Submission Number: 03

Group Number: 02

Group Members:

Full Legal Name	Location (Country)	E-Mail Address	Non-Contributing Member (X)
Bui Khac Tu	Vietnam	bkt992@gmail.com	
Aghogho Esuoma Monorien	Nigeria	monorienaghogho@gmail.com	
Emmanuel Amfo	Ghana	emmanuel.belvin@yahoo.com	

Statement of integrity:

Bui Khac Tu, Aghogho Esuoma Monorien, Emmanuel Amfo

1. **Which model - VARMA , classification NN, or regression NN -- provides the best fit to the data? Why?**

Among the models, the VARMA model provides a better fit to the data because less standard error is produced when modeling using VARMA than Neural Networks. And in forecasting, the correlation shown by VARMA is better as compared to Neural Networks. VARMA shows cross-correlations between series by estimating time series. VARMA can be used in multivariate cases and reduce the effects of co-integration. Data with nonlinear dependencies can be better handled with Neural Networks.

2. **Which model provides better interpretation of the results?**

The results are closely connected to the metric evaluations in the sense that a particular model does not fully outweigh the other in overall advantage. In Neural networks modeling, pre-processing of the data being inputted to it, has caused the errors and inaccuracies having

lesser interpretation. So among the models, the VARMA model is the most preferable in terms of interpretability of the results.

3. **Report on group member contribution**

Our group has divided the work below:

a. Bui and Aghogho with some work experiences in relation to python completed the project with python codes

b. Emmanuel completed the report with regard to the specifics in the interpretation of the models' result with the assistance from Bui and Aghogho, explaining results of VARMA and NN modeling

c. Questions 1 through 8 are done jointly by Bui and Aghogho.

d. Questions 9 to 13 is done by Emmanuel

e. We worked together on generally editing and finalizing the code and questions.

4. **Write a TECHNICAL 1-page report of your findings to your FE boss**

Introduction

This project report seeks to analyze data modeling with VARMAX and NNs, data exploration, reviewing the Skewness and Kurtosis of the data

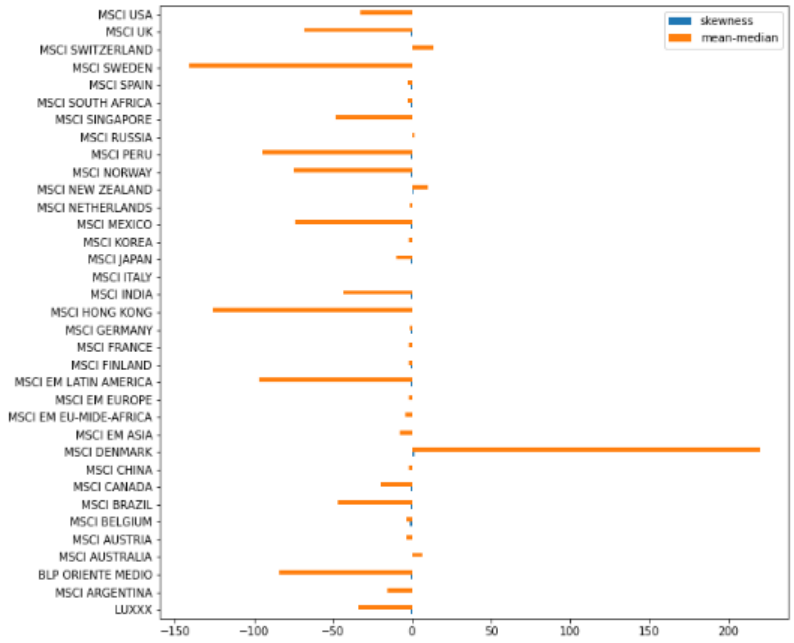
Analysis

In measuring the skewness of the data, the skewness of the return series were computed, and the difference between the mean and median for each series were calculated. By comparing the skewness and the (mean-median) difference, a relationship graph was obtained.

```
plt.figure(figsize=(20,20))
pd.DataFrame([skewness, mmdiff], index=['skewness', 'mean-median']).T.plot(kind='barh', figsize=(10,10))
```

```
<AxesSubplot:>
```

```
<Figure size 1440x1440 with 0 Axes>
```



