Chocolate and nobel prize

From *Chocolate and the Nobel Prize – a true story?*

http://gforge.se/2012/12/chocolate-and-nobel-prize/

First we get nobel prizes per country by reading the table using readHTMLTable() from the XML package. After that we do some cleaning to the dataset.

```
rm(list=ls())
library(XML)
library(httr)
url <- "http://en.wikipedia.org/wiki/List_of_countries_by_Nobel_laureates_per_capita"
theurl <- htmlParse(rawToChar(GET(url)$content))
tables <- readHTMLTable(theurl)
# delete  Faroe Islands
tables <- tables[[2]][-c(1,31),]

nobel_prizes <- tables
# Clean column names
colnames(nobel_prizes) <-
  gsub(" ", "_",
       gsub("(/|\\[[0-9]+\\])", "",
            gsub("\n", " ", colnames(nobel_prizes))
       )
    )

# Delete those that aren't countries and thus lack rank
nobel_prizes$Rank <- as.numeric(as.character(nobel_prizes$Rank))
nobel_prizes <- subset(nobel_prizes, is.na(Rank) == FALSE)

# Create Country
nobel_prizes$Country = nobel_prizes$Entity

# Clean the country names
nobel_prizes$Country <-
  gsub("[^a-zA-Z ]", "", nobel_prizes$Entity)

# Clean the loriates variable
nobel_prizes$Laureates_10_million <-
  as.numeric(as.character(nobel_prizes$Laureates_10_million))
```

The next part is slightly trickier since we need to translate german country names to match the nobel prize data.

```
# Translation from German to English
url <- "http://german.about.com/library/blnation_index.htm"
theurl <- htmlParse(rawToChar(GET(url)$content))
tables <- readHTMLTable(theurl)

translate_german <- tables[[1]]
translate_german <- translate_german[3:NROW(translate_german), 1:2]
colnames(translate_german) <- c("English", "German")
translate_german$German <-
```

```
    gsub("([ ]+(f|pl)\\.$|\\([[:alnum:] -]+\\))", "", translate_german$German)

# Get the consumption from a German list
url <- "http://www.theobroma-cacao.de/wissen/wirtschaft/international/konsum"
theurl <- htmlParse(rawToChar(GET(url)$content))

tables <- readHTMLTable(theurl)

german_chocolate_data <- tables[[1]][2:NROW(tables[[1]]), ]
names(german_chocolate_data) <- c("Country", "Chocolate_consumption")
german_chocolate_data$Country <-
    gsub("([ ]+f\\.$|\\([[:alnum:] -]+\\))", "", german_chocolate_data$Country)

library(sqldf)

sql <- paste0("SELECT gc.*, tg.English as Country_en",
              " FROM german_chocolate_data AS gc",
              " LEFT JOIN translate_german AS tg",
              " ON gc.Country = tg.German",
              " OR gc.Country = tg.English")
german_chocolate_data <- sqldf(sql)

german_chocolate_data$Country <-
    ifelse(is.na(german_chocolate_data$Country_en),
    german_chocolate_data$Country,
    german_chocolate_data$Country_en)

german_chocolate_data$Country_en <- NULL
german_chocolate_data$Chocolate_consumption_tr <- NA
for (i in 1:NROW(german_chocolate_data)) {
    number <- as.character(german_chocolate_data$Chocolate_consumption[i])
    if (length(number) > 0) {
        m <- regexpr("^([0-9]+,[0-9]+)", number)
        if (m > 0) {
            german_chocolate_data$Chocolate_consumption_tr[i] <-
                as.numeric(
                    sub(",", ".", regmatches(number, m))
                )
        } else {
            m <- regexpr("\\(([0-9]+,[0-9]+)", number)

            if (m > 0)
                german_chocolate_data$Chocolate_consumption_tr[i] <-
                    as.numeric(
                        sub("\\(", "",
                            sub(",", ".", regmatches(number, m))
                        )
                    )
        }
    }
}

sql <- paste0("SELECT np.*, gp.Chocolate_consumption_tr AS choc",
```

```
              " FROM nobel_prizes AS np",
              " LEFT JOIN german_chocolate_data AS gp",
              " ON gp.Country = np.Country")
nobel_prizes <- sqldf(sql)
```

Chocalate data for Switzerland

```
nobel_prizes$choc[nobel_prizes$Country == "Switzerland"] <- 11.9
```

This leaves us with 18 countries that have chocolate data.

```
sum(complete.cases(nobel_prizes))
```

```
## [1] 18
```

```
nobel_prizes_cc = nobel_prizes[complete.cases(nobel_prizes),]
nobel_prizes_cc
```

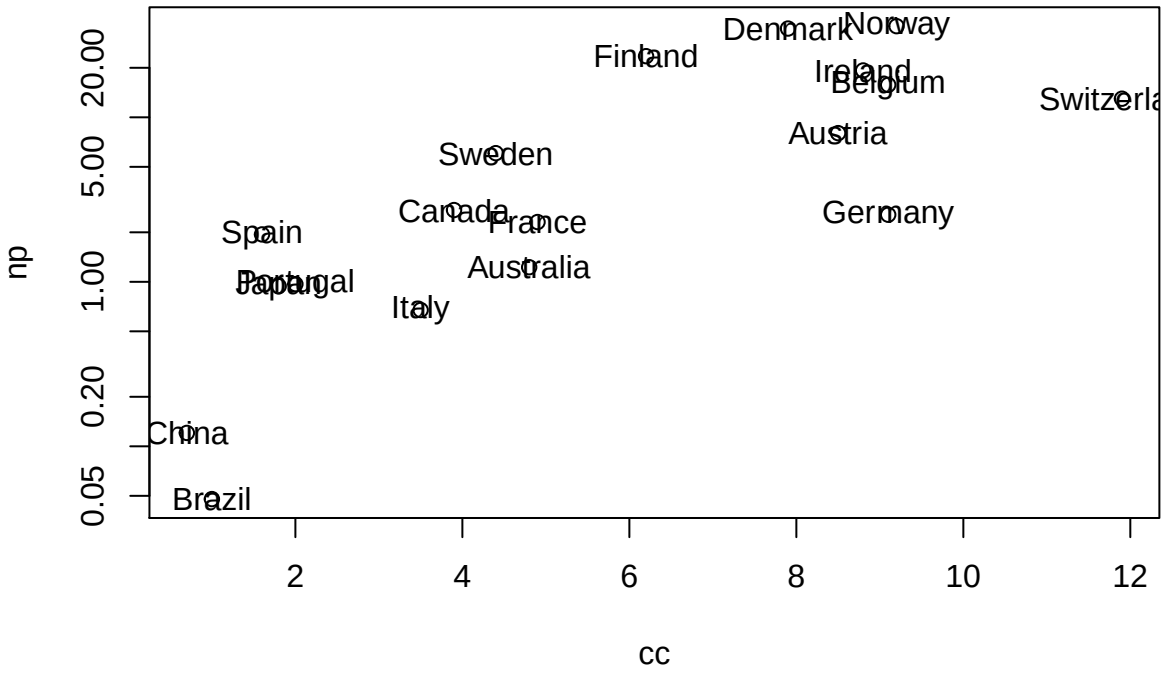
```
##    Rank      Entity Nobel_laureates_(2017) Population_(2017)[note_1]
## 3     3 Switzerland                     21                 8,476,005
## 4     4     Austria                     18                 8,735,453
## 5     5      Sweden                     17                 9,910,701
## 7     7     Denmark                      9                 5,773,551
## 8     8      Norway                      8                 5,305,383
## 10   10     Germany                     91                82,114,524
## 16   16      France                     37                64,979,548
## 17   17      Canada                     20                36,624,199
## 18   18     Finland                      3                 5,523,231
## 19   19     Belgium                      6                11,429,336
## 22   22   Australia                     11                24,450,561
## 23   23     Ireland                      2                 4,671,567
## 28   28       Italy                     13                59,359,900
## 29   29       Japan                     22               127,484,450
## 35   35    Portugal                      1                10,329,506
## 38   38       Spain                      2                46,354,321
## 48   48      Brazil                      1               209,288,278
## 50   50       China                      5             1,409,517,397
##    Laureates_10_million     Country choc
## 3                24.776 Switzerland 11.9
## 4                20.606     Austria  8.5
## 5                17.153      Sweden  4.4
## 7                15.588     Denmark  7.9
## 8                15.079      Norway  9.2
## 10               11.082     Germany  9.1
## 16                5.694      France  4.9
## 17                5.461      Canada  3.9
## 18                5.432     Finland  6.2
## 19                5.250     Belgium  9.1
## 22                4.499   Australia  4.8
## 23                4.281     Ireland  8.8
## 28                2.190       Italy  3.5
## 29                1.758       Japan  1.8
## 35                0.968    Portugal  2.0
## 38                0.431       Spain  1.6
## 48                0.048      Brazil  1.0
## 50                0.035       China  0.7
```

Lets plot Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population (in the log scale)

```
cc = nobel_prizes_cc$choc
paesi = nobel_prizes_cc$Country
pop_temp = as.character(nobel_prizes_cc$`Population_(2017)[note_1]`)
pop = as.numeric(gsub(",", "", pop_temp))
np = 10^7*as.numeric(nobel_prizes_cc$`Nobel_laureates_(2017)`)/pop
lnp = log(np)
cor(lnp,cc)
```

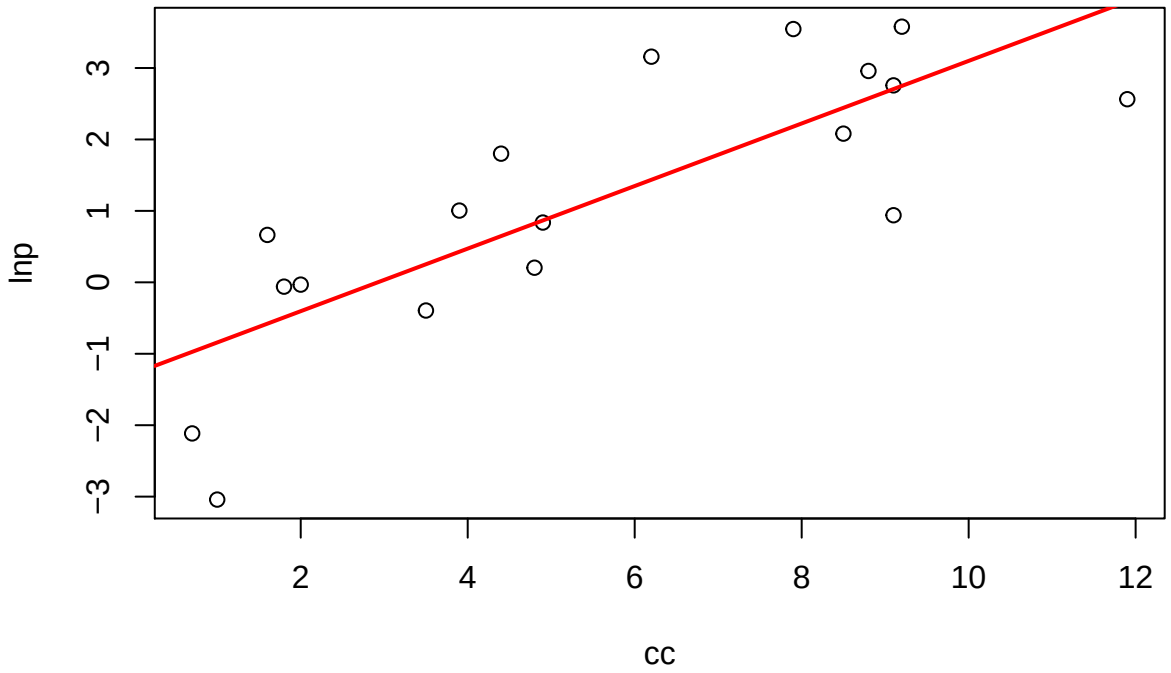
```
## [1] 0.8018757
```

```
plot(cc, np, log="y")
text(cc, np, labels=paesi )
```



Without controlling for other factors:

```
plot(cc, lnp)
abline(lm(lnp ~ cc), lwd=2, col=2)
```




```
summary(lm(lnp ~ cc))
```

```
##
## Call:
## lm(formula = lnp ~ cc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1992 -0.6400  0.2117  0.7662  1.7234
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.27986    0.52569  -2.435    0.027 *
## cc           0.43790    0.08157   5.368 6.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.153 on 16 degrees of freedom
## Multiple R-squared:  0.643,  Adjusted R-squared:  0.6207
## F-statistic: 28.82 on 1 and 16 DF,  p-value: 6.276e-05
```

What else might explain this relationship? How do we identify confounders? How do we control for them?

Best case: randomize treatment. In a randomized experiment, there should be no confounding variables.

In many social science settings, RCT is impossible: subjects (e.g. countries, individuals) choose their own treatment.

The Nordic countries rank at the top of the graph on the right and they are known to rank high on per capita income.

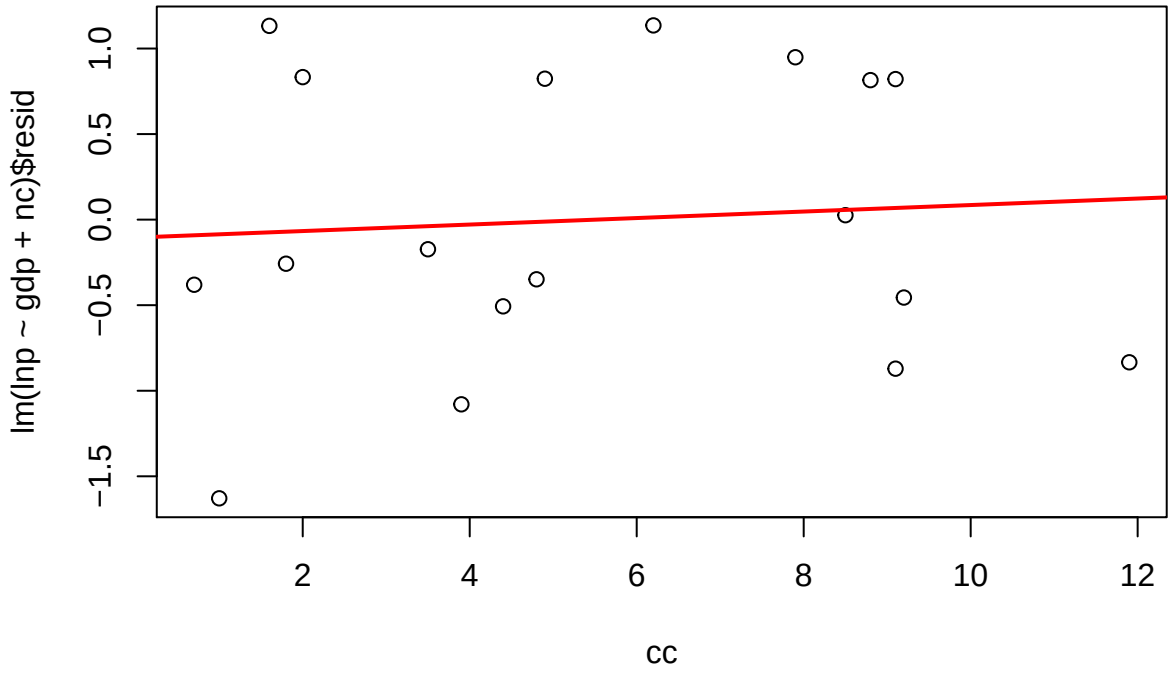
Controlling for Nordic countries and GDP?

```
library("countrycode")
countryc = countrycode(paesi, origin="country.name", destination="wb")
library("wbstats")
```

```
new_wb_cache <- wbcache()
wbsearch("gdp.*capita.*US\\$", cache = new_wb_cache)
```

```
##         indicatorID                          indicator
## 7034 NY.GDP.PCAP.KD GDP per capita (constant 2010 US$)
## 7036 NY.GDP.PCAP.CD       GDP per capita (current US$)
```

```
indicator <- wb(indicator = c("NY.GDP.PCAP.KD"), startdate=2010, enddate=2010, country = countryc)
gdp <- indicator$value[match(paesi, indicator$country)]
nc = paesi %in% c("Sweden","Denmark","Norway","Canada", "Finland", "Ireland","Austria","Switzerland", "
plot(cc,lm(lnp ~ gdp + nc)$resid)
abline(lm( lm(lnp ~ gdp + nc)$resid ~ cc ), lwd=2,col=2 )
```




```
summary(lm(lnp ~ cc + gdp + nc))
```

```
##
## Call:
## lm(formula = lnp ~ cc + gdp + nc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6181 -0.3953 -0.1695  0.7123  1.2147
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.979e+00  5.568e-01  -3.554  0.00318 **
## cc           7.449e-02  1.296e-01   0.575  0.57469
## gdp          4.289e-05  1.762e-05   2.435  0.02889 *
## nc           1.479e+00  7.339e-01   2.016  0.06346 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9258 on 14 degrees of freedom
```

```
## Multiple R-squared:  0.7986, Adjusted R-squared:  0.7554
## F-statistic:  18.5 on 3 and 14 DF,  p-value: 3.833e-05
```