borno follows accordingly in the top 10 recorded Covid Death Cases.

 # what is the relationship between the top 10 no. of cases (lab confirmed), no. of cases (on admission, no. discharged, and No. of Deaths?

Syntax: e_data_3.head(10).sort_values(['No. of Cases (Lab Confirmed)','No. of Cases (on admission)', 'No. Discharged','No. of Deaths'],ascending = False).plot(figsize=(12,5)).set_title('Top 10 Covid cases')

plt.show()

Output:



From the above line chart, it is shown that there is a positive proportionality between the no. of cases (lab confirmed), cases on admission. discharged cases and deaths amongst the top 10 states. This means that the higher the number of cases recorded amongst states leads to a corresponding higher number in cases on admission, Discharged and deaths, but, this is less seen in the number of recorded death covid cases.

 # What Correlation exist amongst the various data obtained?

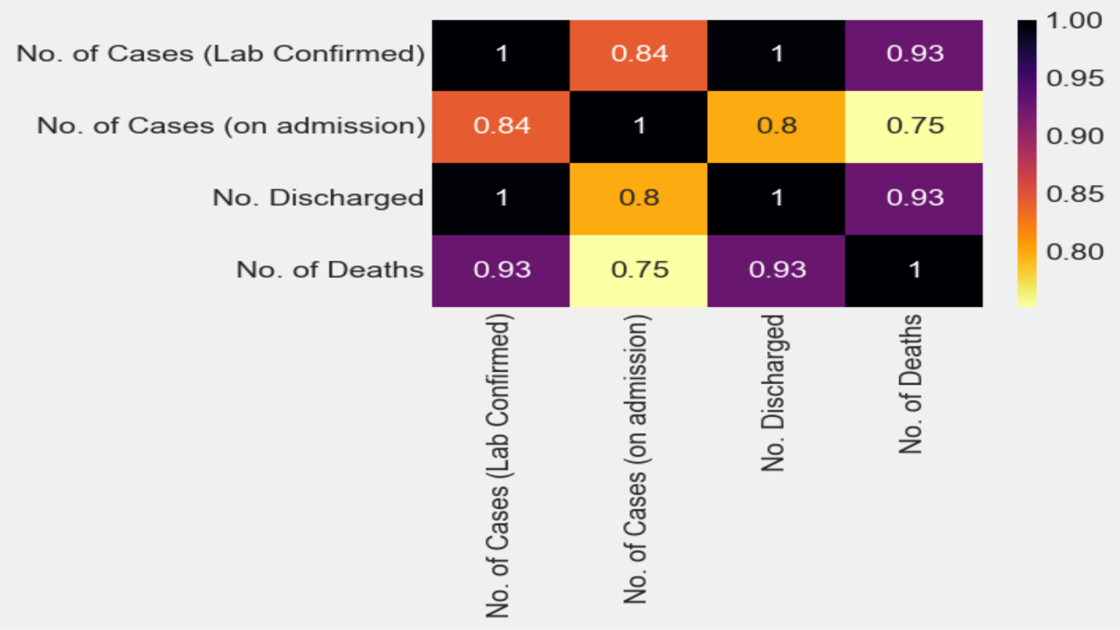
Syntax: plot_7 =e_data_3.corr()

fig, ax= plt.subplots(figsize=(5,3))

sns.heatmap(data= plot_7, annot=True, cmap='inferno_r',ax=ax)

plt.show()

Output:



From the heatmap of plot_7, is shown that there exist a very positive correlation amongst the variables contained in the Data Frame, with the strongest correlations shown or represented in darker colors. This means that the variables are strongly dependent on one another.

 # what is the total daily confirmed covid cases in Nigeria?

Syntax: plot_4=pd.DataFrame(df_2.groupby(pd.Grouper(key = 'Date', freq='1D'))

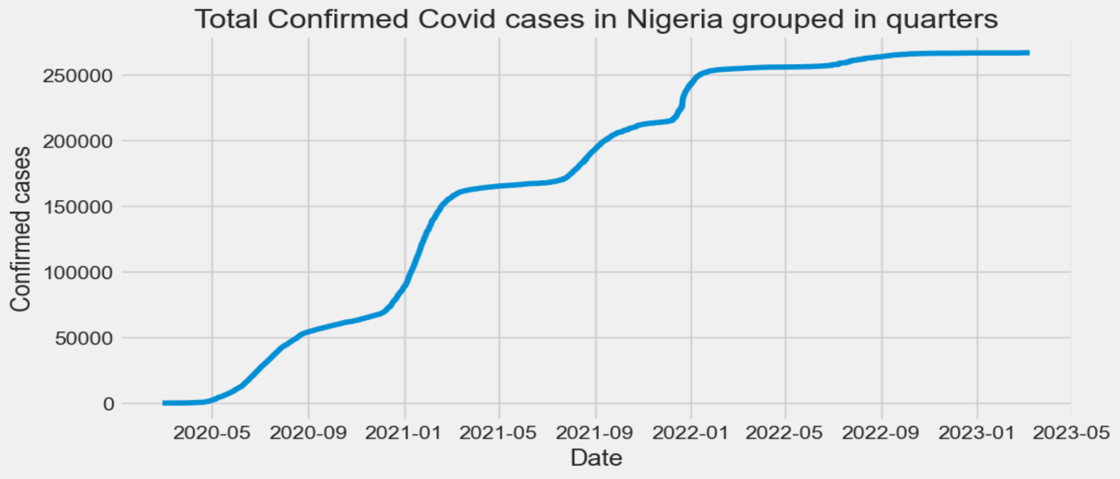
['Confirmed cases','Recovered Cases','Death Cases'].sum()).reset_index(fig, ax= plt.subplots (figsize=(10,5))

sns.lineplot(data=plot_4, x='Date',y ='Confirmed cases', ax=ax)

plt.title("Total Confirmed Covid cases in Nigeria grouped in quarters")

plt.show()

Output:



From the line plot of Data Frame plot_4, is shown that there was a reported increase in the number of confirmed covid cases from the 5th month of 2020 across Nigeria. This rose to above 50,000 in the 9th month of 2020 and exponentially rose until the 9th month of 2022 where it flattened a little higher than 250,000.

 # what is the total daily Recovered covid cases in Nigeria?

Syntax: plot_5=plot_4.copy()

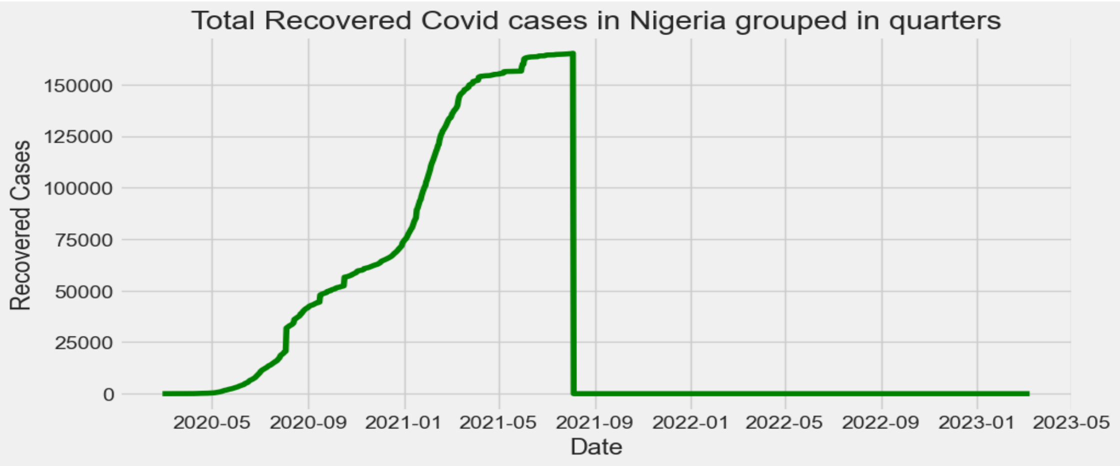
fig, ax= plt.subplots(figsize=(10,5))

sns.lineplot(data=plot_5, x='Date',y ='Recovered Cases', ax=ax, color=('green'))

plt.title("Total Recovered Covid cases in Nigeria grouped in quarters")

plt.show()

Output:



From the plot of Variable plot_5, it is shown that an increase in the recorded recovered covid cases was recorded after the 5th month of 2020 and rose above 25,000 before the 9th month of 2020. The number continued to increase until it got to about 175,000 where it climaxed and no increase further increase was recorded. All this happened before the 9th month of 2021.

