Sample Rmarkdown

Asako Mikami

February 11, 2019

Abstract

This document demonstrates how to weave and execute R codes within an Rmarkdown document with `knitr`. The first section runs a simulation of simultaneous relationship. The second section runs a logistic regression model and displays the result table using the `xtable` and `stargazer` package. The output is available as both `.pdf` and `.html` documents. To render a `pdf` document, run `rmarkdown::render('sample-ta.Rmd', 'pdf_document', params = list(input_type = 'latex'))`. To render an `.html` document, run `rmarkdown::render('sample-ta.Rmd', 'html_document', params = list(input_type = 'html'))`.

```r
# load library
x <- c("dplyr", "purrr", "stargazer", "RColorBrewer", "knitr",
       "xtable")
lapply(x, library, character.only = TRUE, quietly = TRUE)

# set up color palette to be used in this document
color <- brewer.pal(3, "Paired")
```

```r
# For this document, I want to show most of the R source code,
# so I am setting the global option `echo = TRUE`.
#------------------
# I want to save my plot outputs as png and pdf files with
# transparent backgroun din the `fig` folder:
# `fig.path = "fig/", dev = c("png", "pdf"),
# dev.args = list(bg = "transparent")`
#------------------
# When the plot is displayed in the document, I want it to be
# aligned in the center, so I am setting `fig.align = "center"`.
#------------------
knitr::opts_chunk$set(echo = TRUE,
                      fig.path = "fig/",
                      dev = c("png", "pdf"),
                      dev.args = list(bg = "transparent"),
                      fig.align = "center")
# --------------
# All of these global options can be overwritten in each chunk.
# For example, in section 2, I am hiding some chunks by setting
# `echo = FALSE` in the chunk header.
```

Simulating simultaneity

Let's do some simulation of simultaneity.[1] Suppose we have the following data generating process:

$$
\begin{aligned}
&Y = \beta X + \epsilon_1, \quad \epsilon_1 \sim N(0, \sigma^2) \\
&X = \alpha Y + \epsilon_2, \quad \epsilon_2 \sim N(0, \tau^2) \\
&\epsilon_1 \perp \epsilon_2
\end{aligned}
$$

[1] This example is adapted from Haavelmo (1943) and Bellemare, Masaki, and Pepinsky (2017).

where α, β are real constants and σ^2, τ^2 are positive constants. Solving this system of equations, we express Y and X free of each other.

$$
\begin{aligned}
X &= \frac{\alpha\epsilon_1 + \epsilon_2}{1 - \alpha\beta} \\
Y &= \frac{\epsilon_1 + \beta\epsilon_2}{1 - \alpha\beta}
\end{aligned}
$$

This shows that X and Y are multivariate normal with mean zero. Without any loss of generality, we provide the proof for $\mathrm{E[Y] = 0}$.

$$
\begin{aligned}
\mathrm{E[Y]} &= \mathrm{E}[\beta \mathrm{X} + \epsilon_1] \\
&= \mathrm{E}\Big[\beta\Big(\frac{\alpha\epsilon_1 + \epsilon_2}{1 - \alpha\beta}\Big) + \epsilon_1\Big] \\
&= \frac{\beta}{1 - \alpha\beta}\mathrm{E}\Big[\alpha\epsilon_1 + \epsilon_2\Big] + \mathrm{E}[\epsilon_1] \\
&= \frac{\beta}{1 - \alpha\beta}\Big(\alpha\mathrm{E}[\epsilon_1] + \mathrm{E}[\epsilon_2]\Big) + \mathrm{E}[\epsilon_1] \\
&= 0 \quad \text{because } \mathrm{E}[\epsilon_\mathrm{i}] = 0 \text{ for } i = 1, 2
\end{aligned}
$$

Under this data generating process, the least square estimate $\hat{\beta}$ of linear regression model, $\mathrm{E[Y|X]} = \beta\mathrm{X}$, will be biased.

