Data Understanding

Adding to the foundation of Business Understanding, the Data Understanding phase focuses on identifying, collecting, and analyzing data sets that can help the project. This phase also has four tasks:

See *Variables Overview.xlsx* for further information!

1. Collect initial data: Acquire the necessary data and (if necessary) load it into your analysis tool.

* **Data overview**
* # Load dataset with separator ";", a header and empty cells recognized as NAs  
  data <- read.csv("marketing.csv", sep = ";", header = TRUE, na.strings = "")  
    
  # Check the data set------------------------------------------------------------  
  head(data) # Displays the first lines of the data set
* age job marital education default housing loan month day\_of\_week  
  1 56 housemaid married basic.4y no no no may mon  
  2 57 services married high.school <NA> no no may mon  
  3 37 services married high.school no yes no may mon  
  4 40 admin. married basic.6y no no no may mon  
  5 56 services married high.school no no yes may mon  
  6 45 services married basic.9y <NA> no no may mon  
   campaign previous poutcome emp.var.rate cons.price.idx cons.conf.idx  
  1 1 0 nonexistent 1.1 93.994 -36.4  
  2 1 0 nonexistent 1.1 93.994 -36.4  
  3 1 0 nonexistent 1.1 93.994 -36.4  
  4 1 0 nonexistent 1.1 93.994 -36.4  
  5 1 0 nonexistent 1.1 93.994 -36.4  
  6 1 0 nonexistent 1.1 93.994 -36.4  
   euribor3m y  
  1 4.857 no  
  2 4.857 no  
  3 4.857 no  
  4 4.857 no  
  5 4.857 no  
  6 4.857 no
* str(data) # Outputs a summary of the data structure
* 'data.frame': 41188 obs. of 17 variables:  
   $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
   $ job : chr "housemaid" "services" "services" "admin." ...  
   $ marital : chr "married" "married" "married" "married" ...  
   $ education : chr "basic.4y" "high.school" "high.school" "basic.6y" ...  
   $ default : chr "no" NA "no" "no" ...  
   $ housing : chr "no" "no" "yes" "no" ...  
   $ loan : chr "no" "no" "no" "no" ...  
   $ month : chr "may" "may" "may" "may" ...  
   $ day\_of\_week : chr "mon" "mon" "mon" "mon" ...  
   $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
   $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
   $ poutcome : chr "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...  
   $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
   $ cons.price.idx: num 94 94 94 94 94 ...  
   $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
   $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
   $ y : chr "no" "no" "no" "no" ...
* nrow(data) # cross checking number or rows
* [1] 41188
* ->17 variables (columns) and 41’188 observations (rows)
* -> Types: chr, int, and num
* **Missing values (NAs)**
* ***Variable Balance is not in the dataset !***
* # Determine the number of missing values for each variable----------------------  
  missing\_values <- sapply(data, function(x) sum(is.na(x)))  
  missing\_values # Show the number of missing values for each variable
* age job marital education default   
   0 330 80 1731 8597   
   housing loan month day\_of\_week campaign   
   990 990 0 0 0   
   previous poutcome emp.var.rate cons.price.idx cons.conf.idx   
   0 0 0 0 0   
   euribor3m y   
   0 0
* **Relationships between variables:** previous = 0 & poutcome = nonexistent
* # Relationship between previous = 0 & poutcome = nonexistent--------------------  
  # Number of previous = 0  
  count\_previous\_0 <- sum(data$previous == 0)   
  count\_previous\_0
* [1] 35563
* # Number of poutcome ="nonexistent"  
  count\_poutcome\_nonexistent <- sum(data$poutcome == "nonexistent")   
  count\_poutcome\_nonexistent
* [1] 35563
* # Number of previous = 0 & poutcome = "nonexistent" at the same time  
  count <- sum(data$poutcome[data$previous == 0] == "nonexistent")   
  count
* [1] 35563
* # check if relationship is true (all numbers are equal)  
  realtionship <- (count\_previous\_0 == count\_poutcome\_nonexistent) && +  
   (count\_previous\_0 == count) && (count\_poutcome\_nonexistent == count)  
  realtionship
* [1] TRUE

1. Describe data: Examine the data and document its surface properties like data format, number of records, or field identities.

* See *Variables Overview.xlsx* for further information!
* **Statistical Analysis**
* **of numeric variables**
* # statistical analysis of the data----------------------------------------------  
  # statistical Summary of numeric variables  
  summary\_numeric <- summary(data[, sapply(data, is.numeric)])   
  summary\_numeric
* age campaign previous emp.var.rate   
   Min. :17.00 Min. : 1.000 Min. :0.000 Min. :-3.40000   
   1st Qu.:32.00 1st Qu.: 1.000 1st Qu.:0.000 1st Qu.:-1.80000   
   Median :38.00 Median : 2.000 Median :0.000 Median : 1.10000   
   Mean :40.02 Mean : 2.568 Mean :0.173 Mean : 0.08189   
   3rd Qu.:47.00 3rd Qu.: 3.000 3rd Qu.:0.000 3rd Qu.: 1.40000   
   Max. :98.00 Max. :56.000 Max. :7.000 Max. : 1.40000   
   cons.price.idx cons.conf.idx euribor3m   
   Min. :92.20 Min. :-50.8 Min. :0.634   
   1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.344   
   Median :93.75 Median :-41.8 Median :4.857   
   Mean :93.58 Mean :-40.5 Mean :3.621   
   3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961   
   Max. :94.77 Max. :-26.9 Max. :5.045
* # standard deviation of of numeric variables  
  sd\_numeric <- sapply(data[, sapply(data, is.numeric)], sd)  
  sd\_numeric
* age campaign previous emp.var.rate cons.price.idx   
   10.4212500 2.7700135 0.4949011 1.5709597 0.5788400   
   cons.conf.idx euribor3m   
   4.6281979 1.7344474
* **of discrete** **numeric variables (integers)**
* # Frequency of the individual values for discrete numeric variables (integers)  
  frequency\_int\_data <- lapply(data[, sapply(data, is.integer)], table)  
  frequency\_int\_data
* $age  
    
   17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32   
   5 28 42 65 102 137 226 463 598 698 851 1001 1453 1714 1947 1846   
   33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48   
  1833 1745 1759 1780 1475 1407 1432 1161 1278 1142 1055 1011 1103 1030 928 979   
   49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64   
   839 875 754 779 733 684 648 704 646 576 463 283 73 62 55 57   
   65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
   44 57 26 33 34 47 53 34 34 32 24 34 20 27 14 31   
   81 82 83 84 85 86 87 88 89 91 92 94 95 98   
   20 17 17 7 15 8 1 22 2 2 4 1 1 2   
    
  $campaign  
    
   1 2 3 4 5 6 7 8 9 10 11 12 13   
  17642 10570 5341 2651 1599 979 629 400 283 225 177 125 92   
   14 15 16 17 18 19 20 21 22 23 24 25 26   
   69 51 51 58 33 26 30 24 17 16 15 8 8   
   27 28 29 30 31 32 33 34 35 37 39 40 41   
   11 8 10 7 7 4 4 3 5 1 1 2 1   
   42 43 56   
   2 2 1   
    
  $previous  
    
   0 1 2 3 4 5 6 7   
  35563 4561 754 216 70 18 5 1
* # Unique values for discrete numeric variables (integers)  
  unique\_int\_values <- sapply(data[, sapply(data, is.integer)], unique)  
  unique\_int\_values
* $age  
   [1] 56 57 37 40 45 59 41 24 25 29 35 54 46 50 39 30 55 49 34 52 58 32 38 44 42  
  [26] 60 53 47 51 48 33 31 43 36 28 27 26 22 23 20 21 61 19 18 70 66 76 67 73 88  
  [51] 95 77 68 75 63 80 62 65 72 82 64 71 69 78 85 79 83 81 74 17 87 91 86 98 94  
  [76] 84 92 89  
    
  $campaign  
   [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 19 18 23 14 22 25 16 17 15 20 56 39  
  [26] 35 42 28 26 27 32 21 24 29 31 30 41 37 40 33 34 43  
    
  $previous  
  [1] 0 1 2 3 4 5 6 7
* **of non-numeric variables**
* # Frequency of individual categories for non-numeric variables including NA  
  frequency\_data <- lapply(data[, sapply(data, is.character)], function(x) table(x, useNA = "always"))  
  frequency\_data
* $job  
  x  
   admin. blue-collar entrepreneur housemaid management   
   10422 9254 1456 1060 2924   
   retired self-employed services student technician   
   1720 1421 3969 875 6743   
   unemployed <NA>   
   1014 330   
    
  $marital  
  x  
  divorced married single <NA>   
   4612 24928 11568 80   
    
  $education  
  x  
   basic.4y basic.6y basic.9y high.school   
   4176 2292 6045 9515   
   illiterate professional.course university.degree <NA>   
   18 5243 12168 1731   
    
  $default  
  x  
   no yes <NA>   
  32588 3 8597   
    
  $housing  
  x  
   no yes <NA>   
  18622 21576 990   
    
  $loan  
  x  
   no yes <NA>   
  33950 6248 990   
    
  $month  
  x  
   apr aug dec jul jun mar may nov oct sep <NA>   
   2632 6178 182 7174 5318 546 13769 4101 718 570 0   
    
  $day\_of\_week  
  x  
   fri mon thu tue wed <NA>   
  7827 8514 8623 8090 8134 0   
    
  $poutcome  
  x  
   failure nonexistent success <NA>   
   4252 35563 1373 0   
    
  $y  
  x  
   no yes <NA>   
  36548 4640 0
* # Number of unique values for non-numeric variables  
  unique\_data <- sapply(data[, sapply(data, is.character)], function(x) length(unique(x)))  
  unique\_data
* job marital education default housing loan   
   12 4 8 3 3 3   
   month day\_of\_week poutcome y   
   10 5 3 2

1. Explore data: Dig deeper into the data. Query it, visualize it, and identify relationships among the data.
2. Verify data quality: How clean/dirty is the data? Document any quality issues.