MA 541 Group Project - JMPC & Co. Stocks data Analysis

1st Ashish Bodhankar
MS in Data Science, Spring 2021
Stevens Institute of Technology
Hoboken, NJ, USA
abodhank@stevens.edu

2nd Sambhav Chawla
MS in Data Science, Spring 2021
Stevens Institute of Technology
Hoboken, NJ, USA
schawla1@stevens.edu

3rd Karthik Shankar
MS in Data Science, Spring 2021
Stevens Institute of Technology
Hoboken, NJ, USA
kshanka3@stevens.edu

I. Introduction

The objective of this report is to work on the analysis of a dataset that includes four columns/random variables: the daily ETF return; the daily relative change in the price of the crude oil; the daily relative change in the gold price; and the daily return of the JPMorgan Chase & Co stock. The sample size of this dataset is 1000 observations.

Data analysis includes applying hypothesis testing techniques like the normality tests to understand the distribution of these feature variables, followed by a section of work to discuss the importance of central limit theorem.

Here, we split the data across each attribute into multiple subsets using some sampling techniques and observed the sampling distribution to verify the central limit theorem. We then constructed confidence intervals for predicting the population parameters from the sample statistics. We also carried out few hypothesis tests like the T- test and the Chi-Square test to draw conclusions on the sample population Mean and standard deviations across some of the attributes. We then performed few hypothesis tests to compare the population parameters like the mean and variance of the Gold and Oil data, like the two sample t-test and two sample F-test to compare these means and variances.

Our next task was to fit a line to the two dimensional scatter data of ETF and gold variables, assess the goodeness of fit, and evaluate the regression parameters. And in the last section, our goal was to fit a line using multiple linear regression to predict the the Daily ETF return assuming the Oil and Gold features as explanatory variables. And then assess the quality of this model by checking residuals. We achieve this by plotting a few ”Residual” plots and ”Normal Probability” plots to check the four assumptions made for the error terms of a multiple regression model. We were able to infer that these assumptions were infact followed and not violated. And finally we conclude on how to improve the quality of our regression model as per the strategy of model selection.

Towards the end after the ”References” section we have added the appendix section which includes all python codes and excel charts/outputs. This content might not be aesthetically on par with the preceeding main content. But we tried to add necessary supporting comments wherever necessary.

II. Our Solution

This section elaborates our solution to all the parts mentioned in the project description. Each part has been detailed separately on their respective outcomes and how we arrived at them.

A. part - 1: Meet the Data

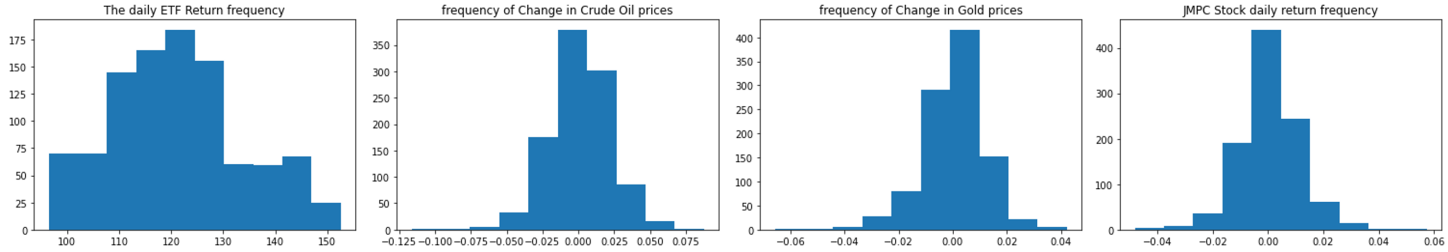
Under this part, our task was to begin the data analysis by calculating some topline information on the measure of the central tendency and the measure of the spread of data under each variable (attribute). After we calculate the mean and the variance, the next task is to indentify the correlation between any two variable. This information helps us with some insights on the dependency of one variable on the other. Below two images are a tabular representation of the desired output.

Variables	Sample Mean	Sample Variance
The Daily ETF Return	121.153	12.564
oil	0.001030	0.0211
gold	0.000663	0.0113
JPM	0.000530	0.0110

Correlation Coefficient Matrix				
	Close_ETF	*oil*	*gold*	*JPM*
Close_ETF	1			
oil	-0.009044842	1		
gold	0.02299557	0.235650372	1	
JPM	0.036807058	-0.12084893	0.100169842	1

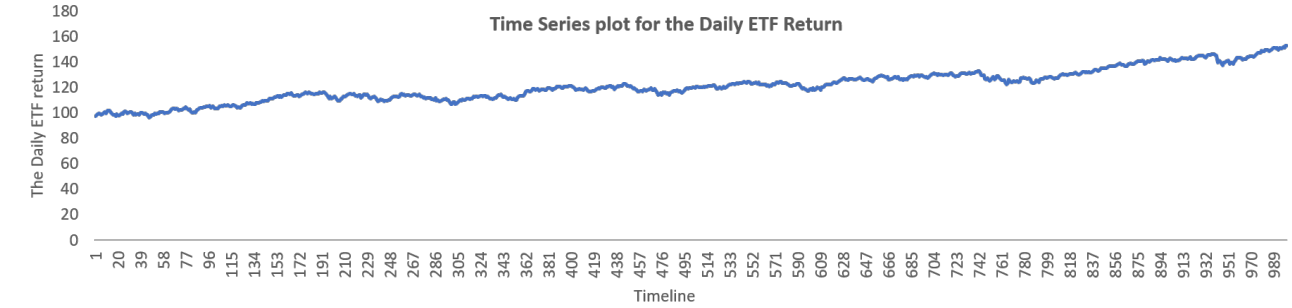
B. part - 2: Describe your Data

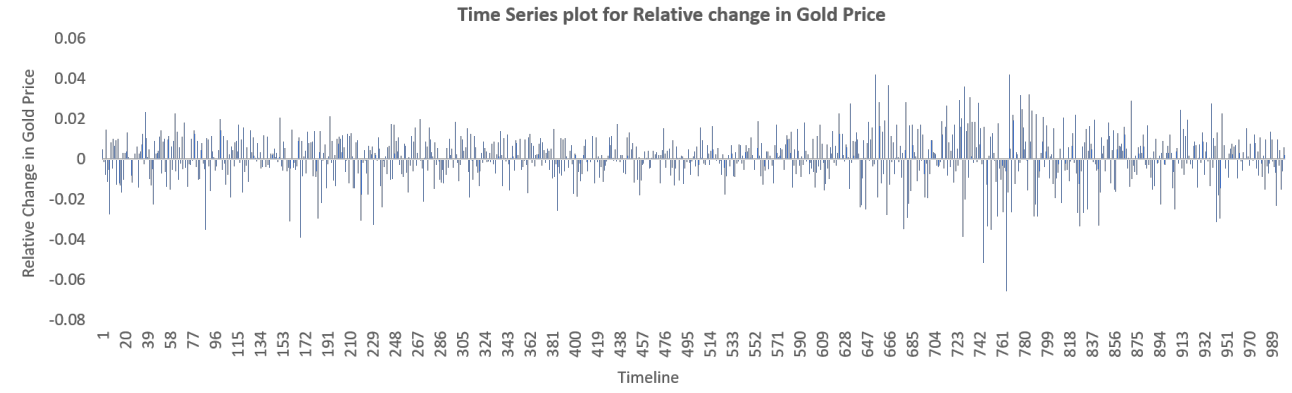
The first task in this section is to plot the histograms from the data under each attribute. In the subsequent section we will conduct some hypthesis tests to verify the resemblence of these distribution with well known standard distrubutions. For example, the normality test.

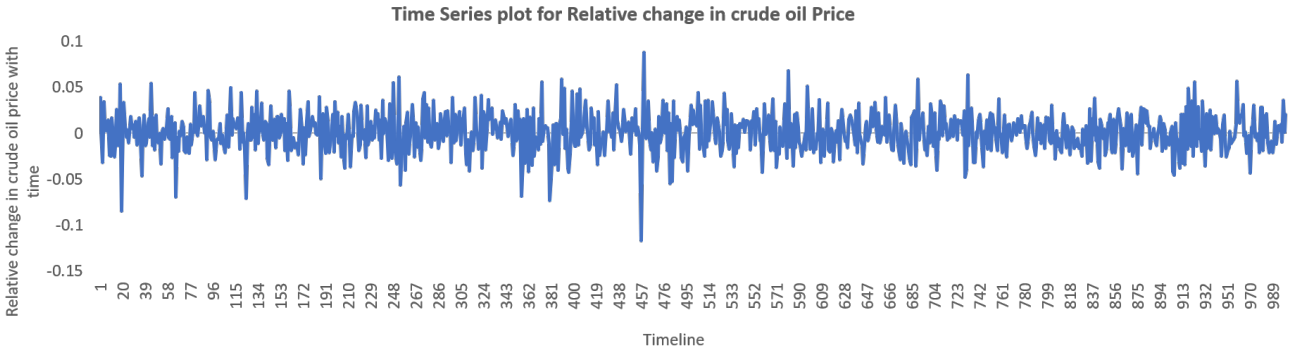


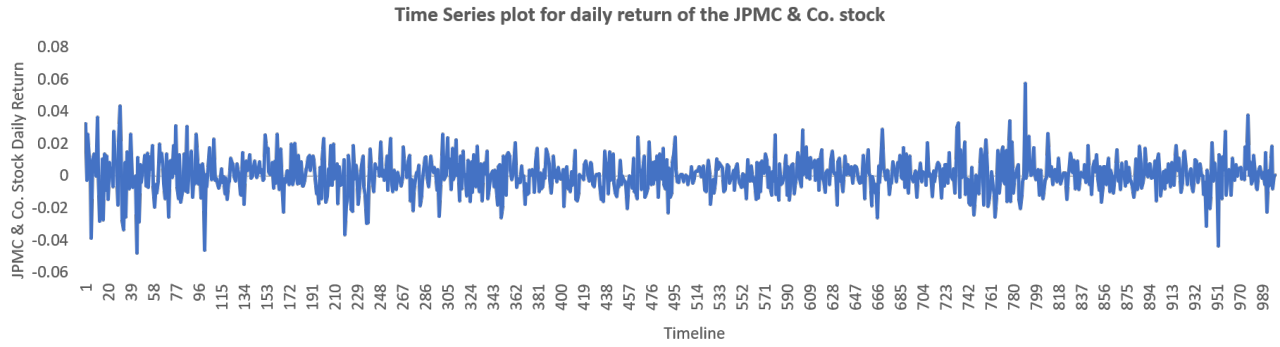
And now, we plot the time series graph for each of the variables separately. Here we used the X-axis as labels for time.(the series "1, 2, 3, . . . , 1000").

We observe that, apart from the time series plot for the daily ETF return column, all the three time series charts are not following any trend. The peaks and lows are very random and almost appears like a noise overall.

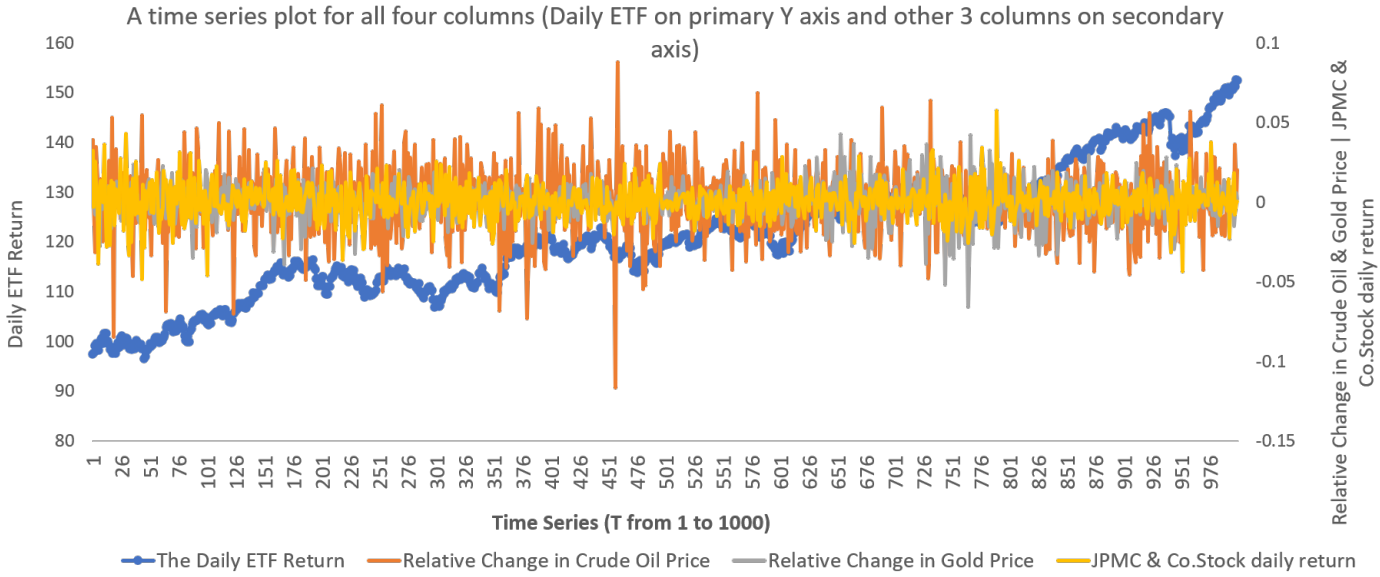




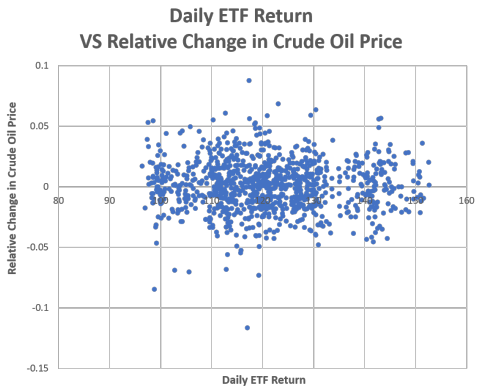


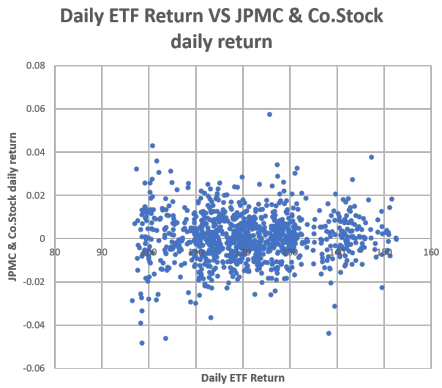


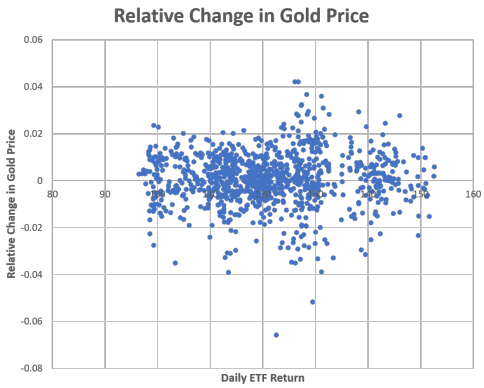
And now we merge all the four charts into one and plot the time series graph from all the 4 columns on the same axes. Notice that the primary axis on the left doesn't start at 0. And hence the daily ETF return graph is more stretched along the Y-axis. This might be a little misleading if you compare to the previous graph for the same variable. But nonetheless, this was done to increase clarity of the chart.



Our final task in this section is to plot three scatter plots to describe the relationships between the ETF column and the OIL column; between the ETF column and the GOLD column; between the ETF column and the JPM column, respectively







C. Part 3: What distribution does your data follow

In this section, we propose an assumption/a hypothesis regarding the type of distribution each column of the data set may follow (i.e., the ETF, OIL, GOLD, and JPM column), based on the plots from Part 2. Then verify or object that assumption/hypothesis with appropriate tests (for example, normality test). SciPy has over 80 distributions that may be used to either generate data or test for fitting of existing data. We fit a number of distributions to our data, compare goodness of fit with a chi-squared value, and test for significant difference between observed and fitted distribution with a Kolmogorov-Smirnov test. The chi-squared value bins data into 50 bins (this could be reduced for smaller data sets) based on percentiles so that each bin contains approximately an equal number of values. For each fitted distribution the expected count of values in each bin is predicted from the distribution. The chi-squared value is the the sum of the relative squared error for each bin.

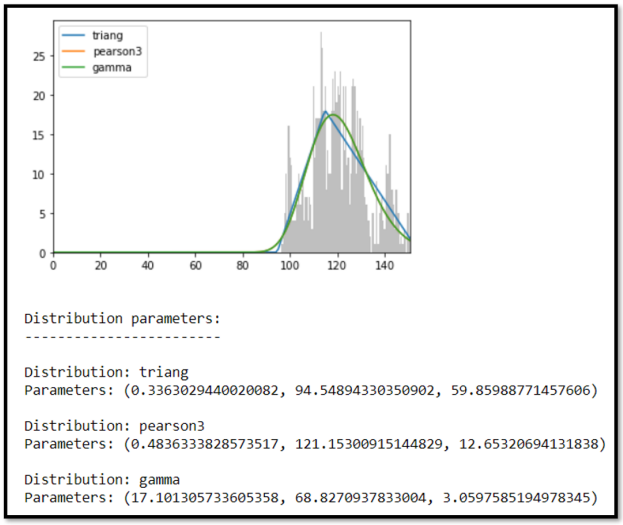
For the observed and predicted we will use the cumulative sum of observed and predicted frequency across the bin range used. The lower the chi-squared value the better the fit. The Kolmogorov-Smirnov test helps us caliberate the P-value. value greater than 0.05 means that the fitted distribution is not significantly different to the observed distribution of the data. It is worth noting that statistical distributions are theoretical models of real-world data. Statistical distributions offer a good way of approximating data (and simplifying huge amounts of data into a few parameters). But when you have a large set of real-world data it is not surprising to find that no theoretical distribution fits the data perfectly.

