

Value at Risk (VaR)

Conditional Value at Risk (CVaR)

VaR

Die Value at Risk-Berechnung gibt eine statistische Schätzung zum finanziellen Risiko, also die Höhe des möglichen Verlustes zu einer gewissen Wahrscheinlichkeit, (z.B. 95%, 99% oder 99.9%...) in einem gewissen Zeitraum (z.B. 1 Tag, 2 Wochen...) aus.

Es gibt drei gängige Methoden zur Berechnung des Value at Risk:

Historischer VaR

Zur Berechnung des historischen Value at Risk werden erstens die vergangenen Schlusskurse der gewählten Anlage/Aktie benötigt.

Beispielhaft werden die historischen Schlusskurse einer Aktie des gesamten vergangenen Jahres herangezogen, die Renditen berechnet und von kleinstem nach größtem Wert geordnet.

Für den VaR(95), werden 5% der Anzahl an erfassten Tagen berechnet ($365 \cdot 0.05 = 18.25$). Im Beispiel würde nun die Rendite des achtzehnten schlechtesten Tages in Bezug auf Rendite hergenommen und als VaR(95) würde die Rendite dieses Tages definiert werden.

Im Falle eines Portfolios werden die summierte Rendite aller Anlagen berechnet und geordnet.

Der Vorteil der historischen VaR-Berechnung ist die simple Kalkulation während er jedoch nur langsam auf sich schnell ändernde Markt-Verhältnisse reagiert.

Varianz-Covarianz VaR/Delta-Normal-Ansatz

Wie in der historischen VaR-Berechnung sind erstens die vergangenen Schlusskurse der gewählten Anlagen/Aktien im Portfolio nötig.

Aus diesen Schlusskursen wird der Durchschnitt und die Varianz der Rendite mithilfe der Covarianz-Matrix (Die Covarianz der Rendite gibt aus, ob die Preise der Anlagen sich zusammen/invers zueinander bewegen oder ob kein Verhältnis besteht) und den Gewichtungen der Assets im Portfolio berechnet. Die Varianz ist ein statistisches Maß des Risikos und gibt an wie stark die einzelnen Werte vom Durchschnitt abweichen.

Die Standardabweichung kann aus der Wurzel der Varianz des Portfolios berechnet werden und gibt die durchschnittliche Abweichung aller Werte vom Mittelwert an.

Der Value at Risk wird letztendlich aus der inversen Normalverteilung des Konfidenz-Intervalls (bei VaR(95) beispielweise 5%) mal der Standardabweichung plus der Durchschnittsrendite des Portfolios berechnet.

Um den Value at Risk auf verschiedene Zeiträume anzupassen kann der VaR für einen Tag einfach mal der Wurzel des erwünschten Zeitraums gerechnet werden.

Monte-Carlo-Simulation

Die Monte-Carlo-Simulation-Berechnung des VaR versucht diesen mithilfe einer Reihe an zufällig generierten Marktpreisänderungen und der aus dieser entstehenden Distribution an Werten zu berechnen.

Mithilfe einer zufälligen Zahl werden mehrmals (meist mehrere tausend Mal) mögliche Preisänderungen mithilfe der Volatilität (Standardabweichung * gewünschtem Intervall) generiert und aufgezeichnet. Der Wert, der dann an der Grenze des Konfidenzintervalls liegt, liefert den berechneten Value at Risk.

CVaR

Conditional Value at Risk ist eine Weiterentwicklung der Value at Risk-Berechnung und beschreibt den durchschnittlichen Maximalverlust, beim Eintreten des „Worst-Case-Szenarios“ des VaR. Beispielweise also der durchschnittlichen Maximalverlust der 5% schlimmsten Fälle bei einem Konfidenzintervall von 95% (VaR(95)).

Backend Development

Python

Python ist eine plattformunabhängige Programmiersprache, welche wegen seiner einfachen Syntax und der weiten Verbreitung gewählt wurde. Verglichen mit Programmiersprachen wie C oder C++ ist die Entwicklung in Python in den meisten Anwendungsfällen schneller bzw. einfacher.

Quandl

Quandl bietet Zugriff auf Finanzdaten als Datasets in Echtzeit und würde somit den Anforderungen des Projektes entsprechen.

pandas

Wie Quandl, bietet die Library pandas, die Möglichkeit, Markt-Daten direkt und in Echtzeit über die gewählte Schnittstelle, Yahoo Finance, zu downloaden.

NumPy

NumPy ist eine Library, welche multi-dimensionale Arrays und Matrizen, sowie komplexe mathematische Berechnungen unterstützt und somit für die Anforderungen des Projektes eine gute Wahl (Covarianz-Matrix, Inverse Normalverteilung, etc.)

Flask

Flask wird als DBC verwendet, um eine saubere Trennung zwischen Frontend und Backend zu gewährleisten. Das erleichtert die Wartbarkeit und die Programmierung der Software. Ein weiterer Punkt, welcher für Flask spricht, ist die Möglichkeit ein Webinterface, statt einem nativen Interface, zu erstellen.

Yahoo Finance – Schnittstelle

Für die Bereitstellung der Aktienkurse in Echtzeit wird auf den Service von Yahoo finance zurückgegriffen.

PyInstaller

PyInstaller erstellt aus Python Sourcecode einfach zu verwendende native ausführbare Dateien. Gewählt wurde PyInstaller, da es als einzige Software die neuste Python-Version unterstützt.

Frontend Development

HTML5

HTML (Hypertext Markup Language) ist eine Auszeichnungssprache, welche vor allem zur Darstellung von Webseiten, aber in den letzten Jahren auch für die Darstellung von Oberflächen in native Anwendungen verwendet wird.

Verglichen mit nativen Technologien wie die WinAPI oder Qt ist HTML flexibler und lässt sich auf mehr Plattformen, sowie in Webanwendungen darstellen und nutzen.

Bootstrap

Bootstrap ist zurzeit, das am meist verwendete CSS-Framework und zeichnet sich durch eine moderne und gewohnte Oberfläche aus.

Jquery

„Jquery wird in 73% der 10 Millionen beliebtesten Webseiten genutzt und wird laufend weiterentwickelt.“

jQuery ist eine schnelle, kleine (lightweight) und mächtige Javascript-Bibliothek, welche mit einer einfach zu erlernenden API die Lesbarkeit des Codes erhöht und den Programmieraufwand verringert.

Dokumentation

Trello

Zur Darstellung des Fortschritts des Projektes wird ein Kanbanboard verwendet, welches den aktuellen Zustand des Projektes (durch Gliederung der Aufgaben anhand ihres Zustandes, z.B. Doing, to review, finished, usw.) darstellt.

Github

Zur Versionierung (und gleichzeitig zur Zeit-/Aufwandsaufzeichnung) wird Github verwendet, somit wird das Zusammenarbeiten in Projekt vereinfacht.

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. -Git