Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

`simrel`: Simulating Linear Model Data in R

Raju Rimal
NMBU

Trygve Almøy
NMBU

Solve Sæbø
NMBU

Abstract

Weather to compare new methods, algorithm and model or to access their properties, using simulated data is common in most research. This paper introduces an R-package `simrel` which provide a simple interface and few parameters to simulate (multi) response multivariate linear model data with wide range of properties. A shiny web application gives the package a friendly and easy-to-use interface and lets users to download and and explore various properties of the data. Two included examples give usage of the package. The first one compares two machine learning algorithms and their interaction with different nature of data. The second one makes a comparative statistical analysis of two multi-variate methods.

Keywords: `simrel`, simulation, linear model, R.

1. Introduction

Simulation is an essential part of many research. Researchers use simulated data for testing their project, evaluate or assess different methods or algorithms, compare several ideas, methods etc.

2. Background

- A linear Model
- Too many parameters
- Two simrel papers about reparameterization
- Tie it to the concept of relevant space and relevant components

2.1. Relevant Space and Relevant Components

- Relevant space discussed in earlier papers
- The concept in envelope papers
- Relevant Components
- How the concept is used in simrel

2.2. The Framework

- The flowchart
- Small discussion around flowchart

3. Simulation

3.1. Simrel object

Simulation Parameters

```
sobj <- simrel(
  n      = 100, # Number of observation
  p      = 10, # Number of predictors
  q      = c(5, 5), # Number of relevant predictors
  m      = 4, # Number of responses
  R2     = c(0.7, 0.8), # Coef. of determination
  gamma  = 0.9, # Level of Multicollinearity
  eta    = 1.2, # Level of response correlation
  relpos = list(1:3, 4:6), # Position index of relevant components
  ypos   = list(c(1, 3), c(2, 4)), # Mixup response components
  type   = "multivariate"
)
ls(sobj)
```

```
 [1] "beta"      "beta0"     "call"      "eta"       "gamma"
 [6] "lambda"    "m"         "minerror"  "muX"       "muY"
[11] "n"         "ntest"     "p"         "q"         "R2"
[16] "relpos"    "relpred"   "rho.out"   "RsqW"      "RsqY"
[21] "Sigma"     "SigmaWX"   "SigmaWZ"   "SigmaYX"   "SigmaYZ"
[26] "testW"     "testX"     "testY"     "testZ"     "type"
[31] "W"         "X"         "Xrotation" "Y"         "ypos"
[36] "Yrotation" "Z"
```

Univariate, Bivariate and Multivariate Simulation

3.2. Properties

Coefficient of determination

```
sobj$RsqW
```

```
     [,1] [,2] [,3] [,4]
[1,]  0.7  0.0    0    0
[2,]  0.0  0.8    0    0
[3,]  0.0  0.0    0    0
[4,]  0.0  0.0    0    0
```

```
sobj$RsqY
```

```
           [,1]       [,2]       [,3]       [,4]
[1,]  0.6417791  0.0000000 -0.6417791  0.0000000
[2,]  0.0000000  0.7334618  0.0000000 -0.7334618
[3,] -0.6417791  0.0000000  0.6417791  0.0000000
[4,]  0.0000000 -0.7334618  0.0000000  0.7334618
```

Model Error

```
sobj$minerror
```

```
          [,1]        [,2]      [,3]        [,4]
[1,]  0.195359  0.00000000 -0.104641  0.00000000
[2,]  0.000000  0.04378128  0.000000 -0.01645756
[3,] -0.104641  0.00000000  0.195359  0.00000000
[4,]  0.000000 -0.01645756  0.000000  0.04378128
```

