A picture containing text

Description automatically generated

Data Analytics Interim project

Adventure Works

Date: 16/09/2022

Group members:

Isaac Nagre

Kip Lemos

Tetiana Kushnirenko

Sherifat Odu

Stephen

Libraries used:

Matplotlib.pyplot - **data visualization** and graphical plotting library for Python

Pandas - used for **data science/data analysis and machine learning tasks**. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays, **one of the most important libraries in Python for Data Analysis, and Data Science**

NumPy - library **used for working with arrays** in many domains like **Scientific Computing, Deep Learning, and Financial analysis**

SciPy - library used **to solve scientific and mathematical problems**

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

SELECT SUM(SalesYTD) as Revenue, SUM(SalesLastYear) AS RevenueLastYear FROM Sales.SalesTerritory

WHERE CountryRegionCode = (SELECT abc.CountryRegionCode

FROM

    (SELECT CountryRegionCode, SUM(SalesYTD) AS RevenueYTD, SUM(SalesLastYear) AS RevenueLastYear

FROM Sales.SalesTerritory

GROUP BY CountryRegionCode) as abc

where abc.RevenueLastYear = (SELECT MAX(ab.RevenueLastYear)

FROM

    (SELECT CountryRegionCode, SUM(SalesYTD) AS RevenueYTD, SUM(SalesLastYear) AS RevenueLastYear

FROM Sales.SalesTerritory

GROUP BY CountryRegionCode) as ab))

group by SalesYTD, SalesLastYear;

Revenue RevenueLastYear

3072175.118 3205014.0767

7887186.7882 3298694.4938

2402176.8476 3607148.9371

2538667.2515 3925071.4318

10510853.8739 5366575.7098

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

us=pd.read\_csv(r'D:\py\us.csv')

print(us)

width = 0.4

x=np.arange(us.Region.count())

plt.bar(x-0.2, us.RevenueLastYear, width, color='royalblue',label="Last Year Revenue")

plt.bar(x+0.2, us.Revenue, width, color='coral',label="Current Year Revenue")

plt.title("Regional sales in the best performing country (US)"+'\n',fontsize=16,color='green')

plt.yticks([0,2e6,4e6,6e6,8e6,10e6,12e6,14e6,16e6],[0,'2m','4m','6m','8m','10m','12m','14m','16m'])

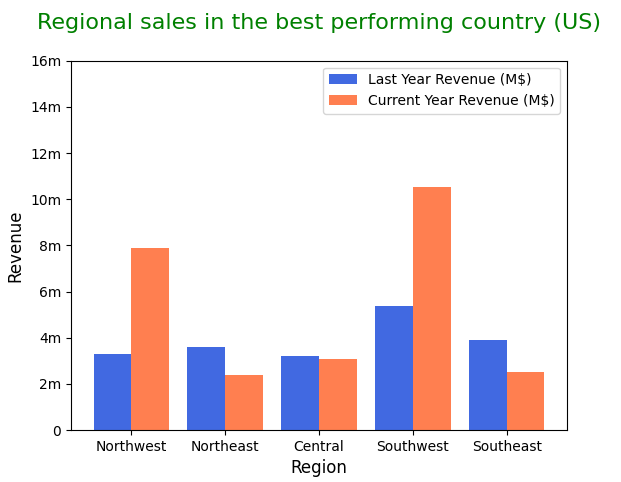
plt.xticks(x,us.Region)

plt.xlabel("Region",fontsize=12)

plt.ylabel("Revenue (M$)",fontsize=12)

plt.legend()

plt.show()



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

SELECT hr.VacationHours, sp.Bonus FROM HumanResources.Employee as hr

INNER JOIN Sales.SalesPerson as sp

ON hr.BusinessEntityID=sp.businessEntityID;

VacationHours Bonus

14 0.00

38 4100.00

27 2000.00

24 2500.00

33 500.00

29 6700.00

22 5000.00

26 3550.00

31 5000.00

23 3500.00

39 3900.00

20 0.00

36 5650.00

21 0.00

35 75.00

37 5150.00

34 985.00

import matplotlib.pyplot as plt

import pandas as pd

import scipy

bonus=pd.read\_csv(r'D:\py\bonus\_unsorted.csv')

plt.scatter(bonus.Bonus, bonus.VacationHours, color="royalblue")

plt.xlabel("Bonus ($)", fontsize=12)

plt.ylabel("Vacation Hours (hrs)", fontsize=12)

prs = round(scipy.stats.pearsonr(bonus.Bonus, bonus.VacationHours)[0],4)

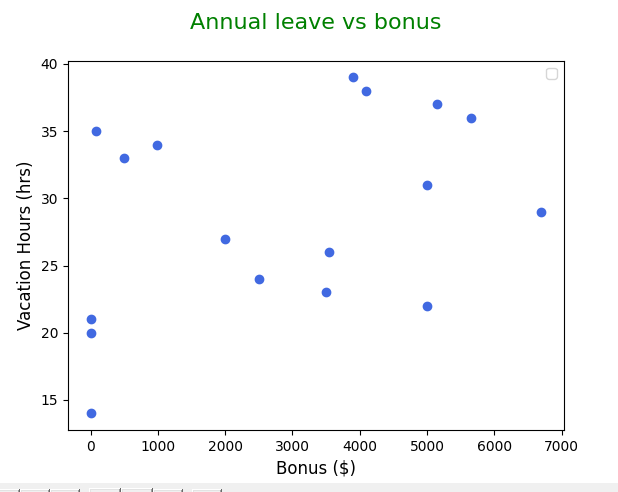
print('Pearson correlation coefficient=',prs)

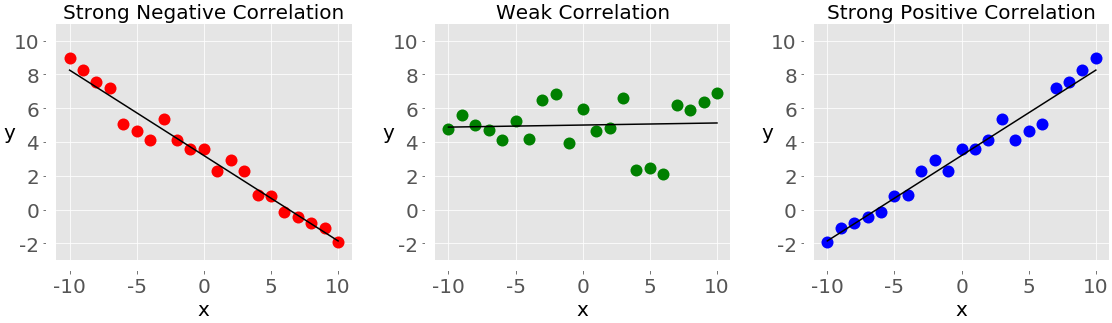
spr =round(scipy.stats.spearmanr(bonus.Bonus, bonus.VacationHours)[0],4)

print('Spearman rank correlation coefficient=',spr)

plt.title("Annual leave vs bonus"+'\n',fontsize=16, color='green')

plt.show()





Adding this screenshot to remind what linear correlations look like in Python, meaning Annual Leave and Bonus have a weak Correlation.

Pearson correlation coefficient=0.3821 which indicates a positive correlation, however, it is a low positive correlation as it does not lie within 0.8 - 1 range.

To confirm low correlation for non-linear relationships the Spearman's rank correlation coefficient can be calculated, which in this case is 0.4991 and confirms low correlation*.*

**3 - What is the relationship between Country and Revenue?**

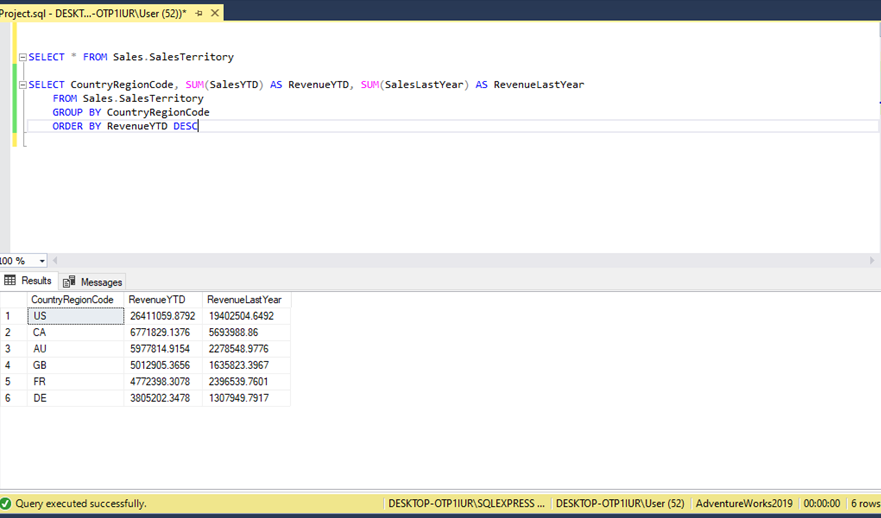
SELECT \* FROM Sales.SalesTerritory

SELECT CountryRegionCode, SUM(SalesYTD) AS RevenueYTD, SUM(SalesLastYear) AS RevenueLastYear

FROM Sales.SalesTerritory

GROUP BY CountryRegionCode

ORDER BY RevenueYTD DESC



|  |  |  |
| --- | --- | --- |
| CountryRegionCode | RevenueYTD | RevenueLastYear |
| US | 26411059.88 | 19402504.65 |
| CA | 6771829.138 | 5693988.86 |
| AU | 5977814.915 | 2278548.978 |
| GB | 5012905.366 | 1635823.397 |
| FR | 4772398.308 | 2396539.76 |
| DE | 3805202.348 | 1307949.792 |

import pandas as pd, numpy as np, matplotlib.pyplot as plt

revenuetable = pd.read\_csv("C:\\Users\\User\\Desktop\\interim project python\\Country Revenue.csv")

print(revenuetable)

width = 0.4

x=np.arange(revenuetable.CountryRegionCode.count())

plt.bar(x-0.2, revenuetable.RevenueLastYear, width, color='royalblue',label = "Last Year Revenue")

plt.bar(x+0.2, revenuetable.RevenueYTD, width, color ='coral',label= "Current Year Revenue")

plt.title("Relationship between country by revenue", fontsize=16, color="green")

