Week 07 Tools for meta data

Open and reproducible science: dependable computations and statistics

In-class tasks solutions

```
sig.level <- 0.05
library(tidyverse)
```

Penguins

The goal of this in-class task is to see the data and metadata from EDI to partly reproduce https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0090081#pone-0090081-t003

Download the data:

```
download.file("https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter-pal.219.5&entityid=00
```

and read them in with the correct variable types:

```
adelie <- read.csv(here::here("data","table_219.csv"))
adelie$Sex_bin <- as.integer(adelie$Sex=="FEMALE")
adelie$Sex <- as.factor(adelie$Sex)
adelie$`Culmen length` = adelie$Culmen.Length..mm.
adelie$`Culmen depth` = adelie$Culmen.Depth..mm.
adelie$`Body mass` = adelie$Body.Mass..g.
adelie$`Flipper length` = adelie$Flipper.Length..mm.
adelie$`Delta15N` = adelie$Delta.15.N..o.oo.
adelie$`Delta13C` = adelie$Delta.13.C..o.oo.
```

Submission

You will amend the homework file `Homework_06.Rmd`. Also add the R packages you need for the analysis to the file `packages.txt`. Stage the files in Git, commit and push when your are finished or also in between.

Preparation

The paper uses 88 penguins to train its models, the rest of the penguins is used as test data set. We cannot reproduce the paper exactly because we do not know which 88 penguins they used. We need to set a random seed before sampling 88 penguins.

```r
set.seed(12345)
# n_train <- dim(adelie)[1]*(2/3)
n_train <- 88
subset_train <- sample(seq_len(n_train))
subset_test <- seq_len(dim(adelie)[1])[!(seq_len(dim(adelie)[1]) %in% subset_train)]
train <- adelie[subset_train,]
test <-  adelie[subset_test,]
```

Reproducing parts of Table 1:

For the Adelie penguins the following models are fit in the paper:

```r
fit1 <- glm(Sex_bin~`Culmen length`+`Culmen depth`+`Body mass`,data=train,
            family = binomial(link='logit'))
fit2 <- glm(Sex_bin~`Culmen length`+`Body mass`,data=train,
            family = binomial(link='logit'))
fit3 <- glm(Sex_bin~`Culmen length`+`Culmen depth`+`Flipper length`+`Body mass`,data=train,
            family = binomial(link='logit'))
nullmod <- glm(Sex_bin~1, data=train, family=binomial(link='logit'))
```

1

Write a function that calculates the number of correctly classified penguins into the binary sexes. For this use the trained models, the function `predict` on the test data and round the response which is between 0 and 1 in order to classify the test penguins.

```r
prop_correct <- function(fit, newdata){
  # predict label of test data
  preds <- predict(fit, newdata=newdata, type="response")
  # round to get binary label, compare with ground truth
  n_correct <- round(preds) == newdata$Sex_bin
  # get proportion of correctly classified (precision)
  sum(n_correct)/length(n_correct)
}
```

2

The following function mimics the Delta AIC calculation of the paper using the package `MuMIn` and the second function provides a way to extract the pseudo r^2mf. Use these functions to try and reproduce the first three rows of Table 1 of the paper. Hints: The Δ AICc is the difference of each AICc with the minimum AICc of all candidate models. Use the function `Weights` of the `MuMIn` package.

```r
aicc_out <- function(fit){
  chat <- deviance(fit) / df.residual(fit)
  if(chat<1){
    aicc <- MuMIn::AICc(fit)
  } else{
    aicc <- MuMIn::QAICc(fit, chat=chat)
  }
  aicc
```

```
}

r2mf <- function(fit,nullmod){
  1-logLik(fit)/logLik(nullmod)
}
```

```
fitls <- list( fit1,fit2,fit3)

# loop through all models, get the formula, extract the explanatory variable
explanatory_variable <- purrr::map_chr(fitls,~as.character(.x$formula)[3])
# calculate aiccs of models
aiccs <- purrr::map_dbl(fitls,~aicc_out(.x))
# subtract the aicc of the first model
deltaAICc <-aiccs-aiccs[1]
# extract weights
weights <- MuMIn::Weights(aiccs)
# calculate the r2mf with respect to the null model for all models
r2mf_val <- purrr::map_dbl(fitls,~r2mf(.x,nullmod))
# calculate the percentage of correct predictions for all models
percent_correct <- purrr::map_dbl(fitls,~prop_correct(.x,test))*100


df <- tibble::tibble(
  Species = "Adelie penguin",
  `Response variable` = "Sex",
  `Model number`=1:3,
  `Explanatory variable` = explanatory_variable,
  `Number of parameters` = c(4,3,5),
  deltaAICc = round(deltaAICc,3),
  w = round(weights,3),
  r2mf = round(r2mf_val,2),
  `% correctly classified` = round(percent_correct,2)
)

knitr::kable(df) %>%
  kableExtra::kable_styling(font_size = 7,full_width = TRUE)
```

Species	Response variable	Model number	Explanatory variable	Number of parameters	deltaAICc	w	r2mf	% correctly classified
Adelie penguin	Sex	1	‘Culmen length‘ + ‘Culmen depth‘ + ‘Body mass‘	4	0.000	0.615	0.56	90.62
Adelie penguin	Sex	2	‘Culmen length‘ + ‘Body mass‘	3	5.290	0.044	0.50	87.50
Adelie penguin	Sex	3	‘Culmen length‘ + ‘Culmen depth‘ + ‘Flipper length‘ + ‘Body mass‘	5	1.182	0.341	0.57	87.50

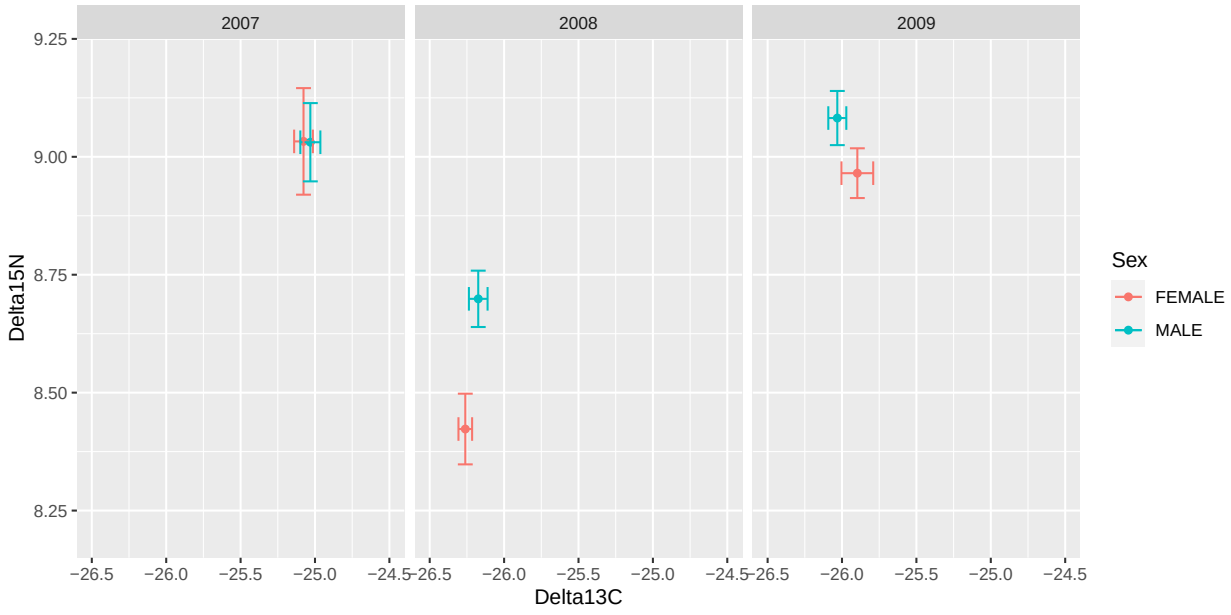
Reproducing Figure 3

3

Use your knowledge of the meta data to reproduce Figure 3 of the paper as closely as possible.

```
adelie_sum <- adelie %>%
  # extract the year
  dplyr::mutate(year=as.integer(stringr::str_extract(Date.Egg,"[:digit:]{4}"))) %>%
  dplyr::group_by(year,Sex) %>%
  # calculate groupwise summaries (groups are rows with the same year and sex)
  dplyr::summarise(Sex=unique(Sex),
                   n=dplyr::n(), # number of elements in group
                   sd_Delta13C=sd(na.omit(Delta13C)), # standard deviation
                   sd_Delta15N=sd(na.omit(Delta15N)), # standard deviation
                   se_Delta13C=sd_Delta13C/sqrt(n), # standard error
                   se_Delta15N=sd_Delta15N/sqrt(n), # standard error
                   Delta13C=mean(na.omit(Delta13C)), # mean
                   Delta15N=mean(na.omit(Delta15N)) # mean
                   ) %>%
  dplyr::filter(Sex != "") %>% # filter groups with missing sex
  dplyr::ungroup()

ggplot(adelie_sum,aes(x=Delta13C,y=Delta15N,color=Sex)) +
  geom_point() +
  facet_wrap(~year) +
  geom_errorbar(aes(xmin=Delta13C-se_Delta13C,xmax=Delta13C+se_Delta13C),
                position = "dodge", width = 0.05) + # horizontal errorbar
  geom_errorbar(aes(ymin=(Delta15N-se_Delta15N),ymax=(Delta15N+se_Delta15N),
                    position = "dodge", width = 0.1)) + # vertical errorbar
  scale_x_continuous(limits=c(-26.5,-24.5)) +
  scale_y_continuous(limits=c(8.2,9.2))
```



Submission

1. Push your Rmd file called `Homework_06.Rmd` (in folder `homework`) to Gitlab.
2. Enter the corresponding Gitlab pages URL into the Open edX text box in the next unit.

Note: the URL will be of the form
`https://opensciencecourse.pages.uzh.ch/hs22_dcas/hs22_dcas/USERNAME_Homework_06.html`,
where `USERNAME` is your Gitlab user name. E.g for user `jdoe`:
`https://opensciencecourse.pages.uzh.ch/hs22_dcas/hs22_dcas/jdoe_Homework_02.html`
In can take a few hours until the webpage is accessible or updated.

Your submission on Open edX will be staff reviewed.