Report 1

Klaudia Weigel

Summary of the model:

```r
summary(lead.cat)
```

```
## Generalized least squares fit by REML
##   Model: y ~ time.cat
##   Data: lead.uni
##        AIC      BIC    logLik
##   1308.337 1354.231 -640.1687
##
## Correlation Structure: General
##  Formula: ~1 | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.401
## 3 0.384 0.731
## 4 0.495 0.507 0.455
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time.cat
##  Parameter estimates:
##        1        2        3        4
## 1.000000 1.528082 1.563877 1.841540
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)  26.540 0.7100723  37.37647       0
## time.cat2   -13.018 1.0309741 -12.62689       0
## time.cat3   -11.026 1.0638646 -10.36410       0
## time.cat4    -5.778 1.1378261  -5.07810       0
##
##  Correlation:
##           (Intr) tm.ct2 tm.ct3
## time.cat2 -0.266
## time.cat3 -0.267  0.704
## time.cat4 -0.055  0.387  0.332
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.8020147 -0.8258011 -0.1227685  0.5184782  4.6654249
##
## Residual standard error: 5.02097
## Degrees of freedom: 200 total; 196 residual
```

Exercise 1

First we retrieve the correlation and variance coefficients:

```
cor.coef <- coef(lead.cat$modelStruct$corStruct, uncons = FALSE, allCoef = TRUE)
var.coef <- coef(lead.cat$modelStruct$varStruct, uncons = FALSE, allCoef = TRUE)
```

```
cor.coef
```

```
## [1] 0.4014637 0.3839701 0.4951173 0.7308190 0.5069640 0.4548186
```

```
var.coef
```

```
##        1        2        3        4
## 1.000000 1.528082 1.563877 1.841540
```

Then, we create matrices with correlations and variances:

```
m.corr1 <- matrix(ncol = 4, nrow = 4)
m.corr1[lower.tri(m.corr1, diag=TRUE)] <- c(1, cor.coef[1:3], 1,
                                             cor.coef[4:5], 1, cor.coef[6], 1)
m.corr <- t(m.corr1)
m.corr[lower.tri(m.corr, diag=TRUE)] <- c(1, cor.coef[1:3], 1,
                                           cor.coef[4:5], 1, cor.coef[6], 1)
```

```
m.var <- matrix(nrow = 4, ncol = 4)
for (i in c(1,2,3,4)) {
  for (j in c(1,2,3,4)) {
    m.var[i,j] = var.coef[i]*var.coef[j]
  }
}
```

Lastly, we get the baseline variance:

```
sig = summary(lead.cat)$sigma^2
```

We calculate the error variance-covariance matrix:

```
m.corr*m.var*sig
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 25.21014 15.46565 15.13824 22.98606
## [2,] 15.46565 58.86654 44.02855 35.96501
## [3,] 15.13824 44.02855 61.65673 33.02153
## [4,] 22.98606 35.96501 33.02153 85.49439
```

We can check our result with the inbuilt funtion:

```
getVarCov(lead.cat)
```

```
## Marginal variance covariance matrix
##        [,1]   [,2]   [,3]   [,4]
## [1,] 25.210 15.466 15.138 22.986
## [2,] 15.466 58.867 44.029 35.965
## [3,] 15.138 44.029 61.657 33.022
## [4,] 22.986 35.965 33.022 85.494
##   Standard Deviations: 5.021 7.6725 7.8522 9.2463
```

Exercise 2

```
lead.cat.ml <- gls(y~factor(time.cat),
                   correlation=corSymm(form= ~1 | id),
                   weights=varIdent(form= ~1 | time.cat),
                   data=lead.uni, method = "ML")
```

```
getVarCov(lead.cat.ml)
```

```
## Marginal variance covariance matrix
##        [,1]   [,2]   [,3]   [,4]
## [1,] 24.706 15.156 14.835 22.525
## [2,] 15.156 57.690 43.148 35.246
## [3,] 14.835 43.148 60.424 32.360
## [4,] 22.525 35.246 32.360 83.782
##   Standard Deviations: 4.9705 7.5954 7.7733 9.1533
```

Exercise 3

```
cov2cor(getVarCov(lead.cat))
```

```
## Marginal variance covariance matrix
##         [,1]    [,2]    [,3]    [,4]
## [1,] 1.00000 0.40146 0.38397 0.49512
## [2,] 0.40146 1.00000 0.73082 0.50696
## [3,] 0.38397 0.73082 1.00000 0.45482
## [4,] 0.49512 0.50696 0.45482 1.00000
##   Standard Deviations: 1 1 1 1
```

Looking at the correlation matrix we see that data is positively correlated. The correlations tend to decrease with time. The variances tend to increase over time.

