Adventure Works Data Analysis
(Group Project)

Presented by Team 3:
Shubhangi T
Nang Mo K
Miles M

List of index:

Conceptual Model

Before designing the schema, the questions were divided into three main categories. First, we grouped Human Resources data (employees, bonuses, job titles, and annual leave). Second, we analysed Sales data (sales figures, territories, and regions) using the Sales table. Finally, we examined store size, workforce demographics, and revenue using the Sales Data view by demographic. The specific schemas utilized are listed below.

Libraries Used in the Project

SQL: Used as the primary tool for querying and extracting the necessary tables and variables for analysis.

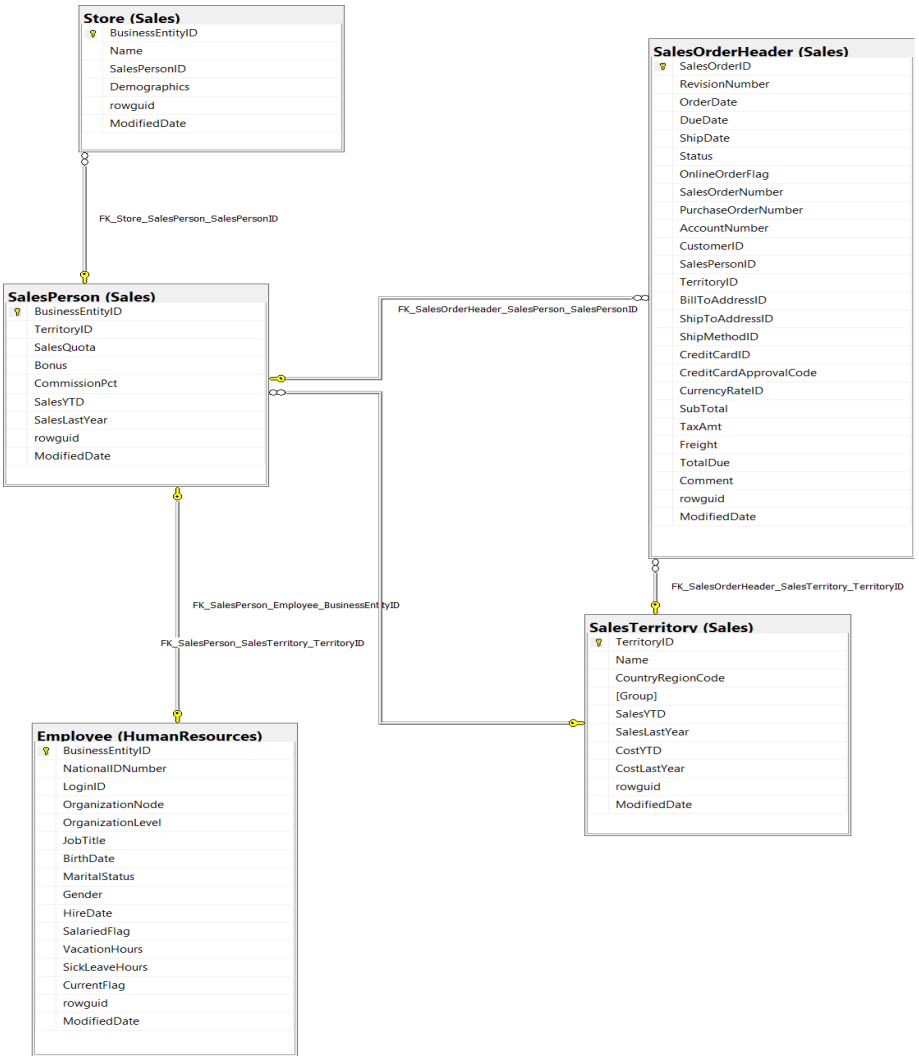
Python (Jupyter Notebook): Functioned as the main environment for scripting and data manipulation tasks, including visualisations

Pandas & NumPy: Pandas was utilized for data handling and preprocessing, and NumPy is used to analyse large datasets.

Data Visualisation: Matplotlib, Plotly, and Seaborn were employed to generate different charts.

PyODBC: Facilitated a secure and reliable connection between the Python environment and the SQL database.

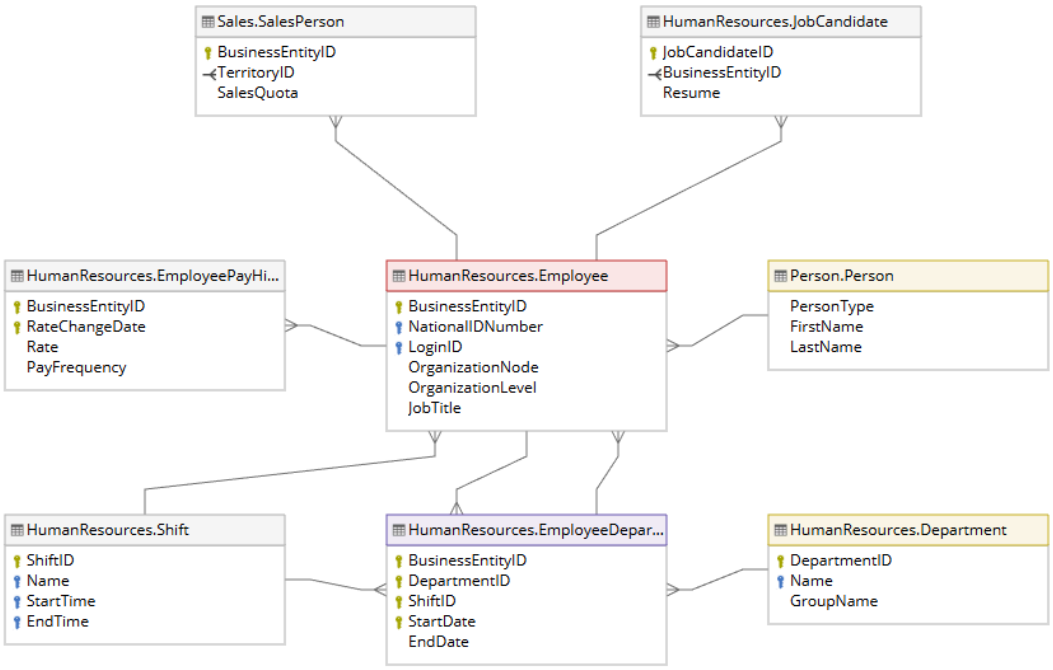
Database Schema



Revenue and Territory Analysis:

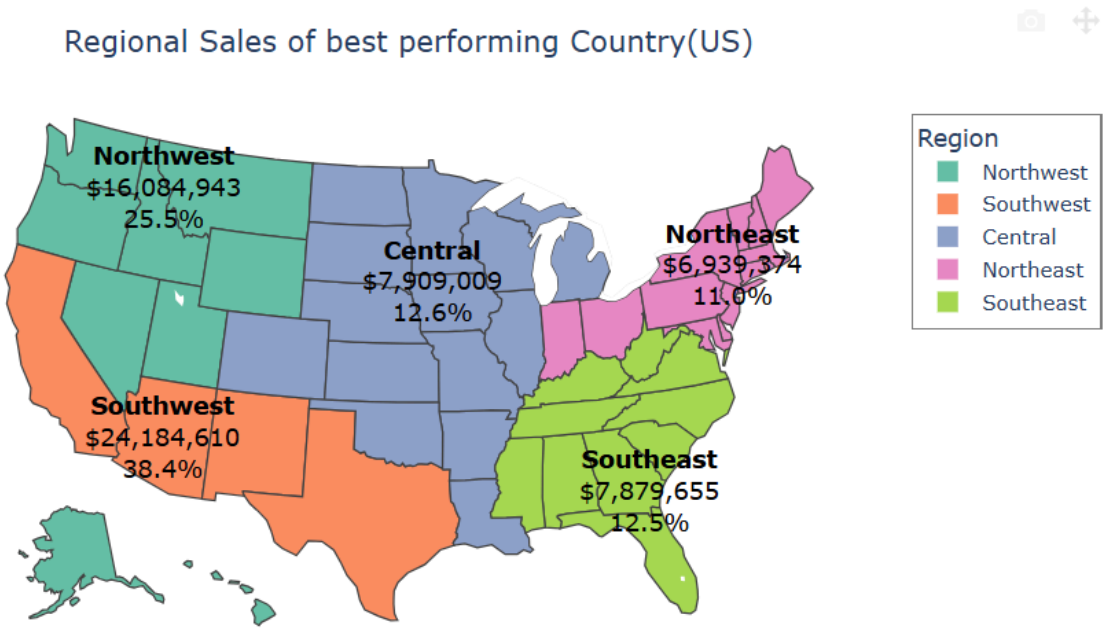
The schema above was used to identify relevant data for revenue and geographical data. By joining sales headers with territory identifiers, global revenue was aggregated to isolate the highest-performing country, and later a smaller scope on the highest-performing region.

Human Resources



Employees and Bonus:
To analyse the relationship between employees, a person table was then referenced to job titles and bonuses.

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



Insight:

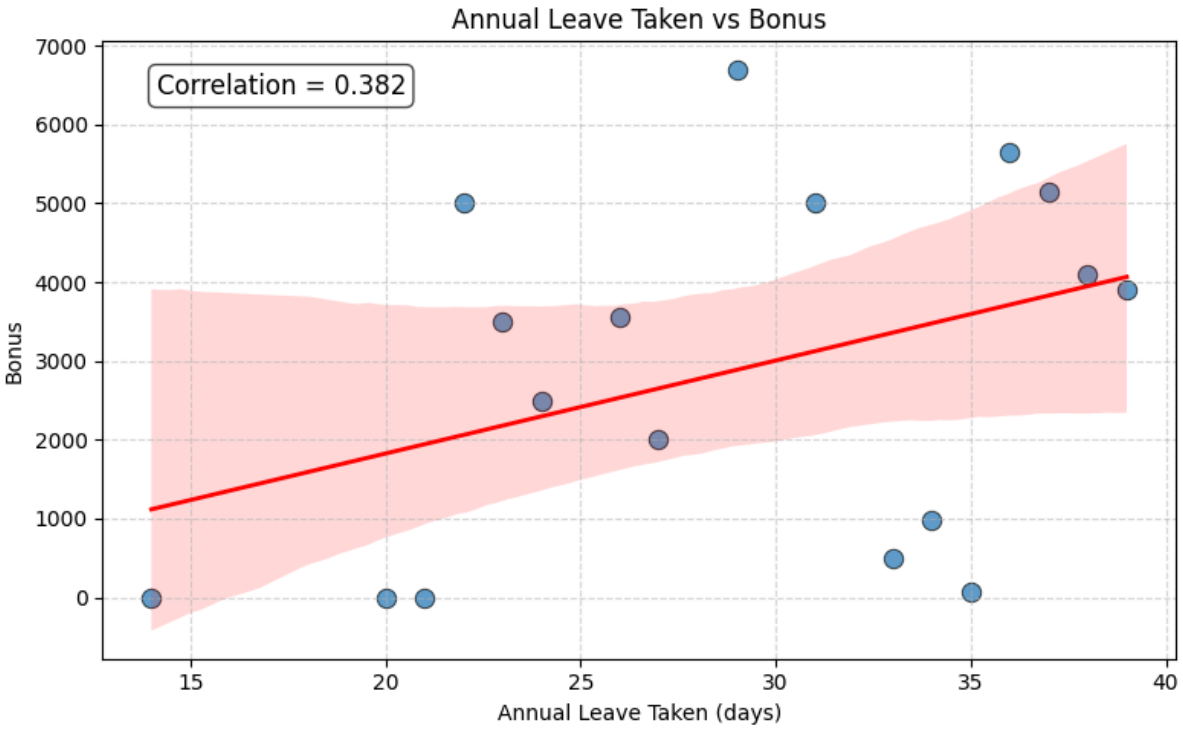
The Southwest region is the market leader, generating over $ 24.184 million in sales and contributing 38.4% of the country's total sales. This performance demonstrates a highly effective business operation that successfully meets the demand in this area.

The Northwest region holds the position of the second-highest market with $16,084,943 in sales, contributing 25.5% of the total. When combined with the Southwest, these two regions account for over 60% of the total sales, clearly indicating that the Western part of the country is the primary market.

