Appendix 2

April 5, 2017

The `greedy` R function, uses the Greedy algorithm to step through a simplified subset of models by deleting “eligible” predictors with minimum $|t|$ values. By eligible predictors we mean that “no main effect can be dropped until all interactions and curvature terms involving that variable have been dropped.”

The current algorithm:

```
# formula = A formula for a linear model.  Must currently be specified using main effects ONLY to allow identification of so-called eligible models.
# Interactions and quadratic terms will be generated by the function.
# data = An optional data frame that contains non-global variables within formula.
# center = If specified, a character vector containing names of quantitative variables (specified in formula) to be centered.
# digits = Number of digits in output.
# inform = Type of information-theoretic criterion to be used in model evaluation, one of "AIC", "BIC", or "PRESS" (predicted R^2).
```

```
# formula = A formula for a linear model.  Must currently be specified using main effects ONLY to allow identification of so-called eligible models.
# Interactions and quadratic terms will be generated by the function.
# data = An optional data frame that contains non-global variables within formula.
# center = If specified, a character vector containing names of quantitative variables (specified in formula) to be centered.
# digits = Number of digits in output.
# inform = Type of information-theoretic criterion to be used in model evaluation, one of "AIC", "BIC", or "PRESS" (predicted R^2).
```

```
greedy <- function(formula, data = NULL, center = NULL, digits = 5, inform = "AIC"){
  require(asbio)
  data <- get_all_vars(formula, data = data)
  if(!is.null(center)){
    w <- which(names(data) == center)
    temp <- apply(data[,w], 2, function(x) x - mean(x))
    data[,w] <- temp
  }
  m <- model.frame(formula, data = data)
  Y <- model.extract(m, "response")
  terms <- terms(m)
  X <- attr(terms,"term.labels")
  k <- length(X)
  steps <- 1 + (k^2 + 3*k)/2

    inf <- function(model, inform){
      switch(inform,
        AIC = AIC(model),
        BIC = BIC(model),
        PRESS = PRESS(model, as.R2 = TRUE))
    }

  tab <- matrix(nrow = steps, ncol = 3)
  colnames(tab) <- c("Model", "Drop", inform)
  test <- lm(formula, data = data)
```

```
if(steps==1) tab[1,] = c(deparse(formula(test$terms)), " ", round(inf(test, inform), digits = digits))

else if(steps>=2){
  d <- attr(terms,"dataClasses")[2:(length(X)+1)]
  Xn <- X[d == "numeric"]
  Xsq <- paste("I(", Xn, "^2)", sep="")
  Xint <- outer(X, X, function(x,y) paste(x,":",y,sep=""))
  Xint <- Xint[upper.tri(Xint)]
  Xall <- c(X, Xint, Xsq)
  if(!any(match(Xn, X))) Xall <- c(X, Xint)
  Yname <- names(m)[1]

  f <- as.formula(paste(c(paste(Yname,"~ 1 "), Xall), collapse=" + "))
  sat <- lm(f, data = data)

  redo <- function(drop1, sumsat){
    sumsat1 <- sumsat[rownames(sumsat)!=drop1,]
    if(class(sumsat1)=="numeric") sumsat1 = t(as.matrix(sumsat1))
    rn <- rownames(sumsat)[rownames(sumsat)!=drop1]
    if(nrow(sumsat1)==1) rownames(sumsat1) = rn
    drop2 <- rn[which(abs(sumsat1[,3]) == min(abs(sumsat1[,3])))]
    sumsat2 = sumsat[rownames(sumsat)!=drop2,]
    res <- list(sumsat = sumsat2, drop1 = drop2)
    res
  }

  drops <-function(mod1, X, data){
    np <- nrow(coef(summary(mod1)))
    if(np == 2){
      new.mod <- update(mod1, ~ 1)
      res <- list(formula = paste(Yname,"~ 1"), model = new.mod, drop = attr(terms(mod1), "term.labels"),
                  inf.crit = round(inf(new.mod, inform), digits = digits))
    }
    if(np > 2){
      sumsat <- coef(summary(mod1))[2:np,]
      drop1 <- rownames(sumsat)[which(abs(sumsat[,3]) == min(abs(sumsat[,3])))]
      mod.terms <- attr(terms(mod1), "term.labels")

      if(any(X==drop1) & length(grep(drop1, mod.terms[mod.terms!=drop1]))>0){
        drop1 <- redo(drop1, sumsat)$drop1
        if(any(X==drop1) & length(grep(drop1, mod.terms[mod.terms!=drop1])>0)){
          drop1 <- redo(drop1, sumsat)$drop1
          if(any(X==drop1) & length(grep(drop1, mod.terms[mod.terms!=drop1])>0)){
            drop1 <- redo(drop1, sumsat)$drop1
          }
        }
      }
      f1 <- as.formula(paste(c(paste(Yname,"~ 1 "), mod.terms[mod.terms!=drop1]), collapse=" + "))
      new.mod <- lm(f1, data= data)
      res <- list(formula = paste(c(paste(Yname,"~ 1 "), mod.terms[mod.terms!=drop1]), collapse=" + "),
                      model = new.mod, drop = drop1, inf.crit = round(inf(new.mod, inform), digits = digits))
        }
        res
      }

      j = 2; temp <- sat
      tab[1,] <-c(paste(c(paste(Yname,"~ 1"), Xall), collapse=" + "), " ", round(inf(temp, inform), digits = digits))
      while(j <= steps){
        temp <- drops(temp, X, data = data)
        tab[j,] <- c(temp$formula, temp$drop, temp$inf.crit)
        temp <- temp$model
        j = j + 1
      }
    }
    if(inform == "AIC" | inform == "BIC") opt <- which(as.numeric(tab[,3])== min(as.numeric(tab[,3])))
    if(inform == "PRESS") opt <- which(as.numeric(tab[,3]) == max(as.numeric(tab[,3])))
      best <- tab[opt,][1]
      best <- lm(noquote(best), data = data)
  res <- list()
    res$out <- data.frame(tab)
    res$method <- inform
    res$best <- best
    res$data <- data
    class(res) <- "greedy"
  res
}

print.greedy <- function (x, ...){
    cat("\n")
    out <- structure(x$out)
    print(out)
    invisible(x)
}
```

Example: Case 0902

Data from Case 0902, "The Statistical Slueth" Ramsey and Schaefer (1997)

```
library(MASS)
library(asbio)
```

```
Loading required package: tcltk
```

```
#readFile <- "Datasets/concreteData.csv"
#---------------------------------------------

varData <- read.csv(file = "C:/Users/esham/Desktop/R Project Stuff/R Directory/Datasets/case0902.csv")
case0902 <- varData
names(case0902) = c("Xs", "Y", "Xb", "Xg", "Xl")
#data(case0902)
```

```r
g0902 <- greedy(log(Y) ~ log(Xb) + Xg + Xl, data = case0902)
g0902
```

```
                                                                                                 Model         Drop       AIC
1  log(Y) ~ 1 + log(Xb) + Xg + Xl + log(Xb):Xg + log(Xb):Xl + Xg:Xl + I(log(Xb)^2) + I(Xg^2) + I(Xl^2)              126.56496
2                log(Y) ~ 1  + log(Xb) + Xg + Xl + I(Xg^2) + I(Xl^2) + log(Xb):Xg + log(Xb):Xl + Xg:Xl I(log(Xb)^2) 124.69777
3                          log(Y) ~ 1  + log(Xb) + Xg + Xl + I(Xg^2) + log(Xb):Xg + log(Xb):Xl + Xg:Xl      I(Xl^2) 123.34415
4                                       log(Y) ~ 1  + log(Xb) + Xg + Xl + I(Xg^2) + log(Xb):Xg + Xg:Xl   log(Xb):Xl 122.98736
5                                               log(Y) ~ 1  + log(Xb) + Xg + Xl + I(Xg^2) + log(Xb):Xg        Xg:Xl  126.1943
6                                                         log(Y) ~ 1  + log(Xb) + Xg + Xl + log(Xb):Xg      I(Xg^2) 131.05325
7                                                              log(Y) ~ 1  + log(Xb) + Xg + log(Xb):Xg           Xl 136.36012
8                                                                           log(Y) ~ 1  + log(Xb) + Xg   log(Xb):Xg 159.07119
9                                                                                log(Y) ~ 1  + log(Xb)           Xg  171.1866
10                                                                                          log(Y) ~ 1      log(Xb) 424.27854
```

```r
g0902$best
```

```

Call:
lm(formula = noquote(best), data = data)

Coefficients:
(Intercept)      log(Xb)           Xg           Xl      I(Xg^2)   log(Xb):Xg        Xg:Xl  
  2.135e+00    7.865e-01    5.778e-03   -7.015e-03    1.281e-05   -1.629e-03   -1.389e-03  
```

```r
s0902<- stepAIC(lm(log(Y) ~ I(log(Xb)^2) + I(Xg^2) + I(Xl^2) + (log(Xb) + Xg + Xl)^2, data = case0902), trace = FALSE)
s0902
```

```

Call:
lm(formula = log(Y) ~ I(Xg^2) + log(Xb) + Xg + Xl + log(Xb):Xg + 
    Xg:Xl, data = case0902)

Coefficients:
(Intercept)      I(Xg^2)      log(Xb)           Xg           Xl   log(Xb):Xg        Xg:Xl  
  2.135e+00    1.281e-05    7.865e-01    5.778e-03   -7.015e-03   -1.629e-03   -1.389e-03  
```

```r
# center Xs to reduce collinearity
library(car)
vif(g0902$best)
```

```
   log(Xb)         Xg         Xl    I(Xg^2) log(Xb):Xg      Xg:Xl 
  9.468999  20.254387   5.434937  23.368211  32.461919   4.016400 
```

```r
g0902c <- greedy(log(Y) ~ log(Xb) + Xg + Xl, data = case0902, center = c("Xg","Xl"))
vif(g0902c$best)
```

```
   log(Xb)         Xg         Xl    I(Xg^2) log(Xb):Xg      Xg:Xl 
  4.246910  12.085050   5.372048   6.119999  14.368900   4.969293 
```