Exercise 4

1) Compound symmetry.

```
lead.cat.cs <- gls(y ~ time.cat,
                   correlation=corCompSymm(form= ~1 | id),
                   data=lead.uni)
```

```
getVarCov(lead.cat.cs)
```

```
## Marginal variance covariance matrix
##        [,1]   [,2]   [,3]   [,4]
## [1,] 57.807 27.768 27.768 27.768
## [2,] 27.768 57.807 27.768 27.768
## [3,] 27.768 27.768 57.807 27.768
## [4,] 27.768 27.768 27.768 57.807
##   Standard Deviations: 7.6031 7.6031 7.6031 7.6031
```

Let's test the compound symmetry model with the likelihood ratio test. The test statistic is of the form: $-2\ln(\frac{\mathcal{L}_s}{\mathcal{L}_c})$, where $\mathcal{L}_s$ is the likelihood of the simpler model and $\mathcal{L}_c$ the likelihood of the complex model.

```
logLik.cs <- summary(lead.cat.cs)$logLik
logLik.orig <- summary(lead.cat)$logLik
test.statistic.cs <- as.numeric(-2*logLik.cs + 2*logLik.orig)
p.val.cs <- pchisq(test.statistic.cs, df = 8, lower.tail = FALSE)
```

```
p.val.cs
```

```
## [1] 3.688198e-05
```

The p-value is smaller than 0.05, so for this significance level we reject the null hypothesis. Therefore the compound symmetry assumption is not reasonable for the data.

2) Autoregressive correlation with heterogenous variance.

```
lead.cat.ar <- gls(y ~ time.cat,
                   correlation=corAR1(form= ~1 | id),
                   weights=varIdent(form= ~1 | time.cat),
                   data=lead.uni)
```

```
getVarCov(lead.cat.ar)
```

```
## Marginal variance covariance matrix
##         [,1]   [,2]   [,3]    [,4]
## [1,] 26.9970 20.370 10.790  7.0357
## [2,] 20.3700 55.899 29.610 19.3080
## [3,] 10.7900 29.610 57.047 37.1980
## [4,]  7.0357 19.308 37.198 88.2180
##   Standard Deviations: 5.1959 7.4766 7.553 9.3925
```

```
logLik.ar <- summary(lead.cat.ar)$logLik
test.statistic.ar <- as.numeric(-2*logLik.ar + 2*logLik.orig)
p.val.ar <- pchisq(test.statistic.ar, df = 5, lower.tail = FALSE)
```

```
p.val.ar
```

```
## [1] 0.000972854
```

We reject this model as well.

Exercise 5

We can obtain the estimated means and the 95% confidence intervals using AICcmodavg library:

```
library(AICcmodavg)
new.dat <- expand.grid(time.cat = c("1","2","3","4"))
predictions <- data.frame(predictSE.gls(lead.cat, newdata = new.dat))
quantile_norm <- qnorm(p = 0.975)
predictions$estimated_lower <- predictions$fit - quantile_norm * predictions$se.fit
predictions$estimated_upper <- predictions$fit + quantile_norm * predictions$se.fit
predictions$time.cat <- factor(c("1","2","3","4"))
```

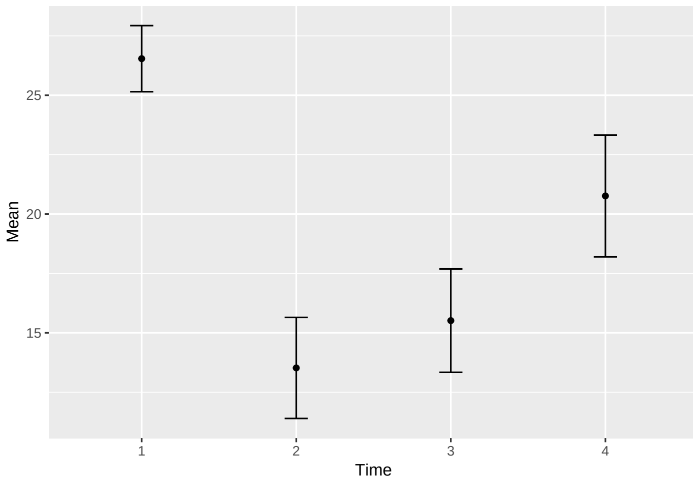
```
predictions
```

```
##      fit    se.fit estimated_lower estimated_upper time.cat
## 1 26.540 0.7100723        25.14828        27.93172        1
## 2 13.522 1.0850488        11.39534        15.64866        2
## 3 15.514 1.1104660        13.33753        17.69047        3
## 4 20.762 1.3076268        18.19910        23.32490        4
```

```r
library(ggplot2)

qplot(x = time.cat, xlab = "Time",
      y = fit, ylab = "Mean",
      data = predictions) +

geom_errorbar(aes(
    ymin  = estimated_lower,
    ymax  = estimated_upper,
    width = 0.15))
```



In order to dismiss the correlations, we can use a linear model:

```r
lead.lm <- lm(y ~ time.cat, data = lead.uni)
pred.lm <- data.frame(predict(lead.lm, new.dat, interval = 'confidence'))
```

```r
pred.lm
```

```
##      fit      lwr      upr
## 1 26.540 24.41947 28.66053
## 2 13.522 11.40147 15.64253
## 3 15.514 13.39347 17.63453
## 4 20.762 18.64147 22.88253
```

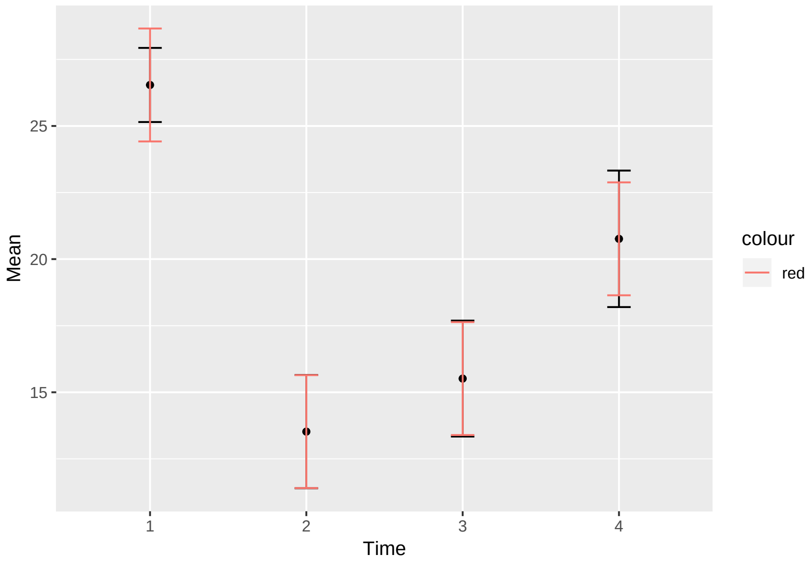
We see that the estimated means are the same, but because the standard errors are computed not taking into account the correlations, the boundries of the confidence intervals differ.

```r
qplot(x = time.cat, xlab = "Time",
      y = fit, ylab = "Mean",
      data = data.frame(predictions, pred.lm)) +

geom_errorbar(aes(
    ymin  = estimated_lower,
    ymax  = estimated_upper,
    width = 0.15)) +

geom_errorbar(aes(
    ymin  = lwr,
    ymax  = upr,
    width = 0.15,
```

```
    colour = "red"
    ))
```



Exercise 6

The difference of the means is:

```
predictions$fit[3] - predictions$fit[2]
```

```
## [1] 1.992
```

We can also compare the means using the t-test. The null hypotesis says that the difference of the means is equal to zero.

```
t.test(y ~ time.cat, data = lead.uni[lead.uni$time.cat %in% c(2,3),], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  y by time.cat
## t = -2.4721, df = 49, p-value = 0.01695
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.6113251 -0.3726749
## sample estimates:
## mean of the differences
##                  -1.992
```

P-value is less than 0.05, so for this significance level we reject the null hypotesis. Therefore we can assume that the difference in means is significant.