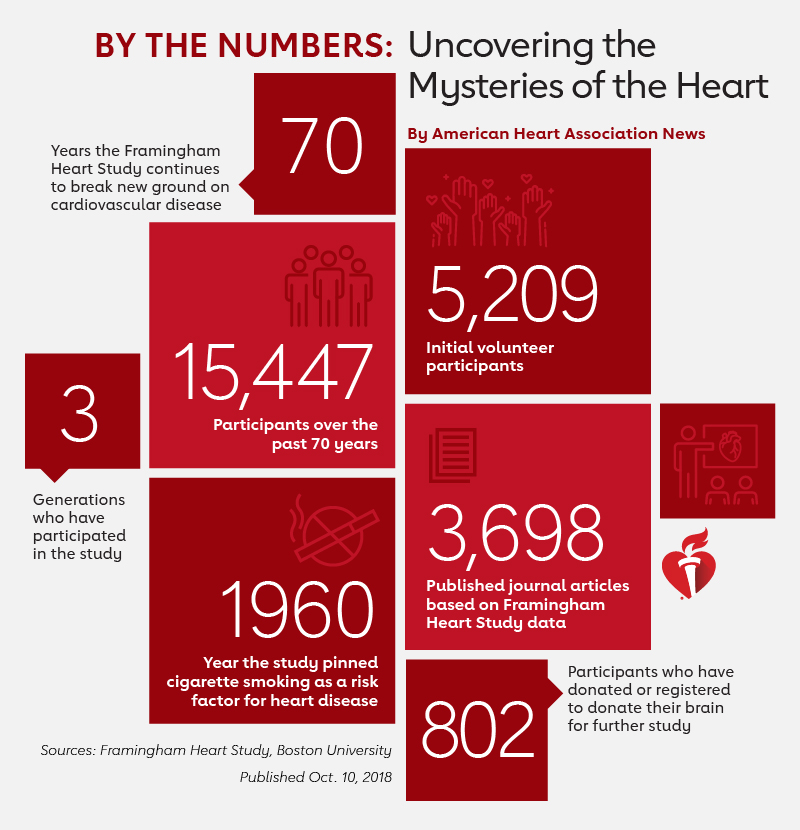
**Lab 3**

[The Framingham Heart Study](https://en.wikipedia.org/wiki/Framingham_Heart_Study) is a long term prospective study of cardiovascular disease among a population of subjects in the community of Framingham, Massachusetts. The Framingham Heart Study was a landmark study in epidemiology in that it was the first prospective study of cardiovascular disease and identified the concept of risk factors and their joint effects over the course of three generations. The study began in 1948 and 5,209 subjects were initially enrolled in the study. Participants have been examined biennially since the inception of the study and all subjects are continuously followed through regular surveillance for cardiovascular outcomes.

You will find the data file [***framinghamHeart.csv***](https://drive.google.com/file/d/1ccwKjZMjS0_togcNTtY-h9ejAKP-fScv/view?usp=sharing), which you can load as **dff**. This is a subset of the data collected as part of the Framingham study. Participant clinic data was collected during three examination periods, approximately 6 years apart, from roughly 1956 to 1968. Each participant was followed for a total of 24 years for the outcome of a specified set of adverse health events. The dependent variable is **TenYearCHD**, specifying whether a subset of events associated with chronic heart disease occurred within 10 years of follow up. The variables are defined below. The purpose of the study is to determine the risk factors of heart disease.

****

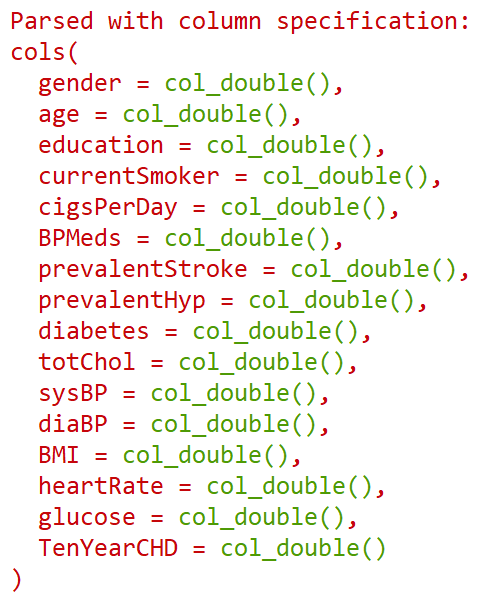
**Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Coding** |
| **gender** | Male or Female | 0 = Female; 1 = Male |
| **age** | Age of the patient |  |
| **education** | Highest level of education achieved | 1 = High School; 2 = High School Diploma or GED; 3 = Some college or vocational School; 4 = College degree |
| **currentSmoker** | Indicates if the person is currently a smoker or not | 0 = Not a smoker; 1 = Is a smoker |
| **cigsPerDay** | The number of cigarettes the person smoked on average in one day |  |
| **BPMeds** | Whether the patient was on blood pressure medication | 0 = Not on BP meds; 1 = On BP meds |
| **prevalentStroke** | Whether the patient previously had a stroke | 0 = Free of disease; 1 = Stroke |
| **prevalentHyp** | Whether the patient has hypertension (high blood pressure) | 0 = Free of disease; 1 = Hypertension |
| **diabetes** | Whether the patient has diabetes | 0 = Free of disease; 1 = Diabetes |
| **totChol** | Total cholesterol level | mg/dL |
| **sysBP** | Systolic blood pressure | mmHg |
| **diaBP** | Diastolic blood pressure | mmHg |
| **BMI** | Body Mass Index | Weight (kg) / Height (meter-squared) |
| **heartRate** | Heart rate | Beats/Min (Ventricular) |
| **glucose** | Glucose level | mg/dL |
| **TenYearCHD** | Coronary heart disease | ‘0' indicates the event did not occur during the 10-year follow up, and ‘1' indicates an event did occur during the follow up |

**Data Analysis**

Before you start, **load the “caret” library** in addition to the usual four libraries we always load.

In addition, pay attention to what R reports after you load the dataset:



Notice that R reads all the columns as numbers. You know from the data dictionary that some variables are supposed to be factors. You need to ask R to convert them into factors:

i. Create a list of columns that are supposed to be factors:

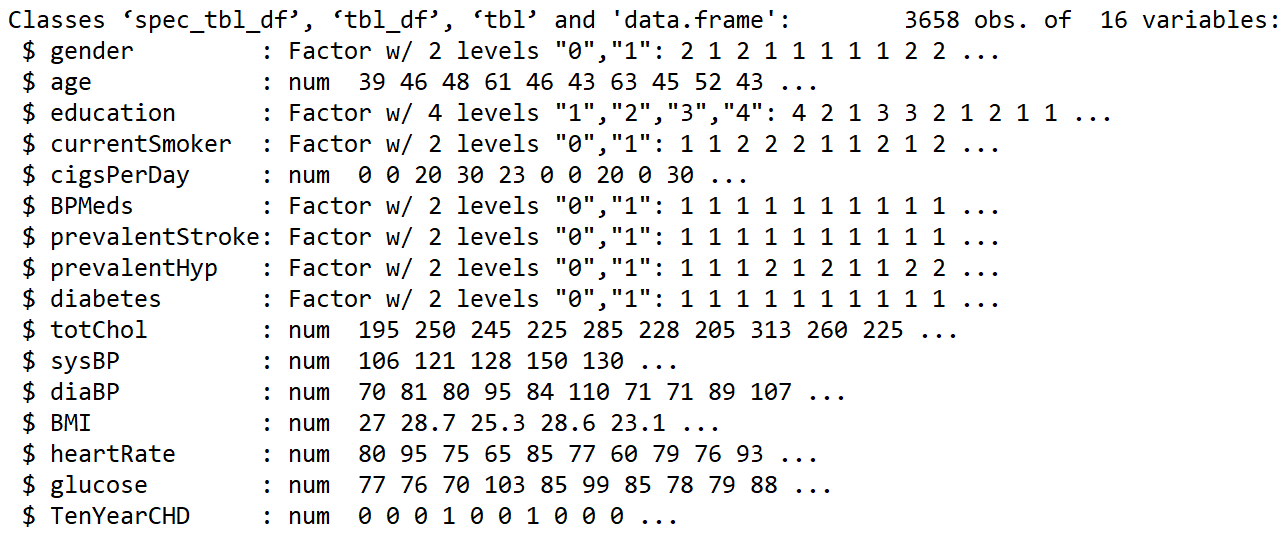
colsToFactor <- c('gender', 'education', 'currentSmoker', 'BPMeds', 'prevalentStroke', 'prevalentHyp', 'diabetes')

ii. Ask R to replace (overwrite) selected variables with their factor conversions:

dff <- dff %>%

  mutate\_at(colsToFactor, ~factor(.))    *=> What do you think* ***mutate\_at*** *does?*

Now, if you run str(dff), you will see that the variables in your data are correctly identified:



1. **Data exploration:** To explore visually whether blood pressure levels and total cholesterol levels are associated with heart disease, create boxplots of *sysBP*, *diaBP*, and *totChol*, broken up by the levels of *TenYearCHD*. [ **Hint:** Dynamic plots may help understanding! ]

1. **Data preprocessing:**

(i) Read the data file into R. Set the seed to **123** and split the data into dffTrain and dffTest. Randomly sample 70% of the data for training, and use the rest as test dataset.

(ii) What are the proportions by gender in your training vs. test set? How does the distribution of age look? Looking at these, do you observe any signs of a sampling bias?

**Hints:**

[A] It’s time to use R like a pro! You can pipe your *dffTrain* into the group\_by(*variable*) function and then into **tally()** *-no arguments-* to get the counts across a group.

* To add percentages, pipe one more step into mutate(pct = 100\*n/sum(n))

[B] For a continuous variable like age, there are so many groups, right? Each age is practically a different group. In such cases, you may want to create your own groups.

* You can use *ageGroup=cut\_interval(age, length=10)* in group\_by()

[C] You can also create a histogram for age, which probably makes more sense.

* After creating the histogram, try adding fill=gender into aes() of ggplot(), and see what happens. In addition, define color='black' inside the histogram!

1. **Linear probability model:** Build a linear probability model fitLPM using all variables in dffTrain. Make sure to check for collinearity by both thinking about the variables, and using VIF values as guiding signals, and take necessary precautions. You know how to mitigate collinearity (if not, please ask during the lab!). After finalizing the model, which of the variables are statistically significant at the 95% level? What does this model tell you about the risk factors of heart disease? Do you have any reservations? Discuss.

**Hints:**

[A] To include all the variables, use a full stop **.** To exclude a variable, use a negative **-**

[B] Run diagnostics to see whether this model violates the linear regression assumptions.

1. Speaking of using R like a pro, a better way to run a model and create a results table with predictions is as follows. Please run this code to make predictions using the LPM model and store them into *resultsLPM*

resultsLPM <-

    lm( *…fill in here…* ) %>%

    predict( *…fill in here…* ) %>%      *=> Use the option* ***type='response'*** *for probabilities*

    bind\_cols(dffTest, predictedProb=.) %>%     *=> The dot marks where to pipe into*

    mutate(predictedClass = *…fill in here…* )     *=> Use 50% as cutoff for classification*

Inspect resultsLPM. Then, **copy and paste your code from Q2-ii** and check the prevalence of *TenYearCHD* in the *test dataset* this time. How many people have heart disease in reality (in the test dataset)? Run the same code for *predictedClass* in the *resultsLPM*. How many people did the model predict having heart disease? Compare and report your observations.

**Before you continue:**

You may have noticed that we did not convert TenYearCHD into a factor yet, even though it is a factor. This is because we wanted to use it in a linear model. It is time to make it a factor.

* Use mutate() to convert TenYearCHD to a factor both in *dffTrain* and *dffTest* datasets.

1. **Logistic regression:** Build a logistic regression using the predictor variables you decided to keep in the model you built in Q3. Which variables are statistically significant at the 95% level? Compare your results with the results you obtained from the model in Q3.

**Hint:** See the appendix for an annotated logistic regression output in R with the definitions.

Interpret the following variables: *age*, *gender*, and *diabetes* (whether significant or not):

* **Hint:** You can run **exp(coef(fit))** after a logistic model to exponentiate the coefficients of all variables at once, and use them in your interpretations.
* **Type these interpretations AFTER completing the lab unless you have any questions.**

Age:

Gender:

Diabetes:

1. Create a new results table ***resultsLog*** by using the logistic model. Let’s continue like a pro.

**Hint:** You will follow the same steps you took in Q4 but this time for logistic regression. This means, **your predictedClass will need to be defined as a factor** (you know how to do this!).

How many people did the logistic model predict having heart disease? Report your observations and compare them with the actual values, and the predictions of the linear probability model from Q4. Do you think the logistic model is an improvement? Why?

**Hint:** For now, continue to use your code from Q2-ii to create the tables for comparison.

1. It is time to create a confusion matrix, a final step before evaluating performance (which we will cover next week). As you’re using R like a pro, it is so easy to create a confusion matrix.

* Pipe the *resultsLog* dataframe you created in **Q6** into the function conf\_mat(truth = ..., estimate = ...)
* **Optional:** Pipe one more step into autoplot(type = 'heatmap') to color code. This is useful when more than two classes are involved. For now, this is just a learning point.

Explain what the matrix tells you in addition to what you learned from the tables in **Q6**.

1. No analysis is complete without a visualization. Plot the relationship between the statistically significant variables (*age*, *cigsPerDay*, *totChol*, *glucose*) and the probability of heart disease:

* Note that you stored the predicted probabilities as *predictedProb* in the *resultsLog* in Q6.
* Use geom\_point() and geom\_smooth() after ggplot(), without adding any parameters
* Be creative. For example, add color=currentSmoker (or =gender) into the aes()
* Add a title for the plots, and label both axes [ **Hint:** You can use the labs() function ]

Discuss your observations.

**Switching to a new framework “Caret” we will continue to use in this course from now on:**

1. You already loaded the “caret” library at the beginning. If not, load it now. Replicate the analysis in Question 6, this time using the caret library. Use Appendix II for guidance.

* Name the results table resultsLogCaret and create it using the train function.
* Inspect resultsLogCaret carefully, compare it with resultsLog from Q6 and discuss.
* Create the confusion matrix using caret, and compare it with the one in Q7. Discuss.
* Don’t worry about the rest of the output after the matrix. We will discuss it next week!

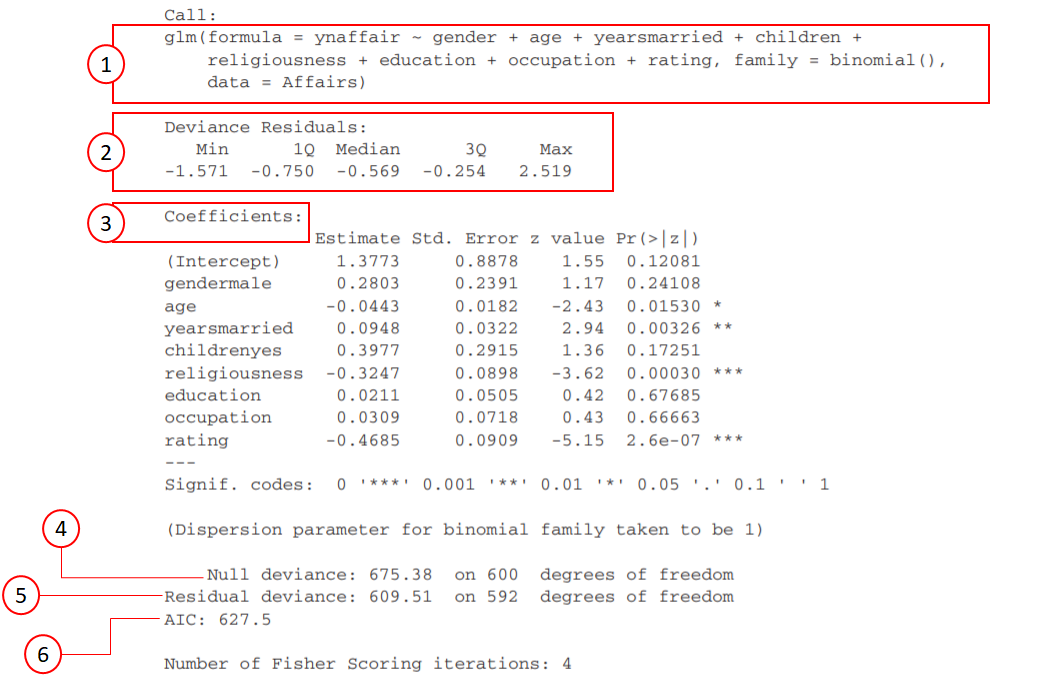
1. Now that you have learned how to use logistic regression for classification, and how to do so **using the caret library**, you can solve another business problem for *Banco Portugal*. See Appendix III for the details of [the dataset](https://drive.google.com/file/d/1Cx0hlASmHUGQMnUD8OOxkdO_f0d1lKtp/view?usp=sharing). The bank runs a telemarketing campaign for a savings account. Have you ever received one of those promotions by the way? “Open a savings account today and get XXX$ bonus!” See this month’s promotions by clicking [here](https://www.nerdwallet.com/blog/banking/best-bank-bonuses-promotions/).

Banco Portugal hires you to predict whether a customer will open an account. The bank will use your model to develop promotional campaigns with higher conversion rates. Load the data, make conversions of variables as you see fit, and build logistic regression models using the caret library. Explore at least three alternative models, compare their performance, and pick a final model. Show your full work in the R Notebook. Below, discuss only your findings, your final decision, and explain how your final model helps Banco Portugal with its purpose.

Now that we have discussed the performance measures, you can decide on a performance metric (or two) beyond just accuracy to compare the models and explain your reasoning. Because the caret library already reports the values of performance measures by default, you don’t need to do any coding -This part is pretty much a thinking and reflecting exercise!

**Appendix I: How to run logistic regression in R and read the regression output**

The output from summary() may seem overwhelming at first, so let’s break it down one item at a time:

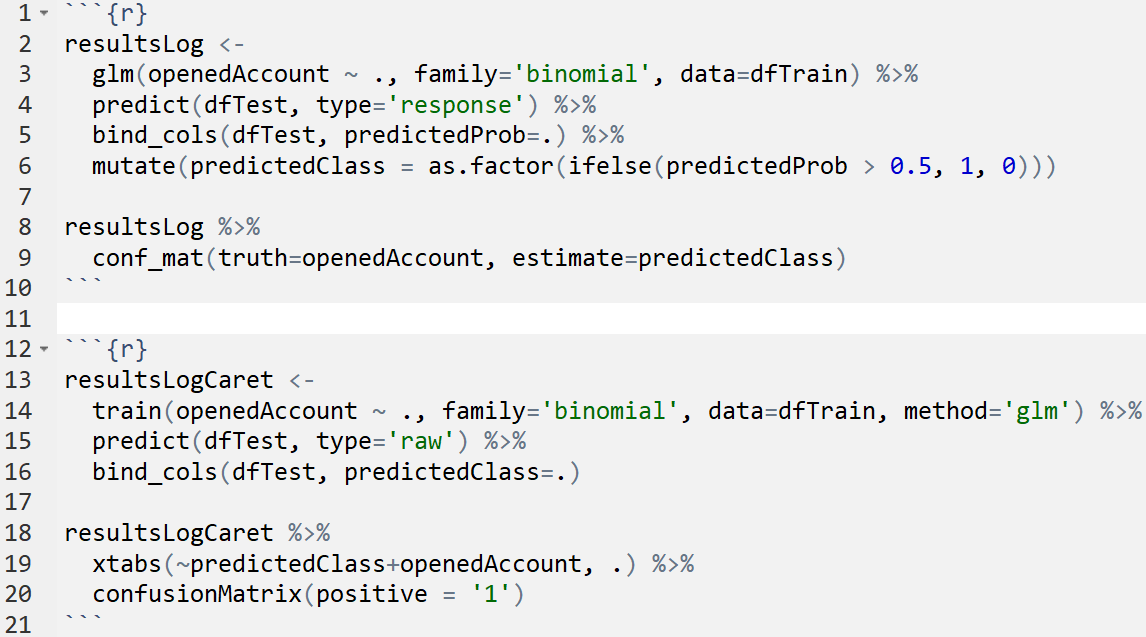


|  |  |  |
| --- | --- | --- |
| **#** | **Item** | **Description** |
| **1** | Formula | Like it was in the linear regression, the *glm()* formula describes the relationship between the dependent and independent variables. Note that you need to include *family = ‘binomial’* as an argument. |
| **2** | Deviance Residuals | Because the difference between the observed and the fitted values are not very informative in a logistic regression, R reports the deviance residuals, which are the signed square roots of the ith observation to the overall deviance, calculated as follows: |
| **3** | Coefficients | The regression coefficients show the change in log(odds) in the dependent variable for a unit change in the predictor variable, holding all other predictor variables constant.  Because log(odds) are difficult to interpret, we usually exponentiate the coefficients and convert them into the odds scale:  exp(the coefficient of yearsmarried) = exp(0.0948) = 1.10,  which means a 1-year increase in the number of years married is associated with an increase in the odds of an affair by a factor of 1.10 (about a 10% increase), holding everything else constant.  *What about a 10-year increase in the number of years married?*  If you interpret a categorical variable like gendermale, exp(0.2803)=1.32 becomes the odds ratio. Therefore, the odds of a male having an affair are about 32% higher than the odds of a female doing so, holding everything else constant.  You can exponentiate all coefficients by running exp(coef(fit)) |
| **4-5** | Null Deviance, and  Residual Deviance | The *null deviance* shows how well the dependent variable is explained by a model that includes only the intercept.  The *residual deviance* shows how well the dependent variable is explained by a model that includes all the independent variables. |
| **6** | AIC | The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models. It’s based on the Deviance measure, but includes a penalty for including additional independent variables. Much like adjusted R-squared, it intends to help you leave irrelevant predictors out.  However, unlike adjusted R-squared, the reported number itself is not meaningful. When you compare nested models, you should select the model that has the smallest AIC.  For BIC, run BIC(fit) after a regression, where *fit* is the model name, and R will report the BIC score. All of this also applies to BIC. |
| **7** | Fisher Scoring | This is just showing the number of iterations the model went through before it converged to this solution (not really useful). |

**Appendix II: Modeling using native way vs. the Caret way**

**Line by line comparison of making predictions using a logistic regression native way vs. caret way:**

Note that the dependent variable is openedAccount in the example below:



**Line 14 vs. Line 3:** Use train() function instead of glm() and define the method in the method argument.

**Line 15 vs. Line 4:** Use predict() with type=’raw’ to get the predicted classes instead of probabilities.

**Line 16 vs. Line 5:** Name the column as predictedClass instead of predictedProb for this reason.

**N/A vs. Line 6:** No need to use a mutate() function to convert probabilities into classes.

**Line 19 vs. N/A:** Use the xtabs() function only because confusionMatrix() needs one.

**Line 20 vs. Line 9:** Use confusionMatrix() rather than conf\_mat() and define the positive class.

**Appendix III: Details of the Banco Portugal savings account dataset**

**Relevant Information:**

The bank’s customer-level data is extended by the addition of five social and economic features/predictors (at the end of data dictionary, national-wide indicators from Portugal), published by the Banco de Portugal and publicly available at [bportugal.pt/estatisticasweb](https://www.bportugal.pt/estatisticasweb)

**Source:**

Sérgio Moro (ISCTE-IUL), Paulo Cortez (Univ. Minho) and Paulo Rita (ISCTE-IUL) @ 2014

**Past Usage:**

The full dataset was described and analyzed in:

S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems (2014), doi:10.1016/j.dss.2014.03.001.

**Objective:**

The classification goal is to predict if a customer will open a savings account (*accountOpened*).

**Data Summary:**

Number of observations: 41188    Number of variables: 20+

**Data Dictionary:**

For more information, you can refer to Moro et al. (2014) cited above.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data type** | **Description** |
| openedAccount | categorical | Has the customer opened a savings account? ("yes","no") |
| newcustomer | categorical | If the customer is a new customer or not (yes = 1, no=0) |
| age | numeric | Age of the customer |
| agegroup | categorical | The age group that the customer belongs to ("Teenagers", "Young Adults", "Adults", "Senior Citizens") |
| job | categorical | Type of job ("admin", "blue-collar", "entrepreneur", "housemaid", "management", "retired", "self-employed", "services", "student", "technician", "unemployed", unknown) |
| marital | categorical | Marital status ("divorced", "married", "single", "unknown"; note: "divorced" means divorced or widowed) |
| education | categorical | Educational qualification ("basic.4y", "basic.6y", "basic.9y", "high.school", "illiterate", "professional.course", "university.degree", "unknown") |
| default | categorical | Has credit in default? ("no", "yes", "unknown") |
| housing | categorical | Has a housing loan? ("no", "yes", "unknown") |
| loan | categorical | Has a personal loan? ("no", "yes", "unknown") |
| contact | categorical | Contact communication type ("cellular", "telephone") |
| month | categorical | Last contact month of year ("jan", "feb", ..., "nov", "dec") |
| day\_of\_week | categorical | Last contact day of the week ( "mon","tue","wed","thu","fri") |
| duration | numeric | Last contact duration, in seconds  **Important**: This attribute highly affects the outcome (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call the outcome is obviously known. So, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model. |
| campaign | numeric | Number of contacts performed during this campaign and for this client (includes the last contact) |
| pdays | numeric | Number of days passed by after the client was last contacted from a previous campaign **(“999” means client was not previously contacted)** |
| previous | numeric | Number of contacts performed before this campaign |
| poutcome | categorical | Outcome of the previous marketing campaign ("failure", "nonexistent" ,"success") |
| emp.var.rate | numeric | Employment variation rate - quarterly indicator |
| cons.price.idx | numeric | Consumer price index - monthly indicator |
| cons.conf.idx | numeric | Consumer confidence index - monthly indicator |
| euribor3m | numeric | Euribor 3 month rate - daily indicator |
| nr.employed | numeric | Number of total employment - quarterly indicator |