LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistische Software (R)

Paul Fink, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Fortgeschrittene Grafiken

Das `type` Argument in `plot()`

```
> y <- rnorm(20)
> plot(y, type = "p")
> plot(y, type = "l")
> plot(y, type = "b")
> plot(y, type = "h")
```

`type` Argument interpretiert x- und y-Werte unterschiedlich:

`type="p"`	x- und y- Werte repräsentieren die Koordinaten von Punkten
`type="l"` `type="o"`	Punkte werden durch Linie verbunden
`type="h"`	Stabdiagramm mit x-Werte die Position der Stäbe und y-Werte als Höhe der Stäbe

Weitere Elemente hinzufügen: *low-level*

Szenario für Grafik:
Kurven von mittleren Abweichungen bei 2 Maschinen:

```
> set.seed(122333)
> maschine1 <- rnorm(30)
> maschine2 <- rnorm(30)
```

Nun sollen beide in einer Grafik auftauchen:

```
> plot(maschine1, type = "l", main = "Verlauf von 2 Maschinen",
+      ylab = "mittlere Abweichung", xlab = "Tage", col = "red")
> # Versuch: Verlauf von y zu Plot hinzufuegen
> plot(maschine2, type = "l", main = "Verlauf von 2 Maschinen",
+      ylab = "mittlere Abweichung", xlab = "Tage", col = "blue")
> # Macht einen neuen Plot!!
```

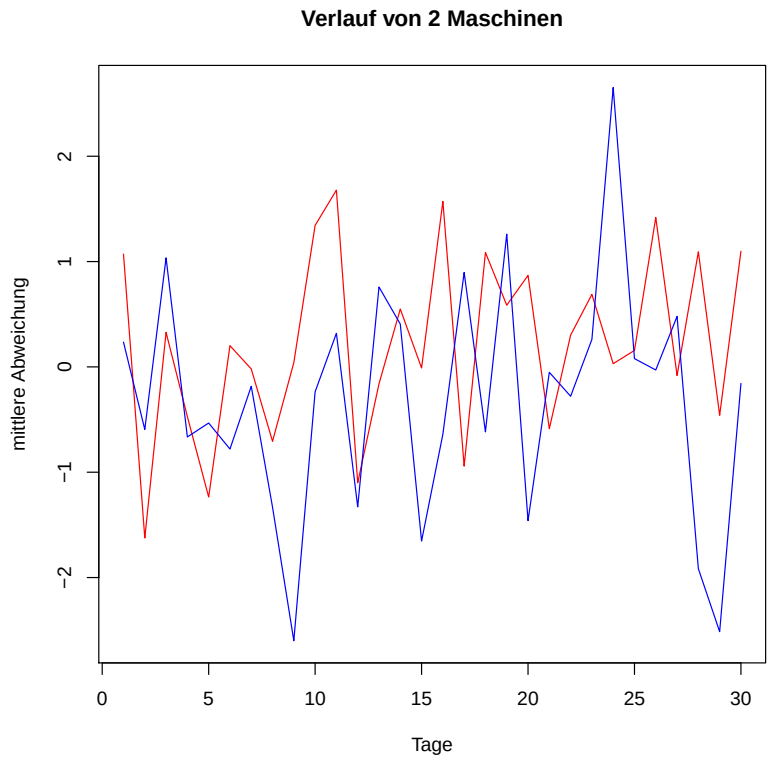
Nicht mit High-Level Funktionen zu machen!

$\Longrightarrow$ Low-Level-Funktionen nutzen

Weitere Elemente hinzufügen: *low-level*

- Zeichnen des Verlaufs der 1. Machine mit `plot()` (high-level)
- Zeichnen des Verlaufs der 2. Machine mit `lines()` (low-level)

```
> yrange <- range(maschine1, maschine2)
> plot(maschine1, type = "l",
+      main = "Verlauf von 2 Maschinen",
+      ylim = yrange, xlab = "Tage",
+      ylab = "mittlere Abweichung",
+      col = "red")
> lines(maschine2, col = "blue")
```



Übersicht low-level Funktionen

Funktion	Beschreibung
`points()`	Punkte an Stellen (x, y)
`lines()`	Linien zwischen den Stellen (x, y)
`segments()`	Liniensegmente, ausgehend von $(x0, y0)$ zu allen Punkten in (x, y)
`arrows()`	ähnlich wie `segements()`, nur mit Pfeilspitzen
`xspline()`	Geglättete Kurve durch die Punkte (x, y)
`rect()`	Rechteck, mit linker unterer Ecke (xl, yb) und rechter oberer Ecke (xr, yt)
`polygon()`	Polygonzug mit Knoten in (x, y)
`text()`	Text hinzufügen an Position (x, y)
`title()`	Beschriftung(en) zur Grafik hinzufügen
`axis()`	Achsen hinzufügen
`abline()`	Eine oder mehrere Geraden
`grid()`	Gitternetz

Globale Konfiguration

Grafiklayout und globale Einstellungen ändern mit Funktion `par()`

`par(Parameter1 = Wert1, usw.)` vor ersten Grafikfunktion

Überblick über meist verwendete Parameter

Parameter	Beschreibung
`ask`	Wenn `TRUE`, Zeichnen eines neuen Plots durch Drücken der Eingabetaste
`cex`	Vergrößerung (siehe Tabelle in Folien 9)
`las`	Ausrichtung der Achsenbeschriftung
`mfrow`, `mfcol`	Mehrere Grafiken in einem Display, Vektor `c(nr, nc)`, Einzeichen zeilen- bzw. spaltenweise
`xaxs`, `xaxt`	Konfiguration der x-Achse
`yaxs`, `yaxt`	Konfiguration der y-Achse

Mathematische Ausdrücke in Grafiken

Text mit z.B. griechischen Symbolen in `expression()` stecken:

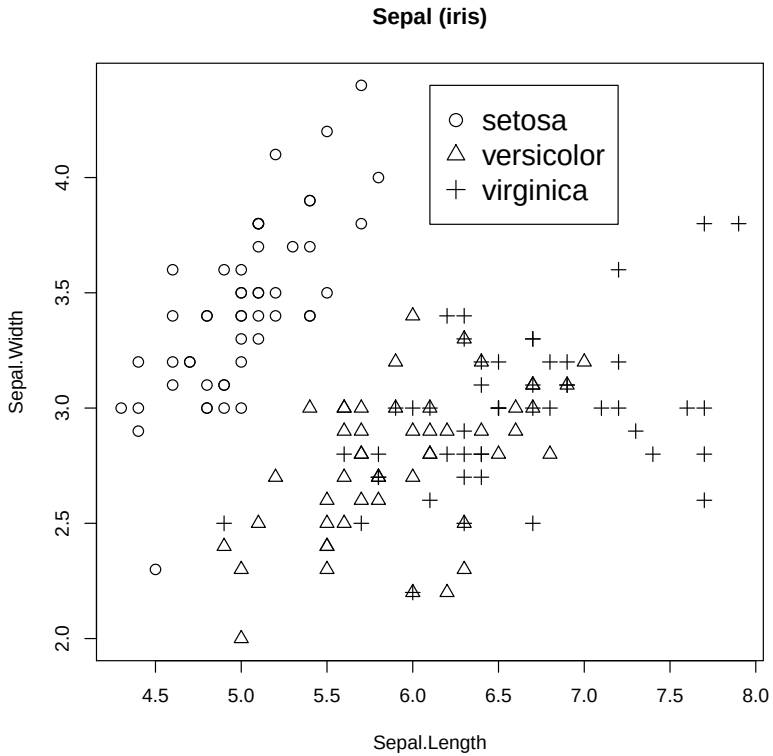
Beispiel aus ?`plotmath`:

```
> x <- seq(-4, 4, len = 101)
> y <- cbind(sin(x), cos(x))
> matplot(x, y, type = "l", xaxt = "n",
+         main = expression(paste(plain(sin) * phi, "  and  ",
+                                 plain(cos) * phi)),
+         ylab = expression("sin" * phi, "cos" * phi), # only 1st is taken
+         xlab = expression(paste("Phase Angle ", phi)),
+         col.main = "blue")
> axis(1, at = c(-pi, -pi/2, 0, pi/2, pi),
+      labels = expression(-pi, -pi/2, 0, pi/2, pi))
```

Legenden

Beispiel:

```
> with(iris,
+ plot(Sepal.Length, Sepal.Width,
+      main = "Sepal (iris)",
+      pch = as.numeric(Species),
+                  cex = 1.2)
+ )
>
> legend(x = 6.1, y = 4.4,
+        legend = c("setosa",
+                   "versicolor",
+                   "virginica"),
+        cex = 1.5, pch = 1:3)
```



Achtung: Es nicht sichergestellt, dass die Legende zur Grafik passt!

Umgang mit Farben

Übersicht über die built-in Farben: `colors()`

Farben sind sehr hilfreich in Grafiken, aber nicht jede Kombination!

Zum Umgang mit Farben: Präsentation von Achim Zeileis (2010)

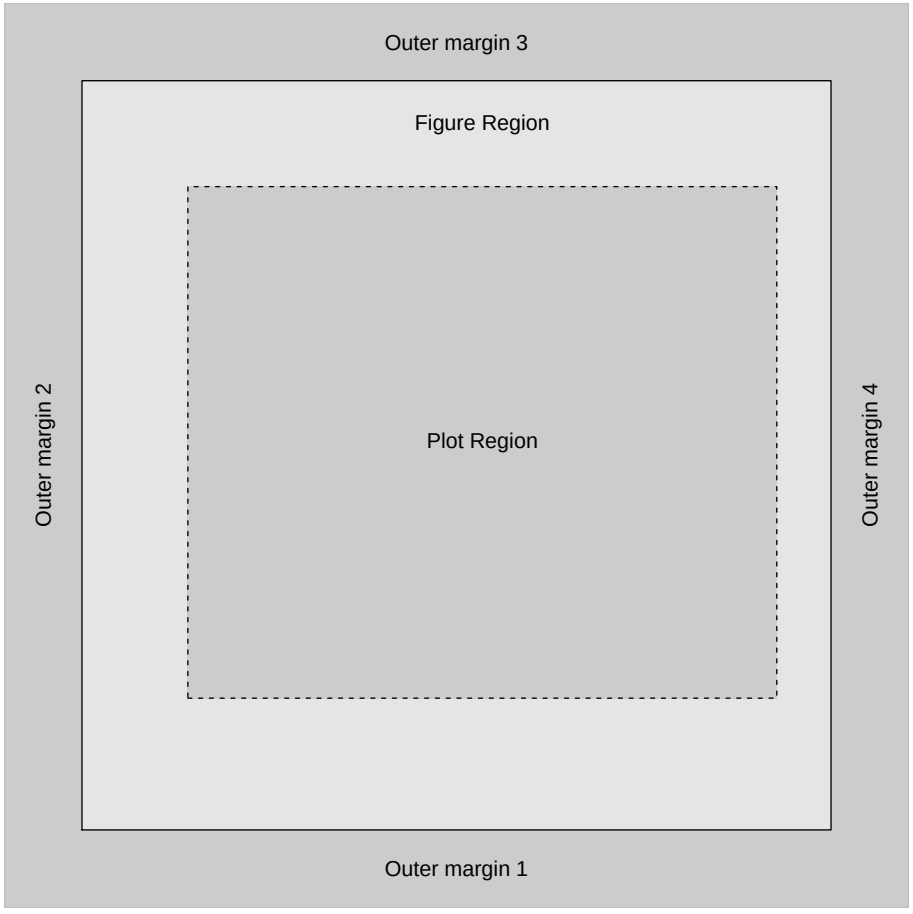
Farbwahl abhängig von Verwendung: Beamer, Ausdruck in verschiedener Qualität, Bildschirm

- Bei Abstufungen nicht zu ähnliche Farben wählen
- Idealerweise Farbblindheit berücksichtigen (Rot-Grün-Schwäche)
- Oft reichen auch Graustufen

Online Colorbrewer liefert einen guten Start zur Farbwahl:
`http://colorbrewer2.org/`

Exkurs: Aufbau von traditionellen Grafiken[1]

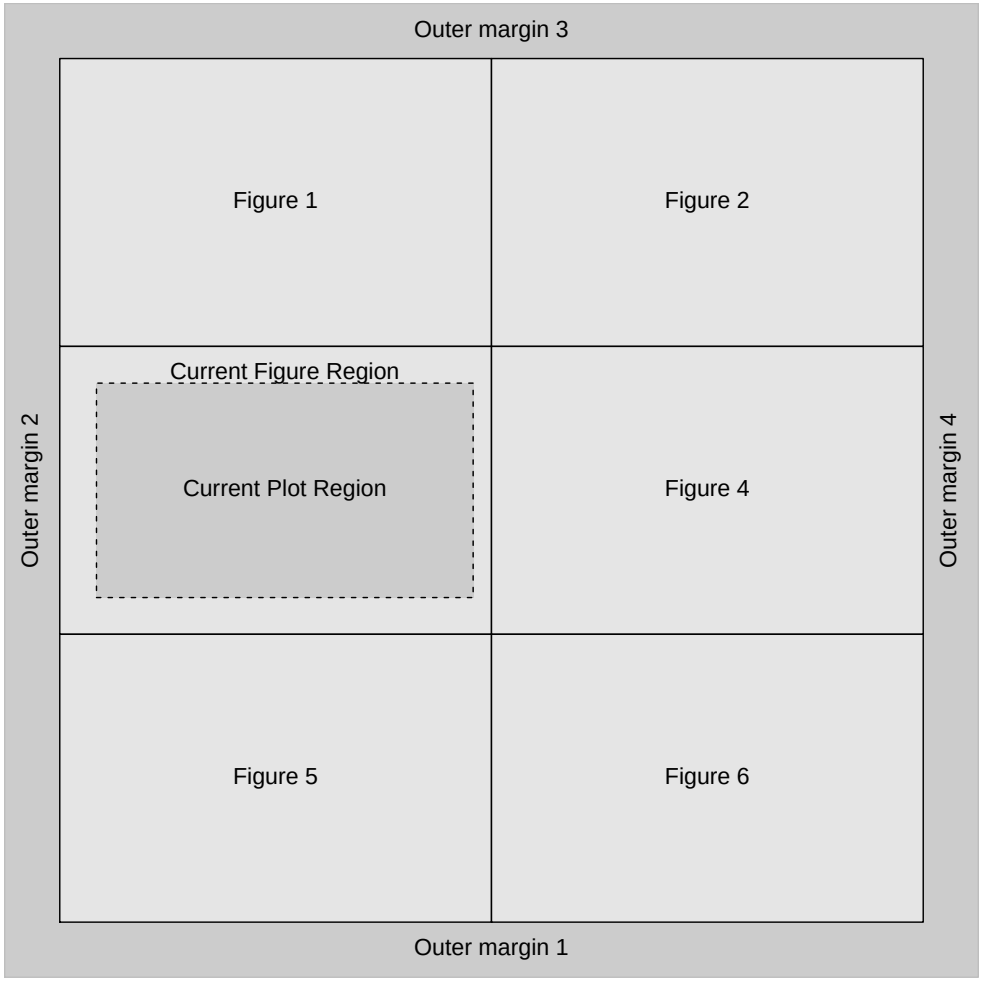
Aufbau eines Plots:



[1] Quelle: *Murrell (2011): R Graphics, 2nd Edition, Chapman & Hall*

Exkurs: Aufbau von traditionellen Grafiken

Aufbau mehrerer Plots:



Exkurs: Aufbau von traditionellen Grafiken

Koordinatensystem: