

# TMA4315: Project 2

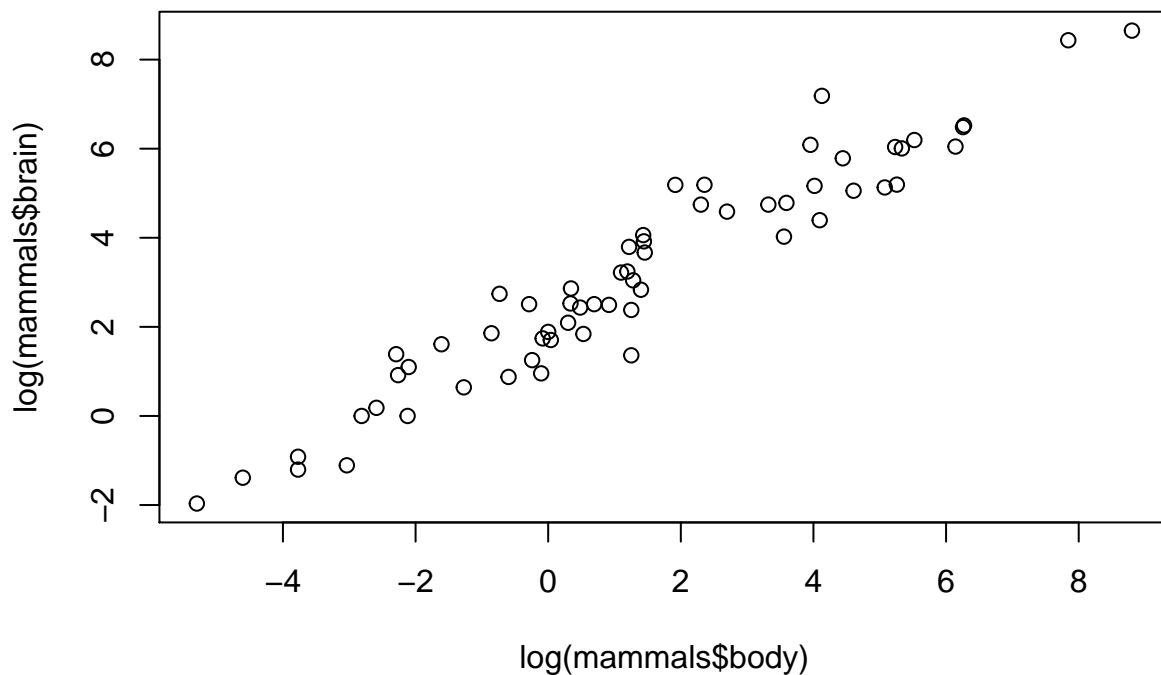
jototlan@stud.ntnu.no (10018), martigtu@stud.ntnu.no (10037)

## Problem 1

```
mammals <- read.table(  
  "https://www.math.ntnu.no/~jarlet/statmod/mammals.dat",  
  header=T)
```

a)

```
plot(log(mammals$body), log(mammals$brain))
```



The log-log plot of the brain mass against body mass seems to reveal a linear trend. We thus fit the following model:

```
mod0 <- lm(log(brain) ~ log(body), data = mammals)  
summary(mod0)
```

```
##
## Call:
## lm(formula = log(brain) ~ log(body), data = mammals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71550 -0.49228 -0.06162  0.43597  1.94829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.13479    0.09604   22.23  <2e-16 ***
## log(body)    0.75169    0.02846   26.41  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6943 on 60 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9195
## F-statistic: 697.4 on 1 and 60 DF,  p-value: < 2.2e-16
```

If we let  $\mathbf{y} = [y_1, \dots, y_n]^T$  denote the brain mass and  $\mathbf{x}_b = [x_{b1}, \dots, x_{bn}]^T$  denote the corresponding body mass, we have fitted the model  $\ln(y_i) = \beta_0 + \beta_1 \ln(x_{bi})$ ,  $i = 1, 2, \dots, n$ , with parameter estimates given in the summary above.

b)

The extended model is fitted below.

```
mammals$is.human = as.factor(mammals$species == "Human")

mod1 <- lm(log(brain) ~ log(body) + is.human, data = mammals)
summary(mod1)

##
## Call:
## lm(formula = log(brain) ~ log(body) + is.human, data = mammals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68392 -0.46764 -0.02398  0.47237  1.64949
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11500    0.09030   23.421  < 2e-16 ***
## log(body)    0.74228    0.02687   27.622  < 2e-16 ***
## is.humanTRUE  2.00691    0.66083   3.037  0.00356 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6511 on 59 degrees of freedom
## Multiple R-squared:  0.9315, Adjusted R-squared:  0.9292
## F-statistic: 401.1 on 2 and 59 DF,  p-value: < 2.2e-16
```

Let  $\hat{\beta} = [\hat{\beta}_0 \ \hat{\beta}_1 \ \hat{\beta}_2]^T$  be the coefficient estimates given in the summary above, where  $\hat{\beta}_2 \approx 2.0069072$  models the effect which being a human has on the (log of) brain size. Since we have used a log-transform on both the brain mass and body mass, humans will according to the model be larger by a factor of  $e^{\hat{\beta}_2} =$

7.4402704.

We use the notation  $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$  to represent the linear model. Here,  $X$  is the  $n \times p$  design matrix, where  $n$  is the number of observations and  $p$  is the number of parameters used in the model. Here,  $X = [\mathbf{1} \quad \ln \mathbf{x}_b \quad \mathbf{x}_h]$ , where  $\mathbf{x}_h = [x_{h1}, \dots, x_{hn}]^T = [0, \dots, 0, 1, 0, \dots, 0]^T$  which has a nonzero entry  $x_{hh} = 1$  only for humans only. As usual,  $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$ . We know that

$$\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1}).$$

Now we want to perform the hypothesis test

$$H_0 : \beta_2 = 0 \quad \text{vs.} \quad H_1 : \beta_2 > 0.$$

Under  $H_0$ , we obtain that (we also index from 0 in the design matrix)

$$\frac{\hat{\beta}_2}{\sigma \sqrt{(X^T X)^{-1}_{2,2}}} \sim \mathcal{N}(0, 1).$$

Combining this with the fact that

$$\frac{(n-p)s^2}{\sigma^2} \sim \chi^2_{n-p},$$

where  $s^2 = SSE/(n-p)$ , we obtain the test statistic

$$T_1 = \frac{\hat{\beta}_2}{s \sqrt{(X^T X)^{-1}_{2,2}}} \sim t_{n-p},$$

under  $H_0$ . We perform the calculations in R:

```
n <- nrow(mammals)
p <- 3
beta.2.hat <- mod1$coefficients[3]
s <- sqrt(deviance(mod1)/(n-p))
X <- model.matrix( ~ log(body) + is.human, data = mammals)
XtX.inv <- solve(t(X) %*% X)

T.1 <- beta.2.hat/(s*sqrt(XtX.inv[3,3]))
p.val <- pt(T.1, n - p, lower.tail = F)
p.val
```

```
## is.humanTRUE
## 0.001777696
```

The calculated p-value is 0.0017777.

c)

We now consider all non-human mammals and construct a one-sided prediction interval for the human brain size. For ease of notation, we define  $z_i := \ln y_i$  and  $v_i = \ln(x_{bi})$ . We also let  $n' = n - 1$  as the number of observations (since we exclude humans). Now,  $z_h = \beta_0 + \beta_1 v_h + \varepsilon_h$  is the stochastic variable from which the log of the human brain mass is realized and  $\hat{z}_h = \hat{\beta}_0 + \hat{\beta}_1 v_h$  is the corresponding estimate. Then we can find the pivotal quantity

$$T_2 = \frac{z_h - \hat{z}_h}{s \sqrt{1 + 1/n' + \frac{(v_h - \bar{v})^2}{\sum_{i=1}^{n'} (v_i - \bar{v})^2}}} \sim t_{n'-2}.$$

We refer to the good old [subject-pages](#) (simple linear regression/prediction and prediction intervals in simple linear regression) for this result. Thus, we can find the one-sided prediction interval:

$$P(T_2 < k) = 1 - \alpha \implies k = t_{n'-2, \alpha}.$$

Rearranging, we arrive at

$$P \left( \underbrace{z_h < t_{n'-2, \alpha} \cdot s \sqrt{1 + 1/n' + \frac{(v_h - \bar{v})^2}{\sum_{i=1}^n (v_i - \bar{v})^2}} + \hat{z}_h}_{= \ln U} \right) = 1 - \alpha.$$

Taking both sides of the inequality to the power of  $e$ , we get

$$P(y_h < U) = 1 - \alpha.$$

In accordance with the task description, we define

$$A = \{y_h \notin (-\infty, U)\} = \{y_h \geq U\}, \quad \text{and} \quad B = \{T_1 \geq t_{n-p, \alpha}\}$$

We now observe that  $A$  is equivalent to  $\{T_2 \geq t_{n'-2, \alpha}\} = \{T_2 \geq t_{n-p, \alpha}\}$ , where  $p = 3$  as before. To show that  $A$  and  $B$  are equivalent, we find the MLE of  $\beta_2$  from the model in b) by considering the profile log-likelihood (here  $\mathbf{x}_i$  denotes the  $i$ 'th row of the previously defined design matrix) :

$$\begin{aligned} l_p(\beta_0, \beta_1) &= \sup_{\beta_2} l(\beta_0, \beta_1, \beta_2) \\ &= \sup_{\beta_2} \ln \left( \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \left( \frac{z_i - \mathbf{x}_i^T \boldsymbol{\beta}}{\sigma} \right)^2} \right) \\ &= \sup_{\beta_2} \left( n \ln \left( \frac{1}{\sqrt{2\pi}\sigma} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^n (z_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 \right) \\ &= \sup_{\beta_2} \left( n \ln \left( \frac{1}{\sqrt{2\pi}\sigma} \right) - \frac{1}{2\sigma^2} \sum_{\substack{i=1 \\ i \neq h}}^n (z_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 - \frac{1}{2\sigma^2} (z_h - \mathbf{x}_h^T \boldsymbol{\beta})^2 \right). \end{aligned}$$

Since  $x_{hi}$  is nonzero for only one term in the sum above (for  $i = h$ ) we only need to consider this term. That is, the term with  $\mathbf{x}_h := [1 \quad \ln x_{bh} \quad 1]^T$ . The constant in front of  $(z_h - \mathbf{x}_h^T \boldsymbol{\beta})^2$  is negative, so the supremum is attained when this is equal to zero. Thus,

$$(z_h - \mathbf{x}_h^T \boldsymbol{\beta})^2 = 0 \implies z_h - \beta_0 - \beta_1 v_h - \beta_2 = 0,$$

which means that  $\beta_2 = z_h - \beta_0 - \beta_1 v_h$ . Due to the invariance of MLEs, we now know that

$$\hat{\beta}_2 = z_h - \hat{\beta}_0 - \hat{\beta}_1 v_h = z_h - \hat{z}_h.$$

We also note that the estimators  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are the same here as in the case where we do not consider humans (since the term involving  $\mathbf{x}_h$  in the log-likelihood evaluates to zero). Thus, since both  $T_1$  and  $T_2$  depend on

the same  $\hat{\beta}_2 = z_h - \hat{z}_h$ , meaning that  $A$  and  $B$  occur when the difference  $z_h - \hat{z}_h$  is large, we can conclude that the two events are equivalent.

More precise than this?

d)

For a gamma-distributed random variable, the pdf takes the form

$$f(x; a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}.$$

Using the parametrization  $\mu = \frac{a}{b}$  and  $\nu = a$ , we construct the GLM with a log-link as follows.

$$y_i \sim \text{Gamma}(\mu_i, \nu_i),$$

with  $E[Y_i] = \mu_i$ , such that

$$\ln(\mu_i) = \eta_i = \mathbf{x}_i^T \boldsymbol{\beta}.$$

Next, we fit the model (note that we use the logarithm of the body mass):

```
mod.gamma <- glm(brain ~ log(body) + is.human, family = Gamma(link = "log"), data = mammals)
summary(mod.gamma)
```

```
##
## Call:
## glm(formula = brain ~ log(body) + is.human, family = Gamma(link = "log"),
##      data = mammals)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4464  -0.6099  -0.2276   0.2725   1.8835
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.32733    0.10298  22.601  <2e-16 ***
## log(body)      0.74193    0.03064  24.212  <2e-16 ***
## is.humanTRUE   1.79601    0.75356   2.383   0.0204 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.5512612)
##
##      Null deviance: 310.710  on 61  degrees of freedom
## Residual deviance:  25.849  on 59  degrees of freedom
## AIC: 523.38
##
## Number of Fisher Scoring iterations: 5
```

e)

We want to test whether the following relationship holds (recall earlier notation):

$$y_i = y_0 x_{bi}^{3/4},$$

where  $y_i$  is the brain mass,  $y_0$  is a constant and  $x_{bi}$  is the body mass. Since this is equivalent to testing

$$\ln(y_i) = \ln(y_0) + \frac{3}{4} \ln(x_{bi}),$$

this simply amounts to performing the hypothesis test:

$$H_0 : \beta_1 = \frac{3}{4} \quad \text{vs.} \quad \beta_1 \neq \frac{3}{4}.$$

## Linear model

We first consider the linear model from (b), and construct a Wald test:

```
# Wald test:
C <- matrix(c(0, 1, 0), nrow = 1)
d <- as.vector(3/4)
r <- 1
p <- 3
n <- nrow(mammals)
beta.hat <- mod1$coefficients
s2 <- deviance(mod1)/(n-p)
X <- model.matrix(mod1)
XtX.inv <- solve(t(X) %*% X)

w <- t((C %*% beta.hat - d) %*% solve(s2*C %*% XtX.inv %*% t(C)) %*% (C %*% beta.hat - d))
p.val <- pchisq(w, r, lower.tail = FALSE)
p.val
```

```
##           [,1]
## [1,] 0.7737976
```

The likelihood-ratio test for the linear model can be carried out as follows:

```
mod1.offset <- lm(log(brain) ~ is.human, offset = 3/4*log(body), data = mammals)
A <- logLik(mod1.offset)
B <- logLik(mod1)

X.stat <- -2 * (as.numeric(A)-as.numeric(B))
p.val <- pchisq(X.stat, df = r, lower.tail = FALSE)
p.val
```

```
## [1] 0.7683572
```

```
# Use this instead? ask about this?
anova(mod1, mod1.offset, test = 'LRT')
```

```
## Analysis of Variance Table
##
## Model 1: log(brain) ~ log(body) + is.human
## Model 2: log(brain) ~ is.human
##   Res.Df    RSS Df Sum of Sq Pr(>Chi)
## 1      59 25.013
## 2      60 25.048 -1  -0.03502  0.7738
```

## Gamma-GLM

For a generalized linear model, the Wald statistic can be written as

$$w = (C\hat{\beta} - d)^T [CF^{-1}(\hat{\beta})C^T]^{-1} (C\hat{\beta} - d),$$

which is asymptotically  $\chi^2$ -distributed with  $r = \text{rank}(C)$  degrees of freedom. We compute its value:

```

beta.hat <- as.vector(mod.gamma$coefficients)
w <- t(C %*% beta.hat - d) %*% solve(C %*% vcov(mod.gamma) %*% t(C)) %*% (C %*% beta.hat - d)

p.val <- pchisq(w, r, lower.tail = FALSE)
p.val

```

```

##           [,1]
## [1,] 0.7922823

```

Next, we perform an LR-test for the GLM:

```

mod.gamma.offset <- glm(brain ~ 1 + is.human, family = Gamma(link = "log"), offset = 3/4*log(body), data = mammals)

A <- logLik(mod.gamma.offset)
B <- logLik(mod.gamma)

X.stat <- -2 * (as.numeric(A) - as.numeric(B))
p.val <- pchisq(X.stat, df = r, lower.tail = FALSE)
p.val

```

```

## [1] 0.7762338

```

We observe that the p-values for the Wald and LR-test are almost equal for the linear model, while for the GLM, the difference is larger. The reason behind this is that the LR-test and Wald test are equivalent for the linear model. This can be shown by noting that the Wald-statistic is equal to the F-statistic, since  $W = rF = F$  (see Fahrmeir p. 131). The LRT-statistic is, in turn, a strictly monotonic function of the F-statistic, showing that the two tests are equivalent.

For the GLM, on the other hand, this is not the case, and even though the Wald test-statistic,  $w$ , and the LRT-statistic,  $lr$ , are asymptotically equivalent, where  $w, lr \stackrel{a}{\sim} \chi_r^2$  (Fahrmeir p. 664), they can give quite different results for finite samples. The likelihood ratio test is generally considered more reliable, which is connected to the fact that it considers the model under both hypotheses, while the Wald test only considers the model under the alternative hypothesis. Other reasons to prefer the LR-test are listed [here](#).

## f)

We need to be careful comparing the log-likelihoods and hence the AICs of the models, because for the GLM we consider  $Y \sim \text{Gamma}$ , while in the linear model we consider  $\ln Y \sim \text{Normal}$ . To make them comparable, we define  $X := \ln(Y)$ . Then (for the linear model)  $Y = e^X$  and the Jacobian transformation yields a density of

$$f_Y(y) = \left| \frac{\partial x}{\partial y} \right| f_X(x) = \frac{1}{y} f_X(x).$$

This then yields a log-likelihood:

$$l_Y(\beta) = l_X(\beta) - \sum_{i=1}^n \ln y_i,$$

where  $l_X(\beta)$  is the log-likelihood of the original linear model. We implement this ‘correction’ in the calculation of AIC below:

```

p = 3
AIC.linear <- 2*p + 2*(logLik(mod1) - sum(log(mammals$brain)))
AIC.gamma <- 2*p + 2*logLik(mod.gamma)
AIC.linear

```

```
## 'log Lik.' -503.0523 (df=4)
```

```
AIC.gamma
```

```
## 'log Lik.' -509.3768 (df=4)
```

We see that the gamma-GLM is superior to the linear mode with respect to AIC. ### Theoretical skew of log of gamma distribution:

Let  $Y$  be gamma distributed with shape parameter  $a$  and rate parameter  $b$ . The moment generating function for  $\ln Y$  is

$$M_{\ln Y}(t) = E[e^{t \ln Y}] = E[Y^t],$$

where the expectation can be calculated as

$$\begin{aligned} E[Y^t] &= \int_0^\infty \frac{b^a}{\Gamma(a)} y^{t+a-1} e^{-by} dy \\ &= \frac{b^a}{\Gamma(a)} \int_0^\infty y^{t+a-1} e^{-by} dy \\ &= \frac{b^a}{\Gamma(a)} \int_0^\infty \left(\frac{\xi}{b}\right)^{t+a-1} e^{-\xi} \frac{d\xi}{b} \\ &= \frac{b^{-t}}{\Gamma(a)} \int_0^\infty \xi^{t+a-1} e^{-\xi} d\xi \\ &= \frac{b^{-t}}{\Gamma(a)} \Gamma(t+a), \end{aligned}$$

where we used the substitution  $\xi = by$ . The cumulant-generating function is defined as the log of the moment generating function,  $K(t) := \ln M(t)$ , so it follows that

$$K_{\ln Y}(t) = \ln M_{\ln Y}(t) = -t \ln b + \ln \Gamma(t+a) - \ln \Gamma(a).$$

The first cumulat is  $K_{\ln Y}^{(1)}(0) = \left. \frac{dK_{\ln Y}(t)}{dt} \right|_{t=0} = -\ln b + \psi(a)$ , where  $\psi^{(0)}(x) = \frac{\Gamma'(x)}{\Gamma(x)}$  is the digamma function. The subsequent cumulants can be derived using the polygamma functions. Recall that the polygamma function of order  $m$  is defined as

$$\psi^{(m)}(x) = \frac{d^{m+1}}{dx^{m+1}} \ln \Gamma(x),$$

so the subsequent cumulants are  $K_{\ln Y}^{(n)}(t) = \psi^{(n-1)}(a)$  for  $n \geq 2$ .

The skew of a random variable  $X$  with mean  $\mu$  and variance  $\sigma$  is defined as

$$\text{Skew}[X] := E \left[ \left( \frac{X - \mu}{\sigma} \right)^3 \right],$$

so it follows that

$$\text{Skew}[\ln Y] = \frac{E[(\ln Y - E[\ln Y])^3]}{(\text{Var}(\ln Y))^{3/2}},$$

where the numerator is the third central moment, equal to the third cumulant and the variance is equal to the second cumulant. Thus the skew of the log of the gamma distribution is

$$\text{Skew}[\ln Y] = \frac{\psi^{(2)}(a)}{(\psi^{(1)}(a))^{3/2}}.$$

In R, GLM with gamma-distribution assumes the shape parameter  $a$  to be constant. To satisfy this condition, a dispersion parameter  $\phi := \frac{1}{a}$  is introduced, which can be found in the summary. The polygamma functions are calculated using the library **pracma**.



```
psi <- pracma::psi
phi <- summary(mod.gamma)$dispersion
a <- 1/phi

theory.skew <- psi(2,a) / (psi(1,a))^(3/2)
theory.skew
```

```
## [1] -0.8244106
```

This gives the estimate for the skew of the log mammalian brain size given the body size as -0.8244106.

### Sample skew of residuals from the LM in (a):

The sample skew is defined as

$$\text{Sample skew} := \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left[ \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{3/2}}.$$

For the linear model fitted in a), we calculate the sample skew of the residuals:

```
# residuals from LM in a)
x <- residuals(mod0)

m.3 <- 1/length(x) * sum((x - mean(x))^3)
s <- sd(x) # = sqrt(1/(length(x)-1) * sum((x-mean(x))^2))

sample.skew <- m.3/s^3
sample.skew
```

```
## [1] 0.3957011
```

We observe that the estimated skew of the log-gamma distributed variable is negative and larger in absolute value than the sample skew of the residuals of the linear model from (a). This arguably makes the linear model more suitable than gamma-GLM, **because...**

## Problem 2

### Assumptions

In this problem we apply ordinal multinomial regression to data from Norway Chess 2021. The response variable  $y_i$  is the outcome of the  $i$ 'th match. This can be considered an ordered categorical variable

$$y_i = \begin{cases} 1 & , \text{ white win} \\ 2 & , \text{ draw} \\ 3 & , \text{ black win,} \end{cases}$$

which may depend on relative strength of different players, which player plays white and black and the type of game played. The response can be determined by an underlying latent variable  $u_i$ , given by

$$u_i = -\mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

where  $\epsilon_i \stackrel{iid}{\sim} f$ , where  $f$  is some standard distribution with cdf  $F$ . In this model, the event  $y_i = r$  occurs if  $\theta_{r-1} < u_i \leq \theta_r$  for some parameters  $\{\theta_i\}_{i=0}^3$  satisfying

$$-\infty = \theta_0 < \theta_1 < \theta_2 < \theta_3 = \infty.$$

It follows that

$$P(y_i \leq r) = P(u_i \leq \theta_r) = P(\epsilon_i \leq \theta_r + \mathbf{x}_i^T \boldsymbol{\beta}) = F(\theta_r + \mathbf{x}_i^T \boldsymbol{\beta}),$$

so the probability of observing a particular outcome of the  $i$ 'th match becomes

$$\begin{aligned} \pi_{ir} = P(y_i = r) &= P(y_i \leq r) - P(y_i \leq r-1) \\ &= F(\theta_r + \mathbf{x}_i^T \boldsymbol{\beta}) - F(\theta_{r-1} + \mathbf{x}_i^T \boldsymbol{\beta}). \end{aligned}$$

This means that our model returns that white wins whenever  $u_i \leq \theta_1$ , draw if  $\theta_1 < u_i \leq \theta_2$  and black win for  $u_i > \theta_2$ . Since ordinal regression only depend on relative orderings, we do not need to include the intercept  $\beta_0$  in  $\boldsymbol{\beta}$ , as then the values of the thresholds  $\theta$ s could be shifted by subtracting the value of the intercept. Using the notation  $\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_k]^T$ , the unknown parameters are therefore  $\{\theta_1, \theta_2, \beta_1, \dots, \beta_k\}$ .

### About the data

Our data set consists of  $n = 44$  chess matches played with six players, where the set of players are

$$P = \{\text{carlsen, firouzja, karjakin, nepomniachtchi, rapport, tari}\}.$$

To get a better understanding of the data set, we give a recap of the tournament regulations. The matches was played over 10 rounds of classical games. The event was a double round-robin, meaning that two players played against each other in two separate rounds, one time with the white pieces and the other time with black. If the result of a the classical game was a draw, then the two players faced each other again to play an armageddon game with the same pieces as in the classical game. In each round, the six players played the three classical games simultaneously. In total, this results in 30 classical games. The remaining 14 comes from the armageddon games.

For each round, the reward was 3 points for winning the classical game and 0 for loosing. If the classical game ended in a draw the armageddon game yielded 1.5 points for winning and 1 point for loosing. If this game also ended in a draw, the winner is declared to be the player with the black pieces. This means that the outcome “draw” and “black win” gave the same points for the armageddon format.

The complete data set can be seen on the next page.

```
df <- read.csv('data/Norway\ Chess\ 2021.csv')
```

```
df$black = as.factor(df$black)
df$white = as.factor(df$white)
df$type = as.factor(df$type)
df$y <- factor(df$y, ordered=TRUE)
df
```

##	round	white	black	type	y
## 1	1	firouzja	carlsen	classic	2
## 2	1	firouzja	carlsen	armageddon	2
## 3	1	tari	rapport	classic	3
## 4	1	nepomniachtchi	karjakin	classic	1
## 5	2	nepomniachtchi	firouzja	classic	2
## 6	2	nepomniachtchi	firouzja	armageddon	1
## 7	2	carlsen	tari	classic	2
## 8	2	carlsen	tari	armageddon	1
## 9	2	karjakin	rapport	classic	2
## 10	2	karjakin	rapport	armageddon	2
## 11	3	firouzja	karjakin	classic	2
## 12	3	firouzja	karjakin	armageddon	3
## 13	3	tari nepomniachtchi		classic	2
## 14	3	tari nepomniachtchi		armageddon	3
## 15	3	rapport	carlsen	classic	2
## 16	3	rapport	carlsen	armageddon	2
## 17	4	tari	karjakin	classic	2
## 18	4	tari	karjakin	armageddon	3
## 19	4	carlsen nepomniachtchi		classic	2
## 20	4	carlsen nepomniachtchi		armageddon	1
## 21	4	rapport	firouzja	classic	1
## 22	5	firouzja nepomniachtchi		classic	1
## 23	5	tari	carlsen	classic	2
## 24	5	tari	carlsen	armageddon	1
## 25	5	rapport	karjakin	classic	1
## 26	6	carlsen	firouzja	classic	1
## 27	6	rapport	tari	classic	1
## 28	6	karjakin nepomniachtchi		classic	2
## 29	6	karjakin nepomniachtchi		armageddon	1
## 30	7	firouzja nepomniachtchi		classic	1
## 31	7	tari	carlsen	classic	3
## 32	7	rapport	karjakin	classic	1
## 33	8	nepomniachtchi	tari	classic	3
## 34	8	carlsen	rapport	classic	1
## 35	8	karjakin	firouzja	classic	3
## 36	9	tari	firouzja	classic	3
## 37	9	carlsen	karjakin	classic	1
## 38	9	rapport nepomniachtchi		classic	2
## 39	9	rapport nepomniachtchi		armageddon	3
## 40	10	firouzja	rapport	classic	1
## 41	10	nepomniachtchi	carlsen	classic	2
## 42	10	nepomniachtchi	carlsen	armageddon	3
## 43	10	karjakin	tari	classic	2
## 44	10	karjakin	tari	armageddon	1

## Assumptions about the residuals

### Propositional odds model / Cumulative Logit

$$F(x) = \frac{e^x}{1 + e^x}, \quad \epsilon_i \sim \text{Logistic}(0, 1)$$

Using a propositional odds model, the residuals are logistic distributed.

### Cumulative Probit

$$F(x) = \Phi(x), \quad \epsilon_i \sim N(0, 1)$$

Using a cumulative probit model, the residuals are normally distributed.

### Cumulative Gumbel

$$F(x) = e^{-e^{-x}}, \quad \epsilon_i \sim \text{Gumbel}(x; 0, 1) = e^{-(x+e^{-x})}$$

### AICc

Since we have a small number of observations, we also consider AICc, which adds a correction term to AIC, given by

$$\text{AICc} := \text{AIC} + \frac{2k^2 + 2k}{n - k - 1},$$

where it is usually assumed that the residuals are normally distributed. We will ignore this for now, although this is not the case for the cumulative logit and Gumbel models.

## Cumulative logit models without interactions

```
fit <- vglm(y ~ factor(white) + factor(black), family=cumulative(parallel = TRUE, link="logitlink"), data=df)
summary(fit) AIC(fit) »»»> 9e3d40e67f7b1aa4a641cdfdd9ee721aa949ae29
```

### Model 0

Lets first assume that there are no interactions. Define a family of models on the form

$$u_i = -(\alpha_{w(i)} + \beta_{b(i)} + \gamma_{t(i)}) + \varepsilon_i,$$

where  $\alpha_{w(i)}$  is the effect of player  $w(i)$  having white pieces,  $\beta_{b(i)}$  is the effect of player  $b(i)$  having black pieces and  $\gamma_{t(i)}$  is the effect of type  $t(i)$  of game played. Since the design matrix must have full rank, R automatically chooses the reference group for the different parameters. This is by default chosen to be the first in the lexicographic order. As so, “carlsen” will be the reference in both  $\alpha$  and  $\beta$ , and “armageddon” will be the reference for type of game. As there are more classical games, we change the reference type using the `relevel()` function.

```
library(VGAM)

## Loading required package: stats4
## Loading required package: splines
# change "classic" to reference type
df <- within(df, type <- relevel(type, ref = "classic"))

fit.0 <- vglm(y ~ white + black + type,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)

c(AIC = AIC(fit.0), AICc = AICc(fit.0))

##      AIC      AICc
## 108.0063 120.1396
```

## Model 1

Since it could be argued that a given players skill could be the same for playing both white and black, we next consider the model where the effect of playing white is equal to the effect of playing black, meaning  $\alpha_j = -\beta_j$ ,  $j = 1, 2, \dots, k$ . This would be a simpler model since there are fewer parameters to estimate. The model becomes

$$u_i = -(\alpha_{w(i)} - \alpha_{b(i)} + \gamma_{t(i)}) + \varepsilon_i.$$

To make this new model, we will do a dummy-encoding of the players, with value 1 if the player is white and -1 if the player is black. To make the design matrix full rank, we drop `carlsen`.

```
dummy_cols <- fastDummies::dummy_cols

# dummy-encode white, add -1 for black
df.dummy <- dummy_cols(df, select_columns = "white")
for(i in 1:nrow(df.dummy)){
  df.dummy[i, paste("white", df.dummy$black[i], sep="_")] = -1
}
# remove black, white, white_carlsen
df.dummy <- subset(df.dummy, select = -c(black, white, white_carlsen))

# alpha = - beta
fit.1 <- vglm(y ~ . - round, family=cumulative(parallel = TRUE, link="logitlink"), data=df.dummy)
summary(fit.1)

##
## Call:
## vglm(formula = y ~ . - round, family = cumulative(parallel = TRUE,
## link = "logitlink"), data = df.dummy)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    -0.4170     0.3901  -1.069 0.285131
## (Intercept):2     1.5711     0.4593   3.421 0.000625 ***
## typearmageddon   -0.1724     0.6341  -0.272 0.785704
## white_firouzja    -0.5634     0.6710  -0.840 0.401101
## white_karjakin    -1.1312     0.7048  -1.605 0.108514
## white_nepomniachtchi -0.9148     0.6281  -1.457 0.145247
## white_rapport     -0.5339     0.6805  -0.785 0.432715
## white_tari        -1.8626     0.6895  -2.701 0.006905 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 85.4581 on 80 degrees of freedom
##
## Log-likelihood: -42.7291 on 80 degrees of freedom
##
## Number of Fisher scoring iterations: 6
##
## No Hauck-Donner effect found in any of the estimates
##
##
```

```
## Exponentiated coefficients:
##      typearmageddon      white_firouzja      white_karjakin
##      0.8416380          0.5692578          0.3226461
## white_nepomniachtchi      white_rapport      white_tari
##      0.4005825          0.5862987          0.1552702
c(AIC = AIC(fit.1), AICc = AICc(fit.1))

##      AIC      AICc
## 101.4581 105.5724
```

## Model 2

Same as model 1, but here we also assume that there are no effect of type of game played, thus

$$u_i = -(\alpha_{w(i)} - \alpha_{b(i)}) + \varepsilon_i.$$

```
# alpha = - beta, gamma = 0
fit.2 <- vglm(y ~ . - type - round, family=cumulative(parallel = TRUE, link="logitlink"), data=df.dummy)
summary(fit.2)
```

```
##
## Call:
## vglm(formula = y ~ . - type - round, family = cumulative(parallel = TRUE,
##   link = "logitlink"), data = df.dummy)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1      -0.4608    0.3398  -1.356 0.175075
## (Intercept):2       1.5219    0.4100   3.712 0.000206 ***
## white_firouzja      -0.5832    0.6659  -0.876 0.381153
## white_karjakin      -1.1616    0.6971  -1.666 0.095655 .
## white_nepomniachtchi -0.9258    0.6278  -1.475 0.140274
## white_rapport       -0.5462    0.6770  -0.807 0.419719
## white_tari          -1.8665    0.6878  -2.714 0.006657 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 85.5253 on 81 degrees of freedom
##
## Log-likelihood: -42.7626 on 81 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##      white_firouzja      white_karjakin white_nepomniachtchi
##           0.5581001           0.3129888           0.3962170
##      white_rapport      white_tari
##           0.5791189           0.1546632
c(AIC = AIC(fit.2), AICc = AICc(fit.2))

##      AIC      AICc
## 99.52528 102.63639
```



### Model 3

Similar as model 3, but we use “parallel = FALSE ~ type”.

```
# alpha = - beta, gamma = 0
fit.3 <- vglm(y ~ . - type - round, family=cumulative(parallel = FALSE ~ type, link="logitlink"), data=
summary(fit.3)

##
## Call:
## vglm(formula = y ~ . - type - round, family = cumulative(parallel = FALSE ~
##       type, link = "logitlink"), data = df.dummy)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1      -0.4608    0.3398  -1.356 0.175075
## (Intercept):2       1.5219    0.4100   3.712 0.000206 ***
## white_firouzja     -0.5832    0.6659  -0.876 0.381153
## white_karjakin     -1.1616    0.6971  -1.666 0.095655 .
## white_nepomniachtchi -0.9258    0.6278  -1.475 0.140274
## white_rapport      -0.5462    0.6770  -0.807 0.419719
## white_tari         -1.8665    0.6878  -2.714 0.006657 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 85.5253 on 81 degrees of freedom
##
## Log-likelihood: -42.7626 on 81 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##      white_firouzja      white_karjakin white_nepomniachtchi
##      0.5581001          0.3129888          0.3962170
##      white_rapport      white_tari
##      0.5791189          0.1546632
c(AIC = AIC(fit.3), AICc = AICc(fit.3))

##      AIC      AICc
## 99.52528 102.63639
```

## Model 4

From the models so far, we see that the model with the lowest AIC and AICc is model 2, the model consisting of only the relative strengths of each player. In the summary, we see that the relative strength of some players are close to one another, in particular “firouzja” and “rapport”, but also note that “nepomniachtchi” and “karjakin” are comparable. Let us try fitting a model where we assume that the relative strength of “firouzja” and “rapport” are equal. Thus our model becomes

$$u_i = -(\alpha_{w(i)} - \alpha_{b(i)}) + \varepsilon_i,$$

where  $\alpha_{\text{firouzja}} = \alpha_{\text{rapport}}$ .

```
fit.4 <- vglm(y ~ I(white_firouzja + white_rapport) + white_nepomniachtchi + white_karjakin + white_tari,
              family=cumulative(parallel = FALSE ~ type, link="logitlink"), data=df.dummy)
summary(fit.4)
```

```
##
## Call:
## vglm(formula = y ~ I(white_firouzja + white_rapport) + white_nepomniachtchi +
##       white_karjakin + white_tari, family = cumulative(parallel = FALSE ~
##       type, link = "logitlink"), data = df.dummy)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1         -0.4599    0.3391  -1.356 0.174994
## (Intercept):2          1.5228    0.4096   3.718 0.000201 ***
## I(white_firouzja + white_rapport) -0.5650    0.5737  -0.985 0.324732
## white_nepomniachtchi    -0.9239    0.6267  -1.474 0.140423
## white_karjakin         -1.1648    0.6967  -1.672 0.094547 .
## white_tari             -1.8687    0.6874  -2.718 0.006558 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 85.528 on 82 degrees of freedom
##
## Log-likelihood: -42.764 on 82 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
## I(white_firouzja + white_rapport)          white_nepomniachtchi
##               0.5683715                      0.3969538
##               white_karjakin                      white_tari
##               0.3119803                      0.1543263
c(AIC = AIC(fit.4), AICc = AICc(fit.4))

##      AIC      AICc
## 97.52800 99.79827
```

## Model 5

Let us try fitting a model where we assume that the relative strength of “nepomniachtchi” and “karjakin” are equal. Thus our model becomes

$$u_i = -(\alpha_{w(i)} - \alpha_{b(i)}) + \varepsilon_i,$$

where  $\alpha_{\text{nepomniachtchi}} = \alpha_{\text{karjakin}}$ .

```
fit.5 <- vglm(y ~ I(white_firouzja + white_rapport) + I(white_nepomniachtchi + white_karjakin) + white_tari,
             family=cumulative(parallel = TRUE, link="logitlink"), data=df.dummy)
summary(fit.5)
```

```
##
## Call:
## vglm(formula = y ~ I(white_firouzja + white_rapport) + I(white_nepomniachtchi +
##     white_karjakin) + white_tari, family = cumulative(parallel = TRUE,
##     link = "logitlink"), data = df.dummy)
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1                      -0.4764     0.3376  -1.411  0.15830
## (Intercept):2                       1.5014     0.4064   3.695  0.00022
## I(white_firouzja + white_rapport)    -0.5356     0.5681  -0.943  0.34579
## I(white_nepomniachtchi + white_karjakin) -1.0183     0.5765  -1.766  0.07734
## white_tari                         -1.8252     0.6787  -2.689  0.00716
##
## (Intercept):1
## (Intercept):2                      ***
## I(white_firouzja + white_rapport)
## I(white_nepomniachtchi + white_karjakin) .
## white_tari                          **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 85.6692 on 83 degrees of freedom
##
## Log-likelihood: -42.8346 on 83 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##           I(white_firouzja + white_rapport)
##                                0.5852954
##           I(white_nepomniachtchi + white_karjakin)
##                                0.3611942
##                                white_tari
##                                0.1611832
c(AIC = AIC(fit.5), AICc = AICc(fit.5))

##           AIC           AICc
## 95.66918 97.24812
```

## Model 6

Let us try fitting a model where we assume that the relative strength of all players except 'carlsen' are equal. Thus our model becomes

$$u_i = -(\alpha_{w(i)} - \alpha_{b(i)}) + \varepsilon_i,$$

where  $\alpha_{\text{carlsen}} = 0$  and  $\alpha_{\text{not carlsen}} \neq 0$ .

```
fit.6 <- vglm(y ~ I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari),
             family=cumulative(parallel = TRUE, link="logitlink"), data=df.dummy)
summary(fit.6)
```

```
##
## Call:
## vglm(formula = y ~ I(white_firouzja + white_rapport + white_nepomniachtchi +
##   white_karjakin + white_tari), family = cumulative(parallel = TRUE,
##   link = "logitlink"), data = df.dummy)
##
## Coefficients:
##                                     Estimate
## (Intercept):1                      -0.4268
## (Intercept):2                       1.3925
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari) -0.9628
##                                     Std. Error
## (Intercept):1                       0.3188
## (Intercept):2                       0.3827
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari) 0.4990
##                                     z value
## (Intercept):1                      -1.339
## (Intercept):2                       3.638
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari) -1.929
##                                     Pr(>|z|)
## (Intercept):1                       0.180576
## (Intercept):2                       0.000274
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari) 0.053699
##
## (Intercept):1
## (Intercept):2
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari) .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 89.9104 on 85 degrees of freedom
##
## Log-likelihood: -44.9552 on 85 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
## I(white_firouzja + white_rapport + white_nepomniachtchi + white_karjakin + white_tari)
##                                     0.3818309
```

```
c(AIC = AIC(fit.6), AICc = AICc(fit.6))
```

```
##      AIC      AICc  
## 95.91039 96.51039
```