Exam: Second Session

Nonparametric Statistics, AY 2021/22

February 11, 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, Matthew O'Fountain, owns $N = 382$ cows and he is the best milk-maker in Ireland. Nevertheless, Matthew O'Fountain is still not satisfied with this result, and he aims at becoming the best milk-maker in the whole world. In order to do so he must expand his market presence in other countries, and he would like to know how to preserve milk quality once it is shipped from Ireland. Specifically, he is testing which type of pasteurization (`Pasteurized` or `Ultra-Pasteurized`) ensures longer shelf life. He thus conducts the following experiment: he collects $N = 382$ milk samples, one for each cow, and he monitors for $T = 100$ days whether the milk gets spoiled or not. The variable `time` indicates at which day the sample quality deteriorated (`spoiled=2`) or if it was still fresh after 100 days (`spoiled=1`). Together with the pasteurization type, he monitors 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.

1. First off, Matthew O'Fountain is interested in knowing whether the type of pasteurization alters the milk quality traits (Milk pH, Casein Micelle Size and κ-casein). By employing a permutation test on the standardized data[1], and using as test statistic the maximum absolute difference between the sample multivariate Tukey medians of the pasteurization types, check whether the milk samples differ in median in the two groups[2]. Plot the permutational cumulative distribution function of the test statistic, report the p-value of the test and comment it.

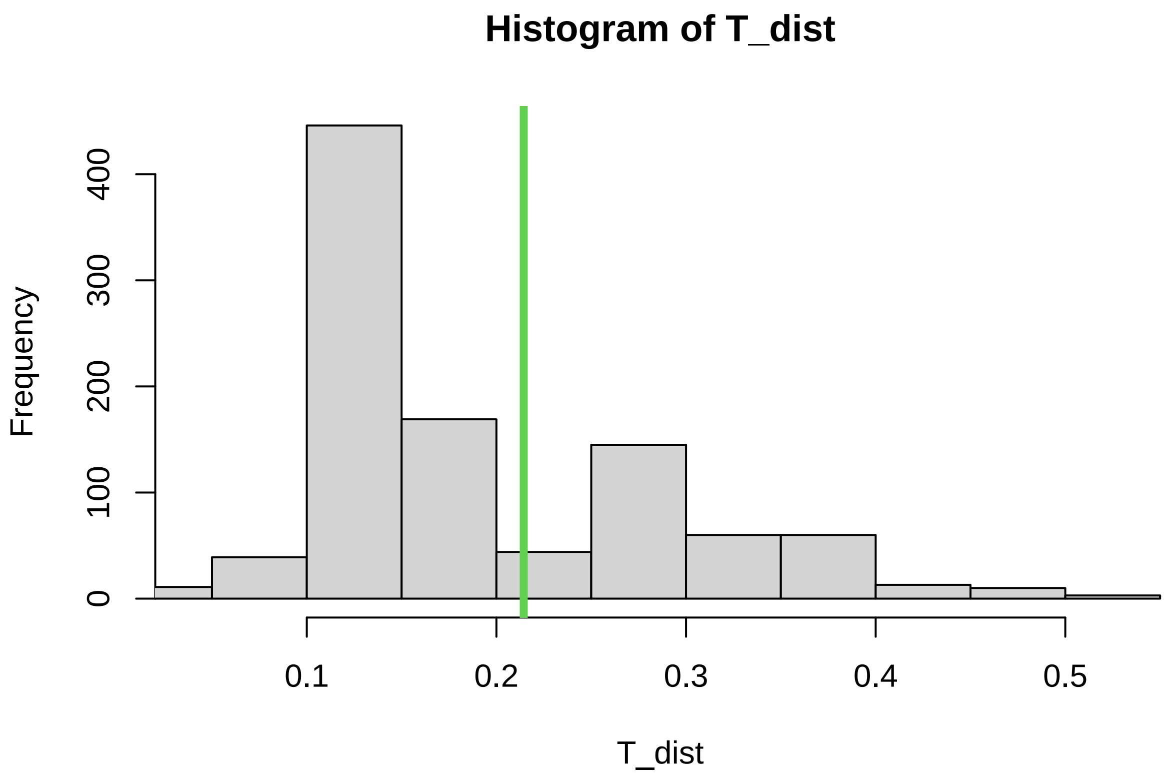
```r
milk_samples_1 <- readRDS(here("2022-02-11/data/milk_samples_1.Rds"))

milk_traits = scale(milk_samples_1[, 1:3])
n <- nrow(milk_traits)
table(milk_samples_1$pasteurization_type)
```

```
##
## Ultra-Pasteurized       Pasteurized
##               135               247
```

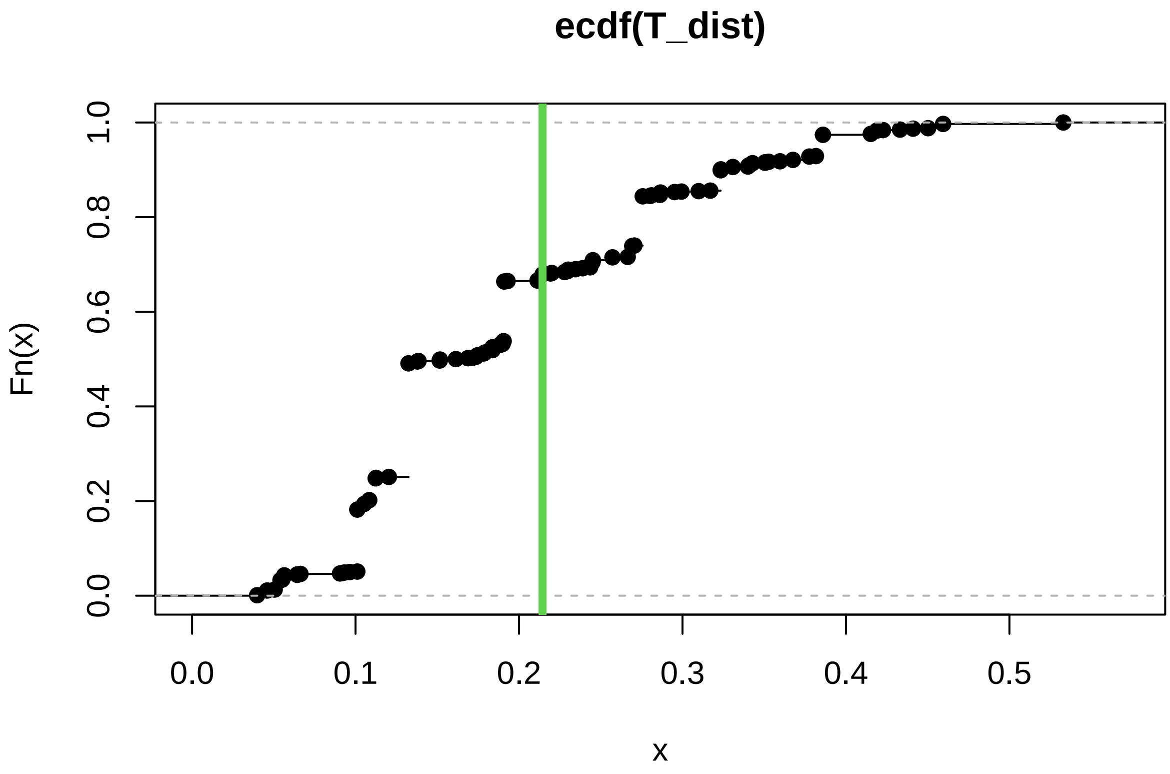
[1] Use the `scale` function

[2] It is going to take a while, use a progress bar if you are anxious.

```r
n1 = table(milk_samples_1$pasteurization_type)[1]
n2 = table(milk_samples_1$pasteurization_type)[2]

groups <- list()
groups$Pasteurized <- milk_traits[milk_samples_1$pasteurization_type=="Pasteurized",]
groups$`Ultra-Pasteurized` <- milk_traits[milk_samples_1$pasteurization_type=="Ultra-Pasteurized",]

median_pasturized = depthMedian(groups$Pasteurized, depth_params = list(method =
                                                                          'Tukey'))
median_ultra_pasturized = depthMedian(groups$`Ultra-Pasteurized`,
                                     depth_params = list(method = 'Tukey'))

t_stat = max(abs(median_ultra_pasturized - median_pasturized))

B = 1000
T_dist = numeric(B)
set.seed(2022)
pb = progress::progress_bar$new(total = B,
                                format = " Processing [:bar] :percent eta: :eta")
for (index in 1:B) {
  perm = sample(1:n)
  milk_traits.p = milk_traits[perm, ]
  median_pasturized.p = depthMedian(milk_traits.p[1:n1, ],
                                    depth_params = list(method ='Tukey'))
  median_ultra_pasturized.p = depthMedian(milk_traits.p[(n1 + 1):n, ],
                                         depth_params = list(method ='Tukey'))
  T_dist[index] = max(abs(median_ultra_pasturized.p - median_pasturized.p))
  pb$tick()
}


hist(T_dist, xlim = range(c(T_dist, t_stat)))
abline(v = t_stat, col = 3, lwd = 4)
```



```
plot(ecdf(T_dist))
abline(v = t_stat, col = 3, lwd = 4)
```



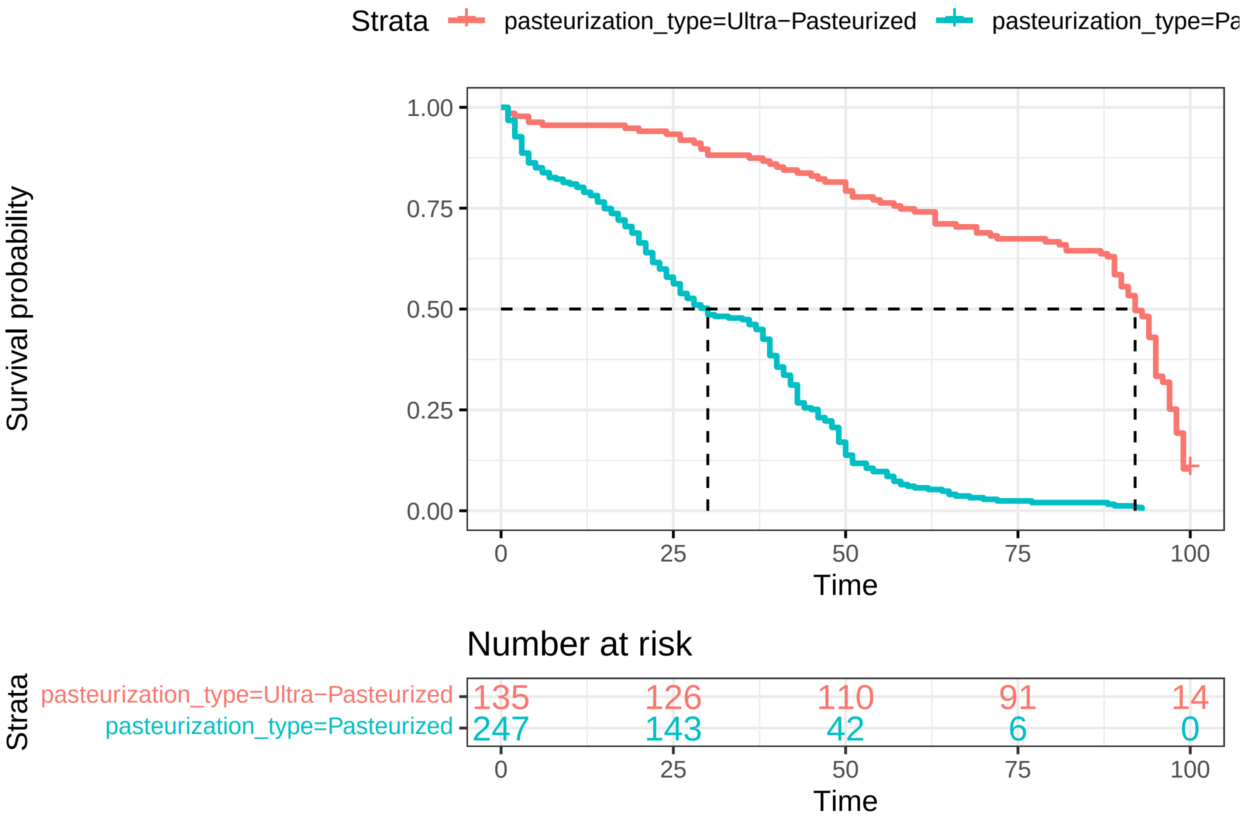
```
p_val <- sum(T_dist >= t_stat) / B
p_val
```

```
## [1] 0.334
```

```
# Type of pasteurization does not alter the quality
```

2. Compute the Kaplan-Meier estimation of the survival curves for the two pasteurization types and plot it. Report median survival times and test if the time-to-event distributions of the two behavioral groups are equal via a Log-rank test. Report the p-value and comment the result.

```
fit <-
  survfit(Surv(time, spoiled == 2) ~ pasteurization_type, data = milk_samples_1)
ggsurvplot(
  fit,
  risk.table = TRUE,
  # Add risk table
  risk.table.col = "strata",
  # Change risk table color by groups
  surv.median.line = "hv",
  # Specify median survival
  ggtheme = theme_bw(),
  # Change ggplot2 theme
)
```



```
surv_median(fit)
```

```
##                                   strata median lower upper
## 1 pasteurization_type=Ultra-Pasteurized     92    90    95
## 2       pasteurization_type=Pasteurized     30    26    38
```

```
log_rank_test <-
  survdiff(Surv(time, spoiled == 2) ~ pasteurization_type,
           data = milk_samples_1, )
```

```
log_rank_test
```

```
## Call:
## survdiff(formula = Surv(time, spoiled == 2) ~ pasteurization_type,
##     data = milk_samples_1)
##
##                                            N Observed Expected (O-E)^2/E (O-E)^2/V
## pasteurization_type=Ultra-Pasteurized 135      121      237      56.8       214
## pasteurization_type=Pasteurized       247      247      131     102.8       214
##
##  Chisq= 214  on 1 degrees of freedom, p= <2e-16
```

```
# Kaplan-Meier curves are statistically different
```

3. Fit a suitable Cox model for long-term survival as a function of all the available covariates. Interpret the estimated coefficients for covariates Milk pH and pasteurization type, including a comment on statistical significance

```
fit_cox <- coxph(Surv(time, spoiled) ~ ., data = milk_samples_1)
summary(fit_cox)
```

```
## Call:
## coxph(formula = Surv(time, spoiled) ~ ., data = milk_samples_1)
##
##   n= 382, number of events= 368
##
##                                     coef exp(coef)  se(coef)      z Pr(>|z|)
## kappa_casein                   0.0521715 1.0535564 0.0323659  1.612 0.106977
## Casein_micelle_size            0.0004595 1.0004596 0.0001387  3.313 0.000924
## Native_pH                      0.0182490 1.0184166 0.4612448  0.040 0.968440
## pasteurization_typePasteurized 2.1616326 8.6853054 0.1588610 13.607  < 2e-16
##
## kappa_casein
## Casein_micelle_size            ***
## Native_pH
## pasteurization_typePasteurized ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                exp(coef) exp(-coef) lower .95 upper .95
## kappa_casein                       1.054     0.9492    0.9888     1.123
## Casein_micelle_size                1.000     0.9995    1.0002     1.001
## Native_pH                          1.018     0.9819    0.4124     2.515
## pasteurization_typePasteurized     8.685     0.1151    6.3615    11.858
##
## Concordance= 0.706  (se = 0.014 )
## Likelihood ratio test= 240.8  on 4 df,   p=<2e-16
## Wald test            = 191.1  on 4 df,   p=<2e-16
## Score (logrank) test = 225.8  on 4 df,   p=<2e-16
```

```
# Milk pH is not significant
# the HR for pasteurization_typePasteurized is exp(coef) = 8.685.
# Holding the other covariates constant, using a Pasteurized pasteurization_type
# increases the hazard by a factor of 8.67.
```

4. Using the previously estimated Cox model, provide an estimate of the median survival time, under both

pasteurization types, for the gold standard sample 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.

```
standard_cow <- c(6,174,7)

new_df <-
  data.frame(
    "kappa_casein"   =   standard_cow[1],
    "Casein_micelle_size" = standard_cow[2],
    "Native_pH" = standard_cow[3],
    pasteurization_type=c("Pasteurized","Ultra-Pasteurized")
  )

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

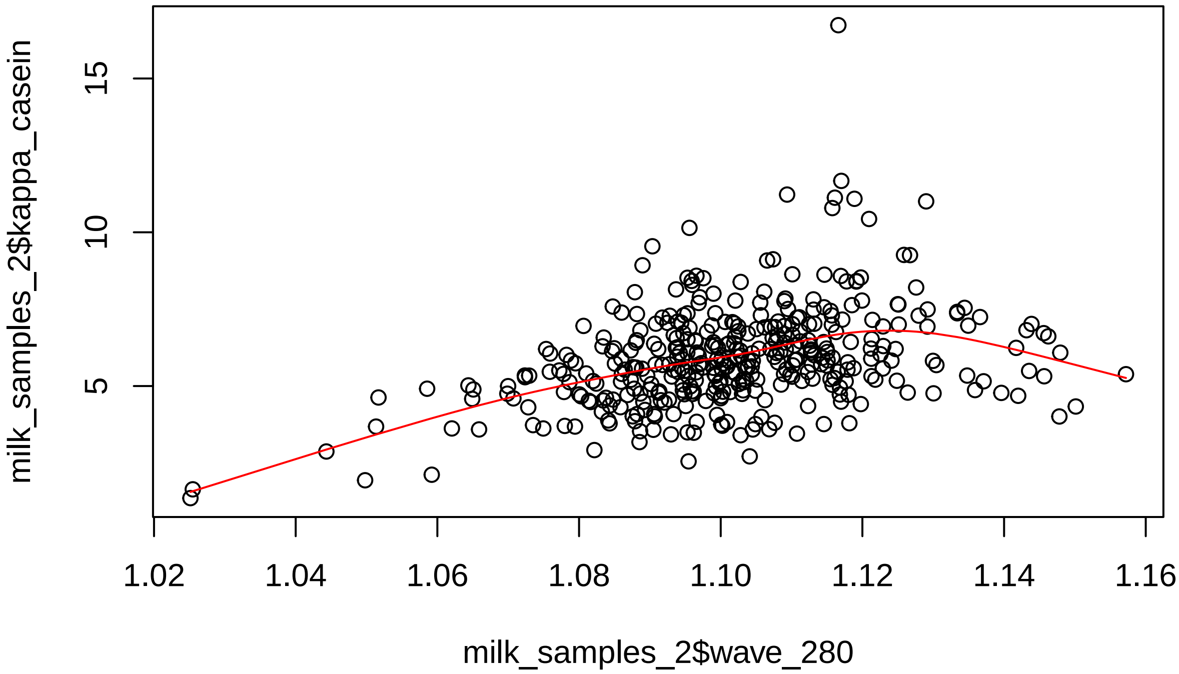
```
## Call: survfit(formula = fit_cox, newdata = new_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 382    368     30      24      41
## 2 382    368     93      89      96
```

Exercise 2

Matthew O'Fountain knows that spectroscopy is the state-of-the-art technology to employ when it comes to evaluate milk quality. Motivated by this he is interested in building a nonparametric model to predict κ-casein by means of the absorbance values at wavenumbers 280 and 700 cm^{-1}, contained in the `milk_samples_2.Rds` file.

1. Build a smoothing spline model to regress κ-casein on the milk absorbance at wavenumber 280 cm^{-1}, selecting λ by means of Generalized Cross Validation. Provide a plot of the regression line and a point-wise estimate for κ-casein when the milk absorbance at wavenumber 280 cm^{-1} is equal to 1.05. By using a bootstrap approach on the residuals, calculate the bias variance and MSE of such prediction (fix the λ value to the one obtained via Generalized Cross Validation).

```
milk_samples_2 <- readRDS(here("2022-02-11/data/milk_samples_2.Rds"))
N <- nrow(milk_samples_2)
plot(milk_samples_2$wave_280, milk_samples_2$kappa_casein)
fit_smooth <-
  smooth.spline(x = milk_samples_2$wave_280,
                y = milk_samples_2$kappa_casein,
                cv = FALSE)
plot(milk_samples_2$wave_280, milk_samples_2$kappa_casein)
lines(fit_smooth, col="red")
```



```
pred <- predict(fit_smooth,x=1.05)
y_hat <- pred$y
y_hat
```

```
## [1] 3.331632
```

```
fitted=predict(fit_smooth,milk_samples_2$wave_280)$y

residuals=milk_samples_2$kappa_casein-fitted

l_GCV <- fit_smooth$lambda
l_GCV
```

```
## [1] 0.006607909
```

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

for(sample in 1:B){

  kappa_casein_boot=fitted+sample(residuals,N,replace=T)
  new_model = smooth.spline(x = milk_samples_2$wave_280,
                           y = kappa_casein_boot,
                           lambda = l_GCV)
  boot_d[sample]=predict(new_model, 1.05)$y
}


(variance_105 <- var(boot_d))
```

```
## [1] 0.137056
```

```
(bias_105=mean(boot_d)-y_hat)
```

```
## [1] 0.04217382
```

```
(MSE_105= variance_105 + bias_105^2)
```

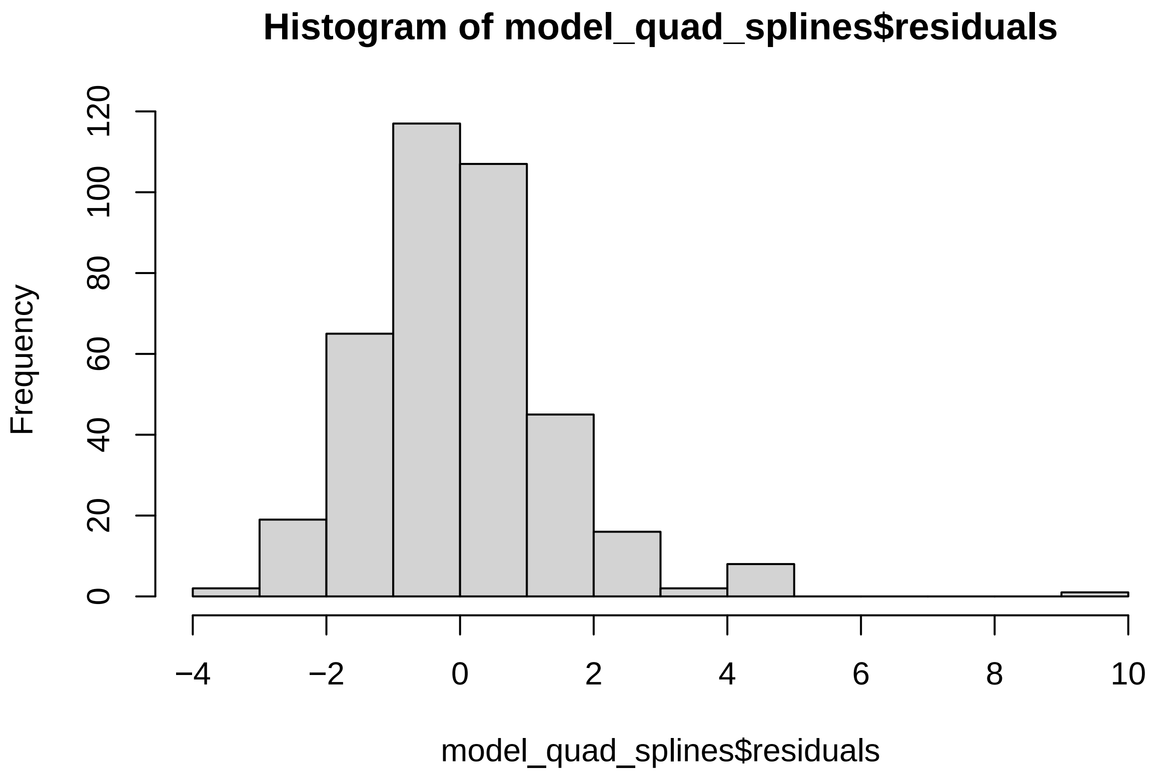
```
## [1] 0.1388347
```

2. Build an additive model for regressing κ-casein on the milk absorbance at wavenumbers 280 cm^{-1} and 700 cm^{-1}, using degree 2 b-spline bases with just one knot at the median as univariate smoother for the two predictors. Report the summary table including a comment on statistical significance. Provide an histogram of the residuals of the model.

```
model_quad_splines <-
  lm(kappa_casein ~ bs(wave_280, degree = 2, df = 3) + bs(wave_700, degree = 2, df =3),
     data = milk_samples_2)

# model_quad_splines_2 <-
#   lm(kappa_casein ~ bs(wave_280,degree = 2, knots = median(milk_samples_2$wave_280)) +
# bs(wave_700,degree = 2,
# knots = median(milk_samples_2$wave_700)), data = milk_samples_2)

summary(model_quad_splines)
```

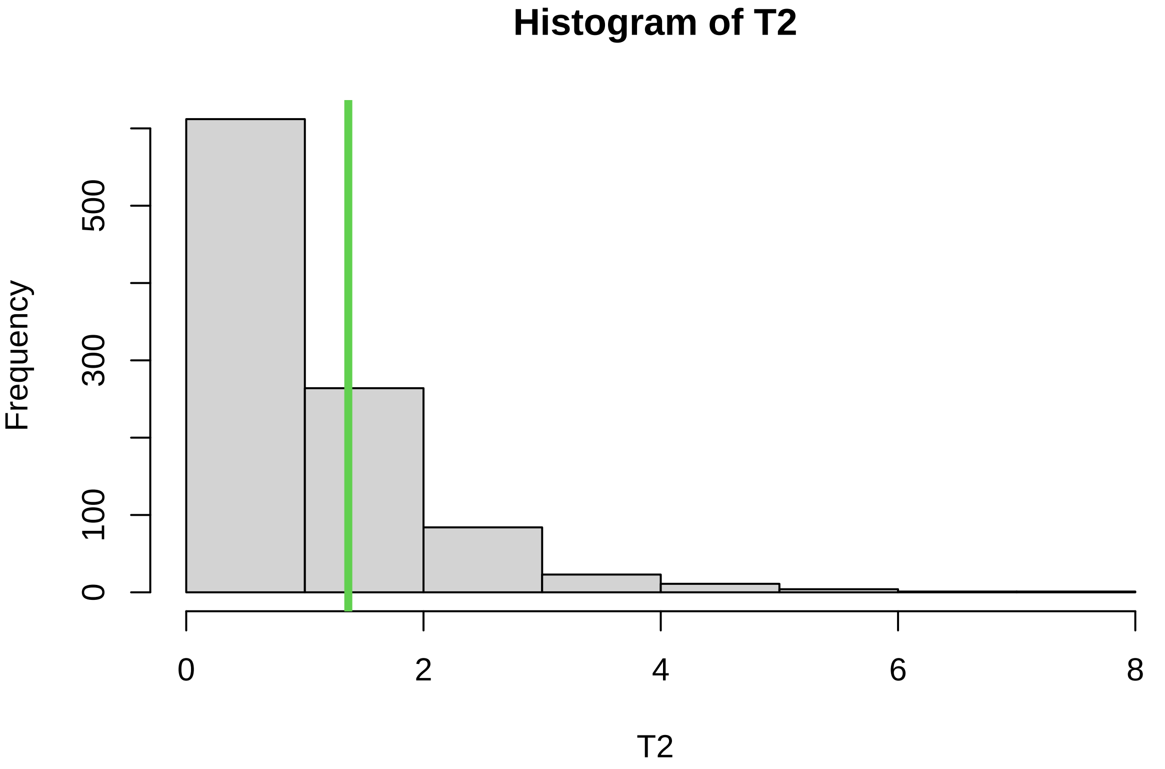
```
##
## Call:
## lm(formula = kappa_casein ~ bs(wave_280, degree = 2, df = 3) +
##     bs(wave_700, degree = 2, df = 3), data = milk_samples_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5920 -0.9446 -0.0862  0.7818  9.9704
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           1.8170     1.1097   1.637  0.10240
## bs(wave_280, degree = 2, df = 3)1   2.2046     1.0640   2.072  0.03894 *
## bs(wave_280, degree = 2, df = 3)2   5.5291     0.8263   6.691 8.07e-11 ***
## bs(wave_280, degree = 2, df = 3)3   3.1419     1.1321   2.775  0.00579 **
## bs(wave_700, degree = 2, df = 3)1  -0.1249     0.9810  -0.127  0.89877
## bs(wave_700, degree = 2, df = 3)2   0.7591     0.7449   1.019  0.30878
## bs(wave_700, degree = 2, df = 3)3  -1.0493     1.1828  -0.887  0.37558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.489 on 375 degrees of freedom
## Multiple R-squared:  0.1975, Adjusted R-squared:  0.1846
## F-statistic: 15.38 on 6 and 375 DF,  p-value: 8.903e-16
```

```
hist(model_quad_splines$residuals)
```

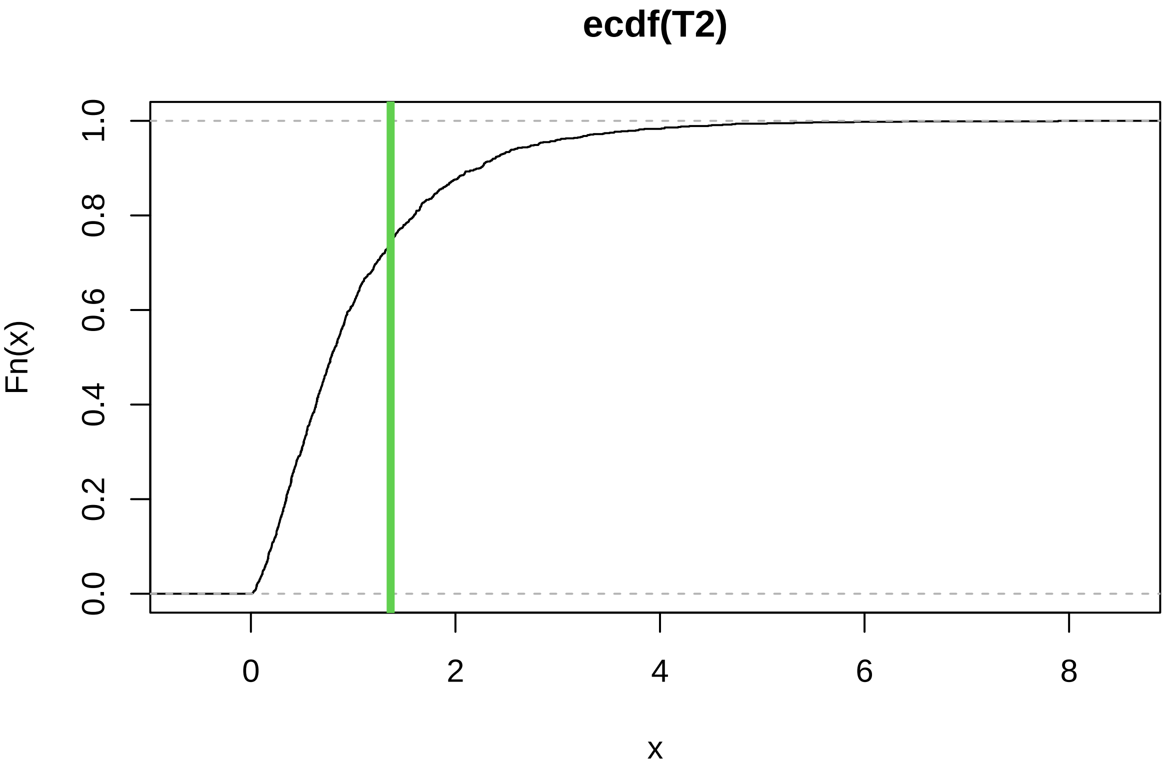


3. Build a reduced version of the previous model considering only the contribution of the milk absorbance at wavenumber 280 cm^{-1} for explaining κ-casein. Employ a permutational Anova (using the F value as test statistic) to validate which model should be preferred, specifying the null and the alternative hypothesis you are testing and report the resulting p-value. Comment on the results.

```
model_quad_splines_reduced <-
  lm(kappa_casein ~ bs(wave_280, degree = 2, df = 3),
     data = milk_samples_2)


fitted.obs <- model_quad_splines_reduced$fitted.values
res.obs <- model_quad_splines_reduced$residuals

T_0 <- anova(model_quad_splines_reduced,model_quad_splines)[2,5]

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

set.seed(2022)

for (perm in 1:B) {
  res_reduced_perm <- res.obs[sample(1:N)]
  y_perm <- fitted.obs + res_reduced_perm
  # Creo un nuovo dataset con la permuted response
  milk_samples_2_perm <- milk_samples_2
  milk_samples_2_perm$kappa_casein <- y_perm
  model_quad_splines_perm <-
  lm(kappa_casein ~ bs(wave_280, degree = 2, df = 3) + bs(wave_700, degree = 2, df =3),
     data = milk_samples_2_perm)
  model_quad_splines_reduced_perm <-
  lm(kappa_casein ~ bs(wave_280, degree = 2, df = 3),
     data = milk_samples_2_perm)
  T2[perm] <- anova(model_quad_splines_reduced_perm,model_quad_splines_perm)[2,5]
}

hist(T2, xlim = range(c(T2, t_stat)))
abline(v = T_0, col = 3, lwd = 4)
```



```
plot(ecdf(T2))
abline(v = T_0, col = 3, lwd = 4)
```



```
p_val <- sum(T2 >= T_0) / B
p_val
```

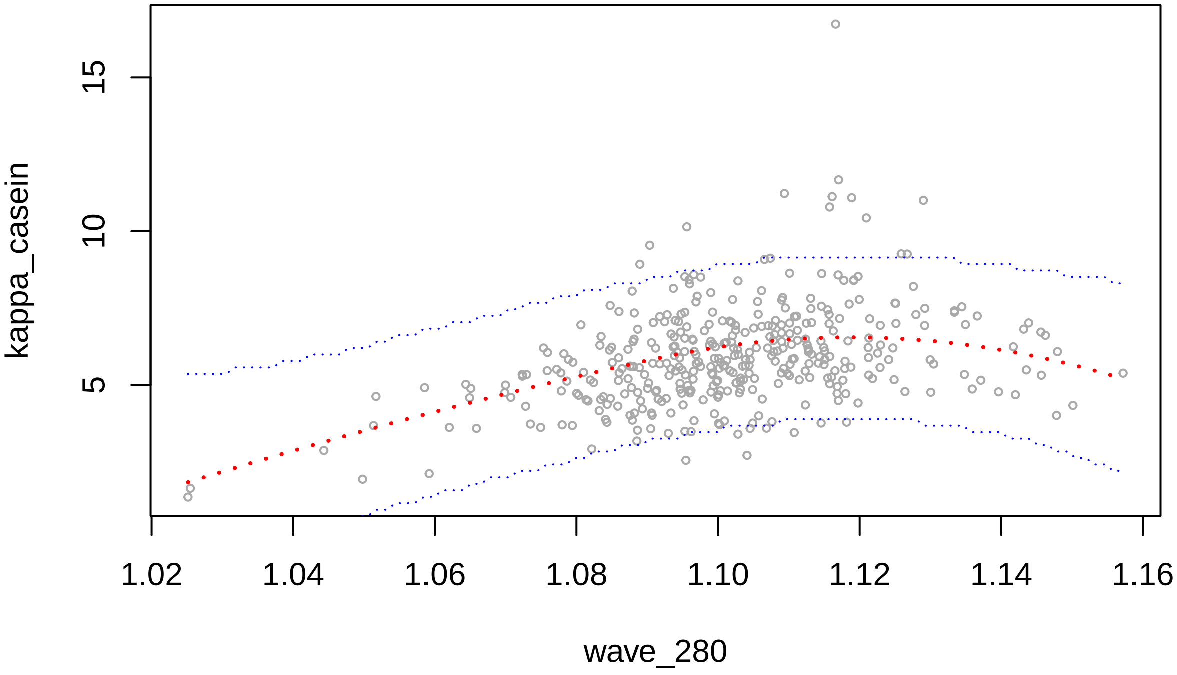
```
## [1] 0.259
```

```
# Cannot reject H0: reduced model is better
# (chemical explanation: wave 700 is associated with the presence of water in the sample,
# and it is known to be not informative of milk quality)
```

4. Compute the prediction bands for the regression model selected according to the test performed in the previous exercise, using a full conformal approach and setting $\alpha = 0.05$ as the miscoverage level

```
wave_280_grid = seq(range(milk_samples_2$wave_280)[1],
                    range(milk_samples_2$wave_280)[2],
                    length.out = 100)

preds = predict(model_quad_splines_reduced,
                list(wave_280 = wave_280_grid),
                se = T)

with(
  milk_samples_2,
  plot(
    wave_280 ,
    kappa_casein ,
    xlim = range(wave_280_grid) ,
    cex = .5,
    col = " darkgrey "
  )
)
# lines(wave_280_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 = bs(milk_samples_2$wave_280, degree = 2, df = 3)
pred_grid = matrix(bs(wave_280_grid, degree = 2, df = 3), nrow = length(wave_280_grid))

c_preds = conformal.pred(
  x = design_matrix,
  y = milk_samples_2$kappa_casein,
  pred_grid,
  alpha = 0.05,
  verbose = F,
  train.fun = lm_train,
  predict.fun = lm_predict,
  num.grid.pts = 200
)

lines(
  wave_280_grid,
  c_preds$pred ,
  lwd = 2,
  col = "red",
  lty = 3
)
matlines(
  wave_280_grid ,
  cbind(c_preds$up, c_preds$lo) ,
  lwd = 1,
  col = " blue",
  lty = 3
)
```



Exercise 3

Matthew O'Fountain has recently become passionate about robust statistics: he therefore would like to exploit these modern statistical methods to further analyze his milk samples.

1. Compute the Minimum Covariance Determinant estimator for the Milk pH, Casein Micelle Size (CMS), expressed in nm, and κ-casein (grams per liter) variables contained in the `milk_samples_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 341. Report the raw MCD estimates of location and scatter. Define a vector `ind_out_MCD` of row indexes identifying the milk samples that are outliers according to the MCD call and report it.

```r
milk_samples_3 <- readRDS(here("2022-02-11/data/milk_samples_3.Rds"))
X_mcd <- milk_samples_3[,1:3]
N <- nrow(milk_samples_3)
set.seed(2022)
fit_MCD <-
  covMcd(
    x = X_mcd,
    alpha = (N - 41) / N,
    nsamp = 1000
  )

fit_MCD$raw.center
```

```
##       kappa_casein Casein_micelle_size           Native_pH
##           5.838249          172.166188            6.652991
```

```r
fit_MCD$raw.cov
```

```
##                     kappa_casein Casein_micelle_size    Native_pH
## kappa_casein          2.42454466          -0.9034641 -0.012079556
## Casein_micelle_size  -0.90346411        1078.8037691  0.105147600
## Native_pH            -0.01207956           0.1051476  0.008539381
```

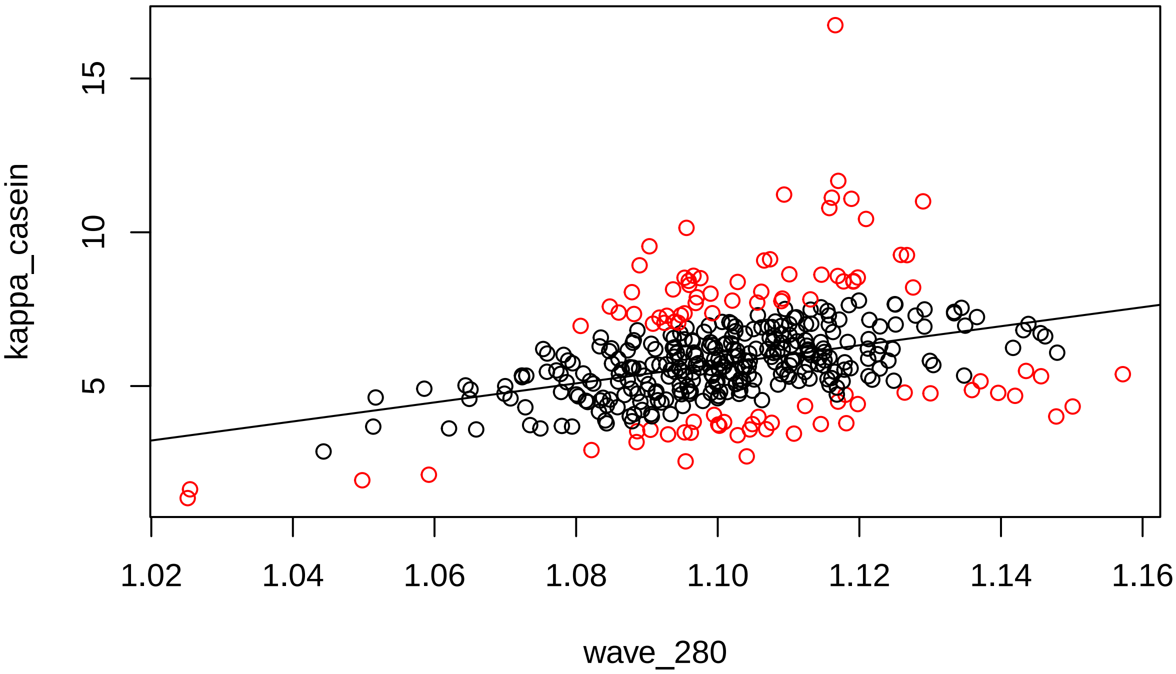
```r
ind_out_MCD <- setdiff(1:N,fit_MCD$best)
ind_out_MCD
```

```
##  [1]   1   2   6  20  37  40  45  49  52  83  95 108 144 161 185 188 200 219 234
## [20] 257 260 264 266 275 281 295 301 303 332 351 368 369 374 375 376 377 378 379
## [39] 380 381 382
```

2. Build a robust linear model to regress κ-casein on the milk absorbance at wavenumber 280 cm^{-1} 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].

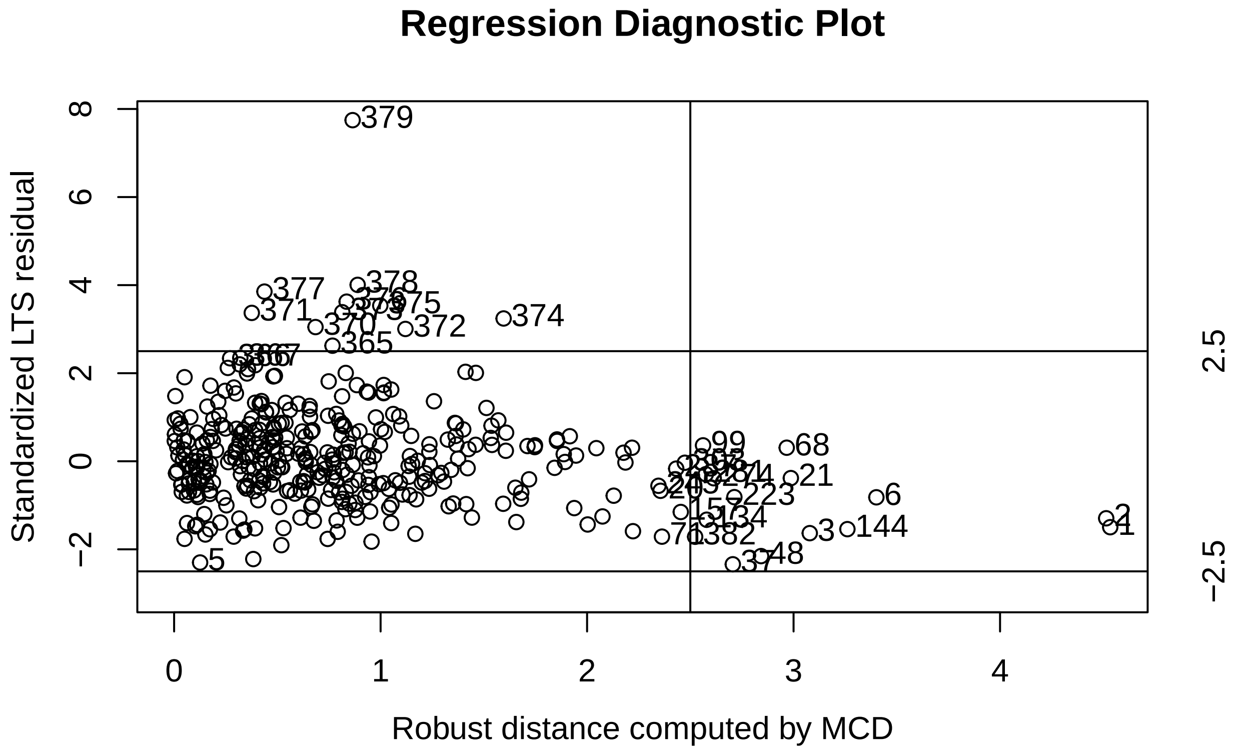
```r
fit_lts <-
  ltsReg(kappa_casein ~ wave_280, alpha = 0.75, data = milk_samples_3)

with(milk_samples_3,
     plot(wave_280 ,
          kappa_casein,
          col = ifelse(1:N%in% fit_lts$best, "black", "red")))
abline(fit_lts)
```

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



3. Provide the outlier map for the robust linear model estimated in the previous exercise. Are bad leverage points present in the dataset according to the diagnostic plot?

```
plot(fit_lts, which="rdiag")
```



```
# It seems that no bad leverage points are present,
# only vertical outliers and good leverage points
```