TMA4315: Project 2

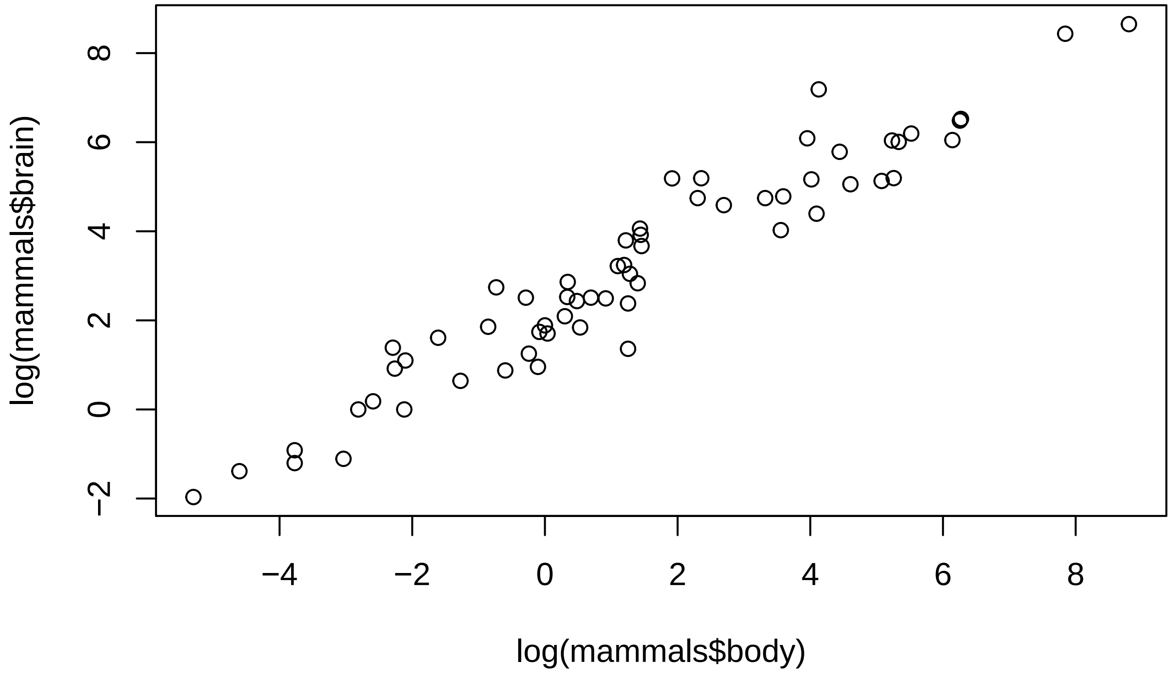
jototlan@stud.ntnu.no (10018), martigtu@stud.ntnu.no (10037)

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 $\boldsymbol{y} = [y_1, \dots, y_n]^T$ denote the brain mass and $\boldsymbol{x_b} = [x_{b1}, \dots, x_{bn}]^T$ 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.

```r
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 $\hat{\boldsymbol{\beta}} = \begin{bmatrix}\hat{\beta}_0 & \hat{\beta}_1 & \hat{\beta}_2\end{bmatrix}^T$ 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 $\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ 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, $X = \begin{bmatrix} \mathbf{1} & \ln \boldsymbol{x}_b & \boldsymbol{x}_h \end{bmatrix}$, where $\boldsymbol{x}_h = [x_{h1}, \dots, x_{hn}]^T = [0, \dots 0, 1, 0, \dots, 0]^T$ 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:

```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. The low p-value implies that the difference between human brain size and expected brain size of mammals with the same body mass is positive.

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 the observation of 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

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

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

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

Raising e to the power of both sides of the inequality, 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 β_2 from the model in b) by considering the profile log-likelihood (here $\boldsymbol{x}_i$ denotes the i'th row of the previously defined design matrix) :

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

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 $\boldsymbol{x}_h := \begin{bmatrix}1 & \ln x_{bh} & 1\end{bmatrix}^T$. The constant in front of $(z_h - \boldsymbol{x}_h^T\boldsymbol{\beta})^2$ is negative, so the supremum is attained when this is equal to zero. Thus,

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

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

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

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 $\boldsymbol{x}_h$ in the log-likelihood evaluates to zero). Thus, since both T_1 and T_2 depend on the same $\hat{\beta}_2 = z_h - \widehat{z}_h$, meaning that A and B occur when the difference $z_h - \widehat{z}_h$ is large, we can conclude that the two events are equivalent.

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 $\text{E}[y_i] = \mu_i$, such that

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

Next, we fit the model:

```
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)

Using the same notation as before, we want to test whether the following relationship holds;

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

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:

```r
# 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:

```r
mod1.offset <- lm(log(brain) ~ is.human, offset = 3/4*log(body), data = mammals)
anova(mod1.offset, mod1, test = 'LRT')
```

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

Gamma-GLM

Next we consider the GLM from (d). For a generalized linear model, the Wald statistic can be written as

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

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

```r
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
```

The liklihood-ratio test for the GLM can be carried out as follows:

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

anova(mod.gamma.offset, mod.gamma, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: brain ~ 1 + is.human
## Model 2: brain ~ log(body) + is.human
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        60     25.881
## 2        59     25.849  1 0.031545   0.8109
```

For both the linear model and for the gamma-GLM, there is no evidence that we should reject H_0, which supports the hypothesis that the relationship $y_i = y_0 x_{bi}^{3/4}$ holds.

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 $w, lr \overset{a}{\sim} \chi^2_r$ (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 $Y \sim$ Gamma, while in the linear model we consider $\ln Y \sim$ 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

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

The connection between the log-likelihoods are thus

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

where $l_X(\boldsymbol{\beta})$ 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

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

where the expectation can be calculated as

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

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 $K^{(1)}_{\ln Y}(0) = \frac{\mathrm{d}K_{\ln Y}(t)}{\mathrm{d}t}\Big|_{t=0} = -\ln b + \psi(a)$, 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

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

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

The skew of a random variable X with mean μ and variance σ is defined as

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

so it follows that

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

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

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

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

The sample skew of a vector x is defined as

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

where $\bar{x}$ is the mean. For the linear model fitted in (a), we calculate the sample skew of the residuals:

```r
# residuals from LM
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 the gamma-GLM unsuitable, since the log-skew does not match the skew of the residuals. However, as previously observed, the gamma-GLM is better with respect to the AIC. Side note: Why do we compare the skew of $\ln Y$ to the sample skew of the residuals from the linear model, i.e. $\ln Y_{\text{lin}} - \ln \widehat{Y}_{\text{lin}}$? The reason is that $Y_{\text{lin}} \sim \mathcal{N}(\boldsymbol{x}_i^T\boldsymbol{\beta}, \sigma^2)$, i.e. its mean depends on the covariates. To remedy this, we subtract the predictions, which gives us the residuals: $\ln Y_{\text{lin}} - \ln \widehat{Y}_{\text{lin}} \sim \mathcal{N}(0, \sigma^2)$. Then we are safe to calculate the sample skew, since the mean is constant.

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

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

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

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

where $\epsilon_i \overset{iid}{\sim} f$, 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

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

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

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

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 β_0 in $\boldsymbol{\beta}$, as then the values of the thresholds θs could be shifted by subtracting the value of the intercept. Using the notation $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \cdots & \beta_k \end{bmatrix}^T$, the unknown parameters are therefore $\{\theta_1, \theta_2, \beta_1, \cdots, \beta_k\}$.

