University of Tartu
Faculty of Science and Technology
Institute of Mathematics and Statistics

Anna-Helena Salurand

Validation and development of risk metrics for the currency portfolio of a global FinTech company

Actuarial and Financial Engineering

Master's Thesis (30 ECTS)

Supervisors: Assoc. Prof. Raul Kangro,
PhD Egor Kraev

Tartu 2021

VALIDATION AND DEVELOPMENT OF RISK METRICS FOR THE CURRENCY PORTFOLIO OF A GLOBAL FINTECH COMPANY

Master's thesis

Anna-Helena Salurand

Abstract. The purpose of this thesis is to validate the Value at Risk (VaR) model of the currency portfolio of the global FinTech company Wise, develop the risk metrics further by introducing the Conditional Value at Risk (CVaR) and estimating VaR and CVaR multipliers for different time horizons using empirical percentile and Bayesian Model Averaging methods. The first two chapters give an overview of the foreign exchange market and some of the most common risk metrics. The third chapter presents the data, methods used, and results. The results show that the current VaR models' assumptions are appropriate to calculate daily 95% VaR and CVaR, but would underestimate 95% CVaR for most of the shorter time horizons. Furthermore, the two applied methods yield similar results, both suitable for computing 95% VaR and CVaR for most time horizons between 1 to 24 hours.

CERCS research specialisation: P160 Statistics, operations research, programming, financial and actuarial mathematics.
Key Words: foreign exchange market, financial risks, risk theory

GLOBAALSE FINTECH ETTEVÕTTE VALUUTAPORTFELLI RISKIMÕÕDIKUTE VALIDEERIMINE JA EDASIARENDUS

Magistritöö

Anna-Helena Salurand

Lühikokkuvõte. Käesoleva magistritöö eesmärgiks on valideerida globaalse *FinTech* ettevõtte Wise valuutaportfelli *Value at Risk* (VaR) mudelit ja arendada riskimõõdikuid lisades *Conditional Value at Risk* (CVaR) mõõdiku ning hinnates VaR ja CVaR kordajate väärtuseid erinevatele ajaperioodidele empiirilise protsentiili ja Bayesi mudelite keskmistamise meetodite abil. Esimesed kaks peatükki annavad ülevaate valuutaturgudest ja levinumatest riskimõõdikutest. Kolmandas peatükis esitletakse kasutatud andmeid, meetodeid ning tulemusi. Tulemused näitavad, et praeguse VaR mudeli eeldused on sobivad päevase 95% VaR ja CVaR arvutamiseks, kuid alahindaksid enamiku lühemate ajahorisontide puhul 95% CVaR'i. Kaks töös rakendatud meetodit annavad sarnaseid tulemusi ning mõlemad sobivad 95% VaR ja CVaR hindamiseks enamiku ajaperioodide jaoks vahemikus 1 kuni 24 tundi.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.
Märksõnad: valuutaturg, finantsriskid, riskiteooria

Contents

Introduction

The foreign exchange market is the largest and most liquid financial market, forming the base for the rest of the financial structure and trading. Among other reasons, a diverse range of market participants engage in forex transactions with the purpose of facilitating global trade, earning risk-based return or hedging currency risk. This thesis focuses on the latter in the context of the global FinTech company Wise.

In order to form hedging decisions for a currency portfolio, risk is usually quantified by a suitable risk metric. Two of the most widespread ways of quantifying market risk are Value at Risk (VaR) and Conditional Value at Risk (CVaR). A model for the former is also in place at Wise. Both of these metrics describe risk in monetary units — FX gain or loss in case of a currency portfolio, making it easy to comprehend and communicate.

The most complex part of computing VaR and CVaR is determining the distribution of the FX gain or loss variable. The current VaR model assumes that FX gain or loss variable is normally distributed with a zero mean, and estimates the standard deviation of the portfolio based on recent history. In this thesis, the portfolio volatility estimate is taken as given. The ratio of the realised risk (FX gain or loss) to the estimated risk (portfolio standard deviation estimate) is first computed based on historical data. This ratio can be used to validate the normality assumption of the current model, but also compute estimates for the so called VaR and CVaR multipliers that, in combination with the portfolio volatility value, can be used to calculate VaR and CVaR values.

The main objective of this thesis is to validate the current Value at Risk model used in the company with historical data, and propose and test confidence interval estimates for the 95% VaR multipliers with two different methods: empirical percentile method and Bayesian Model Averaging method. Furthermore, the aim of this thesis is also to develop the current risk model further to include a metric to describe the potential losses when VaR is exceeded — the CVaR. The goal is to propose and test confidence interval estimates for the 95% CVaR multipliers with the two aforementioned methods. In addition, this thesis also aims to conduct the described validations, estimations and testing of the estimates for time horizons of different length, varying from 1 to 24 hours, to be able to estimate the risk metrics for different time horizons and analyse whether the choice of the time horizon has an impact on the VaR and CVaR multiplier values.

This thesis consists of three chapters. The first chapter gives an overview of the foreign exchange market, its history, participants, main transactions and instruments and the costs and returns involved. The second chapter defines risk and introduces different ways of measuring market risk: volatility, expected loss and downside semi-deviation, VaR and CVaR. The third chapter of the thesis outlines the case of Wise and its VaR model, describes the data used in the empirical part, gives an overview of the two methods used in this thesis, presents the results and their analysis. Data used for computations was obtained from Wise. Computations and visualisations were performed using Python programming language.

The author would like to thank the supervisors of this thesis, Raul Kangro and Egor Kraev for their advice, feedback and support. The author is also grateful to Wise for allowing to use their data for conducting the research.

1 Foreign exchange market

The **foreign exchange market** (also forex, FX or currency market) is the institution facilitating the trading of one currency in exchange for another. While the foreign exchange market initially emerged to enable merchants settle foreign trades, risk management (hedging), arbitrage and speculation are nowadays equally common use cases. As one of the oldest forms of financial markets, it constitutes a base for the rest of the financial structure and trading, providing international liquidity. The foreign exchange market is also the largest and most liquid financial market where daily turnover reaches trillions of dollars. This chapter gives a brief overview of the history of the foreign exchange market and explains its functioning: participants, instruments and mechanics. (Hudson, 2013)

1.1 Brief history

Foreign exchange trading is thought to date back centuries, to the Babylonian period. The oldest method of exchange, dating back to 6000 BC, is the barter system, where goods were exchanged in return for other goods. Over time, certain goods, such as salt and spices, became a popular subject of exchange. (Bradfield, 2018)

In the 6th century BC, first gold coins were produced, forging its way to become the primary medium of exchange. Gold coins satisfied the essential expectations for the main exchange asset: they were portable, durable, divisible, uniform, acceptable and had limited supply. However, their heavy weight made them impractical to carry around, which led to several countries adopting the gold standard in the 19th century. (Bradfield, 2018)

This meant that governments started to issue paper money guaranteeing to redeem it for its value in gold. Such a system, together with the widespread trust in governments, allowed for the medium of exchange itself to be of a marginal worth compared to the worth attributed to it by the counter-parties conducting the trade. The equivalent of the exchange rate between two currencies was the price difference of a unit of gold between the two currencies. (Bradfield, 2018)

As the price of a unit of gold in gold standard currencies was fixed, exchange rates were also fixed. The actual exchange rates could deviate from the official parity only by the cost of shipping gold — one could either exchange their local currency to a foreign currency or exchange their local currency to gold, ship the gold abroad and exchange gold for the foreign currency in the other country. The costs of shipping gold in one direction or the other set the limits for the actual exchange rate on the market and were called the gold points. With the prevalence of the gold standard, most of the commercial world *de facto* operated in a single currency area. (Bradfield, 2018; A. H. Meltzer, 2020)

The gold standard became unfeasible to uphold during World War I — governments started to print more money to cover their military expenses and suspended the free convertibility of gold. After the war ended, countries aimed to restore the gold standard. As a result of the money printing, prices had increased, and the two feasible options for restoring the gold standard were devaluation and deflation. For the intermediary period before all major currencies were again linked to gold, exchange rates were determined by the market, fluctuating from time to time. However, the gold standard established in the end of 1920s was far from that of the prewar times, forcing deflation and unemployment. As a response, many major economies left the gold standard. (A. H. Meltzer, 2020)

In July 1944 with World War II still waging, the Bretton Woods system was established, ending the era of the gold standard and transitioning to the so-called dollar-standard. Under the Bretton Woods agreement, driven by the United States and the United Kingdom, and signed by more than 40 countries, an adjustable pegged foreign currency market was formed. Governments would fix their currency's exchange rate against the US dollar while the US dollar would be pegged to gold. Ultimately, the US dollar became the benchmark currency and reserve currency — the dollar became the most widely used currency in international trade and countries held most of their reserves in interest-bearing dollar securities. (Bradfield, 2018; A. H. Meltzer, 2020)

Over time, it became more difficult for the United States to honour its commitment of converting dollars into gold at a fixed rate. The system of fixed exchange rates had lived its time. In August 1971, President Richard M. Nixon ended the Bretton Woods system by announcing that the United States would no longer sell gold. This gave way to floating exchange rates determined by the market, as governments no longer had to fulfill the obligation of maintaining a fixed exchange rate between its currency and the dollar or gold. Now, currencies could be valued only against each other and the gold market lost its role in the foreign exchange market. This can be interpreted as the beginning of the modern forex market. (A. H. Meltzer, 2020; Coppola, 2018)

Several countries with high dependency on foreign trade disliked the new order of constantly fluctuating exchange rates. Consequently, some government fixed their currency against another currency, like the US dollar or the German mark. To foster greater economic integration, the Euro currency was formed in the 1990s, replacing national currencies of most of the member states of the European Union. This development significantly decreased the exchange rate risk for European banks and businesses. Most economies abandoned their fixed exchange rate systems by the end of the century. For the first time, it was the foreign exchange market trading that determined foreign currency exchange rates, as opposed to the governments, policies or commodity values. (Bradfield, 2018; A. H. Meltzer, 2020; Coppola, 2018)

1.2 Market participants

The 1990s also witnessed rapid developments in electronic communication, which in turn gave way to opening the forex market up to a wider range of participants. In the 1970s and 1980s, foreign exchange market was directly accessible only to large banks and financial institutions. The spread of the Internet facilitated the creation of online networks between banks and businesses for automated quotation and instant trading. Online trading platforms emerged, allowing individuals to participate in the FX market. Such developments grew the foreign exchange market tremendously and diversified both, the spectrum of market participants, the intentions for engaging in the forex market and the range of FX products offered. (Coppola, 2018)

The modern foreign exchange market is a global and decentralised over-the-counter (OTC) market: there are no central regulations or national markets for individual currencies. Most currencies can be traded against each other in any jurisdiction, although there are some restrictions in place in some jurisdictions. Forex markets operate 24/5.5, opening at 5:00am Sydney time on Monday and closing at 5:00pm New York time on Friday. Practically all counter-parties involved in commerce rely on the FX market in one way or another. The main participants of the foreign exchange market are:

- **Commercial banks** operate as liquidity providers or market makers, facilitating the majority of FX trades between other market participants and earning from commission fees and the difference between the buy (bid) and sell (ask) prices. Commercial banks constitute the interbank market
- **Corporations** engage in the foreign exchange market primarily for two reasons: to transact with counter-parties operating in another currency or to mitigate the risk of currency rate fluctuations impacting the value of its assets and liabilities
- **Investment funds and institutional traders** transact on the currency market for executing international securities transactions and managing their global operations. Furthermore, they may invest their assets on the forex markets with the goal of gaining from exchange rate movements or diversification of their portfolio
- **Government bodies (e.g. central banks)** mainly get involved in the FX market with the purpose of policy implementation and commercial activity
- **Information platforms** such as Reuters and Bloomberg provide real-time high accuracy data about the exchange rates and other aspects of the market. Market participants rely on them for daily market analysis and trading decisions (Kolakowski, 2020)

- **Forex brokers** serve as the link between retail forex traders and the interbank market, offering the possibility to purchase and hold currency pairs. Moreover, forex brokers often allow their customers to access forex leverage (Russell, 2021)
- **Individual investors** speculate on forex markets with the purpose of earning risk-based return, often engaging in day-trading. (FXPA, 2013; Segal, 2020)

According to the most recent Triennial Central Bank Survey of the Bank of International Settlements, the daily trading volume on the foreign exchange market reached $6.6 trillion (BIS, 2019). Large capitalisation of the forex market ensures liquidity and reduces volatility. Despite having only a few members, the interbank trading constitutes most of the daily volume, including trading between banks, trades for clients and facilitated trading by banks' own trading desks. Therefore, the interbank market dictates currency values. The biggest share of the foreign exchange trading is conducted on the London market (43.1%), followed by New York, Singapore, Hong Kong and Tokyo. (Venketas, 2019; Amadeo, 2020)

1.3 Transactions and instruments

Contrarily to other financial markets, the forex is not an absolute value market: it is not pricing or valuing an asset in absolute terms. The foreign exchange market is a relative value market where one currency is valued in terms of another currency. This means that each exchange rate value is impacted by developments in the economies on both sides. (FXPA, 2013)

The currencies that are traded most often are called the major currencies. The major currencies are USD, EUR, JPY, GBP, CAD, CHF, AUD. The most commonly traded currency pairs are similary called the major pairs, all of these include USD on one side: EUR/USD, USD/JPY, GBP/USD, USD/CHF, USD/CAD, AUD/USD. Currency pairs which include one of the major currencies but exclude the US dollar are called major currency crosses or minor pairs. The minor pair with the highest turnover is EUR/JPY. Exotic pairs are currency pairs which include the US dollar on one side and a currency other than one of the majors on the other side, for example, USD/NOK, USD/MXN, USD/SGD. (Markets, 2019)

The US dollar remains the most dominant currency in the foreign exchange market. It was on one side of 88% of all trades in 2019, followed by the euro (32%), the Japanese yen (17%) and the British pound (16%). Currencies of emerging market economies have gradually gained market share, reaching 25% of global turnover by 2019. (BIS, 2019)

A foreign exchange trade or a contract is essentially an agreement to exchange two currencies. There are three main types of trades on the OTC foreign exchange market. The following paragraphs will explain their main characteristics.

Spot trade is an agreement to settle the currencies exchanged in two working days. This means there will be a two working day period between the deal date, when the trade agreement is confirmed, and the value date, when the currency amounts are settled. One must note that weekends and holidays may also have an influence on the two working day period. Originally, the two-day period was meant for performing validity checks and allowing time for processing the transaction. Nowadays, payment processing is much speedier, but one must still consider the timezone differences and the remaining need for prudential controls. Spot trades constitute 30% of the global currency market turnover (BIS, 2019). (Walmsley, 2000; Parker, 2018)

Spot exchange rates on the interbank markets are usually the main measure of the value of one currency in relation to another and form the basis for any other foreign currency exchange quotations. Contrarily to the order-driven equity market, spot market is generally quote-driven, meaning that the market makers provide quotes for executing a trade (FXPA, 2013). The currency for which the traded amount is fixed in a quote is called the *base currency* and the one for which the amount is variable is called the *quoted currency*. Quotes are usually given for pairs in the form of *base currency/quoted currency*. For example, when one writes EUR/USD 1.2034, EUR is the base currency and USD is the quoted currency, and 1 EUR = 1.2034 USD. Quotes can be divided into *direct* and *indirect*, where in the case of the former the foreign currency is the base currency and in case of the latter the domestic currency is the base currency. As the interbank market usually quotes against the US dollar, the *exchange cross rates*, which are exchange rates between two currencies where neither one is the dollar, are derived using the two respective dollar quotes. (Walmsley, 2000; Parker, 2018; Coyle, 2000)

There are two spot rates: *bid rate* at which customers can sell the quoted currency in exchange for the base currency, and *ask rate* at which customers can buy the quoted currency in exchange for the base currency. The difference between the two rates is called *bid-ask spread*. The mid-point between the bid and ask quotes is called the *mid rate* or *mid-market rate*. Bid and ask rates are always set in a favourable direction for the market makers: they would always buy the quoted currency at a lower price and sell it at a higher price, thus the bid rate is always lower than the ask rate. The difference — the bid-ask spread — forms the revenue base for the market makers. Usually the more liquid the market, the narrower the bid-ask spread. While trades on the interbank market between large banks are executed at narrow spreads, smaller market participants seeking to trade would normally be quoted a less favourable price, meaning a wider spread. (Coyle, 2000)

Forward trade (also *forward outright*) is also a trade where two counter-parties exchange the agreed currency amounts between each other once at a pre-determined rate. Forward trades allow for counter-parties to fix the exchange rate at which currencies will be exchanged on the deal date and execute the settlement with this pre-agreed rate on the value date. The main difference from a spot trade is the value date: settlement takes place on any other day than two working days from the deal date. The settlement date may be three or more days after the two working days, but may also fall between the deal date and the spot date. (P. Pontikis, 2012) Exchange deals with a value date before the spot date are called *ante-spot*, *pre-spot* or *short-date* trades: *outright value today* or *outright value tomorrow. Fixed tenors* refer to the standard FX contract durations, most common ones being TOD (today), TOM (tomorrow), SN (spot next or one day after spot), 1 WEEK, 2 WEEK, 3 WEEK, 1 MONTH, 2 MONTH etc. (Parker, 2018; Just, 2021)

The forward rate or the rate at which a forward outright trade is quoted does not represent the expectation for the spot rate on two working days prior to the value date of the forward trade. The forward price is a function of the spot price, the relative difference between the interest rates of the two currencies involved in the deal, and the maturity of the contract. The forward exchange rate neutralises the difference between the interest rates of the two currencies. If this was not the case, investors could always buy currencies with higher interest rates, invest them in time deposits and neutralise the exchange rate risk with a forward contract. The difference between the spot and forward exchange rates is called *forward points.* As the forward exchange rate does not reflect or depend on the expectation of the future spot rate, it also does not reflect the future expectations of the interest rate differential. Therefore, forward rate volatility is driven by several factors: the spot price volatility and the interest rate changes for both currencies. (P. Pontikis, 2012; *Foreign Exchange Training Manual*)

FX swap trade involves two outright trades: it is a combination of two agreements to exchange a currency pair with two different value dates. The two agreements are often called the *legs* of the swap, with the earlier one referred to as the *near leg* and the later one as the *far leg.* FX swaps often contain a spot trade and a forward trade, but could also be a combination of two forward trades. One may also view the FX swap as a combination of a money market deposit and loan in different currencies. This yields a negligible effect on the balance sheet compared to outright trades. In April 2019, FX swaps accounted for almost half of global FX trading. (Just, 2021; Neftci, 2008; BIS, 2019)

Swap trades can be used for moving the forward trade settlement date earlier or later in time, by setting one leg of the swap so that it cancels out the forward trade. The rates of the near and the far leg of the swap trade will be agreed upon at the deal date, taking into consideration the spot rate and the interest rate differential between the two currencies similarly to forward outright pricing. FX swaps' price is the difference between the forward and the spot rates and is mostly determined by the interest rate differential. When trading FX swaps, one needs to cross the bid-ask spread only once despite engaging in two currency exchange transactions in total, and the FX swap spread is generally much tighter than the outright spread. (Just, 2021; Neftci, 2008)

In addition to the three aforementioned trade types, there are several other instruments traded on the foreign exchange market, such as futures, options, non-deliverable forwards (NDFs) and cross-currency basis swaps. Furthermore, a small minority of about 3% of the global turnover of the forex market is formed by order driven instruments traded on exchanges as opposed to the OTC market share. (Markets, 2019; P. Pontikis, 2012; BIS, 2019)

It is common to use *leverage* when trading on the foreign exchange market. This means borrowing from an FX broker to execute currency exchange trades and it is usually called **margin trading**. The trader has to provide the initial deposit to the broker which is called *margin* — its own contribution to the trade — and the broker will lend a multiple of it. The margin percentage required by the broker depends on the customer's profile and amounts traded, but can be as low as 1–2%. Leveraged trading allows to reap higher gains from the market, but also bears the risk of suffering higher losses than one would have when trading without leverage. Margin trading also allows to gain more exposure while using up less liquidity than without margin. This could be beneficial for the purpose of hedging, for example. (Fieldhouse, 2012; Carver, 2019)

1.4 Cost and return

This subchapter will focus on different types of costs and returns associated with holding a foreign currency position or transacting on the foreign exchange market. As in the previous section, focus will be on the outcomes of the most common trade types, acknowledging that more sophisticated instruments also come with more complex cost and return structures and calculations. While the main types of costs and returns will be elaborated, different custom fees applied by FX dealers and brokers exist, but are not included in this section. Furthermore, cost and return components will be explained from the viewpoint of a corporation transacting on the foreign exchange market, as the thesis addresses foreign currency risk mitigation from the perspective of a business.

Similarly to the holding of any other asset, be it real estate or equity, holding a non-zero position in a foreign currency means that its value changes over time. However, exchange rates are relative prices of one currency in terms of another. Therefore, to understand if the value of a foreign currency position has increased or decreased, one should look at the exchange rate of the foreign currency against the *functional currency* of the entity holding the currency position. International Financial Reporting Standards (IFRS) state: "An entity's functional currency is the currency of the primary economic environment in which the entity operates (ie the environment in which it primarily generates and expends cash). Any other currency is a foreign currency." The value of assets and liabilities held in foreign currencies are usually translated into the functional currency using the spot rate. (IAS, 2003)

Definition (FX gains and losses)**.** ***Foreign currency (FX) gain or loss*** *is the difference in the value of the foreign currency balance in the functional currency at the time of valuation (fair value) and at the time of acquiring the position (book value) using the spot rates of the foreign currency against the functional currency at the respective points in time*

$$R_{FX} = N_f \cdot (f_f - f_i), \tag{1}$$

where R_{FX} *is the FX gain or loss in the functional currency,* N_f *is the nominal value of the foreign currency position,* f_f *and* f_i *are the spot rates of the foreign currency in relation to the functional currency at the times of valuing and acquiring the position respectively. In this definition, the foreign currency is the base currency and the functional currency is the quoted currency. (AICPA, 2018)*

In case the foreign currency depreciates against the functional currency, those holding a position in that foreign currency will experience a loss whereas the appreciation of the foreign currency against the functional currency introduces a gain. An **unrealised FX gain or loss** should be recognised if the foreign currency position is still open or unsettled at the time of determining its fair value and calculating the gains and losses. A **realised FX gain or loss** should be recorded when the foreign currency position is sold or settled. f_f in this case should be the spot rate at the time of selling or settling the position. (AICPA, 2018)

When executing spot trades, the market maker will offer a different exchange rate to the counter-party depending on whether it wishes to buy or sell a currency: ask or bid rate respectively.

Definition (Spread cost)**.** ***Spread cost*** *of a foreign exchange trade is the mark-up on the mid-market exchange rate earned by the FX dealer and suffered by the business executing the trade.*

In case of buying a currency, spread cost in absolute terms can be calculated as

$$C_s = N_f \cdot (f_a - f_m), \tag{2}$$

and when selling a currency

$$C_s = N_f \cdot (f_m - f_b), \tag{3}$$

where C_s *is the spread cost in the functional currency,* N_f *is the nominal amount of the foreign currency bought or sold and* f_b*,* f_m *and* f_a *are the bid-, mid- and ask-exchange rates of the foreign currency in relation to the functional currency at the time of the transaction. In this definition, the foreign currency is the base currency and the functional currency is the quoted currency. (Kantox, 2021)*

When discussing the pricing of forward trades in 1.3, we introduced the significance of the interest rate differential arising from the difference of interest rates of different currencies. In addition to the gains or losses incurred from the value change of the currency position held, and transaction costs, there might also be costs or returns related to holding currency positions. (Carver, 2019)

Definition (Carry)**.** ***Carry*** *(also interest) is the difference between earnings and costs of holding a currency position arising from the interest rate differential. Carry interest can be calculated as*

$$I_c = N \cdot (i_{foreign} - i_{functional}), \tag{4}$$

where I_c is the carry interest earned in functional currency, N is the nominal value of the foreign currency position in functional currency, $i_{foreign}$ and $i_{functional}$ are the interest rates on the foreign and functional currencies for the period. (Carver, 2019)

One must keep in mind that carry can be both, positive and negative, as currencies can earn both, positive and negative interest rates. Moreover, it is possible to take short positions in currencies, essentially taking a loan in a foreign currency. This is common practice in margin trading. When a positive position is taken in a higher yielding currency against a negative position in a lower yielding currency with the purpose of gaining from the interest rate difference, it is called *carry trading.* (Carver, 2019; R. Zhang, 2019)

2 Measuring risk

There are numerous varying definitions of risk depending on the discipline and context being discussed. In finance and insurance, risk is often related to suffering financial losses. For the purpose of this thesis, risk is defined as follows.

Definition (Risk)**.** ***Risk*** *is the possibility of an adverse deviation from the desired outcome (Outreville, 1998).*

For financial institutions, the outcome in question is usually the future net return for which the company would often have an expectation. There are several fundamental sources of this uncertainty: credit risk, operational risk, liquidity risk and market risk. In FX trading, *market risk* is usually prevailing — it is the uncertainty of future returns due to changes in market conditions. When discussing foreign currency positions and trading, market risk is dominated by *exchange rate risk* which is the uncertainty of future returns due to exchange rate changes. This thesis will concentrate on the latter. (S. Manganelli, 2001; Döhring, 2008)

Once the existence of risk is recognised, there are several ways to operate: decide to *accept* the risk and take no further action with regards to it, *eliminate* the risk with the consideration of possibly having to accept the side effects or costs of elimination, *avoiding* the risk in the first place by modifying behaviour or *transfer* the risk to a third party, often by insurance contracts (Käärik, 2019). If a company's business model demands holding foreign currency positions, avoidance might not be possible and acceptance might not fall under the risk appetite of the business. Therefore, such companies often need to develop strategies to eliminate or reduce the exchange rate risk, for example, by conducting currency hedging.

Definition (Currency hedging)**.** ***Currency hedging*** *by an entity holding a foreign asset is the elimination or reduction of exchange rate risk by taking an opposite position in the foreign currency resulting in the reduction of the volatility of the value of its total position due to exchange rate fluctuations (M. M. Dacorogna, 2001).*

Measuring risk usually comes down to measuring one or both of the two following aspects: the possibility of a negative outcome (*loss*) and the deviation or variability of the outcome over time in terms of the expected result (*volatility*) (Righi, 2019). Decision on whether to include one, the other or both into the risk measurement depends on what is seen as risk by the entity using the risk measurement. This subchapter provides an overview of several ways of measuring risk, concentrating on both of these aspects.

2.1 Volatility

In financial context, *volatility* is often measured as the *standard deviation* of the rates of logarithmic returns of an asset, e.g. foreign currency position (Righi, 2019).

Definition (Logarithmic return)**.** ***Logarithmic return****, shortly log return, is the measure of gain or loss of an investment over a given period of time with respect to changes in its market value, which is computed as*

$$r_t = \ln\left(\frac{S_t}{S_{t-\Delta t}}\right), \tag{5}$$

where r_t *is the log return of an asset for time* t *and time interval of length* Δt*,* $S_{t-\Delta t}$ *is its price at time* $t - \Delta t$*, the beginning of the observed time interval, and* S_t *is its price at time* t*, the end of the observed time interval. (Kotze, 2005)*

Research has shown that the normality assumption for the distributions of returns on foreign exchange markets usually does not hold. There are two phenomena that often occur in the time series of foreign exchange returns: heavy tails and volatility clustering. Therefore, assuming normality for the returns on foreign exchange markets can lead to under-estimating the probability of extreme events, both on the gaining and losing side. (Cotter, 2005)

Standard deviation of log returns — the measure of *volatility* — is the most common measure of *risk*. The higher the standard deviation of the return of an asset, thus also the higher the volatility, the higher the risk associated with the asset. However, the true volatility of an asset of interest is hard to measure as well as constantly changing. Therefore, different methods are used to estimate the volatility. One of the most common methods is calculating historical volatility of an asset. (Kotze, 2005)

Definition (Historical volatility)**.** ***Historical volatility*** *is a measure of risk associated to an asset calculated as the sample standard deviation of a historical time series of the log returns of the asset and is computed as*

$$\hat{\sigma}_r = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_i - \bar{r})^2}, \tag{6}$$

where $\hat{\sigma}_r$ *is the historical volatility of the asset,* n *is the number of equal time intervals in the historical time period from which the sample of log returns is obtained,* r_i *is the log return for the* i*-th time interval and* $\bar{r}$ *is the sample mean log return defined as*

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i. \tag{7}$$

(Kotze, 2005)

As volatility is usually far larger during trading hours than during market off-hours, periods when trading is closed are usually ignored. Therefore, the time series of historical returns used for estimating historical volatility often use consecutive business days or market open hours as the equal time intervals. (Kotze, 2005)

Definition (Historical covariance)**.** ***Historical covariance of two assets*** *j and k is defined as*

$$\hat{c}_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(r_{ij} - \bar{r}_j)(r_{ik} - \bar{r}_k) \tag{8}$$

where $\hat{c}_{jk}$ is the historical covariance of the two assets j and k, n is the number of equal time intervals in the historical time period from which the sample of log returns is obtained, r_{ij} and r_{ik} are the log returns of assets j and k respectively for the i-th time interval and $\bar{r}_j$ and $\bar{r}_k$ are the sample mean log returns of assets j and k respectively. (E. J. Elton, 2009)

Definition (Historical portfolio volatility)**.** ***Historical portfolio volatility*** *of m assets is defined as*

$$\hat{\sigma}_P = \sqrt{\sum_{j=1}^{m}\sum_{k=1}^{m} w_j w_k \hat{c}_{jk}} = \sqrt{\mathbf{w}^T\hat{\mathbf{C}}\mathbf{w}} \tag{9}$$

where $\hat{\sigma}_P$ is the historical volatility of the portfolio, $\hat{c}_{jk}$ is the historical covariance of the returns of two assets estimated over a finite historical time interval, $\hat{c}_{jj} = \hat{\sigma}^2_{r_j}$ is the square of the historical volatility of the log returns of the j-th asset, w_j is the weight of the j-th asset in the portfolio, $\mathbf{w}$ is the vector of weights and $\hat{\mathbf{C}}$ is the historical covariance matrix. (E. J. Elton, 2009)

Assuming that random variables corresponding to log returns for non-overlapping time intervals are iid random variables, the standard deviation of the log returns of an asset or of a portfolio of assets can be scaled for different time periods using the square root of the time interval in the units of the time interval for which the standard deviation is given as the scaling factor

$$\sigma_h = \sigma\sqrt{h}, \tag{10}$$

where σ is the standard deviation for an interval of 1 unit and σ_h is the standard deviation for an interval of h units. Historical volatility for different period lengths can be estimated using the same relation

$$\hat{\sigma}_h \approx \hat{\sigma}\sqrt{h}, \tag{11}$$

where $\hat{\sigma}$ is the historical volatility for an interval of 1 unit and $\hat{\sigma}_h$ is the estimate of historical volatility for an interval of h units. For example, to convert daily volatility to hourly volatility, one would need to execute the following calculation: $\hat{\sigma}_{hour} = \hat{\sigma}_{day}\sqrt{1/24}$. (Kotze, 2005)

One of the main decision points when estimating risk with historical volatility is the choice of n — the length of the historical period over which volatility is calculated. Historical volatility is quite sensitive to outliers: a single volatile interval, depending on the choice of n, might overestimate volatility for the immediate future. However, the opposite might also occur, where historical volatility calculated over a calm period will not give a reasonable volatility estimate for the immediate future in case of the occurrence of a significant market event inducing a high volatility period. Nevertheless, due to volatility clustering, historical volatility might give useful insights for forecasting the immediate future. Historical volatility method can be modified to give different weights to time intervals, e.g. assigning higher weights to the more recent time intervals. (L. H. Ederington, 2006)

Using standard deviation or variance as the standalone measurement of risk is often criticised due to the fact that it only addresses the volatility or uncertainty aspect of risk. If the returns of a particular asset are trending upwards over time, higher volatility can increase the return potential of an investment. This is, however, usually not the case for currency portfolios held for conducting international business and not for speculation. Standard deviation assumes a symmetric risk distribution and assumes that unexpected gains are as risky as unexpected losses. Furthermore, standard deviation is not an intuitive metric to grasp the magnitude of risk or communicate it to relevant stakeholders. Nevertheless, standard deviation can be used as an input to compute more comprehensive risk metrics such as Value at Risk and Conditional Value at Risk, defined in subchapters 2.3 and 2.4. (Keppler, 1990; Sumnicht, 2009)

In addition to the historical volatility method and its modifications that use the historical returns of the asset of interest, implied volatility can be used. Implied volatility is a future-looking theoretical volatility measure of an asset. It is usually calculated based on current option prices for the period of interest, where the underlying instrument is the asset for which volatility estimate is derived. Implied volatility derivation uses option pricing models, such as the Black-Scholes model. Furthermore, GARCH (Generalized Autoregressive Conditional Heteroskedasticity) models are used for volatility estimation as they consider fat tails and volatility clustering, both significant characteristics of returns distributions. (Kotze, 2005; L. Lyons, 2005)

2.2 Expected loss and downside semi-deviation

Investors and asset holders usually perceive potential losses and the volatility of their magnitude as risk. Therefore, metrics focusing only on the downside, are further suggested. Such measures also account for any upside- or downside-bias that can occur. The following subchapters define some of the possible metrics of risk which focus on the downside of returns — the aspect of losses in the concept of risk. (Keppler, 1990; Sumnicht, 2009)

Definition (Loss)**.** ***Loss*** *is a measurement of realised risk calculated as*

$$L = -\min(X, 0), \tag{12}$$

where X is the random variable which corresponds to the monetary outcome of an investment. (Keppler, 1990)

When calculating loss in the context of financial returns, one would take $X = r$ where r is the log return of an asset and loss would then be calculated as

$$L = -\min(r, 0). \tag{13}$$

Expected loss is also used as a measurement of risk. It follows from probability theory that expected loss can be calculated using the conditional expectation formula: expected loss can be calculated as the product of the expected loss amount for the period given that the loss occurred and the probability of making a loss

$$\mathbf{E}[L] = \mathbf{E}[-\min(r, 0)] = -\mathbf{E}[r \mid r < 0] \cdot P(r < 0). \tag{14}$$

Similarly to volatility, expected loss can be estimated using historical time series of log returns. The probability of loss would then be calculated as the proportion of losing periods out of all periods. The conditional expectation of log returns would be calculated as the average loss size over all losing periods. (Keppler, 1990)

Definition (Downside semi-deviation)**.** ***Downside semi-deviation*** *is a downside risk metric measuring the below-mean variability of log returns, calculated as*

$$\sigma_{semi} = \sqrt{\mathbf{E}[(\min(0, r - \mu_r))^2]}, \tag{15}$$

where r is the log return of the asset and μ_r is the mean log return.

Semi-deviation can be estimated by the sample semi-deviation

$$\hat{\sigma}_{semi} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\min(0, r_i - \bar{r}))^2}, \tag{16}$$

where n is the number of equal time intervals in the historical time period from which the sample of log returns is obtained, r_i is the log return for the i-th interval and $\bar{r}$ is the sample mean log return. (Hardy, 2006)

If the investor is concerned by returns below a certain threshold different from the mean return level, e.g. only negative returns, semi-deviation can be modified to calculate the threshold semi-deviation. For that purpose the mean return $\bar{r}$ should be replaced by a suitable threshold value τ, e.g. $\tau = 0$. (Hardy, 2006)

2.3 Value at Risk (VaR)

Definition (VaR)**.** ***Value at Risk (VaR)*** *is a statistical risk metric of potential losses: Value at Risk VaR_α at a given confidence level $\alpha \in (0,1]$ and a given time horizon is the smallest value q for which the probability that the loss variable L for the given time horizon is higher than the threshold value q is smaller than $1-\alpha$ under normal market conditions. VaR_α can be calculated as*

$$VaR_\alpha = min\{q : P(L \leq q) \geq \alpha\}, \tag{17}$$

where L is the loss variable and α is the confidence level. (Jorion, 2007)

VaR_α can also be written as

$$VaR_\alpha = F_L^{-1}(\alpha), \tag{18}$$

where $F_L(y)$ is the cumulative distribution function of the loss random variable L (Hardy, 2006). Mathematically, at the confidence level α, VaR_α is the α-percentile of the distribution of loss for a given period of time. Value at Risk is usually given as a positive number. For the case of foreign exchange gain or loss, the loss variable can be written as

$$L = -R_{FX}. \tag{19}$$

The choices of the confidence level and time horizon are relatively subjective and depend on the nature of the portfolio and the purpose of calculating VaR. Higher confidence level should result in fewer occurrences of losses exceeding VaR, but will also yield a higher VaR. As risk increases with the time horizon, choosing a longer time horizon will also increase VaR. (Jorion, 2007)

Although VaR is more complex to calculate than the previously mentioned risk metrics, it is a forward-looking metric and is easily explainable to various audiences. Furthermore, it gives an aggregate view of a portfolio's risk, taking into consideration the current positions of individual assets, their risk and the correlation of risk between different assets in the portfolio. VaR measures risk in the same units as the bottom line of a company, in monetary units, which makes it easier to comprehend. (Jorion, 2007)

The main steps in estimating VaR for a given portfolio are as follows:

1. mark to market the current portfolio
2. estimate the loss distribution
3. set the time horizon
4. set the confidence level α
5. calculate the VaR_α.

The most complex of the above steps is usually the second one. There are several approaches one could take to estimate the loss distribution which can be divided into: (a) parametric, (b) non-parametric and (c) semiparametric. While parametric VaR models assume the loss distribution to belong to a parametric family, such as the normal distribution or Student's t-distribution, the non-parametric VaR models make no assumptions about the distribution of losses, e.g. using historic data or Monte-Carlo simulations. (S. Manganelli, 2001; Jorion, 2007)

Each approach to estimating the loss distribution has its pros and cons. In case of the parametric approach, the chosen parametric distribution might not represent the actual loss distribution well enough and result in under- or over-estimated VaR. In case of using historic data to calculate VaR, the assumption that losses in the near future are identically distributed as those from the past, which also might not be true and can result in under- or over-estimated VaR respectively, when market moves from a calmer period to a more volatile one or vice versa. (S. Manganelli, 2001)

VaR has some undesirable mathematical characteristics: lack of subadditivity and convexity. Furthermore, it is coherent only under the normality assumption of returns. As an example, VaR for a combination of two portfolios can be greater than the sum of the individual VaRs of the two portfolios. Lastly, VaR can turn out to be difficult to optimise, sometimes resulting in multiple local extrema. (S. Uryasev, 1999)

2.4 Conditional Value at Risk (CVaR)

One of the main aspects of VaR that has received critique is that it carries no information about the losses exceeding the VaR threshold. Conditional Value at Risk (CVaR), also Mean Excess Loss, Mean Shortfall, Expected Shortfall, or Tail VaR, aims to fill this gap. With the same confidence level, portfolios with low CVaR also have a low VaR, as VaR is the lower bound for CVaR by definition. Contrarily to VaR, CVaR possesses some favourable mathematical characteristics, it is a coherent risk measure. (S. Manganelli, 2001; S. Uryasev, 1999)

Definition (CVaR)**.** ***Conditional Value at Risk (CVaR)*** *at a given confidence level* $\alpha \in (0, 1]$ *and a given time horizon is the conditional expectation of losses* *above VaR_α for the same time horizon under normal market conditions and can be computed as*

$$CVaR_\alpha = \mathbf{E}[L \mid L > VaR_\alpha], \tag{20}$$

where L is the loss variable and VaR_α is the value at risk level for the confidence level α.

If the loss distribution is continuous, at least for loss values higher than VaR_α, conditional Value at Risk can also be written as

$$CVaR_\alpha = \frac{1}{1-\alpha} \int_{VaR_\alpha}^{\infty} y f_L(y) dy, \tag{21}$$

where $f_L(y)$ is the distribution function of the loss random variable L. (S. Uryasev, 1999; Hardy, 2006)

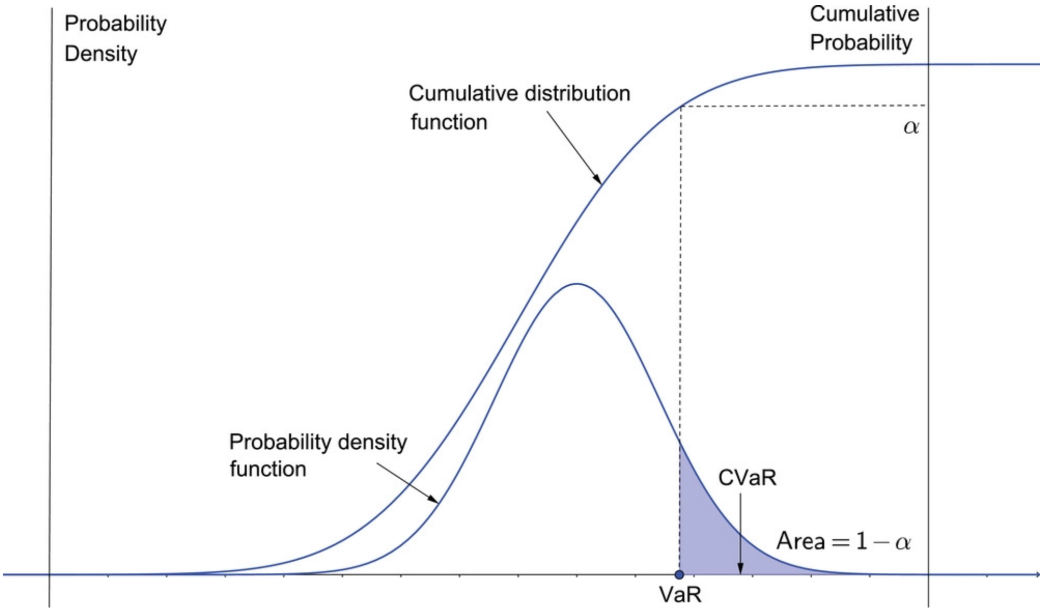

Figure 1: Graphic illustration of VaR and CVaR (C. Filippi, 2020)

Figure 1 illustrates the concepts of VaR and CVaR for a loss random variable with a continuous probability density function. α denotes the confidence level of the VaR and CVaR and the highlighted area under the probability density function where loss is higher than VaR is $1 - \alpha$. Furthermore, one can see how VaR is the effective lower bound for the CVaR value.

3 Validation and development of risk metrics

This chapter focuses on the empirical validation and development of the current VaR model used for monitoring and managing the foreign currency risk at the global FinTech company Wise. Firstly, a brief overview of the company's business model and its relation to foreign exchange markets is given. Secondly, the current VaR model set-up is explained. Thirdly, the data used in the empirical part of the thesis is introduced. Lastly, the two different methods: determining empirical percentile and Bayesian Model Averaging are introduced, applied and compared. In addition to validating and suggesting improvements to the current VaR model of the company, CVaR multiplier estimates based on the same model are also proposed.

3.1 Case of a global FinTech company

Wise is a global FinTech company founded in 2010 with a mission to help people manage their money internationally at a lower cost, higher speed and transparency. The company has over 10 million customers and processes over 5 billion EUR in cross-border transactions every month. Over 30% of its international transfers are instant (delivered in less than 20 seconds). Wise saves around 1.15 billion EUR every year for its customers compared to using a traditional bank for cross-currency transactions. The company offers a range of products involving currency conversions, the main ones being:

- foreign currency transfers at mid-market (spot) rate with a transparent fee charged on top (no hidden spread in the exchange rate)
- multi-currency account to hold several currency balances and convert your balance between currencies at mid-market rate
- local bank details to receive funds to your multi-currency account like a local
- debit card to spend your multi-currency account balance in whichever currency
- direct debits to spend your multi-currency account balance for recurring payments.

Wise operates globally — there are 80 countries one can send money to and 43 countries one can send money from using Wise. The multi-currency transfer service covers 2500 currency routes and offers 55 currencies. The Wise multi-currency account is available in more than 170 countries, offering local bank details in 10 currencies and the ability to hold a balance in additional 45 currencies. Furthermore, Wise has issued 1.4 billion debit cards since their launch in 2018, which can be used for spending in any currency in the world, in which one can make purchases using a debit card.

Wise manages to keep its currency coverage high and fees low by using a network of local bank accounts instead of the traditional slow and costly correspondent banking. When a customer makes a cross-currency transaction with Wise, money never actually crosses borders, as the payin and payout are made to and from the company's local bank accounts. Nevertheless, from the company's point of view, not all routes are symmetric and in the course of its natural business cycle, Wise accumulates excess assets in some currencies and excess liabilities in others. As a result, the treasury department needs to trade currencies in bulk to cater for the liquidity needs of the product offering.

Furthermore, the treasury function is also responsible for monitoring the foreign exchange risk and taking hedging actions to mitigate the risk. As the company offers cross-currency products 24/7, the currency positions are constantly changing. A Value at Risk model has been developed for live monitoring of the foreign currency risk. The model gives a live view of the daily 95% VaR. Several thresholds have been set to the daily VaR level — when they are reached, appropriate hedging and other risk mitigation actions are triggered. The VaR model in use plays an essential role in the risk management framework of the company. Therefore, it is essential to eliminate the model risk and validate the accuracy of the outputs of the VaR model, adjusting its parameters if needed. In addition, Wise does not currently monitor any metrics describing the potential losses in case VaR is exceeded. This thesis also aims to bridge this gap and suggest parameter ranges for calculating the Conditional Value at Risk metric based on the VaR model already in use.

The current VaR model of the company uses a parametric approach, namely the variance-covariance method for the live calculation of the live VaR level:

1. The vector $\mathbf{N}$ of current notional values of foreign currency positions is calculated: the current net currency positions are marked to market — that is, converted to GBP as the local currency for the portfolio — using the latest available mid-market exchange rates obtained mainly from Reuters, but also from XE and some central banks.
2. For each day, the variance-covariance matrix $\hat{\mathbf{C}}$ of log returns of currencies is estimated as follows:
    (a) hourly log returns are calculated from to-GBP mid-market currency rates for the last eight business days (excluding weekends)
    (b) for each date and currency, Kendall tau correlations and standard deviations of the hourly log returns are calculated
    (c) estimates for both, correlations and standard deviations are exponentially smoothed over the eight days using smoothing rate of 0.95 for correlations and 0.8 for standard deviations

(d) exponentially smoothed estimates of correlations and standard deviations of log returns are used to compile the estimated covariance matrix of log returns.

3. The FX gain and loss distribution is assumed to be a normal distribution with zero mean and variance equal to the variance of the portfolio.

$$R_{FX} \sim \mathcal{N}(0, \sigma^2), \tag{22}$$

where variance is equal to

$$\sigma^2 = \mathbf{N}^T \mathbf{C} \mathbf{N}, \tag{23}$$

where $\mathbf{N}$ is the vector of notional values of net foreign currency positions in GBP units and $\mathbf{C}$ is the variance-covariance matrix of the log returns of the assets in the portfolio. However, as the actual covariance matrix and therefore also the actual variance of the portfolio is unknown, it is estimated by

$$\hat{\sigma}^2 = \mathbf{N}^T \hat{\mathbf{C}} \mathbf{N}, \tag{24}$$

where $\hat{\mathbf{C}}$ is the estimate of the variance-covariance matrix introduced in the previous step.

4. The company monitors the daily Value at Risk at a 95% confidence level. Therefore, the live VaR level is calculated as

$$VaR_{95} = |F_{R_{FX}}^{-1}(0.05)|, \tag{25}$$

where $F_{R_{FX}}^{-1}(p)$ is the inverse cumulative distribution function of the FX gain or loss random variable R_{FX}. VaR is usually presented as a positive value, therefore also the absolute value is taken. As the FX gain or loss random variable is assumed to be normally distributed, VaR_{95} is approximately equal to

$$VaR_{95} \approx 1.645 \cdot \hat{\sigma}. \tag{26}$$

Prior research has shown that the normality assumption does not usually hold for returns on the foreign exchange market. Therefore, the current assumption of the model might result in inaccurate VaR estimates, potentially under-estimating the risk. Furthermore, several assumptions are taken when estimating the variance-covariance matrix of the log returns of currencies. For the aforementioned reasons, model validation is necessary to confirm the adequacy of the VaR model in use by the company.

3.2 Data description

The following data sets are gathered from the company's data sources for a 22-month long period from 01/07/2019 to 30/04/2021.

1. **Beginning of the hour net foreign currency positions** for each hour in the period are obtained. GBP is taken as the local currency of the company and there are 143 foreign currencies in which the company held a non-zero position at least once during the period observed. The net position is defined as the sum of assets, liabilities and hedges executed on margin and is given in the units of the respective foreign currency.

2. **Beginning of the hour to-GBP mid-market currency rates** for each hour in the period are gathered. The primary source for the rates stored in the company database is Reuters. XE and local central banks' data sources are used for a few currencies. Exchange rates are gathered for all 143 foreign currencies in which the company held a non-zero position at least once during the period.

   During data gathering, it was noted that rates for some currencies and some hours are missing from the data set. However, the missing data does not constitute a large part of the rates data set: 69% of currencies have $< 0.5\%$ of values missing, 76% of currencies have $< 1\%$, 92% of currencies have $< 2\%$ and 97% of currencies have $< 3\%$ of values missing. Out of the four currencies which have more than 3% of values missing, one is a currency in which the company only held a negligible non-zero position at 2% of the hourly intervals in the observed period, two are currencies in which the company does not hold a large position over the period and one is a currency in which the company holds a moderate position compared to the portfolio size. To address the missing data, forward-filling of rates was used: NaN values were replaced with the previous available hourly rate for the currency. In case the first rate or rates of the period were missing, back-filling was used: NaN values were replaced with the earliest available hourly rate for the currency.

3. **Daily covariance matrix estimates of log returns** for each day and all currencies held in the portfolio were extracted. These are the same covariance matrices that the company uses for its current VaR calculation. The covariance matrix estimates are taken as given in this thesis, as the purpose is to validate the current model without changing the underlying data used by the model.

For the purpose of validating the current VaR model, **the ratio of realised risk to the estimated standard deviation of the risk** is calculated. While the VaR model gives a live estimate of the standard deviation of the risk for the immediate future, the risk amount that will realise in the immediate future is unknown. Information about the relation between the realised risk and the estimated standard deviation of the risk can inform assumptions about the magnitude of risks that can potentially realise in the immediate future. Therefore, the following subchapters will proceed with investigating the mentioned ratio of realised risk to the estimated standard deviation of the risk.

Realised risk is measured as the FX gain or loss and the estimate of standard deviation of the risk is calculated in line with the algorithm used in the current VaR model. The mentioned ratios are calculated for periods of length $h = 1, 2, 3, ..., 24$ hours and for each i-th consecutive non-overlapping interval of length h in the period, where $i = 1, 2, 3, ..., n$ and n is the total number of respective intervals in the period. The value of n is different for each value of h, as only non-overlapping periods are taken into consideration.

$$x_{ih} = \frac{R_{FXih}}{\hat{\sigma}_{ih}} \tag{27}$$

1. **Realised risk** or FX gain or loss is firstly calculated separately for each currency as the product of the beginning of the i-th period net foreign currency position and the difference between the to-GBP exchange rate at the end and at the beginning of the i-th period as defined in equation 1. The total FX gain or loss of the portfolio R_{FXih} is the sum of the FX gains and losses of the individual currency positions for the i-th period.

2. **Estimated standard deviation of the risk** $\hat{\sigma}_{ih}$ is calculated as per equation 24, where the vector of notional values of net foreign currency positions in GBP units is taken for the beginning of the i-th period and the covariance matrix of the respective day in which the i-th period begins, is scaled according to the period length h as per equation 11.

As the raw data obtained has an hourly granularity, when $h = 1$, the ratio x_{i1} is calculated for each hour in the period. For $h \geq 2$, the ratio x_{ih} is calculated for consecutive non-overlapping time periods, first period starting at the first data point available in the raw data: the midnight of the first day in the data set. For example, in case $h = 4$, the first value of the ratio $x_{1,4}$ is calculated for 01/07/2019 00:00 – 04:00, second value $x_{2,4}$ for 01/07/2019 04:00 – 08:00 etc.

Foreign exchange markets operate 24/5.5 and the company is interested in monitoring the risk metrics during the period when markets are open. This is because during the period when markets are closed, the exchange rates do not change and as a result no FX gains or losses are experienced — no risk is realised. Therefore, time periods which contain at least one hour from the market off-hours period of 22:00 Friday to 21:00 Sunday UTC, are excluded from the time series.

The two methods used for validating the VaR model, described in the following subchapters, assume that the vectors $\mathbf{x}_h$ consist of independent observations. To confirm this assumption, the time series for hourly ratios $\mathbf{x}_1$ was tested for autocorrelation. Figure 2 depicts the output of the pandas `autocorrelation_plot` function under the `plotting` module. The horizontal lines in the plot correspond to 95% and 99% confidence bands. One can observe some autocorrelation coefficients in the $\mathbf{x}_1$ time series being slightly over the 95% and 99% confidence bands

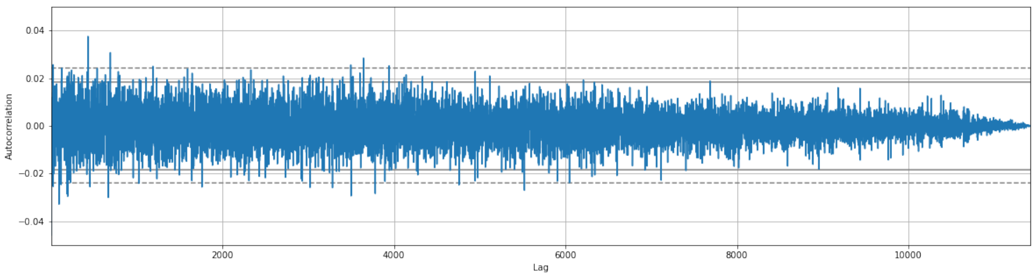

Figure 2: Autocorrelations of $\mathbf{x}_1$, the hourly ratios of realised risk to the estimated standard deviation of the risk.

suggesting a weak autocorrelation. However, the autocorrelation can be deemed weak enough to be insignificant and for further calculations, the vectors $\mathbf{x}_h$ will be assumed to consist of independent observations.

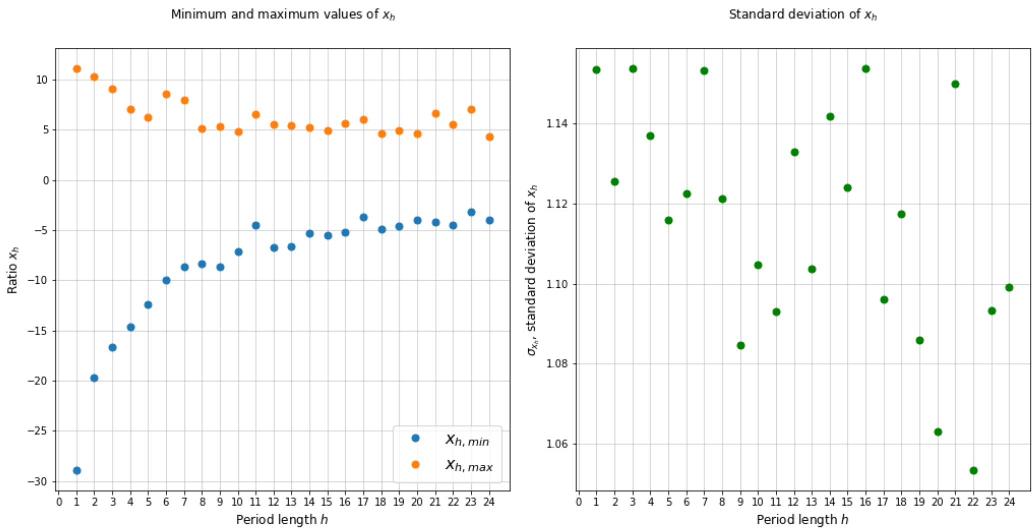

Figure 3: Minimum and maximum values and standard deviation of x_h.

In Appendix 1, Figure 12, the histograms of $\mathbf{x}_h$ for all values of h are included. It can be noted that the sample mean is very close to zero, ranging from -0.0143 to 0.0187. However, the sample standard deviation is larger than one for all values of h (see Figure 3), ranging from 1.0534 to 1.1538. Although $\mathbf{x}_h$ for smaller values of h can be noted to have a standard deviation in the higher range of observed values, there is no clear trend in sample standard deviation of $\mathbf{x}_h$ decreasing with h increasing. Nevertheless, extreme values, especially on the negative side, where realised risk exceeds the estimated standard deviation of the risk, are higher in absolute terms for smaller values of h. $\mathbf{x}_{h,min}$ and $\mathbf{x}_{h,max}$ do not decrease significantly with the increase of h for $h \geq 11$, as can be observed on Figure 3.

3.3 Empirical percentile method

The current VaR model assumes the ratios x_{ih} to be standard normal random variables $x_{ih} \sim \mathcal{N}(0,1)$. Using historical data, the aim of the **empirical percentile method** is to validate this assumption and give empirical estimates for the **VaR multiplier** values $q_{\alpha h}$, calculated as the absolute values of $(1-\alpha)$-percentile of $\mathbf{x}_h$. The empirical percentile values could be used to calculate the live $VaR_{\alpha h}$ for the currency portfolio by

$$VaR_{\alpha h} = q_{\alpha h} \cdot \hat{\sigma}_h, \tag{28}$$

where α is the confidence level of the VaR value, h is the length of the period for which the VaR value is computed in hours and $\hat{\sigma}_h$ is the estimate of the standard deviation of the risk computed as per equation 24 using the current vector of notional values of net foreign currency positions in GBP units and the estimate of the variance-covariance matrix for the current day scaled to the period of length h hours as per equation 11. As the company's risk management policy sets thresholds to 95% VaR, the confidence level is set to $\alpha = 0.95$ for all following computations.

The company does not currently monitor any risk metrics describing the potential losses once VaR is exceeded. To improve the risk management tools of the company, the empirical **CVaR multipliers** $p_{\alpha h}$, which are calculated as the mean tail ratios given $q_{\alpha h}$ is exceeded, is also calculated based on the empirical values of $\mathbf{x}_h$ as

$$p_{\alpha h} = \left| \frac{\sum_{i=1}^{n} x_{ih} I_{q_{\alpha h}}(x_{ih})}{\sum_{i=1}^{n} I_{q_{\alpha h}}(x_{ih})} \right|, \tag{29}$$

where $I_{q_{\alpha h}}(x_{ih})$ is the indicator function indicating whether the VaR level for the respective period length and confidence level has been exceeded

$$I_{q_{\alpha h}}(x_{ih}) := \begin{cases} 1 & \text{if} \;\; x_{ih} < -q_{\alpha h} \;, \\ 0 & \text{if} \;\; x_{ih} \geq -q_{\alpha h} \;. \end{cases} \tag{30}$$

Using the empirical CVaR multiplier $p_{\alpha h}$, the live $CVaR_{\alpha h}$ for the currency portfolio can be calculated similarly to $VaR_{\alpha h}$ as

$$CVaR_{\alpha h} = p_{\alpha h} \cdot \hat{\sigma}_h. \tag{31}$$

3.4 Bayesian Model Averaging method

In addition to the non-parametric empirical percentile method, a simplified Bayesian Model Averaging (BMA) method is used to estimate the values of $q_{\alpha h}$ and $p_{\alpha h}$. This subchapter and the application of the BMA method is based on David J.C.

MacKay's book "Information Theory, Inference, and Learning Algorithms" (MacKay, 2003). The method is based on fitting several different continuous distributions to the data set. Instead of choosing one of the distributions as most suitable as per some appropriate metric, VaR and CVaR multipliers are computed from all the fitted distributions and the final estimates for $q_{\alpha h}$ and $p_{\alpha h}$ are given as the likelihood weighted averages of the individual distributions' VaR and CVaR multipliers.

Firstly, a list of k different candidate distributions is set: double gamma, double Weibull, exponentially modified normal, folded normal, generalised normal, generalised extreme value, left-skewed Gumbel, right-skewed Gumbel, hyperbolic secant, Laplace, logistic, Moyal, non-central Student's t, normal, normal inverse Gaussian, Pearson type III, skew-normal, Student's t, triangular, truncated normal and Tukey-Lambda. The list of candidate distributions is chosen based on available continuous probability distribution functions in the Python `scipy.stats` module with a support for the argument on the whole real line. (SciPy, 2021) Equal prior probabilities are set for all candidate distributions

$$P(\mathcal{H}_k) = \frac{1}{m} \ \forall k, \tag{32}$$

where $\mathcal{H}_k$ denotes the k-th candidate distribution, $k = 1, 2, ..., m$ and $m = 21$ is the number of candidate distributions.

Secondly, all m candidate distributions are fitted to the data sets $\mathbf{x}_h$ and the maximum likelihood estimates (MLE) for distribution parameters are obtained. This step is conducted using the `scipy.stats.rv_continuous.fit` method, which maximises the log-likelihood function. Using the MLE estimates of the parameters, the estimates for $q_{\alpha h}$ and $p_{\alpha h}$ are calculated for each fitted distribution as

$$\hat{q}_{k\alpha h} = -F_k^{-1}(1-\alpha) \tag{33}$$

$$\hat{p}_{k\alpha h} = -\frac{1}{1-\alpha}\int_{-\infty}^{-\hat{q}_{\alpha h}} y f_k(y) dy, \tag{34}$$

where $F_k^{-1}(\beta)$ and $f_k(y)$ denote the inverse cumulative distribution function and the probability distribution function of the k-th fitted distribution.

Next, to obtain weights for the multipliers calculated for each model, the posterior probabilities of each models given the data D need to be assessed. Based on the posterior probability of each model being

$$P(\mathcal{H}_k \mid D) \propto P(D \mid \mathcal{H}_k) P(\mathcal{H}_k) \tag{35}$$

and $P(\mathcal{H}_k)$ being equal for all models, $P(D \mid \mathcal{H}_k)$, the evidence of each model needs to be determined. Evidence can be computed as

$$P(D \mid \mathcal{H}_k) \simeq P(D \mid \mathbf{w}_{k,ML}, \mathcal{H}_k) \cdot P(\mathbf{w}_{k,ML} \mid \mathcal{H}_k) \sigma_{w|D}, \tag{36}$$

where the first term $P(D \mid \mathbf{w}_{k,ML}, \mathcal{H}_k)$ is the best fit likelihood function value and the second term is called 'Occam factor' with a value less than one that penalizes $\mathcal{H}_k$ for having the parameters $\mathbf{w}$. For the purpose of this thesis, a simplified approach is taken, the Occam factor is not estimated and evidence is approximated as

$$P(D \mid \mathcal{H}_k) \approx P(D \mid \mathbf{w}_{k,ML}, \mathcal{H}_k) \tag{37}$$

and the posterior probability based weights of the models are taken to be proportional to the likelihood function value at the MLE estimates for parameters

$$u_k \propto P(D \mid \mathbf{w}_{k,ML}, \mathcal{H}_k). \tag{38}$$

In practice, the following steps are taken to compute the weights u_{hk} for each combination of period length h and model k:

1. Log-likelihood function values l_{hk} for the given data and MLE parameter estimates is calculated using the `scipy.stats.rv_continuous.logpdf` method.

2. The log-likelihood function values are large negative numbers and taking the exponent of them directly to obtain the likelihood function value would often yield 0 as the result due to rounding. Therefore, for each iteration of computing the estimates of $q_{\alpha h}$ and $p_{\alpha h}$, the maximum log-likelihood of all k values $l_{hk,max}$ is determined and subtracted from all k models' log-likelihood values
$$l^*_{hk} = l_{hk} - l_{hk,max}. \tag{39}$$
This essentially scales the log-likelihoods of all models into the units of the log-likelihood of the model with the highest log-likelihood.

