HomeWork # 3 ARINJAY JAIN
A20447307

Problem # 1 Ex. 3.12 Show that ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set.

Given

$$X_{new} = \begin{pmatrix} X_{n \times p} \\ \sqrt{\lambda} I_{p \times p} \end{pmatrix} \qquad Y_{new} = \begin{pmatrix} Y_{n \times 1} \\ 0_{p \times 1} \end{pmatrix}_{n+p}$$

$X_{new} \Rightarrow$ row $= n+p$
column $= p$

$Y_{new} \Rightarrow$ row $= n+p$
column $= 1$.

$$\hat{\beta}^{OLS} = (X^T X)^{-1} \cdot X^T \cdot Y.$$

we have to show

$$\hat{\beta}^{OLS}_{(X_{new}, Y_{new})} = \hat{\beta}^{ridge}.$$

for example $\Rightarrow$ let take $n = 3$ and $p = 2$

$$X_{new} = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \\ \hline \sqrt{\lambda} I & & \sqrt{\lambda} I \\ \sqrt{\lambda} I & & \sqrt{\lambda} I \end{pmatrix} \rightarrow X \qquad Y_{new} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ 0 \\ 0 \end{pmatrix}$$

$$(X_{new}^T \cdot X_{new}) =$$

$$X_{new}^T = \begin{pmatrix} X_{11} & X_{21} & X_{31} & \sqrt{\lambda} I & \sqrt{\lambda} I \\ X_{12} & X_{22} & X_{32} & \sqrt{\lambda} I & \sqrt{\lambda} I \end{pmatrix}$$

$$\downarrow$$

$$X^T$$

$$X_{new}^T X_{new} = \begin{pmatrix} X_{11}^2 + X_{21}^2 + X_{31}^2 + 2\lambda I & X_{11}X_{12} + X_{21}X_{22} + X_{31}X_{32} + 2\lambda I \\ X_{12}X_{11} + X_{22}X_{21} + X_{32}X_{31} + 2\lambda I & X_{11}^2 + X_{21}^2 + X_{31}^2 + 2\lambda I \end{pmatrix}$$

In general

$$\left\{ X_{new}^T X_{new} = X^T X + 2\lambda I \right\}$$

$P \cdot \lambda = \text{constant}$. (λ)

$$\left[X_{new}^T X_{new} = X^T X + \lambda I \right]$$

$$X_{new}^T Y_{new} = \begin{pmatrix} X_{11}Y_1 + X_{21}Y_2 + X_{31}Y_3 + \sqrt{\lambda} I \cdot 0 + \sqrt{\lambda} I \cdot 0 \\ X_{12}Y_1 + X_{22}Y_2 + X_{32}Y_3 + \sqrt{\lambda} I \cdot 0 + \sqrt{\lambda} I \cdot 0 \end{pmatrix}$$

$$X_{new}^T \cdot Y_{new} = X^T \cdot Y \qquad 0$$

$$\hat{\beta}_{new}^{OLS} = (X_{new}^T \cdot X_{new}) X_{new}^T \cdot Y_{new}$$

$$\left\{ \hat{\beta}_{new}^{OLS} = (X^T \cdot X + \lambda I) X^T \cdot Y = \hat{\beta}^{ridge} \right\}$$

Problem #2 Ex 3.30

Consider the elastic-net optimization problem:

$$\min_{\beta} \|y - X\beta\|^2 + \lambda[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1]$$

Show how one can turn this into a lasso problem, using an augmented version of X and y.

Solution ⇒

Augmented version of X and y will be $\tilde{X}$ & $\tilde{y}$

$$\tilde{X} = \begin{bmatrix} X \\ \gamma I_p \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

$$\tilde{X}\beta = \begin{bmatrix} X\beta \\ \gamma\beta \end{bmatrix}$$

from given hint we know that

$$\|\tilde{y} - \tilde{X}\beta\|_2^2 = \left\| \begin{bmatrix} y - X\beta \\ \gamma\beta \end{bmatrix} \right\|_2^2 = \|y - X\beta\|_2^2 + \gamma^2\|\beta\|_2^2$$

elastin-net

$$\min_{\beta}\{\|y - X\beta\|^2 + \lambda[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1]\}$$

$$= \|y - X\beta\|_2^2 + \lambda\alpha\|\beta\|_2^2 + \lambda(1-\alpha)\|\beta\|_1$$

$\lambda\alpha$: constant → γ^2

$\lambda(1-\alpha)$: constant → λ

$$= \underbrace{\|Y - X\beta\|_2^2 + \gamma^2 \|\beta\|_2^2} + \tilde{\lambda} \|\beta\|_1$$

$$= \|\tilde{y} - \tilde{x}\beta\|_2^2 + \tilde{\lambda} \|\beta\|_1 \quad \#$$

This is a lasso objective function in the form of augment $\tilde{x}$ and $\tilde{y}$

Q4 (3.16) Derive the entries in table 3.4 the explicit forms for estimators in the orthogonal case.

Solution :→

Given table :→

Estimator	Formula.
Best subset (size M)	$\hat{\beta}_j \cdot I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_M\rvert)$
Ridge	$\hat{\beta}_j / (1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$

by the defination of orthonormal

$$X^T \cdot X = I$$

OLS

$$\hat{\beta} = (X^T X)^{-1} \cdot X^T \cdot y \Rightarrow (I)^{-1} X^T \cdot y$$

$$\hat{\beta} = X^T \cdot y$$

① Best subset ⇒ will take the M predictor with smallest residual sum of square. (RSS).

we know that columns of x are orthonormal we can construct a basis of euclidean space R^n equiped with the standard inner product. This will be happen by using the first 'p' colums of x and the extending there to 'N-p' linearly independent additional orthonormal vectors.

$$y = \sum_{j=1}^{p} \hat{\beta}_j x_j + \sum_{j=p+1}^{n} \gamma_j z_j$$

where $\hat{\beta}_j$ = Component of $\hat{\beta}$ in eq

γ_j = coefficients of 'j' w.r.t extended basis vector.

Best Subset Selection method estimatge of y can be written as

$$\hat{y} = \sum_{j=1}^{p} I_j \hat{\beta}_j x_j$$

{ where $I_j = 1$ if the predictor x_j are kept or zero otherwise.

As x_j & $\tilde{x}_j$ are orthonormed

So,

$$||Y - \hat{y}||_2^2 = ||y - X\hat{\beta}||_2^2$$

$$||y - \hat{y}||_2^2 = || \underbrace{\sum_{j=1}^{p} \hat{\beta}_j x_j + \sum_{j=p+1}^{N} \gamma_j \tilde{x}_j}_{\Downarrow\ y} - \underbrace{\sum_{j=1}^{p} I_j \hat{\beta}_j x_j}_{\Downarrow\ \hat{y}} ||$$

$$||y - \hat{y}||_2^2 = || \sum \hat{\beta}_j x_j (1 - I_j) x_j + \sum \gamma_j \tilde{x}_j ||_2^2$$

$$= \sum_{j=1}^{p} \hat{\beta}_j^2 (1 - I_j)^2 ||x_j||_2^2 + \sum_{j=p+1}^{N} \gamma_j^2 ||\tilde{x}_j||_2^2$$

$$||y - \hat{y}||_2^2 = \sum_{j=1}^{p} \hat{\beta}_j^2 (1 - I_j)^2 + \sum_{j=1}^{N} \gamma_j^2$$

We can minimize $||y - \hat{y}||_2^2$ we will choose M values of I_j that are equal to one which have the largest values of $\hat{\beta}_j^2$

Indicator function build a reation between A and x.

