Project 1. AdventureWorks

Group 4 (The best one!)

* Ibrahim Saed
* Ezinne Ndubuisi
* Chahat Mahajan
* Natalia Popa

General Info:

DB: AdventureWorks

* Tables
* Views

Tables divided into categories of:

- Human Resources

- Person

- Production

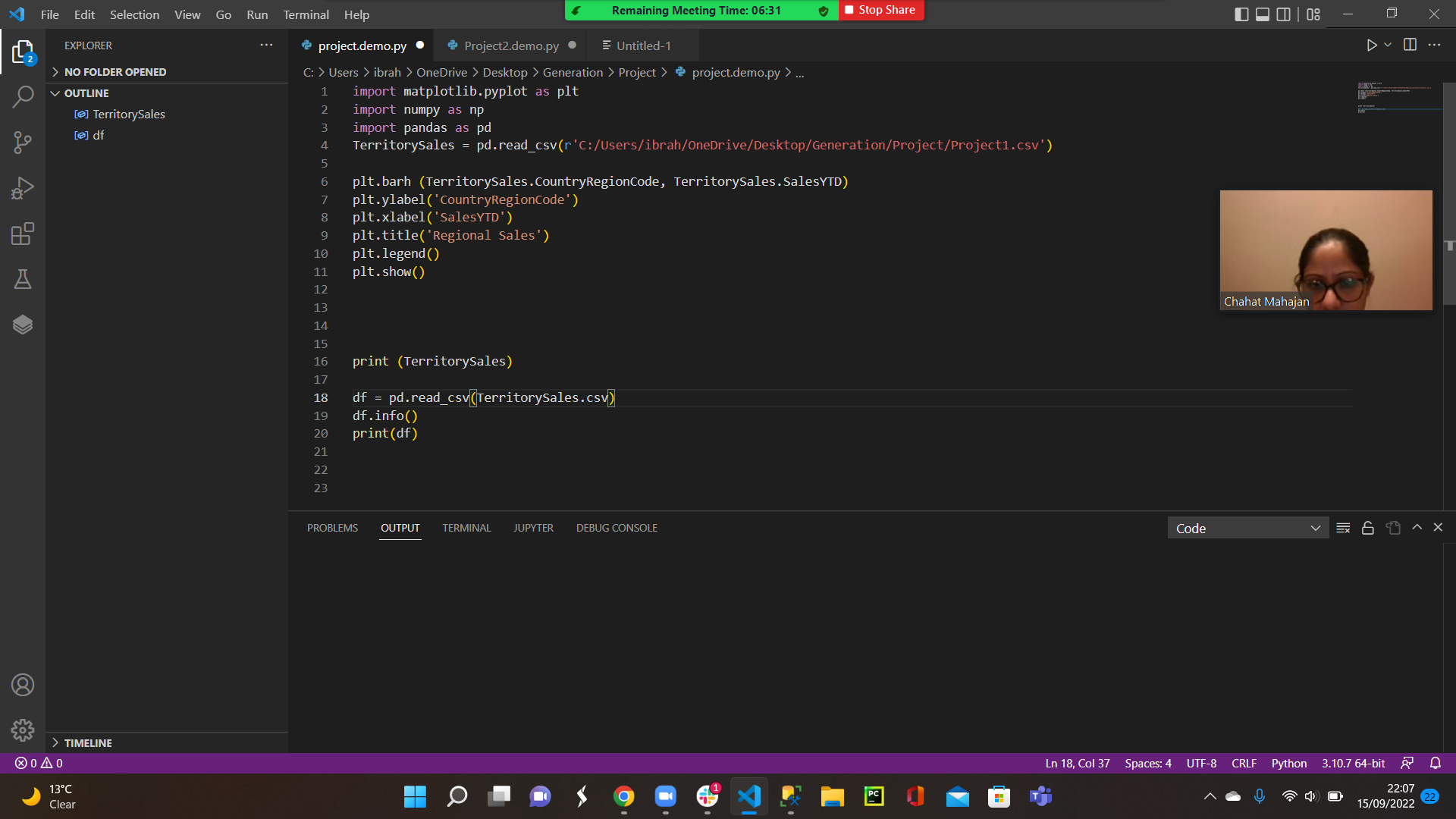
- Purchasing

- Sales

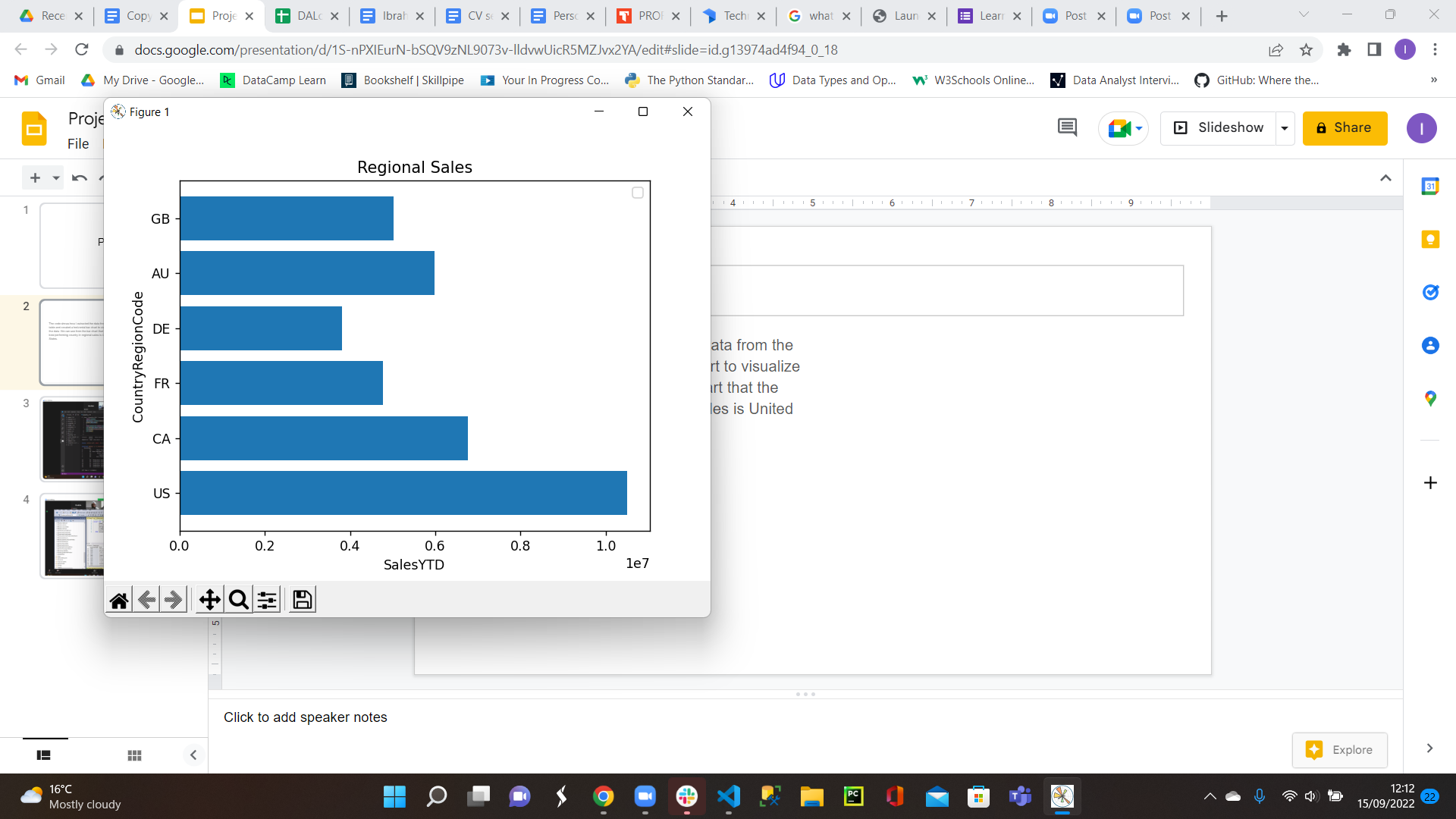
Software used: SQL & Python

**Question 1. What are the regional sales in the best performing country?**

To find the answer I had to the columns is needed from the database and import it to visual studio. From there I started creating the horizontal bar graph, naming the x and y axis and created a title. Out of all the different graphs we felt like the horizontal bar graph was the best fit for visualizing the data.



Result - From the graph we can see that the best performing country is The United States.



**Question 2. What is the relationship between annual leave taken and bonus?**

The **annual leave** variable is a part of HumanResources.Employee table.

We will bring all the columns of HumanResources.Employee table to explore the content of it by using the SQL code:

select \*

from HumanResources.Employee

For the purpose of the project, after exploring the table we will bring the following fields of the table for the future analysis

| Column name | DataType | Description |
| --- | --- | --- |
| JobTitle | nvarchar(50) | Work title such as Buyer or Sales Representative |
| HireDate | date | Employee hired on this date. |
| SalariedFlag | bit | Job classification.  0 = Hourly, not exempt from collective bargaining.  1 = Salaried, exempt from collective bargaining.  Default: 1 |
| VacationHours | smallint | Number of available vacation hours.  Default: 0 |
| SickLeaveHours | smallint | Number of available sick leave hours.  Default: 0 |

While we are interested in the relationship between annual leave taken and bonus, a few variables are investigated to draw a bigger picture of the situation related to bonus variable.

One of the variables of interest is annual leave taken, which is not specified if it is related to vacation or sick leave, therefore we will investigate all three situations: vacation leave, sick leave, and total leave, were

TotalLeaveHours = VacationHours+ SickLeaveHours

The **bonus** variable is a part of Sales.SalesPerson table.

We will bring all the columns of Sales.SalesPerson table to explore the content of it by using the SQL code:

select \*

from Sales.SalesPerson

For the purpose of the project the following fields are of interest

| ColumnName | DataType | Description |
| --- | --- | --- |
| SalesQuota | money | Projected yearly sales. ([SalesQuota]>(0.00)) |
| Bonus | money | Bonus due if quota is met. ([Bonus]>=(0.00)) |

The code that will bring the specified current variables and the new variable TotalLeaveHours calculated as specified above is

SELECT JobTitle, HireDate, SalariedFlag, VacationHours, SickLeaveHours,

(VacationHours+SickLeaveHours) AS TotalLeaveHours, SalesQuota, Bonus

FROM HumanResources.Employee AS hre

INNER JOIN Sales.SalesPerson AS ssp

ON hre.BusinessEntityID=ssp.BusinessEntityID

WHERE SalesQuota IS NOT NULL

After executing the code and getting the table based on required criteria, we export the table into a .csv file for future investigation by using Python.

1. The relationship between Total Annual Leave (Vacation + Bonus, in hours) taken and bonus is generated by the code (in Python)

#Plotting the graph of the Total Annual Leave (Sick + Vacation) versus Bonus Amount

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

LeaveBonus = pd.read\_csv(r'C:/Users/44778/Desktop/ProjectDA\_1/Question2.csv')

print(LeaveBonus)

plt.scatter(LeaveBonus.TotalLeaveHours, LeaveBonus.Bonus)

plt.xlabel('Total Annual Leave (in Hours)')

plt.ylabel('Annual Bonus amount (in $)')

plt.title('Total Annual Leave vs Bonus Amount', fontsize=16, color='tab:blue')

m, b = np.polyfit(LeaveBonus.TotalLeaveHours, LeaveBonus.Bonus, deg=1)

plt.axline(xy1=(0, b), slope=m, color='r', label=f'$y = {m:.2f}x {b:+.2f}$')

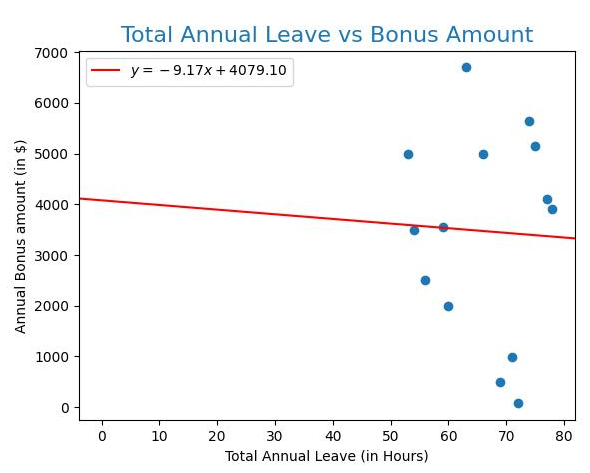
plt.legend()

plt.show()

corr = np.corrcoef(LeaveBonus.TotalLeaveHours, LeaveBonus.Bonus)

print("Correlation: " + str(corr))

Result:



2. The relationship between Vacation Annual Leave (in hours) taken and bonus is generated by the code (in Python)

#Plotting the graph of the Vacation Annual Leave versus Bonus Amount

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

LeaveBonus = pd.read\_csv(r'C:/Users/44778/Desktop/ProjectDA\_1/Question2.csv')

print(LeaveBonus)

plt.scatter(LeaveBonus.VacationHours, LeaveBonus.Bonus, color='green')

plt.xlabel('Vacation Annual Leave (in Hours)')

plt.ylabel('Annual Bonus amount (in $)')

plt.title('Vacation Annual Leave vs Bonus Amount', fontsize=16, color='tab:blue')

m, b = np.polyfit(LeaveBonus.VacationHours, LeaveBonus.Bonus, deg=1)

plt.axline(xy1=(0, b), slope=m, color='r', label=f'$y = {m:.2f}x {b:+.2f}$')

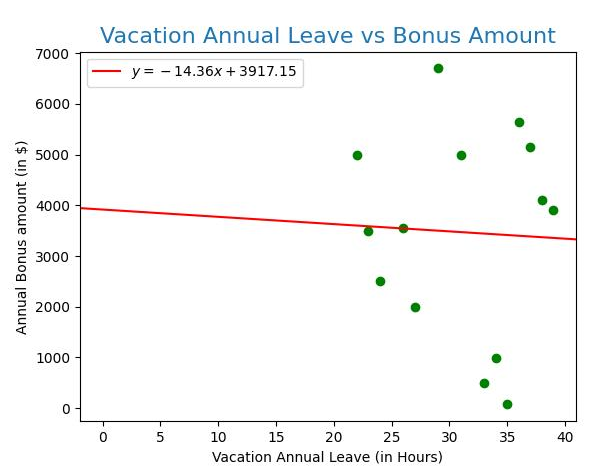
plt.legend()

plt.show()

corr = np.corrcoef(LeaveBonus.VacationHours, LeaveBonus.Bonus)

print("Correlation: " + str(corr))

Result:



3. The relationship between Sick Annual Leave (in hours) taken and bonus is generated by the code (in Python)

#Plotting the graph of the Sick Annual Leave versus Bonus Amount

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

LeaveBonus = pd.read\_csv(r'C:/Users/44778/Desktop/ProjectDA\_1/Question2.csv')

print(LeaveBonus)

plt.scatter(LeaveBonus.SickLeaveHours, LeaveBonus.Bonus, color='tomato')

plt.xlabel('Sick Annual Leave (in Hours)')

plt.ylabel('Annual Bonus amount (in $)')

plt.title('Sick Annual Leave vs Bonus Amount', fontsize=16, color='tab:blue')

m, b = np.polyfit(LeaveBonus.SickLeaveHours, LeaveBonus.Bonus, deg=1)

plt.axline(xy1=(0, b), slope=m, color='r', label=f'$y = {m:.2f}x {b:+.2f}$')

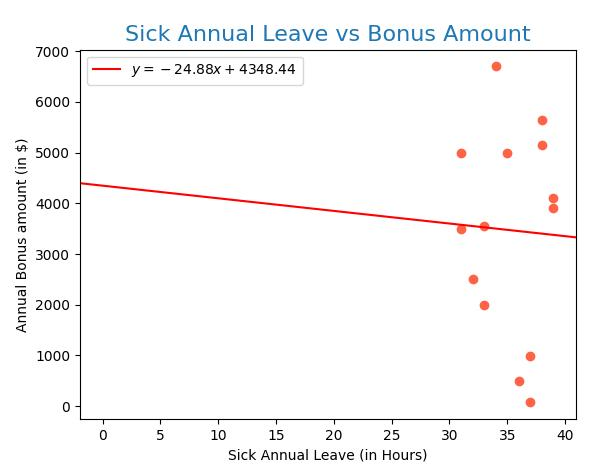
plt.legend()

plt.show()

corr = np.corrcoef(LeaveBonus.SickLeaveHours, LeaveBonus.Bonus)

print("Correlation: " + str(corr))

Result:



**Question 3. What is the relationship between Country and Revenue?**

From the Adventureworks2019 database,Identify the columns needed for the analysis

The columns needed for this analysis are the [SalesYTD] and [CountryRegionCode] columns located in the SalesTerriotory table of the Sales Schema.

**STEP 2:**

Write an Sql query of generate total revenue for each country

SELECT CountryRegionCode AS Country,SUM(SalesYTD) AS Revenue,[Group]

FROM Sales.SalesTerritory

GROUP BY CountryRegionCode,[Group]

ORDER BY Revenue

Which gives this result

Country Revenue Group

DE 3805202.3478 Europe

FR 4772398.3078 Europe

GB 5012905.3656 Europe

AU 5977814.9154 Pacific

CA 6771829.1376 North America

US 26411059.8792 North America

**STEP 3 :**

To import this tabular data into python for visualization , we need to install pandas in python. We do that using pip manager. Afterwards, we also install matplotlib to plot the graph.

Code used on python

import pandas as pd

RevenueByCountry = pd.read\_csv(r"C:\Users\idika\OneDrive\Documents\INTERIM PROJECT\RevenueByCountry.csv")

print(RevenueByCountry)

from matplotlib import pyplot as plt

plt.bar(RevenueByCountry.Country,RevenueByCountry.Revenue)

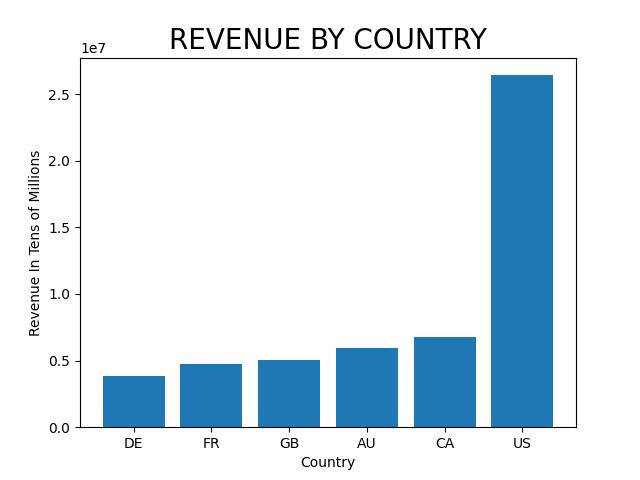
plt.xlabel("Country")

plt.ylabel("Revenue In Tens of Millions")

plt.title("REVENUE BY COUNTRY", fontsize =20)

plt.show()

**Result:**



**INFERENCE;**

The two countries in North America have higher revenues,with USA being the highest.

**Question 4. What is the relationship between sick leave and Job Title (PersonType)?**

* To find the relationship between sick leave and Job Title(Person Type) we tried to understand the snowflake schema from Adventure Works Cycles Data Dictionary, it shows the Fact table HumanResources.Employee is linked with Person.Person table with Foreign key BusinessEntityID.
* We Joined both tables on BusinessEntityID with inner join, from the table HumaResources. We took the Job Title and Sick Leave column. From the table Person.Person we took the Person Type column.
* Initially we thought to make categories according to person type i.e EM and SP. But when we joined the table and rolled up the row of Job titles it gave 63 EM job titles and only 4 SP job titles.
* So we made the categories of Job title using CASE in SSMS with the following query and saved it as a VIEW:

SELECT PersonType, JobTitle, SickLeaveHours,

CASE

WHEN JobTitle Like '%Executive%' THEN 'Executive'

WHEN JobTitle Like '%President%' THEN 'President'

WHEN JobTitle LIKE '%Manager%' THEN 'Manager'

WHEN JobTitle Like '%Engineer%' THEN 'Engineer'

WHEN JobTitle Like '%Designer%' THEN 'Designer'

WHEN JobTitle Like '%Marketing%' THEN 'Marketing'

WHEN JobTitle Like '%Production%' THEN 'Production'

WHEN JobTitle Like '%Shipping%' THEN 'Shipping'

WHEN JobTitle Like '%Stocker%' THEN 'Stocker'

WHEN JobTitle Like '%Quality%' THEN 'Quality'

WHEN JobTitle Like '%Scheduling%' THEN 'Scheduling'

WHEN JobTitle Like '%Master%' THEN 'Master'

WHEN JobTitle Like '%Facilities%' THEN 'Facilities'

WHEN JobTitle Like '%Maintenance%' THEN 'Maintenance'

WHEN JobTitle Like '%Janitor%' THEN 'Janitor'

WHEN JobTitle Like '%Human Resources%' THEN 'HR'

WHEN JobTitle Like '%Accounts%' THEN 'Accounts'

WHEN JobTitle Like '%Buyer%' THEN 'Buyer'

WHEN JobTitle Like '%Purchasing%' THEN 'Purchasing'

WHEN JobTitle Like '%Network%' THEN 'Network'

WHEN JobTitle Like '%Application%' THEN 'Application'

WHEN JobTitle Like '%Database%' THEN 'Database'

WHEN JobTitle Like '%Sales%' THEN 'Sales'

WHEN JobTitle Like '%%' THEN 'Engineer'

ELSE 'Misc' END AS Titles

FROM HumanResources.Employee AS e

INNER JOIN Person.Person AS p

ON e.BusinessEntityID = P.BusinessEntityID

* Then Selected the Job titles, Avg sick leave hours from the view:

SELECT Titles, AVG(SickLeaveHours) AS avg\_sickleavesFROM SickLeaves

GROUP BY Titles

ORDER BY avg\_sickleaves DESC;

* Exported the result and stored as csv file on desktop
* In Visual Studio Code Installed pandas, matplotlib
* We made the dataframe named ‘leaves’ with pandas and then plotted a horizontal bar graph with variable Job titles on x-axis and Avg sick leaves

on the y-axis.

* Gave labels on x-axis, y-axis along with title.

import pandas

leaves = pandas.read\_csv(r'C:/Users/Generation UK&I\Documents/VIEW.csv')

print(leaves)

from matplotlib import pyplot as plt

plt.barh(leaves.Titles, leaves.avg\_sickleaves)

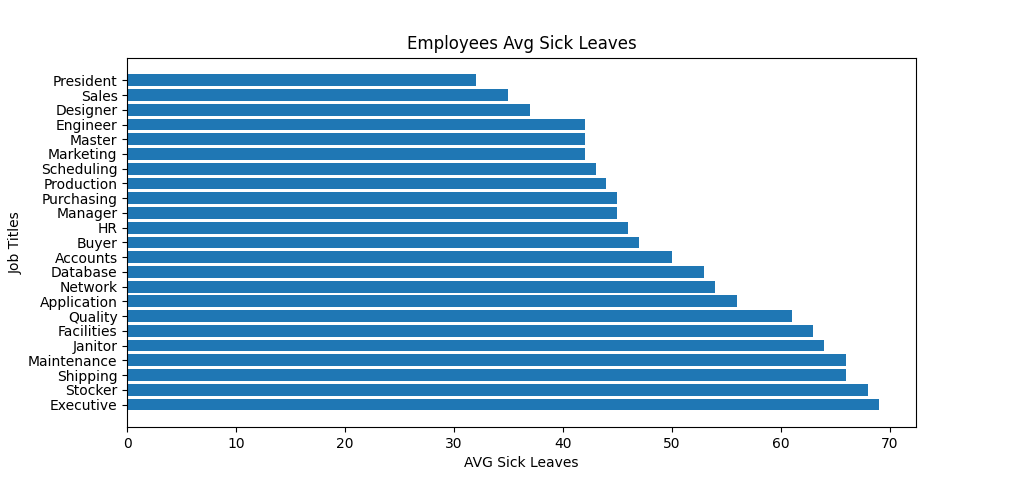
plt.ylabel('Job Titles')

plt.xlabel('AVG Sick Leaves')

plt.title('Employees Avg Sick Leaves')

plt.show()

**Result:**



**INFERENCE:**

The bar graph illustrates plotting average sick leaves (in hours) taken by employees in adventureworks in the year 2019. The maximum average hours of sick leave (68) were taken by Executives and the minimum avg by Presidents (32). The mean value is 50 hours in the data cohort for the year 2019 across different job titles in the company. However there are certain limitations to this graph. There is a degree of confounding amongst job titles in terms of repetition which were minimized however cannot be excluded. Certain job titles were grouped in miscellaneous and have not been displayed in the graph. The average number of sick leaves is across a particular job title however this does not reflect the true pattern of leaves taken by an individual. The graph also makes the assumption that all employees have joined the company at the same time from the point of initiation of data collection. In addition, there is non clarity on the inclusion of employees who have taken sick leaves and who have resigned during this time period.

It is important to note that there is no data provided on the reasons for taking the sick leaves. This is crucial information as the number of hours taken in sick leave will eventually affect the productivity, operations of the company including cost implications. And identification of the reasons will also imply whether any measures can be taken to address these. For example, if the reasons are related to personal health, then provision and engagement of occupational health physicians would be integral to address these issues and hence improve the attendance at work.

**Question 5. What is the relationship between store trading duration and revenue?**

* To get the relationship between store trading duration and revenue, we Joined three tables and one view to get the columns Total revenue and Store trading duration with the following query in SSMS.

SELECT DISTINCT SS.BusinessEntityID AS ID, LineTotal AS TotalRevenue, YearOpened as StoreDuration

FROM Sales.SalesOrderDetail AS SOD

INNER JOIN Sales.SalesOrderHeader AS SOH

ON SOH.SalesOrderID = SOD.SalesOrderID

JOIN Sales.Customer AS SC

ON SOH.CustomerID = SC.CustomerID

JOIN Sales.Store AS SS

ON SC.StoreID = SS.BusinessEntityID

JOIN Sales.vStoreWithDemographics AS SV

ON SS.BusinessEntityID = SV.BusinessEntityID

* Exported the result and stored as csv file on desktop
* In Visual Studio Code Installed pandas, matplotlib
* We made the dataframe named ‘sd’ with pandas and then plotted a horizontal bar graph with variable Store duration on x-axis and Total revenue on y-axis.
* Gave labels on x-axis, y-axis along with title.

import pandas

sd = pandas.read\_csv(r'C:/Users/Generation UK&I/Documents/sd.csv')

print(sd)

from matplotlib import pyplot as plt

plt.barh(sd.StoreDuration, sd.TotalRevenue)

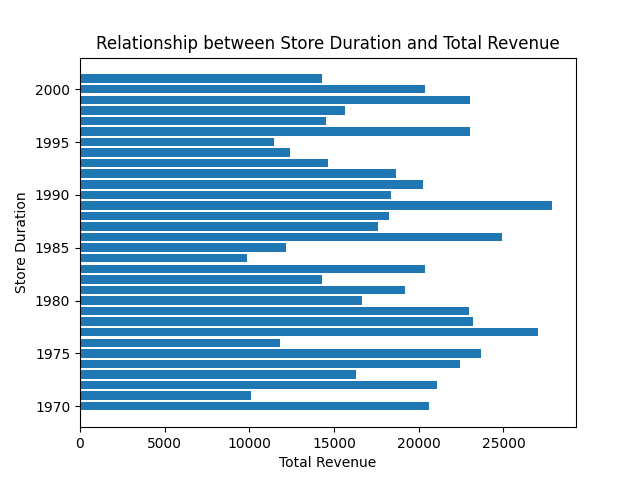
plt.xlabel('Store Duration')

plt.ylabel('Total Revenue')

plt.title('Relationship between Store Duration and Total Revenue')

plt.show()

**RESULT:**



**INFERENCE**:

The graph illustrates revenue generation across sales store trading for the company adventureworks. The total revenue generated was approx 600M with an average of 18.5M earned across 32 stores. The graph shows the maximum revenues generated from stores incepted in the years 1977, 1986 and 1989 which are in excess of 25M. The minimum revenue generated was from the stores incepted in the years 1971 and 1984 of the value of 10M. However there are certain limitations. The time period of data collection is not specified. In addition, there is no data if any trading stores were closed down or have remained dormant in trading.

**Question 6. What is the relationship between the size of the stores, number of employees and revenue?**

To answer this question, we had to join four tables namely:

* [Sales].[SalesOrderDetail] was inner joined to [Sales].[SalesOrderHeader] using Sale Order ID key
* [Sales].[Customer] was joined to [SalesOrderHeader] using SalesOrderID
* [Sales].[Store] was joined to [Sales].[Customer] using storeID and BusinessEntityID
* Lastly the viewStore withdemographics was joined to the [Sales].[Store] table using BusinessEntityID

These tables are all in the sales schema

**SQL query:**

SELECT LineTotal AS TotalRevenue, SS.BusinessEntityID,NumberEmployees AS Employees,SquareFeet AS StoreSize

FROM Sales.SalesOrderDetail AS SOD

INNER JOIN Sales.SalesOrderHeader AS SOH

ON SOH.SalesOrderID = SOD.SalesOrderID

JOIN Sales.Customer AS SC

ON SOH.CustomerID = SC.CustomerID

JOIN Sales.Store AS SS

ON SC.StoreID = SS.BusinessEntityID

JOIN Sales.vStoreWithDemographics AS SV

ON SS.BusinessEntityID = SV.BusinessEntityID

-**The result of this query is saved as view**

CREATE VIEW store\_revenue AS

SELECT LineTotal AS TotalRevenue, SS.BusinessEntityID,NumberEmployees AS Employees,SquareFeet AS StoreSize

FROM Sales.SalesOrderDetail AS SOD

INNER JOIN Sales.SalesOrderHeader AS SOH

ON SOH.SalesOrderID = SOD.SalesOrderID

JOIN Sales.Customer AS SC

ON SOH.CustomerID = SC.CustomerID

JOIN Sales.Store AS SS

ON SC.StoreID = SS.BusinessEntityID

JOIN Sales.vStoreWithDemographics AS SV

ON SS.BusinessEntityID = SV.BusinessEntityID

-This view is stored on the database as:

[dbo].[store\_revenue]

With this view, we carry out a query to find distinct values of the ID,StoreS ize,Employees and Total Revenue.

SELECT DISTINCT

ID,

StoreSize,

SUM(Employees) AS employees,

SUM(TotalRevenue) AS Total\_Revenue

FROM dbo.store\_revenue

GROUP BY ID,StoreSize

ORDER BY ID

Afterwards, the result is exported as a csv file , and uploaded as a dataframe on python using pandas module.

The Python code:

#Question 6

#The relationship between the number of employees and revenue

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

TQ6 = pd.read\_csv(r'C:/Users/44778/Desktop/ProjectDA\_1/Question6.csv')

print(TQ6)

plt.scatter(TQ6.employees, TQ6.Total\_Revenue, color='tomato')

plt.xlabel('Number of employees’')

plt.ylabel('Total Revenue (in $)')

plt.title('Number of employees vs Total Revenue', fontsize=16, color='tab:blue')

m, b = np.polyfit(TQ6.employees, TQ6.Total\_Revenue, deg=1)

plt.axline(xy1=(0, b), slope=m, color='r', label=f'$y = {m:.2f}x {b:+.2f}$')

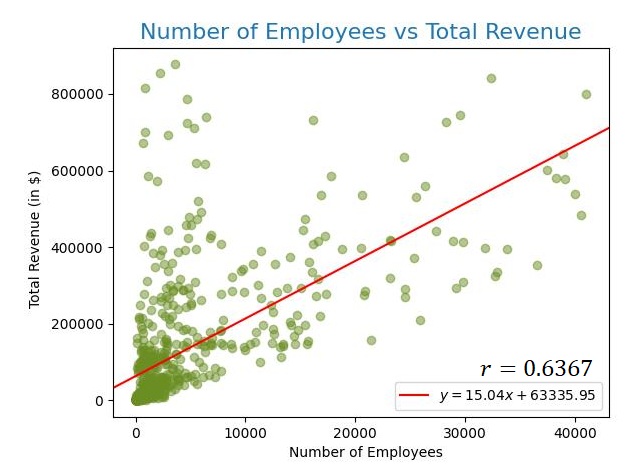
plt.legend()

plt.show()

corr = np.corrcoef(TQ6.employees, TQ6.Total\_Revenue)

print("Correlation: " + str(corr))

Results:



#Question 6

#The relationship between the number of employees and revenue

from matplotlib import pyplot as plt

import numpy as np

import pandas as pd

TQ6 = pd.read\_csv(r'C:/Users/44778/Desktop/ProjectDA\_1/Question6.csv')

print(TQ6)

plt.scatter(TQ6.employees, TQ6.Total\_Revenue, color='olivedrab',alpha=0.5)

plt.xlabel('Store Size (Sq.Feet)')

plt.ylabel('Total Revenue (in $)')

plt.title('Store Size vs Total Revenue', fontsize=16, color='tab:blue')

m, b = np.polyfit(TQ6.employees, TQ6.Total\_Revenue, deg=1)

plt.axline(xy1=(0, b), slope=m, color='r', label=f'$y = {m:.2f}x {b:+.2f}$')

plt.legend()

plt.show()

corr = np.corrcoef(TQ6.employees, TQ6.Total\_Revenue)

print("Correlation: " + str(corr))