The rest of the regions:

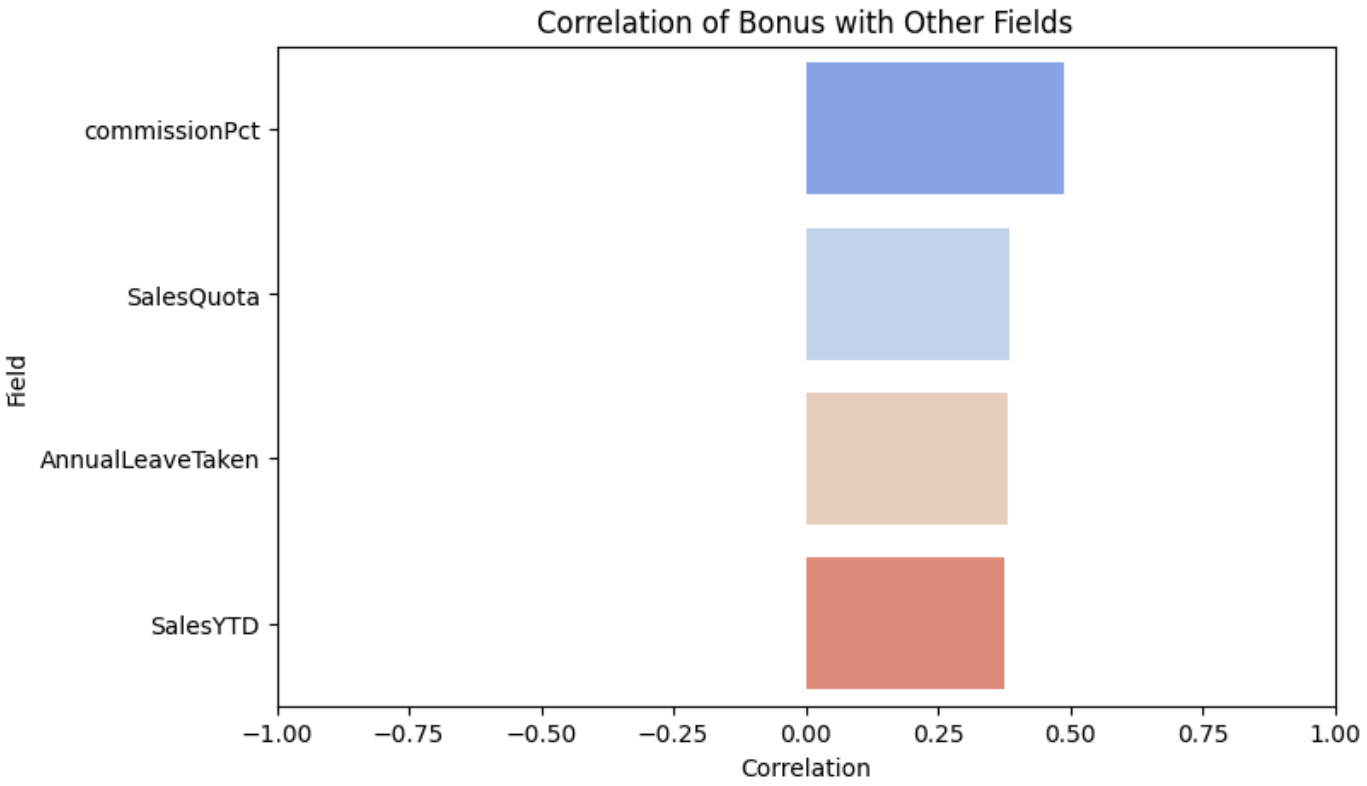
Northeast: $6,939,374 (11.0%)
Southeast: $7,875,655 (12.5%)
Central: $7,909,009 (12.6%)

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



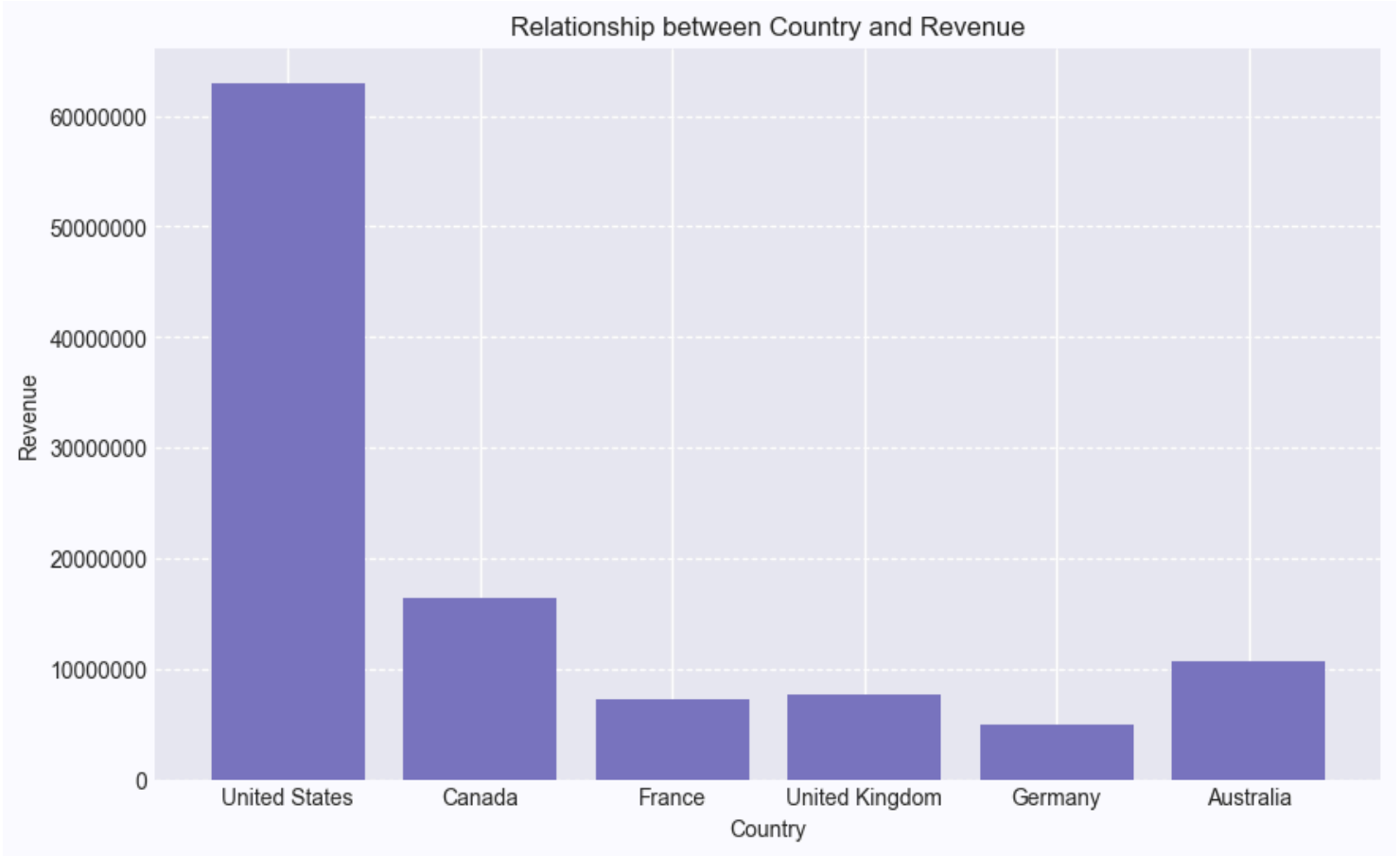
The correlation coefficient of 0.382 indicates a positive relationship between the two variables. However, a correlation coefficient of 0.3 is considered weak. This means the relationship is not strong enough to establish a consistent pattern.

