

1. (Ergänzung von Aufgabe 4 auf Blatt 4.)

- a) Die mit `apply` verwandten Funktionen `lapply` und `sapply` sind speziell für Listen zuständig, um eine Funktion auf die Komponenten einer Liste anzuwenden, aber ebenso auch für die Spalten eines Data Frames verwendbar. Finden Sie heraus, ob es einen und wenn ja welchen Unterschied es macht, wenn auf die Komponenten, sprich die Spalten des Data Frames `cu.summary` die Funktion `class` einerseits mit Hilfe von `apply` und andererseits mit `lapply` bzw. `sapply` angewendet wird! Führen Sie dies auch mit den Funktionen `length` und `summary` anstelle von `class` durch. Halten Sie die Verwendung von `apply` hier für sinnvoll?
- b) Klären Sie, was „inhaltlich“/strukturell passiert, wenn Sie `cu.summary` unter Verwendung der Funktion `as` als eine Matrix (unter einem anderen Namen) speichern wollen (siehe hierzu §2.11.1)!
- c) Machen Sie sich (nochmal?) mit der in §2.10.9 beschriebenen Funktion `with` vertraut und nutzen Sie sie für die nächsten drei Teilaufgaben!
- d) Ermitteln Sie zum einen unter Verwendung der Funktion `unique` und zum anderen der Funktion `levels`, welche verschiedenen Typen von Autos und welche Länder in `cu.summary` auftreten! Was ist der Unterschied zwischen den beiden Ergebnissen?
- e) Erstellen Sie die gemeinsame Häufigkeitstabelle (Kontingenztafel) der beiden Variablen `Country` und `Type` von `cu.summary`!
- f) Bestimmen Sie unter Verwendung der Funktion `tapply` für jeden Autotyp den mittleren Preis und die zugehörige Standardabweichung sowie schließlich den Variationskoeffizienten (= Standardabweichung relativ zum Mittelwert) der Preise eines jeden Autotyps!

2. Eignen Sie sich den Inhalt des Kapitels 3 „Import und Export von Daten bzw. ihre Ausgabe am Bildschirm“ im Vorlesungsskript an! (Er wird ab der nächsten Übungsaufgabe in variierendem Umfang benötigt.) Für diese Aufgabe brauchen Sie keine Bearbeitungen in Stud.IP hochzuladen.

3. Kopieren Sie sich die Dateien `SMSA` und `SMSAID` (zu diesem Aufgabenblatt) aus Stud.IP in Ihr eigenes, übliches **R**-Arbeitsverzeichnis!

Zur Arbeitsweise von `scan`:

- a) Lesen Sie die kopierte Datei `SMSA` aus Ihrem Arbeitsverzeichnis mit Hilfe der Funktion `scan` in **R** in eine Matrix namens `smsa` ein!
- b) Benennen Sie die erste Spalte der Matrix `smsa` mit „ID“ und die restlichen Spalten mit den Buchstaben „A“ bis „K“!

- c) Bestimmen Sie für jede Spalte vom `smsa` die folgenden Kenngrößen: Arithmetisches Mittel, Median, Standardabweichung, erstes und drittes Quartil sowie Minimum und Maximum! (Ist dies für die erste und die letzte Spalte von `smsa` sinnvoll?)
- d) Lesen Sie die kopierte Datei `SMSAID` mit `scan` unter geeigneter Verwendung des Arguments `what` in eine Liste ein! Bauen Sie diese Liste in eine Liste namens `smsaid` um, die *zwei* Komponenten namens `ID` und `City` enthält! In der Komponente `ID` sollen *alle* ID-Nummern stehen und in der Komponente `City` die dazugehörigen Stadtnamen.
- e) Wandeln Sie die Matrix `smsa` in einen Data Frame namens `SMSA.df` um! Benennen Sie die Spalten des Data Frames `SMSA.df` mit kurzen, aber sinnvollen Namen (sinnvoller als „A“ bis „K“!), und zwar gemäß der Bedeutungen der dahinter steckenden Variablen (siehe dazu §3.1.2 im Skript)!
- f) Benennen Sie die Zeilen des Data Frames `SMSA.df` mit den Namen der SMSAs, die in der Komponente `City` der Liste `smsaid` abgelegt wurden! Entfernen Sie aus `SMSA.df`, falls noch nicht geschehen, die Spalte der SMSA-Identifikationscodes!
- g) Wandeln Sie die Variable in `SMSA.df`, die die Codes der geografischen Regionen enthält, in einen (ungeordneten) Faktor mit den Levels `NE`, `NC`, `S` und `W` um, und zwar gemäß der im Skript in §3.1.2 beschriebenen Zuordnung der Codes!
- h) Kontrollieren Sie mit Hilfe der Funktion `str` die Struktur von `SMSA.df` und bestimmen Sie mit `summary` die “summary statistics” von `SMSA.df`!