Exam

Nonparametric Statistics, AY 2022/23

February 10, 2023

Instructions

- 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.
- For solving the exam, you must use one of the templates previously provided and available here. Particularly, for each question you are required to report:
  - *Synthetic description of assumptions, methods, and algorithms*: which methodological procedure you intend to use to answer the question, succinctly describing the main theoretical characteristics of the chosen approach, and why it is suitable for the analytical task at hand,
  - *Results and brief discussion.* the actual result of the procedure applied to the data at hand, including any requested comment, output and plot.
- Data for the exam can be found at this link.

Exercise 1

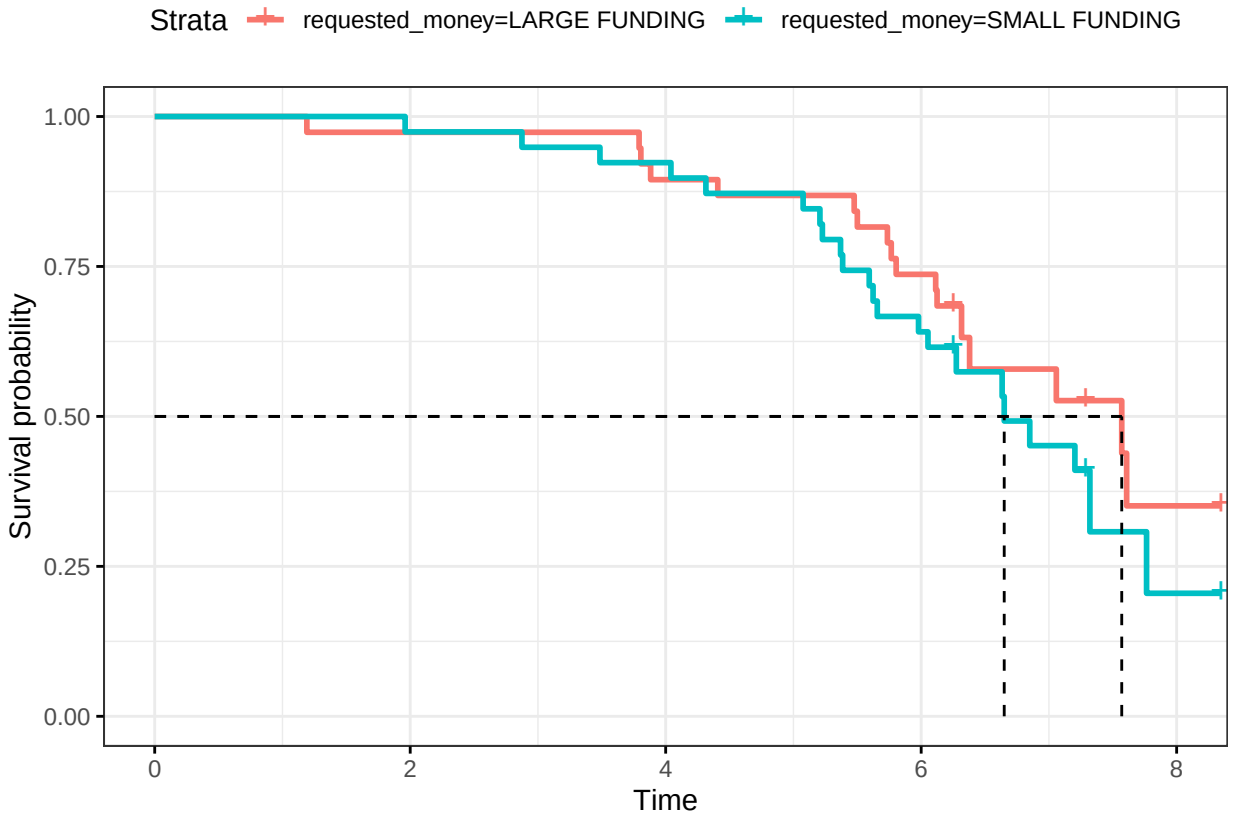
Professor Franziska Iünz, a German biostatistician, is about to apply for a grant. Despite her colleagues being quite positive about her success, she is keen in knowing more about the grants submission process and the subsequent publication of disruptive research. To this extent, she has collected data about 77 previously funded proposals, contained in the `df_1.Rds` file. In details, the following information is recorded:

- `mainpaper.event`: main paper published (1=yes, 0=censored)
- `time.from.funding`: time in years from funding until main paper was published (or censored)
- `members`: number of investigators
- `funding.years`: duration of funding in years
- `requested_money`: a factor indicating whether either SMALL FUNDING or LARGE FUNDING was requested

Help Professor Iünz by solving the following tasks.

1. Compute the Kaplan-Meier curves of the main paper publication event for the two funding types (according to the `requested_money` variable) and plot them. Report the median times for publishing the main paper and test if the time-to-event distributions of the two groups are equal via a Log-rank test. Report the p-value and comment the result.

```
df_1 = readRDS(here("2023-02-10/data/df_1.Rds"))

fit_surv <- survfit(Surv(time = time.from.funding,mainpaper.event)~1,df_1)

fit <-
  survfit(Surv(time.from.funding, mainpaper.event) ~ requested_money, data = df_1)
ggsurvplot(
  fit,
  risk.table = FALSE,
  risk.table.col = "strata",
  surv.median.line = "hv",
  ggtheme = theme_bw(),
)
```




```
surv_median(fit)
```

```
##                          strata   median    lower upper
## 1 requested_money=LARGE FUNDING 7.570157 6.316222    NA
## 2 requested_money=SMALL FUNDING 6.650240 6.053388    NA
```

```
log_rank_test <-
  survdiff(Surv(time.from.funding, mainpaper.event) ~ requested_money,
           data = df_1)
log_rank_test
```

```
## Call:
## survdiff(formula = Surv(time.from.funding, mainpaper.event) ~
##     requested_money, data = df_1)
##
```

```
##                                 N Observed Expected (O-E)^2/E (O-E)^2/V
## requested_money=LARGE FUNDING 38       17     20.1     0.470     0.975
## requested_money=SMALL FUNDING 39       22     18.9     0.498     0.975
##
##  Chisq= 1  on 1 degrees of freedom, p= 0.3
```

```
# No significant difference
```

2. Fit a suitable Cox model for time from funding until publication as a function of all the available covariates. Interpret the estimated coefficient for the `funding.years` covariate, including a comment on statistical significance.

```
fit_cox <- coxph(Surv(time.from.funding, mainpaper.event) ~ ., data = df_1)
summary(fit_cox)
```

```
## Call:
## coxph(formula = Surv(time.from.funding, mainpaper.event) ~ .,
##     data = df_1)
##
##   n= 77, number of events= 39
##
##                                 coef exp(coef)  se(coef)      z Pr(>|z|)
## members                     0.009072  1.009114  0.122151  0.074  0.94079
## funding.years              -0.449022  0.638252  0.168421 -2.666  0.00767 **
## requested_moneySMALL FUNDING  0.037808  1.038531  0.336452  0.112  0.91053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                             exp(coef) exp(-coef) lower .95 upper .95
## members                        1.0091     0.9910    0.7943    1.2821
## funding.years                  0.6383     1.5668    0.4588    0.8879
## requested_moneySMALL FUNDING    1.0385     0.9629    0.5371    2.0082
##
## Concordance= 0.598  (se = 0.055 )
## Likelihood ratio test= 8.71  on 3 df,   p=0.03
## Wald test            = 7.52  on 3 df,   p=0.06
## Score (logrank) test = 7.44  on 3 df,   p=0.06
```

```
# longer study length increases the time to the main paper
```

3. Employing a naive bootstrap approach[1], provide a reverse percentile confidence interval for the `funding.years` coefficient of the Cox model constructed in the previous exercise. Briefly describe the procedure and comment on the result.

```
N <- nrow(df_1)
funding_coef <- fit_cox$coefficients[2]


B <- 1000
T.boot_funding = numeric(B)

set.seed(2022)

for(b in 1:B) {
  boot_id <- sample(x = 1:N,size = N,replace = TRUE)
```

