

1. Bringen Sie für das Folgende **R**s in §2.3 kennen gelernte vektorielle Arithmetik und Funktionen zum Einsatz! (Mehr Funktionalität brauchen Sie hier nicht.)

a) Wählen Sie ein (zweistelliges) $N \in \mathbb{N}$ und lassen Sie für $i = 0, 1, 2, \dots, N$ die Werte $\frac{1}{2^i}$ sowie für $n = 0, 1, 2, \dots, N$ die Werte $S_n := \sum_{i=0}^n \frac{1}{2^i}$ berechnen!

b) Wählen Sie ein (kleines zweistelliges) $n \in \mathbb{N}$ sowie ein $p \in (0, 1)$ und lassen Sie zunächst für $i = 0, 1, 2, \dots, n$ die Werte $W_i := \binom{n}{i} p^i (1-p)^{n-i}$ und dann für $k = 0, 1, 2, \dots, n$ die Werte $P_k := \sum_{i=0}^k W_i$ berechnen!

2. Die folgende Tabelle enthält $n = 14$ bivariate Daten (x_i, y_i) , die Sie zunächst („von Hand“) in **R** erfassen müssen:

x_i	0.61	0.79	0.83	0.66	0.94	0.78	0.81	0.60	0.88	0.90	0.75	0.86	0.74	0.84
y_i	1.01	0.13	1.73	0.67	0.56	0.99	0.49	0.72	0.13	0.48	0.98	0.08	1.23	0.87

- a) Bestimmen Sie den Vektor der Ordnungsstatistiken $(x_{1:n}, \dots, x_{n:n})$ der x_i !
- b) Erzeugen Sie einen **character**-Vektor, der als seine Elemente die insgesamt 14 Zeichenketten `"x(1:14) = 0.6"`, `"x(2:14) = 0.61"` usw. enthält (also alle Paare der Art Ordnungsstatistik = Wert)!
- c) Bilden Sie den Vektor der Konkomitanten $(y_{[1:n]}, \dots, y_{[n:n]})$ der y_i zu den $x_{i:n}$!
- d) Konstruieren Sie den **character**-Vektor, der alle obigen Wertepaare nach den x -Werten geordnet in der Form `"(x(1:14), y[1:14]) = (0.6, 0.72)"`, `"(x(2:14), y[2:14]) = (0.61, 1.01)"` usw. enthält!
- e) Modifizieren Sie Teil d unter Verwendung der Funktion **format** derart, dass die Ausgabe „schön“ formatiert ist, also z. B. `"0.6"` als `"0.60"`, damit es zu `"0.72"` passt, und `"1:14"` als `" 1:14"`, damit es zu `"10:14"` passt!
3. Der in **R** eingebaute Datensatz **faithful** beinhaltet Wartezeiten in Minuten auf Eruptionen und Dauern in Minuten dieser Eruptionen des “Old Faithful Geyser” im Yellowstone National Park/Wyoming, USA. Mit den Ausdrücken

`faithful$waiting` und `faithful$eruptions`

erhalten Sie die jeweiligen Datensätze als Vektoren. (Die Bedeutung des **\$**-Zeichens wird im Kurs später noch erläutert.)

- a) Wie groß ist die kleinste, die größte und die mittlere Wartezeit und wie lautet die Standardabweichung der Wartezeiten?
 - b) Die wievielte Beobachtung ist die kleinste bzw. die größte Wartezeit? (Gibt es denn jeweils nur eine davon bzw. was könnten Sie tun, wenn es mehrere gäbe und Sie alle finden wollen?)
 - c) Über welcher (größten) Schwelle liegen die 30 % längsten Wartezeiten?
 - d) Kurze Wartezeiten seien solche, die höchstens 60 Minuten betragen. Extrahieren Sie die kurzen Wartezeiten!
 - e) Wie viele Wartezeiten sind es insgesamt und wie viele kurze Wartezeiten gibt es?
 - f) Ermitteln Sie die zu den kurzen Wartezeiten gehörenden Eruptionsdauern!
 - g) Wie groß ist die mittlere kurze Wartezeit? Wie lautet der Mittelwert der zugehörigen Eruptionsdauern?
 - h) Wie viele Eruptionsdauern nach kurzer Wartezeit sind mindestens 2.5 Minuten lang?
 - i) Lassen Sie sich durch den Befehl `plot(faithful$waiting, faithful$eruptions)` ein Streudiagramm der Eruptionsdauern gegen die Wartezeiten zeichnen! Versuchen Sie zu interpretieren, was es zeigt!
4. Laut V-Skript und **R**s Hilfeseite liefert die Funktion `duplicated` für einen beliebigen Vektor `v` einen logischen Vektor (derselben Länge wie `v`) zurück, der angibt, welche Elemente von `v` Duplikate von Elementen *mit kleinerem Index* in `v` sind. Mit anderen Worten: Wenn Sie einen Vektor links (bzw. mit dem ersten Element) beginnend durchlaufen, wird das „erstmalige“ Auftreten eines Elements, das „danach“ ein Duplikat ist, ignoriert. Vollziehen Sie dies mit einem Vektor wie dem im Beispiel der Vorlesung (`y <- c(1, 5, 1, 9, 5)`) oder einem längeren nach!
- (Dieses Verhalten passt übrigens genau zur Bedeutung des Wortes „Duplikat“.)

Versuchen Sie nun (ohne eine R-Funktion zu programmieren!) durch die Kombination von `duplicated` (evtl. nicht nur einmal verwendet) und einer logischen Verknüpfung sowie evtl. einer weiteren einfachen R-Funktion, *alle* Elemente eines beliebigen Vektors zu identifizieren, die mehrfach in ihm vorhanden sind! Das Ergebnis für einen Vektor `v` soll also ein logischer Vektor (derselben Länge wie `v`) sein, der angibt, welche Elemente von `v` Duplikate von Elementen *mit beliebigem Index* in `v` sind.