Assignment 2 - Question 3

Toni Ciobanu (20910287)

2024-02-17

QUESTION 3: Robust Regression with Mind Monitor Data

(a) [3 points] Construct a scatter plot of `blinks_minute` versus `jaws_minute`, and add the least squares regression line to this plot. Note that you may use the `lm` function to determine the equation of this line. Be sure to add a title and informative axis labels.

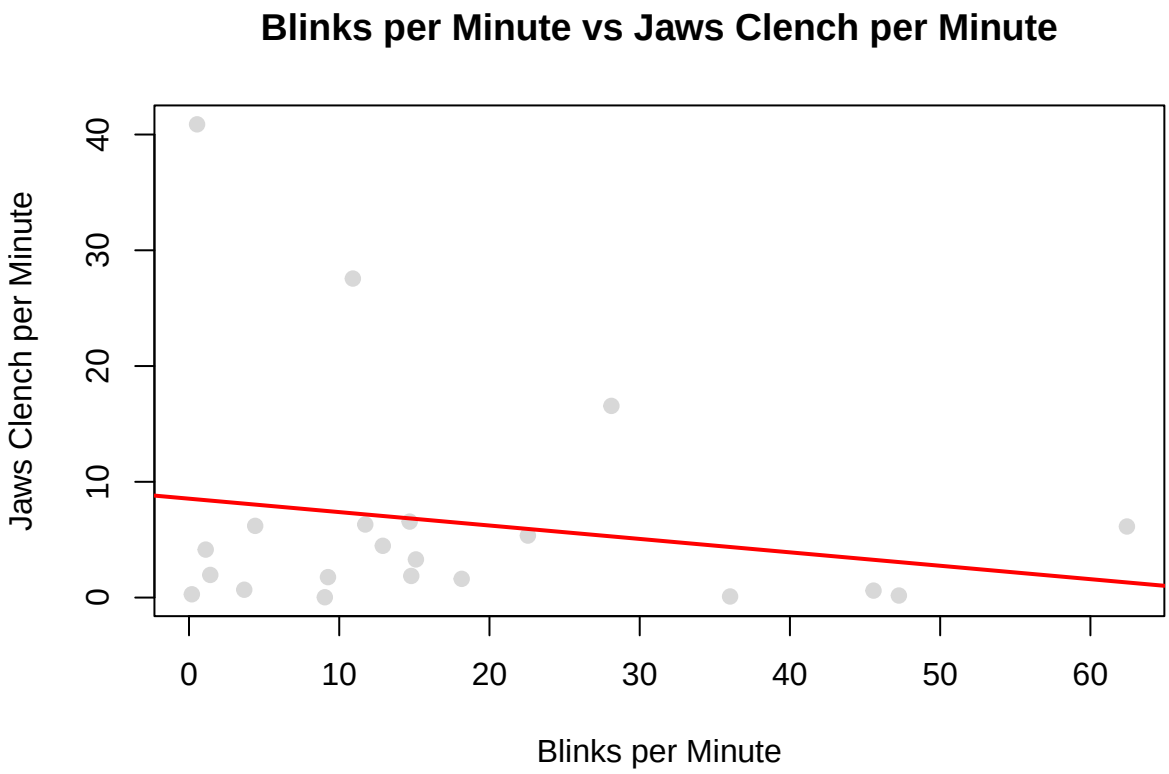
SOLUTION:

```
# load files
eeg_summary <- read.csv("eeg_summary.csv")

# Plot the points and create graph
plot(eeg_summary$blinks_minute, eeg_summary$jaws_minute, pch = 19, cex = 1,
     col = adjustcolor("grey", 0.6),
     xlab = "Blinks per Minute",
     ylab = "Jaws Clench per Minute",
     main = "Blinks per Minute vs Jaws Clench per Minute")

# Fit the linear model
fit = lm(jaws_minute ~ blinks_minute, data = eeg_summary)

# Add line to model
abline(fit, col = "red", lwd = 2)
```



(b) [4 points] For each unit in the population, calculate the influence that it has on the fitted regression line from part (a), using the following definition of influence:

$$\Delta(\boldsymbol{\theta}, u) = ||\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{[-u]}||_2$$

where $\widehat{\boldsymbol{\theta}} = \left(\widehat{\alpha}, \widehat{\beta}\right)^T$ are the regression coefficients estimated from all of the data, $\widehat{\boldsymbol{\theta}}_{[-u]} = \left(\widehat{\alpha}_{[-u]}, \widehat{\beta}_{[-u]}\right)^T$ are the regression coefficients estimated from all of the data excluding unit u, and $||\cdot||_2$ is the Euclidean norm. Construct a scatter plot of all influence values and determine on which `session_num` the two most influential recordings occurred.

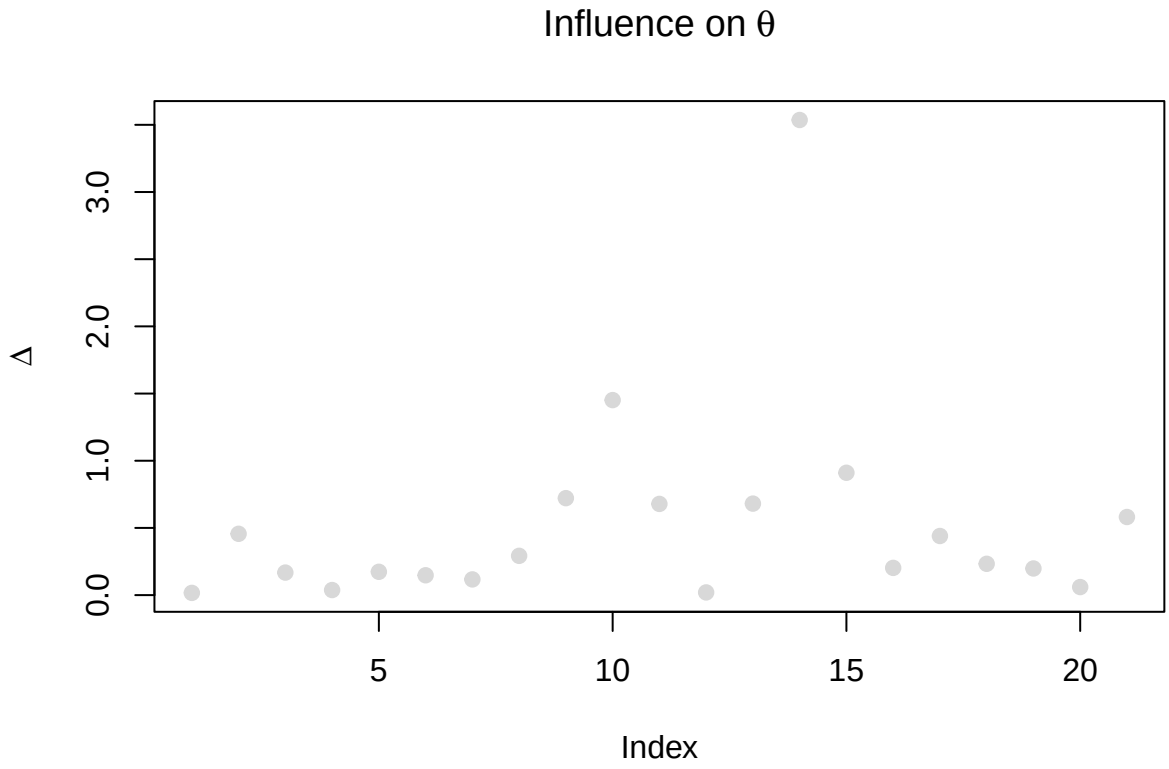
SOLUTION:

```
N <- nrow(eeg_summary)
delta <- matrix(0, nrow = N, ncol = 2)

for (i in 1:N) {
    fit.no.i <- lm(jaws_minute ~ blinks_minute, data = eeg_summary[-i, ])
    delta[i, ] <- abs(fit$coef - fit.no.i$coef)
}

# Calculate the influence values
delta2 <- apply(delta, MARGIN = 1, FUN = function(z) sqrt(sum(z^2)))
```

```
plot(delta2, main = bquote("Influence on" ~ theta), ylab = bquote(Delta),
     pch = 19, col = adjustcolor("grey", 0.6))
```



The two units with the largest influence are: 10, and 14.

```
eeg_summary[delta2 > 1, ]
```

```
##    sessionnum activity session_time mean_alpha mean_beta  mean_gamma mean_delta
## 10         10   active         3207   1.008187 0.5972533  0.02937049  0.5920039
## 14         14  resting         3568   2.277636 1.2297560 -0.11367039  1.0205041
##    mean_theta var_alpha  var_beta var_gamma var_delta var_theta blinks_minute
## 10  0.3828858 0.2719549 0.1398195 0.1518134 0.9615571 0.4005318    10.9073901
## 14  1.3549595 0.6377508 0.3062240 0.2378915 0.6521849 0.4363715     0.5381166
##    jaws_minute mean_pos_xy   mad_accel rmse_gyro
## 10    27.55847   0.1385098 0.019326195  7.448870
## 14    40.88004   0.9220250 0.008687843  5.931053
```

(c) [4 points] Re-construct the scatter plot from part (a). Remove the two most influential observations from the population and calculate the least squares regression line using the observations that remain (you may use `lm` again). Add this line to the plot in a colour different from what you used on the first line. How would your conclusions about the relationship between `blinks_minute` and `jaws_minute` differ when using the line created in part (a) vs. using the line created in part (c)?

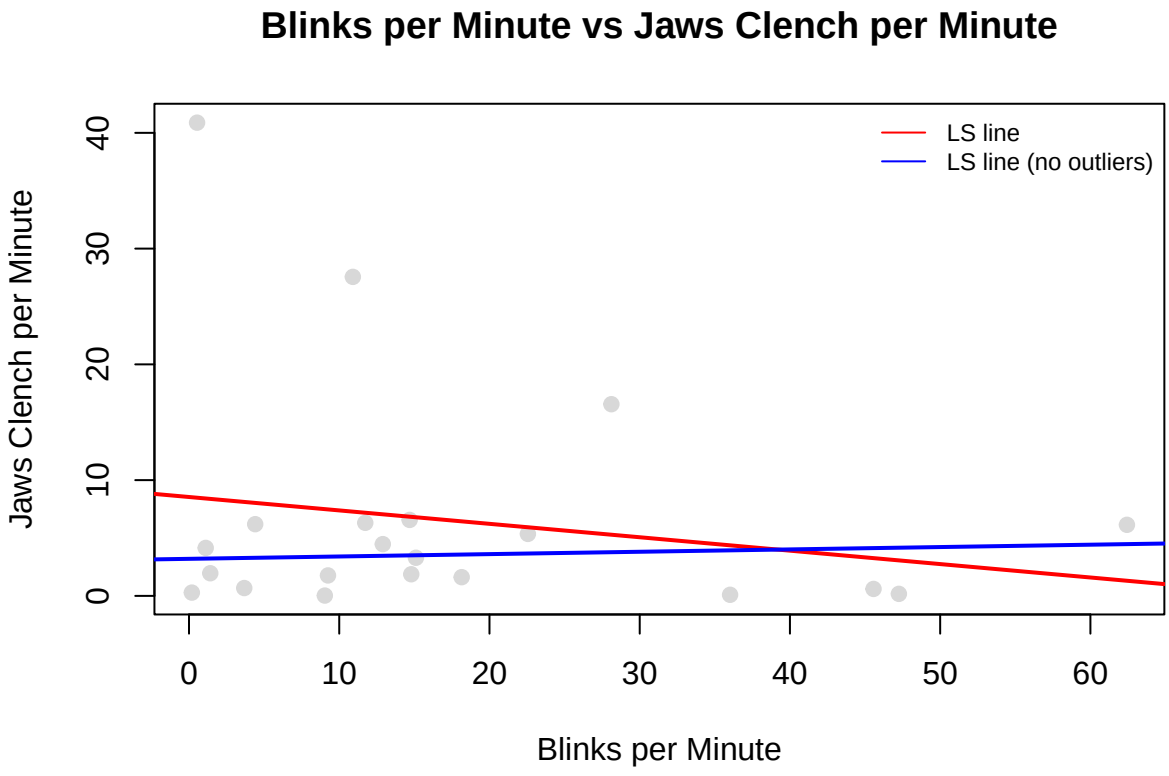
SOLUTION:

```r
# Copy plot from part a
plot(eeg_summary$blinks_minute, eeg_summary$jaws_minute, pch = 19, cex = 1,
     col = adjustcolor("grey", 0.6),
     xlab = "Blinks per Minute",
     ylab = "Jaws Clench per Minute",
     main = "Blinks per Minute vs Jaws Clench per Minute")
fit <- lm(jaws_minute ~ blinks_minute, data = eeg_summary)
abline(fit, col = "red", lwd = 2)

# take out session_num 10 and 14 from data
eeg_summary2 <- eeg_summary[-c(10, 14), ]

# Calculate the new regression line and add it to the plot
abline(lm(jaws_minute ~ blinks_minute, data = eeg_summary2),
       col = "blue", lwd = 2)

# Make legend
legend("topright", legend = c("LS line", "LS line (no outliers)"), col = c("red",
    "blue"), cex = 0.75, bty = "n", lty = 1)
```



The data from the lines in the graph above indicate the presence of a much stronger outline in the data set. Since the 2 points that were removed had a strong effect on the slope of the regression line, we can say that the 2 most influential points may have skewed the relationship between the blinks per minute over the jaws

clench per minute. Removing those points made the regression line more robust.

(d) [12 points] Rather than removing highly influential observations prior to analysis, we could instead mitigate their influence by performing a robust linear regression.

i. [2 points] By taking appropriate derivatives, determine the 2×1 gradient vector $\boldsymbol{g} = \nabla\rho(\boldsymbol{\theta}; \mathcal{P})$. Show your work.

SOLUTION:

Given:

$$\rho(\boldsymbol{\theta}; \mathcal{P}) = \sum_{u \in \mathcal{P}} \rho_{a,b,c}(r_u)$$

where $\boldsymbol{\theta} = (\alpha, \beta)^T$, $r_u = y_u - \alpha - \beta x_u$ is the Huber function.

The corresponding gradient is

$$\boldsymbol{g} = \nabla\rho(\boldsymbol{\theta})$$

which can be decomposed using the Chain Rule as follows:

$$\nabla\rho(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{u \in \mathcal{P}} \rho_{a,b,c}(r_u) = \sum_{u \in \mathcal{P}} \frac{\partial \rho_{a,b,c}(r_u)}{\partial \boldsymbol{\theta}} = \sum_{u \in \mathcal{P}} \frac{\partial \rho_{a,b,c}(r_u)}{\partial r_u} \times \frac{\partial r_u}{\partial \boldsymbol{\theta}}$$

where

$$\frac{\partial r_u}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial r_u}{\partial \alpha} \\ \frac{\partial r_u}{\partial \beta} \end{bmatrix} = - \begin{bmatrix} 1 \\ x_u \end{bmatrix}$$

Thus the gradient can be written as

$$\boldsymbol{g} = - \begin{bmatrix} \sum_{u \in \mathcal{P}} \frac{\partial \rho_{a,b,c}(r_u)}{\partial r_u} \\ \sum_{u \in \mathcal{P}} \frac{\partial \rho_{a,b,c}(r_u)}{\partial r_u}(x_u) \end{bmatrix}$$

ii. [4 points] Write *factory functions* `createRobustHampelRho(x, y, aval, bval, cval)` and `createRobustHampelGradient(x, y, aval, bval, cval)` which take in as inputs the data and values for the constants a, b, c, and which respectively return as output the Hampel objective function and the corresponding gradient function. **Hint:** Use the `createRobustHuberRho` and `createRobustHuberGradient` functions from the lecture notes as a guide. You may also use the following two functions as necessary.

```
hampel.fn <- function(r, a, b, c){
  val <- rep(a*(b-a+c)/2, length(r))
  val[r > -c & r<= -b] <- -a * (c*r[r > -c & r<= -b] + (0.5*r^2)[r > -c & r<= -b])  / (c-b) - 0.5*(a*b^
  val[r > -b & r<= -a] <- -a * r[r > -b & r<= -a] - 0.5*a^2
  val[r > -a & r<= a] <- (0.5*r^2)[r > -a & r<= a]
  val[r > a & r<= b] <- a * r[r > a & r<= b] - 0.5*a^2
  val[r > b & r<= c] <- a * (c*r[r > b & r<= c] - (0.5*r^2)[r > b & r<= c]) / (c-b) - 0.5*(a*b^2)/(c-b)
  return(val)
}
hampel.fn.prime <- function(r, a, b, c){
  val <- rep(0, length(r))
  val[r > -c & r<= -b] <- -a * (c+r[r > -c & r<= -b]) / (c-b)
  val[r > -b & r<= -a] <- -a
  val[r > -a & r<= a] <- r[r > -a & r<= a]
```

```
  val[r > a & r<= b] <- a
  val[r > b & r<= c] <- a * (c-r[r > b & r<= c]) / (c-b)
  return(val)
}
```

SOLUTION:

```
createRobustHampelRho <- function(x, y, aval, bval, cval) {
    ## Return this function
    function(theta) {
        alpha <- theta[1]
        beta <- theta[2]
        sum(hampel.fn(y - alpha - beta * x, a = aval, b = bval, c = cval))
    }
}

createRobustHampelGradient <- function(x, y, aval, bval, cval) {
    function(theta) {
        alpha <- theta[1]
        beta <- theta[2]
        ru = y - alpha - beta * x
        rhoabc = hampel.fn.prime(ru, a = aval, b = bval, c = cval)
        -1 * c(sum(rhoabc * 1), sum(rhoabc * x))
    }
}
```

iii. [2 point] Using the `nlminb` function with the `rho` and `gradient` functions created by your factory functions from part ii., find $\widehat{\boldsymbol{\theta}} = \left(\widehat{\alpha}, \widehat{\beta}\right)^T$, the solution to

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^2}{\operatorname{argmin}} \, \rho(\boldsymbol{\theta}; \mathcal{P}).$$

Start the optimization at the least squares estimates of α and β determined in part (a) and use $a = 2, b = 4, c = 8$. For full points be sure to include the output from the `nlminb` function. **Hint:** Your robust regression estimate for β should take a value between the two least squares estimates for β found in parts (a) and (c).

SOLUTION:

```
## Define Alpha, Beta, x and y.
alpha <- fit$coef[1]
beta <- fit$coef[2]
x <- eeg_summary$blinks_minute
y <- eeg_summary$jaws_minute

# Calculate optimization at the least squares estimate using variables given
hampel <- nlminb(start = c(alpha, beta),
       objective = createRobustHampelRho(x, y, 2, 4, 8),
       gradient = createRobustHampelGradient(x, y, 2, 4, 8))

hampel
```

```
## $par
##   (Intercept) blinks_minute
##     2.8978348    -0.0272739
##
## $objective
## [1] 74.48011
##
## $convergence
## [1] 0
##
## $iterations
## [1] 12
##
## $evaluations
## function gradient
##       21       13
##
## $message
## [1] "relative convergence (4)"
```

iv. [4 points] Re-construct the scatter plot from part (c). In a third colour, add the Hampel regression line that you calculated in part iii. Be sure to include a legend that distinguishes among the three lines. Does the robust regression line convey similar findings about the relationship between blinks_minute and jaws_minute as the other two? With that in mind, what does this suggest about how influential units impact the reliability of our conclusions?

SOLUTION:

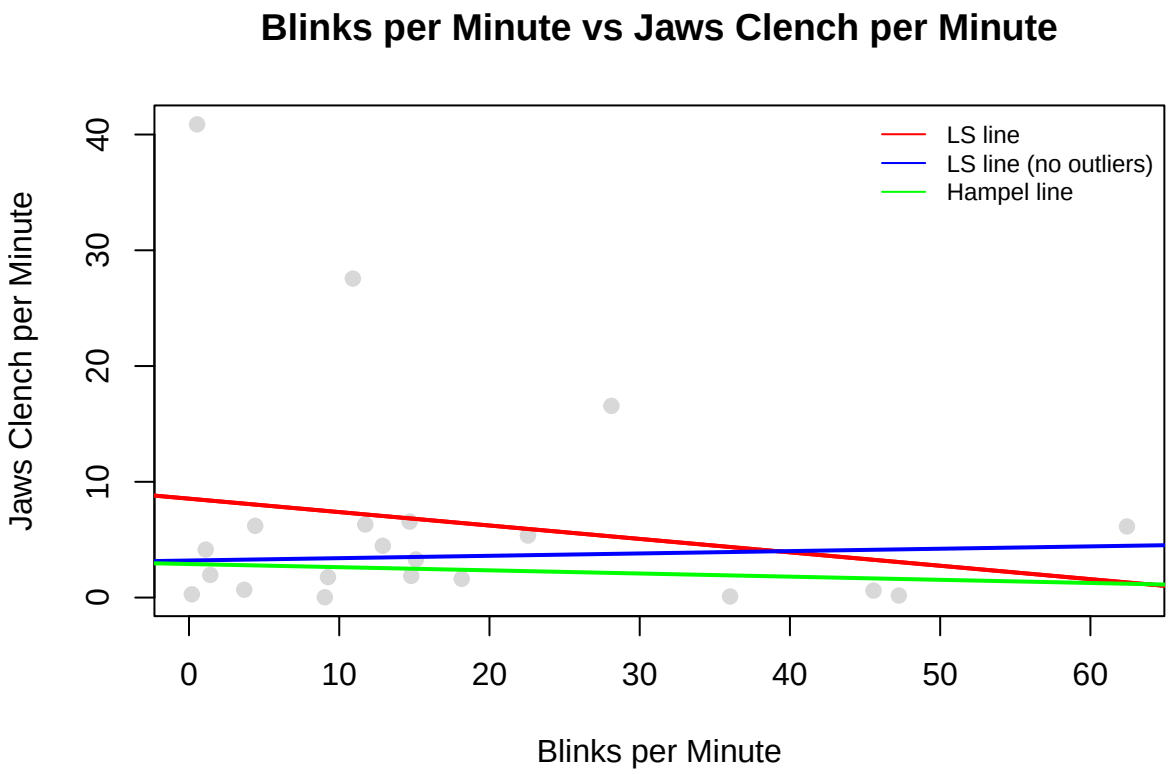
```
# Copy plot from part a
plot(eeg_summary$blinks_minute, eeg_summary$jaws_minute, pch = 19, cex = 1,
     col = adjustcolor("grey", 0.6),
     xlab = "Blinks per Minute",
     ylab = "Jaws Clench per Minute",
     main = "Blinks per Minute vs Jaws Clench per Minute")
abline(fit, col = "red", lwd = 2)

# Add the original least squares regression line
abline(fit, col = "red", lwd = 2)

# Add the least squares line without the two most influential points
abline(lm(jaws_minute ~ blinks_minute, data = eeg_summary2), col = "blue", lwd = 2)

# Add Hampel line parameters found in part iii
abline(a = hampel$par[1], b = hampel$par[2], col = "green", lwd = 2)

# Add legend
legend("topright", legend = c("LS line", "LS line (no outliers)", "Hampel line"),
       col = c("red", "blue", "green"), cex = 0.75, bty = "n", lty = 1)
```



Does the robust regression line convey similar findings about the relationship between blinks_minute and jaws_minute as the other two?

Yes, The least squares line without influential points start at approximately the same y intercept as the Hampel line, and the original least squares line has a negative slope like this one. I observed that the Hampel line is about equivalent to an average of the slope between the 2 other lines, as the original LS line shows a slightly larger negative slope than the LS line without influential points' positive slope. This is reflected in the Hampel line as the line appears almost at a 0 slope, but slightly negative.

With that in mind, what does this suggest about how influential units impact the reliability of our conclusions?

Since the Hampel line is closer to the LS line without the top 2 most influential points than the original LS line, we can say that the original LS estimate was heavily influenced by the 2 outliers. Influential units can drastically impact the data analysis of a data set. We originally believed there to be a large negative correlation between the Jaws Clench per minute and the blinks per minute, however further analysis showed us that there is less of a correlation than we believed. The data was being skewed by the 2 influential units that contradicted the correlation. This suggests that the reliability of our conclusions are dependent on the robustness of the statistical estimation method used.