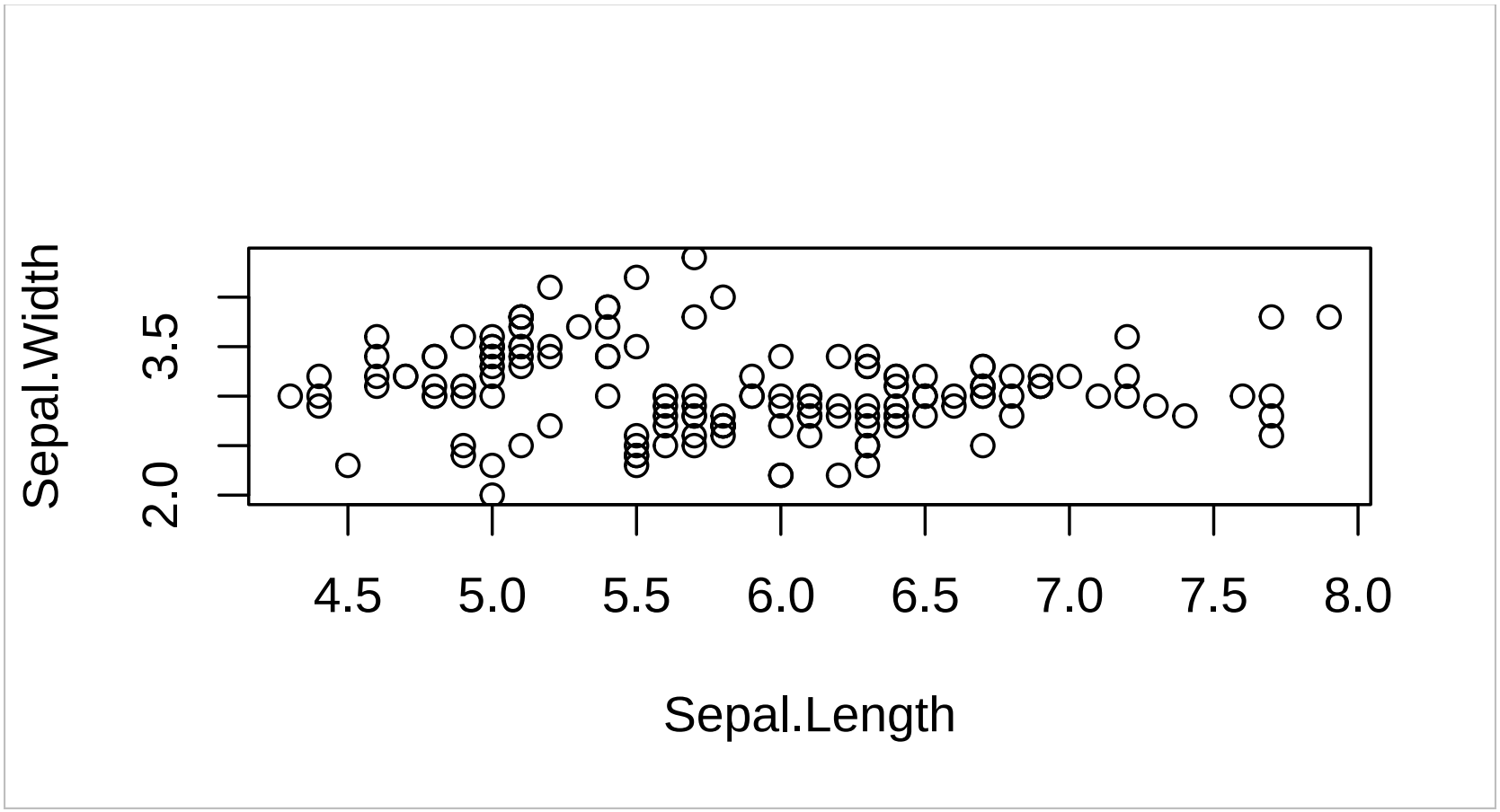
Programmieren mit R für Einsteiger

4. Grafiken / 4.5 Komposition

Berry Boessenkool 2022-02-25 11:41

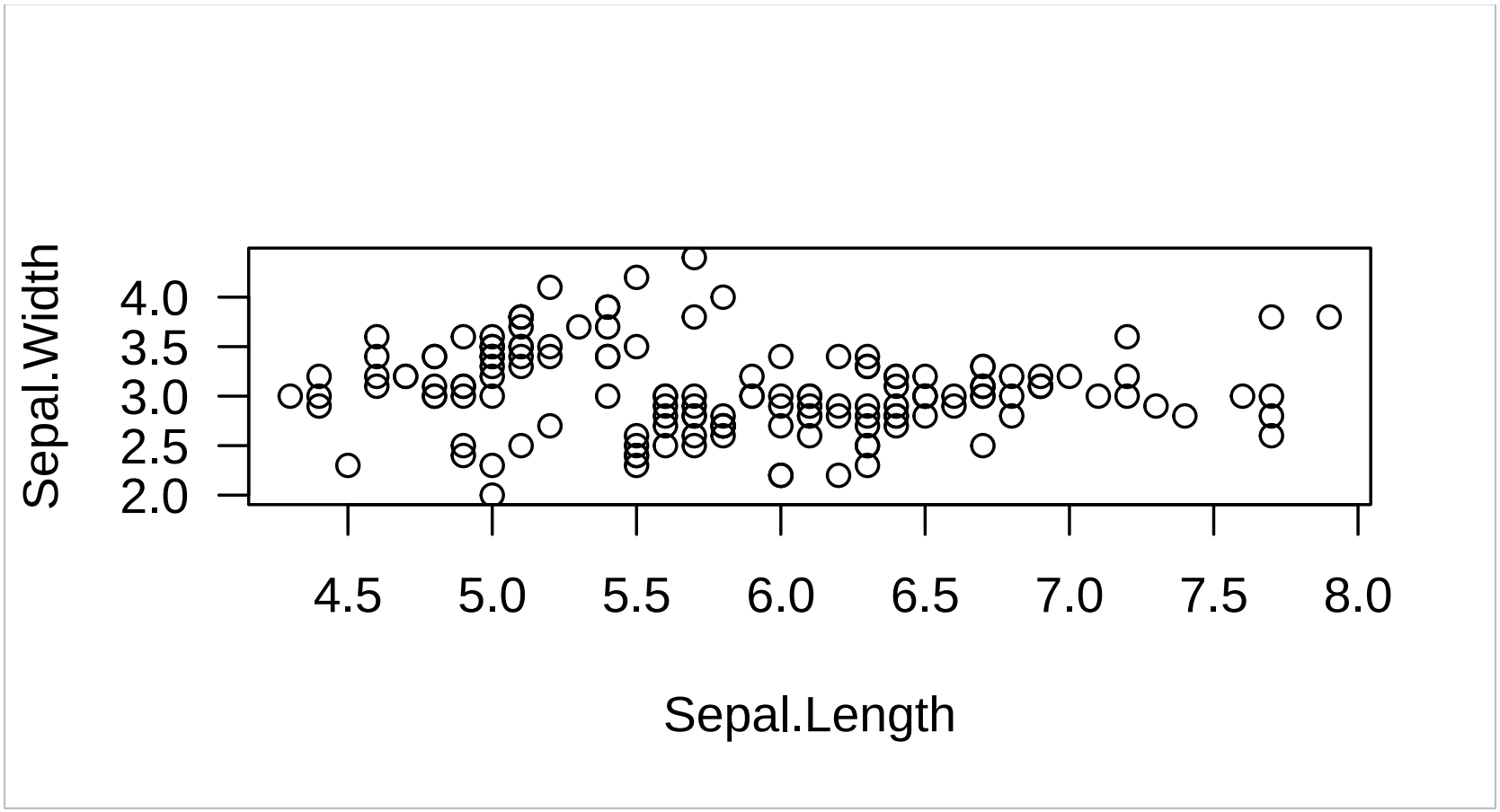
`par`: Default Parameter einer Grafik

```r
# default (standard) Parameter:
plot(Sepal.Width~Sepal.Length, data=iris)
```



