

TEILPRÜFUNG 2

Zufall auf dem Computer

IMPLEMENTIERUNG, SIMULATION UND ANALYSE EINES GLÜCKSSPIELS

Implementierung und Simulation des Glücksspiels
„BINGO“ mittels JAVA am Computer und Analyse
der statistischen Zusammenhänge des Glücksspiels

Alexander Wiltz

Matrikelnummer: 882591

Hochschule Kaiserslautern

Studiengang: IT-Analyst

Teilprüfung 2 – Zufall auf dem Computer

Inhalt

1	Einführung	2
1.1	Motivation	2
1.2	Ziel	2
1.3	Geschichte und Ursprung.....	2
1.4	Varianten	2
2	Hauptteil	4
2.1	Betrachtung der Parameter.....	4
2.2	Besondere Anforderungen an den Code.....	4
2.3	Mersenne Twister vs. Random Generator	4
2.4	Berechnung der theoretischen Gewinnwahrscheinlichkeit	5
2.5	Durchführung mit 20 Ziehungen pro Simulation	5
2.6	Durchführung mit 15 Ziehungen pro Simulation	11
2.7	Durchführung mit 10 Ziehungen pro Simulation	17
2.8	Durchführung mit 5 Ziehungen pro Simulation	23
2.9	Vergleich der Gewinnchancen abhängig der Ziehungen.....	30
3	Resümee	31
3.1	Software-Erweiterung	31
3.2	Weitere Analysen	31
4	Quellen	32
5	Dokumentation.....	33
5.1	Bingo.java	33
5.2	CSimulation.java	34

Teilprüfung 2 – Zufall auf dem Computer

1 Einführung

1.1 Motivation

Glücksspiele üben seit jeher einen besonderen Reiz auf Menschen aus. Sie suchen von Natur aus nach Abwechslung und neuen Erlebnissen, um Spannung zu erleben. Besonders die einfachen Menschen finden in solchen Spielen eine Abwechslung zum Alltag.

Das leicht verständliche und wenig aufwendige Spiel Bingo ist auf vielen Dorffesten zu finden und mit verhältnismäßig geringen Einsätzen spielbar.

Daher bietet sich eine einfache Implementierung mittels JAVA zur Computersimulation an, um statistische Zusammenhänge zu analysieren und anschließend auszuwerten.

1.2 Ziel

Die Simulation des Spiels soll die erforderliche hohe Anzahl an Spieldurchläufen vereinfachen und beschleunigen, zur empirischen Ermittlung der Wahrscheinlichkeiten. Die Untersuchung soll aufzeigen, wie hoch die Gewinnchancen beim Bingo sind.

1.3 Geschichte und Ursprung

Der Spielname Bingo wurde 1929 von dem Amerikaner Edwin Lowe eingeführt. Er hatte auf einem Jahrmarkt in Georgia eine Gruppe von spanischen Spielern beobachtet, die die gezogenen Zahlen mit getrockneten Bohnen abdeckten und bei Gewinn „Beano“ riefen. Als Edwin Lowe seine ersten Beano-Runden in New York veranstaltete, rief eine Gewinnerin im Eifer des Gefechtes „Bingo“. Diesen Namen fand Lowe besser, und so blieb es dabei. Für einen Dollar pro Jahr durften andere das Spiel selbst nutzen, allerdings unter der Auflage, es Bingo zu nennen.

[Quelle: <https://de.wikipedia.org/wiki/Bingo#ursprung>, 07.06.2021]

1.4 Varianten

1.4.1 Spielprinzip

Die Bingo-Matrix besteht aus 5x5 Feldern aus dem Zahlenkreis 1 bis 75. Insgesamt werden 20 zufällige Zahlen aufgerufen. Finden sich fünf Treffer in waagerechter, senkrechter oder diagonaler Reihe heißt das Bingo. Das Feld in der Mitte kann als „Joker“ zählen.

Ein Conférencier zieht wie beim Lotto per Zufall aus einer Trommel Kugeln mit aufgedruckten Zahlen und ruft diese laut aus. Die Spieler markieren gleichzeitig diese Zahlen mit einem Farbstempel oder einem Stein, sofern diese auf ihrer Karte vorhanden sind.

Sobald der erste Spieler auf seinem Los alle 5 Zahlen in waagerechter, senkrechter oder diagonaler Reihenfolge hat, ruft dieser laut „Bingo“. Dann ist die Spielrunde abgeschlossen.

Weitere Anforderungen an das Generieren der Karte ist, dass in der ersten Spalte Wert zwischen 1 und 15 stehen, in der zweiten 16 bis 30, in der dritten 31 bis 45, in der vierten Spalte 46 bis 60 und in der letzten 61 bis 75.

1.4.2 Einfach-Bingo

5 Zahlen stimmen in einer Reihe waagerecht, senkrecht oder diagonal mit den gezogenen Zahlen überein.

Teilprüfung 2 – Zufall auf dem Computer

1.4.3 Doppel-Bingo

2-mal 5 Bingo-Zahlen einer Matrix stimmen in waagerechter, senkrechter oder diagonaler Reihe mit den gezogenen Zahlen überein. Dies können 9 oder 10 Gewinnzahlen sein.

1.4.4 Dreifach-Bingo

In der Gewinnklasse 1 müssen mindestens 3-mal Bingo in einer Matrix (waagerecht, senkrecht oder diagonal) zu finden sein. Bleibt die Gewinnklasse unbesetzt bildet sich ein Jackpot.

1.4.5 Spielvariante

Die in dieser Arbeit dargestellte Bingo-Matrix besteht aus 5x5 Feldern, hat in der Mitte kein „Joker“-Feld und es werden per Zufall 25 Zahlen gezogen.

Anschließend werden weitere 20 Zahlen gezogen, die den Conférencier simulieren.

Diese Zahlen werden verglichen und auf ein mögliches Match geprüft.

1.4.6 Code und Simulationsergebnisse

Die JAVA-Dateien zu dieser Arbeit, sowie die Rohdaten und aufgearbeiteten Sheets stehen auf folgendem GitHub-Account bereit:

<https://github.com/alexander-wiltz/zufall-am-computer/>

2 Hauptteil

2.1 Betrachtung der Parameter

Vor der eigentlichen Entwicklung des Codes für das Glücksspiel „Bingo“, werden Überlegungen zur anschließenden Auswertung und der daraus resultierenden Parameter angestellt.

Da die Basisparameter zur Kartengröße, sowie der zu ziehenden Gewinnzahlen feststehen, sollen im Nachgang die Anzahl der Matches, also wie oft waren 1 bis 5 Treffer auf der Karte, gezählt und ausgewertet werden.

Zusätzlich die Bestimmung, an welcher Stelle der Treffer eines „Bingo“ war, ob in einer Reihe, einer Spalte oder Diagonal und das Detail, in jeweils welcher und zuletzt, ob es ein Gewinn war.

Das Spiel als solches und die Simulation zur Analyse sind getrennt und sollen jeweils anders dargestellt werden, um eine klare Trennung zur Auswertung zu haben.

2.2 Besondere Anforderungen an den Code

Bei dem Generieren der Zahlen für die Bingo-Karte wird jede Zahl innerhalb eines festgelegten Bereichs erzeugt. Dabei ist es der Zahlenmenge gleichgültig, ob eine bestimmte Zahl bereits existiert. Da allerdings auf einer Bingo-Karte keine doppelten Zahlen vorkommen, muss dies entsprechend realisiert werden.

Der Einsatz eines HashSets ist dabei hilfreich. Aber die Ausgabe ist unsortiert und in den meisten Fällen ungleich der hinzugefügten Reihenfolge. Demnach muss hier ein LinkedHashSet verwendet werden. Ansonsten ist die ausgegebene Karte falsch sortiert.

2.3 Mersenne Twister vs. Random Generator

In der Realität folgt der Erwerb der Karte dem Zufall, sowie das Ziehen der Gewinnzahlen durch den Conférencier.

Um eine Spielkarte mit zufälligen Zahlen zu füllen, wird ein Zufallszahlen-Generator einer JAVA-Bibliothek genutzt. Die Grundproblematik ist, dass errechnete Zahlen niemals zufällig sein können. Demnach sind die generierten beziehungsweise berechneten Zahlen Pseudozufallszahlen.

Die Standardklasse `java.util.random` benutzt eine Tiefe von 32bit, was bedeutet, dass sich die Zahlen nach $2^{32}-1$ Zahlen wiederholen. Siehe Dokumentation [WWW 07]

Der Mersenne Twister, aus der `org.apache.commons.math3.random` Bibliothek hingegen bietet eine Periodenlänge von $2^{19937}-1$ [WWW 08]. Die generierte Zahl ist demnach „zufälliger“ als die einer generierten Zahl aus der Standardklasse.

Der Nachteil des Twister ist die Geschwindigkeit. Der Standard-random-Generator ist deutlich schneller.

Teilprüfung 2 – Zufall auf dem Computer

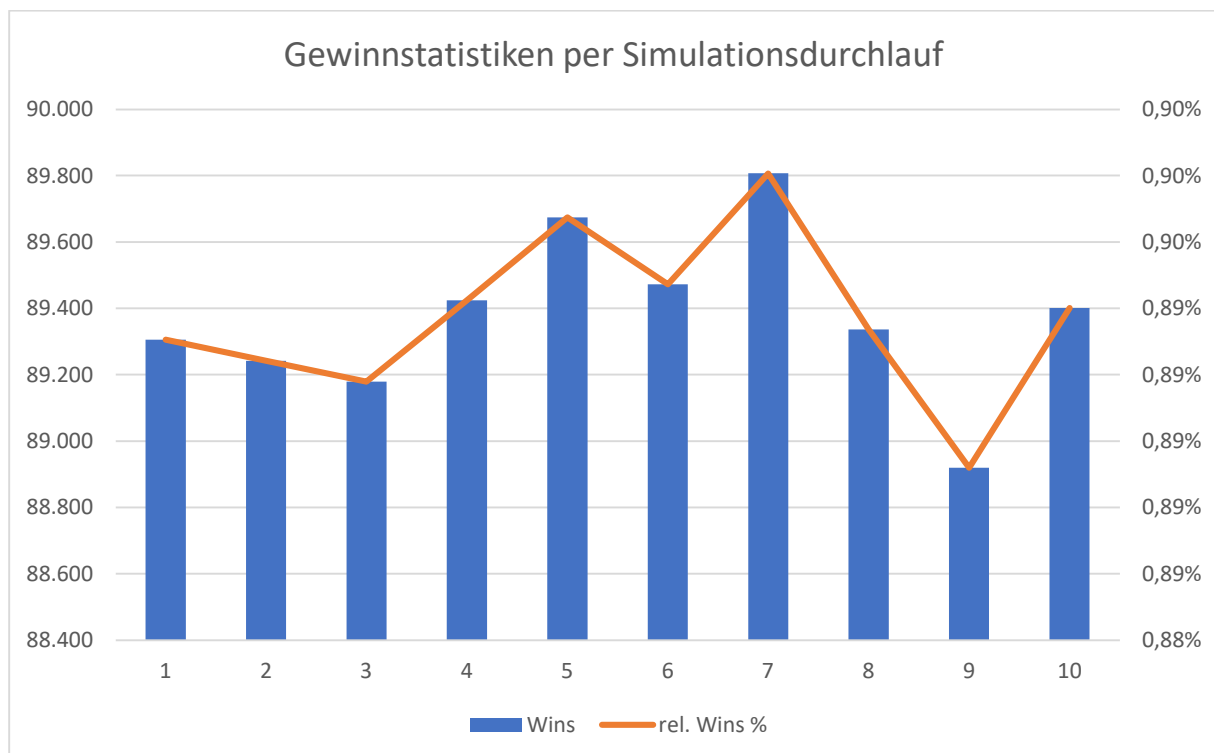
2.4 Berechnung der theoretischen Gewinnwahrscheinlichkeit

Die Berechnung einer theoretischen Wahrscheinlichkeit im Vorfeld der Analysen könnte hilfreich sein, um das empirisch ermittelte Ergebnis zu verifizieren. Allerdings würde die Ausarbeitung im Falle von Bingo den Rahmen dieser Ausarbeitung sprengen. Aus diesem Grund wurde hier auf die weitere theoretische Ausführung verzichtet.

2.5 Durchführung mit 20 Ziehungen pro Simulation

Die ersten zehn Versuche wurden mit jeweils 10.000.000 Spielen bei jeweils 20 Ziehungen der Gewinnzahlen. Es ist zu erwarten, dass die Gewinnwahrscheinlichkeit am höchsten ist.

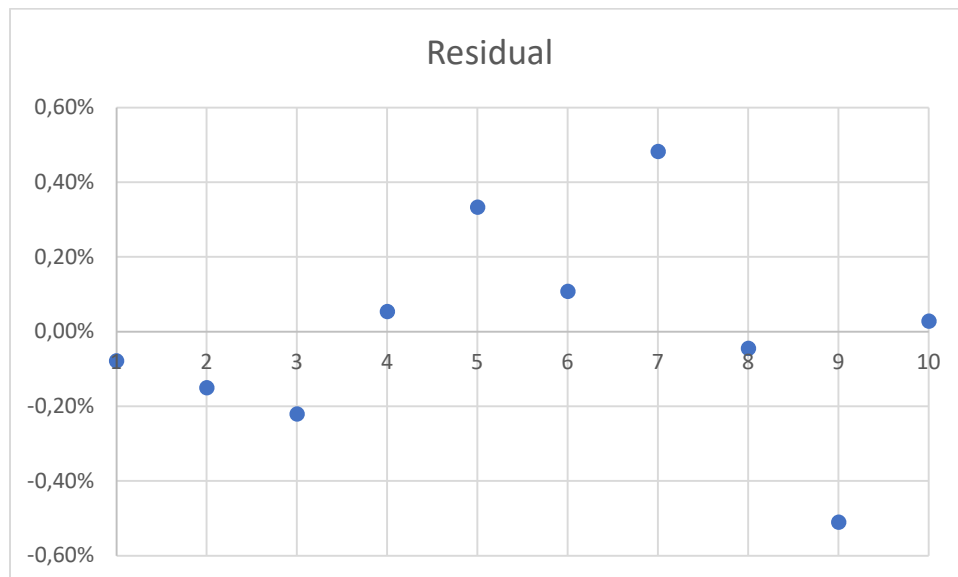
2.5.1 Gewinne per Durchlauf



	Σ	Min	Max	Avg
Wins	893.761	88.920	89.807	89.376
Simulations	100.000.000			
rel. Wins %	0,8938%			
Chance	1:111,89			

Die absolute Anzahl an Gewinnen per 10 Millionen Spiele ist wie zu erwarten annähernd gleich. Die Differenz zwischen dem minimalen und maximalen Wert ist verhältnismäßig gering.

Interessant ist, dass nur etwa 0,89% aller Spiele zu einem Gewinn führen, trotz der hohen Anzahl an Ziehungen von Kugeln. Eine Chance von 1:112 ein Bingo zu treffen besteht.

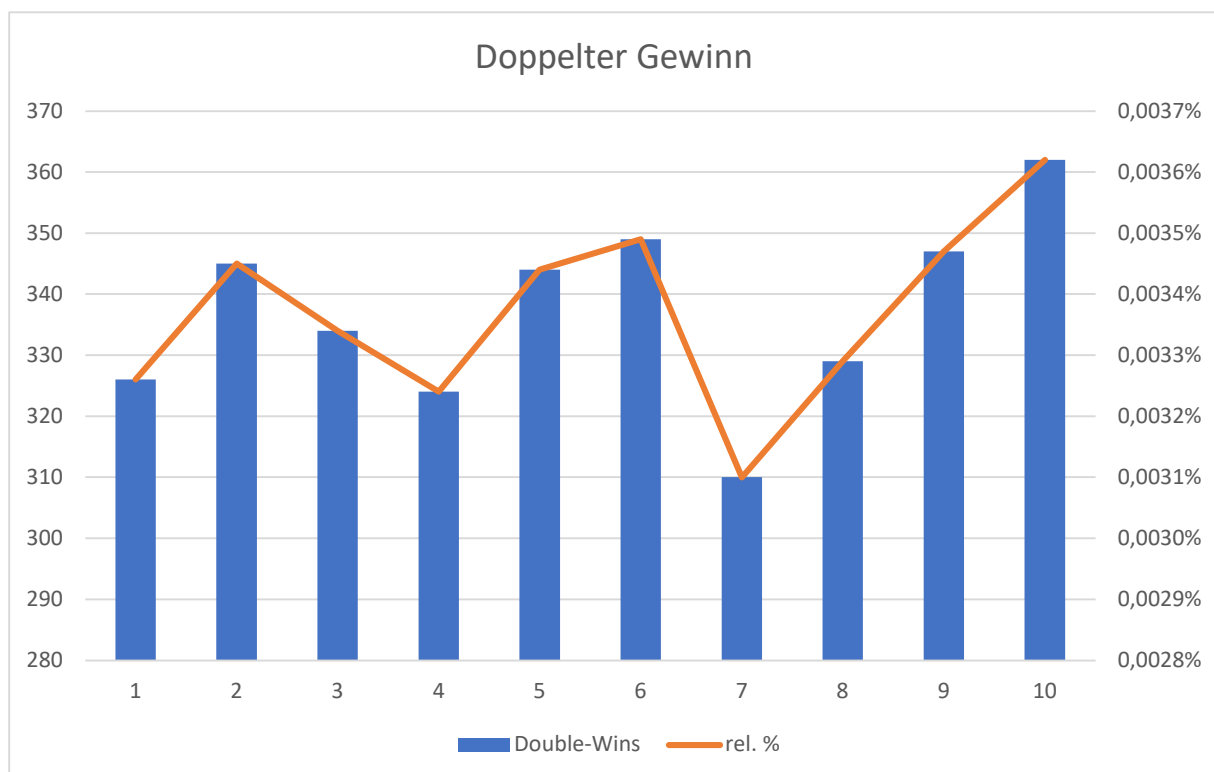


Die prozentuale Abweichung zum Mittelwert ist nahe beieinander. Die Gewinnwerte haben aufgrund ihrer Größe nur wenig Abweichung.

