MA 541 Statistical Methods
Course Project Report

May 12, 2021

AUTHORS:

Student 1: Siddhesh Shaji (CWID: 10458882),
Student 2: Amit Singh (CWID: 10465317),
Student 2: Yogesh Awdhut Gadade (CWID: 10467214),
Student 2: Mandar Parab (CWID: 10473518)

PROJECT SUPERVISOR AND ASSESSOR:

Dr. Xiao (Sophia) Zhong

Contents

1 Section 1 - Objective of the report

Objective:

The objective of this project is to perform exploratory data analysis on the given sample data using various statistical methods like hypothesis testing, Central Limit theorem, Confidence interval, Visualization, etc. and verify its candidacy for simple and multi linear regression models.

2 Section 2 - Part 1 to Part 10

Part 1: Meet the data

Mean: The mean is used to measure of central tendency data.

Population Mean	**Sample Mean**
$\mu = \frac{\sum x}{N}$	$\bar{X} = \frac{\sum x}{n}$

Standard deviation : Standard deviation is a number used to tell how data are spread out from the mean or expected value.

$$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{n}}$$

where,

σ = population standard deviation
$\sum$ = sum of...
μ = population mean
n = number of scores in sample.

Data consist of EFT, Oil, Gold, and JPM columns with 1000 sample size.

Mean of all the columns :
ETF Mean : 121.1529600120001
Oil Mean : 0.0010300354937470015
Gold Mean : 0.0006628360819999998
JPM Mean : 0.0005304110210000002

Standard deviation of all the columns :
ETF standard deviation : 12.563503845944297
Oil standard deviation : 0.021082349463798354
Gold standard deviation : 0.011283414317347945
JPM standard deviation : 0.011011052723643009

Co-relation of the above variables is as follows:

	Close_ETF	oil	gold	JPM
Close_ETF	1.000000	-0.009045	0.022996	0.036807
oil	-0.009045	1.000000	0.235650	-0.120849
gold	0.022996	0.235650	1.000000	0.100170
JPM	0.036807	-0.120849	0.100170	1.000000

As it can be seen from the co-relation matrix in the first column of the matrix the magnitude is below 0.5. If the magnitude is close 1 or -1 then it is considered as strongly co-related variable.

On the basis of the magnitude we can say that there is a weak linear co-relation between the variables. Also oil is negatively co-related with ETF, and gold and JPM are positively co-related with ETF.

Part 2: Describe your data

1) A histogram for each column.
The purpose of a histogram is to graphically summarize the distribution of a data.

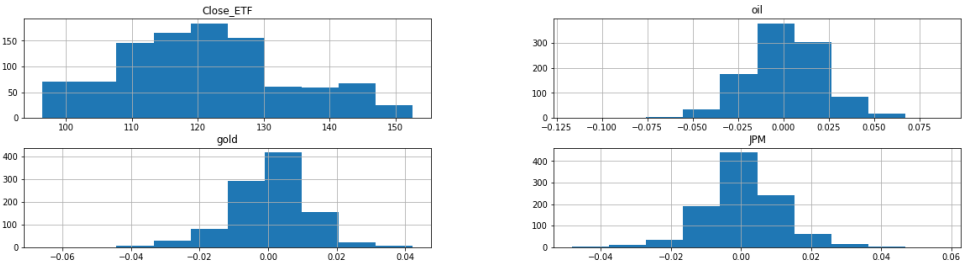


Figure 1: histograms for ETF, Oil, Gold, JPM

Showing histogram of each columns. By histogram its look like each column approximately follows normal distribution.

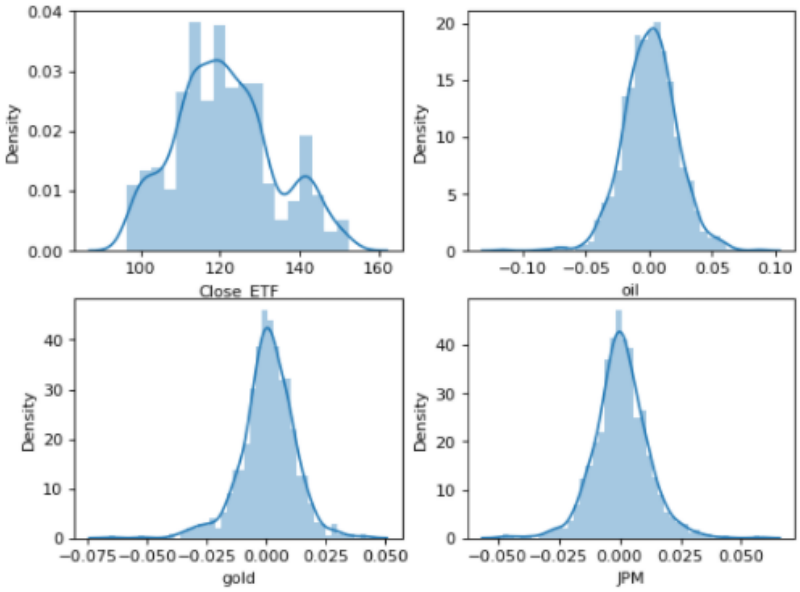


Figure 2: Density graph for ETF, Oil, Gold, JPM

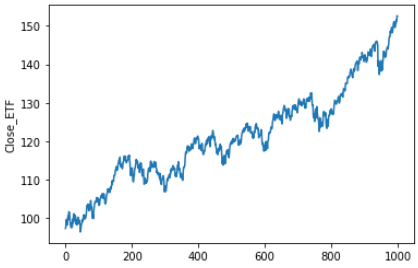
2) A time series plot for each column

Time series graphs can be used to visualize trends in counts or numerical values over time or some other ordering.

	Close_ETF	oil	gold	JPM
0	97.349998	0.039242	0.004668	0.032258
1	97.750000	0.001953	-0.001366	-0.002948
2	99.160004	-0.031514	-0.007937	0.025724
3	99.650002	0.034552	0.014621	0.011819
4	99.260002	0.013619	-0.011419	0.000855

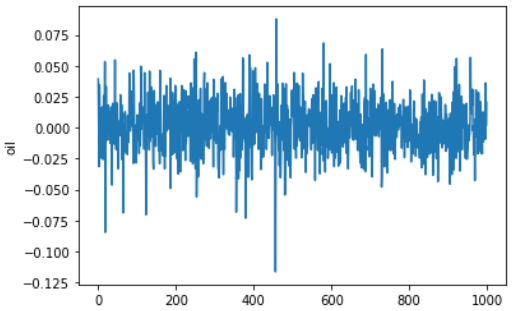
Figure 3: Value of columns ETF, Oil, Gold, JPM

ETF



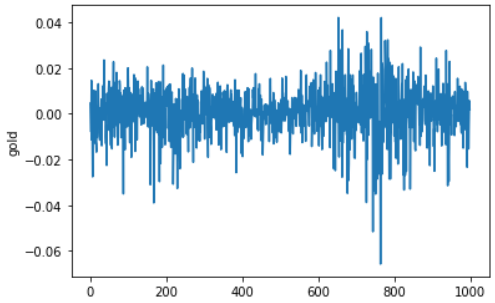
ETF graph showing upward trend.

Oil



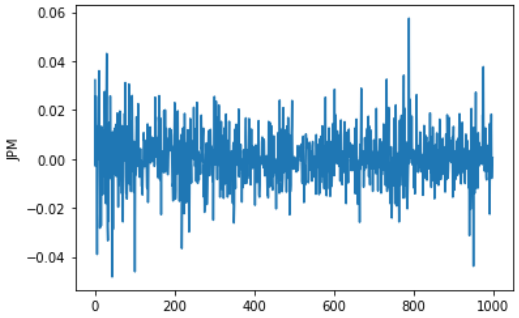
Oil graph showing random variation.

Gold



Gold graph showing random variation.

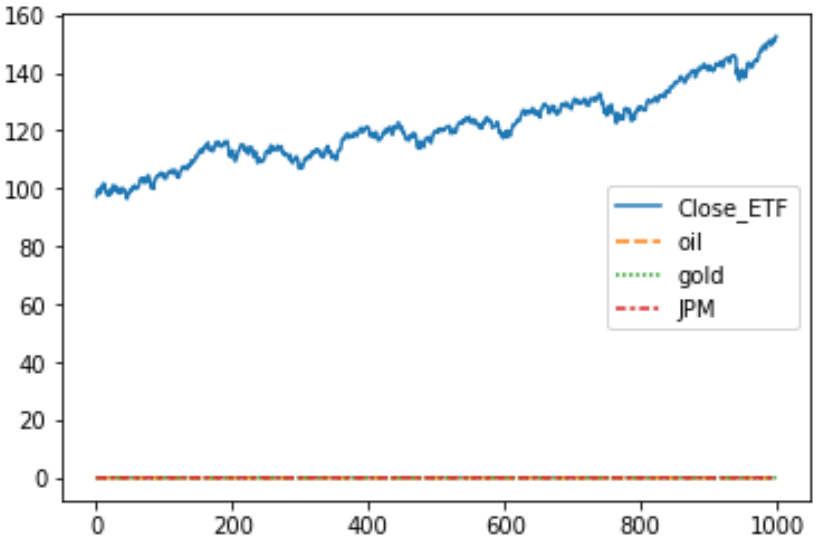
JPM



JPM graph showing random variation.

3) A time series plot for all four columns.

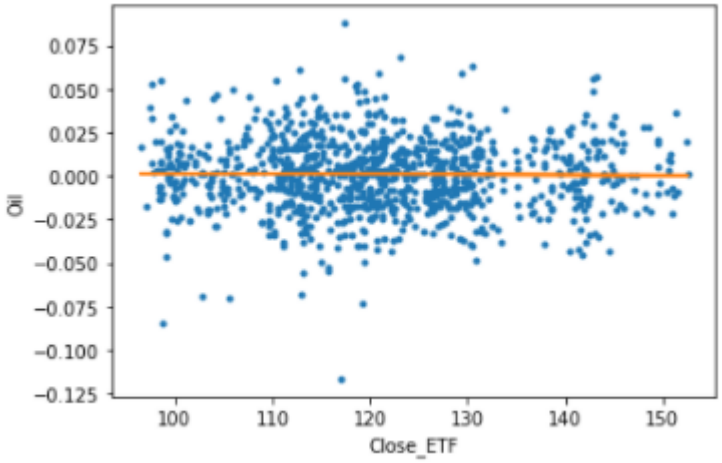
Time series graph showing all four columns:



ETF column showing upward variation and other three column showing random variation.

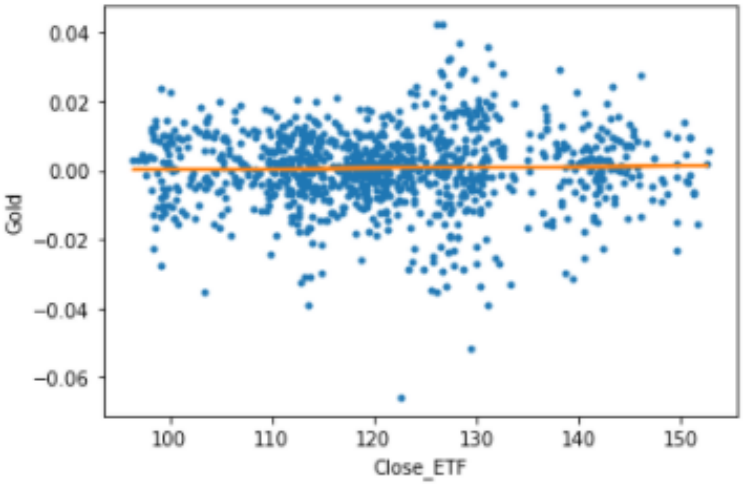
4) Three scatter plots to describe the relationships between the ETF column and the OIL column; between the ETF column and the GOLD column; between the ETF column and the JPM column, respectively.
A scatter plot is used to observe and visually display the relationship between variables.

ETF column and the OIL column



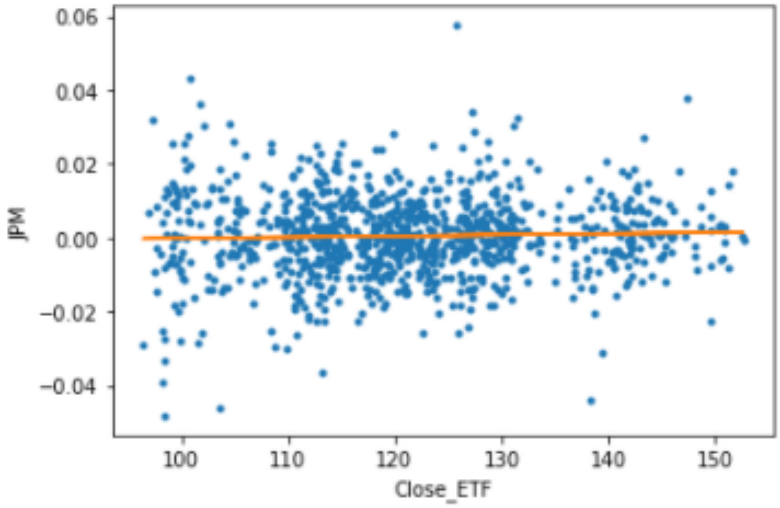
By scatter plot we can observe that there is no relation between ETF and oil.

ETF column and the Gold column



By scatter plot we can observe that there is no relation between ETF and Gold.

ETF column and the JPM column



By scatter plot we can observe that there is no relation between ETF and JPM.

Part 3: What distribution does your data follow

We will check if data is normally distributed or not using below methods :

Graphs for Normality test

1) Histogram

A histogram is a graphical display of data using bars of different heights. A histogram is the most used graph to show frequency distributions.

2) Quantile-Quantile Plot

Quantile-Quantile or QQ plot used to check the distribution of a data sample. A perfect match for the distribution will be shown by a line of dots on a 45-degree angle from the bottom left of the plot to the top right. Often a line is drawn on the plot to help make this expectation clear. Deviations by the dots from the line shows a deviation from the expected distribution

Statistical Tests for Normality

3) Chi-Square Test for Normality

The chi-square goodness of fit test can be used to test the hypothesis that data comes from a normal hypothesis

$$X^2 = \sum \frac{(observed - expected)^2}{(expected)}$$

4) Shapiro-Wilk Test

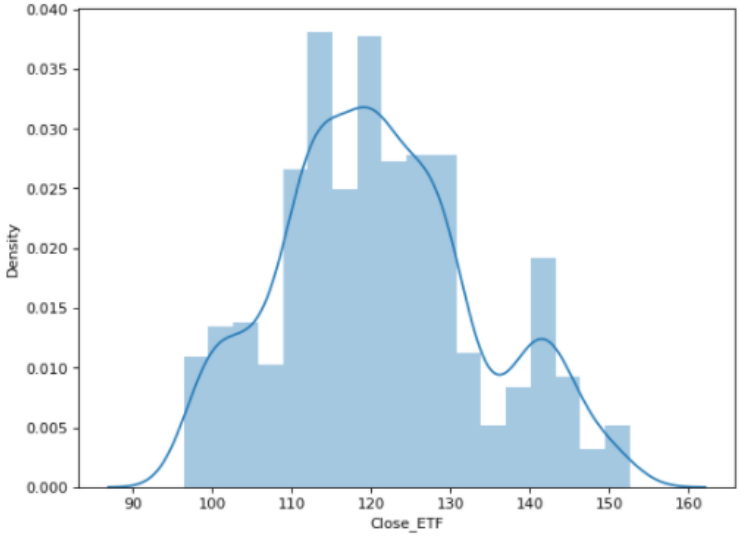
The Shapiro Wilk test checks if the normal distribution model fits the observations.

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2},$$

xi = are the ordered random sample values.
ai = are constants generated from the co-variances, variances and means of the sample (size n) from a normally distributed sample.
W = small values indicate your sample is not normally distributed.

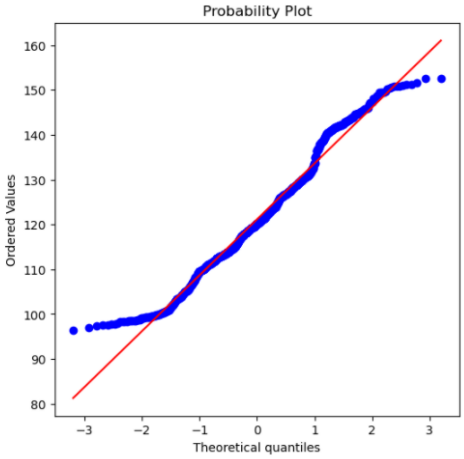
ETF data :

1) Histogram



Conclusion from histogram: Does not look like smooth normal distribution. Looks skewed and bi-modal.

