MINIMUM WORKING EXAMPLES FOR REPRODUCIBLE SCIENCE WITH LaTeX ('BATCH MODE')

Haim Bar and HaiYing Wang
University of Connecticut
September 28, 2020

This supplement provides minimum working examples in R for the procedure with the "batch mode" proposed in the main paper. If the Python package *pygments* is not available on your computer, please install the latest versions of Python and *pygments*, or declare the 'nominted' option, namely, use

```
\usepackage[nominted]{runcode}
```

This will use the *fvextra* package which does not provide syntax highlights.

The program Code/code1.R generates a vector `x` by drawing a random sample of size 100 from a standard normal distribution, and generates `y` as `1+x+`ϵ, where the error term ϵ is also drawn from a standard normal distribution. Then, we fit a linear model, `y`$\sim$`x`. To show the source file's contents, we include the following in the tex document:

```
\showCode{R}{Code/code1.R}
```

which produces the following:

```
set.seed(0) ## fix the random number
x = rnorm(100)
y = 1+x+rnorm(100)
fit = lm(y~x)
print(summary(fit))
```

To show only lines 2-5 of the source code, we use

```
\showCode{R}{Code/code1.R}[2][5]
```

which produces the following:

```
x = rnorm(100)
y = 1+x+rnorm(100)
fit = lm(y~x)
print(summary(fit))
```

To execute the source code in Code/code1.R, we put the following in the tex file:

```
\runExtCode{Rscript --save --restore}{Code/code1.R}{fitLinear}
```

We include the output by using

```
\includeOutput{fitLinear}
```

and we obtain the following result:

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5900 -0.8153 -0.1531  0.6379  2.8379

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.95130    0.09629    9.88   <2e-16 ***
x            1.13879    0.10960   10.39   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9626 on 98 degrees of freedom
Multiple R-squared:  0.5242,        Adjusted R-squared:  0.5193
F-statistic:   108 on 1 and 98 DF,  p-value: < 2.2e-16
```

In the above example, if the third argument in `\runExtCode` is empty, then the second argument in `\includeOutput` should also be empty, namely, use

```
\runExtCode{Rscript --save --restore}{Code/code1.R}{}
\includeOutput{}
```

to include the above output. We recommend avoiding this type of usage because `\includeOutput` shows the results from the latest execution of `\runExtCode` with empty third argument, and this may make referencing harder to manage.

The `--save --restore` option for Rscript is used so that the session can be restored for later calculations. For example, we want to create an ANOVA table and calculate the mean squared error (MSE) using the code in Code/code2.R:

```
library("xtable")
fit.table <- xtable(aov(fit))
MSE = format(sum(fit$residuals^2)/fit$df.residual, digit=2)
print(fit.table)
```

We use

```
\runExtCode{Rscript --save --restore}{Code/code2.R}{linearANOVA}
\includeOutput{linearANOVA}[tex]
```

to obtain:
Here the code produces pure latex output, so we use the assign the value 'tex' to the optional

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x	1	100.02	100.02	107.96	0.0000
Residuals	98	90.80	0.93		

second argument of `\includeOutput` to include the output as-is.

To embed R code in tex file and include the output in line, we use the `\inln` command. For example, we use the following in the tex file to obtain the result: "The MSE is 0.93".

```
The MSE is \inln{Rscript --save --restore}{cat(MSE)}
```

The `\inln` command is also useful to show other types of output from previous calculations. As an example, we use

```
\inln{Rscript --save --restore}{aov(fit)}[vbox]
```

to obtain the following result.

```
Call:
   aov(formula = fit)

Terms:
                        x Residuals
Sum of Squares  100.02380  90.79886
Deg. of Freedom         1        98

Residual standard error: 0.9625586
Estimated effects may be unbalanced
```

We can embed more complicated R code in tex file using LaTeX *filecontents* environment. For instance, the following code in a tex file will create an R script file plot.R in the tmp folder, and then implement it to create a pdf figure tmp/linearScatter.pdf.

```
\begin{filecontents*}{tmp/plot.R}
pdf("tmp/linearScatter.pdf", width=6, height=4)
plot(x, y, pch=19, col="red" ,cex=0.8)
dev.off()
\end{filecontents*}
\runExtCode{Rscript --save --restore}{tmp/plot.R}{linearScatter}
```

We can include the resulting figure using `\includegraphics{}` and the *figure* environment in the usual way, e.g., with

```
\begin{figure}
\centering
\includegraphics[scale=0.7]{tmp/linearScatter.pdf}
\caption{A scatter plot of the simulated data}
\end{figure}
```

This produces Figure 1.

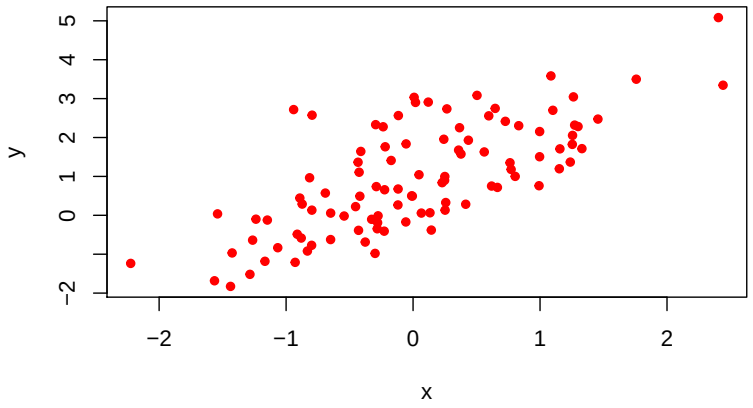

FIGURE 1. A scatter plot of the simulated data