Humboldt-Universität zu Berlin
Bereich Stochastik und Finanzmathematik

Peter Frentrup

Stochastik-Praktikum
Wintersemester 2017/2018

Projektaufgaben Block 2

Aufgabe 1 (Säulen- und Tortendiagramme)

a) Schreiben Sie eine Funktion `stackedBarPlot(data)`, die aus einem $n \times m$-Array `data` ein Säulendiagramm wie in Abb. 1 erstellt (dort für $n = 6$, $m = 3$). Die Einträge von `data` entsprechen den Höhen der einzelnen Säulenabschnitte (z.B. ist `data[5, 2]` die Höhe des roten Abschnitts der letzten Säule in Abb. 1, links).

Benutzen Sie dazu die `matplotlib.pyplot.bar()`-Funktion.

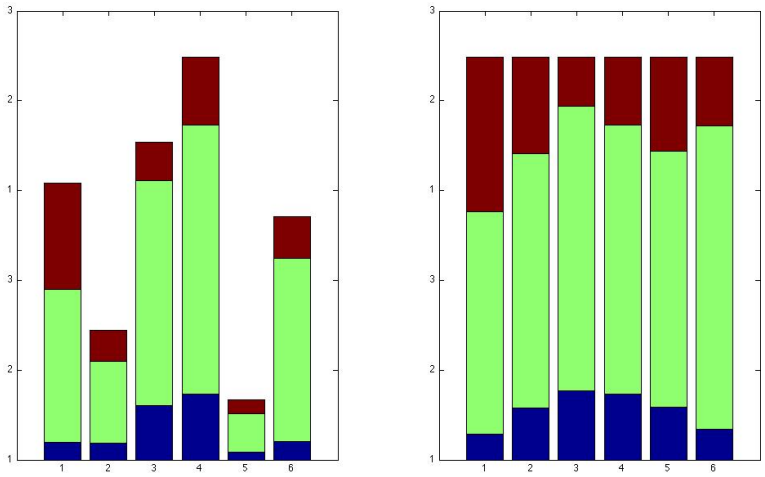

Abbildung 1: gestapelte Säulendiagramme, links unnormiert, rechts normiert

b) Fügen Sie einen optionalen Parameter `colors` für die Angabe einer Liste von Farben hinzu, z.B. `stackedBarPlot(data, colors=['b', 'g', 'r'])`. Die angegebenen Farben sollen ggf. wiederholt werden (falls 1 <= **len**(colors) < n).

Nutzen Sie standardmäßig (**if** colors **is** None: ...) die Farben der Palette

```
matplotlib.pyplot.rcParams['axes.prop_cycle'].by_key()['color']
```

c) Statten Sie Ihre Funktion `barPlot` mit einem hilfreichen „docstring“ aus:

```
def stackedBarPlot(data, colors=None):
    """
    Beschreibung der Funktionalitaet und Parameter...
    """
    ...
```

Orientieren Sie sich dabei an **print**(matplotlib.pyplot.bar.__doc__) (insbesondere Formatierung), sodass ihr Hilfetext z.B. im interaktiven Hilfefenster von Pyzo ähnlich dargestellt wird.

d) Betrachten Sie den Datensatz `miete03p.csv`. Laden Sie die Datei über den relativen Pfad „../code/miete03p.csv“. Nutzen Sie Ihre Funktion `stackedBarPlot`, um die relative Verteilung der drei Wohnungsgrößeklassen (`QMKL`) in den sechs einzelnen Baujahrklassen (`BJKL`) zu veranschaulichen (siehe `inspect-miete.py` für die Definition von `QMKL` und `BJKL`).

Das Ergebnis müsste ähnlich wie Abb. 1 (rechts) aussehen.

e) Er zeugen Sie ein Torten- und Balkendiagramm der Anzahl der Wohnungen in den einzelnen Baujahren (vor 1919, 1919-1948, 1949-1965, 1966-1977, 1978-1983, Neubau), ähnlich wie in Abb. 2. Achten Sie auf die Beschriftung. Beide Diagramme sollen nebeneinander in einem Fenster erscheinen. Nutzen Sie `matplotlib.pyplot.pie()` und entweder Ihre `stackedBarPlot()`-Funktion oder direkt `matplotlib.pyplot.bar()`.

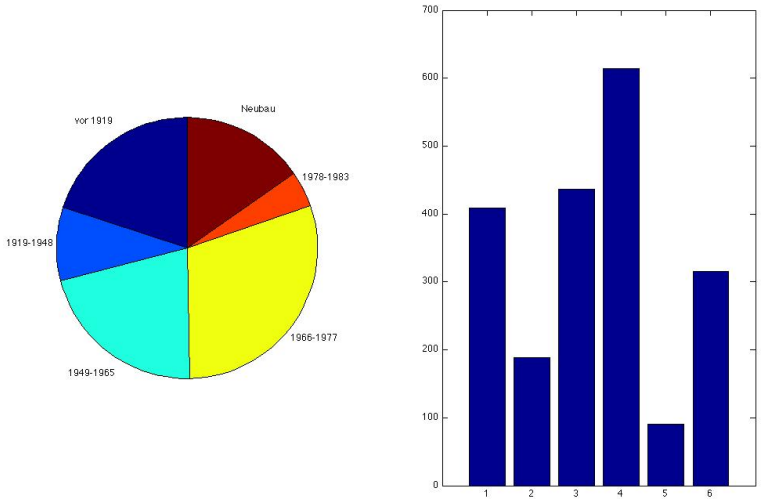

Abbildung 2: Torten- und Säulendiagramm eines Datensatzes

Aufgabe 2 („Methusalems machen Kasse“)
Im Rahmen einer Studie wurde eine Befragung in Unternehmen über die Einstiegsgehälter von Berufsanfängern durchgeführt. Bei der Befragung von 33 Unternehmen wurde das durchschnittliche Einstiegsgehalt in Bezug zur Studiendauer von Akademikern dokumentiert. Es ergaben sich die folgenden Werte:

Studiendauer	6	7	7	8	6	9	8	10	12	9	10
Durchschnittseinstiegsgehalt	30	27	26	21	31	20	27	18	20	23	21
Studiendauer	8	9	12	13	11	10	14	13	14	10	9
Durchschnittseinstiegsgehalt	37	36	28	26	33	35	31	23	28	37	39
Studiendauer	13	13	15	14	17	15	18	16	15	16	17
Durchschnittseinstiegsgehalt	42	41	38	40	37	36	30	35	35	33	36

Die Studiendauer ist in Semestern und das Einstiegsgehalt in 1000 Euro angegeben.

a) Definieren Sie 2 Vektoren mit jeweils den 33 angegebenen Werten für Studiendauer und Einstiegsgehalt. Erzeugen Sie eine 33×2-Matrix aus den beiden Vektoren.

b) Bestimmen Sie die statistischen Kenngrößen Mittelwert, Median, Standardabweichung, Varianz, Minimum, Maximum und die 25- und 75-Prozent-Quantile der Stichprobe für beide Variablen.

c) Definieren Sie eine Variable, die die Einstiegsgehälter in 3 Klassen ($\leq$30, 30-35, >35) einteilt. Definieren Sie außerdem eine Variable, die die Studiendauer in 4 Klassen $\leq$8, 8-10, 10-14, >14) einteilt.

d) Stellen Sie den Zusammenhang von Einstiegsgehalt und Studiendauer anhand der diskreten Einteilungen in Klassen in einem gestapelten Säulendiagramm dar und interpretieren Sie diesen.

   Nutzen Sie hierzu Ihre Funktion `stackedBarPlot` aus Aufgabe 1.

e) Veranschaulichen Sie die relative Häufigkeitsverteilung der 3 Einstiegsgehälter-Klassen in einem Tortendiagramm.

f) Erstellen Sie eine Grafik, in der die Werte der Einstiegsgehälter gegen die Studiendauern aufgetragen sind. Zeichnen Sie eine Regressionsgerade ein. Analysieren Sie das Ergebnis.

Eine ähnliche Studie wie diese (erfundene) betitelte das Handelsblatt mit der Überschrift „Methusalems machen Kasse: Ein langes Studium zahlt sich in barer Münze aus“; allerdings wurde bei dieser Schlussfolgerung eine Einteilung nach Studienfach einfach nicht berücksichtigt.

Die Datei `meth.dat` enthält den vollständigen Datensatz für die Aufgabe mit der zusätzlichen Variablen des Studienfaches: Physik(PH), Chemie(CH) oder Betriebswirtschaftslehre(BWL).

g) Lesen Sie die (gleichen) Daten als Datei diesmal direkt über Python ein (über den relativen Pfad „../code/meth.dat“) und machen Sie sich mit den Eigenschaften und Bezeichnungen der Datei vertraut.
   Nutzen Sie die Funktion `numpy.genfromtxt()` mit geeigneten Parametern `converters=...` oder `dtype=...`.

h) Berechnen Sie (zur Kontrolle nochmals) die statistischen Kenngrößen beider Variablen.

i) Plotten Sie einen Boxplot für beide Variablen.

j) Erstellen Sie denselben Graphen wie oben mit Regressionsgerade für den gesamten Datensatz noch einmal.

k) Kennzeichnen sie die Punkte in dem Graphen dabei farblich unterschiedlich nach Studienfächern.

l) Zeichnen Sie in demselben Graphen zusätzlich Regressionsgeraden für die Teilmengen getrennt nach Studienfächern ein in unterschiedlichen Farben die mittels einer Legende bezeichnet werden. Analysieren sie das Resultat.

Aufgabe 3 (Gauß'sche Prozesse – Simulieren zufälliger Funktionen)
Lesen Sie [RW06, Abschnitt 2.1 und 2.2] (sowie ggf. Kapitel 1) und lösen Sie [RW06, Aufgabe 2.9.1].

Literatur

[RW06] Rasmussen, Carl Edward und Christopher K. I. Williams: *Gaussian processes for machine learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006, ISBN 978-0-262-18253-9. Online verfügbar unter `http://www.gaussianprocess.org/gpml`.

Hinweise zur Abgabe:
Alle Dateien (PDF der Auswertung, Python-Code für jede einzelne Aufgabe) gepackt als Zip-Archiv mit dem Namen *UE1_Student1_Student2.zip* bis zum 28. 11. per E-Mail an *frentrup@math.huberlin.de* schicken.