

dialyzer1.R:

```
library(nlme)
Dialyzer[1:3,]
plot(Dialyzer, outer= ~QB,layout=c(2,1))
m1Dial.lis <- nlsList(rate~SSasypOff(pressure,th1,th2,th3) | QB, data=Dialyzer)
m1Dial.lis
plot(intervals(m1Dial.lis))
t(model.matrix(~QB,data=Dialyzer))

# the model below uses dummy variable coding of the second type mentioned
# in the notes. That is, th1.(Intercept) is the th1-value for QB=200
# (the 1st level of QB) and th1.(Intercept)+th1.QB300 is the th1-value
# for QB=300 (the 2nd level of QB)
m2Dial.gnls <- gnls( rate ~ SSasypOff(pressure, th1,th2,th3),
  data = Dialyzer, params = list(th1 ~ QB, th2 ~ QB, th3 ~QB),
  start=c(44.99,62.22-44.99, .76,.25-.76, .22,0))
summary(m2Dial.gnls)

plot(m2Dial.gnls,grid=F,main="Residuals vs Fitteds")
plot(m2Dial.gnls, resid(.) ~ pressure, abline=0,grid=F,
  main="Residuals vs x=Transmembrane Pressure")

# the model below uses dummy variable coding of the first type mentioned
# in the notes. That is, th1.QB200 is the th1-value for QB=200
# (the 1st level of QB) and th1.QB300 is the th1-value
# for QB=300 (the 2nd level of QB)
m2altDial.gnls <- gnls( rate ~ SSasypOff(pressure, th1,th2,th3),
  data = Dialyzer, params = list(th1 ~ QB-1, th2 ~ QB-1, th3 ~QB-1),
  start=c(44.99,62.22, .76,.25, .22,.22))
summary(m2altDial.gnls)
# Note that m2Dial.gnls and m2altDial.gnls are equivalent models differing
# only in their parameterization. Their equivalence can be seen from the fact
# that they have the same loglikelihood and AIC values

m3Dial.gnls <- update(m2Dial.gnls,weights=varPower())
anova(m2Dial.gnls,m3Dial.gnls)
m4Dial.gnls <- update(m2Dial.gnls,weights=varPower(form = ~ pressure))
anova(m2Dial.gnls,m4Dial.gnls)
anova(m3Dial.gnls,m4Dial.gnls)
plot(ACF(m4Dial.gnls, form=~1|Subject),alpha=.05)
m5Dial.gnls <- update(m4Dial.gnls,corr=corAR1(.7,form=~1|Subject))
anova(m4Dial.gnls,m5Dial.gnls)
plot(ACF(m5Dial.gnls, resType="n",form=~1|Subject),alpha=.05)
summary(m5Dial.gnls)
# The t-test of th3.QB300 in the above summary of model m5Dial.gnls
# is the Wald test of H_0:gamma_3=0 (that is, of the hypothesis
# that the offset parameter does not differ across QB levels
# Alternatively, we can get an equivalent F test via the anova function
# by specifying the hypothesis in the form A*theta=0 where A is given
# by Lmat below. Note that this Wald test is the F test given as (**) at
# the top of p. 174 of the class notes.
Lmat=matrix(c(0,0,0,0,0,1),nrow=1,byrow=T)
Lmat
anova(m5Dial.gnls,type="marginal",L=Lmat,adjustSigma=F)

# Instead of the Wald test of H_0:gamma_3=0, it is better to use a LR test.
# There are two versions of the LR test: an F version (given in two different,
# but equivalent forms on p.171 and toward the top of p.173) and a chi-square
# version (given as 2*log(lambda) in the middle of p.173). The F version is
# preferable, but they are equivalent for large samples.

# To do the LR tests, we need to fit the model under the null hypothesis
```

```

# that theta_3=0. When we do that we need to keep the variance covariance
# parameter fixed at the values from the model where theta_3 is not assumed
# equal to 0 (model m5Dial.gnls)

# The following model call should work but doesn't due to some unknown bug
# in the update.gnls function.
#m6Dial.gnls <- update(m5Dial.gnls, params=list(th1 ~ QB, th2 ~ QB, th3 ~ 1),
# weights=varPower(form = ~ pressure,fixed=.5849288),
# corr=corAR1(value=.7432391,fixed=T,form=~1|Subject),
# start=c(55.11,7.77,.37,-.16,.22))

# So, we'll try to fit this model without using the update function
m6Dial.gnls <- gnls( rate ~ SSasymptOff(pressure, th1,th2,th3),
  data = Dialyzer, params=list(th1 ~ QB, th2 ~ QB, th3 ~ 1),
  weights=varPower(form = ~ pressure,fixed=.5849288),
  corr=corAR1(value=.7432391,fixed=T,form=~1|Subject),
  start=c(47,15,.54,-.33,.22))

# The following statement gives the chi-square version of the LR test,
# but note that it gets the difference in df of the two models wrong, and
# hence also gives the wrong p-value
anova(m5Dial.gnls,m6Dial.gnls) # gives right test stat, wrong df and p-val

# This can be fixed by computing the p-value from the chi-square
# distribution and correctly specifying the df as 1, not 3:
1-pchisq(2*(logLik(m5Dial.gnls)-logLik(m6Dial.gnls)),1) #correct p-value based on 1 df

#A way to get around the above problem that the anova function gets the wrong
# df and p-value for this test is as follows: refit model m5 fixing the
# covariance parameters
# at their estimated values. Call this model m5a (will be exactly the same as
# model m5, but the software will no longer regard the covariance parameters
# as free parameters and won't count them in the df). Then test model m6 vs
# m5a with the anova function.

m5aDial.gnls <- update(m4Dial.gnls,weights=varPower(form=~pressure, fixed=.5849288),
  corr=corAR1(value=.7432391,fixed=T,form=~1|Subject))
logLik(m5Dial.gnls)
logLik(m5aDial.gnls) # should be the same because models are identical

# this will give the correct chi-square version of the LRT
anova(m6Dial.gnls,m5aDial.gnls)

# It is a bit harder to get the F version of the LR test, but here it is
# (cf. the formula on the top of p.173):
lam <- as.numeric(exp(logLik(m5Dial.gnls)-logLik(m6Dial.gnls)))
F <- (lam^(2/nrow(Dialyzer))-1)*( nrow(Dialyzer)-length(coef(m5Dial.gnls)))/1
1-pf( F,1,nrow(Dialyzer)-length(coef(m5Dial.gnls)) )

