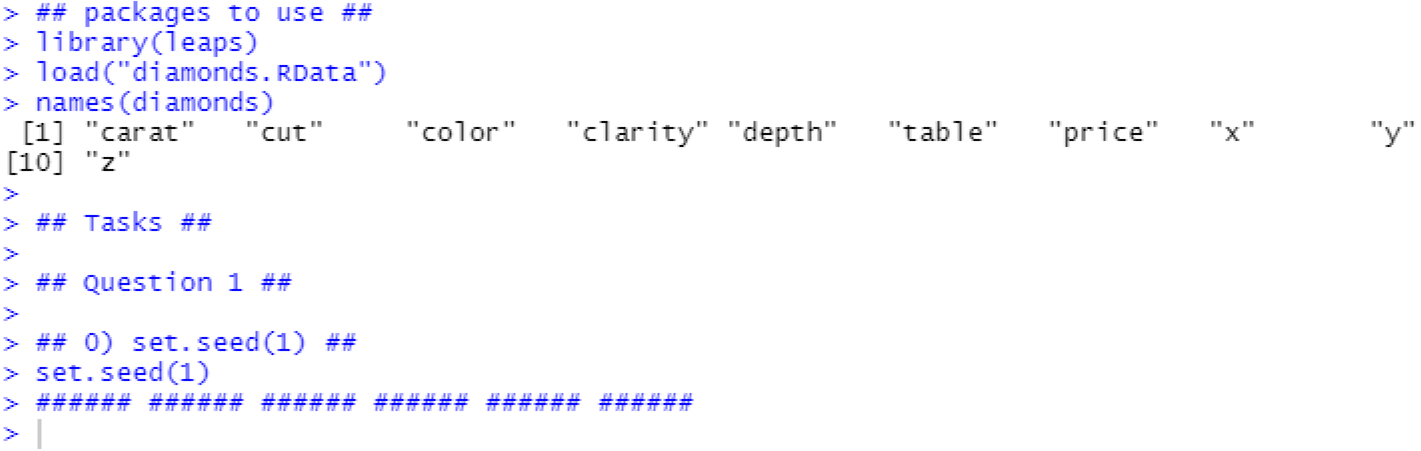
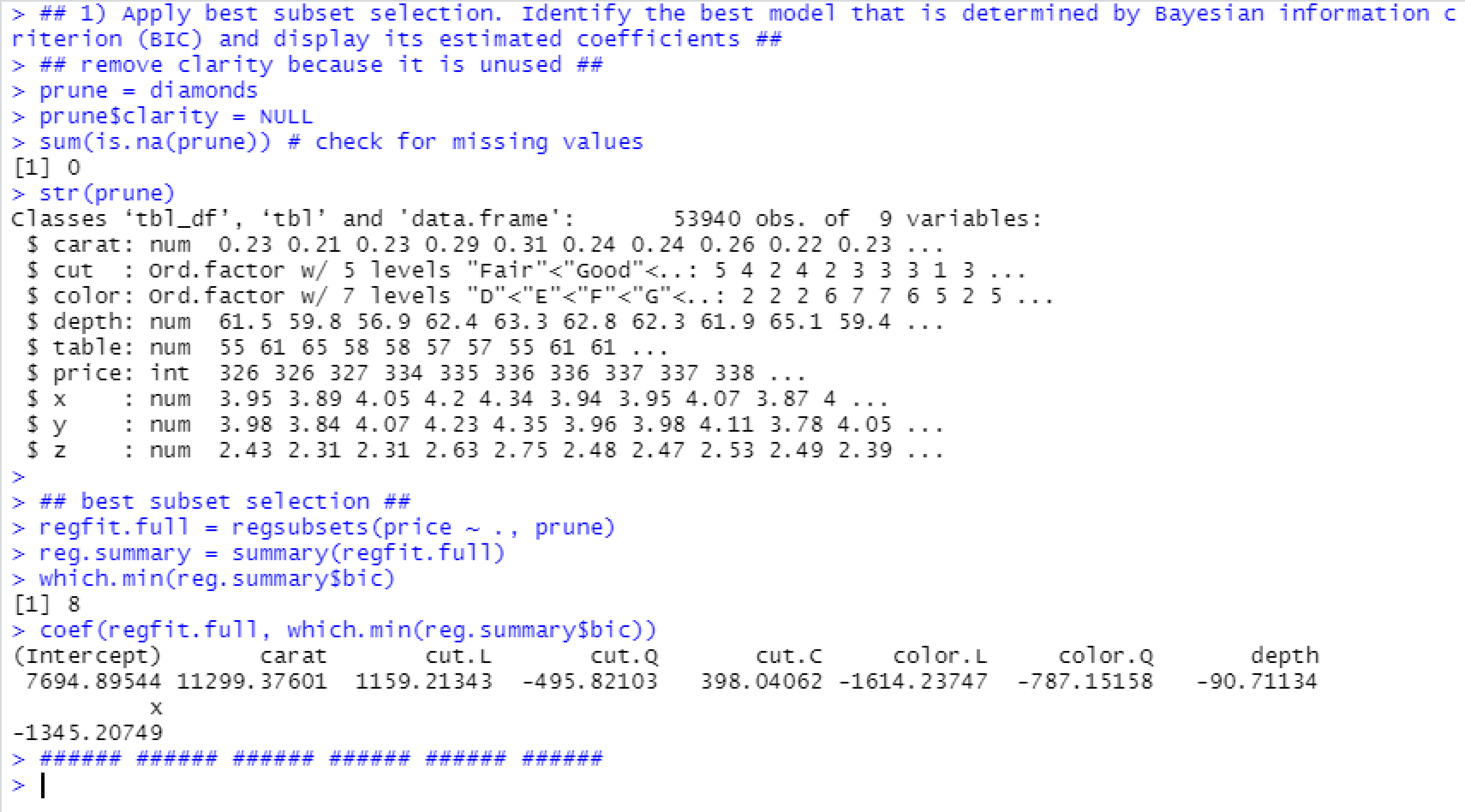
Tasks

Please use the data set `Auto` from the R library `ISLR`. You will build linear models to predict the response `mpg` by choosing the best from 7 predictors `cylinders`, `displacement`, `horsepower`, `weight`, `acceleration`, `year` and `origin`. Whenever random splitting of or random sampling from the data set is needed, please `set.seed(1)`.



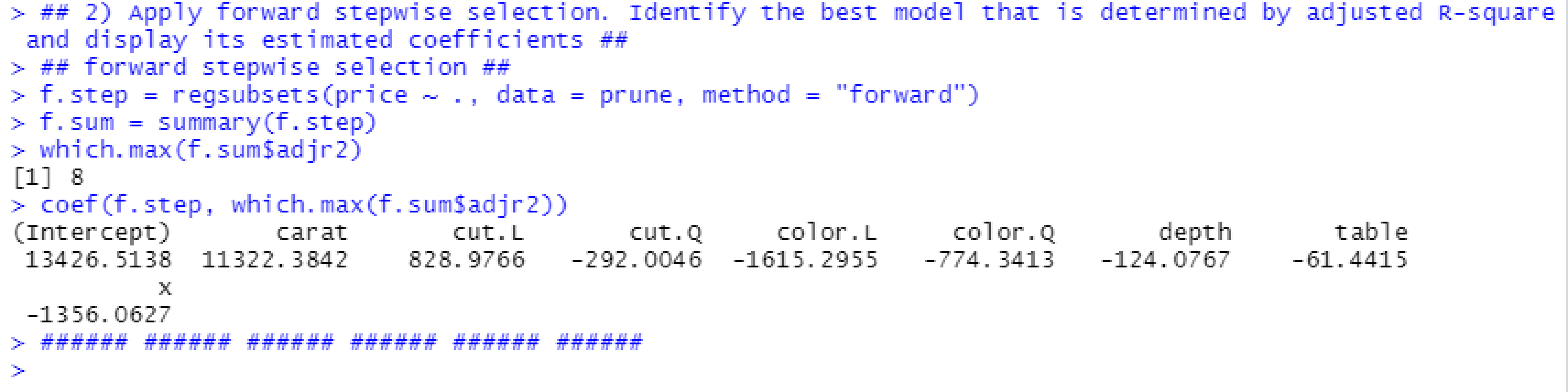
```
> ## packages to use ##
> library(leaps)
> load("diamonds.RData")
> names(diamonds)
 [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"   "x"       "y"      
[10] "z"      
> 
> ## Tasks ##
> 
> ## Question 1 ##
> 
> ## 0) set.seed(1) ##
> set.seed(1)
> ###### ###### ###### ###### ###### ######
> 
```

(1) Apply best subset selection. Identify the best model that is determined by Bayesian information criterion (BIC) and display its estimated coefficients.



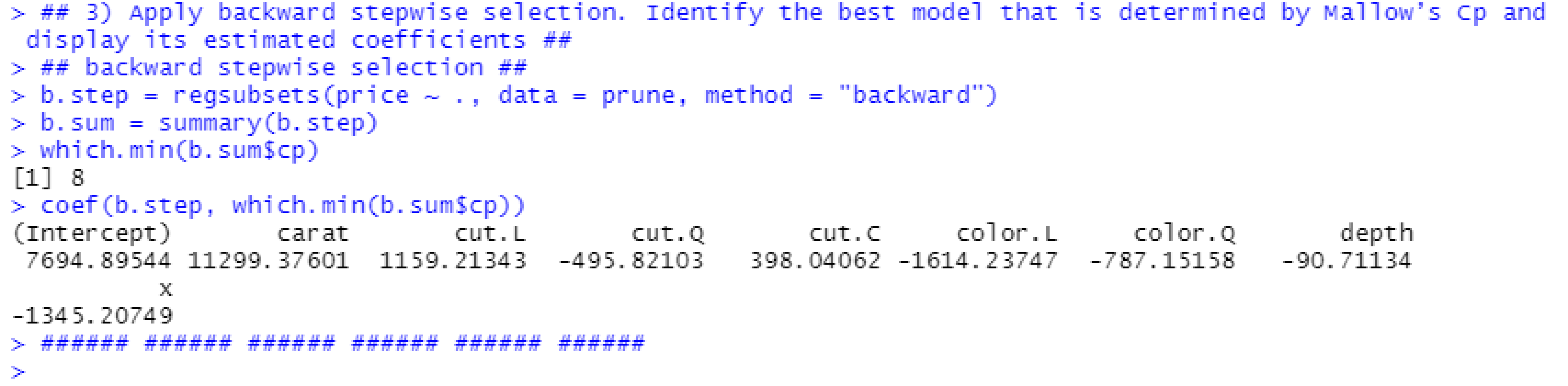
```
> ## 1) Apply best subset selection. Identify the best model that is determined by Bayesian information c
riterion (BIC) and display its estimated coefficients ##
> ## remove clarity because it is unused ##
> prune = diamonds
> prune$clarity = NULL
> sum(is.na(prune)) # check for missing values
[1] 0
> str(prune)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       53940 obs. of  9 variables:
 $ carat: num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut  : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color: Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
 $ depth: num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table: num  55 61 65 58 58 57 57 55 61 61 ...
 $ price: int  326 326 327 334 335 336 336 337 337 338 ...
 $ x    : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y    : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z    : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
>
> ## best subset selection ##
> regfit.full = regsubsets(price ~ ., prune)
> reg.summary = summary(regfit.full)
> which.min(reg.summary$bic)
[1] 8
> coef(regfit.full, which.min(reg.summary$bic))
(Intercept)       carat       cut.L       cut.Q       cut.C     color.L     color.Q       depth 
 7694.89544 11299.37601  1159.21343  -495.82103   398.04062 -1614.23747  -787.15158   -90.71134 
          x 
-1345.20749 
> ###### ###### ###### ###### ###### ######
> 
```

(2) Apply forward stepwise selection. Identify the best model that is determined by adjusted R-square and display its estimated coefficients.



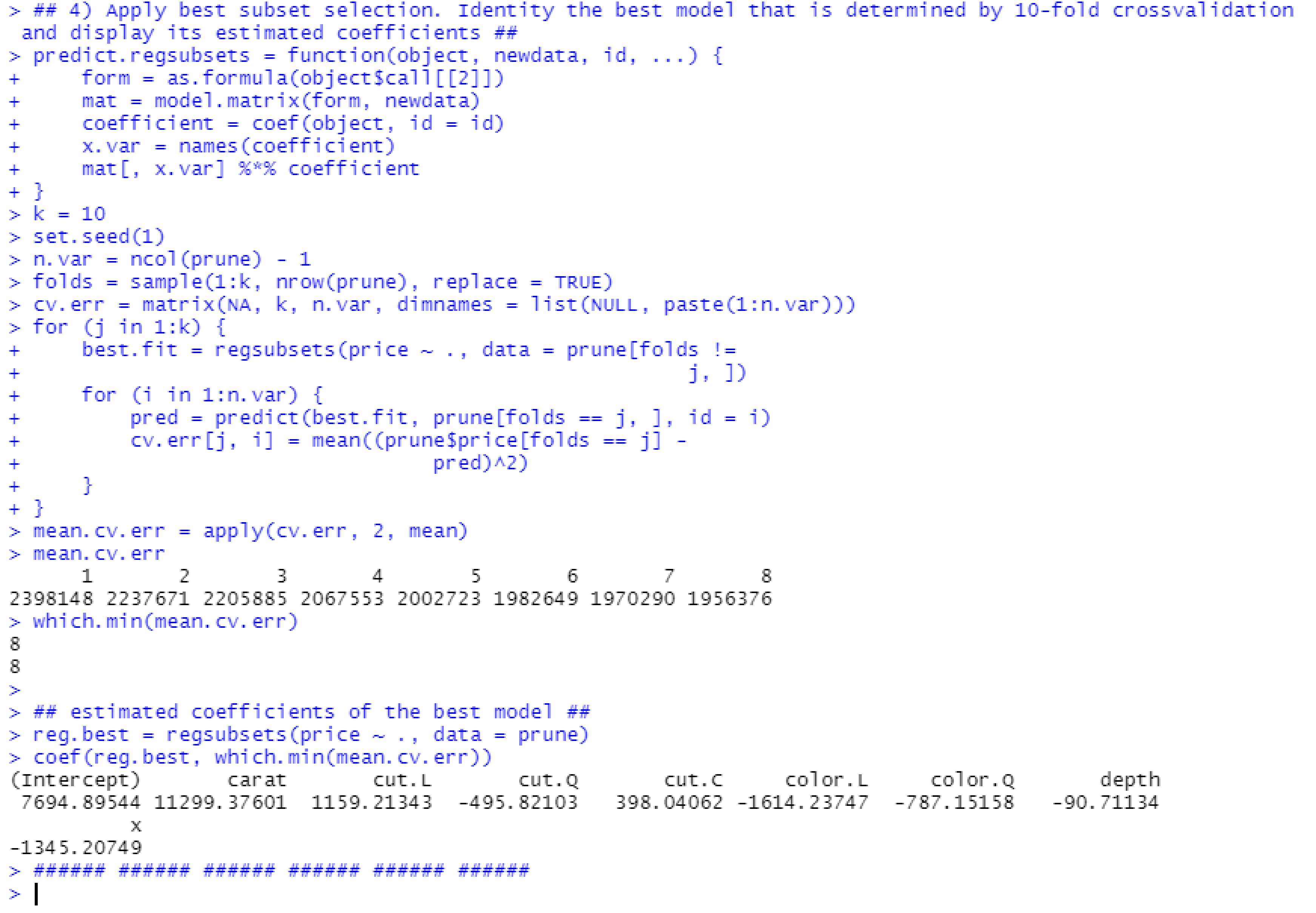
```
> ## 2) Apply forward stepwise selection. Identify the best model that is determined by adjusted R-square
 and display its estimated coefficients ##
> ## forward stepwise selection ##
> f.step = regsubsets(price ~ ., data = prune, method = "forward")
> f.sum = summary(f.step)
> which.max(f.sum$adjr2)
[1] 8
> coef(f.step, which.max(f.sum$adjr2))
(Intercept)       carat       cut.L       cut.Q     color.L     color.Q       depth       table 
 13426.5138  11322.3842    828.9766   -292.0046  -1615.2955   -774.3413   -124.0767    -61.4415 
          x 
 -1356.0627 
> ###### ###### ###### ###### ###### ######
> 
```

(3) Apply backward stepwise selection. Identify the best model that is determined by Mallow's C_p and display its estimated coefficients.



```
> ## 3) Apply backward stepwise selection. Identify the best model that is determined by Mallow's Cp and
 display its estimated coefficients ##
> ## backward stepwise selection ##
> b.step = regsubsets(price ~ ., data = prune, method = "backward")
> b.sum = summary(b.step)
> which.min(b.sum$cp)
[1] 8
> coef(b.step, which.min(b.sum$cp))
