

Projektseminar: Simulationsgestützte Überprüfung von Risikoanalysemethoden

Nils Generlich, Tillmann Krebs
betreut von Prof. Dr. rer. nat. Thomas Buder

26.02.2025

Inhaltsverzeichnis

1	Einführung	3
2	Grenzen von Risiko-Matrizen	4
3	Entwurf eines Benchmark-Konzeptes	6
3.1	Annahmen/Vorraussetzungen	6
3.2	Kriterien	6
3.2.1	Range Compression-Prüfung	6
3.2.2	Risikowertüberschneidungen zwischen Klassen (Overlap)	7
3.2.3	Quantifying Errors (Quantitative Fehler)	7
3.2.4	Rangreihenfolge (Ordnungsmaß)	8
3.2.5	Weitere bewertbare Probleme von Risikomatrizen	9
3.3	Gesamt-Benchmark-Score	9
3.4	Beispiel: optimale 5x5-Matrix von Cox (1)	10
3.5	Beispiel 2: DIN EN 50126	11
4	Erstellung von Simulations-Applikation	12
4.1	Ziel des Programms	12
4.2	Umsetzung	12
4.3	Funktionen	12
4.3.1	Hauptseite	12
4.3.2	Matrix-Erstellen Seite	13
5	Statistische Auswertung des Benchmarks	14
6	Reflexion und Selbstkritik	16
7	Fazit	18
8	Anlagen	19
8.1	Quellenverzeichnis	19
8.1.1	Literaturquellen	19
8.1.2	Bildquellen	19
8.2	Tabelle für statistische Auswertung des Benchmarks mit 17 Risikomatrizen	20
8.3	Benutzerdokumentation für Simulationsprogramm	20
8.4	Programmdokumentation für Simulationsprogramm	20

1 Einführung

Risikomatrizen werden häufig zur qualitativen Bewertung von Risiken verwendet und finden Anwendung in einer Vielzahl von Branchen. Sie bieten eine intuitive, visuelle Darstellung von Risiken, indem sie Eintrittswahrscheinlichkeit und Schadensausmaß in einer Matrix kombinieren. Trotz ihrer Popularität gibt es jedoch zahlreiche wissenschaftliche und praktische Kritikpunkte an ihrer Anwendung. So können Risikomatrizen unter bestimmten Bedingungen zu inkonsistenten oder sogar irreführenden Bewertungen führen.

Um die Qualität und Aussagekraft von Risikomatrizen besser bewerten zu können, ist eine systematische Analyse erforderlich. Im Rahmen des Projektseminars "Risikoanalyse" untersuchten wir die Grenzen von Risikomatrizen und stellen ein Benchmarking-Konzept vor, das anhand quantitativer Kriterien verschiedene Matrizen objektiv bewertet. Ziel ist es, ein Bewertungsverfahren zu entwickeln, das eine fundierte Aussage über die Eignung und Genauigkeit von Risikomatrizen in unterschiedlichen Anwendungsszenarien erlaubt.

Im weiteren Verlauf der Arbeit werden zunächst die wesentlichen Schwächen bestehender Risikomatrizen beleuchtet. Anschließend wird ein Benchmark-Ansatz zur Evaluierung ihrer Qualität vorgestellt, der verschiedene Kriterien wie Informationsverlust (Range Compression), Rangreihenfolgefehler und Überschneidungen zwischen Risikoklassen berücksichtigt. Zur Validierung dieses Konzepts wurde eine Simulationsapplikation entwickelt, mit der sich Risikomatrizen systematisch analysieren und vergleichen lassen. Abschließend erfolgt eine statistische Auswertung der Benchmark-Ergebnisse sowie eine kritische Reflexion des Ansatzes.

2 Grenzen von Risiko-Matrizen

Risikomatrizen sind ein weit verbreitetes Instrument in der Analyse und der Bewertung von Risiken. Sie finden Anwendung in den vielfältigsten Branchen, worunter die Luftfahrt, die Öl- und Gasindustrie (Thomas¹, 2014, S. 56), das Bauwesen oder auch das Projektmanagement(Cox², 2008, S. 497) zählt. Dies verdanken sie vor Allem ihrer einfachen Darstellung und, durch ihre farblich abgestuften Risikoklassen, auch ihrer Intuitivität und ihrer Eingängigkeit (Fekete³, 2024, S. 174). Trotz ihrer Popularität gibt es aber erhebliche situativ abhängige und mathematische Einschränkungen, die ihre Aussagekraft und Zuverlässigkeit infrage stellen.

Risikomatrizen können unter Anderem paradoxerweise dazu führen, dass kleinere Risiken als gravierender eingestuft werden als größere. Dies hängt stark von der gemeinsamen Verteilung der Häufigkeit und des Schweregrads von Risiken ab. Ist die Korrelation zwischen diesen beiden Parametern positiv – das heißt, hohe Wahrscheinlichkeiten treten mit hohen Konsequenzen auf – kann die Matrix eine relativ zuverlässige Bewertung liefern. Liegt jedoch eine negative Korrelation vor (hohe Wahrscheinlichkeit mit niedrigen Konsequenzen und umgekehrt), so sind Risikomatrizen oft nicht besser als zufällige Entscheidungen. Verständlich wird dies durch ein einfaches Beispiel, für welches von einem quantitativen Risiko ausgegangen wird, welches als Produkt von Häufigkeit und Ausmaß errechenbar ist: $Risiko = Häufigkeit \cdot Schweregrad$. Betrachtet wird eine 2x2-Matrix mit einer genormten Skala von 0 bis 1, welche in 3 Risikoklassen unterscheidet und Risiken mit einer Häufigkeit und einer Schwere, die kleiner als 0.5 sind, in "gering" einteilt. Risiken, welche eine Häufigkeit und einen Schweregrad von über 0.5 haben, werden als "hoch" eingeteilt. Liegt ein Wert der Attribute unter 0.5 und einer darüber, so wird es als "mittel" kategorisiert. Bei dieser Matrix kann es dazu kommen, dass ein Risiko, welches eine geringe Eintrittswahrscheinlichkeit, aber im Falle eines Eintritts hohe Konsequenzen mit sich zieht, als mittleres Risiko eingestuft wird. Dies passiert beispielsweise mit dem Risikopunkt $r_1 = (Häufigkeit, Ausmass) = (0.1, 0.55)$. Gleichzeitig kann aber ein Risiko mit mittlerer Wahrscheinlichkeit und Konsequenz als geringes qualitatives Risiko eingestuft werden. Das ist bei dem Risikopunkt $r_2 = (Häufigkeit, Ausmass) = (0.45, 0.45)$ der Fall. Obwohl der quantitative Risikowert des ersten Risikos ($R_{quant,1} = 0.1 \cdot 0.55 = 0.055$) mehr als 3 mal kleiner als der quantitative Risikowert des zweiten Risikos ($R_{quant,2} = 0.45 \cdot 0.45 = 0.2025$) ist, erhält er von der 2x2-Matrix einen ordinalskaliert höheren qualitativen Wert zugewiesen. In solchen Situationen kann die Nutzung von Risikomatrizen zu systematisch falschen Entscheidungen führen, die schlechter sind als zufällige Bewertungen (Cox⁴, 2008, S. 500).

Ein weiteres Problem stellt die Range Compression beziehungsweise die zu geringe qualitative Auflösung von Risiken dar. Die Anzahl der Risikoklassen/-kategorien in einer Risikomatrix bestimmt ihre Auflösung. Bei zu wenigen Risikoklassen („Range Compression“) werden quantitativ stark unterschiedliche Risiken gleich kategorisiert. Gleichzeitig ist aber auch der Einsatz von zu vielen Risikokategorien ein Problem, da damit Rangreihenfolgefehler entstehen können. Bei diesen Rangreihenfolgefehlern werden kleinere quantitative Risiken höher priorisiert beziehungsweise mit einem ordinalskaliert höheren qualitativen Wert versehen, als größere quantitative Risiken (Cox², 2008, S. 506). Dies kann dazu führen, dass Entscheidungen auf Basis der Risikomatrix ineffizient oder gar kontraproduktiv sind.

Studien haben gezeigt, dass Menschen dazu neigen, mittlere Werte zu wählen, wenn sie qualitative Einschätzungen treffen, was auch "Centering Bias" genannt wird (Thomas¹, 2014, S.59). Dies kann dazu führen, dass die Risikomatrix künstlich viele Risiken in der mittleren Kategorie anzeigt und extreme Risiken unter- oder überschätzt werden. Damit ist die subjektive Bewertung von Häufigkeit und Schweregrad ein weiteres zentrales Problem bei der Anwendung von Risikomatrizen. Unterschiedliche Anwender können das gleiche Risiko damit unterschiedlich einstufen, was die Vergleichbarkeit erheblich beeinträchtigt (Cox², 2008, S. 508).

Die Einfachheit und Intuitivität der Risikomatrizen bringt auch nicht nur Vorteile mit sich. Das zeigt die oftmals fehlende Translationsinvarianz: Kleine gleichmäßige Änderungen in den Risikobewertungen können den Rang/ die Klasse der Risiken widersprüchlich verändern (Cox², 2008, S. 506 f.). Eine voll-

¹Philip Thomas, Reidar B. Bratvold, J. Eric Bickel (April 2014). The Risk of Using Risk Matrices [siehe Abschnitt "Summary"]. SPE Economics & Management, 56–66

²Jr., Louis A. Cox (2008). What's wrong with risk matrices?. Risk Analysis, 28 (2), 497–512

³Fekete, Alexander (2024). Risiko, Katastrophen und Resilienz. Springer-Verlag GmbH. <https://doi.org/10.1007/978-3-662-68381-1>

⁴Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Abschnitt "A NORMATIVE DECISION-ANALYTIC FRAMEWORK"]. Risk Analysis, 28 (2), 497–512

ständige Konsistenz kann so mit einer einfachen ordinalen Metrik, wie sie von Risikomatrizen genutzt wird, nie erreicht werden. Denn es sollte nicht passieren, dass man in Wahrheit höhere Risiken niedrigere ordinale Werte zuordnet, als quantitativ niedrigere Risiken. Dies kann neben der bereits erwähnten Korrelation der Parameter auch am "Overlapping" liegen. Dies bezeichnet die Überschneidung der abgedeckten quantitativen Risikobereiche mehrerer qualitativer Risikoklassen. Sind diese Überschneidungsräume groß, kann es zu höherer Inkonsistenz kommen (Krisper⁵, 2021, S. 8 f.). Außerdem fehlen quantitative Informationen, um Ressourcen optimal zur Risikominderung zu verteilen. Wichtige Faktoren wie die Kosten verschiedener Gegenmaßnahmen, Wechselwirkungen zwischen Risiken und Gegenmaßnahmen oder unterschiedlich erzielbare Risikoreduktionen werden nicht oder nicht ausreichend berücksichtigt (Cox⁶, 2008, S. 510).

Solche Einschränkungen können beim Risikomanagement gravierende Folgen aufweisen. So können Ressourcen fehlerhaft allokiert werden, wodurch sie ineffizient auf weniger kritische Risikofaktoren verteilt werden, während bedeutendere Bedrohungen unzureichend adressiert werden (Cox⁶, 2008, S. 507). Außerdem könnten die Risikomatrizen mit ihrer Systematik wie eine fundierte Entscheidungsgrundlage wirken, wodurch Entscheidungsträger im Risikomanagement sich zu stark auf eventuell ungenaue oder inkonsistente Bewertungen verlassen. Vor Allem bei komplexen oder hochkritischen Entscheidungen kann ein alleiniger Einsatz von Risikomatrizen zur Entscheidungsfindung gefährlich werden (Cox⁶, 2008, S. 5010).

Einige dieser Probleme oder Grenzen von Risikomatrizen können aber durch den gezielten Entwurf von optimierten Risikomatrizen minimiert werden. Um dies zu schaffen, ist ein Gütemaß über die Anwendbarkeit und Praktikabilität einer Risikomatrix sehr hilfreich, da mit solch einem Maß entworfene Matrizen untereinander verglichen werden können. Daher setzt sich das nachfolgende Kapitel mit einem Benchmark als Maßstab zur Bewertung von Risikomatrizen auseinander und greift dabei ein paar der nun bekannten Probleme dieser Risiko-Matrixstrukturen auf.

⁵Michael Krisper (2021). Problems with Risk Matrices Using Ordinal Scales. Institute of Technical Informatics Graz University of Technology

⁶Jr., Louis A. Cox (2008). What's wrong with risk matrices?. Risk Analysis, 28 (2), 497–512

3 Entwurf eines Benchmark-Konzeptes

3.1 Annahmen/Vorraussetzungen

Folgendes wird für die Anwendung des Benchmarks vorausgesetzt/angenommen, damit das bestimmte Ergebnis repräsentativ ist:

- quantitatives Risiko wird berechnet mit: $R_{quant} = Haeufigkeit \cdot Ausmass$
- zu bewertende Risikomatrix muss folgende Axiome erfüllen
 - **schwache Konsistenz**⁷: Risikomatrix kann zuverlässig zwischen sehr hohen und sehr niedrigen Risiken unterscheiden
 - * Lemma 1: keine angrenzenden roten und grünen Zellen in Matrix
 - * Lemma 2: keine roten Zellen in linker Spalte oder unterer Reihe der Matrix
 - **Axiom der Zwischenstufen**⁸: Abbildung von kontinuierlichen Risiko-Erhöhungen in Matrix, d.h. bei einem kontinuierlichen Anstieg des quantitativen Risikos von 0 (kein Risiko) bis 1 (höchstes Risiko) sollte die qualitative Kategorisierung stufenweise ansteigen (Existenz einer mittleren Risikostufe notwendig)
 - **Axiom der konsistenten Färbung**⁹:
 - * Zelle ist rot, wenn sie Punkte mit quantitativen Risiken enthält, die mindestens so hoch sind, wie anderen roten Zellen und keine Punkte mit so niedrigen Risiken wie in grünen Zellen
 - * Zelle ist grün, wenn sie Punkte enthält, deren Risiken mindestens so niedrig sind, wie andere grüne Zellen und keine Punkte darin so hohe Risiken wie in roten Zellen vorweisen
 - * Zelle ist in Zwischenfarbe(n), wenn sie zwischen roter und grüner Zelle liegt oder sie sowohl Punkte mit höheren Risiken als einige rote Zellen, als auch Punkte mit niedrigeren Risiken als einige grüne Zelle enthält

3.2 Kriterien

3.2.1 Range Compression-Prüfung

Wie groß ist das abgedeckte quantitative Risiko je Klasse?

Vorgehensweise

- für jede qualitative Klasse C_j (mit $j \in 1, \dots, k$, wobei k =Anzahl qualitativer Risikostufen/-klassen) werden minimale und maximale quantitative Risikowerte, die in dieser Klasse liegen, bestimmt $Range_j = \max(R_{quant} | C_{qual} = j) - \min(R_{quant} | C_{qual} = j)$
- Durchschnittliche Range der qualitativen Klassen einer Risikomatrix berechnen:
 $AvgRange = \frac{1}{k} \cdot \sum_{j=1}^k Range_j$
- Zusätzlich sollten die $Range_j$ -Werte möglichst gleichgroß sein, um übermäßige Informationsverluste in spezifischen Klassen zu vermeiden. Daher wird der Durchschnitt der Differenzen der $Range_j$'s für jede Klasse C_j mit $j \in 1, \dots, k$ errechnet:

für $i=1$ bis $i=k-1$

für $j=i+1$ bis $j=k$

-> Berechne $diff_{i,j} = |Range_i - Range_j|$

Berechne nun den Durchschnitt mit Hilfe der Summe aller Differenzen $diff_{i,j}$:

$$AvgDiff = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^k diff_{i,j}}{\frac{k \cdot (k-1)}{2}}$$

⁷Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497–512

⁸Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Abschnitt: "The Betweenness Axiom: Motivation and Implications" auf Seite 503]. Risk Analysis, 28 (2), 497–512

⁹Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Abschnitt: "Consistent Coloring" auf Seite 503 f.]. Risk Analysis, 28 (2), 497–512

- **Score für Range Compression:** $Score_{Range} = 1 - \frac{0.7 \cdot AvgRange + 0.3 \cdot AvgDiff}{\max(R_{quant}) - \min(R_{quant})}$, d.h. bei Normierung der Häufigkeits- und Ausmassklassen auf 0..1: $Score_{Range} = 1 - 0.7 \cdot AvgRange - 0.3 \cdot AvgDiff$
 - $Score_{Range} = 0 \dots$ maximaler Informationsverlust (d.h. nur eine Klasse)
 - $Score_{Range} = 1 \dots$ kein Informationsverlust (Klassenbreite jeder Klasse $\rightarrow 0$)

3.2.2 Risikowertüberschneidungen zwischen Klassen (Overlap)

Wie groß sind die Überschneidungen der Klassen bezüglich quantitativen Risikowerten, in welchen Rangreihenfolgefehler passieren können?

Vorgehensweise

- Klassen nach ihrer Risikohöhe aufsteigend ordnen
- für jede qualitative Klasse C_j das Intervall $I_j = [\min(R_{quant}|C = j), \max(R_{quant}, C = j)]$ bestimmen
- für $j=1$ bis $j=k-1$, wobei k =Anzahl qualitativer Risikostufen/-klassen für $x=1$ bis $x=k-j$
 - > Berechne $Overlap_{j,j+x} = \max(0, |\min(R_{quant}|C = j+x) - \max(R_{quant}|C = j)|)$
- Gesamtüberschneidung:

$$TotalOverlap = \sum_{j=1}^{k-1} (\sum_{x=1}^{k-j} x \cdot Overlap_{j,j+x})$$

$$MaxOverlap = (1 + \dots + k-1) + (1 + \dots + k-2) + \dots + (1) = (k-1) \cdot (1) + (k-2) \cdot (2) + \dots + (1) \cdot (k-1) = \sum_{x=1}^{k-1} (k-x) \cdot x$$
 Auf die Formel für MaxOverlap kommt man, indem man ein maximales Überlappen vom gesamten quantitativen Risiko-Wertebereich für jedes Overlap einsetzt.
- **Score für Risikowerte-Überschneidung:** $Score_{Overlap} = 1 - \frac{TotalOverlap}{MaxOverlap}$
 - $Score_{Overlap} = 0 \dots$ Es existieren maximale Überschneidungen, d.h. alle Klassen überlappen sich vollständig \rightarrow schlechtester Fall
 - $Score_{Overlap} = 1 \dots$ Es existieren keine Überschneidungen der quantitativen Risikowerte zwischen qualitativen Klassen \rightarrow idealer Fall

3.2.3 Quantifying Errors (Quantitative Fehler)

Identifikation der Anzahl und des Einflusses von kritischen Übergangsstellen zwischen Zellen unterschiedlicher Klasse

Vorgehensweise

- an jedem "Kreuz" der Matrix" (:= Punkt, an welchem 4 Zellen der Risikomatrix aufeinander treffen) wird ein Punkt simuliert/betrachtet
- *Simulationsgestützter Ansatz:*
 - Je "Kreuz"punkt wird eine Simulation von ≥ 10.000 normalverteilt generierten Zufallspunkten mit Erwartungswert $\hat{=}$ Punkt und Varianz $\hat{=}$ 1% Fehlertoleranz durchgeführt.
 - Pro Punkt i ist $QuantError_i = \text{Anz. Klassen mit zugewiesenen Punkten} > 0-0.1\%$
 - \rightarrow Erweiterung: Mehrere Durchläufe j je "Kreuz"punkt mit variabler Fehlertoleranz von z.B.: 1%, 5%, 10% ausführen. Anschließend wird $QuantError_i$ als Mittelwert aus allen $QuantError_{ij}$ gebildet: $QuantError_i = \frac{1}{J} \cdot \sum_{j=1}^J QuantError_{ij}$ mit j =Anzahl Durchläufe je "Kreuz"punkt
- *theoretischer Ansatz:* pro "Kreuz"punkt i ist $QuantError_i$ die unterschiedliche Anzahl der Klassen, an welchen dieser Punkt angrenzt
- $Score_{quantError} = 1 - \frac{\sum_{i=1}^k QuantError_i}{\sum_{i=1}^k \max(QuantError_i)} = 1 - \frac{\sum_{i=1}^k QuantError_i}{k \cdot 4}$ mit k =Anzahl der "Kreuz"punkte der Matrix

3.2.4 Rangreihenfolge (Ordnungsmaß)

Wie stark (positiv) korreliert ist die Abbildung von quantitativen Risiken auf qualitative Risiken durch die Risikomatrix?

Vorgehensweise

- Simulation von ≥ 10.000 gleichverteilten Punkten (im Bereich von $[0,1]$ für Ausmass und Häufigkeit) und Hinzufügen dieser zu einem Array
- jeweils die quantitativen Risiken pro Punkt errechnen und anhand dieser quantitativen Risiken die jeweiligen Punkte im Array aufsteigend sortieren
- jedem Punkt seiner qualitativen Risikoklasse in der Matrix zuordnen
- Betrachtung der Reihenfolge der Risikoklassen
- Bewertung dieser Reihenfolge mit Hilfe des Spearman'schen Rangkorrelationskoeffizienten:
 - **Spearman'scher Rangkorrelationskoeffizient** r_s misst die Stärke und Richtung einer monotonen Beziehung zwischen zwei Variablen. Diese beiden Variablen sind hier die qualitativen und die quantitativen Risikowertefolgen. Er ist vor Allem dann nützlich, wenn die Beziehung nicht linear, sondern nur monoton ist. Also beispielsweise, wenn die quantitative Zahlenfolge monoton steigt und die andere ebenfalls, jedoch nicht dauerhaft und nicht in einem konstanten Verhältnis, wie es hier der Fall ist.
 - Prüfung, ob Anstieg in der einen Variablen/Zahlenfolge mit einem Anstieg oder Abfall der anderen Variable/Zahlenfolge einhergeht. Dabei basiert die Berechnung auf den Rängen der Daten und nicht auf den absoluten Werten, wodurch dieser Koeffizient robust gegenüber Ausreißern und nicht-linearen Zusammenhängen ist.
 - Wertebereich:
 - * $r_s = +1$: Perfekte positive Rangkorrelation (höhere Werte in der einen Variable entsprechen immer höheren Werten in der anderen)
 - * $r_s = -1$: Perfekte negative Rangkorrelation (höhere Werte in der einen Variable entsprechen immer niedrigeren Werten in der anderen) \rightarrow bei diesem Anwendungszweck nicht erreichbar
 - * $r_s = 0$: keine Rangkorrelation
 - Formel: $r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$, wobei:
 - d_i ... Differenz der Ränge zwischen beiden Variablen für das i-te Paar (ein Paar sind die beiden Werte (qualitativ, quantitativ) an der gleichen Array-/Listenposition)
 - n ... Anzahl der Datenpaare (Länge der Liste an gleichverteilten Punkten, d.h. ≥ 10.000)
 - Bei mehreren gleichen Werten, wird der mittlere Rang für diese Werte verwendet, was meist zu einer Verringerung der Stärke der Korrelation führt. Da dies bei den qualitativen Werten sehr oft der Fall ist, wird die maximal mögliche Korrelation errechnet, indem die qualitative Zahlenfolge im idealen Fall auch sortiert wird und mit diesen beiden sortierten Listen jeweils der bestmögliche Rangkorrelationskoeffizient $r_{s,opt}$ errechnet wird.
 - Mit Einbezug von $r_{s,opt}$ wird der Score für dieses Kriterium wie folgt gebildet:
$$Score_{Ordnung} = \frac{r_s(X,Y)}{r_{s,opt}(X,Y)} \text{ für } r_s(X,Y) > 0, \text{ sonst } Score_{Ordnung} = 0$$
- Ergebnis:
 - Idealfall $Score_{Ordnung} = 1$: durchgehend monoton steigende qualitative Risikoklassen im Array (mit zunehmenden Array-Index)
 - Worst-Case $Score_{Ordnung} = 0$: zufälliger Mix an qualitativen Risikoklassen mit aufsteigendem Array-Index

3.2.5 Weitere bewertbare Probleme von Risikomatrizen

Nachfolgend sind weitere Ideen für Kriterien aufgelistet, welche allerdings noch Probleme aufweisen und daher nicht für einen Einsatz in einem praktisch umgesetzten Benchmark ausgereift sind.

- **Neglecting Uncertainty** (Vernachlässigung der Unsicherheit (Krisper¹⁰, 2021, S. 6)): Wie sehr erhöht sich Unsicherheit/ Varianz des Häufigkeit-Schwere-Punktes bei Zuordnung zu einer Risikoklasse (durch den Informationsverlust)?
 - Simulieren einiger Punkte und für jeden dieser Punkte die Erhöhung/Veränderung der Unsicherheit nach qualitativer Einordnung errechnen
 - z.B. Veränderung der statistisch festgelegten Unsicherheit (Varianz) von z.B. 1% (was bei einem Intervall [0,1] 0.01 entspricht) zur Unsicherheit über die Intervalle der Klassen z.B. "vernachlässigbar" und "tolerabel" (wenn Simulation des Punktes ähnlich viele Werte beiden Klassen zuordnete)
 - **Problem:** Wie kann die zahlentechnische Bewertung der qualitativen, wörtlich formulierten Klassen erfolgen?
- **Translationsinvarianz:**
 - (1) zufällige Punkte als Erwartungswert für eine Simulation nehmen und ein Wert des Punktes zufällig verringern
 - (2) anschließend für jeden Punkt schauen, ob eine qualitative Verringerung eintrat
 - wiederholen von (1) und (2) für jeweils anderen Wert des Punktes

3.3 Gesamt-Benchmark-Score

- Dabei sind 2 Optionen möglich:
 - $BenchmarkScore = \frac{Score_{Range} + Score_{Overlap} + \dots}{n}$ mit n...Gesamtzahl Scores
 - bei Bedarf mit einfügen einer Gewichtung (gewichteter Mittelwert):
 $BenchmarkScore = \alpha \cdot Score_{Range} + \beta \cdot Score_{Overlap} + \dots$ mit $\alpha + \beta + \dots = 1$
- **Aktuelle Berechnung des Gesamt-Benchmark-Scores:** Aufgrund unterschiedlich zu erwartender Größe der einzelnen Scores, welche allerdings alle einen gleichen/ähnlichen Einfluss auf den Gesamtscore haben sollten, werden diese mittels Gewichtungsfaktoren $\alpha + \beta + \gamma + \delta = 1$ angepasst.
 - Als Referenzwerte werden ausgerechnete (siehe 3.4) und zum Teil simulierte Kriterienwerte der idealen Matrix von Cox (1) genutzt. Dabei haben die Kriterien-Scores folgende Werte: $Score_{Range} = 0.67$, $Score_{Overlap} = 0.92$, $Score_{quantError} = 0.625$. Da nur der Ordnungs-Score aufgrund der simulationsbasierten Wertermittlung je Benchmarkdurchlauf unterschiedliche Werte annimmt, wird nachfolgend nur für diesen Wert ein Erwartungswert approximativ mittels des Mittelwerts einer Stichprobe errechnet. Dafür wurden 10 Werte von 10 unterschiedlichen Benchmarkdurchläufen ermittelt: $X_1 = 0.9887$, $X_2 = 0.9901$, $X_3 = 0.9891$, $X_4 = 0.9874$, $X_5 = 0.9885$, $X_6 = 0.9893$, $X_7 = 0.9881$, $X_8 = 0.9892$, $X_9 = 0.9884$, $X_{10} = 0.9895$. Der Mittelwert \bar{x} berechnet sich aus

$$\mu_{ScoreOrdnung} \approx \bar{x} = \frac{1}{10} \sum_{i=1}^{10} X_i = \frac{9.8883}{10} = 0.9888.$$

Die Stichprobenvarianz s_x^2 berechnet sich aus

$$s_x^2 = \frac{1}{10-1} \sum_{i=1}^{10} (X_i - \bar{x})^2 \approx \frac{1}{9} \cdot 5.37 \times 10^{-6} = 5.97 \times 10^{-7}.$$

Da die einzelnen Messwerte sehr nahe am Mittelwert liegen, ist die Stichprobenvarianz, also die Streuung der Werte, mit 5.97×10^{-7} äußerst gering. Dies zeigt, dass die Score-Daten des Ordnungskriteriums nur minimale Abweichungen aufweisen. Daher liefern bereits 10 Testdaten eine verlässliche Schätzung des, nachfolgend auf nur 2 Nachkommastellen gerundeten, Erwartungswertes.

¹⁰Michael Krisper (2021). Problems with Risk Matrices Using Ordinal Scales. Institute of Technical Informatics Graz University of Technology

- Gewichtungsfaktoren sollen umgekehrt proportional zu den Referenzwerten sein:
Es gilt daher mit $S = \frac{1}{Score_{Range}} + \frac{1}{Score_{Overlap}} + \frac{1}{Score_{quantError}} + \frac{1}{\mu_{ScoreOrdnung}} = 5.1896$:

$$\alpha = \frac{\frac{1}{Score_{Range}}}{S} = 0.2876, \beta = \frac{\frac{1}{Score_{Overlap}}}{S} = 0.2094, \gamma = \frac{\frac{1}{Score_{quantError}}}{S} = 0.3083,$$

$$\delta = \frac{\frac{1}{\mu_{ScoreOrdnung}}}{S} = 0.1946$$
- **BenchmarkScore** = $\alpha \cdot Score_{Range} + \beta \cdot Score_{Overlap} + \gamma \cdot Score_{quantError} + \delta \cdot Score_{Ordnung} =$

$$0.2876 \cdot Score_{Range} + 0.2094 \cdot Score_{Overlap} + 0.3083 \cdot Score_{quantError} + 0.1946 \cdot Score_{Ordnung}$$

3.4 Beispiel: optimale 5x5-Matrix von Cox (1)

Prob\Consequence	0-0.2	0.2-0.4	0.4-0.6	0.6-0.8	0.8-1
0.8-1	Green	Green	Yellow	Red	Red
0.6-0.8	Green	Green	Yellow	Yellow	Red
0.4-0.6	Green	Green	Green	Yellow	Yellow
0.2-0.4	Green	Green	Green	Green	Green
0-0.2	Green	Green	Green	Green	Green

eine der zwei optimalen 5x5-Matrizen von Louis A. Cox ¹¹

Nachfolgend werden die vorherig beschriebenen Algorithmen und Rechenvorschriften für eine beispielhafte Rechnung benutzt.

- **Range Compression:**

$$Range_1 = 0.4 - 0 = 0.4, Range_2 = 0.64 - 0.24 = 0.4, Range_3 = 1 - 0.48 = 0.52$$

$$\rightarrow AvgRange = \frac{1}{3} \cdot (0.4 + 0.4 + 0.52) = 0.44$$

$$diff_{1,2} = 0, diff_{1,3} = 0.12, diff_{2,3} = 0.12$$

$$\rightarrow AvgDiff = \frac{0.12+0.12}{3} = 0.08$$

$$Score_{Range} = 1 - \frac{0.7 \cdot 0.44 + 0.3 \cdot 0.08}{1} = 0.668$$

- **Overlap:**

$$I_1 = [0, 0.4], I_2 = [0.24, 0.64], I_3 = [0.48, 1]$$

$$\rightarrow j = 1 :$$

$$x = 1 : Overlap_{1,2} = 0.4 - 0.24 = 0.16,$$

$$x = 2 : Overlap_{1,3} = 0$$

$$\rightarrow j = 2 :$$

$$x = 1 : Overlap_{2,3} = 0.64 - 0.48 = 0.16$$

$$TotalOverlap = 1 \cdot 0.16 + 2 \cdot 0 + 1 \cdot 0.16 = 0.32$$

$$MaxOverlap = (3 - 1) \cdot 1 + (3 - 2) \cdot 2 = 4$$

$$Score_{Overlap} = 1 - \frac{0.32}{4} = 0.92$$

- **Quantifying Errors:**

$$Score_{quantError} = 1 - \frac{8 \cdot 1 + 8 \cdot 2}{16 \cdot 4} = 0.625$$

- **Rangreihenfolge/ Ordnungsmaß:**

Da dieses Kriterium durch Simulation mit über 10.000 Punkten bestimmt wird, wäre eine händische Berechnung zu aufwändig. Daher wird hierfür die Stichprobe der Größe $|X_n| = 10$ aus 3.3 genommen:

$$X_1 = 0.9887, X_2 = 0.9901, X_3 = 0.9891, X_4 = 0.9874, X_5 = 0.9885, X_6 = 0.9893, X_7 = 0.9881, X_8 = 0.9892, X_9 = 0.9884, X_{10} = 0.9895$$

$$\text{Bestimmung des Mittelwerts } \bar{x}: \bar{x} = \frac{1}{10} \sum_{i=1}^{10} X_i = \frac{9.8883}{10} = 0.9888 = Score_{Ordnung}.$$

- **Gesamt-Benchmark-Score:**

$$Score_{ges} = \alpha \cdot Score_{Range} + \beta \cdot Score_{Overlap} + \gamma \cdot Score_{quantError} + \delta \cdot Score_{Ordnung} = 0.2876 \cdot$$

$$0.668 + 0.2094 \cdot 0.92 + 0.3083 \cdot 0.625 + 0.1946 \cdot 0.9888 = 0.7699$$

¹¹Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Table IV in Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497-512

3.5 Beispiel 2: DIN EN 50126

Beispiel zur Darstellung von Risiken mittels Risikomatrix nach DIN EN 50126					
Häufigkeitsklassen	Risikostufen				
	Häufig				
	Wahrscheinlich				intolerabel
	Gelegentlich				
	Selten				unerwünscht
	Unwahrscheinlich				tolerabel
Ausmaßklassen	Unvorstellbar				Vernachlässigbar
	Unbedeutend	Marginal	Kritisch	Katastrophal	

4x6-Matrix der Norm DIN EN 50126

Wie bereits in 3.4 werden die vorherig beschriebenen Algorithmen und Rechenvorschriften für eine zweite beispielhafte Rechnung mit einer Matrix, welche nun 4 Risikoklassen besitzt, benutzt.

- **Range Compression:**

$$Range_1 = 0.1667 - 0 = 0.1667, Range_2 = 0.3333 - 0 = 0.3333, Range_3 = 0.5 - 0 = 0.5, Range_4 = 1 - 0.2083 = 0.7917$$

$$\rightarrow AvgRange = \frac{1}{4} \cdot (0.1667 + 0.3333 + 0.5 + 0.7917) = 0.4479$$

$$diff_{1,2} = 0.1666, diff_{1,3} = 0.3333, diff_{1,4} = 0.625, diff_{2,3} = 0.1667, diff_{2,4} = 0.4584, diff_{3,4} = 0.2917,$$

$$\rightarrow AvgDiff = \frac{0.1666 + 0.3333 + 0.625 + 0.1667 + 0.4584 + 0.2917}{6} = 0.3402$$

$$Score_{Range} = 1 - \frac{0.7 \cdot 0.4479 + 0.3 \cdot 0.3402}{1} = 0.5844$$

- **Overlap:**

$$I_1 = [0, 0.1667], I_2 = [0, 0.3333], I_3 = [0, 0.5], I_4 = [0.2083, 1]$$

$$\rightarrow j = 1 :$$

$$x = 1 : Overlap_{1,2} = 0.1667 - 0 = 0.1667,$$

$$x = 2 : Overlap_{1,3} = 0.1667 - 0 = 0.1667,$$

$$x = 3 : Overlap_{1,4} = 0$$

$$\rightarrow j = 2 :$$

$$x = 1 : Overlap_{2,3} = 0.3333 - 0 = 0.3333,$$

$$x = 2 : Overlap_{2,4} = 0.3333 - 0.2083 = 0.125$$

$$\rightarrow j = 3 :$$

$$x = 1 : Overlap_{3,4} = 0.5 - 0.2083 = 0.2917$$

$$TotalOverlap = (1 \cdot 0.1667 + 2 \cdot 0.1667 + 3 \cdot 0) + (1 \cdot 0.3333 + 2 \cdot 0.125) + (1 \cdot 0.2917) = 1.3751$$

$$MaxOverlap = (4 - 1) \cdot 1 + (4 - 2) \cdot 2 + (4 - 3) \cdot 3 = 10$$

$$Score_{Overlap} = 1 - \frac{1.3751}{10} = 0.8625$$

- **Quantifying Errors:**

$$Score_{quantError} = 1 - \frac{2 \cdot 1 + 10 \cdot 2 + 3 \cdot 3}{15 \cdot 4} = 0.4833$$

- **Rangreihenfolge/ Ordnungsmaß:**

Da dieses Kriterium durch Simulation mit über 10.000 Punkten bestimmt wird, wäre eine händische Berechnung zu aufwändig. Daher wird hierfür eine Stichprobe der Größe $|Y_n| = 10$ genommen, welche aus 10 Durchführungen der Benchmark-Berechnung mit dem erstellten Programm (siehe 4.) stammen:

$$Y_1 = 0.9266, Y_2 = 0.9237, Y_3 = 0.9222, Y_4 = 0.9278, Y_5 = 0.9230, Y_6 = 0.9300, Y_7 = 0.9273, Y_8 = 0.9246, Y_9 = 0.9223, Y_{10} = 0.9225$$

$$\text{Bestimmung des Mittelwerts } \bar{x}: \bar{x} = \frac{1}{10} \sum_{i=1}^{10} X_i = \frac{9.25}{10} = 0.925 = Score_{Ordnung}.$$

- **Gesamt-Benchmark-Score:**

$$Score_{ges} = \alpha \cdot Score_{Range} + \beta \cdot Score_{Overlap} + \gamma \cdot Score_{quantError} + \delta \cdot Score_{Ordnung} = 0.2876 \cdot 0.5844 + 0.2094 \cdot 0.8625 + 0.3083 \cdot 0.4833 + 0.1946 \cdot 0.925 = 0.6777$$

4 Erstellung von Simulations-Applikation

4.1 Ziel des Programms

Das Ziel des Programms ist es, Risikomatrizen anhand wissenschaftlicher Kriterien miteinander zu vergleichen und eine fundierte Aussage darüber zu treffen, welche Matrix in bestimmten Aspekten besser abschneidet als eine andere. Derzeit sind bereits drei Risikomatrizen implementiert: Zwei Risikomatrizen stammen von Louis A. Cox, eine entspricht der DIN-Norm 50126. Darüber hinaus ermöglicht das Programm den Nutzern, eigene Risikomatrizen zu erstellen. Diese können entweder mit den vorgefertigten Matrizen oder untereinander verglichen werden.

4.2 Umsetzung

Der erste Ansatz zur Untersuchung der Risikomatrix bestand darin, einen festen Punkt mit einer prozentualen Unsicherheit zu definieren und zu analysieren, wie stark die Risikoauswirkung variieren kann. Dazu entwickelten wir mithilfe von matplotlib und numpy ein Skript, das basierend auf einem Erwartungswert und einer Varianz eine Anzahl zufälliger Punkte erzeugt und diese auf einer Matrix visualisiert. Sowohl die Werte als auch die Matrix waren in dieser frühen Version noch fest im Code hinterlegt.

Als nächstes wollten wir eine interaktive Lösung entwickeln, die es ermöglicht, die Punkte flexibel einzustellen. Zudem sollte das Programm zwei Risikomatrizen gleichzeitig anzeigen können, um einen Vergleich zu ermöglichen. Da wir mit Webentwicklung vertrauter waren als mit klassischer GUI-Programmierung und uns eine Webanwendung größere Anpassungsmöglichkeiten bot, entschieden wir uns für eine Umsetzung mit Flask.

Ein HTML-Formular erlaubt nun die Eingabe von Datenpunkten, die an den Flask-Server übermittelt werden. Dort werden mit matplotlib die entsprechenden Diagramme generiert und anschließend in das SVG-Format konvertiert, sodass sie im Browser angezeigt werden können.

Ein weiteres Problem bestand darin, dass die Risikomatrix bisher direkt mit der Diagrammerstellung verknüpft war. Um mehrere Matrizen effizient verwalten zu können, musste die Erzeugung der Diagramme von der eigentlichen Datenstruktur entkoppelt werden. Dafür entwickelten wir eine eigene Matrix-Klasse, die als Datenmodell fungiert. Sie enthält alle relevanten Attribute zur Diagrammerstellung sowie eine Methode, um Datenpunkte den entsprechenden Feldern der Matrix zuzuordnen. Zudem extrahierten wir die Logik der Diagrammerzeugung aus dem bisherigen Code, sodass eine modulare und flexible Struktur entstand. Dies ermöglichte es, schnell neue Matrizen hinzuzufügen und später sogar Nutzern die Erstellung eigener Matrizen zu ermöglichen.

Daraufhin entwickelten wir eine neue Seite, auf der Nutzer ihre eigene Risikomatrix erstellen können. Diese Implementierung erwies sich als anspruchsvoll, da Risikomatrizen unterschiedliche Zeilen-, Spalten- und Risikoklassengrößen haben können. Wir lösten dies, indem wir dynamische HTML-Tabellen mit JavaScript auslesen und daraus eine neue Matrix-Entität generieren. Zusätzlich integrierten wir unser theoretisches Benchmark-Konzept aus Abschnitt 3, das verschiedene Eigenschaften einer gegebenen Matrix analysiert und einen Qualitätswert berechnet.

4.3 Funktionen

4.3.1 Hauptseite

Set Matrices	Auswahl der Risikomatrizen, die für den Benchmark verwendet werden sollen.
Set Simulation Parameters	<ul style="list-style-type: none">- Simuliert eine festgelegte Anzahl an Simulationen sowie einen Erwartungswert und eine Unsicherheit. Es werden Punkte berechnet, die einen Einsatzfall der Matrix simulieren.- Der Button „Submit“ startet die Simulation und die Berechnung des Benchmarks.
Benchmark	Zeigt nach dem Klicken des Submit-Buttons die Ergebnisse der Benchmark-Berechnung an.
Plots	Zeigt Diagramme an, anhand derer die Simulation ausgewertet werden kann.

4.3.2 Matrix-Erstellen Seite

Enter Matrix Name	- Eingabe des Namens der Matrix. - Dient der Identifikation einer Risikomatrix.
Set Risk Level	- Eingabe der Risikokategorien. - Jede Zeile entspricht einer Risikostufe, die in der Matrix abgebildet werden soll. - Eingabe der Farbbezeichnung für jede Risikokategorie. - Die kleinste Risiko-Nummer entspricht dem geringsten Risiko und die größte Nummer dem höchsten Risiko.
Define Matrix	- Auswahl der Anzahl der Zeilen und Spalten, die die Risikomatrix haben soll. - Durch Klicken auf „Generate Table“ wird eine Tabelle erzeugt, die der Risikomatrix entspricht. - Festlegung der Felder durch Eingabe der entsprechenden Risiko-Level-Nummer aus der vorherigen Tabelle.
Add Matrix	- Nach dem Klicken auf den Button erscheint eine Bestätigungsmeldung, die die erfolgreiche Erstellung der Matrix anzeigt. - Die Matrix kann nun auf der Hauptseite unter dem angegebenen Namen ausgewählt werden. - Bei einem Fehler wird eine entsprechende Fehlermeldung angezeigt.
Matrix Overview	- Zeigt alle gespeicherten Risikomatrizen an.

5 Statistische Auswertung des Benchmarks

Das entworfene Benchmark, welches nun mit Hilfe des Simulationsprogrammes schnell auf unterschiedlichste Risikomatrizen angewendet und errechnet werden kann, soll nun ausgewertet, getestet und untersucht werden. Ziel war es, unser Benchmark besser bewerten zu können und seine Grenzen zu erkennen. Dafür suchten wir 15 unterschiedliche bereits entworfene Risikomatrizen aus verschiedenen Quellen heraus und ließen den Benchmark-Score, mit Hilfe des in Abschnitt 4 erläuterten Programmes, ausrechnen. Anhand der daraus gewonnenen Informationen wurden darauffolgend 2 neue Risikomatrizen von uns entworfen, welche das Ziel haben, einen möglichst hohen Benchmarkscore zu erreichen.

