Daniel Pantjuskin-Moos
108013248222
Vincent König
108011232630
Gruppe: D

Abgabe PHYSEC 3

Messungen

Implementierung Pearson Correlation

Die Aufgabenstellung ist, die in Abbildung 1 abgebildete Formel in einem vorgegeben Python-Code als Funktion *correlation(X, Y)* zu implementieren, sodass es von einem mitgegeben Framework fehlerlos getestet werden kann. Die vollständige *exercise3.py* kann hier heruntergeladen werden.

$$\rho(x,y) = \frac{\sum_{i=0}^{n-1}(x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sqrt{\sum_{i=0}^{n-1}(x_i - \overline{x})^2 \cdot \sum_{i=0}^{n-1}(y_i - \overline{y})^2}}$$

Abbildung 1: Auszug aus dem Assignment

```python
import utils
import numpy

"""
Excersise 3:
Implement the Pearson correlation coefficient.
Do NOT use any given function for standard-deviation
or mean-value but implement them by yourself.

X, Y are given as lists.

Blockwise application is done outside so please use the
whole vectors at once.
"""


def correlation(X, Y):
    mean_x = numpy.mean(X) # Mittel Vektor X
    mean_y = numpy.mean(Y) # Mittel Vektor Y

    numerator = 0 # zaehler
    denominator_x = 0 # der X beinhaltene Faktor im Nenner
    denominator_y = 0 # der Y beinhaltene Faktor im Nenner

    # Falls Vektoren unterschiedlich lang sind
    if len(X) != len(Y):
        raise Exception("Length not equal!\n")

    for i in range(len(X)):
        numerator += (X[i] - mean_x)*(Y[i] - mean_y)
        denominator_x += (X[i]-mean_x)*(X[i]-mean_x)
        denominator_y += (Y[i]-mean_y)*(Y[i]-mean_y)

    denominator = numpy.sqrt(denominator_x * denominator_y)

    # Falls Nenner=0, dann wird der jeweilige Eintrag ignoriert
    # und wird somit auch nicht in die Skizzen eingetragen
    if denominator == 0:
        return float('nan')

    pearson = numerator/denominator

    return pearson
    # return utils.not_yet_implemented("Correlation")
```

Auswertung

Die Messproben wurden mit 74 verschiedenen Tests analysiert. Die Testergebnisse und die dazugehörigen Befehle können hier heruntergeladen werden. Zuerst beschreiben mit römischen Zahlen nummerierte Abschnitte, welche Auffälligkeiten die dazugehörigen Teilaufgaben aus Ausufgabe 1 aufweisen. Danach werden die allgemeinen Fragen aus der Aufgabenstellung zu Aufgabe 3 beantwortet.

Hinweis zu den Tabellen

Alle Fälle werden ignoriert, bei denen die Nenner bei der Berechnung der Pearson Korrelationen gleich Null sind. Bei der Bestimmung der Mittel und Mediane wurden von allen Korrelationen die Beträge gebildet. Es wird auf vier Nachkommastellen gerundet.

i)

Alice sendet an Bob :

Teilaufgabe 1: $A \rightarrow B$				
Blockgröße	Mittel Bob	Median Bob	Mittel Eve	Median Eve
30	0.2290	0.1756	0.1798	0.1428
100	0.2219	0.1848	0.1341	0.1053
200	0.2401	0.2377	0.0946	0.0748
250	0.2591	0.2328	0.1495	0.1260
300	0.2791	0.2108	0.1784	0.1130

Es fällt auf, dass mit zunehmender Blockgröße weniger Nullen in den Nennern auftreten. Bob hat wie zu erwarten, durchgehend größere Korrelationen. Außerdem steigen Bobs Korrelationen in dieser Messung mit der Blockgröße bis mindestens 300. Eves Korrelationen weisen bei den verwendeten Blockgrößen eine negativen Buckel auf, mit Tiefpunkt bei Blockgröße 200. Interessant ist auch, dass bei Blockgröße 250 (und 300), der erste Eintrag deutlich höher liegt als die anderen, siehe Abbildungen 2 bis 5. Außerdem zeichnet sich der Trend ab, dass Eve deutlich mehr negative Korrelationen hat.

