Assignment 5

```r
library(tidyverse)
library(leaps)
```

1.

a) forward selection

```r
# import data
cement <- read.csv("cement.csv")
attach(cement)

# correlation between xi and y.
cor(x1,y)
```

```
## [1] 0.7271548
```

```r
cor(x2,y)
```

```
## [1] 0.819802
```

```r
cor(x3,y)
```

```
## [1] -0.5342085
```

```r
cor(x4,y)
```

```
## [1] -0.8236563
```

```r
cor(x5,y)
```

```
## [1] -0.8358874
```

Because x5 has the largest correlation with y, we first add x5 into our model.

```r
mdl_x5 <- lm(y ~ x5)
summary(mdl_x5)
```

```
##
## Call:
## lm(formula = y ~ x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4828  -7.7636   0.6687   5.1438  16.6690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 117.2848     4.9533  23.678 8.68e-11 ***
## x5           -0.7404     0.1466  -5.051 0.000372 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.627 on 11 degrees of freedom
## Multiple R-squared:  0.6987, Adjusted R-squared:  0.6713
## F-statistic: 25.51 on 1 and 11 DF,  p-value: 0.0003717
```

Because p-value 0.00372 is less than $\alpha_{IN} = 0.10$, x5 is accepted.

```r
# partial correlation
x1_x5 <- lm(x1 ~ x5)
cor(mdl_x5$residuals, x1_x5$residuals)
```

```
## [1] 0.9519312
```

```r
x2_x5 <- lm(x2 ~ x5)
cor(mdl_x5$residuals, x2_x5$residuals)
```

```
## [1] 0.06404031
```

```r
x3_x5 <- lm(x3 ~ x5)
cor(mdl_x5$residuals, x3_x5$residuals)
```

```
## [1] -0.9024243
```

```r
x4_x5 <- lm(x2 ~ x5)
cor(mdl_x5$residuals, x4_x5$residuals)
```

```
## [1] 0.06404031
```

After we fit x5 in our model, x1 has the largest partial correlation, so we next add x1 into our model.

```r
mdl_x5x1 <- lm(y ~ x5 + x1)
summary(mdl_x5x1)
```

```
##
## Call:
## lm(formula = y ~ x5 + x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3855 -1.4921 -0.0183  1.6585  3.3066
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.13118    2.14626  48.051 3.68e-13 ***
## x5           -0.61186    0.04888 -12.518 1.96e-07 ***
## x1            1.38717    0.14115   9.827 1.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.771 on 10 degrees of freedom
## Multiple R-squared:  0.9717, Adjusted R-squared:  0.9661
## F-statistic: 171.9 on 2 and 10 DF,  p-value: 1.805e-08
```

Because p-value 1.86e-06 is less than $\alpha_{IN} = 0.10$, x1 is accepted.

```r
# partial correlation
x2_x5x1 <- lm(x2 ~ x5 + x1)
cor(mdl_x5x1$residuals, x2_x5x1$residuals)
```

```
## [1] 0.6329305
```

```r
x3_x5x1 <- lm(x3 ~ x5 + x1)
cor(mdl_x5x1$residuals, x3_x5x1$residuals)
```

```
## [1] -0.6058938
```

```r
x4_x5x1 <- lm(x4 ~ x5 + x1)
cor(mdl_x5x1$residuals, x4_x5x1$residuals)
```

```
## [1] -0.05926562
```

After we fit x5,x1 in our model, x2 has the largest partial correlation, so we next add x2 into our model.

```r
mdl_x5x1x2 <- lm(y ~ x5 + x1 + x2)
summary(mdl_x5x1x2)
```

```
##
## Call:
## lm(formula = y ~ x5 + x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3553 -1.5636  0.2582  1.3962  3.6144
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.5927    13.3823   5.275  0.00051 ***
## x5           -0.2244     0.1629  -1.377  0.20165
## x1            1.4255     0.1162  12.263  6.4e-07 ***
## x2            0.4317     0.1760   2.453  0.03660 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.262 on 9 degrees of freedom
## Multiple R-squared:  0.9831, Adjusted R-squared:  0.9774
## F-statistic:   174 on 3 and 9 DF,  p-value: 2.757e-08
```

Because p-value 0.03660 is less than $\alpha_{IN} = 0.10$, x2 is accepted.

```r
# partial correlation
x3_x5x1x2 <- lm(x3 ~ x5 + x1 +x2)
cor(mdl_x5x1x2$residuals, x3_x5x1x2$residuals)
```

```
## [1] -0.03476725
```

```r
x4_x5x1x2 <- lm(x4 ~ x5 + x1 +x2)
cor(mdl_x5x1x2$residuals, x4_x5x1x2$residuals)
```

```
## [1] 0.1843683
```

After we fit x5,x1,x2 in our model, x4 has the largest partial correlation, so we next add x4 into our model.

```r
mdl_x5x1x2x4 <- lm(y ~ x5 + x1 + x2 + x4)
summary(mdl_x5x1x2x4)
```

```
##
## Call:
## lm(formula = y ~ x5 + x1 + x2 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2452 -1.0579  0.4423  1.0171  3.3025
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68.0831    14.7308   4.622  0.00171 **
## x5           -0.9115     1.3061  -0.698  0.50502
## x1            1.3797     0.1488   9.272 1.49e-05 ***
## x2            0.4633     0.1929   2.402  0.04307 *
## x4            0.7220     1.3608   0.531  0.61013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.358 on 8 degrees of freedom
## Multiple R-squared:  0.9836, Adjusted R-squared:  0.9754
## F-statistic: 120.2 on 4 and 8 DF,  p-value: 3.542e-07
```

Because p-value 0.61013 is greater than $\alpha_{IN} = 0.10$, we do not add x4 into our model and stop variable selection.

```r
# final model
mdl_x5x1x2
```

```
##
## Call:
## lm(formula = y ~ x5 + x1 + x2)
##
## Coefficients:
## (Intercept)           x5           x1           x2
##     70.5927      -0.2244       1.4255       0.4317
```

The final model is: $y = 70.5927 - 0.224x_5 + 1.4255x_1 + 0.4317x_2$.

b). backward elimination

```r
# full model
full_mdl <- lm(y ~ x1 + x2 + x3 + x4 +x5)
summary(full_mdl)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4135 -0.8228  0.4621  1.1177  3.3117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.1523    75.3743   0.639   0.5433
## x1            1.5839     0.7718   2.052   0.0793 .
## x2            0.6666     0.7796   0.855   0.4208
## x3            0.2172     0.8034   0.270   0.7947
## x4            1.0276     1.8364   0.560   0.5932
## x5           -1.0165     1.4423  -0.705   0.5037
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.508 on 7 degrees of freedom
## Multiple R-squared:  0.9838, Adjusted R-squared:  0.9722
## F-statistic: 85.02 on 5 and 7 DF,  p-value: 4.12e-06
```

After fit the full model, x3 has the largest p-value 0.7947 and is large than $\alpha_{OUT} = 0.10$, so we remove x3 from our model.

```r
# remove x3
mdl_remove_x3 <- lm(y ~ x1 + x2 + x4 + x5)
summary(mdl_remove_x3)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 + x2 + x4 + x5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2452 -1.0579  0.4423  1.0171  3.3025
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68.0831    14.7308   4.622  0.00171 **
## x1            1.3797     0.1488   9.272 1.49e-05 ***
## x2            0.4633     0.1929   2.402  0.04307 *
## x4            0.7220     1.3608   0.531  0.61013
## x5           -0.9115     1.3061  -0.698  0.50502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.358 on 8 degrees of freedom
## Multiple R-squared:  0.9836, Adjusted R-squared:  0.9754
## F-statistic: 120.2 on 4 and 8 DF,  p-value: 3.542e-07
```

After remove x3 from our model, x4 has the largest p-value 0.61013 and is large than $\alpha_{OUT} = 0.10$, so we remove x4 from our model.

```r
# remove x3 and x4
mdl_remove_x3x4 <- lm(y ~ x1 + x2 + x5)
summary(mdl_remove_x3x4)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3553 -1.5636  0.2582  1.3962  3.6144
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  70.5927    13.3823   5.275  0.00051 ***
## x1            1.4255     0.1162  12.263  6.4e-07 ***
## x2            0.4317     0.1760   2.453  0.03660 *
## x5           -0.2244     0.1629  -1.377  0.20165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.262 on 9 degrees of freedom
## Multiple R-squared:  0.9831, Adjusted R-squared:  0.9774
## F-statistic:   174 on 3 and 9 DF,  p-value: 2.757e-08
```

After remove x3 and x4 from our model, x5 has the largest p-value 0.20165 and is large than $\alpha_{OUT} = 0.10$, so we remove x5 from our model.

```r
# remove x3,x4, and x5
mdl_remove_x3x4x5 <- lm(y ~ x1 + x2)
summary(mdl_remove_x3x4x5)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.419 -1.499 -1.446  1.282  3.865
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.39829    2.24310   23.36 4.68e-10 ***
## x1           1.45682    0.11902   12.24 2.42e-07 ***
## x2           0.66685    0.04499   14.82 3.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.361 on 10 degrees of freedom
## Multiple R-squared:  0.9795, Adjusted R-squared:  0.9754
## F-statistic: 238.7 on 2 and 10 DF,  p-value: 3.635e-09
```

After remove x3,x4 and x5 from our model, x1 has the largest p-value 2.42e-07 and is less than $\alpha_{OUT} = 0.10$, so we keep x1 in our model, and stop backward elimination.

```r
# final model
mdl_remove_x3x4x5
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Coefficients:
## (Intercept)           x1           x2
##     52.3983       1.4568       0.6668
```

The final model is: $y = 52.3983 + 1.4568x_1 + 0.6668x_2$.

c).

```r
cor(x4,x5)
```

```
## [1] 0.9992464
```

Because x4 and x5 are extremely correlated, when both x4 and x5 were considered together in the model, it will have multicollinearity problem.

d). stepwise regression

```r
# According to part a, we add x5 first into our model.
mdl_add_x5 <- lm(y ~ x5)
summary(mdl_add_x5)
```

```
##
## Call:
## lm(formula = y ~ x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4828  -7.7636   0.6687   5.1438  16.6690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 117.2848     4.9533  23.678 8.68e-11 ***
## x5           -0.7404     0.1466  -5.051 0.000372 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.627 on 11 degrees of freedom
## Multiple R-squared:  0.6987, Adjusted R-squared:  0.6713
## F-statistic: 25.51 on 1 and 11 DF,  p-value: 0.0003717
```

Because x5's p-value 0.000372 is less than $\alpha_{IN} = 0.10$, we add x5 into our model.

```r
# According to part a, we add x1 next.
mdl_add_x5x1 <- lm(y ~ x5 + x1)
summary(mdl_add_x5x1)
```

```
##
## Call:
## lm(formula = y ~ x5 + x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3855 -1.4921 -0.0183  1.6585  3.3066
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.13118    2.14626  48.051 3.68e-13 ***
## x5           -0.61186    0.04888 -12.518 1.96e-07 ***
## x1            1.38717    0.14115   9.827 1.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.771 on 10 degrees of freedom
## Multiple R-squared:  0.9717, Adjusted R-squared:  0.9661
## F-statistic: 171.9 on 2 and 10 DF,  p-value: 1.805e-08
```

Because x1's p-value 1.86e-06 is less than $\alpha_{IN} = 0.10$, we add x1 into our model.

Because x5's p-value 1.96e-07 is less than $\alpha_{OUT} = 0.10$, we keep x5 in our model.

```r
# According to part a, we add x2 next.
mdl_add_x5x1x2 <- lm(y ~ x5 + x1 + x2)
summary(mdl_add_x5x1x2)
```

```
##
## Call:
## lm(formula = y ~ x5 + x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3553 -1.5636  0.2582  1.3962  3.6144
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.5927    13.3823   5.275  0.00051 ***
## x5           -0.2244     0.1629  -1.377  0.20165
## x1            1.4255     0.1162  12.263  6.4e-07 ***
## x2            0.4317     0.1760   2.453  0.03660 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.262 on 9 degrees of freedom
## Multiple R-squared:  0.9831, Adjusted R-squared:  0.9774
## F-statistic:   174 on 3 and 9 DF,  p-value: 2.757e-08
```

Because x2's p-value 0.03660 is less than $\alpha_{IN} = 0.10$, we add x2 into our model.

Because x1's p-value 6.4e-07 is less than $\alpha_{OUT} = 0.10$, we keep x1 in our model.

Because x5's p-value 0.20165 is larger than $\alpha_{OUT} = 0.10$, we remove x5 from our model.

```r
# According to part a, we add x4 next.
mdl_add_x1x2x4 <- lm(y ~ x1 + x2 + x4)
summary(mdl_add_x1x2x4)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3216 -1.7190  0.0852  1.3843  3.7265
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.1064    14.0251   4.999  0.00074 ***
## x1            1.4416     0.1160  12.425 5.72e-07 ***
## x2            0.4383     0.1841   2.381  0.04115 *
## x4           -0.2196     0.1719  -1.278  0.23320
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.29 on 9 degrees of freedom
## Multiple R-squared:  0.9826, Adjusted R-squared:  0.9768
## F-statistic: 169.8 on 3 and 9 DF,  p-value: 3.078e-08
```

Because x4's p-value 0.23320 is greater than $\alpha_{IN} = 0.10$, we do not add x4 into our model.