 # what is the total daily Death covid cases in Nigeria?

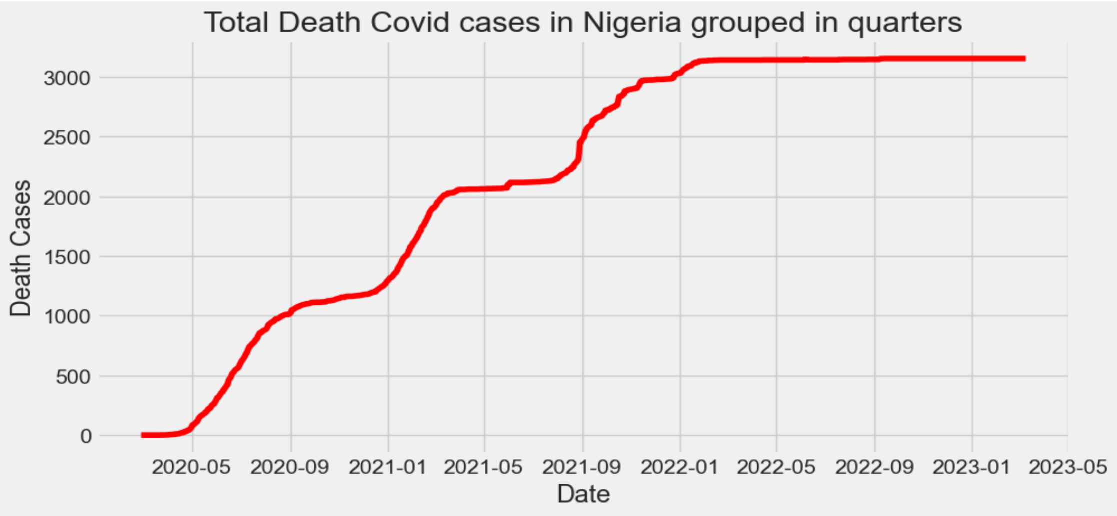
Syntax: plot_6=plot_4.copy()

fig, ax= plt.subplots(figsize=(10,5))

sns.lineplot(data=plot_6, x='Date',y ='Death Cases', ax=ax, color=('red'))

plt.title("Total Death Covid cases in Nigeria grouped in quarters") *plt.show()*

Output:



From the line plot obtained from plot_6, it is shown that an increase in the number of deaths from Covid started after the 5th month (may) of 2020 and this increase continued and about 1,000 total death cases was recorded on the 9th month of 2020. covid total death cases continued to increase across the country until it climaxed to a total of about 4,000 deaths on the first month of 2022 where no further death cases were recorded across the country.

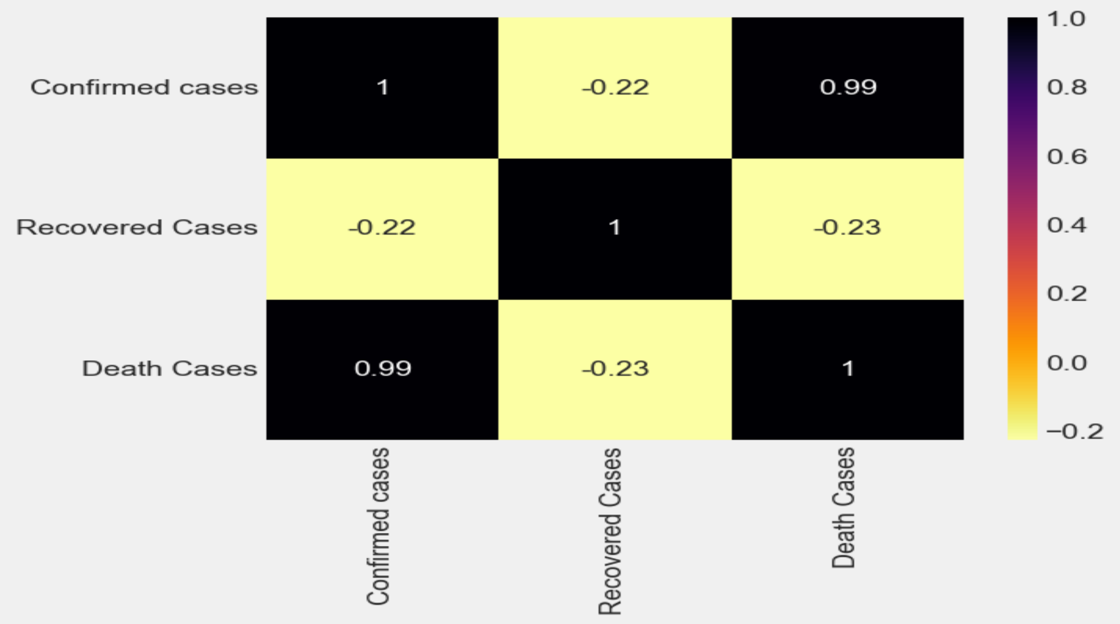
 # What correlation exist between the variables in plot_4?

Syntax: plot_19 = plot_4.corr()

sns.heatmap(data= plot_19, annot=True, cmap= 'inferno_r')

plt.show()

Output:

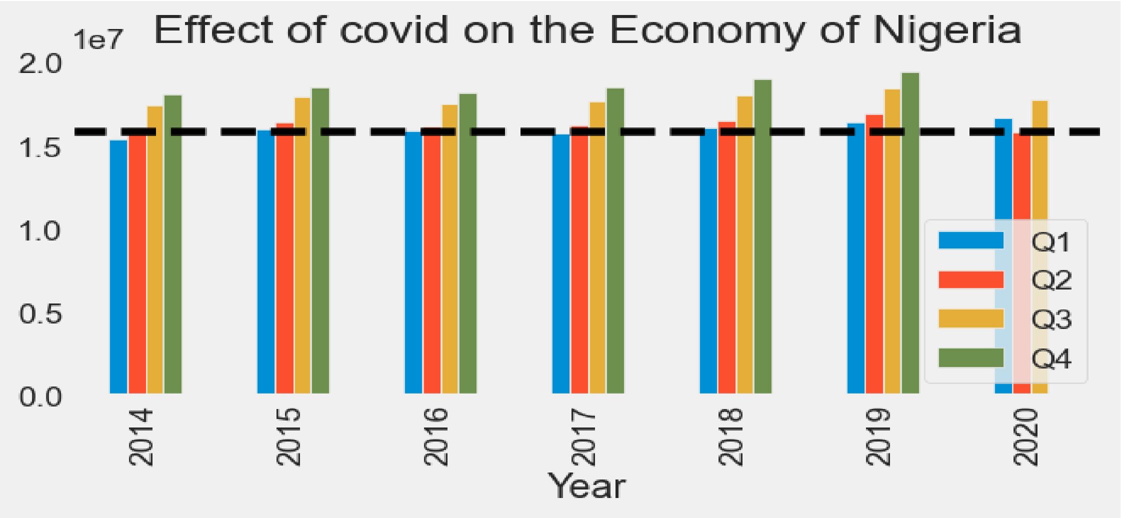


From the heatmap of plot_19, it is shown that there exist a mixed correlation among the variables in the dataset. A negative correlation exists between Confirmed Covid cases and Recovered cases and a positive correlation exist with death cases. Also, a negative correlation exist between Recovered covid cases and both confirmed and death cases. A positive correlation exist between Death cases and confirmed cases while its relationship with Recovered cases is a negative one.

 # What is the effect of the Pandemic on the economy of Nigeria ?

Syntax:
```
plot_8=e_data_4.melt(id_vars=(['Q1','Q2','Q3','Q4']),value_vars=('Year'),
value_name='Year')
plot_8.plot(x='Year',figsize=(7,3), kind='bar', grid=False
        ).set_title("Effect of covid on the Economy of Nigeria")
plt.axhline(y=15890000.00,ls='--', color ='black')
plt.show()
```

Output:



From the above bar chart, it is shown that the Economy of Nigeria increased simultaneously across each quarter prior to Covid, this trend continued in the first quarter of 2020 and there was a reverse in the uptrend at the coming of Covid as the economy saw a downtrend in the second quarter of 2020 to a level similar to the second quarter of 2014. Hence, it is shown that Covid had a negative impact on the economy.

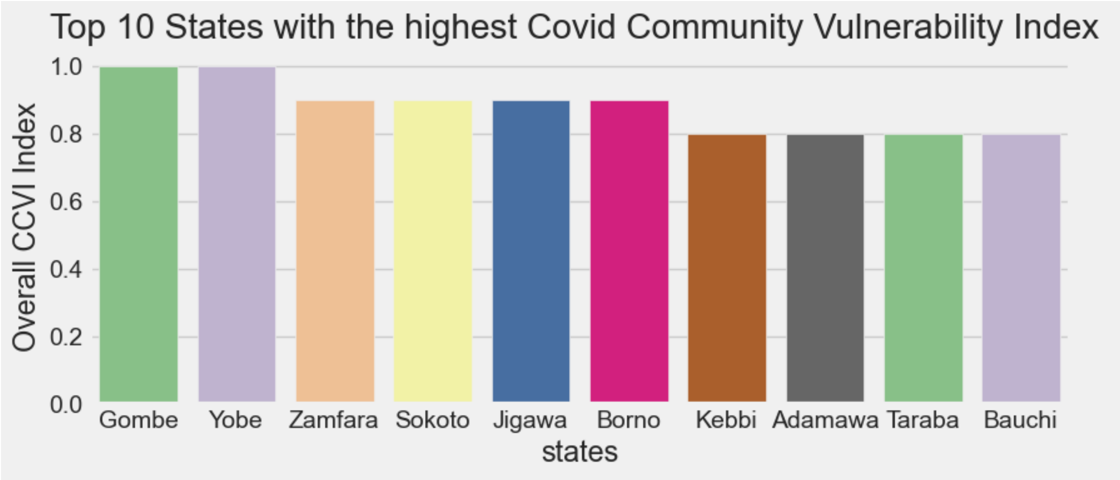
 # What top 10 states has the highest CCVI in Nigeria?

CCVI stands for Covid Community Vulnerability Index. The vulnerability index was computed by considering several factors such as socio-economic status, population density, housing type, transportation, epidemiological, health system etc, these factors are known as themes. Each theme was broken into subthemes, and data was gathered from them to compute the overall vulnerability index score by weighing equally each theme.

The term "vulnerability" refers to the impact of the virus on a community after the virus arrives. It ranks from Very Low (0) to Very High (1+)

```
plot_9=e_data_1.sort_values('Overall CCVI Index',ascending=False).reset_index().drop('in
dex', axis=1).head(10)
fig, ax= plt.subplots(figsize=(9,3.5))
sns.barplot(data=plot_9, x='states', y='Overall CCVI Index', ax=ax,
        palette=sns.color_palette('Accent')).set_title('Top 10 States with the highest Covid Co
mmunity Vulnerability Index ')
plt.show()
```

Output:

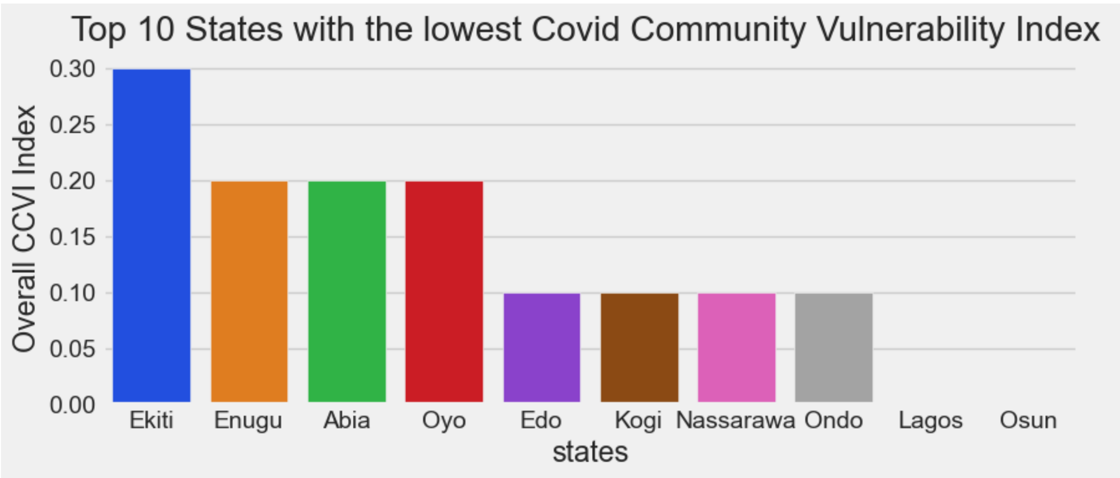


From the above Bar plot, it is shown that the state with the highest covid community vulnerability index are; Gombe and Yobe both with a CCVI of 1.0. Zamfara, Sokoto, Jigawa and Borno are joint second in the list with a CCVI value of about 0.9. Kebbi, Adamawa, Taraba and Bauchi came joint 3rd with a CCVI value 0.8.

 # What states has the least CCVI in Nigeria?

```
plot_10 = e_data_1.sort_values('Overall CCVI Index',ascending=False).reset_index().drop('
index', axis=1).tail(10)

fig, ax= plt.subplots(figsize=(9,3.5))
sns.barplot(data=plot_10, x='states', y='Overall CCVI Index', ax=ax,
        palette=sns.color_palette('bright')).set_title('Top 10 States with the lowest Covid Com
munity Vulnerability Index ')
plt.show()
```

Output:



From the bar plot of plot_10, it is shown that Lagos and Osun state has the least CCVI value of 0, Followed by Ondo, Nassarawa, Kogi and Edo with overall CCVI value of 0.10. Oyo, Abia, Enugu has an overall CCVI value of 0.20 and Ekiti with 0.30 Overall CCVI.

 # What is the correlation between the columns in plot_14?

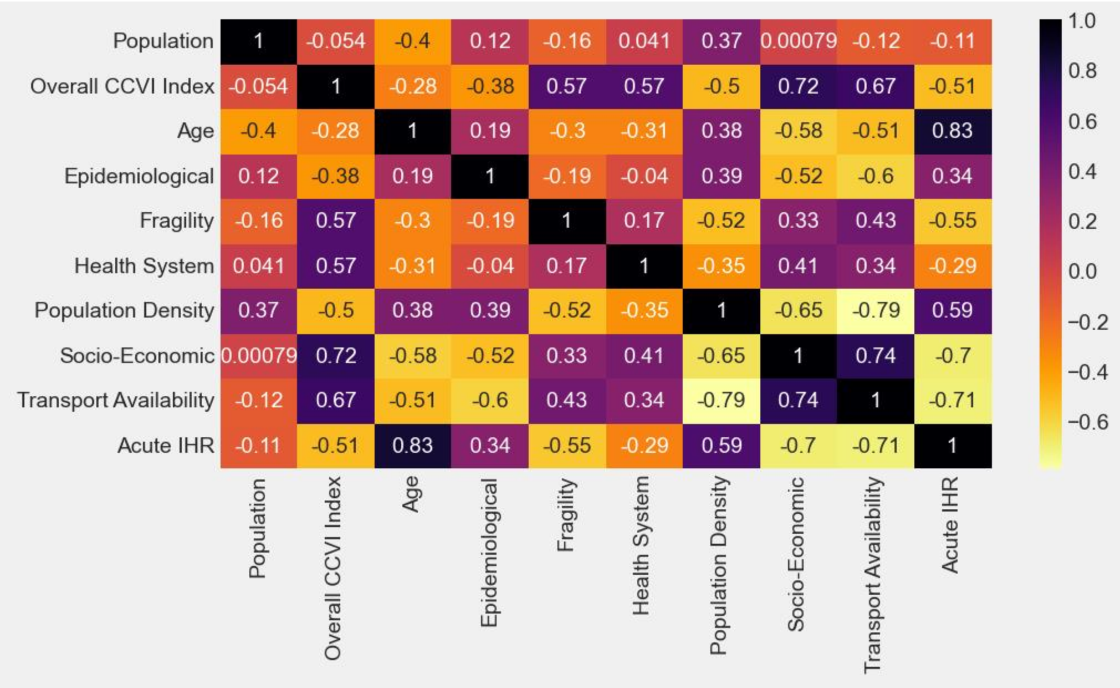
Syntax: plot_14= e_data_1.corr()

fig, ax=plt.subplots(figsize=(10,5))

sns.heatmap(data= plot_14, annot=True, cmap="inferno_r", ax=ax)

plt.show()

Output:



From the Heatmap plot of plot_14, it is seen that a mixed correlation exist between entries of the dataset.

 # what is the effect of covid on the budget of Nigeria?

Syntax: plot_15 = e_data_2.sort_values('Initial_budget (Bn)', ascending=False).reset_index( ).drop('index', axis=1)

plot_15

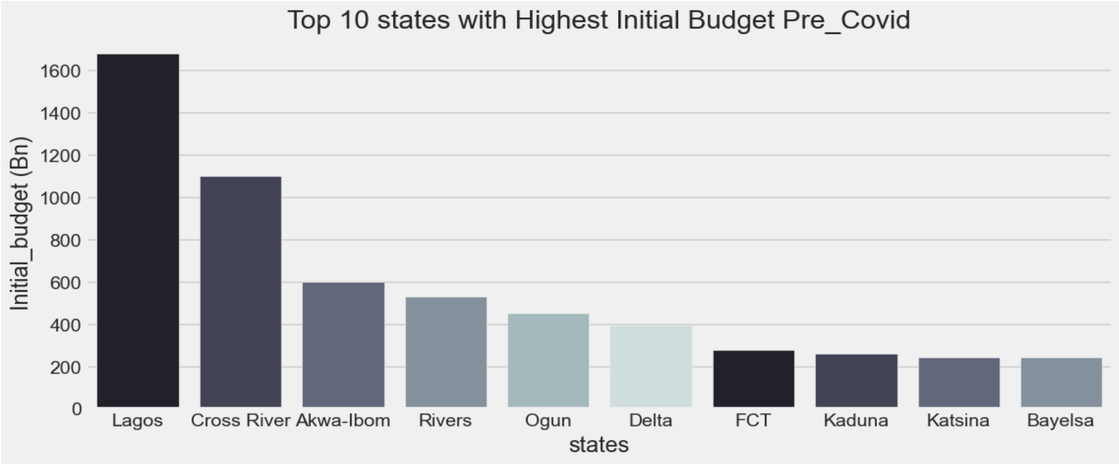
Output:

	states	Initial_budget (Bn)	Revised_budget (Bn)
0	Lagos	1680.00	920.50
1	Cross River	1100.00	147.10
2	Akwa-Ibom	597.73	366.00
3	Rivers	530.80	300.40
4	Ogun	449.90	280.00
5	Delta	395.50	282.30
6	FCT	278.78	199.00
7	Kaduna	259.25	223.60
8	Katsina	244.00	213.00
9	Bayelsa	242.18	183.15
10	Taraba	215.00	150.50
11	Oyo	213.00	174.00
12	Sokoto	202.40	153.00
13	Kano	200.00	138.00
14	Imo	197.60	108.30
15	Benue	189.00	119.00
16	Zamfara	188.50	127.30
17	Ondo	187.80	151.40
18	Adamawa	183.30	139.31
19	Edo	179.20	128.80
20	Ebonyi	178.40	131.80
21	Plateau	177.30	122.00
22	Kogi	176.00	102.00
23	Enugu	169.56	146.40
24	Bauchi	167.20	128.00
25	Kwara	160.00	120.00
26	Niger	155.00	98.00
27	Jigawa	152.92	124.00
28	Borno	146.80	108.80
29	Kebbi	138.00	99.60
30	Anambra	137.10	112.80
31	Abia	136.60	102.70
32	Gombe	130.83	107.40
33	Ekiti	124.50	91.10
34	Osun	119.60	82.20
35	Nasarawa	108.40	62.96
36	Yobe	108.00	86.00

 # which states has the top 10 highest initial budget?

```
Syntax: fig, ax= plt.subplots(figsize=(12,5))
sns.barplot(data=plot_15.head(10),x='states', y='Initial_budget (Bn)' ,ax=ax,
        palette=sns.color_palette('bone')).set_title('Top 10 states with Highest Initial Budget Pre_Covid')
plt.show()
```

Output:



From the above bar plot, it is shown that Lagos had the highest initial budget before covid with more than 1,600 billion initial budget. Cross river come second with an initial budget above 1,000 billion. followed by Akwa-ibom, Rivers, Ogun, Delta, FCT, Kaduna, Katsina and Bayelsa respectively.

 # Which states has the lowest initial budget prior to covid?

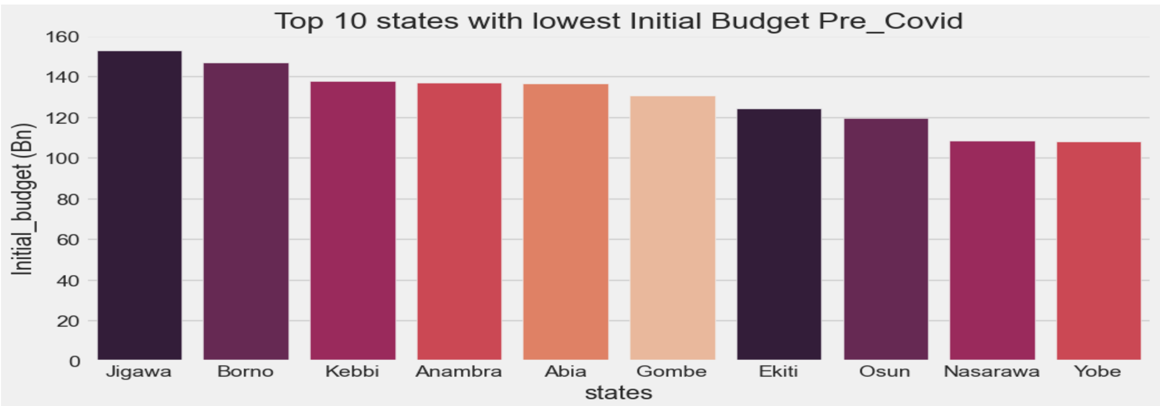
Syntax: fig, ax= plt.subplots(figsize=(11,5))

sns.barplot(data=plot_15.tail(10),x='states', y='Initial_budget (Bn)' ,ax=ax,

palette=sns.color_palette('rocket')).set_title('Top 10 states with lowest Initial Budget Pre_Covid')

plt.show()

Output:



From the above bar plot, it is shown that Yobe has the least initial budget a little above 100 billion, followed by Nasarawa, Osun, Ekiti, Gombe, Abia, Anambra, Kebbi, Borno and Jigawa respectively.

 # which states has the highest budget after revision due to the effect of covid?

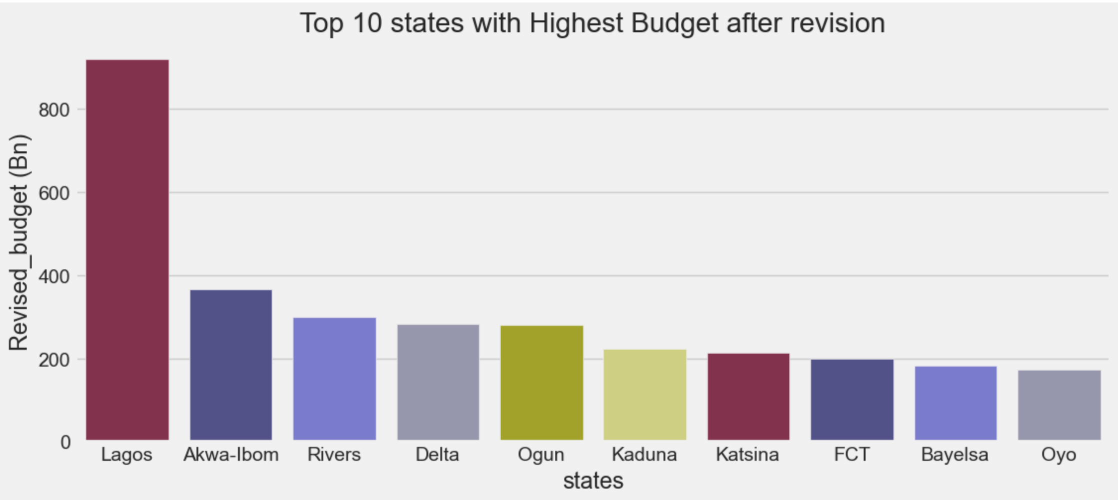
Syntax: plot_16 = e_data_2.sort_values('Revised_budget (Bn)',

ascending=False).reset_index().drop('index', axis=1)

plot_16

```
fig, ax= plt.subplots(figsize=(12,5))
sns.barplot(data=plot_16.head(10),x='states', y='Revised_budget (Bn)' ,ax=ax,
        palette=sns.color_palette('gist_stern')).set_title('Top 10 states with Highest Budget after revision')
plt.show()
```

Output:



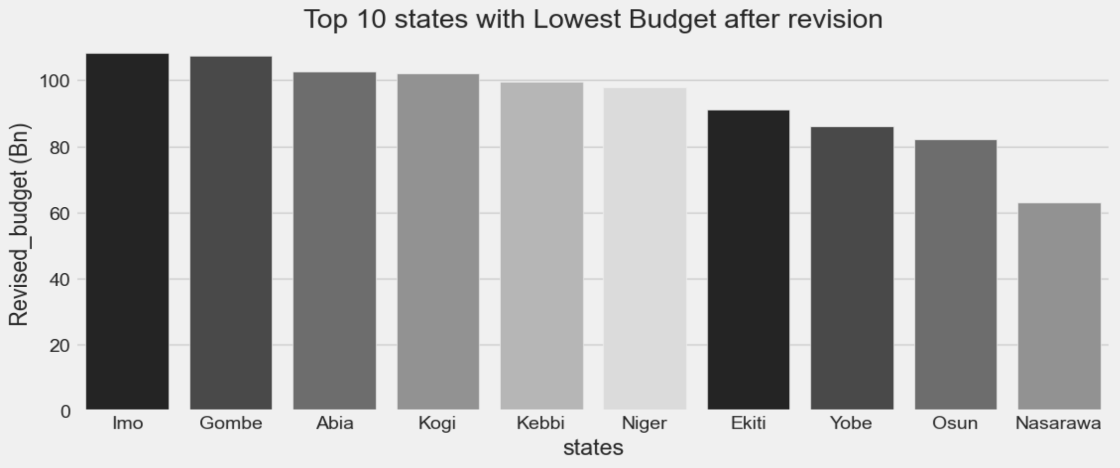
From the above bar plot, it is shown that Lagos has the highest budget after the initial budget was revised. Lagos tops with about 900 billion, followed by Akwa-ibom, Rivers, Delta, Ogun, Kaduna, Katsina, FCT, Bayelsa and Oyo respectively.

 # which states has the top 10 least budget after revision due to the impact of covid?

Syntax:

```
fig, ax= plt.subplots(figsize=(12,5))
sns.barplot(data=plot_16.tail(10),x='states', y='Revised_budget (Bn)' ,ax=ax,
        palette=sns.color_palette('gist_yarg_r')).set_title('Top 10 states with Lowest Budget a fter revision')
plt.show()
```

Output:

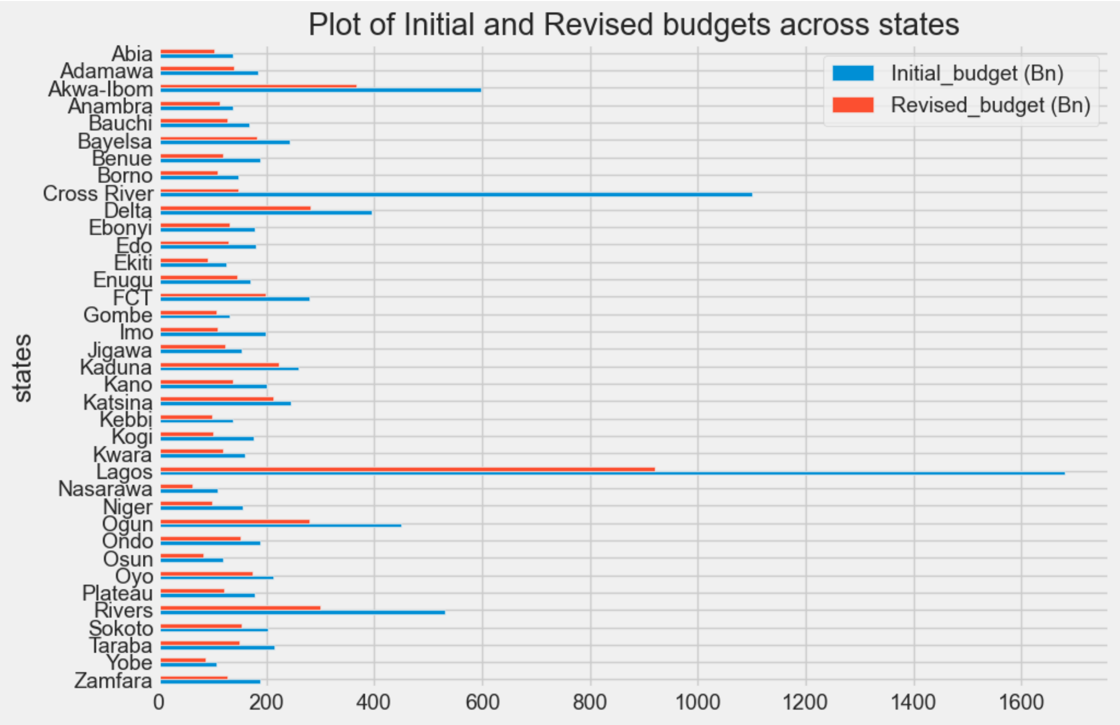


From the above bar chat, it is shown that Nasarawa has the least budget of a little above 60 billion after revision. They are followed by Osun, Yobe, Ekiti, Niger, Kebbi, Kogi, Abia, Gombe and Imo respectively.