2) Quantile-Quantile Plot



Conclusion from QQ Plot: We can see at many places data is deviating from

line. This shows it is not exact normal distribution.

Assumption for below test with alfa 0.05
Ho : ETF follows normal distribution.
H1 : ETF does not follows normal distribution.

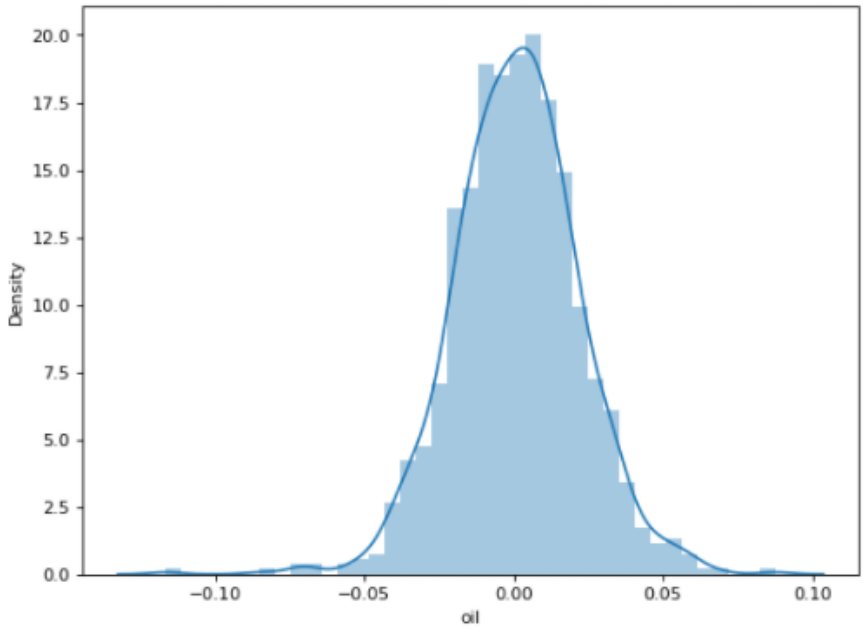
3) Chi-Square Test for Normality
Calculated using Python:
Statistics=1302.829, p=0.000
Sample does not look Gaussian (reject H0)

4) Shapiro-Wilk Test
Calculated using Python:
Statistics=0.980, p=0.000
Sample does not look Gaussian (reject H0)

Conclusion for ETF column: After performing above 4 normality tests, we conclude it is not exactly following normal distribution.

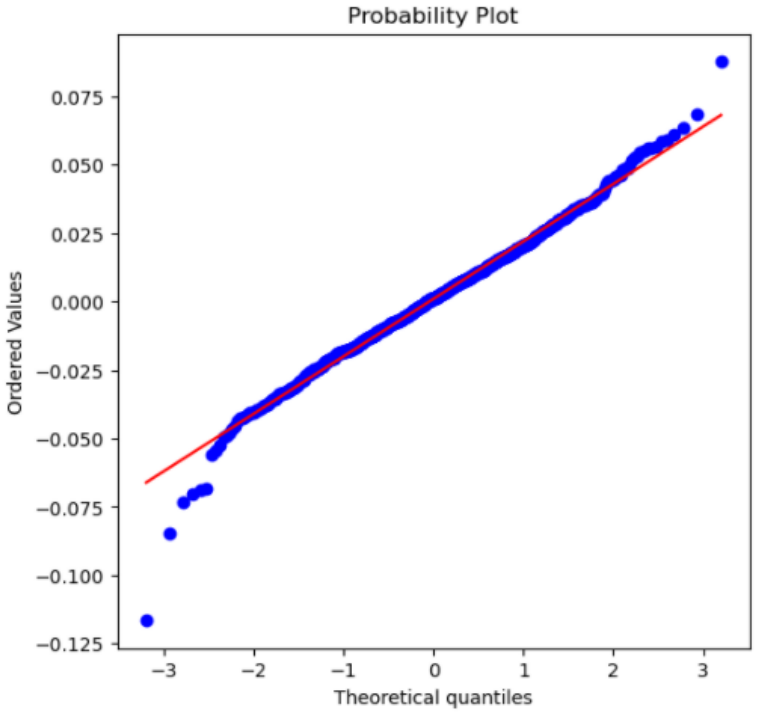
Oil data :

1) Histogram



Conclusion from histogram : Approximately looks like normal distribution.

2) Quantile-Quantile Plot



Conclusion from QQ Plot : Most of line is on line except few outliers. This shows its follow normal distribution.

Assumption for below test with alfa 0.05
Ho : Oil follows normal distribution.
H1 : Oil does not follows normal distribution.

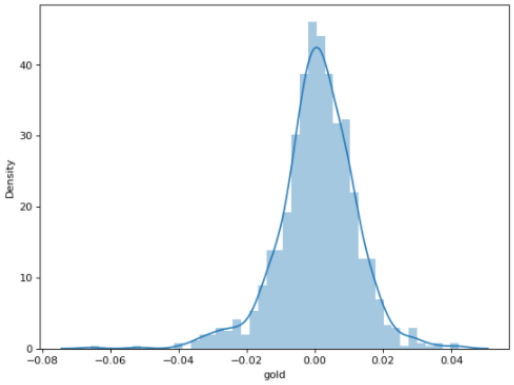
3) Chi-Square Test for Normality
Statistics=431.505, p=1.000
Sample looks Gaussian (fail to reject H0)

4) Shapiro-Wilk Test
Statistics=0.989, p=0.000
Sample does not look Gaussian (reject H0)

Conclusion for Oil column : After performing above 4 normality tests, we conclude its approximately following normal distribution.

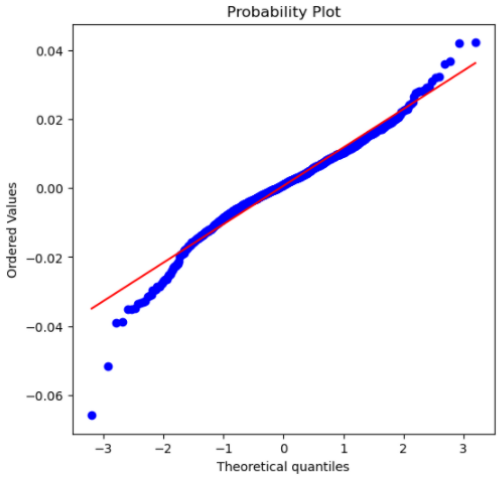
Gold data :

1) Histogram



Conclusion from histogram : Approximately looks like normal distribution.

2) Quantile-Quantile Plot



Conclusion from QQ Plot : Most of line is on line except few outliers. This shows its follow normal distribution.

Assumption for below test with alfa 0.05
Ho : Gold follows normal distribution.
H1 : Gold does not follows normal distribution.

3) Chi-Square Test for Normality
Statistics=192.077, p=1.000
Sample looks Gaussian (fail to reject H0)

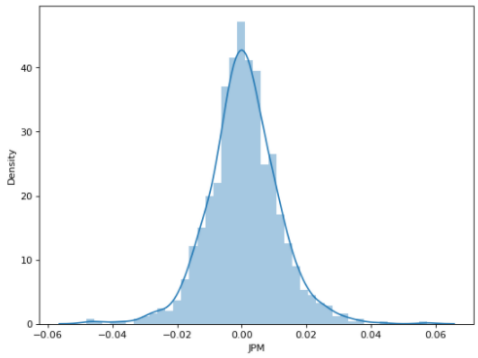
4) Shapiro-Wilk Test
Statistics=0.969, p=0.000

Sample does not look Gaussian (reject H0)

Conclusion for gold column : After performing above 4 normality tests, we conclude it is approximately following normal distribution.

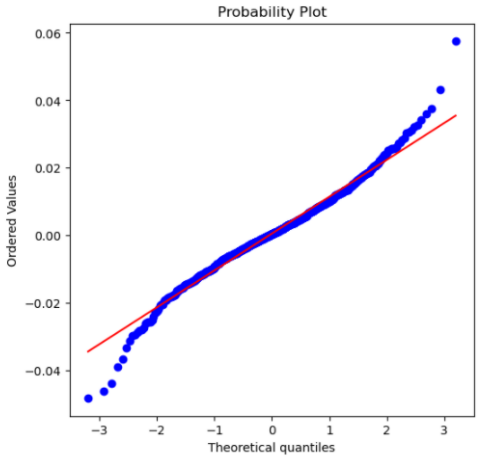
JPM data :

1) Histogram



Conclusion from histogram : Approximately looks like normal distribution.

2) Quantile-Quantile Plot



Conclusion from QQ Plot : Most of line is on line except few outliers. This shows its follow normal distribution.

Assumption for below test with alfa 0.05
Ho : JPM follows normal distribution.
H1 : JPM does not follows normal distribution.

3) Chi-Square Test for Normality
Statistics=228.584, p=1.000
Sample looks Gaussian (fail to reject H0)

4) Shapiro-Wilk Test
Statistics=0.980, p=0.000
Sample does not look Gaussian (reject H0)

Conclusion for JPM column : After performing above 4 normality tests, we conclude it is approximately following normal distribution.

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

The central limit theorem states that if you have a population with mean and standard deviation and take sufficiently large random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed, regardless of that variable's distribution in the population.

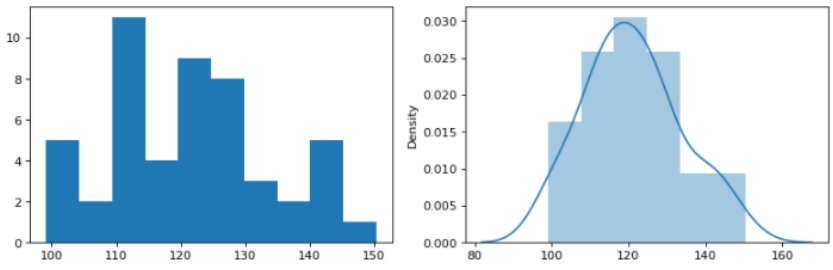
If sample size is small, the shape of sampling distribution of sample mean is skew to right or skew to left.

If sample size is sufficiently large, the shape of sampling distribution of sample mean is approximately normally distributed.

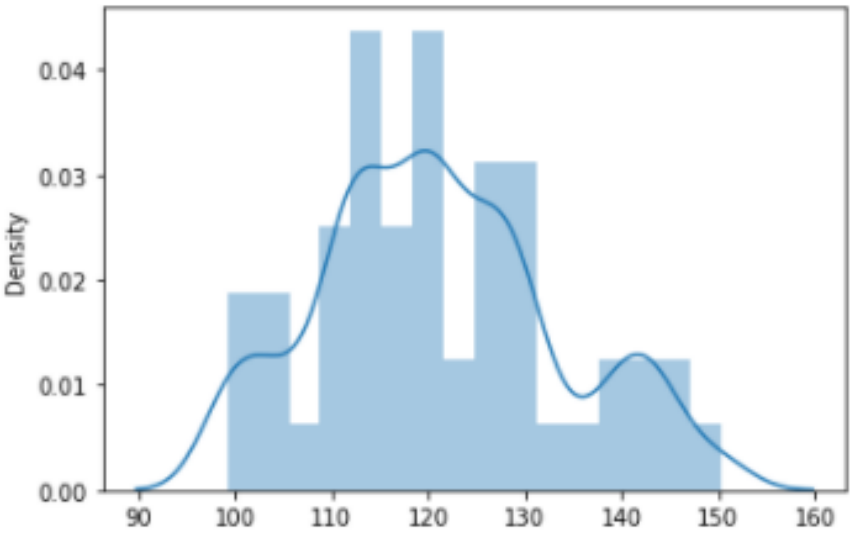
1) Calculate the mean and the standard deviation the population.

Mean of ETF column : 121.1529600120001
Standard Deviation ETF column : 12.563503845944297

2) Break the population into 50 groups sequentially and each group includes 20 values.
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.
4) Calculate the mean and the standard deviation 0f the data including these sample means. Make a comparison between mean and samples mean, between standard deviation.



If we take 50 groups sequentially 1000 times, then we will get below his-



togram.

Mean of ETF column : 121.1529600120001
Standard Deviation ETF column : 12.563503845944297

Comparison between Mean and Sample mean:

Mean of population: 121.1529600120001 and mean of samples : 121.15296001199997

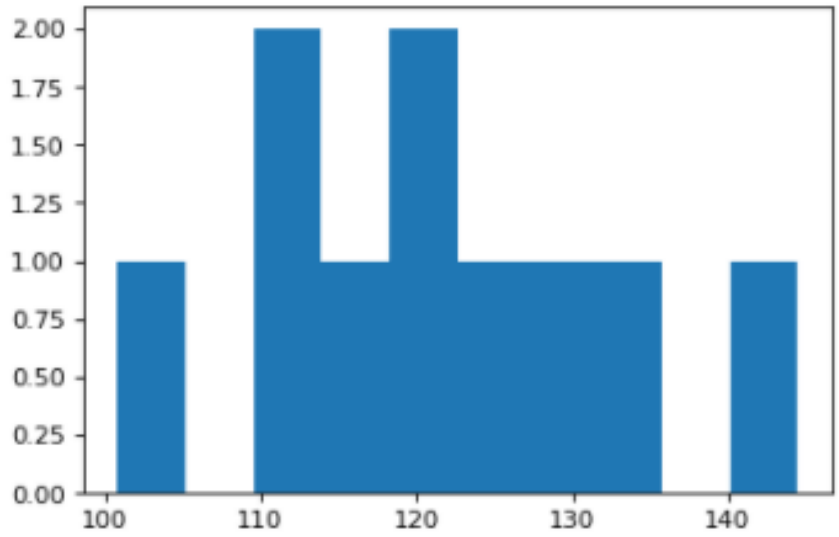
Mean of population and sample mean are almost same.

sd/sqrt(n) : 1.2563503845944297 and standard deviation of samples : 12.16375686089257

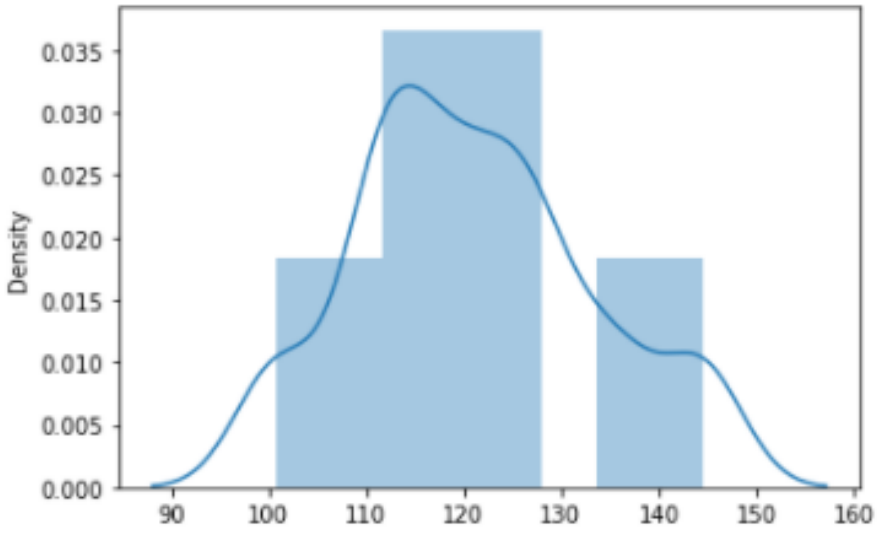
5) Are the results from Items 3) and 4) consistent with the Central Limit Theorem? Why?

Conclusion : We didn't get approximate normal distribution as we have used sequential sample and sample size of 20. This not consistence with Central limit theorem. Normal distribution still like Part 3.

6) Break the population into 10 groups sequentially and each group includes 100 values.
7) Repeat Items 3) 5).



If we take 20 groups sequentially 1000 times, then we will get below histogram.



Mean of sample : 121.15296001199997
Standard deviation of sample : 12.16375686089257

Comparison between Mean and Sample mean:

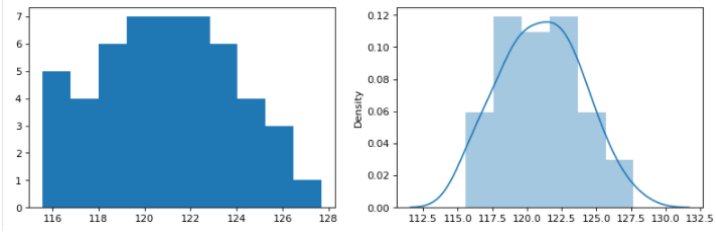
Mean of population : 121.1529600120001 and mean of samples : 121.15296001199997

Mean of population and sample mean are almost same.

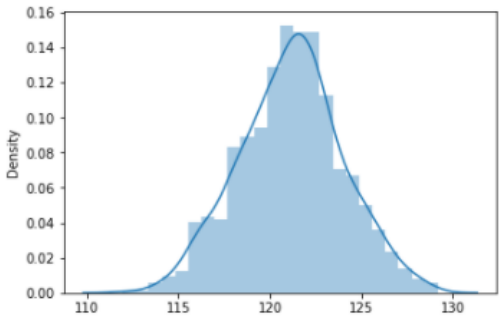
sd/sqrt(n) : 1.2563503845944297
Standard deviation of samples : 12.16375686089257

Conclusion : We didn't get approximate normal distribution as we have used sequential data. Sight improvement over 20 size sample. It not consistence with Central limit theorem.

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.
9) Repeat Items 3) 5).



If we take 50 groups random sample 1000 times, then we will get below histogram.



Mean : 120.983180221
Standard deviation : 2.6861368907082497

Comparison between Mean and Sample mean:

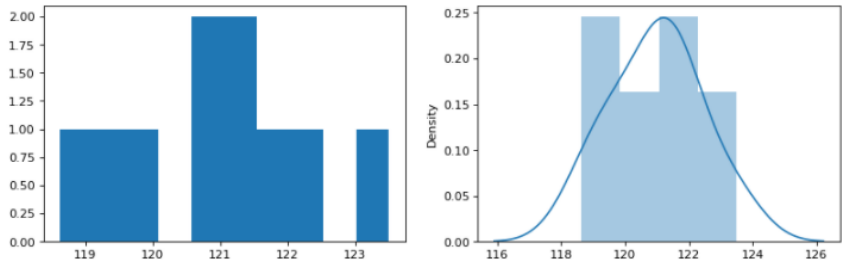
Mean of population : 121.1529600120001 and mean of samples : 120.983180221

Mean of population and sample have sight difference.

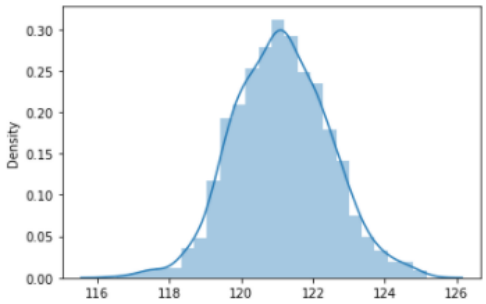
sd/sqrt(n) : 1.7767477529860964 and standard deviation of samples : 2.6861368907082497

Conclusion : We get approximate normal distribution as we have used random data sample. But still need improvement as sample size is 20. Improvement over sequential data and part 3 distribution. Its consistent with Central Limit Theorem.

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.
11) Repeat Items 3) 5).



If we take 20 groups random sample 1000 times, then we will get below histogram.



Mean : 120.763279982

Standard deviation : 1.6665020434714146

Comparison between Mean and Sample mean:

Mean of population : 121.1529600120001 and Mean of samples : 120.763279982

Mean of population and sample have sight difference.

sd/sqrt(n) : 1.7767477529860964 and Standard deviation of samples : 1.6665020434714146

Conclusion : We get approximate normal distribution as we have used random data sample of 100. Improvement over sequential data, random sample with 20 size and part 3 distribution. It is consistent with Central Limit Theorem.

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.

The central limit theorem states that if you have a population with mean and standard deviation and take sufficiently large random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed.
In part 3 we observed that ETF column was not following normal distribution.

Sequential sample :
We observed that when we took 50 sample sequentially, then histogram of sample means is like histogram of population in part 3.
We observed that when we took 100 sample sequentially, then histogram of sample means is smoother than 50 group sample but still like histogram of population in part 3.

Random sample :
We observed that when we took 50 sample randomly with replacement, then histogram of sample means is better normally distributed than sequential sample and distribution in part 3.
We observed that when we took 100 sample randomly with replacement, then histogram of sample means forms smoother normal distribution than 50 sample randomly with replacement, sequential sample data and distribution in part 3.

Conclusion :
Thus we can conclude that as we add more and more random samples with replacement allowed the shape of the graph of distribution tends to normal (bell shaped) distribution.

Part 5: Construct a confidence interval with your data

About Confidence intervals:

Confidence intervals are estimate ranges for population parameters based on the sample observations. The two main factors that decide the range of these estimates are:- 1.The sample size and 2.The population variance. Greater the sample size, greater is the amount of information/data we have to make a good guess regarding the range; which leads to a small (more accurate) range for predicting the population parameter and vice versa. Greater the population variance, greater is the deviation in values which makes it tougher to predict the population parameter with certainty. This causes the range of the intervals to increase and vice versa.

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.

   Solution:

   For this particular question, we are supposed to take one of the 10 randomly generated samples(with 100 data points each) from step 10 of part 4 and calculate an interval with 95 percent confidence, estimating the population mean μ.

   We have used the t.norm.interval() method available in the scipy library. We have made use of the normal distribution and not the student's t distribution in order to compute the interval since we have the sample size n = 100 (which is > 20).

   The formula for μ is given by:-

$$\overline{\mathrm{X}} \pm t \frac{\mathrm{s}}{\sqrt{\mathrm{n}}}$$

   We have the sample mean(x-bar), the sample standard deviation(s) and the sample size(n). Now all we have to do is calculate the t-value cor-

responding to our sample's degrees of freedom and significance value(95 percent).This t-value was found out to be 1.960.

After calculation, the result that we obtained was (119.4678, 124.8443).

Just to confirm the result, we also made use of the st.norm.interval() method offered by the scipy library which is used to calculate the confidence interval for samples. And the results obtained were the same up to 1 decimal point.

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.

Solution:

For this particular question, we are supposed to take one of the 50 randomly generated samples(with 20 data points each) from step 8 of part 4 and calculate an interval with 95 percent confidence, estimating the population mean μ.

We have used the Wilcoxon signed ranked test in r in order to get the results, since the etf feature is not perfectly normal and on top of that the sample size is less than 30.

```
> wilcox.test(x, mu=122.156100, conf.int = T, conf.level=0.95)

    Wilcoxon signed rank test

data:  x
V = 136, p-value = 0.2611
alternative hypothesis: true location is not equal to 122.1561
95 percent confidence interval:
 119.620 132.395
```

The result that we obtained is (119.620 , 132.395).

Also, just for the sake of it, we also tried calculating the confidence interval using the student-t distribution method. scipy.stats.t.interval() method available in the scipy library since the size of the sample size is small. The formula for which is is the same as above, the difference would be in the

values of sample mean which in this case is 125.42, degrees of freedom, sample standard deviation (12.20) and the sample size(20).

The result obtained using this method was (119.56, 131.28).

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?

   Solution:

   The population mean (121.1529) is included in the confidence intervals calculated by both the samples. The confidence interval calculated in step 1 is obviously more accurate since the sample size in that case is larger(5 times) than the sample size from step 2.

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

About Hypothesis testing:

Hypothesis testing is a procedure to take an inferential decision regarding a business problem with help of statistics. Every hypothesis problem consists of two parts:- 1. Null hypothesis and 2. Alternate hypothesis. In case of Z-test, the null hypothesis is rejected when the z-statistic lies on the rejection region, which is determined by the significance level and the type of tail (two-tailed, left-tailed or right-tailed).

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

   Solution:

   Since in this case, the population standard deviation is known and the number of samples is big enough, we will make use of the z-test.

   The formula for the z-statistic is given by :-

$$z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$$

   We get the value of z as 17.63

   Now corresponding to the significance value given (0.05), we need to find the z-critical value for it, which is obtained from the z-table as shown below:-

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Therefore, the rejection region for this two-tailed test is $R = z : |z| > 1.96$ We can also confirm this using the p-value approach, the p-value is coming out to be 0 in this case.

Hence, we reject the null hypothesis in both the methods.

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

Solution:

Since the size of the data-sample in question is less and the distribution is not perfectly normal, we decided to go with the Wilcoxon signed ranked test in r. The test type was given as "two-sided" and the hypothesis mean was taken as 100.

```
> wilcox.test(x, mu=100, conf.int = T, conf.level=0.95)

    Wilcoxon signed rank test

data:  x
V = 209, p-value = 3.815e-06
alternative hypothesis: true location is not equal to 100
95 percent confidence interval:
 119.620 132.395
sample estimates:
(pseudo)median
        126.19
```

Since the p-value came out to be so less, we reject the null hypothesis.

3. Use the same sample you picked up in Step 2) of Part 5 to test $H_0 : \sigma = 15$ vs $H_a : \sigma \neq 15$ at the significance level 0.05. What's your conclusion?

Solution:

We will be using the chi-square technique in order to solve this part since it is used to test if the variance of a population is equal to a specified value.

The test statistic for chi-square test is given by -

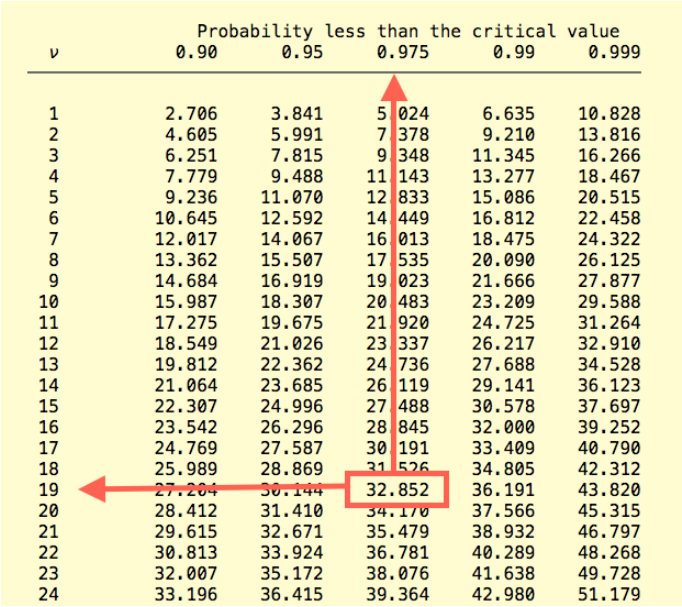
Test Statistic:	$T = (N-1)(s/\sigma_0)^2$

Where (N-1) is degrees of freedom,s is the sample standard deviation and sigma0 is the hypothetical standard deviation

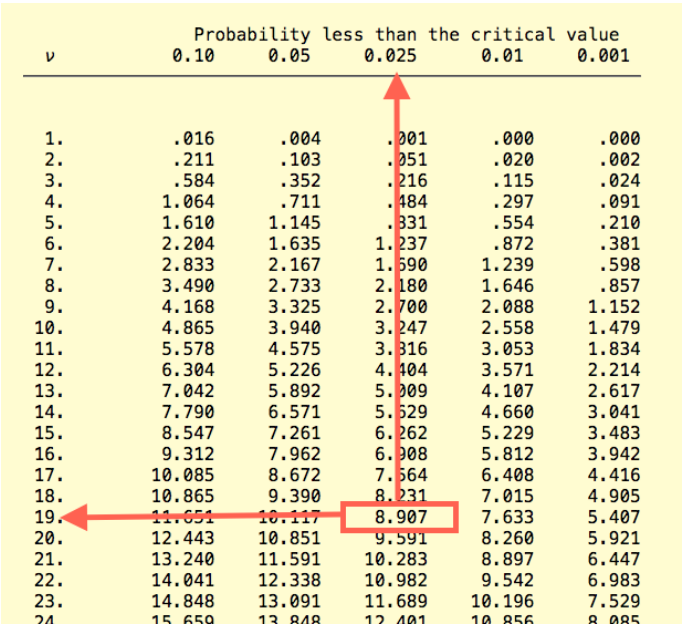
Using the above formula we get,

T = 12.569

Now, the critical values for significance level 0.05 for a two tailed chi-square test is given by x^2 (α/2,N-1) = 8.907 x^2 (1-α/2, N-1) = 32.852



ν	Probability less than the critical value 0.90	0.95	0.975	0.99	0.999
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588
11	17.275	19.675	21.920	24.725	31.264
12	18.549	21.026	23.337	26.217	32.910
13	19.812	22.362	24.736	27.688	34.528
14	21.064	23.685	26.119	29.141	36.123
15	22.307	24.996	27.488	30.578	37.697
16	23.542	26.296	28.845	32.000	39.252
17	24.769	27.587	30.191	33.409	40.790
18	25.989	28.869	31.526	34.805	42.312
19	27.204	30.144	32.852	36.191	43.820
20	28.412	31.410	34.170	37.566	45.315
21	29.615	32.671	35.479	38.932	46.797
22	30.813	33.924	36.781	40.289	48.268
23	32.007	35.172	38.076	41.638	49.728
24	33.196	36.415	39.364	42.980	51.179



ν	Probability less than the critical value 0.10	0.05	0.025	0.01	0.001
1.	.016	.004	.001	.000	.000
2.	.211	.103	.051	.020	.002
3.	.584	.352	.216	.115	.024
4.	1.064	.711	.484	.297	.091
5.	1.610	1.145	.831	.554	.210
6.	2.204	1.635	1.237	.872	.381
7.	2.833	2.167	1.690	1.239	.598
8.	3.490	2.733	2.180	1.646	.857
9.	4.168	3.325	2.700	2.088	1.152
10.	4.865	3.940	3.247	2.558	1.479
11.	5.578	4.575	3.816	3.053	1.834
12.	6.304	5.226	4.404	3.571	2.214
13.	7.042	5.892	5.009	4.107	2.617
14.	7.790	6.571	5.629	4.660	3.041
15.	8.547	7.261	6.262	5.229	3.483
16.	9.312	7.962	6.908	5.812	3.942
17.	10.085	8.672	7.564	6.408	4.416
18.	10.865	9.390	8.231	7.015	4.905
19.	11.651	10.117	8.907	7.633	5.407
20.	12.443	10.851	9.591	8.260	5.921
21.	13.240	11.591	10.283	8.897	6.447
22.	14.041	12.338	10.982	9.542	6.983
23.	14.848	13.091	11.689	10.196	7.529
24.	15.659	13.848	12.401	10.856	8.085

Hence critical region: Reject H_0 if $T < 8.907$ or $T > 32.852$

Since the T value we received is greater than 8.907, we failed to reject the null hypothesis.

4. Use the same sample you picked up in Step 2) of Part 5 to test $H_0 : \sigma = 15$ vs $H_a : \sigma < 15$ at the significance level 0.05. What's your conclusion?

   Solution:

   The entire procedure from above will be the same for this part, the only difference will be the critical value.

   Now, the critical value for significance level 0.05 for a lower tailed chi-square test is given by $x^2(\alpha/2, N-1) = 12.569$

   Hence critical region: Reject H_0 if $T < 10.117$

   Since the T value we received is less than 10.117, we failed to reject the null hypothesis.

   Important Note: All of tests performed until now were performed using random samples with a random-state taken as 100, but it was noted that for Part 6 questions 3 and 4, taking a random state of 0 bought some interesting changes to the results.

   The test statistic value for this sample was coming out to be 6.96 which is below 8.907 and 10.117, hence the null hypothesis was getting rejected in this case. Hence it should be noted that the null hypothesis is getting rejected in some of the samples.

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.

   Solution:

   For this question, we need to compare the means of two independent features, and we decided to do it using the two tailed independent t-test. The method ttestind is used in order to conduct this test and is taken from the scipy.stats module.

   The formula for t-statistic is given by:-

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

   The result of the operation was obtained as follows:- Statistics=0.485, p=0.6274695258

   Since the p-value is greater than 0.05, we accept the null hypothesis.

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.

   Solution:

For this question, we need to compare the means of two independent features(oil and gold) using the paired samples t-test. The method ttest-rel is used in order to conduct this test and is taken from the scipy.stats module. And we have also solved it manually using the formula.

$$t = \frac{\Sigma d}{\sqrt{\frac{n\left(\Sigma d^2\right)-\left(\Sigma d\right)^2}{n-1}}}$$

The result of the operation was obtained as follows:- statistic=0.54133, p-value=0.5884

Since the p-value is greater than 0.05, we accepted the null hypothesis.

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.

Solution:

Since we are required to form a hypothesis regarding the variance of two groups, we should be using the F-test.

The formula for f-statistic is given by:-

$$F = \frac{\sigma_1^2}{\sigma_2^2}$$

$$where\ \sigma^2 = \frac{\sum(x-\mu)^2}{n}$$

Following the formula, we get the f-value = 12.187

After this step, we need to calculate the degrees of freedom for both the

samples. They come out to be 998 for both the samples.

