Homework#2

Arinjay Jain

```
library(class)
library(formatR)

data <- read.table(file = "C:/Arinjay_Personal/Statistical Learning/Homework#2/Grocery.txt",
    header = FALSE, sep = "\t")


dataFrame <- data.frame(data)
names(dataFrame) <- c("Y", "X1", "X2", "X3")

fitModel <- lm(Y ~ X1 + X2 + factor(X3), data = dataFrame)
summary(fitModel)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + factor(X3), data = dataFrame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.05 -110.73  -22.52   79.29  295.75
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.150e+03  1.956e+02  21.220  < 2e-16 ***
## X1           7.871e-04  3.646e-04   2.159   0.0359 *
## X2          -1.317e+01  2.309e+01  -0.570   0.5712
## factor(X3)1  6.236e+02  6.264e+01   9.954 2.94e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 143.3 on 48 degrees of freedom
## Multiple R-squared:  0.6883, Adjusted R-squared:  0.6689
## F-statistic: 35.34 on 3 and 48 DF,  p-value: 3.316e-12
```

```
coefficients <- fitModel$coefficients
std_Dev <- coef(summary(fitModel))[, "Std. Error"]
z_Score <- coef(summary(fitModel))[, "t value"]
p_Values <- coef(summary(fitModel))[, "Pr(>|t|)"]
fitModel_Table <- cbind(coefficients, std_Dev, z_Score, p_Values)
print(fitModel_Table)
```

```
##              coefficients      std_Dev    z_Score     p_Values
## (Intercept)  4.149887e+03 1.955654e+02 21.2199453 4.902653e-26
```

```
## X1           7.870804e-04 3.645540e-04  2.1590228 3.587650e-02
## X2          -1.316602e+01 2.309173e+01 -0.5701616 5.712274e-01
## factor(X3)1  6.235545e+02 6.264095e+01  9.9544230 2.940869e-13
```

```
estimation_SigmaSquare <- (sum((fitModel$residuals)^2))/fitModel$df.residual

cat("estimation sigma_SigmaSquare:", estimation_SigmaSquare)
```

```
## estimation sigma_SigmaSquare: 20531.87
```

```
y_Hat <- predict(fitModel)
```

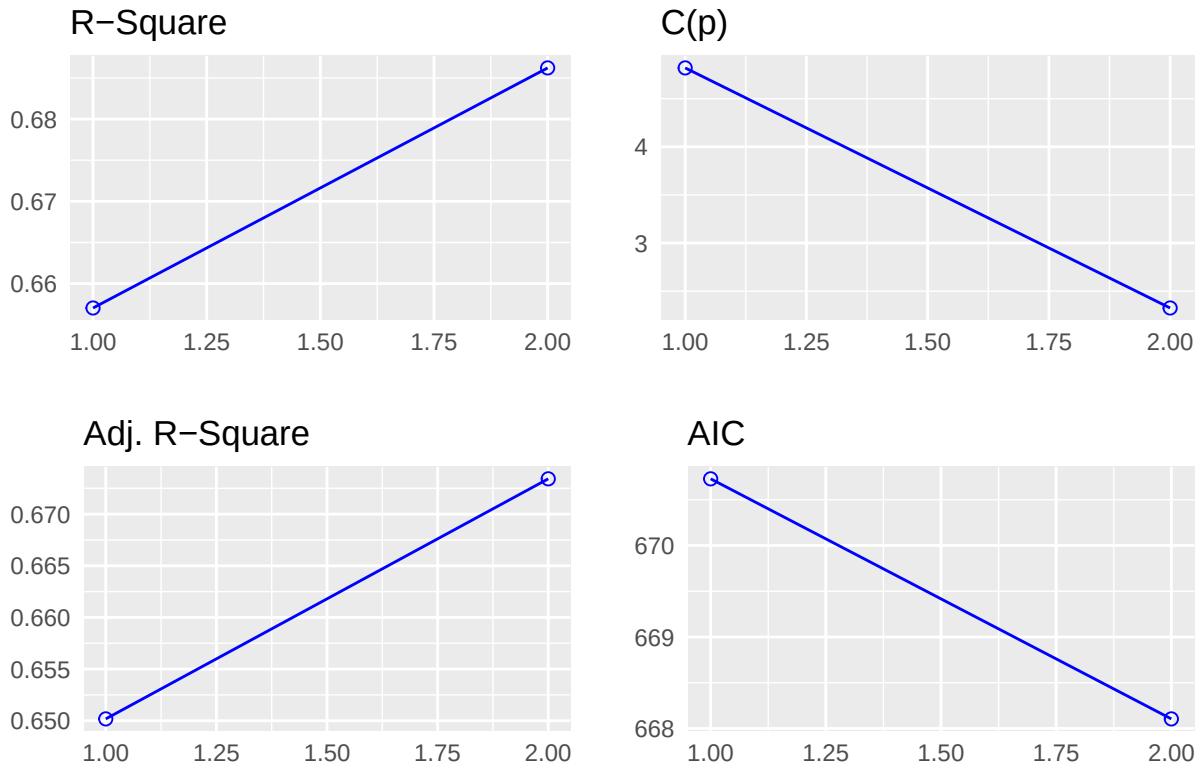
```
#Stepwise

library(olsrr)
```

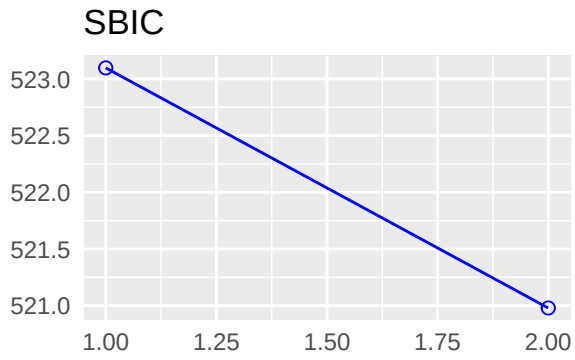
```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

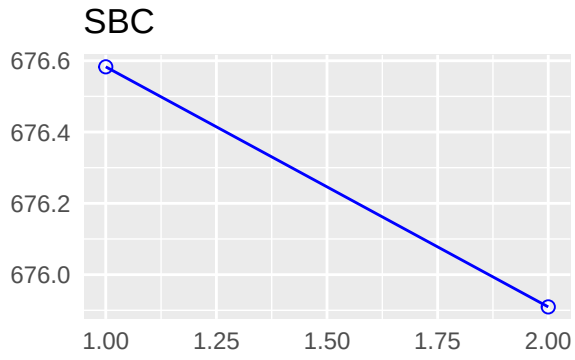
```
forward_Step<-ols_step_forward_p(fitModel)
print(forward_Step)
```

```
##
## Selection Summary
## ------------------------------------------------------------------------
## Variable Adj.
## Step Entered R-Square R-Square C(p) AIC RMSE
## ------------------------------------------------------------------------
## 1 factor(X3) 0.6570 0.6502 4.8198 670.7292 147.2745
## 2 X1 0.6862 0.6734 2.3251 668.1045 142.2992
## ------------------------------------------------------------------------
```

```
plot(ols_step_forward_p(fitModel))
```


R−Square
0.68
0.67
0.66
1.00
1.25
1.50
1.75
2.00
C(p)
4
3
1.00
1.25
1.50
1.75
2.00
Adj. R−Square
0.670
0.665
0.660
0.655
0.650
1.00
1.25
1.50
1.75
2.00
AIC
670
669
668
1.00
1.25
1.50
1.75
2.00






```
back_Step<-ols_step_backward_p(fitModel)
print(back_Step)
```

```
##
##
## Elimination Summary
## ---------------------------------------------------------------------------
## Variable Adj.
## Step Removed R-Square R-Square C(p) AIC RMSE
## ---------------------------------------------------------------------------
## 1 X2 0.6862 0.6734 2.3251 668.1045 142.2992
## ---------------------------------------------------------------------------
```

```
print("From Forward and Backward both approaches giving same results. In our final model, we will keep X
```

```
## [1] "From Forward and Backward both approaches giving same results. In our
final model, we will keep X1, X3 and remove X2"
```

```
finalModel<- lm(Y~X1+factor(X3), data=dataFrame)
summary(finalModel)
```

```
##
## Call:
## lm(formula = Y ~ X1 + factor(X3), data = dataFrame)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -286.249  -99.650   -9.251   70.746  292.311
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.058e+03  1.109e+02  36.592  < 2e-16 ***
## X1          7.704e-04  3.609e-04   2.135   0.0378 *
## factor(X3)1 6.196e+02  6.183e+01  10.021 1.88e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.3 on 49 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6734
## F-statistic: 53.58 on 2 and 49 DF,  p-value: 4.647e-13
```

```
estimation_SigmaSquare_finalModel<- (sum((finalModel$residuals)^2))/finalModel$df.residual

cat("estimation sigma_SigmaSquare_finalModel:", estimation_SigmaSquare_finalModel)
```

```
## estimation sigma_SigmaSquare_finalModel: 20249.07
```

```
## Bestsubset using Cp Criteria
library(leaps)

models <- regsubsets(Y~., data = dataFrame, nvmax = 3)

modelSummary <- summary(models)

CP = which.min(modelSummary$cp)

#best model will have below predictors:
modelSummary$which[CP,]
```

```
## (Intercept)          X1          X2          X3
##        TRUE        TRUE       FALSE        TRUE
```

```
print("Checking the p-values in both small model and full model for the F-test to see the significance l
```

```
## [1] "Checking the p-values in both small model and full model for the F-test
to see the significance level:"
```

```
#From part b: FinalModel #From part a: Fit model
com <- anova(finalModel,fitModel,test='F')

cat("F test value", com$F[2])
```

```
## F test value 0.3250843
```

```r
cat("P-value value", com$`Pr(>F)`[2])
```

```
## P-value value 0.5712274
```

```r
## Using F test formula

rSS_0 <- sum((finalModel$residuals)^2)
rSS_1 <- sum((fitModel$residuals)^2)

f_test = (rSS_0-rSS_1)*(fitModel$df.residual)/rSS_1
f_test
```

```
## [1] 0.3250843
```

```r
# F critical value
f_critical <- qf(p = 0.95, df1 = 1, df2 = 48)
f_critical
```

```
## [1] 4.042652
```

```r
if (f_test < f_critical){
  print("The null hypothesis is accepted")
}
```

```
## [1] "The null hypothesis is accepted"
```

```r
print("Here we can see in the small model (final model) both (x1 and x3) predictors have very significa
```

```
## [1] "Here we can see in the small model (final model) both (x1 and x3)
predictors have very significant (less then alpha{0.05}) p-value but in the
full model we have X2 with non-significan p-value. Hence, we will go with small
model(final model) as it keeps the model simpler with features being
statistically more significant "
```