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

Haim Bar and HaiYing Wang
University of Connecticut
October 2, 2020

This supplement provides minimum working examples in R for the procedure proposed in the main paper using the "server mode" provided by the Python *talk2stat* package. Please make sure you have Python3 available on your computer. Install the *talk2stat* package by the following command line.

```
pip3 install talk2stat
```

If the Python package *pygments* (for grammar highlighting) is not available on your computer please install it, or declare the 'nominted' option, namely, use

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

to load the *runcode* package. This will use the *fvextra* package which does not provide syntax highlights. If *pygments* is available, use

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

to load the *runcode* package. The 'R' option will start an R server.

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+`$\epsilon$, where the error term $\epsilon$ is also drawn from a standard normal distribution. Then, we fit a linear model, `y~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:

```r
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:

```
\runR{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 second argument in `\runR` is empty, then the second argument in `\includeOutput` should also be empty, namely, use

```
\runR{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 `\runR` with empty third argument, and this may make referencing harder to manage.

With the server mode, the R server is continuously running and all variables are always available, so we can use new codes for continuous calculations. For example, we want to use the variable `fit` 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

```
\runR{Code/code2.R}{linearANOVA}
\includeOutput{linearANOVA}[tex]
```

to obtain:

| | 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 | | |

Here the code produces pure latex output, so we use the assign the value 'tex' to the optional 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 `\inlnR` command. For example, we use the following in the tex file to obtain the result: "The MSE is 0.93".

```
The MSE is \inlnR{```cat(MSE)```}
```

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

```
\inlnR{```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*}
 \runR{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.

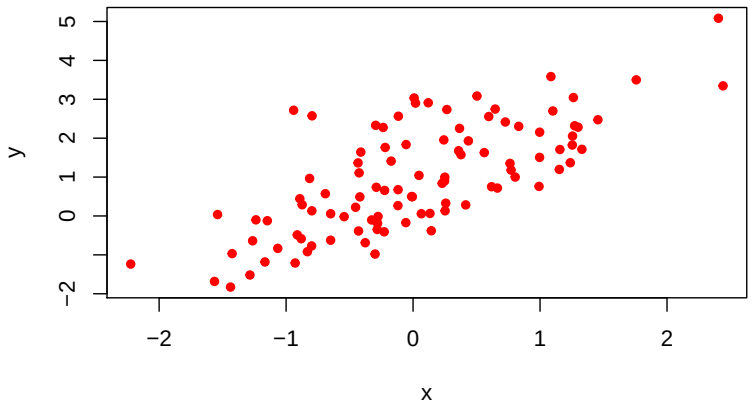


FIGURE 1. A scatter plot of the simulated data

Note that the working directory of the R server section is the same as that of the main tex file, so in the above example we can use relative directory for the pdf figure (`"tmp/linearScatter.pdf"`). If the file locations are changed after the R server section starts, one needs to use an absolute directory for the pdf figure, or one needs to end the R server section using the following command.

```
python3 -c 'from talk2stat.talk2stat import client; client("./","R","QUIT")'
```

After that, a compilation of the tex file in the new location will stat an R server section with the working directory as the new location.