The George Washington University

Laboratory Two:

Calculating, Visualizing, and Describing Correlations for Time Series Applications

Fernando Zambrano

DATS 6450: Multivariate Modeling

Dr. Reza Jafari

5 February 2020

Abstract:

Correlation is a dimensionless measurement that specifically calculates the strength of a linear relationship between two features and is represented by the correlation coefficient *r*. Calculating the correlation coefficient is not an exhaustive method to fully understand the relationships between features. Visualizing features through a scatterplot gives crucial insight into what type of relationship is associated between them. The relationship may be a linear or non-linear relationship which cannot be determined by the value of *r*. Further insight into time series data begins determining stationarity. This can be determined using histograms, differencing, transformations, and Augmented Dickey-Fuller (ADF) test. The concept of calculating and visualizing correlations is introduced by determining and describing statistical relationships within a small company's financial data.

Introduction:

When facing a data driven research problem or project, understanding and understanding how to describe that data is fundamental. Knowing what patterns and relationships are within the data will allow give insights into what kind of analysis can be performed, or what further data augmentation must be done to get the data in the right format for desired analyses. Undermining the importance of this initial exploration process can lead to incorrect results and insights which is accompanied by unnecessary time and effort to correct such results. In the realm of time-series analysis, which is centered on forecasting and predictive modeling, must also undergo the same initial processes. Time-series analysis is a popular and important form a data analysis since a vast majority of phenomena are naturally related and dependent on time. However, a common mistake is to compare two time series datasets and imply that there is a strong connection between them if they both increase or decrease similarly throughout time. A more egregious mistake is to assign a causal relationship between the two features without taking into account context, and further analysis. This will lead to a shallow interpretation relationship of the data with grossly inaccurate results. Instead, a more constructive and conservative process should be applied to better understand the relationship between features before assigning any type causal relationships. Determining what correlations are represented at an early stage of data exploration can give direction what type of relationships are present and what kind of further analysis is needed to further describe and understand the data at hand. Correlation measures the strength of a liner relationship between two features. The correlation can be determined through correlation coefficient and further described through visual interpretation of scatter plots. The purpose of this study is to demonstrate how to calculate the correlation coefficient through programming in Python and algebraically by hand and comment on the consequences of each method. Additionally, the benefits and restrictions of analyzing correlation through the correlation coefficient and visual interpretation will also be compared. Once correlations have been calculated, determining stationarity will be a crucial next step when creating a forecasting model. Using histograms, differencing, transformations, and Augmented Dickey-Fuller (ADF) test are essential tools to determine the stationarity of data. This will be done by analyzing quarterly financial data (sales, advertising budget, and GDP) of small company over the period of 1981-2005 as a case study.

Theory and Methods:

Correlation measures the strength of a linear relationship between two features. However, before understanding the immediate impacts and uses of correlation some general statistical knowledge needs to be developed. This will help with terminology and explaining mathematical formulas. The first term to understand is expected value. The expected value is the average value of a feature. If there are five observations in a dataset all with the value of 1, then the average is 1. Hence, we *expect* the any observation in that dataset to be 1. Next is variance. Variance is a non-negative measurement to quantify the average spread from the mean of a feature. Variance is calculated by taking the sum of all observations minus the expected valued squared. Understanding variance give way to comprehending standard deviation. Standard deviation is the square root of variance which determines how spread out observations are from the mean. Using the example from before, this dataset has a small (zero) variance because all its values are clustered on the mean. If some of the values were changed to 50 and 100, then the data would have a large variance and standard deviation since more values are spread further away from the mean. Expected value, variance, and standard deviations are the building blocks for calculating correlation coefficient.

Now that some basic statistical understanding has been established, that knowledge can be applied to understand the formula for the correlation coefficient, or *r*. Below is the formula for *r*.

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}.$$

The formula can be separated into two parts. The first part is deals with the calculation of the numerator. The numerator is the covariance between two variables. Covariance determines how linearly related two features are associated. The is done by taking the product of each feature's variance. The issue with why covariance is not used as the principle statistic to demonstrate the strength of a linear relationship is that it maintains the units of its features. This makes it difficult to draw comparisons between two features that have large differences in their spread. In order to compare two features without the drawback of dimensionality, the features need to be normalized. This is the second part, deals with the denominator which is the product of the square root of the squared variance of each feature. This process cancels out the dimensions of each feature and forces the resulting value of *r* to fall between -1 and +1.

Correlation analysis does not have to be strictly formulaic. Correlations can subjectively be determining by visualizing the data on a scatter plot. Instead of visualizing each feature over time, a scatter plot can be used to see if there is a linear relationship between two features. Each scatter plot can be described as having zero, weak, moderate, or strong association. The strength of the correlation depends on how obvious the scatter plot illustrates a line. If the scatter plot is more compact, then there is a strong the association whereas the more spread out signals a weaker association. Furthermore, since correlation can be negative or positive the direction of the slope of a scatter plot corresponds respectively. Negative slope equals a negative correlation, and a positive slope means there is a positive correlation.

Datasets are classified as stationary when its statistics act the same way over time so that a pattern can be recognized. It does not mean that the series does not change over time, but rather the way it changes does not vary over time. This pattern gives insight into how future observations should fit given this past data. This allows designing a more reliable predicative model given the consistency in the statistics. Non-stationary datasets are determined if its statistics behave in an inconsistent and unpredictable manner. In other words, unlike stationary data which does not vary in the way it changes through time, non-stationary does. This naturally makes prediction more difficult as there is no reliable pattern to base a forecasting model on. Hence, if the data can be classified early on as stationary, then the process of building a forecasting model is more dependable because the features within that dataset will work well with forecasting analysis.

