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/Assign
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)

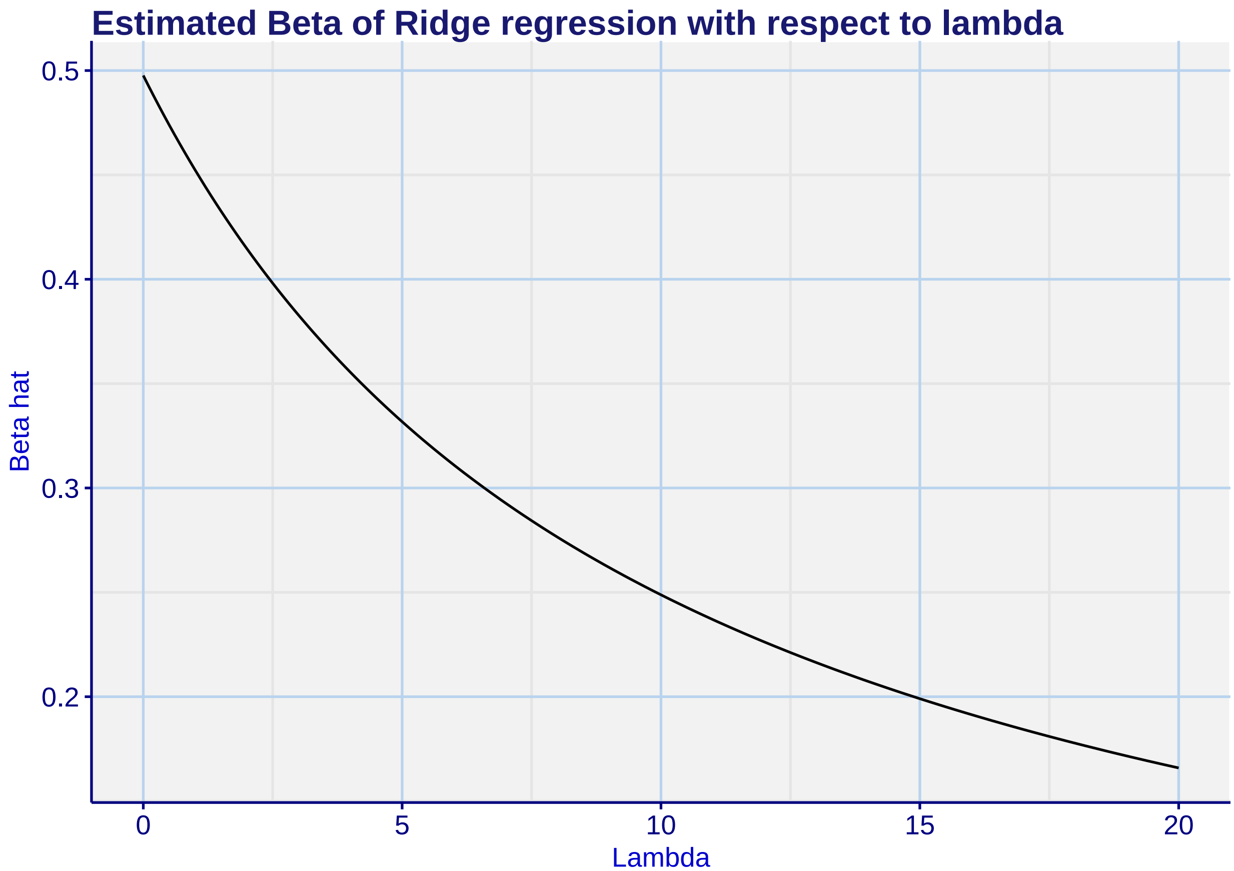
```
simulate_ridge<-function(n=10,sd=2){
  n=n
  e <- rnorm(n=n,mean=0,sd=sd)
  beta<-matrix(1,nrow = n,ncol = 1)
  y <- beta + e

  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	0.0506929	1	-3.3827012
0.1	0.0501910	1	0.2067474
0.2	0.0496990	1	0.2410487
0.3	0.0492164	1	1.9894393
0.4	0.0487432	1	-0.7894659
0.5	0.0482790	1	-2.2853626
0.6	0.0478235	1	-2.2822148
0.7	0.0473766	1	2.2473045
0.8	0.0469379	1	-1.0584694
0.9	0.0465073	1	-3.8939300
1.0	0.0460845	1	-3.3873097
1.1	0.0456693	1	0.2022257
1.2	0.0452615	1	0.2366113
1.3	0.0448610	1	1.9850839
1.4	0.0444675	1	-0.7937417
1.5	0.0440808	1	-2.2895607
1.6	0.0437008	1	-2.2863375
1.7	0.0433273	1	2.2432553
1.8	0.0429601	1	-1.0624472
1.9	0.0425991	1	-3.8978382
2.0	0.0422441	1	-3.3911501
2.1	0.0418950	1	0.1984514
2.2	0.0415516	1	0.2329013
2.3	0.0412138	1	1.9814367
2.4	0.0408814	1	-0.7973277
2.5	0.0405543	1	-2.2930872
2.6	0.0402325	1	-2.2898058
2.7	0.0399157	1	2.2398437
2.8	0.0396039	1	-1.0658034
2.9	0.0392968	1	-3.9011404
3.0	0.0389946	1	-3.3943996
3.1	0.0386969	1	0.1952533
3.2	0.0384037	1	0.2297535
3.3	0.0381150	1	1.9783379
3.4	0.0378305	1	-0.8003786
3.5	0.0375503	1	-2.2960912
3.6	0.0372742	1	-2.2927641
3.7	0.0370021	1	2.2369301
3.8	0.0367340	1	-1.0686733
3.9	0.0364697	1	-3.9039675
4.0	0.0362092	1	-3.3971849
4.1	0.0359524	1	0.1925088
4.2	0.0356992	1	0.2270490
4.3	0.0354496	1	1.9756725
4.4	0.0352034	1	-0.8030057
4.5	0.0349606	1	-2.2986809
4.6	0.0347212	1	-2.2953171
4.7	0.0344850	1	2.2344130
4.8	0.0342520	1	-1.0711553
4.9	0.0340221	1	-3.9064152
5.0	0.0337953	1	-3.3995989
5.1	0.0335715	1	0.1901279
5.2	0.0333506	1	0.2247004
5.3	0.0331326	1	1.9733555
5.4	0.0329175	1	-0.8052917
5.5	0.0327051	1	-2.3009364
5.6	0.0324955	1	-2.2975428
5.7	[illegible]	1	[illegible]

```
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 $\hat{\beta}_0^{ridge}$.

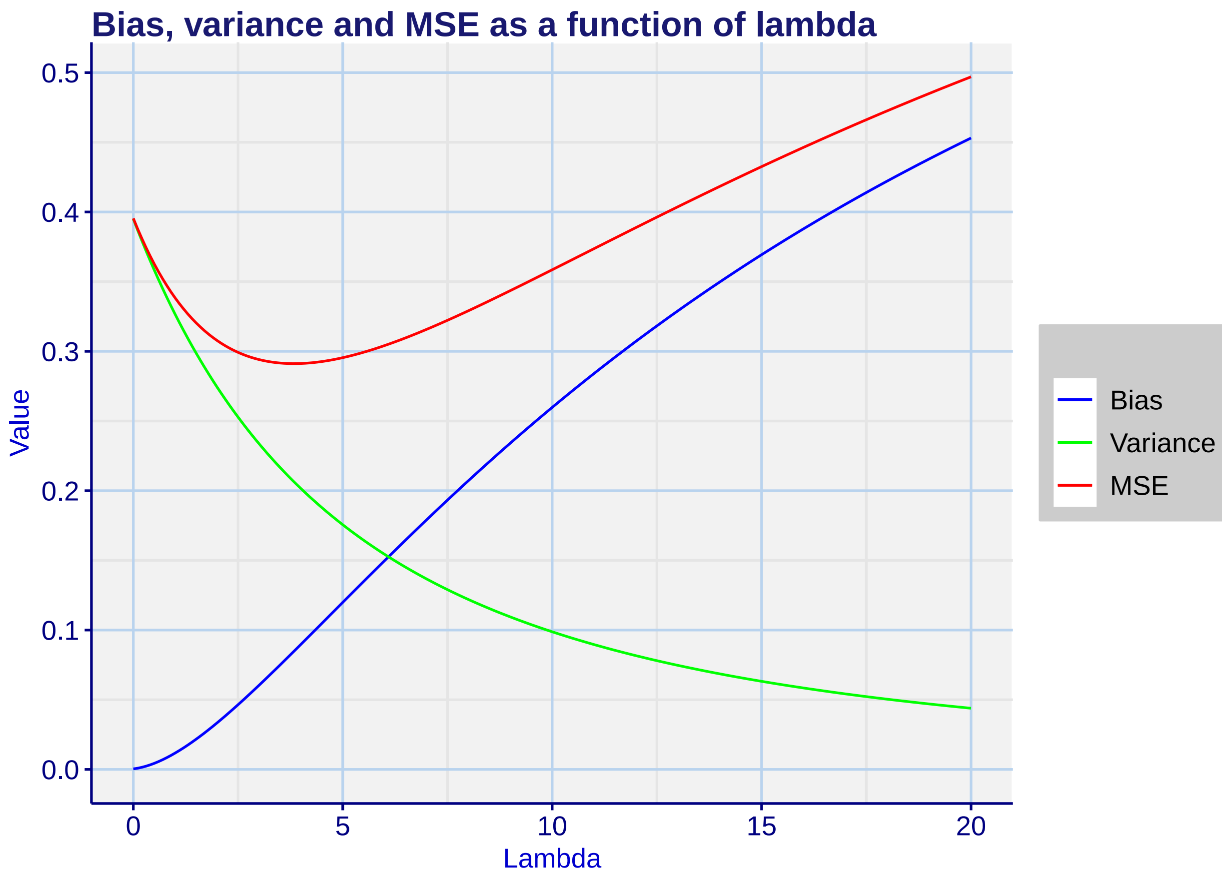
```
library(purrr)
df_1 <- map_df(seq(1,1000,1), ~{
  results = simulate_ridge(n=10)
  tibble(
    lambda = results$lambda,
    beta_hat = results$beta_hat,
    error = 1 - results$beta_hat
  )


}) %>% 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:

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

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

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

From this, it is trivial to see that $u_1^2Var(X)+u_2^2Var(Y)$ 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

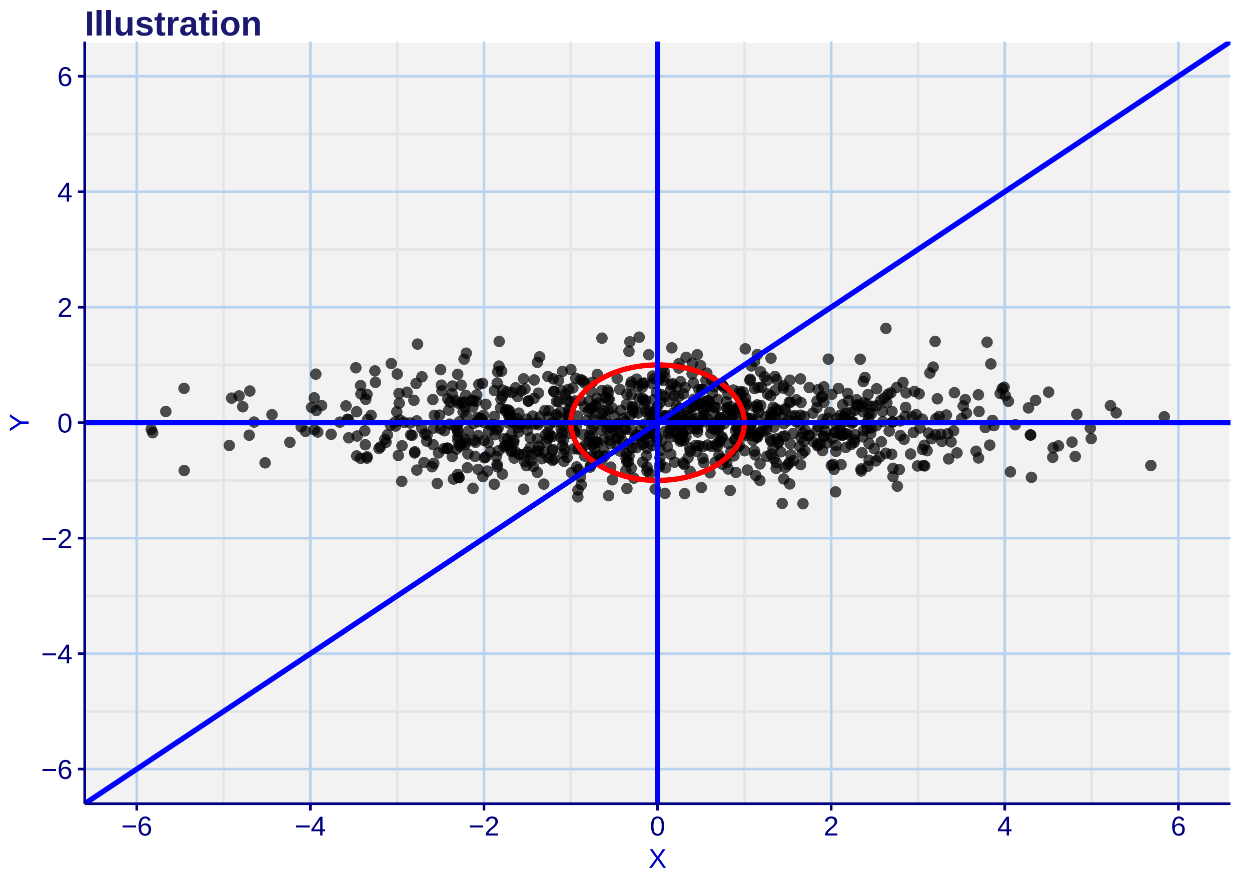
```
set.seed(20220307)
x<-rnorm(1000,mean=0,sd=2)
set.seed(43293)
y<-rnorm(1000,mean=0,sd=0.5)
# Covariance is almost zero
cov(x,y)
```

```
## [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:

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

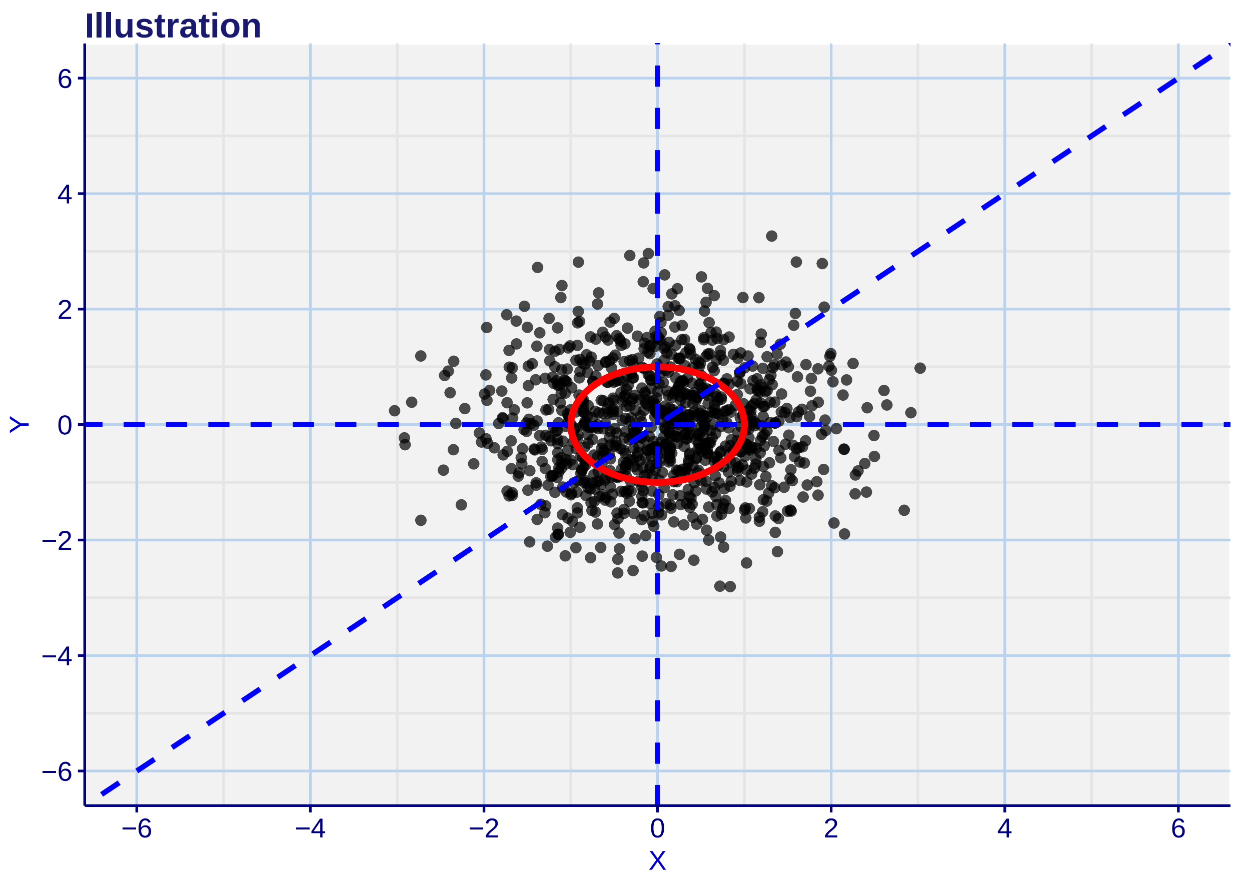
So regardless of the (u_1, u_2) values of the unit vector, the expression will be maximized and its value will be 1 because of the $(u_1^2 + u_2^2 = 1)$ condition. Intuitively, becasue of the equal variance, the X,Y points will form a circle around their mean, and in each direction of a unit vector,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

```
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
```

```
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