## 3.2 Unusual Observations

Before proceeding with the diagnostics for the model assumptions related to the error terms, we first need to detect whether any unusual observations are present. The reason is that least squares regression is very sensitive to individual data points, and it is possible that the inference, $p$-values, parameter estimation, and CI's are all driven by a single data point. Sometimes, the estimated parameters and other related statistics (such as $R^2$) depend heavily on one observation, in the sense that if that observation was removed, the result of the analysis would change. So, we need to understand first if there are any observations that may potentially skew our data.

In the sequel, we are going to discuss **three** types of unusual observations:

- High leverage points: The *leverages* quantify how far a data point is from the center of the whole sample.
- Outliers: *Outliers* are data points that do not fit the model as the other ones.
- Highly influential points: These are individual points that *affect* the model parameter estimators.

### 3.2.1 High Leverage Points

#### Leverage (Definition)

The diagonal elements of the hat matrix, $\mathbf{H}$,

$$h_i = H_{ii}$$

are called **leverages** and are very useful diagnostic tools.

The leverage $h_i$ gives a measure of how far the $i$-th observation is from the center of the data (in the $X$-space) and is invariant under any affine transformation of $\mathbf{X}$.

In the Simple Linear regression, the leverages compute as

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_i (x_i - \bar{x})^2}$$

while in the Multiple Linear regression their form is as follows:

$$\begin{aligned} h_i &= \mathbf{x_i^T}(\mathbf{X^TX})^{-1}\mathbf{x_i} \\ &= \frac{1}{n} + \frac{1}{n-1}(\mathbf{z}_i - \bar{\mathbf{z}})^T \hat{\Sigma}^{-1}(\mathbf{z}_i - \bar{\mathbf{z}}) \end{aligned}$$

where

$$\hat{\Sigma}^{-1}_{(p-1)\times(p-1)} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{z} - \bar{\mathbf{z}})(\mathbf{z} - \bar{\mathbf{z}})^T$$

is the sample covariance of the $(p-1)$ predictor variables. The term

$$\frac{1}{n-1}(\mathbf{z}_i - \bar{\mathbf{z}})^T \hat{\Sigma}^{-1}(\mathbf{z}_i - \bar{\mathbf{z}})$$

is the so-called Mahalanobis distance of $\mathbf{z}_i$ from the data center $\bar{\mathbf{z}}$.

## Properties of the Leverages

The leverages have several interesting properties that are derived from the properties of the hat matrix. Recall that the hat matrix is idempotent, i.e. $\mathbf{H} = \mathbf{H}\mathbf{H}^\top$, and has $tr(\mathbf{H}) = p$. These two imply that

$$\sum_i h_i = p \text{ and } \sum_j H_{ij}^2 = h_i.$$

For a given $i$ we can decompose the last sum as follows:

$$\sum_j H_{ij}^2 = H_{ii}^2 + \sum_{i \neq j} H_{ij}^2$$

Since we said that $\sum_j H_{ij}^2 = h_i$, we also have that

$$H_{ii}^2 + \sum_{i \neq j} H_{ij}^2 = h_i$$

Substituting $H_{ii}^2$ with $h_i^2$ in the above expression, we get

$$h_i^2 + \sum_{i \neq j} H_{ij}^2 = h_i$$

If we now solve with respect to $\sum_{i \neq j} H_{ij}^2$, we obtain

$$\sum_{i \neq j} H_{ij}^2 = h_i(1 - h_i)$$

It is easy to observe that the left hand side is positive, therefore we deduce that

$$h_i(1 - h_i) > 0$$

Taking into account the properties of the hat matrix, we conclude that

$$0 < h_i < 1$$

## Fitted Values and Leverage

Recall the equation $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$. In matrix form this is written as

$$\begin{pmatrix} \hat{y}_1 \\ \dots \\ \hat{y}_i \\ \dots \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} H_{11} & \dots & H_{1n} \\ \dots & \dots & \dots \\ H_{i1} & \dots & H_{in} \\ \dots & \dots & \dots \\ H_{n1} & \dots & H_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_n \end{pmatrix}$$

So, if we only extract the $i$th row, this writes as

$$\begin{aligned} \hat{y}_i &= H_{i1}y_1 + \dots + H_{ii}y_i + \cdots + H_{in}y_n \\ &= H_{i1}y_1 + \dots + h_i y_i + \cdots + H_{in}y_n \end{aligned}$$

Note that $\hat{y}_i$ is a linear combination of the $n$ data points:

$$\hat{y}_i = h_i y_i + \sum_{i \neq j} H_{ij} y_j$$

This means that $h_i = \frac{d\hat{y}_i}{dy_i}$.

When $h_i$ is large (close to 1), $\hat{y}_i$ relies **heavily** on $y_i$ (instead of using the information from other data points), and as a result $\hat{y}_i$ will be "forced" to be close to the observed $y_i$.

Consequently, the variance for the residual $r_i$ will be small, and the variance for the fit $\hat{y}_i$ will be large, since

$$Var\left(\hat{y}_i\right) = \sigma^2 h_i, \;\; Var\left(r_i\right) = \sigma^2(1 - h_i)$$

Using these observations we construct a rule-of-thumb to decide when an observation is a high-leverage point and when not.

### High-Leverage Points Rule-Of-thumb

Since $\sum_i h_i = p$, a rule-of-thumb is that observations with leverages more than $2p/n$ (twice the mean leverage) should be flagged as high-leverage points and should be examined closely.

The high-leverage points can be further classified to *good high-leverage points* and *bad high-leverage points*.

- Good high-leverage point: its $y$ point *follows the pattern* of the rest of the data, but with an $x$ value that is far away from the sample mean.
- Bad high-leverage point: its $y$ value *does not follow the pattern* suggested by the rest of the data; the LS fitting might change a lot if we remove this point.

### Leverages in Birthweight Study

Use the function `lm.influence` is the one to use to extract the leverages.

```
birthweight.leverages = lm.influence(birthweight.mlr1)$hat
head(birthweight.leverages)
```

```
##         1         2         3         4         5         6
## 0.3935661 0.1522053 0.4524529 0.3662474 0.2377534 0.1813611
```

The high leverage points are the ones that exceed the $2 * p/n$ threshold. In our case:

```
n = dim(birthweight2)[1]
p = length(variable.names(birthweight.mlr1))
```

In the code below, we extract the elements of the birthweight.leverages.high structure that exceed the 2p/n threshold:

```
# Determine which leverages exceed the 2p/n threshold
birthweight.leverages.high = birthweight.leverages[birthweight.leverages>2*p/n]
birthweight.leverages.high
```

```
## named numeric(0)
```

In this particular example, there are **no** high leverage points in the data, since none of the leverage values exceeds the $2p/n$ threshold.

### Half-Normal Plots

The half-normal plots are designed to identify unusually large values and assess positive data. The idea is that we plot the data against the positive normal quantiles. Specifically, the process that is followed is: 1. Sort the data:

$$x_{[1]} \leq \ldots \leq x_{[n]}.$$

2. Compute the quantiles:

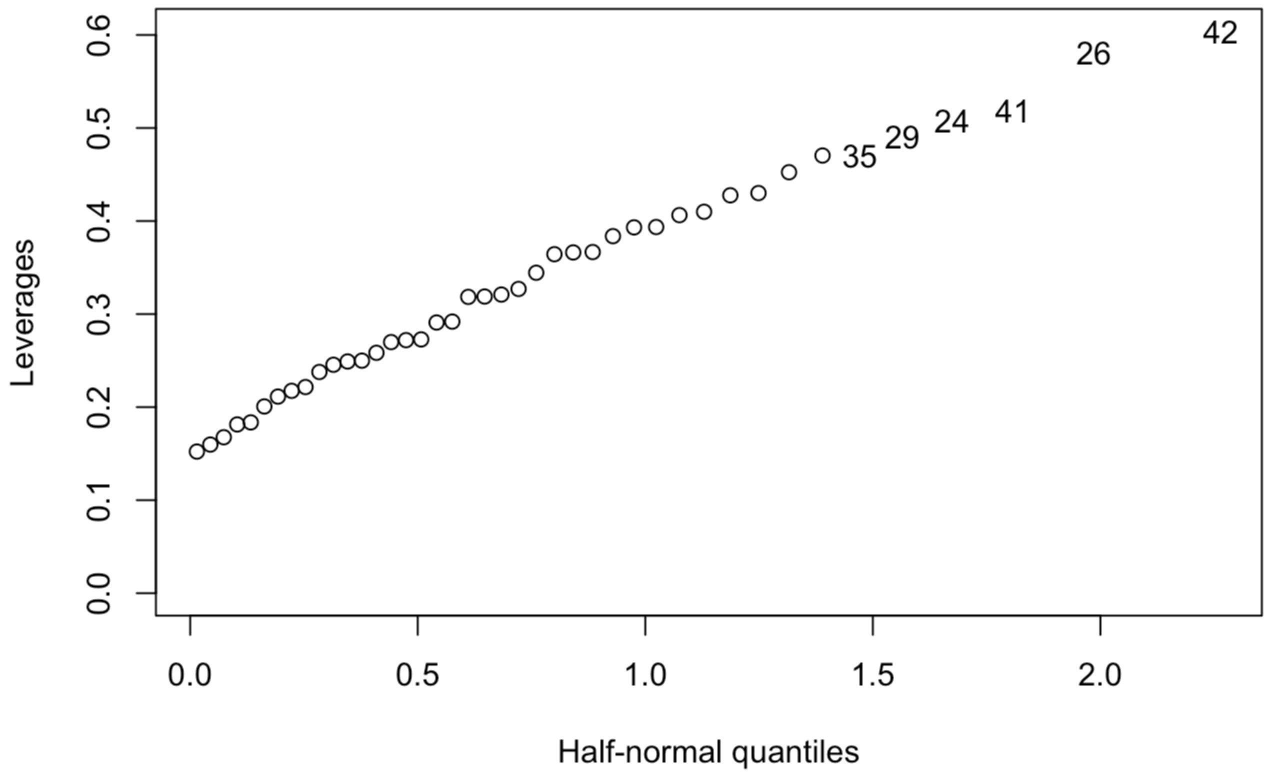
$$u_i = \Phi^{-1}\left(\frac{n+i}{2n+1}\right)$$

3. Plot $x_{[i]}$ against $u_i$.

### The Birthweight Example

The `halfnom` function is part of the `faraway` library:

```r
library(faraway)
halfnorm(birthweight.leverages, nlab=6, labs=as.character(1:length(birthweight.le
```



In the plot above, we do not necessarily look for a straight line fit. We use this plot to identify any leverages that do not follow the pattern suggested by the rest. In this case, none of the leverages seems to be unusually large.

In the following example, we will consider a reduced model in the `Birthweight` example for which we will be able to identify high leverage points. This example is for illustration purposes and will help us understand how to classify the high leverage points to good or bad in practice.

### A modified Birthweight Example

Consider the following model:

```
birthweight.mlr2 = lm(Birthweight ~ Headcirc + Gestation, data = birthweight2)
```

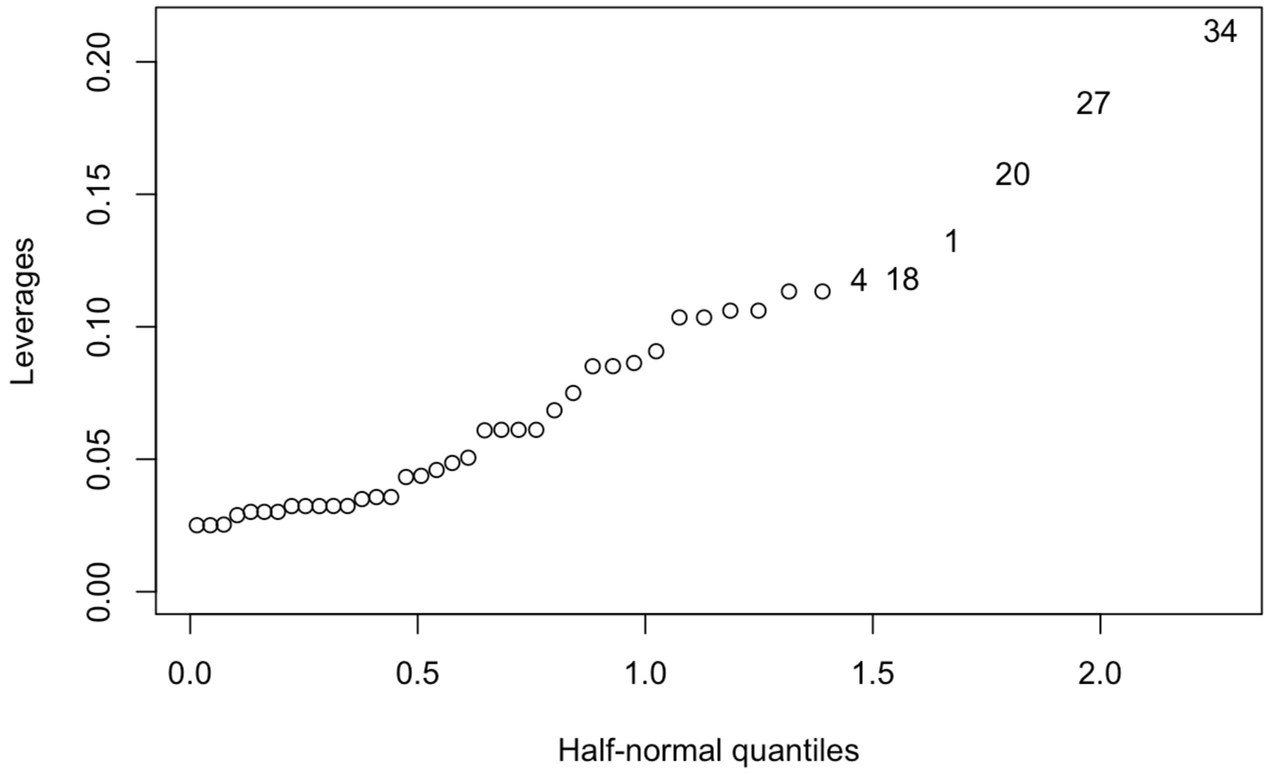
First, we compute the high leverage poins:

```
birthweight.leverages2 = lm.influence(birthweight.mlr2)$hat
head(birthweight.leverages2)
```

```
##          1          2          3          4          5          6
## 0.13280947 0.03230742 0.10607821 0.11814177 0.06090540 0.10351447
```

We can also take a look at the half-normal plot:

```
library(faraway)
halfnorm(birthweight.leverages2, 6, labs=as.character(1:length(birthweight.levera
```



Next, we determine which of the leverages exceed the $2p/n$ threshold. Note that this is a different model, so $p$ is different.

```
pnew = length(variable.names(birthweight.mlr2));
pnew
```

```
## [1] 3
```

```
birthweight.leverages.high2 = birthweight.leverages2[birthweight.leverages2>2*pne
birthweight.leverages.high2
```

```
##        20        27        34
## 0.1581021 0.1848649 0.2120588
```

```
# Sorting the high leverages in descending order
#birthweight.leverages.high2 = sort(abs(birthweight.leverages.high2), decreasing=
length(birthweight.leverages.high2)
```

```
## [1] 3
```

```
length(birthweight.leverages.high2)/n
```

```
## [1] 0.07142857
```

We observe that we have $3$ high leverage points, representing about $7\%$ of the observations. Observations $34$ and $27$ are far from the rest and are flagged in the halfnorm plot as well.

The next step is to classify these high leverage points between *good* or *bad*. *Good high-leverage points* are the ones for which the $y$ value follows the pattern of the rest of the data, but with an $x_i$ value that is far away from the sample mean, whereas *bad high-leverage points* are the ones for which the $y$ value does not follow the pattern suggested by the rest of the data, so the LS fitting might change a lot if we remove this point.

Basically, we're trying to identify which high-leverage points have a value of $y$ ( `Birthweight` , in our case) that don't follow the pattern suggested by the rest of the data. Let's calculate the IQR for our dependent variable `Birthweight` in our original (full) data frame and use this metric to identify the high-leverage observations that don't "follow the pattern of the data"; we'll do it by defining a range with lower bound being the first quantile minus IQR and the upper bound being the third quantile plus IQR. *The bad high-leverage points (among the 3 high-leverage points identified above) will be the ones that fall outside the range.*

```
# Calculate the IQR for the dependent variable
IQR_y = IQR(birthweight$Birthweight)

#Define a range with its lower limit being (Q1 - IQR) and upper limit being (Q3 +
QT1_y = quantile(birthweight$Birthweight,0.25)
QT3_y = quantile(birthweight$Birthweight,0.75)

lower_lim_y = QT1_y - IQR_y
upper_lim_y = QT3_y + IQR_y

vector_lim_y = c(lower_lim_y,upper_lim_y)

# Range for y variable
vector_lim_y
```

```
##    25%    75%
## 2.2325 4.3550
```

Now, we filter the data frame of high-leverage points and extract only the ones outside the range above. Those are the “*bad high-leverage points*”, whereas all the rest are our “*good high-leverage points*”:

```
# Extract observations with high leverage points from the original data frame
birthweight.highlev = birthweight2[birthweight.leverages2>2*pnew/n,]

# Select only the observations with leverage points outside the range
birthweight.highlev_lower = birthweight.highlev[birthweight.highlev$Birthweight <
birthweight.highlev_upper = birthweight.highlev[birthweight.highlev$Birthweight >
birthweight.highlev2 = rbind(birthweight.highlev_lower,birthweight.highlev_upper)
birthweight.highlev2
```

```
##    Length Birthweight Headcirc Gestation smoker mage mnocig mheight mppwt fage
## 27     48        1.92       30        33      1   20      7     161    50   20
##    fedyrs fnocig fheight lowbwt
## 27     10     35     180      1
```

Only observation $27$ can be considered a “bad high-leverage point”.

### 3.2.2 Residuals

The *plain* residuals $r_i = y_i - \hat{y}_i$ do **not** have a constant variance, since $Var(r_i) = \sigma^2(1 - h_i i)$. So, in order to stabilize their variance, they need to be standardized. There are two ways that we typically standardize the residuals:

- Standardized Residuals $r_i^*$: They are **internally** standardized. Under the model assumptions they follow *approximately* a Normal distribution.
- Studentized residuals $t_i$: They are **externally** standardized. They follow a $T$ distribution and will be used to construct an outlier test.

Remark: the residuals are the main ingredient used to build regression diagnostics. In the literature, it is recommended to use the standardized version of the residuals instead of the raw residuals in diagnostic plots.

#### Difference between $\varepsilon$ and $\mathbf{r}$

Recall the differences between the $\varepsilon$ (true error terms, our theoretical quantities and $\mathbf{r}$ the estimated residuals. Although both residuals are normally distributed,

$$\varepsilon \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$$

$$\mathbf{r} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2(\mathbf{I_n} - \mathbf{H})),$$

where $\mathbf{H}$ is the projection/hat matrix, we observe that the error terms $\varepsilon_i$’s have equal variance and are independent, while the residuals $r_i$'s have unequal variance and are correlated. Also, although $\mathbb{E}\left(\varepsilon\right) = \mathbb{E}\left(\mathbf{r}\right) = \mathbf{0}$, we discussed that

$$\sum_i \varepsilon_i \neq 0, \quad \text{while} \sum_i r_i = 0$$

if, of course, we assume that the intercept is included in the model.

## Standardized Residuals

Since $r_i \sim \mathcal{N}(0, \sigma^2(1-h_i))$, it is reasonable to consider a standardization of the residuals in this form:

$$r_i^* = \frac{r_i}{\hat{\sigma}\sqrt{1-h_i}}, \quad i = 1, \ldots, n$$

The **standardized** residuals no longer add up to zero, $\sum_i r_i^* \neq 0$, even when the intercept is included in the model. Furthermore, since $r_i$ **is not independent of** $\hat{\sigma}$, each $r_i^*$ is *not distributed as a $T$ distribution*. As an *approximation*, we can view the $r_i^*$'s as *iid* $\mathcal{N}(0,1)$ random variables, although they are **not** Normally distributed (exactly) and they are *slightly correlated*.

## Studentized Residuals

The **studentized** residuals are based on the idea of leave-one-out (also know as *jackknife* residuals). Here is the *leave-one-out idea*: 1. Run a regression model on the $(n-1)$ samples with the $i$-th sample $(x_i, y_i)$ removed.

2. Denote the leave-one-out estimates of the regression coefficient and error variance by $\hat{\beta}_{(i)}$ and $\hat{\sigma}_{(i)}$, where the notation $(i)$ means "excluding the $i$-th observation".

3. Then, check the **discrepancy** between observations $y_i$ and the fitted value $\hat{y}_{(i)} = \mathbf{x}^T \hat{\beta}_{(i)}$.

4. The Studentized Residual $i$ is defined as:

$$t_i = \frac{y_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}\left(1 + x_i^T (\mathbf{X}_{(\mathbf{i})}^{\mathbf{T}} \mathbf{X}_{(\mathbf{i})})^{-1} x_i\right)^{1/2}} = \frac{y_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}\sqrt{1-h_i}}$$

and can be shown that it follows a $T_{n-p-1}$ distribution if $y_i \sim \mathcal{N}(\mathbf{x}_i^T \beta, \sigma^2)$.

One can also show that $r_i^*$ and $t_i$ are a monotone transformation of each other:

$$t_i = r_i^* \left( \frac{n-p-1}{n-p-{r_i^*}^2} \right)^{1/2}$$

Using this property, we do not need to run the model $n$ times to get the estimates $\hat{\beta}_{(i)}$ and $\hat{\sigma}_{(i)}$, since we can simply use their relation with the standardized residuals and calculate them based on the $r_i^*$s.

## 3.2.3 Outliers

Although **Outliers** are observations that do not fit the model, they are not necessarily observations with large residuals.

An outlier test is a useful tool to distinguish observations that have large residuals from outliers. Therefore, we need to use the studentized residuals to construct an outlier test.

**Outlier Test**

Under the Null hypothesis $H_0$ that no outliers are present,

$$t_i \sim T_{n-p-1}$$

So we can use a simple $t$-test to decide whether the $i$-th observation is an outlier or not.

Generally, we would want to perform this outlier test for all $n$ observations, doing the tests one observation at a time. In order to be certain that the overall **type I error rate** is no greater than $\alpha$, the Bonferroni correction may be used. When doing so, each case would be tested at level $\alpha/n$.

Remark: If we perform the test on the largest observed residuals this would be an example of what we call data snooping, unless somehow these cases were identified before data collection.

### What we should do with outliers?

Once outliers are detected in our sample, then we need to decide how to treat them. In general, observations should not be routinely deleted simply because they do not fit the model. No data snooping! Outliers, as well as other unusual observations discussed here, often flag *potential problems* of the current model. In some sense, outliers maybe thought are normal points that haven't found their distribution yet. So, instead of dropping them, maybe, we should try a new alternative model.

#### Birthweight Study

We are going to check whether there are outliers present in the `birthweight` data set. We are going to use the `rstudent` function to obtain the studentized residuals, and the function `sort` to sort the residuals in decreasing order.

```
birthweight.resid = rstudent(birthweight.mlr1)
birthweight.resid.sorted = sort(abs(birthweight.resid), decreasing=TRUE)[1:10]
birthweight.resid.sorted
```

```
##       20        2       28        1       34       29       42       19
## 2.087824 2.053244 1.882769 1.833892 1.791036 1.712359 1.402567 1.366009
##        6       15
## 1.256217 1.216499
```

If we compute the critival $T_{n-p-1}$ value **with** Bonferroni correction, we have:

```
bonferroni_cv = qt(.05/(2*n), n-p-1)
bonferroni_cv
```

```
## [1] -3.622503
```

Above, we computed a t-value of $|-3.61|$ at $\alpha = 0.05$. If an observation's studentized residual is higher (in absolute value) than the critical value of the $T$ distribution with Bonferroni correction, then this observation will be considered an outlier. According to this criterion, we can see that **we don't have any outliers in the data set, since none of the studentized residuals is higher than $|-3.61|$.**

### 3.2.4 Influential Observations

**Influential Observations**, as the name suggests, are observations whose removal greatly affects the regression analysis.

An influential observation may (or may not) be an outlier or a high-leverage observation; or may be both: an outlier and a high-leverage observation.

We will use the Cook's distance to detect influential observations. Specifically,

$$D_i = \frac{||\mathbf{X}\hat{\beta} - \mathbf{X}\hat{\beta}_{(i)}||^2}{p\hat{\sigma}^2} = \frac{||\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)}||^2}{p\hat{\sigma}^2} = \frac{r_i^{*2}}{p}(\frac{h_i}{1-h_i})$$

From the expression of $D_i$, we can see that highly influential points are either outliers (large $|r_i^*|$) or high-leverage points (large $h_i$) or both.

**A rule-of-thumb to detect Influential Observations**

If the Cook's distance

$$D_i \geq 1$$

then observation $i$ is highly influential.

In the following example, we illustrate how the Cook's distance is computed in `R` :

**Birthweight Study**

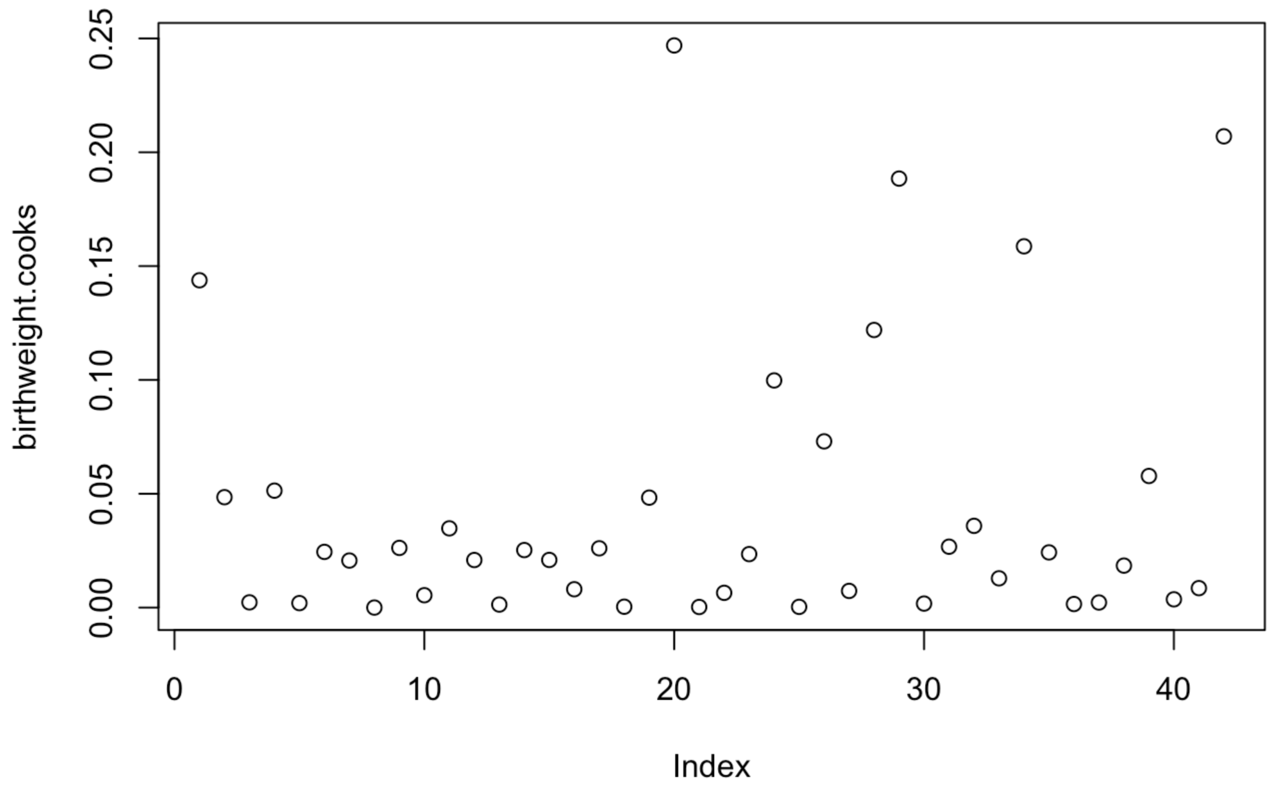
We compute all Cook's distances in the Birthweight data set.

```
birthweight.cooks = cooks.distance(birthweight.mlr1)
sort(birthweight.cooks, decreasing = TRUE)[1:10]
```

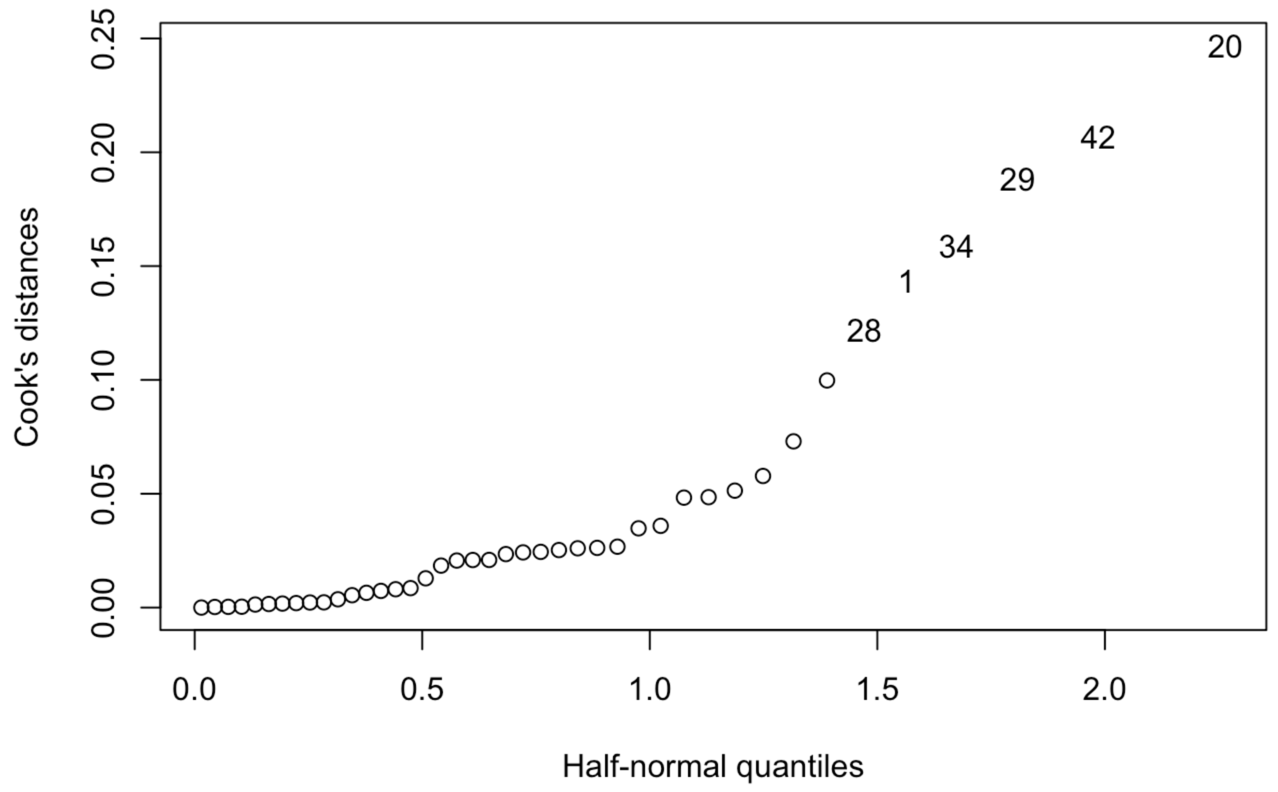
```
##         20         42         29         34          1         28         24
## 0.24692369 0.20699908 0.18843637 0.15868486 0.14376888 0.12193791 0.09977428
##         26         39          4
## 0.07301141 0.05780308 0.05136385
```

Based on the rule-of-thumb (Cook's distance $\geq 1$ for high influential points), we can see that there are no influential points in the data. Still, let's take a look at the vector with the Cook's distance calculated for every observation and prepare an index plot and a half-normal plot of the Cook's distances:

```
plot(birthweight.cooks)
```



```
halfnorm(birthweight.cooks, 6, labs=as.character(1:length(birthweight.cooks)), yl
```



Again, we see that there are no observations with Cook's Distance above 1.