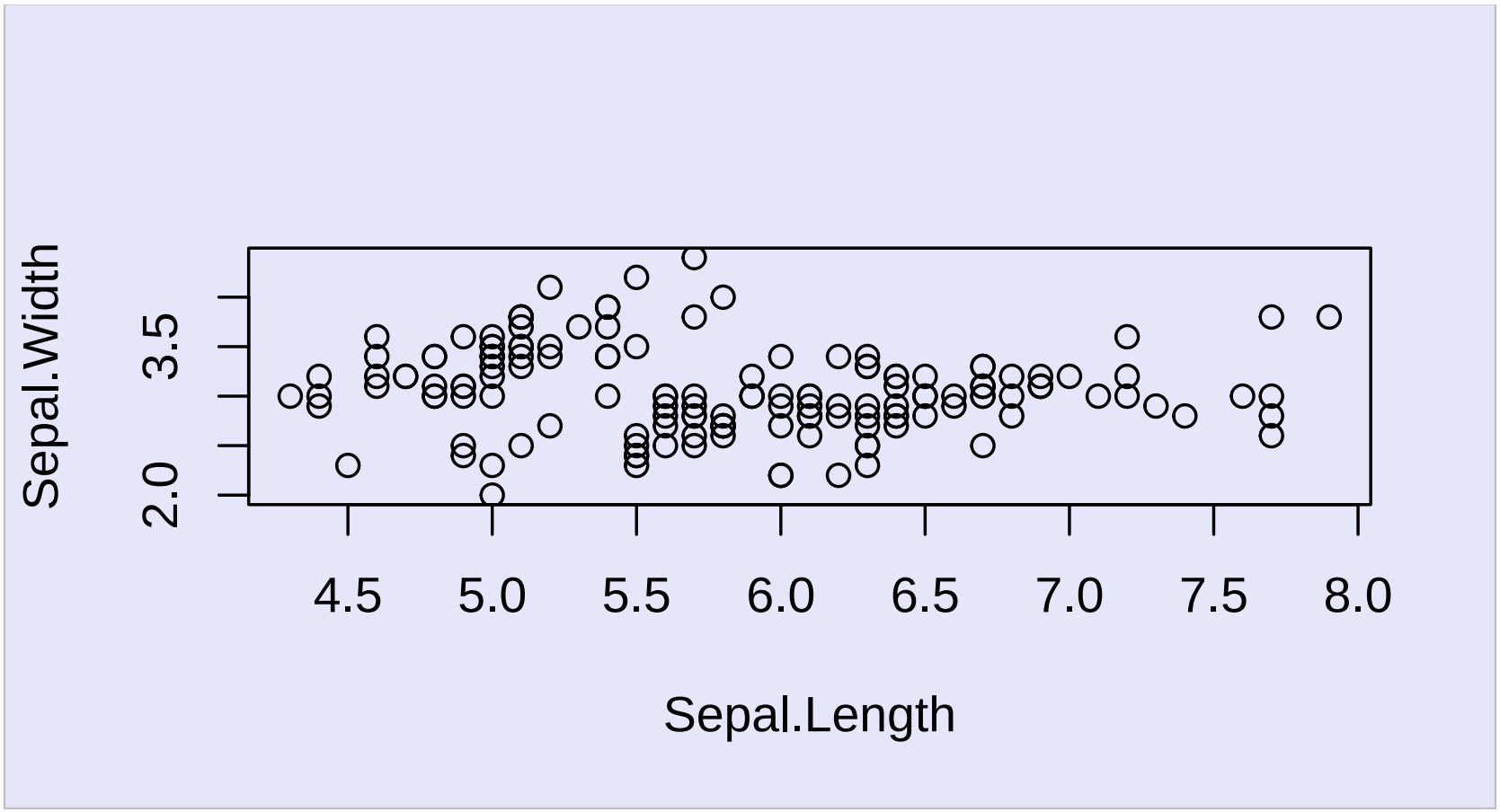
`par`: Parameter der folgenden Grafiken

```
par(las=1) # ab jetzt für alle Plots im gleichen Device:
plot(Sepal.Width~Sepal.Length, data=iris)
```



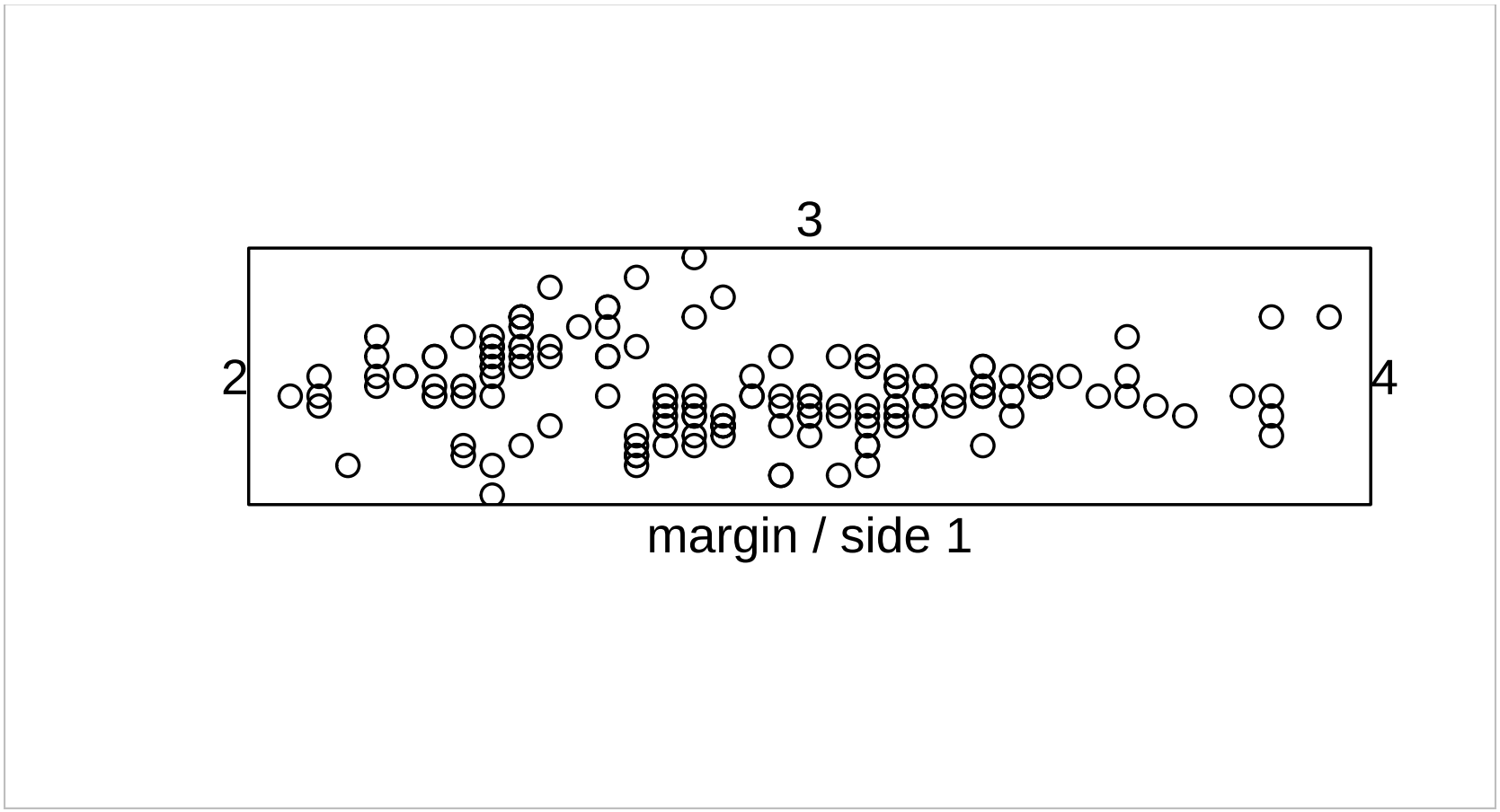
`par` : bg - **b**ack**g**round nicht für `pch=21:25` sondern für Device

```
par(bg="lavender") # Device = zB pdf, RStudio Grafiken
plot(Sepal.Width~Sepal.Length, data=iris)
```