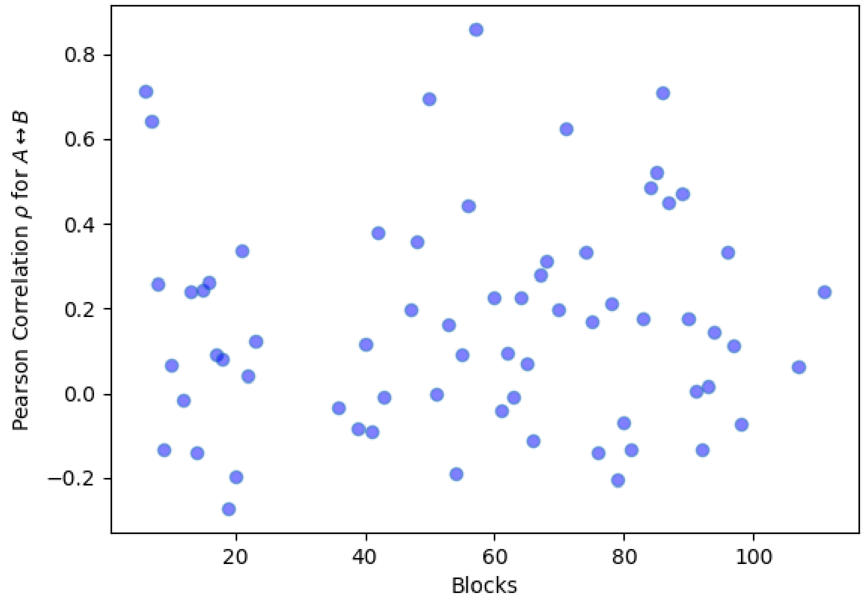

Abbildung 2: Skizze der Korrelation zwischen A und B mit Blockgröße 30

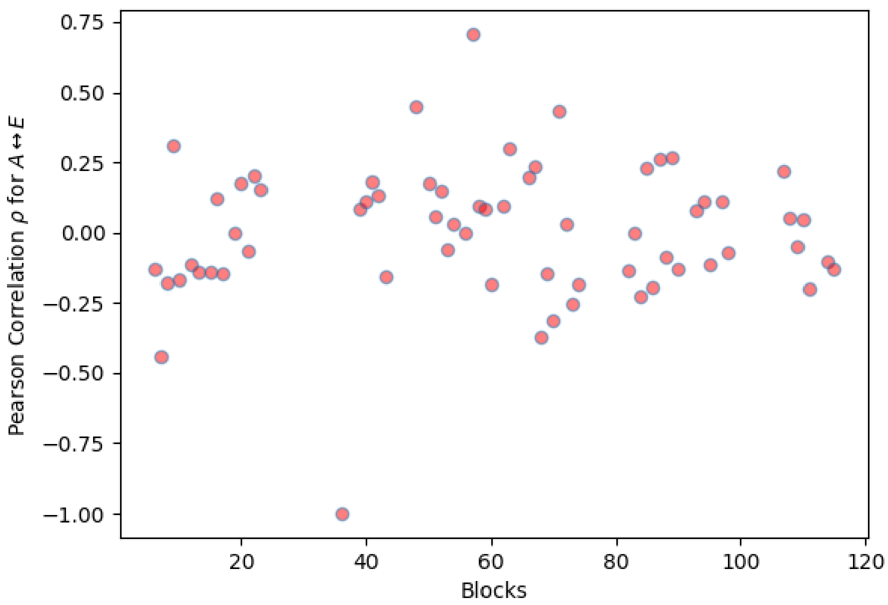

Abbildung 3: Skizze der Korrelation zwischen A und E mit Blockgröße 30

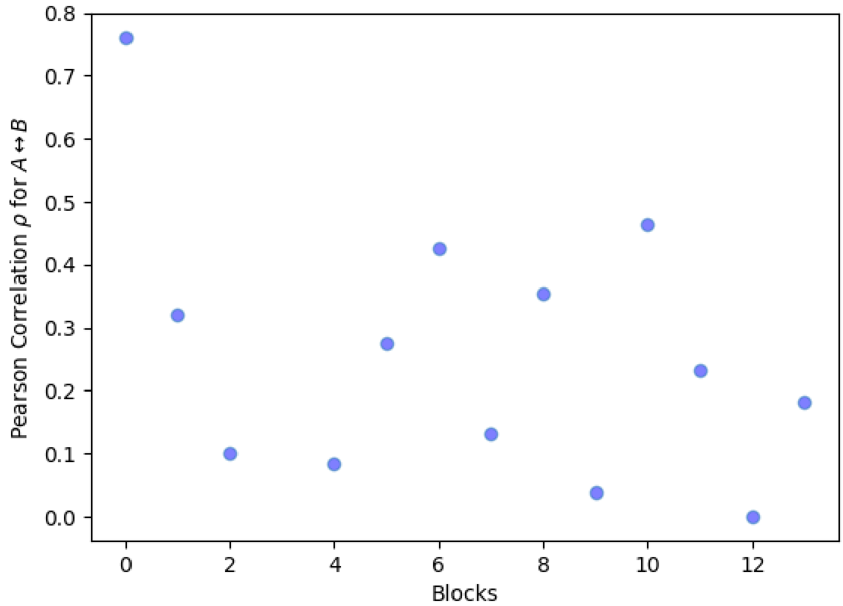

Abbildung 4: Skizze der Korrelation zwischen A und B mit Blockgröße 250

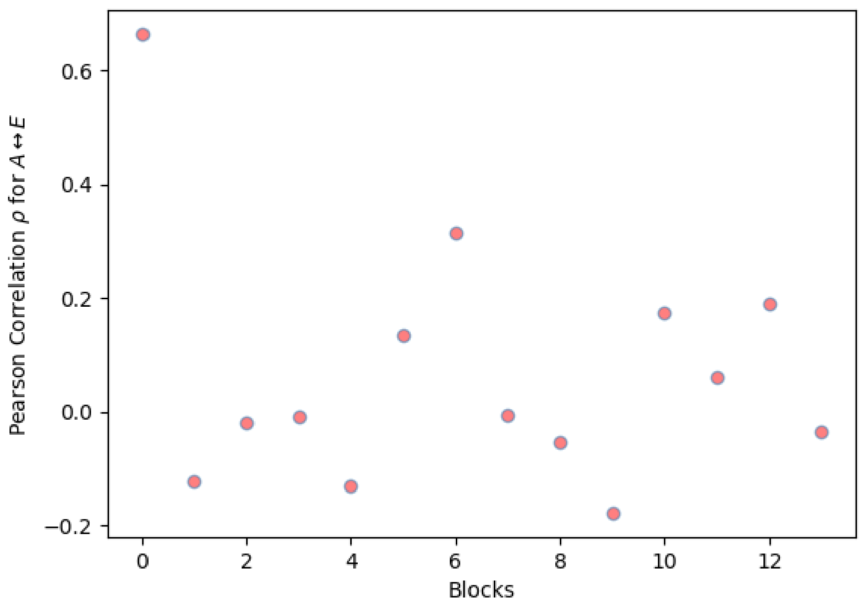

Abbildung 5: Skizze der Korrelation zwischen A und E mit Blockgröße 250

ii)

Alice sendet an Bob, aber diesmal wird für Bewegung zwischen den Knoten gesorgt:

Teilaufgabe 1: $A \rightarrow B$ mit Bewegung zischen den Knoten				
Blockgröße	Mittel Bob	Median Bob	Mittel Eve	Median Eve
30	0.9807	0.9901	0.2522	0.2626
200	0.9912	0.9925	0.2365	0.2337
300	0.9918	0.9932	0.2339	0.2495

