



Statistical tools for high-throughput data analysis






Licence:

[Home](#)[Basics](#)[Data](#)[Visualize](#)[Analyze](#)[Products](#)[Contribute](#)[Support](#)[About](#)[Home](#) / [Articles](#) / [Machine Learning](#) / [Regression Analysis](#) / [Linear Regression Essentials in R](#)

We know what you did  
**LAST MONSOON**

## Articles - Regression Analysis

### Linear Regression Essentials in R

 [kassambara](#) |  11/03/2018 |  2199 |  [Comments \(3\)](#) |  [Regression Analysis](#)

**Linear regression** (or **linear model**) is used to predict a quantitative outcome variable (y) on the basis of one or multiple predictor variables (x) (James et al. 2014, P. Bruce and Bruce (2017)).

The goal is to build a mathematical formula that defines y as a function of the x variable. Once, we built a statistically significant model, it's possible to use it for predicting future outcome on the basis of new x values.

When you build a regression model, you need to assess the performance of the predictive model. In other words, you need to evaluate how well the model is in predicting the outcome of a new test data that have not been used to build the model.

Two important metrics are commonly used to assess the performance of the predictive regression model:

- **Root Mean Squared Error**, which measures the model prediction error. It corresponds to the average difference between the observed known values of the outcome and the predicted value by the model. RMSE is computed as `RMSE = mean((observeds - predicted)^2) %>% sqrt()`. The lower the RMSE, the better the model.
- **R-square**, representing the squared correlation between the observed known outcome values and the predicted values by the model. The higher the R2, the better the model.

A simple workflow to build a predictive regression model is as follow:

1. Randomly split your data into training set (80%) and test set (20%)
2. Build the regression model using the training set
3. Make predictions using the test set and compute the model accuracy metrics

In this chapter, you will learn:

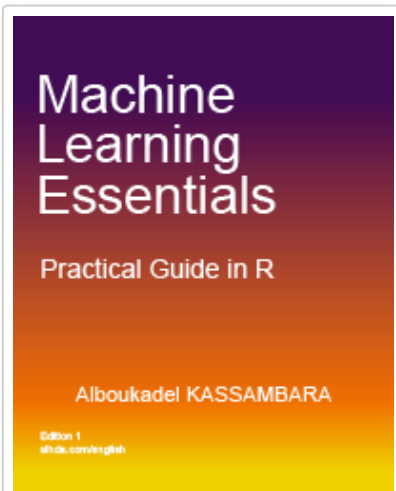
- the basics and the formula of linear regression,

- how to compute simple and multiple regression models in R,
- how to make predictions of the outcome of new data,
- how to assess the performance of the model

#### Contents:

- [Formula](#)
- [Loading Required R packages](#)
- [Preparing the data](#)
- [Computing linear regression](#)
  - [Quick start R code](#)
  - [Simple linear regression](#)
  - [Multiple linear regression](#)
- [Interpretation](#)
  - [Model summary](#)
  - [Coefficients significance](#)
  - [Model accuracy](#)
- [Making predictions](#)
- [Discussion](#)
- [References](#)

#### The Book:



Machine Learning Essentials:  
Practical Guide in R

## Formula

The mathematical formula of the linear regression can be written as follow:

$$y = b_0 + b_1 \cdot x + e$$

We read this as “y is modeled as beta1 ( $b_1$ ) times x, plus a constant beta0 ( $b_0$ ), plus an error term e.”

When you have multiple predictor variables, the equation can be written as  $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n$ , where:

- $b_0$  is the intercept,

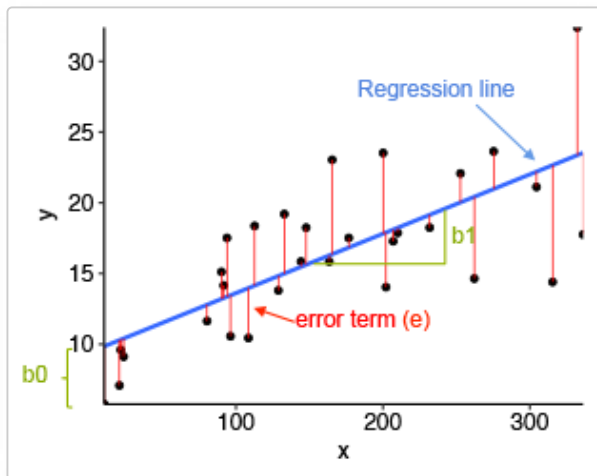
- $b_1, b_2, \dots, b_n$  are the regression weights or coefficients associated with the predictors  $x_1, x_2, \dots, x_n$ .
- $e$  is the *error term* (also known as the *residual errors*), the part of  $y$  that can be explained by the regression model



Note that,  $b_0, b_1, b_2, \dots$  and  $b_n$  are known as the regression beta coefficients or parameters.

The figure below illustrates a simple linear regression model, where:

- the best-fit regression line is in blue
- the intercept ( $b_0$ ) and the slope ( $b_1$ ) are shown in green
- the error terms ( $e$ ) are represented by vertical red lines



From the scatter plot above, it can be seen that not all the data points fall exactly on the fitted regression line. Some of the points are above the blue curve and some are below it; overall, the residual errors ( $e$ ) have approximately mean zero.

The sum of the squares of the residual errors are called the **Residual Sum of Squares** or **RSS**.

The average variation of points around the fitted regression line is called the **Residual Standard Error (RSE)**. This is one of the metrics used to evaluate the overall quality of the fitted regression model. The lower the RSE, the better it is.

Since the mean error term is zero, the outcome variable  $y$  can be approximately estimated as follow:

$$y \sim b_0 + b_1 \cdot x$$

Mathematically, the beta coefficients ( $b_0$  and  $b_1$ ) are determined so that the RSS is as minimal as possible. This method of determining the beta coefficients is technically called **least squares** regression or **ordinary least squares** (OLS) regression.

Once, the beta coefficients are calculated, a t-test is performed to check whether or not these coefficients are significantly different from zero. A non-zero beta coefficients means that there is a significant relationship between the predictors ( $x$ ) and the outcome variable ( $y$ ).

## Loading Required R packages

- **tidyverse** for easy data manipulation and visualization
- **caret** for easy machine learning workflow

```
library(tidyverse)
library(caret)
```

```
theme_set(theme_bw())
```

## Preparing the data

We'll use the `marketing` data set, introduced in the Chapter [@ref\(regression-analysis\)](#), for predicting sales units on the basis of the amount of money spent in the three advertising medias (youtube, facebook and newspaper)

We'll randomly split the data into training set (80% for building a predictive model) and test set (20% for evaluating the model). Make sure to set seed for reproducibility.

```
# 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
```

```
# Split the data into training and test set
set.seed(123)
training.samples <- marketing$sales %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data <- marketing[training.samples, ]
test.data <- marketing[-training.samples, ]
```

## Computing linear regression

The R function `lm()` is used to compute linear regression model.

### Quick start R code

```
# Build the model
model <- lm(sales ~., data = train.data)
# Summarize the model
summary(model)
# Make predictions
predictions <- model %>% predict(test.data)
# Model performance
# (a) Prediction error, RMSE
RMSE(predictions, test.data$sales)
# (b) R-square
R2(predictions, test.data$sales)
```

## Simple linear regression

The **simple linear regression** is used to predict a continuous outcome variable (y) based on one single predictor variable (x).

In the following example, we'll build a simple linear model to predict sales units based on the advertising budget spent on youtube. The regression equation can be written as  $\text{sales} = b_0 + b_1 \cdot \text{youtube}$ .

The R function `lm()` can be used to determine the beta coefficients of the linear model, as follow:

```
model <- lm(sales ~ youtube, data = train.data)
summary(model)$coef
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3839    0.62442   13.4 5.22e-28
## youtube       0.0468    0.00301   15.6 7.84e-34
```

The output above shows the estimate of the regression beta coefficients (column **Estimate**) and their significance levels (column **Pr(>|t|)**). The intercept ( $b_0$ ) is 8.38 and the coefficient of youtube variable is 0.046.

The estimated regression equation can be written as follow:  $\text{sales} = 8.38 + 0.046 \cdot \text{youtube}$ . Using this formula, for each new youtube advertising budget, you can predict the number of sale units.

For example:

- For a youtube advertising budget equal zero, we can expect a sale of 8.38 units.
- For a youtube advertising budget equal 1000, we can expect a sale of  $8.38 + 0.046 \cdot 1000 = 55$  units.

Predictions can be easily made using the R function `predict()`. In the following example, we predict sales units for two youtube advertising budget: 0 and 1000.

```
newdata <- data.frame(youtube = c(0, 1000))
model %>% predict(newdata)
```

```
##      1      2
## 8.38 55.19
```

## Multiple linear regression

**Multiple linear regression** is an extension of simple linear regression for predicting an outcome variable (y) on the basis of multiple distinct predictor variables (x).

For example, with three predictor variables (x), the prediction of y is expressed by the following equation:  $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3$

The regression beta coefficients measure the association between each predictor variable and the outcome. " $b_j$ " can be interpreted as the average effect on y of a one unit increase in " $x_j$ ", holding all other predictors fixed.

In this section, we'll build a multiple regression model to predict sales based on the budget invested in three advertising medias: youtube, facebook and newspaper. The formula is as follow:  $\text{sales} = b_0 + b_1 \cdot \text{youtube} + b_2 \cdot \text{facebook} + b_3 \cdot \text{newspaper}$

You can compute the multiple regression model coefficients in R as follow:

```
model <- lm(sales ~ youtube + facebook + newspaper,
            data = train.data)
summary(model)$coef
```

Note that, if you have many predictor variables in your data, you can simply include all the available variables in the model using `~.`:

```
model <- lm(sales ~., data = train.data)
summary(model)$coef
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.39188    0.44062   7.698 1.41e-12
## youtube      0.04557    0.00159  28.630 2.03e-64
## facebook     0.18694    0.00989  18.905 2.07e-42
## newspaper    0.00179    0.00677   0.264 7.92e-01
```

From the output above, the coefficients table shows the beta coefficient estimates and their significance levels. Columns are:

- **Estimate**: the intercept (b0) and the beta coefficient estimates associated to each predictor variable
- **Std.Error**: the standard error of the coefficient estimates. This represents the accuracy of the coefficients. The larger the standard error, the less confident we are about the estimate.
- **t value**: the t-statistic, which is the coefficient estimate (column 2) divided by the standard error of the estimate (column 3)
- **Pr(>|t|)**: The p-value corresponding to the t-statistic. The smaller the p-value, the more significant the estimate is.

As previously described, you can easily make predictions using the R function `predict()`:

```
# New advertising budgets
newdata <- data.frame(
  youtube = 2000, facebook = 1000,
  newspaper = 1000
)
# Predict sales values
model %>% predict(newdata)
```

```
##      1
## 283
```

## Interpretation

Before using a model for predictions, you need to assess the statistical significance of the model. This can be easily checked by displaying the statistical summary of the model.

## Model summary

Display the statistical summary of the model as follow:

```
summary(model)
```

```
##
## Call:
## lm(formula = sales ~ ., data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.412  -1.110   0.348   1.422   3.499
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.39188    0.44062   7.70  1.4e-12 ***
## youtube      0.04557    0.00159  28.63 < 2e-16 ***
## facebook     0.18694    0.00989  18.90 < 2e-16 ***
## newspaper    0.00179    0.00677   0.26   0.79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.12 on 158 degrees of freedom
## Multiple R-squared:  0.89,    Adjusted R-squared:  0.888
## F-statistic:  427 on 3 and 158 DF,  p-value: <2e-16
```

The summary outputs shows 6 components, including:

- **Call.** Shows the function call used to compute the regression model.
- **Residuals.** Provide a quick view of the distribution of the residuals, which by definition have a mean zero. Therefore, the median should not be far from zero, and the minimum and maximum should be roughly equal in absolute value.
- **Coefficients.** Shows the regression beta coefficients and their statistical significance. Predictor variables, that are significantly associated to the outcome variable, are marked by stars.
- **Residual standard error (RSE), R-squared (R2) and the F-statistic** are metrics that are used to check how well the model fits to our data.

The first step in interpreting the multiple regression analysis is to examine the F-statistic and the associated p-value, at the bottom of model summary.



In our example, it can be seen that p-value of the F-statistic is  $< 2.2e-16$ , which is highly significant. This means that, at least, one of the predictor variables is significantly related to the outcome variable.

## Coefficients significance

To see which predictor variables are significant, you can examine the coefficients table, which shows the estimate of regression beta coefficients and the associated t-statistic p-values.

```
summary(model)$coef
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.39188    0.44062   7.698 1.41e-12
## youtube      0.04557    0.00159  28.630 2.03e-64
## facebook     0.18694    0.00989  18.905 2.07e-42
## newspaper    0.00179    0.00677   0.264 7.92e-01
```

For a given the predictor, the t-statistic evaluates whether or not there is significant association between the predictor and the outcome variable, that is whether the beta coefficient of the predictor is significantly different from zero.



It can be seen that, changing in youtube and facebook advertising budget are significantly associated to changes in sales while changes in newspaper budget is not significantly associated with sales.

For a given predictor variable, the coefficient (b) can be interpreted as the average effect on y of a one unit increase in predictor, holding all other predictors fixed.

For example, for a fixed amount of youtube and newspaper advertising budget, spending an additional 1 000 dollars on facebook advertising leads to an increase in sales by approximately  $0.1885 \times 1000 = 189$  sale units, on average.

The youtube coefficient suggests that for every 1 000 dollars increase in youtube advertising budget, holding all other predictors constant, we can expect an increase of  $0.045 \times 1000 = 45$  sales units, on average.

We found that newspaper is not significant in the multiple regression model. This means that, for a fixed amount of youtube and newspaper advertising budget, changes in the newspaper advertising budget will not significantly affect sales units.

As the newspaper variable is not significant, it is possible to remove it from the model:

```
model <- lm(sales ~ youtube + facebook, data = train.data)
summary(model)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook, data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.481  -1.104   0.349   1.423   3.486
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.43446    0.40877   8.4 2.3e-14 ***
## youtube      0.04558    0.00159  28.7 < 2e-16 ***
## facebook     0.18788    0.00920  20.4 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.11 on 159 degrees of freedom
## Multiple R-squared:  0.89,    Adjusted R-squared:  0.889
## F-statistic: 644 on 2 and 159 DF,  p-value: <2e-16
```



Finally, our model equation can be written as follow:  $\text{sales} = 3.43 + 0.045\text{youtube} + 0.187\text{facebook}$ .



## Model accuracy

Once you identified that, at least, one predictor variable is significantly associated to the outcome, you should continue the diagnostic by checking how well the model fits the data. This process is also referred to as the *goodness-of-fit*

The overall quality of the linear regression fit can be assessed using the following three quantities, displayed in the model summary:

1. Residual Standard Error (RSE),
2. R-squared (R2) and adjusted R2,
3. F-statistic, which has been already described in the previous section

```
##      rse r.squared f.statistic  p.value
## 1 2.11      0.89      644 5.64e-77
```

### 1. Residual standard error (RSE).

The RSE (or model *sigma*), corresponding to the prediction error, represents roughly the average difference between the observed outcome values and the predicted values by the model. The lower the RSE the best the model fits to our data.

Dividing the RSE by the average value of the outcome variable will give you the prediction error rate, which should be as small as possible.



In our example, using only youtube and facebook predictor variables, the RSE = 2.11, meaning that the observed sales values deviate from the predicted values by approximately 2.11 units in average.

This corresponds to an error rate of  $2.11/\text{mean}(\text{train.data}\$sales) = 2.11/16.77 = 13\%$ , which is low.

### 2. R-squared and Adjusted R-squared:

The R-squared (R2) ranges from 0 to 1 and represents the proportion of variation in the outcome variable that can be explained by the model predictor variables.

For a simple linear regression, R2 is the square of the Pearson correlation coefficient between the outcome and the predictor variables. In multiple linear regression, the R2 represents the correlation coefficient between the observed outcome values and the predicted values.

The R2 measures, how well the model fits the data. The higher the R2, the better the model. However, a problem with the R2, is that, it will always increase when more variables are added to the model, even if those variables are only weakly associated with the outcome (James et al. 2014). A solution is to adjust the R2 by taking into account the number of predictor variables.

The adjustment in the "Adjusted R Square" value in the summary output is a correction for the number of x variables included in the predictive model.

So, you should mainly consider the adjusted R-squared, which is a penalized R2 for a higher number of predictors.

- An (adjusted) R2 that is close to 1 indicates that a large proportion of the variability in the outcome has been explained by the regression model.
- A number near 0 indicates that the regression model did not explain much of the variability in the outcome.



In our example, the adjusted R2 is 0.88, which is good.

### 3. F-Statistic:

Recall that, the F-statistic gives the overall significance of the model. It assess whether at least one predictor variable has a non-zero coefficient.

In a simple linear regression, this test is not really interesting since it just duplicates the information given by the t-test, available in the coefficient table.

The F-statistic becomes more important once we start using multiple predictors as in multiple linear regression.



A large F-statistic will correspond to a statistically significant p-value ( $p < 0.05$ ). In our example, the F-statistic equal 644 producing a p-value of  $1.46e-42$ , which is highly significant.

## Making predictions

We'll make predictions using the test data in order to evaluate the performance of our regression model.

The procedure is as follow:

1. Predict the sales values based on new advertising budgets in the test data
2. Assess the model performance by computing:
  - The prediction error RMSE (Root Mean Squared Error), representing the average difference between the observed known outcome values in the test data and the predicted outcome values by the model. The lower the RMSE, the better the model.
  - The R-square ( $R^2$ ), representing the correlation between the observed outcome values and the predicted outcome values. The higher the  $R^2$ , the better the model.

```
# Make predictions
predictions <- model %>% predict(test.data)
# Model performance
# (a) Compute the prediction error, RMSE
RMSE(predictions, test.data$sales)
```

```
## [1] 1.58
```

```
# (b) Compute R-square
R2(predictions, test.data$sales)
```

```
## [1] 0.938
```



From the output above, the  $R^2$  is 0.93, meaning that the observed and the predicted outcome values are highly correlated, which is very good.

The prediction error RMSE is 1.58, representing an error rate of  $1.58/\text{mean}(\text{test.data}\$sales) = 1.58/17 = 9.2\%$ , which is good.

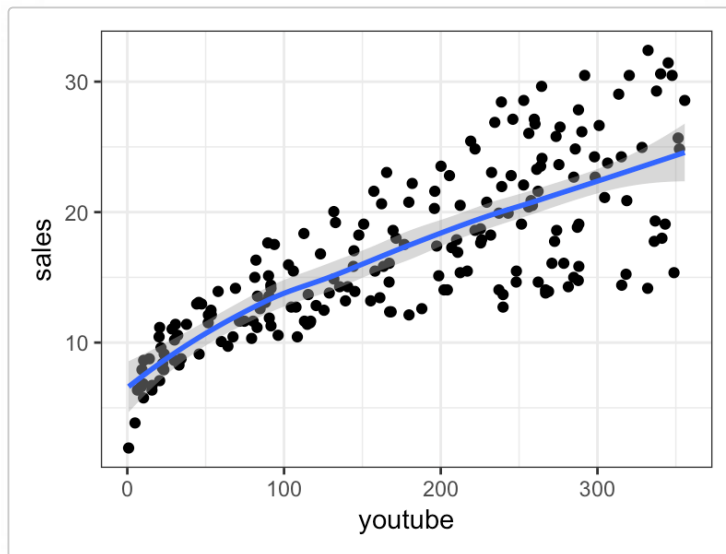
## Discussion

This chapter describes the basics of linear regression and provides practical examples in R for computing simple and multiple linear regression models. We also described how to assess the performance of the model for predictions.

Note that, linear regression assumes a linear relationship between the outcome and the predictor variables. This can be easily checked by creating a scatter plot of the outcome variable vs the predictor variable.

For example, the following R code displays sales units versus youtube advertising budget. We'll also add a smoothed line:

```
ggplot(marketing, aes(x = youtube, y = sales)) +  
  geom_point() +  
  stat_smooth()
```



The graph above shows a linearly increasing relationship between the **sales** and the **youtube** variables, which is a good thing.

In addition to the linearity assumptions, the linear regression method makes many other assumptions about your data (see Chapter @ref(regression-assumptions-and-diagnostics)). You should make sure that these assumptions hold true for your data.

Potential problems, include: a) the presence of influential observations in the data (Chapter @ref(regression-assumptions-and-diagnostics)), non-linearity between the outcome and some predictor variables (@ref(polynomial-and-spline-regression)) and the presence of strong correlation between predictor variables (Chapter @ref(multicollinearity)).

## 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.

★★★★★ 3 Notes



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

Sthda R LM

Create a Graph

Basic Math

Share 42

Like 42

Tweet

Share



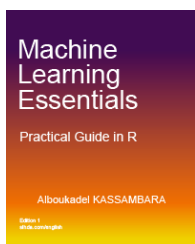
Save

Share

2



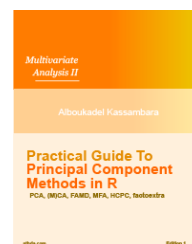
## Recommended for You!



Machine Learning Essentials:  
Practical Guide in R



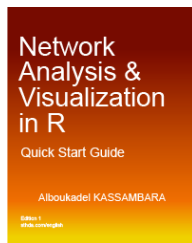
Practical Guide to Cluster  
Analysis in R



Practical Guide to Principal  
Component Methods in R



R Graphics Essentials for Great  
Data Visualization



Network Analysis &  
Visualization in R



More books on R and data  
science

The fields marked with a \* are required !

## Add a comment

Name

Visitor

Message

😊
**B**
*I*
U
~~S~~
💧
**T†**
**A**
☰
☷
□
“”
👁
⚠
🌐
🖼
📷
⌵

Preview

\* Code de vérification

How many vowels are in the word sthda?

Submit

Reset



John Visitor 03/30/2018 at 08h59

Visitor



I like. thanks

#416

**tomer mann** 05/12/2018 at 12h01

Member

a highly clear,easy to read and methodical tutorial. thank you!!!

#461

**kassambara** 05/19/2018 at 15h00

Administrator

Thank you for your positive feedback. Highly appreciated!!

#486

## Sign in

### Login

### Password

### Auto connect


 Register  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](#)

[Regression Sthda](#)

[Data Set Example](#)

[Sthda R LM](#)

 [factoextra](#)

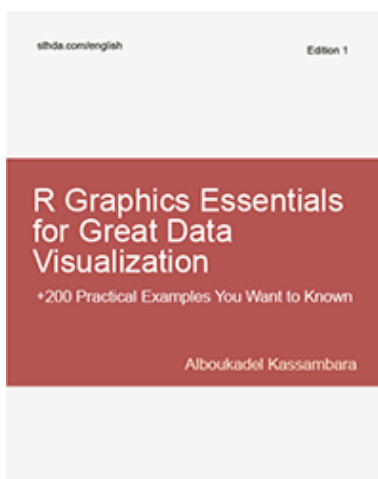
 [survminer](#)

 [ggpubr](#)

 [ggcorrplot](#)

 [fastqcr](#)

## Our Books



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

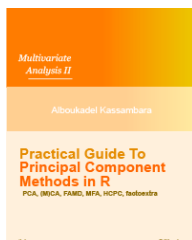
★ **NEW!!**

## 3D Plots in R

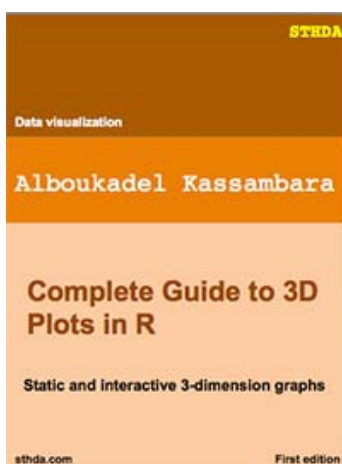




### Practical Guide to Cluster Analysis in R



### Practical Guide to Principal Component Methods in R



## Guest Book

This website is excellent, it's extremely useful. It boasts very good techniques, elegant code, and is well-written and organised. It's one of the best of its kind.

By *Zahra H*

[Guest Book](#)

 R-Bloggers

Newsletter

Email



Boosted by PHPBoost

Recommended for you





Correlation matrix :  
Formatting and visuali...

[www.sthda.com](http://www.sthda.com)



ggplot2.stripchart : Easy  
one dimensional scatt...

[www.sthda.com](http://www.sthda.com)



Guide to Create  
Beautiful Graphics in...

[www.sthda.com](http://www.sthda.com)

AddThis