3. Finally, the weights are calculated as exponents of the scaled log-likelihoods
$$u_{hk} = \exp{(l^*_{hk})}, \tag{40}$$
which still ensures that the weights are proportional to the likelihood function values, as
$$u_{hk} = \exp{(l^*_{hk})} = \exp{(l_{hk} - l_{hk,max})} = \frac{\exp{(l_{hk})}}{\exp{(l_{hk,max})}} \propto \exp{(l_{hk})}, \tag{41}$$
where the likelihood function value L_{hk} can be written as
$$L_{hk} = \exp{(l_{hk})}, \tag{42}$$
and therefore
$$u_{hk} \propto L_{hk}. \tag{43}$$

Lastly, the likelihood-weighted average estimates for VaR and CVaR multipliers are calculated as

$$\hat{q}_{\alpha h} = \frac{\sum_{k=1}^{m} u_{hk}\,\hat{q}_{k\alpha h}}{\sum_{k=1}^{m} u_{hk}} \tag{44}$$

$$\hat{p}_{\alpha h} = \frac{\sum_{k=1}^{m} u_{hk}\,\hat{p}_{k\alpha h}}{\sum_{k=1}^{m} u_{hk}} \tag{45}$$

and these values can be applied to calculate the estimates for $VaR_{\alpha h}$ and $CVaR_{\alpha h}$ exactly the same way as described in the previous subchapter.

3.5 Confidence intervals and testing

Regardless of the method used for deriving the estimates of the VaR and CVaR multipliers $q_{\alpha h}$ and $p_{\alpha h}$, the data set available for the computations is finite. Therefore, to be better informed to make any conclusions from the results, in case of both methods, confidence intervals are calculated for the multiplier values by bootstrapping.

Furthermore, to test the obtained estimates for the VaR and CVaR multipliers, the full data set is split into training and test set. For each value of h, the first 70% of values in $\mathbf{x}_h$ are assigned to the training set and the last 30% to the test set. As n, the length of the time series, varies for different values of h, the size of the training and test set is different for each value of h (see Figure 4).

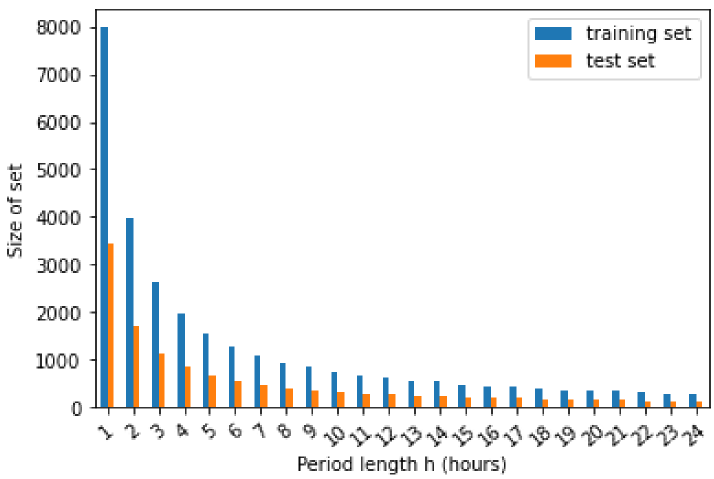

Figure 4: Training and test set sizes for different values of period length h.

3.5.1 Bootstrap confidence intervals

Using the training set, the following steps are taken to obtain 95% confidence intervals for $q_{\alpha h}$ and $p_{\alpha h}$, where $\alpha = 0.95$ and $h = 1, 2, 3, ..., 24$:

1. Sample values for both multiplier estimates $\hat{q}_{\alpha h}$ and $\hat{p}_{\alpha h}$ are calculated using the full training data set. These are used as the estimates for the true values of $q_{\alpha h}$ and $p_{\alpha h}$.

2. To calculate the 95% confidence intervals for the multipliers, we estimate the distributions of
$$\delta_{qh} = \hat{q}_{\alpha h} - q_{\alpha h} \tag{46}$$
and
$$\delta_{ph} = \hat{p}_{\alpha h} - p_{\alpha h} \tag{47}$$
by bootstrapping. Empirical bootstrap sample, that is, a resample from the training data set with the size equal to the training data set size n_{train} is obtained m times. For the empirical percentile method, the number of bootstrap samples taken was $m = 20\,000$ and for the Bayesian Model Averaging method $m = 75$. The latter value of m is much smaller due to restrictions on computation time.

   For each bootstrap sample, multiplier values $q^*_{j\alpha h}$ and $p^*_{j\alpha h}$ are estimated and their difference from the sample values $\hat{q}_{\alpha h}$ and $\hat{p}_{\alpha h}$ are computed. For each bootstrap sample j, where $j = 1, 2, ..., m$ the following values are calculated
$$\delta^*_{jqh} = q^*_{j\alpha h} - \hat{q}_{\alpha h} \tag{48}$$
$$\delta^*_{jph} = p^*_{j\alpha h} - \hat{p}_{\alpha h} \tag{49}$$
δ^*_{qh} and δ^*_{ph} are used as estimates of the distributions of δ_{qh} and δ_{ph}.

3. If the distributions of δ_{qh} and δ_{ph} were known, the 0.025 and 0.975 critical values for both $\delta_{qh,0.025}$, $\delta_{qh,0.975}$ and $\delta_{ph,0.025}$, $\delta_{ph,0.975}$ could be determined and the confidence intervals could be derived from
$$P(\delta_{qh,0.025} \leq \hat{q}_{\alpha h} - q_{\alpha h} \leq \delta_{qh,0.975} \mid q_{\alpha h}) = 0.95 \tag{50}$$
$$P(\delta_{ph,0.025} \leq \hat{p}_{\alpha h} - p_{\alpha h} \leq \delta_{ph,0.975} \mid p_{\alpha h}) = 0.95 \tag{51}$$
which can also be written as
$$P(\hat{q}_{\alpha h} - \delta_{qh,0.025} \geq q_{\alpha h} \geq \hat{q}_{\alpha h} + \delta_{qh,0.975} \mid q_{\alpha h}) = 0.95 \tag{52}$$
$$P(\hat{p}_{\alpha h} - \delta_{ph,0.025} \geq p_{\alpha h} \geq \hat{p}_{\alpha h} + \delta_{ph,0.975} \mid p_{\alpha h}) = 0.95. \tag{53}$$

With the bootstrap method, the critical values are approximated by $\delta^*_{qh,0.025}$, $\delta^*_{qh,0.975}$ and $\delta^*_{ph,0.025}$, $\delta^*_{ph,0.975}$ which are calculated from δ^*_{qh} and δ^*_{ph} as 2.5-percentile and 97.5-percentile of the respective data set of differences. As a result, the 95% bootstrap confidence intervals for $q_{\alpha h}$ and $p_{\alpha h}$ are obtained as

$$[\hat{q}_{\alpha h} - \delta^*_{qh,0.025},\ \hat{q}_{\alpha h} + \delta^*_{qh,0.975}] \tag{54}$$

$$[\hat{p}_{\alpha h} - \delta^*_{ph,0.025},\ \hat{p}_{\alpha h} + \delta^*_{ph,0.975}]. \tag{55}$$

(J. Orloff, 2014)

3.5.2 Performance metrics and testing

To test the computed bootstrap confidence interval estimates of $q_{\alpha h}$ and $p_{\alpha h}$ the following performance metrics are calculated respectively:

- **Percentage of times VaR is exceeded**

$$test_{qh} = \frac{\sum_{i=1}^{n_{test}} I_{q_{\alpha h}}(x_{ih})}{n_{test}} \cdot 100\%, \tag{56}$$

  where n_{test} is the size of the respective test set. As the multiplier values are calculated for 95% confidence level, the benchmark value for $test_{qh}$ is 5%.

- **Percentage deviance of mean tail ratio (given $q_{\alpha h}$ has been exceeded) from $p_{\alpha h}$**

$$test_{ph} = \frac{\sum_{i=1}^{n_{test}} \frac{x_{ih}+p_{\alpha h}}{p_{\alpha h}} I_{q_{\alpha h}}(x_{ih})}{\sum_{i=1}^{n_{test}} I_{q_{\alpha h}}(x_{ih})} \cdot 100\%. \tag{57}$$

  If $I_{q_{\alpha h}}(x_{ih}) = 1$ then $x_{ih} < 0$, and as $p_{\alpha h} > 0$, then negative values of $test_{ph}$ indicate that the mean tail loss given VaR level was exceeded is higher than the CVaR threshold and vice versa. The benchmark value for $test_{ph}$ is 0%.

Similarly to the training set, the test set is also finite. Therefore, the same bootstrapping method as described in subchapter 3.5.1 for computing the confidence interval estimates for VaR and CVaR multipliers is applied to obtain confidence intervals for $test_{qh}$ and $test_{ph}$. For both performance metrics, their 95% bootstrap confidence interval is calculated twice, firstly for $test_{qh,L}$ and $test_{ph,L}$ given the lower bound values of the confidence interval of the respective VaR or CVaR multipliers

$$\hat{q}_{\alpha h,L} = \hat{q}_{\alpha h} - \delta^*_{qh,0.025} \tag{58}$$

$$\hat{p}_{\alpha h,L} = \hat{p}_{\alpha h} - \delta^*_{ph,0.025} \tag{59}$$

and secondly for $test_{qh,U}$ and $test_{ph,U}$ given the upper bound values of the confidence interval of the respective VaR or CVaR multipliers

$$\hat{q}_{\alpha h,U} = \hat{q}_{\alpha h} + \delta^*_{qh,0.975} \tag{60}$$

$$\hat{p}_{\alpha h,U} = \hat{p}_{\alpha h} + \delta^*_{ph,0.975}. \tag{61}$$

In more detail, the following steps are conducted:

1. Using the full test set, sample values for both multipliers' performance metrics' estimates given the upper and lower bounds of the confidence intervals of the VaR and CVaR multipliers are computed. The following estimates are obtained

$$\widehat{test}_{qh,L} = \frac{\sum_{i=1}^{n_{test}} I_{q_{\alpha h,L}}(x_{ih})}{n_{test}} \cdot 100\% \tag{62}$$

$$\widehat{test}_{ph,L} = \frac{\sum_{i=1}^{n_{test}} \frac{x_{ih}+\hat{p}_{\alpha h,L}}{\hat{p}_{\alpha h,L}} I_{q_{\alpha h,L}}(x_{ih})}{\sum_{i=1}^{n_{test}} I_{q_{\alpha h,L}}(x_{ih})} \cdot 100\% \tag{63}$$

   and

$$\widehat{test}_{qh,U} = \frac{\sum_{i=1}^{n_{test}} I_{q_{\alpha h,U}}(x_{ih})}{n_{test}} \cdot 100\% \tag{64}$$

$$\widehat{test}_{ph,U} = \frac{\sum_{i=1}^{n_{test}} \frac{x_{ih}+\hat{p}_{\alpha h,U}}{\hat{p}_{\alpha h,U}} I_{q_{\alpha h,U}}(x_{ih})}{\sum_{i=1}^{n_{test}} I_{q_{\alpha h,U}}(x_{ih})} \cdot 100\%, \tag{65}$$

   where

$$I_{q_{\alpha h,L}}(x_{ih}) := \begin{cases} 1 & \text{if } x_{ih} < -\hat{q}_{\alpha h,L} \\ 0 & \text{if } x_{ih} \geq -\hat{q}_{\alpha h,L} \end{cases}, \tag{66}$$

$$I_{q_{\alpha h,U}}(x_{ih}) := \begin{cases} 1 & \text{if } x_{ih} < -\hat{q}_{\alpha h,U} \\ 0 & \text{if } x_{ih} \geq -\hat{q}_{\alpha h,U} \end{cases}. \tag{67}$$

   These are used as the estimates for the true values of $test_{qh,L}$, $test_{ph,L}$, $test_{qh,U}$ and $test_{ph,U}$ given the lower and upper bounds of the VaR and CVaR multipliers' estimates respectively.

2. To calculate the 95% confidence intervals for the performance metrics, we estimate the distributions of

$$\delta_{test_{qh,L}} = \widehat{test}_{qh,L} - test_{qh,L} \tag{68}$$

$$\delta_{test_{ph,L}} = \widehat{test}_{ph,L} - test_{ph,L} \tag{69}$$

$$\delta_{test_{qh,U}} = \widehat{test}_{qh,U} - test_{qh,U} \tag{70}$$

$$\delta_{test_{ph,U}} = \widehat{test}_{ph,U} - test_{ph,U} \tag{71}$$

by bootstrapping. The number of bootstrapping samples m taken remains the same as used when bootstrapping for VaR and CVaR multipliers' confidence intervals.

For each bootstrap sample, multiplier values $test^*_{jqh,L}$, $test^*_{jph,L}$, $test^*_{jqh,U}$ and $test^*_{jph,U}$ are estimated and their difference from the sample values obtained in the precious step are computed. For each bootstrap sample j, where $j = 1, 2, ..., m$ the following values are calculated

$$\delta^*_{test_{jqh,L}} = test^*_{jqh,L} - \widehat{test}_{qh,L} \tag{72}$$

$$\delta^*_{test_{jph,L}} = test^*_{jph,L} - \widehat{test}_{ph,L} \tag{73}$$

$$\delta^*_{test_{jqh,U}} = test^*_{jqh,U} - \widehat{test}_{qh,U} \tag{74}$$

$$\delta^*_{test_{jph,U}} = test^*_{jph,U} - \widehat{test}_{ph,U} \tag{75}$$

and used as estimates of the distributions of $\delta_{test_{qh,L}}$, $\delta_{test_{ph,L}}$, $\delta_{test_{qh,U}}$ and $\delta_{test_{ph,U}}$ respectively.

3. If the distributions of $\delta_{test_{qh,L}}$, $\delta_{test_{ph,L}}$, $\delta_{test_{qh,U}}$ and $\delta_{test_{ph,U}}$ were known, the 0.025 and 0.975 critical values for all four could be determined and the confidence intervals could be derived from

$$P(\delta_{test_{qh,L},0.025} \leq \widehat{test}_{qh,L} - test_{qh,L} \leq \delta_{test_{qh,L},0.975} \mid test_{qh,L}) = 0.95 \tag{76}$$

$$P(\delta_{test_{ph,L},0.025} \leq \widehat{test}_{ph,L} - test_{ph,L} \leq \delta_{test_{ph,L},0.975} \mid test_{ph,L}) = 0.95 \tag{77}$$

$$P(\delta_{test_{qh,U},0.025} \leq \widehat{test}_{qh,U} - test_{qh,U} \leq \delta_{test_{qh,U},0.975} \mid test_{qh,U}) = 0.95 \tag{78}$$

$$P(\delta_{test_{ph,U},0.025} \leq \widehat{test}_{ph,U} - test_{ph,U} \leq \delta_{test_{ph,U},0.975} \mid test_{ph,U}) = 0.95. \tag{79}$$

With the bootstrap method, the critical values are approximated the same way as described in the previous subchapter. As a result, the 95% bootstrap confidence intervals for $test_{qh,L}$, $test_{ph,L}$, $test_{qh,U}$ and $test_{ph,U}$ are obtained as

$$[\widehat{test}_{qh,L} - \delta^*_{test_{qh,L},0.025},\ \widehat{test}_{qh,L} + \delta^*_{test_{qh,L},0.975}] \tag{80}$$

$$[\widehat{test}_{ph,L} - \delta^*_{test_{ph,L},0.025},\ \widehat{test}_{ph,L} + \delta^*_{test_{ph,L},0.975}] \tag{81}$$

$$[\widehat{test}_{qh,U} - \delta^*_{test_{qh,U},0.025},\ \widehat{test}_{qh,U} + \delta^*_{test_{qh,U},0.975}] \tag{82}$$

$$[\widehat{test}_{ph,U} - \delta^*_{test_{ph,U},0.025},\ \widehat{test}_{ph,U} + \delta^*_{test_{ph,U},0.975}]. \tag{83}$$

We have now obtained 95% bootstrap confidence intervals for performance metrics separately given the lower and upper bound values of the 95% bootstrap confidence intervals of VaR and CVaR multipliers. However, we would like to obtain a single bootstrap confidence interval for both performance metrics, given the 95% bootstrap confidence intervals of VaR and CVaR multipliers.

For any value $q_{\alpha h}$ of the VaR multiplier such that $\hat{q}_{\alpha h,L} \leq q_{\alpha h} \leq \hat{q}_{\alpha h,U}$, the respective performance metric's $test_{qh}$ confidence interval obtained from the same test set should have a lower bound not smaller than $\widehat{test}_{qh,U} - \delta^*_{test_{qh,U},0.025}$ and an upper bound not larger than $\widehat{test}_{qh,L} + \delta^*_{test_{qh,L},0.975}$. Therefore, the confidence interval of $test_{qh}$ obtained from the same test set given $\hat{q}_{\alpha h,L} \leq q_{\alpha h} \leq \hat{q}_{\alpha h,U}$ should fall into the following interval

$$[\widehat{test}_{qh,U} - \delta^*_{test_{qh,U},0.025},\ \widehat{test}_{qh,L} + \delta^*_{test_{qh,L},0.975}]. \tag{84}$$

Similarly, for any value $p_{\alpha h}$ of the CVaR multiplier such that $\hat{p}_{\alpha h,L} \leq p_{\alpha h} \leq \hat{p}_{\alpha h,U}$, the respective performance metric's $test_{ph}$ confidence interval obtained from the same test set should have a lower bound not smaller than $\widehat{test}_{ph,L} - \delta^*_{test_{ph,L},0.025}$ and an upper bound not larger than $\widehat{test}_{ph,U} + \delta^*_{test_{ph,U},0.975}$. Therefore, the confidence interval of $test_{ph}$ obtained from the same test set given $\hat{p}_{\alpha h,L} \leq p_{\alpha h} \leq \hat{p}_{\alpha h,U}$ should fall into the following interval

$$[\widehat{test}_{ph,L} - \delta^*_{test_{ph,L},0.025},\ \widehat{test}_{ph,U} + \delta^*_{test_{ph,U},0.975}]. \tag{85}$$

Conclusively, we can say that equations 84 and 85 outline the bootstrap confidence intervals for $test_{qh}$ and $test_{ph}$ given the 95% bootstrap confidence intervals for $q_{\alpha h}$ and $p_{\alpha h}$.

3.6 Results and analysis

This subchapter present the results of the empirical validation of the current VaR model and the estimation of confidence intervals for the VaR and CVaR multipliers using the two previously described methods. Furthermore, the performance metrics of the obtained VaR and CVaR multipliers are outlined. Lastly, the results of the empirical percentile method and the Bayesian Model Averaging method are compared. The source code used for computing the results is attached in Appendix 4. The numeric results are included in Appendix 3.

3.6.1 Validation of the current model

The current VaR model assumes the FX gain or loss variable R_{FX} to be normally distributed with zero mean and standard deviation of the log returns of the portfolio. Therefore, the current model assumes the ratios $\mathbf{x}_h$ to come from a standard normal distribution. Under this assumption, the current model's VaR and CVaR multipliers would be equal to

$$q_{\alpha h} = -\Phi^{-1}(0.05) \approx 1.645 \ \ \forall h, \tag{86}$$

where $\Phi^{-1}(\beta)$ is the inverse cumulative distribution function of standard normal, and

$$p_{\alpha h} = \frac{1}{0.05} \int_{q_{\alpha h}}^{\infty} y f_Z(y) dy \approx 2.063 \ \ \forall h, \tag{87}$$

where $f_Z(y)$ is the probability density function of standard normal.

These values of $q_{\alpha h}$ and $p_{\alpha h}$ derived from the normality assumption were tested on the full data set available, as there was no need for a separate training set. 95% bootstrap confidence intervals were obtained for the respective performance metrics $test_{qh}$ and $test_{ph}$. Number of bootstrap samples taken was $n = 20\,000$.

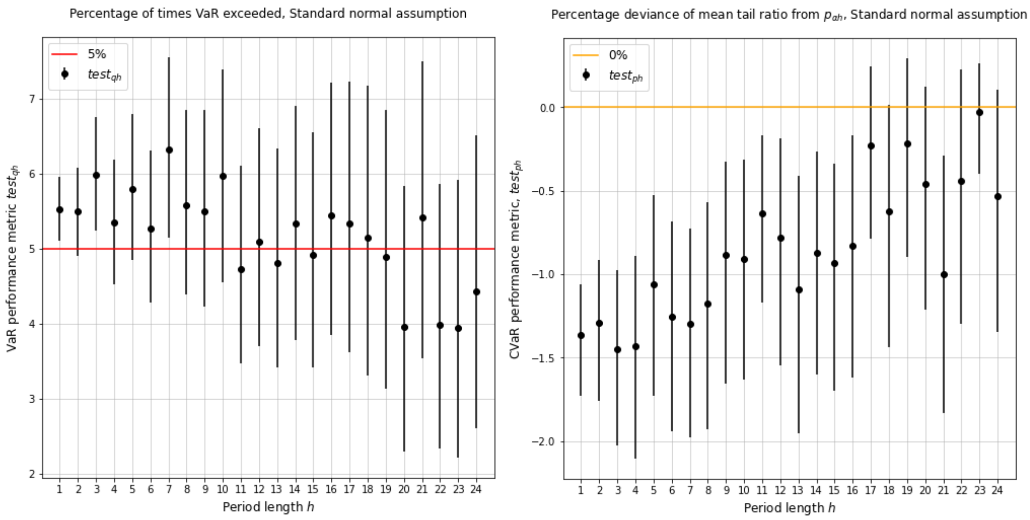

Figure 5: Standard normal VaR and CVaR multipliers' performance metrics' $test_{qh}$ and $test_{ph}$ 95% bootstrap confidence intervals.

