BACS HW12

109090046

2023-05-02

Create a `data.frame` called `cars_log` with log-transformed columns for `mpg`, `weight`, and `acceleration` (`model_year` and `origin` don't have to be transformed)

```
auto <- read.table("D:/下載/auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleratio
n", "model_year", "origin", "car_name")

# Log-transform columns
auto$log_mpg <- log(auto$mpg)
auto$log_weight <- log(auto$weight)
auto$log_acceleration <- log(auto$acceleration)

# Create a new data.frame with the log-transformed columns and non-transformed columns
cars_log <- auto[, c("log_mpg", "log_weight", "log_acceleration", "model_year", "origin")]

# Display the new cars_log data.frame
head(cars_log)
```

```
##    log_mpg log_weight log_acceleration model_year origin
## 1 2.890372   8.161660         2.484907         70      1
## 2 2.708050   8.214194         2.442347         70      1
## 3 2.890372   8.142063         2.397895         70      1
## 4 2.772589   8.141190         2.484907         70      1
## 5 2.833213   8.145840         2.351375         70      1
## 6 2.708050   8.375860         2.302585         70      1
```

Question 1) Let's visualize how weight and acceleration are related to mpg.

a. Let's visualize how weight might moderate the relationship between acceleration and mpg:

i. Create two subsets of your data, one for light-weight cars (less than mean weight) and one for heavy cars (higher than the mean weight)

HINT: consider carefully how you compare log weights to mean weight

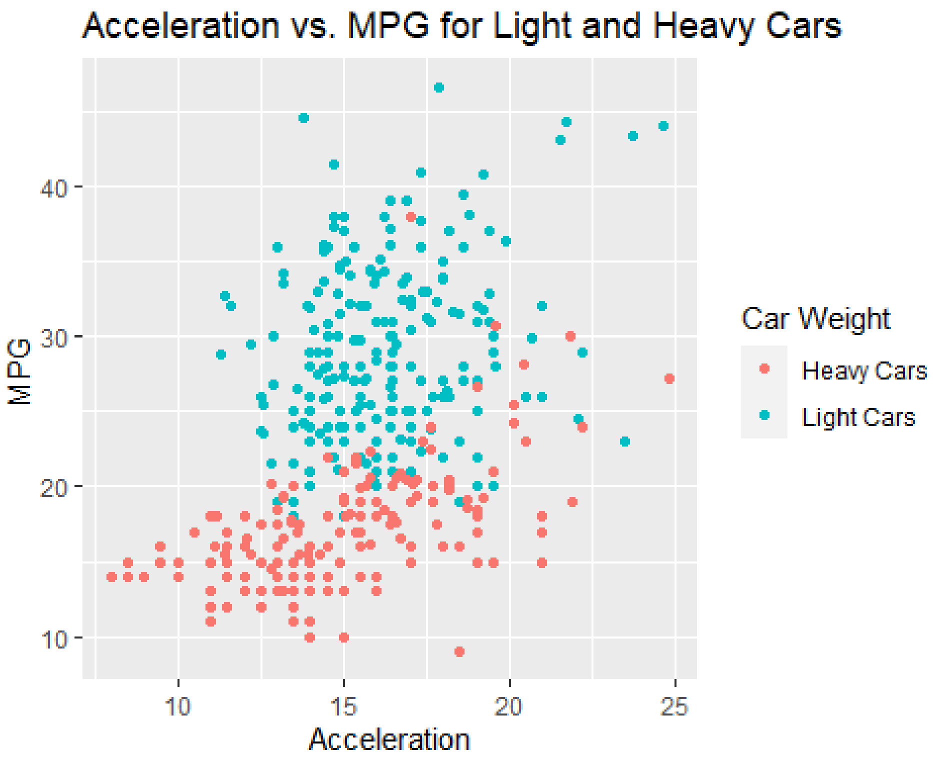
```
# Calculate mean weight
mean_weight <- mean(auto$weight, na.rm = TRUE)

# Create subsets for light and heavy cars
light_cars <- auto[auto$weight < mean_weight, ]
heavy_cars <- auto[auto$weight >= mean_weight, ]
```

ii. Create a single scatter plot of acceleration vs. mpg, with different colors and/or shapes for light versus heavy cars

```
# Create a scatter plot with different colors/shapes for light and heavy cars
scatter_plot <- ggplot() +
  geom_point(data = light_cars, aes(x = acceleration, y = mpg, color = "Light Cars")) +
  geom_point(data = heavy_cars, aes(x = acceleration, y = mpg, color = "Heavy Cars")) +
  labs(title = "Acceleration vs. MPG for Light and Heavy Cars", x = "Acceleration", y = "
MPG", color = "Car Weight") +
  theme_grey()

scatter_plot
```



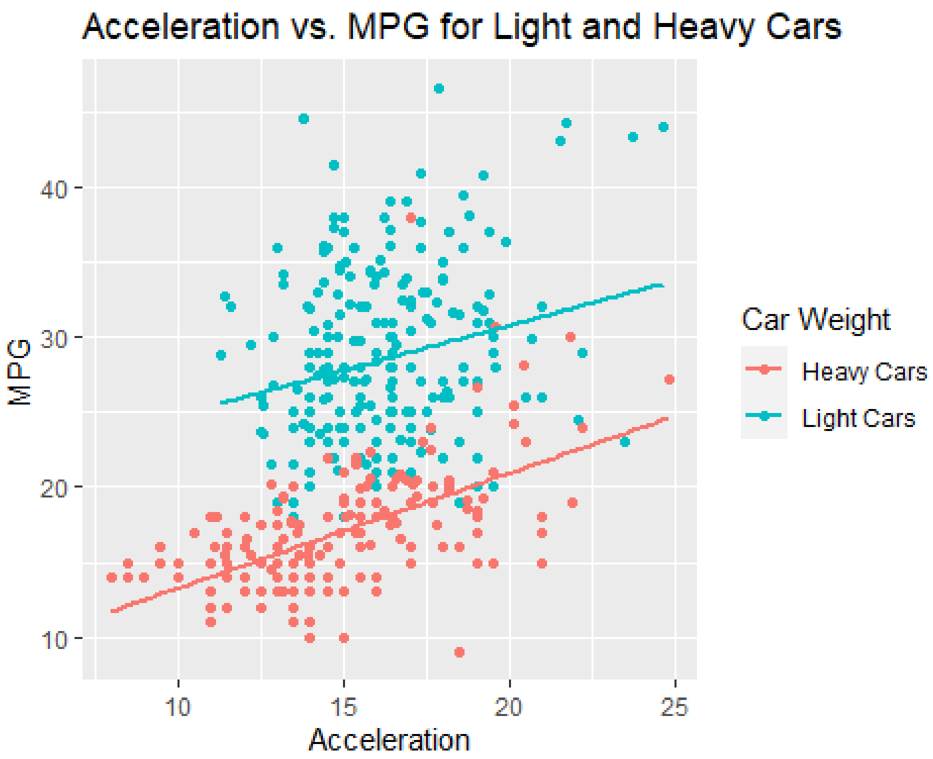
iii. Draw two slopes of acceleration-vs-mpg over the scatter plot:

one slope for light cars and one slope for heavy cars (distinguish them by appearance)

```
# Add slopes to the scatter plot
scatter_plot_with_slopes <- scatter_plot +
  geom_smooth(data = light_cars, aes(x = acceleration, y = mpg, color = "Light Cars"), me
thod = "lm", se = FALSE) +
  geom_smooth(data = heavy_cars, aes(x = acceleration, y = mpg, color = "Heavy Cars"), me
thod = "lm", se = FALSE)

scatter_plot_with_slopes

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



b. Report the full summaries of two separate regressions for light and heavy cars where `log.mpg.` is dependent on `log.weight.`, `log.acceleration.`, `model_year` and `origin`

```
# Run separate regressions for light and heavy cars
regression_light_cars <- lm(log_mpg ~ log_weight + log_acceleration + model_year + origin,
 data = light_cars)
regression_heavy_cars <- lm(log_mpg ~ log_weight + log_acceleration + model_year + origin,
 data = heavy_cars)

# Report regression summaries
summary(regression_light_cars)

##
## Call:
## lm(formula = log_mpg ~ log_weight + log_acceleration + model_year +
##     origin, data = light_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37941 -0.07219 -0.00307  0.06759  0.34454
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.059570   0.526938  13.397   <2e-16 ***
## log_weight       -0.849942   0.056655 -15.002   <2e-16 ***
## log_acceleration  0.108295   0.056775   1.907   0.0578 .
## model_year        0.032895   0.001951  16.858   <2e-16 ***
## origin            0.012824   0.009310   1.377   0.1698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1121 on 222 degrees of freedom
## Multiple R-squared:  0.7233, Adjusted R-squared:  0.7183
## F-statistic: 145.1 on 4 and 222 DF,  p-value: < 2.2e-16

summary(regression_heavy_cars)

##
## Call:
## lm(formula = log_mpg ~ log_weight + log_acceleration + model_year +
##     origin, data = heavy_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36811 -0.06937  0.00607  0.06969  0.43736
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.097038   0.762942   9.302  < 2e-16 ***
## log_weight       -0.822352   0.077206 -10.651  < 2e-16 ***
## log_acceleration  0.040140   0.057380   0.700   0.4852
## model_year        0.030317   0.003573   8.486 1.14e-14 ***
## origin            0.091641   0.040392   2.269   0.0246 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1212 on 166 degrees of freedom
## Multiple R-squared:  0.7179, Adjusted R-squared:  0.7111
## F-statistic: 105.6 on 4 and 166 DF,  p-value: < 2.2e-16
```

c. (not graded) Using your intuition only: What do you observe about light versus heavy cars so far?

ANS:

They both have the trend that as MPG get higher, then the acceleration get larger. However, almost all the light cars with same acceleration with heavy car have higher MPG than heavy cars. From my speculation, I think it’s because lighter cars must be fuel-saving than heavier cars.

Question 2) Use the transformed dataset from above (cars_log), to test whether we have moderation.

a. (not graded) Considering weight and acceleration, use your intuition and experience to state which of the two variables might be a moderating versus independent variable, in affecting mileage.

ANS:

Based on intuition and experience, weight may be the moderating variable because it affects how the relationship between acceleration and mileage (mpg) changes. In this case, acceleration is the independent variable.

b. Use various regression models to model the possible moderation on `log.mpg.`: (use `log.weight.`, `log.acceleration.`, `model_year` and `origin` as independent variables)

i. Report a regression without any interaction terms

```
model_no_interaction <- lm(log_mpg ~ log_weight + log_acceleration + model_year + origin,
 data = cars_log)
summary(model_no_interaction)
```

```
##
## Call:
## lm(formula = log_mpg ~ log_weight + log_acceleration + model_year +
##     origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39581 -0.07037  0.00014  0.06984  0.39638
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.539281   0.314707  23.956   <2e-16 ***
## log_weight       -0.889384   0.028466 -31.243   <2e-16 ***
## log_acceleration  0.062145   0.036679   1.694   0.0910 .
## model_year        0.032106   0.001690  18.999   <2e-16 ***
## origin            0.018352   0.009165   2.002   0.0459 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1164 on 393 degrees of freedom
## Multiple R-squared:  0.8836, Adjusted R-squared:  0.8825
## F-statistic: 746.1 on 4 and 393 DF,  p-value: < 2.2e-16
```

ii. Report a regression with an interaction between weight and acceleration

```
model_interaction <- lm(log_mpg ~ log_weight * log_acceleration + model_year + origin, da
ta = cars_log)
summary(model_interaction)
```

```
##
## Call:
## lm(formula = log_mpg ~ log_weight * log_acceleration + model_year +
##     origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38147 -0.06870  0.00120  0.06595  0.39570
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.773573   2.763699   0.642   0.5214
## log_weight                  -0.179842   0.339101  -0.530   0.5962
## log_acceleration             2.162941   1.001155   2.160   0.0313 *
## model_year                   0.032933   0.001728  19.057   <2e-16 ***
## origin                       0.016595   0.009164   1.811   0.0709 .
## log_weight:log_acceleration -0.261526   0.124550  -2.100   0.0364 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1159 on 392 degrees of freedom
## Multiple R-squared:  0.8849, Adjusted R-squared:  0.8835 
## F-statistic:   603 on 5 and 392 DF,  p-value: < 2.2e-16
```

iii. Report a regression with a mean-centered interaction term

```
# Mean-center log_weight and log_acceleration
cars_log$log_weight_centered <- cars_log$log_weight - mean(cars_log$log_weight)
cars_log$log_acceleration_centered <- cars_log$log_acceleration - mean(cars_log$log_accel
eration)

# Regression with mean-centered interaction term
model_mean_centered_interaction <- lm(log_mpg ~ log_weight_centered * log_acceleration_ce
ntered + model_year + origin, data = cars_log)
summary(model_mean_centered_interaction)
```

```
## 
## Call:
## lm(formula = log_mpg ~ log_weight_centered * log_acceleration_centered + 
##     model_year + origin, data = cars_log)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.38147 -0.06870  0.00120  0.06595  0.39570 
## 
## Coefficients:
##                                                 Estimate Std. Error t value
## (Intercept)                                     0.566397   0.132258   4.283
## log_weight_centered                            -0.893616   0.028415 -31.448
## log_acceleration_centered                       0.082003   0.037725   2.174
## model_year                                      0.032933   0.001728  19.057
## origin                                          0.016595   0.009164   1.811
## log_weight_centered:log_acceleration_centered -0.261526   0.124550  -2.100
##                                                Pr(>|t|)    
## (Intercept)                                    2.33e-05 ***
## log_weight_centered                             < 2e-16 ***
## log_acceleration_centered                        0.0303 *  
## model_year                                      < 2e-16 ***
## origin                                           0.0709 .  
## log_weight_centered:log_acceleration_centered   0.0364 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1159 on 392 degrees of freedom
## Multiple R-squared:  0.8849, Adjusted R-squared:  0.8835 
## F-statistic:   603 on 5 and 392 DF,  p-value: < 2.2e-16
```

iv. Report a regression with an orthogonalized interaction term

```
# Orthogonalize interaction term
cars_log$log_weight_ortho <- residuals(lm(log_weight_centered ~ log_acceleration_centered,
 data = cars_log))
cars_log$log_acceleration_ortho <- residuals(lm(log_acceleration_centered ~ log_weight_ce
ntered, data = cars_log))

# Regression with orthogonalized interaction term
model_orthogonalized_interaction <- lm(log_mpg ~ log_weight_ortho * log_acceleration_orth
o + model_year + origin, data = cars_log)
summary(model_orthogonalized_interaction)

##
## Call:
## lm(formula = log_mpg ~ log_weight_ortho * log_acceleration_ortho +
##     model_year + origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39409 -0.07078 -0.00030  0.07041  0.39596
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              0.630230   0.129838   4.854 1.75e-06
## log_weight_ortho                        -1.107053   0.032816 -33.735  < 2e-16
## log_acceleration_ortho                   0.794986   0.043711  18.187  < 2e-16
## model_year                               0.032138   0.001705  18.850  < 2e-16
## origin                                   0.018287   0.009186   1.991   0.0472
## log_weight_ortho:log_acceleration_ortho -0.021669   0.142650  -0.152   0.8793
##
## (Intercept)                             ***
## log_weight_ortho                        ***
## log_acceleration_ortho                  ***
## model_year                              ***
## origin                                  *
## log_weight_ortho:log_acceleration_ortho
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1166 on 392 degrees of freedom
## Multiple R-squared:  0.8836, Adjusted R-squared:  0.8822
## F-statistic: 595.4 on 5 and 392 DF,  p-value: < 2.2e-16
```

c.For each of the interaction term strategies above (raw, mean-centered, orthogonalized) what is the correlation between that interaction term and the two variables that you multiplied together?

```
# Calculate interaction terms
cars_log$raw_interaction <- cars_log$log_weight * cars_log$log_acceleration
cars_log$mean_centered_interaction <- cars_log$log_weight_centered * cars_log$log_acceler
ation_centered
cars_log$orthogonalized_interaction <- cars_log$log_weight_ortho * cars_log$log_accelerat
ion_ortho

# Correlations between interaction terms and the multiplied variables
cor_raw_interaction_weight <- cor(cars_log$log_weight, cars_log$raw_interaction)
cor_raw_interaction_acceleration <- cor(cars_log$log_acceleration, cars_log$raw_interacti
on)

cor_mean_centered_interaction_weight <- cor(cars_log$log_weight_centered, cars_log$mean_c
entered_interaction)
cor_mean_centered_interaction_acceleration <- cor(cars_log$log_acceleration_centered, car
s_log$mean_centered_interaction)

cor_orthogonalized_interaction_weight <- cor(cars_log$log_weight_ortho, cars_log$orthogon
alized_interaction)
```

```
cor_orthogonalized_interaction_acceleration <- cor(cars_log$log_acceleration_ortho, cars_
log$orthogonalized_interaction)

## Raw interaction term correlations:

## Weight: 0.1083055

## Acceleration: 0.852881

## Mean-centered interaction term correlations:

## Weight: -0.2026948

## Acceleration: 0.3512271

## Orthogonalized interaction term correlations:

## Weight: 0.06876088

## Acceleration: 0.2254387
```

Question 3) We saw earlier that the number of cylinders does not seem to directly influence mpg when car weight is also considered. But might cylinders have an indirect relationship with mpg through its weight?

Let's check whether weight *mediates* the relationship between cylinders and mpg, even when other factors are controlled for. Use `log.mpg.`, `log.weight.`, and `log.cylinders` as your main variables, and keep `log.acceleration.`, `model_year`, and `origin` as control variables (see gray variables in diagram).

a. Let's try computing the direct effects first:

i. Model 1: Regress `log.weight.` over `log.cylinders.` only (check whether number of cylinders has a significant direct effect on weight)

```
# Add log_cylinders to the dataset
cars_log$log_cylinders <- log(auto$cylinders)

# Model 1: Regress log_weight over log_cylinders only
model1 <- lm(log_weight ~ log_cylinders, data = cars_log)
summary(model1)

##
## Call:
## lm(formula = log_weight ~ log_cylinders, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35473 -0.09076 -0.00147  0.09316  0.40374
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.60365    0.03712  177.92   <2e-16 ***
## log_cylinders  0.82012    0.02213   37.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1329 on 396 degrees of freedom
## Multiple R-squared:  0.7762, Adjusted R-squared:  0.7757
## F-statistic:  1374 on 1 and 396 DF,  p-value: < 2.2e-16
```

ii.

Model 2: Regress `log.mpg.` over `log.weight.` and all control variables (check whether weight has a significant direct effect on mpg with other variables statistically controlled)

```
# Model 2: Regress log_mpg over log_weight and all control variables
model2 <- lm(log_mpg ~ log_weight + log_cylinders + log_acceleration + model_year + origi
n, data = cars_log)
summary(model2)
```

```
##
## Call:
## lm(formula = log_mpg ~ log_weight + log_cylinders + log_acceleration +
##     model_year + origin, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41853 -0.06596 -0.00220  0.07017  0.40633
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.277558   0.352735  20.632   <2e-16 ***
## log_weight       -0.831621   0.045426 -18.307   <2e-16 ***
## log_cylinders    -0.071282   0.043745  -1.630   0.1040
## log_acceleration  0.044747   0.038127   1.174   0.2413
## model_year        0.031716   0.001703  18.621   <2e-16 ***
## origin            0.016341   0.009228   1.771   0.0774 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1162 on 392 degrees of freedom
## Multiple R-squared:  0.8844, Adjusted R-squared:  0.883
## F-statistic: 599.9 on 5 and 392 DF,  p-value: < 2.2e-16
```

b.

What is the indirect effect of cylinders on mpg? (use the product of slopes between Models 1 & 2)

```
indirect_effect <- coef(model1)["log_cylinders"] * coef(model2)["log_weight"]
```

```
## Indirect effect: -0.682032
```

c. Let's bootstrap for the confidence interval of the indirect effect of cylinders on mpg

i. Bootstrap regression models 1 & 2, and compute the indirect effect each time: What is its 95% CI of the indirect effect of log.cylinders. on log.mpg.?

```
library(boot)
```

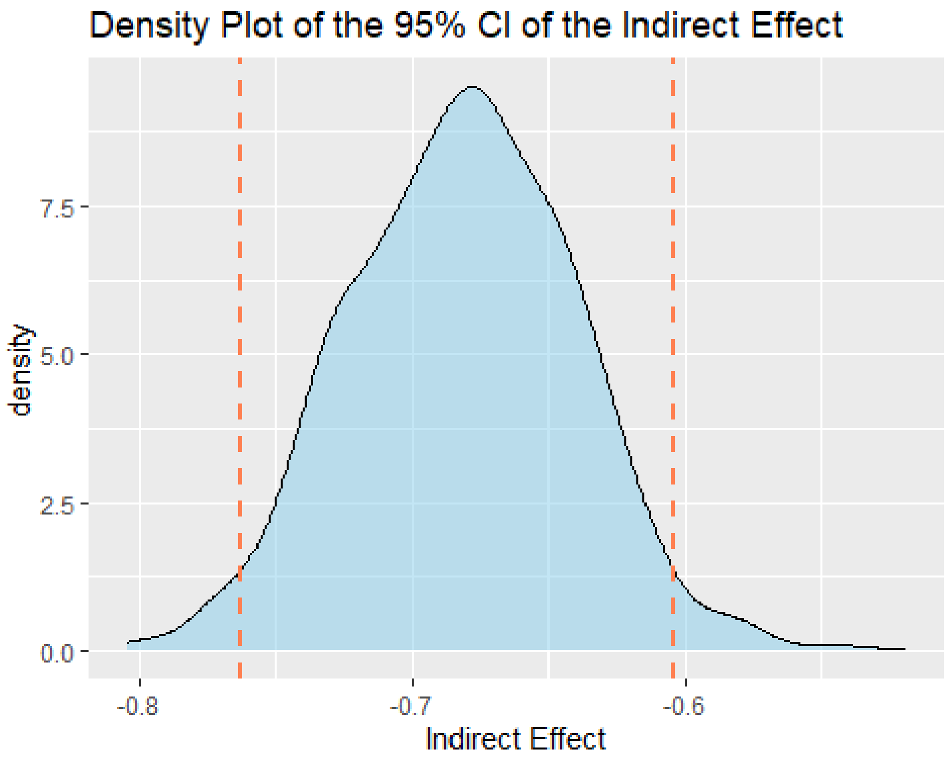
```
## Warning: 套件 'boot' 是用 R 版本 4.2.2 來建造的
```

```
# Define bootstrap function
boot_func <- function(data, indices) {
  data_boot <- data[indices, ]
  model1_boot <- lm(log_weight ~ log_cylinders, data = data_boot)
  model2_boot <- lm(log_mpg ~ log_weight + log_cylinders + log_acceleration + model_year
+ origin, data = data_boot)
  indirect_effect_boot <- coef(model1_boot)["log_cylinders"] * coef(model2_boot)["log_wei
ght"]
  return(indirect_effect_boot)
}
# Bootstrap Models 1 & 2 and compute the indirect effect each time
set.seed(123)
boot_results <- boot(cars_log, boot_func, R = 1000)

## 95% CI of the indirect effect: -0.7631369 to -0.6045036
```

ii. Show a density plot of the distribution of the 95% CI of the indirect effect

```
ggplot(data.frame(Indirect_Effect = boot_results$t), aes(x = Indirect_Effect)) +
  geom_density(fill = "skyblue", alpha = 0.5) +
  geom_vline(aes(xintercept = boot_ci$percent[4]), color = "coral", linetype = "dashed",
linewidth = 1) +
  geom_vline(aes(xintercept = boot_ci$percent[5]), color = "coral", linetype = "dashed",
linewidth = 1) +
  labs(title = "Density Plot of the 95% CI of the Indirect Effect", x = "Indirect Effect")
 +
  theme_grey()
```


Density Plot of the 95% CI of the Indirect Effect
density
Indirect Effect