Then we find out the critical values: Fl and FU corresponding to the significance level and the degrees of freedom. They come out to be Fl=0.883 and FU = 1.132.

Since the calculated value is higher than the table value, we reject the null hypothesis.

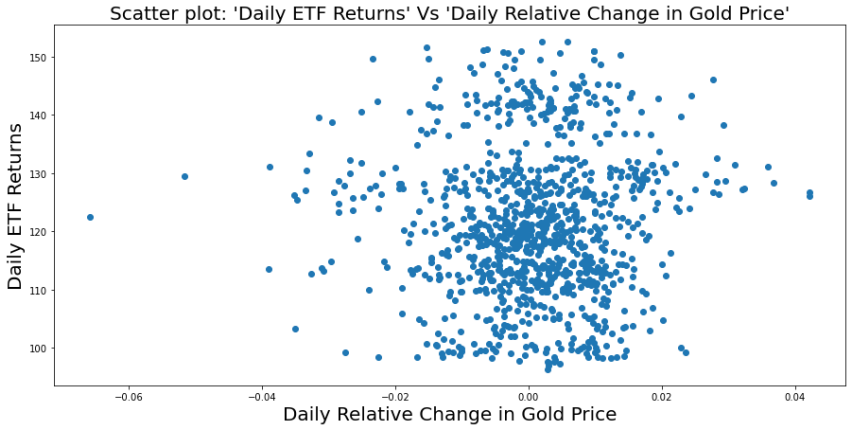
Part 8: Fitting the line to the data

(Consider the data including the ETF column and Gold column only.)

Using Python,

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?

   A Scatter/Dot plot of Daily ETF returns (Y) Vs. Daily relative change in Gold price (X) is as follows:



   Observations:

   - On the x axis the data for "Daily Relative Change of Gold price" does not start from 0. On the y axis the values for the variable "Daily ETF returns" does not start from 0 either. Starting points of the graph are on x axis:-0.065804741 on y axis: 96.419998

   - By looking at the dot/scatter plot it can be said that there is linear relationship between "Daily ETF returns" and "Daily Relative Change of Gold price".

   - But just by looking at the above plot it is hard to say magnitude and direction of co-relation between both of them. (That we can find out using Coefficient of Correlation(r) as follows.

2. Calculate the coefficient of correlation between ETF and Gold and interpret it.

   About a correlation coefficient:

   - Correlation coefficients are used to measure how strong a relationship is between two variables. Returns a value between -1 and 1.

   - Where 1 means that for every positive increase in one variable, there is a positive increase of a fixed proportion in the other.

   - -1 means that for every positive increase in one variable, there is a negative decrease of a fixed proportion in the other.

   - Zero means that for every increase, there isn't a positive or negative increase. The two just aren't related.

   - The absolute value of the correlation coefficient gives us the relationship strength. The larger the number, the stronger the relationship.

   - However the baseline may differ case to case to decide strong or weak relation. For example, $|-.75| = .75$, which has a stronger relationship than .65.

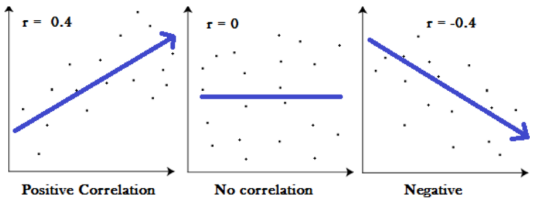


Figure 4: Demo of a correlation of -1, 0 and +1

   Correlation coefficient formula:

   - There are several types of correlation coefficient, but the most popular is Pearson's. Pearson's correlation (R) is a correlation coefficient commonly used in linear regression. Pearson's correlation coefficient(Pearson Product Moment Correlation (PPMC)) formula is as

follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Therefore, using PPMC the correlation of coefficient matrix for the ETF and Gold is:

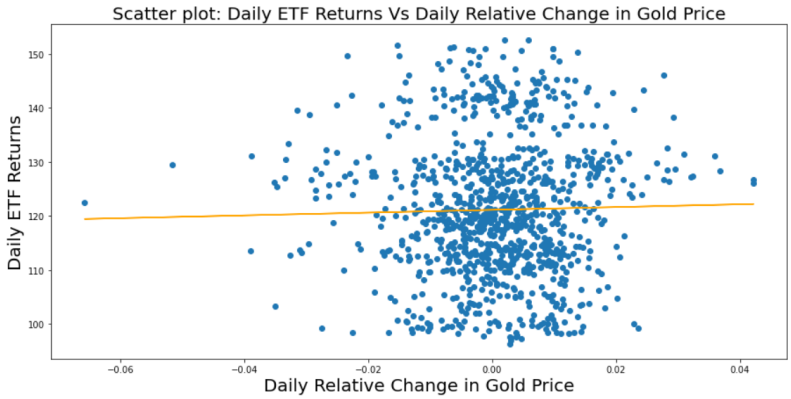
	DailyETFReturns	DailyRelativeChangeGoldPrice
DailyETFReturns	1.0	0.023
DailyRelativeChangeGoldPrice	0.023	1.0

Interpretation of the value of r that we got:

- The $|r|$ value for we got is 0.023.
- It is positive. Indicates there is a positive co-relation between two variables.
- If we take $r \geq 0.5$ as a strong linear relationship then in that case the relationship between between these two variables will be **weak positive linear relationship**
- That is for every positive increase in one variable, there is a positive increase of a fixed proportion in the other.

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?

The Regression Line to the Scatter plot is as follows:



Interpretation of Slope and intercept of this line:

- For the linear relationship between ETF(the daily ETF return) and Gold(the daily relative change in the gold price) we got the slope: 25.604 and intercept: 121.135.
- The slop value indicates: the rate of change in "daily ETF return" per unit change of "daily relative change in Gold price" is by 25.6043
- On the other hand intercept value shows that the value of the daily ETF return is 121.136 if the daily relative change in the gold price is 0. (if it is in the form y = mx + c where x = 'daily relative change in the gold price' and y is 'daily ETF return')

4. Conduct a two-tailed t-test with $H_0 : \beta_1 = 0$. 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?

   About two-tailed t-test

   Hypothesis testing is an essential procedure in statistics. A hypothesis test evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data.

   Two sampled T-test: The Independent Samples t Test or 2-sample t-test compares the means of two independent groups in order to determine whether there is statistical evidence that the associated population means are significantly different. The Independent Samples t Test is a parametric test. This test is also known as: Independent t Test.

   The model behind linear regression:

   When we are examining the relationship between a quantitative outcome and a single quantitative explanatory variable, simple linear regression is the most commonly considered analysis method.

   We postulate a linear relationship between the population mean of the

outcome and the value of the explanatory variable. If we let Y be some outcome, and x be some explanatory variable, then we can express the structural model using the equation

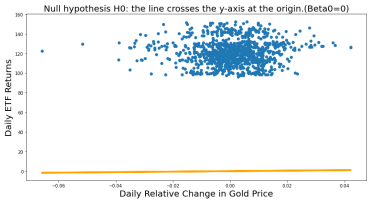
$$E(Y|x) = \beta_0 + \beta_1 x \quad or \quad \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

we are ultimately always interested in drawing conclusions about the population not the particular sample we observed. In the simple regression setting, we are often interested in learning about the population intercept β_0 and the population slope β_1 . As we know, confidence intervals and hypothesis tests are two related, but different, ways of learning about the values of population parameters.

Steps to conduct a two-tailed t-test with $H_0 : \beta_1 = 0$:

An $\alpha - level$ hypothesis test for slope parameter β_1. We follow standard hypothesis test procedures in conducting a hypothesis test for the slope β_1.

(a) First, Specify the null and alternative hypotheses: Null hypothesis $H0 : \beta_1 = 0$, Alternative hypothesis $H_A : \beta_1 \neq 0$. If we put $\beta_1 = 0$ we get following plot



(b) Calculate the value of the test statistic using the following formula:

$$t = \frac{\hat{\beta}_1 - \beta_1}{SE(\hat{\beta}_1)}$$

(c) Third, use the resulting test statistic (t score) to calculate the P-value. As always, the P-value is the answer to the question "how likely is it that we'd get a test statistic t* as extreme as we did if the null hypothesis were true?" The P-value is determined by referring to a t-distribution with n-2 degrees of freedom.

(d) Finally, we make a decision:

- If the P-value is smaller than the significance level α, we reject the null hypothesis in favor of the alternative. We conclude "there is sufficient evidence at the α level to conclude that there is a relationship in the population between the predictor x and response y."
- If the P-value is larger than the significance level α, we fail to reject the null hypothesis. We conclude "there is not enough evidence at the α level to conclude that there is a relationship in the population between the predictor x and response y."

Now using above mentioned steps let's verify the Null hypothesis. For the Linear model with input variable is X = "Daily Relative Change in Gold Price" along with constant and Y = "Daily ETF Returns" we got the following model report in Python:

OLS Regression Results

Dep. Variable:	DailyETFReturns	**R-squared:**	0.001
Model:	OLS	**Adj. R-squared:**	-0.000
Method:	Least Squares	**F-statistic:**	0.5280
Date:	Tue, 04 May 2021	**Prob (F-statistic):**	0.468
Time:	10:14:06	**Log-Likelihood:**	-3949.5
No. Observations:	1000	**AIC:**	7903.
Df Residuals:	998	**BIC:**	7913.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	121.1360	0.398	304.155	0.000	120.354	121.918
DailyRelativeChangeGoldPrice	25.6044	35.236	0.727	0.468	-43.541	94.750

Omnibus:	26.752	**Durbin-Watson:**	0.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	23.045
Skew:	0.305	**Prob(JB):**	9.91e-06
Kurtosis:	2.576	**Cond. No.**	88.6

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Figure 5: Model report

In which the Coefficients Table is as follows:

	coef	std err	t	P>|t|
const	121.1360	0.398	304.155	0.000
DailyRelativeChangeGoldPrice	25.6044	35.236	0.727	0.468

We can see the coefficient of the intercept, or the constant as they've named it in our case. Hence we have

$$\hat{\beta}_0 = 121.1585,\ \hat{\beta}_1 = 25.6044$$

Where β_0 is y-intercept and β_1 is the slope in the regression line.

By default, the test statistic is calculated assuming the user wants to test that the slope is 0. Dividing the estimated coefficient for Daily Relative Change in Gold Price = 25.6044 by the estimated standard error for the Daily Relative Change in Gold Price = 35.236, that model reports printed the test statistic T is

$$t = \frac{25.6044 - 0}{35.236} = 0.727$$

Which is matching with the given t value for this variable in the report too.

Calculate the critical region: Consider the $\alpha = 0.01$ and from the report table we got the

$$Degree\ of\ freedom = 898$$

So using the following table let's derive the critical value:

TABLE 2 Critical Values for Two-Sided and One-Sided Tests Using the Student *t* Distribution

	Significance Level				
Degrees of Freedom	20% (2-Sided) 10% (1-Sided)	10% (2-Sided) 5% (1-Sided)	5% (2-Sided) 2.5% (1-Sided)	2% (2-Sided) 1% (1-Sided)	1% (2-Sided) 0.5% (1-Sided)
1	3.08	6.31	12.71	31.82	63.66
2	1.89	2.92	4.30	6.96	9.92
3	1.64	2.35	3.18	4.54	5.84
4	1.53	2.13	2.78	3.75	4.60
5	1.48	2.02	2.57	3.36	4.03
6	1.44	1.94	2.45	3.14	3.71
7	1.41	1.89	2.36	3.00	3.50
8	1.40	1.86	2.31	2.90	3.36
9	1.38	1.83	2.26	2.82	3.25
10	1.37	1.81	2.23	2.76	3.17
11	1.36	1.80	2.20	2.72	3.11
12	1.36	1.78	2.18	2.68	3.05
13	1.35	1.77	2.16	2.65	3.01
14	1.35	1.76	2.14	2.62	2.98
15	1.34	1.75	2.13	2.60	2.95
16	1.34	1.75	2.12	2.58	2.92
17	1.33	1.74	2.11	2.57	2.90
18	1.33	1.73	2.10	2.55	2.88
19	1.33	1.73	2.09	2.54	2.86
20	1.33	1.72	2.09	2.53	2.85
21	1.32	1.72	2.08	2.52	2.83
22	1.32	1.72	2.07	2.51	2.82
23	1.32	1.71	2.07	2.50	2.81
24	1.32	1.71	2.06	2.49	2.80
25	1.32	1.71	2.06	2.49	2.79
26	1.32	1.71	2.06	2.48	2.78
27	1.31	1.70	2.05	2.47	2.77
28	1.31	1.70	2.05	2.47	2.76
29	1.31	1.70	2.05	2.46	2.76
30	1.31	1.70	2.04	2.46	2.75
60	1.30	1.67	2.00	2.39	2.66
90	1.29	1.66	1.99	2.37	2.63
120	1.29	1.66	1.98	2.36	2.62
∞	1.28	1.64	1.96	2.33	2.58

From the standard t distribution table - With significance Level = 1% (

two sided) we have: Critical value as: 2.58.

Drawing conclusions about slope parameter $\beta_1 = 0$ using T-statistics and critical region values:

- Since Critical region: $(-\infty, -2.581] \cup [2.581, \infty)$ and t-score as 0.727 therefore the t-score does not belongs to the critical region, so we accept H_0 and reject H_a.
- That is the we accept: H0: 'Daily Relative Change in Gold Price' does not affect 'Daily ETF Returns' and reject that H1: that is 'Daily Relative Change in Gold Price' does affect 'Daily ETF Returns'.

p-value significance level α:

By default, the P-value is calculated assuming the alternative hypothesis is a "two-tailed, not-equal-to" hypothesis. From the report we have .

$$degrees\ of\ freedom(n-2) = 898$$

$$P = 0.468\ (\ to\ three\ decimal\ places)$$

Upon calculating the probability that a t-random variable with n-2 degrees of freedom would be larger than —t=0.727—, and multiplying the probability by 2, we have

$$2 \times p = 2 \times 0.468 = 0.936 > \alpha = 0.01$$

Drawing conclusions about slope parameter $\beta_1 = 0$ using T-statistics and critical region values:

- Because the P-value is so big (greater than α), we can accept the null hypothesis H_0 and conclude that $\beta_1 = 0$.

- Therefore, there is sufficient evidence, at the $\alpha = 0.01$ level, to conclude that there is a no relationship in the population between "Daily Relative Change in Gold Price" and "Daily ETF Returns".

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 (R^2 *or* r^2):

   - The COD is the proportion of the variance in the dependent variable that is predictable from the independent variable(s)

     $$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

   - It gives you an idea of how many data points fall within the results of the line formed by the regression equation. The higher the coefficient, the higher percentage of points the line passes through when the data points and line are plotted. For example if the coefficient is 0.80, then 80% of the points should fall within the regression line.
   - Values of 1 or 0 would indicate the regression line represents all or none of the data, respectively. A higher coefficient is an indicator of a better goodness of fit for the observations. The best possible score is 1 which is obtained when the predicted values are the same as the actual values. R^2 score of baseline model is 0.
   - The COD can be negative, although this usually means that your model is a poor fit for your data. It can also become negative if you didn't set an intercept. During the worse cases, R^2 score are negative.

     $$SS_{tot} = \sum (y_i - \bar{y})^2$$
     $$SS_{res} = \sum (y_i - \bar{f}_i)^2 = \sum e_i^2$$

   Although we can consider a value for the comparison let's say 0.5 and can say if the R^2 is greater that this then it is best or good model. But generally it is considered that if R^2 value is near to 1 then it is good model. For the linear model we got

   $$R^2 = 0.0005287962431211879$$

   Therefore we can say that since the $R^2 = 0.0005$ value is very small compared to 0.5 hence model is bad.

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

Four assumptions are being maid:

(a) Assumption of Linearity check:

With assumption linearity or to seen the outcome variable is linearly related to any of the predictors (by straight line) since the predictors combined effect (in order to sum up their linear relation by straight line) will help to make prediction of dependent variable.

Diagnosing linearity using visual inspection method can be difficult since there is no basis (or well defined standards) to determine what is considered linear and and what is not.

