DATA ANALYSIS USING PYTHON ON ETFs DATASET

By :
Uzma Naeem

OUTLINE

- Understanding the Dataset
- Data Cleaning - Preprocessing
- Data Transformation
- Data Visualisation & Analysis
- Data Modelling
- Model Validation

UNDERSTANDING THE DATA SET

Title: Exchange Trade Funds (ETFs) & Mutual Funds Composition & Yield Metrics

Source: Obtained from Kaggle at this link.

Overview:

- This dataset is a comprehensive resource, enabling research, analysis, & comparison of (ETFs) by providing essential metrics like fund size, expense ratio, asset & sector allocation, geographic distribution, dividend yield, auditing company, legal structure, distribution strategies, & past performance.

- It plays a pivotal role in illuminating the ETF market dynamics, offering valuable insights into volatility and risk factors associated with government investments in private equities.

KEY ANALYTICAL QUESTIONS

Driving Question

- Develop optimal portfolios comprising the best ETFs for aggressive, moderate, & conservative risk profiles.

Highlighting the Questions Guiding Our Analysis

- Identify the top-performing ETFs based on year returns.
- Determine the ETFs domicile countries displaying the most substantial exposure.
- Explore the relationship between volatility and return in the context of ETFs.
- Examine the concentration levels per currency within the ETF market.

METHODOLOGY FOR ETFs ANALYSIS

Step1

Group the ETFs into 4 categories - A,B,C,D with 4,6,8 & 14% year return respectively.

Step2

TER < 0.05% over quote Good (1)

Step3

Currency : Hedged is 1 & Unhedged is 0

Step4

UCITSCompliance : Yes is 1, No is 0

Step 5

fundSizeMillions > 10M : Good(1)

DATA CLEANING - PREPROCESSING

Enhanced Dataset Quality

1. **Column Selection:** Kept relevant columns for analysis.
2. **Data Conversion:** Converted relevant column to appropriate data types.

```
In [5]: #Data conversion

        etfFiles['inceptionDate'] = pd.to_datetime(etfFiles['inceptionDate'], errors='coerce')
        etfFiles['quoteDate'] = pd.to_datetime(etfFiles['quoteDate'])
        etfFiles['ter'] = pd.to_numeric(etfFiles['ter'], errors='coerce')
        etfFiles['quote']=pd.to_numeric(etfFiles['quote'],errors='coerce')
        etfFiles['fundSizeMillions']=pd.to_numeric(etfFiles['fundSizeMillions'],errors='coerce')
```

3. **Data Filtering:** Filtered data based on the 'quoteDate' for March 2023, focusing on a specific timeframe.

```
In [6]: # Filter quoteDate of the ETFs - March 2023

        filtered_data = etfFiles[(etfFiles['quoteDate'] >= '2023-03-01') & (etfFiles['quoteDate'] < '2023-04-01')]
        filtered_data

        #filtered_data['quoteDate'].unique()
```

DATA CLEANING - PREPROCESSING

4. **Indexing:** Set the index values using the 'isin' column for better organization.

```
In [7]: # index values
        filtered_data.index = filtered_data['isin'].values
        #filtered_data.set_index('isin',inplace=True)
```

5. **Handling Missing Values:** Dropped rows where 'ytdReturnCUR' or 'yearVolatilityCUR' is null to ensure data completeness.

```
In [8]: # Delete values where the ytdReturnCUR  or yearVolatilityCUR is null
        filtered_data.dropna(subset=['ytdReturnCUR'],inplace=True)
        filtered_data.dropna(subset=['yearVolatilityCUR'],inplace=True)
        filtered_data
```

DATA TRANSFORMATION

6. **<u>Transformation Based on Yearly return:</u>** The data is given specific values based on the Year to date return variable.(A,B,C,D,E)

This is to be able to categorise based on returns by adding a new column with the group identities.

```
oups of ETF according to the ytdReturnCUR

tered_data['Group']='A'
tered_data.loc[(filtered_data['ytdReturnCUR']>=0.04)&(filtered_data['ytdReturnCUR']<0.06),'Group']='B'
tered_data.loc[(filtered_data['ytdReturnCUR']>=0.06)&(filtered_data['ytdReturnCUR']<0.08),'Group']='C'
tered_data.loc[(filtered_data['ytdReturnCUR']>=0.08)&(filtered_data['ytdReturnCUR']<0.14),'Group']='D'
tered_data.loc[(filtered_data['ytdReturnCUR']>0.14),'Group']='E'
```

Group	YTDR
A	<4%
B	4%-6%
C	6%-8%
D	8%-14%
E	>14%

DATA TRANSFORMATION

7. **<u>Data Quality Metrics:</u>** Introduced key metrics to assess data quality:

a. **Fee Comparison: 1 if fee is lesser than 0.5% of the quote price; else 0**

```
In [11]: # What is the equivalency of ter on the quote
         filtered_data['Fee_comparison']=(filtered_data['ter']<0.0005*filtered_data['quote']).astype(int)
```

b. **Currency Risk; 1 if currency is hedged; else 0 (un-hedged)**

```
In [12]: # column "Currency_Risk". If the colum " currencyRisk" is hedge is good=1 else 0
         filtered_data['Currency_Risk'] = filtered_data['currencyRisk'].apply(lambda x: 1 if x == 'Currency hedged' else 0)

         #filtered_data['Currency_Risk'].unique()
         #filtered_data['currencyRisk'].unique()
```

c. **Compliance Evaluation: 1 if ETF is compliant; else 0**

```
In [13]: #"Compliance" . If the column "UCITSCompliance" is yes then 1 else 0
         filtered_data['Compliance'] = filtered_data['UCITSCompliance'].apply(lambda x: 1 if x == 'Yes' else 0)
         #filtered_data['Compliance'].unique()
         #filtered_data['UCITSCompliance'].unique()
```

d. **Volume Assessment; 1 if ETF has volume greater than 10 million; else 0**

```
In [14]: #Volume" if the column "fundSizeMillions" is more than 10 Million then  good=1 else 0
         filtered_data['Volume'] = filtered_data['fundSizeMillions'].apply(lambda x:1 if x > 10 else 0)
         #filtered_data['fundSizeMillions'].info()
         #filtered_data['Volume'].unique()
```

DATA TRANSFORMATION

Risk Groups using K-means

- k-means is an unsupervised machine learning clustering technique to partition data into groups - clusters in a dataset.
- No object can be a member of more than one cluster, & every cluster must have at least one object.
- The groups are created based on mathematical distance between each data point.
- The goal is to minimize the sum of all distances between data points for each cluster.

- 3 clusters -> 3 Risk Groups
- Conservative, Moderate, Aggressive
- Clusters based on Year to date Volatility

Risk Group	# ETFs	Year Today Return		
		Min	Max	Mean
Conservative	999	0.001	0.183	0.126
Moderate	978	0.184	0.519	0.241
Aggressive	59	0.581	1.995	0.838

DATA VISUALIZATION

What is the relationship between volatility and return?

- The primary objective of this analysis is to examine how Volatility influences Year-to-Date Returns across different Risk Groups.

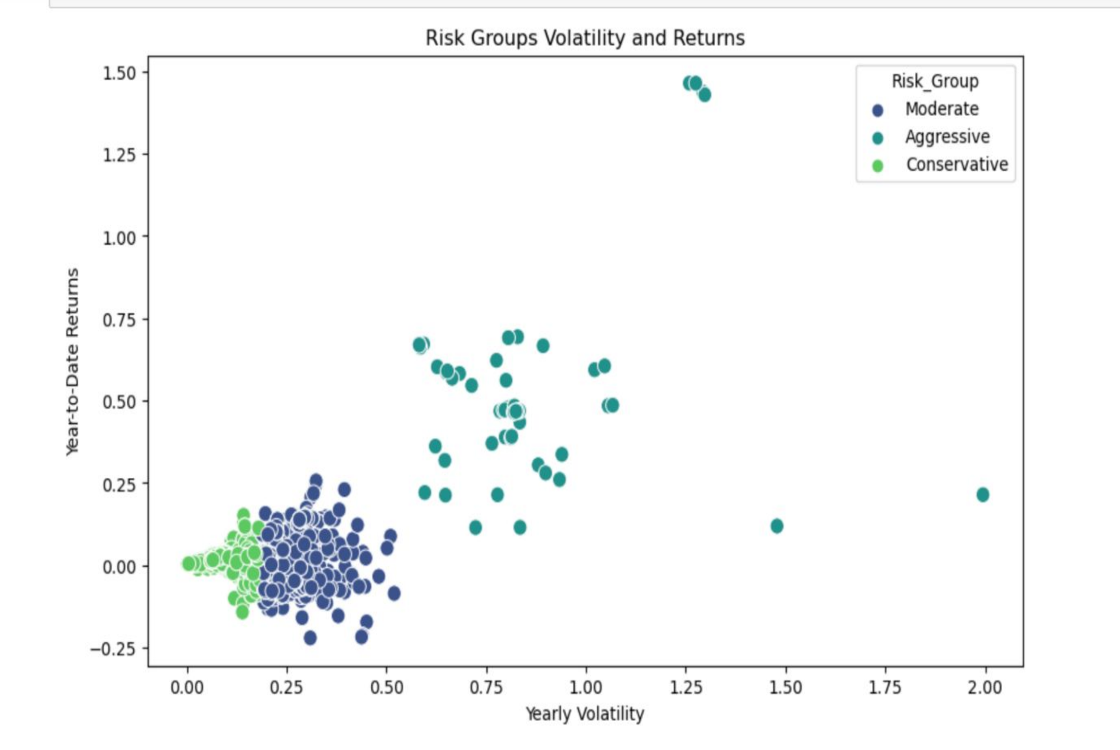
```
In [19]: #Data Visualisations

         #Scatter plot

         plt.figure(figsize=(10, 6))
         sns.scatterplot(x='yearVolatilityCUR', y='ytdReturnCUR', hue='Risk_Group', data=filtered_data, palette='viridis', s=
         plt.title('Risk Groups Volatility and Returns')
         plt.xlabel('Yearly Volatility')
         plt.ylabel('Year-to-Date Returns')
         plt.show()
```

DATA VISUALIZATION

- The graph presented is a scatter plot, visualizing data points based on the 'Yearly Volatility' (on the x-axis) and 'Year-to-Date Returns' (on the y-axis). Each data point is color-coded according to its respective Risk Group, providing a comprehensive view of the interplay between volatility, and returns.



DATA VISUALIZATION

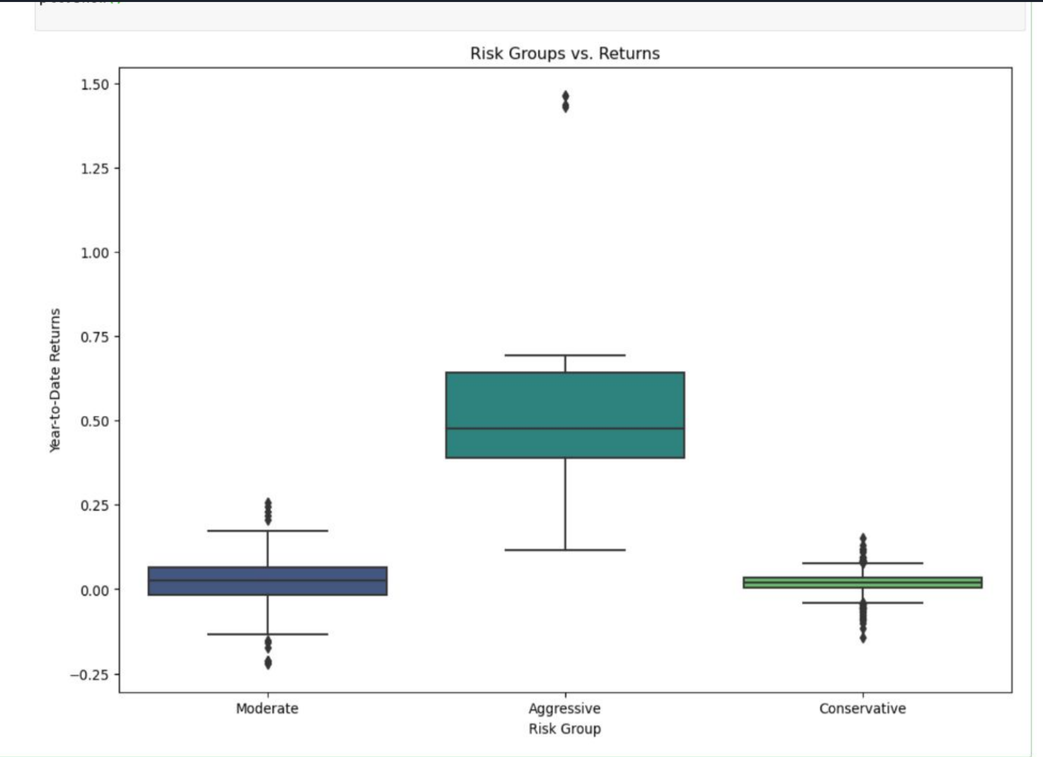
- The box plot encapsulates the distribution of returns, offering insights into central tendencies, variability, and potential outliers.

```
#Boxplot

plt.figure(figsize=(12, 8))
sns.boxplot(x='Risk_Group', y='ytdReturnCUR', data=filtered_data, palette='viridis')
plt.title('Risk Groups vs. Returns')
plt.xlabel('Risk Group')
plt.ylabel('Year-to-Date Returns')
plt.show()
```

DATA VISUALIZATION

- The line within each box represents the median YTD Returns for the respective Risk Group. The box itself spans the Interquartile Range, indicating the middle 50% of Year-to-Date Returns.
- The length of the box reflects the variability within each Risk Group, offering a glimpse into the consistency of performance. The whiskers extend from the box to the minimum and maximum values within a certain range, typically 1.5 times the IQR(interquartile range). Outliers beyond the whiskers are represented as individual data points.



DATA VISUALIZATION

What are the top 5 ETFs with best returns ?

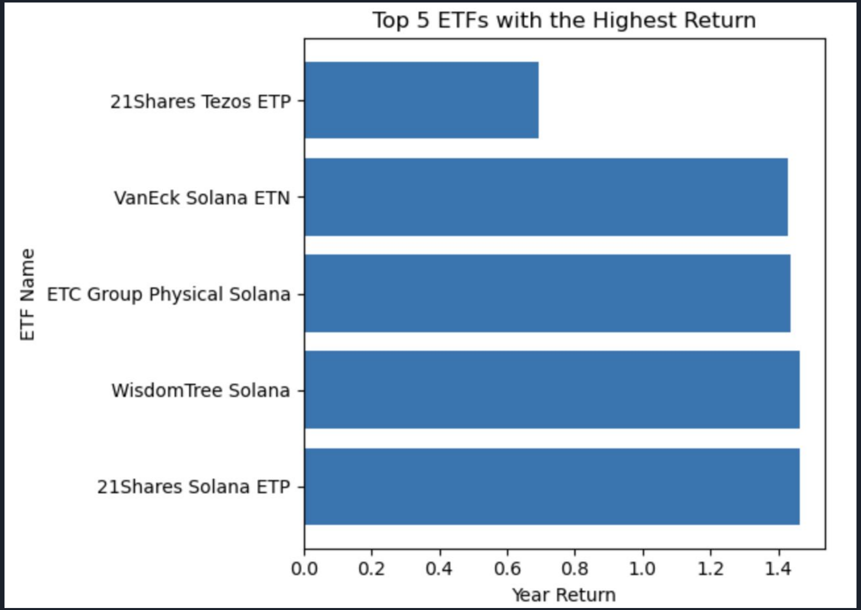
- Overview of the best ETFs in accordance with the highest return.

```
top_5_etfs = filtered_data.sort_values(by='ytdReturnCUR', ascending=False).head(5)

# Create a clustered bar chart
plt.barh(top_5_etfs['name'], top_5_etfs['ytdReturnCUR'],)

# Adding labels and title
plt.title('Top 5 ETFs with the Highest Return')
plt.xlabel('Year Return')
plt.ylabel('ETF Name')

# Save the plot as an image
plt.tight_layout()
plt.savefig('Top_5_ETFs_Highest_Return.png')
```



DATA VISUALIZATION

What is the concentration of ETFs per Domicile Country?

- Overview of the distribution of etfs in different countries, emphasizing which countries are more prevalent in hosting ETFs.
- Understand weather market is diverse, spread over different countries or if its specific to few countries.

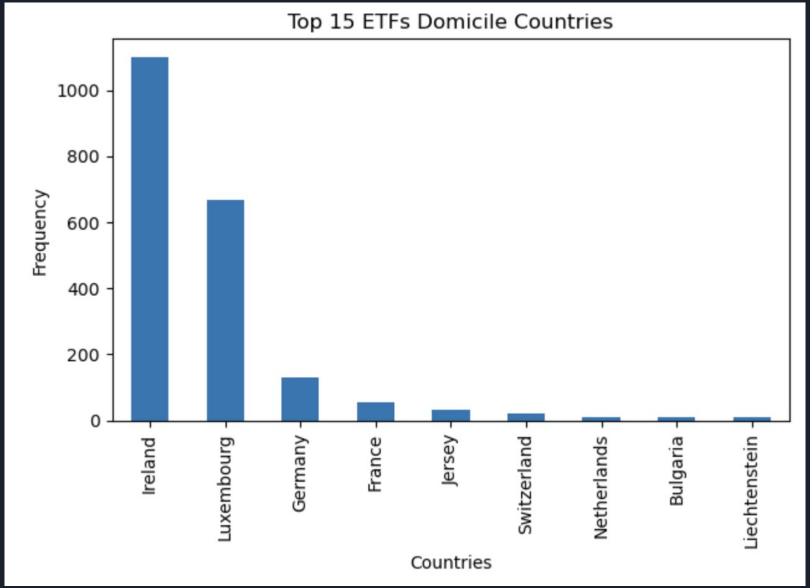
```python
# Top Domicile Countries

# We are counting how many time each country is appearing
filtered_data['domicileCountry'].value_counts().head(15).sort_values(ascending=False).plot(kind='bar')

# Adding labels and title
plt.title('Top 15 ETFs Domicile Countries')
plt.xlabel('Countries')
plt.ylabel('Frequency')

# Save the plot as an image
plt.tight_layout()
plt.savefig('Top_15_ETFs_Countries.png')

plt.show()
```



DATA VISUALIZATION

What is the concentration of ETFs per currency ?

- Overview of the distribution of ETFs across different currencies, emphasizing USD as the most used currency.

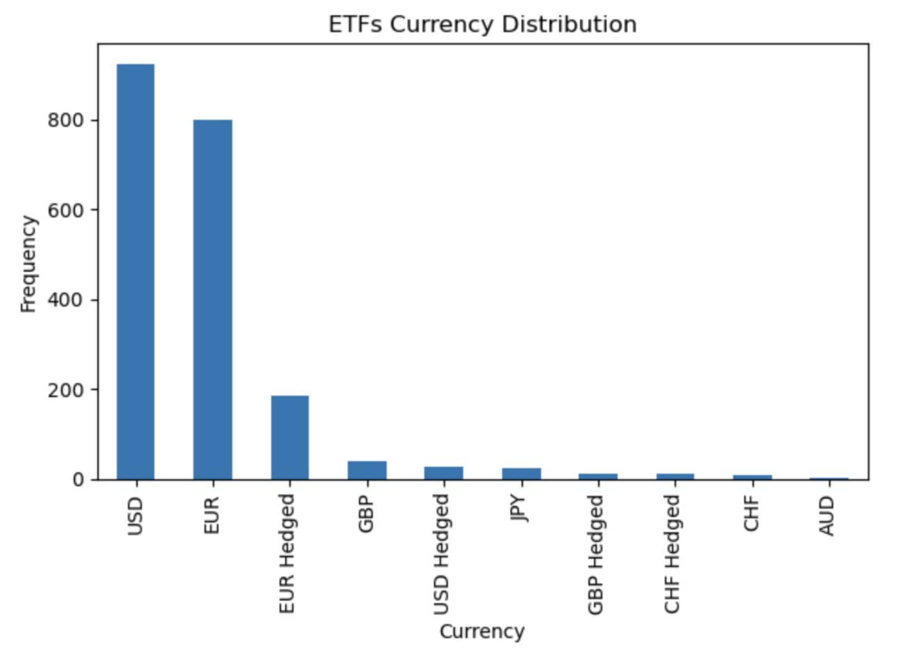
```
# We are counting how many time each currency  is appearing

filtered_data['fundCurrency'].value_counts().head(10).sort_values(ascending=False).plot(kind='bar')

plt.title('ETFs Currency Distribution')
plt.xlabel('Currency')
plt.ylabel('Frequency')

# Save the plot as an image
plt.tight_layout()
plt.savefig('ETFs_Currency_Distribution.png')

plt.show()
```



DATA ANALYSIS: THE BEST ETFs

```
# best ETF that qualify the logic parameters in volume, currency among others

best_etf = filtered_data[(filtered_data['Group']!='A') &
(filtered_data['Volume']==1) & (filtered_data['Compliance']==1)&
(filtered_data['Currency_Risk']==1)& (filtered_data['Fee_comparison']==1)]
best_etf
```

There are 30 ETFs that satisfy the methodology previously explain, However, there are not ETFs that accomplish the filters for aggressive risk profile.

DATA ANALYSIS: THE BEST ETFs

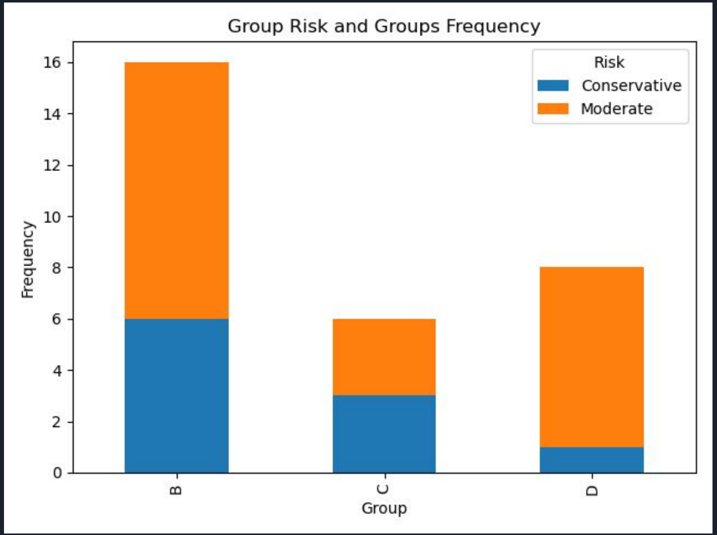
```
grouped_data = best_etf.groupby(['Group', 'Risk_Group']).size().unstack(fill_value=0)

# Plotting the grouped bar chart
grouped_data.plot(kind='bar', stacked=True)

# Adding labels and title
plt.title('Group Risk and Groups Frequency')
plt.xlabel('Group')
plt.ylabel('Frequency')
plt.legend(title='Risk')


# Save the plot as an image
plt.tight_layout()
plt.savefig('Group_Risk_Groups_Frequency.png')

# Display the grouped bar chart
plt.show()
```



20 of the 30 best ETFs have a moderate risk level. In addition the group B (year return 4%-6%) includes 16 out of the best ETFS.

DATA MODELLING

Simple Regression

What is the impact of year volatility in year return for the best ETFs (n=30)?

Final Equation

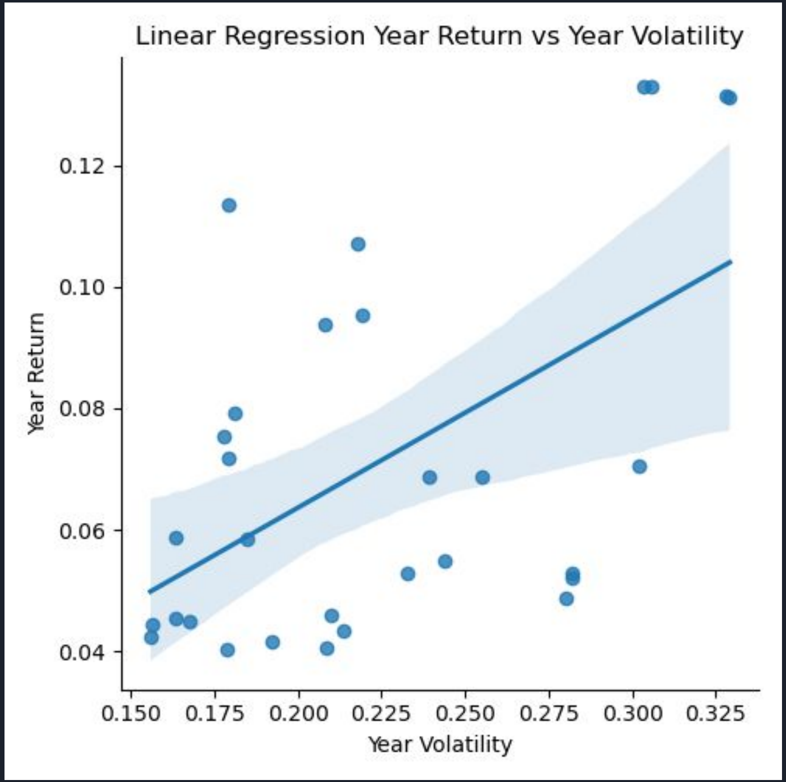
$$Y(YearReturn) = 0.0011 + 0.3126\ YearVolatility$$

F-statistic: 11.76 Prob (F-statistic): 0.00189

At 5% alpha the model is statistically significant, the slope of the predictor is not zero.

R-squared: 0.296

29.6% of YearReturn variance is explained by YearVolatility. For now the model is not a robust return predictor, there are more variables that could be add to get a stronger model.



DATA MODELLING :

```
In [28]: # The summary() method allows to display the results


         print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           ytdReturnCUR   R-squared:                       0.296
Model:                            OLS   Adj. R-squared:                  0.271
Method:                 Least Squares   F-statistic:                     11.76
Date:                Tue, 05 Dec 2023   Prob (F-statistic):            0.00189
Time:                        01:00:47   Log-Likelihood:                 67.161
No. Observations:                  30   AIC:                            -130.3
Df Residuals:                      28   BIC:                            -127.5
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
Intercept             0.0011      0.021      0.052      0.959      -0.042       0.044
yearVolatilityCUR     0.3126      0.091      3.429      0.002       0.126       0.499
==============================================================================
Omnibus:                        3.129   Durbin-Watson:                   0.910
Prob(Omnibus):                  0.209   Jarque-Bera (JB):                1.805
```

DATA MODELLING

Interpretation of Final Model :

Y(YearReturn) = 0.0011 + 0.3126 YearVolatility

The equation indicates the relation between Year Volatility and Year Return.For each unit increase in year volatility the year return is expected to increase 0.3126 times.

Intercept = 0.0011

It represents the minimum expected year return when the volatility is zero. However, in practical uses the intercept is not always relevant.

In essence, this regression equation provides a predictive model, suggesting that higher YearVolatility is associated with higher expected Year Return in a linear fashion.

CONCLUSION

- We have created a model find best ETFs which shows return increases about 30% for each unit increase in Volatility basis the 2 risk groups - Conservative and Moderate. None of the best ETFs fall in the Aggressive risk group per the current analysis.

- Top 5 ETFs with best returns are 21Shares Tezos ETP,VanEck Solana ETN, ETC Group Physical Solana, WisdomTree Solana, 21Shares Solana ETP.

- The top domicile country is Ireland and the top currency is USD.

- Year Volatility is directly proportional to higher expected Year Return. It would be important to add more predictors to make a robust model to predict year return.