$$
\begin{aligned}
\hat{\beta} &= \frac{\mathrm{Cov(X, Y)}}{\mathrm{Var(X)}} \quad \text{because } X \text{ is also random variable} \\
&= \frac{\mathrm{E[XY]}}{\mathrm{Var(X)}} \quad \text{because } \mathrm{Cov(X, Y) = E[XY] - E[X]E[Y] = E[XY]} \\
&= \frac{\mathrm{E}\Big[\Big(\frac{\alpha\epsilon_1 + \epsilon_2}{1 - \alpha\beta}\Big)\Big(\frac{\epsilon_1 + \beta\epsilon_2}{1 - \alpha\beta}\Big)\Big]}{\mathrm{Var}\Big(\frac{\alpha\epsilon_1 + \epsilon_2}{1 - \alpha\beta}\Big)} \\
&= \frac{\mathrm{E}[(\alpha\epsilon_1 + \epsilon_2)(\epsilon_1 + \beta\epsilon_2)]}{\alpha^2\mathrm{Var}(\epsilon_1) + \mathrm{Var}(\epsilon_2)} \quad \text{because } \epsilon_1 \perp \epsilon_2 \\
&= \frac{\mathrm{E}[\alpha\epsilon_1^2 + (\alpha\beta + 1)\epsilon_1\epsilon_2 + \beta\epsilon_2^2]}{\alpha^2\mathrm{Var}(\epsilon_1) + \mathrm{Var}(\epsilon_2)} \\
&= \frac{\alpha\mathrm{E}(\epsilon_1^2) + (\alpha\beta + 1)\mathrm{E}[\epsilon_1]\mathrm{E}[\epsilon_2] + \beta\mathrm{E}(\epsilon_2)^2}{\alpha^2\mathrm{Var}(\epsilon_1) + \mathrm{Var}(\epsilon_2)} \\
&= \frac{\alpha\mathrm{Var}(\epsilon_1) + \beta\mathrm{Var}(\epsilon_2)}{\alpha^2\mathrm{Var}(\epsilon_1) + \mathrm{Var}(\epsilon_2)} \quad \text{because } \mathrm{Var}(\epsilon_\mathrm{i}) = \mathrm{E}[\epsilon_\mathrm{i}^2] - \mathrm{E}[\epsilon_\mathrm{i}]^2 = \mathrm{E}[\epsilon_\mathrm{i}^2] \text{ for } i \in \{1, 2\} \\
&= \frac{\alpha\sigma^2 + \beta\tau^2}{\alpha^2\sigma^2 + \tau^2} \quad \text{which is generally different from } \beta
\end{aligned}
$$

```r
set.seed(02052019)

# set the parameters
param <- list(beta = sample(2:4, 1),
              alpha = sample(7:9, 1)*-1,
```

```
              sigma2 = sample(c(0.3, 1, 1.5), 1),
              tau2 = sample(c(0.2, 1, 1.4), 1),
              n = 500
        )

run_ols <- function(param){
        #---------------------
        # Generates X and Y based on the true data
        # generating process, using the parameters
        # listed in `param`.
        # Output is ols coefficient estimates.
        #---------------------
        # param (a list of parameters)
        #---------------------
        e1 <- rnorm(param$n, mean = 0, sd = param$sigma2)
        e2 <- rnorm(param$n, mean = 0, sd = param$tau2)
        X <- (param$alpha * e1  + e2)/(1 - param$alpha * param$beta)
        Y <- (e1 + param$beta * e2)/(1 - param$alpha * param$beta)
        ols <- lm(Y ~ X - 1) # no intercept
        return(coef(ols))
}

# run simulation and store the result as dataframe
sim <- 9999
library(purrr)
result <- map(seq(sim), ~ run_ols(param)) %>%
        map_dfr(~ as.data.frame(t(as.matrix(.))))
```

Let's plot the sampling distribution of ols estimates for β.[2]

```
# plot the sample distribution of beta.hat
hist(result$X, col = color[1],
     xlab = expression(hat(beta)),
     main = as.expression(bquote(beta==.(param$beta)~", "~
                                  alpha==.(param$alpha)~", "~
                                sigma^2==.(param$sigma2)~", "~
                                tau^2==.(param$tau2)))
     )
```

[2] See this R-blogger post for more details on how to write mixed expression in plot captions and titles.

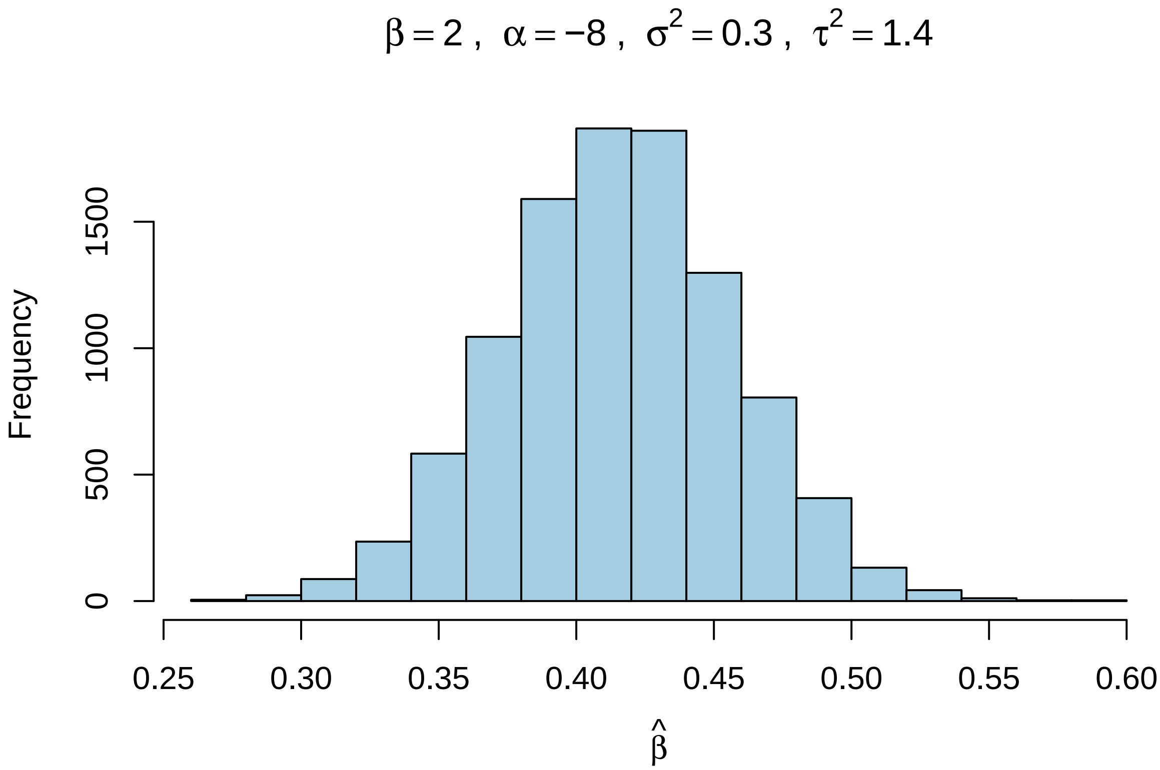

Figure 1: Sampling distribution of OLS estiamtes

Let's simulate across $\beta \in \{0, \pm 0.5, \pm 1.0, \pm 1.5, \pm 2.0\}$, and plot the mean squared error, $(\mathrm{E}[\hat{\beta}] - \beta)^2$, against β.

```
beta <- seq(-2, 2, by = 0.5) # simulation values for beta
sim <- 100
# create a list of parameter lists
param_list <- map(beta, ~update_list(param, beta = .))

run_simulate <- function(param){
        #---------------------------
        # Simulate ols regression for `sim` number of
        # times for each `beta` value.
        # Obtain MSE(beta) of each simulation sample.
        #----------------------------
        # param (a list of parameters)
        #----------------------------

        # generate a sample of ols estimates
        sample <- seq(sim) %>% map(~run_ols(param)) %>%
                map_df(~as.data.frame(.))
        # obtain MSE
        mse <- sample %>% map_df(~(mean(.)-param$beta)^2)
        return(mse)
}

# simulate over `param_list`
ols_mse <- map(param_list, ~run_simulate(.)) %>%
        map_df(~as.data.frame(t(as.matrix(.))))
```

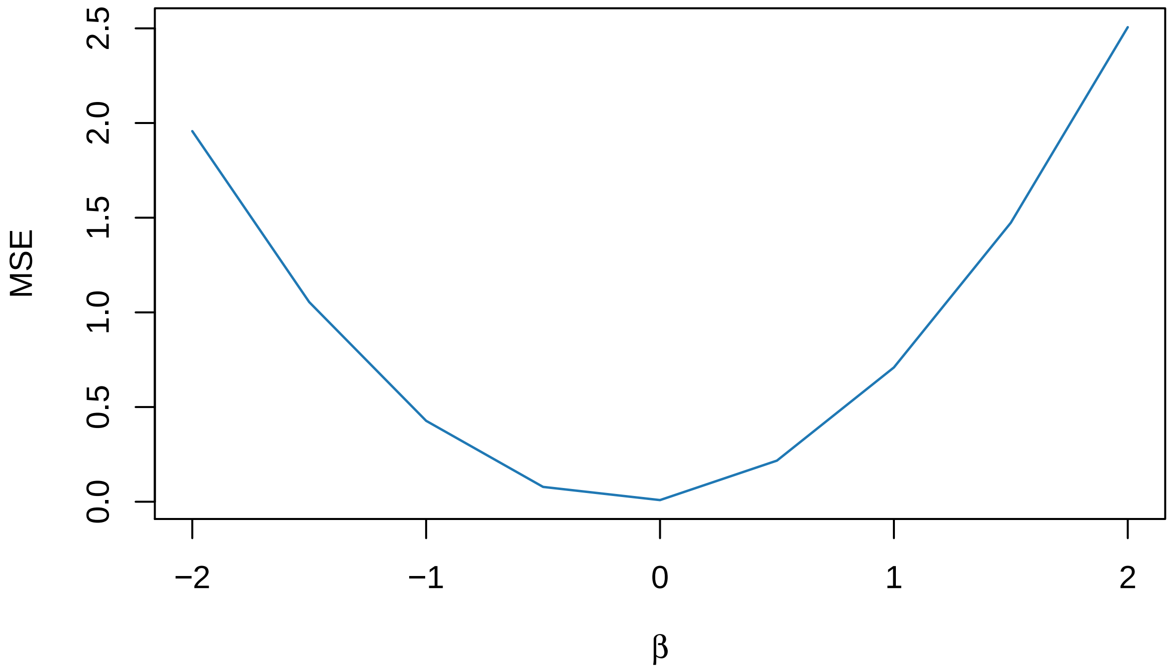

Figure 2: Mean squared error of OLS estimator

```r
df <- cbind(beta, ols_mse) # bind with `beta` vector
names(df) <- c("beta", "MSE") # name the columns
```

```r
# plot MSE against `beta`
plot(MSE ~ beta, data = df, type = "l",
     col = color[2], lwd=1.2,
     xlab = expression(beta))
```

Bank marketing data set

We have data from a Portugese bank's telemarketing campaign where the outcome of interest is whethe a client subscribed to a term deposit at the end of the campaign:[3]

$$Y_i = \begin{cases} 1 & \text{if client } i \text{ subscribed} \\ 0 & \text{otherwise} \end{cases}$$

The code for this section is written in `script/bank-marketing.R` and is printed out in the Appendix.[4]

We want to model π_i, the probability that $Y_i = 1$, given a data matrix X_i:

$$\text{logit}(\pi_\text{i}) = \log \frac{\pi_\text{i}}{1 - \pi_\text{i}} = \text{X}_\text{i}^\text{T} \beta.$$

The data matrix consists of input variables that can be divided into three main groups[5]:

1. client's personal attributes such as her job, marital status, and education;

[3]The data set is available for download at UCI's Machine Learning Repository. Details of the dataset are available in Moro, Cortez, and Rita (2014).

[4]See this page from https://yihui.name/knitr/ and the linked examples to learn more about how to externalize codes.

[5]For details on each variable, see the data description `.txt` file.

2. attirbutes related to the last contact the client received from the campaign such as its month and day of the week;
3. macroeconomic context attributes such as CPI, consumer confidence index, and Euribo 3-month index.

Let's assess the relevance of these three groups of variables by Chi-square ANOVA test. Below I present the ANOVA output in a table created by `xtable::xtable()` function.[6]

% latex table generated in R 3.5.2 by xtable 1.8-3 package % Mon Feb 11 23:14:21 2019

	Residual df	Deviance	Diff. in deviance	Regresssion df	p-value
Null model	595.00	691.39			
Model 1: Null + client attributes	569.00	631.48	26.00	59.91	0.00
Model 2: model 1 + last-contact var.	554.00	476.71	15.00	154.78	0.00
Model 3: model 2 + macroeconomic var.	550.00	410.19	4.00	66.52	0.00

Table 1: ANOVA table comparing nested models

The ANOVA table indicates that all three groups contain some variables that may be relevant for predicting the . Since we do not have a theory on the data-generating process, we run a stepwise variable selection to narrow down the input variables. The final model is summarized in the table below created by `stargazer::stargazer()`.[7]

Table 2: Logistic regression model based on backwards stepwise variable selection

	Dependent variable:
	telemarketing outcome
contact–telephone	−0.953 (0.676)
contact–Aug	−0.036 (0.806)
contact–Dec	−0.702 (1.255)
contact–July	−1.294 (0.878)
contact–June	0.752 (0.765)
contact–Mar	0.895 (0.928)
contact–May	0.209 (0.553)
contact–Nov	−1.462 (1.241)
contact–Oct	−1.564 (1.350)
contact–Sep	−2.049 (1.368)
duration	0.006^{***} (0.001)
number of contacts made	−0.191 (0.122)
previous outcome	1.885^{***} (0.284)
CPI	-2.246^{**} (0.883)
Euribor 3mo. index	2.682^{**} (1.287)
number of employees	-0.061^{***} (0.021)
Constant	512.289^{***} (187.461)
Observations	596
Log Likelihood	−193.424
Akaike Inf. Crit.	420.847
Note:	*p<0.1; **p<0.05; ***p<0.01

[6] There are other options such as `pander::pander()` and `knitr::kable()` that can create tables out of `R` output. See this manual for how to use `xtable()`.

[7] Though the output format is `html`, this manual by Jake Russ is helpful for learning how to use `stargazer()`. For LaTeX-specific manual, see this manual by Marek Hlavac.

Reference

```
## By default, the reference section appears at the end of
## the document.
## If you want to put the reference section before
## the appendix, insert the following html line where
## you want the reference section to show:
## <div id="refs"></div>
```

Bellemare, Marc F, Takaaki Masaki, and Thomas B Pepinsky. 2017. "Lagged Explanatory Variables and the Estimation of Causal Effect." *The Journal of Politics* 79 (3). University of Chicago Press Chicago, IL: 949–63.

Haavelmo, Trygve. 1943. "The Statistical Implications of a System of Simultaneous Equations." *Econometrica, Journal of the Econometric Society*. JSTOR, 1–12.

Moro, Sérgio, Paulo Cortez, and Paulo Rita. 2014. "A Data-Driven Approach to Predict the Success of Bank Telemarketing." *Decision Support Systems* 62 (June). Elsevier: 22–31.

Code Appendix

```r
####------------------------
### This script downloads and fits
### logistic regression model on
### Bank Marketing Data Set from UCI Machine Learning Repos.
###-------------------------

## @knitr download_zip
# download .zip file and extract to "data" folder
temp <- tempfile()
base_url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/"
download.file(url = paste0(base_url, "00222/bank-additional.zip"),
                destfile = temp)
unzip(temp, exdir = "data")
unlink(temp)

## @knitr load_data
# open "bank-additional.csv"
path <- file.path("data", "bank-additional")
bank <- read.csv(file.path(path, "bank-additional.csv"),
                 header = TRUE, sep = ";")
# change month labels to numeric
levels(bank$month) <- c("4", "8", "12", "7", "6", "3", "5", "11",
                        "10", "9")
# recode 'poutcome'
bank$poutcome <- ifelse(bank$poutcome == "nonexistent", NA,
                ifelse(bank$poutcome == "success", 1, 0)) %>%
        as.factor()
# remove na rows
bank <- na.omit(bank)

## @knitr model_comparison
# base model
base <- glm(y ~ 1, data = bank, family = "binomial")
```

```r
# nested model: remove client attributes
nest_client <- update(base,
                ~ . + age  + job + marital + education +
                 default + housing + loan )
# nested model: remove last contact attributes
nest_last <- update(nest_client,
                    ~ . + contact + month + day_of_week + duration)
# nested model: remove macroeconomic context attributes
nest_macro <- update(nest_last,
                     ~ . + emp.var.rate + cons.price.idx +
                             euribor3m + nr.employed)
## @knitr anova
anova.out <- anova(base, nest_client, nest_last, nest_macro,
                   test = "Chisq")
colnames(anova.out) <- c("Residual df",
                         "Deviance",
                         "Diff. in deviance",
                         "Regresssion df",
                          "p-value")
rownames(anova.out) <- c("Null model",
                         "Model 1: Null + client attributes",
                         "Model 2: model 1 + last-contact var.",
                         "Model 3: model 2 + macroeconomic var.")
gen_table <- function(.){
        #--------------------
        # Format `print.xtable()` output depending
        # on the type of table to produce.
        #--------------------
        # . (chr, either "latex" or "html")
        #-------------------
        require(xtable)
        anova.tab <- xtable(anova.out, comment = FALSE,
                            caption = "ANOVA table comparing nested models")
        if (.== "html"){
                print.xtable(anova.tab, type = .,
                html.table.attributes = 'align="center",
                        rules = "row",
                        width = 80%, frame = "below"')
        }
        if (.=="latex"){
                print.xtable(anova.tab, type = .,
                             floating = TRUE,
                             table.placement = "h!")
        }
}


## @knitr step_selection
# full model
full <- glm(y ~ age + job + marital + education +
                    default + housing + loan +
                    contact + month + day_of_week +
                    duration + campaign + pdays + previous +
```

```r
                    poutcome + emp.var.rate + cons.price.idx +
                    euribor3m + nr.employed,
            data = bank, family = "binomial")
# backwards stepwise regression
step.out <- step(full, direction = "backward",
                 trace = 0) # do not print output to console
```

Session Info

```r
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.3
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] xtable_1.8-3       knitr_1.20         RColorBrewer_1.1-2
## [4] stargazer_5.2.2    purrr_0.2.5        dplyr_0.7.6
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.0       crayon_1.3.4     digest_0.6.18    assertthat_0.2.0
##  [5] R6_2.3.0         magrittr_1.5     evaluate_0.11    pillar_1.3.1
##  [9] rlang_0.3.1      stringi_1.2.4    bindrcpp_0.2.2   rmarkdown_1.11
## [13] tools_3.5.2      stringr_1.3.1    glue_1.3.0       yaml_2.2.0
## [17] compiler_3.5.2   pkgconfig_2.0.2  htmltools_0.3.6  tidyselect_0.2.4
## [21] bindr_0.1.1      tibble_2.0.1
```