Refer to the link 1.) under references for more information.

Now we will fit 10 different distributions, rank them by the approximate chi-squared goodness of fit, and report the Kolmogorov-Smirnov (KS) P value results. Remember that we want chi-squared to be as low as possible, and ideally we want the KS P-value to be > 0.05.

Though we are fitting 10 distributions, in the chart we plan to show the top three distributions that fit the best. The first chat below is for the ”The Daily ETF return” column.

Daily ETF Return



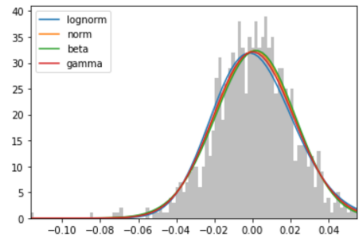

```
Distribution parameters:
------------------------

Distribution: triang
Parameters: (0.3363029440020082, 94.54894330350902, 59.85988771457606)

Distribution: pearson3
Parameters: (0.4836333828573517, 121.15300915144829, 12.65320694131838)

Distribution: gamma
Parameters: (17.101305733605358, 68.8270937833004, 3.0597585194978345)
```

```
Distributions sorted by goodness of fit:
----------------------------------------
  Distribution   chi_square  p_value
8  weibull_min    68.087168  0.09909
6       triang    72.298244  0.03003
5     pearson3    77.399436  0.09353
2        gamma    77.403360  0.09350
3      lognorm    83.661315  0.07501
9  weibull_max    90.698711  0.04371
0         beta   126.580024  0.00342
4         norm   215.783031  0.01238
7      uniform   913.182958  0.00000
1        expon  4885.369836  0.00000
```

We see that ’Triangle’, ’Pearson3’, and, ’gamma’ distributions are the top three fits for this column data. Because they have low chi-square value and P-value greater than 0.05

The next we repeat the same for Oil, Gold and JPMC stock columns. We observe that lognorm & norm distributions fit best to the Oil Price column. beta and Normal distributions fit very well to the gold column. And finally, for the JMPC stock column, Log Normal, Normal and Beta distributions fit the best. Below images make this conclusion very evident.

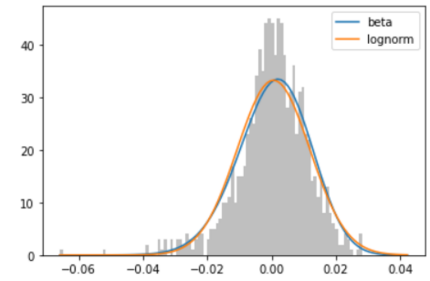
Oil Price




```
Distribution parameters:
------------------------

Distribution: lognorm
Parameters: (0.0810752153755385, -0.2643245680049359, 0.2645027354036523)

Distribution: norm
Parameters: (0.0010300354937470017, 0.02108234946379836)

Distribution: beta
Parameters: (8753.855026685298, 571.503079852072, -7.963684299582814, 8.48470065991468)

Distribution: gamma
Parameters: (544.0024541906312, -0.4946846921282806, 0.000911286048633316)
```

```
Distributions sorted by goodness of fit:
----------------------------------------
  Distribution     chi_square  p_value
3      lognorm      30.477353  0.33996
4         norm      32.749112  0.31363
0         beta      34.292474  0.33238
2        gamma      46.850977  0.18928
9  weibull_max     131.383444  0.00889
8  weibull_min     256.770374  0.00620
6       triang    8493.466471  0.00000
7      uniform   20220.740546  0.00000
1        expon   33089.229010  0.00000
5     pearson3  101749.809394  0.00000
```

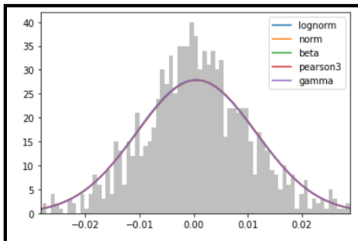
Gold Price




```
Distribution parameters:
------------------------

Distribution: beta
Parameters: (1749234.981397288, 76.4729701213445, -2252.8686925966695, 2252.967844217998)

Distribution: lognorm
Parameters: (0.0029045979064520723, -3.880577091542804, 3.8811802014569414)
```

```
Distributions sorted by goodness of fit:
----------------------------------------
  Distribution     chi_square  p_value
0         beta     152.420094  0.00550
3      lognorm     184.502895  0.00101
4         norm     199.774107  0.00052
2        gamma     244.869495  0.00009
8  weibull_min     306.132827  0.00026
6       triang    9683.769968  0.00000
9  weibull_max   14503.584263  0.00000
7      uniform   22829.234394  0.00000
1        expon   31807.027616  0.00000
5     pearson3  102375.134857  0.00000
```

JPMC stock




```
Distribution parameters:
------------------------

Distribution: lognorm
Parameters: (0.008982592026394995, -1.2240873326336081, 1.2245630909347112)

Distribution: norm
Parameters: (0.0005304110210000001, 0.011011052723643007)

Distribution: beta
Parameters: (1704874.152556062, 1887178.4837689945, -19.834802764251755, 41.79168305830413)

Distribution: pearson3
Parameters: (0.004342904059382132, 0.0005308711551363513, 0.01101069591224272)

Distribution: gamma
Parameters: (34376.02154900927, -2.041180376091465 3, 5.9393440143770516e-05)
```

```
Distributions sorted by goodness of fit:
----------------------------------------
  Distribution     chi_square  p_value
3      lognorm     102.108100  0.02515
4         norm     103.018647  0.02793
0         beta     103.071998  0.02790
5     pearson3     103.135583  0.02722
2        gamma     103.572513  0.02595
8  weibull_min     378.007622  0.00017
6       triang    3761.010318  0.00000
7      uniform    9839.396659  0.00000
9  weibull_max   17728.825103  0.00000
1        expon   27713.252969  0.00000
```

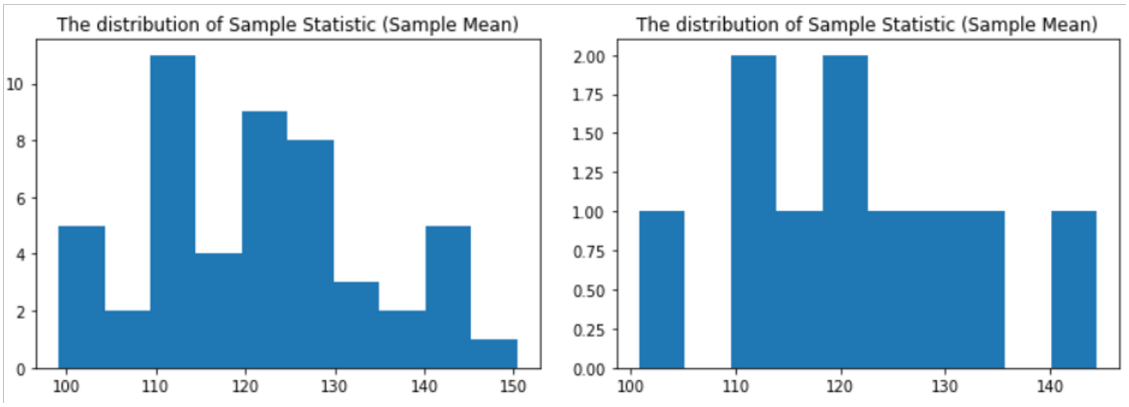
D. Part 4: Break your data into small groups and let them discuss the importance of the Central Limit Theorem

We were asked to consider the ETF column (1000 values) as the population (x).

The first task in this section was to calculate the mean μ_x and the standard deviation σ_x of the population. The Population Mean of the Daily ETF Return = 121.153 and the population Standard Deviation of the Daily ETF Return = 12.57

We were then asked to break the population into 50 groups sequentially such that each group includes 20 values. The sample means for all the 50 groups were calculated to later draw a histogram of all these sample means. The motive was to assess the normality of the data consisting of these sample means. The distribution of the sample statistic is shown in the figure below on the left. Visually we can interpret that this distribution does not resemble a bell shaped curve of a Normal Distribution. By comparing the expected value ($\mu_{\bar{x}}$) of all these sample means (generated by sequential sampling) with the population mean (μ_x), we notice that they are infact the same. This shows that the sample mean is an unbiased estimator of the true population parameter. But when we compare the standard deviation of these sample means about their expected value, we see that this number is, in fact, not equal to the $\frac{1}{\sqrt{n}}$ times the population standard deviation. Because the value for the former metric ($\sigma_{\bar{x}}$) is 12.616 and the value of the later one is 2.811 ($\frac{\sigma_x}{\sqrt{n}}$) for samples of size 20.

Sequential samples of size 20 **Sequential samples of size 100**



Now, we repeat the same proceduce as before, but this time, we break the population into 10 groups sequentially and each group includes 100 values. Even in this case we observe that the expected value (Mean) of sample means ($\mu_{\bar{x}}$) is equal to the population parameter ($\mu_x = 121.153$). But the standard deviation of these sample means ($\sigma_{\bar{x}} = 12.822$) about their expected value, is again, not equal to $\frac{1}{\sqrt{n}}$ times the population standard deviation ($\frac{\sigma_x}{\sqrt{n}} = 1.257$). The above histogram on the right shows the distribution of these 10 sample means.

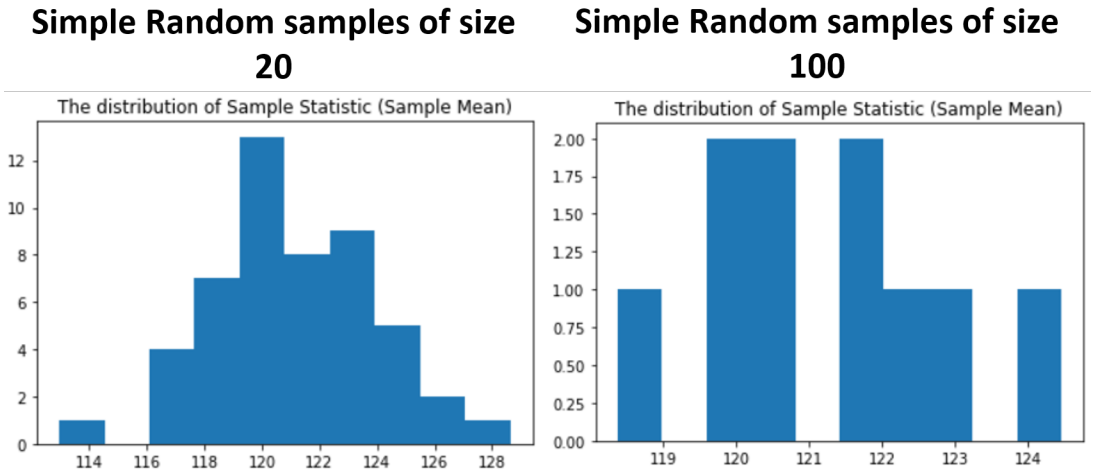
Hence we conclude that the samples generated sequentially are not consistent with the central limit theorem.

The second half of this section follows the last two steps of creating samples of different sizes (size 20 and size 100). But instead of sequential sampling, we perform simple random sampling with repeatations. And assess if central limit theorem applies to samples generated using "Simple Random Sampling". We notice that for both the set of samples generated this way, their expected value is same as population parameter. And, at the same time, the standard deviations of the sample means across both the sets is equal to $\frac{1}{\sqrt{n}}$ the standard deviation of the population.

For the first set (n = 20), the $\sigma_{\bar{x}} = 2.86433$ and $\frac{\sigma_x}{\sqrt{n}} = 2.811$ and for the second set (n=100), the $\sigma_{\bar{x}} = 1.717$ and $\frac{\sigma_x}{\sqrt{n}} = 1.257$. Therefore, we conclude that when sampling is done using SRS, the results are consistent with central limit theorem.

One minor noteworthy observation was that, theoritically, the difference between $\sigma_{\bar{x}}$ and $\frac{\sigma_x}{\sqrt{n}}$ is suppose to be zero asssuming the number of samples available are very high. But in our case the difference isn't that low because we dont have enough samples. When we split the original 1000 observation into samples of size 100, we only have 10 samples. With only 10 such samples, the estimate wasn't that close. But assuming we generated large number of samples of size 100 (using some simulation experiment for example). We would notice that the two values ($\sigma_{\bar{x}}$ and $\frac{\sigma_x}{\sqrt{n}}$) are almost equal.

The distribution of the sample means across both these sets (SRS) are shown in the images below.



The last separate question in this section was to explain the usefulness of having prior information on the distibution with obtaining results of part 4. Our answer to this question is that - Having this information does not change fact that mean of the samples generated with replacement follows a normal distribution when sufficiently large number of samples are available. This is infact the essence of the Central Limit Theorem. Our samples can come from populations following gamma, triangle or normal or any other distribution but the sample means when plotted as histograms are always identical in structure to the bell shape of a normal distribution.

E. Part 5: Construct a confidence interval with your data

In this section we need to pick two simple random samples we created in the previous part. Remember, that we had generated two sets of random samples of two different sizes and conducted same sets of experiments on them. The first was 10 samples of size 100 and then we also created 50 samples of size 20 from the same population (The Daily ETF return). Our task in this part was to randomly pick one sample of size 100 and another one of size 20 and construct an appropriate 95% confidence interval for the mean μ.

To construct confidence intervals for pupulation mean, we can use the z-distribution. But to do this, we need to assume that the population standard deviation is known. The confidence interval using Z-distribution is given by $\bar{x} \pm z \times \frac{\sigma}{\sqrt{n}}$, where $\bar{x}$ is the sample mean and σ is the populations standard deviation. For a 95% confidence level, the Z^*-value is given as 1.96. And so therefore, the confidence interval at this significance level is $(121.197, 126.124)$ for a sample of size 100 that we picked randomly. And next, for the sample of size 20, we arrived at a confidence interval of $(114.8695, 125.8875)$ using the same Z-test.

As we can notice, the first confidence interval is more accurate than the second one because it is clear from the formula we used earlier to determine confidence interval that the width of the interval is inversely proportional to the square root of the sample size. And hence, larger the sample size, narrow is the confidence interval and vice versa.

But what we noticed was that the true population mean ($\sigma = 121.153$) fell only withing the second interval. For the first interval it slightly lies outside the interval with a very low margin. Ideally, theoritically, the population parameter should fall inside the interval but because we only had 10 such samples of size 100. There is a possibility that one of those 10 samples might have a mean that deviates unusually from its expected value. Hence, we observe this discrepancy.

F. Part 6: Form a hypothesis and test it with your data

Our task in this section was to use the same two samples we worked on in the previous part (part 5), random sample of size 100 and size 20 to test a Null hopthesis H_0 that the population mean $\mu = 100$, against an alternate hypothesis $H_1 \neq 100$ at a significance level 0.05.

We can use both the student's t-distribution or the Z-distribution to solve this hypothesis test. In the case of the Z-distribution, assuming we know the population standard deviation (σ of the entire ETF column = 12.5698), testing $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$ at a significance level α is equivalent to computing a $100 \times (1 - \alpha)\%$ confidence interval for μ around μ_0 and rejecting H_0 if $\bar{X}$ is outside of this interval. From confidence intervals we know that:

$$Pr(-z_{\alpha/2} < \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} < z_{\alpha/2}) = 1 - \alpha$$

Therefore, to design a test at the significance level α we choose the critical values a and b as

$$a = \mu_0 - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

$$a = \mu_0 + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

Then we compute the sample mean $\bar{X}$ and reject H_0 if $\bar{X} < a$ or $\bar{X} > b$

for the random sample of size 100, $\alpha = 0.05$, $\sigma = 12.56979$, and $\mu = 100$, $z_{\alpha/2} = z_{0.025} = 1.96$. The confidence interval is $(100 - 1.96 * (\frac{\sigma}{\sqrt{100}}), 100 + 1.96 * (\frac{\sigma}{\sqrt{100}}))$ = $(97.536, 102.464)$. The mean of the sample, $\bar{X} = 123.6603$ does not belong to this interval, hence, we reject the Null Hypothesis H_0. Doing the same for the sample of size 20, we get a confidence interval = $(100 - 1.96 * (\frac{\sigma}{\sqrt{20}}), 100 + 1.96 * (\frac{\sigma}{\sqrt{20}}))$ = $(94.491, 105.509$ using the z-distribution. Even in this case, the random sample mean, $\bar{X} = 120.3785$ does not belong to this interval, hence, we reject the Null Hypothesis H_0.

If we plan to use student's t-distribution assuming unknown population standard deviation, then, for decision making we use the sample variance s^2. We known that in this case the sampling distribution of $\bar{X}$ is the t-distrbution. Critical region at significance level α is $\bar{X} < a$ or $\bar{X} > b$, where

$$a = \mu_0 - t_{\alpha/2}\frac{s}{\sqrt{n}}$$

$$a = \mu_0 + t_{\alpha/2}\frac{s}{\sqrt{n}}$$

where $t_{\alpha/2}$ has $v = n - 1$ degrees of freedom. Equivalently, let $T = \frac{\bar{X}-\mu_0}{s/\sqrt{n}}$ and reject H_0 if $T < -t_{\alpha/2}$ or $T > t_{\alpha/2}$ for $v = n - 1$ degrees of freedom. The t-scores for df = 99 (sample size 100) and df = 19 (sample size 20) at thr probability $P(T \leq t) = 0.975$ are 1.984 and 2.093 respectively. subseqent calculations lead to confidence intervals of $(100 - 1.984 * (\frac{\sigma_{n=100}}{\sqrt{100}}), 100 + 1.984 * (\frac{\sigma_{n=100}}{\sqrt{100}}))$ and $(100 - 2.093 * (\frac{\sigma_{n=20}}{\sqrt{20}}), 100 + 2.093 * (\frac{\sigma_{n=20}}{\sqrt{20}}))$. Solve further, these intervals are as follows - (97.412, 102.588) and (94.983, 105.017).

Notice that the confidence intervals for the Null Hypothesis using both Z-values and t-values are almost the same for both the random samples. Finally, we conclude by rejecting H_0.

the later half of this section uses the sample of size 20 to first test a Null hypothesis H_0 that the population standard deviation $\sigma = 15$ against an alternate hypothesis H_a that $\sigma \neq 15$. The Chi-Square hypothesis test is defined as $H_0 : \sigma^2 = \sigma_0^2$ and $H_a : \sigma^2 \neq \sigma_0^2$. The test statistic is $T = (N-1)(s/\sigma_0)^2$, where N is the sample size and s is the sample standard deviation. The key element of this formula is the ration s/σ_0 which compares the ratio of the sample standard deviation to the target standard deviation. The more this ration deviates from 1, the more likely we are to reject the null hypothesis. At a significance level α, we reject H_0 that the variance is a specified value, σ_0^2, if $T < \chi^2_{\alpha/2,N-1}$ or $T > \chi^2_{1-\alpha/2,N-1}$, where $\chi^2_{.,N-1}$ is the critical value of the chi-square distribution with N-1 degrees of freedom. We use the Chi-Square Table here to determine the values of $\chi^2_{\alpha/2,N-1}$ and $\chi^2_{1-\alpha/2,N-1}$.

for $df = 20 - 1 = 19$, the chi-square test statistic = $19 * (\sigma^2_{n=20}/225) = 9.7038$

$\chi^2_{\alpha/2,N-1}$ for $N = 20$ and $\alpha = 0.05$ = $\chi^2_{0.025,19}$ = 32.852
$\chi^2_{1-\alpha/2,N-1}$ for $N = 20$ and $\alpha = 0.05$ = $\chi^2_{0.975,19}$ = 8.907

Clearly our Chi-Square test statistic lies between these two numbers hence we cannot reject H_0 and therefore, reject the alternate hypothesis H_1

Second, we test $H_0 : \sigma = 15$ vs. $H_a : \sigma < 15$ at the signifiance level 0.05. This is the case of a lower one tail test so we see whether the test statistic's value $< \chi^2_{\alpha,N-1}$. From the Chi Square table we see that for $N = 20$ and $\alpha = 0.05$, $\chi^2_{0.05,19}$ = 30.144. And our test statistic is less than this so we reject the Null Hypothesis in this case.

Link to the chi-square table - Math is Fun Advanced - Chi-Square Table.

G. Part 7: Compare your data with a different data set

the first sub task is to treat the entire Gold column as a random sample from the first population, and the entire Oil column as a random sample from the second population. Assuming these two samples are drawn independently, we form a hypothesis test to see if the Gold and Oil have equal means in the significance level 0.05. We recognize that we have two sample means, one from each set of data, and thus we have two random variables coming from two unknown distributions. To solve the problem we create a new random variable, the difference between the sample means. This new random variable also has a

distribution and, again, the Central Limit Theorem tells us that this new distribution is normally distributed, regardless of the underlying distributions of the original data. For the hypothesis test, we calculate the estimated standard deviation, or standard error, of the difference in sample means, $\overline{X}_1 - \overline{X}_2$. The standard error is: $\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}$. The test statistic (t-score) is calculated as follows:

$$t_c = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{(s_1)^2}{n_1} + \frac{(s_2)^2}{n_2}}}$$

The number of degrees of freedom (df) requires a somewhat complicated calculation. The df are not always a whole number. For the formula to calculate the df numerically, refer to this article - Comparing Two Independent Population Means.

The comparison of two independent population means is very common and provides a way to test the hypothesis that the two groups differ from each other. The test comparing two independent population means with unknown and possibly unequal population standard deviations is called the Aspin-Welch t-test.

Alternatively this test can also be done in microsoft excel directly. The output of "t-test: Two-Sample Assuming Unequal Variances" is as follows:-

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	Variable 2
Mean	0.001030035	0.000662836
Variance	0.00044491	0.000127443
Observations	1000	1000
Hypothesized Mean Difference	0	
df	1528	
t Stat	0.485366614	
P(T<=t) one-tail	0.313742946	
t Critical one-tail	1.645851467	
P(T<=t) two-tail	0.627485893	
t Critical two-tail	1.961517728	

This shows that the test statistic is not greater than the critical value hence we cannot reject the Null Hypothesis. Therefore, the two samples have the same means.

For the second sub task, we subtract the entire Gold column from the entire Oil column and generate a sample of differences. We assume this sample as a random sample from the target population of differences between Gold and Oil and form a hypothesis test to see if the Gold and Oil have equal means in the significance level 0.05. We formulate a Hypothesis test to see if the gold and oil have equal means in the significance level 0.05 as follows. We use the t-test statistic as done similarly in part 6.

That is, $H_0 : \mu = 0$ vs. $H_1 : \mu \neq 0$ at significance level $\alpha = 0.05$.

The confidence interval is $(0 - 1.984 * (\frac{\sigma_{sample}}{\sqrt{1000}}), 0 + 1.984 * (\frac{\sigma_{sample}}{\sqrt{1000}})) = (-0.00134, 0.00134)$ and the sample mean = -0.00037. clearly, the sample mean of differences belongs to the confidence interval and hence we reject the alternate hypothesis H_1.

In the last sub task, we form a hypothesis and test it to see if the Gold and Oil have equal standard deviations in the significance level 0.05. An "F Test" is a catch-all term for any test that uses the F-distribution. In most cases, when people talk about the F-Test, what they are actually talking about is The F-Test to Compare Two Variances. However, the f-statistic is used in a variety of tests including regression analysis, the Chow test and the Scheffe Test (a post-hoc ANOVA test). The "F-test Two sample for Variance" from the data analysis toolkit on excel directly provides the result for this test. Please refer to the table below:

F-Test Two-Sample for Variances

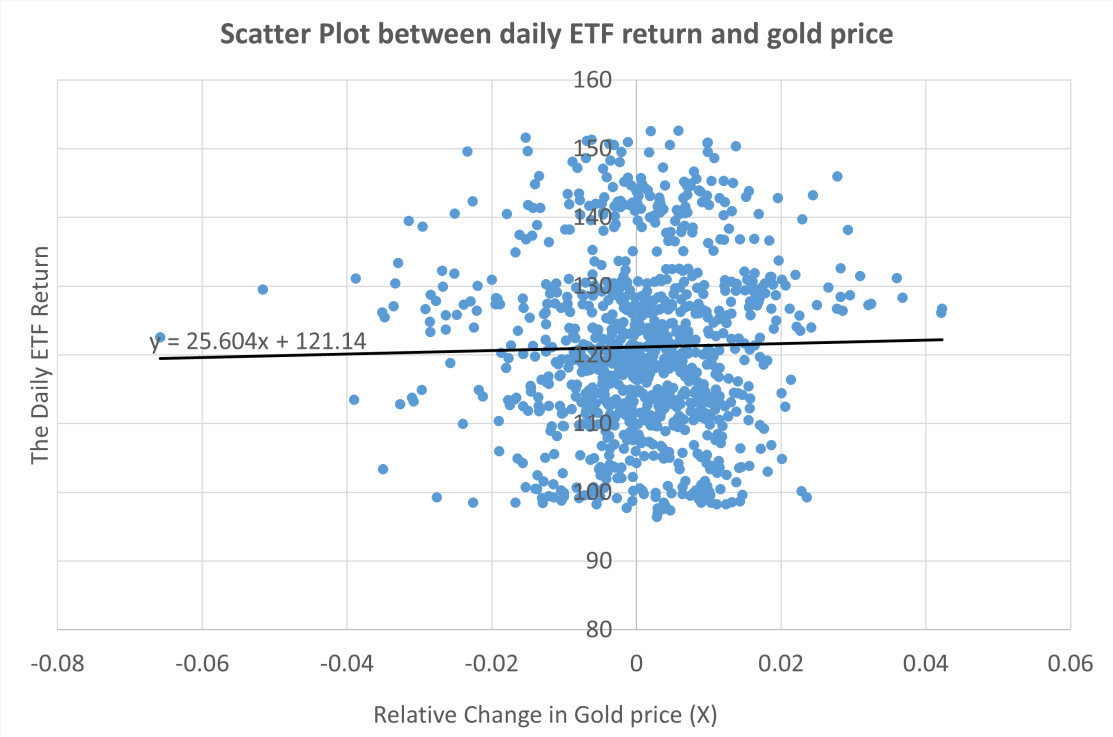
	Variable 1	Variable 2
Mean	0.001030035	0.000662836
Variance	0.00044491	0.000127443
Observations	1000	1000
df	999	999
F	3.491057044	
P(F<=f) one-tail	3.9423E-82	
F Critical one-tail	1.109746138	

The F value is greater than F critical one-tail value. This implies we can reject the Null hypothesis. The Gold and Oil populations' standard deviations are not equal.

For information on this F-test, refer to 2.) under references.

H. Part 8: Fitting the line to the data

Here, firstly, we are asked to draw a scatter plot of ETF (Y) vs. Gold (X) and see if there is any linear relationship between them which can be observed from the scatter plot. And also calculate the coefficient of correlation between ETF and Gold for further interpretation.



It is clear that there is hardly any relation between the two set of random variables. The table for coefficient of correlation below further supports our observation.

	Relative Change in Gold Price	*The Daily ETF Return*
Relative Change in Gold Price	1	
The Daily ETF Return	0.02299557	1

The coefficient of correlation is very close to zero. Almost 0 correlation indicates independence between the two attributes.

Next we have to fit a regression line (or least squares line, best fitting line) to the scatter plot and interpret the regression coefficient. As shown in the picture above, the equation of the best fit line is made visible. The coefficients of linear regression can also be seen in the second table below.

Visually we can see that the regression line is almost parallel to the X-axis. This means, that the variability in the dependent variable is hardly explained by the independent variable as the slope is almost zero. Hence, the y intercept should be the only parameter that explains the total variance of the daily Returns. And since this is a constant term it should be the "Mean of the dependent variable". In other words, the regression line merely outputs the mean of the dependent variable for every value of the independent variable.

SUMMARY OUTPUT for SIMPLE LINEAR REGRESSION

Regression Statistics	
Multiple R	0.02299557
R Square	0.000528796
Adjusted R Square	-0.000472678
Standard Error	12.57276069
Observations	1000

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	121.13599	0.398	304.155	0	120.354	121.918	120.354	121.918
X Variable 1	25.60439	35.236	0.727	0.4676	-43.541	94.750	-43.541	94.750

The standard error regression statistic in the summary output is another goodness-of-fit measure that shows the precision of our regression analysis. It is an absolute measure that shows the average deviation (distance) of the data points from the regression line. Larger the value of this statistic, lesser is the precision of the regression line. For more information checkout 3.) and 4.) under References.

Slope and intercept of the linear regression line are 121.136 and 25.604 respectively. The standard error in measuring the slope is larger than the slope coefficient. This means the coefficient is probably not different from zero.

The fourth point in this sub section wants us to conduct a two-tailed t-test with $H_0 : \beta_1 = 0$ and determine the P-value for this test. Followed by, assessing the linear relationship between ETF (Y) and Gold (X) at the significance level 0.01. The analysis of variance ANOVA table of the Regression analysis (shown below) can help in finding the P-value for this test. We get this in Excel as well. For further reference refer to this link (ANOVA for Regression.)

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression (M)	1	83.46606036	83.466	0.528	0.468
Residual (R)	998	157758.163	158.074		
Total (T)	999	157841.629			

The "F" column provides a statistic for testing the hypothesis that $\beta_1 \neq 0$ against the null hypothesis that $\beta_1 = 0$. The Value F in the above table is calculated as follows:-

$$F = \frac{Mean\ Square\ Model\ (MSM)}{Mean\ square\ Error\ (MSE)} = \frac{83.46606}{158.074311} = 0.528018$$

$$MSM(MS\ Regression) = \frac{SS\ Regression}{df\ Regression} = \frac{83.46606}{1} = 83.46606$$

$$MSE(MS\ Residual) = \frac{SS\ Residual}{df\ Residual} = \frac{157758.162826693}{998} = 158.074311$$

The test statistic is the ratio MSM/MSE, the mean square model term divided by the mean square error term. When the MSM term is large relative to the MSE term, then the ratio is large and there is evidence against the null hypothesis. But in our case the statistic is small and hence we don't have sufficient evidence to support the Alternate hypothesis. For simple linear regression, the statistic MSM/MSE has an F distribution with degrees of freedom (DFM, DFE) = (1, n - 2).

For more information on how to interpret P-values and coefficients in Regression Analysis refer to 5.) We use the F table to ge the p-value for this test. The Significance F (p value) in the ANOVA table of the Regression analysis in excel also outputs this required pvalue and it equals 0.468. We see that it is not less than the significance level of 0.01. So therefore, we can interpret that the data has no sufficient evidence to reject the Null Hypothesis and there is no correlation between the dependent and the independent variable. Thus we conclude that $\beta_1 = 0$

Next, we are supposed to use the coefficient of determination to assess the quality of this fitting. The Coefficient of determination is given by the value of r^2, which is used as an indicator of the goodness of fit. It shows how many points fall on the regression line. For example, 80% means that 80% of the variation of y-values around the mean are explained by the x-values. In other words, 80% of the values fit the model.
It is calculated as follows:-

$$r^2 = \frac{Sum\ of\ Square\ of\ the\ model}{Total\ Sum\ of\ Square} = \frac{83.4660603638622}{157841.628887057} = 0.00053$$

This formalizes the interpretation of r² as explaining the fraction of variability in the data explained by the regression model. In our case the r^2 value is very less. Only 0.053% of the dependent variable values are explained by the independent variable.

FYI - The r^2 value can also be also be calculated as the square of the correlation coefficient.

This shows that the regression line is a very bad fit on our dataset. The linear regression analysis using the data analysis toolkit in excel provides this value directly under summary output.

Following are the assumptions worth mentioning. We assume these for the model fitting:-

- The regression model is linear in the coefficients and the error term
  The error term has a population mean of zero
- All independent variables are uncorrelated with the error term
- Observations of the error term are uncorrelated with each other
- The error term has a constant variance (no heteroscedasticity)
- No independent variable is a perfect linear function of other explanatory variables
- The error term is normally distributed (optional)

lastly, given the daily relative change in the gold price is 0.005127. We need to calculate the 99% confidence interval of the mean daily ETF return, and the 99% prediction interval of the individual daily ETF return. To do this we refer to the following link Confidence and prediction intervals for forecasted values.

The 99% confidence interval for the forecasted values $\hat{y}$ of x is $\hat{y} \pm t_{crit} \times s.e$ where $s.e = s_{y.x}\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{SS_x}}$ and the 99% prediction interval for the forecasted values $\hat{y}$ of x is $\hat{y} \pm t_{crit} \times s.e$ where $s.e = s_{y.x}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{SS_x}}$

For n = 1000, df = 998, the Mean $\mu = 0.000662836$, $x_0 = 0.005127$, predict value $y_0 = 121.2672622$, standard error of the predicted value $s_{y.x} = 12.5727607$, Sum of Square Deviation of the data points from their sample means $SS_x = 0.1273$, and using t-crit (Two tailed inverse of Student's t distribution) = 2.580764586, we calculate the two different respective standard error of predictions separately. One for the confidence interval and the other for the prediction interval calculation.

The standard error of prediction for the confidence interval = 0.427571953 and the same for the prediction interval = 12.58002898. The confidence interval applying the formula above is therefore equal to (120.164,122.371). And the prediction interval is equal to (88.801,153.733).

I. Part 9: Does your model predict?

In this section the requirement was to fit a multiple linear regression model to the data with ETF variable as the response, and Gold and Oil column as the explanatory variables respectively. We are to evaluate the fitness of the curve using the adjusted R^2 value. The image below is a summary of the Ordinary Least Square Regression using the "statsModels" API in python.

R^2-square tends to reward us for including too many independent variables in a regression model, and it doesn't provide any incentive to stop adding more. R-squared increases every time we add an independent variable to the model. The R-squared never decreases, not even when it's just a chance correlation between variables. A regression model that contains more independent variables than another model can look like it provides a better fit merely because it contains more variables. When a model contains an excessive number of independent variables and polynomial terms, it becomes overly customized to fit the peculiarities and random noise in your sample rather than reflecting the entire population. Statisticians call this overfitting the model, and it produces deceptively high R-squared values and a decreased capability for precise predictions. The adjusted R-squared adjusts for the number of terms in the model. Importantly, its value increases only when the new term improves the model fit more than expected by chance alone. The adjusted R-squared value actually decreases when the term doesn't improve the model fit by a sufficient amount.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.027391544
R Square	0.000750297
Adjusted R Square	-0.001254216
Standard Error	12.57767046
Observations	1000

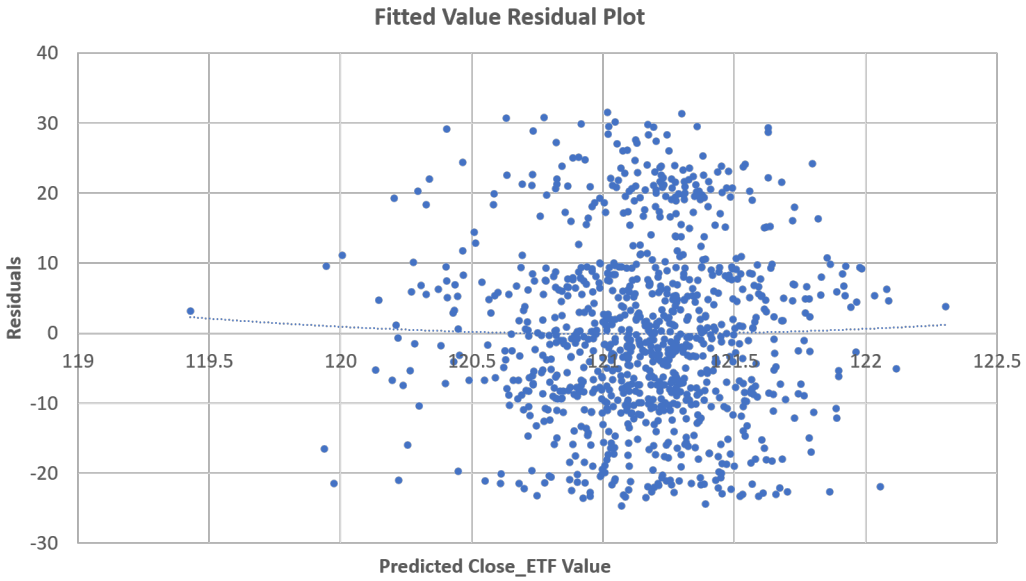
For a good fit, both R^2 and adjusted-R^2 should be atleast 80% or above. In our case, Adjusted R-squared is -0.001 which means only 0.1% of the variation can be explained by the independent variables 'oil' and 'gold'. Hence we can conclude that it is not a good fit. Our model does not predict well.

J. Part 10: Checking residuals and model selection

Our task in this section is to obtain the residuals of the model fitting from the previous part and use them to check the assumptions made for the error term. Residual plots display the residual values on the y-axis and fitted values, or another variable, on the x-axis. After we fit a regression model, it is crucial to check the residual plots. If our plots display unwanted patterns, we can't trust the regression coefficients and other numeric results. There are two fundamental parts to regression models, the deterministic and random components. If our model is not random where it is supposed to be random, it has problems, and this is where residual plots come in.

The deterministic component is the portion of the variation in the dependent variable that the independent variables explain. In other words, the mean of the dependent variable is a function of the independent variables. In a regression model, all of the explanatory power should reside here. The theory here is that the deterministic component of a regression model does such a great job of explaining the dependent variable that it leaves only the intrinsically inexplicable portion of our study area for the error. If we can identify non-randomness in the error term, our independent variables are not explaining everything that they can. To determine whether the residuals are random in regression analysis, we just check that they are randomly scattered around zero for the entire range of fitted values. We can inspect the plot of the residuals vs. the fitted values to assess the assumption of constant variance (homoscedasticity). The below image clearly shows that the residuals center around zero, and they indicate that the model's predictions are correct on average rather than systematically too high or low.

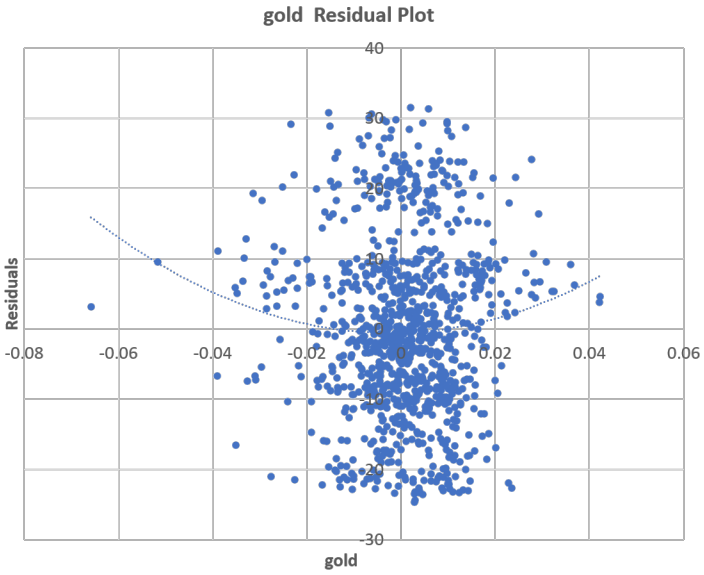
For more information, checout the article 6.) under "References".

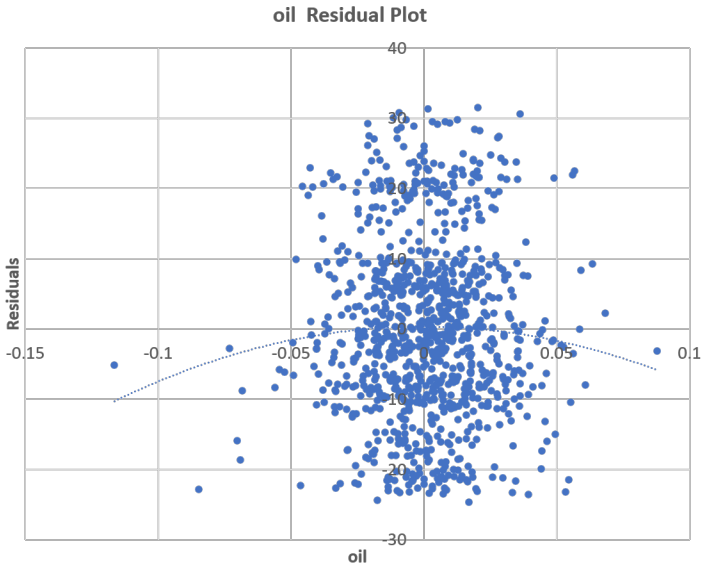


A pattern that is random such as this, also suggests independence. (Assumption that the error is independent is proved).

If we notice, the polynomial best fit curve in the above scatter plot is almost flat and parellel to X axis and also very close to the origin. To further assess what might be causing this slight bent, we separately check the scatter plot distribution of residuals against independent variables.

An alternative to the residuals vs. fits plot is a ”residuals vs. predictor plot.” It is a scatter plot of residuals on the y axis and the predictor (x) values on the x axis. The same polynomial best fit curve is convex for one variable and concave for another. This provides additional insight on how of each independent attribute is influencing the residual distribution. Refer to the chats below.

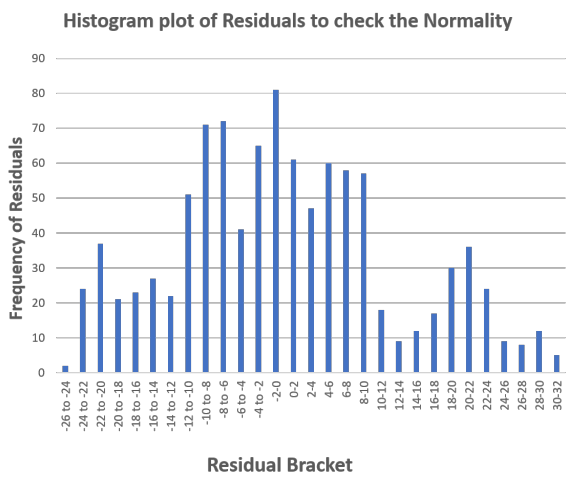


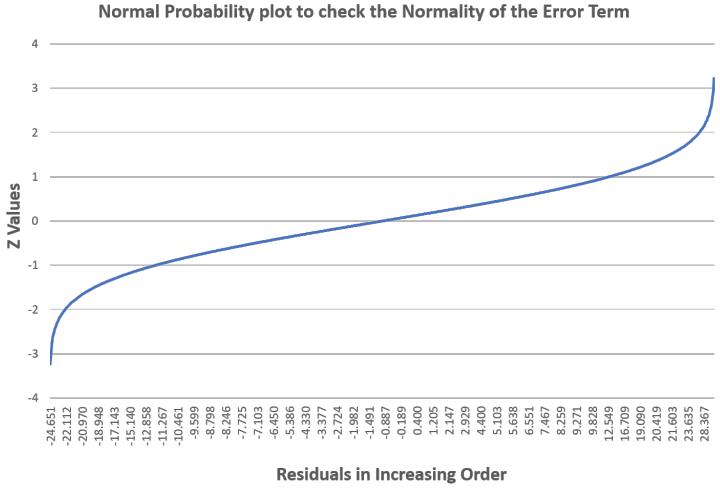


This verifies the assumptions that the error term has a mean 0 and constant variance.

We can also draw a Normal Probability Plot to check the ”third” ”normality” assumption of the error. But before that, we performed a quick check by plotting histograms of the residuals to see if it follows a bell shaped curve of a typical Normal distribution. The below image on the left does not 100% resemble a bell shaped curve. And hence, we don’t immediately conclude, and also constuct a Normal Probability plot as shown in the image below on the right. A normal probability plot is another way we can tell if data fits a normal distribution (a bell curve). With this type of graph, z-scores are plotted against

our residual set (sorted in increasing order). A straight line in a normal probability plot indicates our data does fit a normal probability distribution. A skewed line means that our data is not normal. ("Not normal" in this sense means that it doesn't fit a bell curve).

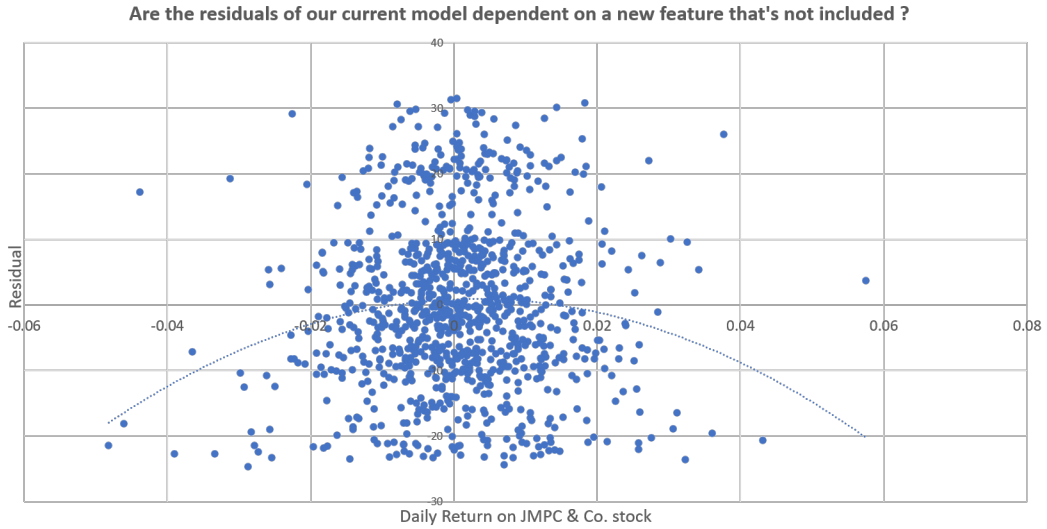




This shows that, if we don't consider some extreme outlier residuals, the normal probability plot almost follows a straight line. Atleast, for majority of the residual values (The plot is almost linear over wide range of residuals). This concludes that the "Normality" assumption of the error term is not violated. We used the article number 7.) under "References" to help us understand Normal Probability Plots.

III. Discussions or Improvements

In our task on multiple regression, we only considered Gold and Oil columns as independent variables. The daily return on JPMC stock column was not considered in this analysis. When we compared the residuals of our model with this new independent variable by plotting a scatter plot, we noticed that the residuals are not entirely scattered randomly around the mean zero. The image below shows that a polynomial best fit curve is far from being a straight line parallel to the X-axis. This shows that we can perhaps include this variable (JPMC Stock column) as a new explanatory feature in further enhancing the model's prediction performance.



And lastly, we never considered normalizing our dataset before building a supervised learning model. We noticed that the oil, gold and JMPC Stock return variables are all centered around 0, while the daily ETF return variable is centered around 121. If we are to scale the daily ETF return data to mean 0, by standardizing the values, we might be able to find more

significant relationship. Overall, this project was a great introduction to empirical data analyses focusing on concepts around statistics. A lot of the topics taught in this course throughout the semester have been used in one way or another to complete this project. Such practical application of data science concepts are crucial to all of us and provides us with the right exposure that's very much necessary. We are thankful to our teaching faculty, Dr. Xiao Zhong for composing such a helpful project for all of us to work on.

IV. References

1.) Python for healthcare modelling and data science.

2.) F Test to Compare Two Variances.

3.) Linear regression analysis in Excel.

4.) Excel Regression Analysis Output Explained.

5.) How to Interpret P-values and Coefficients in Regression Analysis.

6.) Check Your Residual Plots to Ensure Trustworthy Regression Results!

7.) Normal Probability Plot: Definition, Examples

V. Appendix

Appendix includes all the python code written from scratch if needed for reference. All the remaining pages contain pure jupiter .ipynb files exported as PDF pages. All the key takeways has already be mentioned up to this point.

Appendix Part - 1,2 & 3

May 4, 2021

0.1 MA 541 - Statistical Methods.

0.1.1 Part 1: Meet the data

0.1.2 Part 2: Describe your data

0.1.3 Part 3: What distribution does your data follow

[1]:
```
## Import the Necessary Libraries
import pandas as pd
import numpy as np
import scipy
from sklearn.preprocessing import StandardScaler
import scipy.stats
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
# Load data and select first column
dataset = pd.read_excel("the data for your group project_MA541.xlsx")
```

[14]:
```
dataset.describe()
```

[14]:

	Close_ETF	oil	gold	JPM
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	121.152960	0.001030	0.000663	0.000530
std	12.569790	0.021093	0.011289	0.011017
min	96.419998	-0.116533	-0.065805	-0.048217
25%	112.580002	-0.012461	-0.004816	-0.005538
50%	120.150002	0.001243	0.001030	0.000386
75%	128.687497	0.014278	0.007482	0.006966
max	152.619995	0.087726	0.042199	0.057480

[43]:
```
y = dataset.iloc[:,:].values
```

[64]:
```
y
```

[64]:
```
array([[ 9.73499980e+01,  3.92422192e-02,  4.66776500e-03,
         3.22579720e-02],
       [ 9.77500000e+01,  1.95312500e-03, -1.36649400e-03,
        -2.94805400e-03],
```

```
        [ 9.91600040e+01, -3.15139701e-02, -7.93650800e-03,
          2.57243940e-02],
        …,
        [ 1.51300003e+02,  3.61195461e-02, -6.19522500e-03,
         -7.92750100e-03],
        [ 1.52619995e+02,  1.54249576e-03,  5.77771000e-03,
         -3.80517000e-04],
        [ 1.52539993e+02,  2.03295857e-02,  1.96523100e-03,
          3.80662000e-04]])
```

[48]:
```
#Standardize the data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

yy = sc.fit_transform(y)
```

[49]:
```
yy
```

[49]:
```
array([[-1.89461175,  1.81252017,  0.35493945,  2.88142849],
       [-1.86277334,  0.04378494, -0.17985071, -0.31590667],
       [-1.75054318, -1.54366123, -0.76212251,  2.28806306],
       …,
       [ 2.39957287,  1.66440228, -0.60780016, -0.76812928],
       [ 2.50463847,  0.02430755,  0.45330906, -0.08272851],
       [ 2.49827066,  0.91543641,  0.1154256 , -0.01359988]])
```

[51]:
```
y_df = pd.DataFrame(yy, columns=['ETF Return','Crude Oil','Gold Price','JPMC'])
y_df.describe()
```

[51]:

	ETF Return	Crude Oil	Gold Price	JPMC
count	1.000000e+03	1.000000e+03	1.000000e+03	1.000000e+03
mean	5.835332e-16	5.218048e-17	-1.776357e-18	-3.734339e-17
std	1.000500e+00	1.000500e+00	1.000500e+00	1.000500e+00
min	-1.968636e+00	-5.576354e+00	-5.890733e+00	-4.427114e+00
25%	-6.823700e-01	-6.399036e-01	-4.855227e-01	-5.511277e-01
50%	-7.983107e-02	1.009743e-02	3.250381e-02	-1.313830e-02
75%	5.997162e-01	6.283708e-01	6.043441e-01	5.844423e-01
max	2.504638e+00	4.112270e+00	3.681136e+00	5.172015e+00

0.1.4 Fitting a range of distribution and test for goodness of fit

This method will fit a number of distributions to our data, compare goodness of fit with a chi-squared value, and test for significant difference between observed and fitted distribution with a Kolmogorov-Smirnov test.

The chi-squared value bins data into 50 bins (this could be reduced for smaller data sets) based on percentiles so that each bin contains approximately an equal number of values. For each fitted distribution the expected count of values in each bin is predicted from the distribution. The chi-

squared value is the the sum of the relative squared error for each bin, such that:

$$chi - squared = \sum \left(\frac{(observed – predicted)^2}{predicted} \right)$$

For the observed and predicted we will use the cumulative sum of observed and predicted frequency across the bin range used.

The lower the chi-squared value the better the fit.

The Kolmogorov-Smirnov test assumes that data has been standardised: that is the mean is subtracted from all data (so the data becomes centred around zero), and that the results values are divided by the standard deviation (so all data becomes expressed as the number of standard deviations above or below the mean). A value of greater than 0.05 means that the fitted distribution is not significantly different to the observed distribution of the data.

