Linear Models, Non-assessed Practical

Week 3, MT 2024

In addition to the exercises below, there is plenty of other material to practise on – all of the R examples from lectures.

The data set we are considering today describes a cloud seeding experiment aimed at increasing rainfalls, taken from Cook and Weisberg's "Residuals and Inference in Regression" book. It used silver iodide as a catalyst to induce rain, and targeted an area of 3000 square miles north-east of Coral Gable, California for 24 days in the summer of 1975. The following variables were recorded:

- Action (`A`): a classification indicating *seeding* (coded `1`) or *no seeding* (coded `0`).
- Time (`T`): days after the beginning of the experiment.
- Suitability (`SNe`): if `SNe` $\geqslant 1.5$ the day was judged suitable for seeding based on natural conditions.
- Echo coverage (`C`): per cent cloud cover in the area, measured using radar.
- Pre-wetness (`P`): total rainfall in the target area.
- Echo motion (`E`): a classification indicating a moving radar echo (coded `1`) or a stationary radar echo (coded `2`).
- Response (`Y`): amount of rain (in $10^7 m^3$) that fell in the area for a 6-hours period on each suitable day.

Several `R` functions will be suggested and used for the analysis, please use the help (`?fun` or `help("fun")`) to make yourself familiar with them as needed.

Exercise 1: Importing and Exploring the Data

1. Load the data from the file `cloud.seeding.txt`.

   The file is on Canvas, as well as at `http://www.stats.ox.ac.uk/~laws/SB1/data/cloud.seeding.txt`

2. Print the first few lines of the data and explore variable types.

3. Which variables appear to be related to the response variable, and thus may be good choices for an explanatory variable in a linear model? [Use `cor()`.]

4. Perform a graphical inspection of the relationship between the response `Y` and the other variables. Does any variable show a definite trend?

5. Transform `A` and `E` into factors with `as.factor()`. Is `Y` distributed differently for the level of each of these variables?

Solution to Exercise 1

[*Note that the "solutions" here are notes on what you might do, they are not written in the form of a report and they do not constitute a model report.*]

1. To load the data into `R`, use `read.table()`. Note that the first line contains the variable names, so `header = TRUE` must be used to import the data correctly.

```
cloud = read.table("cloud.seeding.txt", header = TRUE)
```

2. Variables types can be printed using `summary()`, `class()` and other functions, but the quickest way is to call `str()` on the whole data frame; and subsetting can be done with `cloud[1:10, ]` or with `head`.

```
head(cloud)
```

```
##   A T  SNe    C     P E     Y
## 1 0 0 1.75 13.4 0.274 2 12.85
## 2 1 1 2.70 37.9 1.267 1  5.52
## 3 1 3 4.10  3.9 0.198 2  6.29
## 4 0 4 2.35  5.3 0.526 1  6.11
## 5 1 6 4.25  7.1 0.250 1  2.45
## 6 0 9 1.60  6.9 0.018 2  3.61
```

```
str(cloud)
```

```
## 'data.frame': 24 obs. of  7 variables:
##  $ A  : int  0 1 1 0 1 0 0 0 0 1 ...
##  $ T  : int  0 1 3 4 6 9 18 25 27 28 ...
##  $ SNe: num  1.75 2.7 4.1 2.35 4.25 1.6 1.3 3.35 2.85 2.2 ...
##  $ C  : num  13.4 37.9 3.9 5.3 7.1 6.9 4.6 4.9 12.1 5.2 ...
##  $ P  : num  0.274 1.267 0.198 0.526 0.25 ...
##  $ E  : int  2 1 2 1 1 2 1 1 1 1 ...
##  $ Y  : num  12.85 5.52 6.29 6.11 2.45 ...
```

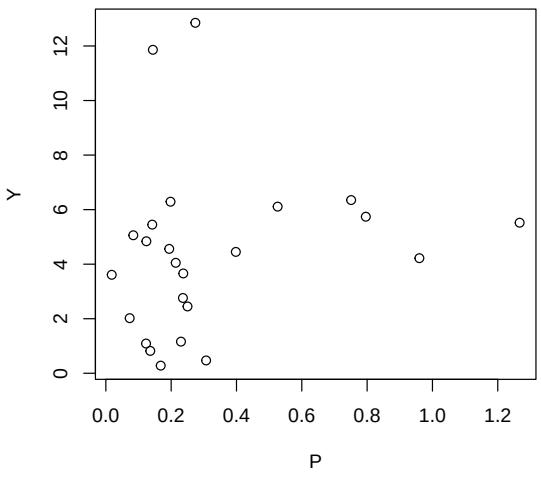
3. We can produce the whole correlation matrix of `cloud` with `cor` and then sort the column (or the row) referring to `Y`.

```
cormat = cor(cloud)
sort(cormat[, "Y"], decreasing = TRUE)
```

```
##      Y      E      C      P      A    SNe      T
##  1.000  0.332  0.270  0.174  0.076 -0.408 -0.496
```

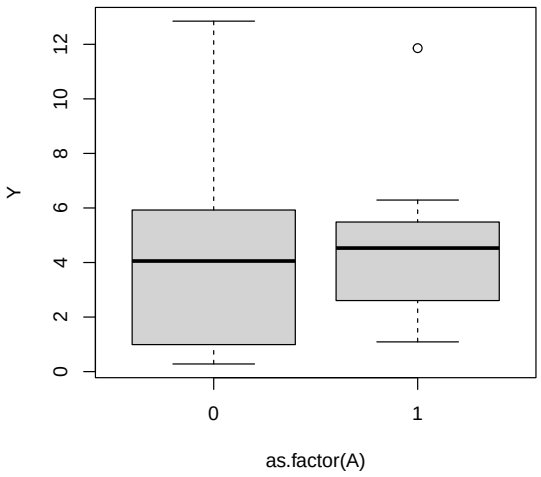
4. A graphical inspection can be carried out simply by calling `plot(cloud)`, but to have larger (and more readable) plots it is better to call `plot` individually for each variable, *e.g.*

```
plot(Y ~ P, data = cloud)
```

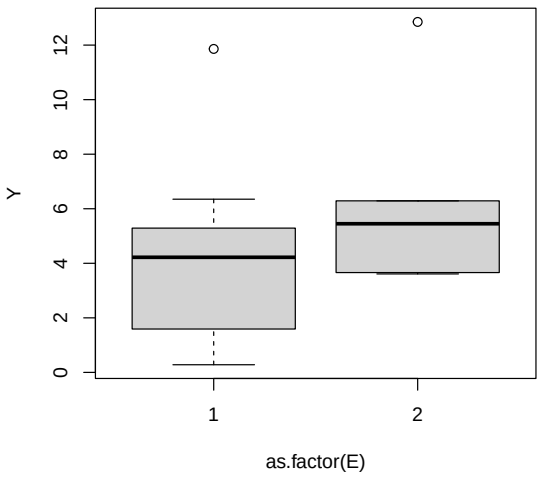


5. After transforming `A` and `E`, there is some evidence that Y assumes different values for different levels of E, not so much for `A`.

```
plot(Y ~ as.factor(A), data = cloud)
```




```
plot(Y ~ as.factor(E), data = cloud)
```



Exercise 2: Model Estimation

1. Fit a simple linear regression using `Y` as the response variable and `T`; save the model in an object called `mT`; and extract regression coefficients, residuals and fitted values.

2. Describe the main quantities present in the output of `summary(mT)`.

3. Is there any evidence that the rainfalls are increasing with time? Use the regression coefficient for `T` to assess whether there is any significant relationship between `Y` and `T`.

4. Now perform a simple linear regression using first `C`, and then `P`, and save them respectively as `mC` and `mP`. Are the respective regression coefficients significant?

5. Try a few transformations of `C`, such as log(`C`) and $\texttt{C}^2$, and then do the same for of `P`; does the model fit the data any better? Does it make sense to compare models after transforming the explanatory variable? [Consider R^2 values.]

6. Now transform `Y` into log(`Y`) and fit a simple linear regression using `C` as the explanatory variable. Does it make sense to compare (using R^2, or the residual standard error) how this model fits compared to previous models?

7. Fit a multiple linear regression with `Y` as the response and `T`, `C` and `P` as explanatory variables, and save it into an object called `mCPT`. Are the regression coefficients the same as in the simple linear regressions fitted above? Why?

8. Include the `A` variable into the previous model, coded as a factor. Describe how it is coded as a contrast. Does it appear to be significant?

9. Fit a model which also includes interaction terms between `A` and the other variables, and describe the resulting set of regression coefficients. [Use `summary()`.]

Solution to Exercise 2

1.
```
mT = lm(Y ~ T, data = cloud)
coef(mT)

## (Intercept)           T
##       6.559      -0.061
```

```
resid(mT)
```

```
##       1       2       3       4       5       6       7
##  6.2907 -0.9783 -0.0862 -0.2052 -3.7431 -2.4000 -4.9908
##       8       9      10      11      12      13      14
## -0.4736  1.4385  0.2095 -2.0294 -0.5564  1.1947  0.4167
##      15      16      17      18      19      20      21
##  7.6198  0.2709  0.3353  1.0173 -1.9816  2.4915 -0.5724
##      22      23      24
## -1.5893 -0.4648 -1.2138
```

```
fitted(mT)
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12
## 6.56 6.50 6.38 6.32 6.19 6.01 5.46 5.03 4.91 4.85 4.79 4.61
##   13   14   15   16   17   18   19   20   21   22   23   24
## 4.55 4.42 4.24 4.18 3.32 3.20 3.14 2.96 2.59 2.41 1.55 1.49
```

2. The main quantities printed by `summary()` are

   - the model formula and the `R` function call used to fit the model (`Call`);
   - the quartiles of the residuals, along with maximum and minimum (`Residuals`);
   - the regression coefficients and their p-values for the t-tests (`Coefficients`);
   - the standard error and the R^2 coefficient, as measures of goodness of fit.

```
summary(mT)
```

```
##
## Call:
## lm(formula = Y ~ T, data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.991 -1.308 -0.335  0.567  7.620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5593     0.9820    6.68    1e-06 ***
## T            -0.0610     0.0228   -2.68    0.014 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.76 on 22 degrees of freedom
## Multiple R-squared:  0.246, Adjusted R-squared:  0.212
## F-statistic: 7.19 on 1 and 22 DF,  p-value: 0.0136
```

3. The regression coefficient is negative, so rainfalls are definitely not increasing as the number of days after the beginning of the experiment increases. We can see from `summary()` that the p-value of the regression coefficient is smaller that the customary $\alpha = 0.05$, so we can conclude that the relationship between `Y` and `T` is significant at $p = 0.014$.

4. From the output of `summary(mC)` and `summary(mP)`, we can see that neither regression coefficient is significant: that for `C` has p-value $p = 0.2015$, and that for `P` has p-value $p = 0.417$.

```r
mC = lm(Y ~ C, data = cloud)
summary(mC)
```

```
##
## Call:
## lm(formula = Y ~ C, data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.14  -2.07  -0.11   1.20   7.73
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.5558     0.8968    3.96  0.00066 ***
## C             0.1169     0.0888    1.32  0.20151
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.06 on 22 degrees of freedom
## Multiple R-squared:  0.073, Adjusted R-squared:  0.0309
## F-statistic: 1.73 on 1 and 22 DF,  p-value: 0.202
```

```r
mP = lm(Y ~ P, data = cloud)
summary(mP)
```

```
##
## Call:
## lm(formula = Y ~ P, data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.898 -1.851 -0.208  1.112  8.539
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.837      0.936    4.10  0.00047 ***
## P              1.729      2.091    0.83  0.41709
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.13 on 22 degrees of freedom
## Multiple R-squared:  0.0302, Adjusted R-squared:  -0.0139
## F-statistic: 0.684 on 1 and 22 DF,  p-value: 0.417
```

5. In the case of `C`, a few transformations provide a better R^2 (an thus a lower residual standard error) than the un-transformed variable; but all models still fit the response poorly.

   Note: to regress `Y` on $\texttt{C}^2$ we need to use `I(C^2)` (because `^` behaves differently to usual when it's within a model formula).

```r
summary(lm(Y ~ C, data = cloud))$r.squared
```

```
## [1] 0.073
```

```r
summary(lm(Y ~ log(C), data = cloud))$r.squared
```

```
## [1] 0.156
```

```r
summary(lm(Y ~ I(C^2), data = cloud))$r.squared
```

```
## [1] 0.0242
```

```
summary(lm(Y ~ sqrt(C), data = cloud))$r.squared

## [1] 0.117
```

In the case of `P`, no transformation seems to make an appreciable difference.

```
summary(lm(Y ~ P, data = cloud))$r.squared

## [1] 0.0302
```

```
summary(lm(Y ~ log(P), data = cloud))$r.squared

## [1] 0.033
```

```
summary(lm(Y ~ I(P^2), data = cloud))$r.squared

## [1] 0.0205
```

```
summary(lm(Y ~ sqrt(P), data = cloud))$r.squared

## [1] 0.0345
```

Comparing these models makes sense because goodness of fit is assessed through the residuals, which are on the same scale in all models because `Y` is not transformed.

6. According to R^2, this model is a poorer fit than the original.

```r
summary(lm(log(Y) ~ C, data = cloud))
```

```
##
## Call:
## lm(formula = log(Y) ~ C, data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.441 -0.350  0.300  0.547  1.317
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9395     0.2760    3.40   0.0025 **
## C             0.0309     0.0273    1.13   0.2700
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.942 on 22 degrees of freedom
## Multiple R-squared:  0.055, Adjusted R-squared:  0.012
## F-statistic: 1.28 on 1 and 22 DF,  p-value: 0.27
```

It does not make sense to use the residual standard error to compare this model to the previous ones, because the residual standard error depends on the scale used (i.e. whether `Y` or `log(Y)`) . This is also true for R^2; a fair comparison would require transforming fitted values and the response variable back to the original scale. However, since R^2 is a correlation and therefore is standardised, which makes the difference in scale less obvious and (in practice) negligible unless it is very large.

7. The regression coefficients for the three explanatory variables are not the same as in the simple linear regressions in which said variables were originally used. This is because the explanatory variables are not perfectly orthogonal (and independent), and thus the estimated regression coefficients are correlated with each other. Adding or removing one can potentially change all the others.

```r
mCPT = lm(Y ~ C + P + T, data = cloud)
coef(mCPT)
```

```
## (Intercept)           C           P           T
##      6.1003      0.0574     -0.3567     -0.0565
```

8. The most convenient way to re-code `A` is to call `as.factor()` and save the re-coded variable in the `cloud` data frame. The resulting factor has two levels, `0` and `1`. The former is merged into the intercept as the reference level, the latter is coded as a contrast that shows up in the model as `A1`.

```
cloud[, "A"] = as.factor(cloud[, "A"])
summary(lm(Y ~ A + C + P + T, data = cloud))
```

```
##
## Call:
## lm(formula = Y ~ A + C + P + T, data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.541 -1.354 -0.082  0.751  7.325
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.8794     1.5752    3.73   0.0014 **
## A1            0.4979     1.2070    0.41   0.6846
## C             0.0518     0.1159    0.45   0.6599
## P            -0.2612     2.5882   -0.10   0.9207
## T            -0.0570     0.0259   -2.20   0.0403 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.93 on 19 degrees of freedom
## Multiple R-squared:  0.264, Adjusted R-squared:  0.109
## F-statistic: 1.71 on 4 and 19 DF,  p-value: 0.19
```

From the p-value of the t-test, we can say that `A` is not significant.

9. The model with the interactions can be written concisely as follows.

```
summary(lm(Y ~ A * (C + P + T), data = cloud))

##
## Call:
## lm(formula = Y ~ A * (C + P + T), data = cloud)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.620 -1.728  0.104  0.898  7.174
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6442     2.6343    1.00    0.330
## A1            3.1468     3.2325    0.97    0.345
## C             0.4884     0.2548    1.92    0.073 .
## P             1.0195     3.6845    0.28    0.786
## T            -0.0619     0.0343   -1.80    0.090 .
## A1:C         -0.4648     0.2859   -1.63    0.123
## A1:P         -1.7500     4.8694   -0.36    0.724
## A1:T          0.0312     0.0489    0.64    0.532
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.72 on 16 degrees of freedom
## Multiple R-squared:  0.466, Adjusted R-squared:  0.232
## F-statistic: 1.99 on 7 and 16 DF,  p-value: 0.12
```

It includes the main effects for all the explanatory variables, including the contrast coding `A`, and an additional regression coefficient for each of `C`, `P` and `T` that represents the interaction of each of these variables with `A`, which makes a contribution when `A` is equal to `1`.

Exercise 3: Model Validation

1. Consider again the model in the `mCPT` object, and call `par()` and `plot()` to plot all the diagnostic plots generated by `plot(mCPT)` in a single figure.

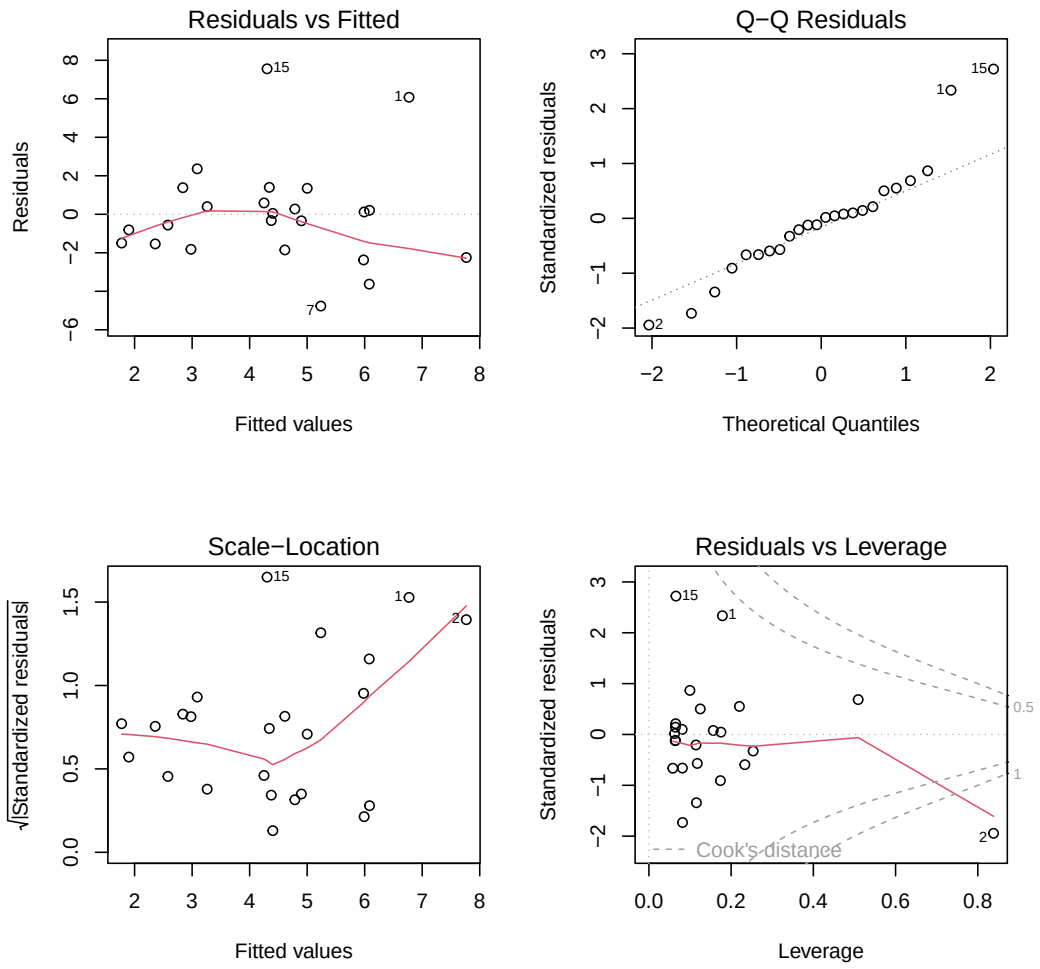
2. Look at the first and second plots: is there any reason to think that the `cloud` data violate the assumptions of the model?

3. Describe the concepts of leverage and influence. Now look at the last plots, locate observations that look problematic and comment on them.

4. Observations 1, 2 and 15 are labelled as possible outliers. Decide which of these to omit and fit the `mCPT` model again (i.e. without some/all of 1, 2, 15) and call it `mCPT2`; does this new model fit the remaining data better than before?

Solution to Exercise 3

1.
```
par(mfrow = c(2, 2))
plot(mCPT)
```



2. The "Residuals vs Fitted" plot does not indicate that the data violate the assumptions of the model; there is no trace of trends or heteroscedasticity in the plot. The standardised residuals are mostly very close to the corresponding theoretical quantiles, but the residuals of observations 1 and 15 are large.

3. See lectures. We could think of leverage as the potential that an observation has to affect the fitted model, e.g. it has more potential if $\mathbf{x}_i$ is an unusual value of $\mathbf{x}$. Whether observation i actually has a large effect, or not, also depends on y_i. The effect is the influence of observation i, measured in the plot of Cook's distances. (Leverages depend on X only; Cook's distances depend on X and y.)

   In the 4th plot, R plots contours of Cook's distances at $C_i = 0.5$ and $C_i = 1$ (these are alternative thresholds for C to the one in lectures).

   Observations 1, 2 and 15 appear to be possible outliers as they are close to the threshold of $C_i = 1$.

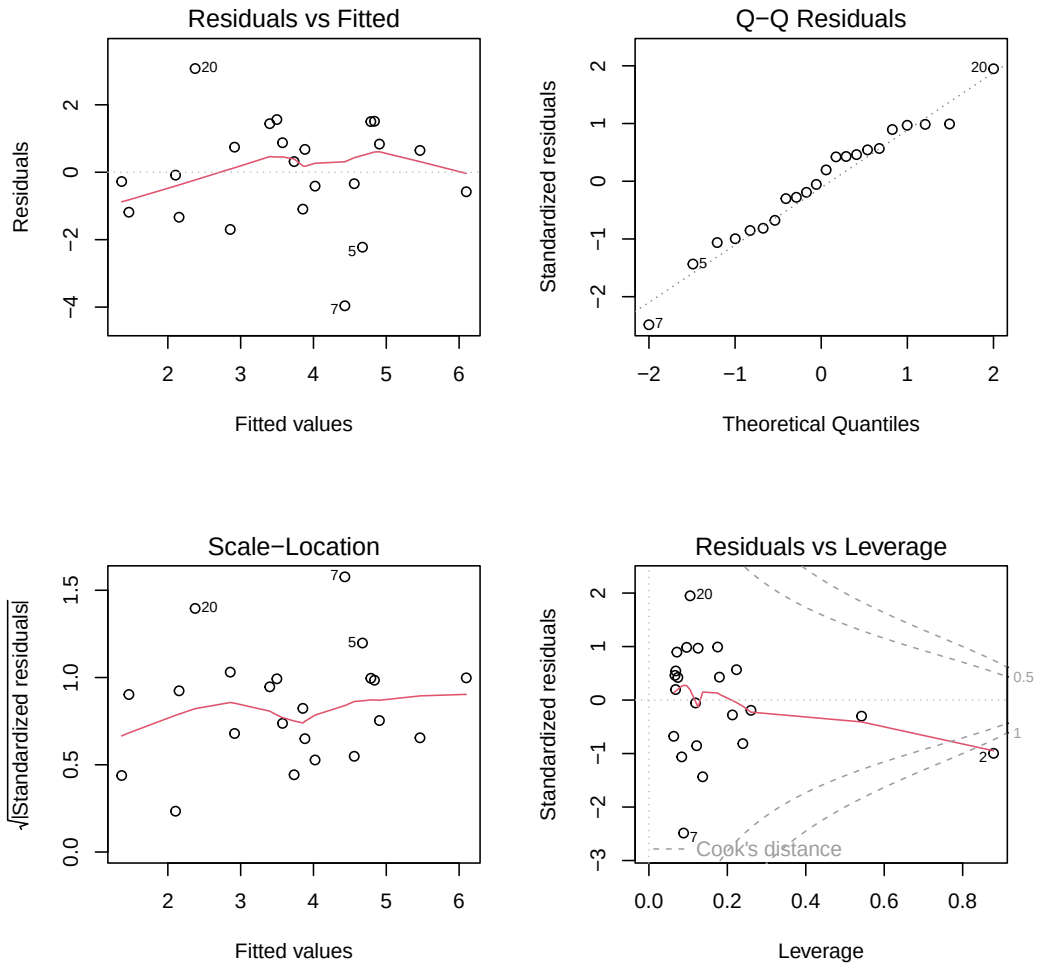
4. We can re-fit the `mCPT` model as follows.

```
mCPT2 = lm(Y ~ C + P + T, data = cloud[-c(1, 15), ])
summary(mCPT2)
```

```
##
## Call:
## lm(formula = Y ~ C + P + T, data = cloud[-c(1, 15), ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.961 -0.965  0.115  0.865  3.075
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5874     0.8817    5.20    6e-05 ***
## C            -0.0372     0.0674   -0.55     0.59
## P             2.3352     1.5281    1.53     0.14
## T            -0.0390     0.0153   -2.55     0.02 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.67 on 18 degrees of freedom
## Multiple R-squared:  0.403, Adjusted R-squared:  0.303
## F-statistic: 4.04 on 3 and 18 DF,  p-value: 0.0232
```

Looking at the `summary()` of the model, we can see that dropping the two observations makes an appreciable difference: the regression coefficients are not the same as before and all goodness-of-fit measures are much improved.

```
par(mfrow = c(2, 2))
plot(mCPT2)
```



Observation 2 is well aligned with the corresponding theoretical quantile, while observations 1 and 15 are not. This suggests that, even though extreme, observation 2 is consistent with the assumptions on the residuals of the model.