BUSINESS SALES AND EXPENSE MONITOR WITH STANDARD DEVIATION AND QUARTERLY TREND ANALYSIS

A System Presented to the
Faculty of College of Arts and Sciences
Rizal Technological University

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Statistics

by
Angelica V. Aniñon
Kyla Krylle P. Asuncion
Christiffany B. Casas
Eunice C. Descutido
Christopher S. Go
Bernadette R. Nonoy
Alexandria D. Sta. Maria

May 2023

Table of Contents

Description of the Proposed System

In today's highly competitive business landscape, small businesses face numerous challenges in keeping up with the market's ever-changing demands. One of the most critical aspects of running a small business is keeping track of its financial health, including sales and expenses. To help address this need, the group has created a Python system called the "Business Sales and Expense Monitor with Standard Deviation and Quarterly Trend Analysis", a useful tool that enables small business owners to track and analyze their financial data easily.

The system simplifies the process of analyzing a business's sales performance by collecting and organizing daily and weekly sales and expenses. With this information, business owners can access a range of essential financial metrics, including net profit, highest and lowest sales, average income, and standard deviation of monthly sales.

In addition, the system also includes a quarterly trend analysis that provides a summary of the business's financial data every three months.With the use of this tool,small businesses owners may recognize long-term patterns and plan for the future of their enterprise.

Purpose

The system is designed to be easy to use and accessible to all business owners, regardless of their experience or background. The system tracks financial transactions automatically or manually, ensuring that all critical data is captured accurately. With daily and weekly sales tracking, business owners can stay on top of their revenue and expenses, enabling them to make informed decisions and plan for the future.

One of the system's key features is its quarterly trend analysis, which provides an in-depth look at the business's performance over time. By analyzing financial data on a year-over-year basis, business owners can identify trends and patterns that may impact their business in the long run. This feature is especially valuable for newcomers to the business world, who may not have the experience or knowledge necessary to analyze financial data effectively.

Altogether, the proposed system is a powerful tool that helps small business owners track and analyze their financial data with ease. With its simple and intuitive interface, business owners can access critical financial metrics, identify trends and patterns, and make informed decisions about their business's future. Whether you're a seasoned business owner or a newcomer to the industry, our system is an invaluable resource that can help you succeed and thrive in today's challenging business environment.

Scope

The proposed system aims to provide small business owners with a comprehensive tool to monitor and analyze their sales and expenses on a daily, weekly and quarterly basis. By utilizing this system, business owners will have a better understanding of the financial performance of their enterprise, allowing them to make informed decisions and improve their overall success. The system will include features such as daily and weekly sales monitoring, daily and weekly expense tracking, net profit calculation, comparison of highest and lowest sales, and quarterly trend analysis. The quarterly trend analysis feature will provide long-term insights into the business's performance and help business owners make informed decisions. The system

will also calculate the standard deviation of the monthly sales data to determine the business's sales volatility.

The system will exclude features such as inventory tracking, purchase order management, and customer relationship management. These functions are outside the scope of the proposed system and will not be included. However, the system will allow for manual or automatic data entry to track financial transactions, which can be useful in maintaining the accuracy of the data.

External factors that may impact the system include changes in tax laws, economic conditions, and the availability of internet connectivity. As tax laws and economic conditions may influence a business's profitability, the system may need to be updated to reflect these changes. Additionally, since the system will rely on internet connectivity to access cloud storage, network disruptions or connectivity issues may impact the system's accessibility and reliability.

Overall, the proposed system has specific functions, features, and processes that will aid small business owners in monitoring their sales and expenses. While certain features will be excluded, the system will provide valuable insights into a business's performance over time. External factors such as tax laws, economic conditions, and internet connectivity may impact the system's effectiveness and will need to be considered.

Definition of Terms

Average Sales. Total earnings of the user that the program sums together, providing the gross amount of the user's Average Sales per Day within a month. The user will be able to identify which days of the month wherein their business' sales are flourishing and where they may need improvements.

Business Sales. The system will collect all the user's earnings that come from their business operations, primarily their daily Sales. The system will calculate these daily Sales along with the user's Total Expenses to provide the user with their business' Total Monthly Sales, Monthly Net Income, and Average Sales per Day of the month. These features reflect the user's financial business state each month of the year and can be used for monthly/annual business management.

Expense Monitor. The system collects all the user's Total Monthly Expenses, providing the gross amount that the user can monitor and compare with other months' Total Expenses. This feature will help the user keep their monthly expenses in line with their monthly/annual business budget.

Net Income. The system displays the Net Income that reflects the user's total earnings after the program has deducted the user's Total Monthly Expenses from their Total Monthly Sales. Providing the user a proper look at their business' total monthly revenue.

Quarterly Trend Analysis. The system utilizes the Quarterly Trend Analysis tool to provide the user with a year-over-year comparison of their business' sales performance. The system will display the user's current Annual Sales from a particular month

(e.g., January Sales 2020) and compare them with previous Annual Sales from that same month (e.g., January Sales 2021). This feature provides the user with information on their business' annual sales performance for better decision-making business actions in the future.

Standard Deviation. This feature in the system will calculate the user's daily Sales, estimating how much they've earned within a month from their business' product/service/property price.

Total Expenses. All expenses that the user has spent from their business' daily Sales within a month. This provides the system a basis to calculate the user's Monthly Net Income by subtracting their Total Expenses from their daily Sales. This feature will also benefit the user to keep track of how much they're spending on their business operation on a monthly basis.

Total Sales. The system will sum up all the user's input Daily Business Sales from every business day operation into Total Monthly Sales. This provides the user with information about their business' seasonality, identifying what months their business performance peaks the most and what months their business struggles.

Monitor. In the context of business sales and expenses, monitoring refers to the systematic tracking and analysis of sales and expense data over time. It involves closely observing and recording relevant financial information to gain insights into the financial performance of the business.

Systems Available in Global Market

According to studies(Bhatt S, 2023), a number of systems with features comparable to the suggested tool are already in use on a global scale. Several instances include:

1. An array of financial management tools, including sales and spending monitoring, standard deviation analysis, and quarterly trend analysis, are offered by the well-known accounting program QuickBooks. Users may create bespoke reports with QuickBooks, which also offers predictive analytics features (Shweta and Bottorff,2023).

2. Another accounting program with capabilities like sales monitoring, spending tracking, and financial reporting is Zoho Books. Standard deviation, trend analysis, and the ability to create custom reports are all included in Zoho Books. (IMTS Enterprise Solutions,2021).

3. Xero is an online accounting program that has functions including spending tracking, invoicing, and bank reconciliation. Standard deviation and trend analysis are also included in Xero, and customers may create financial reports that are tailored to their needs (Financials, 2020).

However, the provided system still offers a number of distinct advantages over current options like QuickBooks, Zoho Books, and Xero. The first thing about it is that it is created especially for tracking sales and expenses, with an emphasis on standard deviation and trend analysis, giving more in-depth

insights into financial data. Second, it is extremely flexible and can provide reports and visualizations that are customized to the user's needs. Lastly, it provides predictive analytics capabilities that let businesses foresee future patterns and take preventative action.

In general, this instrument can be the best option for small businesses that are looking for a specialist financial analysis and forecasting tool that might offer distinct benefits over present methods.

Enhancement

The system, "Business Sales and Expense Monitor with Standard Deviation and Quarterly Trend Analysis," is a dynamic system that has the potential to be enhanced and upgraded in the future. Here are some possible improvements:

First, the system could be moved to a store in the cloud system to increase its scalability. Users would be able to keep a huge amount of data in the cloud and the system would be able to expand in accordance with user needs. Additionally, cloud connectivity would provide remote system access and team collaboration for users.

Second, an automated data extraction tool might be linked with the system to make it more flexible. By automatically obtaining data from numerous sources such as spreadsheets, databases, and social media platforms, this would help users save time and avoid errors.

Third, a mobile app can be developed to increase the system's adaptability. As a result, users could be able to log into the system from anywhere, enter data, view dashboards, and get notifications on their mobile devices.

Fourth, the system might use machine learning algorithms to enable more precise forecasting of trends and patterns in sales and expenses. Users would gain a broader understanding of their business and be more capable to make decisions as an outcome.

Fifth, by integrating the proposed system with other business systems such as financial accounting, inventory and order management, production, customer relationship management, or payroll services, the system might be improved.

Finally, the system might be expanded to provide accessible reports. Users could customize reports based on specified filters, metrics, and time frames and generate them in a variety of formats, including Excel, PDF, and CSV. This would increase the system's flexibility and give users access to the precise data they require for their analyses.

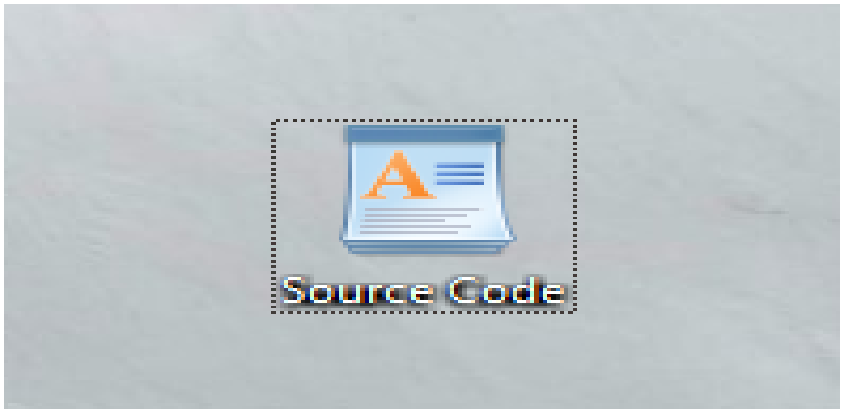
User Manual

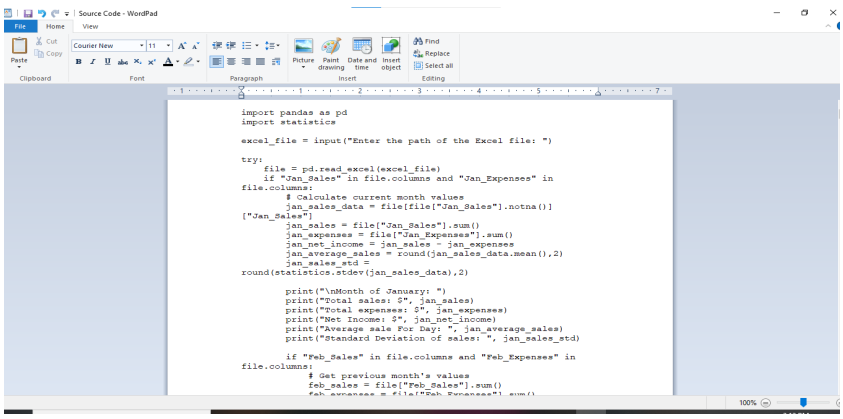
To use the Business Sales and Expense Monitor system program with Standard Deviation and Monthly Trend Analysis, please follow these step-by-step instructions:

1. Ensure that you have Python and the necessary libraries installed on your computer to run the program. You will need libraries such as pandas and statistics for data analysis.

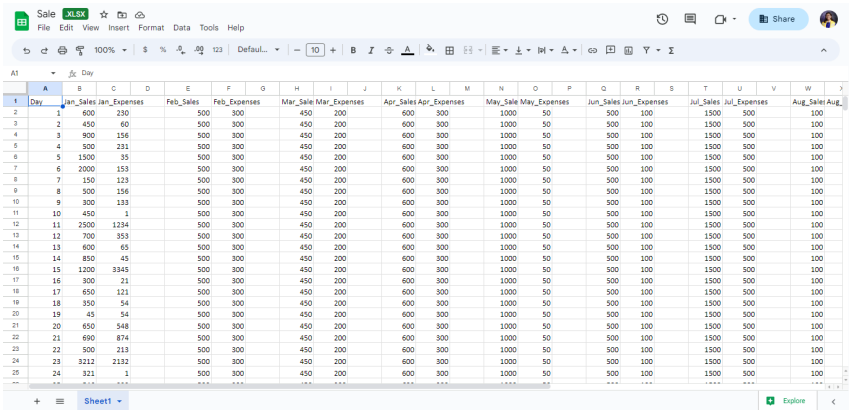
User can use any of the following applications:

2. Download the source code of the Business Sales and Expense Monitor system program.



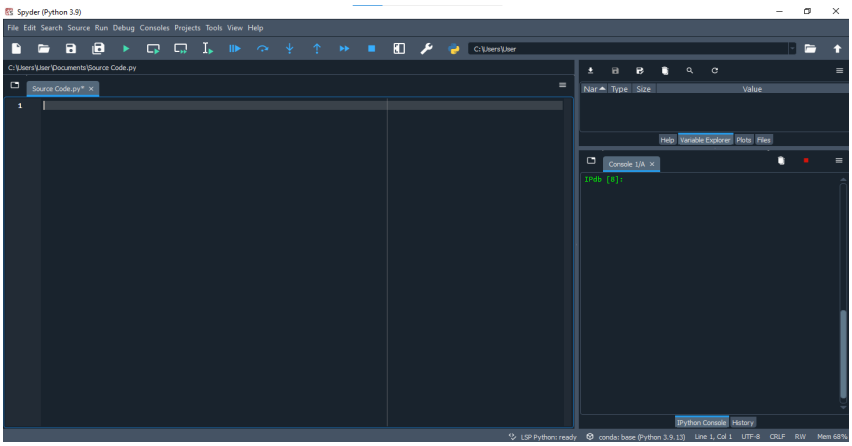


3. Make sure that the Excel file containing your sales and expense data is saved in the same directory as the program source code.

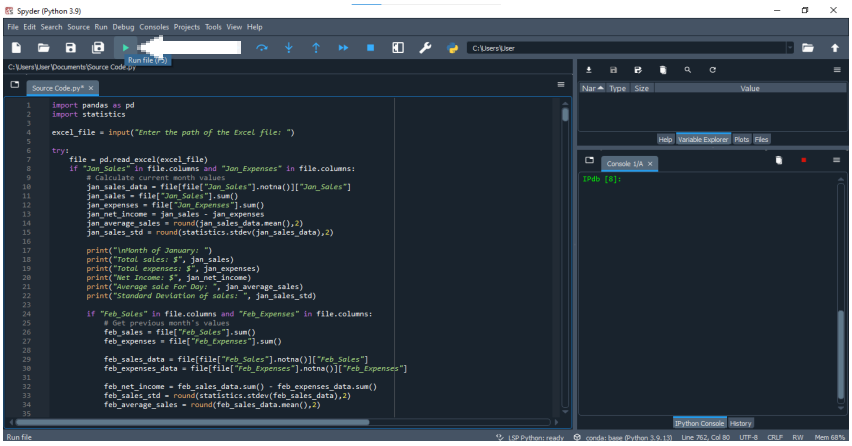


Day	Jan_Sales	Jan_Expenses	Feb_Sales	Feb_Expenses	Mar_Sale	Mar_Expenses	Apr_Sales	Apr_Expenses	May_Sale	May_Expenses	Jun_Sales	Jun_Expenses	Jul_Sales	Jul_Expenses	Aug_Sale	Aug_
1	600	230	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
2	450	60	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
3	900	156	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
4	500	231	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
5	1500	35	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
6	2000	153	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
7	150	123	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
8	500	156	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
9	300	133	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
10	450	1	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
11	2500	1234	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
12	700	353	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
13	600	65	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
14	850	45	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
15	1200	3345	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
16	300	21	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
17	650	121	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
18	350	54	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
19	45	54	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
20	650	548	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
21	690	874	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
22	500	213	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
23	3212	2132	500	300	450	200	600	300	1000	50	500	100	1500	500	100	
24	321	1	500	300	450	200	600	300	1000	50	500	100	1500	500	100	

4. Launch the Python IDE or command prompt and navigate to the directory where the program's source code is located.

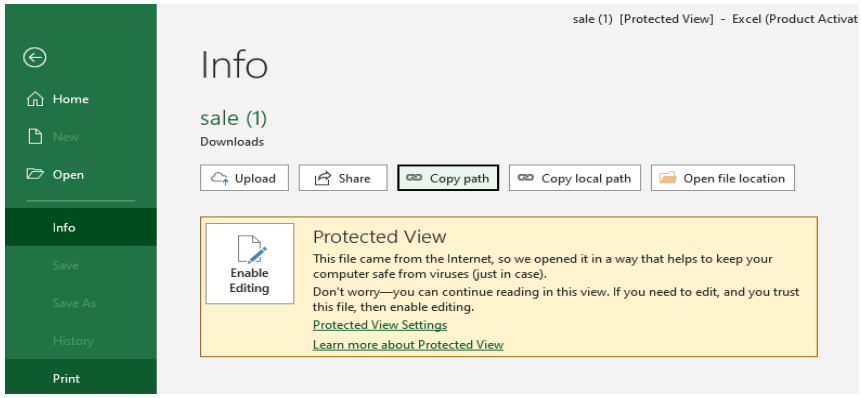


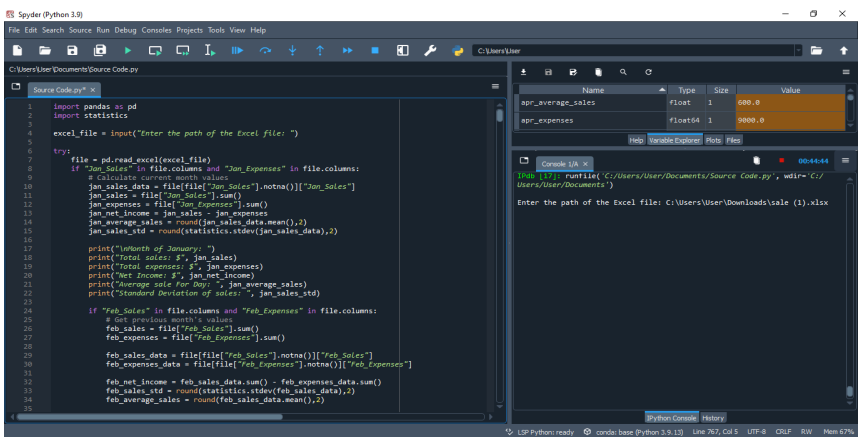
5. Run the program by executing the Python script from the command line or by using the "Run" command in your IDE.



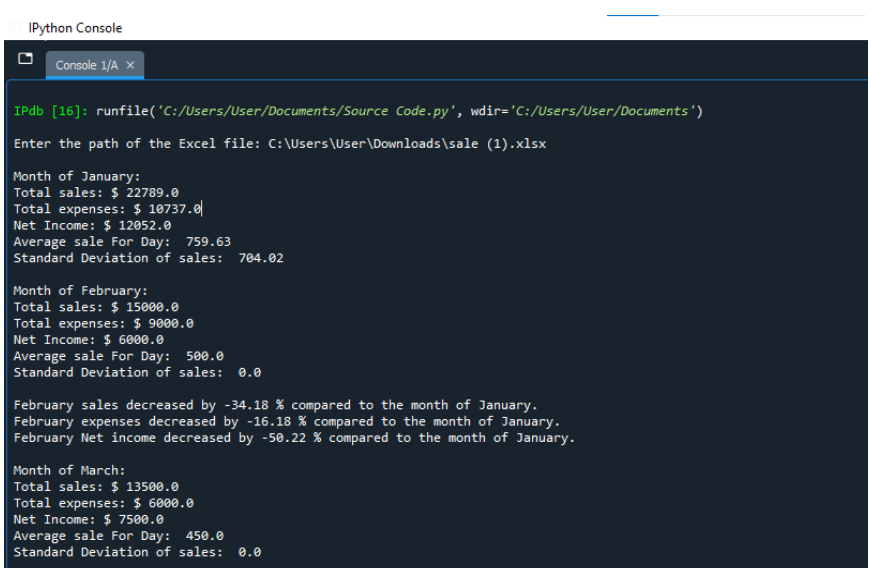
```
import pandas as pd
import statistics

excel_file = input("Enter the path of the Excel file: ")

try:
    file = pd.read_excel(excel_file)
    if "Jan_Sales" in file.columns and "Jan_Expenses" in file.columns:
        # Calculate current month values
        jan_sales_data = file[file["Jan_Sales"].notna()]["Jan_Sales"]
        jan_sales = file["Jan_Sales"].sum()
        jan_expenses = file["Jan_Expenses"].sum()
        jan_net_income = jan_sales - jan_expenses
        jan_average_sales = round(jan_sales_data.mean(),2)
        jan_sales_std = round(statistics.stdev(jan_sales_data),2)

        print("\nMonth of January: ")
        print("Total sales: $", jan_sales)
        print("Total expenses: $", jan_expenses)
        print("Net Income: $", jan_net_income)
        print("Average sale For Day:  ", jan_average_sales)
        print("Standard Deviation of sales: ", jan_sales_std)

        if "Feb_Sales" in file.columns and "Feb_Expenses" in file.columns:
            # Get previous month's values
            feb_sales = file["Feb_Sales"].sum()
            feb_expenses = file["Feb_Expenses"].sum()

            feb_sales_data = file[file["Feb_Sales"].notna()]["Feb_Sales"]
            feb_expenses_data = file[file["Feb_Expenses"].notna()]["Feb_Expenses"]

            feb_net_income = feb_sales_data.sum() - feb_expenses_data.sum()
            feb_sales_std = round(statistics.stdev(feb_sales_data),2)
            feb_average_sales = round(feb_sales_data.mean(),2)
```

6. The program will prompt you to enter the name of the Excel file that contains your sales and expense data. Make sure to provide the correct file name, including the file extension (e.g., "sales.xlsx").

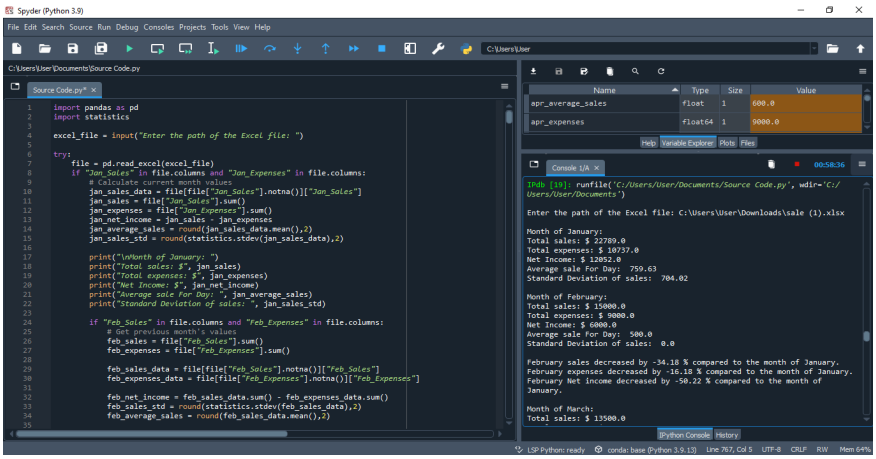




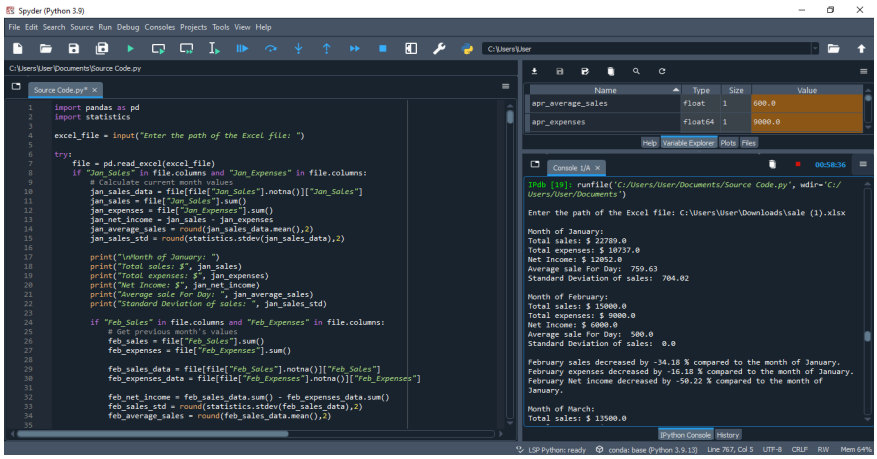
7. Once you've entered the file name, the program will attempt to read the data from the Excel file and perform the necessary calculations for standard deviation and monthly trend analysis.



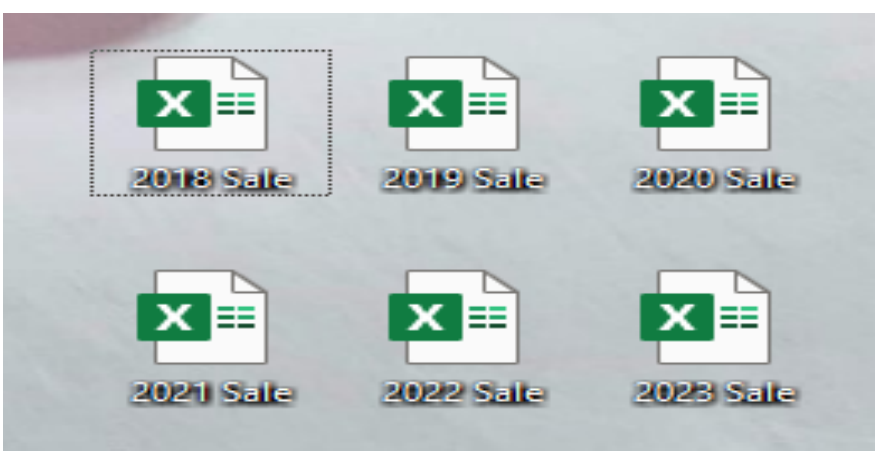
8. If the program runs without any errors, it will display the results directly on the screen.