Because both x1 and x2 p-value is less than $\alpha_{OUT} = 0.10$, we keep x1 and x2 in our model, and stop stepwise regression.

```r
# final model
mdl_add_x1x2 <- lm(y ~ x1 + x2)
mdl_add_x1x2
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Coefficients:
## (Intercept)           x1           x2
##     52.3983       1.4568       0.6668
```

The final model is: $y = 1.4568x_1 + 0.6668x_2$.

d). AIC

The AIC is

$$AIC = nln\frac{SS_{Res}}{n} + 2p$$

```
# AIC for model in a
# n = 13
# p = 3 + 1 = 4
(AIC_a <- 13 * log(sum((mdl_x5x1x2$residuals)^2)/13) + 2 * 4)
```

```
## [1] 24.43856
```

AIC for model in a is 24.43856.

```
# Since we get the same model in b and d, AIC for them are the same.
# n = 13
# p = 2 + 1 = 3
(AIC_b <- 13 * log(sum((mdl_remove_x3x4x5$residuals)^2)/13) + 2 * 3)
```

```
## [1] 24.92546
```

```
(AIC_d <- 13 * log(sum((mdl_add_x1x2$residuals)^2)/13) + 2 * 3)
```

```
## [1] 24.92546
```

AIC for model in b or d is 24.92546.

2.

a).

$H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

All the βs are in the model, it's full model.

$F_0 = \frac{SS_R/k}{SS_{Res}/(n-k-1)} = \frac{MS_R}{MS_{Res}}$ follows the $F_{k,n-k-1}$ distributions where k = 5, n is the number of observations.

b).

$H_0 : \beta_4 = \beta_5 = 0$, given that the first 3 variables are in the model.

partial β to $\beta_a(\beta_0, \beta_1, \beta_2, \beta_3)$ is (p - r) $\times$ 1, $\beta_b(\beta_4, \beta_5)$ is (r $\times$ 1), in this case p = 6, r = 2.

$F_0 = \frac{SS_R(\beta_4,\beta_5|\beta_0,\beta_1,\beta_2,\beta_3)/2}{MS_{Res}}$ follows the $F_{r,n-p}$ distribution where r = 2, p = 6, and n is the number of observations.

3.

```r
# import data
reactor <- read.csv("reactor.csv")
attach(reactor)
```

```
## The following objects are masked from cement:
##
##     X, x1, x2, x3, x4, x5, y
```

```r
reactor <- select(reactor, y:x7)

# p = 2
mdl_p2x1 <- lm(y ~ x1)
sum_p2x1 <- summary(mdl_p2x1)
mdl_p2x2 <- lm(y ~ x2)
sum_p2x2 <- summary(mdl_p2x2)
mdl_p2x3 <- lm(y ~ x3)
sum_p2x3 <- summary(mdl_p2x3)
mdl_p2x4 <- lm(y ~ x4)
sum_p2x4 <- summary(mdl_p2x4)
mdl_p2x5 <- lm(y ~ x5)
sum_p2x5 <- summary(mdl_p2x5)
mdl_p2x6 <- lm(y ~ x6)
sum_p2x6 <- summary(mdl_p2x6)
mdl_p2x7 <- lm(y ~ x7)
sum_p2x7 <- summary(mdl_p2x7)
```

For P = 2, we will choose 1 from 7 predictors, and the other 1 is intercept, So there are 7 combinations in total.
For P = 3, we will choose 2 from 7 predictors, and the other 1 is intercept, So there are 21 combinations in total.
For P = 4, we will choose 3 from 7 predictors, and the other 1 is intercept, So there are 35 combinations in total.
For P = 5, we will choose 4 from 7 predictors, and the other 1 is intercept, So there are 35 combinations in total.
I will manually fit the model get maximum R^2, minimum MS_{Res} and Mallow's C_p Statistic for P = 2, and use `regsubsets()` function in `leap` package to compelete the rest.

a). maximum R^2

```r
# for P = 2
(p2_r2_max <- max(sum_p2x1$r.squared, sum_p2x2$r.squared, sum_p2x3$r.squared, sum_p2x4$r.squared,
                  sum_p2x5$r.squared, sum_p2x6$r.squared, sum_p2x7$r.squared))
```

```
## [1] 0.4131767
```

```r
# regsubsets
p2 <- regsubsets(data = reactor, y ~ ., nbest = 7,  nvmax = 1, method = "exhaustive")
sum_p2 <- summary(p2)
(p2_max <- max(sum_p2$rsq))
```

```
## [1] 0.4131767
```

We get the same answer by fiting the model manually and using regsubsets() function.

```r
# p = 3
p3 <- regsubsets(data = reactor, y ~ ., nbest = 21, nvmax = 2, method = "exhaustive")
sum_p3 <- summary(p3)
(p3_max <- max(sum_p3$rsq))
```

```
## [1] 0.7467871
```

```r
# p = 4
p4 <- regsubsets(data = reactor, y ~ ., nbest = 35, nvmax = 3, method = "exhaustive")
sum_p4 <- summary(p4)
(p4_max <- max(sum_p4$rsq))
```

```
## [1] 0.9464945
```

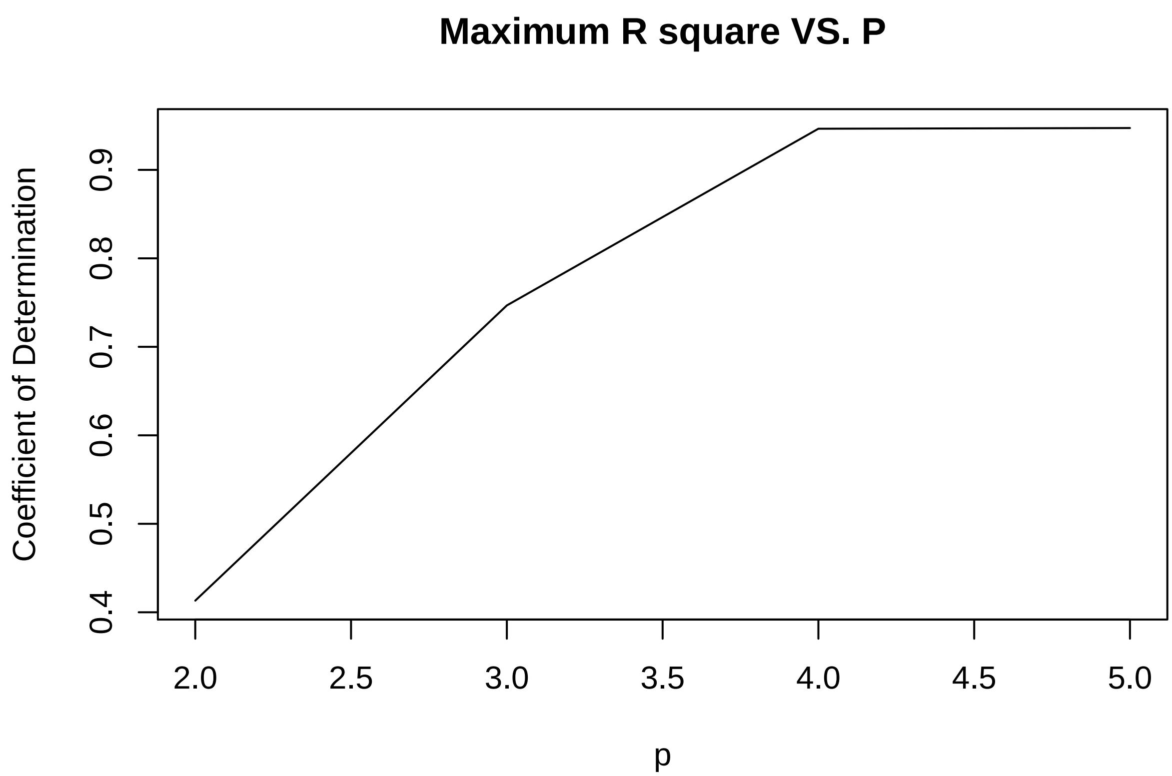
```r
# p = 5
p5 <- regsubsets(data = reactor, y ~ ., nbest = 35, nvmax = 4, method = "exhaustive")
sum_p5 <- summary(p5)
(p5_max <- max(sum_p5$rsq))
```

```
## [1] 0.9472694
```

```r
# maximum R square vector
(max_r2 <- c(p2_max, p3_max, p4_max, p5_max))
```

```
## [1] 0.4131767 0.7467871 0.9464945 0.9472694
```

```r
# plot
p <- c(2,3,4,5)
plot(p, max_r2, type = "l",
     ylab = "Coefficient of Determination",
     main = "Maximum R square VS. P")
```



I would choose the model at p = 4, after p = 4 even though R^2 is still increase, it's only a small increase.

```r
# final model
summary(p4,all.best = FALSE)
```

```
## Subset selection object
## Call: regsubsets.formula(data = reactor, y ~ ., nbest = 35, nvmax = 3,
##     method = "exhaustive")
## 7 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
## 35 subsets of each size up to 3
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4  x5  x6  x7
## 1  ( 1 ) " " " " " " " " "*" " " " "
## 2  ( 1 ) " " " " " " "*" "*" " " " "
## 3  ( 1 ) "*" " " "*" " " " " "*" " "
```

```r
lm(y~ x1 + x3 + x6)
```

```
##
## Call:
## lm(formula = y ~ x1 + x3 + x6)
##
## Coefficients:
## (Intercept)           x1           x3           x6
##  -1.778e-02   -2.982e-01    1.303e+00   -5.534e-06
```

Our final model is: $y = -0.01778 - 0.2982x_1 + 1.303x_3 - 0.000005536x_6$.

b). minimum MS_{Res}

```r
# p = 2
(p2_msr_min <- min(sum_p2x1$sigma^2, sum_p2x2$sigma^2, sum_p2x3$sigma^2, sum_p2x4$sigma^2,
                   sum_p2x5$sigma^2, sum_p2x6$sigma^2, sum_p2x7$sigma^2))
```

```
## [1] 3.910059e-07
```

```r
(p2_min <- min(sum_p2$rss/(28-2))) # divide n - p to get MSR from SSR
```

```
## [1] 3.910059e-07
```

```r
# p = 3
(p3_min <- min(sum_p3$rss/(28-3)))
```

```
## [1] 1.754669e-07
```

```r
# p = 4
(p4_min <- min(sum_p4$rss/(28-4)))
```

```
## [1] 3.862218e-08
```

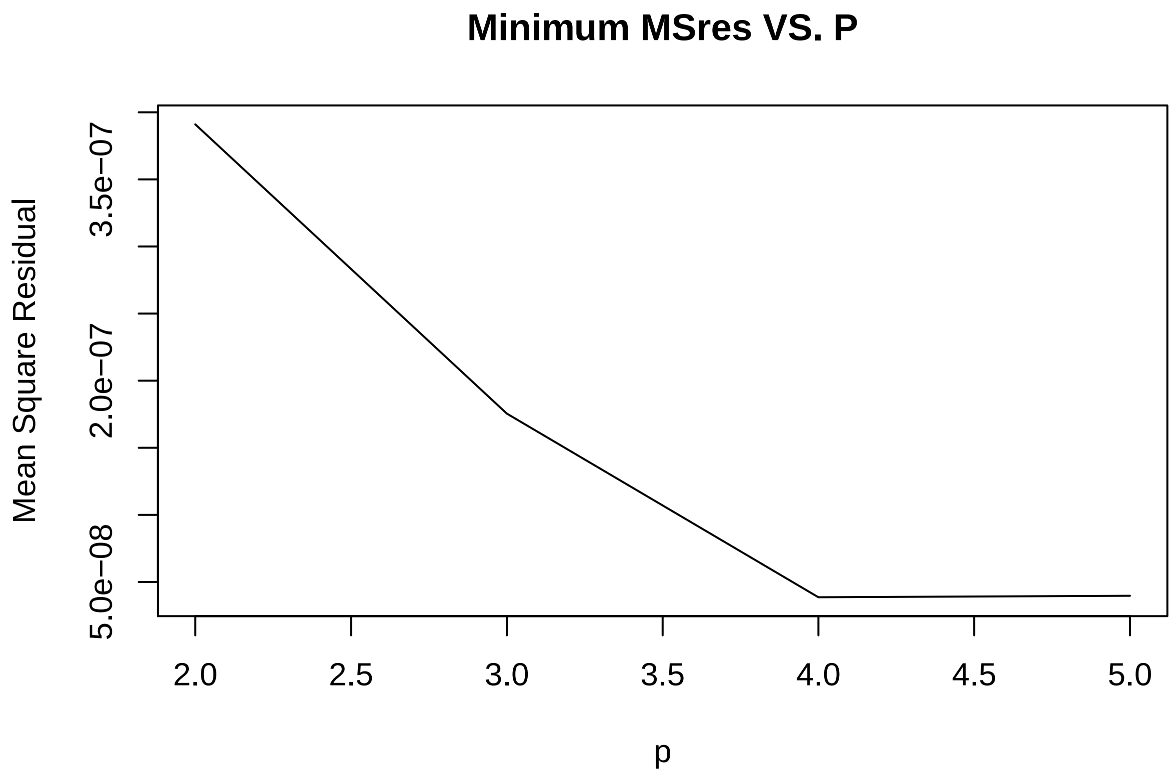
```r
# p = 5
(p5_min <- min(sum_p5$rss/(28-5)))
```

```
## [1] 3.971769e-08
```

```r
# minimum mean square residual vector
(min_msr <- c(p2_min, p3_min, p4_min, p5_min))
```

```
## [1] 3.910059e-07 1.754669e-07 3.862218e-08 3.971769e-08
```

```r
# plot
plot(p, min_msr, type = "l",
     ylab = "Mean Square Residual",
     main = "Minimum MSres VS. P")
```



I would choose the model at p = 4, because at p = 4 we have the minimum MS_{Res}.In other words, MS_{Res} starts increasing after p = 4.

Our final model is the same as part a : $y = -0.01778 - 0.2982x_1 + 1.303x_3 - 0.000005536x_6$.

c). Mallow's C_p Statistic

$$C_P = \frac{SS_{Res}(P)}{\hat{\sigma}^2} - n + 2p$$

```
# we estimate sigma using the MSres from the full model
fullfit <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7)
sum_full <- summary(fullfit)
# unbiased estimated sigma
sig_est <- sum_full$sigma^2
# p = 2
p2_cp_x1 <- sum(sum_p2x1$residuals^2)/sig_est - 28 + 2 * 2
p2_cp_x2 <- sum(sum_p2x2$residuals^2)/sig_est - 28 + 2 * 2
p2_cp_x3 <- sum(sum_p2x3$residuals^2)/sig_est - 28 + 2 * 2
```

```r
p2_cp_x4 <- sum(sum_p2x4$residuals^2)/sig_est - 28 + 2 * 2
p2_cp_x5 <- sum(sum_p2x5$residuals^2)/sig_est - 28 + 2 * 2
p2_cp_x6 <- sum(sum_p2x6$residuals^2)/sig_est - 28 + 2 * 2
p2_cp_x7 <- sum(sum_p2x7$residuals^2)/sig_est - 28 + 2 * 2
# p = 2
c(p2_cp_x1, p2_cp_x2, p2_cp_x3, p2_cp_x4, p2_cp_x5, p2_cp_x6, p2_cp_x7)
```

```
## [1] 273.9205 324.5012 277.0080 358.5307 201.4526 324.6202 359.7071
```

```r
# regsubsets()
(p2_cp <- sum_p2$cp[which.min(abs(sum_p2$cp-2))])
```

```
## [1] 201.4526
```

We choose the cp that is closest to p. when p = 2, we choose cp = 201.4526 when only x5 in our model. Using the same method for p = 3,. . . 5.

```r
# p = 3
(p3_cp <- sum_p3$cp[which.min(abs(sum_p3$cp-3))])
```

```
## [1] 75.28227
```

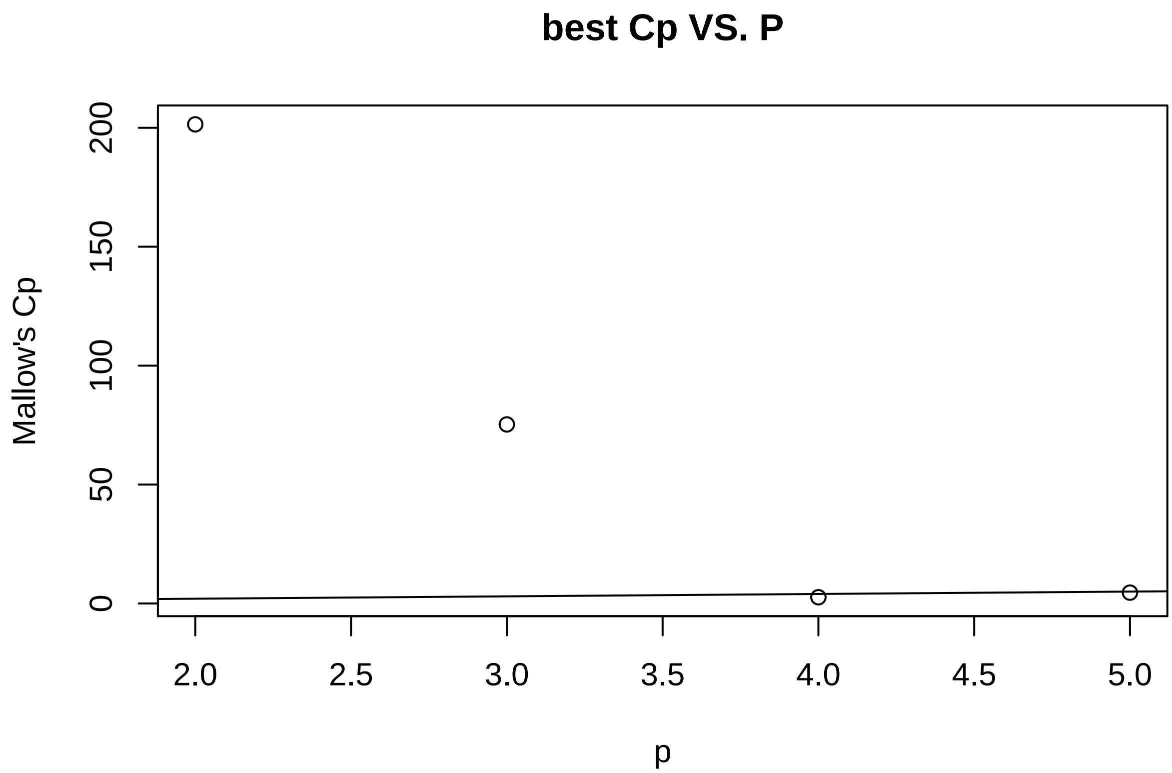
```r
# p = 4
(p4_cp <- sum_p4$cp[which.min(abs(sum_p4$cp-4))])
```

```
## [1] 2.618208
```

```r
# p = 5
(p5_cp <- sum_p5$cp[which.min(abs(sum_p5$cp-5))])
```

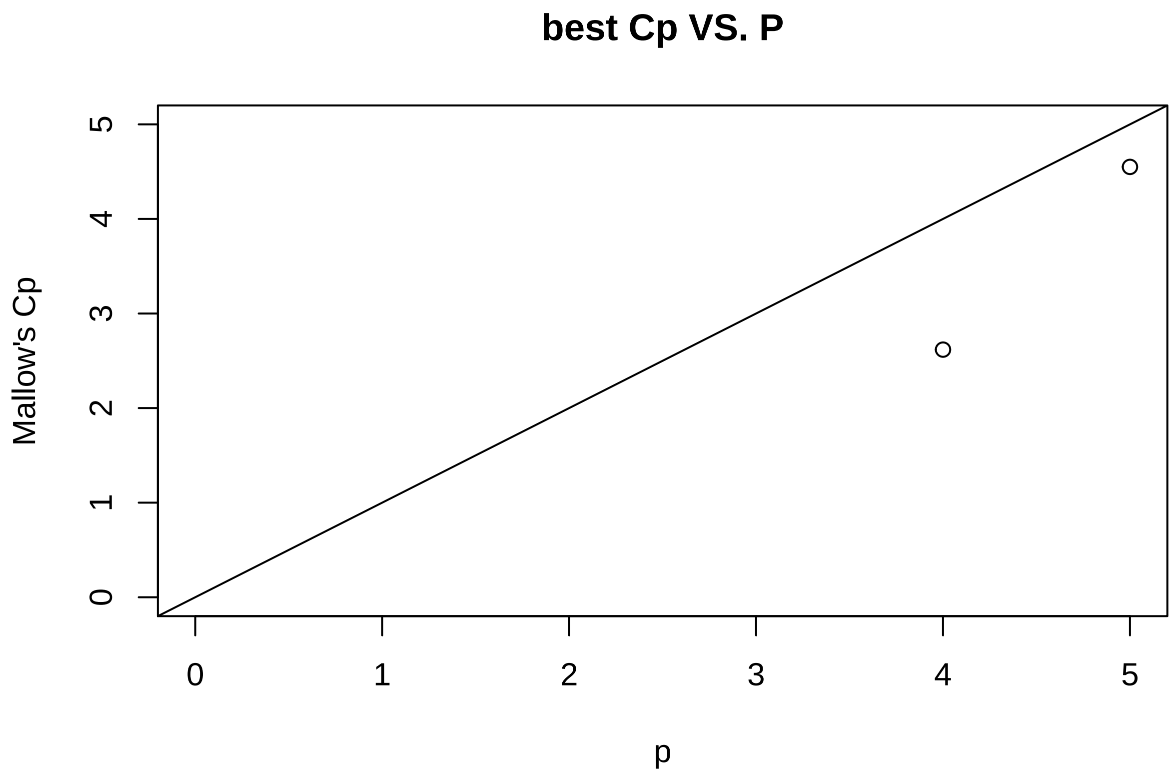
```
## [1] 4.550012
```

```r
# cp vector
cp <- c(p2_cp, p3_cp, p4_cp, p5_cp)
plot(p, cp, abline(0,1),
     ylab = "Mallow's Cp",
     main = "best Cp VS. P",)
```



Again, We will choose the cp that is closest to p. Replot to examine points when p = 4 and p = 5.

```
plot(p, cp, abline(0,1),
     xlim = c(0,5),
     ylim = c(0,5),
     ylab = "Mallow's Cp",
     main = "best Cp VS. P")
```



According to the plot, we can observe that cp is more closer to p when p = 5.

```
# final model
summary(p5, all.best = FALSE)
```

```
## Subset selection object
## Call: regsubsets.formula(data = reactor, y ~ ., nbest = 35, nvmax = 4,
##     method = "exhaustive")
## 7 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
## 35 subsets of each size up to 4
```

```
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4  x5  x6  x7
## 1  ( 1 ) " " " " " " " " "*" " " " "
## 2  ( 1 ) " " " " " " "*" "*" " " " "
## 3  ( 1 ) "*" " " "*" " " " " "*" " "
## 4  ( 1 ) "*" " " "*" " " " " "*" "*"
```

```r
lm(y~ x1 + x3 + x6 + x7)
```

```
##
## Call:
## lm(formula = y ~ x1 + x3 + x6 + x7)
##
## Coefficients:
## (Intercept)           x1           x3           x6           x7
##  -1.790e-02   -2.978e-01    1.308e+00   -5.579e-06    5.204e-03
```

Our final model is: $y = -0.0179 - 0.2978x_1 + 1.308x_3 - 0.000005579x_6 + 0.005204x_7$.