Shraddha A20499171

Exercise-1

We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . ., p predictors. Explain your answers:

a) **Which of the three models with k predictors has the smallest training RSS?**
Answer is best using p predictors so there is no subset model has smallest training RSS as it trains all possible 2p models that couldn't be identified by best subset selection.

b) Which of the three models with k predictors has the smallest test RSS?
As best subset model checks more combinations of predictors so it should result in smallest test RSS, but it is also possible to overfit with any of these three methods. It might be possible that stepwise models perform better on test set compared to best subset. Hence its difficult to decide.

c) True or False
 i. True K+1 is obtained by adding one more variable from k variable model.
 ii. True, same reason we can also obtain by removing 1 variable from K+1 model
 iii. False,
 iv. False
 v. False, may be possible but not all times.

Exercise-2

a) **indicate which of i. through iv. is correct. Justify your answer**
The option **III** is correct as Lasso is more restrictive model, so it increases the bias, it has the chance of less overfitting to the data and reducing the variance.

b) **Repeat (a) for ridge regression relative to least squares**
The option III is correct as ridge regression is less flexible as it reduces predictors which are not associated with target variable. (Lasso and ridge both are same except shape of constraints)

c) **Repeat (a) for non-linear methods relative to least squares.**
The option **II** is correct as we make less assumption about function f which result in reduced bias that is higher than increase in variance hence improve prediction accuracy.

Exercise-3

a) As we increase s from 0, the training RSS will
The option **IV** is correct as we increase s the model flexibility increases and reduce restriction on beta which results in steady decrease in the training RSS.

b) Repeat (a) for test RSS
The option **II** is correct as we increase s model become more flexible as result test RSS decrease initially until increase in variance is less than reduction in bias, but at some point, it starts gradually increase and give the U shape curve.

c) Repeat (a) for variance

The option **III** is correct as we increase s, model become more flexible which result on increase in variance.

d) Repeat (a) for (squared) bias

The option **IV** is correct as we increase s, the model flexibility increases and bias of model decrease.

e) Repeat (a) for the irreducible error

The option **V** is correct as irreducible error does not depend on flexibility of the model and not depend on s value.

Exercise-4

a) The option **III** is correct as we increase lambda implied decrease flexibility of model, as with increase in lambda we restrict the coefficients which increase in the training RSS.

b) The option **II** is correct as we increase lambda model become less flexible which at first reduce in the test RSS then start increasing again with overfitting of model and yield U shape curve.

c) The option **IV** is correct as we increase lambda model become less flexible and variance always decrease as flexibility of model decrease.

d) The option **III** is correct we increase lambda model become less flexible and more constrained hence results steady increase bias.

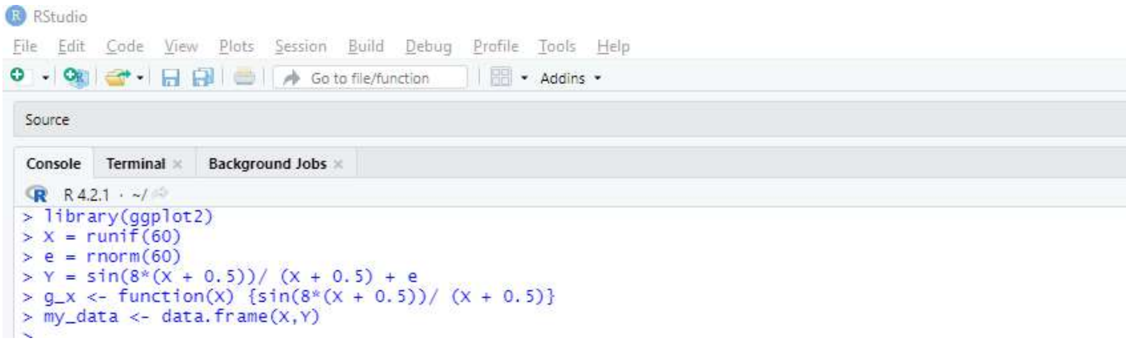
e) The option **V** is correct as irreducible error does not depend on flexibility of the model and not depend on lambda value.

Exercise-5

Chapter-7

Exercise-2

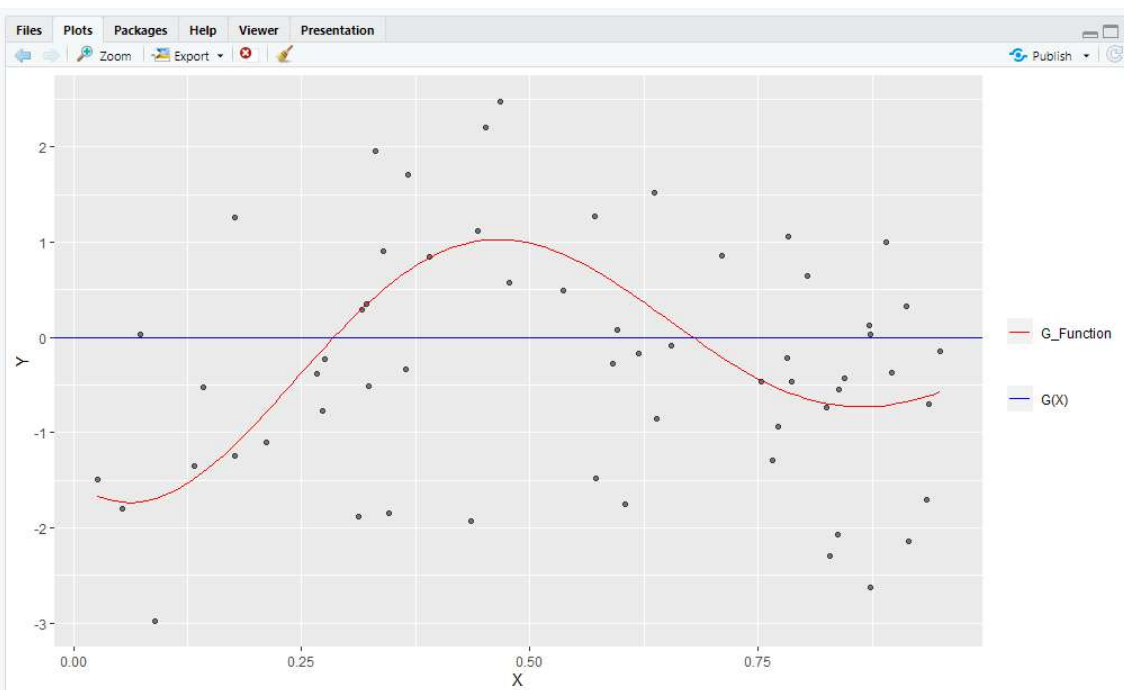
Provide example sketches of ˆg in each of the following scenarios



(a) λ = ∞, m = 0

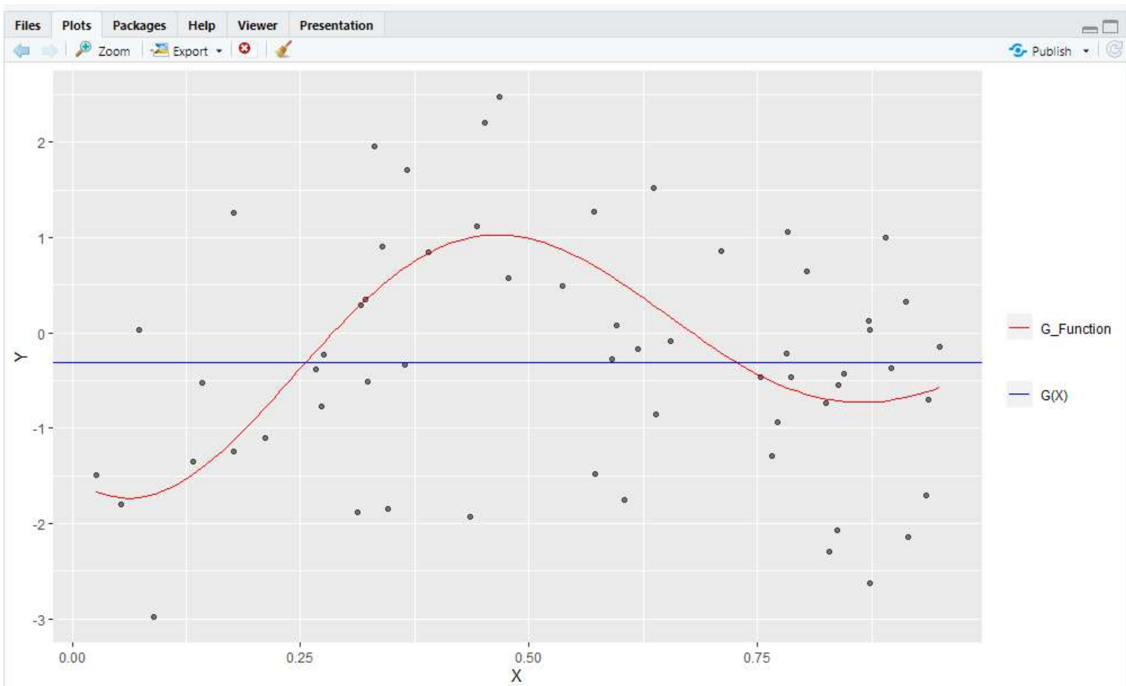
g = 0 as large smoothing parameter results in g(0) (x) → 0

```
> ggplot(my_data,aes(x=X, y=Y))+ geom_point(alpha = 0.5)+ stat_function(fun = g_x, aes(col = "G_Function"))+ g
eom_hline(aes(yintercept = 0, linetype = "G(X)"),col="blue") + scale_colour_manual(values = "red") +  theme(le
gend.position = "right",legend.title = element_blank())
> |
```



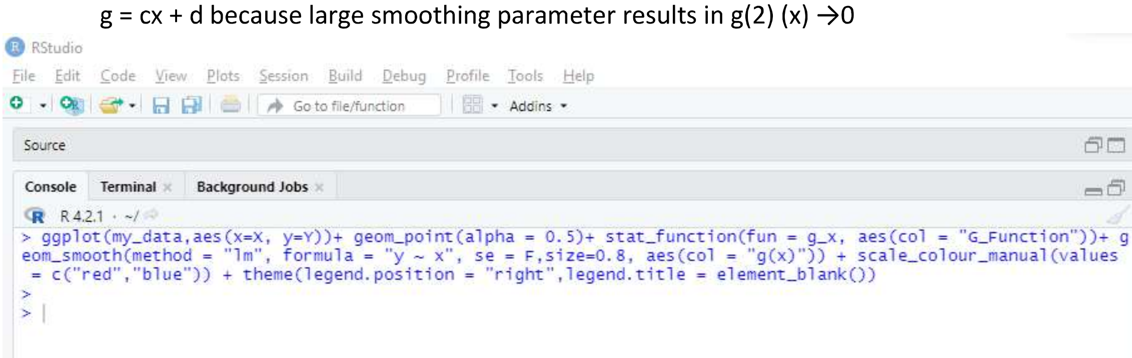
(b) λ = ∞, m = 1.

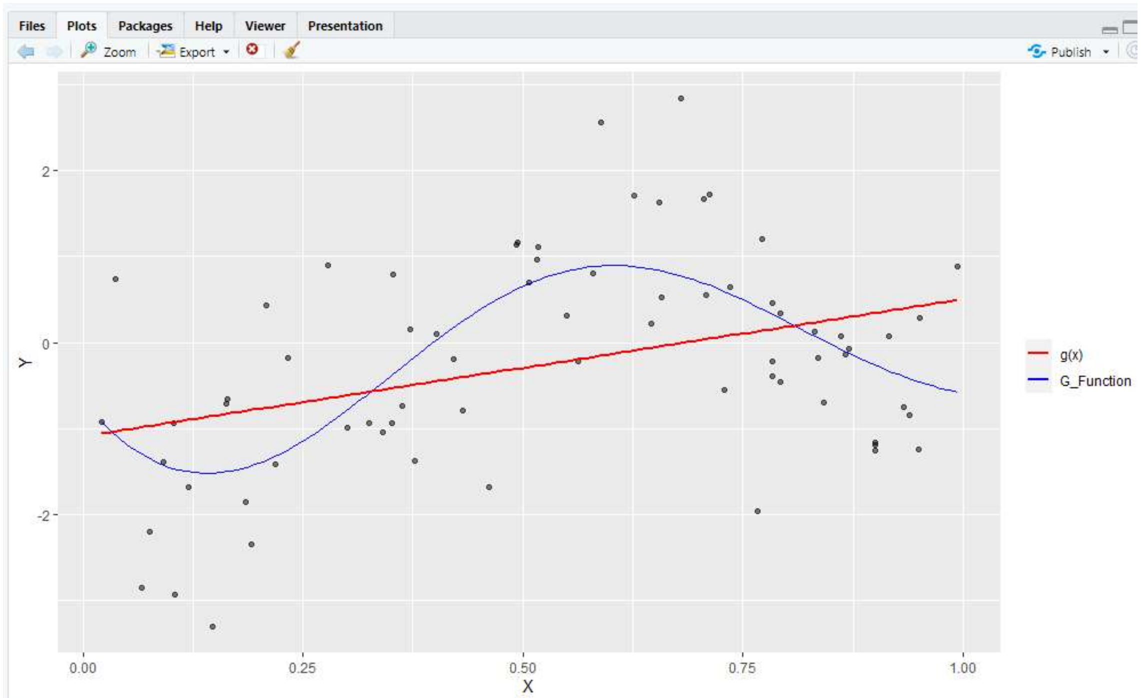
g = c because large smoothing parameter results in g(1) (x) → 0



(c) λ = ∞, m = 2.

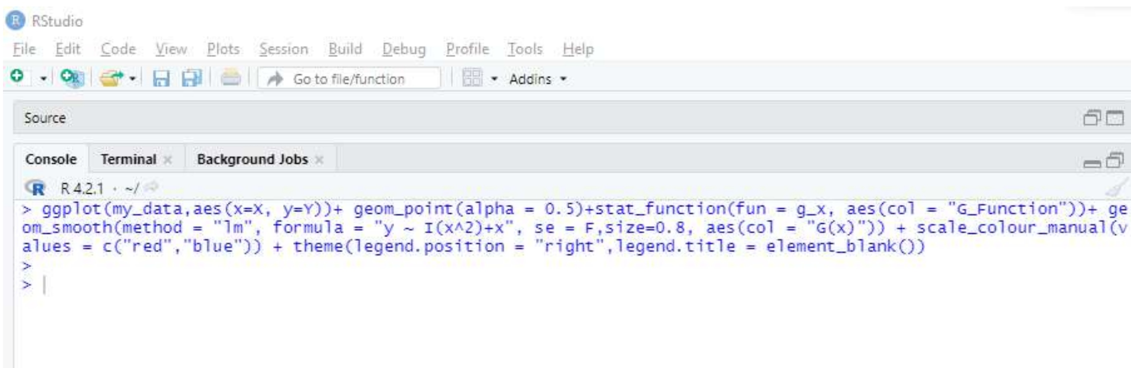
g = cx + d because large smoothing parameter results in g(2) (x) →0

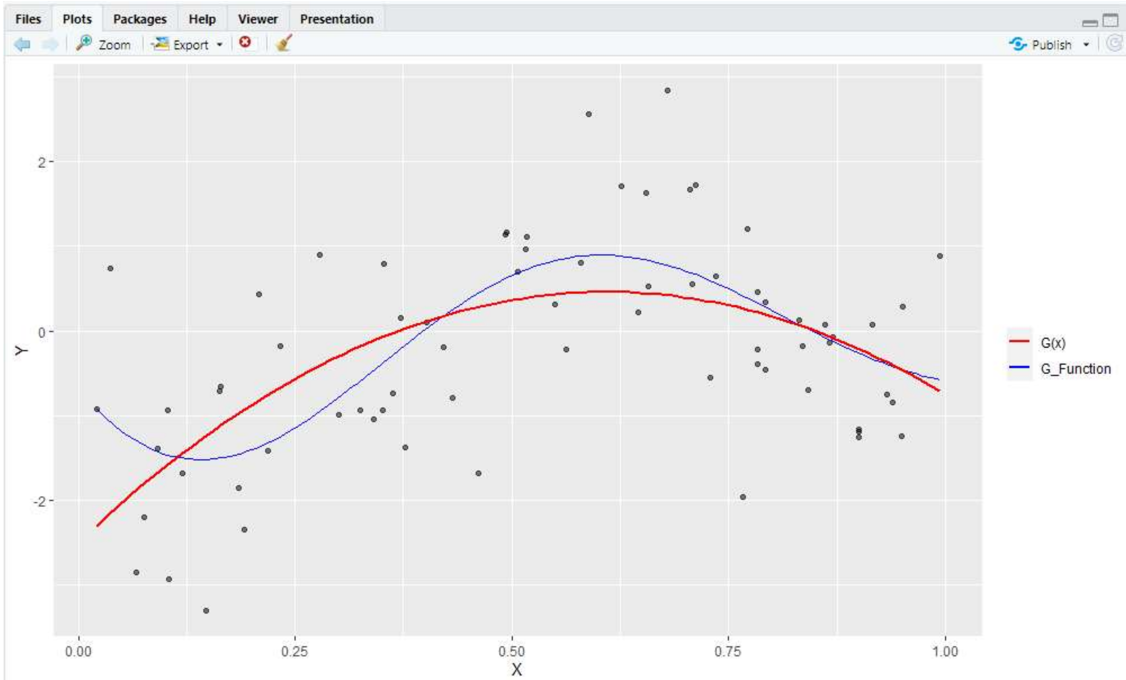




(d) λ = ∞, m = 3

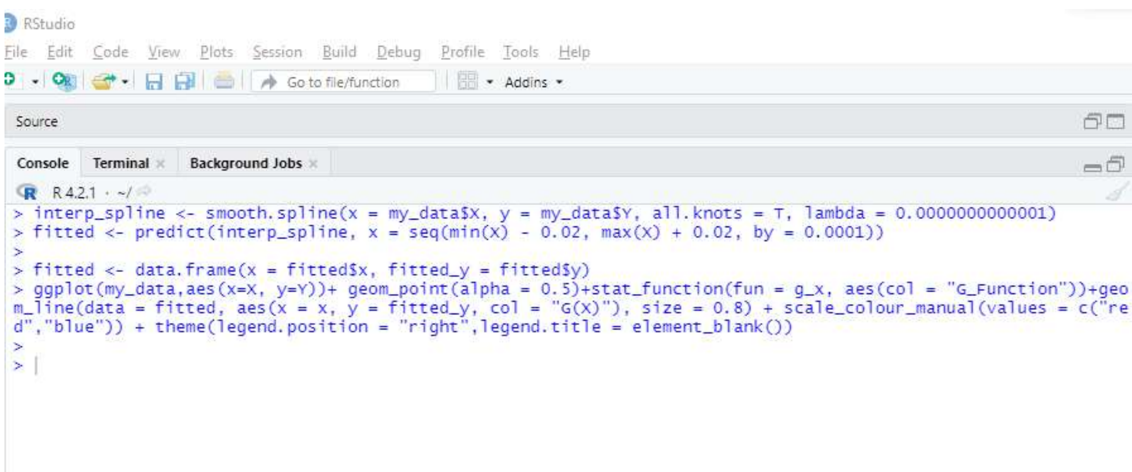
g = cx2 + dx + e because large smoothing parameter results in g(3) (x) → 0

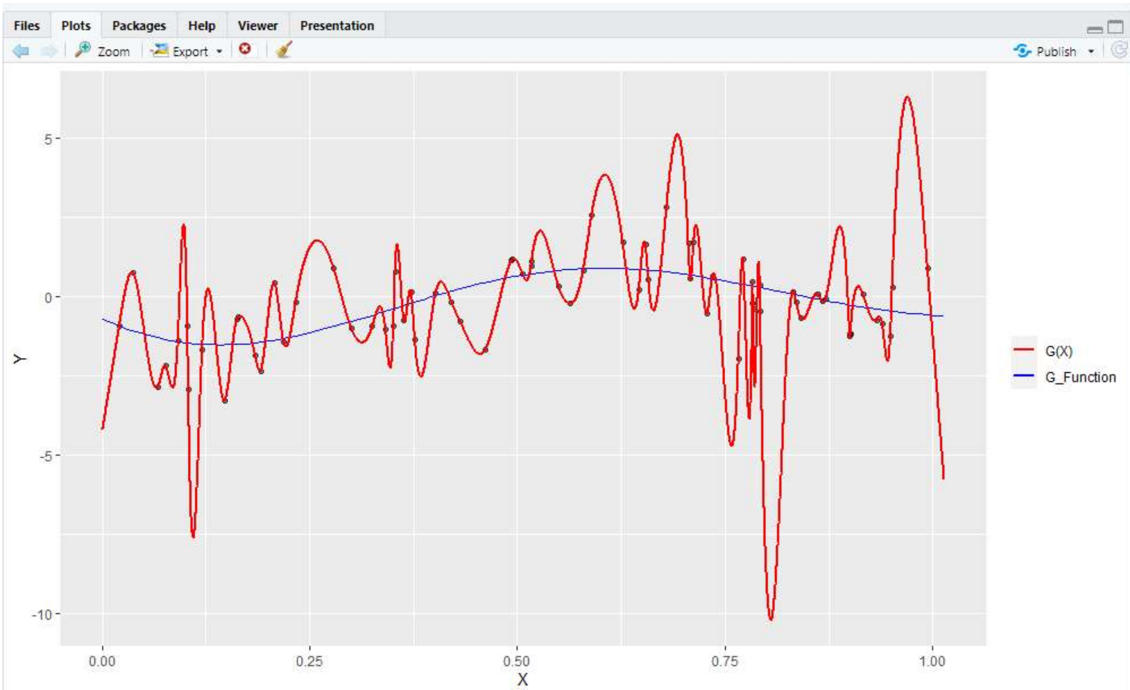




(e) λ = 0, m = 3

Here g is interpolating spline. As an example, it fits low degree polynomials to the small subset of values





Exercise-3

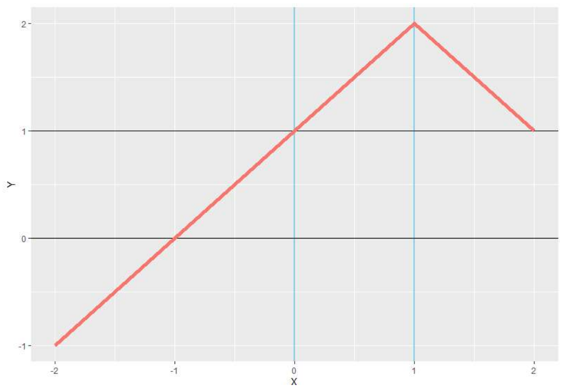
Suppose we fit a curve with basic functions b1(X) = X, b2(X) = $(X - 1)^2 I(X \geq 1)$. (Note that $I(X \geq 1)$ equals 1 for X ≥ 1 and 0 otherwise.)We fit the linear regression model Y = β0 + β1b1(X) + β2b2(X) + ϵ, and obtain coefficient estimates β^0 = 1, β^1 = 1, β^2 = −2. Sketch the estimated curve between X = −2 and X = 2.

Answer:

Range of X is -2 :2

Value of Y from given Equation Y = β0 + β1b1(X) + β2b2(X) + ϵ

Y=1+1b1(x)-2b2(x) => Y= $1+x-2(x-1)^2$ * I (x>=1)



The given curve is:

Linear between −2 and 1

Y=1+x

Quadratic between 1 and 2

Y=1+x−2(x−1)^2.

For x<1, Y=1+x so intersects=1 and slope=1

For x>=1 Y=1+x-2(x-1)^2 slope will vary

Exercise-4

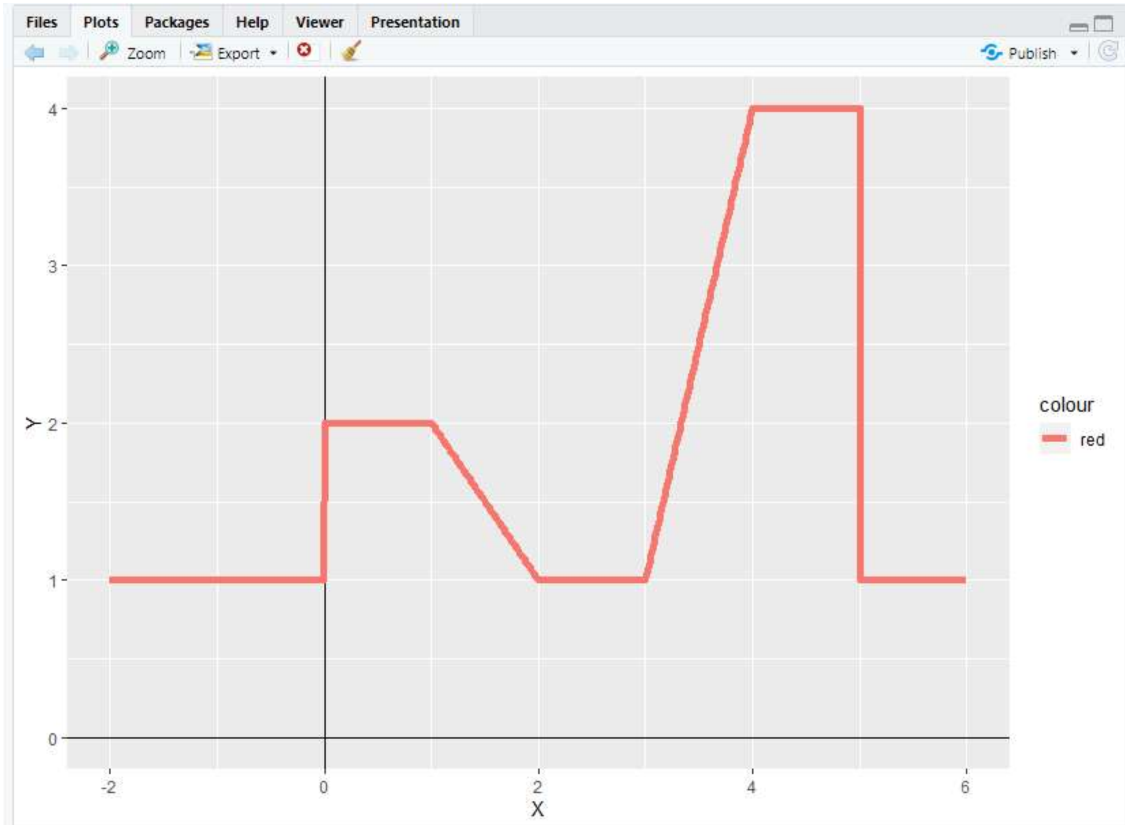
Suppose we fit a curve with basic functions b1(X) = *I*(0 ≤ X ≤ 2) − (X −1)*I*(1 ≤ X ≤ 2), b2(X)=(X −3)I(3 ≤ X ≤ 4) +I(4 < X ≤ 5). We fit the linear regression model Y = β0 + β1b1(X) + β2b2(X) + ϵ, and obtain coefficient estimates βˆ0 = 1, βˆ1 = 1, βˆ2 = 3. Sketch the estimated curve between X = −2 and X = 6. Note the intercepts, slopes, and other relevant information.

Answer:

Y = β0 + β1b1(X) + β2b2(X) + ϵ

Y= 1 + 1* I (0<=X<=2) -(X-1) I (1<=X<=2) + 3 * ((X-3) I (3<=X<=4) + I (4<X<=5))

Y= 1 + 1* I (0<=X<=2) -(X-1) I (1<=X<=2) + 3 * (X-3) I (3<=X<=4) + 3*I (4<X<=5)



The curve is constant between −2 and 0 with slope=0, Constant between 0 and 1 with y=2 and slope=0, curve is Linear between 1 and 2 with y=3−x and slope=-1, curve is constant from 2 to 3 with y=1 and slope=0. For x value greater than 3 and less or equal to 4 slope is y=2x-8 and slope is 2, for value between 4 to 5 remain constant with y=4 and slope=0

Exercise-5

(a) As λ → ∞, will g1 or g2 have the smaller training RSS?

Answer: -

Smoothing spine g2 has higher order polynomial and smaller training RSS due to order of penalty term.

(b) As λ → ∞, will g1 or g2 have the smaller test RSS?

Answer: -

Smoothing spine g1 has smaller test RSS.As Smoothing spine g2 have higher order polynomial and more flexibility so it may overfit the data.

(c) For λ = 0, will g1 or g2 have the smaller training and test RSS

Answer: -

We have λ = 0 implied g1=g2, so for both g1 and g2 we will have same training and testing RSS

Exercise-5

$n = 2, \; p = 2 \quad x_{11} = x_{12}, \; x_{21} = x_{22}$

$y_1 + y_2 = 0 \qquad x_{11} + x_{21} = 0 \qquad x_{12} + x_{22} = 0$

(a)

$$\min \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$\Rightarrow \quad \beta_0 = 0$

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$\Rightarrow$ from given information.

$$(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda(\beta_1^2 + \beta_2^2)$$

$$\Rightarrow (y_1 - B_1 x_{11} - B_2 x_{12})^2 + (y_2 - B_1 x_{21} - B_2 x_{22})^2 + \lambda (B_1^2 + B_2^2)$$

(b)

from equation

$$\Rightarrow (y_1 - B_1 x_{11} - B_2 x_{12})^2 + (y_2 - B_1 x_{21} - B_2 x_{22})^2 + \lambda (B_1^2 + B_2^2)$$

$\Rightarrow$ We have

$$x_{11} + x_{21} = 0 \Rightarrow x_{11} = -x_{21}$$

$$x_{12} + x_{22} = 0 \Rightarrow x_{12} = -x_{22}$$

$$y_1 + y_2 = 0 \Rightarrow y_1 = -y_2$$

$$\Rightarrow (y_1 - B_1 x_{11} - B_2 x_{12})^2 + (-y_1 + B_1 x_{11} + B_2 x_{12})^2 + \lambda (B_1^2 + B_2^2)$$

$$\Rightarrow (y_1 - B_1 x_{11} - B_2 x_{12})^2 + (y_1 - B_1 x_{11} + B_2 x_{12})^2 + \lambda (B_1^2 + B_2^2$$

$$\Rightarrow 2\left[y_1 - B_1 x_{11} - B_2 x_{12} \right]^2 + \lambda \left[B_1^2 + B_2^2 \right.$$

$\Rightarrow$ equating yields

$$2y_1^2 - 4B_1x_{11}y_1 - 4B_2x_{11}y_1 +$$
$$4B_1B_2x_{11}^2 + 2B_1^2x_{11}^2 + 2B_2^2x_{11}^2 +$$
$$\lambda B_1^2 + \lambda B_2^2 .$$

$\Rightarrow$ derivation with respect to B_1 and B_2 and set to zero

gives. $-4y_1x_{11} + 4x_{11}^2B_2 +$
$4x_{11}^2B_1 + 2\lambda B_1 = 0$

$$\Rightarrow 4y_1x_{11} - 4x_{11}^2B_2 = 4x_{11}^2B_1 + 2\lambda B_1$$

$$\Rightarrow 2(2y_1x_{11} - 2x_{11}^2B_2) = 2B_1(2x_{11}^2 + \lambda)$$

$$\Rightarrow 2y_1x_{11} - 2x_{11}^2B_2 = B_1(\lambda + 2x_{11}^2)$$

$$\Rightarrow B_1 = \frac{(2y_1x_{11} - 2x_{11}^2B_2)}{(\lambda + 2x_{11}^2)}$$

Similarly

$$B_2 = \frac{(2y_1x_{11} - 2x_{11}^2\beta_1)}{\lambda + 2x_{11}^2}$$

To prove

$\Rightarrow$ ~~Simplify~~ B_1 & B_2 equal

By putting value of B_2 in B_1.
and B_2 into B_1 will

~~B_1~~ get same result

So can prove

$B_1 = B_2$, as required.

(C) Write down lasso optimization

$$\Rightarrow \min_{\beta_j} \left\{ \sum_{i=1}^{2} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

$$\Rightarrow \min \left[(y_1 - B_1 x_{11} - B_2 x_{12})^2 + (y_2 - B_1 x_{21} - B_2 x_{22})^2 \right] + \lambda \left(|B_1| + |B_2| \right)$$

(d) lasso coefficient B1 and B2 are not unique.

$$\Rightarrow \sum_{i=1}^{n} \left(y_i - B_0 - \sum_{j=1}^{p} B_j x_{ij} \right)^2$$

$$\text{where } \sum_{j=1}^{p} |B_j| \leq s.$$

$$\Rightarrow (y_1 - B_1 x_{11} - B_2 x_{12})^2 + (y_2 - B_1 x_{21} - B_2 x_{22})^2$$

$$B_1 + B_2 \in S$$

$$\Rightarrow 2\left[y_1 - (B_1 + B_2) x_{11} \right]^2 \geq 0.$$

$$\left(\because x_{11} + x_{12} = 0, \quad x_{21} + x_{22} = 0, \quad y_1 + y_2 = 0, \quad x_{11} = x_{12}, \quad x_{21} = x_{22} \right)$$

$\Rightarrow$ RSS has minimum of zero. $(x_{11} = x_{4})$

so $y_1 - (B_1 + B_2)x_1 = 0$

$B_1 + B_2 x_1 = y_1$

$$\boxed{\Rightarrow B_1 + B_2 = y_1/x_1.}$$

Will has RSS of zero.

$\Rightarrow$ We also here to consider lasso constraints.

Solution is the intersection of contours of the function

$[y_1 - (B_1 + B_2)x_1]^2$ with

~~dia~~ lasso diamond.

$\Rightarrow$ So, as B_1, B_2 vary along the line

$B_2 = -B_1 + y_1/x_1$

contour will touch lasso diamond. at many

- points.

⇒ Thus, lasso optimization has many solution not just one.

Solution

$$\beta \in \{\beta_1, \beta_2 : \beta_1 + \beta_2 = 5,\ \beta_1 \in [0, 5],\ \beta_2 \in [0, 5]\}$$

$$\cup$$

$$\{(\beta_1, \beta_2) : \beta_1 + \beta_2 = -5,\ \beta_1 \in [-5, 0],\ \beta_2 \in [-5, 0]\}$$

Assignment-3

Problem 2.1

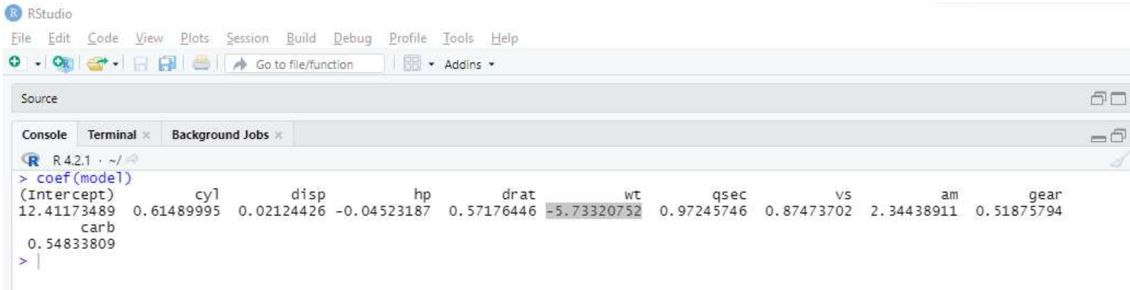
Load the mtcars sample dataset from the built-in datasets (data(mtcars)) into R using a dataframe.

```
> car_data=data.frame(mtcars)
> head(car_data)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> str(car_data)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
>
```

Perform a basic 80/20 test-train split on the data and fit a linear model with mpg as the target response, and all other variables as predictors/features

```
> library(caret)
> set.seed(220)
> data_partition=createDataPartition(car$am,times = 1,p= 0.8,list = F)
> train=car_data[data_partition, ]
> test=car_data[-data_partition, ]
> model=lm(mpg~.,data = train)
> mean((predict(model,test)-test$mpg)^2)
[1] 11.7182
> summary(model)

Call:
lm(formula = mpg ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3884 -1.7832 -0.2373  1.5105  4.3461 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 12.41173   19.80213   0.627   0.5402  
cyl          0.61490    1.15784   0.531   0.6031  
disp         0.02124    0.02062   1.030   0.3191  
hp          -0.04523    0.02823  -1.602   0.1300  
drat         0.57176    2.11851   0.270   0.7909  
wt          -5.73321    2.32916  -2.461   0.0264 *
qsec         0.97246    0.75980   1.280   0.2200  
vs           0.87474    2.38567   0.367   0.7190  
am           2.34439    2.27593   1.030   0.3193  
gear         0.51876    1.77446   0.292   0.7740  
carb         0.54834    1.06856   0.513   0.6153  
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.674 on 15 degrees of freedom
Multiple R-squared:  0.8821,	Adjusted R-squared:  0.8036 
F-statistic: 11.23 on 10 and 15 DF,  p-value: 2.845e-05

>
```

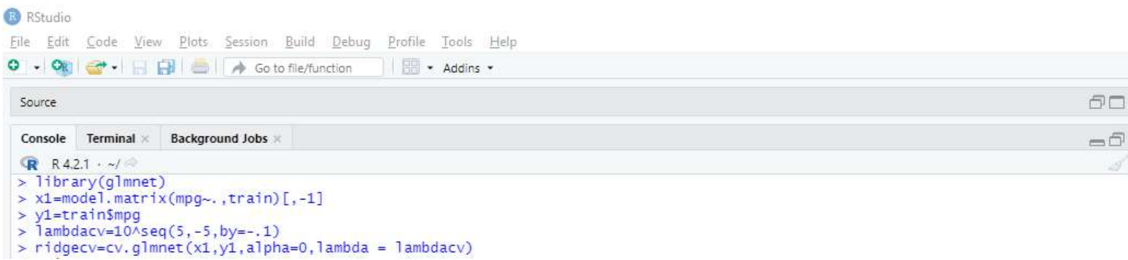
What features are selected as relevant based on resulting t-statistics? What are the associated coefficient values for relevant features?



```
> coef(model)
(Intercept)         cyl        disp          hp        drat          wt        qsec          vs          am        gear
12.41173489  0.61489995  0.02124426 -0.04523187  0.57176446 -5.73320752  0.97245746  0.87473702  2.34438911  0.51875794
       carb
 0.54833809
>
```

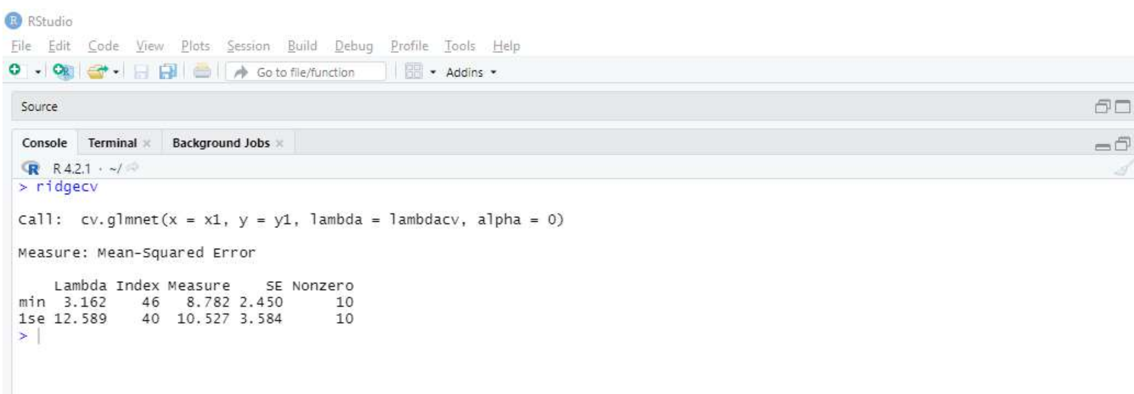
Only feature wt is seems to be relevant feature.

Perform a ridge regression using the glmnet package from CRAN, specifying a vector of 100 values of λ for tuning. Use cross-validation (via cv. glmnet) to determine the minimum value for λ - what do you obtain?



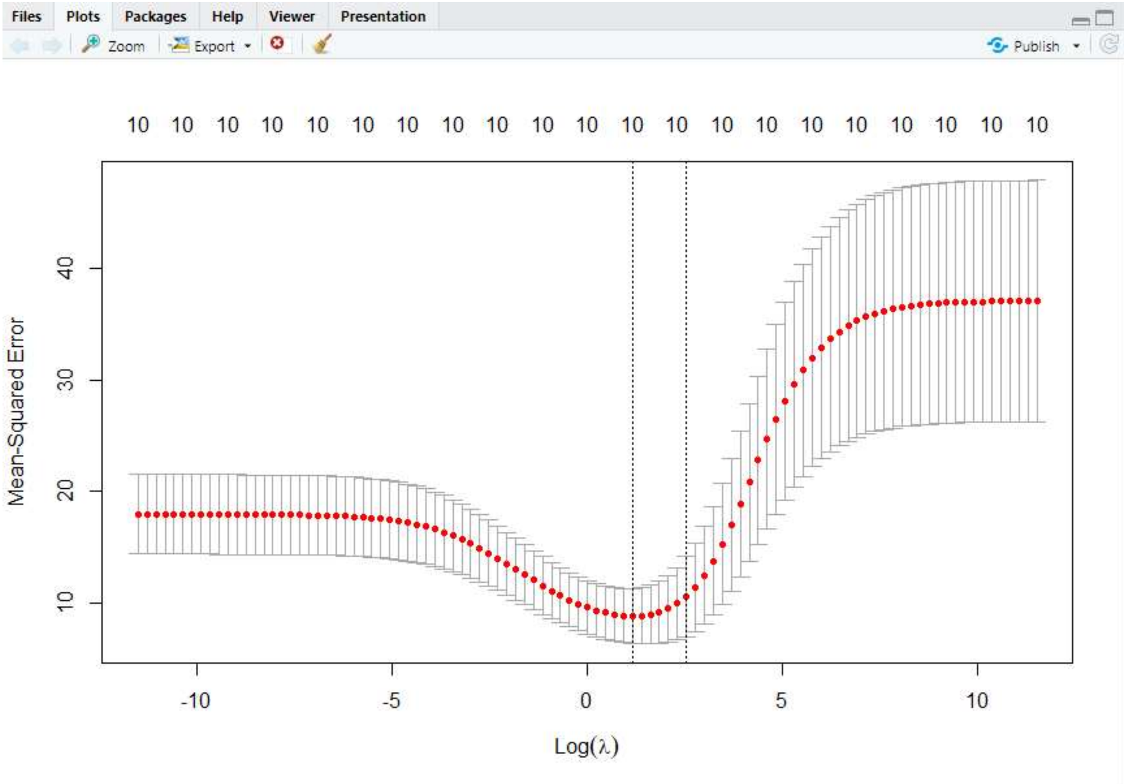
```
> library(glmnet)
> x1=model.matrix(mpg~.,train)[,-1]
> y1=train$mpg
> lambdacv=10^seq(5,-5,by=-.1)
> ridgecv=cv.glmnet(x1,y1,alpha=0,lambda = lambdacv)
```

```
> plot(ridgecv)
>
```



```
> ridgecv

Call:  cv.glmnet(x = x1, y = y1, lambda = lambdacv, alpha = 0)

Measure: Mean-Squared Error

    Lambda Index Measure    SE Nonzero
min  3.162    46   8.782 2.450      10
1se 12.589    40  10.527 3.584      10
>
```

Plot training MSE as a function of λ



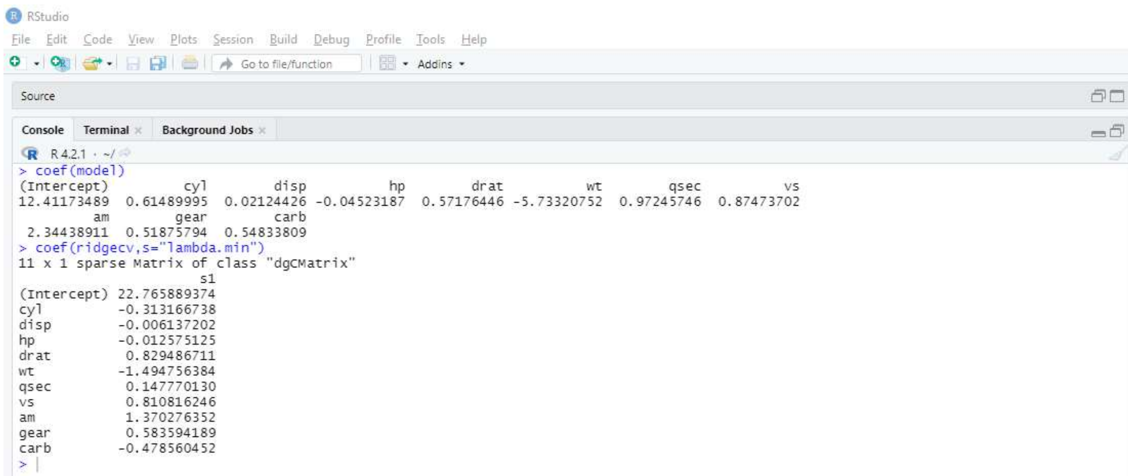
Minimum value for Lambda and regression model using glmnet

```
> min_lambda=ridgecv$lambda.min
> min_lambda
[1] 3.162278
> fit=glmnet(x1,y1,alpha=0,lambda = min_lambda)
> summary(fit)
          Length Class     Mode
a0         1     -none-    numeric
beta      10     dgCMatrix S4
df         1     -none-    numeric
dim        2     -none-    numeric
lambda     1     -none-    numeric
dev.ratio  1     -none-    numeric
nulldev    1     -none-    numeric
npasses    1     -none-    numeric
jerr       1     -none-    numeric
offset     1     -none-    logical
call       5     -none-    call
nobs       1     -none-    numeric
> coef(ridgecv,s="lambda.min")
11 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) 22.765889374
cyl         -0.313166738
disp        -0.006137202
hp          -0.012575125
drat         0.829486711
wt          -1.494756384
qsec         0.147770130
vs           0.810816246
am           1.370276352
gear         0.583594189
carb        -0.478560452
>
```

What is out-of-sample test set performance using predict? and how do the coefficients differ versus the regular linear model?

So, from the result of ridge regression, it shows MSE on test data reduce from 11.7182 to 3.741804

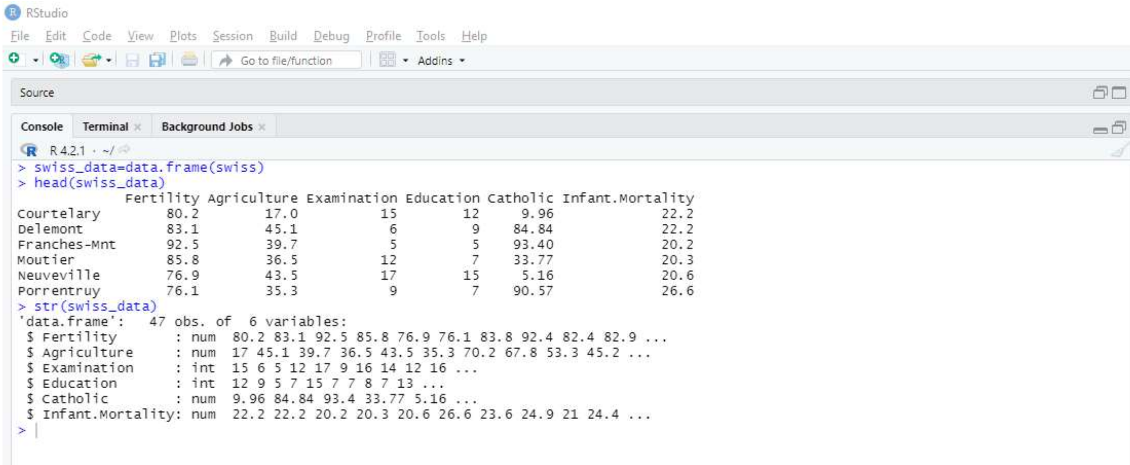
Has ridge regression performed shrinkage, variable selection, or both?



From above ridge Regression the coefficients and glm model coefficients results, we can see that ridge regression coefficients have shrunk and some of these values are close to zero compared to glm model coefficients. So, answer is yes ridge regression has performed shrinkage.

Problem 2.2

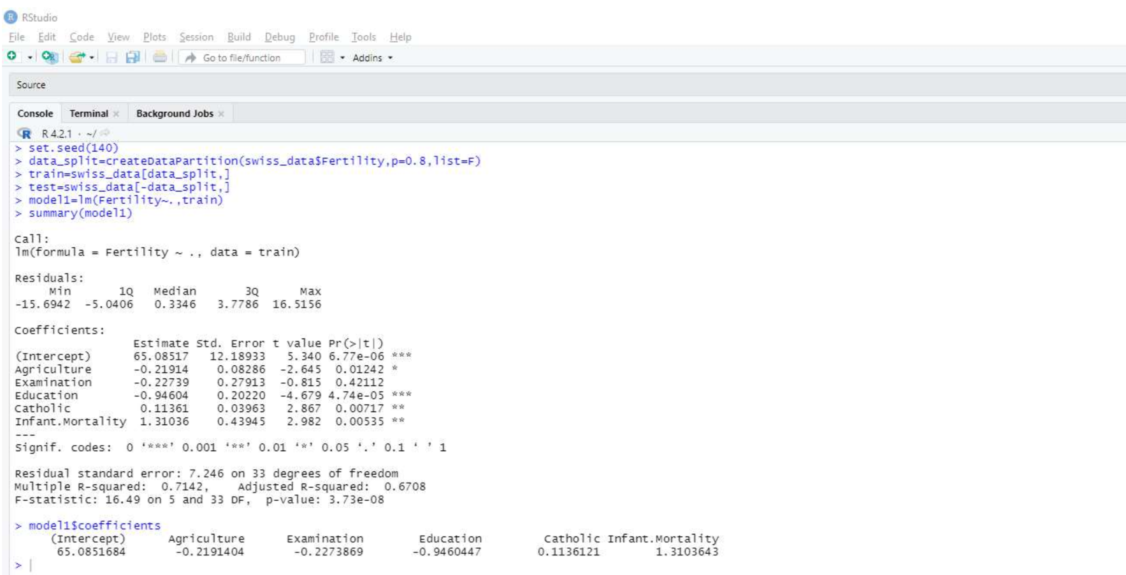
Load the Swiss sample dataset from the built-in datasets (data (Swiss)) into R using a dataframe.



```
> swiss_data=data.frame(swiss)
> head(swiss_data)
             Fertility Agriculture Examination Education Catholic Infant.Mortality
Courtelary        80.2        17.0          15        12     9.96             22.2
Delemont          83.1        45.1           6         9    84.84             22.2
Franches-Mnt      92.5        39.7           5         5    93.40             20.2
Moutier           85.8        36.5          12         7    33.77             20.3
Neuveville        76.9        43.5          17        15     5.16             20.6
Porrentruy        76.1        35.3           9         7    90.57             26.6
> str(swiss_data)
'data.frame':	47 obs. of  6 variables:
 $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
 $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
 $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
 $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
 $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
 $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
> |
```

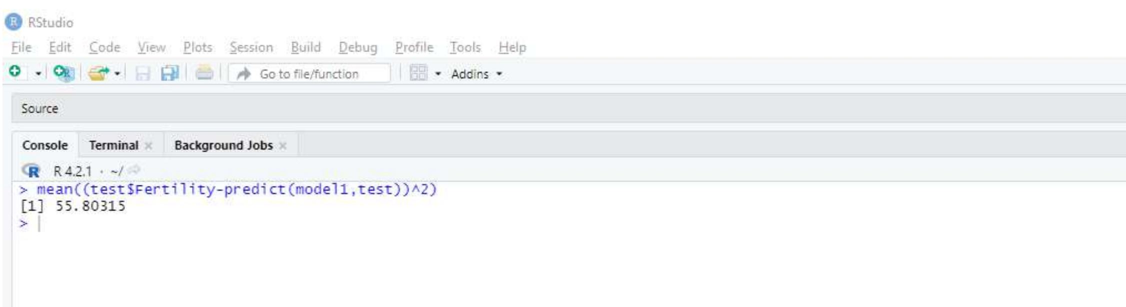
Perform a basic 80/20 test-train split on the data (you may use caret, the sample method, or manually) and fit a linear model with Fertility as the target response, and all other variables as predictors/features.

What features are selected as relevant based on resulting t-statistics? What are the associated coefficient values for relevant features?



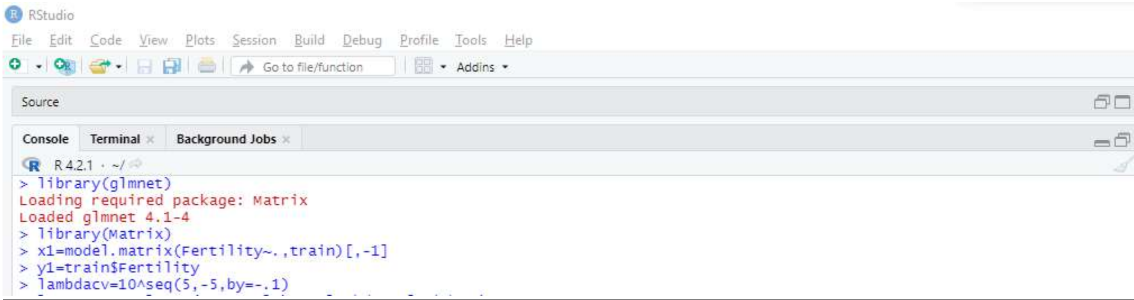
```
> set.seed(140)
> data_split=createDataPartition(swiss_data$Fertility,p=0.8,list=F)
> train=swiss_data[data_split,]
> test=swiss_data[-data_split,]
> model1=lm(Fertility~.,train)
> summary(model1)

Call:
lm(formula = Fertility ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.6942  -5.0406   0.3346   3.7786  16.5156 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      65.08517   12.18933   5.340 6.77e-06 ***
Agriculture      -0.21914    0.08286  -2.645  0.01242 *  
Examination      -0.22739    0.27913  -0.815  0.42112    
Education        -0.94604    0.20220  -4.679 4.74e-05 ***
Catholic          0.11361    0.03963   2.867  0.00717 ** 
Infant.Mortality  1.31036    0.43945   2.982  0.00535 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.246 on 33 degrees of freedom
Multiple R-squared:  0.7142,	Adjusted R-squared:  0.6708 
F-statistic: 16.49 on 5 and 33 DF,  p-value: 3.73e-08

> model1$coefficients
     (Intercept)      Agriculture      Examination        Education         Catholic Infant.Mortality 
      65.0851684       -0.2191404       -0.2273869       -0.9460447        0.1136121        1.3103643 
> |
```

Based on t-statistics except examination, features like agriculture, Education and infant Mortality are selected as relevant.

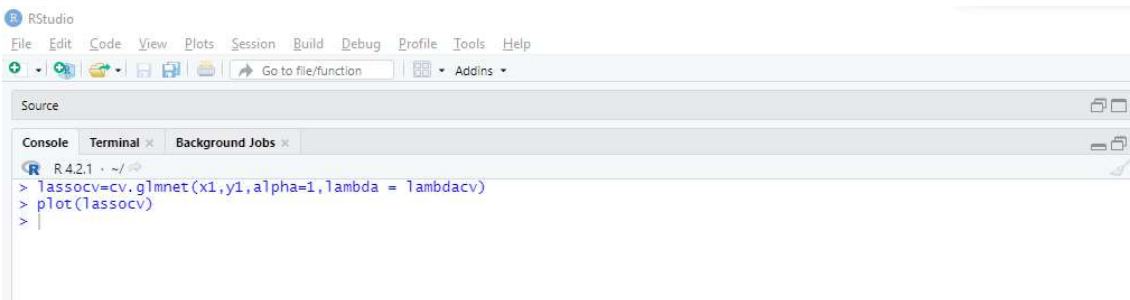


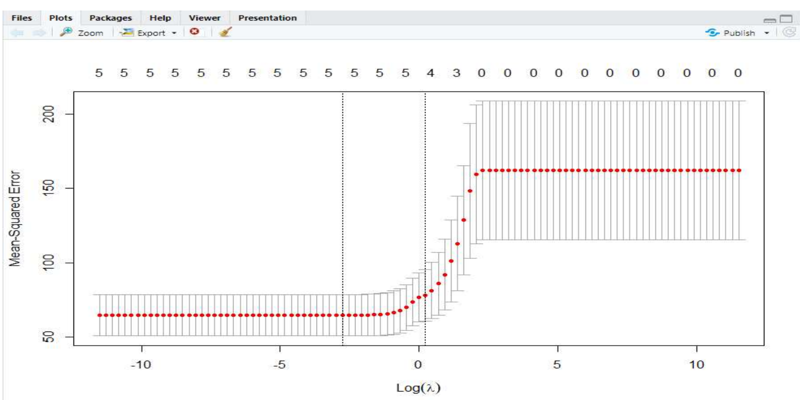
```
> mean((test$Fertility-predict(model1,test))^2)
[1] 55.80315
> |
```

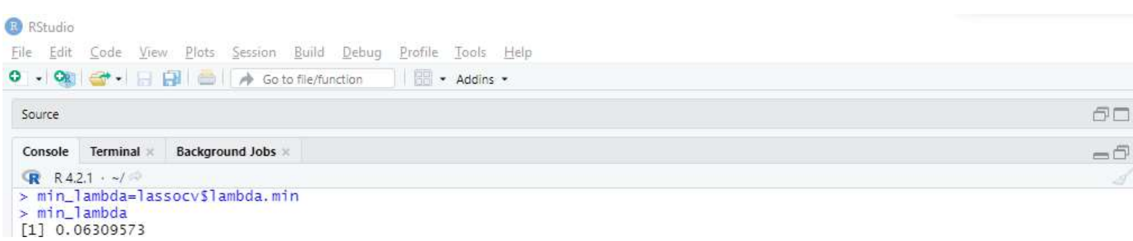
Perform a lasso regression using the glmnet package from CRAN, specifying a vector of 100 values of λ for tuning.



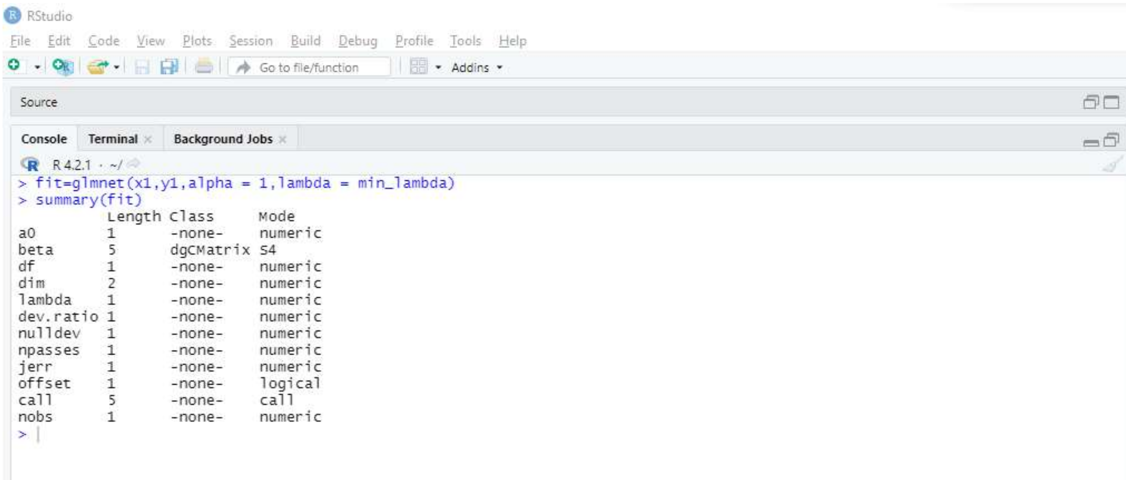
Use cross-validation (via cv. glmnet) to determine the minimum value for λ - what do you obtain? (Hint: You can use doMC in order to speed-up your cross-validation by specifying parallel=TRUE in your glmnet calls.). Plot training MSE as a function of λ (you may also use log λ)







Minimum value for Lambda is 0.0630

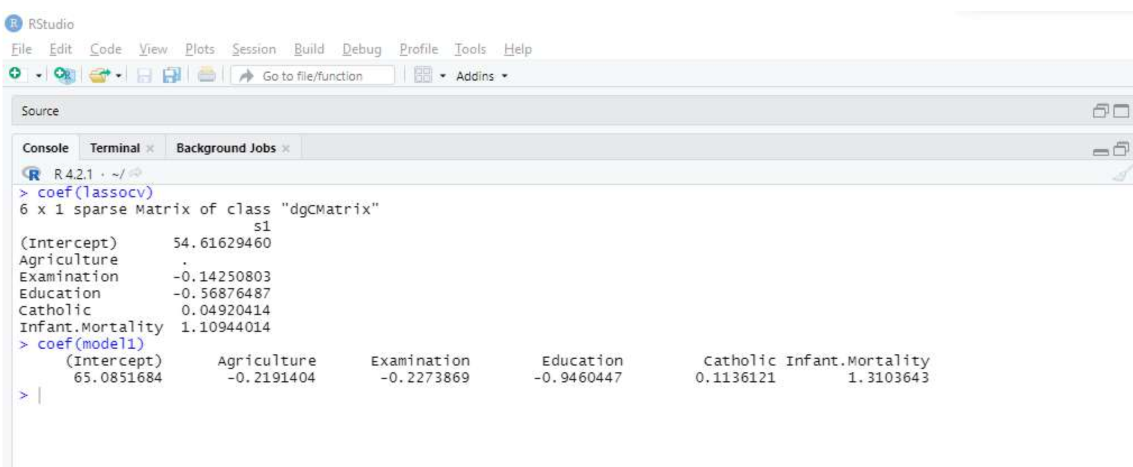


```
> fit=glmnet(x1,y1,alpha = 1,lambda = min_lambda)
> summary(fit)
          Length Class     Mode
a0        1      -none-    numeric
beta      5      dgCMatrix S4
df        1      -none-    numeric
dim       2      -none-    numeric
lambda    1      -none-    numeric
dev.ratio 1      -none-    numeric
nulldev   1      -none-    numeric
npasses   1      -none-    numeric
jerr      1      -none-    numeric
offset    1      -none-    logical
call      5      -none-    call
nobs      1      -none-    numeric
>
```

What is out-of-sample test set performance (using predict)?

```
> x1=model.matrix(Fertility~.,test)[,-1]
> predict_model=predict(fit,s=min_lambda,newx = x1,type = "response")
> mean((predict_model-test$Fertility)^2)
[1] 53.18605
>
```

How do the coefficients differ versus the regular linear model? Has lasso regression performed shrinkage, variable selection, or both?

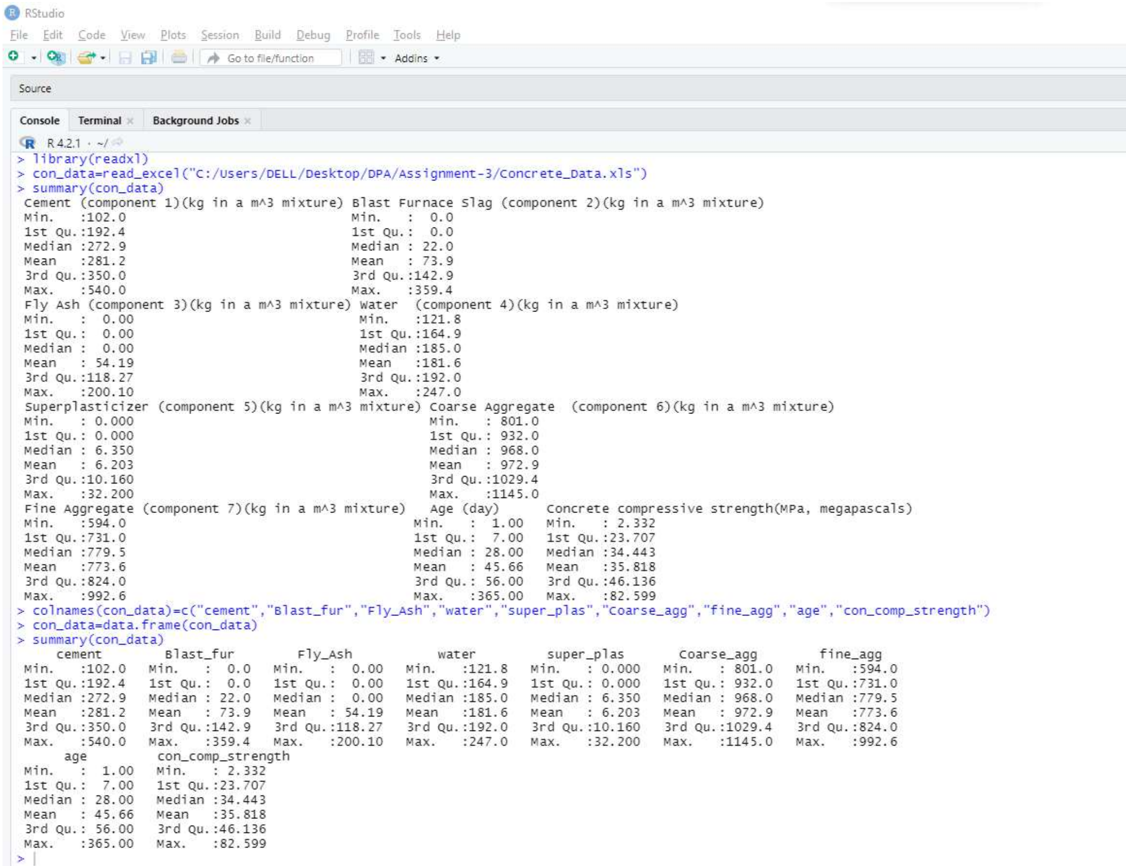


```
> coef(lassocv)
6 x 1 sparse Matrix of class "dgCMatrix"
                         s1
(Intercept)      54.61629460
Agriculture       .
Examination      -0.14250803
Education        -0.56876487
Catholic          0.04920414
Infant.Mortality  1.10944014
> coef(model1)
     (Intercept)      Agriculture      Examination        Education         Catholic Infant.Mortality
      65.0851684       -0.2191404       -0.2273869       -0.9460447        0.1136121        1.3103643
>
```

From the linear model coefficients and lasso regression coefficients, we can say that lasso regression has performed shrinkage.

Problem 2.3

Load the Concrete Compressive Strength sample dataset from the UCI Machine Learning Repository (Concrete Data.xls) into R using a dataframe



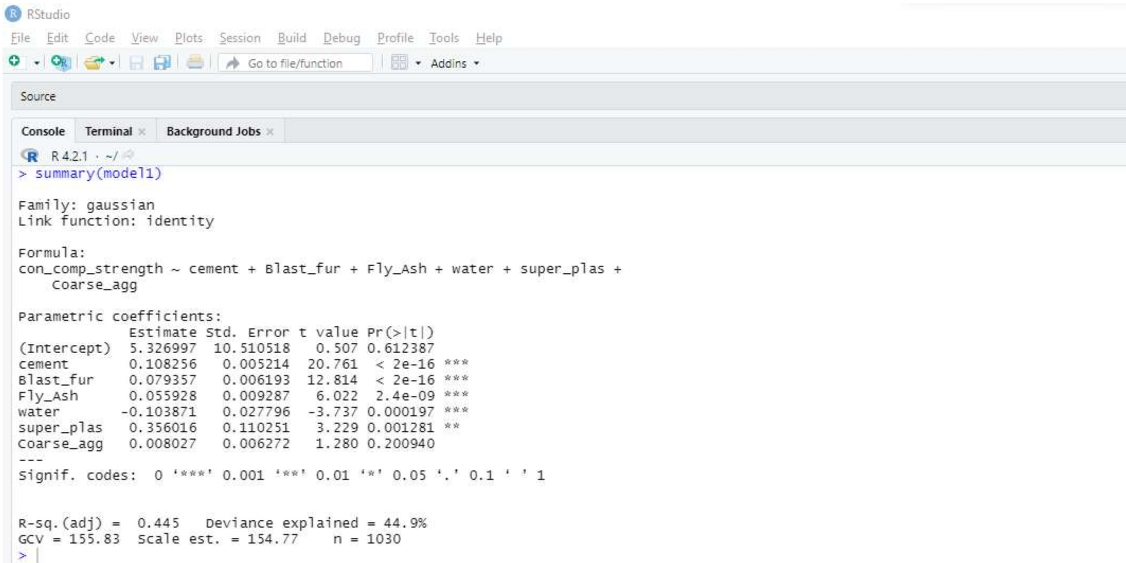
```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function   Addins
Source
Console  Terminal  Background Jobs
R 4.2.1 · ~/
> library(readxl)
> con_data=read_excel("C:/Users/DELL/Desktop/DPA/Assignment-3/Concrete_Data.xls")
> summary(con_data)
 Cement (component 1)(kg in a m^3 mixture) Blast Furnace Slag (component 2)(kg in a m^3 mixture)
 Min.   :102.0                             Min.   :  0.0
 1st Qu.:192.4                             1st Qu.:  0.0
 Median :272.9                             Median : 22.0
 Mean   :281.2                             Mean   : 73.9
 3rd Qu.:350.0                             3rd Qu.:142.9
 Max.   :540.0                             Max.   :359.4
 Fly Ash (component 3)(kg in a m^3 mixture) Water  (component 4)(kg in a m^3 mixture)
 Min.   :  0.00                             Min.   :121.8
 1st Qu.:  0.00                             1st Qu.:164.9
 Median :  0.00                             Median :185.0
 Mean   : 54.19                             Mean   :181.6
 3rd Qu.:118.27                             3rd Qu.:192.0
 Max.   :200.10                             Max.   :247.0
 Superplasticizer (component 5)(kg in a m^3 mixture) Coarse Aggregate  (component 6)(kg in a m^3 mixture)
 Min.   : 0.000                                      Min.   : 801.0
 1st Qu.: 0.000                                      1st Qu.: 932.0
 Median : 6.350                                      Median : 968.0
 Mean   : 6.203                                      Mean   : 972.9
 3rd Qu.:10.160                                      3rd Qu.:1029.4
 Max.   :32.200                                      Max.   :1145.0
 Fine Aggregate (component 7)(kg in a m^3 mixture)   Age (day)      Concrete compressive strength(MPa, megapascals)
 Min.   :594.0                                     Min.   :  1.00   Min.   : 2.332
 1st Qu.:731.0                                     1st Qu.:  7.00   1st Qu.:23.707
 Median :779.5                                     Median : 28.00   Median :34.443
 Mean   :773.6                                     Mean   : 45.66   Mean   :35.818
 3rd Qu.:824.0                                     3rd Qu.: 56.00   3rd Qu.:46.136
 Max.   :992.6                                     Max.   :365.00   Max.   :82.599
> colnames(con_data)=c("cement","Blast_fur","Fly_Ash","water","super_plas","Coarse_agg","fine_agg","age","con_comp_strength")
> con_data=data.frame(con_data)
> summary(con_data)
     cement         Blast_fur        Fly_Ash           water         super_plas       Coarse_agg        fine_agg
 Min.   :102.0   Min.   :  0.0   Min.   :  0.00   Min.   :121.8   Min.   : 0.000   Min.   : 801.0   Min.   :594.0
 1st Qu.:192.4   1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.:164.9   1st Qu.: 0.000   1st Qu.: 932.0   1st Qu.:731.0
 Median :272.9   Median : 22.0   Median :  0.00   Median :185.0   Median : 6.350   Median : 968.0   Median :779.5
 Mean   :281.2   Mean   : 73.9   Mean   : 54.19   Mean   :181.6   Mean   : 6.203   Mean   : 972.9   Mean   :773.6
 3rd Qu.:350.0   3rd Qu.:142.9   3rd Qu.:118.27   3rd Qu.:192.0   3rd Qu.:10.160   3rd Qu.:1029.4   3rd Qu.:824.0
 Max.   :540.0   Max.   :359.4   Max.   :200.10   Max.   :247.0   Max.   :32.200   Max.   :1145.0   Max.   :992.6
      age         con_comp_strength
 Min.   :  1.00   Min.   : 2.332
 1st Qu.:  7.00   1st Qu.:23.707
 Median : 28.00   Median :34.443
 Mean   : 45.66   Mean   :35.818
 3rd Qu.: 56.00   3rd Qu.:46.136
 Max.   :365.00   Max.   :82.599
>
```

Use the mgcv package to create a generalized additive model (via the gam function) to predict the Concrete Compressive Strength (CCS) as a non-linear function of the input components (C1-C6) - compare the R2 value for a GAM with linear terms as well as smoothed terms

We want only 1st six column.

GAM with Linear Terms

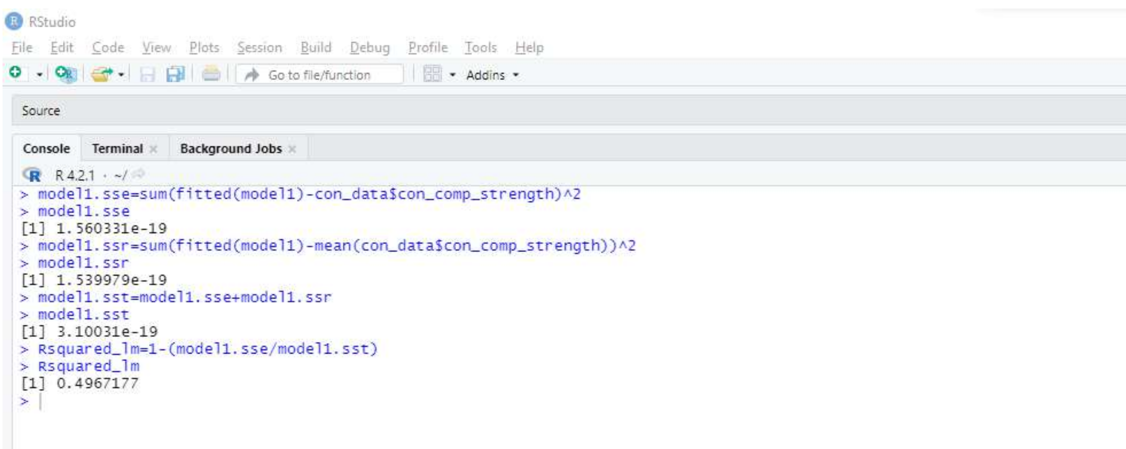
```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function   Addins
Source
Console  Terminal  Background Jobs
R 4.2.1 · ~/
> library(mgcv)
Loading required package: nlme
This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
> model1=gam(con_comp_strength~cement+Blast_fur+Fly_Ash+water+super_plas+Coarse_agg,data = con_data)
> model1

Family: gaussian
Link function: identity

Formula:
con_comp_strength ~ cement + Blast_fur + Fly_Ash + water + super_plas +
    Coarse_agg
Total model degrees of freedom 7

GCV score: 155.8332
>
```



```
> summary(model1)

Family: gaussian
Link function: identity

Formula:
con_comp_strength ~ cement + Blast_fur + Fly_Ash + water + super_plas +
    Coarse_agg

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.326997  10.510518   0.507 0.612387
cement       0.108256   0.005214  20.761  < 2e-16 ***
Blast_fur    0.079357   0.006193  12.814  < 2e-16 ***
Fly_Ash      0.055928   0.009287   6.022  2.4e-09 ***
water       -0.103871   0.027796  -3.737 0.000197 ***
super_plas   0.356016   0.110251   3.229 0.001281 **
Coarse_agg   0.008027   0.006272   1.280 0.200940
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R-sq.(adj) =  0.445   Deviance explained = 44.9%
GCV = 155.83  Scale est. = 154.77    n = 1030
>
```

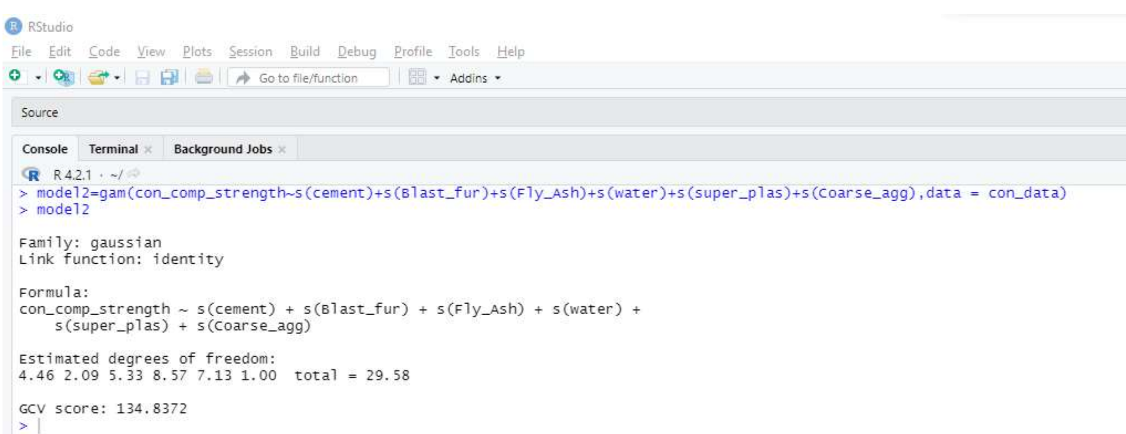
It shows we have statistical effects for cement, Blast furnace but not for coarse aggregate and the adjusted R-squared indicates a significant amount of the variance.

R^2 Calculation



```
> model1.sse=sum(fitted(model1)-con_data$con_comp_strength)^2
> model1.sse
[1] 1.560331e-19
> model1.ssr=sum(fitted(model1)-mean(con_data$con_comp_strength))^2
> model1.ssr
[1] 1.539979e-19
> model1.sst=model1.sse+model1.ssr
> model1.sst
[1] 3.10031e-19
> Rsquared_lm=1-(model1.sse/model1.sst)
> Rsquared_lm
[1] 0.4967177
>
```

GAM with Smoothed term



```
> model2=gam(con_comp_strength~s(cement)+s(Blast_fur)+s(Fly_Ash)+s(water)+s(super_plas)+s(Coarse_agg),data = con_data)
> model2

Family: gaussian
Link function: identity

Formula:
con_comp_strength ~ s(cement) + s(Blast_fur) + s(Fly_Ash) + s(water) +
    s(super_plas) + s(Coarse_agg)

Estimated degrees of freedom:
4.46 2.09 5.33 8.57 7.13 1.00  total = 29.58

GCV score: 134.8372
>
```

```
> summary(model2)

Family: gaussian
Link function: identity

Formula:
con_comp_strength ~ s(cement) + s(Blast_fur) + s(Fly_Ash) + s(water) +
    s(super_plas) + s(Coarse_agg)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.8178     0.3566   100.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                edf Ref.df      F  p-value
s(cement)     4.464  5.513 69.530  < 2e-16 ***
s(Blast_fur)  2.088  2.578 48.091  < 2e-16 ***
s(Fly_Ash)    5.332  6.404  1.784    0.101
s(water)      8.567  8.936 13.504  < 2e-16 ***
s(super_plas) 7.133  8.143  5.498 1.22e-06 ***
s(Coarse_agg) 1.000  1.000  0.018    0.892
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.531   Deviance explained = 54.4%
GCV = 134.84  Scale est. = 130.96    n = 1030
>
```

We can also note that this model accounts for much of the variance in concreate compressive strength, with an adjusted R-squared of .531. Means, it seems like the Cement is associated with concreate compressive strength.

R^2 calculation

```
> model2.sse=sum(fitted(model2)-con_data$con_comp_strength)^2
> model2.sse
[1] 3.017023e-16
> model2.ssr=sum(fitted(model2)-mean(con_data$con_comp_strength))^2
> model2.ssr
[1] 3.017921e-16
> model2.sst=model2.sse+model2.ssr
> model2.sst
[1] 6.034944e-16
> Rsquared_sm=1-(model2.sse/model2.sst)
> Rsquared_sm
[1] 0.5000744
>
```

Linear term vs smoothed term

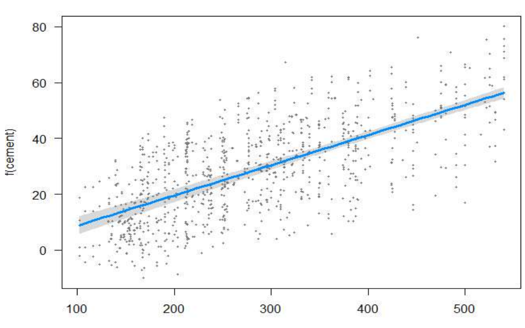
R^2 of the smoothed model is greater than the main R^2 value.

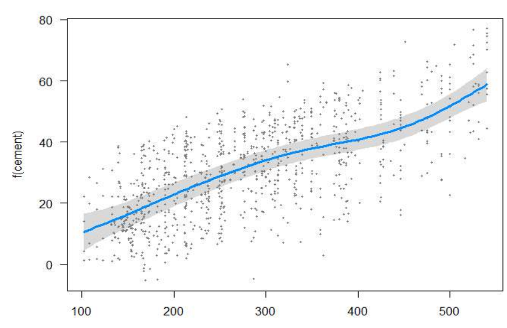
```
> anova(model1,model2,test = "Chisq")
Analysis of Deviance Table

Model 1: con_comp_strength ~ cement + Blast_fur + Fly_Ash + water + super_plas +
    Coarse_agg
Model 2: con_comp_strength ~ s(cement) + s(Blast_fur) + s(Fly_Ash) + s(water) +
    s(super_plas) + s(Coarse_agg)
  Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
1   1023.00     158334
2    996.43     131019 26.574    27315 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Results indicate that integrating nonlinear relationships of the covariates improves the model.

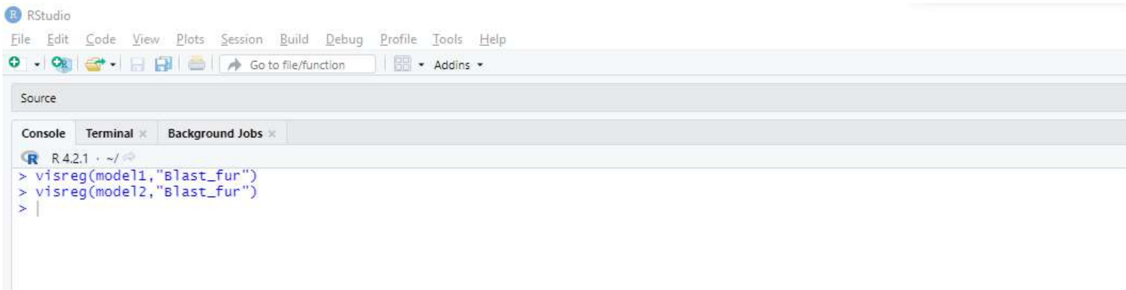
Visualize the regression using the visreg package, showing the fit as a function of each predictor with confidence intervals - comment on the quality of the fit at extreme values of the predictors.

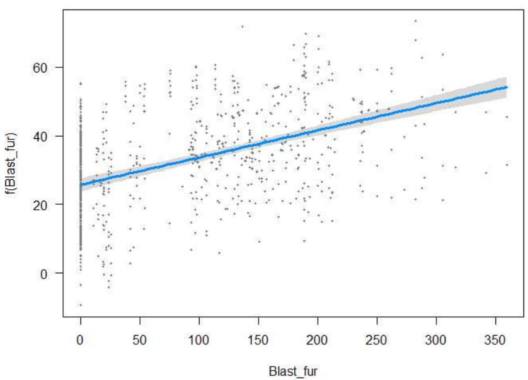
```
> install.packages("visreg")
Installing package into 'C:/Users/DELL/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/visreg_2.7.0.zip'
Content type 'application/zip' length 299852 bytes (292 KB)
downloaded 292 KB

package 'visreg' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\DELL\AppData\Local\Temp\RtmpgLReHB\downloaded_packages
> library(visreg)
> visreg(model1,'cement')
> visreg(model2,'cement')
>
```

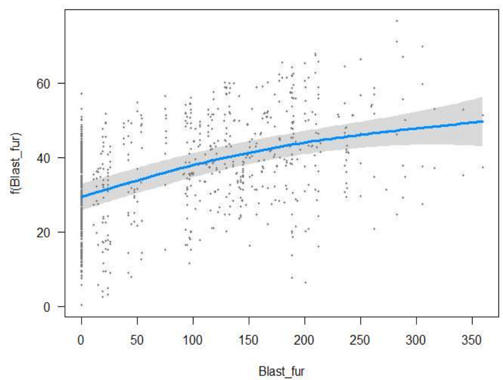


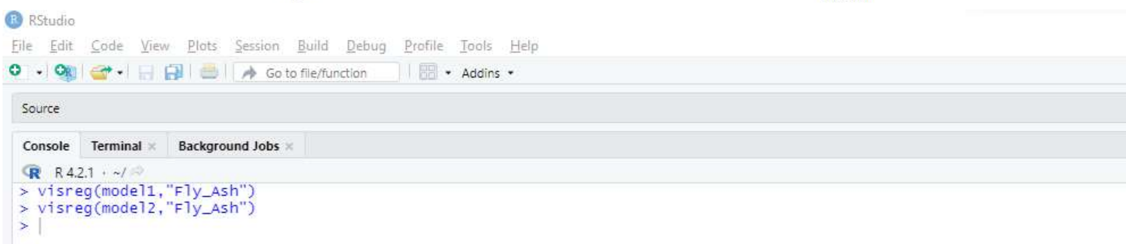


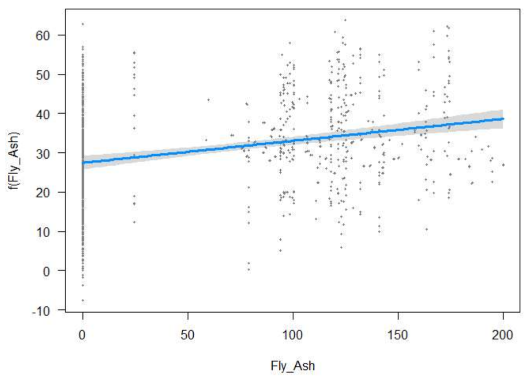
The above graph displays changes in concrete compressive strengths as a function of cement while all other variables remain fixed. In Graph blue line show expected value, gray colour denote confidence interval and dark grapy spots indicate partial residuals .

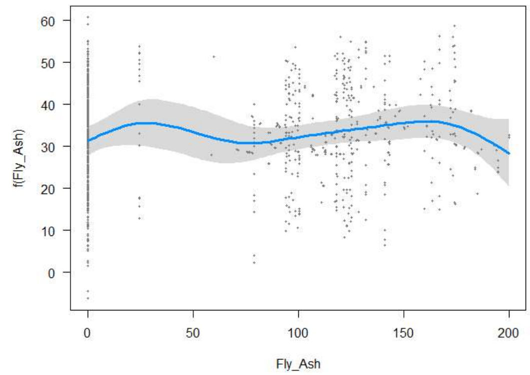


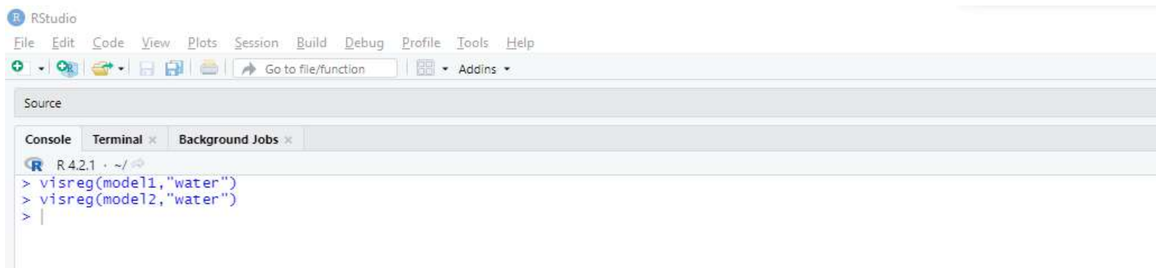


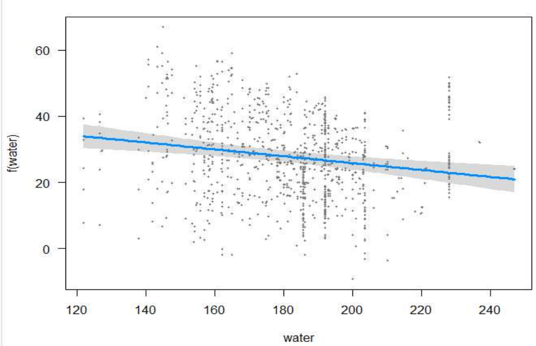


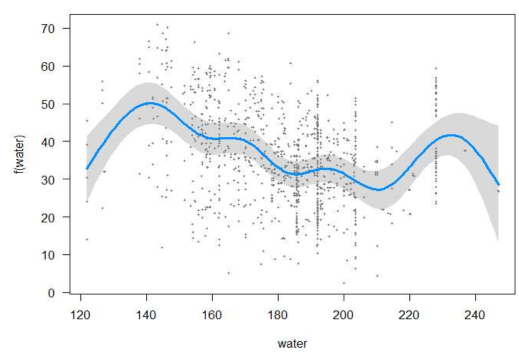


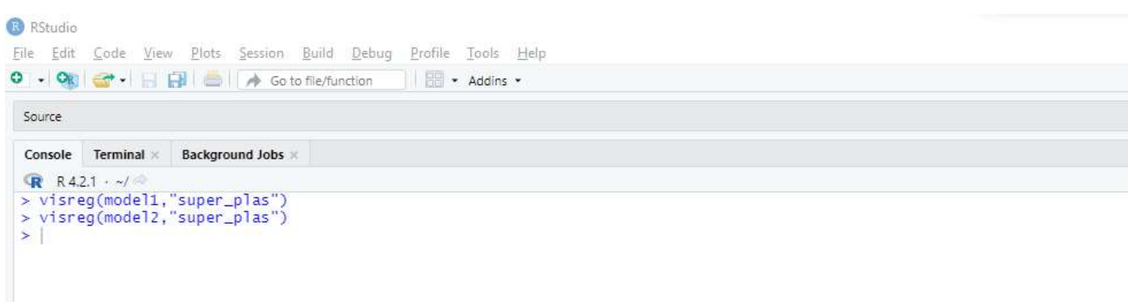

f(Fly_Ash)
Fly_Ash

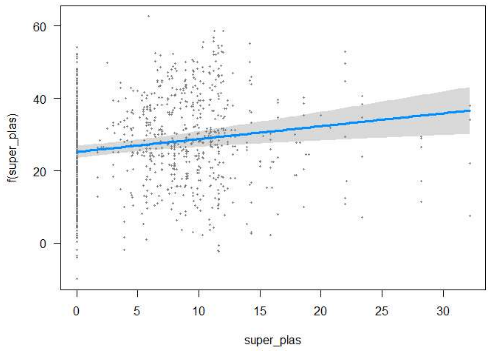

f(Fly_Ash)
Fly_Ash

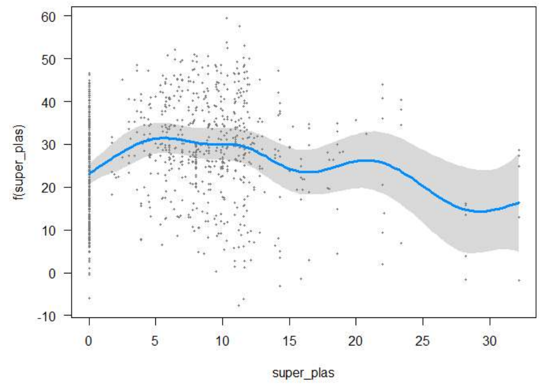

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console Terminal Background Jobs
R 4.2.1 · ~/
> visreg(model1,"water")
> visreg(model2,"water")
>

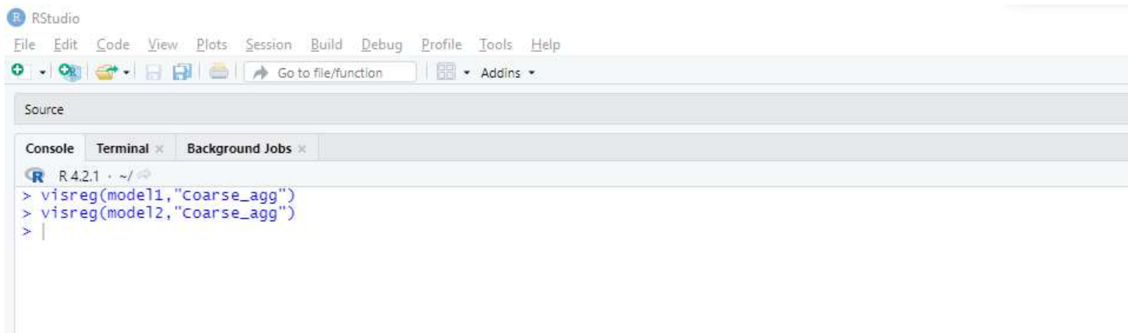

f(water)
water

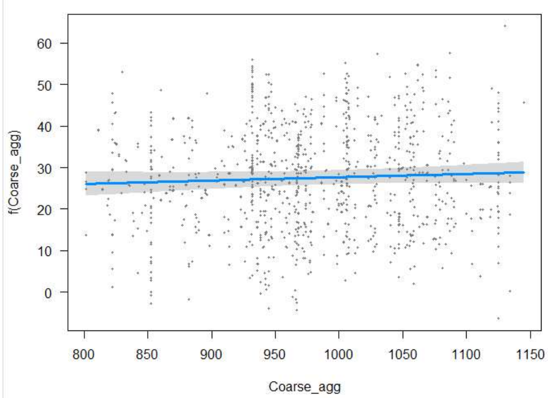

f(water)
water

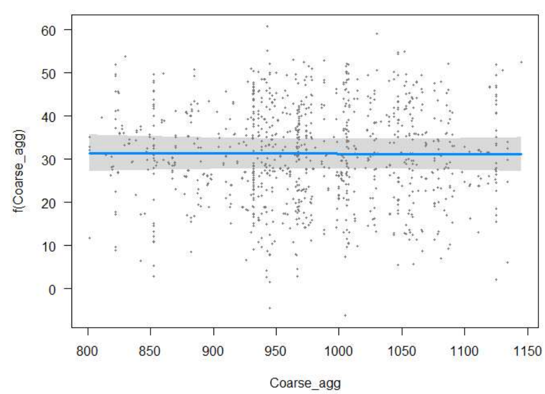

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console Terminal Background Jobs
R 4.2.1 · ~/
> visreg(model1,"super_plas")
> visreg(model2,"super_plas")
>


f(super_plas)
super_plas
0
5
10
15
20
25
30
60
40
20
0


f(super_plas)
super_plas
0
5
10
15
20
25
30
60
50
40
30
20
10
0
-10


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Background Jobs
R 4.2.1 · ~/
> visreg(model1,"Coarse_agg")
> visreg(model2,"Coarse_agg")
>


f(Coarse_agg)
Coarse_agg
800
850
900
950
1000
1050
1100
1150
60
50
40
30
20
10
0


f(Coarse_agg)
Coarse_agg
800
850
900
950
1000
1050
1100
1150
60
50
40
30
20
10
0