Analysing Social Data Using R

Linear regression, part 1

Michał Bojanowski[1], mbojan@icm.edu.pl
Zbigniew Karpiński[2], zkarpinski@ifispan.waw.pl

[1]Interdisciplinary Centre for Mathematical and Computational Modelling
[2]Institute for Philosophy and Sociology

May 7, 2012

Trust and prudence

g5a Most people are basically honest.

g5b Most people are trustworthy.

g5c People are always interested only in their own welfare.

g5d Most people will respond in kind when they are trusted by others.

g5e In this society, one has to be alert or someone is likely to take advantage of you.

g5f In this day and age, one doesn't have to be constantly afraid of being cheated.

g5g Most people are basically good and kind.

g5h Most people are trustful of others.

g5i Most people inwardly dislike putting themselves out to help others.

Trust and prudence

g5a Most people are basically honest.

g5b Most people are trustworthy.

g5c People are always interested only in their own welfare.

g5d Most people will respond in kind when they are trusted by others.

g5e In this society, one has to be alert or someone is likely to take advantage of you.

g5f In this day and age, one doesn't have to be constantly afraid of being cheated.

g5g Most people are basically good and kind.

g5h Most people are trustful of others.

g5i Most people inwardly dislike putting themselves out to help others.

Trust and prudence

g5a Most people are basically honest.

g5b Most people are trustworthy.

g5c People are always interested only in their own welfare.

g5d Most people will respond in kind when they are trusted by others.

g5e In this society, one has to be alert or someone is likely to take advantage of you.

g5f In this day and age, one doesn't have to be constantly afraid of being cheated.

g5g Most people are basically good and kind.

g5h Most people are trustful of others.

g5i Most people inwardly dislike putting themselves out to help others.

The trust index

The index of trust (the variable `s.trust` below) is obtained by averaging subjects' responses to the trust items:

```
for (i in names(subset(p, select = g5a:g5k))) p[,
    i] <- ifelse(p[, i] %in% 1:7, p[, i], NA)

trust <- p[, c("g5a", "g5b", "g5d", "g5g", "g5h",
    "g5k")]

s.trust <- apply(X = trust, MARGIN = 1, FUN = mean,
    na.rm = TRUE)
# or
s.trust <- rowMeans(trust, na.rm = TRUE)
```

The trust index — computation

```
##   g5a g5b g5d g5g g5h g5k s.trust
## 1   5   5   6   5   5   6   5.333
## 2   6   6   2   2   6   2   4.000
## 3   5   3   1   3   7   7   4.333
## 4   6   6   7   6   6   6   6.167
## 5   7   7   7   2   2   1   4.333
## 6   7   1   7   6   6   7   5.667
```

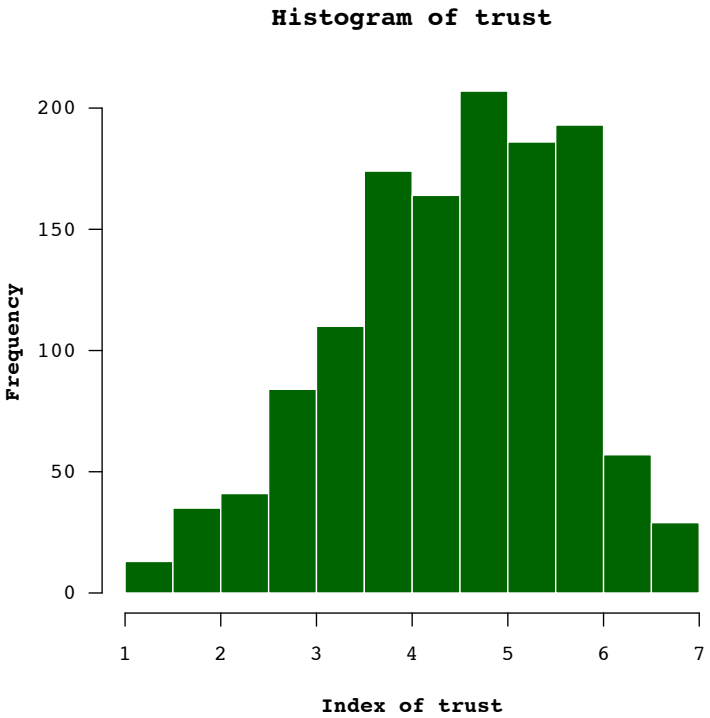
The index of prudence

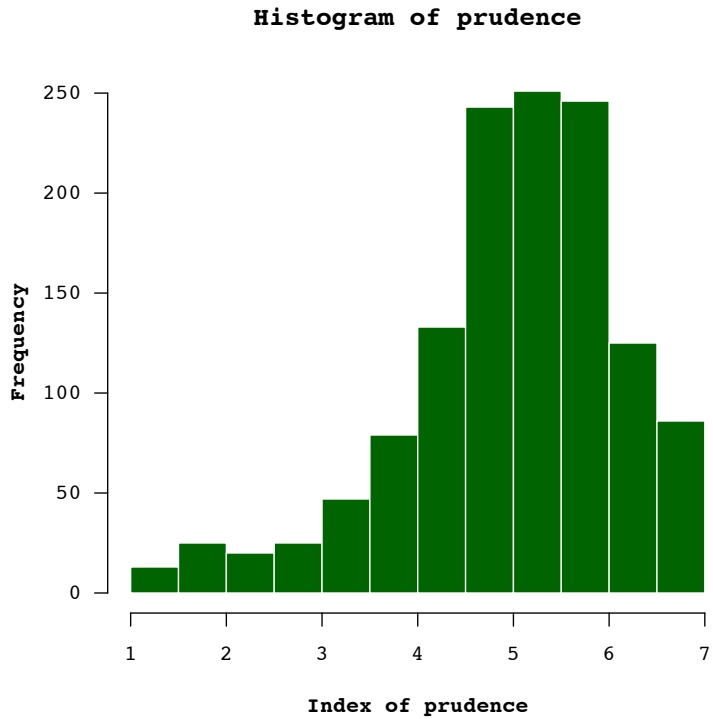
The index of prudence (the variable `s.prud` below) is obtained similarly by averaging subjects' responses to the prudence items:

```
prudence <- p[, c("g5c", "g5e", "g5i", "g5j")]

s.prud <- apply(X = prudence, MARGIN = 1, FUN = mean,
    na.rm = TRUE)
# or
s.prud <- rowMeans(prudence, na.rm = TRUE)
```

Distributions of trust and prudence in the sample





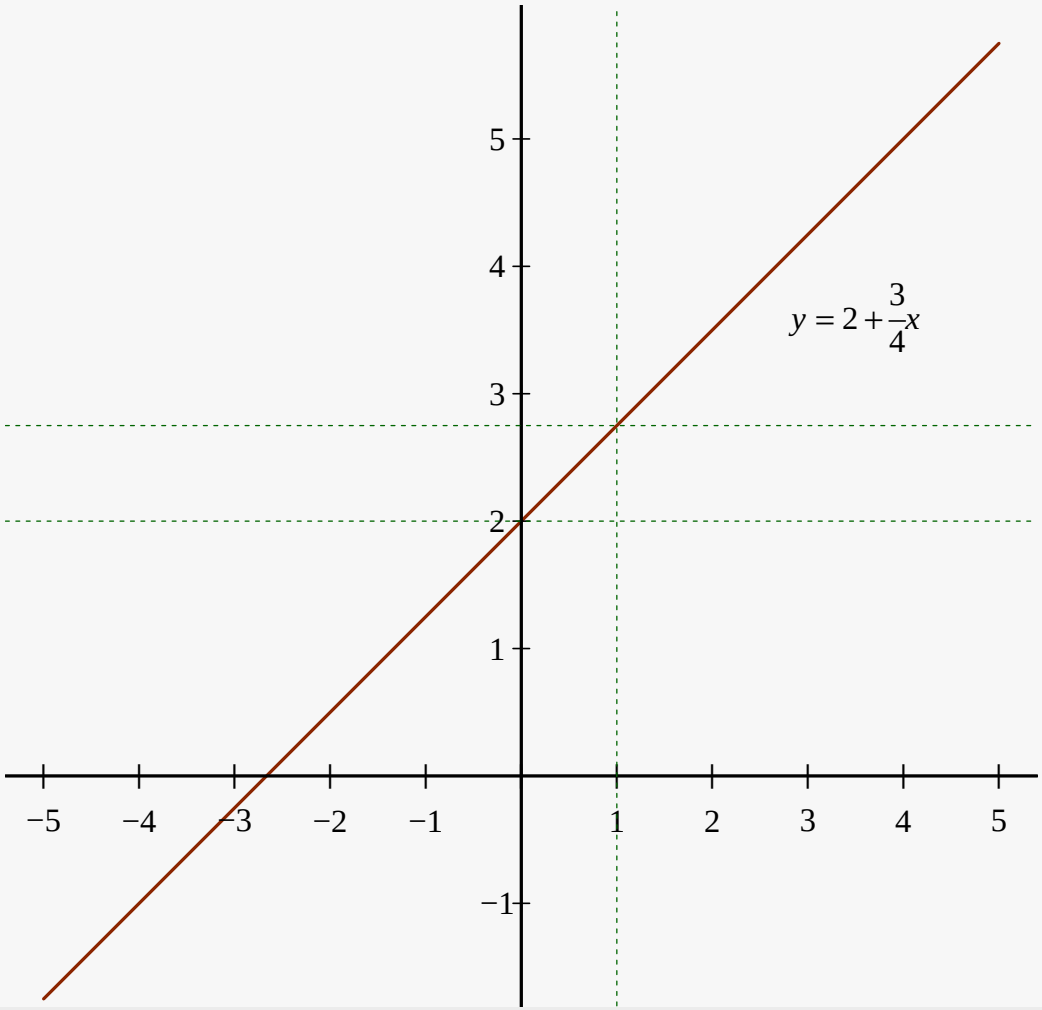
Linear relationship

Linear relationship is one of the form

$$y = a + bx,$$

where a denotes **intercept** — or the point at which the line crosses the verical axis — and b denotes **slope** — or the rate at which y changes given a unit change in x.

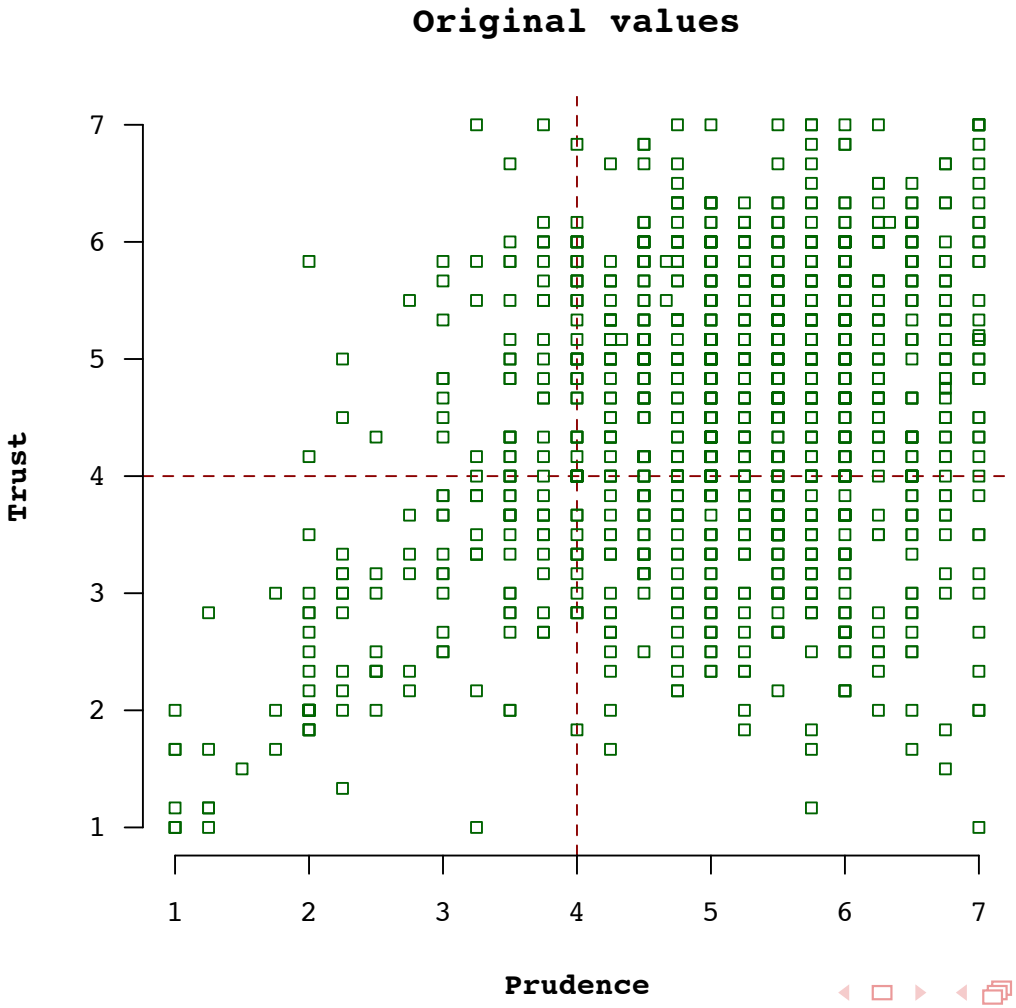
Linear relationship: an illustration



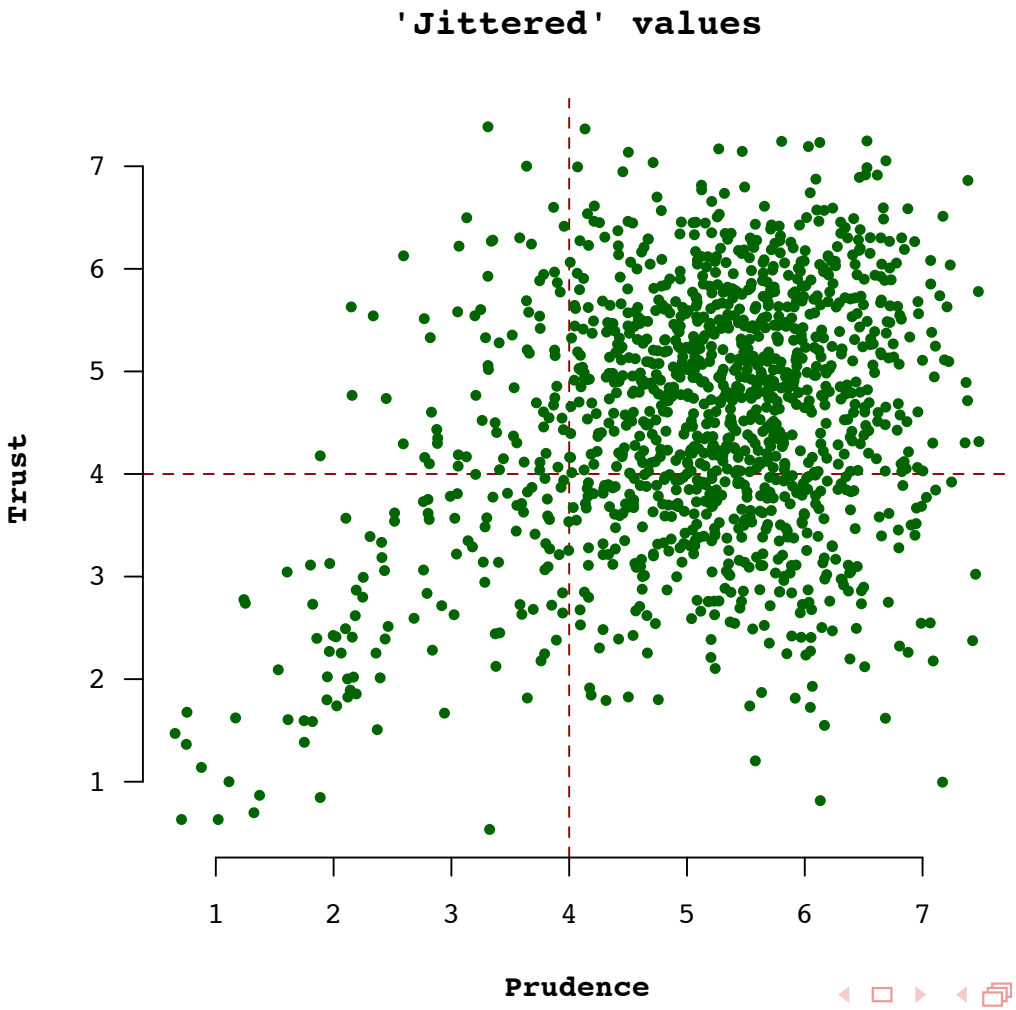
Linear regression

- In linear regression, we seek to describe association between two (or more) variables in terms of a linear equation of the form $y = a + bx$.
- The objective is to find a line that would "fit" the real data as closely as possible.

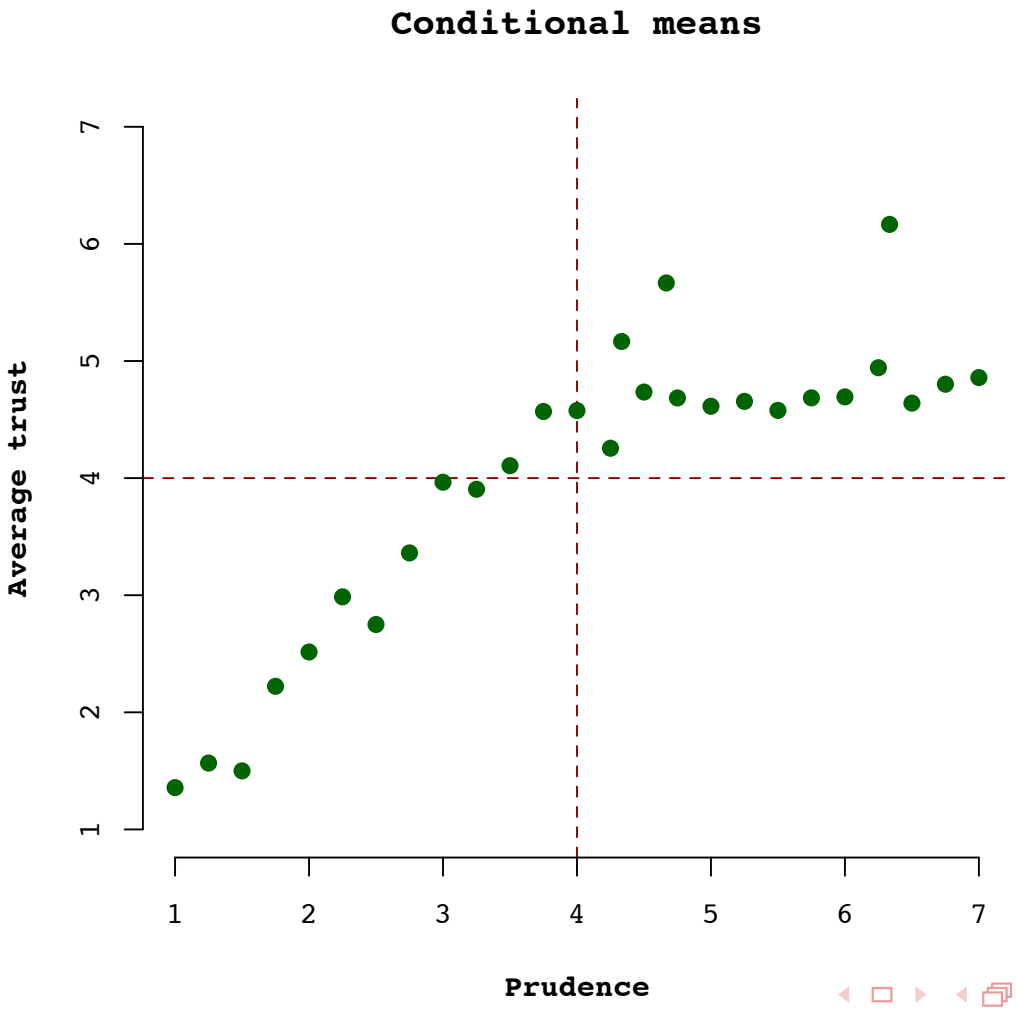
Relationship between trust and prudence



Relationship between trust and prudence (continued)



Relationship between trust and prudence (continued)



The function `lm()`

- Linear regression in R is done using the function `lm()`, which stands for `l`inear `m`odel.
- A critial argument to the function is `formula` which specifies the relationship between the variables of interest.
- In the simplest case, there are only two variables: the dependent one and the independent one.
- As an example, let us consider a model with the index of trust as the dependent and the index of prudence as the independent variable:

$$\texttt{s.trust} = \beta_0 + \beta_1 \texttt{s.prud} + \epsilon.$$

Specification of the model: 1 independent variable

In order to fit the above model to the data in R, one should write:

```
lm(formula = s.trust ~ s.prud)
```

where ~ is used to separate the dependent variable from the independent one(s). In case the variables are in a dataset, its name should be passed on to the funtion `lm()` using the argument `data`:

```
lm(formula = s.trust ~ s.prud, data = DataSet)
```

Extension of the basic model

Suppose we want to add more variables to our model:

Model 1 $\texttt{s.trust} = \beta_0 + \beta_1 \texttt{s.prud} + \beta_2 \texttt{age} + \epsilon$

Model 2 $\texttt{s.trust} = \beta_0 + \beta_1 \texttt{s.prud} + \beta_2 \texttt{age} + \beta_3 \texttt{age}^2 + \epsilon$

Model 3 $\texttt{s.trust} =$
$\beta_0 + \beta_1 \texttt{s.prud} + \beta_2 \texttt{age} + \beta_3 (\texttt{age}^2) + \beta_4 (\texttt{degree}) + \epsilon$

Specification of the regression models: 2 or more independent variables

In order to fit the above models to the data in R, one should write:

```
lm(formula = s.trust ~ s.prud)  # Basic model
lm(formula = s.trust ~ s.prud + age)  # Model 1
lm(formula = s.trust ~ s.prud + age + I(age^2))  #
Model 2
lm(formula = s.trust ~ s.prud + age + I(age^2) +
    degree)  # Model 3
```

The I() operator

The formula

```
y ~ x1 + x2
```

is equivalent to a model with 2 independent variables, whereas the formula

```
y ~ I(x1 + x2)
```

Specifying interactions

The formula

```
y ~ x1 * x2
```

is equivalent to the formula

```
y ~ x1 + x2 + x1:x2
```

where the colon signifies the interaction between the variables x_1 and x_2.

Useful functions

```
mod1 <- lm(formula = s.trust ~ s.prud + age)  #
define the model
summary(mod1)  # Summary of the results
coef(mod1)  # extract the coefficients
predict(mod1)  # extract the predicted values
model.frame(mod1)  # the variables used in the
analysis
anova(mod1, mod0)  # comparison of the models
```