 # Plot a horizontal bar chart showing initial and revised budget

Syntax:
```
plot_17= e_data_2.copy().sort_values('states', ascending=False)
fig, ax= plt.subplots(figsize=(10,7))

plot_17.plot(x='states',ax=ax,kind='barh'
            ).set_title('Plot of Initial and Revised budgets across states')
plt.show()
```

Output:



From the above plot, it is shown that there is a noticeable reduction in the budget of Nigeria across states as a result of the effect of covid.

 # what correlation exist between entries of budget dataset?

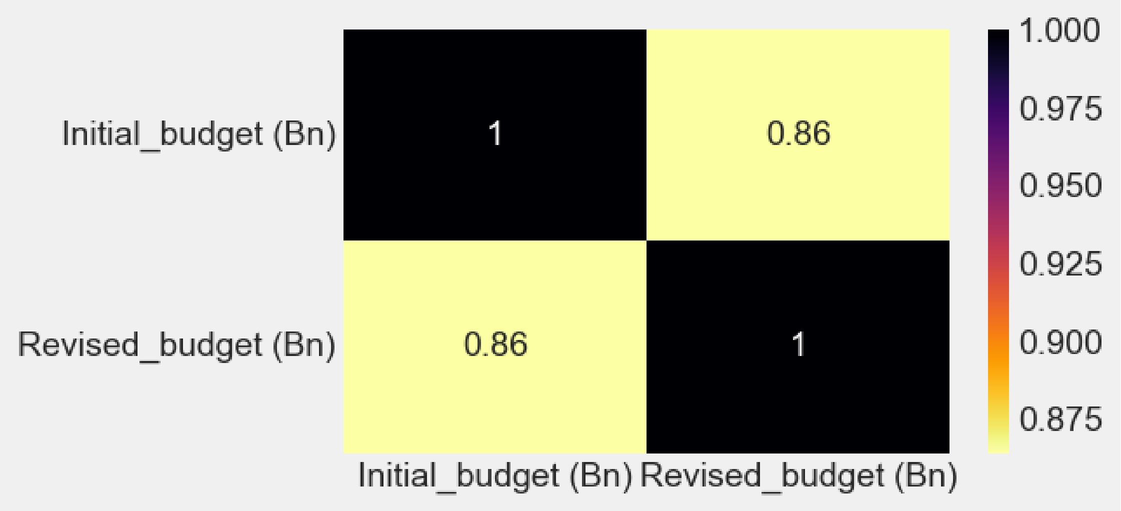
Syntax: plot_20 = e_data_2.corr()

fig, ax= plt.subplots(figsize=(5,3))

sns.heatmap(plot_20, annot=True, cmap="inferno_r",ax=ax)

plt.show()

Output:



From the heatmap plot of the budget dataset, it is shown that all entries has a positive correlation with one another.

 # What is the relationship or correlation across all the external datasets ?

Syntax: a= e_data_1.sort_values('states').reset_index().drop('index', axis=1)

b= e_data_3.sort_values('States Affected').reset_index().drop('index', axis=1)

b.rename(columns={'States Affected':'states'}, inplace=True)

df_3 =(a.merge(e_data_2, how='left', on='states')).merge(b, how='left', on='states')

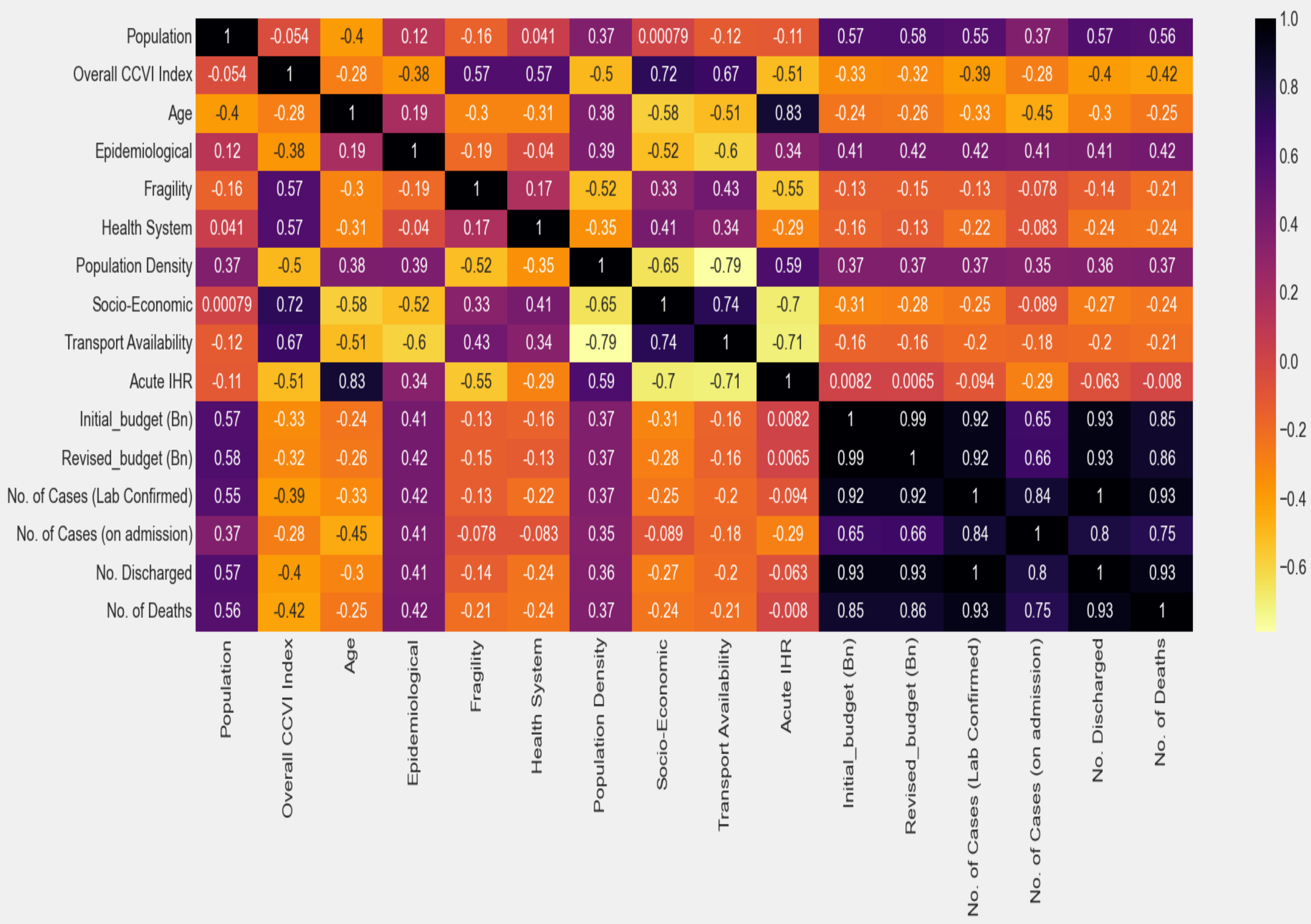
plot_18=df_3.corr()

fig, ax= plt.subplots(figsize=(20,7))

sns.heatmap(plot_18, annot=True, cmap="inferno_r",ax=ax)

plt.show()

Output:



From the heatmap plot of entries in some external dataset, it is show that there exist mixed correlations amongst entries of the dataset.

RECOMMENDATION BASED ON RESULTS OBTAINED

From the above study, it is shown that Covid-19 had a very negative impact on the country at large. In recommendation, to be prepared for any future pandemic, governments at various levels should improve on the fragility of communities (i.e weak governance, limited administrative capacity, chronic humanitarian crises, and persistent social tensions), health care systems across the country, transportation systems, and the social and economic aspects of the country to drastically reduce the impact of the virus on a community after the virus arrives.

2.1.4 MY STRUCTURED QUERY LANGUAGE (SQL)

MySQL is an open-source *Relational Database Management System* (RDBMS) that enables users to store, manage, and retrieve structured data efficiently. It is widely used for various applications, from small-scale projects to large-scale websites and enterprise-level solutions.

OPERATIONS CARRIED OUT ON DATASETS USING MYSQL WORKBENCH

1. Creation of a database
2. Carrying out basic numeric operations such as;
    - Addition
    - Subtraction
    - Multiplication
    - Division
    - Modulus
    - Basic algebra
    - Rounding up of values
    - Bitwise operators ($, |, ^)
    - Comparison operators (= < > <= >= !< !> <> !=)
3. Creation of tables with its columns
4. Working with basic SQL Commands such as;
    - CREATE
    - DROP
    - ALTER
    - TRUNCATE
    - RENAME
    - SELECT
    - INSERT INTO
    - UPDATE
    - DELETE
    - REPLACE
5. Working with basic SQL Logical expressions such as:
    - AND
    - OR

- NOT
- ANY
- SOME
- ALL
- BETWEEN
- IN
- AS
- LIKE
- IS NULL
- UNIQUE

6. Working with basic SQL keywords such as:
    - WHERE
    - DISTINCT
    - ORDER BY
    - DESC
    - ASC
    - SET
    - FROM
    - GROUP BY
    - HAVING
7. Working with basic SQL constraints such as:
    - NOT NULL
    - UNIQUE
    - PRIMARY KEY
    - FOREIGN KEY
8. Working with basic SQL aggregation functions such as:
    - AVG
    - COUNT
    - MAX
    - MIN
    - SUM
9. Working with basic SQL joins such as;

- INNER JOIN
- LEFR JOIN
- RIGHT JOIN

These operations were carried out on a dummy dataset using SQL

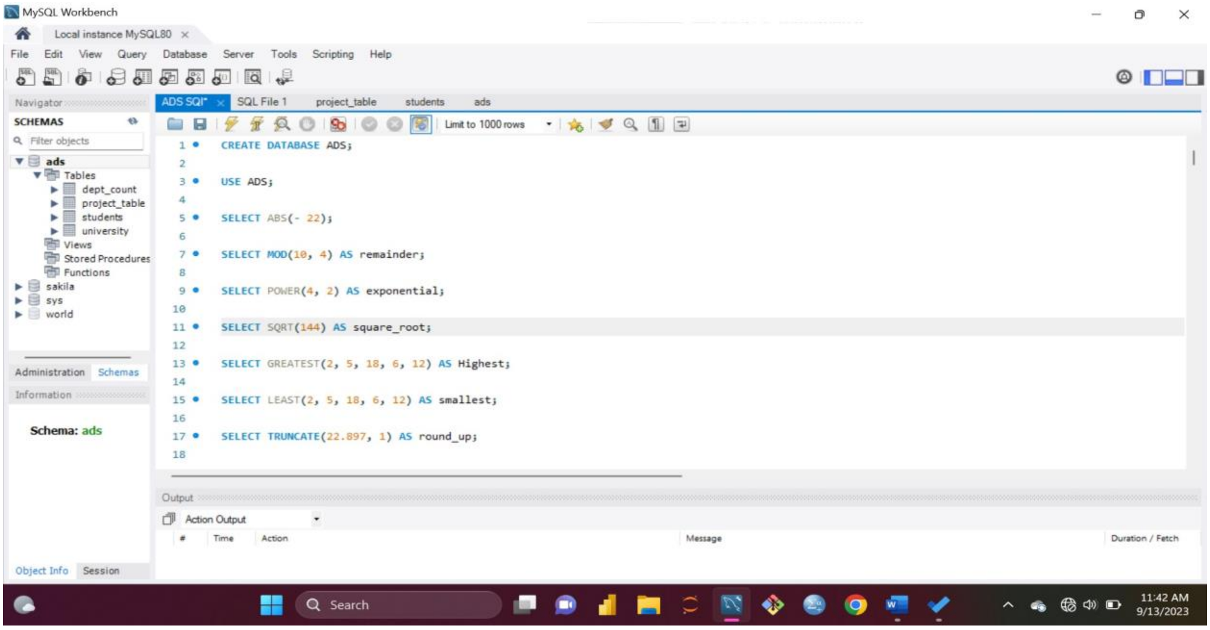


FIG 4. A typical MySQL workbench interface.

2.1.5 IBM SPSS STATISTICS

IBM SPSS Statistics is a powerful statistical software platform. It offers a user-friendly interface and a robust set of features that lets your organization quickly extract actionable insights from your data. Advanced statistical procedures help ensure high accuracy and quality decision making. All facets of the analytics lifecycle are included, from data preparation and management to analysis and reporting.

OPERATIONS CARRIED OUT ON DATASETS USING IBM SPSS STATISTICS 25

1. Entering values into the data view window to create a table and modelling the table in the variable view window of IBM SPSS.
2. Carrying out basic Test operations such as;
    - CHI SQUARE TEST

- T TEST
- CORRELATION AND REGRESSION TEST
- ONE AND TWO-WAY ANOVA TEST
- DESCRIPTIVE TEST
- FORECASTING TEST
- PLOTTING GRAPHS

2.1.5.1 CHI-SQUARE TEST

The Chi-Square test measures how one categorical variable associate with another categorical variable. In the above dataset, the two categorical variables are Sex and Smoking status. The null hypothesis for the dataset holds that Gender and Smoking Status are independent variables in the population**.**

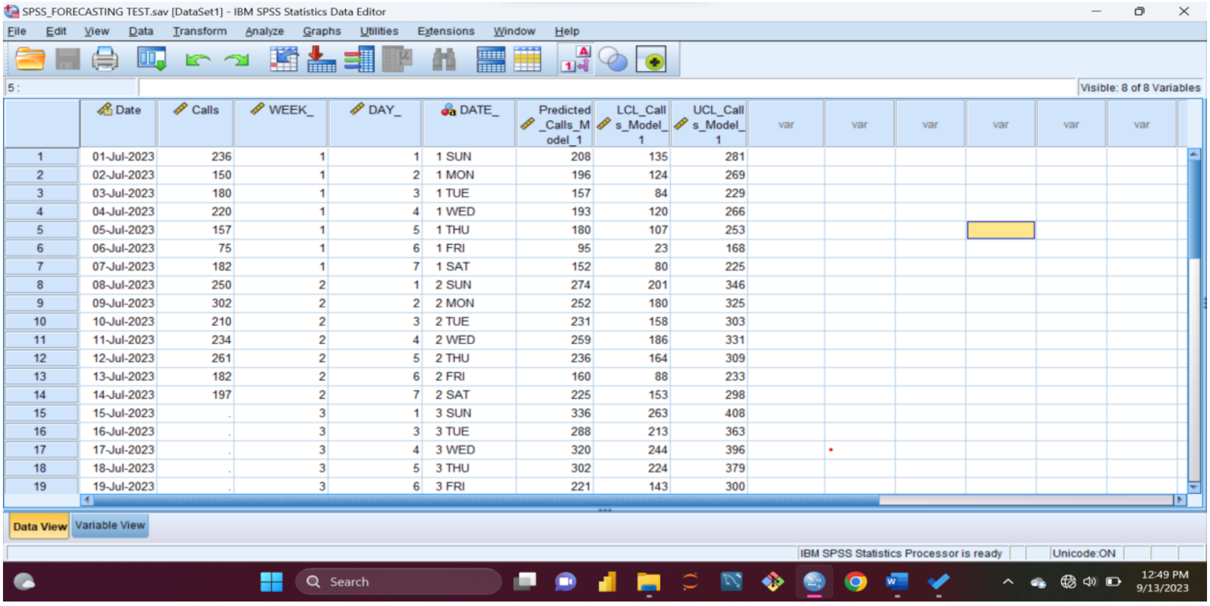
	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.400[a]	1	.527		
Continuity Correction[b]	.000	1	1.000		
Likelihood Ratio	.403	1	.526		
Fisher's Exact Test				1.000	.500
N of Valid Cases	10				

From Pearson's Chi-Square, it is obtained that the p-value for the dataset is 0.527 which is greater than 0.05. Hence, the null hypothesis is accepted.

2.1.5.2 FORECASTING TEST

Forecasting Test is used for carrying out predictive analytics. It uses past records to generate inferences for the future.

For this test, a dummy data containing number of calls received by the customer service of a network provider for a period of 14 days. A forecasting test is to be carried out to enable the company to predict number of calls expected for the following week to enable them know the number of staffs required each day to better attend to calls and reduce customer waiting time.



