breakDown plots : : **visual explanations for lm/glm models**

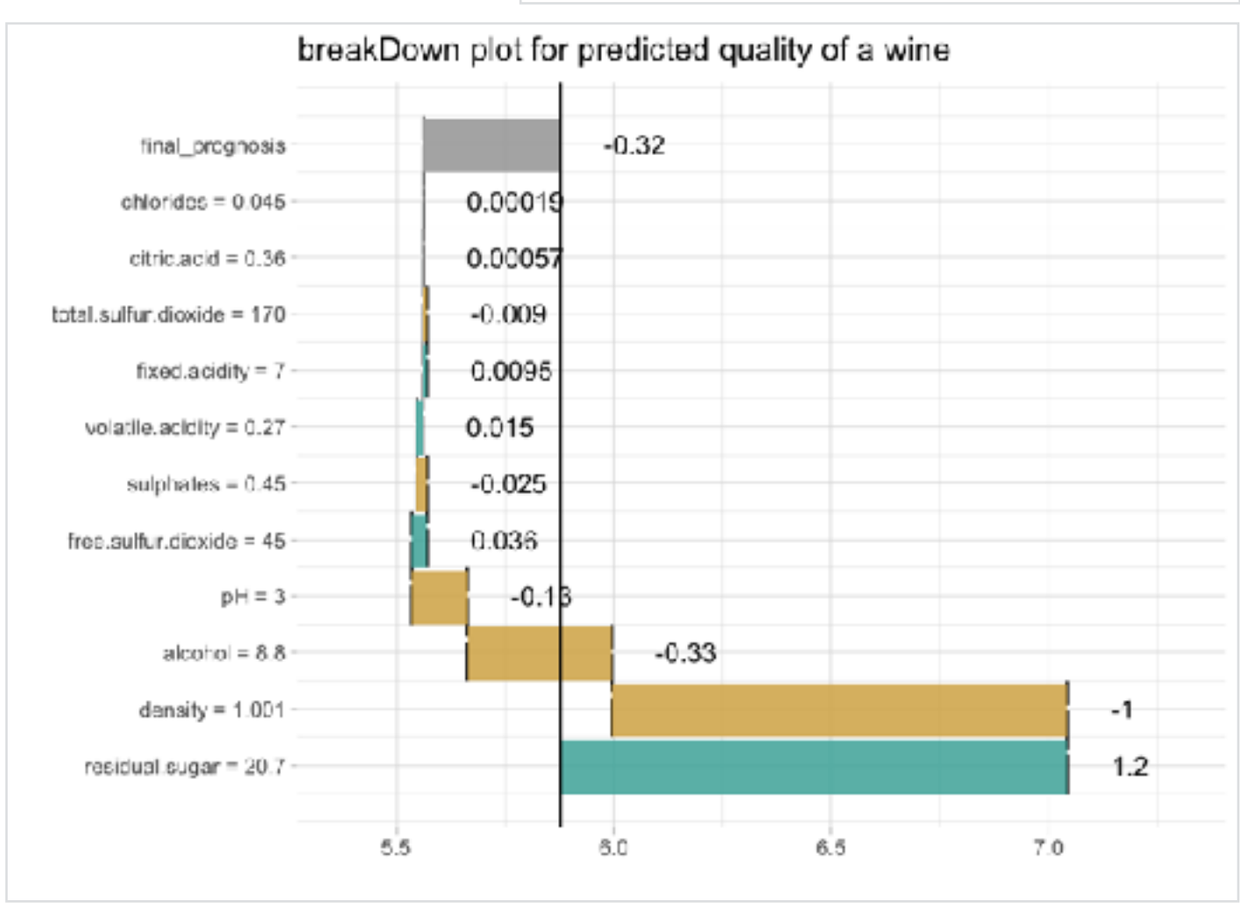
Linear model

Linear models are widely used in predictive modeling.
They have simple structure, which makes them easy to deploy or implement.
But models with many variables are hard to understand.

The **breakDown** plot explains the relation between variables and model prediction for a new observation.

Explanations are generated in three steps:
1) create model with **lm**() function
2) break down model predictions with the **broken()** function
3) plot the graphical summary with the generic **plot**() function.

```
library(breakDown)
library(ggplot2)
model <- lm(quality ~ ., data = wineQuality)
br <- broken(model, wineQuality[1,],
          baseline = "Intercept")
br
#>                                contribution
#> residual.sugar = 20.7               1.20000
#> density = 1.001                    -1.00000
#> alcohol = 8.8                      -0.33000
#> pH = 3                             -0.13000
#> free.sulfur.dioxide = 45            0.03600
#> sulphates = 0.45                   -0.02500
#> volatile.acidity = 0.27             0.01500
#> fixed.acidity = 7                   0.00950
#> total.sulfur.dioxide = 170         -0.00900
#> citric.acid = 0.36                  0.00057
#> chlorides = 0.045                   0.00019
#> final_prognosis                    -0.32000
#> baseline:  5.877909
plot(br)
```



Logistic regression

breakDown plots may be also used to explain predictions from the logistic regression model.

On the OX axis one may present linear predictions (default) or use probit/logit transformation to present contributions of variables from the model. Use the trans= argument to define the transformation.

The baseline is presented by the vertical black line in the plot. One may set the baseline to 0 or to population average *(*use the *baseline = "intercept"* argument*).*

```
library(breakDown)
library(ggplot2)
model <- glm(left~., data = HR_data,
                family = "binomial")
explain_1 <- broken(model, HR_data[11,],
                baseline = "intercept")
explain_1
#>                                 contribution
#> satisfaction_level = 0.45              0.670
#> number_project = 2                     0.570
#> salary = low                           0.390
#> average_montly_hours = 135            -0.290
#> Work_accident = 0                      0.220
#> time_spend_company = 3                -0.130
#> last_evaluation = 0.54                -0.130
#> promotion_last_5years = 0              0.030
#> sales = sales                          0.014
#> final_prognosis                        1.300
#> baseline:  -1.601457
plot(explain_1, trans = function(x)
              exp(x)/(1+exp(x)))
```