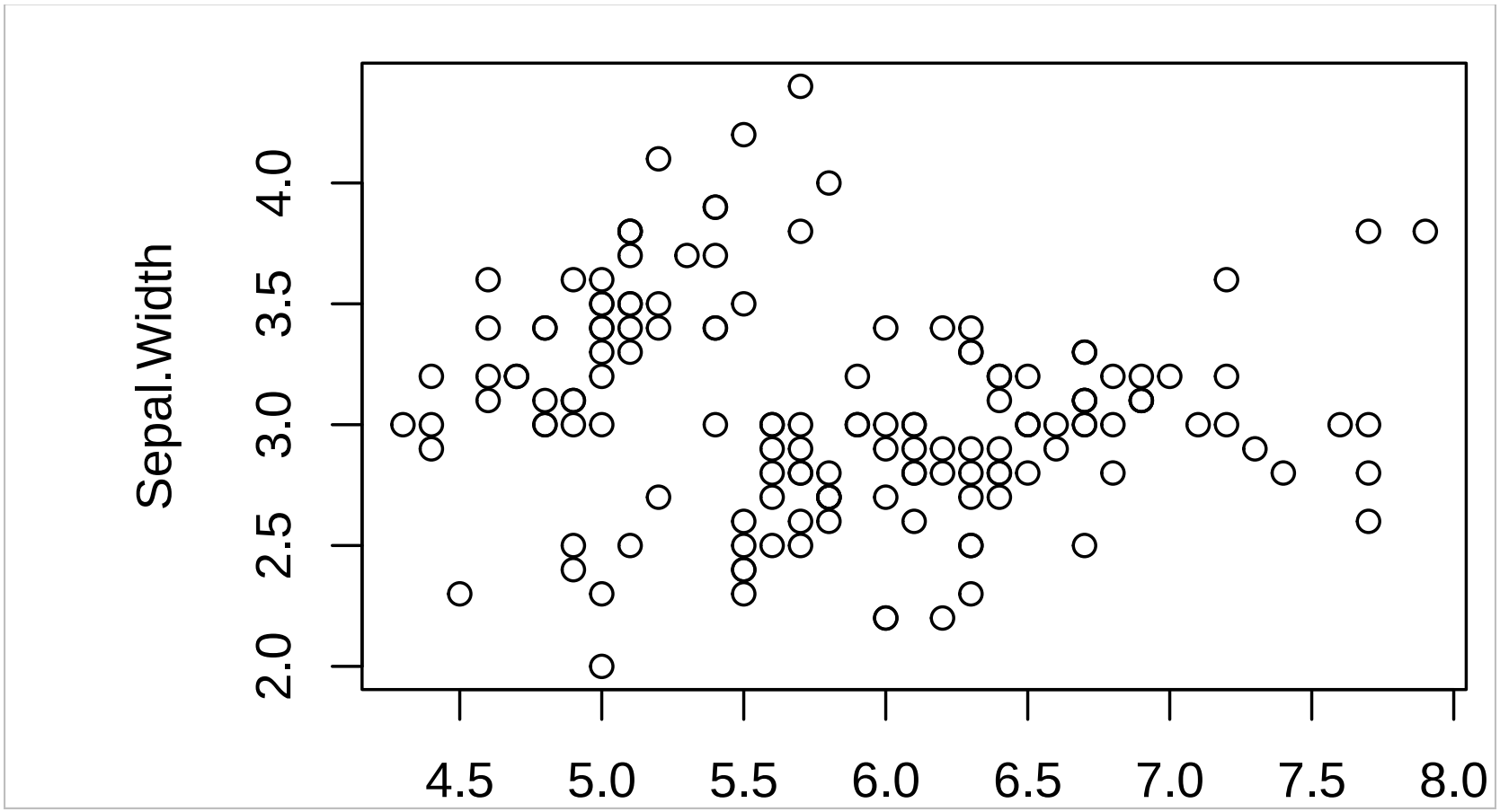
par : Reihenfolge der Rand-Argumente

```
plot(Sepal.Width~Sepal.Length, data=iris, axes=F, ann=F)
mtext(c("margin / side 1",2:4), side=1:4, las=1) ; box()
```



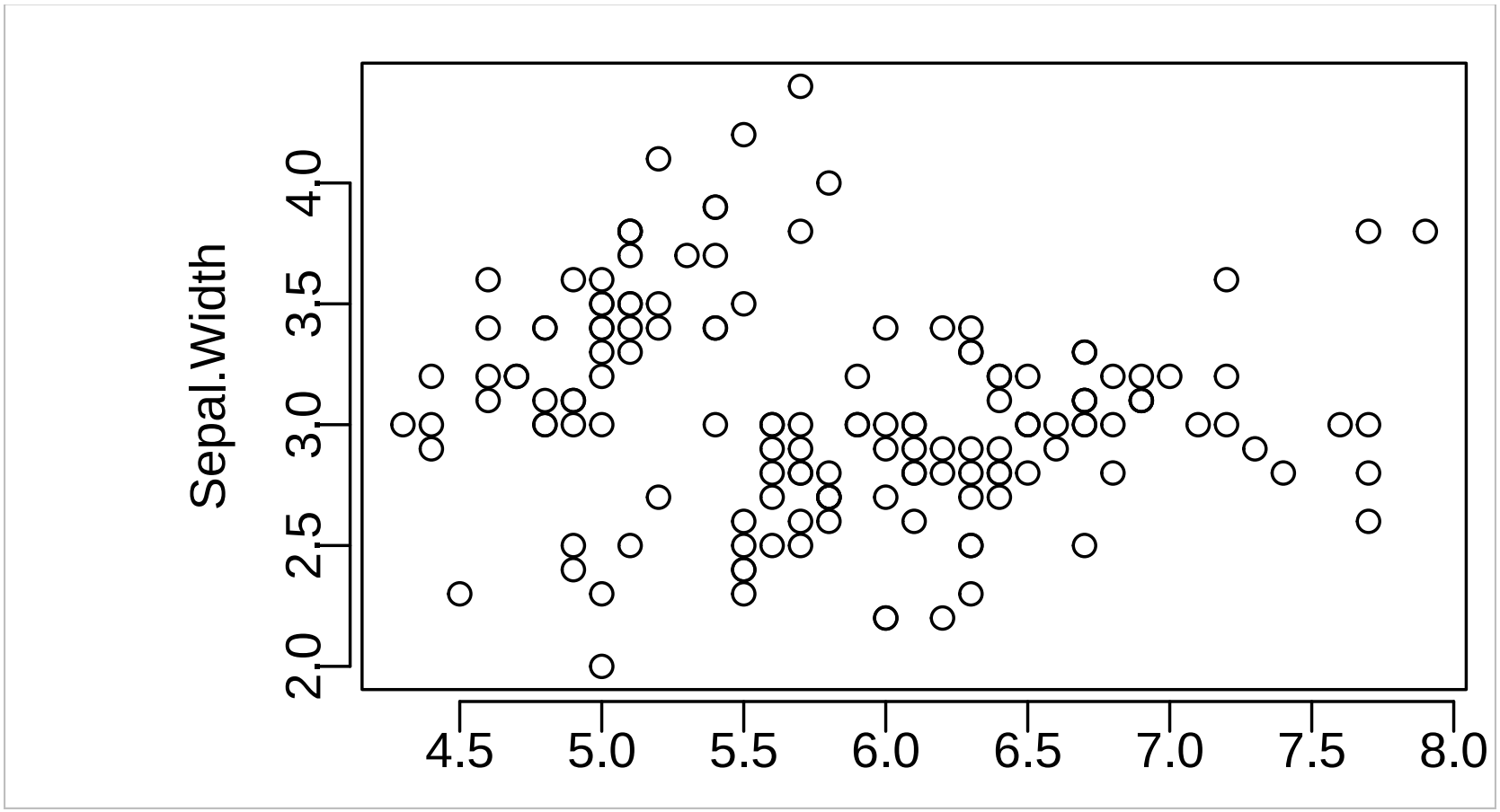
par : mar - margins = Ränder

```
par(mar=c(2,6,1,0.5)) # Einheit: Textzeilen
plot(Sepal.Width~Sepal.Length, data=iris)
```



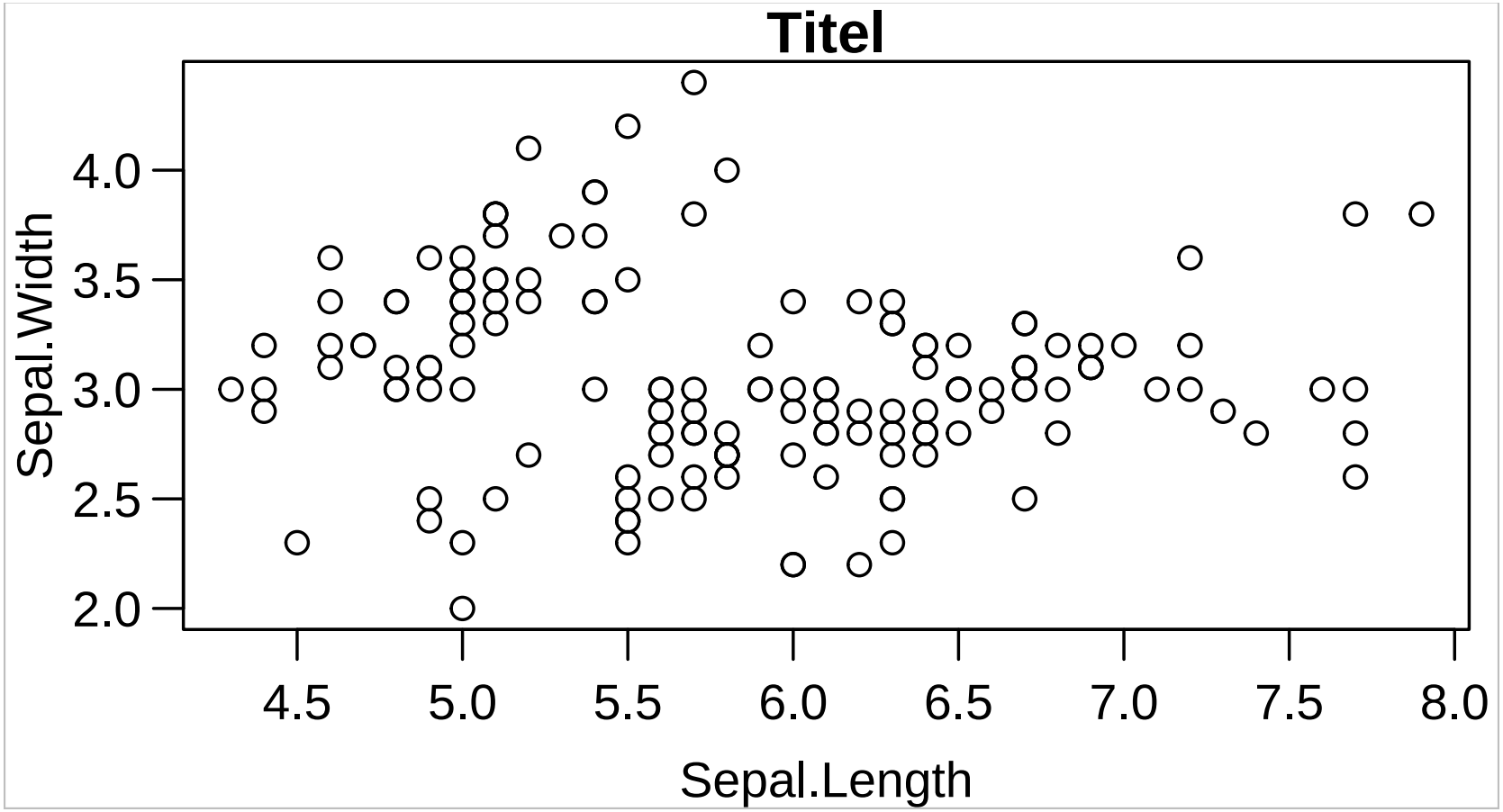
par : mgp - margin placements

```
par(mar=c(2,6,1,0.5), mgp=c(2.1, 0.5, 0.2)) # Abst. Titel,
plot(Sepal.Width~Sepal.Length, data=iris) # Zahlen, Linie
```



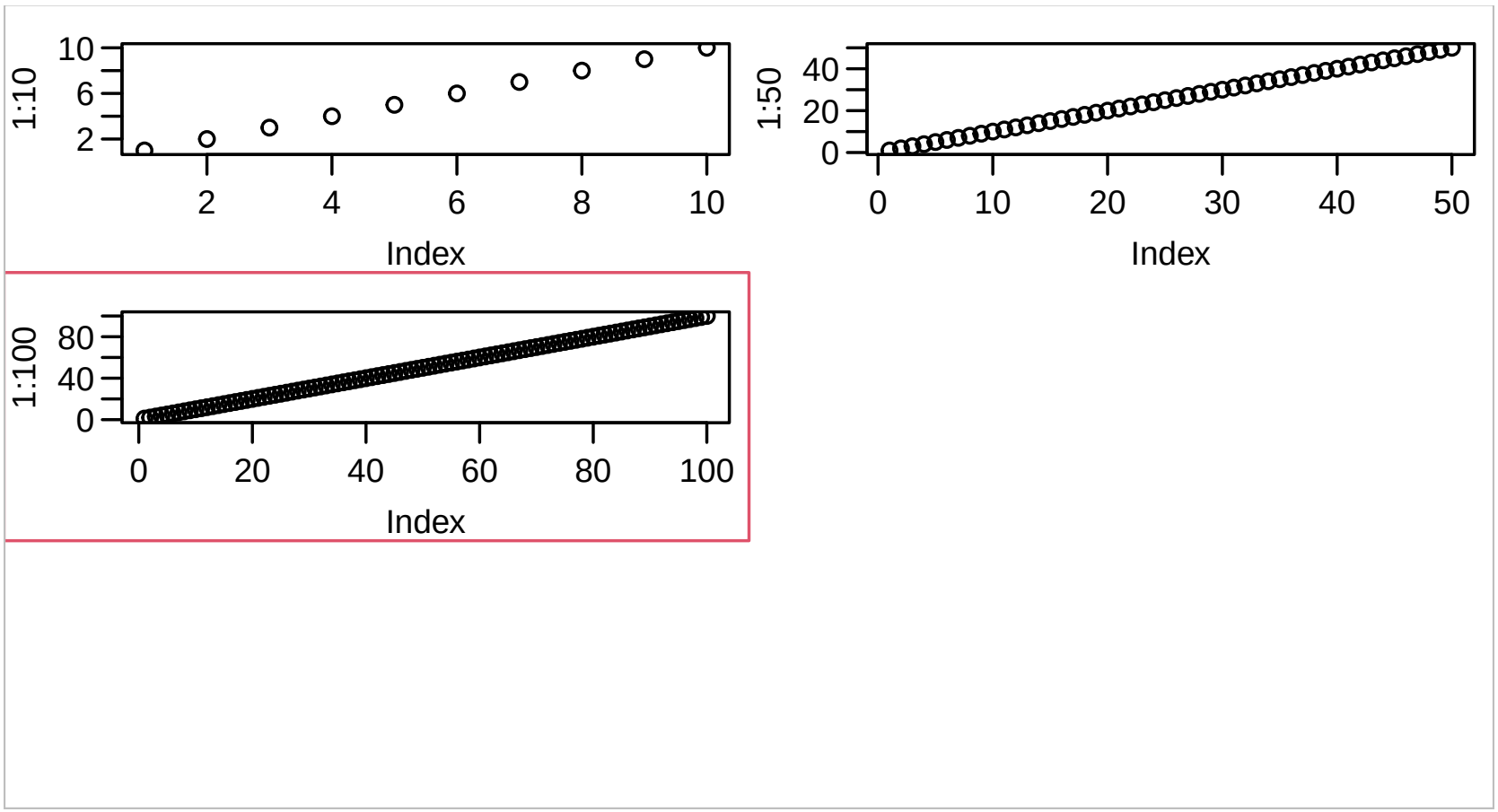
`par`: Häufig praktisch für minimalistische Ränder

```
par(mar=c(3,3,1,0.5), mgp=c(2,0.7,0), las=1)
plot(Sepal.Width~Sepal.Length, data=iris, main="Titel")
```



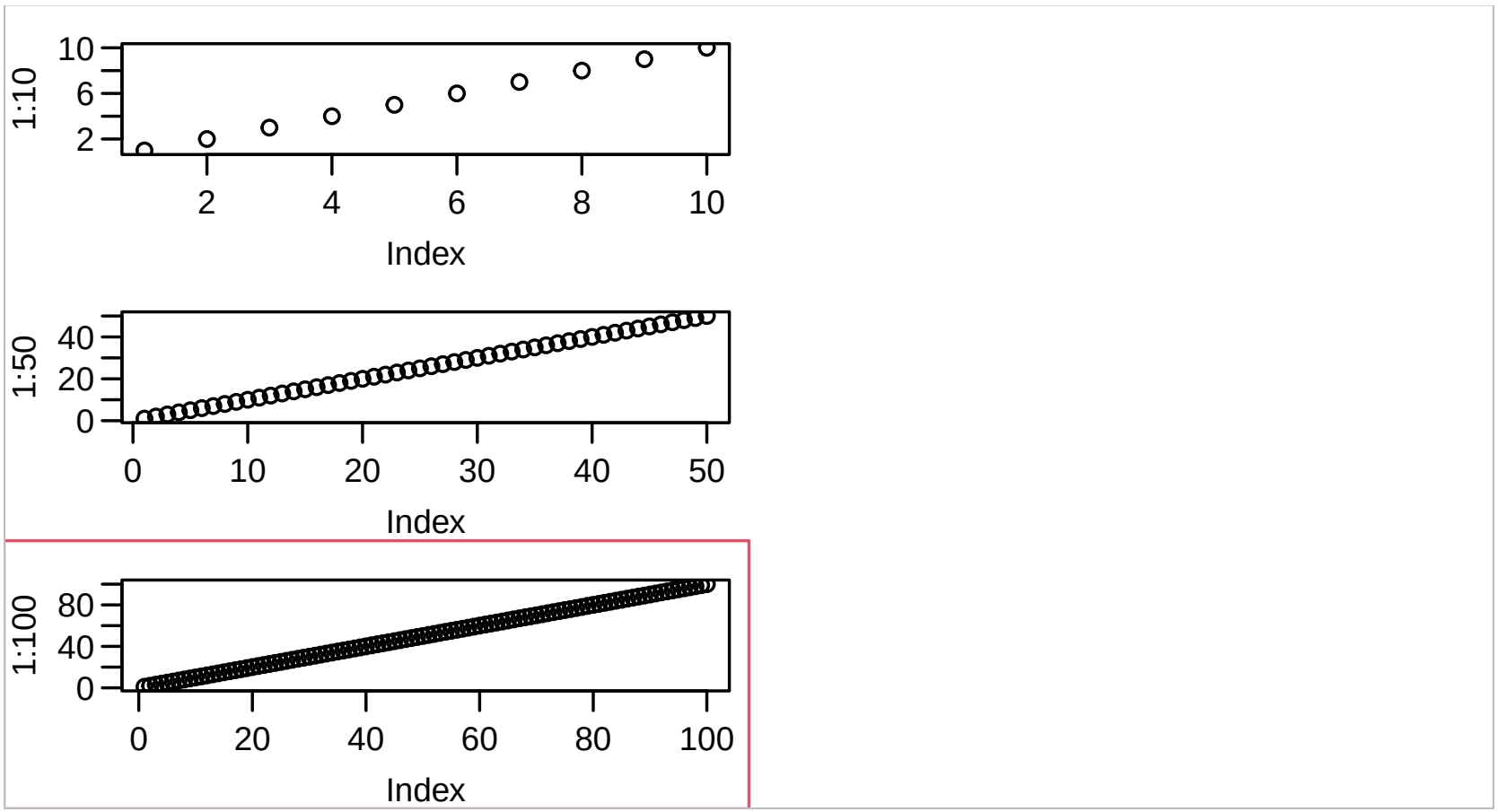
par : mfrow - multiple figures, rowwise gefüllt

```
par(mfrow=c(3,2), mar=c(3,3,1,0.5),mgp=c(2,0.7,0),las=1)
plot(1:10); plot(1:50); plot(1:100); box("figure", col=2)
```



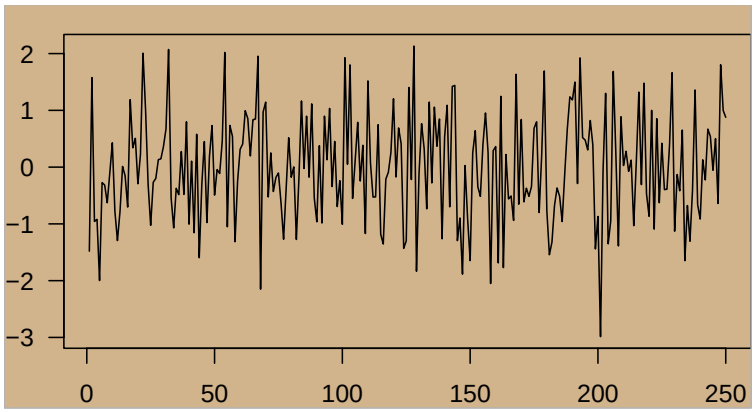
par : mfcol - multiple figures, columnwise gefüllt

```
par(mfcol=c(3,2), mar=c(3,3,1,0.5),mgp=c(2,0.7,0),las=1)
plot(1:10); plot(1:50); plot(1:100); box("figure", col=2)
```



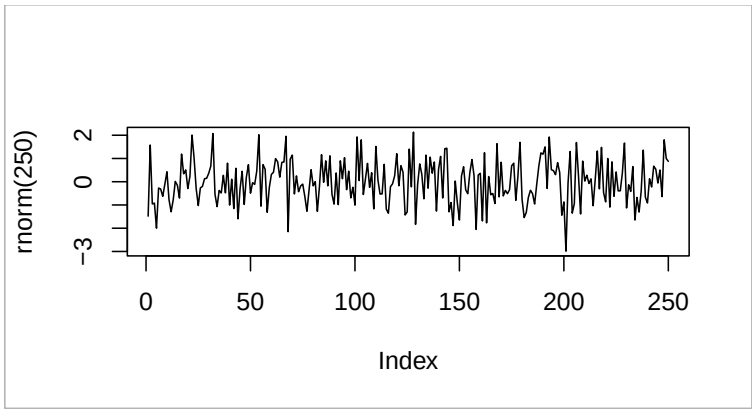
par Einstellungen wiederherstellen

```
op <- par(mar=c(2,2,1,0),
          las=1, bg="tan")
plot(rnorm(250), type="l")
box("figure",col="grey70")
```




```
op # ursprüngliche Parameter
## $mar
## [1] 5.1 4.1 4.1 2.1
## $las
## [1] 0
## $bg
## [1] "transparent"
```

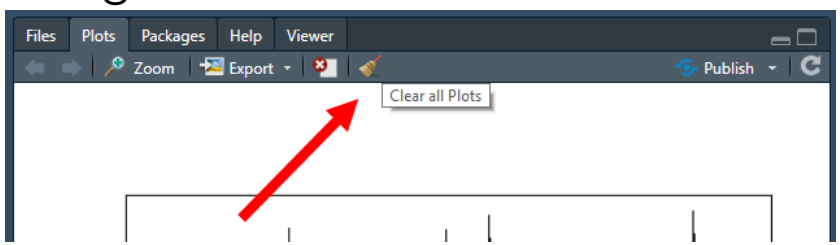
```
par(op)
plot(rnorm(250), type="l")
box("figure",col="grey70")
```



Alternative:

```
dev.off() # graphics.off()
```

Device schließen, beim nächsten Plot sind alle Standards wieder hergestellt.
Das geht auch in RStudio:



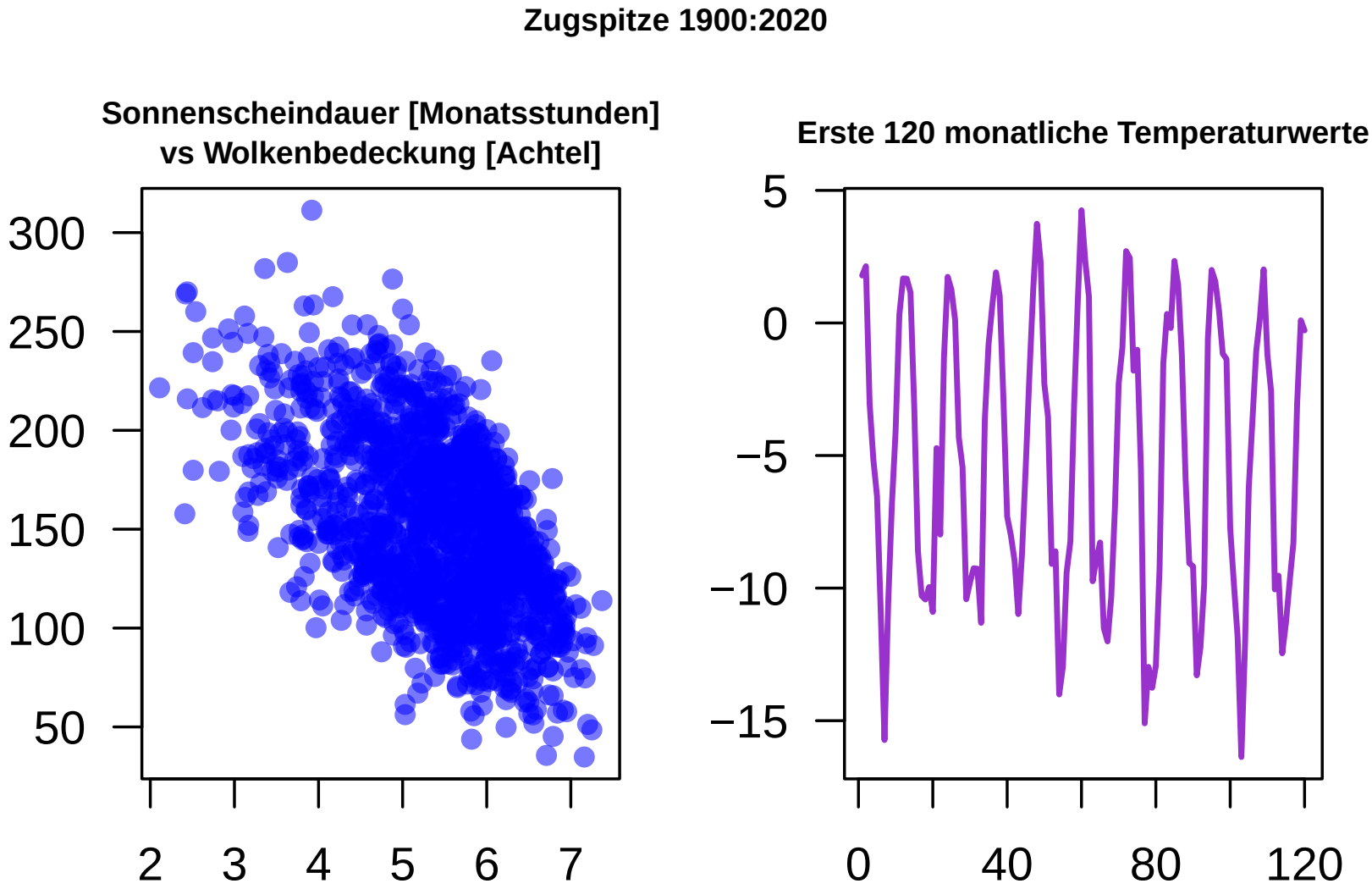
Zusammenfassung

Multipanel Diagramme

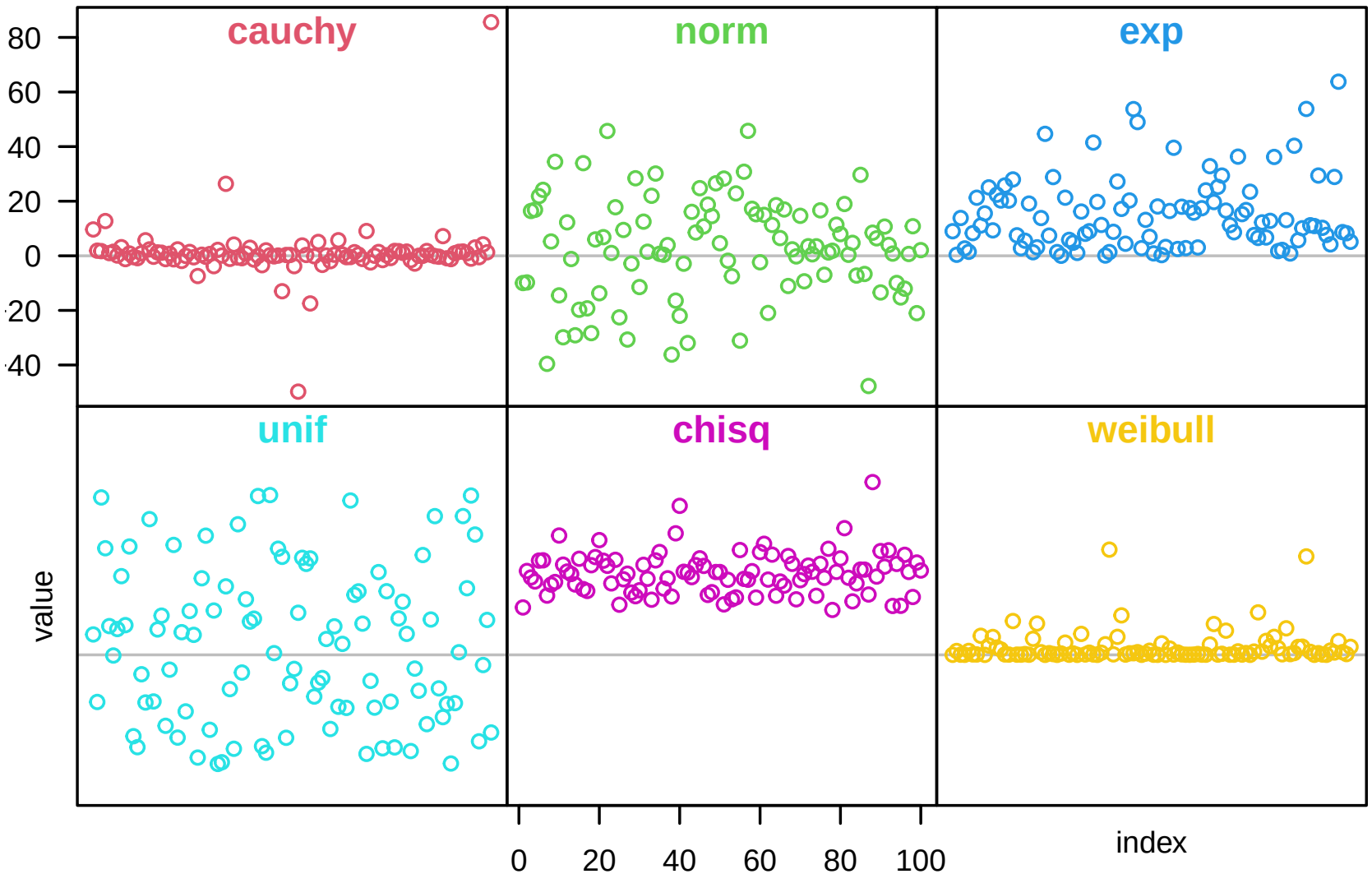
`par` : Einstellungen für Grafiken (Parameter)

- `par(mfrow,mfcol, mar,mgp, las, bg)`

Lösung für Übung A8

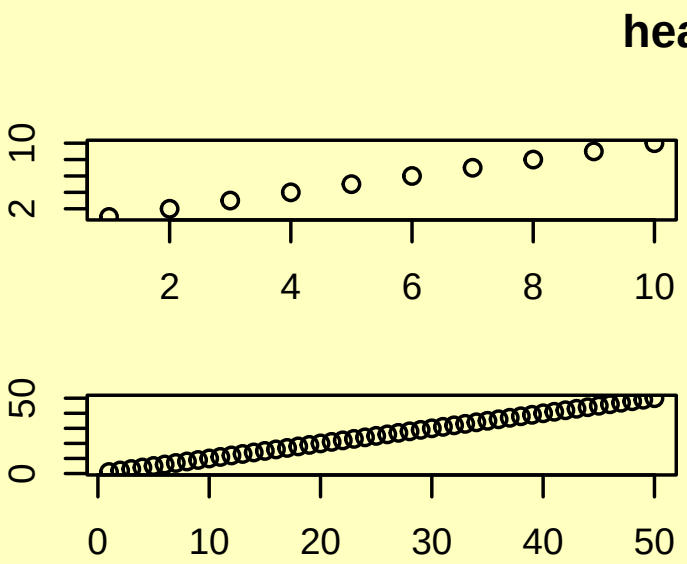


Lösung für Übung B2

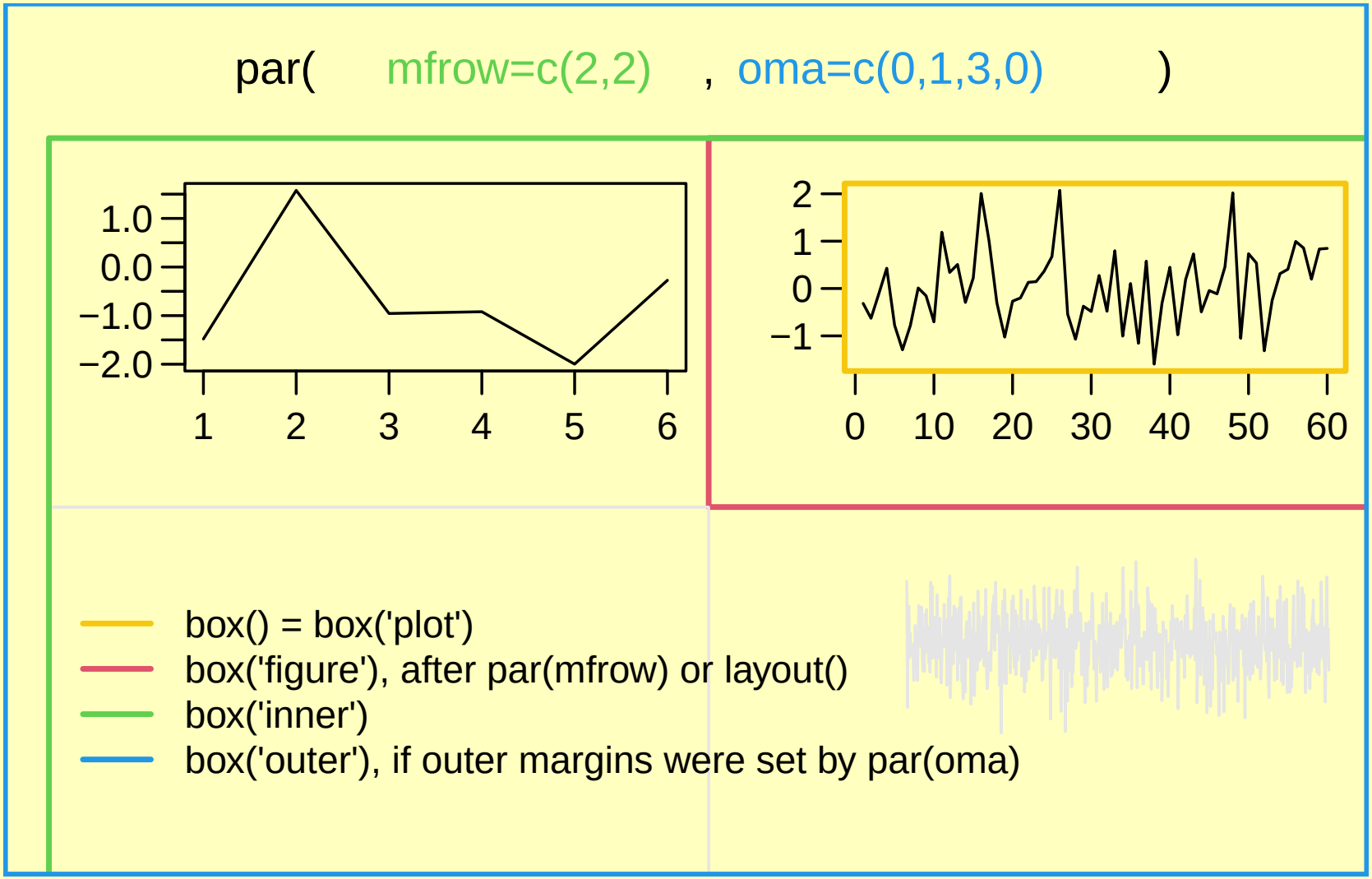


par : oma - outer margins

```
par(oma=c(0,4,3,0), mfcol=c(3,2), mar=c(3,3,1,0.5) )
plot(1:10); plot(1:50); title(main="header", outer=TRUE)
```

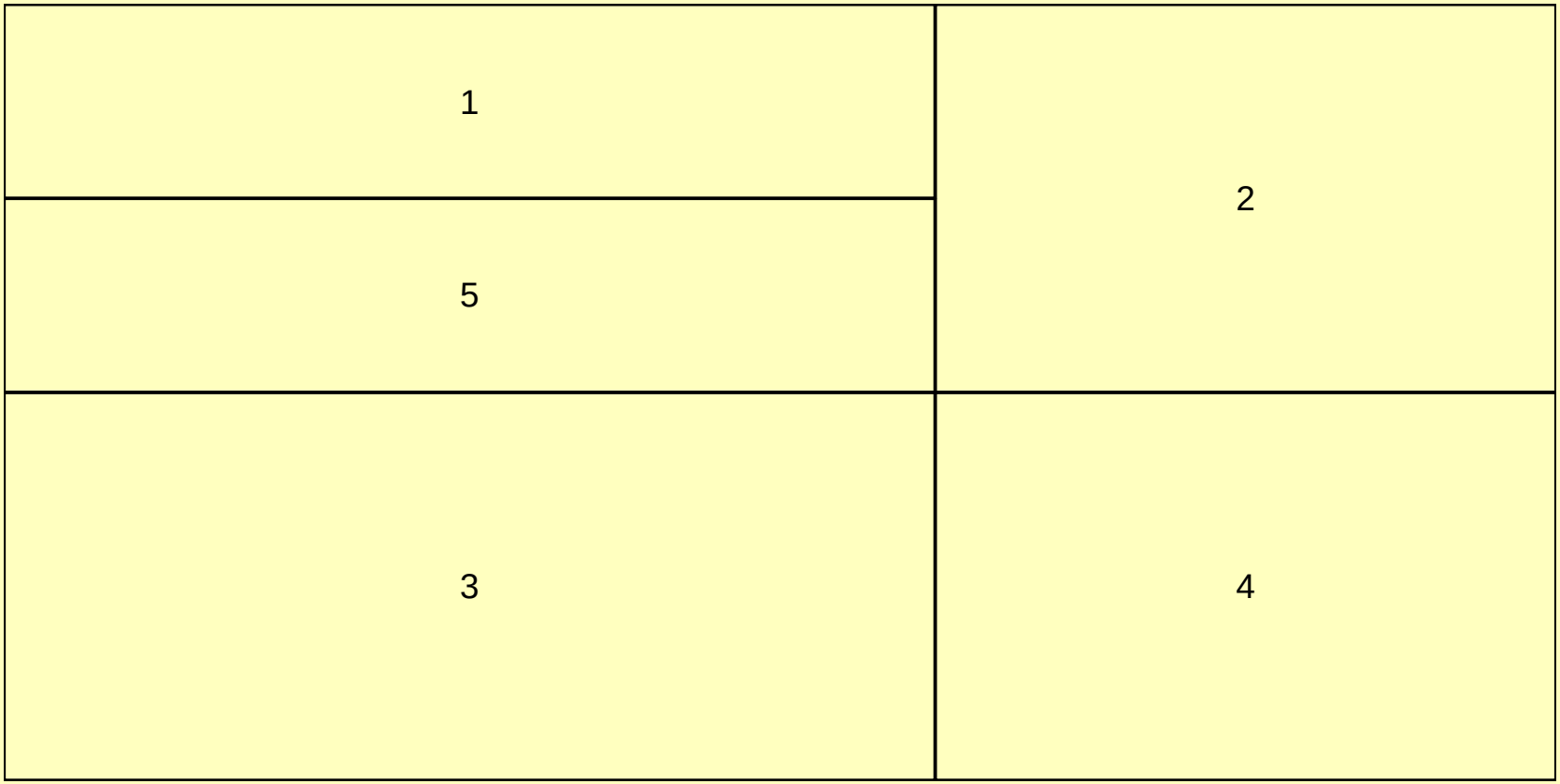


box Übersicht



Flexibles Multipanel Diagramm mit `layout`

```
lay <- layout(matrix(c(1,1,2,2,
                       5,5,2,2,
                       3,3,4,4,
                       3,3,4,4), ncol=4, byrow=TRUE),
              widths=c(6,6,4,4))
layout.show(lay)
```



layout Beispiel mit Verteilung am Rand (marginal histogram)

Code im Examples-Abschnitt zu `?layout`: