

1 Explanation

I haven't had a chance to organize this, obviously. The main purpose of this section is to run MLR on different combinations of variables, and then test them via, Adjusted R^2 , AIC, and BIC. I created a function "test()" to do this. You input an outcome variable, variables to use in every model, and variables to test. The function output is a list of three objects. The first contains the variables ran in the MLR. The second is a matrix of Adj. R^2 , AIC, and BIC values for each model. The third is a list of each lm() output.

The outcome variable I chose was "SYSBP"

Every model includes: "AGE" and "SEX"

I tested "TOTCHOL", "BMI", "GLUCOSE", "CURSMOKE", "DIABETES", "BPMEDS", "HEARTRTE", and "educ".

Adjusted R^2 was highest for the model that includes: "TOTCHOL", "BMI", "GLUCOSE", "CURSMOKE", "BPMEDS", "HEARTRTE", "educ", "AGE", and "SEX". This includes everything but "DIABETES".

AIC and BID values were lowest for the model that includes: "TOTCHOL", "BMI", "GLUCOSE", "BPMEDS", "HEARTRTE", "educ", "AGE", and "SEX". This includes everything but "DIABETES", and "CURSMOKE".

There is a ton of other stuff to say about this. For one BMI was the strongest association for adjusted R^2 , and GLUCOSE was the strongest association for AIC and BIC.

AIC and BIC had the same strongest association per number of variables, but this wasn't true for every combination of variables.

I ranked each model, for each criteria and want to include a graph of these values as well.

You can run the code above. After you do the "yes" object will contain the output of the test() function. The "rank" will contain the matrix of rank of each criteria.

```
require(ggplot2)

## Loading required package: ggplot2

require(RCurl)

## Loading required package: RCurl
## Loading required package: bitops

require(plyr)

## Loading required package: plyr

data <- getURL("https://raw.githubusercontent.com/arvindram12/Final-Project/master/frmgham2.csv",
  ssl.verifypeer = 0L, followlocation = 1L)
writeLines(data, "framingham.csv")
frm <- read.csv("framingham.csv")
frm$RANDID <- as.factor(frm$RANDID)
frm$SEX <- as.factor(frm$SEX)
frm$CURSMOKE <- as.factor(frm$CURSMOKE)
frm$DIABETES <- as.factor(frm$DIABETES)
frm$BPMEDS <- as.factor(frm$BPMEDS)
frm$educ <- as.factor(frm$educ)
frm1 <- frm[which(!duplicated(frm$RANDID)), ]
test <- function(outcome, need, test, data) {
  out1 <- data[, outcome]
  need1 <- as.data.frame(data[, need])
  colnames(need1) <- colnames(data[need])
  test1 <- data[, test]
  x <- 1:ncol(test1)
  combn1 <- function(x, y) {
    combn(y, x)
  }
  dee <- lapply(x, combn1, y = x)
  elem <- function(x, y, z) {
    test2 <- as.data.frame(x = z[, x])
    colnames(test2) <- colnames(z[x])
    mat <- as.data.frame(cbind(test2, need1))
    lm(y ~ ., data = mat)
  }
}
```

```

what <- function(c) {
  apply(dee[[c]], 2, elem, y = out1, z = test1)
}
model <- as.list(lapply(x, what))
model1 <- unlist(model, recursive = FALSE)
call <- function(x) {
  sum <- summary(x)
  sum$adj.r.squared
}
call1 <- function(x) {
  names(x$coef)
}
coef <- lapply(model1, call1)
coef1 <- lapply(coef, rbind)

coef2 <- rbind.fill.matrix(coef1)
adjr <- lapply(model1, call)
adjr1 <- do.call(rbind, adjr)
aics <- lapply(model1, AIC)
aics1 <- do.call(rbind, aics)
bics <- lapply(model1, BIC)
bics1 <- do.call(rbind, bics)
frame <- as.data.frame(cbind(adjr1, aics1, bics1)
  colnames(frame) <- c("Adj R^2", "AIC", "BIC")
  list(coef2, frame, model1)
}
yes <- test(outcome = "SYSBP", need = c("AGE", "SEX"), test = c("TOTCHOL", "BMI",
  "GLUCOSE", "CURSMOKE", "DIABETES", "BPMEDS", "HEARTRTE", "educ"), data = frm1)
yes1 <- yes[[1]]
yes2 <- yes[[2]]
rank <- cbind(nrow(yes2) + 1 - rank(yes2[, 1]), rank(yes2[, 2]), rank(yes2[,
  3]))
rank <- as.data.frame(rank)
yes2 <- as.data.frame(yes2)
yes2[, 4] <- c(rep(1, choose(8, 1)), rep(2, choose(8, 2)), rep(3, choose(8,
  3)), rep(4, choose(8, 4)), rep(5, choose(8, 5)), rep(6, choose(8, 6)), rep(7,
  choose(8, 7)), rep(8, choose(8, 8)))
yes2[, 4] <- as.factor(yes2[, 4])

adjranks <- c(which(rank[, 1] == min(rank[1:8, 1])), which(rank[, 1] == min(rank[9:36,
  1])), which(rank[, 1] == min(rank[37:92, 1])), which(rank[, 1] == min(rank[93:162,
  1])), which(rank[, 1] == min(rank[163:218, 1])), which(rank[, 1] == min(rank[219:246,
  1])), which(rank[, 1] == min(rank[247:254, 1])), which(rank[, 1] == min(rank[255:255,
  1])))

aicranks <- c(which(rank[, 2] == min(rank[1:8, 2])), which(rank[, 2] == min(rank[9:36,
  2])), which(rank[, 2] == min(rank[37:92, 2])), which(rank[, 2] == min(rank[93:162,
  2])), which(rank[, 2] == min(rank[163:218, 2])), which(rank[, 2] == min(rank[219:246,
  2])), which(rank[, 2] == min(rank[247:254, 2])), which(rank[, 2] == min(rank[255:255,
  2])))

bicranks <- c(which(rank[, 3] == min(rank[1:8, 3])), which(rank[, 3] == min(rank[9:36,
  3])), which(rank[, 3] == min(rank[37:92, 3])), which(rank[, 3] == min(rank[93:162,
  3])), which(rank[, 3] == min(rank[163:218, 3])), which(rank[, 3] == min(rank[219:246,
  3])), which(rank[, 3] == min(rank[247:254, 3])), which(rank[, 3] == min(rank[255:255,
  3])))

adjrvalues <- yes2[adjranks, ]
aicbicvalues <- yes2[aicranks, ]

```

```

adjrmodel <- yes1[adjranks, ]
aicbicmodel <- yes1[bicranks, ]

multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
  require(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel ncol: Number of columns of plots nrow: Number of rows
    # needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)), ncol = cols,
                      nrow = ceiling(numPlots/cols))
  }

  if (numPlots == 1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row, layout.pos.col = matchidx$col))
    }
  }
}

pl1 <- ggplot(yes2, aes(x = 1:255, y = yes2[, 1], color = yes2[, 4])) + geom_point(aes(colour = yes2[,
4]), size = 2) + ggtitle("Adjusted R^2 Values of \n Multiple Linear Regression Models") +
ylab("Adjusted R^2") + xlab("Model Number") + labs(colour = "No. Added \n Variables")

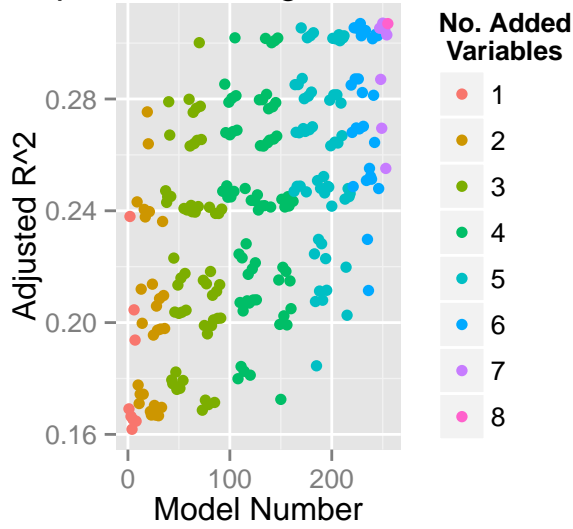
pl2 <- ggplot(yes2, aes(x = 1:255, y = yes2[, 2], color = yes2[, 4])) + geom_point(aes(colour = yes2[,
4]), size = 2) + ggtitle("Akaike Information Criterion Values of \n Multiple Linear Regression Models") +
ylab("AIC") + xlab("Model Number") + labs(colour = "No. Added \n Variables")

pl3 <- ggplot(yes2, aes(x = 1:255, y = yes2[, 3], color = yes2[, 4])) + geom_point(aes(colour = yes2[,
4]), size = 2) + ggtitle("Bayesian Information Criterion Values of \n Multiple Linear Regression Models") +
ylab("BIC") + xlab("Model Number") + labs(colour = "No. Added \n Variables")
layout(matrix(1:3, nrow = 1))
multiplot(pl1, pl2, pl3, cols = 2)

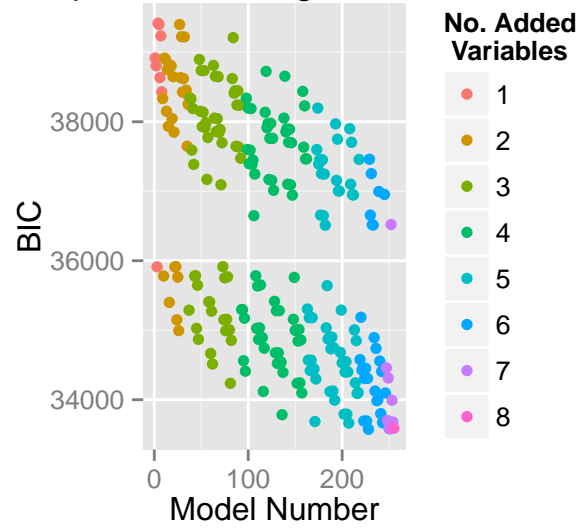
## Loading required package: grid

```

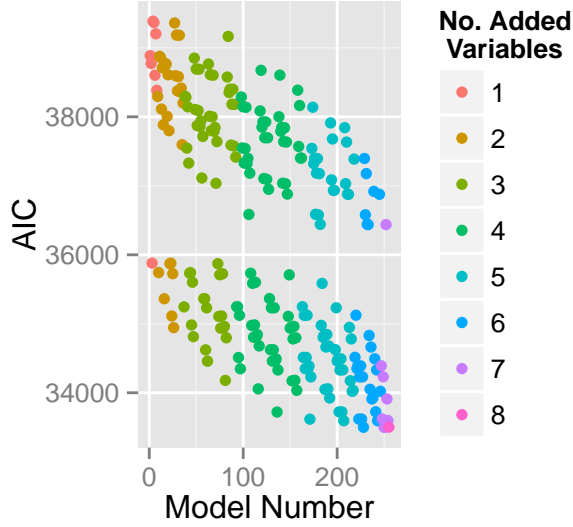
Adjusted R^2 Values of Multiple Linear Regression Models



Bayesian Information Criterion Values of Multiple Linear Regression Models



Akaike Information Criterion Values of Multiple Linear Regression Models



Variables	BMI	BPMed	Heart Rate	Tot Chol	Glucose	Education	Smoke	Diabetes	Adj R^2	AIC	BIC
1	YES	NO	NO	NO	NO	NO	NO	NO	0.237966	38772.70	38804.67
2	YES	YES	NO	NO	NO	NO	NO	NO	0.275382	38006.02	38044.29
3	YES	YES	YES	NO	NO	NO	NO	NO	0.300141	37844.80	37889.45
4	YES	YES	YES	YES	NO	NO	NO	NO	0.301882	37398.21	37449.15
5	YES	YES	YES	YES	YES	NO	NO	NO	0.30543	34387.96	34444.50
6	YES	YES	YES	YES	YES	YES	NO	NO	0.306995	33499.77	33574.84
7	YES	YES	YES	YES	YES	YES	YES	NO	0.307125	33500.04	33581.37
8	YES	YES	YES	YES	YES	YES	YES	YES	0.306962	33501.94	33589.52

Table 1: Multiple Linear Regression Models for Maximum Adjusted R^2

Variables	Glucose	Education	BP Med	BMI	Heart Rate	Tot Chol	Smoking Status	Diabetes	Adj R^2	AIC	BIC
1	YES	NO	NO	NO	NO	NO	NO	NO	0.237966	38772.70	38804.67
2	YES	YES	NO	NO	NO	NO	NO	NO	0.275382	38006.02	38044.29
3	YES	YES	YES	NO	NO	NO	NO	NO	0.300141	37844.80	37889.45
4	YES	YES	YES	YES	NO	NO	NO	NO	0.301882	37398.21	37449.15
5	YES	YES	YES	YES	YES	NO	NO	NO	0.30543	34387.96	34444.50
6	YES	YES	YES	YES	YES	YES	NO	NO	0.306995	33499.77	33574.84
7	YES	YES	YES	YES	YES	YES	YES	NO	0.307125	33500.04	33581.37
8	YES	YES	YES	YES	YES	YES	YES	YES	0.306962	33501.94	33589.52

Table 2: Multiple Linear Regression Models for Minimum AIC and BIC)