Supplementary Troponin Analysis

Improving risk stratification for patients with type 2 myocardial infarction

Karla Monterrubio Gómez and Catalina A. Vallejos

The objective of this analysis is to construct a linear regression model, which permits to predict log troponin I from log tropinin T. Such model, will permit to employ the proposed T2 risk score when we only have access to troponin T.

Data pre-processing

The data here used corresponds to patients with suspected acute coronary syndrome that were recruited from the Emergency Department of the Royal Infirmary of Edinburgh, a tertiary care hospital in Scotland, between 1 June 2013 and 31 March 2017 into a substudy of the HighSTEACS trial. All patients in whom the attending clinician requested cardiac troponin for suspected acute coronary syndrome were eligible. We did not enrol patients with ST-segment elevation myocardial infarction, those unable to provide consent or those from outside our region to ensure complete follow-up. Blood samples were obtained at presentation and at 6 – 12 hours as part of routine clinical care, with surplus serum or lithium-heparin plasma samples collected. Patients provided written informed consent for additional sampling at 1 or 3 hours.

The dataset contains two readings, one of troponin I and one of troponin T, for 1869 patients. In addition, the dataset contains an adjudication code, where:

- adj = 1 corresponds to Type 1 Myocardial infarction
- adj = 2 corresponds to Type 2 Myocardial infarction
- adj = 3 corresponds to Myocardial injury
- adj = 9 corresponds to NA
- adj = NA corresponds to No injury

We load the dataset and make the above adjudication codes explicit:

```r
library(readr)

substudy <-  as.data.frame(read_csv("~/Documents/Postdoc/DEMAND/highsteacs_substudy_troponin.csv"))

str(substudy)
```

```
## 'data.frame':    1869 obs. of  6 variables:
##  $ substudyid : num  1 2 3 5 6 7 8 9 10 11 ...
##  $ tni1_result: num  11508 3 5 3 8 ...
##  $ tni2_result: num  15733 4 5 3 12 ...
##  $ tnt1_result: num  712 4 17 7 6 11 4.99 66 12 20 ...
##  $ tnt2_result: num  NA 6 17 6 7 11 4.99 62 14 25 ...
##  $ adj        : num  1 NA NA NA NA NA NA 2 NA 1 ...
```

```r
substudy$adj[substudy$adj == 1] <- "Type 1 MI"
substudy$adj[substudy$adj == 2] <- "Type 2 MI"
```

```
substudy$adj[substudy$adj == 3] <- "Myocardial injury"
substudy$adj[substudy$adj == 9] <- NA
substudy$adj[is.na(substudy$adj)==TRUE] <- "No injury"
```

Because our aim is to model the relationship between tropinin I and tropinin T. Below, we re-arrange the data by stacking the two available tropinin readings. In addition, we remove rows of the stacked dataset where at leat one of the troponin reading were unvailable.

```
stacked_data <- data.frame(cbind( tni = c(substudy$tni1_result, substudy$tni2_result),
                                  tnt = c(substudy$tnt1_result, substudy$tnt2_result)),
                                  adj = c(substudy$adj, substudy$adj))

#remove NAs
NA.I <- which(is.na(stacked_data$tni)==TRUE)
NA.T <- which(is.na(stacked_data$tnt)==TRUE)
stacked_data <- stacked_data[-c(unique(c(NA.I, NA.T))),]
row.names(stacked_data) <- NULL

#Number of available readings
nrow(stacked_data)
```

```
## [1] 3559
```

We further remove any troponin readings above and below the limit of detection of the assays employed (“ARCHITECT Stat High Sensitivity Troponin-I”)

- Lower limit of detection for troponin I is 3.5 ng/L and for troponin T 6.0 ng/L.
- Upper limit of detection for troponin I is 50,000 ng/L and for troponin T 10,000 ng/L.

```
below_limit <- unique(c(which(stacked_data$tni <= 3.5), which(stacked_data$tnt <= 6)))
above_limit <- unique(c(which(stacked_data$tni >= 50000), which(stacked_data$tnt >= 10000)))

data_LOD <- stacked_data[-c(below_limit, above_limit),]
row.names(data_LOD) <- NULL
nrow(data_LOD)
```

```
## [1] 1361
```

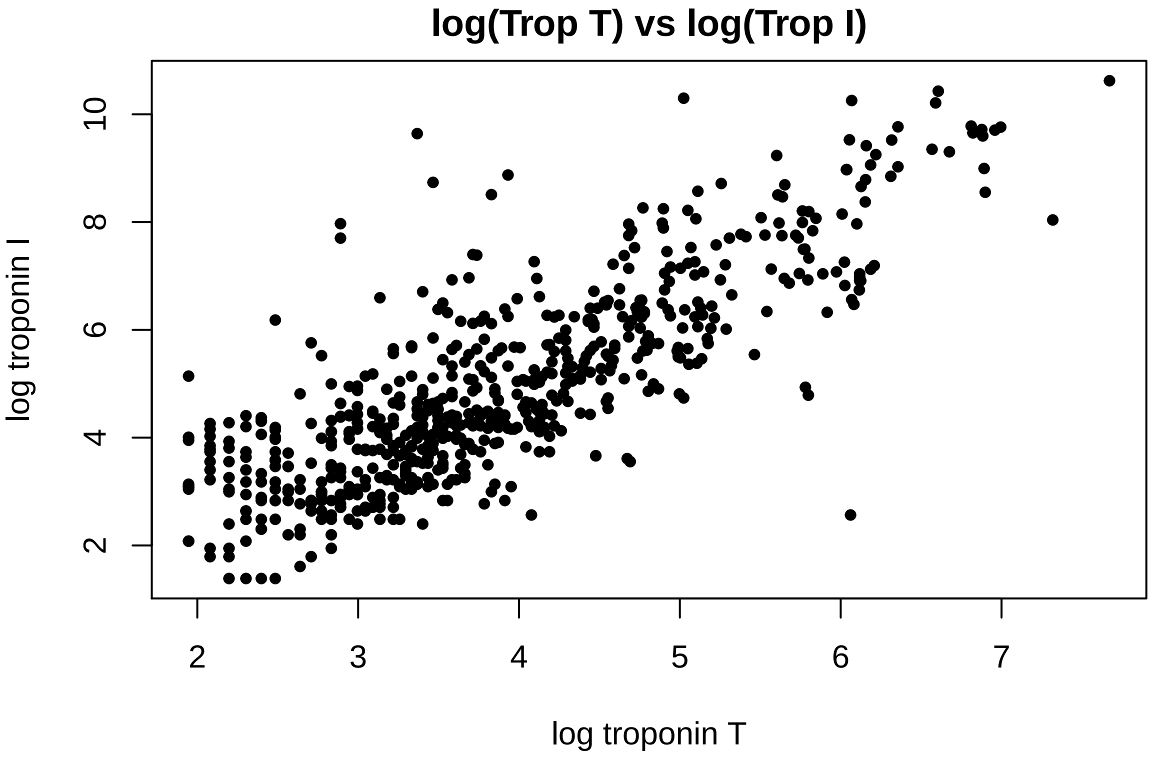
Furthermore, as our objective is to model the relationship between troponin I and T in populations with MI, we remove readings corresponding to those subjects that have an adjudicated diagnose of no injury.

```
data_LOD_subset <- data_LOD[-which(data_LOD$adj == "No injury"),]
row.names(data_LOD_subset) <- NULL

#Number of available troponin readings
nrow(data_LOD_subset)
```

```
## [1] 653
```

Finally, we compute the logarithm of both troponin I and T and produce a scatter plot of the data:

```
data_LOD_subset$log.TropI <- log(data_LOD_subset$tni)
data_LOD_subset$log.TropT <- log(data_LOD_subset$tnt)

par(mar = c(4, 4, 2, .1))
plot(data_LOD_subset$log.TropT, data_LOD_subset$log.TropI,
     pch = 20,
     main = "log(Trop T) vs log(Trop I)",
    xlab = "log troponin T", ylab = "log troponin I")
```



Model fitting

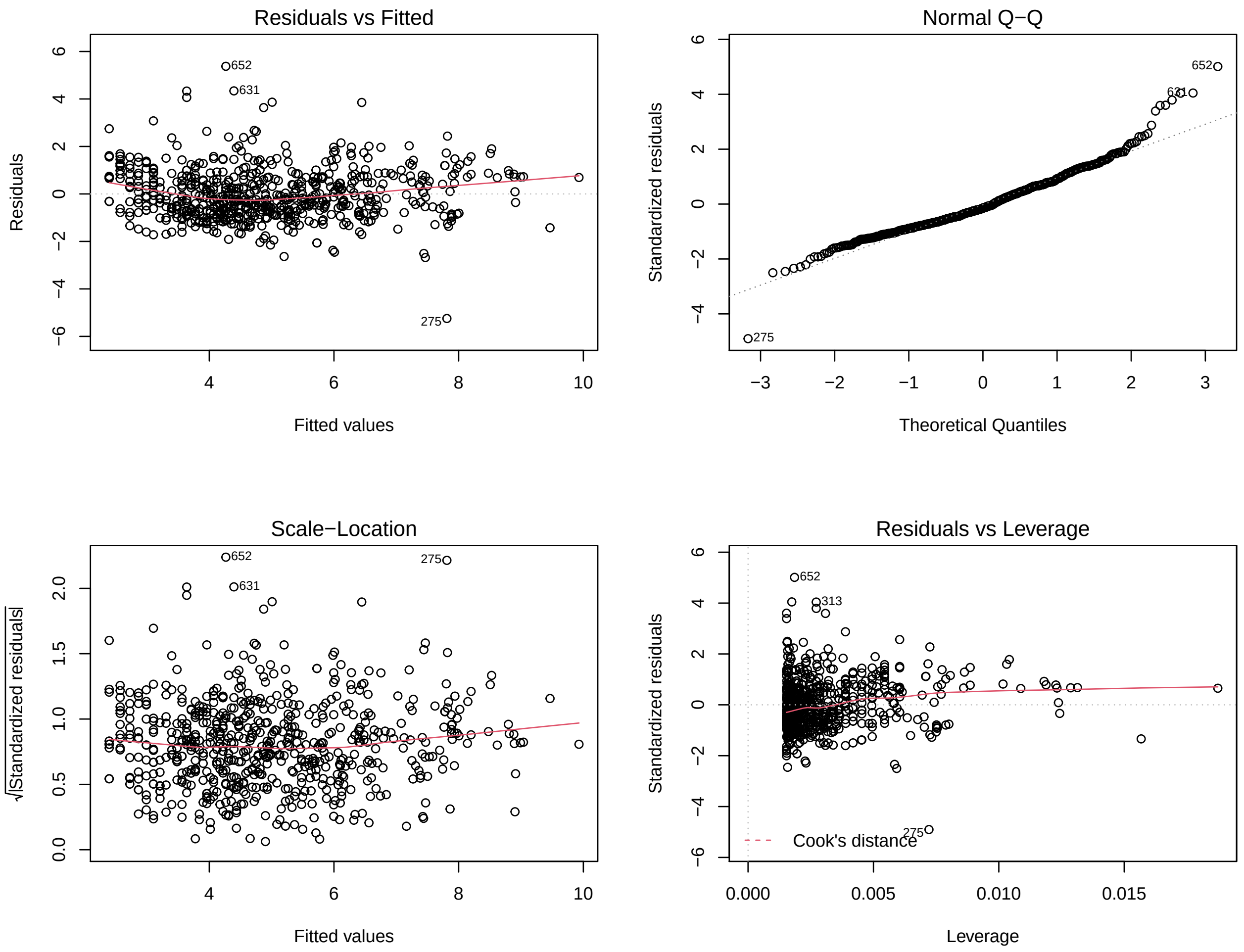
We fit a linear regression model:

```
trop.fit <- lm( log.TropI ~ log.TropT, data = data_LOD_subset)
summary(trop.fit)
```

```
##
## Call:
## lm(formula = log.TropI ~ log.TropT, data = data_LOD_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2463 -0.7296 -0.1568  0.6841  5.3756
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1658     0.1505  -1.102    0.271
## log.TropT     1.3160     0.0372  35.379   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 651 degrees of freedom
## Multiple R-squared:  0.6579, Adjusted R-squared:  0.6573
## F-statistic:  1252 on 1 and 651 DF,  p-value: < 2.2e-16
```

And produce residuals plots:

```r
par(mfrow=c(2,2))
plot(trop.fit)
```

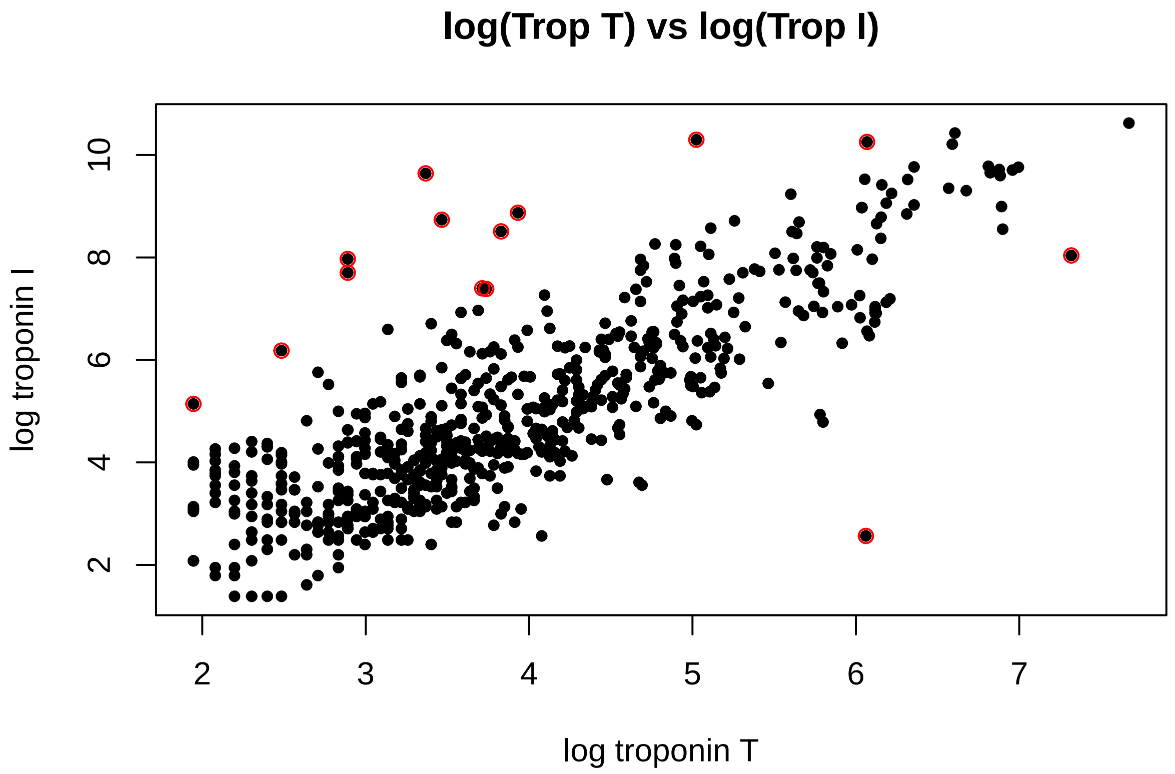


From the plots above, we remove all strong outliers, which are not inline with the overall trend in the data. The removed observations ar shown below and marked in red in the plot:

```r
data_LOD_subset[c(652,313,275,631,635,438,447,582, 461,361,646, 437,112,51),]
```

```
##        tni  tnt                adj log.TropI log.TropT
## 652 15386   29           Type 1 MI  9.641213  3.367296
## 313  2894   18 Myocardial injury  7.970395  2.890372
## 275    13  429           Type 1 MI  2.564949  6.061457
## 631  6223   32           Type 1 MI  8.736007  3.465736
## 635  2213   18 Myocardial injury  7.702104  2.890372
## 438  7137   51           Type 1 MI  8.873048  3.931826
## 447 29679  152           Type 1 MI 10.298195  5.023881
## 582   171    7           Type 1 MI  5.141664  1.945910
## 461   484   12           Type 1 MI  6.182085  2.484907
## 361 28426  432           Type 1 MI 10.255059  6.068426
## 646  4961   46           Type 1 MI  8.509363  3.828641
## 437  1633   41           Type 1 MI  7.398174  3.713572
## 112  1612   42           Type 1 MI  7.385231  3.737670
## 51   3100 1508 Myocardial injury  8.039157  7.318540
```

```
data_LOD_subset2 <-data_LOD_subset[-c(652,313,275,631,635,438,447,582, 461,361,646, 437,112,51),]
row.names(data_LOD_subset2) <- NULL
outliers<-data_LOD_subset[c(652,313,275,631,635,438,447,582, 461,361,646, 437,112,51),]
plot(data_LOD_subset$log.TropT, data_LOD_subset$log.TropI,
     pch = 20,
     main = "log(Trop T) vs log(Trop I)",
     xlab = "log troponin T", ylab = "log troponin I")
points(outliers$log.TropT, outliers$log.TropI, col="red")
```

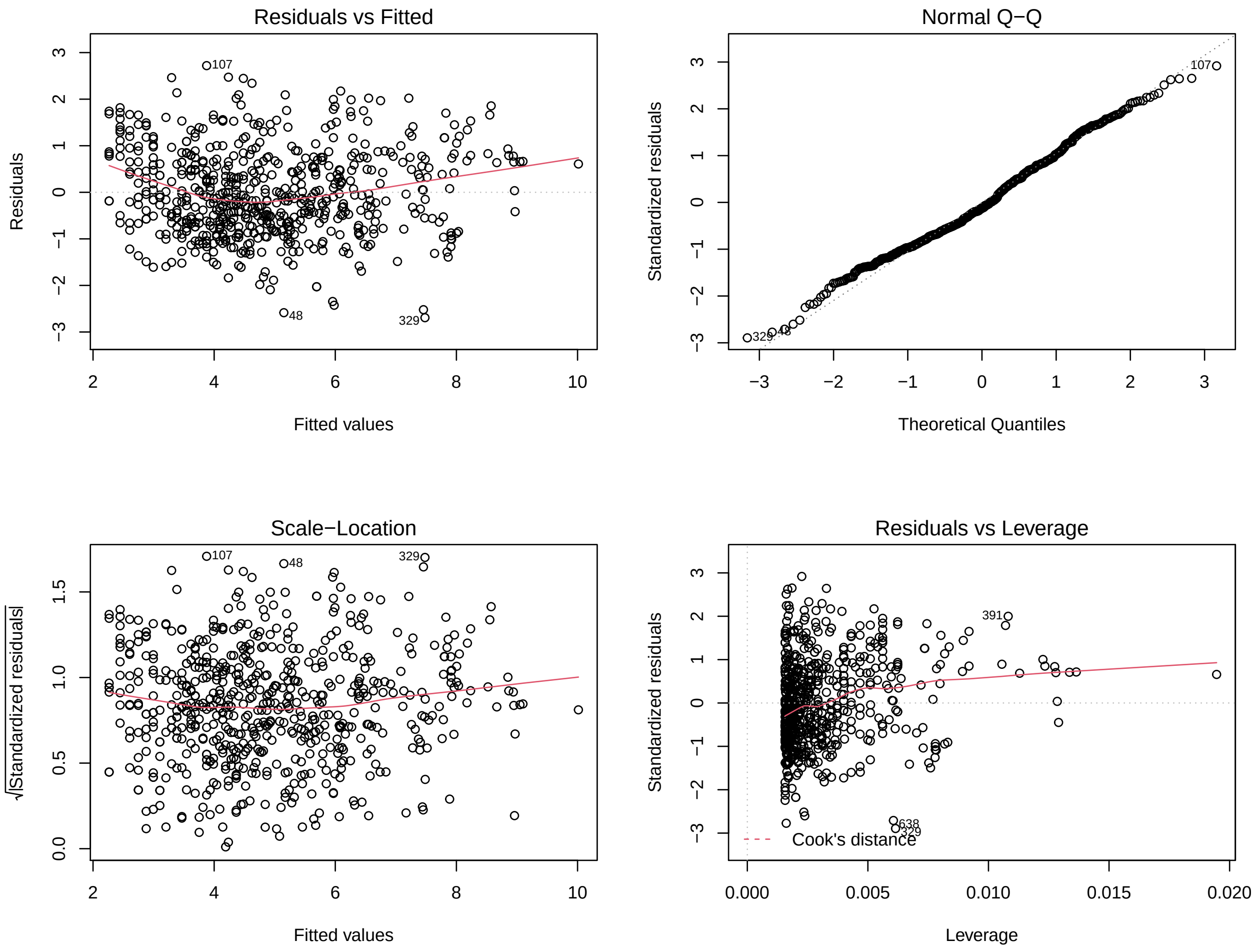


We now re-fit the linear regression model on the reduced dataset and repeat the residual analysis.

```
trop.fit <- lm( log.TropI ~ log.TropT, data = data_LOD_subset2)
summary(trop.fit)
```

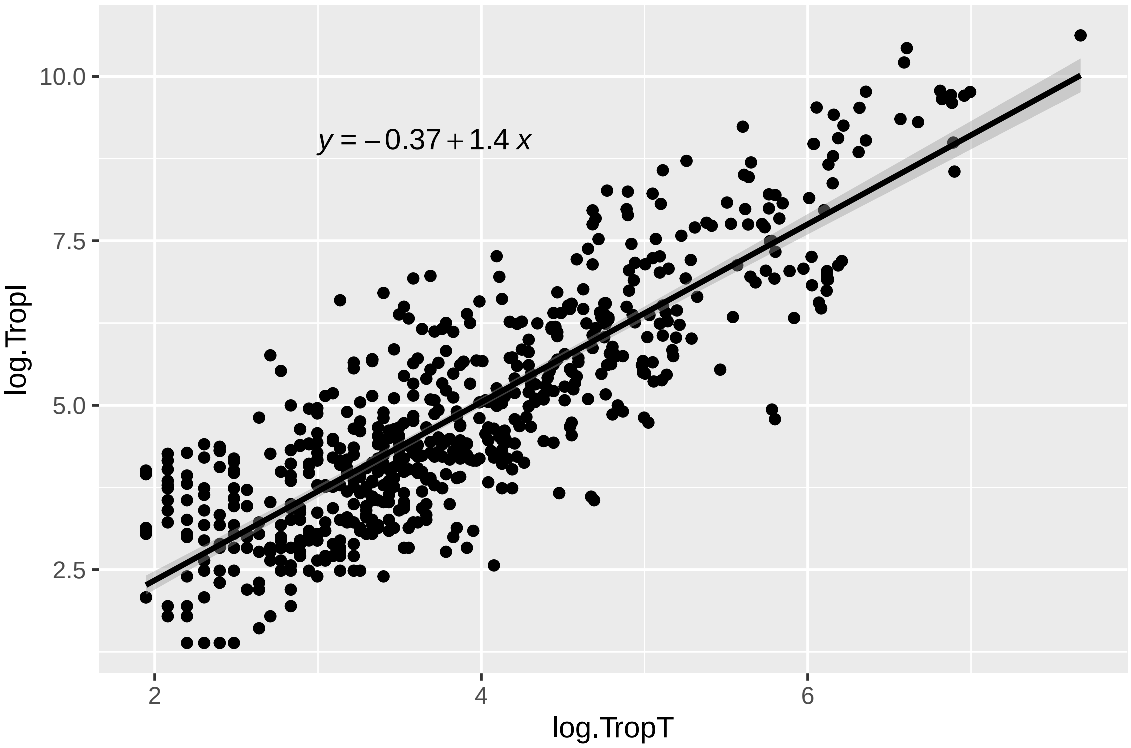
```
##
## Call:
## lm(formula = log.TropI ~ log.TropT, data = data_LOD_subset2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6931 -0.6592 -0.1096  0.6560  2.7199
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.36759    0.13310  -2.762  0.00591 **
## log.TropT    1.35335    0.03294  41.081  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9332 on 637 degrees of freedom
## Multiple R-squared:  0.726,  Adjusted R-squared:  0.7255
## F-statistic:  1688 on 1 and 637 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(trop.fit)
```



Finally, we plot the data along with the fitted regression line, and estimated regression equation.

```
library(ggplot2)
library(ggpubr)

ggplot(data_LOD_subset2, aes(x = log.TropT, y = log.TropI))+ geom_point() +
  geom_smooth(method="lm", col="black") +
  stat_regline_equation(label.x = 3, label.y = 9)
```



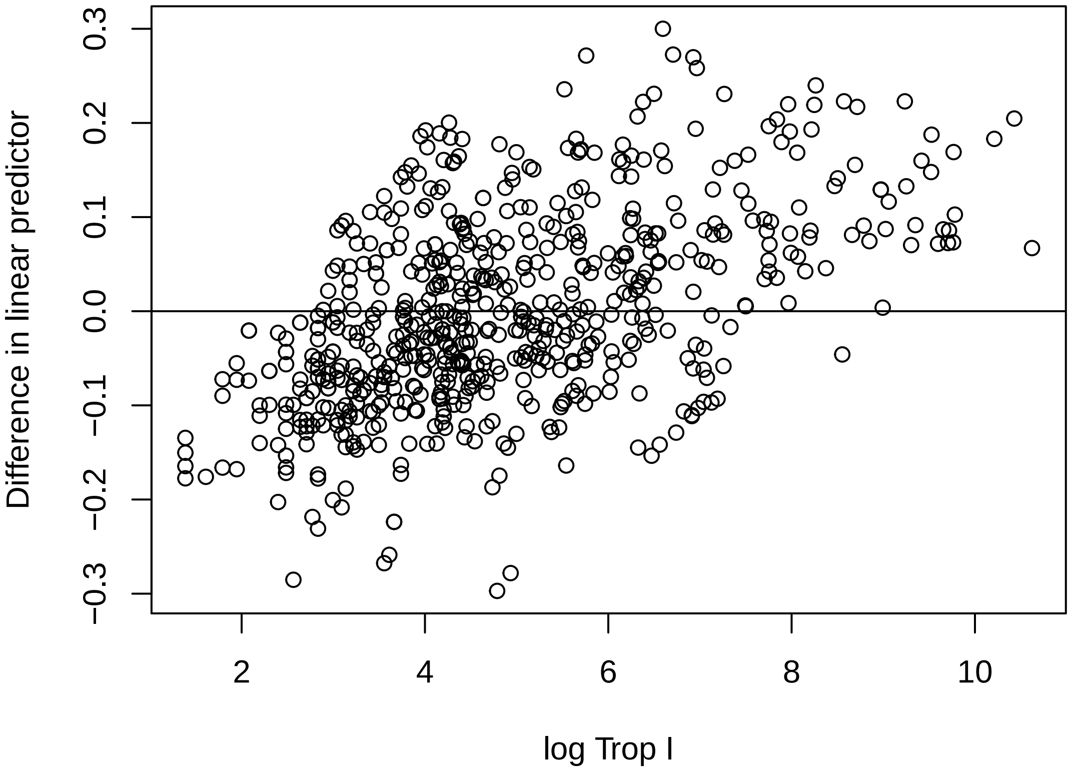
Analysis of predicted values

We compute the predicted log troponin I in our dataset. From this prediction, we can further calculate what will be the difference in the linear predictor of our risk score from using the predicted values rather than the observed ones.

```
# Compute predicted log troponin I:
pred_logI <- predict.lm(trop.fit, data_LOD_subset2)

# Difference in linear predictor
dif_LP <- 0.11030707*(data_LOD_subset2$log.TropI - pred_logI)
```

We produce a scatter plot of the observed log troponin I vs the differences computed above.

```
plot(data_LOD_subset2$log.TropI, dif_LP,
     xlab="log Trop I", ylab = "Difference in linear predictor" )
abline(h = 0)
```



Finally, from the plot above, we expect to slightly underestimate the risk for subjects with small values of troponin I (i.e. trop I $<$ 20) as a result of the negative differences in the residuals. In addition, we expect a slight overestimation of the risk for large values in log trop I (i.e. trop I $>$ 670).

Session info

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggpubr_0.4.0  ggplot2_3.3.5 readr_2.1.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2 xfun_0.30        purrr_0.3.4      lattice_0.20-45
##  [5] splines_4.0.5    carData_3.0-5    colorspace_2.0-3 vctrs_0.4.0
##  [9] generics_0.1.2   htmltools_0.5.2  yaml_2.3.5       mgcv_1.8-40
## [13] utf8_1.2.2       rlang_1.0.2      pillar_1.7.0     glue_1.6.2
## [17] withr_2.5.0      DBI_1.1.2        bit64_4.0.5      lifecycle_1.0.1 
## [21] stringr_1.4.0    munsell_0.5.0    ggsignif_0.6.3   gtable_0.3.0    
## [25] evaluate_0.15    labeling_0.4.2   knitr_1.38       tzdb_0.3.0      
## [29] fastmap_1.1.0    parallel_4.0.5   fansi_1.0.3      highr_0.9       
## [33] broom_0.7.12     polynom_1.4-0    scales_1.1.1     backports_1.4.1 
## [37] vroom_1.5.7      abind_1.4-5      farver_2.1.0     bit_4.0.4       
## [41] hms_1.1.1        digest_0.6.29    stringi_1.7.6    rstatix_0.7.0   
## [45] dplyr_1.0.8      grid_4.0.5       cli_3.2.0        tools_4.0.5     
## [49] magrittr_2.0.3   tibble_3.1.6     crayon_1.5.1     tidyr_1.2.0     
## [53] car_3.0-12       pkgconfig_2.0.3  Matrix_1.4-1     ellipsis_0.3.2  
## [57] assertthat_0.2.1 rmarkdown_2.13   rstudioapi_0.13  R6_2.5.1        
## [61] nlme_3.1-157     compiler_4.0.5
```

References

“ARCHITECT Stat High Sensitivity Troponin-I.” https://www.accessdata.fda.gov/cdrh_docs/pdf19/K191595.pdf.