TMA4315: Project 2

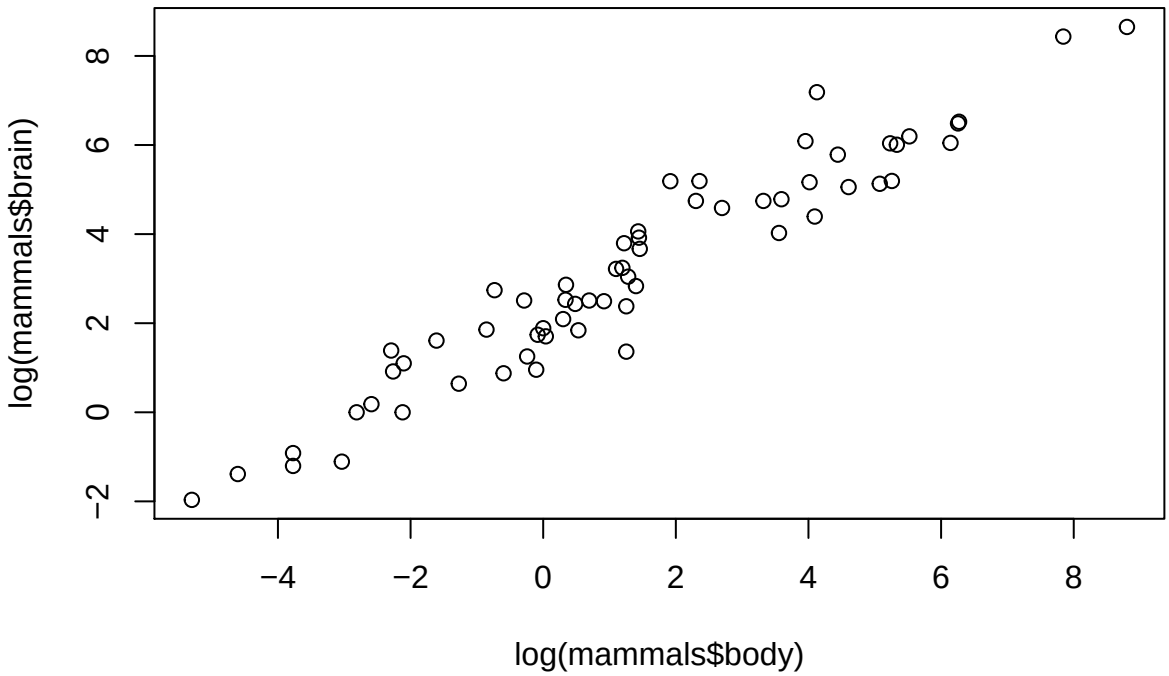
jototlan@stud.ntnu.no (10018), martigtu@stud.ntnu.no (10037)

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
```

This shows that assuming $\ln(brain) = \beta_0 + \beta_1 \ln(body)$ suggest $brain \propto body^{\beta_1}$ with $\beta_1 \approx 0.7516859$.

b)

```
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. Then the estimated effect on brain mass from being a human is $\hat{\beta}_2 \approx 2.0069072$. 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. As usual, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$. This (along with the other usual assumptions how much detail is required here??) gives the well known result:

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

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)

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

Combining this with the fact that

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

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

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

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 (log of) human brain size. Define $n' = n - 1$ as the number of observations and let $Y_h = \beta_0 + \beta_1 x_h + \varepsilon_h$ be the stochastic variable from which the log of the human brain mass is realized and $\widehat{Y}_h = \hat{\beta}_0 + \hat{\beta}_1 x_h$ be the corresponding estimator. Then we can find the pivotal quantity

$$T_2 = \frac{Y_h - \widehat{Y}_h}{s\sqrt{1 + 1/n' + \frac{(x_h - \bar{x})^2}{\sum_{i=1}^{n'}(x_i - \bar{x})^2}}} \sim t_{n'-2}.$$

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

$$P\left(Y_h < \underbrace{t_{n'-2,\ \alpha} \cdot s\sqrt{1 + 1/n' + \frac{(x_h - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}} + \widehat{Y}_h}_{= U}\right) = 1 - \alpha$$

We denote the right hand side of the inequality above by U and in accordance with the task description 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:

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

Since $x_{i,2}$ is nonzero for only one term in the sum above, say for $i = h$, we only need to consider this term. That is, the term with $\boldsymbol{x}_h := \begin{bmatrix} 1 & x_h & 1 \end{bmatrix}^T$. The constant in front of $(y_h - \boldsymbol{x}_h^T \boldsymbol{\beta})^2$ is negative, so the supremum is attained when this is equal to zero. Thus,

$$(y_h - \boldsymbol{x}_h^T \boldsymbol{\beta})^2 = 0 \quad \Longrightarrow \quad y_h - \beta_0 - \beta_1 x_h - \beta_2 = 0,$$

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

$$\hat{\beta}_2 = Y_h - \hat{\beta}_0 - \hat{\beta}_1 x_h = Y_h - \widehat{Y}_h.$$