Therefore, the amount of annual leave taken is not a strong indicator for predicting the bonus amount. Other factors, such as meeting the Sales Quota or the commission, likely have a much stronger correlation with the bonus.

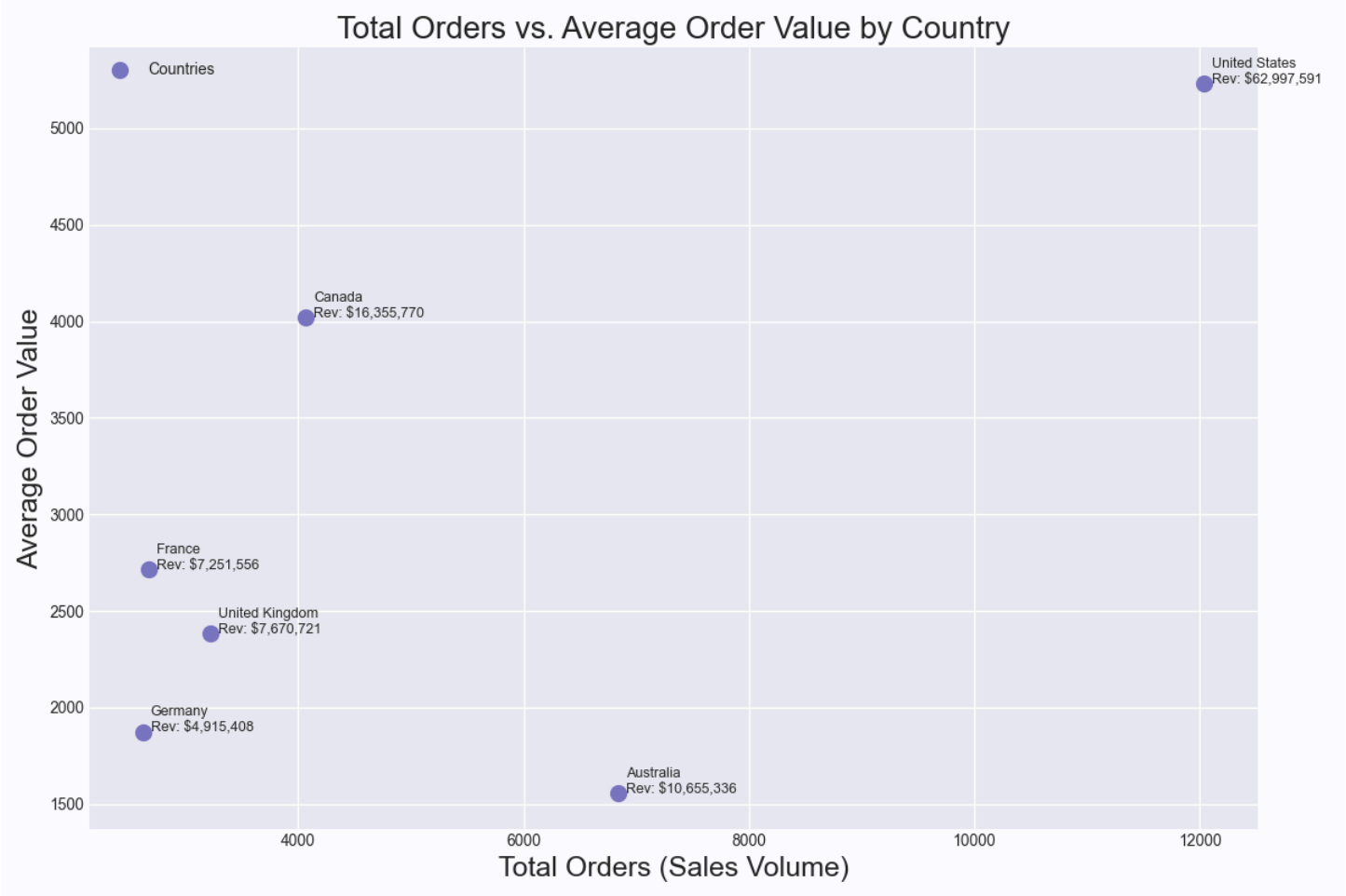


Bonuses are mostly linked to pay-related factors, especially commission percentage and sales quota, rather than leave. This means the bonus system mainly rewards senior roles and sales responsibility, with some influence from actual sales performance (SalesYTD).

3. What is the relationship between Country and Revenue?



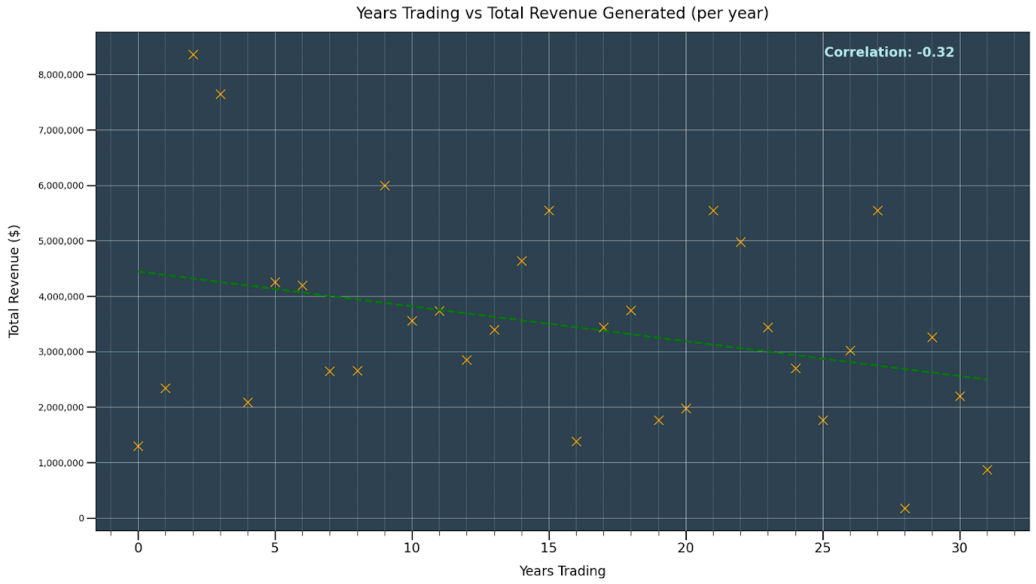
The United States is the company's main market, generating over $60 million in revenue, followed by Canada and Australia, respectively. The European market has been the weakest, with Germany having less than 5 million in sales. No definitive relationship can be drawn from this data, aside from understanding the primary market demographics. Hence, the further analysis to understand

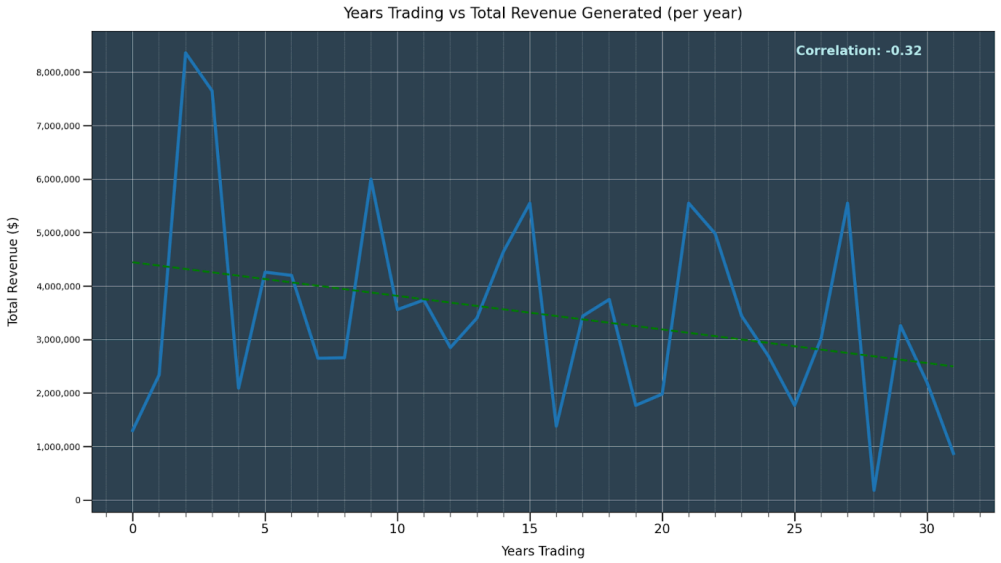


The second graph shows that Australia, despite being the third-highest revenue generator with over $10 million in sales, has the lowest average order value. Australia breaks this general trend, with high sales volume but a low average order value, indicating customers are buying products that are of less value, such as accessories over bikes. Considering the size of the land in Australia and being far away from America, further investment can be made to compete in the local market, to increase individual sales order value. Or perhaps a bundle sale on accessories? Are there not enough distributors in Australia in comparison to Canada?

4. What is the relationship between store trading duration and revenue?

The following chart shows every store as a data point, with total revenue per year as the y-axis and years trading as the x-axis. Correlation is shown at the top right.
Weak/no correlation. Trendline shows steady sales with a slight dip over time.

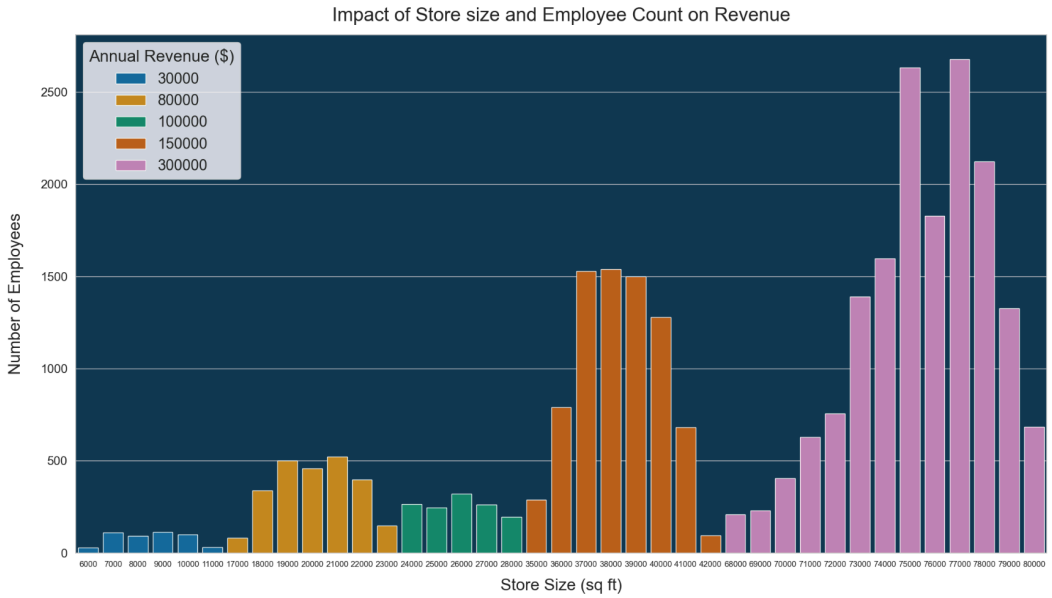




As you can see on this chart, we found that stores generally fall into the same range of revenue regardless of years of trading, with a very slight downward trend. We would ask you not to put much weight behind this trend, as the correlation is very weak.
The key takeaway from this visualisation is the sharp drop in revenue at year 4. This is unusually steep, and there could be some underlying issues with business strategy or management. This could also indicatea better business strategy for newer stores, but with the data provided, it's