It is worth noting that statistical distributions are theoretical models of real-world data. Statistical distributions offer a good way of approximating data (and simplifying huge amounts of data into a few parameters). But when you have a large set of real-world data it is not surprising to find that no theoretical distribution fits the data perfectly. Having the Kolmogorov-Smirnov tests for all distributions produce results of P<0.05 (fitted distribution is statistically different to the observed data distribution) is not unusual for large data sets. In that case, in modelling we are generally happy to continue with a fit that looks 'reasonable', being aware this is one of the simplifications present in any model.

[86]:
```
yy_etf = yy[:,0]
yy_oil = yy[:,1]
yy_gold = yy[:,2]
yy_jpmc = yy[:,3]
```

[87]:
```
yy_etf = yy_etf.flatten()
yy_oil = yy_oil.flatten()
yy_gold = yy_gold.flatten()
yy_jpmc = yy_jpmc.flatten()
```

[70]:
```
size = len(yy_etf)
size
```

[70]:
```
1000
```

[161]:
```
# Set list of distributions to test
# See https://docs.scipy.org/doc/scipy/reference/stats.html for more

# Turn off code warnings (this is not recommended for routine use)
import warnings
warnings.filterwarnings("ignore")

# Set up list of candidate distributions to use
# See https://docs.scipy.org/doc/scipy/reference/stats.html for more
```

```
dist_names = ['beta',
              'expon',
              'gamma',
              'lognorm',
              'norm',
              'pearson3',
              'triang',
              'uniform',
              'weibull_min',
              'weibull_max']

# Set up empty lists to store results
chi_square = []
p_values = []

# Set up 50 bins for chi-square test
# Observed data will be approximately evenly distrubuted aross all bins
percentile_bins = np.linspace(0,100,51)
percentile_cutoffs = np.percentile(yy_jpmc, percentile_bins)
observed_frequency, bins = (np.histogram(yy_jpmc, bins=percentile_cutoffs))
cum_observed_frequency = np.cumsum(observed_frequency)

# Loop through candidate distributions

for distribution in dist_names:
    # Set up distribution and get fitted distribution parameters
    dist = getattr(scipy.stats, distribution)
    param = dist.fit(yy_jpmc)

    # Obtain the KS test P statistic, round it to 5 decimal places
    p = scipy.stats.kstest(yy_jpmc, distribution, args=param)[1]
    p = np.around(p, 5)
    p_values.append(p)

    # Get expected counts in percentile bins
    # This is based on a 'cumulative distrubution function' (cdf)
    cdf_fitted = dist.cdf(percentile_cutoffs, *param[:-2], loc=param[-2],
                          scale=param[-1])
    expected_frequency = []
    for bin in range(len(percentile_bins)-1):
        expected_cdf_area = cdf_fitted[bin+1] - cdf_fitted[bin]
        expected_frequency.append(expected_cdf_area)

    # calculate chi-squared
    expected_frequency = np.array(expected_frequency) * size
    cum_expected_frequency = np.cumsum(expected_frequency)
```

```
    ss = sum (((cum_expected_frequency - cum_observed_frequency) ** 2) /␣
 ↪cum_observed_frequency)
    chi_square.append(ss)

# Collate results and sort by goodness of fit (best at top)

results = pd.DataFrame()
results['Distribution'] = dist_names
results['chi_square'] = chi_square
results['p_value'] = p_values
results.sort_values(['chi_square'], inplace=True)

# Report results

print ('\nDistributions sorted by goodness of fit:')
print ('----------------------------------------')
print (results)
```

```
Distributions sorted by goodness of fit:
----------------------------------------
  Distribution    chi_square  p_value
3      lognorm    102.108100  0.02515
4         norm    103.018647  0.02793
0         beta    103.071998  0.02790
5     pearson3    103.135583  0.02722
2        gamma    103.572513  0.02595
8  weibull_min    378.007622  0.00017
6       triang   3761.010318  0.00000
7      uniform   9839.396659  0.00000
9  weibull_max  17728.825103  0.00000
1        expon  27713.252969  0.00000
```

0.1.5 We will now take the top three fits, plot the fit and return the sklearn parameters. This time we will fit to the raw data rather than the standardised data.

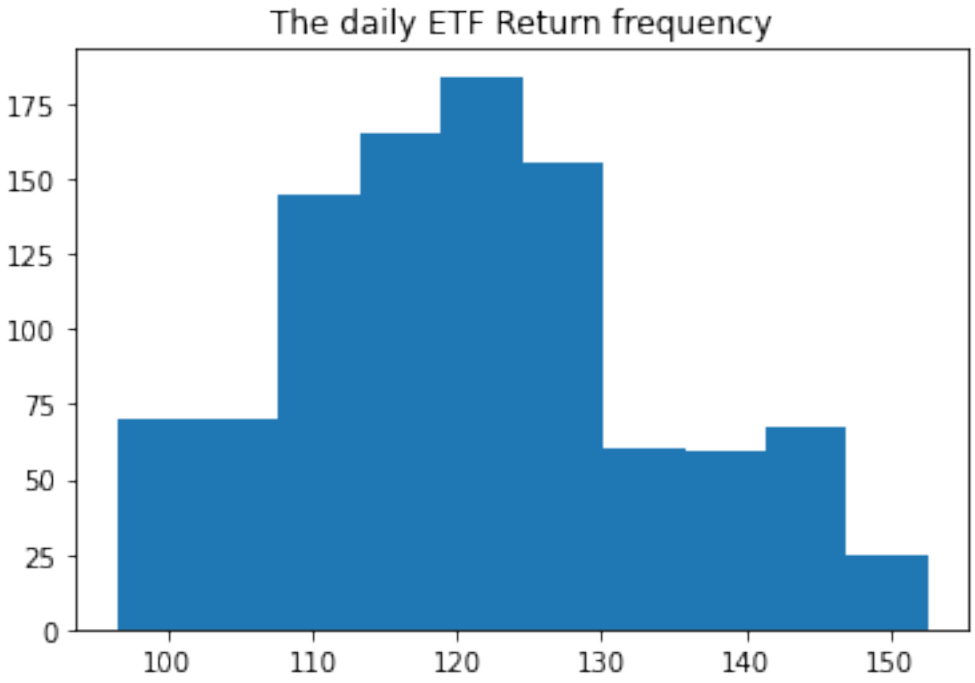
[138]:
```
y_etf = y[:,0]
y_oil = y[:,1]
y_gold = y[:,2]
y_jpmc = y[:,3]

y_etf = y_etf.flatten()
y_oil = y_oil.flatten()
y_gold = y_gold.flatten()
y_jpmc = y_jpmc.flatten()
```

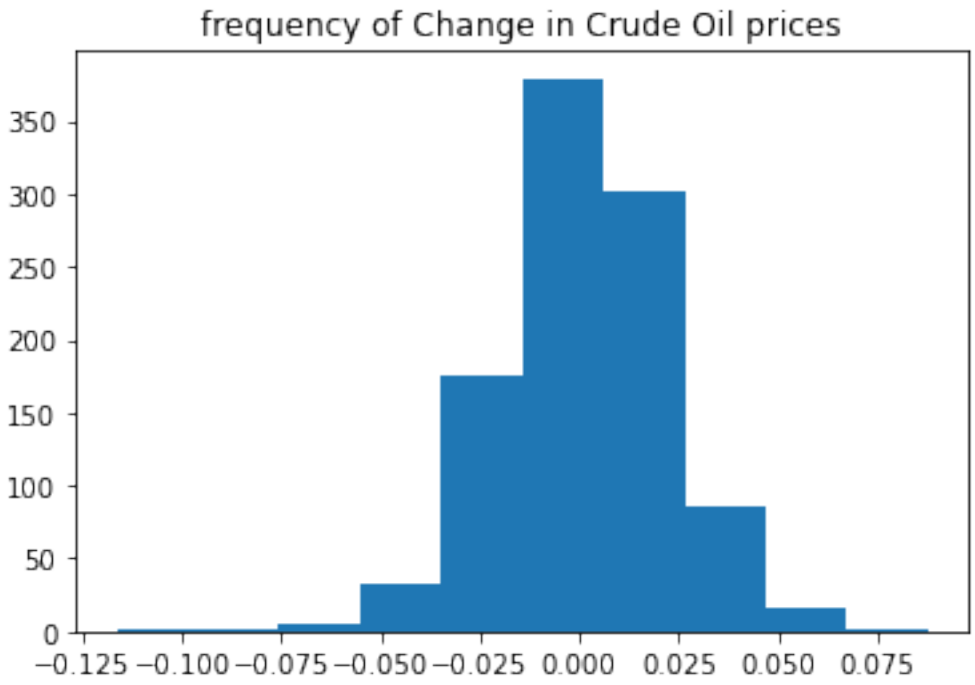
```
#x = np.arange(len(y_etf))
#x = np.linspace(np.min(y_oil), np.max(y_oil),100)
```

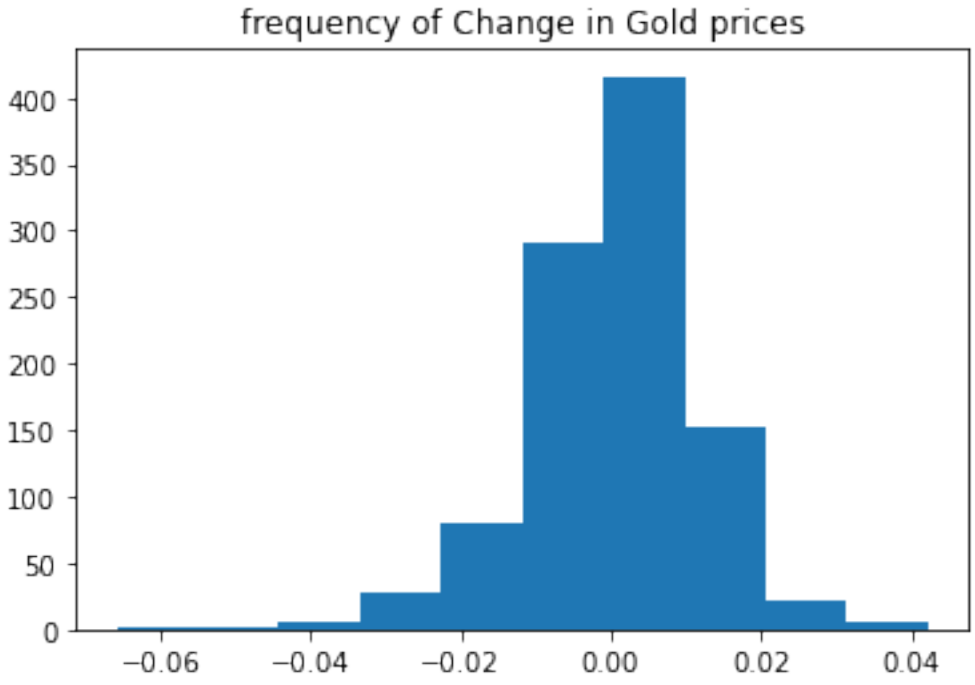
[123]:
```
x.shape
```

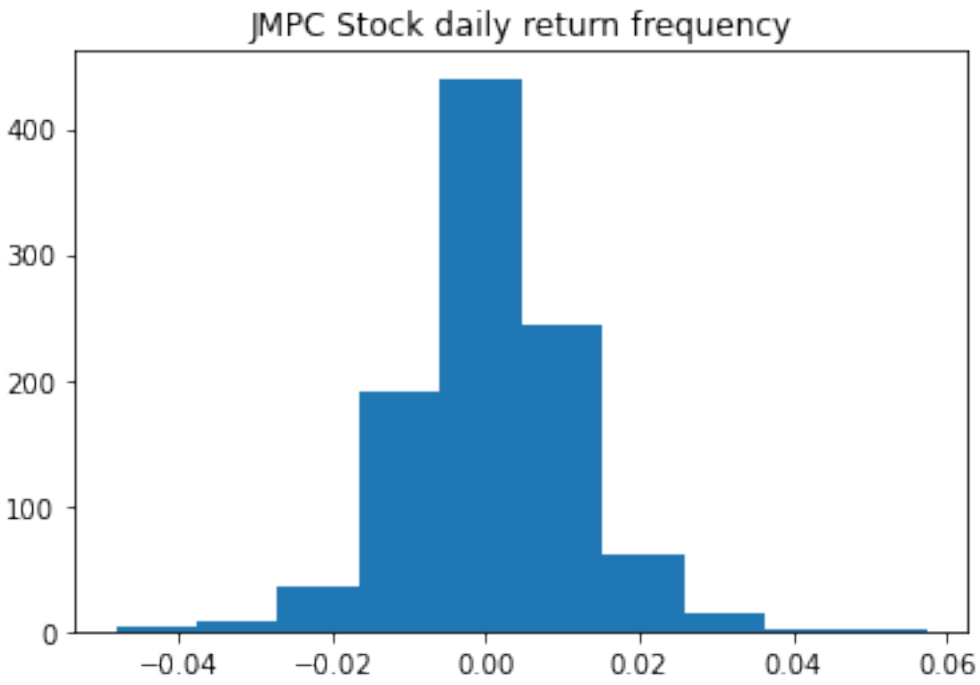
[123]:
```
(100,)
```

[147]:
```
plt.title("The daily ETF Return frequency")
plt.hist(y_etf)
plt.show()
plt.title("frequency of Change in Crude Oil prices")
plt.hist(y_oil)
plt.show()
plt.title("frequency of Change in Gold prices")
plt.hist(y_gold)
plt.show()
plt.title("JMPC Stock daily return frequency")
plt.hist(y_jpmc)
plt.show()
```



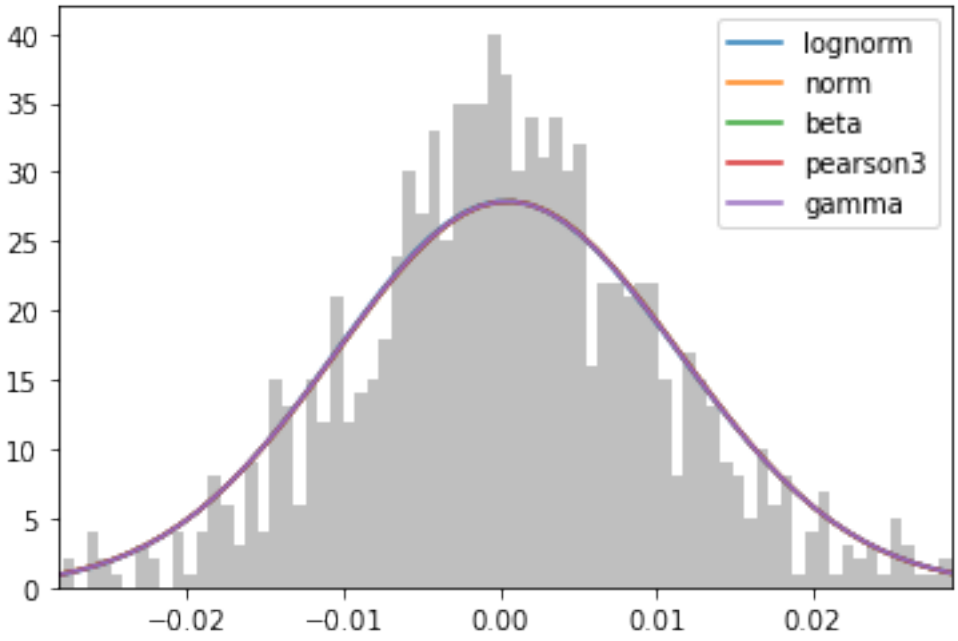

frequency of Change in Crude Oil prices
350
300
250
200
150
100
50
0
−0.125
−0.100
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075


frequency of Change in Gold prices
400
350
300
250
200
150
100
50
0
−0.06
−0.04
−0.02
0.00
0.02
0.04



[164]:
```
# Divide the observed data into 100 bins for plotting (this can be changed)
number_of_bins = 100
bin_cutoffs = np.linspace(np.percentile(y_jpmc,0), np.
 ↪percentile(y_jpmc,99),number_of_bins)

# Create the plot
h = plt.hist(y_jpmc, bins = bin_cutoffs, color='0.75')

# Get the top three distributions from the previous phase
number_distributions_to_plot = 5
dist_names = results['Distribution'].iloc[0:number_distributions_to_plot]

# Create an empty list to stroe fitted distribution parameters
parameters = []

# Loop through the distributions ot get line fit and paraemters

for dist_name in dist_names:
    # Set up distribution and store distribution paraemters
    dist = getattr(scipy.stats, dist_name)
    param = dist.fit(y_jpmc)
    parameters.append(param)

    # Get line for each distribution (and scale to match observed data)
```

```
    x = np.linspace(np.min(y_jpmc), np.max(y_jpmc),100)
    #x = np.linspace(50, np.max(y_etf)+20,1000)
    pdf_fitted = dist.pdf(x, *param[:-2], loc=param[-2], scale=param[-1])
    scale_pdf = np.trapz (h[0], h[1][:-1]) / np.trapz (pdf_fitted, x)
    pdf_fitted *= scale_pdf

    # Add the line to the plot
    plt.plot(x,pdf_fitted, label=dist_name)

    # Set the plot x axis to contain 99% of the data
    # This can be removed, but sometimes outlier data makes the plot less clear
    plt.xlim(np.percentile(y_jpmc,1),np.percentile(y_jpmc,99))

# Add legend and display plot

plt.legend()
plt.show()

# Store distribution paraemters in a dataframe (this could also be saved)
dist_parameters = pd.DataFrame()
dist_parameters['Distribution'] = (
        results['Distribution'].iloc[0:number_distributions_to_plot])
dist_parameters['Distribution parameters'] = parameters

# Print parameter results
print ('\nDistribution parameters:')
print ('------------------------')

for index, row in dist_parameters.iterrows():
    print ('\nDistribution:', row[0])
    print ('Parameters:', row[1] )
```




```
Distribution parameters:
------------------------

Distribution: lognorm
Parameters: (0.008982592026394995, -1.2240873326336081, 1.2245630909347112)

Distribution: norm
Parameters: (0.0005304110210000001, 0.011011052723643007)

Distribution: beta
Parameters: (1704874.152556062, 1887178.4837689945, -19.834802764251755,
41.79168305830413)

Distribution: pearson3
Parameters: (0.004342904059382132, 0.0005308711551363513, 0.01101069591224272)

Distribution: gamma
Parameters: (34376.02154900927, -2.0411803760914653, 5.9393440143770516e-05)
```

Appendix - part 4

May 4, 2021

0.1 Part 4: Break your data into small groups and let them discuss the importance of the Central Limit Theorem

[1]:
```
## Import the Necessary Libraries
import pandas as pd
import numpy as np
import scipy
from sklearn.preprocessing import StandardScaler
import scipy.stats
import matplotlib.pyplot as plt
%matplotlib inline
```

[124]:
```
# Load data and select first column
dataset = pd.read_excel("the data for your group project_MA541.xlsx")
```

[134]:
```
type(dataset)
```

[134]:
```
pandas.core.frame.DataFrame
```

[125]:
```
dataset['Close_ETF'].describe()
```

[125]:
```
count    1000.000000
mean      121.152960
std        12.569790
min        96.419998
25%       112.580002
50%       120.150002
75%       128.687497
max       152.619995
Name: Close_ETF, dtype: float64
```

[126]:
```
describe = dataset['Close_ETF'].describe()
```

0.1.1 1) Calculate the mean μ and the standard deviation σ of the population.