[1] Hint: this boils down to resample with replacement the triplets $(T_i, \delta_i, \mathbf{X}_i), i = 1, \ldots, n$

```
  df_boot <- df_1[boot_id,]

  fit_cox_boot <- coxph(Surv(time.from.funding, mainpaper.event) ~ ., data = df_boot)
  T.boot_funding[b] = fit_cox_boot$coefficients[2]
}

alpha <- 0.05
right.quantile.funding <- quantile(T.boot_funding, 1 - alpha/2)
left.quantile.funding  <- quantile(T.boot_funding, alpha/2)

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

```
##         lwr   pointwise         upr
## -0.80335622 -0.44902176  0.00959247
```

```
# According to naive bootstrap, the coef is not significant at 5%
```

4. Using the previously estimated Cox model, provide an estimate of the median time to publish the main paper for a 2-year long grant with 5 members involved and with LARGE FUNDING requested.

```
new_df <-
  data.frame(
    "members"   =   5,
    "requested_money" = "LARGE FUNDING",
    "funding.years" = 2
  )

fit_new <- survfit(fit_cox, newdata = new_df)
fit_new
```

```
## Call: survfit(formula = fit_cox, newdata = new_df)
##
##       n events median 0.95LCL 0.95UCL
## [1,] 77     39   5.48    4.31      NA
```

Exercise 2

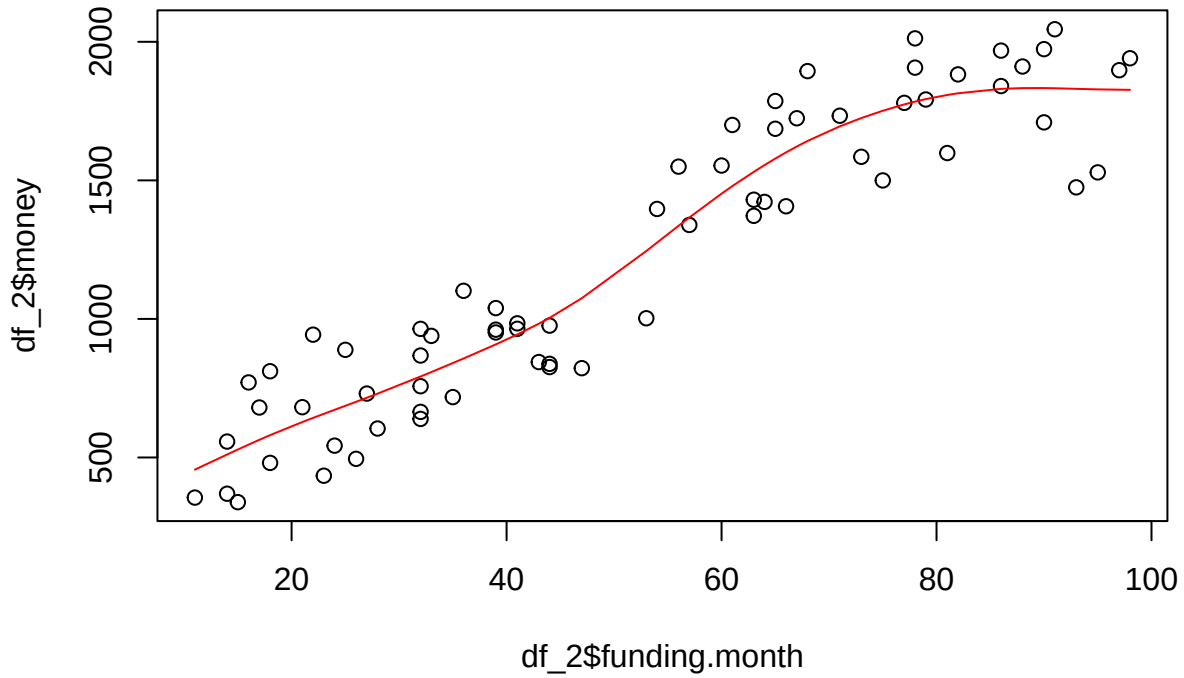
Professor Franziska Iünz is now interested in knowing more about the relationship between the amount of money (in k$) that shall be requested as a function of the duration of funding (measured in month). To do so, she has at her disposal a dataset on 67 proposals, previously funded, contained in the `df_2.Rds` file.

Help Professor Iünz by solving the following tasks.

1. Build a smoothing spline model to regress the `money` response on the `funding.month` variable, selecting λ by means of Generalized Cross Validation (GCV). Provide a plot of the regression line and report the best λ value identified via GCV (round it at the third decimal digit).

```
df_2 <- readRDS(here("2023-02-10/data/df_2.Rds"))
N <- nrow(df_2)

fit_smooth <-
```

```
  smooth.spline(x = df_2$funding.month,
                y = df_2$money,
                cv = FALSE)
plot(df_2$funding.month, df_2$money)
lines(fit_smooth, col="red")
```




```
round(fit_smooth$lambda,3)
```

```
## [1] 0.003
```

2. Ideally, Professor Iünz would like to have a research proposal lasting for five years. Using the model estimated in the previous exercise, provide a point-wise estimate for the amount of money (in k$) she shall request. In addition, by using a bootstrap approach on the residuals, estimate the bias, variance and MSE of such prediction (fix the λ value to the one obtained via Generalized Cross Validation).

```
pred <- predict(fit_smooth,x=60) # 5 years funding
y_hat <- pred$y
y_hat
```

```
## [1] 1451.99
```

```
fitted=predict(fit_smooth,df_2$funding.month)$y

residuals=df_2$money-fitted

l_GCV <- fit_smooth$lambda
l_GCV
```

```
## [1] 0.003107398
```

```
B <- 1000
boot_d=numeric(B)
set.seed(2022)

for(sample in 1:B){
```

```
  money_boot=fitted+sample(residuals,N,replace=T)
  new_model = smooth.spline(x = df_2$funding.month,
                            y = money_boot,
                            lambda = l_GCV)
  boot_d[sample]=predict(new_model, 60)$y
}


(variance_fit <- var(boot_d))
```

```
## [1] 1239.703
```

```
(bias_fit=mean(boot_d)-y_hat)
```

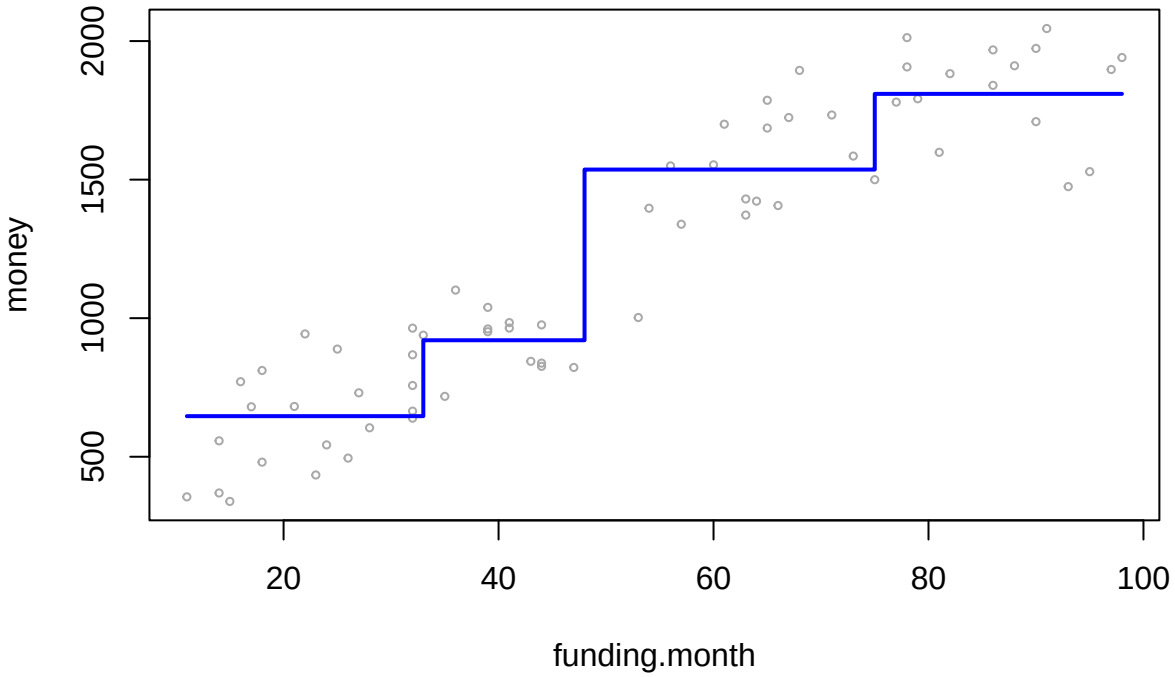
```
## [1] -9.529852
```

```
(MSE_fit= variance_fit + bias_fit^2)
```

```
## [1] 1330.521
```

3. Build a nonparametric model to regress the `money` response on the `funding.month` variable via a step regression procedure, breaking the range of `funding.month` into 4 bins, using the quartiles of the covariate as breaking points. Provide a plot of the regression line and report the summary table including a comment on statistical significance.

```
funding.month_quart <- quantile(df_2$funding.month)
m_cut=lm(money ~ cut(funding.month,breaks=c(-Inf,funding.month_quart[c(-1,-5)],Inf)),data = df_2)

month_grid=with(df_2, seq(range(funding.month)[1],range(funding.month)[2],by=1))
preds=predict(m_cut,list(funding.month=month_grid),se=T)
# se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)
with(df_2, plot(funding.month ,money ,xlim=range(month_grid) ,cex =.5, col =" darkgrey "))
lines(month_grid,preds$fit ,lwd =2, col =" blue",type="s")
```



```
# matlines(month_grid ,se.bands ,lwd =1, col =" blue",lty =3,type="s")

coef(m_cut)
```

```
##                                                                                (Intercept)
##                                                                                   646.4966
##   cut(funding.month, breaks = c(-Inf, funding.month_quart[c(-1, -5)], Inf))(32,47]
##                                                                                   273.8614
##   cut(funding.month, breaks = c(-Inf, funding.month_quart[c(-1, -5)], Inf))(47,74]
##                                                                                   889.8319
## cut(funding.month, breaks = c(-Inf, funding.month_quart[c(-1, -5)], Inf))(74, Inf]
##                                                                                  1162.9517
```

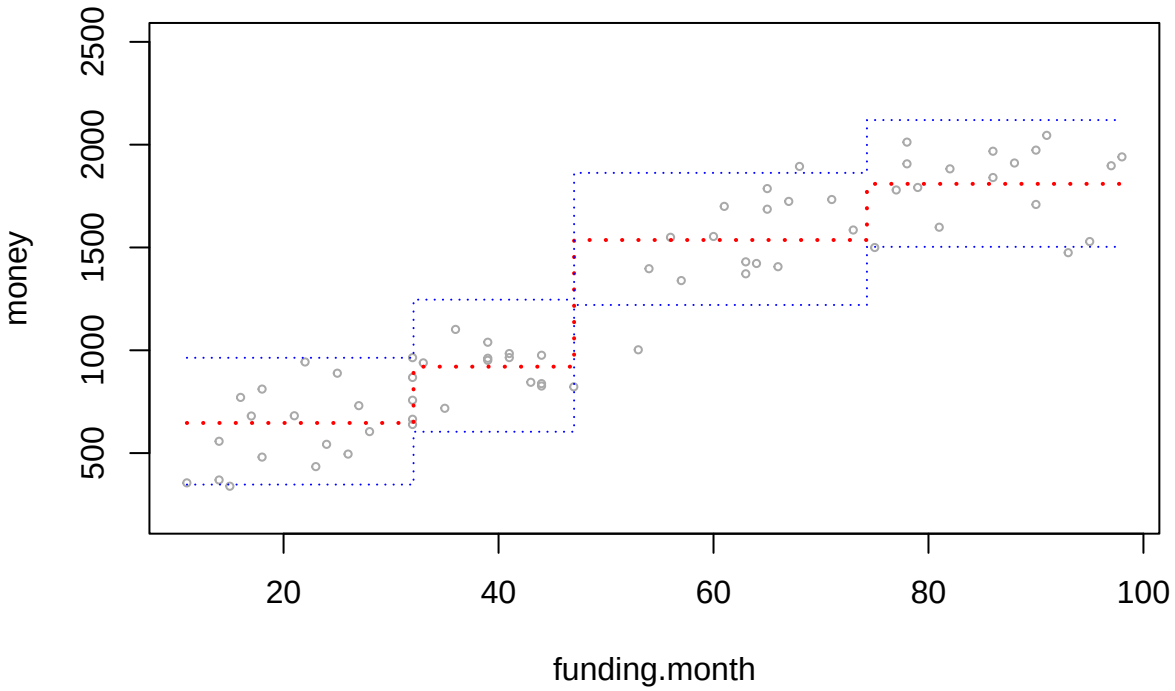
```
# all coefs are highly significant
```

4. Compute the prediction bands for the regression model of the previous exercise, using a full conformal approach and setting $\alpha = 0.1$ as the miscoverage level[2].

```
month_grid = seq(range(df_2$funding.month)[1],
                 range(df_2$funding.month)[2],
                 length.out = 100)

with(
  df_2,
  plot(
    funding.month ,
    money ,
    xlim = range(month_grid) ,
    cex = .5,
    col = " darkgrey "
  ,ylim=c(200,2500))
)
# lines(month_grid,preds$fit ,lwd =2, col =" blue")

lm_train = lm.funs(intercept = T)$train.fun
lm_predict = lm.funs(intercept = T)$predict.fun

design_matrix = model.matrix(m_cut)[,-1]
pred_cut = cut(month_grid,breaks=c(-Inf,funding.month_quart[c(-1,-5)],Inf))
pred_grid = model.matrix(lm(rep(1,length(pred_cut))~pred_cut))[,-1]

c_preds = conformal.pred(
  x = design_matrix,
  y = df_2$money,
  pred_grid,
  alpha = 0.1,
  verbose = F,
  train.fun = lm_train,
  predict.fun = lm_predict,
  num.grid.pts = 200
)

lines(
```

[2]Hint: you may need to use the `model.matrix` function twice to make it work. Or, alternatively, you may need to use a one hot encoding function.

```
  month_grid,
  c_preds$pred ,
  lwd = 2,
  col = "red",
  lty = 3,type="s"
)
matlines(
  month_grid ,
  cbind(c_preds$up, c_preds$lo) ,
  lwd = 1,
  col = " blue",
  lty = 3,type="s"
)
```



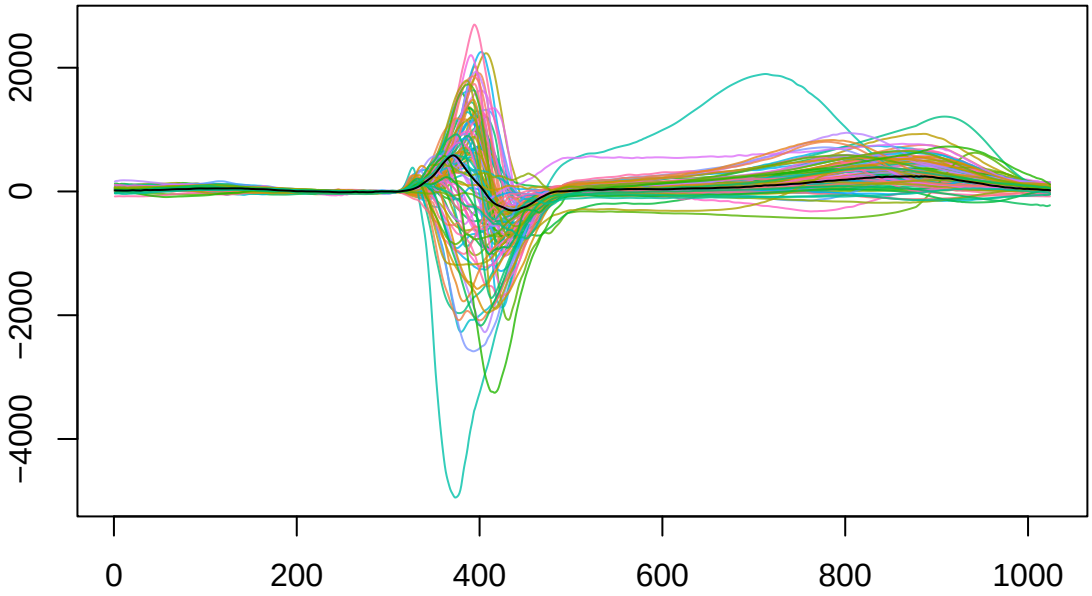
Exercise 3

The research proposal of Professor Franziska Iünz is confidential, yet she is willing to share some classified information with you should you help her provide some preliminary evidence. Professor Iünz aims at using functional data analysis (FDA) to uncover novel insights in healthcare analysis. To do so, she has collected 100 samples of ECG traces: 50 pertaining to healthy subjects, while the remaining ones are related to patients suffering from Left-Bundle-Branch-Block (LBBB), a cardiac pathology of interest. The data are contained in the `df_3.Rds` file. The `type` variable records the subject health status, while the remaining ones, denoted with `V1`,..., `V1024`, contains the values of the discretized signal over an evenly spaced grid of 1024 time points.

1. By suitably defining a functional data object, plot the resulting curves for the $N = 100$ subjects. Making no distinction in the subjects type, compute the sample median curve using the modified band depth and superimpose it to the previous plot.

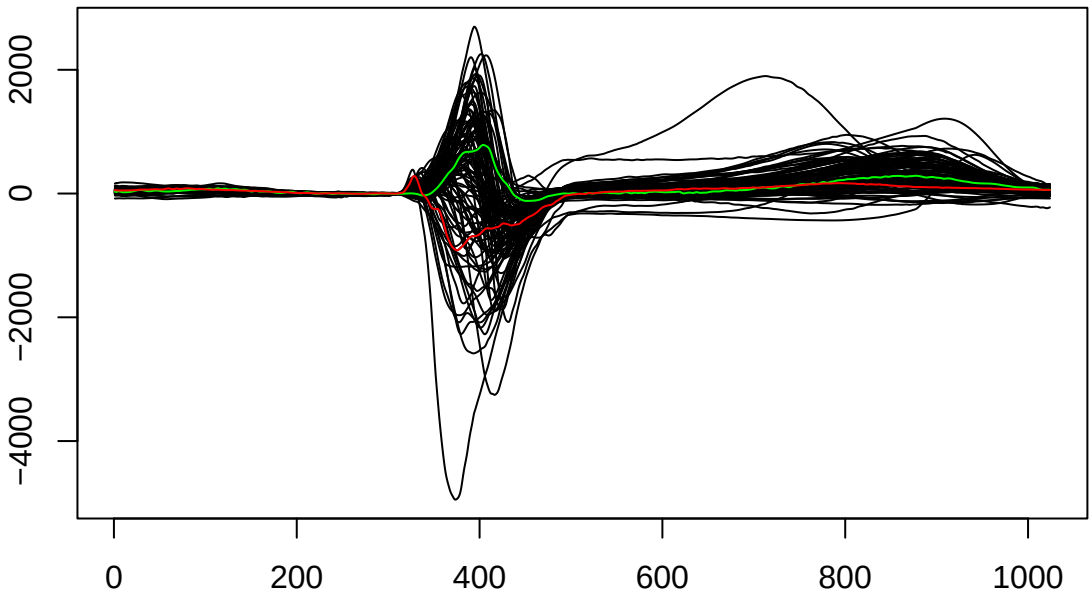
```
df_3 <- readRDS(here("2023-02-10/data/df_3.Rds"))
df_3_no_type <- df_3[,-1]
grid <- 1:ncol(df_3_no_type)
f_data <- fData(grid,df_3_no_type)
```

```
plot(f_data)

band_depth <- MBD(Data = f_data)
median_curve <- median_fData(fData = f_data, type = "MBD")
plot(f_data)
lines(grid,median_curve$values)
```



2. Using again the modified band depth, compute the sample median curves for the `healthy` and `LBBB` subgroups of subjects and provide a relevant plot of them.

```
df_3_split <- split(x = df_3_no_type, f = df_3$type)
healthy_fd=fData(grid,df_3_split$healthy)
LBBB_fd=fData(grid,df_3_split$LBBB)

MBD_healthy <- median_fData(healthy_fd,type='MBD')
MBD_LBBB <- median_fData(LBBB_fd,type='MBD')
df_4_plot <- append_fData(fD1 = f_data,fD2 = append_fData(MBD_healthy,MBD_LBBB))
plot(df_4_plot,col=c(rep("black",100),"green","red"))
```



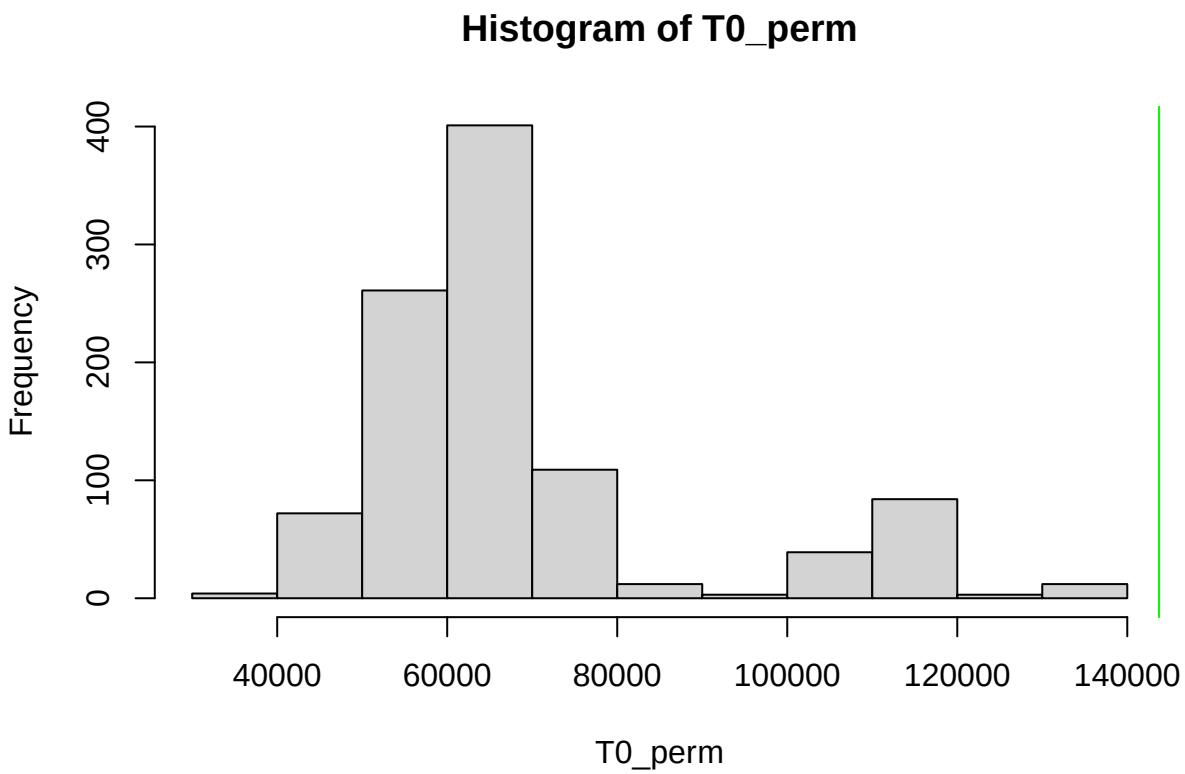
3. Test the equality of the theoretical median curves for the `healthy` and `LBBB` subgroups of subjects by performing a global two-sample permutation test using as a test statistics the L1 norm between the two sample MBD medians[3]. Present the empirical cumulative distribution function of the permutational test statistic as well as the p-value for the test.

```
median_diff <- MBD_healthy$values-MBD_LBBB$values
T0=(sum(abs(median_diff)))
N <- nrow(df_3)
B <- 1000
T0_perm <- numeric(B)
set.seed(2022)
pb=progress::progress_bar$new(total=B, format = " Processing [:bar] :percent eta: :eta")
for(perm in 1:B){
  permutazione <- sample(N)
  df_perm=f_data[permutazione,]
  perm_healthy = df_perm[1:50,]
  perm_LBBB = df_perm[(50+1):100,]
  median_diff_perm = median_fData(perm_healthy, type = 'MBD')$values -
    median_fData(perm_LBBB, type ='MBD')$values
  T0_perm[perm]=sum(abs(median_diff_perm))
  pb$tick()
}

sum(T0_perm >= T0)/B
```

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

```
hist(T0_perm)
abline(v=T0,col='green')
```

[3]Hint: given the `fData` class provided by the `roahd` package, this shall be rather natural




```
# The two medians are statistically different
```