SOAC Project Proposal

Team 21- Olivia Bernard (kdg8ac), Gabe Lawrence (vzh2vs), Camp Hagood (nsh4ad) & Drake Ferri (uvt4ct)

Introduction & Problem

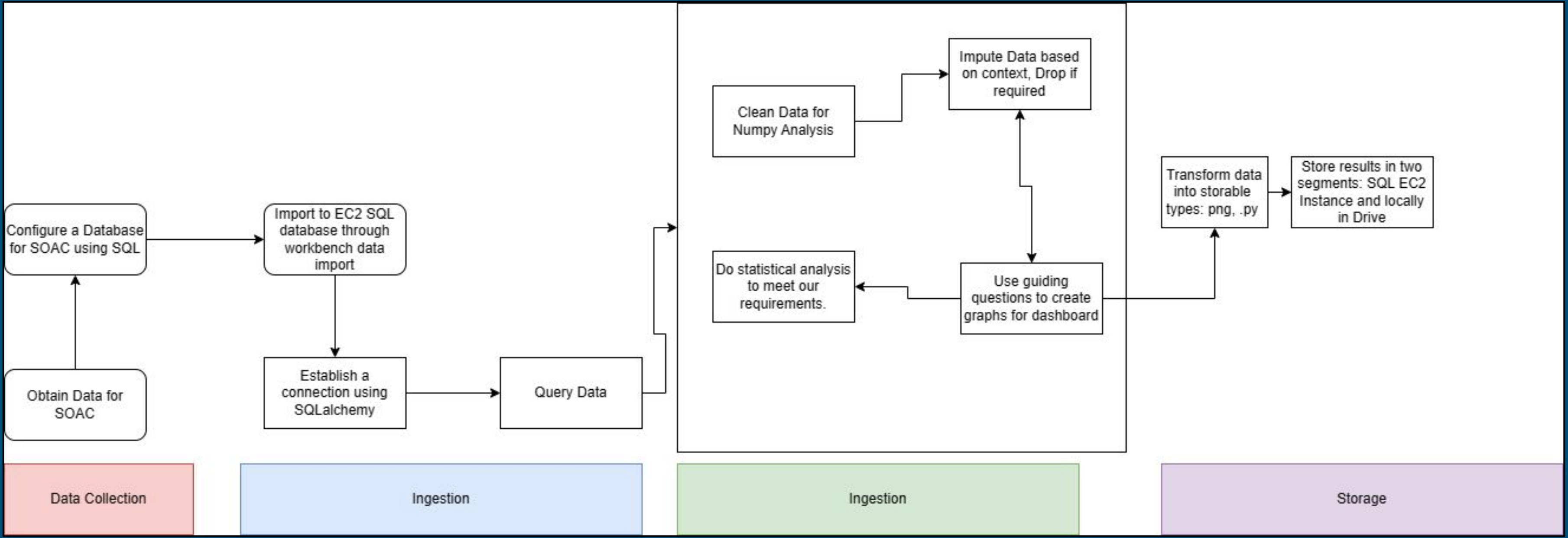
- Investigate the current revenues and costs and the factors that contribute to them
- Focus on enhancing and optimizing profitability through revenues and costs
- Looking at the main costs and cutting back on the money “bleeders” of the company
- Trying to determine if the company is profitable now and where we see it going in the future
- Employ data engineering techniques such as collection, ingestion, cleaning, storage, transformation, visualization, and analysis

Questions

1. How can we increase the amount of revenue per transaction? - Drake
2. How can we optimize salaries to reduce expense while improving profits? - Camp
3. What factors most closely correlate with overall expenses, and how do we optimize them? - Gabe
4. What is the ratio of all expenses over revenue aka the profit margin? - Olivia

Pipeline

Data ingestion, storage, transformation, analysis
Data was queried, cleaned, transformed and visualized using python



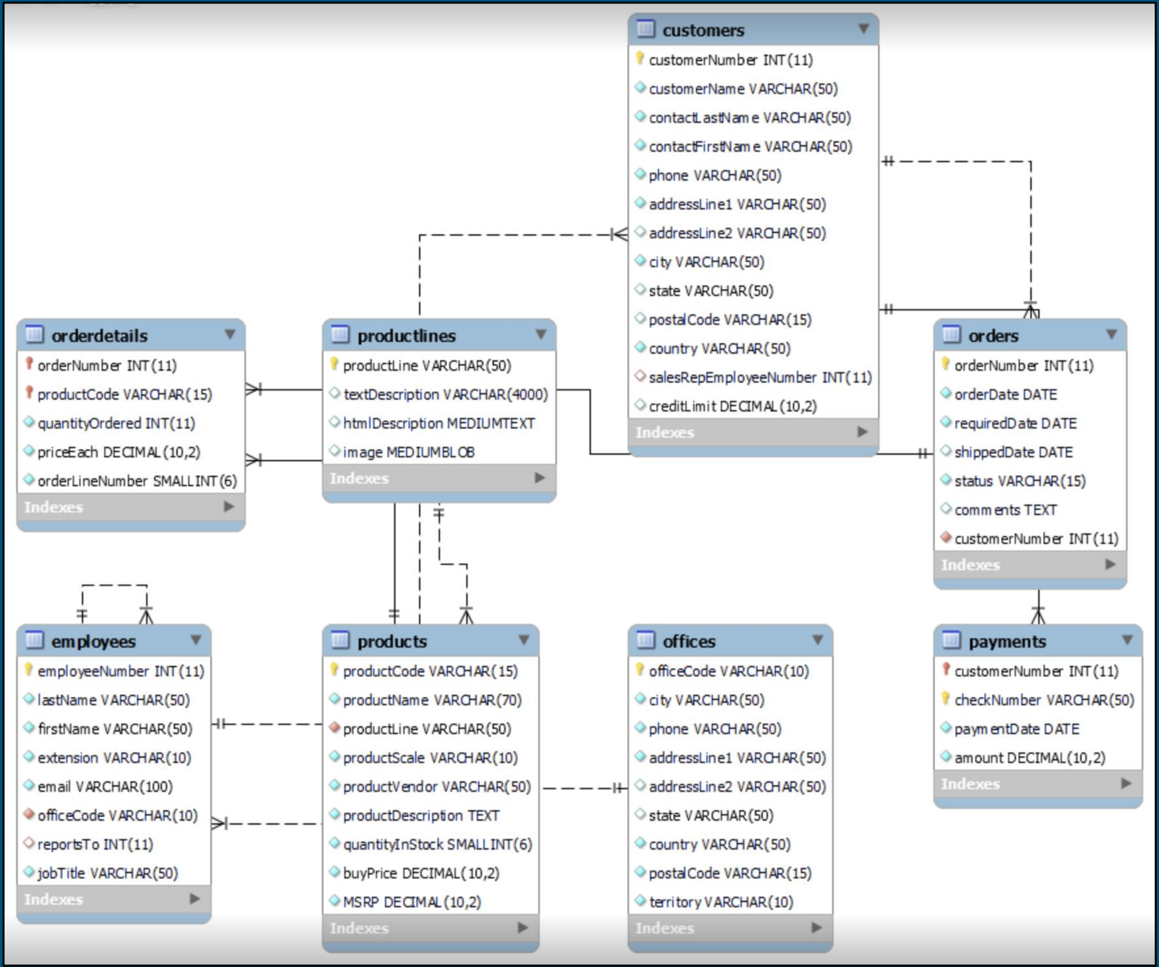
Obtain Data for SOAC
Configure a Database for SOAC using SQL
Import to EC2 SQL database through workbench data import
Establish a connection using SQLalchemy
Query Data
Clean Data for Numpy Analysis
Impute Data based on context, Drop if required
Use guiding questions to create graphs for dashboard
Do statistical analysis to meet our requirements.
Transform data into storable types: png, .py
Store results in two segments: SQL EC2 Instance and locally in Drive
Data Collection
Ingestion
Ingestion
Storage

ERD

Graphical representation of the relationships among contributing elements of the SOAC

Contributing attributes of each entity are grouped

Some of the attributes used to answer our questions



customers
customerNumber INT(11)
customerName VARCHAR(50)
contactLastName VARCHAR(50)
contactFirstName VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(50)
state VARCHAR(50)
postalCode VARCHAR(15)
country VARCHAR(50)
salesRepEmployeeNumber INT(11)
creditLimit DECIMAL(10,2)

orderdetails
orderNumber INT(11)
productCode VARCHAR(15)
quantityOrdered INT(11)
priceEach DECIMAL(10,2)
orderLineNumber SMALLINT(6)

productlines
productLine VARCHAR(50)
textDescription VARCHAR(4000)
htmlDescription MEDIUMTEXT
image MEDIUMBLOB

orders
orderNumber INT(11)
orderDate DATE
requiredDate DATE
shippedDate DATE
status VARCHAR(15)
comments TEXT
customerNumber INT(11)

employees
employeeNumber INT(11)
lastName VARCHAR(50)
firstName VARCHAR(50)
extension VARCHAR(10)
email VARCHAR(100)
officeCode VARCHAR(10)
reportsTo INT(11)
jobTitle VARCHAR(50)

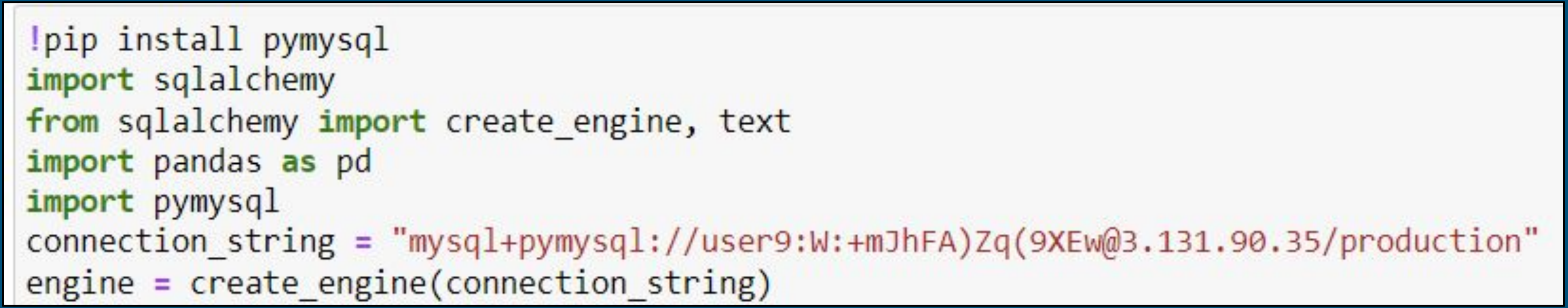
products
productCode VARCHAR(15)
productName VARCHAR(70)
productLine VARCHAR(50)
productScale VARCHAR(10)
productVendor VARCHAR(50)
productDescription TEXT
quantityInStock SMALLINT(6)
buyPrice DECIMAL(10,2)
MSRP DECIMAL(10,2)

offices
officeCode VARCHAR(10)
city VARCHAR(50)
phone VARCHAR(50)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
state VARCHAR(50)
country VARCHAR(50)
postalCode VARCHAR(15)
territory VARCHAR(10)

payments
customerNumber INT(11)
checkNumber VARCHAR(50)
paymentDate DATE
amount DECIMAL(10,2)

Data Storage

Uploaded data to MySQL EC2 instance through a SQL script

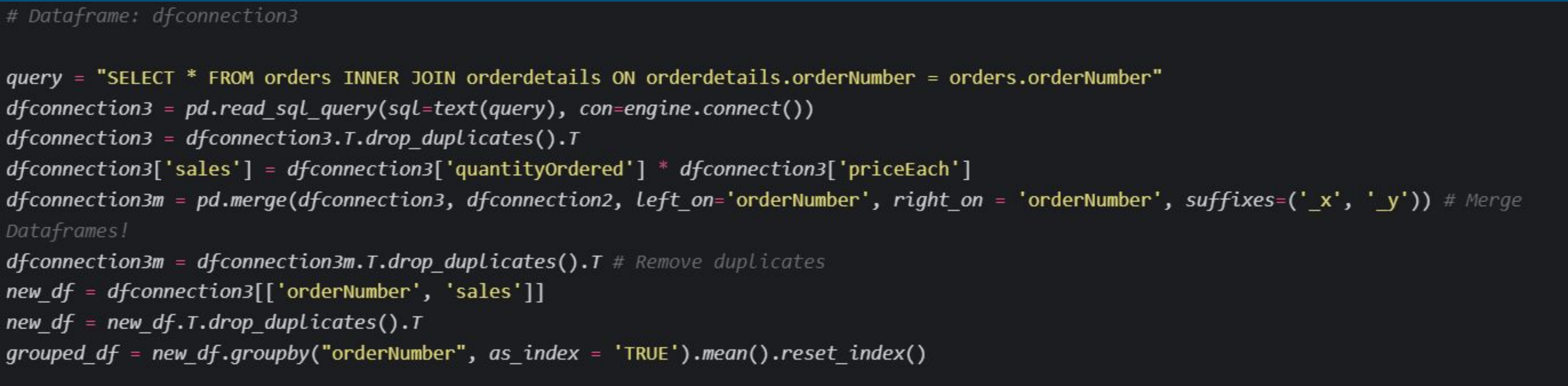


```
!pip install pymysql
import sqlalchemy
from sqlalchemy import create_engine, text
import pandas as pd
import pymysql
connection_string = "mysql+pymysql://user9:W:+mJhFA)Zq(9XEw@3.131.90.35/production"
engine = create_engine(connection_string)
```

Data Cleaning and Processing

- Dropped duplicates, consolidated redundant data, looked for null values, merged columns, grouped by order number

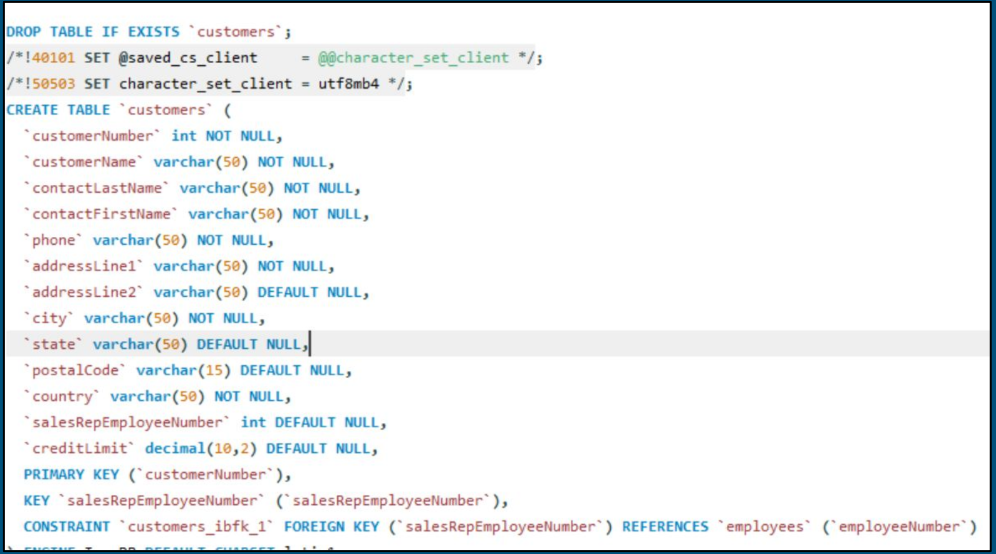
Example code for transforming, cleaning and processing data:



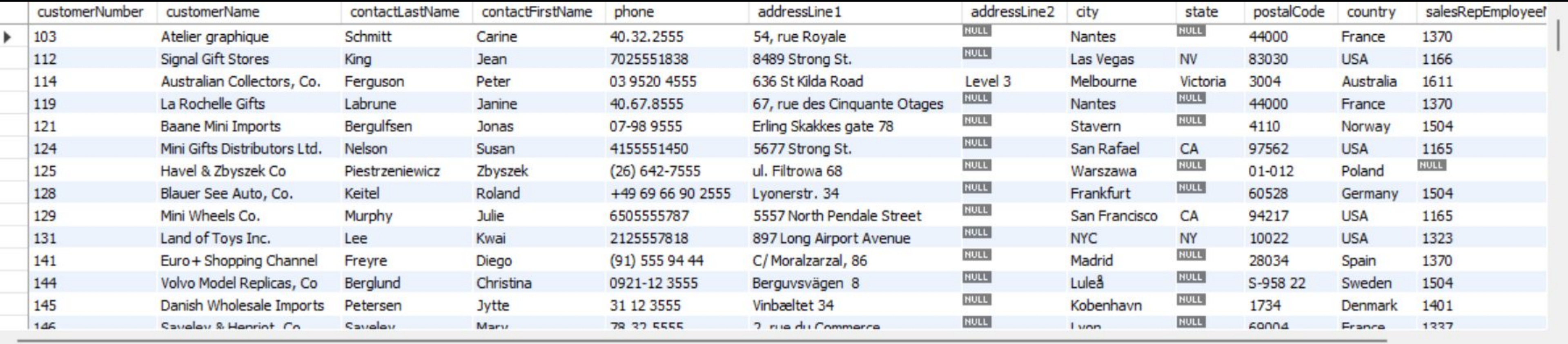
```
# Dataframe: dfconnection3

query = "SELECT * FROM orders INNER JOIN orderdetails ON orderdetails.orderNumber = orders.orderNumber"
dfconnection3 = pd.read_sql_query(sql=text(query), con=engine.connect())
dfconnection3 = dfconnection3.T.drop_duplicates().T
dfconnection3['sales'] = dfconnection3['quantityOrdered'] * dfconnection3['priceEach']
dfconnection3m = pd.merge(dfconnection3, dfconnection2, left_on='orderNumber', right_on = 'orderNumber', suffixes=('_x', '_y')) # Merge
Dataframes!
dfconnection3m = dfconnection3m.T.drop_duplicates().T # Remove duplicates
new_df = dfconnection3[['orderNumber', 'sales']]
new_df = new_df.T.drop_duplicates().T
grouped_df = new_df.groupby("orderNumber", as_index = 'TRUE').mean().reset_index()
```

Data Transformation

The code to the right was used to transform our data into a tabular format. The output can be seen below.



```
DROP TABLE IF EXISTS `customers`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `customers` (
  `customerNumber` int NOT NULL,
  `customerName` varchar(50) NOT NULL,
  `contactLastName` varchar(50) NOT NULL,
  `contactFirstName` varchar(50) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `addressLine1` varchar(50) NOT NULL,
  `addressLine2` varchar(50) DEFAULT NULL,
  `city` varchar(50) NOT NULL,
  `state` varchar(50) DEFAULT NULL,
  `postalCode` varchar(15) DEFAULT NULL,
  `country` varchar(50) NOT NULL,
  `salesRepEmployeeNumber` int DEFAULT NULL,
  `creditLimit` decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (`customerNumber`),
  KEY `salesRepEmployeeNumber` (`salesRepEmployeeNumber`),
  CONSTRAINT `customers_ibfk_1` FOREIGN KEY (`salesRepEmployeeNumber`) REFERENCES `employees` (`employeeNumber`)
```

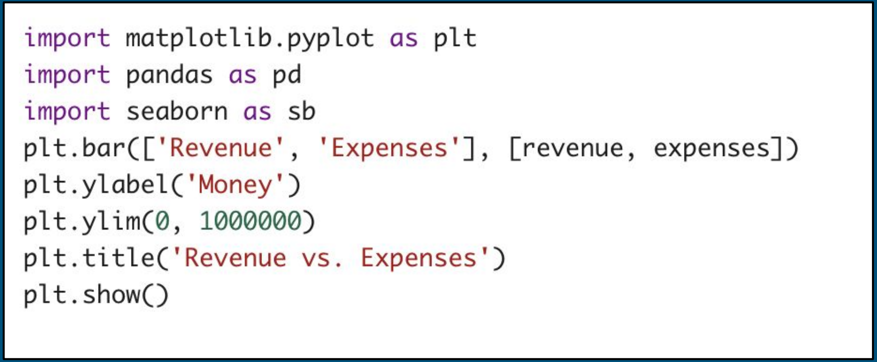


customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployee
103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	NULL	Nantes	NULL	44000	France	1370
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	NULL	Las Vegas	NV	83030	USA	1166
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia	1611
119	La Rochelle Gifts	Labrune	Janine	40.67.8555	67, rue des Cinquante Otages	NULL	Nantes	NULL	44000	France	1370
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555	Erling Skakkes gate 78	NULL	Stavern	NULL	4110	Norway	1504
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA	1165
125	Havel & Zbyszek Co	Piestrzeniewicz	Zbyszek	(26) 642-7555	ul. Filtrowa 68	NULL	Warszawa	NULL	01-012	Poland	NULL
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555	Lyonerstr. 34	NULL	Frankfurt	NULL	60528	Germany	1504
129	Mini Wheels Co.	Murphy	Julie	6505555787	5557 North Pendale Street	NULL	San Francisco	CA	94217	USA	1165
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	NULL	NYC	NY	10022	USA	1323
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44	C/ Moralzarzal, 86	NULL	Madrid	NULL	28034	Spain	1370
144	Volvo Model Replicas, Co	Berglund	Christina	0921-12 3555	Berguvsvägen 8	NULL	Luleå	NULL	S-958 22	Sweden	1504
145	Danish Wholesale Imports	Petersen	Jytte	31 12 3555	Vinbæltet 34	NULL	Kobenhavn	NULL	1734	Denmark	1401
146	Saveley & Henriot, Co.	Saveley	Mary	78 32 5555	2, rue du Commerce	NULL	Lyon	NULL	69004	France	1337

Data Visualization

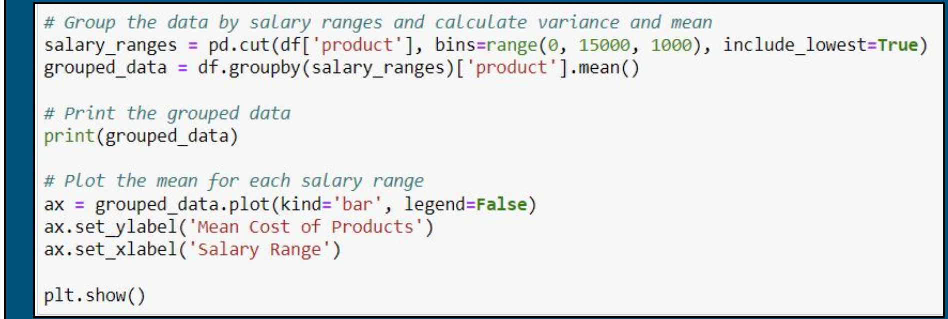
- Python visualization tools on Jupyter Notebook: variation of matplotlib, pandas, seaborn, plotly

Example visuzualtion code:



```python
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sb
plt.bar(['Revenue', 'Expenses'], [revenue, expenses])
plt.ylabel('Money')
plt.ylim(0, 1000000)
plt.title('Revenue vs. Expenses')
plt.show()
```

Code for Question 4 Plot



```python
# Group the data by salary ranges and calculate variance and mean
salary_ranges = pd.cut(df['product'], bins=range(0, 15000, 1000), include_lowest=True)
grouped_data = df.groupby(salary_ranges)['product'].mean()

# Print the grouped data
print(grouped_data)

# Plot the mean for each salary range
ax = grouped_data.plot(kind='bar', legend=False)
ax.set_ylabel('Mean Cost of Products')
ax.set_xlabel('Salary Range')

plt.show()
```

Code for Question 2 Plot

Interactive Dashboard

The Dashboard does the following tasks:

- Calculates Regressions based on the dataset conforming to user requests using Sklearn and streamlit input;
- Displays interactive graphs using plotly and an interactive chart, allowing for easy export of data into different formats;
- Calculates and displays convariences via Numpy and standard scaling;
- Allows for the marking of data on graphs as positive or negative trends.
- Permits the user to move through the different questions and portions of the dataset using interactive menus.

Question 1 (Drake Ferri)

How can we increase the amount of revenue per transaction?

- To first understand the scope of the problem, we wanted to find the quantity ordered per transaction
- Similarly, we also wanted to find the revenue per these same transactions
- Uploading the data from the database, joining tables in sql, and looping through the data gave a comparison of these values
- Utilizing the orders table from the database allows for a direct visualization of quantity and revenue per transaction

Then, it is beneficial to find the employees who gain the most revenue per transaction by comparing the orders and employees tables to find which employees are increasing revenue, which in turn increases profits.

Question 1 Analysis

- The plot is a histogram of the quantity ordered per transaction

As can be seen from the distribution of product orders, the vast majority of orders contain anywhere from 20 to 50 car parts.

- This plot is a histogram of the revenue per transaction

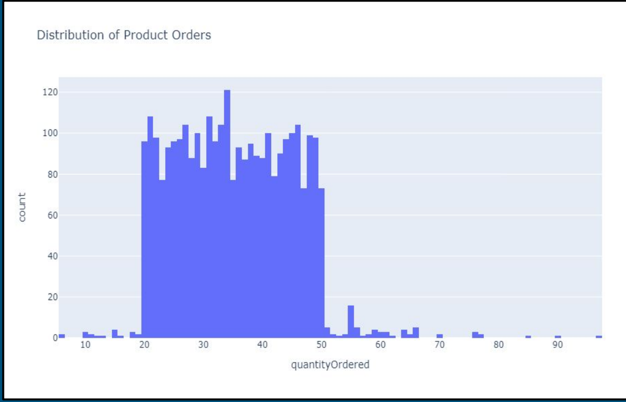
The distribution of revenue per transaction primarily falls within the 2000-4000 range,

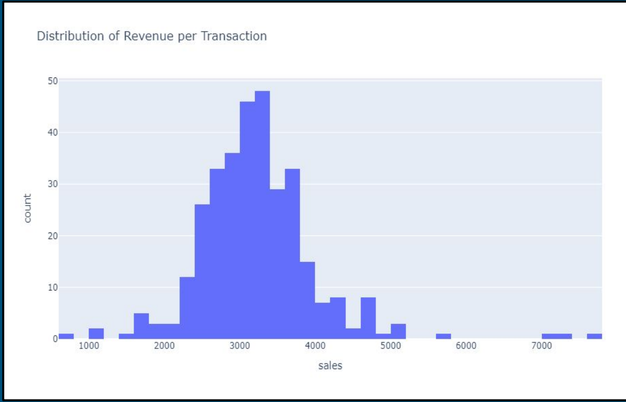
- Most common occurrence is between 3200 and 3400 in revenue with the 3000 to 3200 range having slightly fewer occurrences.

Although the quantity distribution is generally uniform, the revenue per transaction varies much more.

This displays the concept of diminishing returns because as the quantity ordered increases, the revenue per transaction does not always follow the same trend.

Suggestion: Increasing quantity ordered per transaction will not have the same effect on revenue per transaction. SOAC benefits from finding optimal range of quantities.





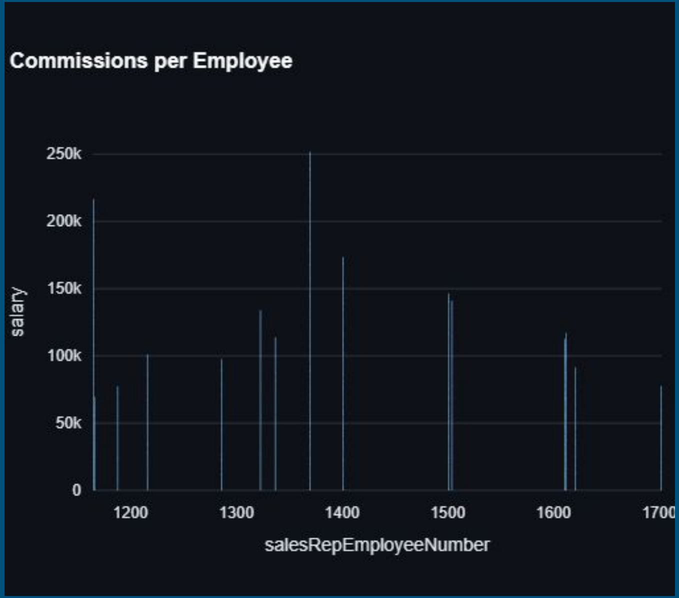
Q1 Analysis Cont.

In this graph, salary from commission is plotted on the y axis while the employee number is plotted on the x axis.

- The commission amongst the employees ranges from around 75k to 250k
- The average of the commission for these employees is in the 100k-150k range

Suggestion:

- This data in this chart should be used when contemplating promotions, hires, as well as layoffs.
- In order to encourage employees to gain more in commission, those whose salaries that are lower should be notified to attempt to improve
    - ○ Allowing for increased revenue per transaction upon improval



Question 2 (Camp Hagood)

How can we optimize salaries to reduce expense while improving profits?

- Analyze the breakdown of each expense to make sure we are not spending money in unnecessary areas
- Perform a statistical analysis in SQL through variance analysis (SQL, VARIANCE()) and T-testing
- Gives us insight into which salary ranges optimize performance on average, as it is known that lower salaries generally decrease the performance of employees and in turn reduce the profit margin and productivity.

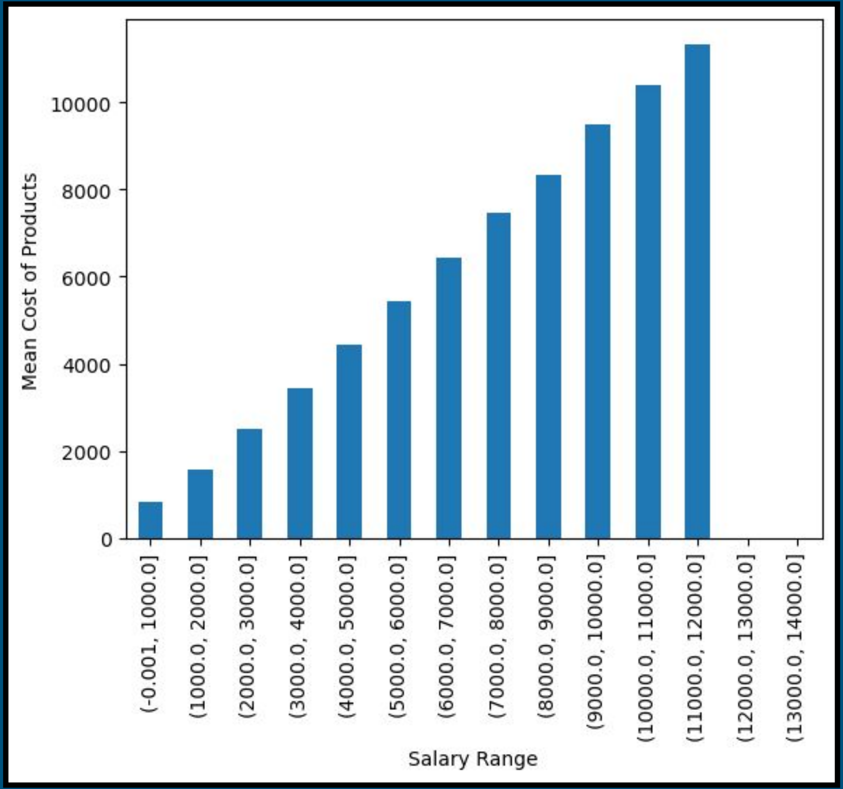
Question 2 (Camp Hagood)

How can we optimize salaries to reduce expense while improving profits?

- To optimize salaries to reduce expense while improving profit we created a plot of the mean and variance for each salary range using the plot function.
- For each salary range group, we calculated the mean price of products ordered along with the price for all order details that fell within that salary range group.
- We then used this information to obtain the mean cost of products ordered for each salary range group.

Question 2 Analysis

- Plot shows the mean product value for each salary range group, allowing us to compare the average performance of order details associated with different salary ranges.
- Shows that lower salaries generally decrease the performance of employees and in turn reduce the profit margin and productivity.
- We can use the graph to determine what salary we want to give employees based off of the mean cost of ordered components.
- For example, if we want an employee to order $6,000 worth of usable car parts we would want to give them a salary in the range of $7,000-$8,000.

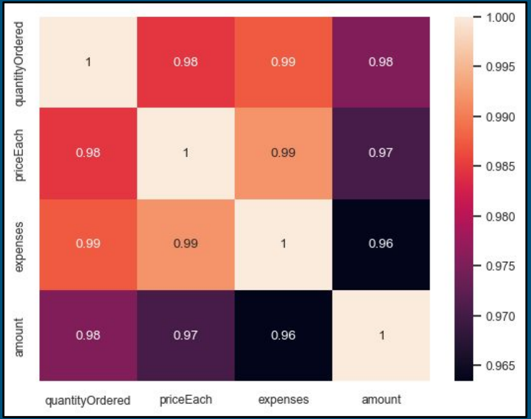


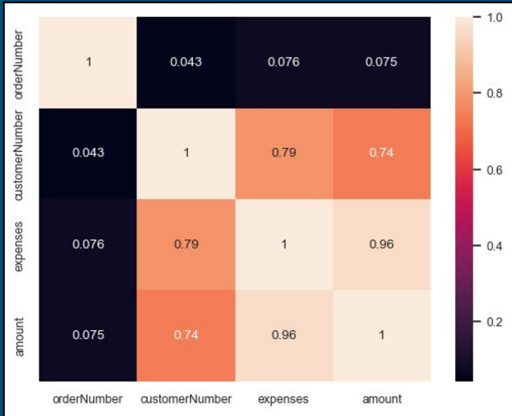
Question 3 (Gabe Lawrence)

What factors most closely correlate with overall expenses, and how do we optimize them?

For the third question **“What factors most closely correlate with overall expenses, and how do we optimize them?**:

- Leverage the fact that the column, “customerNumbers,” is used in the database to link expenses and profit!
- Additionally, this is our first SQL query to make the unified table “SELECT * FROM orders INNER JOIN orderdetails ON orders.orderNumber = orderdetails.orderNumber;”
- Then clean the data frame using drop_duplicates and group_by to make sure that customerNumbers works.
- Then normalize the vectors using standard scaling.
- Build a covariance matrix to determine if expenses could be reduced to improve profit.





Question 3 Analysis

SOAC needs to do two major things, reduce the expenses associated with each unit and increase the number of parts sold:

- Historically, SOAC has been improving their customer selection processes, as there is a correlation between new buyers and more sales, therefore, it might be advantageous to drop or phrase out older customers, as they tend to yield less profit and sales.
- They might need to optimize or reduce the quality of the parts sold to reduce the expenses with each unit; this can be avoided through increasing production, but that might require additional advertisement and investment.

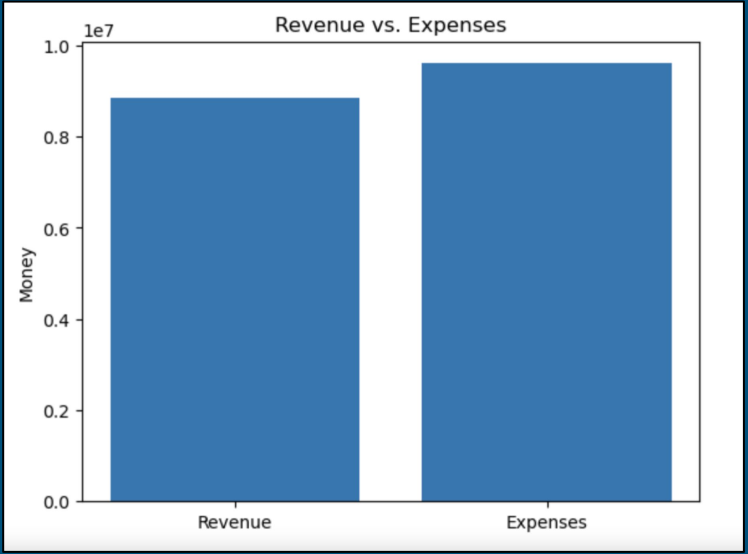
Question 4 (Olivia Bernard)

What is the ratio of all expenses over revenue aka the profit margin?

- Interested in evaluating if the company is profitable with its current revenue stream after it pays for its expenses
- Focused mainly on inventory turnover and its costs VS how much it is sold for
- Evaluated the price of the inventory and the quantity ordered as expenses
- Solved by uploading data from the SOAC database, joining tables in SQL, iterating through tables, creating variables, running equations and visualizing the data with pandas
- Examined the Order Details (cost of acquired inventory) and Payments (money received from customers) tables
- Expenses = price of inventory * quantity ordered
- Revenues = payments from customers

Question 4 Analysis

- Expenses: $9,604,190.61
- Revenues: $8,853,839.23
- Net Profit Margin: $\text{Net Profit Margin} = \frac{\text{Revenue} - \text{Cost}}{\text{Revenue}}$ -7.81%
  - Operating costs are more than revenue
  - Not a sustainable business model
- Profit margin is the measure of a company’s profitability, expressed as the percentage of revenue that the company keeps as profit
- Average profit margin of the Auto Industry is 7.5%



Question 4 Analysis

- Car company profitability tends to be slow starting until it is an established car company (SOAC isn't established) and brand names are very important in auto industry
- A lot of inventory needed for parts and it is very expensive
- Around since 2003 so should be profitable by now so need better management techniques for inventory and need to optimize the price of inventory, when it is ordered and what is used.
- Might need to increase prices of the final product (car) to bring in more revenue

Possible further steps:

- Examining the expenses after other factors such as property, equipment, salaries and wages, buildings and research & development
- Gather data 5-10 years from now
- Generate data on the management of inventory and see if there is room for more efficiency in this process

Conclusion

Problem: Want to improve the company by optimizing business operations while reducing expenses

Steps to Address: Downloaded, transformed, analyzed SOAC’s data

Able to view and manipulate the data to come up with important business metric data and view the strengths and weaknesses of SOAC’s business model.

After analysing revenue and employee salaries, we showed that lower salaries generally decrease the performance of employees and in turn reduce the profit margin and productivity.

In the future, SOAC needs to adjust their business model to reduce cost and continue obtaining new clients with the methodology that they have adopted after the midpoint of their data, as profit is heavily correlated (97%) with cost, and in our normative scenario, profit will be more independent from profit.

SOAC can utilize more efficient inventory management and buying techniques so the inventory is turned over to profit at a faster rate.

By finding the range of quantities per transaction that optimize revenue, SOAC will be able to maximize revenue per transaction.

SOAC can utilize commission per employee to gather necessary data for making employee decisions such as promotions or layoffs.

Future: Take steps to decrease unnecessary or inefficient expenses and then continue to analyze the data and gather more to monitor changes and see if SOAC can become more profitable