Exercise 4

Tobias Raidl, 11717659

2023-10-30

Contents

Setup dataset

```r
library(UsingR)
```

```
## Warning: Paket 'UsingR' wurde unter R Version 4.1.3 erstellt
```

```
## Lade nötiges Paket: MASS
```

```
## Lade nötiges Paket: HistData
```

```
## Lade nötiges Paket: Hmisc
```

```
## Warning: Paket 'Hmisc' wurde unter R Version 4.1.3 erstellt
```

```
##
## Attache Paket: 'Hmisc'
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     format.pval, units
```

```r
library(dplyr)
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.1.3 erstellt
```

```
##
## Attache Paket: 'dplyr'

## Die folgenden Objekte sind maskiert von 'package:Hmisc':
##
##     src, summarize

## Das folgende Objekt ist maskiert 'package:MASS':
##
##     select

## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df = select(fat, -c(case, body.fat.siri, ffweight, density))
X = select(df, -c(body.fat))
y = df$body.fat
```

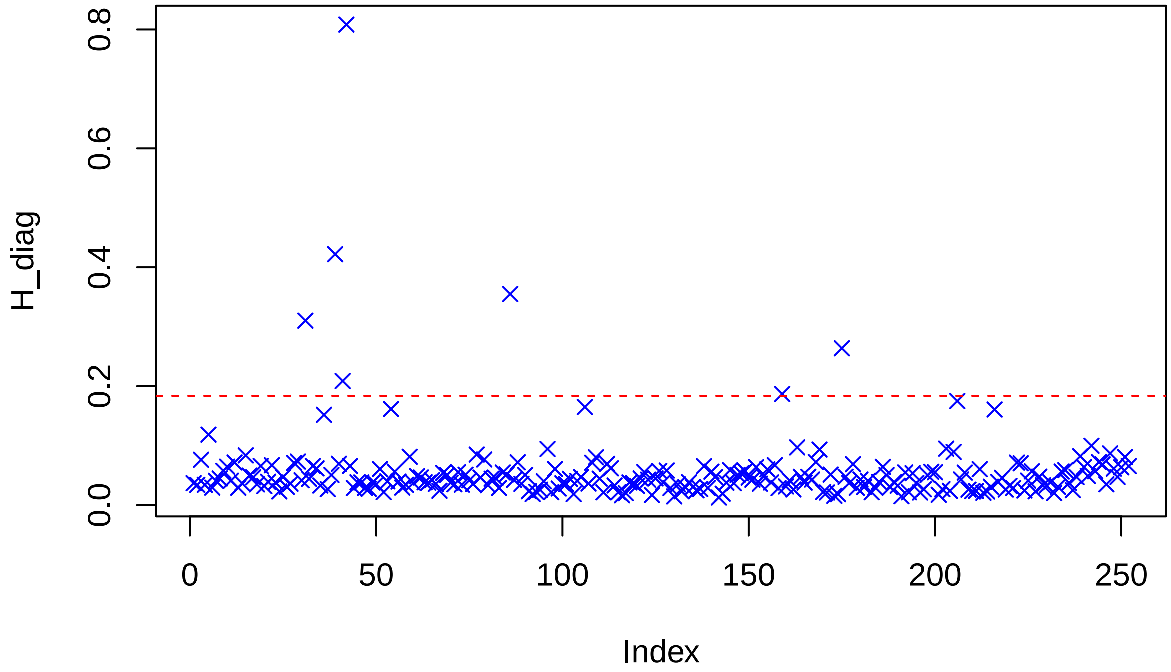
1

Investigate if there are leverage points by using …

a

… classical diagnostic based on the diagonal elements of the hat matrix

```
Xm = data.matrix(X)
H = Xm %*% solve(t(Xm) %*% Xm) %*% t(Xm)
H_diag = diag(H)
plot(H_diag, pch = 4, col = "blue")
abline(h = quantile(H_diag, 0.975), lty = "dashed", col = "red")
```



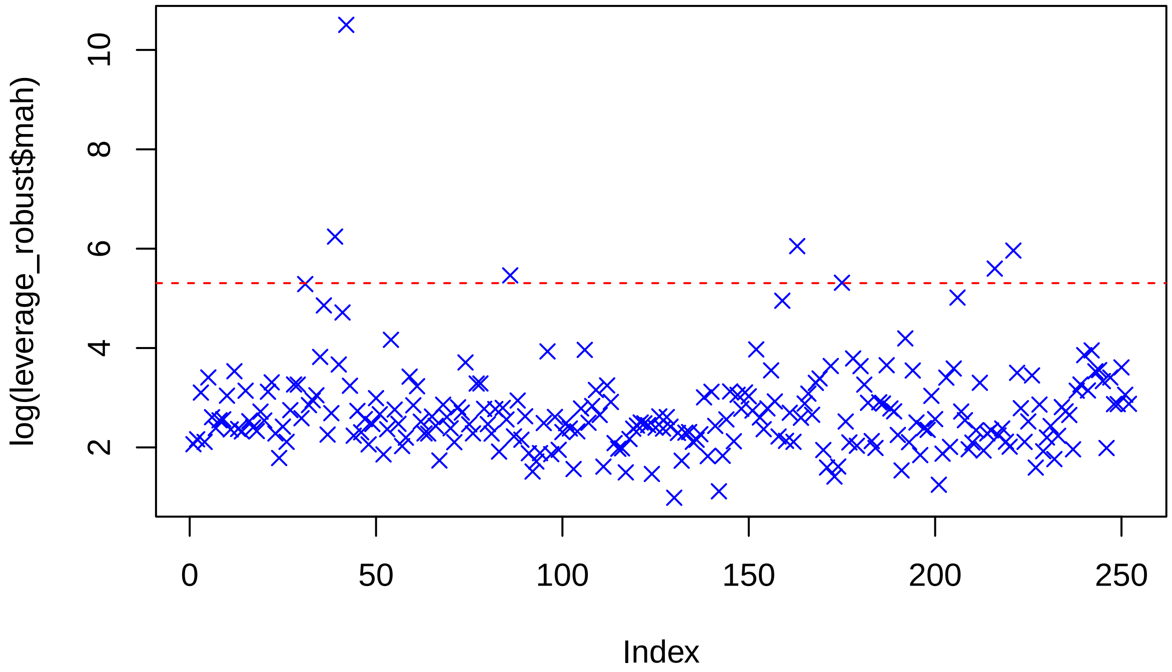
b

… robust diagnostics based on robust Mahalanobis distances. Use the MCD estimator (covMcd() from the package robustbase) for this purpose.

```
library(robustbase)
```

```
## Warning: Paket 'robustbase' wurde unter R Version 4.1.3 erstellt
```

```
leverage_robust = covMcd(Xm)
plot(log(leverage_robust$mah), pch = 4, col = "blue")
abline(h = quantile(log(leverage_robust$mah), 0.975), lty = "dashed", col = "red")
```



```
leverage_points = which(leverage_robust$mah > quantile(leverage_robust$mah, 0.975))
print(paste("leverage points:", paste(leverage_points, collapse = ", ")))
```

```
## [1] "leverage points: 39, 42, 86, 163, 175, 216, 221"
```

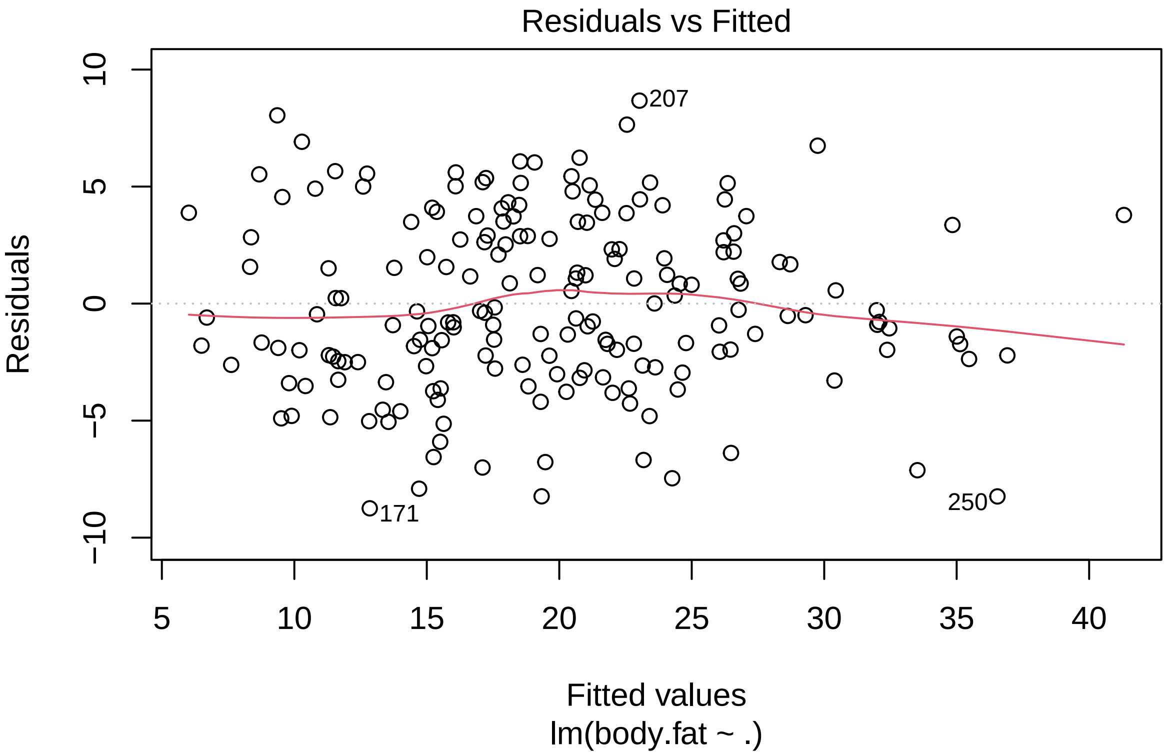
What do you conclude? The classical non robust method results in 6 observations having a p-value greater than 0.975. The robust MCD gets 7 observations with a greater p-value. Therefore the difference seems to be negligible.

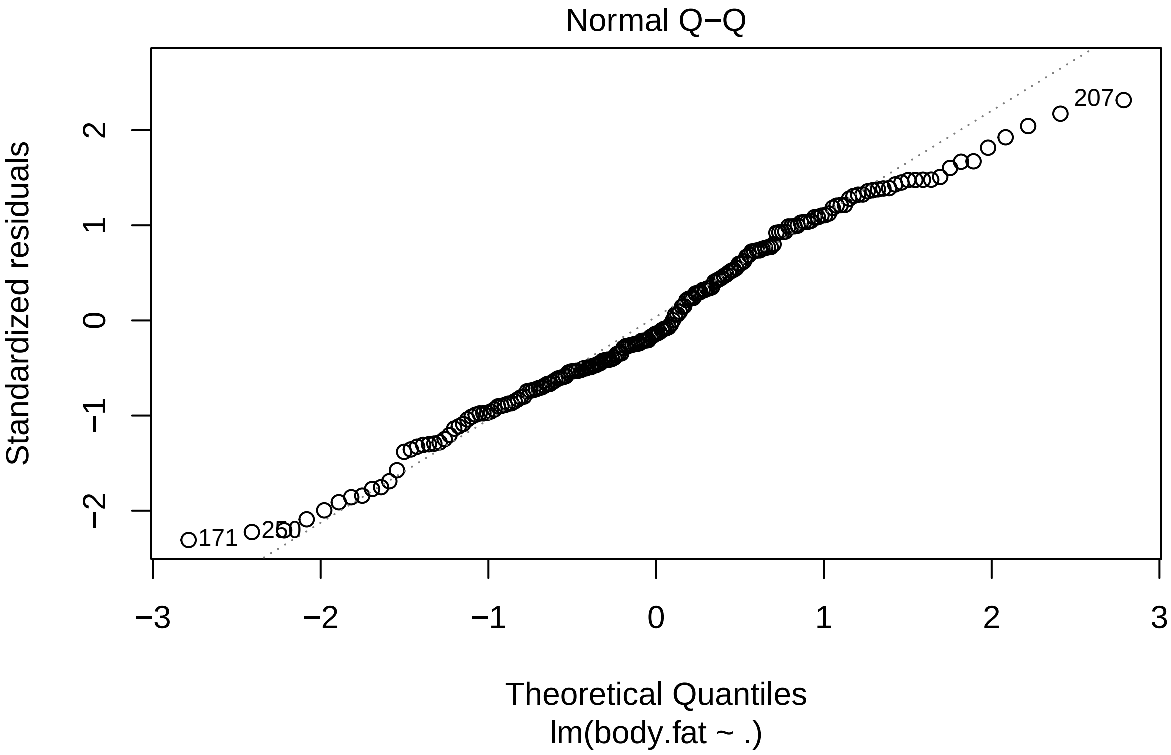
2

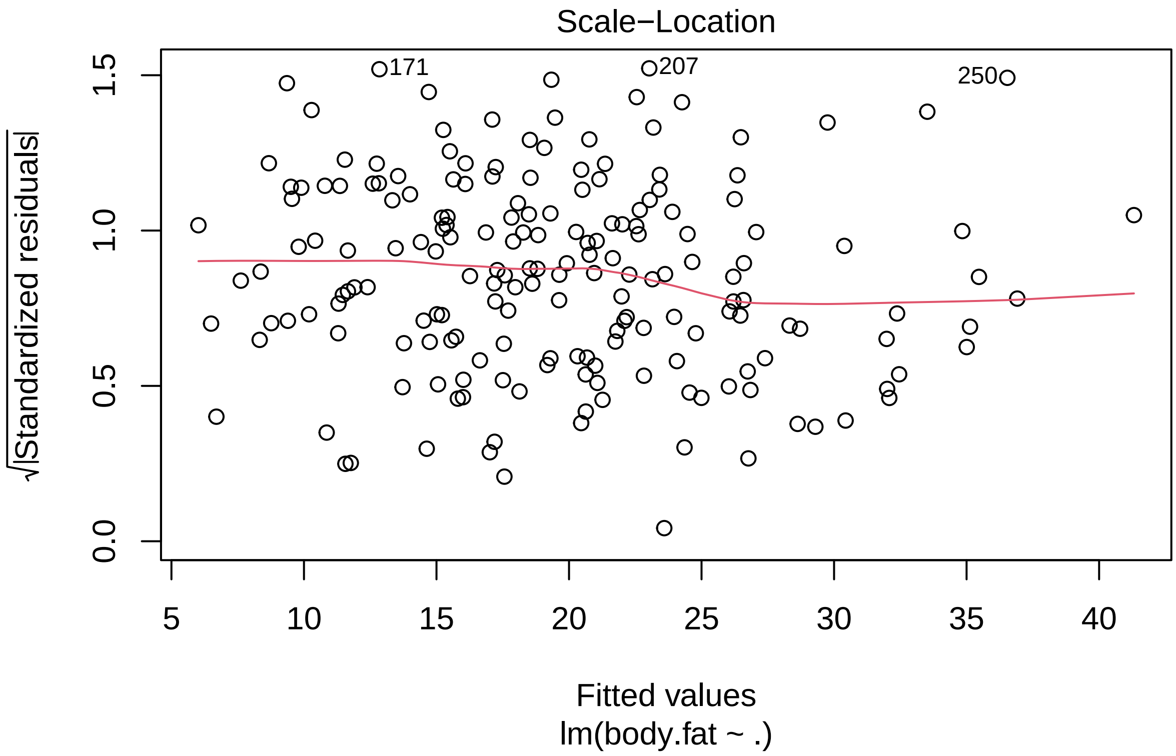
Now split the observations randomly into training and test data, e.g. in a proportion 3:1. Then apply linear regression to the training set, with ## a the least-squares estimator (lm()) and the robust MM-estimator (lmrob() from library(robustbase)). Interpret the results of summary() and plot().

```
set.seed(11717659)
sample <- sample(c(TRUE, FALSE), nrow(df), replace = TRUE, prob = c(3/4, 1/4))
train <- df[sample, ]
test <- df[!sample, ]

library(robustbase)

model = lm(body.fat ~ ., train)
```

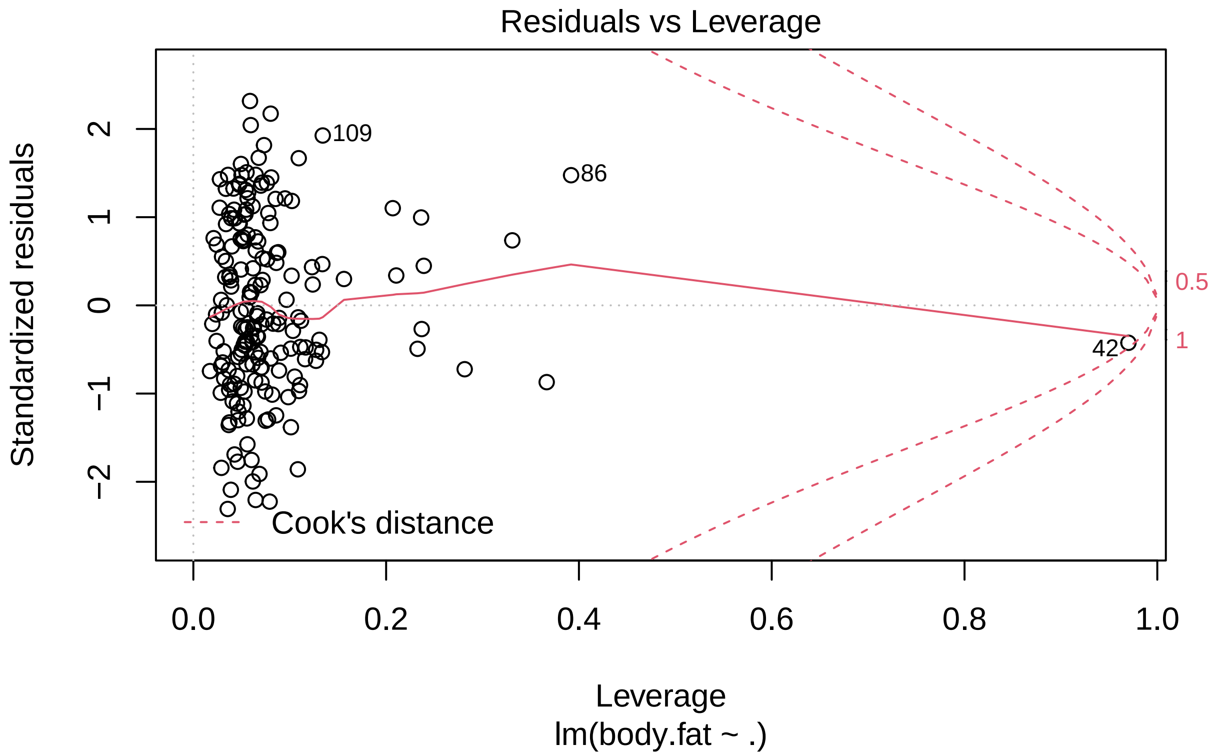
```r
# summary(model)
plot(model)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
207
171
25
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(body.fat ~ .)



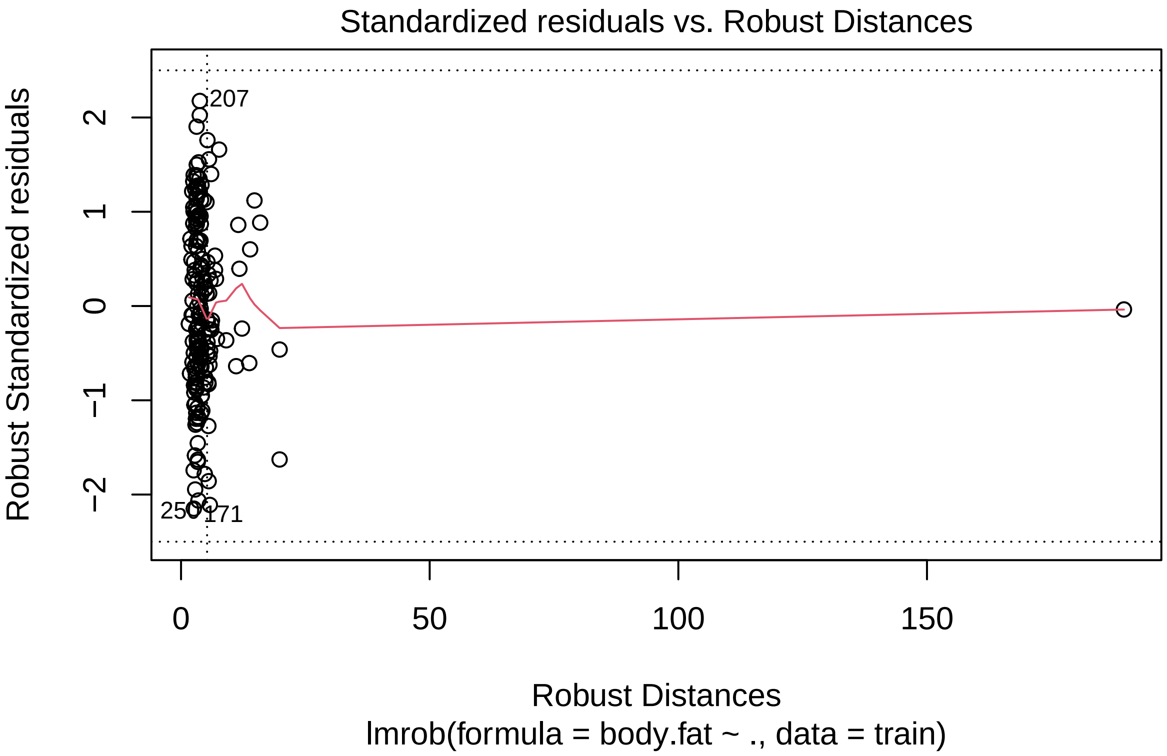
```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs wurden erzeugt

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs wurden erzeugt
```

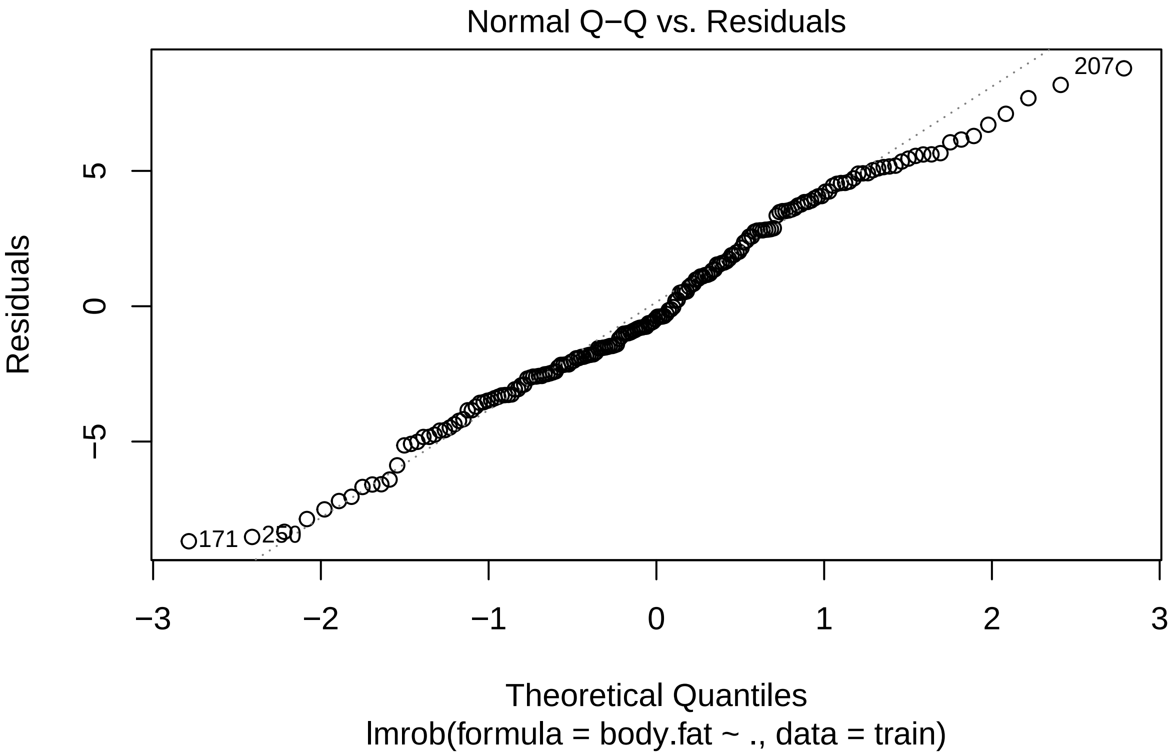


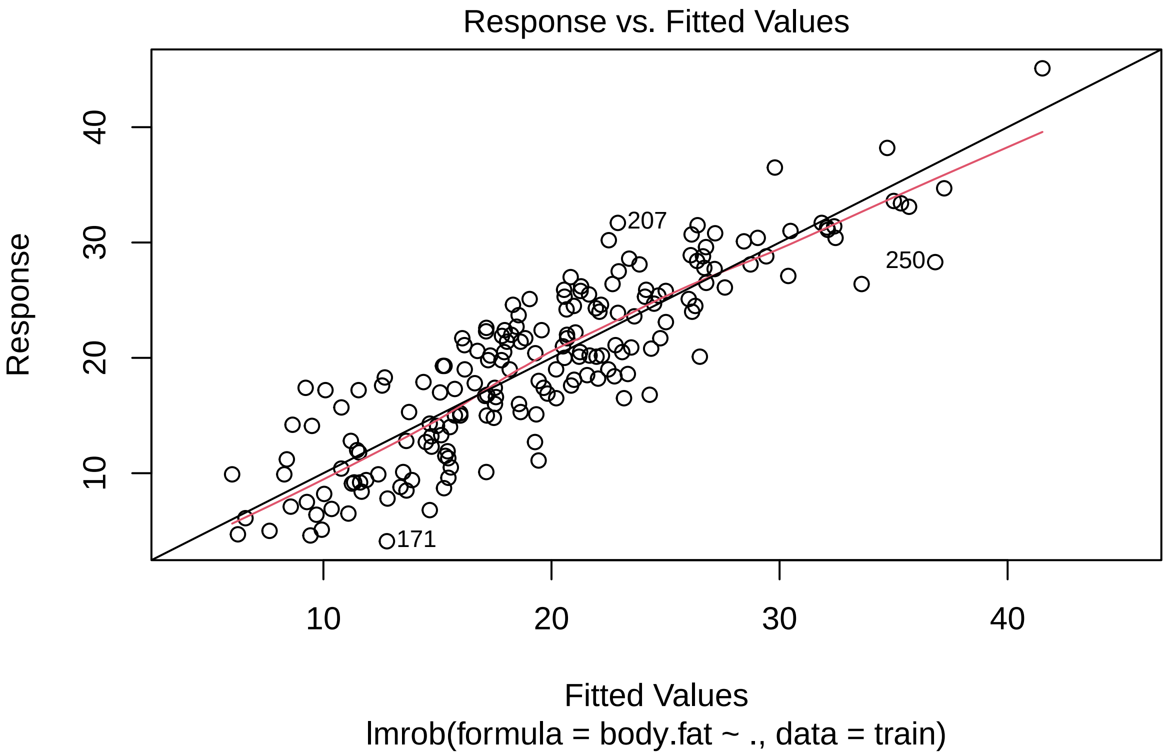
```r
model_rob = lmrob(body.fat ~ ., train)
# summary(model_rob)
plot(model_rob)
```

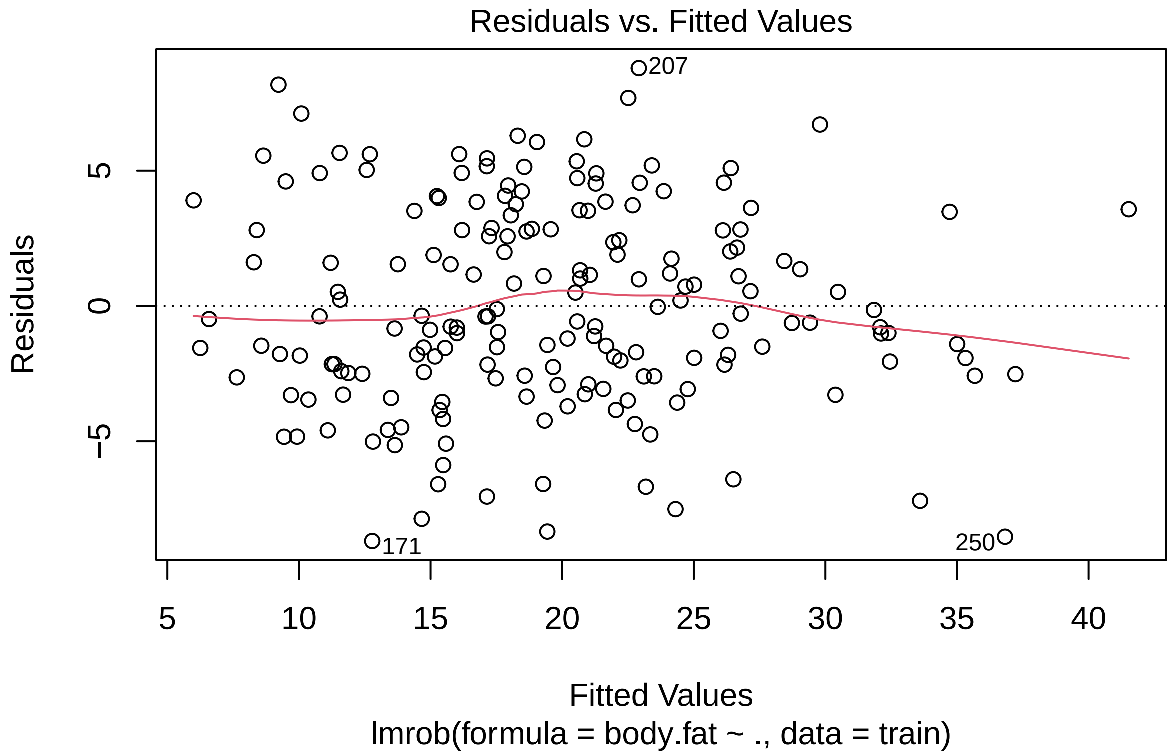
```
## recomputing robust Mahalanobis distances
```

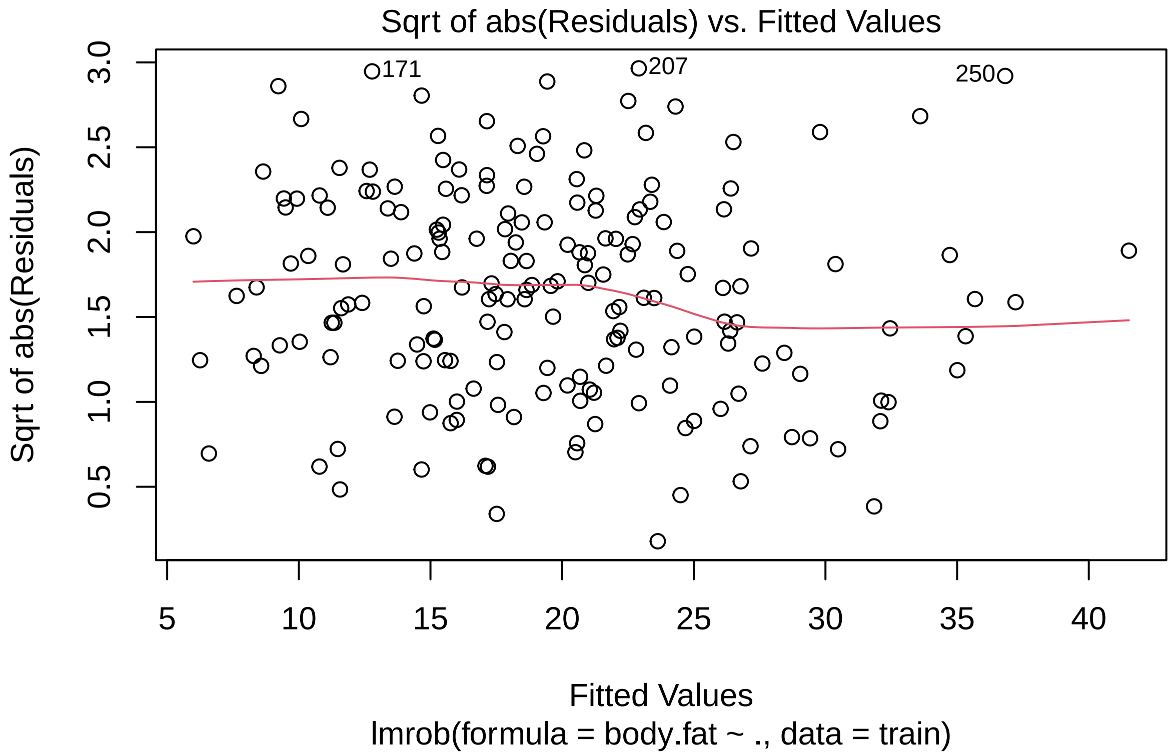
```
## saving the robust distances 'MD' as part of 'model_rob'
```


Standardized residuals vs. Robust Distances
Robust Standardized residuals
2
1
0
−1
−2
207
250
171
0
50
100
150
Robust Distances
lmrob(formula = body.fat ~ ., data = train)


Normal Q−Q vs. Residuals
Residuals
5
0
−5
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lmrob(formula = body.fat ~ ., data = train)
207
171
250


Response vs. Fitted Values
Response
Fitted Values
lmrob(formula = body.fat ~ ., data = train)
10
20
30
40
10
20
30
40
207
250
171


Residuals vs. Fitted Values
Residuals
Fitted Values
lmrob(formula = body.fat ~ ., data = train)
5
0
−5
5
10
15
20
25
30
35
40
207
171
250



Is robustness recommendable? I dont think so because the non robust estimator, detects nearly as many outliers as the non robust estimator.

b

Compute the Cook distances from the least-squares solution.

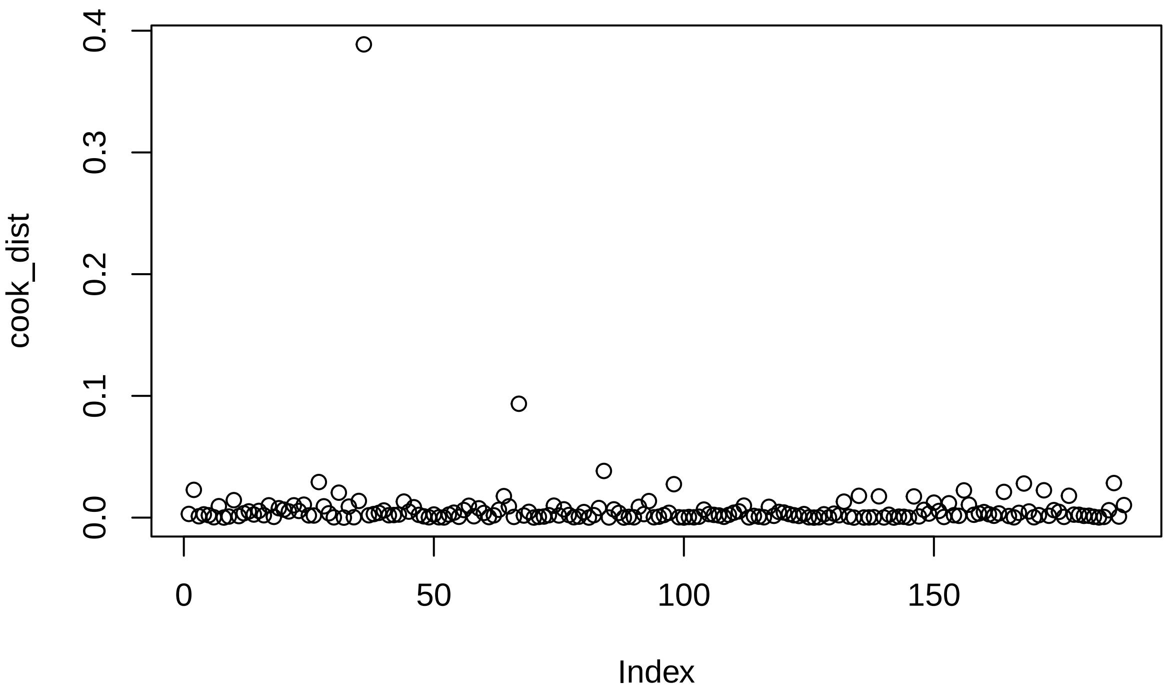
```r
cook_dist = cooks.distance(model)
summary(cook_dist)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000000 0.0007219 0.0023485 0.0074450 0.0063865 0.3887606
```

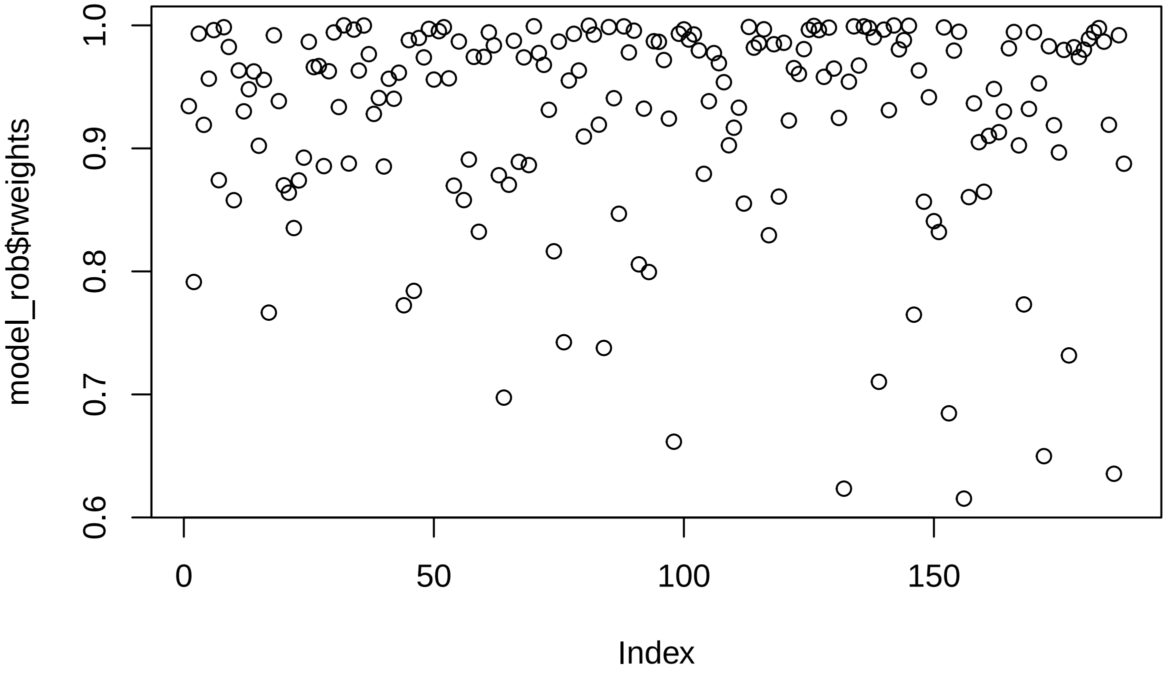
```r
summary(model_rob$weights)
```

```
## Length  Class   Mode 
##      0   NULL   NULL
```

```r
plot(cook_dist)
```



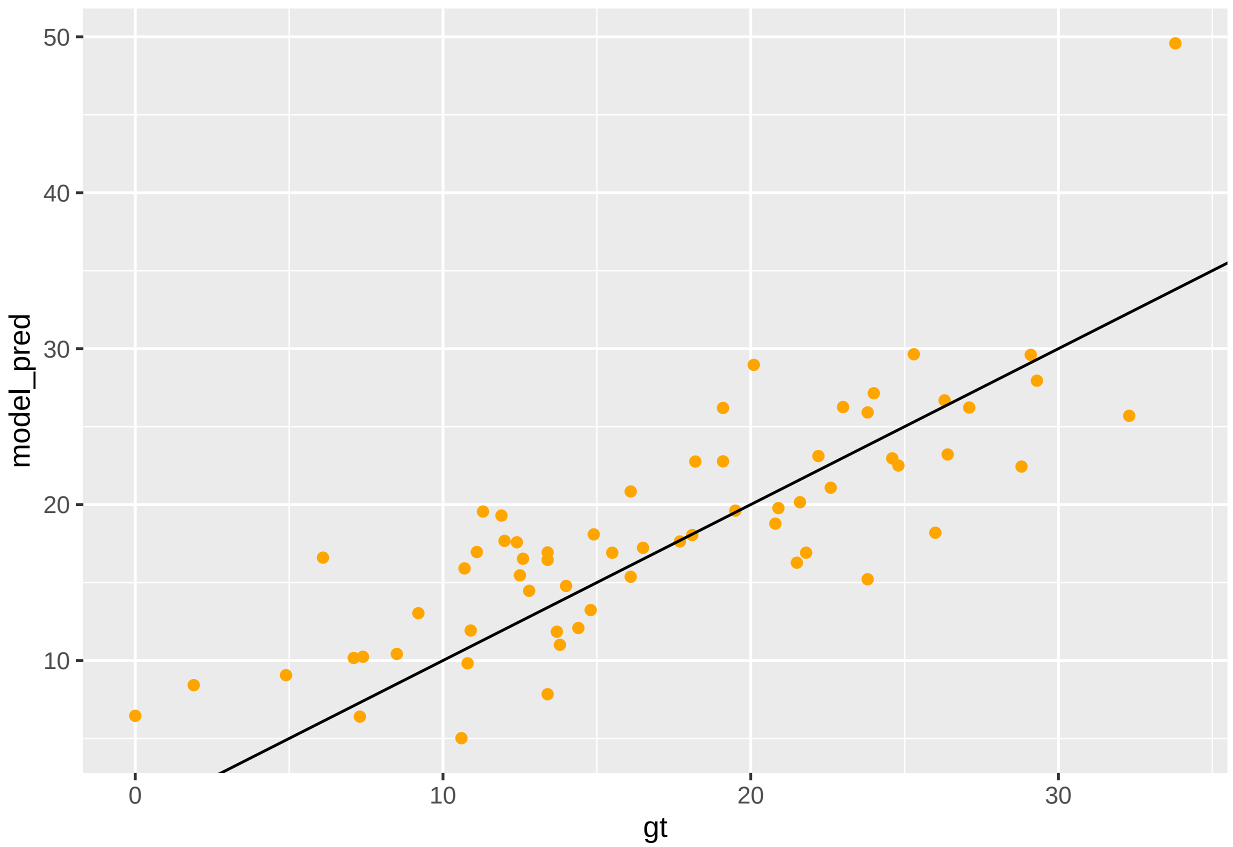
```
plot(model_rob$rweights)
```



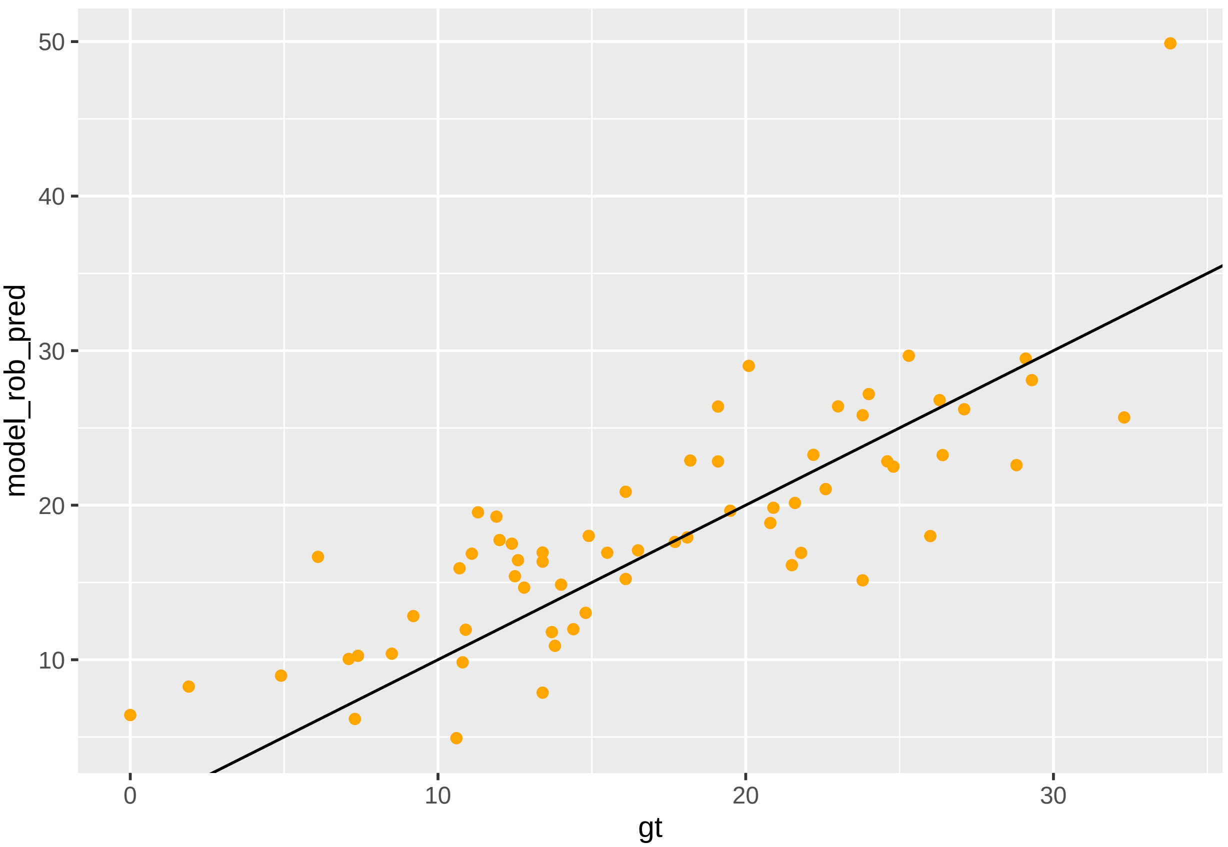
c

Use the models to predict the response of the test set. Compare the classical and robust predictions graphically and numerically using an appropriate measure of prediction accuracy.

```
library(ggplot2)
gt = test$body.fat
model_pred = predict(model, select(test, -body.fat))
model_rob_pred = predict(model_rob, select(test, -body.fat))

res = data.frame(gt, model_pred, model_rob_pred)

ggplot(res) + geom_point(aes(x = gt, y = model_pred), color = "orange") +
↪   geom_abline(intercept = 0,
    slope = 1)
```



```
ggplot(res) + geom_point(aes(x = gt, y = model_rob_pred), color = "orange") +
↪   geom_abline(intercept = 0,
    slope = 1)
```


50
40
30
20
10
model_rob_pred
0
10
20
30
gt