Exam: First Session

Nonparametric Statistics, AY 2021/22

January 21, 2022

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/boostrap 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.
- 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

An Irish farmer, Andrew O'Cappor, owns $N = 382$ cows and he aims at becoming the best quality milk-maker in the whole Ireland. Knowing that he must rely on the most sophisticated analytical techniques to achieve his goal he collects $N = 382$ milk samples, one for each cow, measuring three milk quality traits, namely Milk pH, Casein Micelle Size (CMS), expressed in nm, and κ-casein (grams per liter). The resulting samples are contained in the `milk_samples_1.Rds` file. His first aim is to identify whether some cows in the herd produce anomalous milk, he therefore asks you to:

1. Provide the Tukey median of the milk samples

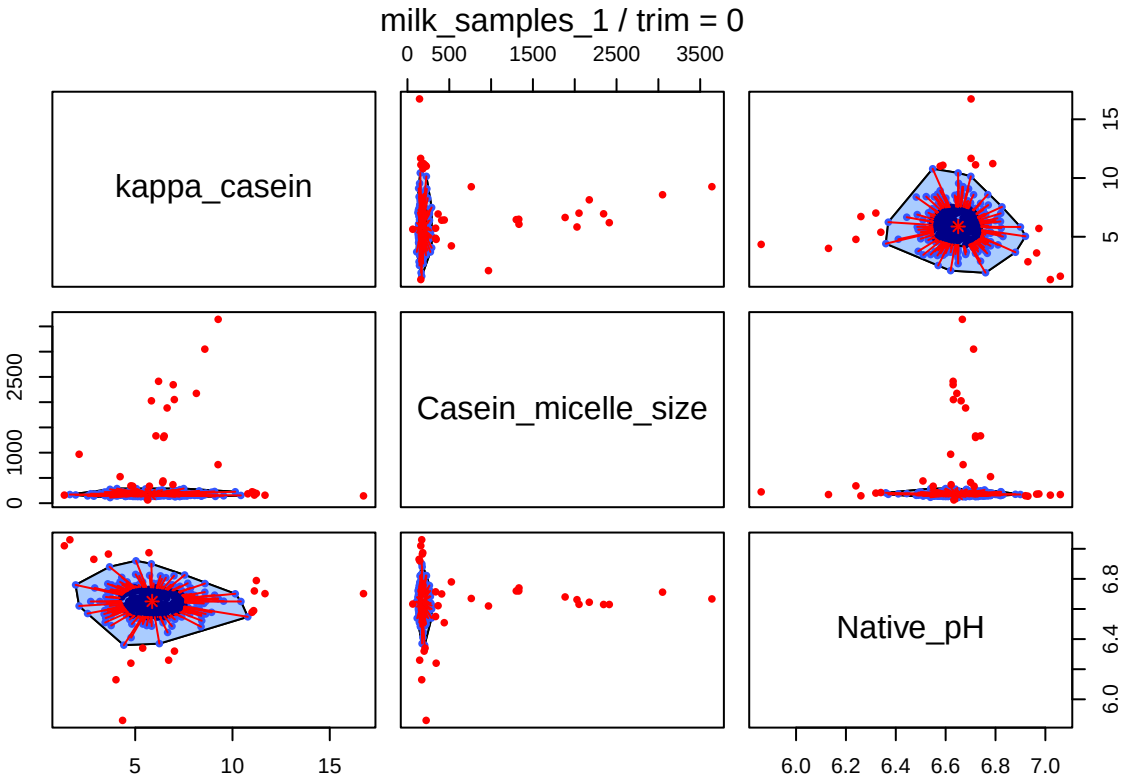
```
milk_samples_1 <- readRDS(here("2022-01-21/data/milk_samples_1.Rds"))
N <- nrow(milk_samples_1)
p <- ncol(milk_samples_1)
(tukey_median <- depthMedian(milk_samples_1,depth_params = list(method='Tukey')))
```

```
##        kappa_casein Casein_micelle_size           Native_pH
##             5.85842           169.30000             6.64000
```

2. Plot a bagplot matrix of the three collected variables, and to determine a vector of row indexes identifying the milk samples that are outliers according to any panel of the bagplot matrix[1]. Report the plot of the bagplot matrix and briefly describe the process that led to the generation of the aforementioned vector of row indexes.

```
bagplot_matrix <- aplpack::bagplot.pairs(milk_samples_1)
```