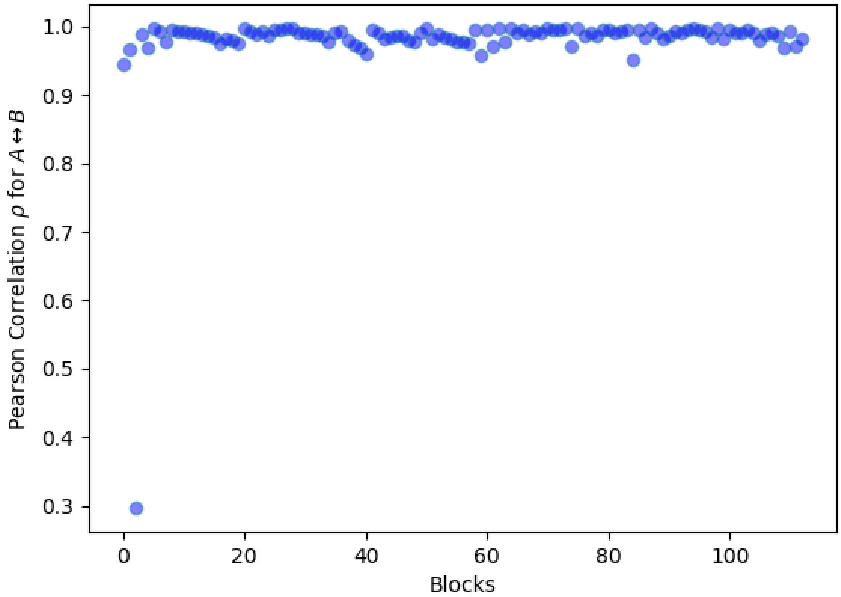

Abbildung 6: Skizze der Korrelation zwischen A und B mit Blockgröße 30 (mit Bewegung)

Auffallend ist , dass der Unterschied nun ohne Probleme mit dem bloßen Auge klar zu erkennen ist. Bobs Korrelationen sind fast durchgehend der eins sehr nahe, wobei Eves Korrelationen besser sind als zuvor und zunehmend mit der Blockgröße seltener im negativen Bereich, aber dennoch weit abgeschlagen hinter Bobs Korrelationen.

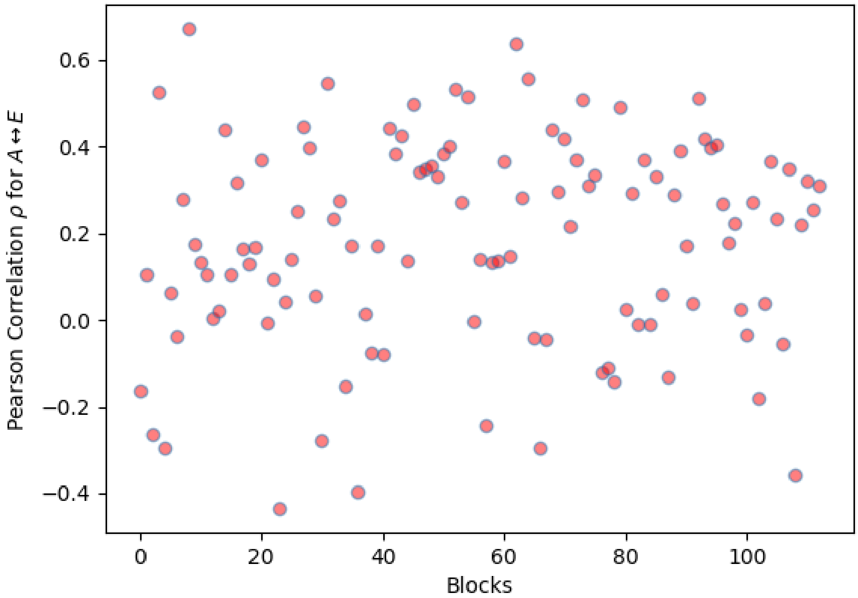

Abbildung 7: Skizze der Korrelation zwischen A und E mit Blockgröße 30 (mit Bewegung)

iii): Keine Bewegung

Es treten sehr oft nullen in den Nennern auf die dann mit *nan* gekennzeichnet wurden. Wie zu erwarten war, gab es mit Abstand etwa 20cm deutlich mehr von diesen Fällen. Bei Bob war sogar bei 20cm jedes einzelne Ergebnis wie zum Beispiel in Abbildung 8 erkennbar *nan* was zur Folge hat, dass diese skizze leer ist. Umgekehrt gibt es bei plus 10m Abstand deutlich weniger *nan*-Fälle. Bob hat da keine einzige leere Skizze. Insgesamt ist die Korrelation bei Bob und Eve klar höher. Es lässt sich beobachten, dass das Mittel bei Eve +10m bei Blockgröße 30 mit 0.1686 deutlich größer ist als bei Blockgröße 300 mit 0.0467.

Keine Bewgung (20cm)

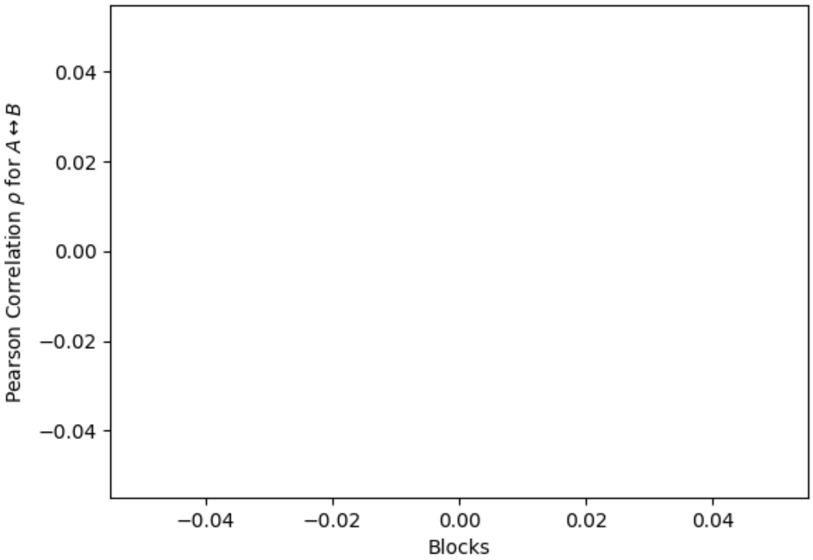

Abbildung 8: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (20cm)

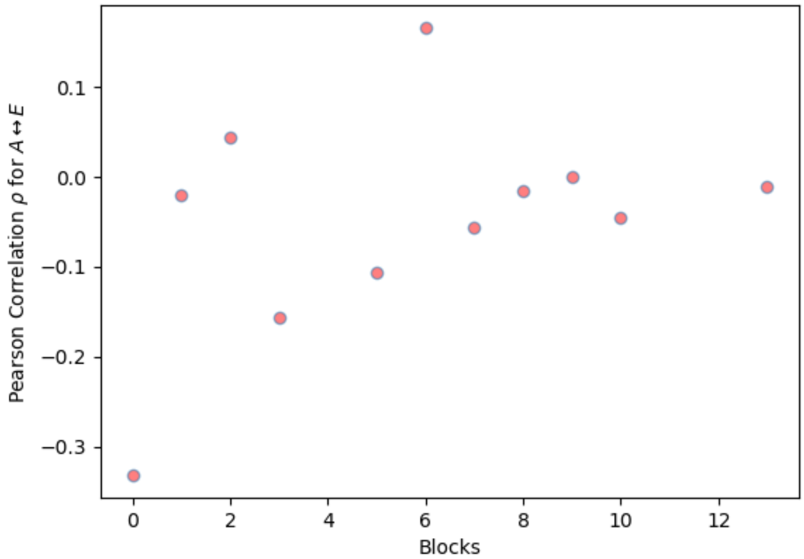

Abbildung 9: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (20cm)

Keine Bewgung (+10m)

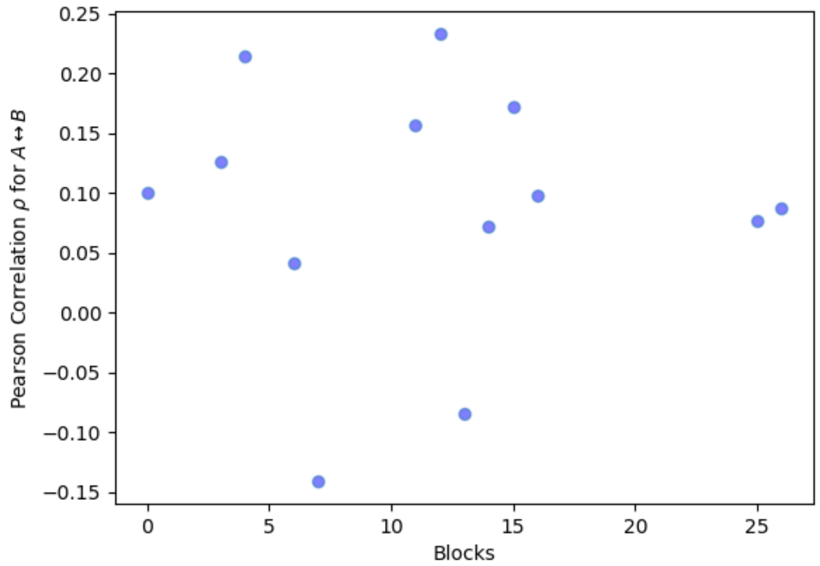

Abbildung 10: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (+10m)

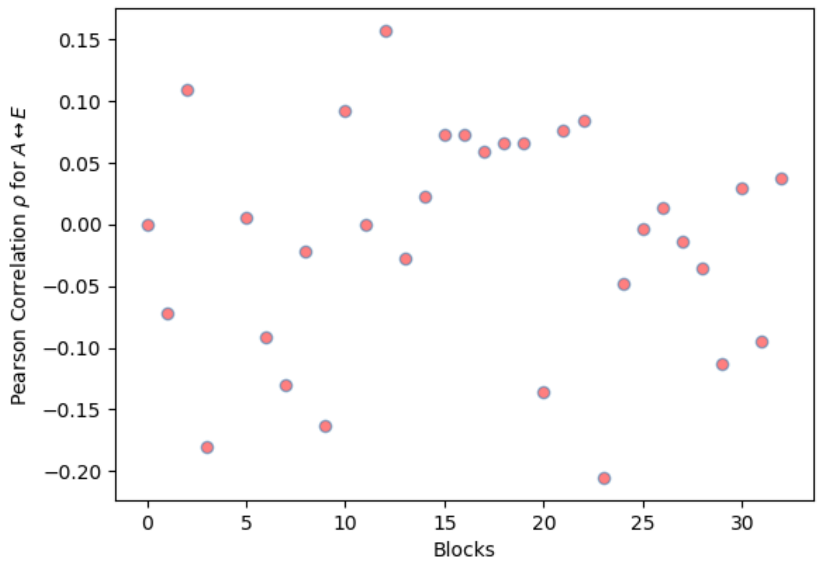

Abbildung 11: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (+10m)

iii): Mit Bewegung

Wie zu erwarten war, sind hier die Koorelationen im Vergleich zu den Abbildungen 8 bis 11 deutlich größer. Die Korrelation bei Bob ist bei +10m ein bisschen kleiner, bei Eve jedoch ist die Korrelation bei +10m sehr deutlich größer. Reichten die Korrelationen bei Eve bei 20cm von etwa -0.4 bis 0.3 so reichen sie bei +10m von etwa 0.3 bis 0.8 siehe Abbildungen 13 und 15

Mit Bewgung (20cm)

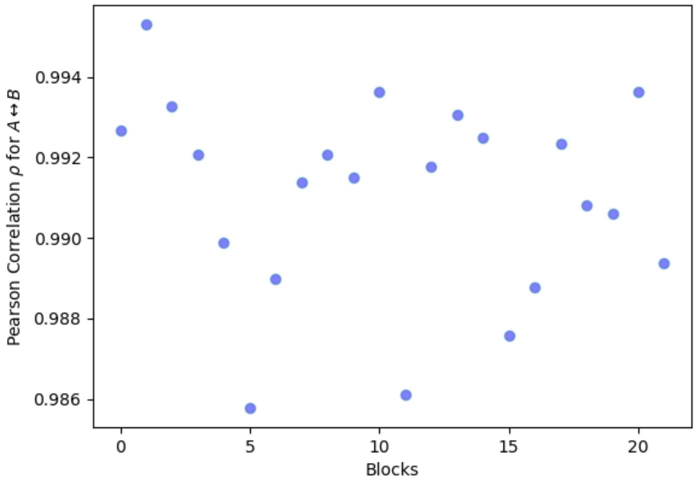

Abbildung 12: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (20cm) mit Bewegung

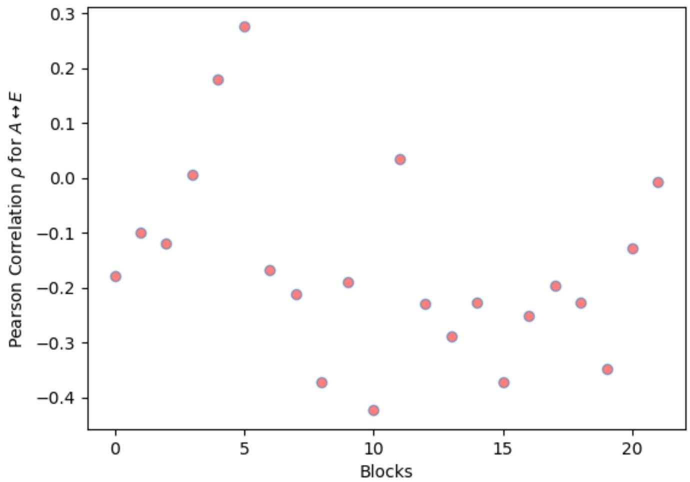

Abbildung 13: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (20cm) mit Bewegung

Mit Bewgung (+10m)

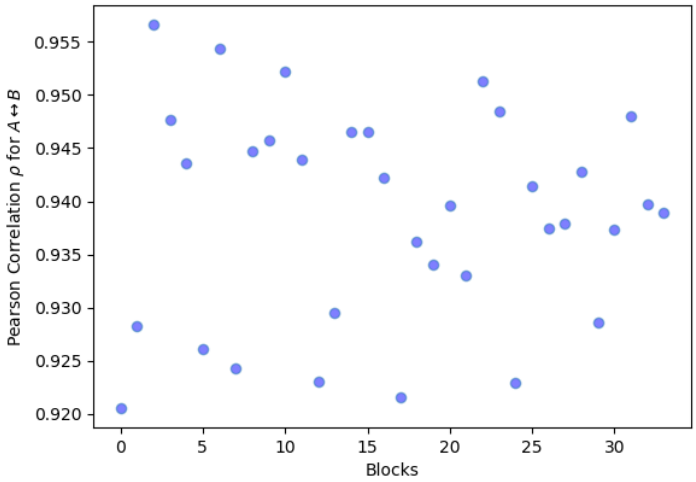

Abbildung 14: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (+10m) mit Bewegung

