
Von der mathematischen Biologie zur Systembiologie

(Vorlesung Prof. Dr. J. Timmer)

Aufgabenzettel Nr. 1

Erzeugen von Plots mit ggplot

Installieren Sie sich zunächst die Pakete `ggplot2` und `grid`, indem Sie in der R-Kommandozeile

```
install.packages('ggplot2')
install.packages('grid')
```

eingeben. Anschließend laden Sie die Pakete mit `load(ggplot2); load(grid)`. Die Plot-Erzeugung mit `ggplot` basiert immer auf `data.frame`-Objekten. Hierbei handelt es sich um ein R-eigenes Tabellenformat, in dem jede Spalte ein Vektor eines anderen Typs sein kann, z.B. `character`, `numeric` oder `factor`. Jede Spalte hat standardmäßig einen Namen, der über den `colnames()`-Befehl selbst gesetzt werden kann.

Bei `ggplot` wird ein Plot zunächst initialisiert und dann mit dem “+”-Operator um grafische Elemente erweitert. Typische Beispiele sind

```
ggplot(mydata, aes(x=V1, y=V2)) + geom_line()
ggplot(mydata, aes(x=V1, y=V2)) + geom_line() + geom_point() + xlab('Äpfel')
```

Hierbei ist `mydata` ein `data.frame` mit Spalten `V1` und `V2` welche die x - und y -Koordinaten enthalten. Um mehrere Kurven in einen Plot zu zeichnen muss das `data.frame` um eine Spalte erweitert werden, z.B.

```
ggplot(mydata, aes(x=V1, y=V2, group=V3, color=V3)) + geom_line()
```

Hierdurch werden für jeden Wert von `V3` die zugehörigen x - und y -Koordinaten als Linie geplottet. Durch das Argument “`color=V3`” werden diese zusätzlich durch unterschiedliche Farben kenntlich gemacht. Um hingegen die Kurven für verschiedene `V3`-Werte in unterschiedliche Panels zu plotten kann

```
ggplot(mydata, aes(x=V1, y=V2)) + facet_wrap(~V3) + geom_line()
```

verwendet werden. Für den Fall, dass ein Plot für Werte von `V3` und `V4` erzeugt werden soll, gibt es

```
ggplot(mydata, aes(x=V1, y=V2)) + facet_grid(V3~V4) + geom_line()
```

Bei `ggplot` lohnt es sich, im Hintergrund stets die Dokumentation mit vielen Beispielen im Browser geöffnet zu haben: <http://docs.ggplot2.org/current/>.

Aufgabe 1: Verhulst-Dynamik

In der Vorlesung wurde gezeigt, dass die Lösung der Differentialgleichung

$$\dot{x} = rx \left(1 - \frac{x}{K}\right), \quad x(0) = x_0 \quad (1)$$

durch

$$x(t) = \frac{Kx_0}{x_0 + (K - x_0)e^{-rt}} \quad (2)$$

gegeben ist. Implementieren Sie zunächst Gleichung (2) in einer Funktion `f <- function(t, p)` welche eine `data.frame` mit den Spalten `t`, `x`, `x0`, `K`, `r` zurückgibt.

- i.) Plotten Sie die Funktion $x(t)$ auf dem Intervall $[0, 1]$ für $K = 0.5, r = 4$ und eine Schar von x_0 -Werten zwischen 0 und 1. Gruppieren Sie nach unterschiedlichen Werten von x_0 .
- ii.) Berücksichtigen Sie nun neben verschiedenen x_0 -Werten auch sechs unterschiedliche K -Werte zwischen 0.2 und 1 und plotten Sie die Kurvenscharen in unterschiedliche zu K gehörige Panels.
- iii.) Variieren Sie nun sowohl K als auch r und erzeugen Sie einen Grid-Plot.

Aufgabe 2: Verhulst-Dynamik

Implementieren Sie nun Gleichung (1) in einer Funktion `VH <- function(x, p)` welche ein `data.frame` mit den Spalten `x`, `dx` zurückgibt.

- i.) Plotten Sie $\dot{x}(t)$ auf dem Intervall $[0, 1]$ für Parameterwerte $K = 0.75, r = 4, x_0 = 1$.
- ii.) Fügen Sie dem Plot Pfeile hinzu, die auf der x -Achse liegen und von x nach $x + \dot{x}dt$ mit $dt = 0.1$ zeigen. Schauen Sie sich hierfür die Beispiele von `geom_segment()` an.

Aufgabe 3: Simulierte Daten mit Fehlerbalken

Betrachten Sie Gleichung (2) auf dem Intervall $[0, 1]$ für Parameterwerte $x_0 = 0.1, K = 0.5, r = 4$. Simulieren Sie Daten für $x(t)$, indem Sie die Funktion an 10 Zeitpunkten zwischen 0 und 1 auswerten und zu den Funktionswerten normalverteilte Zufallszahlen mit Mittelwert 0 und Varianz 0.01 addieren.

- i.) Erzeugen Sie sich ein `data.frame` für die Datenpunkte mit Spalten `t`, `x`, `sigma`, wobei in Spalte `sigma` die Wurzel aus der Varianz eingetragen wird.
- ii.) Erzeugen Sie sich als nächstes das analoge `data.frame` für die unverrauschte Lösung ausgewertet an 100 Zeitpunkten zwischen 0 und 1.
- iii.) Generieren Sie einen Plot mit Datenpunkten (`geom_point()`), Fehlerbalken (`geom_errorbar()`), Lösungskurve (`geom_line()`) und Fehlerband um die Lösungskurve (`geom_ribbon()`).