

### Aufgabe 1:

Die Werte des  $AR(1)$ -Prozesses sind gegeben durch

$$Y_t = \alpha Y_{t-1} + Z_t; Z_t \sim \mathcal{N}(0, \sigma^2)$$

- (a) Schreiben Sie eine R-Funktion `ar1.sim(n,alpha,sigma)`, die  $n$  Werte eines  $AR(1)$  Prozesses zurückgibt.

*Hinweis:* Benutzen Sie nicht `arima.sim`. Programmieren Sie selbst oder benutzen Sie die `filter`-Funktion. Verwenden Sie eine gewisse Anzahl `burnin` von Werten als zu verwerfende Burn-In Phase.

### Lösung:

```
R>library(ts)
R>mysim.ar1 <- function(n, alpha, sigma) {
+   burnin <- 1000
+   x <- rnorm(n + burnin, sd = sigma)
+   x <- filter(x, alpha, method = "recursive")
+   return(x[-c(1:burnin)])
+ }
R>mysim2.ar1 <- function(n, alpha, sigma) {
+   burnin <- 1000
+   x <- rnorm(n + burnin, sd = sigma)
+   for (i in 2:length(x)) {
+     x[i] <- alpha * x[i - 1] + x[i]
+   }
+   return(x[-c(1:burnin)])
+ }
```

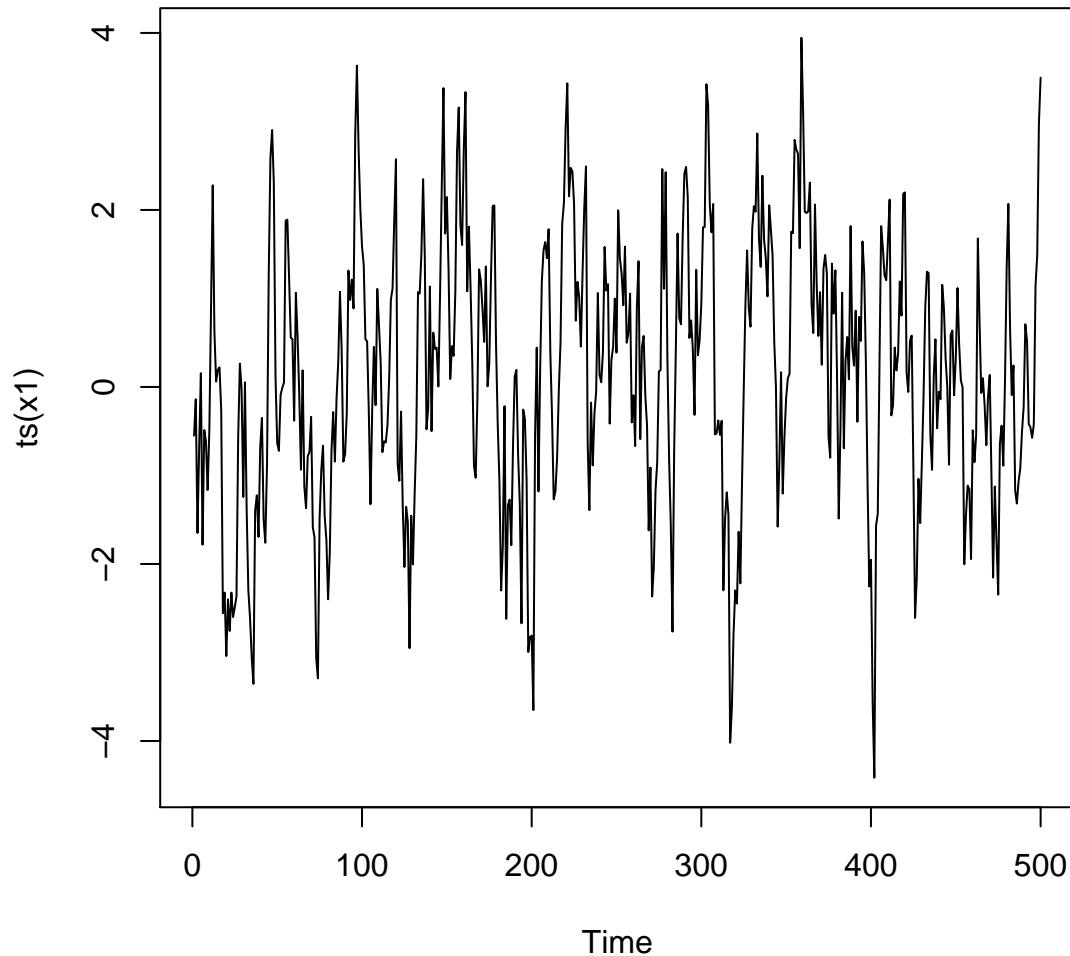
- (b) Plotten Sie  $n = 500$  Werte eines  $AR(1)$ -Prozesses mit  $\alpha = .8$  und  $\sigma^2 = 1$ .

### Lösung:

```
R>set.seed(1234)
R>x1 <- mysim.ar1(n = 500, alpha = 0.8, sigma = 1)
R>set.seed(1234)
R>x2 <- mysim2.ar1(n = 500, alpha = 0.8, sigma = 1)
R>sum(x1 - x2 > 1e-12) == 0

[1] TRUE

R>plot(ts(x1))
```



- (c) Schreiben Sie eine Funktion `my.acf(y, max.lag)` die das Korrelogramm einer Zeitreihe  $\mathbf{Y}$  bis zu einem maximalen Lag `max.lag` berechnet.

**Lösung:**

```
R>my.acf1 <- function(y, lagmax = 100) {
+   n <- length(y)
+   if (n < (lagmax + 1)) {
+     cat("max. Lag longer than time series.\n")
+     cat("max. Lag set to ", lagmax <- floor(n/2), ".\n")
+   }
+   y <- scale(y, scale = F)
+   M <- tcrossprod(y)
+   r <- rep(NA, lagmax + 1)
+   for (i in 1:(lagmax)) {
+     r[i] <- sum(diag(M))
+     M <- M[-1, ]
+   }
+   return(cbind(lag = 0:lagmax, acf = r/r[1]))
+ }
```

```

R>my.acf2 <- function(y, lagmax = 100) {
+   n <- length(y)
+   if (n < (lagmax + 1)) {
+     cat("max. Lag longer than time series.\n")
+     cat("max. Lag set to ", lagmax <- floor(n/2), ".\n")
+   }
+   y <- scale(y, scale = F)
+   r <- rep(NA, lagmax + 1)
+   r[1] <- t(y) %*% y
+   for (i in 1:(lagmax)) {
+     r[i + 1] <- t(y[-(1:i)]) %*% y[-((n - i + 1):n)]
+   }
+   return(cbind(lag = 0:lagmax, acf = r/r[1]))
+ }

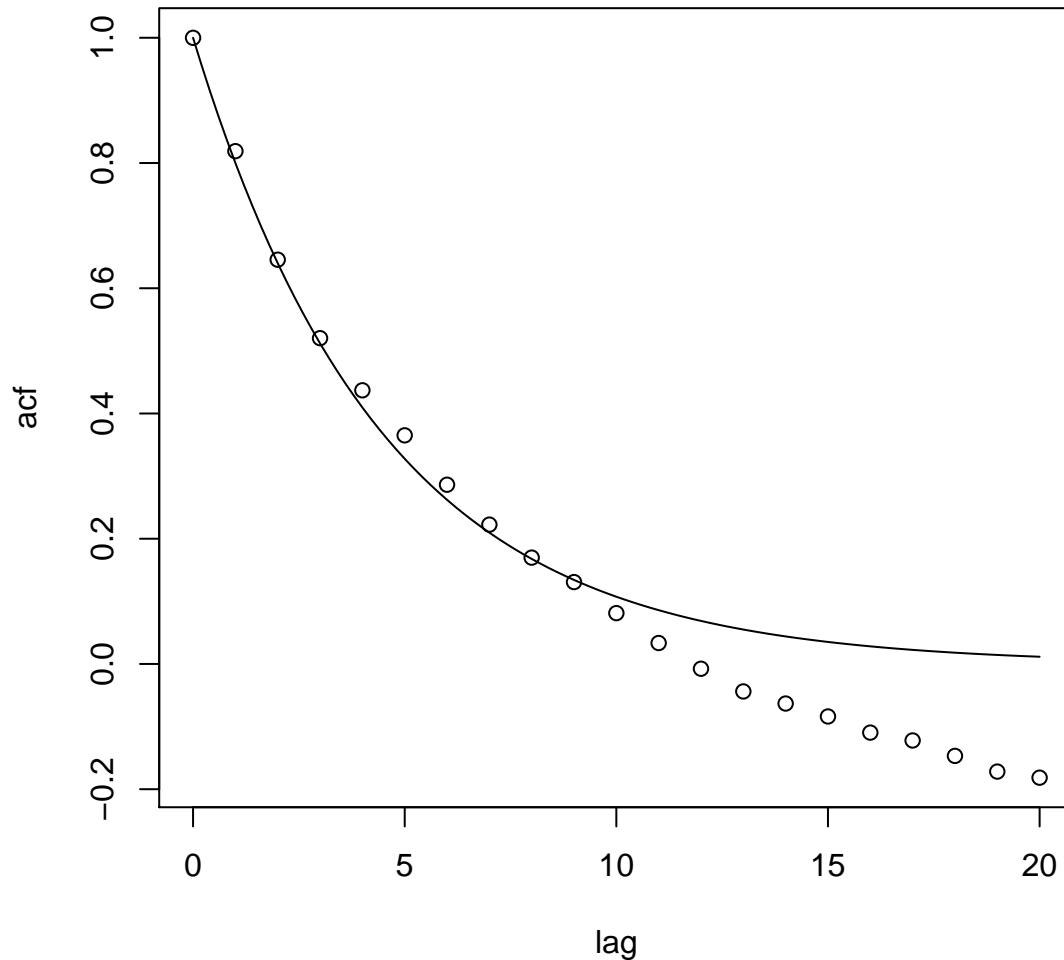
```

- (d) Plotten Sie die empirische und theoretische Autokorrelationsfunktion für die simulierten Daten. **Lösung:**

```

R>set.seed(34)
R>x1 <- mysim.ar1(n = 1000, alpha = 0.8, sigma = 1)
R>plot(my.acf2(x1, 20))
R>curve(0.8^x, from = 0, to = 20, add = T)

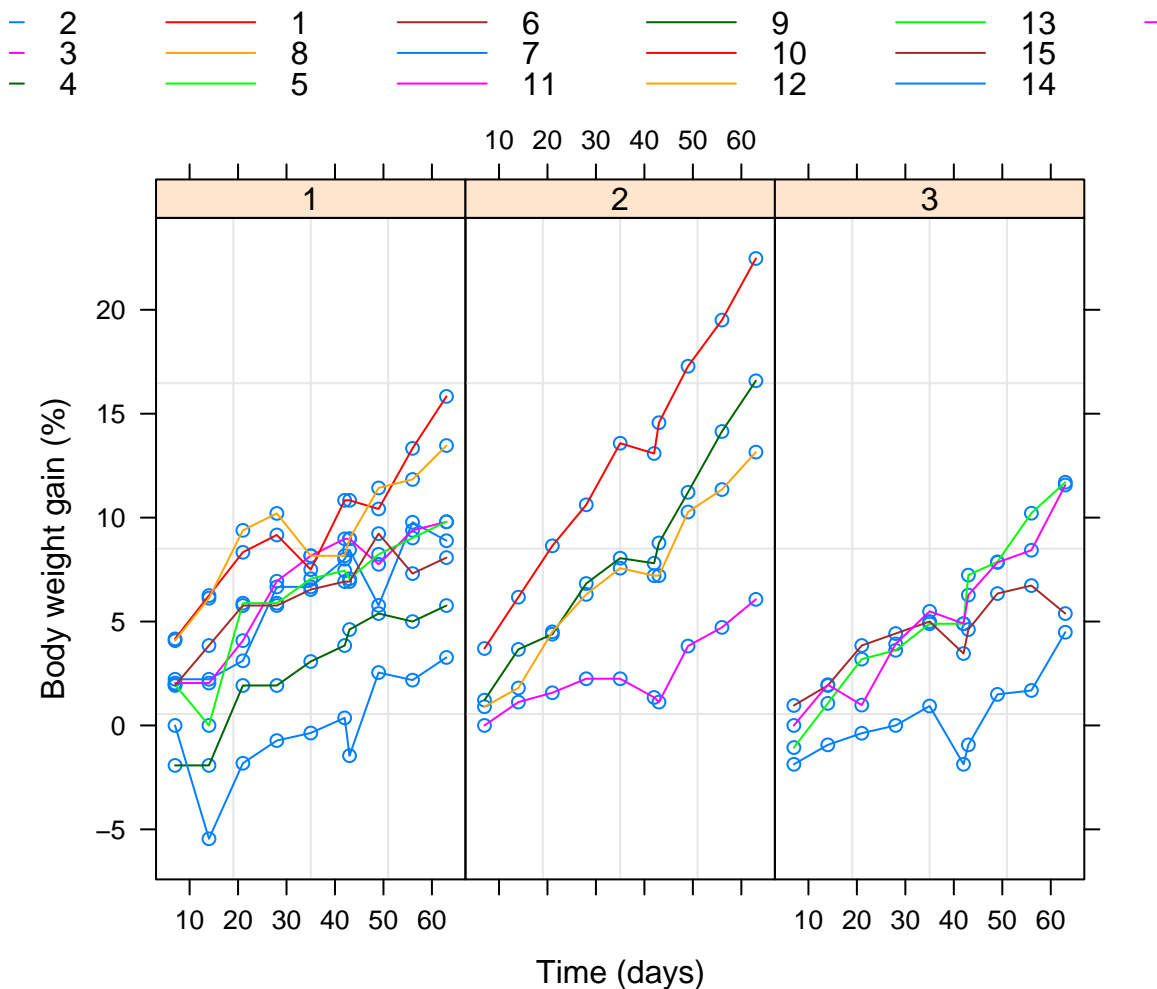
```



## Aufgabe 2:

Betrachten Sie den in `nlme` enthaltenen `BodyWeight` Datensatz. Dieser enthält Messungen des Körpergewichts von 16 Ratten über einen Zeitraum von 64 Tagen. Es gibt 3 Gruppen von Ratten, die unterschiedlich ernährt wurden. Benutzen Sie den folgenden Code um die Daten einzulesen und umzuwandeln.

```
R>require(nlme)
R>data(BodyWeight)
R>BW2 <- BodyWeight
R>weight1 <- BW2$weight[BW2$Time == 1]
R>weight1 <- rep(weight1, each = 11)
R>BW2$gain <- 100 * (BW2$weight - weight1)/weight1
R>BW2 <- BW2[BW2$Time != 1, ]
R>BW2$Time <- BW2$Time - 1
R>attr(BW2, "formula") <- gain ~ Time | Rat
R>attr(BW2, "units")$y <- "(%)"
R>attr(BW2, "labels")$y <- "Body weight gain"
R>print(plot(BW2, outer = ~Diet))
```



Benutzen Sie im Folgenden den Datensatz BW2.

- (a) Schätzen Sie ein gemischtes lineares Modell für die prozentuale Gewichtsveränderung mit unterschiedlichen Zeittrends für die 3 Experimentalgruppen sowie zufälligen Intercepts und Slopes. Überprüfen Sie den Modellfit.

**Lösung:**

```
R>m <- lme(gain ~ Time:Diet, random = ~Time | Rat, data = BW2)
R>summary(m)
```

Linear mixed-effects model fit by REML

Data: BW2

	AIC	BIC	logLik
	623.4122	647.811	-303.7061

Random effects:

Formula: ~Time | Rat

Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr
(Intercept)	2.18633216	(Intr)

Time 0.05670977 0.276  
Residual 1.16767745

Fixed effects: gain ~ Time:Diet

	Value	Std.Error	DF	t-value	p-value
(Intercept)	0.1110769	0.5862676	141	0.189464	0.85
Time:Diet1	0.1348972	0.0213349	141	6.322840	0.00
Time:Diet2	0.2180410	0.0301111	141	7.241217	0.00
Time:Diet3	0.1467468	0.0301111	141	4.873510	0.00

Correlation:

	(Intr)	Tm:Dt1	Tm:Dt2
Time:Diet1	0.090		
Time:Diet2	0.064	0.006	
Time:Diet3	0.064	0.006	0.004

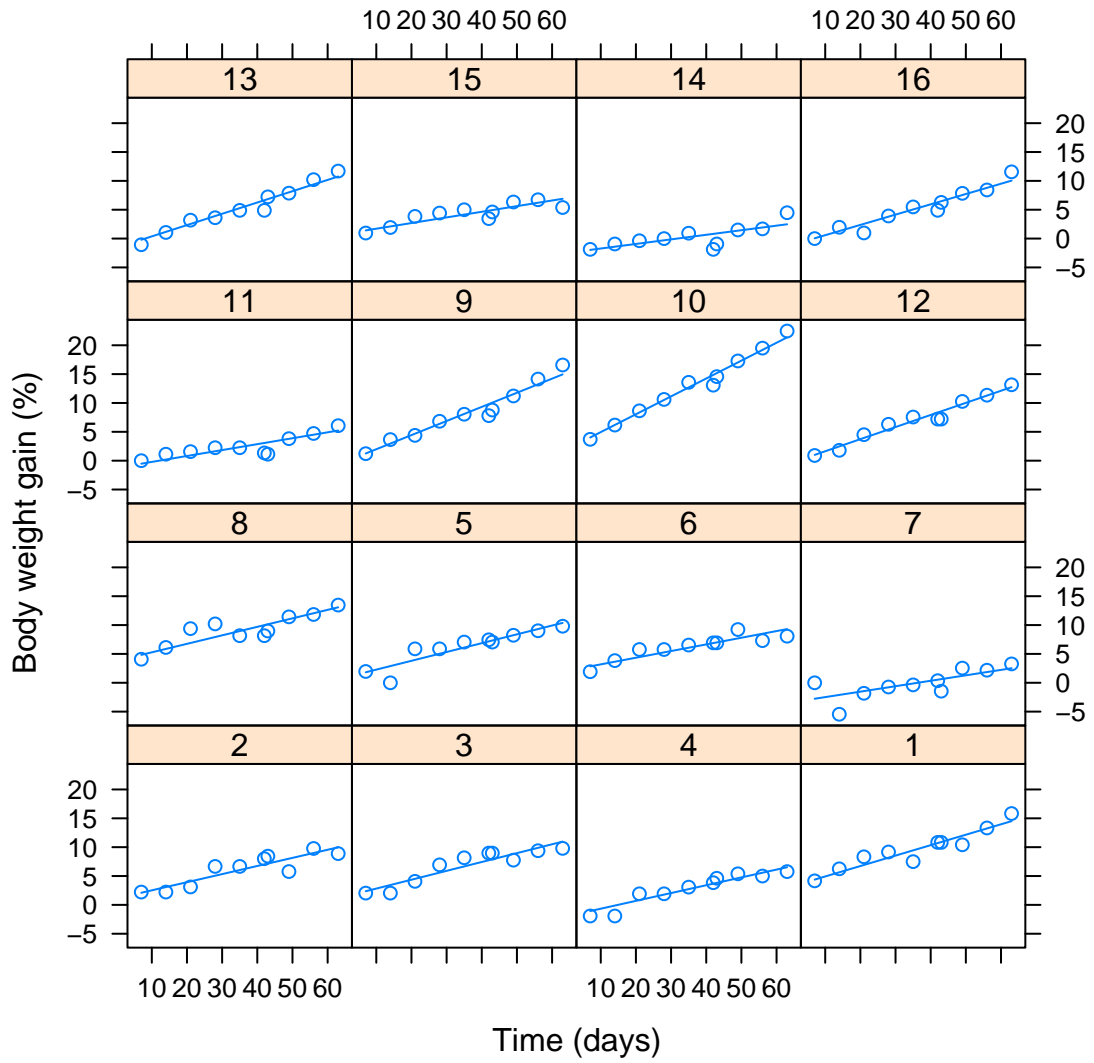
Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.8671647	-0.4923264	0.0922632	0.5945925	2.3698022

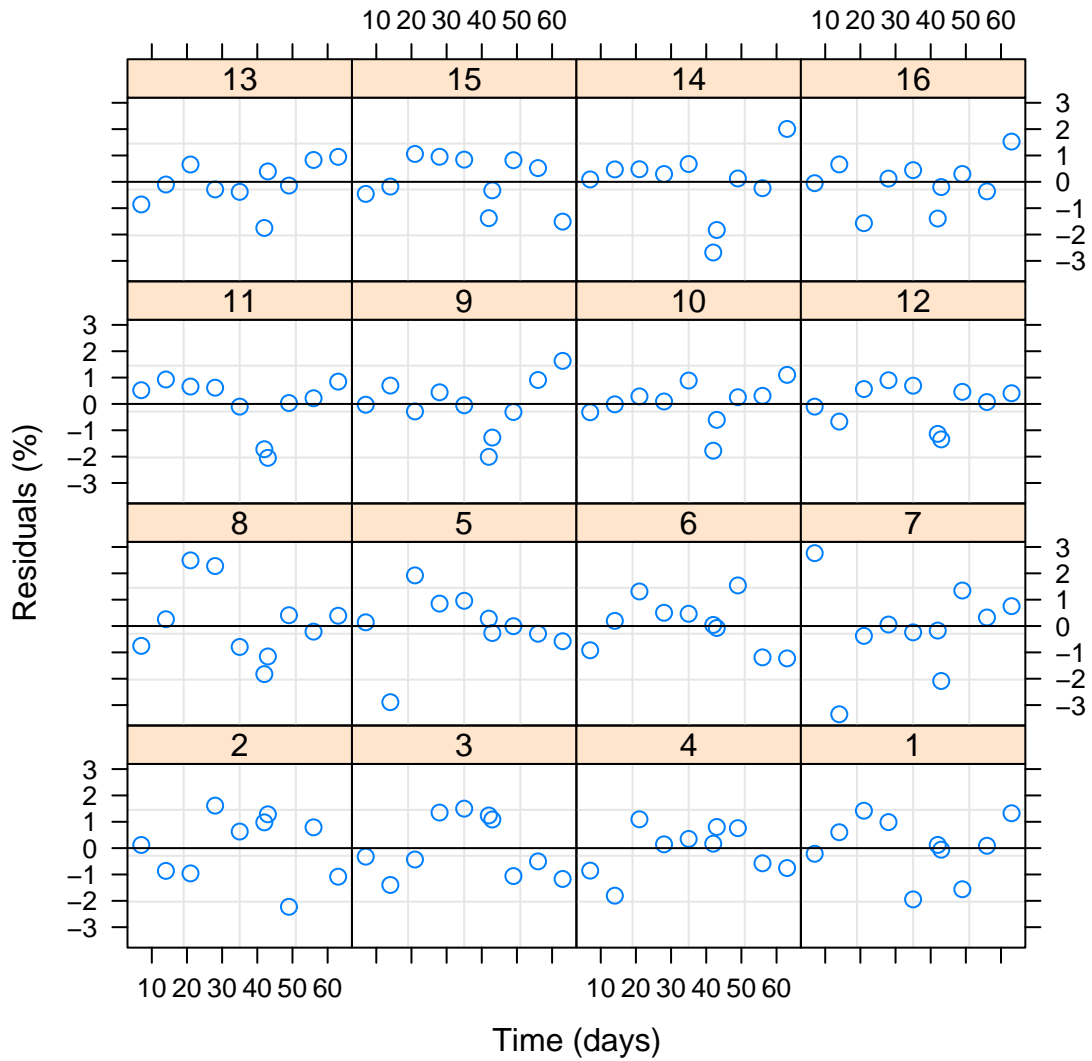
Number of Observations: 160

Number of Groups: 16

`R>print(plot(augPred(m)))`



```
R>print(plot(m, resid(.) ~ Time | Rat, abline = 0))
```



⇒ Autokorrelierte Residuen?

- (b) Berechnen und plotten Sie das empirische Semi-Variogramm für das Modell aus a) (→ `Variogram()`) bis zu einem maximalen zeitlichen Abstand von 42 Tagen. Interpretieren Sie das Ergebnis.

**Lösung:**

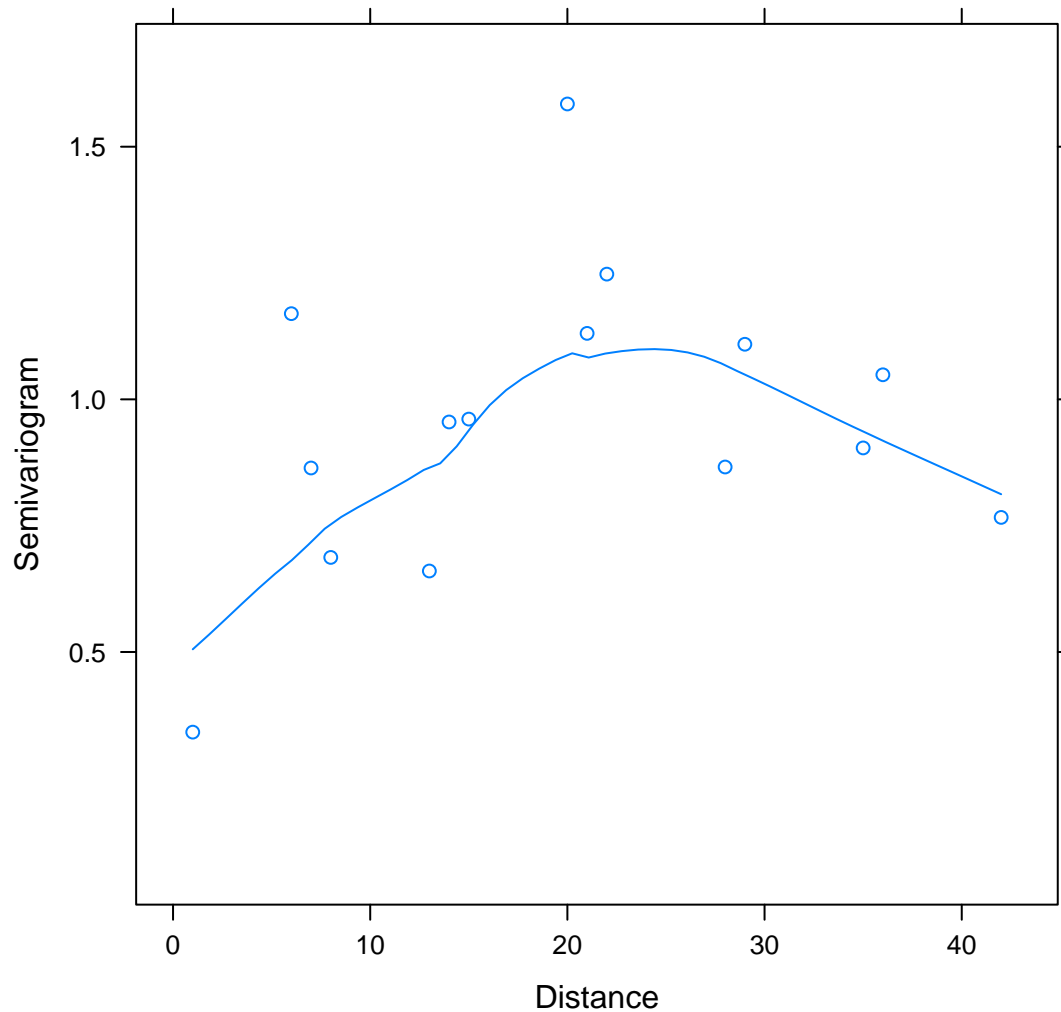
```
R>(Vm <- Variogram(m, form = ~Time, maxDist = 42))
```

	variog	dist	n.pairs
1	0.3414325	1	16
2	1.1696602	6	16
3	0.8641971	7	128
4	0.6871976	8	16
5	0.6603344	13	16
6	0.9552710	14	112
7	0.9609706	15	16
8	1.5846495	20	16
9	1.1306445	21	96

10	1.2478729	22	16
11	0.8661422	28	80
12	1.1090771	29	16
13	0.9038734	35	64
14	1.0486659	36	16
15	0.7664300	42	48

```
R>print(plot(Vm, ylim = c(0, 1.1 * max(Vm$variog)), main = "Semi-Variogram"))
```

### Semi-Variogram



⇒

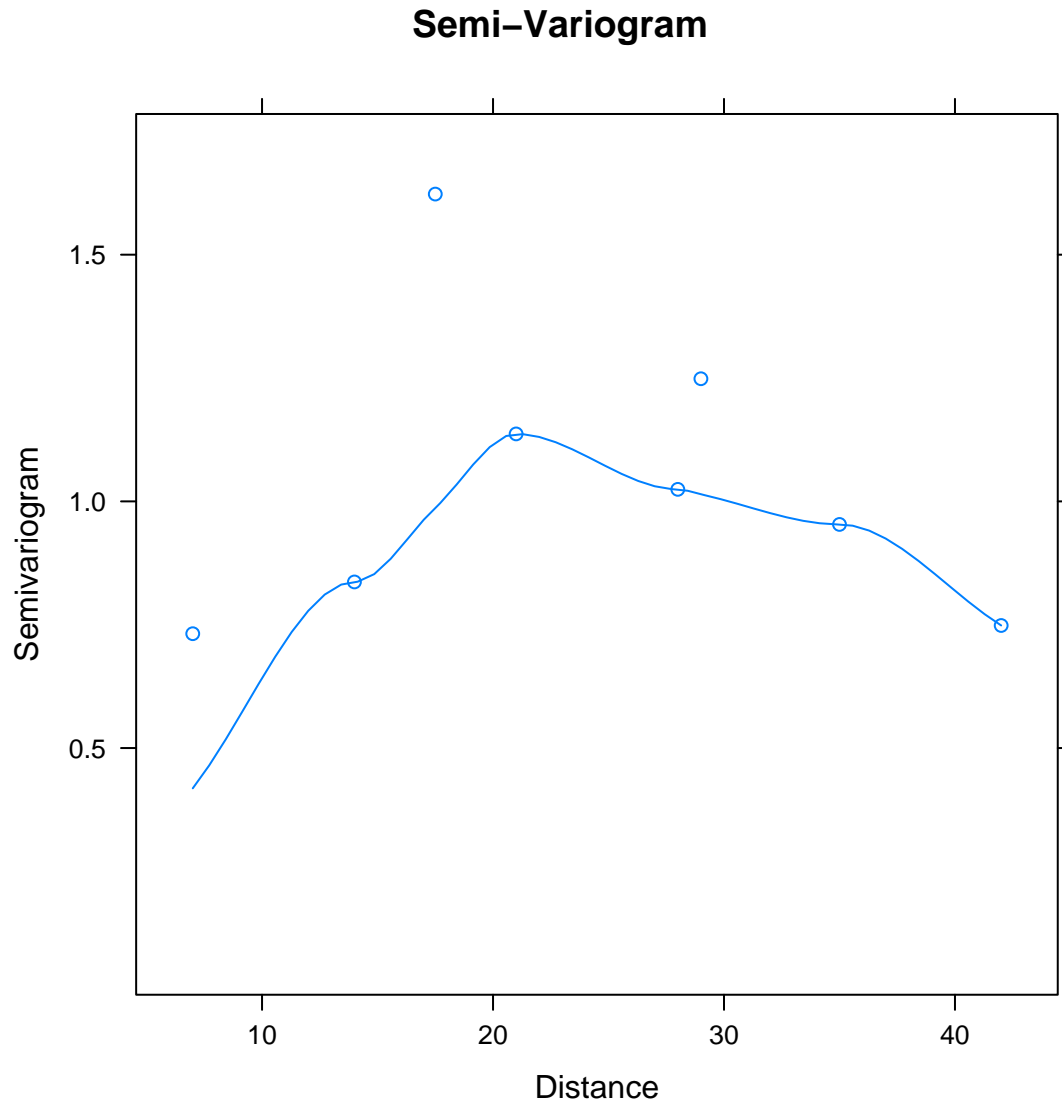
Variogramm bestätigt Vermutung dass (positive) Autokorrelation vorliegt: deutlicher Anstieg der Variogrammwerte für Abstände bis 20 Tage, dann abflachend. Muster wird deutlicher wenn man durch die Argumente `nint` und `robust` die Schätzung robuster gegen Ausreißer macht:

```
R>(Vm2 <- Variogram(m, form = ~Time, maxDist = 42, nint = 10, robust = T))
```

	variog	dist	n.pairs
1	0.7318217	7.0	144
2	0.8367418	14.0	144
3	1.6229177	17.5	32
4	1.1368036	21.0	96
5	1.0244253	28.0	96

```
6 1.2485798 29.0    16
7 0.9533221 35.0    64
8 0.7484582 42.0    64
```

```
R>print(plot(Vm2, ylim = c(0, 1.1 * max(Vm2$variog)), main = "Semi-Variogram"))
```



- (c) Erweitern Sie das Modell aus a) um eine Exponential-Korrelationstruktur der Residuen in `Time`. Vergleichen Sie die Modelle. Vergleichen Sie das Variogramm des Modells mit Exponential-Korrelationstruktur mit dem eines Modells mit Gauss-Korrelation.

**Lösung:**

```
R>mExp <- update(m, corr = corExp(form = ~Time))
R>summary(mExp)
```

Linear mixed-effects model fit by REML

Data: BW2

	AIC	BIC	logLik
	608.7005	636.1492	-295.3502

```

Random effects:
Formula: ~Time | Rat
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev   Corr
(Intercept) 2.08511164 (Intr)
Time         0.05533796 0.354
Residual    1.20906774

```

Correlation Structure: Exponential spatial correlation

```

Formula: ~Time | Rat
Parameter estimate(s):
  range
3.56704

```

```

Fixed effects: gain ~ Time:Diet
          Value Std.Error DF t-value p-value
(Intercept) 0.1023930 0.5745739 141 0.178207 0.8588
Time:Diet1  0.1340719 0.0212586 141 6.306722 0.0000
Time:Diet2  0.2229006 0.0299842 141 7.433931 0.0000
Time:Diet3  0.1515515 0.0299842 141 5.054374 0.0000

```

```

Correlation:
          (Intr) Tm:Dt1 Tm:Dt2
Time:Diet1 0.103
Time:Diet2 0.073 0.008
Time:Diet3 0.073 0.008 0.005

```

Standardized Within-Group Residuals:

```

          Min          Q1          Med          Q3          Max
-3.01438709 -0.45070408 0.01349633 0.52482793 2.11961204

```

Number of Observations: 160

Number of Groups: 16

```

R>mGauss <- update(m, corr = corGaus(form = ~Time))
R>summary(mGauss)

```

Linear mixed-effects model fit by REML

```

Data: BW2
      AIC      BIC    logLik
610.8077 638.2564 -296.4038

```

Random effects:

```

Formula: ~Time | Rat
Structure: General positive-definite, Log-Cholesky parametrization
          StdDev   Corr
(Intercept) 2.18393439 (Intr)
Time         0.05635699 0.271
Residual    1.15641558

```

Correlation Structure: Gaussian spatial correlation

```

Formula: ~Time | Rat
Parameter estimate(s):
  range
1.641663

```

Fixed effects: gain ~ Time:Diet

	Value	Std.Error	DF	t-value	p-value
(Intercept)	0.1183269	0.5849860	141	0.202273	0.84
Time:Diet1	0.1348862	0.0212141	141	6.358314	0.00
Time:Diet2	0.2210955	0.0299434	141	7.383774	0.00
Time:Diet3	0.1494164	0.0299434	141	4.989954	0.00

Correlation:

	(Intr)	Tm:Dt1	Tm:Dt2
Time:Diet1	0.088		
Time:Diet2	0.062	0.005	
Time:Diet3	0.062	0.005	0.004

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-2.94927947	-0.47118522	0.02935903	0.54514708	2.34875528

Number of Observations: 160

Number of Groups: 16

R>anova(m, mExp)

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
m	1	8	623.4122	647.8110	-303.7061			
mExp	2	9	608.7005	636.1492	-295.3502	1 vs 2	16.71174	<.0001

R>anova(m, mGauss)

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
m	1	8	623.4122	647.8110	-303.7061			
mGauss	2	9	610.8077	638.2564	-296.4038	1 vs 2	14.60453	1e-04

R>anova(mGauss, mExp)

	Model	df	AIC	BIC	logLik
mGauss	1	9	610.8077	638.2564	-296.4038
mExp	2	9	608.7005	636.1492	-295.3502

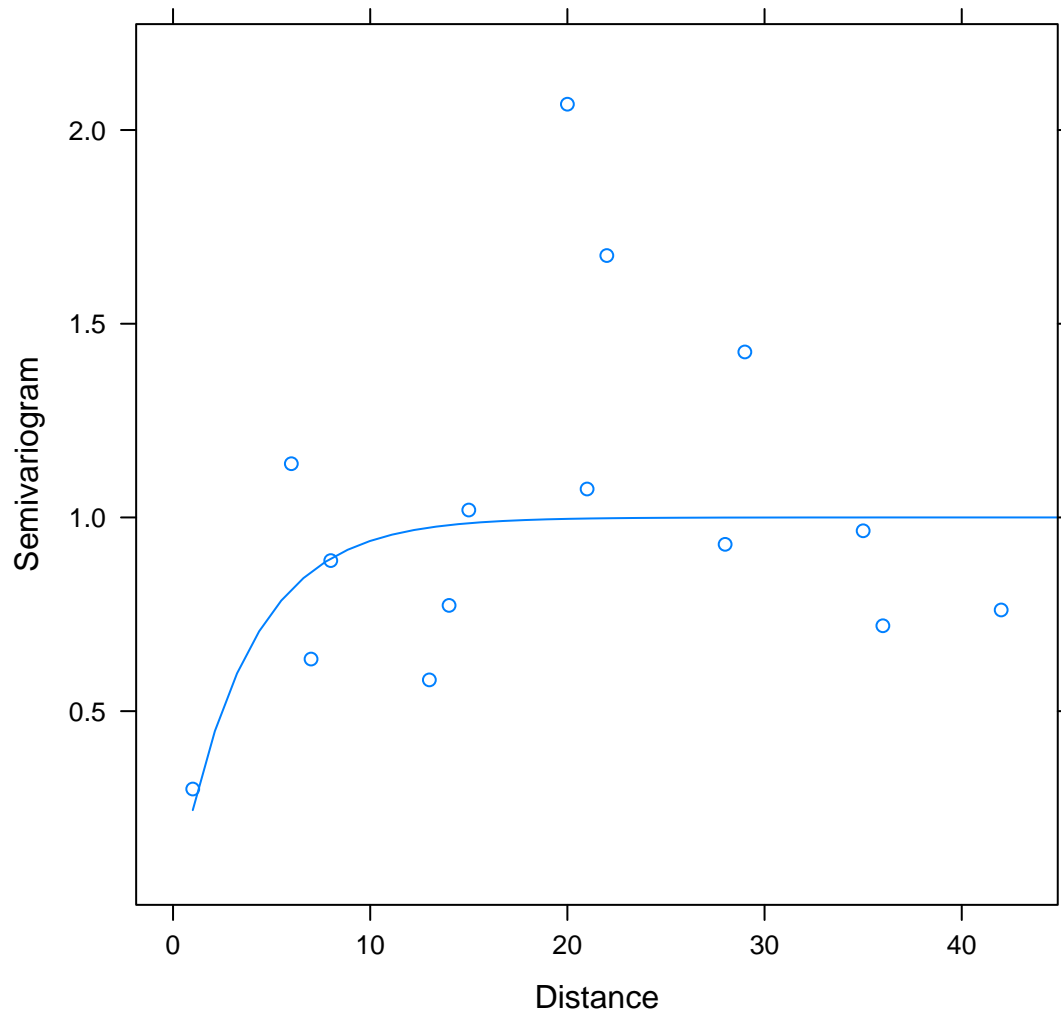
⇒ bester Fit für Exponential-Korrelationstruktur.

R>(VmE <- Variogram(mExp, form = ~Time, maxDist = 42, robust = T))

	variog	dist	n.pairs
1	0.2990407	1	16
2	1.1387424	6	16
3	0.6344832	7	128
4	0.8886967	8	16
5	0.5807070	13	16
6	0.7730133	14	112
7	1.0191890	15	16
8	2.0666996	20	16
9	1.0734388	21	96
10	1.6760730	22	16
11	0.9305821	28	80
12	1.4271597	29	16
13	0.9655721	35	64
14	0.7204433	36	16
15	0.7612143	42	48

```
R>print(plot(VmE, ylim = c(0, 1.1 * max(VmE$variog)), main = "Semi-Variogram Exponential-Correlation"))
```

### Semi-Variogram Exponential-Correlation



```
R>(VmG <- Variogram(mGauss, form = ~Time, maxDist = 42, robust = T))
```

	variog	dist	n.pairs
1	0.3253557	1	16
2	1.2798964	6	16
3	0.6842792	7	128
4	0.9473809	8	16
5	0.6711715	13	16
6	0.8439758	14	112
7	1.0690056	15	16
8	2.2574482	20	16
9	1.1381808	21	96
10	1.8499505	22	16
11	0.9525208	28	80
12	1.3615040	29	16
13	0.9896790	35	64
14	0.8377847	36	16
15	0.6968921	42	48

```
R>print(plot(VmG, ylim = c(0, 1.1 * max(VmG$variog)), main = "Semi-Variogram Gauss-Correlation"))
```

# Semi-Variogram Gauss-Correlation