Im ersten Teil haben wir uns verschiedene Risikomatrizen ($ID=\{1,\dots,15\}$) herausgesucht und das Benchmark auf diese angewendet. Wichtig war beim Heraussuchen, dass ein Großteil dieser Matrizen die Axiome, welche für die Anwendung unseres Benchmarks gelten sollten (siehe Abschnitt 3.1), erfüllt, um eine möglichst repräsentative Auswertung zu erreichen. Bis auf die Risikomatrix der Steuerkanzlei Steinhäuser¹², welche aufgrund angrenzender Zellen mit niedrigster- und höchster Risikoklasse das Axiom der schwachen Konsistenz nicht erfüllt, ist dies gelungen. Als Resultat stellt die Tabelle Risikomatrizen dar, welche aufsteigend nach ihrem Benchmark-Score sortiert wurden. Dabei werden in jeder Zeile für jeweils eine Risikomatrix die verschiedenen Scores aller Merkmale, die der Benchmark abdeckt, abgebildet. Dies kann im Abschnitt 8.2 eingesehen werden.

Hervorzuheben ist, dass die beiden optimalen Matrizen von Cox (mit der ID 5 und 7) von allen der 15 recherchierten Matrizen am besten abschneiden. Dies entspricht unseren Erwartungen, da die optimalen Matrizen mit 3 Risikoklassen für die Dimension 5x5 eine gute Auflösung bieten. Außerdem decken die Klassen den Wertebereich des quantitativen Risikos ungefähr zu ähnlichen Stücken ab. Die Anzahl der Risikostufen ist außerdem optimal, da auch der Quantifying Errors- und Ordnungs-Score hoch ist. So sieht man beispielsweise, dass die 5x5-Matrix mit der ID 13 einen deutlich schlechteren Quantifying Error-Score (0,5156) oder auch Ordnungs-Score (0,8670) hat. Dies ist darauf zurückzuführen, dass an mehr Stellen durch die hohe Anzahl an Klassen in dieser Dimensionalität u.A. auch deutlich gravierendere Rangreihenfolgefehler passieren können.

Außerdem ist zu erkennen, dass Matrizen mit einer kleineren Dimension meistens einen verhältnismäßig schlechten Gesamtbenchmarkscore aufweisen. Dies hängt damit zusammen, dass eine höhere Dimension eine Annäherung an die optimale quantitative Risikowertzuordnung darstellt. Je feiner die Abstufungen innerhalb einer Risikomatrix sind, desto präziser können Risiken quantifiziert und voneinander abgegrenzt werden.

Gleichzeitig zeigt die Analyse, dass Matrizen mit einer sehr hohen Anzahl an Risikoklassen nicht zwangsläufig die besten Ergebnisse liefern. So weist beispielsweise die 6x6-Matrix¹³ mit der ID 8 zwar eine hohe Granularität durch 5 Risikoklassen auf, jedoch geht dies zulasten des Quantifying Errors-Scores (0,4900). Dies verdeutlicht, dass eine zu feingliedrige Risikoeinteilung zu Problemen bei der konsistenten Zuordnung von Risiken führen kann, insbesondere wenn sich benachbarte Risikoklassen nur marginal in ihrem abgedeckten quantitativen Risikowertebereich unterscheiden. Daraus lässt sich ableiten, dass eine ausgewogene Balance zwischen der Anzahl der Risikoklassen und der Matrixdimension notwendig ist, um ein optimales Benchmark-Ergebnis zu erzielen.

Im zweiten Abschnitt haben wir die Werteverteilung der einzelnen Scores analysiert. Ziel dieser Untersuchung war es, genauere Erkenntnisse über die Spannweite der realistisch erreichbaren Wertebereiche zu gewinnen, um dadurch beispielsweise die bisherige Gewichtung des Benchmarks präziser bewerten zu können.

Die Ergebnisse zeigen, dass die Differenz zwischen dem minimalen und maximalen Wert der Scores einzelner Merkmale in den meisten Fällen zwischen 0,1 und 0,3 liegt. Diese relativ geringe Streuung lässt sich darauf zurückführen, dass der Benchmark primär auf Risikomatrizen angewendet wurde, die die drei zentralen Axiome (siehe Abschnitt 3.1) erfüllen. Da diese Axiome grundlegende Qualitätskriterien sichstellen, weisen die analysierten Matrizen bereits eine gewisse methodische Konsistenz und Aussagekraft auf.

Ein weiterer interessanter Aspekt ist, dass die Unterschiede in der Werteverteilung je nach Score-Metrik unterschiedlich stark ausgeprägt sind. Während einige Metriken, wie das Ordnungsmaß, eine

¹²Steuerkanzlei Steinhäuser. (o. J.). Erfolgsstrategien. Zuletzt abgerufen am 23. Februar 2025, von <https://www.steuerkanzlei-stehaeuser.de/erfolgsstrategien/>

¹³LinkedIn. (o. J.). Risk Matrix Horoscope – Best Practice. Zuletzt abgerufen am 23. Februar 2025, von <https://www.linkedin.com/pulse/risk-matrix-horoscope-best-practice->

vergleichsweise geringe Spannweite aufweisen, sind andere, wie der Quantifying Errors-Score, tendenziell stärker von der Struktur der jeweiligen Risikomatrix abhängig. Hier werden erste Grenzen des Benchmarks sichtbar, da beispielsweise die geringe Spannweite des Ordnungsscores darauf hindeutet, dass die aktuelle Gewichtung der einzelnen Metriken möglicherweise nicht ausreichend zwischen Matrizen unterschiedlicher Qualität differenziert.

Die bisherige Analyse lieferte unter Anderem wertvolle Erkenntnisse darüber, welche Eigenschaften eine effektive Risikomatrix auszeichnen und wie sich verschiedene Designentscheidungen auf die Qualität der Risikobewertung auswirken. Im Zuge dessen versuchten wir, durch gezielte Optimierung von Risikomodellen, Matrizen zu entwerfen, welche die optimalen 5x5-Matrizen von Cox¹⁴ in den vom Benchmark untersuchten Merkmalen übertreffen. Das erreichten wir, indem wir weitere Zeilen/ Spalten hinzufügten um die Dimensionalität zu erhöhen (Matrix ID=16). Mit der Erhöhung der Dimensionalität fügten wir für eine bessere Auflösung eine weitere Risikoklasse ein (Matrix ID=17). Durch das weitere Hinzufügen von Klassen und der Dimensionalitätserhöhung nähert sich die Matrix weiter der quantitativen Repräsentation an, welche keinen Informationsverlust aufweist. So erreichten die Matrix ID 16 und 17 schlussendlich bessere Benchmarkscores und zum Teil die besten Scores einzelner Merkmale.

Trotz der grundsätzlich erwartungsgemäßen Ergebnisse des Benchmarks traten in bestimmten Anwendungsfällen Ungereimtheiten auf. Diese wurden besonders deutlich, als wir versuchten, gezielt Matrizen zu konzipieren, bei denen die Scores einzelner Merkmale entweder den höchstmöglichen oder den niedrigstmöglichen Wert annehmen.

Ein bemerkenswertes Beispiel hierfür zeigte sich bei der Range-Compression: Obwohl eine Matrix in dieser Metrik extrem schlecht abschneiden kann, führt dies bei unseren dafür konzipierte Risikomatrix paradoxerweise zu einer Verbesserung der Scores für Overlap und Quantifying Errors. Dies liegt daran, dass in einer solchen Matrix nahezu alle Zellen der gleichen Risikoklasse zugeordnet werden, wodurch Überlappungen minimiert und Quantifizierungsfehler reduziert werden. Durch die Berechnung des gewichteten Mittelwerts, dessen Gewichtungsfaktoren auf Basis der ersten optimalen Cox-Matrix¹⁴ bestimmt wurden, wirkt sich eine Erhöhung des Quantifying-Errors-Scores stärker aus als eine Verschlechterung der Range-Compression. In der Folge kann es passieren, dass Matrizen, die in der Praxis kaum praktikabel wären, dennoch einen hohen Benchmark-Score erhalten.

Ein weiteres Problem ergibt sich bei Matrizen mit einem sehr hohen Range-Compression-Score, die dadurch entstehen, dass jede Zelle der Matrix einer eigenen Risikoklasse zugeordnet wird. Theoretisch sollte eine solche Matrix jedoch einen schlechten Overlap-Score aufweisen, da die Klassengrenzen vieler Risikoklassen in extremer Weise verschwimmen. Der Benchmark weist dieser Konstellation jedoch einen deutlich besseren Overlap-Score zu, als es aus konzeptioneller Sicht sinnvoll wäre.

Diese Beobachtungen zeigen wesentliche Grenzen des Benchmarks auf, die dessen Aussagekraft in speziellen Fällen einschränken. Weitere kritische Aspekte des Benchmarks-Modells sowie generelle Herausforderungen unserer Arbeit werden im nachfolgenden Kapitel detaillierter reflektiert.

¹⁴Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Table IV in Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497–512

6 Reflexion und Selbstkritik

Im Zuge der Entwicklung und Anwendung unseres Benchmark-Modells zur Bewertung von Risikomatrizen und der Auseinandersetzung mit der Zuverlässigkeit der Risikomatrizen wurden verschiedene Herausforderungen und Verbesserungspotenziale identifiziert. Diese betreffen sowohl methodische Aspekte der Benchmark-Konzeption, unsere grundlegende Annahmen und Betrachtungsweisen und unser Vorgehen als Ganzes. Um eine Weiterarbeit zu ermöglichen und unsere Arbeit einzuordnen, werden somit nachfolgend die wichtigsten Herausforderungen und möglichen Lösungsansätze dargestellt.

Einerseits zeigte die Analyse der Benchmark-Ergebnisse, dass die Spannweite des Ordnungsmaß-Scores relativ gering ist. Dies bedeutet, dass sich Matrizen, die sich qualitativ stark unterscheiden, in dieser Metrik nur minimal differenzieren. Da das Ordnungsmaß mit Hilfe des Spearman'schen Rangkorrelationskoeffizienten bestimmt wird, liegt die Ursache dieses Problems möglicherweise in der Art der Datengrundlage. Die untersuchten Matrizen erfüllen weitgehend die im Benchmark als vorausgesetzt angenommenen Axiome, wodurch ihre Ordnungsstruktur bereits eine hohe Qualität aufweist. Dies führt dazu, dass der Score bei den meisten Matrizen auf einem hohen Niveau bleibt. Das kann die Aussagekraft des Benchmarks verringern. Eine Anpassung dieser Berechnungsmethode könnte die Differenzierung verbessern. Beispielsweise könnte der Ordnungsscore basierend auf lokalen Rangfehlern oder gewichteten Differenzen errechnet werden, um eine bessere Abstufung der Bewertungen zu ermöglichen.