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 $T_1, T_2 \propto \hat{\beta}_2$, i.e. both A and B occur when the difference $Y_h - \widehat{Y}_h$ is large, we can conclude that they 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. Let the mammalian brain size given body size be given as

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

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 (note that we use the logarithm of the body mass):

```r
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:

$$Y = Y_0 M^{3/4},$$

where Y is the brain mass, Y_0 is a constant and M is the brain mass. Since this is equivalent to testing

$$\ln(Y) = \ln(Y_0) + \frac{3}{4}\ln(M),$$

This simply amounts to performing the hypothesis test:

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

both for the linear model and the GLM. We first consider the linear model, 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:

```
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
```

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:

```
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

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 likelihood ratio is, in turn, a strictly monotonic function of the F-statistic, showing that the two tests are equivalent. need to show this??. For the GLM, on the other hand, this is not the case, and the Wald test becomes an approximation to the LR-test (because it only considers the model under the alternative hypothesis, whereas the LR-test consider the model under both hypotheses).

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

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

This then yields a log-likelihood:

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

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.' -308.36 (df=4)
```

```
AIC.gamma
```

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

Teoretical skew of log of gamma distribution:

Let Y be gamma distributed with shape parameter a and rate paramter 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 $K^{(1)}_{\ln Y}(0) = \left.\frac{\mathrm{d}K_{\ln Y}(t)}{\mathrm{d}t}\right|_{t=0} = -\ln b + \psi(a)$, 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

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

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

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

so it follows that

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

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

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

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 pracma-library.

```
library(pracma)

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(mod1)

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] 0.1832559
```

How does the skew of the log mammalian brain size compare to the sample skew of the residuals of the LM? - The first is negativly skewed, while the residuals are positivly skewed, soo ...

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 $u_i > \theta_2$.

Models

Propositional odds model / Cummulative Logit

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

Cummulative Probit

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

First we consider the model where

$$u_i = -(\alpha_{j(i)} + \beta_{l(i)}) + \varepsilon_i,$$

where $\alpha_{j(i)}$ is the effect of player $j(i)$ having white pieces, and $\beta_{l(i)}$ is the effect of player $l(i)$ having black pieces.

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

library(VGAM)
```

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

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

```
##
## Attaching package: 'VGAM'
```

```
## The following objects are masked from 'package:pracma':
##
##     erf, erfc, expint, logit, loglog, Rank, zeta
```

```
head(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
```

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

```
## [1] 106.0803
```

```
# P(u <= theta_1), P(u <= theta_2)
p.less_or_equal <- plogis(predict(fit, df))


stats <- cbind('white'=df$white, 'black'=df$black,
               'P(white)'=round(p.less_or_equal[,1],2),
               'P(draw)'=round(p.less_or_equal[,2]-p.less_or_equal[,1],2),
               'P(black)'=round(1-p.less_or_equal[,2],2),
               'outcome'=c('white','draw','black')[df$y])
stats
```

```
##    white            black            P(white) P(draw) P(black) outcome
## 1  "firouzja"       "carlsen"        "0.33"   "0.47"  "0.2"    "draw"
## 2  "firouzja"       "carlsen"        "0.33"   "0.47"  "0.2"    "draw"
## 3  "tari"           "rapport"        "0.1"    "0.38"  "0.52"   "black"
## 4  "nepomniachtchi" "karjakin"       "0.36"   "0.46"  "0.17"   "white"
## 5  "nepomniachtchi" "firouzja"       "0.31"   "0.48"  "0.22"   "draw"
## 6  "nepomniachtchi" "firouzja"       "0.31"   "0.48"  "0.22"   "white"
## 7  "carlsen"        "tari"           "0.7"    "0.25"  "0.05"   "draw"
## 8  "carlsen"        "tari"           "0.7"    "0.25"  "0.05"   "white"
## 9  "karjakin"       "rapport"        "0.35"   "0.47"  "0.19"   "draw"
## 10 "karjakin"       "rapport"        "0.35"   "0.47"  "0.19"   "draw"
## 11 "firouzja"       "karjakin"       "0.55"   "0.36"  "0.09"   "draw"
## 12 "firouzja"       "karjakin"       "0.55"   "0.36"  "0.09"   "black"
## 13 "tari"           "nepomniachtchi" "0.09"   "0.36"  "0.55"   "draw"
## 14 "tari"           "nepomniachtchi" "0.09"   "0.36"  "0.55"   "black"
## 15 "rapport"        "carlsen"        "0.39"   "0.45"  "0.16"   "draw"
## 16 "rapport"        "carlsen"        "0.39"   "0.45"  "0.16"   "draw"
## 17 "tari"           "karjakin"       "0.13"   "0.43"  "0.44"   "draw"
## 18 "tari"           "karjakin"       "0.13"   "0.43"  "0.44"   "black"
## 19 "carlsen"        "nepomniachtchi" "0.71"   "0.24"  "0.05"   "draw"
## 20 "carlsen"        "nepomniachtchi" "0.71"   "0.24"  "0.05"   "white"
## 21 "rapport"        "firouzja"       "0.55"   "0.36"  "0.09"   "white"
## 22 "firouzja"       "nepomniachtchi" "0.44"   "0.42"  "0.13"   "white"
## 23 "tari"           "carlsen"        "0.06"   "0.28"  "0.66"   "draw"
## 24 "tari"           "carlsen"        "0.06"   "0.28"  "0.66"   "white"
```

```
## 25 "rapport"        "karjakin"       "0.61"   "0.32"  "0.07"   "white"
## 26 "carlsen"        "firouzja"       "0.74"   "0.22"  "0.04"   "white"
## 27 "rapport"        "tari"           "0.49"   "0.4"   "0.11"   "white"
## 28 "karjakin"       "nepomniachtchi" "0.32"   "0.47"  "0.2"    "draw"
## 29 "karjakin"       "nepomniachtchi" "0.32"   "0.47"  "0.2"    "white"
## 30 "firouzja"       "nepomniachtchi" "0.44"   "0.42"  "0.13"   "white"
## 31 "tari"           "carlsen"        "0.06"   "0.28"  "0.66"   "black"
## 32 "rapport"        "karjakin"       "0.61"   "0.32"  "0.07"   "white"
## 33 "nepomniachtchi" "tari"           "0.26"   "0.48"  "0.26"   "black"
## 34 "carlsen"        "rapport"        "0.73"   "0.23"  "0.04"   "white"
## 35 "karjakin"       "firouzja"       "0.36"   "0.46"  "0.18"   "black"
## 36 "tari"           "firouzja"       "0.1"    "0.39"  "0.51"   "black"
## 37 "carlsen"        "karjakin"       "0.79"   "0.18"  "0.03"   "white"
## 38 "rapport"        "nepomniachtchi" "0.51"   "0.39"  "0.11"   "draw"
## 39 "rapport"        "nepomniachtchi" "0.51"   "0.39"  "0.11"   "black"
## 40 "firouzja"       "rapport"        "0.47"   "0.41"  "0.12"   "white"
## 41 "nepomniachtchi" "carlsen"        "0.19"   "0.47"  "0.34"   "draw"
## 42 "nepomniachtchi" "carlsen"        "0.19"   "0.47"  "0.34"   "black"
## 43 "karjakin"       "tari"           "0.31"   "0.48"  "0.21"   "draw"
## 44 "karjakin"       "tari"           "0.31"   "0.48"  "0.21"   "white"
```

Since it could be argued that a given players skills with one color should be proportional or equal to the skills with another color, we next consider the model where $\alpha_j = \beta_j, \quad j = 1, 2, \ldots, k$. The model becomes

$$u_i = -(\alpha_{j(i)} - \alpha_{l(i)}) + \varepsilon_i.$$

Need to drop one column from the design matrix in order to get full rank. Why? Silus says you can imagine that one effect dissapears into the intercept...

```
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:pracma':
##
##     expm, lu, tril, triu
```

```
# The 'simpler' model from the lecture (effect of player being white is equal when being black)
df$black = as.factor(df$black)
df$white = as.factor(df$white)
X = data.frame(matrix(0, nrow(df), nlevels(df$black)))
colnames(X) <- levels(df$black)
for(i in 1:nrow(df)){
  black = as.character(df$black[i])
  white = as.character(df$white[i])
  X[i,black] = 1
  X[i, white] = -1
}
rankMatrix((X))
```

```
## [1] 5
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
```

```
## [1] 9.769963e-15
```

```
ncol(X)
```

```
## [1] 6
```

```
# X does not have full rank.

X$type = df$type
X$type = as.factor(X$type)
X$y = df$y

fit.simple <- vglm(y ~ ., family=cumulative(parallel = TRUE, link="logitlink"), data=X[2:ncol(X)])
summary(fit.simple)
```

```
##
## Call:
## vglm(formula = y ~ ., family = cumulative(parallel = TRUE, link = "logitlink"),
##     data = X[2:ncol(X)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept):1   -0.5894     0.5591  -1.054   0.2918
## (Intercept):2    1.3987     0.5913   2.366   0.0180 *
## firouzja         0.5634     0.6710   0.840   0.4011
## karjakin         1.1312     0.7048   1.605   0.1085
## nepomniachtchi   0.9148     0.6281   1.457   0.1452
## rapport          0.5339     0.6805   0.785   0.4327
## tari             1.8626     0.6895   2.701   0.0069 **
## typeclassic      0.1724     0.6341   0.272   0.7857
## ---
## 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:
##       firouzja       karjakin nepomniachtchi        rapport           tari
##       1.756673       3.099371       2.496365       1.705615       6.440387
##    typeclassic
##       1.188159
```

```
AIC(fit.simple)
```

```
## [1] 101.4581
```