$\mathbb{1}$ = all the elements of x in A

$\mathbb{0}$ = all the elements of x not in A.

By the definition of Indicator function we can sort the values of $|\beta_j|$ and get only those values with the indices of largest M meaning where $I_j = 1$ and remaining indices with $I_j = 0$ are taken out.

by using Indicator function

$$\hat{\beta}_j^{\text{best subset (size } M)} = \vec{\beta}_j \times I(\text{rank}(|\hat{\beta}_j| \leq M))$$

$$\left\{ \hat{\beta}_j^{bs} = \vec{\beta}_j^{ls} = x^T.y. \right\}$$

for Ridge Regression :-

we know that.

$$\hat{\beta}^{ridge} = (x^T \cdot x + \lambda I)^{-1} x^T.y. \quad \rightarrow \text{orthonormal.}$$

$$= (I + \lambda I)^{-1} x^T.y$$

$$= \frac{x^T.y}{I(1+\lambda)} = \frac{x^T.y}{1+\lambda} \rightarrow \hat{\beta}^{ls}.$$

$$\boxed{\hat{\beta}^{ridge} = \frac{\hat{\beta}^{ls}}{1+\lambda}}$$

for Lasso :

we know that

$$L(\beta) = (y - x\beta)^T (y - x\beta) + \lambda |\beta|$$

first order derivative. w.r.t β.

$$\frac{\partial f(\beta)}{\partial \beta} = -x^T y + x^T x \beta + \lambda \cdot \text{Sign}(\beta)$$

for max(β) will $\frac{\partial L(\beta)}{\partial \beta} = 0$

$$-x^T y + x^T x \hat{\beta} + \lambda \text{Sign}(\beta) = 0$$

$$x^T x \hat{\beta} = x^T y - \lambda \text{Sign}(\beta)$$

$x^T x$ ↓ orthonormal

$$I \hat{\beta} = x^T y - \lambda \text{Sign}(\beta)$$

$$\hat{\beta} = I^{-1} (x^T y - \lambda \text{Sign}(\beta)) \quad \therefore I^{-1} = I$$

$$= I (x^T y - \lambda \text{Sign}(\beta))$$

$$\hat{\beta}^{\text{lasso}} = \text{Sign}(\beta) (|x^T y| - \lambda)$$

$$\left\{ \hat{\beta}^{\text{lasso}} = \text{Sign}(\beta) (|x^T y| - \lambda)_+ \right\}$$

Homework-3 Problem-3

Arinjay Jain

```
library(MASS)
boston_df <- Boston
```

Part a:

```
## Changing chas variable into factor.
boston_df$chas <- as.factor(boston_df$chas)
attach(boston_df)

#creating vectoe for all p values
p_values <- c()

# model with "zn"
model_zn <- lm(crim ~ zn)
smry_zn <- summary(model_zn)
print(smry_zn)
```

```
##
## Call:
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
pvalue_zn <- smry_zn$coefficients[2,4]
p_values <- append(p_values, pvalue_zn)
```

```r
# model with "indus"
model_indus <- lm(crim ~ indus)
smry_indus <- summary(model_indus)
print(smry_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
pvalue_indus <- smry_indus$coefficients[2,4]
p_values <- append(p_values, pvalue_indus)


# model with "chas"
model_chas <- lm(crim ~ chas)
smry_chas <- summary(model_chas)
print(smry_chas)
```

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas1        -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```r
pvalue_chas <- smry_chas$coefficients[2,4]
p_values <- append(p_values, pvalue_chas)


# model with "nox"
model_nox <- lm(crim ~ nox)
smry_nox <- summary(model_nox)
print(smry_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
pvalue_nox <- smry_nox$coefficients[2,4]
p_values <- append(p_values, pvalue_nox)

# model with "rm"
model_rm <- lm(crim ~ rm)
smry_rm <- summary(model_rm)
print(smry_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
pvalue_rm <- smry_rm$coefficients[2,4]
p_values <- append(p_values, pvalue_rm)

# model with "age"
model_age <- lm(crim ~ age)
smry_age <- summary(model_age)
print(smry_age)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
pvalue_age <- smry_age$coefficients[2,4]
p_values <- append(p_values, pvalue_age)

# model with "dis"
model_dis <- lm(crim ~ dis)
smry_dis <- summary(model_dis)
print(smry_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_dis <- smry_dis$coefficients[2,4]
p_values <- append(p_values, pvalue_dis)

# model with "rad"
model_rad <- lm(crim ~ rad)
smry_rad <- summary(model_rad)
print(smry_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_rad <- smry_rad$coefficients[2,4]
p_values <- append(p_values, pvalue_rad)

# model with "tax"
model_tax <- lm(crim ~ tax)
smry_tax <- summary(model_tax)
print(smry_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_tax <- smry_tax$coefficients[2,4]
p_values <- append(p_values, pvalue_tax)

# model with "ptratio"
model_ptratio <- lm(crim ~ ptratio)
smry_ptratio <- summary(model_ptratio)
print(smry_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
pvalue_ptratio <- smry_ptratio$coefficients[2,4]
p_values <- append(p_values, pvalue_ptratio)

# model with "black"
model_black <- lm(crim ~ black)
smry_black <- summary(model_black)
print(smry_black)
```

```
##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_black <- smry_black$coefficients[2,4]
p_values <- append(p_values, pvalue_black)


# model with "lstat"
model_lstat <- lm(crim ~ lstat)
smry_lstat <- summary(model_lstat)
print(smry_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