True Regression Coefficients

```
rbind(sobj$beta0, sobj$beta)
```

```
           [,1]       [,2]       [,3]       [,4]
 [1,]  0.0000000  0.0000000  0.0000000  0.0000000
 [2,] -0.5034391  0.0000000  0.5034391  0.0000000
 [3,] -0.2554883  0.0000000  0.2554883  0.0000000
 [4,]  0.3150672  0.0000000 -0.3150672  0.0000000
```

```
 [5,]  0.0000000  1.3655943  0.0000000 -1.3655943
 [6,]  0.0000000 -0.5244041  0.0000000  0.5244041
 [7,]  0.0000000  1.4972847  0.0000000 -1.4972847
 [8,]  0.0000000 -1.2769973  0.0000000  1.2769973
 [9,]  0.7955946  0.0000000 -0.7955946  0.0000000
[10,]  0.0000000 -1.0614776  0.0000000  1.0614776
[11,] -0.3517344  0.0000000  0.3517344  0.0000000
```

Relevant Predictors

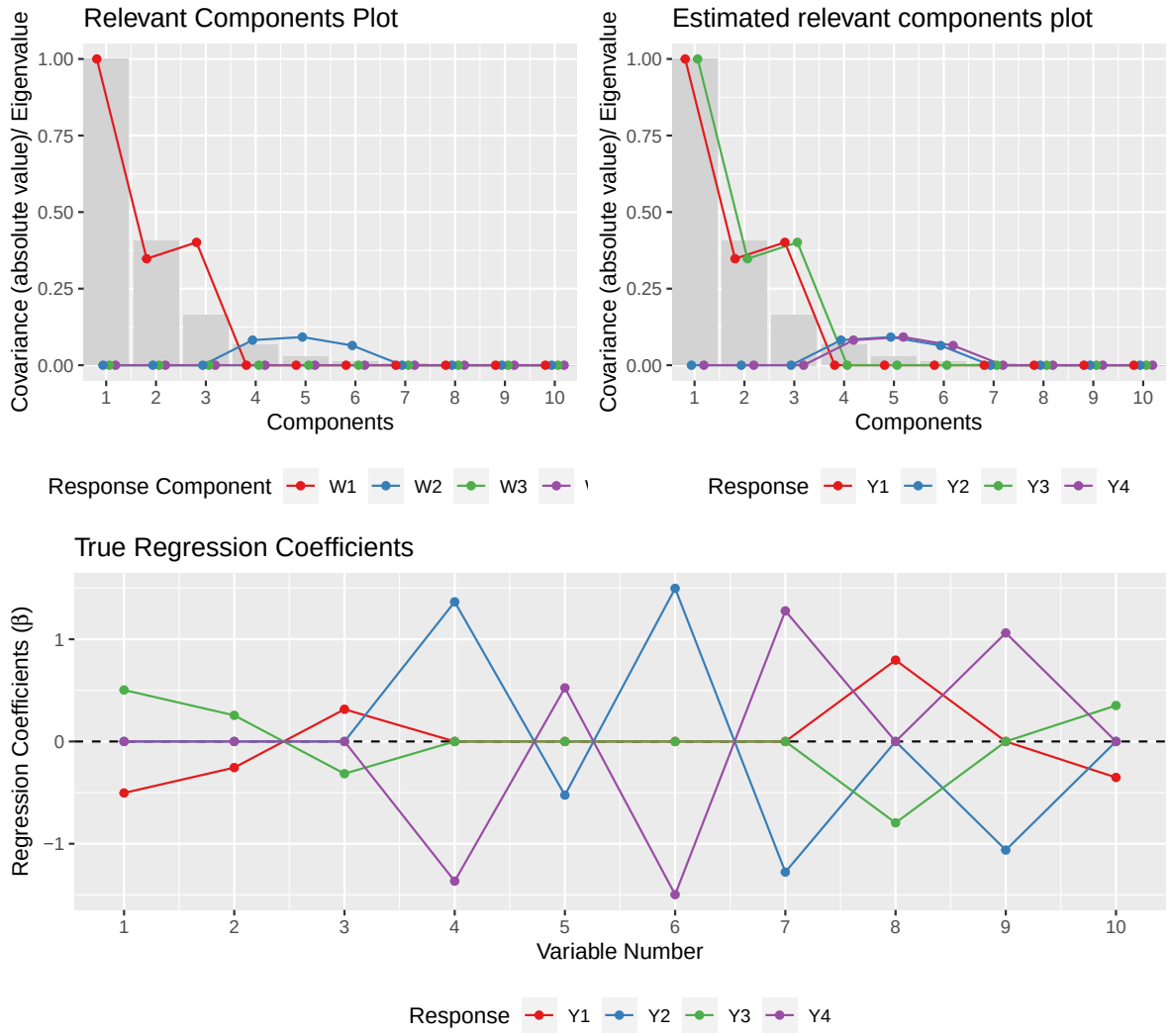
```
sobj$relpred
```

```
[[1]]
[1]  1  2  3 10  8

[[2]]
[1] 4 5 6 7 9

[[3]]
integer(0)

[[4]]
integer(0)

[[5]]
integer(0)
```

3.3. Plots

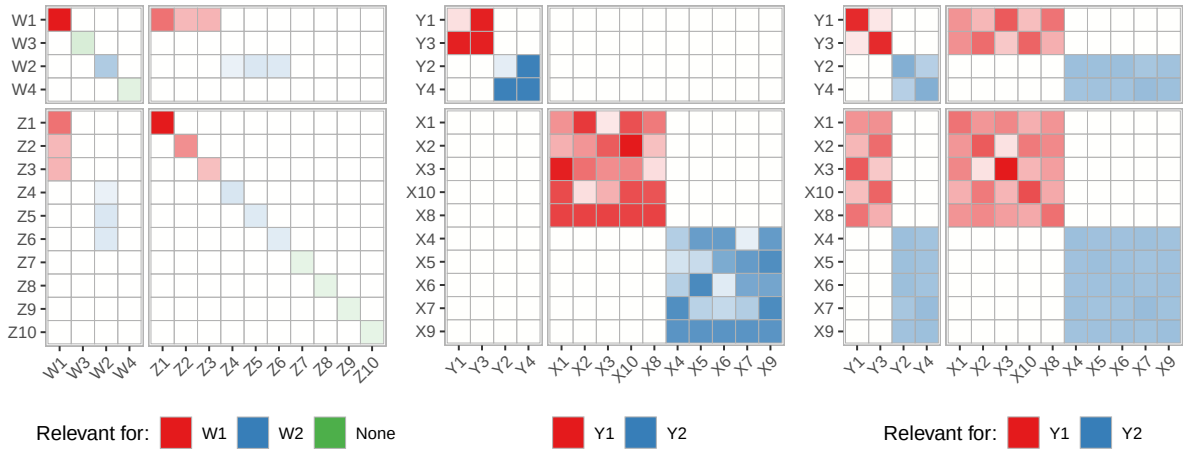
- Various simrel plots

```
ggsimrelplot(sobj)
```



- Covariance plots

```
plts <- lapply(c("relpos", "rotation", "relpred"), function(ptype) {
  plot_cov(sobj, type = ptype, facetting = FALSE) # +
    # theme(axis.text.x = element_text(hjust = 0.5, angle = 0))
})
plts$ncol <- 3
do.call(gridExtra::grid.arrange, plts)
```



4. An Example of Variable Importance

```
set.seed(2019)
sobj <- simrel(
  n = 1e4,
  p = 10,
  q = 7,
  relpos = 1:5,
  gamma = 1.2,
  R2 = 0.75,
  type = "univariate"
)
```

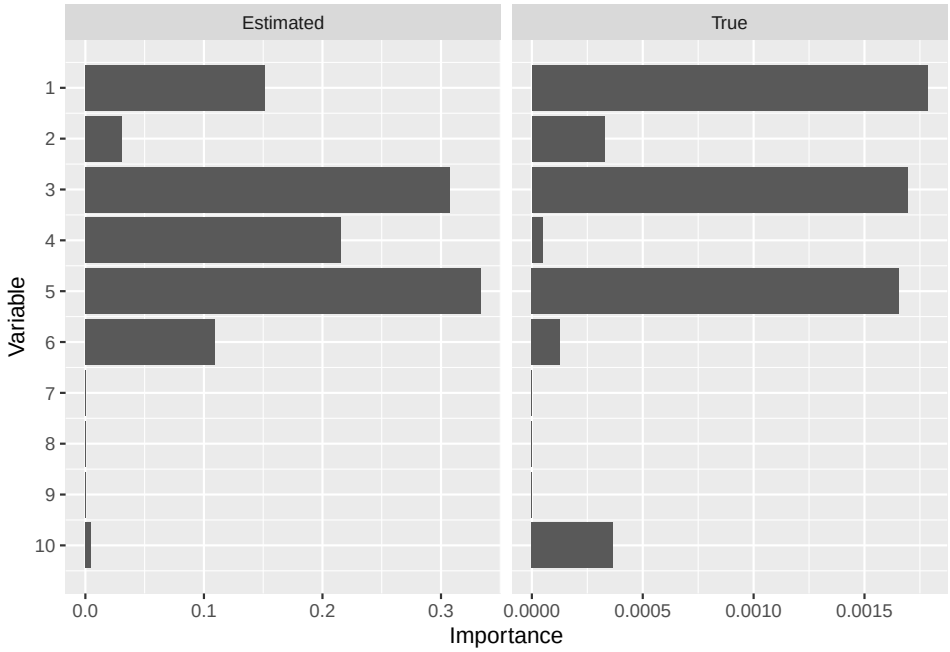
```
get_imp <- function(simrel_obj) {
  ## Complete Model
  rotation <- as.matrix(Matrix::bdiag(1, sobj$Rotation))
  sigma <- rotation %*% sobj$Sigma %*% t(rotation)
  sigma_xy <- sigma[-1,1]
  sigma_yx <- sigma[1,-1]
  sigma_xx <- sigma[-1,-1]
  sigma_yy <- sigma[1,1]
  minerr <- sigma_yy - sigma_yx %*% solve(sigma_xx) %*% sigma_xy

  ## Reduced Model
  imp <- c()
  for (idx in 1:sobj$p) {
    sigma_xiy  <- sigma_xy[-idx]
    sigma_yxi  <- sigma_yx[-idx]
    sigma_xixi <- sigma_xx[-idx,-idx]
    minerr_i   <- sigma_yy - sigma_yxi %*% solve(sigma_xixi) %*% sigma_xiy
    imp[idx]   <- minerr_i - minerr
  }
```

```
  return(imp)
}
```

```
imp        <- get_imp(sobj)
importance <- data.frame(idx = seq_along(imp), imp = imp)
```

```
set.seed(2019)
dta        <- with(sobj, data.frame(Y, X))
control    <- trainControl(method="repeatedcv", number=10, repeats=3)
model      <- train(Y ~ ., data = dta, method="pcr",
                    preProcess="scale", trControl=control)
importance <- varImp(model, scale=FALSE)
```

```
imp_df <- bind_cols(True = imp, Estimated = importance$importance$Overall) %>%
  rownames_to_column("var") %>%
  gather(ImpType, Imp, -var) %>%
  mutate_at("var", parse_integer)

plt <- imp_df %>%
  ggplot(aes(var, Imp)) +
  geom_bar(position = "dodge", stat = "identity") +
  scale_color_discrete(l = 40) +
  coord_flip() +
  scale_x_reverse(breaks = unique(imp_df$var)) +
  facet_grid(.~ImpType, scales = 'free_x') +
  labs(x = "Variable",
       y = "Importance") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0),
        legend.box.margin = margin(3, 3, 3, 3))
plot(plt)
```



5. Summary

Affiliation:

Raju Rimal
Norwegian University of Life Sciences (NMBU)
Ås, 1433, NORWAY
E-mail: raju.rimal@nmbu.no
URL: http://mathatistics.com

Trygve Almøy
Norwegian University of Life Sciences (NMBU)
Ås, 1433, NORWAY
E-mail: trygve.almoy@nmbu.no

Solve Sæbø
Norwegian University of Life Sciences (NMBU)
Ås, 1433, NORWAY
E-mail: solve.sabo@nmbu.no
URL: http://blogg.nmbu.no/solvesabo/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd