In-Class activity 5 - Group Edith

2023-04-04

```
library(DAAG) # it cointains AIS dataset and CVlm (Cross-Validation for Linear Regression)
```

```
head(ais, n=10)
```

```
##     rcc wcc   hc   hg ferr   bmi   ssf pcBfat   lbm    ht   wt sex  sport
## 1  3.96 7.5 37.5 12.3   60 20.56 109.1  19.75 63.32 195.9 78.9   f B_Ball
## 2  4.41 8.3 38.2 12.7   68 20.67 102.8  21.30 58.55 189.7 74.4   f B_Ball
## 3  4.14 5.0 36.4 11.6   21 21.86 104.6  19.88 55.36 177.8 69.1   f B_Ball
## 4  4.11 5.3 37.3 12.6   69 21.88 126.4  23.66 57.18 185.0 74.9   f B_Ball
## 5  4.45 6.8 41.5 14.0   29 18.96  80.3  17.64 53.20 184.6 64.6   f B_Ball
## 6  4.10 4.4 37.4 12.5   42 21.04  75.2  15.58 53.77 174.0 63.7   f B_Ball
## 7  4.31 5.3 39.6 12.8   73 21.69  87.2  19.99 60.17 186.2 75.2   f B_Ball
## 8  4.42 5.7 39.9 13.2   44 20.62  97.9  22.43 48.33 173.8 62.3   f B_Ball
## 9  4.30 8.9 41.1 13.5   41 22.64  75.1  17.95 54.57 171.4 66.5   f B_Ball
## 10 4.51 4.4 41.6 12.7   44 19.44  65.1  15.07 53.42 179.9 62.9   f B_Ball
```

```
library(e1071) # it includes function to compute skewness
library(plyr) # it allows to wrangle data
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:DAAG':
##
##     ozone
```

```
library(ggplot2) # it allows to create a number of different types of plots
```

```
colSums(is.na(ais))
```

```
##    rcc    wcc     hc     hg   ferr    bmi    ssf pcBfat    lbm     ht     wt
##      0      0      0      0      0      0      0      0      0      0      0
##    sex  sport
##      0      0
```

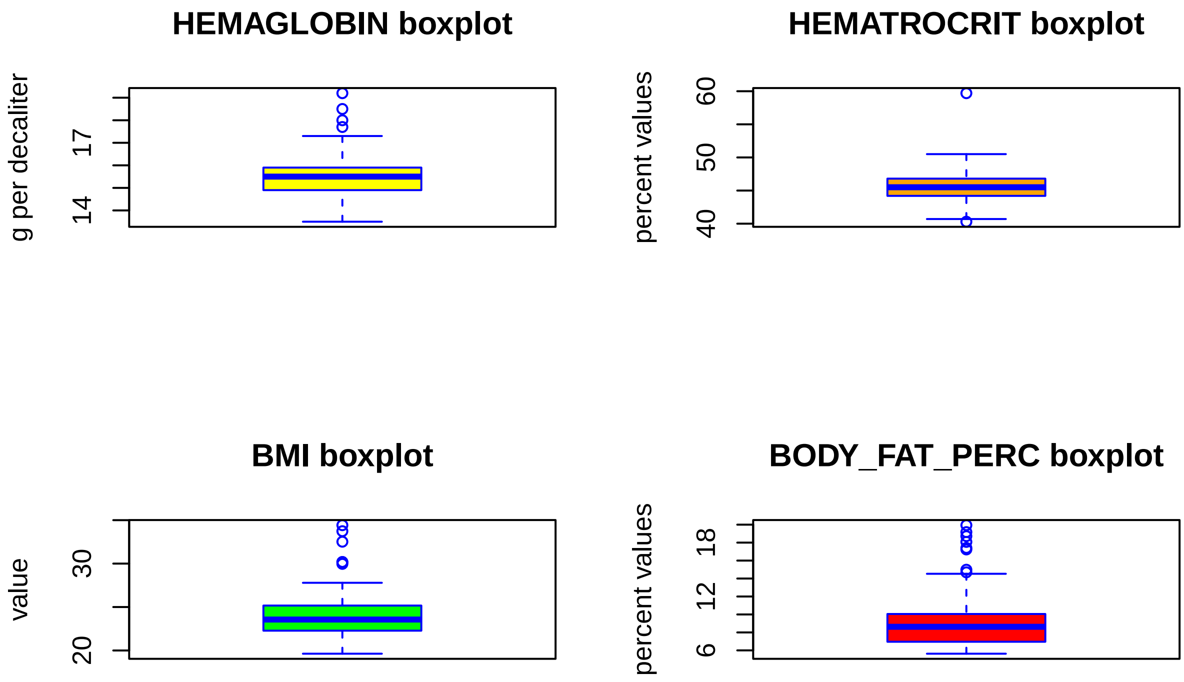
```
ais2 <- subset(ais, sex=="m") # only male athletes
ais3 = ais2[,c(3,4,6,8)] # subset column number that correspond to "hg", "hc", "bmi" and "pcBfat"
newdata <- rename(ais3, c("hg"="HEMAGLOBIN", "hc"="HEMATOCRIT", "bmi"="BMI", "pcBfat"="BODY_FAT_PERC"))
str(newdata)
```

```
## 'data.frame':    102 obs. of  4 variables:
##  $ HEMATOCRIT   : num  46.8 45.2 46.6 44.9 46.1 45.1 47.5 45.5 48.6 44.9 ...
##  $ HEMAGLOBIN   : num  15.9 15.2 15.9 15 15.6 15.2 16.3 15.2 16.5 15.4 ...
##  $ BMI          : num  22.5 23.9 23.7 23.1 22.3 ...
##  $ BODY_FAT_PERC: num  8.47 7.68 6.16 8.56 6.86 ...
```

```
summary(newdata)
```

```
##    HEMATOCRIT      HEMAGLOBIN         BMI         BODY_FAT_PERC
##  Min.   :40.30   Min.   :13.50   Min.   :19.63   Min.   : 5.630
##  1st Qu.:44.23   1st Qu.:14.93   1st Qu.:22.29   1st Qu.: 6.968
##  Median :45.50   Median :15.50   Median :23.56   Median : 8.625
##  Mean   :45.65   Mean   :15.55   Mean   :23.90   Mean   : 9.251
##  3rd Qu.:46.80   3rd Qu.:15.90   3rd Qu.:25.16   3rd Qu.:10.010
##  Max.   :59.70   Max.   :19.20   Max.   :34.42   Max.   :19.940
```

```
par(mfrow=c(2, 2))  # it divides graph area in two parts

boxplot(newdata$HEMAGLOBIN, col = "yellow", border="blue",
        main = "HEMAGLOBIN boxplot",
        ylab = "g per decaliter")

boxplot(newdata$HEMATOCRIT, col = "orange", border="blue",
        main = "HEMATROCRIT boxplot",
        ylab = "percent values")

boxplot(newdata$BMI, col = "green", border="blue",
        main = "BMI boxplot",
        ylab = "value")

boxplot(newdata$BODY_FAT_PERC, col = "red", border="blue",
        main = "BODY_FAT_PERC boxplot",
        ylab = "percent values")
```



```
boxplot.stats(newdata$HEMAGLOBIN)$out # HEMAGLOBIN outliers
```

```
## [1] 18.0 19.2 18.5 17.7
```

```
boxplot.stats(newdata$HEMATOCRIT)$out #HEMATOCRIT outliers
```

```
## [1] 40.3 59.7
```

```
boxplot.stats(newdata$BMI)$out #BMI outliers
```

```
## [1] 29.97 32.52 30.18 34.42 33.73 30.18
```

```
boxplot.stats(newdata$BODY_FAT_PERC)$out #BODY_FAT_PERC outliers
```

```
## [1] 19.94 17.41 18.08 18.72 19.17 17.24 14.69 14.98
```

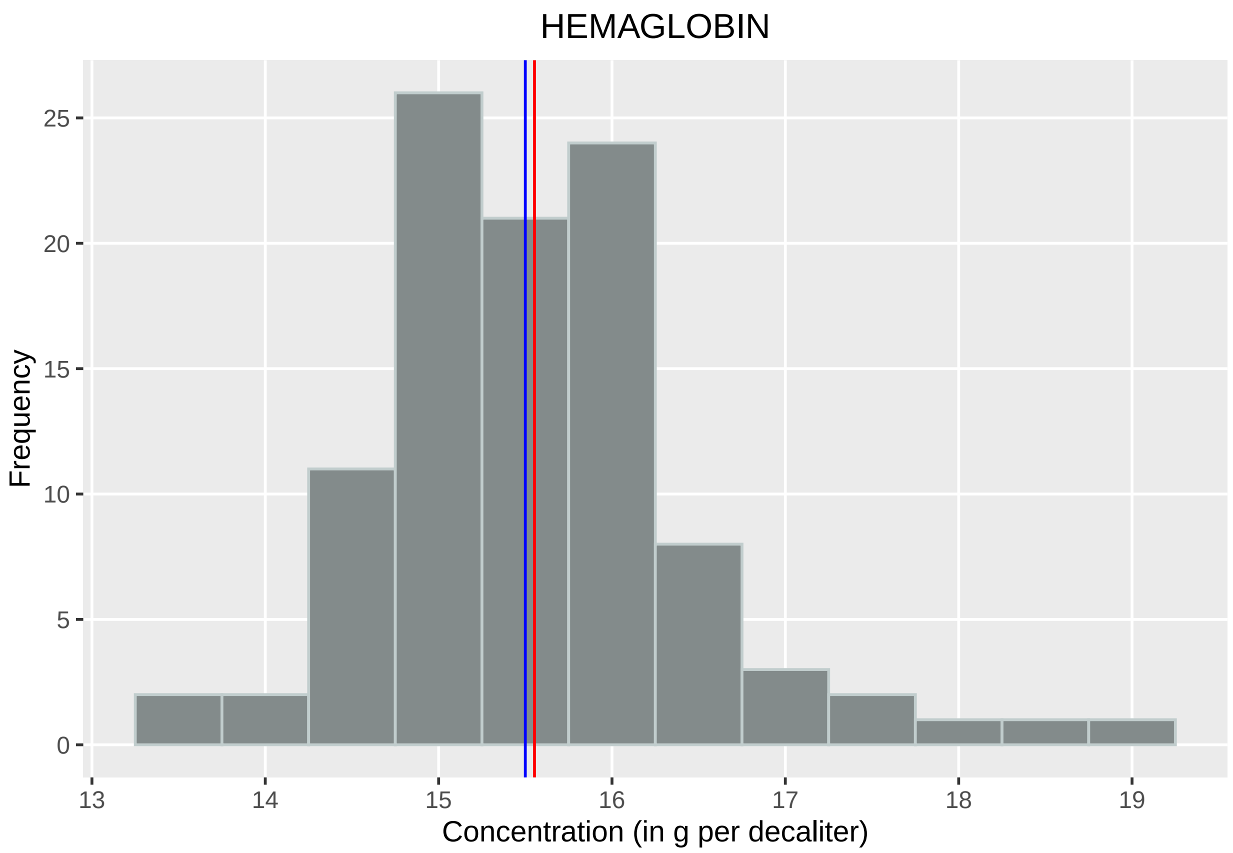
```
# Histogram of HEMAGLOBIN
qplot(HEMAGLOBIN, data = newdata, geom="histogram", binwidth=0.5,
      fill=I("azure4"), col=I("azure3")) +
  labs(title = "HEMAGLOBIN") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x ="Concentration (in g per decaliter)") +
  labs(y = "Frequency") +
  scale_y_continuous(breaks = c(0,5,10,15,20,25,30,35,40,45,50), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(10:25), minor_breaks = NULL) +
  geom_vline(xintercept = mean(newdata$HEMAGLOBIN), show_guide=TRUE, color
             ="red", labels="Average") +
  geom_vline(xintercept = median(newdata$HEMAGLOBIN), show_guide=TRUE, color
             ="blue", labels="Median")
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `show_guide` argument of `layer()` is deprecated as of ggplot2 2.0.0.
## i Please use the `show.legend` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

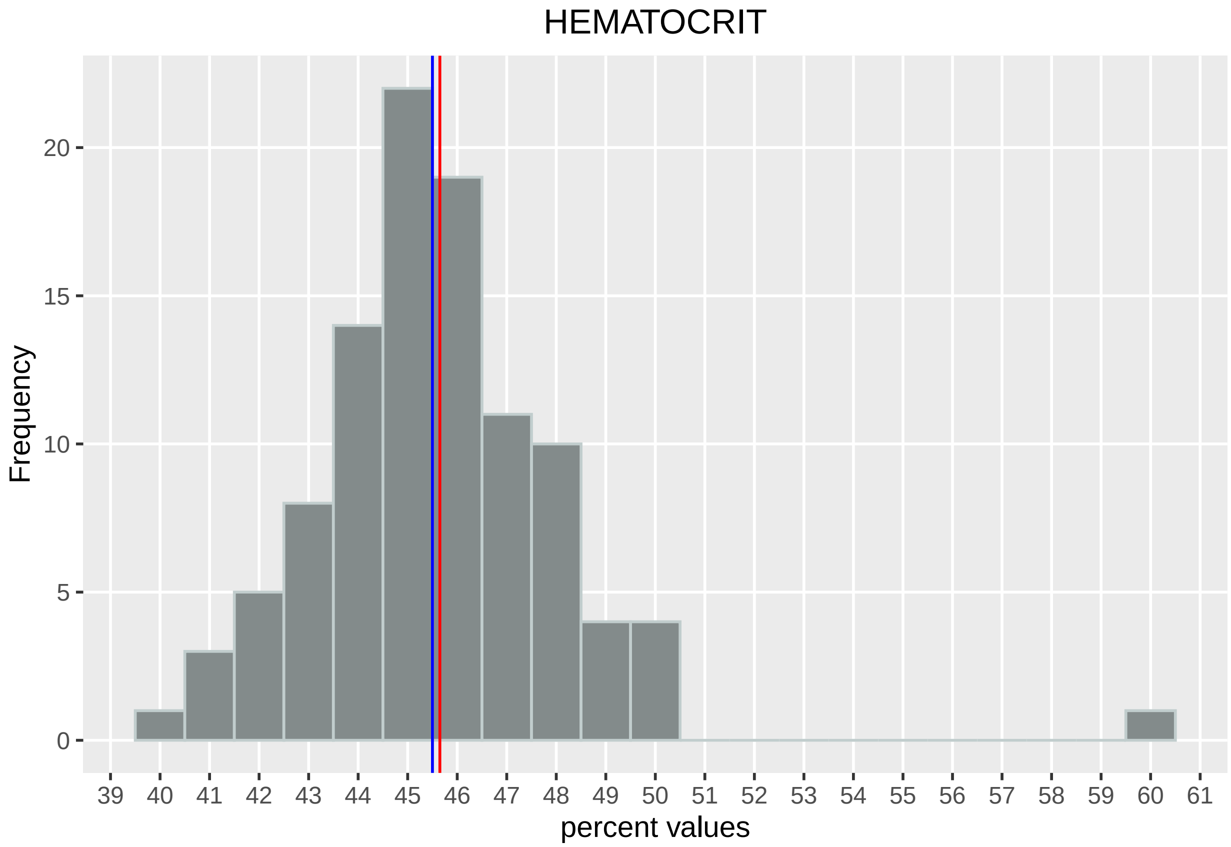
```
## Warning in geom_vline(xintercept = mean(newdata$HEMAGLOBIN), show_guide = TRUE,
## : Ignoring unknown parameters: `labels`
```

```
## Warning in geom_vline(xintercept = median(newdata$HEMAGLOBIN), show_guide =
## TRUE, : Ignoring unknown parameters: `labels`
```



```
# Histogram of HEMATOCRIT
qplot(HEMATOCRIT, data = newdata, geom="histogram", binwidth=1,
      fill=I("azure4"), col=I("azure3")) +
  labs(title = "HEMATOCRIT") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x ="percent values") +
  labs(y = "Frequency") +
  scale_y_continuous(breaks = c(0,5,10,15,20,25), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(30:65), minor_breaks = NULL) +
  geom_vline(xintercept = mean(newdata$HEMATOCRIT), show_guide=TRUE, color
            ="red", labels="Average") +
  geom_vline(xintercept = median(newdata$HEMATOCRIT), show_guide=TRUE, color
            ="blue", labels="Median")
```

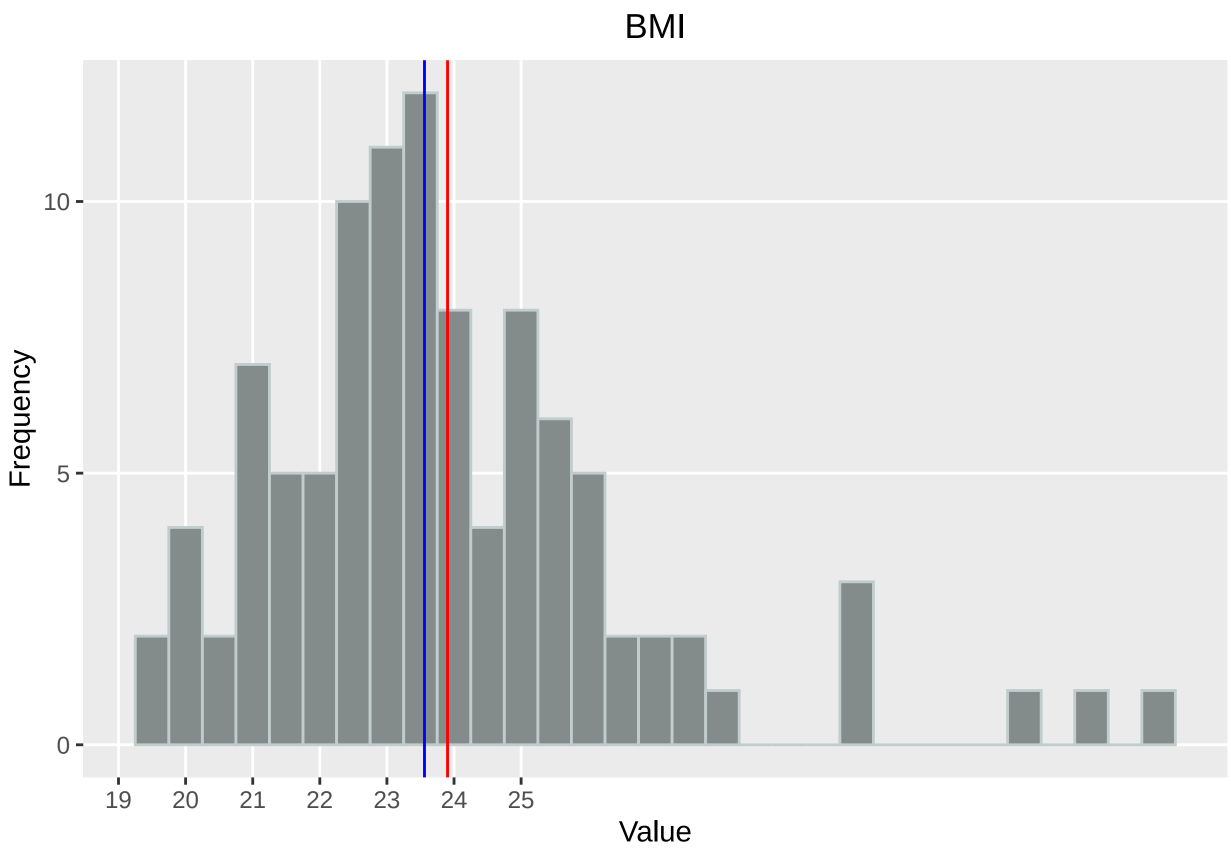
```
## Warning in geom_vline(xintercept = mean(newdata$HEMATOCRIT), show_guide = TRUE,
## : Ignoring unknown parameters: `labels`

## Warning in geom_vline(xintercept = median(newdata$HEMATOCRIT), show_guide =
## TRUE, : Ignoring unknown parameters: `labels`
```



```r
# Histogram of BMI
qplot(BMI, data = newdata, geom="histogram", binwidth=0.5,
      fill=I("azure4"), col=I("azure3")) +
  labs(title = "BMI") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x ="Value") +
  labs(y = "Frequency") +
  scale_y_continuous(breaks = c(0,5,10,15,20,25,30,35,40,45,50), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(10:25), minor_breaks = NULL) +
  geom_vline(xintercept = mean(newdata$BMI), show_guide=TRUE, color
             ="red", labels="Average") +
  geom_vline(xintercept = median(newdata$BMI), show_guide=TRUE, color
             ="blue", labels="Median")
```

```
## Warning in geom_vline(xintercept = mean(newdata$BMI), show_guide = TRUE, :
## Ignoring unknown parameters: `labels`
```

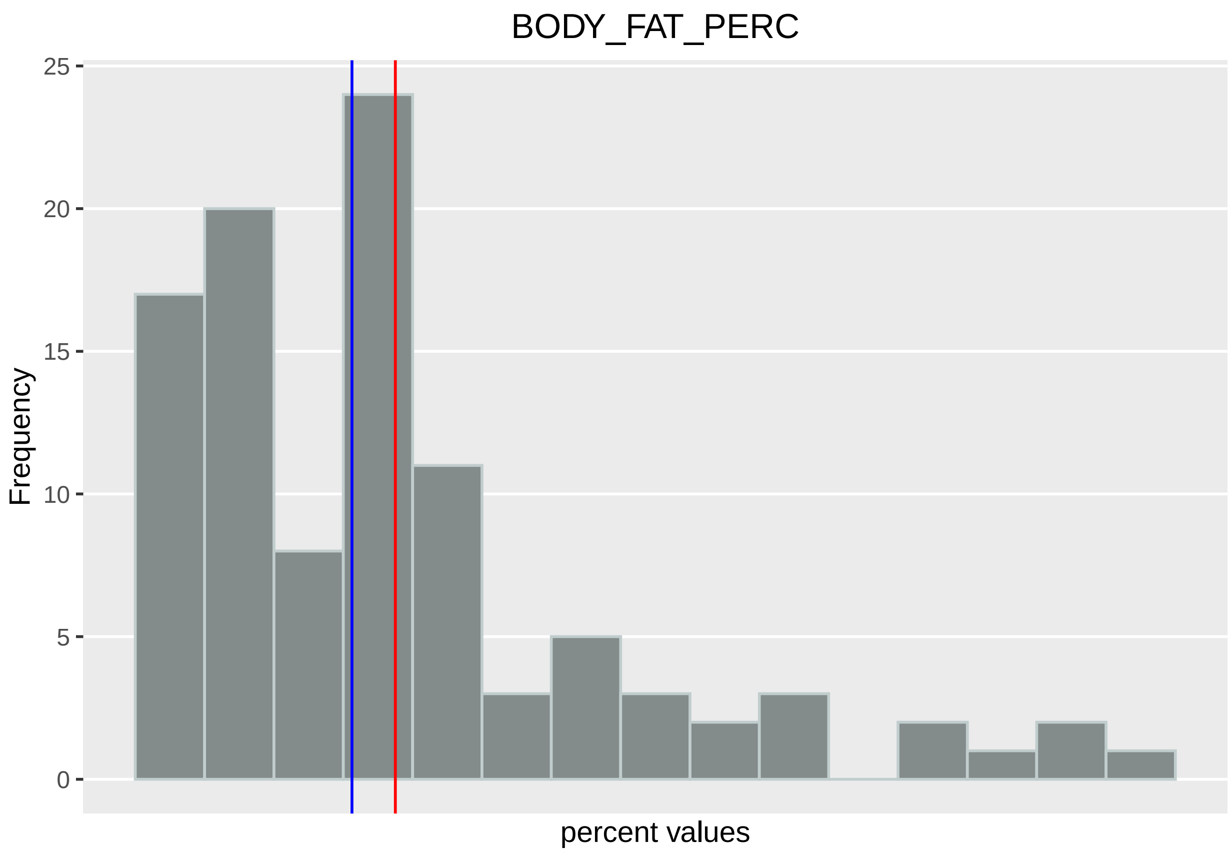
```
## Warning in geom_vline(xintercept = median(newdata$BMI), show_guide = TRUE, :
## Ignoring unknown parameters: `labels`
```



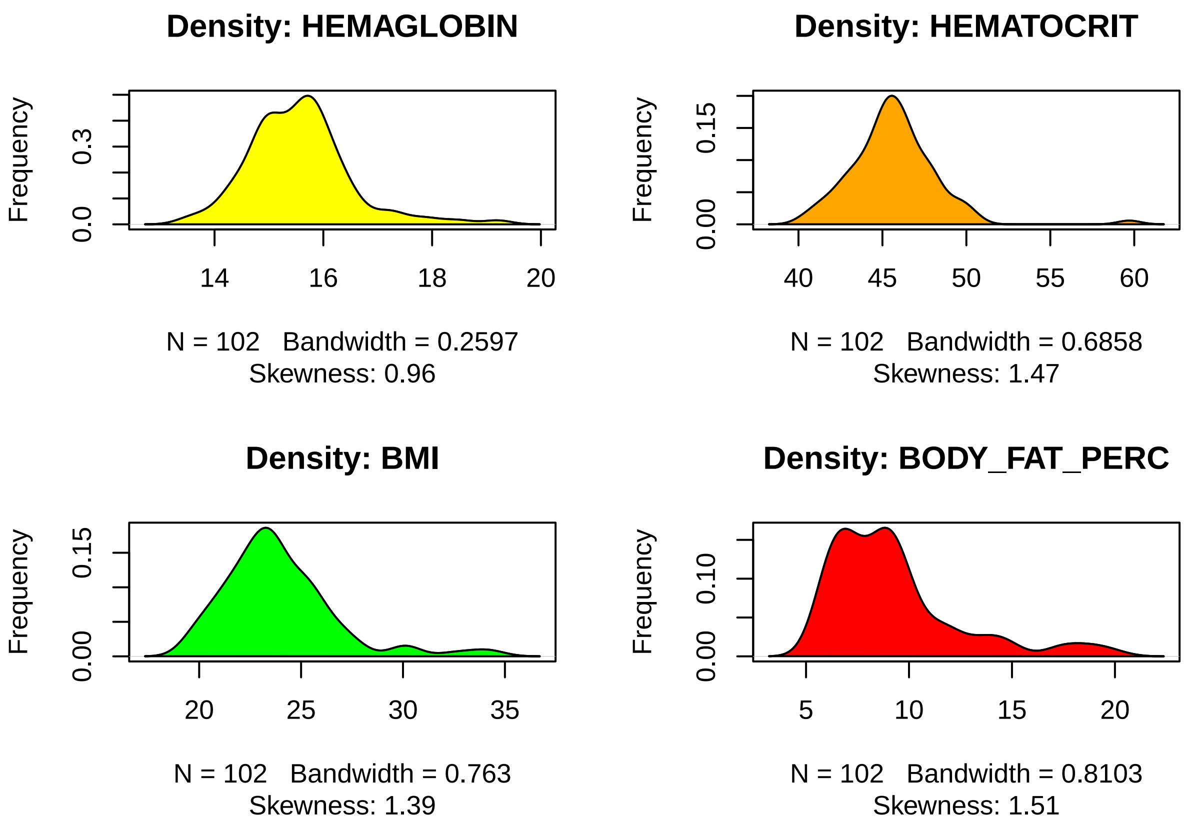
```
# Histogram of BODY_FAT_PERC
qplot(BODY_FAT_PERC, data = newdata, geom="histogram", binwidth=1,
      fill=I("azure4"), col=I("azure3")) +
  labs(title = "BODY_FAT_PERC") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x ="percent values") +
  labs(y = "Frequency") +
  scale_y_continuous(breaks = c(0,5,10,15,20,25), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(30:65), minor_breaks = NULL) +
  geom_vline(xintercept = mean(newdata$BODY_FAT_PERC), show_guide=TRUE, color
             ="red", labels="Average") +
  geom_vline(xintercept = median(newdata$BODY_FAT_PERC), show_guide=TRUE, color
             ="blue", labels="Median")
```

```
## Warning in geom_vline(xintercept = mean(newdata$BODY_FAT_PERC), show_guide =
## TRUE, : Ignoring unknown parameters: `labels`
```

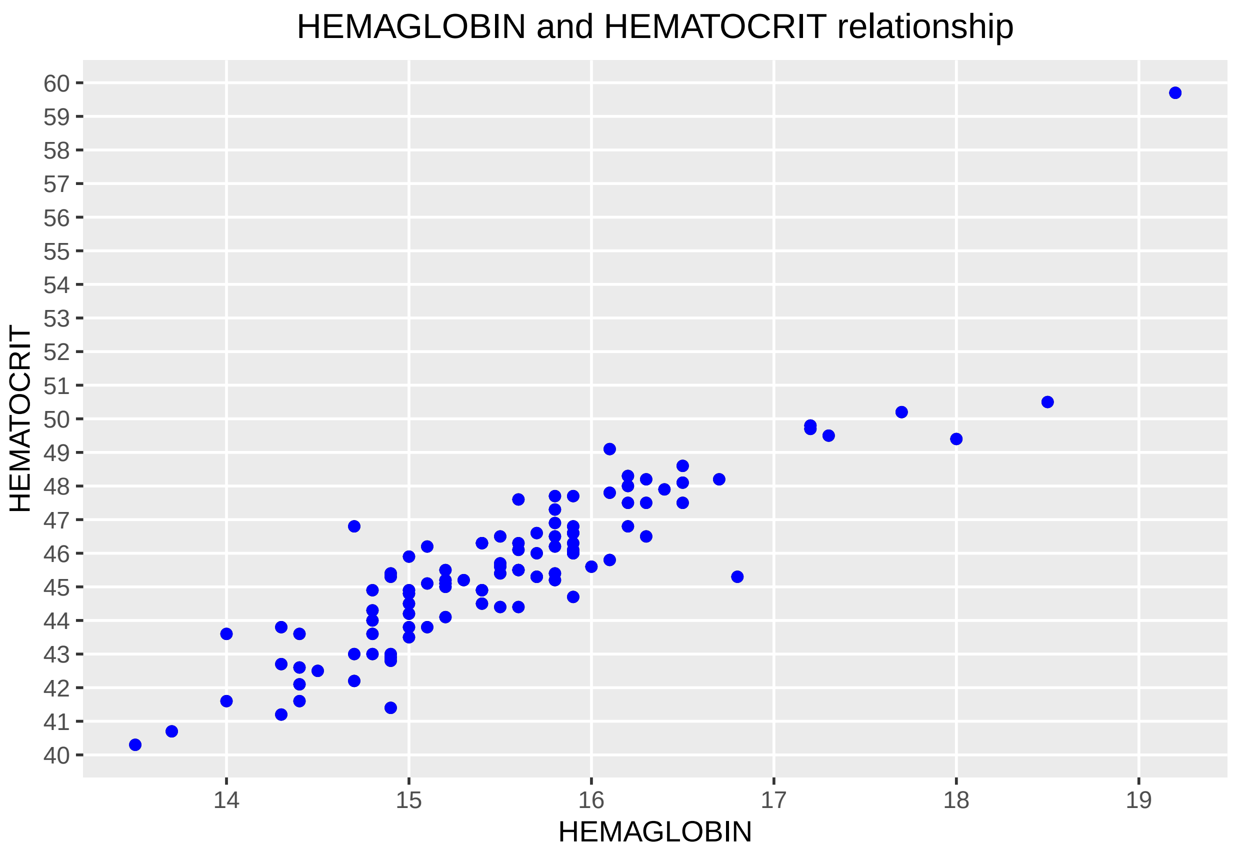
```
## Warning in geom_vline(xintercept = median(newdata$BODY_FAT_PERC), show_guide =
## TRUE, : Ignoring unknown parameters: `labels`
```



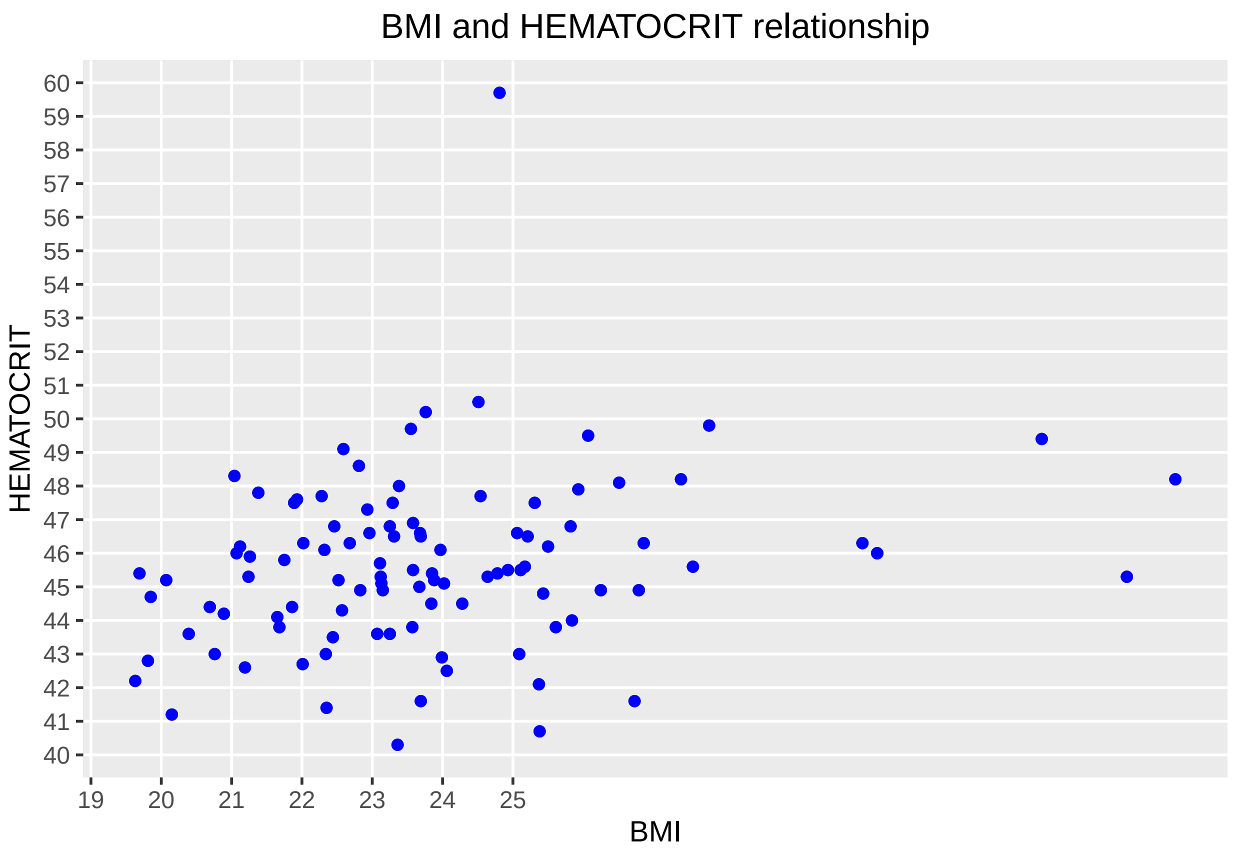
```
par(mfrow=c(2, 2))  # it divides graph area in two parts

plot(density(newdata$HEMAGLOBIN), main="Density: HEMAGLOBIN", ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(newdata$HEMAGLOBIN), 2)))
     polygon(density(newdata$HEMAGLOBIN), col="yellow")

plot(density(newdata$HEMATOCRIT), main="Density: HEMATOCRIT", ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(newdata$HEMATOCRIT), 2)))
     polygon(density(newdata$HEMATOCRIT), col="orange")

plot(density(newdata$BMI), main="Density: BMI", ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(newdata$BMI), 2)))
     polygon(density(newdata$BMI), col="green")

plot(density(newdata$BODY_FAT_PERC), main="Density: BODY_FAT_PERC", ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(newdata$BODY_FAT_PERC), 2)))
     polygon(density(newdata$BODY_FAT_PERC), col="red")
```



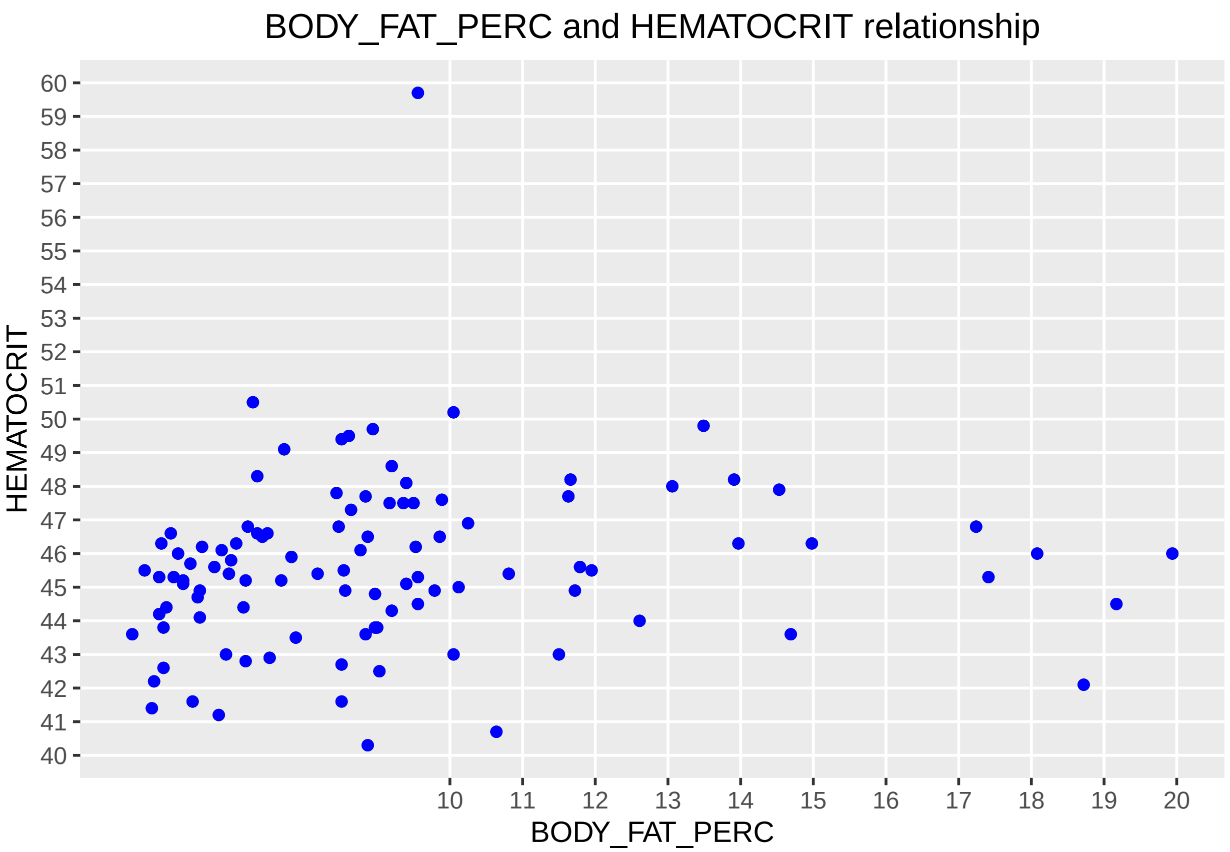
```r
qplot(HEMAGLOBIN, HEMATOCRIT, data = newdata,
      main = "HEMAGLOBIN and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



```
qplot(BMI, HEMATOCRIT, data = newdata,
      main = "BMI and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



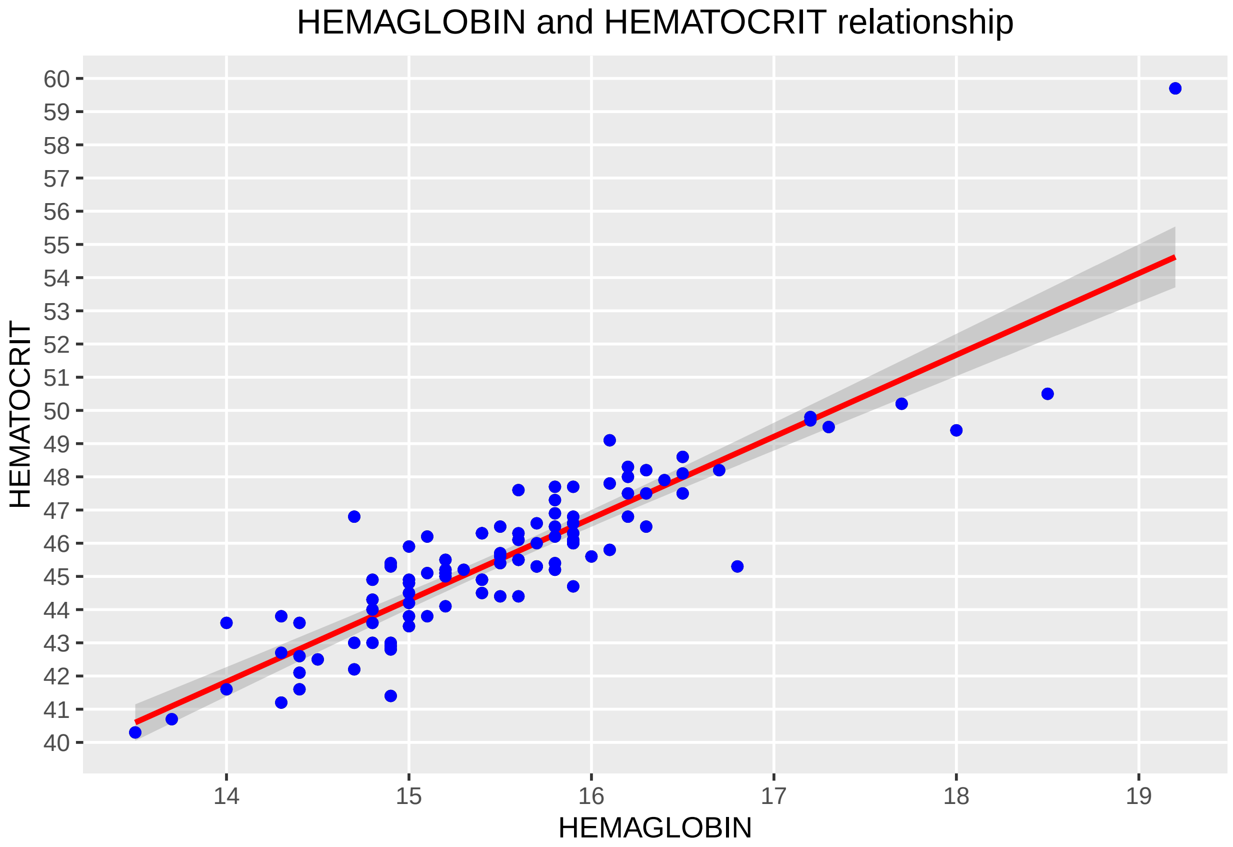
```
qplot(BODY_FAT_PERC, HEMATOCRIT, data = newdata,
      main = "BODY_FAT_PERC and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



```
# Show the relationship creating a regression line
qplot(HEMAGLOBIN, HEMATOCRIT, data = newdata,
      main = "HEMAGLOBIN and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      stat_smooth(method="lm", col="red", size=1) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```

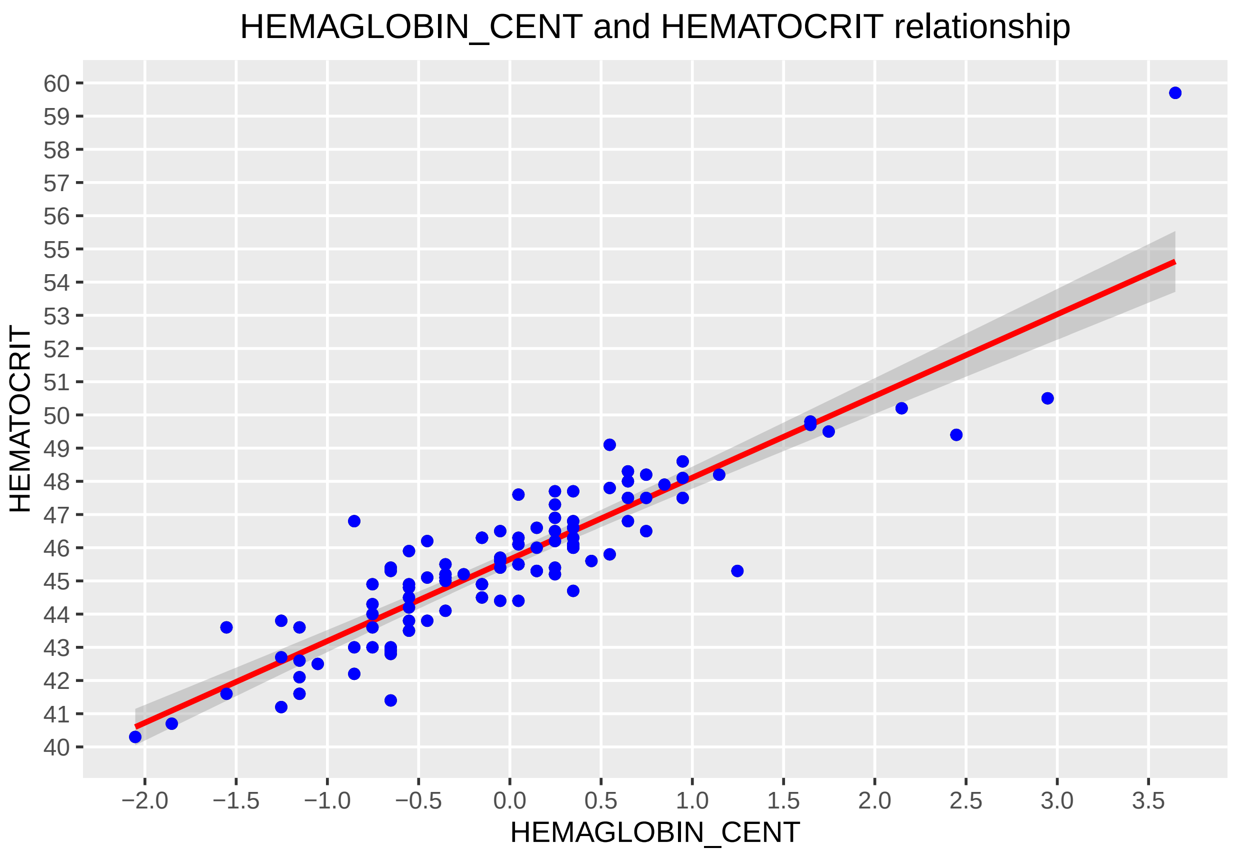
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



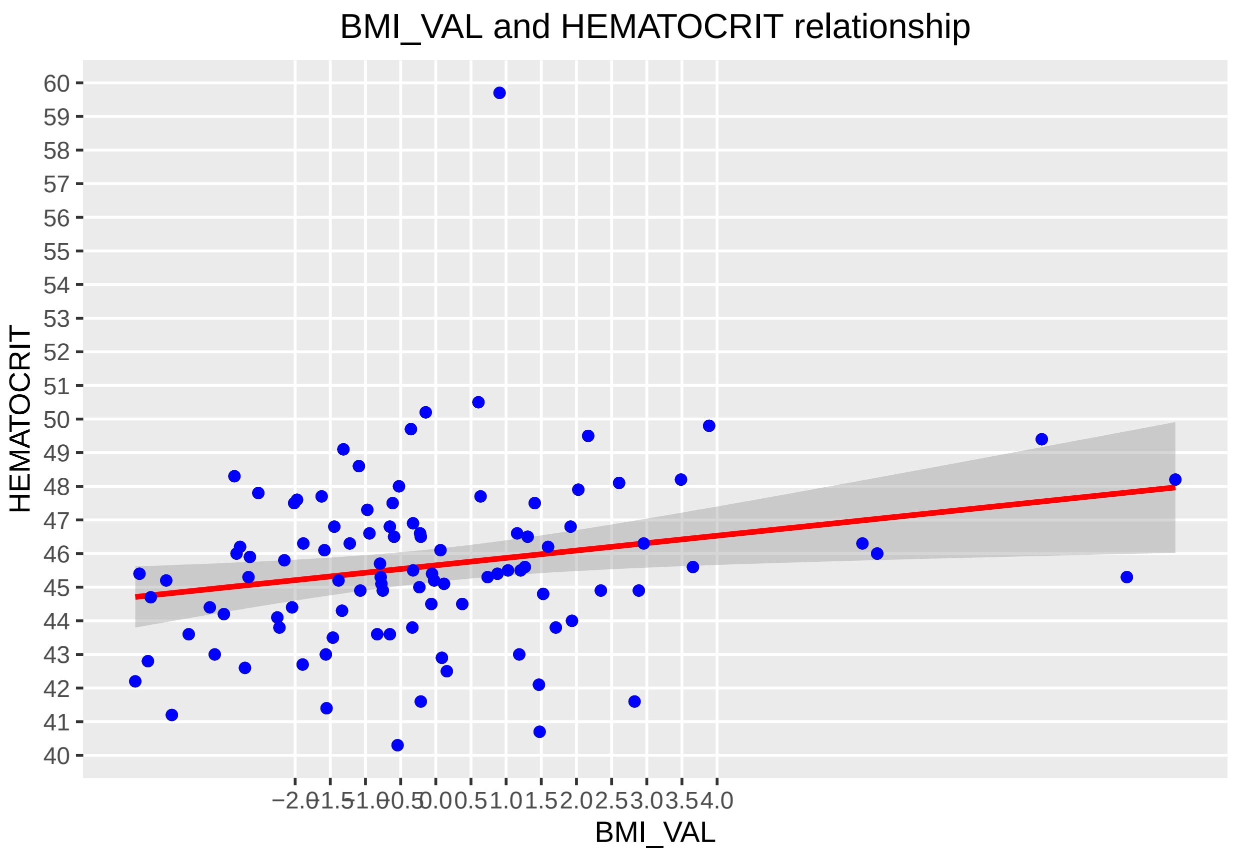
```
set.seed(123) # setting seed to reproduce results of random sampling
HEMAGLOBIN_CENT = scale(newdata$HEMAGLOBIN, center=TRUE, scale=FALSE) # center the variable
# Show the relationship with new variable centered, creating a regression line
qplot(HEMAGLOBIN_CENT, HEMATOCRIT, data = newdata,
      main = "HEMAGLOBIN_CENT and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      stat_smooth(method="lm", col="red", size=1) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2,2.5,3,3.5,4), minor_breaks = NULL)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



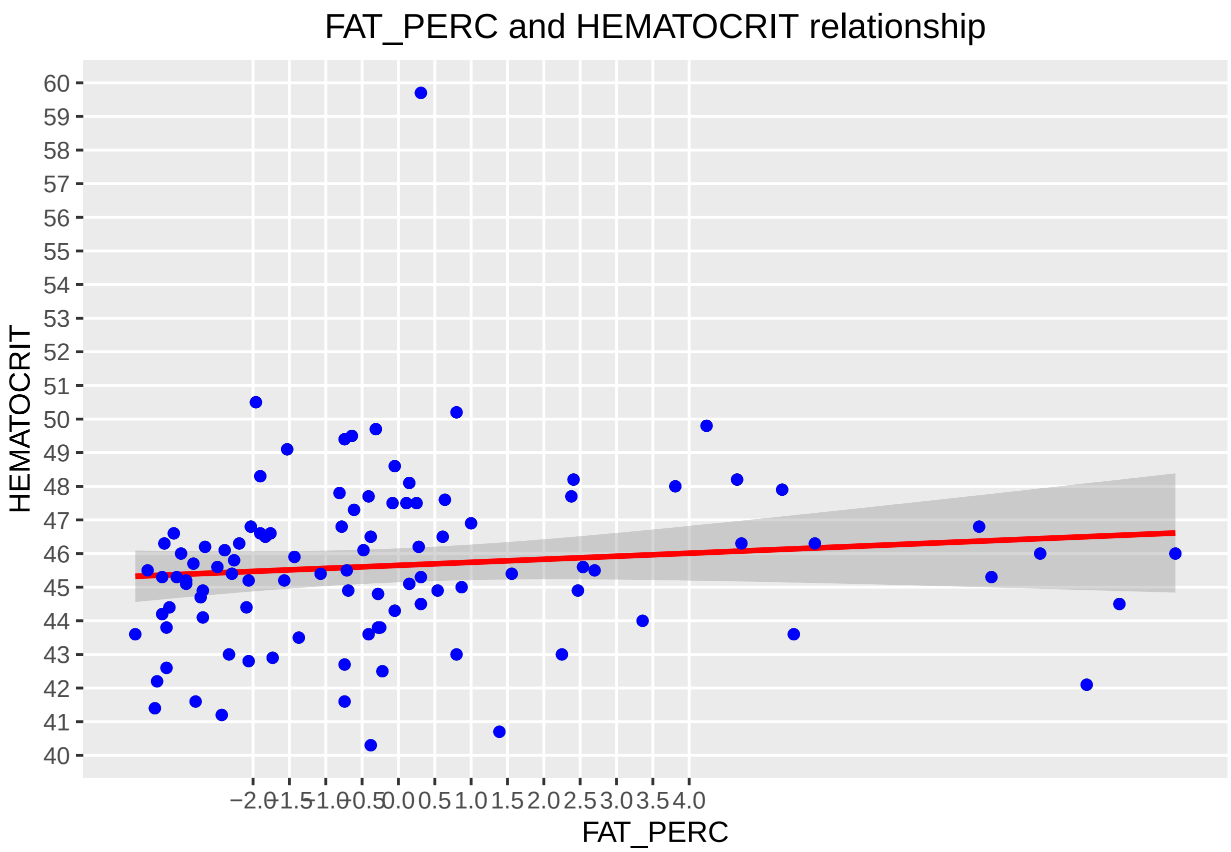
```
set.seed(123) # setting seed to reproduce results of random sampling
BMI_VAL = scale(newdata$BMI, center=TRUE, scale=FALSE) # center the variable
# Show the relationship with new variable centered, creating a regression line
qplot(BMI_VAL, HEMATOCRIT, data = newdata,
      main = "BMI_VAL and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      stat_smooth(method="lm", col="red", size=1) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2,2.5,3,3.5,4), minor_breaks = NULL)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
set.seed(123) # setting seed to reproduce results of random sampling
FAT_PERC = scale(newdata$BODY_FAT_PERC, center=TRUE, scale=FALSE) # center the variable
# Show the relationship with new variable centered, creating a regression line
qplot(FAT_PERC, HEMATOCRIT, data = newdata,
      main = "FAT_PERC and HEMATOCRIT relationship") +
      theme(plot.title = element_text(hjust = 0.5)) +
      stat_smooth(method="lm", col="red", size=1) +
      geom_point(colour = "blue", size = 1.5) +
      scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
      scale_x_continuous(breaks = c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2,2.5,3,3.5,4), minor_breaks = NULL)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
mod1 = lm(HEMATOCRIT ~ HEMAGLOBIN_CENT+FAT_PERC+BMI_VAL, data = newdata)
summary(mod1)
```

```
##
## Call:
## lm(formula = HEMATOCRIT ~ HEMAGLOBIN_CENT + FAT_PERC + BMI_VAL,
##     data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3523 -0.6691 -0.0266  0.5588  4.9926
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     45.65000    0.11443 398.925   <2e-16 ***
## HEMAGLOBIN_CENT  2.49389    0.12937  19.278   <2e-16 ***
## FAT_PERC         0.04711    0.04662   1.010    0.315
## BMI_VAL         -0.05794    0.05577  -1.039    0.301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.156 on 98 degrees of freedom
## Multiple R-squared:  0.8035, Adjusted R-squared:  0.7975
## F-statistic: 133.6 on 3 and 98 DF,  p-value: < 2.2e-16
```

```
modSummary <- summary(mod1)  # capture model summary as an object
modCoeff <- modSummary$coefficients  # model coefficients
```

```
beta.estimate <- modCoeff["HEMAGLOBIN_CENT", "Estimate"]  # get beta coefficient estimate
std.error <- modCoeff["HEMAGLOBIN_CENT", "Std. Error"]  # get standard error
hem_t_value <- beta.estimate/std.error  # calculate t statistic
sprintf(fmt = "%10s is the t-value for HEMAGLOBIN", hem_t_value)
```

```
## [1] "19.2778709228326 is the t-value for HEMAGLOBIN"
```

```
beta.estimate <- modCoeff["FAT_PERC", "Estimate"]  # get beta coefficient estimate
std.error <- modCoeff["FAT_PERC", "Std. Error"]  # get standard error
fat_t_value <- beta.estimate/std.error  # calculate t statistic
sprintf(fmt = "%10s is the t-value for Body Fat Percentage", fat_t_value)
```

```
## [1] "1.01046086146836 is the t-value for Body Fat Percentage"
```

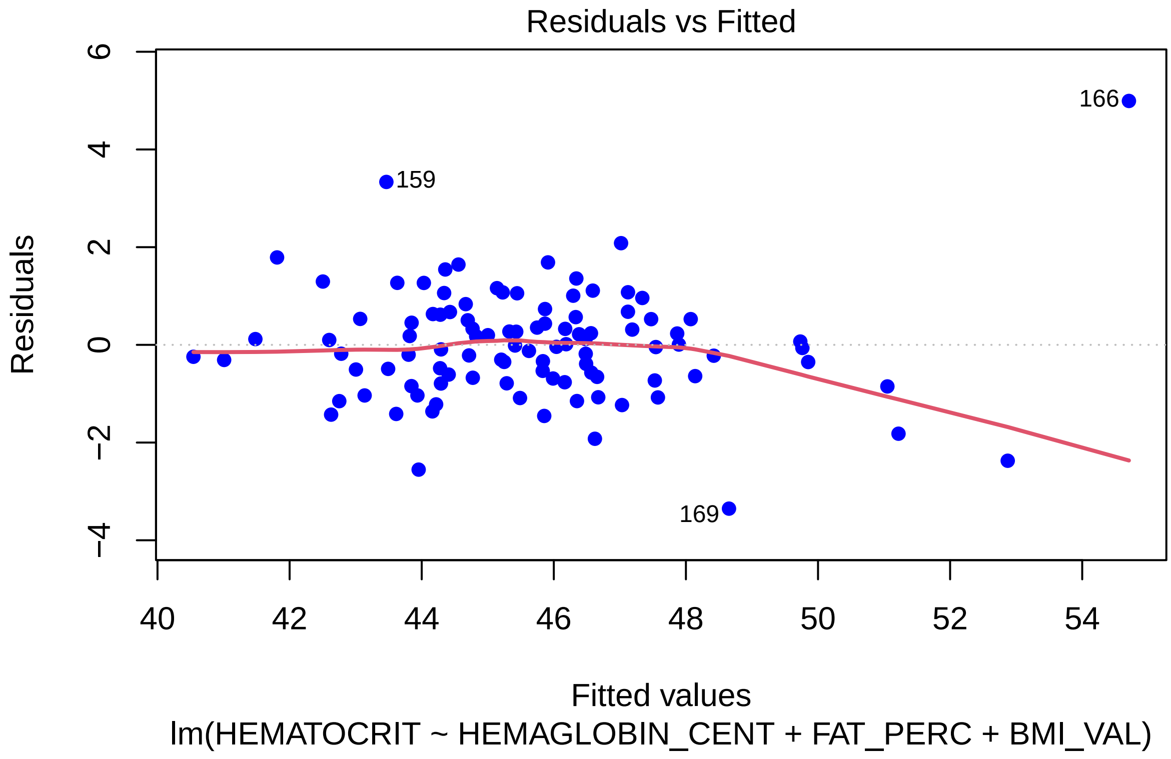
```
beta.estimate <- modCoeff["BMI_VAL", "Estimate"]  # get beta coefficient estimate
std.error <- modCoeff["BMI_VAL", "Std. Error"]  # get standard error
bmi_t_value <- beta.estimate/std.error  # calculate t statistic
sprintf(fmt = "%10s is the t-value for BMI", bmi_t_value)
```

```
## [1] "-1.03883606994576 is the t-value for BMI"
```

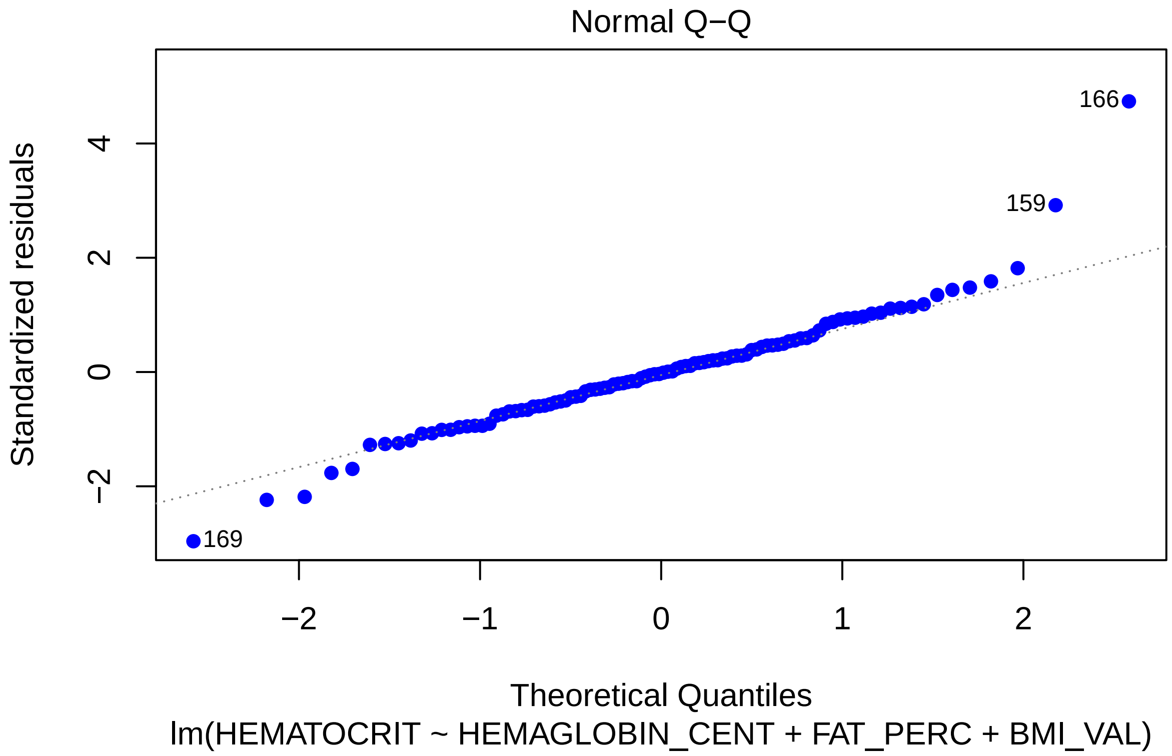
```
f_statistic <- mod1$fstatistic[1]  # calculate F statistic
f <- summary(mod1)$fstatistic  # parameters for model p-value calculation
print(f) # print F value
```

```
##    value    numdf    dendf
## 133.6155   3.0000  98.0000
```

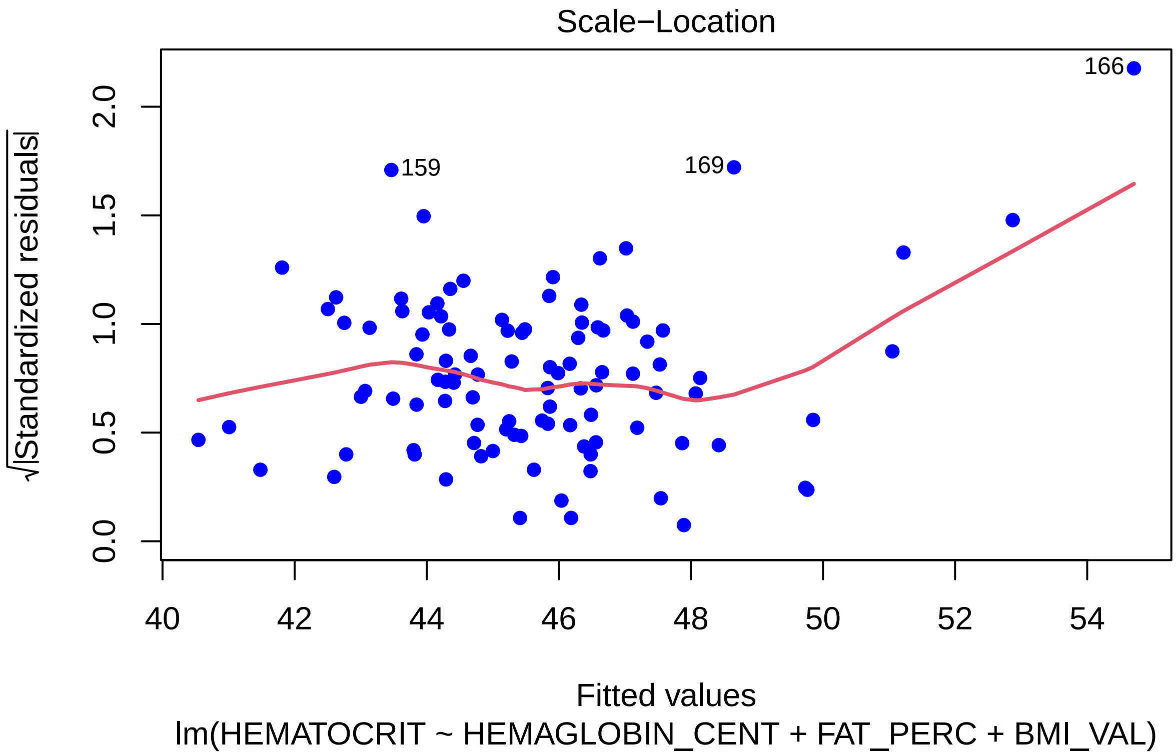
```
plot(mod1, pch=16, col="blue", lty=1, lwd=2, which=1)
```



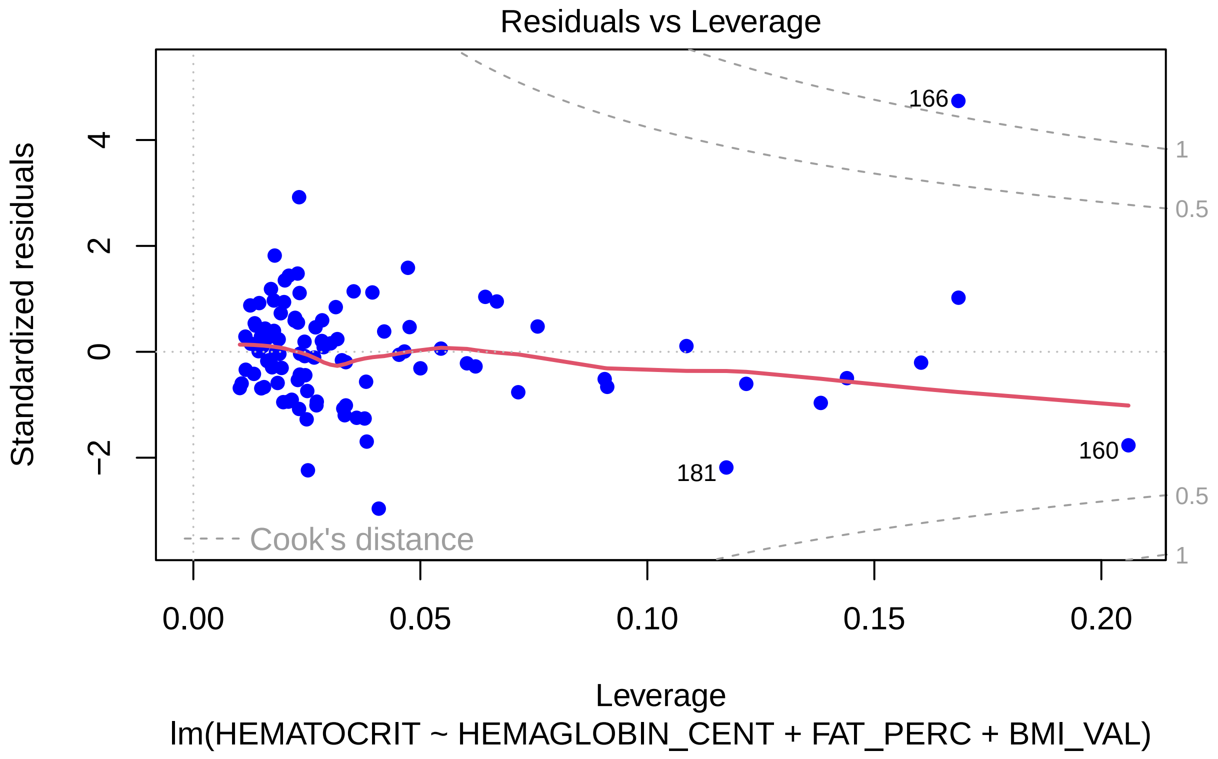
```
plot(mod1, pch=16, col="blue", lty=1, lwd=2, which=2)
```



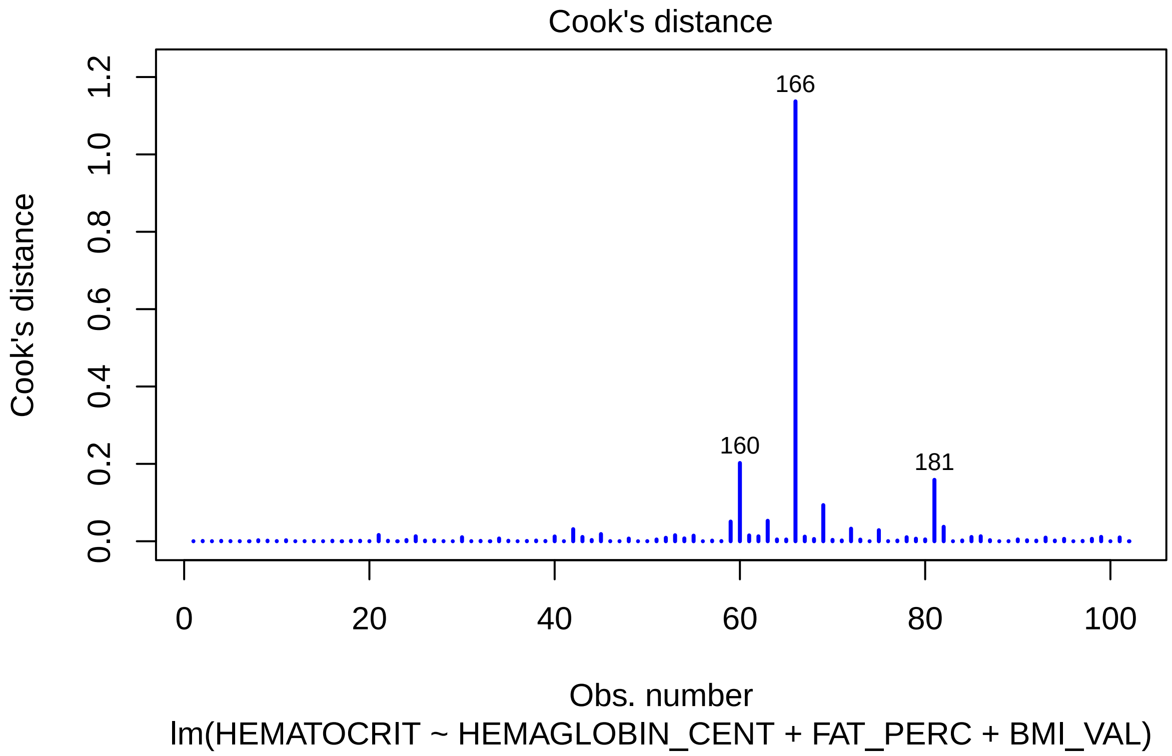
```
plot(mod1, pch=16, col="blue", lty=1, lwd=2, which=3)
```



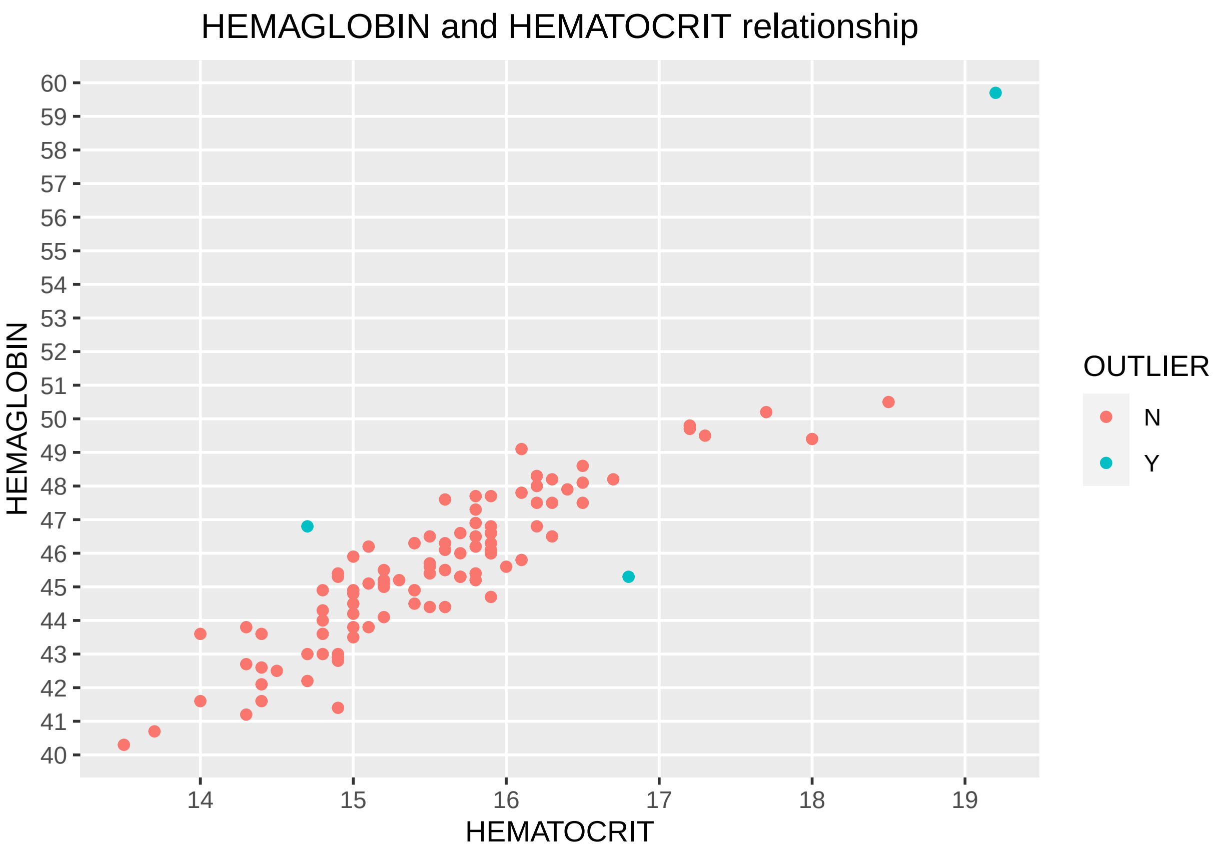
```
plot(mod1, pch=16, col="blue", lty=1, lwd=2, which=5)
```



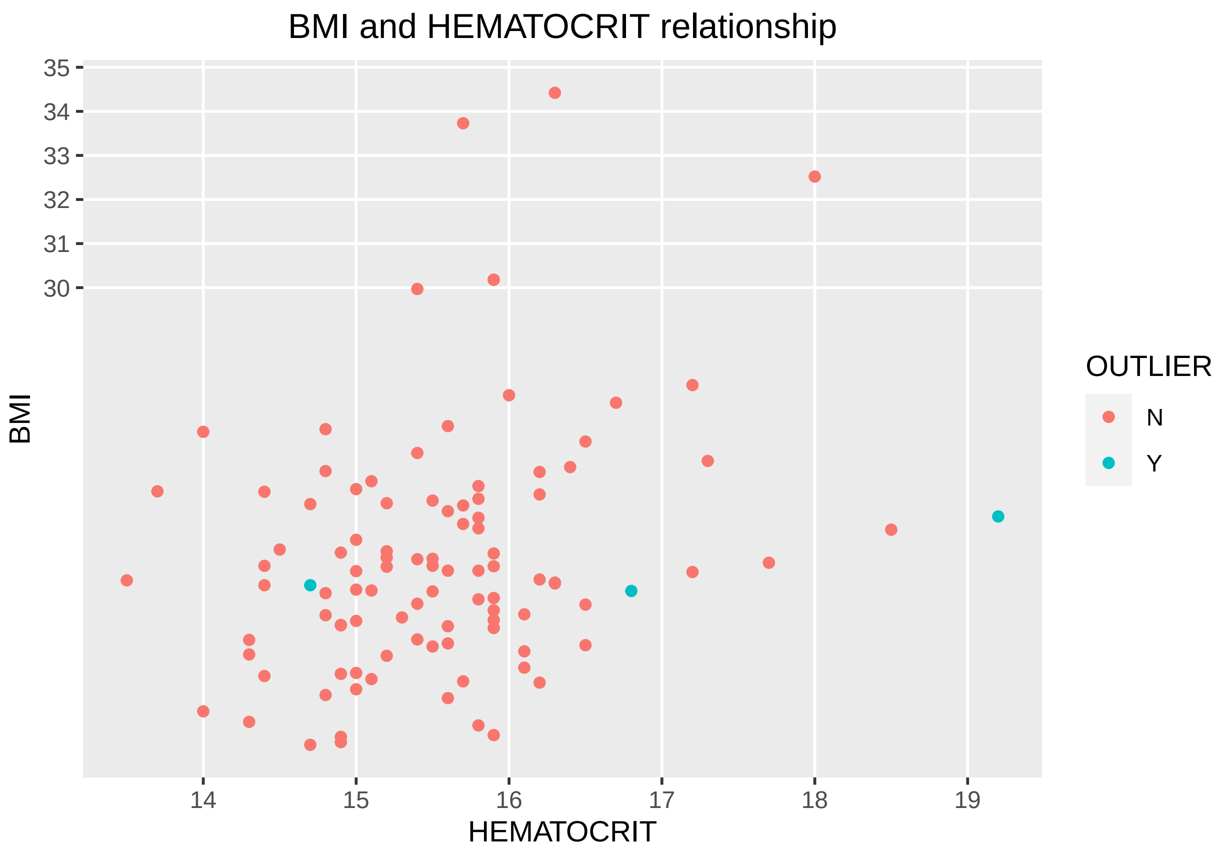
```
plot(mod1, pch=16, col="blue", lty=1, lwd=2, which=4)
```



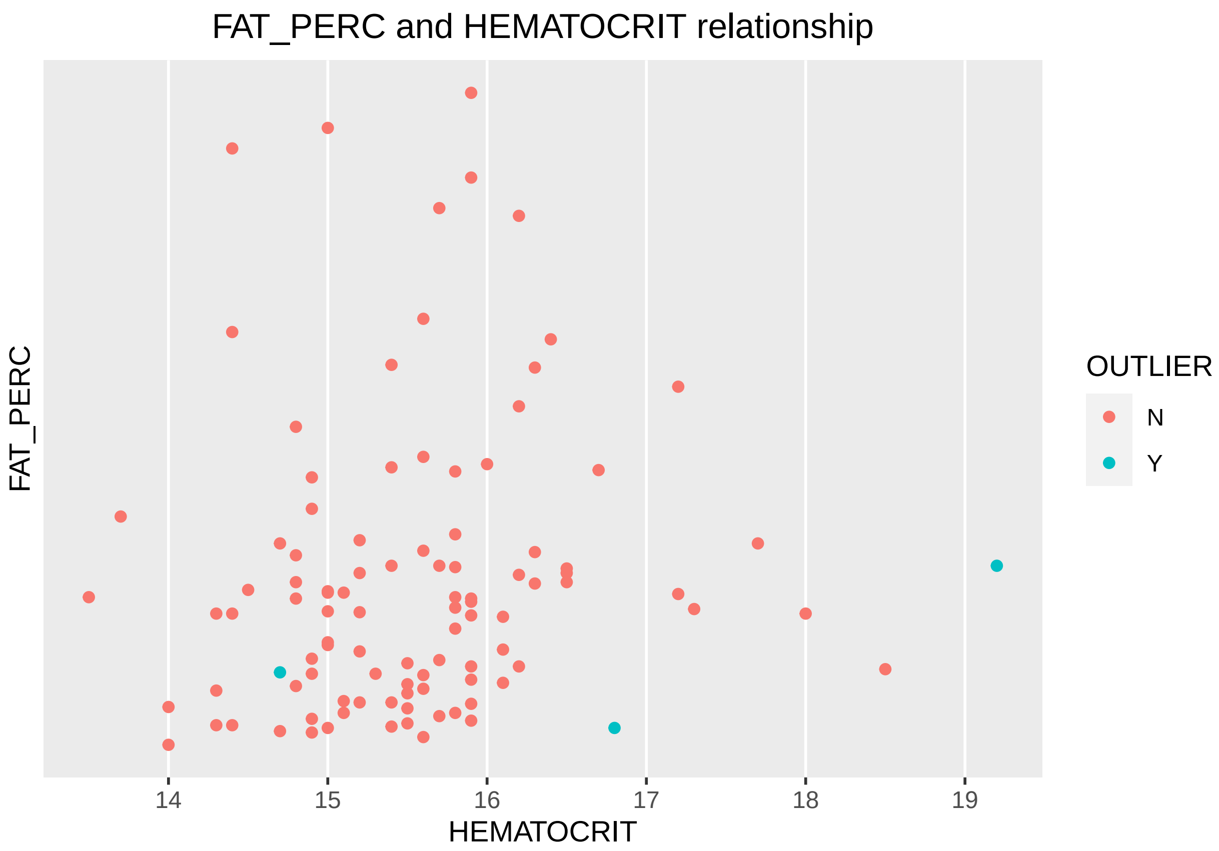
```
newdata1 <- setNames(cbind(rownames(newdata), newdata, row.names = NULL),
                     c("OBS", "HEMAGLOBIN", "HEMATOCRIT", "BMI", "FAT_PERC"))
newdata1$OUTLIER = ifelse(newdata1$OBS %in% c(159,166,169),"Y","N") # create condition Yes/No if outlie

qplot(HEMATOCRIT, HEMAGLOBIN, data = newdata1, colour = OUTLIER,
main = "HEMAGLOBIN and HEMATOCRIT relationship") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



```
qplot(HEMATOCRIT, BMI, data = newdata1, colour = OUTLIER,
main = "BMI and HEMATOCRIT relationship") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



```
qplot(HEMATOCRIT, FAT_PERC, data = newdata1, colour = OUTLIER,
main = "FAT_PERC and HEMATOCRIT relationship") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(breaks = c(30:65), minor_breaks = NULL) +
  scale_x_continuous(breaks = c(10:25), minor_breaks = NULL)
```



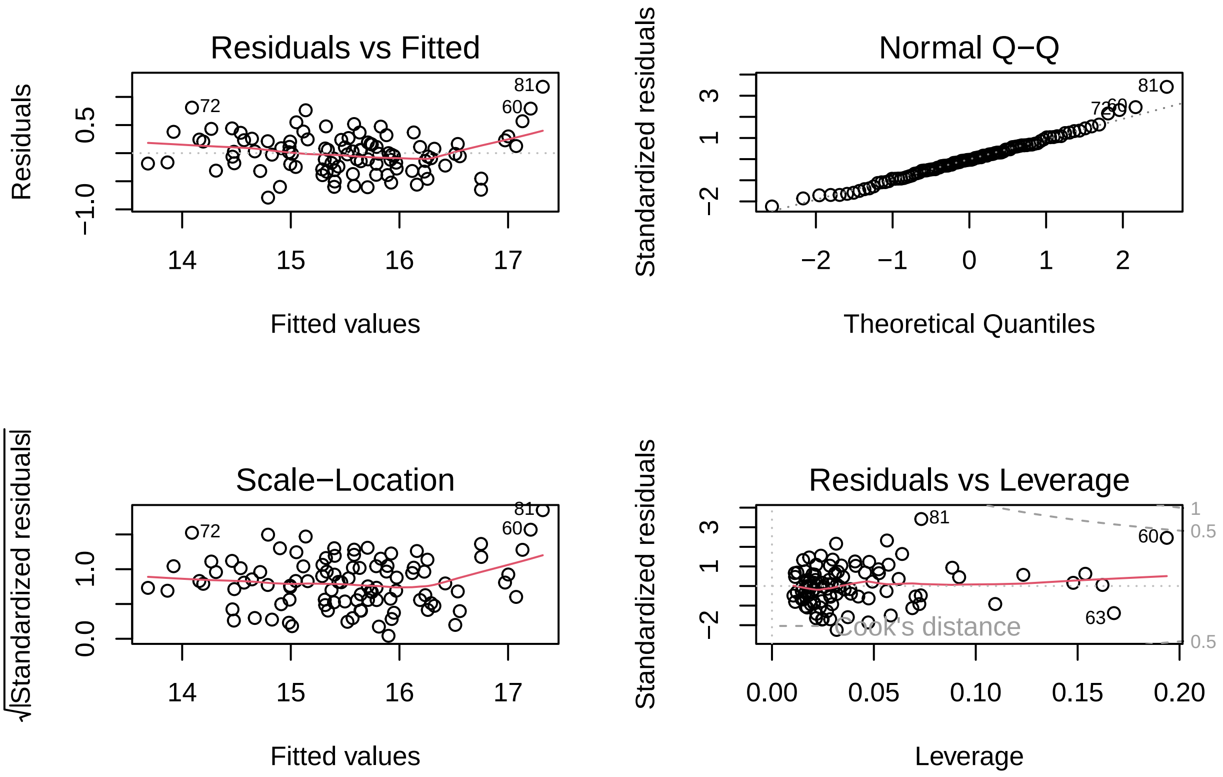
```
newdata2 <- subset(newdata1, OBS != 159 & OBS != 166 & OBS != 169,
                   select=c(HEMAGLOBIN, HEMATOCRIT, BMI, FAT_PERC))
HEMAGLOBIN_CENT = scale(newdata2$HEMAGLOBIN, center=TRUE, scale=FALSE) # center the variable
FAT_CENT = scale(newdata2$FAT_PERC, center=TRUE, scale=FALSE)
BMI_VAL = scale(newdata2$BMI, center=TRUE, scale=FALSE)
```

```
mod2 = lm(HEMATOCRIT ~ HEMAGLOBIN_CENT+BMI_VAL+FAT_CENT, data = newdata2)
summary(mod2)
```

```
##
## Call:
## lm(formula = HEMATOCRIT ~ HEMAGLOBIN_CENT + BMI_VAL + FAT_CENT,
##     data = newdata2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7909 -0.2300 -0.0116  0.2202  1.1808
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     15.51212    0.03606 430.127   <2e-16 ***
## HEMAGLOBIN_CENT  0.34905    0.01727  20.215   <2e-16 ***
## BMI_VAL          0.03701    0.01707   2.168   0.0327 *
## FAT_CENT        -0.01968    0.01452  -1.355   0.1786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3588 on 95 degrees of freedom
## Multiple R-squared:  0.8311, Adjusted R-squared:  0.8257
## F-statistic: 155.8 on 3 and 95 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2)) # display a unique layout for all graphs
plot(mod2)
```



```
AIC(mod1)
```

```
## [1] 324.9052
```

```
AIC(mod2)
```

```
## [1] 83.93676
```

```
BIC(mod1)
```

```
## [1] 338.0301
```

```
BIC(mod2)
```

```
## [1] 96.91236
```

```
set.seed(123)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(newdata2), 0.7*nrow(newdata2))  #  training and testing: 70/30 split
trainingData <- newdata2[trainingRowIndex, ]  # training data
testData  <- newdata2[-trainingRowIndex, ]   # test data
```

```
modTrain <- lm(HEMATOCRIT ~ HEMAGLOBIN+BMI+FAT_PERC, data=trainingData)  # build the model
predict <- predict(modTrain, testData)  # predicted values
```

```
summary(modTrain)
```

```
##
## Call:
## lm(formula = HEMATOCRIT ~ HEMAGLOBIN + BMI + FAT_PERC, data = trainingData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7892 -0.2394 -0.0052  0.2120  1.1949
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.95692    0.97380  -0.983    0.329
## HEMAGLOBIN   0.34466    0.02058  16.750   <2e-16 ***
## BMI          0.03922    0.02468   1.589    0.117
## FAT_PERC    -0.01435    0.01874  -0.766    0.447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3691 on 65 degrees of freedom
## Multiple R-squared:  0.8251, Adjusted R-squared:  0.817
## F-statistic: 102.2 on 3 and 65 DF,  p-value: < 2.2e-16
```

```
act_pred <- data.frame(cbind(actuals=testData$HEMATOCRIT, predicteds=predict)) # actuals_predicteds
cor(act_pred) # correlation_accuracy
```

```
##              actuals predicteds
## actuals    1.0000000  0.9199378
## predicteds 0.9199378  1.0000000
```

```
head(act_pred, n=10)
```

```
##    actuals predicteds
## 1     15.9   15.93254
## 2     15.2   15.44810
## 3     15.9   15.94459
## 10    15.4   15.37968
## 11    16.1   16.23528
## 19    15.4   15.17829
## 20    16.2   16.27280
## 24    15.5   15.89244
## 28    15.6   15.53141
## 35    13.7   13.91348
```

```
min_max <- mean(apply(act_pred, 1, min) / apply(act_pred, 1, max))
print(min_max) # show the result
```

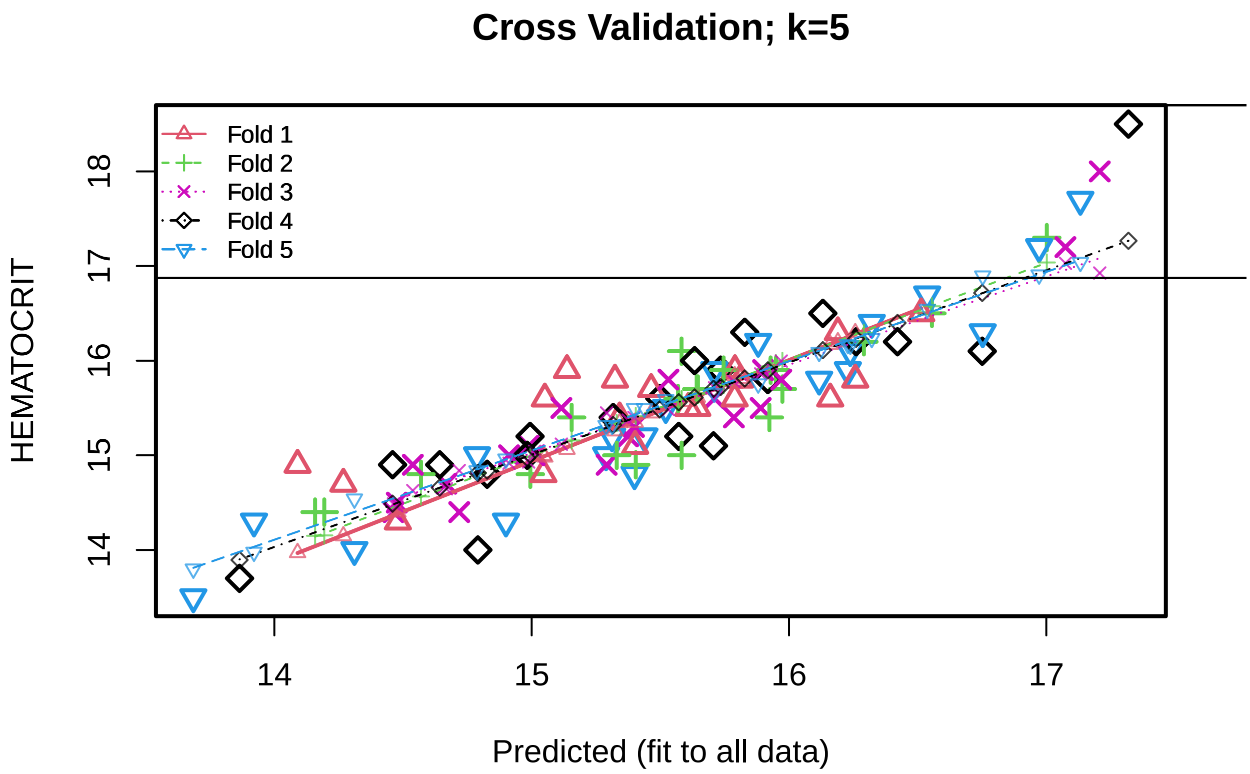
```
## [1] 0.982513
```

```
mape <- mean(abs((act_pred$predicteds - act_pred$actuals))/act_pred$actuals)
print(mape) # show the result
```

```
## [1] 0.01774838
```

```
kfold <- CVlm(data = newdata2, form.lm = formula(HEMATOCRIT ~ HEMAGLOBIN+BMI+FAT_PERC), m=5,
                   dots = FALSE, seed=123, legend.pos="topleft",
                   main="Cross Validation; k=5",
                   plotit=TRUE, printit=FALSE)
```

```
## Warning in CVlm(data = newdata2, form.lm = formula(HEMATOCRIT ~ HEMAGLOBIN + :
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```



```
attr(kfold, 'ms')
```

```
## [1] 0.1469749
```