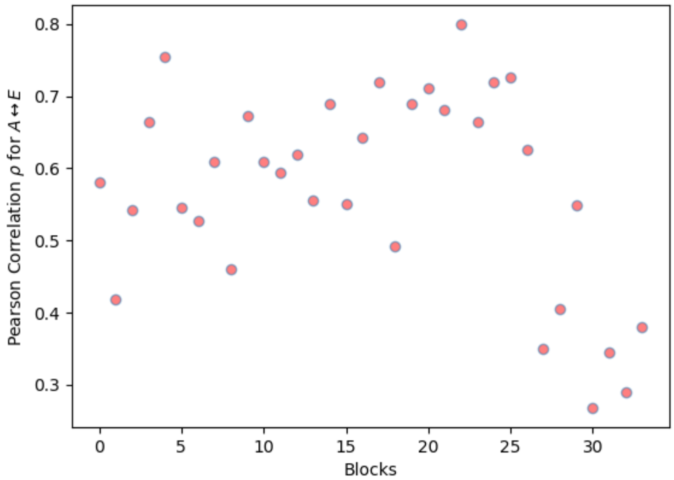

Abbildung 15: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (+10m) mit Bewegung

Quantisierer Jana Multibit

a) Funktionsweise

Alice und Bob sammeln RSS Messungen und bestimmen den Bereich der gemessenen Werte(Range of RSS). Anschließend wird *N*, eine Nummer, wieviele Bits pro Messung extrahiert werden können, bestimmt. Danach werden die Messungen in $M = 2^N$ gleich große Intervalle unterteilt. Für jedes dieser Intervalle wird eine N-Bit Zuordnung gewählt. Liegt die Messung nun im Intervall, so wird sie dem *Bitstream* hinzugefügt. Andernfalls muss sie erst korrigiert werden.

b) Pseudocode

Algorithm 1 Pseudocode

$range = max[RSS] - min[RSS]$
$N \in [0, log_2 RSS]$
$M = 2^N$
$RSS[] \rightarrow M$ intervalls $I[]$ of equal size
Choose N bit assignment $\forall$ M intervalls
for $t \rightarrow len(RSS[])$ **do**
    for $i \rightarrow len(I[]$ **do**
        if $RSS[t] \in I[i]$ **then**
            $bitstream \leftarrow$ bit assignment
        end if
    end for
end for
return $bitstream$

Quantisierer Mathur Suhas

a) Funktionsweise

Bob und Alice haben beide die gleiche Anzahl an Schätzwerten h_a und h_b.
$h_a(j)$ sowie $h_b(j)$ korrespondieren jeweils $\forall j \in [1, len(h_a)]$.
Alice wählt zufällige m-elementige Teilmenge die unter q_- oder über q_+ liegt und speichert diese Messungen in einer Liste L ab. Diese Liste wird nun an Bob gesendet.
Bob überprüft nun für jeden Eintrag der Liste, ob seine Messungen h_b korrekt sind. Jeder Eintrag der nicht übereinstimmt wird in eine Liste L' eingetragen. Diese Liste wird dann an Alice zurückgesendet.
Nun können beide mit Hilfe der Liste L' $Q(h_a)$ bzw. $Q(h_+)$ berechnen.

b) Pseudecode

Algorithm 2 Pseudocode

```
Alice creates List L
Alice sends L → Bob
Bob compares L to his List
Bob creates List L'
Bob sends L' → Alice
Alice and Bobo generate mutual Bitstream
```

Bonus: Reading Assignment

a)

Die Autoren sind auf der Suche nach nach einem effizienten und sicheren Algorithmus. Dieser soll den kabellosen Kanal möglichst effizient nutzen und soll Nutzerfreundlich sein, indem er z.b. einen sicheren Schlüssel in kurzer Zeit generieren kann.

b)

Es ist eminent, dass der Probing-Algorithmus sich den Kanaleigenschaften anpassen kann, da der *probing process* ansonsten nicht effizient sein kann, wenn der Kanal sich nicht ändert. Das kann bspw. passieren, wenn keine Informationen oder redundaten Informationen immer wieder gesendet werden.