Lista 5

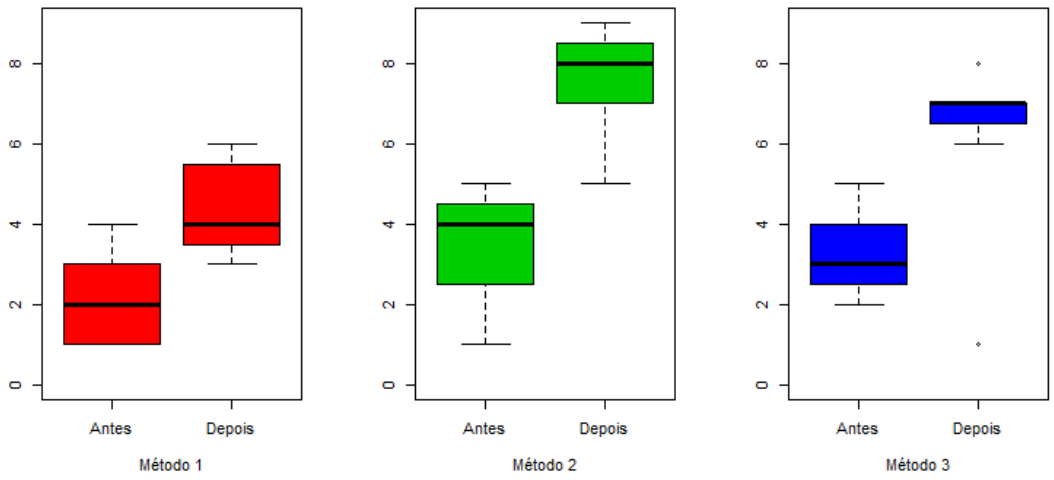
Victor

21 de outubro de 2016

Questão 1

a)

```
df1 <- data.frame(metodo = rep(c("1", "2", "3"), each = 7),
                  antes = c(3, 1, 3, 1, 2, 1, 4,
                            4, 5, 5, 4, 3, 1, 2,
                            3, 2, 2, 3, 4, 5, 4),
                  depois = c(6, 4, 5, 3, 4, 3, 6,
                             8, 9, 7, 9, 8, 5, 7,
                             6, 7, 7, 7, 8, 1, 7))

df1$difer <- df1$depois-df1$antes

par(mfrow = c(1, 3))
boxplot(df1$antes[df1$metodo == "1"],df1$depois[df1$metodo == "1"], ylim =
c(0, 9), col = 2,
        xlab = "Método 1", names = c("Antes", "Depois"))
boxplot(df1$antes[df1$metodo == "2"], df1$depois[df1$metodo == "2"], ylim =
c(0, 9), col = 3,
        xlab = "Método 2", names = c("Antes", "Depois"))
boxplot(df1$antes[df1$metodo == "3"], df1$depois[df1$metodo == "3"], ylim =
c(0, 9), col = 4,
        xlab = "Método 3", names = c("Antes", "Depois"))
```

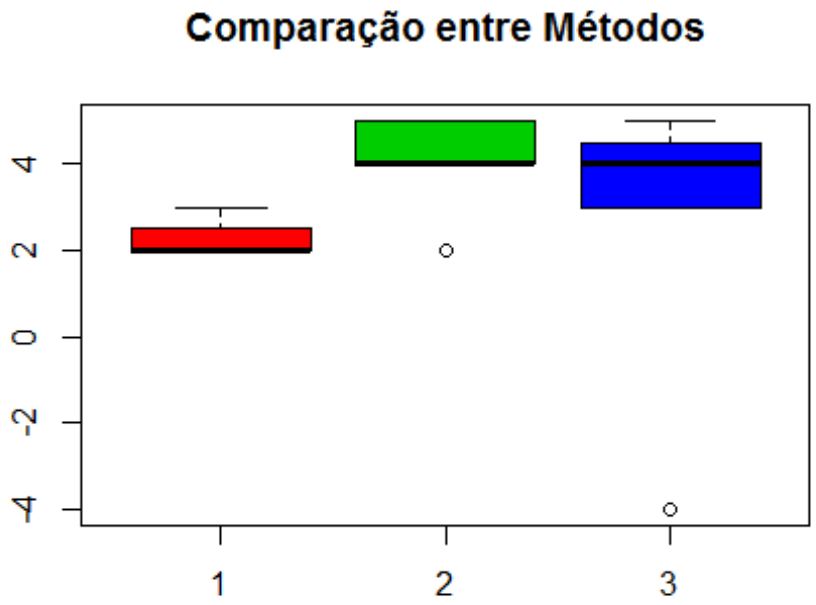


```
par(mfrow = c(1, 1))
```

Os boxplots acima apresentam os escores dos indivíduos do banco antes e depois do treinamento. Eles apontam para uma melhoria geral nos escores depois da realização do treinamento. É possível observar que os escores mais altos após o treinamento foram obtidos pelos sujeitos submetidos ao método 2 de treinamento.