On the left hand side graph of Figure 5, the respective $test_{qh}$ values are outlined, with the red line denoting the 5% threshold value. It can be noted that for most values of h, taking $q_{\alpha h} \approx 1.645$ would result in the benchmark 5% value of percentage of times VaR exceeded to fall into the 95% confidence interval of the respective performance metric. There are a couple of exceptions to this for lower values of h, where with 95% probability VaR would be exceeded more than 5% of times. However, for the current VaR model of the company, where the time horizon is taken to be one day ($h = 24$), the standard normal assumption can not be proven to be inadequate for VaR calculation.

On the right hand side graph of Figure 5, the CVaR multiplier performance metric $test_{ph}$ values are outlined, with the orange line denoting the 0% threshold value. One can observe that for most values of h taking $p_{\alpha h} \approx 2.063$ would result in underestimating the risk given VaR is exceeded. However, for the current VaR model of the company, where the time horizon is taken to be one day ($h = 24$), the standard normal assumption can not be proven to be inadequate for CVaR calculation.

3.6.2 Empirical percentile method results

The VaR multiplier $q_{\alpha h}$ 95% confidence interval estimates for $\alpha = 0.95$ obtained with the empirical percentile method and bootstrapping are outlined on the left hand side of Figure 6. The red line on the graph depicts the VaR multiplier value of the current VaR model used in the company, derived from the standard normal assumption. It can be noted that the VaR multiplier value stemming from the standard normal assumption falls into the confidence intervals of $q_{\alpha h}$ for most values of h, except for $h = 1$ and $h = 20$, where $q_{\alpha h}$ estimate is respectively slightly higher and slightly lower than $-\Phi^{-1}(0.05)$. Although VaR multiplier estimates for higher values of h seem to be slightly lower, no clear relationship between h and $q_{\alpha h}$ can be detected.

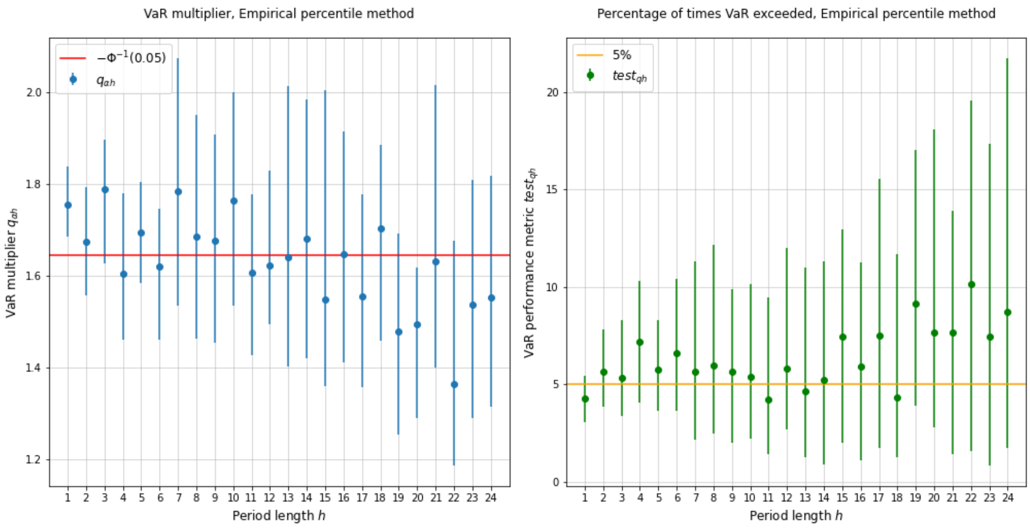

Figure 6: VaR multiplier $q_{\alpha h}$ and the respective performance metric $test_{qh}$ confidence intervals obtained with the empirical percentile method and bootstrapping.

The empirical percentile method VaR multiplier performance metric $test_{qh}$ confidence interval estimates obtained with bootstrapping are outlined on the right hand side of Figure 6. One can observe the 5% threshold for the percentage of times VaR has been exceeded to fall into the range of the performance metric confidence intervals. However, the confidence intervals for $test_{qh}$ are fairly large for higher values of h spanning from 1.7% to 21.7% for $h = 24$. Considering the significantly lower size of training and test sets for higher h values, the confidence intervals are expected to be wider.

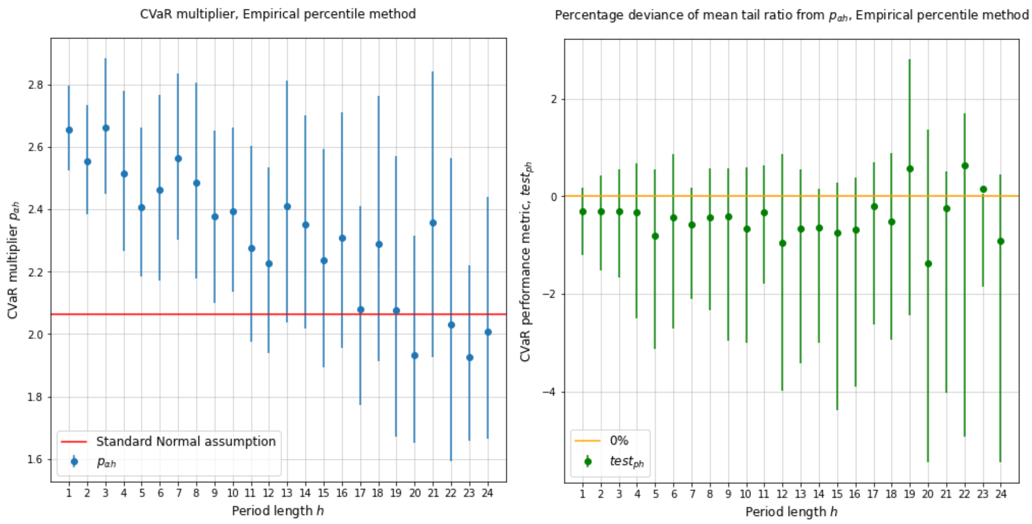

Figure 7: CVaR multiplier $p_{\alpha h}$ and the respective performance metric $test_{ph}$ confidence intervals obtained with the empirical percentile method and bootstrapping.

The left hand side graph on Figure 7 demonstrates the CVaR multiplier $p_{\alpha h}$ 95% confidence interval estimates for $\alpha = 0.95$ obtained with the empirical percentile method and bootstrapping. The CVaR multiplier values decrease with the increase in h. This is in line with Figure 3 where larger extreme values of $\mathbf{x}_h$ were noted for smaller values of h. For $h \leq 10$, the CVaR multiplier 95% confidence intervals do not include the 95% CVaR multiplier value that the standard normal assumption would suggest and are higher than that. This is in line with the validation results of the current model's standard normal assumption.

The empirical percentile method CVaR multiplier performance metric $test_{ph}$ bootstrap confidence interval estimates are illustrated on the right hand side of Figure 7. One can observe the 0% benchmark for the percentage deviance of mean tail ratio from $p_{\alpha h}$ given VaR has been exceeded to fall into the range of the performance metric confidence intervals. Confidence intervals for $test_{ph}$ also increase with the increase of h similarly to the confidence intervals of $test_{qh}$.

3.6.3 Bayesian Model Averaging method results

Before computing the bootstrapping confidence intervals for the VaR and CVaR multipliers with the Bayesian Model Averaging method, the multiplier values were calculated for all period length values $h = 1, 2, 3, ..., 24$ and all $m = 21$ candidate models using the training set. The goodness of fit of the fitted distributions to the $\mathbf{x}_h$ ratios data set was tested on the test set with the Kolmogorov-Smirnov test using the `scipy.stats.kstest` method.

Figure 13 in Appendix 2 illustrates the VaR-CVaR multiplier scatter plots for all values of h, where each dot represents one of the fitted distributions. The scatter plot markers are scaled according to the Kolmogorov-Smirnov test p-values such that a higher p-value would result in a larger marker size. Furthermore, the color of the marker indicates whether the p-value is over or under 0.05. Therefore, for distributions represented by blue markers, the Kolmogorov-Smirnov test rejects the null hypothesis of the data set coming from the fitted distribution with 95% probability. For most period lengths there are several distributions for which the null hypothesis of the Kolmogorov-Smirnov test is not rejected, with the exception of $h = 1, 23, 24$. Therefore, for the majority of period lengths at least some of the candidate distributions would be suitable to describe the ratios $\mathbf{x}_h$.

Figures 8 and 9 illustrate the 95% bootstrap confidence intervals for the VaR and CVaR multipliers $q_{\alpha h}$ and $p_{\alpha h}$ and their respective performance metrics $test_{qh}$ and $test_{ph}$ confidence intervals obtained with the Bayesian Model Averaging method. Similarly to the empirical percentile method, CVaR multiplier values decrease with the increase of h, but no clear relationship between the VaR multiplier $q_{\alpha h}$ and period length h can be observed. The performance metrics' benchmark values 5% and 0% respectively for $test_{qh}$ and $test_{ph}$ fall into the bootstrap confidence intervals of the performance metrics. Confidence intervals for both, VaR and CVaR multipliers and their performance metrics tend to be wider for larger values of h, which can be expected, considering the decrease of available data points for training and testing with the increase of h.

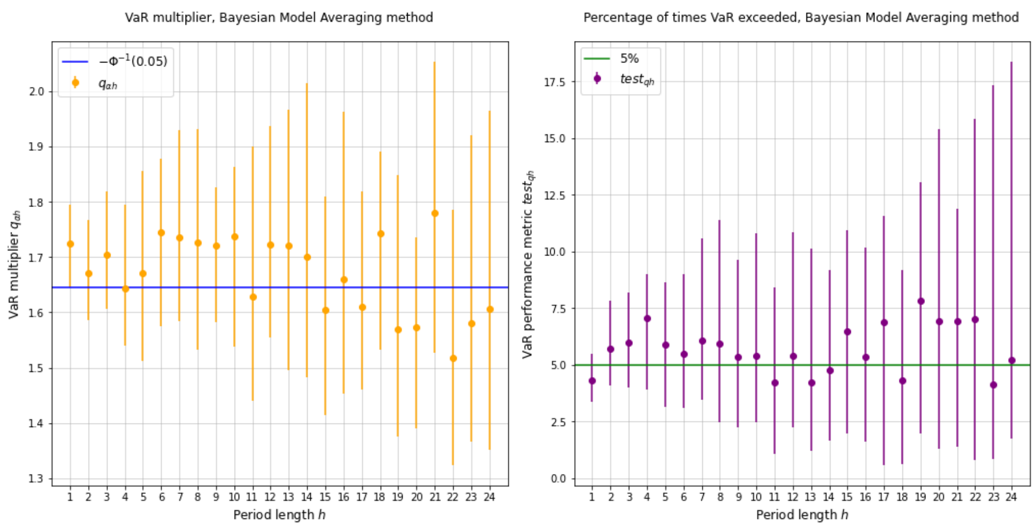

Figure 8: VaR multiplier $q_{\alpha h}$ and the respective performance metric $test_{qh}$ confidence intervals obtained with the Bayesian Model Averaging method and bootstrapping.

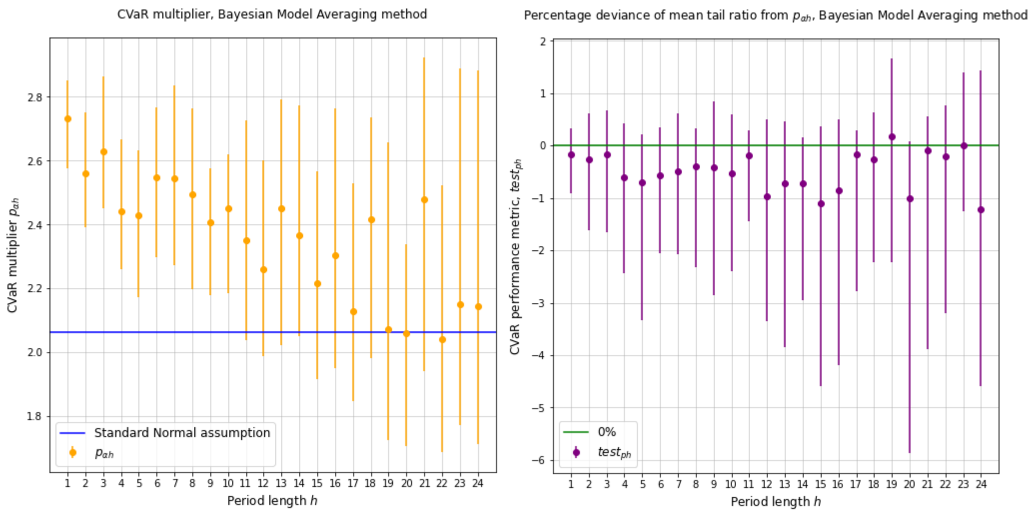

Figure 9: CVaR multiplier $p_{\alpha h}$ and the respective performance metric $test_{ph}$ confidence intervals obtained with the Bayesian Model Averaging method and bootstrapping.

3.6.4 Discussion

In this subchapter, the empirical percentile and Bayesian Model Averaging methods' results are compared against each other and the current VaR model normality assumption. In addition, the VaR and CVaR multipliers' performance metrics' results are compared for the three.

Figure 10 outlines the above-mentioned comparison for VaR multiplier and its performance metric. It can be concluded that the current model's VaR multiplier $q_{\alpha h} = -\Phi^{-1}(0.05) \approx 1.645$ falls into the range of the $q_{\alpha h}$ 95% bootstrap confidence intervals for both methods, except for minor exceptions of $h = 1$ and $h = 20$. Furthermore, for all three options of the VaR multiplier values, the performance metric $test_{qh}$ bootstrap confidence intervals include the expected benchmark value of 5% as the percentage of times VaR is exceeded. There are, however, slight exceptions to this for the current model's VaR multiplier and lower values of h. Therefore, it can be concluded that the current VaR model's normality assumption and respective VaR multiplier are suitable for calculating the daily ($h = 24$) 95% Value at Risk level. If the company decided to monitor the VaR level for shorter time horizons, the current model assumptions would still be suitable.

Figure 11 outlines the comparison for CVaR multiplier and its performance metric. It can be concluded that the current model's CVaR multiplier $p_{\alpha h} \approx 2.063$ falls into the range of the $p_{\alpha h}$ 95% bootstrap confidence intervals for both methods only for period length values $h \geq 11$. However, the performance metric $test_{ph}$ bootstrap confidence intervals indicate that the current model would under-estimate risk for

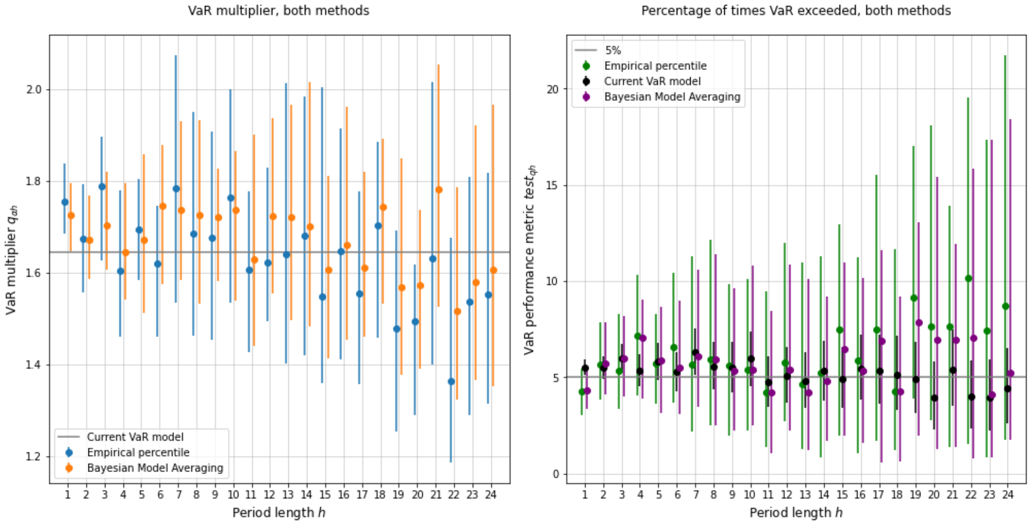

Figure 10: VaR multiplier $q_{\alpha h}$ and the respective performance metric $test_{qh}$ bootstrap confidence intervals with the empirical percentile and Bayesian Model Averaging methods.

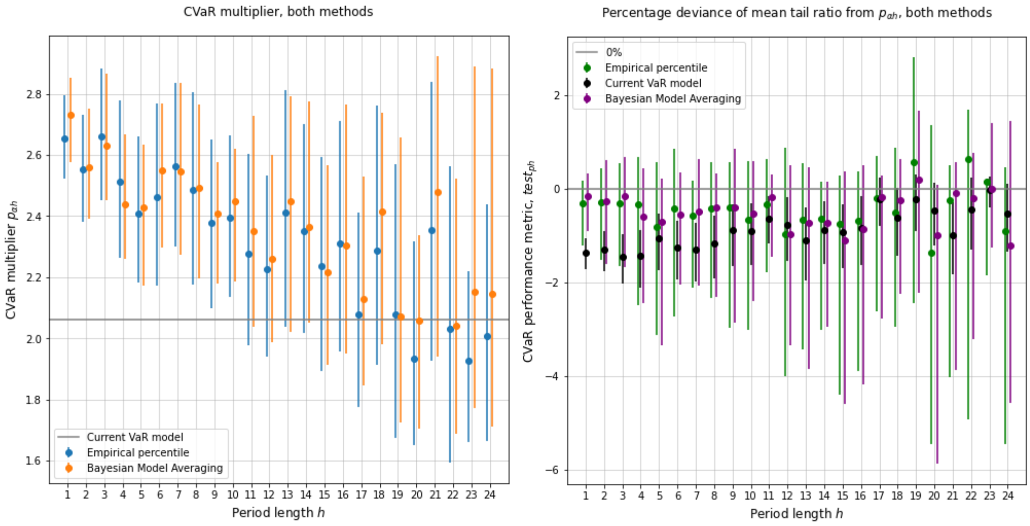

Figure 11: CVaR multiplier $p_{\alpha h}$ and the respective performance metric $test_{ph}$ bootstrap confidence intervals with the empirical percentile and Bayesian Model Averaging methods.

time horizons $h \leq 16$. CVaR multiplier values $p_{\alpha h}$ decrease with the increase of h for both of the applied calculation methods. Proposing higher CVaR multiplier values for shorter time horizons than the current model, the two methods outputs cannot be said to under-estimate the tail risk as the current model would. The performance metric $test_{ph}$ bootstrap confidence intervals for both methods include the benchmark value 0% which corresponds to no deviance of mean tail ratio from CVaR multiplier value, given VaR is exceeded. To conclude, the current model's assumptions would be suitable for calculating the daily 95% CVaR, but would under-estimate the tail risk for most of the shorter time horizons. If the company would like to monitor CVaR levels for shorter time horizons, calculating the CVaR multiplier with either empirical percentile or Bayesian Model Averaging method would be suitable.

Lastly, one can observe the results of the two applied methods to be rather similar for both, VaR and CVaR multipliers, with minor exceptions. For example, CVaR multiplier's confidence intervals are significantly larger for BMA method and $h = 23, 24$. To arrive at narrower confidence intervals, calculations could be conducted with larger data sets. The results of this thesis also demonstrate narrower confidence intervals for shorter time horizons where training and test sets are respectively larger. Furthermore, the Bayesian Model Averaging method could be further developed to incorporate the 'Occam factor' (described in subchapter 3.4) into the calculations. In addition, alternative methods, such as Monte-Carlo simulations could be conducted and compared to the methods applied in this thesis. Lastly, the Value at Risk model could be developed further to, for example, take carry interest costs also into consideration.

Conclusions

The purpose of this thesis is to validate the current Value at Risk model used in the global FinTech company Wise with historical data; propose and test confidence interval estimates for the 95% VaR and CVaR multipliers with two different methods: empirical percentile method and Bayesian Model Averaging method for time horizons of 1 to 24 hours; compare the results of the current model validation and the two applied methods; and analyse whether the choice of the time horizon has an impact on the VaR and CVaR multiplier values.

In the first chapter, an introductory overview of the foreign exchange market was given, describing its history, main participants and instrument types and potential gains and losses involved. In the second chapter, some of the most common risk metrics used for measuring market risk were defined, together with Value at Risk (VaR) and Conditional Value at Risk (CVaR) which are in the focus of this thesis.

In the third chapter, the global FinTech company Wise, its business model, currency portfolio and VaR model were first introduced. Next, the data sets used in this thesis were described and the main variable used in further calculations — the ratio of the realised risk (FX gain or loss) to the estimated risk (portfolio standard deviation estimate) was introduced. Then, the two methods for estimating VaR and CVaR multipliers were outlined: the empirical percentile method and Bayesian Model Averaging. Furthermore, bootstrapping method for obtaining confidence interval estimates for VaR and CVaR multipliers was described. In addition, the respective performance metrics for the two multipliers were introduced: percentage of times VaR is exceeded, and percentage deviance of mean tail ratio from CVaR multiplier, given VaR has been exceeded. Lastly, the results of validation, estimation and testing were presented, analysed and compared.

This thesis found that the current VaR model's assumptions can not be proven to be inadequate for calculating the daily 95% VaR, as the current models' empirical validation results show that the benchmark value 5% of percentage of times VaR exceeded falls into the confidence interval of the VaR performance metric. Furthermore, current model would also be suitable for calculating 95% VaR for most shorter time horizons. In addition, current model did not prove to be inadequate for calculating the daily 95% CVaR level. However, it would under-estimate risk for most of the shorter time horizons.

The empirical percentile method and Bayesian Model Averaging method yielded relatively similar results for both, VaR and CVaR multipliers' 95% confidence interval estimates. While VaR multiplier values did not have a clear relation to the time horizon length, CVaR multiplier estimates decreased with the increase of the length of the time horizon. In case of both methods, multipliers' and their performance metrics' confidence intervals widened with the increase of the time horizon, as the training and test sets' size decreased respectively.

Both methods used for estimating VaR and CVaR multipliers proved to be suitable, as their respective performance metrics confidence intervals included the performance metric benchmark values for most time horizon lengths respectively for VaR and CVaR: 5% of times VaR is exceeded, and 0% deviance of mean tail ratio from CVaR multiplier, given VaR is exceeded. However, to arrive at narrower confidence intervals, calculations could be conducted with larger data sets. Lastly, Bayesian Model Averaging method could be developed further to include the 'Occam factor', and additional alternative methods, such as Monte-Carlo simulations could be applied and compared to the results of this thesis.

References

A. H. Meltzer, M. Friedman (2020). *Money. The decline of gold.* URL: https://www.britannica.com/topic/money/The-decline-of-gold (visited on 03/01/2021).

AICPA (2018). *Audit and Accounting Guide: Investment Companies.* URL: https://books.google.ee/books?id=0DxxDwAAQBAJ&pg=PA529&dq=accounting+fx+gains+and+losses+unrealized+realized&hl=en&sa=X&ved=2ahUKEwjKutSSj5TvAhUR_6QKHZ48A1wQ6AEwAXoECAAQAg#v=onepage&q=accounting%20fx%20gains%20and%20losses%20unrealized%20realized&f=false (visited on 03/03/2021).

Amadeo, K. (2020). *Foreign Exchange Market.* URL: https://www.thebalance.com/foreign-exchange-market-3306088 (visited on 03/02/2021).

BIS (2019). *Foreign exchange turnover in April 2019.* URL: https://www.bis.org/statistics/rpfx19_fx.htm#:~:text=Highlights,%245.1%20trillion%20three%20years%20earlier.&text=Growth%20of%20FX%20derivatives%20trading,outpaced%20that%20of%20spot%20trading. (visited on 03/01/2021).

Bradfield, D. (2018). *The History of Forex.* URL: https://www.dailyfx.com/education/beginner/history-of-forex.html (visited on 03/01/2021).

C. Filippi G. Guastaroba, M.G. Speranza (2020). "Conditional value-at-risk beyond finance: a survey". In: *International Transactions in Operational Research* 27.3.

Carver, R. (2019). *Leveraged Trading: A professional approach to trading FX, stocks on margin, CFDs, spread bets and futures for all traders.* 2nd ed. Hampshire, Great Britain: Harriman House Ltd.

Coppola, F. (2018). *Forex (FX) History: How the Modern FX Market was Born.* URL: https://www.americanexpress.com/us/foreign-exchange/articles/forex-fx-history/ (visited on 03/01/2021).

Cotter, J. (2005). "Tail Behaviour of the Euro". In: *Applied Economics* 37.

Coyle, B. (2000). *Foreign Exchange Markets.* Kent: Financial World Publishing.

Döhring, Björn (Jan. 2008). "Hedging and invoicing strategies to reduce exchange rate exposure: a euro-area perspective". In: *Economic Papers* 299.2, pp. 1–5.

E. J. Elton M. J. Gruber, S. J. Brown W. N. Goetzmann (2009). *Modern Portfolio Theory and Investment Analysis.* 8th ed. USA: John Wiley and Sons, Inc.

Fieldhouse, S. (2012). *FT Guide to Foreign Exchange Trading.* 1st ed. Harlow: Pearson Education Limited.

FXPA, Foreign Exchange Professionals Association (2013). *The Modern Foreign Exchange Market*. URL: https://fxpa.org/wp-content/uploads/2015/09/fxpa-overview-7-15.pdf (visited on 03/01/2021).

Hardy, M. R. (2006). "An introduction to risk measures for actuarial applications". URL: https://www.casact.org/library/studynotes/hardy4.pdf.

Hudson, D. (2013). *Foreign exchange market*. URL: https://www.britannica.com/topic/foreign-exchange-market (visited on 02/21/2021).

IAS (2003). *IAS 21 The Effects of Changes in Foreign Exchange Rates*. URL: https://www.ifrs.org/issued-standards/list-of-standards/ias-21-the-effects-of-changes-in-foreign-exchange-rates/ (visited on 03/03/2021).

J. Orloff, J. Bloom (2014). "Bootstrap confidence intervals". URL: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading24.pdf.

Jorion, P. (2007). *Value at Risk. The new benchmark for managing financial risk*. 3rd ed. The McGraw-Hill Companies, Inc.

Just (2021). *FX Instruments*. URL: https://www.gojust.com/faq/fx-instruments (visited on 03/02/2021).

Kantox (2021). *FX Transaction Costs*. URL: https://www.kantox.com/en/glossary/fx-transaction-costs/ (visited on 03/03/2021).

Keppler, M. (Nov. 1990). "Risk is Not The Same as Volatility". In: *Die Bank* 11, pp. 610–614.

Kolakowski, M. (2020). *Bloomberg vs. Reuters: What's the Difference?* URL: https://www.investopedia.com/articles/investing/052815/financial-news-comparison-bloomberg-vs-reuters.asp (visited on 03/01/2021).

Kotze, A. (2005). "Stock Price Volatility: a primer". URL: https://www.researchgate.net/publication/236169214_Stock_Price_Volatility_a_primer.

Käärik, Meelis (2019). "Non-Life Insurance Mathematics". Lecture Notes.

L. H. Ederington, W. Guan (2006). "Measuring Historical Volatility". In: *Journal of Applied Finance* 16.1.

L. Lyons, C. Notger (2005). "Volatility and its Measurements: The Design of a Volatility Index and the Execution of its Historical Time Series at the DEUTSCHE BÖRSE AG". URL: https://www.eurexchange.com/resource/blob/43140/47ca53f0178cec31caeecdf94cc18f6e/data/volatility_and_its_measurements.pdf.pdf.

Lehman. *Foreign Exchange Training Manual*. URL: https://web.stanford.edu/~jbulow/Lehmandocs/docs/BARCLAYS/LBEX-LL%203356480-3356609.pdf (visited on 03/02/2021).

M. M. Dacorogna R. Gençay, U. A. Müller R. B. Olsen O. V. Pictet (2001). *An Introduction to High-Frequency Finance*. Academic Press.
MacKay, D. J.C. (2003). *Information Theory, Inference, and Learning Algorithms*. 1st ed. Cambridge University Press.
Markets, Shift (2019). *Forex Market Overview*. URL: https://www.nasdaq.com/articles/forex-market-overview-2019-06-07 (visited on 03/02/2021).
Neftci, S. N. (2008). *Principles of Financial Engineering*. 2nd ed. Canada: Academic Press.
Outreville, J. François (1998). *Theory and Practice of Insurance. The Meaning of Risk*. Boston: Springer.
P. Pontikis A. Douglas, L. Lovrencic (2012). *FX Trading: A Guide to Trading Foreign Exchange*. 2nd ed. Milton: Wrightbooks.
Parker, P. J.L. (2018). *Treasury Operations Handbook. Foreign Exchange*. 3rd ed. London: Multimedia TradeWind Limited.
R. Zhang M. Aarons, G. Loeper (2019). *Optimal FX Hedge Tenor with Liquidity Risk*. URL: https://arxiv.org/ftp/arxiv/papers/1903/1903.06346.pdf (visited on 03/03/2021).
Righi, M. B. (2019). "A composition between risk and deviation measures". In: *Annals of Operations Research*.
Russell, J. (2021). *The Role of a Forex Brokerage*. URL: https://www.thebalance.com/what-is-a-forex-brokerage-1344933 (visited on 03/01/2021).
S. Manganelli, R. F. Engle (Aug. 2001). "Value at Risk Models in Finance". In: *European Central Bank Working Paper Series* 2, pp. 6–7.
S. Uryasev, R. T. Rockafellar (1999). "Optimization of Conditional Value-at-Risk". URL: http://janroman.dhis.org/finance/VaR/cvar2.pdf.
SciPy (2021). *Statistical functions (scipy.stats)*. Version 1.6.3. URL: https://docs.scipy.org/doc/scipy/reference/stats.html (visited on 05/16/2021).
Segal, T. (2020). *Forex Market: Who Trades Currencies and Why*. URL: https://www.investopedia.com/articles/forex/11/who-trades-forex-and-why.asp (visited on 03/01/2021).
Sumnicht, V. (2009). "Practical Applications of Post-Modern Portfolio Theory". URL: https://isectors.com/wp-content/uploads/2013/08/Standard-Deviation-as-Risk-Measure.pdf.
Venketas, W. (2019). *Forex Market Size: A Trader's Advantage*. URL: https://www.dailyfx.com/education/beginner/forex-market-size.html (visited on 03/01/2021).
Walmsley, J. (2000). *The Foreign Exchange and Money Markets Guide. Foreign Exchange Calculations*. 2nd ed. New York: John Wiley Sons, Inc.

Appendix 1. Histograms of $\mathbf{x}_h$

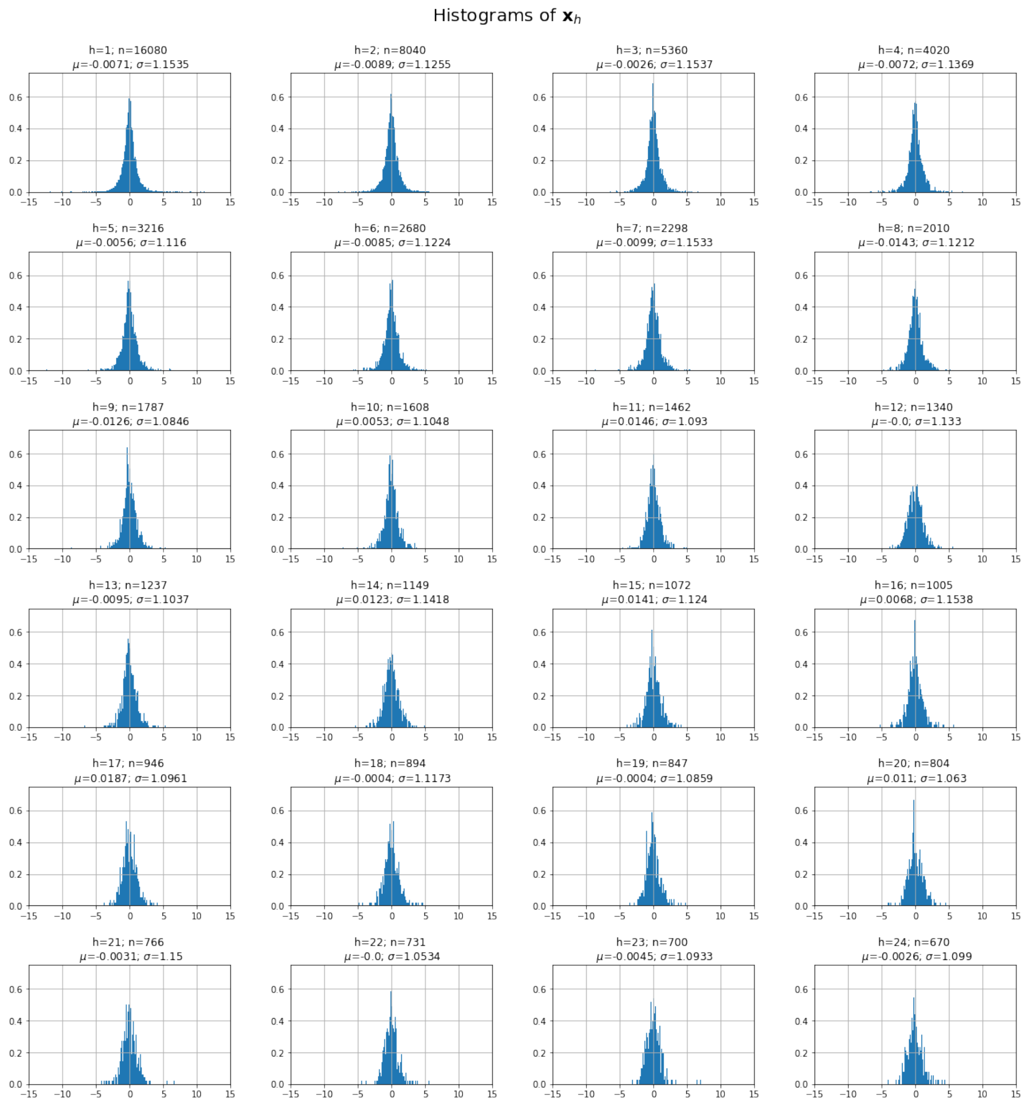

Figure 12: Histograms of $\mathbf{x}_h$, the ratios of realised risk to the estimated standard deviation of the risk for $h = 1, 2, 3, ..., 24$.

Appendix 2. VaR-CVaR multiplier scatter plots, BMA

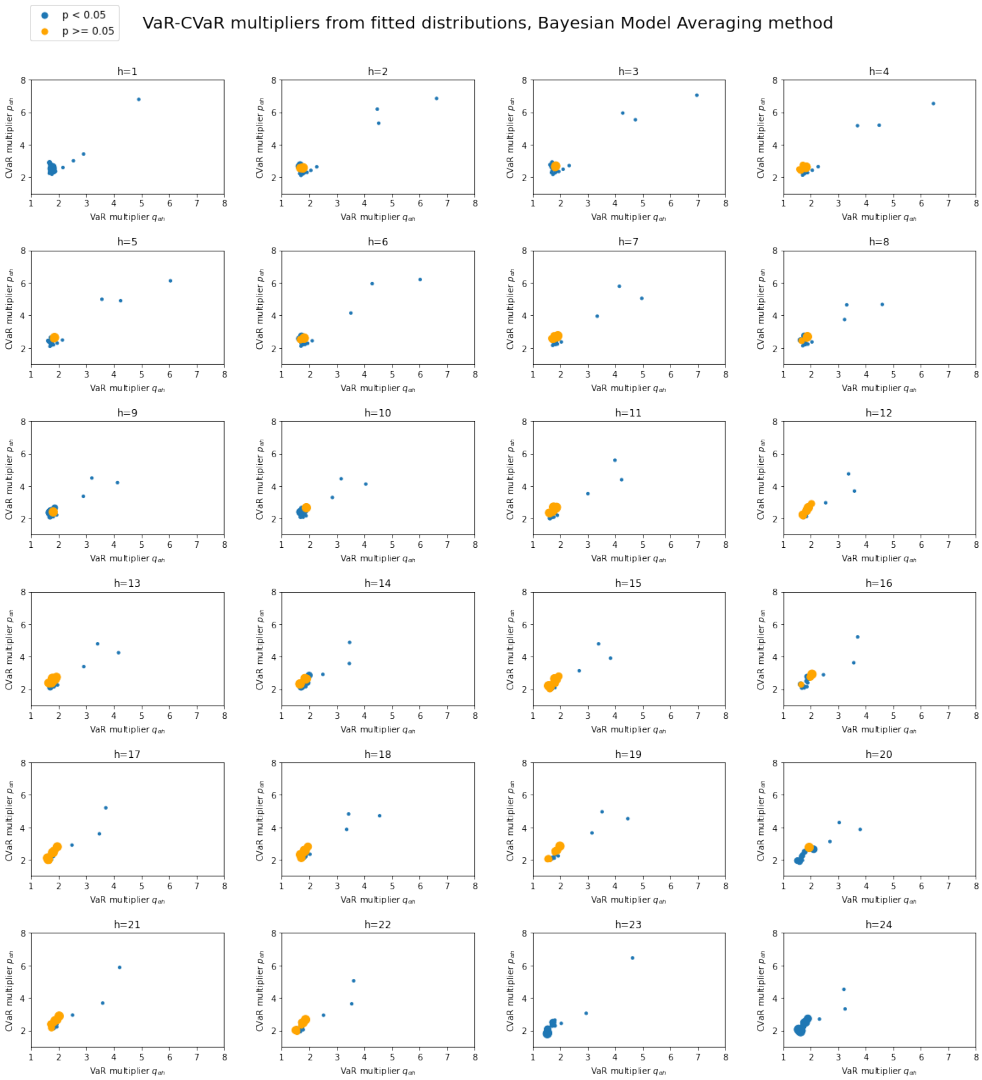

Figure 13: VaR-CVaR multiplier scatter plots from fitted distributions.

Appendix 3. Results tables

h	n_{train}	n_{test}	q	p	$test_q$	$test_{qL}$	$test_{qU}$	$test_p$	$test_{pL}$	$test_{pU}$
1	7996	3426	1.6449	2.0627	5.5244	5.1129	5.9447	-1.3623	-1.7247	-1.0634
2	3965	1698	1.6449	2.0627	5.4918	4.9091	6.0922	-1.2939	-1.7516	-0.9149
3	2621	1123	1.6449	2.0627	5.9829	5.235	6.7308	-1.4504	-2.026	-0.9782
4	1949	835	1.6449	2.0627	5.352	4.5259	6.2141	-1.4348	-2.0887	-0.8982
5	1546	662	1.6449	2.0627	5.7971	4.846	6.7935	-1.0591	-1.7161	-0.5321
6	1277	547	1.6449	2.0627	5.2632	4.2763	6.3048	-1.2571	-1.9547	-0.6776
7	1076	460	1.6449	2.0627	6.3151	5.1432	7.5521	-1.2963	-1.9803	-0.7192
8	941	403	1.6449	2.0627	5.5804	4.3899	6.8452	-1.1786	-1.9078	-0.5783
9	829	355	1.6449	2.0627	5.4899	4.223	6.7568	-0.8858	-1.6331	-0.3275
10	740	316	1.6449	2.0627	5.9659	4.5455	7.4811	-0.9105	-1.6543	-0.317
11	666	285	1.6449	2.0627	4.7319	3.47	6.0988	-0.6347	-1.1747	-0.1644
12	605	259	1.6449	2.0627	5.0926	3.7037	6.5972	-0.7795	-1.5458	-0.1861
13	553	237	1.6449	2.0627	4.8101	3.4177	6.3291	-1.0945	-1.9603	-0.4089
14	538	230	1.6449	2.0627	5.3385	3.776	7.0313	-0.8755	-1.6019	-0.2688
15	471	201	1.6449	2.0627	4.9107	3.4226	6.5476	-0.9351	-1.7046	-0.3311
16	437	187	1.6449	2.0627	5.4487	3.6859	7.3718	-0.8319	-1.6364	-0.1823
17	407	174	1.6449	2.0627	5.3356	3.6145	7.2289	-0.2288	-0.7818	0.2446
18	381	163	1.6449	2.0627	5.1471	3.3088	6.9853	-0.6249	-1.4356	0.0131
19	358	153	1.6449	2.0627	4.8924	3.1311	6.8493	-0.2188	-0.9062	0.2914
20	336	144	1.6449	2.0627	3.9583	2.2917	5.8333	-0.4574	-1.2067	0.1348
21	336	144	1.6449	2.0627	5.4167	3.5417	7.5	-0.9995	-1.8499	-0.2926
22	299	128	1.6449	2.0627	3.9813	2.3419	5.8548	-0.4399	-1.2923	0.2373
23	285	121	1.6449	2.0627	3.9409	2.2167	5.9113	-0.0292	-0.3938	0.2623
24	269	115	1.6449	2.0627	4.4271	2.6042	6.5104	-0.5313	-1.3551	0.1051

Table 1. Training and test set sizes and results of the validation of the current VaR model's normality assumption.

h	q	q_L	q_U	p	p_L	p_U
1	1.7534	1.6852	1.8375	2.6548	2.5221	2.7981
2	1.6731	1.5558	1.7962	2.553	2.3816	2.731
3	1.788	1.6272	1.8949	2.6621	2.4544	2.8829
4	1.6031	1.4589	1.7841	2.514	2.2643	2.785
5	1.6931	1.582	1.792	2.4073	2.1815	2.6589
6	1.6197	1.46	1.7452	2.4609	2.179	2.7681
7	1.7838	1.5379	2.0741	2.5624	2.3009	2.8355
8	1.6855	1.4541	1.9499	2.4851	2.1763	2.8089
9	1.6768	1.4361	1.9088	2.3764	2.0956	2.6598
10	1.7626	1.5354	1.9989	2.3944	2.1373	2.6609
11	1.6051	1.4266	1.778	2.2764	1.9766	2.6012
12	1.6213	1.4933	1.8281	2.2253	1.941	2.533
13	1.6393	1.4064	2.0051	2.4104	2.0437	2.7977
14	1.6803	1.4218	1.9833	2.3502	2.0242	2.702
15	1.5489	1.3566	2.004	2.2372	1.8952	2.5882
16	1.6469	1.4113	1.9093	2.309	1.9613	2.7089
17	1.5537	1.3562	1.774	2.0792	1.7776	2.411
18	1.7022	1.4588	1.8854	2.2882	1.9129	2.7699
19	1.4785	1.25	1.6961	2.0776	1.6811	2.5875
20	1.4941	1.2902	1.6171	1.9344	1.6493	2.3204
21	1.6316	1.3921	2.0152	2.356	1.9254	2.845
22	1.3633	1.1861	1.6764	2.0317	1.5967	2.5658
23	1.5364	1.2888	1.8091	1.9256	1.6605	2.2137
24	1.553	1.3137	1.8057	2.0088	1.6649	2.4351

Table 2. Empirical percentile method results for VaR and CVaR multipliers.

h	$test_q$	$test_{qL}$	$test_{qU}$	$test_p$	$test_{pL}$	$test_{pU}$
1	4.2615	3.0648	5.3999	-0.3154	-1.1923	0.1785
2	5.6537	3.828	7.8327	-0.2988	-1.4924	0.4203
3	5.3428	3.3838	8.2814	-0.306	-1.6761	0.5384
4	7.1856	4.0719	10.2994	-0.3385	-2.4952	0.692
5	5.7402	3.6254	8.3082	-0.8111	-3.1016	0.5606
6	6.5814	3.6563	10.4205	-0.4281	-2.7092	0.882
7	5.6522	2.1739	11.3043	-0.5779	-2.0936	0.1788
8	5.9553	2.2333	12.4069	-0.4271	-2.2996	0.5799
9	5.6338	1.9718	10.1408	-0.4053	-2.9517	0.568
10	5.3797	2.2152	10.1266	-0.6692	-3.0217	0.5939
11	4.2105	1.7544	9.1228	-0.3344	-1.7979	0.6114
12	5.7915	2.7027	11.9691	-0.962	-3.9818	0.8788
13	4.6414	1.2658	10.9705	-0.6748	-3.3587	0.5278
14	5.2174	0.8696	11.3043	-0.6358	-2.96	0.1429
15	7.4627	1.99	12.9353	-0.7469	-4.4329	0.2783
16	5.8824	1.0695	11.2299	-0.6941	-3.9062	0.3762
17	7.4713	1.7241	15.5172	-0.2096	-2.6336	0.6983
18	4.2945	1.227	11.6564	-0.5101	-2.9351	0.8833
19	9.1503	3.268	16.9935	0.5613	-2.3927	2.5585
20	7.6389	2.7778	18.0556	-1.3779	-5.4711	1.3228
21	7.6389	1.3889	13.8889	-0.2476	-4.0299	0.5115
22	10.1563	1.5625	19.5313	0.641	-4.7879	1.6806
23	7.438	0.8264	17.3554	0.1501	-1.8691	0.0478
24	8.6957	1.7391	21.7391	-0.906	-5.4463	0.4543

Table 3. Empirical percentile method results for performance metrics.

h	q	q_L	q_U	p	p_L	p_U
1	1.7246	1.667	1.7883	2.7329	2.6342	2.8543
2	1.671	1.5817	1.7904	2.561	2.3919	2.7949
3	1.7036	1.614	1.8729	2.6303	2.4719	2.8102
4	1.6438	1.5578	1.873	2.4398	2.3039	2.7525
5	1.6706	1.5404	1.8777	2.4275	2.2281	2.7163
6	1.7453	1.5579	1.8769	2.549	2.262	2.7521
7	1.7354	1.5853	1.8803	2.5455	2.2601	2.7703
8	1.7257	1.4783	1.933	2.4941	2.1253	2.7974
9	1.7208	1.5618	1.8494	2.4082	2.1454	2.613
10	1.7367	1.5313	1.9022	2.4498	2.1147	2.6984
11	1.629	1.4625	1.9168	2.3511	2.0637	2.7991
12	1.7229	1.5516	1.8796	2.2592	2.0007	2.492
13	1.7202	1.4383	2.0243	2.4493	1.9067	2.9041
14	1.7013	1.5161	1.9578	2.365	2.0741	2.7363
15	1.6055	1.4452	1.8534	2.215	1.9148	2.6294
16	1.6592	1.4522	1.9967	2.3039	1.9958	2.7857
17	1.611	1.4217	1.8512	2.1293	1.7931	2.5211
18	1.7438	1.4571	1.9823	2.4167	1.8756	2.7702
19	1.5687	1.3596	1.8261	2.0708	1.6876	2.5377
20	1.5726	1.3247	1.6917	2.0581	1.6224	2.3235
21	1.7805	1.5114	2.1036	2.4787	1.9193	2.9963
22	1.5171	1.3017	1.8339	2.0406	1.6737	2.644
23	1.58	1.3568	1.8311	2.1517	1.6794	2.7206
24	1.6057	1.353	1.8983	2.1442	1.6476	2.7107

Table 4. Bayesian Model Averaging method results for VaR and CVaR multipliers.

h	test_q	test_{qL}	test_{qU}	test_p	test_{pL}	test_{pU}
1	4.3199	3.2983	5.3123	-0.1635	-0.8025	0.3629
2	5.7126	4.0695	7.8416	-0.2596	-1.4693	0.4477
3	5.9662	3.4728	8.3348	-0.1643	-2.0102	0.4642
4	7.0659	4.0359	9.1198	-0.6083	-2.6863	0.5658
5	5.8912	3.1495	8.5498	-0.7105	-2.4574	0.3703
6	5.4845	3.053	8.9854	-0.5625	-2.2627	0.3078
7	6.087	2.8261	10.4674	-0.4849	-2.4826	0.5643
8	5.9553	2.196	11.4888	-0.4042	-2.7852	0.5123
9	5.3521	2.7746	9.2958	-0.4098	-2.9279	0.6832
10	5.3797	2.2152	10.7595	-0.5324	-3.1335	0.6769
11	4.2105	0.6491	7.8246	-0.19	-1.2171	0.3512
12	5.4054	3.0888	10.0965	-0.9672	-3.5436	0.6026
13	4.2194	1.2658	11.3924	-0.7257	-4.1455	0.5898
14	4.7826	1.3043	8.9565	-0.7219	-3.3481	0.3194
15	6.4677	2.9104	11.4428	-1.1054	-4.7677	0.5474
16	5.3476	1.0695	10.6952	-0.859	-3.9822	0.3093
17	6.8966	1.6379	12.069	-0.1751	-2.6503	0.4096
18	4.2945	1.227	12.362	-0.2548	-2.8789	0.8313
19	7.8431	1.9608	15.1307	0.1823	-3.0139	1.6389
20	6.9444	1.9792	17.3611	-1.0008	-6.2105	0.4826
21	6.9444	1.2847	12.7083	-0.0928	-4.5848	0.6171
22	7.0313	0.7813	16.5234	-0.2115	-4.1736	0.6969
23	4.1322	0.8264	17.3554	0.0029	-1.5795	1.2192
24	5.2174	2.4783	18.2609	-1.2204	-4.6209	1.1238

Table 5. Bayesian Model Averaging method results for performance metrics.

Appendix 4. Source code

```python
""## Include packages
import keyring
import pandas as pd
import numpy as np
import scipy
from scipy import stats
import matplotlib.pyplot as plt
import datetime as dt
import time
import copy
import random
import sqlalchemy
import snowflake.connector
from statsmodels.graphics.tsaplots import plot_acf
from snowflake.sqlalchemy import URL
from os import name as os_name
from scipy import stats

## Define database connection functions

def sf_connect():
    # ...
    # / Excluded from thesis appendix due to privacy reasons /

## Define functions to extract data and calculate ratios x_h

def get_base(con, start_time, end_time):
    # make sure to exclude GBP positions

    """
    Query hourly exposures from Treasury
    """

    qt = """
    with data as   (select SNAPSHOT_TIME HOURLY_TIMESTAMP,
                           CCY,
                           sum(BALANCE)           BALANCE_CCY
                    from  // excluded //
                            left join // excluded //
                    where ACCOUNT.TYPE in ('HEDGE', 'BORDERLESS', 'CUSTOMER')
                      and CCY != 'GBP'
                      and SNAPSHOT_TIME >= '{start_time}'
                      and SNAPSHOT_TIME < '{end_time}'
                    group by 1, 2
                    having BALANCE_CCY != 0)
    select distinct CCY
    from data;
    """.format(start_time=start_time, end_time=end_time)

    try:
        datetimes = pd.DataFrame(
            {'HOURLY_TIMESTAMP': pd.date_range(start_time, end_time, freq='1H', closed='left')})
        df_ccy = pd.read_sql(qt, con)
        base = datetimes.assign(key=1).merge(df_ccy.assign(key=1), on='key').drop('key', axis=1)

        return base
```

```python

    except Exception as e:
        print(e)


def get_hourly_rates(con, start_time, end_time):
    qt = """
              select
                   rh.HOURLY_TIMESTAMP,
                   rh.CCY,
                   rh.rate_to_gbp as MID_RATE
              from
                    // excluded // as rh
              where
                    rh.hourly_timestamp >= '{start_time}'
                    and rh.hourly_timestamp < '{end_time}';
                """.format(start_time=start_time, end_time=end_time)
    try:
        df_new = pd.read_sql(qt, con)

        return df_new

    except Exception as e:
        print(traceback.format_exc())


def get_hourly_positions(con, base, rates, start_time, end_time):
    # make sure to exclude GBP positions

    """
    Query hourly exposures from Treasury
    """

    qt = """
    select SNAPSHOT_TIME HOURLY_TIMESTAMP,
           CCY,
           sum(BALANCE)          BALANCE_CCY
    from // excluded //
             left join // excluded //
    where ACCOUNT.TYPE in ('HEDGE', 'BORDERLESS', 'CUSTOMER')
      and CCY != 'GBP'
      and SNAPSHOT_TIME >= '{start_time}'
      and SNAPSHOT_TIME < '{end_time}'
    group by 1, 2;
    """.format(start_time=start_time, end_time=end_time)

    try:
        df_balances = pd.read_sql(qt, con)
        df_rates = copy.deepcopy(rates)

        # merge positions and rates
        exposures = pd.merge(base, df_balances, how='left', on=['HOURLY_TIMESTAMP', 'CCY'])
        exposures = pd.merge(exposures, df_rates, how='left', on=['HOURLY_TIMESTAMP', 'CCY'])
        exposures['BALANCE_CCY'] = exposures['BALANCE_CCY'].fillna(0)
        exposures['MID_RATE'] = exposures.groupby('CCY')['MID_RATE'].fillna(method="ffill")
        exposures['MID_RATE'] = exposures.groupby('CCY')['MID_RATE'].fillna(method="bfill")

        # calculate exposures in GBP
```

```
        exposures['BALANCE_GBP'] = exposures['BALANCE_CCY'] * exposures['MID_RATE']
        exposures = exposures.sort_values(by='HOURLY_TIMESTAMP', ascending=True)

        return exposures

    except Exception as e:
        print(e)


def get_hourly_fx(positions, period_hours):
    df_fx = copy.deepcopy(positions)
    # set off-hours rate change to NaN
    df_fx.loc[((df_fx['HOURLY_TIMESTAMP'].dt.dayofweek == 4) & (df_fx['HOURLY_TIMESTAMP'].dt.hour >= 22)) | (
            df_fx['HOURLY_TIMESTAMP'].dt.dayofweek == 5) | ((df_fx['HOURLY_TIMESTAMP'].dt.dayofweek == 6) & (
            df_fx['HOURLY_TIMESTAMP'].dt.hour <= 21)), 'MID_RATE'] = np.nan

    # depending on period length, find the end-of-period rate
    df_fx['END_RATE'] = df_fx.groupby('CCY')['MID_RATE'].shift(-period_hours)

    # calculate rate change
    df_fx['RATE_CHANGE'] = df_fx['END_RATE'] - df_fx['MID_RATE']

    df_fx['FX'] = df_fx['BALANCE_CCY'] * df_fx['RATE_CHANGE']
    return df_fx


def get_ratios_freq(start_time, end_time, period_hours):
    # get list of currencies
    ccy = pd.read_csv('covariance/2020-01-01_covmat.csv')
    ccy.rename(columns={'ccy1': 'CCY'}, inplace=True)
    ccy = ccy['CCY']

    # generate date range
    start = dt.datetime.strptime(start_time, "%Y-%m-%d")
    end = dt.datetime.strptime(end_time, "%Y-%m-%d")
    dates = [start + dt.timedelta(days=x) for x in range(0, (end - start).days)]

    # placeholder for output
    output = pd.DataFrame(columns=['datetime', 'FX', 'std', 'FX/std ratio'])

    # generate data
    base = get_base(con, start_time, end_time)
    hourly_rates = get_hourly_rates(con, start_time, end_time)
    positions = get_hourly_positions(con, base, hourly_rates, start_time, end_time)
    fx = get_hourly_fx(positions, period_hours)

    # calculate hourly total FX
    hourly_fx = fx.groupby(['HOURLY_TIMESTAMP'])['FX'].apply(pd.DataFrame.sum, skipna=False).reset_index()

    for date in dates:

        # get covmatrix and convert to respective frequency
```

```python
        df_covmatrix = pd.read_csv('covariance/' + str(date.strftime("%Y-%m-%d")) + '_covmat.csv')
        df_covmatrix.rename(columns={'ccy1': 'CCY'}, inplace=True)
        covmat = df_covmatrix.drop(columns=['CCY']).values
        covmat_hourly = covmat / (24 / period_hours)

        # generate datetimes in the day
        datetimes = [date + dt.timedelta(hours=x) for x in range(0, 24)]

        for time in datetimes:
            # get FX for the datetime
            time_fx = hourly_fx.loc[hourly_fx['HOURLY_TIMESTAMP'] == str(time), 'FX'].values[0]

            # get position for the datetime
            time_positions = positions.loc[positions['HOURLY_TIMESTAMP'] == str(time)]
            time_exposures = pd.merge(df_covmatrix['CCY'], time_positions, how='left', on=['CCY'])
            time_exposures = time_exposures.fillna(0)
            vec_exp = time_exposures['BALANCE_GBP'].values

            # calculate variance for the datetime
            variance = (vec_exp.dot(covmat_hourly)).dot(vec_exp.T)

            # calculate std for the datetime
            std = np.sqrt(variance)

            # calculate ratios for the datetime
            std_ratio = time_fx / std

            # append to output
            output = output.append({'datetime': time, 'FX': time_fx, 'std': std, 'FX/std ratio': std_ratio},
                                   ignore_index=True)

    output = output.sort_values(by='datetime', ascending=True)
    return output


## Define functions for Bayesian Model Averaging method

def fit_distributions_light(dataset, alpha):  # set alpha for confidence level

    # fit distributions
    list_of_dists = ['dgamma', 'dweibull', 'exponnorm', 'foldnorm', 'gennorm', 'genextreme', 'gumbel_r', 'gumbel_l',
                     'hypsecant', 'laplace', 'logistic', 'moyal', 'nct', 'norm', 'norminvgauss', 'pearson3', 'skewnorm',
                     't', 'triang', 'truncnorm', 'tukeylambda']

    results = []

    # fit all distributions
    for d in list_of_dists:
        dist = getattr(stats, d)

        # get parameters by fitting distribution to empirical data (training set)
        param = dist.fit(dataset)

```

```
        # calculate likelihood (training set)
        loglh = dist.logpdf(dataset, *param).sum()
        lh = np.exp(loglh)

        # calculate VaR multiplier
        VaR = dist.ppf(alpha, *param)

        # assign parameters
        n = len(param)
        shape = param[0:n - 2]
        loc = param[n - 2]
        scale = param[n - 1]

        # calculate tail loss and CVaR multiplier
        tail_loss = dist.expect(lambda x: x, args=shape, loc=loc, scale=scale, ub=VaR)
        CVaR = (1 / alpha) * tail_loss

        results.append([d, loglh, lh, -VaR, -CVaR])

    # save results to dataframe
    results.sort(key=lambda x: float(x[2]), reverse=True)
    df_results = pd.DataFrame(results, columns=['distribution', 'log-likelihood', 'likelihood', 'VaR', 'CVaR'])

    # get weights from likelihoods
    max_loglh = df_results['log-likelihood'].max()
    df_results['adj-log-likelihood'] = df_results['log-likelihood'] - max_loglh
    df_results['lh-weight'] = np.exp(df_results['adj-log-likelihood'])

    # calculate multipliers
    mults = []
    lhsum = df_results['lh-weight'].sum()

    VaR_estimate = (df_results['lh-weight'] / lhsum * df_results['VaR']).sum()
    CVaR_estimate = (df_results['lh-weight'] / lhsum * df_results['CVaR']).sum()

    # save results to dataframe
    mults.append([VaR_estimate, CVaR_estimate])

    return mults


def fit_distributions_forplotting(dataset, test, alpha):  # set alpha for confidence level

    # fit distributions
    list_of_dists = ['dgamma', 'dweibull', 'exponnorm', 'foldnorm', 'gennorm', 'genextreme', 'gumbel_r', 'gumbel_l',
                     'hypsecant', 'laplace', 'logistic', 'moyal', 'nct', 'norm', 'norminvgauss', 'pearson3', 'skewnorm',
                     't', 'triang', 'truncnorm', 'tukeylambda']

    results = []

    # fit all distributions
    for d in list_of_dists:
        dist = getattr(stats, d)

```

```
        # get parameters by fitting distribution to empirical data (training set)
        param = dist.fit(dataset)

        # perform Kolmogorov-Smirnov test for goodness of fit (test set)
        ks = stats.kstest(test, d, args=param)

        # calculate likelihood (training set)
        loglh = dist.logpdf(dataset, *param).sum()
        lh = np.exp(loglh)

        # calculate VaR multiplier
        VaR = dist.ppf(alpha, *param)

        # assign parameters
        n = len(param)
        shape = param[0:n - 2]
        loc = param[n - 2]
        scale = param[n - 1]

        # calculate tail loss and CVaR multiplier
        tail_loss = dist.expect(lambda x: x, args=shape, loc=loc, scale=scale, ub=VaR)
        CVaR = (1 / alpha) * tail_loss

        results.append([d, ks[0], ks[1], loglh, lh, -VaR, -CVaR])

    # save results to dataframe
    results.sort(key=lambda x: float(x[2]), reverse=True)
    df_results = pd.DataFrame(results,
                              columns=['distribution', 'KS-statistic', 'KS-pvalue', 'log-likelihood', 'likelihood',
                                       'VaR', 'CVaR'])

    # get weights from likelihoods
    max_loglh = df_results['log-likelihood'].max()
    df_results['adj-log-likelihood'] = df_results['log-likelihood'] - max_loglh
    df_results['lh-weight'] = np.exp(df_results['adj-log-likelihood'])

    return df_results


## Connect to database

con = sf_connect()


## Generate data

start_date = '2019-07-01'
end_date = '2021-05-01'

h = 24
raw_datasets = []

for f in range(1,h+1):
    print("Period:", f, 'hour(s)')
    data_f = get_ratios_freq(start_date, end_date, f)
    raw_datasets.append([f, data_f])


```

```
## Describe data (histograms)

a = 6  # number of rows
b = 4  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(20, 21))
stats = []

for f in range(1, h + 1):
    # get the full (overlapping) dataset for the frequency
    df = raw_datasets[f - 1][1]

    # get indexes and data for non-overlapping periods
    size = df.shape[0]
    indexes = np.arange(0, size, f)
    df_nonoverlap = df.iloc[indexes, :]
    mean = np.round(df_nonoverlap['FX/std ratio'].mean(), 4)
    std = np.round(df_nonoverlap['FX/std ratio'].std(), 4)
    n = df_nonoverlap.shape[0]
    minimum = df_nonoverlap['FX/std ratio'].min()
    maximum = df_nonoverlap['FX/std ratio'].max()
    stats.append([f, n, mean, std, minimum, maximum])

    plt.subplot(a, b, c)
    plt.title('h={}; n={} \n $\mu$={}; $\sigma$={}'.format(f, n, mean, std))
    df_nonoverlap['FX/std ratio'].hist(density=True, bins=np.arange(int(np.floor(df_nonoverlap['FX/std ratio'].min())),
                                                                    int(np.ceil(df_nonoverlap['FX/std ratio'].max())),
                                                                    0.1))
    plt.xlim(-15, 15)
    plt.ylim(0, 0.75)
    c = c + 1

df_stats = pd.DataFrame(stats, columns=['h', 'n', 'mu', 'sigma', 'min', 'max'])

fig.subplots_adjust(hspace=0.5)
fig.subplots_adjust(wspace=0.3)
fig.suptitle("Histograms of $\mathbf{x}_h$", fontsize=20, y=0.93)
plt.show()

df_stats

## Plot autocorrelation plot

hourly_data = raw_datasets[0][1].dropna(subset=['FX/std ratio'])
plt.figure(figsize=(20,5))
pd.plotting.autocorrelation_plot(hourly_data['FX/std ratio'])
plt.ylim(-0.05,0.05)


## Current VaR model validation

from scipy.stats import norm

# assign parameters
loc = 0
scale = 1

# calculate tail loss and CVaR multiplier
```

```python
norm_VaR = -norm.ppf(0.05, loc=loc, scale=scale)
norm_tail_loss = norm.expect(lambda x: x, loc=loc, scale=scale, ub = norm_VaR)
norm_CVaR = -(1/0.05)*norm_tail_loss

random.seed(0)
bs_count = 20000
confidence_level = 95
train_pct = 0.7
norm_results = []

for f in range(1, h + 1):
    print("Period:", f, 'hour(s)')

    # Take full df with overlapping periods
    full_df = raw_datasets[f - 1][1]

    # Extract only a subset of datapoints that are not over-lapping (currently for starting hour 00:00)
    size = full_df.shape[0]
    indexes = np.arange(0, size, f)
    data_f = full_df.iloc[indexes, :]

    # Exclude NaN values
    data_f = data_f.dropna(subset=['FX/std ratio'])

    data = data_f
    n = data_f.shape[0]
    indexes = np.arange(0, n)

    norm_test = []

    # test mean multipliers on test dataset
    var_exceeded = data['FX/std ratio'] < -norm_VaR
    VaR_exceeded_pct_mean = var_exceeded.sum() / n * 100
    cvar_mean_diff = (var_exceeded * ((data['FX/std ratio'] + norm_CVaR) / norm_CVaR)).mean()
    CVaR_mean_diff_pct_mean = cvar_mean_diff * 100

    # bootstrapping for testing
    for i in range(0, bs_count):
        # get bootstrap sample
        bs_indexes = np.random.choice(indexes, size=n)
        bs_sample = data.iloc[bs_indexes, :]

        # test multipliers on test dataset (upper)
        var_exceeded = bs_sample['FX/std ratio'] < -norm_VaR
        var_exceeded_pct = var_exceeded.sum() / n * 100
        cvar_mean_diff = (var_exceeded * ((bs_sample['FX/std ratio'] + norm_CVaR) / norm_CVaR)).mean()
        cvar_mean_diff_pct = cvar_mean_diff * 100

        # save results
        norm_test.append([var_exceeded_pct, cvar_mean_diff_pct])
        df_norm_test = pd.DataFrame(norm_test, columns=['VaR_exceeded_pct', 'CVaR_mean_diff_pct'])

    # calculate CI for test results
    sample = df_norm_test['VaR_exceeded_pct'] - VaR_exceeded_pct_mean
    VaR_exceeded_pct_CIl, VaR_exceeded_pct_CIu = VaR_exceeded_pct_mean + np.percentile(sample, (
```

```python
                (100 - confidence_level) / 2)), VaR_exceeded_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    sample = df_norm_test['CVaR_mean_diff_pct'] - CVaR_mean_diff_pct_mean
    CVaR_mean_diff_pct_CIl, CVaR_mean_diff_pct_CIu = CVaR_mean_diff_pct_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), CVaR_mean_diff_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    # save results
    norm_results.append([f,
                         norm_VaR,
                         norm_CVaR,
                         VaR_exceeded_pct_mean,
                         VaR_exceeded_pct_CIl, VaR_exceeded_pct_CIu,
                         CVaR_mean_diff_pct_mean,
                         CVaR_mean_diff_pct_CIl, CVaR_mean_diff_pct_CIu])

df_norm_results = pd.DataFrame(norm_results,
                               columns=['period_in_h',
                                        'norm_VaR_mult',
                                        'norm_CVaR_mult',
                                        'VaR_exceeded_pct_mean',
                                        'VaR_exceeded_pct_CIl', 'VaR_exceeded_pct_CIu',
                                        'CVaR_mean_diff_pct_mean',
                                        'CVaR_mean_diff_pct_CIl', 'CVaR_mean_diff_pct_CIu'])
df_norm_results


## Empirical percentile method

random.seed(0)
bs_count = 20000
confidence_level = 95
train_pct = 0.7
emp_results = []

for f in range(1, h + 1):
    print("Period:", f, 'hour(s)')

    # Take full df with overlapping periods
    full_df = raw_datasets[f - 1][1]

    # Extract only a subset of datapoints that are not over-lapping (currently for starting hour 00:00)
    size = full_df.shape[0]
    indexes = np.arange(0, size, f)
    data_f = full_df.iloc[indexes, :]

    # Exclude NaN values
    data_f = data_f.dropna(subset=['FX/std ratio'])

    # Split data to train and test (70/30)
    n = data_f.shape[0]
    train = data_f.head(int(np.ceil(train_pct * n)))
    test = data_f.tail(n - int(np.ceil(train_pct * n)))

    n_train = train.shape[0]
```

```
    n_test = test.shape[0]

    train_indexes = np.arange(0, n_train)
    test_indexes = np.arange(0, n_test)

    emp_mults = []
    emp_test_u = []
    emp_test_l = []

    # get mean estimates from training dataset
    emp_VaR_mult_mean = np.abs(np.percentile(train['FX/std ratio'], 5))
    emp_CVaR_mult_mean = np.abs(train.loc[train['FX/std ratio'] < -emp_VaR_mult_mean, 'FX/std ratio'].mean())

    # test mean multipliers on test dataset
    var_exceeded = test['FX/std ratio'] < -emp_VaR_mult_mean
    VaR_exceeded_pct_mean = var_exceeded.sum() / n_test * 100
    cvar_mean_diff = (var_exceeded * ((test['FX/std ratio'] + emp_CVaR_mult_mean) / emp_CVaR_mult_mean)).mean()
    CVaR_mean_diff_pct_mean = cvar_mean_diff * 100

    # bootstrapping for multipliers
    for i in range(0, bs_count):
        # get bootstrap sample
        bs_indexes = np.random.choice(train_indexes, size=n_train)
        bs_sample = train.iloc[bs_indexes, :]

        # calculate multipliers on bootstrap sample of training dataset
        emp_VaR_mult = np.abs(np.percentile(bs_sample['FX/std ratio'], 5))
        emp_CVaR_mult = np.abs(bs_sample.loc[bs_sample['FX/std ratio'] < -emp_VaR_mult, 'FX/std ratio'].mean())

        # save results
        emp_mults.append([emp_VaR_mult, emp_CVaR_mult])
        df_emp_mults = pd.DataFrame(emp_mults, columns=['emp_VaR_mult', 'emp_CVaR_mult'])

    # calculate CI for multipliers
    sample = df_emp_mults['emp_VaR_mult'] - emp_VaR_mult_mean
    emp_VaR_mult_CIl, emp_VaR_mult_CIu = emp_VaR_mult_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), emp_VaR_mult_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    sample = df_emp_mults['emp_CVaR_mult'] - emp_CVaR_mult_mean
    emp_CVaR_mult_CIl, emp_CVaR_mult_CIu = emp_CVaR_mult_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), emp_CVaR_mult_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    # bootstrapping for testing
    for i in range(0, bs_count):
        # get bootstrap sample
        bs_indexes = np.random.choice(test_indexes, size=n_test)
        bs_sample = test.iloc[bs_indexes, :]

        # test multipliers on test dataset (upper)
        var_exceeded_u = bs_sample['FX/std ratio'] < -emp_VaR_mult_CIu
        var_exceeded_pct_u = var_exceeded_u.sum() / n_test * 100
        cvar_mean_diff_u = (
```

```
                    var_exceeded_u * ((bs_sample['FX/std ratio'] + emp_CVaR_mult_CIu) / emp_CVaR_mult_CIu)).mean()
        cvar_mean_diff_pct_u = cvar_mean_diff_u * 100

        # test multipliers on test dataset (lower)
        var_exceeded_l = bs_sample['FX/std ratio'] < -emp_VaR_mult_CIl
        var_exceeded_pct_l = var_exceeded_l.sum() / n_test * 100
        cvar_mean_diff_l = (
                    var_exceeded_l * ((bs_sample['FX/std ratio'] + emp_CVaR_mult_CIl) / emp_CVaR_mult_CIl)).mean()
        cvar_mean_diff_pct_l = cvar_mean_diff_l * 100

        # save results
        emp_test_u.append([var_exceeded_pct_u, cvar_mean_diff_pct_u])
        emp_test_l.append([var_exceeded_pct_l, cvar_mean_diff_pct_l])

        df_emp_test_u = pd.DataFrame(emp_test_u, columns=['VaR_exceeded_pct_u', 'CVaR_mean_diff_pct_u'])
        df_emp_test_l = pd.DataFrame(emp_test_l, columns=['VaR_exceeded_pct_l', 'CVaR_mean_diff_pct_l'])

    # calculate CI for test results
    sample = df_emp_test_u['VaR_exceeded_pct_u'] - VaR_exceeded_pct_mean
    VaR_exceeded_pct_u_CIl, VaR_exceeded_pct_u_CIu = VaR_exceeded_pct_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), VaR_exceeded_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    sample = df_emp_test_u['CVaR_mean_diff_pct_u'] - CVaR_mean_diff_pct_mean
    CVaR_mean_diff_pct_u_CIl, CVaR_mean_diff_pct_u_CIu = CVaR_mean_diff_pct_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), CVaR_mean_diff_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    sample = df_emp_test_l['VaR_exceeded_pct_l'] - VaR_exceeded_pct_mean
    VaR_exceeded_pct_l_CIl, VaR_exceeded_pct_l_CIu = VaR_exceeded_pct_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), VaR_exceeded_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    sample = df_emp_test_l['CVaR_mean_diff_pct_l'] - CVaR_mean_diff_pct_mean
    CVaR_mean_diff_pct_l_CIl, CVaR_mean_diff_pct_l_CIu = CVaR_mean_diff_pct_mean + np.percentile(sample, (
                (100 - confidence_level) / 2)), CVaR_mean_diff_pct_mean + np.percentile(sample, (
                confidence_level + (100 - confidence_level) / 2))

    # save results
    emp_results.append([f, n_train, n_test,
                        emp_VaR_mult_mean, emp_VaR_mult_CIl, emp_VaR_mult_CIu,
                        emp_CVaR_mult_mean, emp_CVaR_mult_CIl, emp_CVaR_mult_CIu,
                        VaR_exceeded_pct_mean,
                        VaR_exceeded_pct_u_CIl, VaR_exceeded_pct_u_CIu,
                        VaR_exceeded_pct_l_CIl, VaR_exceeded_pct_l_CIu,
                        CVaR_mean_diff_pct_mean,
                        CVaR_mean_diff_pct_u_CIl, CVaR_mean_diff_pct_u_CIu,
                        CVaR_mean_diff_pct_l_CIl, CVaR_mean_diff_pct_l_CIu])
```

```python

df_emp_results = pd.DataFrame(emp_results,
                              columns=['period_in_h', 'train_size', 'test_size',
                                       'emp_VaR_mult', 'emp_VaR_mult_CIl', 'emp_VaR_mult_CIu',
                                       'emp_CVaR_mult', 'emp_CVaR_mult_CIl', 'emp_CVaR_mult_CIu',
                                       'VaR_exceeded_pct_mean',
                                       'VaR_exceeded_pct_u_CIl', 'VaR_exceeded_pct_u_CIu',
                                       'VaR_exceeded_pct_l_CIl', 'VaR_exceeded_pct_l_CIu',
                                       'CVaR_mean_diff_pct_mean',
                                       'CVaR_mean_diff_pct_u_CIl', 'CVaR_mean_diff_pct_u_CIu',
                                       'CVaR_mean_diff_pct_l_CIl', 'CVaR_mean_diff_pct_l_CIu'])
df_emp_results


## Bayesian Model Averaging method

## Plot VaR-CVaR

random.seed(0)
confidence_level = 95
train_pct = 0.7
bam_results = []
alpha = 0.05

for f in range(1, h + 1):
    print("Period:", f, 'hour(s)')

    # Take full df with overlapping periods
    full_df = raw_datasets[f - 1][1]

    # Extract only a subset of datapoints that are not over-lapping
    size = full_df.shape[0]
    indexes = np.arange(0, size, f)
    data_f = full_df.iloc[indexes, :]

    # Exclude NaN values
    data_f = data_f.dropna(subset=['FX/std ratio'])

    # Split data to train and test (70/30)
    n = data_f.shape[0]
    train = data_f.head(int(np.ceil(train_pct * n)))
    test = data_f.tail(n - int(np.ceil(train_pct * n)))

    # get mean estimates from training dataset
    bam_data = fit_distributions_forplotting(train['FX/std ratio'], test['FX/std ratio'], alpha)

    # save results
    bam_results.append([f, bam_data])

a = 6  # number of rows
b = 4  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(20, 21))
```

```
stats = []

for f in range(1, h + 1):
    data = bam_results[f - 1][1]
    data.loc[data['KS-pvalue'] >= 0.05, 'category'] = 'p >= 0.05'
    data.loc[data['KS-pvalue'] < 0.05, 'category'] = 'p < 0.05'

    maxp = data['KS-pvalue'].max()
    data['s'] = data['KS-pvalue'].to_numpy() / maxp
    data['dotsize'] = 100 * data['s'] ** 0.5
    data.loc[data['dotsize'] < 10, 'dotsize'] = 10

    set1 = data[data['category'] == 'p < 0.05']
    set2 = data[data['category'] == 'p >= 0.05']

    plt.subplot(a, b, c)
    plt.scatter(set1.VaR, set1.CVaR, marker='o', s=set1['dotsize'], label='p
     < 0.05')
    plt.scatter(set2.VaR, set2.CVaR, marker='o', s=set2['dotsize'], label='p
     >= 0.05', c='orange')

    plt.title('h={}'.format(f))
    plt.xlabel(r'VaR multiplier $q_{\alpha h}$')
    plt.ylabel(r'CVaR multiplier $p_{\alpha h}$')

    plt.xlim(1, 8)
    plt.ylim(1, 8)
    c = c + 1

fig.subplots_adjust(hspace=0.5)
fig.subplots_adjust(wspace=0.3)
fig.suptitle("VaR-CVaR multipliers from fitted distributions, Bayesian Model
     Averaging method", fontsize=20, y=0.93)
lgnd = fig.legend(['p < 0.05', 'p >= 0.05'], loc=(0.07, 0.935), fontsize=12)
lgnd.legendHandles[0]._sizes = [50]
lgnd.legendHandles[1]._sizes = [50]
plt.show()

## Calculate multipliers

random.seed(0)
bs_count = 75
confidence_level = 95
train_pct = 0.7
bam_results = []
alpha = 0.05

for f in range(1, h + 1):
    print("Period:", f, 'hour(s)')

    # Take full df with overlapping periods
    full_df = raw_datasets[f - 1][1]

    # Extract only a subset of datapoints that are not over-lapping (
    currently for starting hour 00:00)
    size = full_df.shape[0]
    indexes = np.arange(0, size, f)
    data_f = full_df.iloc[indexes, :]

    # Exclude NaN values
    data_f = data_f.dropna(subset=['FX/std ratio'])

```

```python
    # Split data to train and test (70/30)
    n = data_f.shape[0]
    train = data_f.head(int(np.ceil(train_pct * n)))
    test = data_f.tail(n - int(np.ceil(train_pct * n)))

    n_train = train.shape[0]
    n_test = test.shape[0]

    train_indexes = np.arange(0, n_train)
    test_indexes = np.arange(0, n_test)

    bam_mults = []
    bam_test_u = []
    bam_test_l = []

    # get mean estimates from training dataset
    bam_mult_mean = fit_distributions_light(train['FX/std ratio'], alpha)
    bam_VaR_mult_mean = bam_mult_mean[0][0]
    bam_CVaR_mult_mean = bam_mult_mean[0][1]

    # test mean multipliers on test dataset
    var_exceeded = test['FX/std ratio'] < -bam_VaR_mult_mean
    VaR_exceeded_pct_mean = var_exceeded.sum() / n_test * 100
    cvar_mean_diff = (var_exceeded * ((test['FX/std ratio'] + bam_CVaR_mult_mean) / bam_CVaR_mult_mean)).mean()
    CVaR_mean_diff_pct_mean = cvar_mean_diff * 100

    # bootstrapping for multipliers
    for i in range(0, bs_count):
        print('calculating multipliers, bootstrap sample:', i)
        # get bootstrap sample
        bs_indexes = np.random.choice(train_indexes, size=n_train)
        bs_sample = train.iloc[bs_indexes, :]

        # calculate multipliers on bootstrap sample of training dataset
        bam_mult = fit_distributions_light(bs_sample['FX/std ratio'], alpha)
        bam_VaR_mult = bam_mult[0][0]
        bam_CVaR_mult = bam_mult[0][1]

        # save results
        bam_mults.append([bam_VaR_mult, bam_CVaR_mult])
        df_bam_mults = pd.DataFrame(bam_mults, columns=['bam_VaR_mult', 'bam_CVaR_mult'])

    # calculate CI for multipliers
    sample = df_bam_mults['bam_VaR_mult'] - bam_VaR_mult_mean
    bam_VaR_mult_CIl, bam_VaR_mult_CIu = bam_VaR_mult_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), bam_VaR_mult_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    sample = df_bam_mults['bam_CVaR_mult'] - bam_CVaR_mult_mean
    bam_CVaR_mult_CIl, bam_CVaR_mult_CIu = bam_CVaR_mult_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), bam_CVaR_mult_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    # bootstrapping for testing
    for i in range(0, bs_count):
        # get bootstrap sample
```

```
        print('testing, bootstrap sample:', i)
        bs_indexes = np.random.choice(test_indexes, size=n_test)
        bs_sample = test.iloc[bs_indexes, :]

        # test multipliers on test dataset (upper)
        var_exceeded_u = bs_sample['FX/std ratio'] < -bam_VaR_mult_CIu
        var_exceeded_pct_u = var_exceeded_u.sum() / n_test * 100
        cvar_mean_diff_u = (
                    var_exceeded_u * ((bs_sample['FX/std ratio'] + bam_CVaR_mult_CIu) / bam_CVaR_mult_CIu)).mean()
        cvar_mean_diff_pct_u = cvar_mean_diff_u * 100

        # test multipliers on test dataset (lower)
        var_exceeded_l = bs_sample['FX/std ratio'] < -bam_VaR_mult_CIl
        var_exceeded_pct_l = var_exceeded_l.sum() / n_test * 100
        cvar_mean_diff_l = (
                    var_exceeded_l * ((bs_sample['FX/std ratio'] + bam_CVaR_mult_CIl) / bam_CVaR_mult_CIl)).mean()
        cvar_mean_diff_pct_l = cvar_mean_diff_l * 100

        # save results
        bam_test_u.append([var_exceeded_pct_u, cvar_mean_diff_pct_u])
        bam_test_l.append([var_exceeded_pct_l, cvar_mean_diff_pct_l])

        df_bam_test_u = pd.DataFrame(bam_test_u, columns=['VaR_exceeded_pct_u', 'CVaR_mean_diff_pct_u'])
        df_bam_test_l = pd.DataFrame(bam_test_l, columns=['VaR_exceeded_pct_l', 'CVaR_mean_diff_pct_l'])

    # calculate CI for test results
    sample = df_bam_test_u['VaR_exceeded_pct_u'] - VaR_exceeded_pct_mean
    VaR_exceeded_pct_u_CIl, VaR_exceeded_pct_u_CIu = VaR_exceeded_pct_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), VaR_exceeded_pct_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    sample = df_bam_test_u['CVaR_mean_diff_pct_u'] - CVaR_mean_diff_pct_mean
    CVaR_mean_diff_pct_u_CIl, CVaR_mean_diff_pct_u_CIu = CVaR_mean_diff_pct_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), CVaR_mean_diff_pct_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    sample = df_bam_test_l['VaR_exceeded_pct_l'] - VaR_exceeded_pct_mean
    VaR_exceeded_pct_l_CIl, VaR_exceeded_pct_l_CIu = VaR_exceeded_pct_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), VaR_exceeded_pct_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    sample = df_bam_test_l['CVaR_mean_diff_pct_l'] - CVaR_mean_diff_pct_mean
    CVaR_mean_diff_pct_l_CIl, CVaR_mean_diff_pct_l_CIu = CVaR_mean_diff_pct_mean + np.percentile(sample, (
            (100 - confidence_level) / 2)), CVaR_mean_diff_pct_mean + np.percentile(sample, (
            confidence_level + (100 - confidence_level) / 2))

    # save results
    bam_results.append([f, n_train, n_test,
                        bam_VaR_mult_mean, bam_VaR_mult_CIl, bam_VaR_mult_CIu,
```

```
                            bam_CVaR_mult_mean, bam_CVaR_mult_CIl, bam_CVaR_mult_CIu,
                            VaR_exceeded_pct_mean,
                            VaR_exceeded_pct_u_CIl, VaR_exceeded_pct_u_CIu,
                            VaR_exceeded_pct_l_CIl, VaR_exceeded_pct_l_CIu,
                            CVaR_mean_diff_pct_mean,
                            CVaR_mean_diff_pct_u_CIl, CVaR_mean_diff_pct_u_CIu,
                            CVaR_mean_diff_pct_l_CIl, CVaR_mean_diff_pct_l_CIu])

df_bam_results = pd.DataFrame(bam_results,
                              columns=['period_in_h', 'train_size', 'test_size',
                                       'bam_VaR_mult', 'bam_VaR_mult_CIl', 'bam_VaR_mult_CIu',
                                       'bam_CVaR_mult', 'bam_CVaR_mult_CIl', 'bam_CVaR_mult_CIu',
                                       'VaR_exceeded_pct_mean',
                                       'VaR_exceeded_pct_u_CIl', 'VaR_exceeded_pct_u_CIu',
                                       'VaR_exceeded_pct_l_CIl', 'VaR_exceeded_pct_l_CIu',
                                       'CVaR_mean_diff_pct_mean',
                                       'CVaR_mean_diff_pct_u_CIl', 'CVaR_mean_diff_pct_u_CIu',
                                       'CVaR_mean_diff_pct_l_CIl', 'CVaR_mean_diff_pct_l_CIu'])

df_bam_results
```

Non-exclusive licence to reproduce thesis and make thesis public

I, Anna-Helena Salurand,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, "Validation and development of risk metrics for the currency portfolio of a global FinTech company", supervised by Assoc. Prof. Raul Kangro and PhD Egor Kraev.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Anna-Helena Salurand
25/05/2021