TD 6

*Outils pour l'exploration de données* - Regression analysis.

# 1 Introduction

In statistics, regression analysis is a statistical technique for estimating the relationships among variables. More specifically, regression analysis helps one to understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables, that is, the average value of the dependent variable when the independent variables are fixed.

Regression models involve the following variables:

- The unknown parameters, denoted as $\boldsymbol{\beta}$, which may represent a scalar or a vector.
- The independent variables, X.
- The dependent variable, Y.

In several fields of application, different terminologies are used in place of dependent and independent variables. For example, target variable meaning dependent variable and predictors or explanatory variables meaning the independent variables.

A regression model relates Y to a function of X and $\boldsymbol{\beta}$.

$$Y \approx f(\mathbf{X}, \boldsymbol{\beta})$$

Assuming that the vector of unknown parameters $\boldsymbol{\beta}$ is of length $k$, in order to perform a regression analysis the user must provide information about the dependent variable Y. The most common situation is where $N > k$ data points are observed. In this case, there is enough information in the data to estimate a unique value for $\boldsymbol{\beta}$ that best fits the data in some sense. Also, in this context, the regression analysis provides the tools for finding a solution for unknown parameters $\boldsymbol{\beta}$ that will, for example, minimize the distance between the measured and predicted values of the dependent variable Y: **the method of least squares** (*méthode des moindres carrés*).

# 2 Simple regression

Before accomplish the task in Section 3, in this section we provide a step-by-step example of a simple regression. This example be found in [1].
**Problem description**

Currently, the air pollution is one the main problems in terms of public health. Several epidemiological studies have allowed to put in evidence the influence on the health of certain chemicals such as sulfur dioxide (SO2), nitrogen dioxide (NO2), ozone (O3) or particles, as dust, in the air.

Associations monitoring air quality exist throughout the French territory and measure the concentration of pollutants. They also record the weather conditions such as temperature, cloud cover, wind, etc. In this example, the aim is to analyze the relationship between the daily maximum ozone concentration (in $\mu g/m^3$) and temperature. We have 112 observations collected during the summer of 2001 in Rennes. **These observation are in two different the files (`ozoneTrain.txt` and `ozoneTest.txt`) that you can download**

**from CELENE.**

1. Load the data.
2. Represent the data as a scatter plot of $(x_i, y_i)$.
3. Estimate the parameters.
4. Plot the regression line.
5. Analyze the residuals/diagnostics.
6. Forecast new values.

## 2.1 Loading the data

Supposing that the data set is in your current working directory, load it into R. Then, display a summary of only the variable of interest: `maxO3` (ozone) and `T12` (temperature).

```
ozoneTrain <- read.table("ozoneTrain.txt", header=TRUE)
summary(ozoneTrain[,c("maxO3","T12")])
```

## 2.2 Represent the data as a scatter plot

```
plot(maxO3 ~ T12,data=ozoneTrain,pch=15,cex=0.5,col="blue")
```

Each point in the graphic generated represents, for a given day, a measure of the temperature at 12AM and maximum of the ozone of the day.

**From the graphic generated what type of relation can you observe between the temperature and concentration of ozone? Does this seem linear?**

## 2.3 Estimating the parameters of regression model

The R function `lm` allows the adjustment of a linear model, via a least-squares procedure, to the data. More specifically, in our case we want to forecast the maximum concentration of ozone (`maxO3`) based on the temperature (`T12`). We will store the model generated in the object `reg.simple`.

```
> reg.simple <- lm(maxO3 ~ T12, data = ozoneTrain)
> summary(reg.simple)
```

The application of the function `summary` to the model returns important information. For example, we have a matrix `Coefficients` that contains for each parameter (each line) four columns:

- `Estimate`: the estimation from the data for the parameter value.
- `Std. Error`: the estimated standard deviation for the parameter.
- `t value`: the value observed for the statics used in the hypothesis test ($\mathbf{H}_0 : \beta_i = 0$ against $\mathbf{H}_1 : \beta_i \neq 0$).
- `Pr(>|t|)`: probability for a null hypothesis that the coefficients have values of zero.

In the specific case of our data, the values estimated for $\beta_0$ and $\beta_1$ are, respectively, `-20.6098` and `5.1330`. That is,

$$\hat{y_i} = -20.6098 + 5.1330 * x_i$$

In both cases ($\hat{\beta_0}$ and $\hat{\beta_1}$), the null hypotheses are rejected, since the critical probabilities (respectively, `0.0318` and `<2e-16`) are below the significance level of `0.05` and `0.001` — often, the default for the significance level is of 0.05 (5%) . For example, a significance level is of 0.05 (5%) has the meaning that we are 95% confident that the coefficient is not zero.

**Important: The critical probability smaller than 5% for the constant term ($\beta_0$) indicates that it should appear in the model. The critical probability smaller than 5% for the slope (*pente*), $\beta_1$, indicates a significant link between `maxO3` and `T12`.**

At the bottom of the output of `summary` we find other important information such as the standard deviation about the regression or `residual standard error` (`17.44`, in this example), `Multiple R-squared` (0.5827) and the `Adjusted R-squared` (0.5784). These last two values represents the proportion of the total sample variability. They are often used as an estimate of the *goodness* of the model generated: the close to 1 the better.

The last line is mostly used in the context of multiple regression. It indicates the result of a test comparing the model generated and a simple model that uses only $\beta_0$ like explanatory variable (that is, $\mathbf{H}_0 : \beta_1 = ... = \beta_p = 0$). If the null hypothesis is not rejected this will imply that there is no evidence for the relation between the explanatory variable and the target variable.

To have access to the different lists that compose the output of the function `lm`, you can type:

```
>names(reg.simple)
```

## 2.4 Plotting the regression line

Here we plot first the scatter plot of temperature versus ozone and, then, for comparison, we plot the least-squares fit we have generated (`reg.simple`).

```
>plot(maxO3 ~ T12,data=ozoneTrain,pch=15,cex=0.5,col="blue")
>abline(reg.simple,col="red")
```

## 2.5 Analyzing the residuals/Diagnostics

When we are perform a regression analysis we make some assumptions such as the errors are normally distributed. One way to check this assumption is by visualizing a plot of the residuals and the predicted values. The residuals (errors) can be obtained via the the function `residuals`. Often, to compare residuals at different inputs, one needs to adjust the residuals by the expected variability of residuals, which is called studentizing (`rstudent`). In the case of our data, we could use the following sequence:

```
> res.stu <- rstudent(reg.simple)
> plot(res.stu,pch=15,cex=0.5,ylab="Residuals",ylim=c(-3,3),col="blue")
> abline(h=c(-2,0,2),lty=c(2,1,2),col="red")
```

In theory, 95% of the studentizing residuals should be distribute around the interval [-2,2]. In the context of this data, looking at the figure generated there are only about four residuals that are outside the [-2,2] range. They should be inspect because they might represent outliers.

We can also try to detect nonlinearity by using a plot of the observed versus predicted values. In the case of linearity, the points should be symmetrically distributed around a diagonal line. For example, looking at

the plot generated by the command below we can see that in the our case we can observe that points are indeed symmetrically distributed around a diagonal.This plot also gives information about the performance of the model — if each observation had been predicted correctly all points in the plot would be on the diagonal.

```
> plot(reg.simple$fitted.values,ozoneTrain[,"maxO3"])
> abline(0,1)
```

### 2.6 Forecasting new values

Having a new observation or a set of new observations, we can directly apply the model generated. For example, suppose that we want to use the model `reg.simple` to predict the values of `maxO3` for the data in `ozoneTest.txt`, we can proceed as follows:

```
ozoneTest <- read.table("ozoneTest.txt", header=TRUE)
reg.pred <- predict(reg.simple, ozoneTest)
```

After making the predictions is interesting to have some kind of visual inspections of the quality of the results. A possibility is to use a scatter plot of the errors. One can produce this plot using the following sequence of commands:

```
reg.pred <- predict(reg.simple, ozoneTest)
plot(reg.pred,ozoneTest[,"maxO3"],main="linear Model", xlab="Predictions",ylab="True values")
abline(0,1)
```

In the ideal scenario that the model makes the correct predictions for all cases, all circles in the plot should lie on the line (`abline(0,1)`). Such a line cross the origin of the plot and represents the points where the X-coordinate is equal to the Y-coordinate. Given that each circle in the plot obtains its coordinates from the predicted and the true values of the target variable, if these values were equal, the circle would be place on this ideal line.

In the case of our data, as we can observe most values are symmetrically distributed and close to the diagonal line. Another way to make a more quantitative analysis of result obtained is by the Normalized Mean Mean Squared Error (NMSE). This statistic calculates a ratio between the performance of our model and that of a baseline predictor, usually taken as the mean value of the target variable [2]:

```
>(nmse.reg <- mean((reg.pred - ozoneTest[,"maxO3"])^2)/
+ mean((mean(ozoneTest[,"maxO3"])-ozoneTest[,"maxO3"])^2))
```

In the case for the predictions for `ozoneTest` we obtain a NMSE of `0.3442664`. The values of NMSE usually range from 0 to 1. If the model is performing better than the very simple baseline predictor, then the NMSE should be clearly less than 1. That is, the smaller the NSME, the better. Values greater than 1 mean that the model generated is performing worse than simply predicting always the average for the observations.

## 3 Multiple regression

The task is to model, use linear regression, the problem of prediction of algae frequency that we started to work in TD4. For this we have two data files: the "Training data" contains the 200 water samples in the file `algae.RData`, while the "Test data" stored in `testAlgae.RData` contains the 140 test samples. There is an additional file (`algaeSols.RData`) that contains the algae frequencies of the 140 test samples. This latter file will be used to check the performance of our predictive models and will be taken as unknown information at the time of model construction. All these files can be loaded in R using `load('filename.RData')` . After loading, the names of the respective data frames will be: `algae`, `test.algae` and `algae.sols`.

In the previous context, you should generate the regression model using `testAlgae.RData` and test it with `testAlgae.RData`. Important: before generating the model, apply filters to remove/replace missing values. Also, let's focus on the prediction of only `a1`. For example, supposing that we have stored the "clean" data

in the variable `cleanAlgae`, you can generate the regression model using:

```
lm.a1 <- lm(a1 ~ .,data=cleanAlgae[,1:12])
```

**However, before doing the previous step we should first visualize data as a scatter plot of the *numeric* explanatory variables with respect to the target variable `a1`. This could give you some insight about if there exists some linear relation.**
In terms of analysis, we can follow the steps described in the previous section.

# References


[1] P-A. Cornillon, A. Guyader, F. Husson, N. Jégou, J. Josse, M. Kloareg, E. Matzner-Løber, and L. Rouvière.
*Statistiques avec R.*
Presses Universitaires de Rennes, France, 2008.

[2] Luis Torgo.
*Data Mining with R: Learning with Case Studies.*
Chapman & Hall, USA, 2011.