Assignment 4

Ilaria Crippa

2024-02-09

Contents

1 Multiple linear regression and diagnostics

1.1 Joint null hypothesis to test whether the effect of a group of variables is the same and the remaining one has a negligible effect on the response.

Our objective is to formulate a joint null hypothesis to test whether the effect of the latent heat flux (LH) has a negligible effect, jointly with the null hypothesis that relative minimum humidity (RHmin) and precipitation (PPT) have the same effect on the response variable: H_0: $_6 = 0$ and H_0: $_2 = _4$.

```
model = lm(Next_Tmax ~ Tmax_lapse + RHmin + CC + PPT + WS + LH + grouped_station, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ Tmax_lapse + RHmin + CC + PPT + WS +
##     LH + grouped_station, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9240 -0.9047  0.0331  0.9607  6.4807
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.0696948  0.3036707  16.695  < 2e-16 ***
## Tmax_lapse                      0.8335212  0.0079315 105.090  < 2e-16 ***
## RHmin                           0.0450551  0.0019803  22.752  < 2e-16 ***
## CC                             -3.6091610  0.1513448 -23.847  < 2e-16 ***
## PPT                             0.0687725  0.0229643   2.995  0.00276 **
```

```
## WS                                -0.1268319  0.0090427 -14.026  < 2e-16 ***
## LH                                 0.0059488  0.0005777  10.298  < 2e-16 ***
## grouped_stationLow DEM station    -0.5897407  0.0520836 -11.323  < 2e-16 ***
## grouped_stationMedium DEM station -0.3004025  0.0510428  -5.885 4.14e-09 ***
## grouped_stationHigh DEM station   -0.4934508  0.0621308  -7.942 2.27e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.559 on 7578 degrees of freedom
## Multiple R-squared:  0.7492, Adjusted R-squared:  0.7489
## F-statistic:  2515 on 9 and 7578 DF,  p-value: < 2.2e-16
```

In order to build this joint hypothesis test, we build a matrix A with two rows, corresponding to the number of hypotheses, and 10 columns, corresponding to the number of coefficients. We then create a 0 vector, whose entries correspond to the values assigned to the conditions of the null hypotheses.

```
A = matrix(0, nrow = 2, ncol = 10)
b = c(0, 0)
A[1, 3] = 1
A[1, 5] = -1
A[2, 7] = 1
print(A)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    0    0    1    0   -1    0    0    0    0     0
## [2,]    0    0    0    0    0    0    1    0    0     0
```

```
linearHypothesis(model, A, b)
```

```
## Linear hypothesis test
##
## Hypothesis:
## RHmin - PPT = 0
## LH = 0
##
## Model 1: restricted model
## Model 2: Next_Tmax ~ Tmax_lapse + RHmin + CC + PPT + WS + LH + grouped_station
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   7580 18696
## 2   7578 18428  2    267.69 55.038 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the test is lower than our confidence level of 0.05; therefore, we can reject our null hypotheses. However, we can note that there is not enough evidence to reject the null hypothesis H_0: $_2 = _4$ (corresponding to the coefficients for minimum relative humidity and precipitation), when tested on its own.

1.2 Diagnostics

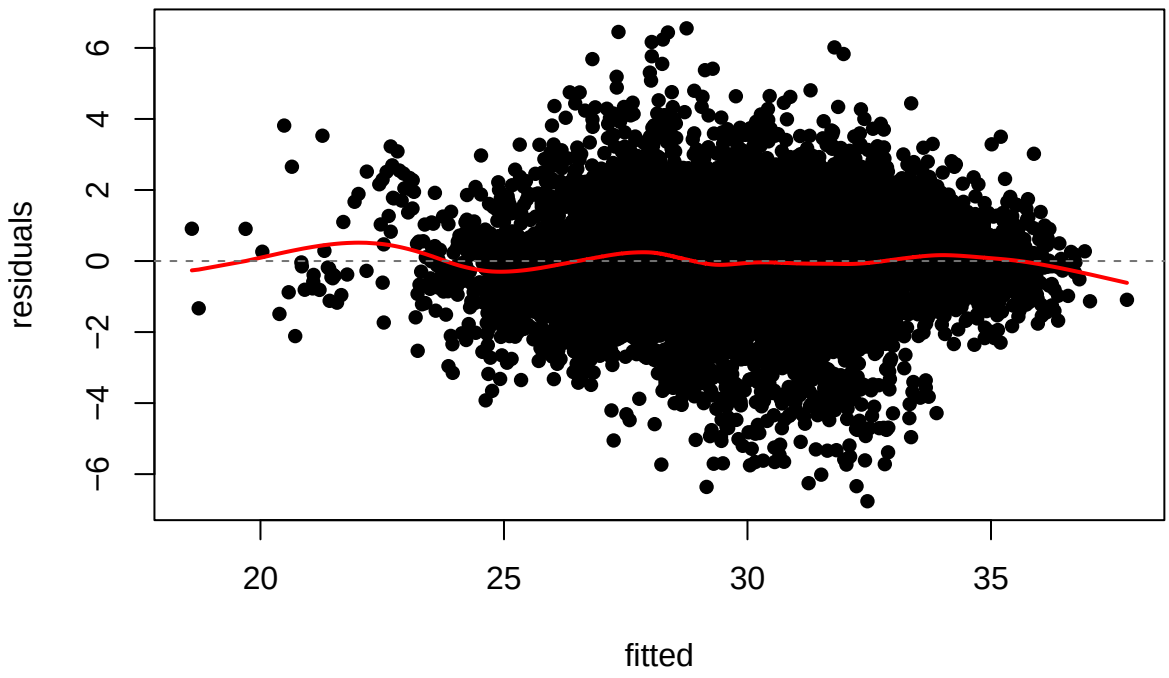
1.2.1 Constant variance assumptions for the errors

Given the multiple linear model fitted in the previous assignment, we perform regression diagnostics, starting with the constant variance of the residuals assumption. To do so, we check the plot of the residuals against the fitted values; the red line helps in the visualization of the mean function and to assess the linearity of the residuals.

```r
model2 = lm(Next_Tmax ~ Tmax_lapse + RHmin + CC + LH + WS * grouped_station, data = data)
summary(model2)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ Tmax_lapse + RHmin + CC + LH + WS *
##     grouped_station, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7630 -0.9070  0.0356  0.9629  6.5507
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           5.2543339  0.3249995  16.167  < 2e-16 ***
## Tmax_lapse                            0.8355530  0.0079162 105.550  < 2e-16 ***
## RHmin                                 0.0451864  0.0019813  22.806  < 2e-16 ***
## CC                                   -3.4360532  0.1418890 -24.216  < 2e-16 ***
## LH                                    0.0061237  0.0005745  10.659  < 2e-16 ***
## WS                                   -0.1700371  0.0203623  -8.351  < 2e-16 ***
## grouped_stationLow DEM station       -1.0836094  0.1847759  -5.864 4.70e-09 ***
## grouped_stationMedium DEM station    -0.5326490  0.1829706  -2.911  0.00361 **
## grouped_stationHigh DEM station      -0.9307140  0.2028191  -4.589 4.53e-06 ***
## WS:grouped_stationLow DEM station     0.0721424  0.0255765   2.821  0.00481 **
## WS:grouped_stationMedium DEM station  0.0339584  0.0256905   1.322  0.18627
## WS:grouped_stationHigh DEM station    0.0611552  0.0260670   2.346  0.01900 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.56 on 7576 degrees of freedom
## Multiple R-squared:  0.7492, Adjusted R-squared:  0.7488
## F-statistic:  2057 on 11 and 7576 DF,  p-value: < 2.2e-16
```

```r
residuals = residuals(model2)
fitted = fitted.values(model2)
res_vs_fitted = smooth.spline(x = fitted, y = residuals)
plot(fitted, residuals, pch = 16)
lines(res_vs_fitted, col = "red", lwd = 2)
abline(h = 0,
       col = "grey45",
       lty = 2)
```

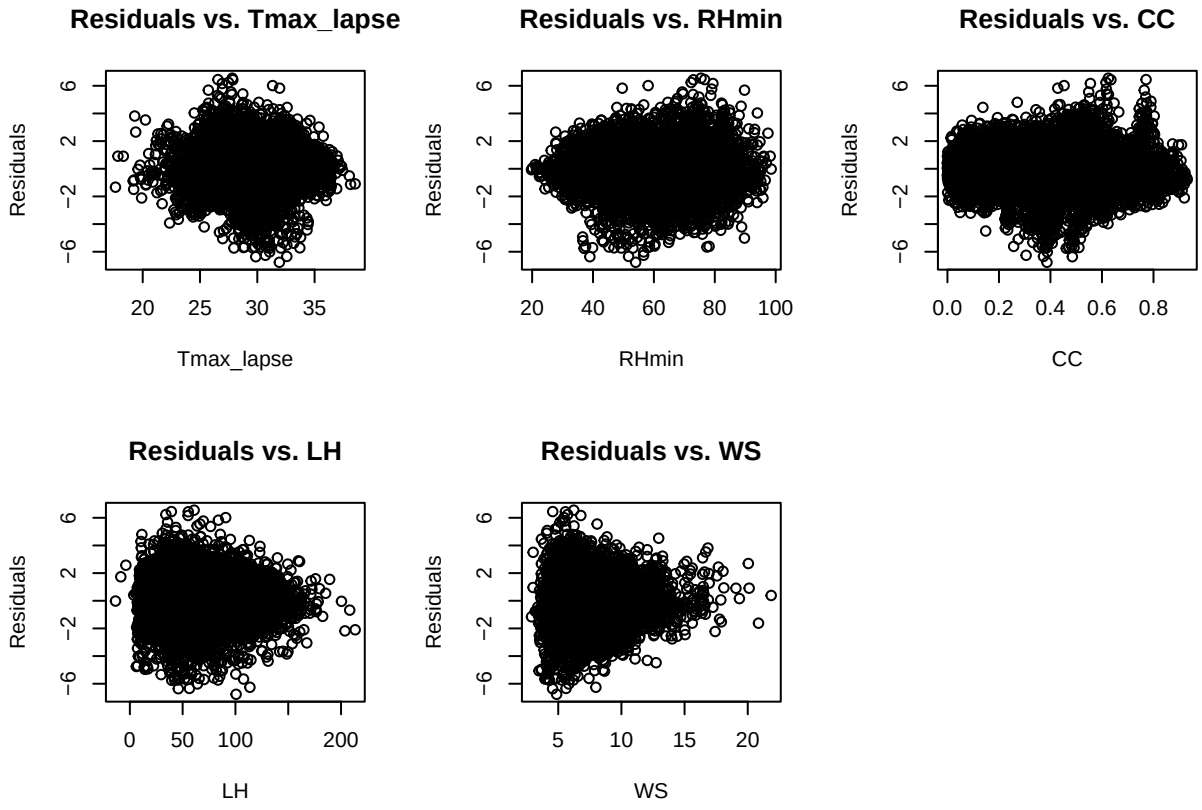


We can note that the residuals have a tendency to be homoscedastic.

It is also helpful to check the residuals plotted against each predictor variable to detect anomalies in the variance:

```
par(mfrow = c(2, 3))
plot(
  residuals(model2) ~ data$Tmax_lapse,
  main = "Residuals vs. Tmax_lapse",
  xlab = "Tmax_lapse",
  ylab = "Residuals"
)
plot(
  residuals(model2) ~ data$RHmin,
  main = "Residuals vs. RHmin",
  xlab = "RHmin",
  ylab = "Residuals"
)
plot(
  residuals(model2) ~ data$CC,
  main = "Residuals vs. CC",
  xlab = "CC",
  ylab = "Residuals"
)
plot(
  residuals(model2) ~ data$LH,
  main = "Residuals vs. LH",
```

```
  xlab = "LH",
  ylab = "Residuals"
)
plot(
  residuals(model2) ~ data$WS,
  main = "Residuals vs. WS",
  xlab = "WS",
  ylab = "Residuals"
)
```



We can notice that the issue of non-constant variance is particularly present in the plots of the predictors of latent heat flux (LH) and wind speed (WS). The former could be attributed to the observation previously made on the seemingly non-linear relationship between the predictor variable LH and the dependent variable (see Assignment 1).

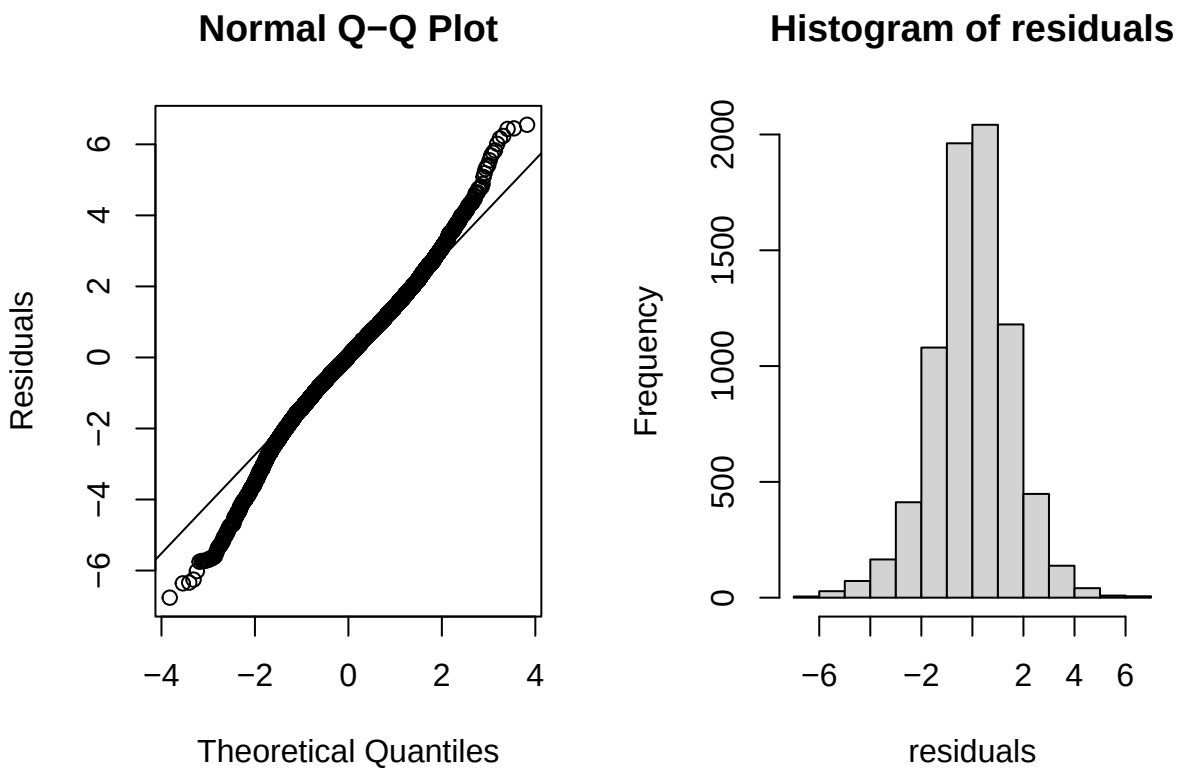
1.2.2 Structure of the relationship between the predictors and the response

The residuals are not perfectly linear, as shown by the red line in the first plot, although the curvatures are not too wide.

1.2.3 Normality assumption

We can use a Q-Q plot and a histogram to check whether the residuals are normally distributed. The residuals of the fitted model seem to have a short-tailed distribution, which indicates some departure from normality, particularly in the tails. However, upon inspecting the histogram of the residuals, it appears symmetric and bell-shaped, suggesting a nearly normal distribution. In any case, a short-tailed distribution does not require adjustments.

```r
par(mfrow = c(1, 2))
qqnorm (residuals (model2), ylab = "Residuals")
qqline (residuals (model2))
hist(residuals)
```

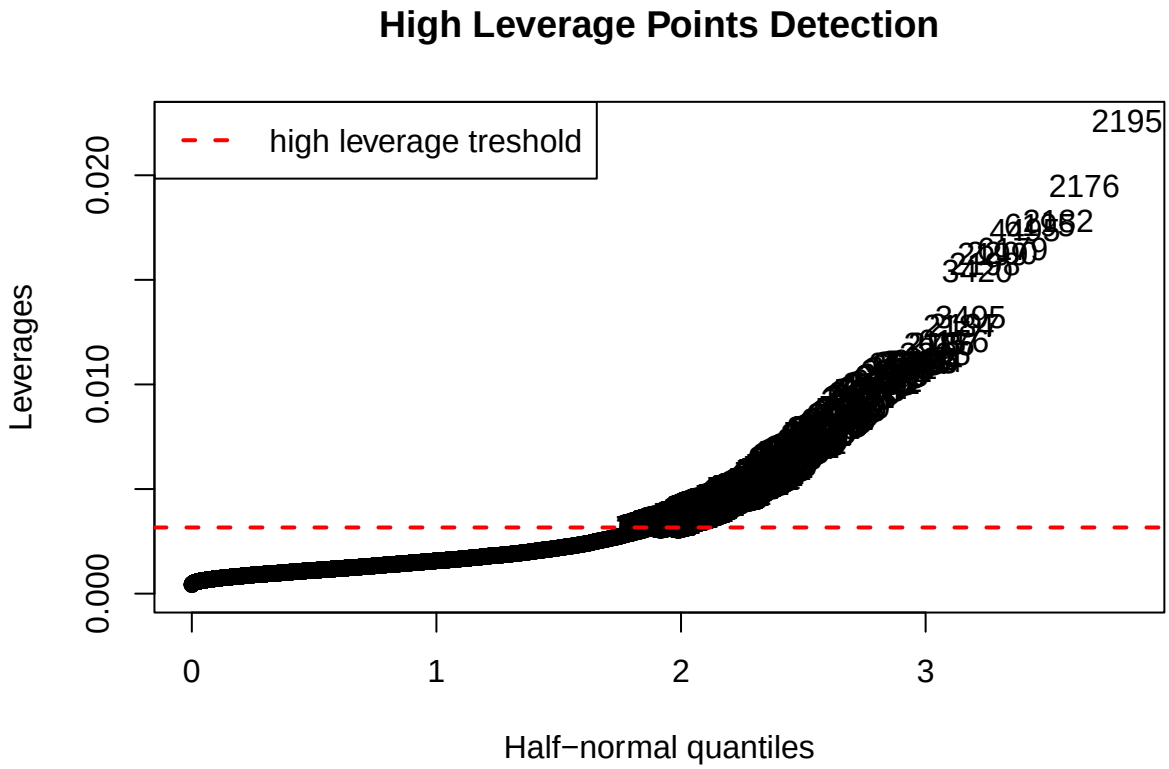


1.2.4 Large leverage points

The following plot is used to show the high leverage points in our residuals. By setting the threshold to the value 2*(p+1)/n, we can detect 459 high leverage points.

```r
infl = influence(model2)
hat = infl$hat
n_hat = hat[which(hat >= (2 * 12 / nrow(data)))]
halfnorm(hat,
         459,
         labs = rownames(data),
         ylab = "Leverages",
         main = "High Leverage Points Detection")
abline(
  h = 2 * 12 / nrow(data),
  col = 'red',
  lty = 2,
  lwd = 2
)
legend(
  x = "topleft",
```

```
  legend = "high leverage treshold",
  lty = 2,
  lwd = 2,
  col = 'red'
)
```



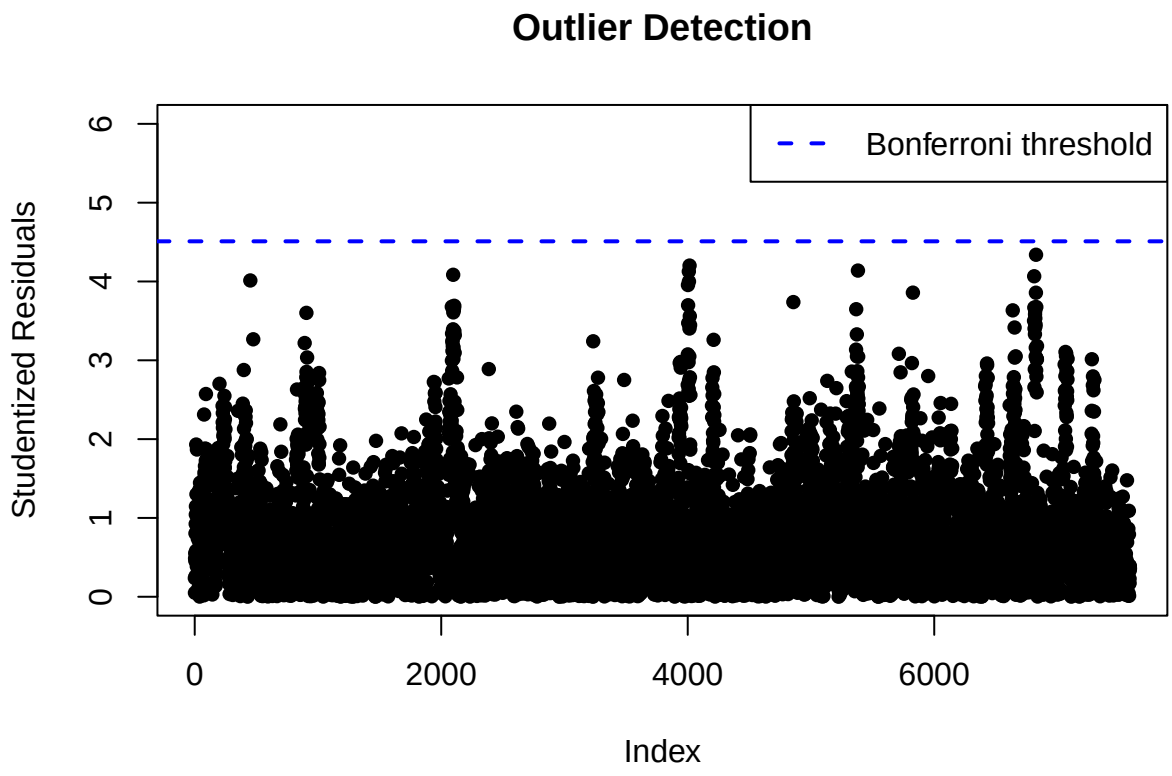
1.2.5 Outliers

To test if any of the datapoints are outliers, we can resort to the Bonferroni method. This approach involves adjusting the significance level (alpha) of hypothesis tests to account for multiple comparisons. In the context of outlier detection, the Bonferroni method entails setting a stricter threshold for outlier identification: we use a central student-t distribution with n − p − 1 degrees of freedom, evaluated in the nominal significance level (in this test equal to 0.05) divided by the number of comparisons being made (i.e. the number of residual observations) multiplied by 2. Any observations with p-values below this adjusted threshold are considered outliers.

```
#Bonferroni correction
alpha = 0.05
res_standard = rstandard(model2)
n = 7588

bonferroni_quantile = qt(1 - alpha / (2 * n), df = model2$df.residual)
bonferroni_quantile
```

```
## [1] 4.50971
```

```r
plot(
  abs(res_standard),
  pch = 16,
  ylim = c(0, 6),
  main = "Outlier Detection",
  ylab = "Studentized Residuals"
)
abline(
  h = bonferroni_quantile,
  col = 'blue',
  lty = 2,
  lwd = 2
)
legend(
  "topright",
  legend = c("Bonferroni threshold"),
  lty = c(2, 2),
  lwd = c(2, 2),
  col = c('blue')
)
```



None of the observations can be identified as an outlier.

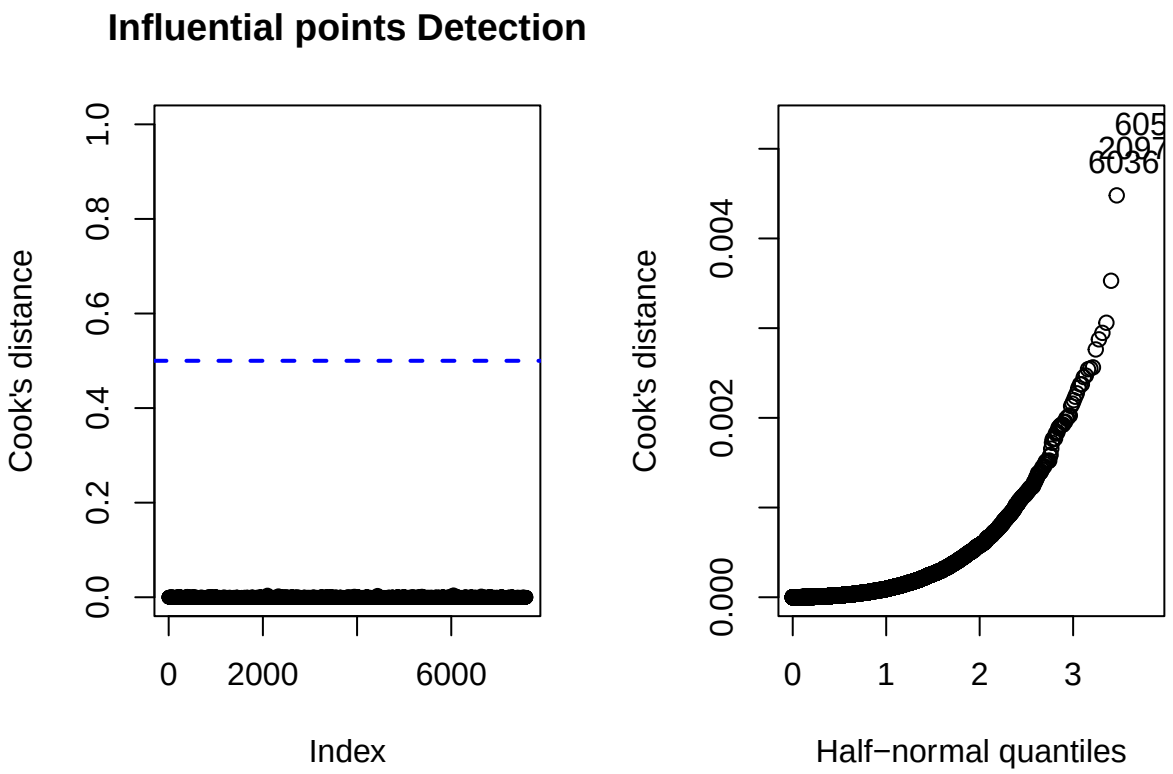
1.2.6 Influential points

The detection of influential points is crucial for understanding the impact of individual observations on the model's fit and estimates. Cook's distance, a commonly employed metric for identifying influential points, quantifies the influence of each observation by measuring how much the model's parameters change when the observation is excluded. In this particular analysis, Cook's distance was applied to the residuals of the model to identify any influential points. However, none of the observations were flagged as influential based on the calculations, with a threshold equal to 0.05. This indicates that no individual data point exerted a disproportionately large influence on the model's estimates or fit, since the highest Cook value is 0.00527.

```
par(mfrow = c(1, 2))
cook <- cooks.distance(model2)
plot(
  cook,
  pch = 16,
  ylim = c(0, 1),
  ylab = "Cook's distance",
  main = "Influential points Detection"
)
abline(
  h = 0.5,
  col = 'blue',
  lty = 2,
  lwd = 2
)
cook[which.max(cook)]
```

```
##        6170
## 0.005272179
```

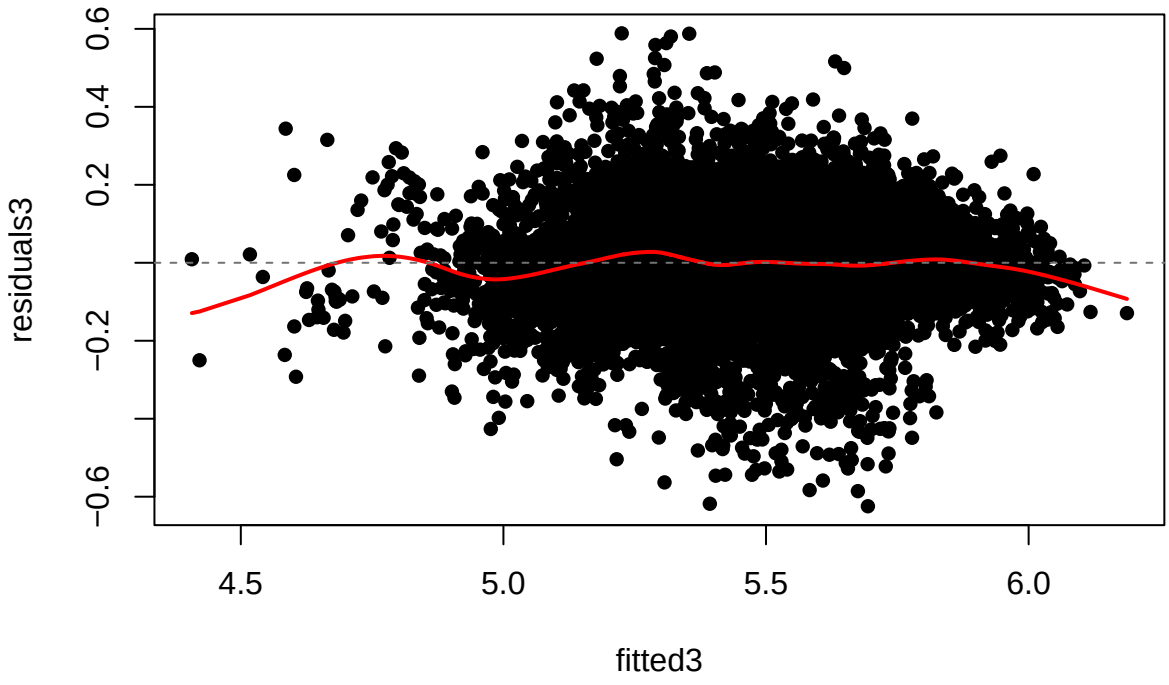
```
halfnorm(cook, 3, ylab = "Cook's distance")
```



1.3 Improved model

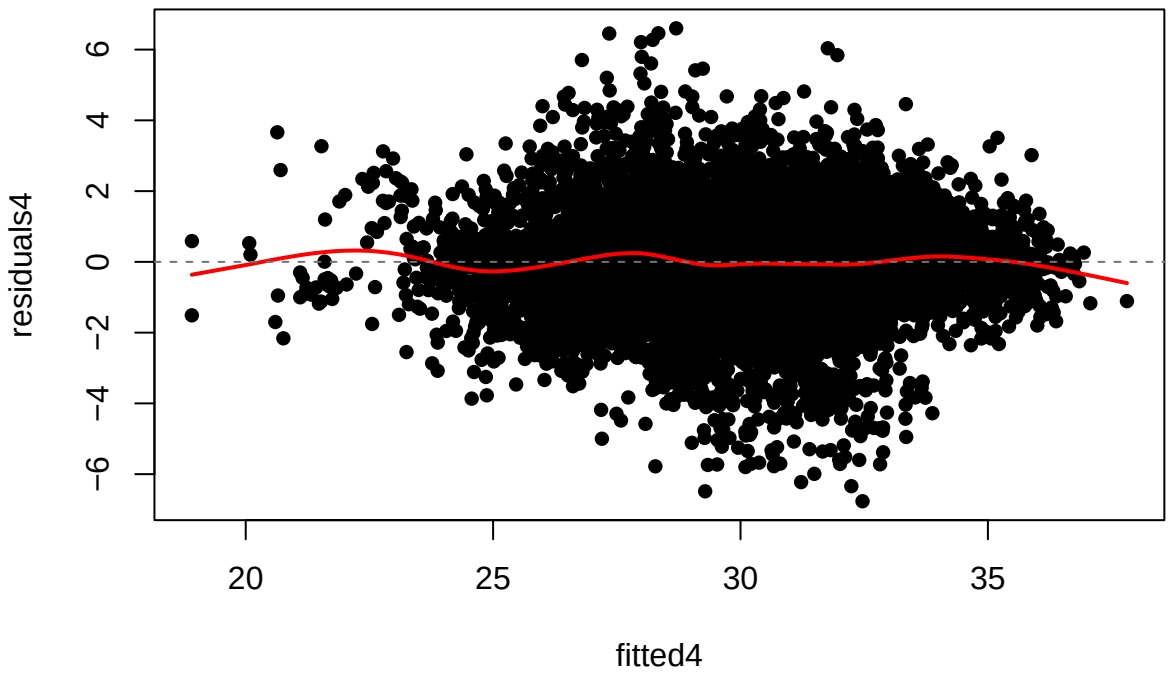
The model should be corrected to have a constant variance of the residuals, which exhibit a heteroscedastic behavior. We can improve the model by applying a variance-stabilizing transformation of the dependent variable. Among various transformations (logarithmic, inverse, arcsine square root), the most visible results were obtained with the square root transformation of the predicted variable, although the center of the plot still resulted fuller than the tails.

```
data$transformed_Next_TMax = (data$Next_Tmax) ^ 0.5
model3 = lm(data$transformed_Next_TMax ~ Tmax_lapse + RHmin + CC + LH + WS * grouped_station,
            data = data)

residuals3 = residuals(model3)
fitted3 = fitted.values(model3)
res_vs_fitted3 = smooth.spline(x = fitted3, y = residuals3)
plot(fitted3, residuals3, pch = 16)
lines(res_vs_fitted3, col = "red", lwd = 2)
abline(h = 0,
       col = "grey45",
       lty = 2)
```

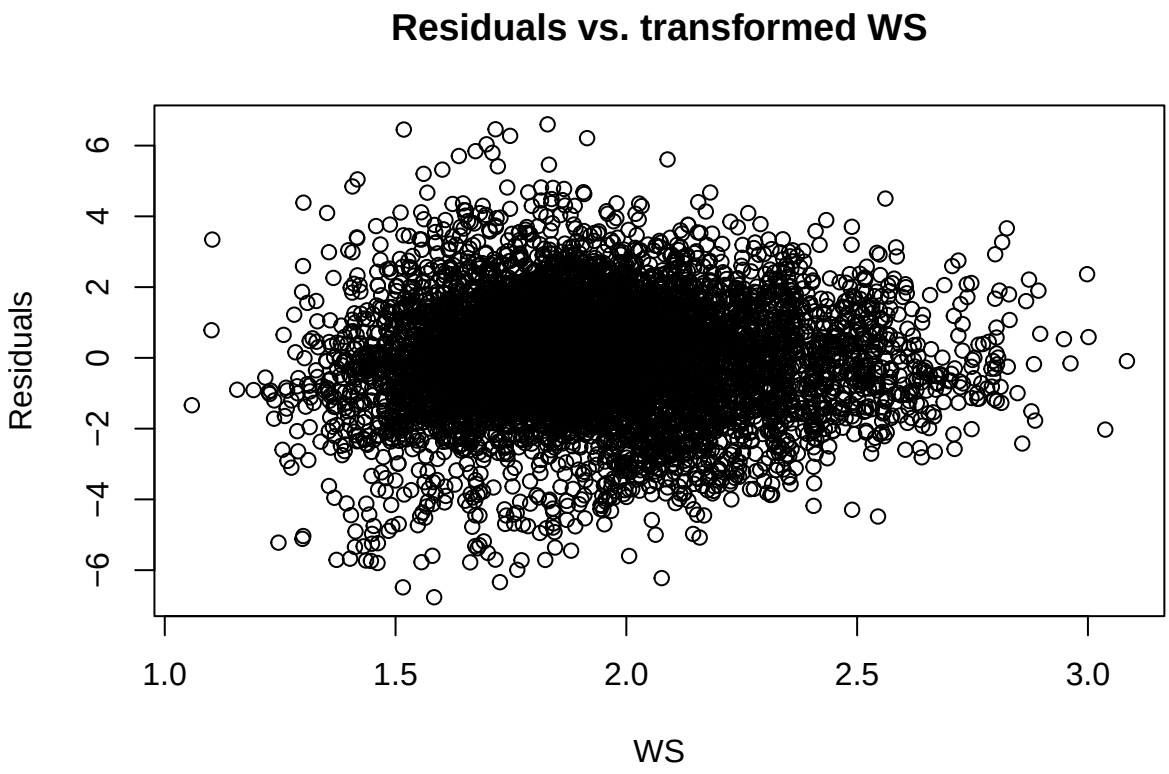


A second logarithmic transformation was attempted on the predictor WS, to improve the linearity of the residuals. this second model was quite effective in correcting the curvatures of the first fitted model and in improving the non-constant variance issue. We will perform the following diagnostics on this model. Indeed, the shape of the plotted residuals against the selected variable has improved.

```
data$transformed_WS = log(data$WS)

model4 = lm(
  data$Next_Tmax ~ Tmax_lapse + RHmin + CC + LH + data$transformed_WS * grouped_station,
  data = data
)

residuals4 = residuals(model4)
fitted4 = fitted.values(model4)
res_vs_fitted4 = smooth.spline(x = fitted4, y = residuals4)
plot(fitted4, residuals4, pch = 16)
lines(res_vs_fitted4, col = "red", lwd = 2)
abline(h = 0,
       col = "grey45",
       lty = 2)
```



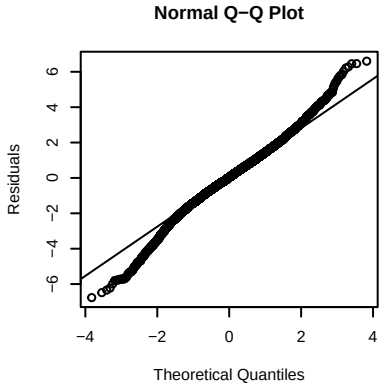

```
plot(
  residuals(model4) ~ data$transformed_WS,
  main = "Residuals vs. transformed WS",
  xlab = "WS",
  ylab = "Residuals"
)
```

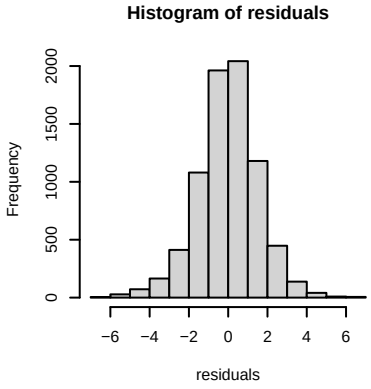


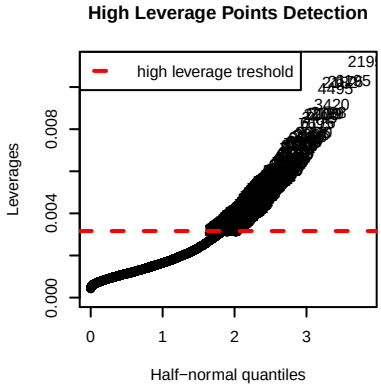
We perform the same diagnostic analysis on the selected model. We conclude that the previously made observations are still valid for this new model.

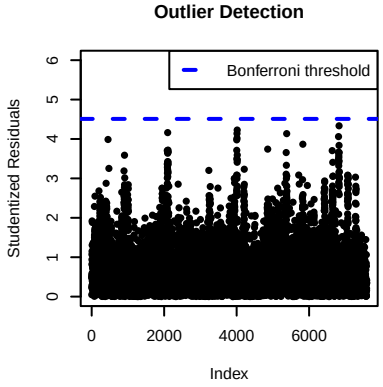
```
par(mfrow = c(2, 3), cex = 0.5)
qqnorm (residuals (model4), ylab = "Residuals")
qqline (residuals (model4))
hist(residuals)

infl = influence(model4)
hat = infl$hat
n_hat = hat[which(hat >= (2 * 12 / nrow(data)))]
halfnorm(hat,
         510,
         labs = rownames(data),
         ylab = "Leverages",
         main = "High Leverage Points Detection")
abline(
  h = 2 * 12 / nrow(data),
  col = 'red',
  lty = 2,
  lwd = 2
)
legend(
  x = "topleft",
  legend = "high leverage treshold",
  lty = 2,
  lwd = 2,
```

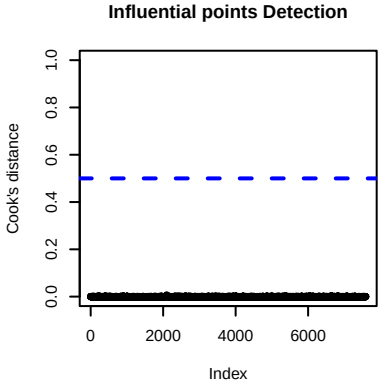
```
  col = 'red'
)

alpha = 0.05
res_standard = rstandard(model4)

bonferroni_quantile = qt(1 - alpha / (2 * n), df = model4$df.residual)

plot(
  abs(res_standard),
  pch = 16,
  ylim = c(0, 6),
  main = "Outlier Detection",
  ylab = "Studentized Residuals"
)
abline(
  h = bonferroni_quantile,
  col = 'blue',
  lty = 2,
  lwd = 2
)
legend(
  "topright",
  legend = c("Bonferroni threshold"),
  lty = c(2, 2),
  lwd = c(2, 2),
  col = c('blue')
)

cook <- cooks.distance(model4)
plot(
  cook,
  pch = 16,
  ylim = c(0, 1),
  ylab = "Cook's distance",
  main = "Influential points Detection"
)
abline(
  h = 0.5,
  col = 'blue',
  lty = 2,
  lwd = 2
)
```


Normal Q−Q Plot
Residuals
Theoretical Quantiles


Histogram of residuals
Frequency
residuals


High Leverage Points Detection
high leverage treshold
Leverages
Half−normal quantiles


Outlier Detection
Bonferroni threshold
Studentized Residuals
Index


Influential points Detection
Cook's distance
Index