With the result obtained, for example, MSCI Belgium has a negative skewness as a result of the mean less than the median. For MSCI Denmark, the mean was greater than the median resulting in positive skewness. Whenever the mean, median and mode are all equal, the skewness is zero, showing a symmetrical shape. This can be seen in the case of MSCI USA, which has a very low skewness to be a symmetric distribution.

```
negative_skew = skewness[skewness < 0]
mean_less_median = mmdiff[negative_skew.index]

pd.DataFrame([negative_skew, mean_less_median], index=['skewness', 'mean-median']).T
```

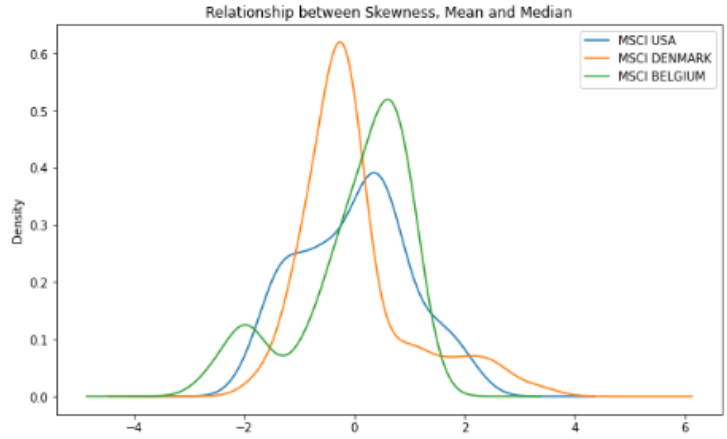
	skewness	mean-median
LUXXX	-0.743562	-33.849095
BLP ORIENTE MEDIO	-0.597361	-83.931072
MSCI AUSTRIA	-0.156242	-3.216111
MSCI BELGIUM	-1.090393	-3.724762
MSCI BRAZIL	-0.549140	-46.970833
MSCI CANADA	-0.835155	-19.808373
MSCI CHINA	-0.209975	-1.921786
MSCI EM ASIA	-0.319299	-7.896944
MSCI EM EU-MIDE-AFRICA	-0.272154	-4.066091
MSCI EM EUROPE	-0.104341	-2.345357
MSCI EM LATIN AMERICA	-0.750381	-96.492262
MSCI FINLAND	-0.648941	-2.194722
MSCI FRANCE	-0.226843	-2.164524
MSCI GERMANY	-0.509066	-1.122103
MSCI HONG KONG	-0.333212	-125.678849
MSCI INDIA	-0.423050	-43.322024
MSCI ITALY	-0.192386	-0.160873
MSCI JAPAN	-0.437756	-9.907857
MSCI KOREA	-0.195284	-2.258095
MSCI MEXICO	-0.865101	-73.685000
MSCI NETHERLANDS	-0.059582	-1.686825
MSCI NORWAY	-0.380425	-74.667817
MSCI PERU	-0.405124	-94.627183
MSCI SINGAPORE	-0.254885	-48.059245
MSCI SOUTH AFRICA	-0.560744	-2.966865
MSCI SPAIN	-0.651333	-2.606587
MSCI SWEDEN	-0.175451	-141.255952
MSCI UK	-0.950727	-68.507421

```
positive_skew = skewness[skewness > 0]
mean_greater_median = mmdiff[positive_skew.index]

pd.DataFrame([positive_skew, mean_greater_median], index=['skewness', 'mean-median']).T
```

