Multivariate Statistics - Exercise 1

Philipp Satlawa - h0640348

11/04/2021

This document contains the answered questions of exercise 1 of the course "Multivariate Statistics".

Linear regression

1.install and load the packages & load and summerize the data

```
# install package
#install.packages("gamair")
# import necessary libraries
library(gamair)
library(leaps)
# load data
data(mpg)
# overview over dataset
str(mpg)
```

```
## 'data.frame':    205 obs. of  26 variables:
##  $ symbol    : int  3 3 1 2 2 2 1 1 1 0 ...
##  $ loss      : int  NA NA NA 164 164 NA 158 NA 158 NA ...
##  $ make      : Factor w/ 22 levels "alfa-romero",..: 1 1 1 2 2 2 2 2 2 2 ...
##  $ fuel      : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ aspir     : Factor w/ 2 levels "std","turbo": 1 1 1 1 1 1 1 1 2 2 ...
##  $ doors     : Factor w/ 2 levels "four","two": 2 2 2 1 1 2 1 1 1 2 ...
##  $ style     : Factor w/ 5 levels "convertible",..: 1 1 3 4 4 4 4 5 4 3 ...
##  $ drive     : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 2 1 2 2 2 2 1 ...
##  $ eng.loc   : Factor w/ 2 levels "front","rear": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wb        : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ length    : num  169 169 171 177 177 ...
##  $ width     : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...
##  $ height    : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
##  $ weight    : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...
##  $ eng.type  : Factor w/ 7 levels "dohc","dohcv",..: 1 1 6 4 4 4 4 4 4 4 ...
##  $ cylinders : Factor w/ 7 levels "eight","five",..: 3 3 4 3 2 2 2 2 2 2 ...
##  $ eng.cc    : int  130 130 152 109 136 136 136 136 131 131 ...
##  $ fuel.sys  : Factor w/ 8 levels "1bbl","2bbl",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ bore      : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
```

```
##  $ stroke    : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
##  $ comp.ratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
##  $ hp        : int  111 111 154 102 115 110 110 110 140 160 ...
##  $ rpm       : int  5000 5000 5000 5500 5500 5500 5500 5500 5500 5500 ...
##  $ city.mpg  : int  21 21 19 24 18 19 19 19 17 16 ...
##  $ hw.mpg    : int  27 27 26 30 22 25 25 25 20 22 ...
##  $ price     : int  13495 16500 16500 13950 17450 15250 17710 18920 23875 NA ...
```

```
# show if attributes are factors
unlist(lapply(mpg, is.factor))
```

```
##     symbol       loss       make       fuel      aspir      doors      style
##      FALSE      FALSE       TRUE       TRUE       TRUE       TRUE       TRUE
##      drive    eng.loc         wb     length      width     height     weight
##       TRUE       TRUE      FALSE      FALSE      FALSE      FALSE      FALSE
##   eng.type  cylinders     eng.cc   fuel.sys       bore     stroke comp.ratio
##       TRUE       TRUE      FALSE       TRUE      FALSE      FALSE      FALSE
##         hp        rpm   city.mpg     hw.mpg      price
##      FALSE      FALSE      FALSE      FALSE      FALSE
```

```
# using result of the above calculation to determine the numeric attributes
# possibility 1:
sum(!unlist(lapply(mpg, is.factor)))
```

```
## [1] 16
```

```
# possibility 2:
length(mpg) - sum(unlist(lapply(mpg, is.factor)))
```

```
## [1] 16
```

```
# possibility 3:
sum(unlist(lapply(mpg, is.numeric)))
```

```
## [1] 16
```

The `mpg` dataset consists of 26 attributes and 205 observations as expected. 16 of all attributes are numeric and 10 are categorical.

2. Data preperation

```
# restrict dataset to certain attributes
mpg2 <- mpg[c("hw.mpg", "wb", "length", "width", "height", "weight", "eng.cc",
                "bore", "stroke", "hp")]
# remove records with 'NA' values
mpg2 <- na.omit(mpg2)
# get number of rows
nrow(mpg2)
```

```
## [1] 199
```

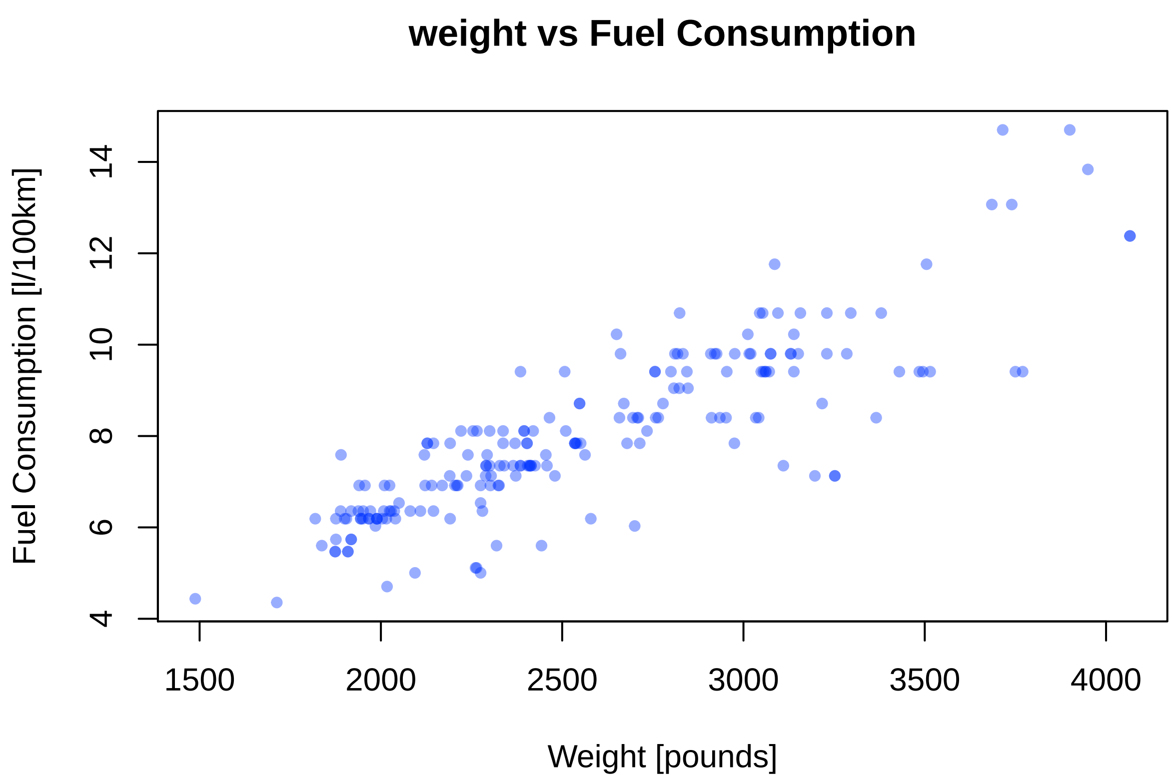
After removing all records containing `NA`s, the dataset consists of 199 records.

3. Convert attribute "fuel efficiency" into the metric system

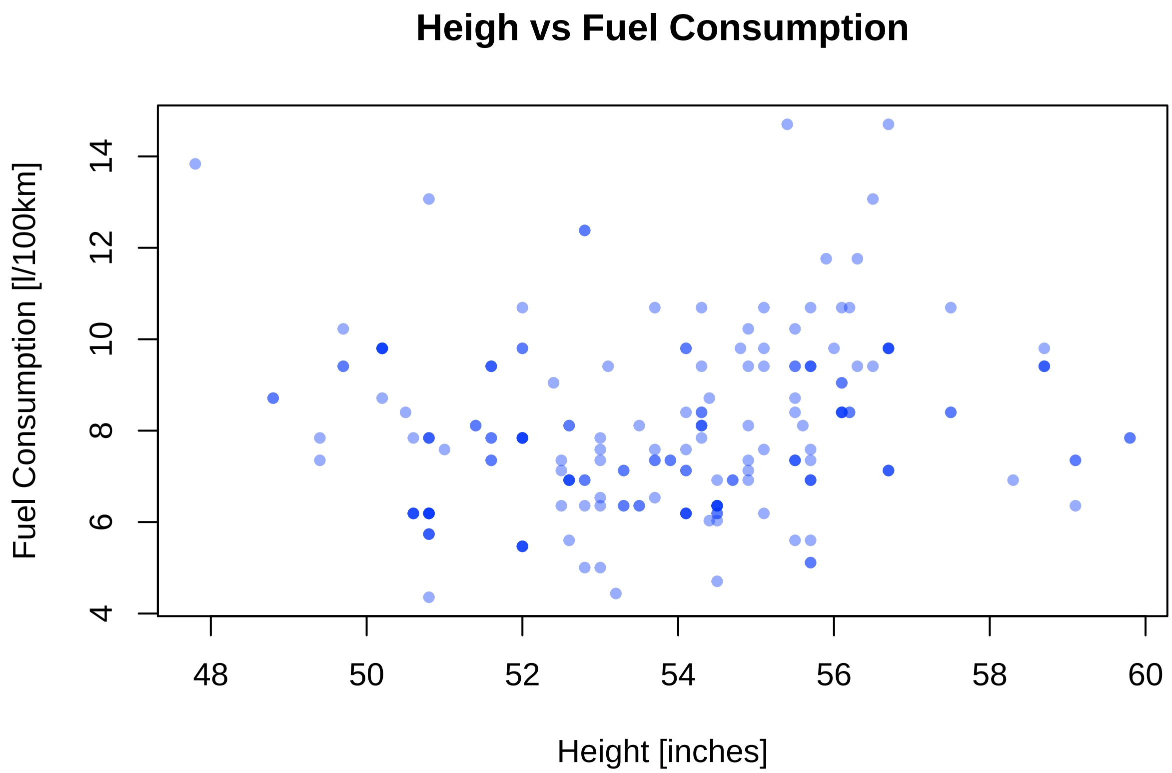
```
# calculating "lphk" (in liters per 100 kilometer)
mpg2["lphk"] <- 100 / ((mpg2["hw.mpg"] / 0.621371) * 0.264172)
# removing "hw.mpg"
mpg2 <- mpg2[ , ! names(mpg2) %in% c("hw.mpg")]
```

4. Plot data

```
# scatterplot "weight" vs "lphk"
plot(mpg2[c("weight","lphk")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "weight vs Fuel Consumption",
      xlab = "Weight [pounds]", ylab = "Fuel Consumption [l/100km]")
```




```
# scatterplot "height" vs "lphk"
plot(mpg2[c("height","lphk")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Heigh vs Fuel Consumption",
      xlab = "Height [inches]", ylab = "Fuel Consumption [l/100km]")
```



The first scatterplot indicates that there is a positive correlation between the variables `lphk` and `weight`, because with the increase of the attribute `weight` the attribute `lphk` also increases. In contrast to the first finding there seems to be no correlation between the attributes `height` and `lphk`. The data points are spread out more or less evenly without a clear correlation.

5. Linear regression

```
# create linear model_weight using "lphk"|"weight"
model_weight <- lm(lphk ~ weight, data = mpg2)
summary(model_weight)
```

```
##
## Call:
## lm(formula = lphk ~ weight, data = mpg2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9567 -0.3131 -0.0184  0.5239  3.2257
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3162155  0.3374467   0.937     0.35
## weight      0.0030038  0.0001292  23.255   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9597 on 197 degrees of freedom
## Multiple R-squared:  0.733,  Adjusted R-squared:  0.7316
## F-statistic: 540.8 on 1 and 197 DF,  p-value: < 2.2e-16
```

```r
# create linear model_height using "lphk"|"height"
model_height <- lm(lphk ~ height, data = mpg2)
summary(model_height)
```

```
##
## Call:
## lm(formula = lphk ~ height, data = mpg2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.502 -1.471 -0.079  1.217  6.538
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.54138    2.94301   0.864   0.3889
## height       0.10147    0.05463   1.857   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.841 on 197 degrees of freedom
## Multiple R-squared:  0.01721,    Adjusted R-squared:  0.01222
## F-statistic:  3.45 on 1 and 197 DF,  p-value: 0.06475
```

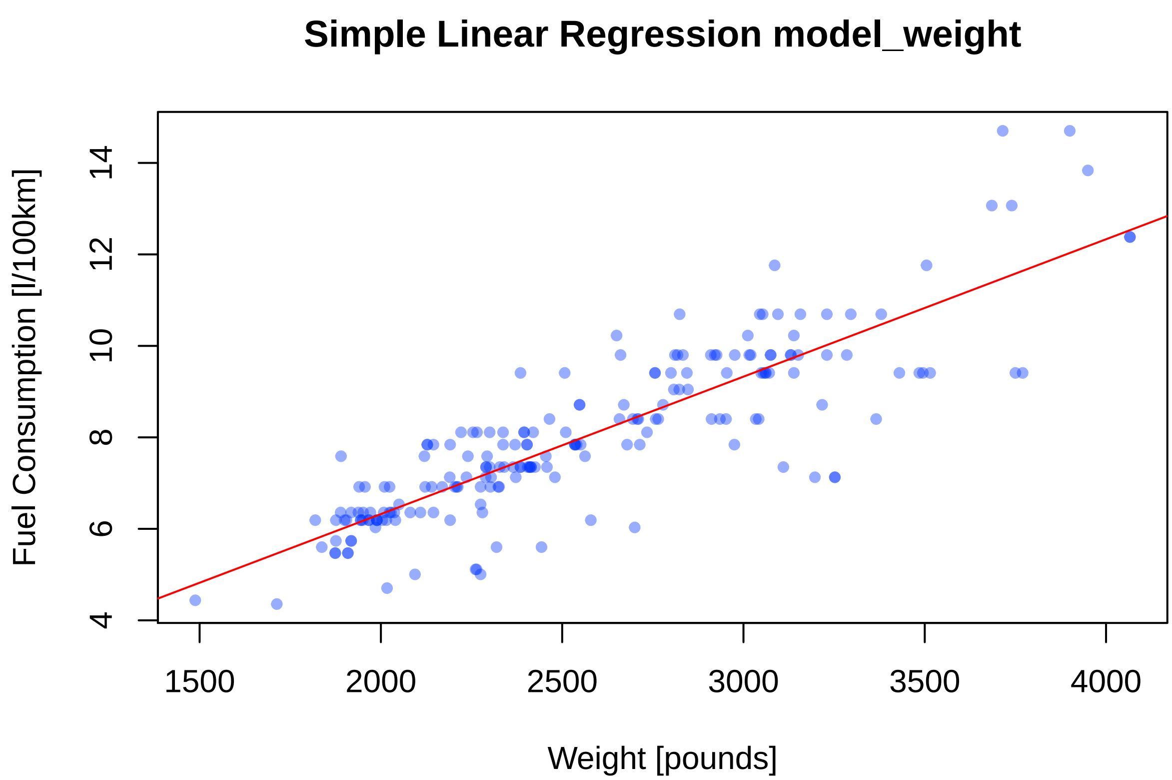
```r
# create linear model_weight_height using "lphk"|("weight", "height")
model_weight_height <- lm(lphk ~ weight+height, data = mpg2)
summary(model_weight_height)
```

```
##
## Call:
## lm(formula = lphk ~ weight + height, data = mpg2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7553 -0.3536  0.0151  0.5296  3.2293
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.5084790  1.4956955   3.683 0.000298 ***
## weight       0.0031418  0.0001314  23.917  < 2e-16 ***
## height      -0.1030426  0.0289607  -3.558 0.000469 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9325 on 196 degrees of freedom
## Multiple R-squared:  0.7492, Adjusted R-squared:  0.7466
## F-statistic: 292.7 on 2 and 196 DF,  p-value: < 2.2e-16
```

Comparing the two bivariate models `model_weight` and `model_weight` we can clearly see the superiority of `model_weight` with R^2 0.733 over `model_weight` with R^2 0.0172. While comparing `model_weight` with the multivariate model `model_weight_height` we have to use the adjusted R^2. Therefore `model_weight_height`'s adjusted R^2 is 0.747 is slightly higher compared to `model_weight`'s adjusted R^2 0.732. Hence, I would prefer to use `model_weight_height` due the better performance with the assumption that obtaining the additional variable `height` is not linked with additional costs.

6. Plot data with regression line

```
# scatterplot with regression line of model_weight
plot(mpg2[c("weight","lphk")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "Simple Linear Regression model_weight",
      xlab = "Weight [pounds]", ylab = "Fuel Consumption [l/100km]")
abline(model_weight, lwd = 1, col = "red")
```



7. Predict data with models

```
# predict using model_weight ("lphk"|"weight")
predict(model_weight, newdata = data.frame(weight = 2750))
```

```
##        1
## 8.576558
```

```r
# predict using model_weight_heigth ("lphk"|("weight", "height"))
predict(model_weight_height, newdata = data.frame(weight = 2750, height = 55))
```

```
##        1
## 8.480999
```

Predicting the fuel consumption `lphk` with the model `model_weight` for a car with the weight of 2750 pounds results in 8.58 liters per 100 km. While the predicted fuel consumption `lphk` with the model `model_weight_height` (`weight` = 2750) results in 8.48 liters per 100 km.

8. Create linear regression model with all attributes

```r
# create linear model using all attributes
model_all <- lm(lphk ~ ., data = mpg2)
summary(model_all)
```

```
##
## Call:
## lm(formula = lphk ~ ., data = mpg2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8552 -0.4169 -0.0113  0.3422  2.2134
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.9186556  3.5481036   1.386 0.167295
## wb           0.0307424  0.0253713   1.212 0.227141
## length       0.0179790  0.0133935   1.342 0.181086
## width       -0.0563829  0.0616953  -0.914 0.361937
## height      -0.0244138  0.0359745  -0.679 0.498195
## weight       0.0013023  0.0003747   3.476 0.000631 ***
## eng.cc       0.0091018  0.0033891   2.686 0.007885 **
## bore        -0.6737750  0.2922415  -2.306 0.022222 *
## stroke      -0.6371891  0.1993171  -3.197 0.001629 **
## hp           0.0169020  0.0031222   5.413 1.86e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8084 on 189 degrees of freedom
## Multiple R-squared:  0.8182, Adjusted R-squared:  0.8096
## F-statistic: 94.52 on 9 and 189 DF,  p-value: < 2.2e-16
```

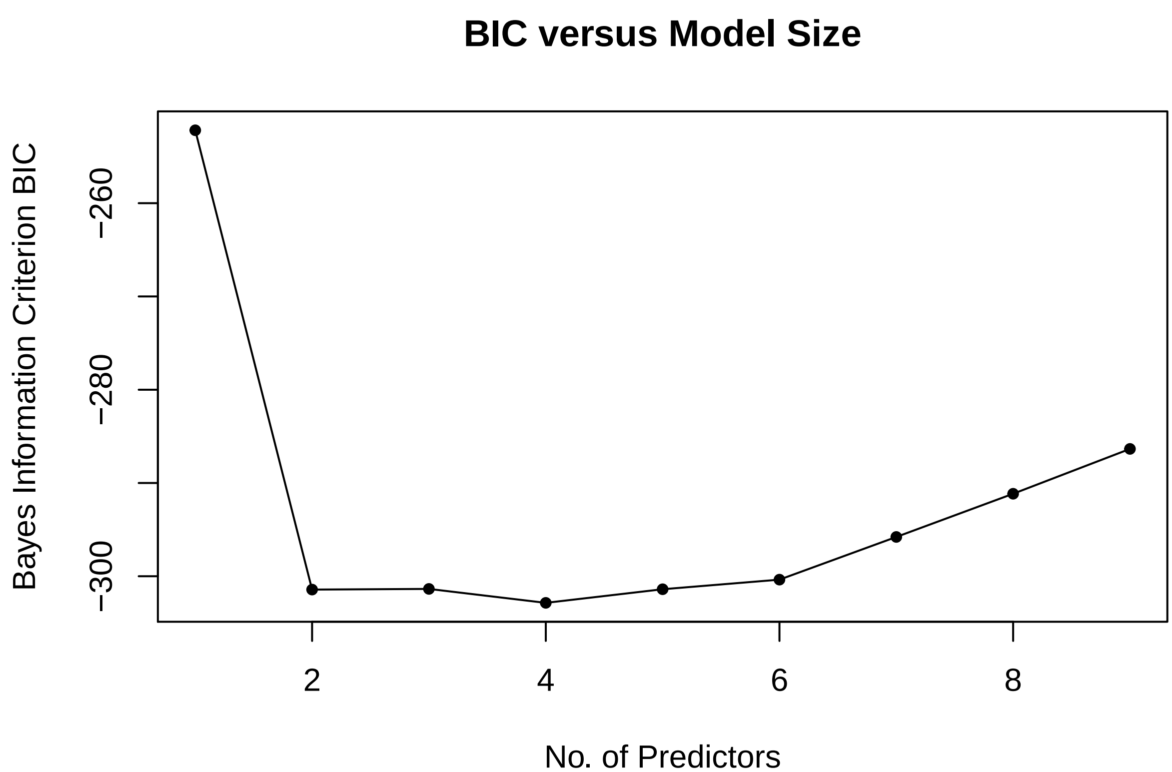
The `model_all` seems to predict `lphk` much better than the bivariate `model_weight`, achieving an adjusted R^2 of 0.8096 compared to adjusted R^2 of 0.7316 respectively. Due to the better performance of `model_all` based on the adjusted R^2 metric, I would choose `model_all` over `model_weight`. The regression coefficients of `model_weight` show that `lphk` is positively correlated with the attributes `wb`, `widthv`, `height`, `weight`, `eng.cc` and negatively correlated with the attributes `hw.mpg`, `length`, `bore`, `stroke`, `hp`.

9. Search for best variables and create best model

```
# calculate best predictor variables for model using n variables
regss <- regsubsets(lphk ~ ., data = mpg2, nbest = 1, nvmax = 9,
                    intercept = TRUE, method = "exhaustive")
# results
sum_regss <- summary(regss)
sum_regss$which
```

```
##   (Intercept)    wb length width height weight eng.cc  bore stroke    hp
## 1        TRUE FALSE  FALSE FALSE  FALSE   TRUE  FALSE FALSE  FALSE FALSE
## 2        TRUE FALSE  FALSE FALSE  FALSE   TRUE  FALSE FALSE  FALSE  TRUE
## 3        TRUE FALSE  FALSE FALSE  FALSE   TRUE  FALSE FALSE   TRUE  TRUE
## 4        TRUE FALSE  FALSE FALSE  FALSE   TRUE   TRUE FALSE   TRUE  TRUE
## 5        TRUE FALSE  FALSE FALSE  FALSE   TRUE   TRUE  TRUE   TRUE  TRUE
## 6        TRUE FALSE   TRUE FALSE  FALSE   TRUE   TRUE  TRUE   TRUE  TRUE
## 7        TRUE  TRUE   TRUE FALSE  FALSE   TRUE   TRUE  TRUE   TRUE  TRUE
## 8        TRUE  TRUE   TRUE  TRUE  FALSE   TRUE   TRUE  TRUE   TRUE  TRUE
## 9        TRUE  TRUE   TRUE  TRUE   TRUE   TRUE   TRUE  TRUE   TRUE  TRUE
```

```
# plot BIC versus Model Size
plot(x = apply(sum_regss$which, 1, sum) - 1,
      y = sum_regss$bic, pch = 20, type = "o", main = "BIC versus Model Size",
      xlab = "No. of Predictors", ylab = "Bayes Information Criterion BIC")
```



```r
# create best model according to "BIC versus Model Size"
model_best <- lm(lphk ~ weight+eng.cc+stroke+hp, data = mpg2)
(summary(model_best))
```

```
##
## Call:
## lm(formula = lphk ~ weight + eng.cc + stroke + hp, data = mpg2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1996 -0.3869  0.0020  0.3515  2.4271
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.8556198  0.6654773   4.291 2.80e-05 ***
## weight       0.0016994  0.0002173   7.821 3.27e-13 ***
## eng.cc       0.0085761  0.0033059   2.594  0.01021 *
## stroke      -0.5295017  0.1930823  -2.742  0.00667 **
## hp           0.0136463  0.0026539   5.142 6.61e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8182 on 194 degrees of freedom
## Multiple R-squared:  0.8089, Adjusted R-squared:  0.8049
## F-statistic: 205.3 on 4 and 194 DF,  p-value: < 2.2e-16
```

After computing the BIC versus the number of predictors the optimal number of predictors (minimizing BIC) is 4. The best variables for predicting `lphk` in a linear regression model are `weight`, `eng.cc`, `stroke` and `hp`.

Table 1: Comparison of the calculated linear regression models.

Regression model	**adjusted R^2**	**n var**
model_weight	0.7316	1
model_all	0.8096	9
model_best	0.8049	4

As shown in Table 1 regression model `model_all` has a slightly higher adj. R^2 0.8096 compared to `model_best` with an adj. R^2 of 0.8049 and both show a better performance than `model_weight`'s adjusted R^2 of 0.7316. Additionally all variables in `model_best` all predictor variables are significant. Considering that we strive to choose a regression model that is as simple as possible given a good performance, I would choose `model_best` as the model for production. `model_best` performs similarly to `model_all`, despite using 5 predictor variables less. Looking at the regression coefficients we can see that `lphk` is positively correlated with the attributes `weight` `eng.cc` and `hp`, which makes sense since the heavier and more powerful a car the higher the fuel consumption. However fuel consumption `lphk` is negatively correlated with the attribute `stroke`, that implies the higher the number of strokes a car has the less it consumes.

10. Plot multiple regression model

```r
# predict values using model_best
mpg2["pred"] <- predict(model_best, newdata = +
                        mpg2[c("weight", "eng.cc", "stroke", "hp")])

# scatterplot true Response Values vs Predicted Values
plot(mpg2[c("pred","lphk")], col = rgb(0,0.2,1,0.4), pch = 20,
      main = "True Response Values vs Predicted Values",
      xlab = "Predicted Fuel Consumption [l/100km]",
      ylab = "Observed Fuel Consumption [l/100km]")
abline(a = 0, b = 1, col = "red")
```