There are two methods that will be explored in order to distinguish whether a dataset is stationary or not. The first option is to visualize the data by plotting its features over time and looking for obvious patterns. This method is highly subjective on the readers interpretation of the data which may result in an inaccurate conclusion of stationarity. However, it offers the analyst into what kind of patterns the data may follow.

There are three patterns that can help determine stationarity of the dataset: trend, seasonality, and cyclicity. Trend is the "direction" which the data is moving towards. Is it mainly increasing or decreasing over time? If there is an obvious trend, then it is likely the data is non-stationary as the mean is increasing at a varying rate over time. Seasonality is the effect of the specific points in time that change the behavior of the data. These changes happen in fixed and known frequencies which allows for predictable changes. Cyclicity refers to the data exhibiting cycle of rises and falls that occur in irregular frequencies. Cyclic behavior can be clear in hindsight but becomes more unpredictable in the long-term future. The differences between seasonal and cyclic is that seasonal changes occur at constant length, whereas changes in cyclic vary. Additionally, for reemphasis, seasonality is dependent directly with time, while cyclicity is a result of previous values. Furthermore, seasonality and cyclicity are not clear signs of the data being non-stationary, as long as the change in the variance remains constant over time the data will remain stationary.

When the raw data appears to be non-stationary by simply looking for signs of trend, seasonality, and cyclic behavior then the data can be augmented by either differencing or transformation. Differencing is a method that helps stabilize the mean from changing over time. This is done by subtracting the previous observation from the current observation. Below are the formulas for first and second order differencing, respectively.

$$\Delta y(t) = y(t) - y(t-1)$$

$$\Delta^2 y(t) = y(t) - 2y(t-1) + y(t-2)$$

The first-time differencing is applied is called "first order differencing," and the second time its applied is called "second order differencing," and so on. First order differencing usually detrends the data and second order removes seasonality. One of the consequences of differencing is that one data observation is lost for each order of differencing applied. Therefore, if data is precious, a

logarithmic transformation can be used instead. Transforming the entire dataset by taking either the log or natural log, seasonality can usually be mitigated. It is important that anytime differencing or transformations are applied to data that a reverse transformation is also done.

Histograms also offer a visual median to analyze if data is stationary or non-stationary. A histogram is similar to a bar chart, but instead of plotting two variables it is plotting the frequency distribution of a single variable. For example, how many times does a salesman sell a product at different price points.

A popular option to determine if a dataset is stationary is to perform a statistical hypothesis test which determines the likelihood of the dataset being non-stationary. The Augmented Dickey-Fuller (ADF) test, or "unit root test" makes a strong assumption that dataset in hand is non-stationary. This test sets up two hypotheses:

Null Hypothesis (H0): if failed to be rejected, then the time-series has a unit root and the data is non-stationary.

The Alternative Hypothesis (H1): if the null hypothesis is rejected, then there is no unit root and the data is stationary.

In order to reject the null hypothesis, there must be strong evidence to suggest that the likelihood of the data being non-stationary is significantly unlikely. This is done by setting high confidence levels which are determined by the predetermined critical values. The standard practice is to have between a 95% - 99% confidence level. A 0.05 and 0.01 critical values correspond with a 95% and 99% confidence level, respectively. The ADF test will result with a p-value which is compared to the set critical value. If the p-value is less than the critical value, then there is significantly strong evidence to suggest that the null hypothesis (non-stationary) be rejected in favor of the alternative (stationary). Conversely, if the p-value is greater than the critical value, then there is not enough significantly strong evidence to support rejecting the null hypothesis

Implementation and Results:

The first method for implementation is to calculate the correlation coefficient based upon the formula below in Python without using any built-in functions. The return value for the function is *r*. The code for this customized function can be found at the top of the Appendix section under the function name "correlation_coefficient_calc."

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}.$$

The results of the Python programming method will be compared to the results calculated by hand using the same formula. Each method will be tested on the simple preliminary data shown below.

```
# 2
# Test correlation function on simple datasets
# Dummy variable lists
X = [1,2,3,4,5]
Y = [1,2,3,4,5]
Z = [-1,-2,-3,-4,-5]
G = [1,1,0,-1,-1,0,1]
H = [0,1,1,1,-1,-1,-1]
```

The results calculated by hand are below. The step-by-step solution for calculating the correlation coefficient for the preliminary dataset can be found in the Appendix after the code section.

The correlation coefficient between X and Y is: 1
The correlation coefficient between X and Z is: -1
The correlation coefficient between G and H is: 0

The results from the correlation_coefficient_calc function are below:

```
mean of X: 3.0
mean of Y: 3.0
Cross variance: 10.0
standard devation of X: 3.1622776601683795
standard devation of Y: 3.1622776601683795
The correlation coefficent between X and Y is: 0.9999999999999998
```

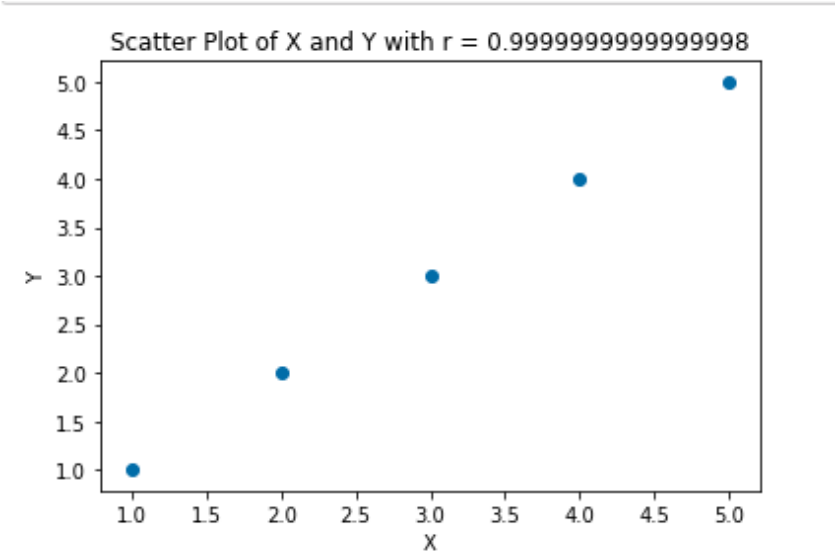
```
mean of X: 3.0
mean of Z: -3.0
Cross variance: -10.0
standard devation of X: 3.1622776601683795
standard devation of Z: 3.1622776601683795
The correlation coefficent between X and Z is: -0.9999999999999998
```

```
mean of G: 0.14285714285714285
mean of H: 0.0
Cross variance: 0.0
standard devation of G: 2.2038926600773587
standard devation of H: 2.449489742783178
The correlation coefficent between G and H is: 0.0
```