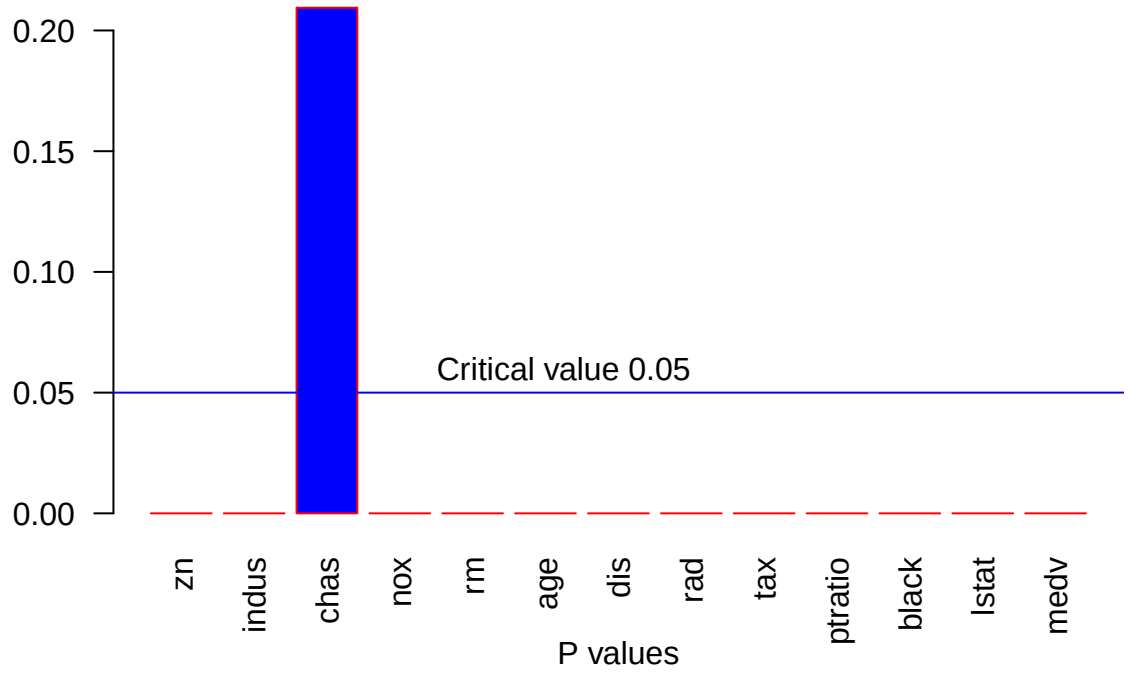
```
pvalue_lstat <- smry_lstat$coefficients[2,4]
p_values <- append(p_values, pvalue_lstat)

# model with "medv"
model_medv <- lm(crim ~ medv)
smry_medv <- summary(model_medv)
print(smry_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
```

```
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_medv <- smry_medv$coefficients[2,4]
p_values <- append(p_values, pvalue_medv)
```

```
# Plot the bar chart on P values
barplot(p_values,las=2,names.arg=colnames(boston_df)[-1],xlab="P values",col="blue",border="red")
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.06, "Critical value 0.05")
```




```
print("Here we can clearly see the P-value of chas is greater than '0.05' that means chas is not a signi
```

```
## [1] "Here we can clearly see the P-value of chas is greater than '0.05' that means chas is not a
significant predictor for our model and we can include all other varibles in our future model."
```

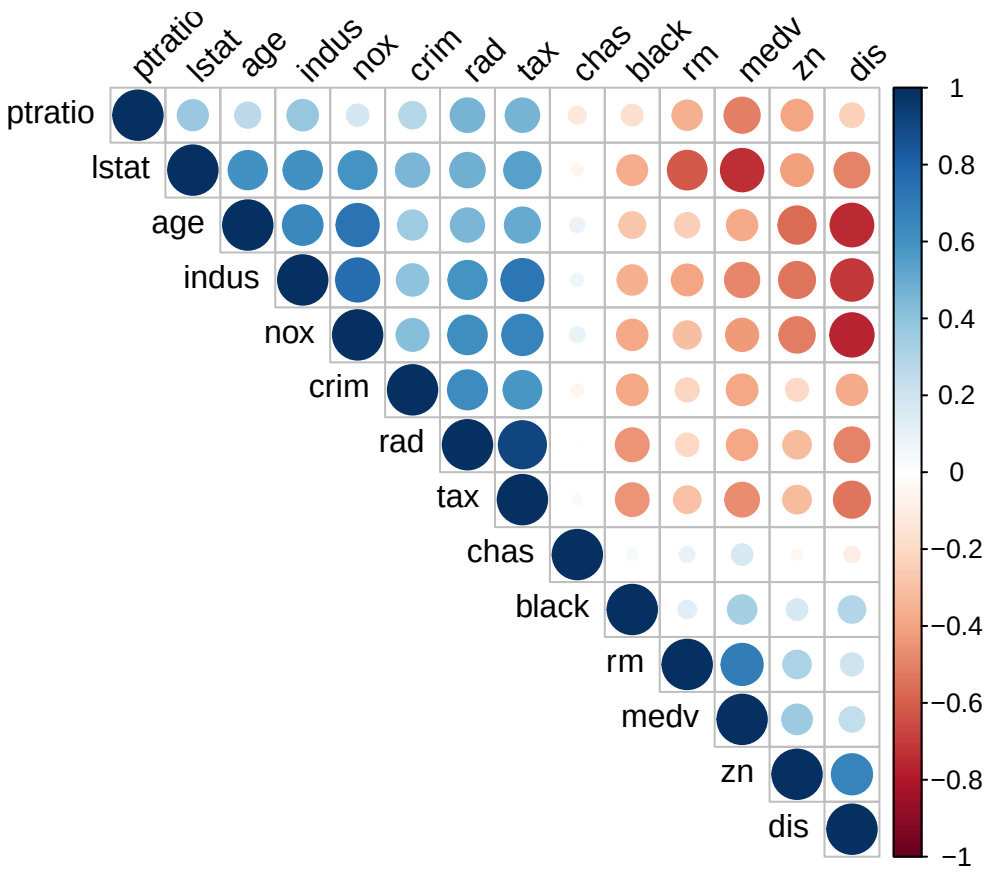
```
## Correlation matrix

res <- cor(Boston)
print(res)
```

```
##                crim          zn       indus         chas         nox
## crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##                  rm         age         dis          rad         tax    ptratio
## crim    -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.2899456
## zn       0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.3916785
## indus   -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.3832476
## chas     0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.1215152
## nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
## rm       1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
## age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
## rad     -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
## tax     -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
## ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
## black    0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801 -0.1773833
## lstat   -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
##                black      lstat       medv
## crim    -0.38506394  0.4556215 -0.3883046
## zn       0.17552032 -0.4129946  0.3604453
## indus   -0.35697654  0.6037997 -0.4837252
## chas     0.04878848 -0.0539293  0.1752602
## nox     -0.38005064  0.5908789 -0.4273208
## rm       0.12806864 -0.6138083  0.6953599
## age     -0.27353398  0.6023385 -0.3769546
## dis      0.29151167 -0.4969958  0.2499287
## rad     -0.44441282  0.4886763 -0.3816262
## tax     -0.44180801  0.5439934 -0.4685359
## ptratio -0.17738330  0.3740443 -0.5077867
## black    1.00000000 -0.3660869  0.3334608
## lstat   -0.36608690  1.0000000 -0.7376627
## medv     0.33346082 -0.7376627  1.0000000
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(res, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45, sig.level = 0.05)
```



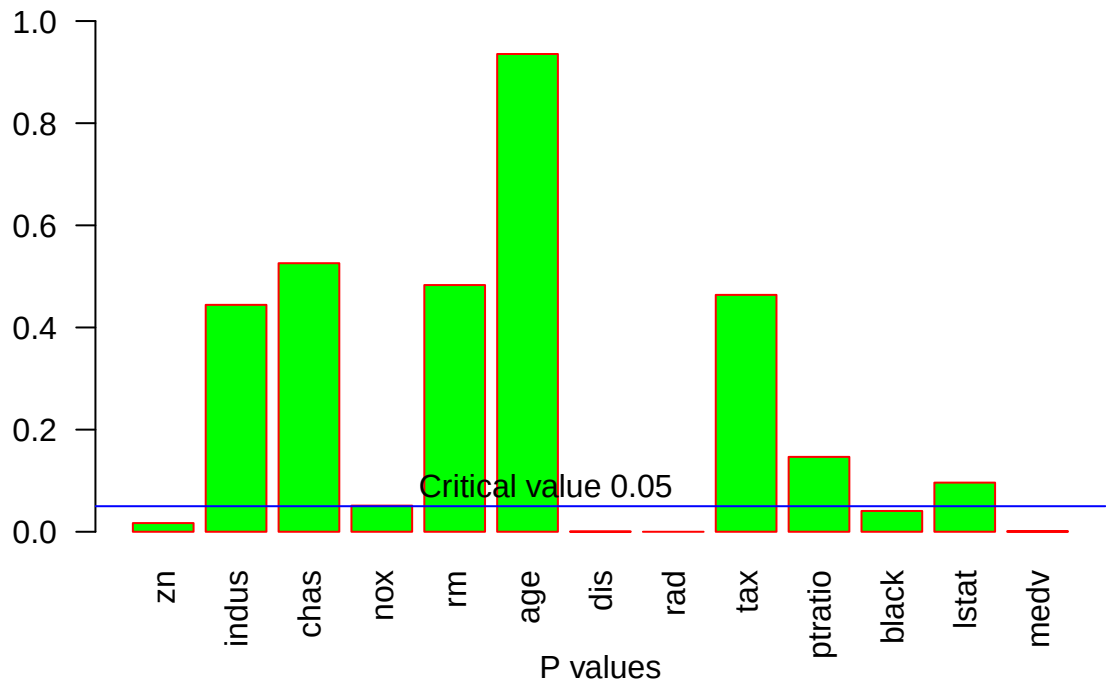
Part b:

```
full_model <- lm(crim ~., data = boston_df)
smry_full<- summary(full_model)
print(smry_full)
```

```
##
## Call:
## lm(formula = crim ~ ., data = boston_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas1        -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
```

```
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
p_values_full_model <- smry_full$coefficients[2:14,4]

# Plot the bar chart on P values
barplot(p_values_full_model,las=2,names.arg=colnames(boston_df)[-1],xlab="P values",ylim = c(0,1),col="
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```

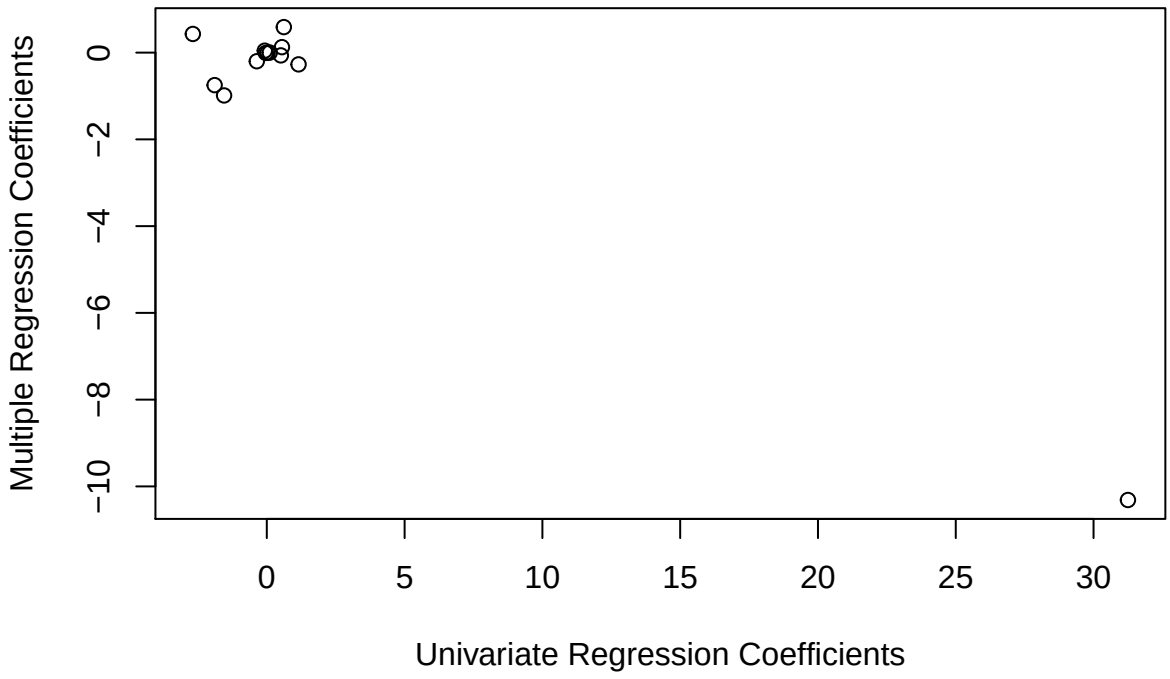



```
print("From summary of full model we can reject the null hypothesis for 'zn', 'dis', 'rad', 'black' and
```

```
## [1] "From summary of full model we can reject the null hypothesis for 'zn', 'dis', 'rad',
'black' and 'medv' predictors."
```

Part c:

```
# getting all coff into single vector
simple_reg_coff <- c() #empty vector

simple_reg_coff <- c(model_zn$coefficients[2],model_indus$coefficients[2],model_chas$coefficients[2],mod
                     model_dis$coefficients[2],
                     model_rad$coefficients[2],
                     model_tax$coefficients[2],
                     model_ptratio$coefficients[2],
                     model_black$coefficients[2],
                     model_lstat$coefficients[2],
                     model_medv$coefficients[2])

multiple_reg_coff <- full_model$coefficients[-1]

plot(simple_reg_coff,multiple_reg_coff,xlab = "Univariate Regression Coefficients", ylab = "Multiple Reg
```




```
print("Here we can the diffrence in simple regression coefficients and multiple regression cofficients a
```

```
## [1] "Here we can the diffrence in simple regression coefficients and multiple regression
cofficients and the reason behind is in the simple regression models we are considering one
preditor at a time due to this the estimate cofficent beta represent the average effect of an
increase in the predictor, not taking other predictors into account. On other side we have multiple
```

regression model the estimate cofficents beta represents the average effect of an increase in the predictor, while holding other predictors fixed. It does make sense for the multiple regression to suggest no relationship between the response and some of the predictors while the simple linear regression implies the opposite because the correlation between the predictors show some strong relationships between some of the predictors."

```
## Correlation matrix

cor_Mat <- cor(Boston[-c(1,4)])
print(cor_Mat)
```

```
##                 zn      indus        nox         rm        age        dis
## zn       1.0000000 -0.5338282 -0.5166037  0.3119906 -0.5695373  0.6644082
## indus   -0.5338282  1.0000000  0.7636514 -0.3916759  0.6447785 -0.7080270
## nox     -0.5166037  0.7636514  1.0000000 -0.3021882  0.7314701 -0.7692301
## rm       0.3119906 -0.3916759 -0.3021882  1.0000000 -0.2402649  0.2052462
## age     -0.5695373  0.6447785  0.7314701 -0.2402649  1.0000000 -0.7478805
## dis      0.6644082 -0.7080270 -0.7692301  0.2052462 -0.7478805  1.0000000
## rad     -0.3119478  0.5951293  0.6114406 -0.2098467  0.4560225 -0.4945879
## tax     -0.3145633  0.7207602  0.6680232 -0.2920478  0.5064556 -0.5344316
## ptratio -0.3916785  0.3832476  0.1889327 -0.3555015  0.2615150 -0.2324705
## black    0.1755203 -0.3569765 -0.3800506  0.1280686 -0.2735340  0.2915117
## lstat   -0.4129946  0.6037997  0.5908789 -0.6138083  0.6023385 -0.4969958
## medv     0.3604453 -0.4837252 -0.4273208  0.6953599 -0.3769546  0.2499287
##                rad        tax    ptratio      black      lstat       medv
## zn      -0.3119478 -0.3145633 -0.3916785  0.1755203 -0.4129946  0.3604453
## indus    0.5951293  0.7207602  0.3832476 -0.3569765  0.6037997 -0.4837252
## nox      0.6114406  0.6680232  0.1889327 -0.3800506  0.5908789 -0.4273208
## rm      -0.2098467 -0.2920478 -0.3555015  0.1280686 -0.6138083  0.6953599
## age      0.4560225  0.5064556  0.2615150 -0.2735340  0.6023385 -0.3769546
## dis     -0.4945879 -0.5344316 -0.2324705  0.2915117 -0.4969958  0.2499287
## rad      1.0000000  0.9102282  0.4647412 -0.4444128  0.4886763 -0.3816262
## tax      0.9102282  1.0000000  0.4608530 -0.4418080  0.5439934 -0.4685359
## ptratio  0.4647412  0.4608530  1.0000000 -0.1773833  0.3740443 -0.5077867
## black   -0.4444128 -0.4418080 -0.1773833  1.0000000 -0.3660869  0.3334608
## lstat    0.4886763  0.5439934  0.3740443 -0.3660869  1.0000000 -0.7376627
## medv    -0.3816262 -0.4685359 -0.5077867  0.3334608 -0.7376627  1.0000000
```

```
## find most correlated variable

cor_Mat[lower.tri(cor_Mat,diag=TRUE)]<-NA
cor_Cof<- as.data.frame(as.table(cor_Mat))
#removing NA
cor_Cof<-cor_Cof[complete.cases(cor_Cof),]
cor_Cof<-cor_Cof[order(abs(cor_Cof$Freq),decreasing = TRUE),]
# TOP 5 STRONGEST ABSOLUTE CORRELATION
cor_Cof[1:5,]
```

```
##      Var1 Var2       Freq
## 91    rad  tax  0.9102282
## 63    nox  dis -0.7692301
## 26  indus  nox  0.7636514
## 65    age  dis -0.7478805
## 143 lstat medv -0.7376627
```

```
print("Here we can see, 'age' and 'dis' having negative correlation , In SLR 'crim' versus 'age', we sa
```

```
## [1] "Here we can see, 'age' and 'dis' having negative correlation , In SLR 'crim' versus 'age',
we saw higher values of 'age' are associated with higher values of 'crim', even though 'age' does
not actually affect 'crim'. So 'age' is a surrogate for 'dis'; 'age' gets credit for the effect of
'dis' on 'crim'."
```

Part d:Non-linear

```
## Y=??0+??1X+??2X2+??3X3+??.

## I am skipping chas as perditor because it is a factor variable
poly_p_values <- c()
poly_p_values_2 <- c()
poly_p_values_3 <- c()

# poly model with "zn"
poly_model_zn <- lm(crim ~ poly(zn, 3))
smry_poly_model_zn <- summary(poly_model_zn)
print(smry_poly_model_zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

```
poly_pvalue_zn <- smry_poly_model_zn$coefficients[2,4]
poly_pvalue_zn_2 <- smry_poly_model_zn$coefficients[3,4]
poly_pvalue_zn_3 <- smry_poly_model_zn$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_zn)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_zn_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_zn_3)
```

```r
# poly model with "indus"
poly_model_indus <- lm(crim ~ poly(indus, 3))
smry_poly_model_indus <- summary(poly_model_indus)
print(smry_poly_model_indus)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_pvalue_indus <- smry_poly_model_indus$coefficients[2,4]
poly_pvalue_indus_2 <- smry_poly_model_indus$coefficients[3,4]
poly_pvalue_indus_3 <- smry_poly_model_indus$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_indus)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_indus_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_indus_3)


# poly model with "nox"
poly_model_nox <- lm(crim ~ poly(nox, 3))
smry_poly_model_nox <- summary(poly_model_nox)
print(smry_poly_model_nox)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
```

```
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_nox <- smry_poly_model_nox$coefficients[2,4]
poly_pvalue_nox_2 <- smry_poly_model_nox$coefficients[3,4]
poly_pvalue_nox_3 <- smry_poly_model_nox$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_nox)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_nox_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_nox_3)


# poly model with "rm"
poly_model_rm <- lm(crim ~ poly(rm, 3))
smry_poly_model_rm <- summary(poly_model_rm)
print(smry_poly_model_rm)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```
poly_pvalue_rm <- smry_poly_model_rm$coefficients[2,4]
poly_pvalue_rm_2 <- smry_poly_model_rm$coefficients[3,4]
poly_pvalue_rm_3 <- smry_poly_model_rm$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_rm)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_rm_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_rm_3)


# poly model with "age"
poly_model_age <- lm(crim ~ poly(age, 3))
smry_poly_model_age <- summary(poly_model_age)
print(smry_poly_model_age)
```

```
##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_age <- smry_poly_model_age$coefficients[2,4]
poly_pvalue_age_2 <- smry_poly_model_age$coefficients[3,4]
poly_pvalue_age_3 <- smry_poly_model_age$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_age)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_age_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_age_3)



# poly model with "dis"
poly_model_dis <- lm(crim ~ poly(dis, 3))
smry_poly_model_dis <- summary(poly_model_dis)
print(smry_poly_model_dis)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_dis <- smry_poly_model_dis$coefficients[2,4]
poly_pvalue_dis_2 <- smry_poly_model_dis$coefficients[3,4]
poly_pvalue_dis_3 <- smry_poly_model_dis$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_dis)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_dis_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_dis_3)

# poly model with "rad"
poly_model_rad <- lm(crim ~ poly(rad, 3))
smry_poly_model_rad <- summary(poly_model_rad)
print(smry_poly_model_rad)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_rad <- smry_poly_model_rad$coefficients[2,4]
poly_pvalue_rad_2 <- smry_poly_model_rad$coefficients[3,4]
poly_pvalue_rad_3 <- smry_poly_model_rad$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_rad)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_rad_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_rad_3)

# poly model with "tax"
poly_model_tax <- lm(crim ~ poly(tax, 3))
smry_poly_model_tax <- summary(poly_model_tax)
print(smry_poly_model_tax)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3))
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_tax <- smry_poly_model_tax$coefficients[2,4]
poly_pvalue_tax_2 <- smry_poly_model_tax$coefficients[3,4]
poly_pvalue_tax_3 <- smry_poly_model_tax$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_tax)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_tax_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_tax_3)

# poly model with "ptratio"
poly_model_ptratio <- lm(crim ~ poly(ptratio, 3))
smry_poly_model_ptratio <- summary(poly_model_ptratio)
print(smry_poly_model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```
poly_pvalue_ptratio <- smry_poly_model_ptratio$coefficients[2,4]
poly_pvalue_ptratio_2 <- smry_poly_model_ptratio$coefficients[3,4]
```

```r
poly_pvalue_ptratio_3 <- smry_poly_model_ptratio$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_ptratio)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_ptratio_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_ptratio_3)

# poly model with "black"
poly_model_black <- lm(crim ~ poly(black, 3))
smry_poly_model_black <- summary(poly_model_black)
print(smry_poly_model_black)
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
poly_pvalue_black <- smry_poly_model_black$coefficients[2,4]
poly_pvalue_black_2 <- smry_poly_model_black$coefficients[3,4]
poly_pvalue_black_3 <- smry_poly_model_black$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_black)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_black_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_black_3)


# poly model with "lstat"
poly_model_lstat <- lm(crim ~ poly(lstat, 3))
smry_poly_model_lstat <- summary(poly_model_lstat)
print(smry_poly_model_lstat)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
```