(Intercept)       carat       cut.L       cut.Q       cut.C     color.L     color.Q       depth 
 7694.89544 11299.37601  1159.21343  -495.82103   398.04062 -1614.23747  -787.15158   -90.71134 
          x 
-1345.20749 
> ###### ###### ###### ###### ###### ######
> 
```

(4) Apply best subset selection. Identity the best model that is determined by 10-fold cross-validation and display its estimated coefficients.



```
> ## 4) Apply best subset selection. Identity the best model that is determined by 10-fold crossvalidation
 and display its estimated coefficients ##
> predict.regsubsets = function(object, newdata, id, ...) {
+     form = as.formula(object$call[[2]])
+     mat = model.matrix(form, newdata)
+     coefficient = coef(object, id = id)
+     x.var = names(coefficient)
+     mat[, x.var] %*% coefficient
+ }
> k = 10
> set.seed(1)
> n.var = ncol(prune) - 1
> folds = sample(1:k, nrow(prune), replace = TRUE)
> cv.err = matrix(NA, k, n.var, dimnames = list(NULL, paste(1:n.var)))
> for (j in 1:k) {
+     best.fit = regsubsets(price ~ ., data = prune[folds !=
+                                                      j, ])
+     for (i in 1:n.var) {
+         pred = predict(best.fit, prune[folds == j, ], id = i)
+         cv.err[j, i] = mean((prune$price[folds == j] -
+                                   pred)^2)
+     }
+ }
> mean.cv.err = apply(cv.err, 2, mean)
> mean.cv.err
      1       2       3       4       5       6       7       8 
2398148 2237671 2205885 2067553 2002723 1982649 1970290 1956376 
> which.min(mean.cv.err)
8 
8 
> 
> ## estimated coefficients of the best model ##
> reg.best = regsubsets(price ~ ., data = prune)
> coef(reg.best, which.min(mean.cv.err))
(Intercept)       carat       cut.L       cut.Q       cut.C     color.L     color.Q       depth 
 7694.89544 11299.37601  1159.21343  -495.82103   398.04062 -1614.23747  -787.15158   -90.71134 
          x 
-1345.20749 
> ###### ###### ###### ###### ###### ######
> 
```