	Date	Calls	WEEK_	DAY_	DATE_	Predicted_Calls_Model_1	LCL_Calls_Model_1	UCL_Calls_Model_1	var	var	var	var	var	var
1	01-Jul-2023	236	1	1	1 SUN	208	135	281						
2	02-Jul-2023	150	1	2	1 MON	196	124	269						
3	03-Jul-2023	180	1	3	1 TUE	157	84	229						
4	04-Jul-2023	220	1	4	1 WED	193	120	266						
5	05-Jul-2023	157	1	5	1 THU	180	107	253						
6	06-Jul-2023	75	1	6	1 FRI	95	23	168						
7	07-Jul-2023	182	1	7	1 SAT	152	80	225						
8	08-Jul-2023	250	2	1	2 SUN	274	201	346						
9	09-Jul-2023	302	2	2	2 MON	252	180	325						
10	10-Jul-2023	210	2	3	2 TUE	231	158	303						
11	11-Jul-2023	234	2	4	2 WED	259	186	331						
12	12-Jul-2023	261	2	5	2 THU	236	164	309						
13	13-Jul-2023	182	2	6	2 FRI	160	88	233						
14	14-Jul-2023	197	2	7	2 SAT	225	153	298						
15	15-Jul-2023	.	3	1	3 SUN	336	263	408						
16	16-Jul-2023	.	3	3	3 TUE	288	213	363						
17	17-Jul-2023	.	3	4	3 WED	320	244	396						
18	18-Jul-2023	.	3	5	3 THU	302	224	379						
19	19-Jul-2023	.	3	6	3 FRI	221	143	300						

FIG 5. A TYPICAL IBM SPSS WINDOW VIEW

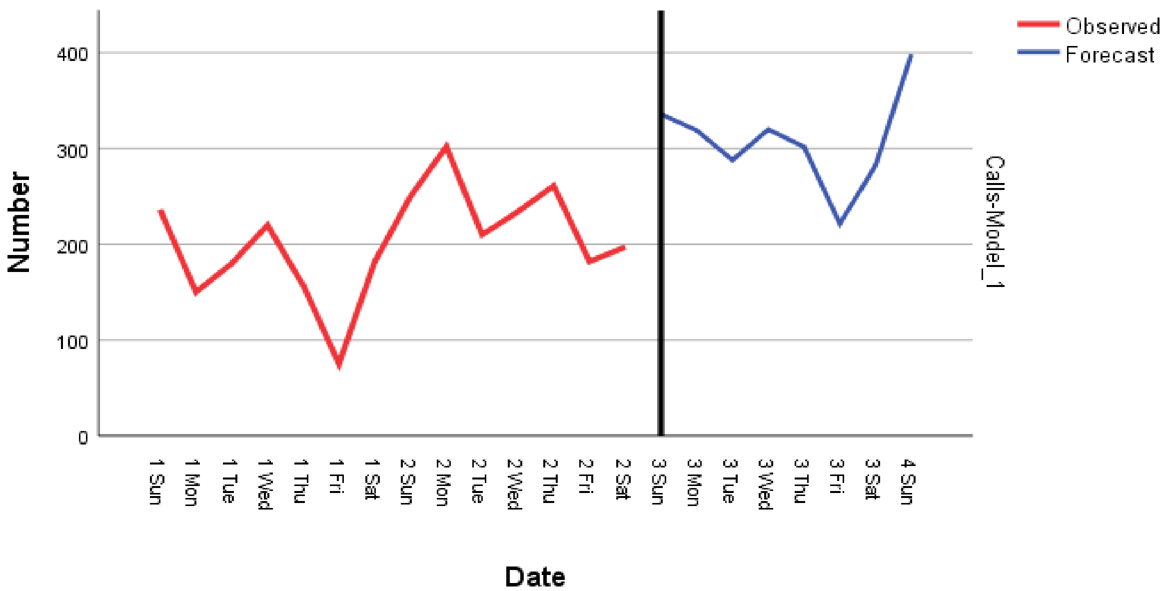


FIG 6. GRAPH SHOWING RESULT OF FORECASTING TEST

2.1.5.3 DESCRIPTIVE TEST

Descriptive Test basically gives information about the characteristics of a dataset. It could include; measure of central tendency and measure of central dispersion.

For this test, a dummy dataset containing 10 random age variable was created.

Statistics

Age

N	Valid	10
	Missing	0
Mean		45.70
Std. Error of Mean		8.367
Median		44.00
Mode		16[a]
Std. Deviation		26.458
Variance		700.011
Range		82
Minimum		16
Maximum		98
Sum		457

a. Multiple modes exist. The smallest value is shown

Age

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	16	2	20.0	20.0	20.0
	23	1	10.0	10.0	30.0
	34	1	10.0	10.0	40.0
	35	1	10.0	10.0	50.0
	53	2	20.0	20.0	70.0
	54	1	10.0	10.0	80.0
	75	1	10.0	10.0	90.0
	98	1	10.0	10.0	100.0
	Total	10	100.0	100.0	

2.1.5.4 ANALYSIS OF VARIANCE [ANOVA] TEST

ANOVA test is used when a categorical variable (independent) and a continuous Variable (dependent) exist in a dataset. It helps compare group means to find out if they are statistically different or they are similar. It could be a One-way ANOVA/single factor ANOVA or a Two-way ANOVA/full Factorial ANOVA.

For this test, One-Way ANOVA is employed on a dataset containing Test methods (classroom, online and blend) and the respective Test scores of Students. The aim is to determine if there exist a significant statistical difference among the groups.

ONE-WAY ANOVA

Scores

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	160.000	2	80.000	.085	.919
Within Groups	11338.400	12	944.867		
Total	11498.400	14			

There was no statistically significant difference between the groups as demonstrated by one-way ANOVA ($F(2,12) = 0.085$, $p = 0.919$). For there to exist a significant difference, alpha (p) should be less than the standard 0.05 alpha value).

2.1.5.5 T TEST

The T Test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features.

The t-test for the difference in means is an hypothesis test that tests the null hypothesis that the means for both groups are equal, versus the alternative hypothesis that the means are not equal (2-tail) or that the mean for one of the groups is larger than the mean for the other group (1-tail).

In this test, a dataset containing learning types (classroom and online) of students and their respective scores was used. The aim is to determine if there is a significant different between the means (classroom and online). Hence, the null hypothesis is tested.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means	
		F	Sig.	t	df
Scores	Equal variances assumed	2.107	.164	.418	18
	Equal variances not assumed			.418	11.515

Group Statistics

	learning	N	Mean	Std. Deviation	Std. Error Mean
Scores	.00	10	11.60	10.627	3.361
	1.00	10	10.10	4.012	1.269

Independent Samples Test

		t-test for Equality of Means			
					95% Confidence Interval of the Difference
		Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower
Scores	Equal variances assumed	.681	1.500	3.592	-6.047
	Equal variances not assumed	.684	1.500	3.592	-6.363

Independent Samples Test

		t-test for Equality of Means
		95% Confidence Interval of the Difference
		Upper
Scores	Equal variances assumed	9.047
	Equal variances not assumed	9.363

From the above, the p-value is obtained from Sig. (2-tailed) which is 0.684. This means that the difference in means is not statistically significant. Therefore, the null hypothesis holds.

2.1.6 POWER BUSINESS INTELLIGENCE (BI)

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Your data might be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets you easily connect to your data sources, visualize and discover what's important, and share that with anyone or everyone you want.

Power BI consists of several elements that all work together, starting with these three basics:

- A Windows desktop application called Power BI Desktop.
- An online software as a service (SaaS) service called the Power BI service.
- Power BI Mobile apps for Windows, iOS, and Android devices.

WORKING ON THE VOLVE PRODUCTION DATASET USING POWER BI

The Steps taken when working on the Volve Production Dataset are as follows;

1. Loading the dataset into the Power Query Editor to clean, transform and organize the dataset to prepare it for use. Steps applied to achieve the above are as follows;
    - Get the data from source
    - Promoted Headers
    - Changed Data Type
    - Removed unwanted Columns
    - Removed Top Rows
    - Filtered Rows
    - Sorted Rows
    - Renamed Columns
    - Reordered Columns

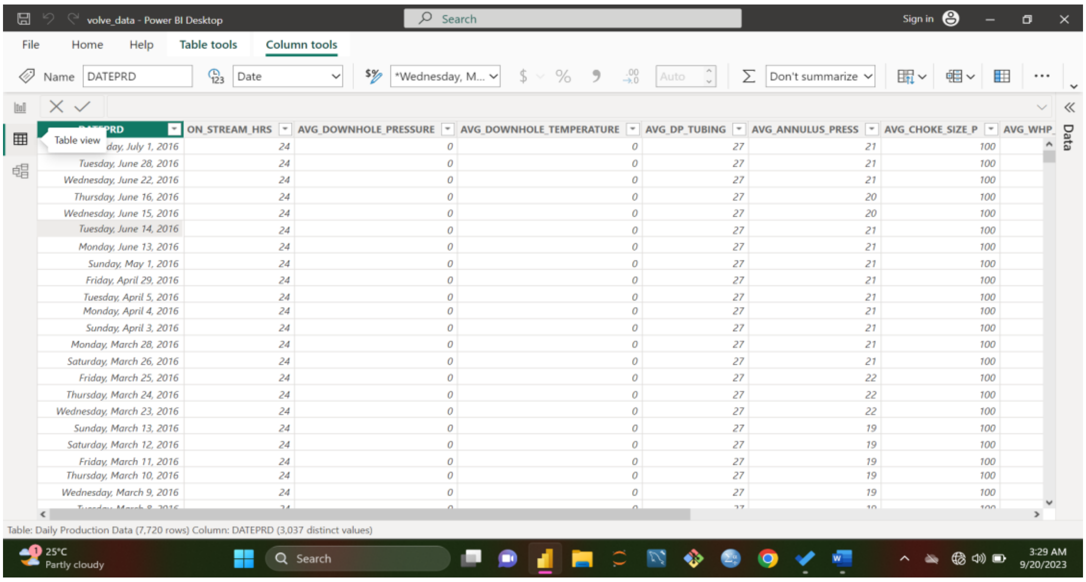


Fig 7. Image showing the Table View of the Transformed Volve Production Dataset.

2. Loading the transformed dataset into Power BI Desktop to model the dataset and create an interactive dashboard. Modelling the dataset involves linking common columns contained in various tables that makes up the dataset. Steps applied to achieve the above are as follows;
   - Linked the Norwegian Petroleum Directorate code (NPDcode) columns in the Daily and Monthly production table.
   - Linked “wellbore name” columns in both Daily and Monthly production table.
   - Visualized the sum of gas, oil and water produced using a Power BI card
   - Inserted a slicer containing NPDCode that serves as a filter for the entire dashboard.
   - Visualized the sum of gas produced by month using a funnel chart.
   - Visualized the sum of gas produced by year using a clustered column chart
   - Visualized the sum of gas produced by each wellbore using a donut chart.

