Advance Analytics and Machine Learning Project

Time Series Volatility Model for SOFR

Prepared by:

Shrey Kshatriya, Hariom Mehta,

Kunwar Vaibhav Singh, Kavit Shah

Table of Contents:

Problem Understanding

Identify:

It is important to know what is SOFR, LIBOR and why they are important to the banks and government before understanding the problem.

SOFR:

The secured overnight financing rate, or SOFR, is an influential interest rate that banks use to price U.S. dollar-denominated derivatives and loans. The daily secured overnight financing rate (SOFR) is based on transactions in the Treasury repurchase market, where investors offer banks overnight loans backed by their bond assets.

Benchmark rates such as the secured overnight financing rate (SOFR) are essential in the trading of derivatives—particularly

interest-rate swaps, which corporations and other parties use to manage interest-rate risk and to speculate on changes in borrowing costs.

Interest-rate swaps are agreements in which the parties exchange fixed-rate interest payments for floating-rate interest payments. In a "vanilla" swap, one party agrees to pay a fixed interest rate, and, in exchange, the receiving party agrees to pay a floating interest rate based on the secured overnight financing rate (SOFR)—the rate may be higher or lower than SOFR, based on the party's credit rating and interest-rate conditions. [1]

LIBOR:

LIBOR, or London Interbank Offered Rate, was originally based on interbank lending transactions, but due to changes in how banks fund themselves the underlying bank-to-bank lending market has shrunk significantly. With few, if any, actual transactions for banks to base their quotes on, LIBOR submissions became dependent on the panel banks' own judgement. This led to an increase in fraud and manipulation (LIBOR rigging scandals) for an index that is tied to almost $200 trillion in financial products around the world.

Unlike the LIBOR, there's extensive trading in the Treasury repo market—roughly 1,500 times that of interbank loans as of 2018—theoretically making it a more accurate indicator of borrowing costs.

Moreover, the secured overnight financing rate (SOFR) is based on data from observable transactions rather than on estimated borrowing rates, as is sometimes the case with LIBOR.[5]

Define:

To effectively build a time-series forecasting model for SOFR data using Machine Learning and Data Mining solutions. Developing a front-end for the developed model so that investors can directly use the application for prediction of SOFR rate.

Motivation:

As compared to LIBOR, SOFR is a relatively newer system altogether. With SOFR being first introduced in 2018, it is crucial to identify the highs and lows of the market in the coming near future. Forecasting the rate of SOFR can help banks and investors to identify when would be the best time to invest/lend a loan and what rate to ask for it.

Performing Data Mining on this problem will help us to scrape the past data and forecast the future SOFR rates. This will help firms to consider the properties of the SOFR rate into pricing or simulating the path of the products referencing SOFR rate, especially in the transition since the term-structure is not yet available and the jumps in the overnight structure might have significant implications on the price.

Data Understanding

The SOFR data is available on the Federal Reserve Bank of New York. The SOFR is calculated as a volume-weighted median of transaction-level tri-party repo data collected from the Bank of New York Mellon as well as GCF Repo transaction data and data on bilateral Treasury repo transactions cleared through FICC's DVP service, which are obtained from DTCC Solutions LLC, an affiliate of the Depository Trust & Clearing Corporation. Each business day, the New York Fed publishes the SOFR on the New York Fed website at approximately 8:00 a.m. ET

The dataset includes time-series data of overnight USD SOFR rates starting from its first publish date i.e 3rd April 2018 to 18th November 2020. It contains observations of rates on a day-to-day basis over the sample period.

	RATE\n(PERCENT)	1ST\n(PERCENT)	25TH\n(PERCENT)	75TH\n(PERCENT)	99TH\n(PERCENT)
count	660.000000	659.000000	659.000000	659.000000	659.000000
mean	1.531939	1.463763	1.508012	1.584613	1.677815
std	0.925336	0.893740	0.921982	0.943244	1.004444
min	0.010000	-0.030000	0.010000	0.010000	0.120000
25%	0.110000	0.070000	0.080000	0.150000	0.215000
50%	1.855000	1.800000	1.820000	1.910000	1.980000
75%	2.262500	2.160000	2.240000	2.320000	2.410000
max	5.250000	2.700000	5.000000	5.850000	9.000000

The primary SOFR rate is calculated as a volume-weighted median and is the rate generated with 50 percentile of transaction volume. The 1st, 25th, 75th, and 99th percentiles for each rate are also calculated using the same volume-weighted methodology and rounded to the nearest basis point.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 660 entries, 0 to 659
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   DATE                  660 non-null    object
 1    BENCHMARK NAME       660 non-null    object
 2   RATE
(PERCENT)        660 non-null    float64
 3   1ST
(PERCENT)         659 non-null    float64
 4   25TH
(PERCENT)        659 non-null    float64
 5   75TH
(PERCENT)        659 non-null    float64
 6   99TH
(PERCENT)        659 non-null    float64
 7   VOLUME (US$BILLIONS)  660 non-null    object
dtypes: float64(5), object(3)
memory usage: 41.4+ KB
```

Data Preparation

The data was taken from the site SOFR Averages and Index Data.

Since the data was raw, we had to check for any null values in the data since the SOFR rates are not generated for holidays and weekends. We found out that the data contained null values for holidays and weekends and the null values were taken care of during the later part of pre-processing.

The second step was to convert the DATE column to the datetime type for plotting time series graph for SOFR with respect to the rate. Percentiles columns were dropped as we were not taking it under consideration. They were dropped as we know that while forecasting, the features variables should be independent of the target variables. And since the percentiles were derived from the Rate itself, they were dropped.

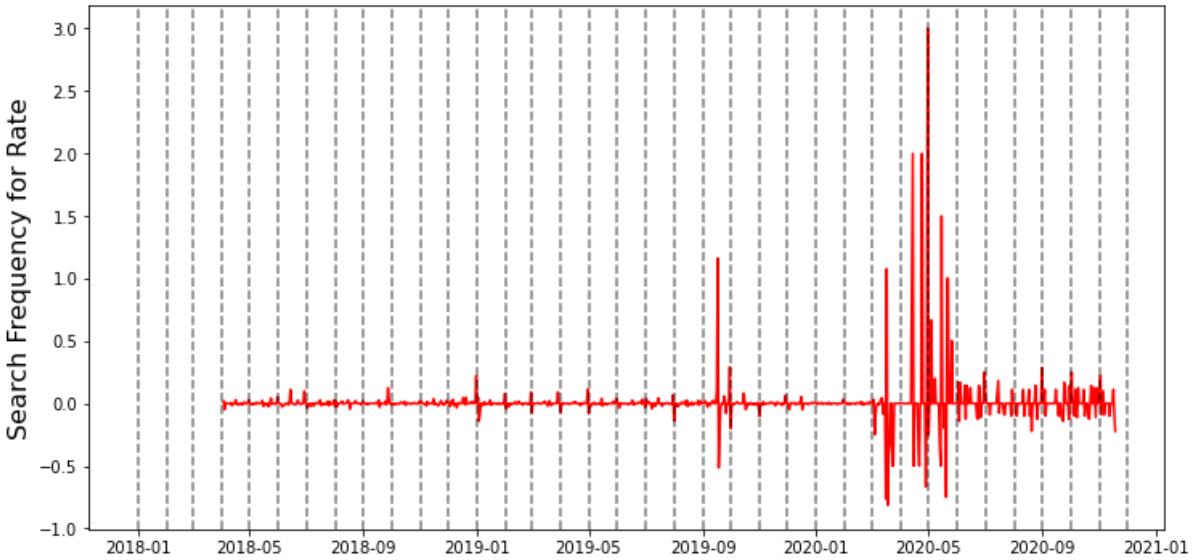
Only the columns (Date, Rate and Volume) were taken into consideration.

As seen before, there were many null values because of public holidays and weekends so those data for those days are not available therefore the missing values were imputed by a forward-filling algorithm where the values from the previous row is considered. This was done as the rate for the holidays remains the same and it does not fluctuate during those days. Hence to maintain consistency and for better visualization, we perform imputations.

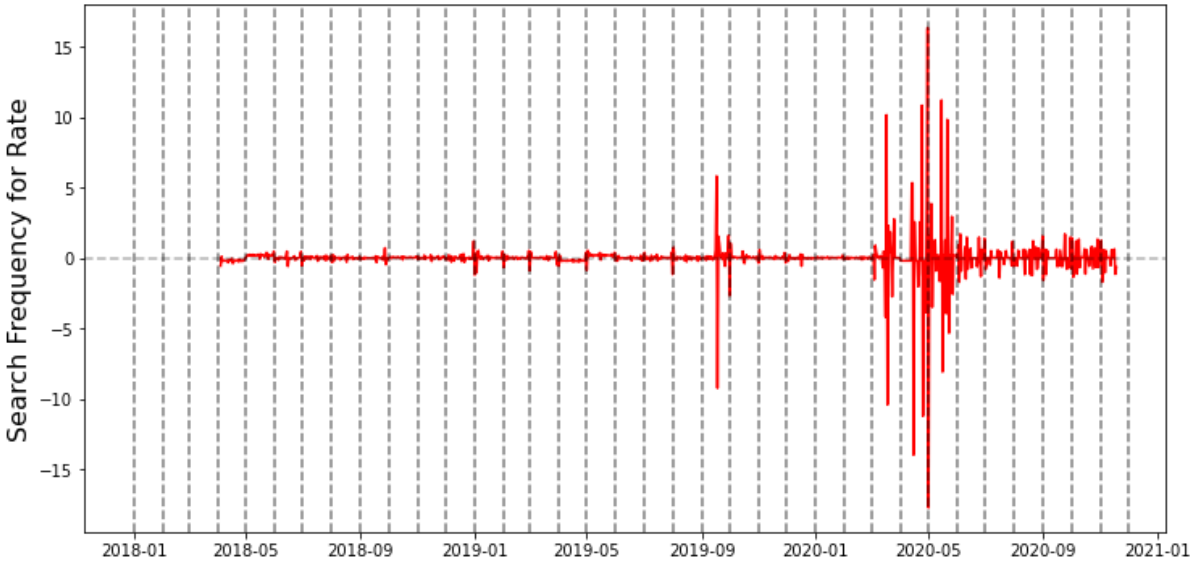
While studying the data, we found out that the data had trends and seasonality. Hence, to remove the trends and seasonality, we normalized the data. To normalize the data, we computed the first difference to remove the trends and then removed seasonality.

Below we can see our data before and after Normalizing.

BEFORE



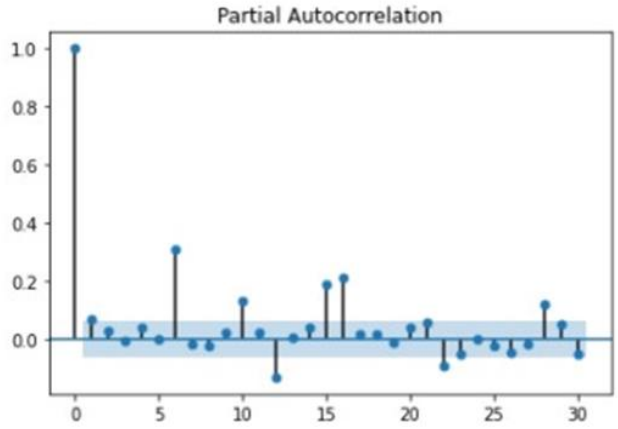
AFTER



For better statistical evaluation we converted the SOFR into Log-returns. ACF and PACF plots were made for better understanding of the pattern and to determine our model.

Modelling

The following steps were performed for Data Modelling:

- After Data preparation we performed these following tests on data to understand the volatility of SOFR rate.
    - ARCH Test
    - PACF and ACF
    - 

- Based on evaluation of the above plots we decided to implement these following models
    - GARCH(1,1) model
    - Skewness and Kurtosis Test
    - GARCH (skewness) distribution
    - GJR-GARCH
- After looking at PACF plots of data, our 1st lag is just above the significance level and we decide to consider this lag in our model.
- After doing ARCH test and pacf plot Garch(1,1) is suitable for our model
- Looking at coefficients of our model all the coefficient has significant p value and our fitted garch model has good AIC value.

- Moreover, we also implemented the GJR- GARCH (1,1) model for better AIC value but we alpha and gamma coefficients were insignificant.
- Finally after trying various models of garch we found GARCH(1,1) is the most suitable for our model, It captures the volatility spikes greatly, but we found it is unable to capture the negative spikes, most of the negative spikes were in early 2020 , unprecedented times and situation its highly impossible to capture the negative spikes but our model is doing very good when we look closely on actual and predicted rolling forecast of volatility , after positive volatility spike when it drops rapidly we captures the behaviour that volatility is going in negative direction but its not capturing the 100% accurate value. We believe if our model is capturing the negative spike behaviour of volatility then it is really performing well.
- One more reason why our model did not predict the negative spikes is, insufficient data of historical SOFR rate. Since the Federal Bank of Newyork started publishing the data from 2018 we do not have the data before that. But we believe our model will perform even more accurately with good historical data.
- Though, positive spikes are very crucial to determine the loan rates,mortgage rate and other interest rates.
- SOFR rate is highly volatile in recent covid situations, since our model is very good at predicting volatility it will help the business to understand how yesterday's market transaction will affect tomorrow's interest rate. Currently, many big firms are starting to switch from LIBOR to SOFR because SOFR is highly based on USD denominated benchmark interest rate. According to the Federal Reserve, LIBOR will be replaced by SOFR by June 2023 and announced that banks are instructed to stop writing contracts using LIBOR by the end of 2021.

Evaluation:

In financial time series, modelling real data needs proper attention and suitable model selection is required to better understand the structure of the candidate data which ultimately helps in better forecasting [6]. Firstly, to predict the time series SOFR data, we build an ARIMA (p,d,q) model which fits the general time-series pattern for the data. Moving forward, we try to capture the volatility of the data in our model. Since GARCH(p,q) modelling better captures the momentum of conditional variance in time-series data when compared to an ARCH(q) model, we decided to model our data with different variants of the GARCH(p,q) that were discussed over the sessions of the course.

Serial Auto-correlations & ADF Test:

Based on the Ljung-Box statistics for the time-series data, we check the presence of sample auto-correlations in the dataset. To further understand the influence of lagged values on current SOFR data, we perform the Augmented Dickey-Fuller Test and compare the ACF and PACF plots for the time-series data.

```
In [15]:
#Ljung box test
res = sm.tsa.ARMA(rate, (1,1)).fit(disp=-1)
#sm.stats.acorr_ljungbox(res.resid, lags=[10])
pval = sm.stats.acorr_ljungbox(res.resid, lags=[5])
# pvalue is high, reject null hypo, there is no serial corelation

print("p-value for Ljung-Box Test with 5 Lags is: ", pval[1])
```

```
p-value for Ljung-Box Test with 5 Lags is:  [0.10545982]
```

Since the p-value for the L-jung Box Test is greater than 0.05, we can comfortably reject the null hypothesis and infer that the time series SOFR data has serial auto-correlations.

```
stats = sm.stats.diagnostic.het_arch(res.resid)
#Lagrange multiplier test statistic, p-value for Lagrange multiplier test

#fstatistic for F test, alternative version of the same test based on F test for the parameter restriction

#p value is very low, reject nul hypo, there is arch effects

print("p-value for unit-root ADF Test:" , stats[1])
```

```
p-value for unit-root ADF Test: 8.974739974486243e-30
```

As we can see, the p-value for the unit root ADF-Test is less than 0.05, therefore, we cannot reject the null hypothesis. This confirms the presence of unit root in the time-series SOFR data and calls for a possible need of differencing of the series. To quantify the order of the differencing required, we plot the Partial Autocorrelation Function(PACF) plot for the time-series. Based on the PACF plot, we can decide whether to carry out a first order differencing of the time-series data which was cross-checked by comparing the AIC values for all the models with d=0 and d=1 (p,q = 0 to 2) .

To evaluate the performance of the different models employed, which were GARCH (1,1), GJR-GARCH (1,1) and GARCH (1,1) with student skewed t-distribution, we compare the Akaike-Information Criteria (AIC) values for each model. Furthermore, to even further analyze the models, we compared the p-values (significant if <0.05 assuming a 95% confidence interval) for the mean model coefficients for each model.

a. GARCH (1,1) model :

Constant Mean - GARCH Model Results

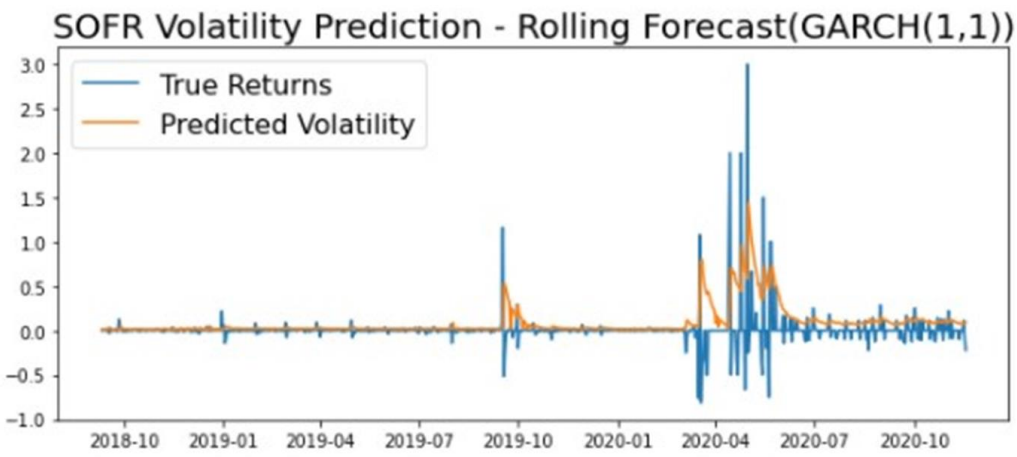
Dep. Variable:	RATE (PERCENT)	**R-squared:**	-0.003
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.003
Vol Model:	GARCH	**Log-Likelihood:**	1106.15
Distribution:	Normal	**AIC:**	-2204.30
Method:	Maximum Likelihood	**BIC:**	-2184.83
		No. Observations:	961
Date:	Sun, Dec 13 2020	**Df Residuals:**	957
Time:	14:15:57	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	-1.1573e-03	3.752e-03	-0.308	0.758	[-8.512e-03,6.197e-03]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	6.5798e-04	3.102e-04	2.121	3.389e-02	[5.008e-05,1.266e-03]
alpha[1]	0.2000	0.104	1.929	5.369e-02	[-3.174e-03, 0.403]
beta[1]	0.7800	5.552e-02	14.050	7.692e-45	[0.671, 0.889]



b. GJR-GARCH (1,1) :

Constant Mean - GJR-GARCH Model Results

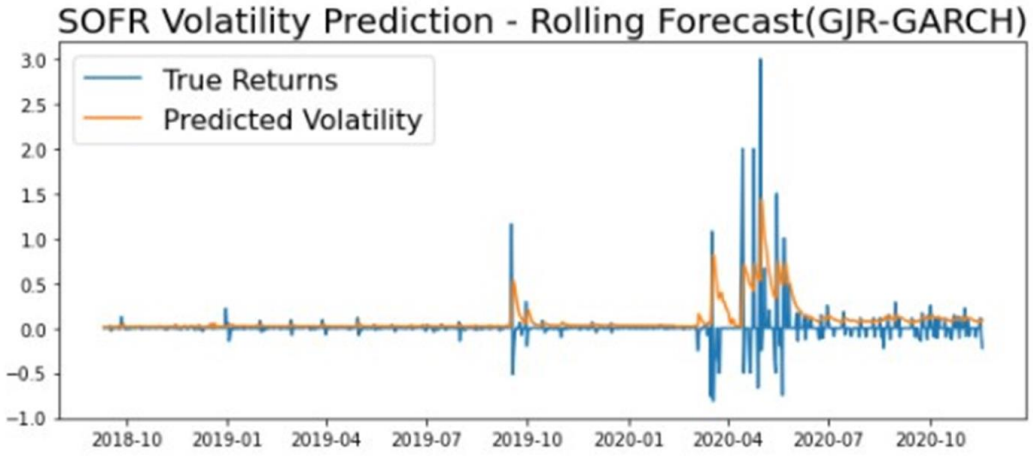
Dep. Variable:	RATE (PERCENT)	**R-squared:**	-0.003
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.003
Vol Model:	GJR-GARCH	**Log-Likelihood:**	1104.65
Distribution:	Normal	**AIC:**	-2199.29
Method:	Maximum Likelihood	**BIC:**	-2174.95
		No. Observations:	961
Date:	Sun, Dec 06 2020	**Df Residuals:**	956
Time:	23:04:49	**Df Model:**	5

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	-1.5688e-03	2.618e-03	-0.599	0.549	[-6.700e-03,3.563e-03]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	6.5815e-04	3.652e-04	1.802	7.153e-02	[-5.765e-05,1.374e-03]
alpha[1]	0.2000	0.198	1.011	0.312	[-0.188, 0.587]
gamma[1]	9.9988e-03	0.250	3.996e-02	0.968	[-0.480, 0.500]
beta[1]	0.7750	5.321e-02	14.564	4.740e-48	[0.671, 0.879]



c. GARCH (1,1) with student t-distribution :

Constant Mean - GARCH Model Results

Dep. Variable:	RATE (PERCENT)	**R-squared:**	-0.034
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.034
Vol Model:	GARCH	**Log-Likelihood:**	441.881
Distribution:	Standardized Skew Student's t	**AIC:**	-871.762
Method:	Maximum Likelihood	**BIC:**	-842.554
		No. Observations:	961
Date:	Sun, Dec 06 2020	**Df Residuals:**	955
Time:	22:44:59	**Df Model:**	6

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.0415	3.017e-02	1.375	0.169	[-1.764e-02, 0.101]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0344	2.452e-03	14.047	8.061e-45	[2.964e-02,3.925e-02]
alpha[1]	0.3229	0.235	1.373	0.170	[-0.138, 0.784]
beta[1]	0.1845	0.106	1.741	8.162e-02	[-2.316e-02, 0.392]

Distribution

	coef	std err	t	P>|t|	95.0% Conf. Int.
nu	130.5636	9.863	13.237	5.337e-40	[1.112e+02,1.499e+02]
lambda	0.1646	0.128	1.283	0.199	[-8.682e-02, 0.416]

As can be seen, among the AIC values for all the three models [GARCH(1,1): -2204 , GJR-GARCH(1,1): -2199 and GARCH(1,1) with Student-t : -871], the lowest value is for GARCH (1,1). Furthermore, the p-values for the volatility model features are also minimum for the GARCH(1,1) among that for all the models. Therefore, we decide to recommend and use GARCH(1,1) as the optimum model for capturing the volatility of time-series SOFR data.

Deployment

We decided to build a front-end application / local host for our SOFR volatility prediction model. For this purpose, we use streamlit library within python to develop an interface API which gives the volatility prediction for a particular day for the SOFR data. To use this information/app, a firm could effectively predict and store the volatility of the SOFR dataset for their required number of days in the future and couple this information with an ARIMA model prediction for the SOFR data to predict the SOFR data as per their required forecast duration.

The prediction results provided for volatility of SOFR for a particular date are solely based on publicly available SOFR data and are completely safe to use for business purposes.

The risks associated with our proposed plan is that since SOFR is a relatively new rate, the available historical data for SOFR is relatively less when compared to its counterparts such as LIBOR. In addition to this, to capture the negative volatility shocks further better, we recommend building the same model but with a markov-chain machine learning algorithm.

References

1. https://www.investopedia.com/secured-overnight-financing-rate-sofr-4683954#:~:text=The%20secured%20overnight%20financing%20rate%2C%20or%20SOFR%2C%20is%20an%20influential,dollar%2Ddenominated%20derivatives%20and%20loans.&text=Interest%2Drate%20swaps%20are%20agreements,for%20floating%2Drate%20interest%20payments.
2. https://apps.newyorkfed.org/markets/autorates/SOFR
3. https://www.morganstanley.com/ideas/libor-its-end-transition-to-sofr
4. https://www.investopedia.com/ask/answers/052715/how-big-derivatives-market.asp
5. https://www.pensford.com/resources/libor-vs-sofr/
6. https://grapestat.se/sites/default/files/GRAPES/Workshop2010/FarrukhJaved.pdf