Stat 6021: Guided Question Set 9

Tom Lever

10/30/22

For this guided question set, we will use the data set `nfl.txt`, which contains data on NFL team performance from the 1976 season. The variables are:

- y: Games won in the 14-game 1976 season
- x_1: Rushing yards
- x_2: Passing yards
- x_3: Punting average (yards / punt)
- x_4: Field-goal percentage (field goals made / field goals attempted)
- x_5: Turnover differential (turnovers acquired - turnovers lost)
- x_6: Penalty yards
- x_7: Percent rushing (rushing plays / total plays)
- x_8: Opponents' rushing yards
- x_9: Opponents' passing yards

1. Use the `regsubsets` function from the `leaps` package to run all possible regressions. Set `nbest` to 2.

```
library(leaps)
data_set <- read.table(
    "../../Module_6--Introduction_to_Multiple_Linear_Regression/Guided_Question_Set/nfl.txt",
    header = TRUE
)
head(data_set, n = 3)
```

```
##    y   x1   x2   x3   x4 x5  x6   x7   x8   x9
## 1 10 2113 1985 38.9 64.7  4 868 59.7 2205 1917
## 2 11 2003 2855 38.8 61.3  3 615 55.0 2096 1575
## 3 11 2957 1737 40.1 60.0 14 914 65.6 1847 2175
```

```
nrow(data_set)
```

```
## [1] 28
```

```
subset_selection_object <- regsubsets(y ~ ., data = data_set, nbest = 2)
summary_for_subset_selection_object <- summary(subset_selection_object)
summary_for_subset_selection_object
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = data_set, nbest = 2)
## 9 Variables  (and intercept)
##    Forced in Forced out
## x1     FALSE      FALSE
## x2     FALSE      FALSE
## x3     FALSE      FALSE
## x4     FALSE      FALSE
## x5     FALSE      FALSE
```

```
## x6     FALSE      FALSE
## x7     FALSE      FALSE
## x8     FALSE      FALSE
## x9     FALSE      FALSE
## 2 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4  x5  x6  x7  x8  x9
## 1  ( 1 ) " " " " " " " " " " " " " " "*" " "
## 1  ( 2 ) "*" " " " " " " " " " " " " " " " "
## 2  ( 1 ) " " "*" " " " " " " " " " " "*" " "
## 2  ( 2 ) " " "*" " " " " " " " " "*" " " " "
## 3  ( 1 ) " " "*" " " " " " " " " "*" "*" " "
## 3  ( 2 ) "*" "*" " " " " " " " " " " "*" " "
## 4  ( 1 ) " " "*" " " " " " " " " "*" "*" "*"
## 4  ( 2 ) "*" "*" " " " " " " " " " " "*" "*"
## 5  ( 1 ) "*" "*" " " " " " " " " "*" "*" "*"
## 5  ( 2 ) " " "*" " " "*" " " " " "*" "*" "*"
## 6  ( 1 ) " " "*" "*" "*" " " " " "*" "*" "*"
## 6  ( 2 ) "*" "*" " " "*" " " " " "*" "*" "*"
## 7  ( 1 ) " " "*" "*" "*" " " "*" "*" "*" "*"
## 7  ( 2 ) "*" "*" " " "*" " " "*" "*" "*" "*"
## 8  ( 1 ) "*" "*" "*" "*" " " "*" "*" "*" "*"
## 8  ( 2 ) " " "*" "*" "*" "*" "*" "*" "*" "*"
```

2. Identify the multiple linear regression model that is best in terms of

   (a) Adjusted R^2

```
adjusted_R2 <- summary_for_subset_selection_object$adjr2
index_of_model_with_maximum_adjusted_R2 <- which.max(adjusted_R2)
index_of_model_with_maximum_adjusted_R2
```

```
## [1] 7
```

```
matrix_of_models <- summary_for_subset_selection_object$outmat
matrix_of_models[index_of_model_with_maximum_adjusted_R2, ]
```

```
##  x1  x2  x3  x4  x5  x6  x7  x8  x9
## " " "*" " " " " " " " " "*" "*" "*"
```

```
coef(subset_selection_object, index_of_model_with_maximum_adjusted_R2)
```

```
##  (Intercept)           x2           x7           x8           x9
## -1.821703427  0.003818572  0.216894094 -0.004014887 -0.001634926
```

   (b) Mallow's C_p

```
Cp <- summary_for_subset_selection_object$cp
index_of_model_with_minimum_Cp <- which.min(Cp)
index_of_model_with_minimum_Cp
```

```
## [1] 5
```

```
matrix_of_models[index_of_model_with_minimum_Cp, ]
```

```
##  x1  x2  x3  x4  x5  x6  x7  x8  x9
## " " "*" " " " " " " " " "*" "*" " "
```

```
coef(subset_selection_object, index_of_model_with_minimum_Cp)
```

```
##   (Intercept)           x2           x7           x8
## -1.808372059  0.003598070  0.193960210 -0.004815494
```

(c) Schwartz Bayesian Information Criterion ($BIC_{Schwartz}$)

```
BICSchwartz <- summary_for_subset_selection_object$bic
index_of_model_with_minimum_BICSchwartz <- which.min(BICSchwartz)
index_of_model_with_minimum_BICSchwartz
```

```
## [1] 5
```

```
matrix_of_models[index_of_model_with_minimum_BICSchwartz, ]
```

```
##  x1  x2  x3  x4  x5  x6  x7  x8  x9
## " " "*" " " " " " " " " "*" "*" " "
```

```
coef(subset_selection_object, index_of_model_with_minimum_BICSchwartz)
```

```
##   (Intercept)           x2           x7           x8
## -1.808372059  0.003598070  0.193960210 -0.004815494
```

3. Run forward selection, starting with an intercept-only model. Report the predictors and the estimated coefficients of the model selected.

```
intercept_only_model <- lm(y ~ 1, data = data_set)
full_model <- lm(y ~ ., data = data_set)
step(
    intercept_only_model,
    scope = list(lower = intercept_only_model, upper = full_model),
    direction = "forward"
)
```

```
## Start:  AIC=70.81
## y ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + x8    1   178.092 148.87 50.785
## + x1    1   115.068 211.90 60.669
## + x7    1    97.238 229.73 62.931
## + x5    1    86.116 240.85 64.255
## + x2    1    76.193 250.77 65.385
## + x9    1    30.167 296.80 70.104
## <none>              326.96 70.814
## + x4    1    21.844 305.12 70.878
## + x6    1    16.411 310.55 71.372
## + x3    1     2.135 324.83 72.631
##
## Step:  AIC=50.78
## y ~ x8
##
##        Df Sum of Sq     RSS    AIC
## + x2    1    64.934  83.938 36.741
## + x5    1    11.607 137.265 50.512
## <none>              148.872 50.785
## + x1    1     6.636 142.236 51.508
## + x3    1     6.368 142.504 51.561
## + x4    1     6.345 142.527 51.565
## + x7    1     0.974 147.898 52.601
```

```
## + x6    1     0.487 148.385 52.693
## + x9    1     0.008 148.864 52.783
##
## Step:  AIC=36.74
## y ~ x8 + x2
##
##        Df Sum of Sq    RSS    AIC
## + x7    1   14.0682 69.870 33.604
## + x1    1   11.1905 72.748 34.734
## + x3    1    8.9010 75.037 35.602
## + x5    1    5.8147 78.124 36.730
## <none>              83.938 36.741
## + x9    1    2.0256 81.913 38.057
## + x6    1    1.3216 82.617 38.296
## + x4    1    0.0161 83.922 38.735
##
## Step:  AIC=33.6
## y ~ x8 + x2 + x7
##
##        Df Sum of Sq    RSS    AIC
## + x9    1    4.8657 65.004 33.583
## <none>              69.870 33.604
## + x3    1    1.3873 68.483 35.043
## + x4    1    0.9792 68.891 35.209
## + x1    1    0.9022 68.968 35.240
## + x6    1    0.4879 69.382 35.408
## + x5    1    0.2987 69.571 35.484
##
## Step:  AIC=33.58
## y ~ x8 + x2 + x7 + x9
##
##        Df Sum of Sq    RSS    AIC
## <none>              65.004 33.583
## + x1    1   1.86452 63.140 34.768
## + x4    1   1.74260 63.262 34.822
## + x3    1   0.70148 64.303 35.279
## + x6    1   0.45071 64.554 35.388
## + x5    1   0.32667 64.678 35.442

##
## Call:
## lm(formula = y ~ x8 + x2 + x7 + x9, data = data_set)
##
## Coefficients:
## (Intercept)           x8           x2           x7           x9
##   -1.821703    -0.004015     0.003819     0.216894    -0.001635
```

4. Run backward elimination, starting with the model with all predictors. Report the predictors and the estimated coefficients of the model selected.

```
step(
    full_model,
    scope = list(lower = intercept_only_model, upper = full_model),
    direction = "backward"
)
```

```
## Start:  AIC=41.48
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x5    1     0.000  60.293 39.476
## - x1    1     0.549  60.842 39.730
## - x3    1     0.746  61.039 39.821
## - x6    1     0.803  61.096 39.847
## - x4    1     1.968  62.261 40.376
## - x7    1     3.451  63.744 41.035
## <none>               60.293 41.476
## - x9    1     5.348  65.642 41.856
## - x8    1    12.072  72.365 44.587
## - x2    1    62.448 122.741 59.380
##
## Step:  AIC=39.48
## y ~ x1 + x2 + x3 + x4 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     0.553  60.846 37.732
## - x3    1     0.750  61.043 37.822
## - x6    1     0.818  61.111 37.854
## - x4    1     2.053  62.346 38.414
## - x7    1     3.859  64.152 39.213
## <none>               60.293 39.476
## - x9    1     5.351  65.644 39.857
## - x8    1    12.086  72.379 42.592
## - x2    1    66.979 127.272 58.395
##
## Step:  AIC=37.73
## y ~ x2 + x3 + x4 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x6    1     0.690  61.536 36.048
## - x3    1     1.715  62.561 36.510
## - x4    1     3.051  63.897 37.102
## <none>               60.846 37.732
## - x9    1     4.852  65.698 37.880
## - x7    1     8.961  69.807 39.579
## - x8    1    16.599  77.445 42.486
## - x2    1    67.010 127.856 56.524
##
## Step:  AIC=36.05
## y ~ x2 + x3 + x4 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x3    1     1.726  63.262 34.822
## - x4    1     2.767  64.303 35.279
## <none>               61.536 36.048
## - x9    1     4.831  66.367 36.164
## - x7    1     9.390  70.926 38.024
## - x8    1    18.314  79.851 41.343
## - x2    1    66.447 127.984 54.552
##
```

```
## Step:  AIC=34.82
## y ~ x2 + x4 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x4    1     1.743  65.004 33.583
## <none>               63.262 34.822
## - x9    1     5.629  68.891 35.209
## - x8    1    17.701  80.962 39.730
## - x7    1    18.583  81.845 40.033
## - x2    1    75.598 138.860 54.835
##
## Step:  AIC=33.58
## y ~ x2 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## <none>               65.004 33.583
## - x9    1     4.866  69.870 33.604
## - x7    1    16.908  81.913 38.057
## - x8    1    23.299  88.303 40.160
## - x2    1    82.892 147.897 54.601
```

```
##
## Call:
## lm(formula = y ~ x2 + x7 + x8 + x9, data = data_set)
##
## Coefficients:
## (Intercept)           x2           x7           x8           x9
##   -1.821703     0.003819     0.216894    -0.004015    -0.001635
```

5. Run stepwise regression, starting with an intercept-only model. Report the predictors and the estimated coefficients of the model selected.

```
step(
    intercept_only_model,
    scope = list(lower = intercept_only_model, upper = full_model),
    direction = "both"
)
```

```
## Start:  AIC=70.81
## y ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + x8    1   178.092 148.87 50.785
## + x1    1   115.068 211.90 60.669
## + x7    1    97.238 229.73 62.931
## + x5    1    86.116 240.85 64.255
## + x2    1    76.193 250.77 65.385
## + x9    1    30.167 296.80 70.104
## <none>              326.96 70.814
## + x4    1    21.844 305.12 70.878
## + x6    1    16.411 310.55 71.372
## + x3    1     2.135 324.83 72.631
##
## Step:  AIC=50.78
## y ~ x8
##
```

```
##        Df Sum of Sq    RSS    AIC
## + x2    1    64.934  83.94 36.741
## + x5    1    11.607 137.27 50.512
## <none>              148.87 50.785
## + x1    1     6.636 142.24 51.508
## + x3    1     6.368 142.50 51.561
## + x4    1     6.345 142.53 51.565
## + x7    1     0.974 147.90 52.601
## + x6    1     0.487 148.39 52.693
## + x9    1     0.008 148.86 52.783
## - x8    1   178.092 326.96 70.814
##
## Step:  AIC=36.74
## y ~ x8 + x2
##
##        Df Sum of Sq     RSS    AIC
## + x7    1    14.068  69.870 33.604
## + x1    1    11.190  72.748 34.734
## + x3    1     8.901  75.037 35.602
## + x5    1     5.815  78.124 36.730
## <none>               83.938 36.741
## + x9    1     2.026  81.913 38.057
## + x6    1     1.322  82.617 38.296
## + x4    1     0.016  83.922 38.735
## - x2    1    64.934 148.872 50.785
## - x8    1   166.833 250.771 65.385
##
## Step:  AIC=33.6
## y ~ x8 + x2 + x7
##
##        Df Sum of Sq     RSS    AIC
## + x9    1     4.866  65.004 33.583
## <none>               69.870 33.604
## + x3    1     1.387  68.483 35.043
## + x4    1     0.979  68.891 35.209
## + x1    1     0.902  68.968 35.240
## + x6    1     0.488  69.382 35.408
## + x5    1     0.299  69.571 35.484
## - x7    1    14.068  83.938 36.741
## - x8    1    41.400 111.270 44.633
## - x2    1    78.028 147.898 52.601
##
## Step:  AIC=33.58
## y ~ x8 + x2 + x7 + x9
##
##        Df Sum of Sq     RSS    AIC
## <none>               65.004 33.583
## - x9    1     4.866  69.870 33.604
## + x1    1     1.865  63.140 34.768
## + x4    1     1.743  63.262 34.822
## + x3    1     0.701  64.303 35.279
## + x6    1     0.451  64.554 35.388
## + x5    1     0.327  64.678 35.442
## - x7    1    16.908  81.913 38.057
```

```
## - x8    1    23.299  88.303 40.160
## - x2    1    82.892 147.897 54.601
```

```
##
## Call:
## lm(formula = y ~ x8 + x2 + x7 + x9, data = data_set)
##
## Coefficients:
## (Intercept)           x8           x2           x7           x9
##   -1.821703    -0.004015     0.003819     0.216894    -0.001635
```

6. The PRESS statistic can be used as a criterion in model validation as well as model selection. Unfortunately, the `regsubsets` function from the `leaps` package does not compute the `PRESS` statistic. The `PRESS` statistic can be written as

$$PRESS = \sum_{i=1}^{n} \left[\left(y_i - \hat{y}_{(i)} \right)^2 \right]$$

$$PRESS = \sum_{i=1}^{n} \left[\left(\frac{e_i}{1 - h_{ii}} \right)^2 \right]$$

where h_{ii} denotes the ith diagonal element of the hat matrix. Write a function that computes the PRESS statistic for a regression model. Hint: the diagonal elements from the hat matrix can be found using the `lm.influence` function.

```r
library(TomLeversRPackage)
calculate_PRESS(full_model)
```

```
## [1] 145.9139
```

7. Using the function you wrote in part 6, calculate the PRESS statistic for your regression model with x_2, x_7, and x_8 as predictors. Calculate and compare ${R_{prediction}}^2$ and R^2 for this model. What comments can you make about the likely predictive performance of this model?

```r
library(TomLeversRPackage)
reduced_model <- lm(y ~ x2 + x7 + x8, data = data_set)
calculate_PRESS(reduced_model)
```

```
## [1] 87.46123
```

While 76.0 percent of variability in existing observations is explained by the reduced MLR model, only 51.4 percent of variability in new observations the model might be able to explain.