hard to tell either way.
The data we were provided didn't give a clear picture of yearly revenue by store over time. Instead, we had to take the total revenue to date and work from that.
Something we would highly recommend is that you log individual store revenue year by year; it could produce a clearer picture of this relationship and give you the opportunity to see store-by-store breakdowns over time.

5. What is the relationship between the size of the stores, the number of employees and revenue?



This chart shows the number of employees on the y-axis, with store size increasing across the x-axis. Revenue is displayed by bar colour, with 4 groupings indicating very small, small, medium, and large store sizes. Larger stores generate more revenue. Some of the larger stores have fewer employees than the medium-sized store, but pull in more revenue. The top 5 stores by employee count are in the highest revenue category.

Store sizes fall into 5 categories of revenue, with the largest stores by square footage outperforming all other stores. Although employee count doesn't factor into revenue as much as store size, profits could be increased by downsizing employee count in these very large stores to be more in line with the beginning and end of their distribution curves.
As you can see, even though they have the 2nd highest combined employee count, large stores with a size between 3500 and 42000(orange) sq ft are outperforming the medium-sized stores (green) by 50% per store. Considering the number of employees is almost 6 (5.97) times more for the 2nd largest category of stores, we found that medium stores are much more profitable. We would suggest future stores fall within this range to maximise profits.

APPENDIX

Question1: What are the regional sales in the best-performing country?

SQL:

```
ALTER PROCEDURE usp_GetCountrySales

AS
BEGIN
WITH Country AS (
   SELECT TOP 1
      st.CountryRegionCode
   FROM Sales.SalesOrderHeader soh
   JOIN Sales.SalesTerritory st
      ON soh.TerritoryID = st.TerritoryID
   GROUP BY st.CountryRegionCode
   ORDER BY SUM(soh.SubTotal) DESC
)
SELECT
   st.Name AS Region,
  ROUND(SUM(soh.SubTotal), 2)  AS RegionalSales
FROM Sales.SalesOrderHeader soh
JOIN Sales.SalesTerritory st
   ON soh.TerritoryID = st.TerritoryID
WHERE st.CountryRegionCode = (SELECT CountryRegionCode FROM Country)
GROUP BY st.Name
ORDER BY RegionalSales DESC;
END
```

Python Code

```
import pyodbc
import pandas as pd
import plotly.express as px

# Connect to SQL Server ---
conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=.;'
    'DATABASE=AdventureWorks2019;'
    'Trusted_Connection=yes;'
)

#  Call the stored procedure ---
df = pd.read_sql_query("EXEC usp_GetCountrySales", conn)
```

```
#  Inspect DataFrame ---
print(df.head())


region_states = {
    "Northwest": ['AK', 'HI','ID','MT','NV','OR','UT','WA','WY'],
    "Southwest": ['AZ', 'CA', 'GU', 'NM', 'TX' ],
    "Central":   [ 'AR','CO','IA','IL','KS','LA','MI','MN', 'MO', 'ND',
'NE', 'OK', 'SD', 'WI'],
    "Northeast": ['CT',
'DC','DE','IN','MA','MD','ME','NH','NJ','NY','OH','PA', 'RI','VT'],
    "Southeast": ['AL', 'FL', 'GA', 'KY', 'MS', 'NC', 'PR', 'SC', 'TN',
'VA', 'WV']
}


#  Expand DataFrame to State Level ---
rows = []
for region, states in region_states.items():
    sales = df.loc[df['Region'] == region, 'RegionalSales'].values[0]
    for state in states:
        rows.append([state, region, sales])

df_states = pd.DataFrame(rows, columns=['State', 'Region',
'RegionalSales'])

# Plot Choropleth ---
fig = px.choropleth(
    df_states,
    locations='State',
    locationmode='USA-states',
    color='Region',                # fill color by region
    scope='usa',
    hover_name='Region',
    hover_data={'RegionalSales': True, 'State': False},
    color_discrete_sequence=px.colors.qualitative.Set2
)

# Add RegionalSales Labels at Region Centroids ---
region_coords = {
    "Northwest": [46.5, -121],
    "Southwest": [36, -112],
    "Central":   [43, -88],
```

```
    "Northeast": [42, -74],
    "Southeast": [33.5, -82]
}

for _, row in df.iterrows():
    lat, lon = region_coords[row['Region']]
    fig.add_trace(px.scatter_geo(
        lat=[lat], lon=[lon],
        text=[f"${row['RegionalSales']:,}"]
    ).data[0])

#  Layout ---
fig.update_layout(
    title_text='USA Regional Sales by Region',
    title_x=0.5,
    geo=dict(showland=True, landcolor='lightgray')
)

fig.show()
# Close connection ---
conn.close()
```

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

SQL CODE:

```
CREATE  PROCEDURE [dbo].[usp_BonusDetails]

AS
BEGIN
SELECT
    e.VacationHours AS AnnualLeaveTaken,
    sp.Bonus,
    sp.commissionPct,
    ROUND(sp.SalesYTD,2) AS SalesYTD,
    ROUND(sp.SalesQuota,2) AS SalesQuota
FROM HumanResources.Employee e
JOIN Sales.SalesPerson sp
    ON e.BusinessEntityID = sp.BusinessEntityID

END
```

Python code

```
import pyodbc
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# Connect to SQL Server ---
conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=.;'
    'DATABASE=AdventureWorks2019;'
    'Trusted_Connection=yes;'
)

#  Call the stored procedure ---
df = pd.read_sql_query("EXEC usp_BonusDetails", conn)

#  Inspect DataFrame ---
print(df.head())
```

```
# Correlation
corr = df['AnnualLeaveTaken'].corr(df['Bonus'])

# Plot
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='AnnualLeaveTaken', y='Bonus', s=80, alpha=0.7,
edgecolor='black')
sns.regplot(data=df, x='AnnualLeaveTaken', y='Bonus', scatter=False,
color='red', line_kws={'linewidth': 2})

# Annotate
plt.text(0.05, 0.95, f"Correlation = {corr:.3f}",
transform=plt.gca().transAxes,
         fontsize=12, verticalalignment='top',
         bbox=dict(boxstyle="round,pad=0.3", facecolor="white",
alpha=0.7))

plt.title("Annual Leave Taken vs Bonus")
plt.xlabel("Annual Leave Taken (days)")
plt.ylabel("Bonus")
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()

# Compute correlations with Bonus
corr = df.corr(numeric_only=True)["Bonus"].sort_values(ascending=False)

# Remove Bonus itself from the list
corr = corr.drop("Bonus")

plt.figure(figsize=(8,5))
sns.barplot(x=corr.values, y=corr.index, palette="coolwarm")
plt.title("Correlation of Bonus with Other Fields")
plt.xlabel("Correlation")
plt.ylabel("Field")
plt.xlim(-1, 1)
plt.show()
```

Question 3: What is the relationship between Country and Revenue?

SQL CODE:

```
SELECT
    PCR.Name AS Country,
    CAST(SUM(SOH.SubTotal) AS DECIMAL(10,2)) AS Revenue,
    COUNT(SOH.SalesOrderID) AS TotalOrdersCount,
    SUM(SOH.SubTotal) / COUNT(SOH.SalesOrderID) AS AverageOrderValue
FROM
    Sales.SalesOrderHeader AS SOH
JOIN
    Sales.SalesTerritory AS ST
    ON SOH.TerritoryID = ST.TerritoryID
JOIN
    Person.CountryRegion AS PCR
    ON ST.CountryRegionCode = PCR.CountryRegionCode
GROUP BY
    PCR.Name
ORDER BY AverageOrderValue DESC;
```

SQL OUTPUT:
CSV FORMAT:

```
Country, Revenue, TotalOrdersCount, AverageOrderValue
United States,62997590.71,12041,5231.9234
Canada,16355770.46,4067,4021.5811
France,7251555.65,2672,2713.9055
United Kingdom,7670721.04,3219,2382.9515
Germany,4915407.60,2623,1873.964
Australia,10655335.96,6843,1557.1147
```

SCREENSHOT:

Results | Messages

	Country	Revenue	TotalOrders	AverageOrderValue
1	United States	62997590.71	12041	5231.9234
2	Canada	16355770.46	4067	4021.5811
3	France	7251555.65	2672	2713.9055
4	United Kingdom	7670721.04	3219	2382.9515
5	Germany	4915407.60	2623	1873.964
6	Australia	10655335.96	6843	1557.1147

PYTHON CODE:

```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('https://github.com/dhaitz/matplotlib-stylesheets/raw/master
/pitayasmoothie-light.mplstyle')
```

```
cv = pd.read_csv('CV.csv')
print(cv.info())
```

FEW CHARTS:
RELATIONSHIP BETWEEN COUNTRY AND REVENUE

```
plt.figure(figsize=(10, 6))
plt.bar(q3['Country'], q3['AverageOrderValue'])
plt.title('Relationship between Country and AverageOrderValue')
plt.xlabel('Country')
plt.ylabel('Revenue')
ax = plt.gca()
ax.ticklabel_format(style='plain', axis='y')
plt.show()
```

SECOND CHART: COUNTRY AND TOTAL ORDERS

```
#Create a figure and a scatter plot
corr = q3['TotalOrdersCount'].corr(q3['AverageOrderValue'])
plt.figure(figsize=(12, 8))
plt.scatter(q3['TotalOrdersCount'], q3['AverageOrderValue'], s=100)

#iterate each row for the labels and revenue
for i, row in q3.iterrows():
    label = f"{row['Country']}\nRev: ${row['Revenue']:,.0f}"

    plt.annotate(label,
                 (row['TotalOrdersCount'], row['AverageOrderValue']),
                 xytext=(5,0),
                 textcoords='offset points',
                 fontsize=16)

plt.title('Total Orders vs. Average Order Value by Country',  fontsize=20)
plt.xlabel('Total Orders (Sales Volume)', fontsize=18)
plt.ylabel('Average Order Value', fontsize=18)
plt.grid(True)
plt.legend()
```

```
plt.tight_layout()
plt.savefig('scatter_revenue_labels.png')
```

Question 4: The relationship between store trading duration and revenue.

SQL
******

```
SELECT  YearOpened, SUM(AnnualRevenue) AS total_revenue_for_year, (2001 - YearOpened)
AS years_trading
 FROM Sales.vStoreWithDemographics
 GROUP BY  YearOpened
 ORDER BY years_trading ASC
```

Takes the Year Opened and the sum of annual revenue for that year from the vStorewithDemographics view, alongside creating a new column showing total years opened by subtracting the year opened from the year 2001 (last year in the db). Groups results by year and sorts by the new "years_trading" column in ascending order.

```
SELECT
    BusinessEntityID,
    YearOpened,
    AnnualRevenue,
    SUM(AnnualRevenue) OVER (PARTITION BY YearOpened) AS total_revenue_for_year,
    (2001 - YearOpened) AS years_trading
FROM Sales.vStoreWithDemographics
ORDER BY total_revenue_for_year DESC;
```

Shows BusinessEntityID, annual revenue total and the year opened. Used to identify the most profitable years of trading.

Python
****

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import MultipleLocator

#csv filepath
data_path = "C:\\Users\\M\\Documents\\.Data Analytics\\Week 8\\Day
1\\Interim Project\\ipvis\\rby.csv"
```

```
#import csv as pandas dataframe
rby = pd.read_csv(data_path)
#test print
print(rby)
#divider for visual clarity
print("**************************")
#print correlation values
correlation = rby.corr()
print(correlation)

#correlation value as variable for label
relevant_corr = correlation.iloc[1,2]

#create scatterplot / line plot
fig, ax = plt.subplots()
ax.set_facecolor("#2E4352")
#plt.scatter(rby['years_trading'],
rby['total_revenue_for_year'],marker='x',s=200,color="orange")    #comment
out for line plot
plt.plot(rby['years_trading'], rby['total_revenue_for_year'],linewidth=5)
#comment out for scatter plot
plt.title("Years Trading vs Total Revenue Generated (per
year)",fontsize=24, pad=20)

#labels size and padding
plt.xlabel('Years Trading',fontsize=20,labelpad=15)
plt.ylabel('Total Revenue ($)',fontsize=20,labelpad=25)
#remove scientific notation
plt.ticklabel_format( axis='both', style='plain')
#tick label size
ax.set_xticklabels(ax.get_xticklabels(), fontsize=20)
ax.set_yticklabels(ax.get_yticklabels(), fontsize=14)
#tick mark size
plt.tick_params(axis='x', length=14, width=2)
plt.tick_params(axis='y', length=14, width=2)
plt.tick_params(which='minor', axis='x', length=7, width=1)
#add commas as thousands separators
current_values = plt.gca().get_yticks()
plt.gca().set_yticklabels(['{:,.0f}'.format(x) for x in current_values])
#add minor ticks to better indicate years
plt.gca().xaxis.set_minor_locator(MultipleLocator(1))
#set grid lines
plt.grid(which='major', color='white', linestyle='-', linewidth=0.5)
```

```
plt.grid(which='minor', color='white', linestyle='--', linewidth=0.3)

#create scatterpolt trendline
trendline = np.poly1d(
    np.polyfit(rby['years_trading'], rby['total_revenue_for_year'], 1)
)

# Plot the trendline
plt.plot(
    rby['years_trading'],
    trendline(rby['years_trading']),
    linestyle='--',
    linewidth=3,
    color='green'
)


#dynamically display correlation value on scatter
plt.text(
    0.78, 0.95,
     f"Correlation: {relevant_corr:.2f}", #fstring required
    color="#BAEEF0",
    fontweight='bold',
    fontsize=20,
    transform=ax.transAxes
)


plt.show()
```

	YearOpened	total_revenue_for_year	years_trading
YearOpened	1.000000	0.316066	-1.000000
total_revenue_for_year	0.316066	1.000000	-0.316066
years_trading	-1.000000	-0.316066	1.000000

Terminal output showing correlation values.

Question 5:

SQL
*****
SELECT *

```
FROM Sales.vStoreWithDemographics
```

Takes all data from vStoreWithDemographics view.

```
SELECT SUM(total_employees) AS final_count
FROM (
   SELECT SUM(NumberEmployees) as total_employees, SquareFeet
   FROM Sales.vStoreWithDemographics
   WHERE SquareFeet BETWEEN 6000 AND 11000
   GROUP BY SquareFeet) as x

SELECT SUM(total_employees) AS final_count
FROM (
   SELECT SUM(NumberEmployees) as total_employees, SquareFeet
   FROM Sales.vStoreWithDemographics
   WHERE SquareFeet BETWEEN 1700 AND 23000
   GROUP BY SquareFeet) as x

SELECT SUM(total_employees) AS final_count
FROM (
   SELECT SUM(NumberEmployees) as total_employees, SquareFeet
   FROM Sales.vStoreWithDemographics
   WHERE SquareFeet BETWEEN 24000 AND 28000
   GROUP BY SquareFeet) as x


SELECT SUM(total_employees) AS final_count
FROM (
   SELECT SUM(NumberEmployees) as total_employees, SquareFeet
   FROM Sales.vStoreWithDemographics
   WHERE SquareFeet BETWEEN 35000 AND 42000
   GROUP BY SquareFeet) as x

SELECT SUM(total_employees) AS final_count
FROM (
   SELECT SUM(NumberEmployees) as total_employees, SquareFeet
   FROM Sales.vStoreWithDemographics
   WHERE SquareFeet BETWEEN  68000 AND 80000
   GROUP BY SquareFeet) as x
```

Counts total employees by store size.

Python
****

```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


#csv filepath
data_path = "C:\\Users\\M\\Documents\\.Data Analytics\\Week 8\\Day
1\\Interim Project\\ipvis\\vStoreWithDemographics.csv"

vswd = pd.read_csv(data_path)

print(vswd)

sns.set_theme(style = 'whitegrid')

# #group revenue by quartile
# vswd['Revenue'] = pd.qcut(
#     vswd['AnnualRevenue'],
#     q=4,
#     labels=['Low', 'Medium', 'High'],
#     duplicates='drop'
# )

#sum total employees per store size (square feet)
totalemp =
vswd.groupby(['SquareFeet','AnnualRevenue'])['NumberEmployees'].sum().rese
t_index()

#sort store sizes by total employees (descending)
size_total = totalemp.groupby('SquareFeet')['NumberEmployees'].sum()

#convert index to list
size_total_list = size_total.index.tolist()

#SquareFeet to ordered category variable
totalemp['SquareFeet'] = pd.Categorical(totalemp['SquareFeet'],
categories=size_total_list, ordered=True)

#remove stores with no revenue data, testing only. Don't use this.
#totalemp['SquareFeet'] =
totalemp['SquareFeet'].cat.remove_unused_categories()
```

```
#set colour palette
sns.set_palette("mako")

fig, ax = plt.subplots()
ax.set_facecolor("#113953")
bplot =sns.barplot(
    data=totalemp,
    x="SquareFeet",
    y="NumberEmployees",
    hue="AnnualRevenue",    #colour by revenue
    width=0.8,
    palette="colorblind",
    ci=None
)


#labels size and padding
plt.xlabel('Store Size (sq ft)',fontsize=24,labelpad=15)
plt.ylabel('Number of Employees',fontsize=24,labelpad=25)
plt.title("Impact of Store size and Employee Count on
Revenue",fontsize=28, pad=20)
plt.legend(title='Annual Revenue ($)',title_fontsize=24, loc='upper left',
fontsize=22)
#tick label size
ax.set_xticklabels(ax.get_xticklabels(), fontsize=12)
ax.set_yticklabels(ax.get_yticklabels(), fontsize=18)
#tick mark size
plt.tick_params(axis='x', length=8, width=2)
plt.tick_params(axis='y', length=8, width=2)
# show the graph
plt.show()
```