	skewness	mean-median
MSCI ARGENTINA	0.274648	-15.322738
MSCI AUSTRALIA	0.171393	6.565159
MSCI DENMARK	1.368847	219.852738
MSCI NEW ZEALAND	0.963410	10.054444
MSCI RUSSIA	0.160923	1.082738
MSCI SWITZERLAND	0.250563	13.266468
MSCI USA	0.068534	-32.631468

```
example = etf.loc[:,['MSCI USA', 'MSCI DENMARK', 'MSCI BELGIUM']]
example = (example - example.mean())/example.std()
example.plot(kind='kde', title='Relationship between Skewness, Mean and Median', figsize=(10,6))
```

```
<AxesSubplot:title={'center':'Relationship between Skewness, Mean and Median'}, ylabel='Density'>
```



The Kurtosis of our response return series was calculated. We then identified a regime shift that divides the data to 2 regime by running a threshold regression model. A key issue with time-series models is when the data under-goes a regime change. This is when different segments of the data have different statistical properties. We analyzed the LUXX response series for 2 different regimes using the autoregression classifier and discovered a structural break resulting in the 2 different regimes. The standard deviations of the return series for the regimes were calculated.

```
luxx = etf['LUXXX']
luxx_return = luxx.apply(np.log).diff(1).dropna().to_numpy()

kur = pd.Series(stats.kurtosis(luxx_return))

print('Kurtosis for LUXXX return series: ', kur)
```

```
Kurtosis for LUXXX return series:  0    4.310033
dtype: float64
```

```
================================================================================
|                                Lower Regime                                  |
================================================================================

                            OLS Regression Results
================================================================================
Dep. Variable:                      y   R-squared:                       0.107
Model:                            OLS   Adj. R-squared:                  0.087
Method:                 Least Squares   F-statistic:                     5.274
Date:                Fri, 14 Jan 2022   Prob (F-statistic):             0.0265
Time:                        22:09:17   Log-Likelihood:                 79.525
No. Observations:                  46   AIC:                            -155.1
Df Residuals:                      44   BIC:                            -151.4
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         -0.0126      0.006     -1.951      0.057      -0.026       0.000
x1             0.3378      0.147      2.296      0.026       0.041       0.634
==============================================================================
Omnibus:                       13.958   Durbin-Watson:                   1.941
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.527
Skew:                          -1.130   Prob(JB):                     0.000425
Kurtosis:                       4.730   Cond. No.                         22.7
==============================================================================

================================================================================
|                                Upper Regime                                  |
================================================================================

                            OLS Regression Results
================================================================================
Dep. Variable:                      y   R-squared:                       0.077
Model:                            OLS   Adj. R-squared:                  0.058
Method:                 Least Squares   F-statistic:                     4.078
Date:                Fri, 14 Jan 2022   Prob (F-statistic):            0.00338
Time:                        22:09:17   Log-Likelihood:                 430.16
No. Observations:                 201   AIC:                            -850.3
Df Residuals:                     196   BIC:                            -833.8
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          0.0022      0.002      0.981      0.328      -0.002       0.007
x1            -0.1745      0.069     -2.533      0.012      -0.310      -0.039
x2            -0.0364      0.088     -0.414      0.679      -0.209       0.137
x3            -0.2047      0.071     -2.877      0.004      -0.345      -0.064
x4            -0.1250      0.066     -1.889      0.060      -0.256       0.006
==============================================================================
Omnibus:                       13.528   Durbin-Watson:                   2.018
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               33.051
Skew:                           0.173   Prob(JB):                     6.65e-08
Kurtosis:                       4.956   Cond. No.                         44.2
==============================================================================
```

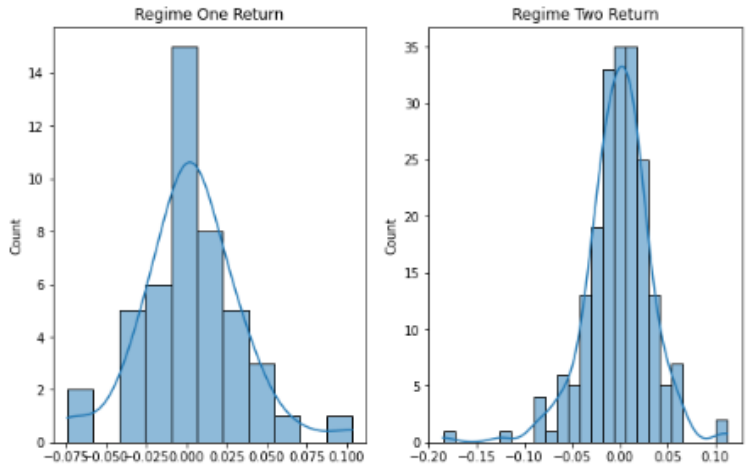
```
print(f' Standard deviation of regime one: {regime_one.std()}')
print(f' Standard deviation of regime two: {regime_two.std()}')
```

```
 Standard deviation of regime one: 0.030944929282437538
 Standard deviation of regime two: 0.03438310084262847
```

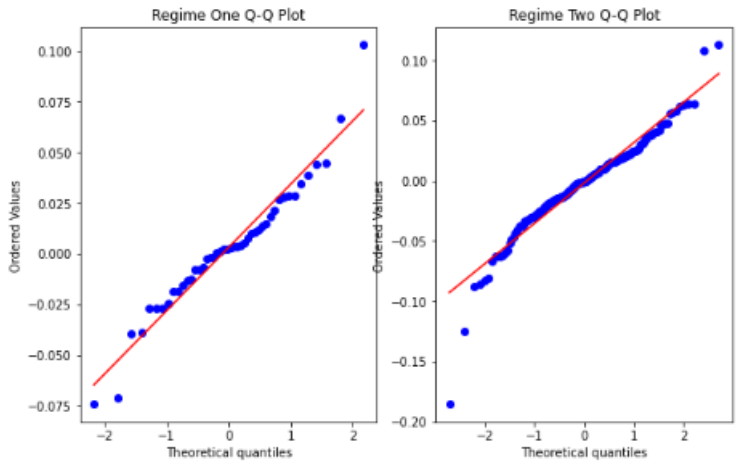
In Visualizing Distributions, we ran the regime shift models with the response variable to show both the histograms and QQ plot for the return series.

```
plt.figure(figsize=(15,6))
plt.subplot(1, 3, 1)
sns.histplot(data = regime_one, kde = True)
plt.title('Regime One Return')

plt.subplot(1, 3, 2)
sns.histplot(data = regime_two, kde = True)
plt.title('Regime Two Return')
```

```
Text(0.5, 1.0, 'Regime Two Return')
```



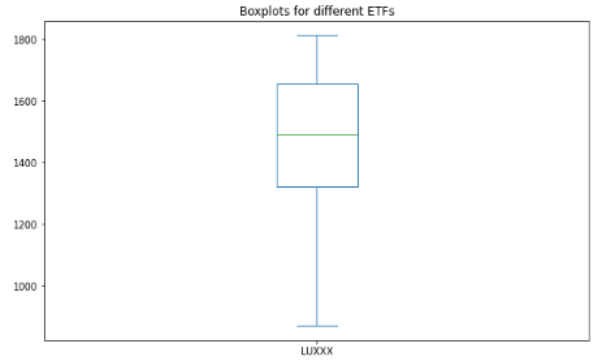

```
plt.figure(figsize=(15,6))

plt.subplot(1, 3, 1)
stats.probplot(regime_one, dist="norm", plot=matplotlib.pyplot)
plt.title('Regime One Q-Q Plot')

plt.subplot(1, 3, 2)
stats.probplot(regime_two, dist="norm", plot=matplotlib.pyplot)
plt.title('Regime Two Q-Q Plot')

plt.show()
```



We identified outliers using a combination of computed stats and visuals. The boxplots below indicate there are outliers in MSCI BELGIUM but no outliers in LUXX ETF.

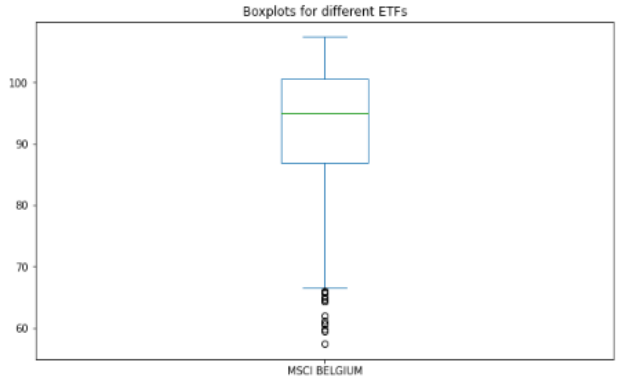
```
example_one = etf.loc[:,['LUXXX', 'MSCI BELGIUM']]
example_one['LUXXX'].plot(kind='box', title='Boxplots for different ETFs', figsize=(10,6))
```

```
<AxesSubplot:title={'center':'Boxplots for different ETFs'}>
```




```
example_one['MSCI BELGIUM'].plot(kind='box', title='Boxplots for different ETFs', figsize=(10,6))
```

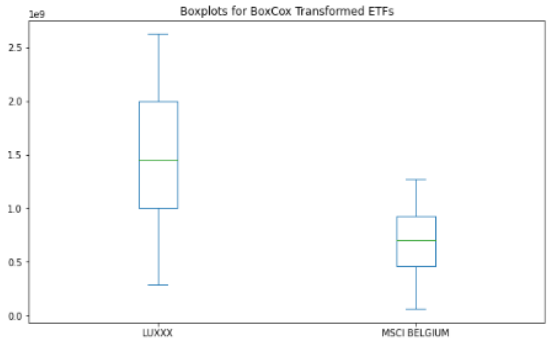
```
<AxesSubplot:title={'center':'Boxplots for different ETFs'}>
```



In order to remove the outliers, a monotone transformation such as Boxcox transformation was performed to remove the outliers as shown below.

```
example_one = etf.loc[:,['LUXXX', 'MSCI DENMARK', 'MSCI BELGIUM']]

belgium_a,belgium_b = stats.boxcox(example_one['MSCI BELGIUM'])
luxx_a,luxx_b = stats.boxcox(example_one['LUXXX'])

df = pd.DataFrame([luxx_a, belgium_a], index=['LUXXX', 'MSCI BELGIUM']).T
df.plot(kind='box', title='Boxplots for BoxCox Transformed ETFs', figsize=(10,6))
```

```
<AxesSubplot:title={'center':'Boxplots for BoxCox Transformed ETFs'}>
```



A VARMA model was performed to model our response against lagged versions using variables from LASSO regression. VARMA can model current output as a linear relationship with current and past values of a stochastic (error) term. Comparing VARMA and LASSO, VARMA models a much more complex relationship with past values while LASSO which can be used for variable selection and regularization only model linear relationships at the current time.

```
PRED = res.predict()['LUXXX']
```

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
mean_squared_error(etf['LUXXX'], PRED), r2_score(etf['LUXXX'], PRED)
```

```
(2062.2231198948193, 0.9636352398451485)
```

```
PRED2 = res.predict()['LUXXX']
mean_squared_error(etf['LUXXX'], PRED2), r2_score(etf['LUXXX'], PRED2)
```

```
(1498.6600544264, 0.9735729791276648)
```

A classification NN with a categorical response was performed using suitable layers based on cross-validation of results. Additionally, a performance of Regression NN with a continuous response was carried out with the series weekly return value with a suitable number of layers.

In [36]:
```
# Input -> 34 other variable => result = label

log_returns = etf.select_dtypes(exclude=['object']).apply(np.log).diff(1).dropna()
# X_cs = log_returns[1:].to_numpy()

X = etf.iloc[1:-1, 2:]
X.head()
```

Out[36]:

	MSCI ARGENTINA	BLP ORIENTE MEDIO	MSCI AUSTRALIA	MSCI AUSTRIA	MSCI BELGIUM	MSCI BRAZIL	MSCI CANADA	MSCI CHINA	MSCI DENMARK	MSCI EM ASIA	MSCI EM EU-MIDE-AFRICA	MSCI EM EUROPE	MSCI EM LATIN AMERICA	MSCI FINLAND	MSCI FRANCE
1	2260.85	3280.6683	1005.56	97.66	99.35	952.01	1586.18	54.63	8183.00	377.65	194.927	231.93	1674.47	115.86	122.85
2	2217.50	3118.2981	985.38	93.54	97.32	904.64	1541.08	51.54	7755.73	363.18	181.941	217.90	1625.73	112.37	119.45
3	2281.98	2935.0677	985.87	95.79	100.73	879.17	1582.10	51.15	8035.69	363.10	185.011	221.47	1620.67	115.08	123.00
4	2462.19	3134.0840	1005.56	96.93	103.05	958.97	1638.84	52.13	8211.17	374.07	201.288	237.29	1744.05	116.55	125.61
5	2478.91	3230.4914	1000.92	94.41	96.23	982.59	1626.67	50.88	7590.21	371.95	201.061	234.77	1751.62	107.28	119.84

```
# Evaluate accuracy

test_loss, test_acc = model.evaluate(X_test_cs,  y_test_cs, verbose=2)
print('\nTest accuracy:', test_acc)
```

```
3/3 - 0s - loss: 1.8248 - accuracy: 0.4578

Test accuracy: 0.45783132314682007
```

```
# Evaluate accuracy

test_loss_rg, test_acc_rg = model_rg.evaluate(X_test_rg,  y_test_rg, verbose=2)
print('\nTest accuracy:', test_acc_rg)
```

```
3/3 - 0s - loss: 0.0537 - mse: 0.0048

Test accuracy: 0.004806651268154383
```

Based on the results obtained Regression is better for predicting the values but might cause more incorrection on up/down. Classification is better to predict the up/down scope

The VARMAX model can model much more complex relationships including past values of the variables and their stochastic terms.

5. Write a non-technical 1 paragraph email of your findings for senior management

We write this email to summarize our research findings into modeling techniques for the firm. Our focus was on VARMAX, a time series forecasting method, and Neural Networks(NN), a machine learning method for supervised and unsupervised learning. VARMA is a linear forecasting method of time series variables; they provide less parameterized representation of the linear data generation processes. Neural Networks (NN) started from the idea of an artificial neuron called a 'perceptron' by Frank Rosenblatt in the 1950s. Perceptrons receive multiple inputs and produce a single binary output. Neural networks expand on the idea of a single artificial neuron, by combining them into a network, neurons feeding their outputs into another layer of neurons. We can safely say that in time series forecasting, the VARMAX model has more limitations in application as compared to NN. The flexibility in NN has made it successful in

different fields, and this originates from the choices the analyst can make. One choice made by such analysts is using non-linear functions to capture non-linear relationships in the modeled data. Based on these, NN is used in future modeling instead of VARMAX.

Regards,
Group 02

References:

1. Walter Enders : Applied Econometric Time Series (4th ed)
2. James Ma Weiming : Mastering Python for Finance (2nd ed)
3. Stats Model. "Markov Switching Dynamic Regression Model". Accessed 2021-01-15. https://www.statsmodels.org/stable/examples/notebooks/generated/markov_regression.html
4. Nielsen, Michael. "Neural Networks and Deep Learning". Determination Press,2015. Accessed 2021-01-16. http://neuralnetworksanddeeplearning.com/chap1.html
5. Sheng-Hsun Hsu, JJ Po-An Hsieh, Ting-Chih Chih, Kuei-Chu Hsu. "A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression". Expert Systems with Applications vol 36, 2009