Trabalho 3

Daniel Krügel

2023-01-19

Exercício 7

```
#Carregando dados
data <- read.csv(file = "http://leg.ufpr.br/~lucambio/ADC/healthcare_worker.csv")

#Criando tabela de contingência
c.table <- array(data = c(data[,2],(data[,3]-data[,2])),
                 dim = c(5,2),
                 dimnames = list(occupation = data[,1], Hepatite = c("+","-")))

c.table
```

```
##                     Hepatite
## occupation            +    -
##   Exposure prone      5 2200
##   Fluid contact      17 6190
##   Lab staff           3  530
##   Patient contact     2 1236
##   No patient contact  3  468
```

Incluí este chunk apenas para demonstrar como foi feita a tabela de contingência

```
#Teste qui quadrado de pearson
chisq.test(c.table)
```

```
## Warning in chisq.test(c.table): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  c.table
## X-squared = 4.5043, df = 4, p-value = 0.342
```

```
#Teste LR
vcd::assocstats(c.table)
```

```
##                     X^2 df P(> X^2)
## Likelihood Ratio 3.7350  4  0.44305
```

```
## Pearson          4.5043  4  0.34204
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.021
## Cramer's V        : 0.021
```

Para este teste esperavamos ver um p-valor alto pois assim não teriamos dependencia entre o nível de exposição e a quantidade de casos de hepatite C, como foi demonstrado no exercício 19 do trabalho 2. No teste qui quadrado de pearson encontramos um p-valor de 0.342 muito elevado para o padrão ouro de p-valor de 0.05, portanto podemos assumir que existe independencia entre o nível de exposição ao paciente e a presença de Hepatite C.

Exercício 11

Questão a

```
cereal = read.csv( file = "http://leg.ufpr.br/~lucambio/ADC/cereal_dillons.csv")
head(cereal)
```

```
##   ID Shelf                              Cereal size_g sugar_g fat_g sodium_mg
## 1  1     1 Kellog's Razzle Dazzle Rice Crispies     28      10     0       170
## 2  2     1            Post Toasties Corn Flakes     28       2     0       270
## 3  3     1                Kellogg's Corn Flakes     28       2     0       300
## 4  4     1               Food Club Toasted Oats     32       2     2       280
## 5  5     1                     Frosted Cheerios     30      13     1       210
## 6  6     1             Food Club Frosted Flakes     31      11     0       180
```

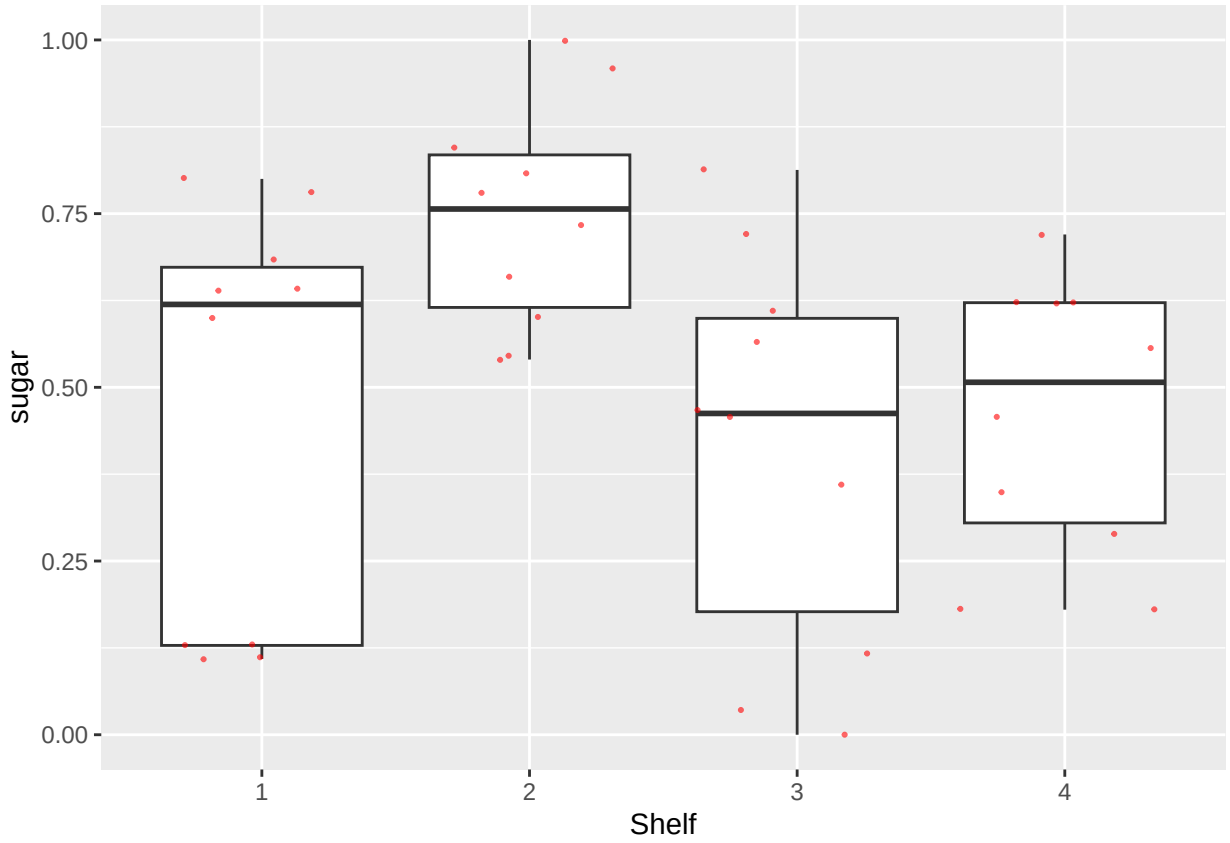
```
stand01 <- function (x) { (x - min(x))/( max(x) - min(x)) }
cereal2 <- data.frame (Shelf = cereal$Shelf, sugar = stand01 (x = cereal$sugar_g / cereal$size_g ),
                       fat = stand01 (x = cereal$fat_g / cereal$size_g ),
                       sodium = stand01 (x = cereal$sodium_mg / cereal$size_g ))

VGAM::vglm(Shelf ~ sugar + fat + sodium,
           family = 'multinomial',
           data = cereal2)
```

```
##
## Call:
## VGAM::vglm(formula = Shelf ~ sugar + fat + sodium, family = "multinomial",
##     data = cereal2)
##
##
## Coefficients:
## (Intercept):1 (Intercept):2 (Intercept):3       sugar:1       sugar:2
##   -21.3002093   -14.3886214     0.3925159    11.4011669    14.0866802
##       sugar:3         fat:1         fat:2         fat:3     sodium:1
##    -0.8225701     0.8703232     4.9348954     0.3127917    24.6861059
##      sodium:2      sodium:3
##     7.1830761    -0.3051198
##
```

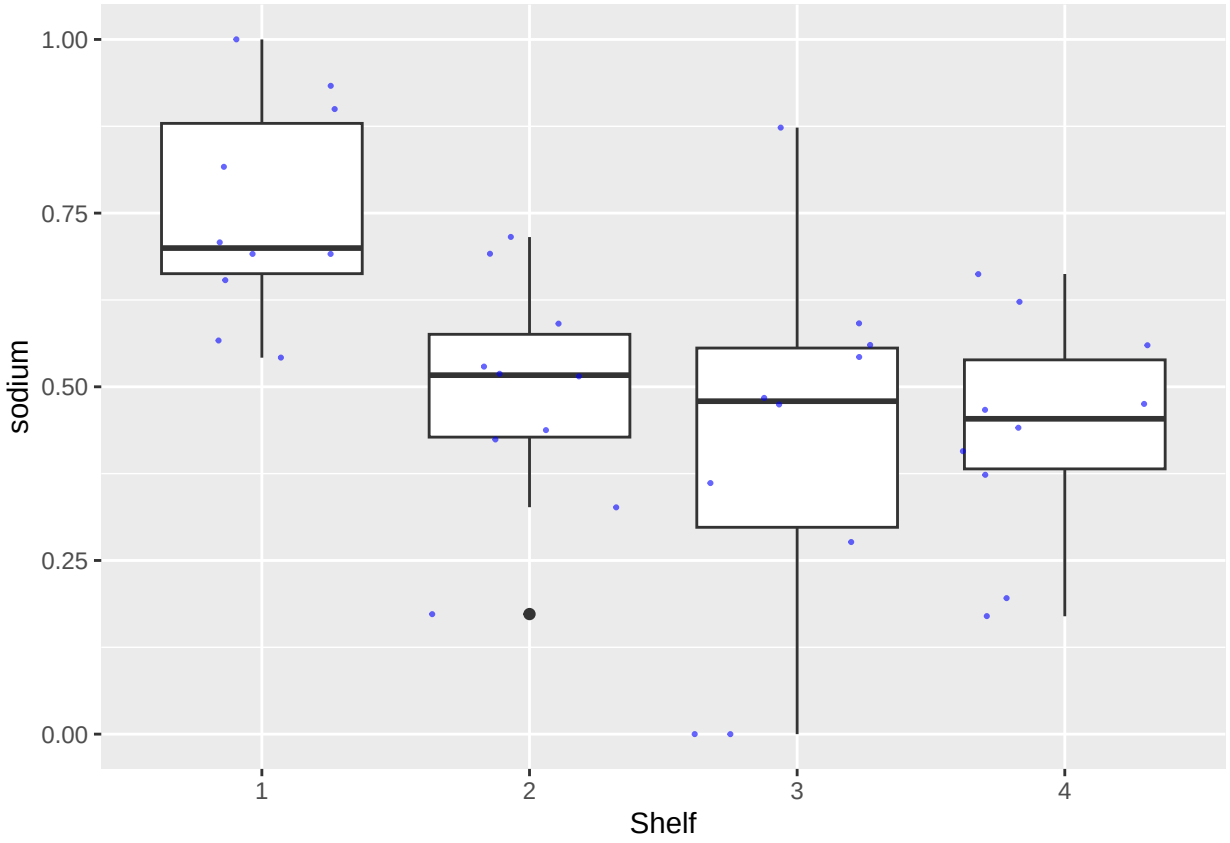
```
## Degrees of Freedom: 120 Total; 108 Residual
## Residual deviance: 67.19027
## Log-likelihood: -33.59514
##
## This is a multinomial logit model with 4 levels
```

Questão b

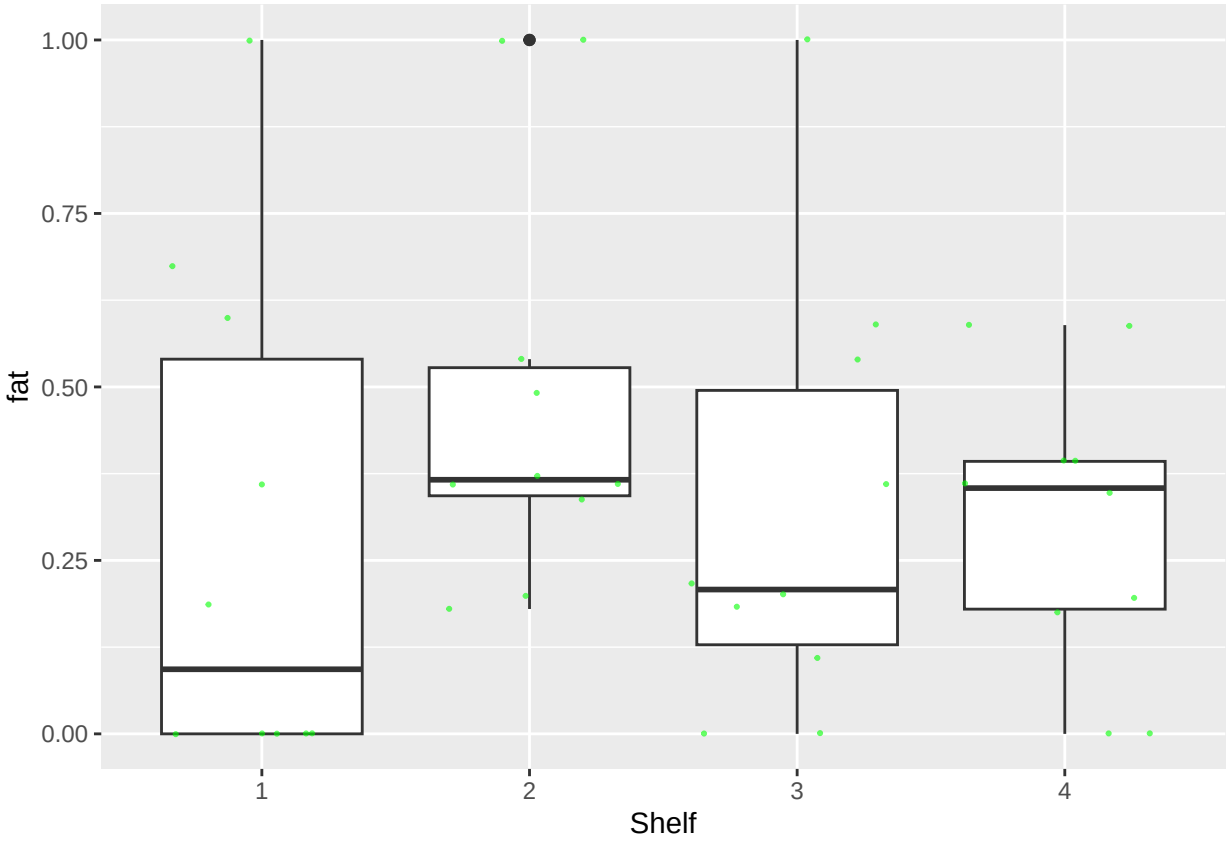
```
cerealb <- cereal2

cerealb$Shelf <- as.factor(cerealb$Shelf)

ggplot(cerealb, aes(x = Shelf, y = sugar)) +
  geom_boxplot() +
  geom_jitter(color = 'red', size = 0.4, alpha = 0.6)
```




```
ggplot(cerealb, aes(x = Shelf, y = sodium)) +
  geom_boxplot() +
  geom_jitter(color = 'blue', size = 0.4, alpha = 0.6)
```




```
ggplot(cerealb, aes(x = Shelf, y = fat)) +
  geom_boxplot() +
  geom_jitter(color = 'green', size = 0.4, alpha = 0.6)
```



Questão c

Levariamos em conta a ordinalidade da variável Shelf no caso da posição de cada uma das prateleiras, assim colocando alimentos com mais açucar, sódio e gordura em prateleiras mais altas, para evitar que crianças as peguem com facilidade, por exemplo.

Questão d

```
fit0 <- nnet::multinom(formula = Shelf ~ sugar + fat + sodium,
               data = cereal2)
```

```
## # weights:  20 (12 variable)
## initial  value 55.451774
## iter  10 value 37.329384
## iter  20 value 33.775257
## iter  30 value 33.608495
## iter  40 value 33.596631
## iter  50 value 33.595909
## iter  60 value 33.595564
## iter  70 value 33.595277
## iter  80 value 33.595147
## final  value 33.595139
## converged
```

```
summary(fit0)
```

```
## Call:
## nnet::multinom(formula = Shelf ~ sugar + fat + sodium, data = cereal2)
##
## Coefficients:
##   (Intercept)      sugar        fat    sodium
## 2    6.900708   2.693071  4.0647092 -17.49373
## 3   21.680680 -12.216442 -0.5571273 -24.97850
## 4   21.288343 -11.393710 -0.8701180 -24.67385
##
## Std. Errors:
##   (Intercept)    sugar      fat   sodium
## 2    6.487408 5.051689 2.307250 7.097098
## 3    7.450885 4.887954 2.414963 8.080261
## 4    7.435125 4.871338 2.405710 8.062295
##
## Residual Deviance: 67.19028
## AIC: 91.19028
```

```
car::Anova(fit0)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##        LR Chisq Df Pr(>Chisq)
## sugar   22.7648  3  4.521e-05 ***
## fat      5.2836  3     0.1522
## sodium  26.6197  3  7.073e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As variáveis Sódio e Açúcar são as variáveis mais influentes para a variável Shelf

Questão e)

```
## # weights:  12 (6 variable)
## initial  value 55.451774
## iter  10 value 52.046597
## final  value 52.024518
## converged
```

```
## # weights:  16 (9 variable)
## initial  value 55.451774
## iter  10 value 49.022189
## iter  20 value 48.069630
## iter  30 value 48.068914
## final  value 48.068908
## converged
```

```
## # weights:  16 (9 variable)
## initial  value 55.451774
## iter  10 value 49.373042
## iter  20 value 48.314828
## iter  30 value 48.287369
## iter  40 value 48.284782
## iter  50 value 48.284475
## final  value 48.284384
## converged

## # weights:  16 (9 variable)
## initial  value 55.451774
## iter  10 value 50.284604
## iter  20 value 50.145968
## final  value 50.145742
## converged
```

Com todos os modelos ajustados vamos olhar para as Anovas e ver se há alguma interação das variáveis sendo relevantes.

```
car::Anova(fitAll)
```

```
## # weights:  8 (3 variable)
## initial  value 55.451774
## final  value 55.451774
## converged

## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##                  LR Chisq Df Pr(>Chisq)
## sugar:fat:sodium   6.8545  3    0.07668 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(fitSugar)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##              LR Chisq Df Pr(>Chisq)
## sugar:fat      4.9091  3    0.17858
## sugar:sodium   7.3766  3    0.06082 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(fitFat)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
```

```
##            LR Chisq Df Pr(>Chisq)
## fat:sugar   10.1700  3    0.01717 *
## fat:sodium   6.9456  3    0.07365 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(fitSodium)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##              LR Chisq Df Pr(>Chisq)
## sodium:sugar   6.4473  3    0.09176 .
## sodium:fat     0.7554  3    0.86010
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Com as anovas em mãos vemos que a única interação que mostrou alguma significancia foi a de gordura em relação a açúcar, porém demonstrou em um nível de significânica de 90%.

Questão f)

```
kelloggs <- array(c(0.28,0.12,0.5),
                  dim = c(1,3))
kelloggs <- as.data.frame(kelloggs)

colnames(kelloggs) <- c("sugar","fat","sodium")

pihat <- predict(object = fit0,
                 newdata = kelloggs,
                 type = "probs")
```

A uma probabilidade de 0.51 do cereal estar na prateleira 3 e de 0.48 na prateleira 4

Questão h)

```
sd.cereal <- apply (X = cereal2, MARGIN = 2, FUN = sd)
c.value <- c(1,sd.cereal)

beta.hat <- coefficients(fit0)[1,2:4]
round(1/exp(c.value[c(-1,-2)] * beta.hat),2)
```

```
##  sugar    fat sodium
##   0.48   0.30  55.74
```

O fator mais relevante para a explicação de qual prateleira o produto é colocado é a quantidade de sódio presente no cereal, tendo uma razão de chances de 55x

Questão 20

```
## Carregando pacotes exigidos: stats4
```

```
## Carregando pacotes exigidos: splines
```

```
## [1] "mild"   "normal" "severe"
```

```
head(data.20)
```

```
## # A tibble: 6 x 3
##   exposure.time name   value
##           <dbl> <fct>  <dbl>
## 1           5.8 normal    98
## 2           5.8 mild       0
## 3           5.8 severe     0
## 4          15   normal    51
## 5          15   mild       2
## 6          15   severe     1
```

Tentei ajustar primeiramente a variável explicativa exposure.time como fator pois era me pareceu lógico porém ao fazer a predição realizei que tinha me equivocado porém decidi apenas comentar o código feito para deixar registrado.

```
## # weights:  9 (4 variable)
## initial  value 407.585159
## iter  10 value 208.809599
## final  value 208.724782
## converged
```

```
car::Anova(fit.20)
```

```
## # weights:  6 (2 variable)
## initial  value 407.585159
## iter  10 value 252.581034
## iter  10 value 252.581034
## iter  10 value 252.581034
## final  value 252.581034
## converged
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: name
##               LR Chisq Df Pr(>Chisq)
## exposure.time   87.713  2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Aqui vemos que conseguimos convergir um modelo utilizando uma regressão multinomial com a quantidade de casos sendo usados como pesos para balancear a variável explicativa que é o tempo passado nas minas de carvão.

Para realizar o predict precisamos criar um dataframe com os valores sugeridos de se fazer a análise

```r
predição <- data.frame(c(5,10,15,20,25))
colnames(predição) <- c("exposure.time")
predict(fit.20, newdata = predição)
```

```
## [1] normal normal normal normal normal
## Levels: normal mild severe
```

Em todos os casos a predição utilizando o modelo ajustado anteriormente dão como resultado mais provável que seja desenvolvido casos considerados normais.