Dalton Murray

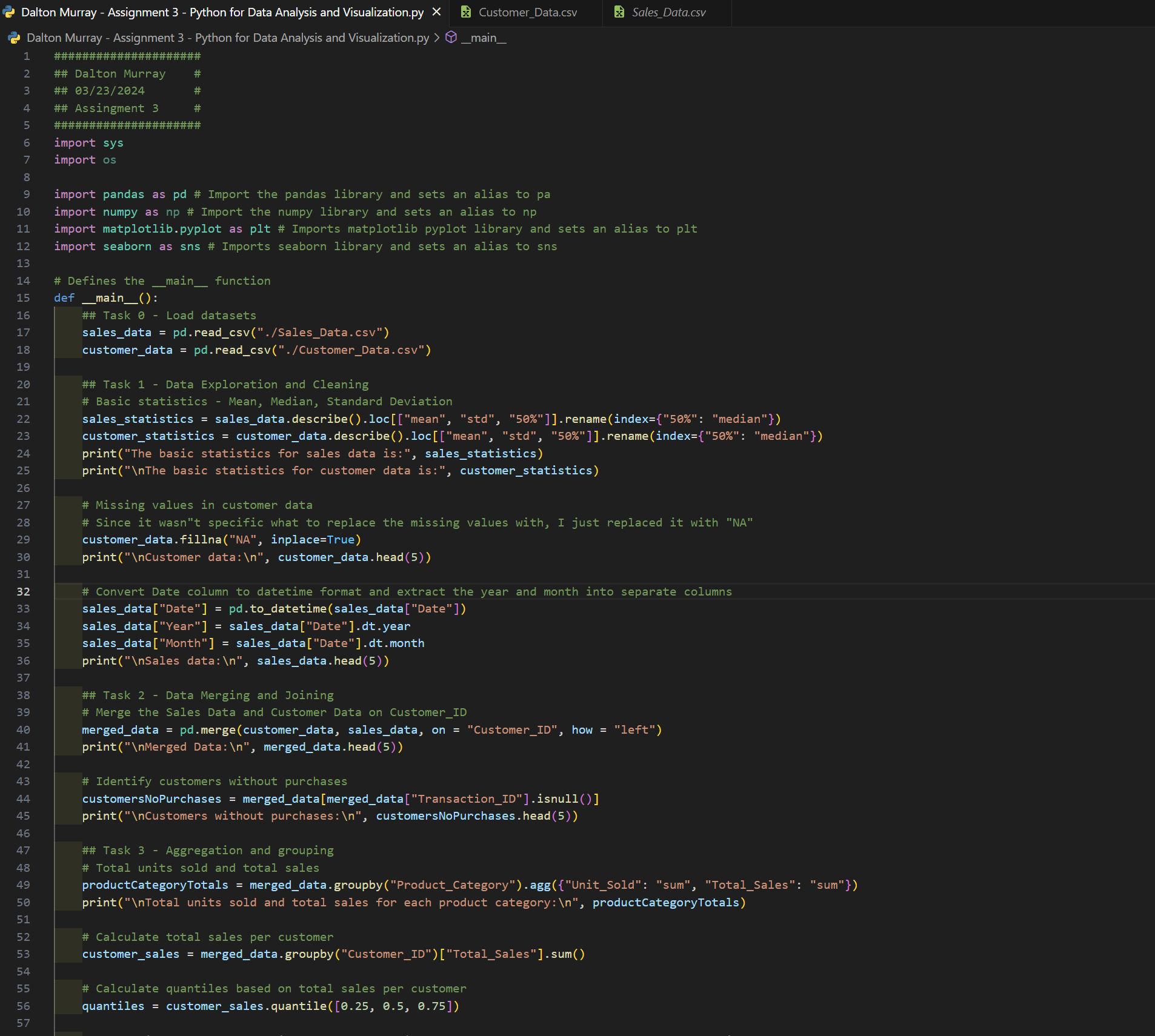
Python for Data Analysis and Visualization

Professor Shams

March 23, 2024

**Assignment #3**

**Overall**



A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen with text on it

Description automatically generated

Code:

#####################

## Dalton Murray #

## 03/23/2024 #

## Assingment 3 #

#####################

import sys

import os

import pandas as pd # Import the pandas library and sets an alias to pa

import numpy as np # Import the numpy library and sets an alias to np

import matplotlib.pyplot as plt # Imports matplotlib pyplot library and sets an alias to plt

import seaborn as sns # Imports seaborn library and sets an alias to sns

# Defines the \_\_main\_\_ function

def \_\_main\_\_():

## Task 0 - Load datasets

sales\_data = pd.read\_csv("./Sales\_Data.csv")

customer\_data = pd.read\_csv("./Customer\_Data.csv")

## Task 1 - Data Exploration and Cleaning

# Basic statistics - Mean, Median, Standard Deviation

sales\_statistics = sales\_data.describe().loc[["mean", "std", "50%"]].rename(index={"50%": "median"})

customer\_statistics = customer\_data.describe().loc[["mean", "std", "50%"]].rename(index={"50%": "median"})

print("The basic statistics for sales data is:\n", sales\_statistics)

print("\nThe basic statistics for customer data is:\n", customer\_statistics)

# Missing values in customer data

# Since it wasn"t specific what to replace the missing values with, I just replaced it with "NA"

customer\_data.fillna("NA", inplace = True)

print("\nCustomer data:\n", customer\_data.head(5))

# Convert Date column to datetime format and extract the year and month into separate columns

sales\_data["Date"] = pd.to\_datetime(sales\_data["Date"])

sales\_data["Year"] = sales\_data["Date"].dt.year

sales\_data["Month"] = sales\_data["Date"].dt.month

print("\nSales data:\n", sales\_data.head(5))

## Task 2 - Data Merging and Joining

# Merge the Sales Data and Customer Data on Customer\_ID

merged\_data = pd.merge(customer\_data, sales\_data, on = "Customer\_ID", how = "left")

print("\nMerged Data:\n", merged\_data.head(5))

# Identify customers without purchases

customersNoPurchases = merged\_data[merged\_data["Transaction\_ID"].isnull()]

print("\nCustomers without purchases:\n", customersNoPurchases.head(5))

## Task 3 - Aggregation and grouping

# Total units sold and total sales

productCategoryTotals = merged\_data.groupby("Product\_Category").agg({"Unit\_Sold": "sum", "Total\_Sales": "sum"})

print("\nTotal units sold and total sales for each product category:\n", productCategoryTotals)

# Calculate total sales per customer

customer\_sales = merged\_data.groupby("Customer\_ID")["Total\_Sales"].sum()

# Calculate quantiles based on total sales per customer

quantiles = customer\_sales.quantile([0.25, 0.5, 0.75])

# Categorize customers as "High spender", "Medium spender", and "Low spender" based on quantiles

customerQuantiles = customer\_sales.apply(

lambda x: "High spender"

if x >= quantiles[0.75]

else ("Medium spender"

if x >= quantiles[0.5]

else "Low spender")

)

print("Average purchase amount per customer:\n", customer\_sales.head(5))

print("\nCustomer quantiles:\n", customerQuantiles.head(5))

## Task 4 - Time series analysis

# Analyze sales trends over months

monthly\_sales = merged\_data.groupby("Month")["Total\_Sales"].sum()

plt.figure(figsize = (10, 6))

monthly\_sales.plot(kind="line", marker = "o")

plt.title("Monthly Sales Trend")

plt.xlabel("Month")

plt.ylabel("Total Sales")

plt.grid(True)

plt.xticks(range(1, 13))

# plt.show()

# Identify the month with the highest sales

monthHighestSale = monthly\_sales.idxmax()

highestSalesAmount = monthly\_sales.max()

print("\nThe month with the highest sales is:", monthHighestSale)

print("\nThis month has a total sales of: $", highestSalesAmount)

# Month-Over-Month growth rate in sales

monthlyGrowthRate = monthly\_sales.pct\_change() \* 100

print("\nMonth-over-month growth rate in sales:\n", monthlyGrowthRate)

## Task 5 - Customer Demographics Analysis

# Age distribution of customers

plt.figure(figsize = (10, 6))

sns.histplot(customer\_data["Age"], bins = 20, kde = True)

plt.title("Age Distribution of Customers")

plt.xlabel("Age")

plt.ylabel("Frequency")

plt.grid(True)

# plt.show()

# Most prevalent age group

mostPrevalentAgeGroup = customer\_data["Age"].mode()[0]

print("\nThe most prevalent age group is:", mostPrevalentAgeGroup)

# Membership tier distribution

plt.figure(figsize = (8, 5))

sns.countplot(x = "Membership\_Tier", data = merged\_data, palette = "Set2", hue = "Membership\_Tier")

plt.title("Membership Tier Distribution")

plt.xlabel("Membership Tier")

plt.ylabel("Count")

plt.grid(axis = "y")

# plt.show()

# Calculate total sales for each membership tier

membershipSales = merged\_data.groupby("Membership\_Tier")["Total\_Sales"].sum()

print("\nTotal sales for each membership tier:\n", membershipSales)

# Analyze the correlation between membership tiers and sales

correlation = merged\_data.groupby("Membership\_Tier")["Total\_Sales"].mean()

print("\nCorrelation between membership tiers and average sales:\n", correlation)

## Task 6 - Regional analysis

# Total sales and average unit price in each region

region\_data = merged\_data.groupby("Region")

regionAvgPrice = region\_data.agg({"Total\_Sales": "sum", "Price\_per\_Unit": "mean"})

print("Total sales and average unit price in each region:\n", regionAvgPrice)

# Product category highest sales

regionHighestSalesCat = region\_data.apply(lambda x: x.groupby("Product\_Category")["Total\_Sales"].sum().idxmax())

print("\nProduct category with the highest sales in each region:\n", regionHighestSalesCat)

## Task 7 - Advanced analysis

# Top 10 customers purchase amount

topCustomers = merged\_data.groupby("Customer\_ID")["Total\_Sales"].sum().nlargest(10)

print("\nTop 10 customers based on purchase amount:\n", topCustomers)

# Analyze buying pattern

topCustomers\_data = merged\_data[merged\_data["Customer\_ID"].isin(topCustomers.index)]

print("\nBuying patterns of top customers:\n", topCustomers\_data.head(5))

# Pivot table showing total units sold per prduct category across each region

pivotTableTopCustomer = pd.pivot\_table(topCustomers\_data, values = "Unit\_Sold", index = "Product\_Category", columns = "Region", aggfunc = "sum", fill\_value = 0)

print("\nPivot table showing total units sold for each product category across each region:\n", pivotTableTopCustomer)

## Task 8 - Visualization

# Bar chart showing number of transactions per product category

transactionCount = merged\_data["Product\_Category"].value\_counts()

plt.figure(figsize=(10, 6))

transactionCount.plot(kind = "bar")

plt.title("Number of Transactions per Product Category")

plt.xlabel("Product Category")

plt.ylabel("Number of Transactions")

plt.xticks(rotation = 45, ha = "right")

plt.tight\_layout()

# plt.show()

# Line graph of monthly sales over year

monthlySales = merged\_data.groupby("Month")["Total\_Sales"].sum()

plt.figure(figsize = (10, 6))

monthly\_sales.plot(marker = "o", linestyle = "-")

plt.title("Monthly Sales Over the Year")

plt.xlabel("Month")

plt.ylabel("Total Sales")

plt.xticks(monthly\_sales.index)

plt.grid(True)

plt.tight\_layout()

# plt.show()

# Checks if the "\_\_name\_\_" variable equals "\_\_main\_\_"

if \_\_name\_\_ == "\_\_main\_\_":

sys.exit(\_\_main\_\_()) # Calls the "\_\_main\_\_" function and then after running exits smoothly

**Task 0 - Load datasets**

A black background with text

Description automatically generated

In order to start doing anything I must first read and save the datasets as a variable.

**Task 1 - Data Exploration and Cleaning**

A computer screen with colorful text

Description automatically generated

Basic statistics –

Instead of using separate calculations on each column I simply use .describe(). Within this it automatically performs a bunch of different basic calculations. I extract the mean, standard deviation, and 50%. I then rename 50% as median. I then print out this data.

A screenshot of a computer

Description automatically generated

Missing values –

Since there weren’t specifics of what to replace the missing values with I decided to just fill it in with “NA” and to do it in place because I don’t want to create a new dataframe/table. I then print this out.

A screen shot of a computer program

Description automatically generated

Convert date column –

In order to convert the date into year, and month into date time format I use pd.to\_datetime and tell it to go based on “Date”. This then allows me to make year and month columns based on using the date datetime formatted column.

A screenshot of a computer screen

Description automatically generated

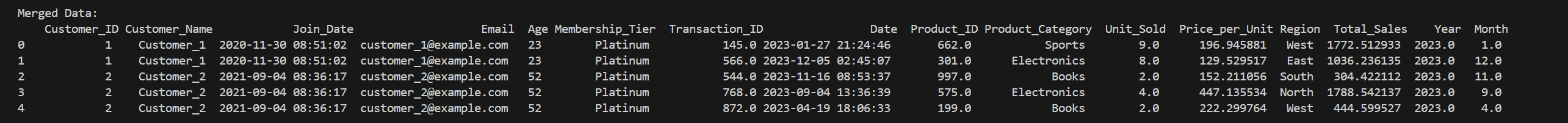
**Task 2- Data Merging and Joining**

**A screen shot of a computer code

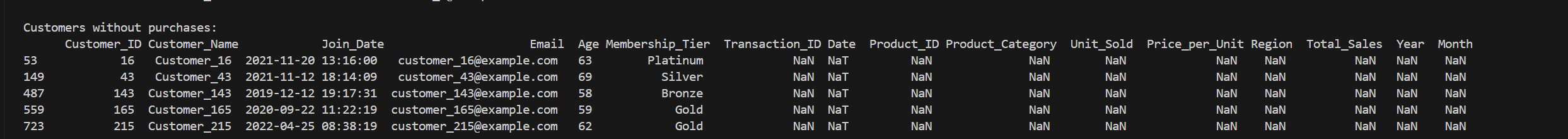
Description automatically generated**

Merge sales data and customer data on customer\_id –

I decided on doing a left merge on customer\_id because this allows me to keep customer\_data on the left side of the new dataframe and just easily combine sales\_data onto the right side of it where there are matching rows with customer\_id. This makes it so that if there is a missing customer id on the left side that the right side will be filled with NaN, allowing me to keep all sakes data and match it where it can.



Customers without purchases –

There wasn’t a given way to decide to base this on what to determine where there were no sales, however, I can easily just say where a certain column is equal to null, such as Transaction\_ID, which would then mean there were no purchases by that customer. 

**Task 3 - Aggregation and grouping**

**A screen shot of a computer code

Description automatically generated**

Total units sold and total sales –

With this I grouped by Product\_Category and aggregated the Unit\_Sold, Total\_Sales columns and got the sums of both of them so this then associates with product\_category.

A screenshot of a computer screen

Description automatically generated

Average purchase amount per customer –

With this all I did was use sum on total sales grouping by customer id.

A screen shot of a computer

Description automatically generated

Calculate quantiles and label high spender, medium, low –

In this I calculated the total sales per customer, I then calculated quantiles as 0.25, 0.5, and 0.75. Based on the quantiles I then labeled the customer as a higher spender, medium spender, or a low spender.

A screen shot of a computer

Description automatically generated

**Task 4 - Time series analysis**

**A computer screen with colorful text

Description automatically generated**

Analyze sales trends over months –

In order to do this I believed a line graph would be best. I calculated the monthly sales by grouping by month and summing the total sales. I then made a basic plot and printed it out.

A graph with blue lines and numbers

Description automatically generated

Month with highest sales –

In order to do this I took the row label maximum value of the month sales and then got the maximum value of this.

A black background with white text

Description automatically generated

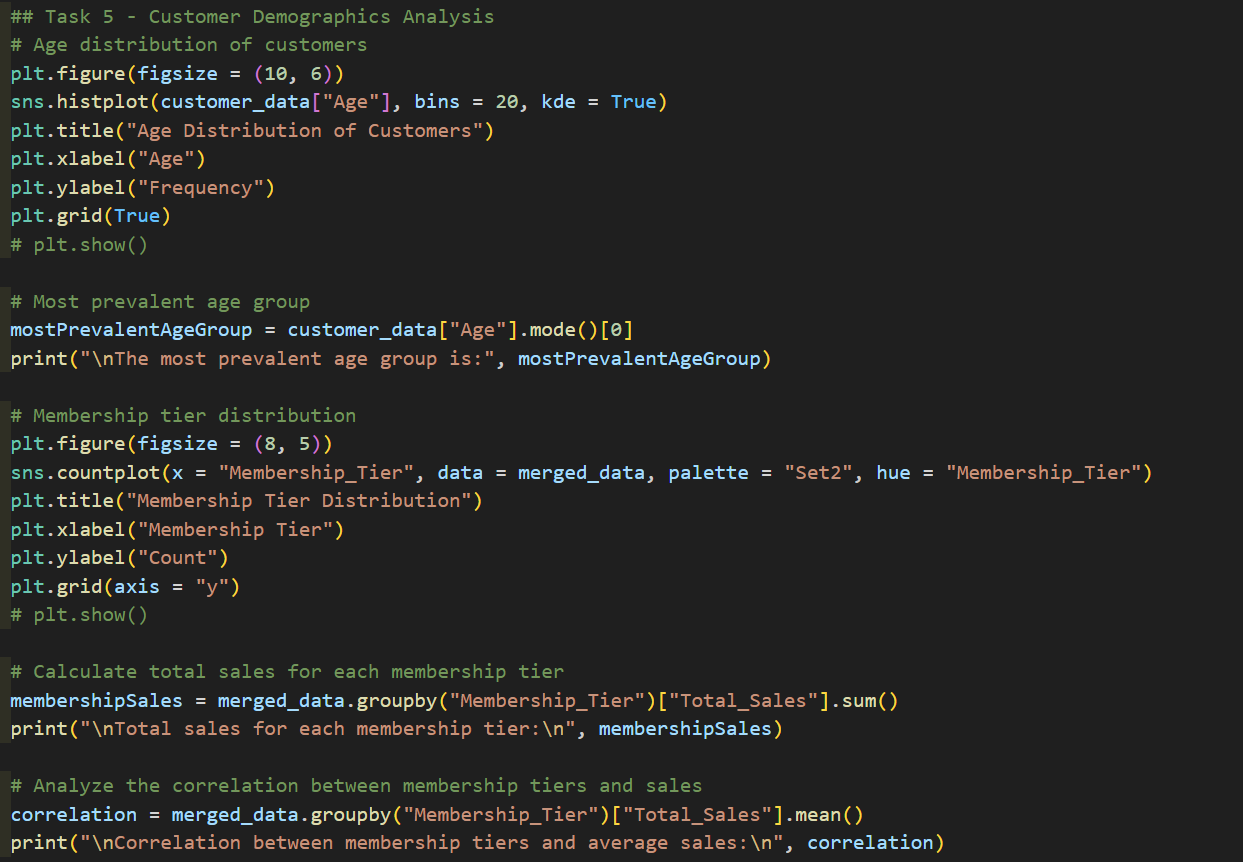
Month-over-month growth rate –

In order to do this I took the monthly sales fractional change between the current element and the prior element, think of it like shifting between the months, and then I multiplied this by 100 to get a percentage.

A screenshot of a computer

Description automatically generated

**Task 5 - Customer Demographics Analysis**



For the first one showing the age distribution all I did was make a histogram plot and I based this off of the customer data age. I then set how many bins I want and showed the kernel density element to get a line showing the probability of someone being that age.

A graph with blue lines and a blue line

Description automatically generated

Most prevalent age group –

All I did for this was get the mode of age

A black background with white text

Description automatically generated

Membership tier distribution –

To do this I used another count plot based on Membership tier.

A screenshot of a computer

Description automatically generated

Total sales each membership tier –

For this all I did was take the sum of total sales in groups of membership tiers A screenshot of a computer screen

Description automatically generated

Correlation between membership tiers and sales –

For this since I am basing it off of membership tier all I have to do is take the mean of total sales because it is already creating a correlation rather than using .corr()

A black screen with white text

Description automatically generated

Using this I am also able to tell that there is a correlation between who buys what tier. Most people buy the gold tier, then the platinum tier, then the bronze tier, then the silver tier.

**Task 6 - Regional analysis**

A screen shot of a computer code

Description automatically generated  
Total sales and average unit price in each region –

All I do for this is get the region data by grouping by it, then I aggregate total sales with sum and price per unit with mean.

A screen shot of a computer

Description automatically generated

Product category with highest sales –

To do this I use a lambda function to go through each category in product categories and get the sum of the total sales and get the max per each.

A black screen with white text

Description automatically generated

**Task 7 - Advanced analysis**

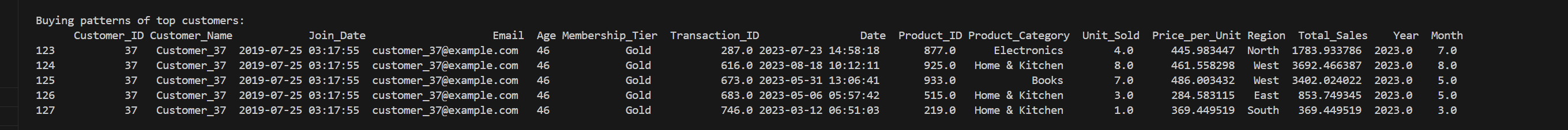
Top 10 customer purchase amount –

For this one I summed total sales and got the top 10 largest sums, and grouped by customer id

A screenshot of a computer

Description automatically generated

Analyze buying pattern – I wasn’t completely sure what was meant by this so all I did was take customer id and say if it’s in top customers to then get all of the rest of their information.



Pivot table –

For this what I did was make a pivot data based on the top customer data using the values of unit sold, index product category, columns region, sum data, and fill missing information, not that there should be any anyways as it was done previously.

A screen shot of a computer

Description automatically generated

**Task 8 – Visualization**

Bar chart –

For this one it was to do a bar chart based on product category count, so all I did was make the data the counts of each category then made it a bar chart.

A screenshot of a graph

Description automatically generated

Line graph –

For this one it was to make a line graph of monthly sales over the year, which I basically already did but did again. I did this grouped by months using sum of total sales.

A graph with blue lines

Description automatically generated

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.

*Dalton Murray*