Exam

Nonparametric Statistics, AY 2022/23

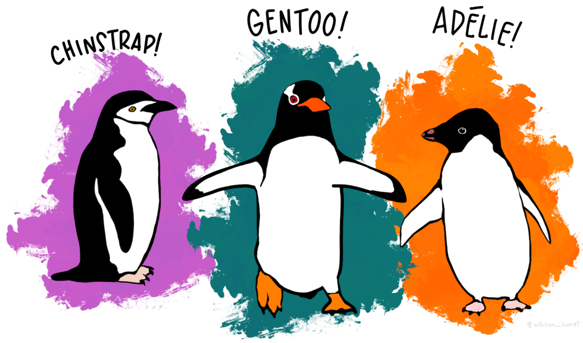
January 19, 2023

Algorithmic Instructions

- All the numerical values required need to be put on an A4 sheet and uploaded, alongside the required plots.
- For all computations based on permutation/bootstrapping, use $B = 1000$ replicates, and $seed = 2022$ every time a permutation/bootstrap procedure is run.
- For Full Conformal prediction intervals, use a regular grid, where, for each dimension, you have $N = 20$ equispaced points with lower bound $\min(data) - 0.25 \cdot range(data)$ and upper bound $\max(data) + 0.25 \cdot range(data)$. Moreover, do not exclude the test point when calculating the conformity measure. Be advised that, except for the number of points, these are the default conditions of the `ConformalInference` R package.
- Both for confidence and prediction intervals, as well as tests, if not specified otherwise, set $\alpha = 0.05$.
- When reporting univariate confidence/prediction intervals, always provide upper and lower bounds.
- Data for the exam can be found at this link

Exercise 1

Dr. Matteus Fontansen is a Norwegian ecologist leading an integrative study of the population structure of Pygoscelis penguins along the western Antarctic Peninsula. To this aim, he has collected information about 333 penguins from three different species, namely Adélie, Chinstrap and Gentoo.



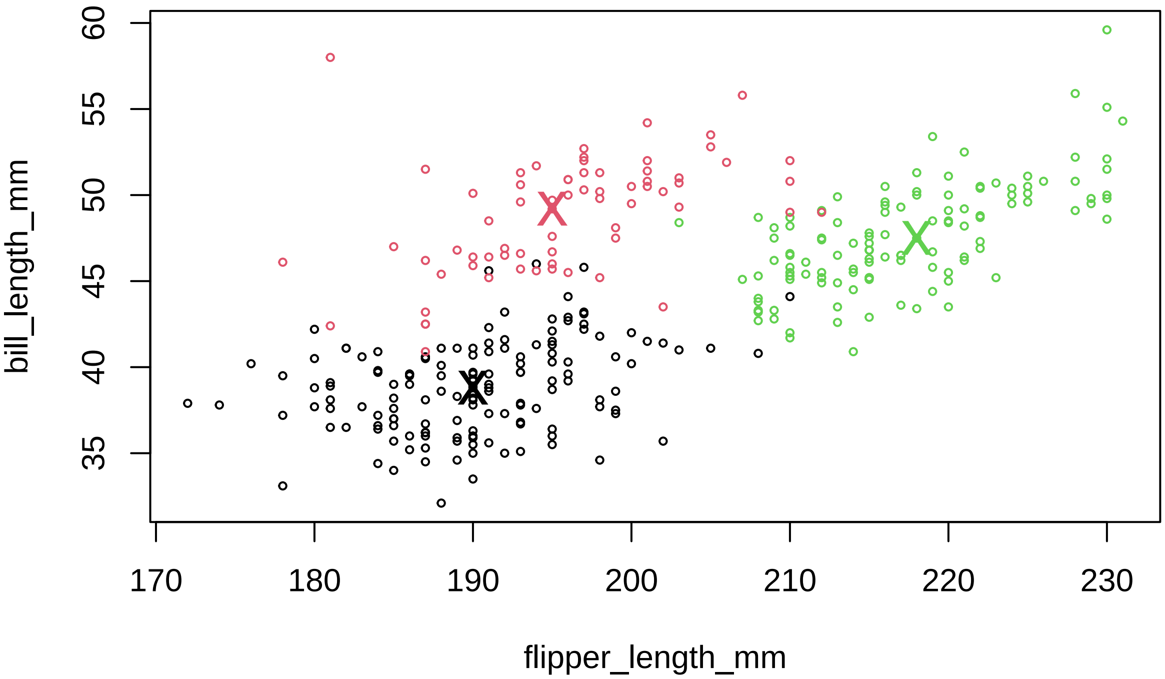
Particularly, he is interested in knowing how the flipper length (`flipper_length_mm`) and the bill length (`bill_length_mm`), both measured in millimeters, vary across species. The resulting samples are contained in the `df_1.Rds` file. Help Dr. Matteus Fontansen by solving the following tasks:

1. Provide the Mahalanobis medians for the three penguin species and superimpose them to the scatterplot of flipper length vs bill length

```
df_1 = readRDS(here("2023-01-19/data/df_1.rds"))
df_split <- split(x = df_1[, 1:2], f = df_1$species)
(maha_medians <-
    lapply(df_split, function(x)
      depthMedian(x, depth_params = list(method = 'Mahalanobis'))))
```

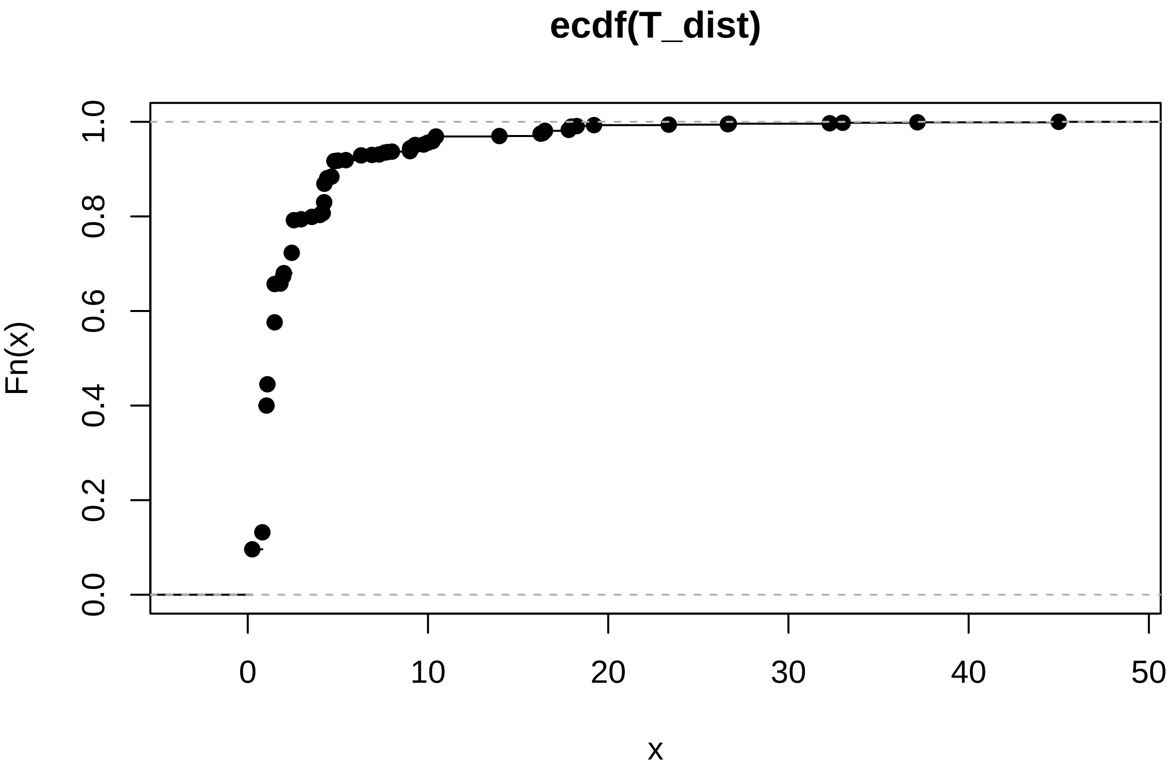
```
## $Adelie
## flipper_length_mm    bill_length_mm
##             190.0              38.8
##
## $Chinstrap
## flipper_length_mm    bill_length_mm
##             195.0              49.2
##
## $Gentoo
## flipper_length_mm    bill_length_mm
##             218.0              47.5
```

```
plot(df_1[, 1:2], col = df_1$species, cex = .5)
for (i in 1:length(maha_medians)) {
  points(
    x = maha_medians[[i]][1],
    maha_medians[[i]][2],
    col = i,
    cex = 2,
    pch = "x"
  )
}
```



2. Test the equality of the theoretical Mahalanobis medians for the Adelie and Chinstrap species by performing a two-sample permutation test using as a test statistics the squared euclidean distance between the two sample Mahalanobis medians. Please describe briefly the properties of permutation tests and present the empirical cumulative distribution function of the permutational test statistic as well as the p-value for the test.

```r
maha_med_Adelie <- maha_medians$Adelie
maha_med_Chinstrap <- maha_medians$Chinstrap

X_Adelie <- df_split$Adelie
X_Chinstrap <- df_split$Chinstrap

X_all <- rbind(X_Adelie,X_Chinstrap)


B=1000
T_dist=numeric(B)

t.stat=sum((maha_med_Chinstrap - maha_med_Adelie)^2)

n1=nrow(X_Adelie)
n2=nrow(X_Chinstrap)

set.seed(2022)
for(index in 1:B){
  perm <- sample(1:(n1+n2))
  X_all.p <- X_all[perm,]
  mean1.p <- depthMedian(X_all.p[1:n1,],depth_params = list(method='Mahalanobis'))
  mean2.p <- depthMedian(X_all.p[(n1+1):(n1+n2),],depth_params = list(method='Mahalanobis'))
  T_dist[index]=sum((mean2.p-mean1.p)^2)
}


plot(ecdf(T_dist))
abline(v=t.stat)
```



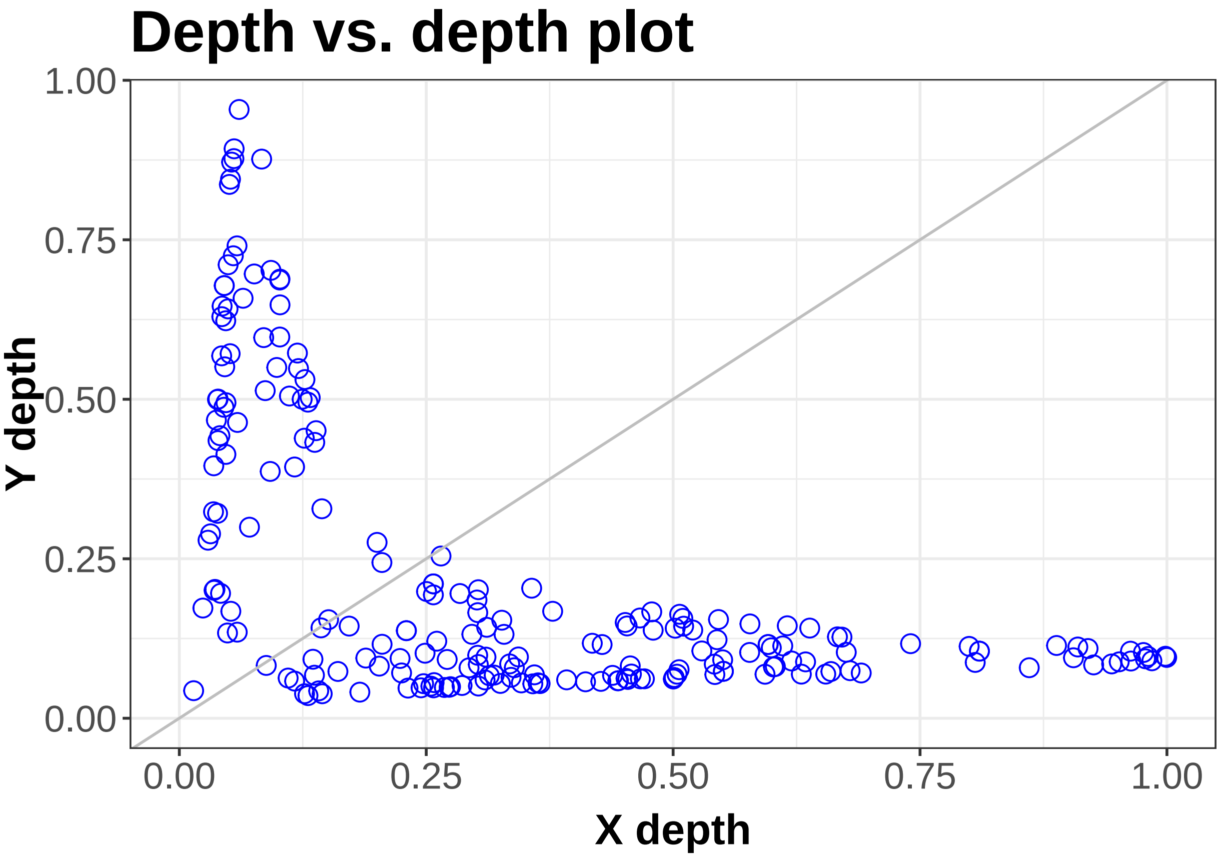
```
sum(T_dist>=t.stat)/B
```

```
## [1] 0
```

3. Employing the Mahalanobis depth, build a DD-plot for the empirical distributions of the Adelie vs Chinstrap species. Can you conclude that the two empirical distributions are identical?

```
ddPlot(x = X_Adelie,y = X_Chinstrap,depth_params = list(method='Mahalanobis'))
```

```
## DDPlot
```



```
##
## Depth Metohod:
##   Mahalanobis
```

```
# The two empirical distributions are not identical
```

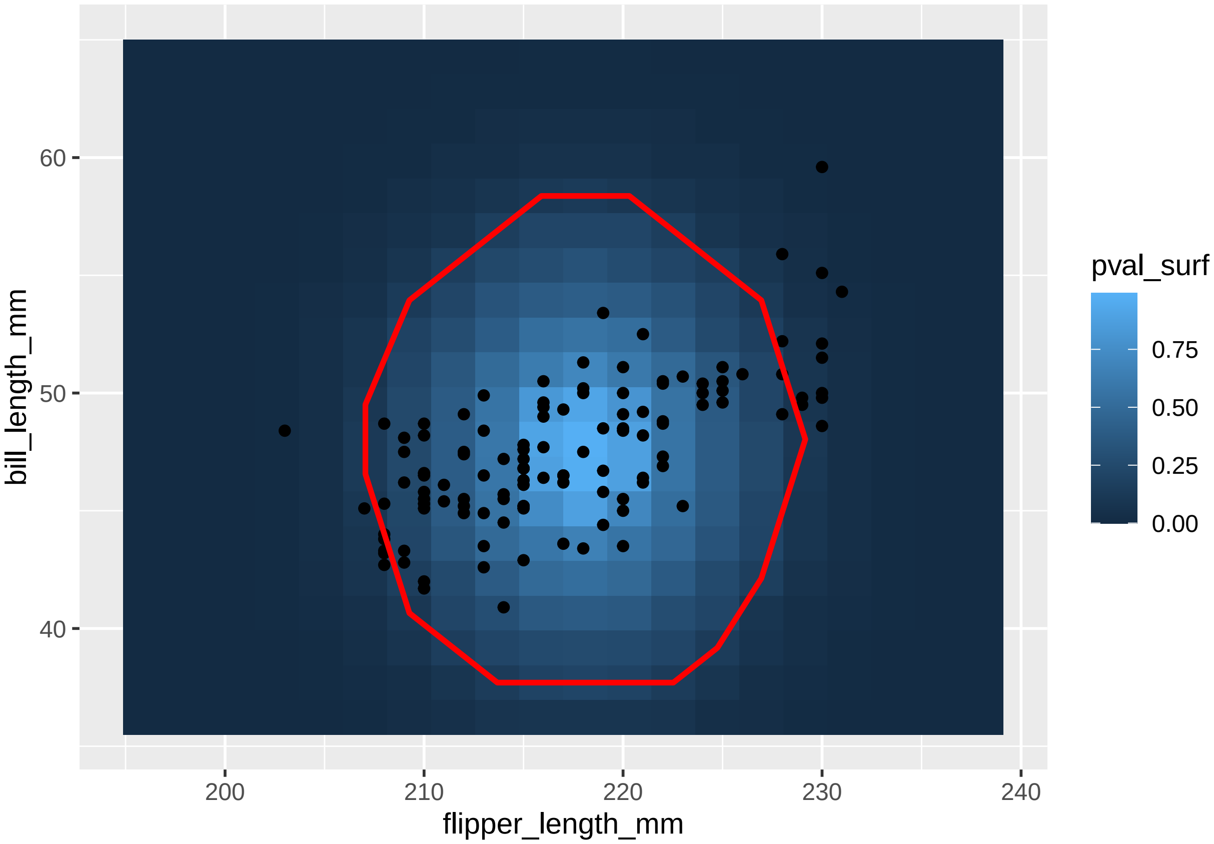
4. Provide a Full Conformal $1 - \alpha = 90\%$ prediction region of flipper and bill lengths for a new Gentoo penguin[1], using the squared euclidean distance between the new data point and the sample Mahalanobis median of the augmented data set as non-conformity measure. After having discussed the theoretical properties of the prediction region, provide a plot of it.

```
data_predict = df_split$Gentoo
n_grid = 20
grid_factor = 0.25
alpha = .1
n = nrow(data_predict)
```

[1] Hint: you are specifically interested in building a prediction region for this species

```r
range_x = range(data_predict[, 1])[2] - range(data_predict[, 1])[1]
range_y = range(data_predict[, 2])[2] - range(data_predict[, 2])[1]


test_grid_x = seq(
  min(data_predict[, 1]) - grid_factor * range_x,
  max(data_predict[, 1]) + grid_factor * range_x,
  length.out = n_grid
)
test_grid_y = seq(
  min(data_predict[, 2]) - grid_factor * range_y,
  max(data_predict[, 2]) + grid_factor * range_y,
  length.out = n_grid
)
xy_surface = expand.grid(test_grid_x, test_grid_y)
colnames(xy_surface) = colnames(data_predict)

wrapper_multi_conf = function(test_point) {
  newdata = rbind(test_point, data_predict)
  newmedian = depthMedian(newdata, depth_params = list(method = 'Mahalanobis'))
  depth_surface_vec = rowSums(t(t(newdata) - newmedian) ^ 2)
  sum(depth_surface_vec[-1] >= depth_surface_vec[1]) / (n + 1)
}


pval_surf = pbapply::pbapply(xy_surface, 1, wrapper_multi_conf)
data_plot = cbind(pval_surf, xy_surface)


p_set = xy_surface[pval_surf > alpha, ]
poly_points = p_set[chull(p_set), ]

ggplot() +
  geom_raster(data = data_plot, aes(flipper_length_mm, bill_length_mm, fill = pval_surf)) +
  geom_point(data = data.frame(data_predict), aes(flipper_length_mm, bill_length_mm)) +
  geom_polygon(
    data = poly_points,
    aes(flipper_length_mm, bill_length_mm),
    color = 'red',
    size = 1,
    alpha = 0.01
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```



Exercise 2

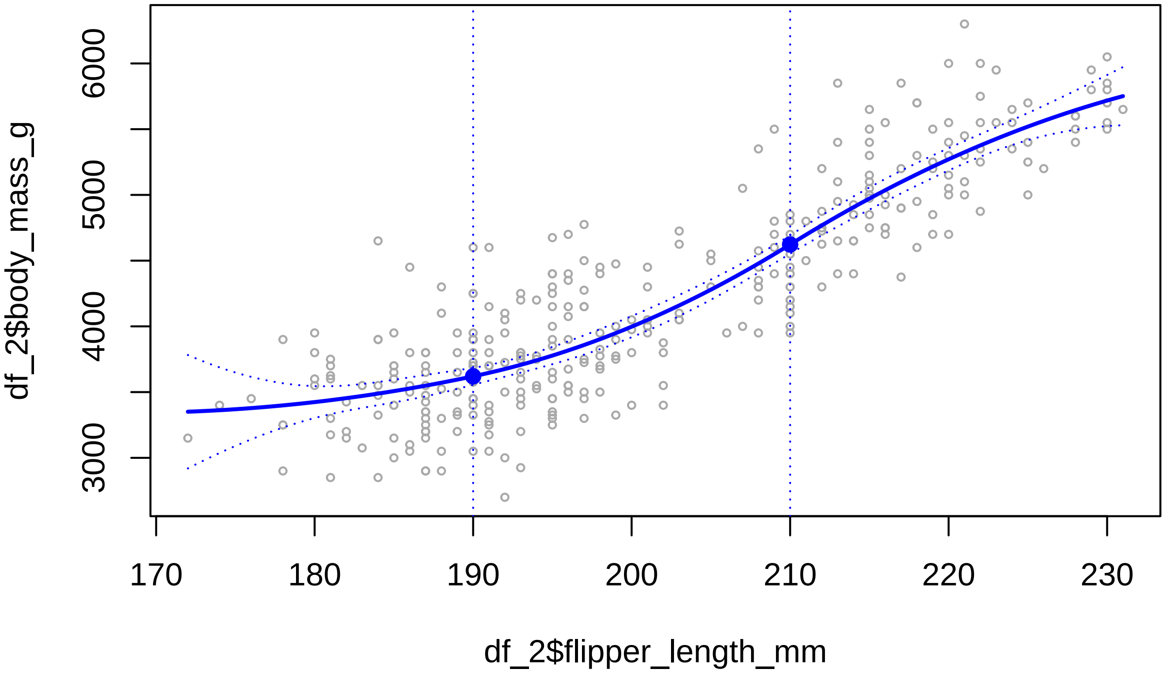
Dr. Matteus Fontansen is now interested in building some regression models to predict the penguins body mass (measured in grams) as a function of species, flipper and bill lengths: the related data are contained in the `df_2.rds` file. He therefore asks you to:

1. Build a quadratic spline model to regress body mass as a function of the flipper length only. Set two knots at 190 and 210 mm for the explanatory variable. Provide a plot of the regression line with standard errors for the prediction, a table summarizing the coefficients and comment the results.

```
df_2=readRDS(here("2023-01-19/data/df_2.rds"))
knots <- c(190,210)
fit_spline_1 <-
  lm(body_mass_g ~ bs(flipper_length_mm, knots = knots, degree = 2),
     data = df_2)

knitr::kable(broom::tidy(summary(fit_spline_1)))
```

term	estimate	std.error	statistic	p.value
(Intercept)	3350.81049	215.7153	15.5334879	0.0000000
bs(flipper_length_mm, knots = knots, degree = 2)1	40.60031	269.0198	0.1509194	0.8801321
bs(flipper_length_mm, knots = knots, degree = 2)2	522.94340	206.2521	2.5354576	0.0116940
bs(flipper_length_mm, knots = knots, degree = 2)3	2058.36883	243.3999	8.4567377	0.0000000
bs(flipper_length_mm, knots = knots, degree = 2)4	2400.58954	236.7316	10.1405517	0.0000000

```
new_data_seq <-
  seq(min(df_2$flipper_length_mm),
      max(df_2$flipper_length_mm),
      length.out = 100)

preds=predict(fit_spline_1, newdata = list(flipper_length_mm=new_data_seq),se=T)
se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)

plot(y=df_2$body_mass_g,x=df_2$flipper_length_mm  ,cex =.5, col =" darkgrey ")
lines(new_data_seq,preds$fit ,lwd =2, col =" blue")
matlines(new_data_seq, se.bands ,lwd =1, col =" blue",lty =3)

knots_pred=predict(fit_spline_1,list(flipper_length_mm=knots))
points(knots,knots_pred, col='blue',pch=19)
abline(v = knots, lty=3, col="blue")
```



2. Build a semiparametric additive model for regressing body mass on the flipper length, bill length and species, using penalized cubic b-spline terms for smoothing the continuous predictors. After having written in proper mathematical terms the additive model you have estimated, report the adjusted R2 and the p-values of the tests.

$$\text{body mass}_i = \beta_0 + f(\text{ flipper length}_i) + f(\text{ bill length}_i) + \beta_1 \text{species}_{i\,Chinstrap} + \beta_2 \text{species}_{i\,Gentoo} + \epsilon_i, i = 1, \dots, N$$

```
fit_gam <- gam(body_mass_g~s(flipper_length_mm,bs = "cr")+
                 s(bill_length_mm,bs = "cr")+ species,data = df_2)
table_fit_gam <- summary(fit_gam)
(r_2_squared <- table_fit_gam$r.sq)
```

```
## [1] 0.8260795
```

```
table_fit_gam$p.pv
```

```
##      (Intercept) speciesChinstrap    speciesGentoo
##    9.774693e-225     5.086259e-15     7.967739e-01
```

```
table_fit_gam$s.table
```

```
##                            edf   Ref.df        F p-value
## s(flipper_length_mm) 2.432081 3.078325 26.04204       0
## s(bill_length_mm)    2.122374 2.708036 24.36025       0
```

3. Build a reduced version of the semiparametric additive model of the previous exercise by letting the covariate `bill_length_mm` enter linearly in the model specification. Use an appropriate Anova F test statistic[2] to assess whether a smooth function is needed for `bill_length_mm`, specifying the null and the alternative hypothesis you are testing and report the resulting p-value. Comment on the results.

```
fit_gam_reduced <-
  gam(body_mass_g ~ s(flipper_length_mm,bs = "cr")+
                 bill_length_mm+ species,data = df_2)
shapiro.test(fit_gam$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit_gam$residuals
## W = 0.99182, p-value = 0.06325
```

```
shapiro.test(fit_gam_reduced$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit_gam_reduced$residuals
## W = 0.99165, p-value = 0.05746
```

```
# Residuals for both models can be assumed to be normally distributed at
# significance level alpha=0.05,
# so I do not need to use a permutational approach
# (I could have done it of course, but it is a lot of work)
anova(fit_gam_reduced, fit_gam, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: body_mass_g ~ s(flipper_length_mm, bs = "cr") + bill_length_mm +
##     species
## Model 2: body_mass_g ~ s(flipper_length_mm, bs = "cr") + s(bill_length_mm,
##     bs = "cr") + species
##   Resid. Df Resid. Dev     Df Deviance      F  Pr(>F)
## 1    326.21   37320942
## 2    324.21   36698951 1.9972   621991 2.7618 0.06474 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# At significance level alpha=0.05 I can be satisfied with the reduced model
```

4. By using a bootstrap approach, provide a reverse percentile confidence interval for the `bill_length_mm` coefficient of the reduced semiparametric additive model employed in the previous exercise.

[2] Hint: if the assumptions are not satisfied a permutational approach shall be used

```
fitted.obs <- fit_gam_reduced$fitted.values
res.obs <- fit_gam_reduced$residuals

bill_length_mm_obs = summary(fit_gam_reduced)$p.table[2,1]
B <- 1000
T.boot_bill_length = numeric(B)

set.seed(2022)

for(b in 1:B) {

  body_mass_boot <- fitted.obs + sample(res.obs, replace = T)
  fit_gam_reduced_boot <-
    gam(body_mass_boot ~ s(df_2$flipper_length_mm,bs = "cr")+
                 df_2$bill_length_mm+ df_2$species)

  fit_gam_reduced_boot_table = summary(fit_gam_reduced_boot)

  T.boot_bill_length[b] = fit_gam_reduced_boot_table$p.table[2,1]

}

alpha <- 0.05
right.quantile.bill_length <- quantile(T.boot_bill_length, 1 - alpha/2)
left.quantile.bill_length  <- quantile(T.boot_bill_length, alpha/2)

CI.RP.bill_length <-
  c(
    bill_length_mm_obs - (right.quantile.bill_length - bill_length_mm_obs),
    bill_length_mm_obs,
    bill_length_mm_obs - (left.quantile.bill_length- bill_length_mm_obs))
names(CI.RP.bill_length)=c('lwr','pointwise','upr')
CI.RP.bill_length
```

```
##       lwr pointwise       upr
##  42.76339  58.04810  71.49593
```

Exercise 3

Dr. Matteus Fontansen is afraid some of the measurements he has collected may have been wrongly recorded. To this extent, he is interested in performing some statistical analyses using robust methods.

1. Focusing on the Gentoo species, compute the Minimum Covariance Determinant estimator for the `flipper_length_mm` and `bill_length_mm` variables contained in the `df_3.rds` dataset. Consider 1000 subsets for initializing the algorithm and set the sample size of H, the subset over which the determinant is minimized, equal to 100. Report the (reweighted) MCD estimates of location and scatter. Define a vector `ind_out_MCD` of row indexes identifying the samples (within the Gentoo subpopulation) that are outliers according to the MCD call and report it.

```
df_3 <- readRDS(here("2023-01-19/data/df_3.rds"))
X_gentoo <- df_3[df_3$species=="Gentoo",2:3]

N <- nrow(X_gentoo)
set.seed(2022)
fit_MCD <-
```

```
  covMcd(
    x = X_gentoo,
    alpha = (N - 19) / N,
    nsamp = 1000
  )

fit_MCD$center
```

```
## flipper_length_mm    bill_length_mm
##         217.15517          47.38534
```

```
fit_MCD$cov
```

```
##                   flipper_length_mm bill_length_mm
## flipper_length_mm          45.02582       13.30863
## bill_length_mm             13.30863        8.93390
```

```
ind_out_MCD <- setdiff(1:N,fit_MCD$best)
ind_out_MCD
```

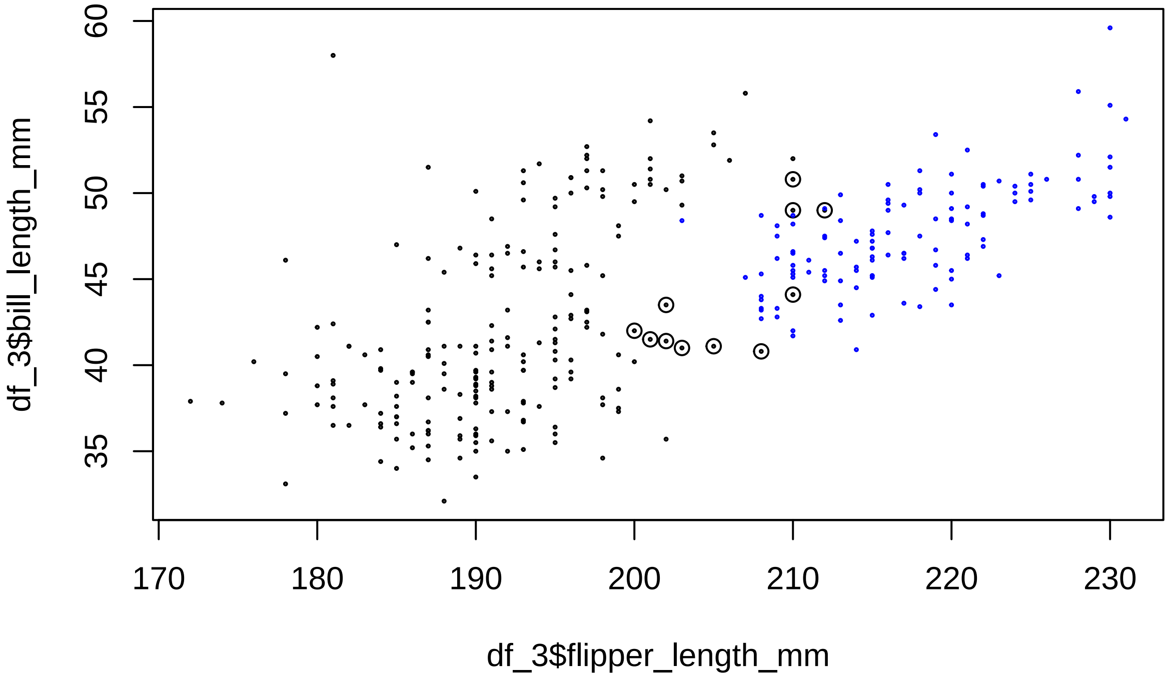
```
##  [1]   2  11  33  39  63  65  66  67  74  78  88  90  97 100 102 105 109 111 113
```

2. Dr. Matteus Fontansen believes some Gentoo penguins have been wrongly labeled as to belong to the Adelie and Chinstrap species. Employing the (reweighted) MCD estimates obtained in the previous exercise, check if some Adelie and Chinstrap penguins have been wrongly labeled by computing robust squared Mahalanobis distances using $\chi^2_{2,0.975}$ as cut-off value, with $\chi^2_{p,\alpha}$ denoting the α-quantile of a χ^2 distribution with p degrees of freedom

```
X_no_gentoo <- df_3[df_3$species != "Gentoo", 2:3]
ind_wrongly_labeled_obs <-
  which(
    mahalanobis(
      x = X_no_gentoo,
      center = fit_MCD$center,
      cov = fit_MCD$cov
    ) <= qchisq(p = .975, df = 2)
  )
length(ind_wrongly_labeled_obs)
```

```
## [1] 11
```

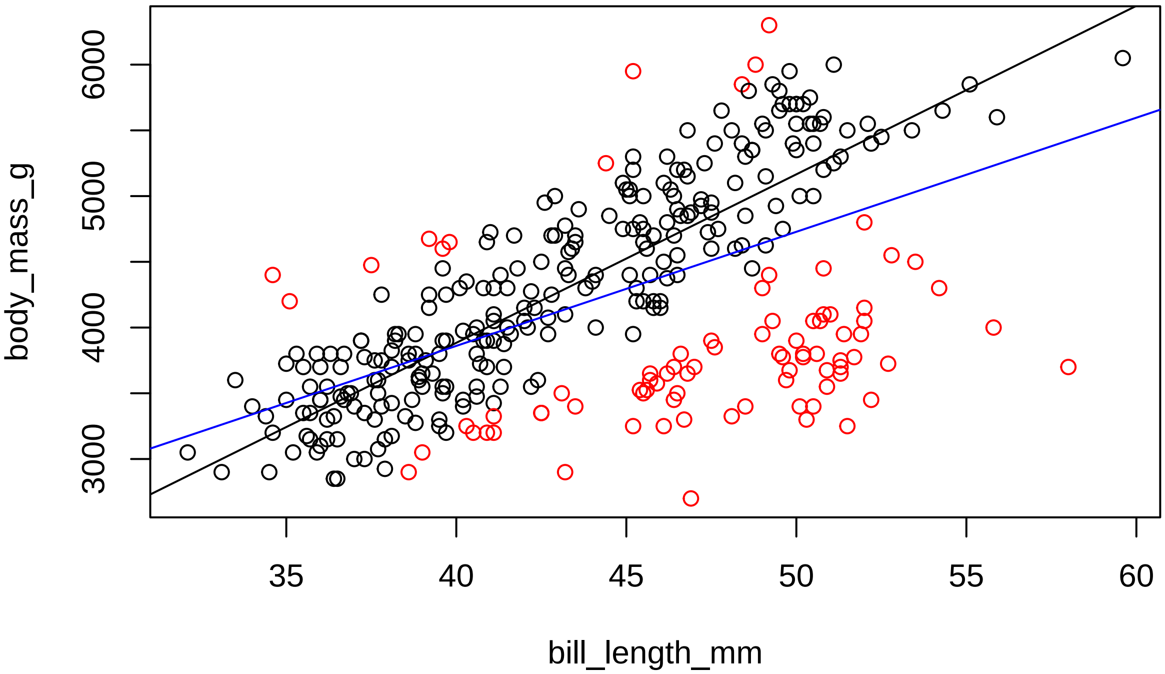
```
# 11 obs may have been wrongly labeled!
plot(df_3$flipper_length_mm, df_3$bill_length_mm, type = "n")
points(X_no_gentoo$flipper_length_mm,
       X_no_gentoo$bill_length_mm,
       cex = .2)
points(X_no_gentoo$flipper_length_mm[ind_wrongly_labeled_obs],
       X_no_gentoo$bill_length_mm[ind_wrongly_labeled_obs])
points(
  X_gentoo$flipper_length_mm,
  X_gentoo$bill_length_mm,
  cex = .2,
  col = "blue"
)
```



3. Since the species variable could be unreliable, Dr. Matteus Fontansen asks you to build a robust linear model for the entire dataset, to regress body mass on the bill length using a Least Trimmed Squares (LTS) approach, setting the hyperparameter $\alpha = 0.75$. Provide a plot of the regression line, flagging the units (i.e., color them in red in the scatterplot) whose squared residuals were not minimized in the LTS call[3]. Superimpose the fit we would obtain if we were to use OLS and comment accordingly.

```
fit_lts <-
  ltsReg(body_mass_g ~ bill_length_mm, alpha = 0.75, data = df_3)

with(df_3,
     plot(bill_length_mm ,
          body_mass_g,
          col = ifelse(1:nrow(df_3)%in% fit_lts$best, "black", "red")))
abline(fit_lts)
abline(lm(body_mass_g ~ bill_length_mm, data = df_3), col="blue")
```

[3] Hint: the `ltsReg` function provides the `best` argument in output that may result useful



```
# Penguins with long bills but low body mass
# can bias the estimates (we know they actually are Chinstrap)
```