Ein weiteres methodisches Problem ergibt sich aus der Gewichtung der einzelnen Scores im Benchmark. Der gewichtete Mittelwert zur Berechnung des Gesamt-Benchmark-Scores basiert auf den Ausgabewerten der ersten optimalen Cox-Matrix. Dies führt in bestimmten Konstellationen zu unerwarteten Verzerrungen. Ein Beispiel ist die Range-Compression-Metrik: Eine Matrix mit einer extrem schlechten Range-Compression kann paradoxerweise einen hohen Benchmark-Score erhalten, da dies die Werte für Overlap und Quantifying Errors verbessert. Dies liegt daran, dass eine geringe Anzahl an Risikoklassen dazu führt, dass nahezu die gesamte Matrix einer einzigen Klasse zugeordnet wird, wodurch Überschneidungen reduziert und Quantifizierungsfehler minimiert werden. Da die Gewichtungsfaktoren auf Basis der Benchmark-Scores der optimalen Cox-Matrix¹⁵ bestimmt wurden, wird insbesondere der darin niedrig ausfallende Quantifying-Errors-Score stark gewichtet. In der Folge kann es passieren, dass Matrizen mit schlechter praktischer Anwendbarkeit dennoch hohe Benchmark-Werte erzielen. Als Folge dessen sollten die Gewichtungsfaktoren nicht ausschließlich aus den Werten der optimalen Cox-Matrix abgeleitet werden, sondern auf einer breiteren statistischen Basis beruhen. Eine Lösung wäre die Nutzung von Durchschnittswerten aus einer großen Stichprobe repräsentativer Risikomatrizen. Zudem könnten Matrizen mit extrem niedrigen Einzelwerten (z. B. ein Quantifying-Errors-Score unter 0,375) aus der Bewertung ausgeschlossen werden, da diese in der Praxis kaum relevant sind. Auch hierfür können globale Minimalwerte einer großen Stichprobe aus Risikomatrizen, die über eine gewisse Mindestqualität durch Axiom-Einhaltung verfügen, genutzt werden. Ein gering skaliertes Beispiel bietet hierzu die Tabelle aus Abschnitt 8.2.

Die konkrete Berechnung des Overlap-Scores zeigt eine weitere Konzeptionsschwäche des Benchmarks. Eine Matrix mit sehr vielen Risikoklassen sollte theoretisch einen niedrigen Overlap-Score aufweisen, da die Grenzen zwischen den Klassen stark verschwimmen. Tatsächlich ergeben sich hier jedoch oft überraschend gute Overlap-Scores, was auf die Art der Normierung zurückzuführen ist. Das Problem liegt in der Berechnung des MaxOverlap-Werts (siehe Abschnitt 3.2.2), der zur Normierung des Scores verwendet wird. Da dieser Wert die maximale theoretische Überschneidung der Klassen berücksichtigt, werden hohe TotalOverlap-Werte bei einer hohen Klassenanzahl rechnerisch abgeschwächt. Dies kann dazu führen, dass eine Matrix mit zufällig verteilten Risikoklassen einen besseren Overlap-Score erhält als eine methodisch sinnvoll aufgebaute Risikomatrix. Eine mögliche Verbesserung wäre die Einführung einer adaptiven Normierungsmethode, die nicht nur die Anzahl der Klassen, sondern auch die tatsächliche Verteilung der Überschneidungen in die Berechnung einbezieht. Wichtig hierbei ist aber auch noch der Fakt, dass eine solche Inkonsistenz vor Allem dann auftritt, wenn der Praxis fernliegende Risikomatrizen an das Benchmark angelegt werden. Mit der Voraussetzung, dass die 3 Axiome, wie in Abschnitt 3.1 benannt, erfüllt werden, wird dieses Problem nur noch sehr selten auftreten.

Eine weitere Einschränkung ergibt sich aus unserer sich durch das gesamte Projekt ziehenden Annahme einer linearen Achseneinteilung in Risikomatrizen. In der Praxis verhalten sich Risikofaktoren oft

¹⁵Jr., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Table IV in Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497–512

nicht linear und würden so besser mit nicht-linearen Skalen abgebildet werden können (z. B. logarithmisch oder polynomiell). Als Beispiel kann hier die Fahrgeschwindigkeit als Risikofaktor genannt werden. Durch Risikomatrizen modelliert, wird die Geschwindigkeit als linearer Risikofaktor wahrgenommen. Tatsächlich steigt das Risiko aber quadratisch oder exponentiell mit der Geschwindigkeit (Krisper¹⁶, 2021, S. 3 f.). Durch die derzeitige Implementierung des Benchmarks wird jedoch eine gleichmäßige Verteilung der Risikoklassen entlang der Achsen vorausgesetzt. Dies kann zu Ungereimtheiten führen, wenn in der Praxis andere Skalierungsansätze genutzt werden. Eine Erweiterung des Modells um nicht-lineare Achsen-skalierungen könnte also dazu beitragen, die Anwendbarkeit des Benchmarks auf realistische Risikobewertungen zu verbessern.

Ein weiterer kritischer Punkt ist die fehlende Einbindung eines realen Unternehmenskontexts. Das Benchmark wurde primär aus einer mathematischen Perspektive entwickelt, ohne dabei praktische Rahmenbedingungen oder konkrete Anwendungsfälle aus der Wirtschaft zu berücksichtigen. Beispielsweise berücksichtigt der Benchmark keine betriebswirtschaftlichen Faktoren, wie die Kosten-Nutzen-Relation oder die wirtschaftliche Tragfähigkeit bestimmter Risikomanagement-Strategien (Cox¹⁷, 2008, S. 510). Dadurch ist die Anwendbarkeit des Benchmarks in realen Entscheidungsprozessen begrenzt. Somit bietet das erstellte Programm Möglichkeiten zur Erweiterung. So könnte man ein ergänzendes Modul entwickeln, welches betriebswirtschaftliche Parameter einbezieht.

Ein wesentlicher kritisch anzumerkender Punkt betrifft unser Vorgehen bei der Programmentwicklung, beziehungsweise unsere gewählte Reihenfolge der Umsetzung. Die Entwicklung des Simulationprogrammes erfolgte in mehreren Schritten, wobei sich im Nachhinein gezeigt hat, dass eine frühere Definition der Matrixstruktur als Klasse hilfreich gewesen wäre. So hätte man gemäß der Grundprinzipien der Software-Entwicklung nach der Anforderungsanalyse beginnend einen Entwurf der Programmstruktur erstellen und diesen bei Bedarf agil anpassen sollen (Beneken¹⁸, 2022, S. 9 ff.). Dieser hätte uns die Implementierung in Python erleichtert und zugleich den Aufwand des Code-Refactorings gesenkt. Denn wir mussten mühsam die Simulation, Diagrammerstellung und Darstellung von der Matrixstruktur entkoppeln, nachdem wir bemerkten, dass die zusätzliche Implementierung des Benchmarks in diesem Programm sonst nicht effizient umsetzbar wäre. In zukünftigen Projekten sollten wir also einen strukturierten Entwicklungsansatz verfolgen, bei welchem wir bereits frühzeitig komponentenbasierte Entwürfe entwickeln, welche die Wiederverwendbarkeit einzelner Codeelemente beträchtlich erhöhen.

Die durchgeführte Analyse verdeutlicht, dass das entwickelte Benchmark-Modell wertvolle Einblicke in die Qualität von Risikomatrizen liefert. Gleichzeitig wurden aber methodische Grenzen identifiziert, die die Aussagekraft des Benchmarks und unseres Programms in bestimmten Fällen einschränken. Um diese Limitationen zu adressieren, sollte die zukünftige Forschungsarbeit insbesondere auf eine statistisch fundiertere Gewichtung der Metriken abzielen, die auf einer breiten Datenbasis beruht. Darüber hinaus ist eine Überarbeitung der Overlap-Score-Berechnung erforderlich, um methodische Verzerrungen zu minimieren. Ein weiterer wichtiger Aspekt wäre die Erweiterung des Modells um nicht-lineare Skalen, um realistische Risikostrukturen besser abzubilden. Zudem wäre eine empirische Validierung der Ergebnisse im Unternehmenskontext notwendig, um die Praxistauglichkeit des Benchmarks sicherzustellen. Durch diese Anpassungen könnte das Benchmark-Modell gemeinsam mit dem entwickelten Simulationprogramm zu einem noch präziseren und anwendungsorientierteren Instrument im Risikomanagement weiterentwickelt werden.

¹⁶Michael Krisper (2021). Problems with Risk Matrices Using Ordinal Scales. Institute of Technical Informatics Graz University of Technology

¹⁷Jr., Louis A. Cox (2008). What's wrong with risk matrices?. Risk Analysis, 28 (2), 497–512

¹⁸Beneken, G., Hummel, F., & Kucich, M. (2022). Grundkurs agiles Software-Engineering: Ein Handbuch für Studium und Praxis. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-37371-9>

7 Fazit

Die vorliegende Arbeit konzentrierte sich auf die Entwicklung eines objektiven Bewertungsverfahrens für Risikomatrizen. Basierend auf Erkenntnissen aus der Literatur wurden zentrale Schwächen von Risikomatrizen identifiziert und darauf aufbauend ein Benchmarking-Konzept entworfen, das verschiedene quantitative Metriken zur Bewertung der Qualität einer Matrix definiert. Die anschließende Umsetzung dieses Benchmarks als Simulationsapplikation ermöglicht eine systematische Analyse von Risikomatrizen und erlaubt Nutzern, eigene Risikomatrizen zu erstellen, zu bewerten und mit bewährten Matrizen zu vergleichen.

Die Tests mit verschiedenen Matrizen, darunter wissenschaftlich fundierte und einer Risikomatrix aus der DIN Norm 50126, haben gezeigt, dass die entwickelten Metriken nachvollziehbare Bewertungen liefern. Durch die Benchmark-Scores lassen sich Unterschiede zwischen Risikomatrizen quantifizieren und ihre Eignung besser beurteilen.

Trotz der vielversprechenden Ergebnisse gibt es weiterhin Herausforderungen und Verbesserungspotenziale. So könnte die Methode durch eine breitere empirische Validierung mit realen Risikodaten und praxisrelevanten Risikomatrizen weiter optimiert werden. Insbesondere die Berechnung der Gewichtung der einzelnen Benchmark-Kriterien bedarf einer tieferen Untersuchung, um sicherzustellen, dass die Gesamtbewertung die tatsächliche Aussagekraft einer Risikomatrix bestmöglich widerspiegelt.

Weitere Ausbaumöglichkeiten bestehen darin, auch Risikomatrizen mit nicht-linearen Skalen in den Benchmark einzubeziehen sowie die in Kapitel 3.2.5 vorgestellten, aber noch nicht umgesetzten Bewertungsansätze weiterzuentwickeln. Eine zukünftige Erweiterung des Tools könnte zudem die Integration neuer Berechnungsverfahren oder eine verbesserte Visualisierung der Benchmark-Ergebnisse umfassen.

8 Anlagen

8.1 Quellenverzeichnis

8.1.1 Literaturquellen

- Philip Thomas, Reidar B. Bratvold, J. Eric Bickel (April 2014). The Risk of Using Risk Matrices. SPE Economics & Management, S. 56–66
- Jr., Louis A. Cox (2008). What’s wrong with risk matrices?. Risk Analysis, 28 (2), S. 497–512
- Fekete, Alexander (2024). Risiko, Katastrophen und Resilienz. Springer-Verlag GmbH. <https://doi.org/10.1007/978-3-662-68381-1>
- Michael Krisper (2021). Problems with Risk Matrices Using Ordinal Scales. Institute of Technical Informatics Graz University of Technology
- Beneken, G., Hummel, F., & Kucich, M. (2022). Grundkurs agiles Software-Engineering: Ein Handbuch für Studium und Praxis. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-37371-9>

8.1.2 Bildquellen

1. Steuerkanzlei Steinhäuser. (o. J.). Erfolgsstrategien. Zuletzt abgerufen am 23. Februar 2025, von <https://www.steuerkanzlei-steinhaeuser.de/erfolgsstrategien/>
2. LinkedIn. (o. J.). Risk Matrix Horoscope – Best Practice. Zuletzt abgerufen am 23. Februar 2025, von <https://www.linkedin.com/pulse/risk-matrix-horoscope-best-practice->
3. Tacticx (o. J.). Risikomanagement in der Informationssicherheit. Zuletzt abgerufen am 23. Februar 2025, von <https://www.tacticx.de/informationssicherheit/risikomanagement/>
4. ScienceDirect (2016). Research Article on Risk Matrices. Zuletzt abgerufen am 23. Februar 2025, von <https://www.sciencedirect.com/science/article/abs/pii/S2468896716300374#f0040>
5. Google-Cache (o. J.). Bild einer Risikomatrix. Zuletzt abgerufen am 23. Februar 2025, von https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRIslpF1wNp-vSCXK6SbCVvoMU_I5R2Ay_ygQ&s
6. Romeike, F., & Stallinger, M. (2021). Risiken und Chancen als Wahrscheinlichkeitsverteilungen darstellen. In: Stochastische Szenariosimulation in der Unternehmenspraxis (S. 69). Springer Gabler, Wiesbaden. https://doi.org/10.1007/978-3-658-34063-6_4
7. AgileRisk (o. J.). Risikomanagement für agile Projekte. Zuletzt abgerufen am 23. Februar 2025, von <https://agilerisk.ilmcon.de/>
8. Maschinen-Sicherheit.net (o. J.). Risikomatrix nach Nohl. Zuletzt abgerufen am 23. Februar 2025, von <https://www.maschinen-sicherheit.net/07-seiten/0590-risikomatrix-nohl>
9. TemplateLab (o. J.). Risk Matrix Templates. Zuletzt abgerufen am 23. Februar 2025, von <https://templatelab.com/risk-matrix/>
10. Peter Johann Consulting (o. J.). Risikomatrix – Grundlagen und Anwendung. Zuletzt abgerufen am 23. Februar 2025, von <https://www.peterjohann-consulting.de/risikomatrix/>
11. HP Reliability (o. J.). Criticality Analysis & Risk Matrix (6x6). Zuletzt abgerufen am 23. Februar 2025, von <https://hpreliability.com/what-is-a-criticality-analysis-how-does-it-work/6x6-matrix/>

8.2 Tabelle für statistische Auswertung des Benchmarks mit 17 Risikomatrizen

siehe "Risikomatrix Tests" bzw. PDF "RisikomatrixTests.pdf"

8.3 Benutzerdokumentation für Simulationsprogramm

siehe "Benutzerdokumentation" bzw. PDF "benutzerdokumentation.pdf"

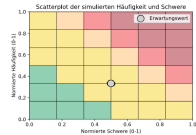
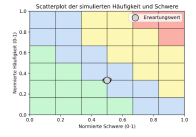
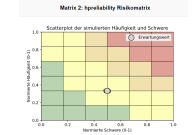
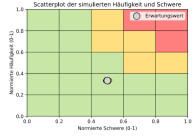
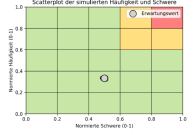
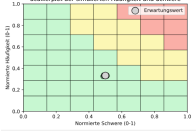
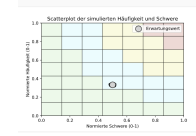
8.4 Programmdokumentation für Simulationsprogramm

siehe "Programmdokumentation" bzw. PDF "programmdokumentation.pdf"

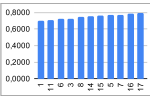
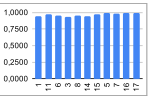
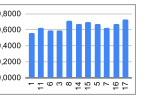
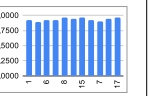
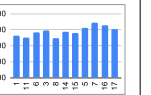
Risikomatrix Tests

1. Untersuchung verschiedener Risikomatrizen nach eigenem Benchmark

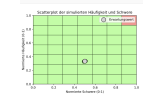
ID	Bild/ Screenshot	Benchmark-Score	Ordnungsmaß	Range-Compression	Overlap	Quantifying Errors	Quelle
10	<p>Matrix 2: Steuerkredit Risikomatrix</p>	0.6364	0.9039	0.5519	0.8889	0.3750	[1]
12	<p>Matrix 2: Textus Risikomatrix</p>	0.6377	0.8746	0.5417	0.8750	0.4167	[3]
13	<p>Scatterplot der simulierten Häufigkeit und Schwere</p>	0.6749	0.8670	0.5580	0.8920	0.5156	[4]
2	<p>Scatterplot der simulierten Häufigkeit und Schwere</p>	0.6836	0.9238	0.5843	0.8916	0.4833	Matrix 2: DIN EN 50126 Matrix
4	<p>Matrix 2: Robin-Dale Risikomatrix</p>	0.6903	0.9103	0.6594	0.9313	0.4167	[5]
9	<p>Matrix 2: Inrcos GmbH</p>	0.6983	0.9403	0.5333	0.9000	0.5625	[7]
1	<p>Matrix 2: Risikomatrix nach Nofl</p>	0.6992	0.9425	0.5563	0.9219	0.5278	[8]
11	<p>Scatterplot der simulierten Häufigkeit und Schwere</p>	0.7069	0.9695	0.6185	0.8888	0.5000	[9]
6	<p>Matrix 2: Rumelko, F., Böttger, M. (2021)</p>	0.7212	0.9558	0.5880	0.9200	0.5625	[6]
3	<p>Matrix 2: Petz-Johann-Consulting Risikomatrix</p>	0.7269	0.9354	0.5880	0.9200	0.5938	[10]

8		0,7451	0,9579	0,7144	0,9652	0,4900	[2]
14		0,7516	0,9441	0,6736	0,9472	0,5700	Matrix unabhängig erstellt
15		0,7619	0,9694	0,6910	0,9639	0,5600	[11]
5		0,7697	0,9879	0,6679	0,9199	0,6250	r., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Table IV in Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497–512
7		0,7714	0,9821	0,6253	0,9000	0,6875	r., Louis A. Cox (2008). What's wrong with risk matrices? [Siehe Table IV in Abschnitt: "Discussion of Weak Consistency" auf Seite 502]. Risk Analysis, 28 (2), 497–512
16		0,7849	0,9947	0,6727	0,9387	0,6527	Matrix anhand Erkenntnisse der statistischen Auswertung unabhängig erstellt
17		0,7907	0,9917	0,7276	0,9653	0,6042	Matrix anhand Erkenntnisse der statistischen Auswertung unabhängig erstellt

2. Statistiken

	Benchmark-Score	Ordnungsmaß	Range-Compression	Overlap	Quantifying Errors
Verteilung					
Wertintervalle	[0,6364 0,7907]	[0,8670 0,9947]	[0,5333 0,7276]	[0,8750 0,9653]	[0,3750 0,6875]

3. Grenztestung des Benchmark Modells

Untersuchtes Merkmal	Matrix	Benchmark Ergebnis
Range Compression (schlecht)	<p>Matrix 2: Range Compression schlecht</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.7095108513104049</p> <p><u>Ordnungsmaß:</u> 0.9359648243985869</p> <p><u>Range-Compression:</u> 0.3480000000000001</p> <p><u>Overlap:</u> 0.96</p> <p><u>Quantifying Errors:</u> 0.734375</p>
Range Compression (gut)	<p>Matrix 1: Range Compression gut</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.6074479778068977</p> <p><u>Ordnungsmaß:</u> 0.8924072283961024</p> <p><u>Range-Compression:</u> 0.788375</p> <p><u>Overlap:</u> 0.9860284117647059</p> <p><u>Quantifying Errors:</u> 0.0</p>
Ordnungsmaß (gut)	<p>Matrix 2: Ordnungsmaß gut</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.7304579589167883</p> <p><u>Ordnungsmaß:</u> 0.9876772787520383</p> <p><u>Range-Compression:</u> 0.4295916367346938</p> <p><u>Overlap:</u> 0.8775510204081632</p> <p><u>Quantifying Errors:</u> 0.7430555555555556</p>
Ordnungsmaß (schlecht)	<p>Matrix-Matrix</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.4405580910769231</p> <p><u>Ordnungsmaß:</u> 0</p> <p><u>Range-Compression:</u> 0.83168</p> <p><u>Overlap:</u> 0.9605692307692308</p> <p><u>Quantifying Errors:</u> 0.0</p>
Overlap (gut)	<p>Matrix 1: Range Compression gut</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.6074479778068977</p> <p><u>Ordnungsmaß:</u> 0.8924072283961024</p> <p><u>Range-Compression:</u> 0.788375</p> <p><u>Overlap:</u> 0.9860284117647059</p> <p><u>Quantifying Errors:</u> 0.0</p>
Quantifying Errors (gut)	<p>Matrix 2: Quantifying Errors gut</p> <p>Skalierung der einzelnen Häufigkeit und Schwere</p> 	<p>Benchmark-Score: 0.39443</p> <p><u>Ordnungsmaß:</u> 0</p> <p><u>Range-Compression:</u> 0.30000000000000004</p> <p><u>Overlap:</u> 0</p> <p><u>Quantifying Errors:</u> 1</p>

Benutzerdokumentation

Ziel des Programms

Das Programm dient dazu, Risikomatrizen anhand wissenschaftlicher Kriterien zu vergleichen und zu bewerten, welche Matrix besser ist als eine andere. Drei Risikomatrizen sind bereits vorgegeben: zwei aus der Fachliteratur und eine, die eine DIN-Norm darstellt. Darüber hinaus kann der Nutzer eigene Matrizen erstellen und diese entweder mit den vorgegebenen oder untereinander vergleichen.

