

Aufgabe 1:

Eine Möglichkeit, die Verteilung der Likelihood-Quotienten-Statistik unter der Nullhypothese zu überprüfen, ist es, eine Simulation durchzuführen. Die Funktion `simulate.lme` überprüft diese Verteilung für zwei genestete Modelle:

- (a) Betrachten Sie die `Orthodont`-Daten. Fitten Sie ein gemischtes Modell mit Zeittrend und zufälligem Intercept und ein gemischtes Modell mit zusätzlichem zufälligen subjektspezifischem Zeittrend. Testen Sie mit der Funktion `anova()` die Hypothese, dass der zufällige subjektspezifische Zeittrend nicht benötigt wird.

Lösung:

```
R>library(nlme)
R>m0 <- lme(distance ~ Sex * I(age - 11), random = ~1 | Subject,
+ data = Orthodont)
R>m1 <- update(m0, random = ~I(age - 11) | Subject)
R>(lr01 <- anova(m0, m1))
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
m0	1	6	445.7572	461.6236	-216.8786			
m1	2	8	448.5817	469.7368	-216.2908	1 vs 2	1.175588	0.5556

- (b) Da sich die Nullhypothese auf dem Rand des Parameterraumes befindet, ist der LQ-Test konservativ. Stram und Lee (1994) empfehlen als approximative Verteilung der LQ-Statistik unter H_0 die Mischung aus zwei χ^2 -Verteilungen, hier $0.5\chi_1^2 + 0.5\chi_2^2$. Berechnen Sie im obigen Beispiel den daraus folgenden p-Wert und vergleichen Sie mit dem Ergebnis aus Teil (a).

Lösung:

```
R>pchismix <- function(q, df = c(1, 2)) {
+ sum(0.5 * pchisq(rep(q, 2), df))
+ }
R>1 - pchismix(lr01$L.Ratio[2], df = c(1, 2))
```

```
[1] 0.4169038
```

```
R>lr01$"p-value"[2]
```

```
[1] 0.5555516
```

Die Mischung $0.5\chi_1^2 + 0.5\chi_2^2$ wird hier benutzt weil die Null-Hypothese 2 Restriktionen an den Parametervektor impliziert, nämlich dass die Varianz des Random Slope und seine Korrelation mit dem Random Intercept 0 sind. Die erste Restriktion liegt am Rand des Parameterraumes, deswegen greift die übliche Asymptotik, die zu einer χ_2^2 -Verteilung führen würde, hier nicht.

- (c) Benutzen Sie die Funktion `simulate.lme()` um je 1000 Simulationen der beiden Modelle zu erstellen. Plotten Sie das resultierende Objekt und interpretieren Sie die Ergebnisse.

Lösung:

```
R>n.sim <- 1000
R>lr01.sim <- simulate.lme(m0, m2 = m1, nsim = n.sim)
R>print(plot(lr01.sim, df = c(1, 2)))
R>lrt01 <- 2 * (lr01.sim$alt$REML[, "logLik"] - lr01.sim>null$REML[,
+ "logLik"])
R>(p01.sim <- mean(lrt01 > lr01$L.Ratio[2]))

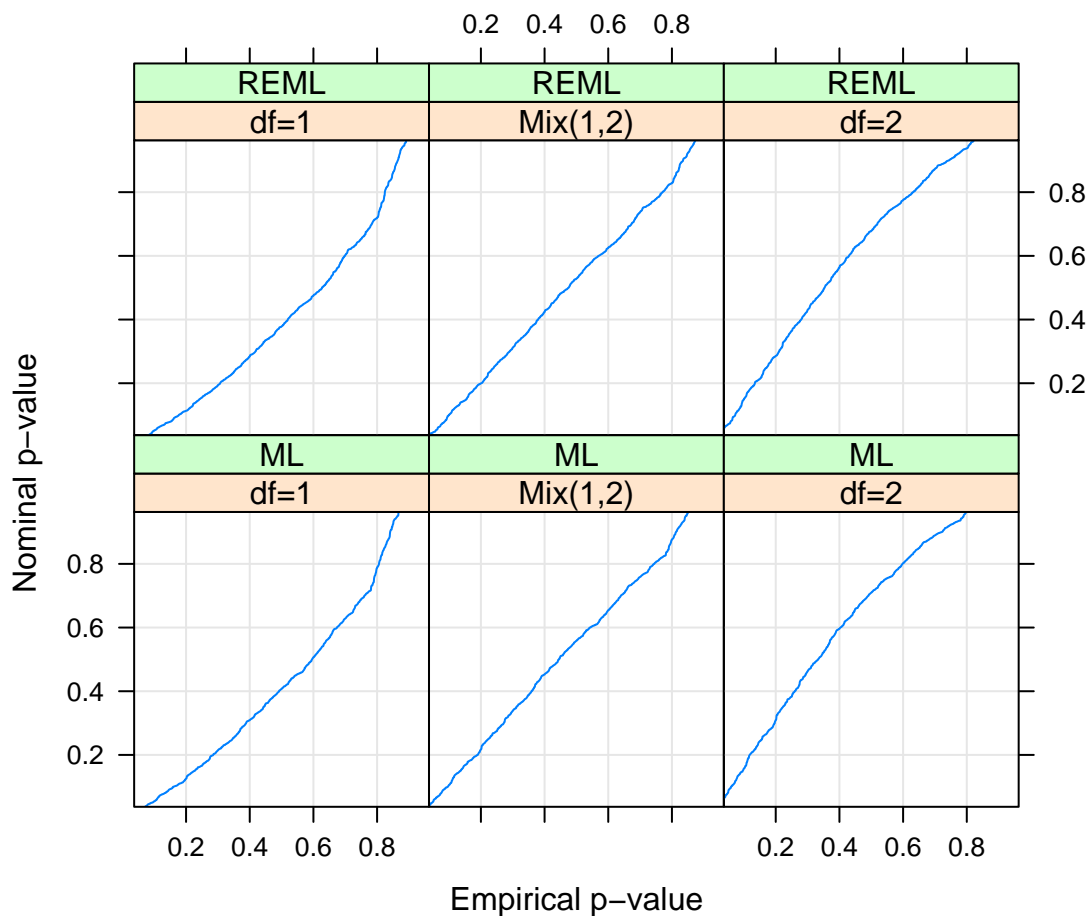
[1] 0.395

R>1 - pchisqmix(lr01$L.Ratio[2], df = c(1, 2))

[1] 0.4169038

R>lr01$"p-value"[2]

[1] 0.5555516
```



`simulate.lme()` führt einen parametrischen Bootstrap des Null-Modells durch, d.h. es werden `nsim` Responsevektoren aus der multivariaten Normalverteilung, die durch das geschätzte Null-Modell spezifiziert wird, erzeugt und jeweils Modelle unter der Null- und Alternativhypothese mit ML und REML geschätzt.

Obiger Plot vergleicht die Verteilung der resultierenden 'empirischen' p-Werte, die aus der Verteilung der simulierten (R)LR-Statistiken bestimmt werden, mit Verteilungen von p-Werten aus in Frage kommenden theoretischen Verteilungen, genauer: χ^2 -Verteilungen und 50 : 50-Mischungen von χ^2 -Verteilungen.

Für ML- und REML-Schätzung zeigt sich hier ein ähnliches Bild: χ_1^2 ist anti-konservativ (simulierte/empirische p-Werte größer als theoretische/nominale), χ_2^2 deutlich konservativ (simulierte p-Werte kleiner als theoretische), die $0.5\chi_1^2 : 0.5\chi_2^2$ -Mischung entspricht der simulierten Verteilung recht gut.

(d) Diskutieren Sie die Ergebnisse des folgenden Codes.

Hinweis: Die Verteilung des LR-Tests auf eine Varianzkomponente σ_b^2 (d.h. $H_0 : \sigma_b^2 = 0$), die unkorreliert mit den anderen zufälligen Effekten ist, kann durch die Mischung einer Punktmasse in 0 und der χ_1^2 -Verteilung approximiert werden.

```
R>m0 <- lme(distance ~ Sex * I(age - 11), random = ~1 | Subject,
+ data = Orthodont)
R>m1 <- update(m0, random = ~I(age - 11) | Subject)
R>m2 <- update(m0, random = list(Subject = pdDiag(~I(age - 11))))
R>summary(m0)
R>summary(m1)
R>summary(m2)

R>(lr21 <- anova(m2, m1))
R>lr21.sim <- simulate.lme(m2, m2 = m1, nsim = n.sim)
R>print(plot(lr21.sim, df = c(1, 2)))
R>lrt21 <- 2 * (lr21.sim$alt$REML[, "logLik"] - lr21.sim$null$REML[,
+ "logLik"])
R>(p21.sim <- mean(lrt21 > lr21$L.Ratio[2]))
R>1 - pchisq(lr21$L.Ratio[2], df = 1)

R>(lr02 <- anova(m0, m2))
R>lr02.sim <- simulate.lme(m0, m2 = m2, nsim = n.sim)
R>print(plot(lr02.sim, df = c(0, 1)))
R>lrt02 <- 2 * (lr02.sim$alt$REML[, "logLik"] - lr02.sim$null$REML[,
+ "logLik"])
R>(p02.sim <- mean(lrt02 > lr02$L.Ratio[2]))
R>1 - 0.5 * pchisq(lr02$L.Ratio[2], df = 1) - 0.5
```

Lösung:

```
R>m0 <- lme(distance ~ Sex * I(age - 11), random = ~1 | Subject,
+ data = Orthodont)
R>m1 <- update(m0, random = ~I(age - 11) | Subject)
R>m2 <- update(m0, random = list(Subject = pdDiag(~I(age - 11))))
R>summary(m0)
```

Linear mixed-effects model fit by REML

Data: Orthodont

AIC	BIC	logLik
445.7572	461.6236	-216.8786

Random effects:

Formula: ~1 | Subject

(Intercept) Residual

StdDev: 1.816214 1.386382

Fixed effects: distance ~ Sex * I(age - 11)

	Value	Std.Error	DF	t-value	p-value
(Intercept)	24.968750	0.4860008	79	51.37595	0.0000
SexFemale	-2.321023	0.7614168	25	-3.04829	0.0054
I(age - 11)	0.784375	0.0775011	79	10.12082	0.0000
SexFemale:I(age - 11)	-0.304830	0.1214209	79	-2.51052	0.0141

Correlation:

	(Intr)	SexFml	I(-11)
SexFemale		-0.638	
I(age - 11)	0.000	0.000	
SexFemale:I(age - 11)	0.000	0.000	-0.638

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-3.59804400	-0.45461690	0.01578365	0.50244658	3.68620792

Number of Observations: 108

Number of Groups: 27

R>summary(m1)

Linear mixed-effects model fit by REML

Data: Orthodont

AIC	BIC	logLik
448.5817	469.7368	-216.2908

Random effects:

Formula: ~I(age - 11) | Subject

Structure: General positive-definite, Log-Cholesky parametrization

StdDev Corr

(Intercept) 1.8303268 (Intr)

I(age - 11) 0.1803454 0.206

Residual 1.3100397

Fixed effects: distance ~ Sex * I(age - 11)

	Value	Std.Error	DF	t-value	p-value
(Intercept)	24.968750	0.4860007	79	51.37595	0.0000
SexFemale	-2.321023	0.7614168	25	-3.04829	0.0054
I(age - 11)	0.784375	0.0859995	79	9.12069	0.0000
SexFemale:I(age - 11)	-0.304830	0.1347353	79	-2.26243	0.0264

Correlation:

	(Intr)	SexFml	I(-11)
SexFemale		-0.638	
I(age - 11)	0.102	-0.065	

```
SexFemale:I(age - 11) -0.065 0.102 -0.638
```

```
Standardized Within-Group Residuals:
```

```
      Min      Q1      Med      Q3      Max
-3.168078328 -0.385939104 0.007103934 0.445154638 3.849463324
```

```
Number of Observations: 108
```

```
Number of Groups: 27
```

```
R>summary(m2)
```

```
Linear mixed-effects model fit by REML
```

```
Data: Orthodont
```

```
      AIC      BIC    logLik
446.8426 465.3533 -216.4213
```

```
Random effects:
```

```
Formula: ~I(age - 11) | Subject
```

```
Structure: Diagonal
```

```
(Intercept) I(age - 11) Residual
```

```
StdDev: 1.830327 0.1803455 1.310040
```

```
Fixed effects: distance ~ Sex * I(age - 11)
```

```
      Value Std.Error DF t-value p-value
(Intercept) 24.968750 0.4860008 79 51.37595 0.0000
SexFemale -2.321023 0.7614168 25 -3.04829 0.0054
I(age - 11) 0.784375 0.0859995 79 9.12069 0.0000
SexFemale:I(age - 11) -0.304830 0.1347353 79 -2.26243 0.0264
```

```
Correlation:
```

```
      (Intr) SexFml I(-11)
SexFemale -0.638
I(age - 11) 0.000 0.000
SexFemale:I(age - 11) 0.000 0.000 -0.638
```

```
Standardized Within-Group Residuals:
```

```
      Min      Q1      Med      Q3      Max
-3.06658939 -0.39982538 0.02559617 0.43693650 3.85940303
```

```
Number of Observations: 108
```

```
Number of Groups: 27
```

```
R>(lr21 <- anova(m2, m1))
```

```
      Model df      AIC      BIC    logLik    Test L.Ratio p-value
m2      1 7 446.8426 465.3533 -216.4213
m1      2 8 448.5817 469.7368 -216.2908 1 vs 2 0.260933 0.6095
```

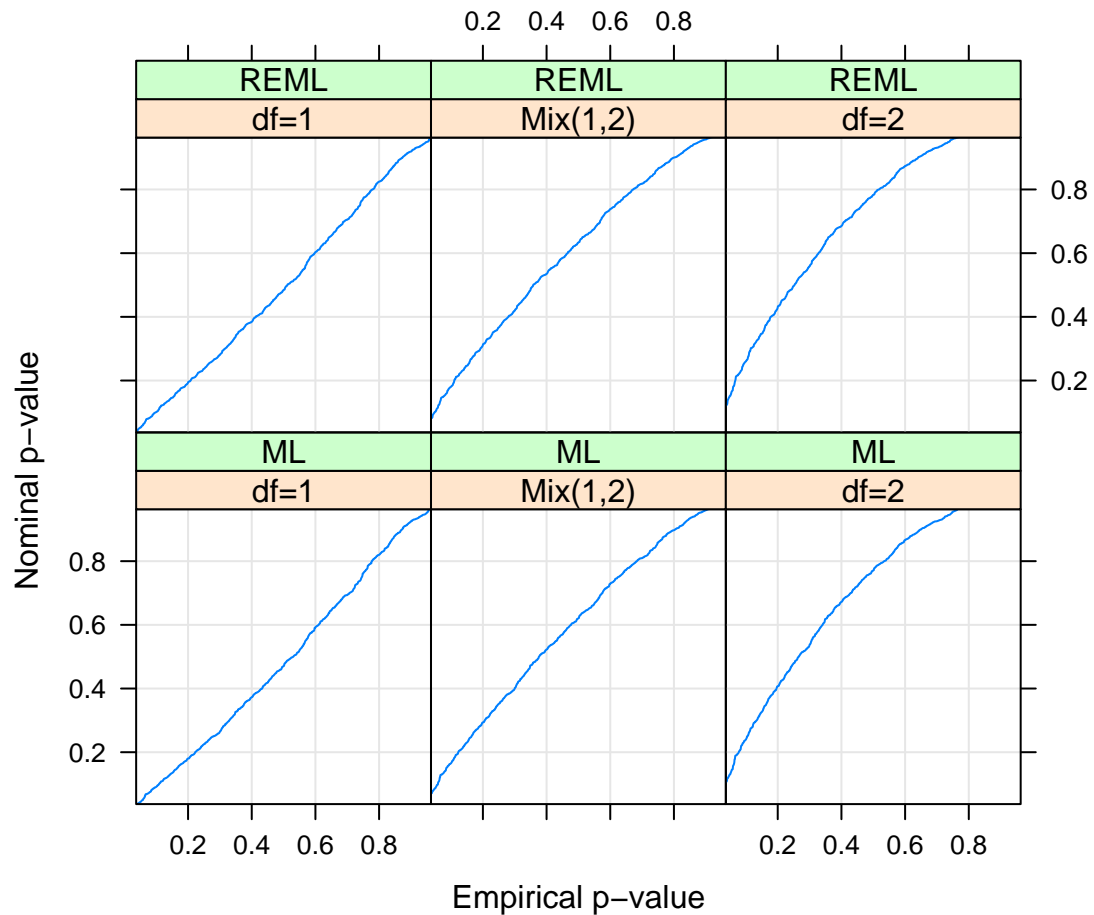
```
R>lr21.sim <- simulate.lme(m2, m2 = m1, nsim = n.sim)
```

```
R>print(plot(lr21.sim, df = c(1, 2)))
```

```
R>lrt21 <- 2 * (lr21.sim$alt$REML[, "logLik"] - lr21.sim$null$REML[,
+ "logLik"])
```

```
R>(p21.sim <- mean(lrt21 > lr21$L.Ratio[2]))
```

```
[1] 0.607
```



⇒ Für diesen Test ob Random Intercept und Random Slope korreliert sind ist die übliche Verteilungsannahme des LR-Tests gültig, da die Nullhypothese (Korrelation=0) nicht am Rand des Parameterraums liegt.

```
R>(lr02 <- anova(m0, m2))
```

Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
m0	1	6 445.7572	461.6236	-216.8786			
m2	2	7 446.8426	465.3533	-216.4213	1 vs 2	0.9146547	0.3389

```
R>lr02.sim <- simulate.lme(m0, m2 = m2, nsim = n.sim)
```

```
R>print(plot(lr02.sim, df = c(0, 1)))
```

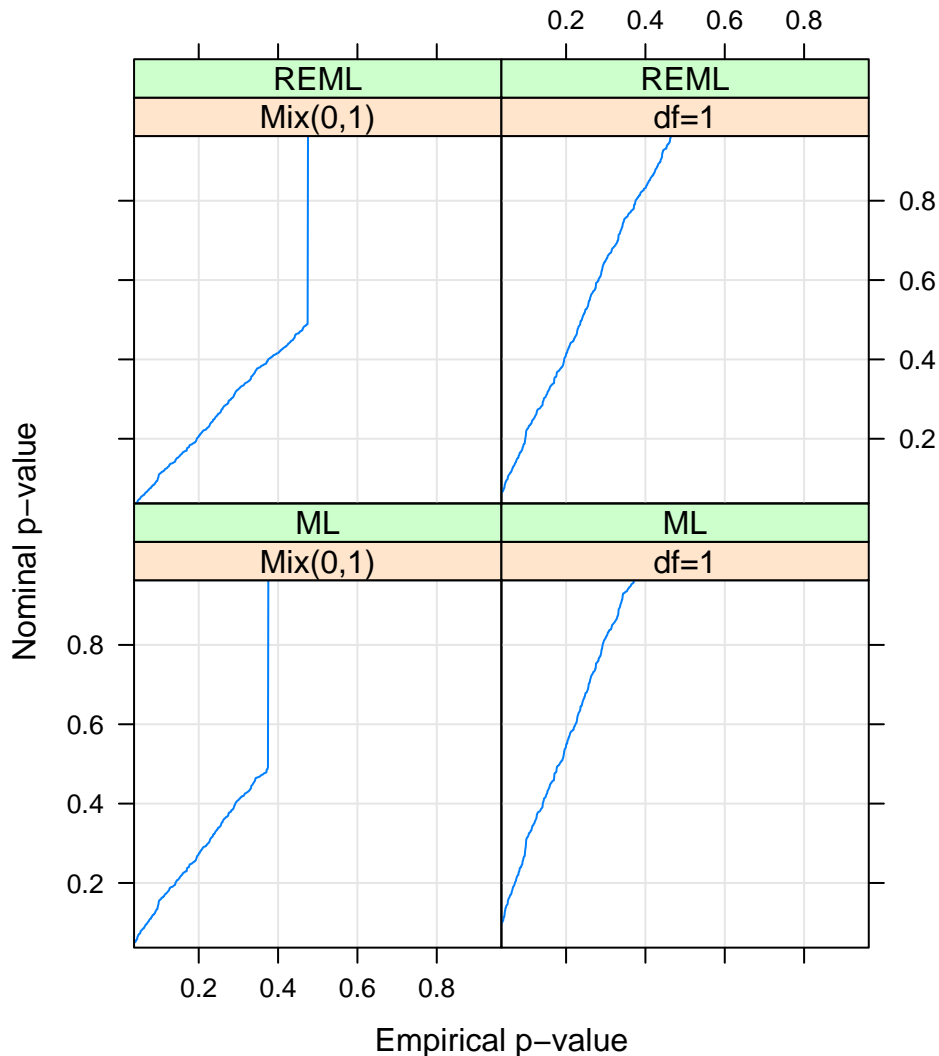
```
R>lrt02 <- 2 * (lr02.sim$alt$REML[, "logLik"] - lr02.sim>null$REML[,  
+ "logLik"])
```

```
R>(p02.sim <- mean(lrt02 > lr02$L.Ratio[2]))
```

```
[1] 0.167
```

```
R>1 - 0.5 * pchisq(lr02$L.Ratio[2], df = 1) - 0.5
```

```
[1] 0.1694412
```



⇒ Für diesen Test ob der Random Slope benötigt wird ist die übliche Verteilungsannahme des LR-Tests ungültig, da die Nullhypothese (Varianz=0) am Rand des Parameterraums liegt. Eine – in diesem Fall recht gute – mögliche Approximation an die Verteilung der REML-Teststatistik unter der Null-Hypothese ist die Mischung einer Punktmasse in 0 und der χ^2_1 -Verteilung.

In vielen anderen Beispielen kann auch diese Approximation sehr schlecht sein und zu sehr konservativen Tests führen, v.a. in stark unbalancierten Designs mit wenigen Gruppen oder wenn zusätzliche sehr kleine andere Varianzkomponenten im Modell sind. Die Self-Liang-Approximationen funktionieren deutlich besser für Tests basierend auf dem Quotienten der Restricted Likelihoods, wie man auch in obigem Plot erkennen kann.

Aufgabe 2:

Betrachten Sie den folgenden Ausschnitt aus der Zeitreihe `wertpap` der Zinsen deutscher Wertpapiere (zu lesen von links nach rechts).

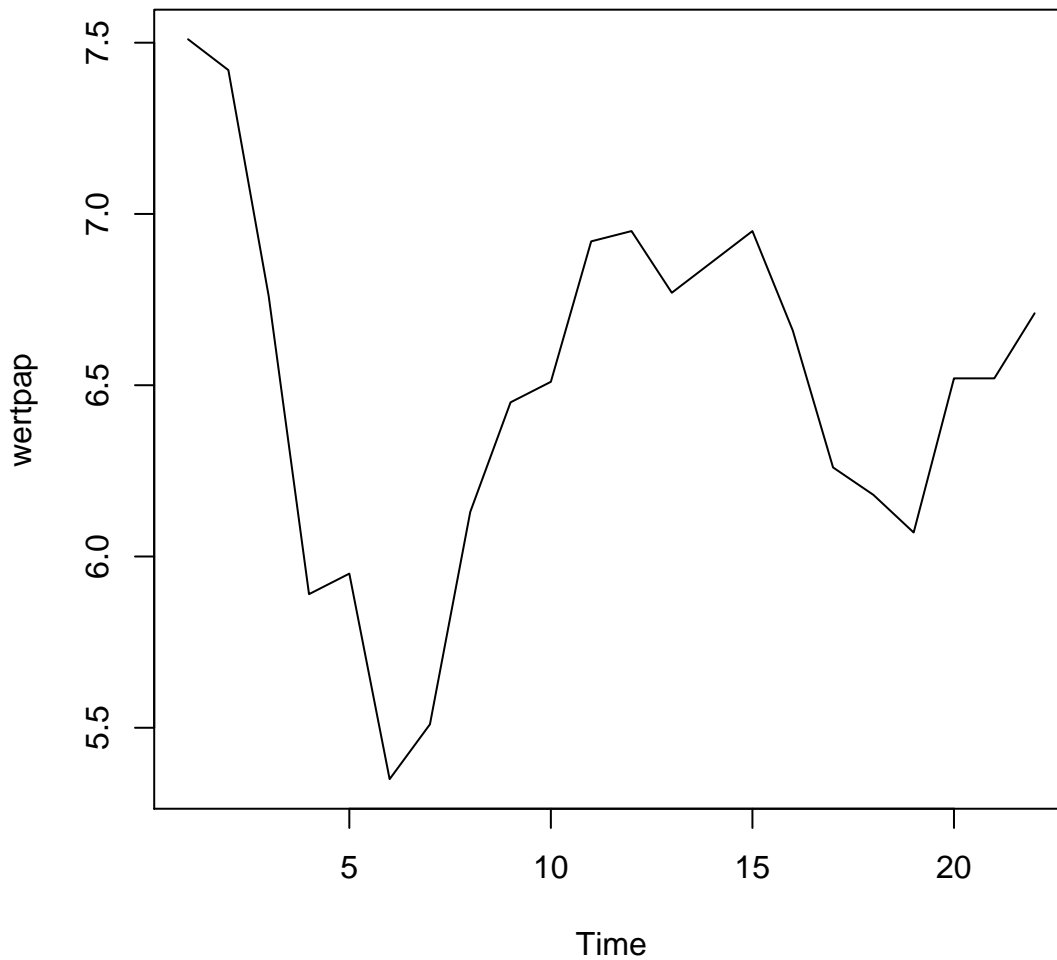
7.51 7.42 6.76 5.89 5.95 5.35 5.51 6.13 6.45 6.51 6.92
 6.95 6.77 6.86 6.95 6.66 6.26 6.18 6.07 6.52 6.52 6.71

- (a) Erstellen Sie einen Plot der Zeitreihe in R.
- (b) Bestimmen Sie den gleitenden 3er und 11er-Durchschnitt mit der Funktion `filter` und stellen Sie die Resultate graphisch dar.

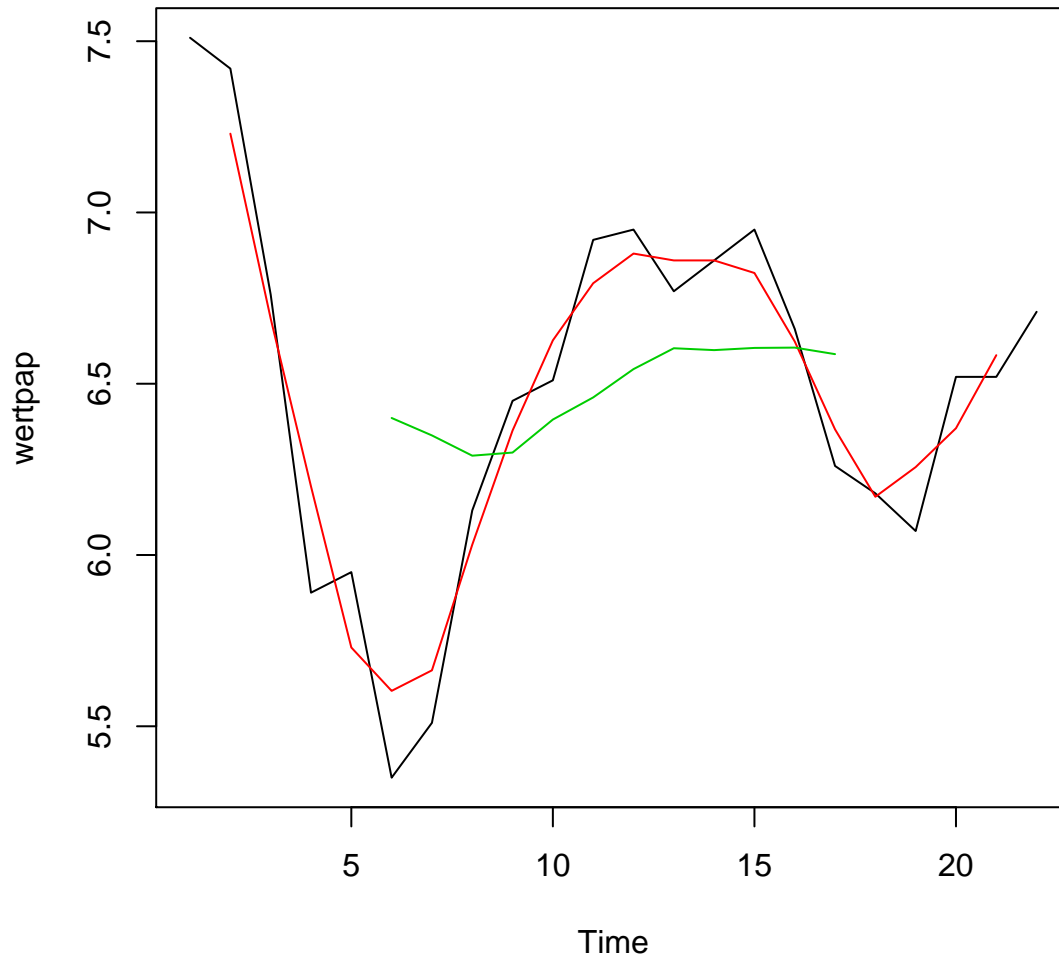
Lösung:

(übernommen von WiSe 2004)

```
R>wertpap <- c(7.51, 7.42, 6.76, 5.89, 5.95, 5.35, 5.51, 6.13,  
+ 6.45, 6.51, 6.92, 6.95, 6.77, 6.86, 6.95, 6.66, 6.26, 6.18,  
+ 6.07, 6.52, 6.52, 6.71)  
R>plot.ts(wertpap)
```



```
R>ma3 <- filter(wertpap, rep(1/3, 3))  
R>ma11 <- filter(wertpap, rep(1/11, 11))  
R>plot.ts(cbind(wertpap, ma3, ma11), plot.type = "single", col = c(1,  
+ 2, 3), ylab = "wertpap")
```

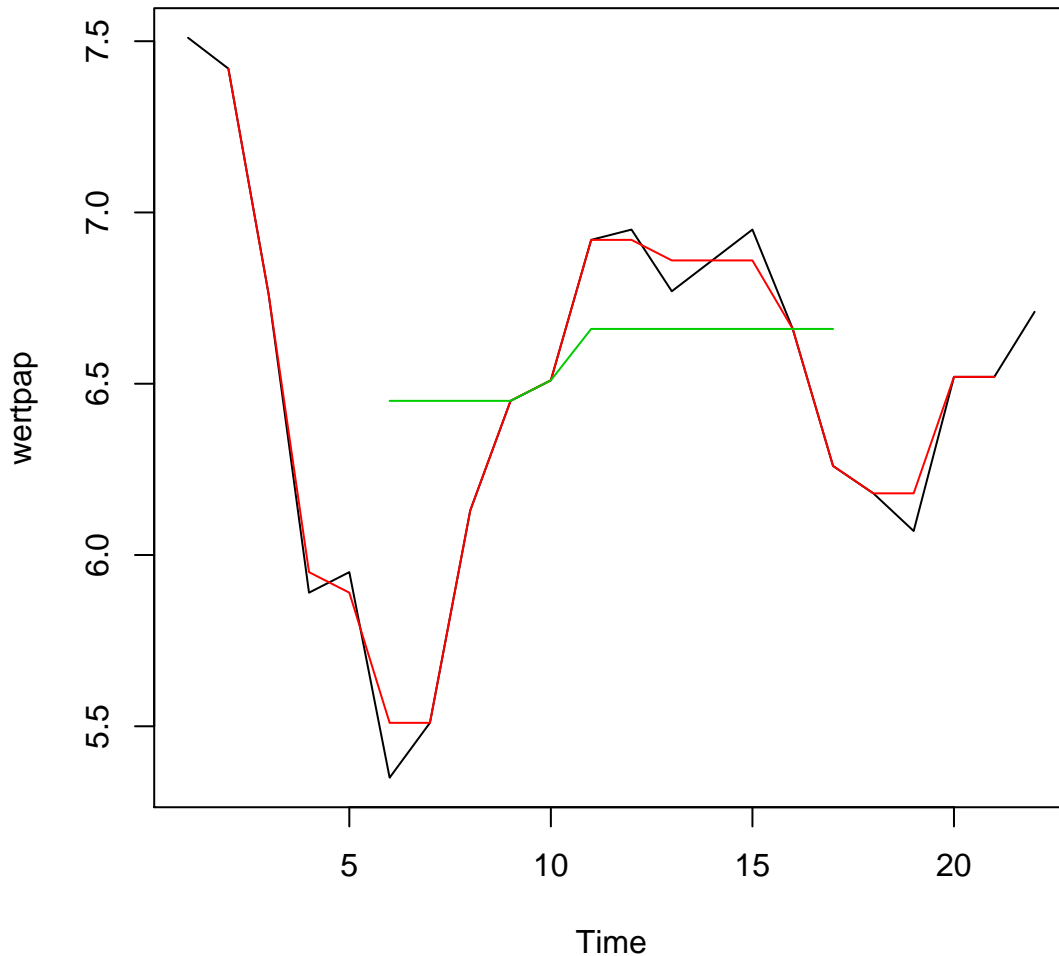


- (c) Anstelle gleitender Durchschnitte können zur Glättung einer Zeireihe auch gleitende Mediane verwendet werden, die analog zu den gleitenden Durchschnitten definiert sind. Berechnen Sie die entsprechenden 3er- und 11er Mediane und zeichnen Sie die Resultate.

Lösung:

Michael Höhle, WiSe 2004:

```
R>filter.med <- function(x, q = 1) {
+   res <- rep(NA, q)
+   for (i in (q + 1):(length(x) - q)) {
+     res <- append(res, median(x[(i - q):(i + q)]))
+   }
+   res <- append(res, rep(NA, q))
+   return(res)
+ }
R>me3 <- filter.med(wertpap, 1)
R>me11 <- filter.med(wertpap, 5)
R>plot.ts(cbind(wertpap, me3, me11), plot.type = "single", col = c(1,
+   2, 3), ylab = "wertpap")
```



Aufgabe 3:

Zur Glättung durch ein lokales Polynom soll ein Polynom $m(t) = \beta_1 + \beta_2 t + \beta_3 t^2$ zweiten Grades an die fünf Werte $y_{t-2}, y_{t-1}, y_t, y_{t+1}, y_{t+2}$ angepasst werden. Die Glättung für y_t bei der lokalen polynomialen Regression ist dann $\hat{y}_t = m(t)$.

(a) Zeigen Sie, dass für $t > 2$

$$\hat{y}_t = \frac{1}{35}(-3y_{t-2} + 12y_{t-1} + 17y_t + 12y_{t+1} - 3y_{t+2})$$

Hinweis: Zur Vereinfachung können die Zeitpunkte t neu durchnummeriert werden von -2 bis +2. Dann ist bei der Anpassung die Funktion

$$Q = \sum_{t=-2}^2 (y_t - \beta_1 - \beta_2 t - \beta_3 t^2)^2$$

zu minimieren. Die Vorhersage $\hat{y}_0 = \hat{\beta}_1$ ist der gewünschte Wert.

Lösung:

Beachte: $\sum_{t=-2}^2 t = 0$; $\sum_{t=-2}^2 t^2 = 10$; $\sum_{t=-2}^2 t^3 = 0$; $\sum_{t=-2}^2 t^4 = 34$.

$$\begin{aligned}
Q &= \sum_{t=-2}^2 (y_t - \beta_1 - \beta_2 t - \beta_3 t^2)^2 \rightarrow \min_{\beta} \\
&\Rightarrow \\
\frac{\partial Q}{\partial \beta_1} &= - \sum_{t=-2}^2 (y_t - \beta_1 - \beta_2 t - \beta_3 t^2) \stackrel{!}{=} 0 \\
&= - \sum_{t=-2}^2 y_t + 5\beta_1 + 10\beta_3 \stackrel{!}{=} 0 \\
\frac{\partial Q}{\partial \beta_2} &= - \sum_{t=-2}^2 ((y_t - \beta_1 - \beta_2 t - \beta_3 t^2)t) \stackrel{!}{=} 0 \\
&= - \sum_{t=-2}^2 t y_t + 10\beta_2 \stackrel{!}{=} 0 \\
\frac{\partial Q}{\partial \beta_3} &= - \sum_{t=-2}^2 ((y_t - \beta_1 - \beta_2 t - \beta_3 t^2)t^2) \stackrel{!}{=} 0 \\
&= - \sum_{t=-2}^2 t^2 y_t + 10\beta_1 + 34\beta_3 \stackrel{!}{=} 0
\end{aligned}$$

Uns interessiert nur der Ausdruck für $\hat{\beta}_1 = \hat{y}_0$:

$$\begin{aligned}
&\left[\begin{array}{ccc|c} 5 & 0 & 10 & \sum_{t=-2}^2 y_t \\ 0 & 10 & 0 & \sum_{t=-2}^2 t y_t \\ 10 & 0 & 34 & \sum_{t=-2}^2 t^2 y_t \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 2 & \frac{1}{5} \sum_{t=-2}^2 y_t \\ 0 & 10 & 0 & \sum_{t=-2}^2 t y_t \\ 0 & 0 & 14 & \sum_{t=-2}^2 t^2 y_t - 2 \sum_{t=-2}^2 y_t \end{array} \right] \\
\rightarrow &\left[\begin{array}{ccc|c} 1 & 0 & 0 & \frac{1}{5} \sum_{t=-2}^2 y_t - \frac{1}{7} (\sum_{t=-2}^2 t^2 y_t - 2 \sum_{t=-2}^2 y_t) \\ 0 & 10 & 0 & \sum_{t=-2}^2 t y_t \\ 0 & 0 & 14 & \sum_{t=-2}^2 t^2 y_t - 2 \sum_{t=-2}^2 y_t \end{array} \right]
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \hat{\beta}_1 &= \frac{1}{5} (y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2}) \\
&\quad - \frac{1}{7} (4y_{t-2} + y_{t-1} + y_{t+1} + 4y_{t+2}) \\
&\quad + \frac{2}{7} (y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2}) \\
&= \frac{1}{35} (-3y_{t-2} + 12y_{t-1} + 17y_t + 12y_{t+1} - 3y_{t+2})
\end{aligned}$$

q.e.d.

- (b) Schreiben Sie eine Funktion `fit.poly2(y)` in R die für einen Datenvektor \mathbf{y} der Länge 5 mittels der Funktion `lm` ein Polynom $m(t)$ der Ordnung $p = 2$ und Zeitpunkten $t = (-2, -1, 0, 1, 2)$ an \mathbf{y} anpasst. Die Funktion soll $\hat{y}_0 = m(0)$ zurückgeben. Benutzen Sie diese Funktion um eine lokale polynomiale Regression der gegebenen Daten durchzuführen und stellen Sie das Resultat graphisch dar. Vergleichen Sie das Resultat mit dem Resultat des äquivalenten entsprechenden GD-Filters.

Lösung:

Code von Michael Höhle.

```
R>fit.poly2 <- function(y) {
+   n <- length(y)
+   if (n != 5)
+     stop("wrong length for y!")
+   t <- c(-2, -1, 0, 1, 2)
+   m <- lm(y ~ 1 + t + I(t^2))
+   yhat <- coef(m)[1]
+   return(yhat)
+ }
R>filter.poly <- function(y, q = 2) {
+   res <- rep(NA, length(y))
+   for (i in (q + 1):(length(y) - q)) {
+     yhat <- fit.poly2(y[(i - q):(i + q)])
+     res[i] <- yhat
+   }
+   return(res)
+ }
R>fit.poly <- function(y, p, pred = "t") {
+   q <- (length(y) - 1)/2
+   t <- seq(-q, q)
+   model <- "y~1"
+   for (i in 1:p) {
+     model <- paste(model, " + I(t^", i, ")", sep = "")
+   }
+   m <- lm(as.formula(model))
+   switch(pred, t = {
+     yhat = predict(m)[q + 1]
+   }, left = {
+     yhat = predict(m)[1:(q)]
+   }, right = {
+     yhat = predict(m)[(q + 2):(2 * q + 1)]
+   })
+   return(yhat)
+ }
R>filter.poly.edge <- function(y, q = 2, p = 2) {
+   res <- fit.poly(y[1:(2 * q + 1)], p = p, pred = "left")
+   for (i in (q + 1):(length(y) - q)) {
+     yhat <- fit.poly(y[(i - q):(i + q)], p = 2)
+     res <- append(res, yhat)
+   }
+   n <- length(y)
+   res <- append(res, fit.poly(y[(n - 2 * q):n], p = p, pred = "right"))
+   return(res)
+ }
R>fP <- filter.poly(wertpap, q = 2)
R>plot.ts(cbind(wertpap, fP), col = 1:2, plot.type = "single",
+   ylab = "wertpap")
R>D <- 1/35 * c(-3, 12, 17, 12, -3)
R>fD <- filter(wertpap, D)
R>fP - fD
```

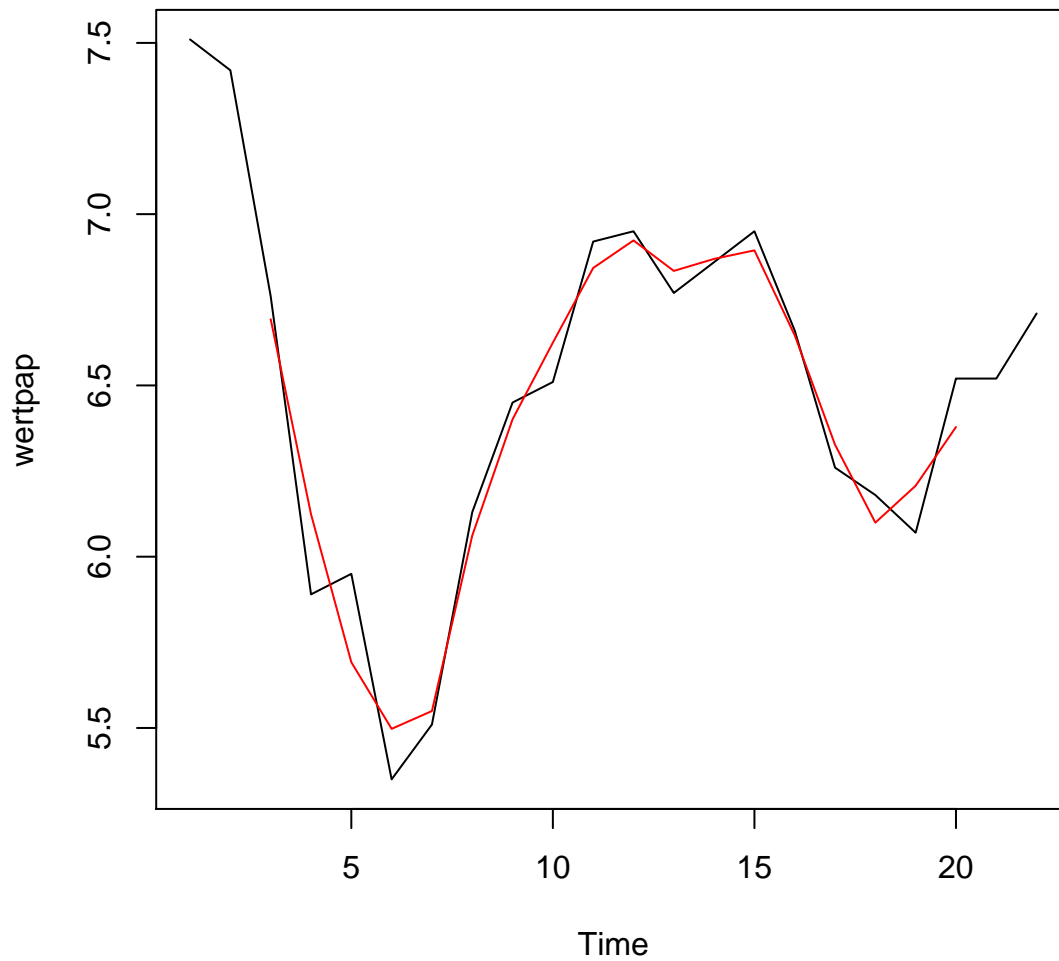
Time Series:

Start = 1

End = 22

Frequency = 1

```
[1] NA NA 0.000000e+00 0.000000e+00 -8.881784e-16
[6] -2.664535e-15 -8.881784e-16 -8.881784e-16 0.000000e+00 0.000000e+00
[11] -8.881784e-16 -1.776357e-15 0.000000e+00 0.000000e+00 -8.881784e-16
[16] 0.000000e+00 0.000000e+00 0.000000e+00 -1.776357e-15 0.000000e+00
[21] NA NA
```



```
R>fPe <- filter.poly.edge(wertpap, q = 2, p = 2)
R>plot.ts(cbind(wertpap, fPe), col = 1:2, plot.type = "single",
+        ylab = "wertpap")
```

