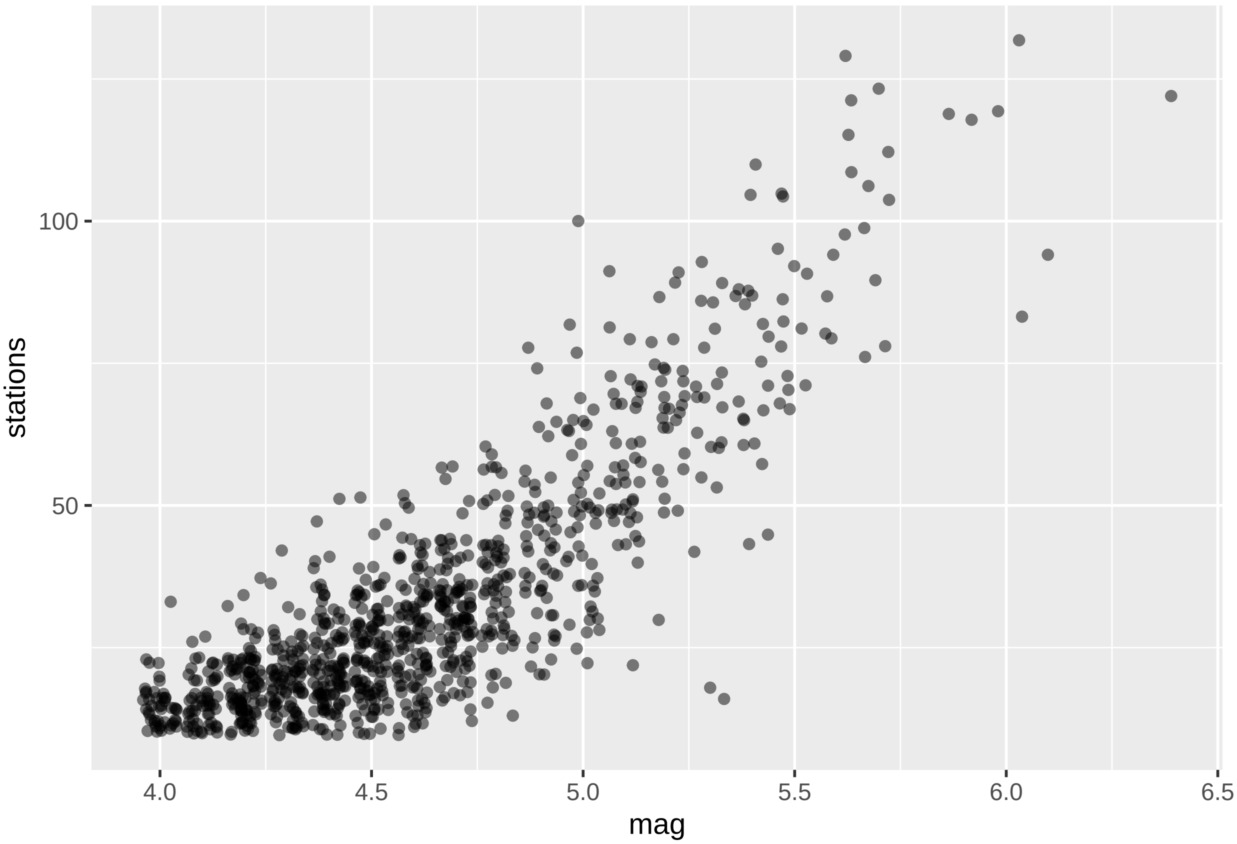
Lab 2: Linear Regression

An Island Never Cries

YILIN Li

Exercise 1

```r
library(ggplot2)
p1 = ggplot(data = quakes, aes(x = mag, y = stations)) +
  geom_jitter(alpha = 0.5)
plot(p1)
```



From the graph above, I think number of stations which report the Earthquakes is positively correlated to the magnitude of the Earthquake.

Exercise 2

```r
b0 = mean(quakes$stations)
b1 = 0
b1
```

```
## [1] 0
```

```
b0
```

```
## [1] 33.418
```

If in fact there's no relationship between magnitude of Earthquakes and the number of stations that report it, then the slope will be 0 and the intercept will be the average value of the number of the stations, which is around 33.

Exercise 3

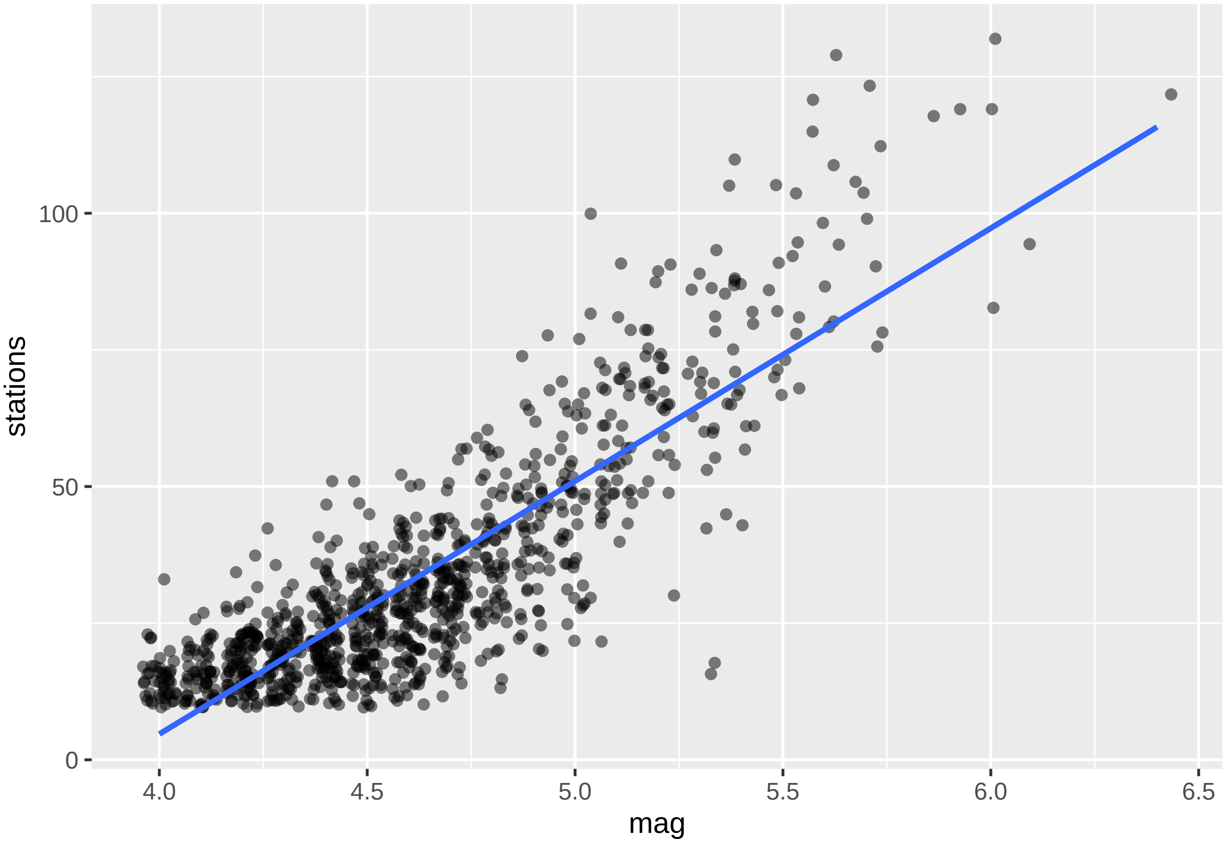
```
m1 = lm(stations ~ mag, data = quakes)
b0 = m1$coef[1]
b1 = m1$coef[2]
p2 = p1 + geom_smooth(method = 'lm', se = FALSE)
b0
```

```
## (Intercept)
##   -180.4243
```

```
b1
```

```
##      mag
## 46.28221
```

```
plot(p2)
```



1. The intercept is -180.4243, meaning if the magnitude of the Earthquake is 0, then about -180 stations will report. Even without practical meaning at a first glance, the negative value of the intercept indicates that no station will report if there's no Earthquake.

2. The slope is 46.28221, which means that as the magnitude of the Earthquake increases for 1 level, about 46 more stations will report the Earthquake.

Exercise 4

```
x = quakes$mag
y = quakes$stations
slope = (sd(y)/sd(x))*cor(x,y)
(b1 - slope) < 0.0001
```

```
##  mag
## TRUE
```

lm() function calculates the same value of slope as we calculate using the formula.

Exercise 5

```
n = nrow(quakes)
slope = m1$coefficients[2]
SE_slope = summary(m1)$coef[2,2]
t_stat = qt(0.025, df= n-2)

LB = slope + t_stat * SE_slope
UP = slope - t_stat * SE_slope
LB - confint(m1)[2,1] < 0.0001
```

```
##  mag
## TRUE
```

```
UP - confint(m1)[2,2] < 0.0001
```

```
##  mag
## TRUE
```

Two methods indeed construct the same confidence interval.

Exercise 6

```
pred_stations =predict(m1, data.frame(mag = 7.0))
pred_stations
```

```
##        1
## 143.5511
```

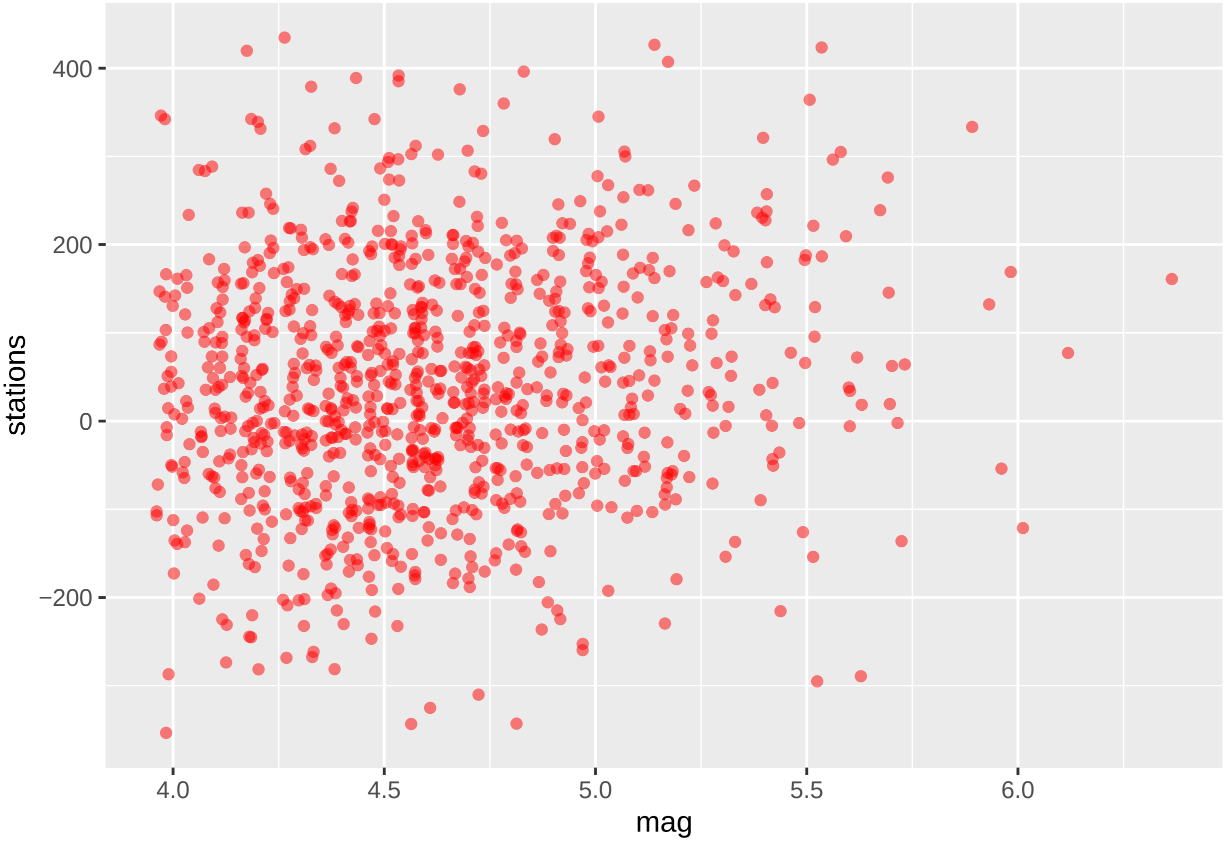
If the magnitude is 7.0, then about 144 stations will report the Earthquake.

Exercise 7

1. data description by plotting and inference of the relation bewteen magnitude and stations by checking the plot.
2. Inference
3. Inference by explaining the meaning of intercept and slope.
4. Inference.
5. Inference.
6. Prediction of future Earthquakes.

Simulation

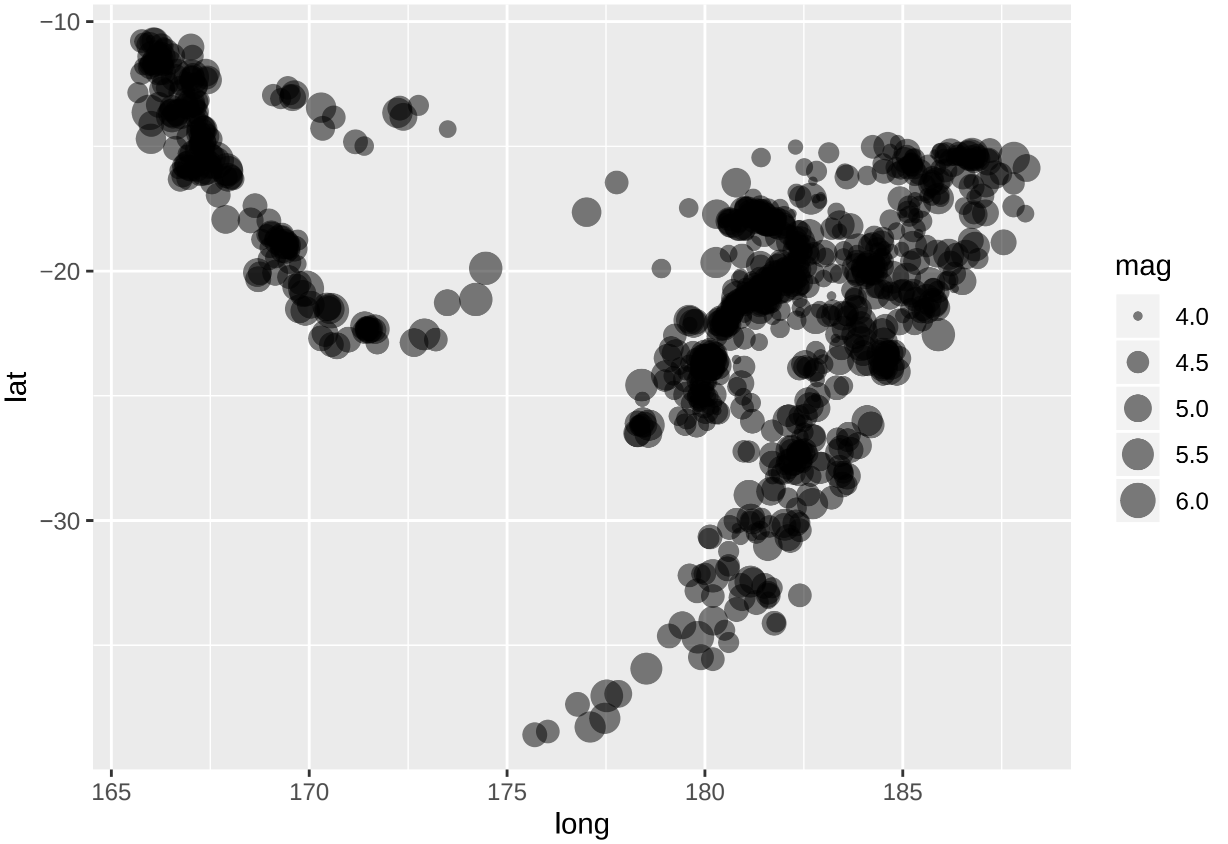
```
x = quakes$mag
f_hat = function(x){
  predict(m1, data.frame(mag = x))
}
y_hat = f_hat(x)
sigma_square = sum(m1$res^2)/(n - 2)
y = y_hat + rnorm(n, mean = 0, sd = sigma_square)
newdata = data.frame('mag' = x, 'stations' = y)
ggplot(data = newdata, aes(x = mag, y = stations)) +
  geom_jitter(alpha = 0.5, color = 'red')
```



Actually, it's not quite similar to the original data. The simulated data seems losing the linearity between magnitude and stations that report it. I think one of the reason is caused by the large residuals from the model, which create a large fluctuation of the irreducible error term. One solution could be adding mag^2 term to the model and fitting it.

Challenge Problem

```
ggplot(data = quakes, aes(x = long, y = lat, size = mag)) +
  geom_point(alpha = 0.5)
```



Problem Set 2

Chapter 3 exercises

Exercise 1

The null hypothesis for p-value of "TV" is that given the advertisement of radio and newspaper, TV has no effect on sales. The null hypothesis for p-value of "radio" is that given the advertisement of TV and newspaper, radio has no effect on sales.The null hypothesis for p-value of "newspaper" is that given the advertisement of radio and TV, newspaper has no effect on sales. We can draw a conclusion that given the advertising bugets of TV, radio and newspaper, it suggests that TV and radio have some relation with sales, but newspaper has no relation with sales.

Exercise 4

a. In fact, the cubic regression will have a lower training RSS rather than linear regression, because cubic regression is a more flexible model than linear regression. Therefore, with the same dataset, the cubic regression will fit with a lower RSS.
b. Estimate the cubic regression model with a test set will generate a higher RSS rather than the linear model because the cubic model will overfit the data and thus unable to reflect the true relationship.
c. In fact, the cubic regression will have a lower training RSS no matter what the true relationship is, because the cubic regression is more flexible than the linear regression. Thus, the cubic regression will have a fitting line closer to the data points, resulting a lower training RSS.
d. There is not enough information to tell which one has the lower testing RSS. If the true relationship is closer to the linear regression than the cubic regression, then the linear regression will have the lower

testing RSS. If the true relationship is closer to the cubic regression, then the cubic regression will have the lower testing RSS.

Exercise 5

$$\hat{y_i} = x_i\hat{\beta} = x_i \frac{\sum_{j=1}^{n} x_j y_j}{\sum_{k=1}^{n} x_k^2} = \sum_{j=1}^{n} \frac{x_i x_j}{\sum_{k=1}^{n} x_k^2} y_j = \sum_{j=1}^{n} a_j y_j$$

Thus we know $a_j = \frac{x_i x_j}{\sum_{k=1}^{n} x_k^2}$ where x_i is the corresponding input value for a given predicted value.

Additional exercises

The k-nearest neighbor regression was defined as:

$$\hat{f}(x) = \frac{1}{k} \sum_{x_i \in \mathcal{N}(x)} y_i$$

We have derived from class that:

$$MSE_{test} = Var(\hat{f(x_0)}) + [E(f(x_0) - \hat{f(x_0)})]^2 + Var(\epsilon)$$

Thus, the variance of KNN regression model is:

$$\begin{aligned}
Var(\hat{f(x_0)}) &= Var(\frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} y_i) \\
&= \frac{1}{k^2} Var(\sum_{x_i \in \mathcal{N}(x_0)} y_i) \\
&= \frac{1}{k^2} Var(\sum_{x_i \in \mathcal{N}(x_0)} (f(x_i) + \epsilon_i)) \\
&= \frac{1}{k^2} Var(\sum_{x_i \in \mathcal{N}(x_0)} \epsilon_i) && x_0 \text{ has been fixed.} \\
&= \frac{1}{k^2} |\mathcal{N}(x_0)| \sigma_\epsilon^2 \\
&= \frac{\sigma_\epsilon^2}{k}
\end{aligned}$$

Then, the $Bias^2$ of KNN regression model is:

$$\begin{aligned}
Bias^2(\hat{f}(x_0)) &= [E(f(x_0) - \hat{f(x_0)})]^2 \\
&= [E(f(x_0) - \frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} y_i)]^2 \\
&= [E(f(x_0) - \frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} (f(x_i) + \epsilon))]^2 \\
&= [f(x_0) - \frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} f(x_i)]^2
\end{aligned}$$

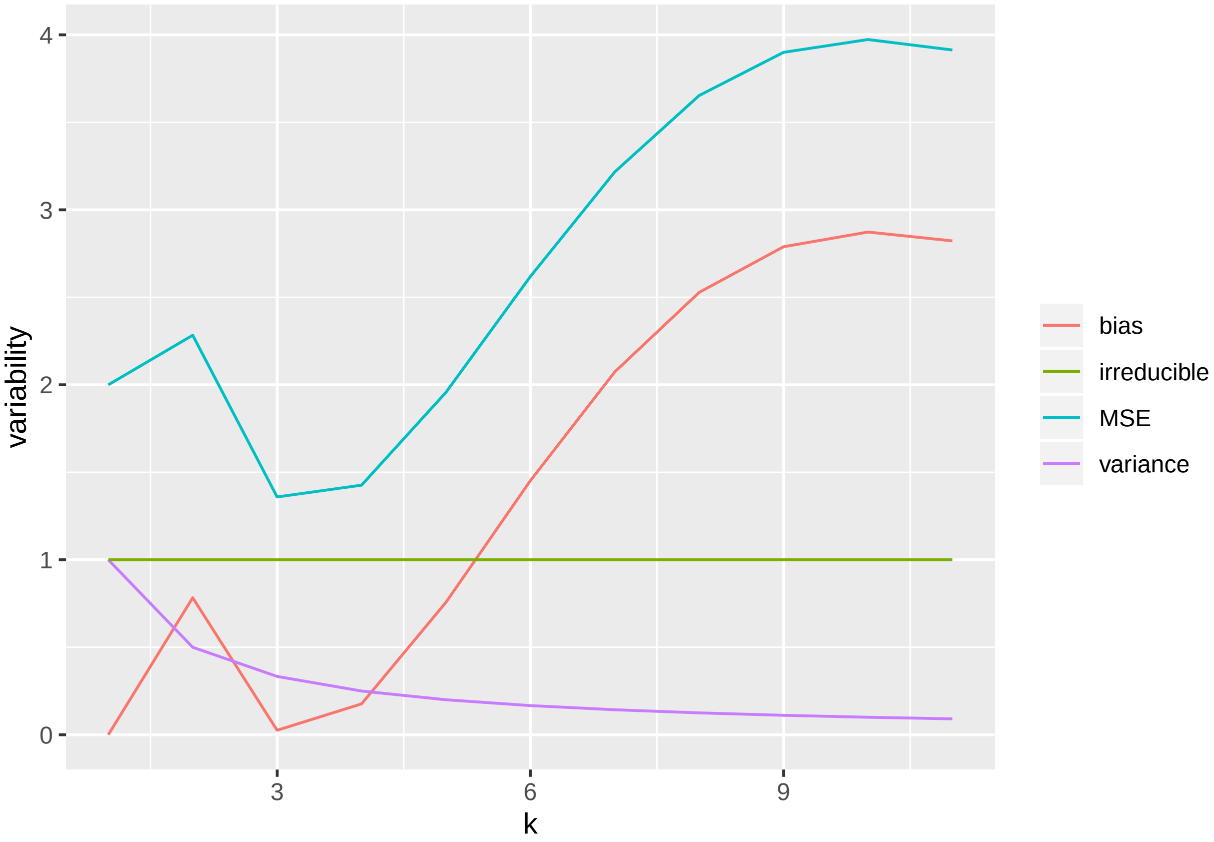
Therefore, we derived the MSE_{test} as follows:

$$\begin{aligned} MSE_{test} &= Var(f(\hat{x_0})) + Bias^2(\hat{f}(x_0)) + Var(\epsilon) \\ &= \frac{\sigma_\epsilon^2}{k} + [f(x_0) - \frac{1}{k}\sum_{x_i \in \mathcal{N}(x_0)} f(x_i)]^2 + \sigma_\epsilon^2 \end{aligned}$$

And we're given the true function to be:

$$f(x) = -9.3 + 2.6x - 0.3x^2 + .01x^3$$

```r
library(ggplot2)
x <- c(1:3, 5:12)
y <- c(-7.1, -7.1, .5, -3.6, -2, -1.7, -4, -.2, -1.2, -1.2, -3.5)

#true function f(x)
true_func <- function(x){
  -9.3 + 2.6*x - 0.3*x^2 + .01*x^3
}

#bias_sq calculate the bias^2 based on test data(x0,y0) and number of nearest neighbors k
bias_sq <- function(k,x0,x) {
  (true_func(x0) - sum(true_func(sort(abs(x - x0),index.return=TRUE)$ix[1:k])/k))^2
}

bias = rep(NA,11)
for (k in 1:11){
  bias[k] = bias_sq(k,2,x) # Assume the test data is (2.0, 1.0)
}

k = 1:11
sigma2 = 1  # Assume variance of the irreducible error term is 1
variance = sigma2/k

ggplot()+
  geom_line(aes(x = k, y = bias, color = "bias"))+
  geom_line(aes(x = k, y = variance, color = "variance"))+
  geom_line(aes(x = k, y = sigma2, color = "irreducible" ))+
  geom_line(aes(x = k, y = bias + variance + sigma2, color = "MSE"))+
  theme(legend.title = element_blank())+
  ylab("variability")
```


4
3
2
1
0
variability
3
6
9
k
bias
irreducible
MSE
variance