Systemanforderungen und Installation

- **Voraussetzungen:**
 - Python Version 3.12 oder höher

- **Installation:**

	Linux (empfohlen)	Windows
Programm herunterladen	- In der Konsole ausführen: <code>git clone https://github.com/nilsgnh/Projektseminar-Risikomatrix</code> - Alternativ: ZIP-Datei entpacken	- ZIP-Datei entpacken
Erstmaliger Start	- Wechsel in Projektordner - In der Konsole ausführen: <code>./risikomatrix.sh</code>	- Wechsel in Projektordner - Doppelklick auf <code>start.bat</code>
Bei installierten Abhängigkeiten	- Wechsel in Ordner: <code>src/</code> - in der Konsole ausführen: <code>python app.py</code>	Wechsel in Ordner: <code>src/</code> <code>python app.py</code>
Programm öffnen	URL aufrufen: http://127.0.0.1:5000	URL aufrufen: http://127.0.0.1:5000

Funktionalität

Benchmark durchführen

1. Zwei Matrizen unter "Set Matrices" auswählen und die Auswahl mit dem Button bestätigen.
2. Simulationsparameter nach Belieben anpassen, um einen Risikofall zu simulieren.
3. Durch Klicken auf "**Submit**" wird der Benchmark und die Simulation durchgeführt.
4. Die jeweils bessere Bewertung im Benchmark wird grün markiert.
5. Die Plots zeigen die Simulation des Risikofalls visuell an.

Eigene Matrix erstellen

1. Auf **"Go To Custom-Matrix Page"** klicken.
2. Einen Namen für die Matrix eingeben (sollte wiedererkennbar sein).
3. **Risikoklassen definieren**
 - So viele Zeilen hinzufügen, wie es Risikostufen in der Matrix gibt.
 - Je kleiner die *Risk Number*, desto geringer das Risiko der Risikoklasse.
 - Eine Farbe für jede Risikoklasse auswählen, indem auf das Farbfeld unter "Risk Color" geklickt wird.
 - Einen Namen für jede Risikoklasse vergeben.
4. **Risikomatrix definieren**
 - Anzahl der Zeilen und Spalten der Risikomatrix eingeben.
 - Auf "Generate Table" klicken, um die Tabellendarstellung der Matrix zu erzeugen.
 - In jedes Feld der Tabelle die entsprechende *Risk Number* aus der zuvor definierten Risikoklassentabelle eintragen.
5. **Risikomatrix speichern**
 - Auf **"Submit Table Data"** klicken.
 - Bei Erfolg erscheint eine Meldung: *"Risk Matrix successfully saved"*.
 - Der Name der Matrix erscheint unter "Matrix Overview" und kann nun verwendet werden.
6. Auf **"Go Back"** am Seitenanfang klicken, um eine Benchmark-Simulation durchzuführen.

The screenshot shows the 'Enter Matrix Name' section with a text input field containing 'BeispielMatrix'. Below it is the 'Set Risk Levels' section, which contains a table with three rows. Each row has a 'Risk Number' (1, 2, 3), a 'Risk Color' (green, yellow, red), and a 'Risk Name' (leicht, mittel, schwer). There are 'Add Row' and 'Remove Row' buttons below the table. At the bottom is the 'Define Matrix' section, which asks the user to define the number of rows and columns (max 7 each) and provides a 'Generate Table' button. Below this, a 3x3 grid of input fields is shown, each containing a risk number (1, 2, or 3).

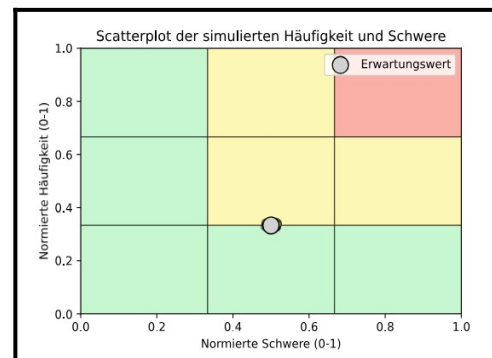
Risk Number	Risk Color	Risk Name
1	Green	leicht
2	Yellow	mittel
3	Red	schwer

Buttons: Add Row, Remove Row

Define Matrix: Rows (max 7): 3, Columns (max 7): 3, Generate Table

Generated Table (3x3):

1	2	3
1	2	2
1	1	1



Hinweise

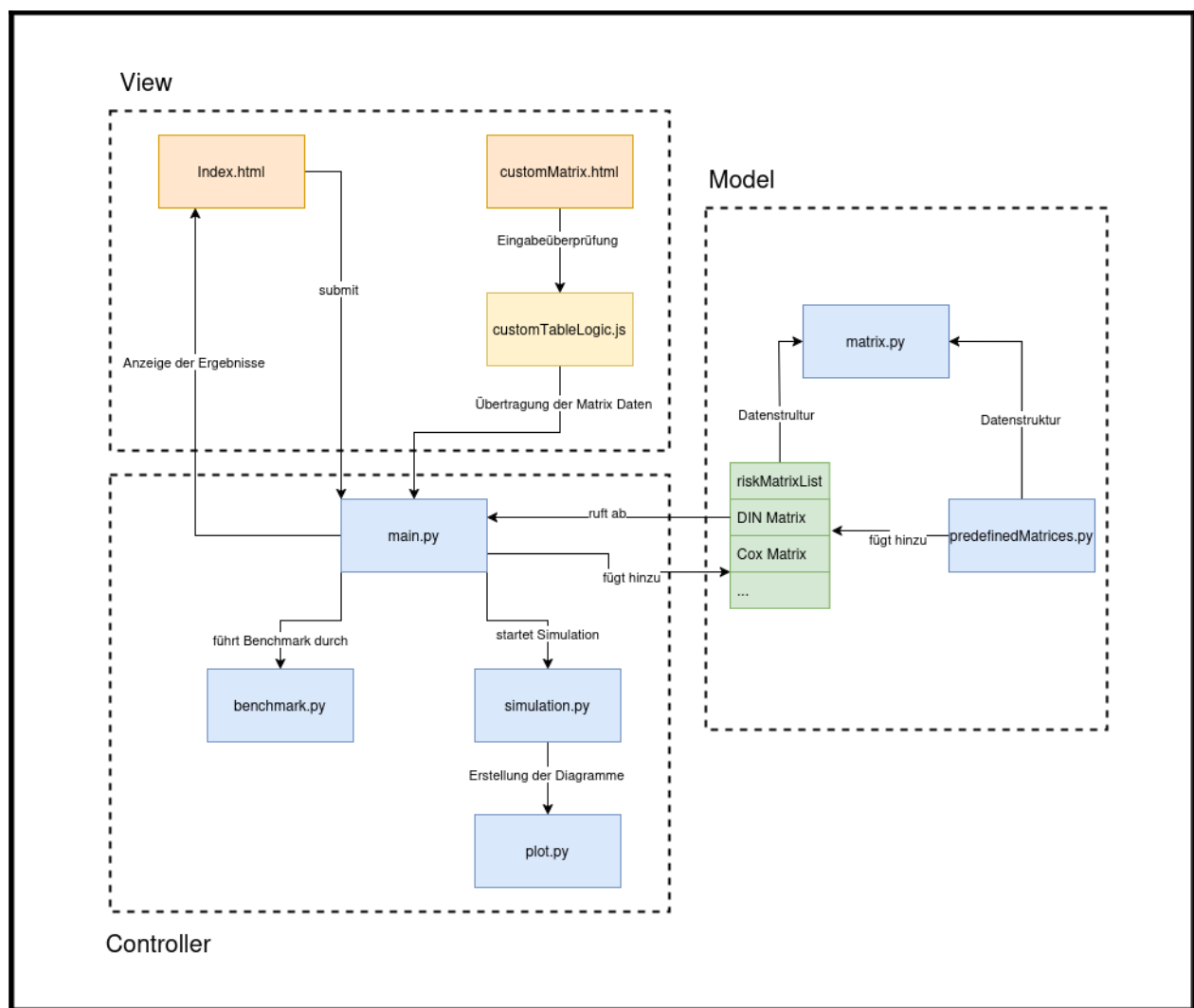
- Selbst erstellte Matrizen werden nur im Hauptspeicher gespeichert. Sobald das Programm beendet wird, gehen sie verloren.

Programmdokumentation

Dependencies und Versionen

Name	Version
Python	3.12
Flask	3.0.2
Matplotlib	3.6.3
numpy	1.26.4
SciPy	1.11.4
Pandas	2.1.4
Seaborn	0.13.2

Programmübersicht



Quelldateien

HTML, CSS, JS

Dateiname	Erklärung
index.html	HTML-Konstrukt für die Hauptseite
customMatrix.html	HTML-Konstrukt für die Matrix Erstellung Seite
Index.css	Styling
customTableLogic.js	Logik zur Eingabe der Risikomatrix

Python Scripte

Dateiname	Erklärung
app.py	startet die Flask Anwendung
matrix.py	Klassendefinition für eine Risikomatrix
main.py	steuert die Kernlogik der Anwendung, indem sie Risikomatrizen verwaltet, Simulationen durchführt und Ergebnisse auf der Webseite darstellt
Benchmark.py	Berechnet den Benchmark für eine gegebene Risikomatrix
Plot.py	erzeugt die Diagramme für eine übergebene Matrix und gegebene Simulationspunkte
predefinedMatrices	Definiert 3 vorgegebene Risikomatrizen
Simulation	simuliert anhand des gegebenen Erwartungswertes und Varianz Punkte auf der Matrix stellt Funktionen bereit, die eine prozentuale Unsicherheit in Standardabweichung und Varianz umrechnen

Matrix Datenstruktur

Attribut	Bedeutung	Beispiel
STRING: name	Name der Matrix zur Identifikation	<code>name = "DIN EN 50126 Matrix"</code>
2D-ARRAY<INT> : representation	Ein Array, in dem jedes Element ein Feld darstellt. Die Risikoklasse eines Feldes wird numerisch definiert, wobei 1 die niedrigste Klasse ist	<code>matrix_rep = np.array([[3, 4, 4, 4], [2, 3, 4, 4], [2, 3, 3, 4], [1, 2, 3, 3], [1, 1, 2, 2], [1, 1, 1, 1],])</code>
INT : Rows, Cols	Anzahl der Zeilen und Spalten der Matrix, automatisch aus der Repräsentation berechnet	-
2D-ARRAY<INT> : representation	Fortlaufende Nummerierung der einzelnen Felder der Matrix	<code>field_nums = np.zeros((len(matrix_rep), len(matrix_rep[0])), dtype=int)</code> <code>counter = 1</code> <code>for i in range (len(matrix_rep)):</code> <code>for j in range (len(matrix_rep[0])):</code> <code>field_nums[i][j] = counter</code> <code>counter += 1</code>
DICT<INT, STRING> RiskLabels	Zuordnung jeder Zahl aus der Repräsentation zu einer Risikoklassenbezeichnung	<code>risk_labels = {1: "Vernachlässigbar", 2: "Tolerabel", 3: "Unerwünscht", 4: "Intolerabel"}</code>
ARRAY<STRING> riskColors	Hexadezimale Farbwerte für jede Risikoklasse als Liste	<code>risk_colors = ["#92D050", "#8EB4E3", "#FFC000", "#FF0000"]</code>
ARRAY<STRING> xLabels, yLabels	Liste der Beschriftungen für die X- und Y-Achse	<code>y_beschriftungen = ["Häufig", "Wahrscheinlich", "Gelegentlich", "Selten", "Unwahrscheinlich", "Unvorstellbar"]</code> <code>x_beschriftungen = ["Unbedeutend", "Marginal", "Kritisch", "Katastrophal"]</code>