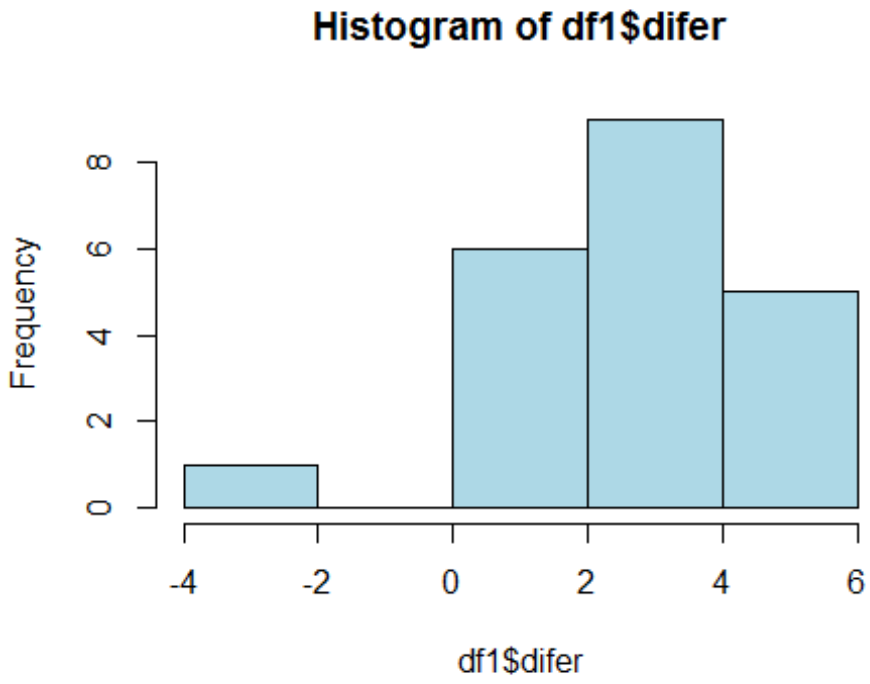
Criamos também um escore de diferenças, com o objetivo de quantificar a mudança no escore alcançado em função dos diferentes métodos de treinamento.

```
boxplot(df1$difer~df1$metodo, col = c(2, 3, 4), main = "Comparação entre
Métodos")
```

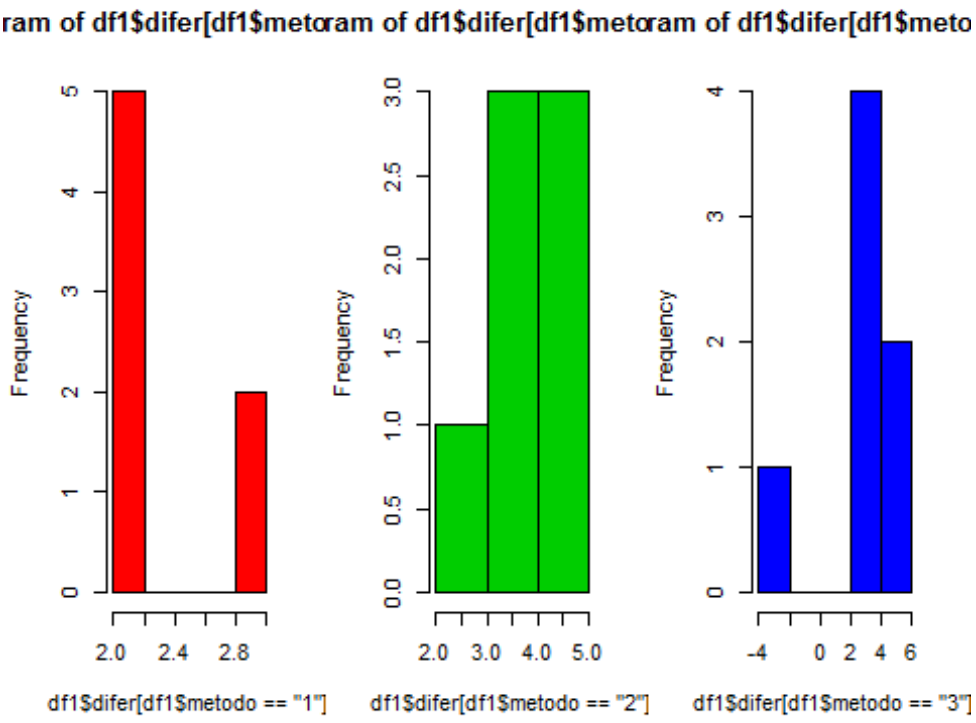


Os boxplots apontam para que a maior parte dos escores aumentou em função dos treinamentos. Contudo, é importante salientar que existem algumas observações que destoam fortemente da distribuição dos outros escores de diferença. Esses pontos podem levar a problemas quando do encaixe de algum modelo aos dados.

```
hist(df1$difer, col = "light blue")
```




```r
par(mfrow = c(1, 3))
hist(df1$difer[df1$metodo == "1"], col = 2)

hist(df1$difer[df1$metodo == "2"], col = 3)

hist(df1$difer[df1$metodo == "3"], col = 4)
```



Construímos também o histograma dos escores para observar qual a distribuição dos escores de diferença totais bem como separado entre grupos. É difícil tirar muitas conclusões em função do baixo número de observações no banco de dados.

b

Como os valores de diferença são escores numa prova, parece razoável encaixar um modelo com y com distribuição de Normal. Nesse caso:

$$\hat{y}_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}$$

Em que:

y_i: é a mudança de escore observada no i-ésimo indivíduo

$x_{1i} = 1$ se o método de treinamento do i-ésimo indivíduo for o método 2 e 0 caso contrário.

$x_{2i} = 1$ se o método de treinamento do i-ésimo indivíduo for o método 3 e 0 caso contrário.

Onde, $i = 1, \ldots, 21$

c

Os parâmetros devem ser interpretados de acordo com o pertencimento a determinados método de treinamento. Ou seja, β_0 será o valor estimado de $\hat{y}$ quando $x_{1i} = x_{2i} = 0$, ou seja, quando o indivíduo for treinado pelo método 1. Com isso, β_1 é a mudança média observada na diferença entre escores quando os sujeitos do método 1 e do método 2 são

comparados, e β_2 possui a mesma interpretação só que referente à comparação entre sujeitos dos grupos 1 e 3.

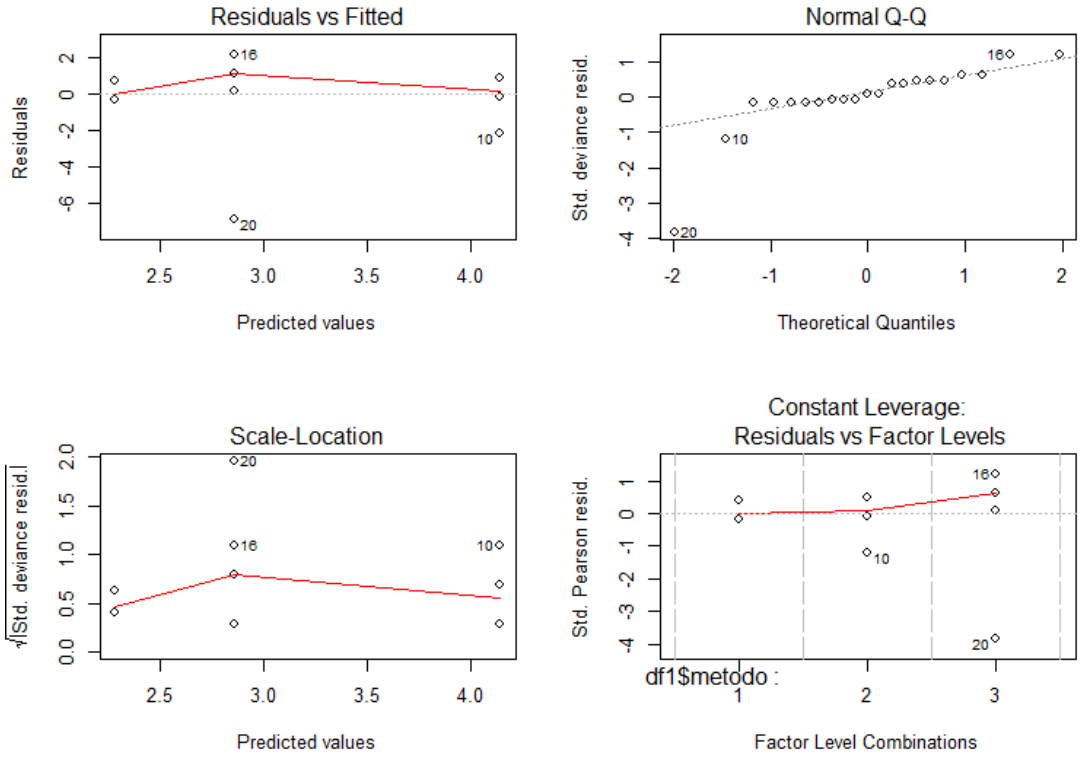
d

```
#Fit1 - Modelo Normal
fit1 <- glm(df1$difer~df1$metodo)
summary(fit1)
```

```
##
## Call:
## glm(formula = df1$difer ~ df1$metodo)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.8571  -0.2857   0.1429   0.8571   2.1429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2857     0.7300   3.131  0.00577 **
## df1$metodo2   1.8571     1.0324   1.799  0.08882 .
## df1$metodo3   0.5714     1.0324   0.554  0.58672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.730159)
##
##     Null deviance: 79.810  on 20  degrees of freedom
## Residual deviance: 67.143  on 18  degrees of freedom
## AIC: 92.004
##
## Number of Fisher Scoring iterations: 2
```

Procedemos à análise dos resíduos para identificar problemas do encaixe

```
par(mfrow = c(2, 2))
plot(fit1)
```




```
par(mfrow = c(1, 1))
```

Parece que a observação de número 20 é a que mais distoa das outras. Encaixamos o modelo novamente sem essa observação.

```
#Fit2 - Modelo Normal sem Observação influente
fit2 <- glm(df1$difer~df1$metodo, data = df1[-20,])
summary(fit2)

##
## Call:
## glm(formula = df1$difer ~ df1$metodo, data = df1[-20, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.8571  -0.2857   0.1429   0.8571   2.1429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2857     0.7300   3.131  0.00577 **
## df1$metodo2   1.8571     1.0324   1.799  0.08882 .
## df1$metodo3   0.5714     1.0324   0.554  0.58672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 3.730159)
##
##     Null deviance: 79.810  on 20  degrees of freedom
## Residual deviance: 67.143  on 18  degrees of freedom
## AIC: 92.004
##
## Number of Fisher Scoring iterations: 2
```

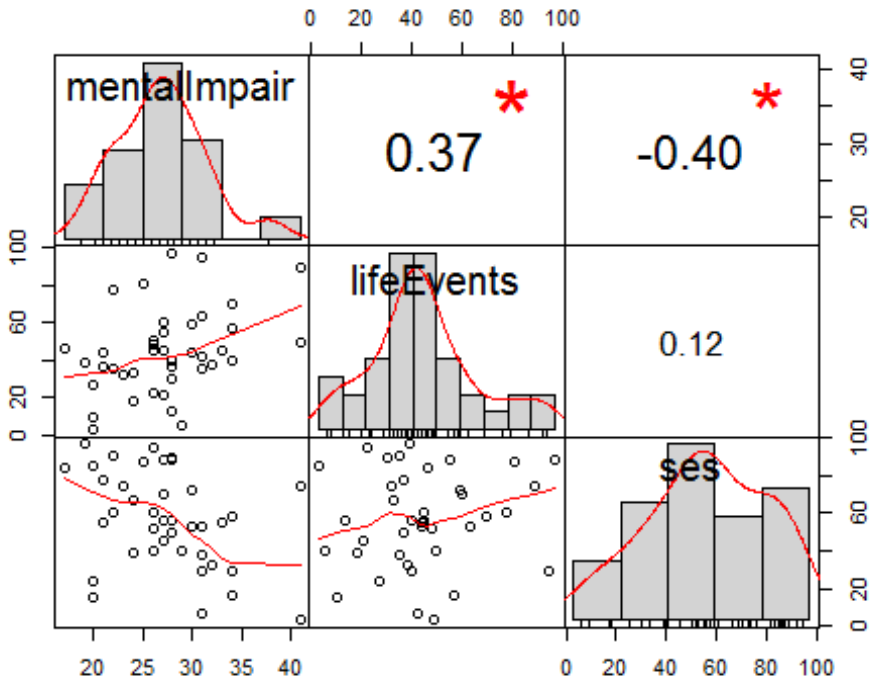
Apenas o parâmetro β_1 aproximou-se de ser significativo, indicando um aumento médio de 1,86 na diferença entre os escores iniciais e finais dos grupos que foram submetidos ao método 2 de treinamento quando comparado ao grupo que foi submetido ao método 1 de treinamento.

Q2

```
df2 <- read.dta("afMentalHealth.dta")
```

a)

```
chart.Correlation(df2)
```



Os gráficos apontam para a presença de relações lineares entre a variável dependente e as variáveis preditoras. A relação entre a VD e o escore relacionado aos eventos da vida é positiva, com uma de correlação de pearson de 0.3722. Contrariamente, a relação entre a VD e a nível socioeconômico é negativa, com correlação de Pearson de -0.4.

b)

```
#Ajustando um modelo linear simples com função de ligação identidade
fit2.1 <- lm(df2$mentalImpair~df2$lifeEvents)
summary(fit2.1)

##
## Call:
## lm(formula = df2$mentalImpair ~ df2$lifeEvents)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4415  -3.6899  -0.5973   3.6215  13.2890
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    23.30949    1.80675  12.901 1.85e-15 ***
## df2$lifeEvents  0.08983    0.03633   2.472    0.018 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.133 on 38 degrees of freedom
## Multiple R-squared:  0.1385, Adjusted R-squared:  0.1159
## F-statistic: 6.112 on 1 and 38 DF,  p-value: 0.01802
```

O coeficiente de 0.0893 indica o aumento médio na variável dependente associado ao acréscimo de um ponto na variável dependente.

A proporção de variância explicada é dada pelo $R^2 = 0.1385$ que, no caso de uma regressão normal de uma variável é o quadrado da correlação de pearson, ou seja:

$$R^2 = 0.1385 = 0.3722^2$$

c)

Existem algumas estratégias para avaliar a linearidade da relação entre duas variáveis, apresentaremos três delas

A primeira é o gráfico de dispersão apresentado no item anterior, que aponta para a linearidade da relação entre VI e VD.

Uma segunda forma de avaliar essa linearidade é com a inclusão de um termo quadrático na regressão e a avaliação de sua significância dentro do modelo, esperamos que o termo quadrático seja considerado relevante apenas quando da existência de uma relação não-linear entre as variáveis.

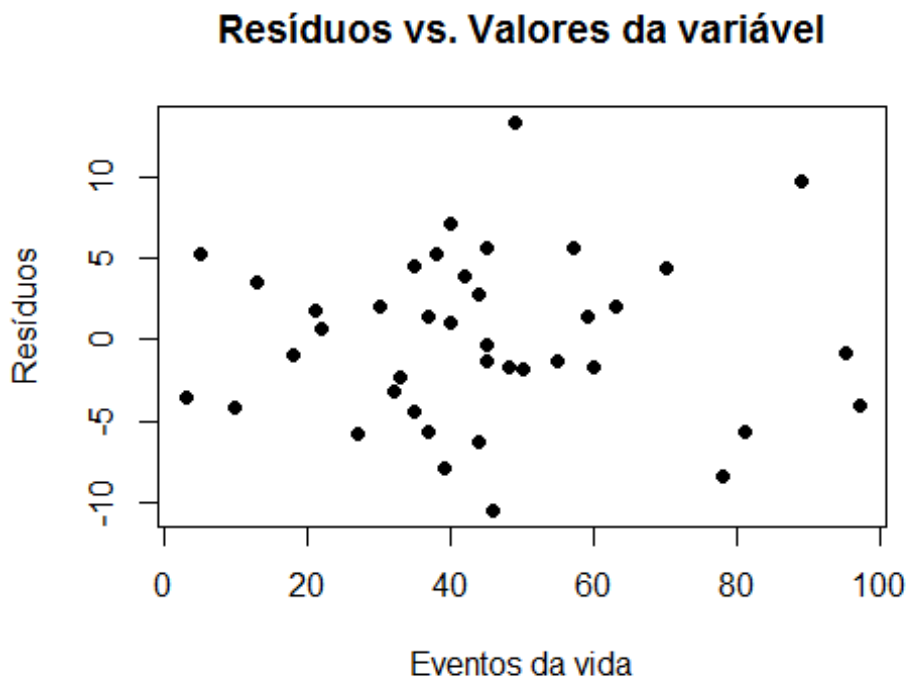
```
fit2.2 <- lm(df2$mentalImpair~df2$lifeEvents + I(df2$lifeEvents^2))

summary(fit2.2)
```

```
##
## Call:
## lm(formula = df2$mentalImpair ~ df2$lifeEvents + I(df2$lifeEvents^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6490  -3.2865  -0.3971   3.7625  13.0755
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         22.5146193  3.0023355   7.499  6.2e-09 ***
## df2$lifeEvents       0.1301916  0.1264041   1.030     0.31
## I(df2$lifeEvents^2) -0.0004038  0.0012098  -0.334     0.74
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.194 on 37 degrees of freedom
## Multiple R-squared:  0.1411, Adjusted R-squared:  0.09471
## F-statistic:  3.04 on 2 and 37 DF,  p-value: 0.05993
```

O que não se verifica!

Uma terceira forma de avaliar a não-linearidade da relação consiste em avaliar os resíduos do modelo encaixado plotados contra a variável independente.

```
par(mfrow = c(1, 1))
plot(df2$lifeEvents, fit2.1$residuals, main = "Resíduos vs. Valores da
variável", ylab = "Resíduos",
     xlab = "Eventos da vida", pch = 19)
```



Na qual espera-se que os pontos distribuam-se de maneira amorfa e não seja possível observar alguma tendência nos dados. Que é o caso.

d)

```
fit2.3 <- lm(df2$mentalImpair~df2$lifeEvents + df2$ses)
summary(fit2.3)

##
## Call:
## lm(formula = df2$mentalImpair ~ df2$lifeEvents + df2$ses)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.678 -2.494 -0.336  2.886 10.891
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    28.22981    2.17422  12.984 2.38e-15 ***
## df2$lifeEvents  0.10326    0.03250   3.177  0.00300 **
## df2$ses        -0.09748    0.02908  -3.351  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.556 on 37 degrees of freedom
## Multiple R-squared:  0.3392, Adjusted R-squared:  0.3034
## F-statistic: 9.495 on 2 and 37 DF,  p-value: 0.0004697
```

A partir do momento em que estamos rodando uma regressão múltipla, passamos a interpretar os coeficientes tendo em vista que as outras variáveis estão fixas. No caso específico da variável "Eventos da Vida", observamos um aumento no valor do coeficiente e no grau de significância após a inclusão da variável SES. Nesse caso, temos que o aumento de 1 ponto na variável "Eventos da Vida" está associado a um aumento médio de .10 pontos na variável dependente, quando a variável SES é mantida fixa, ou seja, controlando pela variável SES.

e)

Os valores ajustados estão disponíveis no objeto com o ajuste do modelo, dentro da alça "$fitted.values". A correlação entre eles e os valores observados é calculada a seguir

```
#Correlação entre valores preditos e observados
cor(fit2.3$fitted.values, df2$mentalImpair)

## [1] 0.5823736
```

Notamos que esse valor é a raiz quadrada do coeficiente de determinação do modelo.

$$0.582 = \sqrt{0.339}$$

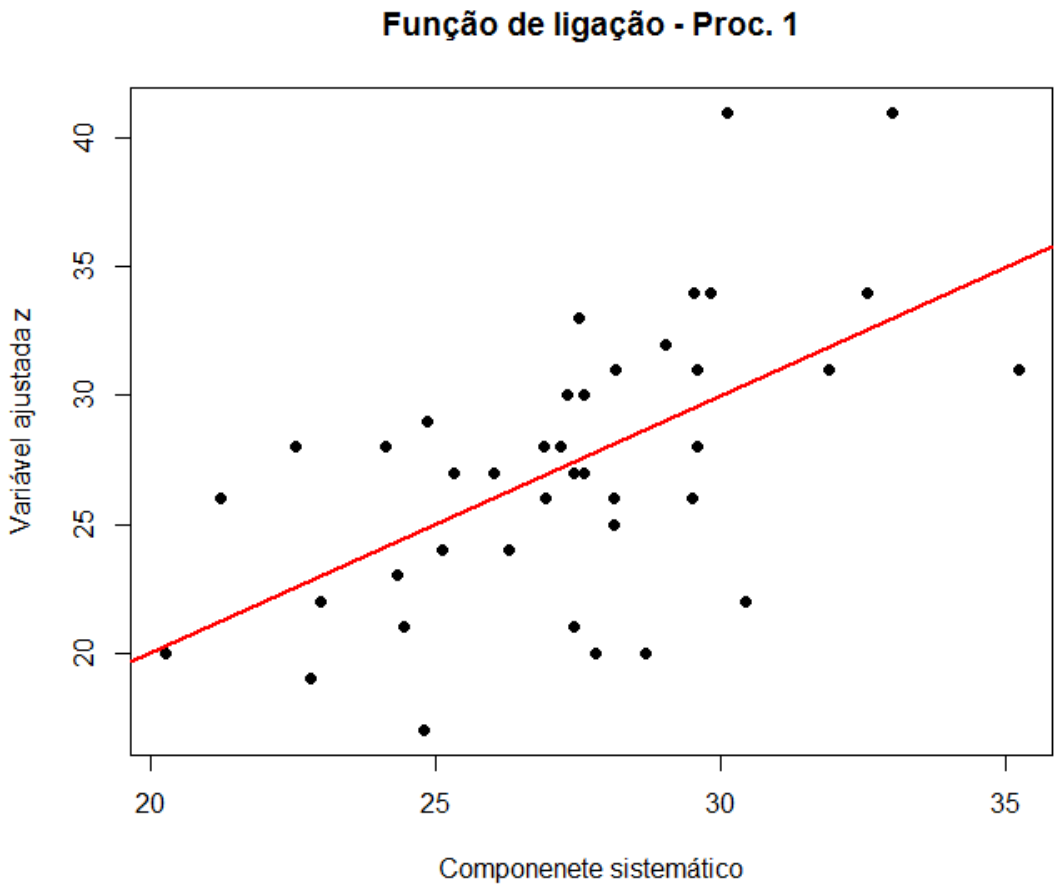
f)

Para averiguar a adequação da função de ligação é importante lembrar que a função de ligação deve produzir uma relação linear entre a porção sistemática do modelo ($\boldsymbol{\eta}$) e a variável ajustada ($\mathbf{z}$). Assim, testaremos essa relação.

Sabemos que no modelo linear a porção determinística é dada por $\boldsymbol{\eta} = \mathbf{X}^T\boldsymbol{\beta}$ e a variável ajustada é o próprio $\mathbf{y}$.

Dito isso, temos nosso primeiro procedimento (informal) de avaliação da qualidade da função de ligação, exemplificado no gráfico a seguir.

```
plot(fit2.3$fitted.values, df2$mentalImpair, pch = 19, main = "Função de
ligação - Proc. 1",
     xlab = "Componenete sistemático", ylab = "Variável ajustada z")
abline(lm(df2$mentalImpair~fit2.3$fitted.values), col = "red", lwd = 2)
```



Observamos uma relação linear entre os dois vetores, apontando para a adequação da função de ligação.

O segundo procedimento consistirá em incluir o vetor η^2 como covariável do modelo e utilizar um teste de razão de verossimilhanças para avaliar a pertiência dessa nova variável ao modelo.

```
fit2.4 <- lm(df2$mentalImpair~df2$lifeEvents + df2$ses +
I(fit2.3$fitted.values^2))
summary(fit2.4)

##
## Call:
## lm(formula = df2$mentalImpair ~ df2$lifeEvents + df2$ses +
I(fit2.3$fitted.values^2))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.480 -2.271 -0.254  3.104 10.968
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                7.55490   41.39953   0.182    0.856
## df2$lifeEvents            -0.04218    0.29266  -0.144    0.886
## df2$ses                    0.03936    0.27519   0.143    0.887
## I(fit2.3$fitted.values^2)  0.02568    0.05135   0.500    0.620
##
## Residual standard error: 4.603 on 36 degrees of freedom
## Multiple R-squared:  0.3437, Adjusted R-squared:  0.289
## F-statistic: 6.285 on 3 and 36 DF,  p-value: 0.001531
```

```
anova(fit2.3, fit2.4)
```

```
## Analysis of Variance Table
##
## Model 1: df2$mentalImpair ~ df2$lifeEvents + df2$ses
## Model 2: df2$mentalImpair ~ df2$lifeEvents + df2$ses +
I(fit2.3$fitted.values^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     37 768.16
## 2     36 762.86  1    5.2999 0.2501   0.62
```

O resultado aponta que o acréscimo da variável no modelo não apresenta um ganho significativo no deviance. Isso é mais um indício de que a função de ligação é adequada.

Q3

a)

A figura 1 apresenta os diagramas de dispersão entre a carga na esteira ergométrica e o consumo de oxigênio por cada sub-grupo de pacientes. Os diagramas apontam uma relação linear e heterocedástica entre as variáveis para todos os grupos, o que é indicativo de que o modelo linear é adequado.

b)

A tabela 1 refuta a hipótese nula de que as médias da variável dependente são iguais para todos os subgrupos de pacientes ($p < .001$). Isso indica a existência de alguma relação entre o tipo paciente e o consumo médio de oxigênio.

c)

Os dados da tabela 2 apontam que levando em conta os dados é muito pouco provável ($p < .001$) que valores pelo menos tão extremos de β fossem encontrados sob a hipótese nula de que eles são iguais a zero. Os testes são feitos para cada um dos β_i separadamente. Entretanto, não é possível concluir que eles sejam diferentes entre si, seria necessária uma análise mais cautelosa da variância de cada um deles para ter essa certeza. Contudo, ao observar os intervalos de confiança é possível perceber que o IC do β_{13} não contem o valor de β_{12}, o que é um indicativo de que eles sejam estatisticamente diferentes entre si.

d)

Os próximos passos seriam no sentido de construir uma matriz de contrastes tal que as diferenças entre os β_{ij} pudessem ser testadas. Um possível formato para a matriz C seria o seguinte:

$$\mathbf{C}_{6,5} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

Isso ia permitir testar a hipótese de igualdade entre cada um dos coeficientes da regressão.

Q4

O código será comentado no sentido de definir o que acontece em cada passo.

```
fit.model <- fit2.4
#Obtenção da matriz de design do modelo, que é a matriz X com a primeira
coluna de 1s para incorporar o intercepto
X <- model.matrix(fit.model)
#Obtendo o número de observações da matriz
n <- nrow(X)
#Obtendo o número de colunas da matriz
p <- ncol(X)
#Obtenção do vetor de valores finals de pesos (w)
w <- fit.model$weights
#Transformando o vetor numa matriz diagonal
W <- diag(w)
#Invertendo a matriz reponderada pela matriz W
H <- solve(t(X)%*%W%*%X)
#Obtendo a matriz H que estima a variância e covariância das estimativas do
modelo
H <- sqrt(W)%*%X%*%H%*%t(X)%*%sqrt(W)
#Obtendo a diagonal da matriz acima, referente à variância de cada estimativa
de beta
h <- diag(H)
#Obtendo os resíduos de pearson padronizados
ts <- resid(fit.model,type="pearson")/sqrt(1-h)
#Obtendo os resíduos de deviance padronizados
td <- resid(fit.model,type="deviance")/sqrt(1-h)
#Calculando as distâncias de cook para avaliar pontos de alta leverage
di <- (h/(1-h))*(ts^2)
#Obtendo os valores máximos dos desvios estudentizados (utilizado para
definir os limites do gráfico)
a <- max(td)
#Obtendo os valores máximos dos desvios estudentizados (utilizado para
```

```r
definir os limites do gráfico)
b <- min(td)

#Plotando os gráficos
par(mfrow=c(2,2))
plot(fitted(fit.model),h,xlab="Valores Ajustados", ylab="Medida h", pch=16)
plot(di,xlab="Indice", ylab="Distancia de Cook",pch=16)
plot(td,xlab="Indice", ylab="Residuo Componente do Desvio", ylim=c(b-1,a+1),
pch=16)
abline(2,0,lty=2)
abline(-2,0,lty=2)

#Obtendo os valores do preditor linear eta
eta = predict(fit.model)

#Obtendo a variável modificada z com base nos resíduos, no vetor de w e no
eta.
z = eta + resid(fit.model, type="pearson")/sqrt(w)

#Plotando
plot(predict(fit.model),z,xlab="Preditor Linear", ylab="Variavel z", pch=16)

#Incorporando as linhas do envelope
lines(smooth.spline(predict(fit.model), z, df=2))
```