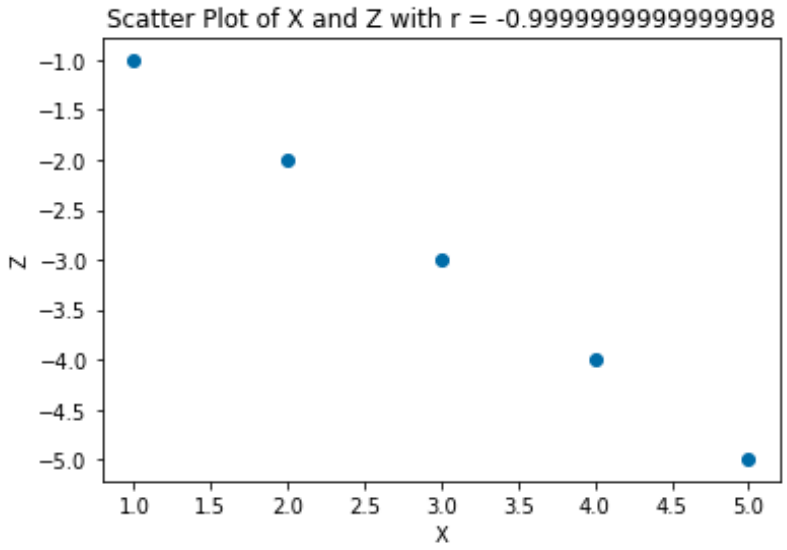
In terms of magnitude there is an extremely small difference between the results of the Python function and the results derived by hand. The results derived by hand for X and Y, and X and Z, were a bit higher than the results from Python. One explanation for this slight difference in values is a consequence of the different types of precisions each method can handle. Python is limited in

its precision; in that it cannot handle irrational numbers. For example, according to Python, the correlation coefficient between X and Y is 0.999…98, yet when derived by hand the result is 1. When derived by hand the standard deviation for X and Y is the square root of 10 so when both are multiplied together the square root is canceled out leaving a cross variance of 10 divide by 10 which is equal to 1. Instead Python has to round these irrational numbers up or down which cause some error. However, there is no such natural phenomena that results in a perfect correlation of either -1 or 1 as they are purely theoretical values. Hence, even though the Python function has some mathematical error it does represent a more realistic approach to the range which *r* can take.

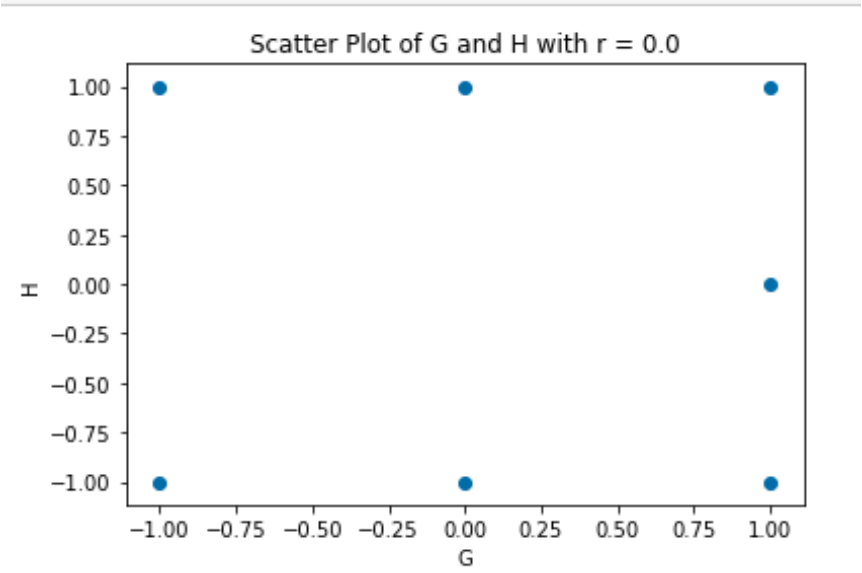
The correlation coefficient can give useful information such as the strength of the linear relationship between two features and the direction of that relationship, is it negative or positive. However, the true relationship between the features may not be linear at all but depending on the data the resulting correlation coefficient may still capture that there is some linear relationship. Hence, it is good practice to plot your data with a scatter plot to get visual evidence of how the data is actually related to support or refute its correlation coefficient. Below are the scatter plots for X vs Y, X vs Z, and G vs H, respectively.



The *r* value of 0.99999 for X and Y makes sense since X and Y have the exact same numbers. This creates a strong linear relationship between both features. As X increases, Y increases by the exact same amount which makes the correlation positive. Having the exact same numbers would theoretically mean r = 1, but that does not exist in the real world.

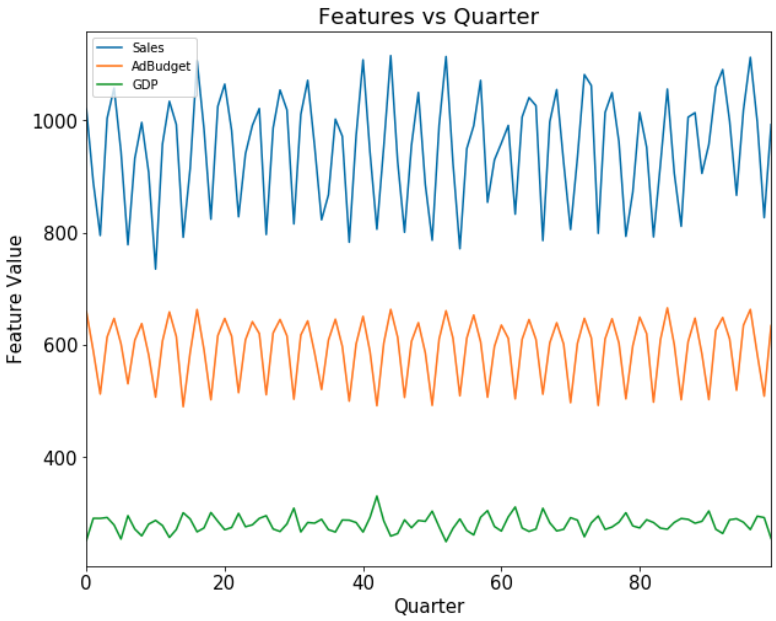


The *r* value of -0.99999 for X and Z makes sense since X and Z have the exact same numbers, which creates a strong linear relationship, but Z is in the negative direction. Having the exact same magnitude but with different directions (negative vs positive) would theoretically mean an r = -1, but that does not exist in the real world.

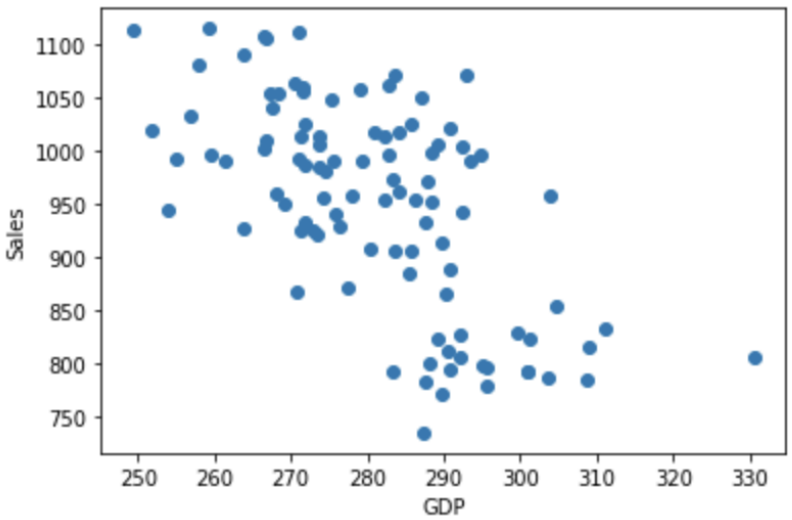


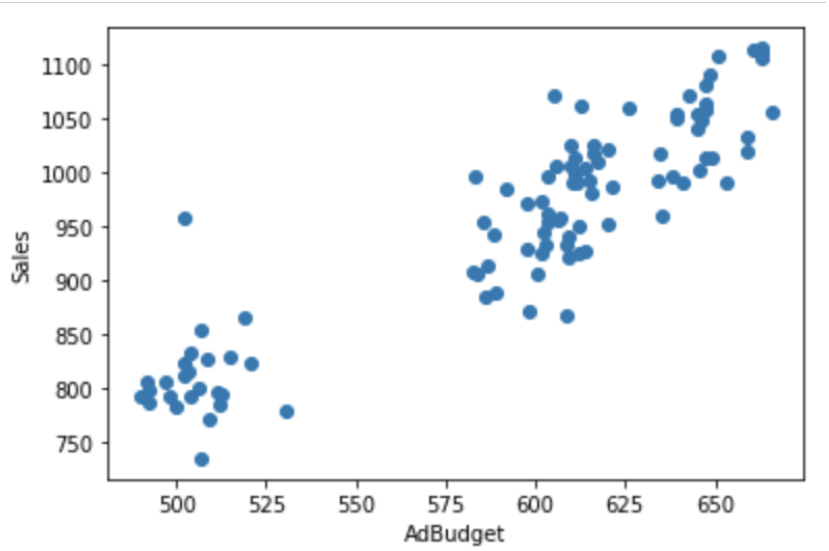
The *r* value of 0.0for G and H makes sense since G and H have no obvious positive or negative relationship. There is no clear linear relationship between the two data sets. Furthermore, the cross variance between G and H is 0, which mathematically supports that r = 0, since 0 is the numerator in the equation.

Moving beyond the simple datasets, the Python function and visualization methods can be applied to a small company's financial data. The data contains the quarterly sales, advertising budget, and GDP (revenue) from 1981 to 2005. Below is a graph plotting the raw data over time.



Instead of visualizing each feature over time, a scatter plot can be used to see if there is a linear relationship between two features. The two following scatter plots first compare GDP with Sales, and the second one compares AdBudget with Sales.



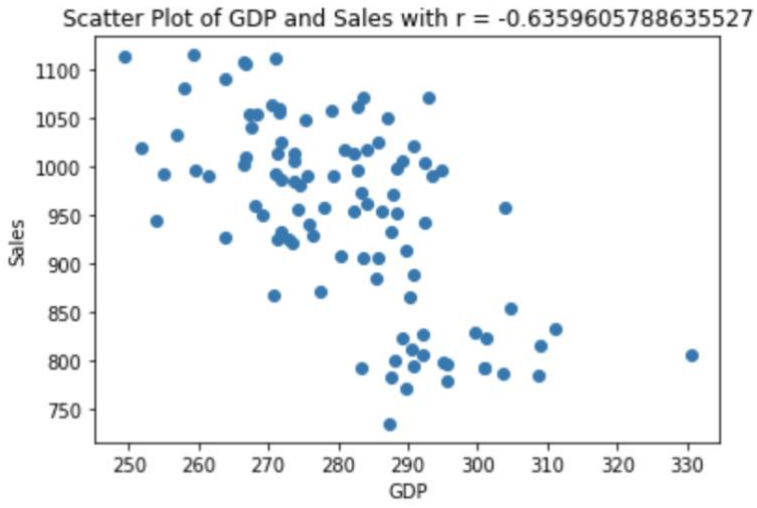


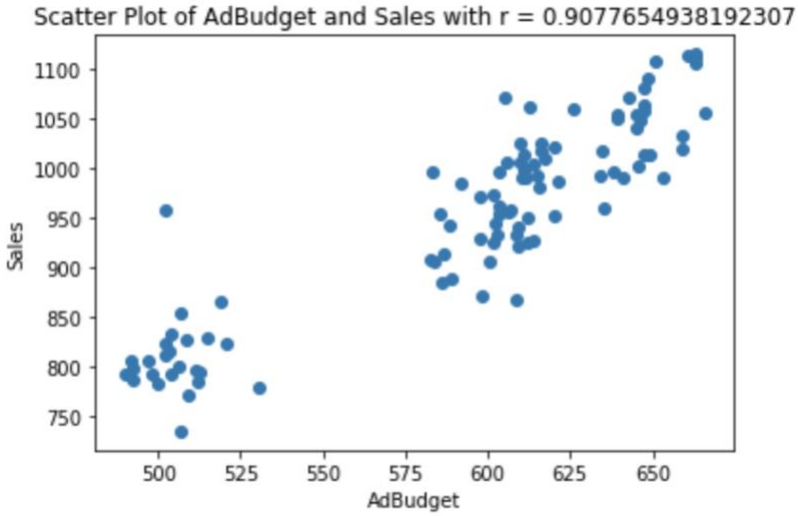
Without any knowledge of knowing the correlation coefficient for each scatter plot, the relationships between each comparison can be described. There is an obvious negative association between GDP and Sales. The strength of a linear relationship is moderate since the data in the scatter plot are not very compact. Conversely, the relationship between AdBudget and Sales has a positive association. The strength of this linear relationship looks much stronger since the data points are more compact to fit a line. To quantify the strength of this relationship, correlation coefficient needs to be calculated. Below are the results of calculating *r* for each comparison using the custom Python function from before.

```
mean of GDP: 281.18300000000005
mean of Sales: 948.737
Cross variance: -88895.36710000003
standard devation of GDP: 142.98454846590943
standard devation of Sales: 977.596917497186
The correlation coefficent between GDP and Sales is: -0.6359605788635527
```

```
mean of AdBudget: 591.933
mean of Sales: 948.737
Cross variance: 479833.3079
standard devation of AdBudget: 540.7006575731159
standard devation of Sales: 977.596917497186
The correlation coefficent between AdBudget and Sales is: 0.9077654938192307
```

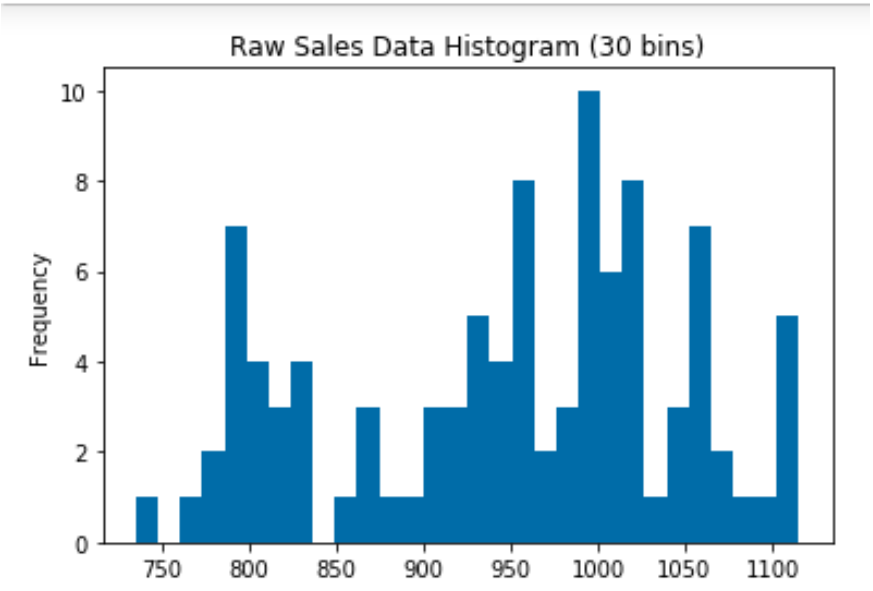
The following scatter plots are the same ones as before, but with their respective r value in the title to compare visual interpretation of correlation with a statistical one.





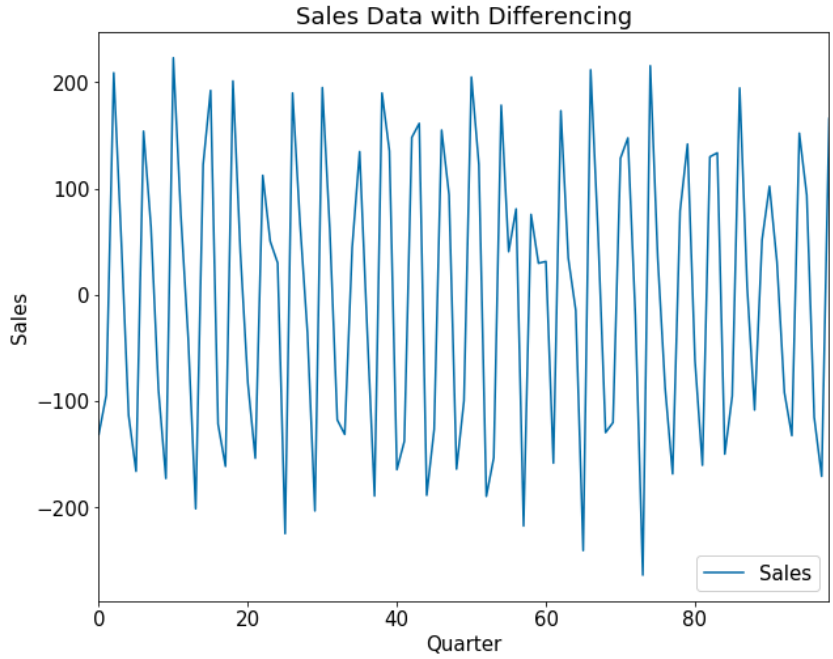
As discussed earlier, a causal effect cannot be concluded from a correlation analysis, visual or statistical. Correlation only determines to what extent a linear relation exists between two random variables. An educated inference can be stated on what may the reason such an association exists between two features. In this case study more information is needed to find the hidden factors that cause negative association between GDP and Sales, and what causes a positive association between AdBudget and Sales.

To check if the sales data is appropriate for time-series analysis it has to be stationary. This means that its core statistics such as the mean and variance do not vary over time. Stationarity can be determined by either visual or statistical analysis. A visual method is to plot the histogram of the data. Below is a histogram of the Sales data spread across 30 bins.



The closer a histogram resembles a normal, or Gaussian distribution then the more likely it is to be stationary. However, like most visual interpretations, this is subjective to the individual analyst. In this case, the data coarsely resembles a normal distribution. Some objections to classifying it as normal is the large cluster around 800, and that the mean seems to be skewed to the left. Yet, there are only a total of 100 samples which may not be enough to accurately determine normality. Thus, the raw Sales data is more likely non-stationarity.

Since it is likely that the Sales data is non-stationary, the differencing method can be applied to remove any trend in the data. By removing the trend, the mean of the data will no longer vary over time. Below is the graph of first order differencing of the Sales data.



Analyzing the differenced Sales data there is an absence of any obvious patterns that would characterize a non-stationary dataset. There is no trend, seasonality, or cyclic behavior. Based upon this evidence the Sales data is stationary in this format.

An additional visual method that is similar to differencing is to transform the raw data logarithmically. This process removes any varying variance in the data over time. The graph below illustrates transforming the raw Sales data by log base 10.

The logarithmic transformation produces a similar result as the differencing method; however, the scales are different. There is no trend, seasonality, or cyclic behavior. This demonstrates that the Sales data was probably stationary to begin with. Based upon this evidence the Sales data is stationary in this format.

The final method to test for stationarity is to use a more objective method such as an ADF test. The results of applying this test on the raw Sales data are below.

```
ADF Statistic: -3.262755
p-value: 0.016628
Critical Values:
        1%: -3.505190
        5%: -2.894232
        10%: -2.584210
```

Applying an ADF test on the Sales data, the results show that the data is stationary. The p-value, 0.016628, is below the critical value threshold of 0.05. This means that there is enough evidence to reject the null hypothesis that the data is non-stationary in favor of the alternative, that it is stationary.

Conclusion:

Overall, there are benefits and drawbacks when doing correlation analysis either visually or by calculating the correlation coefficient. The correlation coefficient quantifies how strong the linear relationship is between two features, but it cannot tell an analyst if a linear relationship is the best one to characterize such relationship. Using scatter plots give visual insight to how the two features actually respond to one another, and if there is doubt to the direction of strength of that linear relationship, the correlation coefficient can determine that. The important take away is to always remember that correlation does not equal causation, even if contextually makes sense. This eager causal determination can lead to wrong conclusions. There are also several ways to determine if a dataset is stationary, whether it is through visualizing histograms, trends, seasonality, or the effects of transformations. Ultimately the ADF test gives the most objective determination of stationarity.

Appendix:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
from statsmodels.tsa.stattools import adfuller



# 1
# Create a function that calculates the correlation coefficent of x and y variables
def correlation_coefficent_cal(dat1, dat2, x, y):
    # calculate cross_variance between x and y
    # convert list data into numpy arrays
    dat1_mean = np.array(dat1).mean()
    print("mean of " + str(x) + ":", dat1_mean)
    dat2_mean = np.array(dat2).mean()
    print("mean of " + str(y) + ":", dat2_mean)
    cross_v = sum((dat1 - dat1_mean) * (dat2 - dat2_mean))
    print("Cross variance:", cross_v)
    dat1_sd = np.sqrt(sum(np.square(dat1 - dat1_mean)))
    print("standard devation of " + str(x) + ":", dat1_sd)
    dat2_sd = np.sqrt(sum(np.square(dat2 - dat2_mean)))
    print("standard devation of " + str(y) + ":", dat2_sd)
    r = cross_v / (dat1_sd * dat2_sd)
```

```
    print("The correlation coefficent between " + str(x) + " and " + str(y) + " is:", r)
    return r

# Create a function that takes the scatter plot of two variables
def scatter_plt(dat1,dat2,x,y,r):
    ax = plt.scatter(dat1,dat2)
    plt.xlabel(str(x))
    plt.ylabel(str(y))
    plt.title("Scatter Plot of " + str(x) + " and " + str(y) + " with r = {}".format(r))
    plt.show()
    return ax

# 2
# Test correlation function on simple datasets
# Dummy variable lists
X = [1,2,3,4,5]
Y = [1,2,3,4,5]
Z = [-1,-2,-3,-4,-5]
G = [1,1,0,-1,-1,0,1]
H = [0,1,1,1,-1,-1,-1]

# A
# test correlation_coefficent function
r_xy = correlation_coefficent_cal(X,Y,"X","Y")

# Plot X and Y
xy_scatter = scatter_plt(X,Y,"X","Y",r_xy)

# B
# test for r

r_xz = correlation_coefficent_cal(X,Z,"X","Z")

# Plot scatter plot between X, Z

xz_scatter = scatter_plt(X,Z,"X","Z",r_xz)
```

```
# C
# test for r

r_gh = correlation_coefficent_cal(G,H,"G","H")
# Scatter plot for G and H
xz_scatter = scatter_plt(G,H,"G","H",r_gh)

# All graphs x-axis and y-axis  with r values in title

# 3
# Load the dataset tute 1

#os.listdir()
df = pd.read_csv("tute1.csv")
#df

# 4
# Dataset relates to the quartlery sales for a small company over the period 1981 - 2005

# 5
# Sales contains quarterly sales, AdBudget is the advertisement budget, and GDP is the gross domestic product for a
small company.

# 6
# Plot Sales, AdBudget, and GDP versus time-steps

ax = df[['Sales','AdBudget',"GDP"]].plot(kind='line',figsize=(10,8), fontsize=15)
plt.legend(loc='upper left', fontsize=10)
ax.set_xlabel('Quarter', fontsize=15)
ax.set_ylabel('Feature Value', fontsize=15)
ax.set_title('Features vs Quarter',fontsize=18)


# 7
# Graph scatter plot for Sales and GDP. Y = Sales, X = GDP. NO TITLE
# Create a function that takes the scatter plot of two variables
def scatter_plt_df(X,Y,x,y):
```

```
    ax = plt.scatter(X,Y)
    plt.xlabel(str(x))
    plt.ylabel(str(y))
    #plt.title("Scatter Plot of " + str(x) + " and " + str(y) + " with r = {}".format(r))
    plt.show()
    return ax

Sgdp = scatter_plt_df(df.GDP,df.Sales,"GDP","Sales")

# 8
# Graph scatter plot for Sales and AdBudget. Y = Sales, X = GDP. NO TITLE
Sadb = scatter_plt_df(df.AdBudget,df.Sales,"AdBudget","Sales")

# 9
# Use the function correlation_coefficient_cal
# Y = Sales, X = GDP
r_xy = correlation_coefficent_cal(df.GDP,df.Sales,"AdBudget","Sales")

# 10
# Use the function correlation_coefficient_cal
# Y = Sales, Z = AdBudget
r_yz = correlation_coefficent_cal(df.AdBudget,df.Sales,"GDP","Sales")

# 11
# incldue r_xy and r_yz in the title of the graphs
# new fucnction to include titels for this data
def scatter_plt_dft(X,Y,x,y,r):
    ax = plt.scatter(X,Y)
    plt.xlabel(str(x))
    plt.ylabel(str(y))
    plt.title("Scatter Plot of " + str(x) + " and " + str(y) + " with r = {}".format(r))
    plt.show()
    return ax

# Scatter plot of GDP and Sales
Sgdp = scatter_plt_dft(df.GDP,df.Sales,"GDP","Sales",r_xy)
```

```
# Scatter plot of AdBudget and Sales

Sadb = scatter_plt_dft(df.AdBudget,df.Sales,"AdBudget","Sales",r_yz)

# 12
# Looking at the correlation coefficents, what effect does AdBudget and GDP have on Sales?

# 13
# Perform an ADF test.
# Plot the histogram of raw Sales data
# Plot the first order differnce sales data
# Plot the logorithmic transformation of sales data
# Which sales dataset is stationary?
# Justify using the ADF and histogram plot

# Create function to calculate ADF score of a dataset
def ADF_Cal(x):
    result = adfuller(x)
    print("ADF Statistic: %f" %result[0])
    print("p-value: %f" %result[1])
    print("Critical Values:")
    for key, value in result[4].items():
        print('\t%s: %3f' % (key,value))


# ADF test for Raw Sales Data
sales_adf = ADF_Cal(df.Sales)

# Plot histogram of raw Sales data
df["Sales"].plot(kind='hist', bins=30,title = "Raw Sales Data Histogram (30 bins)")
plt.show()

# Find the first order difference of Sales
sales_raw = np.array(df.Sales)
print("Length of Sales Raw:",len(sales_raw))
# apply first order differncing
sales_diff = np.diff(df.Sales)
```

```
print("Length of Sales Difference:",len(sales_diff))
print("Input array  : ", sales_raw)
print("First order difference  : ", sales_diff)


# Convert first order differencing array into a pandas dataframe
df_sales_diff = pd.DataFrame(sales_diff)
# Rename columns
df_sales_diff = df_sales_diff.rename(columns={0: "Sales"})

# Plot the first order differnce of Sales
ax = df_sales_diff.plot(kind='line',figsize=(10,8), fontsize=15)
plt.legend(loc='lower right', fontsize=15)
ax.set_xlabel('Quarter', fontsize=15)
ax.set_ylabel('Sales', fontsize=15)
ax.set_title('Sales Data with Differencing',fontsize=18)
plt.show()

# Plot the logorithmic transformation of sales data
df["LogSales"] = np.log(df.Sales)

# Plot the logarithmic transformation
# Logoarithmic transformation removes any varrying variance
ax = df.LogSales.plot(kind='line',figsize=(10,8), fontsize=15)
plt.legend(loc='lower right', fontsize=15)
ax.set_xlabel('Quarter', fontsize=15)
ax.set_ylabel('Sales', fontsize=15)
ax.set_title('Sales Data Logarithmic Transformation',fontsize=18)
plt.show()

# Which sales dataset is stationary?
# Justify using the ADF and histogram plot
```

$X = [1, 2, 3, 4, 5]$
$Y = [1, 2, 3, 4, 5]$

$$r = \frac{\sum (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum (x_t - \bar{x})^2} \sqrt{\sum (y_t - \bar{y})^2}} = 10$$

$$\bar{x} = \frac{1+2+3+4+5}{5} = 3$$

$$\bar{y} = \frac{1+2+3+4+5}{5} = 3$$

$\sum (x_t - \bar{x})(y_t - \bar{y}) = (1-3)*(1-3) = 4 +$ $= 10$

$(2-3)*(2-3) = 2 +$
$(3-3)*(3-3) = 0 +$
$(4-3)*(4-3) = 2 +$
$(5-3)*(5-3) = 4$

$\sqrt{\sum (x_t - \bar{x})^2} = (1-3)^2 = 4 \quad \sum 4, 1, 0, 1, 4 = 10$

$(2-3)^2 = 1$
$(3-3)^2 = 0$
$(4-3)^2 = 1$
$(5-3)^2 = 4$

$\sqrt{10}$

$$\sqrt{\sum (y_t - \bar{y})^2} = (1-3)^2 = 4 \quad \sum 4, 1, 0, 1, 4$$

$(2-3)^2 = 1$
$(3-3)^2 = 0$
$(4-3)^2 = 1$
$(5-3)^2 = 4$

$\sum = 10$

$$r = \frac{10}{\sqrt{10} * \sqrt{10}} = \frac{10}{10} = 1$$

$Z = [-1, -2, -3, -4, -5]$

$X = [1, 2, 3, 4, 5]$

$\bar{Z} = \frac{-1-2-3-4-5}{5} = \boxed{-3}$

$\bar{X} = \frac{1+2+3+4+5}{5} = \boxed{3}$

$$r = \frac{\sum (X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\sum (X_t - \bar{X})^2} \quad \sqrt{\sum (Y_t - \bar{Y})^2}} = \underline{-10}$$

$\sum (X_t - \bar{X})(Y_t - \bar{Y}) = (1-3)^{-2} * (-1+3)^{2} = -4$

$(2-3)^{-1} * (-2+3) = -1$

$(3-3) * (-3+3) = 0$

$(4-3) * (-4+3) = -1$

$(5-3) * (-5+3) = -4$

$\sum -4, -1, 0, -1, -8 = \boxed{-10}$

$\sqrt{\sum (X_t - \bar{X})^2} = (1-3)^2 = 4 \quad \sum 4, 1, 0, 1, 4 = 10$

$(2-3)^2 = 1$

$(3-3)^2 = 0 \quad \sqrt{10}$

$(4-3)^2 = 1$

$(5-3)^2 = 4$

$\sqrt{\sum (Y_t - \bar{Y})^2} = (-1+3)^2 = 4 \quad \sum = 4, 1, 0, 1, 4$

$(-2+3)^2 = 1$

$(-3+3)^2 = 0 \quad \sum = 10$

$(-4+3)^2 = 1$

$(-5+3)^2 = 4 \quad \sqrt{10}$

$r = \frac{-10}{\sqrt{10}\sqrt{10}} = \boxed{-1}$

$G = [1, 1, 1, 0, -1, -1, 0, 1]$

$H = [0, 1, 1, 1, 1, -1, -1, -1]$

$\bar{G} = \frac{1+1+0-1-1+0+1}{7} = \frac{1}{7}$

$\bar{H} = \frac{0+1+1+1-1-1-1}{7} = 0$

$$r = \frac{\sum (G_t - \bar{G})(H_t - \bar{H})}{\sqrt{\sum (G_t - \bar{G})^2}\sqrt{\sum (H_t - \bar{H})^2}} = \frac{\emptyset}{}$$

$\sum (G_t - \bar{G})(H_t - \bar{H}) = (1 - \frac{1}{7})(0-0) = 0$

$(1 - \frac{1}{7})(1-0) = 6/7$

$(0 - \frac{1}{7})(1-0) = -1/7$

$(-1 - \frac{1}{7})(1-0) = -8/7$

$(-1 - \frac{1}{7})(-1-0) = 8/7$

$(0 - \frac{1}{7})(-1-0) \quad 1/7$

$(1 - \frac{1}{7})(-1-0) \quad -6/7$

$\boxed{\sum = 0}$

$\sqrt{\sum (G_t - \bar{G})^2} = \sqrt{301/49}$

$(1 - \frac{1}{7})^2 = (6/7)^2 = 36/49$

$(1 - \frac{1}{7})^2 = (6/7)^2 = 36/49$

$(0 - \frac{1}{7})^2 = (-6/7)^2 = 36/49$

$(-1 - \frac{1}{7})^2 = (-8/7)^2 = 64/49$

$(-1 - \frac{1}{7})^2 = (-8/7)^2 = 64/49$

$(0 - \frac{1}{7})^2 = (-1/7)^2 = 1/49$

$(1 - \frac{1}{7})^2 = (-8/7)^2 = 64/49$

$\sum = 301/49$

$\sqrt{\sum (H_t - \bar{H})^2} = \sqrt{6}$

$(0-0)^2 = 0$

$(1-0)^2 = 1$

$(1-0)^2 = 1$

$(1-0)^2 = 1$

$(-1-0)^2 = 1$

$(-1-0)^2 = 1$

$(-1-0)^2 = 1$

$\sum = 6$

$$r = \frac{0}{\sqrt{301/49}\sqrt{6}} = 0$$

References: