first assignment for 7003

2024-09-21

- **student no:303624496**
- **student name: Liu sheng**

Q1

(a)

- iii is correct
- We can find that the function is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5$$
$$x_4 = x_1 * x_2$$
$$x_5 = x_1 * x_3$$

- so
- $$\frac{\partial(salary)}{\partial(level)} = 35 - 10x_1$$
- we can find that when the gpa is high enough(x>3.5),high school graduate earn more.

(b)

- $$y = 50 + 20X_1 + 0.07X_2 + 35X_3 + 0.01X_4 + (-10)X_5$$
- when IQ=110 and GPA=4.0
- $$Y = 50 + 20 * 4.0 + 0.07 * 110 + 35 * 1 + 0.01 * 110 * 4.0 - 10 * 4.0$$

```
Y<-50+ 20 * 4.0 + 0.07 * 110 + 35 *  1+ 0.01*110*4.0-10*4.0
```

- when IQ=110 and GPA=4.0, a college graduate's predicted salary is 137100 dollars

(c)

- False. When we judge whether the model has an interaction effect, we should pay attention to whether the p-value of the interaction coefficient is credible within a certain level.

Q2

Modified according to the warm tips of dear professor

(a)&(b)

- suppose that the true relationship between X and Y is linear.
- Through the following simulation, we can find that when x and y satisfy a linear relationship, adding quadratic and cubic terms will reduce the RSS. However, if we do not know the original data, we cannot predict the change of RSS.

```
# help to reproduce results
set.seed(12)

## Generate random numbers
x <- rnorm(1:100, mean = 8, sd = 16)  # x must be in order, otherwise the lines will not be connected in order.
y <- 10.7+2*x+rnorm(100,mean=0,sd=20)
data<-data.frame(x = x, y = y)

## fit different regression model
model_linear<-lm(y~x) #model1 <- lm(y ~ x + I(x^2))
model_cubic<- lm(y ~ poly(x, 3), data = data) #model2<-lm(y~x+I(x^2)+I(x^3))

#Calculate different RSS
#sum(resid(model_linear)^2)
#sum(resid(model_cubic)^2)

# Get the R-squared values for the models
linear_summary <- summary(model_linear)
cubic_summary <- summary(model_cubic)

# Extract R-squared values
linear_r_squared <- linear_summary$r.squared
cubic_r_squared <- cubic_summary$r.squared

# Predict using the training data
linear_predict <- predict(model_linear, data)
cubic_predict <- predict(model_cubic, data)

# Calculate MSE on training data
linear_mse <- mean((data$y - linear_predict)^2)
cubic_mse <- mean((data$y - cubic_predict)^2)

# Print R-squared values
cat("R-squared for the linear model:", linear_r_squared, "\n")
```

```
## R-squared for the linear model: 0.6626485
```

```
cat("R-squared for the cubic model:", cubic_r_squared, "\n")
```

```
## R-squared for the cubic model: 0.6746818
```

```
cat("Train MSE for the linear model:", linear_mse, "\n")
```

```
## Train MSE for the linear model: 395.1851
```

```
cat("Train MSE for the cubic model:", cubic_mse, "\n")
```

```
## Train MSE for the cubic model: 381.0889
```

- multiple simulations

```r
#modify my original code to equation form
# Load libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```
# Simulate data
simulate_data <- function(n) {
  x <- rnorm(n, mean = 8, sd = 16)
  y <- 10.7 + 2 * x + rnorm(n, mean = 100, sd = 20)
  data.frame(x = x, y = y)
}

# Simulate a dataset
set.seed(123)
data <- simulate_data(100)

## fit different regression model
model_linear<-lm(y~x,data = data)
model_cubic<- lm(y ~ poly(x, 3), data = data)

# Number of iterations and sample size
n_iter <- 1000
sample_size <- 100

simulate_iteration <- function(sample_size) {
  t_data <- simulate_data(sample_size)

  linear_test_predictions <- predict(model_linear, t_data)
  cubic_test_predictions <- predict(model_cubic, t_data)

  linear_test_mse <- mean((t_data$y - linear_test_predictions)^2)
  cubic_test_mse <- mean((t_data$y - cubic_test_predictions)^2)

  return(data.frame(
    linear_test_mse = linear_test_mse,
    cubic_test_mse = cubic_test_mse
  ))
}

# Simulate multiple iterations
results <- map_dfr(1:n_iter, ~simulate_iteration(sample_size))

# Count the number of times linear MSE is smaller than cubic MSE
linear_better <- mean(results$linear_test_mse < results$cubic_test_mse)
linear_better_in_sample<- mean(linear_mse < cubic_mse)
cat("linear MSE is smaller than cubic MSE in sample: ", linear_better_in_sample,
"\n")
```

```
## linear MSE is smaller than cubic MSE in sample:  0
```

```
cat("Proportion of times linear MSE is smaller than cubic MSE in out_of_sample: ", li
near_better, "\n")
```

```
## Proportion of times linear MSE is smaller than cubic MSE in out_of_sample:  0.769
```

(c)&(d)

- suppose that the true relationship between X and Y is not linear.

- Through the following simulation, we can find that when x and y do not satisfy the linear relationship, adding quadratic and cubic terms will make the RSS drop more significantly. When it is known that the original variables do not satisfy the linear conditions, we expect that adding square and cubic terms will improve the model.

```
# Load libraries
library(tidyverse)
# Simulate data
simulate_data <- function(n) {
  x <- rnorm(n, mean = 8, sd = 16)
  y <- 10.7+x^2+rnorm(n,mean=0,sd=20)
  data.frame(x = x, y = y)
}

# Simulate a dataset
set.seed(123)
data <- simulate_data(100)

## fit different regression model
model_linear<-lm(y~x,data = data)
model_cubic<- lm(y ~ poly(x, 3), data = data)

# Get the R-squared values for the models
linear_summary <- summary(model_linear)
cubic_summary <- summary(model_cubic)

# Extract R-squared values
linear_r_squared <- linear_summary$r.squared
cubic_r_squared <- cubic_summary$r.squared

# Predict using the training data
linear_predict <- predict(model_linear, data)
cubic_predict <- predict(model_cubic, data)

# Calculate MSE on training data
linear_mse <- mean((data$y - linear_predict)^2)
cubic_mse <- mean((data$y - cubic_predict)^2)

# Print R-squared values
cat("R-squared for the linear model:", linear_r_squared, "\n")
```

```
## R-squared for the linear model: 0.5028718
```

```
cat("R-squared for the cubic model:", cubic_r_squared, "\n")
```

```
## R-squared for the cubic model: 0.9977616
```

```
cat("Train MSE for the linear model:", linear_mse, "\n")
```

```
## Train MSE for the linear model: 81072.71
```

```
cat("Train MSE for the cubic model:", cubic_mse, "\n")
```

```
## Train MSE for the cubic model: 365.0487
```

```
# Number of iterations and sample size
n_iter <- 1000
sample_size <- 100

simulate_iteration <- function(sample_size) {
  t_data <- simulate_data(sample_size)

  linear_test_predictions <- predict(model_linear, t_data)
  cubic_test_predictions <- predict(model_cubic, t_data)

  linear_test_mse <- mean((t_data$y - linear_test_predictions)^2)
  cubic_test_mse <- mean((t_data$y - cubic_test_predictions)^2)

  return(data.frame(
    linear_test_mse = linear_test_mse,
    cubic_test_mse = cubic_test_mse
  ))
}

# Simulate multiple iterations
results <- map_dfr(1:n_iter, ~simulate_iteration(sample_size))

# Count the number of times linear MSE is smaller than cubic MSE
linear_better <- mean(results$linear_test_mse < results$cubic_test_mse)
linear_better_in_sample<- mean(linear_mse < cubic_mse)
cat("linear MSE is smaller than cubic MSE in sample: ", linear_better_in_sample,
"\n")
```

```
## linear MSE is smaller than cubic MSE in sample:  0
```

```
cat("Proportion of times linear MSE is smaller than cubic MSE in out_of_sample: ", li
near_better, "\n")
```

```
## Proportion of times linear MSE is smaller than cubic MSE in out_of_sample:  0
```

Q3

(a)

- this is the multiple model by R

```
data <- read.csv("https://raw.githubusercontent.com/kwan-MSDA/MSDA7003/main/Carseats.
csv")
###
y<-data$Sales
#table(data$Urban)
x1<-data$Price
x2<-as.factor(data$Urban)
x3<-as.factor(data$US)
modelsales<-lm(y~x1+x2+x3)
summary(modelsales)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## x1          -0.054459   0.005242 -10.389  < 2e-16 ***
## x2Yes       -0.021916   0.271650  -0.081    0.936
## x3Yes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

(b)

- x1:show that sales will decrease(-0.054459) when the prices go up.this result is statistically reliable at the 99 percent confidence level.

(c)

- this is the equation form:
- $$y = 13.04 - 0.05x_1 - 0.02x_2 + 1.20x_3 + \epsilon$$

$$f(x) = \begin{cases} 13.04 - 0.05 * x1 - 0.02 * x_2 + 1.20 * x_3 + \epsilon & \text{if } x_2 = 1 \quad x_3 = 1 \\ 13.04 - 0.05 * x1 - 0.02 * x_2 + \epsilon & \text{if } x_2 = 1 \quad x_3 = 0 \\ 13.04 + 1.20 * x_3 + \epsilon & \text{if } x_2 = 0 \quad x_3 = 1 \\ 13.04 - 0.05 * x1 + \epsilon & \text{if } x_2 = 0 \quad x_3 = 0 \end{cases}$$

(d)

- null hypothesis of x1: $\beta_1 = 0$
- null hypothesis of X3: $\beta_3 = 0$
- We can reject the null hypothesis of x1 and x3 at a significance level of 0.01 (p<0.01)

(e)

- According to the previous regression results, removing x2 and the new model is as follows:

```
modelsales1<-lm(y~x1+x3)
summary(modelsales1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## x1          -0.05448    0.00523 -10.416  < 2e-16 ***
## x3Yes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

(f)

- Model (e) is superior to model (a) in terms of the overall explanation of the whole model (r^2) and the significance of the parameters(p-value).

(g)

- we can calculate confidence intervals:

```
confint(modelsales1, level = 0.95)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## x1          -0.06475984 -0.04419543
## x3Yes        0.69151957  1.70776632
```

Q4

```
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100) / 10
y <- 2+2*x1+0.3*x2+rnorm(100)
```

(a)

$$y = \beta_0 + \beta_1 x1 + \beta_2 x2 + \epsilon$$

- the regression coefficients is :

$$\begin{cases} \beta_0 = 2 \\ \beta_1 = 2 \\ \beta_3 = 0.3 \end{cases}$$

(b)

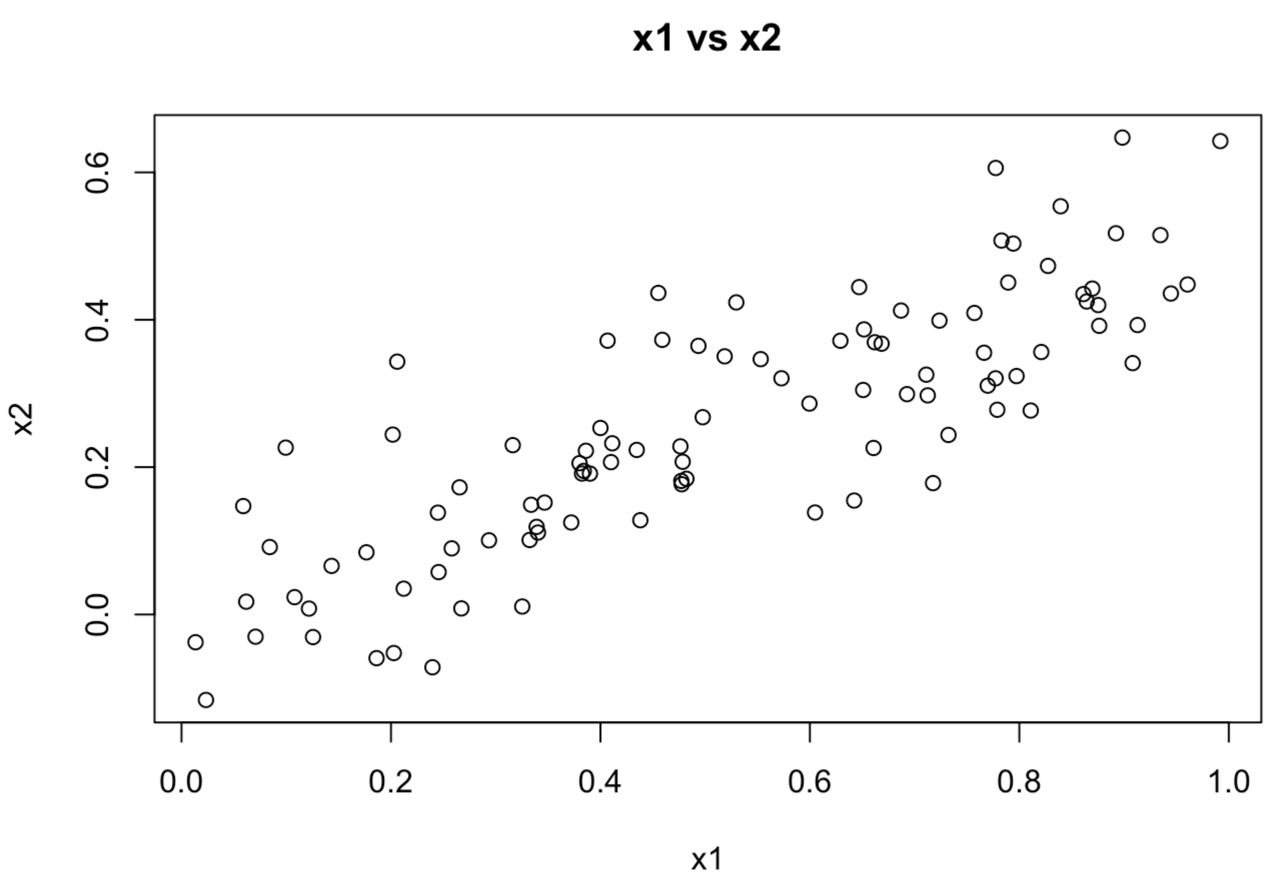
- correlation:

```
cor(x1, x2, method = "pearson")
```

```
## [1] 0.8351212
```

- plot:

```
plot(x1,x2)
title(main = "x1 vs x2")
```



(c)

```
model4c <- lm(y ~ x1+x2)
summary(model4c)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

- the regression coefficients:

$$\begin{cases} \widetilde{\beta}_0 = 2.1305 \\ \widetilde{\beta}_1 = 1.4396 \\ \widetilde{\beta}_2 = 1.0097 \end{cases}$$

- We found that the coefficients of the regression results are the same as the original construction coefficients
  1.$\widetilde{\beta}_0$ We reject the null hypothesis with 99.9 percent confidence.(p<0.01) 2.$\widetilde{\beta}_2$ we can not reject the null hypothesis

(d)

```
model4d <- lm(y ~ x1)
summary(model4d)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

- yes, we can(p<0.1)

(e)

```
model4e <- lm(y ~ x2)
summary(model4e)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

- yes,we can(p<0.1)

(f)

- No contradiction.
- In the original model, there is a linear relationship between x1 and x2. This will lead to multicollinearity problems between x1 and x2 in model (c). This affects the interpretability of the model.

(g)

- The newly added values affect the model in many ways (parameters, significance of parameter estimates, model interpretability)
- In the entire model, this observation is an outlier， We can find out from the scatter plot

```
x1<-c(x1,0.1)
x2<-c(x2,0.8)
y<-c(y,6)

model4g <- lm(y ~ x1+x2)
summary(model4g)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
model4g1 <- lm(y ~ x1)
summary(model4g1)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
model4g2 <- lm(y ~ x2)
summary(model4g2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

```
plot(x1,x2)
```