Linear Models and Regression Analysis Homework 5

Yifei Dong

9/30/2020

0. Data Preparation

```
setwd("/Users/yifei/Documents/Teachers College/Linear Models and Regression/Week 5/hw5")
getwd()
```

```
## [1] "/Users/yifei/Documents/Teachers College/Linear Models and Regression/Week 5/hw5"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(tinytex)
library(latex2exp)
```

1. Brand Preference

In a small-scale experimental study of the relation between degree of brand liking (Y) and moisture conten (X_1) and sweetness (X_2) of the product, the following results were obtained from the experiment based on a completely randomized design (data are coded):

(a) Obtain the scatter plot matrix and the correlation matrix. What information do these diagnostic aids provide here?

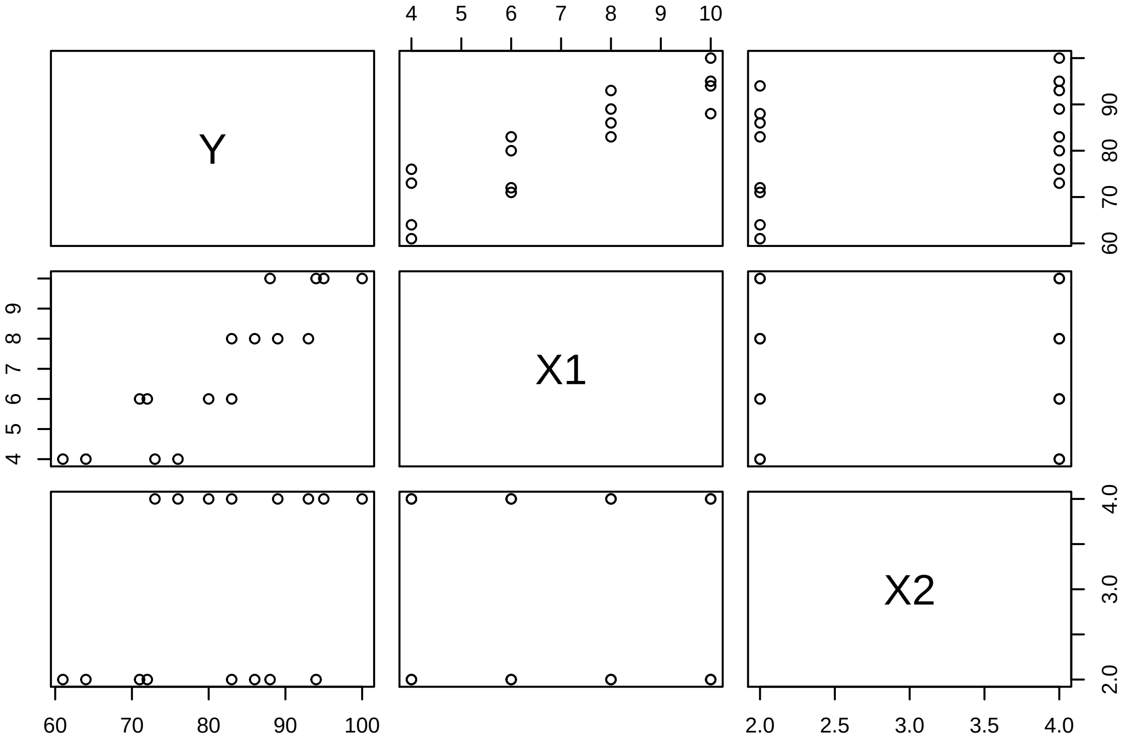
```
# Load data
mydata <- read.table(paste("http://users.stat.ufl.edu/~rrandles",
                           "/sta4210/Rclassnotes/data/textdatasets",
                           "/KutnerData/Chapter%20%206%20Data%20Sets/CH06PR05.txt",
                           sep = ""))
# Rnames variables use dplyr package
mydata <- mydata %>%
  rename("Y"=V1, "X1"=V2, "X2"=V3)
mydata
```

```
##      Y X1 X2
## 1   64  4  2
## 2   73  4  4
## 3   61  4  2
## 4   76  4  4
## 5   72  6  2
## 6   80  6  4
## 7   71  6  2
## 8   83  6  4
## 9   83  8  2
## 10  89  8  4
## 11  86  8  2
## 12  93  8  4
## 13  88 10  2
## 14  95 10  4
## 15  94 10  2
## 16 100 10  4
```

```
# Obtain scatter plot matrix and correlation matrix
attach(mydata)
cor(mydata)
```

```
##            Y        X1        X2
## Y  1.0000000 0.8923929 0.3945807
## X1 0.8923929 1.0000000 0.0000000
## X2 0.3945807 0.0000000 1.0000000
```

```
pairs(mydata)
```



The correlation matrix is:

$$\begin{array}{c} Y \\ X_1 \\ X_2 \end{array} \begin{bmatrix} 1.000 & 0.892 & 0.395 \\ 0.892 & 1.000 & 0.000 \\ 0.395 & 0.000 & 1.000 \end{bmatrix} \tag{1}$$

ANSWER: Based on the correlation matrix, there exists a strong positive correlation between Y and X_1, which is 0.892. Also, there exists a positve correlation between Y and X_2, but the relationship is not strong, which is 0.395. Finnaly, there is no correlation between X_1 and X_2.

(b) Fit regression model (6.1) to the data. State the estimated regression function. How is b_1 interpreted here?

```r
reg <- lm(Y~X1+X2)
summary(reg)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.400 -1.762  0.025  1.587  4.200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.6500     2.9961  12.566 1.20e-08 ***
## X1            4.4250     0.3011  14.695 1.78e-09 ***
## X2            4.3750     0.6733   6.498 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.693 on 13 degrees of freedom
## Multiple R-squared:  0.9521, Adjusted R-squared:  0.9447
## F-statistic: 129.1 on 2 and 13 DF,  p-value: 2.658e-09
```

$b_0 = 37.650$, $b_1 = 4.425$, $b_2 = 4.375$, then the estimated regression function is:

$$\widehat{Y} = 37.650 + 4.425X_1 + 4.375X_2 \tag{2}$$

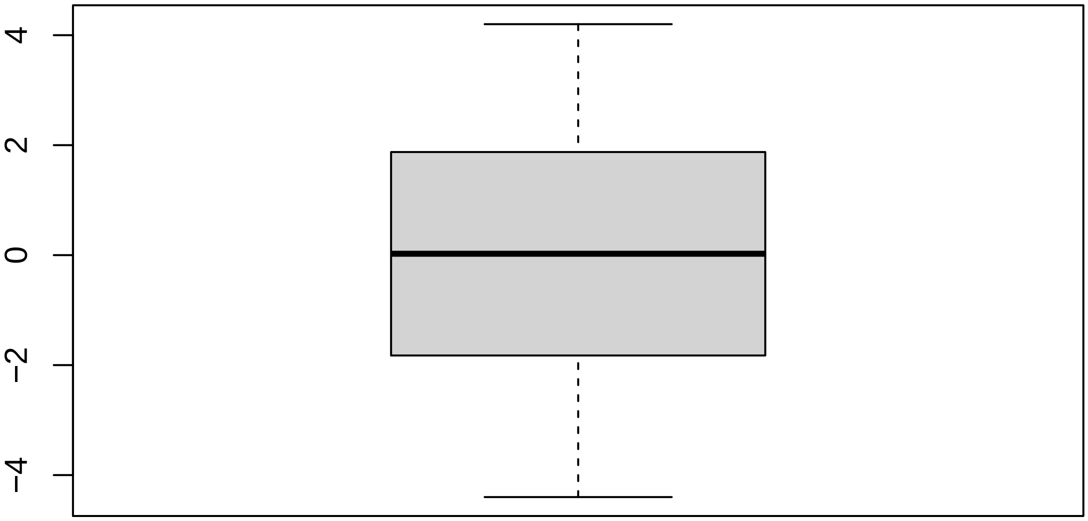
Interpretation of b_1: For every 1 unit increase in moisture content of the product, the degree of brand liking will increase by 4.425 units, keeping sweetness of the product fixed.

(c) Obtain the residuals and prepare a box plot of the residuals. What information does this plot provide?

```
# Obtain residuals
e <- residuals(reg)
e
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## -0.10  0.15 -3.10  3.15 -0.95 -1.70 -1.95  1.30  1.20 -1.55  4.20  2.45 -2.65
##    14    15    16
## -4.40  3.35  0.60
```

```
# Box plot of the residuals
boxplot(e)
```



ANSWER: Based on the box plot of the residuals, the normality assumption of residuals seems hold and there are no outliers.

(d) Plot the residuals against $\widehat{Y}, X_1, X_2$, and X_1X_2 on separate graphs. Also prepare a normal probability plot. Interpret the plots and summarize your findings.

```
# Obtain fitted values
y_hat <- predict(reg)
mydata <- cbind(mydata, e, y_hat)
```

```
# Diagnostic Plots
# Residuals against Y_hat
g1 <- ggplot(mydata, aes(x = y_hat, y = e))+geom_point(color = "black") + xlab("Fitted")+
  ylab("Residual")+ggtitle(TeX("(a) Residual Plot against $\\widehat{Y}$"))+
  geom_hline(yintercept = 0, linetype = "dashed", color="red")
```

```
# Residuals against X1
g2 <- ggplot(mydata, aes(x = X1, y = e)) + geom_point(color = "black")+xlab(TeX("X_{1}"))+
  ylab("Residual")+ggtitle(TeX("(b) Residual Plot against X_{1}"))+
  geom_hline(yintercept = 0, linetype="dashed", color="red")
```

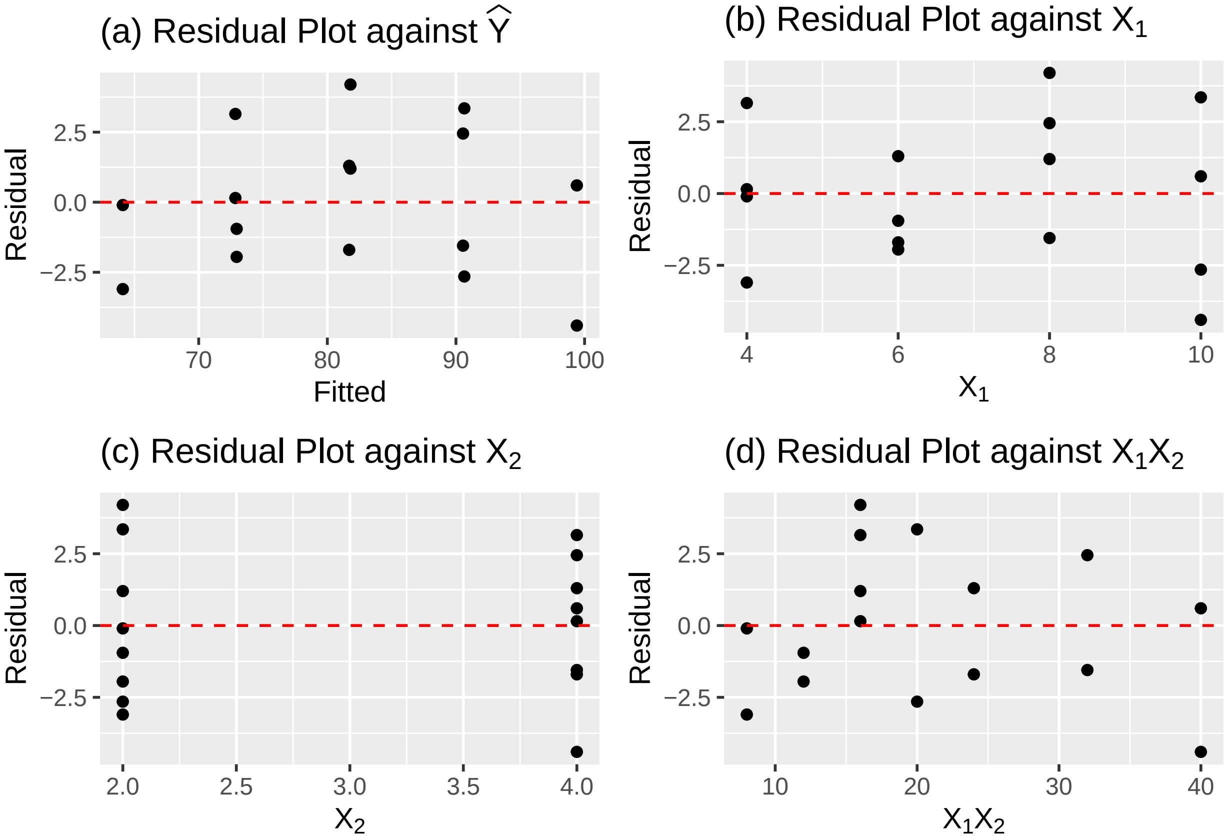
```
# Residuals against X2
g3 <- ggplot(mydata, aes(x = X2, y = e)) + geom_point(color = "black")+xlab(TeX("X_{2}"))+
  ylab("Residual")+ggtitle(TeX("(c) Residual Plot against X_{2}"))+
  geom_hline(yintercept = 0, linetype="dashed", color="red")
```

```
# create interaction term first
mydata <- mydata %>%
  mutate(mydata, X1X2 = X1*X2)
head(mydata, 10)
```

```
##     Y X1 X2     e y_hat X1X2
## 1  64  4  2 -0.10 64.10    8
## 2  73  4  4  0.15 72.85   16
## 3  61  4  2 -3.10 64.10    8
## 4  76  4  4  3.15 72.85   16
## 5  72  6  2 -0.95 72.95   12
## 6  80  6  4 -1.70 81.70   24
## 7  71  6  2 -1.95 72.95   12
## 8  83  6  4  1.30 81.70   24
## 9  83  8  2  1.20 81.80   16
## 10 89  8  4 -1.55 90.55   32
```

```
# Residuals against X1X2
g4 <- ggplot(mydata, aes(x = X1X2, y = e))+geom_point(color = "black")+xlab(TeX("X_{1}X_{2}"))+
  ylab("Residual")+ggtitle(TeX("(d) Residual Plot against X_{1}X_{2}"))+
  geom_hline(yintercept = 0, linetype="dashed", color="red")
```

```
# Combine 4 graphs together
grid.arrange(g1, g2, g3, g4, ncol =2, nrow =2)
```



(a) Residual Plot against $\widehat{Y}$

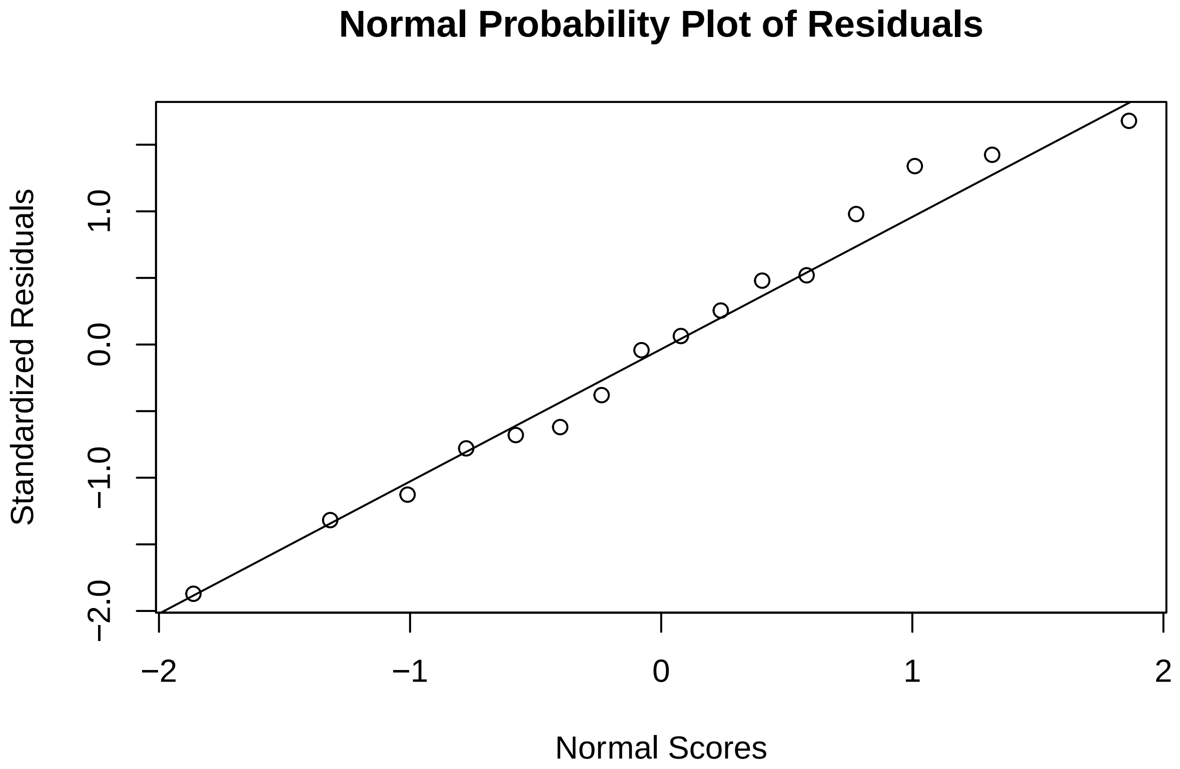
(b) Residual Plot against X_1

(c) Residual Plot against X_2

(d) Residual Plot against X_1X_2

ANSWER: The plot (a) of residuals e against the fitted values $\widehat{Y}$ does not suggest any systematic deviations from the reponse plane nor the variance of the error terms varies with the level of $\widehat{Y}$. Plots of the residuals e against X_1 and X_2, in Figure (b) and (c), respectively, are entirely consistent with the conclusions of fit by the reponse function and constant variance of the error terms. There is also no systematic pattern for the residuals e against the interaction term X_1X_2. Hence, no interaction effects reflected.

```
# Normal Probability Plot
# We first compute the standardized with the rstandard function first
stdres <- rstandard(reg)
qqnorm(stdres, xlab="Normal Scores", ylab="Standardized Residuals", main =
         "Normal Probability Plot of Residuals")
qqline(stdres)
```



```
# Extra check of normality assumption use Shapiro-Wilk normality test
shapiro.test(stdres)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.97496, p-value = 0.9111
```

ANSWER: Based on the normal probability plot of residuals, the normality assumption does hold. We can also use Shapiro-Wilk normality test to provide additional evidence for the normality. The p-value = 0.9111, which is greater than 0.05. We cannot reject the null hypothesis, therefore, the normality assumption holds.

(e) Conduct the Breusch-Pagan test for constancy of the error variance, assumig $log\ \sigma_i^2 = \gamma_0 + \gamma_1 X_{i1} + \gamma_2 X_{i2}$; Use $\alpha = .01$ State the alternatives, decision rule, and conclusion.

```
bptest(reg)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg
## BP = 2.0441, df = 2, p-value = 0.3599
```

ANSWER: P-value=0.3599 is not less than 0.01. Do not reject H_0: Homoskedasticity. Conclude error variance constant.

2. Matrix Methods

Assume that regression model (6.1) with indepdent normal error terms is appropriate. Using matrix methods, obtain

```
# Load data
mydata2 <- read.table(paste("http://users.stat.ufl.edu/~rrandles",
                            "/sta4210/Rclassnotes/data/textdatasets/KutnerData",
                            "/Chapter%20%206%20Data%20Sets/CH06PR27.txt", sep=""))
mydata2 <- mydata2 %>%
  rename("y"=V1, "x1"=V2, "x2"=V3)
mydata2
```

```
##    y x1 x2
## 1 42  7 33
## 2 33  4 41
## 3 75 16  7
## 4 28  3 49
## 5 91 21  5
## 6 55  8 31
```

(a) b

```
# Creating the observations y and design matrix X
Y <- mydata2$y
X <- cbind(rep(1, nrow(mydata2)), mydata2$x1, mydata2$x2)
```

```
# X'X
t(X) %*% X
```

```
##      [,1] [,2] [,3]
## [1,]    6   59  166
## [2,]   59  835 1007
## [3,]  166 1007 6206
```

```
# X'Y
t(X) %*% Y
```

```
##      [,1]
## [1,]  324
## [2,] 4061
## [3,] 6796
```

```
# b = (X'X)^(-1)X'Y
b <- solve(t(X) %*% X) %*% t(X) %*% Y
b
```

```
##             [,1]
## [1,] 33.9321033
## [2,]  2.7847614
## [3,] -0.2644189
```

$$\begin{bmatrix} 33.93210 \\ 2.78476 \\ -0.26442 \end{bmatrix} \tag{3}$$

(b) e

```
# sample size
n <- nrow(mydata2)
# Identity Matrix
I <- diag(rep(1, n))
# H Matrix = X(X'X)^(-1)X'
H <- X %*% solve(t(X) %*% X) %*% t(X)
# Matrix Method for residuals
(I-H) %*% Y
```

```
##             [,1]
## [1,] -2.69960842
## [2,] -1.22997279
## [3,] -1.63735316
## [4,] -1.32985996
## [5,] -0.08999801
## [6,]  6.98679233
```

$$\begin{bmatrix} -2.6996 \\ -1.2300 \\ -1.6374 \\ -1.3299 \\ -0.0900 \\ 6.9868 \end{bmatrix} \tag{4}$$

(c) H

```
# H Matrix = X(X'X)^(-1)X'
H <- X %*% solve(t(X) %*% X) %*% t(X)
H
```

```
##             [,1]        [,2]        [,3]       [,4]        [,5]       [,6]
## [1,]  0.23143293  0.25167585  0.21178735  0.1488684 -0.05475543 0.21099091
## [2,]  0.25167585  0.31240459  0.09437844  0.2662773 -0.14787283 0.22313666
## [3,]  0.21178735  0.09437844  0.70442026 -0.3191744  0.10446672 0.20412159
## [4,]  0.14886839  0.26627729 -0.31917435  0.6142563  0.14143492 0.14833743
## [5,] -0.05475543 -0.14787283  0.10446672  0.1414349  0.94039955 0.01632707
## [6,]  0.21099091  0.22313666  0.20412159  0.1483374  0.01632707 0.19708635
```

$$\begin{bmatrix} .2314 & .2517 & .2118 & .1489 & -.0548 & .2110 \\ .2517 & .3124 & .0944 & .2663 & -.1479 & .2231 \\ .2118 & .0944 & .7044 & -.3192 & .1045 & .2041 \\ .1489 & .2663 & -.3192 & .6143 & .1414 & .1483 \\ -.0548 & -.1479 & .1045 & .1414 & .9404 & .0163 \\ .2110 & .2231 & .2041 & .1483 & .0163 & .1971 \end{bmatrix} \tag{5}$$

(d) SSR

```
# J matrix
J <- matrix(rep(1, n^2), n, n)
# SSR = Y'(H-(1/n)J)Y
SSR <- t(Y) %*% (H-(J/n)) %*% Y
SSR
```

```
##           [,1]
## [1,] 3009.926
```

(e) $s^2\{b\}$

```
# SSE = e'e = Y'Y-Y'Xb
SSE <- t(Y) %*% Y - t(Y) %*% X %*% b
MSE <- SSE/(n-3)
# cov(b)=MSE(X'X)^(-1)
s.sq.b <- drop(MSE) * solve(t(X) %*% X)
s.sq.b
```

```
##             [,1]        [,2]        [,3]
## [1,] 715.47114 -34.1589166 -13.5949371
## [2,] -34.15892   1.6616664   0.6440674
## [3,] -13.59494   0.6440674   0.2624678
```

$$\begin{bmatrix} 715.4711 & -34.1589 & -13.5949 \\ -34.1589 & 1.6617 & .6441 \\ -13.5949 & .6441 & .2625 \end{bmatrix} \tag{6}$$

(f) $\widehat{Y_h}$ when $X_{h1} = 10$, $X_{h2} = 30$

```
x.new <- c(1, 10, 30)
y.hat.h <- t(x.new) %*% b
y.hat.h
```

```
##           [,1]
## [1,] 53.84715
```

(g) $s^2\{\widehat{Y_h}\}$ when when $X_{h1} = 10$, $X_{h2} = 30$

```
s.sq.y.hat.h <- drop(MSE)*(t(x.new) %*% solve(t(X) %*% X) %*% x.new)
s.sq.y.hat.h
```

```
##          [,1]
## [1,] 5.42462
```