About the data

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

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

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.

```r
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

In our analysis, we will assume that the residuals are homoscedastic, are independent and have zero-mean. In addition, we will also assume that the distribution is symmetric (with no skew). There are two common distributions with these properties to consider. The first is the cumulative logit model, given by

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

Using the cumulative model, the residuals are standard logistic distributed. The second model is the cumulative probit model, given by

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

where the residuals are normally distributed.

Some sources indicate playing white gives a slight advantage over playing black, see for instance here. In addition, the fact that the armageddon games gives the black player no incentive to win the game (since a draw yields the same score), one could argue that the residuals should therefore be slightly negatively skewed. We will ignore this, as the resulting probabilities do not only depended on the shape of F and the value of $\hat{u}_i = -x_i^T \hat{\boldsymbol{\beta}}$, but also on the estimated thresholds $\hat{\theta}_1$ and $\hat{\theta}_2$.

Assumptions of the player stregths

There are multiple factors which can influence the outcome of a chess game. In our data we only have access to the player who played white, the player who played black, what type of game it was and which round the game unfolded. The difference in playing styles, psychology and players earlier encounters might give more insight into the relative strength of the players, but we restrict us to this event only. Here we assume that each player is static in the sense that their relative strengths stays fixed throughout the tournament. In the first set of models, we assume that there are no interactions between individual players and other factors.

Later we will investigate some potential interactions in the data. Some players might have particular areas that they are better at than others, such as playing with a specific color or playing armageddon. Others might get tired during the event, resulting in poorer results in the later rounds.

Model selection

For model selection, we will use AIC and do liklihood-ratio tests comparing different models. 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 assumed that the residuals are normally distributed. Although the residuals in the logit model is not normal, we will still calculate the AICc value here too.

The liklihood-ratio test are based on change in deviance. For two nested models M_1 and M_2, where M_2 contains the parameters in M_1 and an additional k parameters, then under the null hypothesis that M_1 is the true model, it follows from Wilk's theorem that the change in deviance between the two models is approximately chi-squared distributed with k degrees of freedom. This can be written as

$$\Delta D = D_2 - D_1 \overset{\text{a}}{\sim} \chi^2_k.$$

For the hypothesis test

$$H_0 : \text{keep } M_1 \qquad \text{vs.} \qquad H_1 : \text{replace } M_1 \text{ in favor of } M_2,$$

then if the null hypothesis is correct, we reject H_0 with $1-\alpha$ certainty if $\Delta D < \chi^2_{k,\ \alpha}$.

Simple models without interactions

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

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

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, $\delta_{r(i)}$ is the effect of round $r(i)$ 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 $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, 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.

Let the base model be basic model by `fit.all`. We fit the model in R:

```
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
# change "classic" to reference type
df <- within(df, type <- relevel(type, ref = "classic"))

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

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

```
##      AIC     AICc
## 110.0002 124.4829
```

```
anova(fit.all, type = 3, test = "LRT")
```

```
## Analysis of Deviance Table (Type III tests: each term added last)
##
## Model: 'cumulative', 'VGAMordinal', 'VGAMcategorical'
##
## Links: 'logitlink'
##
## Response: y
##
##       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## white  5  10.5206        79     92.521  0.06176 .
## black  5   0.8605        79     82.861  0.97302
## round  1   0.0061        75     82.006  0.93786
## type   1   0.0800        75     82.080  0.77732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fit.all)
```

```
##
## Call:
## vglm(formula = y ~ white + black + round + type, family = cumulative(parallel = TRUE,
##     link = "logitlink"), data = df)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept):1        0.554279   1.334841   0.415   0.6780
## (Intercept):2        2.671673   1.391950   1.919   0.0549 .
## whitefirouzja       -1.159106   1.183802  -0.979   0.3275
## whitekarjakin       -1.615421   1.142305  -1.414   0.1573
## whitenepomniachtchi -1.896386   1.233158  -1.538   0.1241
## whiterapport        -0.907528   1.141104  -0.795   0.4264
## whitetari           -3.235747   1.193313  -2.712   0.0067 **
## blackfirouzja        0.588209   1.097860   0.536   0.5921
## blackkarjakin        0.871262   0.978823   0.890   0.3734
## blacknepomniachtchi  0.464388   0.958249   0.485   0.6279
## blackrapport         0.531689   1.215035   0.438   0.6617
## blacktari            0.399125   1.165903   0.342   0.7321
## round               -0.008377   0.107331  -0.078   0.9378
## typearmageddon      -0.198348   0.666431  -0.298   0.7660
## ---
## 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: 82.0002 on 74 degrees of freedom
##
## Log-likelihood: -41.0001 on 74 degrees of freedom
##
## Number of Fisher scoring iterations: 6
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##       whitefirouzja       whitekarjakin whitenepomniachtchi        whiterapport
##          0.31376644          0.19880692          0.15011015          0.40352031
##           whitetari       blackfirouzja       blackkarjakin blacknepomniachtchi
##          0.03933079          1.80076010          2.38992476          1.59103959
##        blackrapport           blacktari               round      typearmageddon
##          1.70180353          1.49052064          0.99165848          0.82008441
```

From the summary, we observe that `round` and `type` has the highest p-values, so we try to fit a model without `round` and a model without `type`. Denote these models by `fit.no_r` and `fit.no_t`, respectively. Then we compare to the base model with a deviance-based LR-test.

```
fit.no_r <-  vglm(y ~ white + black + type,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
anova(fit.no_r, fit.all, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ white + black + type
## Model 2: y ~ white + black + round + type
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        75     82.006
## 2        74     82.000  1 0.0060779   0.9379
```

```
fit.no_t <-  vglm(y ~ white + black + round,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
anova(fit.no_t, fit.all, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ white + black + round
## Model 2: y ~ white + black + round + type
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        75      82.08
## 2        74      82.00  1  0.07998   0.7773
```

The LR-tests get p-values of 0.9379 and 0.7773 respectively, which with $\alpha = 0.05$ does not yield evidence that the more complicated model are more suitable than the simpler models. In addition, the AIC and AICc are lower for both the simpler models:

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

```
##      AIC     AICc
## 108.0063 120.1396
```

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

```
##      AIC     AICc
## 108.0802 120.2135
```

Based on this, we choose to investigate a simpler model where $\delta = 0$ and $\gamma = 0$. This means we are investigating the model with the form

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

which suggests that the outcome of each game only depend on the relative strengths of playing white and black. Denote this simpler model by `fit.no_rt`. First we compare this model to the first model `fit.all` to test if removing these two parameters are fine.

```
fit.no_rt <-  vglm(y ~ white + black,
                           family=cumulative(parallel = TRUE, link="logitlink"), data=df)
anova(fit.no_rt, fit.all, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ white + black
## Model 2: y ~ white + black + round + type
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        76      82.08
## 2        74      82.00  2 0.080168   0.9607
```

The p-value of 0.9607 suggests again that there are no evidence that the more complicated model `fit.all` is better than the simple model. The AIC for the simple model is:

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

```
##      AIC     AICc
## 106.0803 116.1449
```

Testing different relative strengths

Since it could be argued that the given skill level of a player could be the same for playing both white and black, we next consider a model where the effect of playing white is equal to the effect of playing black, meaning $\alpha_p = -\beta_p, \quad \forall p \in \text{players}$. This would be a simpler model since there are fewer parameters to estimate. The model becomes

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

To make this new model, we dummy-encode the players, with value 1 if the player is white and -1 if the player is black. To make the design matrix have full rank, we remove `carlsen` from the formulas.

```
dummy_cols <- fastDummies::dummy_cols

df <- dummy_cols(df, select_columns = "white")
for(i in 1:nrow(df)){
    df[i, paste("white", df$black[i], sep="_")] = -1
}
df <- subset(df, select = -c(white_carlsen)) # remove carlsen
# change columns ('white_name' -> 'name')
colnames(df)[6:10] <- c("firouzja","karjakin", "nepo", "rapport", "tari")
```

```
fit.simple <- vglm(y ~ firouzja + karjakin + nepo + rapport + tari,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
summary(fit.simple)
```

```
##
## Call:
## vglm(formula = y ~ firouzja + karjakin + nepo + rapport + tari,
##     family = cumulative(parallel = TRUE, link = "logitlink"),
##     data = df)
##
## 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 ***
## firouzja       -0.5832     0.6659  -0.876 0.381153
## karjakin       -1.1616     0.6971  -1.666 0.095655 .
## nepo           -0.9258     0.6278  -1.475 0.140274
## rapport        -0.5462     0.6770  -0.807 0.419719
## 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:
```

```
##  firouzja  karjakin      nepo   rapport      tari
## 0.5581001 0.3129888 0.3962170 0.5791189 0.1546632
```

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

```
##       AIC      AICc
##  99.52528 102.63639
```

Since `fit.no_rt` and `fit.simple` not nested models (they use different covariates), we cannot use the simple LR test performed earlier. However, comparing `fit.no_rt` with `fit.simple`, we observe that `fit.simple` has a lower AIC, suggesting that it fitted the data better.

From the models so far, we see that the model with the lowest AIC and AICc is `fit.simple`, 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 $\alpha_{\text{firouzja}} = \alpha_{\text{rapport}}$. In R we can make this model by simply adding the columns `firourzja` and `rapport`, as both players cannot play the same color simultaneously. Denote this model by `fit.simple2`. We also define the model where in addition to `firourzja` and `rapport`, we assume that `nepomniachtchi` and `karjakin` also have the same strength. Define this model by `fit.simple3`.

```
fit.simple2 <- vglm(y ~ I(firouzja + rapport) + nepo + karjakin + tari,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
c(AIC = AIC(fit.simple2), AICc = AICc(fit.simple2))
```

```
##      AIC     AICc
## 97.52800 99.79827
```

```
fit.simple3 <- vglm(y ~ I(firouzja + rapport) + I(nepo + karjakin) + tari,
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
c(AIC = AIC(fit.simple3), AICc = AICc(fit.simple3))
```

```
##      AIC     AICc
## 95.66918 97.24812
```

Then we use the deviance-based LR-test comparing the simple models:

```
anova(fit.simple2, fit.simple, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ I(firouzja + rapport) + nepo + karjakin + tari
## Model 2: y ~ firouzja + karjakin + nepo + rapport + tari
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        82     85.528
## 2        81     85.525  1 0.0027225   0.9584
```

```
anova(fit.simple3, fit.simple, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ I(firouzja + rapport) + I(nepo + karjakin) + tari
## Model 2: y ~ firouzja + karjakin + nepo + rapport + tari
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        83     85.669
## 2        81     85.525  2   0.1439   0.9306
```

```r
anova(fit.simple2, fit.simple3, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ I(firouzja + rapport) + nepo + karjakin + tari
## Model 2: y ~ I(firouzja + rapport) + I(nepo + karjakin) + tari
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        82     85.528
## 2        83     85.669 -1 -0.14117   0.7071
```

Again, we see no clear evidence to suggest that the more complex model is preferred. To test the limit as how simple the model can be made, we define a model where all players except `carlsen` is assumed to have the same strength. Define this model by `fit.simple4`. Thus our model becomes

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

where $\alpha_p = \alpha_q \ \forall p, q \in \text{players} \setminus \{\text{carlsen}\}$.

```r
fit.simple4 <- vglm(y ~ I(firouzja + rapport+ nepo + karjakin + tari),
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
summary(fit.simple4)
```

```
##
## Call:
## vglm(formula = y ~ I(firouzja + rapport + nepo + karjakin + tari),
##     family = cumulative(parallel = TRUE, link = "logitlink"),
##     data = df)
##
## Coefficients:
##                                                Estimate Std. Error z value
## (Intercept):1                                   -0.4268     0.3188  -1.339
## (Intercept):2                                    1.3925     0.3827   3.638
## I(firouzja + rapport + nepo + karjakin + tari)  -0.9628     0.4990  -1.929
##                                                Pr(>|z|)
## (Intercept):1                                  0.180576
## (Intercept):2                                  0.000274 ***
## I(firouzja + rapport + nepo + karjakin + tari) 0.053699 .
## ---
## 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(firouzja + rapport + nepo + karjakin + tari)
##                                      0.3818309
```

```
c(AIC = AIC(fit.simple4), AICc = AICc(fit.simple4))
```

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

The LR-test becomes:

```
anova(fit.simple4, fit.simple, type = 1, test = 'LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ I(firouzja + rapport + nepo + karjakin + tari)
## Model 2: y ~ firouzja + karjakin + nepo + rapport + tari
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        85     89.910
## 2        81     85.525  4   4.3851   0.3564
```

This, in turn, gives no clear indication of keeping the more complicated model. However, this might not be accurate, as the model gives higher AIC than `fit.simple2`. A problem with these tests is that we have a small number of observations, and the deviance-based LR-test is not necessarily accurate for so few observations. The best model so far is the model where the relative strengths are divided into four levels of strengths.

Models with rating included

```r
rating = data.frame("carlsen"=2855, "firouzja"=2754, "rapport"=2760,
            "nepomniachtchi"=2792, "karjakin"=2758, "tari"=2642)

for(i in 1:nrow(df)){
  df[i,"rating_white"] = rating[df[i,"white"]]
  df[i,"rating_black"] = rating[df[i,"black"]]
}
```

```r
fit.7 <- vglm(y ~ I(rating_white - rating_black) + I((rating_white - rating_black)**3),
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
summary(fit.7)
```

```
##
## Call:
## vglm(formula = y ~ I(rating_white - rating_black) + I((rating_white -
##     rating_black)^3), family = cumulative(parallel = TRUE, link = "logitlink"),
##     data = df)
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1                     -3.628e-01  3.385e-01  -1.072  0.28376
## (Intercept):2                      1.662e+00  4.215e-01   3.943 8.03e-05 ***
## I(rating_white - rating_black)     1.669e-02  6.211e-03   2.687  0.00721 **
## I((rating_white - rating_black)^3) -3.122e-07  1.759e-07  -1.775  0.07585 .
## ---
## 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: 84.2279 on 84 degrees of freedom
##
## Log-likelihood: -42.1139 on 84 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##     I(rating_white - rating_black) I((rating_white - rating_black)^3)
##                          1.0168276                          0.9999997
```

```r
c(AIC = AIC(fit.7), AICc = AICc(fit.7))
```

```
##      AIC     AICc
## 92.22789 93.25353
```

Performance rating (not include this, and the subsequent parts)

```r
info = data.frame("carlsen"=c(0,0,0), "firouzja"=c(0,0,0), "rapport"=c(0,0,0),
                                 "nepomniachtchi"=c(0,0,0), "karjakin"=c(0,0,0), "tari"=c(0,0,0))
# 1: total_opponents_rating
# 2: wins - losses
# 3: n_games

info
```

```
##   carlsen firouzja rapport nepomniachtchi karjakin tari
## 1       0        0       0              0        0    0
## 2       0        0       0              0        0    0
## 3       0        0       0              0        0    0
```

```
for(i in 1:nrow(df)){
  # white win
  if(df[i,'y']==1){
    winner <- as.character(df$white[i])
    looser <- as.character(df$black[i])

    info[1:3,winner] = info[,winner] + c(as.numeric(rating[looser]), 1, 1)
    info[1:3,looser] = info[,looser] + c(as.numeric(rating[winner]),-1, 1)
  }
  # black win
  else if(df[i,'y']==3){
    winner <- as.character(df$black[i])
    looser <- as.character(df$white[i])

    info[1:3,winner] = info[,winner] + c(as.numeric(rating[looser]), 1, 1)
    info[1:3,looser] = info[,looser] + c(as.numeric(rating[winner]),-1, 1)
  }
  # draw
  else {
    winner <- as.character(df$black[i])
    looser <- as.character(df$white[i])

    info[1:3,winner] = info[,winner] + c(as.numeric(rating[looser]), 0, 1)
    info[1:3,looser] = info[,looser] + c(as.numeric(rating[winner]), 0, 1)
  }
}

performance = data.frame("carlsen"=0, "firouzja"=0, "rapport"=0,
            "nepomniachtchi"=0, "karjakin"=0, "tari"=0)

for(name in colnames(info)){
  performance[name] = (info[1,name] + 400*info[2,name])/(info[3,name])
}

# add performance
for(i in 1:nrow(df)){
  df[i,"performance_white"] = performance[df[i,"white"]]
  df[i,"performance_black"] = performance[df[i,"black"]]
}
```

```
fit.8 <- vglm(y ~ I(performance_white - performance_black),
              family=cumulative(parallel = TRUE, link="logitlink"), data=df)
summary(fit.8)
```

```
##
## Call:
```

```
## vglm(formula = y ~ I(performance_white - performance_black),
##     family = cumulative(parallel = TRUE, link = "logitlink"),
##     data = df)
##
## Coefficients:
##                                           Estimate Std. Error z value Pr(>|z|)
## (Intercept):1                            -0.461559   0.321492  -1.436 0.151094
## (Intercept):2                             1.367529   0.380395   3.595 0.000324
## I(performance_white - performance_black)  0.004311   0.002122   2.032 0.042178
##
## (Intercept):1
## (Intercept):2                            ***
## I(performance_white - performance_black) *
## ---
## 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: 89.7557 on 85 degrees of freedom
##
## Log-likelihood: -44.8778 on 85 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
## I(performance_white - performance_black)
##                                   1.00432
```

```
c(AIC = AIC(fit.8), AICc = AICc(fit.8))
```

```
##      AIC     AICc
## 95.75569 96.35569
```