Lecture 2: Exploratory Data Analysis

Falco J. Bargagli Stoffi

05/06/2020

Lecture 2: Exploratory Data Analysis

In this lecture we will see how to use vizualization, transformation and modeling to explore your data in a systematic way. This task is usually referred by statisticians as exploratory data analysis, or EDA for short. EDA is an iterative cycle in which you: 1. generate questions about your data; 2. search for answers by vizualizing, transforming and modeling your data; 3. use what you learn to refine your question and/or generate new questions.

During the initial phases of the EDA you should feel totally free to explore and investigate any idea that occurs to you. EDA is a creative process, and the key is to ask a large quantity of questions to your data. Indeed, the ultimate goal of EDA is to develop an understanding of your data and the best way to do it is by using questions as tools to guide you through the investigation.

There is no general rule about which questions you should ask to guide you through the research. However, two types of questions will always be useful for making discoveries within your data: 1. what type of variation occurs within my variables? 2. what type of covariation occurs between my variables?

Before going through the exploratory analysis of your data you need to upload your data in the R environment. You can consider this step as a "pre-processing" phase of the analysis. In the last lecture we saw the main vectors in R (atomic vectors and lists). Here, we will see how to upload and handle matrices of data. The most widely used data matrices in R are data frames and tibbles. A data frame is simply a matrix where each column can be of a different type (i.e., numeric, character, logical). In a data frame}rows correspond to observations while columns correspond to variables.

R datasets

R comes with several built-in datasets, including the famous "iris data" collected by Anderson and analyzed by Fisher in the 1930s. Another widely used dataset for introduction to data anlysis in R is the "cars dataset". You can type "iris" or "cars" to see the dataset.

```
# Upload the "iris" dataset and "assign" it to a dataframe
data <- iris
```

The first thing you can do is to check the dimensions of the dataset and the names of the columns.

```
# Check out the dimension of the data
dim(data)
```

```
## [1] 150   5
```

```
# Check out the names of the columns
ls(data) # alphabetic order
```

```
## [1] "Petal.Length" "Petal.Width"  "Sepal.Length" "Sepal.Width"
## [5] "Species"
```

```r
colnames(data) #dataset order
```

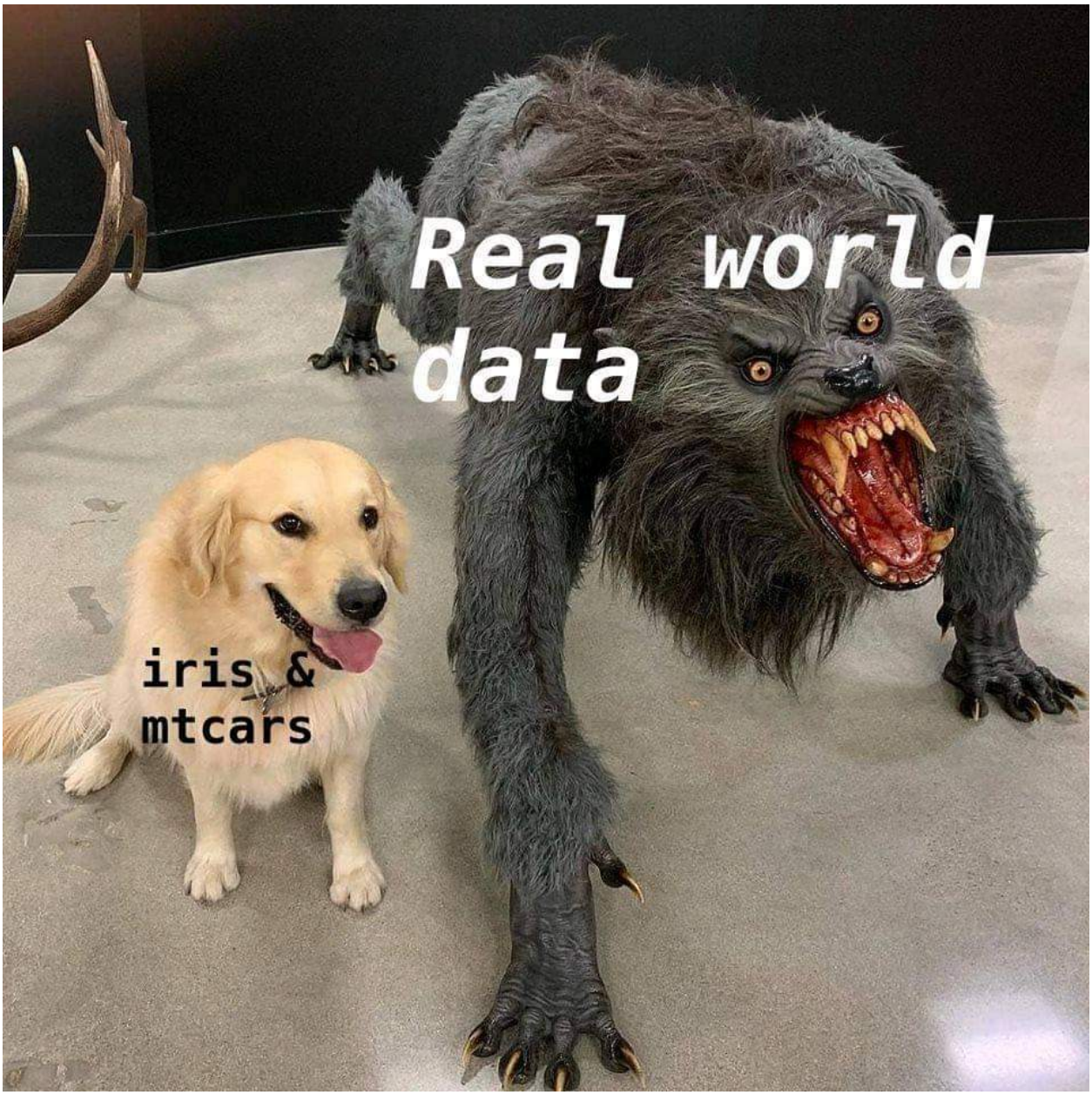
```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"
```

A very useful command is the "summary()" function. "summary()" depicts a series of descriptive statistics for each numeric column (i.e., minimum, maximum, median, mean, 1st and 3rd decile).

```r
summary(data)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

"iris" and "cars" data are used for 101 R programming classes. However, these dataset just contain numeric variables and are really well-behaved data very different from "real world data".



Uploading Data

Before uploading your data, make sure that you set the working directory in the same folder of your data.

```
setwd('...')
```

Uploading Data from Text

R has build-in functions to upload text data, the “read.table()” and “read.csv()” functions. You may want to use the “header = TRUE” (this function tells R that the first row contains the names of the columns of the table), “sep=”;"" (this function tells her that data in your text file are separated by “;”) and “stringsAsFactors

= FALSE" (this function is a logical that indicates whether strings in a data frame should be treated as factor variables or as just plain strings) options depending on the data source you are currently using.

An equivalent version is the "read_csv()" function from the "readr" library.

```
library(readr)
dataset <- read_csv(NULL)
```

Uploading Data from Stata/SAS/SPSS

In order to upload data from Stata, SAS or SPSS you need to install the "haven" library.

```
library(haven)
dataset <- read_sav(NULL) # for SPSS data
dataset <- read_sas(NULL, NULL) # for SAS data
dataset <- read_stata(NULL) # for Stata data
```

Uploading Data from Excel

In order to upload data from Excel you need to install the "readxl" library.

In the following we will upload the Compustat Data that can be found here. Compustat is a database of financial, statistical and market information on active and inactive global companies throughout the world. Here, we will focus just on northern-American enterprises financial account data in the years from 1997 to 2017.

```
library(readxl)
data <- read_excel("G:\\Il mio Drive\\Econometrics Lab\\Data\\Compustat Data.xlsx")
```

To see the data that you just uploaded you can use the "View" function. Beware that, as the size of your dataset increases, it may take some seconds to view your data.

```
View(data)
```

When you upload "complex" data sources the first thing that you should check is the type of each column vector in the dataframe. You can do so by using the "str()" function.

```
str(data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    266663 obs. of  32 variables:
##  $ Global Company Key                                : num  1004 1004 1004 1004 1004 ...
##  $ Data Date                                         : chr  "31/05/1998" "31/05/1999" "31/05/2000" "3
##  $ Data Year - Fiscal                                : num  1997 1998 1999 2000 2001 ...
##  $ Industry Format                                   : chr  "INDL" "INDL" "INDL" "INDL" ...
##  $ Level of Consolidation - Company Annual Descriptor: chr  "C" "C" "C" "C" ...
##  $ Population Source                                 : chr  "D" "D" "D" "D" ...
##  $ Data Format                                       : chr  "STD" "STD" "STD" "STD" ...
##  $ Ticker Symbol                                     : chr  "AIR" "AIR" "AIR" "AIR" ...
##  $ ISO Currency Code                                 : chr  "USD" "USD" "USD" "USD" ...
##  $ Current Assets - Total                            : num  468 508 511 486 437 ...
##  $ Assets - Total                                    : num  671 727 741 702 710 ...
##  $ Average Short-Term Borrowings                     : num  20.7 45.5 94.9 71.9 36.2 ...
##  $ Long-Term Debt Due in One Year                    : num  0.237 0.42 0.429 0.41 2.025 ...
##  $ Debt in Current Liabilities - Total               : num  0.237 0.42 26.314 13.652 42.525 ...
##  $ Long-Term Debt - Total                            : num  178 181 180 180 218 ...
##  $ Earnings Before Interest and Taxes                : num  64.72 77.38 70.66 45.79 4.71 ...
```

```
##  $ Earnings Before Interest                              : num  79 94.4 89 64.4 27.2 ...
##  $ Employees                                             : num  2.7 2.9 2.9 2.5 2.2 2.1 2.3 2.6 3.3 3.9
##  $ Current Liabilities - Total                           : num  149 174 164 125 150 ...
##  $ Liabilities - Total                                   : num  370 401 401 362 400 ...
##  $ Net Income (Loss)                                     : num  35.7 41.7 35.2 18.5 -58.9 ...
##  $ Net Interest Income                                   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Nonperforming Assets - Total                          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ In Process R&D Expense                                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sales/Turnover (Net)                                  : num  782 918 1024 874 639 ...
##  $ Interest Expense - Total (Financial Services)         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Income Taxes - Total                                  : num  15.5 18.11 14.36 1.69 -39.29 ...
##  $ Active/Inactive Status Marker                         : chr  "A" "A" "A" "A" ...
##  $ Research Co Reason for Deletion                       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ GIC Groups                                            : num  2010 2010 2010 2010 2010 2010 2010 2010 1
##  $ GIC Sectors                                           : num  20 20 20 20 20 20 20 20 20 20 ...
##  $ Standard Industry Classification Code                 : num  5080 5080 5080 5080 5080 5080 5080 5080 5
```

As you can see, these data are a collection of "character" (chr) and "numeric" (num) vectors. "str()" reports also the dimension of the dataset: 266663 observations and 32 variables; and the classes of the dataset: "tibble" (tbl_df, tbl) and "data frame" (data.frame).

What happens if we run the "summary()" function on non-numeric data? "R" just tells you that those columns contain "character} vectors.

Moreover, "summary()" gives you the number of "NA's" for each numeric column.

```
summary(data)
```

```
##  Global Company Key  Data Date         Data Year - Fiscal
##  Min.   :  1004     Length:266663      Min.   :1997
##  1st Qu.: 17673     Class :character   1st Qu.:2001
##  Median : 62110     Mode  :character   Median :2007
##  Mean   : 79169                        Mean   :2007
##  3rd Qu.:145786                        3rd Qu.:2012
##  Max.   :330227                        Max.   :2017
##                                        NA's   :844
##  Industry Format    Level of Consolidation - Company Annual Descriptor
##  Length:266663      Length:266663
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
##  Population Source  Data Format        Ticker Symbol
##  Length:266663      Length:266663      Length:266663
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  ISO Currency Code  Current Assets - Total Assets - Total
##  Length:266663      Min.   :    -7.76      Min.   :      0
##  Class :character   1st Qu.:     8.79      1st Qu.:     38
##  Mode  :character   Median :    58.86      Median :    309
```

```
##                    Mean   :   917.93      Mean   :  12063
##                    3rd Qu.:   317.01      3rd Qu.:   1731
##                    Max.   :192486.65      Max.   :3771200
##                    NA's   :98138          NA's   :39030
##  Average Short-Term Borrowings Long-Term Debt Due in One Year
##  Min.   :     0.0              Min.   :     0.0
##  1st Qu.:     0.0              1st Qu.:     0.0
##  Median :     0.0              Median :     0.4
##  Mean   :   198.2              Mean   :   302.1
##  3rd Qu.:     0.0              3rd Qu.:    11.1
##  Max.   :335926.0              Max.   :496570.1
##  NA's   :245632                NA's   :51973
##  Debt in Current Liabilities - Total Long-Term Debt - Total
##  Min.   : -2267.1                    Min.   :      0
##  1st Qu.:     0.0                    1st Qu.:      0
##  Median :     2.5                    Median :     16
##  Mean   :  1219.2                    Mean   :   2066
##  3rd Qu.:    36.0                    3rd Qu.:    300
##  Max.   :605462.5                    Max.   :3296298
##  NA's   :40727                       NA's   :39524
##  Earnings Before Interest and Taxes Earnings Before Interest
##  Min.   :-80053.00                  Min.   :-76735.00
##  1st Qu.:    -1.97                  1st Qu.:    -0.86
##  Median :     6.17                  Median :    11.12
##  Mean   :   329.49                  Mean   :   468.21
##  3rd Qu.:    78.07                  3rd Qu.:   115.87
##  Max.   :130622.00                  Max.   :130622.00
##  NA's   :67546                      NA's   :70945
##    Employees       Current Liabilities - Total Liabilities - Total
##  Min.   :   0.00   Min.   :     0.0            Min.   :      0
##  1st Qu.:   0.09   1st Qu.:     4.4            1st Qu.:     12
##  Median :   0.48   Median :    27.6            Median :    161
##  Mean   :   7.98   Mean   :   730.4            Mean   :  10409
##  3rd Qu.:   3.12   3rd Qu.:   181.2            3rd Qu.:   1135
##  Max.   :2545.21   Max.   :329795.0            Max.   :3589783
##  NA's   :76862     NA's   :97225               NA's   :39377
##  Net Income (Loss)   Net Interest Income Nonperforming Assets - Total
##  Min.   :-99289.00   Min.   :-3546.00    Min.   :     0.0
##  1st Qu.:    -4.68   1st Qu.:    8.21    1st Qu.:     0.3
##  Median :     1.59   Median :   25.23    Median :     4.0
##  Mean   :   140.94   Mean   :  656.08    Mean   :   426.5
##  3rd Qu.:    34.68   3rd Qu.:   92.98    3rd Qu.:    21.2
##  Max.   :104821.00   Max.   :54652.00    Max.   :316713.0
##  NA's   :65728       NA's   :245224      NA's   :246675
##  In Process R&D Expense Sales/Turnover (Net)
##  Min.   :-6831.27       Min.   :-15009.3
##  1st Qu.:    0.00       1st Qu.:    11.7
##  Median :    0.00       Median :   101.1
##  Mean   :   -1.35       Mean   :  2394.4
##  3rd Qu.:    0.00       3rd Qu.:   738.9
##  Max.   :   11.00       Max.   :496785.0
##  NA's   :117161         NA's   :65747
##  Interest Expense - Total (Financial Services) Income Taxes - Total
##  Min.   :    0.00                              Min.   :-45415.00
```

```
##  1st Qu.:    6.17                                1st Qu.:     0.00
##  Median :   15.56                                Median :     0.66
##  Mean   :  617.64                                Mean   :    70.88
##  3rd Qu.:   53.06                                3rd Qu.:    13.54
##  Max.   :85948.88                                Max.   : 37162.00
##  NA's   :250704                                  NA's   :40255
##  Active/Inactive Status Marker Research Co Reason for Deletion
##  Length:266663                 Min.   : 1.0
##  Class :character              1st Qu.: 1.0
##  Mode  :character              Median : 1.0
##                                Mean   : 3.8
##                                3rd Qu.: 7.0
##                                Max.   :20.0
##                                NA's   :150836
##    GIC Groups     GIC Sectors    Standard Industry Classification Code
##  Min.   :1010    Min.   :10.00   Min.   : 100
##  1st Qu.:2030    1st Qu.:20.00   1st Qu.:3569
##  Median :3520    Median :35.00   Median :6020
##  Mean   :3326    Mean   :33.09   Mean   :5084
##  3rd Qu.:4040    3rd Qu.:40.00   3rd Qu.:6722
##  Max.   :6010    Max.   :60.00   Max.   :9997
##  NA's   :31855   NA's   :31855
```

A good way to start exploring your data is by checking if there is something "strange". As you can't go through all the observation, a good rule of thumbs is to check the "head" and "tail" of your data. The "head" and "tail" commands provide a good way to explore the first 6 and the last 6 observations in your data.

```
head(data)
```

```
## # A tibble: 6 x 32
##   `Global Company... `Data Date` `Data Year - Fi... `Industry Forma...
##                <dbl> <chr>                   <dbl> <chr>
## 1              1004 31/05/1998               1997 INDL
## 2              1004 31/05/1999               1998 INDL
## 3              1004 31/05/2000               1999 INDL
## 4              1004 31/05/2001               2000 INDL
## 5              1004 31/05/2002               2001 INDL
## 6              1004 31/05/2003               2002 INDL
## # ... with 28 more variables: `Level of Consolidation - Company Annual
## #   Descriptor` <chr>, `Population Source` <chr>, `Data Format` <chr>,
## #   `Ticker Symbol` <chr>, `ISO Currency Code` <chr>, `Current Assets -
## #   Total` <dbl>, `Assets - Total` <dbl>, `Average Short-Term
## #   Borrowings` <dbl>, `Long-Term Debt Due in One Year` <dbl>, `Debt in
## #   Current Liabilities - Total` <dbl>, `Long-Term Debt - Total` <dbl>,
## #   `Earnings Before Interest and Taxes` <dbl>, `Earnings Before
## #   Interest` <dbl>, Employees <dbl>, `Current Liabilities - Total` <dbl>,
## #   `Liabilities - Total` <dbl>, `Net Income (Loss)` <dbl>, `Net Interest
## #   Income` <dbl>, `Nonperforming Assets - Total` <dbl>, `In Process R&D
## #   Expense` <dbl>, `Sales/Turnover (Net)` <dbl>, `Interest Expense -
## #   Total (Financial Services)` <dbl>, `Income Taxes - Total` <dbl>,
## #   `Active/Inactive Status Marker` <chr>, `Research Co Reason for
## #   Deletion` <dbl>, `GIC Groups` <dbl>, `GIC Sectors` <dbl>, `Standard
## #   Industry Classification Code` <dbl>
```

```
tail(data)
```

```
## # A tibble: 6 x 32
##   `Global Company... `Data Date` `Data Year - Fi... `Industry Forma...
##                <dbl> <chr>                   <dbl> <chr>
## 1           328795 31/12/2013              2013 INDL
## 2           328795 31/12/2014              2014 INDL
## 3           328795 31/12/2015              2015 INDL
## 4           328795 31/12/2016              2016 INDL
## 5           328795 31/12/2017              2017 INDL
## 6           330227 30/09/2017              2017 INDL
## # ... with 28 more variables: `Level of Consolidation - Company Annual
## #   Descriptor` <chr>, `Population Source` <chr>, `Data Format` <chr>,
## #   `Ticker Symbol` <chr>, `ISO Currency Code` <chr>, `Current Assets -
## #   Total` <dbl>, `Assets - Total` <dbl>, `Average Short-Term
## #   Borrowings` <dbl>, `Long-Term Debt Due in One Year` <dbl>, `Debt in
## #   Current Liabilities - Total` <dbl>, `Long-Term Debt - Total` <dbl>,
## #   `Earnings Before Interest and Taxes` <dbl>, `Earnings Before
## #   Interest` <dbl>, Employees <dbl>, `Current Liabilities - Total` <dbl>,
## #   `Liabilities - Total` <dbl>, `Net Income (Loss)` <dbl>, `Net Interest
## #   Income` <dbl>, `Nonperforming Assets - Total` <dbl>, `In Process R&D
## #   Expense` <dbl>, `Sales/Turnover (Net)` <dbl>, `Interest Expense -
## #   Total (Financial Services)` <dbl>, `Income Taxes - Total` <dbl>,
## #   `Active/Inactive Status Marker` <chr>, `Research Co Reason for
## #   Deletion` <dbl>, `GIC Groups` <dbl>, `GIC Sectors` <dbl>, `Standard
## #   Industry Classification Code` <dbl>
```

Subset your Data

You can subset your data in two ways: 1. get a subset of columns (variables); 2. get a subset of rows (observations).

In order to extract from your dataset a single column you can proceed in three ways: (i) extract the column by recalling its position, (ii-iii) extracting the column by its name. Keep in mind that R by defauls shows you the first 1000 observations of the selected column.

```
colnames(data)
```

```
##  [1] "Global Company Key"
##  [2] "Data Date"
##  [3] "Data Year - Fiscal"
##  [4] "Industry Format"
##  [5] "Level of Consolidation - Company Annual Descriptor"
##  [6] "Population Source"
##  [7] "Data Format"
##  [8] "Ticker Symbol"
##  [9] "ISO Currency Code"
## [10] "Current Assets - Total"
## [11] "Assets - Total"
## [12] "Average Short-Term Borrowings"
## [13] "Long-Term Debt Due in One Year"
## [14] "Debt in Current Liabilities - Total"
## [15] "Long-Term Debt - Total"
## [16] "Earnings Before Interest and Taxes"
## [17] "Earnings Before Interest"
## [18] "Employees"
```

```
## [19] "Current Liabilities - Total"
## [20] "Liabilities - Total"
## [21] "Net Income (Loss)"
## [22] "Net Interest Income"
## [23] "Nonperforming Assets - Total"
## [24] "In Process R&D Expense"
## [25] "Sales/Turnover (Net)"
## [26] "Interest Expense - Total (Financial Services)"
## [27] "Income Taxes - Total"
## [28] "Active/Inactive Status Marker"
## [29] "Research Co Reason for Deletion"
## [30] "GIC Groups"
## [31] "GIC Sectors"
## [32] "Standard Industry Classification Code"
```

```
head(data[,15])
```

```
## # A tibble: 6 x 1
##   `Long-Term Debt - Total`
##                      <dbl>
## 1                     178.
## 2                     181.
## 3                     180.
## 4                     180.
## 5                     218.
## 6                     165.
```

```
head(data$`Long-Term Debt - Total`)
```

```
## [1] 177.509 180.939 180.447 179.987 217.699 164.658
```

```
head(data[,c("Long-Term Debt - Total")]) #approximation
```

```
## # A tibble: 6 x 1
##   `Long-Term Debt - Total`
##                      <dbl>
## 1                     178.
## 2                     181.
## 3                     180.
## 4                     180.
## 5                     218.
## 6                     165.
```

To subset a number of observation select them as follows:

```
data[1:10,]
```

```
## # A tibble: 10 x 32
##    `Global Company... `Data Date` `Data Year - Fi... `Industry Forma...
##                 <dbl> <chr>                   <dbl> <chr>
##  1              1004 31/05/1998               1997 INDL
##  2              1004 31/05/1999               1998 INDL
##  3              1004 31/05/2000               1999 INDL
##  4              1004 31/05/2001               2000 INDL
##  5              1004 31/05/2002               2001 INDL
##  6              1004 31/05/2003               2002 INDL
##  7              1004 31/05/2004               2003 INDL
```

```
##  8             1004 31/05/2005              2004 INDL
##  9             1004 31/05/2006              2005 INDL
## 10             1004 31/05/2007              2006 INDL
## # ... with 28 more variables: `Level of Consolidation - Company Annual
## #   Descriptor` <chr>, `Population Source` <chr>, `Data Format` <chr>,
## #   `Ticker Symbol` <chr>, `ISO Currency Code` <chr>, `Current Assets -
## #   Total` <dbl>, `Assets - Total` <dbl>, `Average Short-Term
## #   Borrowings` <dbl>, `Long-Term Debt Due in One Year` <dbl>, `Debt in
## #   Current Liabilities - Total` <dbl>, `Long-Term Debt - Total` <dbl>,
## #   `Earnings Before Interest and Taxes` <dbl>, `Earnings Before
## #   Interest` <dbl>, Employees <dbl>, `Current Liabilities - Total` <dbl>,
## #   `Liabilities - Total` <dbl>, `Net Income (Loss)` <dbl>, `Net Interest
## #   Income` <dbl>, `Nonperforming Assets - Total` <dbl>, `In Process R&D
## #   Expense` <dbl>, `Sales/Turnover (Net)` <dbl>, `Interest Expense -
## #   Total (Financial Services)` <dbl>, `Income Taxes - Total` <dbl>,
## #   `Active/Inactive Status Marker` <chr>, `Research Co Reason for
## #   Deletion` <dbl>, `GIC Groups` <dbl>, `GIC Sectors` <dbl>, `Standard
## #   Industry Classification Code` <dbl>
```

While to get a random sample of observation you can use the "sample function". This will be very useful when we will discuss machine learning algorithms as you will usually devide your dataset in a training set (on which you will build your machine learning algorithm) and a test set (on which you will test your algorithm).

```
sample_of_observations <- sample(seq_len(nrow(data)), size = nrow(data)*0.1)
head(data[sample_of_observations,]) #10 percent of obs
```

```
## # A tibble: 6 x 32
##   `Global Company... `Data Date` `Data Year - Fi... `Industry Forma...
##              <dbl> <chr>                    <dbl> <chr>
## 1            13579 31/12/2002                2002 FS
## 2             5442 31/12/1999                1999 INDL
## 3            11985 31/12/1998                1998 INDL
## 4            24857 31/12/1997                1997 INDL
## 5            61899 31/12/2016                2016 INDL
## 6            64279 31/12/2002                2002 INDL
## # ... with 28 more variables: `Level of Consolidation - Company Annual
## #   Descriptor` <chr>, `Population Source` <chr>, `Data Format` <chr>,
## #   `Ticker Symbol` <chr>, `ISO Currency Code` <chr>, `Current Assets -
## #   Total` <dbl>, `Assets - Total` <dbl>, `Average Short-Term
## #   Borrowings` <dbl>, `Long-Term Debt Due in One Year` <dbl>, `Debt in
## #   Current Liabilities - Total` <dbl>, `Long-Term Debt - Total` <dbl>,
## #   `Earnings Before Interest and Taxes` <dbl>, `Earnings Before
## #   Interest` <dbl>, Employees <dbl>, `Current Liabilities - Total` <dbl>,
## #   `Liabilities - Total` <dbl>, `Net Income (Loss)` <dbl>, `Net Interest
## #   Income` <dbl>, `Nonperforming Assets - Total` <dbl>, `In Process R&D
## #   Expense` <dbl>, `Sales/Turnover (Net)` <dbl>, `Interest Expense -
## #   Total (Financial Services)` <dbl>, `Income Taxes - Total` <dbl>,
## #   `Active/Inactive Status Marker` <chr>, `Research Co Reason for
## #   Deletion` <dbl>, `GIC Groups` <dbl>, `GIC Sectors` <dbl>, `Standard
## #   Industry Classification Code` <dbl>
```

```
head(data[-sample_of_observations,]) #90 percent of obs
```

```
## # A tibble: 6 x 32
##   `Global Company... `Data Date` `Data Year - Fi... `Industry Forma...
##              <dbl> <chr>                    <dbl> <chr>
```

```
## 1             1004 31/05/1998             1997 INDL
## 2             1004 31/05/1999             1998 INDL
## 3             1004 31/05/2000             1999 INDL
## 4             1004 31/05/2001             2000 INDL
## 5             1004 31/05/2002             2001 INDL
## 6             1004 31/05/2003             2002 INDL
## # ... with 28 more variables: `Level of Consolidation - Company Annual
## #   Descriptor` <chr>, `Population Source` <chr>, `Data Format` <chr>,
## #   `Ticker Symbol` <chr>, `ISO Currency Code` <chr>, `Current Assets -
## #   Total` <dbl>, `Assets - Total` <dbl>, `Average Short-Term
## #   Borrowings` <dbl>, `Long-Term Debt Due in One Year` <dbl>, `Debt in
## #   Current Liabilities - Total` <dbl>, `Long-Term Debt - Total` <dbl>,
## #   `Earnings Before Interest and Taxes` <dbl>, `Earnings Before
## #   Interest` <dbl>, Employees <dbl>, `Current Liabilities - Total` <dbl>,
## #   `Liabilities - Total` <dbl>, `Net Income (Loss)` <dbl>, `Net Interest
## #   Income` <dbl>, `Nonperforming Assets - Total` <dbl>, `In Process R&D
## #   Expense` <dbl>, `Sales/Turnover (Net)` <dbl>, `Interest Expense -
## #   Total (Financial Services)` <dbl>, `Income Taxes - Total` <dbl>,
## #   `Active/Inactive Status Marker` <chr>, `Research Co Reason for
## #   Deletion` <dbl>, `GIC Groups` <dbl>, `GIC Sectors` <dbl>, `Standard
## #   Industry Classification Code` <dbl>
```

Moreover, you may want to subset your data based on a certain value of a variable.

```
sub_data <-subset(data, !is.na(data$Employees))
```

Missing Data

Before starting your analysis it is central to check how the missing values are distributed. There are three categories of missing data: 1. missing completely at random (MCAR); 2. missing at random (MAR); 3. missing not at random (MNAR).

Missing Completely at Random, MCAR, means there is no relationship between the missingness of the data and any values, observed or missing. Those missing data points are a random subset of the data. There is nothing systematic going on that makes some data more likely to be missing than others.

Missing at Random, MAR, means there is a systematic relationship between the propensity of missing values and the observed data. Whether an observation is missing has nothing to do with the missing values, but it does have to do with the values of an individual's observed variables. So, for example, if men are more likely to tell you their weight than women, weight is MAR.

Missing Not at Random, MNAR, means there is a relationship between the propensity of a value to be missing and its values. This is a case where the people with the lowest education are missing on education or the sickest people are most likely to drop out of the study.

MNAR is called "non-ignorable" because the missing data mechanism itself has to be modeled as you deal with the missing data. You have to include some model for why the data are missing and what the likely values are.

"Missing Completely at Random" and "Missing at Random" are both considered 'ignorable' because we don't have to include any information about the missing data itself when we deal with the missing data.

Each of these possible scenarios requires a different way to be handled. For instance, multiple imputation assumes the data are at least missing at random. So the important distinction here is whether the data are MAR as opposed to MNAR. Listwise deletion, however, requires the data are MCAR in order to not introduce bias in the results.

1. MCAR vs. MAR and MNAR

There is a very useful test for MCAR, Little's test. Using Little's test we can tests the null hypothesis that the missing data is MCAR (source). A p.value of less than 0.05 is usually interpreted as being that the missing data is not MCAR (i.e., is either Missing At Random or MNAR).

```
BaylorEdPsych::LittleMCAR(data)
```

But like all tests of assumptions, it's not definitive. So run it, but use it as only one piece of information.

A second technique is to create dummy variables for whether a variable is missing.

$$k = \begin{cases} 1 = missing; \\ 0 = observed. \end{cases} \tag{1}$$

You can then run t-tests and chi-square tests between this variable and other variables in the data set to see if the missingness on this variable is related to the values of other variables.

For example, if women really are less likely to tell you their weight than men, a chi-square test will tell you that the percentage of missing data on the weight variable is higher for women than men.

2. MAR vs. MNAR

The only true way to distinguish between MNAR and MAR is to measure some of that missing data. It's a common practice among professional surveyors to, for example, follow-up on a paper survey with phone calls to a group of the non-respondents and ask a few key survey items. This allows you to compare respondents to non-respondents.

If their responses on those key items differ by very much, that's good evidence that the data are MNAR.

However in most missing data situations, we don't have the luxury of getting a hold of the missing data. So while we can't test it directly, we can examine patterns in the data get an idea of what's the most likely mechanism (source).

The Amelia package, developed by Harvard's professor Gary King and his colleagues, is probably the best tool to perform such an analysis.

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.6.2
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

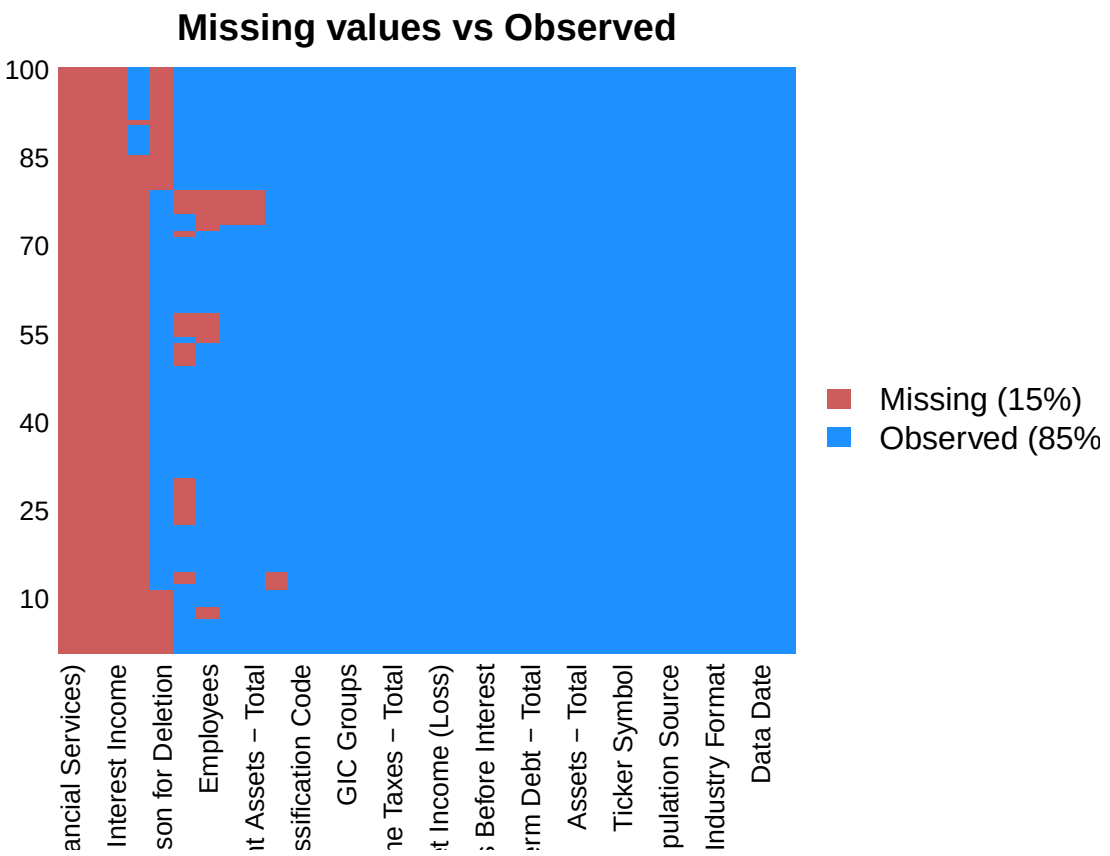
```
missmap(data[1:100,], main = "Missing values vs Observed")
```

```
## Warning in if (class(obj) == "amelia") {: la condizione la lunghezza > 1 e
## solo il promo elemento verrà utilizzato
```

```
## Warning: Unknown or uninitialised column: 'arguments'.
```

```
## Warning: Unknown or uninitialised column: 'arguments'.
```

```
## Warning: Unknown or uninitialised column: 'imputations'.
```



A good idea is not just to check the first observations, but to check a random sample of observations:

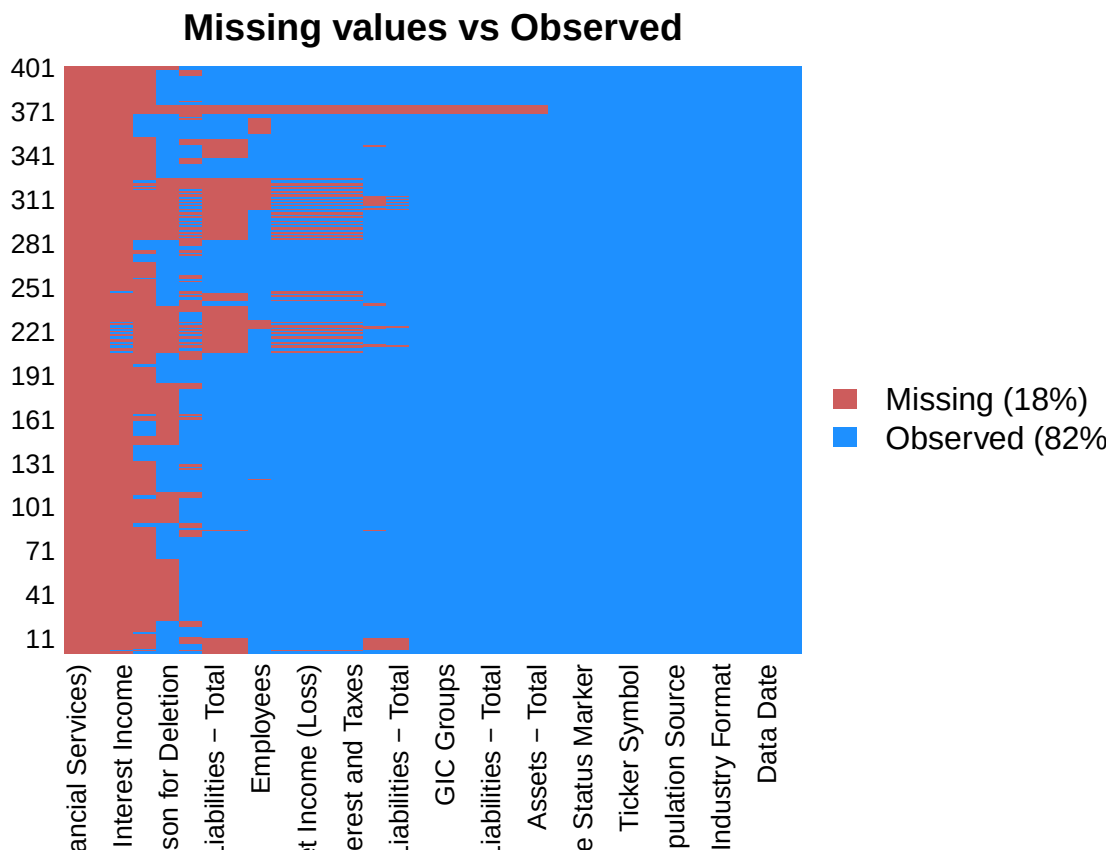
```
missmap(data[1798:2198,], main = "Missing values vs Observed")
```

```
## Warning in if (class(obj) == "amelia") {: la condizione la lunghezza > 1 e
## solo il promo elemento verrà utilizzato
```

```
## Warning: Unknown or uninitialised column: 'arguments'.
```

```
## Warning: Unknown or uninitialised column: 'arguments'.
```

```
## Warning: Unknown or uninitialised column: 'imputations'.
```



Three variables are found to have a high missing values rates (more than 90% of missing values).

```
summary(data$`Net Interest Income`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -3546.00     8.21    25.23   656.08    92.98 54652.00   245224
```

```
length(which(!is.na(data$`Net Interest Income`)))/nrow(data)
```

```
## [1] 0.08039736
```

```
summary(data$`Nonperforming Assets - Total`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##      0.0      0.3      4.0    426.5     21.2 316713.0   246675
```

```
length(which(!is.na(data$`Nonperforming Assets - Total`)))/nrow(data)
```

```
## [1] 0.07495603
```

```
summary(data$`Interest Expense - Total (Financial Services)`)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##     0.00     6.17    15.56   617.64    53.06 85948.88   250704
```

```
length(which(!is.na(data$`Interest Expense - Total (Financial Services)`)))/nrow(data)
```

```
## [1] 0.05984707
```

A way to deal with highly missing variables is to delete them from your data (using a subsetting option).

```r
dim(data)
```

```
## [1] 266663     32
```

```r
data <- data[, !names(data) %in% c("Interest Expense - Total (Financial Services)", "Net Interest Incom
dim(data)
```

```
## [1] 266663     29
```

Once we excluded the three variables with a high number of missing values (probably MNAR) we can remove the missing values from the dataset by making the explicit assumtion of MAR. This assumption is quite strong and "you do not want to do it in your real data analysis". You can try some different imputation methods in the packages "amelia" and "mice".

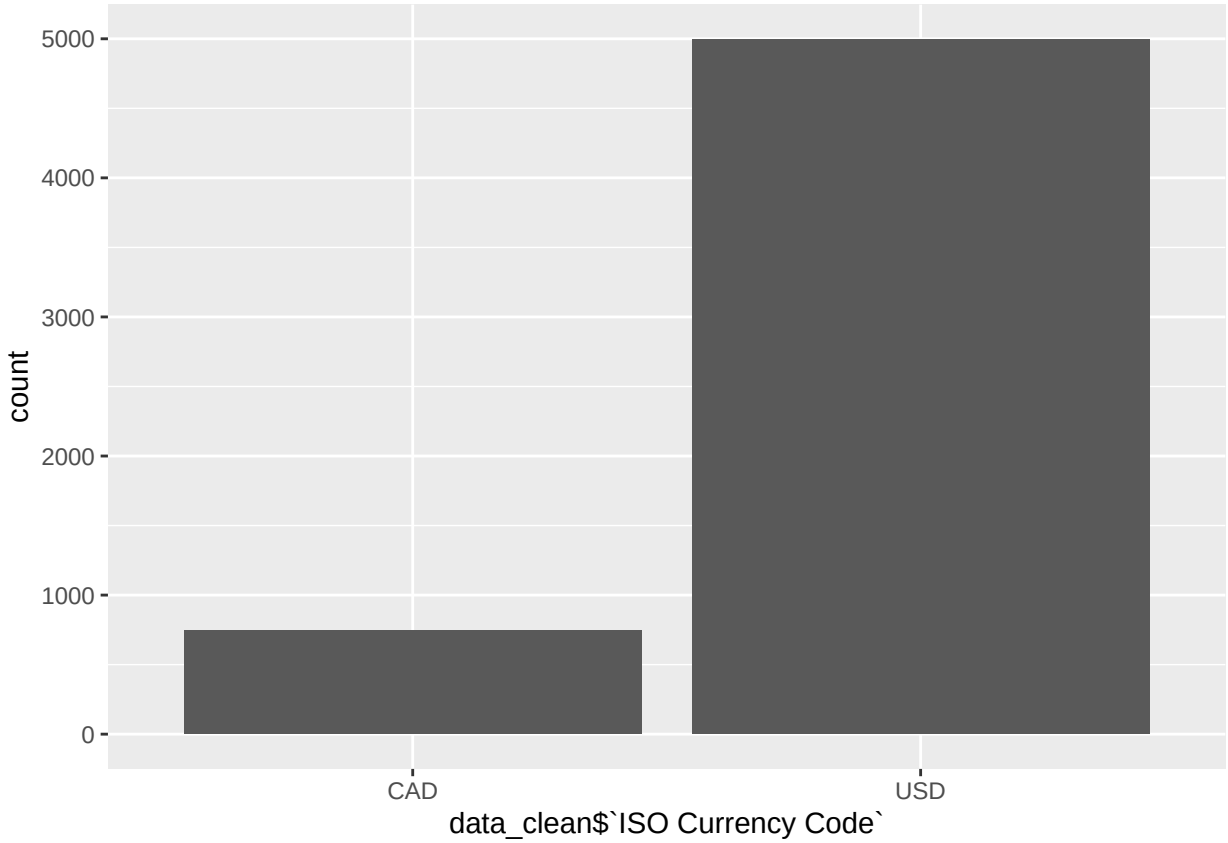
```r
data_clean <- na.omit(data)
```

Plotting

A good way to start exploring variation in your data is to visualize the distribution of your variables' values. R comes with some implemented functions for plotting (i.e., "plot", "hist", etc). Another good alternative for better looking plots is to use the functions implemented in the package "ggplot2". Here we will see how to make basic plots in both ways.

How to visualize the distribution of a variable will depend on whether the variable is categorical or continuous. A variable is categorical if it can only take a small set of values. In R, categorical variables are usually saved as character vectors. You can use a bar chart to examine their distribution.

```r
library(ggplot2)
ggplot(data = data_clean) +
  geom_bar(mapping = aes(x = data_clean$`ISO Currency Code`))
```



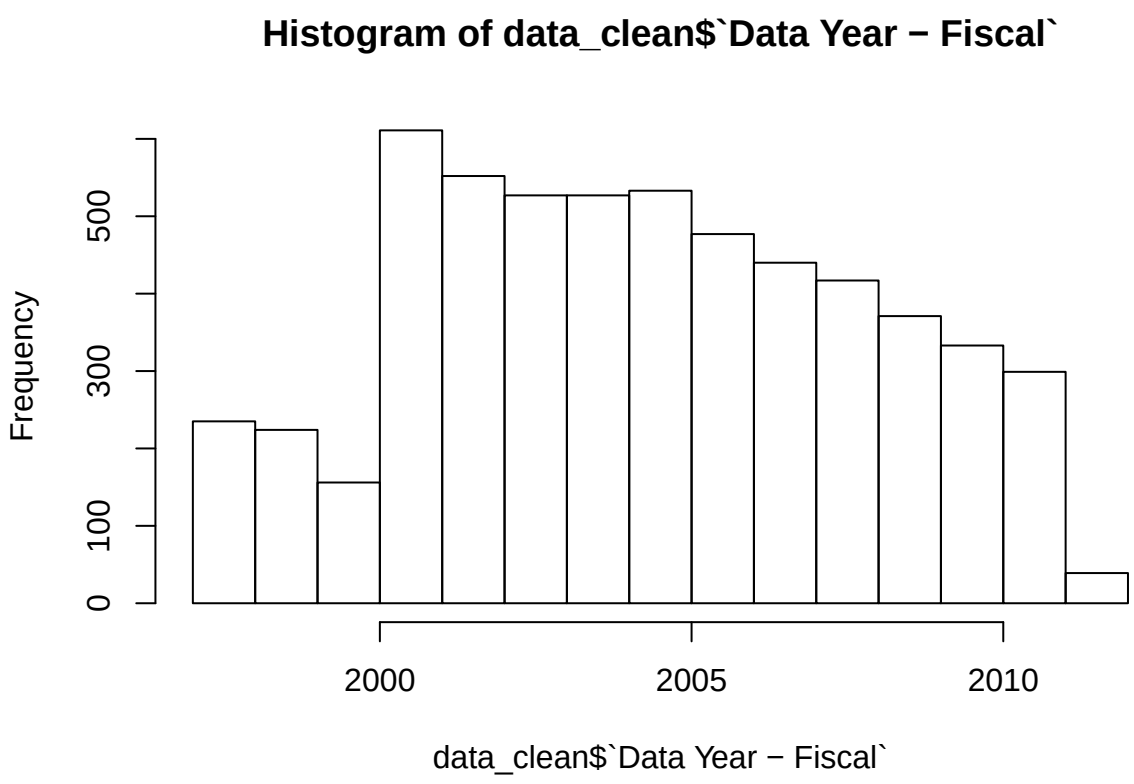
The height of the bars displays how many observations occurred for each value of x. This can be done with the function "table".

```
table(data_clean$`ISO Currency Code`)
```

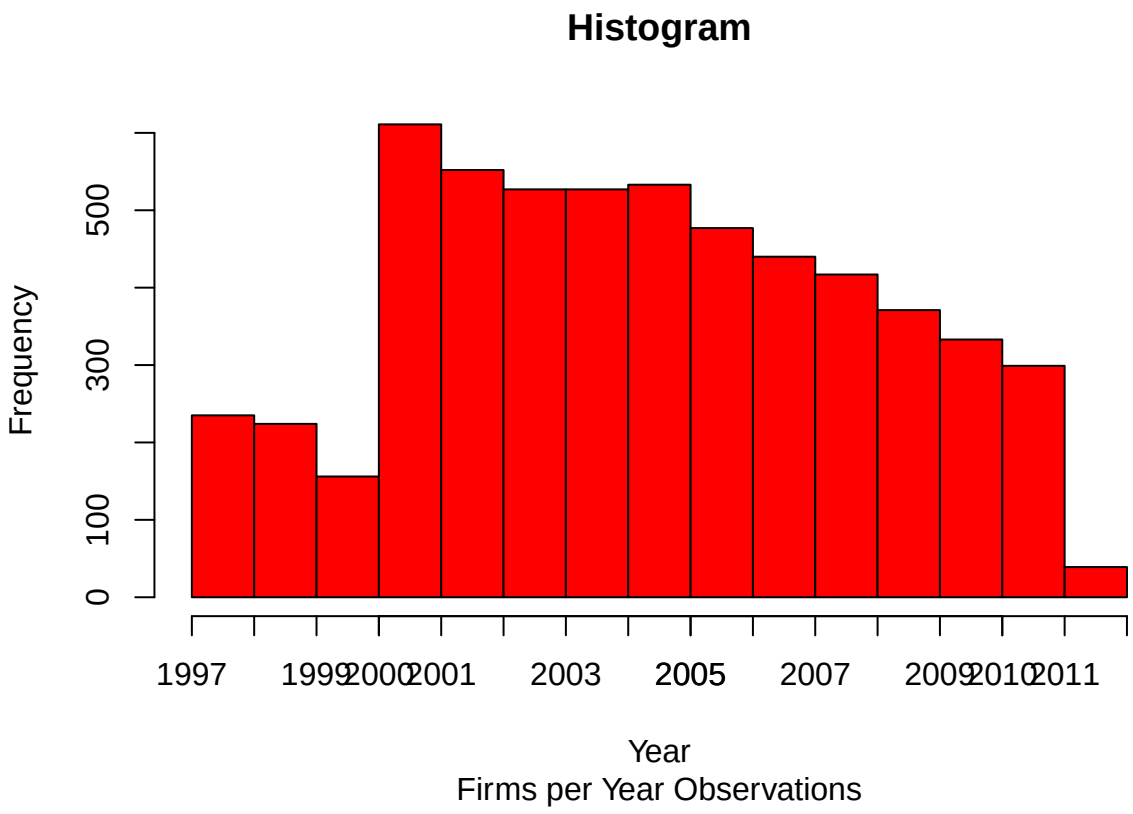
```
##
##  CAD  USD
##  745 4996
```

A variable is continuous if it can take any of a large set of ordered values. You can examine the distribution of a contnuous variable by using a histogram.

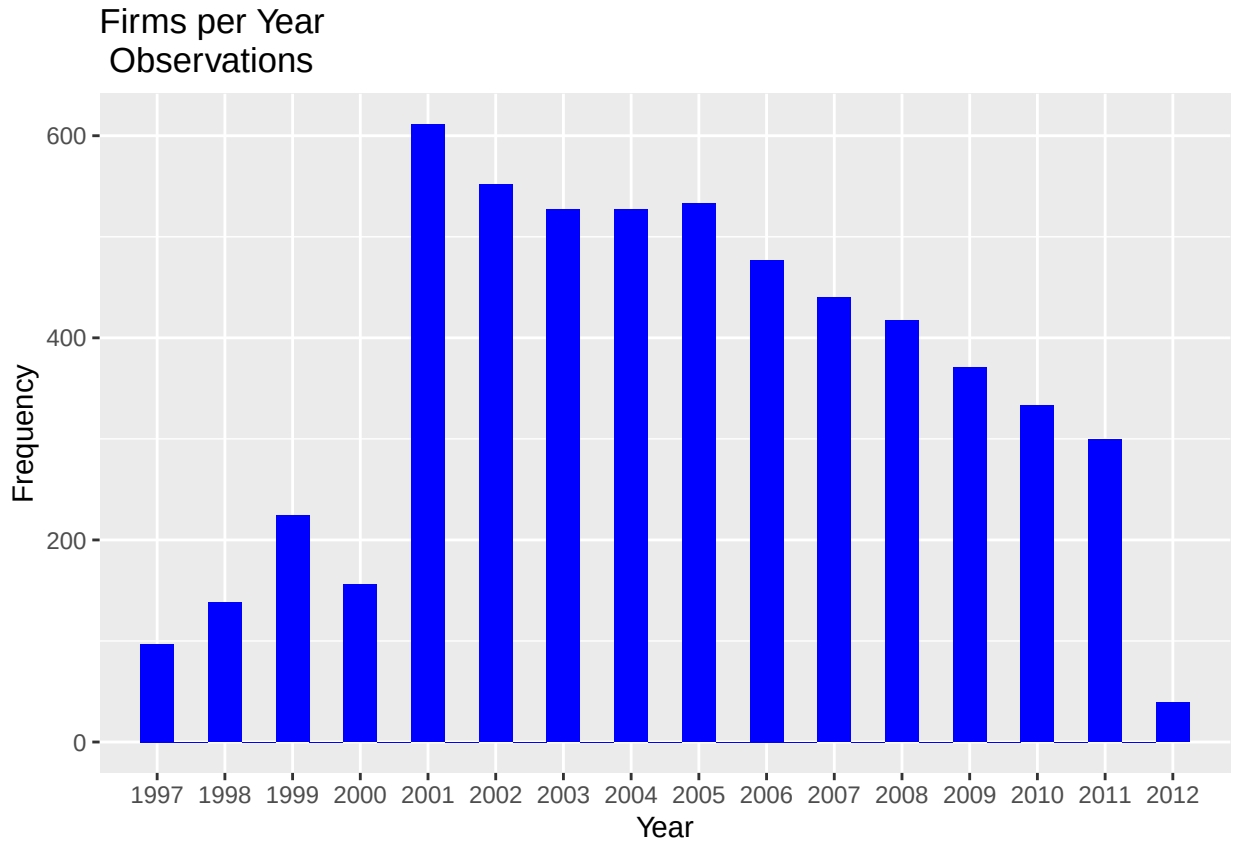
```
# Lets First Compare the Obs in Data v. Data_clean
# Histogram
hist(data_clean$`Data Year - Fiscal`)
```




```
hist(data_clean$`Data Year - Fiscal`,
     col=2,
     main = "Histogram",
     sub = "Firms per Year Observations",
     xlab = "Year",
     ylab = "Frequency")
axis(1, at=1997:2012, labels=c(seq(1997,2012,1)))
```




```
ggplot(data = data_clean) +
  geom_histogram(mapping = aes(x = data_clean$`Data Year - Fiscal`),
                 binwidth = 0.5,
                 fill="blue") +
  scale_x_discrete(name ="Year", limits=c(seq(1997, 2012, 1))) +
  scale_y_continuous(name = "Frequency") +
  ggtitle("Firms per Year \n Observations") #\n to split long title into multiple lines.
```

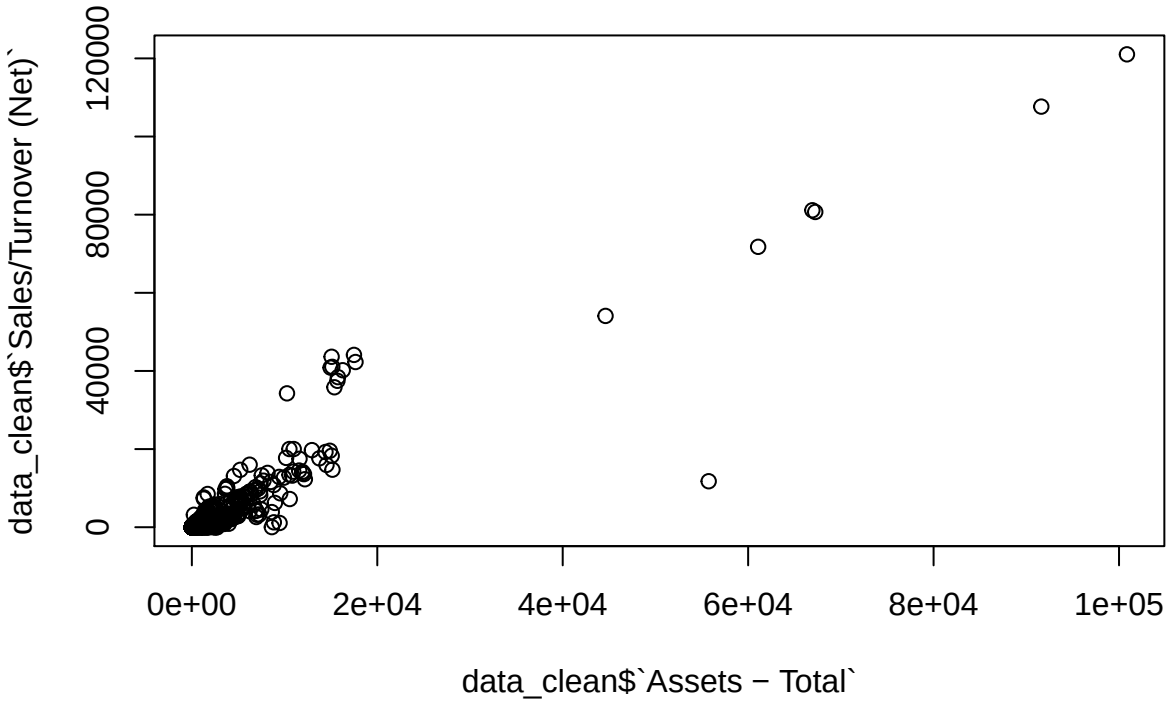



```
table(data_clean$`Data Year - Fiscal`)
```

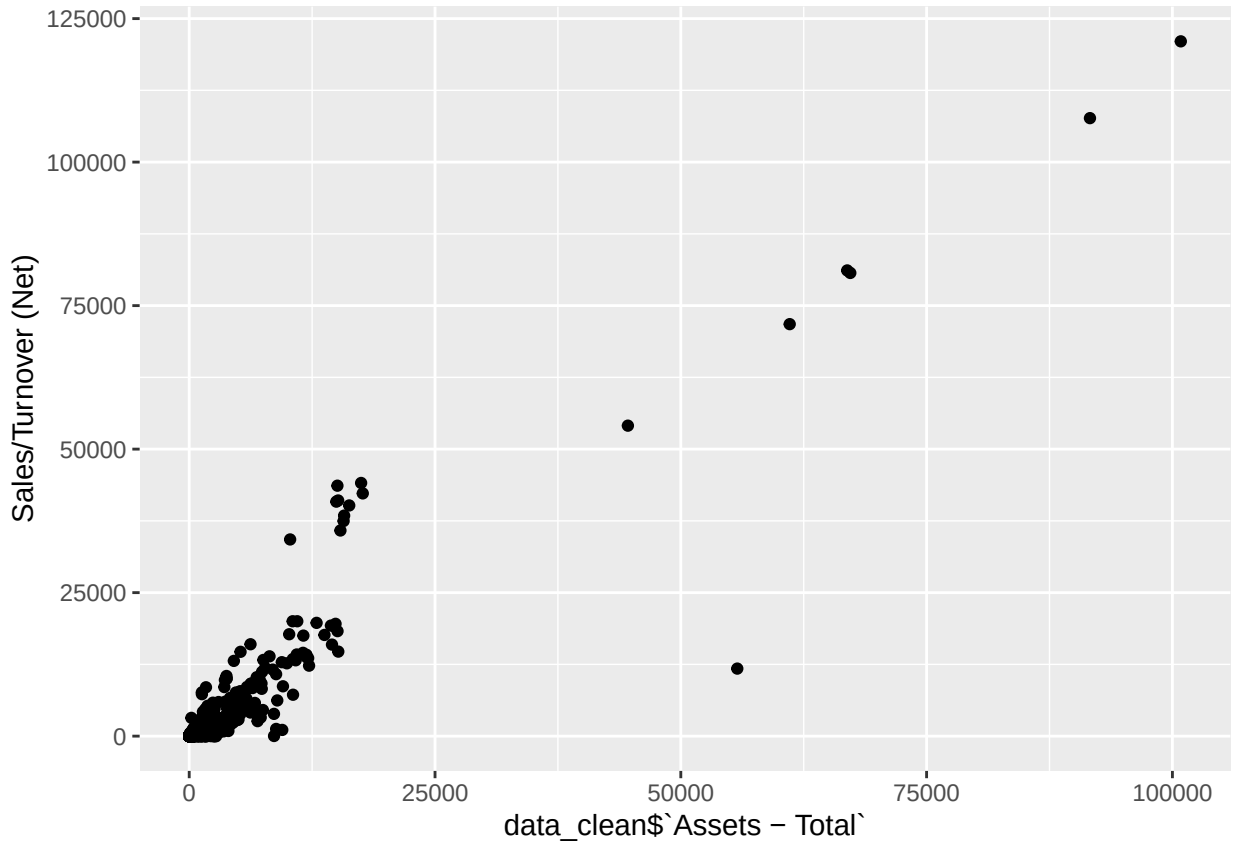
```
##
## 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
##   97  138  224  156  611  552  527  527  533  477  440  417  371  333  299
## 2012
##   39
```

To check how two variable are covarying you can use a scatter plot (or two-way plot).

```
plot(data_clean$`Assets - Total`, data_clean$`Sales/Turnover (Net)`)
```

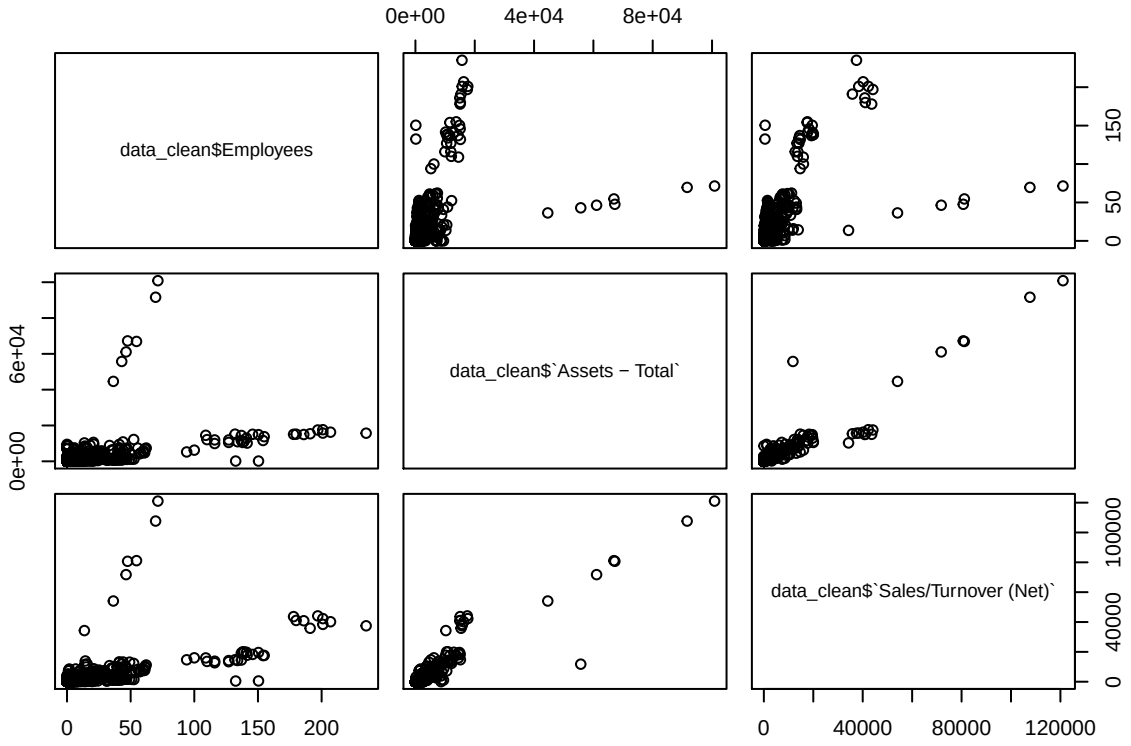



```
ggplot(data = data_clean,
       mapping = aes(x = data_clean$`Assets - Total`,
                     y = `Sales/Turnover (Net)`)) +
  geom_point()
```



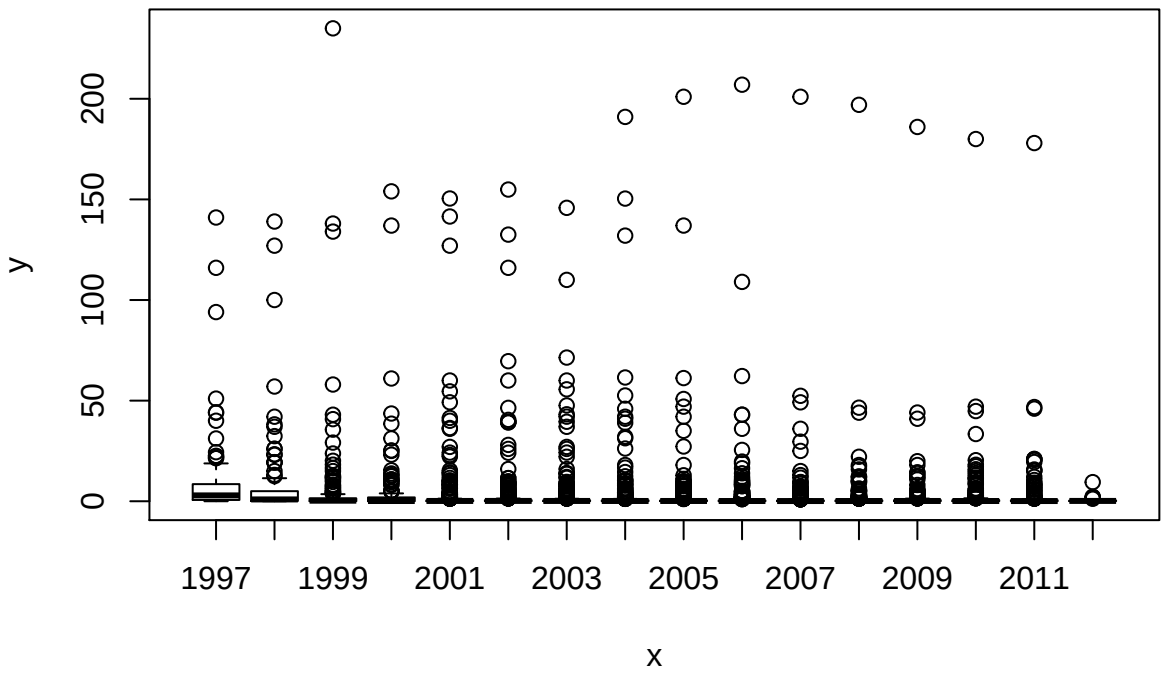
To draw multiple scatter plots you can use the "pairs" command.

```
pairs(~ data_clean$Employees + data_clean$`Assets - Total` + data_clean$`Sales/Turnover (Net)` , data_cl
```

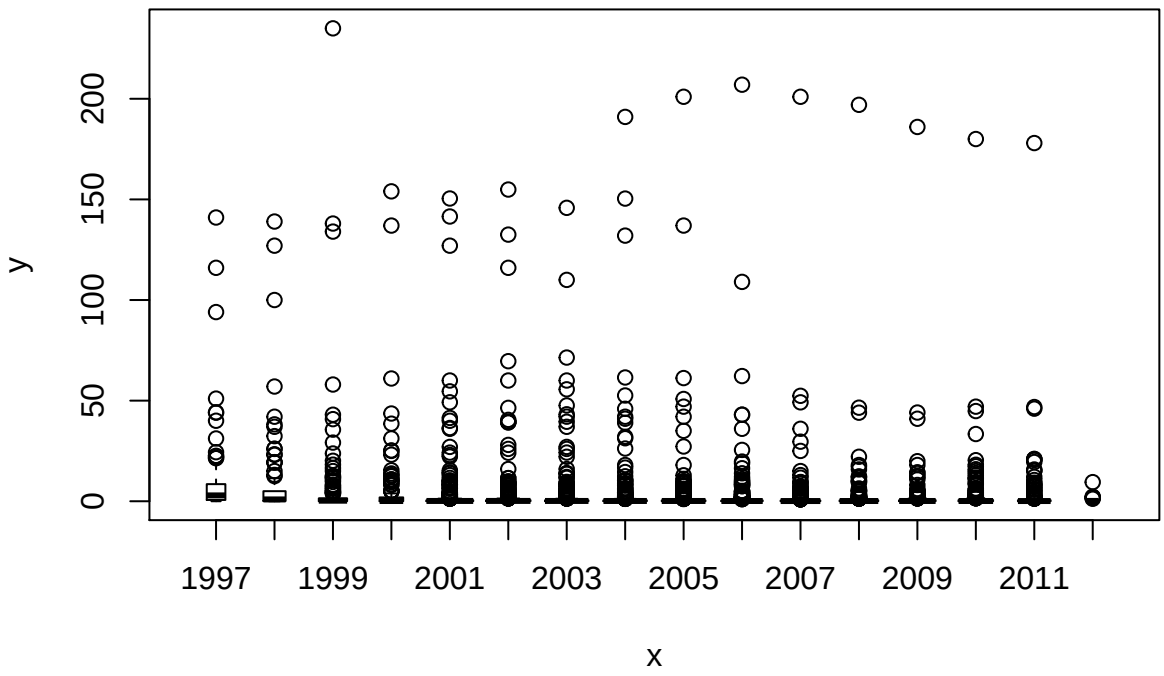


An easy way to check if there are outliers in your data is by using a "boxplot".

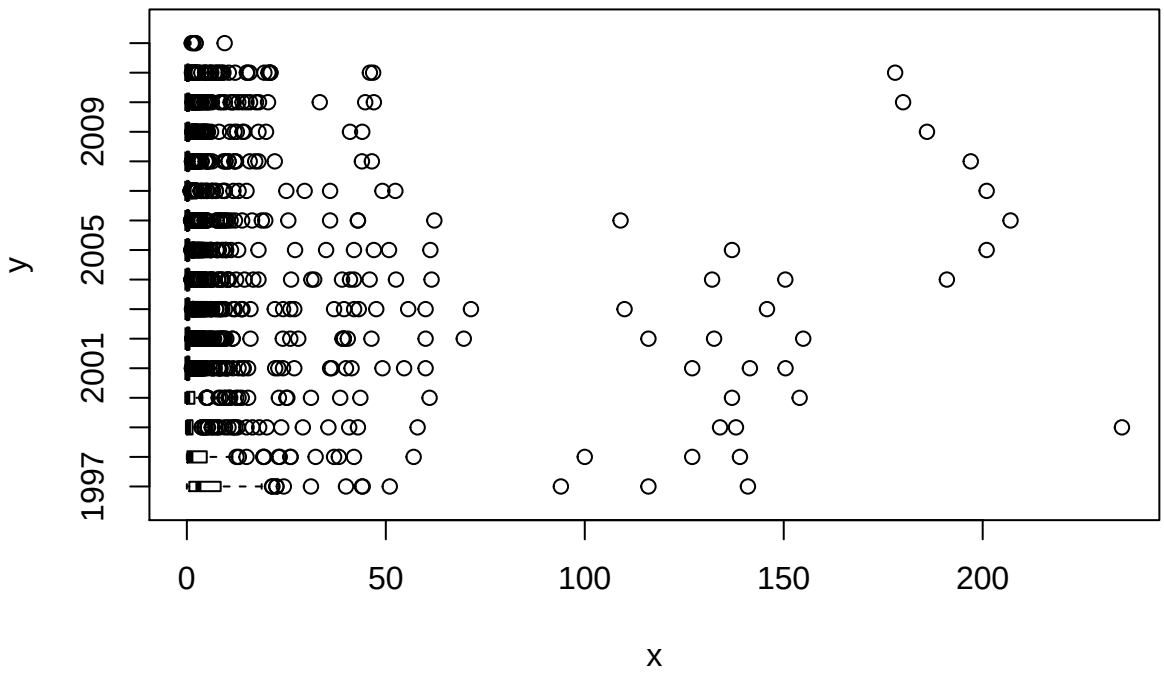
```
year <- as.factor(data_clean$`Data Year - Fiscal`)
plot(year, data_clean$Employees)
```



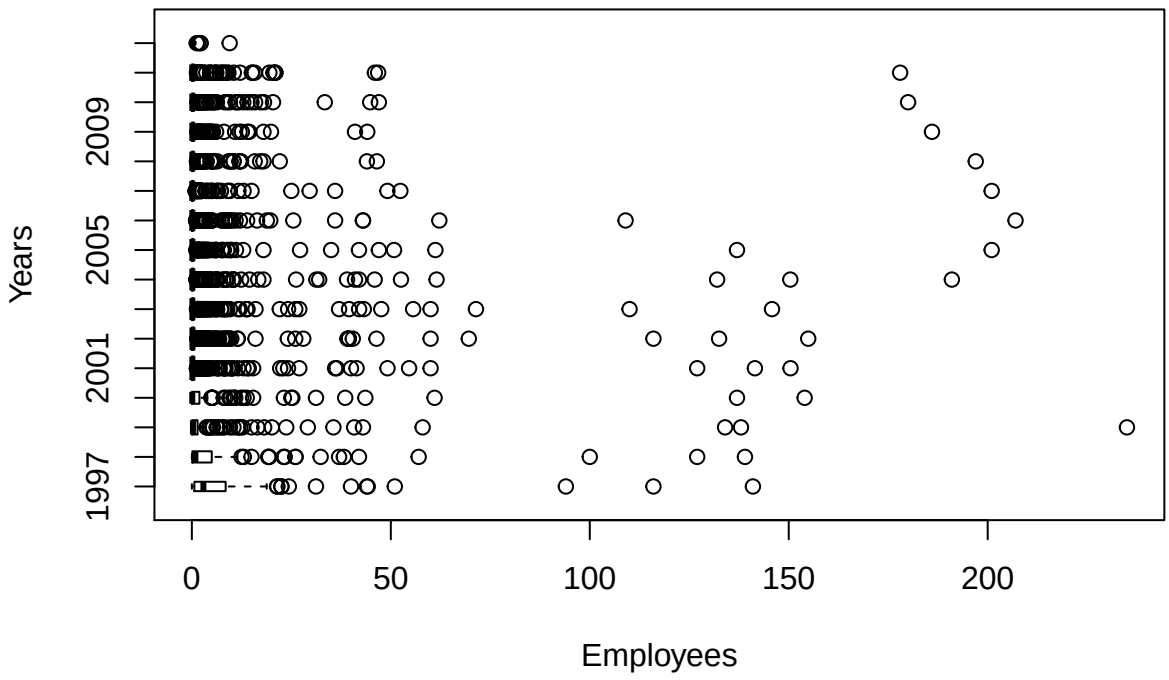

```
plot(year, data_clean$Employees, varwidth=T)
```




```
plot(year, data_clean$Employees, varwidth=T, horizontal=T)
```

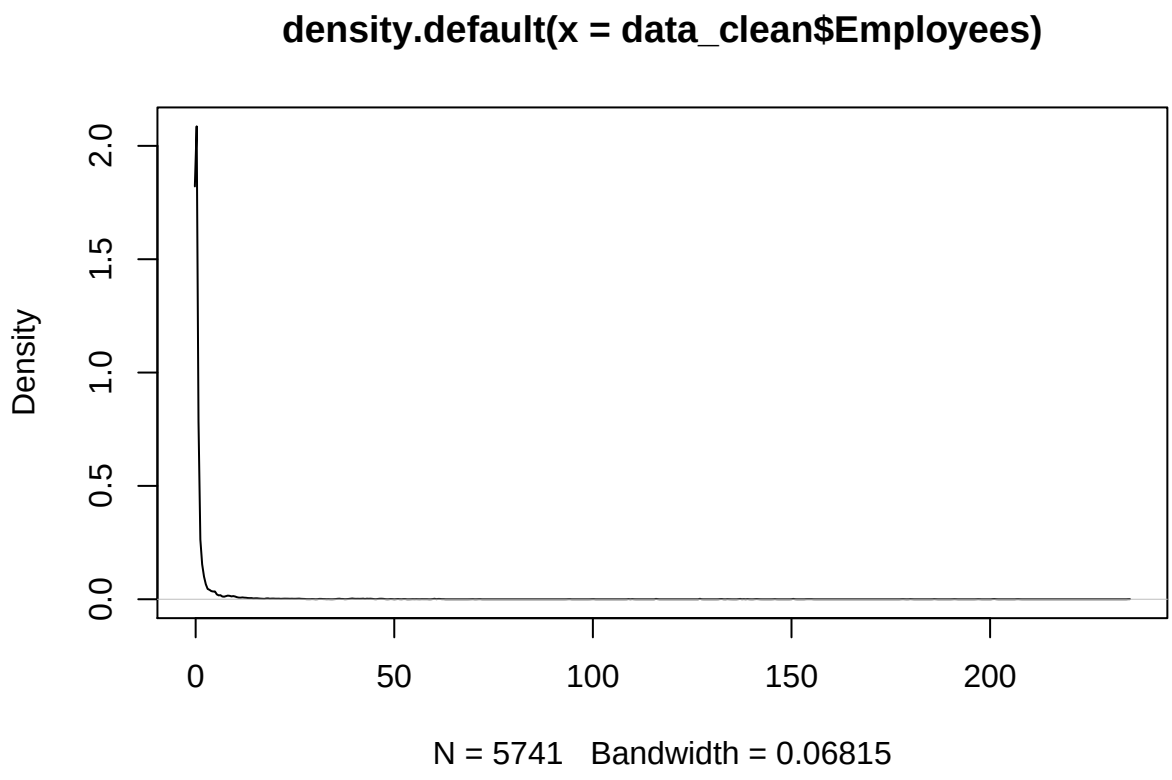



```
plot(year, data_clean$Employees, varwidth=T, horizontal=T, xlab="Employees",ylab="Years")
```



Let's now focus on the density distribution of employees.

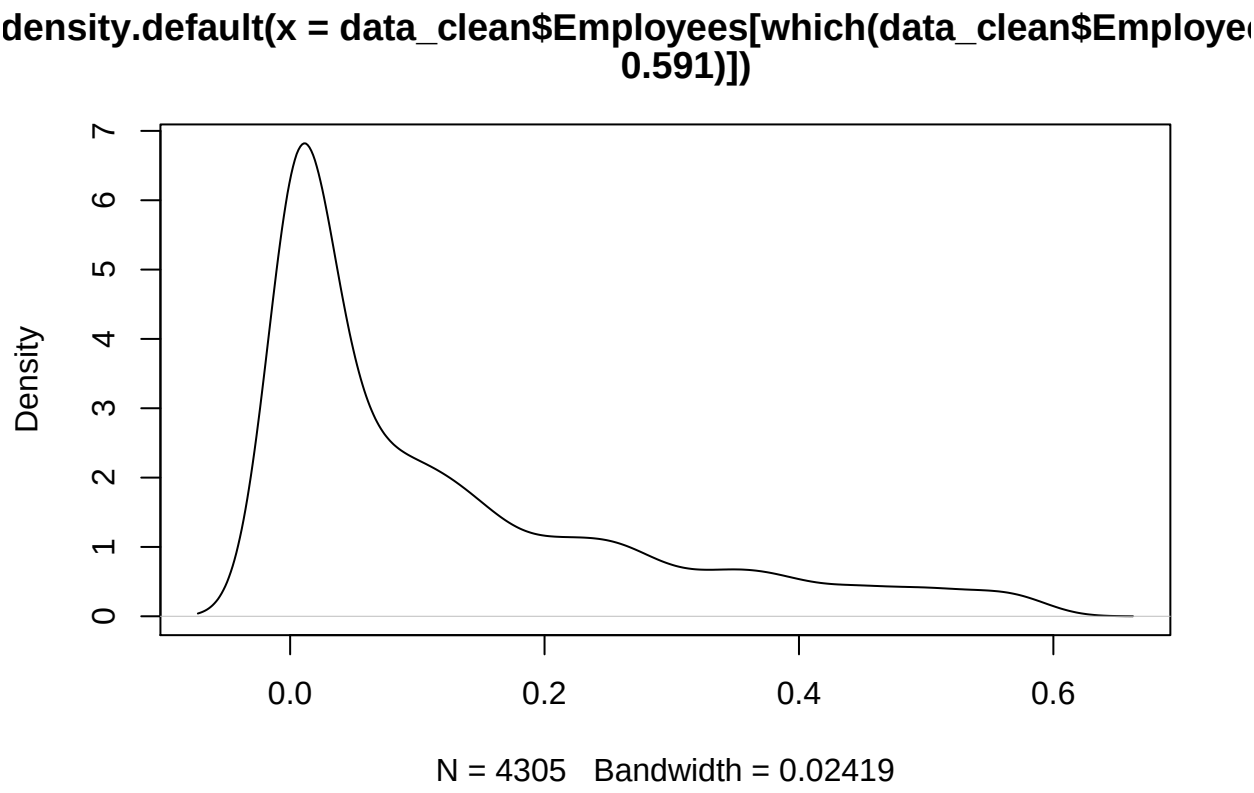
```
plot(density(data_clean$Employees))
```




```
summary(data_clean$Employees)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.018   0.139   2.363   0.591 235.000
```

```
plot(density(data_clean$Employees[which(data_clean$Employees<0.591)]))
```

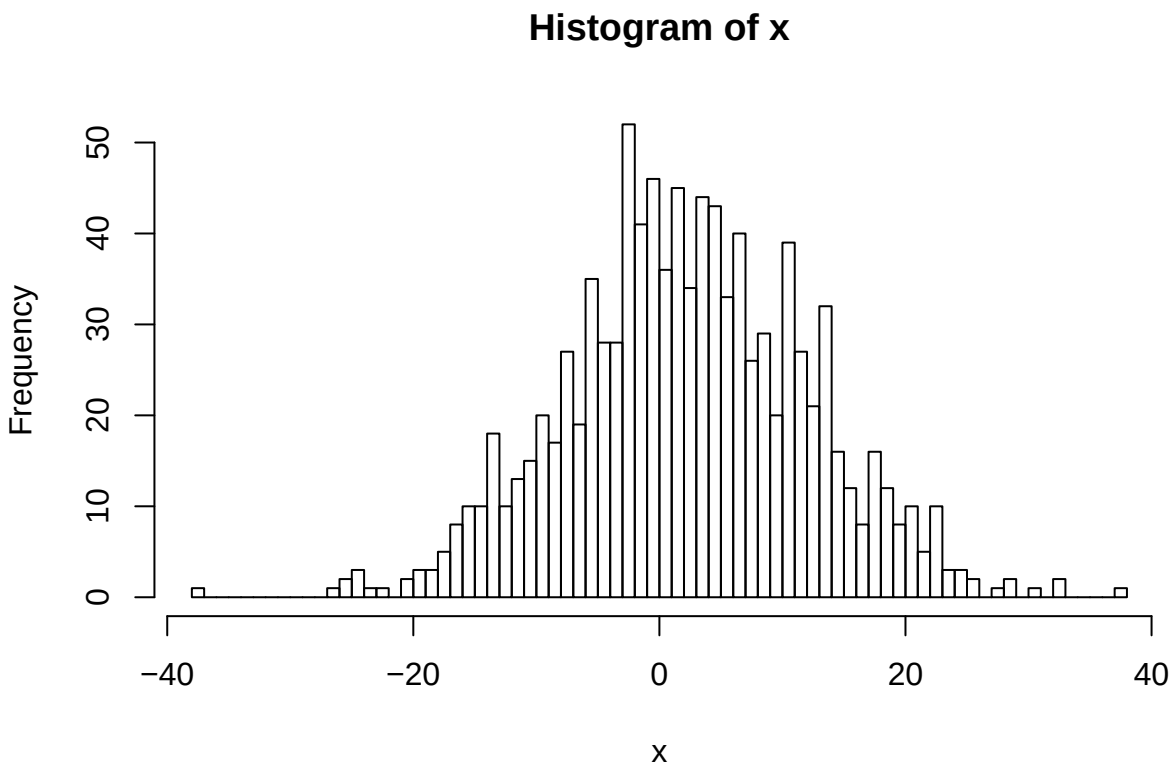


Variables Simulation to Fit the Empirical Distribution

You may want to compare the empirical distribution of a certain variable with a set of simulated distributions. Any statistical distribution can be generated from a statistical model and it provides a description of how the data were generated.

Before comparing empirical and simulated distributions, we can introduce the normal distribution. You can generate normally distributed data by using the "rnorm" function.

```
x <- rnorm(1000, mean = 2, sd = 10)
bin <- hist(x,100)
```

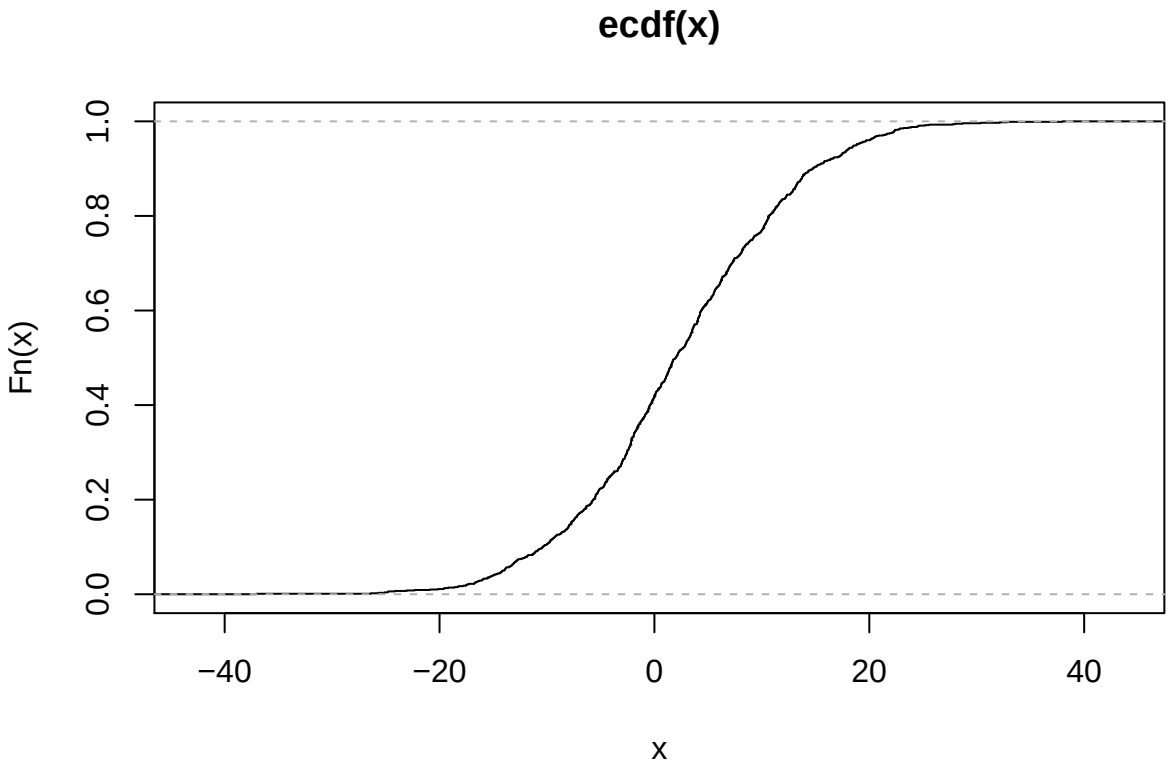


You can explore the cumulative density distribution of x by running the "ecdf()" function.

```
ecdf(x)
```

```
## Empirical CDF
## Call: ecdf(x)
##  x[1:1000] = -37.052, -26.403, -25.717,  ..., 32.837, 37.979
```

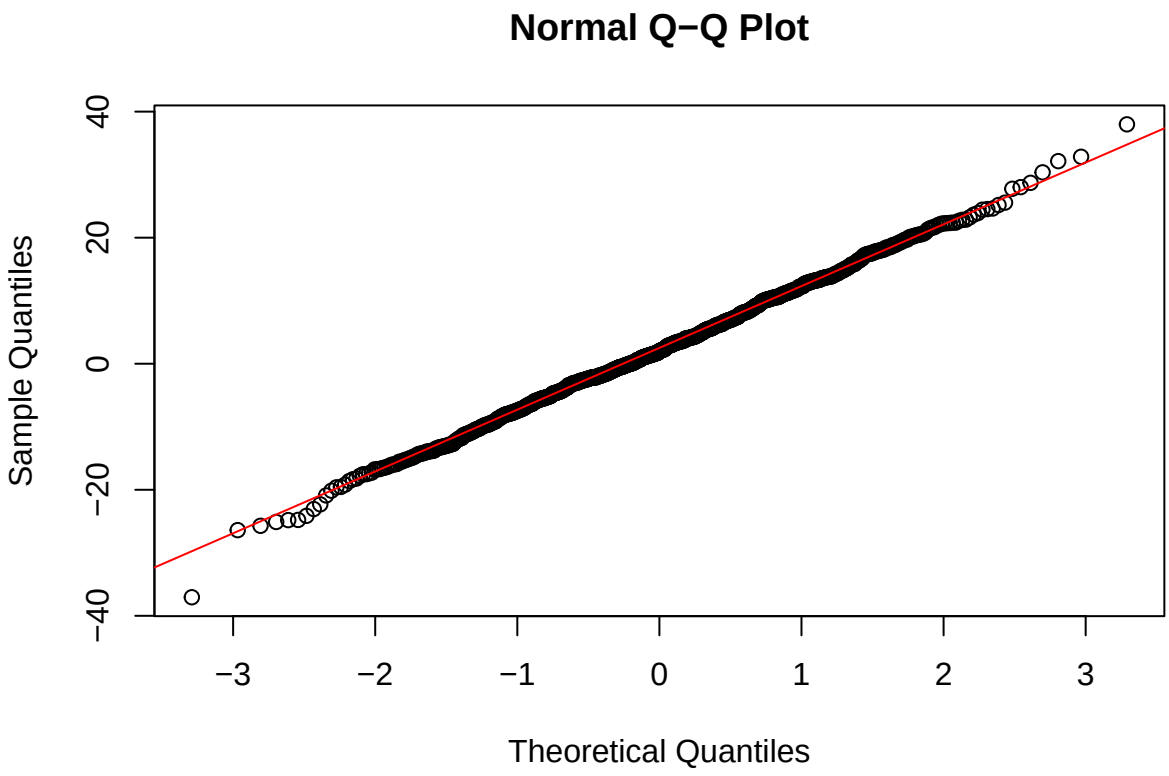
```
plot(ecdf(x))
```



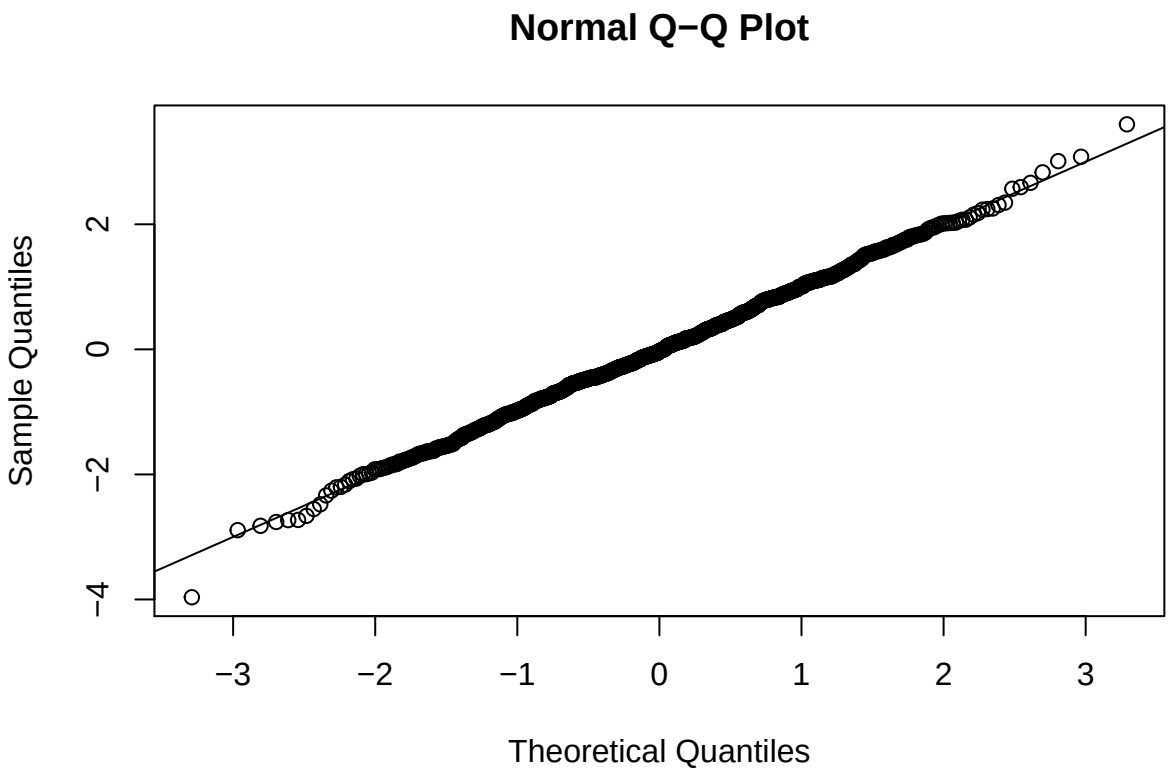
An plot that is extensively used to compare different distributions is quantile-quantile plot (remember that it is not a propbability distribution plot).

A point (x, y) on the qq-plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate). Thus the line is a parametric curve with the parameter which is the (number of the) interval for the quantile. If the two distribution are similar the points of the qqplot are on the line x=y.

```
qqnorm(x)
qqline(x, col="red")
```

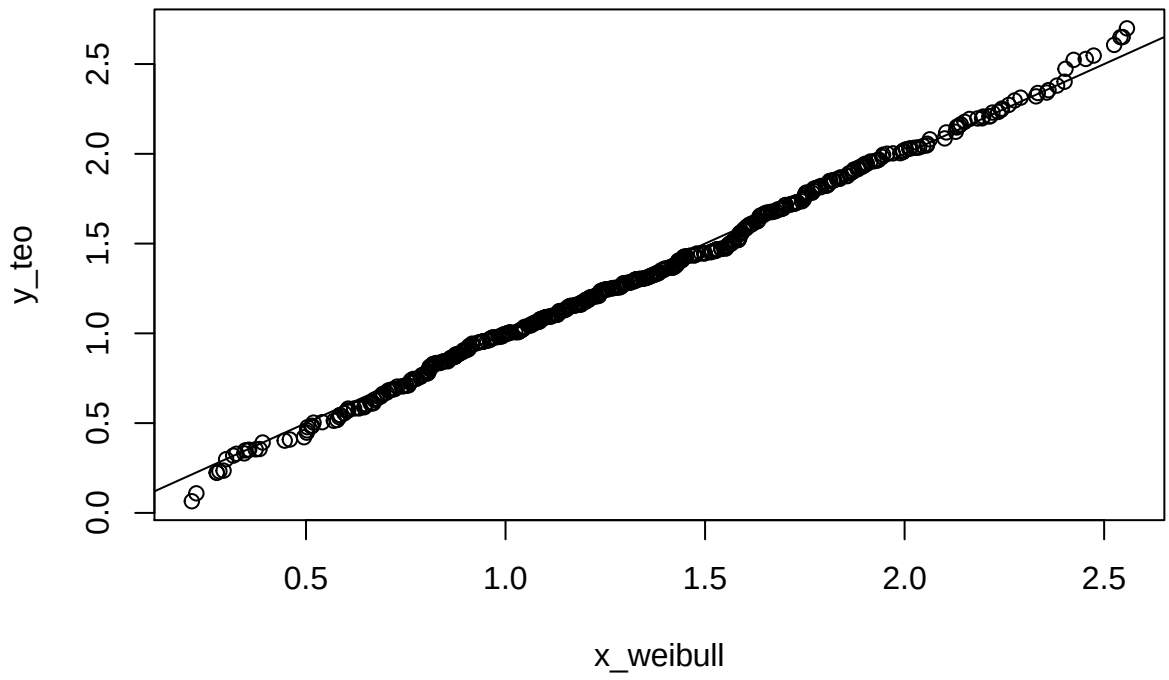



```
x_norm<-(x-mean(x))/sd(x)
qqnorm(x_norm)
abline(0,1)
```



If you want to test another distribution, you can still use the qq-plot.

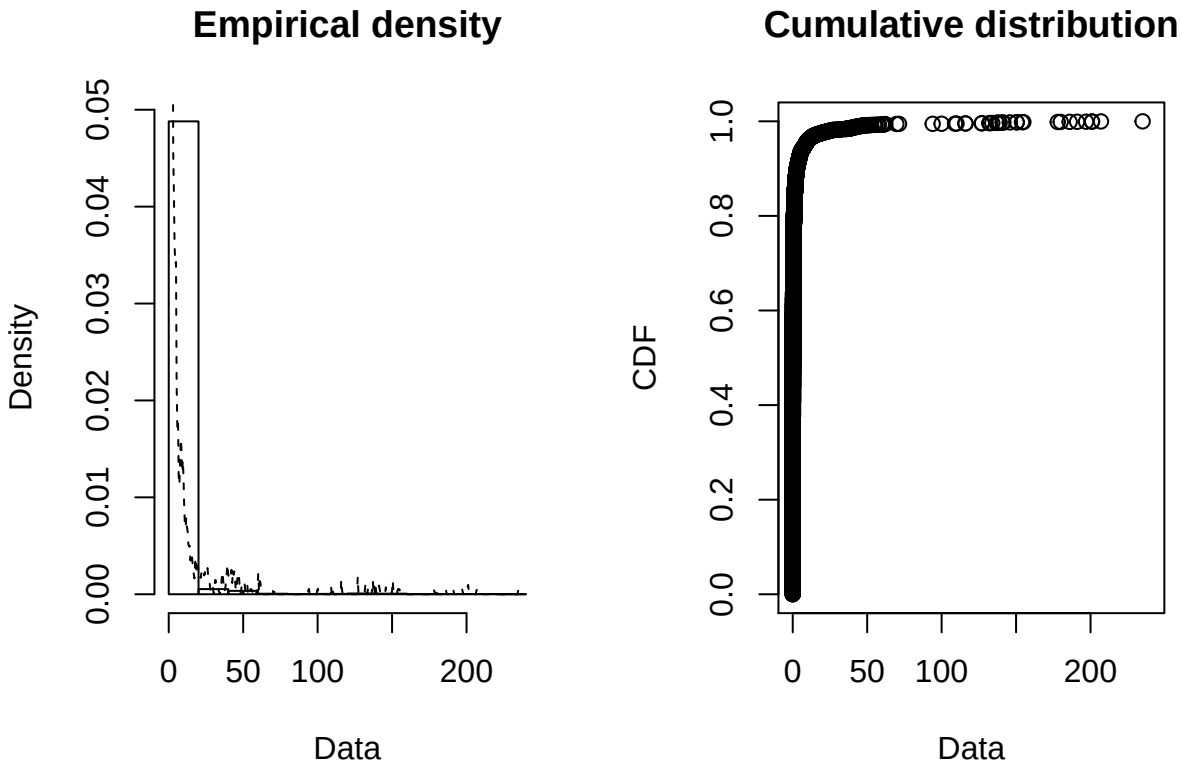
```
x_weibull<-rweibull(n=500,shape=3,scale=1.5)
y_teo<-rweibull(n=500, shape=2.9, scale=1.45)
qqplot(x_weibull,y_teo)
abline(0,1)
```




```
#install.packages("fitdistrplus")
library("fitdistrplus")
```

```
## Warning: package 'fitdistrplus' was built under R version 3.6.1
## Loading required package: MASS
## Loading required package: survival
## Loading required package: npsurv
## Loading required package: lsei
```

```
plotdist(data_clean$Employees, histo = TRUE, demp = TRUE)
```

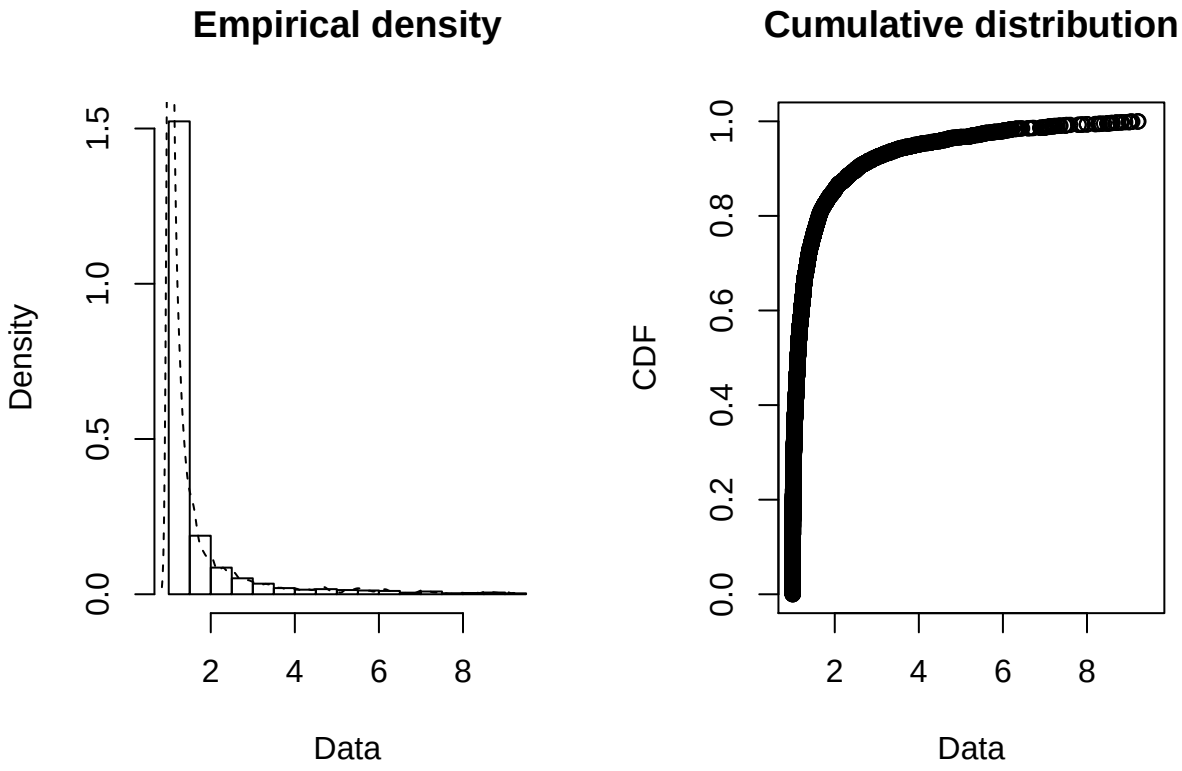


Let's now cut the tail of the distribution to get a "clearer" density of the variable.

```
quantile(data_clean$Employees, probs = c(0.05, 0.95))
```

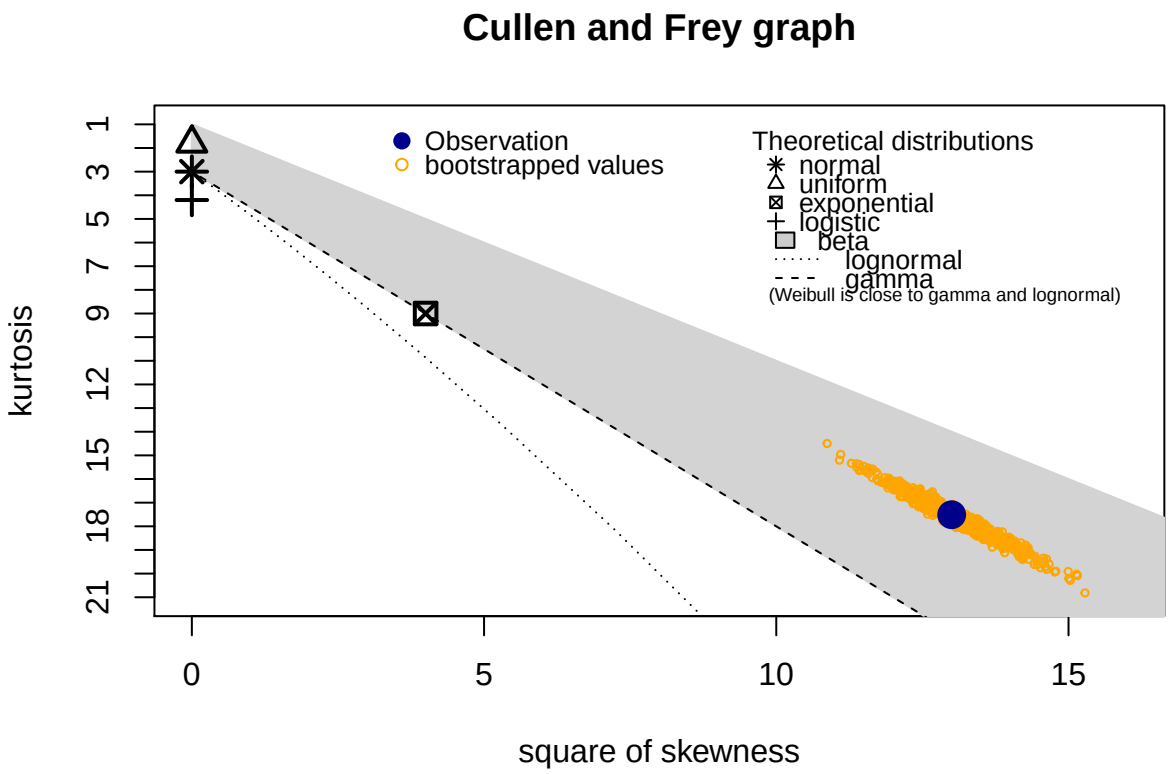
```
##  5% 95%
## 0.0 8.3
```

```
employees <- data_clean$Employees[which(data_clean$Employees<8.3)] + 1
plotdist(employees, histo = TRUE, demp = TRUE)
```



We can now try to fit a series of theoretical distribution to the empirical distribution of data. Let's first get some summary statistics on the min, max, median, mean, standard deviation, skewness (of the asymmetry of the probability distribution of a real-valued random variable about its mean; i.e., positive skewness means left leaning curve, while negative skewness means right leaning curve) and kurtosis (in probability theory and statistics, kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable) of the distribution.

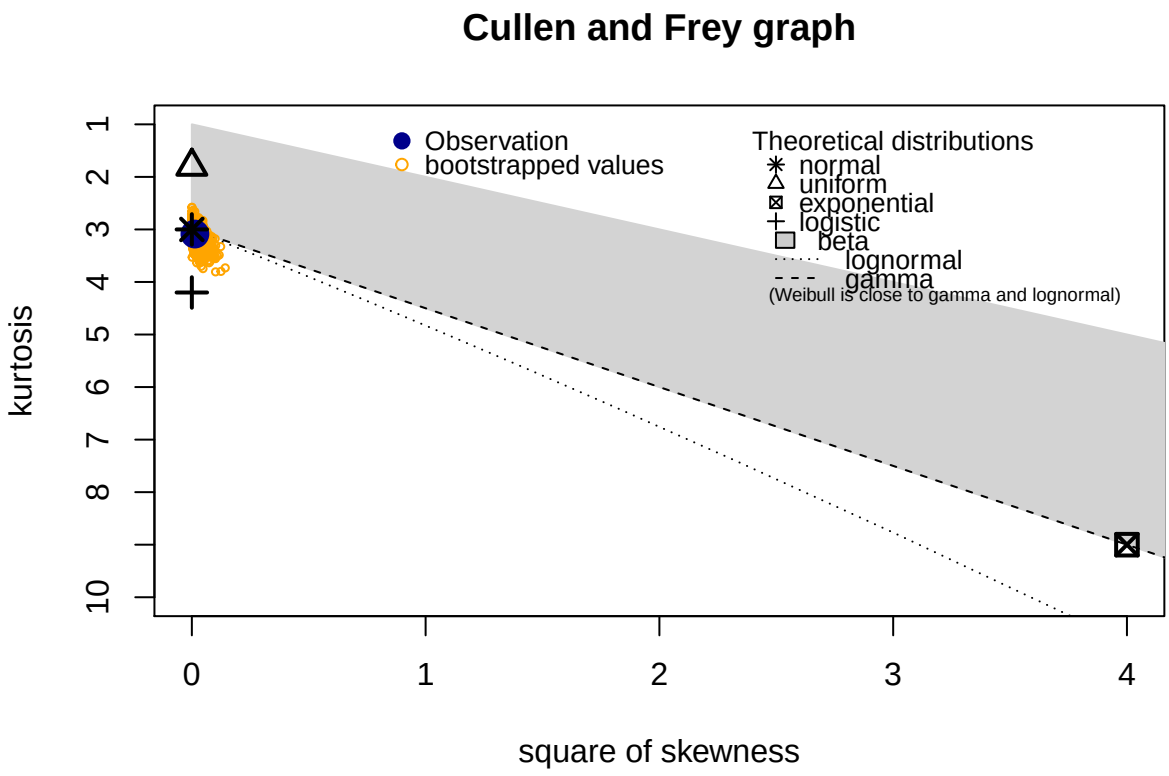
```
descdist(employees, boot = 1000)
```




```
## summary statistics
## ------
## min:  1   max:  9.221
## median:  1.121
## mean:  1.555673
## estimated sd:  1.172585
## estimated skewness:  3.605735
## estimated kurtosis:  17.51095
```

You can compare these values with the values of the standard normal distribution that previously generate.

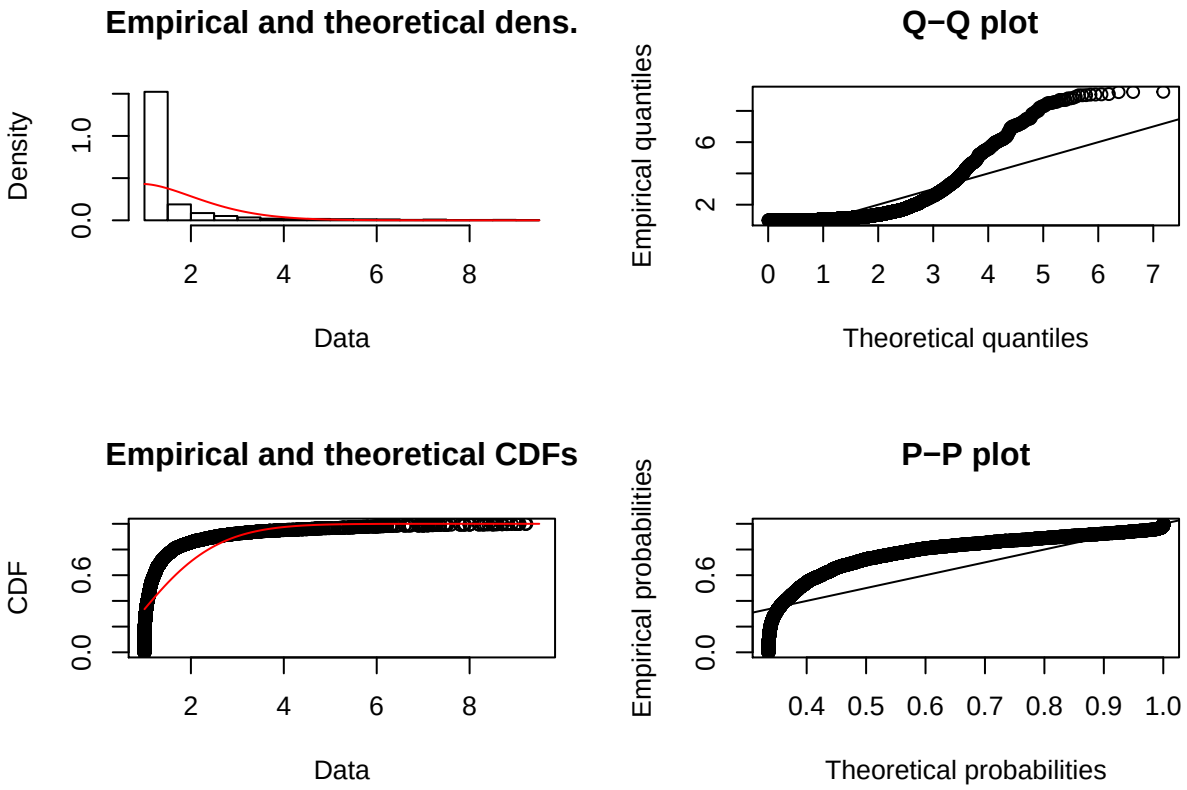
```
x<-rnorm(1000, mean = 0, sd = 1)
descdist(x, boot = 1000)
```



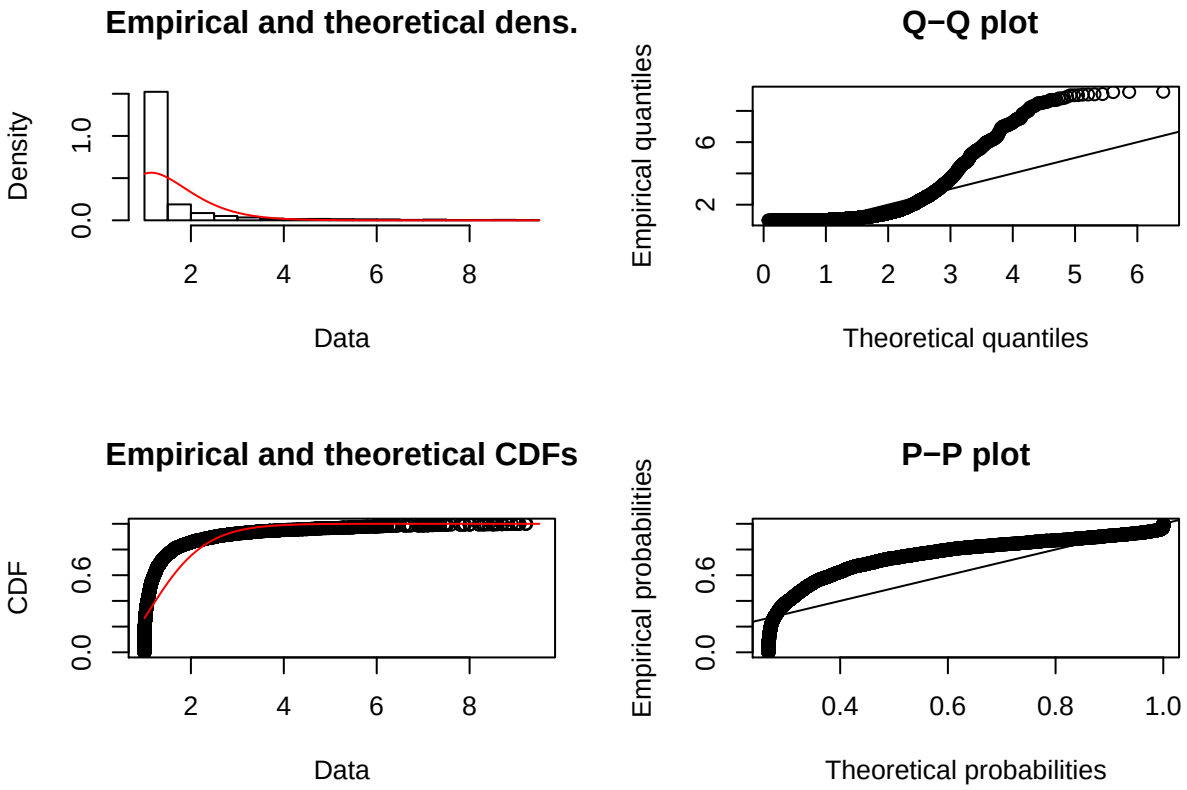

```
## summary statistics
## ------
## min:  -4.13514   max:  2.972876
## median:  -0.03158496
## mean:  -0.05410018
## estimated sd:  1.021524
## estimated skewness:  -0.1143976
## estimated kurtosis:  3.087732
```

Clearly, the distribution of the number of employees is not a normal distribution. Let's see how this distribution "visually" compares with a Weibull, a Gamma and a log-normal distribution.

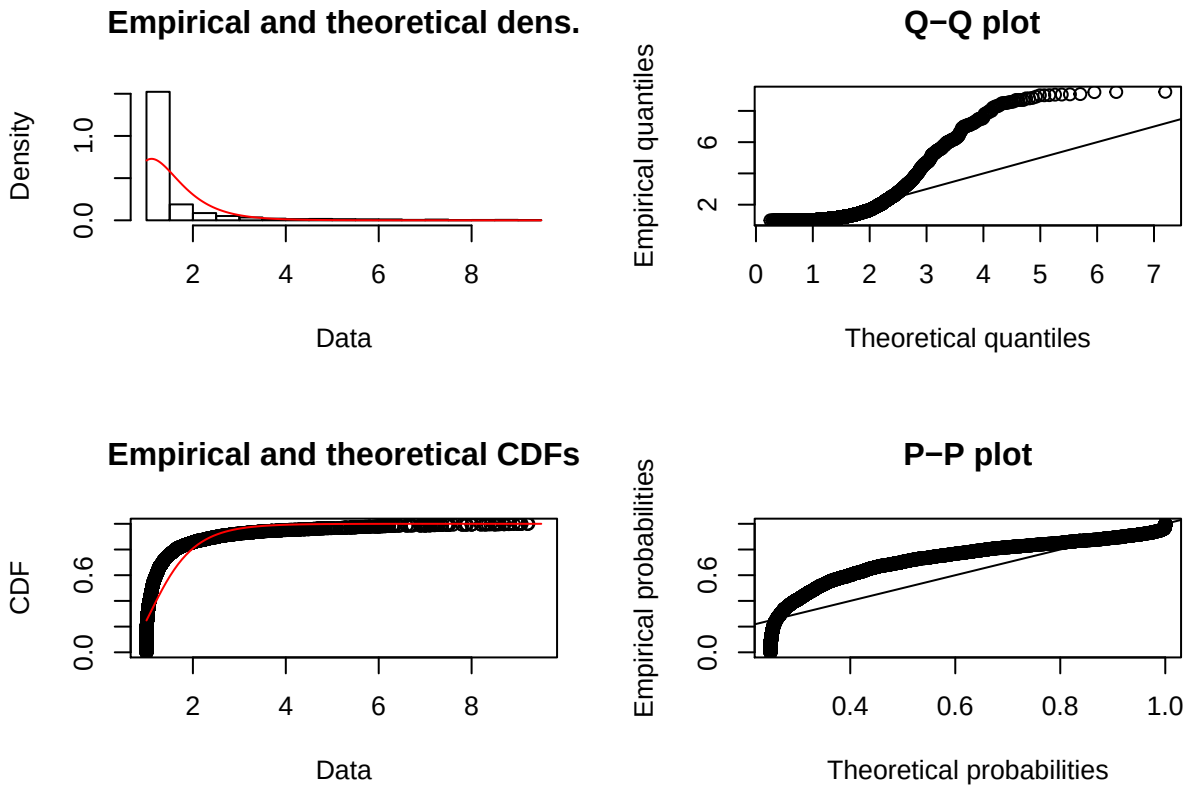
```
# Weibull
fw <- fitdist(employees, "weibull", method = "mle", lower = c(0, 0))
plot(fw)
```




```r
# Gamma
fg <- fitdist(employees, "gamma", method = "mle", lower = c(0, 0))
plot(fg)
```

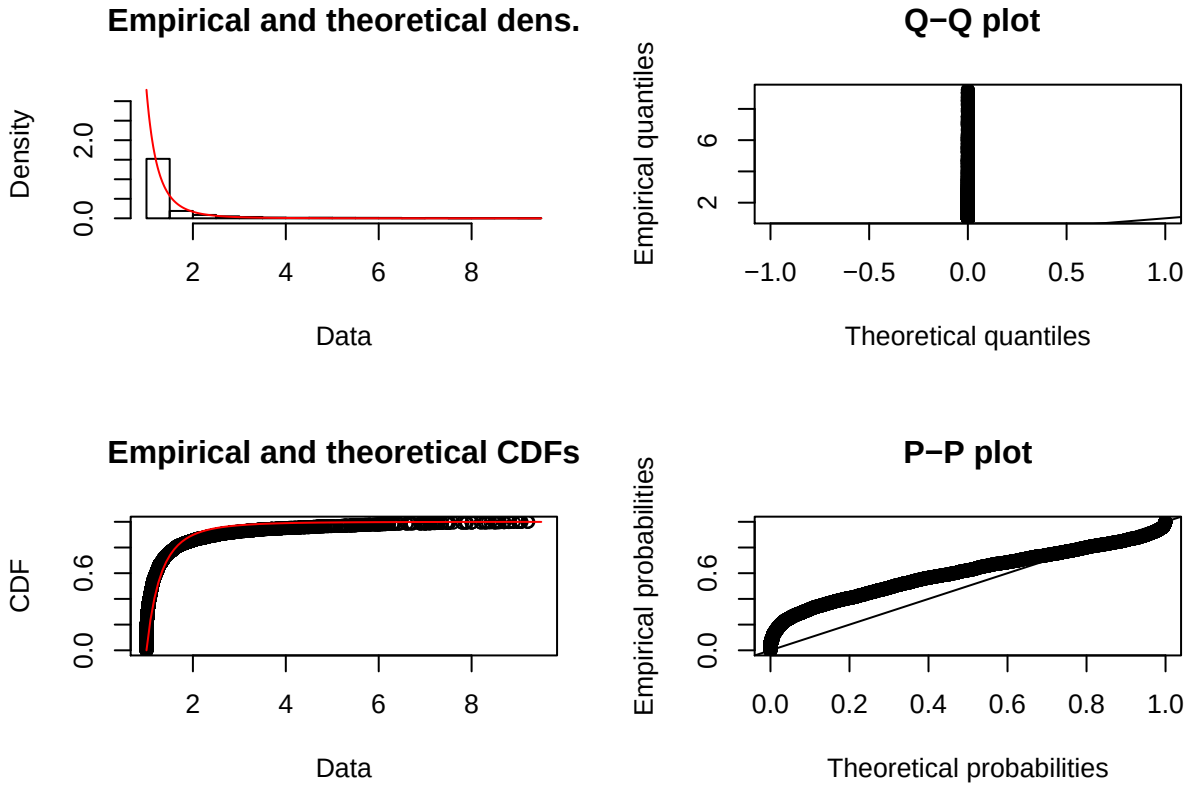



```r
# log-normal
fln <- fitdist(employees, "lnorm",  method = "mle", lower = c(0, 0))
plot(fln)
```



How about a power law distribution?

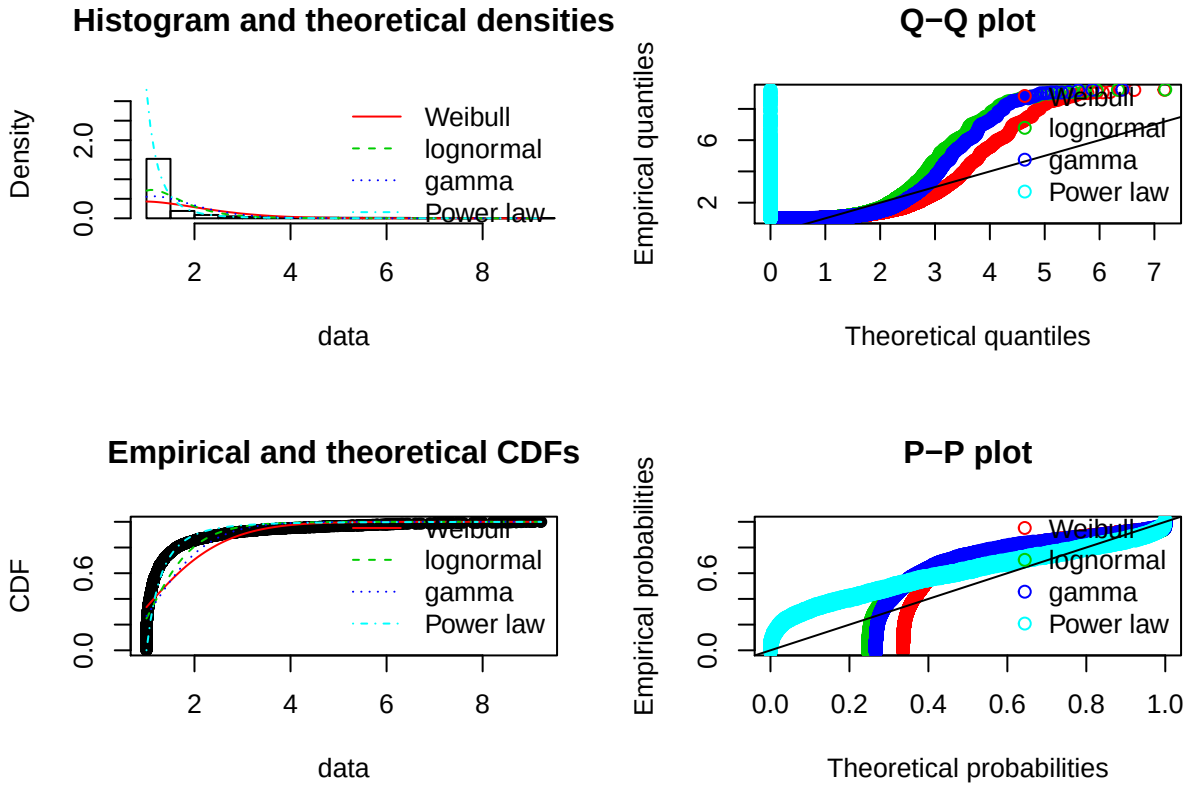
```
# power-law distribution
plw <- fitdist(employees, "plcon",
               start = list(xmin=1,alpha=2),
               lower = c(xmin = 0, alpha = 1))
plot(plw)
```



Now we can make a comparative plot with all the theoretical distributions and the empirical distribution.

```
par(mfrow = c(2, 2))
plot.legend <- c("Weibull", "lognormal", "gamma", "Power law")

denscomp(list(fw, fln, fg, plw), legendtext = plot.legend)
qqcomp(list(fw, fln, fg, plw), legendtext = plot.legend)
cdfcomp(list(fw, fln, fg, plw), legendtext = plot.legend)
ppcomp(list(fw, fln, fg, plw), legendtext = plot.legend)
```



The P-P plot (probability-probability plot or percent-percent plot or P value plot) is a probability plot for assessing how closely two data sets agree, which plots the two cumulative distribution functions against each other.

Kolmogorov-Smirnov, Cramer-von Miser and Anderson-Darling are all goodness-of-fit statistics based on the CDF distance (i.e., the Kolmogorov-Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples). AIC and BIC are classical penalized criteria based on the loglikehood

```
gofstat(list(fw, fln, fg, plw),fitnames = c("weibull", "lognorma", "gamma", "Power law"))
```

```
## Goodness-of-fit statistics
##                                 weibull    lognorma       gamma
## Kolmogorov-Smirnov statistic   0.3357433   0.2480156   0.2667879
## Cramer-von Mises statistic   147.7416083 105.3514542 133.3044502
## Anderson-Darling statistic   759.7990779 569.3733191 694.8111291
##                                Power law
## Kolmogorov-Smirnov statistic    0.2249882
## Cramer-von Mises statistic    105.7809480
## Anderson-Darling statistic   1374.1499989
##
## Goodness-of-fit criteria
##                                 weibull lognorma    gamma Power law
## Akaike's Information Criterion 13700.11  9995.39 12016.61  1234.001
## Bayesian Information Criterion 13713.31 10008.60 12029.82  1247.209
```