[127]:
```
print("The Population Mean of the Daily ETF Return =␣
↪"+str(round(describe['mean'],3)))
```

```
print("The Population Standard Deviation of the Daily ETF Return =␣
 ↪"+str(round(describe['std'],3)))
```

```
The Population Mean of the Daily ETF Return = 121.153
The Population Standard Deviation of the Daily ETF Return = 12.57
```

0.1.2 2) Break the population into 50 groups sequentially and each group includes 20 values.

[119]:
```
etf = dataset['Close_ETF']
etf = np.array(etf)
```

[120]:
```
etf_subsets = np.split(etf,50)
etf_subsets = np.array(etf_subsets)
```

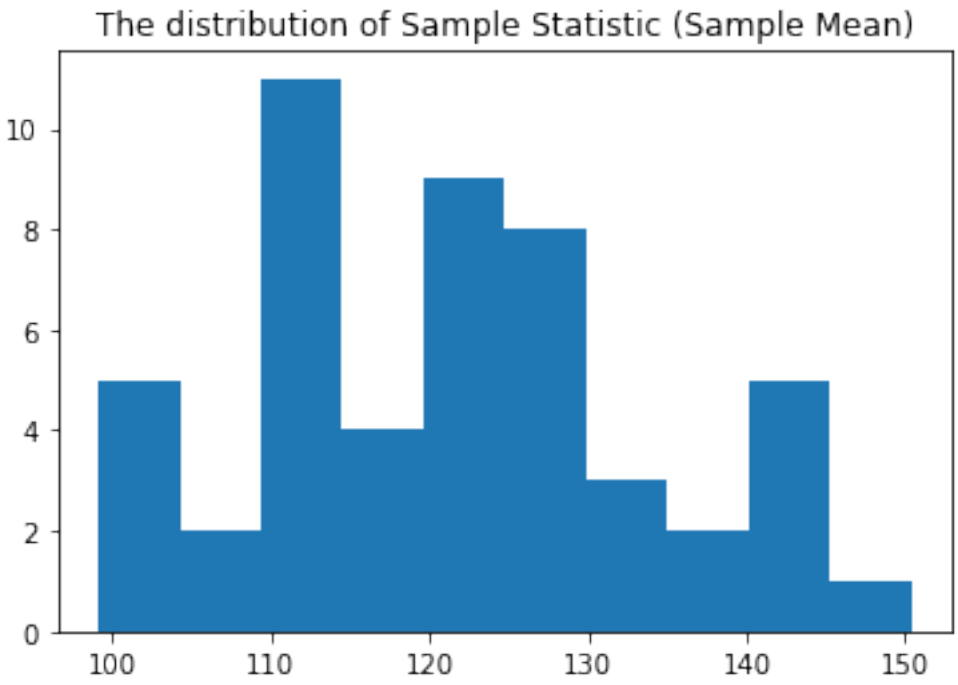
0.1.3 3) Calculate the sample mean ( ) of each group. Draw a histogram of all the sample means. Comment on the distribution of these sample means, i.e., use the histogram to assess the normality of the data consisting of these sample means.

[121]:
```
etf_subsets_mean = np.mean(etf_subsets,axis=1)
```

[122]:
```
## Sample Mean for each group
etf_subsets_mean
```

[122]:
```
array([ 99.3210008 ,  99.55399975,  99.15400055, 102.5505004 ,
       103.29199995, 105.09350015, 106.75099975, 111.6580009 ,
       114.49950015, 114.40050045, 112.7764996 , 112.2859998 ,
       111.8089993 , 113.27149915, 109.9474991 , 110.1430004 ,
       112.53550035, 112.0754997 , 117.78150055, 120.0504997 ,
       118.2080009 , 119.98099935, 119.76750025, 116.80299985,
       117.24199985, 120.55450105, 121.09150045, 123.40999985,
       122.7170002 , 120.61099995, 120.50799975, 125.79700005,
       126.88300015, 127.3025002 , 128.4375004 , 130.13649915,
       130.5825005 , 128.15899955, 125.12550015, 126.06000055,
       129.02949995, 131.8114998 , 135.97399985, 138.857     ,
       141.2884986 , 142.17150035, 144.6245003 , 140.5229988 ,
       144.69050135, 150.35049895])
```

[128]:
```
plt.title("The distribution of Sample Statistic (Sample Mean)")
#plt.hist(list(dataset['Close_ETF']))
plt.hist(list(etf_subsets_mean))
plt.show()
```



The Distribution of the sample Mean doesn't quite follow the normal distribution. Atleast as per visual guess.

0.1.4 4) Calculate the mean ( ) and the standard deviation ( ) of the data including these sample means. Make a comparison between and , between √ and . Here, is the number of sample means calculated from Item 3) above.

[130]:
```
etf_subsets_mean = pd.DataFrame(etf_subsets_mean,columns=['Sample Mean'])
```

[135]:
```
etf_subsets_mean['Sample Mean'].describe()
```

[135]:
```
count     50.000000
mean     121.152960
std       12.615973
min       99.154001
25%      112.348375
50%      120.279250
75%      128.367875
max      150.350499
Name: Sample Mean, dtype: float64
```

[136]:
```
etf_subsets_mean = etf_subsets_mean['Sample Mean'].describe()
```

[139]:
```
etf_subsets_mean['mean']
```

[139]:
```
121.15296001200001
```

By comparing the expected value of sample mean $\mu_{\bar{x}}$ (by averaging all 50 individual sample means) with the total population mean (μ_x) as calculated in point 1.) - shows that,

The sample mean is unbiased estimator of the population parameter as they are both equal

$$\mu_{\bar{x}} = \mu_x$$

[140]:
```
etf_subsets_mean['std']
```

[140]:
```
12.615972812491506
```

[146]:
```
if round(etf_subsets_mean['std'],3) != round(describe['std']/np.sqrt(20),3):
    print('std. deviation of the sample mean is not equal to (1/√)*population␣
 ↪variance')
else:
    print('std. deviation of the sample mean is equal to (1/√)*population␣
 ↪variance')
```

```
std. deviation of the sample mean is not equal to (1/√)*population variance
```

[144]:
```
describe['std']/np.sqrt(20)
```

[144]:
```
2.810690560303399
```

[145]:
```
etf_subsets_mean['std']
```

[145]:
```
12.615972812491506
```

0.1.5 5) Are the results from Items 3) and 4) consistent with the Central Limit Theorem? Why?

They are not consistent with the central limit theorem. Because the standard deviation of the Sample Mean ($\sigma_{\bar{x}}$) is not equal to $\frac{\sigma_x}{\sqrt{n}}$

0.1.6 6) Break the population into 10 groups sequentially and each group includes 100 values.

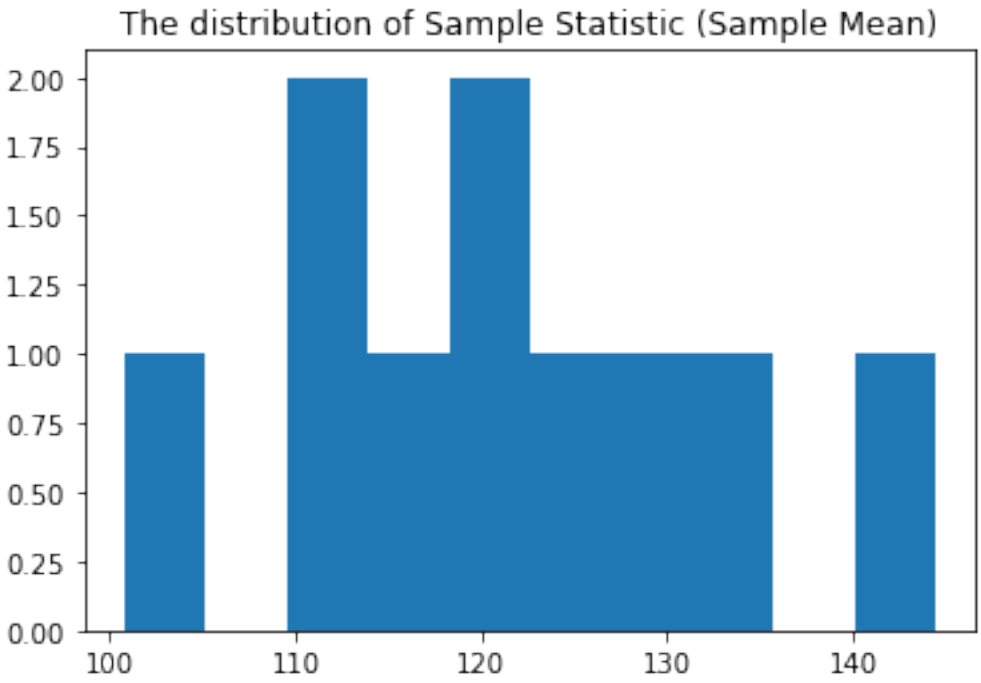
[182]:
```
etf = dataset['Close_ETF']
etf = np.array(etf)
etf_subsets = np.split(etf,10)
etf_subsets = np.array(etf_subsets)
```

0.1.7 7) Repeat Items 3) ~ 5).

[183]:
```
## Sample means of the 10 samples generated sequentially
etf_subsets_mean = np.mean(etf_subsets,axis=1)
list(etf_subsets_mean)
```

[183]:
```
[100.77430028999999,
 110.48050028,
 112.01809938999999,
 114.51720014,
 118.40030004,
 121.67680030000001,
 125.78560011000002,
 128.01269998,
 135.39209964,
 144.47199995]
```

[184]:
```
plt.title("The distribution of Sample Statistic (Sample Mean)")
#plt.hist(list(dataset['Close_ETF']))
plt.hist(list(etf_subsets_mean))
plt.show()
```



The Distribution of the sample Mean doesn't quite follow the normal distribution. Atleast as per visual guess.

[185]:
```
etf_subsets_mean = pd.DataFrame(etf_subsets_mean,columns=['Sample Mean'])
etf_subsets_mean = etf_subsets_mean['Sample Mean'].describe()
```

[186]:
```
# Expected value of the sample Mean
etf_subsets_mean['mean']
```

[186]: 121.152960012

[187]:
```
# Standard Deviation of the Sample Means (sample size 100 generated␣
↪sequentially).
etf_subsets_mean['std']
```

[187]: 12.821725528306828

Let us look at a comparison between $\frac{\sigma_x}{\sqrt{n}}$ and $\sigma_{\bar{x}}$.

[188]:
```
#  /√ Value
describe['std']/np.sqrt(100)
```

[188]: 1.2569790313110745

Similar to the case before, they do not match even in this case.

0.1.8 8) Generate 50 simple random samples or groups (with replacement) from the population. The size of each sample is 20, i.e., each group includes 20 values.

[189]:
```
import random
```

[194]:
```
sample_sets = []

for i in range(50):
    sample = random.choices(list(etf),k=20)
    sample_sets.append(sample)
```

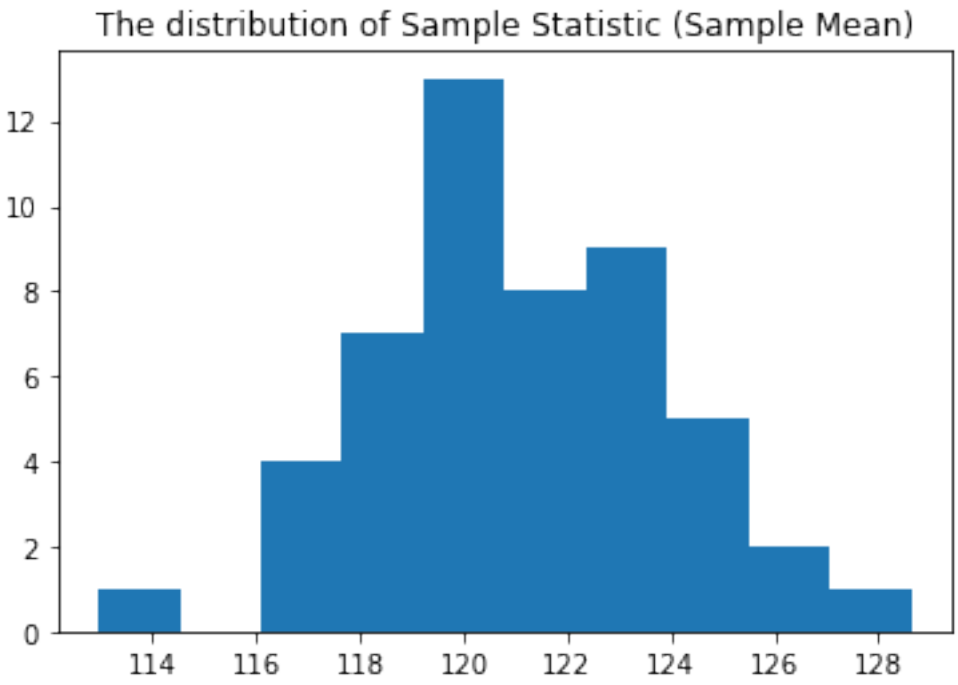
0.1.9 9) Repeat Items 3) ~ 5).

[195]:
```
etf_subsets = np.array(sample_sets)
etf_subsets.shape
```

[195]: (50, 20)

[196]:
```
## Sample means of the 10 samples generated sequentially
etf_subsets_mean = np.mean(etf_subsets,axis=1)
```

[197]:
```
plt.title("The distribution of Sample Statistic (Sample Mean)")
#plt.hist(list(dataset['Close_ETF']))
plt.hist(list(etf_subsets_mean))
plt.show()
```



The distribution of the sample means in this case (sampling with replacement) almost resembles a Normal Distribution Curve. Further steps help us to assess the normality of the distribution numerically by checking its consistency with the Central Limit Theorem

[198]:
```
etf_subsets_mean = pd.DataFrame(etf_subsets_mean,columns=['Sample Mean'])
etf_subsets_mean = etf_subsets_mean['Sample Mean'].describe()
```

[199]:
```
etf_subsets_mean['mean']
```

[199]: 121.11553995600005

[ ]:
```
# Compare the standard deviation of the expected Mean with the population␣
↪parameter:-  /√
```

[200]:
```
etf_subsets_mean['std']
```

[200]: 2.8643300165818575

[202]:
```
describe['std']/np.sqrt(20)
```

[202]: 2.810690560303399

The results are almost consistent with the Central Limit Theorem which states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement, then the distribution

of the sample means will be approximately normally distributed. This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually n > 30). If the population is normal, then the theorem holds true even for samples smaller than 30. This means that we can use the normal probability model to quantify uncertainty when making inferences about a population mean based on the sample mean.

0.1.10 10) Generate 10 simple random samples or groups (with replacement) from the population. The size of each sample is 100, i.e., each group includes 100 values.

[203]:
```
import random
```

[226]:
```
sample_sets = []

for i in range(10):
    sample = random.choices(list(etf),k=100)
    sample_sets.append(sample)
```

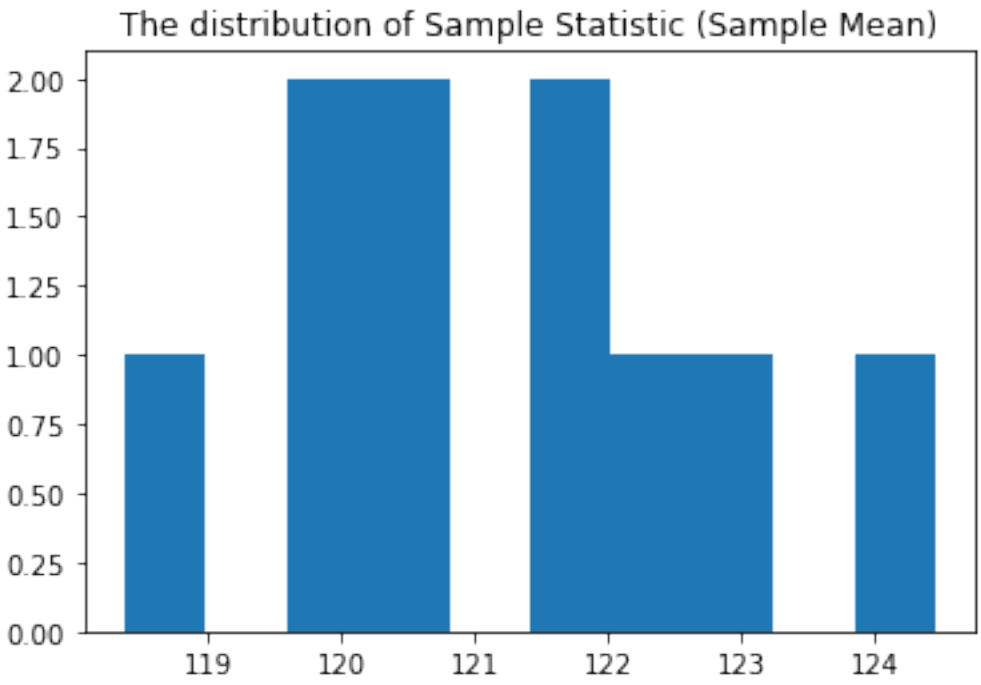
0.1.11 11) Repeat Items 3) ~ 5).

[227]:
```
etf_subsets = np.array(sample_sets)
etf_subsets.shape
```

[227]:
```
(10, 100)
```

[228]:
```
## Sample means of the 10 samples generated sequentially
etf_subsets_mean = np.mean(etf_subsets,axis=1)
```

[229]:
```
plt.title("The distribution of Sample Statistic (Sample Mean)")
#plt.hist(list(dataset['Close_ETF']))
plt.hist(list(etf_subsets_mean))
plt.show()
```



[233]:
```
etf_subsets_mean = pd.DataFrame(etf_subsets_mean,columns=['Sample Mean'])
etf_subsets_mean = etf_subsets_mean['Sample Mean'].describe()
```

[234]:
```
etf_subsets_mean['mean']
```

[234]: 121.298929964

[235]:
```
# Compare the standard deviation of the expected Mean with the population␣
↪parameter:-  /√
```

[236]:
```
etf_subsets_mean['std']
```

[236]: 1.7172269671956664

[237]:
```
describe['std']/np.sqrt(100)
```

[237]: 1.2569790313110745

The results are nearly consistent with Central Limit theorem but not as good as the previous case. Though the sample size increase should converge more in consistency, but since we only generated far less number of samples than before, we observe slightly poor consistency. With the equal number of samples, we should have far better consistency than the previous case.

0.1.12 12) In Part 3 of the project, you have figured out the distribution of the population (the entire ETF column). Does this information have any impact on the distribution of the sample mean(s)? Explain your answer.

Though we have prior information on the distribution of the population data. This doesn't change the fact that mean of the samples generated with replacement follows a normal distribution when sufficiently large number of samples are generated (using some simulation). This is infact the essence of the Central Limit Theorem. My samples can come from populations following gamma, triangle or normal or any other distribution. The sample means when plotted as histograms are almost identical in structure to the bell shape of a normal distribution.

Part 5: Construct a confidence interval with your data.

1) Pick up one of the 10 simple random samples you generated in Step 10) of Part 4, construct an appropriate 95% confidence interval of the mean μ.

In [1]:
```
## Import the Necessary Libraries
import pandas as pd
import numpy as np
import scipy
from sklearn.preprocessing import StandardScaler
import scipy.stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Load data and select first column
dataset = pd.read_excel("the data for your group project_MA541.xlsx")
```

In [3]:
```
describe = dataset['Close_ETF'].describe()
```

In [4]:
```
print("The Population Mean of the Daily ETF Return = "+str(round(describe['mean'],3)))
print("The Population Standard Deviation of the Daily ETF Return = "+str(round(describe
```

```
The Population Mean of the Daily ETF Return = 121.153
The Population Standard Deviation of the Daily ETF Return = 12.57
```

In [5]:
```
etf = dataset['Close_ETF']
etf = np.array(etf)
```

In [6]:
```
import random
sample_sets_100 = []

for i in range(10):
    sample = random.choices(list(etf),k=100)
    sample_sets_100.append(sample)
```

In [7]:
```
random_sample_100 = random.choice(sample_sets_100)
random_sample_100 = np.array(random_sample_100)
random_sample_100_mean = np.mean(random_sample_100)
```

In [8]:
```
## Mean of this random Sample
random_sample_100_mean
```

Out[8]:
```
123.66029981999998
```

Assuming we know the standard deviation of the population we can use the Z-Distribution to calculate the confidence interval. We need to get the z-value for a certain confidence interval

z*–values for Various Confidence Levels

Confidence Level	z*-value
80%	1.28
90%	1.645 (by convention)
95%	1.96
98%	2.33
99%	2.58

As we can see - for a 95% confidence level - The Z-value is given by 1.96

When the population standard deviation is known, the formula for a confidence interval (CI) for a population mean is

$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}}$, where $\bar{x}$ is the sample mean, σ is the population standard

In [9]:
```
z = 1.96
sigma = describe['std']
n = 100
```

In [10]:
```
Confidence_interval_100 = (random_sample_100_mean-(z*(sigma/np.sqrt(n))),random_sample_
```

In [11]:
```
# Therefore the confidence interval is given as follows
Confidence_interval_100
```

Out[11]: (121.19662091863027, 126.12397872136968)

2) Pick up one of the 50 simple random samples you generated in Step 8) of Part 4, construct an appropriate 95% confidence interval of the mean μ.

In [13]:
```
sample_sets_20 = []

for i in range(50):
    sample = random.choices(list(etf),k=20)
    sample_sets_20.append(sample)
```

In [14]:
```
random_sample_20 = random.choice(sample_sets_20)
random_sample_20 = np.array(random_sample_20)
random_sample_20_mean = np.mean(random_sample_20)
```

In [15]:
```
## Mean of this random Sample
random_sample_20_mean
```

Out[15]: 120.3784999

In [16]:
```
z = 1.96
sigma = describe['std']
n = 20
```

In [17]:
```
Confidence_interval_20 = (random_sample_20_mean-(z*(sigma/np.sqrt(n))),random_sample_20
```

In [18]:
```
# Therefore the confidence interval is given as follows
Confidence_interval_20
```

Out[18]: (114.86954640180534, 125.88745339819465)

3) In Part 1, you have calculated the mean μ of the population (the entire ETF column) using Excel function. Do the two intervals from 1) and 2) above include (the true value of) the mean μ? Which one is more accurate? Why?

In [19]:
```
if Confidence_interval_100[0] <= describe['mean'] <= Confidence_interval_100[1]:
    print('The True population mean belongs the first confidence interval')

if Confidence_interval_20[0] <= describe['mean'] <= Confidence_interval_20[1]:
    print('The True population mean also belongs the Second confidence interval')
```

```
The True population mean also belongs the Second confidence interval
```

The first interval is more accurate than the second interval because it is clear from the formula we used earlier to determine confidence interval that the width of the interval is inversely proportional to the square root of the sample size. And hence, larger the sample size, narrow is the confidence interval and vice versa.

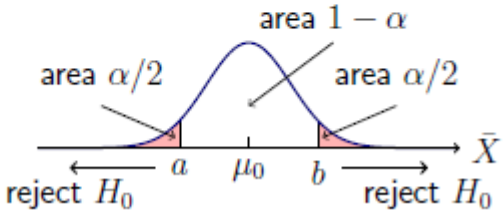
Part 6: Form a hypothesis and test it with your data

1) Use the same sample you picked up in Step 1) of Part 5 to test $H0$: μ=100 vs. Ha: $\mu \neq 100$ at the significance level 0.05. What's your conclusion?

Assuming we have a known population standard Deviation. The entire ETF column is assumed to be the population with a standard deviation computed above.

From confidence intervals we know that

$$\Pr\left(-z_{\alpha/2} < \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} < z_{\alpha/2}\right) = 1 - \alpha$$



Testing $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$ at a significance level α is equivalent to computing a $100 \times (1-\alpha)\%$ *confidence interval* for μ and H_0 if μ_0 is outside this interval.

Therefore, to design a test at the level of significance α we choose the critical values a and b as

$$a = \mu_0 - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$
$$b = \mu_0 + z_{\alpha/2}\frac{\sigma}{\sqrt{n}},$$

then we collect the sample, compute the sample mean $\bar{X}$ and reject H_0 if $\bar{X} < a$ **or** $\bar{X} > b$.

For a significance Level $\alpha = 0.05$ it is equivalent to computing a 95% confidence interval for $\mu = 100$

We reject H_0 if the $\bar{X}$ falls outside this interval.

Given the value of $Z_{\alpha/2} = Z_{0.025} = 1.96$

In [20]:
```
### given standard deviation of the population
describe['std']
```

Out[20]:
```
12.569790313110744
```

In [21]:
```
## The confidence interval are as follows:-
a = 100 - (1.96*(describe['std']/np.sqrt(100)))
b = 100 + (1.96*(describe['std']/np.sqrt(100)))
conf_interval = (a,b)
conf_interval
```

Out[21]:
```
(97.5363210986303, 102.4636789013697)
```

In [22]:
```
## Let us compare this to the Random sample mean that we have.
random_sample_100_mean
```

Out[22]: 123.66029981999998

The random sample mean does not belong to the interval (a, b). Hence we reject the Null Hypothesis H_0

2) Use the same sample you picked up in Step 2) of Part 5 to test $H0$: μ=100 vs. Ha: $\mu \neq 100$ at the significance level 0.05. What's your conclusion?

In [23]:
```
### The confidence interval this time would change as the sample size is different.

a = 100 - (1.96*(describe['std']/np.sqrt(20)))
b = 100 + (1.96*(describe['std']/np.sqrt(20)))
conf_interval = (a,b)
conf_interval
```

Out[23]: (94.49104650180534, 105.50895349819466)

In [24]:
```
## Let us compare this to the Random sample mean that we have.
random_sample_20_mean
```

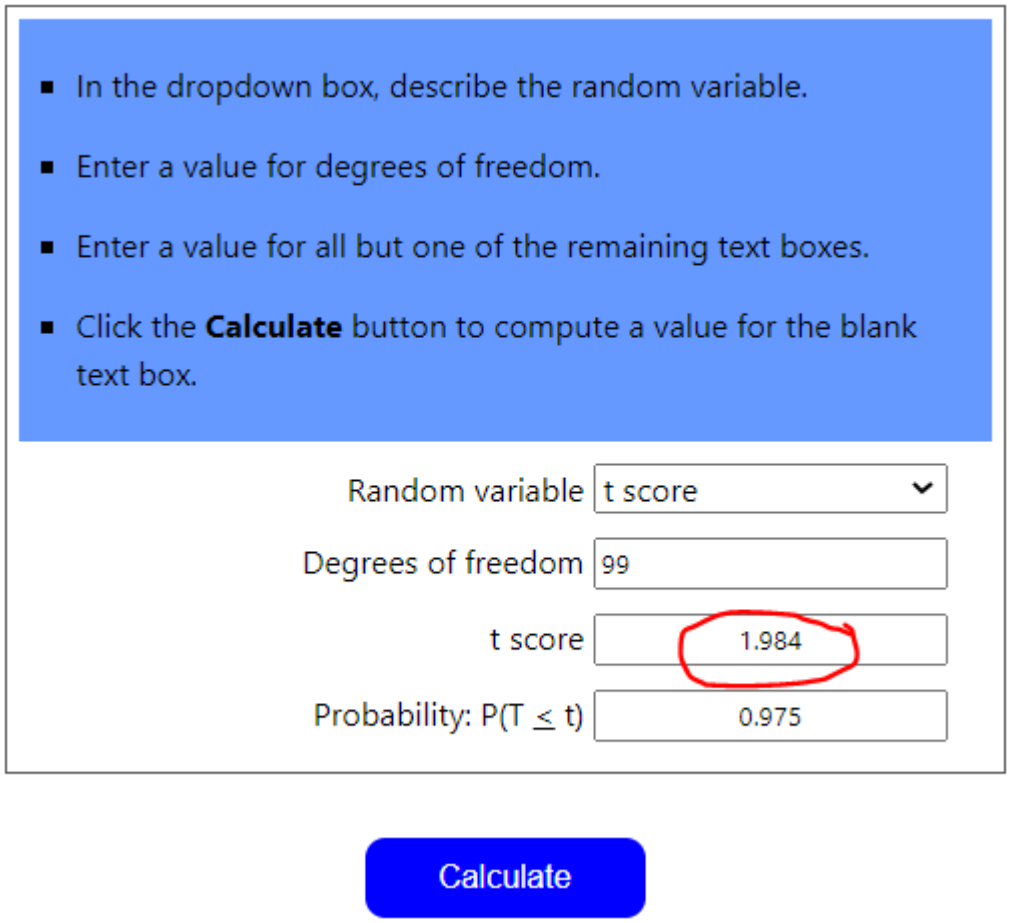
Out[24]: 120.3784999

The random sample mean even in this case does not belong to the confidence interval as per H_0. Hence we reject the Null hypothesis here as well

Repeat the first two steps of part 6 but this time use the t distribution instead of Z distribution

Assuming the population standard deviation is unknown then in that case, for decision making we use the sample mean $\bar{X}$ and the sample variance s^2.

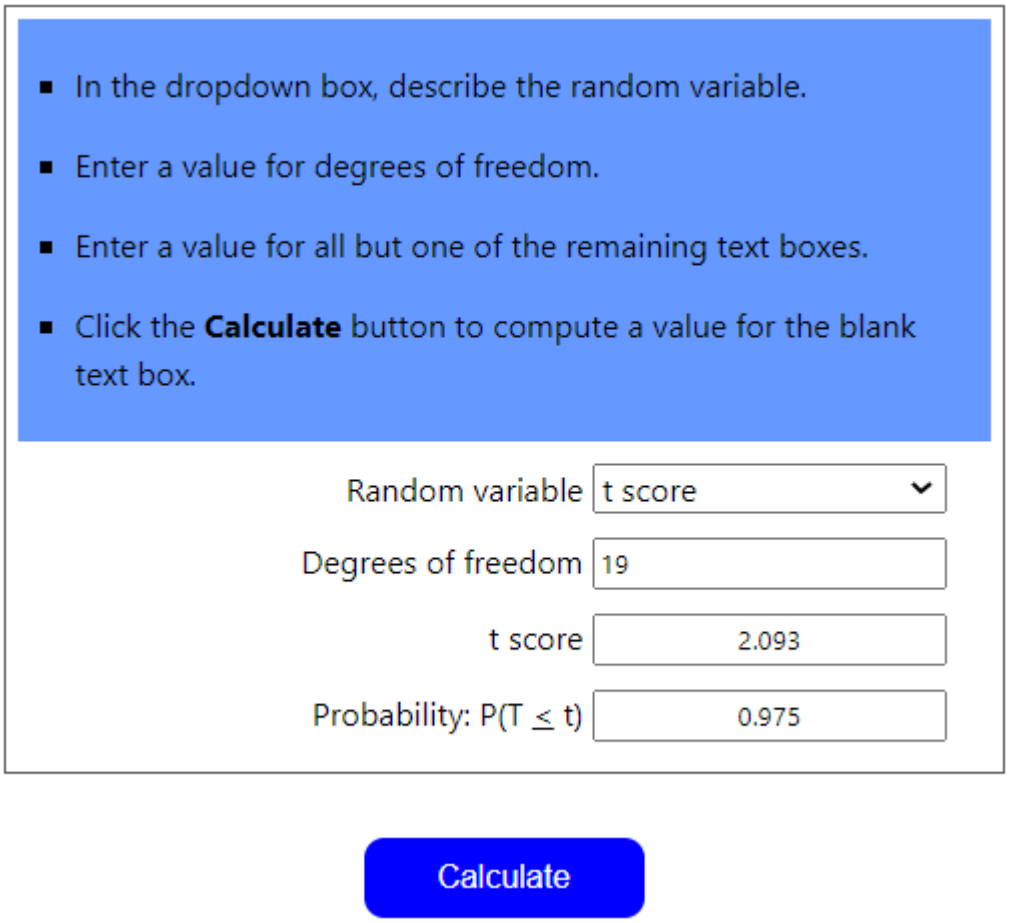
We know that in this case the sampling distribution for $\bar{X}$ is the t-distribution.



In [28]:
```
## The confidence interval for sample size 100 are as follows:-
a = 100 - (1.984*(np.std(random_sample_100)/np.sqrt(100)))
b = 100 + (1.984*(np.std(random_sample_100)/np.sqrt(100)))
conf_interval = (a,b)
conf_interval
```

Out[28]: (97.41230387883235, 102.58769612116765)

And now for the sample size 20 case:-



In [29]:
```
## The confidence interval for sample size 20 are as follows:-
a = 100 - (2.093*(np.std(random_sample_20)/np.sqrt(20)))
b = 100 + (2.093*(np.std(random_sample_20)/np.sqrt(20)))
conf_interval = (a,b)
conf_interval
```

Out[29]: (94.98304846181482, 105.01695153818518)

The confidence intervals for the Null Hypothesis using both Z values and t values are almost the same for both the random samples.

3) Use the same sample you picked up in Step 2) of Part 5 to test $H0$: σ=15 vs. Ha: $\sigma \neq 15$ at the significance level 0.05. What's your conclusion?

In [30]:
```
var_sample_20 = np.var(random_sample_20)
std_sample_20 = np.std(random_sample_20)
```

In [31]:
```
## sample Variance
var_sample_20
```

Out[31]: 114.91357895370618

In [32]:
```
df = 20-1 #degrees of freedom
chi_sq_stat = (df*(var_sample_20))/(225)
chi_sq_stat
```

Out[32]: 9.70381333386852

The chi-square hypothesis test is defined as:

H_0:	$\sigma^2 = \sigma_0^2$	
H_a:	$\sigma^2 < \sigma_0^2$	for a lower one-tailed test
	$\sigma^2 > \sigma_0^2$	for an upper one-tailed test
	$\sigma^2 \neq \sigma_0^2$	for a two-tailed test
Test Statistic:	$T = (N-1)(s/\sigma_0)^2$	
	where N is the sample size and S is the sample standard deviation. The key element of this formula is the ratio s/σ_0 which compares the ratio of the sample standard deviation to the target standard deviation. The more this ratio deviates from 1, the more likely we are to reject the null hypothesis.	
Significance Level:	α.	
Critical Region:	Reject the null hypothesis that the variance is a specified value, σ_0^2, if	
	$T > \chi^2_{1-\alpha,\, N-1}$	for an upper one-tailed alternative
	$T < \chi^2_{\alpha,\, N-1}$	for a lower one-tailed alternative
	$T < \chi^2_{\alpha/2,\, N-1}$ or $T > \chi^2_{1-\alpha/2,\, N-1}$	for a two-tailed alternative
	where $\chi^2_{.,\, N-1}$ is the critical value of the chi-square distribution with N - 1 degrees of freedom.	

Ours is the case of a two tailed test

We use the Chi-Square Table here to determine the values of $\chi^2_{\alpha/2,N-1}$ and $\chi^2_{1-\alpha/2,N-1}$

Link to the chi-square table - https://www.mathsisfun.com/data/chi-square-table.html

$\chi^2_{\alpha/2,N-1}$ for $N = 20$ and $\alpha = 0.05$ = $\chi^2_{0.025,19}$ = 32.852

$\chi^2_{1-\alpha/2,N-1}$ for $N = 20$ and $\alpha = 0.05$ = $\chi^2_{0.975,19}$ = 8.907

Clearly our Chi-Square test statistic lies between these two numbers hence we cannot reject H_0 and therefore, reject the alternate hypothesis H_1

4) Use the same sample you picked up in Step 2) of Part 5 to test $H0$: σ=15 vs. Ha: σ<15 at the significance level 0.05. What's your conclusion?

This is the case of a lower one tail test so we see whether the test statistic's value $< \chi^2_{\alpha,N-1}$

From the Chi Square table we see that for $N = 20$ and $\alpha = 0.05$, $\chi^2_{0.05,19}$ = 30.144. And our test statistic is less than this so we reject the Null Hypothesis in this case.

THE END

Part 7: Compare your data with a different data set

1) Consider the entire Gold column as a random sample from the first population, and the entire Oil column as a random sample from the second population. Assuming these two samples be drawn independently, form a hypothesis and test it to see if the Gold and Oil have equal means in the significance level 0.05.

In [1]:
```
## Import the Necessary Libraries
import pandas as pd
import numpy as np
import scipy
from sklearn.preprocessing import StandardScaler
import scipy.stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Load data and select first column
dataset = pd.read_excel("the data for your group project_MA541.xlsx")
```

In [3]:
```
dataset.columns
```

Out[3]:
```
Index(['Close_ETF', 'oil', 'gold', 'JPM'], dtype='object')
```

In [4]:
```
gold_data = dataset['gold']
oil_data = dataset['oil']
```

https://opentextbc.ca/introbusinessstatopenstax/chapter/comparing-two-independent-population-means/

The formula for the test statistic is given as:-

$$t_c = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(S_1)^2}{n_1} + \frac{(S_2)^2}{n_2}}}$$

In [14]:
```
def test_statistic(X1,X2):
    X1_mean = np.mean(X1)
    X2_mean = np.mean(X2)
    n1 = len(X1)
    n2 = len(X2)
    S1 = np.std(X1)
    S2 = np.std(X2)
    value = (X1_mean - X2_mean)/np.sqrt(((S1)**2/n1) + ((S2)**2/n2))
    return value
```

In [15]:
```
t_c = test_statistic(gold_data,oil_data)
```

In [17]:
```
## The test statistic (t-score) is calculated as follows:
t_c
```

Out[17]:
```
-0.48560947929481046
```

The critical value on the t-table using the degrees of freedom. The critical value,

1.96151, is found in the .025 column, this is $\alpha/2$, at 1528 degrees of freedom.

This shows that the test statistic is not greater than the critical value hence we cannot reject the Null Hypothesis.

Therefore, the two samples have the same means.

This entire analysis can be done in excel easily using the data analysis toolbar

t-Test: Two-Sample Assuming Unequal Variances		
	Variable 1	*Variable 2*
Mean	0.001030035	0.000662836
Variance	0.00044491	0.000127443
Observations	1000	1000
Hypothesized Mean Difference	0	
df	1528	
t Stat	0.485366614	
P(T<=t) one-tail	0.313742946	
t Critical one-tail	1.645851467	
P(T<=t) two-tail	0.627485893	
t Critical two-tail	1.961517728	

2) Subtract the entire Gold column from the entire Oil column and generate a sample of differences. Consider this sample as a random sample from the target population of differences between Gold and Oil. Form a hypothesis and test it to see if the Gold and Oil have equal means in the significance level 0.05.

In [20]:
```
diff_data = gold_data - oil_data
```

In [22]:
```
## Sample size
len(diff_data)
```

Out[22]: 1000

We formulate a Hypothesis test to see if the gold and oil have equal means in the significance level 0.05 as follows. We use the t-test statistic as done similarly in part 6

$$H_0 : \mu = 0$$

$$H_1 : \mu \neq 0 \text{ and } \alpha = 0.05$$

In [24]:
```
## The confidence interval for sample size 1000 are as follows:-
a = 0 - (1.984*(np.std(diff_data)/np.sqrt(1000)))
b = 0 + (1.984*(np.std(diff_data)/np.sqrt(1000)))
conf_interval = (a,b)
conf_interval
```

Out[24]: (-0.0013451277998628784, 0.0013451277998628784)

In [25]:
```
## Check if the Random Sample mean belongs to this interval or not
np.mean(diff_data)
```

Out[25]: -0.00036719941174700126

In [28]:
```
if conf_interval[0] <= np.mean(diff_data) <= conf_interval[1]:
    print("The sample mean of differences belongs to the confidence interval and hence
else:
    print("The sample mean of differences does not belongs to the confidence interval a
```

```
The sample mean of differences belongs to the confidence interval and hence we reject H
1
```

3) Consider the entire Gold column as a random sample from the first population, and the entire Oil column as a random sample from the second population. Assuming these two samples be drawn independently, form a hypothesis and test it to see if the Gold and Oil have equal standard deviations in the significance level 0.05.

F-Test Two-Sample for Variances		
	Variable 1	*Variable 2*
Mean	0.001030035	0.000662836
Variance	0.00044491	0.000127443
Observations	1000	1000
df	999	999
F	3.491057044	
P(F<=f) one-tail	3.9423E-82	
F Critical one-tail	1.109746138	

The F value is greater than F critical one-tail value. This implies we can reject the Null hypothesis. The Gold and Oil populations' standard deviations are not equal.

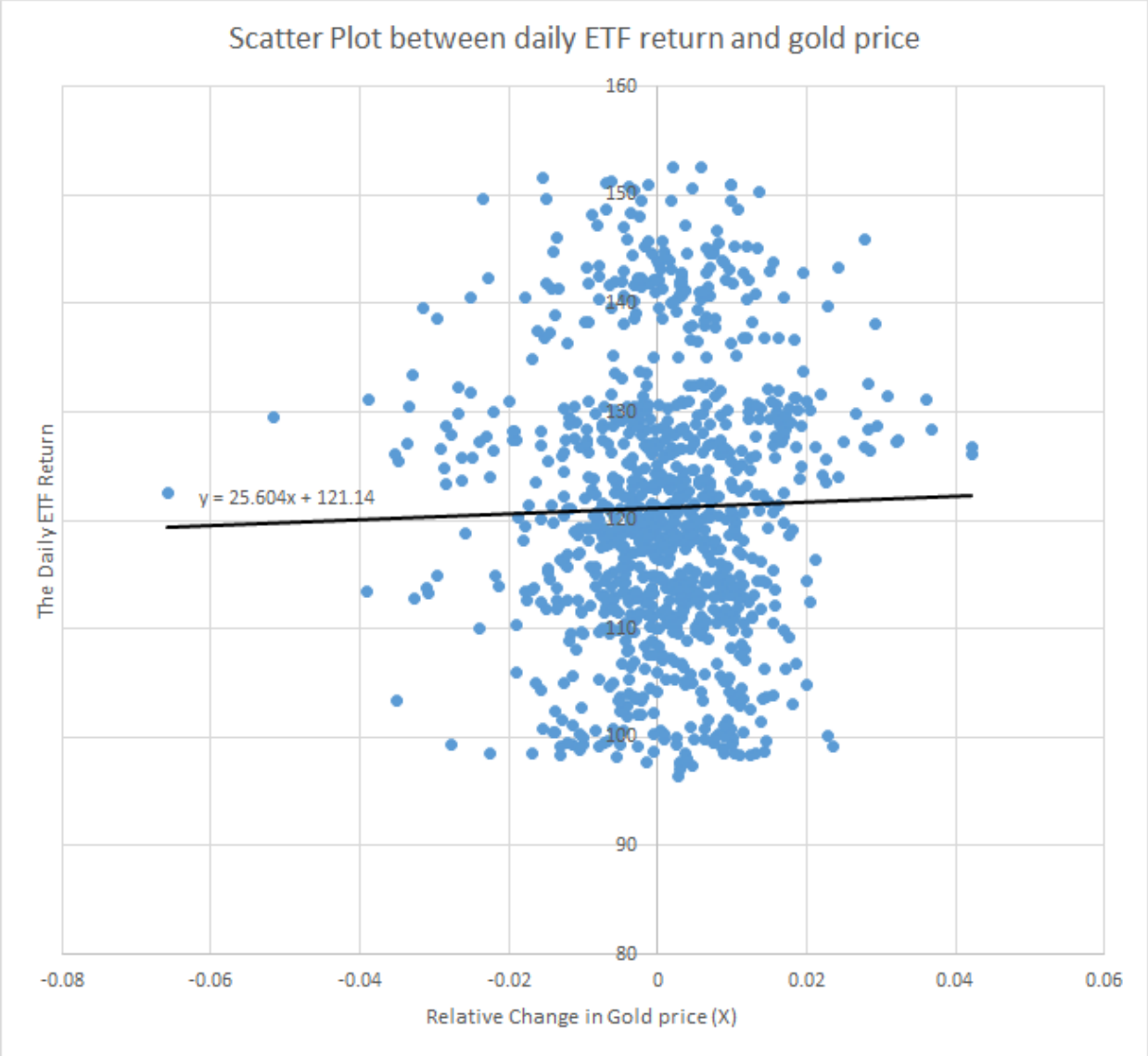
part 8

Part 8: Fitting the line to the data

Requirements –

Consider the data including the ETT column and Gold column only. Using any software,

1) Draw a scatter plot of ETF (Y) vs. Gold (X). Is there any linear relationship between them which can be observed from the scatter plot?
2) Calculate the coefficient of correlation between ETF and Gold and interpret it.



	Relative Change in Gold Price	*The Daily ETF Return*
Relative Change in Gold Price	1	
The Daily ETF Return	0.02299557	1

The coefficient of correlation is very close to zero.

Almost 0 correlation indicates independence between the two attrbutes.

3) Fit a regression line (or least squares line, best fitting line) to the scatter plot. What are the intercept and slope of this line? How to interpret them?

As shown in the picture above, the equation of the best fit line is made visible. The coefficients of linear regression can also be seen below.

Visually we can see that the regression line is almost parallel to the X-axis. This means, that the variability in the dependent variable is hardly explained by the independent variable as the slope is almost zero. Hence, the y intercept should be the only parameter that explains the total variance of the daily Returns. And since this is a constant term it should be the "Mean of the dependent variable". In other words, the regression line merely outputs the mean of the dependent variable for every value of the independent variable.

Regression analysis output: Coefficients

SUMMARY OUTPUT	
Regression Statistics	
Multiple R	0.02299557
R Square	0.000528796
Adjusted R Square	-0.000472678
Standard Error	12.57276069
Observations	1000

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	121.1359885	0.398271024	304.1546615	0	120.3544438	121.9175332	120.3544438	121.9175332
X Variable 1	25.60438932	35.236285	0.726648377	0.467611781	-43.54131771	94.75009636	-43.54131771	94.75009636

https://www.ablebits.com/office-addins-blog/2018/08/01/linear-regression-analysis-excel/

The standard error regression statistic in the summary output is another goodness-of-fit measure that shows the precision of our regression analysis. It is an absolute measure that shows the average deviation (distance) of the data points from the regression line. Larger the value of this statistic, lesser is the precision of the regression line.

https://www.statisticshowto.com/probability-and-statistics/excel-statistics/excel-regression-analysis-output-explained/

Slope and intercept of the linear regression line are 121.136 and 25.604 respectively.

The standard error in measuring the slope is larger than the slope coefficient. This means the coefficient is probably not different from zero.

4) Conduct a two-tailed t-test with $H0$: $\beta1=0$. What is the P-value of

the test? Is the linear relationship between ETF (Y) and Gold (X) significant at the significance level 0.01? Why or why not?

http://www.stat.yale.edu/Courses/1997-98/101/anovareg.htm

The analysis of variance ANOVA table of the Regression analysis can help in finding the P-value for this test.

ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	83.46606036	83.46606036	0.528017864	0.467611781
Residual	998	157758.1628	158.0743114		
Total	999	157841.6289			

The "F" column provides a statistic for testing the hypothesis that $\beta_1 \neq 0$ against the null hypothesis that $\beta_1 = 0$.

The Value F in the above table is calculated as follows:-

$$F = \frac{MeanSquareModel(MSM)}{MeassquareError(MSE)} = \frac{83.46606}{158.074311} = 0.528018$$

$$MSM(MSRegression) = \frac{SSRegression}{dfRegression} = \frac{83.46606}{1} = 83.46606$$

$$MSE(MSResidual) = \frac{SSResidual}{dfResidual} = \frac{157758.162826693}{998} = 158.074311$$

The test statistic is the ratio MSM/MSE, the mean square model term divided by the mean square error term. When the MSM term is large relative to the MSE term, then the ratio is large and there is evidence against the null hypothesis.

But in our case the statistic is small and hence we don't have sufficient evidence to support the Alternate hypothesis. For simple linear regression, the statistic MSM/MSE has an F distribution with degrees of freedom (DFM, DFE) = (1, n - 2).

https://statisticsbyjim.com/regression/interpret-coefficients-p-values-regression/

We use the F table to ge the p-value for this test. The Significance F (p value) in the ANOVA table of the Regression analysis in excel also outputs this required pvalue and it equals 0.468. We see that it is not less than the significance level of 0.01. So therefore, we can interpret that the data has no sufficient evidence to reject the Null Hypothesis and there is no correlation between the dependent and the independent variable. Thus we conclude that $\beta_1 = 0$

There is no signficant linear relationship between ETF (Y) and Gold (X) as per the test.

5) Suppose that you use the coefficient of determination to assess the quality of this fitting. Is it a good model? Why or why not?

The Coefficient of determination is given by the value of r^2, which is used as an indicator of the goodness of fit. It shows how many points fall on the regression line. For example, 80% means that 80% of the variation of y-values around the mean are explained by the x-values. In other words, 80% of the values fit the model.

It is calculated as follows:-

$$r^2 = \frac{Sum of Square of the model}{Total Sum of Square} = \frac{83.4660603638622}{157841.628887057} = 0.00053$$

This formalizes the interpretation of r^2 as explaining the fraction of variability in the data explained by the regression model.

In our case the r^2 value is very less. Only 0.053% of the dependent variable values are explained by the independent variable.

FYI - The r^2 value can also be also be calculated as the square of the correlation coefficient.

This shows that the regression line is a very bad fit on our dataset.

The linear regression analysis using the data analysis toolkit in excel provides this value directly under summary output.

6) What are the assumptions you made for this model fitting?

1.) The regression model is linear in the coefficients and the error term

2.) The error term has a population mean of zero

3.) All independent variables are uncorrelated with the error term

4.) Observations of the error term are uncorrelated with each other

5.) The error term has a constant variance (no heteroscedasticity)

6.) No independent variable is a perfect linear function of other explanatory variables

7.) The error term is normally distributed (optional)

7) Given the daily relative change in the gold price is 0.005127. Calculate the 99% confidence interval of the mean daily ETF return, and the 99% prediction interval of the individual daily ETF return.

Confidence Interval for the forecasted Value				Prediction Interval for the forecasted Value			
n	1000	E2					
df	998	E3					
mean(x)	0.000662836	E4					
x0	0.005127	E5					
y0_pred	121.2672622	E6		121.2673			
Standard Error	12.5727607	E7					
Sum of Square Deviation of the data points from their sample means	0.127315439	E8					
SE	0.427571953	E9		12.58003			
t-crit	2.580764586	E10		2.580765			
Lower	120.1637996			88.80117			
Upper	122.3707248			153.7334			

https://www.real-statistics.com/regression/confidence-and-prediction-intervals/

In [ ]: