PD 9 wyjasnialne uczenie maszynowe

Daniel Ponikowski

29 maja 2019

Wczytanie i podział danych

```
X <- read.table("rotatingHyperplane.data")
y <- read.table("rotatingHyperplane.labels")

X1 <- X[1:20000,]
y1 <- y$V1[1:20000] %>% as.factor()
train1_num <- createDataPartition(y = y1,p = 0.8,list = FALSE)
train1 <- X1[train1_num,]
test1 <- X1[-train1_num,]
y1_train <- y1[train1_num]
y1_test <- y1[-train1_num]

X2 <- X[180001:200000,]
y2 <- y$V1[180001:200000] %>% as.factor()
train2_num <- createDataPartition(y = y2,p = 0.8,list = FALSE)
train2 <- X2[train2_num,]
test2 <- X2[-train2_num,]
y2_train <- y2[train2_num]
y2_test <- y2[-train2_num]
```

Modele

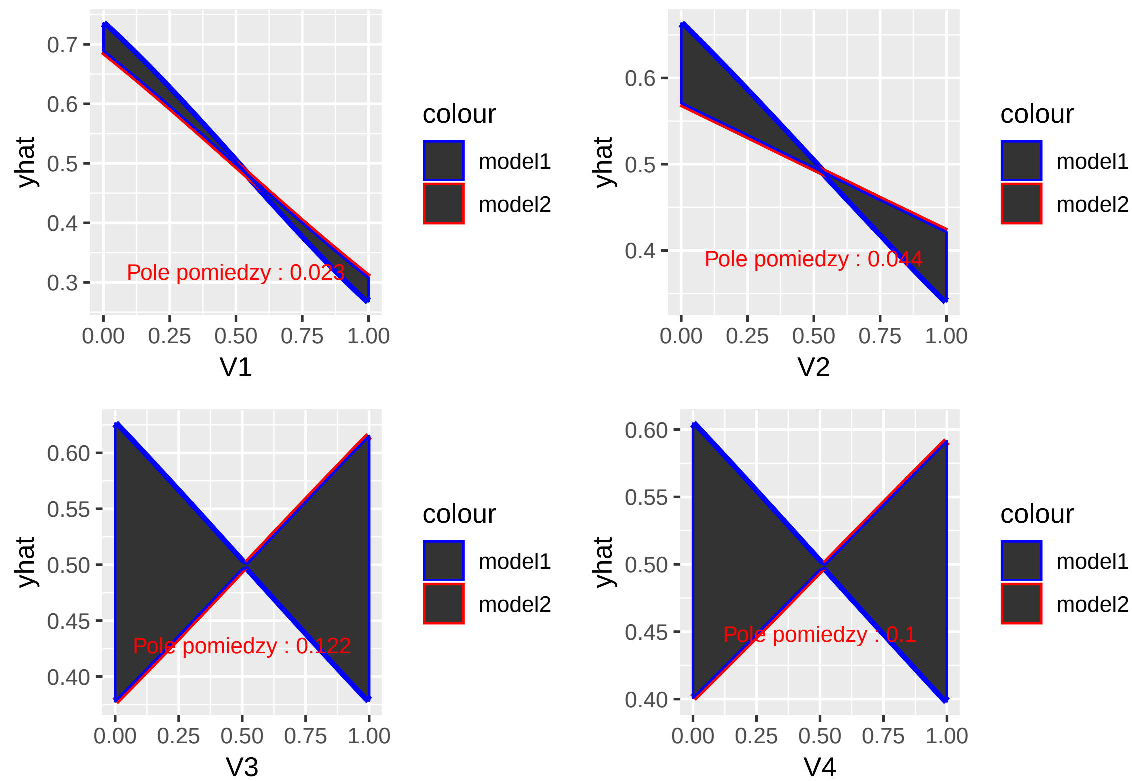
Dopasowałem 2 modele, jeden na zbiorze treningowym pochodzacym z pierwszych 10% calego zbioru, drugi na zbiorze treningowym pochodzącym z ostatich 10% całego zbioru.

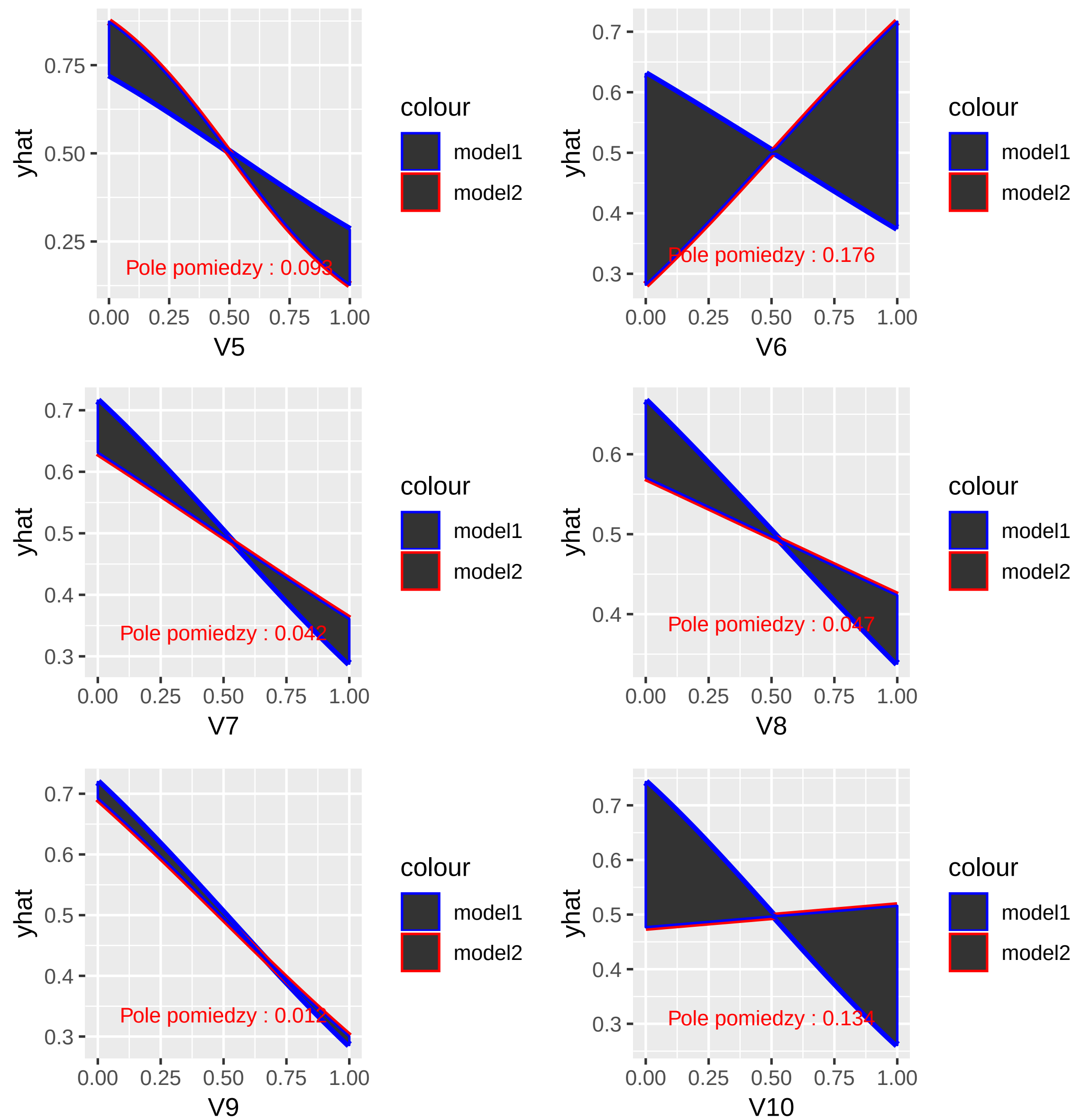
```
logit1 <- train(x = train1,y = y1_train,method = "glmnet",family = "binomial")
logit2 <- train(x = train2,y = y2_train,method = "glmnet",family = "binomial")
```

PD ploty zmiennych

```
pdp_2models <- function(model1,model2,n=51,ncol=2,zmienne){
  variables <- model1$trainingData %>% colnames() %>%
    "["(1:(model1$trainingData %>%
             colnames() %>% length() -1)) %>% "["(zmienne)
  a <- list()
  for (zmienna in variables){
  p1 <- partial(model1, pred.var = zmienna,prob = TRUE,grid.resolution = n)
  p2 <- partial(model2, pred.var = zmienna,prob = TRUE,grid.resolution = n)
  df2 <- cbind(model1 = p1, model2 = p2)
  df2$max <- ifelse(df2$model1.yhat > df2$model2.yhat,df2$model1.yhat,df2$model2.yhat )
  df2$min <- ifelse(df2$model1.yhat < df2$model2.yhat,df2$model1.yhat,df2$model2.yhat )
  colnames(df2) <- c("V1","yhat1","V2","yhat2","max","min")
  intersect <- sum(p1[[zmienna]] %>% diff() * (df2$max[2:length(df2$max)] - df2$min[
    2:length(df2$max)])) %>% round(3)
  a[[zmienna]] <- ggplot(df2,aes(x = V1, y = yhat1,col = "model1")) +
```

```
    geom_line(size =1.5) +
    geom_line(aes(x = V2, y = yhat2,
                  col = "model2"),size=1.5)+
    geom_ribbon(aes(ymin=min,ymax = max))+
    scale_fill_manual(values = c("blue", "red"))+
    scale_color_manual(values = c("blue", "red"))+
    xlab(zmienna)+ylab("yhat") +
    annotate(geom = "text",x = min(p1[,1]) + 0.5, y = min(df2$min) + 0.05,
             label = paste("Pole pomiedzy :",intersect)
             ,color="red", size = 3)
  }
  grid.arrange(grobs = a,ncol = ncol, nrow = 2)
}
```

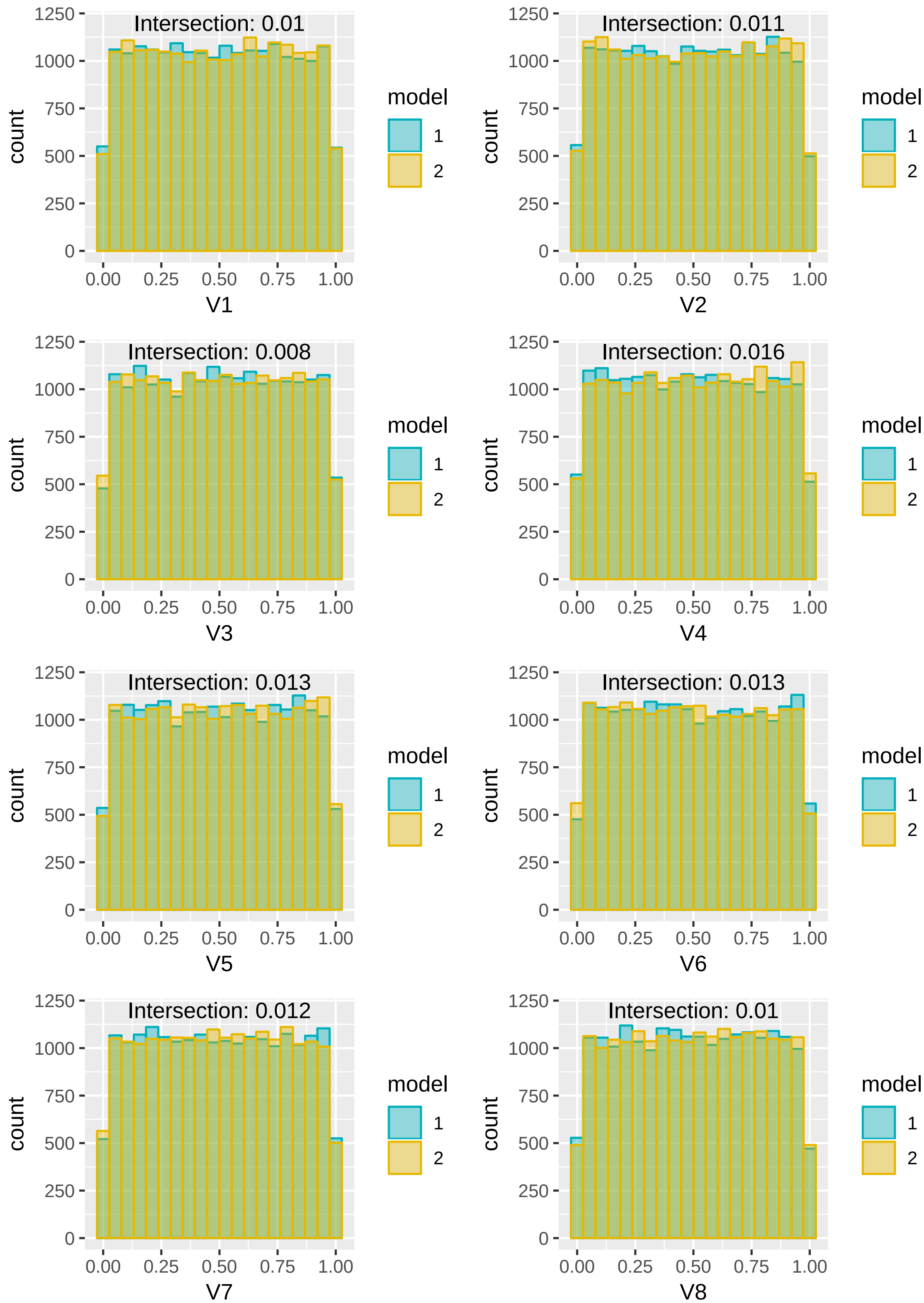


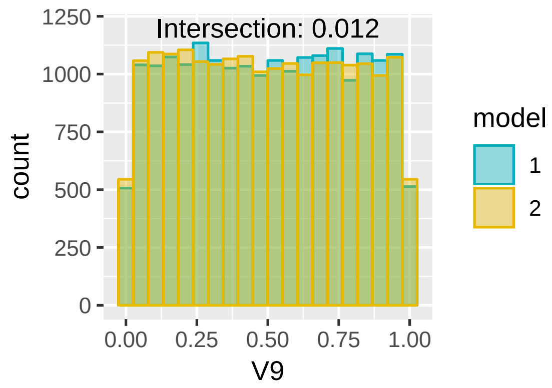


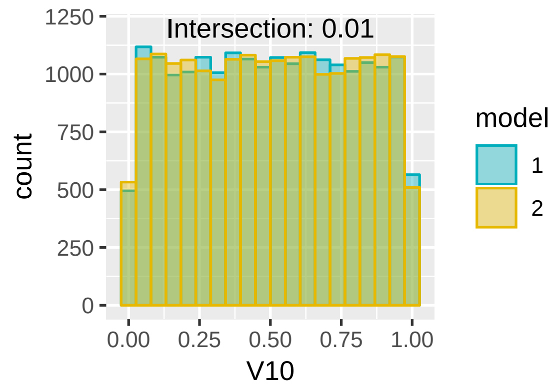
Dla zmiennych **V2**,**V3**,**V4**,**V6** oraz **V10**, modele "wychwycily" odwrotna zaleznosc prawdopodobienstwa klasy 1, tzn model 1 wraz ze wzrostem tych zmiennych przewiduje coraz mniejsze prawdopodbienstwo przynaleznosci do klasy 1, a model zbudowany na drugiej czesci zbioru wraz ze wzrostem tych zmiennych zwieksza prawdopodobienstwo klasy 1. Dla pozostalych zmiennych obserwujemy wieksze badz mniejsze roznice w krzywych, jednak zachowuja monotonicznosc.

Histogramy zmiennych objasniajacych

```
intersection <- function(x1,x2){
    dens1 <- density(x1)
    dens2 <- density(x2)
    minimum <- data.frame(y1 = dens1$y,y2 = dens2$y) %>% apply(MARGIN = 1,min)
    1 - sum(dens1$x %>% diff() * (minimum[2:length(minimum)]))
}


data_do_histogram <- rbind(X1,X2)

data_do_histogram$model <- c(rep(1,nrow(X1)),rep(2,nrow(X2))) %>% as.factor()

hist_all <- function(data_do_histogram,nbins = 20,zmienne){
  n <- nrow(data_do_histogram)
  variables <- data_do_histogram %>% colnames() %>%
    "["(1:(length(data_do_histogram %>% colnames())-1)) %>% "["(zmienne)
  a <- list()
  for (zmienna in variables){
    df <- data.frame(V1 = data_do_histogram %>% "[["(zmienna),model = data_do_histogram %>%
                       "[["("model") %>% as.factor())
      x1 <- data_do_histogram[1:(n/2),zmienna]
      x2 <- data_do_histogram[(n/2):n,zmienna]
      int <- intersection(x1,x2) %>% round(3)
      a[[zmienna]] <- ggplot(data = df, aes(x = V1)) +
      geom_histogram(aes(color = model, fill = model),
                     position = "identity", bins = nbins, alpha = 0.4) +
      scale_color_manual(values = c("#00AFBB", "#E7B800")) +
      scale_fill_manual(values = c("#00AFBB", "#E7B800")) + xlab(zmienna) +
        annotate(geom = "text", label = paste("Intersection:",int), y = 1200,x = 0.5)
  }
  grid.arrange(grobs = a,ncol = 2, nrow = 2)
}
```


Intersection: 0.01
count
V1
model
1
2
Intersection: 0.011
count
V2
model
1
2
Intersection: 0.008
count
V3
model
1
2
Intersection: 0.016
count
V4
model
1
2
Intersection: 0.013
count
V5
model
1
2
Intersection: 0.013
count
V6
model
1
2
Intersection: 0.012
count
V7
model
1
2
Intersection: 0.01
count
V8
model
1
2
0
250
500
750
1000
1250
0.00
0.25
0.50
0.75
1.00



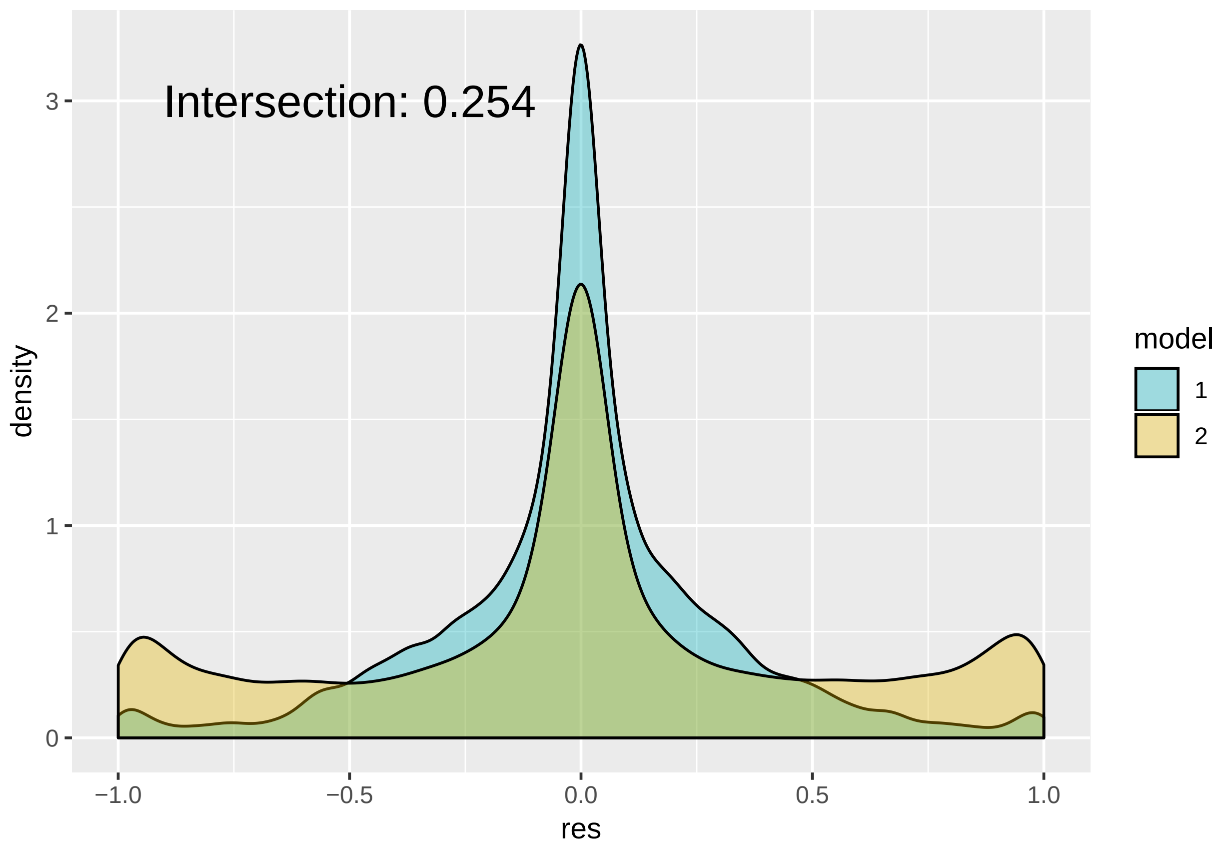


Rozkład zmiennych objasnianych w poszczegolnych fragmentach zbioru danych nie rozni sie znaczaco, co pokazuja wartosci **Intersection**.

Porownanie rozkladu reszt

Porownam rozklady reszt tzn. roznice w prawdziwej wartosci zmiennej objasnianej z odpowiedzia modelu (modleu dopasowanego na pierwszej czesci zbioru). Dokladniej, niebieskim kolorem oznaczona jest wyestymowana gestosc reszt pierwszego modelu na pierwszym zbiorze testowym, a zoltym gestosc reszt oblicoznych na ostatnich 10% calego zbioru.

```
residual1 <- as.numeric(y1_test) - 1 - predict(logit1,newdata = test1,"prob")[,2]

residual2 <- as.numeric(y2) - 1 - predict(logit1,newdata = X2,"prob")[,2]

reszty <- data.frame(res = c(residual1,residual2),
                     model = factor(c(rep(1,nrow(test1)),rep(2,nrow(X2)))))
dens1 <- density(residual1)
dens2 <- density(residual2)
minimum <- data.frame(y1 = dens1$y,y2 = dens2$y) %>% apply(MARGIN = 1,min)
intersection <- 1 - sum(dens1$x %>% diff() * (minimum[2:length(minimum)])) %>% round(3)


ggplot(reszty, aes(res, fill = model)) +
  geom_density(alpha = 0.35)+
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800")) +
  annotate(geom = "text",x = -0.5,y = 3,label = paste("Intersection:",intersection),size = 6)
```



Widac wyrazne przesuniecie sie reszt na “ogony”, tzn wieksza masa jest w okolicach punktow 1 i -1. Rozkład reszt dla drugiej czesci zbioru jest o wiele mniej skoncetrowany wokol zera.