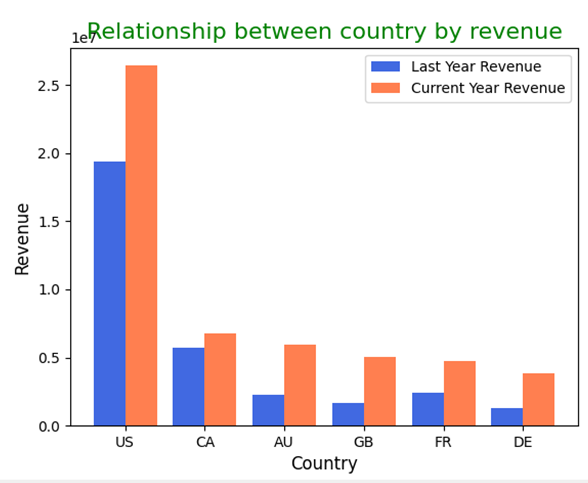
plt.xticks(x,revenuetable.CountryRegionCode)

plt.xlabel("Country", fontsize=12)

plt.ylabel("Revenue", fontsize=12)

plt.legend()

plt.show()



The grouped bar chart shows the relationship between last year revenue and current year revenue.  It indicates that the current year revenue of the countries is higher compared to the countries’ last year revenue. Overall, it shows that US  revenue is significantly higher in both years suggesting better performance in sales. 

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

SELECT JobTitle, sum(SickLeaveHours)

FROM HumanResources.Employees

ORDER by SickLeaveHours Desc

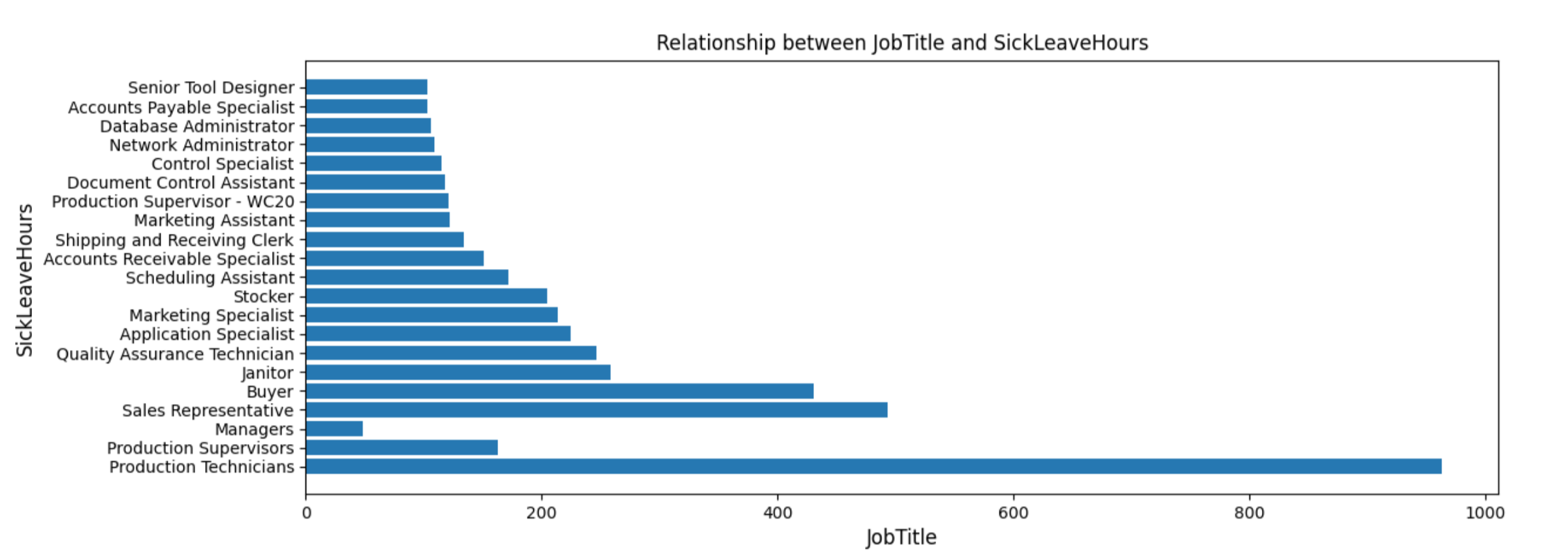
|  |  |
| --- | --- |
| JobTitle | SickLeaveHour |
| Production Technicians | 962.7143 |
| Production Supervisors | 162.5713 |
| Managers | 48.5 |
| Sales Representative | 493 |
| Buyer | 430 |
| Janitor | 258 |
| Quality Assurance Technician | 246 |
| Application Specialist | 224 |
| Marketing Specialist | 214 |
| Stocker | 205 |
| Scheduling Assistant | 172 |
| Accounts Receivable Specialist | 151 |
| Shipping and Receiving Clerk | 134 |
| Marketing Assistant | 122 |
| Production Supervisor - WC20 | 121 |
| Document Control Assistant | 118 |
| Control Specialist | 115 |
| Network Administrator | 109 |
| Database Administrator | 106 |
| Accounts Payable Specialist | 103 |
| Senior Tool Designer | 103 |

