DATS 6401: Visualization of Complex Data

Spring 2022

Instructor: R. Jafari, Ph.D

Lab #1

Student: Lydia Teinfalt

Initials: LT

Last updated: January 26, 2022

Table of Contents

Abstract

Exploratory data analysis (EDA) methods were utilized to understand two datasets. In this lab, we calculated descriptive statistics of the data -- the mean, the variance and Pearson's correlation coefficient between two variables. After calculating the statistics, we displayed the results to the console. In addition to descriptive statistics, the line and histogram plots were employed to show the relationships in the data. In the first dataset, we generated our own data of random variables X and Y. We were provided with a second dataset from "tute1.csv" which had Sales, Ad Budget, and GDP information from 1981 through 2005 for analysis in this lab.

Introduction

In this lab, a program written in Python was used to calculate three statistics: mean, variance and Pearson's correlation coefficient between two variables. The mean is the average value found in a sample and is calculated using a formula displayed in Figure 1 where you sum up all values in the sample and divide it by the total number of values (via StatisticsHowTo).

$$\bar{x} = (\Sigma x_i) / n$$

Figure 1: Sample Mean

The variance tells you how spread out the data is from the mean in the sample and can be calculated taking the square of standard deviation of a random variable (Wikipedia):

$$\frac{1}{n-1} \sum_{i=1}^{n} \left(Y_i - \bar{Y} \right)^2$$

Figure 2: Variance

The Pearson's correlation coefficient is a numeric representation of the linear interdependence between two variables with values ranging between -1 and 1 (Zach, 2019). If the coefficient is -1 then there is perfectly negative linear relationship between the two variables. If the coefficient is 1 then there is a perfectly positive linear relationship between the two variables. A coefficient value of 0 indicates the two variables are linearly independent. The following formula was used to calculate the coefficient (Zach, 2019):

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

Figure 3: Pearson's Correlation Coefficient

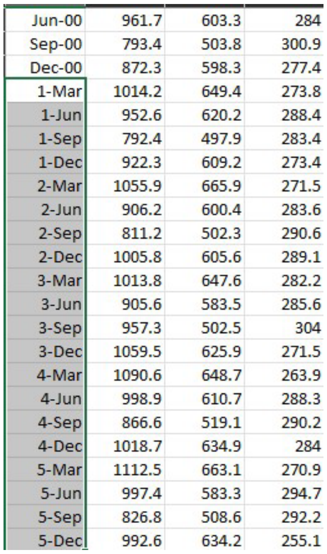
In addition to using statistics to understand the data from a numerical perspective, a line and histogram plot were used as visual aids.

Methodology

Dataset

First data set comprised of two random variables X and Y. By definition, a random variable stores results of a random process like throwing a die (Zach, 2019). In this lab, we created random process 1000 times and stored the outcomes in variables named X and Y. The X variable is normally distributed random variable where the mean is equal to zero and variance equal to 1. The second variable, Y, is also a random variable that should be normally distributed with mean equal to 5 and variance of 2. Variables X and Y were implemented using NumPy package and contained a sample of 1000 numbers of float() data type.

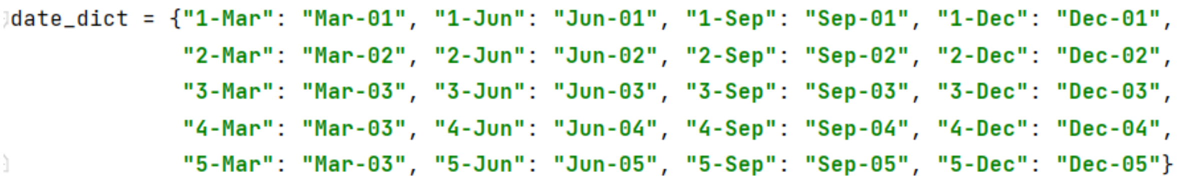
The second dataset was obtained from Professor R. Jafari's Complex Visualization repository on GitHub called "tute1.csv". The Pandas package read_csv method was given the URL directly to the data's raw format from his repository. This dataset had four columns. The first column was a date in the format of abbreviated month plus a dash with a two-digit year. The column did not have a label so it was given a column name of "Date". The remaining columns were Sales, AdBudget and GDP. The Date column contained some bad data. Prior to 2001, all date was stored as abbreviated month and two-digit year such as "Mar-81" but dates starting in 2001 had a bad format of "1-Mar" as highlighted in Figure 4.



Jun-00	961.7	603.3	284
Sep-00	793.4	503.8	300.9
Dec-00	872.3	598.3	277.4
1-Mar	1014.2	649.4	273.8
1-Jun	952.6	620.2	288.4
1-Sep	792.4	497.9	283.4
1-Dec	922.3	609.2	273.4
2-Mar	1055.9	665.9	271.5
2-Jun	906.2	600.4	283.6
2-Sep	811.2	502.3	290.6
2-Dec	1005.8	605.6	289.1
3-Mar	1013.8	647.6	282.2
3-Jun	905.6	583.5	285.6
3-Sep	957.3	502.5	304
3-Dec	1059.5	625.9	271.5
4-Mar	1090.6	648.7	263.9
4-Jun	998.9	610.7	288.3
4-Sep	866.6	519.1	290.2
4-Dec	1018.7	634.9	284
5-Mar	1112.5	663.1	270.9
5-Jun	997.4	583.3	294.7
5-Sep	826.8	508.6	292.2
5-Dec	992.6	634.2	255.1

Figure 4: Poorly Formatted Dates in tute1.csv

To fix this issue, the Pandas package data frame replace() method supplied with a dictionary variable was used to map badly formatted date to correct date. The date_dict variable directs Python to map the Date column if it finds "4-Sept" to replace it with Sept-04. After the mapping, the year portion of the Date was expanded to 4-digit format.



```
date_dict = {"1-Mar": "Mar-01", "1-Jun": "Jun-01", "1-Sep": "Sep-01", "1-Dec": "Dec-01",
             "2-Mar": "Mar-02", "2-Jun": "Jun-02", "2-Sep": "Sep-02", "2-Dec": "Dec-02",
             "3-Mar": "Mar-03", "3-Jun": "Jun-03", "3-Sep": "Sep-03", "3-Dec": "Dec-03",
             "4-Mar": "Mar-03", "4-Jun": "Jun-04", "4-Sep": "Sep-04", "4-Dec": "Dec-04",
             "5-Mar": "Mar-03", "5-Jun": "Jun-05", "5-Sep": "Sep-05", "5-Dec": "Dec-05"}
```

Pearson’s Correlation Coefficient Function

A Python function called corr_coeff was created accepting two input variables x and y. Within the function sample mean of x and y is calculated using NumPy mean method and stored them into two variables. Iterate through each element of the x and uses formula in Figure 3 to return a number representing Pearson’s correlation coefficient. The coefficient has rounded to display to the requirement of 2-digit precision.

Lab Answers

Question 3

a. The sample mean of random variable x is 0.02
b. The sample mean of random variable y is 5.10
c. The sample variance of random variable x is 0.96
d. The sample variance of random variable y is 1.99
e. The sample Pearson's correlation coefficient between X and Y is -0.04

Question 4: Display Line Plot of Variable X and Y

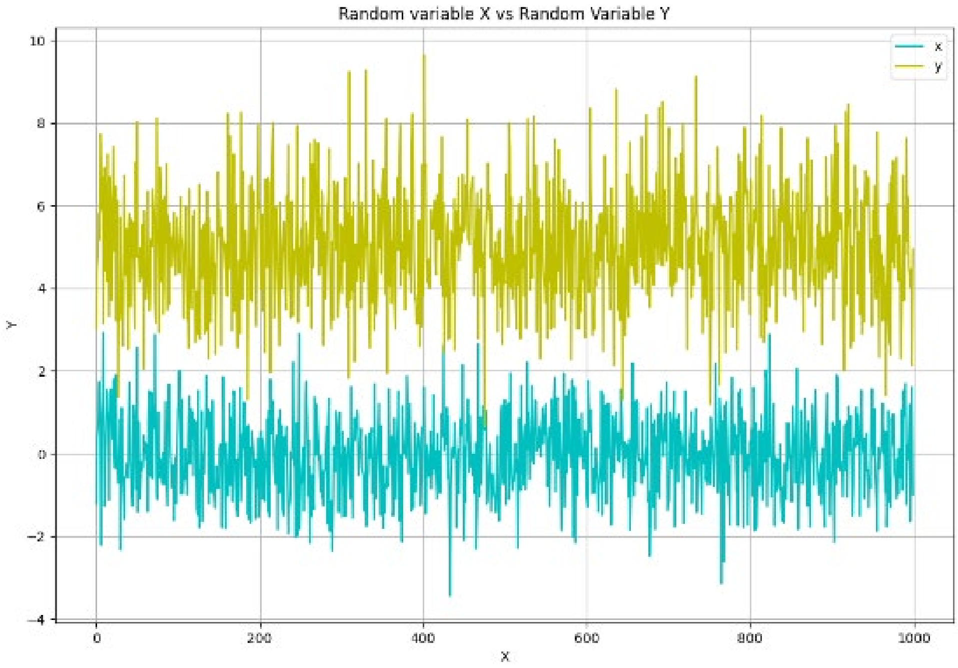


Figure 5: Line Plot of Random Variables X and Y

Question 5: Display Histogram of X and Y

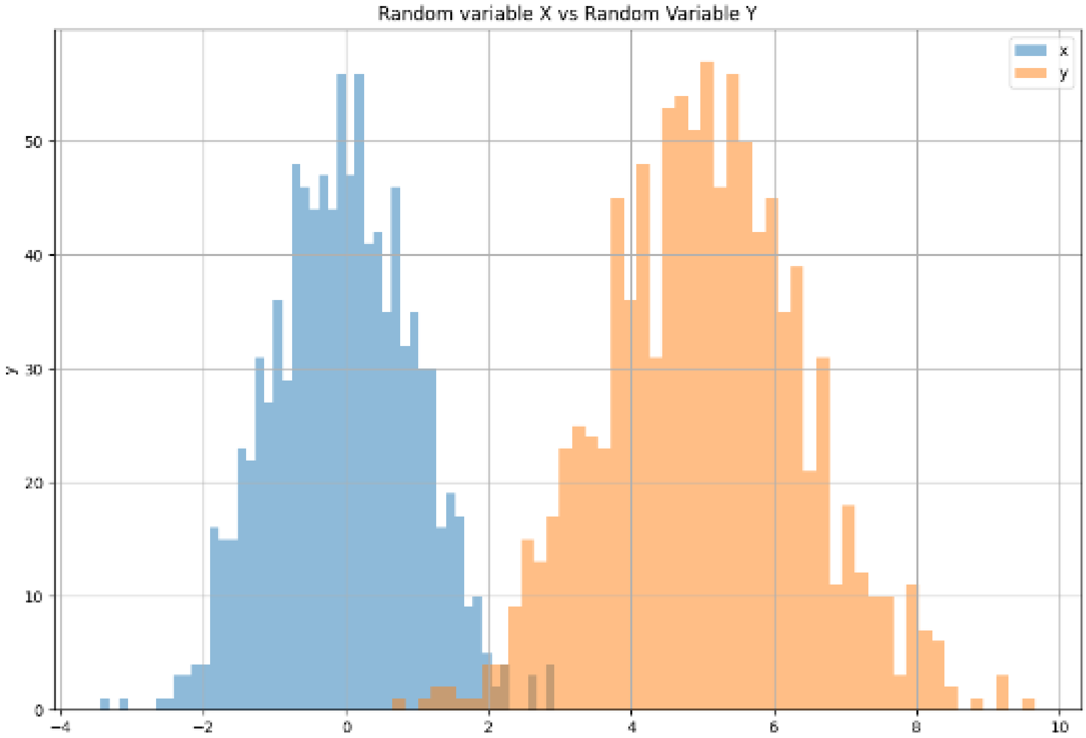


Figure 6: Histogram of X and Y

Question 8

a. The sample Pearson’s correlation coefficient between Sales & AdBudget is: 0.91
b. The sample Pearson’s correlation coefficient between Sales and GDP is: -0.64
c. The sample Pearson’s correlation coefficient between AdBudget and GDP is: -0.77

