

Lösung zu Kapitel 5: Beispiel 1

Erstellen Sie ein Working Directory (wie unten angegeben) und kopieren Sie die Dateien `bspdat.csv` und `bspcode.R`, die sich im Datenverzeichnis des Begleitmaterials finden, dort hin.

Ist außerhalb von R durchzuführen

Lösung zu Kapitel 5: Beispiel 2

Setzen Sie Ihr Working Directory auf das Verzeichnis, in dem Sie die Dateien gespeichert haben.

R

```
> setwd("C:/R-Beispiele")
```

Lösung zu Kapitel 5: Beispiel 3

Lesen Sie den Datensatz aus der Datei `bspdat.csv` ein und speichern Sie ihn in einem Objekt namens `bspdat`.

R

```
> bspdat <- read.csv2("bspdat.csv")
```

Lösung zu Kapitel 5: Beispiel 4

Untersuchen Sie die Struktur des Datensatzes mit `head()` und `str()`.

R

```
> head(bspdat)
```

	age	size	weight	female
1	29	173.8	73	0
2	21	179.7	69	0
3	20	162.5	60	1
4	29	174.6	69	1
5	25	163.3	55	1
6	30	181.4	72	0

R

```
> str(bspdat)
```

```
'data.frame':      100 obs. of  4 variables:
 $ age   : int  29 21 20 29 25 30 27 29 20 25 ...
 $ size  : num  174 180 162 175 163 ...
 $ weight: int  73 69 60 69 55 72 67 73 69 59 ...
 $ female: int   0 0 1 1 1 0 1 0 1 0 ...
```

Lösung zu Kapitel 5: Beispiel 5

Der Data Frame `bspdat` enthält eine Variable `female` mit den Werten 1 für *weiblich* und 0 für *männlich*. Fügen Sie dem Datensatz eine neue Variable namens `Geschlecht` hinzu, wozu Sie `female` verwenden. `Geschlecht` soll ein Faktor sein mit den Werten *männlich* bzw. *weiblich*.

R

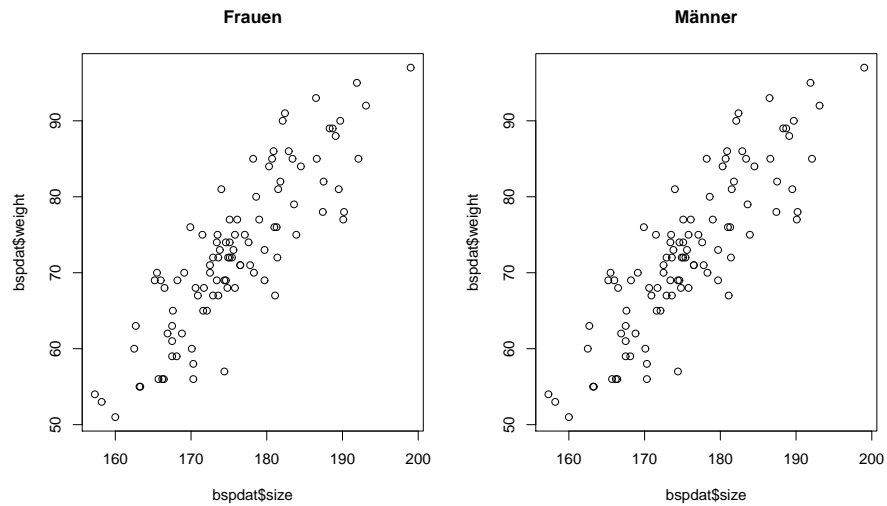
```
> bspdat$Geschlecht <- ifelse(bspdat$female == 1, "weiblich", "männlich")
```

Lösung zu Kapitel 5: Beispiel 6

Stellen Sie mit `par()` ein, dass zwei Plots in einer Zeile erzeugt werden, und erstellen Sie links ein Streudiagramm mittels `plot()` für Größe und Gewicht, wobei Sie mit `subset()` nur Frauen auswählen. Rechts machen Sie das Gleiche, wobei Sie hier nur Männer plotten.

R

```
> par(mfrow = c(1, 2))
> plot(bspdat$size, bspdat$weight, subset = bspdat$Geschlecht ==
+      "weiblich", main = "Frauen")
> plot(bspdat$size, bspdat$weight, subset = bspdat$Geschlecht ==
+      "männlich", main = "Männer")
```



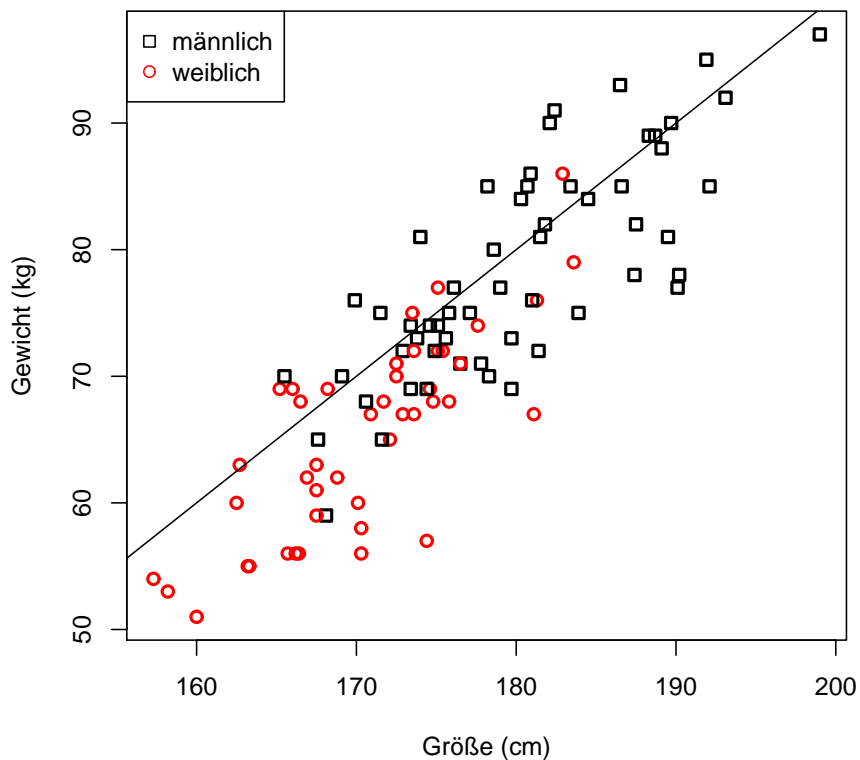
Lösung zu [Kapitel 5: Beispiel 7](#)

Verwenden Sie `source()`, um die Datei `bspcode.R` auszuführen.



```
> source(file("bspcode.R"))
```

Gratulation!



Lösung zu Kapitel 5: Beispiel 8

Woran liegt es, wenn der erzeugte Plot bei Ihnen nur die Hälfte des Fensters ausfüllt? Ist dies bei Ihnen der Fall, so schließen Sie das Grafikfenster und führen Sie `bspcode.R` nochmals aus.

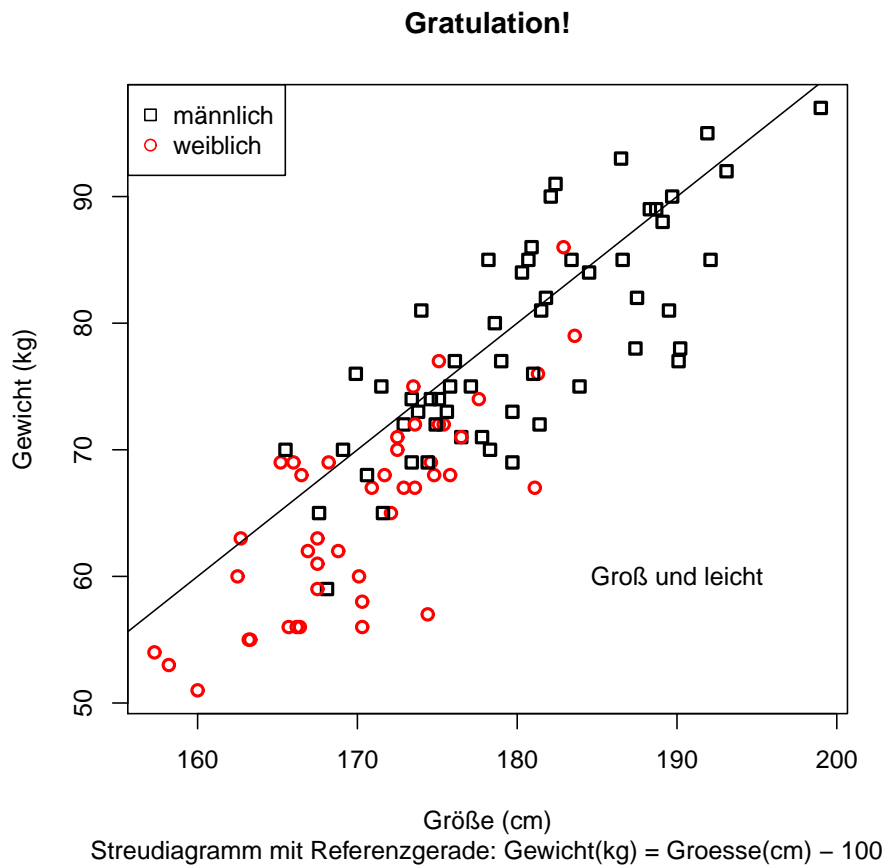
Der Grund ist Übung 6, in der das Grafikfenster für zwei Plots ausgelegt wurde, ohne diese Festlegung wieder rückgängig zu machen.

Grafikparameter können wie bereits im vorhergehenden Kapitel beschreiben auf zwei Arten gesetzt werden: Erstens durch die Funktion `par()`, dann bleiben die Einstellungen so lange bestehen, wie Sie sie nicht durch einen weiteren `par()`-Befehl ändern oder bis Sie das Grafikfenster schließen.

Lösung zu Kapitel 5: Beispiel 9

Fügen Sie an den Koordinaten (190, 60) einen beliebigen Text ein.

```
> source(file("bspcode.R"))
> text(190, 60, "Groß und leicht")
```



Lösung zu [Kapitel 5: Beispiel 10](#)

Speichern Sie die Grafik als Metafile mit der Endung `.emf` ab.

Wird nicht direkt durch R-Befehle veranlasst.