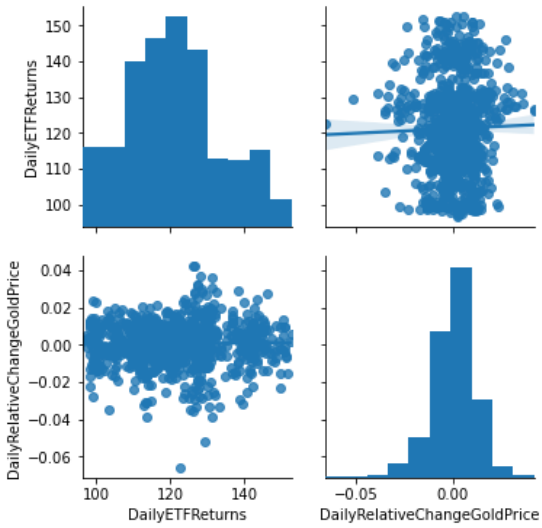
- Method 1: Using co-relation matrix

	DailyETFReturns	DailyRelativeChangeGoldPrice
DailyETFReturns	1.0	0.023
DailyRelativeChangeGoldPrice	0.023	1.0

Figure 6: correlation of coefficient matrix for ETF and Gold

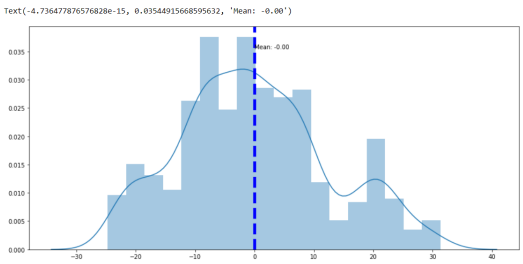
Dependent variables and the independent variable are having weak liner co-relation. But the the co-relation is positive ¿ 0. So we can say that the independent variable shows positive weak co-relation with dependent one.

- Method 2: Using Matrix scatter plots and prediction line
  The first assumption we check is linearity. We can visually check this by fitting ordinary least squares (OLS) on a data.



Observations/Linearity Diagnosis: Check scatter plot between Dependent variable and the other independent variables.
With assumption of linearity we are looking to see if the outcome variable is linearly related to each of the independent variable

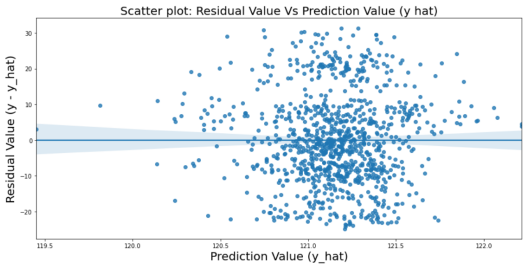
We draw a histogram of the residuals, and then examine the normality of the residuals. If the residuals are not skewed, that means that the assumption is satisfied. It can be checked the predictor not so linearly related to dependent variable that is ETF. It is also cleared from the co-relation table too.



Even though is slightly skewed, but it is not hugely deviated from being a normal distribution. We can say that this distribution satisfies the normality assumption.

(b) Independent Errors assumption

The assumption of IE is that successive residuals should be indepen-

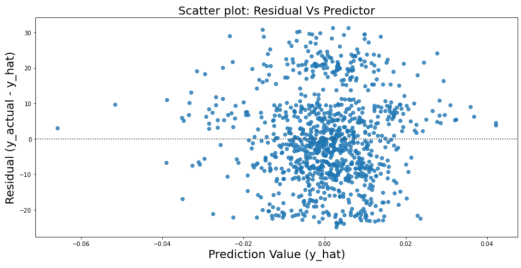


dent that means there is no patterns to the residuals. The residuals aren't highly co-related there is no long runs of positive or negative residuals

When the successive residuals are correlated we refer to this condition as 'auto-corelation' and that frequently occurs when the data are collected over a period of time. In our case There is no specific pattern that is it does not include or indicate auto-corelation. Hence Independent Errors assumption is not violated from the graph

(c) Assumption of Homoscedasticity:

From the previous explanation as we know that the dots are scattered



the way that they are it indicates the data meet the assumption of the errors being normally distributed also because the dots are scattered it indicates that the variances of the residuals are constant

(d) Multi Co-linearity:

This occurs when two or more variables very closely linearly related as co-linearity increases so do the standard errors of the be codified coefficient the beta coefficient so if you think back to what the standard error represents the big standard errors for the beta coefficients mean that the beta coefficients vary more across the samples in essence our betas become less trustworthy it also makes it difficult to assess the individual importance of a predictor if the predictors are highly co-related and each accounts for similar variants in the outcome then how do we know how can we tell which of the two variables is important we really cant the model could include either interchangeably one way we can check for molecularity is to scan a correlation matrix for high coefficients those that are above say 0.8 or 0.9 in another words what we are looking at the correlation matrix to check for multi col-linearity we want to have low correlation coefficients between each of the independent variables and each of the other independent variable you do not really need to include the dependent variable in the correlation matrix to check this assumption though.

	VIF	variable	Tolerance
0	1.003451	const	0.996561
1	1.000000	gold	1.000000

Hence it can be seen that for the gold the assumption is not violated if compared with constant.

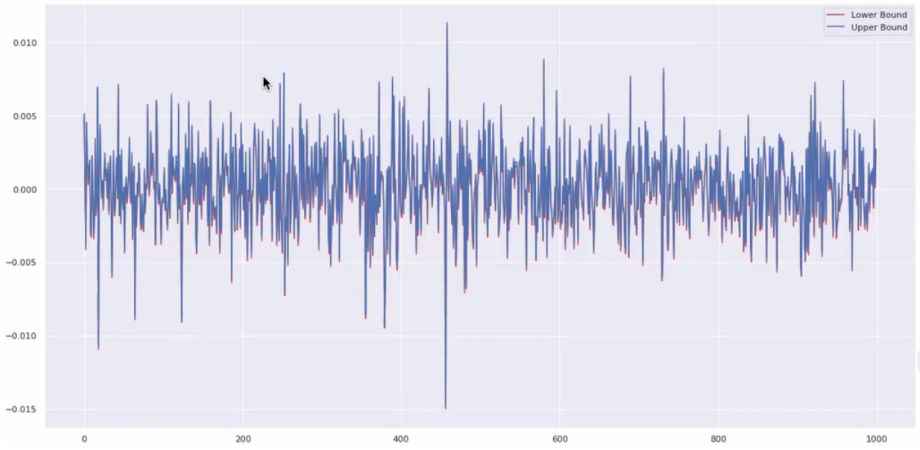
7. Given the daily relative change in the gold price is 0.005127. Calculate the 99% confidence interval of the mean daily ETF return, and the 99% prediction interval of the individual daily ETF return.

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

Where CI: confidence interval, $\bar{x}$ is sample mean, z is confidence level value, s is sample standard deviation, n sample size.

Confidence interval for ETF is $(118.57220037473246, 123.73371964926773)$
For alpha: 0.99

Also the 99% prediction interval of the individual daily ETF return



Interpretation:
There is 99% chance that the confidence interval of (array([118.57220037, -2.5800968]), array([123.73371965, 2.58142247])) contains the true mean.

Part 9: Does your model predict?

Consider the data including the ETF, Gold and Oil column. Using any software, fit a multiple linear regression model to the data with the ETF variable as the response. Evaluate your model with adjusted R square.

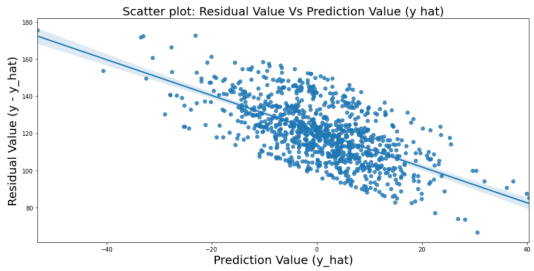
Multiple regression is like linear regression, but with more than one independent value, meaning that we try to predict a value based on two or more variables. Before fitting a model for the data let's check the how the variables are co-related:

	Close_ETF	oil	gold
Close_ETF	1.0	-0.012	0.018
oil	-0.012	1.0	0.17
gold	0.018	0.17	1.0

As it can be seen that the variables are weakly co-related to each other as follows:

- The magnitude shows that it is < 0.5 hence are weakly correlated
- The Oil is negatively co-related with the ETF and on the other hand gold is positively.

If we draw a graph for the Predict Vs Residual To Check Linearity we get following plot



From the graph it is hard to tell anything about linearity assumption since there are more than one independent variables hence using 2d the Predict Vs Residual will not help.

After model is fit we got the summary of the model as follows:

OLS Regression Results

Dep. Variable:	DailyETFReturns	**R-squared:**	0.001
Model:	OLS	**Adj. R-squared:**	-0.001
Method:	Least Squares	**F-statistic:**	0.3743
Date:	Tue, 04 May 2021	**Prob (F-statistic):**	0.688
Time:	21:49:57	**Log-Likelihood:**	-3949.4
No. Observations:	1000	**AIC:**	7905.
Df Residuals:	997	**BIC:**	7919.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	121.1427	0.399	303.856	0.000	120.360	121.925
DailyRelativeChangeOilPrice	-9.1261	19.413	-0.470	0.638	-47.221	28.968
DailyRelativeChangeGoldPrice	29.6226	36.272	0.817	0.414	-41.555	100.800

Omnibus:	26.565	**Durbin-Watson:**	0.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	22.981
Skew:	0.306	**Prob(JB):**	1.02e-05
Kurtosis:	2.579	**Cond. No.**	92.2

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The formula for the R^2 is as follows:

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

From the report the Value of the R^2 is as follows:

$$R^2 = -0.001$$

Evaluate your model with adjusted R^2

If we recall in Part 8 the we got the R^2 as 0.000 and after adding oil as the new feature for model training, we get the new value of adjusted R^2 as -0.001. Now, in order to interpret this change, let's take a look at the formula for adjusted r-square which is given by:-

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where, the p term in the denominator acts as a penalty term which penalizes the model performance score(adjusted r-square) if the newly introduced feature is not helping in improving the model performance because it is located in the denominator of the equation. If however, the model performance increases due to the newly introduced variable, the value of R^2(located in the denominator) will compensate for the penalization and would improve the overall score.

It shows that the newly added feature(oil) is not adding any value to the model performance.

Part 10: Checking residuals and model selection

Calculate the residuals of the model fitting you did in Part 9.

Check the four assumptions made for the error terms of the multiple regression model using these residuals (mean 0; constant variance; normality; and the independence). Plot the residuals to check these assumptions. For example, draw a Normal Probability Plot to check the normality assumption; draw a scatter plot of Residuals vs. Fitted Values to check the constant variance assumption and the independence assumption; and so on. You may refer to the following link https://www.youtube.com/watch?v=4zQkJw73U6I for some hints. In your project report, all the relevant plots and at least one paragraph of summary of checking the four assumptions using those plots must be included.

Multiple regressions are based on the assumption that there is a linear relationship between both the dependent and independent variables. It also assumes no major correlation between the independent variables.

There are 4 assumptions associated with a linear regression model:

1. Linearity: The relationship between X and the mean of Y is linear.
2. Homoscedasticity: The variance of residual is the same for any value of X.
3. Independence: Observations are independent of each other.
4. Normality: For any fixed value of X, Y is normally distributed.

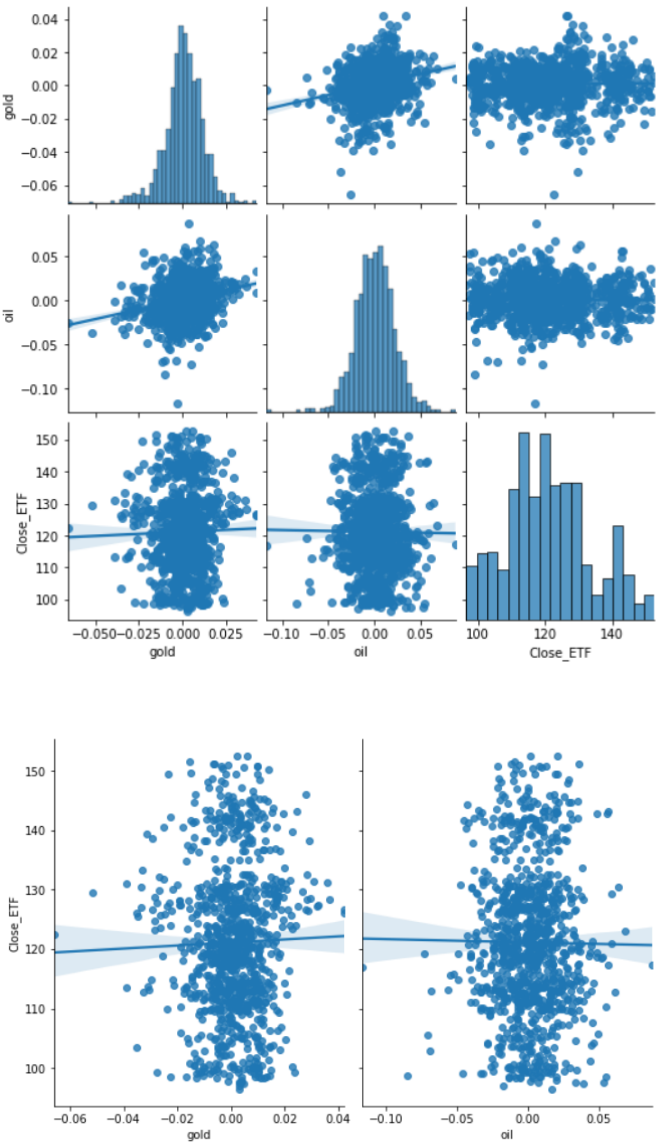
We will use the diagnostics plots and will focus on the residuals - or errors - of a model, which is mathematical jargon for the difference between the actual value and the predicted value. These plots examine a few different assumptions about the model and the data:

1. The data can be fit by a line (this includes any transformations made to the predictors, e.g., $x^2, \sqrt{x}$)
2. Errors are normally distributed with mean zero
3. Errors have constant variance, i.e., homoscedasticity
4. There are no high leverage points.

let's check the 4 assumptions one by one as follows:

1. Assumption of Linearity check:

- First, linear regression needs the relationship between the independent and dependent variables to be linear. It is also important to check for outliers since linear regression is sensitive to outlier effects. The linearity assumption can best be tested with scatter plots.



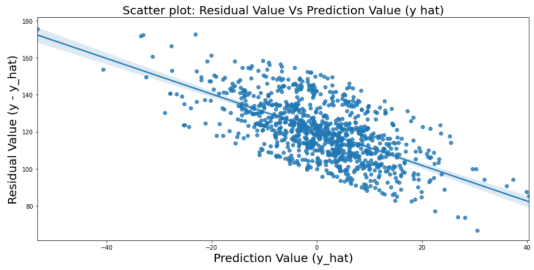
Since it is tough to see the magnitude of the relationship but there is linear

relationship between dependent and independent variables. This we can check using the the following co-relation matrix as well:

	Close_ETF	gold	oil
Close_ETF	1.000000	0.022996	-0.009045
gold	0.022996	1.000000	0.235650
oil	-0.009045	0.235650	1.000000

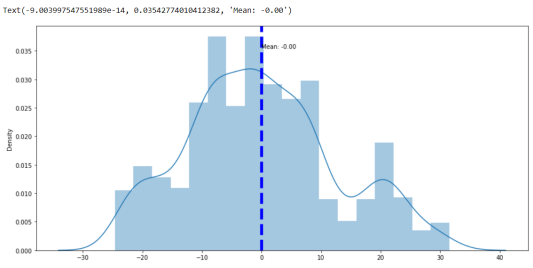
From the above matrix it is clear that the ETF and gold having positive weak linear corelationa and ETF and oil is having negative weak linear co-relation. But if compare the magnitude it seems ETF and gold is having strong co-relation and it is positive one.

Note that in the part 9 we have seen that if we draw a graph for the Predict Vs Residual to Check Linearity:



We can see from the graph it is hard to tell anything about linearity assumption since there are more than one independent variables hence using 2-dimensional plot of Predict Vs Residual will not help in this regard.

If we draw the histogram of residuals:



Even though is slightly skewed, but it is not hugely deviated from-being a normal distribution. We can say that this distribution satisfies the normality assumption.

2. Independent Errors assumption:

Independence of errors says that there is no relationship between the residuals and the variable; in other words, is independent of errors. We can check this assumption by examining a scatter-plot of "residuals versus fits"; the correlation should be approximately 0.

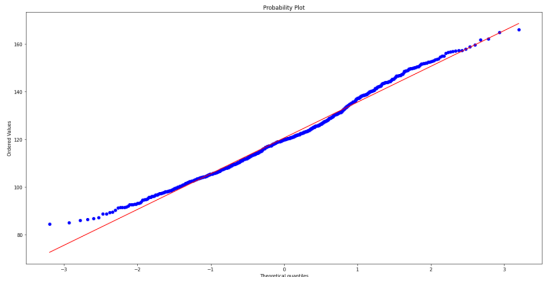


Figure 7: Normal Q-Q Plot

This plot shows if the residuals are normally distributed. From the above QQ plot it can be see that the plot is having fat tails, and the data is not Gaussian at the tails. This is indicative of the errors not being normally distributed, in fact our model suffers from "heavy tails". So it seems it is violating the assumption of independent errors.