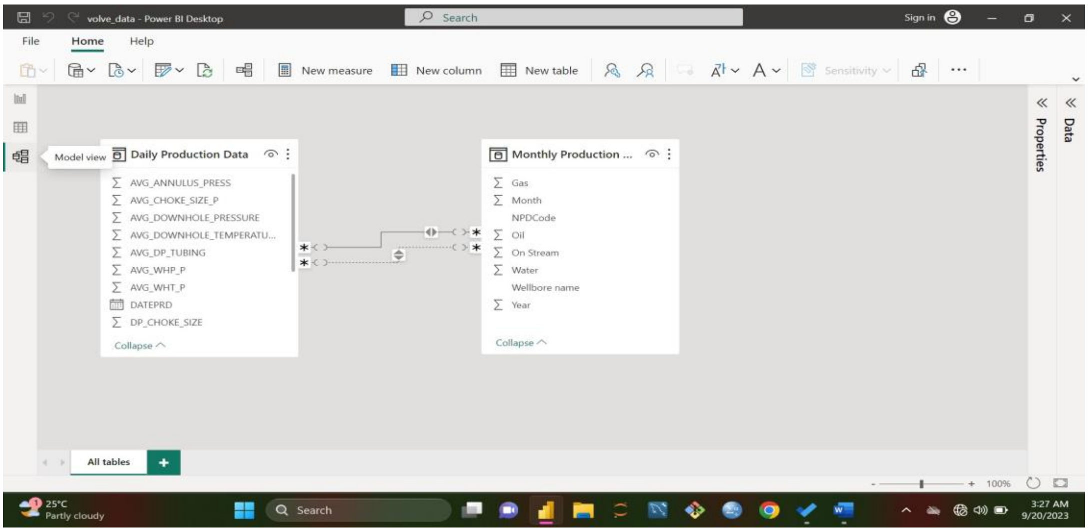


Fig 8. Image showing the Model view of Volve Production Dataset

3. Result Obtained from the above steps;
   - Total sum of gas produced is 1.19 billion cubic meters from 2008 to 2016
   - Total sum of oil produced is 8.06 million cubic meters from 2008 to 2016
   - Total sum of water produced is 14.78 million cubic meters from 2008 to 2016
   - Well '15/9-F-12' produced the most gas with about 45.25 percent of the total gas produced.
   - Between 2008 and 2016, sum of gas produced was highest in 2010.
   - Sum of gas produced by month was highest in may for the period of year 2008 to 2016.

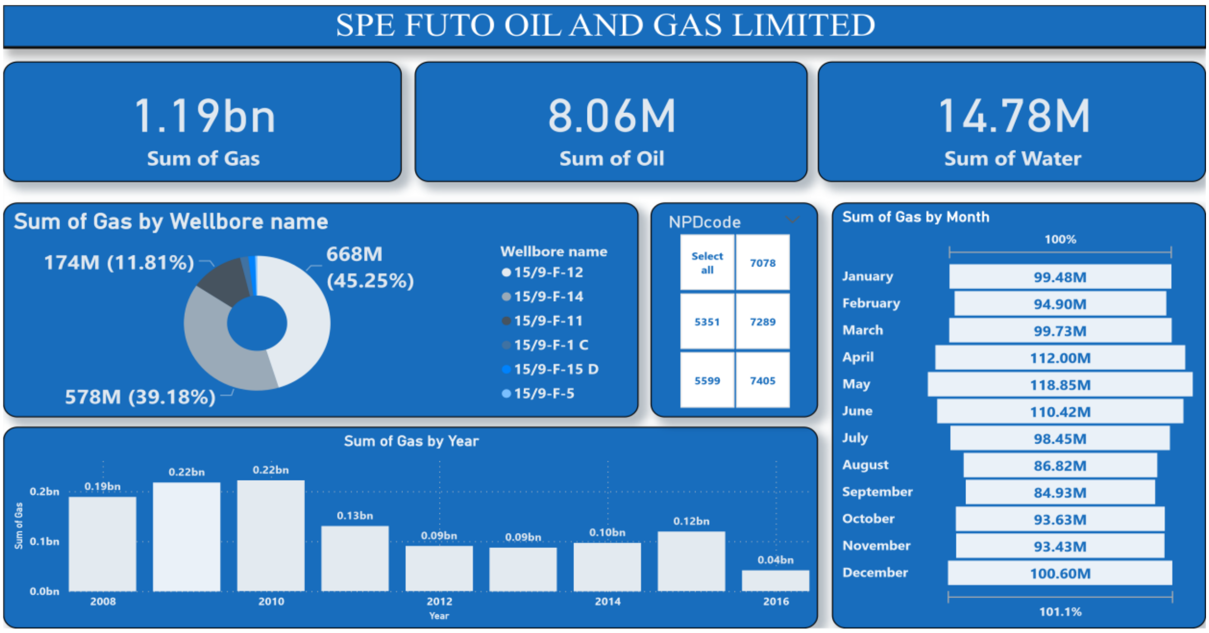


fig 9. Image showing an interactive Dashboard of the Volve Production Dataset.

CHAPTER THREE

3.0 WHY OIL AND GAS COMPANIES MUST ACT ON ANALYTICS

According to V. Tan et al in their Data Analytics for Effective Project Management in the Oil and Gas Industry publication, the oil and gas industry play a significant role in the global economy as the world's main fuel sources. Oil and gas demand are projected to continue to rise for the next 20 years. This perspective emphasizes the role of improving production chain efficiency through innovation to meet growing demand. Today's powerful tools use a combination of state-of-the-art engineering, data science (which also includes data analysis), and computing power to identify superior solutions to complex production optimization problems. They will not replace the conventional models and physical understanding of O&G asset operation—they will supplement them, filling in the performance gaps that hold back production.

Data analytics incorporates numerous tools, methods, and processes for the analysis and management of the data. It includes gathering raw data, classifying and organizing the data, storing it, and using statistical analysis methods to derive trends and resolve problems for faster and more accurate decision-making. Data analytics enables analysts to focus on a specific or particular set or population from a huge pool of data gathered. Oil and gas firms can leverage data analytics to mine increased volume of gas and oil from reservoirs, reduce their costs (operational and initial capital investment), and improve investment decision-making accuracy.

CHAPTER FOUR

4.0 CONCLUSION

Companies now realize that data constitute a vital commodity and the value of data can be realized through the power of data analytics (Saputelli 2016). Leveraging hidden insights from mining data can help the oil and gas industry make faster and better decisions that can reduce operational costs, improve efficiency, and increase production and reservoir recovery. Data analytics can thus play an important role in reducing the risks inherent in the development of subsurface resources. These analytic advantages can improve production gains by 6 to 8% (Bertocco and Padmanabhan 2014). While data analytics has broad applications in reservoir engineering, the vast number of wells and pace of operations in unconventionals allow data to play a critical role in the decisions that create value.

With the above, I conclude that for this internship as a Data Analyst, the knowledge of tools and software proved to be beneficial not only for the tasks that were assigned to me, also it proves beneficial to me as an aspiring Petroleum Engineer that will eventually solve problems facing the energy sector by blending Domain knowledge with Analytics. It was a great time interning for this I.T training and consultancy company and I'm glad I was able to pull through the various tasks assigned to me in record time. I have become more skilled particularly in analyzing data, and exposed to an ideal working environment.

4.1 LIMITATIONS

For the duration of the internship, the basic limitation I encountered is not having access to adequate petroleum engineering datasets due to data privacy of most Petroleum companies. Hence, I couldn't get the practical experience of carrying out data analysis on oil and gas datasets. Having access to them will be of great value to me, because it will enable me practice my newly acquired Data Analytics skill which in turn help the petroleum industry as an aspiring Petroleum Engineer.

Also, due to the new economic policy by the government regarding subsidy which resulted to inflation, the cost of transportation skyrocketed and it became a financial burden to me.

4.2 RECOMMENDATION

With reference to the above stated limitations, I would recommend that as much as possible, oil and gas datasets be made accessible to students and academia by the various oil and gas companies in Nigeria. This will go a long way to stimulate learning and research and better equip students to become valuable assets to the industry at large. For the case of financial burden, I will recommend that companies provide financial incentives outside stipulated salaries to students during the period of their internship with them to enable interns manage such situations, this is because interns also offer value to them.

4.3 REFERENCES

- Victor Tan, Kamarulzaman Ab. Aziz(B), and Seyed Hadi Razavi. (2022). Data Analytics for Effective Project Management in the Oil and Gas Industry. pp. 233–242. *https://doi.org/10.2991/978-94-6463-080-0_20*
- https://www.mckinsey.com/industries/oil-and-gas/our-insights/why-oil-and-gas-companies-must-act-on-analytics
- *https://g.co/kgs/dFXmf6*
- https://www.hostinger.com/tutorials/what-is-mysql
- *https://www.ibm.com/products/spss-statistics*
- *https://github.com/adityakumaar/DataAnalytics-Internship-Project/blob/master/Reports%20and%20Presentations/Aditya%20Kumar%20-%20Internship%20Report.pdf*
- https://learn.microsoft.com/en-us/power-bi/fundamentals/desktop-getting-started