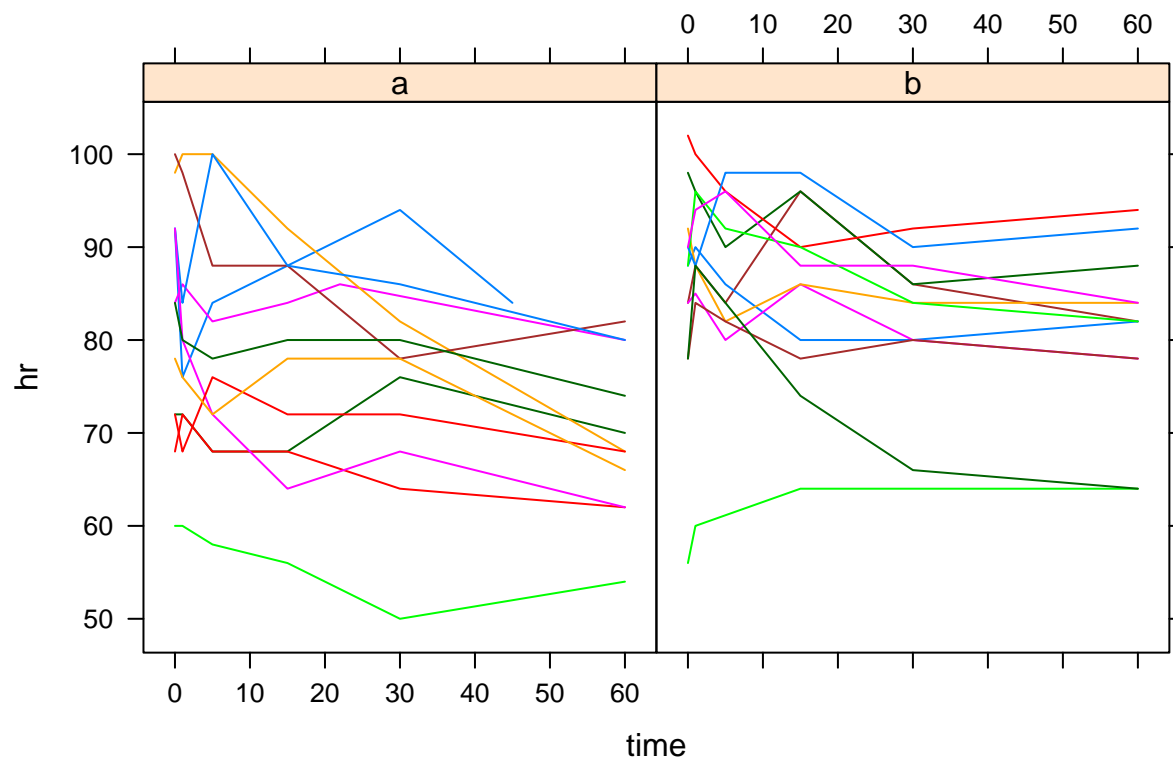


Complex Data - lab4

Stanisław Wilczyński

```
hrunbalanced <- read.csv(file = "../data/hrunbalanced.csv", header = TRUE)
means <- tapply(hrunbalanced$hr, list(hrunbalanced$time, hrunbalanced$drug), mean)
xyplot(hr~time|factor(drug), type='l', groups=id, data=hrunbalanced, main = "Spaghetti plot of the heart rate over time in 2 groups")
```

Spaghetti plot of the heart rate over time in 2 groups



```
## Random intercept
fit.RandInt <- lme(hr ~ time*factor(drug),
  random=~1|id,
  data=hrunbalanced)
```

```
fit.RandSlp <- lme(hr ~ time*factor(drug),
  random=~1+time|id,
  data=hrunbalanced)
```

```
anova(fit.RandInt, fit.RandSlp)
```

##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	fit.RandInt	1	6	945.3558	962.9193	-466.6779		
##	fit.RandSlp	2	8	937.4996	960.9176	-460.7498	1 vs 2	11.85621

```
summary(fit.RandSlp)
```

```
## Linear mixed-effects model fit by REML
## Data: hrunbalanced
```

```
##           AIC           BIC      logLik
##    937.4996 960.9176 -460.7498
##
## Random effects:
## Formula: ~1 + time | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 10.6367042 (Intr)
## time         0.1088219 -0.543
## Residual     4.3604148
##
## Fixed effects: hr ~ time * factor(drug)
##           Value Std.Error DF   t-value p-value
## (Intercept)  80.51400  3.145580 116 25.595917  0.0000
## time        -0.16445  0.040023 116 -4.108978  0.0001
## factor(drug)b  6.07479  4.449952  22  1.365135  0.1860
## time:factor(drug)b 0.05824  0.056411 116  1.032359  0.3041
## Correlation:
##           (Intr) time   fctr()
## time          -0.504
## factor(drug)b  -0.707  0.356
## time:factor(drug)b 0.358 -0.709 -0.505
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -2.51679655 -0.56562082  0.03649041  0.48976992  3.15964954
##
## Number of Observations: 142
## Number of Groups: 24

varCorr <- VarCorr(fit.RandSlp)

covar <- getVarCov(fit.RandSlp)

print(matrix(fit.RandSlp$coefficients$fixed, ncol = 1))

##           [,1]
## [1,] 80.51400239
## [2,] -0.16445398
## [3,]  6.07478669
## [4,]  0.05823689
```

Task 1

```
coefs <- fit.RandSlp$coefficients$fixed
coefs <- matrix(c(coefs[1], coefs[3], coefs[2], coefs[4]), ncol = 1)

ones <- matrix(rep(1,6), nrow=6)
zeros <- matrix(rep(0,6), nrow = 6)
tN <- matrix(c(0,1,5,15,30,60), nrow = 6)
t1 <- matrix(hrunbalanced$time[1:6], nrow = 6)
t2 <- matrix(hrunbalanced$time[7:12], nrow = 6)
t13 <- matrix(hrunbalanced$time[73:76], nrow = 4)
```

```
x1 <- cbind(ones,zeros,t1,zeros)
x2 <- cbind(ones,zeros, t2,zeros)
x13 <- cbind(ones[1:4,],ones[1:4,],t13,t13)
xNA <- cbind(ones, zeros, tN, zeros)
xNB <- cbind(ones, ones, tN, ones)
ZN <- cbind(ones,tN)
Z1 <- cbind(ones,t1)
Z2 <- cbind(ones, t2)
Z13 <- cbind(ones[1:4], t13)
```

Means:

```
mA <- xNA%*%coefs
mB <- xNB%*%coefs
m1 <- x1%*%coefs
m2 <- x2%*%coefs
m13 <- x13%*%coefs
knitr::kable(mA, caption = 'A')
```

```
## Warning in kable_markdown(x, padding = padding, ...): The table should have
## a header (column names)
```

Table 1: A

80.51400
80.34955
79.69173
78.04719
75.58038
70.64676

Covariances:

```
getCov <- function(Z){
  Z%*%covar%*%t(Z) + as.numeric(varCorr[3,1])*diag(1,nrow = dim(Z)[1], ncol = dim(Z)[1])
}
print(getCov(ZN))
```

##	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
## [1,]	132.15269	112.51142	109.99918	103.71859	94.29769	75.45591
## [2,]	112.51142	130.90842	109.43033	103.26816	94.02490	75.53839
## [3,]	109.99918	109.43033	126.16816	101.46645	92.93373	75.86828
## [4,]	103.71859	103.26816	101.46645	115.97541	90.20580	76.69301
## [5,]	94.29769	94.02490	92.93373	90.20580	105.12712	77.93011
## [6,]	75.45591	75.53839	75.86828	76.69301	77.93011	99.41753

```
print(getCov(Z1))
```

##	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
## [1,]	132.15269	112.51142	109.99918	103.71859	94.29769	84.87680
## [2,]	112.51142	130.90842	109.43033	103.26816	94.02490	84.78164
## [3,]	109.99918	109.43033	126.16816	101.46645	92.93373	84.40100
## [4,]	103.71859	103.26816	101.46645	115.97541	90.20580	83.44941
## [5,]	94.29769	94.02490	92.93373	90.20580	105.12712	82.02201
## [6,]	84.87680	84.78164	84.40100	83.44941	82.02201	99.60783

```
print(getCov(Z2))
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 132.15269 112.51142 109.99918 103.71859 99.32217 75.45591
## [2,] 112.51142 130.90842 109.43033 103.26816 98.95464 75.53839
## [3,] 109.99918 109.43033 126.16816 101.46645 97.48452 75.86828
## [4,] 103.71859 103.26816 101.46645 115.97541 93.80921 76.69301
## [5,] 99.32217 98.95464 97.48452 93.80921 110.24971 77.27033
## [6,] 75.45591 75.53839 75.86828 76.69301 77.27033 99.41753
```

```
print(getCov(Z13))
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 132.15269 112.51142 103.71859 75.45591
## [2,] 112.51142 130.90842 103.26816 75.53839
## [3,] 103.71859 103.26816 115.97541 76.69301
## [4,] 75.45591 75.53839 76.69301 99.41753
```

Task 2

```
## step one
coef.hr <- matrix(NA, ncol = 4, nrow = 0)
for (i in unique(hrunbalanced$id)) {
  model <- lm(hr~time, data = hrunbalanced, subset = id == i)
  model.drug <- unique(hrunbalanced[hrunbalanced$id == i, "drug"])
  coef.hr <- rbind(coef.hr, c(i, model.drug, coef(model)))
}
colnames(coef.hr) <- c("id", "drug", "intercept", "slope")
coef.hr <- as.data.frame(coef.hr)
coef.hr
```

```
##   id drug intercept      slope
## 1  1    1  85.08455  0.078048780
## 2  2    1  84.80138 -0.066099604
## 3  3    1  70.78740  0.011492122
## 4  4    1  69.50324 -0.135310473
## 5  5    1 100.08156 -0.544949027
## 6  6    1  58.54853 -0.119740500
## 7  7    1  94.21909 -0.282113068
## 8  8    1  91.90646 -0.193141798
## 9  9    1  79.46728 -0.349582947
## 10 10    1  81.36336 -0.109731233
## 11 11    1  72.36206 -0.055607044
## 12 12    1  77.28650 -0.141612604
## 13 13    2  59.21327  0.094038623
## 14 14    2  87.84628 -0.063762743
## 15 15    2  86.53482 -0.118999073
## 16 16    2  83.98752 -0.098424467
## 17 17    2  95.09719 -0.149397590
## 18 18    2  97.64869 -0.107136237
## 19 19    2  87.35792 -0.073401297
## 20 20    2  92.06833 -0.183873957
## 21 21    2  80.93272 -0.050417053
## 22 22    2  92.79011 -0.006672845
```

```
## 23 23      2  92.85301 -0.154216867
## 24 24      2  82.25740 -0.356255792

### step two
hr.int <- lm(intercept~drug, data = coef.hr)
hr.slp <- lm(slope~drug, data=coef.hr)

summary(hr.int)

##
## Call:
## lm(formula = intercept ~ drug, data = coef.hr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.3357  -4.6227   0.8607   6.2569  19.6306
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   74.353      7.055  10.539 4.61e-10 ***
## drug           6.098      4.462   1.367   0.186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.93 on 22 degrees of freedom
## Multiple R-squared:  0.07825,    Adjusted R-squared:  0.03635
## F-statistic: 1.868 on 1 and 22 DF,  p-value: 0.1856

summary(hr.slp)

##
## Call:
## lm(formula = slope ~ drug, data = coef.hr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38592 -0.04489  0.02057  0.06470  0.23708
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.21235    0.09128  -2.326   0.0296 *
## drug         0.05332    0.05773   0.924   0.3657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1414 on 22 degrees of freedom
## Multiple R-squared:  0.03732,    Adjusted R-squared:  -0.006434
## F-statistic: 0.853 on 1 and 22 DF,  p-value: 0.3657
```

te 4 wartosci do tabelki dla two-staged mozna wziac jakos wpolczynniki z intercepts (pierwsze 2) i sloper (drugie 2)