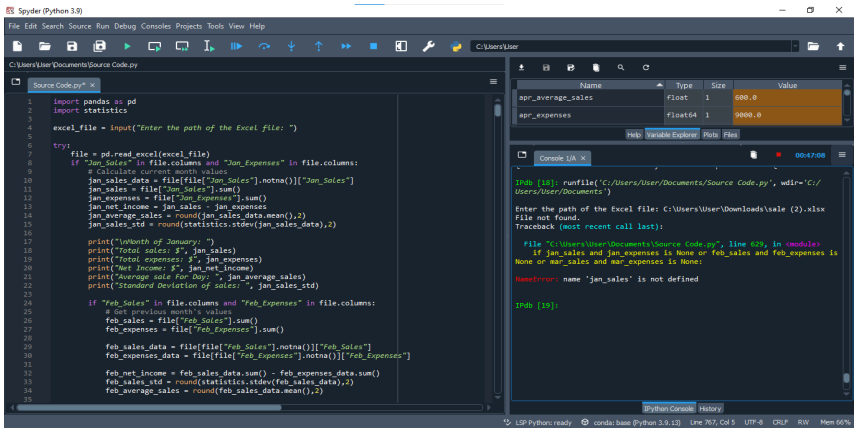
9. Analyze the results and use them to gain insights into your business's performance. Identify any significant deviations from the average, understand the trends, and make informed decisions based on the findings.



10. If needed, you can repeat the process with different Excel files.



11. If an error occurs during the program execution, the program may display an error message. In such cases, you can try running the program again to see if the error was a temporary issue. Make sure to double-check that the Excel file is in the correct format and located in the same directory as the program source code.



12. If the error persists or you need further assistance, consider checking the program's documentation or contacting the program's developers for support.

You can contact the developer thru email:

2021-100211@rtu.edu.ph - Aniñon, Angelica V.
2021-100241@rtu.edu.ph - Asuncion, Kylakrylle P.
2021-100476@rtu.edu.ph - Casas, Christiffany B.
2021-100607@rtu.edu.ph - Descutido, Eunice C.
2021-110719@rtu.edu.ph - Go, Christopher S.
2021-110744@rtu.edu.ph - Nonoy, Bernadette R.
2021-110788@rtu.edu.ph - Sta. Maria, Alexandria D.

Developer Organization

System Analyst:

Go, Christopher

Aniñon, Angelica V.

Sta. Maria, Alexandria D.

Program Designer:

Go, Christopher

Computer Programmer:

Go, Christopher S.

Sta. Maria, Alexandria D.

Researchers:

Aniñon, Angelica V.

Asuncion, Kylakrylle P.

Casas, Christiffany B.

Descutido, Eunice C.

Nonoy, Bernadette R.

Sta. Maria, Alexandria D.

Documentation:

Aniñon, Angelica V.

Source Code

```
import pandas as pd
import statistics

excel_file = input("Enter the path of the Excel file: ")

try:
    file = pd.read_excel(excel_file)
        if "Jan_Sales" in file.columns and "Jan_Expenses" in
file.columns:
        # Calculate current month values
                                                jan_sales_data     =
file[file["Jan_Sales"].notna()]["Jan_Sales"]
        jan_sales = file["Jan_Sales"].sum()
        jan_expenses = file["Jan_Expenses"].sum()
        jan_net_income = jan_sales - jan_expenses
        jan_average_sales = round(jan_sales_data.mean(),2)
                                                 jan_sales_std     =
round(statistics.stdev(jan_sales_data),2)

        print("\nMonth of January: ")
        print("Total sales: $", jan_sales)
        print("Total expenses: $", jan_expenses)
        print("Net Income: $", jan_net_income)
        print("Average sale For Day: ", jan_average_sales)
        print("Standard Deviation of sales: ", jan_sales_std)

           if "Feb_Sales" in file.columns and "Feb_Expenses" in
file.columns:
            # Get previous month's values
            feb_sales = file["Feb_Sales"].sum()
            feb_expenses = file["Feb_Expenses"].sum()

                                                  feb_sales_data    =
file[file["Feb_Sales"].notna()]["Feb_Sales"]
                                                feb_expenses_data    =
file[file["Feb_Expenses"].notna()]["Feb_Expenses"]

                       feb_net_income  =  feb_sales_data.sum()  -
feb_expenses_data.sum()
                                                   feb_sales_std    =
round(statistics.stdev(feb_sales_data),2)
            feb_average_sales = round(feb_sales_data.mean(),2)

            # Calculate monthly changes
            feb_sales_change = feb_sales - jan_sales
```

```
            feb_expenses_change = feb_expenses - jan_expenses
              feb_net_income_change = feb_sales - feb_expenses -
(jan_sales - jan_expenses)

            # Calculate percentage changes
                                feb_sales_percentage_change   =
round((feb_sales_change / jan_sales) * 100,2)
                              feb_expenses_percentage_change   =
round((feb_expenses_change / jan_expenses) * 100,2)
                             feb_net_income_percentage_change   =
round((feb_net_income_change  /  (jan_sales  -  jan_expenses))  *
100,2)

            print("\nMonth of February:")
            print("Total sales: $", feb_sales)
            print("Total expenses: $", feb_expenses)
            print("Net Income: $", feb_net_income)
            print("Average sale For Day: ", feb_average_sales)
                       print("Standard  Deviation  of  sales:  ",
feb_sales_std)


            if feb_sales_change > 0:
                         print("\nFebruary  sales  increased  by",
feb_sales_percentage_change,"%   compared   to   the   month   of
January.")
            elif feb_sales_change < 0:
                         print("\nFebruary  sales  decreased  by",
feb_sales_percentage_change,"%   compared   to   the   month   of
January.")
            else:
                      print("\nFebruary sales remained the same
compared to the month of January.")

            if feb_expenses_change > 0:
                        print("February  expenses  increased  by",
feb_expenses_percentage_change,"%  compared  to  the  month  of
January.")
            elif feb_expenses_change < 0:
                        print("February  expenses  decreased  by",
feb_expenses_percentage_change,"%  compared  to  the  month  of
January.")
            else:
                     print("February expenses remained the same
compared to the month of January.")

            if feb_net_income_change > 0:
```

```
                    print("February Net income increased by",
feb_net_income_percentage_change,"% compared to the month of
January.")
            elif feb_net_income_change < 0:
                    print("February Net income decreased by",
feb_net_income_percentage_change,"% compared to the month of
January.")
            else:
                    print("February Net income remained the same
compared to the month of January.")


             if "Mar_Sales" in file.columns and "Mar_Expenses" in
file.columns:
                # Get previous month's values
                mar_sales = file["Mar_Sales"].sum()
                mar_expenses = file["Mar_Expenses"].sum()

                                                mar_sales_data =
file[file["Mar_Sales"].notna()]["Mar_Sales"]
                                              mar_expenses_data =
file[file["Mar_Expenses"].notna()]["Mar_Expenses"]

                        mar_net_income = mar_sales_data.sum() -
mar_expenses_data.sum()
                                                 mar_sales_std =
round(statistics.stdev(mar_sales_data),2)
                                              mar_average_sales =
round(mar_sales_data.mean(),2)

                # Calculate monthly changes
                mar_sales_change = mar_sales - feb_sales
                           mar_expenses_change = mar_expenses -
feb_expenses
                 mar_net_income_change = mar_sales - mar_expenses
- (feb_sales - feb_expenses)

                # Calculate percentage changes
                                   mar_sales_percentage_change =
round((mar_sales_change / feb_sales) * 100,2)
                                 mar_expenses_percentage_change =
round((mar_expenses_change / feb_expenses) * 100,2)
                               mar_net_income_percentage_change =
round((mar_net_income_change / (feb_sales - feb_expenses)) *
100,2)

                print("\nMonth of March:")
```

```
                print("Total sales: $", mar_sales)
                print("Total expenses: $", mar_expenses)
                print("Net Income: $", mar_net_income)
                              print("Average sale For Day: ",
mar_average_sales)
                        print("Standard Deviation of sales: ",
mar_sales_std)

                if mar_sales_change > 0:
                            print("\nMarch sales increased by",
mar_sales_percentage_change, "compared to the month of
February.")
                elif mar_sales_change < 0:
                            print("\nMarch sales decreased by",
mar_sales_percentage_change, "compared to the month of
February.")
                else:
                          print("\nMarch sales remained the same
compared to the month of February.")

                if mar_expenses_change > 0:
                            print("March expenses increased by",
mar_expenses_percentage_change, "compared to the month of
February.")
                elif mar_expenses_change < 0:
                            print("March expenses decreased by",
mar_expenses_percentage_change, "compared to the month of
February.")
                else:
                        print("March expenses remained the same
compared to the month of February.")

                if mar_net_income_change > 0:
                          print("March Net income increased by",
mar_net_income_percentage_change, "compared to the month of
February.")
                elif mar_net_income_change < 0:
                          print("March Net income decreased by",
mar_net_income_percentage_change, "compared to the month of
February.")
                else:
                       print("March Net income remained the same
compared to the month of February.")


                            if "Apr_Sales" in file.columns and
"Apr_Expenses" in file.columns:
```

```
                    # Get previous month's values
                    apr_sales = file["Apr_Sales"].sum()
                    apr_expenses = file["Apr_Expenses"].sum()

                                                    apr_sales_data  =
file[file["Apr_Sales"].notna()]["Apr_Sales"]
                                                  apr_expenses_data  =
file[file["Apr_Expenses"].notna()]["Apr_Expenses"]

                         apr_net_income = apr_sales_data.sum() -
apr_expenses_data.sum()
                                                     apr_sales_std  =
round(statistics.stdev(apr_sales_data),2)
                                                  apr_average_sales  =
round(apr_sales_data.mean(),2)

                    # Calculate monthly changes
                    apr_sales_change = apr_sales - mar_sales
                            apr_expenses_change = apr_expenses -
mar_expenses
                             apr_net_income_change = apr_sales -
apr_expenses - (mar_sales - mar_expenses)

                    # Calculate percentage changes
                                     apr_sales_percentage_change  =
round((apr_sales_change / mar_sales) * 100,2)
                                  apr_expenses_percentage_change  =
round((apr_expenses_change / mar_expenses) * 100,2)
                                 apr_net_income_percentage_change  =
round((apr_net_income_change  /  (mar_sales  -  mar_expenses))  *
100,2)

                    print("\nMonth of April:")
                    print("Total sales: $", apr_sales)
                    print("Total expenses: $", apr_expenses)
                    print("Net Income: $", apr_net_income)
                                print("Average  sale  For  Day:  ",
apr_average_sales)
                           print("Standard Deviation of sales: ",
apr_sales_std)


                    if apr_sales_change > 0:
                             print("\nApril sales increased by",
apr_sales_percentage_change,"% compared to the month of March.")
                    elif apr_sales_change < 0:
```

```
                            print("\nApril sales decreased by", apr_sales_percentage_change,"% compared to the month of March.")
                    else:
                          print("\nApril sales remained the same compared to the month of March.")

                    if apr_expenses_change > 0:
                            print("April expenses increased by", apr_expenses_percentage_change,"% compared to the month of March.")
                    elif apr_expenses_change < 0:
                            print("April expenses decreased by", apr_expenses_percentage_change,"% compared to the month of March.")
                    else:
                          print("April expenses remained the same compared to the month of March.")

                    if apr_net_income_change > 0:
                          print("April Net income increased by", apr_net_income_percentage_change,"% compared to the month of March.")
                    elif apr_net_income_change < 0:
                          print("April Net income decreased by", apr_net_income_percentage_change,"% compared to the month of March.")
                    else:
                            print("April Net income remained the same compared to the month of March.")


                            if "May_Sales" in file.columns and "May_Expenses" in file.columns:
                        # Get previous month's values
                        may_sales = file["May_Sales"].sum()
                                                may_expenses = file["May_Expenses"].sum()

                                                may_sales_data = file[file["May_Sales"].notna()]["May_Sales"]
                                                may_expenses_data = file[file["May_Expenses"].notna()]["May_Expenses"]

                          may_net_income = may_sales_data.sum() - may_expenses_data.sum()
                                                may_sales_std = round(statistics.stdev(may_sales_data),2)
```

```
may_average_sales = round(may_sales_data.mean(),2)

# Calculate monthly changes
may_sales_change = may_sales - apr_sales
may_expenses_change = may_expenses - apr_expenses
may_net_income_change = may_sales - may_expenses - (apr_sales - apr_expenses)

# Calculate percentage changes
may_sales_percentage_change = round((may_sales_change / apr_sales) * 100,2)
may_expenses_percentage_change = round((may_expenses_change / apr_expenses) * 100,2)
may_net_income_percentage_change = round((may_net_income_change / (apr_sales - apr_expenses)) * 100,2)

print("\nMonth of May:")
print("Total sales: $", may_sales)
print("Total expenses: $", may_expenses)
print("Net Income: $", may_net_income)
print("Average sale For Day: ", may_average_sales)
print("Standard Deviation of sales: ", may_sales_std)


if may_sales_change > 0:
print("\nMay sales increased by", may_sales_percentage_change,"% compared to the month of April.")
elif may_sales_change < 0:
print("\nMay sales decreased by", may_sales_percentage_change,"% compared to the month of April.")
else:
print("\nMay sales remained the same compared to the month of April.")

if may_expenses_change > 0:
print("May expenses increased by", may_expenses_percentage_change,"% compared to the month of April.")
elif may_expenses_change < 0:
print("May expenses decreased by", may_expenses_percentage_change,"% compared to the month of April.")
```

```
                        else:
                                print("May expenses remained the
same compared to the month of April.")

                        if may_net_income_change > 0:
                             print("May Net income increased by",
may_net_income_percentage_change,"%  compared  to  the  month  of
April.")
                        elif may_net_income_change < 0:
                             print("May Net income decreased by",
may_net_income_percentage_change,"%  compared  to  the  month  of
April.")
                        else:
                               print("May Net income remained the
same compared to the month of April.")

                               if "Jun_Sales" in file.columns and
"Jun_Expenses" in file.columns:
                            # Get previous month's values
                            jun_sales = file["Jun_Sales"].sum()
                                                  jun_expenses  =
file["Jun_Expenses"].sum()

                                                jun_sales_data  =
file[file["Jun_Sales"].notna()]["Jun_Sales"]
                                             jun_expenses_data  =
file[file["Jun_Expenses"].notna()]["Jun_Expenses"]

                                                jun_net_income  =
jun_sales_data.sum() - jun_expenses_data.sum()
                                                 jun_sales_std  =
round(statistics.stdev(jun_sales_data),2)
                                             jun_average_sales  =
round(jun_sales_data.mean(),2)

                            # Calculate monthly changes
                                  jun_sales_change = jun_sales -
may_sales
                             jun_expenses_change = jun_expenses -
may_expenses
                              jun_net_income_change = jun_sales -
jun_expenses - (may_sales - may_expenses)

                            # Calculate percentage changes
                                   jun_sales_percentage_change =
round((jun_sales_change / may_sales) * 100,2)
```

```
                                jun_expenses_percentage_change =
round((jun_expenses_change / may_expenses) * 100,2)
                              jun_net_income_percentage_change =
round((jun_net_income_change  /  (may_sales  -  may_expenses))  *
100,2)

                            print("\nMonth of June:")
                            print("Total sales: $", jun_sales)
                                      print("Total expenses: $",
jun_expenses)
                                          print("Net Income: $",
jun_net_income)
                                 print("Average sale For Day: ",
jun_average_sales)
                              print("Standard Deviation of sales:
", jun_sales_std)


                            if jun_sales_change > 0:
                                    print("\nJune sales increased
by", jun_sales_percentage_change,"% compared to the month of
May.")
                            elif jun_sales_change < 0:
                                    print("\nJune sales decreased
by", jun_sales_percentage_change,"% compared to the month of
May.")
                            else:
                                 print("\nJune sales remained the
same compared to the month of May.")

                            if jun_expenses_change > 0:
                                   print("June expenses increased
by", jun_expenses_percentage_change,"% compared to the month of
May.")
                            elif jun_expenses_change < 0:
                                   print("June expenses decreased
by", jun_expenses_percentage_change,"% compared to the month of
May.")
                            else:
                                    print("June expenses remained
the same compared to the month of May.")

                            if jun_net_income_change > 0:
                                 print("June Net income increased
by", jun_net_income_percentage_change,"% compared to the month
of May.")
                            elif jun_net_income_change < 0:
```

```
                                    print("June Net income decreased by", jun_net_income_percentage_change,"% compared to the month of May.")
                                else:
                                        print("June Net income remained the same compared to the month of May.")

                                if "Jul_Sales" in file.columns and "Jul_Expenses" in file.columns:
                                    # Get previous month's values
                                                                jul_sales = file["Jul_Sales"].sum()
                                                                jul_expenses = file["Jul_Expenses"].sum()

                                                                jul_sales_data = file[file["Jul_Sales"].notna()]["Jul_Sales"]
                                                                jul_expenses_data = file[file["Jul_Expenses"].notna()]["Jul_Expenses"]

                                                                jul_net_income = jul_sales_data.sum() - jul_expenses_data.sum()
                                                                jul_sales_std = round(statistics.stdev(jul_sales_data),2)
                                                                jul_average_sales = round(jul_sales_data.mean(),2)

                                    # Calculate monthly changes
                                        jul_sales_change = jul_sales - jun_sales
                                                                jul_expenses_change = jul_expenses - jun_expenses
                                                                jul_net_income_change = jul_sales - jul_expenses - (jun_sales - jun_expenses)

                                    # Calculate percentage changes
                                        jul_sales_percentage_change = round((jul_sales_change / jun_sales) * 100,2)
                                    jul_expenses_percentage_change = round((jul_expenses_change / jun_expenses) * 100,2)
                                    jul_net_income_percentage_change = round((jul_net_income_change / (jun_sales - jun_expenses)) * 100,2)

                                    print("\nMonth of July:")
                                                                print("Total sales: $", jul_sales)
```

```
                                        print("Total expenses: $",
jul_expenses)
                                            print("Net Income: $",
jul_net_income)
                                    print("Average sale For Day: ",
jul_average_sales)
                                       print("Standard Deviation of
sales: ", jul_sales_std)


                                if jul_sales_change > 0:
                                             print("\nJuly sales
increased by", jul_sales_percentage_change,"% compared to the
month of June.")
                                elif jul_sales_change < 0:
                                             print("\nJuly sales
decreased by", jul_sales_percentage_change,"% compared to the
month of June.")
                                else:
                                     print("\nJuly sales remained
the same compared to the month of June.")

                                if jul_expenses_change > 0:
                                            print("July expenses
increased by", jul_expenses_percentage_change,"% compared to the
month of June.")
                                elif jul_expenses_change < 0:
                                            print("July expenses
decreased by", jul_expenses_percentage_change,"% compared to the
month of June.")
                                else:
                                            print("July expenses
remained the same compared to the month of June.")

                                if jul_net_income_change > 0:
                                           print("July Net income
increased by", jul_net_income_percentage_change,"% compared to
the month of June.")
                                elif jul_net_income_change < 0:
                                           print("July Net income
decreased by", jul_net_income_percentage_change,"% compared to
the month of June.")
                                else:
                                           print("July Net income
remained the same compared to the month of June.")
```

```
                                    if "Aug_Sales" in file.columns
and "Aug_Expenses" in file.columns:
                                            # Get previous month's
values
                                                    aug_sales =
file["Aug_Sales"].sum()
                                                    aug_expenses =
file["Aug_Expenses"].sum()

                                                    aug_sales_data =
file[file["Aug_Sales"].notna()]["Aug_Sales"]
                                                    aug_expenses_data =
file[file["Aug_Expenses"].notna()]["Aug_Expenses"]

                                                    aug_net_income =
aug_sales_data.sum() - aug_expenses_data.sum()
                                                    aug_sales_std =
round(statistics.stdev(aug_sales_data),2)
                                                    aug_average_sales =
round(aug_sales_data.mean(),2)

                                    # Calculate monthly changes
                                    aug_sales_change = aug_sales
- jul_sales
                                            aug_expenses_change =
aug_expenses - jul_expenses
                                            aug_net_income_change =
aug_sales - aug_expenses - (jul_sales - jul_expenses)

                                            # Calculate percentage
changes
                                    aug_sales_percentage_change
= round((aug_sales_change / jul_sales) * 100,2)

aug_expenses_percentage_change = round((aug_expenses_change /
jul_expenses) * 100,2)

aug_net_income_percentage_change = round((aug_net_income_change
/ (jul_sales - jul_expenses)) * 100,2)

                                    print("\nMonth of August:")
                                          print("Total sales: $",
aug_sales)
                                       print("Total expenses: $",
aug_expenses)
                                            print("Net Income: $",
aug_net_income)
```

```
                                        print("Average sale For Day: ", aug_average_sales)
                                        print("Standard Deviation of sales: ", aug_sales_std)


                                    if aug_sales_change > 0:
                                            print("\nAugust sales increased by", aug_sales_percentage_change,"% compared to the month of July.")
                                    elif aug_sales_change < 0:
                                            print("\nAugust sales decreased by", aug_sales_percentage_change,"% compared to the month of July.")
                                    else:
                                            print("\nAugust sales remained the same compared to the month of July.")

                                    if aug_expenses_change > 0:
                                           print("August expenses increased by", aug_expenses_percentage_change,"% compared to the month of July.")
                                       elif aug_expenses_change < 0:
                                           print("August expenses decreased by", aug_expenses_percentage_change,"% compared to the month of July.")
                                    else:
                                           print("August expenses remained the same compared to the month of July.")

                                       if aug_net_income_change > 0:
                                          print("August Net income increased by", aug_net_income_percentage_change,"% compared to the month of July.")
                                        elif aug_net_income_change < 0:
                                          print("August Net income decreased by", aug_net_income_percentage_change,"% compared to the month of July.")
                                    else:
                                          print("August Net income remained the same compared to the month of July.")

                                                  if "Sep_Sales" in file.columns and "Sep_Expenses" in file.columns:
```

```
# Get previous month's values
sep_sales = file["Sep_Sales"].sum()
sep_expenses = file["Sep_Expenses"].sum()

sep_sales_data = file[file["Sep_Sales"].notna()]["Sep_Sales"]
sep_expenses_data = file[file["Sep_Expenses"].notna()]["Sep_Expenses"]

sep_net_income = sep_sales_data.sum() - sep_expenses_data.sum()
sep_sales_std = round(statistics.stdev(sep_sales_data),2)
sep_average_sales = round(sep_sales_data.mean(),2)

# Calculate monthly changes
sep_sales_change = sep_sales - aug_sales
sep_expenses_change = sep_expenses - aug_expenses
sep_net_income_change = sep_sales - sep_expenses - (aug_sales - aug_expenses)

# Calculate percentage changes

sep_sales_percentage_change = round((sep_sales_change / aug_sales) * 100,2)

sep_expenses_percentage_change = round((sep_expenses_change / aug_expenses) * 100,2)

sep_net_income_percentage_change = round((sep_net_income_change / (aug_sales - aug_expenses)) * 100,2)

print("\nMonth of September:")
print("Total sales: $", sep_sales)
print("Total expenses: $", sep_expenses)
```

```
                                        print("Net Income: $",
sep_net_income)
                                       print("Average sale For
Day: ", sep_average_sales)
                                               print("Standard
Deviation of sales: ", sep_sales_std)


                                    if sep_sales_change > 0:
                                           print("\nSeptember
sales increased by", sep_sales_percentage_change,"% compared to
the month of August.")
                                     elif sep_sales_change <
0:
                                           print("\nSeptember
sales decreased by", sep_sales_percentage_change,"% compared to
the month of August.")
                                    else:
                                           print("\nSeptember
sales remained the same compared to the month of August.")

                                     if sep_expenses_change >
0:
                                             print("September
expenses   increased   by",   sep_expenses_percentage_change,"%
compared to the month of August.")
                                     elif sep_expenses_change
< 0:
                                             print("September
expenses   decreased   by",   sep_expenses_percentage_change,"%
compared to the month of August.")
                                    else:
                                             print("September
expenses remained the same compared to the month of August.")

                                     if sep_net_income_change
> 0:
                                         print("September Net
income   increased   by",   sep_net_income_percentage_change,"%
compared to the month of August.")
                                                          elif
aug_net_income_change < 0:
                                         print("September Net
income   decreased   by",   sep_net_income_percentage_change,"%
compared to the month of August.")
                                    else:
```

```
                                                print("September Net
income remained the same compared to the month of August.")

                                                  if "Oct_Sales" in
file.columns and "Oct_Expenses" in file.columns:
                                                     # Get previous
month's values
                                                        oct_sales =
file["Oct_Sales"].sum()
                                                     oct_expenses =
file["Oct_Expenses"].sum()

                                                   oct_sales_data =
file[file["Oct_Sales"].notna()]["Oct_Sales"]
                                                oct_expenses_data =
file[file["Oct_Expenses"].notna()]["Oct_Expenses"]

                                                   oct_net_income =
oct_sales_data.sum() - oct_expenses_data.sum()
                                                    oct_sales_std =
round(statistics.stdev(oct_sales_data),2)
                                                oct_average_sales =
round(oct_sales_data.mean(),2)

                                                # Calculate monthly
changes
                                                 oct_sales_change =
oct_sales - sep_sales
                                                oct_expenses_change
= oct_expenses - sep_expenses

oct_net_income_change = oct_sales - oct_expenses - (sep_sales -
sep_expenses)

                                                        # Calculate
percentage changes

oct_sales_percentage_change    =    round((oct_sales_change    /
sep_sales) * 100,2)

oct_expenses_percentage_change  =  round((oct_expenses_change  /
oct_expenses) * 100,2)

oct_net_income_percentage_change  =  round((oct_net_income_change
/ (sep_sales - sep_expenses)) * 100,2)
```

```
                                                    print("\nMonth of
September:")
                                                  print("Total sales:
$", oct_sales)
                                                         print("Total
expenses: $", oct_expenses)
                                                   print("Net Income:
$", oct_net_income)
                                                  print("Average sale
For Day: ", oct_average_sales)
                                                      print("Standard
Deviation of sales: ", oct_sales_std)


                                                  if oct_sales_change
> 0:
                                                     print("\nOctober
sales increased by", oct_sales_percentage_change,"% compared to
the month of September.")
                                                                 elif
oct_sales_change < 0:
                                                     print("\nOctober
sales decreased by", oct_sales_percentage_change,"% compared to
the month of September.")
                                              else:
                                                     print("\nOctober
sales remained the same compared to the month of September.")

                                                                   if
oct_expenses_change > 0:
                                                       print("October
expenses   increased   by",   oct_expenses_percentage_change,"%
compared to the month of September.")
                                                                 elif
oct_expenses_change < 0:
                                                       print("October
expenses   decreased   by",   oct_expenses_percentage_change,"%
compared to the month of September.")
                                              else:
                                                       print("October
expenses remained the same compared to the month of September.")

                                                                   if
oct_net_income_change > 0:
                                                       print("October
Net  income  increased  by",  oct_net_income_percentage_change,"%
compared to the month of September.")
```

```
elif
oct_net_income_change < 0:
print("October
Net income decreased by", oct_net_income_percentage_change,"%
compared to the month of September.")
else:
print("October
Net income remained the same compared to the month of
September.")

if "Nov_Sales" in
file.columns and "Nov_Expenses" in file.columns:
# Get previous
month's values
nov_sales =
file["Nov_Sales"].sum()
nov_expenses =
file["Nov_Expenses"].sum()

nov_sales_data =
file[file["Nov_Sales"].notna()]["Nov_Sales"]

nov_expenses_data =
file[file["Nov_Expenses"].notna()]["Nov_Expenses"]

nov_net_income =
nov_sales_data.sum() - nov_expenses_data.sum()
nov_sales_std =
round(statistics.stdev(nov_sales_data),2)

nov_average_sales = round(nov_sales_data.mean(),2)

# Calculate
monthly changes
nov_sales_change
= nov_sales - oct_sales

nov_expenses_change = nov_expenses - oct_expenses

nov_net_income_change = nov_sales - nov_expenses - (oct_sales -
oct_expenses)

# Calculate
percentage changes

nov_sales_percentage_change = round((nov_sales_change /
oct_sales) * 100,2)
```

```
nov_expenses_percentage_change = round((nov_expenses_change / oct_expenses) * 100,2)

nov_net_income_percentage_change = round((nov_net_income_change / (oct_sales - oct_expenses)) * 100,2)

print("\nMonth of November:")
print("Total sales: $", nov_sales)
print("Total expenses: $", nov_expenses)
print("Net Income: $", nov_net_income)
print("Average sale For Day: ", nov_average_sales)
print("Standard Deviation of sales: ", nov_sales_std)


if nov_sales_change > 0:
print("\nNovember sales increased by", nov_sales_percentage_change,"% compared to the month of October.")
elif nov_sales_change < 0:
print("\nNovember sales decreased by", nov_sales_percentage_change,"% compared to the month of October.")
else:
print("\nNovember sales remained the same compared to the month of October.")

if nov_expenses_change > 0:
print("November expenses increased by", nov_expenses_percentage_change,"% compared to the month of October.")
elif nov_expenses_change < 0:
```

```
print("November expenses decreased by", nov_expenses_percentage_change,"% compared to the month of October.")
                                                else:

print("November expenses remained the same compared to the month of October.")

                                                                if nov_net_income_change > 0:

print("November Net income increased by", nov_net_income_percentage_change,"% compared to the month of October.")
                                                                elif nov_net_income_change < 0:

print("November Net income decreased by", nov_net_income_percentage_change,"% compared to the month of October.")
                                                else:

print("November Net income remained the same compared to the month of October.")

                                                        if "Dec_Sales" in file.columns and "Oct_Expenses" in file.columns:
                                                                # Get previous month's values
                                                                dec_sales = file["Dec_Sales"].sum()
                                                                dec_expenses = file["Dec_Expenses"].sum()


dec_sales_data = file[file["Dec_Sales"].notna()]["Dec_Sales"]

dec_expenses_data = file[file["Dec_Expenses"].notna()]["Dec_Expenses"]


dec_net_income = dec_sales_data.sum() - dec_expenses_data.sum()

dec_sales_std = round(statistics.stdev(dec_sales_data),2)

dec_average_sales = round(dec_sales_data.mean(),2)
```

```
# Calculate monthly changes

dec_sales_change = dec_sales - nov_sales

dec_expenses_change = dec_expenses - nov_expenses

dec_net_income_change = dec_sales - dec_expenses - (nov_sales - nov_expenses)

# Calculate percentage changes

dec_sales_percentage_change = round((dec_sales_change / nov_sales) * 100,2)

dec_expenses_percentage_change = round((dec_expenses_change / nov_expenses) * 100,2)

dec_net_income_percentage_change = round((dec_net_income_change / (nov_sales - nov_expenses)) * 100,2)


print("\nMonth of December:")
print("Total sales: $", dec_sales)
print("Total expenses: $", dec_expenses)
print("Net Income: $", dec_net_income)

print("Average sale For Day: ", dec_average_sales)

print("Standard Deviation of sales: ", dec_sales_std)


if dec_sales_change > 0:

print("\nDecember sales increased by", dec_sales_percentage_change,"% compared to the month of Novemeber.")
elif dec_sales_change < 0:

print("\nDecember sales decreased by",
```

```
dec_sales_percentage_change,"% compared to the month of November.")
                                                    else:
print("\nDecember sales remained the same compared to the month of November.")
                                                                if dec_expenses_change > 0:
print("December expenses increased by", dec_expenses_percentage_change,"% compared to the month of November.")
                                                                elif dec_expenses_change < 0:
print("December expenses decreased by", dec_expenses_percentage_change,"% compared to the month of November.")
                                                    else:
print("December expenses remained the same compared to the month of November.")
                                                                if dec_net_income_change > 0:
print("December Net income increased by", dec_net_income_percentage_change,"% compared to the month of November.")
                                                                elif dec_net_income_change < 0:
print("December Net income decreased by", dec_net_income_percentage_change,"% compared to the month of November.")
                                                    else:
print("December Net income remained the same compared to the month of November.")
                                                else:
print("\nDec_Sales or Dec_Expenses columns not found in the Excel file.")
                                                else:
```

```
                                                    print("\nNov_Sales
or Nov_Expenses columns not found in the Excel file.")

                                        else:
                                                  print("\nSep_Sales or
Sep_Expenses columns not found in the Excel file.")

                                    else:
                                                  print("\nAug_Sales or
Aug_Expenses columns not found in the Excel file.")

                                else:
                                                 print("\nJul_Sales or
Jul_Expenses columns not found in the Excel file.")

                            else:
                                                print("\nJune_Sales or
June_Expenses columns not found in the Excel file.")

                        else:
                                   print("\nMay_Sales or May_Expenses
columns not found in the Excel file.")
                    else:
                           print("\nApr_Sales or Apr_Expenses columns
not found in the Excel file.")
                else:
                       print("\nMar_Sales or Mar_Expenses columns not
found in the Excel file.")

            else:
                 print("\nFeb_Sales or Feb_Expenses columns not found
in the Excel file.")

        else:
             print("\nJan_Sales or Jan_Expenses columns not found in
the Excel file.")


except FileNotFoundError:
    print("File not found.")
except Exception as e:
    print("An error occurred:", str(e))


if jan_sales and jan_expenses is None or feb_sales and
feb_expenses is None or mar_sales and mar_expenses is None:
    print("Warning: missing data for one or more months")
```

```
else:
    quarter_one_sales = jan_sales + feb_sales + mar_sales
        quarter_one_expenses  =  jan_expenses  +  feb_expenses  +
mar_expenses
             quarter_one_net_profit   =   quarter_one_sales   -
quarter_one_expenses
    print("\nQuarter 1 Report")
    print("\nTotal Sales:", quarter_one_sales)
    print("Total Expenses:", quarter_one_expenses)
    print("Net Profit:", quarter_one_net_profit)

if  apr_sales  and  apr_expenses  is  None  or  may_sales  and
may_expenses is None or jun_sales and jun_expenses is None:
    print("Warning: missing data for one or more months")
else:
    quarter_two_sales = apr_sales + may_sales + jun_sales
        quarter_two_expenses  =  apr_expenses  +  may_expenses  +
jun_expenses
             quarter_two_net_profit   =   quarter_two_sales   -
quarter_two_expenses
    print("\nQuarter 2 Report")
    print("\nTotal Sales:", quarter_two_sales)
    print("Total Expenses:", quarter_two_expenses)
    print("Net Profit:", quarter_two_net_profit)

    # Calculate monthly changes
            quarter_two_sales_change   =   quarter_two_sales   -
quarter_one_sales
         quarter_two_expenses_change  =  quarter_two_expenses  -
quarter_one_expenses
       quarter_two_net_income_change  =  quarter_two_net_profit  -
quarter_one_net_profit

    # Calculate percentage changes
                      quarter_two_sales_percentage_change      =
round((quarter_two_sales_change / quarter_one_sales) * 100,2)
                    quarter_two_expenses_percentage_change     =
round((quarter_two_expenses_change    /  quarter_one_expenses)  *
100,2)
                  quarter_two_net_income_percentage_change     =
round(quarter_two_net_income_change /( quarter_one_net_profit)  *
100,2)

if quarter_two_sales_change > 0:
               print("\nQuarter   2   sales   increased   by",
quarter_two_sales_percentage_change,"%  compared  to  the  Quarter
1.")
```

```
elif quarter_two_sales_change < 0:
                print("\nQuarter 2 sales decreased by",
quarter_two_sales_percentage_change,"% compared to the Quarter
1.")
else:
     print("\nQuarter 2 sales remained the same compared to the
Quarter 1.")

if quarter_two_expenses_change > 0:
               print("Quarter 3 expenses increased by",
quarter_two_expenses_percentage_change,"% compared to the
Quarter 1.")
elif quarter_two_expenses_change< 0:
               print("Quarter 2 expenses decreased by",
quarter_two_expenses_percentage_change,"% compared to the
Quarter 1.")
else:
     print("Quarter 2 expenses remained the same compared to the
Quarter 1.")

if quarter_two_net_income_change > 0:
             print("Quarter 2 Net income increased by",
quarter_two_net_income_percentage_change,"% compared to the
Quarter 1.")
elif quarter_two_net_income_change  < 0:
             print("Quarter 2 Net income decreased by",
quarter_two_net_income_percentage_change,"% compared to the
Quarter 1.")
else:
     print("Quarter 2 Net income remained the same compared to
the Quarter 1.")


if jul_sales and jul_expenses is None or aug_sales and
aug_expenses is None or sep_sales and sep_expenses is None:
    print("Warning: missing data for one or more months")
else:
    quarter_three_sales = jul_sales + aug_sales + sep_sales
       quarter_three_expenses = jul_expenses + aug_expenses +
sep_expenses
            quarter_three_net_profit = quarter_three_sales -
quarter_three_expenses
    print("\nQuarter 3 Report")
    print("\nTotal Sales:", quarter_three_sales)
    print("Total Expenses:", quarter_three_expenses)
    print("Net Profit:", quarter_three_net_profit)
```

```
    # Calculate monthly changes
          quarter_three_sales_change   =   quarter_three_sales   -
quarter_two_sales
       quarter_three_expenses_change  =  quarter_three_expenses  -
quarter_two_expenses
     quarter_three_net_income_change = quarter_three_net_profit -
quarter_two_net_profit

    # Calculate percentage changes
                      quarter_three_sales_percentage_change      =
round((quarter_three_sales_change / quarter_two_sales) * 100,2)
                    quarter_three_expenses_percentage_change     =
round((quarter_three_expenses_change   /  quarter_two_expenses)  *
100,2)
                  quarter_three_net_income_percentage_change    =
round(quarter_three_net_income_change  /(  quarter_two_net_profit)
* 100,2)

if quarter_three_sales_change > 0:
               print("\nQuarter   3   sales   increased   by",
quarter_three_sales_percentage_change,"% compared to the Quarter
2.")
elif quarter_three_sales_change < 0:
               print("\nQuarter   3   sales   decreased   by",
quarter_three_sales_percentage_change,"% compared to the Quarter
2.")
else:
     print("\nQuarter 3 sales remained the same compared to the
Quarter 2.")

if quarter_three_expenses_change > 0:
              print("Quarter   3   expenses   increased   by",
quarter_three_expenses_percentage_change,"%   compared   to   the
Quarter 2.")
elif quarter_three_expenses_change< 0:
              print("Quarter   3   expenses   decreased   by",
quarter_three_expenses_percentage_change,"%   compared   to   the
Quarter 2.")
else:
     print("Quarter 3 expenses remained the same compared to the
Quarter 2.")

if quarter_three_net_income_change > 0:
            print("Quarter   3   Net   income   increased   by",
quarter_three_net_income_percentage_change,"%  compared  to  the
Quarter 2.")
elif quarter_three_net_income_change  < 0:
```

```
            print("Quarter 3 Net income decreased by",
quarter_three_net_income_percentage_change,"% compared to the
Quarter 2.")
else:
     print("Quarter 3 Net income remained the same compared to
the Quarter 2.")


if oct_sales and oct_expenses is None or nov_sales and
nov_expenses is None or dec_sales and dec_expenses is None:
    print("Warning: missing data for one or more months")
else:
    quarter_four_sales = oct_sales + nov_sales + dec_sales
        quarter_four_expenses = oct_expenses + nov_expenses +
dec_expenses
            quarter_four_net_profit = quarter_four_sales -
quarter_four_expenses
    print("\nQuarter 4 Report")
    print("\nTotal Sales:", quarter_four_sales)
    print("Total Expenses:", quarter_four_expenses)
    print("Net Profit:", quarter_four_net_profit)

    # Calculate monthly changes
            quarter_four_sales_change = quarter_four_sales -
quarter_three_sales
        quarter_four_expenses_change = quarter_four_expenses -
quarter_three_expenses
     quarter_four_net_income_change = quarter_four_net_profit -
quarter_three_net_profit

    # Calculate percentage changes
                     quarter_four_sales_percentage_change =
round((quarter_four_sales_change / quarter_three_sales) * 100,2)
                    quarter_four_expenses_percentage_change =
round((quarter_four_expenses_change / quarter_three_expenses) *
100,2)
                  quarter_four_net_income_percentage_change =
round(quarter_four_net_income_change /(
quarter_three_net_profit) * 100,2)

if quarter_four_sales_change > 0:
               print("\nQuarter 4 sales increased by",
quarter_four_sales_percentage_change,"% compared to the Quarter
3.")
elif quarter_four_sales_change < 0:
```

```
                print("\nQuarter 4 sales decreased by",
quarter_four_sales_percentage_change,"% compared to the Quarter
3.")
else:
     print("\nQuarter 4 sales remained the same compared to the
Quarter 3.")

if quarter_four_expenses_change > 0:
               print("Quarter 4 expenses increased by",
quarter_four_expenses_percentage_change,"% compared to the
Quarter 3.")
elif quarter_four_expenses_change< 0:
               print("Quarter 4 expenses decreased by",
quarter_four_expenses_percentage_change,"% compared to the
Quarter 3.")
else:
     print("Quarter 4 expenses remained the same compared to the
Quarter 3.")

if quarter_four_net_income_change > 0:
             print("Quarter 4 Net income increased by",
quarter_four_net_income_percentage_change,"% compared to the
Quarter 3.")
elif quarter_four_net_income_change  < 0:
             print("Quarter 4 Net income decreased by",
quarter_four_net_income_percentage_change,"% compared to the
Quarter 3.")
else:
      print("Quarter 4 Net income remained the same compared to
the Quarter 3 .")
```

Bibliography

Barth, M. E. (2015). Financial Accounting Research, Practice, and Financial Accountability. *Abacus*, *51*(4), 499–510. Wiley. https://doi.org/10.1111/abac.12057

Beers, B. (2019). *Determining Risk With Standard Deviation*. Investopedia. https://www.investopedia.com/ask/answers/021915/how-standard-deviation-used-determine-risk.asp

FreshBooks. (2023). *What Is Expense Tracking and How Can It Help Your Business?* FreshBooks. https://www.freshbooks.com/hub/accounting/expense-tracking

Henricks, M. (2022). *Risk Management Experts Break Down Standard Deviation*. Business Class: Trends and Insights | American Express. https://www.americanexpress.com/en-us/business/trends-and-insights/articles/risk-management-experts-break-down-standard-deviation/

Nicholls, J. A. (2020). Integrating financial, social and environmental accounting. *Sustainability Accounting, Management and Policy Journal*. https://doi.org/10.1108/sampj-01-2019-0030

SumUp Payments Limited. (2011). *Sales – What are sales? | SumUp Invoices*. SumUp - a Better Way to Get Paid. https://www.sumup.com/en-gb/invoices/dictionary/sales/

The Sage Group . (2018, January 5). *What Is Net Profit? Net Profit Calculation | Sage Advice US*. Sage Advice United States of America. https://www.sage.com/en-us/blog/glossary/what-is-net-profit/

Tuovila, A. (2022). *Business Income*. Investopedia. https://www.investopedia.com/terms/b/businessincome.asp

Shweta (2023) What is QuickBooks & How Does It Work?, Forbes.

https://www.forbes.com/advisor/business/software/what-is-quickbooks/

Solutions, I.E. (no date) What are zoho books?, LinkedIn.

https://www.linkedin.com/pulse/what-zoho-books-imtsenterprise?trk=organization-update-content_share-article

Admin (2023) What is Xero, and why should you use xero for your small business?, Financials and Accounting Blog. Available at:

https://www.123financials.com/insights/what-is-xero/

Python compilers: List of top 7 awesome compilers of python program (2023) EDUCBA. Available at:

https://www.educba.com/python-compilers/

Calzon, B. (2022) *Financial graphs and charts – see here 25 business examples*, *datapine*. Available at:

https://www.datapine.com/blog/financial-graphs-and-charts-examples

Bhatt, S. (2023) *Why use python for business analytics?*, *BoTree Technologies*. Available at:

https://www.botreetechnologies.com/blog/python-for-business-analytics/#:~:text=One%20of%20the%20greatest%20advantages,ability%20to%20automate%20and%20replicate.