Exemplos Estimadores GLS e GEE

José Luiz Padilha da Silva

15 de agosto de 2018

Exemplo: Chumbo em Crianças

O estudo, *Tratamento de Crianças Expostas ao chumbo (TLC)*, foi um ensaio clínico aleatorizado envolvendo crianças com níveis de chumbo no sangue entre $20 - 44$ microgramas/dL. Os grupos de comparação são placebo e um tratamento ativo. Os dados consistem de quatro medidas repetidas de níveis de chumbo no sangue obtidos na linha de base (semana 0), semana 1, semana 4 e semana 6 em 100 crianças aleatoriamente alocadas entre os dois grupos.

Análise Exploratória

```
library(reshape); library(plyr); library(nlme); library(ggplot2);
datawide<-read.table("chumbo.txt",header=T)
head(datawide)
```

```
##   ID Grupo Sem0 Sem1 Sem4 Sem6
## 1  1     P 30.8 26.9 25.8 23.8
## 2  2     A 26.5 14.8 19.5 21.0
## 3  3     A 25.8 23.0 19.1 23.2
## 4  4     P 24.7 24.5 22.0 22.5
## 5  5     A 20.4  2.8  3.2  9.4
## 6  6     A 20.4  5.4  4.5 11.9
```

Os dados estão no formato *largo*, no qual cada indivíduo é representado por uma linha e as medidas repetidas são apresentadas por colunas. A seguir um resumo das observações por grupo.

```
summary(subset(datawide, Grupo=="A")[,3:6]) #Grupo tratamento
```

```
##       Sem0            Sem1             Sem4             Sem6
##  Min.   :19.70   Min.   : 2.800   Min.   : 3.000   Min.   : 4.10
##  1st Qu.:22.12   1st Qu.: 7.225   1st Qu.: 9.125   1st Qu.:15.40
##  Median :26.20   Median :12.250   Median :15.350   Median :18.85
##  Mean   :26.54   Mean   :13.522   Mean   :15.514   Mean   :20.76
##  3rd Qu.:29.55   3rd Qu.:17.500   3rd Qu.:19.725   3rd Qu.:23.75
##  Max.   :41.10   Max.   :39.000   Max.   :40.400   Max.   :63.90
```

```
summary(subset(datawide, Grupo=="P")[,3:6]) #Grupo controle
```

```
##       Sem0            Sem1            Sem4            Sem6
##  Min.   :19.70   Min.   :14.90   Min.   :15.30   Min.   :13.50
##  1st Qu.:21.88   1st Qu.:20.93   1st Qu.:19.82   1st Qu.:19.95
##  Median :25.25   Median :24.10   Median :22.45   Median :22.35
##  Mean   :26.27   Mean   :24.66   Mean   :24.07   Mean   :23.65
##  3rd Qu.:29.73   3rd Qu.:27.82   3rd Qu.:27.45   3rd Qu.:27.50
##  Max.   :38.10   Max.   :40.80   Max.   :38.60   Max.   :43.30
```

Nota-se uma pequena diminuição das médias dos níveis de chumbo no grupo controle ao longo do tempo. Para o grupo tratamento há um grande decréscimo do baseline para a primeira semana e subsequentes aumentos nas semanas seguintes.

Temos as seguintes estimativas para as correlações:

```
round(cor(datawide[,3:6]),2) #Todos os indivíduos
```

```
##      Sem0 Sem1 Sem4 Sem6
## Sem0 1.00 0.42 0.47 0.56
## Sem1 0.42 1.00 0.84 0.56
## Sem4 0.47 0.84 1.00 0.58
## Sem6 0.56 0.56 0.58 1.00
```

```
round(cor(subset(datawide, Grupo=="A")[,3:6]),2) # Grupo tratamento
```

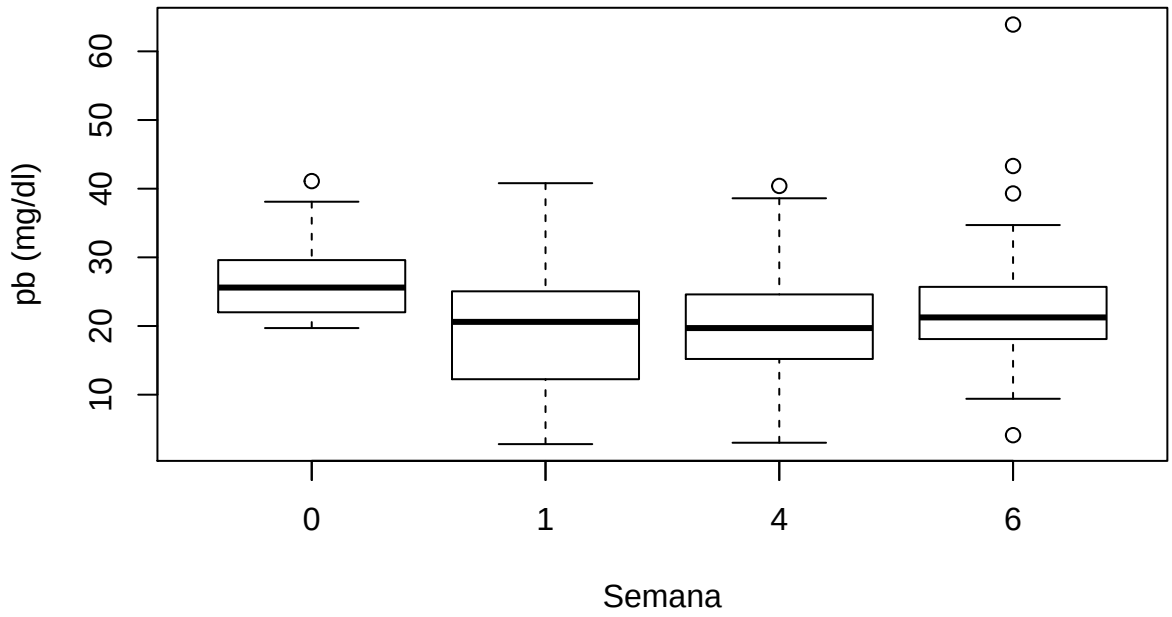
```
##      Sem0 Sem1 Sem4 Sem6
## Sem0 1.00 0.40 0.38 0.50
## Sem1 0.40 1.00 0.73 0.51
## Sem4 0.38 0.73 1.00 0.45
## Sem6 0.50 0.51 0.45 1.00
```

```
round(cor(subset(datawide, Grupo=="P")[,3:6]),2) # Grupo controle
```

```
##      Sem0 Sem1 Sem4 Sem6
## Sem0 1.00 0.83 0.84 0.76
## Sem1 0.83 1.00 0.86 0.76
## Sem4 0.84 0.86 1.00 0.87
## Sem6 0.76 0.76 0.87 1.00
```

A seguir os boxplots marginais

```
with(datawide, boxplot(Sem0,Sem1,Sem4,Sem6,ylab="pb (mg/dl)",xlab="Semana"))
axis(1, 1:4, c(0,1,4,6))
```



Devemos ter cuidado pois o boxplot não considera a estrutura longitudinal dos dados.

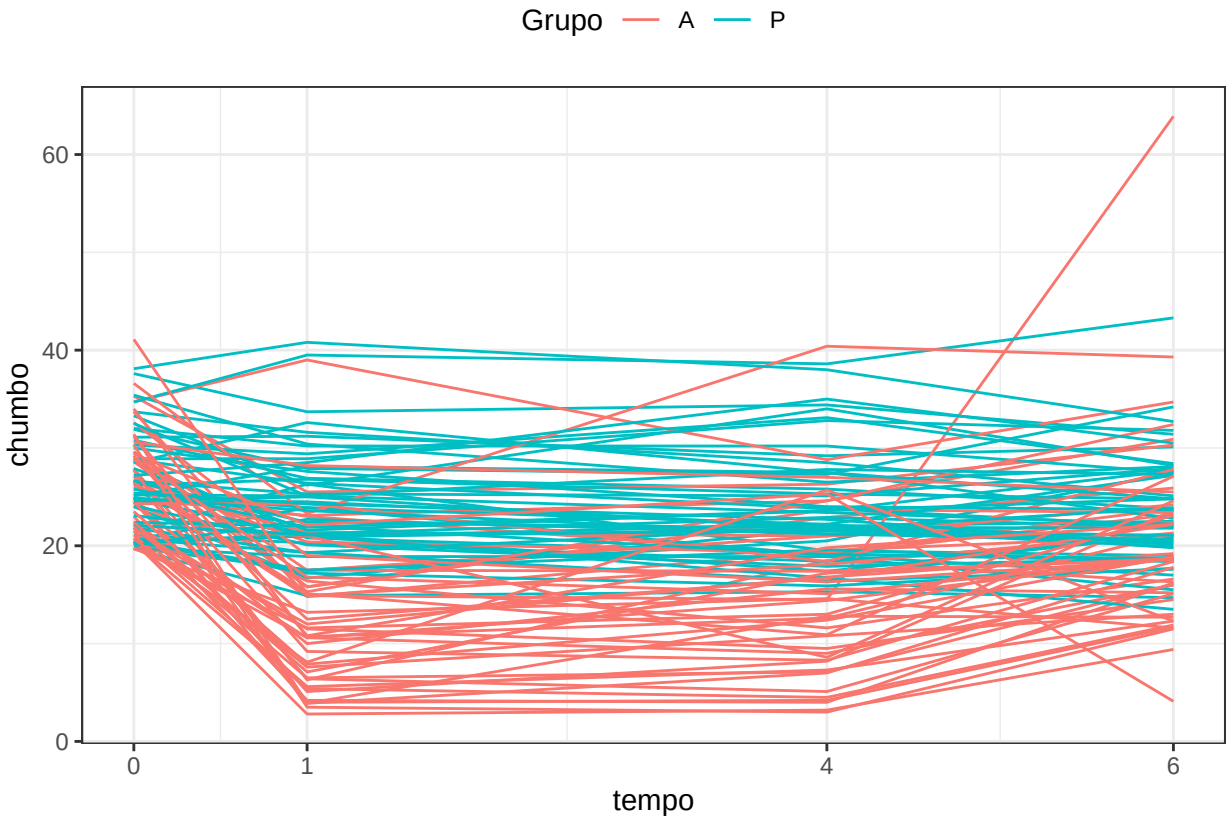
É mais conveniente trabalhar com os dados no formato *longo*, no qual cada variável é representada por uma coluna e temos uma linha para cada medida repetida do indivíduos. Vamos usar a função **reshape** do pacote de mesmo nome.

```
datalong<-reshape(data=datawide,direction="long", idvar="ID", v.names="chumbo",
                  varying = list(names(datawide)[3:6]), time= c(0,1,4,6), timevar="tempo")
datalong=arrange(datalong, ID) #Ordenamos os dados por ID, função do pacote plyr
head(datalong, 8)
```

```
##   ID Grupo tempo chumbo
## 1  1     P     0   30.8
## 2  1     P     1   26.9
## 3  1     P     4   25.8
## 4  1     P     6   23.8
## 5  2     A     0   26.5
## 6  2     A     1   14.8
## 7  2     A     4   19.5
## 8  2     A     6   21.0
```

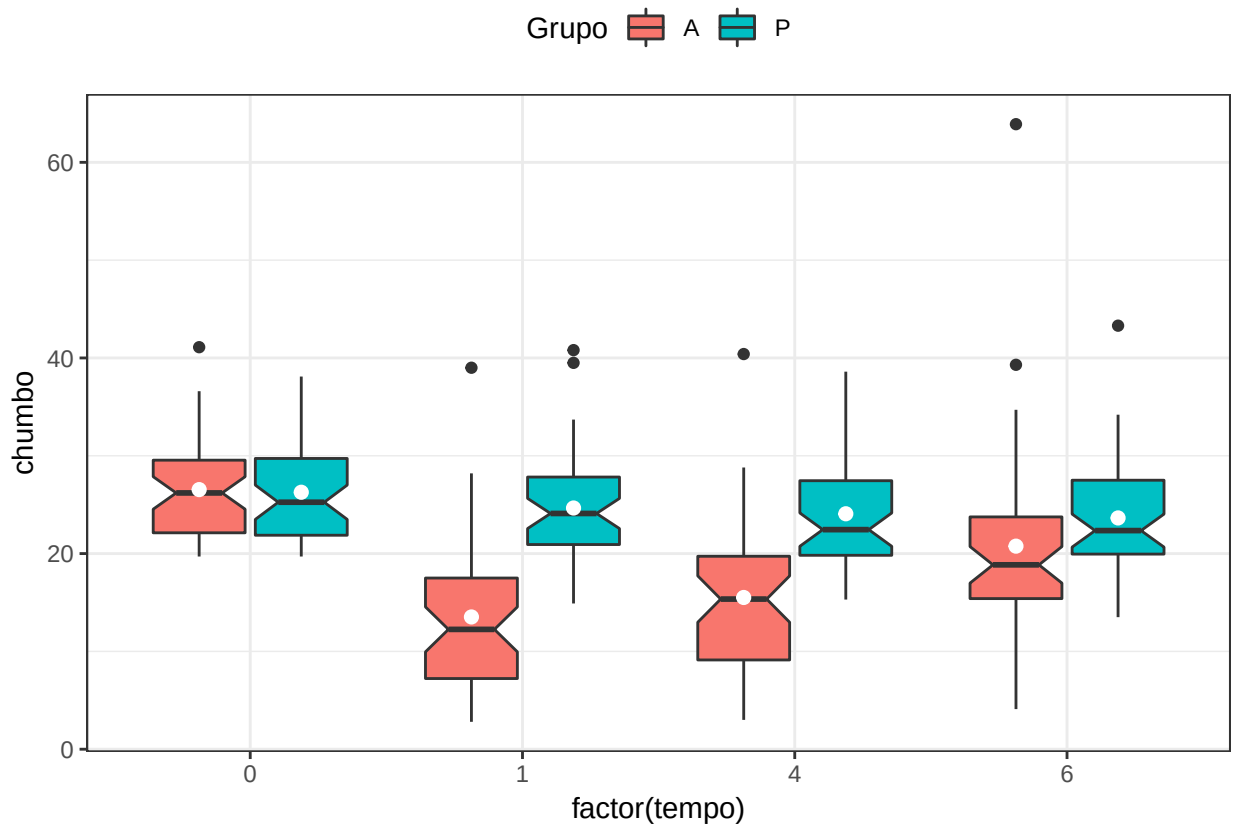
A representação gráfica mais interessante em nível individual é o gráfico de perfis:

```
p1=ggplot(datalong, aes(x=tempo, y=chumbo, color=Grupo)) +  theme_bw() +
   geom_line(aes(group=ID)) + theme(legend.position="top") +
   scale_x_continuous(breaks=unique(datalong$tempo))
p1
```



Podemos examinar as diferenças dentro de cada tempo por meio de boxplots:

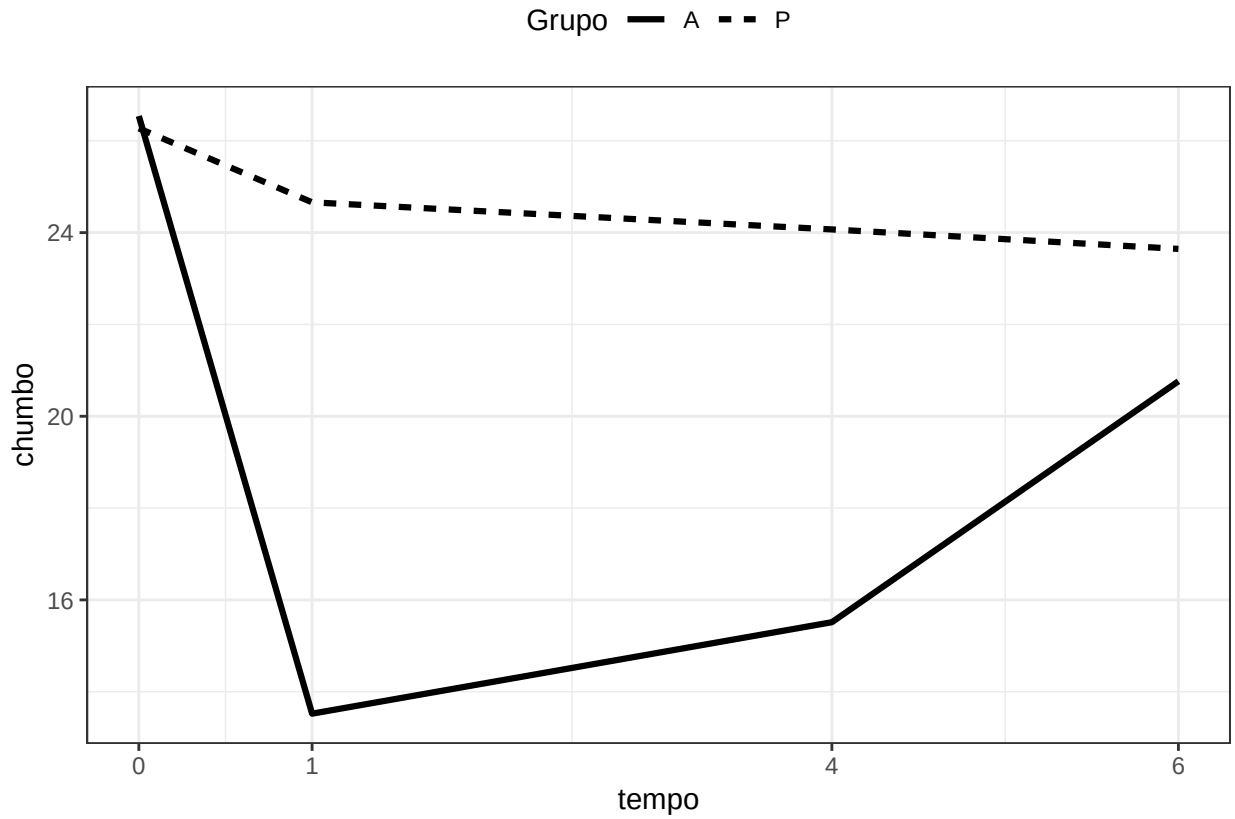
```
p2=ggplot(datalong, aes(x=factor(tempo), y=chumbo, fill=Grupo)) + geom_boxplot(notch=TRUE) +
   theme_bw() + theme(legend.position="top") + stat_summary(fun.y="mean", geom="point", size=2,
    position=position_dodge(width=0.75), color="white", show.legend=FALSE)
p2
```



Da ajuda do `geom_boxplot`: *In a notched box plot, the notches extend '1.58IQR / sqrt(n)'. This gives a roughly 95% confidence interval for comparing medians. See McGill et al. (1978) for more details.*

Os valores centrais em branco representam as médias. Uma representação mais destacada para as médias pode ser obtida fazendo

```
p3=ggplot(datalong, aes(x=tempo, y=chumbo, group = Grupo, shape = Grupo)) + theme_bw() +
        stat_summary(fun.y="mean",geom="line", size=1.1, aes(linetype = Grupo)) +
        theme(legend.position="top") + scale_x_continuous(breaks=unique(datalong$tempo))
p3
```



Como vimos, as maiores diferenças ocorrem no tempo 1 e vão diminuindo ao longo das semanas. Passaremos aos ajustes dos modelos por mínimos quadrados generalizados. Consideraremos as estruturas de correlação do tipo *independente*, *simetria composta*, *AR(1)* e *não estruturada*.

Estimador GLS

Modelo 1: Modelo linear de efeito fixo (com intercepto)

O primeiro modelo é da forma `chumbo ~ tempo*Grupo` e tem a seguinte representação:

$$\begin{aligned} E(Y_{ij}) =& \beta_1 + \beta_2 I(tempo_j = 1) + \beta_3 I(tempo_j = 4) + \beta_4 I(tempo_j = 6) + \beta_5 I(Grupo_i = P) + \\ & \beta_6 I(Grupo_i = P) \times I(tempo_j = 1) + \beta_7 I(Grupo_i = P) \times I(tempo_j = 4) + \\ & \beta_8 I(Grupo_i = P) \times I(tempo_j = 6). \end{aligned}$$

Fazemos o ajuste no `R` através dos seguintes comandos:

```
datalong$tempo=as.factor(datalong$tempo)
gls1.ind<-lm(chumbo ~ tempo*Grupo, data=datalong) #Independente
gls1.exch<-gls(chumbo ~ tempo*Grupo, correlation=corCompSymm(form=~1|ID),
               data=datalong) #Simetria composta
gls1.ar1<-gls(chumbo ~ tempo*Grupo, correlation=corAR1(form=~1|ID),
              data=datalong) #AR(1)
gls1.unst<-gls(chumbo ~ tempo*Grupo, correlation=corSymm(form=~1|ID),
               data=datalong) #Não estruturada
```

Estamos interessados nos quatro últimos coeficientes, que estão relacionados às comparações entre os grupos dentro de cada tempo.

```r
# Independente
round(summary(gls1.ind)$coef,3)
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     26.540      0.937  28.324    0.000
## tempo1         -13.018      1.325  -9.824    0.000
## tempo4         -11.026      1.325  -8.321    0.000
## tempo6          -5.778      1.325  -4.360    0.000
## GrupoP          -0.268      1.325  -0.202    0.840
## tempo1:GrupoP   11.406      1.874   6.086    0.000
## tempo4:GrupoP    8.824      1.874   4.709    0.000
## tempo6:GrupoP    3.152      1.874   1.682    0.093
```

```r
# Simetria composta
round(coef(summary(gls1.exch)),3)
```

```
##                 Value Std.Error t-value p-value
## (Intercept)    26.540     0.937  28.324   0.000
## tempo1        -13.018     0.843 -15.445   0.000
## tempo4        -11.026     0.843 -13.082   0.000
## tempo6         -5.778     0.843  -6.855   0.000
## GrupoP         -0.268     1.325  -0.202   0.840
## tempo1:GrupoP  11.406     1.192   9.569   0.000
## tempo4:GrupoP   8.824     1.192   7.403   0.000
## tempo6:GrupoP   3.152     1.192   2.644   0.009
```

```r
# AR(1)
round(coef(summary(gls1.ar1)),3)
```

```
##                 Value Std.Error t-value p-value
## (Intercept)    26.540     0.932  28.482   0.000
## tempo1        -13.018     0.801 -16.261   0.000
## tempo4        -11.026     1.022 -10.785   0.000
## tempo6         -5.778     1.140  -5.067   0.000
## GrupoP         -0.268     1.318  -0.203   0.839
## tempo1:GrupoP  11.406     1.132  10.075   0.000
## tempo4:GrupoP   8.824     1.446   6.103   0.000
## tempo6:GrupoP   3.152     1.613   1.955   0.051
```

```r
# Não estruturada
round(coef(summary(gls1.unst)),3)
```

```
##                 Value Std.Error t-value p-value
## (Intercept)    26.540     0.937  28.310   0.000
## tempo1        -13.018     0.843 -15.450   0.000
## tempo4        -11.026     0.858 -12.856   0.000
## tempo6         -5.778     0.903  -6.396   0.000
## GrupoP         -0.268     1.326  -0.202   0.840
## tempo1:GrupoP  11.406     1.192   9.572   0.000
## tempo4:GrupoP   8.824     1.213   7.275   0.000
## tempo6:GrupoP   3.152     1.278   2.467   0.014
```

Modelo 2: Modelo linear de efeito fixo (sem intercepto)

O segundo modelo é da forma `chumbo ~ tempo*Grupo - Grupo - 1` e tem a seguinte representação:

$$\begin{aligned} E(Y_{ij}) =& \beta_1 I(tempo_j = 0) + \beta_2 I(tempo_j = 1) + \beta_3 I(tempo_j = 4) + \beta_4 I(tempo_j = 6) + \\ & \beta_5 I(Grupo_i = P) \times I(tempo_j = 0) + \beta_6 I(Grupo_i = P) \times I(tempo_j = 1) + \\ & \beta_7 I(Grupo_i = P) \times I(tempo_j = 4) + \beta_8 I(Grupo_i = P) \times I(tempo_j = 6). \end{aligned}$$

Nessa parametrização é mais simples de comparar os grupos em cada tempo.

```
gls2.ind<-lm(chumbo ~ tempo*Grupo - Grupo - 1, data=datalong) #Independente
gls2.exch<-gls(chumbo ~ tempo*Grupo - Grupo - 1,
               correlation=corCompSymm (form=~1|ID), data=datalong) #Simetria Composta
gls2.ar1<-gls(chumbo ~ tempo*Grupo - Grupo - 1,
              correlation=corAR1 (form=~1|ID), data=datalong) #AR1
gls2.unst<-gls(chumbo ~ tempo*Grupo - Grupo - 1,
               correlation=corSymm (form=~1|ID), data=datalong) #Não estruturada
```

```
# Independente
round(summary(gls2.ind)$coef,3)
```

```
##               Estimate Std. Error t value Pr(>|t|)
## tempo0          26.540      0.937  28.324     0.00
## tempo1          13.522      0.937  14.431     0.00
## tempo4          15.514      0.937  16.557     0.00
## tempo6          20.762      0.937  22.158     0.00
## tempo0:GrupoP   -0.268      1.325  -0.202     0.84
## tempo1:GrupoP   11.138      1.325   8.405     0.00
## tempo4:GrupoP    8.556      1.325   6.457     0.00
## tempo6:GrupoP    2.884      1.325   2.176     0.03
```

```
# Simetria composta
round(coef(summary(gls2.exch)),3)
```

```
##                Value Std.Error t-value p-value
## tempo0        26.540     0.937  28.324    0.00
## tempo1        13.522     0.937  14.431    0.00
## tempo4        15.514     0.937  16.557    0.00
## tempo6        20.762     0.937  22.158    0.00
## tempo0:GrupoP -0.268     1.325  -0.202    0.84
## tempo1:GrupoP 11.138     1.325   8.405    0.00
## tempo4:GrupoP  8.556     1.325   6.457    0.00
## tempo6:GrupoP  2.884     1.325   2.176    0.03
```

```
# AR(1)
round(coef(summary(gls2.ar1)),3)
```

```
##                Value Std.Error t-value p-value
## tempo0        26.540     0.932  28.482   0.000
## tempo1        13.522     0.932  14.512   0.000
## tempo4        15.514     0.932  16.649   0.000
## tempo6        20.762     0.932  22.282   0.000
## tempo0:GrupoP -0.268     1.318  -0.203   0.839
## tempo1:GrupoP 11.138     1.318   8.452   0.000
## tempo4:GrupoP  8.556     1.318   6.493   0.000
## tempo6:GrupoP  2.884     1.318   2.189   0.029
```

```r
# Não estruturada
round(coef(summary(gls2.unst)),3)
```

```
##                Value Std.Error t-value p-value
## tempo0        26.540     0.937  28.310    0.00
## tempo1        13.522     0.937  14.424    0.00
## tempo4        15.514     0.937  16.549    0.00
## tempo6        20.762     0.937  22.147    0.00
## tempo0:GrupoP -0.268     1.326  -0.202    0.84
## tempo1:GrupoP 11.138     1.326   8.401    0.00
## tempo4:GrupoP  8.556     1.326   6.454    0.00
## tempo6:GrupoP  2.884     1.326   2.175    0.03
```

Estimador GEE

Modelo 1: Modelo linear de efeito fixo (com intercepto)

Para fazer o ajuste GEE no R podemos utilizar os seguintes comandos:

```r
library(gee)
datalong$tempo=as.factor(datalong$tempo); datalong$Grupo=as.factor(datalong$Grupo)
gee1.ind<-gee(chumbo ~ tempo*Grupo, corstr="independence", id=ID,
                family="gaussian", data=datalong) #Independente
```

```
##   (Intercept)        tempo1        tempo4        tempo6        GrupoP
##        26.540       -13.018       -11.026        -5.778        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.406         8.824         3.152
```

```r
gee1.exch<-gee(chumbo ~ tempo*Grupo, corstr="exchangeable", id=ID,
                 family="gaussian", data=datalong) #Simetria composta
```

```
##   (Intercept)        tempo1        tempo4        tempo6        GrupoP
##        26.540       -13.018       -11.026        -5.778        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.406         8.824         3.152
```

```r
gee1.ar1<-gee(chumbo ~ tempo*Grupo, corstr="AR-M", id=ID,
                data=datalong) #AR(1)
```

```
##   (Intercept)        tempo1        tempo4        tempo6        GrupoP
##        26.540       -13.018       -11.026        -5.778        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.406         8.824         3.152
```

```r
gee1.unst<-gee(chumbo ~ tempo*Grupo, corstr="unstructured", id=ID,
                 data=datalong) #Não estruturada
```

```
##   (Intercept)        tempo1        tempo4        tempo6        GrupoP
##        26.540       -13.018       -11.026        -5.778        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.406         8.824         3.152
```

Estamos interessados nos quatro últimos coeficientes, que estão relacionados às comparações entre os grupos dentro de cada tempo.

```r
# Independente
round(coef(summary(gee1.ind)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)     26.540      0.937  28.324       0.703   37.756
## tempo1         -13.018      1.325  -9.824       1.021  -12.755
## tempo4         -11.026      1.325  -8.321       1.053  -10.469
## tempo6          -5.778      1.325  -4.360       1.126   -5.130
## GrupoP          -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.406      1.874   6.086       1.109   10.288
## tempo4:GrupoP    8.824      1.874   4.709       1.141    7.734
## tempo6:GrupoP    3.152      1.874   1.682       1.244    2.534
```

```r
# Simetria composta
round(coef(summary(gee1.exch)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)     26.540      0.937  28.324       0.703   37.756
## tempo1         -13.018      0.847 -15.369       1.021  -12.755
## tempo4         -11.026      0.847 -13.017       1.053  -10.469
## tempo6          -5.778      0.847  -6.821       1.126   -5.130
## GrupoP          -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.406      1.198   9.522       1.109   10.288
## tempo4:GrupoP    8.824      1.198   7.366       1.141    7.734
## tempo6:GrupoP    3.152      1.198   2.631       1.244    2.534
```

```r
# AR(1)
round(coef(summary(gee1.ar1)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)     26.540      0.937  28.324       0.703   37.756
## tempo1         -13.018      0.787 -16.544       1.021  -12.755
## tempo4         -11.026      1.010 -10.917       1.053  -10.469
## tempo6          -5.778      1.131  -5.108       1.126   -5.130
## GrupoP          -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.406      1.113  10.250       1.109   10.288
## tempo4:GrupoP    8.824      1.428   6.178       1.141    7.734
## tempo6:GrupoP    3.152      1.600   1.970       1.244    2.534
```

```r
# Não estruturada
round(coef(summary(gee1.unst)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)     26.540      0.937  28.324       0.703   37.756
## tempo1         -13.018      0.996 -13.072       1.021  -12.755
## tempo4         -11.026      0.984 -11.207       1.053  -10.469
## tempo6          -5.778      0.932  -6.202       1.126   -5.130
## GrupoP          -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.406      1.408   8.099       1.109   10.288
## tempo4:GrupoP    8.824      1.391   6.342       1.141    7.734
## tempo6:GrupoP    3.152      1.318   2.392       1.244    2.534
```

Note a diferença entre as estimadores dos erros padrões nas versões naive e robusta (baseada no estimador sanduíche).

Modelo 2: Modelo linear de efeito fixo (sem intercepto)

```
gee2.ind<-gee(chumbo ~ tempo*Grupo - Grupo - 1, corstr="independence", id=ID,
                family="gaussian", data=datalong) #Independente
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
```

```
##        tempo0        tempo1        tempo4        tempo6 tempo0:GrupoP
##        26.540        13.522        15.514        20.762        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.138         8.556         2.884
```

```
gee2.exch<-gee(chumbo ~ tempo*Grupo - Grupo - 1, corstr="exchangeable", id=ID,
                 family="gaussian", data=datalong) #Simetria composta
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate
```

```
##        tempo0        tempo1        tempo4        tempo6 tempo0:GrupoP
##        26.540        13.522        15.514        20.762        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.138         8.556         2.884
```

```
gee2.ar1<-gee(chumbo ~ tempo*Grupo - Grupo - 1, corstr="AR-M", id=ID, data=datalong) #AR(1)
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate
```

```
##        tempo0        tempo1        tempo4        tempo6 tempo0:GrupoP
##        26.540        13.522        15.514        20.762        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.138         8.556         2.884
```

```
gee2.unst<-gee(chumbo ~ tempo*Grupo - Grupo - 1, corstr="unstructured", id=ID,
                 data=datalong) #Não estruturada
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate
```

```
##        tempo0        tempo1        tempo4        tempo6 tempo0:GrupoP
##        26.540        13.522        15.514        20.762        -0.268
## tempo1:GrupoP tempo4:GrupoP tempo6:GrupoP
##        11.138         8.556         2.884
```

As estimativas são então dadas por:

```
# Independente
round(coef(summary(gee2.ind)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## tempo0          26.540      0.937  28.324       0.703   37.756
## tempo1          13.522      0.937  14.431       1.074   12.589
## tempo4          15.514      0.937  16.557       1.099   14.113
## tempo6          20.762      0.937  22.158       1.294   16.039
## tempo0:GrupoP   -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.138      1.325   8.405       1.318    8.448
## tempo4:GrupoP    8.556      1.325   6.457       1.363    6.278
## tempo6:GrupoP    2.884      1.325   2.176       1.516    1.902
```

```r
# Simetria composta
round(coef(summary(gee2.exch)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## tempo0          26.540      0.937  28.324       0.703   37.756
## tempo1          13.522      0.937  14.431       1.074   12.589
## tempo4          15.514      0.937  16.557       1.099   14.113
## tempo6          20.762      0.937  22.158       1.294   16.039
## tempo0:GrupoP   -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.138      1.325   8.405       1.318    8.448
## tempo4:GrupoP    8.556      1.325   6.457       1.363    6.278
## tempo6:GrupoP    2.884      1.325   2.176       1.516    1.902
```

```r
# AR(1)
round(coef(summary(gee2.ar1)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## tempo0          26.540      0.937  28.324       0.703   37.756
## tempo1          13.522      0.937  14.431       1.074   12.589
## tempo4          15.514      0.937  16.557       1.099   14.113
## tempo6          20.762      0.937  22.158       1.294   16.039
## tempo0:GrupoP   -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.138      1.325   8.405       1.318    8.448
## tempo4:GrupoP    8.556      1.325   6.457       1.363    6.278
## tempo6:GrupoP    2.884      1.325   2.176       1.516    1.902
```

```r
# Não estruturada
round(coef(summary(gee2.unst)),3)
```

```
##               Estimate Naive S.E. Naive z Robust S.E. Robust z
## tempo0          26.540      0.937  28.324       0.703   37.756
## tempo1          13.522      0.937  14.431       1.074   12.589
## tempo4          15.514      0.937  16.557       1.099   14.113
## tempo6          20.762      0.937  22.158       1.294   16.039
## tempo0:GrupoP   -0.268      1.325  -0.202       0.994   -0.270
## tempo1:GrupoP   11.138      1.325   8.405       1.318    8.448
## tempo4:GrupoP    8.556      1.325   6.457       1.363    6.278
## tempo6:GrupoP    2.884      1.325   2.176       1.516    1.902
```

Exemplo: Dados de Crescimento

Potthoff & Roy (1964) apresentaram um conjunto de dados de crescimento de 11 meninas e 16 meninos. As medidas referem-se à distância entre dois marcos faciais (do centro da pituitária à fissura do maxilar) em quatro idades (8, 10, 12 e 14 anos). O objetivo é descrever e comparar o crescimento de meninos e meninas.

Análise Exploratória

Os dados estão disponíveis no `R` no pacote `mice` e podem ser acessados como:

```
library(mice)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'mice'
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
data(potthoffroy)
head(potthoffroy)
```

```
##   id sex   d8  d10  d12  d14
## 1  1   F 21.0 20.0 21.5 23.0
## 2  2   F 21.0 21.5 24.0 25.5
## 3  3   F 20.5 24.0 24.5 26.0
## 4  4   F 23.5 24.5 25.0 26.5
## 5  5   F 21.5 23.0 22.5 23.5
## 6  6   F 20.0 21.0 21.0 22.5
```

A seguir um resumo dos dados por sexo:

```
with(potthoffroy,by(potthoffroy[,-c(1,2)],sex,summary,digits=3))
```

```
## sex: F
##        d8            d10            d12            d14
##  Min.   :16.5   Min.   :19.0   Min.   :19.0   Min.   :19.5
##  1st Qu.:20.2   1st Qu.:21.0   1st Qu.:21.8   1st Qu.:22.8
##  Median :21.0   Median :22.5   Median :23.0   Median :24.0
##  Mean   :21.2   Mean   :22.2   Mean   :23.1   Mean   :24.1
##  3rd Qu.:22.2   3rd Qu.:23.5   3rd Qu.:24.2   3rd Qu.:25.8
##  Max.   :24.5   Max.   :25.0   Max.   :28.0   Max.   :28.0
## ------------------------------------------------------------
## sex: M
##        d8            d10            d12            d14
##  Min.   :17.0   Min.   :20.5   Min.   :22.5   Min.   :25.0
##  1st Qu.:21.9   1st Qu.:22.4   1st Qu.:23.9   1st Qu.:26.0
##  Median :23.0   Median :23.5   Median :25.0   Median :26.8
##  Mean   :22.9   Mean   :23.8   Mean   :25.7   Mean   :27.5
##  3rd Qu.:24.1   3rd Qu.:25.1   3rd Qu.:26.6   3rd Qu.:28.8
##  Max.   :27.5   Max.   :28.0   Max.   :31.0   Max.   :31.5
```

Notamos que as meninas possuem menores valores médios que os meninos. As correlações marginais são dadas a seguir no geral e por sexo.

```r
cor(potthoffroy[,-c(1:2)])
```

```
##            d8       d10       d12       d14
## d8  1.0000000 0.6255833 0.7108079 0.5998338
## d10 0.6255833 1.0000000 0.6348775 0.7593268
## d12 0.7108079 0.6348775 1.0000000 0.7949980
## d14 0.5998338 0.7593268 0.7949980 1.0000000
```

Os dados mostram forte correlação positiva.

```r
with(potthoffroy,by(potthoffroy[,-c(1,2)],sex,cor))
```

```
## sex: F
##            d8       d10       d12       d14
## d8  1.0000000 0.8300900 0.8623146 0.8413558
## d10 0.8300900 1.0000000 0.8954156 0.8794236
## d12 0.8623146 0.8954156 1.0000000 0.9484070
## d14 0.8413558 0.8794236 0.9484070 1.0000000
## ------------------------------------------------------------
## sex: M
##            d8       d10       d12       d14
## d8  1.0000000 0.4373932 0.5579310 0.3152311
## d10 0.4373932 1.0000000 0.3872909 0.6309234
## d12 0.5579310 0.3872909 1.0000000 0.5859866
## d14 0.3152311 0.6309234 0.5859866 1.0000000
```

Contudo, as meninas apresentam correlação entre as medidas repetidas consideravelmente maiores que os meninos. Além disso, as correlações para o grupo dos meninos é comparativamente mais variável enquanto para as meninas é mais homogênea.

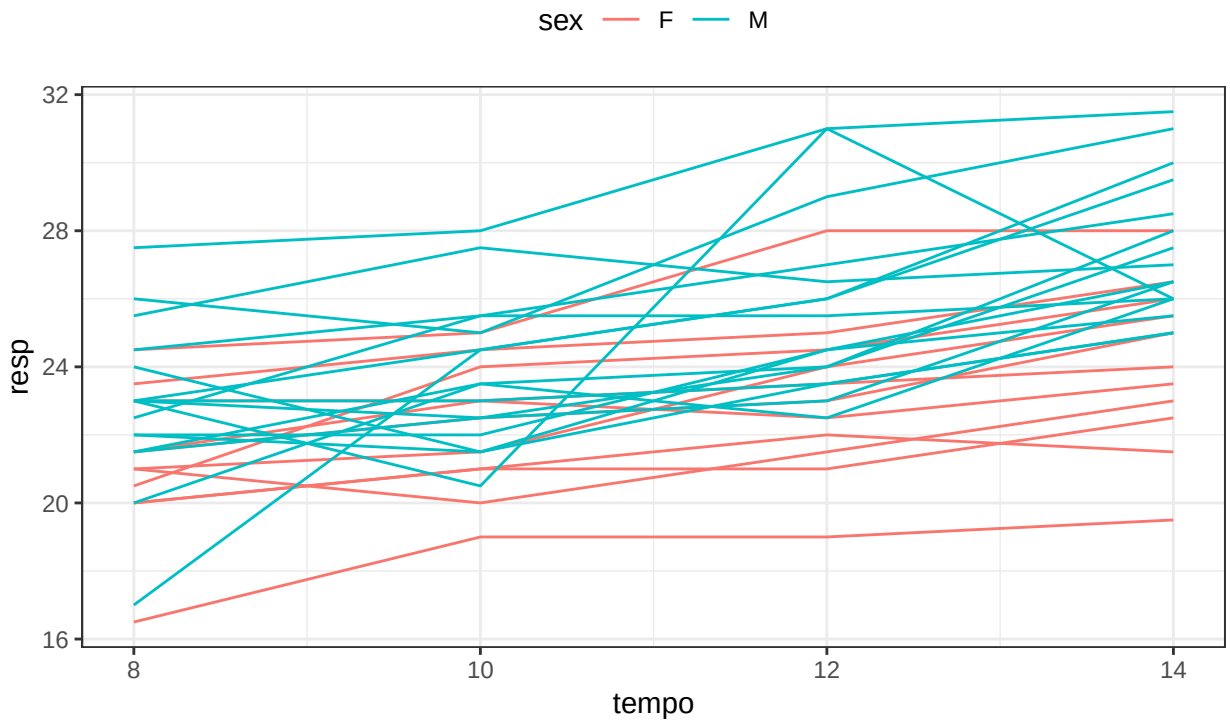
A seguir transformamos os dados para o formato longo.

```r
dados=reshape(data=potthoffroy,direction="long", idvar="id", v.names="resp",
              varying = list(names(potthoffroy)[3:6]), time= c(8,10,12,14), timevar="tempo")
dados=arrange(dados, id) #Ordenamos os dados por ID, função do pacote plyr
head(dados, 8)
```

```
##   id sex tempo resp
## 1  1   F     8 21.0
## 2  1   F    10 20.0
## 3  1   F    12 21.5
## 4  1   F    14 23.0
## 5  2   F     8 21.0
## 6  2   F    10 21.5
## 7  2   F    12 24.0
## 8  2   F    14 25.5
```

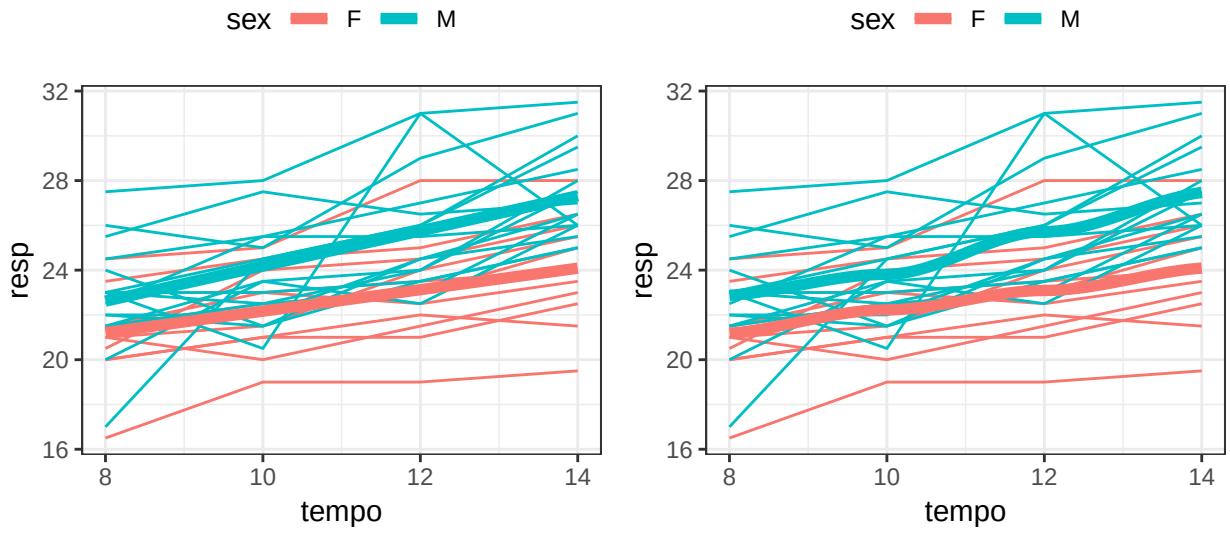
Na sequência o gráfico de perfis:

```r
p1=ggplot(dados, aes(x=tempo, y=resp, color=sex)) + theme_bw() +
   geom_line(aes(group=id)) + theme(legend.position="top") +
   scale_x_continuous(breaks=unique(dados$tempo))
p1
```



Uma linha de regressão linear ou suavizada pode ser adicionada ao gráfico fazendo

```
library(gridExtra)
p11 = p1 + geom_smooth(method="lm",se=FALSE,size=2)
p12 = p1 + geom_smooth(method="loess",se=FALSE,size=2)
grid.arrange(p11,p12,ncol=2)
```



Como vemos, o comportamento longitudinal é aproximadamente linear e um modelo com interação sexo e tempo parece ser adequado. O modelo a ser ajustado é dado por

$$E(Y_{ij}) = \beta_0 + \beta_1 \times sexo_i + \beta_2 \times tempo_j + \beta_3 \times tempo_j \times sexo_i.$$

Estimador GLS

Consideraremos novamente as estruturas de correlação do tipo *independente*, *simetria composta*, *AR(1)* e *não estruturada*. Para fins de análise as idades foram centradas em um valor comum, no caso a média de 11 anos.

```
dados$tempo=dados$tempo-11
gls2.ind<-gls(resp ~ sex*tempo, data=dados) #Independente
gls2.exch<-gls(resp ~ sex*tempo, correlation=corCompSymm(form=~1|id), data=dados) #Simetria composta
gls2.ar1<-gls(resp ~ sex*tempo, correlation=corAR1(form=~1|id), data=dados) #AR(1)
gls2.unst<-gls(resp ~ sex*tempo, correlation=corSymm(form=~1|id), data=dados) #Não estruturada
```

Os resultados dos ajustes são mostrados a seguir:

```
# Independente
round(coef(summary(gls2.ind)),3)
```

```
##              Value Std.Error t-value p-value
## (Intercept) 22.648     0.340  66.562   0.000
## sexM         2.321     0.442   5.251   0.000
## tempo        0.480     0.152   3.152   0.002
## sexM:tempo   0.305     0.198   1.542   0.126
```

```
# Simetria composta
round(coef(summary(gls2.exch)),3)
```

```
##              Value Std.Error t-value p-value
## (Intercept) 22.648     0.586  38.639   0.000
## sexM         2.321     0.761   3.048   0.003
## tempo        0.480     0.093   5.130   0.000
## sexM:tempo   0.305     0.121   2.511   0.014
```

```
# AR(1)
round(coef(summary(gls2.ar1)),3)
```

```
##              Value Std.Error t-value p-value
## (Intercept) 22.643     0.529  42.797   0.000
## sexM         2.418     0.687   3.519   0.001
## tempo        0.484     0.141   3.430   0.001
## sexM:tempo   0.285     0.183   1.558   0.122
```

```
# Não estruturada
round(coef(summary(gls2.unst)),3)
```

```
##              Value Std.Error t-value p-value
## (Intercept) 22.645     0.585  38.697   0.000
## sexM         2.355     0.760   3.098   0.003
## tempo        0.476     0.099   4.791   0.000
## sexM:tempo   0.348     0.129   2.696   0.008
```

Note como as estimativas das estruturas *independente* e *simetria composta* são similares. Isso ocorre por conta do balanceamento (no tempo). Interessante notar como o valor p é bastante pequeno para *simetria composta* e *não estruturada* e alto para as demais estruturas. Das correlações marginais vimos que as estruturas independente e autorregressiva não são adequadas a esses dados.

Estimador GEE

Ajustamos agora as mesmas estruturas de correlação e estimamos os modelos pelo método GEE.

```
library(geepack)
gee2.ind<-geeglm(resp ~ sex*tempo, id=id, corstr="independence", data=dados) #Independente
gee2.exch<-geeglm(resp ~ sex*tempo, id=id, corstr="exchangeable", data=dados) #Simetria composta
gee2.ar1<-geeglm(resp ~ sex*tempo, id=id, corstr="ar1", data=dados) #AR(1)
gee2.unst<-geeglm(resp ~ sex*tempo, id=id, corstr="unstructured", data=dados) #Não estruturada
```

As estimativas são dados por:

```
# Independente
round(coef(summary(gee2.ind)),3)
```

```
##             Estimate Std.err     Wald Pr(>|W|)
## (Intercept)   22.648   0.605 1400.761    0.000
## sexM           2.321   0.750    9.583    0.002
## tempo          0.480   0.063   57.697    0.000
## sexM:tempo     0.305   0.117    6.803    0.009
```

```
# Simetria composta
round(coef(summary(gee2.exch)),3)
```

```
##             Estimate Std.err     Wald Pr(>|W|)
## (Intercept)   22.648   0.605 1400.761    0.000
## sexM           2.321   0.750    9.583    0.002
## tempo          0.480   0.063   57.697    0.000
## sexM:tempo     0.305   0.117    6.803    0.009
```

```
# AR(1)
round(coef(summary(gee2.ar1)),3)
```

```
##             Estimate Std.err     Wald Pr(>|W|)
## (Intercept)   22.641   0.618 1341.792    0.000
## sexM           2.452   0.758   10.458    0.001
## tempo          0.484   0.063   58.979    0.000
## sexM:tempo     0.283   0.124    5.216    0.022
```

```
# Não estruturada
round(coef(summary(gee2.unst)),3)
```

```
##             Estimate Std.err     Wald Pr(>|W|)
## (Intercept)   22.656   0.599 1431.397    0.000
## sexM           2.337   0.736   10.077    0.002
## tempo          0.478   0.064   56.023    0.000
## sexM:tempo     0.310   0.117    6.997    0.008
```

As estimativas de erro padrão dos coeficientes são similares entre as diferentes estruturas, o que mostra a robustez do método GEE à má especificação da estrutura de dependência entre as medidas repetidas. Agora o efeito de interação é significativo em todas as análises. Podemos concluir que meninos e meninas crescem em ritmos distintos.

Comentários sobre a coincidência entre as estimativas de independência e simetria composta

Como vimos, as análises independente e simetria composta dão as mesmas estimativas e erro padrão robusto (mas não naive) porque os dados são balanceados. Vamos criar alguns "dados ausentes" e ver o que acontece. Deletamos as últimas duas observações dos primeiros cinco indivíduos para criar desbalanceamento.

```
dados2=dados[-c(3,4,7,8,11,12,15,16,19,20),]
head(dados2)
```

```
##    id sex tempo resp
## 1   1   F    -3 21.0
## 2   1   F    -1 20.0
## 5   2   F    -3 21.0
## 6   2   F    -1 21.5
## 9   3   F    -3 20.5
## 10  3   F    -1 24.0
```

```
gee3.ind<-geeglm(resp ~ sex*tempo, id=id, corstr="independence", data=dados2) #Independente
gee3.exch<-geeglm(resp ~ sex*tempo, id=id, corstr="exchangeable", data=dados2) #Simetria composta
round(coef(summary(gee3.ind)),3)
```

```
##             Estimate Std.err    Wald Pr(>|W|)
## (Intercept)   22.408   0.779 827.602    0.000
## sexM           2.561   0.896   8.169    0.004
## tempo          0.369   0.127   8.469    0.004
## sexM:tempo     0.416   0.160   6.723    0.010
```

```
round(coef(summary(gee3.exch)),3)
```

```
##             Estimate Std.err     Wald Pr(>|W|)
## (Intercept)   22.518   0.656 1179.456    0.000
## sexM           2.451   0.791    9.597    0.002
## tempo          0.415   0.073   32.167    0.000
## sexM:tempo     0.370   0.123    9.102    0.003
```

Por conta do desbalanceamento os resultados são diferentes para as duas estruturas.