

Trimming R objects to reduce memory

By Nanye Long, iCER Research Consultant

Sometimes you have created an R object but only need some of its components to carry out subsequent analysis. In this case, you may want to trim your object so as to free up memory.

As an example, you fit a glm model in order to use it to make prediction on a new dataset. If the training dataset (the one used for fitting your glm) is large, the size of the resulting glm object can be large too, although many of its components are unnecessary for prediction.

In the example code below, all components except **qr** and **coefficients** can be removed without affecting the result. Notably, the size of the glm object was reduced from 330 Mb to 50 Kb.

```
# Training data (big)
bigdata <- iris[sample(1:150, 500000, replace=TRUE),]

# Fit a glm model
fit <- glm(I(Species=="setosa") ~ ., data=bigdata)

# Get size of the fitted glm model object "fit"
format(object.size(fit), unit="Mb")

## [1] "330.6 Mb"

# Make prediction using the full "fit"
pred.results <- predict(fit, newdata=iris)

# Trimming begins
fit$data <- NULL
fit$y <- NULL
fit$linear.predictors <- NULL
fit$weights <- NULL
fit$fitted.values <- NULL
fit$model <- NULL
fit$prior.weights <- NULL
fit$residuals <- NULL
fit$effects <- NULL
fit$qr$qr <- NULL
```

Jan. 2018



```
# Now look at the size of the trimmed "fit"  
format(object.size(fit), unit="Kb")  
  
## [1] "49.3 Kb"  
  
# Check if prediction results from trimmed "fit" are identical to those from full "fit"  
all(predict(fit, newdata=iris)==pred.results)  
  
## [1] TRUE
```

If you have any further question, please feel free to contact us.