3. Assumption of homoscedasticity:

Scale Location plot is a way to check if the residuals suffer from non-constant variance, aka **heteroscedasticity**.

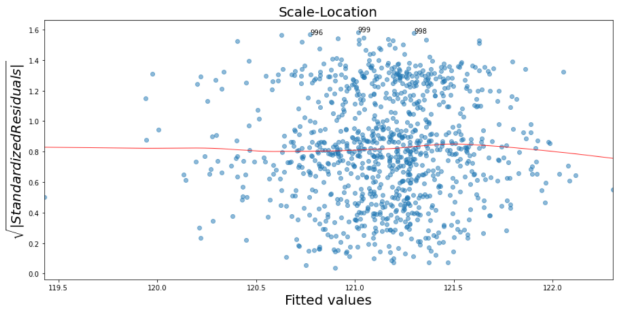


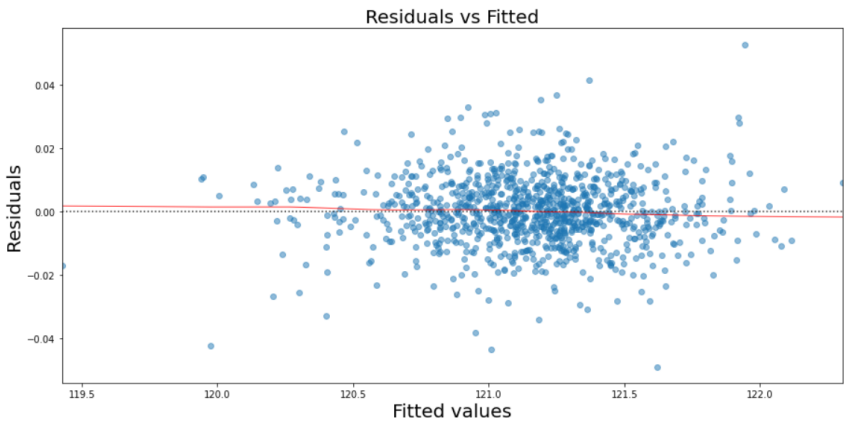
Figure 8: Scale-Location

This particular plot (with the housing data) is a tricky one to debug. The more horizontal the red line is, the more likely the data is **homoscedastic**.

While a typical heteroscedastic plot has a sideways "V" shape, our graph has higher values on the left and in the middle versus in the right. This might be caused by not capturing the non-linearities in the model (see Residuals vs Fitted plot at the end of this section) and merits further investigation or model tweaking. The two most common methods of "fixing" heteroscedasticity is using a weighted least squares approach, or using a heteroscedastic-corrected covariance matrix (hccm). Both of these methods are beyond the scope of this project report.

Therefore from the plot we can see that the red-line in the above plot is more or less horizontal to the x axis with little bents in the middle and on the right if we ignore these bents then the assumption of homoscedastic is valid and if we consider those bents and take into account that the red-line in the plot is not horizontal enough then we can say that the assumption of homoscedastic is violated.

Now let's check the Residuals vs Fitted(predicted) plot: From the graph the



points are scattered across the horizontal line crossing 0. Since the points are more are less scattered equally across the line it indicates the assumption of homoscedastic is not violated.

Therefore, in case of Scale-Location plot if we consider the red line is close to be horizontal line and if consider Residuals vs Fitted(predicted) plot then can conclude that the homoscedastic assumption is not violated.

4. Assumption of Multi-Collinearity:

If we consider VIF values for for the multi-linear model we got the following values:

	oil	gold
vif	1.059952	1.059952

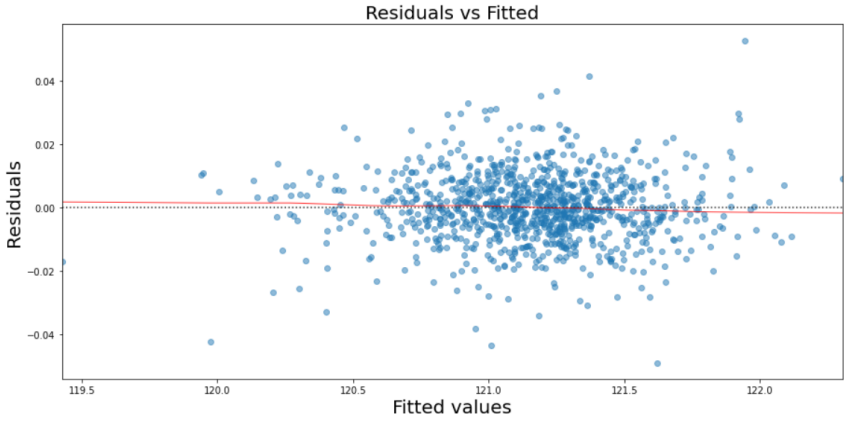
Therefore, its following multi-collinearity assumption since the values are less than 5.

5. Additional Analysis:

a) Residuals vs Fitted:

In addition to above analysis the plot of Residuals vs Fitted graph shows if there are any nonlinear patterns in the residuals, and thus in the data as well. One of the mathematical assumptions in building an OLS model is that the data can be fit by a line. If this assumption holds and our data can be fit by a linear model, then we should see a relatively flat line when looking at the residuals vs fitted. An example of this failing would be trying to fit the function $f(x) = x^2$ with a linear regression $y = \beta_0 + \beta_1 x$. Clearly, the relationship is nonlinear and thus the residuals have non-random patterns.

In our case the plot will look like this:

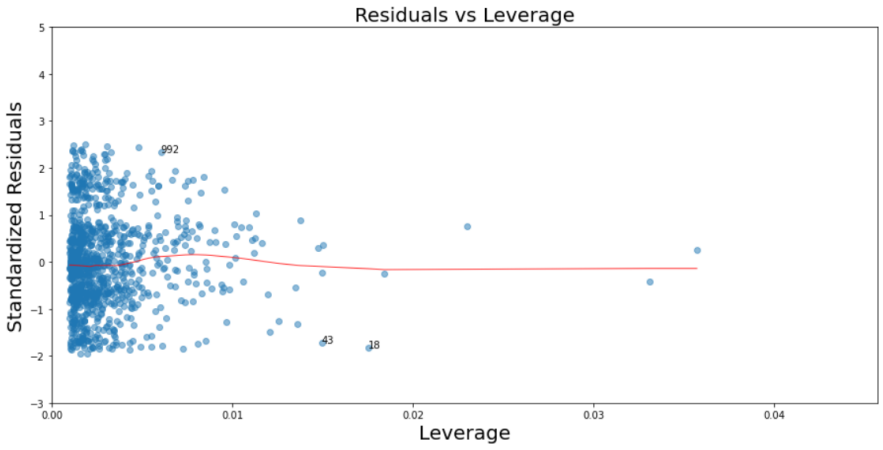


An ideal Residuals vs Fitted plot will look like random noise; there won't be any apparent patterns in the scatter-plot and the red line would be horizontal. Notice the line in red? if it is bow shaped then we are under-fitting the model. In out case it is not bow shaped but it is not horizontal either this is an indicator that we are failing to capture some of the non-linear features of the model. In other words, Perhaps the variance in the data might be better captured using the square (or some other non-linear transformation) of one or more of the features. Which feature(s) specifically is beyond the scope of this report.

b) Residuals vs Leverage:

Unlike outliers, which have an unusually large y value, leverage points have extreme x values. This may not seem so bad at face value, but it can have damaging effects on the model because the β coefficients are very sensitive to leverage points. The purpose of the Residuals vs Leverage plot is to identify these problematic observations.

Consider the Cook's Distance, we only need to find leverage points that have



a distance greater than 0.5. In this plot, we do not have any leverage points that meet this criteria. Therefore not required to remove any point since all the points are falling below 0.5. Note in case if we may have found any point above 0.5 then removal of that point and re-execution of above steps again may resulted in the improvement of model.

3 Section 3 - Discussions or Improvements

1. Data prepossessing such as scaling, data transformation can tried out.

2. We can try other models too along with simple and multi linear regression models.

3. We can try including JPM column along with other columns for the multi regression model.

4 References

Part 1, 2, 3

1. https://machinelearningmastery.com/time-series-data-visualization-with-python/
2. https://towardsdatascience.com/normality-tests-in-python-31e04aa4f411

Part 4

https://medium.com/analytics-vidhya/central-limit-theorem-and-machine-learning-part-1-af3b65dc9d32

Part 5

1. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ wilcox.test
2. https://www.statology.org/confidence-intervals-python/
3. https://www.youtube.com/watch?v=MUD390jtgQs

Part 6

1. https://mathcracker.com/z-test-for-one-mean
2. https://github.com/sharmasw/Data-Science-with-python/blob/master/ Statistics20Notebooks/Hypothesis20Testing.ipynb
3. https://www.youtube.com/watch?v=kd6zKBa9Rfk
4. https://www.statisticshowto.com/probability-and-statistics/ find-critical-values/
5. https://www.statisticshowto.com/probability-and-statistics/ t-distribution/t-score-formula/
6. https://sixsigmastudyguide.com/1-sample-wilcoxon-non- parametric-hypothesis-test/
7. http://www.sthda.com/english/wiki/one-sample-wilcoxon-signed- rank-test-in-r
8. https://www.itl.nist.gov/div898/handbook/eda/section3/eda358.htm
9. https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm

Part 7

1. https://www.socscistatistics.com/pvalues/tdistribution.aspx

2. https://www.youtube.com/watch?v=0Pd3dc1GcHc
3. https://www.youtube.com/watch?v=8aaIdXENNJI
4. https://github.com/bhattbhavesh91/GA$_S$$essions/blob/master/t_test_independence$
$/T_Test_Sales.ipynb$
5.$https://www.youtube.com/watch?v = DZyDbEzaiK0$
6.$https://www.youtube.com/watch?v = mizwZNj2aZA$
7.$https://www.statisticshowto.com/probability-and-statistics/hypothesis-$
$testing/f-test/$
8.$https://mathcracker.com/f-critical-valuesresults$

Part 8, Part 9, Part 10

1.$SourceforAboutaCorrelationcoefficient : https://www.statisticshowto.com/probability-$
$and-statistics/correlation-coefficient-formula/$
2.$Hypothesistesting : https://online.stat.psu.edu/stat501/node/644$
3.$https://365datascience.com/tutorials/statistics-tutorials/null-hypothesis/$
4.$https://jeffmacaluso.github.io/post$
$/LinearRegressionAssumptions/$
5.$https://www.youtube.com/watch?v = 4zQkJw73U6I$
6.$https://www.statisticshowto.com/probability-and-statistics/statistics-$
$definitions/adjusted-r2/::$
$text = Adjusted\%20R2\%20is\%20a,of\%20terms\%20in\%20a\%20model.$

5 Appendix

Python Code for Part 1:

```
# ---------- Python code starts ----------
import pandas as pd
import numpy as np

print("----------Mean of columns----------")
print("Close_ETF Mean : " + str(np.mean(df['Close_ETF'])))
print("Oil Mean : " + str(np.mean(df['oil'])))
print("Gold Mean : " + str(np.mean(df['gold'])))
print("JPM Mean : " + str(np.mean(df['JPM'])))

print("----------Standard Deviation of columns----------")
print("Close_ETF standard deviation : " + str(np.std(df['Close_ETF'])))
print("Oil standard deviation : " + str(np.std(df['oil'])))
print("Gold standard deviation : " + str(np.std(df['gold'])))
print("JPM standard deviation : " + str(np.std(df['JPM'])))

#Output co-relation between columns
df.corr()

#Describe detail about columns
df.describe()

# ---------- Python code ends ----------
```

Paython code for part 2:

1) A histogram for each column

```
# ———————————— Python code starts ————————————
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

figure(figsize=(8, 6), dpi=80)
plt.subplot(2,2,1)
plt.hist(df['Close_ETF'])
plt.subplot(2,2,2)
plt.hist(df['oil'])
plt.subplot(2,2,3)
plt.hist(df['gold'])
plt.subplot(2,2,4)
plt.hist(df['JPM'])

# ———————————— Python code ends ————————————
```

2) A time series plot for each column

```
# ———————————— Python code starts ————————————
import seaborn as sns

#Time series plot for ETF
sns.lineplot(data=df['Close_ETF'])

#Time series plot for oil
sns.lineplot(data=df['oil'])

#Time series plot for gold
sns.lineplot(data=df['gold'])

#Time series plot for JPM
sns.lineplot(data=df['JPM'])

# ———————————— Python code ends ————————————
```

3) A time series plot for all four columns

```
# ———————————— Python code starts ————————————
import seaborn as sns

#Time series plot for All columns
sns.lineplot(data=df)

# ———————————— Python code ends ————————————
```

4) 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

```
# ———————————— Python code starts ————————————
import seaborn as sns

#Scatter plots plot between the ETF column and the OIL column
sns.scatterplot(data=df, x="Close_ETF", y="oil")

plt.plot(df["Close_ETF"], df["oil"], '.')
m, b = np.polyfit(df["Close_ETF"], df["oil"], 1)
plt.plot(df["Close_ETF"], m*df["Close_ETF"] + b)
plt.ylabel('Oil')
plt.xlabel('Close_ETF')

#Scatter plots plot between the ETF column and the GOLD column
sns.scatterplot(data=df, x="Close_ETF", y="gold")

plt.plot(df["Close_ETF"], df["gold"], '.')
m, b = np.polyfit(df["Close_ETF"], df["gold"], 1)
plt.plot(df["Close_ETF"], m*df["Close_ETF"] + b)
plt.ylabel('Gold')
plt.xlabel('Close_ETF')

#Scatter plots plot between the ETF column and the JPM column
sns.scatterplot(data=df, x="Close_ETF", y="JPM")

plt.plot(df["Close_ETF"], df["JPM"], '.')
m, b = np.polyfit(df["Close_ETF"], df["JPM"], 1)
plt.plot(df["Close_ETF"], m*df["Close_ETF"] + b)
```

```
plt.ylabel('JPM')
plt.xlabel('Close_ETF')

# ———————————— Python code ends ————————————
```

Paython code for part 3:

```
# ---------- Python code starts ----------
import numpy as np
import pylab
import scipy.stats as stats
import seaborn as sns
from scipy.stats import chisquare

print('---------- Normality test for ETF ----------')
print("---------- ETF Histogram ----------")
figure(figsize=(6, 6), dpi=80)
sns.distplot(df['Close_ETF'], hist=True, kde=True)

print("---------- QQ Plot ----------")
figure(figsize=(6, 6), dpi=100)
stats.probplot(df["Close_ETF"], dist="norm", plot=pylab)

print('---------- Chisquare ----------')
stat, p1 = chisquare(df["Close_ETF"])
print('Statistics=%.3f, p=%.3f' % (stat, p1))
alpha = 0.05
if p1 > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Shapiro-Wilk ----------')

stat, p2 = shapiro(df["Close_ETF"])
print('Statistics=%.3f, p=%.3f' % (stat, p2))

alpha = 0.05
if p2 > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')
```

```
print('---------- Normality test for Oil ----------')
print("---------- Oil Histogram ----------")
figure(figsize=(6, 6), dpi=80)
sns.distplot(df['oil'], hist=True, kde=True)

print("---------- QQ Plot ----------")
figure(figsize=(6, 6), dpi=100)
stats.probplot(df["oil"], dist="norm", plot=pylab)

print('---------- Chisquare ----------')
stat, p1 = chisquare(df["oil"])
print('Statistics=%.3f, p=%.3f' % (stat, p1))
alpha = 0.05
if p1 > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Shapiro-Wilk ----------')

stat, p2 = shapiro(df["oil"])
print('Statistics=%.3f, p=%.3f' % (stat, p2))

alpha = 0.05
if p2 > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Normality test for Gold ----------')
print("---------- Gold Histogram ----------")
figure(figsize=(6, 6), dpi=80)
sns.distplot(df['gold'], hist=True, kde=True)

print("---------- QQ Plot ----------")
figure(figsize=(6, 6), dpi=100)
stats.probplot(df["gold"], dist="norm", plot=pylab)
```

```
print('---------- Chisquare --------------------')
stat, p = chisquare(df["gold"])
print('Statistics=%.3f, p=%.3f' % (stat, p1))
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Shapiro-Wilk ----------')
stat, p = shapiro(df["gold"])
print('Statistics=%.3f, p=%.3f' % (stat, p2))
# interpret
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Normality test for JPM ----------')
print("---------- JPM Histogram ----------")
figure(figsize=(6, 6), dpi=80)
sns.distplot(df['JPM'], hist=True, kde=True)

figure(figsize=(6, 6), dpi=100)
stats.probplot(df["JPM"], dist="norm", plot=pylab)

print('---------- Chisquare ----------')
stat, p = chisquare(df["JPM"])
print('Statistics=%.3f, p=%.3f' % (stat, p))
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

print('---------- Shapiro-Wilk ----------')
```

```
stat, p = shapiro(df["JPM"])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')

# ———————— Python code ends ————————
```

Paython code for part 4:

```
# ———————————— Python code starts ————————————
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
from random import choices
import scipy.stats as stats
from scipy.stats import shapiro, normaltest
from matplotlib.pyplot import figure

#Sequential split of data
def split_data_seq(data, size):
    return np.array_split(data, size)

#To print sample mean
def print_mean(data):
    for index, value in enumerate(data):
        print("group" + str(index+1) + "——> " + str(np.mean(value)))

# Return array of sample mean
def mean_array(data):
    mean_value = []
    for value in data:
        mean_value.append(np.mean(value))
    return mean_value

# mean of sample means
def mean_mean(data):
    return np.mean(mean_array(data))

# standard deviation of mean
def std_of_samples(data):
    return np.std(mean_array(data))
```

```
# For spliting data randomly
def split_data_random(data, size, groups):
    random_array = []
    for i in range(groups):
        random_array.append(choices(data, k=size))
    return random_array

#1) Calculate the mean and the standard deviation of the population.

print('Mean of ETF column', np.mean(df['Close_ETF']))
print('Standard Deviation ETF column', np.std(df['Close_ETF']))

#2) Break the population into 50 groups sequentially and each
# group includes 20 values.
# Spliting 50 groups sequentially
seq_data_50 = split_data_seq(df['Close_ETF'], 50)

#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.

#sample mean
print("Sample mean of each group")
print_mean(seq_data_50)

#Histogram of mean of samples
mean_seq_data_50 = mean_array(seq_data_50)
figure(figsize=(12, 8), dpi=80)
plt.subplot(2,2,1)
plt.hist(mean_seq_data_50)
plt.subplot(2,2,2)
sns.distplot(mean_seq_data_50, hist=True, kde=True)

#Histogram of mean of samples for 1000 sample
array_of_mean = []
for i in range(20):
    seq_data = split_data_seq(df['Close_ETF'], 50)
```

```
    for value in seq_data:
        array_of_mean.append(np.mean(value))

print("Total Sample : ", len(array_of_mean))
sns.distplot(array_of_mean, hist=True, kde=True)

#4) Calculate the mean and the standard deviation of the
data including these sample means. Make a comparison between
#sd/sqrt(n) and standard deviation.

print("Mean : " + str(mean_mean(seq_data_50)))
print("Standard deviation : " + str(std_of_samples(seq_data_50)))

print("Mean : ", np.mean(df['Close_ETF'], axis=0) ,
"and mean of samples : ", mean_mean(seq_data_50))
print("sd/sqrt(n)" , (np.std(df['Close_ETF'])/math.sqrt(50)) ,
"and standard deviation of samples : ", std_of_samples(seq_data_50)

#5) Are the results from Items 3) and 4) consistent with the Central
Limit Theorem? Why?
#6) Break the population into 10 groups sequentially and each group
includes 100 values.

#Spliting 10 groups sequentially
seq_data_100 = split_data_seq(df['Close_ETF'], 10)

#7) Repeat Items 3) ~ 5).
print("Sample mean of each group")
print_mean(seq_data_100)

#Histogram of sample mean
mean_seq_data_100 = mean_array(seq_data_100)
figure(figsize=(12, 8), dpi=80)
plt.subplot(2,2,1)
plt.hist(mean_seq_data_100)
plt.subplot(2,2,2)
sns.distplot(mean_seq_data_100, hist=True, kde=True)
```

```
#Histogram of sample mean for 1000 size
array_of_mean = []
for i in range(100):
    seq_data = split_data_seq(df['Close_ETF'], 10)
    for value in seq_data:
        array_of_mean.append(np.mean(value))

print("Total Sample : ", len(array_of_mean))
sns.distplot(array_of_mean, hist=True, kde=True)

print("Mean : " + str(mean_mean(seq_data_100)))
print("Standard deviation : " + str(std_of_samples(seq_data_100)))

print("Mean : ", np.mean(df['Close_ETF'], axis=0) ,"and mean of samples : ",
mean_mean(seq_data_100))
print("sd/sqrt(n) : ", (np.std(df['Close_ETF'])/math.sqrt(100)) ,
"and standard deviation of samples : ", std_of_samples(seq_data_100))

#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.

#Spliting 50 simple random groups
random_data_50 = split_data_random(df['Close_ETF'], 20, 50)

#9) Repeat Items 3) ~ 5)
print("Sample mean of each group")
print_mean(random_data_50)

#Histogram of sample mean
mean_random_data_50 = mean_array(random_data_50)
figure(figsize=(12, 8), dpi=80)
plt.subplot(2,2,1)
plt.hist(mean_random_data_50)
plt.subplot(2,2,2)
sns.distplot(mean_random_data_50, hist=True, kde=True)

#Histogram of sample mean of 1000 sample
```

```
array_of_mean = []
for i in range(20):
    seq_data = split_data_random(df['Close_ETF'], 20, 50)
    for value in seq_data:
        array_of_mean.append(np.mean(value))

print("Total Sample : ", len(array_of_mean))
sns.distplot(array_of_mean, hist=True, kde=True)

print("Mean: " + str(mean_mean(random_data_50)))
print("Standard deviation : " + str(std_of_samples(random_data_50)))

print("Mean: ", np.mean(df['Close_ETF'], axis=0) ,"and mean of samples : ",
mean_mean(random_data_50))
print("sd/sqrt(n) " , (np.std(df['Close_ETF'])/math.sqrt(50)) ,
"and standard deviation of samples : ", std_of_samples(random_data_50))

#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.
#Spliting 10 simple random samples or groups
random_data_100 = split_data_random(df['Close_ETF'], 100, 10)

#11) Repeat Items 3) ~ 5).
print("Sample mean of each group")
print_mean(random_data_100)

#Histogram of mean of samples
mean_random_data_100 = mean_array(random_data_100)
figure(figsize=(12, 8), dpi=80)
plt.subplot(2,2,1)
plt.hist(mean_random_data_100)
plt.subplot(2,2,2)
sns.distplot(mean_random_data_100, hist=True, kde=True)

#Histogram of mean of 1000 samples
array_of_mean = []
for i in range(100):
```

```
    seq_data = split_data_random(df['Close_ETF'], 100, 10)
    for value in seq_data:
        array_of_mean.append(np.mean(value))

print("Total Sample : ", len(array_of_mean))
sns.distplot(array_of_mean, hist=True, kde=True)

print("Mean: " + str(mean_mean(random_data_100)))
print("Standard deviation : " + str(std_of_samples(random_data_100)))

print("Mean : ", np.mean(df['Close_ETF'], axis=0) ,"and mean of samples : ",
mean_mean(random_data_100))
print("sd/sqrt(n), (np.std(df['Close_ETF'])/math.sqrt(100)) ,
"and standard deviation of samples : ", std_of_samples(random_data_100))

# ---------- Python code ends ----------
```

Python Code for Part 5:

```
# ———————————— Part 5 Question 1 ————————————

# read input data
df = pd.read_csv('df.csv')

df_etf = df['Close_ETF']

# Make a sample

etf_sample_100 = df_etf.sample(n=100, replace=True, random_state=100)

122.156 - 1.960*(np.std(etf_sample_100)/np.sqrt(len(etf_sample_100))),
122.156 + 1.960*(np.std(etf_sample_100)/np.sqrt(len(etf_sample_100)))

# 95% Confidence Interval: (119.48113668779801, 124.830863312202)



% ———————————— Part 5 Question 2 ————————————

#x <- c(119.86, 126.20, 131.47, 138.58, 136.83, 102.94, 138.66, 140.53,

110.51, 140.74, 138.08, 99.62, 123.15, 132.52,

119.52,127.90,127.73,111.86,127.37,114.30)

# > wilcox.test(x, mu=122.156100, conf.int = T)

# alternative hypothesis: true location is not equal to 122.1561
# 95 percent confidence interval:
# 119.620 <-> 132.395



% ———————————— Part 5 Question 3 ————————————
```

 # Observation :

 # 121.152960 is the population mean and yes it is included in both the intervals calculated above the confidence interval constructed using the first sample gives more *#accurate result , since the* population size **is** greater **in** that case.

Python Code for Part 6:

```
# ———————————— Part 6 Question 1 ————————————

meanSampData = np.mean(etf_sample_100)
hypMean = 100
n = 100
std_pop = np.std(df_etf)

# We went with the z-test since the popluation std deviation is known
# which is 12.569 and the sample size is >30
# even though z-test assumes normal distribution and the data is not
# normally distributed, the sample size
# (100 in this case) is large enough to conduct the test
# Now the formula for z - value is

z = (meanSampData-hypMean)/(std_pop/np.sqrt(n))
# z = 17.6352

# Method 1(using p-value)
# Using the P-value approach: The p-value is p=0 and since 0<0.05
# it is concluded that the null hypothesis is rejected.

# Method 2(using critical values):
# this is a 2 sided test
# value of z at .05 making it .025 for 2 sided we know from z table
# z = (+ 1.96 to -1.96)

# as calculated z score 17.63 is greater than 1.96 (tabular z score),
we reject the null hypothesis
# Observed z-value = 17.63
# Critical value = 1.96



% ———————————— Part 6 Question 2 ————————————
```

```
x <- c(119.86, 126.2, 131.47, 138.58,136.83,
 102.94,138.66,140.53,110.51,140.74,138.08,
 99.62,123.15,132.5,119.5,127.90,127.73,
 111.86,127.37,114.30)
# > wilcox.test(x, mu = 100, alternative = "two.sided")
#     Wilcoxon signed rank test
# data:  x
# V = 209, p-value = 3.815e-06
# alternative hypothesis: true location is not equal to 100



% ---------- Part 6 Question 3 ----------

# Using the Chi-Square method(two tailed)
etf_sample_20 = df_etf.sample(n=20, replace=True,  random_state=100)
N = len(etf_sample_20)
stdSampData = np.std(etf_sample_20)
hypStd = 15

T = [(N-1) * ((stdSampData/hypStd)**2)]
# T = 12.569%

# https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm
# reject if greater than 32.852  and less than 8.907
# Hence we failed to reject the null hypothesis

# But for random_state = 0 the null hypothesis is getting rejected

---------- Part 6 Question 4 ----------

# Using the Chi-Square method one tailed

N = 20
stdSampData = np.std(etf_sample_20)
hypStd = 15

T = [(N-1) * ((stdSampData/hypStd)**2)]
```

```
# T = 12.569

# https://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm
# reject if less than 10.117
# Hence we failed to reject the null hypothesis

# But for random_state = 0 the null hypothesis is getting rejected
```

Python Code for Part 7:

```
# ———————————— Part 7 Question 1 ————————————

# manual formula method
x = np.array(df['oil'])
y = np.array(df['gold'])

t = (np.mean(x)-np.mean(y))/np.sqrt(((np.std(x)*np.std(x))/len(x))+
((np.std(y)*np.std(y))/len(y)))
pval = st.t.sf(np.abs(t), 1000-1)*2

t , pval
# (0.48560947929478604, 0.6273505577888208)

# Observations:-
# We failed to reject the null hypothesis
(the means of oil and gold are equal) for 0.05 significance level

% ———————————— Part 7 Question 2 ————————————

# library method

ttest_rel(df['oil'], df['gold'])

#Ttest_relResult(statistic=0.5413, pvalue=0.5884)

# Observations:

# In the example we find a t-value of 0.5413 with a significance of
0.58 or 58%
# This means that there is a 58% chance of a t-value of less than
-0.5413 or a t-value of more than 0.5413,
# hence we failed to reject the null hypothesis

# With a usual 0.05 significance level we consider this chance to be
so high, that most likely there is actually not a significant
```

```
difference in the population as well

# manual formual mehtod

d = df['oil'] - df['gold']
mean_d = np.mean(d)
std_d = np.std(d)
d_sqrd = d*d

t = np.sum(d)/np.sqrt((1000*np.sum(d_sqrd)-(np.sum(d)*np.sum(d)))/(998))
t
# 0.5410599236152 62

# the critical value for significance level 0.05 and dof 999 is 1.96

# since 0.541 is between + or - 1.96
(we failed to reject the null hypothesis)

% ---------- Part 7 Question 3 ----------

var_oil = np.var(x)
var_gold = np.var(y)

F = (np.power(var_oil,2))/np.power(var_gold,2)
F
# 12.187479283391903

# Observations

# Critical f-values:     FL=0.883 and FU=1.132

# since the F value is towards the right of the critical value,
we are in the rejection region

# hence we reject the null hypothesis
```

Python Code for Part 8:

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?

```
# ———————————— Python code starts ————————————
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
import seaborn as sns
%matplotlib inline

# read input data
pdInputData = pd.read_excel("ProjectInputData.xlsx")

# rename columns to meaningful column names
pdInputData.rename(columns={"Close_ETF" : "DailyETFReturns", "oil":
"DailyRelativeChangeOilPrice", \
"gold": "DailyRelativeChangeGoldPrice",
"JPM": "DailyJPMStockReturns"}, inplace = True)


# Keeping ETF and Gold columns only removing oil, JPM columns
pdInputDataP8 = pdInputData[['DailyETFReturns',
'DailyRelativeChangeGoldPrice']].copy()

# Draw a scatter plot of ETF (Y) vs. Gold (X)
x = pdInputDataP8['DailyRelativeChangeGoldPrice']
y = pdInputDataP8['DailyETFReturns']
plt.figure(figsize=(15,7))
plt.scatter(x, y)                    #c = 'DarkBlue'
plt.xlabel("Daily Relative Change in Gold Price", fontsize = 20)
plt.ylabel("Daily ETF Returns", fontsize = 20)


#  save image or graph as .png
plt.title("Scatter plot: 'Daily ETF Returns' Vs 'Daily
```

```
Relative␣Change␣in␣Gold␣Price'", fontsize = 20)
plt.savefig('Part8Q1.png', bbox_inches='tight')
plt.show()
# ———————————— Python code ends ————————————
```

2) Calculate the coefficient of correlation between ETF and Gold and interpret it.

```
# ———————————— Python code starts ————————————
pdInputDataP8.corr(method='pearson').\
 style.format("{:.2}").background_gradient
 (cmap=plt.get_cmap('coolwarm'), axis=1)

# ———————————— Python code ends ————————————
```

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?

```
# ———————————— Python code starts ————————————
# Get xs and ys
x = pdInputDataP8['DailyRelativeChangeGoldPrice']
y = pdInputDataP8['DailyETFReturns']

X = np.array(x)
Y = np.array(y)

# Method 0:
import statsmodels.api as sm

# We have line equation as Y = 0 + 1X + u

# Add constant along with variables
X = sm.add_constant(x)
```

```
print("Variables and Constant \n\n", X.head())

# Find the best fit line
results = sm.OLS(y, X).fit()


Param_interceptConst = results.params.to_dict()['const']
Param_slopeValue = results.params.to_dict()['DailyRelativeChangeGoldPrice']

regression_line = [(Param_slopeValue*
xi)+Param_interceptConst for xi in
X['DailyRelativeChangeGoldPrice']]

print("The slope of line (m) and the intercept (b) are:",
Param_slopeValue, Param_interceptConst)

# Draw the scatter plot and regression line graph
plt.figure(figsize=(15,7))
plt.scatter(x, y)
plt.xlabel("Daily Relative Change in Gold Price", fontsize = 20)
plt.ylabel("Daily ETF Returns", fontsize = 20)
plt.plot(x, regression_line, 'c', c = 'orange')          #c = 'DarkBlue'
plt.title("Scatter plot: Daily ETF Returns Vs Daily
Relative Change in Gold Price", fontsize = 20)

#  Save image or graph as .png

plt.savefig('Part8Q3.png', bbox_inches='tight')
plt.show()
# ---------------------- Python code ends ----------------------
```

4) Conduct a two-tailed t-test with $H_0 : \beta_1 = 0$. 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?

```
# ---------------------- Python code starts ----------------------
Param_interceptConst = results.params.to_dict()['const']
Param_slopeValue = results.params.to_dict()['DailyRelativeChangeGoldPrice']

#The standard errors show the accuracy of prediction for each variable.
#The lower the standard error, the better the estimate!
SE_Beta_0 = results.bse.to_dict()['const']
SE_Beta_1 = results.bse.to_dict()['DailyRelativeChangeGoldPrice']

# Get slope and intercept of the line: (that is parameters)
print("The constant term is {} with a standard error of {}"
.format(Param_interceptConst, SE_Beta_0))
print("The slope parameter is {} with a standard error of  {}"
.format(Param_slopeValue, SE_Beta_1))


# Method 1: Get two tailed test t score
tValues = results.tvalues
print("The t-value for \nconstant:                    {}\
                      \nDailyRelativeChangeGoldPrice: {}"\
.format(tValues.const , tValues.DailyRelativeChangeGoldPrice ))

# Consider the line: Y_hat = Beta_0 + Beta_1 x          OR
Y =  0  +  1X

# In case null hypothesis is Beta_0 = 0 (default case) then t score becomes
# t = ( Beta_hat_0 - Beta_0 ) / ( SE(Beta_0) )
print("Null hypothesis H0: the line crosses the y-axis at the origin.")
print("t Score (Beta0 = 0): ", ( Param_interceptConst  - 0 ) / SE_Beta_0)


# In case null hypothesis is Beta_1 = 0 (default case) then t score becomes
# t = ( Beta_hat_1 - Beta_1 ) / ( SE(Beta_1) )
print("\n\nNull hypothesis H0:
'Daily Relative Change in Gold Price' does not affect 'Daily ETF Returns'")
t_score = ( Param_slopeValue - 0 ) / SE_Beta_1
print("t Score (Beta1 = 0): ", t_score)
```

```
## Null hypothesis H0: the line crosses the y-axis at the origin.(Beta0=0)
plt.figure(figsize=(14,7))
plt.scatter(x,y)

y_hat = Param_slopeValue  x + 0

fig = plt.plot(x, y_hat, lw=4, c= 'orange', label = 'regression line')

plt.title("Null hypothesis H0: the line crosses the y-axis \
at the origin.(Beta0=0)", fontsize = 20)
plt.xlabel("Daily Relative Change in Gold Price", fontsize = 20)
plt.ylabel("Daily ETF Returns", fontsize = 20)

# Save the file
plt.savefig('Part8Q4_3.png', bbox_inches='tight')
plt.show()

## Null hypothesis H0: 'Daily Relative Change
# in Gold Price' does not affect 'Daily ETF Returns'

plt.figure(figsize=(15,7))
plt.scatter(x,y)

y_hat = 0 * x + Param_interceptConst

fig = plt.plot(x, y_hat, lw=4, c= 'orange', label = 'regression line')
plt.title("H0: 'Daily Relative Change in Gold Price' does not affect
'Daily ETF Returns'(B1=0)", fontsize = 20)
plt.xlabel("Daily Relative Change in Gold Price", fontsize = 20)
plt.ylabel("Daily ETF Returns", fontsize = 20)
plt.show()
degree_of_freedom = 998  #DF of Residual in the report
print("Degree of freedom is:", degree_of_freedom)
critical_value = 2.58
print("From the standard t distribution table - With
significance Level = 1% ( two sided ) we have:
Critical value as:", critical_value)
```

```
# Method 1: Calculate p value
pValues = results.pvalues
print("The p-value for \nconstant: ____________________{}\
_____________________\nDailyRelativeChangeGoldPrice: {}"\
      .format(pValues.const , pValues.DailyRelativeChangeGoldPrice ))

# ---------------------- Python code ends ------------------
```

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?

```
# ---------------------- Python code starts ------------------
def coefficient_of_determination(ys_orig,ys_line):
    y_mean_line = [mean(ys_orig) for y in ys_orig]
    squared_error_regr = squared_error(ys_orig, ys_line)
    squared_error_y_mean = squared_error(ys_orig, y_mean_line)
    return 1 - (squared_error_regr/squared_error_y_mean)

r_squared = coefficient_of_determination(ys, regression_line)
print("R_square:", r_squared)
# ---------------------- Python code ends ------------------
```

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

```
# ---------------------- Python code starts ------------------
#1. Assumption of Linearity check:
#Method 1: Corelation matrix
print(pdInputData[["Close_ETF", "gold"]].corr())
#Method 2 by using Matrix scatter plots
#Plot pairwise relationships in a dataset with one independent
vairable xi and one dependent variable yi
```

```
sns.pairplot(pdInputData[["Close_ETF", "gold"]], kind ='reg')

# Or just use first column which gives more clear picture:
sns.pairplot(pdInputData , x_vars=['gold', "oil"], y_vars=["Close_ETF"],
             height=5, aspect=.8, kind="reg");

#Method 3 plot residuals against predictor variable (Yi)
#Before running a model this can be done:
plt.figure(figsize=(15,7))
sns.residplot(x = 'gold',
              y = "Close_ETF",
              data = pdInputData)

plt.xlabel("Prediction Value (y_hat)", fontsize = 20)
plt.ylabel("Residual (y_actual - y_hat)", fontsize = 20)
plt.title("Scatter plot: Residual Vs Predictor", fontsize = 20)
plt.show()

### # Plot Predict Vs Residual To Check Linearity
import seaborn as sns   # For Visualization
plt.figure(figsize=(15,7))
sns.regplot(x=y_pred,y=serResidual)
plt.xlabel("Prediction Value (y_hat)", fontsize = 20)
plt.ylabel("Residual Value (y - y_hat)", fontsize = 20)
plt.title("Scatter plot: Residual Value Vs
Prediction Value (y_hat)", fontsize = 20)
plt.show()

plt.figure(figsize=(15,7))
ax = sns.distplot(serResidual)
plt.axvline(np.mean(serResidual), color="b", linestyle="dashed",
linewidth=5)
_, max_ = plt.ylim()
plt.text(          serResidual.mean() + serResidual.mean() / 10,
max_ - max_ / 10, "Mean: {:.2f}".format(serResidual.mean()),
    )

#2. Independent Errors assumption
```

```
#Before running a model this can be done:
plt.figure(figsize=(15,7))
sns.residplot(x = 'gold',
              y = "Close_ETF",
              data = pdInputData)
#This method is used to plot the residuals of linear regression.
#This method will regress y on x and then draw a scatter
# plot of the residuals.
#You can optionally fit a lowess smoother to the residual plot,
# which can help in determining if
#there is a structure to the residuals.
# Return: Axes with the regression plot.
plt.xlabel("Prediction Value (y_hat)", fontsize = 20)
plt.ylabel("Residual (y_actual - y_hat)", fontsize = 20)
plt.title("Scatter plot: Residual Vs Predictor", fontsize = 20)
plt.show()

#3 Assumption of Homoscedasticity
def homoscedasticity_assumption(model, features, label):
    """
    Homoscedasticity: Assumes that the errors exhibit constant variance
    """
    print('Assumption 5: Homoscedasticity of Error Terms', '\n')

    print('Residuals should have relative constant variance')

    # Plotting the residuals
    #Before running a model this can be done:
    plt.figure(figsize=(15,7))
    sns.residplot(x = 'gold',
                  y = "Close_ETF",
                  data = pdInputData)
    #This method is used to plot the residuals of linear
    # regression.
    #This method will regress y on x and then draw a scatter
    # plot of the residuals.
    #You can optionally fit a lowess smoother to the residual
    #plot, which can help in determining if
```

```
    #there is a structure to the residuals.
    # Return: Axes with the regression plot.
    plt.xlabel("Prediction Value (y_hat)", fontsize = 20)
    plt.ylabel("Residual (y_actual - y_hat)", fontsize = 20)
    plt.title("Scatter plot: Residual Vs Predictor", fontsize = 20)
    plt.show()

homoscedasticity_assumption(model, pdInputData[['gold']],
                            pdInputData["Close_ETF"])

#4 Multi Collinearity
#calculate VIF for each explanatory variable
def get_vif(X_features):
    pdVif = pd.DataFrame()
    pdVif['VIF'] = [variance_inflation_factor(X.values, i)
    for i in range(X.shape[1])]
    pdVif['variable'] = X.columns
    return pdVif

def get_tolerance_value(pdVif):
    pdVif = get_vif(X_features)
    return 1/pdVif['VIF']

X_features = pdInputData[['gold']]
pdVif = get_vif(X_features)
pdVif['Tolerance'] = get_tolerance_value(pdVif)
print(pdVif)
%# ---------- Python code ends ----------
```

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.

```
# ---------- Python code starts ----------
def get_confidance_intervals(floatAlpha, n_1, pop_mean):
    return st.t.interval(alpha=floatAlpha, df=n_1, loc=pop_mean)

# 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

n_1 = len(pdInputDataP8)-1
daily_relative_cahnge = 0.005127
floatAlpha = 0.99

confidence_interval = get_confidance_intervals(floatAlpha, n_1,
daily_relative_cahnge)
print("Confidance interval for gold is (with daily
relative change in the gold price is 0.005127) ",
confidence_interval, "For alpha: ", floatAlpha)

# Method 1: Without scaled

import scipy.stats as st
#create 99% confidence interval for same sample

n_1 = len(pdInputDataP8)-1
pop_mean = np.mean(pdInputDataP8.DailyETFReturns)
floatAlpha = 0.99
confidence_interval  = get_confidance_intervals(floatAlpha,
n_1, pop_mean)
print("Confidance interval for Close_ETF is ", confidence_interval,
"For alpha: ", floatAlpha)

from statsmodels.sandbox.regression.predstd import wls_prediction_std
#sdev, lower_pred, upper_pred
sdev, lower_pred, upper_pred =  wls_prediction_std(fitted,
exog=np.array(pdInputDataP8['DailyRelativeChangeGoldPrice'])
.reshape(-1,1), alpha=0.99)
plt.plot(lower_pred, 'r-', label = 'Lower Bound')
plt.plot(upper_pred,'b-',label = 'Upper Bound')
plt.legend()
plt.show()
# ---------- Python code ends ----------
```

Python Code for Part 9:

Consider the data including the ETF, Gold and Oil column. Using any software, fit a multiple linear regression model to the data with the ETF variable as the response. Evaluate your model with adjusted R^2.

```
# ---------- Python code starts ----------
pdInputDataP9 = pdInputData.copy()
pdInputDataP9.drop(['DailyJPMStockReturns'], axis=1, inplace=True)
pdInputDataP9.head()

feature_column_names = ['DailyRelativeChangeOilPrice',
'DailyRelativeChangeGoldPrice']

import statsmodels.api as sm

x = pdInputDataP9[feature_column_names]
y = pdInputDataP9.DailyETFReturns

#fit an OLS model to data
x_with_constant = sm.add_constant(x)
model = sm.OLS(y, x_with_constant)
results = model.fit()

print(results.summary())
print(results.rsquared_adj)

### # Plot Predict Vs Residual To Check Linearity
import seaborn as sns   # For Visualization
plt.figure(figsize=(15,7))
serResidual = results.resd
sns.regplot(x=y_pred,y=serResidual)
plt.xlabel("Prediction Value (y_hat)", fontsize = 20)
plt.ylabel("Residual Value (y - y_hat)", fontsize = 20)
plt.title("Scatter plot: Residual Value Vs Prediction \
Value (y_hat)", fontsize = 20)
plt.show()
# ---------- Python code ends ----------
```

Python Code for Part 10:

Calculate the residuals of the model fitting you did in Part 9. Check the four assumptions made for the error terms of the multiple regression model using these residuals (mean 0; constant variance; normality; and the independence). You may draw some plots over the residuals to check these assumptions. For example, draw a Normal Probability Plot to check the normality assumption; draw a scatter plot of Residuals vs. Fitted Values to check the constant variance assumption and the independence assumption; and so on.

```
# ———————————— Python code starts ————————————
# 1. Assumption of linearity
#Plot pairwise relationships in a dataset.
print(sns.pairplot(
 pdInputData[['gold', "oil", 'Close_ETF']], kind ='reg') )

# Or just use first column which gives more clear picture:
sns.pairplot(pdInputData , x_vars=['gold', "oil"],
y_vars=["Close_ETF"], height=5, aspect=.8, kind="reg");

# Corelation between predictors and output Close_ETF + among the
# predictors
print(pdInputData[["Close_ETF", "gold", "oil"]].corr() )

# normality of residuals
plt.figure(figsize=(15,7))
ax = sns.distplot(serResidual)
plt.axvline(np.mean(serResidual), color="b", linestyle="dashed",
linewidth=5)
_, max_ = plt.ylim()
plt.text(          serResidual.mean() + serResidual.mean() / 10, max_ -
max_ / 10, "Mean: {:.2f}".format(serResidual.mean()),
    )
# 2. Independent Errors assumption
fig, ax = plt.subplots(figsize=(20,10))
_, (__, ___, r) = sp.stats.probplot(serResidual, plot=ax, fit=True)

#3 Assumption of Homoscedasticity
```

```
#Scale Location plot
# model values
model_fitted_y = results.predict()
# model residuals
model_residuals = results.resid
# normalized residuals
model_norm_residuals = results.get_influence().resid_studentized_internal
# absolute squared normalized residuals
model_norm_residuals_abs_sqrt = np.sqrt(np.abs(model_norm_residuals))
# absolute residuals
model_abs_resid = np.abs(model_residuals)
# leverage, from statsmodels internals
model_leverage = results.get_influence().hat_matrix_diag
# cook's distance, from statsmodels internals
model_cooks = results.get_influence().cooks_distance[0]

plot_lm_3 = plt.figure(figsize=(15,7))
plt.scatter(model_fitted_y, model_norm_residuals_abs_sqrt, alpha=0.5);
sns.regplot(model_fitted_y, model_norm_residuals_abs_sqrt,
            scatter=False,
            ci=False,
            lowess=True,
            line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8});
plot_lm_3.axes[0].set_title('Scale-Location', size = 20)
plot_lm_3.axes[0].set_xlabel('Fitted values', size = 20)
plot_lm_3.axes[0].set_ylabel('$\sqrt{|Standardized Residuals|}$', size = 20);

# annotations
abs_sq_norm_resid = np.flip(np.argsort(model_norm_residuals_abs_sqrt), 0)
#abs_norm_resid_top_3 = abs_norm_resid[:3]
abs_sq_norm_resid_top_3 = abs_sq_norm_resid[:3]
for i in abs_sq_norm_resid_top_3:
    plot_lm_3.axes[0].annotate(i,
                               xy=(model_fitted_y[i],
                                   model_norm_residuals_abs_sqrt[i]));

#Residuals vs Fitted
plot_lm_1 = plt.figure(figsize=(15,7))
```

```
plot_lm_1.axes[0] = sns.residplot(model_fitted_y, pdInputData.columns[-1], \
                          data=pdInputData,
                          lowess=True,
                          scatter_kws={'alpha': 0.5},
                          line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})

plot_lm_1.axes[0].set_title('Residuals vs Fitted', size = 20)
plot_lm_1.axes[0].set_xlabel('Fitted values', size = 20)
plot_lm_1.axes[0].set_ylabel('Residuals', size = 20)


#4 Multi Collinearity
X = pdInputData[['gold', "oil"]]
### This will work in case of multiple columns of
### the features for single input column following code does not work
vif = [variance_inflation_factor(X.values, i) for i in
range(X.shape[1])]
pdVif = pd.DataFrame({'vif': vif[0:]}, index=X.columns).T
print(pdVif)

print("For Gold the tolerance is: ", 1/pdVif['gold']['vif'])
print("For Oil the tolerance is: ", 1/pdVif['oil']['vif'])

# 5. Additional analysis
# Residuals vs Leverage:
plot_lm_4 = plt.figure(figsize=(15,7))
plt.scatter(model_leverage, model_norm_residuals, alpha=0.5)
sns.regplot(model_leverage, model_norm_residuals,
              scatter=False,
              ci=False,
              lowess=True,
              line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plot_lm_4.axes[0].set_xlim(0, max(model_leverage)+0.01)
plot_lm_4.axes[0].set_ylim(-3, 5)
plot_lm_4.axes[0].set_title('Residuals vs Leverage', size = 20)
plot_lm_4.axes[0].set_xlabel('Leverage', size = 20)
plot_lm_4.axes[0].set_ylabel('Standardized Residuals', size = 20)
```

```
# annotations
leverage_top_3 = np.flip(np.argsort(model_cooks), 0)[:3]
for i in leverage_top_3:
    plot_lm_4.axes[0].annotate(i,
                               xy=(model_leverage[i],
                                   model_norm_residuals[i]))
# ------------------- Python code ends ------------------
```