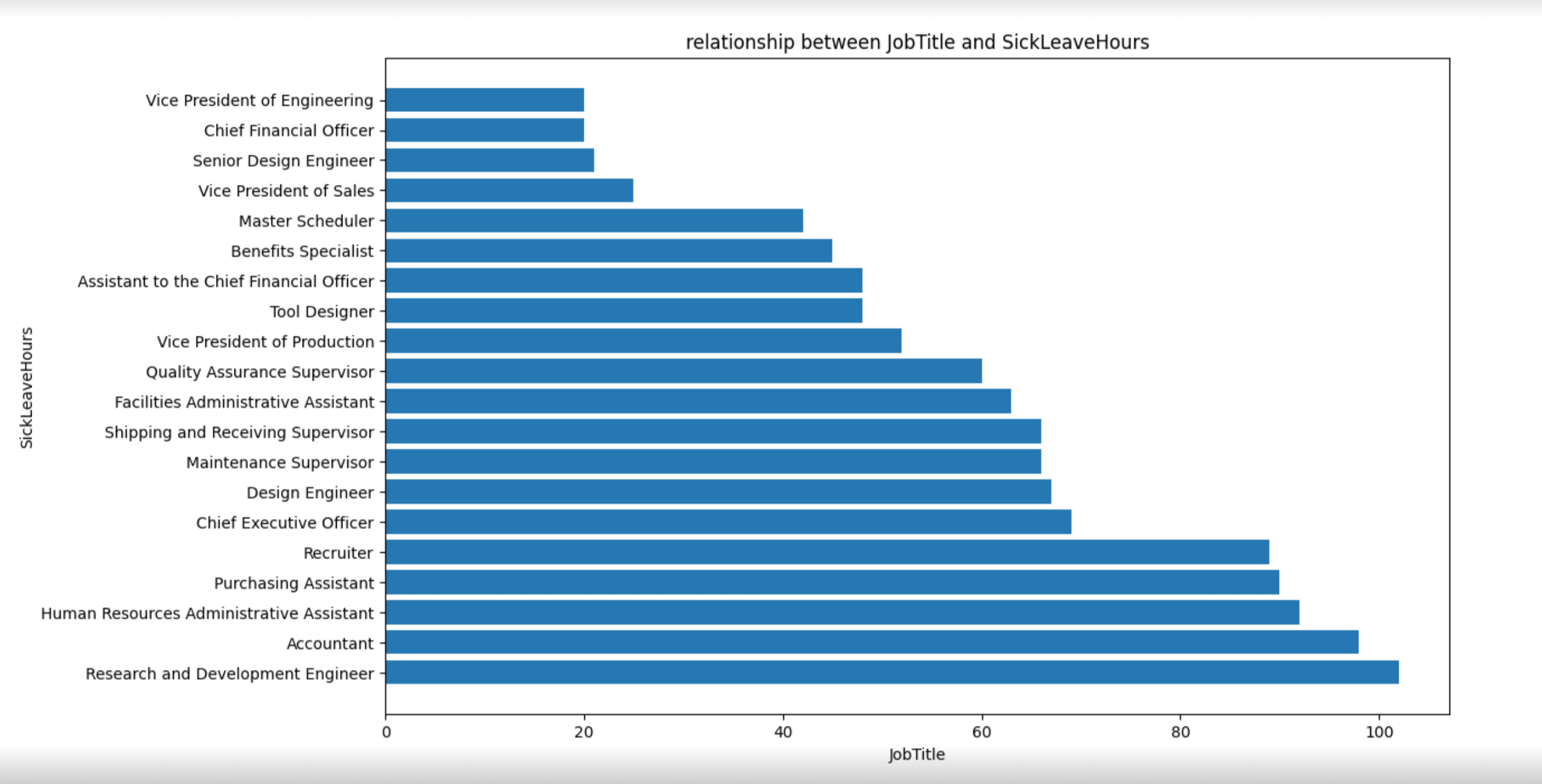
|  |  |
| --- | --- |
| Research and Development Engineer | 102 |
| Accountant | 98 |
| Human Resources Administrative Assistant | 92 |
| Purchasing Assistant | 90 |
| Recruiter | 89 |
| Chief Executive Officer | 69 |
| Design Engineer | 67 |
| Maintenance Supervisor | 66 |
| Shipping and Receiving Supervisor | 66 |
| Facilities Administrative Assistant | 63 |
| Quality Assurance Supervisor | 60 |
| Vice President of Production | 52 |
| Tool Designer | 48 |
| Assistant to the Chief Financial Officer | 48 |
| Benefits Specialist | 45 |
| Master Scheduler | 42 |
| Vice President of Sales | 25 |
| Senior Design Engineer | 21 |
| Chief Financial Officer | 20 |
| Vice President of Engineering | 20 |

Source code:

mport pandas as pd, matplotlib.pyplot as plt  
csvtable1 = pd.read\_csv('question 4\\table3.csv')  
print(csvtable1)  
plt.barh( csvtable1['JobTitle'],csvtable1['SickLeaveHours'])  
plt.title('Relationship between JobTitle and SickLeaveHours', fontsize=12)  
plt.xlabel('JobTitle', fontsize=12)  
plt.ylabel('SickLeaveHours', fontsize=12)  
#plt.xticks(rotation=90)  
plt.show()

import pandas as pd, matplotlib.pyplot as plt  
csvtable2 = pd.read\_csv('question 4\\table4.csv')  
print(csvtable2)  
plt.barh(csvtable2['JobTitle'], csvtable2['SickLeaveHours'])  
plt.title('relationship between JobTitle and SickLeaveHours')  
plt.xlabel('JobTitle')  
plt.ylabel('SickLeaveHours')  
#plt.xticks(rotation=90)  
plt.show()

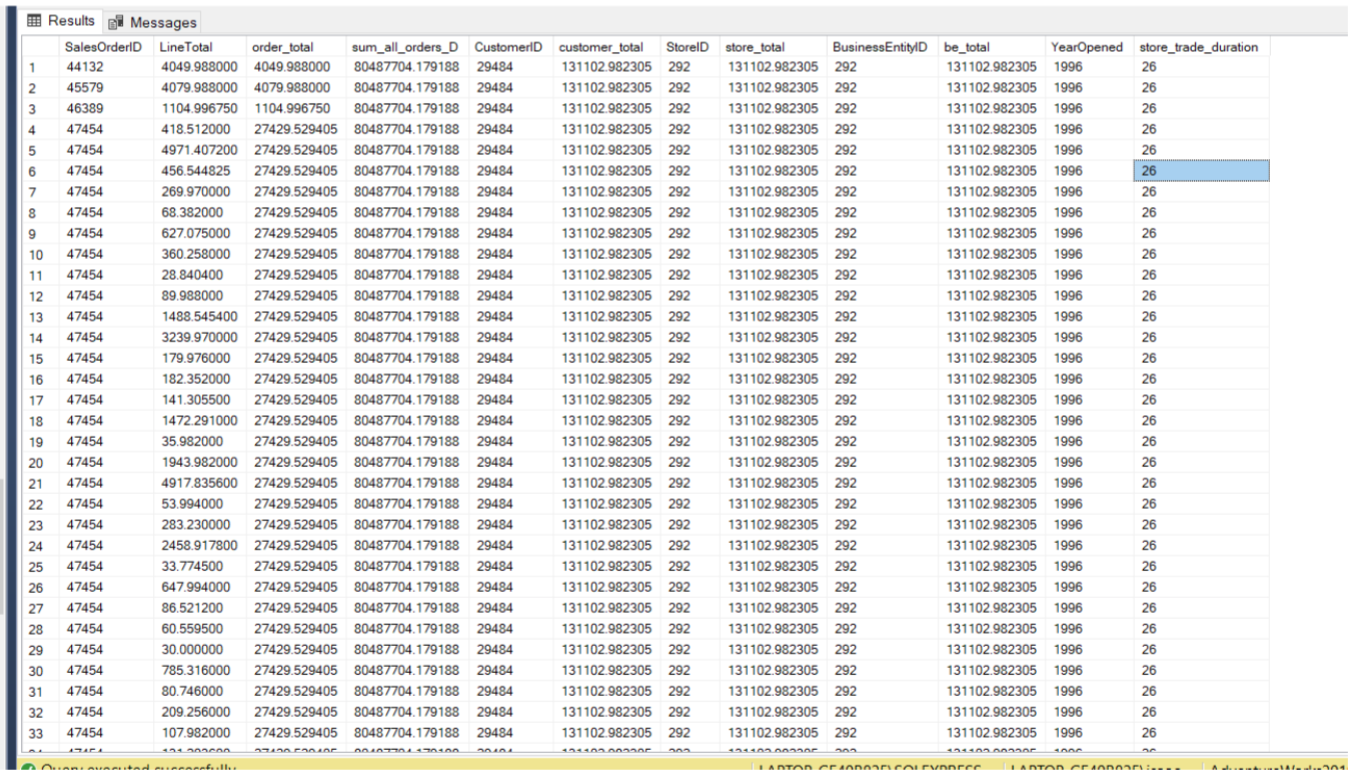




**5 - What is the relationship between store trading duration and revenue?**

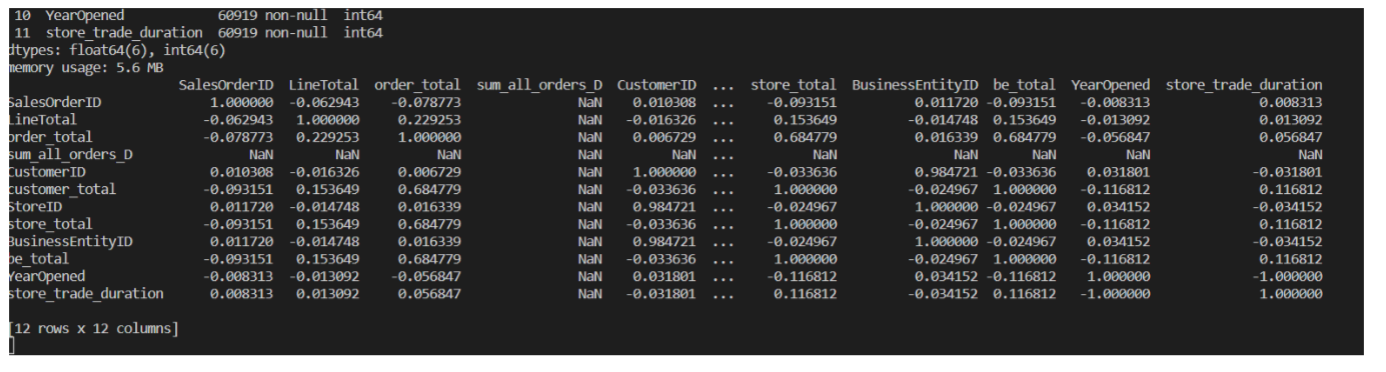
SELECT[AdventureWorks2019].[Sales].[SalesOrderDetail].[SalesOrderID]  
 , [LineTotal]  
 , SUM(LineTotal) OVER  
 (PARTITION BY [AdventureWorks2019].[Sales].  
[SalesOrderDetail].SalesOrderID )  
 AS order\_total   
 , SUM(LineTotal) OVER  
 ()  
 AS sum\_all\_orders\_D  
 ,[AdventureWorks2019].[Sales].[SalesOrderHeader].[CustomerID]  
 , SUM(LineTotal) OVER  
 (PARTITION BY [AdventureWorks2019].[Sales].[SalesOrderHeader].  
[CustomerID])  
 AS customer\_total  
 ,[AdventureWorks2019].[Sales].[Customer].[StoreID]  
 , SUM(LineTotal) OVER  
 (PARTITION BY [AdventureWorks2019].[Sales].[Customer].[StoreID])  
 AS store\_total  
 ,[AdventureWorks2019].[Sales].[vStoreWithDemographics].  
[BusinessEntityID]  
 , SUM(LineTotal) OVER  
 (PARTITION BY [AdventureWorks2019].[Sales].  
[vStoreWithDemographics].[BusinessEntityID])  
 AS be\_total

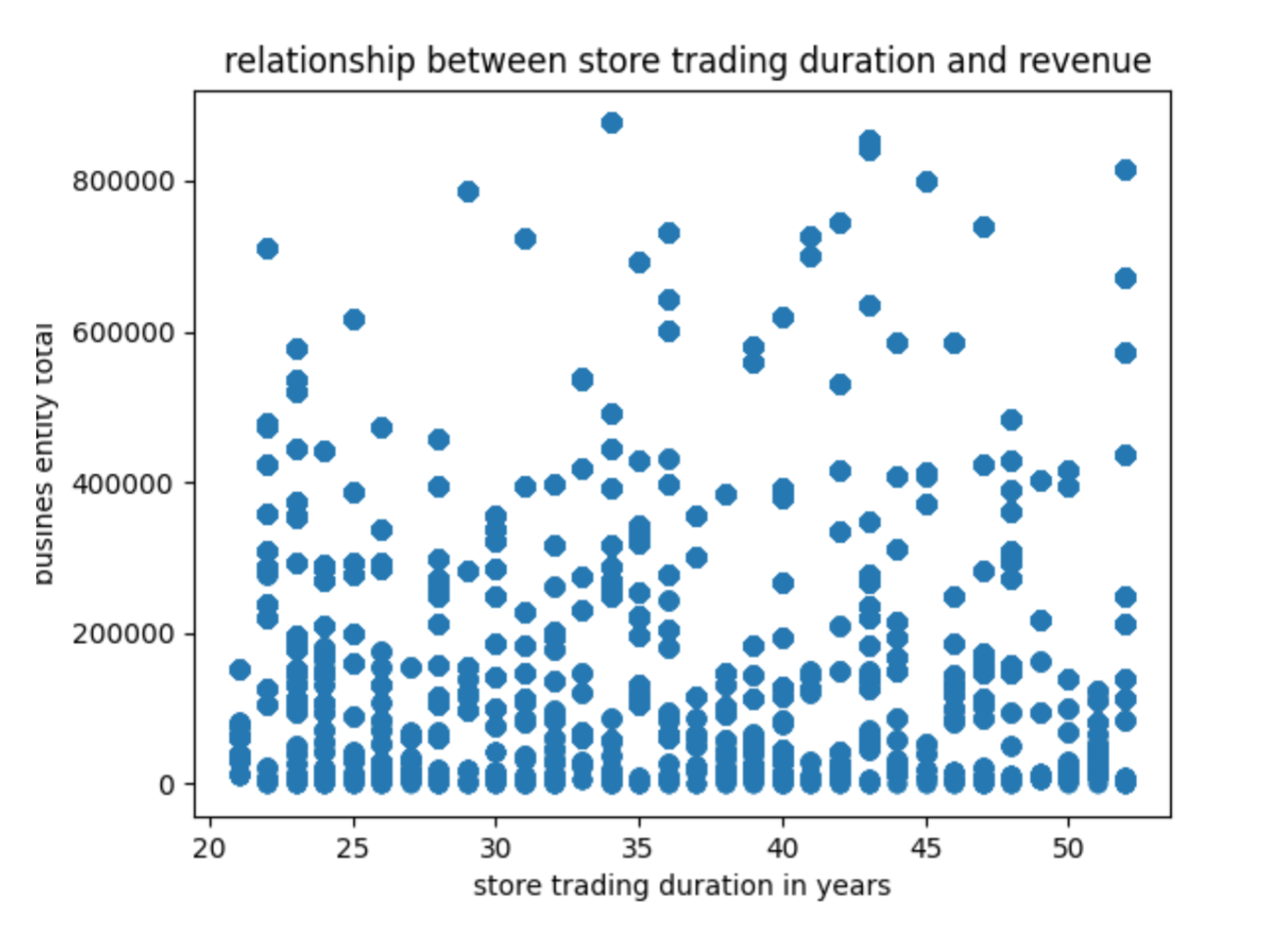
,[AdventureWorks2019].[Sales].[vStoreWithDemographics].  
[YearOpened]  
 ,(2022 - [AdventureWorks2019].[Sales].[vStoreWithDemographics].  
[YearOpened]) AS store\_trade\_duration  
FROM [AdventureWorks2019].[Sales].[SalesOrderDetail]  
LEFT JOIN [AdventureWorks2019].[Sales].[SalesOrderHeader]  
ON [AdventureWorks2019].[Sales].[SalesOrderDetail].[SalesOrderID] =  
 [AdventureWorks2019].[Sales].[SalesOrderHeader].[SalesOrderID]  
LEFT JOIN [AdventureWorks2019].[Sales].[Customer]  
ON [AdventureWorks2019].[Sales].[SalesOrderHeader].[CustomerID] =  
 [AdventureWorks2019].[Sales].[Customer].[CustomerID]  
LEFT JOIN [AdventureWorks2019].[Sales].[vStoreWithDemographics]  
ON [AdventureWorks2019].[Sales].[Customer].[StoreID] =  
 [AdventureWorks2019].[Sales].[vStoreWithDemographics].[BusinessEntityID]



import pandas aliased as pd and matlib plot aliased as plt  
import pandas as pd, matplotlib.pyplot as plt  
#import csv as pandas data frame named df1. replace "question 3\\  
relevent data4.csv" with the relevent path your csv file  
df1 = pd.read\_csv("question 5\POWANDb.csv")  
#display the info on the dataframe to check the import has been   
sucsessfull and that the data is in the corect format  
df1.info()

#calculate and print corralation between variables. may or may not   
aplly to your data set, if in doubt delete  
cdf1 = df1.corr(method='pearson')  
print(cdf1)  
#plot graph. replace "scatter" with plot type see matplotlib   
dcoumentation for details.  
# replace x = df1["store\_trading\_duration\_years"] with x = df1["your   
colum name"]  
# replace y = (df1["av\_an\_rev"]/100000) with y = df1["your other colum   
name"]  
plt.scatter(x = (df1["store\_trade\_duration"]),y = df1["be\_total"])  
plt.title(" relationship between store trading duration and revenue")  
plt.xlabel("store trading duration in years")  
plt.ylabel("busines entity total")  
#show graph  
plt.show()





It is still difficult to determine a relationship from looking at the scatter plot, but the   
correlation figure quantifies the relationship.  
The value of 0.116812 suggests a very weak positive correlation between how many a store   
has been open for and its revenue.

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

To answer this question the join done in previous question 5 we used with the small alterations to include store size and Employee Number.

The result was stored in a view.

CREATE VIEW Revenues\_StoreSize\_EmployeeNumber AS

In order to analyze the data the stores were grouped by four types according to their size as follows:

|  |  |  |
| --- | --- | --- |
| Store Type | Size ft2 | Employee Number |
| Small | < 15000 | <15 |
| Medium | 15000 – 30000 | 15-30 |
| Large | 30000 – 50000 | 30-60 |
| Mega | 30000 – 50000 | >60 |

SELECT \*

  FROM [AdventureWorks2019].[dbo].[Revenues\_StoreSize\_EmployeeNumber]

SELECT storetype = 'Small', SUM(q1.TotStoRev) AS TotalRevenues, AVG(q1.TotStoRev) AS AverageRevenues, AVG(q1.StoreSize) AS AverageStoreSize FROM

(SELECT sum(Revenues) AS TotStoRev, StoreSize

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE StoreSize < 15000

GROUP BY StoreSize) q1

UNION

SELECT storetype = 'Medium', SUM(q2.TotStoRev) AS TotalRevenues, AVG(q2.TotStoRev) AS AverageRevenues, AVG(q2.StoreSize) AS AverageStoreSize FROM

(SELECT SUM(Revenues) AS TotStoRev, StoreSize

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE StoreSize > 15000 AND StoreSize < 30000

GROUP BY StoreSize) q2

UNION

SELECT storetype = 'Large', SUM(q3.TotStoRev) AS TotalRevenues, AVG(q3.TotStoRev) AS AverageRevenues, AVG(q3.StoreSize) AS AverageStoreSize FROM

(SELECT SUM(Revenues) AS TotStoRev, StoreSize

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE StoreSize > 30000 AND StoreSize < 50000

GROUP BY StoreSize) q3

UNION

SELECT storetype = 'Mega', SUM(q4.TotStoRev) AS TotalRevenues, AVG(q4.TotStoRev) AS AverageRevenues, AVG(q4.StoreSize) AS AverageStoreSize FROM

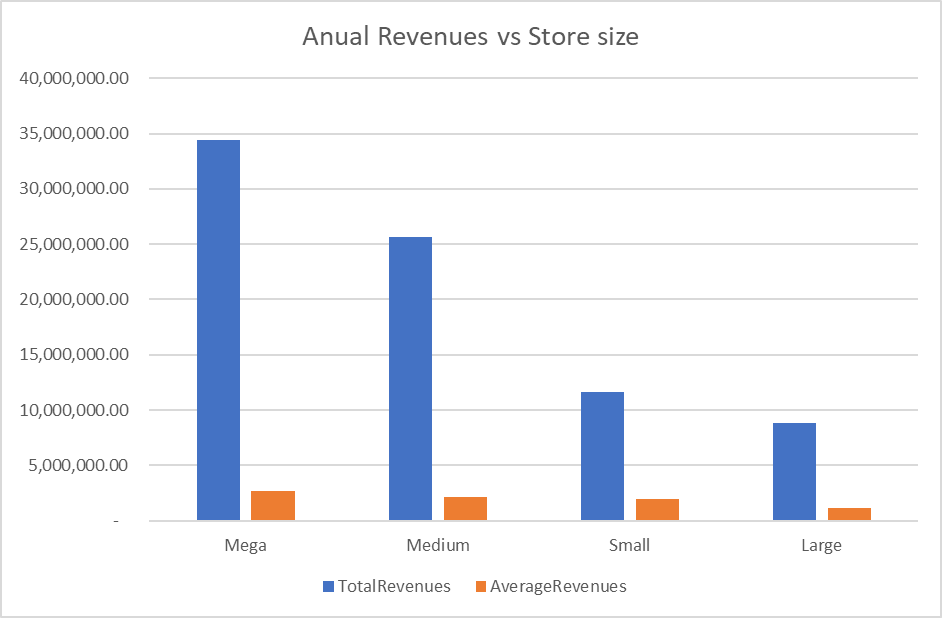
(SELECT SUM(Revenues) AS TotStoRev, StoreSize

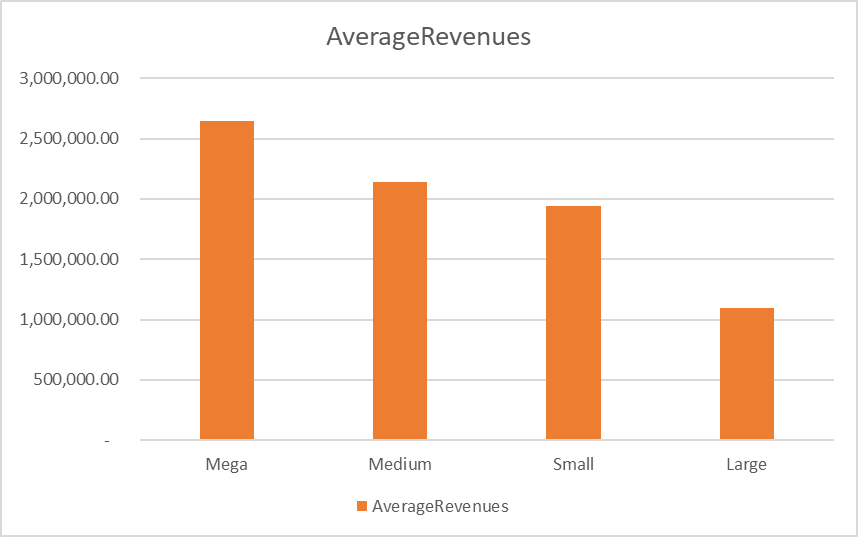
FROM Revenues\_StoreSize\_EmployeeNumber

WHERE StoreSize > 50000

GROUP BY StoreSize)  q4

|  |  |  |  |
| --- | --- | --- | --- |
| Store Type | Total Revenues | Average Revenues | Average StoreSize ft2 |
| Large | £8,794,527.38 | £1,099,315.92 | 38,500 |
| Medium | £25,645,142.03 | £2,137,095.17 | 22,500 |
| Mega | £34,410,720.78 | £2,646,978.52 | 74,000 |
| Small | £11,637,313.99 | £1,939,552.33 | 8,500 |





SELECT \*

  FROM [AdventureWorks2019].[dbo].[Revenues\_StoreSize\_EmployeeNumber]

SELECT storetype = 'Small', SUM(e1.TotStoRev) AS TotalRevenues, AVG(e1.TotStoRev) AS AverageRevenues, AVG(e1.Employee) AS AverageStaff FROM

(SELECT SUM(Revenues) AS TotStoRev, Employee

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE Employee < 15

GROUP BY Employee) e1

UNION

SELECT storetype = 'Medium', SUM(e2.TotStoRev) AS TotalRevenues, AVG(e2.TotStoRev) AS AverageRevenues, AVG(e2.Employee) AS AverageStaff FROM

(SELECT SUM(Revenues) AS TotStoRev, Employee

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE Employee > 15 AND Employee < 30

GROUP BY Employee) e2

UNION

SELECT storetype = 'Large', SUM(e3.TotStoRev) AS TotalRevenues, AVG(e3.TotStoRev) AS AverageRevenues, AVG(e3.Employee) AS AverageStaff FROM

(SELECT SUM(Revenues) AS TotStoRev, Employee

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE Employee > 30 AND Employee < 60

GROUP BY Employee) e3

UNION

SELECT storetype = 'Mega', SUM(e4.TotStoRev) AS TotalRevenues, AVG(e4.TotStoRev) AS AverageRevenues, AVG(e4.Employee) AS AverageStaff FROM

(SELECT SUM(Revenues) AS TotStoRev, Employee

FROM Revenues\_StoreSize\_EmployeeNumber

WHERE Employee > 60

GROUP BY Employee) e4

|  |  |  |  |
| --- | --- | --- | --- |
| Store Type | Total Revenues | Average Revenues | Average Staff |
| Large | £12,566,690.43 | £571,213.20 | 42 |
| Medium | £13,771,741.16 | £1,377,174.12 | 22 |
| Mega | £30,487,363.03 | £1,051,288.38 | 84 |
| Small | £22,442,209.45 | £1,726,323.80 | 8 |

