① . $y_i = \beta_0 + \beta_1 x_i + \beta_2 z_i + u_i$.

number of hot dog = $\beta_0 + \beta_1$(ticket) + β_2(maximum temp) + error

$\Rightarrow$ $x_i \Rightarrow$ Variable

$z_i \Rightarrow$ variable

$y_i \Rightarrow$ variable.

β_0 > Coefficient

β_1 > Coef-icient

β_2 > Coefficient

$\Rightarrow$ β_0 : no unit, just a number (intercept) !

$\Rightarrow$ β_0 is expectation mean of sell y when both x_i and z_i equal to zero. Since in real life people cannot purchase hot without pay for ticket, therefor. β_0 is a number.

β_1 : \$

$\Rightarrow$ β_1 describe the linear relationship between one more ticket sell with one more sell in the hot dog.

$\Rightarrow$ measure the effect of each predictor after taking account of the all other predictor in the model.

$\Rightarrow \beta_2$: degree

$\Rightarrow$ the linear relationship bettween the one more decree on the daily maximum temprery with one more hot dog sell.

③: $\beta_0 \Rightarrow$ positive

$\Rightarrow$ β_0 is the mean average of hot dog sell when tick and daily maximue temp equal to zero. However, hot dog sell cannot be negative and no ticket sell no people in the park therefore no hot dog sell.

$$Y = \beta_0 + \beta_1 X + \beta_2 Z_i + U_i.$$

$\therefore \Rightarrow$ when $X = 0$. $Y \geq 0$.

$\Rightarrow Y > 0$ (no negative sell)

$\therefore \beta_0 > 0$

β_1 : positive.

⇒ logitally, the more ticket sell, there will be more people in the park such they will be higher chance people will purchase hot dog.

β_2 : positive or unsure.

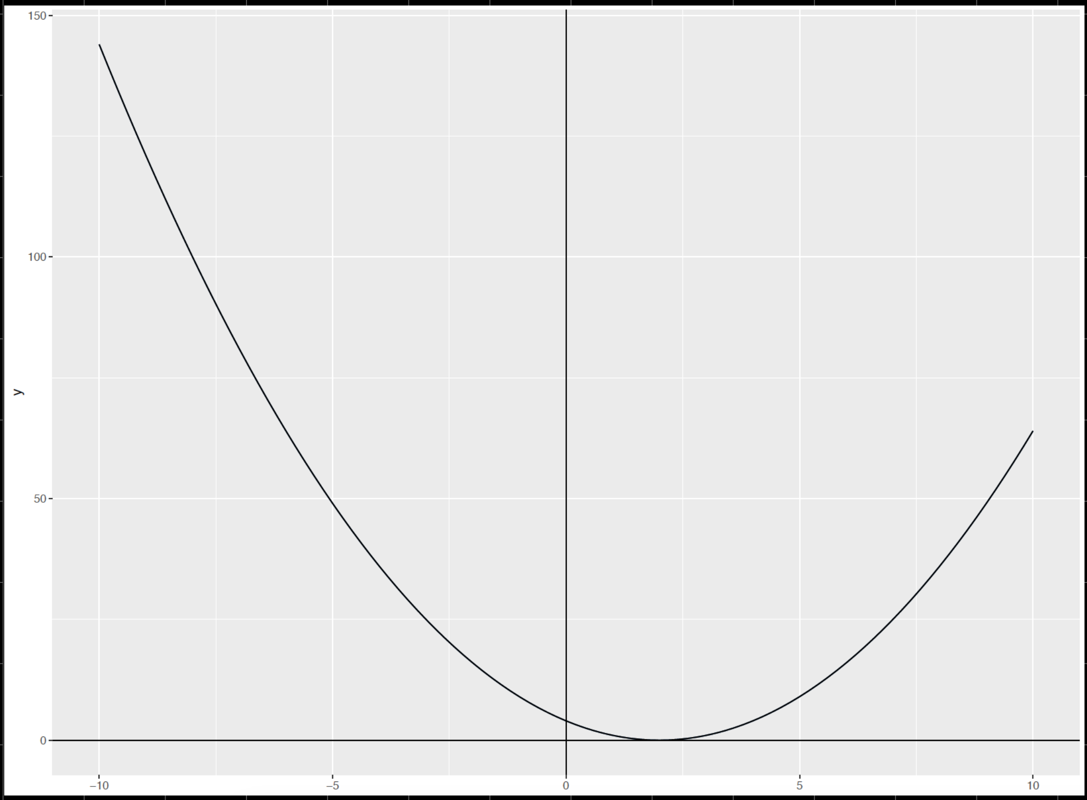
⇒ people or human will only go to the park within a certain tempure range. Such it nonlogic to calm people are more likly went to an amusement park in 40°C wealth than 20°C bring positive impact on hot dog sell. Therefore, it should be unsure.

D) Yes. ①: no perfect linear relationship in real life.

②. the excist of error term.

③: the excist, random even that could increase or decrease hot dog sell.

```
#2-1 plot (2-b)^2
eq = function(b){(2-b)^2}

#set the equation
x <- seq(-10,10, by=0.01)
#give the x a group of number from -100 to 100
y <- eq(x)
#y is equal to the number of that processed by the function
df <- data.frame(x,y)
#make it as data frame
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  +
  geom_hline(yintercept = 0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
#
#faster way
#eq = function(b){(2-b)^2}
#curve(eq, -10,10)

##saving the graph
pdf("pics/TieMa_homework1_Q2_1.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



$$\Rightarrow \sum_{i=1}^{n} \hat{u} = 0$$

$$\because \hat{u} = \hat{y} - \hat{\beta}_1 x$$

$$\therefore \frac{1}{n} \sum_{i=1}^{n} (\hat{y} - \hat{\beta}_1 x)^2 = 0.$$

$$\min_{\hat{\beta}_1} \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (\hat{y} - \hat{\beta}_1 x_1)^2 = 0.$$

FOC

$$\min_{\beta_1} \sum_{i=1}^{n} \cdot \frac{1}{n} \cdot 2 \cdot x_1 (\hat{y} - \hat{\beta}_1 x_1) = 0.$$

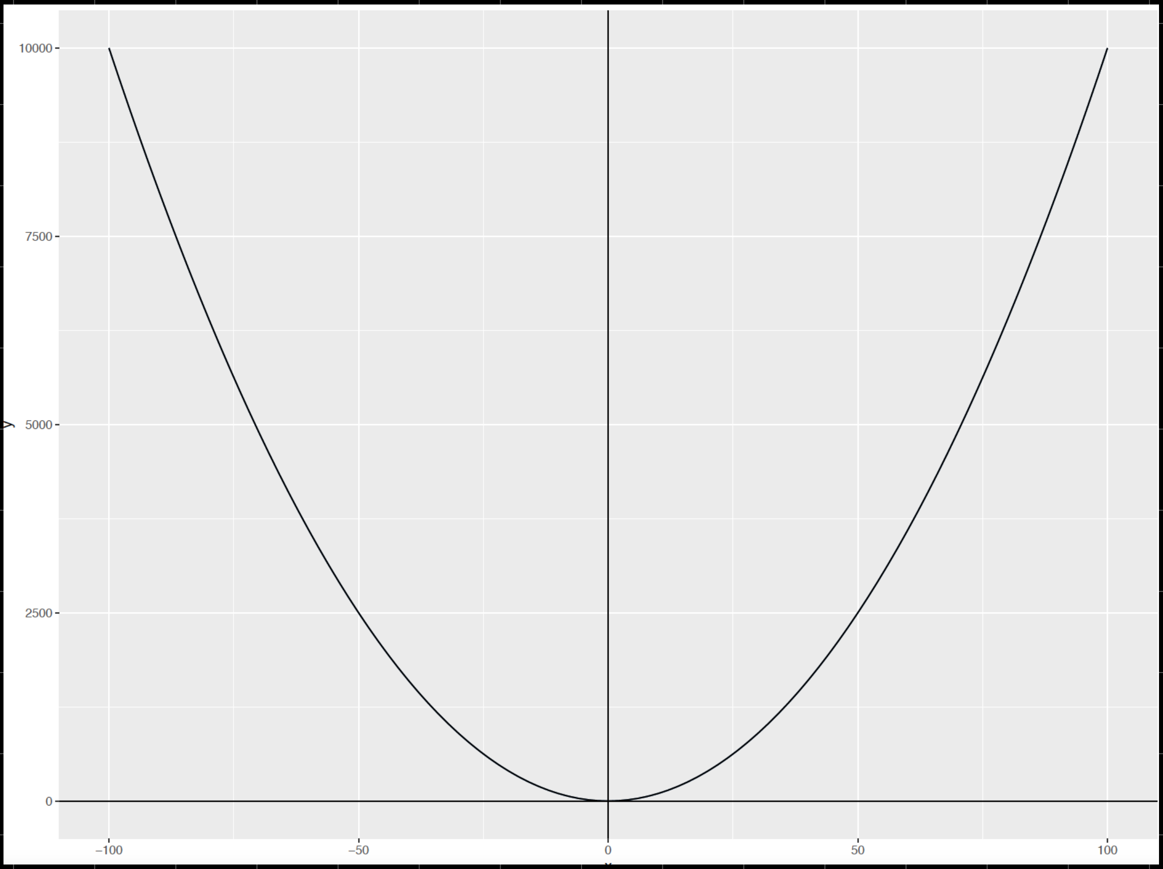
$$\min_{\beta_1} \sum_{i=1}^{n} \cdot \frac{1}{n} \cdot 2 \cdot x_1 (\hat{y} - \hat{\beta}_1 x_1) = 0.$$

$$\because n = 1 \quad y_1 = 2 \quad x_1 = 1$$

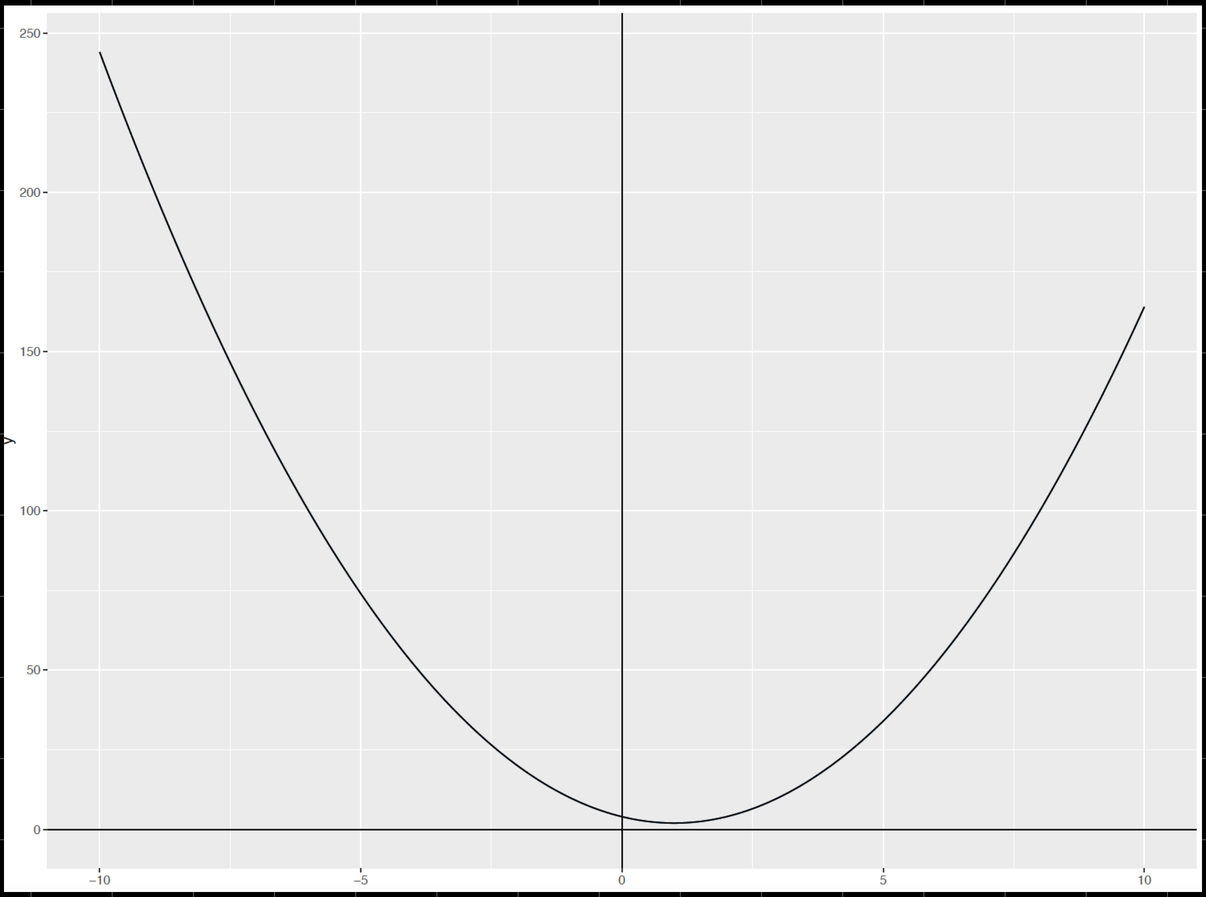
$$2 \cdot (2 - \hat{\beta}_1) = 0$$

$$\hat{\beta}_{OLS} = 2.$$

```
#2-c
eq = function(b){b^2}
x <- seq(-100,100, by=0.01)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q2_2.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()

rm(list = ls())
```



```
#2-d
#plot the graphy of (2-b)^2 + b^2


eq = function(b){(2-b)^2 + b^2}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)

##saving the graph
pdf("pics/TieMa_homework1_Q2_3.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



Q2-e.

```
rm(list = ls())

eq = function(b){(2-b)^2 + b^2}
ans <- optimize(eq, interval = c(-10,10))

x_min = ans$minimum
y_min = ans$objective

print(y_min)
#[1] 2
print(x_min)
#[1] 1
```

$$(2-B)^2 + B^2$$

$$\Rightarrow 4 - 4B + B^2 + B^2 = 0$$

$$4 - 4B + 2B^2$$

$$\min_{B} \; 4 - 4B + 2B^2 = 0.$$

FOC

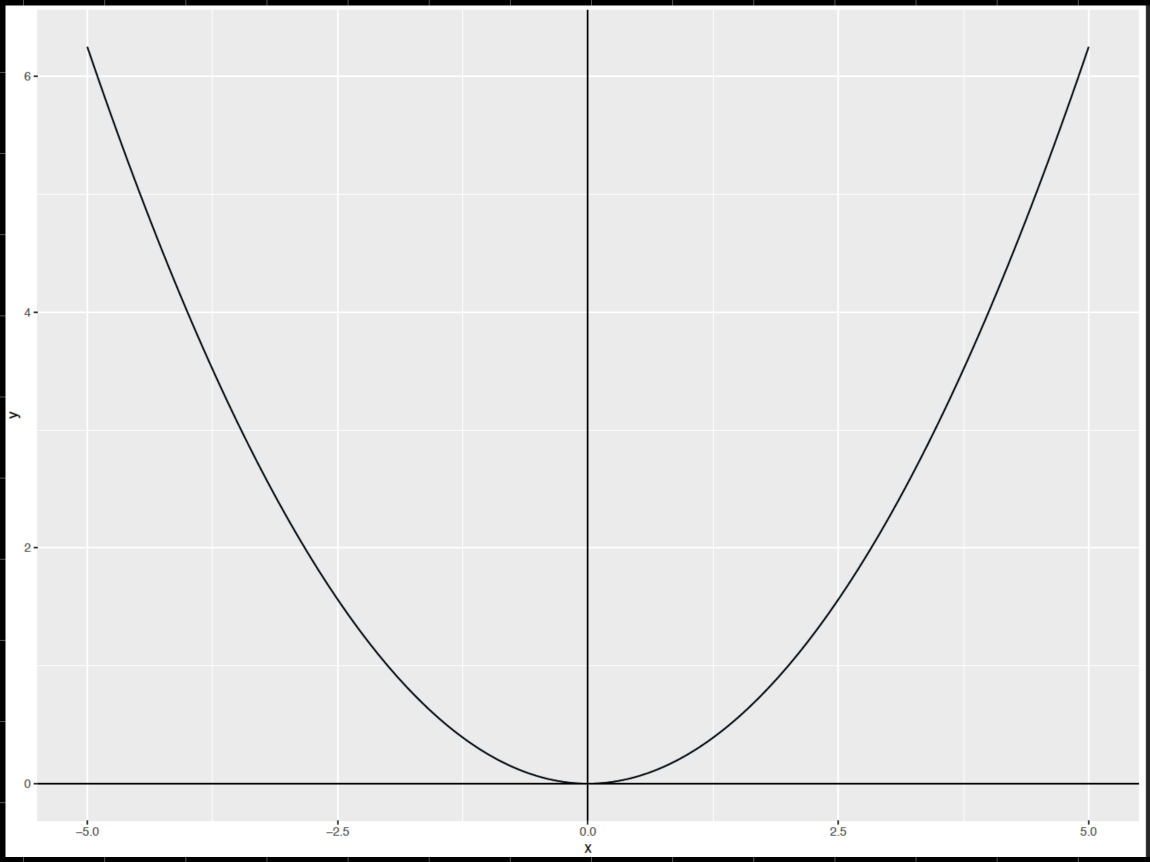
$$-4 + 4B = 0.$$

$$\therefore \; B = 1$$

$$\therefore \; \hat{B}_{ridge} = 1$$

2 - f - repeat c

```
eq = function(b){0.5*(b^2)}
x <- seq(-5,5, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q2_f.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#find the value of lambda beta hat
rm(list = ls())
eq = function(b){0.5*(b^2)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 0
print(x_min)
#[1] 0
```

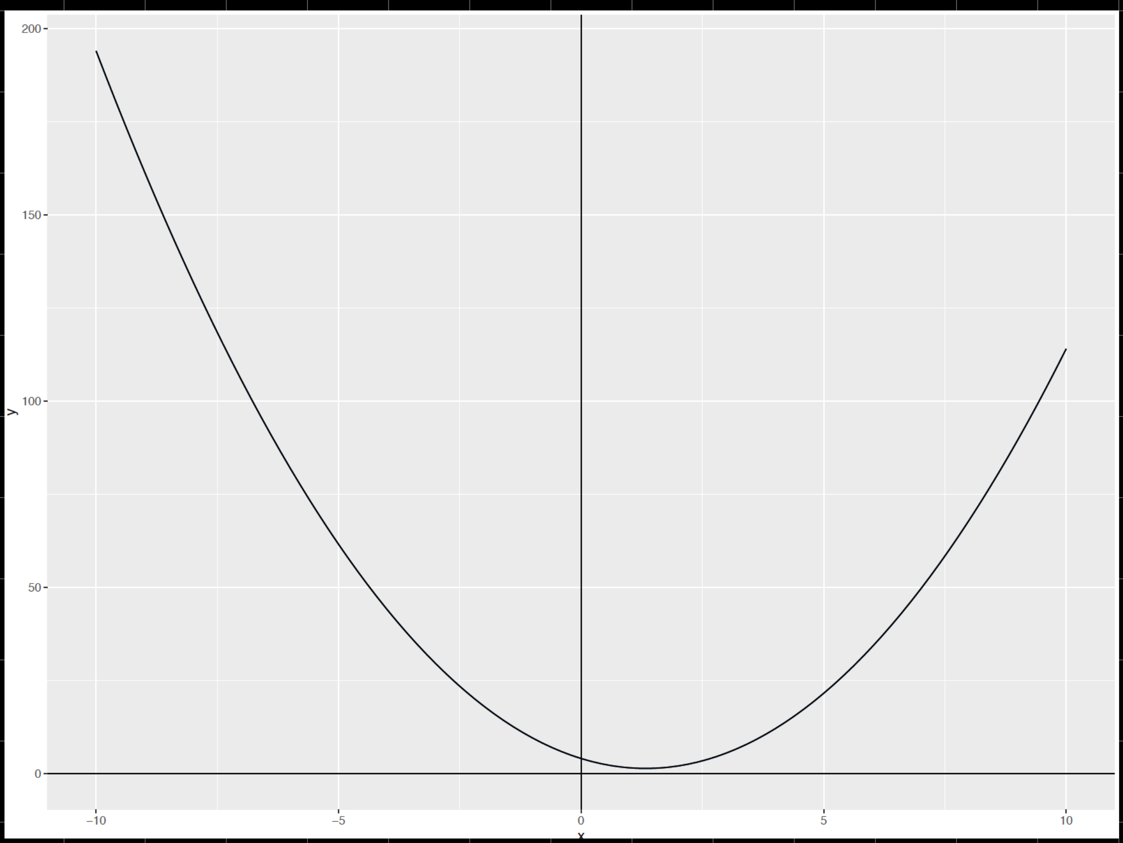
$\Rightarrow \quad 0.5\beta^2$

$\min_{\beta} \quad 0.5\beta^2 = 0$

$b = 0$

$\hat{\beta}_{ridge} = 0.$

```
#2-f-2 repeat the (d) and find the value of lambda beta hat

#first, let's plot the graphy
#(2-b)^2 + 0.5*b^2

eq = function(b){(2-b)^2 + 0.5*(b^2)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept = 
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)

##saving the graph
pdf("pics/TieMa_homework1_Q2_f_repeat_question_d_plot_the graphy.pdf", height = 9, 
width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept = 
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#find the value of lambda beta hat
rm(list = ls())
eq = function(b){(2-b)^2 + 0.5*(b^2)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 1.333333
print(x_min)
#[1] 1.333333
```

$$(2-\beta)^2 + (0.5\beta)^2.$$

ff

$$\min_{\beta} (2-\beta)^2 + (0.5\beta)^2 = 0$$

FOC

$$-2(2-\beta) + 2 \cdot 0.5 \cdot \beta = 0.$$

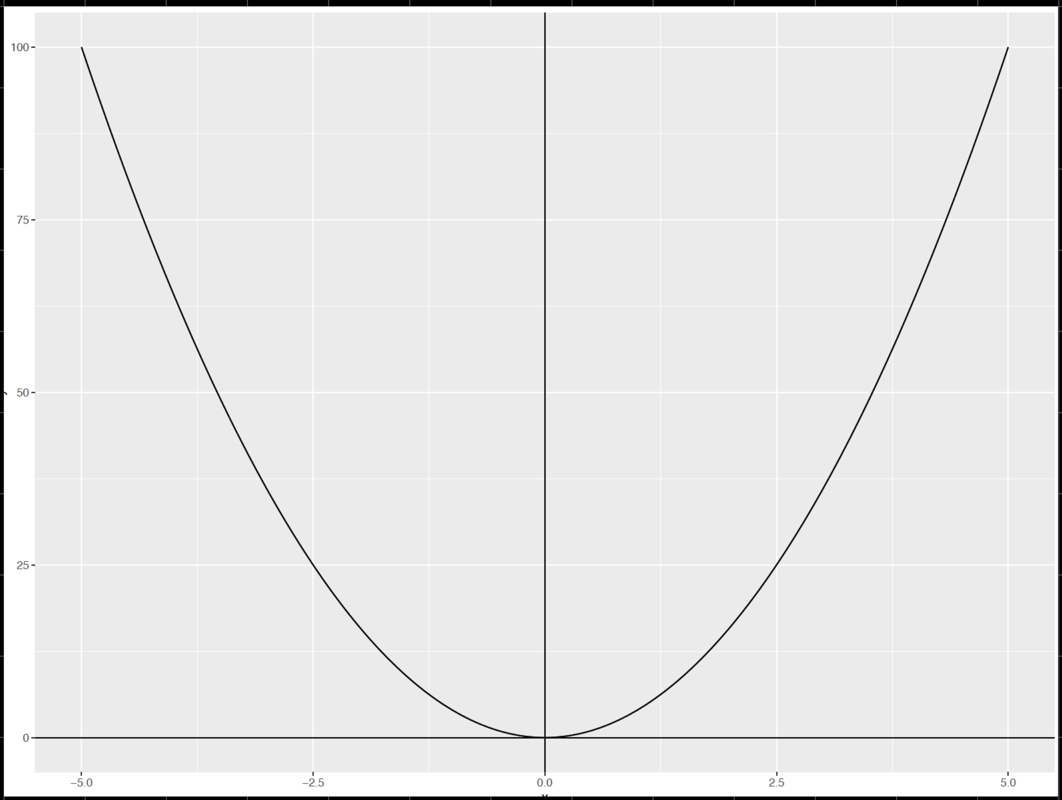
$$-4 + 2\beta + \beta = 0.$$

$$-4 + 3\beta = 0.$$

$$\therefore \beta = \frac{4}{3} = 1.\overline{33}$$

$$\therefore \hat{\beta}_{ridge} = \frac{4}{3}$$

```
#2-g-1 lambda ridge = 4 and repeat question c
rm(list = ls())

eq = function(b){4*(b^2)}
x <- seq(-5,5, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
plot(df)

##saving the graph
pdf("pics/TieMa_homework1_Q2_g_1.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#find the value of lambda beta hat
rm(list = ls())
eq = function(b){4*(b^2)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 0
print(x_min)
#[1] 0
```

$\hat{\beta}_{ridge} = 0.$

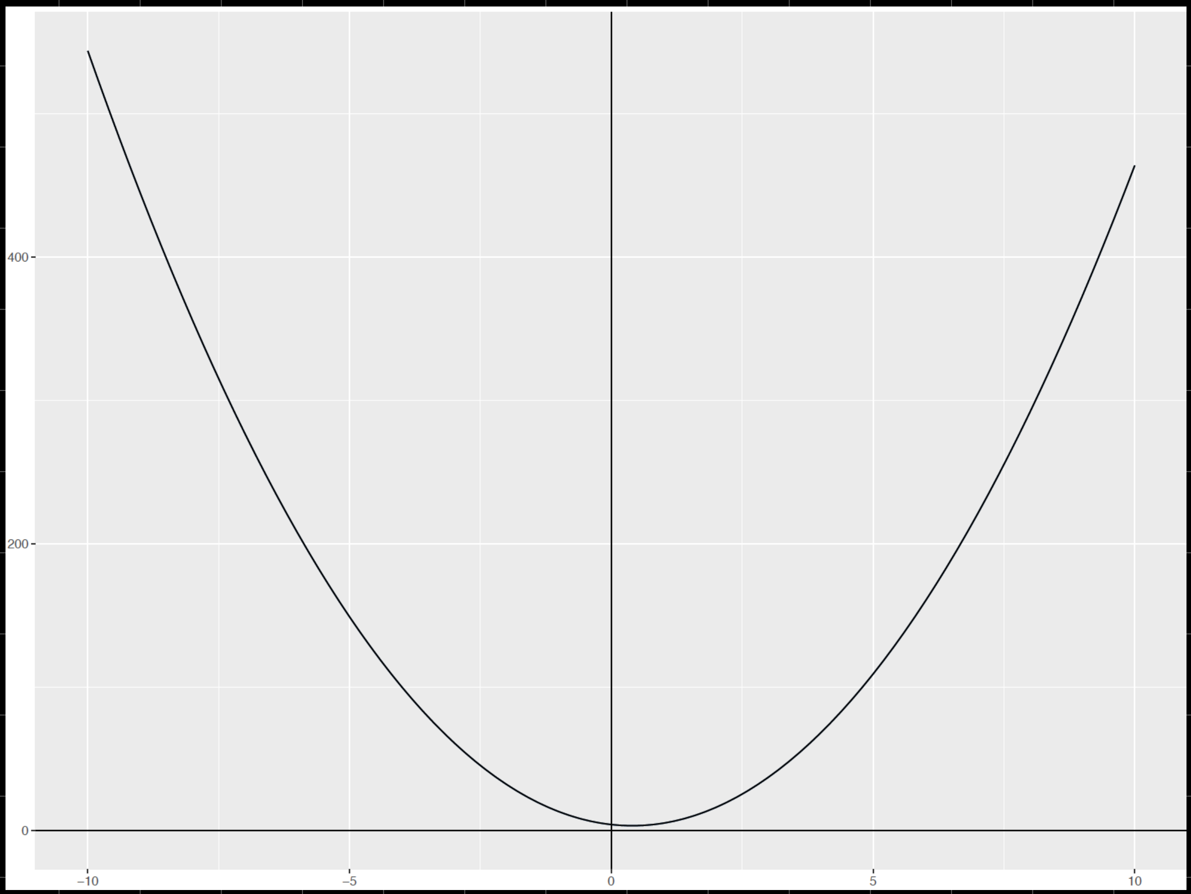
$\Rightarrow 4\beta^2$

$\min 4\beta^2 = 0$

$8\beta = 0$

$\hat{\beta}_{ridge} = 0$

```
rm(list = ls())
#2-g-2- lambda ridge = 4 and redue the question d
#first, let's plot the graphy
#(2-b)^2 + 4*b^2

eq = function(b){(2-b)^2 + 4*(b^2)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q2_g_2.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#find the value of lambda beta hat
rm(list = ls())
eq = function(b){(2-b)^2 + 4*(b^2)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 3.2
print(x_min)
#[1] 0.4
```

$$(2-\beta)^2 + 4\beta^2$$

$$\min_{\beta} (2-\beta)^2 + 4\beta^2 = 0$$

FOC

$$-2(2-b) + 8\beta = 0$$

$$-4 + 2\beta + 8\beta = 0$$

$$10\beta = 4$$

$$\beta = \frac{4}{10} = 0.4$$

$$\hat{\beta}_{ridge} = 0.4$$

```
#3-a

eq = function(b){(2-b)^2}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_a.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()

rm(list = ls())
eq = function(b){(2-b)^2}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 0
print(x_min)
#[1] 2
```

$\Rightarrow \sum_{i=1}^{n} \hat{u} = 0$

$\because \hat{u} = \hat{y} - \hat{\beta}_1 x$

$\therefore \frac{1}{n} \sum_{i=1}^{n} (\hat{y} - \hat{\beta}_1 x)^2 = 0.$

$\min_{\hat{\beta}_1} \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (\hat{y} - \hat{\beta}_1 x_1)^2 = 0.$

FOC

$\min_{\beta_1} \sum_{i=1}^{n} \cdot \frac{1}{n} \cdot 2 \cdot x_1 (\hat{y} - \hat{\beta}_1 x_1) = 0.$

$\min_{\beta_1} \sum_{i=1}^{n} \cdot \frac{1}{n} \cdot 2 \cdot x_1 (\hat{y} - \hat{\beta}_1 x_1) = 0.$

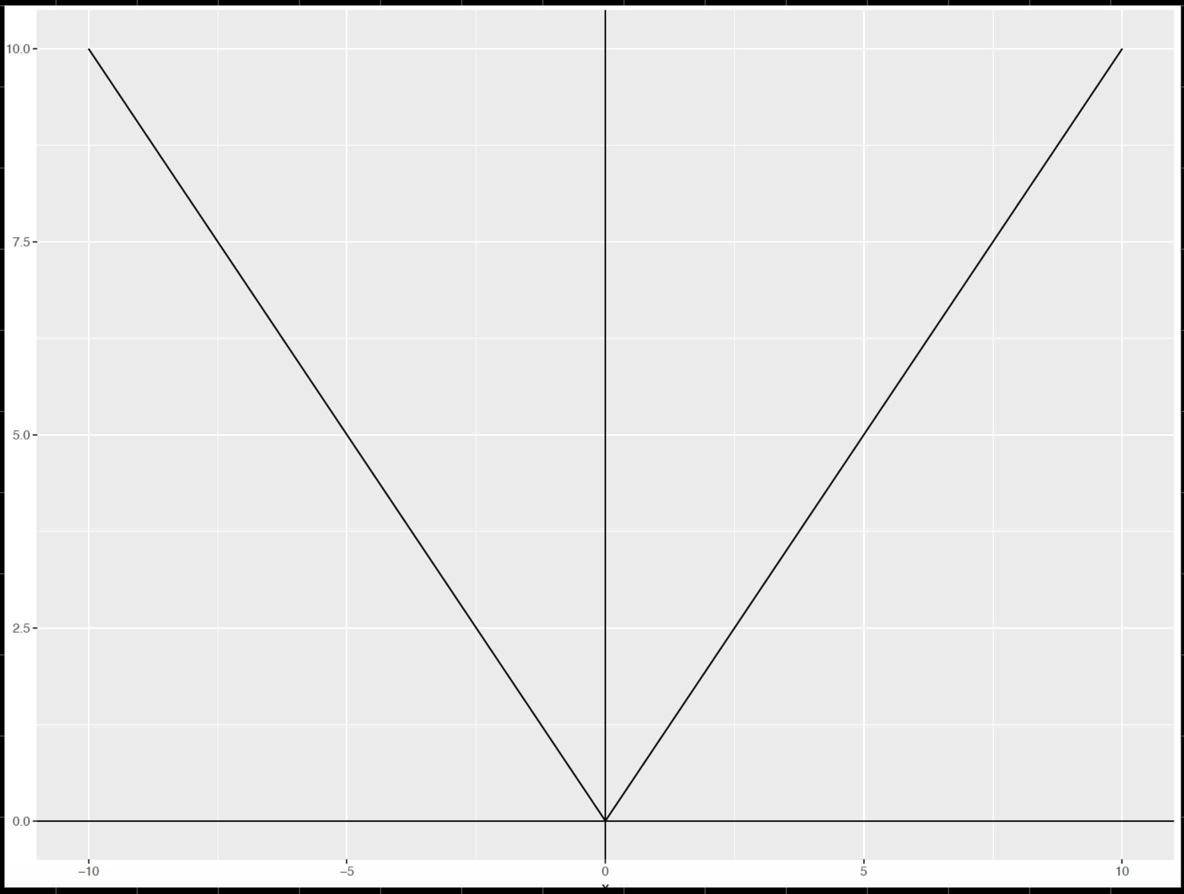
$\because n = 1 \quad y_1 = 2 \quad x_1 = 1$

$2 \cdot (2 - \hat{\beta}_1) = 0$

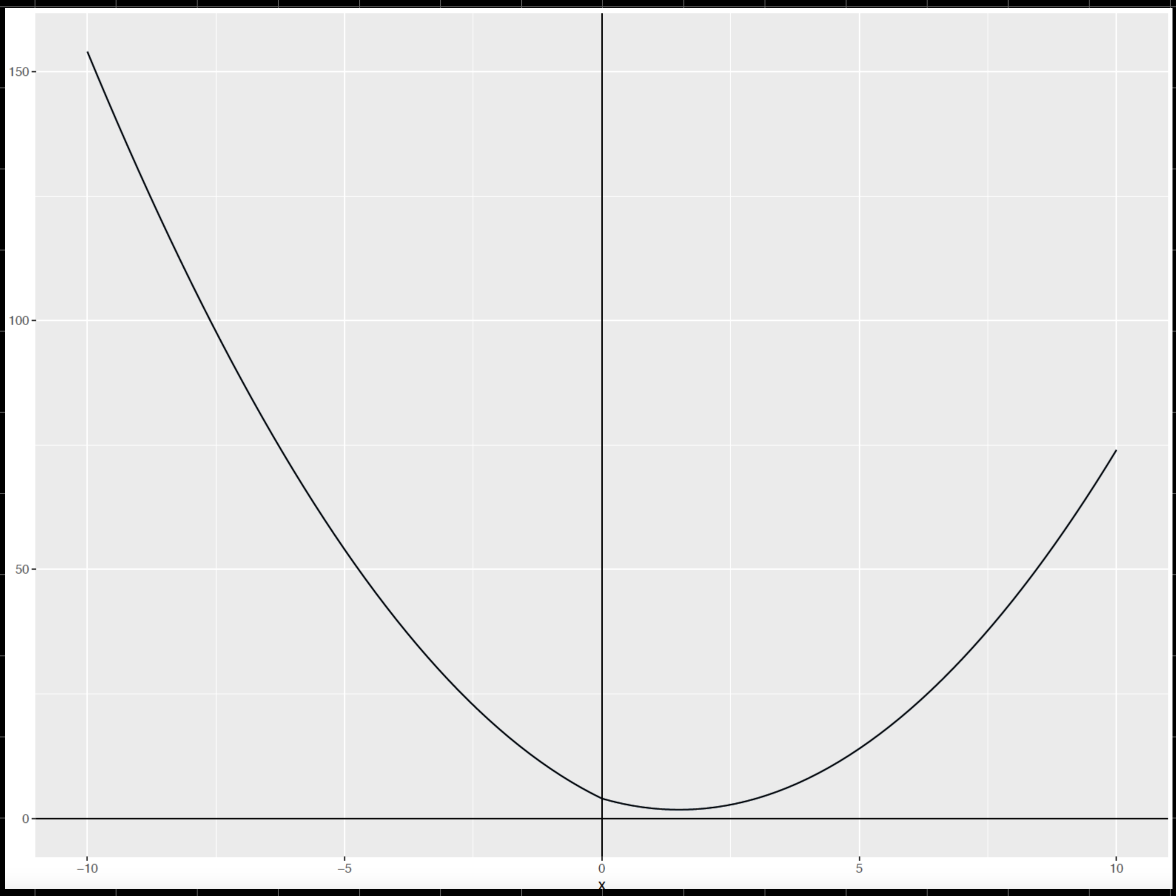
$\hat{\beta}_{OLS} = 2.$

3c

```
eq = function(b){abs(b)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept = 0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_C.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept = 0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#3-d
rm(list = ls())
eq = function(b){(2-b)^2 + abs(b)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_d.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
#3-e
rm(list = ls())
eq = function(b){(2-b)^2 + abs(b)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 1.75
print(x_min)
#[1] 1.5
```

$$(2-\beta)^2 + |\beta|$$

$$\min_{\beta} (2-\beta)^2 + |\beta| = 0.$$

FOC

$$-2(2-\beta) + \frac{\beta}{|\beta|} = 0.$$

$$-4 + 2\beta + \frac{\beta}{|\beta|} = 0.$$

if $\beta > 0$.

$$-4 + 2\beta + 1 = 0.$$

$$2\beta = 3$$

$$\underline{\beta = \frac{3}{2} = 1.5}$$

smaller.

if $\beta < 0$.

$$-4 + 2\beta - 1 = 0$$

$$2\beta = 5$$

$$\beta = \frac{5}{2} = 2.5$$

$$\therefore \hat{\beta}_{lasso} = 1.5$$

```
#3-f-1 labamta = 0.5 repeat c and d

#c
eq = function(b){0.5*abs(b)}
x <- seq(-5,5, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_f_1.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```

```
rm(list = ls())
eq = function(b){0.5*abs(b)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 6.661338e-16
print(x_min)
#[1] 1.332268e-15
```

≈ 0

$\approx 0.$

$$0.5 \cdot \mathrm{abs}(\beta)$$

$$\min_{\beta} 0.5\, \mathrm{abs}(\beta) = 0$$

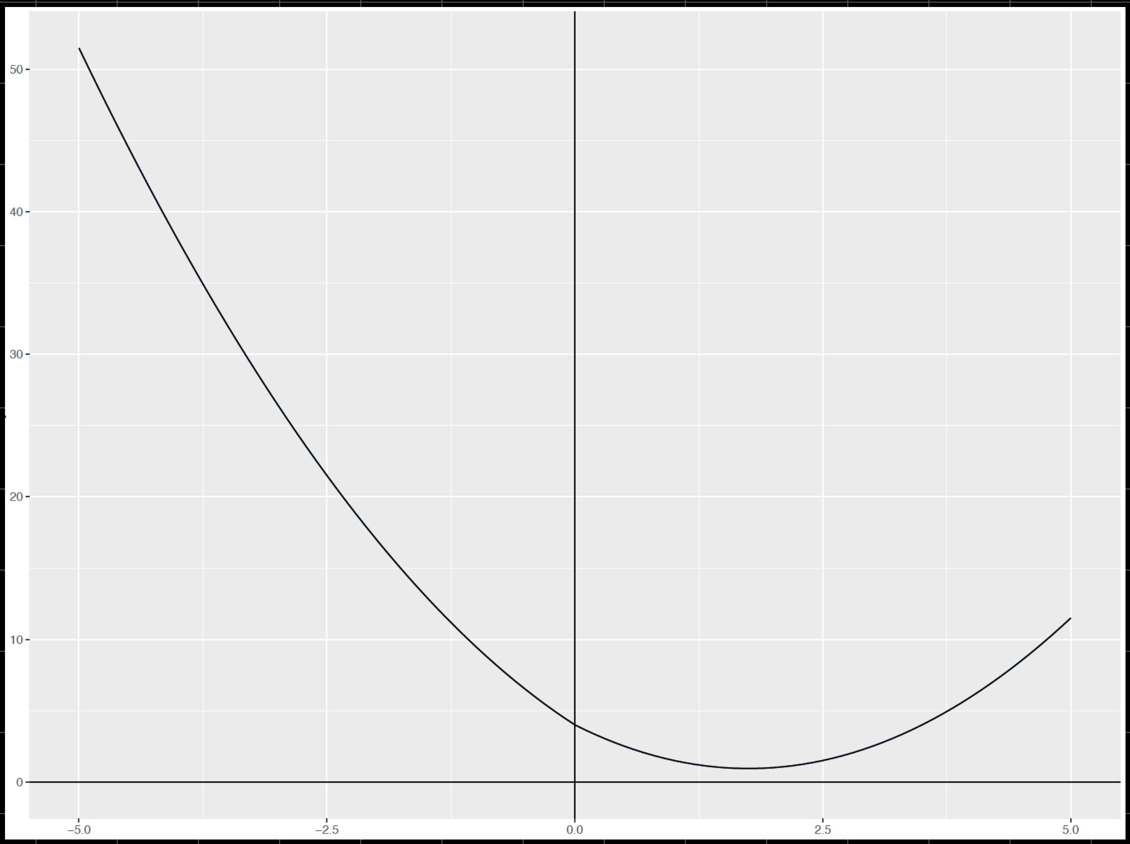
FOC

$$0.5 \cdot \frac{\beta}{|\beta|} = 0.$$

$$\beta = 0.$$

$$\hat{\beta}_{Lasso} = 0.$$

```
#d
eq = function(b){(2-b)^2 + 0.5*abs(b)}
x <- seq(-13,15, by=0.1)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_f_2.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```r
rm(list = ls())
eq = function(b){(2-b)^2 + 0.5*abs(b)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 0.9375
print(x_min)
#[1] 1.75
```

$$(2-\beta)^2 + 0.5|\beta|$$

$$\min_{\beta} \; (2-\beta)^2 + 0.5|\beta| = 0.$$

FOC

$$-2(2-\beta) + 0.5 \cdot \frac{\beta}{|\beta|} = 0$$

$$-4 + 2\beta + 0.5\frac{\beta}{|\beta|} = 0.$$

2. 2.5

2) 4.5

4.5

if $\beta > 0$

$$-4 + 2\beta + 0.5 = 0.$$

$$2\beta = 3.5$$

$$\beta = \frac{3.5}{2}$$

$$\underline{\beta = 1.75}$$

$$\hat{\beta}_{Lasso} = 1.75$$

If $\beta < 0.$

$$-4 - 2\beta - 0.5\beta = 0.$$

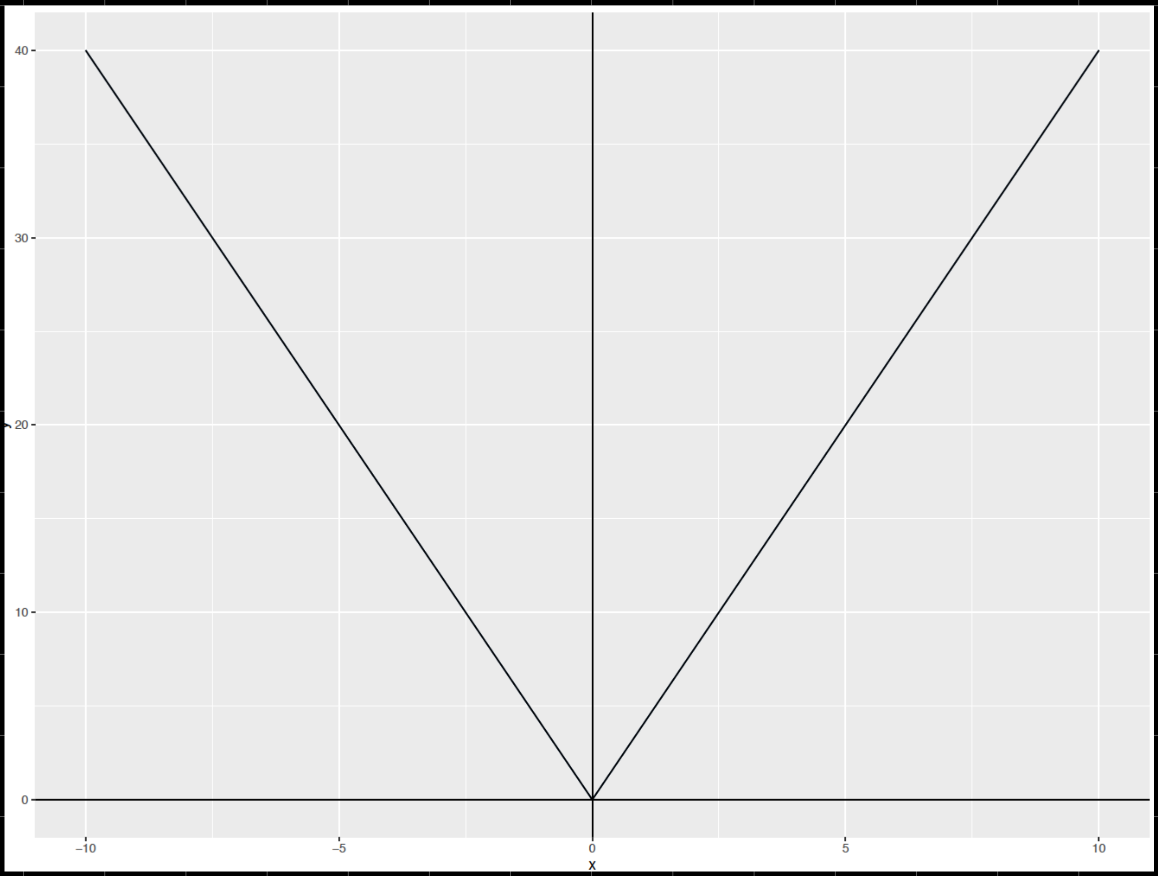
$$-4 - 2.5\beta = 0.$$

$$-2.5\beta = 4$$

$$\beta = -1.6.$$

#Q3-g labamta = 4 repeat c and d

```
#c
eq = function(b){4 * abs(b)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_g_1.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```

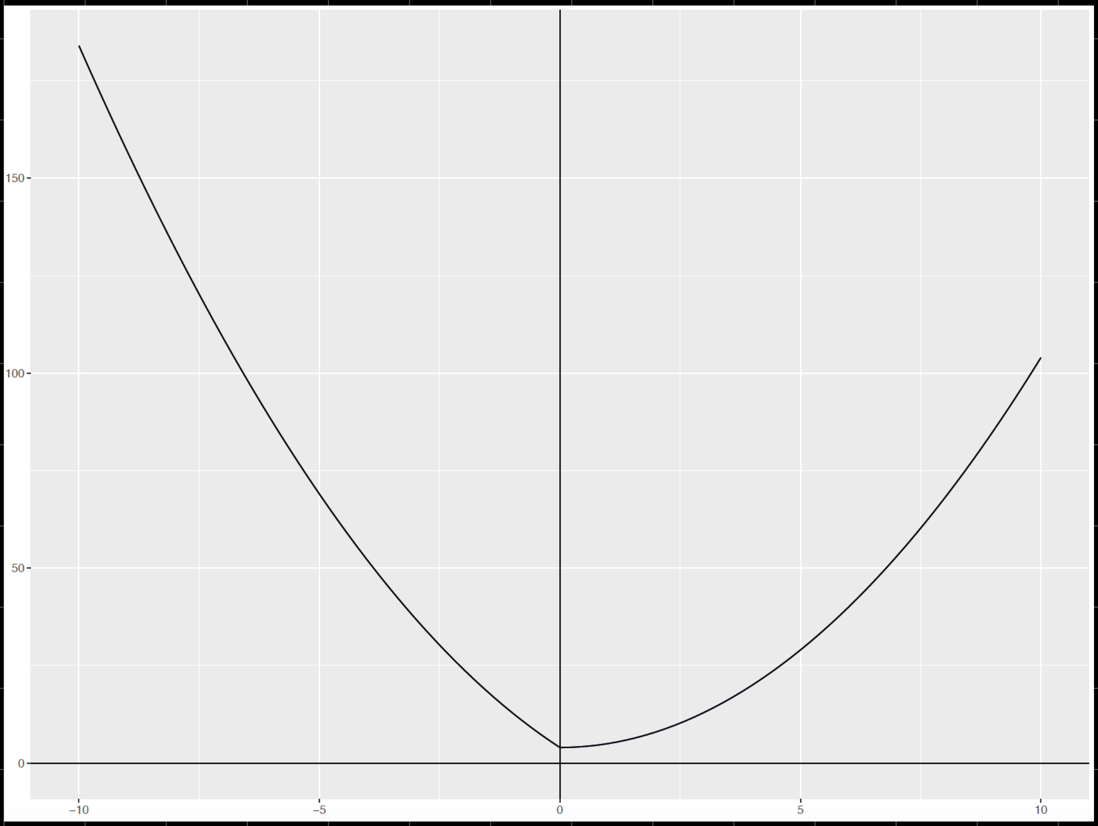


```
rm(list = ls())
eq = function(b){4 * abs(b)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 5.329071e-15
print(x_min)
#[1] 1.332268e-15
```

← no idea why not equal zero . . . .

$\hat{\beta}_{lasso} = 0$

```
#d
eq = function(b){(2-b)^2 + 4*abs(b)}
x <- seq(-10,10, by=0.001)
y <- eq(x)
df <- data.frame(x,y)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)


##saving the graph
pdf("pics/TieMa_homework1_Q3_g_2.pdf", height = 9, width = 12)
ggplot(df, aes(x=x, y=y)) + geom_line(col='#003b6f')  + geom_hline(yintercept =
0)+geom_vline(xintercept = 0) + stat_function(fun = eq)
dev.off()
```



```
rm(list = ls())
eq = function(b){(2-b)^2 + 4*abs(b)}
ans <- optimize(eq, interval = c(-10,10))
x_min = ans$minimum
y_min = ans$objective
print(y_min)
#[1] 4
print(x_min)
#[1] 2.88658e-15
```

$\Leftarrow$ almost zero.

$$(2-\beta)^2 + 4\cdot|\beta|$$

$$\min_{\beta} (2-\beta)^2 + 4\cdot|\beta|$$

$$-2(2-\beta) + 4\frac{\beta}{|\beta|} = 0$$

$$-4+2\beta + 4\frac{\beta}{|\beta|} = 0$$

if $\beta > 0$

$$-4 + 2\beta + 4 = 0.$$

$$\hat{\beta}_{Lasso} = 0$$

If $\beta < 0$.

$$-4 - 2\beta - 4\beta = 0$$

$$-2 - 3\beta = 0$$

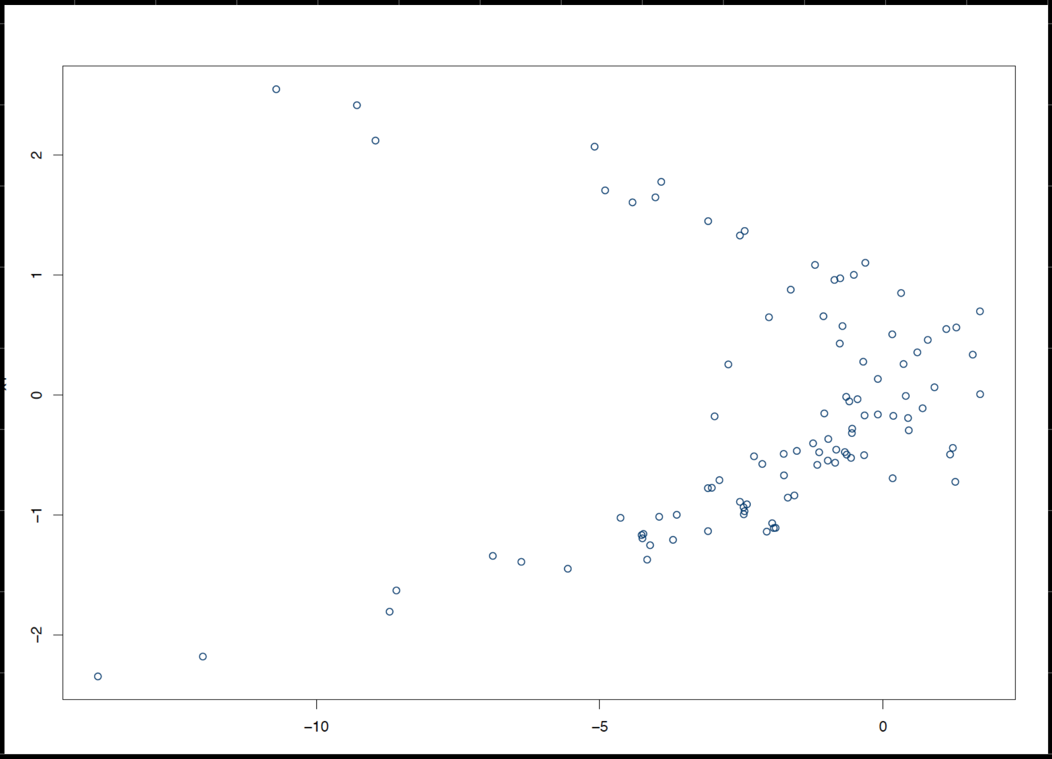
NA

#Exercise 4

4.a $Y = \beta_0 + 2x_1^2 + \epsilon$

```
#4-b
df <-data.frame(y,x1)
plot(df, pch=1,col='#003b6f', type = "p")


#save the graphy
pdf("pics/TieMa_homework1_Q4_b.pdf", height = 9, width = 12)
plot(df, pch=1,col='#003b6f', type = "p")
dev.off()

#clean the enviroment!
rm(list = ls())
#4-b-2
```

From the graphy we could find nonlinear relationship between x1 and y, we find most of point are on the right side of graph and x and y have a non linear relationship/



```
##################################################
#generate the simulated data (again, in order to avoid this section been polluted)
set.seed(1234)
n.obs <- 100
x1 <- rnorm(n.obs)
x2 <- x1^2
x3 <- x1^3
x4 <- x1^4
y <- x1 - 2*x1^2 + rnorm(n.obs)
Q4_data_set <- data.frame(y, x1, x2, x3, x4)
##################################################

model_one <- lm(y ~ x1, data=Q4_data_set)
##########

model_two <- lm(y ~ x1+x2, data=Q4_data_set)
#########

model_three <- lm(y ~ x1+x2+x3, data=Q4_data_set)
##########

model_four <- lm(y ~ x1+x2+x3+x4, data=Q4_data_set)

evil <- rbind(CV(model_one), CV(model_two), CV(model_three), CV(model_four))
rownames(evil) <- c('Model1', 'Model2', 'Model3', 'Model4')
evil

#.            CV        AIC        AICc       BIC        AdjR2
#Model1 9.217431 218.96020 219.21020 226.77571 0.009513121
#Model2 1.094918  11.79099  12.21204  22.21167 0.876435778
#Model3 1.101478  13.45234  14.09064  26.47819 0.875570749
#Model4 1.115254  15.42932  16.33255  31.06034 0.874289903


#by compare AIC and BIC we can concluded following
#model 2 > model 3 > model4 > model 1
#the smaller both AIC and BIC the better the model describe the relationship.] therefore, the
model 2 have lowest AIC and BIC, it is relative better model to use and the one I perfer
```

```
# Q4 - v Compute the k-fold cross-validation errors that result from fitting the four models. Use
#k = 5. Which model would you prefer? Is this what you expected? Explain your answer.

######################################################################

#clean the enviroment and generate everything again....
rm(list = ls())

#create the chart that carry the number od the cross-vaidation error
cv.error <- rep(NA,4)

#simulate data and enviorment.

set.seed(1234)
n.obs <- 100
x1 <- rnorm(n.obs)
x2 <- x1^2
x3 <- x1^3
x4 <- x1^4
y <- x1 - 2*x1^2 + rnorm(n.obs)
Q4_data_set_2 <- data.frame(y, x1, x2, x3, x4)

head(Q4_data_set_2, 2) #check it!!

model_one_with_data_set.cv <- glm(y ~ x1)
cv.error[1] <-cv.glm(Q4_data_set_2, model_one_with_data_set.cv, K=5)$delta[1]
print(cv.error)

######################################################################
#model 2 data set

model_two_with_data_set.cv <- glm(y ~ x1 + x2)
cv.error[2] <-cv.glm(Q4_data_set_2, model_two_with_data_set.cv, K=5)$delta[1]
print(cv.error)

######################################################################

#model 3

model_three_with_data_set.cv <- glm(y ~ x1 + x2 + x3)
cv.error[3] <-cv.glm(Q4_data_set_2, model_three_with_data_set.cv, K=5)$delta[1]
print(cv.error)

######################################################################
#model 4


model_four_with_data_set.cv <- glm(y ~ x1 + x2 +x3 +x4)
cv.error[4] <-cv.glm(Q4_data_set_2, model_four_with_data_set.cv, K=5)$delta[1]
print(cv.error)
###############################################################################
```

The 5- fold corss-validation errors
#8.865048 1.078610 1.157496 1.115773

The 5-fold cross-validation error suggests model 2 is the one with the better approach to the data set. AIC/BIC tests are working to find the best-fit model with the smallest number of parameter possible. AIC/BIC prefer the model with fewer parameters because it gives heavy punishment for size of parameters. Considering the size of the observation and all available model numbers, the size parameters are relatively small. Therefore, it is unsurprising that both AIC, BIC, and k-fold cross-validation errors choose the same outcome: model two.

q4 - e

```
summary(model_one)
#Coefficients:
#            Estimate Std. Error t value Pr(>|t|)
#(Intercept)  -2.0980     0.2966  -7.074 2.25e-10 ***
#  x1          0.4095     0.2932   1.397    0.166

summary(model_two)
#.           Estimate Std. Error t value Pr(>|t|)
#(Intercept)  0.13954    0.13506   1.033    0.304
#x1           1.00098    0.10597   9.446 2.11e-15 ***
#x2          -2.09591    0.07987 -26.241  < 2e-16 ***

summary(model_three)
#Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
#(Intercept)  0.13247    0.13610   0.973    0.333
  #x1          0.91259    0.18789   4.857 4.63e-06 ***
  #x2         -2.10637    0.08222 -25.618  < 2e-16 ***
 # x3          0.03231    0.05661   0.571    0.570

summary(model_four)
  #           Estimate Std. Error t value Pr(>|t|)
#(Intercept)  0.118715   0.165425   0.718    0.475
#x1           0.910799   0.189241   4.813 5.60e-06 ***
#x2          -2.075771   0.222837  -9.315 4.81e-15 ***
#x3           0.034229   0.058369   0.586    0.559
#x4          -0.006444   0.043575  -0.148    0.883
```

Q-5a

q4-e continue

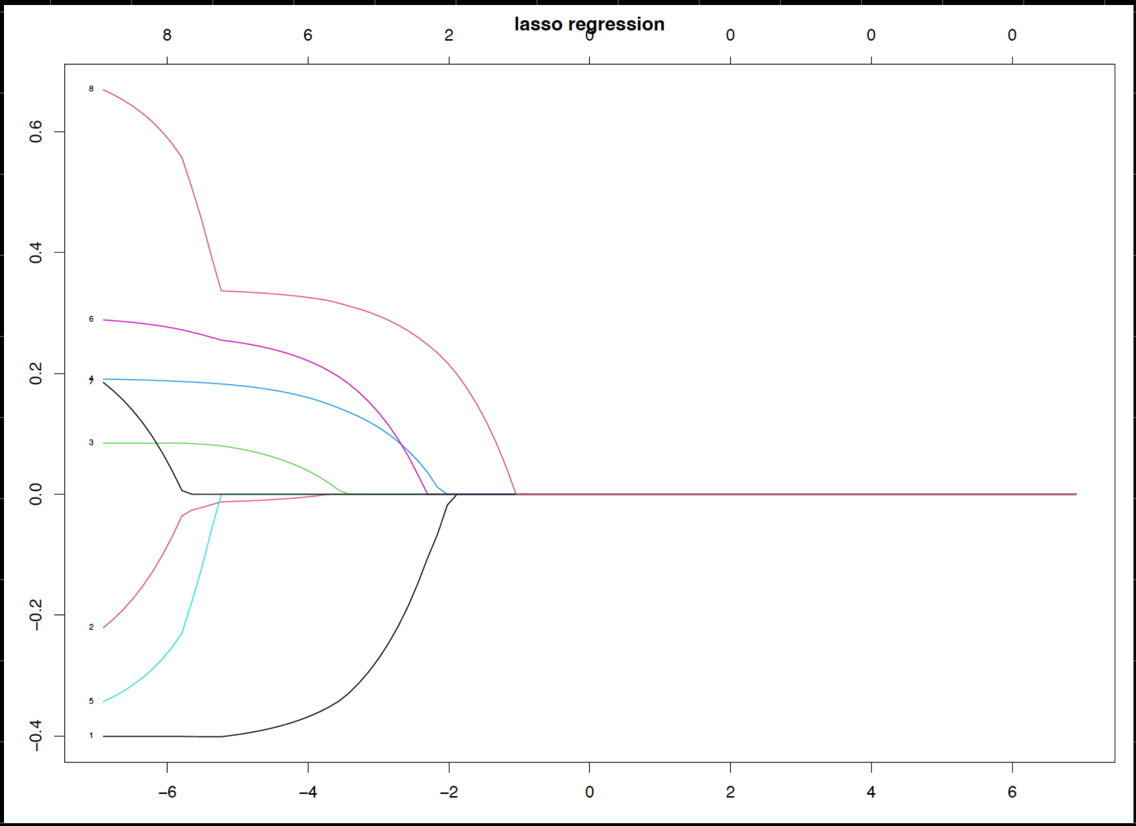
The summary of model_two, model_three, and model_four suggests that X1 and X2 have extremely high statistical significance, and the rest, X3 and X4, are not statistically significance. Therefore, model 2, which only has x1 and x2, is relatively the best. It is held consistent with the conclusion in questions 4-c and 4-d.

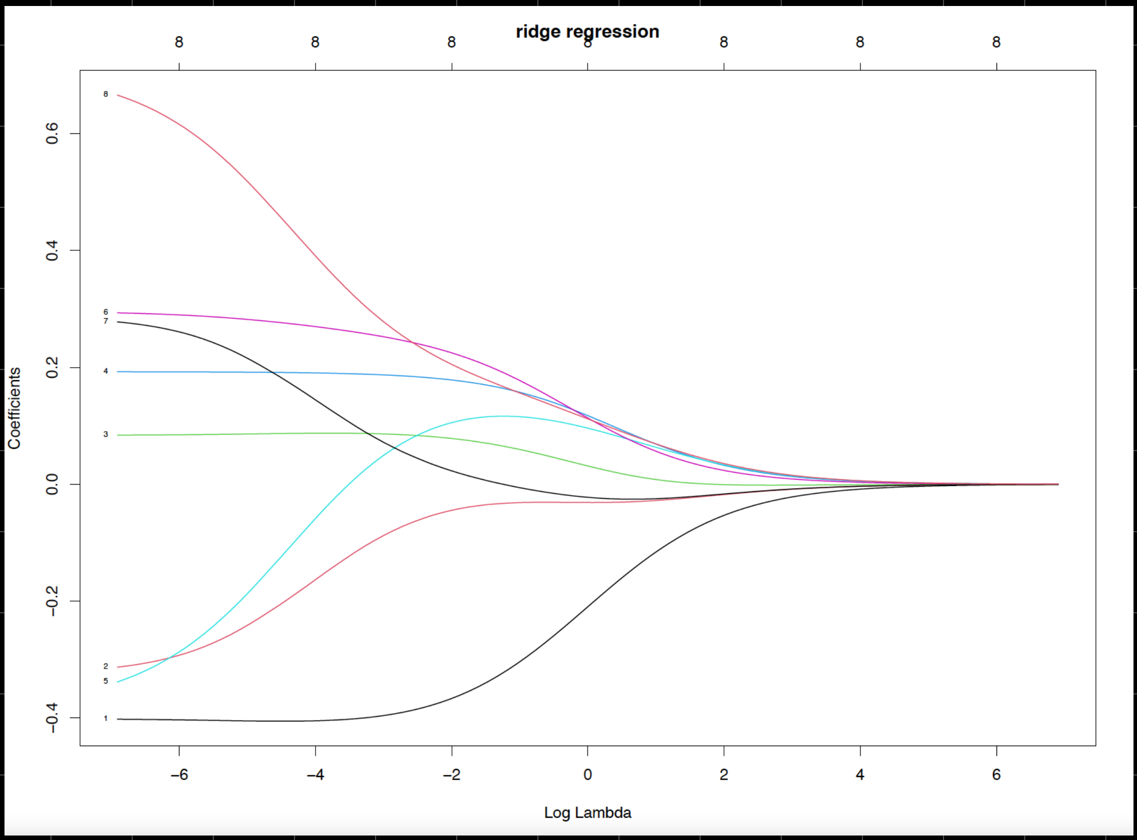
```
#Exercise 5-a
#Create a matrix X (545 × 9) with the 7 explanatory variables described above plus experience and schooling squared.
#Scale the matrix X such that all variables have the same variance. Create a vector y (545 × 1) with log wage.
library("leaps")

#lode the data
X_df <-read.csv("data/males1987.csv", header = TRUE)

#check the data!
class(X_df)
str(X_df)
head(X_df, 4)

#move the logwage in the first col
X_relocated <- relocate(X_df, LOGWAGE, before = )
#AFTER SPEND 3 HOURS FINALLY DOWN

#check the data!
str(X_relocated)
head(X_relocated, 4)
#It look good


#now square experience and school

X_relocated_final<- X_relocated %>% select(LOGWAGE, BLACK, EXPER, HISP, MAR, SCHOOL,
UNION, EXPER2) %>% mutate(SCHOOL2 = SCHOOL^2)

#check the data
head(X_relocated_final, 4)
#its look good!

X_Scale <- X_relocated_final%>%
  select(LOGWAGE, BLACK, EXPER, HISP, MAR, SCHOOL, UNION, SCHOOL2, EXPER2)  %>%
  transmute(LOGWAGE_scale =scale(LOGWAGE) , BLACK, EXPER_scale = scale(EXPER),
            HISP, MAR, SCHOOL_scale = scale(SCHOOL), UNION, EXPER2_scale = scale(EXPER2),
SCHOOL2_scale = scale(SCHOOL2))
#sorry for this line of code is way too long...
#I did not really sure how to shrink it down...
#It select the all the variable
#and using the scale function to scale the non-dummy variable...
```

```
#check the data!
head(X_Scale, 4)
#it look good!

#check what the data is
class(X_Scale)
# its data frame!

#transfer to matrix
X <- data.matrix(X_Scale)

###############################
#Create a vector y with log wage
#recall we have the matrix X that include all the data

x<- X_Scale%>%
  dplyr::select(
    BLACK, EXPER_scale, HISP, MAR, SCHOOL_scale, UNION, EXPER2_scale,
    SCHOOL2_scale
  )%>%
  data.matrix()

#check the data!
head(x,5)

y_THE_LOGWAGE <- X_Scale$LOGWAGE_scale
data_question5 <- data.frame(y_THE_LOGWAGE, x)

#check the data
head(data_question5, 5)

#the final matrix
matrix_X <-   data.matrix(data_question5)
```

← the final matrix

```
#Q5-b
# ridge regression for different values of lambda
evil_grid <- 10^seq(3, -3, length = 100)
ridge.mod <- glmnet(x, y_THE_LOGWAGE, alpha = 0, lambda = evil_grid)

#plot ridge results
plot(ridge.mod, xvar = "lambda", label = TRUE, main = "ridge regression standardized
coefficients")

#save the graphy
pdf("pics/TieMa_homework1_Q5_b_ridge.pdf", height = 9, width = 12)
plot(ridge.mod, xvar = "lambda", label = TRUE, main = "ridge regression")
dev.off()

#plot the lasso
evil_grid <- 10^seq(3, -3, length = 100)
lasso.mod <- glmnet(x, y_THE_LOGWAGE, alpha = 1, lambda = 10^seq(3, -3, length = 100))
plot(lasso.mod, xvar = "lambda", label = TRUE, main = "lasso regression")

#save the graphy
pdf("pics/TieMa_homework1_Q5_b_lasso.pdf", height = 9, width = 12)
plot(lasso.mod, xvar = "lambda", label = TRUE, main = "lasso regression")
dev.off()
```





Each line corresponds to a coefficient estimate. but lasso performs the variable selection therefore we can see some coeffcient end really first but ridge does not, which make it end in the later part of graphy.

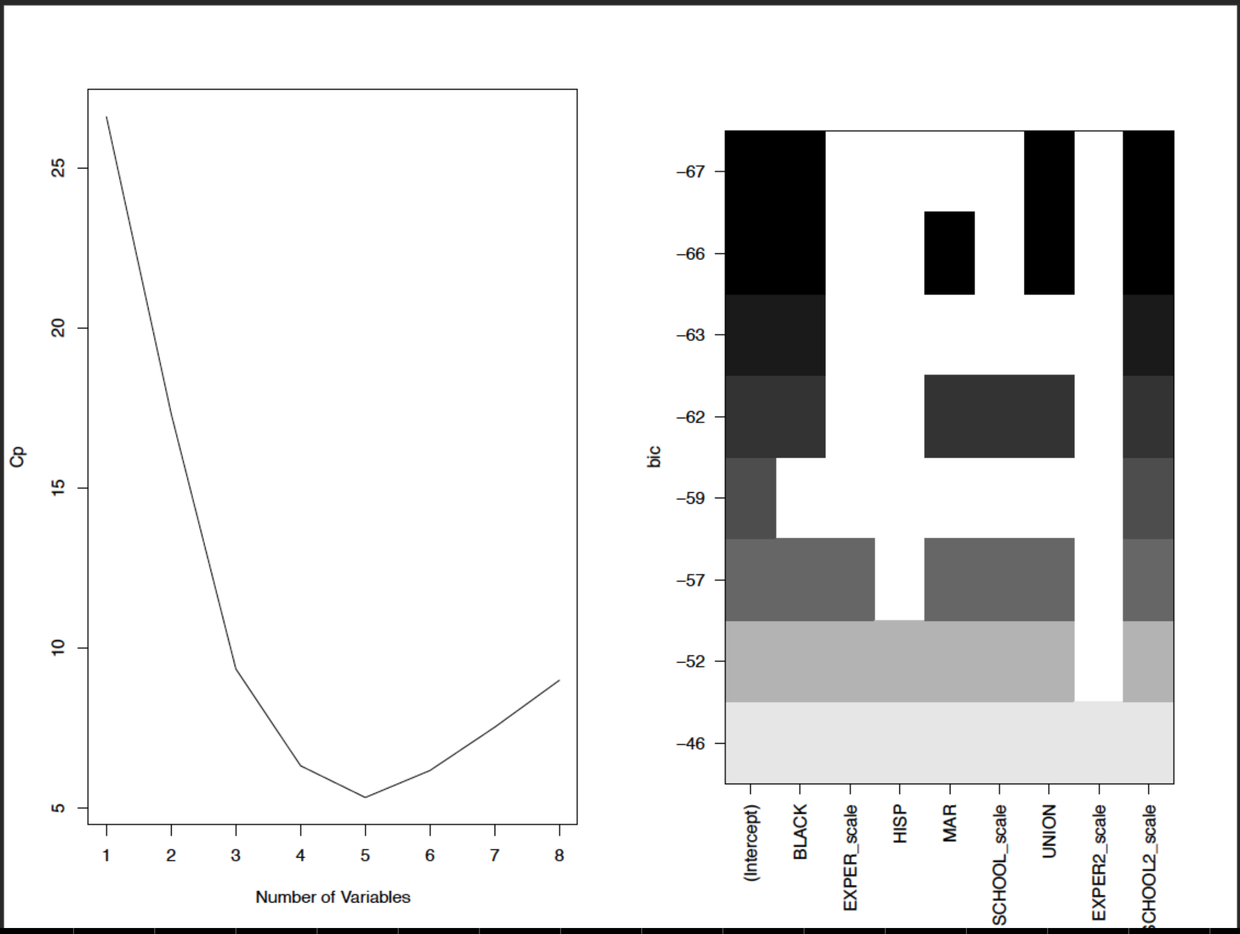
```
##################################################################################
############################################################
#Q5-C-1 Estimate the parameters by OLS using the full sample.

# create a table
parameters_of_Q5<- matrix(rep(NA),9,1)

coef1 <- coef(lm(y_THE_LOGWAGE ~ x, data= data_question5))

#fill it with data
parameters_of_Q5[,1] <- coef1

#put the row name
rownames(parameters_of_Q5) <- colnames(data_question5)
colnames(parameters_of_Q5)[1] <-'OLS'

#summon the table!
print(parameters_of_Q5)


##################################################################################
#Q5-C-2 Which variables are statistically significant at the 10% level?

#yes, its just same things..
#I realize I was over thigns again....

something <- lm(y_THE_LOGWAGE ~ x, data= data_question5)
something_summary <- summary(something)
print(something_summary)



#Coefficients:
```

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-0.16299	0.07359	-2.215	0.02718	*
xBLACK	-0.40050	0.12984	-3.085	0.00214	**
xEXPER_scale	-0.33403	0.38512	-0.867	0.38614	
xHISP	0.08331	0.11300	0.737	0.46126	
xMAR	0.19264	0.08260	2.332	0.02006	*
xSCHOOL_scale	-0.39173	0.36589	-1.071	0.28483	
xUNION	0.29683	0.09228	3.217	0.00138	**
xEXPER2_scale	0.29545	0.40474	0.730	0.46572	
xSCHOOL2_scale	0.71771	0.34767	2.064	0.03947	*

Black, MAR, Union and xSCHOOL2_scale plus intercept are statistically significant at the 10% level.

#Q5-D Based on the BIC and BIC, which variables are included in the best model?

```
regfit.all <- regsubsets(y_THE_LOGWAGE~. , data_question5, nvmax = 10)
something_summary_1<- summary(regfit.all)

par(mfrow=c(1,2))
plot(something_summary_1$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
m.cp <- which.min(something_summary_1$cp)
plot(regfit.all, scale = "bic")
layout(1)

pdf("pics/TieMa_homework1_Q5_d.pdf", height = 9, width = 12)
par(mfrow=c(1,2))
plot(something_summary_1$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
m.cp <- which.min(something_summary_1$cp)
plot(regfit.all, scale = "bic")
layout(1)
dev.off()
```



According to AIC(CP), the best k = 5 (5 variable)
According the BIC, also the best k at 5 and those 5 variable are BLACK, MR, Union, school2_scaled plus intercept. It hold consistent with the conclution at question 5-b.

Q5 - e

```
lasso.cv <- cv.glmnet(x, y_THE_LOGWAGE, alpha = 1, nfolds = 5)
model3 <- glmnet(x, y_THE_LOGWAGE, alpha = 1, lambda = lasso.cv$lambda.min)
coef3 <- coef(model3)
print (coef3)
```

```
                    s0
(Intercept)       -0.123735661
BLACK             -0.373701647
EXPER_scale       -0.005662817
HISP               0.045290208
MAR                0.163475106
SCHOOL_scale      .
UNION              0.226330843
EXPER2_scale      .
SCHOOL2_scale      0.327358681
```

- The variable that been force to be zero. are the school _scale and the exper2_sclaed.
- compare with OLS, we could find out such variabes with high statistically significant will get higher number in the lasso coefficients. Same as the the 5 variables that AIC/BIC suggested : BLACK, MR, Union, school2_scaled plus intercep

```r
#Q5 - f


# split the sample into train/test sets
train <- sample(nrow(data_question5), round(nrow(data_question5)/2))
cv.error <- rep(NA,3)

# least squares
ols.cv <- lm(y_THE_LOGWAGE ~ x, data = data_question5, subset = train)
cv.error[1] <- mean((y_THE_LOGWAGE - predict(ols.cv,  data_question5))[-train]^2)

# ridge
ridge.cv <- cv.glmnet(x[train,], y_THE_LOGWAGE[train], alpha = 0, nfolds = 10)
ridge.lam <- ridge.cv$lambda.min
cv.error[2] <- mean((y_THE_LOGWAGE - predict(ridge.cv, s = ridge.lam, newx = x))[-
train]^2)

# lasso
lasso.cv <- cv.glmnet(x[train,], y_THE_LOGWAGE[train], alpha = 1, nfolds = 10)
lasso.lam <- lasso.cv$lambda.min
cv.error[3] <- mean((y_THE_LOGWAGE - predict(lasso.cv, s = lasso.lam, newx = x))[-
train]^2)

cv.error <- data.frame(cv.error)
rownames(cv.error) <- c("ols","ridge","lasso")
cv.error
```

```
       cv.error
ols   0.7658656
ridge 0.7677745
lasso 0.7686222
```

The test sample error of ols, ridge and lasso are extremely close, which provides a similar result on which variable is better on fit within data. Such results match the outcome of questions c, d and e.