Econ 21410 - Example Knitr File

John Eric Humphries[*]

April 5, 2014

This file provides a working example of using Knitr where all of your R code is saved in an external file rather than ”in line” in the .Rnw file. To generate this pdf, you need the ”example_Knitr.R” and ”example_Knitr.Rnw” files together in the same directory.

1 Running the first bit of code.

Suppose that the first thing I want to do is run some code to ”set up” my data, for example:

```r
# Section 0: setup ========================

# setwd('/mnt/ide0/home/johneric/sbox/projects/neighborhoodVision/')
rm(list = ls())  # Clear the workspace
set.seed(907)

library(xtable)  # A simple way to make latex tables


# ================================
```

Note that t ran everything between ”## @knitr code_part0” and ”## @knitr code_part1”.

But maybe, I do not want to print this code to the pdf, but just run it, en that case, I would set ”echo” and ”eval” to false like this (note you will need to open up the example_Knitr.Rnw file to see the actual code):

(which runs the same code as above, but does not display the code or the output.)

2 Generate Data

Quickly generating some data (and displaying the code)

[*] Please email johneric@uchicago.edu and obrowne@uchicago.edu if you have questions.

```
# Section 1: Generating Data =================================

a <- runif(100)
b <- runif(100)
c <- 12.345

x <- matrix(rnorm(200, mean = 1), 100, 2)
epsilon <- rnorm(100)
y <- 1 + 2 * x[, 1] - 3 * x[, 2] + epsilon



# ================================
```

3 Performing some calculations.

Next I want to show and actually calculate some numbers in code_part1 which I can do with

```
# Section 2: stuff with data =================================

mean(a + b)


## [1] 0.9556


obs <- length(a)
class(b)


## [1] "numeric"




# ================================
```

If I only wanted to show the code I could do:

```
# Section 2: stuff with data =================================

mean(a + b)
obs <- length(a)
class(b)


# ================================
```

If I only wanted to show results I could do:

```
## [1] 0.9556
## [1] "numeric"
```

I can also return specific numbers in the middle of the text such as $c = 12.345$ (see the .Rnw file to see how this is done).

4 Regression and making a table

Next, I will run some regression code and print a summary of the regression results to screen.

```
# Section 3: stuff with data =================================

# saving regression results
y.reg <- lm(y ~ x)

# displaying regression results
summary(y.reg)


##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7297 -0.5480 -0.0008  0.6520  2.0676
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9801     0.1724    5.69  1.4e-07 ***
## x1            2.0179     0.1014   19.90  < 2e-16 ***
## x2           -3.0576     0.0978  -31.27  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.994 on 97 degrees of freedom
## Multiple R-squared:  0.932,Adjusted R-squared:  0.93
## F-statistic:  660 on 2 and 97 DF,  p-value: <2e-16



# making table of regression results
```

But what if I wanted to show the results as a nice latex table? The package ”xtable” will do this for us quickly with the command:

```r
xtable(y.reg, caption = "A TABLE OF REGRESSION RESULTS")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9801	0.1724	5.69	0.0000
x1	2.0179	0.1014	19.90	0.0000
x2	-3.0576	0.0978	-31.27	0.0000

Table 1: A TABLE OF REGRESSION RESULTS

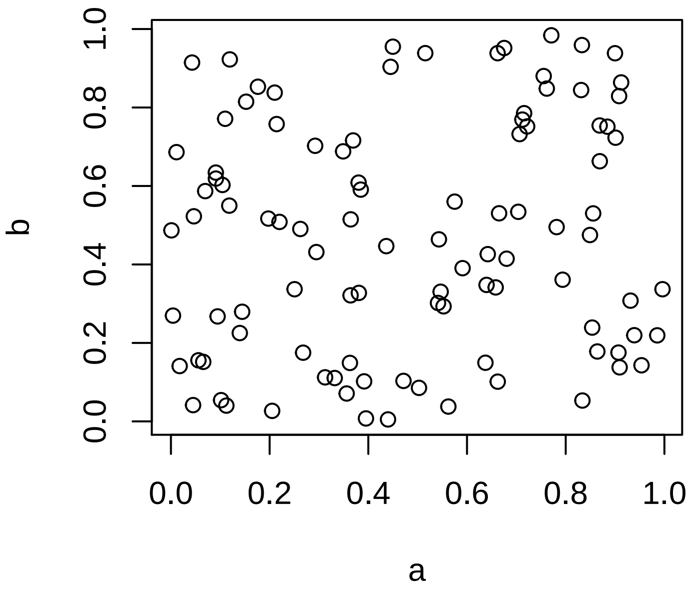
If I did not want to show any R output, I would just run:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9801	0.1724	5.69	0.0000
x1	2.0179	0.1014	19.90	0.0000
x2	-3.0576	0.0978	-31.27	0.0000

Table 2: A TABLE OF REGRESSION RESULTS

5 An example plot

If I wanted to add a plot (but no R code), I can run


1.0
0.8
0.6
0.4
0.2
0.0
b
0.0
0.2
0.4
0.6
0.8
1.0
a