Manuel J. A. Eugster

Programming R

Chapter 2: Sample Session

Last modification on April 23, 2012

Draft of the R programming book I always wanted to read
http://mjaeugster.github.com/progr

2 Sample Session

An exemplary analysis.

This is based on the analysis provided in Chapter 1 of the book (available as vignette int the `HSAUR2` package):

- *A Handbook of Statistical Analyses Using R* by Everitt and Hothorn (2008)

2.1 The Forbes 2000 Ranking of the World's Biggest Companies

In this sample session the 2000 world's biggest companies are analyzed. The data are collected by the Forbes Magazine in 2004; and provided as an R object by the `HSAUR2` package.

```
> install.packages("HSAUR2")
```

Is the package installed, the data set object can be attach to the global environment (without loading the complete package).

```
> data("Forbes2000", package = "HSAUR2")
> ls()

[1] "Forbes2000" "txt"
```

The data set's help page provides a detailed description of the variables.

```
> help("Forbes2000", package = "HSAUR2")
```

Note that the shortcut `?` does not work when the package is not loaded.

For a first inspection, take a look at the structure of the data set using the `str()` function.

```
> str(Forbes2000, vec.len = 1)

'data.frame': 2000 obs. of  8 variables:
 $ rank       : int  1 2 ...
 $ name       : chr  "Citigroup" ...
 $ country    : Factor w/ 61 levels "Africa","Australia",..: 60 60 ...
 $ category   : Factor w/ 27 levels "Aerospace & defense",..: 2 6 ...
 $ sales      : num  94.7 ...
 $ profits    : num  17.9 ...
 $ assets     : num  1264 ...
 $ marketvalue: num  255 ...
```

The output of `str()` shows that `Forbes2000` is an object of the class `data.frame` with 2000 observations of 8 variables. For each variable, its class is listed as well as the first few observations. The argument `vec.len` defines how many "first few" elements are displayed of each vector.

Use the `print()` function to see the complete data set, and the `head()` and `tail()` functions to see the first and last parts, respectively. Note that the `print()` function is implicitly executed when only the name of the data set is entered.

Basic characteristics of the data set are provided by: `ncol()` for the number of variables; `nrow()` for the number of observations; `names()` for the variable names.

```
> c(rows = nrow(Forbes2000), cols = ncol(Forbes2000))

rows cols
2000    8

> names(Forbes2000)

[1] "rank"        "name"        "country"     "category"    "sales"
[6] "profits"     "assets"      "marketvalue"
```

2.2 Simple Summary Statistics

A statistical overview is provided by the `summary()` function. It provides a meaningful summary for each variable class; e.g., a five-point summary plus mean for numeric variables and the absolute frequencies for factor variables.

```
> summary(Forbes2000)

      rank          name                     country
 Min.   :   1   Length:2000        United States :751
 1st Qu.: 501   Class :character   Japan         :316
 Median :1000   Mode  :character   United Kingdom:137
 Mean   :1000                      Germany       : 65
 3rd Qu.:1500                      France        : 63
 Max.   :2000                      Canada        : 56
                                   (Other)       :612
                  category        sales           profits
 Banking                : 313   Min.   :  0.01   Min.   :-25.830
 Diversified financials: 158   1st Qu.:  2.02   1st Qu.:  0.080
 Insurance              : 112   Median :  4.37   Median :  0.200
 Utilities              : 110   Mean   :  9.70   Mean   :  0.381
 Materials              :  97   3rd Qu.:  9.55   3rd Qu.:  0.440
 Oil & gas operations   :  90   Max.   :256.33   Max.   : 20.960
 (Other)                :1120                    NA's   :  5.000
    assets         marketvalue
 Min.   :   0.27   Min.   :  0.02
 1st Qu.:   4.03   1st Qu.:  2.72
 Median :   9.35   Median :  5.15
 Mean   :  34.04   Mean   : 11.88
 3rd Qu.:  22.79   3rd Qu.: 10.60
 Max.   :1264.03   Max.   :328.54
```

Common summary statistics for numerical variables are available via the functions `mean()`, `median()`, `range()`, `quantile()`, etc. The absolute frequencies for a factor is available via the `table()` function, the number of levels via `nlevels` and the different levels via `levels()`.

In order to access a variable of a `data.frame` use the `$` operator.

```
> median(Forbes2000$sales)

[1] 4.365

> range(Forbes2000$assets)

[1]    0.27 1264.03
```

```
> table(Forbes2000$category)


            Aerospace & defense                         Banking
                             19                             313
    Business services & supplies                 Capital goods
                             70                              53
                      Chemicals                 Conglomerates
                             50                              31
                   Construction              Consumer durables
                             79                              74
         Diversified financials          Drugs & biotechnology
                            158                              45
           Food drink & tobacco                   Food markets
                             83                              33
Health care equipment & services     Hotels restaurants & leisure
                             65                              37
   Household & personal products                     Insurance
                             44                             112
                      Materials                         Media
                             97                              61
           Oil & gas operations                     Retailing
                             90                              88
                Semiconductors             Software & services
                             26                              31
 Technology hardware & equipment      Telecommunications services
                             59                              67
               Trading companies                  Transportation
                             25                              80
                      Utilities
                            110
```

2.3 Subsets

A subset of the elements (rows and columns) of a `data.frame` can be extracted using the `[` operator. This operator takes two index vectors separated by a comma—the first for the observations (rows) and the second for the variables (columns). A missing index vector means that all available data of this dimension are selected.

The first ten observations and all variables:

```
> Forbes2000[1:10, ]
```

All observations of the second variable:

```
> Forbes2000[, 2]       # by index
> Forbes2000[, "name"]  # by name
```

The first three companies:

```
> vars <- c("name", "sales", "profits", "assets")
> Forbes2000[1:3, vars]


                 name  sales profits assets
1           Citigroup  94.71   17.85 1264.0
2    General Electric 134.19   15.59  626.9
3 American Intl Group  76.66    6.46  647.7
```

An index vector can also be a logical vector. In this case, the vector must be of the same length as the number of rows or columns. Values corresponding to `TRUE` are selected, `FALSE` are omitted.

All companies with assets greater than 1000:

```
> table(Forbes2000$assets > 1000)


FALSE  TRUE
 1997     3

> Forbes2000[Forbes2000$assets > 1000, "name"]

[1] "Citigroup"        "Fannie Mae"       "Mizuho Financial"
```

All companies from Australia and with a rank lower than 100:

```
> Forbes2000[Forbes2000$country == "Australia" &
+            Forbes2000$rank < 100, ]


   rank                name   country category sales profits assets
86   86 Natl Australia Bank Australia  Banking 15.34    2.69  269.9
   marketvalue
86       36.51
```

See also: Use the function `subset()` to avoid a lot of typing in complex subsets.

2.4 Missing Values

The `data.frame`'s summary shows for the numerical variable `profits` an additional statistics, i.e., `NA's`.

```
> summary(Forbes2000$profits)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
-25.800   0.080   0.200   0.381   0.440  21.000   5.000
```

For some of the companies the measurement of the `profits` variable are missing. In R a value which is *not available* or a *missing value* in the statistical sense, is indicated by the special value `NA`.

This affects a lot of summary statistics and statistical methods; e.g.,

```
> mean(Forbes2000$profits)

[1] NA
```

is `NA` because of the available missing values. One has to explicitly decide to ignore the missing values:

```
> mean(Forbes2000$profits, na.rm = TRUE)

[1] 0.3811
```

Use the `is.na()` function to find the missing values,

```
> which(is.na(Forbes2000$profits))

[1]  772 1085 1091 1425 1909
```

and `complete.cases()` to find all complete cases.

The function `na.omit()` removes all observations with missing values.

```
> Forbes2000cc <- na.omit(Forbes2000)
> mean(Forbes2000cc$profits)

[1] 0.3811
```

2.5 Basic Data Manipulations

A new variable is added to a `data.frame` using the `$<-` operator:

```
> Forbes2000cc$costs <- Forbes2000cc$sales - Forbes2000cc$profits
```

A variable is removed by assigning `NULL`:

```
> Forbes2000cc$category <- NULL
```

Using the subset operator we can work on a set of variables; e.g., scale all numerical variables:

```
> Forbes2000cc[, 4:7] <- scale(Forbes2000cc[, 4:7])
```

Note that `scale()` can be applied to a set of variables. Other functions can only be applied to vectors, and hence must be applied to each column in turn:

```
> median(Forbes2000cc[, 4:7])

Error: need numeric data
```

```
> c(median(Forbes2000cc[, 4]),
+   median(Forbes2000cc[, 5]))

[1] -0.2968 -0.1026
```

See also: Use `sapply()` to apply a function on each individual variable of a data.frame.

Operating on each row is much more tricker. Whereas each column is a variable of a single class (numeric, factor, etc.) a row can be rather diverse.

```
> Forbes2000cc[722, ]

    rank              name country   sales profits assets marketvalue
722  722 Skandia Insurance  Sweden 0.04555 -0.4991 0.2117     -0.2871
    costs
722 11.03
```

Manipulating a cell is done using the `[<-` operator.

```
> Forbes2000cc[722, "sales"] <- 0.05
> Forbes2000cc[722, ]

    rank              name country sales profits assets marketvalue costs
722  722 Skandia Insurance  Sweden  0.05 -0.4991 0.2117     -0.2871 11.03
```

See also: `apply()` which applies a function over margins (i.e., on rows, columns or cells).

2.6 Saving data

Save an R specific external representation of a data set (or object in general) to a file in the active working directory:

```
> saveRDS(Forbes2000cc, file = "Forbes2000cc.rds")
> rm(Forbes2000cc)
```

At a later date we can load the data set via:

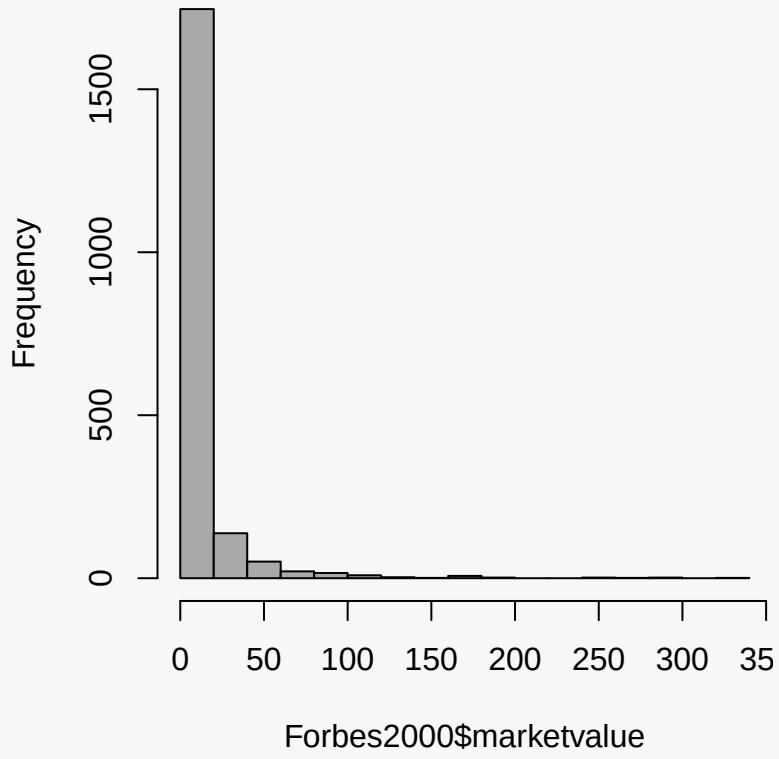
```
> Forbes2000cc <- readRDS("Forbes2000cc.rds")
```

See also: See `?write.table` for saving a textual representation of data sets; `?save` to save more than one object; and `?save.image` to save the complete workspace.

2.7 Simple Graphics

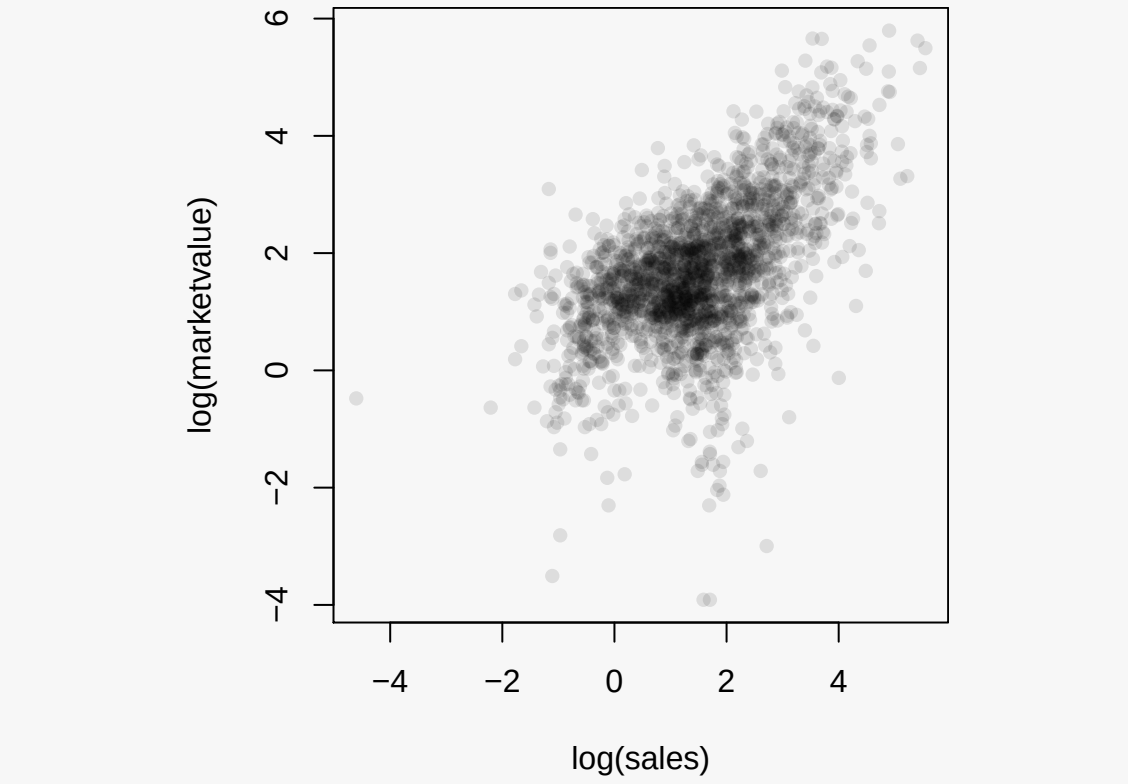
The degree of skewness of a distribution can be investigated by constructing histograms using the `hist()` function. For example, for the `marketvalue` variable:

```
> par(mar = c(4, 4, 0, 0))
> hist(Forbes2000$marketvalue, main = "", col = "darkgray")
```



Bivariate relationships of two continuous variables are usually depict as scatterplots. For example, the dependence of the marketvalue from sales (log-scaled):

```
> par(mar = c(4, 4, 0, 0))
> plot(log(marketvalue) ~ log(sales), data = Forbes2000,
+      pch = 16, col = rgb(0, 0, 0, 0.1))
```



2.8 Simple Linear Regression

A first analysis of the bivariate relationships between the two numerical variables can be done with a simple linear regression.

```
> m1 <- lm(log(marketvalue) ~ log(sales), data = Forbes2000)
```

The estimated coefficients are:

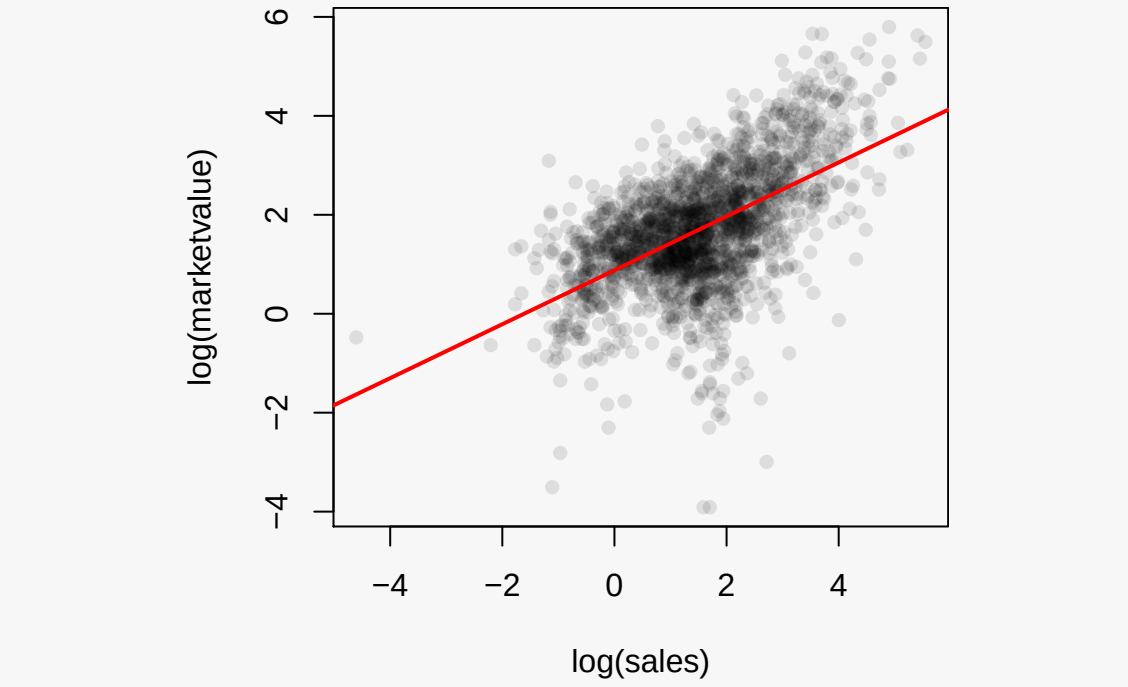
```
> m1


Call:
lm(formula = log(marketvalue) ~ log(sales), data = Forbes2000)

Coefficients:
(Intercept)   log(sales)
      0.878        0.545
```

The summary method of an object of class `lm` returns details about the linear model fit:

```
> summary(m1)


Call:
lm(formula = log(marketvalue) ~ log(sales), data = Forbes2000)

Residuals:
   Min     1Q Median     3Q    Max
-5.718 -0.512  0.066  0.642  2.855

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8780     0.0347    25.3   <2e-16 ***
log(sales)    0.5450     0.0180    30.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.988 on 1998 degrees of freedom
Multiple R-squared: 0.316,Adjusted R-squared: 0.315
F-statistic:  921 on 1 and 1998 DF,  p-value: <2e-16
```

Finally, we can visualize the fit within the data:

```
> par(mar = c(4, 4, 0, 0))
> plot(log(marketvalue) ~ log(sales), data = Forbes2000,
+      pch = 16, col = rgb(0, 0, 0, 0.1))
> abline(m1, col = "red", lwd = 2)
```


log(marketvalue)
6
4
2
0
−2
−4
−4
−2
0
2
4
log(sales)

Bibliography

Brian S. Everitt and Torsten Hothorn. *A Handbook of Statistical Analyses Using R*. Chapman & Hall, 2 edition, 2008. ISBN 1584885394.