TMA4315: Project 2

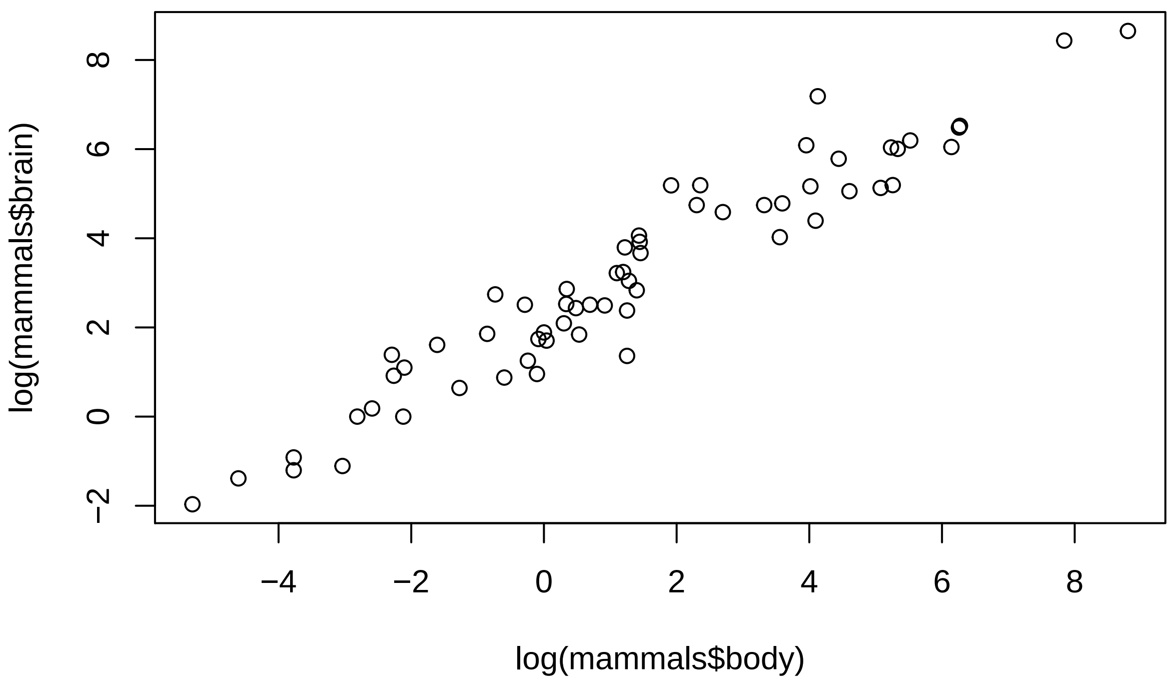
jototlan@stud.ntnu.no (10018), martigtu@stud.ntnu.no (10037)

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 $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ denote the brain mass and $\boldsymbol{x_b} = [x_{b1}, \ldots, x_{bn}]^T$ denote the corresponding body mass, we have fitted the model $\ln(y_i) = \beta_0 + \beta_1 \ln(x_{bi}))$, $i = 1, 2 \ldots n$, 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 $\boldsymbol{x}_h = [x_{h1}, \ldots, x_{hn}]^T = [0, \ldots 0, 1, 0, \ldots, 0]^T$ 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

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

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 $s^2 = SSE/(n-p)$, 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 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

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

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

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

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 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,

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

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.

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 $\text{E}[Y_i] = \mu_i$, 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):

```
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 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 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 $Y \sim \text{Gamma}$, 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:

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

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

```
library(pracma)

phi <- summary(mod.gamma)$dispersion
```

```r
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:

```r
# 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

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

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.

```
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
```

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

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

```
##
## Call:
## vglm(formula = y ~ factor(white) + factor(black), family = cumulative(parallel = TRUE,
##     link = "logitlink"), data = df)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept):1                 0.4212     1.1763   0.358  0.72026
## (Intercept):2                 2.5345     1.2316   2.058  0.03960 *
## factor(white)firouzja        -1.1282     1.1815  -0.955  0.33961
## factor(white)karjakin        -1.6452     1.1249  -1.463  0.14360
## factor(white)nepomniachtchi  -1.8933     1.2274  -1.543  0.12295
## factor(white)rapport         -0.8866     1.1327  -0.783  0.43377
## factor(white)tari            -3.2146     1.1911  -2.699  0.00696 **
## factor(black)firouzja         0.6488     1.0739   0.604  0.54571
## factor(black)karjakin         0.9151     0.9677   0.946  0.34429
## factor(black)nepomniachtchi   0.4857     0.9538   0.509  0.61055
## factor(black)rapport          0.5874     1.1911   0.493  0.62187
## factor(black)tari             0.4298     1.1497   0.374  0.70853
## ---
## 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: 82.0803 on 76 degrees of freedom
##
## Log-likelihood: -41.0402 on 76 degrees of freedom
##
## Number of Fisher scoring iterations: 6
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##       factor(white)firouzja       factor(white)karjakin
##                  0.32360820                  0.19297701
## factor(white)nepomniachtchi        factor(white)rapport
##                  0.15056899                  0.41204954
##           factor(white)tari       factor(black)firouzja
```

```
##                  0.04017041                  1.91332483
##       factor(black)karjakin factor(black)nepomniachtchi
##                  2.49711289                  1.62536741
##        factor(black)rapport           factor(black)tari
##                  1.79938594                  1.53696166
```

```r
AIC(fit)
```

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

```r
# 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

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

To get a design matrix of full rank, we need to remove one of the columns. Here we drop `carlsen`

```r
# 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
}
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   -0.4608     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 .
## nepomniachtchi   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 nepomniachtchi        rapport           tari
##       1.791793       3.195003       2.523869       1.726761       6.465661
```

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

```
## Analysis of Deviance Table
##
## Model 1: y ~ factor(white) + factor(black)
## Model 2: y ~ .
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        76     82.080
## 2        81     85.525 -5  -3.4449   0.6317
```

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

```
## [1] 99.52528
```

```
df.residual(fit)
```

```
## [1] 76
```

```
# Simple with type
X$type = df$type
fit.simple2 <- vglm(y ~ ., family=cumulative(parallel = TRUE, link="logitlink"), data=X[2:ncol(X)])
AIC(fit.simple2)
```

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

```
summary(fit.simple2)
```

```
##
## 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
```

```
fit.simple3 <- vglm(y ~ ., family=cumulative(parallel = FALSE ~ type, link="logitlink"), data=X[2:ncol(X
AIC(fit.simple3)
```

```
## [1] 99.33017
```

```
summary(fit.simple3)
```

```
##
## Call:
## vglm(formula = y ~ ., family = cumulative(parallel = FALSE ~
##     type, link = "logitlink"), data = X[2:ncol(X)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -0.04865    0.60581  -0.080  0.93599
## (Intercept):2   0.90898    0.62527   1.454  0.14602
## firouzja        0.52081    0.68791   0.757  0.44900
## karjakin        1.37989    0.73735   1.871  0.06129 .
## nepomniachtchi  1.00503    0.64912   1.548  0.12155
## rapport         0.56125    0.69782   0.804  0.42123
## tari            2.02188    0.72766   2.779  0.00546 **
## typeclassic:1  -0.62114    0.74021  -0.839  0.40139
## typeclassic:2   1.10561    0.81125   1.363  0.17293
## ---
## 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: 81.3302 on 79 degrees of freedom
##
## Log-likelihood: -40.6651 on 79 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
##      1.6833858      3.9744557      2.7319848      1.7528626      7.5525235
```

```
## typeclassic:1 typeclassic:2
##     0.5373322     3.0210699
```