[1] Hint: you know how to determine outliers from a bagplot, a bagplot matrix is just a collection of bidimensional bagplots




```
bagplot_12 <- compute.bagplot(milk_samples_1[, 1], milk_samples_1[, 2])
bagplot_13 <- compute.bagplot(milk_samples_1[, 1], milk_samples_1[, 3])
bagplot_23 <- compute.bagplot(milk_samples_1[, 2], milk_samples_1[, 3])

ind_out_12 <-
  which(apply(milk_samples_1[, 1:2], 1, function(x)
    all(x %in% bagplot_12$pxy.outlier)))
ind_out_13 <-
  which(apply(milk_samples_1[, c(1, 3)], 1, function(x)
    all(x %in% bagplot_13$pxy.outlier)))
ind_out_23 <-
  which(apply(milk_samples_1[, 2:3], 1, function(x)
    all(x %in% bagplot_23$pxy.outlier)))
ind_out <- unique(c(ind_out_12, ind_out_13, ind_out_23))
milk_samples_1_out <- milk_samples_1[ind_out, ]
col_outlier <- factor(ifelse(1:N %in% ind_out, "yes", "no"))
ind_out
```

```
##  [1]   1  45  83  95 177 188 200 219 234 257 260 264 266 275 295 301 351 368 369
## [20] 373 374 375 376 377 378 379 380 381 382   2   6  20  37  49 144 185 281 303
## [39]  35 105 233
```

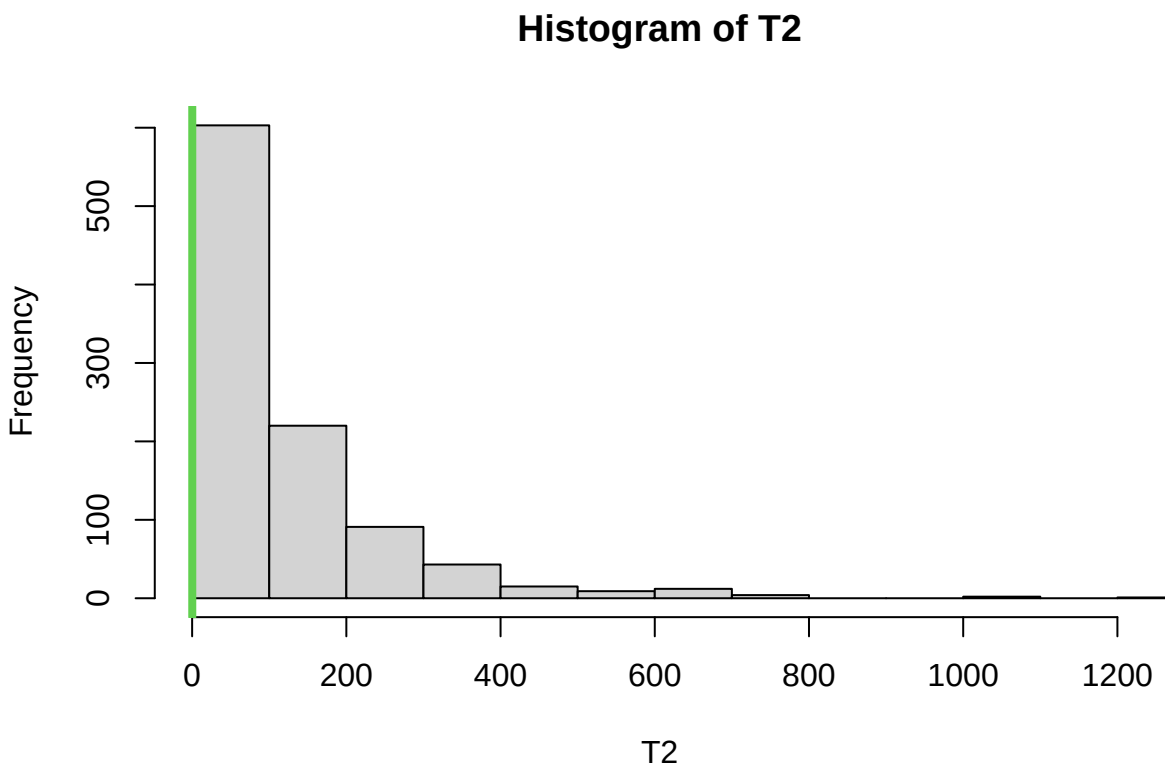
```
# pairs(milk_samples_1, col=col_outlier)
```

3. Test whether the 341 milk samples, obtained by discarding the 41[2] units that were flagged as outliers by the bagplots, comply with the gold standard in terms of milk quality, for which κ-casein must be equal to 6 grams per liter, CMS to 174 *nm* and Milk pH to 7. Perform a permutation test using

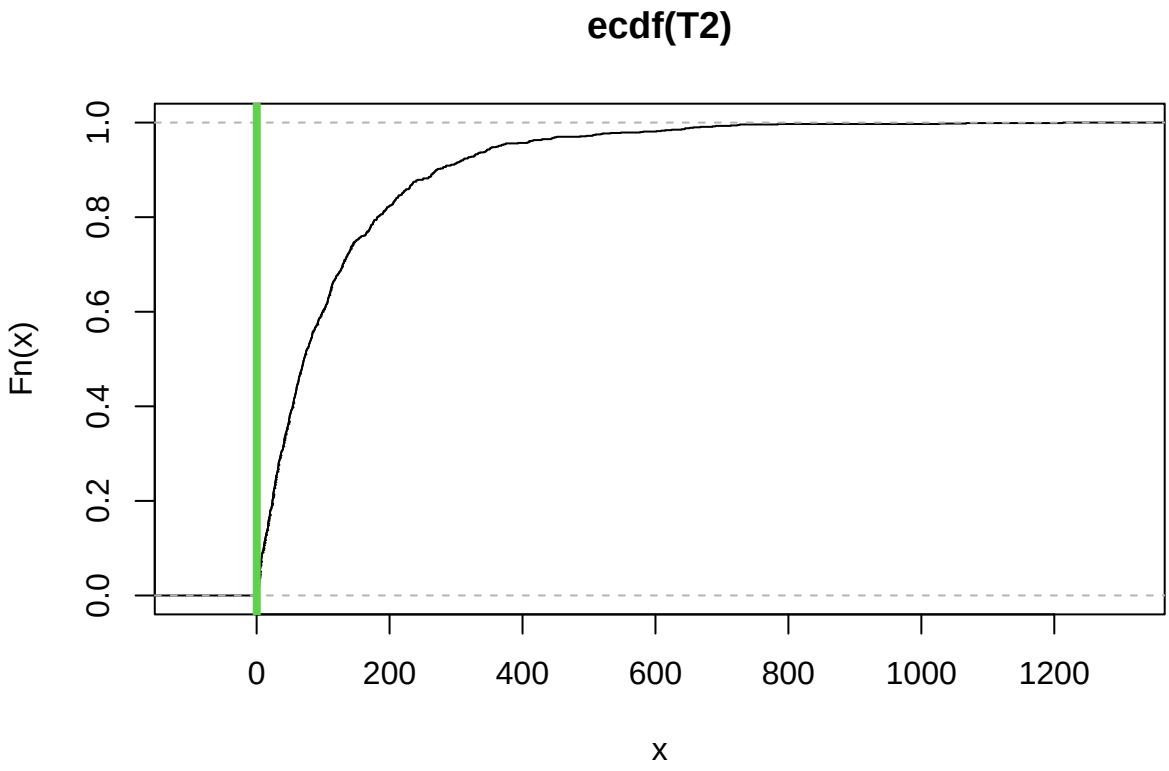
[2] if you did not manage to solve point 2., here you find the vector of row indexes identifying the outlying milk samples:
1 2 6 20 35 37 45 49 83 95 105 144 177 185 188 200 219 233 234 257 260 264 266 275 281 295 301 303 351 368 369 373 374 375 376 377 378 379 380 381 382

as test statistic the squared euclidean distance between the sample mean and the gold standard. Specify assumptions, the null and the alternative hypothesis you are testing, provide the histogram of the permuted distribution of the test statistic, its cumulative distribution function and the p-value, commenting the results.

```
milk_samples_1_extra <- milk_samples_1
milk_samples_1_extra$outlier_indicator <- col_outlier
milk_samples_1_no_out <- milk_samples_1[-ind_out,]
N_no_out <- nrow(milk_samples_1_no_out)
standard_cow <- c(6,174,7)

sample_mean <- colMeans(milk_samples_1_no_out)

#permutation test
# Computing a proper test statistic
# (i.e., L2 norm between the sample mean vector and the hypothesized center of symmetry)

T_20 <- norm(sample_mean-standard_cow,"2")^2


# Estimating the permutational distribution under H0
B <- 1000
T2 <- numeric(B)

set.seed(2022)

pb=progress::progress_bar$new(total=B, format = " Processing [:bar] :percent eta: :eta")

for(perm in 1:B) {
  # Permuted dataset
  signs.perm <- rbinom(N_no_out, 1, 0.5) * 2 - 1
  df_perm <-
    matrix(standard_cow,
           nrow = N_no_out,
           ncol = p,
           byrow = T) + (milk_samples_1_no_out - standard_cow) * matrix(signs.perm,
                                                 nrow = N_no_out,
                                                 ncol = p,
                                                 byrow = FALSE)

  x.mean_perm <- colMeans(df_perm)
  T2[perm]  <- norm(x.mean_perm - standard_cow, type = "2") ^ 2
  pb$tick()
}

# plotting the permutational distribution under H0
hist(T2,xlim=range(c(T2,T_20)))
abline(v=T_20,col=3,lwd=4)
```




```
plot(ecdf(T2))
abline(v=T_20,col=3,lwd=4)
```




```
# p-value
p_val <- sum(T2>=T_20)/B
p_val
```

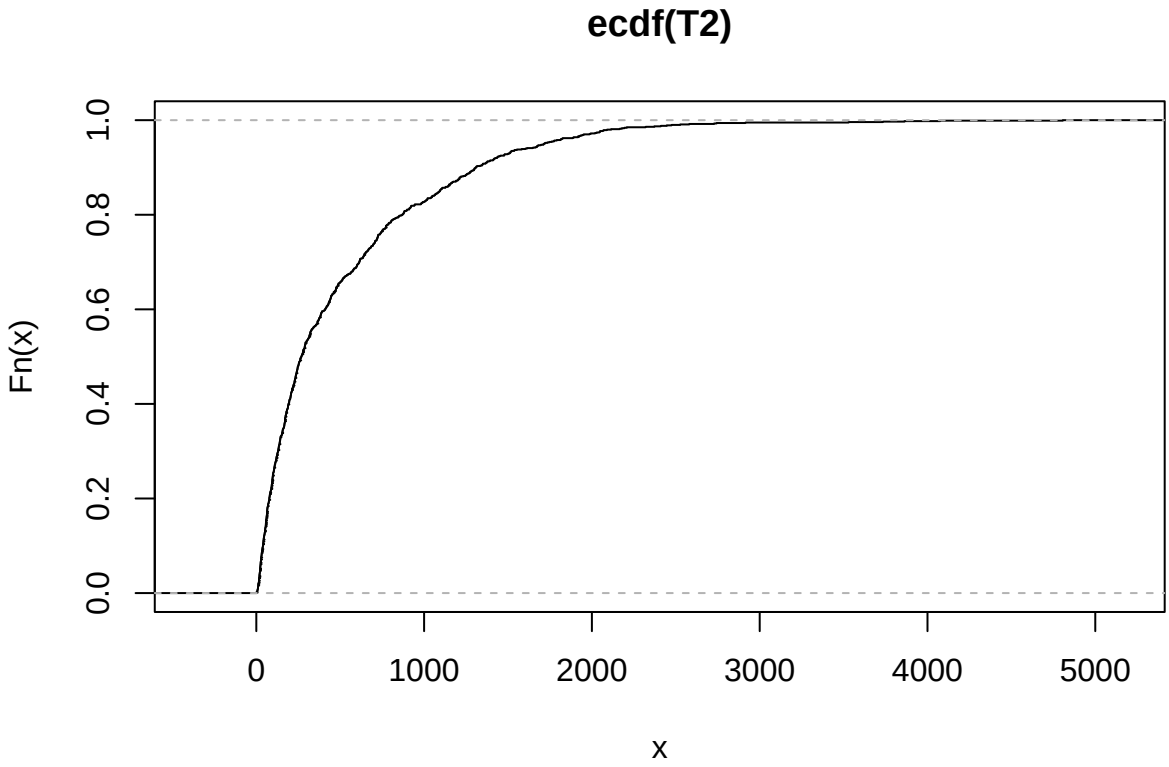
```
## [1] 0.997
```

4. Perform the exact same test but this time consider the original $N = 382$ milk samples. Does your conclusion change? If so, what kind of estimator would you propose that does not imply the a-priori removal of a portion of the data[3]?

```r
sample_mean <- colMeans(milk_samples_1)

#permutation test
# Computing a proper test statistic

T_20 <- norm(sample_mean-standard_cow,"2")^2


# Estimating the permutational distribution under H0
B <- 1000
T2 <- numeric(B)

set.seed(2022)

pb=progress::progress_bar$new(total=B, format = " Processing [:bar] :percent eta: :eta")

for(perm in 1:B) {
  # Permuted dataset
  signs.perm <- rbinom(N, 1, 0.5) * 2 - 1
  df_perm <-
    matrix(standard_cow,
           nrow = N,
           ncol = p,
           byrow = T) + (milk_samples_1 - standard_cow) * matrix(signs.perm,
                                                 nrow = N,
                                                 ncol = p,
                                                 byrow = FALSE)

  x.mean_perm <- colMeans(df_perm)
  T2[perm]  <- norm(x.mean_perm - standard_cow, type = "2") ^ 2
  pb$tick()
}

# plotting the permutational distribution under H0
hist(T2,xlim=range(c(T2,T_20)))
abline(v=T_20,col=3,lwd=4)
```

[3] No calculation required


```
plot(ecdf(T2))
```




```
# p-value
p_val <- sum(T2>=T_20)/B
p_val
```

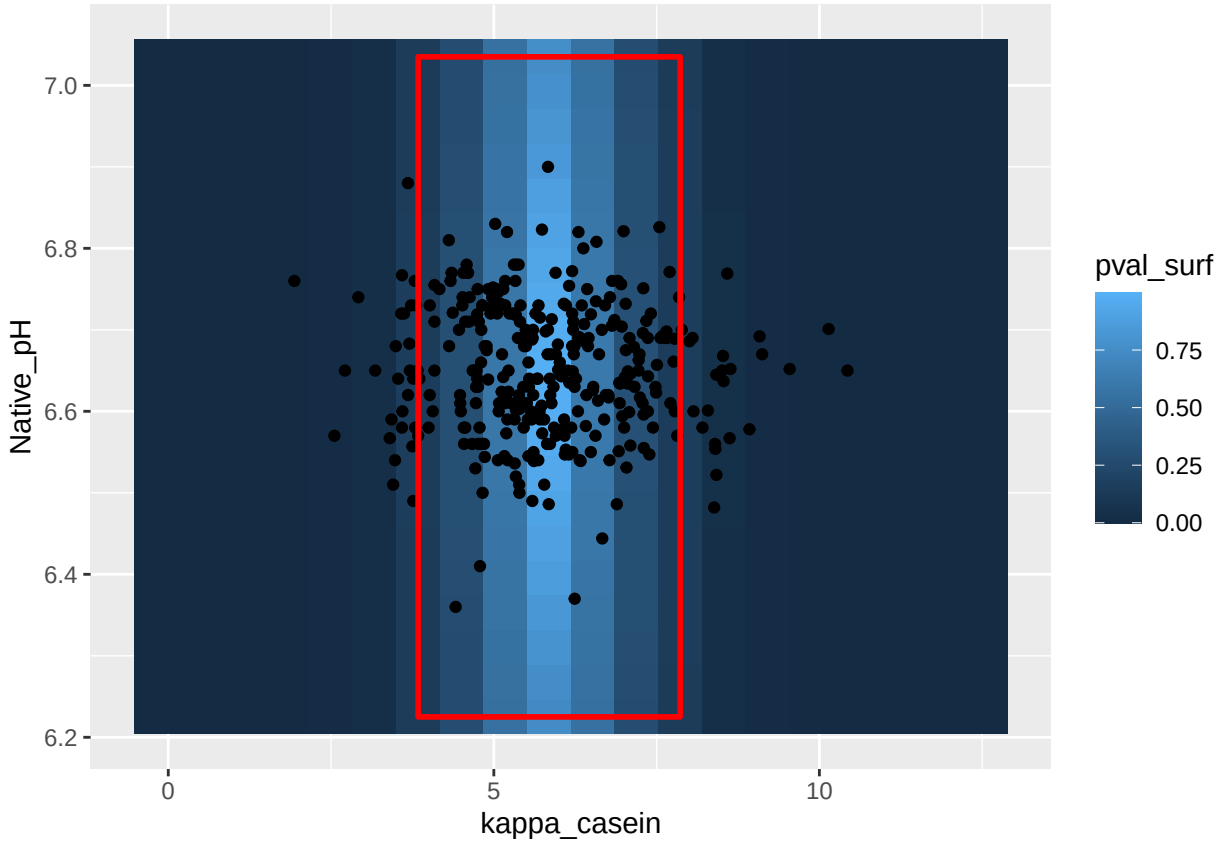
```
## [1] 0.001
```

```
# Possible solution:
# Use a different test statistic considering testing, for example,
# the equality of the Tukey median to the gold standard
```

4. Provide a Full Conformal $1 - \alpha = 90\%$ prediction region for the κ-casein and Milk pH of a new milk sample, using the euclidean distance between the new data point and the sample Tukey 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 = milk_samples_1_no_out[, -2]
n_grid = 20
grid_factor = 0.25
alpha = .1
n = nrow(data_predict)

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 = 'Tukey'))
  depth_surface_vec = rowSums(t(t(newdata) - newmedian) ^ 2) #In this case I am using the L^2 norm...
  sum(depth_surface_vec[-1] >= depth_surface_vec[1]) / (n + 1)
}


pval_surf = 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_tile(data = data_plot, aes(kappa_casein, Native_pH, fill = pval_surf)) +
  geom_point(data = data.frame(data_predict), aes(kappa_casein, Native_pH)) +
  geom_polygon(
    data = poly_points,
    aes(kappa_casein, Native_pH),
```

```
    color = 'red',
    size = 1,
    alpha = 0.01
  )
```



Exercise 2

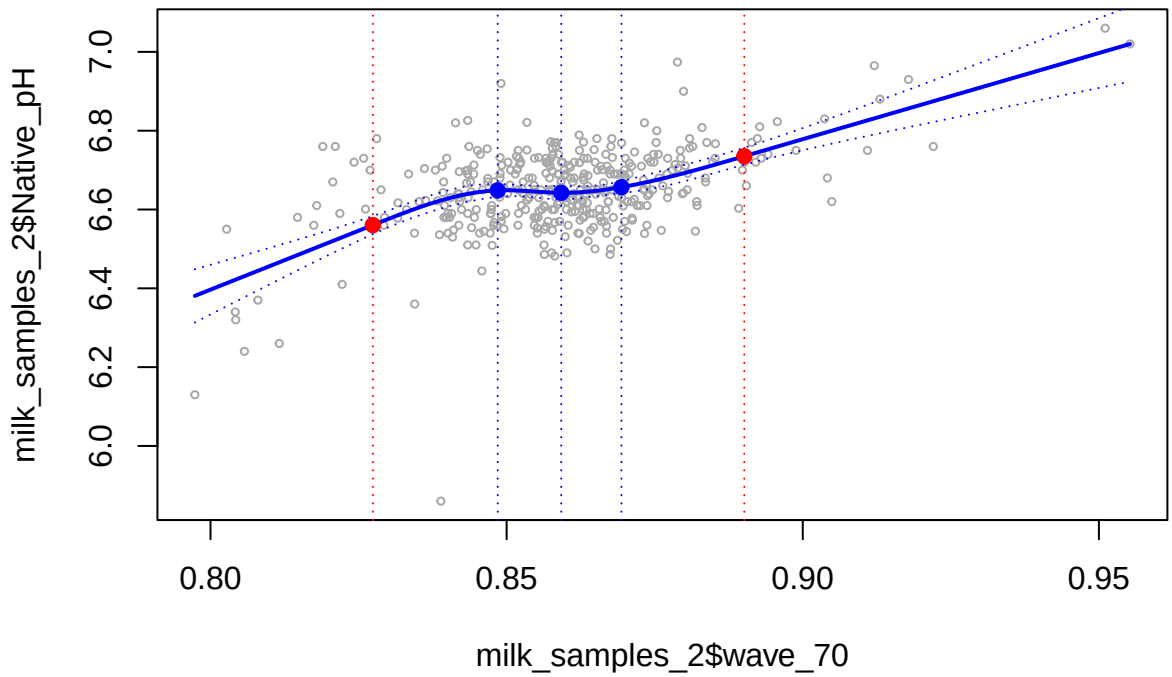
A friend of Andrew O'Cappor, Dr. Alexander House, statistician and chemometrician, advises the farmer that spectroscopy is the state-of-the-art technology to employ when it comes to evaluate milk quality. Motivated by this, and advised by Dr House, Andrew O'Cappor is interested in building a nonparametric model to predict Milk pH by means of the absorbance values at wavenumbers 70 and 300 cm^{-1}, contained in the `milk_samples_2.Rds` file. He therefore asks you to:

1. Build a natural cubic spline model to regress Milk pH on the milk absorbance at wavenumber 70 cm^{-1}. Set knots at the three quartiles of the explanatory variable and boundary knots at the 5th and 95th percentiles. Provide a plot of the regression line with standard errors for the prediction, a table summarizing the coefficients and comment the results.

```
milk_samples_2 <- readRDS(here("2022-01-21/data/milk_samples_2.Rds"))

knots <- quantile(milk_samples_2$wave_70,probs = c(.25,.5,.75))
boundary_knots <- quantile(milk_samples_2$wave_70,probs = c(.05,.95))
fit_spline <- lm(Native_pH~ns(wave_70,knots = knots,Boundary.knots = boundary_knots),
                 data = milk_samples_2)

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

term	estimate	std.error	statistic	p.value
(Intercept)	6.5605245	0.0114798	571.486104	0.00e+00
ns(wave_70, knots = knots, Boundary.knots = boundary_knots)1	0.0750578	0.0187168	4.010177	7.32e-05
ns(wave_70, knots = knots, Boundary.knots = boundary_knots)2	0.0625313	0.0158273	3.950853	9.30e-05
ns(wave_70, knots = knots, Boundary.knots = boundary_knots)3	0.2365391	0.0243784	9.702806	0.00e+00
ns(wave_70, knots = knots, Boundary.knots = boundary_knots)4	0.1196908	0.0144234	8.298355	0.00e+00

```
new_data_seq <-
  seq(min(milk_samples_2$wave_70),
      max(milk_samples_2$wave_70),
      length.out = 100)

preds=predict(fit_spline, newdata = list(wave_70=new_data_seq),se=T)
se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)

plot(y=milk_samples_2$Native_pH,x=milk_samples_2$wave_70  ,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,list(wave_70=knots))
points(knots,knots_pred, col='blue',pch=19)
boundary_pred <- predict(fit_spline,list(wave_70=boundary_knots))
points(boundary_knots,boundary_pred,col='red',pch=19)
abline(v = knots, lty=3, col="blue")
abline(v = boundary_knots, lty=3, col="red")
```



1. Build an additive model for regressing Milk pH on the milk absorbance at wavenumbers $70cm^{-1}$ and

$300cm^{-1}$ and their interaction, using penalized cubic b-spline terms for smoothing. After having written in proper mathematical terms the additive model you have estimated, report the adjusted R2 and the p-values of the tests.

$$pH_i = \beta_0 + f(\text{ wave }_{300_i}) + f(\text{ wave }_{70_i}) + f(\text{ wave }_{70_i} * \text{ wave }_{300_i}) + \epsilon_i, i = 1, \ldots, N$$

```
fit_gam <- gam(Native_pH~s(wave_70,bs = "cr")+
                 s(wave_300,bs = "cr")+
                 s(I(wave_300*wave_70),bs = "cr"),data = milk_samples_2)
table_fit_gam <- summary(fit_gam)
(r_2_squared <- table_fit_gam$r.sq)
```

```
## [1] 0.3478556
```

```
table_fit_gam$p.pv
```

```
## (Intercept)
##           0
```

```
table_fit_gam$s.table
```

```
##                                edf   Ref.df        F      p-value
## s(wave_70)                4.452492 5.348768 4.772843 0.0002977267
## s(wave_300)               2.696870 3.493472 2.113757 0.0803365804
## s(I(wave_300 * wave_70)) 3.061677 3.791753 2.088150 0.1105661789
```

2. Build an additive model for regressing Milk pH on the milk absorbance at wavenumbers $70cm^{-1}$ and $300cm^{-1}$ with no interaction, using again cubic b-spline terms. Assuming the residuals to be normal, use the Anova F test statistic to assess the significance of the interaction term, specifying the null and the alternative hypthotesis you are testing and report the resulting p-value. Comment on the results.

```
fit_gam_reduced <-
  gam(Native_pH ~ s(wave_70, bs = "cr") + s(wave_300, bs = "cr"), data = milk_samples_2)

anova(fit_gam_reduced, fit_gam, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: Native_pH ~ s(wave_70, bs = "cr") + s(wave_300, bs = "cr")
## Model 2: Native_pH ~ s(wave_70, bs = "cr") + s(wave_300, bs = "cr") +
##     s(I(wave_300 * wave_70), bs = "cr")
##   Resid. Df Resid. Dev     Df Deviance      F  Pr(>F)
## 1    371.08     2.9385
## 2    368.37     2.8638 2.7137 0.074648 3.5616 0.01767 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Full model is preferred
```

3. Build a semi-parametric model for regressing Milk pH on the milk absorbance at wavenumbers $70cm^{-1}$ and $300cm^{-1}$, assuming the effect of wavenumber $300cm^{-1}$ to be linear on the response and no interaction. Using a bootstrap approach, provide a reverse percentile confidence interval for the parametric effect.

```
fit_semi <- gam(Native_pH~s(wave_70,bs = "cr")+wave_300,data = milk_samples_2)

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

```
wave300_obs = summary(fit_semi)$p.table[2,1]
T.boot.wave300 = numeric(B)

set.seed(2022)

for(b in 1:B) {


  Native_pH_boot <- fitted.obs + sample(res.obs, replace = T)
  fit_semi_boot <-
    gam(Native_pH_boot ~ s(milk_samples_2$wave_70, bs = "cr") +
          milk_samples_2$wave_300)

  fit_semi_boot_table = summary(fit_semi_boot)

  T.boot.wave300[b] = fit_semi_boot_table$p.table[2,1]

}

alpha <- 0.05
right.quantile.wave300 <- quantile(T.boot.wave300, 1 - alpha/2)
left.quantile.wave300  <- quantile(T.boot.wave300, alpha/2)

CI.RP.wave300 <-
  c(
    wave300_obs - (right.quantile.wave300 - wave300_obs),
    wave300_obs - (left.quantile.wave300- wave300_obs))
names(CI.RP.wave300)=c('lwr','upr')
CI.RP.wave300
```

```
##       lwr       upr
## 0.6153986 3.0572096
```

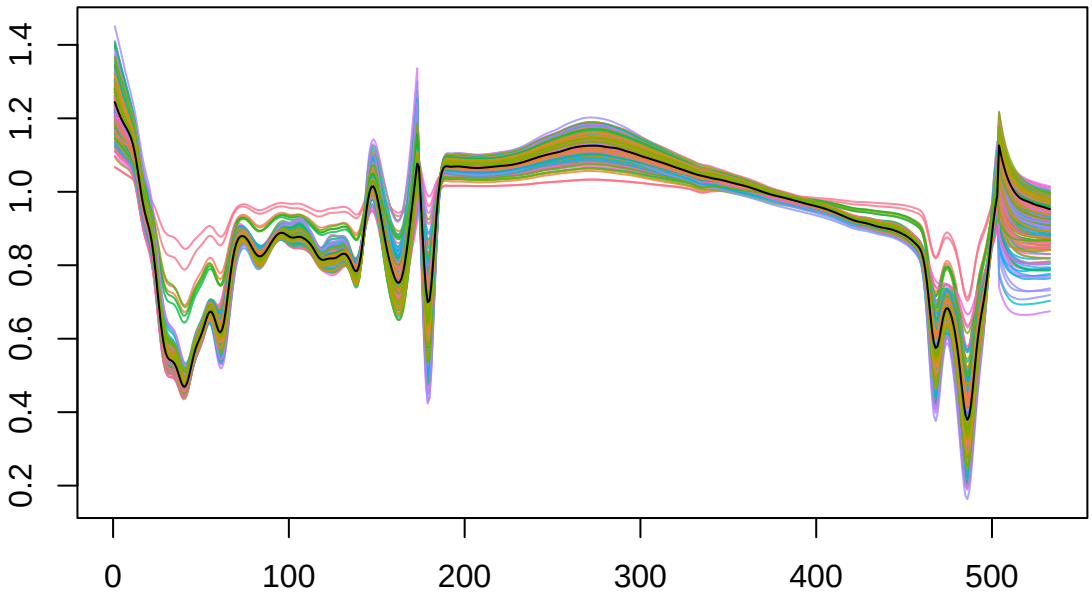
Exercise 3

Andrew O'Cappor has recently become passionate about functional data analysis (FDA): Dr. Alexander House is then ready to unleash the power of the MilkoScan FT6000 milk analyzer, producing MIRS transmittance spectra in the entire mid-infrared light region (wavenumbers 70 and 300 cm^{-1} of the previous exercise were just two points of this region). The MIRS for the $N = 382$ milk samples are contained in the `milk_samples_3.Rds` file. Andrew O'Cappor would like to exploit these modern statistical methods to further analyze his milk samples. To this extent, he requires you to:

1. By suitably defining a functional data object, plot the resulting spectra for the $N = 382$ milk samples. Then, compute the median spectrum using the modified band depth and superimpose it to the previous plot.

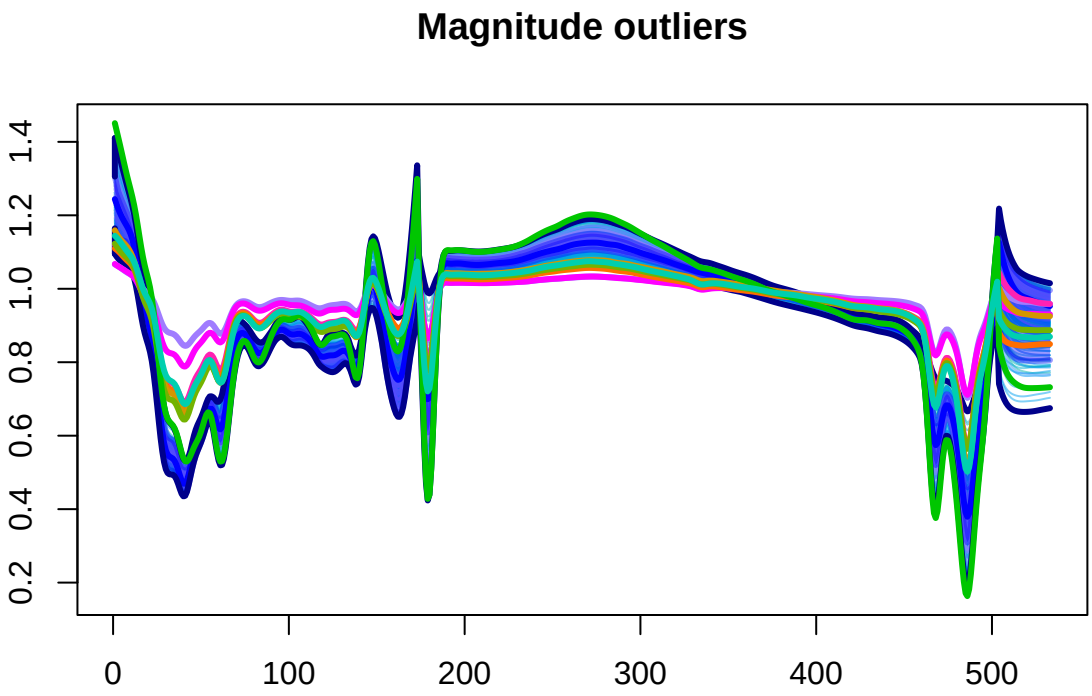
```
milk_samples_3 <- readRDS(here("2022-01-21/data/milk_samples_3.Rds"))
grid <- 1:ncol(milk_samples_3)
f_data <- fData(grid,milk_samples_3)
plot(f_data)

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

```
lines(grid,median_curve$values)
```



2. Produce a functional boxplot and use it to identify the outlying curves. Are there amplitude outliers? Dr. House had previously mentioned that mid-infrared spectra directly reflects milk quality traits, such as Milk pH, Casein Micelle Size (CMS) and κ-casein. Do you agree with this statement[4]? Motivate your answer.

```
fb_plot <- fbplot(f_data, main="Magnitude outliers")
```




```
FDA_out <- c(fb_plot$ID_outliers)
FDA_out
```

```
## [1]   1   2   3   6  20  21 144 382
```

[4]Hint: try to cross-check which milk samples are flagged to be outliers from this procedure and from that of Exercise 1

```
mean(FDA_out %in% ind_out)
```

```
## [1] 0.75
```

```
# 6 out of 8 samples showcasing outlying spectra also possess outlying Milk pH,
# Casein Micelle Size (CMS)
# and $\kappa$-casein as per ind_out
```

3. Build a split conformal prediction band for a new milk spectrum using the previously computed functional median as pointwise predictor, the sup norm of absolute value of the functional residuals from the functional median as a NCM and the identity function as the modulation function

```
alpha=.1
n=nrow(milk_samples_3)
set.seed(2022)
i1=sample(1:n,n/2)
t_set=milk_samples_3[i1,]
c_set=milk_samples_3[-i1,]

mu <- median_fData(fData = fData(grid,t_set), type = "MBD")
mu=mu$values
res=c_set-t(mu)

ncm=apply(res,2,max)
ncm_sort=c(sort(ncm),Inf)
d=ncm_sort[ceiling((n/2 + 1)*(1-alpha))]

matplot(cbind(t(mu),t(mu)+d,t(mu)-d),type='l')
```