ML-Problem-Set-2

Gyongyver Kamenar (2103380)

3/7/2022

```
library(glmnet)
library(ggplot2)
library(purrr)
library(tidyverse)
library(kableExtra)
library(ggforce)
# My theme
devtools::source_url('https://raw.githubusercontent.com/gyongyver-droid/ceu-data-analysis/master/Assignm
theme_set(theme_gyongyver())
```

Problem 1

A)

Show that the solution to this problem is given by $\hat{\beta}_0^{ridge} = \sum_{i=1}^{n} Y_i/(n+\lambda)$. Compare this to the OLS estimator.

To minimize the expression we have to take the derivative and set it equal to 0.

$$\sum_{i=1}^{n} 2 * (Y_i - b) * (-1) + 2\lambda b = 0$$

Transform to

$$-2\sum_{i=1}^{n}(Y_i - b) + 2\lambda b = 0$$

Divide by 2

$$-\sum_{i=1}^{n}(Y_i - b) + \lambda b = 0$$

Divide the summa into 2 parts. Only the Y part contains i and the b is taken n times.

$$-[\sum_{i=1}^{n}(Y_i) - nb] + \lambda b = 0$$

Reorganize the sides:

$$nb + \lambda b = \sum_{i=1}^{n}(Y_i)$$

$$(n+\lambda)b = \sum_{i=1}^{n}(Y_i)$$

Divide by $n + \lambda$

$$(n + \lambda)b = \sum_{i=1}^{n}(Y_i)$$

$$b = \sum_{i=1}^{n}(Y_i)/(n + \lambda)$$

Which is the solution of the problem:

$$\hat{\beta}_0^{ridge} = \sum_{i=1}^{n}(Y_i)/(n + \lambda)$$

Compating this to the OLS:

$$\hat{\beta}_0^{OLS} = \overline{Y} = \sum_{i=1}^{n}(Y_i)/n$$

So based on the two above formulas, we can see that $\hat{\beta}_0^{ridge}$ has $+\lambda$ in the denominator. We know that $\lambda = 0$ in the Ridge regression so the $\hat{\beta}_0^{ridge}$ coefficient will be smaller than the OLS coefficient. The higher the λ (penalty term) the higher the denominator so the ridge coefficient will be smaller. So we can see that λ is really a penalty / shrinkage parameter.

b)

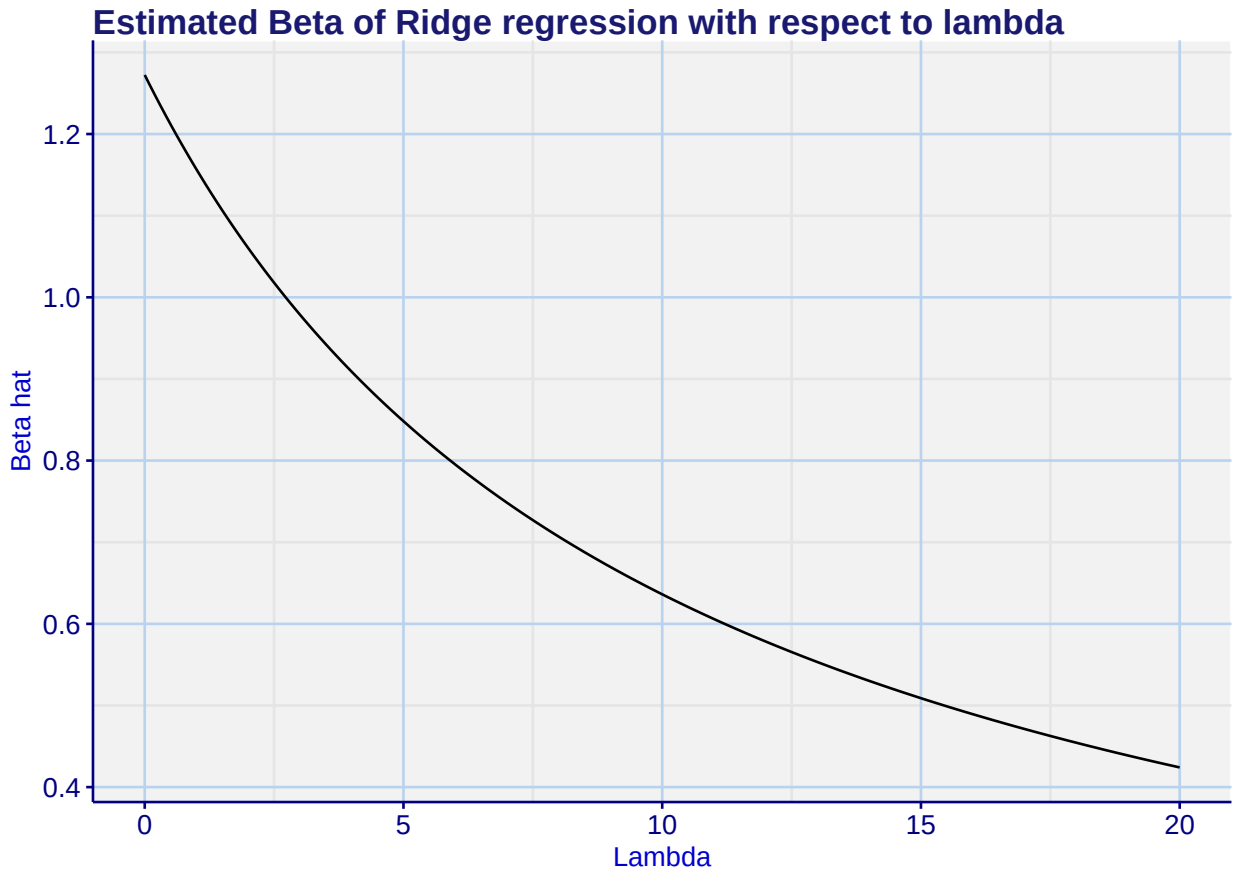
```
set.seed(111111)
simulate_ridge<-function(n=10,sd=2){
  n=n
  e <- rnorm(n=n,mean=0,sd=sd)
  beta<-matrix(1,nrow = n,ncol = 1)
  y <- beta + e

  lambda <-seq(0,20,0.1)
  beta_hat <-sum(y)/(n+lambda)
  data.frame(lambda,beta_hat, beta=1,y_hat=beta_hat+e)
}

simulate_ridge() %>% kable()
```

lambda	beta_hat	beta	y_hat
0.0	1.8272325	1	2.6968993
0.1	1.8091411	1	8.7440552
0.2	1.7914044	1	5.4299928
0.3	1.7740121	1	1.7952500
0.4	1.7569543	1	2.3370546
0.5	1.7402214	1	0.2249771
0.6	1.7238042	1	2.1502067
0.7	1.7076939	1	-0.9976626
0.8	1.6918819	1	1.1467062
0.9	1.6763601	1	2.2435516
1.0	1.6611205	1	2.5307872
1.1	1.6461554	1	8.5810695
1.2	1.6314576	1	5.2700460
1.3	1.6170199	1	1.6382578
1.4	1.6028355	1	2.1829358
1.5	1.5888978	1	0.0736535
1.6	1.5752004	1	2.0016029
1.7	1.5617372	1	-1.1436194
1.8	1.5485021	1	1.0033264
1.9	1.5354895	1	2.1026811
2.0	1.5226937	1	2.3923605
2.1	1.5101095	1	8.4450236
2.2	1.4977316	1	5.1363200
2.3	1.4855549	1	1.5067928
2.4	1.4735746	1	2.0536749
2.5	1.4617860	1	-0.0534583
2.6	1.4501845	1	1.8765870
2.7	1.4387657	1	-1.2665908
2.8	1.4275254	1	0.8823497
2.9	1.4164593	1	1.9836509
3.0	1.4055635	1	2.2752303
3.1	1.3948340	1	8.3297481
3.2	1.3842670	1	5.0228555
3.3	1.3738590	1	1.3950969
3.4	1.3636063	1	1.9437066
3.5	1.3535056	1	-0.1617388
3.6	1.3435533	1	1.7699558
3.7	1.3337464	1	-1.3716102
3.8	1.3240815	1	0.7789058
3.9	1.3145558	1	1.8817473
4.0	1.3051661	1	2.1748329
4.1	1.2959096	1	8.2308237
4.2	1.2867835	1	4.9253719
4.3	1.2777850	1	1.2990229
4.4	1.2689115	1	1.8490118
4.5	1.2601603	1	-0.2550840
4.6	1.2515291	1	1.6779316
4.7	1.2430153	1	-1.4623413
4.8	1.2346166	1	0.6894409
4.9	1.2263305	1	1.7935221
5.0	1.2181550	1	2.0878218
5.1	1.2100877	1	8.1450019
5.2	1.2021266	1	4.8407151
5.3	1.1942696	1	1.2155075
5.4	1.1865146	1	1.7666149
5.5	1.1788597	1	-0.3363847
5.6	1.1713029	1	1.5977054
5.7	1.1638424	1	1.5415148

```r
ggplot(simulate_ridge())+
  geom_line(aes(x=lambda,y=beta_hat))+
  labs(title = "Estimated Beta of Ridge regression with respect to lambda",y="Beta hat", x="Lambda")
```



C)

Repeat part b) 1000 times, for each value of lambda compute bias, variance and MSE of $\hat{\beta}_0^{ridge}$.

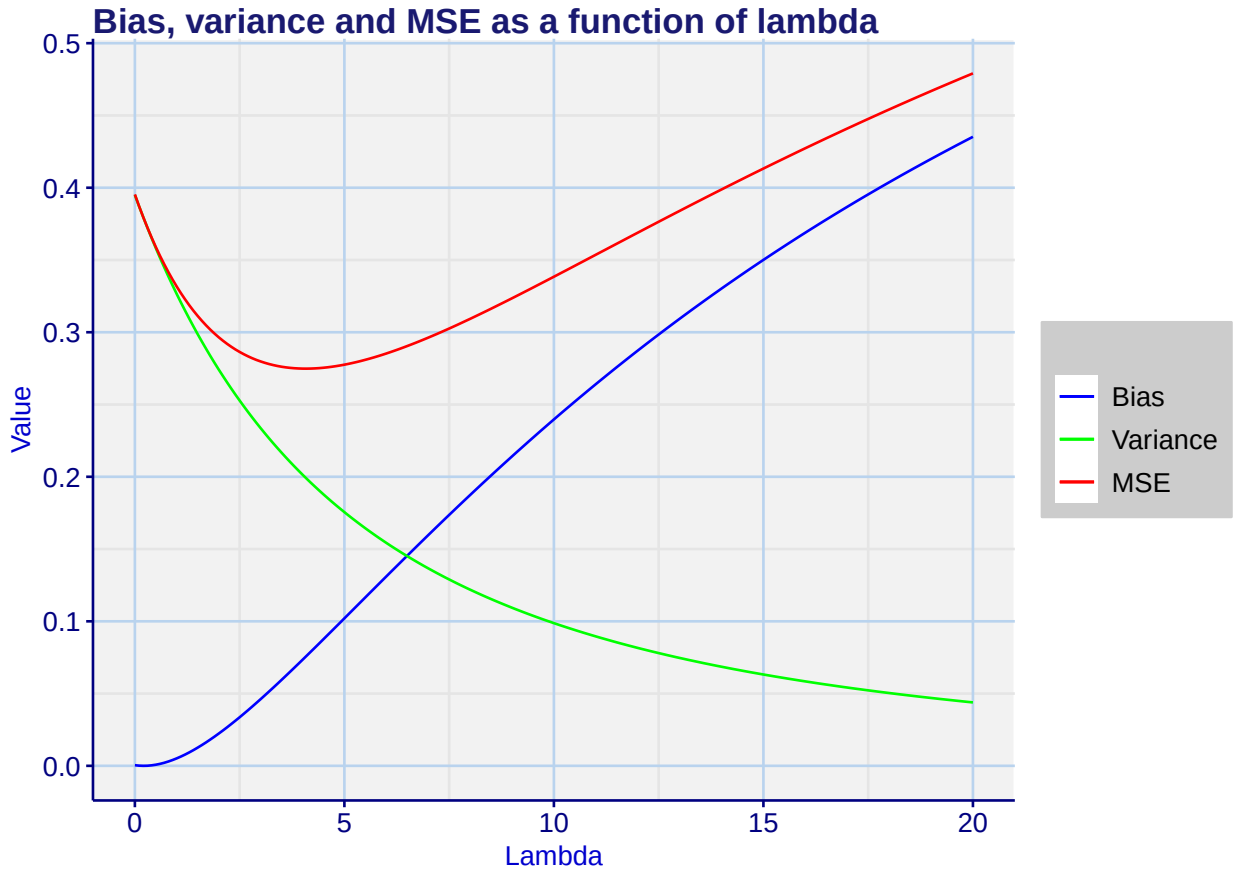
```r
library(purrr)
df_1 <- map_df(seq(1,1000,1), ~{
  results = simulate_ridge(n=10)
  tibble(
    lambda = results$lambda,
    beta_hat = results$beta_hat,
    error = 1 - results$beta_hat
  )


}) %>% group_by(lambda) %>% summarise(bias=mean(error)^2, var=var(beta_hat), mse=bias+var)
```

D)

Plot bias, variance and MSE as a function of lambda and interpret the result.

```
ggplot(df_1, aes(x=lambda))+
  geom_line(aes(y=bias,color="Bias"))+
  geom_line(aes(y=var, color="Variance"))+
  geom_line(aes(y=mse, color="MSE"))+
  scale_colour_manual("",
                      breaks = c("Bias", "Variance", "MSE"),
                      values = c("blue", "green", "red"))+
  labs(title = "Bias, variance and MSE as a function of lambda",y="Value",x="Lambda")
```



We can see, that as lambda is increasing, the bias is also increasing and the variance decreasing as we had expected based on on the theory of bias-variance tradeoff. The MSE takes U-shape as expected, so we can calculate that the lowest MSE is around lamdba = 5.

Problem 2

A)

$$max_{u_1,u_2}\ Var(u_1X + u_2Y) \quad s.t.\ \ u_1^2 + u_2^2 = 1$$

and suppose that

$$Var(X) > Var(Y) \quad and \quad Cov(X,Y) = E(XY) = 0.$$

We can expand the variance formula:

$$Var(u_1X + u_2Y) = u_1^2 Var(X) + u_2^2 Var(Y) + 2u_1u_2Cov(X,Y)$$

and we know that the covariance is 0, so the problem is the following:

$$max_{u_1,u_2}\ (u_1^2 Var(X)+u_2^2 Var(Y)) \qquad s.t.\ \ u_1^2+u_2^2=1\ \ and\ \ Var(X)>Var(Y)\ \ \ and\ \ \ Cov(X,Y)=E(XY)=0$$

From this, it is trivial to see that $u_1^2 Var(X)+u_2^2 Var(Y)$ will be maximized if $u_1^2=1\ and\ u_2^2=0$ because of the $Var(X)>Var(Y)$ condition. Therefore, there is no need to actually derive optimization problem.

The first principle component vector is $(u_1,u_2)=(1,0)$.

Illustration

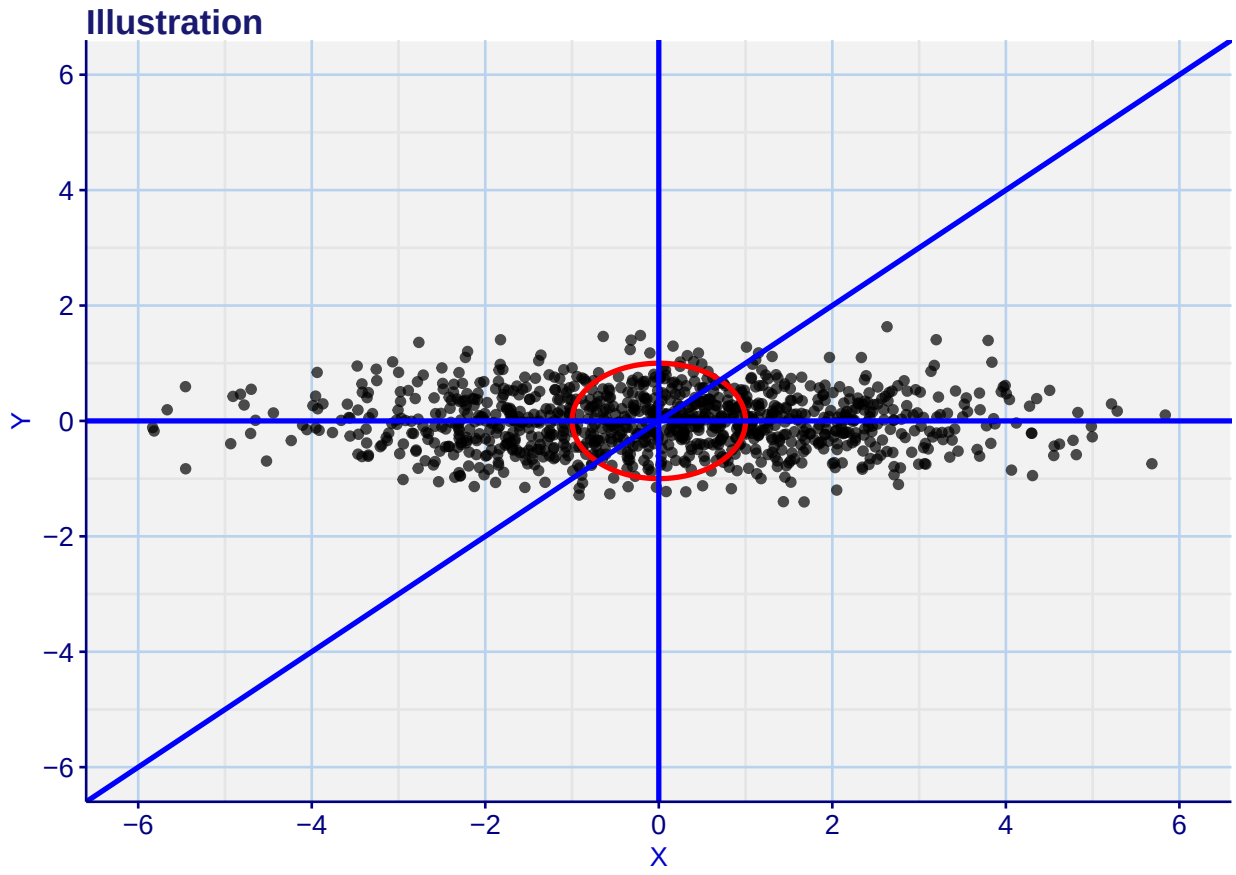
```
set.seed(20220307)
x<-rnorm(1000,mean=0,sd=2)
set.seed(43293)
y<-rnorm(1000,mean=0,sd=0.5)
# Covariance is almost zero
cov(x,y)
```

```
## [1] 0.007823966
```

```
circles <- data.frame(
  x0 = 0,
  y0 = 0,
  r = 1
)

# Behold the some circles

data.frame(x,y) %>% ggplot()+
  geom_point(aes(x=x,y=y), alpha=0.7)+
  geom_circle(aes(x0 = x0, y0 = y0, r = r), data = circles, color="red", size=1)+
  geom_abline(intercept = 0,slope = c(0,1,99999999), color="blue", size=1)+
  scale_x_continuous(limits = c(-6,6), breaks = seq(-6,6,2))+
  scale_y_continuous(limits = c(-6,6), breaks = seq(-6,6,2))+
  labs(title = "Illustration",x="X",y="Y")
```



B)

The problem:

$$max_{u_1,u_2}\ Var(u_1X + u_2Y) \quad s.t.\ \ u_1^2 + u_2^2 = 1 \ \ and \ \ Var(X) = Var(Y) = 1 \ \ and \ \ Cov(X,Y) = E(XY) = 0\ .$$

We can expand the vaiance formula as before and neglect the covatiance term becasue it is zero:

$$Var(u_1X + u_2Y) = u_1^2Var(X) + u_2^2Var(Y) + 2u_1u_2Cov(X,Y) = u_1^2Var(X) + u_2^2Var(Y) \ \ .$$

We can substitute 1 instead of $Var(X)$ and $Var(Y)$:

$$u_1^2Var(X) + u_2^2Var(Y) = u_1^2 * 1 + u_2^2 * 1 = u_1^2 + u_2^2$$

So the maximization problem is:

$$max_{u_1,u_2}\ (u_1^2 + u_2^2) \quad s.t. \ \ u_1^2 + u_2^2 = 1\ .$$

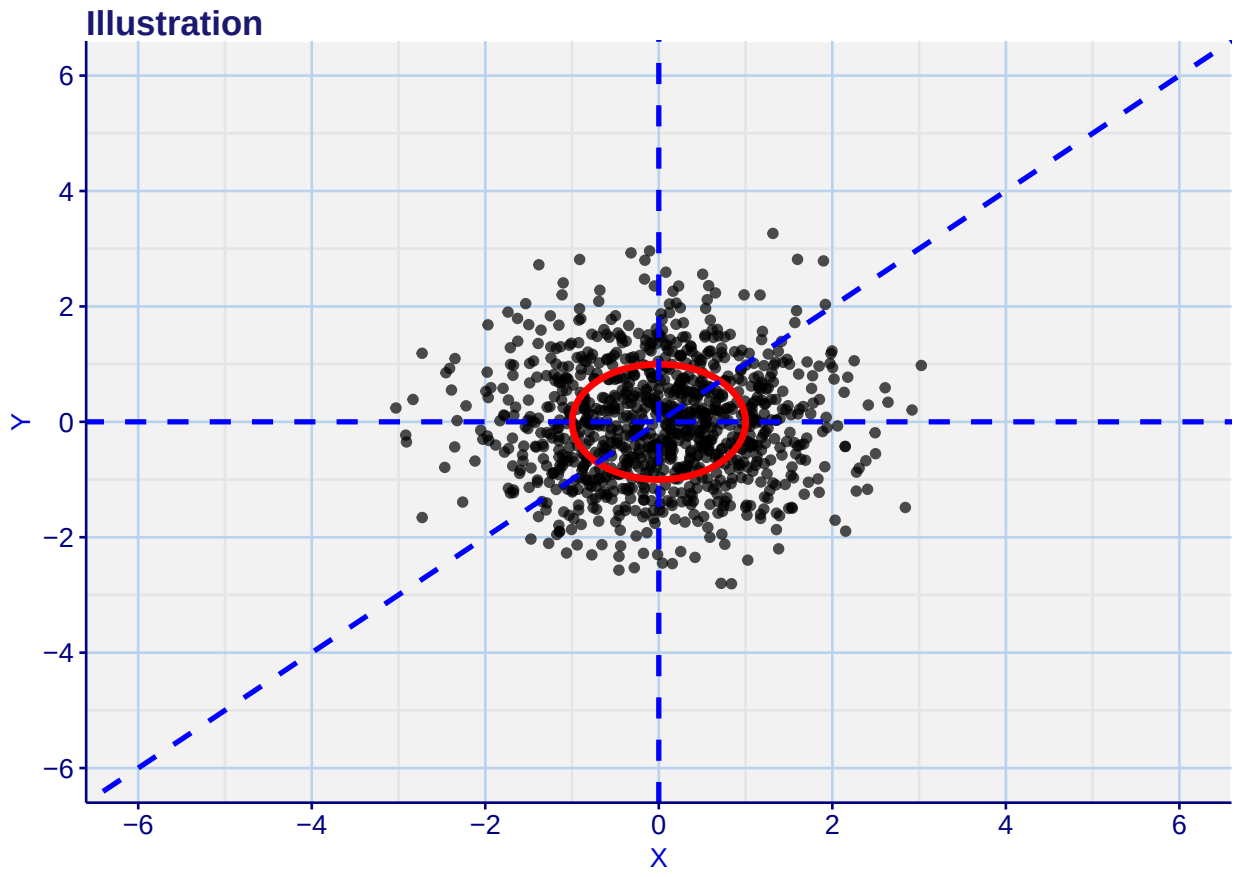
So regardless of the (u_1, u_2) values of the unit vector, the expression will be maximized and its value will be 1 because of the $(u_1^2 + u_2^2 = 1)$ condition. Intuitively, becasue of the equal variance, the X,Y points will form a circle around their mean, and in each direction of a unit vector, the variance will be the same. See the illustration below:

```r
set.seed(20220307)
x<-rnorm(1000,mean=0,sd=1)
set.seed(43293)
y<-rnorm(1000,mean=0,sd=1)
# Covariance is almost zero
cov(x,y)
```

```
## [1] 0.007823966
```

```r
circles <- data.frame(x0 = 0,y0 = 0,r = 1)

# Plot
data.frame(x,y) %>% ggplot()+
  geom_point(aes(x=x,y=y), alpha=0.7)+
  geom_circle(aes(x0 = x0, y0 = y0, r = r), data = circles, color="red", size=1.3)+
  geom_abline(intercept = 0,slope = c(0,1,99999999), color="blue", size=1, linetype="dashed")+
  scale_x_continuous(limits = c(-6,6), breaks = seq(-6,6,2))+
  scale_y_continuous(limits = c(-6,6), breaks = seq(-6,6,2))+
  labs(title = "Illustration",x="X",y="Y")
```



Problem 3

A)

Solution: iv) Steadily decrease

For s=0 all $\beta_j = 0$ so in this case the training error will be its maximum, becasue there are basically no explanatory variables just a constant β_0 which will be the mean. Then, for each increased s the β_j coefficients will increase and explain more and more (fit better and better on the data), so the RSS will be lower and lower. Eventually, with high enough s the β_j coefficients will reach the OLS estimates and in this case the RSS will be lower then in any case before. **So the training RSS will steadily decrease.**

B)

Solution: ii) Decrease initially, and then eventually start increasing in a U shape.

Initially, similarly to the previous case, at $s = 0$ there will be no explanatory coefficients so the RSS will be really high. Then with higher s, the β_j coefficients will increase, the model will better fit the data so the RSS will decrease. However, after a certain point we will see the difference on training and test RSS. The model will better and better fit the training data, but it will result overfitting. The model will not fit the test data as well as the test data. SO eventually the test RSS will start to increase. **Overall, this will result a U shaped RSS curve respect to s value.**

C)

Solution: iii) Steadily increase

Initially, for $s = 0$ there will be no β_j coefficients so the model's prediction will be a constant, which has no variance. With higher and higher s the β_j coefficient are increasing, more and more coefficients will we included in the model inducing more and more variance in the estimation. Eventually, the variance will be around the variance of the training data $\hat{y}$. **So the variance will steadily increase.**

D)

Solution: iv) Steadily decreasing

Whan $s = 0$ the bias will be the highest, because with no explanatory variance the model will underfit the data. Increasing the s value, the β_j coefficients will also increase and more coefficients will be included, resulting a better fit and lower bias. Eventually, when s is high enough we can reach the OLS estimation, of which the estimates are unbiased. **SO overall, the bias will steadily decrease.**

E)

SOlution: v) Remain constant

By definition, irreducible error is that we cannot remove or influence by the model because it is causes by outside factors. So for a given data, the irreducible error will remain the same, regardless of the model parameters. **So the irreducible error will remain constant**.