Question 9: Display Line Plot of Sales, AdBudget, and GCP versus Time

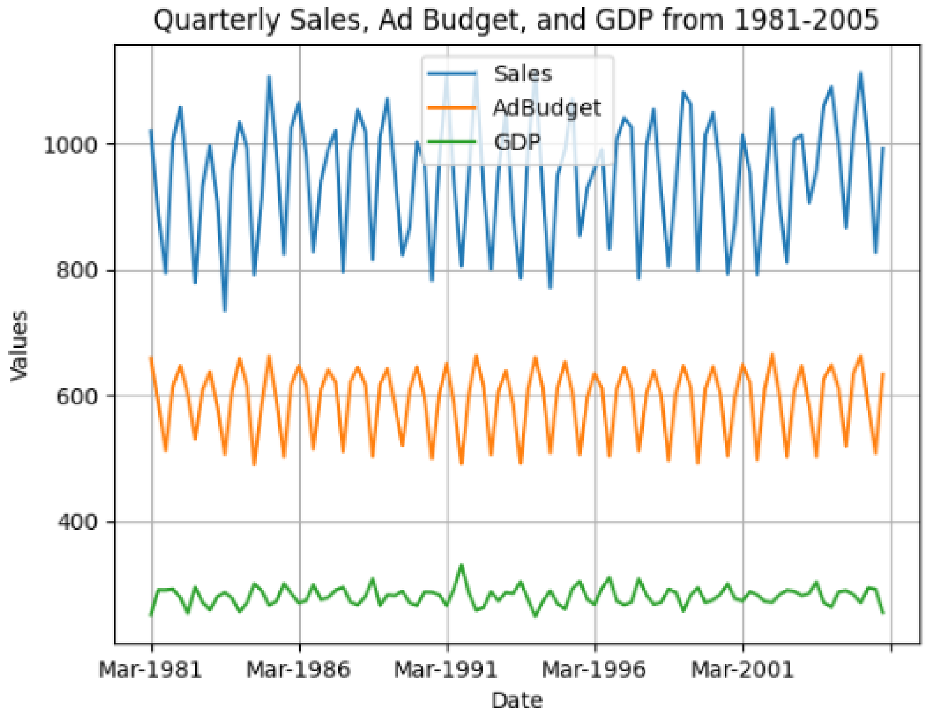


Figure 7: Line Plot of Sales, Ad Budget, GDP

Question 10: Display Histogram of Sales, AdBudget and GDP in a single plot

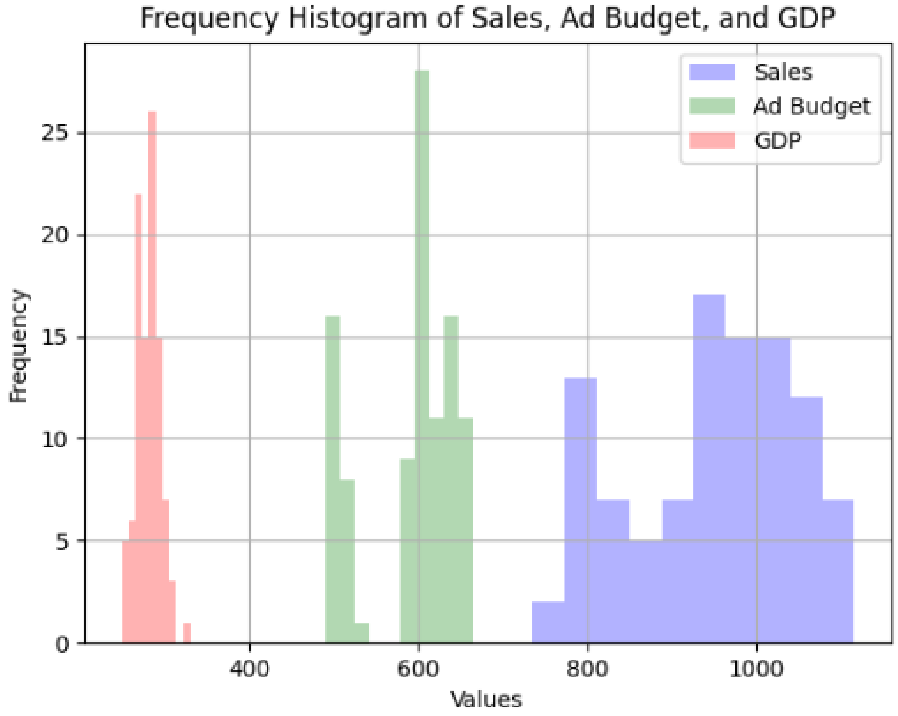


Figure 8: Histogram of Sales, Ad Budget and GDP

Conclusion

For the first dataset, the NumPy package was used to generate sample of random numbers and stored into two variables called x and y. The sample mean, variance and Pearson's correlation coefficient between *x* and *y* are calculated and unique with each execution of the Python code because they are randomly generated. A random seed = 42 was set so we can reproduce results. Variable x contained 1000 numbers, normally distributed, and though we specified it to have a mean of 0 and variance of 1, the actual sample mean was 0.02 with sample variance of 0.96. The *y* variable with 1000 numbers, normally distributed, and we specified mean of 5 and variance of 2 but the random sample had an actual mean of 5.10 and a variance of 1.99. The line plot in figure 6 clearly the difference in means between *x* and *y*. The histogram plot in figure 8 demonstrates that the sample is normally distributed but center line of *y* is higher than center line of *x*. The Pearson's correlation coefficient between x and y was 0.04 which means the two variables have almost no linear relationship.

In the second dataset, the "tute1.csv" was directly read from Professor R. Jafari's GitHub repository. It is a time series exploring Sales, AdBudget and GDP. The Pearson's correlation coefficient between Sales and AdBudget is equal to 0.91 indicating that there is a positive linear relationship. Figure 7's line plot shows this that when there is a peak in Sales, there is a corresponding peak in AdBudget. The same plot line shows that when there is a peak in Sales and AdBudget, there is a valley for GDP. This relationship is confirmed with the Pearson's correlation coefficient being negative -0.64 between Sales and GDP and -0.77 between ADBudget and GDP. The histogram plot with Sales, AdBudget and GDP that the data may not have normal distribution. The Sales data is closest to being a normal distribution. AdBudget looks like it has two peaks and furthest from a normal distribution between Sales and GDP. The GDP histogram appears to be skewed left. The histogram in figure 8 show that the numerical GDP data is greater than numerical data of AdBudget and greater than Sales. We have no context for the numbers representing Sales, AdBudget and GDP – there is no reference for scale and can only guess that GDP means Gross Domestic Product, it would be speculative to interpret the tute1.csv data further.

Appendix A: Code

```
# DATS 6401: Lab 1 (Spring 22)
# Lydia Teinfalt
# 01/26/2022
import matplotlib.pyplot
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Q1-Using the NumPy package in python create a random variable x,
# normally distributed about the mean of zero and variance 1.
# Create a random variable y normally distributed about the mean of 5 and
variance of 2.
# Number of samples for both x and y = 1000.
import pandas

print("-----------------------------------------------------------------------
---")
print("Q1: sing the NumPy package in python create a random variable x,")
print("normally distributed about the mean of zero and variance 1.")
print("Create a random variable y normally distributed about the mean of 5
and variance of 2.")
print("Number of samples for both x and y = 1000.")

print("Formula: standard deviation is equal to the square root of variance.")
np.random.seed(42)
N = 1000
xmean = 0
std_x = np.sqrt(1)

ymean = 5
std_y = np.sqrt(2)

# Random number with normal distribution
x = np.random.normal(xmean, std_x, N)
y = np.random.normal(ymean, std_y, N)

# Q2: Write a python program that calculate the Pearson’s correlation
coefficient between
# two random variables x and y in question 1. Hint: You need to implement the
following formula
print("-----------------------------------------------------------------------
---")
print("Q2: Write a python program that calculates the Pearson's correlation "
      "coefficient between the two random variables.")

def corr_coeff(x, y):
    diff_x = []
    diff_y = []
    r = []
    diff_x_squared = []
    diff_y_squared = []

    mean_x = np.mean(x)
    mean_y = np.mean(y)
    for i in range(len(x)):
        x_hat = (x[i] - mean_x)
        diff_x.append(x_hat)
        diff_x_squared.append(np.square(x_hat))
        y_hat = (y[i]- mean_y)
        r.append((x_hat * y_hat))
        diff_y_squared.append(np.square(y_hat))
    return
(np.sum(r))/(np.sqrt(np.sum(diff_x_squared))*np.sqrt(np.sum(diff_y_squared)))


#x = [1, 2, 3, 4, 5]
#y=  [12, 14, 16, 18, 20]
corr = corr_coeff(x, y)
#print(f"The Pearson's correlation coefficient is {corr:.2f} ")
print("-----------------------------------------------------------------------
---")
print("Q3")
print(f"The sample mean of random variable x is {np.mean(x):.2f}")
print(f"The sample mean of random variable y is {np.mean(y):.2f}")
print(f"The sample variance of random variable x is {np.var(x):.2f}")
print(f"The sample variance of random variable y is {np.var(y):.2f}")
print(f"The sample Pearson's correlation coefficient between X and Y is
{corr:.2f}")

print("-----------------------------------------------------------------------
---")
print("Q4: Line Plot of Variable X x Variable Y")

# line plot
plt.figure(figsize=(12, 8))
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Random variable X vs Random Variable Y")
plt.plot(x, 'c', label='x')
plt.plot(y, 'y', label='y')
plt.legend()
plt.grid()
plt.show()

print("-----------------------------------------------------------------------
---")
print("Q5: Histogram of Variable X x Variable Y")
# histogram
plt.figure(figsize=(12, 8))
plt.hist(x, bins=50, label='x', alpha=.5)
plt.hist(y, bins=50, label='y', alpha=.5)
plt.title("Random variable X vs Random Variable Y")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid()
plt.show()

print("-----------------------------------------------------------------------
---")
print("Q6: Using pandas package in python read the ‘tute1.csv’ dataset. "
      "The set is the timeseries dataset with Sales, AdBudget and GDP
column.")


df = pd.read_csv("https://raw.githubusercontent.com/rjafari979/Complex-Data-
Visualization-/main/tute1.csv", parse_dates=[0])
df.columns = ["Date", "Sales", "AdBudget", "GDP"]

date_dict = {"1-Mar": "Mar-01", "1-Jun": "Jun-01", "1-Sep": "Sep-01", "1-
Dec": "Dec-01",
             "2-Mar": "Mar-02", "2-Jun": "Jun-02", "2-Sep": "Sep-02", "2-
Dec": "Dec-02",
             "3-Mar": "Mar-03", "3-Jun": "Jun-03", "3-Sep": "Sep-03", "3-
Dec": "Dec-03",
             "4-Mar": "Mar-03", "4-Jun": "Jun-04", "4-Sep": "Sep-04", "4-
Dec": "Dec-04",
             "5-Mar": "Mar-03", "5-Jun": "Jun-05", "5-Sep": "Sep-05", "5-
Dec": "Dec-05"}
df = df.replace({"Date": date_dict})
df['Date'] = pd.to_datetime(df['Date'], format="%b-%y").dt.strftime("%b-%Y")
df['Month'] = pd.to_datetime(df['Date']).dt.month
df['Year'] = pd.to_datetime(df['Date']).dt.year


print("-----------------------------------------------------------------------
---")
print("Q7: Find the Pearson's correlation coefficient between Sales,
AdBudget, and GDP.")
print("-----------------------------------------------------------------------
---")
print("Q8: Display message on the console:")
print(f"8a.The sample Pearson’s correlation coefficient between Sales &
AdBudget is:  {corr_coeff(df.Sales, df.AdBudget):.2f}")
print(f"8b.The sample Pearson’s correlation coefficient between Sales and GDP
is: {corr_coeff(df.Sales, df.GDP):.2f}")
print(f"8c.The sample Pearson’s correlation coefficient between AdBudget and
GDP is: {corr_coeff(df.AdBudget, df.GDP):.2f}")
print("-----------------------------------------------------------------------
---")
print("Q9: Display the line plot of Sales, AdBudget, and GDP versus time.")
# Line plot

plt.figure()
df.plot(x='Date', y=['Sales', 'AdBudget', 'GDP'])
plt.xlabel("Date")
plt.title("Quarterly Sales, Ad Budget, and GDP from 1981-2005")
plt.ylabel("Values")
plt.grid()
plt.show()



print("-----------------------------------------------------------------------
---")
print("Q10: Plot the histogram plot of Sales, AdBudget, and GDP on one
plot.")
# histogram plot
```

```
plt.figure()
plt.hist(df['Sales'],color='b',alpha=0.3, label='Sales',
histtype='stepfilled')
plt.hist(df['AdBudget'],color='g',alpha=0.3, label='Ad Budget',
histtype='stepfilled')
plt.hist(df['GDP'],color='r',alpha=0.3, label='GDP', histtype='stepfilled')
plt.title("Frequency Histogram of Sales, Ad Budget, and GDP")
plt.legend()
plt.xlabel("Values")
plt.ylabel("Frequency")
plt.grid()
plt.show()
```

References

Sample Mean: Symbol (X Bar), Definition, Standard Error. (n.d.). Statistics How To. https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/sample-mean/

Wikipedia Contributors. (2019, January 8). Variance. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Variance

Zach. (2019, January 3). Pearson Correlation Coefficient. Statology. https://www.statology.org/pearson-correlation-coefficient/