```

Output from dialyzer 1.R:

```

> library(nlme)
> Dialyzer[1:3,]
Grouped Data: rate ~ pressure | Subject
  Subject  QB pressure  rate index
1      1  200   0.240  0.645     1
2      1  200   0.505 20.115     2
3      1  200   0.995 38.460     3

```

```

> plot(Dialyzer, outer= ~QB,layout=c(2,1))
> m1Dial.lis <- nlsList(rate~SSasypOff(pressure,th1,th2,th3) | QB, data=Dialyzer)
> m1Dial.lis
Call:
  Model: rate ~ SSasypOff(pressure, th1, th2, th3) | QB
  Data: Dialyzer

```

```

Coefficients:
      th1      th2      th3
200 44.989 0.76486 0.22424
300 62.217 0.25282 0.22484

```

```

Degrees of freedom: 140 total; 134 residual
Residual standard error: 3.8043

```

```

> plot(intervals(m1Dial.lis))
> t(model.matrix(~QB,data=Dialyzer))
      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      128 129 130 131 132 133 134 135 136 137 138 139 140
(Intercept) 1 1 1 1 1 1 1 1 1 1 1 1 1 1
QB300        1 1 1 1 1 1 1 1 1 1 1 1 1 1
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$QB
[1] "contr.treatment"

```

```

>
> # the model below uses dummy variable coding of the second type mentioned
> # in the notes. That is, th1.(Intercept) is the th1-value for QB=200
> # (the 1st level of QB) and th1.(Intercept)+th1.QB300 is the th1-value
> # for QB=300 (the 2nd level of QB)
> m2Dial.gnls <- gnls( rate ~ SSasypOff(pressure, th1,th2,th3),
+ data = Dialyzer, params = list(th1 ~ QB, th2 ~ QB, th3 ~QB),
+ start=c(44.99,62.22-44.99, .76,.25-.76, .22,0))
> summary(m2Dial.gnls)

```

```

Generalized nonlinear least squares fit
  Model: rate ~ SSasypOff(pressure, th1, th2, th3)
  Data: Dialyzer
      AIC      BIC  logLik
779.29 799.88 -382.64

```

```

Coefficients:
      Value Std.Error t-value p-value
th1.(Intercept) 44.989  0.75284  59.758  0.000
th1.QB300       17.229  1.43471  12.009  0.000
th2.(Intercept)  0.765  0.08465   9.036  0.000

```

```

th2.QB300      -0.512   0.11090  -4.617   0.000
th3.(Intercept) 0.224   0.01409  15.920   0.000
th3.QB300      0.001   0.02156   0.028   0.978

```

Correlation:

```

          t1.(I) t1.QB3 t2.(I) t2.QB3 t3.(I)
th1.QB300      -0.525
th2.(Intercept) -0.650  0.341
th2.QB300       0.496 -0.724 -0.763
th3.(Intercept) -0.218  0.114  0.562 -0.429
th3.QB300       0.142 -0.313 -0.367  0.580 -0.653

```

Standardized residuals:

```

      Min      Q1      Med      Q3      Max
-3.17673 -0.48606  0.09116  0.59929  2.70751

```

Residual standard error: 3.8043

Degrees of freedom: 140 total; 134 residual

```

>
> plot(m2Dial.gnls,grid=F,main="Residuals vs Fitteds")
> plot(m2Dial.gnls, resid(.) ~ pressure, abline=0,grid=F,
+      main="Residuals vs x=Transmembrane Pressure")
>
> # the model below uses dummy variable coding of the first type mentioned
> # in the notes. That is, th1.QB200 is the th1-value for QB=200
> # (the 1st level of QB) and th1.QB300 is the th1-value
> # for QB=300 (the 2nd level of QB)
> m2altDial.gnls <- gnls( rate ~ SSasympOff(pressure, th1,th2,th3),
+ data = Dialyzer, params = list(th1 ~ QB-1, th2 ~ QB-1, th3 ~QB-1),
+ start=c(44.99,62.22, .76,.25, .22,.22))
> summary(m2altDial.gnls)
Generalized nonlinear least squares fit
Model: rate ~ SSasympOff(pressure, th1, th2, th3)
Data: Dialyzer
      AIC      BIC logLik
779.29 799.88 -382.64

```

Coefficients:

```

      Value Std.Error t-value p-value
th1.QB200 44.989   0.75284  59.758  0e+00
th1.QB300 62.217   1.22131  50.943  0e+00
th2.QB200  0.765   0.08465   9.036  0e+00
th2.QB300  0.253   0.07166   3.528  6e-04
th3.QB200  0.224   0.01409  15.920  0e+00
th3.QB300  0.225   0.01633  13.770  0e+00

```

Correlation:

```

          t1.QB2 t1.QB3 t2.QB2 t2.QB3 t3.QB2
th1.QB300  0.000
th2.QB200 -0.650  0.000
th2.QB300  0.000 -0.842  0.000
th3.QB200 -0.218  0.000  0.562  0.000
th3.QB300  0.000 -0.369  0.000  0.613  0.000

```

Standardized residuals:

```

      Min      Q1      Med      Q3      Max
-3.17673 -0.48606  0.09116  0.59929  2.70751

```

Residual standard error: 3.8043

Degrees of freedom: 140 total; 134 residual

```

> # Note that m2Dial.gnls and m2altDial.gnls are equivalent models differing
> # only in their parameterization. Their equivalence can be seen from the fact
> # that they have the same loglikelihood and AIC values

```

```

>
> m3Dial.gnls <- update(m2Dial.gnls,weights=varPower())
> anova(m2Dial.gnls,m3Dial.gnls)
      Model df      AIC      BIC  logLik  Test L.Ratio p-value
m2Dial.gnls  1  7 779.29 799.88 -382.64
m3Dial.gnls  2  8 751.93 775.47 -367.97 1 vs 2  29.356 <.0001
> m4Dial.gnls <- update(m2Dial.gnls,weights=varPower(form = ~ pressure))
> anova(m2Dial.gnls,m4Dial.gnls)
      Model df      AIC      BIC  logLik  Test L.Ratio p-value
m2Dial.gnls  1  7 779.29 799.88 -382.64
m4Dial.gnls  2  8 750.37 773.90 -367.18 1 vs 2  30.919 <.0001
> anova(m3Dial.gnls,m4Dial.gnls)
      Model df      AIC      BIC  logLik
m3Dial.gnls  1  8 751.93 775.47 -367.97
m4Dial.gnls  2  8 750.37 773.90 -367.18
> plot(ACF(m4Dial.gnls, form=~1|Subject),alpha=.05)
> m5Dial.gnls <- update(m4Dial.gnls,corr=corAR1(.7,form=~1|Subject))
> anova(m4Dial.gnls,m5Dial.gnls)
      Model df      AIC      BIC  logLik  Test L.Ratio p-value
m4Dial.gnls  1  8 750.37 773.90 -367.18
m5Dial.gnls  2  9 661.46 687.93 -321.73 1 vs 2  90.912 <.0001
> plot(ACF(m5Dial.gnls, resType="n",form=~1|Subject),alpha=.05)
> summary(m5Dial.gnls)
Generalized nonlinear least squares fit
Model: rate ~ SSasympOff(pressure, th1, th2, th3)
Data: Dialyzer
      AIC      BIC  logLik
661.46 687.93 -321.73

Correlation Structure: AR(1)
Formula: ~1 | Subject
Parameter estimate(s):
  Phi
0.74325
Variance function:
Structure: Power of variance covariate
Formula: ~pressure
Parameter estimates:
  power
0.58493

Coefficients:
      Value Std.Error t-value p-value
th1.(Intercept) 47.335  1.57702  30.016  0.0000
th1.QB300       15.550  2.51263   6.189  0.0000
th2.(Intercept)  0.536  0.05387   9.948  0.0000
th2.QB300      -0.327  0.07517  -4.346  0.0000
th3.(Intercept)  0.210  0.00558  37.609  0.0000
th3.QB300       0.010  0.00833   1.251  0.2131

Correlation:
      t1.(I) t1.QB3 t2.(I) t2.QB3 t3.(I)
th1.QB300  -0.628
th2.(Intercept) -0.691  0.434
th2.QB300     0.495 -0.714 -0.717
th3.(Intercept) -0.308  0.193  0.138 -0.099
th3.QB300     0.206 -0.274 -0.092  0.142 -0.670

Standardized residuals:
      Min      Q1      Med      Q3      Max
-2.56311 -0.72770  0.08389  0.64486  2.59294

Residual standard error: 3.1667

```

```

Degrees of freedom: 140 total; 134 residual
> # The t-test of th3.QB300 in the above summary of model m5Dial.gnls
> # is the Wald test of H_0:gamma_3=0 (that is, of the hypothesis
> # that the offset parameter does not differ across QB levels
> # Alternatively, we can get an equivalent F test via the anova function
> # by specifying the hypothesis in the form A*theta=0 where A is given
> # by Lmat below. Note that this Wald test is the F test given as (**) at
> # the top of p. 174 of the class notes.
> Lmat=matrix(c(0,0,0,0,0,1),nrow=1,byrow=T)
> Lmat
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
> anova(m5Dial.gnls,type="marginal",L=Lmat,adjustSigma=F)
Denom. DF: 134
F-test for linear combination(s)
[1] 1
      numDF F-value p-value
1         1  1.5654  0.2131
>
> # Instead of the Wald test of H_0:gamma_3=0, it is better to use a LR test.
> # There are two versions of the LR test: an F version (given in two different,
> # but equivalent forms on p.171 and toward the top of p.173) and a chi-square
> # version (given as 2*log(lambda) in the middle of p.173). The F version is
> # preferable, but they are equivalent for large samples.
>
> # To do the LR tests, we need to fit the model under the null hypothesis
> # that theta_3=0. When we do that we need to keep the variance covariance
> # parameter fixed at the values from the model where theta_3 is not assumed
> # equal to 0 (model m5Dial.gnls)
>
> # The following model call should work but doesn't due to some unknown bug
> # in the update.gnls function.
> #m6Dial.gnls <- update(m5Dial.gnls, params=list(th1 ~ QB, th2 ~ QB, th3 ~ 1),
> # weights=varPower(form = ~ pressure,fixed=.5849288),
> # corr=corAR1(value=.7432391,fixed=T,form=~1|Subject),
> # start=c(55.11,7.77,.37,-.16,.22))
>
> # So, we'll try to fit this model without using the update function
> m6Dial.gnls <- gnls( rate ~ SSasymOff(pressure, th1,th2,th3),
+ data = Dialyzer, params=list(th1 ~ QB, th2 ~ QB, th3 ~ 1),
+ weights=varPower(form = ~ pressure,fixed=.5849288),
+ corr=corAR1(value=.7432391,fixed=T,form=~1|Subject),
+ start=c(47,15,.54,-.33,.22))
>
> # The following statement gives the chi-square version of the LR test,
> # but note that it gets the difference in df of the two models wrong, and
> # hence also gives the wrong p-value
> anova(m5Dial.gnls,m6Dial.gnls) # gives right test stat, wrong df and p-val
      Model df    AIC    BIC  logLik  Test L.Ratio p-value
m5Dial.gnls    1  9 661.46 687.93 -321.73
m6Dial.gnls    2  6 657.08 674.73 -322.54 1 vs 2  1.6227  0.6542
>
> # This can be fixed by computing the p-value from the chi-square
> # distribution and correctly specifying the df as 1, not 3:
> 1-pchisq(2*(logLik(m5Dial.gnls)-logLik(m6Dial.gnls)),1) #correct p-value based on 1
df
[1] 0.20271
attr(,"df")
[1] 9
attr(,"class")
[1] "logLik"
attr(,"null")
[1] 140

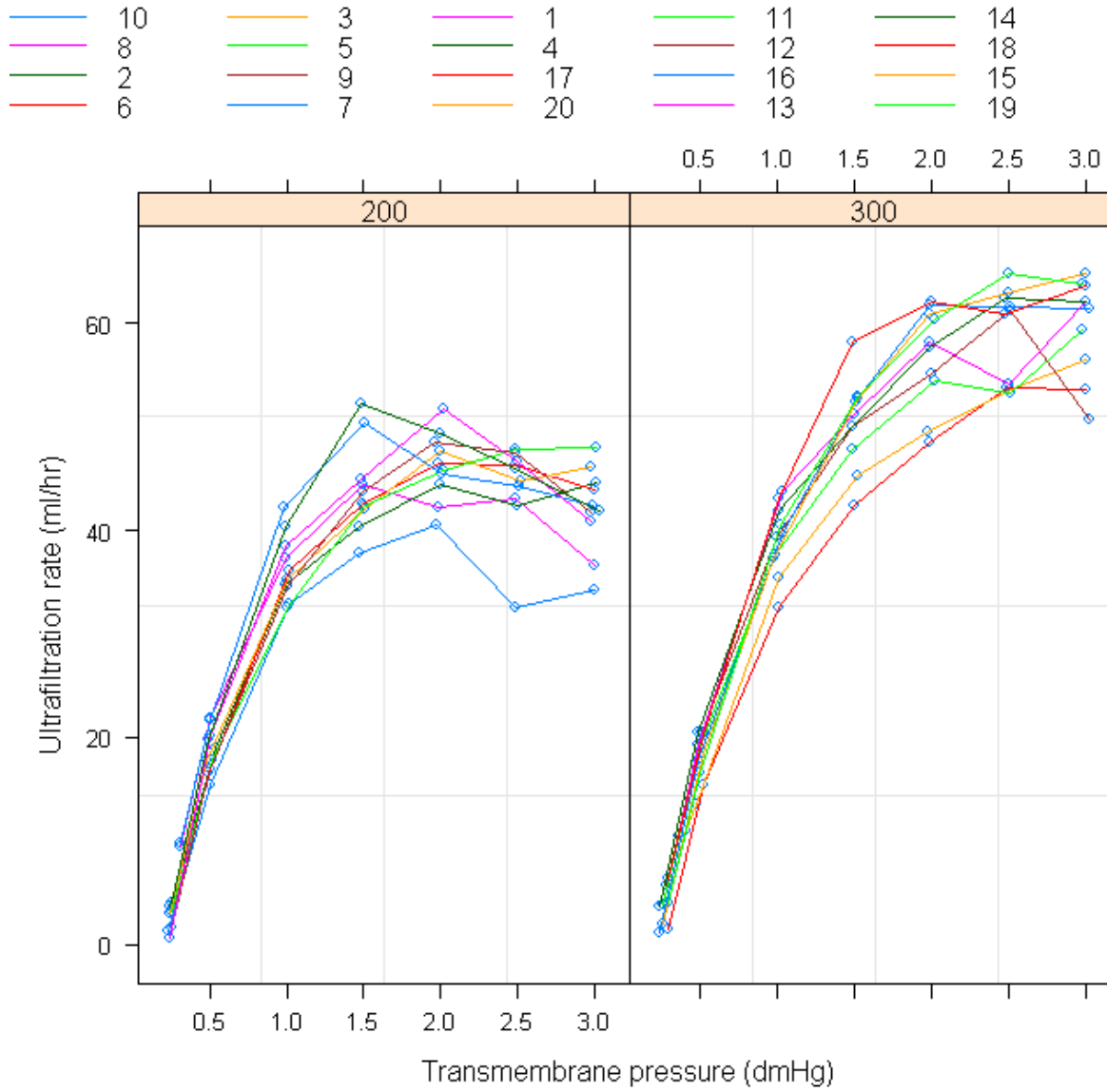
```

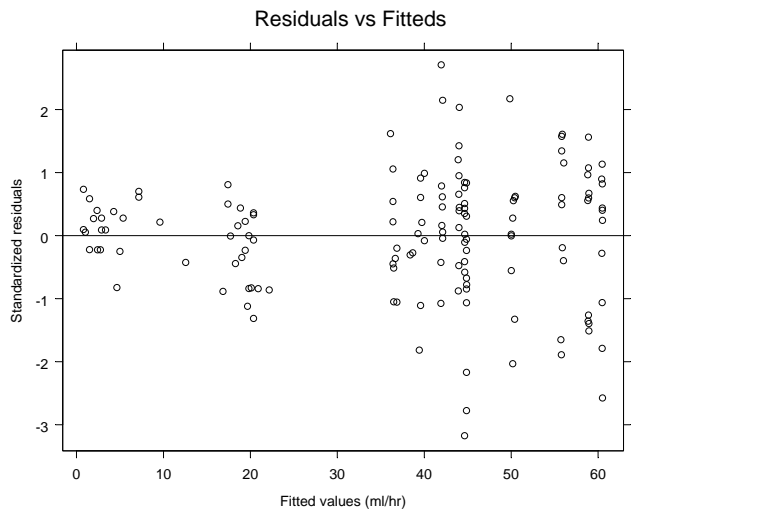
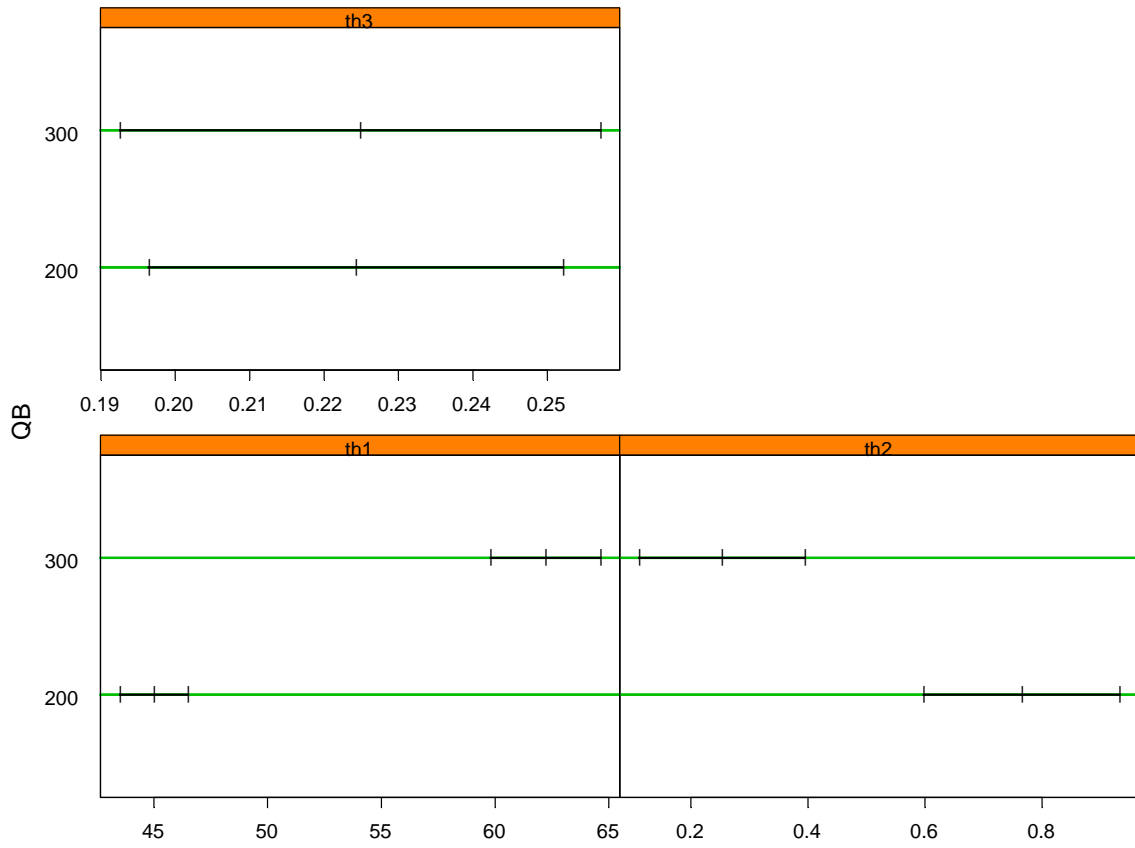
```

attr(,"nobs")
[1] 140
>
> #A way to get around the above problem that the anova function gets the wrong
> # df and p-value for this test is as follows: refit model m5 fixing the
> # covariance parameters
> # at their estimated values. Call this model m5a (will be exactly the same as
> # model m5, but the software will no longer regard the covariance parameters
> # as free parameters and won't count them in the df). Then test model m6 vs
> # m5a with the anova function.
>
> m5aDial.gnls <- update(m4Dial.gnls,weights=varPower(form=~pressure, fixed=.5849288),
+   corr=corAR1(value=.7432391,fixed=T,form=~1|Subject))
> logLik(m5Dial.gnls)
'log Lik.' -321.73 (df=9)
> logLik(m5aDial.gnls) # should be the same because models are identical
'log Lik.' -321.73 (df=7)
>
> # this will give the correct chi-square version of the LRT
> anova(m6Dial.gnls,m5aDial.gnls)
      Model df    AIC    BIC  logLik  Test L.Ratio p-value
m6Dial.gnls     1  6 657.08 674.73 -322.54
m5aDial.gnls     2  7 657.46 678.05 -321.73 1 vs 2  1.6227  0.2027
>
>
> # It is a bit harder to get the F version of the LR test, but here it is
> # (cf. the formula on the top of p.173):
> lam <- as.numeric(exp(logLik(m5Dial.gnls)-logLik(m6Dial.gnls)))
> F <- (lam^(2/nrow(Dialyzer))-1)*( nrow(Dialyzer)-length(coef(m5Dial.gnls)))/1
> 1-pf( F,1,nrow(Dialyzer)-length(coef(m5Dial.gnls)) )
[1] 0.21352
>
>
>

```

Plots from dialyzer 1.R:





Residuals vs x=Transmembrane Pressure

