



Statistical tools for high-throughput data analysis

Licence:

 [Home](#)[Basics](#)[Data](#)[Visualize](#)[Analyze](#)[Products](#)[Contribute](#)[Support](#)[About](#)

[Home](#) / [Articles](#) / [Machine Learning](#) / [Regression Model Diagnostics](#) / [Linear Regression Assumptions and Diagnostics in R: Essentials](#)

TRUE VALUE
HASSLE-FREE
DOCUMENTATION

QUALITY
PRE-OWNED CARS
KNOW MORE

Articles - Regression Model Diagnostics

Linear Regression Assumptions and Diagnostics in R: Essentials

 [kassambara](#) |  11/03/2018 |  2184 |  [Comment \(1\)](#) |  [Regression Model Diagnostics](#)

Linear regression (Chapter [@ref\(linear-regression\)](#)) makes several assumptions about the data at hand. This chapter describes **regression assumptions** and provides built-in plots for **regression diagnostics** in R programming language.

After performing a regression analysis, you should always check if the model works well for the data at hand.

A first step of this regression diagnostic is to inspect the significance of the regression beta coefficients, as well as, the R^2 that tells us how well the linear regression model fits to the data. This has been described in the Chapters [@ref\(linear-regression\)](#) and [@ref\(cross-validation\)](#).

In this current chapter, you will learn additional steps to evaluate how well the model fits the data.

For example, the linear regression model makes the assumption that the relationship between the predictors (x) and the outcome variable is linear. This might not be true. The relationship could be polynomial or logarithmic.

Additionally, the data might contain some influential observations, such as outliers (or extreme values), that can affect the result of the regression.

Therefore, you should closely diagnostic the regression model that you built in order to detect potential problems and to check whether the assumptions made by the linear regression model are met or not.

To do so, we generally examine the distribution of **residuals errors**, that can tell you more about your data.

In this chapter,

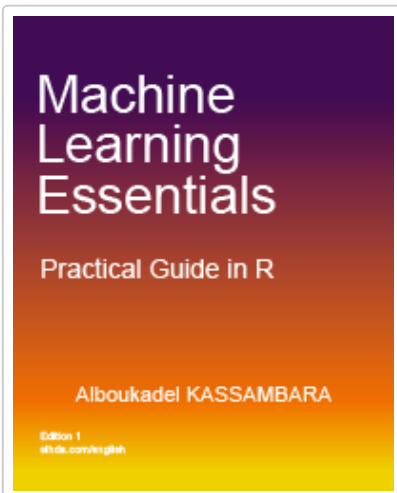
- we start by explaining **residuals errors** and **fitted values**.

- next, we present linear **regression assumptions**, as well as, potential problems you can face when performing regression analysis.
- finally, we describe some built-in **diagnostic plots** in R for testing the assumptions underlying linear regression model.

Contents:

- [Loading Required R packages](#)
- [Example of data](#)
- [Building a regression model](#)
- [Fitted values and residuals](#)
- [Regression assumptions](#)
- [Regression diagnostics {reg-diag}](#)
 - [Diagnostic plots](#)
- [Linearity of the data](#)
- [Homogeneity of variance](#)
- [Normality of residuals](#)
- [Outliers and high leverage points](#)
- [Influential values](#)
- [Discussion](#)
- [References](#)

The Book:



[Machine Learning Essentials: Practical Guide in R](#)

Loading Required R packages

- **tidyverse** for easy data manipulation and visualization
- **broom**: creates a tidy data frame from statistical test results

```
library(tidyverse)
library(broom)
theme_set(theme_classic())
```

Example of data

We'll use the data set `marketing` [datarium package], introduced in Chapter [@ref\(regression-analysis\)](#).

```
# Load the data
data("marketing", package = "datarium")
# Inspect the data
sample_n(marketing, 3)
```

```
##      youtube facebook newspaper sales
## 58    163.4      23.0      19.9  15.8
## 157   112.7      52.2      60.6  18.4
## 81     91.7      32.0      26.8  14.2
```

Building a regression model

We build a model to predict sales on the basis of advertising budget spent in youtube medias.

```
model <- lm(sales ~ youtube, data = marketing)
model
```

```
##
## Call:
## lm(formula = sales ~ youtube, data = marketing)
##
## Coefficients:
## (Intercept)      youtube
##      8.4391      0.0475
```

Our regression equation is: $y = 8.43 + 0.07 \cdot x$, that is $\text{sales} = 8.43 + 0.047 \cdot \text{youtube}$.

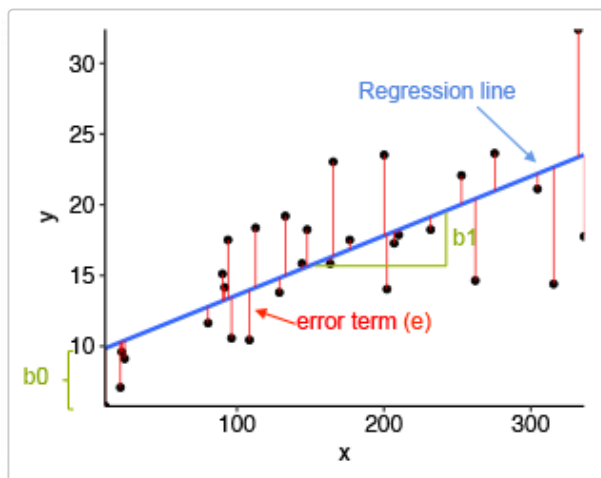
Before, describing regression assumptions and regression diagnostics, we start by explaining two key concepts in regression analysis: Fitted values and residuals errors. These are important for understanding the diagnostic plots presented hereafter.

Fitted values and residuals

The **fitted** (or **predicted**) values are the y-values that you would expect for the given x-values according to the built regression model (or visually, the best-fitting straight regression line).

In our example, for a given youtube advertising budget, the fitted (predicted) sales value would be, $\text{sales} = 8.44 + 0.0048 \cdot \text{youtube}$.

From the scatter plot below, it can be seen that not all the data points fall exactly on the estimated regression line. This means that, for a given youtube advertising budget, the observed (or measured) sale values can be different from the predicted sale values. The difference is called the **residual errors**, represented by a vertical red lines.



In R, you can easily augment your data to add fitted values and residuals by using the function `augment()` [broom package]. Let's call the output `model.diag.metrics` because it contains several metrics useful for regression diagnostics. We'll describe them later.

```
model.diag.metrics <- augment(model)
head(model.diag.metrics)
```

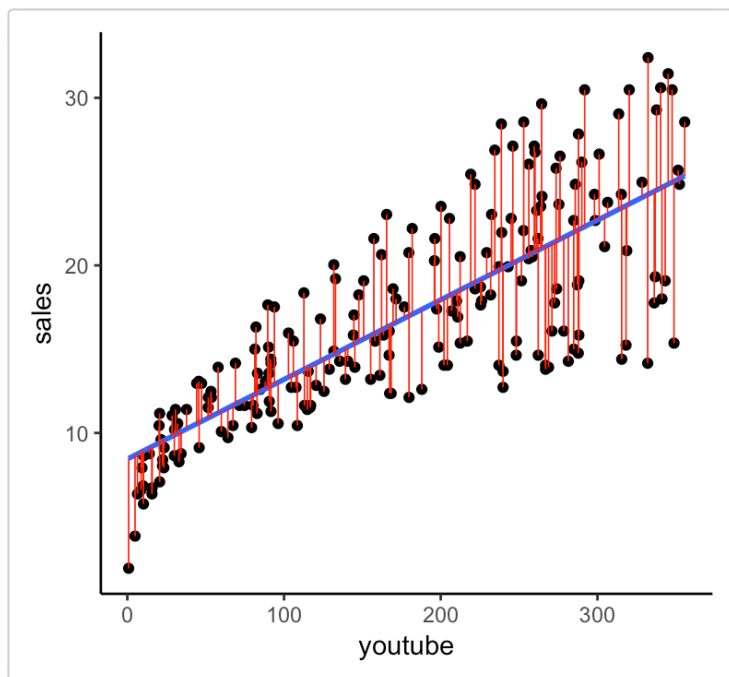
```
##  sales youtube .fitted .se.fit .resid .hat .sigma .cooksd .std.resid
## 1  26.52  276.1  21.56  0.385  4.955 0.00970  3.90 7.94e-03  1.2733
## 2  12.48   53.4  10.98  0.431  1.502 0.01217  3.92 9.20e-04  0.3866
## 3  11.16   20.6   9.42  0.502  1.740 0.01649  3.92 1.69e-03  0.4486
## 4  22.20  181.8  17.08  0.277  5.119 0.00501  3.90 4.34e-03  1.3123
## 5  15.48  217.0  18.75  0.297 -3.273 0.00578  3.91 2.05e-03 -0.8393
## 6   8.64   10.4   8.94  0.525 -0.295 0.01805  3.92 5.34e-05 -0.0762
```

Among the table columns, there are:

- `youtube`: the invested youtube advertising budget
- `sales`: the observed sale values
- `.fitted`: the fitted sale values
- `.resid`: the residual errors
- ...

The following R code plots the residuals error (in red color) between observed values and the fitted regression line. Each vertical red segments represents the residual error between an observed sale value and the corresponding predicted (i.e. fitted) value.

```
ggplot(model.diag.metrics, aes(youtube, sales)) +
  geom_point() +
  stat_smooth(method = lm, se = FALSE) +
  geom_segment(aes(xend = youtube, yend = .fitted), color = "red", size = 0.3)
```



In order to check regression assumptions, we'll examine the distribution of residuals.

Regression assumptions

Linear regression makes several assumptions about the data, such as :

1. **Linearity of the data.** The relationship between the predictor (x) and the outcome (y) is assumed to be linear.
2. **Normality of residuals.** The residual errors are assumed to be normally distributed.
3. **Homogeneity of residuals variance.** The residuals are assumed to have a constant variance (**homoscedasticity**)
4. **Independence of residuals error terms.**

You should check whether or not these assumptions hold true. Potential problems include:

1. **Non-linearity** of the outcome - predictor relationships
2. **Heteroscedasticity:** Non-constant variance of error terms.
3. **Presence of influential values** in the data that can be:
 - Outliers: extreme values in the outcome (y) variable
 - High-leverage points: extreme values in the predictors (x) variable

All these assumptions and potential problems can be checked by producing some diagnostic plots visualizing the residual errors.

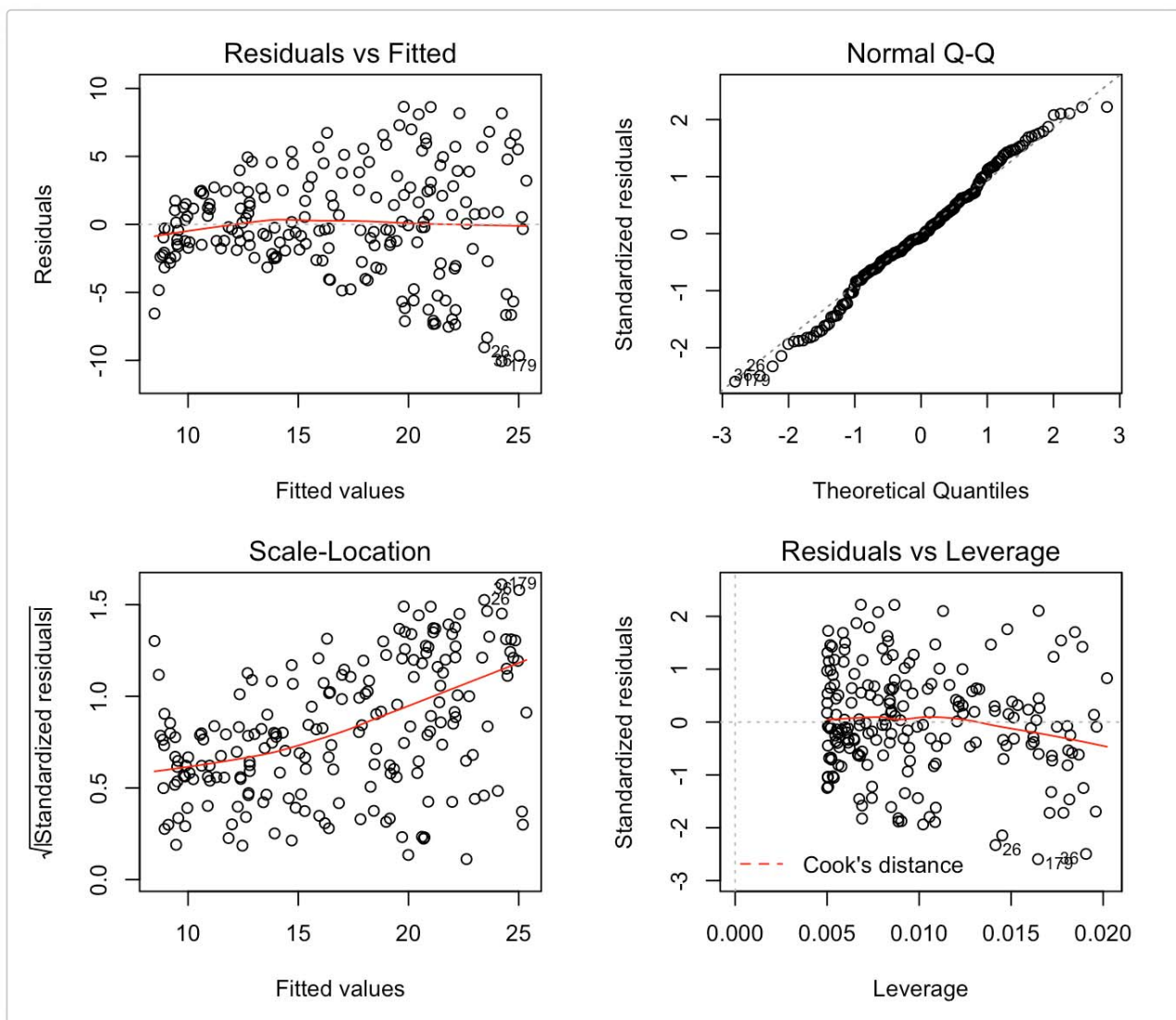
Regression diagnostics {reg-diag}

Diagnostic plots

Regression diagnostics plots can be created using the R base function `plot()` or the `autoplot()` function [ggfortify package], which creates a ggplot2-based graphics.

- Create the diagnostic plots with the R base function:

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



- Create the diagnostic plots using `ggfortify`:

```
library(ggfortify)
autoplot(model)
```

The diagnostic plots show residuals in four different ways:

1. **Residuals vs Fitted.** Used to check the linear relationship assumptions. A horizontal line, without distinct patterns is an indication for a linear relationship, what is good.
2. **Normal Q-Q.** Used to examine whether the residuals are normally distributed. It's good if residuals points follow the straight dashed line.
3. **Scale-Location** (or Spread-Location). Used to check the homogeneity of variance of the residuals (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity. This is not the case in our example, where we have a heteroscedasticity problem.

4. **Residuals vs Leverage.** Used to identify influential cases, that is extreme values that might influence the regression results when included or excluded from the analysis. This plot will be described further in the next sections.

The four plots show the top 3 most extreme data points labeled with with the row numbers of the data in the data set. They might be potentially problematic. You might want to take a close look at them individually to check if there is anything special for the subject or if it could be simply data entry errors. We'll discuss about this in the following sections.

The metrics used to create the above plots are available in the `model.diag.metrics` data, described in the previous section.

```
# Add observations indices and
# drop some columns (.se.fit, .sigma) for simplification
model.diag.metrics <- model.diag.metrics %>%
  mutate(index = 1:nrow(model.diag.metrics)) %>%
  select(index, everything(), -.se.fit, -.sigma)
# Inspect the data
head(model.diag.metrics, 4)
```

```
##   index sales youtube .fitted .resid   .hat .cooksd .std.resid
## 1     1  26.5   276.1  21.56  4.96 0.00970 0.00794    1.273
## 2     2  12.5    53.4  10.98  1.50 0.01217 0.00092    0.387
## 3     3  11.2    20.6   9.42  1.74 0.01649 0.00169    0.449
## 4     4  22.2   181.8  17.08  5.12 0.00501 0.00434    1.312
```

We'll use mainly the following columns:

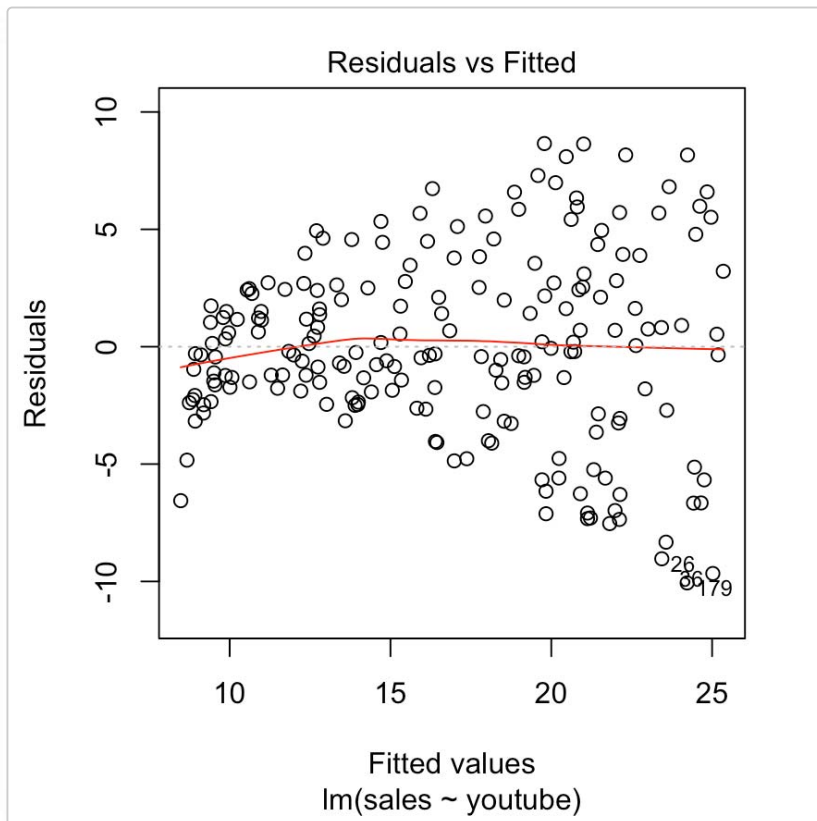
- **.fitted**: fitted values
- **.resid**: residual errors
- **.hat**: hat values, used to detect high-leverage points (or extreme values in the predictors x variables)
- **.std.resid**: standardized residuals, which is the residuals divided by their standard errors. Used to detect outliers (or extreme values in the outcome y variable)
- **.cooksd**: Cook's distance, used to detect influential values, which can be an outlier or a high leverage point

In the following section, we'll describe, in details, how to use these graphs and metrics to check the regression assumptions and to diagnostic potential problems in the model.

Linearity of the data

The linearity assumption can be checked by inspecting the **Residuals vs Fitted** plot (1st plot):

```
plot(model, 1)
```



Ideally, the residual plot will show no fitted pattern. That is, the red line should be approximately horizontal at zero. The presence of a pattern may indicate a problem with some aspect of the linear model.



In our example, there is no pattern in the residual plot. This suggests that we can assume linear relationship between the predictors and the outcome variables.

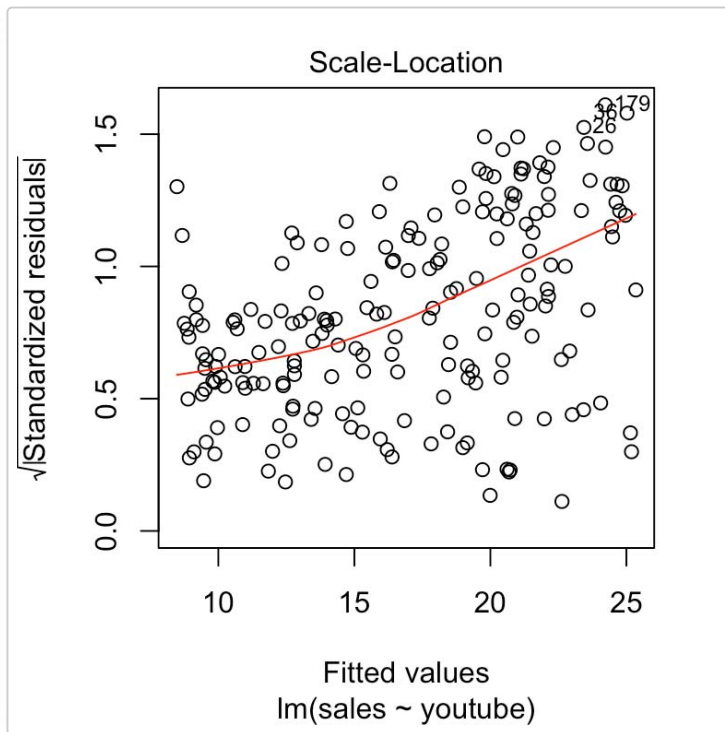


Note that, if the residual plot indicates a non-linear relationship in the data, then a simple approach is to use non-linear transformations of the predictors, such as $\log(x)$, \sqrt{x} and x^2 , in the regression model.

Homogeneity of variance

This assumption can be checked by examining the *scale-location plot*, also known as the *spread-location plot*.

```
plot(model, 3)
```

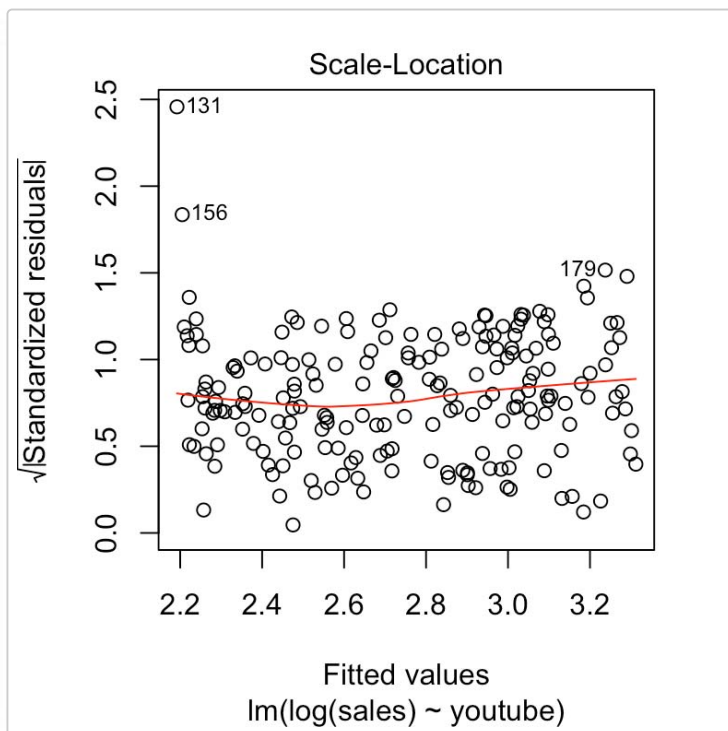



This plot shows if residuals are spread equally along the ranges of predictors. It's good if you see a horizontal line with equally spread points. In our example, this is not the case.

It can be seen that the variability (variances) of the residual points increases with the value of the fitted outcome variable, suggesting non-constant variances in the residuals errors (or *heteroscedasticity*).

A possible solution to reduce the heteroscedasticity problem is to use a log or square root transformation of the outcome variable (y).

```
model2 <- lm(log(sales) ~ youtube, data = marketing)
plot(model2, 3)
```

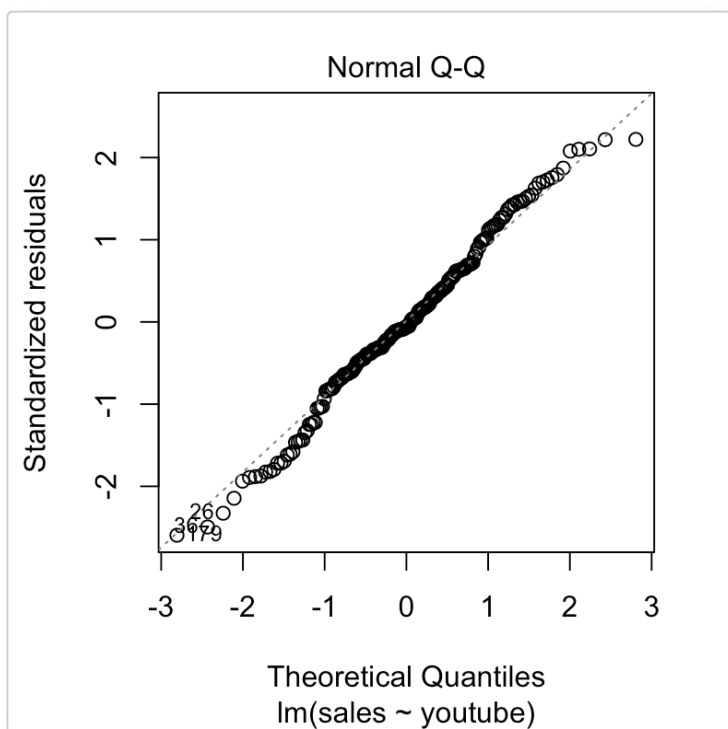


Normality of residuals

The QQ plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line.

In our example, all the points fall approximately along this reference line, so we can assume normality.

```
plot(model, 2)
```



Outliers and high leverage points

Outliers:

An outlier is a point that has an extreme outcome variable value. The presence of outliers may affect the interpretation of the model, because it increases the RSE.

Outliers can be identified by examining the *standardized residual* (or *studentized residual*), which is the residual divided by its estimated standard error. Standardized residuals can be interpreted as the number of standard errors away from the regression line.

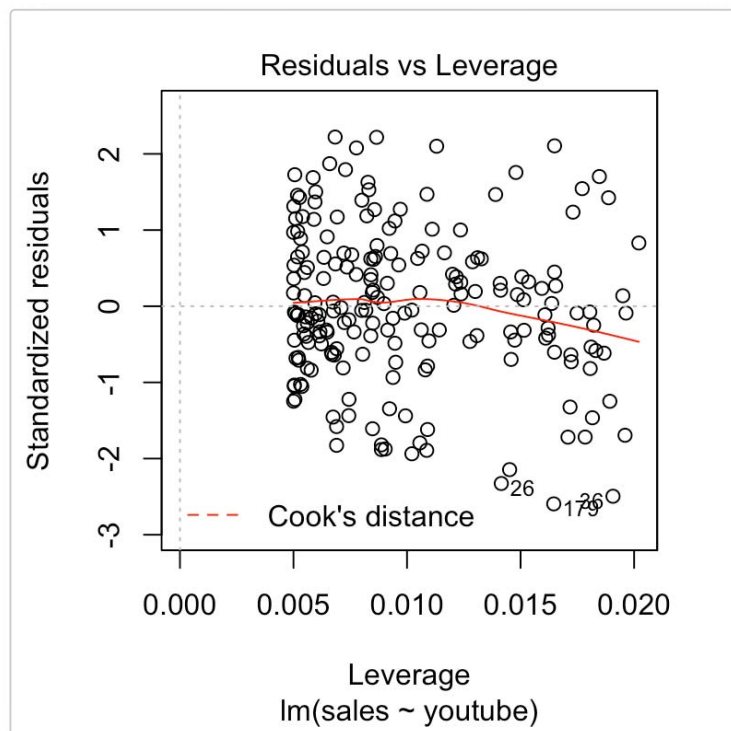
Observations whose standardized residuals are greater than 3 in absolute value are possible outliers (James et al. 2014).

High leverage points:

A data point has high leverage, if it has extreme predictor x values. This can be detected by examining the leverage statistic or the *hat-value*. A value of this statistic above $2(p + 1)/n$ indicates an observation with high leverage (P. Bruce and Bruce 2017); where, p is the number of predictors and n is the number of observations.

Outliers and high leverage points can be identified by inspecting the *Residuals vs Leverage* plot:

```
plot(model, 5)
```



✓ The plot above highlights the top 3 most extreme points (#26, #36 and #179), with a standardized residuals below -2. However, there is no outliers that exceed 3 standard deviations, what is good.

Additionally, there is no high leverage point in the data. That is, all data points, have a leverage statistic below $2(p + 1)/n = 4/200 = 0.02$.

Influential values

An influential value is a value, which inclusion or exclusion can alter the results of the regression analysis. Such a value is associated with a large residual.

Not all outliers (or extreme data points) are influential in linear regression analysis.

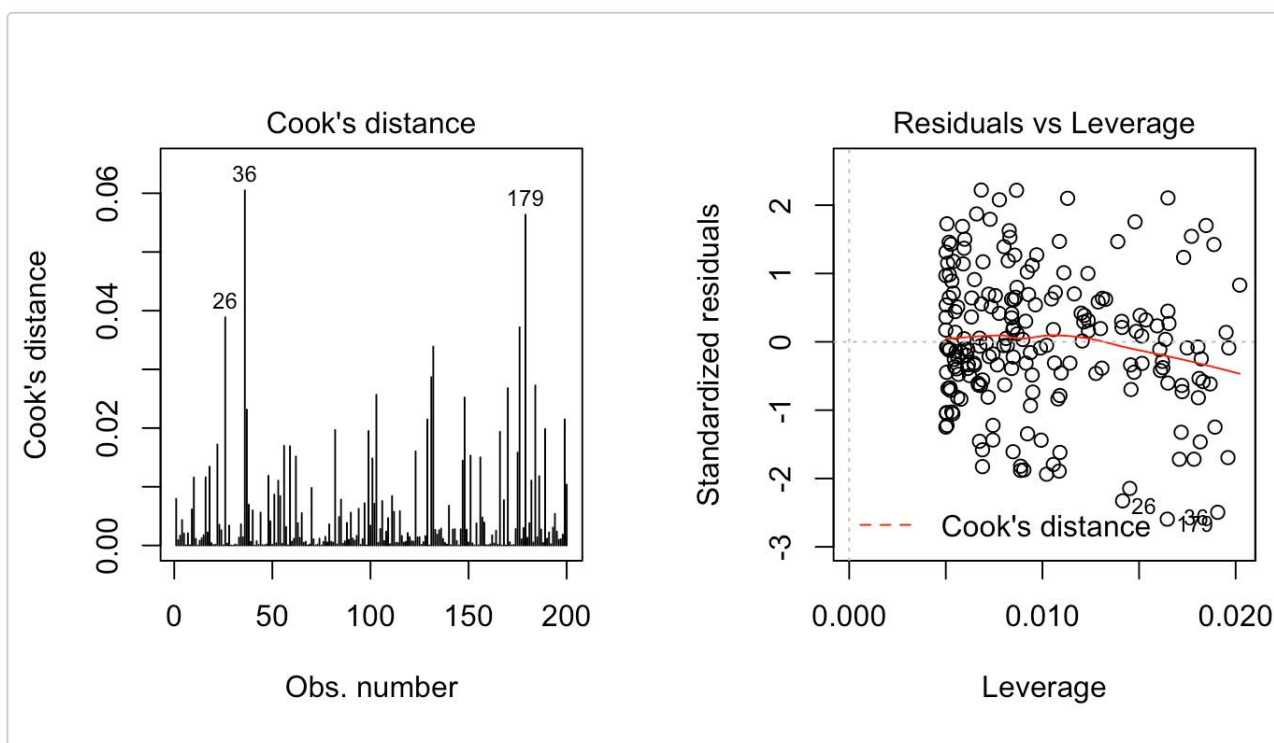
Statisticians have developed a metric called *Cook's distance* to determine the influence of a value. This metric defines influence as a combination of leverage and residual size.

A rule of thumb is that an observation has high influence if Cook's distance exceeds $4/(n - p - 1)$ (P. Bruce and Bruce 2017), where n is the number of observations and p the number of predictor variables.

The *Residuals vs Leverage* plot can help us to find influential observations if any. On this plot, outlying values are generally located at the upper right corner or at the lower right corner. Those spots are the places where data points can be influential against a regression line.

The following plots illustrate the Cook's distance and the leverage of our model:

```
# Cook's distance
plot(model, 4)
# Residuals vs Leverage
plot(model, 5)
```



By default, the top 3 most extreme values are labelled on the Cook's distance plot. If you want to label the top 5 extreme values, specify the option `id.n` as follow:

```
plot(model, 4, id.n = 5)
```

If you want to look at these top 3 observations with the highest Cook's distance in case you want to assess them further, type this R code:

```
model.diag.metrics %>%  
  top_n(3, wt = .cooks)
```

```
##   index sales youtube .fitted .resid   .hat .cooks .std.resid  
## 1    26  14.4    315   23.4  -9.04 0.0142 0.0389    -2.33  
## 2    36  15.4    349   25.0  -9.66 0.0191 0.0605    -2.49  
## 3   179  14.2    332   24.2 -10.06 0.0165 0.0563    -2.59
```



When data points have high Cook's distance scores and are to the upper or lower right of the leverage plot, they have leverage meaning they are influential to the regression results. The regression results will be altered if we exclude those cases.



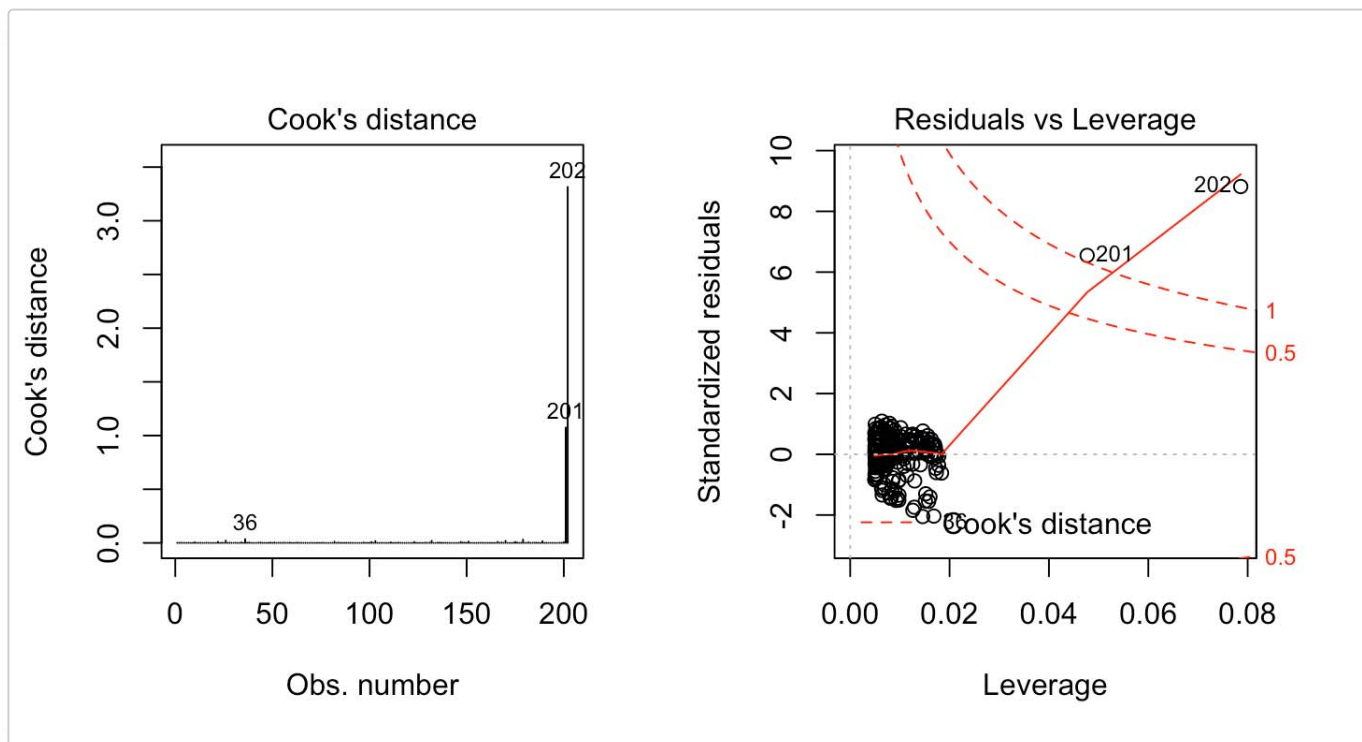
In our example, the data don't present any influential points. Cook's distance lines (a red dashed line) are not shown on the Residuals vs Leverage plot because all points are well inside of the Cook's distance lines.

Let's show now another example, where the data contain two extremes values with potential influence on the regression results:

```
df2 <- data.frame(  
  x = c(marketing$youtube, 500, 600),  
  y = c(marketing$sales, 80, 100)  
)  
model2 <- lm(y ~ x, df2)
```

Create the *Residuals vs Leverage* plot of the two models:

```
# Cook's distance  
plot(model2, 4)  
# Residuals vs Leverage  
plot(model2, 5)
```



On the Residuals vs Leverage plot, look for a data point outside of a dashed line, Cook's distance. When the points are outside of the Cook's distance, this means that they have high Cook's distance scores. In this case, the values are influential to the regression results. The regression results will be altered if we exclude those cases.

In the above example 2, two data points are far beyond the Cook's distance lines. The other residuals appear clustered on the left. The plot identified the influential observation as #201 and #202. If you exclude these points from the analysis, the slope coefficient changes from 0.06 to 0.04 and R^2 from 0.5 to 0.6. Pretty big impact!

Discussion

This chapter describes linear regression assumptions and shows how to diagnostic potential problems in the model.

The diagnostic is essentially performed by visualizing the residuals. Having patterns in residuals is not a stop signal. Your current regression model might not be the best way to understand your data.

Potential problems might be:

- A non-linear relationships between the outcome and the predictor variables. When facing to this problem, one solution is to include a quadratic term, such as polynomial terms or log transformation. See Chapter @ref(polynomial-and-spline-regression).
- Existence of important variables that you left out from your model. Other variables you didn't include (e.g., age or gender) may play an important role in your model and data. See Chapter @ref(confounding-variables).
- Presence of outliers. If you believe that an outlier has occurred due to an error in data collection and entry, then one solution is to simply remove the concerned observation.

References

Bruce, Peter, and Andrew Bruce. 2017. *Practical Statistics for Data Scientists*. O'Reilly Media.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.



Enjoyed this article? Give us 5 stars ★★★★★ (just above this text block)! Reader needs to be STHDA member for voting. I'd be very grateful if you'd help it spread by emailing it to a friend, or sharing it on Twitter, Facebook or Linked In.

Show me some love with the like buttons below... Thank you and please don't forget to share and comment below!!

Ads by Google

Add Testing

Plot Sale

Create a Graph

Share 34

Like 34

Tweet

Share

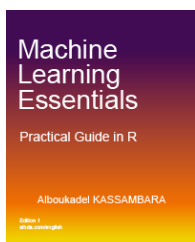


Save

Share

1

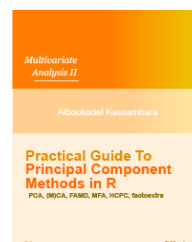
Recommended for You!



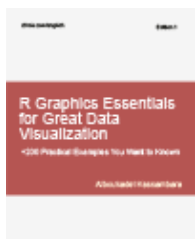
Machine Learning Essentials:
Practical Guide in R



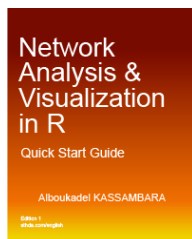
Practical Guide to Cluster
Analysis in R



Practical Guide to Principal
Component Methods in R



R Graphics Essentials for Great
Data Visualization



Network Analysis and
Visualization in R



More books on R and data
science

The fields marked with a * are required !

Add a comment

Name

Visitor

Message



Preview

* Code de vérification

What is the result of 5 + ten?

Submit

Reset



tomer mann 05/29/2018 at 17h43

Member

brilliant as always!

#505

Sign in

Login

Password

Auto connect

[!\[\]\(cf531ed27e91483460120fcc057b3901_img.jpg\) Register](#)[!\[\]\(4b7a79268f6ba26c1471d4232fffa85a_img.jpg\) Forgotten password](#)


Welcome!

Want to Learn More on R Programming and Data Science?

Follow us [by Email](#)

by [FeedBurner](#)

on Social Networks

 Ads by Google

[Data Analysis R](#)[Histogram Graph](#)[R Programming](#)[Add Testing](#)

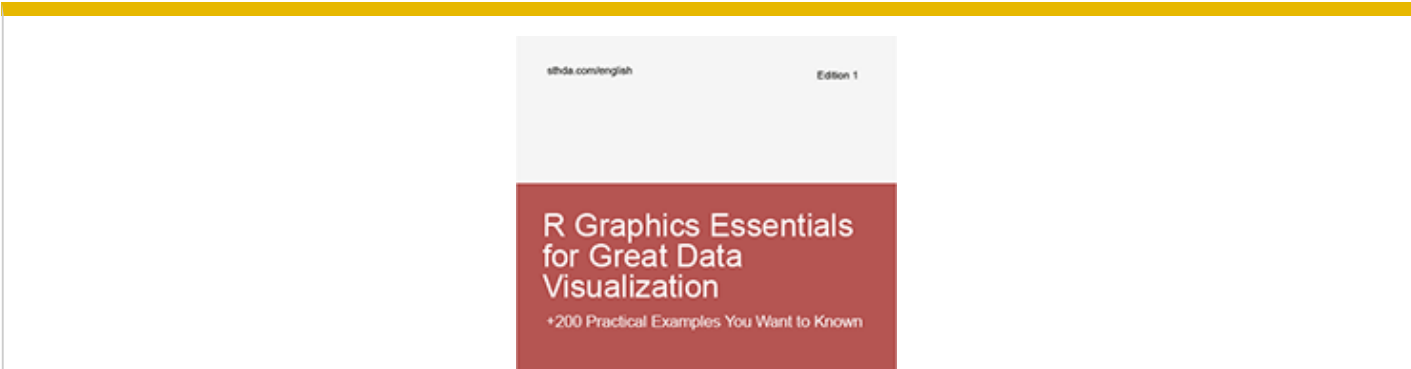
 **factoextra**

 **survminer**

-  `ggpubr`
-  `ggcorrplot`
-  `fastqcr`



Our Books





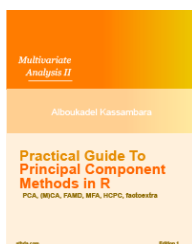
Alboukadel Kassambara

R Graphics Essentials for Great Data Visualization: 200 Practical Examples You Want to Know for Data Science

★ NEW!!

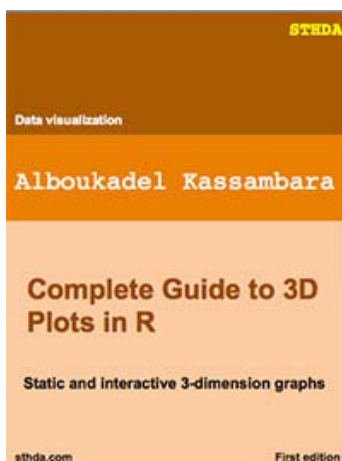


Practical Guide to Cluster Analysis in R



Practical Guide to Principal Component Methods in R

3D Plots in R



Guest Book

I'm psychologist, from Chile. This website is WONDERFUL!! Comprehensive, clear, simple, great!!!!

Thank you, thank you!!!!

Pablo

By *Visitor*

[Guest Book](#)



Newsletter



Boosted by PHPBoost

Recommended for you



Clustering Distance
Measures Essentials...

www.sthda.com



Impressive package for
3D and 4D graph - R s...

www.sthda.com



Penalized Regression
Essentials: Ridge, Lass...

www.sthda.com

AddThis