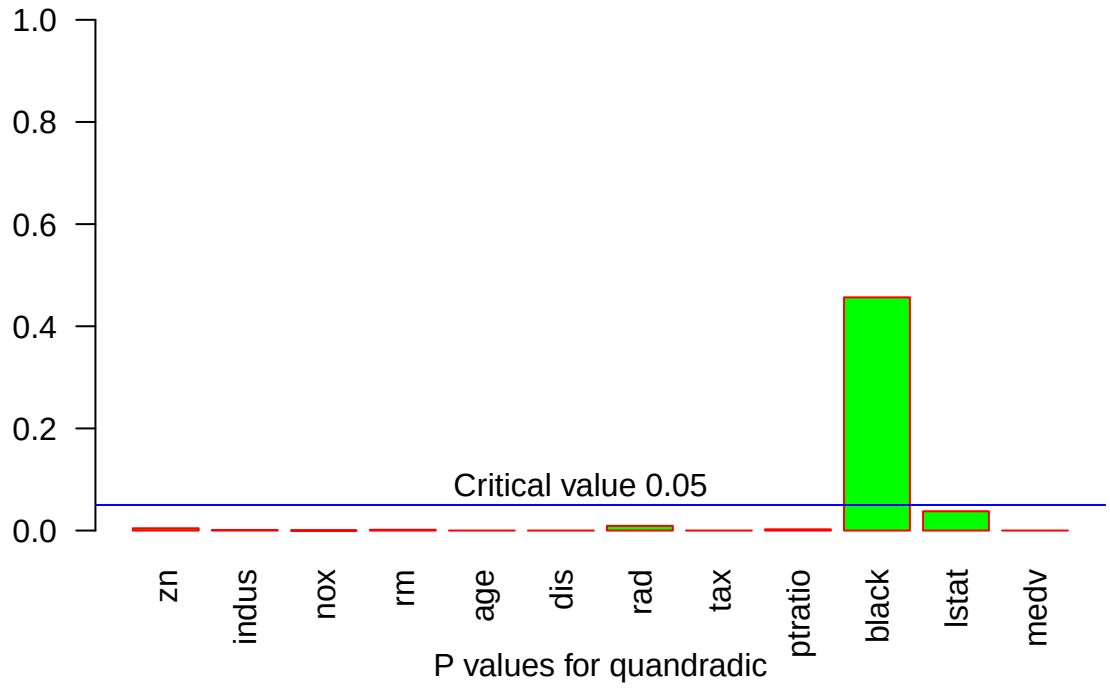
```
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_lstat <- smry_poly_model_lstat$coefficients[2,4]
poly_pvalue_lstat_2 <- smry_poly_model_lstat$coefficients[3,4]
poly_pvalue_lstat_3 <- smry_poly_model_lstat$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_lstat)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_lstat_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_lstat_3)

# poly model with "medv"
poly_model_medv <- lm(crim ~ poly(medv, 3))
smry_poly_model_medv <- summary(poly_model_medv)
print(smry_poly_model_medv)
```

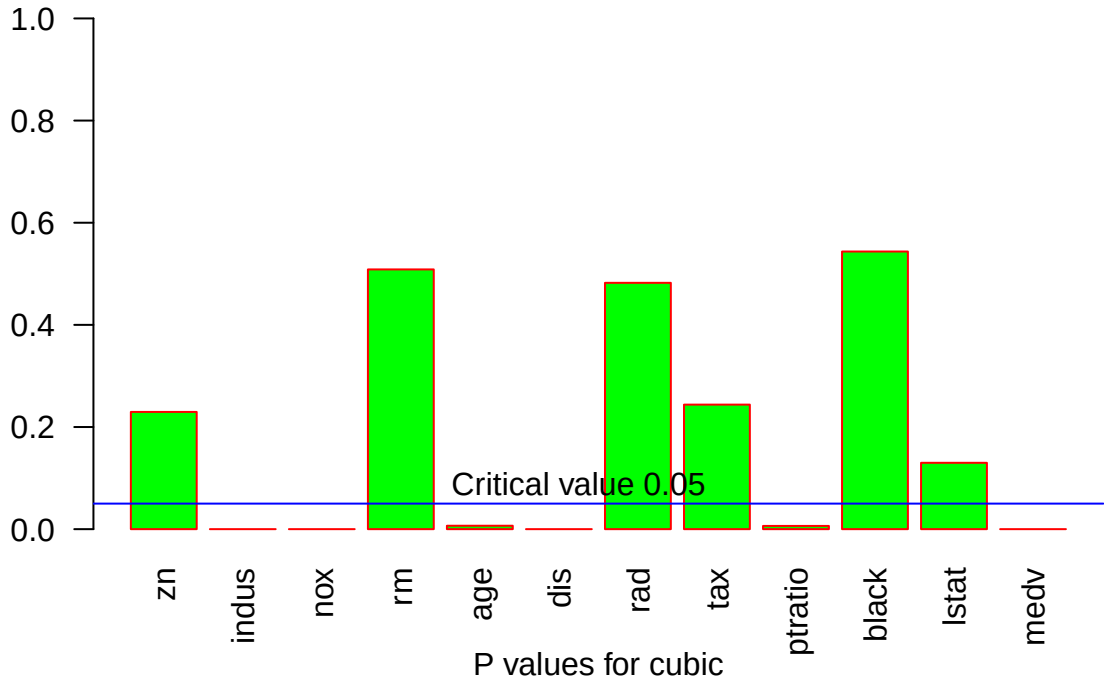
```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_medv <- smry_poly_model_medv$coefficients[2,4]
poly_pvalue_medv_2 <- smry_poly_model_medv$coefficients[3,4]
poly_pvalue_medv_3 <- smry_poly_model_medv$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_medv)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_medv_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_medv_3)
```

```r
# Plot the bar chart on P values for quandradic
barplot(poly_p_values_2,las=2,names.arg=colnames(boston_df)[-c(1,4)],xlab="P values for quandradic",yli
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```




```r
# Plot the bar chart on P values for cubic
barplot(poly_p_values_3,las=2,names.arg=colnames(boston_df)[-c(1,4)],xlab="P values for cubic",ylim = c
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```




```
print("From both graph P value for quandradic and p value for cubic we can conclude that 'zn', 'rm', 'ra
```

```
## [1] "From both graph P value for quandradic and p value for cubic we can conclude that 'zn',
'rm', 'rad', 'tax', 'black' and 'lstat' as perdictor are not statistically significant but for
other predictors like 'indus', 'nox', 'age', 'dis', 'ptratio' and 'medv' with acceptable p value.
In case of full model incuding all predictors in non-linear is not visible"
```

Homework#3 Problem-4

Arinjay Jain

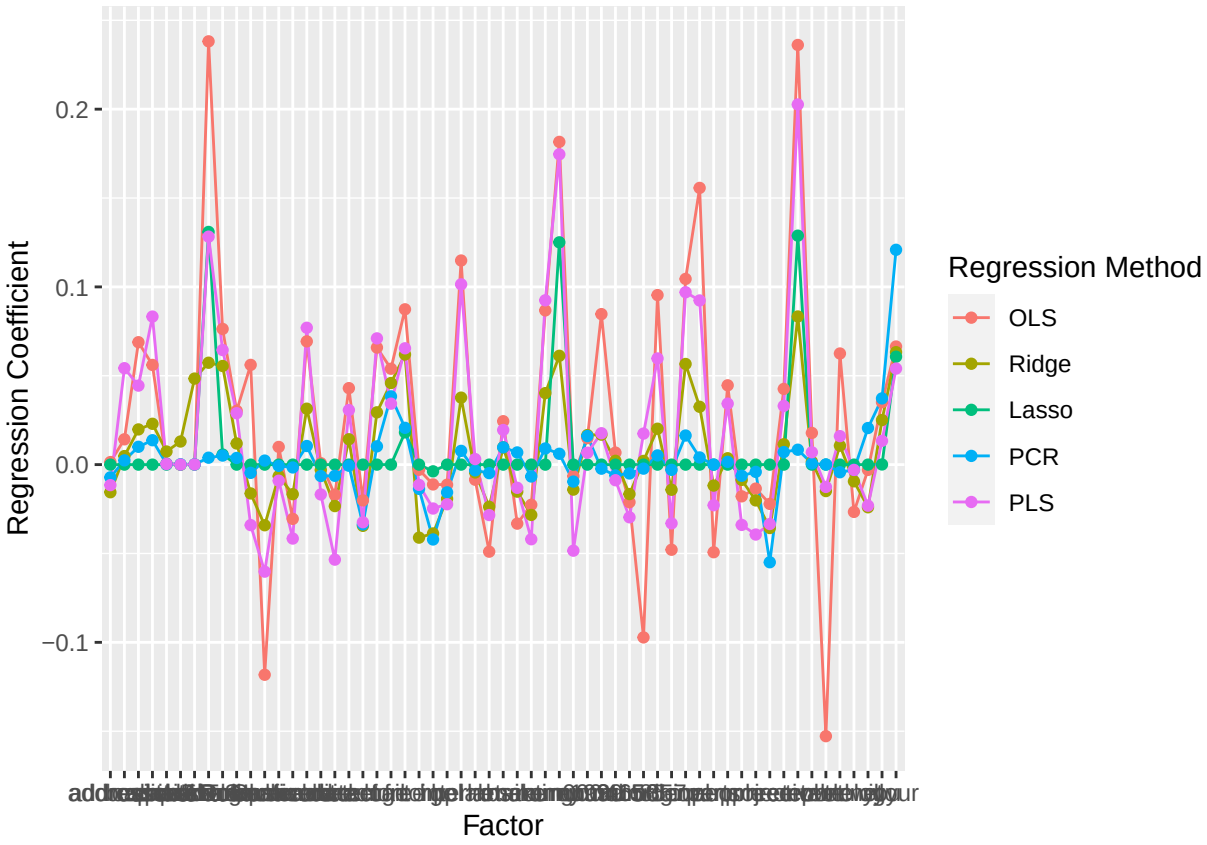
```
mod.ls <- lm(type ~ .-1, spam_df)
mod.ridge <- lm.ridge(type ~ ., spam_df)
mod.pcr <- pcr(formula =type ~ ., data = spam_df, validation ="CV")
mod.plsr <- plsr(formula =type ~ ., data = spam_df, validation ="CV")

mod.lasso <- lars( as.matrix(spam_df[,1:ncol(spam_df) - 1]) ,
spam_df[, ncol(spam_df)],type ="lasso")

mods.coeffs <- data.frame(ls = mod.ls$coef,
ridge = mod.ridge$coef,
lasso = mod.lasso$beta[10 ,],
pcr = mod.pcr$coef[ , ,10],
plsr = mod.plsr$coef [ , ,10]
)
print(mods.coeffs)
```

```
##                           ls        ridge         lasso           pcr
## make             -3.320614e-02 -0.015210964  0.000000e+00  6.852457e-03
## address           1.445165e-03 -0.015544883  0.000000e+00 -7.179303e-03
## all               6.886632e-02  0.019799864  0.000000e+00  1.002829e-02
## num3d             1.488479e-02  0.016624613  0.000000e+00  1.607165e-02
## our               1.044914e-01  0.056624615  0.000000e+00  1.637237e-02
## over              1.556968e-01  0.032537865  0.000000e+00  4.027027e-03
## remove            2.361170e-01  0.083344676  1.288261e-01  8.388154e-03
## internet          1.148364e-01  0.037692167  0.000000e+00  7.743828e-03
## order             9.539755e-02  0.020190220  0.000000e+00  5.172981e-03
## mail              2.441980e-02  0.009713729  0.000000e+00  9.728780e-03
## receive           4.259627e-02  0.011457923  0.000000e+00  7.064517e-03
## will             -2.967378e-03 -0.024003875  0.000000e+00  2.059066e-02
## people            4.461982e-02  0.003583153  0.000000e+00  1.571913e-03
## report            1.782142e-02  0.001628810  0.000000e+00  6.847997e-04
## addresses         1.423489e-02  0.004794431  0.000000e+00  2.377096e-03
## free              8.738923e-02  0.061978312  1.811450e-02  2.070649e-02
## business          5.607731e-02  0.022962142  0.000000e+00  1.372234e-02
## email             6.586632e-02  0.029419879  0.000000e+00  1.028947e-02
## you               3.509620e-02  0.025090798  8.243847e-05  3.725894e-02
## credit            6.934306e-02  0.031460401  0.000000e+00  1.047542e-02
## your              6.644707e-02  0.063268269  6.072533e-02  1.208755e-01
## font              5.395411e-02  0.045915010  0.000000e+00  3.840729e-02
## num000            1.815880e-01  0.061223563  1.250773e-01  6.105218e-03
## money             8.676865e-02  0.040227122  0.000000e+00  9.082325e-03
## hp               -1.114529e-02 -0.038729258 -3.787290e-03 -4.212542e-02
## hpl              -1.132212e-02 -0.019181830  0.000000e+00 -1.561358e-02
## george           -2.894929e-03 -0.041081890  0.000000e+00 -1.364055e-02
```

```
## num650             6.756574e-03  0.002147293  0.000000e+00 -6.090028e-03
## lab               -8.492401e-03 -0.004419532  0.000000e+00 -3.257424e-03
## labs              -4.897516e-02 -0.023721288  0.000000e+00 -4.782031e-03
## telnet            -2.659938e-02 -0.009395594  0.000000e+00 -2.935618e-03
## num857            -9.721247e-02  0.002080130  0.000000e+00 -2.255951e-03
## data              -1.699704e-02 -0.023336347  0.000000e+00 -6.296787e-03
## num415             8.467819e-02  0.016846585  0.000000e+00 -2.288171e-03
## num85             -2.114293e-02 -0.016588055  0.000000e+00 -4.966276e-03
## technology         6.252226e-02  0.010660327  0.000000e+00 -4.296559e-03
## num1999           -6.526106e-03 -0.014062357  0.000000e+00 -9.551978e-03
## parts             -4.929705e-02 -0.011789989  0.000000e+00  7.146520e-06
## pm                -1.784907e-02 -0.008585837  0.000000e+00 -6.273293e-03
## direct             4.302282e-02  0.014261336  0.000000e+00 -6.161637e-04
## cs                 8.134337e-04 -0.003020819  0.000000e+00 -6.413008e-03
## meeting           -2.260310e-02 -0.028312897  0.000000e+00 -6.677286e-03
## original          -4.790158e-02 -0.014152090  0.000000e+00 -2.938567e-03
## project           -1.349720e-02 -0.020137399  0.000000e+00 -4.080641e-03
## re                -2.208665e-02 -0.035661420  0.000000e+00 -5.493849e-02
## edu               -2.013634e-02 -0.034449413  0.000000e+00 -3.365203e-02
## table             -1.527856e-01 -0.014885396  0.000000e+00  8.938687e-05
## conference        -3.052703e-02 -0.016634503  0.000000e+00 -1.596912e-03
## charSemicolon     -1.182385e-01 -0.034107231  0.000000e+00  2.269575e-03
## charRoundbracket   5.610857e-02 -0.016208572  0.000000e+00 -4.638150e-03
## charSquarebracket  9.956557e-03 -0.006459196  0.000000e+00 -7.150786e-04
## charExclamation    7.633179e-02  0.055502858  5.673513e-03  5.091158e-03
## charDollar         2.381562e-01  0.057328041  1.309207e-01  3.895747e-03
## charHash           3.028379e-02  0.011888664  0.000000e+00  3.746403e-03
## capitalAve         5.830736e-04  0.007381713  0.000000e+00  3.925910e-04
## capitalLong       -4.652227e-05  0.013008326  0.000000e+00  1.737404e-04
## capitalTotal       1.123789e-04  0.048419005  1.580004e-05  1.169891e-04
##                           plsr
## make              -1.306140e-02
## address           -1.153633e-02
## all                4.446061e-02
## num3d              6.722095e-03
## our                9.698674e-02
## over               9.236715e-02
## remove             2.026510e-01
## internet           1.014336e-01
## order              5.973678e-02
## mail               1.946713e-02
## receive            3.302884e-02
## will              -2.310075e-02
## people             3.429487e-02
## report             6.946563e-03
## addresses          5.425777e-02
## free               6.543878e-02
## business           8.330377e-02
## email              7.102060e-02
## you                1.338095e-02
## credit             7.700125e-02
## your               5.404990e-02
## font               3.429337e-02
## num000             1.746184e-01
```

```
## money              9.245503e-02
## hp                -2.470157e-02
## hpl               -2.229266e-02
## george            -1.140491e-02
## num650            -8.773347e-03
## lab                3.101716e-03
## labs              -2.837491e-02
## telnet            -2.995848e-03
## num857             1.745671e-02
## data              -5.346435e-02
## num415             1.766427e-02
## num85             -2.964026e-02
## technology         1.602170e-02
## num1999           -4.842552e-02
## parts             -2.283924e-02
## pm                -3.392767e-02
## direct             3.069870e-02
## cs                -1.679095e-02
## meeting           -4.197962e-02
## original          -3.304476e-02
## project           -3.934925e-02
## re                -3.343689e-02
## edu               -3.242758e-02
## table             -1.254560e-02
## conference        -4.159883e-02
## charSemicolon     -6.021949e-02
## charRoundbracket  -3.403010e-02
## charSquarebracket -9.055554e-03
## charExclamation    6.446419e-02
## charDollar         1.282678e-01
## charHash           2.892184e-02
## capitalAve         3.150601e-04
## capitalLong        3.996749e-05
## capitalTotal       8.533364e-05
```

```
mods.coeffs$xs = row.names(mods.coeffs)
plot.data <- melt(mods.coeffs, id ="xs")



ggplot (data = plot.data,
aes (x = factor(xs),y= value,group = variable,
colour = variable)) +
geom_line() +
geom_point() +
xlab("Factor") +
ylab("Regression Coefficient") +
scale_colour_hue(name ="Regression Method ",
labels = c("OLS",
"Ridge",
"Lasso",
"PCR",
"PLS")
)
```


0.2
0.1
0.0
−0.1
Regression Coefficient
Factor
Regression Method
OLS
Ridge
Lasso
PCR
PLS