

Praktikum 2: Grundlagen Stochastik

Aufgaben Vorlesungs-Skript

Bearbeiten Sie die folgenden Aufgaben des Vorlesungs-Skripts:

Aufgabe 2.1 (Würfeln), 2.2 (Beweis Satz 2.2.3), 2.3 (Bayessche Formel: medizinischer Test), 2.4 (Bayessche Formel: Werkstücke), 2.5 (Erwartungswert, Varianz Würfel und Summe zweier), 2.6 (Varianz Normalverteilung), 2.7 (IQ).

Einige Ihrer Lösungen können Sie anhand der folgenden Programmieraufgaben überprüfen.

Aufgaben Programmieren

Laden Sie sich die Datei `P2vorgabe.zip` herunter und benutzen Sie das darin enthaltene Programm `main2.py`, als Grundlage für die folgenden Programmieraufgaben. Alle zu programmierenden Funktionen sind dort vorgegeben, und enthalten teilweise Hinweise zur Umsetzung. `main2.py` ist so lauffähig, aber liefert natürlich noch falsche Ergebnisse und Plots; es werden auch schon die nötigen Bibliotheken geladen, die Sie benutzen dürfen.

Die Programmieraufgaben gliedern sich in die folgenden vier Themen:

- **Bayessche Formel**

- `Bayes_Formel(P_B_A, P_A, P_B)`
- `totale_Wahrscheinlichkeit(P_B_Ai, P_Ai)`
- `Test(Korrektheit, a_priori_Wahrscheinlichkeit)`

Definieren Sie die Formeln wie in der Vorlesung; dabei sind die Argumente *floats* oder *iterables*. `Test` soll ein Paar, also ein 2-Tupel, bestehend aus der totalen Wahrscheinlichkeit und der a-posteriori-Wahrscheinlichkeit zurückgeben. Benutzen Sie nun die Funktionen, um diese beiden Wahrscheinlichkeiten in Abhängigkeit von der a-priori-Wahrscheinlichkeit zu plotten; siehe `main2.py` für Vorschläge zu den Plot-Befehlen.

- **Verteilungen**

- `normal_10_3(rng, n)`
- `uniform_100_10(rng, n)`
- `mixed_normal(rng, n, M, S, P)`

Lesen Sie sich in die Bibliothek `numpy.random` ein. In `main2.py` ist das Objekt `rnd` der Klasse `Generator` definiert, mit dessen Hilfe sich Zufallszahlen von einer Vielzahl von Verteilungen generieren lassen. Definieren Sie hiermit obige Funktionen die Arrays der Größe `n` von Zufallszahlen (a) der Normalverteilung $N(10, 3^2)$, (b) der Gleichverteilung mit Erwartungswert 100 und Standardabweichung 10, sowie (c) einer sogenannten *gemischten Gaußverteilung*. Dabei sind `M`, `S` und `P` *iterables* der gleichen Länge `k`, die Erwartungswerte μ_i , Standardabweichungen σ_i bzw. Wahrscheinlichkeiten p_i mit $\sum_{i=1}^k p_i = 1$ enthalten, und die `n` Samples des Rückgabe-Arrays sollen nach folgender Regel gezogen werden: Zunächst wird ein $i \in \{1, 2, \dots, k\}$ mit Wahrscheinlichkeit p_i gewählt und dann das Sample gemäß $N(\mu_i, \sigma_i^2)$ gezogen. In `main2.py` sind Beispiele, wie diese Funktion benutzt wird. *Vermeiden Sie bei der Implementierung Schleifen, sondern nutzen Sie die Geschwindigkeitsvorteile von numpy. Vielleicht hilft Ihnen die Funktion `rng.multinomial` aus der Klasse `Generator` von `numpy.random`.* Plotten Sie zur Kontrolle wie in `main2.py` angegeben Histogramme der Verteilungen.

- **Gesetz der großen Zahlen (GdGZ)**

- `rel_variance_of_mean(rng, distr, no_samples, no_runs)`

Die Funktion soll für eine vorgegebene Verteilung (`distr`) eine Stichprobe der Größe `no_samples` ziehen und hiervon den Mittelwert berechnen. Von dieser Mittelwertberechnung soll die Varianz bestimmt werden. Dazu werden viele solche Stichproben gezogen (nämlich so viele wie `no_runs` angibt) und von all diesen Mittelwerten die Varianz gebildet. Rückgabewert der Funktion ist der Quotient gebildet aus dieser Varianz und der Varianz der Gesamtstichprobe der Größe `no_samples * no_runs`. Hier ist es zweckmäßig die Daten in einer Matrix anzuordnen, siehe `main2.py`. *Und auch hier und überhaupt gilt: Vermeiden Sie bei der Implementierung Schleifen, sondern nutzen Sie die Geschwindigkeitsvorteile von numpy.*

Ziel soll es hier sein, das Gesetz der großen Zahlen zu verstehen. Wählen Sie einen geeigneten Bereich von Stichprobengrößen und plotten Sie obige relative Varianz in Abhängigkeit von der Stichprobengröße für verschiedene Verteilungen in das gleiche Bild (siehe `main2.py`). Was ergeben sich für Graphen? Wie erklärt das das Gesetz der großen Zahlen?

- **Zentraler Grenzwertsatz (ZGWS)**

- `centralized_sample(rng, distr, no_samples, no_runs)`

Erstellen Sie ganz analog zur vorigen Aufgabe eine Matrix deren Spalten Stichproben der Größe `no_samples` sind, und berechnen Sie dann wie in der Vorlesung die *standardisierte zentralisierte Zufallsvariable* für jede Spalte; das Ergebnis ist dann eine Stichprobe der Größe `no_runs` von solchen Zufallsvariablen, die von der Funktion zurückgegeben werden soll.

Mit dem Code in `main2.py` können Sie ein Histogramm dieser standardisierten Zufallsvariable aus dem Gesetz der großen Zahlen erstellen. Was erkennt man an den Bildern?

Abgabe

- Laden Sie das Archiv `P2vorgabe.zip` von [moodle](#) herunter, entpacken Sie es, und testen Sie Ihre Programme, indem Sie `test2.py` im gleichen Verzeichnis mit `python` ausführen. Erhalten Sie `ERROR`, so entspricht Ihr Programm nicht der Spezifikation von oben. Erhalten Sie `FAIL`, so ist Ihr Programm zwar lauffähig, aber die berechneten Werte sind fehlerhaft.
- **Abgaben, bei denen der Test gar nicht durchläuft oder mit `ERROR`, werden nicht akzeptiert**, `FAIL` führt nur zu Punktabzug.
- Sie finden in obigem Archiv auch die Datei `info2.md` mit anzugebenden Informationem zu Ihrem Team und Ihrer Abgabe, **bitte füllen Sie diese nach dortiger Anleitung aus, und vergessen Sie nicht die Quellenangabe**. Abgaben mit unvollständiger Datei `info2.md` können nicht gewertet werden. **Belassen Sie diese Datei in der UTF-8-Kodierung**.
- Komprimieren und bündeln Sie alle oben erzeugten oder geänderten Dateien, indem Sie ein ZIP-Archiv erstellen. Sollten Sie nicht wissen, wie das geht, konsultieren Sie dazu die Dokumentation Ihres Betriebssystems.
- Nennen Sie Ihr ZIP-Archiv `P2.zip`.
- Schreiben Sie eine Email an rosehr@hm.edu mit Betreff **Abgabe Datenanalyse**, Dateianhang **P2.zip** und irgendwelchem sonstigen Inhalt. **Der Automat akzeptiert die Abgabe nur, wenn diese Angaben (Betreff, Dateiname), sowie Ihre persönlichen Angaben (Nach-, Vor- und Teamname) korrekt sind**. Sie erhalten innerhalb der nächsten 24 Stunden (meistens aber deutlich schneller) eine Bestätigungsemail, wenn alles korrekt war.

Praktikumstermine P2: 08.05. (Gruppe a) und 15.05. (Gruppe b), **Abgabetermin:** 17.05.