2.5.2 Doppelter Gewinn

Wie wahrscheinlich ist es, dass entweder mehrere Reihen gleichzeitig oder mehrere Spalten oder eine Kombination aus beidem gewinnt?

Bei 20 Ziehungen gab es während der Simulationsdurchläufe insgesamt 3370 Doppelgewinne. Durchschnittlich waren in jedem Spiel 337 Stück.



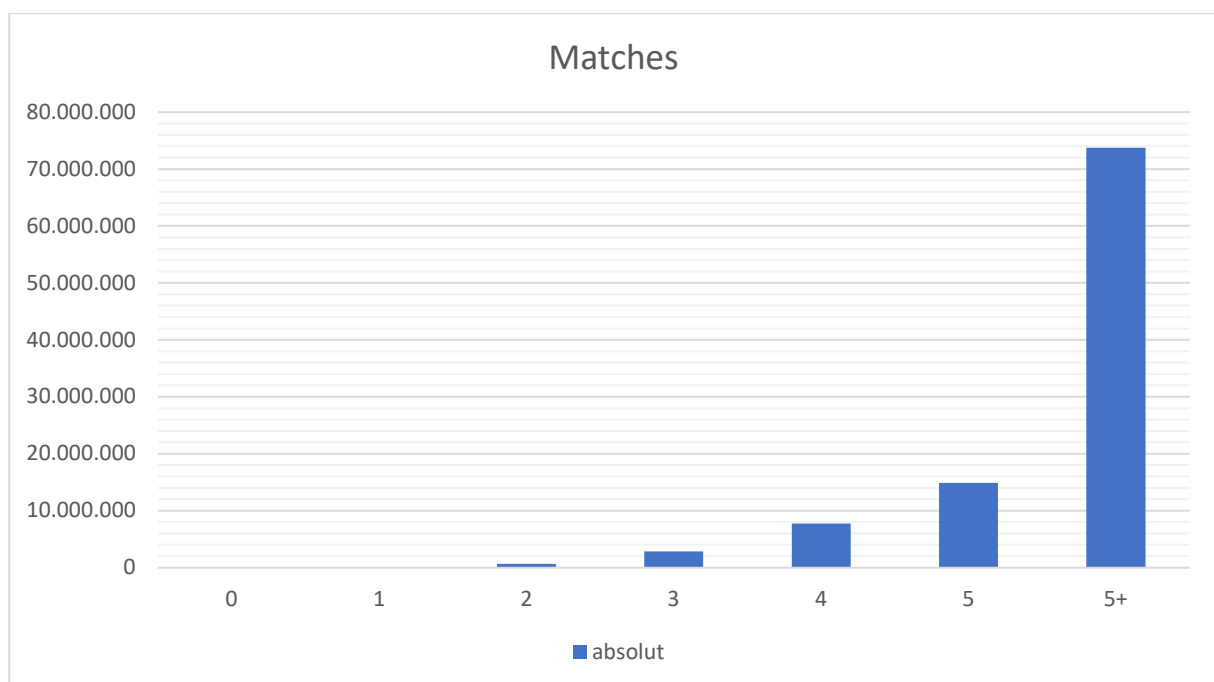
Teilprüfung 2 – Zufall auf dem Computer

2.5.3 Trefferverteilung per Kugeln

Um weitere Rückschlüsse auf die Unterscheidung der Anzahl von Ziehungen vornehmen zu können, wurden die Anzahl der Treffer getrackt.

Matches	absolut			relativ		
	Σ	Min	Max	Σ	Min	Max
0	5.812	548	604	0,01%	0,01%	0,01%
1	94.368	9.299	9.534	0,09%	0,09%	0,10%
2	674.814	67.015	67.812	0,67%	0,67%	0,68%
3	2.821.632	280.948	282.970	2,82%	2,81%	2,83%
4	7.755.226	774.816	776.075	7,76%	7,75%	7,76%
5	14.891.060	1.488.239	1.490.322	14,89%	14,88%	14,90%
5+	73.757.088	7.375.003	7.376.823	73,76%	73,75%	73,77%
Σ	100.000.000					

Aufgrund der 20 Ziehungen pro Spiel, liegt die Wahrscheinlichkeit fünf oder mehr Treffer auf dem Bingo-Feld zu haben bei etwa 74%. Eine erste Prognose deutet darauf hin, dass man zur Steigerung der Schwierigkeit entweder die Anzahl der Ziehungen reduzieren müsste oder aber die Anzahl der Fünferreihen erhöhen müsste, die für einen Gewinn sorgen.

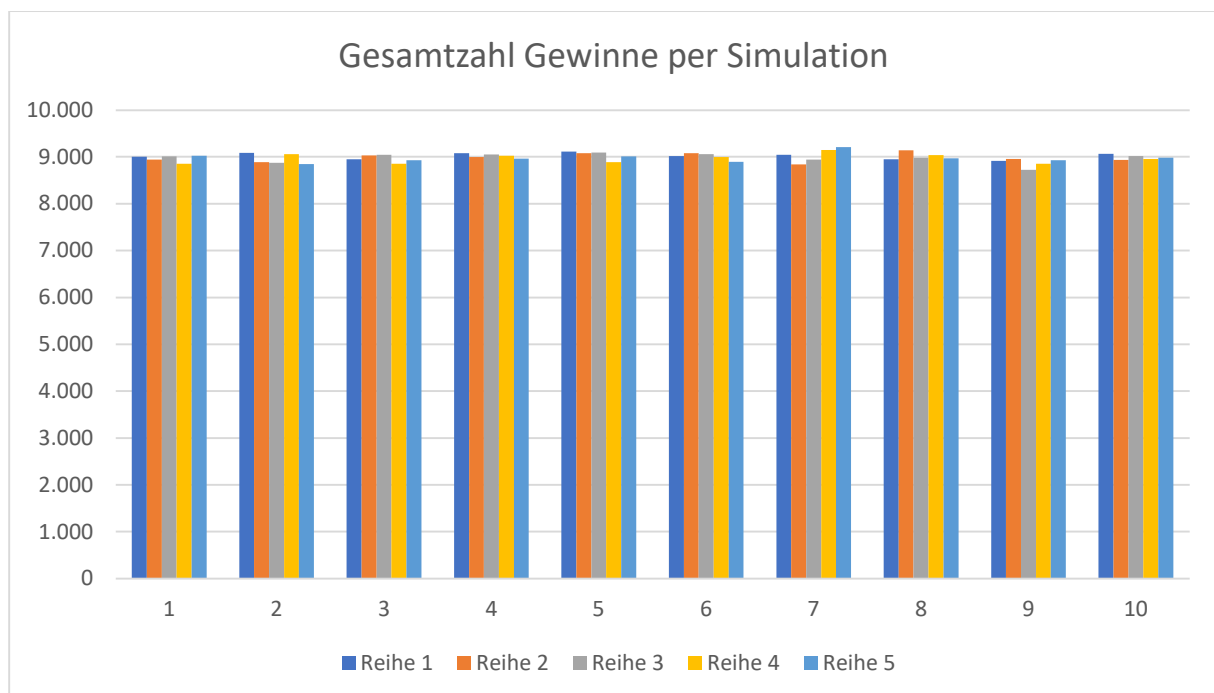


Teilprüfung 2 – Zufall auf dem Computer

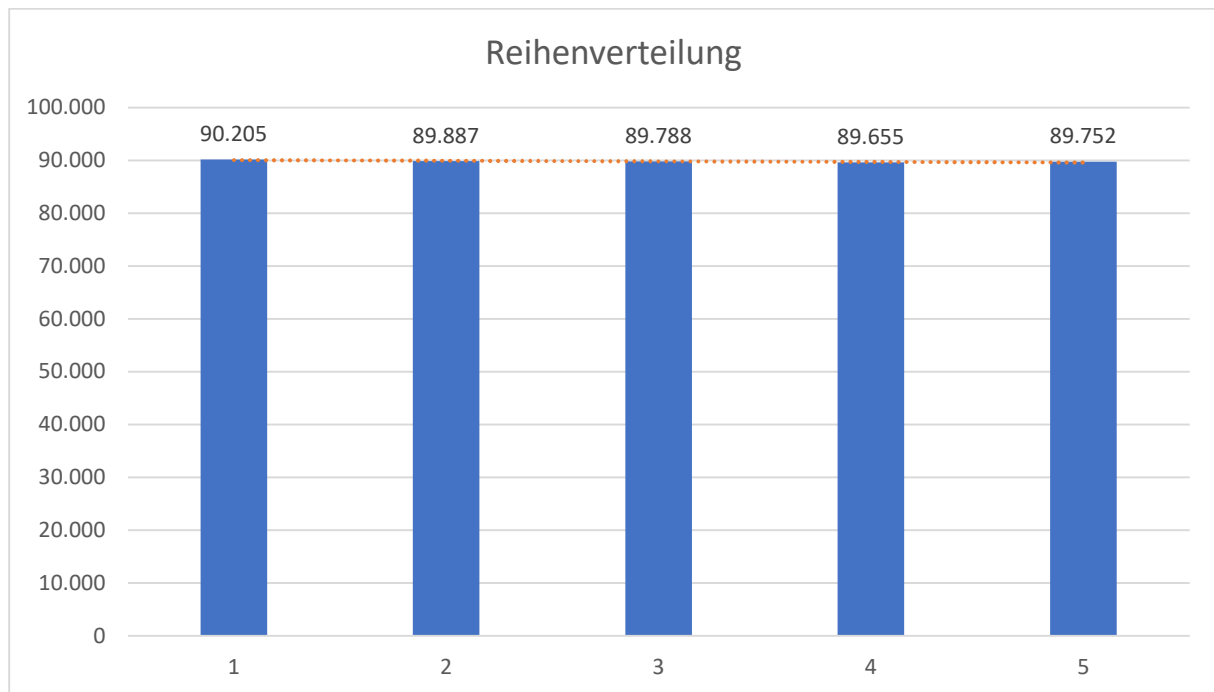
2.5.4 Verteilung der Gewinntreffer per Reihe

Um Abschätzen zu können, ob die Verteilung der Gewinne auf die Reihen gleichermaßen ist, oder ob eine bestimmte Reihe heraussticht, bietet sich an während eines Durchlaufs im Falle eines Gewinnes die entsprechend passende Reihe zu speichern.

	Σ	Avg	Row Win	Row p Win	Row p Game
Row					
1	90.205	9.021	20,08%	10,09%	0,09%
2	89.887	8.989	20,01%	10,06%	0,09%
3	89.788	8.979	19,98%	10,05%	0,09%
4	89.655	8.966	19,95%	10,03%	0,09%
5	89.752	8.975	19,98%	10,04%	0,09%
abs. RowWins	449.287	44.929	100,00%	50,27%	0,45%
Wins	893.761	89.376			
Simulations	100.000.000				



Teilprüfung 2 – Zufall auf dem Computer



Die absolute Verteilung auf die fünf Reihen ist nahe beieinander. Somit ist gezeigt, dass jede Reihe annähernd gleich fair ist.

Die relativen Ergebnisse der Verteilung zeigen, dass diese etwa 20% beträgt und pro Spiel 0,45% einen Reihengewinn treffen.

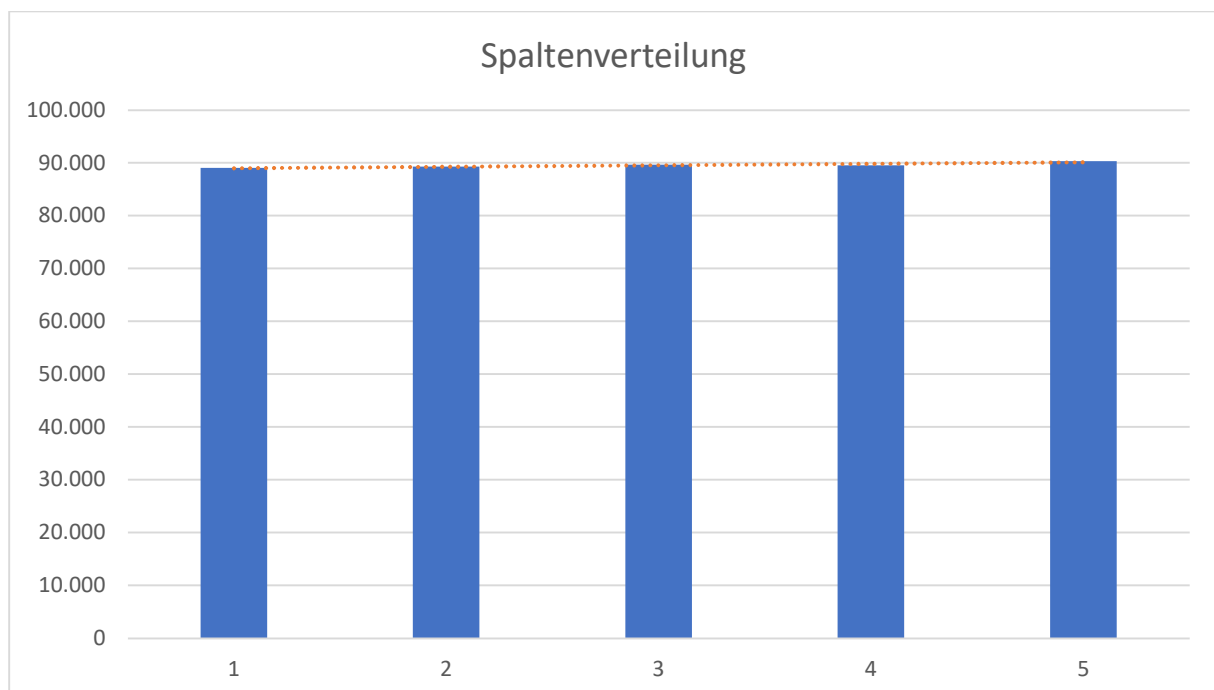
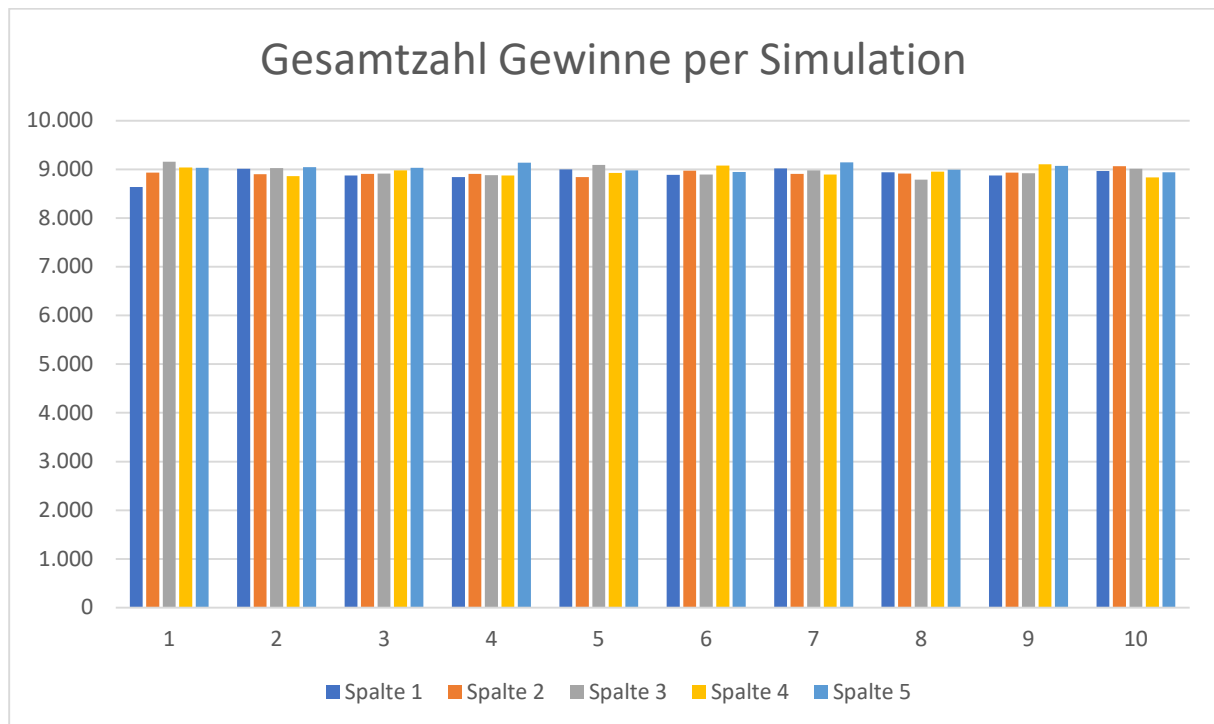
Aufgrund der Tatsache, dass circa 10% der Gewinne auf einer Reihe liegen, folgt eine Untersuchung der Spaltentreffer.

2.5.5 Verteilung der Gewinntreffer per Spalte

	Σ	Avg	Col Win	Col p Win	Col p Game
Col					
1	89.047	8.905	19,88%	9,96%	0,09%
2	89.279	8.928	19,94%	9,99%	0,09%
3	89.653	8.965	20,02%	10,03%	0,09%
4	89.542	8.954	19,99%	10,02%	0,09%
5	90.323	9.032	20,17%	10,11%	0,09%
abs. ColWins	447.844	44.784	100,00%	50,11%	0,45%
Wins	893.761	89.376			
Simulations	100.000.000				

Die Zahlen scheinen sich denen der Reihenauswertung anzugleichen. Folgende Grafiken untermauern die Vermutung:

Teilprüfung 2 – Zufall auf dem Computer



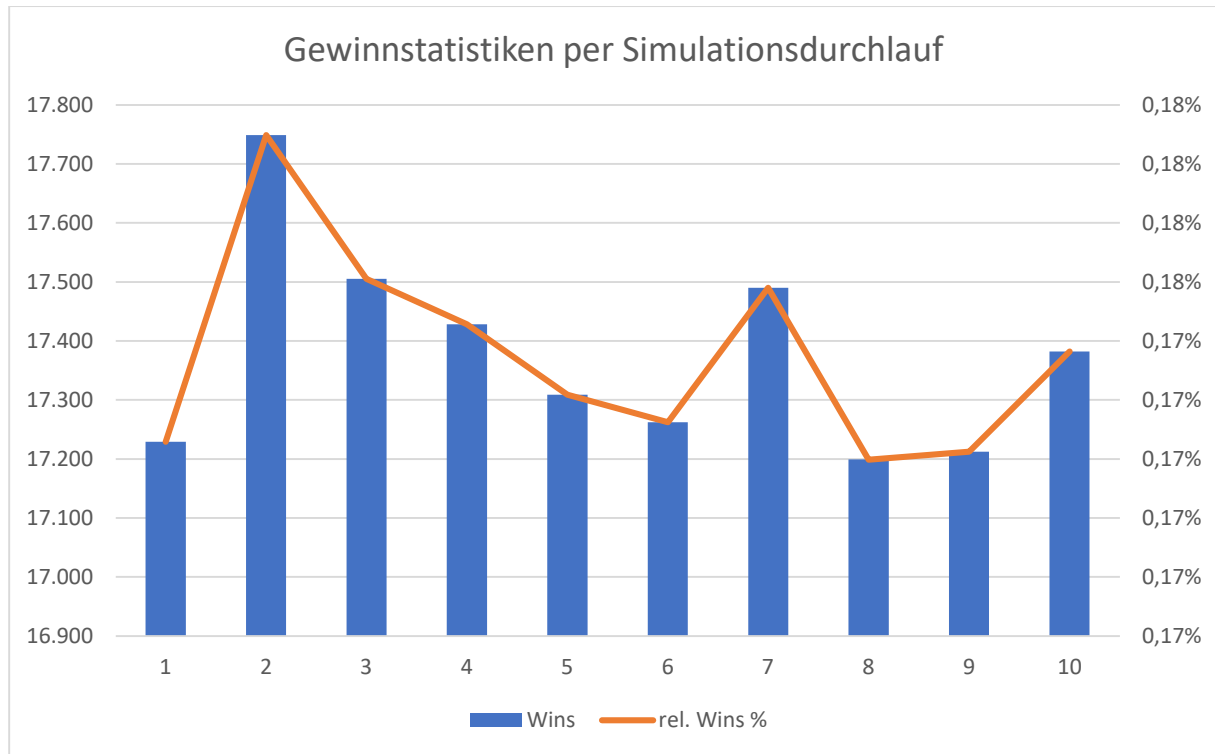
Auch hier ist die Verteilung auf die Spalte jeweils etwa 20%. 0,45% aller Spiele könnten eine Chance auf einen Treffer auf einer Spalte geben.

Teilprüfung 2 – Zufall auf dem Computer

2.6 Durchführung mit 15 Ziehungen pro Simulation

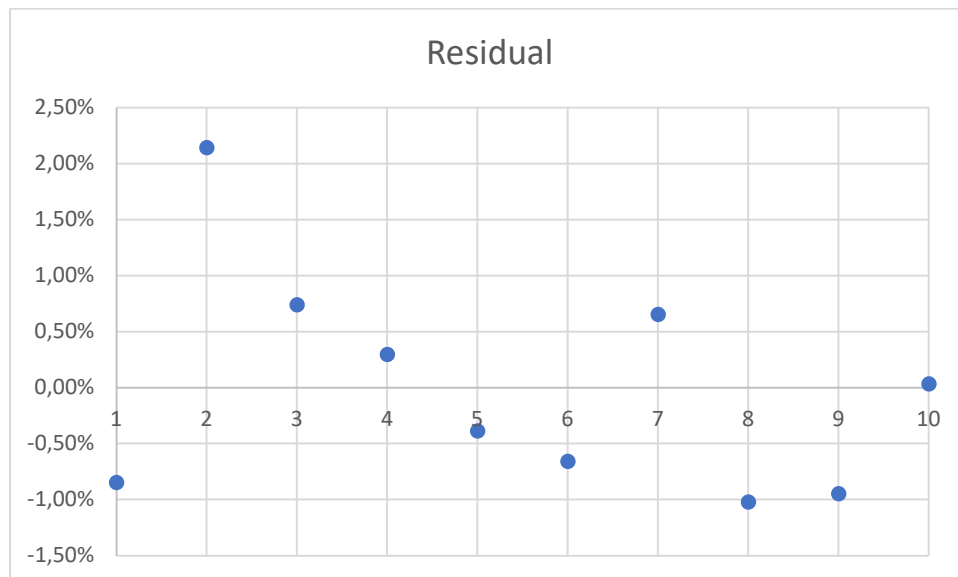
Aufgrund der These aus 2.5, dass eine Reduktion der Gewinnwahrscheinlichkeit durch Verringern der Ziehungen pro Spiel mit sich bringt, sind weitere Durchläufe mit jeweils 15 Ziehungen pro Spiel bei 10 Millionen Durchläufen ausgeführt worden.

2.6.1 Gewinne per Durchlauf



	Σ	Min	Max	Avg
Wins	173.765	17.199	17.749	17.377
Simulations	100.000.000			
rel. Wins %	0,1738%			
Chance	1:575,49			

Die Gewinnverteilung pro Durchlauf ist annähernd fair verteilt. Allerdings hat sich die relative Anzahl der Gewinne von 0,89% auf 0,17% reduziert. Damit sei gezeigt, dass eine Reduktion der Ziehungen die Chance auf einen Gewinn verringert.



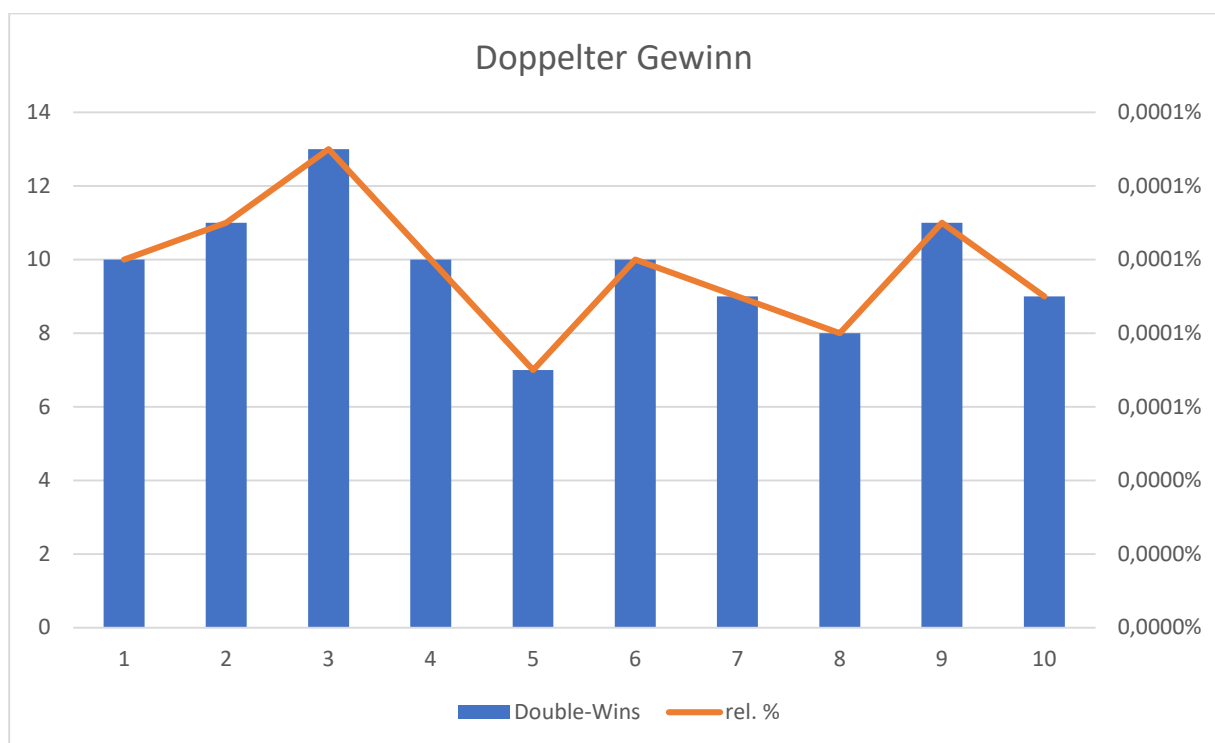
Die prozentuale Abweichung hat sich hier merklich erhöht. Die Gewinne werden nicht nur seltener, sondern auch schwankender.

2.6.2 Doppelter Gewinn

Wie wahrscheinlich ist es, dass auch doppelte Gewinne vorkommen.

Bei 15 Ziehungen gab es während der Simulationsdurchläufe insgesamt 98 Doppelgewinne. Durchschnittlich waren in jedem Spiel 10 Stück.

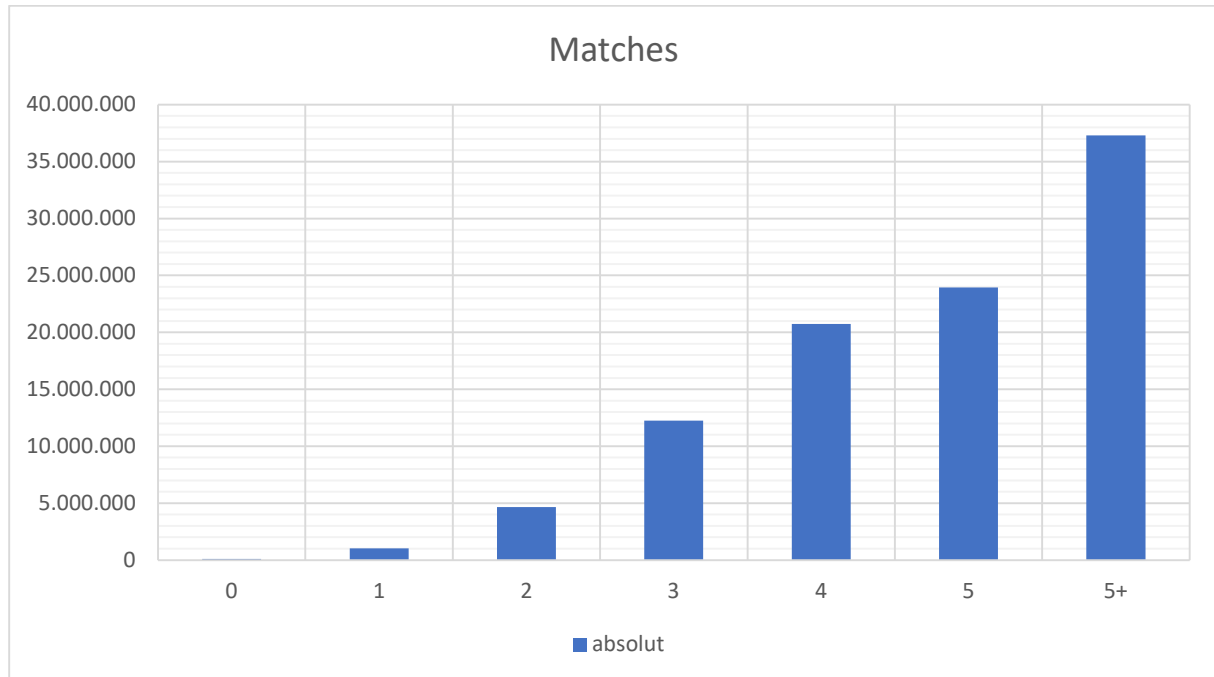
Erwartungsgemäß haben sich die Anzahl der Doppelte Gewinne stark reduziert.



Teilprüfung 2 – Zufall auf dem Computer

2.6.3 Trefferverteilung per Kugeln bei 15 Ziehungen

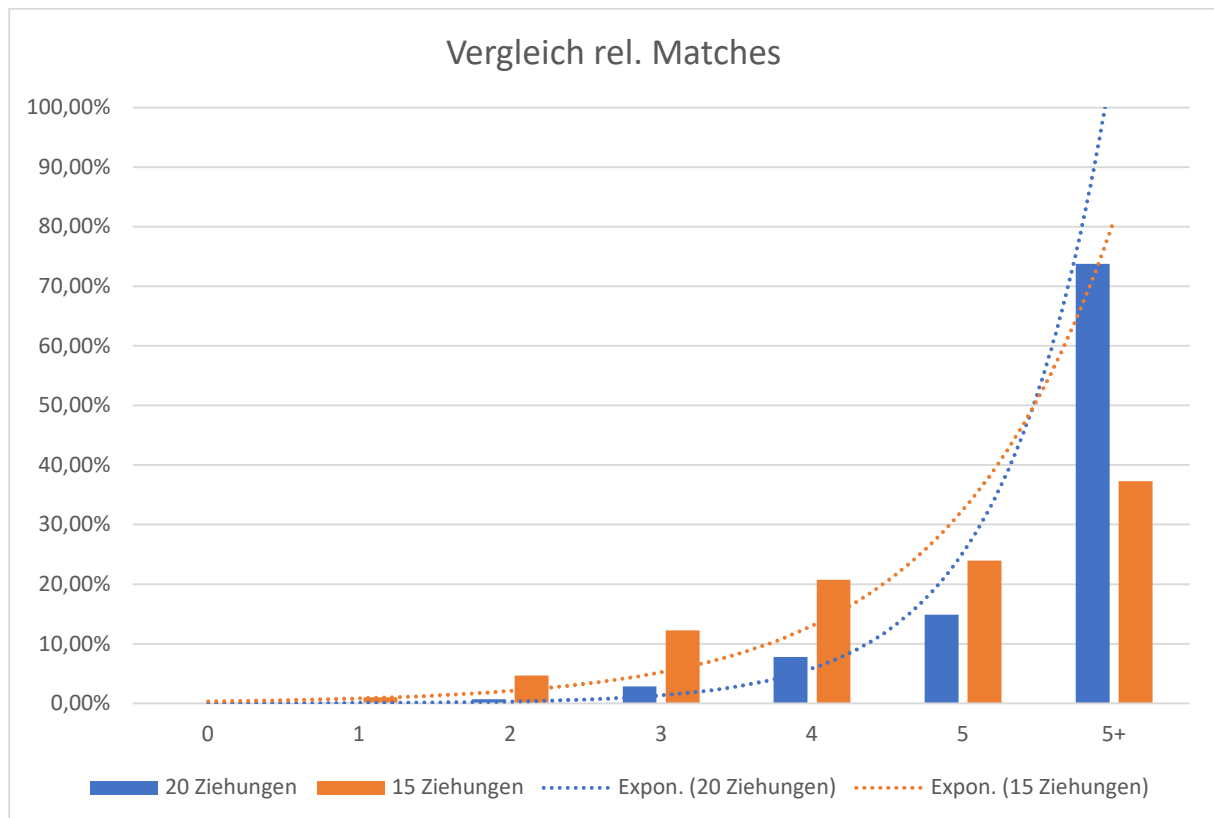
Die Vermutung liegt nahe, dass bei 15 Ziehungen auch die Treffer der Matches fällt.



	absolut			relativ		
	Σ	Min	Max	Σ	Min	Max
Matches						
0	97.946	9.582	9.923	0,10%	0,10%	0,10%
1	1.027.465	102.449	103.086	1,03%	1,02%	1,03%
2	4.669.482	466.187	468.205	4,67%	4,66%	4,68%
3	12.249.256	1.222.992	1.226.422	12,25%	12,23%	12,26%
4	20.728.112	2.070.520	2.074.572	20,73%	20,71%	20,75%
5	23.928.230	2.389.885	2.394.376	23,93%	23,90%	23,94%
5+	37.299.509	3.728.723	3.731.134	37,30%	37,29%	37,31%
Σ	100.000.000					

Die Verteilung hat sich scheinbar nur wenig verändert. Betrachtet man die rohen Zahlen, zeigt sich aber, dass die Anzahl ohne Treffer prozentual um das 10-fache angestiegen ist. Die Zahl der Treffer bei „5 und mehr“ ist dahingehend um ungefähr 50% gesunken.

Teilprüfung 2 – Zufall auf dem Computer

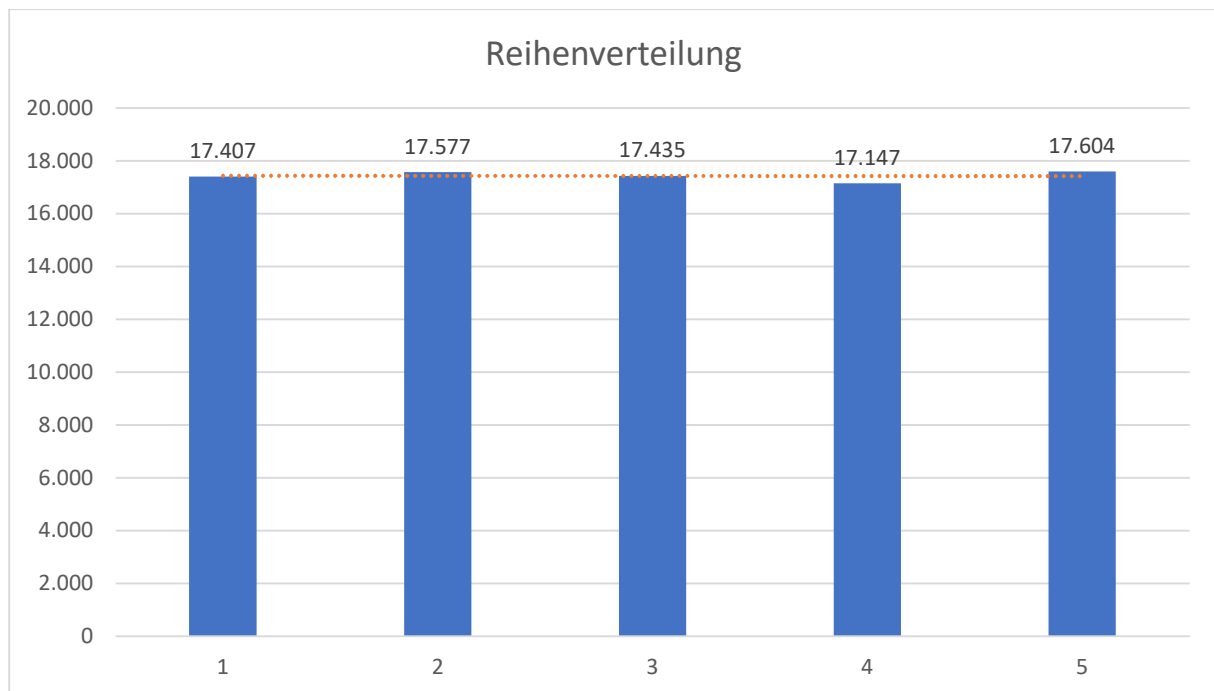
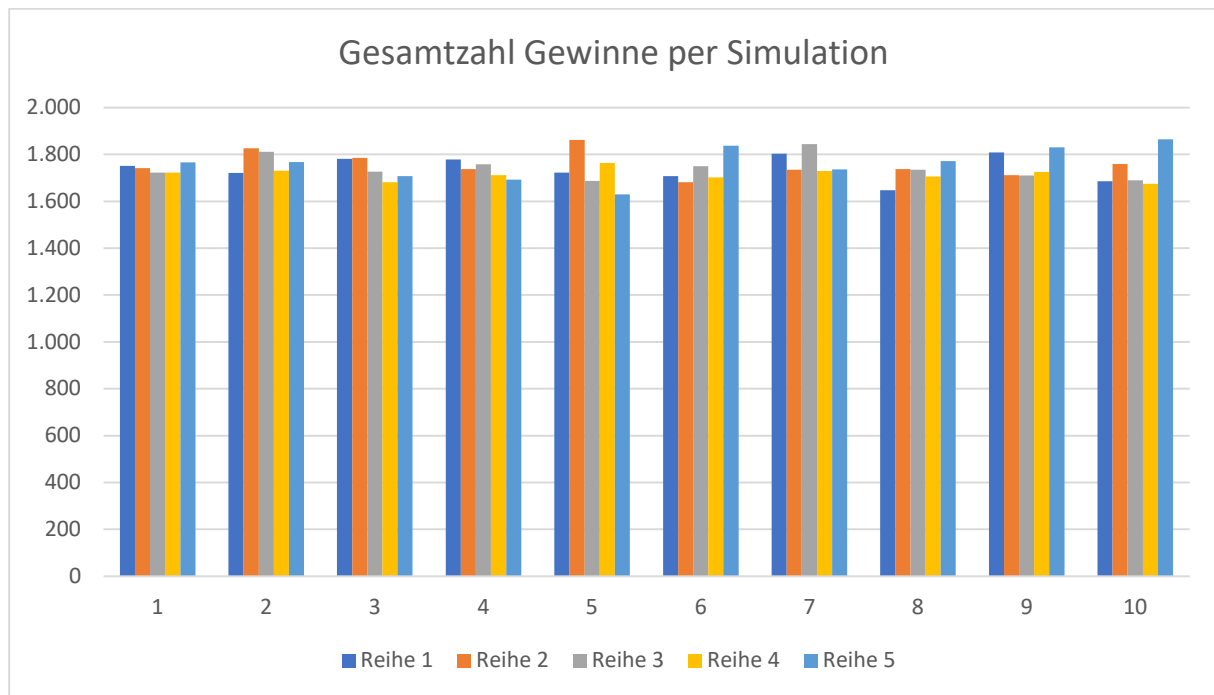


2.6.4 Verteilung der Gewinntreffer per Reihe

Weiterhin wird geprüft, ob sich die Verteilung der Gewinne auf die Reihen sich weiterhin gleich verhält.

	Σ	Avg	Row Win	Row p Win	Row p Game
Row					
1	17.407	1.741	19,97%	10,02%	0,02%
2	17.577	1.758	20,16%	10,12%	0,02%
3	17.435	1.744	20,00%	10,03%	0,02%
4	17.147	1.715	19,67%	9,87%	0,02%
5	17.604	1.760	20,20%	10,13%	0,02%
abs. RowWins	87.170	8.717	100,00%	50,17%	0,09%
Wins	173.765	17.377			
Simulations	100.000.000				

Teilprüfung 2 – Zufall auf dem Computer



Die absolute Verteilung auf die fünf Reihen ist nahe beieinander. Somit ist gezeigt, dass jede Reihe annähernd gleich fair ist.

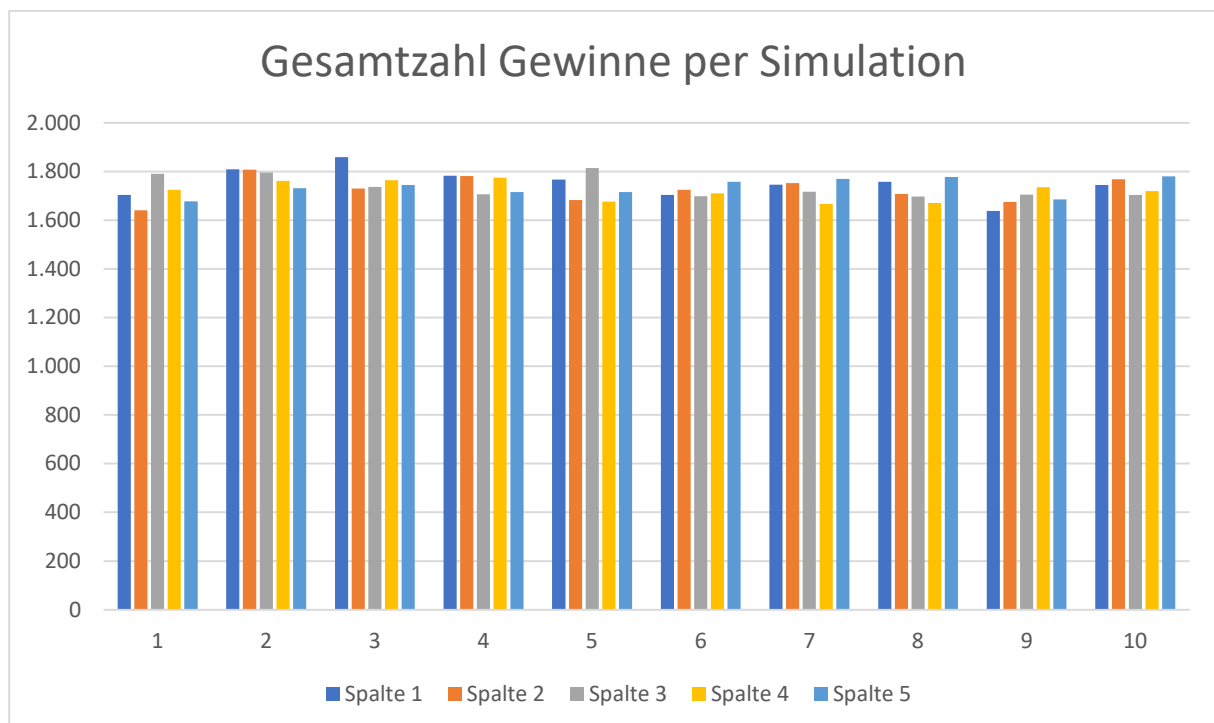
Die relativen Ergebnisse der Verteilung zeigen, dass diese etwa 20% beträgt und pro Spiel 0,09% einen Reihengewinn treffen.

Aufgrund der Tatsache, dass circa 10% der Gewinne auf einer Reihe liegen, folgt erneut eine Untersuchung der Spaltentreffer.

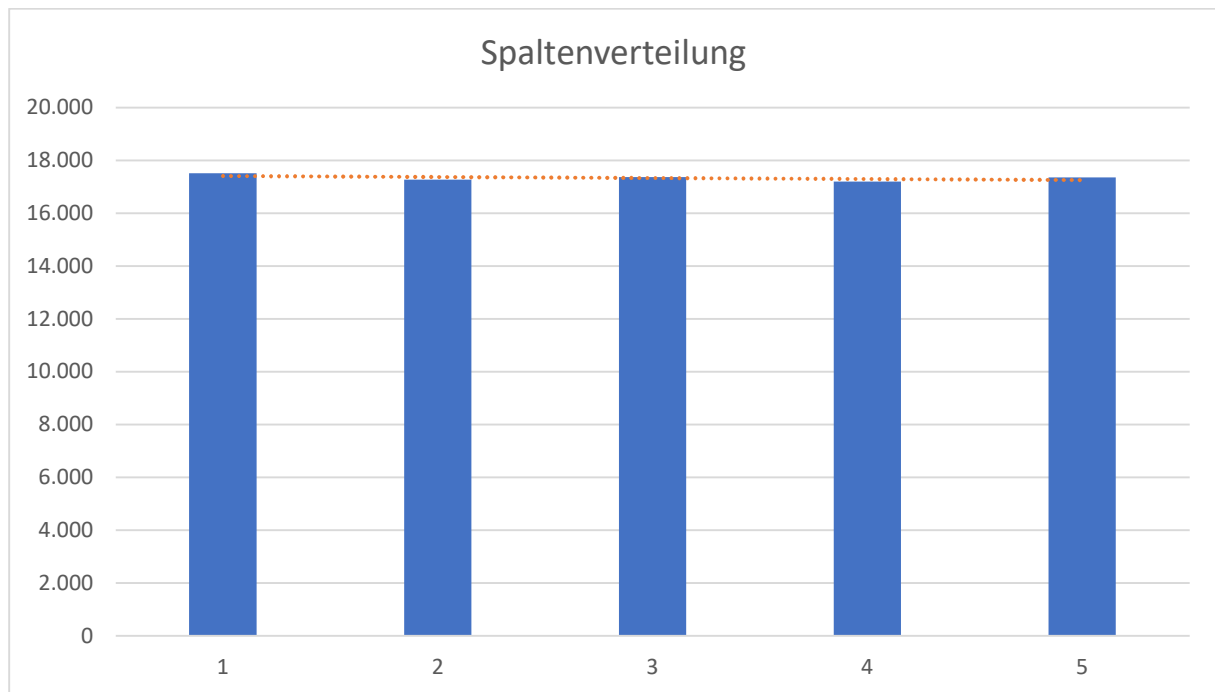
Teilprüfung 2 – Zufall auf dem Computer

2.6.5 Verteilung der Gewinntreffer per Spalte

	Σ	Avg	Col Win	Col p Win	Col p Game
Col					
1	17.509	1.751	20,20%	10,08%	0,02%
2	17.269	1.727	19,92%	9,94%	0,02%
3	17.363	1.736	20,03%	9,99%	0,02%
4	17.201	1.720	19,84%	9,90%	0,02%
5	17.351	1.735	20,01%	9,99%	0,02%
abs. ColWins	86.693	8.669	100,00%	49,89%	0,09%
Wins	173.765	17.377			
Simulations	100.000.000				



Teilprüfung 2 – Zufall auf dem Computer

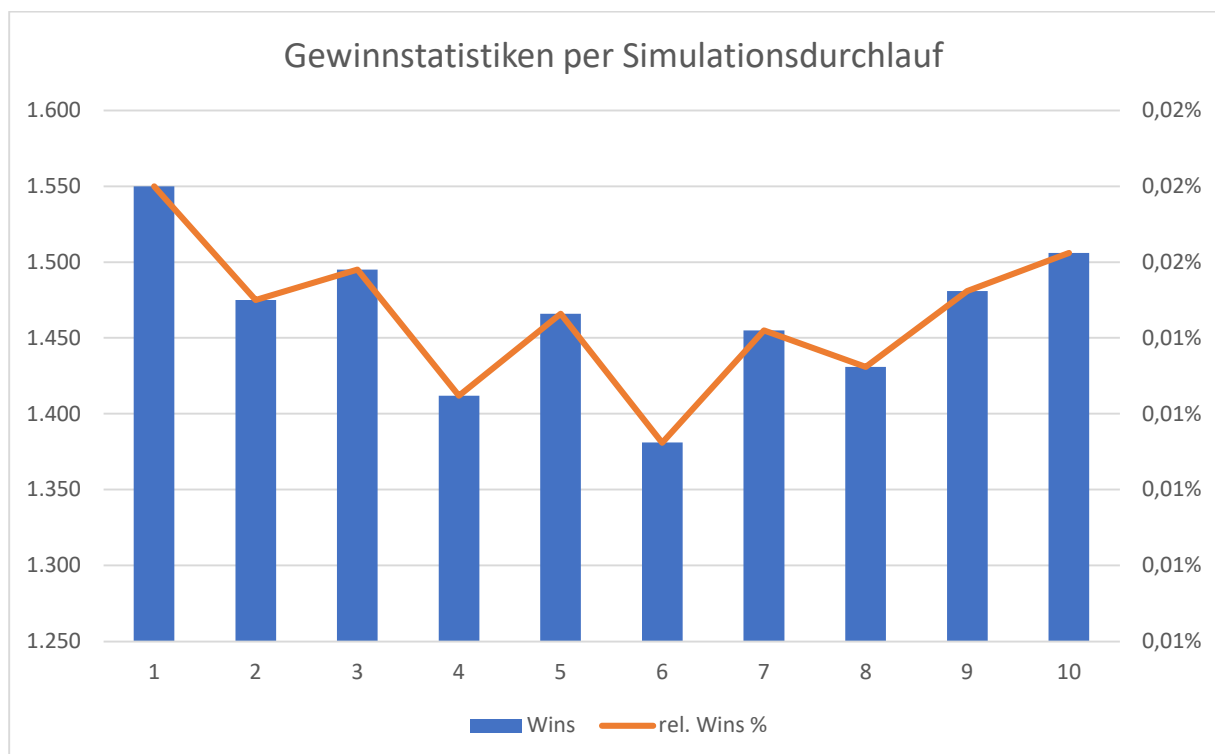


Auch hier ist die Verteilung auf die Spalte jeweils etwa 20%. 0,09% aller Spiele könnten eine Chance auf einen Treffer auf einer Spalte geben.

2.7 Durchführung mit 10 Ziehungen pro Simulation

Um die vorangegangenen Aussagen zu referenzieren, wird eine weitere Simulation mit jeweils 10 Ziehungen pro Spielrunde durchgeführt und aufgezeichnet.

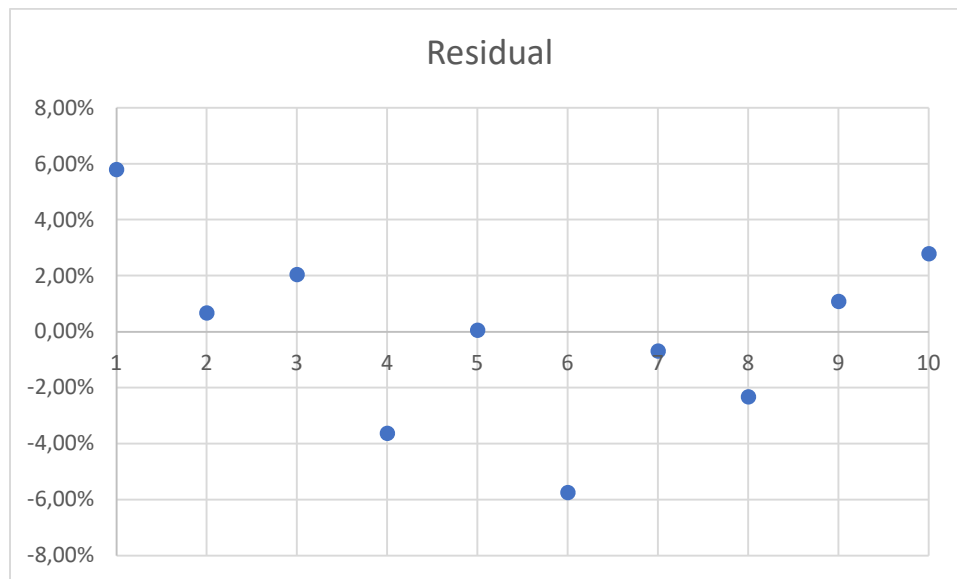
2.7.1 Gewinne per Durchlauf



Teilprüfung 2 – Zufall auf dem Computer

	Σ	Min	Max	Avg
Wins	14.652	1.381	1.550	1.465
Simulations	100.000.000			
rel. Wins %	0,0147%			
Chance	1:6825,01			

Die relativen Gewinne sind um weitere 0,17% gefallen. Bei Halbierung der Ziehungen, reduziert sich die Gewinnwahrscheinlichkeit um etwa das 90-fache.



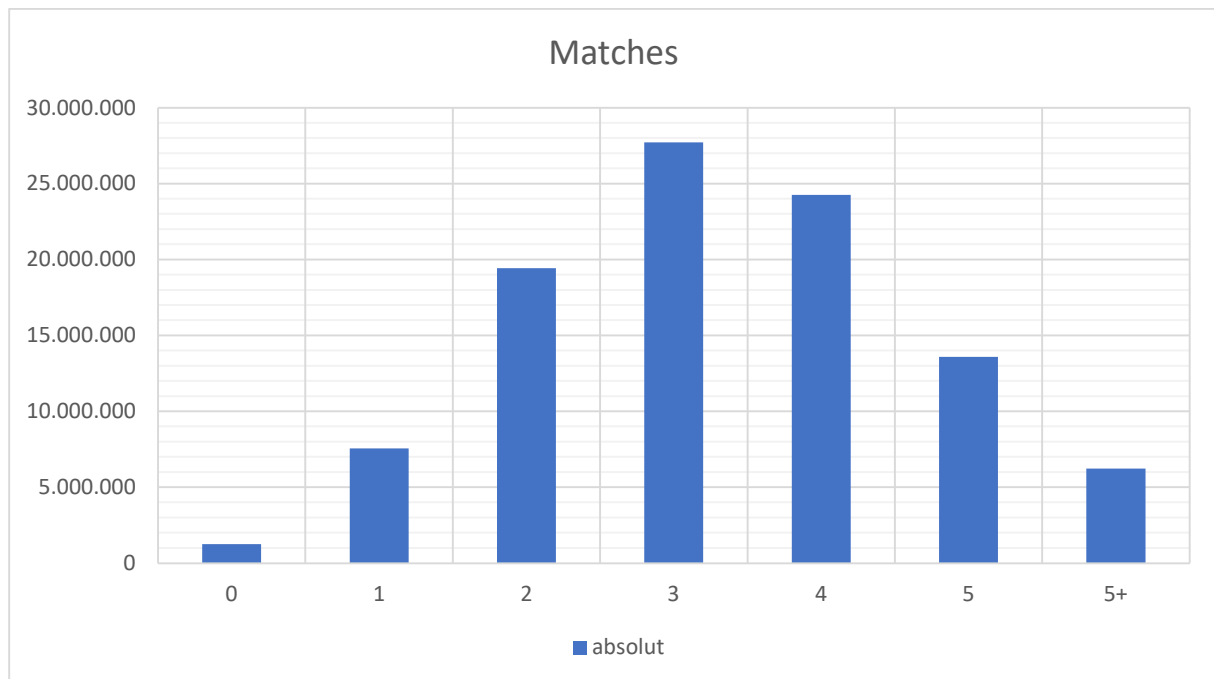
Die Streuung um den Mittelwert vergrößert sich.

2.7.2 Doppelte Gewinne

Bei 10 Ziehungen pro Durchlauf ergaben sich keine doppelten Gewinne mehr.

Teilprüfung 2 – Zufall auf dem Computer

2.7.3 Trefferverteilung per Kugeln bei 10 Ziehungen

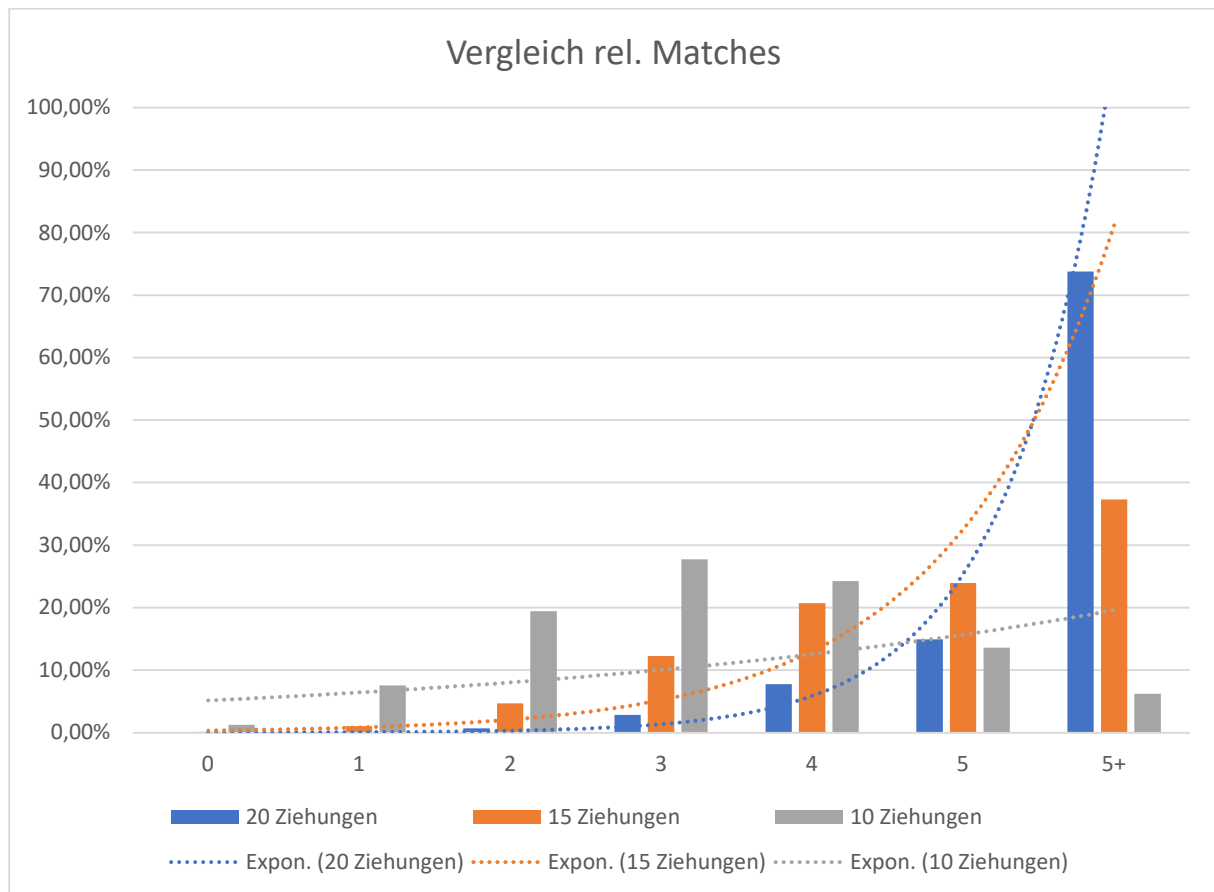


Die Verteilung der Treffer verschiebt sich stärker nach unten. Etwa 28% der Ziehungen treffen nur drei Zahlen auf einer Spielkarte. Die Wahrscheinlichkeit 5 oder mehr zu treffen, oder nur 1 Zahl ist fast gleich hoch.

	absolut			relativ		
	Σ	Min	Max	Σ	Min	Max
Matches						
0	1.240.136	123.075	124.648	1,24%	1,23%	1,25%
1	7.559.674	754.570	757.500	7,56%	7,55%	7,58%
2	19.428.707	1.941.586	1.945.283	19,43%	19,42%	19,45%
3	27.715.647	2.767.874	2.774.863	27,72%	27,68%	27,75%
4	24.242.498	2.422.335	2.427.601	24,24%	24,22%	24,28%
5	13.584.596	1.357.085	1.359.822	13,58%	13,57%	13,60%
5+	6.228.742	621.813	623.928	6,23%	6,22%	6,24%
Σ	100.000.000					

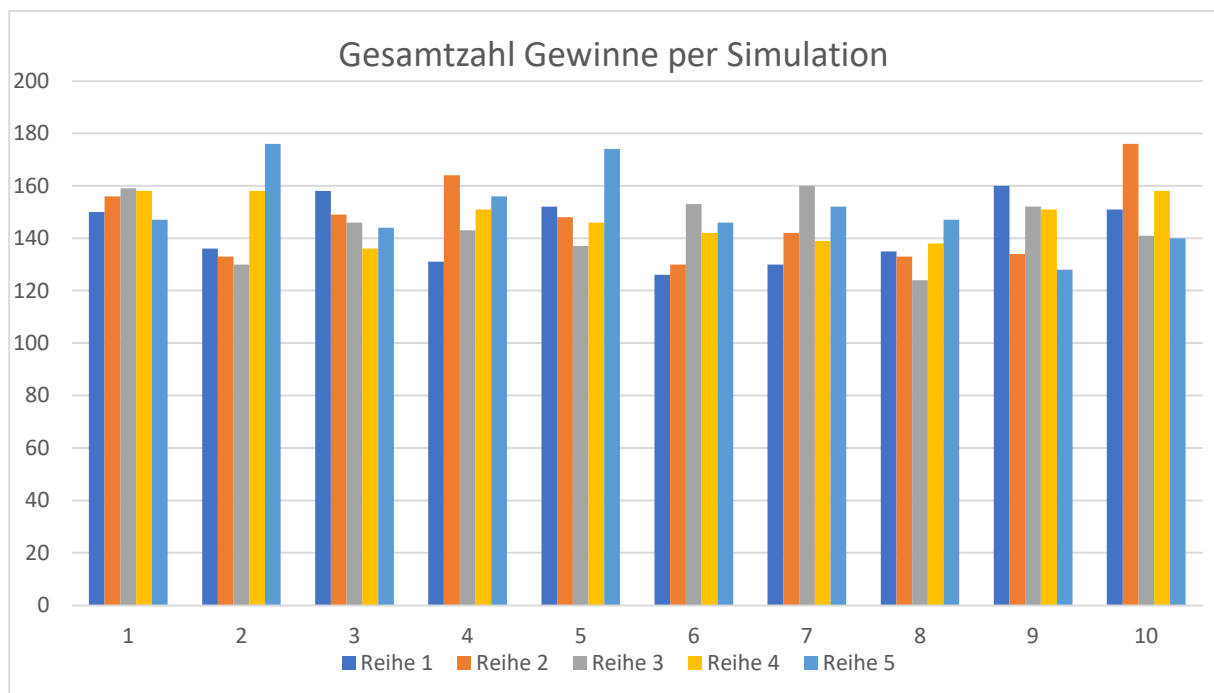
Die Trefferquote für kein Match ist weiter um das 12-fache gestiegen und mittlerweile 124-mal so hoch wie bei 20 Ziehungen.

Teilprüfung 2 – Zufall auf dem Computer

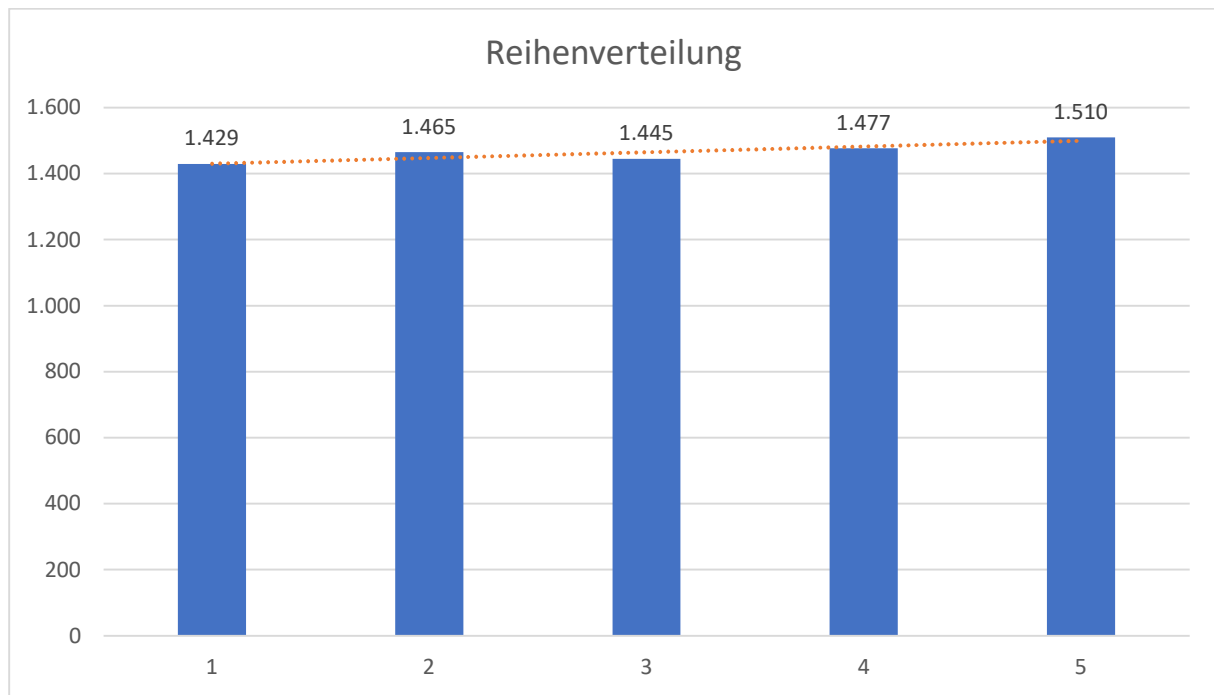


2.7.4 Verteilung der Gewinntreffer per Reihe

Die Verteilung auf die Reihen verhält sich ähnlich den vorangegangenen Ergebnissen und ist verhältnismäßig stetig. Die optischen Abweichungen resultieren aus den kleinen Gesamtwerten.



Teilprüfung 2 – Zufall auf dem Computer



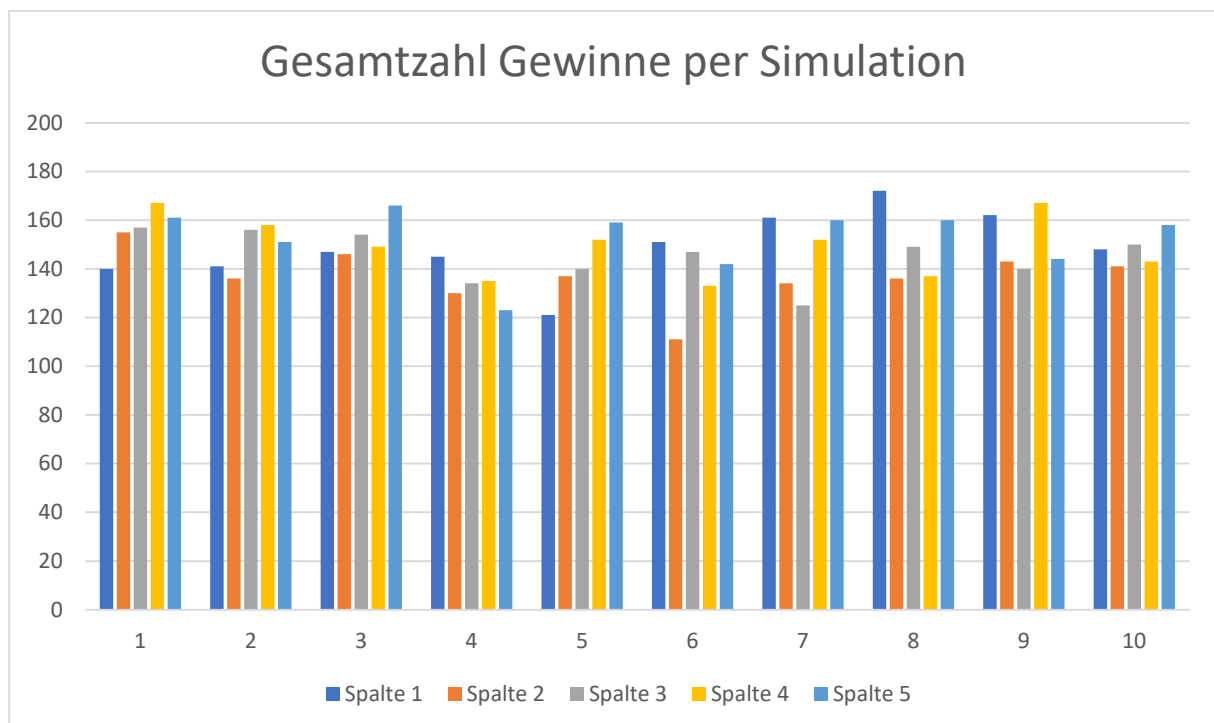
	Σ	Avg	Row Win	Row p Win	Row p Game
Row					
1	1.429	143	19,51%	9,75%	0,00%
2	1.465	147	20,00%	10,00%	0,00%
3	1.445	145	19,72%	9,86%	0,00%
4	1.477	148	20,16%	10,08%	0,00%
5	1.510	151	20,61%	10,31%	0,00%
abs. RowWins	7.326	733	100,00%	50,00%	0,01%
Wins	14.652	1.465			
Simulations	100.000.000				

Die relativen Ergebnisse der Verteilung zeigen, dass diese immer noch etwa 20% beträgt und pro Spiel 0,01% einen Reihengewinn treffen.

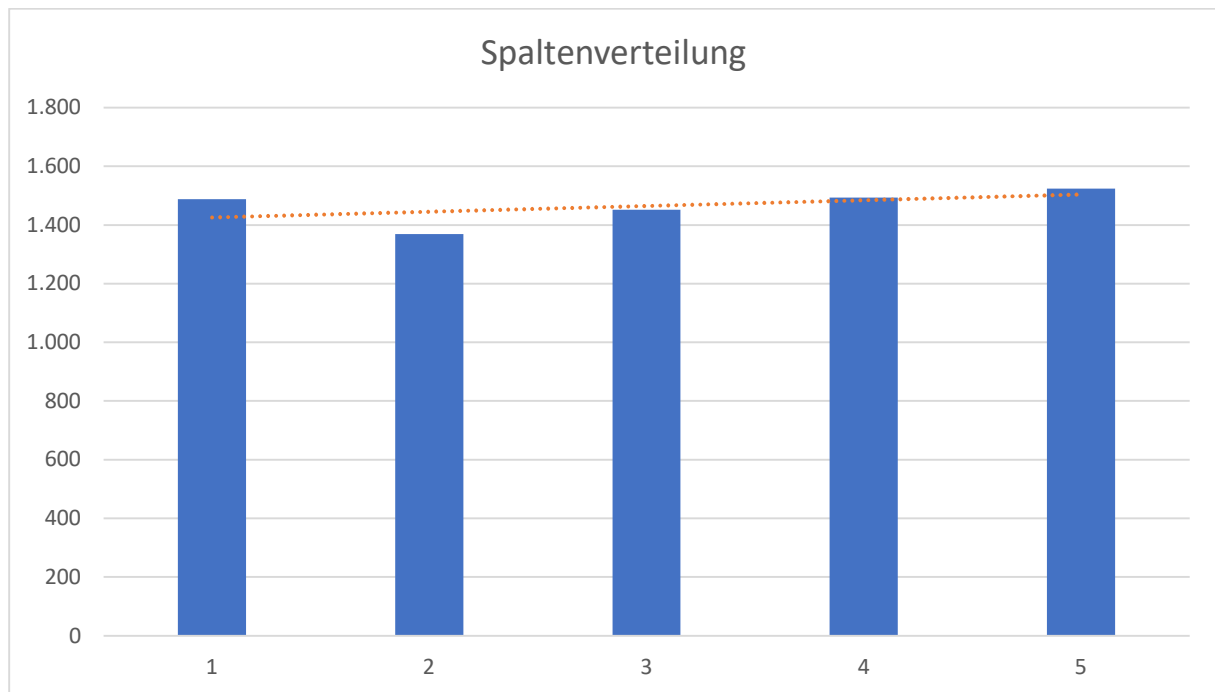
Teilprüfung 2 – Zufall auf dem Computer

2.7.5 Verteilung der Gewinntreffer per Spalte

	Σ	Avg	Col Win	Col p Win	Col p Game
Col					
1	1.488	149	20,31%	10,16%	0,00%
2	1.369	137	18,69%	9,34%	0,00%
3	1.452	145	19,82%	9,91%	0,00%
4	1.493	149	20,38%	10,19%	0,00%
5	1.524	152	20,80%	10,40%	0,00%
abs. ColWins	7.326	733	100,00%	50,00%	0,01%
Wins	14.652	1.465			
Simulations	100.000.000				



Teilprüfung 2 – Zufall auf dem Computer

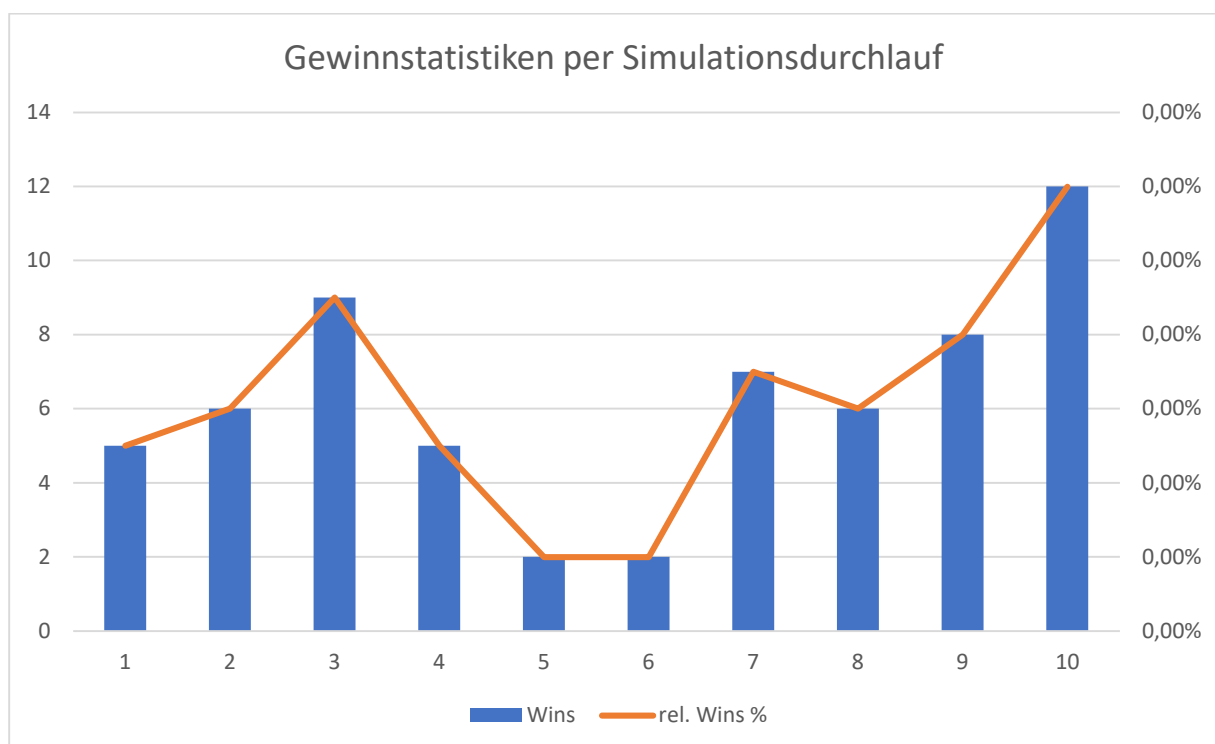


Bei den Spalten zeichnet sich weiterhin das gleiche Bild ab, wie bei den Reihenuntersuchungen und den vorangegangenen Ziehungen.

2.8 Durchführung mit 5 Ziehungen pro Simulation

Der Vollständigkeit wegen und der starken Verschiebungen der Trefferwahrscheinlichkeiten, werden weitere Simulationen mit lediglich 5 Ziehungen durchgeführt.

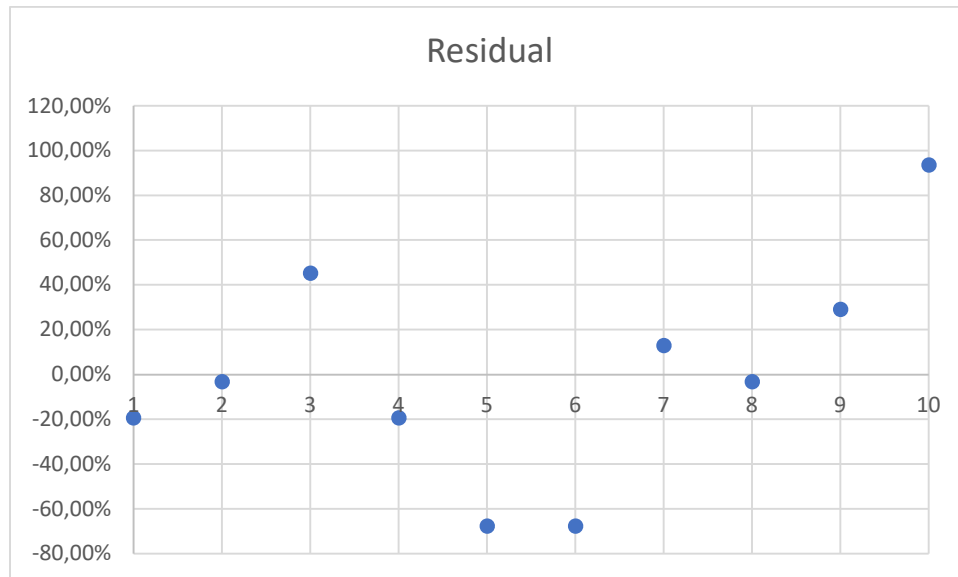
2.8.1 Gewinne per Durchlauf



Teilprüfung 2 – Zufall auf dem Computer

	Σ	Min	Max	Avg
Wins	62	2	12	6
Simulations	100.000.000			
rel. Wins %	0,0001%			
Chance	1:1612903,23			

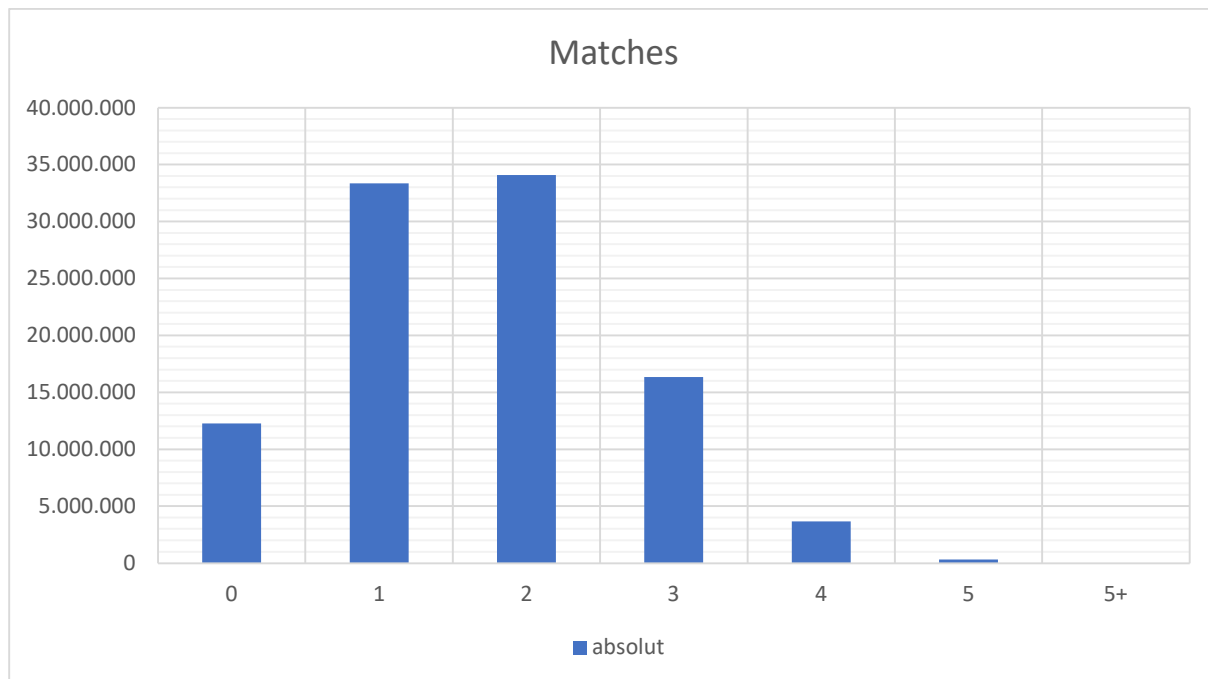
Die relative Wahrscheinlichkeit auf einen Gewinn ist durch Halbierung der Ziehungen um das 147-fache gesunken. Die Chance steht bei 1:1,6 Millionen.



Das Streuungsergebnis aus dem Residualplot zeigt, dass die Schwankungen bei den wenigen Gewinnen stark sind.

Teilprüfung 2 – Zufall auf dem Computer

2.8.2 Trefferverteilung per Kugeln bei 5 Ziehungen

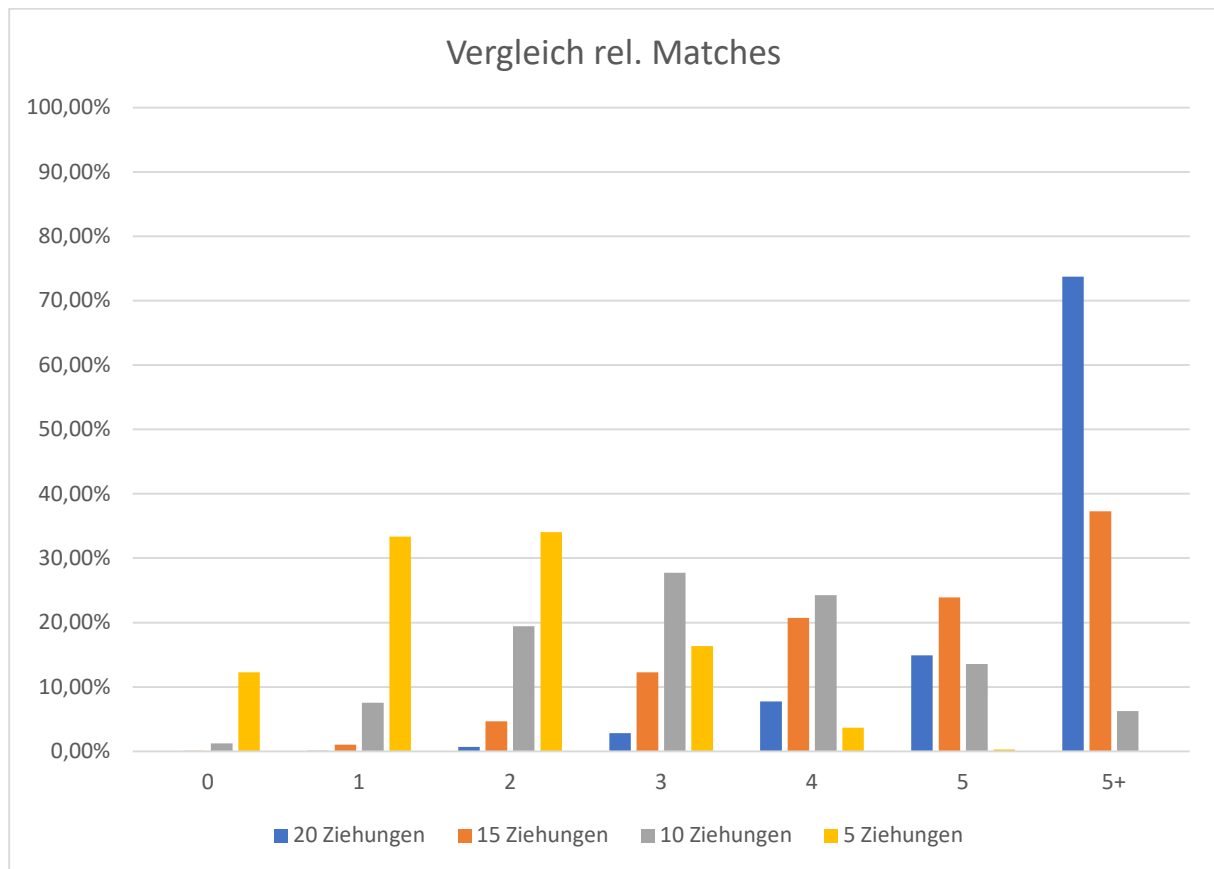


Die Verteilung der Treffer verschiebt sich stärker nach unten. Jeweils ein Drittel der Ziehungen treffen nur eine oder zwei Zahlen auf einer Spielkarte. Die Wahrscheinlichkeit 5 zu treffen tendiert gegen 0.

	absolut			relativ		
	Σ	Min	Max	Σ	Min	Max
Matches						
0	12.269.715	1.224.686	1.229.663	12,27%	12,25%	12,30%
1	33.353.086	3.333.482	3.338.379	33,35%	33,33%	33,38%
2	34.073.554	3.405.743	3.410.274	34,07%	34,06%	34,10%
3	16.332.473	1.631.777	1.634.920	16,33%	16,32%	16,35%
4	3.664.139	364.992	367.673	3,66%	3,65%	3,68%
5	307.033	30.407	30.869	0,31%	0,30%	0,31%
5+	0	0	0	0,00%	0,00%	0,00%
Σ	100.000.000					

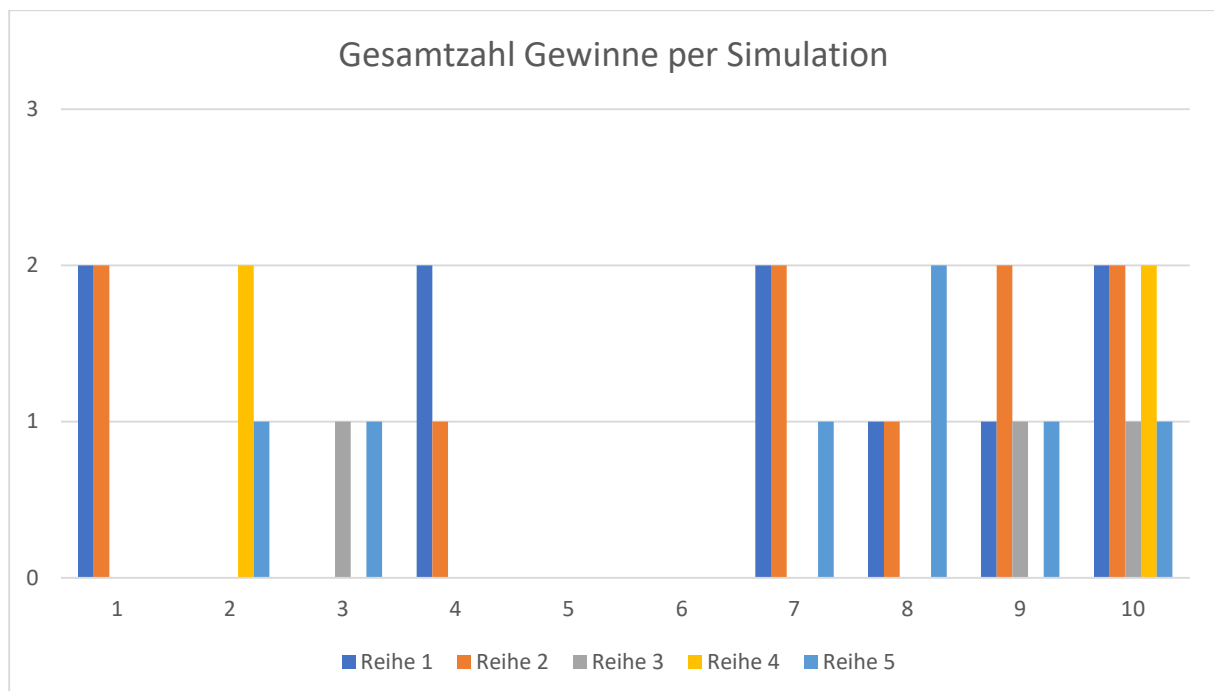
Die Trefferquote für kein Match ist weiter um das 12-fache gestiegen und mittlerweile 2111-mal so hoch als bei 20 Ziehungen.

Teilprüfung 2 – Zufall auf dem Computer



Die Verschiebung der durchschnittlichen Treffer ist hier im Vergleich deutlich zu erkennen.

2.8.3 Verteilung der Gewinntreffer per Reihe

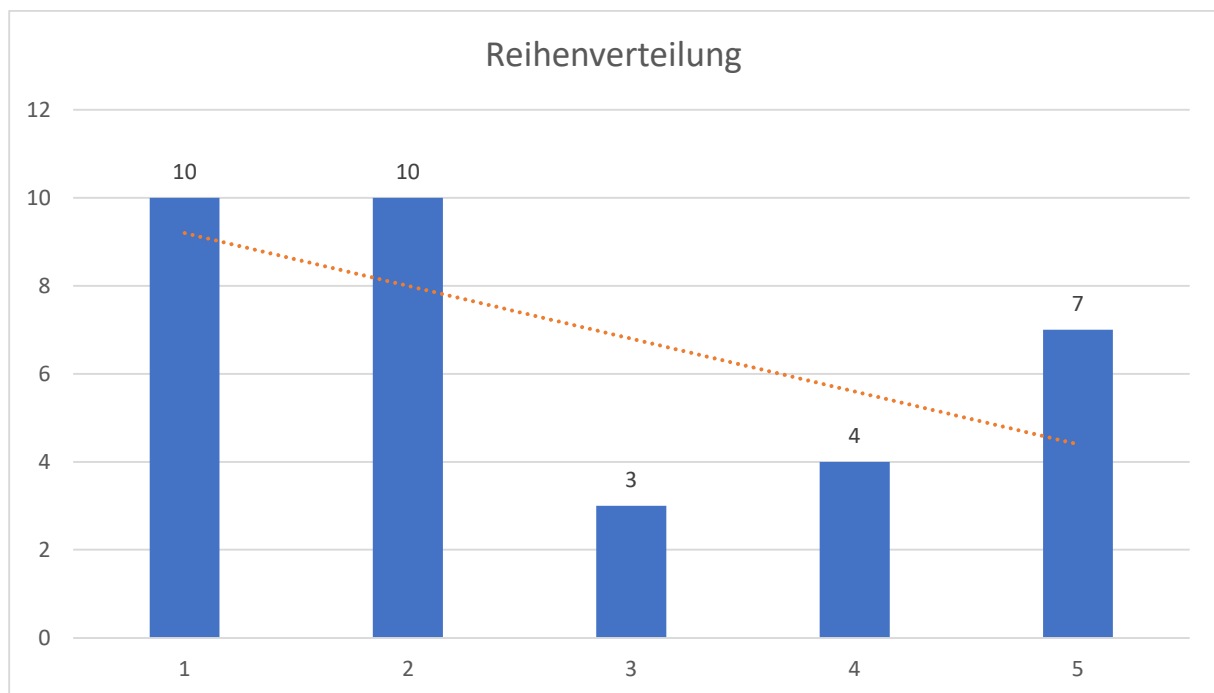


Die Trefferverteilung bezogen auf die Reihen ist bei der geringen Anzahl nicht mehr aussagekräftig.

Teilprüfung 2 – Zufall auf dem Computer

	Σ	Avg	Row Win	Row p Win	Row p Game
Row					
1	10	1	29,41%	16,13%	0,00%
2	10	1	29,41%	16,13%	0,00%
3	3	0	8,82%	4,84%	0,00%
4	4	0	11,76%	6,45%	0,00%
5	7	1	20,59%	11,29%	0,00%
abs. RowWins	34	3	100,00%	54,84%	0,00%
Wins	62	6			
Simulations	100.000.000				

Eine faire Verteilung hätte ein Ergebnis von $34/5 = 6,8$ Treffer pro Reihe ergeben.

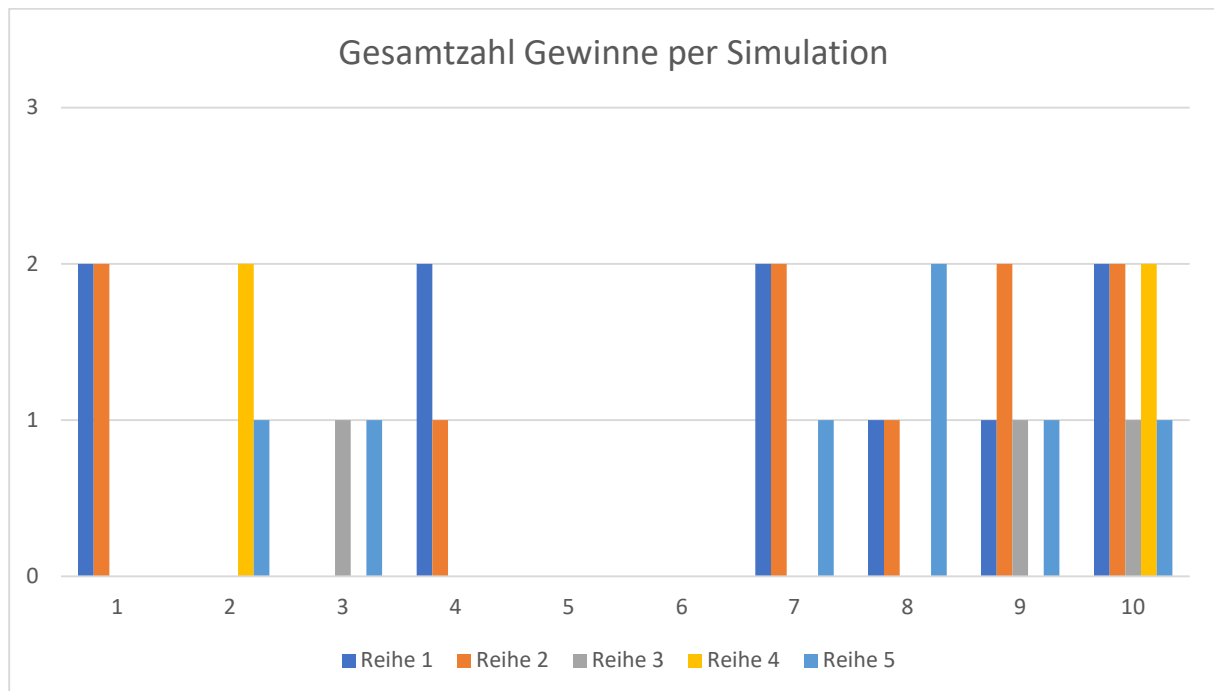


Tatsächlich aussagen lässt lässt sich lediglich, dass etwa 55% aller Gewinne eine Reihe treffen, aber die relative Wahrscheinlichkeit eine Reihe zu treffen bei 0% liegt.

Teilprüfung 2 – Zufall auf dem Computer

2.8.4 Verteilung der Gewinntreffer per Spalte

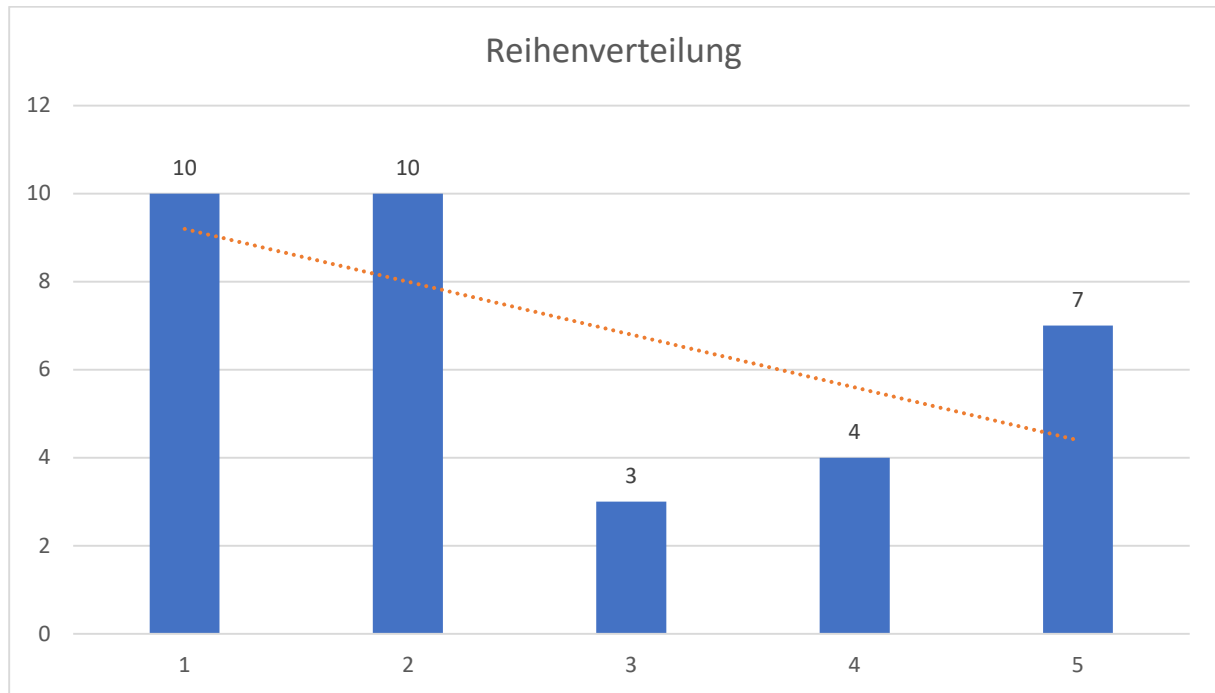
Auch bei der Spaltenverteilung zeichnet sich ein ähnliches Bild wie bei der Reihenverteilung ab.



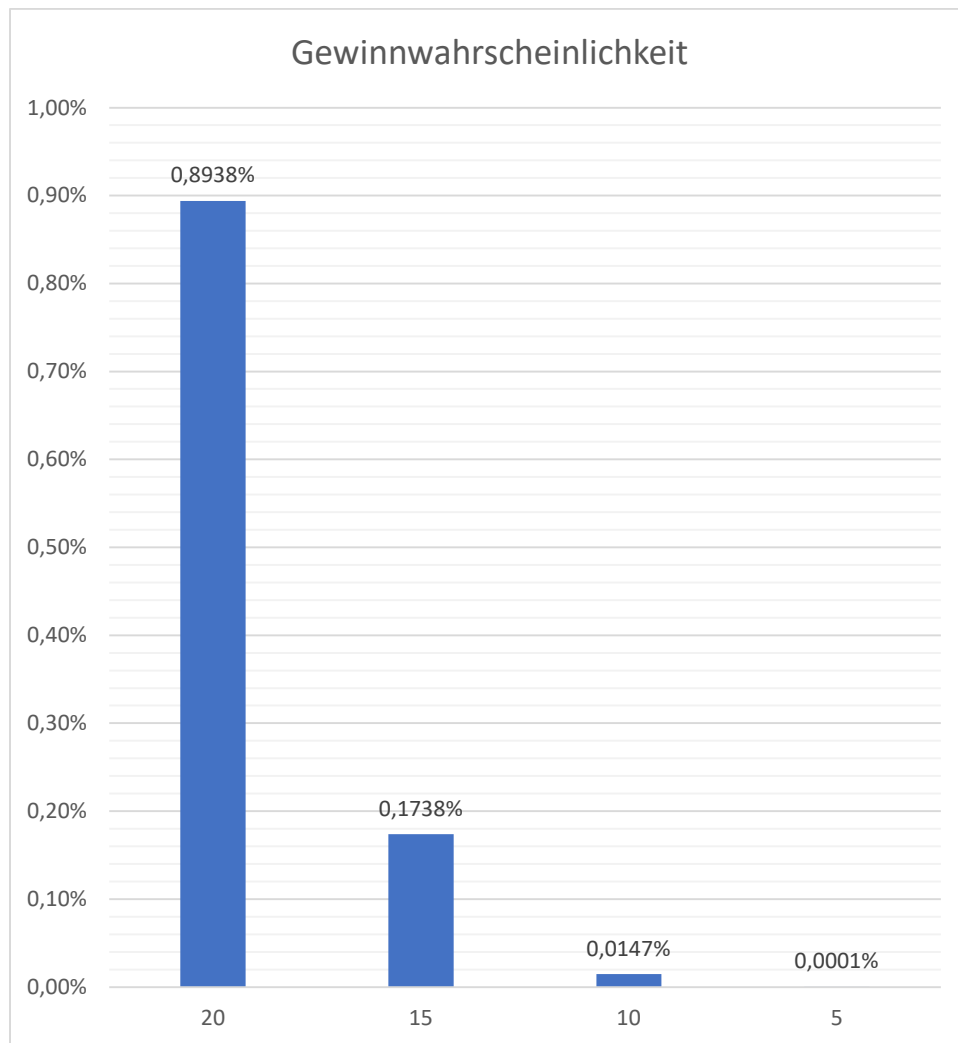
Eine Gleichverteilung ist nicht erkennbar.

	Σ	Avg	Row Win	Row p Win	Row p Game
Row					
1	10	1	29,41%	16,13%	0,00%
2	10	1	29,41%	16,13%	0,00%
3	3	0	8,82%	4,84%	0,00%
4	4	0	11,76%	6,45%	0,00%
5	7	1	20,59%	11,29%	0,00%
abs. RowWins	34	3	100,00%	54,84%	0,00%
Wins	62	6			
Simulations	100.000.000				

Teilprüfung 2 – Zufall auf dem Computer



2.9 Vergleich der Gewinnchancen abhängig der Ziehungen



3 Resümee

Neben der Herausforderung das Spiel mittels JAVA zu implementieren, galt es eine geeignete Simulationsbasis zu entwickeln. Durch das Trennen beider Teile, ist das Spiel auch für weitere Anwendungen oder Analysen bereit.

Die Ergebnisse zeigen, dass das Spiel Bingo je nach Absicht des Veranstalters durch geringen Aufwand zu beeinflussen ist. Entweder durch verändern der zu ziehenden Zahlen, Verändern der Karte oder durch das Festlegen von Gewinnbedingungen, wie mehrfach Treffer als Gewinnvoraussetzung.

Weiterhin hat sich gezeigt, dass die Anzahl der Simulationsläufe das Ergebnis beeinflusst. Je weniger Durchläufe, desto ungenauer beziehungsweise weniger aussagekräftig sind die Ergebnisse.

Je mehr Ziehungen vorgenommen werden, desto höher ist die Gewinnwahrscheinlichkeit.

In der Praxis werden häufig 5 aus 22 gewählt, was eine hohe Chance bietet zu Gewinnen. Allerdings ist der Anteil der Gewinnsumme und der absolute Gewinn sehr gering. Internetseiten von Spielhäusern bieten oft die Varianten aus 1.4 an und schreiben einen höheren Gewinn bei der Variante mit der geringsten Gewinnwahrscheinlichkeit aus.

3.1 Software-Erweiterung

Mögliche Erweiterungen des Bingo-Spiels sind unter anderem das Hinzufügen eines „Joker“-Feldes in der Mitte oder das Markieren der getroffenen Zahlen durch Ersetzen mit zum Beispiel einem „X“.

Zusätzlich ist es denkbar eine passende GUI sowohl für das Spiel als auch für die Simulation zu entwickeln, um ein entsprechendes Analyseerlebnis zu bieten.

3.2 Weitere Analysen

Im weiteren Verlauf wäre es interessant zu wissen, wie viele Spiele es im Schnitt mindestens benötigt, um ein Bingo zu treffen. Dazu können zum Beispiel die bereits generierten „Roundtrip“-Files benutzt werden.

Im Rahmen der möglichen Erweiterungen stellt sich die Frage, wie wirkt sich das „Joker“-Feld auf die Gewinnchance aus und wie ändert sich die tatsächliche Wahrscheinlichkeit durch Erhöhung der Ziehungen, bei gleichbleibender Anzahl an Gewinnzahlen.

Teilprüfung 2 – Zufall auf dem Computer

4 Quellen

[WWW 01] **Bingo, Suchtrisiko Bingo**, zuletzt abgerufen: 07.06.2021

<https://www.spielen-mit-verantwortung.de/gluecksspiel/uebersicht/suchtrisiko-bingo/>

[WWW 02] **75-Ball Bingo**, zuletzt abgerufen: 07.06.2021

https://de.wikipedia.org/wiki/Bingo#75-Ball_Bingo

[WWW 03] **DAS IST BINGO, Lotto im MV**, zuletzt abgerufen: 07.06.2021

<https://www.lottomv.de/sonstige/bingo/spielanleitung?gbn=16>

[WWW 04] **OpenCSV Example**, zuletzt abgerufen: 10.06.2021

<https://mkyong.com/java/how-to-export-data-to-csv-file-java/>

[WWW 05] **OpenCSV MavenRepository Download**, zuletzt abgerufen: 10.06.2021

<https://mvnrepository.com/artifact/com.opencsv/opencsv>

[WWW 06] **Apache Commons -Package Download**, zuletzt abgerufen: 10.06.2021

http://commons.apache.org/proper/commons-math/download_math.cgi

[WWW 07] **Dokumentation Random – Class**, zuletzt abgerufen: 11.06.2021

<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

[WWW 08] **Dokumentation MersenneTwister – Class**, zuletzt abgerufen: 11.06.2021

<https://commons.apache.org/proper/commons-math/javadocs/api-3.3/org/apache/commons/math3/random/MersenneTwister.html>

[WWW 09] **progressbar – Class, tongfei**, zuletzt abgerufen: 11.06.2021

<https://github.com/ctongfei/progressbar>

Teilprüfung 2 – Zufall auf dem Computer

5 Dokumentation

5.1 Bingo.java

Class Bingo

Vor dem Starten sollten in dieser Klasse die Grundparameter in den ersten sechs Zeilen der Klasse festgelegt werden. Das betrifft die Größe des Bingo-Feldes (sollte quadratisch sein, da dies in diversen Methoden per Zusicherung abgefangen wird!), die minimale und maximale Größe der Bingo-Werte, mit denen die Karte gefüllt wird und die Anzahl der Ziehungen pro Spiel.

Um ein Bingospiel spielen zu können, muss die Klasse instanziiert werden. Optional kann mit der Methode `setLogger(boolean switch)` eine erweiterte Konsolenausgabe ein- oder ausgeschaltet werden. Mittels `setNoDrawings(int number)` kann der default-Wert für die Anzahl der Ziehungen angepasst werden. Diese Methode ist vor allem dann interessant, wenn man automatisiert mehrere Simulationen durchlaufen will, mit variablen Ziehungen.

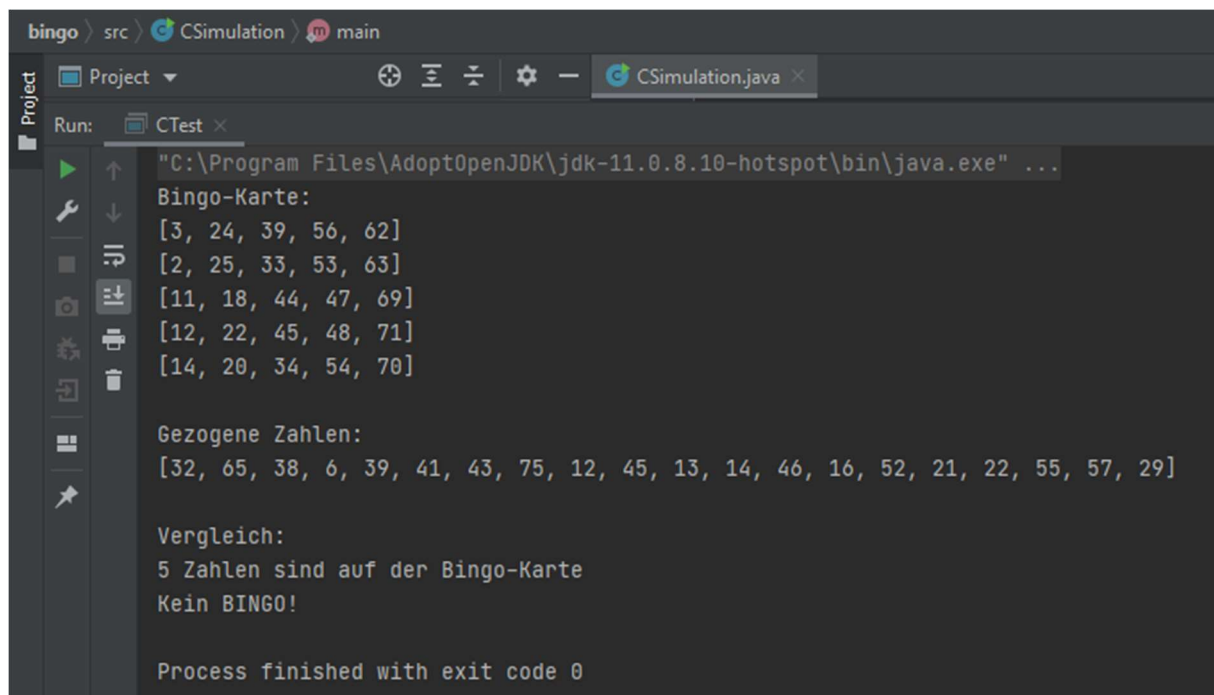
Zu erst muss eine Bingo-Karte erzeugt werden, in dem man ein `int`-Multi-Array mit `createCard()` füllt. Mit `showBingoCard(int[][] bingocard)` kann diese in der Konsole quadratisch ausgegeben werden. Wie bei einem realen Spiel, werden nun die definierte Anzahl an Zahlen gezogen, mit Hilfe der Methode `drawNumbers()`.

Um auf einen Gewinn zu prüfen, kann auf ein `boolean` mit `checkBingo(int[][] bingocard, int[] numbercard)` referenziert werden, um beispielsweise eine Ausgabe zu generieren.

Zur Erzeugung der Pseudozufallszahlen wird ein *MersenneTwister* und ein *RandomDataGenerator* aus der Bibliothek `org.apache.commons.math3.random` benutzt

Class CTest

Es befindet sich eine Testklasse innerhalb der Datei, die ausgeführt werden kann, um eine Partie Bingo zu spielen.



```
bingo > src > CSimulation > main
Run: CTest x
"C:\Program Files\AdoptOpenJDK\jdk-11.0.8.10-hotspot\bin\java.exe" ...
Bingo-Karte:
[3, 24, 39, 56, 62]
[2, 25, 33, 53, 63]
[11, 18, 44, 47, 69]
[12, 22, 45, 48, 71]
[14, 20, 34, 54, 70]

Gezogene Zahlen:
[32, 65, 38, 6, 39, 41, 43, 75, 12, 45, 13, 14, 46, 16, 52, 21, 22, 55, 57, 29]

Vergleich:
5 Zahlen sind auf der Bingo-Karte
Kein BINGO!

Process finished with exit code 0
```

Teilprüfung 2 – Zufall auf dem Computer

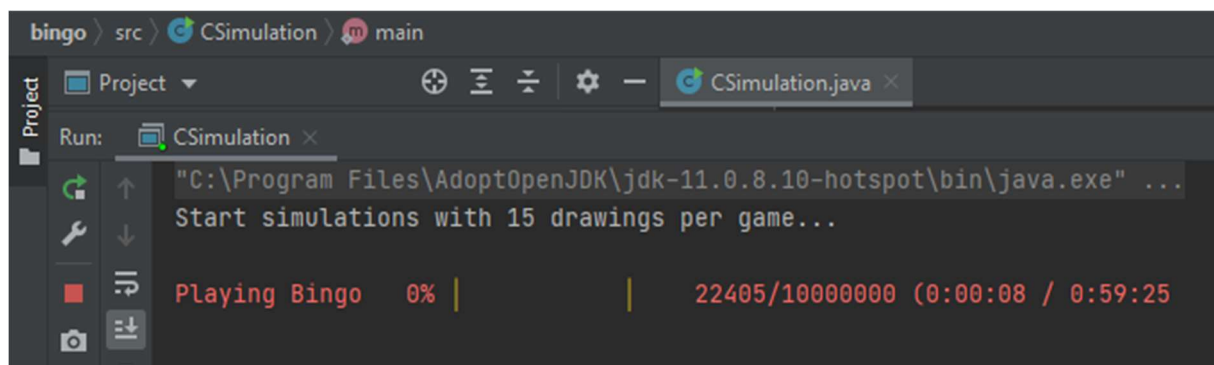
5.2 CSimulation.java

Mit Hilfe der Klasse CSimulation kann das Bingo-Spiel automatisiert durchlaufen werden. Dazu können am Anfang der Klasse 4 Parameter eingestellt werden, mit denen die Anzahl der Spielsimulationen festgelegt wird, die Anzahl der Ziehungen, der Name der Ausgabedatei (csv-Format) und der Name der roundtrip-tabelle (csv-Format).

Zu Beginn der Hauptmethode zum Ausführen der Datei, wird mit einer int-Variable die Anzahl der Durchläufe definiert. (default = 10).

Als erstes wird ein Statusbalken [WWW09] initiiert, um den Fortschritt innerhalb der Konsole grafisch darzustellen. Im Anschluss werden die eigentlichen Simulationen gestartet. Die Simulation definiert den Logger-Zustand, setzt die Anzahl an Ziehungen, erzeugt eine Bingokarte, erzeugt die Ziehungen und prüft anschließend auf Gewinn. Sollte ein Gewinn vorliegen, werden Parameter wie, in welcher Reihe, Spalte oder Diagonal fand der Gewinn statt, gab es einen Mehrfachgewinn, die Methode speichert die laufende Nummer, in der der Gewinn ermittelt wurde und gibt die Anzahl der Zahlen, die sich mit der Bingo-Karte decken zurück. un werden die Matchings gespeichert.

Sind alle Simulationen durchlaufen, wird ein Systemton erzeugt, es werden die csv-Files geschrieben, die Konsole erzeugt ein paar Informationen und beendet den Durchgang mit dem Ablöschen der statischen Variablen. Der nächste Durchgang beginnt.



```
bingo > src > CSimulation > main
Project
Run: CSimulation x
"C:\Program Files\AdoptOpenJDK\jdk-11.0.8.10-hotspot\bin\java.exe" ...
Start simulations with 15 drawings per game...
Playing Bingo 0% | 22405/100000000 (0:00:08 / 0:59:25)
```